Package ‘jointVIP’

November 22, 2024

Title Prioritize Variables with Joint Variable Importance Plot in Observational Study Design

Version 1.0.0

Description In the observational study design stage, matching/weighting methods are conducted. However, when many background variables are present, the decision as to which variables to prioritize for matching/weighting is not trivial. Thus, the joint treatment-outcome variable importance plots are created to guide variable selection. The joint variable importance plots enhance variable comparisons via unadjusted bias curves derived under the omitted variable bias framework. The plots translate variable importance into recommended values for tuning parameters in existing methods. Post-matching and/or weighting plots can also be used to visualize and assess the quality of the observational study design. The method motivation and derivation is presented in ``Prioritizing Variables for Observational Study Design using the Joint Variable Importance Plot'' by Liao et al. (2024) <doi:10.1080/00031305.2024.2303419>. See the package paper by Liao and Pimentel (2024) <doi:10.21105/joss.06093> for a beginner friendly user introduction.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 3.3)

Suggests causaldata, devtools (>= 2.4.5), knitr, MatchIt, WeightIt, optmatch, optweight (>= 0.2.4), rmarkdown (>= 2.18), testthat (>= 3.0.0), stringr

Config/testthat/edition 3

Collate 'data.R' 'support.R' 'check_measures.R' 'create_jointVIP.R' 'create_post_jointVIP.R' 'get_measures.R' 'get_post_measures.R' 'get_boot_measures.R' 'plot.R' 'print.R' 'summary.R'

Imports ggrepel (>= 0.9.2), ggplot2 (>= 3.4.0)

VignetteBuilder knitr

URL `https://github.com/ldliao/jointVIP`, `https://ldliao.github.io/jointVIP/`

BugReports https://github.com/ldliao/jointVIP/issues

LazyData true

NeedsCompilation no

Author Lauren D. Liao [aut, cre] (<https://orcid.org/0000-0003-4697-6909>), Samuel D. Pimentel [aut] (<https://orcid.org/0000-0002-0409-6586>)

Maintainer Lauren D. Liao <ldliao@berkeley.edu>

Repository CRAN

Date/Publication 2024-11-22 03:10:02 UTC

Contents

`add_bias_curves` *support function to plot bias curves*

Description

support function to plot bias curves

Usage

```
add_bias_curves(p, ...)
```

Arguments

`p`	plot made with jointVIP object
`...`	encompasses other variables needed

Value

a joint variable importance plot of class `ggplot` with curves

`add_variable_labels` *support function to plot variable text labels*

Description

support function to plot variable text labels

Usage

```
add_variable_labels(p, ...)
```

Arguments

`p`	plot made with jointVIP object
`...`	encompasses other variables needed

Value

a joint variable importance plot of class `ggplot` with curves

`bootstrap.plot` *plot the bootstrap version of the jointVIP object*

Description

plot the bootstrap version of the jointVIP object

Usage

```
bootstrap.plot(
  x,
  ...,
  smd = "cross-sample",
  use_abs = TRUE,
  plot_title = "Joint Variable Importance Plot",
  B = 100
)
```

Arguments

`x`	a jointVIP object
`...`	custom options: `bias_curve_cutoffs`, `text_size`, `max.overlaps`, `label_cut_std_md`, `label_cut_outcome_cor`, `label_cut_bias`, `bias_curves`, `add_var_labs`
`smd`	specify the standardized mean difference is `cross-sample` or `pooled`
`use_abs`	TRUE (default) for absolute measures
`plot_title`	optional string for plot title
`B`	100 (default) for the number of times the bootstrap step wished to run

Value

a joint variable importance plot of class `ggplot`

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
covariates = names(analysis_df)[!names(analysis_df)
                                %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)
# more bootstrap number B would be typically used in real settings
# this is just a small example
set.seed(1234567891)
bootstrap.plot(new_jointVIP, B = 15)
```

`brfss` *2015 Behavioral Risk Factor Surveillance System*

Description

A subset of data from the Centers for Disease Control and Prevention 2015 Behavioral Risk Factor Surveillance System (BRFSS) Survey

Usage

```
brfss
```

Format

`brfss`**:**

A data frame with 5,000 rows and 17 columns:

COPD Chronic obstructive pulmonary disease

smoke Smoke

sex Sex

weight Weight

average_drinks Average drinks answers to: during the past 30 days, when you drank, how many drinks did you drink on average?

race_white, race_black, race_hispanic, race_other Race group

age_18to24, age_25to34, age_35to44, age_45to54, age_55to64, age_over65 Age groups

Source

`http://static.lib.virginia.edu/statlab/materials/data/brfss_2015_sample.csv`

`ceiling_dec` *support function for ceiling function with decimals*

Description

support function for ceiling function with decimals

Usage

```
ceiling_dec(num, dec_place = 1)
```

Arguments

`num`	numeric
`dec_place`	decimal place that is desired ceiling for

Value

numeric number desired

`check_measures` *Check measures Check to see if there is any missing values or variables without any variation or identical rows (only unique rows will be used)*

Description

Check measures Check to see if there is any missing values or variables without any variation or identical rows (only unique rows will be used)

Usage

```
check_measures(measures)
```

Arguments

`measures`	measures needed for jointVIP

Value

measures needed for jointVIP

`create_jointVIP` *create jointVIP object*

Description

This is creates the jointVIP object & check inputs

Usage

```
create_jointVIP(treatment, outcome, covariates, pilot_df, analysis_df)
```

Arguments

`treatment`	string denoting the name of the binary treatment variable, containing numeric values: 0 denoting control and 1 denoting treated
`outcome`	string denoting the name of a numeric outcome variable
`covariates`	vector of strings or list denoting column names of interest
`pilot_df`	data.frame of the pilot data; character and factor variables are automatically one-hot encoded
`analysis_df`	data.frame of the analysis data; character and factor variables are automatically one-hot encoded

Value

a jointVIP object

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
covariates = names(analysis_df)[!names(analysis_df)
                                %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)
```

`create_post_jointVIP` *create post_jointVIP object*

Description

This is creates the post_jointVIP object & check inputs

Usage

```
create_post_jointVIP(object, post_analysis_df, wts = NA)
```

Arguments

`object`	a jointVIP object
`post_analysis_df`	post matched or weighted data.frame
`wts`	user-supplied weights

Value

a post_jointVIP object (subclass of jointVIP)

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
covariates = names(analysis_df)[!names(analysis_df)
                                %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)

## at this step typically you may wish to do matching or weighting
## the results after can be stored as a post_data
## the post_data here is not matched or weighted, only for illustrative purposes
post_data <- data.frame(year = rnorm(50, 200, 5),
                        pop = rnorm(50, 1000, 500),
                        gdpPercap = runif(50, 100, 1000),
                        trt = rbinom(50, 1, 0.5),
                        out = rnorm(50, 1, 0.2))
post_dat_jointVIP =  create_post_jointVIP(new_jointVIP, post_data)
```

`floor_dec` *support function for floor function with decimals*

Description

support function for floor function with decimals

Usage

```
floor_dec(num, dec_place = 1)
```

Arguments

`num`	numeric
`dec_place`	decimal place that is desired floor for

Value

numeric number desired

`get_boot_measures` *Calculate bootstrapped variation additional tool to help calculate the uncertainty of each variable's bias*

Description

Calculate bootstrapped variation additional tool to help calculate the uncertainty of each variable's bias

Usage

```
get_boot_measures(object, smd = "cross-sample", use_abs = TRUE, B = 100)
```

Arguments

`object`	jointVIP object
`smd`	calculate standardized mean difference either using `cross-sample` or `pooled`
`use_abs`	TRUE (default) for absolute measures
`B`	100 (default) for the number of times the bootstrap step wished to run

Value

bootstrapped measures needed for bootstrap-jointVIP

`get_measures` *Prepare data frame to plot standardized omitted variable bias Marginal standardized mean differences and outcome correlation*

Description

Prepare data frame to plot standardized omitted variable bias Marginal standardized mean differences and outcome correlation

Usage

```
get_measures(object, smd = "cross-sample")
```

Arguments

`object`	jointVIP object
`smd`	calculate standardized mean difference either using `cross-sample` or `pooled`

Value

measures needed for jointVIP

`get_post_measures` *Post-measures data frame to plot post-standardized omitted variable bias*

Description

Post-measures data frame to plot post-standardized omitted variable bias

Usage

```
get_post_measures(object, smd = "cross-sample")
```

Arguments

`object`	post_jointVIP object
`smd`	calculate standardized mean difference either using `cross-sample` or `pooled`

Value

measures needed for jointVIP

`one_hot` *support function for one-hot encoding*

Description

support function for one-hot encoding

Usage

```
one_hot(df)
```

Arguments

`df`	data.frame object for performing one-hot encoding

Value

data.frame object with factor variables one-hot encoded for each level

---

`plot.jointVIP` *plot the jointVIP object*

Description

plot the jointVIP object

Usage

```
## S3 method for class 'jointVIP'
plot(
  x,
  ...,
  smd = "cross-sample",
  use_abs = TRUE,
  plot_title = "Joint Variable Importance Plot"
)
```

Arguments

`x`	a jointVIP object
`...`	custom options: `bias_curve_cutoffs`, `text_size`, `max.overlaps`, `label_cut_std_md`, `label_cut_outcome_cor`, `label_cut_bias`, `bias_curves`, `add_var_labs`, `expanded_y_curvelab`
`smd`	specify the standardized mean difference is `cross-sample` or `pooled`
`use_abs`	TRUE (default) for absolute measures
`plot_title`	optional string for plot title

Value

a joint variable importance plot of class `ggplot`

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
covariates = names(analysis_df)[!names(analysis_df)
```

```
                               %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)
plot(new_jointVIP)
```

`plot.post_jointVIP` *plot the post_jointVIP object this plot uses the same custom options as the jointVIP object*

Description

plot the post_jointVIP object this plot uses the same custom options as the jointVIP object

Usage

```
## S3 method for class 'post_jointVIP'
plot(
  x,
  ...,
  smd = "cross-sample",
  use_abs = TRUE,
  plot_title = "Joint Variable Importance Plot",
  add_post_labs = TRUE,
  post_label_cut_bias = 0.005
)
```

Arguments

`x`	a post_jointVIP object
`...`	custom options: `bias_curve_cutoffs`, `text_size`, `max.overlaps`, `label_cut_std_md`, `label_cut_outcome_cor`, `label_cut_bias`, `bias_curves`, `add_var_labs`, `expanded_y_curvelab`
`smd`	specify the standardized mean difference is `cross-sample` or `pooled`
`use_abs`	TRUE (default) for absolute measures
`plot_title`	optional string for plot title
`add_post_labs`	TRUE (default) show post-measure labels
`post_label_cut_bias`	0.005 (default) show cutoff above this number; suppressed if show_post_labs is FALSE

Value

a post-analysis joint variable importance plot of class `ggplot`

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
covariates = names(analysis_df)[!names(analysis_df)
                                %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)

## at this step typically you may wish to do matching or weighting
## the results after can be stored as a post_data
## the post_data here is not matched or weighted, only for illustrative purposes
post_data <- data.frame(year = rnorm(50, 200, 5),
                        pop = rnorm(50, 1000, 500),
                        gdpPercap = runif(50, 100, 1000),
                        trt = rbinom(50, 1, 0.5),
                        out = rnorm(50, 1, 0.2))
post_dat_jointVIP = create_post_jointVIP(new_jointVIP, post_data)
plot(post_dat_jointVIP)
```

`print.jointVIP` *Obtains a print for jointVIP object*

Description

Obtains a print for jointVIP object

Usage

```
## S3 method for class 'jointVIP'
print(x, ..., smd = "cross-sample", use_abs = TRUE, bias_tol = 0.01)
```

Arguments

`x`	a jointVIP object
`...`	not used

`smd`	specify the standardized mean difference is `cross-sample` or `pooled`
`use_abs`	TRUE (default) for absolute measures
`bias_tol`	numeric 0.01 (default) any bias above the absolute bias_tol will be printed

Value

measures used to create the plot of jointVIP

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
covariates = names(analysis_df)[!names(analysis_df)
                                %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)
print(new_jointVIP)
```

`print.post_jointVIP` *Obtains a print for post_jointVIP object*

Description

Obtains a print for post_jointVIP object

Usage

```
## S3 method for class 'post_jointVIP'
print(x, ..., smd = "cross-sample", use_abs = TRUE, bias_tol = 0.01)
```

Arguments

`x`	a post_jointVIP object
`...`	not used
`smd`	specify the standardized mean difference is `cross-sample` or `pooled`
`use_abs`	TRUE (default) for absolute measures
`bias_tol`	numeric 0.01 (default) any bias above the absolute bias_tol will be printed

Value

measures used to create the plot of jointVIP

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
covariates = names(analysis_df)[!names(analysis_df)
                                %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)

## at this step typically you may wish to do matching or weighting
## the results after can be stored as a post_data
## the post_data here is not matched or weighted, only for illustrative purposes
post_data <- data.frame(year = rnorm(50, 200, 5),
                        pop = rnorm(50, 1000, 500),
                        gdpPercap = runif(50, 100, 1000),
                        trt = rbinom(50, 1, 0.5),
                        out = rnorm(50, 1, 0.2))
post_dat_jointVIP = create_post_jointVIP(new_jointVIP, post_data)
print(post_dat_jointVIP)
```

`summary.jointVIP` *Obtains a summary jointVIP object*

Description

Obtains a summary jointVIP object

Usage

```
## S3 method for class 'jointVIP'
summary(object, ..., smd = "cross-sample", use_abs = TRUE, bias_tol = 0.01)
```

Arguments

`object`	a jointVIP object
`...`	not used
`smd`	specify the standardized mean difference is `cross-sample` or `pooled`
`use_abs`	TRUE (default) for absolute measures
`bias_tol`	numeric 0.01 (default) any bias above the absolute bias_tol will be summarized

Value

no return value

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
covariates = names(analysis_df)[!names(analysis_df)
                                %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)
summary(new_jointVIP)
```

`summary.post_jointVIP` *Obtains a summary post_jointVIP object*

Description

Obtains a summary post_jointVIP object

Usage

```
## S3 method for class 'post_jointVIP'
summary(
  object,
  ...,
  smd = "cross-sample",
  use_abs = TRUE,
  bias_tol = 0.01,
  post_bias_tol = 0.005
)
```

Arguments

`object`	a post_jointVIP object
`...`	not used
`smd`	specify the standardized mean difference is `cross-sample` or `pooled`
`use_abs`	TRUE (default) for absolute measures
`bias_tol`	numeric 0.01 (default) any bias above the absolute bias_tol will be summarized
`post_bias_tol`	numeric 0.005 (default) any bias above the absolute bias_tol will be summarized

Value

no return value

Examples

```
data <- data.frame(year = rnorm(50, 200, 5),
                   pop = rnorm(50, 1000, 500),
                   gdpPercap = runif(50, 100, 1000),
                   trt = rbinom(50, 1, 0.5),
                   out = rnorm(50, 1, 0.2))
# random 20 percent of control as pilot data
pilot_sample_num = sample(which(data$trt == 0),
                          length(which(data$trt == 0)) *
                          0.2)
pilot_df = data[pilot_sample_num, ]
analysis_df = data[-pilot_sample_num, ]
treatment = "trt"
outcome = "out"
```

```
covariates = names(analysis_df)[!names(analysis_df)
                                 %in% c(treatment, outcome)]
new_jointVIP = create_jointVIP(treatment = treatment,
                               outcome = outcome,
                               covariates = covariates,
                               pilot_df = pilot_df,
                               analysis_df = analysis_df)

## at this step typically you may wish to do matching or weighting
## the results after can be stored as a post_data
## the post_data here is not matched or weighted, only for illustrative purposes
post_data <- data.frame(year = rnorm(50, 200, 5),
                        pop = rnorm(50, 1000, 500),
                        gdpPercap = runif(50, 100, 1000),
                        trt = rbinom(50, 1, 0.5),
                        out = rnorm(50, 1, 0.2))
post_dat_jointVIP = create_post_jointVIP(new_jointVIP, post_data)
summary(post_dat_jointVIP)
```

Index