Estimate and plot ROC Curves

Description

Estimate and Plot ROC curves. Bootstrap confidence intervals for ROC(f) at specified False positive rate f, or ROC^(-1)(t) at specified true positive rate t are optionally included. Parametric and Non-parametric methods are available. Optional covariate adjustment can be achieved. Algorithms use the percentile value formulation of the ROC curve.

Usage

```
roccurve(dataset = NULL, d, markers, rocmeth = "nonparametric",
   link = "probit", interval = c(0, 1, 10), ordinal=FALSE,
   c_fpr=NULL, c_tpr=NULL, nograph = FALSE, bw = FALSE,
   roc = NULL, rocinv = NULL, offset = 0.006,
   pvcmeth = "empirical", tiecorr = FALSE, adjcov = NULL,
   adjmodel = "stratified", nsamp = 1000, noccsamp = FALSE,
   nostsamp = FALSE, cluster = NULL, bsparam = TRUE, level = 95,
   genrocvars = FALSE, genpcv = FALSE, replace = FALSE,
   nobstrap = FALSE, titleOverride = NULL, dupStata = TRUE)
```

Arguments

`dataset`	optional character string specifying the name of the dataset to be used for analysis.
`d`	character string specifying the name of the 0/1 outcome vector.
`markers`	vector of character strings specifying the names of the test markers variables.
`rocmeth`	character string specifying the ROC calculation method as "nonparametric" (empirical ROC, the default) or "parametric".
`link`	character string specifying the ROC generalized linear models link function as "probit" (default) or "logit"; for use with rocmeth="parametric" only. "probit" corresponds to the binormal ROC model, that is, PHI^(-1){ROC(f)} = intercept + slope * PHI^(-1)(f), where PHI is the standard normal cumulative distribution function. "logit" corresponds to the bilogistic ROC model, that is, logit{ROC(f)} = intercept + slope * logit(f).
`interval`	numeric vector (a,b,np) specifying an interval (a,b) in (0,1), and the number of points, np, over which the parametric ROC model is to be fit. Valid only for `rocmeth`="parametric" option. Ignored if `ordinal`=TRUE. The default is (0,1,10).
`ordinal`	logical. If TRUE, test marker variable(s) are specified as ordinal-valued ratings, rather than continuous measures. This option affects the fitting algorithm for the parametric ROC estimator when `rocmeth`="parametric" is specified and also affects the covariate adjustment options for both ROC estimators. Must be TRUE if adjmodel is "ologit" or "oprobit". "linear" model adjustment is not permitted with `ordinal`=TRUE. The default is FALSE.
`c_fpr`	specify FPR=f at which to return the corresponding estimated marker threshold value(s). More details below.
`c_tpr`	specify TPF=t at which to return the corresponding estimated marker threshold value(s). The threshold is determined indirectly for TPR=t: The corresponding false positive rate, f = ROC^(-1)(t), is first determined for the specified t, then the corresponding threshold(s) are

	determined as for `c_fpr(f)`.
`nograph`	logical. If TRUE, the ROC plot is suppressed and only numerical results are returned; default is FALSE.
`bw`	logical. If TRUE, plot black line types rather than solid colour lines to distinguish ROC curves; default is FALSE.
`roc`	specify FPR, f, at which to include bootstrap percentile-based confidence intervals (CIs) for ROC(f). The argument must be between 0 and 1. Only one of roc=f or rocinv=t can be specified.
`rocinv`	specify TPR, t, at which to include bootstrap percentile-based confidence intervals (CIs) for ROC^(-1)(t). The argument must be between 0 and 1. Only one of roc=f or rocinv=t can be specified.
`offset`	specify the x- or y-axis offset from f (or t) for the placement of 2nd and subsequent marker CI's, to avoid superimposed interval bars. The argument must be between 0 and 0.2; default is offset=0.006.
`pvcmeth`	character string specifying PV calculation method as "empirical" (default) or "normal". "empirical" uses the empirical distribution of the test measure among controls (D=0) as the reference distribution for the calculation of case PVs. The PV for the case measure y_i is the proportion of control measures that are smaller than y_i. "normal" models the test measure among controls with a normal distribution. The PV for the case measure y_i is the standard normal cumulative distribution function of (y_i - mean)/sd, where the mean and the standard deviation (sd) are calculated by using the control sample.
`tiecorr`	logical. If FALSE (default), no correction for ties. If TRUE, it indicates that a correction for ties between case and control values is included in the empirical PV calculation. The correction is important only in calculating summary indices, such as the area under the ROC curve. The tie-corrected PV for a case with the marker value y_i is the proportion of control values Y_Db < y_i plus one half the proportion of control values Y_Db = y_i, where Y_Db denotes controls. By default, the PV calculation includes only the first term, i.e. the proportion of control values Y_Db < y_i. This option applies only to the empirical PV calculation method.
`adjcov`	character string vector specifying covariates to adjust for.
`adjmodel`	character string specifying how the covariate adjustment is to be done: "stratified" (default), "linear", "oprobit" (ordered probit), or "ologit" (ordered logit). If "stratified", PVs are calculated separately for each stratum defined by `adjcov`. This is the default if `adjmodel` is not specified and `adjcov` is. Each case-containing stratum must include at least two controls. Strata that do not include cases are excluded from calculations. "linear" fits a linear regression of the marker distribution on the adjustment covariates among controls. Standardized residuals based on this fitted linear model are used in place of the marker values for cases and controls. "oprobit" calculates PVs based on the fit of an ordered probit regression model of the marker on the adjustment covariates among controls. "ologit" calculates PVs based on the fit of an ordered logit regression model of the marker on the adjustment covariates among controls. "oprobit" and "ologit" assume that `markers` consists of ordinal-valued marker variables.
`nsamp`	number of bootstrap samples to be drawn for estimating sampling variability of estimates; default is nsamp=1000.
`nobstrap`	logical. If TRUE, omit boostrap sampling and estmation of standard errors and CIs. If `nsamp` is specified, `nobstrap` will override it. Default is FALSE.

noccsamp	logical. If TRUE, bootstrap samples are drawn from the combined sample (cohort sampling) rather than sampling separately from cases and controls (case-control sampling); default is FALSE (case-control sampling).
nostsamp	logical. If TRUE (default), bootstrap samples are drawn without respect to covariate strata. By default, samples are drawn from within covariate strata when stratified covariate adjustment is requested via the adjcov and adjmodel options.
cluster	character string specifying variables that identify bootstrap resampling clusters.
bsparam	logical. If TRUE (default), obtain bootstrap se's and CI's for binormal ROC intercept and slope parameters.
level	specify confidence level for CIs as a percentage; default is level=95.
genrocvars	logical. If TRUE, generate new variables, tpf\# and fpf\# to hold (TPF, FPF) coordinates for each marker \#. Points resulting from the empirical rocmeth are to be plotted as a right-continuous step function. New variable names are numbered according to the marker variable order in markers. Default is FALSE.
genpcv	logical. If TRUE, generate new variables, pcv\# to hold percentile values for each marker in markers. New variable numbers correspond to the marker variable order in markers. Default is FALSE.
replace	logical. If TRUE, overwrite existing tpf\#, fpf\#, or pcv\# variables by genrocvars or genpcv; default is FALSE.
titleOverride	If non-null, a string which will be used as the main title on the ROC plot; default is NULL.
dupStata	logical. If TRUE, setup plot to look like the Stata program's output. If FALSE, do a "standard" R plot, allowing for typical plot layout in R to be controlled outside the function; default is TRUE.

Details

roccurve estimates and plots ROC curves for one or more continuous disease marker or diagnostic test variables used to classify a 0/1 outcome indicator variable. Bootstrap confidence intervals for either ROC(f) at specified f or the inverse, ROC^(-1)(t), at specified t, are optionally included.

ROC calculations are based on percentile values (PVs) of the case measures relative to the corresponding marker distribution among controls (!!!!!include references - Pepe and Longton, Huang and Pepe).

The empirical ROC is calculated as the empirical cumulative distribution function of the case PV complements (1 - PV):

```
ROC(f) = P( 1-PV_D <= f ) = P( PV_D >= 1-f )
```

A parameteric distribution-free estimator of either the classic binormal ROC,

```
PHI^(-1)[ROC(f)] = a + b*PHI^(-1)(f),
```

or the bilogistic ROC,

```
logit[ROC(f)] = a + b*logit(f)
```

can be optionally fit within a generalized linear models binary regression framework by specifying rocmeth="parametric" and either link="probit" or link="logit",

respectively (!!!!!include references - Pepe, Section 5.5.2; Alonzo and Pepe).

Optional covariate adjustment can be achieved either by stratification or with a linear regression approach (Janes and Pepe (2008); Janes and Pepe (2009)). Ordered regression covariate adjustment options are available if the test measures are ordinal (Morris, Pepe, Barlow (in press)).

The marker threshold value(s) for a specified false positive rate, FPR=f can be returned, i.e. c such that P[Y_db >= c] <= f. Cannot be specified if the marker is `ordinal` and is less meaningful for markers with a few distinct values. If `adjmodel` is "stratified" or "linear", a matrix of thresholds for all combinations of adjustment covariate values is returned. In the absence of covariate adjustment and with empirical PV calculation, the threshold is calculated as the (1-f)th quantile of the empirical marker distribution among controls. With normal PV calculation, the (1-f)th quantile of the normal distribution defined by the control sample mean and variance is used. Similarly, with stratified covariate adjustment the within-stratum empirical or normal control distributions are used and separate thresholds calculated for each stratum. With linear covariate adjustment, thresholds are based on the empirical or normal distributions of the standardized residuals from a fitted linear model among controls.

A companion program for the Stata software package is available. A detailed description of the methods and algorithms are provide in two articles in the Stata Journal which can be obtained upon request from Gary Longton (glongton@fhcrc.org). Corresponding articles for this program are forthcoming.

Value

`c`	c = c_fpr(f) for marker number \# in the absense of covariate adjustment.
`ROC_ci`	n x 3 matrix of `roc(f)` or `rocinv(t)` estimates and confidence limits returned when either option is specified. Columns correspond to the point estimate and the lower and upper confidence bounds. Rows correspond to the marker variables included in `markers`.
`BNParm`	n x 2 matrix of binormal or bilogistic curve intercept and slope parameter estimates when `rocmeth`="parametric" is specified. Columns correspond to alpha_0 and alpha_1 parameters, and rows correspond to markers.
`BNP_se`	n x 2 matrix of bootstrap standard error estimates for binormal or bilogistic curve parameters when `rocmeth`="parametric" is specified along with the `bsparam` option. Columns correspond to alpha_0 and alpha_1 standard errors and rows correspond to markers.
`BNP_ci`	n x 4 matrix of bootstrap percentile-based confidence limits for the binormal or bilogistic curve parameters when `rocmeth`="parametric" is specified along with the `bsparam` option. Columns correspond to alpha_0 lower and upper bound limits and alpha_1 lower and upper bound limits. Rows correspond to markers.
`C`	n x k matrix of covariate-adjusted marker thresholds corresponding to FPR = f specified with c_fpr(f) for marker \#. First column holds threshold values. k-1 covariates specified with `adjcov` are in the remaining columns. Rows correspond to n distinct combinations of covariate values.

Author(s)

Aasthaa Bansal, University of Washington, Seattle, WA. abansal@u.washington.edu

Daryl Morris, University of Washington, Seattle, WA. darylm@u.washington.edu

Gary Longton, Fred Hutchinson Cancer Research Center, Seattle, WA. glongton@fhcrc.org

Margaret Pepe, Fred Hutchinson Cancer Research Center and University of Washington, Seattle, WA. mspepe@u.washington.edu

Holly Janes, Fred Hutchinson Cancer Research Center and University of Washington, Seattle, WA. hjanes@fhcrc.org

References

Dodd, L., Pepe, M.S. 2003. Partial AUC estimation and regression. *Biometrics* **59**,614–623.

Huang, Y., Pepe, M.S. 2009. Biomarker evaluation using the controls as a reference population. *Biostatistics 2*,228–44.

Janes, H., Pepe, M.S. 2008. Adjusting for covariates in studies of diagnostic, screening, or prognostic markers: an old concept in a new setting. *American Journal of Epidemiology* **168**,89–97.

Janes, H., Pepe, M.S. 2009. Adjusting for covariate effects on classification accuracy using the covariate-adjusted ROC curve. *Biometrika* **96**,383–398.

Janes, H., Longton G, Pepe, M.S. 2009. Accommodating covariates in receiver operating characteristic analysis. *Stata Journal* **9(1)**,17–39.

Morris, D.E., Pepe, M.S., Barlow, W.E. Contrasting Two Frameworks for ROC Analysis of Ordinal Ratings. *Medical Decision Making* (in press)

Pepe, M.S., Longton, G. 2005. Standardizing markers to evaluate and compare their performances. *Epidemiology* **16(5)**,598-603.

Pepe MS, Longton G, Janes H. 2009. Estimation and comparison of receiver operating characteristic curves. *Stata Journal* **9(1)**,1–16.

Pepe, M.S. 2003. *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press.

See Also

comproc, rocreg

Examples

```
nnhs2 <- read.csv("http://labs.fhcrc.org/pepe/book/data/nnhs2.csv",
                  header = TRUE, sep = ",")
## Three ways of producing the same plot
roccurve(dataset="nnhs2", d="d", markers="y1")          # Vectors part of a data frame
roccurve(d="nnhs2$d", markers="nnhs2$y1")

disease <- nnhs2$d
marker1 <- nnhs2$y1
roccurve(d="disease", markers="marker1")                # Independent vectors, not in a data frame

## Multiple markers
roccurve(d="nnhs2$d", markers=c("nnhs2$y1", "nnhs2$y2"))

## Sampling Variability
#roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"), roc=0.10, nsamp=5000)
#roccurve(dataset="nnhs2", d="d", markers=c("y1","y2","y3"), roc=0.15, level=90)
# Get ROC(0.10), using cohort sampling and 5000 bootstrap samples
roccurve(dataset="nnhs2", d="d", markers="y1", roc=0.10, noccsamp=TRUE, nsamp=5000)
# Get ROC(0.15), generating a 90
```

```
roccurve(d="nnhs2$d", markers=c("nnhs2$y1", "nnhs2$y2"), roc=0.15, level=90)
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2","y3"), roc=0.15, level=90,
   cluster="y1")

## Percentile value calculation method
# Using tie correction
roccurve(d="nnhs2$d", markers=c("nnhs2$y1", "nnhs2$y2"), tiecorr=TRUE)
# Assuming normal distribution
roccurve(d="nnhs2$d", markers=c("nnhs2$y1", "nnhs2$y2"), pvcmeth="normal")

## Parametric ROC curves
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"), roc=0.2,
         rocmeth="parametric")
roccurve(dataset="nnhs2", d="d", markers="y1", roc=0.2, rocmeth="parametric",
         link="logit")
roccurve(dataset="nnhs2", d="d", markers="y1", roc=0.05, rocmeth="parametric",
         interval=c(0, 0.1, 10))

## Get ROC Inverse, ROC^-1(0.8)
roccurve(dataset="nnhs2", d="d", markers="y1", rocinv=0.8)

## New variable options
# Generate pcv variable containing percentile values for marker y1
roccurve(dataset="nnhs2", d="d", markers="y1", roc=0.2, genpcv=TRUE)
# Try to store percentile values when pcv variable already exists
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"), roc=0.2, genpcv=TRUE)
# Try to store percentile values when pcv variable already exists,
#   specifying we want to replace old values
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"), roc=0.2,
         genpcv=TRUE, replace=TRUE)
#Graph options - don't generate a plot
roccurve(dataset="nnhs2", d="d", markers=c("y1"), roc=0.2, nograph=TRUE)
## With Covariate Adjustment
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"),
         adjcov=c("currage","gender"))
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"),
         adjcov=c("currage","gender"), adjmodel="linear")
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"), adjcov="currage",
         adjmodel="linear", pvcmeth="normal", roc=0.20)
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"), adjcov="currage",
         rocmeth="parametric")
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"), adjcov="currage",
         rocmeth="parametric", interval=c(0,0.2,5))
roccurve(dataset="nnhs2", d="d", markers=c("y1","y2"), adjcov="currage",
         genrocvars=TRUE, genpcv=TRUE)
```
