Package 'testandroll.pkg'

July 18, 2021

Type Package

Title What the Package Does (Title Case)

Version 0.1.0

Author Who wrote it

Maintainer The package maintainer <yourself@somewhere.net>

Description More about what it does (maybe more than one line)
Use four spaces when indenting paragraphs within the Description.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 2.10)

R topics documented:

`error_rate_nn` *Computes the rate of incorrect deployments with symmetric normal priors Where response is normal with symmetric normal priors*

Description

Computes the rate of incorrect deployments with symmetric normal priors Where response is normal with symmetric normal priors

Usage

```
error_rate_nn(n, s, sigma)
```

Arguments

`n`	vector of length 2 containing the sample sizes
`s`	vector of length 1 (symmetrical) containing the standard deviations of the outcome
`sigma`	vector of length 1 (symmetrical) containing the standard deviations of the prior on the mean response

Value

rate of incorrect deployments

Examples

```
error_rate_nn(n=100, s=.5, sigma=.2)
error_rate_nn(n=c(100,200), s=.5, sigma=.2)
```

`months.means` *Sample data means*

Description

The average proportion of customers that stayed on the website for longer than 5 minutes each month.

Usage

```
months.means
```

Format

A vector of length 12

months.sd — *Sample data standard deviations*

Description

The standard deviations of the proportion of customers that stayed on the website for longer than 5 minutes each month.

Usage

```
months.sd
```

Format

A vector of length 12

one_rep_profit — *Utility function used in function 'profit_nn_sim()' to simulate one set of potential outcomes Draws a true mean for each arm and generates N observations from each arm Returns profits and error rates under perfect information, test & roll, and Thomson sampling*

Description

Utility function used in function 'profit_nn_sim()' to simulate one set of potential outcomes Draws a true mean for each arm and generates N observations from each arm Returns profits and error rates under perfect information, test & roll, and Thomson sampling

Usage

```
one_rep_profit(n, N, s, mu, sigma, K, TS = FALSE)
```

Arguments

n	vector of length K containing the sample sizes
N	the deployment population
s	vector of length K containing the standard deviations of the outcome
mu	vector of length K containing the means of the prior on the mean response
sigma	vector of length K containing the standard deviations of the prior on the mean response
K	the number of arms (treatments)
TS	whether or not to run Thomson sampling, 'TRUE' or 'FALSE'

Value

The profits and error rates under perfect information, test & roll, and Thomson sampling

Examples

```
one_rep_profit(n=c(100,100), N=1000, s=c(.1,.1), mu=c(.1,.1), sigma=c(.05,.05), K=2, TS=FALSE)
one_rep_profit(n=c(100,200,300), N=1000, s=c(.1,.2,.3), mu=c(.1,.2,.3), sigma=c(.01,.03,.05), K=3, TS=FALSE)
```

`one_rep_test_size` *Utility function used in test_size_nn_sim() to simulate one set of potential outcomes Returns profits for all possible equal sample sizes*

Description

Utility function used in test_size_nn_sim() to simulate one set of potential outcomes Returns profits for all possible equal sample sizes

Usage

```
one_rep_test_size(n_vals, N, s, mu, sigma, K)
```

Arguments

N	deployment population
s	standard deviations of the outcome
mu	means of the priors on the mean response
sigma	standard deviations of the priors on the mean response
K	number of arms (treatments)

Value

A 2-column matrix with values of n in the first column and profits in the second column

Examples

```
one_rep_test_size(c(20,50,100), N=1000, s=c(10,10,10), mu= 20, sigma=10, K=2)
```

`plot_prior_effect_nn` *Plots prior densities against treatment effect (difference in profit per customer)*

Description

Plots prior densities against treatment effect (difference in profit per customer)

Usage

```
plot_prior_effect_nn(mu, sigma, abs = FALSE)
```

Arguments

mu	means of the prior on the mean response
sigma	standard deviations of the prior on the mean response
abs	whether or not to take the absolute difference, 'TRUE' or 'FALSE'

Value

graph plotting prior density vs. difference in profit per customer

Examples

```
plot_prior_effect_nn(mu=c(5,10), sigma=c(10,10), abs=FALSE)
```

`plot_prior_mean_resp_nn` *Plots prior densities against mean response (profit per customer)*

Description

Plots prior densities against mean response (profit per customer)

Usage

```
plot_prior_mean_resp_nn(mu, sigma)
```

Arguments

`mu`	means of the prior on the mean response
`sigma`	standard deviations of the prior on the mean response

Value

graph plotting prior density vs. profit per customer

Examples

```
#plot_prior_mean_resp_nn(mu=c(5,10), sigma=c(10,10))
```

`plot_prior_resp_nn` *Plots prior densities against response (profit per customer)*

Description

Plots prior densities against response (profit per customer)

Usage

```
plot_prior_resp_nn(s, mu, sigma)
```

Arguments

`s`	known standard deviations of the response
`mu`	means of the prior on the mean response
`sigma`	standard deviations of the prior on the mean response

Value

graph plotting prior density vs. profit per customer

Examples

```
plot_prior_resp_nn(s=c(10,20), mu=c(5,10), sigma=c(10,10))
```

`profit_nn` *Computes the per-customer profit for test & roll with 2 arms*

Description

Computes the per-customer profit for test & roll with 2 arms

Usage

```
profit_nn(n, N, s, mu, sigma, log_n = FALSE)
```

Arguments

`n`	vector of length 2 containing the sample sizes
`N`	size of deployment population
`s`	vector of length 2 containing the known standard deviations of the outcome
`mu`	vector of length 2 containing the means of the prior on the mean response
`sigma`	vector of length 2 containing the standard deviations of the prior on the mean response
`log_n`	whether or not log(n) is an input rather than n (to avoid negative solutions), 'TRUE' or 'FALSE'

Value

per-customer profit for N customers

Examples

```
profit_nn(n=100, N=10000, s=.1, mu=c(.7,.5), sigma=c(.2,.2))
profit_nn(n=c(100,200), N=10000, s=.1, mu=c(.7,.5), sigma=c(.2,.2))
```

profit_nn_sim — *Computes the per-customer profit for test & roll with K arms*

Description

Computes the per-customer profit for test & roll with K arms

Usage

```
profit_nn_sim(n, N, s, mu, sigma, K = 2, TS = FALSE, R = 1000)
```

Arguments

n	sample sizes for test & roll
N	deployment population
s	standard deviations of the outcome
mu	means of the priors on the mean response (vector of length 1 or K)
sigma	standard deviations of the priors on the mean response (vector of length 1 or K)
K	number of arms
TS	whether or not to run Thomson sampling, 'TRUE' or 'FALSE'
R	number of simulation replications

Value

A list containing the profit, regret, and error rates

Examples

```
profit_nn_sim(n=100, N=1000, s=10, mu=10, sigma=10, K=2, TS=FALSE, R=100)
profit_nn_sim(n=c(100,200,300), N=1000, s=c(.1,.2,.3), mu=c(.1,.2,.3), sigma=c(.01,.03,.05), K=3, TS=FALSE, R
```

profit_perfect_nn — *Computes per-customer profit with perfect information Where response is normal with symmetric normal priors*

Description

Computes per-customer profit with perfect information Where response is normal with symmetric normal priors

Usage

```
profit_perfect_nn(mu, sigma)
```

Arguments

mu	means of the prior on the mean response
sigma	standard deviations of the prior on the mean response

Value

per-customer profit with perfect information

Examples

```
profit_perfect_nn(mu=.7, sigma=.02)
```

`test_eval_nn` *Provides summary of a test & roll plan*

Description

Provides summary of a test & roll plan

Usage

```
test_eval_nn(n, N, s, mu, sigma)
```

Arguments

`n`	vector of length 2 containing the sample sizes
`N`	deployment population
`s`	known standard deviations of the outcome
`mu`	means of the priors on the mean response
`sigma`	standard deviations of the priors on the mean response

Value

a data frame containing summary statistics such as profit per customer, profits from test phase, error rates, etc.

Examples

```
test_eval_nn(n=c(100,100), N=1000, s=.1, mu=.1, sigma=.05)
test_eval_nn(n=c(100,200), N=1000, s=c(.1,.2), mu=c(.1,.2), sigma=c(.05,.1))
```

`test_size_nht` *Computes the recommended sample sizes for a null hypothesis test*

Description

Computes the recommended sample sizes for a null hypothesis test

Usage

```
test_size_nht(s, d, conf = 0.95, power = 0.8, N = NULL)
```

Arguments

s	vector of length 1 (symmetric) or 2 (asymmetric) indicating response standard deviations
d	minimum detectable difference between treatments
conf	1 - type I error rate
power	1 - type II error rate
N	finite deployment population, if NULL no finite population correction is used

Value

recommended sample sizes as a vector

Examples

```
test_size_nht(s=c(0.5,0.10), d=0.2, conf=0.95, power=0.8, N=NULL)
test_size_nht(s=0.5, d=0.2, conf=0.95, power=0.8, N=NULL)
```

---

test_size_nn *Computes the profit-maximizing test size for test & roll with 2 arms*

Description

Computes the profit-maximizing test size for test & roll with 2 arms

Usage

```
test_size_nn(N, s, mu, sigma)
```

Arguments

N	size of deployment population
s	vector of length 2 containing the standard deviations of the outcome
mu	vector of length 2 containing the means of the prior on the mean response
sigma	

Value

a vector containing the sample sizes

Examples

```
test_size_nn(N=10000, s=.1, mu=c(.7,.7), sigma=c(.05,.05))
test_size_nn(N=10000, s=c(.1,.2), mu=c(.7,.7), sigma=c(.05,.05))
```

`test_size_nn_sim` *Computes the profit-maximizing test size for a multi-armed test & roll*

Description

Computes the profit-maximizing test size for a multi-armed test & roll

Usage

```
test_size_nn_sim(N, s, mu, sigma, K = 2, R = 1000)
```

Arguments

`N`	deployment population
`s`	standard deviations of the response (length 1(symmetric) or K)
`mu`	vector of length K containing means of the priors on the mean response
`sigma`	vector of length K containing the standard deviations of the priors on the mean response
`K`	number of arms (treatments)
`R`	number of simulation replications

Value

a list with the sample sizes and expected profit per customer

Examples

```
test_size_nn_sim(N=1000, s=.1, mu=.1, sigma=.05, K=2, R=1000)
test_size_nn_sim(N=1000, s=c(.1,.2,.3), m=c(.1,.2,.3), sigma=c(.05,.08,.1), K=3, R=1000)
```

Index