Package ‘PSSTools’

February 14, 2023

Type Package

Title PSSTools: Power and Sample Size Tools

Version 0.1.0

Description This package accompanies the textbook ``Power and Sample Size in R" by
Catherine M. Crespi for calculating power and sample size requirements for designing studies.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Suggests designsize,
longpower,
npsurvSS,
powerSurvEpi,
ssanv,
testthat (>= 3.0.0)

Config/testthat/edition 3

Imports clusterPower,
dplyr,
exact2x2,
magrittr,
MESS,
MKmisc,
plyr,
PowerTOST,
precisely,
presize,
purrr,
pwr,
pwr2,
pwr2ppl,
pwrAB,
SampleSize4ClinicalTrials,
stats,
Superpower

Depends R (>= 2.10)

R topics documented:

`anova3_ss` *One way ANOVA sample size calculation for omnibus F test with 3 group levels*

Description

This function searches through different group sample sizes until the specified power level is achieved. It allows for unequal standard deviations between groups and unequal group allocation. This function is a wrapper for the 'anova1f_3' function in the 'pwr2ppl' package.

Usage

```
anova3_ss(
  means,
  sds,
  alphalevel,
  powerlevel,
  groupratios = NULL,
  Nmin = NULL,
  Nmax = NULL,
  upperbound = 500
)
```

Arguments

`means`	A vector of group means.
`sds`	A vector of group standard deviations. This function allows unequal standard deviations between groups.
`alphalevel`	The significance level.

powerlevel	The level of power to achieve as a decimal.
groupratios	A 3 element vector of allocation ratios to each group. Default is a 1:1:1 ratio, supplied as c(1,1,1).
Nmin	The minimum total sample size N to search through. Default start is 30.
Nmax	The maximum total sample size N to search through. Default ending is 300.
upperbound	The largest integer to multiply the allocation ratios by. Default is 500 meaning if there is even allocation, this will allow a search of up to 500 people per group.

Value

Returns a data frame with sample sizes for each group indicated in n1, n2, and n3, actual power, and any notes from the computation.

Examples

```
# Sample Size for 3 groups with unequal allocation at 0.05 significance and 80% power
# explores range itself from N = 30 to 300
anova3_ss(means = c(5, 10, 12), sds = c(10, 8, 10), alphalevel = 0.05,
powerlevel = 0.8, groupratios = c(1, 1, 3),
Nmin = NULL, Nmax = NULL)
```

anova4_ss *One way ANOVA sample size calculation for omnibus F test with 4 group levels*

Description

This function searches through different group sample sizes until the specified power level is achieved. It allows for unequal standard deviations between groups and unequal group allocation. This is a wrapper for 'anova1f_4' function from the 'pwr2ppl' package.

Usage

```
anova4_ss(
  means,
  sds,
  alphalevel,
  powerlevel,
  groupratios = NULL,
  Nmin = NULL,
  Nmax = NULL,
  upperbound = 500
)
```

Arguments

means	A vector of group means.
sds	A vector of group standard deviations. This function allows unequal standard deviations between groups.
alphalevel	The significance level.

`powerlevel`	The level of power to achieve.
`groupratios`	A 4 element vector of allocation ratios to each group. Default is a 1:1:1:1 ratio, supplied as c(1,1,1,1).
`Nmin`	The minimum total sample size N to search through. Default start is 30.
`Nmax`	The maximum total sample size N to search through. Default ending is 300.
`upperbound`	The largest integer to multiply the allocation ratios by. Default is 500 meaning if there is even allocation, this will allow a search of up to 500 people per group.

Value

Returns a data frame with sample sizes for each group indicated in n1, n2, n3, and n4, actual power, and any notes from the computation.

Examples

```
# We find the sample size per group required for 4 group ANOVA with equal
# standard deviations and 2:1:1:2 group allocation ratio.
anova4_ss(means = c(5, 10, 12, 15), sds = c(10, 10, 10, 10), alphalevel = 0.05,
          powerlevel = 0.8, groupratios = c(2, 1, 1, 2),
          Nmin = 60, Nmax = 120)
```

`crossover_power` *Cross Over Power Calculation*

Description

Calculate power for either repeated cross sectional or closed cohort cross over designs.

Usage

```
crossover_power(design = "RCS", alpha, J, m, rho, rho_b, xi = 0, d)
```

Arguments

`design`	Either "RCS" for repeated cross-sectional design or "Cohort" for closed cohort design. Default is "RCS".
`alpha`	The significance level or type 1 error rate
`J`	Number of clusters
`m`	The number of individuals in each cluster at each time period
`rho`	Intraclass correlation coefficient (ICC)
`rho_b`	Between period ICC (correlation between outcomes of two individuals in the same cluster but different time periods)
`xi`	Within-cluster, within-subject correlation (correlation between two measurements within the same subject). Note this is different from subject autocorrelation. We expect xi to be larger than rho and rho_b. This is 0 for RCS cross over designs.
`d`	The standardized effect size

Value

power

Examples

```
# Repeated cross sectional cross over power calculation as in Example 12.3
crossover_power(design = "RCS", alpha = 0.05, J = 8, m = 30, rho = 0.05,
                rho_b = 0.025, xi = 0, d = 0.3)
# Closed cohort cross over power calculation as i Example 12.3
crossover_power(design = "Cohort", alpha = 0.05, J = 8, m = 30, rho = 0.05,
                rho_b = 0.025, xi = .4, d = 0.3)
```

---

crossover_ss *Cross Over Sample Size Calculation*

Description

Calculate sample size requirements for either repeated cross sectional or closed cohort cross over designs.

Usage

```
crossover_ss(design = "RCS", alpha, power, m, rho, rho_b, xi = 0, d)
```

Arguments

design	Either "RCS" for repeated cross-sectional design or "Cohort" for closed cohort design. Default is "RCS".
alpha	The significance level or type 1 error rate
power	Specified power level to achieve
m	The number of individuals in each cluster at each time period
rho	Intraclass correlation coefficient (ICC)
rho_b	Between period ICC (correlation between outcomes of two individuals in the same cluster but different time periods)
xi	Within-cluster, within-subject correlation (correlation between two measurements within the same subject). Note this is different from subject autocorrelation. We expect xi to be larger than rho and rho_b. This is 0 for RCS cross over designs.
d	The standardized effect size

Value

Calculated number of clusters required and adjusted number of clusters to account for normal approximation

Examples

```
# repeated cross sectional sample size as in Example 12.4
crossover_ss(design = "RCS", alpha = 0.05, power = 0.8, m = 30, rho = 0.05,
            rho_b = 0.025, xi = 0, d = 0.3)
# closed cohort sample size requirement
crossover_ss(design = "Cohort", alpha = 0.05, power = 0.8, m = 30,
            rho = 0.05, rho_b = 0.025, xi = 0.4, d = 0.3)
```

---

mcnemar_ss *Sample Size Calculation for McNemar's Test for paired proportions*

Description

Calculates the number of pairs needed to achieve specified level of power from marginal probabilities and the correlation between probabilities (p1, p2, rho) OR from discordant cell probabilities (p01 and p10).

Usage

```
mcnemar_ss(
  p1 = NULL,
  p2 = NULL,
  rho = NULL,
  p01 = NULL,
  p10 = NULL,
  alpha = 0.05,
  power = 0.8,
  one_or_twosides = "two"
)
```

Arguments

p1	Marginal probability of success for outcome 1. Either NULL or supplied alongside p2 and rho.
p2	Marginal probability of success for outcome 2. Either NULL or supplied alongside p1 and rho.
rho	Correlation between proportions. Either NULL or supplied alongside p1 and p2.
p01	First discordant cell probability. Either NULL or supplied alongside p10.
p10	Second discordant cell probability. Either NULL or supplied alongside p01.
alpha	Significance level. Default is 0.05
power	Required power. Default is 0.8.
one_or_twosides	Whether the hypothesis test of interest is one or two-sided. Default is "two".

Value

Returns a data frame with total number of observations needed (N_obs) and number of pairs needed (N_pairs).

Examples

```
# Calculate sample size supplying marginal probabilities of 0.8 and 0.9
# at 90% power without any correlation between probabilities.
mcnemar_ss(p1 = 0.8, p2 = 0.9, rho = 0, alpha = 0.05, power = 0.9)
# Calculate sample size supplying cell probabilities at 90% power
mcnemar_ss(p10 = 0.18, p01 = 0.08, alpha = 0.05, power = 0.9)
```

---

meandiff_ci_power *Calculate Power of a confidence interval for a Difference of Two Means*

Description

This function calculates the power of a confidence interval for the difference of two means accounting for estimating sigma and allowing for unequal allocation between the two groups.

Usage

```
meandiff_ci_power(n1, n2, alpha, d = NULL, halfwidth = NULL, sigma = NULL)
```

Arguments

n1	The sample size in group 1
n2	The sample size in group 2
alpha	The significance level
d	The standardized halfwidth (halfwidth / sigma). Either d OR halfwidth and sigma need specified.
halfwidth	The halfwidth; half of the confidence interval width. If halfwidth is specified, sigma needs specified too.
sigma	The estimated standard deviation. If sigma is specified, halfwidth needs specified too.

Value

The total N, group sample sizes, and corresponding power.

Examples

```
# Calculate power for a 95% confidence interval to have 0.25 standard deviation width
# when there are 100 and 160 people in each group
meandiff_ci_power(n1 = 100, n2 = 160, alpha = 0.05, d = 0.25, halfwidth = NULL, sigma = NULL)
```

meandiff_ci_ss *Calculate Sample Size needed for adequate power for Confidence Interval of Difference in Means.*

Description

Calculate sample sizes in two groups (allowing unequal allocation) that will adequately power the confidence interval of the difference of group means. This function accounts for estimating sigma, the standard deviation.

Usage

```
meandiff_ci_ss(
  N_min,
  N_max,
  ratios = c(1, 1),
  alpha,
  power,
  d = NULL,
  halfwidth = NULL,
  sigma = NULL
)
```

Arguments

N_min	The minimum total sample size.
N_max	The maximum total sample size
ratios	A 2 element vector of group allocation ratios. Equal allocation is default, which is specified as c(1,1). A allocation ratio of 2 (r = n2/n1) would be specified as c(1,2).
alpha	The significance level
power	The specified power to achieve
d	The standardized halfwidth (halfwidth / sigma). Either d OR halfwidth and sigma need specified.
halfwidth	The halfwidth; half of the confidence interval width. If halfwidth is specified, sigma needs specified too.
sigma	The estimated standard deviation. If sigma is specified, halfwidth needs specified too.

Value

returns all sample sizes and power that satisfy the specified power level.

Examples

```
# Find sample sizes with allocation ratio of 2 between two groups for a 95% confidence
# interval to have 80% probability of a 0.25 standard deviation halfwidth or smaller.
meandiff_ci_ss(N_min = 200, N_max = 325, r = c(1,2),
               alpha = 0.05, power = 0.8, d = 0.25)
```

multisite.data *Multisite Dataframe*

Description

A simulated dataset for multisite trials with and without interactions.

Usage

```
multisite.data
```

Format

## ‘multisite.data‘ A data frame with 200 rows and 9 columns:

id Individual observation number which goes from id = 1 to 200 (N)

j Site number which runs from j = 1 to 10 (J)

i Within site observation number. This runs from i = 1 to 20 (m)

xij Group level indicator variable coded so that within each site xij sums to 0.

Yij Outcome variable with a site by group interaction.

Wij Outcome variable without a site by group interaction.

onemean_ci_power *Calculate Power for a Confidence Interval for One Mean*

Description

Calculates the power of a confidence interval for a single mean accounting for estimation in sigma, the standard deviation.

Usage

```
onemean_ci_power(N, d, alpha)
```

Arguments

N	The number of observations
d	The standardized halfwidth (halfwidth / standard deviation)
alpha	The level of significance

Value

The statistical power associated with the number of observations.

Examples

```
# Calculate probability that a 95% confidence interval for group size of 73
# to have 0.25 standard deviation halfwidth or less
onemean_ci_power(N = 73, d=0.25, alpha=0.05)
```

onemean_ci_ss *Sample Size Calculation for One Mean Confidence Interval*

Description

This function calculates the minimum sample size required to achieve certain precision in estimating a confidence interval for a single mean. This function accounts for estimation in sigma and searches over a range of supplied N. 'Power' used here is the power of a confidence interval, which is not true statistical power; instead it is the probability of the interval obtaining a certain halfwidth.

Usage

```
onemean_ci_ss(N_min, N_max, d, alpha, power)
```

Arguments

N_min	The minimum number of observations to explore power.
N_max	The maximum number of observations to explore power.
d	The standardized halfwidth (halfwidth / standard deviation).
alpha	The significance level.
power	The desired level of power.

Value

The minimum sample size required to achieve the specified level of power.

Examples

```
# Example 1: Find the required sample size that achieves a half width of 0.25 standard deviations
# at significance level of 0.05 with 80% power. We specify to search between 62 and 75 people.
onemean_ci_ss(N_min = 62, N_max = 75, d=0.25, alpha=0.05, power=0.8)
```

oneprop_ss *One proportion sample size calculation*

Description

Calculate sample size requirements for one proportion using the conditional or unconditional method.

Usage

```
oneprop_ss(
  p0,
  pA,
  alpha,
  power,
  method = "conditional",
  one.or.two.sided = "two"
)
```

Arguments

p0	Null hypothesis proportion.
pA	Alternative hypothesis proportion.
alpha	Significance level.
power	Power level as a decimal (0.8 for 80 percent power).
method	Either "conditional" or "unconditional" method for calculation. Default is conditional.
one.or.two.sided	Either "one" or "two" to specify a one or two sided hypothesis test. Default is two-sided.

Value

Returns n, the sample size needed for a one proportion test.

Examples

```
# Example 5.1: Calculate sample size needed for experimental therapy
# to be considered promising (at least 20%) if the true proportion of
# responders is 0.3. Using a one-sided test with significance at 0.05
# and power at 0.8.
oneprop_ss(p0 = 0.2, pA = 0.3, alpha = 0.05, power = 0.8,
           method = "conditional", one.or.two.sided = "one")
oneprop_ss(p0 = 0.2, pA = 0.3, alpha = 0.05, power = 0.8,
           method = "unconditional", one.or.two.sided = "one")
# Example 5.2: Same set up as above, but now the criteria are promising
# at 0.4 and true response is 0.5.
oneprop_ss(p0 = 0.4, pA = 0.5, alpha = 0.05, power = 0.8,
           method = "unconditional", one.or.two.sided = "one")
oneprop_ss(p0 = 0.4, pA = 0.5, alpha = 0.05, power = 0.8,
           method = "conditional", one.or.two.sided = "one")
```

parallelCRT_power *Parallel Cluster Randomized Trial Power Calculation*

Description

Parallel Cluster Randomized Trial Power Calculation

Usage

```
parallelCRT_power(alpha, J, m, d, rho, rho_c, rho_s)
```

Arguments

alpha	The significance level or type 1 error rate
J	Number of clusters
m	The number of individuals in each cluster at each time period
d	The standardized effect size
rho	Intraclass correlation coefficient (ICC)

rho_c	Cluster autocorrelation (proportion of cluster-level variance that is time invariant)
rho_s	Subject autocorrelation (This is 0 in repeated cross sectional designs because people are measured only once)

Value

power

Examples

```
# closed cohort design as in Example 12.1
parallelCRT_power(alpha = 0.05, J = 16, m = 30, d = 0.3, rho = 0.05, rho_c = 0.4, rho_s = 0.5)
# repeated cross sectional design as in Example 12.1
parallelCRT_power(alpha = 0.05, J = 16, m = 30, d = 0.3, rho = 0.05, rho_c = 0.4, rho_s = 0)
```

parallelCRT_ss *Parallel Cluster Randomized Trial Sample Size Calculation*

Description

Parallel Cluster Randomized Trial Sample Size Calculation

Usage

```
parallelCRT_ss(alpha, power, m, d, rho, rho_c, rho_s)
```

Arguments

alpha	The significance level or type 1 error rate
power	The specified level of power to achieve
m	The number of individuals in each cluster at each time period
d	The standardized effect size
rho	Intraclass correlation coefficient (ICC)
rho_c	Cluster autocorrelation (proportion of cluster-level variance that is time invariant)
rho_s	Subject autocorrelation (This is 0 in repeated cross sectional designs because people are measured only once)

Value

Calculated number of clusters required, adjusted number of clusters, and suggested number of clusters that includes the value of K

Examples

```
# example 12.2
parallelCRT_ss(alpha = 0.05, power = 0.8, m = 30, d = 0.3, rho = 0.05, rho_c = 0.4, rho_s = 0.5)
```

`search_power` *Search through multiple variables to achieve adequate power*

Description

This function will search through any power function with up to 4 variables to search through. Because the search vectors are supplied without names, they must be provided in the default order of the power function.

Usage

```
search_power(
  powerfunction,
  searchvector1,
  searchvector2 = NULL,
  searchvector3 = NULL,
  searchvector4 = NULL,
  ...
)
```

Arguments

powerfunction	The power function of interest. Any function can be specified.
searchvector1	One vector of variable values to search through. This is a required argument.
searchvector2	Optional second vector of variable values to search through. This must either be the same length as the searchvector1 or can be a multiple of searchvector1. If searchvector1 has 4 elements then searchvector2 needs to have either 2, 4, 8, etc elements.
searchvector3	Optional third vector of variable values to search through. Similarly to searchvector2, this must either be the same length as other searchvectors or a multiple.
searchvector4	Optional fourth vector of variable values to search through. This must either be the same length as other searchvectors or a multiple.
...	Additional arguments to pass to the specified powerfunction.

Value

A data frame with the resulting power at the specified search values.

Examples

```
# Example 1: Search through group sizes 3 and 4 (k) and group sample sizes 80
# and 100 (n) in one way ANOVA. alpha = 0.05, delta = 3, sigma = 8.5 are all
# arguments that are being passed to the pwr.1way function. These arguments are
# required for this function.
library(pwr2)
search_power(powerfunction = pwr.1way, searchvector1 = c(3, 3, 4, 4),
                     searchvector2 = c(80, 100, 80, 100),
                     searchvector3 = NULL, searchvector4= NULL,
                     alpha = 0.05, delta = 3, sigma = 8.5)

# Example 2: Search through total number of observations 80, 100, or 120 (N) and
# allocation percents of 0.4 and 0.5 to group B (percent_B)
library(pwrAB)
search_power(powerfunction = AB_t2n, searchvector1 = c(80, 80, 100, 100, 120, 120),
                     searchvector2 = c(0.4, 0.5), mean_diff = -2,
                     sd_A = 4 , sd_B = 6, sig_level = 0.05,
                     power = NULL, alt = c("less"))
```

swd_1trt_power *Stepped Wedge Design Power Calculation with 1 Treatment*

Description

Calculates power for stepped wedge design with one treatment and one control condition.

Usage

```
swd_1trt_power(alpha = 0.05, J, m, K, b, d, rho, rho_c, rho_s)
```

Arguments

alpha	The significance level or type 1 error rate
J	Number of clusters
m	The number of individuals in each cluster at each time period
K	The number of steps (if one baseline period, then this is periods - 1)
b	The number of baseline periods
d	The standardized effect size
rho	Intraclass correlation coefficient (ICC)
rho_c	Cluster autocorrelation (proportion of cluster-level variance that is time invariant)
rho_s	Subject autocorrelation (This is 0 in repeated cross sectional designs because people are measured only once)

Value

power

Examples

```
# closed cohort design as in Example 12.5
swd_1trt_power(alpha = 0.05, m = 30, J = 5, K = 5, b = 1, d = 0.4,
               rho = 0.05, rho_c = 0.2, rho_s = 0.4)
# repeated cross sectional design as in Example 12.5
swd_1trt_power(alpha = 0.05, m = 30, J = 5, K = 5, b = 1, d = 0.4,
               rho = 0.05, rho_c = 0.2, rho_s = 0)
```

swd_1trt_ss *Stepped Wedge Design Sample Size Calculation with 1 Treatment*

Description

Calculates sample size requirements for stepped wedge design with one treatment and one control condition.

Usage

```
swd_1trt_ss(alpha, power, m, K, b, d, rho, rho_c, rho_s)
```

Arguments

alpha	The significance level or type 1 error rate
power	The specified level of power
m	The number of individuals in each cluster at each time period
K	The number of steps (if one baseline period, then this is periods - 1)
b	The number of baseline periods
d	The standardized effect size
rho	Intraclass correlation coefficient (ICC)
rho_c	Cluster autocorrelation (proportion of cluster-level variance that is time invariant)
rho_s	Subject autocorrelation (This is 0 in repeated cross sectional designs because people are measured only once)

Value

Calculated number of clusters required, adjusted number of clusters, and suggested number of clusters that includes the value of K

Examples

```
# Repeated Cross Sectional Stepped Wedge Sample Size Calculation
# with 3 steps, 1 baseline period, 80% power, 0.05 significance, 30 ppl per
# cluster, effect size of 0.4, ICC of 0.05, and cluster autocorrelation of 0.2
# Following Example 12.6 in text
swd_1trt_ss(alpha = 0.05, power = 0.8, m = 30, K = 3, b = 1, d = 0.4,
            rho = 0.05, rho_c = 0.2, rho_s = 0)
```

`tandz_ratio` *Ratio Evaluating Accuracy of Normal Approximation for T Distribution in Sample Size Calculations*

Description

This function calculates the ratio between (t_(nu, 1-beta) + t_(nu, 1 - alpha / 2))^2 / (z_(1-beta) + z_(1-alpha/2))^2. Sample size requirements using a normal distribution with include (z_(1-beta) + z_(1-alpha/2))^2 as a term, and sample sizes using the t distribution will include (t_(nu, 1-beta) + t_(nu, 1 - alpha / 2))^2 as a term. The ratio of these two terms provides the discrepancy between the normal approximation and t distribution. Ratio values that exceed 1 indicate the normal approximation is underestimating the required sample size. Sample size requirements can be adjusted based on this ratio such that adjusted N = (unadjusted N) * ratio.

Usage

```
tandz_ratio(alpha, power, df)
```

Arguments

`alpha`	The level of precision or type 1 error
`power`	The level of desired power
`df`	Degrees of freedom

Value

Ratio that can be multiplied to normal approximation based sample size calculations.

Examples

```
# Suppose we calculated we needed 80 people using a normal approximation for
# 80% power at 0.05 alpha, then we can update the sample size calculation as:
original_ss <- 80
ratio <- tandz_ratio(alpha = 0.05, power = 0.8, df = 20)
N_adjusted <- ceiling(original_ss * ratio)
N_adjusted
```

Index