

# Package: SUMnlmr (via r-universe)

September 26, 2024

**Title** Non-Linear Mendelian Randomization On Partially Summarized Data

**Version** 0.0.0.9000

**Description** Runs non-linear MR calculations on partly-summarized data.

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**Config/testthat/edition** 3

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Depends** R (>= 3.1.0)

**Imports** stats, methods, ggplot2, metafor, matrixStats, utils,  
assertthat, dplyr, survival

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

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

**RemoteUrl** <https://github.com/amymariemason/SUMnlmr>

**RemoteRef** HEAD

**RemoteSha** bc2ed34922f1839e3f5d12ccf3ad443d7aeb00fb

## Contents

bmi_acm . . . . .	2
create_ind_data . . . . .	3
create_nlmr_summary . . . . .	4
create_summary_data . . . . .	6
frac_poly_summ_mr . . . . .	7
generated_data . . . . .	9
getGRvalues . . . . .	10
hamardman.prod . . . . .	11
iv_free . . . . .	12

LDL_CAD . . . . .	13
LDL_CAD_covar . . . . .	13
lowfun . . . . .	14
piecewise_summ_figure . . . . .	14
piecewise_summ_mr . . . . .	15
print.frac_poly_mr . . . . .	18
print.summary.frac_poly_mr . . . . .	18
print.summary.piecewise_summ_mr . . . . .	19
repro . . . . .	19
summary.frac_poly_mr . . . . .	20
summary.piecewise_summ_mr . . . . .	20
upfun . . . . .	21
<b>Index</b>	<b>22</b>

---

bmi_acm	<i>Hunt dataset containing the allele score-associations with bmi and all cause mortality, by 100 quantiles of residual BMI. As used in the paper "Body mass index and all cause mortality in HUNT and UK Biobank studies: linear and non-linear mendelian randomisation analyses" <a href="https://doi.org/10.1136/bmj.11042">https://doi.org/10.1136/bmj.11042</a></i>
---------	--

---

## Description

Hunt dataset containing the allele score-associations with bmi and all cause mortality, by 100 quantiles of residual BMI. As used in the paper "Body mass index and all cause mortality in HUNT and UK Biobank studies: linear and non-linear mendelian randomisation analyses" <https://doi.org/10.1136/bmj.11042>

## Usage

bmi\_acm

## Format

A data frame with 100 rows and 5 variables:

**beta\_bmi** linear association between BMI and allele score, adjusted for age, sex and age-squared

**se\_bmi** standard error for the beta\_bmi term from the linear regression

**beta\_acm** Cox proportional hazards regression association between all cause mortality and allele score, adjusted for age and sex

**se\_acm** standard error for the beta\_acm term from the Cox regression

**mean\_bmi** Average BMI in stratum ...

---

create\_ind\_data      *Generation of individual level data*

---

### Description

generates individual level data with a single genetic variant

### Usage

```
create_ind_data(
  N,
  gpar = 0.3,
  par1 = 1,
  par2 = 0,
  beta0 = 0,
  beta1 = 3,
  beta2 = 7,
  confound = 0.8
)
```

### Arguments

N	number of individuals to create
gpar	genetic parameter; used to create g: single genetic snp, from a binomial distribution with n=2 and p = gpar.
par1	power parameter for fractional poly generation. See details
par2	power parameter for fractional poly generation. . See details.
beta0	covariate parameter. See details
beta1	covariate parameter. See details
beta2	covariate parameter. See details
confound	confounding parameter,c. See details.

### Value

data A data-frame containing the values of g, the genetic variate; X, the exposure; and a variety of Y, the outcome values. All outcomes are continuous not binary.

### Note

This function generates a database with genetic relationships suitable for evaluating non-linear MR relationships. A unknown covariate,u, is generated as a N(0,1) variable. Error terms are generated:  $E_x \sim \exp(1)$  and for  $E_y \sim N(0,1)$   $X = 2 + 0.25 * g + u + E_x$  Outcomes are as follows

- Linear:  $Y = b_0 + b_1X + cU + E_y$
- Quadratic  $Y = b_0 + b_1X + b_2X^2 + cU + E_y$

- Squareroot  $Y = b_0 + b_1\sqrt{X} + cU + E_y$
- Log  $Y = b_0 + b_1\log(X) + cU + E_y$
- Threshold  $Y = b_0 + b_1X + cU + E_y$  if  $X > b_2$  and  $Y = b_0 + cU + E_y$  otherwise
- fracpoly  $Y = b_0 + b_1X^{p_1} + b_2X^{p_2} + cU + E_y$  with the usual adaptations for  $p=0$  or  $p_1=p_2$

### Author(s)

Amy Mason

---

create\_nlmr\_summary    *Creation of summarised mendelian randomisation local estimates*

---

### Description

create\_nlmr\_summary takes individual level data and creates summarised dataset, ready to save and share for summarised nlmr

### Usage

```
create_nlmr_summary(
  y,
  x,
  g,
  covar = NULL,
  family = "gaussian",
  controlonly = FALSE,
  q,
  prestrat = 1,
  strata_method = "ranked",
  strata_bound = c(0.2, 0.1, 0.8, 0.9),
  extra_statistics = FALSE,
  report_GR = FALSE,
  report_het = FALSE,
  seed = 1234
)
```

### Arguments

y	vector of outcome values.
x	vector of exposure values.
g	the instrumental variable.
covar	a matrix of covariates.
family	a description of the error distribution and link function to be used in the model. This is a character string naming either the gaussian (i.e. "gaussian" for continuous outcome data) or binomial (i.e. "binomial" for binary outcome data) family function. "Coxph" can be used to fit survival data

- in this case y must be a Surv object.

controlonly	whether to estimate the gx association in all people, or in controls only. This is set to FALSE as default. It has no effect if family is set to "gaussian"
q	the number of quantiles the exposure distribution is to be split into. Within each quantile a causal effect will be fitted, known as a localised average causal effect (LACE). The default is deciles (i.e. 10 quantiles).
prestrat	the proportional size of pre-strata in the doubly-ranked method. If prestrat = 1 (default), then pre-strata will contain the number of individuals equal to the number of strata, and 1 individual from each pre-stratum is selected into each stratum. If prestrat = 10, then pre-strata contain 10 times the number of individuals as the number of strata, and 10 individuals from each pre-stratum are selected into each stratum. Larger pre-strata can improve the differentiation between pre-strata, although if pre-strata are too large such that the instrument values vary strongly within pre-strata, then the benefit of the doubly-ranked method is lost.
strata_method	what method to use for determining strata. By default this is set to "ranked", using Haodong Tian's double ranked version to calculate strata. The alternative is "residual" for determining the strata from the residual of the exposure regressed on the instrument (As in Statley and Burgess paper). The residual method relies on a constant relationship between the instrument and the exposure across the range of the exposure.
strata_bound	controls what range to use for the LACE estimates in graphs display. By default this is set to restricted, taking the 10th and 90th percentile of internal strata and the 20th and 80th for the bottom of the lowest strata and top of the highest strata. It is a vector taking the percentiles for the lowers bounds of the bottom and then other strata and then upper bounds of top and other strata. This only impacts the "max" and "min" values for the summary table This can be overridden in piecewise_summ_mr by using the xbreaks argument to hardset different breakpoints or replacing default with c(0,0,1,1) to return to true max and minimum
extra_statistics	This will add a second output reporting extra statistics for each strata. These include the true max and min of each strata (regardless of strata_bound setting) and the f statistic and p-value for the regressions
report_GR	This will add the Gelman-Rubin statistics for each strata to the output. Note this only works if strata_method="ranked".
report_het	This will add p-values for assessing the heterogeneity of the instrument - exposure relationship. The first column is the p-value of the Cochran Q heterogeneity test (Q); the second column is the p-value from the trend test (trend).
seed	The random seed to use when generating the quantiles (for reproducibility). If set to NA, the random seed will not be set.

## Value

model the model specifications. The first column is the number of quantiles (q); the second column is the position used to relate x to the LACE in each quantiles (xpos); the third column is the type of confidence interval constructed (ci); the fourth column is the number of bootstrap replications performed (nboot).

**Author(s)**

Amy Mason, leaning heavily on work by James Statley and Matt Arnold

---

create\_summary\_data    *generation of summary level data*

---

**Description**

create\_summary\_data generates semi-summarized level data

**Usage**

```
create_summary_data(
  Ytype,
  q = 10,
  keep = FALSE,
  family = "gaussian",
  controlsonly = "TRUE",
  ...
)
```

**Arguments**

Ytype	The relationship between X and Y; can be "linear", "quad", "sqrt", "log", "threshold" or "fracpoly"
q	The number of quantiles.
keep	Whether to retain the individual level data as well as the summary data
family	a description of the error distribution and link function to be used in the model. For piecewise_mr this can be a character string naming either the gaussian (i.e. "gaussian" for continuous data) or binomial (i.e. "binomial" for binary data) family function.
controlsonly	whether to estimate the gx association in all people, or in controls only. This is set to TRUE as default. It has no effect if family is set to "gaussian"
...	parameters passed to mr_create_data for control of X-Y relationship

**Value**

summary A data-frame containing the semi-summarised beta\_X and beta\_Y values, mean of X\_0, mean of X, max of X and min of X for each quantile.

**Author(s)**

Amy Mason

---

frac_poly_summ_mr	<i>Instrumental variable analysis using fractional polynomials based on summary data</i>
-------------------	--

---

## Description

frac\_poly\_summ\_mr performs instrumental variable analysis by fitting fractional polynomial models to localised average causal effects using meta-regression.

Please note that if you provide synthetic data that are too regular (eg the by associations are all zero or the xmean values are exactly 1, 2, 3, ...), the function may error as several fractional polynomials provide the same fit.

## Usage

```
frac_poly_summ_mr(  
  by,  
  bx,  
  byse,  
  bxse,  
  xmean,  
  method = "FE",  
  d = "both",  
  powers = c(0, -2, -1.5, -1, -0.5, 1, 2),  
  pd = 0.05,  
  average.exposure.associations = FALSE,  
  ci = "model_se",  
  nboot = 100,  
  fig = FALSE,  
  family = "binomial",  
  offset = 0,  
  pref_x = "x",  
  pref_y = "y",  
  ref = NA,  
  ci_type = "overall",  
  breaks = NULL,  
  ylim_lower = NA,  
  ylim_upper = NA,  
  xlim_lower = NA,  
  xlim_upper = NA,  
  seed = 335  
)
```

## Arguments

by	vector of gene-outcome associations.
bx	vector of gene-exposure associations.

<code>byse</code>	vector of standard errors of gene-outcome associations.
<code>bxse</code>	vector of standard errors of gene-exposure associations.
<code>xmean</code>	average value of the exposure in each stratum (or whatever summary of the exposure level in the stratum is desired).
<code>method</code>	meta-regression method parsed to the <code>rma</code> package. The default is fixed-effects ('FE').
<code>d</code>	fractional polynomial degree. The default is degree 1. The other options are: 1, 2, or 'both'.
<code>powers</code>	fractional polynomial powers to test.
<code>pd</code>	p-value cut-off for choosing the best-fitting fractional polynomial of degree 2 over the best-fitting fractional polynomial degree 1. This option is only used if <code>d='both'</code> . The default is 0.05.
<code>average.exposure.associations</code>	TRUE means that the <code>bx</code> estimates are averaged across strata, FALSE means that they are not. Default option is FALSE.
<code>ci</code>	the type of 95% confidence interval. There are three options: (i) using the model standard errors ('model_se'), (ii) using bootstrap standard errors ('bootstrap_se'), (iii) using bootstrap percentile confidence intervals ('bootstrap_per'). The default is the model standard errors.
<code>nboot</code>	the number of bootstrap replications (if required). The default is 100 replications.
<code>fig</code>	a logical statement as to whether the user wants the results displayed in a figure. The default is false.
<code>family</code>	a character string named either 'gaussian' (for continuous data) or binomial (for binary data) or cox (for survival data) family function. This only affects the plotting function - whether the y-axis is log-transformed or not, and the graph's default label.
<code>offset</code>	offset on the x-axis (default is zero).
<code>pref_x</code>	the prefix/label for the x-axis. The default is 'x'.
<code>pref_y</code>	the prefix/label for the y-axis. The default is 'y'.
<code>ref</code>	the reference point for the figure. This is the value of the function that represents the expected difference in the outcome compared with this reference value when the exposure is set to different values. If <code>ref = NA</code> (the default option), then it is set to the mean of <code>x</code> .
<code>ci_type</code>	the type of confidence interval to be displayed on the graph. The default is 'overall' where confidence intervals are presented as bands across the range of <code>x</code> . The alternative option is 'quantile' where the confidence intervals are presented as error bars at the mean in each quantile of <code>x</code> .
<code>breaks</code>	breaks on the y-axis of the figure.
<code>ylim_lower</code>	lower limit for the y-axis of the figure.
<code>ylim_upper</code>	upper limit for the y-axis of the figure.
<code>xlim_lower</code>	lower limit for the x-axis of the figure.
<code>xlim_upper</code>	upper limit for the x-axis of the figure.
<code>seed</code>	The random seed to use when generating the bootstrap samples (for reproducibility). If set to NA, the random seed will not be set.



**Value**

model the model specifications. The first column is the number of quantiles (q); the second column is the position used to relate x to the LACE in each quantiles (xpos); the third column is the type of confidence interval constructed (ci); the fourth column is the number of bootstrap replications performed (nboot).

powers the powers of the chosen polynomial.

coefficients the regression estimates. The first column is the regression coefficients (beta); the second column is the standard errors of regression coefficients (se); the third column is the lower confidence interval (lci); the fourth column is the upper confidence interval (uci); the fifth column is the p-value (pval).

lace the localised average causal effect estimate in each quantile. The first column is the regression coefficients (beta); the second column is the standard errors of regression coefficients (se); the third column is the lower confidence interval (lci); the fourth column is the upper confidence interval (uci); the fifth column is the p-value (pval).

xcoef the association between the instrument and the exposure in each quantile. The first column is the regression coefficients (beta); the second column is the standard errors of regression coefficients (se).

p\_tests the p-value of the non-linearity tests. The first column is the p-value of the test between the fractional polynomial degrees (fp\_d1\_d2); the second column is the p-value from the fractional polynomial non-linearity test (fp); the third column is the p-value from the quadratic test (quad); the fourth column is the p-value from the Cochran Q test (Q). The first column is the p-value of the Cochran Q heterogeneity test (Q); the second column is the p-value from the trend test (trend).

figure ggplot command to produce a figure.

**Author(s)**

Stephen Burgess [sb452@medschl.cam.ac.uk](mailto:sb452@medschl.cam.ac.uk), leaning heavily on James R Staley [js16174@bristol.ac.uk](mailto:js16174@bristol.ac.uk)

---

generated\_data

*Artificial genetic & phenotype data, for purposes of package tests*

---

**Description**

Artificial genetic & phenotype data, for purposes of package tests

**Usage**

generated\_data

**Format**

A data frame with 10000 rows and 1 variables:

**g** gene count of 0, 1 or 2. Distributed binomial with n=2 and p =0.3

**u** "unmeasured" confounder N(0,1)

**errorX** Error term for X,  $\sim \text{exp}(1)$

**errorY** Error term for Y,  $\sim N(0,1)$

**X** Exposure.  $X = 2 + 0.25 * g + u + \text{errorX}$

**linear.Y** Linear outcome.  $Y = X + 0.8u + \text{errorY}$

**quadratic.Y** Quadratic outcome.  $Y = 2X^2 X + 0.8u + \text{error Y}$

**sqrt.Y** Square root outcome.  $Y = \sqrt{X} + 0.8 * u + \text{errorY}$

**log.Y** Log outcome.  $Y = \log(X) + 0.8 * U + \text{errorY}$

**threshold.Y**  $X + 0.8 * U + \text{errorY}$  if  $X > 2$  and  $0.8U + \text{errorY}$  otherwise

**fracpoly.Y** Fractional polynomial  $Y = X + 2/X + 0.8U + \text{errorY}$

---

getGRvalues

*Gelman–Rubin (GR) Statistics*

---

## Description

getGRvalues calculates the the Gelman–Rubin (GR) statistics for each strata (i.e. LACE-strata) as described in Haodong’s paper on doubly-ranked methods in Text S1

## Usage

```
getGRvalues(X = X, Zstratum = Zstratum, NC = 2, t1 = 1.02)
```

## Arguments

X	vector of coarsened exposures
Zstratum	vector of pre-strata index column
NC	number of chains - a lower NC is encouraged
t1	threshold level, should be >1

## Value

max\_GR is the maximum G-R statistic accross all strata

stafied\_strata is a list of the strata where the GR-statistic meets the threshold limit

GR\_values is a table of the a and b values for each strata

## Author(s)

Haodong Tian for underlying code, Amy Mason for conversion to R function [haodong.tian@mrc-bsu.cam.ac.uk](mailto:haodong.tian@mrc-bsu.cam.ac.uk)

**Examples**

```

#Toy data example-----
N<-10000 #N is the total sample size
Z<-rnorm( N,0,0.5 ) # instrument
U1<-rnorm(N,0,1)
U2<-rnorm(N,0,1) #U2 is confounder
h_X<-function(Z){ 0.5*Z }
X<-h_X(Z)+U2+U1
fitX<-lm( X~Z )
RR<-as.numeric(summary(fitX)$r.squared[1])
No<-1000 #No is the number of pre-strata (i.e. IV-strata)
X_<-X # true exact X
X<-round(X) #coarsened by round
dat<-cbind( Z, X, X_ ) # X: coarsened exposure; X_: exact exposure
dat_order<-dat[ order(dat[,1] ), ] #ordered by Z
dat_order<-cbind( dat_order , sort(rep( 1:No, N/No )))
dat_order<-as.data.frame( dat_order )
names(dat_order)<-c( 'Z_order', 'X', 'X_', 'ZStratum')
getGRvalues( X=dat_order$X, Zstratum = dat_order$ZStratum, NC=2, t1 = 1.02 )
#-----

```

---

hamardman.prod

*Hamardman product*


---

**Description**

hamardman.prod computes the Hamardman product of a vector of regression coefficients and a matrix of covariates.

**Usage**

```
hamardman.prod(coef, covar)
```

**Arguments**

coef            vector of regression coefficients.  
covar            a matrix of covariates

**Author(s)**

James R Staley [js16174@bristol.ac.uk](mailto:js16174@bristol.ac.uk)

---

iv_free	<i>IV-free exposure</i>
---------	-------------------------

---

### Description

iv\_free computes the IV-free exposure.

### Usage

```
iv_free(y, x, g, covar = NULL, q = 10, family = "gaussian", controlsonly = T)
```

### Arguments

y	vector of outcome values.
x	vector of exposure values.
g	the instrumental variable.
covar	a matrix of covariates.
q	the number of quantiles the exposure distribution is to be split into. The default is deciles (i.e. 10 quantiles).
family	a description of the error distribution and link function to be used in the model (either "gaussian" or "binomial" can be specified).
controlsonly	whether to estimate the gx association in all people, or in controls only. This is set to TRUE by default, but has no effect if family is gaussian.

### Value

xcoef	the association between the exposure and the instrument
x0	the IV-free exposure.
x0q	the quantiles of x0.

### Author(s)

Amy Mason [am2609@medschl.cam.ac.uk](mailto:am2609@medschl.cam.ac.uk) based on similar function in nlme by James R Staley

---

LDL_CAD	<i>UK Biobank dataset containing the allele score-associations with ldl-cholesterol and CAD diagnosis, by 10 quantiles of residual ldl.</i>
---------	---

---

**Description**

UK Biobank dataset containing the allele score-associations with ldl-cholesterol and CAD diagnosis, by 10 quantiles of residual ldl.

**Usage**

LDL\_CAD

**Format**

A data frame with 10 rows and 8 variables:

**bx** linear association between ldl-cholesterol and allele score, unadjusted

**by** logistic regression coefficient between CAD diagnosis and allele score, unadjusted

**bxse** standard error for the bx term from the linear regression

**byse** standard error for the by term from the logistic regression

**xmean** Average ldl in stratum

**xmin** Minimum ldl in stratum

**xmax** Maximum ldl in stratum ...

---

LDL_CAD_covar	<i>UK Biobank dataset containing the allele score-associations with ldl-cholesterol and CAD diagnosis adjusted for age, sex and first 10 principle components, by 10 quantiles of residual ldl.</i>
---------------	---

---

**Description**

UK Biobank dataset containing the allele score-associations with ldl-cholesterol and CAD diagnosis adjusted for age, sex and first 10 principle components, by 10 quantiles of residual ldl.

**Usage**

LDL\_CAD\_covar

**Format**

A data frame with 10 rows and 8 variables:

**bx** linear association between ldl-cholesterol and allele score, adjusted

**by** logistic regression coefficient between CAD diagnosis and allele score, adjusted

**bxse** standard error for the bx term from the linear regression

**byse** standard error for the by term from the logistic regression

**xmean** Average ldl in stratum

**xmin** Minimum ldl in stratum

**xmax** Maximum ldl in stratum ...

---

lowfun	<i>Lower Coefficient value</i>
--------	--------------------------------

---

**Description**

function to obtain the lower coefficient values (i.e. b in Text S1) for each pre-strata

**Usage**

```
lowfun(vec)
```

**Arguments**

vec	input vector
-----	--------------

---

piecewise_summ_figure	<i>Piecewise linear figure</i>
-----------------------	--------------------------------

---

**Description**

piecewise\_figure plots the piecewise linear function.

**Usage**

```
piecewise_summ_figure(  
  xcoef,  
  coef,  
  xmean,  
  lci,  
  uci,  
  xbreaks,  
  family = "gaussian",  
  ref = mean(xmean),
```

```

    pref_x = "x",
    pref_x_ref = "x",
    pref_y = "y",
    breaks = NULL,
    ci_fig = "point"
)

```

### Arguments

xcoef	the association between the exposure and the instrument.
coef	the coefficients of the localized causal effects.
xmean	mean of each x stratum (for plotting point estimates)
lci	upper confidence interval range for each causal effects
uci	upper confidence interval range for each causal effects.
xbreaks	breakpoints in x (for plotting line estimates)
family	a description of the error distribution and link function to be used in the model. This is a character string naming either the gaussian (i.e. "gaussian" for continuous data) or binomial (i.e. "binomial" for binary data) family function.
ref	the reference point for the figure. This is the value of the that represents the expected difference in the outcome compared with this reference value when the exposure is set to different values. The default is the mean of x.
pref_x	the prefix/label for the x-axis. The default is "x".
pref_x_ref	the prefix for the reference value displayed on the y-axis. The default is "x".
pref_y	the prefix/label for the y-axis. The default is "y".
breaks	breaks on the y-axis of the figure.
ci_fig	point confidence intervals, or as ribbon ("point" or "ribbon")

### Value

the plot of the piecewise linear function.

### Author(s)

Amy Mason [am2609@medschl.cam.ac.uk](mailto:am2609@medschl.cam.ac.uk), leaning on work by James Statley and Matt Arnold

---

piecewise\_summ\_mr      *Instrumental variable analysis using piecewise linear method based on summary data*

---

### Description

piecewise\_summ\_mr performs instrumental variable analysis by fitting piecewise linear functions to localised average causal effects

**Usage**

```

piecewise_summ_mr(
  by,
  bx,
  byse,
  bxse,
  xmean,
  xmin,
  xmax,
  xbreaks = NULL,
  family = "gaussian",
  average.exposure.associations = FALSE,
  ci = "model_se",
  nboot = 1000,
  fig = T,
  ref = mean(xmean),
  pref_x = "x",
  pref_x_ref = "x",
  pref_y = "y",
  breaks = NULL,
  ci_fig = "point",
  seed = 875
)

```

**Arguments**

by	vector of gene-outcome associations.
bx	vector of gene-exposure associations.
byse	vector of standard errors of gene-outcome associations.
bxse	vector of standard errors of gene-exposure associations.
xmean	average value of the original exposure in each iv-free strata (or whatever summary of the exposure level in the stratum is desired).
xmin	min value of the original exposure in each stratum (see note)
xmax	max value of the original exposure in each stratum (see note)
xbreaks	break points for the stratum x values (see note)
family	a character string named either 'gaussian' (for continuous data) or binomial (for binary data) or cox (for survival data) family function. This only affects the plotting function - whether the y-axis is log-transformed or not, and the graph's default label. This should match whichever option was used in creating the summary data.
average.exposure.associations	TRUE means that the bx estimates are averaged across strata, FALSE means that they are not. Default option is FALSE.
ci	the type of 95% confidence interval. There are four options: (i) using the model standard errors ('model_se'), (ii) using bootstrap standard errors ('bootstrap_se'), (iii) using bootstrap percentile confidence intervals ('bootstrap_per') The default is the model standard errors.



nboot	the number of bootstrap replications (if required). The default is 1000 replications.
fig	a logical statement as to whether the user wants the results displayed in a figure. The default is false.
ref	the reference point for the figure. This is the value of the function that represents the expected difference in the outcome compared with this reference value when the exposure is set to different values. The default is the mean of x.
pref_x	the prefix/label for the x-axis. The default is "x".
pref_x_ref	the prefix for the reference value displayed on the y-axis. The default is "x".
pref_y	the prefix/label for the y-axis. The default is "y".
breaks	breaks on the y-axis of the figure.
ci_fig	setting confidence interval type. "point" places error bars at the mean of each stratum; "ribbon" draws upper and lower piecewise lines.
seed	The random seed to use when generating the bootstrap samples (for reproducibility). If set to NA, the random seed will not be set.

### Value

model the model specifications. The first column is the number of quantiles (q); the second column is the position used to relate x to the LACE in each quantiles (xpos); the third column is the type of confidence interval constructed (ci); the fourth column is the number of bootstrap replications performed (nboot).

powers the powers of the chosen polynomial.

coefficients the regression estimates. The first column is the regression coefficients (beta); the second column is the standard errors of regression coefficients (se); the third column is the lower confidence interval (lci); the fourth column is the upper confidence interval (uci); the fifth column is the p-value (pval).

### Note

The non-linearity tests uses 'method="DL"' to calculate the p-value for the heterogeneity trend. The fractional polynomial equivalent function allows you to set the method, meaning you may get different results.

There is no option for covariates; they would need to be applied at an earlier stage in the individual data, using the mr\_summarise function.

The min and max of x stratum values are used to choose the appropriate range for fitting of each causal estimate. In the code for summarising data, this is set at the 10% point, and the 90% of each stratum; 20% and 80% in the external ends of the end strata. The first lower value and all upper value are used to set the break points for the estimates in the graph. Alternatively you can hardset this using xbreaks.

### Author(s)

Amy Mason, leaning heavily on work by James Statley and Matt Arnold

---

`print.frac_poly_mr`      *Create plot of Fractional Polynomial Fit*

---

**Description**

summary method for class 'frac\_poly\_mr'.

**Usage**

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

**Arguments**

`x`                    an object of class 'frac\_poly\_mr'.  
`...`                additional arguments affecting the summary produced.

**Author(s)**

James R Staley [js16174@bristol.ac.uk](mailto:js16174@bristol.ac.uk)

---

`print.summary.frac_poly_mr`  
*Print Summary Fractional Polynomial Fits*

---

**Description**

`print.summary` method for class 'frac\_poly\_mr'.

**Usage**

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

**Arguments**

`x`                    an object of class 'frac\_poly\_mr'.  
`...`                additional arguments affecting the summary produced.

**Author(s)**

James R Staley [js16174@bristol.ac.uk](mailto:js16174@bristol.ac.uk)

---

```
print.summary.piecewise_summ_mr
```

*Print summary of piecewise linear fits*

---

### Description

print summary method for class "piecewise\_mr".

### Usage

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

### Arguments

x                    an object of class "piecewise\_mr".  
...                  Arguments to be passed to or from other methods,

### Author(s)

Amy Mason [am2609@medschl.cam.ac.uk](mailto:am2609@medschl.cam.ac.uk)

---

```
reproW                    Repeat rows
```

---

### Description

This function creates a matrix of a repeated vector where each row is the same.

### Usage

```
reproW(x, n)
```

### Arguments

x                    vector to be repeated  
n                    number of repeats

### Author(s)

James R Staley [js16174@bristol.ac.uk](mailto:js16174@bristol.ac.uk)

---

summary.frac\_poly\_mr    *Summarizing Fractional Polynomial Fits*

---

**Description**

summary method for class 'frac\_poly\_mr'.

**Usage**

```
## S3 method for class 'frac_poly_mr'  
summary(object, ...)
```

**Arguments**

object            an object of class 'frac\_poly\_mr'.  
...                additional arguments affecting the summary produced.

**Author(s)**

James R Staley [js16174@bristol.ac.uk](mailto:js16174@bristol.ac.uk)

---

summary.piecewise\_summ\_mr  
                          *Summary of piecewise linear fits*

---

**Description**

summary method for class "piecewise\_summ\_mr".

**Usage**

```
## S3 method for class 'piecewise_summ_mr'  
summary(object, ...)
```

**Arguments**

object            an object of class "piecewise\_summ\_mr".  
...                Arguments to be passed to or from other methods,

**Author(s)**

Amy Mason [am2609@medschl.cam.ac.uk](mailto:am2609@medschl.cam.ac.uk)

---

upfun	<i>Upper Coefficient value</i>
-------	--------------------------------

---

**Description**

function to obtain the upper coefficient values (i.e. a in Text S1) for each pre-strata

**Usage**

upfun(vec)

**Arguments**

vec            input vector

# Index

## \* datasets

- bmi\_acm, [2](#)
- generated\_data, [9](#)
- LDL\_CAD, [13](#)
- LDL\_CAD\_covar, [13](#)

bmi\_acm, [2](#)

create\_ind\_data, [3](#)  
create\_nlmr\_summary, [4](#)  
create\_summary\_data, [6](#)

frac\_poly\_summ\_mr, [7](#)

generated\_data, [9](#)  
getGRvalues, [10](#)

hamardman.prod, [11](#)

iv\_free, [12](#)

LDL\_CAD, [13](#)  
LDL\_CAD\_covar, [13](#)  
lowfun, [14](#)

piecewise\_summ\_figure, [14](#)  
piecewise\_summ\_mr, [15](#)  
print.frac\_poly\_mr, [18](#)  
print.summary.frac\_poly\_mr, [18](#)  
print.summary.piecewise\_summ\_mr, [19](#)

repro, [19](#)

summary.frac\_poly\_mr, [20](#)  
summary.piecewise\_summ\_mr, [20](#)

upfun, [21](#)