Package: BayesGWQS (via r-universe)

October 21, 2024

Type Package

Title Bayesian Grouped Weighted Quantile Sum Regression

Version 0.1.1

Author David Wheeler, Matthew Carli

Maintainer Matthew Carli <carlimm@mymail.vcu.edu>

Description Fits Bayesian grouped weighted quantile sum (BGWQS) regressions for one or more chemical groups with binary outcomes. Wheeler DC et al. (2019) <doi:10.1016/j.sste.2019.100286>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 4.0.0)

SystemRequirements JAGS

Imports coda, stats, rjags, stringr, plyr

Suggests testthat

NeedsCompilation no

Date/Publication 2022-01-20 22:42:54 UTC

Repository https://mmcarli.r-universe.dev

RemoteUrl https://github.com/cran/BayesGWQS

RemoteRef HEAD

RemoteSha cd612d26c0175d991e2ced1ebbd11630be153ad0

Contents

bgwqs.fit *Bayesian Grouped WQS Regression*

Description

This function fits a Bayesian grouped weighted quantile sum (BGWQS) regression model.

Usage

```
bgwqs.fit(
  y,
  x,
  z,
  x.s,
  n.quantiles = 4,
  working.dir,
  n.chains = 1,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 1,
  n.adapt = 500,
  DIC = FALSE
)
```

Arguments

`y`	A vector containing outcomes.
`x`	A matrix of component data.
`z`	A vector or matrix of controlling covariates.
`x.s`	A vector of the number of components in each index.
`n.quantiles`	The number of quantiles to apply to the component data.
`working.dir`	A file path to the directory.
`n.chains`	The number of Markov chains; must be a positive integer.
`n.iter`	The number of total iterations per chain, including burn in.
`n.burnin`	The number of iterations to discard at the beginning.
`n.thin`	The thinning rate; must be a positive integer.
`n.adapt`	The number of adaption iterations.
`DIC`	Logical; whether or not the user desires the function to return DIC.

Value

A list which includes BUGS output, sample chains post-burnin, and convergence test results.

Examples

```
## Not run:
data("simdata")
group_list <- list(c("pcb_118", "pcb_138", "pcb_153", "pcb_180", "pcb_192"),
                   c("as", "cu", "pb", "sn"),
                  c("carbaryl", "propoxur", "methoxychlor", "diazinon", "chlorpyrifos"))
x.s <- make.x.s(simdata, 3, group_list)
X <- make.X(simdata, 3, group_list)
Y <- simdata$Y
work_dir <- tempdir()
results <- bgwqs.fit(y = Y, x = X, x.s = x.s, n.quantiles=4,
                     working.dir = work_dir,
                 n.chains = 1, n.iter = 10000, n.burnin = 5000, n.thin = 1, n.adapt = 500)


## End(Not run)
```

`make.X` *Forms matrix of components*

Description

This function returns a matrix of component variables, X. The user can specify the desired chemicals and order by creating a list of string vectors, each vector containing the variable names of all desired elements of that group.

Usage

```
make.X(df, num.groups, groups)
```

Arguments

df	A dataframe containing named component variables
num.groups	An integer representing the number of component groups desired
groups	A list, each item in the list being a string vector of variable names for one component group

Value

A matrix of component variables

Examples

```
data("simdata")
group_list <- list(c("pcb_118", "pcb_138", "pcb_153", "pcb_180", "pcb_192"),
                   c("as", "cu", "pb", "sn"),
                  c("carbaryl", "propoxur", "methoxychlor", "diazinon", "chlorpyrifos"))
X <- make.X(simdata, 3, group_list)
X
```

make.x.s *Forms component group ID vector of X*

Description

This function returns a vector which lets WQS.fit know the size and order of groups in X

Usage

```
make.x.s(df, num.groups, groups)
```

Arguments

df	A dataframe containing named component variables
num.groups	An integer representing the number of component groups desired
groups	A list, each item in the list being a string vector of variable names for one component group

Value

A vector of integers, each integer relating how many columns are in each group

Examples

```
data("simdata")
group_list <- list(c("pcb_118", "pcb_138", "pcb_153", "pcb_180", "pcb_192"),
                   c("as", "cu", "pb", "sn"),
                  c("carbaryl", "propoxur", "methoxychlor", "diazinon", "chlorpyrifos"))
x.s <- make.x.s(simdata, 3, group_list)
x.s
```

simdata *Simulated data of chemical concentrations and one binary outcome variable*

Description

Data were simulated to have 0.7 in-group correlation and 0.3 between-group correlation. There are three groups, with the third being significantly correlated to the outcome variable.

Usage

```
simdata
```

Format

A data frame with 1000 rows and 15 variables:

pcb_118 a numeric vector; part of group 1

pcb_138 a numeric vector; part of group 1

pcb_153 a numeric vector; part of group 1

pcb_180 a numeric vector; part of group 1

pcb_192 a numeric vector; part of group 1

as a numeric vector; part of group 2

cu a numeric vector; part of group 2

pb a numeric vector; part of group 2

sn a numeric vector; part of group 2

carbaryl a numeric vector; part of group 3

propoxur a numeric vector; part of group 3

methoxychlor a numeric vector; part of group 3

diazinon a numeric vector; part of group 3

chlorpyrifos a numeric vector; part of group 3

Y a numeric vector; the outcome variable

`weight.plot` *Generates Plots of weights by group*

Description

This function takes the object created by the bgwqs.fit function and a vector of group names and generates a random forest variable importance plot for each group. The weights in each group are listed in descending order.

Usage

```
weight.plot(fit.object, group.names, group.list, x.s)
```

Arguments

`fit.object`	The object that is returned by the bgwqs.fit function
`group.names`	A string vector containing the name of each group included in the BGWQS regression. Will be used for plot titles.
`group.list`	A list, each item in the list being a string vector of variable names for one component group.
`x.s`	A vector of the number of components in each index.

Value

A plot for each group of the BGWQS regression

Index