Package 'EPLSIM'

May 5, 2025

Type Package

Title Partial Linear Single Index Models for Environmental Mixture Analysis

Version 0.1.1

Date 2025-05-04

Maintainer Yuyan Wang <yuyan.wang@nyumc.org>

Description Collection of ancillary functions and utilities for Partial Linear Single Index Models for Environmental mixture analyses, which currently provides functions for scalar outcomes. The outputs of these functions include the single index function, single index coefficients, partial linear coefficients, mixture overall effect, exposure main and interaction effects, and differences of quartile effects. In the future, we will add functions for binary, ordinal, Poisson, survival, and longitudinal outcomes, as well as models for time-dependent exposures. See Wang et al (2020) <doi:10.1186/s12940-020-00644-4> for an overview.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 2.10)

Imports splines, ggplot2, MASS, ciTools

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

URL https://github.com/YuyanWangSixTwo/EPLSIM

BugReports https://github.com/YuyanWangSixTwo/EPLSIM/issues

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Yuyan Wang [aut, cre] (ORCID: <https://orcid.org/0000-0003-3747-0762>),
Mengling Liu [aut, ctb],
Myeonggyun Lee [ctb]

Repository CRAN

Date/Publication 2025-05-05 14:40:02 UTC

Contents

`confounder.trans` *Transformation for confounder vector Z*

Description

Transformation for confounder vector Z

Usage

```
confounder.trans(Z_continuous, Z_discrete, data)
```

Arguments

`Z_continuous`	A character name vector for continuous confounders
`Z_discrete`	A character name vector for discrete confounders
`data`	Orginial data set

Value

Transformed confounder vector and data set ready for further analysis.

Author(s)

Yuyan Wang

Examples

```
# example to normalize the continuous confounders and
# make dummy variables for categorical confoduners
dat.cov <- data.frame(
  age = c(1.5, 2.3, 3.1, 4.8, 5.2),
  sex = c(1, 2, 1, 2, 2),
  race = c(1, 2, 3, 4, 5)
)
```

```
# specify the confounder vector
Z.name <- c("age", "sex", "race")

# set levels and make the reference level first for categorical confounders
dat.cov$sex <- factor(dat.cov$sex, 1:2, c('Male', 'Female'))
dat.cov$race <- factor(dat.cov$race,1:5,c("NH-White", "NH-Black",
                                          "MexicanAmerican", "OtherRace", "Hispanic"))

# transform the confounder vector and check
cov_m <- confounder.trans(Z_continuous = c("age"), Z_discrete = c("sex", "race"), data = dat.cov)
Z.name <- cov_m$New.Name
dat.cov <- cov_m$Updated.data
print(Z.name)
```

e.interaction.plot *plot interaction effect of two exposures*

Description

plot interaction effect of two exposures

Usage

```
e.interaction.plot(fit, data, exp_1, exp_2)
```

Arguments

fit	Fitted model from function 'plsi.lr.v1'
data	Original data set
exp_1	exposure name hoping to be checked
exp_2	exposure name hoping to be checked

Value

plot of interaction effect of two exposures with others at average level

Author(s)

Yuyan Wang

Examples

```
# example to plot interaction effect of two exposures
data(nhanes.new)
dat <- nhanes.new

# specify variable names and parameters
```

```
Y.name <- "log.triglyceride"
X.name <- c("X1_trans.b.carotene", "X2_retinol", "X3_g.tocopherol", "X4_a.tocopherol",
            "X5_PCB99", "X6_PCB156", "X7_PCB206",
            "X8_3.3.4.4.5.pncb", "X9_1.2.3.4.7.8.hxcdf", "X10_2.3.4.6.7.8.hxcdf")
Z.name <- c("AGE.c", "SEX.Female", "RACE.NH.Black",
           "RACE.MexicanAmerican", "RACE.OtherRace", "RACE.Hispanic" )
spline.num = 5
spline.degree = 3
initial.random.num = 1

# run PLSI linear regression
set.seed(2023)
model_1 <- plsi.lr.v1(data = dat, Y.name = Y.name, X.name = X.name, Z.name = Z.name,
                      spline.num, spline.degree, initial.random.num)

# plot two exposures' interaction effect
e.interaction.plot(model_1, dat, "X4_a.tocopherol", "X3_g.tocopherol")
e.interaction.plot(model_1, dat, "X4_a.tocopherol", "X10_2.3.4.6.7.8.hxcdf")

# exchange exposures' names
e.interaction.plot(model_1, dat, "X8_3.3.4.4.5.pncb", "X6_PCB156")
e.interaction.plot(model_1, dat, "X6_PCB156", "X8_3.3.4.4.5.pncb")
```

e.main.plot *plot single exposure's main effect*

Description

plot single exposure's main effect

Usage

```
e.main.plot(fit, data, exp_name)
```

Arguments

`fit`	Fitted model from function 'plsi.lr.v1'
`data`	Original data set
`exp_name`	exposure name hoping to be plotted

Value

plot of exposure's main effect with other exposures at average level 0

Author(s)

Yuyan Wang

Examples

```
# example to plot some exposure's main effect
data(nhanes.new)
dat <- nhanes.new

# specify variable names and parameters
Y.name <- ”log.triglyceride”
X.name <- c(”X1_trans.b.carotene”, ”X2_retinol”, ”X3_g.tocopherol”, ”X4_a.tocopherol”,
            ”X5_PCB99”, ”X6_PCB156”, ”X7_PCB206”,
            ”X8_3.3.4.4.5.pncb”, ”X9_1.2.3.4.7.8.hxcdf”, ”X10_2.3.4.6.7.8.hxcdf”)
Z.name <- c(”AGE.c”, ”SEX.Female”, ”RACE.NH.Black”,
           ”RACE.MexicanAmerican”, ”RACE.OtherRace”, ”RACE.Hispanic” )
spline.num = 5
spline.degree = 3
initial.random.num = 1

# run PLSI linear regression
set.seed(2023)
model_1 <- plsi.lr.v1(data = dat, Y.name = Y.name, X.name = X.name, Z.name = Z.name,
                      spline.num, spline.degree, initial.random.num)

# plot some exposure's main effect
e.main.plot(model_1, dat, exp_name = c(”X4_a.tocopherol”))
e.main.plot(model_1, dat, exp_name = c(”X5_PCB99”))
e.main.plot(model_1, dat, exp_name = c(”X10_2.3.4.6.7.8.hxcdf”))
```

interquartile.quartile.plot

plot interquartile effect of specific exposure based on quartile of other exposures

Description

plot interquartile effect of specific exposure based on quartile of other exposures

Usage

```
interquartile.quartile.plot(fit, data)
```

Arguments

fit	Fitted model from function ’plsi.lr.v1’
data	Original data set

Value

plot of main interquartile effect of exposure based on quartile of other exposures

Author(s)

Yuyan Wang

Examples

```
# example to interquartile effect based on quartile of other exposures
data(nhanes.new)
dat <- nhanes.new

# specify variable names and parameters
Y.name <- "log.triglyceride"
X.name <- c("X1_trans.b.carotene", "X2_retinol", "X3_g.tocopherol", "X4_a.tocopherol",
            "X5_PCB99", "X6_PCB156", "X7_PCB206",
            "X8_3.3.4.4.5.pncb", "X9_1.2.3.4.7.8.hxcdf", "X10_2.3.4.6.7.8.hxcdf")
Z.name <- c("AGE.c", "SEX.Female", "RACE.NH.Black",
           "RACE.MexicanAmerican", "RACE.OtherRace", "RACE.Hispanic" )
spline.num = 5
spline.degree = 3
initial.random.num = 1

# run PLSI linear regression
set.seed(2023)
model_1 <- plsi.lr.v1(data = dat, Y.name = Y.name, X.name = X.name, Z.name = Z.name,
                      spline.num, spline.degree, initial.random.num)

# plot interquartile quartile
interquartile.quartile.plot(model_1, dat)
```

`mixture.overall.plot` *plot mixture's overall effect based on quantile of exposures*

Description

plot mixture's overall effect based on quantile of exposures

Usage

```
mixture.overall.plot(fit, data)
```

Arguments

fit	Fitted model from function 'plsi.lr.v1'
data	Original data set

Value

plot of predicted outcomes based on quantile of exposures

Author(s)

Yuyan Wang

Examples

```
# example to plot mixture's overall effect
data(nhanes.new)
dat <- nhanes.new

# specify variable names and parameters
Y.name <- "log.triglyceride"
X.name <- c("X1_trans.b.carotene", "X2_retinol", "X3_g.tocopherol", "X4_a.tocopherol",
            "X5_PCB99", "X6_PCB156", "X7_PCB206",
            "X8_3.3.4.4.5.pncb", "X9_1.2.3.4.7.8.hxcdf", "X10_2.3.4.6.7.8.hxcdf")
Z.name <- c("AGE.c", "SEX.Female", "RACE.NH.Black",
           "RACE.MexicanAmerican", "RACE.OtherRace", "RACE.Hispanic" )
spline.num = 5
spline.degree = 3
initial.random.num = 1

# run PLSI linear regression
set.seed(2023)
model_1 <- plsi.lr.v1(data = dat, Y.name = Y.name, X.name = X.name, Z.name = Z.name,
                      spline.num, spline.degree, initial.random.num)

# plot mixture overall effect
mixture.overall.plot(model_1, dat)
```

---

nhanes *This is data from NHANES 2003–2004*

Description

A data set containing outcome triglyceride, ten exposures, and three confounders.

Usage

```
nhanes
```

Format

An object of class `data.frame` with 800 rows and 14 columns.

Details

triglyceride outcome triglyceride level, unite mg/dl

a1.trans.b.carotene exposure: trans-b-carotene (ug/dL)

a5.Retinol exposure: retinol (ug/dL)

a6.g.tocopherol exposure: g-tocopherol (ug/dL)

a7.a.Tocopherol exposure: a-tocopherol (ug/dL)

a10.PCB99 exposure: Polychlorinated Biphenyl (PCB) 99 Lipid Adj (ng/g)

a13.PCB156 exposure: Polychlorinated Biphenyl (PCB) 156 Lipid Adj (ng/g)

a19.PCB206 exposure: Polychlorinated Biphenyl (PCB) 206 Lipid Adj (ng/g)

a20.3.3.4.4.5.pncb exposure: 3,3,4,4,5-Pentachlorobiphenyl (pncb) Lipid Adj (pg/g)

a21.1.2.3.4.7.8.hxcdf exposure: 1,2,3,4,7,8-hxcdf Lipid Adj (pg/g)

a22.2.3.4.6.7.8.hxcdf exposure: 2,3,4,6,7,8-hxcdf Lipid Adj (pg/g)

age subject age at measurement

sex subject sex

race subject race

Author(s)

Yuyan Wang <yuyan.wang@nyumc.org>

---

nhanes.new *This is updated data from original data based on NHANES 2003–2004 survey*

Description

A data set containing outcome triglyceride, re-named ten exposures, and transformed confounders.

Usage

```
nhanes.new
```

Format

An object of class `data.frame` with 789 rows and 17 columns.

Details

triglyceride outcome triglyceride level, unite mg/dl

X1_trans.b.carotene renamed exposure: trans-b-carotene (ug/dL)

X2_retinol renamed exposure: retinol (ug/dL)

X3_g.tocopherol renamed exposure: g-tocopherol (ug/dL)

X4_a.tocopherol renamed exposure: a-tocopherol (ug/dL)

X5_PCB99 renamed exposure: Polychlorinated Biphenyl (PCB) 99 Lipid Adj (ng/g)

X6_PCB156 renamed exposure: Polychlorinated Biphenyl (PCB) 156 Lipid Adj (ng/g)

X7_PCB206 renamed exposure: Polychlorinated Biphenyl (PCB) 206 Lipid Adj (ng/g)

X8_3.3.4.4.5.pncb renamed exposure: 3,3,4,4,5-Pentachlorobiphenyl (pncb) Lipid Adj (pg/g)

X9_1.2.3.4.7.8.hxcdf renamed exposure: 1,2,3,4,7,8-hxcdf Lipid Adj (pg/g)

X10_2.3.4.6.7.8.hxcdf renamed exposure: 2,3,4,6,7,8-hxcdf Lipid Adj (pg/g)

AGE.c rescaled continuous confounder: subject age at measurement

SEX.Female categorical confounder dummy variable: subject sex as Female

RACE.NH.Black categorical dummy variable: subject race as Non-Hispanic Black

RACE.MexicanAmerican categorical dummy variable: subject race as Mexican American

RACE.OtherRace categorical dummy variable: subject race as Other Races

RACE.Hispanic categorical dummy variable: subject race as Hispanic

Author(s)

Yuyan Wang <yuyan.wang@nyumc.org>

---

plsi.lr.v1 *Partial linear single index linear regression for scalar outcome*

Description

Partial linear single index linear regression for scalar outcome

Usage

```
plsi.lr.v1(
  data,
  Y.name,
  X.name,
  Z.name,
  spline.num,
  spline.degree,
  initial.random.num
)
```

Arguments

`data`	A data set including all needed variables
`Y.name`	Variable name for scalar outcome
`X.name`	Variable name vector for exposures
`Z.name`	Variable name vector for confounders
`spline.num`	A number representing the degree of freedom of B-spline basis for link function
`spline.degree`	A number representing the degree of the piece-wise polynomial of B-spline basis for link function
`initial.random.num`	A number representing the number of random initials used in the function

Value

A list of model estimation and prediction results

Author(s)

Yuyan Wang

Examples

```
# example to run the function
data(nhanes.new)
dat <- nhanes.new

# specify variable names
Y.name <- ”log.triglyceride”
X.name <- c(”X1_trans.b.carotene”, ”X2_retinol”, ”X3_g.tocopherol”, ”X4_a.tocopherol”,
            ”X5_PCB99”, ”X6_PCB156”, ”X7_PCB206”,
            ”X8_3.3.4.4.5.pncb”, ”X9_1.2.3.4.7.8.hxcdf”, ”X10_2.3.4.6.7.8.hxcdf”)
Z.name <- c(”AGE.c”, ”SEX.Female”, ”RACE.NH.Black”,
           ”RACE.MexicanAmerican”, ”RACE.OtherRace”, ”RACE.Hispanic” )

# specify spline degree of freedom
spline.num = 5
# specify spline degree
spline.degree = 3
# specify number of random initials for estimation
initial.random.num = 1

# run the model
set.seed(2023)
model_1 <- plsi.lr.v1(data = dat, Y.name = Y.name, X.name = X.name, Z.name = Z.name,
                      spline.num, spline.degree, initial.random.num)
```

---

si.coef.plot *plot estimated single index coefficients*

Description

plot estimated single index coefficients

Usage

```
si.coef.plot(si.coef.est)
```

Arguments

si.coef.est	A data set of estimated single index coefficients

Value

single index coefficient plot

Author(s)

Yuyan Wang

Examples

```
# example to plot estimated single index coefficients
data(nhanes.new)
dat <- nhanes.new

# specify variable names and parameters
Y.name <- "log.triglyceride"
X.name <- c("X1_trans.b.carotene", "X2_retinol", "X3_g.tocopherol", "X4_a.tocopherol",
            "X5_PCB99", "X6_PCB156", "X7_PCB206",
            "X8_3.3.4.4.5.pncb", "X9_1.2.3.4.7.8.hxcdf", "X10_2.3.4.6.7.8.hxcdf")
Z.name <- c("AGE.c", "SEX.Female", "RACE.NH.Black",
           "RACE.MexicanAmerican", "RACE.OtherRace", "RACE.Hispanic" )
spline.num = 5
spline.degree = 3
initial.random.num = 1

# run PLSI linear regression
set.seed(2023)
model_1 <- plsi.lr.v1(data = dat, Y.name = Y.name, X.name = X.name, Z.name = Z.name,
                      spline.num, spline.degree, initial.random.num)

# plot estimated single index coefficients
si.coef.plot(model_1$si.coefficient)

# check estimated single index coefficients
model_1$si.coefficient
```

si.fun.plot *plot estimated single index function*

Description

plot estimated single index function

Usage

```
si.fun.plot(si.ci)
```

Arguments

si.ci A data set of estimated index and corresponding single index values

Value

Single index function plot

Author(s)

Yuyan Wang

Examples

```
# example to plot estimated single index function
data(nhanes.new)
dat <- nhanes.new

# specify variable names and parameters
Y.name <- "log.triglyceride"
X.name <- c("X1_trans.b.carotene", "X2_retinol", "X3_g.tocopherol", "X4_a.tocopherol",
            "X5_PCB99", "X6_PCB156", "X7_PCB206",
            "X8_3.3.4.4.5.pncb", "X9_1.2.3.4.7.8.hxcdf", "X10_2.3.4.6.7.8.hxcdf")
Z.name <- c("AGE.c", "SEX.Female", "RACE.NH.Black",
           "RACE.MexicanAmerican", "RACE.OtherRace", "RACE.Hispanic" )
spline.num = 5
spline.degree = 3
initial.random.num = 1

# run PLSI linear regression
set.seed(2023)
model_1 <- plsi.lr.v1(data = dat, Y.name = Y.name, X.name = X.name, Z.name = Z.name,
                      spline.num, spline.degree, initial.random.num)

# plot single index function
si.fun.plot(model_1$si.fun)
```

Index