Package 'Sequential'

April 21, 2026

Type Package

Title Exact Sequential Analysis for Poisson and Binomial Data

Version 4.6.0

Date 2026-04-20

Maintainer Ivair Ramos Silva <ivair@ufop.edu.br>

Description Functions to calculate exact critical values, statistical power, expected time to signal, and required sample sizes for performing exact sequential analysis. All these calculations can be done for either Poisson or binomial data, for continuous or group sequential analyses, and for different types of rejection boundaries. In case of group sequential analyses, the group sizes do not have to be specified in advance and the alpha spending can be arbitrarily settled. For regression versions of the methods, Monte Carlo and asymptotic methods are used.

License GPL-2

LazyLoad yes

Imports boot, pmultinom, maxLik

NeedsCompilation no

Author Ivair Ramos Silva [aut, cre],
Martin Kulldorff [aut]

Repository CRAN

Date/Publication 2026-04-21 05:10:40 UTC

Contents

`Sequential-package` *Analysis Support, Critical Values, Power, Time to Signal and Sample Size for Sequential Analysis with Poisson and Binomial Data.*

Description

`Sequential` is designed for continuous and group sequential analysis, where statistical hypothesis testing is conducted repeatedly on accumulating data that gradually increases the sample size. This is different from standard statistical analysis, where a single analysis is performed using a fixed sample size. It is possible to analyze either Poisson type data or binomial 0/1 type data. For binomial data, it is possible to incorporate an off-set term to account for variable matching ratios. For Poisson data, the critical value is based on a Wald-type upper boundary, which is flat on the scale of the log-likelihood ratio, and on a predetermined maximum sample size. Alternatively, it is also possible to apply a user defined alpha spending function in order to specify non-flat signaling thresholds. For group sequential analyses, there are functions for pre-specified group sizes and for the situation when the group sizes are not known a priori. It is also possible to perform mixed continuous/group sequential analysis, where, for example, there is at first a big batch of data that arrives in one group, followed by continuous sequential analysis. All non regressional results are exact, based on iterative numerical calculations, rather than asymptotic theory or computer simulations. For regression versions of the methods, Monte Carlo and asymptotic methods are used. In the `Sequential` package, there are functions to calculate critical values, statistical power, expected

time to signal, and expected sample size at the end of the sequential analyses whether the null hypothesis was rejected or not. For example, for any desired power, relative risk and alpha level, the package can calculate the required upper limit on the sample size (maximum length of surveillance), the critical value needed, and the corresponding expected time to signal when the null hypothesis is rejected.

Details

Package:	Sequential
Type:	Package
Version:	4.6.0
Date:	2026-04-20
License:	GPL 2
LazyLoad:	yes

Index:

`Performance.Threshold.Binomial`	Power, Expected Time to Signal, Expected Sample Size and Alpha Spending for User Defined Threshold with Binomial Data.
`Performance.Threshold.CondPoisson`	Power, Expected Time to Signal, Expected Sample Size and Alpha Spending for User Defined Threshold with Conditional Poisson Data.
`Performance.Threshold.Poisson`	Power, Expected Time to Signal, Expected Sample Size and Alpha Spending for User Defined Threshold with Poisson Data.
`Analyze.Binomial`	Function to Conduct Group Sequential Analyses for Binomial Data When the Goup Sizes are not Known a Priori.
`AnalyzeRegression.Binomial`	Function to Conduct Group Sequential Regression Analyses for Binomial Data When the Goup Sizes are not Known a Priori.
`AnalyzeSetUpRegression.Binomial`	Function to Set Up the Input Parameters Before Using the `AnalyzeRegression.Binomial` Function for the First Time.
`AnalyzeSetUp.Binomial`	Function to Set Up the Input Parameters Before Using the `Analyze.Binomial` Function for the First Time.
`Analyze.Multinomial`	Function to Conduct Group Sequential Analyses for multinomial Data When the Goup Sizes are not Known a Priori.
`AnalyzeSetUp.Multinomial`	Function to Set Up the Input Parameters Before Using the `Analyze.Multinomial` Function for the First Time.
`Analyze.wBinomial`	Function for group sequential analyses of multiple weighted binomial endpoints.
`AnalyzeSetUp.wBinomial`	Function to set up input parameters before using the `Analyze.wBinomial` function for the first time.
`Analyze.Poisson`	Function to Conduct Group Sequential Analyses for Poisson Data When the Goup Sizes are not Known a Priori.
`AnalyzeRegression.Poisson`	Function to Conduct Group Sequential Regression Analyses for Poisson Data When the Goup Sizes are not Known a Priori.
`AnalyzeSetUpRegression.Poisson`	Function to Set Up the Input Parameters Before Using the

	AnalyzeRegression.Poisson Function for the First Time.
AnalyzeSetUp.Poisson	Function to Set Up the Input Parameters Before Using the Analyze.Poisson Function for the First Time.
Analyze.CondPoisson	Function to Conduct Group Sequential Analyses for Conditional Poisson Data When the Goup Sizes are not Known a Priori.
AnalyzeRegression.CondPoisson	Function to Conduct Group Sequential Regression Analyses for Conditional Poisson Data When the Goup Sizes are not Known a Priori.
AnalyzeSetUp.CondPoisson	Function to Set Up the Input Parameters Before Using the Analyze.CondPoisson Function for the First Time.
AnalyzeSetUpRegression.CondPoisson	Function to Set Up the Input Parameters Before Using the AnalyzeRegression.CondPoisson Function for the First Time.
ConfidenceInterval.Binomial	Confidence interval for the relative risk following a sequential test with binomial data.
CV.Binomial	Critical Values for Group and Continuous Sequential Analysis with Binomial Data.
CV.Poisson	Critical Values for Group and Continuous Sequential Analysis with Poisson Data.
CV.CondPoisson	Critical Values for Continuous Sequential CMaxSPRT for Limited Information from Historical Cohort Using Conditional Poisson Data.
Optimal.Binomial	Optimal alpha spending for continuous and group sequential analysis with binomial data.
Performance.Binomial	Power, Expected Time to Signal and Expected Sample Size for Group and Continuous Sequential Analysis with Binomial Data.
Performance.Poisson	Power, Expected Time to Signal and Expected Sample Size for Continuous Sequential Analysis with Poisson Data.
Performance.CondPoisson	Power, Expected Time to Signal and Expected Sample Size for Continuous Sequential CMaxSPRT Using Limited Information from Historical Cohort with Conditional Poisson Data.
SampleSize.Binomial	Sample Size Calculation for Continuous Sequential Analysis with Binomial Data.
SampleSize.Poisson	Sample Size Calculation for Continuous Sequential Testing with Poisson Data.
SampleSize.CondPoisson	Sample Size Calculation for Continuous Sequential CMaxSPRT with Poisson Data.
Performance.AlphaSpend.Binomial	Power, Expected Time to Signal, Expected Sample Size and Threshold for User Defined Alpha Spending with Binomial Data.
Performance.AlphaSpend.CondPoisson	Power, Expected Time to Signal, Expected Sample Size and Threshold for User Defined Alpha Spending with Conditional Poisson Data.
Performance.AlphaSpend.Poisson	Power, Expected Time to Signal, Expected Sample Size and Threshold for User Defined Alpha Spending with Poisson Data.

Overview

Most of the sequential analysis methods found in the literature are based on asymptotic results. In contrast, this package contains functions for the exact calculation of critical values, statistical power, expected time to signal when the null is rejected and the maximum sample size needed when the null is not rejected. This is done for Poisson and binomial type data with a Wald-type upper boundary, which is flat with respect to the likelihood ratio function, and a predetermined upper limit on the sample size. For a desired statistical power, it is also possible to calculate the latter. The motivation for this package is post-market near real-time drug and vaccine safety surveillance, where the goal is to detect rare but serious safety problems as early as possible, in many cases after only a hand full of adverse events. The package can also be used in other application areas, such as clinical trials.

A tutorial paper by Silva et al. (2021) explains the main features of the package up to version 3.3.1, where sequential test designing and real data analyses are dicussed in more details.

The basis for this package is the Maximized Sequential Probability Ratio Test (MaxSPRT) statistic (Kulldorff et al., 2011), which is a variant of Wald's Sequential Probability Ratio Test (SPRT) (Wald, 1945,47). MaxSPRT uses a composite alternative hypothesis, and upper boundary to reject the null hypothesis when there are more events than expected, no lower boundary, and an upper limit on the sample size at which time the sequential analyses end without rejecting the null. MaxSPRT was developed for post-market vaccine safety surveillance as part of the Vaccine Safety Datalink project run by the Centers for Disease Control and Prevention.

In this package, all critical values, alpha spending strategies, statistical power, expected time to signal and required sample size to achieve a certain power, are obtained exactly to whatever decimal precision desired, using iterative numerical calculations. None of the results are based on asymptotic theory or computer simulations.

Poisson Data

To start, consider continuous sequential analysis for Poisson data. Let C_t be the random variable that counts the number of events up to time t. Suppose that, under the null hypothesis, C_t has a Poisson distribution with mean μ_t, where μ_t is a known function reflecting the population at risk. Under the alternative hypothesis, suppose that C_t has a Poisson distribution with mean RRμ_t, where "RR" is the unknown increased relative risk due to the vaccine. The MaxSPRT statistic defined in terms of the log likelihood ratio is given by:

$$LLR_t = (\mu_t - c_t) + c_t \log c_t/\mu_t,$$

when c_t is at least μ_t, and $LLR_t = 0$, otherwise. For continuous sequential analysis, the test statistic, LLR_t, is monitored at all times $t \in (0, T]$, where T= SampleSize. SampleSize is defined a priori by the user in order to achieve the desired statistical power, which can be calculated using the `SampleSize.Poisson` function. The sequential analysis ends, and H_0 is rejected if, and when, $LLR_t \geq CV$, where CV is calculated using the `CV.Poisson` function. If μ_t= SampleSize, the sequential analysis ends without rejecting the null hypothesis. To calculate other important performance metrics, such as the expected time to signal when the null hypothesis is rejected, use the `Performance.Poisson`, `Performance.Threshold.Poisson` and `Performance.AlphaSpend.Poisson`

functions. These functions also work for group sequential analysis when the group sizes are fixed a priori. For with fixed and/or unpredictable group sizes, see the function `Analyze.Poisson`.

If the first event occurs sufficiently early, the sequential analysis may end with the null hypothesis rejected after a single event. There is an option to require a minimum number of observed events, c_t= M, before the null can be rejected. Setting M in the range [3,6] is often a good choice (Kulldorff and Silva, 2012). If there is a delay until the sequential analysis starts, but it continuous continuously thereafter, there is an option for that as well, requiring a minimum number μ_t= D of expected events before the null can be rejected.

With continuous sequential analysis, investigators can repeatedly analyze the data as often as they want, ensuring that the overall probability of falsely rejecting the null hypothesis at any time during the analysis is controlled at the desired nominal significance level (Wald, 1945, 1947). Continuous sequential methods are suitable for real-time or near real-time monitoring. When data is only analyzed intermittently, group sequential methods are used instead (Chin, 2012; Cook and DeMets, 2007; Xia, 2007; Friedman et al., 2010; Ghosh and Sen, 1991; Jennison and Turnbull, 2000; Mukhopadhyay and Silva, 2002; Whitehead, 1997). The data is then analyzed at regular or irregular discrete time intervals after a certain amount of data is accessible. Group sequential statistical methods are commonly used in clinical trials, where a trial may be stopped early due to either efficacy or unexpected adverse events (Jennison and Turnbull, 2000).

The same test statistic, LLR_t, is used for group sequential analyses (Silva and Kulldorff, 2012). The times when LLR_t is evaluated can be defined in several ways, using regular or irregular time intervals that are referenced by calendar period, sample size or some scale involving the distribution of the data. For Poisson data, the group sequential analysis must be conducted with equal size groups, with a constant expected number of adverse events between looks at the accumulating data. In another words, LLR_t is compared against CV whenever μ_t is a multiple of SampleSize/Looks, where 'Looks' is the total number of looks at the data.

Binomial Data

The MaxSPRT method can also be applied to binomial/Bernoulli data. Let n denote the total number of events that has been observed in a sequential monitoring up to a certain moment in time. Suppose that these n events are categorized as cases and controls. For example, cases may be adverse events happening to a person taking drug A, while controls may be the same adverse event happening to someone in a matched set of individuals taking drug B. As another example, in a self-control sequential analysis, cases may be adverse events happening during the 1-28 days following vaccination, while controls are the same adverse events happening 29-56 days after vaccination.

Let C_t to denote the number of cases among the n events, and assume that C_t follows a binomial distribution with success probability equal to p, where $p = 1 = (1 + z)$, and z is the matching ratio between the occurrence of a case and of a control under the null hypothesis. For example, if the probability of having a case (instead of a control) is $p = 1 = (1 + z) = 0.5$, then z=1 (1:1 matching ratio), or, $p = 0.25$ for z=3 (1:3 matching ratio), etc.

The MaxSPRT statistic (Kulldorff et al., 2011) for a continuous binomial surveillance is:

$$LR_n = \frac{(c_n/n)^{c_n}\left[(n-c_n)/n\right]^{n-c_n}}{\left[1/(1+z)\right]^{c_n}\left[z/(1+z)\right]^{n-c_n}},$$

if $zc_n/(n - c_n) > 1$, and $LR_n = 1$ otherwise.

The monitoring is continued until either there is a signal rejecting the null hypothesis ($LR_n > CV$) or until $n = N$, which indicates that the null is not to be rejected. To perform the calculations, use

the `CV.Binomial`, `SampleSize.Binomial`, `Performance.Binomial`, `Performance.Threshold.Binomial` and `Performance.AlphaSpend.Binomial` functions for continuous and group sequential fashions. For group sequential analysis, the group sizes are fixed a priori. For fixed and/or unpredictable group sizes, see the function `Analyze.Binomial`.

The main assumptions behind the method above are: (i) the monitoring is truly performed in a continuous fashion; (ii) the matching ratio (z) is constant for all of the n events, and (iii) it uses a Wald type rejection boundary that is flat in terms of the likelihood function. Relaxing these assumptions, Fireman et al. (2013) developed exact sequential analysis for group sequential data with varying matching ratios, and for any user specified alpha rejection plan.

Multiple Weighted Binomial Endpoints

When there are multiple different outcomes, one can use weights reflecting practical interpretations, such as an outcome severity in comparison to the others. For example, for two different outcomes, the first with weight w, and the second with weight 1. This way, a single event of the first outcome is equivalent to w outcomes of the second type. This type of analysis is treated by Silva et al (2019). To run the analysis with multiple weighted binary endpoints with fixed and/or unpredictable group sizes, see the function `Analyze.wBinomial`.

Conditional Poisson data with limited information from historical cohort - CMaxSPRT

In Poisson MaxSPRT, the expected mean μ_t is assumed to be a known function reflecting the baseline adverse event risk in the absence of the exposure of interest. In practice, it is estimated with historical data and the uncertainty associated with the estimated counts may or may not have a non-negligible impact on the performance of the sequential analysis method. Li and Kulldorff (Li and Kulldorff, 2010) showed in their simulation study that uncertainty in the estimated baseline means can be ignored when the total number of events in the historical data is at least 5 times the specified upper limit T. Otherwise, it is recommended to implement a statistical procedure that takes in account the variation from the historical data, i.e. a procedure that conditionates the likelihood function of the historical Poisson data, here simply called "conditional Poisson data". For this, the Conditional Maximized Sequential Probabilit Ratio Test (CMaxSPRT) to account for variation in both the historical and surveillance cohorts.

Let c and V denote the total number of events and the cumulative person-time in the historical data, let P_k denote the cumulative person-time observed in the surveillance population when the kth event occurred. The CMaxSPRT statistic defined in terms of the log likelihood ratio is given by

$$U_k = clog(\frac{c(1+P_k/V)}{c+k}) + klog(\frac{k(1+P_k/V)}{P_k/V(c+k)}),$$

when $k/c > P_k/V$, and $U_k = 0$, otherwise. In the original publication (Li and Kulldorff, 2010), the method was introduced as a continuous sequential analytic approach with the upper limit defined in terms of the maximum number of observed events, i.e., $k \leq K$, and the critical value calculated via a Monte Carlo approach. A large number of Monte Carlo simulations (e.g., 10 million) might be needed to calculate the critical values with a reasonable precision.

In Silva et al. (2019a), the method was extended i) with another option of defining the surveillance length in terms of the maximum cumulative person-time divided by the total cumulative person-time in the historical cohort, i.e., $P_k/V \leq T$, ii) with an exact calculation of the critical values for both surveillance length definitions, and iii) for group sequential analysis with data updated and analyzed intermittently instead of continuously. The exact critical values are calculated using the interval havling method to solve for the root of a complex, non-linear equation such that the overall

Type I error rate is preserved at the nominal level. As K increases, the computing time for the exact critical values increases exponentially.

Silva et al. (2019a) also proposed two approximation methods to calculate the critical values that require substantially less computing time. One approch may overestimate the critical values and thus is referred to as the conservative approach as it may yield lower-than-nominal Type I error rates; the other approach may underestmate the critical values and thus is referred to as the liberal approach as it may yield higher-than-nominal Type I error rates. The recommendation is to use the exact approach when K is small (e.g., 10), use the conservative approach when K is medium or large but c is small, and use the liberal approach when c is medium (e.g., 50) or large. Exact calculations for selected tuning parameters show that the three approaches yield very similar results when K and c are reasonably large.

For calculating critical values for a Wald type rejection boundary, use the `CV.CondPoisson` function. For statistical power, expected time to signal, expected time of surveillance, and maximum sample size requirements, use the `Performance.CondPoisson`, `Performance.Threshold.CondPoisson`, `Performance.AlphaSpend.CondPoisson`, and `SampleSize.CondPoisson` functions. For fixed and/or unpredictable group sizes, see the function `Analyze.CondPoisson`.

Multinomial Data

The function `Analyze.Multinomial` performs continuous or group sequential analysis for multinomial data. It can also be used for mixed continuous-group sequential analysis where some data arrives continuously while other data arrives in groups. There is (i) no need to pre-specify the group sizes before the sequential analysis starts, and (ii) a variety of alpha spending functions are available.

The function `Analyze.Multinomial` runs the adaptive sequential multiple hypotheses testing for seasonal concomitant vaccines safety surveillance introduced by Silva and Maro(2025a), which was developed for rapidly detecting increased risks of adverse events from one or more combinations of simultaneous vaccine exposure. As explained by Silva and Maro(2025a), multiple seasonal vaccines may be recommended for some strata of the population defined by covariates, such as gender and age, for example. Such vaccines can be administered simultaneously or at least within the same vaccination season. Thus, monitoring adverse events origination from multiple vaccines may be more informative and statistically powerful than monitoring each vaccine separately.

The method is an extension of the binomial MaxSPRT method to a multinomial exposure model. With an exact analytical alpha spending approach, the computationally feasible limit of multiple exposures is likely limited to no more than two vaccines. For more complex situations with several vaccines (>2), which can multiple types of endpoints (i.e. designated adverse events), a valid Monte Carlo decision rule is constructed.

To adjust for covariates confounders, `Analyze.Multinomial` utilizes the maxLik package (Henningsen et al., 2011) to compute the MLE of the baseline covariates effects as derived by Silva and Maro(2025).

Alpha spending function for unpredictable group sizes

The alpha spending function specifies the cumulative amount, $F_\alpha(t)$, of Type I error probability related to each of the possible values of n. Thus, at the end of the monitoring the alpha spending corresponds to a value smaller than or equal to the overall amount of Type I error probability defined for the overall nominal significance level, α.

Denote the single probability of rejecting the null hypothesis at the j-th test by α_j. Then, the alpha spending at test i is given by $F_\alpha(t_i) = \sum_{j=1}^{i} \alpha_j \leq \alpha$.

There is a vast number of proposals for choosing the shape of the alpha spending function. Jennison and Turnbull (2000) present a rich discussion about this topic. They dedicated a special attention to the alpha spending of the form: $F_\alpha(t) = \alpha t^\rho$, where $\rho > 0$, and t represents a fraction of the maximum length of surveillance.

A new aproach for alpha spending selection is the optimal solution, introduced by Silva and Kulldorff (2020). The optimal solution is based on exact calculations through linear programing, and it is operated through the function `Optimal.Binomial`.

For MaxSPRT with Poisson data, Silva (2018b) presents a rule of thumb for a near-optimal alpha spending shape. Likewise, Silva et al. (2019b) discuss the choice of alpha spending shapes for CMaxSPRT.

For multinomial data, the alpha spending plan is designed in a way to ensure that a target statistical power, gamma, is endured for detecting a target increased risk, R1, for each multinomial entry. This is possible by means of a new concept introduced by Silva and Maro(2025a), the robust alpha spending plan. For more details about the method and the construction of the critical values, see Silva and Maro(2025a).

To run continuous or group sequential analysis with an user defined alpha spending function, and/or, when the group sizes are not known a prior, `Analyze.Binomial`, `Analyze.Multinomial`,`Analyze.wBinomial` (for multiple weighted binomial endpoints), `Analyze.Poisson`, and `Analyze.CondPoisson` should be used for binomial and Poisson, and conditional Poisson data, respectively. These functions work differently than the other functions mentioned above. Those other functions are designed to be used before the start of the sequential analysis, in order to determine what the maximum sample size and critical value should be. Once the sequential analysis is under way, the test statistic is then calculated using a hand calculator or an excel spread sheet, and compared with the critical value. The functions `Analyze.Binomial`, `Analyze.Multinomial`, `Analyze.Poisson`, and `Analyze.CondPoisson` work very differently, in that they are run at each look at the accumulating data, whenever a new group of data arrives, and it is meant to perform the test itself, i.e., there is no need to use hand calculators or excel spread sheets or any other auxiliar code. The results and conclusions, including a descriptive table and illustrative graphics, are automatically provided after running `Analyze.Binomial`, `Analyze.Multinomial`, `Analyze.wBinomial`, `Analyze.Poisson`, or `Analyze.CondPoisson`.

Important: before using these functions, though, it is necessary to first run the functions `AnalyzeSetup.Binomial`, `AnalyzeSetup.Multinomial`,`AnalyzeSetup.wBinomial`, `AnalyzeSetup.Poisson`, or `AnalyzeSetup.CondPoisson` once in order to set everything up for the sequential analysis.

Comparison with Other R Packages for Sequential Analysis

The R `Sequential` package is designed for sequential analysis where statistical hypothesis testing is performed using gradually accumulating data. It is not designed for quality control problems, where a process is monitored over time to detect an emerging problem due to a sudden increase in the excess risk. Although the methods for sequential analysis and quality control may seem similar, as they both analyze gradually accumulating data, they are actually very different in both their purpose and design. Under the sequential hypothesis testing approach, the objective is to quickly determine if there is some intrinsic excess risk, with the assumption that this risk does not change over time. For example, we may want to know if drug A is better than drug B, and there is no reason to believe that the behavior of the drugs change over time. In the quality control setting, the objective is instead to detect a possible change in a stochastic process that may occur in the future, and to detect that change as soon as possible after it occurs. For example, the heart of a hospital

patient is beating as it should, but if there is a sudden deterioration, the alarm should sound as soon as possible without generating a lot of false alarms. This package is only meant for sequential analysis of the former type, and it should not be used for quality control type problems. For quality control type analyses, there are other R packages available, such as `graphicsQC`, `IQCC`, `MetaQC`, `MSQC`, `qcc`, and `qcr`.

In a number of ways, the R Sequential package differs from other R packages for sequential analyses. Historically, most sequential analysis has been conducted using asymptotic statistical theory, and that is also what is used in the `gsDesign`, `ldbounds`, `PwrGSD`, `seqDesign`, `seqmon`, and `sglr` R packages. In contrast, the R Sequential package is based on exact results, using iterative numerical calculations, rather than using asymptotic theory or computer simulations.

With this package, it is only possible to analyze binomial/Bernoulli, Poisson, or conditional Poisson data. For other probability distributions, such as normal or exponential data, other R packages should be consulted, such as `GroupSeq` or `SPRT`. Moreover, all functions in this package uses a one-sided upper bound to reject the null hypothesis, while the analyses end without rejecting the null when an upper limit on the sample size is reached. For two sided sequential analysis, or other types of rejection boundaries, other R packages must be used, such as e.g. `ldbounds` and `Binseqtest`. Finally, in this package, there are functions for both continuous and group sequential analysis, and it is also possible to analyze situations where some of the data arrives continuously while other parts of the data arrives in groups. Most other R packages are exclusively designed for group sequential analysis, but there are some that also do continuous sequential analysis, such as `Binseqtest` and `SPRT`, but `Binseqtest` is only for binomial data type, and `SPRT` is for simple alternative hypothis, while `Sequential` can be used for binomial and Poisson data and is meant to composite alternative hypothesis. The present package offers the possibility to calculate the expected time to signal through the `Performance.Poisson`, `Performance.G.Poisson`, `Performance.Binomial`, `Performance.G.Binomial`, and Performance.CondPoisson functions, which is not offered by the other packages cited above. Another important advantage of the `Sequential` package is the possibility of eliciting, through exact calculations, the minimum sample size needed to accomplish with target statistical powers through the functions `SampleSize.Poisson`, `SampleSize.CondPoisson`, and `SampleSize.Binomial`.

Acknowledgements

Development of the R Sequential package has been funded and supported by:
- Food and Drug Administration, USA, through the Mini-Sentinel Project (v1.0,1.1,2.0).
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0 to 3.1).
- Federal University of Ouro Preto (UFOP), through contract under internal UFOP's resolution CEPE 4600 (v2.0 to 3.1).
- National Council of Scientific and Technological Development (CNPq), Brazil, process number 301391/2019-0. (v3.1 to v4.4).
- National Council of Scientific and Technological Development (CNPq), Brazil, process number 302882/2022-7. (v4.3.1).
- Support Foundation to Minas Gerais State Research-Fapemig, Brazil, grant numbers PQ-00787-21 and RED-00133-21. (v3.1 to v4.3).
- Bank for Development of the Minas Gerais State (BDMG), Brazil (v1.0).
- Vaccine Safety Datalink - VSD Infrastructure Activities project at HPHCI. (v4.3.4).
- Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute (v4.4).

Feedback from users is greatly appreciated. Very valuable suggestions concerning the R Sequential package have been received from various individuals, including:
- Ron Berman, University of California Berkeley.
- Claudia Coronel-Moreno, Harvard Pilgrim Health Care Institute.
- Bruce Fireman, Kaiser Permanente Northern California.
- Josh Gagne, Harvard Medical School and Brigham and Women's Hospital.
- Ned Lewis, Kaiser Permanente Northern California.
- Judith Maro, Harvard Medical School and Harvard Pilgrim Health Care Institute.
- Azadeh Shoaibi, Food and Drug Administration.
- Katherine Yih, Harvard Medical School and Harvard Pilgrim Health Care Institute.
- Jie Tang, Clinical biostatistics, Janssen R and D US, Johnson and Johnson LLC.
- Tuomo A. Nieminen, The National Institute for Health and Welfare (THL), Finland.
- Andreia Leite, Department of Infectious Disease Epidemiology, London School of Hygiene and Tropical Medicine.
- Laura Hou, Harvard Pilgrim Health Care Institute, Boston, USA.
- Abdurrahman Abdurrob, Division of Pharmacoepidemiology and Pharmacoeconomics, Department of Medicine, Brigham and Women's Hospital and Harvard Medical School.
- Kirk Snyder, Information Management Services, Inc.
- Joselito M. Montalban, Active Living, Population and Public Health Branch, Winnipeg, Manitoba, Canada.
- Sarah Spruin and Vaccine Safety Surveillance, Public Health Agency of Canada.
- Lola Adewale, Stateam LLC, New Jersey, United States.
- Lian Lin, Ph.D., Director, Safety Statistics, Stats and Programming, Moderna.

Version History of the R Sequential Package

Version 1.1, February 2013
Exact sequential analysis for Poisson data:
- Exact continuous sequential analysis.
- Exact group sequential analysis with pre-defined and constant groups sizes.
- Wald type rejection boundary.
- Statistical power, expected time to signal and sample size calculations.
- User guide.

Version 1.2, January 2014
- Improved code structure and efficiency.
- More extensive user guide.

Version 2.0, June 2015
Exact sequential analysis for binomial data:
- Continuous sequential analysis.
- Group sequential analysis with pre-defined group sizes.
- Group sequential analysis with unpredictable group sizes, not specified a priori.
- Fixed or variable binomial probabilities (matching ratios).
- User specified alpha spending function.
- Statistical power, expected time to signal and sample size calculations.
- Updated user guide.

Version 2.0.1, June 2015
- Correction of bugs in `CV.Poisson` function.
- Updated user guide.

Version 2.0.2, October 2015
- Improved user guide.

Version 2.1, May 2016
Exact sequential analysis for Poisson data:
- Group sequential analysis with unpredictable group sizes, not specified a priori.
- User specified alpha spending function.
- Mixed group-continuous sequential analysis.
- Statistical power, expected time to signal and sample size calculations for non-constant groups sizes.
Other:
- Directory address parameter in AnalyzeSetUp functions.
- Probability parameter in binomial functions.
- Updated user guide.

Version 2.1.1, June 2016
- Correction of bugs in Poisson functions.
- Updated user guide.

Version 2.2, July 2016
- Critical Value, Performance, and SampleSize calculations for CMaxSPRT with Poisson data.
- Updated user guide.

Version 2.2.1, September 2016
- Correction of bugs in `CV.Poisson` and `CV.G.Poisson` functions.
- Updated user guide.

Version 2.3, Dec 2016
- Correction of bugs in the `SampleSize.Binomial` function.
- Improvement of SampleSize functions for considering vectors for the input parameters R and power.
- Inclusion of the new functions `AnalyzeSetUp.CondPoisson` and `Analyze.CondPoisson`.
- Updated user guide.

Version 2.3.1, Feb 2017
- Correction of bugs in `Analyze.Binomial` and `AnalyzeSetUp.Poisson` functions.
- Adjustment on the relative risk estimation method for `Analyze.Binomial` function.
- Updated user guide.

Version 2.3.2, Aug 2017
- Correction of bugs in `Analyze.Binomial`.
- Updated user guide.

Version 3.0, Jan 2019
- Functions `Optimal.Binomial` and `Analyze.wBinomial`.
- Updated user guide.

Version 3.0.1, Feb 2019
- Updated user guide.

Version 3.1, Sept 2019
- New Functions `Performance.Threshold.Binomial`, `Performance.Threshold.CondPoisson`, `Performance.Threshold.Poisson`, `Performance.AlphaSpend.Binomial`, `Performance.AlphaSpend.CondPoisson`, `Performance.AlphaSpend.Poisson`.
- Implemented two-tailed testing for the functions: `CV.Poisson`, `Performance.Poisson`, `SampleSize.Poisson`, `Analyze.wBinomial`, `Optimal.Binomial`, `Performance.Threshold.Binomial`, `Performance.AlphaSpend.Binomial`, `Performance.Threshold.Poisson`.
- Updated user guide.
- Concatenation of functions for continuous and group fashions for critical values and performance.

Version 3.2, Nov 2020
- Function `Analyze.Binomial` improved for fixed-width and fixed-accuracy confidence intervals,
- Function `Performance.AlphaSpend.Binomial` improved for new inputs having lower and upper alpha spending in two tailed testing,
- Correction of bugs in `Analyze.Poisson` and `CV.Poisson` functions,
- Updated user guide.

Version 3.2.1, Dec 2020
- Correction of bugs in the `Analyze.Poisson` function,
- Updated user guide.

Version 3.3, Feb 2021
- Optimized `Analyze.Poisson` function for running three times faster than earlier versions.
- Updated user guide.

Version 3.3.1, Feb 2021
- Adjusted `Analyze.Binomial` function for a flexible H0:R<=R0 under the null hypothesis.
- Updated user guide.

Version 3.3.2, Aug 2021
- Adjusted functions: `Analyze.Poisson`, `AnalyzeSetUp.Poisson`, `Performance.AlphaSpend.Poisson`, and `SampleSize.Poisson` for the surveillance of COVID-19 vaccination.
- Updated user guide.

Version 3.3.3, Aug 2021
- Adjusted `Performance.AlphaSpend.Poisson` function.
- Updated user guide.

Version 3.4, Sept 2021
- Adjusted `Analyze.Poisson` function.

- Updated user guide.

Version 4.0, Sept 2021
- New function `ConfidenceInterval.Binomial`, adjusted `Analyze.wBinomial` and new parameter R0 for `Analyze.Poisson` and `Performance.AlphaSpend.Poisson`.
- Updated user guide.

Version 4.1, Oct 2021
- Adjusted `SampleSize.Poisson`.
- Updated user guide.

Version 4.2, Feb 2022
- Adjusted `AnalyzeSetUp.Poisson`.
- Updated user guide.

Version 4.3, April 2022
- Adjusted `Analyze.Binomial` for bugs on the relative risk estimate under variable matching ratio.
- Updated user guide.

Version 4.3.1, Oct 2023
- Updated reference for the bounded-length confidence interval following sequential analysis with binary data.

Version 4.3.2, Dec 2023
- Correction of bugs in the Analyze.CondPoisson function.

Version 4.3.3, Feb 2024
- Updated user guide.

Version 4.3.4, Oct 2024
- Correction of bugs in theAnalyze.CondPoisson function.

Version 4.4, May 2025
- Inclusion of the Analyze.Multinomial function, inclusion of confidence intervals in the output tables of the Analyze.Binomial, Analyze.Poisson and Analyze.CondPoisson functions, and management of unstable data by means of a robust alpha spending approach.

Version 4.5, October 2, 2025
- Adjustments of input parameters for the Analyze.Multinomial function.

Version 4.5.1, October 29, 2025
- Adjustments of input parameters for the Analyze.Multinomial function.

Version 4.5.2, November 30, 2025
- Adjustments of output tables for the Analyze.Multinomial function.

Version 4.6.0, April 20, 2026
- Adjustments of output tables for the Analyze.Multinomial function.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.
Maintainer: Ivair Ramos Silva <ivair@ufop.edu.br>

References

Chin R. (2012), Adaptive and Flexible Clinical Trials, Boca Raton, FL: Chapman and Hall/CRC.

Cook TD, DeMets DL. (2007), Introduction to Statistical Methods for Clinical Trials: Chapman and Hall/CRC Texts in Statistical Science.

Fireman B, et al. (2013), Exact sequential analysis for binomial data with timevarying probabilities. Manuscript in Preparation.

Friedman LM, Furberg CD, DeMets D. (2010), Fundamentals of Clinical Trials, 4th ed.: Springer.

Ghosh BK, Sen PK. (1991), Handbook of Sequential Analysis, New York: MARCEL DEKKER, Inc.

Ghosh M, Mukhopadhyay N, Sen PK. (2011), Sequential Estimation: Wiley.

Henningsen A, Toomet O (2011), maxLik: A package for maximum likelihood estimation in R. Computational Statistics, 26(3), 443–458.

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011), A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30: 58–78.

Kulldorff M, Silva IR. (2015), Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Li L, Kulldorff M. (2010), A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

Mukhopadhyay N, Silva BM. (2002), Sequential Methods and Their Applications, 1th ed.: Chapman and Hall/CRC.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71(3), 851–858.

Silva IR, Maro J, Kulldorff M. (2021), Exact sequential test for clinical trials and post-market drug and vaccine safety surveillance with Poisson and binary data. Statistics in Medicine, 40(22), 4890–4913.

Silva IR, Maro J. (2025a), Adaptive Sequential Multiple Hypotheses Testing for Seasonal Concomitant Vaccines Safety Surveillance. Working paper, Sentinel (TIDE), Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Silva IR, Maro J. (2025b), Robust Alpha Spending for Unstable Data. Working paper, Sentinel (TIDE), Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Silva IR, Montalban, J., Oliveira, F. (2025c), Maximized Sequential Probability Ratio Test Regression. Working paper - Sentinel (TIDE) project, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Silva IR, Kulldorff M, Yih W. Katherine. (2020), Optimal alpha spending for sequential analysis with binomial data. Journal of the Royal Statistical Society Series B, 82(4) p. 1141–1164.

Silva IR, Gagne J, Najafzadeh M, Kulldorff M. (2020), Exact Sequential Analysis for Multiple Weighted Binomial Endpoints. Statistics in Medicine, 39(3), 340–351.

Silva IR, Li L, Kulldorff M. (2019a), Exact conditional maximized sequential probability ratio test adjusted for covariates. Sequential Analysis, 38(1), 115–133.

Silva IR. (2018a), Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107–118.

Silva IR. (2018b), Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance With Poisson Data. Methodol Comput Appl Probab, 20(2), 739–750.

Silva IR, Lopes LM, Dias P, Yih WK. (2019b), Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, 38(12), 2126–2138.

Silva IR, Montalban, J. (2023), The person-time ratio distribution for the exact monitoring of adverse events: Historical vs surveillance Poisson data, Statistics in Medicine, 42(18), 3283–3301.

Silva IR, Oliveira, F. (2023), Matching ratio and sample size for optimal sequential testing with binomial data, Statistical Methods in Medical Research, DOI: 10.1177/0962280223117603.

Silva IR, Zhuang, Y. (2022), Bounded-width confidence interval following optimal sequential analysis of adverse events with binary data, Statistical Methods in Medical Research, 31(12), 2323–2337.

Xia Qi. (2007), A Procedure for Group Sequential Comparative Poisson Trials. Journal of Biopharmaceutical Statistics, 17, 869–881.

Wald A. (1945), Sequential Tests of Statistical Hypotheses, Annals of Mathematical Statistics, 16, 117–186.

Wald A. (1947), Sequential Analysis. New York: John Wiley and Sons.

Whitehead J. (1997), The Design and Analysis of Sequential Clinical Trials, 2th ed.: Wiley.

Examples

```
## Critical value for continuous sequential analyses for Poisson Data.
## Maximum sample size = 10, alpha = 0.05 and minimum number of events = 3:

cvt<- CV.Poisson(SampleSize=10,D=0,M=3,alpha=0.05)

## Statistical power and the expected time to signal for relative risk RR=2:

result<- Performance.Poisson(SampleSize=10,D=0,M=3,cv=cvt,RR=2)

# And if you type:
result

# Then you will see the following:
#          Power ESignalTime ESampleSize
#     [1,] 0.7329625    4.071636    5.654732
```

`Analyze.Binomial` *Function for group sequential analyses for binomial data, without the need to know group sizes a priori.*

Description

The function `Analyze.Binomial` is used for either continuous or group sequential analysis, or for a combination of the two. Unlike `CV.Binomial`, it is not necessary to pre-specify the group sizes before the sequential analysis starts. Moreover, under the null hypothesis, the binomial probability, p, can be different for different observations. In a matched case-control setting, this means that the matching ratios can be different for different matched sets. It is possible to use either a Wald type rejection boundary, which is flat with respect to the likelihood ratio, or a user defined alpha spending function. `Analyze.Binomial` is run at each look at the data. Before running it by the first time, it is necessary to run the `AnalyzeSetUp.Binomial` function.

Usage

```
Analyze.Binomial(name,test,z="n",p="n",cases,controls,AlphaSpend="n")
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and it must be the same as the name given by the `AnalyzeSetup.Binomial` function. Should never be the same as another sequential analysis that is run simultaneously on the same computer.
`test`	An integer indicating the number of hypothesis tests performed up to and including the current test. For example, if there were four prior looks at the data, and this is the fifth one, then "test=5". This number should be increased by one each time that the `Analyze.Binomial` function is run for a new group of data, when it is part of the same sequential analysis. If not, there is an error message.
`z`	For a matched case-control analysis, z is the number of controls matched to each case. For example, if there are 3 controls matched to each case, "z=3". In a self-control analysis, z is the ratio of the length of the control interval to the length of the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, "z=7/2". In terms of p, the binomial probability under the null hypothesis, "p=1/(1+z)", or equivalently, "z=1/p-1". The parameter z must be a positive number. The default value is z=1 (p=0.5). If the ratio is the same for all observations, then z can be any positive number. If the ratio is different for different observations, then z is a vector of positive numbers.
`p`	The probability of having a case under the null hypothesis. There is no default value.
`cases`	A number or a vector of the same length as z containing the number of cases.
`controls`	A number or a vector of the same length as z containing the number of controls.

AlphaSpend — The alpha spending function is specified in the AnalyzeSetUp.Binomial function. At any look at the data, it is possible to over ride that pre-specified alpha spending plan by using the AlphaSpend parameter. AlphaSpend is a number representing the maximum amount of alpha (Type I error probabiliy) to be spent up to and including the current test. Because of the discrete nature of the binomial distribution, the actual amount of alpha spent may be less than the maximum amount specified. It must be in the range (0,alpha]. The default value is no override, which means that, if AlphaSpend= "n", then the function will use the alpha spending plan specified in the AnalyzeSetUp.Binomial function.

Details

The function Analyze.Binomial performs continuous or group sequential analysis for Bernoulli or binomial data. It can also be used for mixed continuous-group sequential analysis where some data arrives continuously while other data arrives in groups. Unlike CV.Binomial, there is (i) no need to pre-specify the group sizes before the sequential analysis starts, (ii) a variety of alpha spending functions are available, and (iii) it is possible to include an offset term where, under the null hypothesis, different observations have different binomial probabilities p.

In sequential analysis, data is formed by cumulative information, collected in separated chunks or groups, which are observed at different moments in time. Analyze.Binomial is run each time a new group of data arrives at which time a new sequential test is conducted. When running Analyze.Binomial, only the data from the new group should be included when calling the function. The prior data has been stored, and it will be automatically retrieved by Analyze.Binomial, with no need to reenter that data. Before running Analyze.Binomial for the first time, it is necessary to set up the sequential analysis using the AnalyzeSetUp.Bionimial function, which is run once, and just once, to define the sequential analysis parameters. For information about this, see the description of the AnalyzeSetUp.Binomial function.

The function Analyze.Binomial calculates critical values to determine if the null hypothesis should be rejected or not at each analysis. Critical values are given in the scale of the number of cases. This is done for a pre-specified overall statistical significance level (alpha), and for an upper limit on the sample size (N). The exact analytical solution is obtained through numerical calculations. Based on the data and the critical value, the function determines if the null hypothesis should be rejected or not, and if subsequent tests should be conducted. After each test, the function also provides information about the amount of alpha that has been spent, the cumulative number of cases and controls, and the maximum likelihood estimate of the relative risk.

For binomial and Bernoulli data, there are a number of 0/1 observations that can either be a case or a control. Under the null hypothesis, the probability of being a case is p, and the probability of being a control is 1-p. If data comes from a self-control analysis, the observation is a case if the event occurred in the risk interval, and it is a control if the event occurred in the control interval. Under the null hypothesis, we then have that $p = 1/(1 + z)$, where z is the ratio of the length of the control interval to the length of the risk interval. This ratio, and hence p, does not need to be the same for all observations. If data comes from a matched set of exposed and unexposed individuals, then the observation is a case if the event occurred among one of the exposed, and it is a control if it occurred among one of the unexposed. Under the null hypothesis, $p = 1/(1 + z)$, where z is the number of unexposed individuals divided by the number of exposed individuals in the matched set. Again, this ratio does not have to be the same for all matched sets. The variable z can be any positive number.

If the ratio parameter z, and hence p, is the same for all observations in the same group of data, then z is just a positive number. On the other hand, if different observations in the same group of data have different values for z, then z is a vector, representing multiple z values. For each value of z, it is necessary to specify the number of cases and the number of controls. This means that for a group of data, the vector of zs has to be of the same length as the vector of cases and the vector of controls. The first entry of the vector z is the matching ratio associated to the first entries of cases and of controls. The second entry of z is the matching ratio with respect to the second entries of cases and of controls, and so on. For example, consider that each of five observations came from four different matching ratios. In this situation, the vectors cases, controls and z are all of length four. For example, suppose "z=c(2,1,0.5,3)", "cases=c(1,1,0,0)" and "controls=c(0,0,1,2)". The matching ratio for the first observation, which turned out as a case, is equal to 2. For the second observation, also a case, the matching is equal to 1. With a matching ration of 0.5, the third observation turned out to be a control. The two last observations both had a matching ratio of 3, and both of them were controls. If all observations in the same data group has the same ratio, the vectors are of size one, that is, they are simple numbers. For example, if there were ten observations that all had a ratio of 2, with seven cases and three controls, we have "z=2", "cases=7", and "controls=3".

Alternatively, instead of z the user can specify p directly. Note that only one of these inputs, z or p, has to be specified, but if both are entered the code will only work if z and p are such that p=1/(1+z). Otherwise, an error message will appear to remind that such condition must be complied.

Before running `Analyze.Binomial`, it is necessary to specify a planned default alpha spending function, which is done using the AlphaSpendType parameter in the `AnalyzeSetUp.Binomial` function. The default alpha spending plan can be either, (i) the optimal alpha spending derived by Silva and Kulldorff (2018), which demands users to choose between minimizing expected time to signal or expected sample size, or (ii) the polynomial power-type alpha spending plan, which is parameterized with rho, which, according to Silva (2018), 'rho=0.5' is indicated when expected time to signal is the design criterion, hence the default in `AnalyzeSetUp.Binomial`, or (iii) the alpha spending associated to the Wald-type rejection boundary, which is flat with respect to the likelihood ratio. See the `AnalyzeSetUp.Binomial` for more details.

In most cases, this pre-specified alpha spending function is used throughout the analysis, but if needed, it is possible to override it at any or each of the sequential tests. This is done using the AlphaSpend parameter, which specifies the maximum amount of alpha to spend up to and including the current test. In this way, it is possible to use any alpha spending function, and not only those available in `AnalyzeSetUp.Binomial`. It is also possible to use a flexible adaptive alpha spending plan that is not set in stone before the sequential analysis starts. The only requirement is that for a particular test with a new group of data, AlphaSpend must be decided before knowing the number of cases and controls in that group. To ensure a statistically valid sequential analysis, AlphaSpend can only depend on the number of events (cases + controls) at prior tests and the total number of events in the current test. This is important.

The function `Analyze.Binomial` is meant to perform the binomial sequential analysis with a certain level of autonomy. After running a test, the code offers a synthesis about the general parameter settings, the main conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests. A table with the main analyses results is automatically printed in the R console. Each column of the table contains a historical characteristic, including the information for the current test. Each line of the table corresponds to a specific test organized by calendar time. The table is titled with the title input defined through the function `AnalyzeSetUp.Binomial`, and its columns are organized and labeled in the following way: "Test", "Cases", "Controls", "Cumulative Cases", "Cumulative Controls", "Cumulative E[Cases]", "RR",

"LLR", "target", "actual", "CV", "Reject H0". Here follows a short description of each column:

- "Test" shows the order of the analysis, i.e., the arrival order of each chunk of data.

- "Cases" and "Controls" present the total of cases and controls that entered at each test, respectively.

- "Cumulative Cases" and "Cumulative Controls" in the i-th line have the cumulative counts of cases and controls up to the i-th test, respectively.

- "Cumulative E[Cases]" in line i is the expected cumulative number of cases for the i-th test under the null hypothesis.

- "RR" is the estimated relative risk for test i.

- "LLR" is the observed log-likelihood ratio test statistic.

- "target" is the target alpha spending for the i-th test.

- "actual" is the actual alpha spent up to the i-th test.

- "CV" is the critical value in the scale of the number of cases, showing how many casesa re needed to reject the null hypothesis at this test.

- "Reject H0" is a logical variable that is "Yes" when the null hypothesis is rejected, and the label "No" when H0 is not to be rejected

Observe that, because the binomial distribution is discrete, the target alpha spending will rarely be reached. The actual alpha spending is then shown to facilitate a realistic interpretation of the results.

The function `Analyze.Binomial` was designed to instruct the user with minimal information about bugs from the code, or about non-applicable parameter input usages. Some entries are not applicable for the parameter inputs. For example, the input "z" must be a positive number, and then if the user sets "z= -1", the code will report an error with the message "the entries of the vector "z" must be positive numbers". Thus, messages will appear when mistakes and inconsistencies are detected, and instructions about how to proceed to solve such problems will automatically appear.

Value

`result`	A table containing the main characteristics, conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests.

Acknowledgements

Development of the `Analyze.Binomial` function was funded by: - Food and Drug Administration, Center for Drug Evaluation and Research, through Mini-Sentinel Project: base version, documentation, unequal matching ratios;
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999: user-defined alpha spending functions, power-type alpha spending function, increased computational speed, confidence intervals for relative risks, end of schedule analysis using left-over alpha, enhanced error handling and messages, improved documentation.

We thank Claudia Coronel-Moreno for valuable editorial support, Bruce Fireman for general guidance, and Josh Gagne for important feedback on the unequal matching ratio feature.

See also

AnalyzeSetUp.Binomial: for setting up sequential analysis with the Analyze.Binomial function, before the first look at the data.
SampleSize.Binomial: for calculating the needed sample size to achieve the desired statistical power for continuous sequential analysis with binomial data.

Author(s)

Ivair Ramos Silva, Ned Lewis, Martin Kulldorff.

References

Fireman B, et al. (2013). Exact sequential analysis for binomial data with time varying probabilities. Manuscript in preparation.

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials. London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30, 58–78.

Kulldorff M, Silva IR. (2015). Continuous post-market sequential safety surveillance with minimum events to signal. arxiv:1503.01978 [stat.ap].

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71(3), 851–858.

Silva IR, Kulldorff M, Yih W. Katherine. (2020), Optimal alpha spending for sequential analysis with binomial data. Journal of the Royal Statistical Society Series B, 82(4) p. 1141–1164.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107-118.

Silva IR, Zhuang, Y. (2022), Bounded-width confidence interval following optimal sequential analysis of adverse events with binary data, Statistical Methods in Medical Research, 31(12), 2323–2337.

Examples

```
### Example. Four chunks of data.

### Firstly, it is necessary to set up the input parameters.
##  Here we use the Wald type alpha spending.
##  Note: cut off the "#" symbol before running the two lines below.
#     AnalyzeSetUp.Binomial(name="VaccineA",N=200,alpha=0.05,zp=1,M=3,
#     AlphaSpendType="Wald", title="Monitoring_vaccineA",
#     address="C:/Users/Ivair/Documents")

### Now we apply sequential tests to each of four chunks of data.
# ------------------------------------------------------------------------

## Test 1 - Situation where each individual event came from a different
## matching ratio.
## This first test uses the default Wald type alpha spending (AlphaSpend="n").
## Note: cut off the "#" symbol before running the line below.
```

```
#  Analyze.Binomial(name= "VaccineA",test=1,z=c(1.1,1.3,1.2,1),
#  cases= c(1,0,0,0), controls= c(0,1,1,1) )

## Test 2 - Situation where some of the events came from the same matching
## ratio.
## Observe that here we use an arbitrary alpha spending of 0.02.
## Note: cut off the "#" symbol before running the line below.
#  Analyze.Binomial(name= "VaccineA",test=2,z=c(1,1.5),cases= c(12,1),
#  controls= c(0,10), AlphaSpend=0.02)

## Test 3 - Situation of elevated number of events, but now the
## arbitrary alpha spending is of 0.04, and p is entered instead of z.
## Note: cut off the "#" symbol before running the line below.
#  Analyze.Binomial(name= "VaccineA",test=3,p=c(0.4,0.5),cases= c(12,10),
#  controls= c(10,14), AlphaSpend=0.04)

## Test 4 - Situation where all the events came from the same matching
## ratio.
## Here the original target alpha spending is used.
## Note: cut off the "#" symbol before running the line below.
#  Analyze.Binomial(name= "VaccineA",test=4,z=2,cases= 20,controls= 10)
```

`Analyze.CondPoisson` *Function to conduct group sequential analyses for conditional Poisson data without the need to know group sizes a priori.*

Description

The function `Analyze.CondPoisson` is used for either continuous, group, or mixed continuous-group sequential analysis for Poisson data conditioned on observed historical data. Unlike `CV.CondPoisson`, it is not necessary that data arrives in a near-continuous fashion. It is possible to use either a Wald type rejection boundary, which is flat with respect to the likelihood ratio, or a user defined alpha spending function. `Analyze.CondPoisson` is run at each look at the data. Before running it by the first time, it is necessary to run the `AnalyzeSetUp.CondPoisson` function.

Usage

```
Analyze.CondPoisson(name,test,events,PersonTimeRatio,AlphaSpend="n")
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and it must be the same as the name given by the `AnalyzeSetup.CondPoisson` function. Should never be the same as another sequential analysis that is run simultaneously on the same computer.
`test`	An integer indicating the number of hypothesis tests performed up to and including the current test. For example, if there were four prior looks at the data,

and this is the fifth one, then "test=5". This number should be increased by one each time that the Analyze.CondPoisson function is run for a new group of data, when it is part of the same sequential analysis. If not, there is an error message.

events The test-specific number of events, instead of the cumulative number, observed during the surveillance period.

PersonTimeRatio
The observed ratio between the punctual, instead of cumulative from previous tests, person-time observed in the current test, by the total person-time observed in the historical period.

AlphaSpend The alpha spending function is specified in the AnalyzeSetUp.CondPoisson function. At any look at the data, it is possible to over ride that pre-specified alpha spending plan by using the AlphaSpend parameter. AlphaSpend is a number representing the maximum amount of alpha (Type I error probabiliy) to be spent up to and including the current test. Because of the discrete nature of the Poisson distribution, the actual amount of alpha spent may be less than the maximum amount specified. It must be in the range (0,alpha]. The default value is no override, which means that, if AlphaSpend= "n", then the function will use the alpha spending plan specified in the AnalyzeSetUp.CondPoisson function.

Details

The function Analyze.CondPoisson performs continuous or group sequential analysis for Poisson data conditioned on observed historical data, (Li and Kulldorff, 2010). It can also be used for mixed continuous-group sequential analysis where some data arrives continuously while other data arrives in groups. Unlike CV.CondPoisson, there is a variety of alpha spending functions are available.

In sequential analysis, data is formed by cumulative information, collected in separated chunks or groups, which are observed at different moments in time. Analyze.CondPoisson is run each time a new group of data arrives at which time a new sequential test is conducted. When running Analyze.CondPoisson, only the data from the new group should be included when calling the function. The prior data has been stored, and it will be automatically retrieved by Analyze.CondPoisson, with no need to reenter that data. Before running Analyze.CondPoisson for the first time, it is necessary to set up the sequential analysis using the AnalyzeSetUp.CondPoisson function, which is run once, and just once, to define the sequential analysis parameters. For information about this, see the description of the AnalyzeSetUp.CondPoisson function.

The function Analyze.CondPoisson calculates critical values to determine if the null hypothesis should be rejected or not at each analysis. Critical values are given in the scale of the number of events. This is done for a pre-specified overall statistical significance level (alpha), and for an upper limit on the sample size, which is given by "T" or "K". Go to the documentation of AnalyzeSetUp.CondPoisson for more details about the choice between "T" or "K".

The exact analytical solution is obtained through numerical calculations. Based on the data and the critical value, the function determines if the null hypothesis should be rejected or not, and if subsequent tests should be conducted. After each test, the function also provides information about the amount of alpha that has been spent, the cumulative number of events, and the observed log-likelihood ratio statistic.

Before running Analyze.CondPoisson, it is necessary to specify a planned default alpha spending function, which is done using the AlphaSpendType parameter in the AnalyzeSetUp.CondPoisson

function. The default alpha spending plan can be either, (i) the polynomial power-type alpha spending plan, which is parameterized with rho, or (ii) the alpha spending associated to the Wald-type rejection boundary, which is flat with respect to the likelihood ratio. See the AnalyzeSetUp.CondPoisson for more details.

In most cases, this pre-specified alpha spending function is used throughout the analysis, but if needed, it is possible to override it at any or each of the sequential tests. This is done using the AlphaSpend parameter, which specifies the maximum amount of alpha to spend up to and including the current test. In this way, it is possible to use any alpha spending function, and not only those available in AnalyzeSetUp.CondPoisson. It is also possible to use a flexible adaptive alpha spending plan that is not set in stone before the sequential analysis starts. The only requirement is that for a particular test with a new group of data, AlphaSpend must be decided before knowing the PersonTimeRatio in that group. Hence, in order to ensure a statistically valid sequential analysis, AlphaSpend can only depend on the cumulative events. This is important.

The function Analyze.CondPoisson is meant to perform the conditional Poisson sequential analysis with a certain level of autonomy. After running a test, the code offers a synthesis about the general parameter settings, the main conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests. A table with the main analyses results is automatically printed in the R console. Each column of the table contains a historical characteristic, including the information for the current test. Each line of the table corresponds to a specific test organized by calendar time. The table is titled with the title input defined through the function AnalyzeSetUp.CondPoisson, and its columns are organized and labeled in the following way: "Test","Person-timeR","events","Cumulative Person-timeR","Cumulative events","LLR", "target","actual","CV","Reject H0". Here follows a short description of each column:

- "Test" shows the order of the analysis, i.e., the arrival order of each chunk of data.

- "Person-timeR" shows the observed ratio between the punctual person-time observed in the current test by the total person-time observed in the historical period.

- "Events" presents the observed number of events from the Poisson counting entered at each test.

- "Cumulative Person-timeR" shows the observed person-time ratio up to the current test.

- "Cumulative events" presents the observed number of events from the Poisson counting up to the current test.

- "LLR" is the observed log-likelihood ratio test statistic.

- "target" is the target alpha spending for the i-th test.

- "actual" is the actual alpha spent up to the i-th test.

- "CV" is the critical value in the scale of the log-likelihood ratio test statistic.

- "Reject H0" is a logical variable that receives the label "Yes" when the null hypothesis is rejected, and the label "No" when H0 is not to be rejected

Observe that, depending on the choices of the input parameters M and alpha through the AnalyzeSetUp.CondPoisson function, the actual alpha spending can differ from the target one. The actual alpha spending is then shown in order to favor a realistic interpretation of the results.

The function Analyze.CondPoisson was designed to instruct the user with minimal information about bugs from the code, or about non-applicable input parameters usage. Some entries are not applicable. For example, the input "Person-timeR" must be a positive number, and then if the user sets "Person-timeR= -1", then the code will report an error with the message "the entry of "Person-timeR" must be a number greater than zero". Thus, messages will appear when mistakes

and inconsistencies are detected. Instructions about how to proceed to solve such problems will automatically appear too.

Value

`result` A table containing the main characteristics, conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests.

Acknowledgements

Development of the `Analyze.CondPoisson` function was funded by: - National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999. - Foundation for Research Support of Minas Gerais State (FAPEMIG), MG, Brazil, through the grant Demanda Universal.

See also

`AnalyzeSetUp.CondPoisson`: for setting up sequential analysis with the `Analyze.CondPoisson` function, before the first look at the data.
`AnalyzeSetUp.Poisson`: for setting up sequential analysis with the `Analyze.Poisson` function, before the first look at the data.
`SampleSize.Poisson`: for calculating the needed sample size to achieve the desired statistical power for continuous sequential analysis with Poisson data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Fireman B, et al. (2013). Exact sequential analysis for Poisson data with time varying probabilities. Manuscript in preparation.

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials. London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30, 58–78.

Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Li L, Kulldorff M. (2010). A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71(3), 851–858.

Silva IR, Li L, Kulldorff M. (2019). Exact conditional maximized sequential probability ratio test adjusted for covariates. Sequential Analysis, 38(1), 115–133.

Silva IR., Lopes LM., Dias P., Yih WK. (2019). Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, DOI: 10.1002/sim.8097, 1–13.

Examples

```
### Example. Three chunks of data with a total person time
#   in the historical period equal to 10,000.

### Firstly, it is necessary to set up the input parameters.
##  Here we use the Wald type alpha spending.
##  Note: cut off the "#" symbol before running the two lines below, and,
##  Important: choose an actual "address" to save your set up information.
#   AnalyzeSetUp.CondPoisson(name="TestA", SampleSizeType="Events", K=100,
#   cc=20,alpha=0.05, M=1,AlphaSpendType="power-type",rho=0.5,title="n",
#   address="C:/Users/Ivair/Documents")

### Now we apply a test for each one of three chunks of data.
# ------------------------------------------------------------------------

## Test 1 - Situation where the fixed number of events is equal to 5.
## The observed information is "PersonTimeRatio=5000/10000=0.5".
## Note: cut off the "#" symbol before running the line below.

#  Analyze.CondPoisson(name="TestA",test=1,events=5,PersonTimeRatio=0.5)


## Test 2 - Situation where the fixed number of new events is equal to 6.
## The observed information is "PersonTimeRatio=3000/10000=0.3".

#Analyze.CondPoisson(name="TestA",test=2,events=6,PersonTimeRatio=0.3)


## Test 3 - Situation where the fixed number of events is equal to 10.
## The observed information is "PersonTimeRatio=1000/10000=0.1".

#Analyze.CondPoisson(name="TestA",test=3,events=10,PersonTimeRatio=0.1)
```

`Analyze.Multinomial` *Function for group sequential analyses for multinomial data, without the need to know group sizes a priori.*

Description

The function `Analyze.Multinomial` is used for either continuous or group sequential analysis, or for a combination of the two. `Analyze.Multinomial` is run at each look at the data. Before running it by the first time, it is necessary to run the `AnalyzeSetUp.Multinomial` function.

Usage

```
Analyze.Multinomial(name,test,cases,controls,N_exposures,N_controls,
exposure_group,strata_group_cases="n",strata_group_controls="n",
AlphaSpend="n")
```

Arguments

name	The name of the sequential analysis. Must be identical for all looks at the data, and it must be the same as the name given by the AnalyzeSetup.Multinomial function. Should never be the same as another sequential analysis that is run simultaneously on the same computer.
test	An integer indicating the number of hypothesis tests performed up to and including the current test. For example, if there were four prior looks at the data, and this is the fifth one, then "test=5". This number should be increased by one each time that the Analyze.Multinomial function is run for a new group of data, when it is part of the same sequential analysis. If not, there is an error message.
cases	A vector with the number of events per combination of exposure_group and strata_group_cases. There is no default.
controls	A vector with the number of events per per strata_group_controls. There is no default.
N_exposures	A vector with the number of individuals in the same risk window per combination of exposure_group and strata_group_cases excluding the entry for the control group. Must have the same dimension as cases.
N_controls	The number of individuals in the control (unexposed) group per strata_group_controls. There is no default.
exposure_group	Labels to indicate the exposure group of each entry in "cases". Each entry in "exposure_group" must belong to the set of labels in "ExposuresNames", and "ExposuresNames" is one of the inputs of the AnalyzeSet.Multinomial function. There is no default.
strata_group_cases	Labels to indicate the strata group, e.g., by combining the categorical covariates for age and gender, of each entry in "cases". The default is strata_group_cases="n" to indicate that there are no covariates to adjust for.
strata_group_controls	Labels to indicate the strata group, e.g., by combining the categorical covariates for age and gender, of each entry in "controls". The default is strata_group_controls="n" to indicate that there are no covariates to adjust for.
AlphaSpend	The alpha spending function is specified in the AnalyzeSetUp.Multinomial function. At any look at the data, it is possible to over ride that pre-specified alpha spending plan by using the AlphaSpend parameter. AlphaSpend is a number representing the maximum amount of alpha (Type I error probabiliy) to be spent up to and including the current test. Because of the discrete nature of the multinomial distribution, the actual amount of alpha spent may be less than the maximum amount specified. It must be in the range (0, alpha]. The default value is no override, which means that, if AlphaSpend= "n", then the function will use the alpha spending plan specified in the AnalyzeSetUp.Multinomial function.

Details

The function Analyze.Multinomial performs continuous or group sequential analysis for multinomial data. It can also be used for mixed continuous-group sequential analysis where some data

arrives continuously while other data arrives in groups. There is (i) no need to pre-specify the group sizes before the sequential analysis starts, and (ii) a variety of alpha spending functions are available.

This function runs the adaptive sequential multiple hypotheses testing for seasonal concomitant vaccines safety surveillance introduced by Silva and Maro(2025), which was developed for rapidly detecting increased risks of adverse events from one or more combinations of simultaneous vaccine exposures. As explained by Silva and Maro(2025), multiple seasonal vaccines may be recommended for some strata of the population defined by covariates, such as gender and age, for example. Such vaccines can be administered simultaneously or at least within the same vaccination season. Thus, monitoring adverse events origination from multiple vaccines may be more informative and statistically powerful than monitoring each vaccine separately.

The method is an extension of the binomial MaxSPRT method to a multinomial exposure model. With an exact analytical alpha spending approach, the computationally feasible limit of multiple exposures is likely limited to no more than two vaccines. For more complex situations with several vaccines, which can multiple types of endpoints (i.e. designated adverse events), a valid Monte Carlo decision rule is constructed.

The alpha spending plan is designed in a way to ensure that a target statistical power, gamma, is ensured for detecting a target increased risk, r1, for each multinomial entry. This is possible by means of a new concept introduced by Silva and Maro(2025), the robust alpha spending plan. For more details about the method and the construction of the critical values, see Silva and Maro(2025).

Before running `Analyze.Multinomial`, it is necessary to specify a planned default alpha spending function, which is done using the AlphaSpendType parameter in the `AnalyzeSetUp.Multinomial` function. As detailed by Silva and Maro(2025), there are two robust alpha spending plan choices, AlphaSpendType=1 or AlphaSpendType=2. See `AnalyzeSetUp.Multinomial` for more details.

The function `Analyze.Multinomial` is meant to perform the multinomial sequential analysis with a certain level of autonomy. After running a test, the code offers a synthesis about the general parameter settings, the main conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests. Tables with the main analyses results are automatically printed in the R console. The columns of the tables show historical characteristics, including the information for the current test. Each line of the tables corresponds to a specific test organized by calendar time. Observe that, because the multinomial distribution is discrete, the target alpha spending will rarely be reached. The actual alpha spending is then shown to facilitate a realistic interpretation of the results.

To adjust for covariates confounders, `Analyze.Multinomial` utilizes the maxLik package (Henningsen et al., 2011) to compute the MLE of the baseline covariates effects as derived by Silva and Maro(2025).

Value

`result` Table containing the main characteristics, conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests.

Acknowledgements

Development of the `Analyze.Multinomial` function was funded by: - Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute. - Conselho Nacional

de Desenvolvimento Cientifico e Tecnologico (CNPq), Brazil, and Fundacao de Amparo a Pesquisa do Estado de Minas Gerais (FAPEMIG), Brazil.

See also

AnalyzeSetUp.Binomial: for setting up sequential analysis with the Analyze.Binomial function, before the first look at the data.
SampleSize.Binomial: for calculating the needed sample size to achieve the desired statistical power for continuous sequential analysis with binomial data.

Author(s)

Ivair Ramos Silva, Judith Maro.

References

Silva IR, Maro J. (2025), Adaptive Sequential Multiple Hypotheses Testing for Seasonal Concomitant Vaccines Safety Surveillance. Working paper, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Henningsen A, Toomet O (2011), maxLik: A package for maximum likelihood estimation in R. Computational Statistics, 26(3), 443–458.

Examples

```
### Example. Two chunks of data.

### Firstly, it is necessary to set up the input parameters.
##  Note: cut off the "#" symbol before running the four lines below.

#   AnalyzeSetUp.Multinomial(name="VaccinesABC",N=200,alpha=0.05,R0=1,rho=1,
#   m=10000, title="Title of the output table",
#   ExposuresNames=c("A","B","C","AB","AC","BC","ABC"),
#   address="C:/Users")


### Running sequential tests for two chunks of data.
# ------------------------------------------------------------------------

## Test 1: in this first test, events were observed for each of
#          the exposures "A","B","C","AB","AC","BC","ABC", and
#          two binary covariates are present to adjust for gender
#          and age with categories "F-Y", "F-O", "M-Y", "M-O".
# Note: cut off the "#" symbol before running the lines below.


#res1<- Analyze.Multinomial(name="VaccinesABC", test=1,
#cases= c(2,1,0,1,0,1,0,
#         1,2,0,1,0,1,1,
#         3,3,1,1,0,0,1,
#         2,0,1,1,1,0,0),
```

```
#controls= c(3,3,3,3),

#N_exposures= c(1000,1500,300,300,320,280,240,
#               1000,1500,300,300,320,280,240,
#               1000,1500,300,300,320,280,240,
#               1000,1500,300,300,320,280,240),

#N_controls= c(500,500,500,500),

#exposure_group= c("A","B","C","AB","AC","BC","ABC",
#                  "A","B","C","AB","AC","BC","ABC",
#                  "A","B","C","AB","AC","BC","ABC",
#                  "A","B","C","AB","AC","BC","ABC"),

#strata_group_cases= c("F-Y","F-Y","F-Y","F-Y","F-Y","F-Y","F-Y",
#                      "F-O","F-O","F-O","F-O","F-O","F-O","F-O",
#                      "M-Y","M-Y","M-Y","M-Y","M-Y","M-Y","M-Y",
#                      "M-O","M-O","M-O","M-O","M-O","M-O","M-O"),

#strata_group_controls= c("F-Y","F-O","M-Y","M-O"),

#AlphaSpend= "n"            )



## Test 2: in this second test, events were observed for each of
#          the exposures "A","B","C","AB","AC","BC","ABC", but
#          there are no covariates to adjust for.
# Note: cut off the "#" symbol before running the lines below.

#res2<- Analyze.Multinomial(name="VaccinesABC", test=2,
#cases= c(7,7,3,5,2,3,0),
#controls= 2,
#N_exposures= c(1200,1600,400,250,300,300,280),
#N_controls= 600,
#exposure_group= c("A","B","C","AB","AC","BC","ABC"),
#strata_group_cases= "n",
#strata_group_controls= "n",
#AlphaSpend= "n"
#                          )
```

`Analyze.Poisson` *Function to conduct group sequential analyses for Poisson data without the need to know group sizes a priori.*

Description

The function Analyze.Poisson is used for either continuous or group sequential analysis, or for a combination of the two. Unlike CV.Poisson, it is not necessary to pre-specify the group sizes before the sequential analysis starts. Moreover, under the null hypothesis, the expected number of events, mu0, can be different for different observations. It is possible to use either a Wald type rejection boundary, which is flat with respect to the likelihood ratio, or a user defined alpha spending function. Analyze.Poisson is run at each look at the data. Before running it by the first time, it is necessary to run the AnalyzeSetUp.Poisson function.

Usage

```
Analyze.Poisson(name,test,mu0="n",cum.mu0="n",events,AlphaSpend="n")
```

Arguments

name	The name of the sequential analysis. Must be identical for all looks at the data, and it must be the same as the name given by the AnalyzeSetup.Poisson function. Should never be the same as another sequential analysis that is run simultaneously on the same computer.
test	An integer indicating the number of hypothesis tests performed up to and including the current test. For example, if there were four prior looks at the data, and this is the fifth one, then "test=5". This number should be increased by one each time that the Analyze.Poisson function is run for a new group of data, when it is part of the same sequential analysis. If not, there is an error message.
mu0	The test specific expected number of events under the null hypothesis. The parameter mu0 must be a positive number. There is no default value unless cum.mu0 is specified instead.
cum.mu0	The cumulative expected number of events under the null hypothesis. The parameter cum.mu0 must be a positive number. There is no default value unless mu0 specified instead.
events	The number of observed events.
AlphaSpend	The alpha spending function is specified in the AnalyzeSetUp.Poisson function. At any look at the data, it is possible to over ride that pre-specified alpha spending plan by using the AlphaSpend parameter. AlphaSpend is a number representing the maximum amount of alpha (Type I error probabiliy) to be spent up to and including the current test. Because of the discrete nature of the Poisson distribution, the actual amount of alpha spent may be less than the maximum amount specified. It must be in the range (0,alpha]. The default value is no override, which means that, if AlphaSpend= "n", then the function will use the alpha spending plan specified in the AnalyzeSetUp.Poisson function.

Details

The function Analyze.Poisson performs continuous or group sequential analysis for Poisson data. It can also be used for mixed continuous-group sequential analysis where some data arrives continuously while other data arrives in groups. Unlike CV.Poisson, there is (i) no need to pre-specify

the group sizes before the sequential analysis starts, (ii) a variety of alpha spending functions are available, and (iii) it is possible to include an offset term where, under the null hypothesis, different observations have different Poisson rates mu0.

In sequential analysis, data is formed by cumulative information, collected in separated chunks or groups, which are observed at different moments in time. `Analyze.Poisson` is run each time a new group of data arrives at which time a new sequential test is conducted. When running `Analyze.Poisson`, only the data from the new group should be included when calling the function. The prior data has been stored, and it will be automatically retrieved by `Analyze.Poisson`, with no need to reenter that data. Before running `Analyze.Poisson` for the first time, it is necessary to set up the sequential analysis using the `AnalyzeSetUp.Poisson` function, which is run once, and just once, to define the sequential analysis parameters. For information about this, see the description of the `AnalyzeSetUp.Poisson` function.

The function `Analyze.Poisson` calculates critical values to determine if the null hypothesis should be rejected or not at each analysis. Critical values are given in the scale of the number of events. This is done for a pre-specified overall statistical significance level (alpha), and for an upper limit on the sample size (N). The exact analytical solution is obtained through numerical calculations. Based on the data and the critical value, the function determines if the null hypothesis should be rejected or not, and if subsequent tests should be conducted. After each test, the function also provides information about the amount of alpha that has been spent, the cumulative number of events, and the estimated relative risk.

Before running `Analyze.Poisson`, it is necessary to specify a planned default alpha spending function, which is done using the AlphaSpendType parameter in the `AnalyzeSetUp.Poisson` function. The default alpha spending plan can be either, (i) the polynomial power-type alpha spending plan, which is parameterized with rho, and the default is rho=0.5 as suggested by Silva (2018), or (ii) the alpha spending associated to the Wald-type rejection boundary, which is flat with respect to the likelihood ratio. See the `AnalyzeSetUp.Poisson` for more details.

In most cases, this pre-specified alpha spending function is used throughout the analysis, but if needed, it is possible to override it at any or each of the sequential tests. This is done using the AlphaSpend parameter, which specifies the maximum amount of alpha to spend up to and including the current test. In this way, it is possible to use any alpha spending function, and not only those available in `AnalyzeSetUp.Poisson`. It is also possible to use a flexible adaptive alpha spending plan that is not set in stone before the sequential analysis starts. The only requirement is that for a particular test with a new group of data, AlphaSpend must be decided before knowing the number of events in that group. To ensure a statistically valid sequential analysis, AlphaSpend can only depend on cumulative mu0 values at prior tests and of the mu0 value in the current test. This is important.

The function `Analyze.Poisson` is meant to perform the Poisson sequential analysis with a certain level of autonomy. After running a test, the code offers a synthesis about the general parameter settings, the main conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests. A table with the main analyses results is automatically printed in the R console. Each column of the table contains a historical characteristic, including the information for the current test. Each line of the table corresponds to a specific test organized by calendar time. The table is titled with the title input defined through the function `AnalyzeSetUp.Poisson`, and its columns are organized and labeled in the following way: "Test", "mu0", "Events", "Cumulative mu0", "Cumulative Events", "RR", "LLR", "target", "actual", "CV", "Reject H0". Here follows a short description of each column:

- "Test" shows the order of the analysis, i.e., the arrival order of each chunk of data.

- "mu0" is the expected number of events under the null hypothesis for the chunck of data to be analyzed at each test.

- "Events" presents the observed number of events from the Poisson counting entered at each test.

- "Cumulative mu0" expected number of events under the null hypothesis up to the i-th test.

- "Cumulative Events" observed number of events up to the i-th test.

- "RR" is the estimated relative risk for test i.

- "LLR" is the observed log-likelihood ratio test statistic.

- "target" is the target alpha spending for the i-th test.

- "actual" is the actual alpha spent up to the i-th test.

- "CV" is the critical value in the scale of the number of events, showing how many events are needed to reject the null hypothesis at this test.

- "Reject H0" is a logical variable that is "Yes" when the null hypothesis is rejected, and the label "No" when H0 is not to be rejected

Observe that, because the Poisson distribution is discrete, the target alpha spending will rarely be reached. The actual alpha spending is then shown to facilitate a realistic interpretation of the results.

The function `Analyze.Poisson` was designed to instruct the user with minimal information about bugs from the code, or about non-applicable parameter input usages. Some entries are not applicable for the parameter inputs. For example, the input "mu0" must be a positive number, and then if the user sets "mu0= -1", the code will report an error with the message "the entry of "mu0" must be a number greater than zero". Thus, messages will appear when mistakes and inconsistencies are detected, and instructions about how to proceed to solve such problems will automatically appear.

Value

`result`	A table containing the main characteristics, conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests.

Acknowledgements

Development of the `Analyze.Poisson` function was funded by: - National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999.

See also

`AnalyzeSetUp.Poisson`: for setting up sequential analysis with the `Analyze.Poisson` function, before the first look at the data.
`SampleSize.Poisson`: for calculating the needed sample size to achieve the desired statistical power for continuous sequential analysis with Poisson data.

Author(s)

Ivair Ramos Silva, Ned Lewis, Martin Kulldorff.

References

Fireman B, et al. (2013). Exact sequential analysis for Poisson data with time varying probabilities. Manuscript in preparation.

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials. London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30, 58–78.

Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance With Poisson Data. Methodol Comput Appl Probab, 20(2), 739-750.

Silva IR, Maro J, Kulldorff M. (2021). Exact sequential test for clinical trials and post-market drug and vaccine safety surveillance with Poisson and binary data. Statistics in Medicine, DOI: 10.1002/sim.9094.

Examples

```
### Example. Four chunks of data.

### Firstly, it is necessary to set up the input parameters.
##  Here we use the Wald type alpha spending.
##  Note: cut off the "#" symbol before running the two lines below, and,
##  very important, choose an actual "address" to save your set up information.
#   AnalyzeSetUp.Poisson(name="VaccineA", SampleSize=100, alpha=0.05,
#   M=1,AlphaSpendType="power-type",rho=0.5,title="n",
#   address="C:/Users/Ivair/Documents")

### Now we can sequentially apply a test for each one of three chunks of data.
# ------------------------------------------------------------------------

## Test 1 - Situation where the expected number of events under H0 is equal to 2.
## This first test uses the default Wald type alpha spending (AlphaSpend="n").
## Note: cut off the "#" symbol before running the line below.
#  Analyze.Poisson(name="VaccineA",test=1,mu0=2,events=1,AlphaSpend="n")

## Test 2 - Situation where the expected number of events under H0 is equal to 0.8.
## Observe that here we use an arbitrary alpha spending of 0.02.
## Note: cut off the "#" symbol before running the line below.
#  Analyze.Poisson(name="VaccineA",test=2,mu0=0.8,events=2, AlphaSpend=0.02)

## Test 3 - Situation of elevated number of events, but now the
## arbitrary alpha spending is of 0.04.
## Note: cut off the "#" symbol before running the line below.
#  Analyze.Poisson(name="VaccineA",test=3,mu0=9,events=10, AlphaSpend=0.04)
```

`Analyze.wBinomial` *Function for group sequential analyses of multiple weighted binomial endpoints, without the need to know group sizes a priori.*

Description

The function `Analyze.wBinomial` is used for either continuous or group sequential analysis, or for a combination of the two. Unlike `CV.Binomial` and `CV.G.Binomial`, it is not necessary to pre-specify the group sizes before the sequential analysis starts. More important, this function is designed specifically when multiple outcomes with weights are analyzed. This is done using user defined alpha spending functions. For single binomial outcomes, please use `Analyze.Binomial`. `Analyze.wBinomial` is run at each look at the data. Before running it by the first time, it is necessary to run the `AnalyzeSetUp.wBinomial` function.

Usage

```
Analyze.wBinomial(name,test,z,w,ExposureA,ExposureB,AlphaSpend="n")
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and it must be the same as the name given by the `AnalyzeSetup.wBinomial` function. Should never be the same as another sequential analysis that is run simultaneously on the same computer.
`test`	An integer indicating the number of hypothesis tests performed up to and including the current test. For example, if there were four prior looks at the data, and this is the fifth one, then "test=5". This number should be increased by one each time that the `Analyze.wBinomial` function is run for a new group of data, when it is part of the same sequential analysis. If not, there is an error message.
`z`	For a matched case-control analysis, z is the number of controls matched to each case. For example, if there are 3 controls matched to each case, "z=3". In a self-control analysis, z is the ratio of the length of the control interval to the length of the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, "z=7/2". In terms of p, the binomial probability under the null hypothesis, "p=1/(1+z)", or equivalently, "z=1/p-1". The parameter z must be a positive number. The default value is z=1 (p=0.5). If the ratio is the same for all observations, then z can be any positive number.
`w`	A vector containing the weights associated to each outcome.
`ExposureA`	A number or a vector of the same length as w containing the number of cases from an exposure A per outcome. The length of ExposureA equals to the number of outcomes.
`ExposureB`	A number or a vector of the same length as w containing the number of cases from an exposure B per outcome. The length of ExposureB equals to the number of outcomes.

AlphaSpend — The alpha spending function is specified in the AnalyzeSetUp.wBinomial function. At any look at the data, it is possible to over ride that pre-specified alpha spending plan by using the AlphaSpend parameter. AlphaSpend is a number representing the maximum amount of alpha (Type I error probabiliy) to be spent up to and including the current test. Because of the discrete nature of the binomial distribution, the actual amount of alpha spent may be less than the maximum amount specified. It must be in the range (0,alpha]. The default value is no override, which means that, if AlphaSpend= "n", then the function will use the alpha spending plan specified in the AnalyzeSetUp.wBinomial function.

Details

The function Analyze.wBinomial performs exact sequential testing for multiple weighted binomial endpoints, that is, the analysis reflects the drugs combined benefit and safety profile. It works with a variety of alpha spending functions for continuous, group or mixed group-continuous sequential analysis. The binomial probabilities, given by $p = 1/(1 + z)$.

The test statistic is based on the weighted sum of binomial endpoints introduced by Silva et al (2020).

Unlike CV.Binomial and CV.G.Binomial, there is (i) no need to pre-specify the group sizes before the sequential analysis starts, (ii) a variety of alpha spending functions are available, and (iii) it is designed for multiple weighted binomial endpoints.

In sequential analysis, data is formed by cumulative information, collected in separated chunks or groups, which are observed at different moments in time. Analyze.wBinomial is run each time a new group of data arrives at which time a new sequential test is conducted. When running Analyze.wBinomial, only the data from the new group should be included when calling the function. The prior data has been stored, and it will be automatically retrieved by Analyze.wBinomial, with no need to reenter that data. Before running Analyze.wBinomial for the first time, it is necessary to set up the sequential analysis using the AnalyzeSetUp.wBionimial function, which is run once, and just once, to define the sequential analysis parameters. For information about this, see the description of the AnalyzeSetUp.wBinomial function.

The function Analyze.wBinomial calculates critical values to determine if the null hypothesis should be rejected or not at each analysis. The null hypothesis is that the relative risk of each outcome is equal to 1.

Critical values are given in the scale of the ratio S_A/S_B, where $S_A = w_1 ExposeA_1 + w_2 ExposeA_2 + ... + w_k ExposeA_k$, and $S_B = w_1 ExposeB_1 + w_2 ExposeB_2 + ... + w_k ExposeB_k$, and k is the length of w.

Critical values for each test are elicited for a pre-specified overall statistical significance level (alpha), and for an upper limit on the sample size (N). The exact analytical solution is obtained through numerical calculations. Based on the data and the critical value, the function determines if the null hypothesis should be rejected or not, and if subsequent tests should be conducted. After each test, the function also provides information about the amount of alpha that has been spent, the cumulative number of cases and controls, and the maximum likelihood estimate of the relative risk.

For binomial and Bernoulli data, there are a number of 0/1 observations that can either be an ExposureA or an ExposureB. Under the null hypothesis, the probability of being an ExposureA is p, and the probability of being an ExposureB is 1-p. If data comes from a self-control analysis, the observation is an ExposureA if the event occurred in the risk interval, and it is an ExposureB if the

event occurred in the control interval. Under the null hypothesis, we then have that $p = 1/(1 + z)$, where z is the ratio of the length of the control interval to the length of the risk interval.

Before running `Analyze.wBinomial`, it is necessary to specify a planned default alpha spending function, which is done using the AlphaSpendType parameter in the `AnalyzeSetUp.wBinomial` function. The default alpha spending is of the polynomial power-type parameterized with rho, which, according to Silva (2018), 'rho=0.5' is indicated when expected time to signal is the design criterion, hence the default in `AnalyzeSetUp.wBinomial`. See the `AnalyzeSetUp.wBinomial` for more details.

In most cases, this pre-specified alpha spending function is used throughout the analysis, but if needed, it is possible to override it at any or each of the sequential tests. This is done using the AlphaSpend parameter, which specifies the maximum amount of alpha to spend up to and including the current test. In this way, it is possible to use any alpha spending function, and not only those available in `AnalyzeSetUp.wBinomial`. It is also possible to use a flexible adaptive alpha spending plan that is not set in stone before the sequential analysis starts. The only requirement is that for a particular test with a new group of data, AlphaSpend must be decided before knowing the number of ExposureA and ExposureB in that group. To ensure a statistically valid sequential analysis, AlphaSpend can only depend on the number of events (ExposureA + ExposureB) at prior tests and the total number of events in the current test. This is important.

The function `Analyze.WBinomial` is meant to perform the binomial sequential analysis with a certain level of autonomy. After running a test, the code offers a synthesis about the general parameter settings, the main conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests. A table with the main analyses results is automatically printed in the R console. Each column of the table contains a historical characteristic, including the information for the current test. Each line of the table corresponds to a specific test organized by calendar time. The table is titled with the title input defined through the function `AnalyzeSetUp.wBinomial`, and its columns are organized and labeled in the following way: "Test", "#Events", "Relative Risk", "Test Statistic", "Critical Value", "Alpha", "Reject H0". Here follows a short description of each column:

- "Test" shows the order of the analysis, i.e., the arrival order of each chunk of data.

- "#Events" present the total of events per outcome.

- "Relative Risk" is the estimated relative risk per outcome.

- "Test Statistic" the ratio of weighted sum of binomial endpoints between ExposureA and ExposureB.

- "Critical Value" is the signaling threshold for each test.

- "Alpha" shows the target and actual alpha spending up to the i-th test.

- "Reject H0" is a logical variable that is "Yes" when the null hypothesis is rejected, and the label "No" when H0 is not to be rejected

Observe that, because the binomial distribution is discrete, the target alpha spending will rarely be reached. The actual alpha spending is then shown to facilitate a realistic interpretation of the results.

The function `Analyze.wBinomial` was designed to instruct the user with minimal information about bugs from the code, or about non-applicable parameter input usages. Some entries are not applicable for the parameter inputs. For example, the input "z" must be a positive number, and then if the user sets "z= -1", the code will report an error with the message "z must be a positive number". Thus, messages will appear when mistakes and inconsistencies are detected, and instructions about how to proceed to solve such problems will automatically appear.

Value

`result` A table containing the main characteristics, conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests.

Acknowledgements

Development of the `Analyze.wBinomial` function was funded by: - National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999: user-defined alpha spending functions, power-type alpha spending function, increased computational speed, end of schedule analysis using left-over alpha, enhanced error handling and messages, improved documentation.

See also

`AnalyzeSetUp.wBinomial`: for setting up sequential analysis with the `Analyze.wBinomial` function, before the first look at the data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Silva IR, Gagne J, Najafzadeh M, Kulldorff M. (2020). Exact Sequential Analysis for Multiple Weighted Binomial Endpoints. Statistics in Medicine, 39(3), 340–351.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107–118.

Examples

```
### Example. Four chunks of data.

### Firstly, it is necessary to set up the input parameters.
##  Note: cut off the "#" symbol before running the two lines below.

#   AnalyzeSetUp.wBinomial(name="Rofe_Naisds",N=1000,alpha=0.05,M=1,
#   rho=0.5,title="rofecoxib (Vioxx) vs. NSAID comparison",
#   address="C:/Users/Ivair/Documents",
#   Tailed="two")

### Now we apply sequential tests to each of two chunks of data.
# ------------------------------------------------------------------------

## This example is based on two outcomes, myocardinal
#  infarction (w1=2.2), and major bleeding (w2=0.04), obtained
#  from a study comparing risk of myocardial infarction and
#  gastrointestinal bleeding. See details in Silva et al (2020).

## Test 1
```

```
## Note: cut off the "#" symbol before running the line below.
#  Analyze.wBinomial(name="Rofe_Naisds",test=1,z=1,w=c(2.2,0.04),
#  ExposureA=c(11,12),ExposureB=c(13,10),AlphaSpend="n" )

## Test 2
## Note: cut off the "#" symbol before running the line below.
#  Analyze.wBinomial(name="Rofe_Naisds",test=2,z=c(1,1),
#  w=c(2.2,0.04),ExposureA=c(19,12),ExposureB=c(16,11),AlphaSpend="n")
```

`AnalyzeRegression.Binomial`

Function for MaxSPRT regression analyses with binary/binomial data, without the need to know group sizes a priori.

Description

The function `AnalyzeRegression.Binomial` is used for either continuous or group sequential analysis, or for a combination of the two. Unlike `CV.Binomial`, it is not necessary to pre-specify the group sizes before the sequential analysis starts. Moreover, under the null hypothesis, the binomial probability, p, can be different for different observations. In a matched case-control setting, this means that the matching ratios can be different for different matched sets. `AnalyzeRegression.Binomial` is run at each look at the data. Before running it by the first time, it is necessary to run the `AnalyzeSetUpRegression.Binomial` function.

Usage

```
AnalyzeRegression.Binomial(name,test,z="n",p="n",cases,controls,covariates,AlphaSpend="n")
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and it must be the same as the name given by the `AnalyzeSetupRegression.Binomial` function. Should never be the same as another sequential analysis that is run simultaneously on the same computer.
`test`	An integer indicating the number of hypothesis tests performed up to and including the current test. For example, if there were four prior looks at the data, and this is the fifth one, then "test=5". This number should be increased by one each time that the `AnalyzeRegression.Binomial` function is run for a new group of data, when it is part of the same sequential analysis. If not, there is an error message.
`z`	For a matched case-control analysis, z is the number of controls matched to each case. For example, if there are 3 controls matched to each case, "z=3". In a self-control analysis, z is the ratio of the length of the control interval to the length of the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, "z=7/2". In terms of p, the binomial probability under the null hypothesis, "p=1/(1+z)", or equivalently, "z=1/p-1". The parameter z must

be a positive number. The default value is z=1 (p=0.5). If the ratio is the same for all observations, then z can be any positive number. If the ratio is different for different observations, then z is a vector of positive numbers.

`p` The probability of having a case under the null hypothesis. There is no default value.

`cases` A number or a vector of the same length as z containing the number of cases per stratum of individuals defined by the matrix covariates.

`controls` A number or a vector of the same length as z containing the number of controls per stratum of individuals defined by the matrix covariates.

`covariates` Matrix with the covariates for the regression model. The i-th line of covariates has the information related to the i-th entry of cases and controls. Each column of covariates corresponds to a different explanatory covariate.

`AlphaSpend` The alpha spending function is specified in the `AnalyzeSetUpRegression.Binomial` function. At any look at the data, it is possible to over ride that pre-specified alpha spending plan by using the AlphaSpend parameter. AlphaSpend is a number representing the maximum amount of alpha (Type I error probabiliy) to be spent up to and including the current test. Because of the discrete nature of the binomial distribution, the actual amount of alpha spent may be less than the maximum amount specified. It must be in the range (0,alpha]. The default value is no override, which means that, if AlphaSpend= "n", then the function will use the alpha spending plan specified in the `AnalyzeSetUpRegression.Binomial` function.

Details

The function `AnalyzeRegression.Binomial` performs continuous or group MaxSPRT regression analysis for Bernoulli or binomial data based on the method proposed by Silva et al.(2025).

It can also be used for mixed continuous-group sequential analysis where some data arrives continuously while other data arrives in groups.

Unlike `CV.Binomial`, there is (i) no need to pre-specify the group sizes before the sequential analysis starts, (ii) a variety of alpha spending functions are available, and (iii) it is possible to include an offset term z where, under the null hypothesis, different observations have different binomial probabilities p.

In sequential analysis, data is formed by cumulative information, collected in separated chunks or groups, which are observed at different moments in time. `AnalyzeRegression.Binomial` is run each time a new group of data arrives at which time a new sequential test is conducted. When running `AnalyzeRegression.Binomial`, only the data from the new group should be included when calling the function. The prior data has been stored, and it will be automatically retrieved by `AnalyzeRegression.Binomial`, with no need to reenter that data. Before running `AnalyzeRegression.Binomial` for the first time, it is necessary to set up the sequential analysis using the `AnalyzeSetUpRegression.Binomial` function, which is run once, and just once, to define the sequential analysis parameters. For information about this, see the description of the `AnalyzeSetUpRegression.Binomial` function.

The function `AnalyzeRegression.Binomial` calculates critical values to determine if the null hypothesis should be rejected or not at each analysis. Critical values are given by the value of the AlphaSpending function, and the test statistic is the Monte Carlo p-value calculated as proposed by

Silva et al.(2025). This way, the null hypothesis H0:RR<= R0, where R0 is the testing margin given by the analyst in the `AnalyzeSetUpRegression.Binomial`, and RR is the true unknow relative risk to test.

After each test, the function also provides information about the amount of alpha that has been spent, the cumulative number of cases and controls, regression coefficient estimates, confidence intervals, and the maximum likelihood estimate of the relative risk per stratum of individuals observed during the sequential analysis.

For binomial and Bernoulli data, there are a number of 0/1 observations that can either be a case or a control. Under the null hypothesis, the probability of being a case is p, and the probability of being a control is 1-p. If data comes from a self-control analysis, the observation is a case if the event occurred in the risk interval, and it is a control if the event occurred in the control interval. Under the null hypothesis, we then have that $p = 1/(1 + z)$, where z is the ratio of the length of the control interval to the length of the risk interval. This ratio, and hence p, does not need to be the same for all observations.

If data comes from a matched set of exposed and unexposed individuals, then the observation is a case if the event occurred among one of the exposed, and it is a control if it occurred among one of the unexposed. Under the null hypothesis, $p = 1/(1 + z)$, where z is the number of unexposed individuals divided by the number of exposed individuals in the matched set. Again, this ratio does not have to be the same for all matched sets. The variable z can be any positive number.

The ratio parameter z is a vector, representing multiple z values. For each value of z, it is necessary to specify the number of cases and the number of controls. This means that for a chunk of data, the vector of zs has to be of the same length as the vector of cases and the vector of controls. Therefore, the first entry of the vector z is the matching ratio associated to the first entries of cases and of controls. The second entry of z is the matching ratio with respect to the second entries of cases and of controls, and so on. For example, consider that each of five observations came from four different matching ratios. In this situation, the vectors cases, controls and z are all of length four. For example, suppose "z=c(2,1,0.5,3)", "cases=c(1,1,0,0)" and "controls=c(0,0,1,2)". The matching ratio for the first observation, which turned out as a case, is equal to 2. For the second observation, also a case, the matching is equal to 1. With a matching ration of 0.5, the third observation turned out to be a control. The two last observations both had a matching ratio of 3, and both of them were controls. To complete this example, the "covariates" input has four lines (one line per "cases" entry). For this example, suppose that two explanatory continuous variables are available, covariates=matrix(c(0.1,1.01, 0.2,0, 0.3,1, 0.15,0.9),4,2).

If all observations in the same data group has the same ratio, the vectors are of size one, that is, they are simple numbers. For example, if there were ten observations that all had a ratio of 2, with seven cases and three controls, we have "z=2", "cases=7", and "controls=3". In this case, the the "covariates" matrix has only one line. For example, covariates=matrix(c(0.1,1.01),1,2), which represents the situation with two explanatory variables (covariates has two columns).

Alternatively, instead of z the user can specify p directly. Note that only one of these inputs, z or p, has to be specified, but if both are entered the code will only work if z and p are such that p=1/(1+z). Otherwise, an error message will appear to remind that such condition must be complied.

Before running `AnalyzeRegression.Binomial`, it is necessary to specify a planned default alpha spending function, which is done using the AlphaSpendType parameter in the `AnalyzeSetUpRegression.Binomial` function. The default alpha spending plan is the polynomial power-type alpha spending plan parameterized with "rho=1". Different alpha spending plans can be obtained by selecting different values for rho (Silva, 2018).

In most cases, this pre-specified alpha spending function is used throughout the analysis, but if needed, it is possible to override it at any or each of the sequential tests. This is done using the AlphaSpend parameter in `AnalyzeRegression.Binomial`, which specifies the maximum amount of alpha to spend up to and including the current test. In this way, it is possible to use any alpha spending function, and not only the power-type available in `AnalyzeSetUpRegression.Binomial`. It means that the AlphaSpend parameter can be used to promote a flexible adaptive alpha spending plan that is not set in stone before the sequential analysis starts. The only requirement is that for a particular test with a new group of data, AlphaSpend must be decided before knowing the number of cases and controls in that group. To ensure a statistically valid sequential analysis, AlphaSpend can only depend on the number of events (cases + controls) at prior tests and the total number of events in the current test. This is important.

The function `AnalyzeRegression.Binomial` is meant to perform the binomial sequential analysis with a certain level of autonomy. After running a test, the code offers a synthesis about the general parameter settings, the main conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests.

Observe that, because the binomial distribution is discrete, the target alpha spending will rarely be reached. The actual alpha spending is then shown to facilitate a realistic interpretation of the results.

The function `AnalyzeRegression.Binomial` was designed to instruct the user with minimal information about bugs from the code, or about non-applicable parameter input usages. Some entries are not applicable for the parameter inputs. For example, the input "z" must be a positive number, and then if the user sets "z= c(-1,0,2)", the code will report an error with the message "the entries of the vector "z" must be positive numbers". Thus, messages will appear when mistakes and inconsistencies are detected, and instructions about how to proceed to solve such problems will automatically appear.

Value

`result`	Four data.frames (Decision_table, Relative_risk_estimates, Coefficients, Confidence_Intervals) with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests.

Acknowledgements

Development of the `AnalyzeRegression.Binomial` function was funded by: - Food and Drug Administration, Center for Drug Evaluation and Research, through the Mini-Sentinel Project (base version, documentation);
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (user defined alpha spending functions, improved documentation);
- National Council of Scientific and Technological Development (CNPq), Brazil, process number 302882/2022-7. (v4.3.1).
- Support Foundation to Minas Gerais State Research-Fapemig, Brazil, grant numbers PQ-00787-21 and RED-00133-21. (v3.1 to v4.3).

See also

`AnalyzeSetUpRegression.Binomial`: for setting up sequential analysis with the `AnalyzeRegression.Binomial`

function, before the first look at the data.

Author(s)

Ivair Ramos Silva.

References

Fireman B, et al. (2013). Exact sequential analysis for binomial data with time varying probabilities. Manuscript in preparation.

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials. London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30, 58–78.

Kulldorff M, Silva IR. (2015). Continuous post-market sequential safety surveillance with minimum events to signal. arxiv:1503.01978 [stat.ap].

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71(3), 851–858.

Silva IR, Kulldorff M, Yih W. Katherine. (2020), Optimal alpha spending for sequential analysis with binomial data. Journal of the Royal Statistical Society Series B, 82(4) p. 1141–1164.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107-118.

Silva IR, Zhuang, Y. (2022), Bounded-width confidence interval following optimal sequential analysis of adverse events with binary data, Statistical Methods in Medical Research, 31(12), 2323–2337.

Silva IR, Montalban, J., Oliveira, F. (2025), Maximized Sequential Probability Ratio Test Regression. Working paper - Sentinel (TIDE) project, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Examples

```
### Example. Three chunks of data.

### Firstly, it is necessary to set up the input parameters.
##  Note: cut off the "#" symbol before running the lines below.
#     AnalyzeSetUpRegression.Binomial(name="VaccineA",N=200,alpha=0.05,
#     R0=1, rho=1,mref=999,title="Monitoring_vaccineA",
#     address="C:/Users/Ivair/Documents")

### Now we apply sequential tests to each of three chunks of data.
# ------------------------------------------------------------------------

## Test 1 - Situation where each stratum came from a different
## matching ratio.
## This first test uses the default power-type (rho=1) alpha spending (AlphaSpend="n").
## Note: cut off the "#" symbol before running the lines below.
#  AnalyzeRegression.Binomial(name= "VaccineA",test=1,z=c(1.1,1.3,1.2,1),
#  cases= c(1,0,0,0), controls= c(0,1,1,1),
```

```
#  covariates=matrix(c(0.1,1.01, 0.2,0, 0.3,1, 0.15,0.9),4,2)  )

## Test 2 - Situation where some of the strata came from the same matching
## ratio.
## Observe that here we use an arbitrary alpha spending of 0.02.
## Note: cut off the "#" symbol before running the line below.
#  AnalyzeRegression.Binomial(name= "VaccineA",test=2,z=c(1,1,1.5),cases= c(12,1,4),
#  controls= c(0,10,5), covariates=matrix(c(0.3,0.9, 0.5,0.4, 0.55,1.1),3,2),
#  AlphaSpend=0.02)

## Test 3 - Situation of elevated number of events, but now the
## arbitrary alpha spending is of 0.04, and p is entered instead of z.
## Note: cut off the "#" symbol before running the line below.
#  AnalyzeRegression.Binomial(name= "VaccineA",test=3,p=c(0.6,0.4),cases= c(15,12),
#  controls= c(13,16), covariates=matrix(c(0.1,1.1, 0.6,0.35),2,2),
#  AlphaSpend=0.04)
```

`AnalyzeRegression.CondPoisson`

Function for MaxSPRT regression analyses with conditional Poisson data, without the need to know group sizes a priori.

Description

The function `AnalyzeRegression.CondPoisson` is used for either continuous or group sequential analysis, or for a combination of the two. Unlike `CV.CondPoisson`, it is not necessary to pre-specify the group sizes before the sequential analysis starts. `AnalyzeRegression.CondPoisson` is run at each look at the data. Before running it by the first time, it is necessary to run the `AnalyzeSetUpRegression.CondPoisson` function.

Usage

```
AnalyzeRegression.CondPoisson(name,test,events,PersonTimeRatio="n",
covariates,AlphaSpend="n")
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and it must be the same as the name given by the `AnalyzeSetupRegression.CondPoisson` function. Should never be the same as another sequential analysis that is run simultaneously on the same computer.
`test`	An integer indicating the number of hypothesis tests performed up to and including the current test. For example, if there were four prior looks at the data, and this is the fifth one, then "test=5". This number should be increased by one each time that the `AnalyzeRegression.CondPoisson` function is run for a new

group of data, when it is part of the same sequential analysis. If not, there is an error message.

`events` The test specific vector with the number of observed events per covariates combination.

`PersonTimeRatio`
Vector with the ratio between the punctual, instead of cumulative from previous tests, person-time observed in the current test, by the total person-time observed in the historical period, per covariates combination.

`covariates` Matrix with the covariates for the regression model. The i-th line of covariates has the information related to the i-th entry of the vector events. Each column of covariates corresponds to a different explanatory covariate.

`AlphaSpend` The alpha spending function is specified in the `AnalyzeSetUpRegression.CondPoisson` function. At any look at the data, it is possible to over ride that pre-specified alpha spending plan by using the AlphaSpend parameter. AlphaSpend is a number representing the maximum amount of alpha (Type I error probabiliy) to be spent up to and including the current test. Because of the discrete nature of the Poisson distribution, the actual amount of alpha spent may be less than the maximum amount specified. It must be in the range (0,alpha]. The default value is no override, which means that, with the default AlphaSpend= "n", the function will use the alpha spending plan specified in the `AnalyzeSetUpRegression.CondPoisson` function.

Details

The function `AnalyzeRegression.CondPoisson` performs continuous or group MaxSPRT regression analysis for conditional Poisson data based on the method proposed by Silva et al.(2025).

It can also be used for mixed continuous-group sequential analysis where some data arrives continuously while other data arrives in groups.

Unlike `CV.CondPoisson`, there is (i) no need to pre-specify the group sizes before the sequential analysis starts, (ii) a variety of alpha spending functions are available, and (iii) it is possible to include an offset term where, under the null hypothesis, different observations have different Poisson person-time ratios between the surveillance and the historical information.

In sequential analysis, data is formed by cumulative information, collected in separated chunks or groups, which are observed at different moments in time. `AnalyzeRegression.CondPoisson` is run each time a new group of data arrives at which time a new sequential test is conducted. When running `AnalyzeRegression.CondPoisson`, only the data from the new group should be included when calling the function. The prior data has been stored, and it will be automatically retrieved by `AnalyzeRegression.CondPoisson`, with no need to reenter that data. Before running `AnalyzeRegression.CondPoisson` for the first time, it is necessary to set up the sequential analysis using the `AnalyzeSetUpRegression.CondPoisson` function, which is run once, and just once, to define the sequential analysis parameters. For information about this, see the description of the `AnalyzeSetUpRegression.CondPoisson` function.

The function `AnalyzeRegression.CondPoisson` calculates critical values to determine if the null hypothesis should be rejected or not at each analysis. Critical values are given by the value of the AlphaSpending function, and the test statistic is the Monte Carlo p-value calculated as proposed by Silva et al.(2025). This way, the null hypothesis H0:RR<= R0, where R0 is the testing margin given

by the analyst in the AnalyzeSetUpRegression.CondPoisson, and RR is the true unknow relative risk to test.

After each test, the function also provides information about the amount of alpha that has been spent, the cumulative number of cases and controls, regression coefficient estimates, confidence intervals, and the maximum likelihood estimate of the relative risk per stratum of individuals observed during the sequential analysis.

Before running AnalyzeRegression.CondPoisson, it is necessary to specify a planned default alpha spending function, which is done using the AlphaSpendType parameter in the AnalyzeSetUpRegression.CondPoisson function. The default alpha spending plan is the polynomial power-type alpha spending plan parameterized with "rho=1". Different alpha spending plans can be obtained by selecting different values for rho (Silva et al, 2019).

In most cases, this pre-specified alpha spending function is used throughout the analysis, but if needed, it is possible to override it at any or each of the sequential tests. This is done using the AlphaSpend parameter in AnalyzeRegression.CondPoisson, which specifies the maximum amount of alpha to spend up to and including the current test. In this way, it is possible to use any alpha spending function, and not only the power-type available in AnalyzeSetUpRegression.CondPoisson. It means that the AlphaSpend parameter can be used to promote a flexible adaptive alpha spending plan that is not set in stone before the sequential analysis starts. The only requirement is that for a particular test with a new group of data, AlphaSpend must be decided before knowing the number of events in that group. To ensure a statistically valid sequential analysis, AlphaSpend can only depend on the values of PersonTimeRatio. This is important.

The function AnalyzeRegression.CondPoisson is meant to perform the Poisson sequential analysis with a certain level of autonomy. After running a test, the code offers a synthesis about the general parameter settings, the main conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests.

Observe that, because the Poisson distribution is discrete, the target alpha spending will rarely be reached. The actual alpha spending is then shown to facilitate a realistic interpretation of the results.

The function AnalyzeRegression.CondPoisson was designed to instruct the user with minimal information about bugs from the code, or about non-applicable parameter input usages. Some entries are not applicable for the parameter inputs.

Value

`result` Four data.frames (Decision_table, Relative_risk_estimates, Coefficients, Confidence_Intervals) with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests.

Acknowledgements

Development of the AnalyzeRegression.CondPoisson function was funded by: - Food and Drug Administration, Center for Drug Evaluation and Research, through the Mini-Sentinel Project (base version, documentation);
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (user defined alpha spending functions, improved documentation);
- National Council of Scientific and Technological Development (CNPq), Brazil, process number 302882/2022-7. (v4.3.1).

- Support Foundation to Minas Gerais State Research-Fapemig, Brazil, grant numbers PQ-00787-21 and RED-00133-21. (v3.1 to v4.3).

See also

AnalyzeSetUpRegression.CondPoisson: for setting up sequential analysis with the AnalyzeRegression.CondPoisson function, before the first look at the data.

Author(s)

Ivair Ramos Silva.

References

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials. London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30, 58–78.

Kulldorff M, Silva IR. (2015). Continuous post-market sequential safety surveillance with minimum events to signal. arxiv:1503.01978 [stat.ap].

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71(3), 851–858.

Silva IR, Kulldorff M, Yih W. Katherine. (2020), Optimal alpha spending for sequential analysis with binomial data. Journal of the Royal Statistical Society Series B, 82(4) p. 1141–1164.

Silva IR, Lopes LM, Dias P, Yih WK. (2019), Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, 38(12), 2126–2138.

Silva IR, Montalban, J., Oliveira, F. (2025), Maximized Sequential Probability Ratio Test Regression. Working paper - Sentinel (TIDE) project, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Examples

```
### Example. Three chunks of data.

### Firstly, it is necessary to set up the input parameters.
##  Note: cut off the "#" symbol before running the lines below.
#     AnalyzeSetUpRegression.CondPoisson(name="VaccineA",N=100,cc=20,alpha=0.05,
#     R0=1, rho=1,mref=999,title="Monitoring_vaccineA",
#     address="C:/Users/Ivair/Documents")


### Now we apply sequential tests to each of three chunks of data.
# ------------------------------------------------------------------------

## Test 1 - Situation of three covariates combinations, i.e.,
## three lines in the covariates matrix, and two covariates.
## Regarding the PersonTimeRatio parameter, in this first test
```

```
## we illustrate the situation with the same information 0.05 per
## covariates combination. For example, if the total historical person-time
## is V= 5000, and if the test-specific person-time (per covariates combination)
## is P=250, then PersonTimeRatio= c(250,250,250)/5000,
## which gives PersonTimeRatio=c(0.05,0.05,0.05).
## Note: cut off the "#" symbol before running the lines below.
#  AnalyzeRegression.CondPoisson(name= "VaccineA",test=1,events= c(3,2,4),
#  PersonTimeRatio= c(0.05,0.05,0.05),
#  covariates=matrix(c(0.1,1,0.2,0,0.3,1),3,2)  )

## Test 2 - Situation where there is only one event per covariates combination:
## Note: cut off the "#" symbol before running the line below.
#  AnalyzeRegression.CondPoisson(name= "VaccineA",test=2,events= c(1,1,1),
#  PersonTimeRatio= c(0.05,0.05,0.05),
#  covariates=matrix(c(0.3,0.9,0.25,0.1,0.35,15),3,2) )

## Test 3 - Situation of elevated number of events and only two groups:
## Note: cut off the "#" symbol before running the line below.
#  AnalyzeRegression.CondPoisson(name= "VaccineA",test=3,events= c(10,13),
#  PersonTimeRatio= c(0.25,0.3),
#  covariates=matrix(c(0.5,1,0.35,0.2),2,2) )
```

`AnalyzeRegression.Poisson`

Function for MaxSPRT regression analyses with Poisson data, without the need to know group sizes a priori.

Description

The function `AnalyzeRegression.Poisson` is used for either continuous or group sequential analysis, or for a combination of the two. Unlike `CV.Poisson`, it is not necessary to pre-specify the group sizes before the sequential analysis starts. Moreover, under the null hypothesis, the expected number of events, mu0, can be different for different observations. `AnalyzeRegression.Poisson` is run at each look at the data. Before running it by the first time, it is necessary to run the `AnalyzeSetUpRegression.Poisson` function.

Usage

```
AnalyzeRegression.Poisson(name,test,mu0="n",events,covariates,AlphaSpend="n")
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and it must be the same as the name given by the `AnalyzeSetupRegression.Poisson` function. Should never be the same as another sequential analysis that is run simultaneously on the same computer.

test	An integer indicating the number of hypothesis tests performed up to and including the current test. For example, if there were four prior looks at the data, and this is the fifth one, then "test=5". This number should be increased by one each time that the AnalyzeRegression.Poisson function is run for a new group of data, when it is part of the same sequential analysis. If not, there is an error message.
mu0	The test specific vector with expected number of events under the null hypothesis. The parameter mu0 must contain positive numbers. There is no default value
events	The test specific vector with the number of observed events per covariates combination.
covariates	Matrix with the covariates for the regression model. The i-th line of covariates has the information related to the i-th entry of the vector events. Each column of covariates corresponds to a different explanatory covariate.
AlphaSpend	The alpha spending function is specified in the AnalyzeSetUpRegression.Poisson function. At any look at the data, it is possible to over ride that pre-specified alpha spending plan by using the AlphaSpend parameter. AlphaSpend is a number representing the maximum amount of alpha (Type I error probabiliy) to be spent up to and including the current test. Because of the discrete nature of the Poisson distribution, the actual amount of alpha spent may be less than the maximum amount specified. It must be in the range (0,alpha]. The default value is no override, which means that, with the default AlphaSpend= "n", the function will use the alpha spending plan specified in the AnalyzeSetUpRegression.Poisson function.

Details

The function AnalyzeRegression.Poisson performs continuous or group MaxSPRT regression analysis for Poisson data based on the method proposed by Silva et al.(2025).

It can also be used for mixed continuous-group sequential analysis where some data arrives continuously while other data arrives in groups.

Unlike CV.Poisson, there is (i) no need to pre-specify the group sizes before the sequential analysis starts, (ii) a variety of alpha spending functions are available, and (iii) it is possible to include an offset term where, under the null hypothesis, different observations have different Poisson rates mu0.

In sequential analysis, data is formed by cumulative information, collected in separated chunks or groups, which are observed at different moments in time. AnalyzeRegression.Poisson is run each time a new group of data arrives at which time a new sequential test is conducted. When running AnalyzeRegression.Poisson, only the data from the new group should be included when calling the function. The prior data has been stored, and it will be automatically retrieved by AnalyzeRegression.Poisson, with no need to reenter that data. Before running AnalyzeRegression.Poisson for the first time, it is necessary to set up the sequential analysis using the AnalyzeSetUpRegression.Poisson function, which is run once, and just once, to define the sequential analysis parameters. For information about this, see the description of the AnalyzeSetUpRegression.Poisson function.

The function AnalyzeRegression.Poisson calculates critical values to determine if the null hypothesis should be rejected or not at each analysis. Critical values are given by the value of the

AlphaSpending function, and the test statistic is the Monte Carlo p-value calculated as proposed by Silva et al.(2025). This way, the null hypothesis H0:RR<= R0, where R0 is the testing margin given by the analyst in the AnalyzeSetUpRegression.Poisson, and RR is the true unknow relative risk to test.

After each test, the function also provides information about the amount of alpha that has been spent, the cumulative number of cases and controls, regression coefficient estimates, confidence intervals, and the maximum likelihood estimate of the relative risk per stratum of individuals observed during the sequential analysis.

Before running AnalyzeRegression.Poisson, it is necessary to specify a planned default alpha spending function, which is done using the AlphaSpendType parameter in the AnalyzeSetUpRegression.Poisson function. The default alpha spending plan is the polynomial power-type alpha spending plan parameterized with "rho=1". Different alpha spending plans can be obtained by selecting different values for rho (Silva, 2018).

In most cases, this pre-specified alpha spending function is used throughout the analysis, but if needed, it is possible to override it at any or each of the sequential tests. This is done using the AlphaSpend parameter in AnalyzeRegression.Poisson, which specifies the maximum amount of alpha to spend up to and including the current test. In this way, it is possible to use any alpha spending function, and not only the power-type available in AnalyzeSetUpRegression.Poisson. It means that the AlphaSpend parameter can be used to promote a flexible adaptive alpha spending plan that is not set in stone before the sequential analysis starts. The only requirement is that for a particular test with a new group of data, AlphaSpend must be decided before knowing the number of events in that group. To ensure a statistically valid sequential analysis, AlphaSpend can only depend on the values of m0. This is important.

The function AnalyzeRegression.Poisson is meant to perform the Poisson sequential analysis with a certain level of autonomy. After running a test, the code offers a synthesis about the general parameter settings, the main conclusions concerning the acceptance or rejection of the null hypothesis, and the historical information from previous tests.

Observe that, because the Poisson distribution is discrete, the target alpha spending will rarely be reached. The actual alpha spending is then shown to facilitate a realistic interpretation of the results.

The function AnalyzeRegression.Poisson was designed to instruct the user with minimal information about bugs from the code, or about non-applicable parameter input usages. Some entries are not applicable for the parameter inputs.

Value

result	Four data.frames (Decision_table, Relative_risk_estimates, Coefficients, Confidence_Intervals) with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests.

Acknowledgements

Development of the AnalyzeRegression.Poisson function was funded by: - Food and Drug Administration, Center for Drug Evaluation and Research, through the Mini-Sentinel Project (base version, documentation);
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (user defined alpha spending functions, improved documentation);

- National Council of Scientific and Technological Development (CNPq), Brazil, process number 302882/2022-7. (v4.3.1).
- Support Foundation to Minas Gerais State Research-Fapemig, Brazil, grant numbers PQ-00787-21 and RED-00133-21. (v3.1 to v4.3).

See also

AnalyzeSetUpRegression.Poisson: for setting up sequential analysis with the AnalyzeRegression.Poisson function, before the first look at the data.

Author(s)

Ivair Ramos Silva.

References

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials. London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30, 58–78.

Kulldorff M, Silva IR. (2015). Continuous post-market sequential safety surveillance with minimum events to signal. arxiv:1503.01978 [stat.ap].

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71(3), 851–858.

Silva IR, Kulldorff M, Yih W. Katherine. (2020), Optimal alpha spending for sequential analysis with binomial data. Journal of the Royal Statistical Society Series B, 82(4) p. 1141–1164.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance With Poisson Data. Methodol Comput Appl Probab, 20(2), 739-750.

Silva IR, Montalban, J., Oliveira, F. (2025), Maximized Sequential Probability Ratio Test Regression. Working paper - Sentinel (TIDE) project, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Examples

```
### Example. Three chunks of data.

### Firstly, it is necessary to set up the input parameters.
##  Note: cut off the "#" symbol before running the lines below.
#     AnalyzeSetUpRegression.Poisson(name="VaccineA",N=100,alpha=0.05,
#     R0=1, rho=1,mref=999,title="Monitoring_vaccineA",
#     address="C:/Users/Ivair/Documents")


### Now we apply sequential tests to each of three chunks of data.
# ------------------------------------------------------------------------

## Test 1 - Situation of three covariates combinations, i.e.,
```

```
## three lines in the covariates matrix.
## Note: cut off the "#" symbol before running the lines below.
#  AnalyzeRegression.Poisson(name= "VaccineA",test=1,mu0= c(2,2,2),
#  events= c(2,2,2), covariates=matrix(c(0.1,1,0.2,0,0.3,1),3,2)  )

## Test 2 - Situation where there is only one event per covariates combination:
## Note: cut off the "#" symbol before running the line below.
#  AnalyzeRegression.Poisson(name= "VaccineA",test=2,mu0= c(1.5,1.2,0.9),
#  events= c(1,1,1), covariates=matrix(c(0.3,0.9,0.25,0.1,0.35,15),3,2) )

## Test 3 - Situation of elevated number of events and only two groups:
## Note: cut off the "#" symbol before running the line below.
#  AnalyzeRegression.Poisson(name= "VaccineA",test=3,mu0= c(8.2,9.7),
#  events= c(10,13),covariates=matrix(c(0.5,1,0.35,0.2),2,2) )
```

`AnalyzeSetUp.Binomial` *Function to set up input parameters before using the* `Analyze.Binomial` *function for the first time.*

Description

The function `AnalyzeSetUp.Binomial` must be run ahead of `Analyze.Binomial` in order to set up the sequential analysis before the first group of data is analyzed. The function obtains the main parameter settings and performs basic calculations that are necessary for the subsequent sequential analysis.

Usage

```
AnalyzeSetUp.Binomial(name,N="n",alpha=0.05,zp=1,pp="n",
M=1,AlphaSpendType="optimal",power=0.9,RR=2,ConfIntWidth="n",ConfTimes=1,
Gamma=0.9,R0=1,ObjectiveMin="ETimeToSignal",rho=1,title="n",
address="n",Tailed="upper",cases_fraction=0,events_fraction=0)
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and the same as the name given in the subsequent calls to the `Analyze.Binomial` function. It cannot be the same as for another sequential analysis that is run simultaneously on the same computer. There is no default.
`N`	The maximum sample size, at which the sequential analysis stops without rejecting the null hypothesis. The default N="n" means that the optimal procedure will also find out the optimal sample size for the target power.
`alpha`	The overall significance level. Must be in the range (0,0.5]. The default is "alpha=0.05".

`zp` The prediction for z, the expected ratio between cases and controls under the null hypothesis that will be specified in the `Analyze.Binomial` function. This variable is only needed when AlphaSpendType= "Wald", and it is used to calculate the appropriate rejection boundary. If the z used in `Analyze.Binomial` during the actual sequential analysis is different from zp, that is okay, and the sequential analysis will still maintain the correct alpha level. The default value is "zp=1".

`pp` The prediction for p, the expected probability under the null hypothesis that will be specified in the `Analyze.Binomial` function. This variable is only needed when AlphaSpendType= "Wald", and it is used to calculate the appropriate rejection boundary. If the p used in `Analyze.Binomial` during the actual sequential analysis is different from pp, that is okay, and the sequential analysis will still maintain the correct alpha level. There is no default value.

`M` The minimum number of events required before the null hypothesis can be rejected. It must be a positive integer. The default value is "M=1".

`AlphaSpendType` The type of alpha spending function to be used. The options are AlphaSpendType= "optimal", the default, AlphaSpendType= "Wald", AlphaSpendType= "power-type". With the 'Wald' option, the Wald type upper rejection boundary is used, which is flat with respect to the likelihood ratio. With the power-type option, the alpha spending uses a power function with parameter rho, with rho defined by the user. With the optimal option, the code uses the alpha spending that minimizes expected time to signal or expected sample size. For more information, see Details. The alpha spending setting is automatically used when the `Analyze.Binomial` function is run, but, during the sequential analysis, and before each test, the user can always specify an arbitrary amount of alpha spending to be used up until and including that test. See below for details. The default is "optimal".

`power` The target power to be used as a constraint in the optimal alpha spending solution and for the robust alpha spending.

`RR` The relative risk for the target power. It is only applicable for 'AlphaSpendType=optimal'.

`ConfIntWidth` Positive value for a fixed-width and fixed accuracy confidence interval for the relative risk. Default is without confidence coeficient.

`ConfTimes` Times where the restriction on the confidence interval width is initiated for each entry of ConfIntWidth. Default is 1.

`Gamma` Confidence coefficient for the bounded width interval estimator of the relative risk for AlphaSpendType= "optimal". Default is 0.9. It has no effect when ConfIntWidth="n".

`R0` A positive real-valued number for the relative risk under H0, where R<=R0 if "Tailed=lower", R>=R0 if "Tailed=upper", or a two-dimensional vector for H0: R0_1<= R <= R0_2 if "Tailed=two". Default is 1.

`ObjectiveMin` The objective function to minimize in case of 'AlphaSpendType=optimal'. The default is 'ObjectiveMin=ETimeToSignal'. The other option is 'ObjectiveMin=ESampleSize'.

`rho` The parameter rho is used to build the target alpha spending function according to a power-type function. See below for details. It is not used for other alpha

spending options. The variable rho must be a positive number. The default value is "rho=1".

`title` Title for the results shown in the output tables and the illustrative graphics. It can be any text string. The default is that there is no title.

`address` The address of the directory where the settings information of this sequential analysis is saved.

`Tailed` Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1. For this version of the package, only Tailed="upper" is active.

`cases_fraction` This activates the construction of a robust alpha spending due to unstable data. The deafult is cases_fraction=0, which means that controls will not be wrongly registered as cases. If cases_fraction is in (0, 1), then it represents a percent of the cummulative number of events during the surveillance. If cases_fraction >= 1, it represents the number of controls taken as cases in each testing time.

`events_fraction` This activates the construction of a robust alpha spending due to unstable data. The deafult is events_fraction=0, which means that there will not occur events disappearing during the surveillance. If events_fraction is in (0, 1), then it represents a percent of the cummulative number of events during the surveillance. If events_fraction >= 1, it represents the number of events disappearing during the analysis.

Details

The function `AnalyzeSetUp.Binomial` has to be executed once, but just once, to set up the general statistical characteristics of the intended sequential analysis, which is performed using the companion `Analyze.Binomial` function.

Sequential analysis methods are devoted to analyze data sets that accrue cumulatively over time, by conducting multiple statistical tests sequentially as more data accrues. In such a setting, it is important to carefully plan the sequential analysis before the first data arrives. For example, it is important to maintain certain analysis parameter values over time to avoid counting the same data twice, and to make sure that there are no changes in the past data that has already been included in a prior test. To avoid these kinds of problems, the `AnalyzeSetUp.Binomial` function is used to set the analysis parameters a priori and to create a place to save the data as it accumulates over time. At the time of each sequential test, this information is then automatically imported by the `Analyze.Binomial` function, to ensure the correct concatenation of old and new information.

At each test, the function `Analyze.Binomial` makes this concatenation automatically, but it will only work if the function `AnalyzeSetUp.Binomial` is executed before performing the very first test.

When running `AnalyzeSetUp.Binomial`, the user has the opportunity to choose the directory where the file with the general setup information and the historical data are to be saved. Important: The location of this parameter and data file is saved in the temporary directory, so that directory cannot be cleaned until the sequential analysis has been completed. Each sequential analysis needs a different identifier, which is set using the "name" parameter. Once a name is chosen, it has to be written exactly the same way when running the function `Analyze.Binomial`.

`AnalyzeSetUp.Binomial` and `Analyze.Binomial` works for different types of alpha spending plans ($F(t)$). One option is to use the classical Wald type upper rejection boundary, which is

flat with respect to the likelihood function. This is the same boundary used by the `CV.Binomial` and `CV.G.Binomial` functions. In order to use this boundary, one should pre-specify the binomial probability p under the null hypothesis, or, equivalently, the ratio $z = 1/p - 1$, which is the number of controls matched to each case in a matched analysis. For example, if the probability of having a case (instead of a control) is $p = 1 = (1 + z) = 0.5$, then we have "z=1" (1:1 matching ratio), and, if p = 0.25, we have "z=3" (1:3 matching ratio). A third option, the default, is the optimal alpha spending derived by Silva and Kulldorff (2018), which demands users to choose between minimizing expected time to signal or expected sample size. In this case, it is necessary to specify target power and relative risk. The faults are 'power=0.9' and 'RR=2'.

In `AnalyzeSetUp.Binomial`, the predicted z is specified (the input zp), but if it turns out that the actual z is different, that is okay, since the actual z that is specified in `Analyze.Binomial` does not have to be the same as the predicted zp that is specified in `AnalyzeSetUp.Binomial`. The latter is only used to set the alpha spending plan. The former, the actual z, is used to calculate the likelihood function which in turn determines whether the null hypothesis should be rejected or not. If the actual z is variable, so that it is different for different observations, we recommend setting the predicted z to be our best guess about the average of the actual zs. Alternatively, instead of zp the user can specify pp, the best guess about the average of the actual ps. Note that only one of these parameters has to be specified, but if both are entered the code will only work if zp and pp are such that pp=1/(1+zp). Otherwise, an error message will appear to remind that such condition must be complied.

Another alpha spending option is the power-type alpha spending plan (Kim and DeMetz 1987, p150; Jennison and Turnbull 2000, p148), with parameter rho: $F(t) = alpha * t^{rho}$, where $alpha$ is the overall significance level and t is a fraction of N, the maximum length of sequential analysis. According to Silva (2018), 'rho=0.5' is indicated when expected time to signal is the design criterion. The choice 'rho=1' is a proper option that provides a balance between time to signal and maximum lenght and surveillance, hence this is the default in `AnalyzeSetUp.Binomial`.

The third option for alpha spending selection is the optimal solution. This is the default. In this case, the alpha spending is obtained by means of the method introduced by Silva and Kulldorff method (Silva and Kulldorff, 2018), which is an exact method for finding the optimal alpha spending through linera programing. The optimal option works for large sample sizes such as N=300, but it can take very long time to run in such cases. For moderate N values, such as N=120, the code takes around 10 minutes to run in a regular PC(Windows 7, Intel(R) Core(TM) i7-2675QM CPU, 2.20GHz). Although "optimal" is the default, an error message will appear, asking for another AlphaSpendType choice, if this default is used combined with N greater than 300.

Another important issue involving the option "optimal" is the choice of tuning parameters behind the method of Silva and Kulldorff (2018). According to Silva and Kulldorff (2018), the user has to specify a target power and a target relative risk, besides the sample size N, to use their proposed optimal method. For simplicity, as there are probabilistical restrictions to use certain combinations of target power and relative risk with cetain N values, the default values are 'power=0.9' and 'RR=2'. This is no a critical issue because, as explained by Silva and Kulldorff (2018), there is no serious impact from chosen RR very different from the actual relative risk.

In addition to selecting the alpha spending plan, it is necessary to specify the overall alpha, or maximum Type I error probability, for the sequential analysis as a whole. It is also necessary to specify the maximum length of the sequential analysis, N, so that the sequential analysis stops without rejecting the null hypothesis when a total of N observations are obtained.

The possibility of constructing the alpha spending restricted to a desired maximum length for the confidence interval, activated by the "ConfIntWidth" parameter, is based on the method proposed

by Silva and Zhuang (2022).

Value

`inputSetUp` The `AnalyzeSetUp.Binomial` function creates a data.frame with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests. The 'inputSetUp' data.frame is used by `Analyze.Binomial`, then it must be available when running `Analyze.Binomial`, but there is no need to manually look at it.

Acknowledgements

Development of the AnalyzeSetUp.Binomial function was funded by:
- Food and Drug Administration, Center for Drug Evaluation and Research, through the Mini-Sentinel Project (base version, documentation);
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (user defined alpha spending functions, improved documentation);
We thank Claudia Coronel-Moreno for valuable editorial support.

See also

`Analyze.Binomial`: for running the sequential analysis that was set up using the `AnalyzeSetUp.Binomial` function.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, *no. ISBN 0-8493-0316-8, London: Chapman and Hall/CRC*.

Kim K, DeMets DL. (1987), Design and Analysis of Group Sequential Tests Based on the Type I Error Spending Rate Function. Biometrika, **74**, n.1: 149–154.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Kulldorff M, Silva IR. (2015). Continuous post-market sequential safety surveillance with minimum events to signal. arxiv:1503.01978 [stat.ap].

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR, Kulldorff M, Yih W. Katherine. (2020), Optimal alpha spending for sequential analysis with binomial data. Journal of the Royal Statistical Society Series B, 82(4) p. 1141–1164.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107-118.

Silva IR, Zhuang, Y. (2022), Bounded-width confidence interval following optimal sequential analysis of adverse events with binary data, Statistical Methods in Medical Research, 31(12), 2323–2337.

Examples

```
# See example in the description of the Analyze.Binomial function.
```

AnalyzeSetUp.CondPoisson

Function to set up input parameters before using the Analyze.CondPoisson *function for the first time.*

Description

The function AnalyzeSetUp.CondPoisson must be run ahead of Analyze.CondPoisson in order to set up the sequential analysis before the first group of data is analyzed. The function obtains the main parameter settings and performs basic calculations that are necessary for the subsequent sequential analysis.

Usage

```
AnalyzeSetUp.CondPoisson(name,SampleSizeType="Events",T="n",K="n",cc,
alpha=0.05,M=1, AlphaSpendType="Wald",rho=1,title="n",address="n",Tailed="upper",
events_fraction=0,power=0.9,RR=2)
```

Arguments

name	The name of the sequential analysis. Must be identical for all looks at the data, and the same as the name given in the subsequent calls to the Analyze.CondPoisson function. It cannot be the same as for another sequential analysis that is run simultaneously on the same computer. There is no default.
SampleSizeType	It is a string specifying the scale of the maximum sample size at which the sequential analysis stops without rejecting the null hypothesis. The only two possibilities are "SampleSizeType=PersonTimeRatio" or "SampleSizeType=Events". The default is "SampleSizeType=Events". See details.
T	Maximum sample size defined in the scale of the ratio between surveillance and historical person-time. This only produces effects when "SampleSizeType=PersonTimeRatio".
K	Maximum sample size defined in the scale of the number of events observed in the surveillance period. This only produces effects when "SampleSizeType=Events".
cc	Number of events observed in the historical period.
alpha	The overall significance level. Must be in the range (0,0.5]. The default is "alpha=0.05".
M	The minimum number of events required before the null hypothesis can be rejected. It must be a positive integer. The default value is "M=1".

`AlphaSpendType` The type of alpha spending function to be used. The options are AlphaSpendType="Wald" and AlphaSpendType="power-type". With the 'Wald' option, the Wald type upper rejection boundary is used, which is flat with respect to the likelihood ratio. With the power-type option, the alpha spending uses a power function with parameter rho, with rho defined by the user. This alpha spending setting is automatically used when the `Analyze.CondPoisson` function is run, but, during the sequential analysis, and before each test, the user can always specify an arbitrary amount of alpha spending to be used up until and including that test. See below for details.

`rho` The parameter rho is used to build the target alpha spending function according to a power-type function. See below for details. It is not used for other alpha spending options. The variable rho must be a positive number. The default value is "rho=1".

`title` Title for the results shown in the output tables and the illustrative graphics. It can be any text string. The default is that there is no title.

`address` The address of the directory where the settings information of this sequential analysis is saved.

`Tailed` Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

`events_fraction`
This activates the construction of a robust alpha spending due to unstable data. The default is events_fraction=0, which means that there will not occur events added to the data by mistake during the surveillance. If events_fraction is in (0, 1), then it represents a percent of the total sample size, K, under H0. If events_fraction >= 1, it represents a constant number of events added by mistake during the analysis.

`power` The target power for the robust alpha spending. The default is power= 0.9.

`RR` The target relative risk for the target power when the robust alpha spending is used. The default is RR= 2.

Details

The function `AnalyzeSetUp.CondPoisson` has to be executed once, but just once, to set up the general statistical characteristics of the intended sequential analysis, which is performed using the companion `Analyze.CondPoisson` function.

Sequential analysis methods are devoted to analyze data sets that accrue cumulatively over time, by conducting multiple statistical tests sequentially as more data accrues. In such a setting, it is important to carefully plan the sequential analysis before the first data arrives. For example, it is important to maintain certain analysis parameter values over time to avoid counting the same data twice, and to make sure that there are no changes in the past data that has already been included in a prior test. To avoid these kinds of problems, the `AnalyzeSetUp.CondPoisson` function is used to set the analysis parameters a priori and to create a place to save the data as it accumulates over time. At the time of each sequential test, this information is then automatically imported by the `Analyze.CondPoisson` function, to ensure the correct concatenation of old and new information.

At each test, the function `Analyze.CondPoisson` makes this concatenation automatically, but it will only work if the function `AnalyzeSetUp.CondPoisson` is executed before performing the very first test.

When running `AnalyzeSetUp.CondPoisson`, the user has to choose the directory where the file with the general setup information and the historical data are to be saved. This step is mandatory and error messages are reported if a non-valid address is informed. Important: The location of this parameter and data file is saved in the temporary directory, so that directory cannot be cleaned until the sequential analysis has been completed. Each sequential analysis needs a different identifier, which is set using the "name" parameter. Once a name is chosen, it has to be written exactly in the same way when running the function `Analyze.CondPoisson`.

`AnalyzeSetUp.CondPoisson` and `Analyze.CondPoisson` work for different types of alpha spending plans ($F(t)$). One option is to use the classical Wald type upper rejection boundary, which is flat with respect to the likelihood function. This is the same boundary used by the `CV.CondPoisson` function.

Another alpha spending option is the power-type alpha spending plan (Kim and DeMetz 1987, p150; Jennison and Turnbull 2000, p148), with parameter rho: $F(t) = alpha * t^{rho}$, where $alpha$ is the overall significance level and t is a fraction of SampleSize, the maximum length of sequential analysis.

Attention is required for the input parameter "SampleSizeType". With this parameter, the user can choose the scale of the maximum sample size at which the surveillance is stoped without rejecting the null hypothesis. The idea of having two options for defining the scale of the maximum sample size, "SampleSizeType=PersonTimeRatio" or "SampleSizeType=Events", was introduced by Silva et al. (2019a). With SampleSizeType="PersonTimeRatio", the upper limit on the time of surveillance is given in the scale of the ratio between the cumulative person-time from the surveillance data up to the kth event, P_k, by the person-time from the historical data, denoted by V. These are the notations used by Silva et al. (2019a) and Silva et al. (2019b).

If SampleSizeType="PersonTimeRatio", then the user has to inform a positive value for the input parameter "T". Usually, choices between 2 and 5 are adequate. With SampleSizeType="Events", the upper limit is given in the scale of the number of events observed during the surveillance, and hence the user must specify a positive integer for the input parameter "K", such as e.g. "K=50" or "K=150". For more details on the exact calculations of critical values and alpha spending implemented in this package, and of all the other statistical performance measures also available, see the works of Silva et al. (2019a) and Silva et al. (2019b).

Value

`inputSetUp` The `AnalyzeSetUp.CondPoisson` function creates a data.frame with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests. The 'inputSetUp' data.frame is used by `Analyze.CondPoisson`, then it must be available when running `Analyze.CondPoisson`, but there is no need to manually look at it.

Acknowledgements

Development of the AnalyzeSetUp.CondPoisson function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999.
- Foundation for Research Support of Minas Gerais State (FAPEMIG), MG, Brazil, through the grant Demanda Universal.

See also

Analyze.CondPoisson: for running the sequential analysis that was set up using the AnalyzeSetUp.CondPoisson function.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, *no. ISBN 0-8493-0316-8, London: Chapman and Hall/CRC*.

Kim K, DeMets DL. (1987), Design and Analysis of Group Sequential Tests Based on the Type I Error Spending Rate Function. Biometrika, **74**, n.1: 149–154.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30: 58–78.

Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Li L, Kulldorff M. (2010). A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71(3), 851–858.

Silva IR, Li L, Kulldorff M. (2019a), Exact conditional maximized sequential probability ratio test adjusted for covariates. Sequential Analysis, 38(1), 115–133.

Silva IR., Lopes LM., Dias P., Yih WK. (2019b). Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, 38(12), 2126–2138.

Examples

```
# See example in the description of the Analyze.CondPoisson function.
```

AnalyzeSetUp.Multinomial

Function to set up input parameters before using the Analyze.Multinomial *function for the first time.*

Description

The function AnalyzeSetUp.Multinomial must be run ahead of Analyze.Multinomial in order to set up the sequential analysis before the first group of data is analyzed. The function obtains the main parameter settings and performs basic calculations that are necessary for the subsequent sequential analysis.

Usage

```
AnalyzeSetUp.Multinomial(name,N=200,alpha=0.05,AlphaSpendType=1,
R0=1,R1=2,rho=1,pmin=0.05, pmax=0.95,target_power=0.8,
gamma=0.9,m=100000,title="n",ExposuresNames="n",address="n")
```

Arguments

name	The name of the sequential analysis. Must be identical for all looks at the data, and the same as the name given in the subsequent calls to the Analyze.Multinomial function. It cannot be the same as for another sequential analysis that is run simultaneously on the same computer. There is no default.
N	The maximum sample size, at which the sequential analysis stops without rejecting the null hypothesis.
alpha	The overall significance level. Must be in the range (0,0.5]. The default is "alpha=0.05".
AlphaSpendType	Type of robust alpha spending. The possible values are 1 and 2 according to Silva and Maro(2025a). The default is equal to 1.
R0	Test margin under the null hypothesis. Must be a positive number. Default is R0=1.
R1	Relative risk under the alternative hypothesis given the target_power.
rho	The parameter rho is used to build the target alpha spending function according to a power-type function. The variable rho must be a positive number. The default value is "rho=1".
pmin	Minimum value of the multinomial probability to activate its portion of alpha spending at each test. See Silva and Maro(2025) for more details. The default is pmin=0.05
pmax	Maximum value of the multinomial probability to activate its portion of alpha spending at each test. See Silva and Maro(2025) for more details. The default is pmin=0.05
target_power	The target power for detecting an elevated risk (>R0) for each entry of the multinomial vector.
gamma	Confidence coefficient for the interval estimators of the relative risk of each multinomial entry. Default is 0.9.
m	Monte Carlo replications of the multinomial for critical values calculations.
title	Title for the results shown in the output tables and the illustrative graphics. It can be any text string. The default is that there is no title.
ExposuresNames	This is to inform the name of the exposures related to each entry of the multinomial vector. For example, it can be c("A","B","AB") for a three-exposure concomitant vaccination with two vaccines, vaccines "A" and "B". There is no default.
address	The address of the directory where the settings information of this sequential analysis is saved.

Details

The function `AnalyzeSetUp.Multinomial` has to be executed once, but just once, to set up the general statistical characteristics of the intended sequential analysis, which is performed using the companion `Analyze.Multinomial` function.

Sequential analysis methods are devoted to analyze data sets that accrue cumulatively over time, by conducting multiple statistical tests sequentially as more data accrues. In such a setting, it is important to carefully plan the sequential analysis before the first data arrives. For example, it is important to maintain certain analysis parameter values over time to avoid counting the same data twice, and to make sure that there are no changes in the past data that has already been included in a prior test. To avoid these kinds of problems, the `AnalyzeSetUp.Multinomial` function is used to set the analysis parameters a priori and to create a place to save the data as it accumulates over time. At the time of each sequential test, this information is then automatically imported by the `Analyze.Multinomial` function, to ensure the correct concatenation of old and new information.

At each test, the function `Analyze.Multinomial` makes this concatenation automatically, but it will only work if the function `AnalyzeSetUp.Multinomial` is executed before performing the very first test.

When running `AnalyzeSetUp.Multinomial`, the user has the opportunity to choose the directory where the file with the general setup information and the historical data are to be saved. Important: The location of this parameter and data file is saved in the temporary directory, so that directory cannot be cleaned until the sequential analysis has been completed. Each sequential analysis needs a different identifier, which is set using the "name" parameter. Once a name is chosen, it has to be written exactly the same way when running the function `Analyze.Multinomial`.

The alpha spending plan is designed in a way to ensure that a target statistical power, gamma, is endured for detecting a target increased risk, R1, for each multinomial entry. This is possible by means of a new concept introduced by Silva and Maro(2025), the robust alpha spending plan. For more details about the method and the construction of the critical values, see Silva and Maro(2025).

In addition to selecting the alpha spending plan, it is necessary to specify the overall alpha, or maximum Type I error probability, for the sequential analysis as a whole. It is also necessary to specify the maximum length of the sequential analysis, N, so that the sequential analysis stops without rejecting the null hypothesis when a total of N observations are obtained.

Value

`inputSetUp` The `AnalyzeSetUp.Multinomial` function creates a data.frame with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests.

Acknowledgements

Development of the AnalyzeSetUp.Multinomial function was funded by:
- Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.
- Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq), Brazil, and Fundacao de Amparo a Pesquisa do Estado de Minas Gerais (FAPEMIG), Brazil.

See also

Analyze.Multinomial: for running the sequential analysis that was set up using the AnalyzeSetUp.Multinomial function.

Author(s)

Ivair Ramos Silva, Judith Maro.

References

Silva IR, Maro J. (2025), Adaptive Sequential Multiple Hypotheses Testing for Seasonal Concomitant Vaccines Safety Surveillance. Working paper, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Examples

```
# See example in the description of the Analyze.Multinomial function.
```

AnalyzeSetUp.Poisson *Function to set up input parameters before using the* Analyze.Poisson *function for the first time.*

Description

The function AnalyzeSetUp.Poisson must be run ahead of Analyze.Poisson in order to set up the sequential analysis before the first group of data is analyzed. The function obtains the main parameter settings and performs basic calculations that are necessary for the subsequent sequential analysis.

Usage

```
AnalyzeSetUp.Poisson(name,SampleSize,alpha=0.05,D=0,M=1,
AlphaSpendType="Wald",rho=1,R0=1,title="n",address="n",
Tailed="upper",events_fraction=0,power=0.9,RR=2)
```

Arguments

name	The name of the sequential analysis. Must be identical for all looks at the data, and the same as the name given in the subsequent calls to the Analyze.Poisson function. It cannot be the same as for another sequential analysis that is run simultaneously on the same computer. There is no default.
SampleSize	The maximum length of surveillance at which the sequential analysis stops without rejecting the null hypothesis. It is defined in terms of the expected sample size under the null hypothesis. There is no default value.

`alpha` The overall significance level. Must be in the range (0,0.5]. The default is "alpha=0.05".

`D` The expected number of events under the null hypothesis at the first look at the data. This is used when there is an initial large chunk of data arriving, followed by continuous sequential analysis. The default value is D=0, which is also the best choice. This means that there is no delay in the start of the sequential analyses. If D is very large, the maximum sample size will be set equal to D if a non-sequential analysis provides the desired power.

`M` The minimum number of events needed before the null hypothesis can be rejected. It must be a positive integer. A good rule of thumb is to set M=4 (Kulldorff and Silva, 2015). The default value is M=1, which means that even a single event can reject the null hypothesis if it occurs sufficiently early.

`AlphaSpendType` The type of alpha spending function to be used. The options are AlphaSpendType="Wald" and AlphaSpendType="power-type". With the 'Wald' option, the Wald type upper rejection boundary is used, which is flat with respect to the likelihood ratio. With the power-type option, the alpha spending uses a power function with parameter rho, with rho defined by the user. This alpha spending setting is automatically used when the `Analyze.Poisson` function is run, but, during the sequential analysis, and before each test, the user can always specify an arbitrary amount of alpha spending to be used up until and including that test. See below for details.

`rho` The parameter rho is used to build the target alpha spending function according to a power-type function. See below for details. It is not used for other alpha spending options. The variable rho must be a positive number. The default value is "rho=1".

`R0` A positive real-valued number for the relative risk under H0, where R<=R0 if "Tailed=lower", R>=R0 if "Tailed=upper", or a two-dimensional vector for H0: R0_1<= R <= R0_2 if "Tailed=two". Default is 1.

`title` Title for the results shown in the output tables and the illustrative graphics. It can be any text string. The default is that there is no title.

`address` The address of the directory where the settings information of this sequential analysis is saved.

`Tailed` Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

`events_fraction`

This activates the construction of a robust alpha spending due to unstable data. The deafult is events_fraction=0, which means that there will not occur events added to the data by mistake during the surveillance. If events_fraction is in (0, 1), then it represents a percent of the total sample size under H0. If events_fraction >= 1, it represents constant number of events added by mistake during the analysis.

`power` The target power for the robust alpha spending. The default is power= 0.9.

`RR` The target relative risk for the target power when the robust alpha spending is used. The default is RR= 2.

Details

The function AnalyzeSetUp.Poisson has to be executed once, but just once, to set up the general statistical characteristics of the intended sequential analysis, which is performed using the companion Analyze.Poisson function.

Sequential analysis methods are devoted to analyze data sets that accrue cumulatively over time, by conducting multiple statistical tests sequentially as more data accrues. In such a setting, it is important to carefully plan the sequential analysis before the first data arrives. For example, it is important to maintain certain analysis parameter values over time to avoid counting the same data twice, and to make sure that there are no changes in the past data that has already been included in a prior test. To avoid these kinds of problems, the AnalyzeSetUp.Poisson function is used to set the analysis parameters a priori and to create a place to save the data as it accumulates over time. At the time of each sequential test, this information is then automatically imported by the Analyze.Poisson function, to ensure the correct concatenation of old and new information.

At each test, the function Analyze.Poisson makes this concatenation automatically, but it will only work if the function AnalyzeSetUp.Poisson is executed before performing the very first test.

When running AnalyzeSetUp.Poisson, the user has to choose the directory where the file with the general setup information and the historical data are to be saved. This step is mandatory and error messages are reported if non-valid address is informed. Important: The location of this parameter and data file is saved in the temporary directory, so that directory cannot be cleaned until the sequential analysis has been completed. Each sequential analysis needs a different identifier, which is set using the "name" parameter. Once a name is chosen, it has to be written exactly the same way when running the function Analyze.Poisson.

AnalyzeSetUp.Poisson and Analyze.Poisson work for different types of alpha spending plans ($F(t)$). One option is to use the classical Wald type upper rejection boundary, which is flat with respect to the likelihood function. This is the same boundary used by the CV.Poisson and CV.G.Poisson functions.

Another alpha spending option is the power-type alpha spending plan (Kim and DeMetz 1987, p150; Jennison and Turnbull 2000, p148), with parameter rho: $F(t) = alpha * t^{rho}$, where $alpha$ is the overall significance level and t is a fraction of SampleSize, the maximum length of sequential analysis. According to Silva (2018), the choice 'rho=1' is indicated when minimization of expected time to signal is a design criterion, which is then the default in AnalyzeSetUp.Poisson..

Value

inputSetUp — The AnalyzeSetUp.Poisson function creates a data.frame with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests. The 'inputSetUp' data.frame is used by Analyze.Poisson, then it must be available when running Analyze.Poisson, but there is no need to manually look at it.

Acknowledgements

Development of the AnalyzeSetUp.Poisson function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999.

See also

Analyze.Poisson: for running the sequential analysis that was set up using the AnalyzeSetUp.Poisson function.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, *no. ISBN 0-8493-0316-8, London: Chapman and Hall/CRC*.

Kim K, DeMets DL. (1987), Design and Analysis of Group Sequential Tests Based on the Type I Error Spending Rate Function. Biometrika, **74**, n.1: 149–154.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance With Poisson Data. Methodol Comput Appl Probab, 20(2), 739-750.

Silva IR, Maro J, Kulldorff M. (2021). Exact sequential test for clinical trials and post-market drug and vaccine safety surveillance with Poisson and binary data. Statistics in Medicine, DOI: 10.1002/sim.9094.

Examples

```
# See example in the description of the Analyze.Poisson function.
```

AnalyzeSetUp.wBinomial

Function to set up input parameters before using the Analyze.wBinomial *function for the first time.*

Description

The function AnalyzeSetUp.wBinomial must be run ahead of Analyze.wBinomial in order to set up the sequential analysis before the first group of data is analyzed. The function obtains the main parameter settings and performs basic calculations that are necessary for the subsequent sequential analysis.

Usage

```
AnalyzeSetUp.wBinomial(name,N,alpha=0.05,M=1,rho=0.5,
title="n",address="n",Tailed="upper")
```

Arguments

name	The name of the sequential analysis. Must be identical for all looks at the data, and the same as the name given in the subsequent calls to the Analyze.wBinomial function. It cannot be the same as for another sequential analysis that is run simultaneously on the same computer. There is no default.
N	The maximum sample size, at which the sequential analysis stops without rejecting the null hypothesis.
alpha	The overall significance level. Must be in the range (0,0.5]. The default is "alpha=0.05".
M	The minimum number of events required before the null hypothesis can be rejected. It must be a positive integer. The default value is "M=1".
rho	The parameter rho is used to build the target alpha spending function according to a power-type function. See below for details. It is not used for other alpha spending options. The variable rho must be a positive number. The default value is "rho=0.5".
title	Title for the results shown in the output tables and the illustrative graphics. It can be any text string. The default is that there is no title.
address	The address of the directory where the settings information of this sequential analysis is saved.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="two" for H0:RR=1.

Details

The function AnalyzeSetUp.wBinomial has to be executed once, but just once, to set up the general statistical characteristics of the intended sequential analysis, which is performed using the companion Analyze.wBinomial function.

Sequential analysis methods are devoted to analyze data sets that accrue cumulatively over time, by conducting multiple statistical tests sequentially as more data accrues. In such a setting, it is important to carefully plan the sequential analysis before the first data arrives. For example, it is important to maintain certain analysis parameter values over time to avoid counting the same data twice, and to make sure that there are no changes in the past data that has already been included in a prior test. To avoid these kinds of problems, the AnalyzeSetUp.wBinomial function is used to set the analysis parameters a priori and to create a place to save the data as it accumulates over time. At the time of each sequential test, this information is then automatically imported by the Analyze.wBinomial function, to ensure the correct concatenation of old and new information.

At each test, the function Analyze.wBinomial makes this concatenation automatically, but it will only work if the function AnalyzeSetUp.wBinomial is executed before performing the very first test.

When running AnalyzeSetUp.wBinomial, the user has the opportunity to choose the directory where the file with the general setup information and the historical data are to be saved. Important:

The location of this parameter and data file is saved in the temporary directory, so that directory cannot be cleaned until the sequential analysis has been completed. Each sequential analysis needs a different identifier, which is set using the "name" parameter. Once a name is chosen, it has to be written exactly the same way when running the function `Analyze.wBinomial`.

`AnalyzeSetUp.wBinomial` and `Analyze.wBinomial` works for different types of alpha spending plans ($F(t)$). The alpha spending option is the power-type alpha spending plan (Kim and DeMetz 1987, p150; Jennison and Turnbull 2000, p148), with parameter rho: $F(t) = alpha * t^{rho}$, where $alpha$ is the overall significance level and t is a fraction of N, the maximum length of sequential analysis. According to Silva (2018), 'rho=0.5' is indicated when expected time to signal is the design criterion, hence this is the default in `AnalyzeSetUp.wBinomial`.

In addition to selecting the alpha spending plan, it is necessary to specify the overall alpha, or maximum Type I error probability, for the sequential analysis as a whole. It is also necessary to specify the maximum length of the sequential analysis, N, so that the sequential analysis stops without rejecting the null hypothesis when a total of N observations are obtained.

Value

`inputSetUp` The `AnalyzeSetUp.wBinomial` function creates a data.frame with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests. The 'inputSetUp' data.frame is used by `Analyze.wBinomial`, then it must be available when running `Analyze.wBinomial`, but there is no need to manually look at it.

Acknowledgements

Development of the AnalyzeSetUp.Binomial function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (user defined alpha spending functions, improved documentation);

See also

`Analyze.wBinomial`: for running the sequential analysis that was set up using the `AnalyzeSetUp.wBinomial` function.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, *no. ISBN 0-8493-0316-8, London: Chapman and Hall/CRC*.

Kim K, DeMets DL. (1987), Design and Analysis of Group Sequential Tests Based on the Type I Error Spending Rate Function. Biometrika, **74**, n.1: 149–154.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Kulldorff M, Silva IR. (2015). Continuous post-market sequential safety surveillance with minimum events to signal. arxiv:1503.01978 [stat.ap].

Silva IR, Gagne J, Najafzadeh M, Kulldorff M. (2020). Exact Sequential Analysis for Multiple Weighted Binomial Endpoints. Statistics in Medicine, 39(3), 340–351.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva, IR and Kulldorff, M. and Yih, W. Katherine (2020). Optimal alpha spending for sequential analysis with binomial data. Journal of the Royal Statistical Society Series B, 82(4) p. 1141–1164.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107-118.

Examples

```
# See example in the description of the Analyze.wBinomial function.
```

`AnalyzeSetUpRegression.Binomial`
Function to set up input parameters before using the `AnalyzeRegression.Binomial` *function for the first time.*

Description

The function `AnalyzeSetUpRegression.Binomial` must be run ahead of `AnalyzeRegression.Binomial` in order to set up the sequential analysis before the first group of data is analyzed. The function obtains the main parameter settings and performs basic calculations that are necessary for the subsequent sequential analysis.

Usage

```
AnalyzeSetUpRegression.Binomial(name,N="n",alpha=0.05,
R0=1,rho=1,mref=999,title="n",address="n")
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and the same as the name given in the subsequent calls to the `AnalyzeRegression.Binomial` function. It cannot be the same as for another sequential analysis that is run simultaneously on the same computer. There is no default.
`N`	The maximum sample size, at which the sequential analysis stops without rejecting the null hypothesis.
`alpha`	The overall significance level. Must be in the range (0,0.5]. The default is "alpha=0.05".
`R0`	A positive real-valued number for the relative risk under R<=R0. Default is R0=1.

rho	The parameter rho is used to build the target alpha spending function according to a power-type function. See below for details. The variable rho must be a positive number. The default value is "rho=1".
mref	The parameter mref is the minimum number of Monte Carlo replications to calculate the p-value per test following the method by Silva at al(2025).
title	Title for the results shown in the output tables and the illustrative graphics. It can be any text string. The default is that there is no title.
address	The address of the directory where the settings information of this sequential analysis is saved.

Details

The function AnalyzeSetUpRegression.Binomial has to be executed once, but just once, to set up the general statistical characteristics of the intended sequential analysis, which is performed using the companion AnalyzeRegression.Binomial function.

Sequential analysis methods are devoted to analyze data sets that accrue cumulatively over time, by conducting multiple statistical tests sequentially as more data accrues. In such a setting, it is important to carefully plan the sequential analysis before the first data arrives. For example, it is important to maintain certain analysis parameter values over time to avoid counting the same data twice, and to make sure that there are no changes in the past data that has already been included in a prior test. To avoid these kinds of problems, the AnalyzeSetUpRegression.Binomial function is used to set the analysis parameters a priori and to create a place to save the data as it accumulates over time. At the time of each sequential test, this information is then automatically imported by the AnalyzeRegression.Binomial function, to ensure the correct concatenation of old and new information.

At each test, the function AnalyzeRegression.Binomial makes this concatenation automatically, but it will only work if the function AnalyzeSetUpRegression.Binomial is executed before performing the very first test.

When running AnalyzeSetUpRegression.Binomial, the user has the opportunity to choose the directory where the file with the general setup information and the historical data are to be saved. Important: The location of this parameter and data file is saved in the temporary directory, so that directory cannot be cleaned until the sequential analysis has been completed. Each sequential analysis needs a different identifier, which is set using the "name" parameter. Once a name is chosen, it has to be written exactly the same way when running the function AnalyzeRegression.Binomial.

AnalyzeSetUpRegression.Binomial and AnalyzeRegression.Binomial work for the power-type alpha spending plan (Kim and DeMetz 1987, p150; Jennison and Turnbull 2000, p148), with parameter rho: $F(t) = alpha * t^{rho}$, where $alpha$ is the overall significance level and t is a fraction of N, the maximum length of sequential analysis. According to Silva (2018), 'rho=0.5' is indicated when expected time to signal is the design criterion. The choice 'rho=1' is a proper option that provides a balance between time to signal and maximum lenght and surveillance, hence this is the default in AnalyzeSetUpRegression.Binomial.

In addition to selecting the alpha spending plan, it is necessary to specify the overall alpha, or maximum Type I error probability, for the sequential analysis as a whole. It is also necessary to specify the maximum length of the sequential analysis, N, so that the sequential analysis stops without rejecting the null hypothesis when a total of N observations are obtained.

Value

`inputSetUp` The `AnalyzeSetUpRegression.Binomial` function creates four data.frames (Decision_table, Relative_risk_estimates, Coefficients, Confidence_Intervals) with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests. The 'inputSetUp' list is used by `AnalyzeRegression.Binomial`, then it must be available when running `AnalyzeRegression.Binomial`, but there is no need to manually look at it.

Acknowledgements

Development of the AnalyzeSetUpRegression.Binomial function was funded by:
- Food and Drug Administration, Center for Drug Evaluation and Research, through the Mini-Sentinel Project (base version, documentation);
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (user defined alpha spending functions, improved documentation);
- National Council of Scientific and Technological Development (CNPq), Brazil, process number 302882/2022-7. (v4.3.1).
- Support Foundation to Minas Gerais State Research-Fapemig, Brazil, grant numbers PQ-00787-21 and RED-00133-21. (v3.1 to v4.3).

See also

`AnalyzeRegression.Binomial`: for running the sequential analysis that was set up using the `AnalyzeSetUpRegression.Binomial` function.

Author(s)

Ivair Ramos Silva.

References

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, *no. ISBN 0-8493-0316-8, London: Chapman and Hall/CRC.*

Kim K, DeMets DL. (1987), Design and Analysis of Group Sequential Tests Based on the Type I Error Spending Rate Function. Biometrika, **74**, n.1: 149–154.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107-118.

Silva IR, Montalban, J., Oliveira, F. (2025), Maximized Sequential Probability Ratio Test Regression. Working paper - Sentinel (TIDE) project, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Examples

```
# See example in the description of the AnalyzeRegression.Binomial function.
```

`AnalyzeSetUpRegression.CondPoisson` *Function to set up input parameters before using the* `AnalyzeRegression.CondPoisson` *function for the first time.*

Description

The function `AnalyzeSetUpRegression.CondPoisson` must be run ahead of `AnalyzeRegression.CondPoisson` in order to set up the sequential analysis before the first group of data is analyzed. The function obtains the main parameter settings and performs basic calculations that are necessary for the subsequent sequential analysis.

Usage

```
AnalyzeSetUpRegression.CondPoisson(name,N="n",cc="n",alpha=0.05,
R0=1,rho=1,mref=999,title="n",address="n")
```

Arguments

`name`	The name of the sequential analysis. Must be identical for all looks at the data, and the same as the name given in the subsequent calls to the `AnalyzeRegression.CondPoisson` function. It cannot be the same as for another sequential analysis that is run simultaneously on the same computer. There is no default.
`N`	The maximum sample size in the surveillance period, at which the sequential analysis stops without rejecting the null hypothesis.
`cc`	Number of events observed in the historical period.
`alpha`	The overall significance level. Must be in the range (0,0.5]. The default is "alpha=0.05".
`R0`	A positive real-valued number for the relative risk under R<=R0. Default is R0=1.
`rho`	The parameter rho is used to build the target alpha spending function according to a power-type function. See below for details. The variable rho must be a positive number. The default value is "rho=1".
`mref`	The parameter mref is the minimum number of Monte Carlo replications to calculate the p-value per test following the method by Silva at al(2025).
`title`	Title for the results shown in the output tables and the illustrative graphics. It can be any text string. The default is that there is no title.
`address`	The address of the directory where the settings information of this sequential analysis is saved.

Details

The function AnalyzeSetUpRegression.CondPoisson has to be executed once, but just once, to set up the general statistical characteristics of the intended sequential analysis, which is performed using the companion AnalyzeRegression.CondPoisson function.

Sequential analysis methods are devoted to analyze data sets that accrue cumulatively over time, by conducting multiple statistical tests sequentially as more data accrues. In such a setting, it is important to carefully plan the sequential analysis before the first data arrives. For example, it is important to maintain certain analysis parameter values over time to avoid counting the same data twice, and to make sure that there are no changes in the past data that has already been included in a prior test. To avoid these kinds of problems, the AnalyzeSetUpRegression.CondPoisson function is used to set the analysis parameters a priori and to create a place to save the data as it accumulates over time. At the time of each sequential test, this information is then automatically imported by the AnalyzeRegression.CondPoisson function, to ensure the correct concatenation of old and new information.

At each test, the function AnalyzeRegression.CondPoisson makes this concatenation automatically, but it will only work if the function AnalyzeSetUpRegression.CondPoisson is executed before performing the very first test.

When running AnalyzeSetUpRegression.CondPoisson, the user has the opportunity to choose the directory where the file with the general setup information and the historical data are to be saved. Important: The location of this parameter and data file is saved in the temporary directory, so that directory cannot be cleaned until the sequential analysis has been completed. Each sequential analysis needs a different identifier, which is set using the "name" parameter. Once a name is chosen, it has to be written exactly the same way when running the function AnalyzeRegression.CondPoisson.

AnalyzeSetUpRegression.CondPoisson and AnalyzeRegression.CondPoisson work for the power-type alpha spending plan (Kim and DeMetz 1987, p150; Jennison and Turnbull 2000, p148), with parameter rho: $F(t) = alpha * t^{rho}$, where $alpha$ is the overall significance level and t is a fraction of N, the maximum length of sequential analysis. According to Silva et al(2019), 'rho=1.5' is indicated for minimizing expected time to signal and expected sample size. The choice 'rho=1' is also a proper option that provides a balance between time to signal and power, hence this is the default in AnalyzeSetUpRegression.CondPoisson.

In addition to selecting the alpha spending plan, it is necessary to specify the overall alpha, or maximum Type I error probability, for the sequential analysis as a whole. It is also necessary to specify the maximum length of the sequential analysis, N, so that the sequential analysis stops without rejecting the null hypothesis when a total of N observations are obtained.

Value

inputSetUp — The AnalyzeSetUpRegression.CondPoisson function creates four data.frames (Decision_table, Relative_risk_estimates, Coefficients, Confidence_Intervals) with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests. The 'inputSetUp' list is used by AnalyzeRegression.CondPoisson, then it must be available when running AnalyzeRegression.CondPoisson, but there is no need to manually look at it.

Acknowledgements

Development of the AnalyzeSetUpRegression.CondPoisson function was funded by:
- Food and Drug Administration, Center for Drug Evaluation and Research, through the Mini-Sentinel Project (base version, documentation);
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (user defined alpha spending functions, improved documentation);
- National Council of Scientific and Technological Development (CNPq), Brazil, process number 302882/2022-7. (v4.3.1).
- Support Foundation to Minas Gerais State Research-Fapemig, Brazil, grant numbers PQ-00787-21 and RED-00133-21. (v3.1 to v4.3).

See also

`AnalyzeRegression.CondPoisson`: for running the sequential analysis that was set up using the `AnalyzeSetUpRegression.CondPoisson` function.

Author(s)

Ivair Ramos Silva.

References

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, *no. ISBN 0-8493-0316-8, London: Chapman and Hall/CRC*.

Kim K, DeMets DL. (1987), Design and Analysis of Group Sequential Tests Based on the Type I Error Spending Rate Function. Biometrika, **74**, n.1: 149–154.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR, Lopes LM, Dias P, Yih WK. (2019), Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, 38(12), 2126–2138.

Silva IR, Montalban, J., Oliveira, F. (2025), Maximized Sequential Probability Ratio Test Regression. Working paper - Sentinel (TIDE) project, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Examples

```
# See example in the description of the AnalyzeRegression.CondPoisson function.
```

AnalyzeSetUpRegression.Poisson
Function to set up input parameters before using the AnalyzeRegression.Poisson *function for the first time.*

Description

The function AnalyzeSetUpRegression.Poisson must be run ahead of AnalyzeRegression.Poisson in order to set up the sequential analysis before the first group of data is analyzed. The function obtains the main parameter settings and performs basic calculations that are necessary for the subsequent sequential analysis.

Usage

```
AnalyzeSetUpRegression.Poisson(name,N="n",alpha=0.05,
R0=1,rho=1,mref=999,title="n",address="n")
```

Arguments

name	The name of the sequential analysis. Must be identical for all looks at the data, and the same as the name given in the subsequent calls to the AnalyzeRegression.Poisson function. It cannot be the same as for another sequential analysis that is run simultaneously on the same computer. There is no default.
N	The maximum sample size, at which the sequential analysis stops without rejecting the null hypothesis.
alpha	The overall significance level. Must be in the range (0,0.5]. The default is "alpha=0.05".
R0	A positive real-valued number for the relative risk under R<=R0. Default is R0=1.
rho	The parameter rho is used to build the target alpha spending function according to a power-type function. See below for details. The variable rho must be a positive number. The default value is "rho=1".
mref	The parameter mref is the minimum number of Monte Carlo replications to calculate the p-value per test following the method by Silva at al(2025).
title	Title for the results shown in the output tables and the illustrative graphics. It can be any text string. The default is that there is no title.
address	The address of the directory where the settings information of this sequential analysis is saved.

Details

The function AnalyzeSetUpRegression.Poisson has to be executed once, but just once, to set up the general statistical characteristics of the intended sequential analysis, which is performed using the companion AnalyzeRegression.Poisson function.

Sequential analysis methods are devoted to analyze data sets that accrue cumulatively over time, by conducting multiple statistical tests sequentially as more data accrues. In such a setting, it is important to carefully plan the sequential analysis before the first data arrives. For example, it is important to maintain certain analysis parameter values over time to avoid counting the same data twice, and to make sure that there are no changes in the past data that has already been included in a prior test. To avoid these kinds of problems, the `AnalyzeSetUpRegression.Poisson` function is used to set the analysis parameters a priori and to create a place to save the data as it accumulates over time. At the time of each sequential test, this information is then automatically imported by the `AnalyzeRegression.Poisson` function, to ensure the correct concatenation of old and new information.

At each test, the function `AnalyzeRegression.Poisson` makes this concatenation automatically, but it will only work if the function `AnalyzeSetUpRegression.Poisson` is executed before performing the very first test.

When running `AnalyzeSetUpRegression.Poisson`, the user has the opportunity to choose the directory where the file with the general setup information and the historical data are to be saved. Important: The location of this parameter and data file is saved in the temporary directory, so that directory cannot be cleaned until the sequential analysis has been completed. Each sequential analysis needs a different identifier, which is set using the "name" parameter. Once a name is chosen, it has to be written exactly the same way when running the function `AnalyzeRegression.Poisson`.

`AnalyzeSetUpRegression.Poisson` and `AnalyzeRegression.Poisson` work for the power-type alpha spending plan (Kim and DeMetz 1987, p150; Jennison and Turnbull 2000, p148), with parameter rho: $F(t) = alpha * t^{rho}$, where $alpha$ is the overall significance level and t is a fraction of N, the maximum length of sequential analysis. According to Silva (2018), 'rho=0.5' is indicated when expected time to signal is the design criterion. The choice 'rho=1' is a proper option that provides a balance between time to signal and maximum lenght and surveillance, hence this is the default in `AnalyzeSetUpRegression.Poisson`.

In addition to selecting the alpha spending plan, it is necessary to specify the overall alpha, or maximum Type I error probability, for the sequential analysis as a whole. It is also necessary to specify the maximum length of the sequential analysis, N, so that the sequential analysis stops without rejecting the null hypothesis when a total of N observations are obtained.

Value

`inputSetUp` The `AnalyzeSetUpRegression.Poisson` function creates four data.frames (Decision_table, Relative_risk_estimates, Coefficients, Confidence_Intervals) with the main information concerning the tuning parameterization for the planned surveillance and the historical information about the performed tests. The 'inputSetUp' list is used by `AnalyzeRegression.Poisson`, then it must be available when running `AnalyzeRegression.Poisson`, but there is no need to manually look at it.

Acknowledgements

Development of the AnalyzeSetUpRegression.Poisson function was funded by:
- Food and Drug Administration, Center for Drug Evaluation and Research, through the Mini-Sentinel Project (base version, documentation);
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (user defined alpha spending functions, improved documentation);

- National Council of Scientific and Technological Development (CNPq), Brazil, process number 302882/2022-7. (v4.3.1).
- Support Foundation to Minas Gerais State Research-Fapemig, Brazil, grant numbers PQ-00787-21 and RED-00133-21. (v3.1 to v4.3).

See also

AnalyzeRegression.Poisson: for running the sequential analysis that was set up using the AnalyzeSetUpRegression.Poi function.

Author(s)

Ivair Ramos Silva.

References

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, *no. ISBN 0-8493-0316-8, London: Chapman and Hall/CRC*.

Kim K, DeMets DL. (1987), Design and Analysis of Group Sequential Tests Based on the Type I Error Spending Rate Function. Biometrika, **74**, n.1: 149–154.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR. (2018b), Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance With Poisson Data. Methodol Comput Appl Probab, 20(2), 739–750.

Silva IR, Montalban, J., Oliveira, F. (2025), Maximized Sequential Probability Ratio Test Regression. Working paper - Sentinel (TIDE) project, Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute.

Examples

```
# See example in the description of the AnalyzeRegression.Poisson function.
```

ConfidenceInterval.Binomial

Confidence Interval for the Relative Risk Following a Sequential Test with Binary Data.

Description

The function ConfidenceInterval.Binomial is used for constructing confidence interval for the relative risk in either continuous or group sequential analysis, or for a combination of the two, on termination of the sequential surveillance. This function is useful, for example, in combination with the Analyze.Binomial function.

Usage

```
ConfidenceInterval.Binomial(Gamma=0.9,CV.lower="n",CV.upper="n",
GroupSizes,z="n",p="n",Cum.cases,Tailed="upper")
```

Arguments

Gamma	Confidence coefficient for the interval estimator of the relative risk. The default is Gamma=0.9.
CV.lower	Signaling threshold for the evidence of "RR<R0", where R0 is a user-defined positive value when constructing this lower signaling threshold. It is given in the scale of the binomial cumulative data. Put NA for initial looks at the data when tests were not applicable. The default CV.lower="n" means that the sequential test was not designed to detect RR<R0 by the time of the confidence interval construction.
CV.upper	Signaling threshold for evidence of "RR>R0", where R0 is a user-defined positive value when constructing this upper signaling threshold. It is given in the scale of the the binomial cumulative data. Put NA for initial looks at the data when tests were not applicable. The default CV.upper="n" means that the sequential test was not designed to detect RR>R0 by the time of the confidence interval construction.
GroupSizes	Vector with the total number of events (cases+controls) between two looks at the data with regular and irregular group sizes. There is no default value.
z	For a matched case-control analysis, z is the number of controls matched to each case under the null hypothesis. If just a single number is given, then it will be used as a constant matching ratio for all groups. Otherwise, the dimension of z must coincide with the dimension of GroupSizes. The default z="n" means that the input p will be used instead.
p	The probability of having a case under the null hypothesis H0:RR<=1. If just a single number is given, then it will be used as a constant probability for all groups. Otherwise, the dimension of p must coincide with the dimension of GroupSizes. The default p="n" means that the input z will be used instead.
Cum.cases	Total number of cumulative cases on termination of the analysis. There is no default.
Tailed	Tailed="upper" (default) for H0:RR<=R0, Tailed="lower" for H0:RR>=R0 or Tailed="two" for H0:R01<=RR<=R02. Important: R0, R01, and R02 are not parameters of this function. It is supposed that they were used when the user somehow constructed CV.lower and CV.upper.

Details

For continuous and group sequential analysis with binomial data, the confidence interval for the relative risk, RR, is constructed with `ConfidenceInterval.Binomial`.

For two-tailed testing (`Tailed="two"`), both lower and upper signaling thresholds must be informed through `CV.lower` and `CV.upper`. See details in Silva et al (2021).

z is a vector of positive numbers representing the matching ratios for each test (group). If a single number is given, then it will be used as a constant matching ratio for all tests (groups). Otherwise, the dimension of z must coincide with the dimension of GroupSizes. z represents the number of controls matched to each case. For example, if there are 3 controls matched to each case, z=3. In a self-control analysis, z is the ratio of the control interval to the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, z=7/2. In terms of p, the binomial probability under the null hypothesis, p=1/(1+z), or equivalently, z=1/p-1.

Alternatively, instead of z the user can specify p directly. Note that only one of these inputs, z or p, has to be specified, but if both are entered the code will only work if z and p are such that p=1/(1+z). Otherwise, an error message will appear to remind that such condition must be complied.

With GroupSizes the user informs the sample size of each subsequent test. Therefore, only positive integers are accepted in GroupSizes.

The confidence interval is calculated by pivoting the probability of rejecting the null hypothesis with one of the thresholds CV.lower or CV.upper. This is equivalent to inverting the two-tailed testing as described by Jennison and Turnbull (2000), page 181 of Section 8.5. See also Silva and Zhuang (2022) for more details on the calculations.

Value

RRl	The lower limit of the confidence interval for RR.
RRu	The upper limit of the confidence interval for RR.

Acknowledgements

Development of the ConfidenceInterval.Binomial function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0,2.0 to 3.1).
- National Council of Scientific and Technological Development (CNPq), Brazil, process number 301391/2019-0. (v1.0).
- Research Support Foundation of the State of Minas Gerais (FAPEMIG), Brazil, grant number PQ-00787-21.
- Federal University of Ouro Preto (UFOP), through contract under internal UFOP's resolution CEPE 4600 (v2.0 to 3.1).

See also

Performance.AlphaSpend.Binomial: for calculating signaling threshold for user-specified alpha spending with binomial data.
CV.Binomial: for calculating Wald-type signaling thresholds for continuous sequential analysis with binomial data.
Analyze.Binomial: for performing sequential analysis with group, continuous or unpredictable sequential fashion with binomial data. Analyze.wBinomial: for performing sequential analysis with group, continuous or unpredictable sequential fashion with multiple weighted binary endpoints.

Author(s)

Ivair Ramos Silva and Martin Kulldorf.

References

Jennison C, Turnbull B. (2000), Group Sequential Methods with Applications to Clinical Trials, London: Chapman and Hall/CRC.

Silva IR, Zhuang, Y. (2022), Bounded-width confidence interval following optimal sequential analysis of adverse events with binary data, Statistical Methods in Medical Research, 31(12), 2323–2337.

Silva IR, Maro J, Kulldorff M. (2021), Exact sequential test for clinical trials and post-market drug and vaccine safety surveillance with Poisson and binary data. Statistics in Medicine, DOI: 10.1002/sim.9094.

Examples

```
# ConfidenceInterval.Binomial(Gamma=0.9,CV.lower=c(NA,1,1,2),
# CV.upper=c(8,13,15,18), GroupSizes=c(8,5,5,6),z=1,
# Cum.cases=18,Tailed="two")
```

`CV.Binomial` *Calculates exact critical values for group and continuous sequential analysis with binomial data.*

Description

The function `CV.Binomial` obtains critical values for the group continuous sequential MaxSPRT test with binomial data, using a Wald-type upper boundary, which is flat with respect to the likelihood ratio function, and an pre-specified upper limit on the sample size.

Usage

```
CV.Binomial(N,alpha=0.05,M=1,z="n",p="n",GroupSizes=1,Tailed="upper")
```

Arguments

`N`	The upper limit on the sample size (length of surveillance) expressed in terms of the total number of events (cases plus controls). "N" must be a positive integer. To avoid very large computation times, we suggest not using values greater than 1000. Typically, this is not a major restriction. For example, for "RR=1.1", "alpha=0.01" and "z=1", the statistical power is approximately 1 for "N>500". There is no default value.
`alpha`	The significance level. The "alpha" level must be in the range (0,0.5]. The default value is "alpha=0.05".
`M`	The minimum number of events needed before the null hypothesis can be rejected. "M" must be a positive integer, and the default value is "M=1".
`z`	For a matched case-control analysis, z is the number of controls matched to each case under the null hypothesis. There is no default value.

p	The probability of having a case under the null hypothesis. There is no default value.
GroupSizes	Vector with the number of events (cases+controls) between two consecutive looks (tests) at the data, i.e, the group sizes. The length of this vector is equal to the maximum number of tests. The entries do not have to be the same, but they must sum up "N". If the group sizes is an integer instead of a vector, then that integer is the group size for all looks at the data, and the number of looks is "N/GroupSizes". The default is GroupSizes=1 for continuous sequential analysis.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

For the continuous and group binomial MaxSPRT, CV.Binomial calculates the upper boundary used to determine if the null hypothesis is to be rejected at each analysis. This is done for pre-specified values of the statistical significance level (alpha) and an upper limit on the sample size equal to N.

The input z represents the number of controls matched to each case. For example, if there are 3 controls matched to each case, "z=3". In a self-control analysis, z is the ratio of the control interval to the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, z=7/2. In terms of p, the binomial probability under the null hypothesis, p=1/(1+z), or equivalently, z=1/p-1. The parameter z must be a positive number.

Alternatively, instead of z the user can specify p directly. Note that only one of these inputs, z or p, has to be specified, but if both are entered the code will only work if z and p are such that p=1/(1+z). Otherwise, an error message will appear to remind that such condition must be complied.

For details about the algorithm used to calculate the critical value, see the paper by Kulldorff et al. (2011).

For some configurations of N and alpha and GroupSizes, there is no critical value that gives a Type I error probability that is exactly equal to the requested "alpha". This is because of the discrete nature of binomial data. In such situations, CV.Binomial returns the greatest critical value that guarantees a Type I error probability smaller than "alpha". Thus the critical value for the binomial sequential analysis is conservative in this sense.

Value

cv	The critical value for a significance level equal to alpha. The largest conservative value is provided when it is not possible to have an Type I error exactly equal to alpha.
Type_I_Error	The exact Type I error probability given cv. Always less than or equal to alpha.

Acknowledgements

Development of the CV.Binomial function was funded by:
- Food and Drug Administration, Center for Drug Evaluation and Research, through the Mini-Sentinel Project; base version, documentation;
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999; code revisions, increased computational speed, improved documentation.

We thank Ron Berman, University of California, Berkeley, for a key suggestion to speed up the calculations, and Bruce Fireman for helpful discussions.

See also

Analyze.Binomial: for performing sequential analysis with group, continuous or unpredictable sequential fashion.

Author(s)

Ivair Ramos Silva, Ned Lewis, Ron Berman, Martin Kulldorff.

References

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Silva IR, Kulldorff M. (2015), Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Examples

```
# Example 1:
## Critical value for continuous binomial sequential analysis with
#  a maximum sample size of 20 events, requiring at
#  least 3 events to reject the null, and with a significance level of 0.05:

CV.Binomial(N=20,alpha=0.05,M=3,z=1.1)

# Example 2:
## Critical value for five-group sequential analysis with
#  a maximum sample size of 25 events, requiring at
#  least 1 event to reject the null, and with a significance level of 0.05:
result<- CV.Binomial(N=25,alpha=0.05,M=1,z=7/2,GroupSizes=5)
# if you type:
result
# then you will get the following output:
# [[1]]
# [1] 1.9852

# [[2]]
# [1] 0.04775995

# Example 3:
## Critical value for four-group sequential analysis with
#  a maximum sample size of 50 events, requiring at
#  least 1 event to reject the null, and with a significance level of 0.05:
result<- CV.Binomial(N=50,alpha=0.05,M=1,z=7/2,GroupSizes=c(10,10,15,15))
cv<- as.numeric(result[1])
# if you type:
cv
# then you will get the following output:
```

```
# [1] 1.99202
```

`CV.CondPoisson` *Critical values for continuous sequential CMaxSPRT for Poisson data with limited information from historical cohort.*

Description

The function `CV.CondPoisson` calculates critical values for the continuous sequential CMaxSPRT, using a Wald-type upper boundary, which is flat with respect to the likelihood ratio function, and a pre-specified upper limit on surveillance length.

Usage

```
CV.CondPoisson(Inference="exact", StopType="Cases",T="n",K="n",cc,
D=0,M=1,alpha=0.05,Tailed="upper")
```

Arguments

`Inference`	Inference='liberal', 'exact', or 'conservative' for the computation approach. Inference='liberal' for the liberal approach with possibly underestimated critival values and higher-than-nominal Type I error rate, Inference='exact' for the exact approach with exact critival values and nominal Type I error rates, Inference='conservative' for the conservative approach with possibly overestimated critival values and lower-than-nominal Type I error rates. The default is Inference="exact".
`StopType`	StopType='Tal' or 'Cases' for the type of surveillance length definition. With StopType='Tal', the maximum surveillance length (i.e., the upper limit) is defined in terms of the ratio of the cumulative person-time in the surveillance population divided by the total cumulative person-time in historical data, i.e., $P_k/V \leq T$; with StopType='Cases', the maximum surveillance length is defined interms of the observed number of events in the surveillance population, i.e., $k \leq K$. The default is StopType="Cases"
`cc`	The total number of observed adverse events in the historical data. There is no default value.
`K`	The upper limit on length of surveillance expressed in terms of the observed number of events in the surveillance population, i.e., $k \leq K$. This argument K is used if and only if StopType='Cases'. There is no default value.
`T`	The upper limit on length of surveillance expressed in terms of the ratio of the cumulative person-time in the surveillance population divided by the total cumulative person-time in historical data, i.e., $P_k/V \leq T$. This argument T is used if and only if StopType='Tal'. There is no default value.

D — The minium number for the ratio P_k/V before the null hypothesis can be rejected. This argument is used together with T. The default value is $D = 0$. A delayed start with $D > 0$ is recommended to avoid signaling very early on such that very little information would be available to judge whether the signal is more likely to be a true signal or chance finding.

M — The minimum number of events needed before the null hypothesis can be rejected. This argument is used together with K. A delayed start with $M > 1$ is recommended to avoid signaling very early on such that very little information would be available to judge whether the signal is more likely to be a true signal or chance finding. The default value is M=1.

alpha — The significance level, or the type 1 error probability, which is the probability of rejecting the null hypothesis when it is true. The alpha level must be in the range (0,0.5]. The default value is alpha=0.05.

Tailed — Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

For continuous sequential analysis with CMaxSPRT by Li and Kulldorff (2010) for Poisson data and limited historical data, CV.CondPoisson calculates the critical value that constitutes the upper boundary used to determine if the null hypothesis should be rejected. This is done for pre-specified values of the statistical significance level (alpha) and an upper limit which can be defined based on either the observed number of events, "K", or the ratio "T" between the cumulative person-times in the surveillance population versus the historical data, as well as other parameter settings.

The test is one-sided, so that the null hypothesis is only rejected when there are more events than expected.

Following the results of Silva et al. (2016), the function offers three computation approaches which calculate liberal, exact, and conservative critical values respectively. When the upper limit is medium (e.g., $K = 50$) or large, the computational requirements for the exact approach can be high. The recommendation is to use the exact approach when the upper limit is small (e.g., $K = 10$), use the conservative approach when the upper limit is medium ($K = 50$) or large but cc is small, and use the liberal approach when cc is medium (e.g., 50) or large. Exact numerical results show that the three approaches yield very similar results when K and cc are reasonably large.

Value

Type_I_Error — The actual Type I error, for the exact approach. It equals the nominal level specified by the argument "alpha". For the liberal approach, the actual Type I error rate may be higher than the specified nominal level. For the conservative approach, the actual Type I error rate may be lower than the specified nominal level.

cv — The critical value for a significance level. For the exact approach, it is the exact critical value, for the liberal approach, it is the smallest liberal vaue, for the conservative approach, it is the largest conservative value.

Acknowledgements

Development of the CV.CondPoisson function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0.1, v2.0.2).

See also

SampleSize.CondPoisson: calculating the upper limit with given Alpha, RR, and a desired power level for continuous CMaxSPRT.
Performance.CondPoisson: calculating the statistical power, expected time to signal and expected time of analysis for continuous CMaxSPRT.

Author(s)

Ivair Ramos Silva, Lingling Li

References

Li L, Kulldorff M. (2010). A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

Silva IR, Li L, Kulldorff M. (2019). Exact Conditional Sequential Testing for Poisson Data. Sequential Analysis, in press.

Silva IR., Lopes LM., Dias P., Yih WK. (2019). Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, DOI: 10.1002/sim.8097, 1–13.

Examples

```
# Calculates the exact critical value with upper limit of
# K=20 and a delayed start of a minimum of 2 cases, historical
# data has 20 events, and for a statistical significance level
# of 0.05.
# res<- CV.CondPoisson(Inference="exact", StopType="Cases",K=20,cc=20,
# M=2,alpha=0.05)

# which gives the results:
# res
# $Type_I_Error
# [1] 0.05
# $cv
# [1] 3.149115

# Calculates the liberal critical value with a uppe limit of
# T=0.5 and a delayed start of D=0.1, i.e., the cumulative
# person-time in the surveillance population is at least
# one-tenth of the total cumulative person-time in historical
# data, historical data has 20 events, and for a statistical
# significance level of 0.05.
# res2<- CV.CondPoisson(Inference="liberal",StopType="Tal",T=0.5,cc=20,
# D=0.1,alpha=0.05)
```

```
# which gives the results:
# res2
# $Type_I_Error
# [1] 0.05
# $cv
# [1] 2.874993
```

CV.Poisson *Critical values for group and continuous sequential analysis with Poisson data.*

Description

The function CV.Poisson obtains critical values for the group and continuous sequential MaxSPRT test with Poisson data, using a Wald type upper boundary, which is flat with respect to the likelihood ratio function, and with a pre-specified upper limit on the sample size.

Usage

```
CV.Poisson(SampleSize,D=0,M=1,alpha=0.05,GroupSizes="n",Tailed="upper")
```

Arguments

SampleSize	The upper limit on the sample size (length of surveillance) expressed in terms of the expected number of events under the null hypothesis. The SampleSize must be greater than 0. To avoid very large computation times, we suggest not using values greater than 1000. Typically, this is not a major restriction. For example, for RR=1.1 and alpha=0.01, the statistical power is approximately 1 for a maximum sample size greater than 500. There is no default value.
D	The expected number of events under the null hypothesis before the first look at the data. The default is D=0, which is also the best choice. This means that there is no delay in the start of the sequential analyses. It is required that D<=SampleSize.
M	The minimum number of events needed before the null hypothesis can be rejected. The default value is M=1, which means that even a single event can reject the null hypothesis if it occurs sufficiently early. A good rule of thumb is to set M=4 (Kulldorff and Silva, 2015).
alpha	The significance level, or the type 1 error probability, which is the probability of rejecting the null hypothesis when it is true. The alpha level must be in the range (0,0.5]. The default value is alpha=0.05.

GroupSizes	Vector containing the expected number of events under H0 for each test. The values must be positive numbers. The dimension of this vector must be equal to the maximum number of sequential tests. Thus, the sum of the entries in GroupSizes has to be equal to SampleSize. The default is GroupSizes="n" for continuous sequential analysis.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

For the group and continuous sequential analysis with Poisson data, using the maximized sequential probability ratio test (MaxSPRT), CV.Poisson calculates the upper boundary used to determine if the null hypothesis should be rejected. This is done for pre-specified values on the statistical significance level (alpha) and the upper limit on the sample size, determining the maximum length of surveillance. The algorithm used to calculate the critical value is described by Kulldorff et al. (2011).

For some configurations of SampleSize, D and alpha, there is no critical value that gives a significance level that is exactly equal to the requested alpha. In such situations, CV.Poisson returns the greatest critical value that will guarantee an alpha level less than the alpha specified, so that sequential analysis is conservative.

For large values of SampleSize, such as 200 or more, the computational requirements can be high. To speed things up, the function will sometimes use one of two look-up tables that contain pre-calculated critical values for a pre-selected set of parameter values (TableCV.PoissonD and TableCV.PoissonM).

Value

cv	The critical value for a significance level equal to alpha. The largest conservative value is provided when it is not possible to have a Type I error exactly equal to alpha.

Acknowledgements

Development of the CV.Poisson function was funded by:
- Food and Drug Administration, Center for Biologics Evaluation and Research, through the Mini-Sentinel Post-Rapid Immunization Safety Monitoring (PRISM) program (v1.0).
- National Council of Scientific and Technological Development (CNPq), Brazil (v1.0).
- Bank for Development of the Minas Gerais State (BDMG), Brazil (v1.0).
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0.1, 2.0.2).

See also

SampleSize.Poisson: for calculating the sample size needed for Continuous Sequential Analysis with Poisson Data.
Performance.Poisson: for calculating the statistical power, expected time to signal and expected time of analysis for continuous sequential analysis with Poisson data.
CV.Binomial: for calculating critical values in continuous sequential analysis with binomial data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Examples

```
## Calculates the critical value for continuous sequential analysis with
## a maximum sample size of ten expected cases under the null hypothesis,
## requiring at least 3 events to reject the null, and with a significance
## level of 0.05:

CV.Poisson(SampleSize=10,D=0,M=3,alpha=0.05)

#  Example 1:
## In this example, no critical value exist that will give the desired 0.05
## alpha level exactly. Instead, the function produces the critical value
## that makes the alpha as large as possible without exceeding 0.05.

CV.Poisson(SampleSize=3,D=1,M=1,alpha=0.05)

#  Example 2:
## Calculates the critical value for five-group sequential looks, at 5, 11,
## 17, 22 and 30 expected events under the null hypothesis, and for a
## statistical signifi-
## cance level of 0.05.

CV.Poisson(SampleSize=30,alpha=0.05,GroupSizes= c(5,6,6,5,8))
```

Optimal.Binomial — *Optimal alpha spending for minimizing expected time to signal for continuous and group sequential analysis with binomial data.*

Description

The function `Optimal.Binomial` obtains the optimal alpha spending function that minimizes the expected time to signal for target statistical power and fixed relative risk, when doing continuous or group sequential analysis for binomial data.

Usage

```
Optimal.Binomial(Objective="ETimeToSignal",N="n",
z="n",p="n",alpha,power,RR,GroupSizes="n",Tailed= "upper",
ConfIntWidth="n",ConfTimes=1,Gamma=0.9,R0=1)
```

Arguments

Objective	Statistical performance measure to minimize. Options are "ETimeToSignal", for minimizing expected time to signal, and "ESampleSize", for minimizing expected sample size. Default is "ETimeToSignal".
N	The upper limit on the sample size (length of surveillance) expressed in terms of the total number of events (cases plus controls). "N" must be a positive integer. To avoid very large computation times, we restrict the usage of values greater than 240. Typically, this is not a major restriction. For example, for "RR=2.5", "alpha=0.05" and "z=1", the statistical power is approximately 1 for "N>=150". Default is 'n', which means that the optimal 'N' will also be delivered by this function.
z	For a matched case-control analysis, z is the number of controls matched to each case under the null hypothesis. There is no default value.
p	The probability of having a case under the null hypothesis. There is no default value.
alpha	The significance level. The default value is "alpha=0.05". Must be in the range (0, 0.5].
power	The target statistical power to detect an increased risk of the relative risk (RR). There is no default value.
RR	A target relative risk to be detected with the requested statistical power.
GroupSizes	Vector with the number of events (exposed+unexposed) between two looks at the data, i.e, irregular group sizes. Important: Must sums up N. The default 'n' means continuous sequential testing.
Tailed	Defines between one-tailed and two-tailed testing. Possible entries are "lower", "upper", and "two". Default is "upper".
ConfIntWidth	Positive values for a bounded-width confidence interval for the relative risk. The default ConfIntWidth="n" means no constraint for ensuring the confidence interval width bound.
ConfTimes	Times when the bound on the confidence interval width are initiated for each entry of ConfIntWidth. Default is 1.
Gamma	Confidence coefficient. Default is 0.9.
R0	A positive real-valued number for the relative risk under H0, where R0<=1 if "Tailed=lower", R0>=1 if "Tailed=upper", or a two-dimensional vector for H0: R0_1<= R <= R0_2 if "Tailed=two". Default is 1.

Details

The function `Optimal.Binomial` elicits the optimal alpha spending for continuous and group binomial sequential testing in order to provide the desired statistical power for a user-specified relative risk RR. The alpha spending provided minimizes the expected time to signal, for Objective="ETimeToSignal", which is expected number of events when the null hypothesis is rejected, or expected length of surveillance if Objective="ESampleSize".

The sample size, N, is given in the scale of the total number of events (cases+controls). The input z represents the number of controls matched to each case. For example, if there are 3 controls matched to each case, "z=3".

In a self-control analysis, z is the ratio of the control interval to the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, z=7/2. In terms of p, the binomial probability under the null hypothesis, p=1/(1+z), or equivalently, z=1/p-1. The parameter z must be a positive number.

Alternatively, instead of z the user can specify p directly. Note that only one of these inputs, z or p, has to be specified, but if both are entered the code will only work if z and p are such that p=1/(1+z). Otherwise, an error message will appear to remind that such condition must be complied.

The optimal alpha spending solution is obtained by means of linear programing, which is possible following the exact derivations introduced by Silva and Kulldorff (2018). For the linear programing part, the code uses the function `simplex`.

`Optimal.Binomial` works for large sample sizes such as 300 in the continuous sequential fashion, but it can take very long time to run in such cases. Thus, for continuous sequential analysis, the usage is restricted for N values of at most 240. The computation time for N=240 under continuous fashion can take one day or more. But, for smaller values, like e.g. N=150, the execution time is around 2 hours. For N=120 this time reduces to something around 10 minuts, and, for N<100, `Optimal.Binomial` takes only a few seconds to run. But, processing time is much smaller for group sequential analysis. For example, take N=240. In this case, the optimum solution for two-stage (G=2) group sequential analysis takes 16 minutes to run. The cases of N values of 200 or less, for G values of 10 or less, it will take just a few seconds to run. These execution times were estimated using a regular PC(Windows 7, Intel(R) Core(TM) i7-2675QM CPU, 2.20GHz).

Value

`optimal_alpha_spending`
: The optimal cumulative alpha spending. In case of Tailed="two", there are 'optimal_alpha_spending_lower' and 'optimal_alpha_spending_upper', for the lower and upper signaling threshold, respectively.

`minTimeToSignal`
: The minimum expected time to signal under RR. This is provided only if Objective="ETimeToSignal".

`minESampleSize`
: The minimum expected time to signal under RR. This is provided only if Objective="ESampleSize".

`ETimeToSignal`
: The expected time to signal associated to the alpha spending solution irrespectively to the content of the input 'Objective' under RR.

`EsampleSize`
: The expected sample size associated to the alpha spending solution irrespectively to the content of the input 'Objective' under RR.

Power	Statistical power, obtained by usage of the optimal alpha spending with relative risk equal to RRtrue.
solved	Logical variable. It is equal to 1 if the linear programing procedure reached the optimal solution, 0 for inconclusive solution, and -1 if the method fails to find the solution.
Optimal_N	The maximum length of surveillance at which the optimal alpha spending reaches 'alpha' by a precision of 10^-6. This is returned only for N="n".

Acknowledgements

Development of the Optimal.Binomial function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999.

See also

CV.Binomial: for calculating critical values in continuous sequential analysis with binomial data.
Performance.Binomial: for calculating the statistical power, expected time to signal and expected time of analysis for continuous sequential analysis with binomial data.
SampleSize.Poisson: sample size calculation for continuous sequential analysis with Poisson data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Silva, I.R. and Kulldorff, M. and Yih, W. Katherine (2020). Optimal alpha spending for sequential analysis with binomial data. Journal of the Royal Statistical Society Series B, 82(4) p. 1141–1164.

Examples

```
#system.time(resESS<- Optimal.Binomial(Objective="ESampleSize",
# N=120,z=1,p="n",alpha=0.05,power=0.9,RR=2,GroupSizes="n",Tailed= "upper"))
```

Performance.AlphaSpend.Binomial

Calculates performance and signaling threshold for user-defined alpha spending for sequential analysis with binomial data.

Description

The function Performance.AlphaSpend.Binomial calculates power, expected time to signal, expected sample size and signaling threshold (critical values) associated to any user-specified alpha spending for continuous or group sequential analysis with binomial data. The user can select the scale for the signaling threshold among MaxSPRT, Pocock, OBrien-Fleming, or Wang-Tsiatis test statistics, all for a pre-specified upper limit on the sample size.

Usage

```
Performance.AlphaSpend.Binomial(N,alpha,AlphaSpend=1,AlphaSpendLower="n",
AlphaSpendUpper="n",z="n",p="n",GroupSizes="n",
Tailed="upper",rho=0.5,gamma="n",RR=2,Statistic=c("MaxSPRT", "Pocock",
"OBrien-Fleming", "Wang-Tsiatis"),Delta="n")
```

Arguments

N	The upper limit on the sample size (length of surveillance) expressed in terms of the total number of events (cases plus controls). There is no default value.
alpha	Overall significance level. Must be a number in the (0, 0.5] interval. There is no default value.
AlphaSpend	A vector with the cummulative Type I error probabiliy to be spent up to each test when "Tailed=lower" or "Tailed=upper". Alternatively, one can use an integer between 1 to 4. Default is 1. See Details.
AlphaSpendLower	A vector with the cummulative Type I error probabiliy regarding the lower signaling threshold to be spent up to each test when "Tailed=two". Alternatively, one can use an integer between 1 to 4. Default is "n", which means the specified "AlphaSpend" divided by 2.
AlphaSpendUpper	A vector with the cummulative Type I error probabiliy regarding the upper signaling threshold to be spent up to each test when "Tailed=two". Alternatively, one can use an integer between 1 to 4. Default is "n", which means the specified "AlphaSpend" divided by 2.
z	For a matched case-control analysis, z is the number of controls matched to each case under the null hypothesis. There is no default value.
p	The probability of having a case under the null hypothesis. If just a single number is given, then it will be used as a constant probability for all groups. Otherwise, the dimension of p must coincide with the dimension of GroupSizes. There is no default value.
GroupSizes	Vector with the total number of events (cases+controls) between two looks at the data with regular and irregular group sizes. Important: Must sums up N. For continuos sequential analysis, specify GroupSizes=1. There is no default value.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.
rho	Positive number used for the power-type alpha spending function (AlphaSpend=1) only. The default value is "rho=0.5". See Details.
gamma	Positive number used for the gamma-type alpha spending function (AlphaSpend=4) only. There is no default value. See Details.
RR	Vector of relative risks for performance calculation. There is no default value.
Statistic	The test statistic scale used for "cvs.lower" and "cvs.upper", therefore, this input has no effect if "cases.lower" and "cases.upper" are used instead. There is no default.

`Delta` Parameter needed for calculation of Wang-Tsiatis test statistic if this is the option selected in "Statistic". Must be a number in the (0, 0.5] interval. There is no default value.

Details

For continuous and group sequential analysis with binomial data, signaling thresholds for user-specified alpha spending are calculated with `Performance.AlphaSpend.Binomial`.

`N` must be a positive integer defining the maximum length of surveillance. To avoid very large computation times, we suggest not using values greater than 1000.

`AlphaSpend` is used for arbitrary cumulative type I error probability spending defined by the user. Alternatively, the user can select among one of the four classical alpha spending shapes bellow:
$F_1(t) = \alpha t^{\rho}$, where $\rho > 0$,
$F_2(t) = 2 - 2\Phi(x_\alpha\sqrt{t^{-1}})$, where $x_\alpha = \Phi^{-1}(1-\alpha/2)$,
$F_3(t) = \alpha \times log(1 + [exp1 - 1] \times t)$,
$F_4(t) = \alpha[1 - exp(-t\gamma)]/[1 - exp(-\gamma)]$ with $\gamma \in \Re$,
and t represents a fraction of the maximum length of surveillance. For more details on these alpha spending choices, see the paper by Silva et al. (2019), Section 2.6.

To select one of the four alpha spending types above, and using an integer i to indicate the type among $i = 1$, 2, 3, and 4, for $F_1(t)$, $F_2(t)$, $F_3(t)$ and $F_4(t)$, respectively, one needs to set `AlphaSpend=i`. Specifically for `AlphaSpend=1`, it is necessary to choose a `rho` value, or a `gamma` value if `AlphaSpend=4` is used.

`z` is a vector of positive numbers representing the matching ratios for each test (group). If a single number is given, then it will be used as a constant matching ratio for all tests (groups). Otherwise, the dimension of `z` must coincide with the dimension of `GroupSizes`. `z` represents the number of controls matched to each case. For example, if there are 3 controls matched to each case, `z=3`. In a self-control analysis, `z` is the ratio of the control interval to the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, `z=7/2`. In terms of `p`, the binomial probability under the null hypothesis, `p=1/(1+z)`, or equivalently, `z=1/p-1`.

Alternatively, instead of `z` the user can specify `p` directly. Note that only one of these inputs, `z` or `p`, has to be specified, but if both are entered the code will only work if `z` and `p` are such that `p=1/(1+z)`. Otherwise, an error message will appear to remind that such condition must be complied.

For details about the algorithm used to calculate the critical value, see the paper by Silva (2018).

With `GroupSizes` the user informs the sample size of each subsequent test. Therefore, only positive integers are accepted in `GroupSizes`.

The input `Statistic` specifies the scale selected by the user to inform `cvs.lower` and `cvs.upper`among the classic methods: MaxSPRT (Kulldorf et al., 2011), Pocock (Pocock, 1977), OBrien-Fleming (O'Brien and Fleming, 1979), or Wang-Tsiatis (Jennison and Turnbull, 2000). For `Statistic="Wang-Tsiatis"`, the user has to choose a number in the (0, 0.5] interval for `Delta`.

For `RR` the user must specify the target relative risks for calculation of the statistical performance measures to be delivered in the output. It can be a vector of positive number or a single number.

Value

`CV` Signaling threshold in the scale of the selected `Statistic` for the user-defined alpha spending.

`CV.cases`	Signaling threshold in the scale of the binomial data for the user-defined alpha spending.
`ActualSpend`	The actual Type I error probability with the calculated threshold.
`Performance`	A matrix with the following three performance measures for each target RR: statistical power, expected time to signal and expected sample size.

Acknowledgements

Development of the Performance.AlphaSpend.Binomial function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0,2.0 to 3.1).
- Federal University of Ouro Preto (UFOP), through contract under internal UFOP's resolution CEPE 4600 (v2.0 to 3.1).

See also

`Performance.Threshold.Binomial`: for calculating performance and alpha spending for user-specified signaling threshold with binomial data.
`CV.Binomial`: for calculating Wald-type signaling thresholds for continuous sequential analysis with binomial data.
`Analyze.Binomial`: for performing sequential analysis with group, continuous or unpredictable sequential fashion with binomial data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials, London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

O'Brien PC, Fleming TR. (1979). A multiple testing procedure for clinical trials. Biometrics. 35:549–556.

Pocock SJ. (1977).Group sequential methods in the design and analysis of clinical trials. Biometrika. 64:191–199.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107-118.

Silva IR, Kulldorff M. (2015). Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR, Maro J, Kulldorff M. (2019). Exact Sequential Analysis Using R Sequential. Working Paper.

Examples

```
## Performance and threshold for group binomial sequential analysis
#  with a maximum sample size of 50 events for upper-tailed testing
#  that is, H0:RR<=1, with irregular group sizes 15, 15, 10, and 10.
#  This is done for alpha spending of the power-type.
#  The statistical performance is evaluated for RR= 2:

# res<- Performance.AlphaSpend.Binomial(N=50,alpha=0.05,AlphaSpend=1,z=c(1,1.5,2,1.3),
# p="n",GroupSizes=c(15,15,10,10),Tailed="upper",rho=0.5, RR=2,Statistic="MaxSPRT")
```

Performance.AlphaSpend.CondPoisson

Calculates performance and signaling thresholds for user-defined alpha spending for sequential analysis with conditional Poisson data.

Description

The function `Performance.AlphaSpend.CondPoisson` calculates power, expected time to signal and signaling threshold (critical values) associated to any user-specified alpha spending, flat or non-flat, for continuous or group sequential analysis with conditional Poisson data, all for a pre-specified upper limit on the sample size.

Usage

```
Performance.AlphaSpend.CondPoisson(K,cc,alpha,AlphaSpend="n",
GroupSizes="n",rho=0.5,gamma="n",Tailed="upper",RR)
```

Arguments

`K`	The upper limit on the sample size (length of surveillance) expressed in terms of the number of events arriving in the surveillance period. There is no default value.
`cc`	Number of events observed in the historical period. There is no default value.
`alpha`	The overall significance level.
`AlphaSpend`	A vector with the cummulative Type I error probabiliy to be spent up to each test. Alternatively, one can use an integer between 1 to 4. There is no default value. See Details.
`GroupSizes`	Test-specific number of events between two looks at the data for group or continuous sequential analysis. There is no default value. See Details.
`rho`	Positive number used for the power-type alpha spending function (`AlphaSpend=1`) only. The default value is "rho=0.5". See Details.
`gamma`	Positive number used for the gamma-type alpha spending function (`AlphaSpend=4`) only. There is no default value. See Details.

Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.
RR	Vector of relative risks for performance calculation. There is no default value.

Details

For continuous and group sequential analysis based on monitoring Poisson data conditioned on matched historical Poisson data, the threshold impplied by user-specified alpha spending is calculated with Performance.AlphaSpend.CondPoisson. The function delivers the threshold in two scales, the the Conditional Maximized Sequential Probability Ratio Test statistic (CMaxSPRT) scale (Li and Kulldorff, 2010), and the surveillance versus historical person-time ratio (Silva et al., 2019a). For the CMaxSPRT scale, the null hypothesis is rejected if the test statistic exceeds the critical value at some test. Regarding the person-time ratio scale, using the notation by Silva et al. (2019a) and Silva et al. (2019b), let V denote the total person-time from the historical data, where cc events were observed, and let $P_{k(i)}$ denote the the cummulative person-time from the surveillance data at the i-th test with a cummulative $k(i)$ events. Suppose that for a three-group sequential plan, with sample sizes of 20, 15, 25, the critival values were 0.1, 0.5, and 1. This way, H0 is rejected if: P_20/V <= 0.1 in the first test, or P_35/V <= 0.5 in the second test, or P_60/V <= 1 in the third test.

AlphaSpend is used for arbitrary cumulative type I error probability spending defined by the user. Alternatively, the user can select among one of the four classical alpha spending shapes bellow:
$F_1(t) = \alpha t^\rho$, where $\rho > 0$,
$F_2(t) = 2 - 2\Phi(x_\alpha\sqrt{t^{-1}})$, where $x_\alpha = \Phi^{-1}(1-\alpha/2)$,
$F_3(t) = \alpha \times log(1 + [exp1 - 1] \times t)$,
$F_4(t) = \alpha[1 - exp(-t\gamma)]/[1 - exp(-\gamma)]$ with $\gamma \in \Re$,
and t represents a fraction of the maximum length of surveillance. For more details on these alpha spending choices, see the paper by Silva et al. (2019c), Section 2.6.

To select one of the four alpha spending types above, and using an integer i to indicate the type among $i = 1$, 2, 3, and 4, for $F_1(t)$, $F_2(t)$, $F_3(t)$ and $F_4(t)$, respectively, one needs to set AlphaSpend=i. Specifically for AlphaSpend=1, it is necessary to choose a rho value, or a gamma value if AlphaSpend=4 is used.

With GroupSizes the user informs the sample size of each test in the scale of the number of events in the surveillance period. Therefore, only positive numbers are accepted in GroupSizes. For irregular group sizes, a vector must be informed with each test-specific number of events between two looks at the data, therefore the entries of GroupSizes must sums up K. For regular group sizes, a single number can be informed for the constant sample size of each test. For example, for continuous sequential analysis, GroupSizes=1. For ten-group sequential analysis with K=50, GroupSizes=5.

For RR the user must specify the target relative risks for calculation of statistical performance measures. It can be a vector of positive numbers or a single number.

For details on the calculation of signaling thresholds and alpha spending for Poisson data conditioned to historical data, see the papers by Silva et al. (2019a) and Silva et al. (2019b), respectively.

Value

CV	Signaling threshold in the scale of CMaxSPRT associated to the user-specified alpha spending.

`Person-timeRatioH0`
: Signaling threshold in the P_k/V scale.

`Performance`
: A matrix with the following three performance measures for each target RR: statistical power, expected time to signal and expected sample size.

Acknowledgements

Development of the Performance.AlphaSpend.CondPoisson function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0,2.0 to 3.1).
- Federal University of Ouro Preto (UFOP), through contract under internal UFOP's resolution CEPE 4600 (v2.0 to 3.1).

See also

`Performance.Threshold.CondPoisson`: for calculating performance and alpha spending for user-specified signaling threshold with conditional Poisson data.
`CV.CondPoisson`: for calculating Wald-type signaling thresholds for continuous sequential analysis with conditional Poisson data based on the CMaxSPRT test statistic.
`Analyze.CondPoisson`: for performing sequential analysis with group, continuous or unpredictable sequential fashion for condicional Poisson data based on the CMaxSPRT test statistic.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials, London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Li L, Kulldorff M. (2010). A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

O'Brien PC, Fleming TR. (1979). A multiple testing procedure for clinical trials. Biometrics. 35:549–556.

Pocock SJ. (1977). Group sequential methods in the design and analysis of clinical trials. Biometrika. 64:191–199.

Silva IR, Li L, Kulldorff M. (2019a), Exact conditional maximized sequential probability ratio test adjusted for covariates. Sequential Analysis, 38(1), 115–133.

Silva IR., Lopes LM., Dias P., Yih WK. (2019b). Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, 38(12), 2126–2138.

Silva IR, Kulldorff M. (2015). Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR, Maro J, Kulldorff M. (2019c). Exact Sequential Analysis Using R Sequential. Working Paper.

Examples

```
#### Example 1
## Performance and threshold for three group CMaxSPRT sequential with
#  a maximum sample size of 30 events for upper-tailed
#  testing, i.e. H0:RR<=1, with regular groups of sizes 10
#  and alpha spending of 0.01, 0.02, and 0.05 for tests
#  1, 2 and 3, respectively.
#  The statistical performance is evaluated for three
#  target RR= 1.2, 1.5, 2:

# res<- Performance.AlphaSpend.CondPoisson(K=30,cc=10,alpha=0.05,
# AlphaSpend=c(0.01,0.02,0.05),GroupSizes=c(10,10,10),RR=c(1.2,1.5,2))

#### Example 2
## Performance and threshold for three group CMaxSPRT with
#  a maximum sample size of 30 events for upper-tailed
#  testing, i.e. H0:RR<=1, with regular groups of sizes 10
#  and alpha spending of the power-type.
#  The statistical performance is evaluated for three
#  target RR= 1.2, 1.5, 2:

# res<- Performance.AlphaSpend.CondPoisson(K=30,cc=10,alpha=0.05,
# AlphaSpend=1,GroupSizes=c(10,10,10),rho=0.75,RR=c(1.2,1.5,2))
```

`Performance.AlphaSpend.Poisson`

Calculates performance and signaling thresholds for user-defined alpha spending for sequential analysis with Poisson data.

Description

The function `Performance.AlphaSpend.Poisson` calculates power, expected time to signal, expected sample size and the signaling threshold (critical values) associated to any user-specified alpha spending for continuous sequential analysis with Poisson data, or for a pre-specified upper limit on the sample size.

Usage

```
Performance.AlphaSpend.Poisson(SampleSize,alpha=0.05,D=0,M=1,RR,alphaSpend=1,rho=0.5,
R0=1,gamma="n",Statistic=c("MaxSPRT", "Pocock", "OBrien-Fleming", "Wang-Tsiatis"),
Delta="n",Tailed="upper")
```

Arguments

`SampleSize`	The upper limit on the sample size (length of surveillance) expressed in terms of the expected number of events under the null hypothesis. There is no default value.

alpha	The overall significance level.
D	The expected number of events under the null hypothesis at the first look at the data. This is used when there is an initial large chunk of data arriving, followed by continuous sequential analysis. The default value is D=0, which is also the best choice. This means that there is no delay in the start of the sequential analyses. If D is very large, the maximum sample size will be set equal to D if a non-sequential analysis provides the desired power.
M	The minimum number of events needed before the null hypothesis can be rejected. It must be a positive integer. A good rule of thumb is to set M=4 (Kulldorff and Silva, 2015). The default value is M=1, which means that even a single event can reject the null hypothesis if it occurs sufficiently early.
RR	Vector of relative risks for performance calculation. There is no default value.
alphaSpend	A vector with the cummulative Type I error probabiliy to be spent up to each test. Alternatively, one can use an integer between 1 to 4. Default is 1. See Details.
rho	Positive number used for the power-type alpha spending function (AlphaSpend=1) only. The default value is "rho=0.5". See Details.
R0	A positive real-valued number for the relative risk under H0, where R<=R0 if "Tailed=lower", R>=R0 if "Tailed=upper", or a two-dimensional vector for H0: R0_1<= R <= R0_2 if "Tailed=two". Default is 1.
gamma	Positive number used for the gamma-type alpha spending function (AlphaSpend=4) only. There is no default value. See Details.
Statistic	The test statistic scale to deliver the signaling threshold. See Details.
Delta	Parameter needed for calculation of Wang-Tsiatis test statistic if this is the option selected in "Statistic". Must be a number in the (0, 0.5] interval. There is no default value.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

For continuous and group sequential analysis based on monitoring Poisson data, the threshold implied by user-specified alpha spending is calculated with Performance.AlphaSpend.Poisson. The function delivers the threshold in the scale of a test statistic selected by the user with the input Statistic among the classic methods: MaxSPRT (Kulldorf et al., 2011), Pocock (Pocock, 1977), OBrien-Fleming (O'Brien and Fleming, 1979), or Wang-Tsiatis (Jennison and Turnbull, 2000). For Statistic="Wang-Tsiatis", the user has to choose a number in the (0, 0.5] interval for Delta.

alphaSpend is used for arbitrary cumulative type I error probability spending defined by the user. Alternatively, the user can select among one of the four classical alpha spending shapes bellow:
$F_1(t) = \alpha t^{\rho}$, where $\rho > 0$,
$F_2(t) = 2 - 2\Phi(x_\alpha\sqrt{t^{-1}})$, where $x_\alpha = \Phi^{-1}(1-\alpha/2)$,
$F_3(t) = \alpha \times log(1 + [exp1 - 1] \times t)$,
$F_4(t) = \alpha[1 - exp(-t\gamma)]/[1 - exp(-\gamma)]$ with $\gamma \in \Re$,
and t represents a fraction of the maximum length of surveillance.

To select one of the four alpha spending types above, and using an integer i to indicate the type among $i = 1, 2, 3$, and 4, for $F_1(t)$, $F_2(t)$, $F_3(t)$ and $F_4(t)$, respectively, one needs to set

alphaSpend=i. Specifically for alphaSpend=1, it is necessary to choose a rho value, or a gamma value if alphaSpend=4 is used.

For more details on these alpha spending choices, see the paper by Silva et al. (2021), Section 2.7.

For RR the user must specify the target relative risks for calculation of statistical performance measures. It can be a vector of positive numbers or a single number.

For details on the calculation of signaling thresholds and alpha spending for Poisson data, see the paper by Silva et al. (2018).

Value

cvs	Signaling threshold in the scale of the Statistic selected for the user-specified alpha spending.
Performance	A matrix with the following three performance measures for each target RR: statistical power, expected time to signal and expected sample size.

Acknowledgements

Development of the Performance.AlphaSpend.Poisson function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0,2.0 to 3.1).
- Federal University of Ouro Preto (UFOP), through contract under internal UFOP's resolution CEPE 4600 (v2.0 to 3.1).

See also

Performance.Threshold.Poisson: for calculating performance and alpha spending for user-specified signaling threshold with Poisson data.
CV.Poisson: for calculating Wald-type signaling thresholds for continuous sequential analysis with Poisson data based on the CMaxSPRT test statistic.
Analyze.Poisson: for performing sequential analysis with group, continuous or unpredictable sequential fashion for Poisson data based on the CMaxSPRT test statistic.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials, London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Li L, Kulldorff M. (2010). A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

O'Brien PC, Fleming TR. (1979). A multiple testing procedure for clinical trials. Biometrics. 35:549–556.

Pocock SJ. (1977). Group sequential methods in the design and analysis of clinical trials. Biometrika. 64:191–199.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance With Poisson Data. Methodol Comput Appl Probab, 20(2), 739–750.

Silva IR, Kulldorff M. (2015). Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR, Maro J, Kulldorff M. (2021). Exact sequential test for clinical trials and post-market drug and vaccine safety surveillance with Poisson and binary data. Statistics in Medicine, DOI: 10.1002/sim.9094.

Examples

```
#### Example
## Perfofmance and threshold for continuous sequential analysis
#  with a maximum sample size of 30 events for upper-tailed
#  testing, i.e. H0:RR<=1, with alpha spending of the
#  power-type and threshold delivered in the MaxSPRT
#  test statistic scale.
#  The statistical performance is evaluated for three
#  target RR= 1.5:

# res<- Performance.AlphaSpend.Poisson(SampleSize=30, alpha=0.05,alphaSpend=1,
# RR=1.5,rho=0.5,gamma="n",Statistic="MaxSPRT")
```

`Performance.Binomial` *Statistical power, expected time to signal and expected sample size for group and continuous sequential analysis with binomial data.*

Description

The function `Performance.Binomial` calculates several performance metrics for group and continuous binomial MaxSPRT for fixed upper limit on the sample size ("N"), minimum number of events required before rejecting the null hypothesis ("M"), critical value ("cv") and a relative risk ("RR"). The metrics calculated are the statistical power, the expected time to signal when the null hypothesis is rejected, and the expected sample size at the end of the analysis whether the null hypothesis was rejected or not.

Usage

```
Performance.Binomial(N,M=1,cv,z="n",p="n",RR=2,GroupSizes=1,Tailed="upper")
```

Arguments

`N`	The upper limit on the sample size (length of surveillance) expressed in terms of the total number of events. "N" must be greater than 0. To avoid very large computation times, we suggest not using values greater than 1000. Typically,

this is not a major restriction. For example, for "RR=1.1" and "alpha=0.01" and "z=1", the statistical power is approximately 1 for "N>500". There is no default value.

`M` The minimum number of events needed before the null hypothesis can be rejected. The default value is 'M=1', which means that even a single event can reject the null hypothesis if it occurs sufficiently early. A good rule of thumb is to set 'M=4' (Kulldorff and Silva, 2015). It must be a positive integer.

`cv` Critical value, defining the upper rejection boundary. The null hypothesis is rejected when the log-likelihood value is greater than "cv". The "cv" parameter is usually obtained by first running `CV.Binomial`. It must be a positive number. The default is GroupSizes=1 for continuous sequential analysis.

`z` For a matched case-control analysis, z is the number of controls matched to each case under the null hypothesis. There is no default value.

`p` The probability of having a case under the null hypothesis. There is no default value.

`RR` The relative risk (>=1) for which statistical power, expected signal time and expected length of surveillance are calculated. The default is RR=2.

`GroupSizes` Vector with the number of events (cases+controls) between two consecutive looks (tests) at the data, i.e, the group sizes. The length of this vector is equal to the maximum number of looks. The entries do not have the same, but they sum up to N. If the group sizes is an integer instead of a vector, then that integer is the group size for all looks at the data, and the number of looks is "N/GroupSizes".

`Tailed` Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

For group and continuous Binomial MaxSPRT, the function `Performance.Binomial` calculates the statistical power, the expected time to signal when the null hypothesis is rejected, and the expected sample size until the analysis ends whether the null is rejected or not. When the null hypothesis is true, the probability of having a case, instead of a control, is $p = 1/(1 + z)$. But, if the null hypothesis is false, and the true relative risk is a value 'RR>1', then the probability of having a case is $p = RR/(RR + z)$. If the user wants to calculate the exact Type I error probability for a given "cv", that can be done by setting "RR=1", in which case the power output value is the exact size of the test.

The input z represents the number of controls matched to each case. For example, if there are 3 controls matched to each case, "z=3". In a self-control analysis, z is the ratio of the control interval to the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, z=7/2. In terms of p, the binomial probability under the null hypothesis, p=1/(1+z), or equivalently, z=1/p-1. The parameter z must be a positive number.

Alternatively, instead of z the user can specify p directly. Note that only one of these inputs, z or p, has to be specified, but if both are entered the code will only work if z and p are such that p=1/(1+z). Otherwise, an error message will appear to remind that such condition must be complied.

Value

`Power` The statistical power.

ESignaltime	The expected time to signal given that the null hypothesis is rejected.
EsampleSize	The expected sample size when the analysis ends (length of surveillance) whether the null hypothesis was rejected or not.

Acknowledgements

Development of the Performance.Binomial function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999.

See also

CV.Binomial: for calculating critical values in continuous sequential analysis with binomial data.
SampleSize.Binomial: for calculating the minimum sample size given a target power in continuous sequential analysis with binomial data.

Author(s)

Ivair Ramos Silva and Martin Kulldorff.

References

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Examples

```
# Example 1:
# Performance of a continuous MaxSPRT sequential analysis
result<- Performance.Binomial(N=30,M=1,cv=2,z=1,RR=2)
# if you type:
result
# then you will get the following output:
# $power
# [1] 0.658732

# $signaltime
# [1] 10.7893

# $surveillancetime
# [1] 17.3453

# Example 2:
# Performance of a 20-group MaxSPRT sequential analysis
result<- Performance.Binomial(N=40,M=1,cv=2.5,z=1,RR=2,GroupSizes=2)
# if you type:
result
# then you will get the following output:
# $Power
```

```
# [1] 0.6594118

# $`ESignalTime`
# [1] 17.18626

# $`ESampleSize`
# [1] 24.95635
```

`Performance.CondPoisson`

Statistical power, expected time to signal and expected sample size for the continuous sequential CMaxSPRT for Poisson data with limited information from historical cohort.

Description

The function `Performance.CondPoisson` calculates several performance metrics for the continuous CMaxSPRT for selected computation approach, the type of upper limit definition and its value, critical value, number of historical data events, criteria of delayed start, the Type I error rate, and a relative risk. The metrics calculated are the statistical power, the expected time to signal when the null hypothesis is rejected, and the expected sample size at the end of the analysis whether the null hypothesis was rejected or not.

Usage

```
Performance.CondPoisson(Inference="exact",cv,StopType="Cases",
T="n",K="n",cc,D=0,M=1,RR=2,Tailed="upper")
```

Arguments

`Inference`	Inference='liberal', 'exact', or 'conservative' for the computation approach. Inference='liberal' for the liberal approach with possibly underestimated critival values and higher-than-nominal Type I error rate, Inference='exact' for the exact approach with exact critival values and nominal Type I error rates, Inference='conservative' for the conservative approach with possibly overestimated critival values and lower-than-nominal Type I error rates. The default is Inference='exact'.
`cv`	Critical value, defining the upper rejection boundary for the log-likelihood ratio test statistic. The null hypothesis is rejected when the log-likelihood value is greater than 'cv'. The 'cv' parameter is usually obtained by first running `CV.CondPoisson`. It must be a positive number,and there is no default.
`StopType`	StopType='Tal' or 'Cases' for the type of surveillance length definition. The default value is 'StopType=Cases'. See details.
`T`	The upper limit on length of surveillance expressed in terms of the ratio of the cumulative person-time in the surveillance population divided by the total cumulative person-time in historical data, i.e., $P_k/V \leq T$. This argument T is used if and only if StopType='Tal'. There is no default value.

K	The upper limit on length of surveillance expressed in terms of the observed number of events in the surveillance population, i.e., $k \leq K$. This argument K is used if and only if StopType='Cases'. There is no default value.
cc	The total number of observed adverse events in the historical data. There is no default value.
D	The minium number for the ratio P_k/V before the null hypothesis can be rejected. This argument is used together with StopType='Tal'. The default value is $D = 0$.
M	The minimum number of events needed before the null hypothesis can be rejected. This argument is used together with StopType='Cases'. The default value is M=1.
RR	The target relative risk for which statistical power, expected signal time and expected length of surveillance are calculated. The default is 'RR=2'.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

For continuous sequential analysis with Poisson data with limited historical information, (Li and Kulldorff, 2010), the Performance.CondPoisson function calculates the statistical power, the expected time to signal when the null hypothesis is rejected and the expected sample size until the analysis ends whether the null is rejected or not. The sample size (i.e., the upper limit) can be expressed either in terms of the ratio "T" of the cumulative person-time in the surveillance population divided by the total cumulative person-time in historical data (StopType="Tal"), i.e., $P_k/V \leq T$, or in terms of the observed number of events "K" in the surveillance population (StopType="Cases"), i.e., $k \leq K$. Large values of the SampleSize, greater than say 1000, may leads to long computing times. When the statistical power is close to 1, then the expected time to signal will be very close to the expected sample size.

For the parameter of delayed start, "D", a delayed start with $D > 0$ is recommended to avoid signaling very early on such that very little information would be available to judge whether the signal is more likely to be a true signal or chance finding. Similarly, if the delayed start is defined in terms of the number of events, "M", a setting such that $M > 1$ is recommended to avoid signaling very early on such that very little information would be available to judge whether the signal is more likely to be a true signal or chance finding.

Value

Power	The statistical power.
ESignalTime	The expected time to signal given that the null hypothesis is rejected.
ESampleSize	The expected sample size when the sequential analysis ends (length of surveillance) whether the null hypothesis was rejected or not.

Acknowledgements

Development of the Performance.CondPoisson function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0.1, v2.0.2). - Foundation for Research Support of Minas Gerais State (FAPEMIG), MG, Brazil, through the grant Demanda Universal.

See also

CV.CondPoisson: calculating the critical value for continuous CMaxSPRT.
SampleSize.CondPoisson: calculating the sample size for continuous CMaxSPRT.

Author(s)

Ivair Ramos Silva, Lingling Li

References

Li L, Kulldorff M. (2010). A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

Silva IR, Li L, Kulldorff M. (2019). Exact conditional maximized sequential probability ratio test adjusting for covariates. Sequential Analysis, in press.

Silva IR., Lopes LM., Dias P., Yih WK. (2019). Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, DOI: 10.1002/sim.8097, 1–13.

Examples

```
## First calculate the critical value with upper limit defined in terms of
## the number of observed events in surveillance population (K=50), with 50
## events in historical data, no delayed start, and alpha=0.05:
# res<-CV.CondPoisson(Inference="exact",StopType="Cases",K=20,cc=50,M=1,
# alpha=0.05)

# cvt<- res[[2]]

## calculate the performance using the critical value 'cvt' from the previous
## step, under RR=1.5:
#Performance.CondPoisson(Inference="exact",cv=cvt,StopType="Cases",K=20,cc=50,
# M=1,RR=1.5)
```

`Performance.Poisson` *Calculates statistical power, expected time to signal and expected sample size for group and continuous sequential analysis with Poisson data.*

Description

The `Performance.Poisson` function calculates three different performance metrics for group and continuous sequential analysis with Poisson data: the statistical power, the expected time to signal when the null hypothesis is rejected and the expected sample size at the end of the analysis whether the null hypothesis was rejected or not. The user specifies the relative risk under the alternative hypothesis (RR), as well as the sequential analysis parameters. To calculate the statistical significance level alpha, RR=1, in which case the power output value is the alpha level.

Usage

```
Performance.Poisson(SampleSize, D = 0, M = 1, cv, RR = 2,GroupSizes="n",Tailed="upper")
```

Arguments

SampleSize	The upper limit on the sample size (length of surveillance) expressed in terms of the expected number of events under the null hypothesis. The SampleSize must be greater than 0. There is no default value.
D	The expected number of events under the null hypothesis at the first look at the data. The default is D=0, which is also the best choice. This means that there is no delay in the start of the sequential analysis. It is required that D$\leq$SampleSize.
M	The minimum number of events needed to be observed before the null hypothesis can be rejected. The default is M=1, which means that even a single event can reject the null hypothesis if it occurs sufficiently early. A good rule of thumb is to set M=4 (Kulldorff and Silva, 2015).
cv	The critical value constituting the upper rejection boundary. This can be calculated using the CV.Poisson function.
RR	The relative risk under the alternative hypothesis. It is required that RR$\geq$ 1. The default value is RR=2.
GroupSizes	Vector containing the expected number of events under H0 for each test. The values must be positive numbers. The dimension of this vector must be equal to the maximum number of sequential tests. Thus, the sum of the entries in GroupSizes has to be equal to SampleSize. The default is GroupSizes="n" for continuous sequential analysis.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

For group and continuous sequential analysis with Poisson data, the Performance.Poisson function calculates the statistical power, the expected time to signal when the null hypothesis is rejected and the expected sample size until the analysis ends whether the null is rejected or not. The sample size is expressed in terms of the expected number of events under the null hypothesis. Large values of the SampleSize, greater than say 1000, may leads to long computing times. When the statistical power is close to 1, then the expected time to signal will be very close to the expected sample size, since both are measured in information time, using the expected events under the null hypothesis as the unit.

To avoid very large computation times, we suggest not using values greater than 1000. Typically, this is not a major restriction. For example, for RR=1.1 and alpha=0.01, the statistical power is approximately 1 for a maximum sample size greater than 500.

Value

Power	The statistical power.
ESignalTime	The expected time to signal given that the null hypothesis is rejected.
ESampleSize	The expected sample size when the sequential analysis ends (length of surveillance) whether the null hypothesis was rejected or not.

Acknowledgements

Development of the Performance.Poisson function was funded by:
- Food and Drug Administration, Center for Biologics Evaluation and Research, through the Mini-Sentinel Post-Rapid Immunization Safety Monitoring (PRISM) program (v1.0).
- National Council of Scientific and Technological Development (CNPq), Brazil (v1.0).
- Bank for Development of the Minas Gerais State (BDMG), Brazil (v1.0).
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0.1,2.0.2).

See also

CV.Poisson: Calculates critical values for continuous sequential analysis with Poisson data.
SampleSize.Poisson: Sample size calculations for continuous sequential analysis with Poisson data.

Author(s)

Ivair Ramos Silva and Martin Kulldorff

References

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Vaccine Safety Surveillance. Sequential Analysis, 30: 58–78. Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Examples

```
# Example 1:
## Suppose we want to find the statistical power to detect a relative risk
## of 2 when doing up to at most 20 months of surveillance, as well as the
## expected time to signal when the null hypothesis is rejected. During
## each month, we expected to see 0.5 events if the null hypothesis is true.
## This means that the upper limit on the sample size is 20*0.5=10 expected
## events under the null hypothesis. We will then first calculate the critical
## value for an upper limit on the sample size equal to 10 and a significance
## level of alpha=0.05:

# cvt<- CV.Poisson(SampleSize=10,alpha=0.05)
# cvt
# [1] 3.467952

## After that, we use the Performance.Poisson function to calculate the
## power and the expected time to signal when the null hypothesis is
## rejected for the alternative hypothesis with a relative risk equal to 2:

## Power, expected signal time and expected sample size for a relative risk
## equal to 2:
# Performance.Poisson(SampleSize=10,cv=cvt,RR=2)
#         Power    ESignalTime    ESampleSize
# [1,] 0.6850634     4.130985      5.979353
```

```
## From the results, we see that the statistical power is 68.5%. When the null
## is rejected, the expected time to signal is 4.13 in the unit of events
## expected under the null. If data is collected uniformly over time at the
## rate of 0.5 expected counts per month,the expected time to signal is
## 4.13/0.5= 8.26 months.

## The above calculations can also be accomplished using one single command line:

#  Performance.Poisson(SampleSize=10,cv=CV.Poisson(SampleSize=10,alpha=0.05),RR=2)
#         Power    ESignalTime    ESampleSize
#  [1,] 0.6850634     4.130985      5.979353

# Example 2:
## First use the CV.Poisson function to calculate the critical value for
## 5 sequential looks  at the data, spaced six units apart, and with a
## statistical significance level of 0.05:

cvt<- CV.Poisson(SampleSize=30,alpha=0.05,GroupSizes=c(6,6,6,6,6))

## For an alternative hypothesis of a relative risk of RR=1.5, calculates the
## statistical power, the expected time to signal, and the expected sample size
## at the end of the sequential analysis.

(Performance.Poisson(SampleSize=30,cv=cvt,GroupSizes=c(6,6,6,6,6),RR=1.5))
```

`Performance.Threshold.Binomial`

Statistical Performance and Alpha Spending For User-defined Signaling Threshold With Binomial Data.

Description

The function `Performance.Threshold.Binomial` calculates power, expected time to signal, expected sample size and alpha spending associated to any user-specified signaling threshold, flat or non-flat, for continuous or group sequential analysis with binomial data. The user can select the scale for the signaling threshold among MaxSPRT, Pocock, OBrien-Fleming, or Wang-Tsiatis test statistics. Alternatively, the threshold can be informed also in the scale of the binomial data.

Usage

```
Performance.Threshold.Binomial(N,CV.lower="n",CV.upper="n",z="n",p="n",
GroupSizes="n",Tailed="upper",Statistic=c("MaxSPRT", "Pocock",
"OBrien-Fleming", "Wang-Tsiatis","Cases"),Delta="n",RR)
```

Arguments

`N`	The upper limit on the sample size (length of surveillance) expressed in terms of the total number of events (cases plus controls). There is no default value.

CV.lower	Signaling threshold for evidence of "RR<1". It is given in the scale of the selected test statistic infomed in "Statistic". There is no default value.
CV.upper	Signaling threshold for evidence of "RR>1". It is given in the scale of the selected test statistic infomed in "Statistic". There is no default value.
z	For a matched case-control analysis, z is the number of controls matched to each case under the null hypothesis. There is no default value.
p	The probability of having a case under the null hypothesis. If just a single number is given, then it will be used as a constant probability for all groups. Otherwise, the dimension of p must coincide with the dimension of GroupSizes. There is no default value.
GroupSizes	Vector with the total number of events (cases+controls) between two looks at the data with regular and irregular group sizes. Important: Must sums up N. For continuos sequential analysis, specify GroupSizes=1. There is no default value.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.
Statistic	The test statistic scale used for "CV.lower" and "CV.upper". There is no default.
Delta	Parameter needed when "Statistic=Wang-Tsiatis" is selected. Must be a number in the (0, 0.5] interval. There is no default value.
RR	Vector of relative risks for performance calculation. There is no default value.

Details

For continuous and group sequential analysis with binomial data, alpha spending for user-specified thresholds are calculated with Performance.Threshold.Binomial.

N must be a positive integer defining the maximum length of surveillance. To avoid very large computation times, we suggest not using values greater than 1000.

For two-tailed testing (Tailed="two"), both lower and upper signaling thresholds must be informed through CV.lower and CV.upper. If the user desires a constant threshold (critical value) in the scale of a test statistic, then a single number can be informed. For time-variable (non-constant) thresholds, the length of CV.upper and CV.lower must coincide with the length of GroupSizes.

z is a vector of positive numbers representing the matching ratios for each test (group). If a single number is given, then it will be used as a constant matching ratio for all tests (groups). Otherwise, the dimension of z must coincide with the dimension of GroupSizes. z represents the number of controls matched to each case. For example, if there are 3 controls matched to each case, z=3. In a self-control analysis, z is the ratio of the control interval to the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, z=7/2. In terms of p, the binomial probability under the null hypothesis, p=1/(1+z), or equivalently, z=1/p-1.

Alternatively, instead of z the user can specify p directly. Note that only one of these inputs, z or p, has to be specified, but if both are entered the code will only work if z and p are such that p=1/(1+z). Otherwise, an error message will appear to remind that such condition must be complied.

With GroupSizes the user informs the sample size of each subsequent test. Therefore, only positive integers are accepted in GroupSizes.

The input Statistic specifies the scale selected by the user to inform CV.lower and cvs.upperamong the classic methods: MaxSPRT (Kulldorf et al., 2011), Pocock (Pocock, 1977), OBrien-Fleming

(O'Brien and Fleming, 1979), or Wang-Tsiatis (Jennison and Turnbull, 2000). For `Statistic="Wang-Tsiatis"`, the user has to choose a number in the (0, 0.5] interval for `Delta`.

Important: for time-variable matching ratios (i.e. when `z` or `p` changes from a test to another), only the "Statistic=Cases" option works. This is because the test statistic options are non-monotone with the number of cumulative cases under a variable `p` or `z` situation.

For `RR` the user must specify the target relative risks for calculation of the statistical performance measures to be delivered in the output. It can be a vector of positive number or a single number.

For details about the algorithm used to calculate the critical value, see the paper by Silva (2018).

Value

`AlphaSpend`	The alpha spending associated to the user-specified threshold.
`Performance`	A matrix with the following three performance measures for each target RR: statistical power, expected time to signal and expected sample size.

Acknowledgements

Development of the Performance.Threshold.Binomial function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0,2.0 to 3.1).
- Federal University of Ouro Preto (UFOP), through contract under internal UFOP's resolution CEPE 4600 (v2.0 to 3.1).

See also

`Performance.AlphaSpend.Binomial`: for calculating signaling threshold for user-specified alpha spending with binomial data.
`CV.Binomial`: for calculating Wald-type signaling thresholds for continuous sequential analysis with binomial data.
`Analyze.Binomial`: for performing sequential analysis with group, continuous or unpredictable sequential fashion with binomial data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials, London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

O'Brien PC, Fleming TR. (1979). A multiple testing procedure for clinical trials. Biometrics. 35:549–556.

Pocock SJ. (1977).Group sequential methods in the design and analysis of clinical trials. Biometrika. 64:191–199.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance with Binomial Data. Statistics in Medicine, 15;37(1), 107-118.

Silva IR, Kulldorff M. (2015). Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR, Maro J, Kulldorff M. (2021), Exact sequential test for clinical trials and post-market drug and vaccine safety surveillance with Poisson and binary data. Statistics in Medicine, DOI: 10.1002/sim.9094.

Examples

```
## Performance and Alpha spending of a four-group sequential
#  analysis with threshold informed in the scale of the
#  binomial data, i.e. Statistic="Cases".
#  The analysis is for a maximum sample size of 50 events under
#  upper-tailed testing, that is, H0:RR<=1, with irregular group
#  sizes of 12, 25, 35, and 45.
#  The matching ratio also changes in time with z= 1, 1.5, 2, 1.3.
#  The statistical performance is evaluated for RR= 1.2, 1.5, 2:

# res<- Performance.Threshold.Binomial(N=50,CV.upper=c(12,25,35,45),
# z=c(1,1.5,2,1.3),GroupSizes=c(15,15,10,10),Tailed="upper",
# Statistic="Cases", RR=c(1.2,1.5,2))
```

Performance.Threshold.CondPoisson

Performance and alpha spending for user-defined signaling threshold for sequential analysis with conditional Poisson data.

Description

The function `Performance.Threshold.CondPoisson` calculates the statistical power, expected time to signal, expected sample size and alpha spending associated to any user-specified signaling threshold, flat or non-flat, for continuous or group sequential analysis with conditional Poisson data, all for a pre-specified upper limit on the sample size.

Usage

```
Performance.Threshold.CondPoisson(K,cc,CV.upper="n",
Person_timeRatioH0="n",GroupSizes="n",Tailed="upper",RR)
```

Arguments

`K`	The upper limit on the sample size (length of surveillance) expressed in terms of the number of events arriving in the surveillance period. There is no default value.
`cc`	Number of events observed in the historical period. There is no default value.

`CV.upper` User-specified signaling threshold given in the scale of the CMaxSPRT test statistic. There is no default value.

`Person_timeRatioH0`
Test-specific amount of information of each test given in terms of the ratio between the person-time in the surveillance period and the overall person-time of the historical period. See Details. There is no default value.

`GroupSizes` Test-specific number of events between two looks at the data for group or continuous sequential analysis. There is no default value. See Details.

`Tailed` Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

`RR` Vector of relative risks for performance calculation. There is no default value.

Details

For continuous and group sequential analysis based on monitoring Poisson data conditioned on matched historical Poisson data, the power, expected time to signal, expected sample size and alpha spending impplied by user-specified thresholds are calculated with `Performance.Threshold.CondPoisson`. The user can select one between two scales to enter with the threshold, the Conditional Maximized Sequential Probability Ratio Test statistic (CMaxSPRT) scale (Li and Kulldorff, 2010), or the surveillance versus historical person-time ratio (Silva et al., 2019a). For the CMaxSRT scale, the input is `CV.upper`. This can be entered as a vector for group sequential analysis. For example, for a three-group sequential test, the i-th entry represents the critical value for the i-th test, with i=1, 2, 3. If a single number is informed in `CV.upper`, then a flat critical value for all tests is used for both continuous or group sequential fashions. The number of tests is defined with the input `GroupSizes`, as shall be detailed here after the desciption of `Person_timeRatioH0`.

An alternative way to inform the threshold is by using `Person_timeRatioH0`, which is in the scale of the ratio between the person-time in the surveillance period and the overall person-time of the historical period. Using the notation by Silva et al. (2019a) and Silva et al. (2019b), let V denote the total person-time from the historical data, where `cc` events were observed, and let $P_{k(i)}$ denote the the cummulative person-time from the surveillance data at the i-th test with a cummulative $k(i)$ events. With `Person_timeRatioH0`, the entries must have increasing numbers, from the first to the last. For example, for a three-group sequential plan with sample sizes of 20, 15, 25, a hypothetical choice is `Person_timeRatioH0=c(0.1, 0.5, 1)`. This way, H0 is rejected if: P_20/V <= 0.1 in the first test, or P_35/V <= 0.5 in the second test, or P_60/V <= 1 in the third test.

Note: only one of the inputs `CV.upper` or `Person_timeRatioH0` is to be used.

With `GroupSizes` the user informs the sample size of each test in the scale of the number of events in the surveillance period. Therefore, only positive numbers are accepted in `GroupSizes`. For irregular group sizes, a vector must be informed with each test-specific number of events between two looks at the data, therefore the entries of `GroupSizes` must sums up `K`. For regular group sizes, a single number can be informed for the constant sample size of each test. For example, for continuous sequential analysis, `GroupSizes=1`. For ten-group sequential analysis with `K=50`, `GroupSizes=5`.

For `RR` the user must specify the target relative risks for calculation of statistical performance measures. It can be a vector of positive numbers or a single number.

For details on the calculation of signaling thresholds and alpha spending for Poisson data conditioned to historical data, see the papers by Silva et al. (2019a) and Silva et al. (2019b), respectively.

Value

`Performance`	A matrix with the following three performance measures for each target RR: statistical power, expected time to signal and expected sample size.
`AlphaSpend`	The alpha spending associated to the user-specified threshold.

Acknowledgements

Development of the Performance.Threshold.CondPoisson function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0,2.0 to 3.1).
- Federal University of Ouro Preto (UFOP), through contract under internal UFOP's resolution CEPE 4600 (v2.0 to 3.1).

See also

`Performance.AlphaSpend.CondPoisson`: for calculating performance and signaling threshold for user-specified alpha spending with conditional Poisson data.
`CV.CondPoisson`: for calculating Wald-type signaling thresholds for continuous sequential analysis with conditional Poisson data based on the CMaxSPRT test statistic.
`Analyze.CondPoisson`: for performing sequential analysis with group, continuous or unpredictable sequential fashion for condicional Poisson data based on the CMaxSPRT test statistic.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials, London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

Li L, Kulldorff M. (2010). A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

O'Brien PC, Fleming TR. (1979). A multiple testing procedure for clinical trials. Biometrics. 35:549–556.

Pocock SJ. (1977). Group sequential methods in the design and analysis of clinical trials. Biometrika. 64:191–199.

Silva IR, Li L, Kulldorff M. (2019a), Exact conditional maximized sequential probability ratio test adjusted for covariates. Sequential Analysis, 38(1), 115–133.

Silva IR., Lopes LM., Dias P., Yih WK. (2019b). Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, 38(12), 2126–2138.

Silva IR, Kulldorff M. (2015). Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR, Maro J, Kulldorff M. (2019). Exact Sequential Analysis Using R Sequential. Working Paper.

Examples

```
#### Example 1
## Power, expected time to signal, expected sample size and
## alpha spending of three group CMaxSPRT sequential analysis with
#  a maximum sample size of 30 events for upper-tailed
#  testing, i.e. H0:RR<=1, with regular groups of sizes 10
#  and a flat threshold equal to 3.6.
#  The statistical performance is evaluated for two
#  target RR= 1.5, 2:

# Performance.Threshold.CondPoisson(K=30,cc=10,CV.upper=3.6, Person_timeRatioH0="n",
# GroupSizes=10,RR=c(1.5,2))

#### Example 2
## Power, expected time to signal, expected sample size and
#  alpha spending of three group CMaxSPRT sequential analysis with
#  a maximum sample size of 30 events for upper-tailed
#  testing, i.e. H0:RR<=1, with regular groups of sizes 10
#  and thresholds given in the P_k/V scale:
#  "Person_timeRatioH0=c(0.1, 0.5, 1)".
#  The statistical performance is evaluated for two
#  target RR= 1.5, 2:

# Performance.Threshold.CondPoisson(K=30,cc=10,CV.upper="n", Person_timeRatioH0=c(0.1, 0.5, 1),
# GroupSizes=10,RR=c(1.5,2))
```

Performance.Threshold.Poisson

Performance and alpha spending for user-defined signaling threshold for sequential analysis with Poisson data.

Description

The function `Performance.Threshold.Poisson` calculates power, expected time to signal and alpha spending associated to any user-specified signaling threshold, flat or non-flat, for continuous or group sequential analysis with Poisson data. The user can select the scale for the signaling threshold among MaxSPRT, Pocock, OBrien-Fleming, or Wang-Tsiatis test statistics, all for a pre-specified upper limit on the sample size.

Usage

```
Performance.Threshold.Poisson(SampleSize,CV.lower="n",CV.upper="n",
CV.events.upper="n",M=1,D=0,GroupSizes="n",Tailed="upper",
Statistic=c("MaxSPRT", "Pocock", "OBrien-Fleming", "Wang-Tsiatis"),
```

```
Delta="n",RR)
```

Arguments

SampleSize	The upper limit on the sample size (length of surveillance) expressed in terms of the expected number of events under the null hypothesis. There is no default value.
CV.lower	Signaling threshold for evidence of "RR<1". It is given in the scale of the selected test statistic infomed in "Statistic". There is no default value.
CV.upper	Signaling threshold for evidence of "RR>1". It is given in the scale of the selected test statistic infomed in "Statistic". There is no default value.
CV.events.upper	Signaling threshold for evidence of "RR>1". It is given in the scale of the events. The is no default value unless when one uses "CV.upper" instead.
M	The minimum number of events needed before the null hypothesis can be rejected. It must be a positive integer. A good rule of thumb is to set M=4 (Kulldorff and Silva, 2015). The default value is M=1, which means that even a single event can reject the null hypothesis if it occurs sufficiently early.
D	The expected number of events under the null hypothesis at the first look at the data. This is used when there is an initial large chunk of data arriving, followed by continuous sequential analysis. The default value is D=0, which is also the best choice. This means that there is no delay in the start of the sequential analyses. If D is very large, the maximum sample size will be set equal to D if a non-sequential analysis provides the desired power.
GroupSizes	Vector with the test-specific expected number of events under the null hypothesis between two looks at the data. Important: Must sums up "SampleSize". There is no default value.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.
Statistic	The test statistic scale used for "CV.lower" and "CV.upper". There is no default.
Delta	Parameter needed for calculation of Wang-Tsiatis test statistic if this is the option selected in "Statistic". Must be a number in the (0, 0.5] interval. There is no default value.
RR	Vector of relative risks for performance calculation. There is no default value.

Details

For continuous and group sequential analysis with Poisson data, the alpha spending for user-specified thresholds are calculated with Performance.Threshold.Poisson.

The inputs CV.lower, CV.upper, if two-tailed testing (Tailed="two"), both lower and upper signaling thresholds must be informed. If the user desires a constant threshold (critical value), then a single number can be informed for each of these inputs. For time-variable (non-constant) thresholds, the length of CV.lower and CV.upper must coincide with the length of GroupSizes.

For details about the algorithm used to calculate the critical value, see the paper by Silva (2018).

With GroupSizes the user informs the sample size of each subsequent test. Therefore, only positive numbers are accepted in GroupSizes.

The input Statistic specifies the scale selected by the user to inform CV.lower and CV.upper among the classic methods: MaxSPRT (Kulldorf et al., 2011), Pocock (Pocock, 1977), OBrien-Fleming (O'Brien and Fleming, 1979), or Wang-Tsiatis (Jennison and Turnbull, 2000). For Statistic="Wang-Tsiatis", the user has to choose a number in the (0, 0.5] interval for Delta.

For RR the user must specify the target relative risks for calculation of the statistical performance measures to be delivered in the output. It can be a vector of positive number or a single number.

Value

AlphaSpend	The alpha spending associated to the user-specified threshold.
Performance	A matrix with the following three performance measures for each target RR: statistical power, expected time to signal and expected sample size.

Acknowledgements

Development of the Performance.Threshold.Poisson function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0,2.0 to 3.1).
- Federal University of Ouro Preto (UFOP), through contract under internal UFOP's resolution CEPE 4600 (v2.0 to 3.1).

See also

Performance.AlphaSpend.Poisson: for calculating performance and signaling threshold for user-specified alpha spending with Poisson data.
CV.Poisson: for calculating Wald-type signaling thresholds for continuous and group sequential analysis with Poisson data.
Analyze.Poisson: for performing sequential analysis with group, continuous or unpredictable sequential fashion for Poisson data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Jennison C, Turnbull B. (2000). Group Sequential Methods with Applications to Clinical Trials, London: Chapman and Hall/CRC.

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. *Sequential Analysis*, **30**: 58–78.

O'Brien PC, Fleming TR. (1979). A multiple testing procedure for clinical trials. Biometrics. 35:549–556.

Pocock SJ. (1977). Group sequential methods in the design and analysis of clinical trials. Biometrika. 64:191–199.

Silva IR. (2018). Type I Error Probability Spending for Post-Market Drug and Vaccine Safety Surveillance With Poisson Data. Methodol Comput Appl Probab, 20(2), 739–750.

Silva IR, Kulldorff M. (2015). Continuous versus Group Sequential Analysis for Vaccine and Drug Safety Surveillance. Biometrics, 71 (3), 851–858.

Silva IR, Maro J, Kulldorff M. (2021). Exact sequential test for clinical trials and post-market drug and vaccine safety surveillance with Poisson and binary data. Statistics in Medicine, DOI: 10.1002/sim.9094.

Examples

```
## Example 1
#  Performance and alpha spending of group Poisson sequential analysis
#  with a maximum sample size of 90 expected events for two-tailed
#  testing, i.e. H0:RR=1, with irregular group sizes and different
#  and lower and upper thresholds with irregular
#  sample sizes 25, 20, 20, and 25.
#  The statistical performance is evaluated for four different
#  target RR= 1, 1.2, 2, 3:

# res<- Performance.Threshold.Poisson(SampleSize=90,CV.lower=c(2.5,2.6,2.7,2.8),
# CV.upper=c(3,3.1,3.2,3.3),GroupSizes=c(25,20,20,25),Tailed="two",
# Statistic="MaxSPRT",Delta="n",RR=c(1,1.2,2,3))

## Example 2
#  Suppose that the Analyze.Poisson function was used for an actual analysis.
#  For evaluating the cumulative power after a certain number of subsequent tests,
#  one can enter with the critical values delivered by Analyze.Poisson in the
#  Performance.Threshold.Poisson.
#  For example, suppose that the following thresholds in the scale of the events
#  were printed by Analyze.Poisson for the first three tests:
#     cv.events<- c(2,3,5)
#  which were obtained for the following specific sample sizes:
#     mus<- c(0.05,0.5,1.2)
#  Calculating the cumulative power, the expected time to signal, and
#  the expected sample size for RR=2:

# res<-Performance.Threshold.Poisson(SampleSize=sum(mus),CV.events.upper=cv.events,
# GroupSizes=mus, Statistic="MaxSPRT",RR=2)


# This returns a power about 30%.
```

`SampleSize.Binomial` *Sample size calculation for continuous sequential analysis with binomial data.*

Description

The function SampleSize.Binomial obtains the sample size needed to guarantee a desired statistical power, for a fixed true relative risk, when doing continuous sequential analysis for binomial data with a Wald-type upper boundary, which is flat with respect to the log-likelihood ratio. It can also be used to approximate the sample size needed when doing group sequential analysis for binomial data.

Usage

```
SampleSize.Binomial(RR,alpha=0.05,power=0.9,M=1,z="n",p="n",Tailed="upper")
```

Arguments

RR	A target vector of relative risks to be detected with the requested statistical powers.
alpha	The significance level. The default value is "alpha=0.05". Must be in the range (0, 0.5].
power	The target vector of overall statistical powers to detect an increased risk of the relative risk (RR). The default value is "power=0.90".
M	The minimum number of events needed before the null hypothesis can be rejected. It must be a positive integer. The default value is "M=1".
z	For a matched case-control analysis, z is the number of controls matched to each case under the null hypothesis. There is no default value.
p	The probability of having a case under the null hypothesis. There is no default value.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

The function SampleSize.Binomial calculates the sample size N, to be used for the continuous binomial MaxSPRT in order to provide the desired statistical power for a user-specified relative risk RR. The required sample size (Required_N) is expressed in terms of the total number of observations, and it is the number of observations by which the sequential analysis will end without rejected the null hypothesis. The solution is exact using iterative numerical calculations.

The required sample size, N, increases for increasing values of power, while N decreases for increasing values of alpha, the relative risk RR and the minimum number of events needed to signal M. For increasing values of z, the required sample size N can either decrease or increase.

While this function calculates the required sample size for continuous sequential analysis, it can also be used as an approximation for group sequential analyses. With the same Required_N, and all other parameters being the same, a group sequential analysis will always give higher statistical power than a continuous sequential analysis, so SampleSize.Binomial can be use to ensure the required statistical power for group sequential analyses.

The input z represents the number of controls matched to each case. For example, if there are 3 controls matched to each case, "z=3". In a self-control analysis, z is the ratio of the control interval

to the risk interval. For example, if the risk interval is 2 days long and the control interval is 7 days long, z=7/2. In terms of p, the binomial probability under the null hypothesis, p=1/(1+z), or equivalently, z=1/p-1. The parameter z must be a positive number.

Alternatively, instead of z the user can specify p directly. Note that only one of these inputs, z or p, has to be specified, but if both are entered the code will only work if z and p are such that p=1/(1+z). Otherwise, an error message will appear to remind that such condition must be complied.

Value

`SampleSize_by_RR_Power`

A table containing the main performance measures associated to the required samples sizes for each combination of RR and power. The sample size N is provided in terms of the total number of observations. In the case-control setting, this is equal to the total number of cases and controls. In the self-control setting, it is equal to the total number of events in either the risk or the control interval.

Acknowledgements

Development of the SampleSize.Binomial function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999.

See also

`CV.Binomial`: for calculating critical values in continuous sequential analysis with binomial data.
`Performance.Binomial`: for calculating the statistical power, expected time to signal and expected time of analysis for continuous sequential analysis with binomial data.
`SampleSize.Poisson`: sample size calculation for continuous sequential analysis with Poisson data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events to Signal. REVSTAT Statistical Journal, 15(3): 373–394.

Examples

```
result<- SampleSize.Binomial(RR=5,alpha=0.01,power=0.88,M=1,z=2)
# if you type:
result
# then you will get the following output:
# $Required_N
# [1] 25

# $cv
# [1] 4.59581

# $Type_I_Error
```

```
# [1] 0.009755004

# $Actual_power
# [1] 0.8855869
```

`SampleSize.CondPoisson`

Sample size calculation for the continuous sequential CMaxSPRT for Poisson data with limited information from historical cohort.

Description

The function `SampleSize.CondPoisson` obtains the required sample size (length of surveillance) needed to guarantee a desired statistical power for a pre-specified relative risk, when doing continuous sequential CMaxSPRT, using a Wald-type upper boundary, which is flat with respect to the likelihood ratio function.

Usage

```
SampleSize.CondPoisson(cc,D=0,M=1,alpha=0.05,power=0.9,RR=2,
Tailed="upper",NegBinApprox= TRUE)
```

Arguments

`cc`	The total number of observed adverse events in the historical data. There is no default.
`D`	The minium number for the ratio P_k/V before the null hypothesis can be rejected. The default value is $D = 0$. A delayed start with $D > 0$ is recommended to avoid signaling very early on such that very little information would be available to judge whether the signal is more likely to be a true signal or chance finding.
`M`	The minimum number of events needed before the null hypothesis can be rejected. The default value is M=1. A delayed start with $M > 1$ is recommended to avoid signaling very early on such that very little information would be available to judge whether the signal is more likely to be a true signal or chance finding.
`alpha`	The significance level. It must be in the range (0,0.5]. The default value is alpha=0.05.
`power`	The target vector of overall statistical powers to detect an increased relative risk (RR). The default value is power=0.90.
`RR`	The target vector of relative risks to be detected with the requested vector of statistical powers. The default value is RR=2.
`Tailed`	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.
`NegBinApprox`	NegBinApprox=TRUE (default) for a fast calculation based on the negative binomial approximation. See details.

Details

When using the CMaxSPRT (Li and Kulldorff, 2010) and the `CV.CondPoisson` function to conduct continuous sequential analysis for Poisson data and limited historical data, the null hypothesis is rejected when the log likelihood ratio exceeds the predetermined critical value calculated by `CV.CondPoisson`. The sequential analysis ends without rejecting the null hypothesis when a predetermined upper limit on the sample size is reached, expressed either in terms of the ratio of the cumulative person-time in the surveillance population divided by the total cumulative person-time in historical data (StopType="Tal"), or in terms of the observed number of events in the surveillance population (StopType="Cases"). For example, the sequential analysis may end as soon as the sample size is such that the cumulative person-time in the surveillance population is twice the cumulative person-time in historical data, or there are 50 observed events in the surveillance population. The function `SampleSize.CondPoisson` calculates what the upper limit on the sample size (length of surveillance) that is required for the continuous CMaxSPRT to achieve the desired statistical power for a pre-specified relative risk RR. It outputs the upper limit on sample size for both definitions of the surveillance length, one expressed in terms of the ratio of the cumulative person-time in the surveillance population divided by the total cumulative person-time in historical data (T), and the other one expressed in terms of the observed number of events in the surveillance population (K). To save computing time, the negative binomial approximation proposed by Silva and Montalban (2023) is used.

Value

`SampleSize_by_RR_Power`
: A table containing the main performance measures associated to the required samples sizes, expressed in the scale of the number of events in the surveillance period, for each combination of RR and power.

Acknowledgements

Development of the `SampleSize.CondPoisson` function was funded by:
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0.1, v2.0.2). - Foundation for Research Support of Minas Gerais State (FAPEMIG), MG, Brazil, through the grant Demanda Universal.

See also

`CV.CondPoisson`: calculating the critical value for continuous CMaxSPRT.
`Performance.CondPoisson`: calculating the statistical power, expected time to signal and expected time of analysis for continuous CMaxSPRT.

Author(s)

Ivair Ramos Silva, Lingling Li

References

Li L, Kulldorff M. (2010). A conditional maximized sequential probability ratio test for Pharmacovigilance. Statistics in Medicine, 29 (2), 284–295.

Silva IR, Li L, Kulldorff M. (2019). Exact Conditional Sequential Testing for Poisson Data. Sequential Analysis, in press.

Silva IR., Lopes LM., Dias P., Yih WK. (2019). Alpha Spending for Historical Versus Surveillance Poisson Data With CMaxSPRT. Statistics in Medicine, DOI: 10.1002/sim.8097, 1–13.

Silva IR, Montalban, J. (2023), The person-time ratio distribution for the exact monitoring of adverse events: Historical vs surveillance Poisson data, Statistics in Medicine, 42(18), 3283–3301.

Examples

```
# Sample size required to obtain a power of 90%, for a relative risk of 2,
# no delay for starting the surveilla# nce (D=0), under an alpha level of 5%,
# with 50 events in the historical data.

# res<- SampleSize.CondPoisson(cc=50,D=0,M=1,alpha=0.05,power=0.9,RR=2)
```

`SampleSize.Poisson` *Sample size calculation for continuous sequential analysis with Poisson data.*

Description

The function `SampleSize.Poisson` obtains the required sample size (length of surveillance) needed to guarantee a desired statistical power for a pre-specified relative risk, when doing continuous sequential analysis for Poisson data with a flat upper boundary in the scale of the Wald type MaxSPRT (log-likelihood ratio scale), Pocock, OBrien-Fleming, or Wang-Tsiatis scales. Alternatively, `SampleSize.Poisson` calculates sample sizes for non-flat signaling thresholds for user-defined alpha spending functions. It can also be used to approximate the sample size needed when doing group sequential analysis for Poisson data.

Usage

```
SampleSize.Poisson(alpha=0.05,power=0.9,M=1,D=0,RR=2,
precision=0.001,alphaSpend="n",rho="n",gamma="n",
Statistic=c("MaxSPRT", "Pocock", "OBrien-Fleming", "Wang-Tsiatis"),
Delta="n",Tailed="upper")
```

Arguments

`alpha`	The significance level. The default value is alpha=0.05. Must be in the range (0,0.5].
`power`	The target vector of overall statistical powers to detect an increased relative risk (RR). The default value is power=0.90.

M	The minimum number of events needed before the null hypothesis can be rejected. It must be a positive integer. A good rule of thumb is to set M=4 (Kulldorff and Silva, 2015). The default value is M=1, which means that even a single event can reject the null hypothesis if it occurs sufficiently early.
D	The expected number of events under the null hypothesis at the first look at the data. This is used when there is an initial large chunk of data arriving, followed by continuous sequential analysis. The default value is D=0, which is also the best choice. This means that there is no delay in the start of the sequential analyses. If D is very large, the maximum sample size will be set equal to D if a non-sequential analysis provides the desired power.
RR	The target vector of relative risks to be detected with the requested statistical vector of powers. The default value is RR=2.
precision	The tolerance for the difference between the requested and actual statistical power. Should be very small. The default value is precision=0.001.
alphaSpend	An integer between 1 to 4. Default is "n". See Details.
rho	Positive number used for the power-type alpha spending function (AlphaSpend=1) only. The default value is "rho=0.5". See Details.
gamma	Positive number used for the gamma-type alpha spending function (AlphaSpend=4) only. There is no default value. See Details.
Statistic	The test statistic scale to deliver the signaling threshold. See Details.
Delta	Parameter needed for calculation of Wang-Tsiatis test statistic if this is the option selected in "Statistic". Must be a number in the (0, 0.5] interval. There is no default value.
Tailed	Tailed="upper" (default) for H0:RR<=1, and Tailed="lower" for H0:RR>=1 or Tailed="two" for H0:RR=1.

Details

When using the MaxSPRT and the CV.Poisson function to conduct continuous sequential analysis for Poisson data, the null hypothesis is rejected when the log likelihood ratio exceeds the predetermined critical value calculated by CV.Poisson. The sequential analysis ends without rejecting the null hypothesis when a predetermined upper limit on the sample size is reached, expressed in terms of the expected number of events under the null hypothesis. For example, the sequential analysis may end as soon as the sample size is such that there are 50 expected events under the null.

The default in the function SampleSize.Poisson is for calculating the upper limit on the sample size (length of surveillance) required for the continuous Poisson based MaxSPRT (alphaSpend="n") to achieve the desired statistical power for a pre-specified relative risk RR. The solution is exact using iterative numerical calculations (Kulldorff et al., (2011).

While designed for continuous sequential analysis with flat threshold in the scale of the MaxSPRT statistic, the SampleSize.Poisson function can also be used to approximate the required upper limit on the sample size that is needed when doing group sequential analysis for Poisson data, using the CV.G.Poisson function.

There is also the possibility to calculate the sample size for an user-defined alpha spending plan. This is possible with the input parameter alphaSpend. The user can select among one of the four classical alpha spending shapes bellow:

$F_1(t) = \alpha t^{\rho}$, where $\rho > 0$,
$F_2(t) = 2 - 2\Phi(x_{\alpha}\sqrt{t^{-1}})$, where $x_{\alpha} = \Phi^{-1}(1 - \alpha/2)$,
$F_3(t) = \alpha \times log(1 + [exp1 - 1] \times t)$,
$F_4(t) = \alpha[1 - exp(-t\gamma)]/[1 - exp(-\gamma)]$ with $\gamma \in \Re$,
and t represents a fraction of the maximum length of surveillance.

To select one of the four alpha spending types above, and using an integer i to indicate the type among $i = 1$, 2, 3, and 4, for $F_1(t)$, $F_2(t)$, $F_3(t)$ and $F_4(t)$, respectively, one needs to set `alphaSpend=i`. Specifically for `alphaSpend=1`, it is necessary to choose a `rho` value, or a `gamma` value if `alphaSpend=4` is used.

For more details on these alpha spending choices, see the paper by Silva et al. (2021), Section 2.7.

When one sets `alphaSpend=i`, the threshold impplied by the correspondent alpha spending is calculated. The function delivers the threshold in the scale of a test statistic selected by the user with the input `Statistic` among the classic methods: MaxSPRT (Kulldorf et al., 2011), Pocock (Pocock, 1977), OBrien-Fleming (O'Brien and Fleming, 1979), or Wang-Tsiatis (Jennison and Turnbull, 2000). For `Statistic="Wang-Tsiatis"`, the user has to choose a number in the (0, 0.5] interval for `Delta`.

Value

`SampleSize_by_RR_Power`
: A table containing the main performance measures associated to the required samples sizes, expressed in terms of the expected number of events under the null hypothesis, for each combination of RR and power.

Acknowledgements

Development of the `SampleSize.Poisson` function was funded by:
- National Council of Scientific and Technological Development (CNPq), Brazil (v1.0).
- Bank for Development of the Minas Gerais State (BDMG), Brazil (v1.0).
- National Institute of General Medical Sciences, NIH, USA, through grant number R01GM108999 (v2.0.1,2.0.2).

See also

`CV.Poisson`: for calculating critical values for continuous sequential analysis with Poisson data.
`Performance.Poisson`: for calculating the statistical power, expected time to signal and expected sample size for continuous sequential analysis with Poisson data
`SampleSize.Binomial`: for calculating the minimum sample size given a target power in continuous sequential analysis with binomial data.

Author(s)

Ivair Ramos Silva, Martin Kulldorff.

References

Kulldorff M, Davis RL, Kolczak M, Lewis E, Lieu T, Platt R. (2011). A Maximized Sequential Probability Ratio Test for Drug and Safety Surveillance. Sequential Analysis, 30: 58–78. Kulldorff M, Silva IR. (2015). Continuous Post-market Sequential Safety Surveillance with Minimum Events

to Signal. REVSTAT Statistical Journal, 15(3): 373–394. Silva IR, Maro J, Kulldorff M. (2021). Exact sequential test for clinical trials and post-market drug and vaccine safety surveillance with Poisson and binary data. Statistics in Medicine, DOI: 10.1002/sim.9094.

Examples

```
### Example 1:
##  Sample size required to obtain a power of 80%, for a relati-
##  ve risk of 3, no delay for starting the surveillance (D=0),
##  and when the null hypothesis can be rejected with one event
##  (M=1) under an alpha level of 5%.

# result1<- SampleSize.Poisson(alpha=0.05,power=0.8,M=1,D=0,RR=3)
# result1


## Example 2:
##  Sample size required to obtain a power of 90%, for a relati-
##  ve risk of 2, no delay for starting the surveillance (D=0),
##  and when the null hypothesis can be rejected only after 2
##  events (M=2) under an alpha level of 10%.
##
# result2<- SampleSize.Poisson(alpha=0.1,power=0.9,M=2,D=0,RR=2)
# result2

## Example 3:
##  Sample size calculated for the non-flat threshold using the
##  power-type alpha spending (alphaSpend=1), with rho=1,
##  to obtain a power of 80% for a relative risk of 2.5, delay to
##  start the surveillance equal to 1 (D=1), and the null hypo-
##  thesis can be rejected with 3 events (M=3) under an alpha
##  level of 5%. The critical values will be shown in the scale
##  of the MaxSPRT statistic.

# result3<- SampleSize.Poisson(alpha=0.05,power=0.8,M=3,D=1,RR=2.5,
# alphaSpend=1,rho=1,Statistic="MaxSPRT")
# result3
```

Index