Package 'lwqs'

March 4, 2021

Type Package

Title Lagged Weighted Quantile Sum Regression

Version 0.5.0

Description Wrapper functions for the implementation of lagged weighted quantile sum regression, as per 'Gennings et al' (2020) <doi:10.1016/j.envres.2020.109529>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports data.table, ggplot2, plyr, gridExtra, gWQS, gamm4

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Paul Curtin [aut, cre],
Stefano Renzetti [aut],
Chris Gennings [aut]

Maintainer Paul Curtin <paul.curtin@mssm.edu>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2021-03-04 10:00:02 UTC

R topics documented:

extract_mixture 2
extract_weights 3
lwqs 4
lwqs_data 5

Index **7**

`extract_mixture` *Function to extract time-varying mixture (wqs) index from lWQS object*

Description

Function to extract time-varying mixture (wqs) index from lWQS object

Usage

```
extract_mixture(lobj)
```

Arguments

`lobj`	An object returned from lWQS function

Value

Data frame containing the time index, wqs index estimated at each repeated measure, subject ID, and the outcome variable.

Examples

```
# identify predictor variables used in mixture
mixvars=names(lwqs_data)[5:9]

# run model. Note for example run-time only 1 bootstrap (b=1) is used. Set b to be >50
model=lwqs(data=lwqs_data,
           timevar="time",
           wqs_parms=list(formula=out ~ wqs,
              data = lwqs_data,
              mix_name=mixvars,
              b1_constr = TRUE,
              b1_pos=FALSE,
              b = 5,
              q = 5,
              validation = 0,
              family = "gaussian",
              seed = 1),
              outcome="out",
              ID="ID")

# use extract_mixture to access time-varying wqs index
mixtime=extract_mixture(model)
```

extract_weights *Function to extract time-varying weights from lWQS object*

Description

Function to extract time-varying weights from lWQS object

Usage

```
extract_weights(lobj)
```

Arguments

lobj	An object returned from lWQS function

Value

A (long-form) data frame containing the time index and corresponding variable weights estimated in an lWQS

Examples

```
# identify predictor variables used in mixture
mixvars=names(lwqs_data)[5:9]

# run model
model=lwqs(data=lwqs_data,
           timevar="time",
           wqs_parms=list(formula=out ~ wqs,
              data = lwqs_data,
              mix_name=mixvars,
              b1_constr = TRUE,
              b1_pos=TRUE,
              b = 5,
              q = 5,
              validation = 0,
              family = "gaussian",
              seed = 1),
              outcome="out",
              ID="ID")

# use extract_weights to access time-varying predictor weights
timeweights=extract_weights(model)
```

`lwqs` *Wrapper function for the implementaion of lagged WQS.*

Description

Wrapper function for the implementaion of lagged WQS.

Usage

```
lwqs(
  data,
  timevar,
  wqs_parms,
  outcome,
  ID,
  rDLM_parms = list(formula = wqs ~ s(time, by = y, bs = "cr"), random = ~(1 | id))
)
```

Arguments

`data`	Data frame containing observations in long format.
`timevar`	Enquoted variable name identifying the repeated measure / time variable
`wqs_parms`	A list containing parameters to be passed to the WQS algorithm. See gWQS package for details.
`outcome`	An enquoted variable name identifying the outcome measure
`ID`	An enquoted variable name identifying the subject identifier
`rDLM_parms`	(optional). A list containing parameters to be passed to the GAM algorithm. See gamm4 package for details. Parameters wqs, time, by, and id (see above) are created by the lwqs function and passed to the gamm4 function automatically.

Value

The `lwqs` function returns a list containing final model output and time-specific model parameters.

`parameters`	This list contains several objects summarizing different stages of the lagged ensemble model. The first object, `res`, contains output from the gWQS algorithm applied to each discreet repeated measure in the overall model; see package gWQS for details. The second output, `wqstime`, provides the mixture index, identified as "wqs", estimated for each subject at each discrete time point. The third item, `weightstime`, provides the weights estimated for each predictor at each discrete time point.
`plot`	This list contains two plots (as grobs) which summarize output of the lwqs algorithm.

Examples

```
 # identify predictor variables used in mixture
mixvars=names(lwqs_data)[5:9]

model=lwqs(data=lwqs_data,
           timevar="time",
           wqs_parms=list(formula=out ~ wqs,
              data = lwqs_data,
              mix_name=mixvars,
              b1_constr = TRUE,
              b1_pos=TRUE,
              b = 5,
              q = 5,
              validation = 0,
              family = "gaussian",
              seed = 1),
              outcome="out",
              ID="ID")
```

`lwqs_data` *Simulated dataset for accompanying vignette*

Description

Simulated dataset for accompanying vignette

Usage

```
data(lwqs_data)
```

Value

A data frame containing simulated data to explore the `lwqs` algorithm. Variables included are as follows:

`ID`	Variable identifying each simulated subject. Data reflect 30 successive measures per subject.
`Sex`	A simulated binary covariate, either 1 or 0.
`time`	Variable identifying the successive timing of each repeated measure
`out`	Simulated outcome on standardized scale
`pred1`	First simulated time-varying predictor. This has a large positive association with "out" from times 11-20.
`pred2`	Second simulated time-varying predictor. This has a moderate positive association with "out" from times 11-20.
`pred3`	Third simulated time-varying predictor. This has a moderate negative association with "out" from times 1-10.

pred4	Fourth simulated time-varying predictor. This has a strong negative association with "out" from times 1-10.
pred5	Fifth simulated time-varying predictor. This has no significant association with "out".

Index

extract_mixture, 2
extract_weights, 3

lwqs, 4
lwqs_data, 5