Package 'gWQSRS'

August 30, 2019

Type Package

Title Generalized Weighted Quantile Sum Regression Random Subset

Version 1.0.0

Author Stefano Renzetti, Paul Curtin, Chris Gennings

Maintainer Stefano Renzetti <stefano.renzetti88@gmail.com>

Description Fits Weighted Quantile Sum Random Subset (WQSRS) regressions for continuous, binomial, multinomial and count outcomes.
Paul Curtin, Joshua Kellogg, Nadja Cech, Chris Gennings (2019) <doi:10.1080/03610918.2019.1577971>.

Imports Rsolnp, gWQS (>= 2.0.0), ggplot2, dplyr, stats, broom, rlist, MASS, reshape2, plotROC, knitr, kableExtra, nnet, future, future.apply, ggrepel, pscl

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-30 09:00:09 UTC

R topics documented:

gwqsrs *Fitting Weighted Quantile Sum Random Subset regression models*

Description

Fits Weighted Quantile Sum Random Subset (WQSRS) regressions for continuous, binomial, multinomial and count outcomes.

Usage

```
gwqsrs(formula, data, na.action, weights, mix_name, stratified, valid_var,
  rs = 100, n_vars = NULL, b1_pos = TRUE, b1_constr = FALSE,
  zero_infl = FALSE, q = 4, validation = 0.6, family = gaussian,
  zilink = c("logit", "probit", "cloglog", "cauchit", "log"),
  seed = NULL, pred = 0, plots = FALSE, tables = FALSE,
  plan_strategy = "sequential", control = list(rho = 1, outer.iter =
  400, inner.iter = 800, delta = 1e-07, tol = 1e-08, trace = 0))
```

Arguments

formula	An object of class `formula` specifying the relationship to be tested. The `wqs` term must be included in `formula`, e.g. `y ~ wqs + ...`. To test for an interaction term with a continuous variable `a` or for a quadratic term we can specify the `formula` as below: `y ~ wqs*a + ...` and `y ~ wqs + I(wqs^2) + ...`, respectively.
data	The `data.frame` containing the variables to be included in the model.
na.action	`model.frame`. `na.omit` is the default.
weights	an optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector.
mix_name	A character vector listing the variables contributing to a mixture effect.
stratified	The character name of the variable for which you want to stratify for. It has to be a `factor`.
valid_var	A character value containing the name of the variable that identifies the validation and the training dataset. You previously need to create a variable in the dataset which is equal to 1 for the observations you want to include in the validation dataset, equal to 0 for the observation you want to include in the training dataset (use 0 also for the validation dataset if you want to train and validate the model on the same data) and equal to 2 if you want to keep part of the data for the predictive model.
rs	Number of random subset samples used in parameter estimation.
n_vars	The number of mixture components to be included at each random subset step.
b1_pos	A logical value that determines whether weights are derived from models where the beta values were positive or negative.

b1_constr	A logial value that determines whether to apply positive (if b1_pos = TRUE) or negative (if b1_pos = FALSE) constraints in the optimization function for the weight estimation.
zero_infl	A logical value (TRUE or FALSE) that allows to fit a zero inflated model in case family = "poisson" or family = "negbin".
q	An integer to specify how mixture variables will be ranked, e.g. in quartiles (q = 4), deciles (q = 10), or percentiles (q = 100). If q = NULL then the values of the mixture variables are taken (these must be standardized).
validation	Percentage of the dataset to be used to validate the model. If validation = 0 then the test dataset is used as validation dataset too.
family	A character value that allows to decide for the glm: "gaussian" for linear regression, "binomial" for logistic regression "multinomial" for multinomial regression, "poisson" for Poisson regression, "quasi-poisson" for quasi-Poisson regression, "negbin" for negative binomial regression.
zilink	character specification of link function in the binary zero-inflation model (you can choose among "logit", "probit", "cloglog", "cauchit", "log").
seed	An integer value to fix the seed, if it is equal to NULL no seed is chosen.
pred	Percentage of the dataset to be used for the predictive model. If pred = 0 then no predicitve model is going to be built.
plots	A logical value indicating whether plots should be generated with the output (plots = TRUE) or not (plots = FALSE).
tables	A logical value indicating whether tables should be generated in the output (tables = TRUE) or not (tables = FALSE).
plan_strategy	A character value that allows to choose the evaluation strategies for the plan function. You can choose among "sequential", "transparent", "multisession", "multicore", "multiprocess", "cluster" and "remote" (see plan help page for more details).
control	The control list of optimization parameters. See solnp for details.

Details

gWQS uses the glm function in the **stats** package to fit the linear, logistic, the Poisson and the quasi-Poisson regression, while the glm.nb function from the **MASS** package is used to fit the negative binomial regression respectively. The nlm function from the **stats** package was used to optimize the log-likelihood of the multinomial regression.

The solnp optimization function is used to estimate the weights in each random subset sample.

The seed argument specifies a fixed seed through the set.seed function.

The plots argument produces three figures (two if family = binomial or "multinomial") through the ggplot function. One more plot will be printed if pred > 0 and family = binomial.

The tables argument produces two tables in the viewr pane through the use of the functions kable and kable_styling.

Value

gwqsrs return the results of the WQSRS regression as well as many other objects and datasets.

fit	The object that summarizes the output of the WQSRS model, reflecting a linear, logistic, multinomial, Poisson, quasi-Poisson or negative binmial regression depending on how the family parameter was specified. The summary function can be used to call and print fit data (not for multinomial regression).
conv	Indicates whether the solver has converged (0) or not (1 or 2).
bres	Matrix of estimated weights, mixture effect parameter estimates and the associated standard errors, statistics and p-values estimated for each bootstrap iteration.
wqs	Vector containing the wqs index for each subject.
q_i	List of the cutoffs used to divide in quantiles the variables in the mixture
slctd_vars	List of vectors containing the names of the mixture components selected at each random subset step.
tindex	Vector containing the rows used to estimate the weights in each random subset.
vindex	Vector containing the rows used to estimate the parameters of the final model.
final_weights	data.frame containing the final weights associated to each chemical.
y_wqs_df	data.frame containing the dependent variable values adjusted for the residuals of a fitted model adjusted for covariates (original values when family = binomial or "multinomial") and the wqs index estimated values.
df_pred	data.frame containing the variables to print the ROC curve. It is generated only when pred > 0
pindex	Vector containing the subjects used for prediction. It is generated only when pred > 0

Author(s)

Stefano Renzetti, Paul Curtin, Chris Gennings

References

Paul Curtin, Joshua Kellogg, Nadja Cech & Chris Gennings (2019): A random subset implementation of weighted quantile sum (WQSRS) regression for analysis of high-dimensional mixtures, Communications in Statistics - Simulation and Computation, DOI: 10.1080/03610918.2019.1577971. https://doi.org/10.1080/03610918.2019.1577971.

Examples

```
# we save the names of the mixture variables in the variable "toxic_chems"
toxic_chems = c("log_LBX074LA", "log_LBX099LA", "log_LBX105LA", "log_LBX118LA",
"log_LBX138LA", "log_LBX153LA", "log_LBX156LA", "log_LBX157LA", "log_LBX167LA",
"log_LBX170LA", "log_LBX180LA", "log_LBX187LA", "log_LBX189LA", "log_LBX194LA",
"log_LBX196LA", "log_LBX199LA", "log_LBXD01LA", "log_LBXD02LA", "log_LBXD03LA",
"log_LBXD04LA", "log_LBXD05LA", "log_LBXD07LA", "log_LBXF01LA", "log_LBXF02LA",
```

```
"log_LBXF03LA", "log_LBXF04LA", "log_LBXF05LA", "log_LBXF06LA", "log_LBXF07LA",
"log_LBXF08LA", "log_LBXF09LA", "log_LBXPCBLA", "log_LBXTCDLA", "log_LBXHXCLA")

# To run a linear model and save the results in the variable "results". This linear model
# (family="gaussian") will rank/standardize variables in quartiles (q = 4), perform a
# 40/60 split of the data for training/validation (validation = 0.6), and estimate weights
# over 10 random subset samples (rs = 10; in practical applications at least 1000 random
# subsets should be used). The number of chemicals to be included at each random subset
# step is left to the function which automatically chooses the rounded square root of the
# toxic_chems vector's length (n_vars = NULL). Weights will be derived from mixture effect
# parameters that are positive (b1_pos = TRUE). A unique seed was specified (seed = 2016)
# so this model will be reproducible, and plots describing the variable weights and linear
# relationship will be generated as output (plots = TRUE). In the end tables describing the
# weights values and the model parameters with the respectively statistics are generated in
# the plots window
results = gwqsrs(y ~ wqs, mix_name = toxic_chems, data = wqs_data, q = 4,
               validation = 0.6, rs = 10, n_vars = NULL, b1_pos = TRUE, b1_constr = FALSE,
                 family = gaussian, seed = 2018, plots = TRUE, tables = TRUE)

# to test the significance of the covariates
summary(results$fit)
```

---

wqs_data *Exposure concentrations of 34 PCB (simulated dataset)*

Description

We created the 'wqs_data' dataset to show how to use this function. These data reflect 34 exposure concentrations simulated from a distribution of PCB exposures measured in subjects participating in the NHANES study (2001-2002). Additionally, an end-point meaure, simulated from a distribution of leukocyte telomere length (LTL), a biomarker of chronic disease, is provided as well (variable name: y), as well as simulated covariates, e.g. sex, and a dichotomous outcome variable (variable name: disease_state). This dataset can thus be used to test the 'gWQS' package by analyzing the mixed effect of the 34 simulated PCBs on the continuous or binary outcomes, with adjustments for covariates.

Usage

```
wqs_data
```

Format

A data frame with 500 rows and 37 variables

Details

y continuous outcome, biomarker of chronic disease

disease_state dichotomous outcome, state of disease

sex covariate, gender of the subject

log_LBX 34 exposure concentrations of PCB exposures ...

Index