Package 'amp'

April 6, 2022

Type Package

Title Statistical Test for the Multivariate Point Null Hypotheses

Version 1.0.0

Description A testing framework for testing the multivariate point null hypothesis.
A testing framework described in Elder et al. (2022) <arXiv:2203.01897> to test the multivariate point null hypothesis. After the user selects a parameter of interest and defines the assumed data generating mechanism, this information should be encoded in functions for the parameter estimator and its corresponding influence curve. Some parameter and data generating mechanism combinations have codings in this package, and are explained in detail in the article.

License GPL (>= 3)

Encoding UTF-8

BugReports https://github.com/adam-s-elder/amp/issues

Imports MASS, stats, SuperLearner, glmnet, dplyr, rlang, ggplot2, tidyr

RoxygenNote 7.1.2

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Adam Elder [aut, cre] (<https://orcid.org/0000-0003-1665-2639>), Marco Carone [ths] (<https://orcid.org/0000-0003-2106-0953>), Alex Luedtke [ths] (<https://orcid.org/0000-0002-9936-3236>)

Maintainer Adam Elder <shmelder@gmail.com>

Repository CRAN

Date/Publication 2022-04-06 07:32:30 UTC

R topics documented:

`accept_rate` *Estimate the local acceptance rate*

Description

This a helper function used to estimate the acceptance rate for a simple norm based test under a sequence of local alternatives in a given direction given the estimated limiting distribution.

Usage

```
accept_rate(mc_limit_dstr, dir, null_quants, norms_idx = 2, norm_type = "lp")
```

Arguments

`mc_limit_dstr`	MC draws from an estimate of the (centered) limiting distribution. Data where columns correspond to different covariates, and rows are independent observations.
`dir`	a vector in the direction for which we wish to estimate power.
`null_quants`	the cutoff values for the distribution under the null for each lp norm.
`norms_idx`	the index of the norms to be used (ideally integers).
`norm_type`	string indicating the class of norms to select over (sum of squares ("ssq") or lp norms ("lp") are currently supported).

Value

The estimated acceptance rate for the specified norm-based test for a given local alternative.

add_oth_pvals *Add Other p-values*

Description

Add pvalues for the Liu and Xie and Bonferroni based tests using the estimated parameter estimates and corresponding standard errors.

Usage

```
add_oth_pvals(test_result)
```

Arguments

test_result	The test result from mv_pn_test

Value

The same test result object with additional p-values for the Liu and Xie (2021) test (liu_xie_pvalue) and the Bonferroni based test (bonf_pvalue)

Examples

```
## NOTE: More monte-carlo samples should be taken are taken here.  This is
## only done to lower computation time.
set.seed(10)
test <- mv_pn_test(data.frame(y = rnorm(100), x = rnorm(100)),
                   ic.pearson, test.control(n_peld_mc_samples = 20,
                                            ts_ld_bs_samp = 20))
test_with_extra <- add_oth_pvals(test)
test_with_extra[c("pvalue", "liu_xie_pvalue", "bonf_pvalue")]
```

calc_gam_star *A helper function for* mv_pn_test*, calculating the test statistic for both the vector of parameter estimates, and the draws from the corresponding estimated limiting distribution.*

Description

A helper function for mv_pn_test, calculating the test statistic for both the vector of parameter estimates, and the draws from the corresponding estimated limiting distribution.

Usage

```
calc_gam_star(obs_data, param_est, control, lm_dst = NULL, return_lmd = FALSE)
```

Arguments

obs_data	The observed data used to calculate the test statistic. Each row is an observation and each column corresponds to either the outcome (first column) or a covariate.
param_est	A function used to estimate both they parameter of interest and the IC of the corresponding estimator.
control	A list providing control arguments for the function.
lm_dst	A list containing an estimate of the limiting distribution and corresponding norm specific test cutoffs if it is provided.
return_lmd	Boolean for whether to return the estimated limiting distribution.

Value

Calculated test statistic for the given data.

---

est_pows *A helper function to estimate the power using the generated test statistic, and estimated distribution of the test statistic.*

Description

A helper function to estimate the power using the generated test statistic, and estimated distribution of the test statistic.

Usage

```
est_pows(tr_lm_dstr, ts_vec)
```

Arguments

tr_lm_dstr	limiting distribution of the test statistic
ts_vec	Vector containing the test statistic

Value

Return the estimated power of the test based for each of the possible sum of squares norms

find_mag *A helper function to find the multiplicative distance of a specified alternative from the alternative in the same direction that obtains a power of 80%.*

Description

A helper function to find the multiplicative distance of a specified alternative from the alternative in the same direction that obtains a power of 80%.

Usage

```
find_mag(one_obs, dir, cutoff, nrm_idx, nrm_type)
```

Arguments

one_obs	A single observed value from the limiting distribution.
dir	The shift by which the single observation will moved. Generally the estimated parameter.
cutoff	The cutoff value of the normed limiting distribution. The estimated parameters which will be the mean of the multiplier bootstrap sample.
nrm_idx	index of the potential norms to be used.
nrm_type	specifies the type of norm to be used.

Value

The magnitude by which the shift must be multiplied to surpass the cutoff for the given observation.

gen_boot_sample *A helper function to generate a multiplier bootstrap sample*

Description

A helper function to generate a multiplier bootstrap sample

Usage

```
gen_boot_sample(epsilon_mat, obs_ic, center = TRUE, param_est = 0, rate = "n")
```

Arguments

epsilon_mat	The matrix of Normal observations with independent observations from a normal with an identity Covariance matrix.
obs_ic	The empirical estimate of the influence curve to be used in the multiplier bootstrap.
center	Boolean. If true, the bootstrapped data will be centered at zero. Otherwise, it will be centered at param_est.
param_est	The estimated parameters which will be the mean of the multiplier bootstrap sample.
rate	Normalizing constant. Should either be "n" or "rootn".
n	Number of desired observations from your bootstrapped sample.

Value

A sample of size n generated using a multiplier bootstrap with a variance given by t(obs_ic)

---

get_test_stat *Helper function for the Zhang and Laber test.*

Description

Helper function for the Zhang and Laber test.

Usage

```
get_test_stat(obs_data)
```

Arguments

obs_data	The observed data

Value

Returns the estimated t-statistics for the Zhang and Laber tests.

`ic.data.examp` *Function for calculating the influence function used for the real data example.*

Description

Function for calculating the influence function used for the real data example.

Usage

```
ic.data.examp(obs_data, what = "both", control = NULL)
```

Arguments

`obs_data`	the observed data. The first column should be the outcome.
`what`	the desired return value. Should be one of `"ic"` (influence curve), `"est"` (estimate), or `"both"`.
`control`	any other control parameters to be passed to the estimator.

Value

If `what` is

- `"est"`, then return the estimated parameter.

- `"ic"`, then return the estimated IC of the parameter estimate.

- `"both"`, then return both the parameter estimate and corresponding estimated IC.

Examples

```
expit <- function(x) exp(x) / (1 + exp(x))
ws <- matrix(rnorm(3000), ncol = 3)
probs <- expit(ws  %*% c(-1, 0, 2))
y <- rbinom(n = nrow(probs), size = 1, prob = probs[, 1])
wts <-   abs(rnorm(length(y))) + 1
wts <- length(wts) * wts / sum(wts)
cats <- rep(1:10, 100)
obs_dat <- cbind(y, "cat" = cats, "wt" = wts, ws)
est_ic <- ic.data.examp(obs_dat, what = "both")
my_est <- est_ic$est
my_ic <- est_ic$ic / nrow(ws)
var_mat <- t(my_ic) %*% my_ic
sqrt(diag(var_mat))
for(cov_idx in 1:ncol(ws)){
 print(summary(stats::glm(y ~ ws[, cov_idx], weights = obs_dat[, "wt"],
                    family = binomial))$coefficients[2, 1:2])
}
```

`ic.pearson` *Pearson Correlation IC and estimate*

Description

This function takes a set of observations, and returns an estimate and its corresponding estimated IC matrix for the estimates of the pearson correlation. Estimates of the covariance are generated using the empirical influence function. The first column of your data should correspond to the variable of interest (the variable for which pearson correlation is calculated).

Usage

```
ic.pearson(observ, what = "both", control = NULL)
```

Arguments

`observ`	the observed data. The first column should be the outcome.
`what`	the desired return value. Should be one of `"ic"` (influence curve), `"est"` (estimate), or `"both"`.
`control`	any other control parameters to be passed to the estimator.

Value

If `what` is:

- `"est"`, then return the estimated person correlation.

- `"ic"`, then return the estimated IC of the person correlation estimate.

- `"both"`, then return both the estimated pearson correlation and the estimated IC of the person correlation estimate.

Examples

```
dat <- matrix(rnorm(80), nrow = 20)
ic.pearson(dat, what = "both")
## Note that the estimate is the same as what is found using \code{cor}
cor(dat)[1, ]
```

`ic.proj.rr` *Estimate both the parameter, and the influence curves used for estimating the projected risk ratio. The first column of your data should correspond to the variable of interest.*

Description

Estimate both the parameter, and the influence curves used for estimating the projected risk ratio. The first column of your data should correspond to the variable of interest.

Usage

```
ic.proj.rr(obs_data, what = "both", control = NULL)
```

Arguments

`obs_data`	the observed data. The first column should be the outcome.
`what`	the desired return value. Should be one of `"ic"` (influence curve), `"est"` (estimate), or `"both"`.
`control`	any other control parameters to be passed to the estimator.

Value

If `what` is

- `"est"`, then return the estimated parameter.

- `"ic"`, then return the estimated IC of the parameter estimate.

- `"both"`, then return both the parameter estimate and corresponding estimated IC.

Examples

```
# not run (make sure to load in SuperLearner if running)
# set.seed(1010)
# fake_dat <- data.frame(y = rbinom(100, size = 1, prob = 0.5),
#                        delta = rbinom(100, size =  1, prob = 0.5),
#                       w = matrix(rnorm(500), ncol = 5))
# ic.proj.rr.nolas(fake_dat)
```

`ic.proj.rr.nolas` *Estimate both the parameter, and or the influence curves used for estimating the projected risk ratio (not using lasso). The first column of your data should correspond to the variable of interest.*

Description

Estimate both the parameter, and or the influence curves used for estimating the projected risk ratio (not using lasso). The first column of your data should correspond to the variable of interest.

Usage

```
ic.proj.rr.nolas(obs_data, what = "both", control = NULL)
```

Arguments

`obs_data`	the observed data. The first column should be the outcome.
`what`	the desired return value. Should be one of `"ic"` (influence curve), `"est"` (estimate), or `"both"`.
`control`	any other control parameters to be passed to the estimator.

Value

If `what` is

- `"est"`, then return the estimated parameter.

- `"ic"`, then return the estimated IC of the parameter estimate.

- `"both"`, then return both the parameter estimate and corresponding estimated IC.

Examples

```
# not run (make sure to load in SuperLearner if running)
# set.seed(1010)
# fake_dat <- data.frame(y = rbinom(100, size = 1, prob = 0.5),
#                        delta = rbinom(100, size =  1, prob = 0.5),
#                       w = matrix(rnorm(500), ncol = 5))
# ic.proj.rr.nolas(fake_dat)
```

look_IC *This a function used to look at the IC.*

Description

This a function used to look at the IC.

Usage

```
look_IC(IC)
```

Arguments

IC	an influence curve evaluated at each observation

Value

Histogram of the IC at each obeservation for each covariate. Vertical lines indicate the mean of the IC.

Examples

```
set.seed(20)
ic <- amp::ic.pearson(matrix(rnorm(120), ncol = 6))$ic
look_IC(ic)
```

l_p_norm *A function used to calculate various L_p norms*

Description

A function used to calculate various L_p norms

Usage

```
l_p_norm(x, p = "max", type = "lp")
```

Arguments

x	Observed data
p	index of the norm
type	Kind of norm used (currently only lp and sum of squares norms are supported)

Value

The norm of x, of type type of index p. For example, the euclidean norm x has p = 2, type = "lp"

Examples

```
x <- c(3, 4)
l_p_norm(x, p = 2, type = "lp")
l_p_norm(x, p = 2, type = "ssq")
l_p_norm(x, p = "max", type = "lp")
l_p_norm(x, p = 1, type = "ssq")

y <- c(3, 4, 5, 6)
l_p_norm(y, p = 4, type = "lp")
l_p_norm(y, p = 3, type = "ssq")
```

`mag_for_pow` *This function is used to estimate the magnitude needed for a certain direction to achieve 80% power for a proposed alternative.*

Description

This function is used to estimate the magnitude needed for a certain direction to achieve 80% power for a proposed alternative.

Usage

```
mag_for_pow(mc_limit_dstr, dir, norms_idx = 2, null_quants, norm_type = "lp")
```

Arguments

`mc_limit_dstr`	MC draws from the estimated (centered) limiting distribution. Columns correspond to different covariates, and rows are independent observations.
`dir`	the direction for which we wish to find the magnitude needed to achieve 80% power.
`norms_idx`	the norms to be considered.
`null_quants`	ninety five percent quantiles for each of the different norms
`norm_type`	the class of norms to be used

Value

Magnitude for the specified lp norm for a given local alternative.

`make_data` *Generate data using one of the four specified models from MCKEAGUE and QIAN paper*

Description

Generate data using one of the four specified models from MCKEAGUE and QIAN paper

Usage

```
make_data(ss, dim, rho, model = 1, b = NULL)
```

Arguments

`ss`	sample size (number of generated observations)
`dim`	dimension of the generated data
`rho`	between x correlation
`model`	number indicating which model should be used
`b`	local alternative to be used

Value

A dataframe with outcome in the first column and covariates in the others. Data are generated according to the models described originally in McKeague and Qian (2015).

Examples

```
# Sample data
null_dat <- make_data(ss = 100, dim = 10, rho = 0, model = 1)
model_two_dat <- make_data(ss = 100, dim = 10, rho = 0, model = 2)
model_three_dat <- make_data(ss = 100, dim = 10, rho = 0, model = 2)
```

`mv_pn_test` *Runs a multivariate point null test. This function returns an approximate p-value for the specified test statistic.*

Description

Runs a multivariate point null test. This function returns an approximate p-value for the specified test statistic.

Usage

```
mv_pn_test(obs_data, param_est = NULL, control = test.control())
```

Arguments

obs_data	The observed data to be used for finding the optimal norm (training), and finding the test statistic (testing). Similar to above, each row is an observation and each column corresponds to either the outcome (first column) or a covariate.
param_est	Function used to estimate the parameter and corresponding influence curve.
control	List used to define controls for test.

Value

The test will always return the following output:

- pvalue: The approximate value of the test statistic

- test_stat: The approximate value of the test statistic

- test_st_eld: The approximate limiting distribution of the test statistic (with length equal to ts_ld_bs_samp).

- chosen_norm: A vector indicating which norm was chosen by the adaptive test

- param_ests: The parameter estimate.

- param_ses: An estimate of the standard error of each element of param_ests

- oth_ic_inf: Any other information provided by the param_est function when calculating the IC and parameter estimates. Additional information may be returned by specifying it in the test.control function:

- If "var_est" is contained in other_output, the test output will contain will have var_mat returned which is the empirical second moment of the IC (equal asymptotically to the variance estimator).

- If "obs_data" is contained in the other_output, the test output will return the data passed to the testing function.

Examples

```
set.seed(10)
## NOTE: More monte-carlo samples should be taken are taken here.  This is
## only done to lower computation time.
test <- mv_pn_test(data.frame(y = rnorm(100), x = rnorm(100)),
                   ic.pearson, test.control(n_peld_mc_samples = 20,
                                            ts_ld_bs_samp = 20))
```

---

pval_for_mag *A helper function that calculates the estimated p-value for a given observed alternative and a given norm.*

Description

A helper function that calculates the estimated p-value for a given observed alternative and a given norm.

Usage

```
pval_for_mag(mc_limit_dstr, dir, norms_idx = 2, norm_type = "lp", ...)
```

Arguments

mc_limit_dstr	the simulated data draw from the limiting distribution under the null
dir	the observed estimate of the parameter
norms_idx	the index for the norm used
norm_type	the type of norm used
...	additional arguments that may be passed to pval_for_mag, but which will be ignored.

Value

The p-value for a test with estimate given by `dir` and limiting distribution given by `mc_limit_dstr`.

---

rr.msm.ic *Estimate both the parameter, and the influence curves used for estimating the projected risk ratio from a working marginal structural mean model.*

Description

The function takes a dataset, and will returns the estimate for the parameter, and or the estimated influence curve at each observation. The first column of obs_data should be the binary outcome of interest.

Usage

```
rr.msm.ic(obs_data, what = "both", control = NULL)
```

Arguments

obs_data	the observed data. The first column should be the outcome.
what	the desired return value. Should be one of ‘"ic"‘ (influence curve), ‘"est"‘ (estimate), or ‘"both"‘.
control	any other control parameters to be passed to the estimator.

Details

Most of this code has been copied and slightly modified from the ltmle package and is copyrighted by Joshua Schwab under the terms of the GPL-2 license.

Obtain an estimator of the probability delta = 1 given w

Value

If ‘what‘ is

- ‘"est"‘, then return the estimated parameter.

- ‘"ic"‘, then return the estimated IC of the parameter estimate.

- ‘"both"‘, then return both the parameter estimate and corresponding estimated IC.

Examples

```
#not run (make sure to load in SuperLearner if running)
#set.seed(1010)
#fake_dat <- data.frame(y = rbinom(100, size = 1, prob = 0.5),
#                       a = rbinom(100, size =  1, prob = 0.5),
#                        w = matrix(rnorm(500), ncol = 5))
# rr.msm.ic(fake_dat)
```

`rr.msm.jn.ic` *Estimate both the parameter, and the influence curves used for estimating the projected risk ratio from a working marginal structural mean model. This function only uses elastic net for the estimation (rather than also using other learners).*

Description

The function takes a dataset, and will returns the estimate for the parameter, and or the estimated influence curve at each observation. The first column of obs_data should be the binary outcome of interest.

Usage

```
rr.msm.jn.ic(obs_data, what = "both", control = NULL)
```

Arguments

`obs_data`	the observed data. The first column should be the outcome.
`what`	the desired return value. Should be one of ‘"ic"‘ (influence curve), ‘"est"‘ (estimate), or ‘"both"‘.
`control`	any other control parameters to be passed to the estimator.

Details

Most of the code for this function has been copied and slightly modified from the ltmle package and is copyrighted by Joshua Schwab under the terms of the GPL-2 license.

Obtain an estimator of the probability delta = 1 given w

Value

If ‘what‘ is

- ‘"est"‘, then return the estimated parameter.

- ‘"ic"‘, then return the estimated IC of the parameter estimate.

- ‘"both"‘, then return both the parameter estimate and corresponding estimated IC.

Examples

```
#not run (make sure to load in SuperLearner if running)
# set.seed(1010)
# fake_dat <- data.frame(y = rbinom(100, size = 1, prob = 0.5),
#                        delta = rbinom(100, size =  1, prob = 0.5),
#                        w = matrix(rnorm(500), ncol = 5))
# rr.msm.jn.ic(fake_dat)
```

`test.control` *Control function for the adaptive norm test*

Description

Control function for the adaptive norm test

Usage

```
test.control(
  n_peld_mc_samples = 300,
  nrm_type = "lp",
  perf_meas = "est_acc",
  pos_lp_norms = c(1, 2, 3, "max"),
  ld_est_meth = "par_boot",
  ts_ld_bs_samp = 250,
  other_output = c(),
  ...
)
```

Arguments

`n_peld_mc_samples`	Number of samples to be used in approximating the estimated limiting distribution of the parameter estimate under the null. Increasing this value reduces the approximation error of the test statistic.
`nrm_type`	The type of norm to be used for the test. Generally the l_p norm
`perf_meas`	the preferred measure used to generate the test statistic.
`pos_lp_norms`	The index of the norms to be considered. For example if we use the l_p norm, norms_indx specifies the different p’s to try.

ld_est_meth	String indicating method for estimating the limiting distribution of the test statistic parametric bootstrap or permutation.
ts_ld_bs_samp	The number of test statistic limiting distribution bootstrap samples to be drawn.
other_output	A vector indicating additional data that should be returned. Currently only "var_est" and data is supported.
...	Other arguments needed in other places.

Value

A list that provide controls for mv_pn_test (specified by the arguments passed to test.control).

Examples

```
test.control()
```

ZL *Carry out a simplified version of the Zhang and Laber test.*

Description

Carry out a simplified version of the Zhang and Laber test.

Usage

```
ZL(observed_data, ts_sims, ld_sims)
```

Arguments

observed_data	The observed data.
ts_sims	The number of draws from the test statistic distribution
ld_sims	The number of draws from the limiting distribution to estimate each test statistic.

Value

A list containing:

- pvalue: The p-value of the test
- test_stat: The corresponding test statistic
- test_st_eld: The corresponding estimated limiting distribution of the test statistic.
- param_ests: The estimates of the measure of association (correlation)
- param_ses: The corresponding standard errors of the estimates

Examples

```
ZL(data.frame(z = rnorm(100), x1 = rnorm(100), x2 = rnorm(100)), 10, 10)
```

Index