

# Package ‘rifreg’

July 23, 2025

**Type** Package

**Title** Estimate Recentered Influence Function Regression

**Version** 1.1.0

**Maintainer** Samuel Meier <samuel.meier+cran@immerda.ch>

**Description** Provides functions to compute recentered influence functions (RIF) of a distributional variable at the mean, quantiles, variance, gini or any custom functional of interest. The package allows to regress the RIF on any number of covariates. Generic print, plot and summary functions are also provided. Reference: Firpo, Sergio, Nicole M. Fortin, and Thomas Lemieux. (2009) <[doi:10.3982/ECTA6822](https://doi.org/10.3982/ECTA6822)>. ``Unconditional Quantile Regressions.".

**License** GPL (>= 3)

**Depends** ggplot2, R (>= 2.10)

**Imports** Formula, Hmisc, methods, parallel, pbapply, sandwich, stats

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** xz

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** David Gallusser [aut],  
Samuel Meier [aut, cre]

**Repository** CRAN

**Date/Publication** 2025-06-21 13:20:02 UTC

## Contents

|                                                |   |
|------------------------------------------------|---|
| check_weights . . . . .                        | 2 |
| compute_generalized_lorenz_ordinates . . . . . | 3 |

|                                              |           |
|----------------------------------------------|-----------|
| compute_gini . . . . .                       | 3         |
| compute_weighted_ecdf . . . . .              | 4         |
| get_rif . . . . .                            | 5         |
| get_rif_gini . . . . .                       | 7         |
| get_rif_interquantile_range . . . . .        | 8         |
| get_rif_interquantile_ratio . . . . .        | 9         |
| get_rif_mean . . . . .                       | 10        |
| get_rif_quantiles . . . . .                  | 11        |
| get_rif_variance . . . . .                   | 12        |
| integrate_generalized_lorenz_curve . . . . . | 12        |
| men8385 . . . . .                            | 13        |
| plot.rifreg . . . . .                        | 14        |
| print.rifreg . . . . .                       | 15        |
| rifreg . . . . .                             | 16        |
| summary.rifreg . . . . .                     | 19        |
| <b>Index</b>                                 | <b>20</b> |

---

|               |                      |
|---------------|----------------------|
| check_weights | <i>Check weights</i> |
|---------------|----------------------|

---

**Description**

Helper function to check a weights vector. Makes sure the weights are positive numeric values (not all zeros) and of the same length as the dependent variable dep\_var. Replaces all NAs with 0 and sets all weights to 1 if weights is set to NULL.

**Usage**

```
check_weights(dep_var, weights)
```

**Arguments**

- dep\_var            dependent variable of distributional function. Can be any discrete or continuous vector of length 1 or more.
- weights           positive numeric vector of length(dep\_var) containing the weights or NULL.

**Value**

positive numeric vector of length(dep\_var) containing the checked weights. If weights = NULL, all weights are set to 1.

**Examples**

```
dep_var <- c(1, 3, 9, 16, 3, 7, 4, 9)
weights <- c(2, 1, 3, 4, 4, 1, 6, 3)
check_weights(dep_var, weights)
```

---

compute\_generalized\_lorenz\_ordinates  
*Generalized Lorenz ordinates*

---

### Description

Compute the generalized Lorenz ordinates of dep\_var (i.e. the share of total income observations up to each value in dep\_var amass scaled by the mean income).

### Usage

```
compute_generalized_lorenz_ordinates(dep_var, weights)
```

### Arguments

|         |                                                                                         |
|---------|-----------------------------------------------------------------------------------------|
| dep_var | dependent variable of a distributional function. Discrete or continuous numeric vector. |
| weights | numeric vector of non-negative observation weights, hence of same length as dep_var.    |

### Value

thes generalized Lorenz ordinates for a vector dep\_var.

### Examples

```
dep_var <- c(1, 3, 9, 16, 3, 7, 4, 9)
weights <- c(2, 1, 3, 4, 4, 1, 6, 3)
generalized_lorenz_ordinates <-
  compute_generalized_lorenz_ordinates(
    dep_var = dep_var,
    weights = weights
  )
```

---

compute\_gini      *Compute Gini coefficient*

---

### Description

Compute a weighted Gini coefficient by integrating the generalized Lorenz curve.

### Usage

```
compute_gini(dep_var, weights)
```

**Arguments**

|         |                                                                                      |
|---------|--------------------------------------------------------------------------------------|
| dep_var | values of a non-negative continuous variable                                         |
| weights | numeric vector of non-negative observation weights, hence of same length as dep_var. |

**Value**

The numeric value indicating the weighted Gini coefficient of the the dependent variable.

**References**

Firpo, Sergio P., Nicole M. Fortin, and Thomas Lemieux. 2018. “Decomposing Wage Distributions Using Recentered Influence Function Regressions.” *Econometrics* 6(2), 28.

**Examples**

```
set.seed(123)
dep_var <- rlnorm(100)
weights <- rep(1, 100)
compute_gini(dep_var, weights)
```

---

|                       |                            |
|-----------------------|----------------------------|
| compute_weighted_ecdf | <i>Weighted ECDF value</i> |
|-----------------------|----------------------------|

---

**Description**

Compute values of the ECDF for a vector dep\_var (i.e. the empirical probability for each observation in dep\_var that a value in dep\_var is smaller or equal).

**Usage**

```
compute_weighted_ecdf(dep_var, weights)
```

**Arguments**

|         |                                                                                         |
|---------|-----------------------------------------------------------------------------------------|
| dep_var | dependent variable of a distributional function. Discrete or continuous numeric vector. |
| weights | numeric vector of non-negative observation weights, hence of same length as dep_var.    |

**Value**

the values of ECDF for a vector dep\_var.

**Examples**

```

dep_var <- c(1, 3, 9, 16, 3, 7, 4, 9)
weights <- c(2, 1, 3, 4, 4, 1, 6, 3)
value_of_ecdf <-
  compute_weighted_ecdf(
    dep_var = dep_var,
    weights = weights
  )

```

get\_rif

*Estimate Recentered Influence Functions***Description**

This function estimates the recentered influence function (RIF) of a chosen distributional statistic (e.g. quantiles, variance or gini).

**Usage**

```

get_rif(
  dep_var,
  weights = NULL,
  statistic,
  probs = NULL,
  custom_rif_function = NULL,
  ...
)

```

**Arguments**

|                     |                                                                                                                                                                                                                                                                    |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dep_var             | dependent variable of distributional function. Discrete or continuous numeric vector.                                                                                                                                                                              |
| weights             | numeric vector of non-negative observation weights, hence of same length as dep_var. The default (NULL) is equivalent to <code>weights = rep(1, length(dep_var))</code> .                                                                                          |
| statistic           | string containing the distributional statistic for which to compute the RIF. Can be one of "mean", "variance", "quantiles", "gini", "interquantile_range", "interquantile_ratio", or "custom". If "custom" is selected a custom_rif_function needs to be provided. |
| probs               | a vector of length 1 or more with quantile positions to calculate the RIF. Each quantile is indicated with value between 0 and 1. Only required if statistic = "quantiles".                                                                                        |
| custom_rif_function | the RIF function to compute the RIF of the custom distributional statistic. Default is NULL. Only needs to be provided if statistic = "custom". Every custom_rif_function needs the parameters dep_var, weights and probs. If they                                 |

are not needed they must be set to NULL in the function definition (e.g. probs = NULL). A custom function must return a data frame containing at least a "rif" and "weights" column. See examples for further details.

... additional parameters passed to the custom\_rif\_function. Apart from dep\_var, weights and probs they must have a different name than the ones in rif. For instance, if you want to pass a parameter statistic to the custom\_rif\_function, name it custom\_statistic.

## Value

a data frame with the RIF value for each observation and in the case of several quantiles a column for each quantile.

## References

Firpo, Sergio P., Nicole M. Fortin, and Thomas Lemieux. 2009. "Unconditional Quantile Regressions." *Econometrica* 77(3): 953–73.

Cowell, Frank A., and Emmanuel Flachaire. 2015. "Statistical Methods for Distributional Analysis." In Anthony B. Atkinson and François Bourguignon (eds.), *Handbook of Income Distribution*. Amsterdam: Elsevier.

## Examples

```
dep_var <- c(1, 3, 9, 16, 3, 7, 4, 9)
probs <- seq(1:9) / 10
weights <- c(2, 1, 3, 4, 4, 1, 6, 3)
rif <- get_rif(
  dep_var = dep_var,
  weights = weights,
  statistic = "quantiles",
  probs = probs
)

# custom function
custom_variance_function <- function(dep_var, weights, probs = NULL) {
  weighted_mean <- weighted.mean(x = dep_var, w = weights)
  rif <- (dep_var - weighted_mean)^2
  rif <- data.frame(rif, weights)
  names(rif) <- c("rif_variance", "weights")
  return(rif)
}

set.seed(123)
dep_var <- rlnorm(100)
weights <- rep(1, 100)

# custom function top 10% percent income share
# (see Essam-Nassah & Lambert, 2012, and Rios-Avila, 2020)
custom_top_income_share_function <- function(dep_var, weights, probs = 0.1) {
  probs <- 1 - probs
  weighted_mean <- weighted.mean(x = dep_var, w = weights)
```

```

weighted_quantile <- Hmisc::wtd.quantile(x = dep_var, weights = weights, probs = probs)
lorenz_ordinate <-
  sum(dep_var[which(dep_var <= weighted_quantile)] *
    weights[which(dep_var <= weighted_quantile)]) / sum(dep_var * weights)
if_lorenz_ordinate <- -(dep_var / weighted_mean) * lorenz_ordinate +
  ifelse(dep_var < weighted_quantile,
    dep_var - (1 - probs) * weighted_quantile,
    probs * weighted_quantile
  ) / weighted_mean
rif_top_income_share <- (1 - lorenz_ordinate) - if_lorenz_ordinate
rif <- data.frame(rif_top_income_share, weights)
names(rif) <- c("rif_top_income_share", "weights")
return(rif_top_income_share)
}

rif_custom <- get_rif(
  dep_var = dep_var,
  weights = weights,
  statistic = "custom",
  custom_rif_function = custom_variance_function
)

```

---

|              |                                         |
|--------------|-----------------------------------------|
| get_rif_gini | <i>Estimate RIF of Gini coefficient</i> |
|--------------|-----------------------------------------|

---

## Description

Compute the recentered influence function (RIF) of a weighted Gini coefficient.

## Usage

```
get_rif_gini(dep_var, weights)
```

## Arguments

|         |                                                                                      |
|---------|--------------------------------------------------------------------------------------|
| dep_var | values of a non-negative continuous dependent variable                               |
| weights | numeric vector of non-negative observation weights, hence of same length as dep_var. |

## Value

A data frame with one column containing the RIF of the Gini coefficient for each observation.

## References

- Cowell, Frank A., and Emmanuel Flachaire. 2007. "Income distribution and inequality measurement: The problem of extreme values." *Journal of Econometrics*, 141(2), 1044-1072.
- Firpo, Sergio P., Nicole M. Fortin, and Thomas Lemieux. 2018. "Decomposing Wage Distributions Using Recentered Influence Function Regressions." *Econometrics* 6(2), 28.
- Monti, Anna Clara. 1991. "The study of the Gini concentration ratio by means of the influence function." *Statistica* 51(4), 561–577.

## Examples

```
set.seed(123)
dep_var <- rlnorm(100)
weights <- rep(1, 100)
rif_gini <- get_rif_gini(dep_var = dep_var, weights = weights)
rif_gini

gini <- compute_gini(dep_var = dep_var, weights = weights)
all.equal(gini, mean(rif_gini$rif_gini))
```

---

```
get_rif_interquantile_range
```

*Estimate RIF of interquantile range*

---

## Description

Compute the recentered influence function (RIF) of a weighted interquantile range.

## Usage

```
get_rif_interquantile_range(dep_var, weights, probs, ...)
```

## Arguments

|         |                                                                                                                                                                                                                                                       |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dep_var | dependent variable of distributional function. Discrete or continuous numeric vector.                                                                                                                                                                 |
| weights | numeric vector of non-negative observation weights, hence of same length as dep_var. The default (NULL) is equivalent to weights = rep(1, length(dep_var)).                                                                                           |
| probs   | a vector of length 2 with probabilities corresponding to the limits of the interquantile range of interest. The interquantile range is defined as difference between the quantile with the larger probability and the one with the lower probability. |
| ...     | further arguments passed on to <a href="#">density</a> .                                                                                                                                                                                              |

**Value**

A data frame with one column containing the RIF of the interquantile range for each observation and one column containing the weights.

**References**

Firpo, Sergio P., Nicole M. Fortin, and Thomas Lemieux. 2018. “Decomposing Wage Distributions Using Recentered Influence Function Regressions.” *Econometrics* 6(2), 28.

**Examples**

```
set.seed(123)
dep_var <- rlnorm(100)
weights <- rep(1, 100)
get_rif_interquantile_range(dep_var, probs = c(0.1, 0.9), weights = weights)
```

---

```
get_rif_interquantile_ratio
```

*Estimate RIF of interquantile ratio*

---

**Description**

Compute the recentered influence function (RIF) of a weighted interquantile ratio.

**Usage**

```
get_rif_interquantile_ratio(dep_var, weights, probs, ...)
```

**Arguments**

|         |                                                                                                                                                                                                                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dep_var | dependent variable of a distributional function. Discrete or continuous numeric vector.                                                                                                                                                                                                                |
| weights | numeric vector of non-negative observation weights, hence of same length as dep_var. The default (NULL) is equivalent to <code>weights = rep(1, length(dep_var))</code> .                                                                                                                              |
| probs   | a vector of length 2 with probabilities corresponding to the quantiles in the ratio's numerator and the denominator. The function defines the interquantile ratio as the ratio between the quantile with the larger probability (numerator) and the quantile with the lower probability (denominator). |
| ...     | further arguments passed on to <a href="#">density</a> .                                                                                                                                                                                                                                               |

**Value**

A data frame with one column containing the RIF of the interquantile ratio for each observation.

## References

Chung, Choe, and Philippe Van Kerm. 2018. "Foreign workers and the wage distribution: What does the influence function reveal?", *Econometrics* 6(3), 41.

## Examples

```
set.seed(123)
dep_var <- rlnorm(100)
weights <- rep(1, 100)
get_rif_interquantile_ratio(dep_var, probs = c(0.1, 0.9), weights = weights)
```

---

|              |                                 |
|--------------|---------------------------------|
| get_rif_mean | <i>Estimate RIF at the Mean</i> |
|--------------|---------------------------------|

---

## Description

Function to estimate the recentered influence function (RIF) at the mean of a weighted distribution of a dependent variable.

## Usage

```
get_rif_mean(dep_var)
```

## Arguments

|         |                                                                                         |
|---------|-----------------------------------------------------------------------------------------|
| dep_var | dependent variable of a distributional function. Discrete or continuous numeric vector. |
|---------|-----------------------------------------------------------------------------------------|

## Value

A data frame with one column of length(dep\_var) containing the RIF at the mean.

## Examples

```
dep_var <- c(1, 3, 9, 16, 3, 7, 4, 9)
get_rif_mean(dep_var)
```

---

|                   |                                  |
|-------------------|----------------------------------|
| get_rif_quantiles | <i>Estimate RIF at Quantiles</i> |
|-------------------|----------------------------------|

---

## Description

Function to estimate the recentered influence function (RIF) at one or several specified quantiles of a weighted distribution of a dependent variable.

## Usage

```
get_rif_quantiles(dep_var, weights, probs, ...)
```

```
get_rif_quantile(dep_var, weights, probs, ...)
```

## Arguments

|         |                                                                                                                                                                           |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dep_var | dependent variable of a distributional function. Discrete or continuous numeric vector.                                                                                   |
| weights | numeric vector of non-negative observation weights, hence of same length as dep_var. The default (NULL) is equivalent to <code>weights = rep(1, length(dep_var))</code> . |
| probs   | the specific quantile at which to estimate the RIF.                                                                                                                       |
| ...     | further arguments passed on to <a href="#">density</a> .                                                                                                                  |

## Value

A data frame with the number of columns equaling the length of vector probs and an additional column containing the weights. Each column contains the RIF values at the quantile's probabilities.

## Functions

- `get_rif_quantile()`: Helper function to estimate the RIF values at a specific quantile.

## Examples

```
dep_var <- c(1, 3, 9, 16, 3, 7, 4, 9)
probs <- seq(1:9) / 10
weights <- c(2, 1, 3, 4, 4, 1, 6, 3)
get_rif_quantiles(dep_var, probs, weights = weights)
```

---

|                  |                                 |
|------------------|---------------------------------|
| get_rif_variance | <i>Estimate RIF of variance</i> |
|------------------|---------------------------------|

---

**Description**

Function to estimate the recentered influence function (RIF) of the variance of a weighted distribution of a dependent variable.

**Usage**

```
get_rif_variance(dep_var, weights)
```

**Arguments**

|         |                                                                                                                                                                           |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dep_var | dependent variable of a distributional function. Discrete or continuous numeric vector.                                                                                   |
| weights | numeric vector of non-negative observation weights, hence of same length as dep_var. The default (NULL) is equivalent to <code>weights = rep(1, length(dep_var))</code> . |

**Value**

A data frame with one column containing the RIF of the variance for each observation and one column containing the weights.

**Examples**

```
dep_var <- c(1, 3, 9, 16, 3, 7, 4, 9)
weights <- c(2, 1, 3, 4, 4, 1, 6, 3)
get_rif_variance(dep_var, weights = weights)
```

---

|                                    |                                           |
|------------------------------------|-------------------------------------------|
| integrate_generalized_lorenz_curve | <i>Integrate generalized Lorenz curve</i> |
|------------------------------------|-------------------------------------------|

---

**Description**

Computes the area under the lorenz curve.

**Usage**

```
integrate_generalized_lorenz_curve(dep_var, weights)
```

**Arguments**

|                      |                                                                                                                                                                                         |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>dep_var</code> | dependent variable of a distributional function. Discrete or continuous numeric vector.                                                                                                 |
| <code>weights</code> | numeric vector of non-negative observation weights, hence of same length as <code>dep_var</code> . The default (NULL) is equivalent to <code>weights = rep(1, length(dep_var))</code> . |

**Value**

the size of the area under the lorenz curve (the integrated lorenz curve).

**Examples**

```
dep_var <- c(1, 3, 9, 16, 3, 7, 4, 9)
weights <- c(2, 1, 3, 4, 4, 1, 6, 3)
integrated_lorenz_curve <-
  integrate_generalized_lorenz_curve(
    dep_var = dep_var,
    weights = weights
  )
```

men8385

*Sample of male wage data from the CPS 1983-1985***Description**

A sample of the the Merged Outgoing Rotation Group of the Current Population Survey of 1983, 1984 and 1985 used by Firpo, Fortin & Lemieux (2009). The data contains a selection of 10 variables and a sample of 26,695 observations of male workers – corresponding to a tenth of the original 266,956 observations. See Lemieux (2006) for details on data selection and recoding.

**Usage**

```
men8385
```

**Format**

A data frame with 26,695 rows and 10 variables.

**wage** Hourly wage in US dollars at constant prices

**union** Union status indicator

**nonwhite** Non-white indicator

**married** Married indicator

**education** Factor variable with 6 education levels: high-school graduates (reference), elementary, high-school dropouts, some college, college graduates, post college graduates

**experience** Factor variable with 9 potential experience levels, each of five years gap, 20 to 24 years as reference level)

**weights** CPS sample weights

**age** Age in years

**education\_in\_years** Education in years

**experience\_in\_years** Experience in years

## Source

Sergio Firpo, Nicole M. Fortin, and Thomas Lemieux, "Unconditional Quantile Regressions", *Econometrica*, Vol. 77, No. 3 (May, 2009), pp. 953-973.

Replication files: <<https://www.econometricsociety.org/publications/econometrica/2009/05/01/unconditional-quantile-regressions>>

Thoms Lemieux, "Increasing Residual Wage Inequality: Composition Effects, Noisy Data, or Rising Demand for Skill?", *American Economic Review*, Vol. 96, No. 3 (June, 2006), pp. 461-498.

---

plot.rifreg

*Plot the coefficients of a rifreg object*

---

## Description

Coefficients are plotted for each quantile and each covariate. Specific covariates can be selected and standard errors displayed if desired.

## Usage

```
## S3 method for class 'rifreg'
plot(
  x,
  varselect = NULL,
  confidence_level = 0.05,
  vcov = sandwich::sandwich,
  ...
)
```

## Arguments

**x** an object of class "rifreg", usually, a result of a call to [rifreg](#) with statistic = "quantiles".

**varselect** vector of length 1 or more containing the names of the covariates to display.

**confidence\_level** numeric value between 0 and 1 (default = 0.95) that defines the confidence interval plotted as a ribbon and defined as  $qnorm(confidence\_level/2) * \text{standard error}$ .

|      |                                                                                                                                                                                                                                                                                                                               |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| vcov | Function to estimate covariance matrix of rifreg coefficients if covariance matrix has not been bootstrapped. Per default, heteroscedasticity-consistent (HC) standard errors are calculated using <a href="#">sandwich</a> . Note: These standard errors do not take the variance introduced by estimating RIF into account. |
| ...  | other parameters to be passed to plotting function. See <a href="#">ggplot</a> for further information.                                                                                                                                                                                                                       |

## Value

a "ggplot" containing the coefficients for each (selected) covariate

## Examples

```
rifreg <- rifreg(
  formula = log(wage) ~ union +
    nonwhite +
    married +
    education +
    experience,
  data = men8385,
  statistic = "quantiles",
  probs = seq(0.1, 0.9, 0.1),
  weights = weights
)

plot(rifreg)

plot(rifreg, varselect = c("age", "unionyes"), confidence_level = 0.1)
```

---

|              |                                        |
|--------------|----------------------------------------|
| print.rifreg | <i>Print method for class "rifreg"</i> |
|--------------|----------------------------------------|

---

## Description

Print method for class "rifreg"

## Usage

```
## S3 method for class 'rifreg'
print(x, ...)
```

## Arguments

|     |                                                                                      |
|-----|--------------------------------------------------------------------------------------|
| x   | an object of class "rifreg", usually, a result of a call to <a href="#">rifreg</a> . |
| ... | other parameters to be passed to printing function.                                  |

**Value**

the function `print.rifreg()` returns the the covariates' coefficients of the RIF regressions derived from the fitted linear model given in object `x`.

**Examples**

```

rifreg <- rifreg(
  formula = log(wage) ~ union +
    nonwhite +
    married +
    education +
    experience,
  data = men8385,
  statistic = "quantiles",
  probs = seq(0.1, 0.9, 0.1),
  weights = weights
)

print(rifreg)

```

---

rifreg

*RIF regression*


---

**Description**

Estimate a recentered influence function (RIF) regression for a distributional statistic of interest.

**Usage**

```

rifreg(
  formula,
  data,
  statistic = "quantiles",
  weights = NULL,
  probs = c(1:9)/10,
  custom_rif_function = NULL,
  na.action = na.omit,
  bootstrap = FALSE,
  bootstrap_iterations = 100,
  cores = 1,
  ...
)

```

**Arguments**

|                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>formula</code>              | an object of class "formula". See <a href="#">lm</a> for further details.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>data</code>                 | a data frame containing the variables in the model.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>statistic</code>            | string containing the distributional statistic for which to compute the RIF. Can be one of "quantiles", "mean", "variance", "gini", "interquantile_range", "interquantile_ratio", or "custom". Default is "quantiles". If "custom" is selected, a <code>custom_rif_function</code> needs to be provided.                                                                                                                                                                                                                                     |
| <code>weights</code>              | numeric vector of non-negative observation weights, hence of same length as <code>dep_var</code> . The default (NULL) is equivalent to <code>weights = rep(1, length(dep_var))</code> .                                                                                                                                                                                                                                                                                                                                                      |
| <code>probs</code>                | a vector of length 1 or more with probabilities of quantiles. Each quantile is indicated with a value between 0 and 1. Default is <code>c(1:9)/10</code> . If <code>statistic = "quantiles"</code> , a single RIF regression for every quantile in <code>probs</code> is estimated. An interquantile ratio (range) is defined by the ratio (difference) between the <code>max(probs)</code> -quantile and the <code>min(probs)</code> -quantile.                                                                                             |
| <code>custom_rif_function</code>  | the RIF function to compute the RIF of the custom distributional statistic. Default is NULL. Only needs to be provided if <code>statistic = "custom"</code> . Every <code>custom_rif_function</code> needs the parameters <code>dep_var</code> , <code>weights</code> and <code>probs</code> . If they are not needed, they must be set to NULL in the function definition (e.g. <code>probs = NULL</code> ). A custom function must return a data frame containing at least a "rif" and "weights" column. See examples for further details. |
| <code>na.action</code>            | generic function that defines how NAs in the data should be handled. Default is <code>na.omit</code> , leading to exclusion of observations that contain one or more missings. See <a href="#">na.action</a> for further details.                                                                                                                                                                                                                                                                                                            |
| <code>bootstrap</code>            | boolean (default = FALSE) indicating if bootstrapped standard errors will be computed                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>bootstrap_iterations</code> | positive integer indicating the number of bootstrap iterations to execute. Only required if <code>bootstrap = TRUE</code> .                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>cores</code>                | positive integer indicating the number of cores to use when computing bootstrapped standard errors. Only required if <code>bootstrap = TRUE</code> .                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>...</code>                  | additional parameters passed to the <code>custom_rif_function</code> . Apart from <code>dep_var</code> , <code>weights</code> and <code>probs</code> they must have a different name than the ones in <code>rifreg</code> . For instance, if you want to pass a parameter <code>statistic</code> to the <code>custom_rif_function</code> , name it <code>custom_statistic</code> .                                                                                                                                                           |

**Value**

`rifreg` returns an object of class "rifreg".

A "rifreg" object is a list containing the following components:

|                        |                                                                                                                                                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>estimates</code> | a matrix of RIF regression coefficients for each covariate and the intercept. In case of several quantiles, coefficient estimates for each quantile are provided. Equivalent to <code>coef()</code> call of an object of class "lm". |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                  |                                                                                                                    |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <code>rif_lm</code>              | one or several objects of class "lm", containing the detailed RIF regression results.                              |
| <code>rif</code>                 | a data frame containing the RIF for each observation.                                                              |
| <code>bootstrap_se</code>        | bootstrapped standard errors for each coefficient. Only provided if <code>bootstrap = TRUE</code> .                |
| <code>bootstrap_vcov</code>      | the bootstrapped variance-covariance matrix for each coefficient. Only provided if <code>bootstrap = TRUE</code> . |
| <code>statistic</code>           | the distributional statistic for which the RIF was computed.                                                       |
| <code>custom_rif_function</code> | The custom RIF function in case it was provided.                                                                   |
| <code>probs</code>               | the probabilities of the quantiles that were computed, in case the distributional statistic requires quantiles.    |

## References

Firpo, Sergio P., Nicole M. Fortin, and Thomas Lemieux. 2009. "Unconditional Quantile Regressions." *Econometrica* 77(3): 953–73.

Cowell, Frank A., and Emmanuel Flachaire. 2015. "Statistical Methods for Distributional Analysis." In Anthony B. Atkinson and François Bourguignon (eds.), *Handbook of Income Distribution*. Amsterdam: Elsevier.

## Examples

```

rifreg <- rifreg(
  formula = log(wage) ~ union +
    nonwhite +
    married +
    education +
    experience,
  data = men8385,
  statistic = "quantiles",
  weights = weights,
  probs = seq(0.1, 0.9, 0.1),
  bootstrap = FALSE
)

# custom function
custom_variance_function <- function(dep_var, weights, probs = NULL) {
  weighted_mean <- weighted.mean(x = dep_var, w = weights)
  rif <- (dep_var - weighted_mean)^2
  rif <- data.frame(rif, weights)
  names(rif) <- c("rif_variance", "weights")
  return(rif)
}

rifreg <- rifreg(
  formula = log(wage) ~ union + nonwhite + married + education + experience,
  data = men8385,

```

```

    statistic = "custom",
    weights = weights,
    probs = NULL,
    custom_rif_function = custom_variance_function,
    bootstrap = FALSE
  )

```

---

|                |                                          |
|----------------|------------------------------------------|
| summary.rifreg | <i>summary method for class "rifreg"</i> |
|----------------|------------------------------------------|

---

## Description

summary method for class "rifreg"

## Usage

```

## S3 method for class 'rifreg'
summary(object, vcov = sandwich::sandwich, ...)

```

## Arguments

|        |                                                                                                                                                                                                                                                                                                                               |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object | an object of class "rifreg", usually, a result of a call to <a href="#">rifreg</a> .                                                                                                                                                                                                                                          |
| vcov   | Function to estimate covariance matrix of rifreg coefficients if covariance matrix has not been bootstrapped. Per default, heteroscedasticity-consistent (HC) standard errors are calculated using <a href="#">sandwich</a> . Note: These standard errors do not take the variance introduced by estimating RIF into account. |
| ...    | other parameters to be passed to summary functions.                                                                                                                                                                                                                                                                           |

## Value

the function `summary.rifreg()` returns a list of summary statistics derived from the rifreg object given in object. For further details see [summary.lm](#).

## Examples

```

rifreg <- rifreg(
  formula = log(wage) ~ union +
    nonwhite +
    married +
    education +
    experience,
  data = men8385,
  statistic = "quantiles",
  probs = seq(0.1, 0.9, 0.1),
  weights = weights
)
summary(rifreg)

```

# Index

## \* datasets

men8385, [13](#)

check\_weights, [2](#)

class, [17](#)

compute\_generalized\_lorenz\_ordinates,  
[3](#)

compute\_gini, [3](#)

compute\_weighted\_ecdf, [4](#)

density, [8](#), [9](#), [11](#)

get\_rif, [5](#)

get\_rif\_gini, [7](#)

get\_rif\_interquantile\_range, [8](#)

get\_rif\_interquantile\_ratio, [9](#)

get\_rif\_mean, [10](#)

get\_rif\_quantile (get\_rif\_quantiles), [11](#)

get\_rif\_quantiles, [11](#)

get\_rif\_variance, [12](#)

ggplot, [15](#)

integrate\_generalized\_lorenz\_curve, [12](#)

lm, [17](#)

men8385, [13](#)

na.action, [17](#)

plot.rifreg, [14](#)

print.rifreg, [15](#)

rifreg, [14](#), [15](#), [16](#), [19](#)

sandwich, [15](#), [19](#)

summary.lm, [19](#)

summary.rifreg, [19](#)