

# Package: jointVIP (via r-universe)

September 2, 2024

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

**Version** 0.1.2.001

**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](https://doi.org/10.1080/00031305.2024.2303419)>. See the package paper by Liao and Pimentel (2023) <[arxiv:2302.10367](https://arxiv.org/abs/2302.10367)> for a beginner friendly user introduction.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**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

**Repository** <https://ldliao.r-universe.dev>

**RemoteUrl** <https://github.com/ldliao/jointvip>

**RemoteRef** HEAD

**RemoteSha** 5eef4b0bd32cff6069f1b03cf69174f0b0d0cb48

## Contents

add_bias_curves . . . . .	2
add_variable_labels . . . . .	3
bootstrap.plot . . . . .	3
brfss . . . . .	5
ceiling_dec . . . . .	5
check_measures . . . . .	6
create_jointVIP . . . . .	6
create_post_jointVIP . . . . .	7
floor_dec . . . . .	8
get_boot_measures . . . . .	9
get_measures . . . . .	9
get_post_measures . . . . .	10
one_hot . . . . .	10
plot_jointVIP . . . . .	11
plot_post_jointVIP . . . . .	12
print_jointVIP . . . . .	13
print_post_jointVIP . . . . .	14
summary_jointVIP . . . . .	16
summary_post_jointVIP . . . . .	17
<b>Index</b>	<b>19</b>

---

add_bias_curves	<i>support function to plot bias curves</i>
-----------------	---

---

## 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](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	<i>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)</i>
----------------	--

---

**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	<i>create jointVIP object</i>
-----------------	-------------------------------

---

**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	<i>Calculate bootstrapped variation additional tool to help calculate the uncertainty of each variable's bias</i>
-------------------	---

---

**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	<i>Prepare data frame to plot standardized omitted variable bias Marginal standardized mean differences and outcome correlation</i>
--------------	---

---

**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	<i>Post-measures data frame to plot post-standardized omitted variable bias</i>
-------------------	---

---

**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	<i>support function for one-hot encoding</i>
---------	--

---

**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	<i>plot the jointVIP object</i>
---------------	---------------------------------

---

## 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	<i>Obtains a print for jointVIP object</i>
----------------	--

---

**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

## \* datasets

- brfss, 5
- add\_bias\_curves, 2
- add\_variable\_labels, 3
- bootstrap.plot, 3
- brfss, 5
- ceiling\_dec, 5
- check\_measures, 6
- create\_jointVIP, 6
- create\_post\_jointVIP, 7
- floor\_dec, 8
- get\_boot\_measures, 9
- get\_measures, 9
- get\_post\_measures, 10
- one\_hot, 10
- plot\_jointVIP, 11
- plot\_post\_jointVIP, 12
- print\_jointVIP, 13
- print\_post\_jointVIP, 14
- summary\_jointVIP, 16
- summary\_post\_jointVIP, 17