Package 'bandit'

October 12, 2022

Type Package

Title Functions for Simple a/B Split Test and Multi-Armed Bandit
Analysis

Version 0.5.1

Date 2022-06-028

Imports boot, gam (>= 1.09)

Author Thomas Lotze and Markus Loecher

Maintainer Markus Loecher <markus.loecher@gmail.com>

Description A set of functions for doing analysis of A/B split test data and web metrics in general.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-29 11:20:12 UTC

R topics documented:

---

bandit-package   *Functions for simple A/B split test and multi-armed bandit analysis*

Description

A set of functions for doing analysis of A/B split test data and web metrics in general.

Details

Package:	bandit
Type:	Package
Title:	Functions for simple A/B split test and multi-armed bandit analysis
Version:	0.5.0
Date:	2014-05-03
Imports:	boot, gam (>= 1.09)
Author:	Thomas Lotze and Markus Loecher
Maintainer:	Thomas Lotze <thomaslotze@thomaslotze.com>
License:	GPL-3

Author(s)

Thomas Lotze and Markus Loecher

---

best_binomial_bandit   *best_binomial_bandit*

Description

Compute the Bayesian probabilities for each arm being the best binomial bandit.

Usage

```
best_binomial_bandit(x, n, alpha=1, beta=1)
```

Arguments

x	as in prop.test, a vector of the number of successes
n	as in prop.test, a vector of the number of trials
alpha	shape parameter alpha for the prior beta distribution.
beta	shape parameter beta for the prior beta distribution.

Value

a vector of probabilities for each arm being the best binomial bandit; this can be used for future randomized allocation

Author(s)

Thomas Lotze <thomaslotze@thomaslotze.com> and Markus Loecher

References

Steven L. Scott, A modern Bayesian look at the multi-armed bandit, Appl. Stochastic Models Bus. Ind. 2010; 26:639-658. (<http://www.economics.uci.edu/~ivan/asmb.874.pdf>)

See Also

[prop.test](#)

Examples

```
x=c(10,20,30,50)
n=c(100,102,120,130)
arm_probabilities = best_binomial_bandit(x,n)
print(arm_probabilities)
paste("The best arm is likely ", which.max(arm_probabilities), ", with ",
round(100*max(arm_probabilities), 2), " percent probability of being the best.", sep="")

best_binomial_bandit(c(2,20),c(100,1000))

best_binomial_bandit(c(2,20),c(100,1000), alpha = 2, beta = 5)

#quick look at the various shapes of the beta distribution as we change the shape params:
AlphaBeta = cbind(alpha=c(0.5,5,1,2,2),beta=c(0.5,1,3,2,5))
M = nrow(AlphaBeta)
y= matrix(0,100,ncol=M)
x = seq(0,1,length=100)
for (i in 1:M) y[,i] = dbeta(x,AlphaBeta[i,1],AlphaBeta[i,2])
matplot(x,y,type="l", ylim = c(0,3.5), lty=1, lwd=2)
param_strings = paste("a=", AlphaBeta[,"alpha"], ", b=", AlphaBeta[,"beta"], sep="")
legend("top", legend = param_strings, col=1:M, lty=1)
```

---

best_binomial_bandit_sim

best_binomial_bandit_sim

Description

Compute the Bayesian probabilities for each arm being the best binomial bandit, using simulation.

Usage

```
best_binomial_bandit_sim(x, n, alpha = 1, beta = 1, ndraws = 5000)
```

Arguments

x	as in prop.test, a vector of the number of successes
n	as in prop.test, a vector of the number of trials
alpha	shape parameter alpha for the prior beta distribution.
beta	shape parameter beta for the prior beta distribution.
ndraws	number of random draws from the posterior

Value

a vector of probabilities for each arm being the best binomial bandit; this can be used for future randomized allocation

Author(s)

Thomas Lotze and Markus Loecher

References

Steven L. Scott, A modern Bayesian look at the multi-armed bandit, Appl. Stochastic Models Bus. Ind. 2010; 26:639-658.

(<http://www.economics.uci.edu/~ivan/asmb.874.pdf>)

See Also

prop.test

Examples

```
x=c(10,20,30,33)
n=c(100,102,120,130)
best_binomial_bandit_sim(x,n, ndraws=1000)
round(best_binomial_bandit(x,n),3)

best_binomial_bandit_sim(c(2,20),c(100,1000))

best_binomial_bandit_sim(c(2,20),c(100,1000), alpha = 2, beta = 5)

#quick look at the various shapes of the beta distribution as we change the shape params:
AlphaBeta = cbind(alpha=c(0.5,5,1,2,2),beta=c(0.5,1,3,2,5))
M = nrow(AlphaBeta)
y= matrix(0,100,ncol=M)
x = seq(0,1,length=100)
for (i in 1:M) y[,i] = dbeta(x,AlphaBeta[i,1],AlphaBeta[i,2])
matplot(x,y,type="l", ylim = c(0,3.5), lty=1, lwd=2)
```

```
param_strings = paste("a=", AlphaBeta[,"alpha"], ", b=", AlphaBeta[,"beta"], sep="")
legend("top", legend = param_strings, col=1:M, lty=1)
```

---

best_poisson_bandit   *best_poisson_bandit*

Description

Compute the Bayesian probabilities for each arm being the best poisson bandit.

Usage

```
best_poisson_bandit(x, n = NULL)
```

Arguments

x	as in prop.test, a vector of the number of successes; it may alternatively be a list of vectors of the results of each trial, if n is not provided
n	as in prop.test, a vector of the number of trials; if it is not provided, x must be a list of vectors of the results of each trial

Value

a vector of probabilities for each arm being the best poisson bandit; this can be used for future randomized allocation

Author(s)

Thomas Lotze <thomaslotze@thomaslotze.com>

References

Steven L. Scott, A modern Bayesian look at the multi-armed bandit, Appl. Stochastic Models Bus. Ind. 2010; 26:639-658. (<http://www.economics.uci.edu/~ivan/asmb.874.pdf>)

See Also

prop.test

Examples

```
p1 = rpois(100, lambda=10)
p2 = rpois(100, lambda=9)
x = sapply(list(p1, p2), sum)
n = sapply(list(p1, p2), length)
best_poisson_bandit(x,n)
```

---

deseasonalized_trend   *deseasonalized_trend*

Description

A convenience function to analyze a timeseries and return an estimate (via gam, using day of week factors and smoothed timestamp) of whether, after accounting for day-of-week, there is a significant time-based influence and what that influence is.

Usage

```
deseasonalized_trend(df, w=NULL)
```

Arguments

df   a data frame containing timestamp and value entries

w   number of attempts (n for binomial data)

Value

a list with the following items:

pval   pval given by anova on gam, to indicate whether s(timestamp) is significant

smoothed_prediction

                a smoothed prediction over time (on Wednesdays), to give a human-understandable idea of what the change over time has been

Author(s)

Thomas Lotze <thomaslotze@thomaslotze.com>

Examples

```
timestamps = as.numeric(as.POSIXct(seq(as.Date("2012-01-01"),as.Date("2012-05-03"),by=1)))
df=data.frame(timestamp = timestamps, value = rnorm(length(timestamps)))
dt = deseasonalized_trend(df)
if (dt$pval < 0.01) {
  print("Significant time-based factor")
  plot(df$timestamp, dt$smoothed_prediction)
} else {
  print("No significant time-based factor")
}

df=data.frame(timestamp = timestamps,
              value = sapply(timestamps, function(t) {rpois(1, lambda=t-min(timestamps))}))
dt = deseasonalized_trend(df)
if (dt$pval < 0.01) {
  print("Significant time-based factor")
  plot(df$timestamp, dt$smoothed_prediction)
```

```
} else {
  print("No significant time-based factor")
}
```

---

distribution_estimate   *summarize_metrics*

Description

A convenience function to perform overall metric analysis: mean, median, CI.

Usage

```
distribution_estimate(v, successes=NULL, num_quantiles=101, observed=FALSE)
```

Arguments

v	a vector of values to be analyzed (for nonbinary data), or number of trials (for binary data)
successes	number of successes (for binary data)
num_quantiles	number of quantiles to split into
observed	whether to generate the observed distribution (rather than the estimated distribution of the mean); default FALSE

Value

a data frame with the following columns:

quantiles	the estimated quantiles (0,0.01,0.02,...,1) for the mean, using a Beta-binomial estimate of p for binomial data, a bootstrapped quantile distribution for real-valued numbers
x	x values for plotting a lineplot of the estimated distribution
y	y values for plotting a lineplot of the estimated distribution
mids	mid values for plotting a barplot of the estimated distribution
lefts	left values for plotting a barplot of the estimated distribution
rights	right values for plotting a barplot of the estimated distribution
widths	width values for plotting a barplot of the estimated distribution
heights	height values for plotting a barplot of the estimated distribution
probabilities	probabilities indicating how much probability is contained in each barplot

Author(s)

Thomas Lotze <thomaslotze@thomaslotze.com>

Examples

```
metric_list = list(rbinom(n=100,size=1,prob=0.5),
                    rbinom(n=100,size=1,prob=0.7),
                    rpois(n=100, lambda=5))
distribution_estimate(length(metric_list[[1]]), sum(metric_list[[1]]))
distribution_estimate(length(metric_list[[2]]), sum(metric_list[[2]]))
de = distribution_estimate(metric_list[[3]])
plot(de$x, de$y, type="l")
barplot(de$heights, de$widths)
distribution_estimate(metric_list[[3]], observed=TRUE)
```

---

prob_winner   *prob_winner*

Description

Function to compute probability that each arm is the winner, given simulated posterior results

Usage

```
prob_winner(post)
```

Arguments

post   the simulated results from the posterior, provided by sim_post

Author(s)

Thomas Lotze and Markus Loecher

Examples

```
x=c(10,20,30,50)
n=c(100,102,120,130)
betaPost = sim_post(x,n)
prob_winner(betaPost)
```

---

significance_analysis *significance_analysis*

Description

A convenience function to perform overall proportion comparison using prop.test, before doing pairwise comparisons, to see what outcomes seem to be better than others.

Usage

```
significance_analysis(x, n)
```

Arguments

x	as in prop.test, a vector of the number of successes
n	as in prop.test, a vector of the number of trials

Value

a data frame with the following columns:

successes	x
totals	n
estimated_proportion	
	x/n
lower	0.95 confidence interval on the estimated amount by which this alternative outperforms the next-lower alternative
upper	0.95 confidence interval on the estimated amount by which this alternative outperforms the next-lower alternative
significance	p-value for the test that this alternative outperforms the next-lower alternative
order	order, by highest success proportion
best	1 if it is part of the 'highest performing group' – those groups which were not significantly different from the best group
p_best	Bayesian posterior probability that this alternative is the best binomial bandit

Note

This is intended for use in A/B split testing – so sizes of n should be roughly equal. Also, note that alternatives which have the same rank are grouped together for analysis with the 'next-lower' alternative, so you may want to check to see if ranks are equal.

Author(s)

Thomas Lotze <thomaslotze@thomaslotze.com>

See Also

prop.test

Examples

```
x = c(10,20,30,50)
n = c(100,102,120,130)
sa = significance_analysis(x,n)
sa[rev(order(sa$estimated_proportion)), ]

x = c(37,41,30,43,39,30,31,35,50,30)
n = rep(50, length(x))
sa = significance_analysis(x,n)
sa[rev(order(sa$estimated_proportion)), ]

x = c(37,41,30,43,39,30,31,37,50,30)
n = rep(50, length(x))
sa = significance_analysis(x,n)
sa[rev(order(sa$estimated_proportion)), ]
```

---

sim_post   *sim_post*

Description

Simulate the posterior distribution the Bayesian probabilities for each arm being the best binomial bandit

Usage

```
sim_post(x, n, alpha = 1, beta = 1, ndraws = 5000)
```

Arguments

x	as in prop.test, a vector of the number of successes
n	as in prop.test, a vector of the number of trials
alpha	shape parameter alpha for the prior beta distribution.
beta	shape parameter beta for the prior beta distribution.
ndraws	number of random draws from the posterior

Author(s)

Thomas Lotze and Markus Loecher

Examples

```
x=c(10,20,30,50)
n=c(100,102,120,130)
sim_post(x,n)
```

---

summarize_metrics   *summarize_metrics*

Description

A convenience function to perform overall metric analysis: mean, median, CI.

Usage

```
summarize_metrics(v, successes=NULL)
```

Arguments

v	a vector of values to be analyzed (for nonbinary data), or number of trials (for binary data)
successes	number of successes (for binary data)

Value

a list with the following items:

mean	mean
median	median
lower	0.95 confidence interval on the mean
upper	0.95 confidence interval on the mean
num_obs	number of observations of this metric
total	the sum of all values of this metric (mean*num_obs)

Author(s)

Thomas Lotze <thomaslotze@thomaslotze.com>

Examples

```
metric_list = list(rbinom(n=100,size=1,prob=0.5),
                    rbinom(n=100,size=1,prob=0.7),
                    rpois(n=100, lambda=5))
summarize_metrics(length(metric_list[[1]]), sum(metric_list[[1]]))
summarize_metrics(length(metric_list[[2]]), sum(metric_list[[2]]))
summarize_metrics(metric_list[[3]])
```

---

value_remaining   *value_remaining*

Description

Compute the "value_remaining" in the binomial bandits

Usage

```
value_remaining(x, n, alpha = 1, beta = 1, ndraws = 10000)
```

Arguments

x	as in prop.test, a vector of the number of successes
n	as in prop.test, a vector of the number of trials
alpha	shape parameter alpha for the prior beta distribution.
beta	shape parameter beta for the prior beta distribution.
ndraws	number of random draws from the posterior

Value

value_remaining distribution; the distribution of improvement amounts that another arm might have over the current best arm

Author(s)

Thomas Lotze and Markus Loecher

Examples

```
x=c(10,20,30,80)
n=c(100,102,120,240)
vr = value_remaining(x, n)
hist(vr)
best_arm = which.max(best_binomial_bandit(x, n))
# "potential value" remaining in the experiment
potential_value = quantile(vr, 0.95)
paste("Were still unsure about the CvR for the best arm (arm ", best_arm,
"), but whatever it is, one of the other arms might beat it by as much as ",
round(potential_value*100, 4), " percent.", sep="")
```

Index