Package: repolr (via r-universe)

October 15, 2026

Type Package

Title Repeated Measures Proportional Odds Logistic Regression

Version 3.4

Date 2016-02-26

Author Nick Parsons

Maintainer Nick Parsons <nick.parsons@warwick.ac.uk>

Description Fits linear models to repeated ordinal scores using GEE methodology.

License GPL-3

Imports Rcpp (>= 0.11.3), Matrix, methods

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

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

Date/Publication 2016-02-27 10:11:34 UTC

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

RemoteRef HEAD

RemoteSha dd36501d4283a5f53394212217b633377596fcea

Contents

repolr-package 2
achilles 5
begonia 6
confint.repolr 7
HHSpain 8
mobility 8
ord.expand 9
polycuts 10
predict.repolr 11
QIC 12
QoL 13

summary.repolr 13
vcov.repolr 14
work.corr 15

Index **16**

`repolr-package` *Repeated Measures Proportional Odds Logistic Regression using GEE*

Description

The package allows regression models to be fitted to repeated ordinal scores, for the proportional odds model, using a generalized estimating equation (GEE) methodology. The algorithm estimates the correlation parameter by minimizing the generalized variance of the regression parameters at each step of the fitting algorithm. Parameter estimation is available for the uniform and first-order autoregressive correlation models, for data potentially recorded at irregularly spaced time intervals. A test for the proportional odds assumption is available, as is an option to fit a polynomial model to the the cut-point parameters.

Usage

```
repolr(formula, subjects, data, times, categories, corr.mod = "independence",
         alpha = 0.5, po.test = FALSE, fixed = FALSE, poly = NULL,
         space = NULL, diffmeth = "analytic", fit.opt = rep(NA, 5))
```

Arguments

`formula`	a formula, as for other regression models.
`subjects`	a character string specifying the name of the subject variable.
`data`	a dataframe in which to interpret the variables occurring in the formula.
`times`	a vector of times which occur within subject clusters; e.g. for four evenly spaced times c(1, 2, 3, 4).
`categories`	a numeric variable indicating the number of ordinal score categories.
`corr.mod`	a character string specifying the correlation structure. The following are permitted: "`ar1`", "`uniform`" and "`independence`".
`alpha`	an initial value for the correlation parameter.
`po.test`	a logical variable; if true a score test for proportional odds is reported.
`fixed`	a logical variable; if true the correlation is fixed at the initial value (`alpha`) during model fitting.
`poly`	a numeric variable indicating the order of the polynomial contrasts used for the cut-point model.
`space`	a vector indicating the category spacing when fitting the polynomial model; can generally be set to 1:`categories`
`diffmeth`	a character string specifying the method used for estimation of `alpha`. The folowing are available:"`analytic`" and "`numeric`"
`fit.opt`	a vector of options to control the behaviour of the fitting algorithm.

Details

The `repolr` function fits models for repeated ordinal scores using GEE methodology.

The user is required to specify, as a minimum: (i) a data set name (`data`), (ii) a model formula (`formula`), (iii) a cluster identification variable (`subjects`), (iv) a time variable (`time`) and (v) the number of categories used for the response variable (`categories`).

The data may contain records with missing data for either the response variable or the explanatory variables. The response variable must have at least three ordered categories (K greater than or equal to 3) and, as K-1 cut-point parameters are estimated, an additional intercept term can not be explicitly included in the model formula. A subject variable, which takes integer values from 1 to N with no missing values allowed, indicates the data clusters (patients or experimental units) and a time variable indicates the within cluster ordering; times must be ordered integers starting from one and spaced to indicate the relative distance between successive times. For instance, four observations at equally spaced times would be entered as 1, 2, 3 and 4, whereas if the first two observations were separated by half the time interval of the other observations then coding would be 1, 2, 4 and 6. The data must be pre-sorted by time clusters within each subject, and complete, i.e. where data is missing for a particular time it must be indicated as such. the datasets provided with this package provide exemplars of the required data formatting; e.g. `HHSpain` and `QoL`.

The available options for the correlation model (`corstr`) are AR1, uniform, fixed and independence, with default setting independence.

Additionally there are a number of other algorithm related options.

The algorithm is generally robust to the initial value for `alpha` (default setting = 0.5), where estimation is rerquired, however a starting value for `alpha` can be set. If required the correlation parameter, set via `alpha`, can be fixed throughout model fitting, and not updated, by setting the option `fixed` to `TRUE`.

The partial derivatives of the log of the determinant of the robust variance matrix (generalized variance), with respect to `alpha`, can either be determined analytically ("`analytic`" setting for `diffmeth`) or numerically by finite differencing ("`numeric`" setting for `diffmeth`). The latter method is often quicker for complex regression models, or if K is large.

Function `poly`, specifies the order of fitted orthogonal polynomial contrasts for the cut-point parameters; the default setting for **repolr** is to fit the complete set of cut-point parameters. Fitting polynomial contrasts can be particularly useful for long ordinal scores, where K is large, or where a particular form for the cut-point parameters is preferred; e.g. increasing uniformly at the extremes of the score scale. The order of the polynomial must be an integer less than K-1. The function requires one additional argument, `space`, that indicates the spacing between categories. This is normally set to 1:`categories` when all categories are observed, but can be modified to provide more realsitic models if for instance some categories are not observed.

The proportional odds assumption can be tested, using a score test, by setting the option `po.test` to TRUE.

The `fit.opt` argument provides control of the fitting algorithm; the defaults are

`c(cmaxit = 10, omaxit = 5, ctol = 0.001, otol = 0.00001, h = 0.01)`.

These control the maximum number of iterations for updating estimates of `alpha`, the maximum number of iterations for updating the regression parameters within each of the updating steps for `alpha`, the convergence tolerances for estimation of `alpha` and the regression parameters, and the interval `h` for finite differencing, if the "`numeric`" option is selected.

Model fitting is implemented via a suite of of functions developed using **RcppArmadillo** (**Rcpp**) that construct correlations between derived binary variables at each time-point (`smat`) and between time-points (`cmat`). Complete (sparse) model covariance matrices are constructed using `hgmat` and `alphpow`, with model parameter estimation implemented in `ordgee`, using current estimates of the inverse of the correlation matrix from `icormat`. Function `upalpha` provides updates of correlation parameter estimates, and `potest` implements the test of proportional odds. These functions are not documented in detail here as they are primarily for internal use within **repolr**. There use outside of this setting is not recommended.

Value

The function `summary.repolr` is used to obtain and print a summary of the fitted model.

The fitted model is an object of class "`repolr`" and has the following values:

`poly.mod`	polynomial model for cut points: a `list` with elements `poly`, `polycuts` and `space`.
`y`	the response variable.
`linear.predictors`	a vector of linear predictors.
`fitted.values`	a vector of the fitted values.
`coefficients`	a named vector of regression coefficients.
`robust.var`	the robust (sandwich) variance matrix.
`naive.var`	the naive variance matrix.
`alpha`	an estimate of the correlation parameter.
`convergence`	a logical variable to reported if convergence was achieved.
`iter`	the number of iterations.
`grad1`	first derivative of generalized variance at convergence.
`grad2`	second derivative of generalized variance at convergence.
`crit`	convergence criterion.
`po.test`	results of po test: a `list` with elements `po.stat`, `po.df` and `po.chi`.

References

Parsons NR, Costa ML, Achten J, Stallard N. Repeated measures proportional odds logistic regression analysis of ordinal score data in the statistical software package R. *Computational Statistics and Data Analysis* 2009; 53:632-641.

Parsons NR, Edmondson RN, Gilmour SG. A generalized estimating equation method for fitting autocorrelated ordinal score data with an application in horticultural research. *Journal of the Royal Statistical Society C* 2006; 55:507-524.

Stiger TR, Barnhart HX, Williamson JM. Testing proportionality in the proportional odds model fitted with GEE. *Statistics in Medicine* 1999; 18:1419-1433.

Parsons NR. Proportional-odds models for repeated composite and long ordinal outcome scales. *Statistics in Medicine* 2013; 32:3181-3191.

See Also

QIC, polycuts, work.corr

Examples

```
data(HHSpain)
mod.0 <- repolr(HHSpain~Sex*Time, data=HHSpain, categories=4, subjects="Patient",
            times=c(1,2,5), corr.mod="uniform", alpha=0.5)
summary(mod.0)
summary(update(mod.0, diffmeth = "numeric"))
summary(update(mod.0, fixed = TRUE, alpha =0.5))
```

---

achilles *Achilles Tendon Rupture*

Description

EuroQol activity scores, measured on a three point ordinal score scale, post-surgery for 48 patients at three occasions.

Usage

```
data(achilles)
```

Format

A data frame with 144 observations on the following four variables.

`Patient` a patient identifier variable.

`Treat` post-surgery treatments are either immediate mobilisation in a carbon-fibre orthosis with three 1.5cm heel raises (1) or traditional plaster cast immobilisation (2).

`Time` recorded at baseline (1), six months (2) and one year (3) post-surgery.

`Activity` ability to undertake usual activities post-surgery; this was scored by each patient as either no problem (1), some problem (2) or an inability (3) to perform usual activity (e.g. work, leisure, housework etc).

Source

Costa, M.L., MacMillan, K., Halliday, D., Chester, R., Shepstone, L., Robinson, A.H.N., Donell, S.T. (2006). Randomised controlled trials of immediate weight-bearing mobilisation for rupture of the tendon Achillis. *Journal of Bone and Joint Surgery (British)* 88-B, 69-77.

Examples

```
data(achilles)
head(achilles)
```

`begonia` *Begonia Pot Plant Quality Scores*

Description

Begonia pot plant quality scores (for 2 varieties and 3 transport chains), during 5 weeks in simulated shelf-life conditions (temperature, light and irrigation). Quality scores were originally made on an ordinal scale from 1 to 10 (highest quality). However, only categories 2 to 9 were used, so these were re-coded to scale from 1 to 8. In addition to overall quality scores, a range of plant physiological characteristics were also observed.

Usage

```
data(begonia)
```

Format

A data frame with 720 observations on the following variables.

`Pot` pot plant identifier; 1 to 144.

`Plot` location in growing compartment; 1 to 48.

`Week` week number in simulated shelf-life; 1 to 5.

`Temp` temperature in compartment; `16C` or `21C`.

`Light` light level in compartment; `High` or `Low`.

`Chain` transport chain; `Cold`, `Comm` (commercial) or `Optm` (optimum).

`Irrig` irrigation in compartment; `Cont` (control) or `Fluct` (fluctuating).

`Variety` variety; `Balli` or `Batik`.

`Qual` quality score; 1 to 8.

`FDrop` count of dropped flowers.

`CBDrop` count of dropped coloured buds.

`GBDrop` count of dropped green buds.

`FDam` count of damaged flowers.

`FSing` count of single flowers.

`FDoub` count of double flowers.

`LYell` count of yellow leaves.

Source

Parsons NR, Edmondson RN, Gilmour SG. A generalized estimating equation method for fitting autocorrelated ordinal score data with an application in horticultural research. *Journal of the Royal Statistical Society C* 2006; 55:507-524.

Examples

```
data(begonia)
head(begonia)
```

---

confint.repolr *Confidence Intervals for repolr Model Parameters*

Description

Computes confidence intervals for one or more parameters in a fitted `repolr` model object.

Usage

```
## S3 method for class 'repolr'
confint(object, parm, level = 0.95, robust.var = TRUE, ...)
```

Arguments

`object`	is a model fitted using `repolr`.
`parm`	a specification of which parameters are to be used, either a vector of numbers or a vector of names. If missing, all parameters are considered.
`level`	the confidence level required.
`robust.var`	logical; if `TRUE`, intervals are based on the robust variance matrix.
`...`	further arguments passed to or from other methods.

Details

The method assumes normality and uses as default the estimated robust variance matrix.

Value

A matrix (or vector) with columns giving lower and upper confidence limits for each parameter.

Examples

```
data(HHSpain)
mod.0 <- repolr(HHSpain~Sex*Time, data=HHSpain, categories=4, subjects="Patient",
            times=c(1,2,5), corr.mod="uniform", alpha=0.5)
confint(mod.0, robust.var = FALSE)
```

HHSpain *Harris Hip Pain Scores*

Description

Harris hip pain scores, measured on a four point ordinal score scale, post-surgery for 58 patients at three occasions.

Usage

```
data(HHSpain)
```

Format

A data frame with 174 observations on the following four variables.

`Patient` a patient identifier variable.

`Sex` a factor with levels `F` or `M`

`Time` recorded at baseline (1), two years (2) and five years (5) post-surgery.

`HHSpain` a pain score coded as none (1), slight (2), mild (3) and moderate or marked (4).

Examples

```
data(HHSpain)
head(HHSpain)
```

mobility *Mobility after Hip Fracture Fixation Surgery*

Description

Patient mobility score after hip fracture fixation surgery, using two distinct procedures.

Usage

```
data(mobility)
```

Format

A data frame with 600 observations on the following variables.

`subject` patient identifier.

`time` assessemnt occasions; month 1, 2, 3 and 4.

`treat` intervention group; `A` or `B`.

`age` patient age in years.

`gender` gender; `F` or `M`.

`mobility` mobility score; 1, 2 and 3.

Examples

```
data(mobility)
head(mobility)
```

`ord.expand` *Expand Ordinal Data for Model Fitting with repolr*

Description

Expands ordinal score data into an appropriate form, K-1 new binary variables for an ordinal score with K categories, for model fitting using `repolr`.

Usage

```
ord.expand(space, formula, times, poly, data, subjects, categories)
```

Arguments

`space`	a vector indicating the category spacing when fitting the polynomial model.
`formula`	a formula, as for other regression models.
`times`	a vector of times which occur within subject clusters.
`poly`	a numeric variable indicating the order of the polynomial contrasts used for the cut-point model.
`data`	a dataframe in which to interpret the variables occurring in the formula.
`subjects`	a character string specifying the name of the subject variable.
`categories`	a numeric variable indicating the number of ordinal score categories.

Details

For internal use with `repolr`.

Value

An expanded dataframe.

polycuts *Estimates Cut-point Parameters for Fitted repolr Model*

Description

After fitting a model using `repolr`, function `polycuts` gives estimates and standard errors for the K-1 cut-point parameters, based on the polynomial model from the fit of `repolr`. Polynomial cut-point parameter estimates from the orginal model are also shown.

Usage

```
polycuts(object, digits = 3, robust.var = TRUE)
```

Arguments

`object`	is a model fitted using `repolr`.
`digits`	the number of decimal places to display in reported summaries.
`robust.var`	a logical variable: if `TRUE` standard errors are based on robust variance estimates, otherwise naive estimates are used.

Value

`coef`	polynomial parameter estimates from `repolr`.
`poly`	a vector of K-1 cut-point parameters.
`order`	the order of the polynomial.

Examples

```
data(HHSpain)
mod.0 <- repolr(HHSpain~Sex*Time, data=HHSpain, categories=4, subjects="Patient",
            times=c(1,2,5), corr.mod="uniform", alpha=0.5)
summary(mod.0)
mod.1 <- update(mod.0, poly=1)
summary(mod.1)
polycuts(mod.1)
mod.2 <- update(mod.0, poly=2)
summary(mod.2)
polycuts(mod.2)
```

predict.repolr *Predict Method for Fitted repolr Model*

Description

Calculates predictions and standard errors of predictions for a fitted repolr model object.

Usage

```
## S3 method for class 'repolr'
predict(object, newdata = NULL, se.fit = FALSE,
           robust.var = TRUE, type = c("link", "response", "terms"), ...)
```

Arguments

object	is a model fitted using repolr.
newdata	optionally, a data frame in which to find variables with which to predict; if missing the model fitted values are reported.
se.fit	Logical indicating if standard errors are required.
robust.var	logical; if TRUE, standard errors are based on the robust variance matrix.
type	is the type of prediction required. The default "link" is to use the scale of the linear predictors; i.e. the log-odds of cumulative probabilities. The alternative is to report the predicted cumulative probabilities; "response". The "terms" option returns the matrix of fitted values for each model term on the scale of the linear predictor.
...	further arguments passed to or from other methods.

Details

If newdata is missing predictions are based on the data used to fit the repolr model. If newdata are supplied then the format of these data must conform to the same format required for model fitting using repolr. See repolr for details.

Value

fit	Predictions.
se.fit	Estimated standard errors.

Examples

```
data(HHSpain)
mod.0 <- repolr(HHSpain~Sex*Time, data=HHSpain, categories=4, subjects="Patient",
           times=c(1,2,5), corr.mod="uniform", alpha=0.5)
predict(mod.0, newdata = data.frame(Patient = rep(100, 3), Time = c(1, 2, 5),
       Sex = factor(rep(1, 3), levels=1:2, labels=c("F", "M"))),
       type="link", se.fit = TRUE)
```

QIC *Quasilikelihood Information Criterion*

Description

The quasilikelihood information criterion (QIC) developed by Pan (2001) is a modification of the Akaike information criterion (AIC) for models fitted by GEE. QIC is used for choosing the best correaltion structure and QICu is used for choosing the best subset of covariates. The quasilikelihood (QLike) is also reported for completeness. When choosing between two or more models, with different subset of covariates, the one with the smallest QICu measure is preferred and similarly, when choosing between competing correlation structures, with the same subset of covariates in both, the model with the smallest QIC measure is preferred.

Usage

```
QIC(object, digits = 3)
```

Arguments

object	is a fitted model using repolr.
digits	the number of decimal places to display in reported summaries.

Value

QLike	model quasilikelihood.
QIC	model QIC.
QICu	model QICu.

References

Pan W. Akaikes information criterion in generalized estimating equations. *Biometrics* 2001; 57:120-125.

Examples

```
data(HHSpain)
mod.0 <- repolr(HHSpain~Time, data=HHSpain, categories=4, subjects="Patient",
            times=c(1,2,5), corr.mod="independence", alpha=0.5)
QIC(mod.0)
QIC(update(mod.0, formula = HHSpain~Time + Sex))$QICu
QIC(update(mod.0, formula = HHSpain~Time * Sex))$QICu
```

QoL *Quality of Life Scores*

Description

Quality of life scores after hip replacement.

Usage

```
data(QoL)
```

Format

A data frame with 336 observations on the following four variables.

`QoL` a numeric vector.

`Patient` a numeric vector.

`Time` a numeric vector.

`Treat` a factor with levels `A` `B`.

Examples

```
data(QoL)
head(QoL)
```

summary.repolr *Summary of Fitted repolr Model*

Description

Function to summarise the fit of a repolr model.

Usage

```
## S3 method for class 'repolr'
summary(object, digits, robust.var, ...)
```

Arguments

`object`	fitted model.
`digits`	integer for number formatting.
`robust.var`	logical; if `TRUE`, standard errors are based on robust variance estimates
`...`	further arguments passed to or from other methods.

Details

Default is to use robust variance estimates. However, if `robust.var` is set to `FALSE`, naive variance estimates are used.

`vcov.repolr` *Calculates the Variance-Covariance Matrix for Fitted repolr Model*

Description

Returns the variance-covariance matrix of the main parameters of a fitted `repolr` model object.

Usage

```
## S3 method for class 'repolr'
vcov(object, robust.var = TRUE, ...)
```

Arguments

`object`	is a model fitted using `repolr`.
`robust.var`	logical; if `TRUE`, outputs the robust variance matrix.
`...`	further arguments passed to or from other methods.

Details

Default is to output the estimated robust variance matrix. However, if `robust.var` is set to `FALSE`, the naive variance matrix is reported.

Value

A matrix of the estimated covariances between the parameter estimates.

Examples

```
data(HHSpain)
mod.0 <- repolr(HHSpain~Sex*Time, data=HHSpain, categories=4, subjects="Patient",
            times=c(1,2,5), corr.mod="uniform", alpha=0.5)
vcov(mod.0, robust.var = FALSE)
```

work.corr *Working Correlation Matrix for Fitted repolr Model*

Description

The working correlation matrix for the selected model; "ar1", "uniform" or "independence".

Usage

```
work.corr(object, digits = 3)
```

Arguments

object	is a model fitted using repolr.
digits	integer for number formatting.

Value

A $T(K\text{-}1)$ correlation matrix.

Examples

```
data(HHSpain)
mod.0 <- repolr(HHSpain~Sex*Time, data=HHSpain, categories=4, subjects="Patient",
            times=c(1,2,5), corr.mod="uniform", alpha=0.5)
work.corr(mod.0)
```

Index

∗ **datasets**
achilles, 5
begonia, 6
HHSpain, 8
mobility, 8
QoL, 13
∗ **models**
ord.expand, 9
polycuts, 10
QIC, 12
repolr-package, 2

achilles, 5
alphpow *(*repolr-package*)*, 2

begonia, 6

cmat *(*repolr-package*)*, 2
coef.repolr *(*repolr-package*)*, 2
confint.repolr, 7

fitted.repolr *(*repolr-package*)*, 2

hgmat *(*repolr-package*)*, 2
HHSpain, *3*, 8

icormat *(*repolr-package*)*, 2

mobility, 8
model.matrix.repolr *(*repolr-package*)*, 2

ord.expand, 9
ordgee *(*repolr-package*)*, 2

polycuts, *5*, 10
potest *(*repolr-package*)*, 2
predict.repolr, 11
print.repolr *(*repolr-package*)*, 2
print.summary.repolr *(*summary.repolr*)*, 13

QIC, *5*, 12
QoL, *3*, 13

repolr, *11*
repolr *(*repolr-package*)*, 2
repolr-package, 2
residuals.repolr *(*repolr-package*)*, 2

smat *(*repolr-package*)*, 2
summary.repolr, 13

terms.repolr *(*repolr-package*)*, 2

upalpha *(*repolr-package*)*, 2

vcov.repolr, 14

work.corr, *5*, 15