Package ‘qvcalc’

February 15, 2020

Version 1.0.2

Date 2020-02-14

Title Quasi Variances for Factor Effects in Statistical Models

Author David Firth

Maintainer David Firth <d.firth@warwick.ac.uk>

URL https://davidfirth.github.io/qvcalc

Description Functions to compute quasi variances and associated measures of approximation error.

Suggests relimp, MASS

Enhances psychotools, survival

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-02-15 21:20:05 UTC

R topics documented:

indentPrint 2
plot.qv 2
qvcalc 4
worstErrors 8

Index **10**

indentPrint — *Print with Line Indentation*

Description

Same as `print`, but adds a specified amount of white space at the start of each printed line

Usage

```
indentPrint(object, indent=4, ...)
```

Arguments

`object`	any printable object
`indent`	a non-negative integer, the number of spaces to insert
`...`	other arguments to pass to `print`

Value

`object` is returned invisibly

Author(s)

David Firth, <d.firth@warwick.ac.uk>

Examples

```
indentPrint("this indented by 10 spaces", indent=10)
```

plot.qv — *Plot method for objects of class qv*

Description

Provides visualization of estimated contrasts using intervals based on quasi standard errors.

Usage

```
## S3 method for class 'qv'
plot(x, intervalWidth = 2, ylab = "estimate",
    xlab = "", ylim = NULL,
    main = "Intervals based on quasi standard errors",
    levelNames = NULL, ...)
```

Arguments

`x`	an object of class `"qv"`, typically the result of calling `qvcalc`
`intervalWidth`	the half-width, in quasi standard errors, of the plotted intervals
`ylab`	as for `plot.default`
`xlab`	as for `plot.default`
`ylim`	as for `plot.default`
`main`	as for `plot.default`
`levelNames`	labels to be used on the x axis for the levels of the factor whose effect is plotted
`...`	other arguments understood by `plot`

Details

If `levelNames` is unspecified, the row names of `x$qvframe` will be used.

Value

`invisible(x)`

Author(s)

David Firth, <`d.firth@warwick.ac.uk`>

References

Easton, D. F, Peto, J. and Babiker, A. G. A. G. (1991) Floating absolute risk: an alternative to relative risk in survival and case-control analysis avoiding an arbitrary reference group. *Statistics in Medicine* **10**, 1025–1035.

Firth, D. (2000) Quasi-variances in Xlisp-Stat and on the web. *Journal of Statistical Software* **5.4**, 1–13. At `http://www.jstatsoft.org`

Firth, D. (2003) Overcoming the reference category problem in the presentation of statistical models. *Sociological Methodology* **33**, 1–18.

Firth, D. and Mezezes, R. X. de (2004) Quasi-variances. *Biometrika* **91**, 65–80.

McCullagh, P. and Nelder, J. A. (1989) *Generalized Linear Models*. London: Chapman and Hall.

Menezes, R. X. (1999) More useful standard errors for group and factor effects in generalized linear models. *D.Phil. Thesis*, Department of Statistics, University of Oxford.

See Also

`qvcalc`

Examples

```
##  Overdispersed Poisson loglinear model for ship damage data
##  from McCullagh and Nelder (1989), Sec 6.3.2
library(MASS)
data(ships)
ships$year <- as.factor(ships$year)
ships$period <- as.factor(ships$period)
shipmodel <- glm(formula = incidents ~ type + year + period,
    family = quasipoisson,
    data = ships, subset = (service > 0), offset = log(service))
qvs <- qvcalc(shipmodel, "type")
summary(qvs, digits = 4)
plot(qvs, col = c(rep("red", 4), "blue"))
## if we want to plot in decreasing order (of estimates):
est <- qvs$qvframe$estimate
qvs2 <- qvs
qvs2$qvframe <- qvs$qvframe[order(est, decreasing = TRUE), , drop = FALSE]
plot(qvs2)
```

---

qvcalc *Quasi Variances for Model Coefficients*

Description

Computes a set of quasi variances (and corresponding quasi standard errors) for estimated model coefficients relating to the levels of a categorical (i.e., factor) explanatory variable. For details of the method see Firth (2000), Firth (2003) or Firth and de Menezes (2004). Quasi variances generalize and improve the accuracy of “floating absolute risk” (Easton et al., 1991). This device for economical model summary was first suggested by Ridout (1989).

Usage

```
qvcalc(object, ...)

## Default S3 method:
qvcalc(object, factorname = NULL,  coef.indices = NULL,
      labels = NULL, dispersion = NULL,
      estimates = NULL, modelcall = NULL, ...)

## S3 method for class 'lm'
qvcalc(object, factorname = NULL, coef.indices = NULL,
       dispersion = NULL, ...)

## S3 method for class 'coxph'
qvcalc(object, factorname = NULL, coef.indices = NULL, ...)

## S3 method for class 'survreg'
qvcalc(object, factorname = NULL, coef.indices = NULL,
```

```
...)

## S3 method for class 'itempar'
qvcalc(object, ...)
```

Arguments

`object`	For `qvcalc.default`, this is the covariance (sub)matrix for the parameters of interest (including any that may have been constrained to zero). For the generic `qvcalc`, the `object` can be any object for which the relevant S3 method has been defined. These currently include many types of regression model (via `qvcalc.lm`), including objects of classes `coxph` and `survreg`; and also objects of class `itempar`.
`factorname`	Either `NULL`, or a character vector of length 1
`coef.indices`	Either `NULL`, or a numeric vector of length at least 3
`labels`	An optional vector of row names for the `qvframe` component of the result (redundant if `object` is a model)
`dispersion`	an optional scalar multiplier for the covariance matrix, to cope with overdispersion for example
`estimates`	an optional vector of estimated coefficients (redundant if `object` is a model, for example)
`modelcall`	optional, the call expression for the model of interest (redundant if `object` is a model with its own `call` component)
`...`	other arguments to pass to `qv.default`

Details

The `qvcalc.default` method is the computational backend for all other, class-specific methods.

In `qvcalc.default`, none of the arguments other than `object` is used in computing the result. The remaining arguments are simply passed through to the result object as components to help with record-keeping etc.

In `qvcalc.lm`, at least one of `factorname` or `coef.indices` must be non-`NULL`. The value of `coef.indices`, if non-`NULL`, determines which rows and columns of the model's variance-covariance matrix to use. If `coef.indices` contains a zero, then an extra row and column are included at the indicated position, to represent the zero variances and covariances associated with a reference level. If `coef.indices` is `NULL`, then `factorname` should be the name of a factor effect in the model, and is used in order to extract the necessary variance-covariance estimates.

For `qvcalc.itempar`, the `"itempar"` object must have the full variance-covariance matrix in its `"vcov"` attribute, and must have its `"alias"` attribute be `TRUE`. These attributes result from use of the default arguments `vcov = TRUE,alias = TRUE` when the `itempar` function is called.

Ordinarily the quasi variances are positive and so their square roots (the quasi standard errors) exist and can be used in plots, etc.

Occasionally one (and only one) of the quasi variances is negative, and so the corresponding quasi standard error does not exist (it appears as `NaN`). This is fairly rare in applications, and when it occurs it is because the factor of interest is strongly correlated with one or more other predictors in the model. It is not an indication that quasi variances are inaccurate. An example is shown below using data from the `car` package: the quasi variance approximation is exact (since `type` has only 3 levels), and there is a negative quasi variance. The quasi variances remain perfectly valid (they can be used to obtain inference on any contrast), but it makes no sense to plot 'comparison intervals' in the usual way since one of the quasi standard errors is not a real number.

Value

A list of class `qv`, with components

`covmat`	the full variance-covariance matrix for the estimated coefficients corresponding to the factor of interest
`qvframe`	a data frame with variables `estimate`, `SE`, `quasiSE` and `quasiVar`, the last two being a quasi standard error and quasi-variance for each level of the factor of interest
`relerrs`	relative errors for approximating the standard errors of all simple contrasts
`factorname`	the factor name if given
`coef.indices`	the coefficient indices if given
`modelcall`	if `object` is a model, `object$call`; otherwise `NULL`

Author(s)

David Firth, <d.firth@warwick.ac.uk>

References

Easton, D. F, Peto, J. and Babiker, A. G. A. G. (1991) Floating absolute risk: an alternative to relative risk in survival and case-control analysis avoiding an arbitrary reference group. *Statistics in Medicine* **10**, 1025–1035.

Firth, D. (2000) Quasi-variances in Xlisp-Stat and on the web. *Journal of Statistical Software* **5.4**, 1–13. At `http://www.jstatsoft.org`

Firth, D. (2003) Overcoming the reference category problem in the presentation of statistical models. *Sociological Methodology* **33**, 1–18.

Firth, D. and de Mezezes, R. X. (2004) Quasi-variances. *Biometrika* **91**, 65–80.

McCullagh, P. and Nelder, J. A. (1989) *Generalized Linear Models*. London: Chapman and Hall.

Menezes, R. X. de (1999) More useful standard errors for group and factor effects in generalized linear models. *D.Phil. Thesis*, Department of Statistics, University of Oxford.

Ridout, M.S. (1989). Summarizing the results of fitting generalized linear models to data from designed experiments. In: *Statistical Modelling: Proceedings of GLIM89 and the 4th International Workshop on Statistical Modelling held in Trento, Italy, July 17–21, 1989* (A. Decarli et al., eds.), pp 262–269. New York: Springer.

See Also

`worstErrors`, `plot.qv`.

Examples

```
##  Overdispersed Poisson loglinear model for ship damage data
##  from McCullagh and Nelder (1989), Sec 6.3.2
if (require(MASS)) {
    data(ships)
    ships$year <- as.factor(ships$year)
    ships$period <- as.factor(ships$period)
    shipmodel <- glm(formula = incidents ~ type + year + period,
                     family = quasipoisson,
                     data = ships,
                     subset = (service > 0),
                     offset = log(service))
    shiptype.qv <- qvcalc(shipmodel, "type")

    ## We can plot "comparison intervals" as follows:
    ##   plot(shiptype.qv, xlab = "ship type")

    ## An equivalent result by using the coef.indices argument instead:
    ##   shiptype.qv2 <- qvcalc(shipmodel, coef.indices = c(0, 2:5))

    summary(shiptype.qv, digits = 4)
}

## Example of a "coxph" model
if(require(survival)) {
    data("veteran", package = "survival")
    cancer_model <- coxph(Surv(time,status) ~ celltype, data = veteran)
    celltype_qv <- qvcalc(cancer_model, "celltype")
    summary(celltype_qv)
}

## Example of a "survreg" model
if(require(survival)) {
    data("veteran", package = "survival")
    cancer_model2 <- survreg(Surv(time,status) ~ celltype, data = veteran,
                             dist = "weibull")
    celltype_qv2 <- qvcalc(cancer_model2, "celltype")
    summary(celltype_qv2)
}

## Based on an example from ?itempar
if(require(psychotools)) {
    data("VerbalAggression", package = "psychotools")
    raschmod <- raschmodel(VerbalAggression$resp2)
    ip1 <- itempar(raschmod)
    qv1 <- qvcalc(ip1)
    summary(qv1) }

##  Example of a negative quasi variance
##  Requires the "car" package
## Not run:
    library(car)
```

```
    data(Prestige)
    attach(Prestige)
    mymodel <- lm(prestige ~ type + education)
    library(qvcalc)
    type.qvs <- qvcalc(mymodel, "type")
    ##  Warning message:
    ##  In sqrt(qv) : NaNs produced
    summary(type.qvs)
    ##  Model call:  lm(formula = prestige ~ type + education)
    ##  Factor name:  type
    ##          estimate       SE  quasiSE  quasiVar
    ##    bc    0.000000 0.000000 2.874361  8.261952
    ##    prof  6.142444 4.258961 3.142737  9.876793
    ##    wc   -5.458495 2.690667      NaN -1.022262
    ##  Worst relative errors in SEs of simple contrasts (%):  0 0
    ##  Worst relative errors over *all* contrasts (%):  0 0
    plot(type.qvs)
    ##  Error in plot.qv(type.qvs) :  No comparison intervals available,
    ##  since one of the quasi variances is negative.  See ?qvcalc for more.

## End(Not run)
```

`worstErrors` *Accuracy of a Quasi-variance Approximation*

Description

Computes the worst relative error, among all contrasts, for the standard error as derived from a set of quasi variances. For details of the method see Menezes (1999) or Firth and Menezes (2004).

Usage

```
worstErrors(qv.object)
```

Arguments

`qv.object`	An object of class qv

Value

A numeric vector of length 2, the worst negative relative error and the worst positive relative error.

Author(s)

David Firth, <d.firth@warwick.ac.uk>

References

Firth, D. and Mezezes, R. X. de (2004) Quasi-variances. *Biometrika* **91**, 69–80.

McCullagh, P. and Nelder, J. A. (1989) *Generalized Linear Models*. London: Chapman and Hall.

Menezes, R. X. (1999) More useful standard errors for group and factor effects in generalized linear models. *D.Phil. Thesis*, Department of Statistics, University of Oxford.

See Also

qvcalc

Examples

```
##  Overdispersed Poisson loglinear model for ship damage data
##  from McCullagh and Nelder (1989), Sec 6.3.2
library(MASS)
data(ships)
ships$year <- as.factor(ships$year)
ships$period <- as.factor(ships$period)
shipmodel <- glm(formula = incidents ~ type + year + period,
    family = quasipoisson,
    data = ships, subset = (service > 0), offset = log(service))
shiptype.qvs <- qvcalc(shipmodel, "type")
summary(shiptype.qvs, digits = 4)
worstErrors(shiptype.qvs)
```

Index

∗Topic **IO**
    indentPrint, 2
∗Topic **hplot**
    plot.qv, 2
∗Topic **models**
    plot.qv, 2
    qvcalc, 4
    worstErrors, 8
∗Topic **regression**
    qvcalc, 4
    worstErrors, 8

coxph, *5*

indentPrint, 2
itempar, *5*

plot.default, *3*
plot.qv, 2, *6*
print, *2*
print.qv *(*qvcalc*)*, 4

qvcalc, *3*, 4, *9*

summary.qv *(*qvcalc*)*, 4
survreg, *5*

worstErrors, *6*, 8