**indentPrint**  \hspace{1cm} *Print with Line Indentation*

**Description**

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

**Usage**

```r
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**

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

---

**plot.qv** \hspace{1cm} *Plot method for objects of class qv*

**Description**

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

**Usage**

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

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


See Also

`qvcalc`
## Overdispersed Poisson loglinear model for ship damage data

from McCullagh and Nelder (1989), Sec 6.3.2

```r
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**

```r
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,
```

---
qvcalc

...)

## 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


See Also

worstErrors.plot.qv.
Examples

```r
## 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 = Admit ~ rank + dept + evaluator + year, 
                   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)
```

worstErrors

### 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

```r
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>
worstErrors

References


See Also

qvcalc

Examples

```r
## 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

* IO
  indentPrint, 2

* hplot
  plot.qv, 2

* models
  plot.qv, 2
  qvcalc, 4
  worstErrors, 8

* 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