

rocreg — Receiver operating characteristic (ROC) regression

[Description](#)[Quick start](#)[Menu](#)[Syntax](#)[Options](#)[Remarks and examples](#)[Stored results](#)[Methods and formulas](#)[Acknowledgments](#)[References](#)[Also see](#)

Description

The `rocreg` command is used to perform receiver operating characteristic (ROC) analyses with rating and discrete classification data under the presence of covariates.

The two variables `refvar` and `classvar` must be numeric. The reference variable indicates the true state of the observation—such as diseased and nondiseased or normal and abnormal—and must be coded as 0 and 1. The `refvar` coded as 0 can also be called the control population, while the `refvar` coded as 1 comprises the case population. The rating or outcome of the diagnostic test or test modality is recorded in `classvar`, which must be ordinal, with higher values indicating higher risk.

`rocreg` can fit three models: a nonparametric model, a parametric probit model that uses the bootstrap for inference, and a parametric probit model fit using maximum likelihood.

Quick start

Nonparametric estimation with bootstrap resampling

Area under the ROC curve for test classifier `v1` and true state `true` using seed 20547

```
rocreg true v1, bseed(20547)
```

Add `v2` as an additional classifier

```
rocreg true v1 v2, bseed(20547)
```

Same as above, but estimate ROC value for a false-positive rate of 0.7

```
rocreg true v1 v2, bseed(20547) roc(.7)
```

Covariate stratification of controls by categorical variable `a` using seed 121819

```
rocreg true v1 v2, bseed(121819) ctrlcov(a)
```

Linear control covariate adjustment with binary variable `b` and continuous variable `x`

```
rocreg true v1 v2, bseed(121819) ctrlcov(b x) ctrlmodel(linear)
```

Parametric estimation

Area under the ROC curve for test classifier `v1` and true state `true` by estimating equations using seed 200512

```
rocreg true v1, probit bseed(200512)
```

And save results to `myfile.dta` for use by `rocregplot`

```
rocreg true v1, probit bseed(200512) bsave(myfile)
```

Add `v2` as a classifier and `x` as a control covariate in a linear control covariate-adjustment model

```
rocreg true v1 v2, probit bseed(200512) ctrlcov(x) ctrlmodel(linear)
```

Also treat `x` as a ROC covariate

```
rocreg true v1 v2, probit bseed(200512) ctrlcov(x) ///  
      ctrlmodel(linear) roccov(x)
```

Estimate AUC by maximum likelihood instead of bootstrap resampling

```
rocreg true v1, probit ml
```

Menu

Statistics > Epidemiology and related > ROC analysis > ROC regression models

Syntax

Perform nonparametric analysis of ROC curve under covariates, using bootstrap

```
rocreg refvar classvar [classvars] [if] [in] [, np_options]
```

Perform parametric analysis of ROC curve under covariates, using bootstrap

```
rocreg refvar classvar [classvars] [if] [in], probit [probit_options]
```

Perform parametric analysis of ROC curve under covariates, using maximum likelihood

```
rocreg refvar classvar [classvars] [if] [in] [weight], probit ml
[probit_ml_options]
```

<i>np_options</i>	Description
Model	
<code>auc</code>	estimate total area under the ROC curve; the default
<code>roc(numlist)</code>	estimate ROC for given false-positive rates
<code>invroc(numlist)</code>	estimate false-positive rates for given ROC values
<code>pauc(numlist)</code>	estimate partial area under the ROC curve (pAUC) up to each false-positive rate
<code>cluster(varname)</code>	variable identifying resampling clusters
<code>ctrlcov(varlist)</code>	adjust control distribution for covariates in <i>varlist</i>
<code>ctrlmodel(strata linear)</code>	stratify or regress on covariates; default is <code>ctrlmodel(strata)</code>
<code>pvc(empirical normal)</code>	use empirical or normal distribution percentile value estimates; default is <code>pvc(empirical)</code>
<code>tiecorrected</code>	adjust for tied observations; not allowed with <code>pvc(normal)</code>
Bootstrap	
<code>nobootstrap</code>	do not perform bootstrap, just output point estimates
<code>bseed(#)</code>	random-number seed for bootstrap
<code>breps(#)</code>	number of bootstrap replications; default is <code>breps(1000)</code>
<code>bootcc</code>	perform case-control (stratified on <i>refvar</i>) sampling rather than cohort sampling in bootstrap
<code>nobstrata</code>	ignore covariate stratification in bootstrap sampling
<code>nodots</code>	suppress bootstrap replication dots
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>

<i>probit_options</i>	Description
Model	
* probit	fit the probit model
roccov (<i>varlist</i>)	covariates affecting ROC curve
fprpts (#)	number of false-positive rate points to use in fitting ROC curve; default is fprpts (10)
ctrlfprall	fit ROC curve at each false-positive rate in control population
cluster (<i>varname</i>)	variable identifying resampling clusters
ctrlcov (<i>varlist</i>)	adjust control distribution for covariates in <i>varlist</i>
ctrlmodel (<i>strata</i> <i>linear</i>)	stratify or regress on covariates; default is ctrlmodel (<i>strata</i>)
pvc (<i>empirical</i> <i>normal</i>)	use empirical or normal distribution percentile value estimates; default is pvc (<i>empirical</i>)
tiecorrected	adjust for tied observations; not allowed with pvc (<i>normal</i>)
Bootstrap	
nobootstrap	do not perform bootstrap, just output point estimates
bseed (#)	random-number seed for bootstrap
breps (#)	number of bootstrap replications; default is breps (1000)
bootcc	perform case–control (stratified on <i>refvar</i>) sampling rather than cohort sampling in bootstrap
nobstrata	ignore covariate stratification in bootstrap sampling
nodots	suppress bootstrap replication dots
bsave (<i>filename</i> , ...)	save bootstrap replicates from parametric estimation
bfile (<i>filename</i>)	use bootstrap replicates dataset for estimation replay
Reporting	
level (#)	set confidence level; default is level (95)

***probit** is required.

<i>probit_ml_options</i>	Description
Model	
* probit	fit the probit model
* ml	fit the probit model by maximum likelihood estimation
roccov (<i>varlist</i>)	covariates affecting ROC curve
cluster (<i>varname</i>)	variable identifying clusters
ctrlcov (<i>varlist</i>)	adjust control distribution for covariates in <i>varlist</i>
Reporting	
level (#)	set confidence level; default is level (95)
display_options	control column formats, line width, and display of omitted variables
Maximization	
maximize_options	control the maximization process; seldom used

***probit** and **ml** are required.

fweights, **iwweights**, and **pweights** are allowed with maximum likelihood estimation; see [U] 11.1.6 **weight**.

collect is allowed; see [U] 11.1.10 **Prefix commands**.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Options are presented under the following headings:

Options for nonparametric ROC estimation, using bootstrap

Options for parametric ROC estimation, using bootstrap

Options for parametric ROC estimation, using maximum likelihood

Options for nonparametric ROC estimation, using bootstrap

Model

auc estimates the total area under the ROC curve. This is the default summary statistic.

roc(*numlist*) estimates the ROC corresponding to each of the false-positive rates in *numlist*. The values of *numlist* must be in the range (0,1).

invroc(*numlist*) estimates the false-positive rates corresponding to each of the ROC values in *numlist*. The values of *numlist* must be in the range (0,1).

pauc(*numlist*) estimates the partial area under the ROC curve up to each false-positive rate in *numlist*. The values of *numlist* must in the range (0,1].

cluster(*varname*) specifies the variable identifying resampling clusters.

ctrlcov(*varlist*) specifies the covariates to be used to adjust the control population.

ctrlmodel(*strata*|*linear*) specifies how to model the control population of classifiers on **ctrlcov**(*varlist*). When **ctrlmodel**(*linear*) is specified, linear regression is used. The default is **ctrlmodel**(*strata*); that is, the control population of classifiers is stratified on the control variables.

`pvc(empirical|normal)` determines how the percentile values of the control population will be calculated. When `pvc(normal)` is specified, the standard normal cumulative distribution function (CDF) is used for calculation. Specifying `pvc(empirical)` will use the empirical CDFs of the control population classifiers for calculation. The default is `pvc(empirical)`.

`tierecorrected` adjusts the percentile values for ties. For each value of the classifier, one half the probability that the classifier equals that value under the control population is added to the percentile value. `tierecorrected` is not allowed with `pvc(normal)`.

Bootstrap

`nobootstrap` specifies that bootstrap standard errors not be calculated.

`bseed(#)` specifies the random-number seed to be used in the bootstrap.

`breps(#)` sets the number of bootstrap replications. The default is `breps(1000)`.

`bootcc` performs case–control (stratified on *refvar*) sampling rather than cohort bootstrap sampling.

`nobstrata` ignores covariate stratification in bootstrap sampling.

`nodots` suppresses bootstrap replication dots.

Reporting

`level(#)`; see [\[R\] Estimation options](#).

Options for parametric ROC estimation, using bootstrap

Model

`probit` fits the probit model. This option is required and implies parametric estimation.

`roccov(varlist)` specifies the covariates that will affect the ROC curve.

`fprpts(#)` sets the number of false-positive rate points to use in modeling the ROC curve. These points form an equispaced grid on (0,1). The default is `fprpts(10)`.

`ctrlfprall` models the ROC curve at each false-positive rate in the control population.

`cluster(varname)` specifies the variable identifying resampling clusters.

`ctrlcov(varlist)` specifies the covariates to be used to adjust the control population.

`ctrlmodel(strata|linear)` specifies how to model the control population of classifiers on `ctrlcov()`. When `ctrlmodel(linear)` is specified, linear regression is used. The default is `ctrlmodel(strata)`; that is, the control population of classifiers is stratified on the control variables.

`pvc(empirical|normal)` determines how the percentile values of the control population will be calculated. When `pvc(normal)` is specified, the standard normal CDF is used for calculation. Specifying `pvc(empirical)` will use the empirical CDFs of the control population classifiers for calculation. The default is `pvc(empirical)`.

`tierecorrected` adjusts the percentile values for ties. For each value of the classifier, one half the probability that the classifier equals that value under the control population is added to the percentile value. `tierecorrected` is not allowed with `pvc(normal)`.

Bootstrap

`nobootstrap` specifies that bootstrap standard errors not be calculated.

`bseed(#)` specifies the random-number seed to be used in the bootstrap.

`breps(#)` sets the number of bootstrap replications. The default is `breps(1000)`.

`bootcc` performs case–control (stratified on *refvar*) sampling rather than cohort bootstrap sampling.

`nobstrata` ignores covariate stratification in bootstrap sampling.

`nodots` suppresses bootstrap replication dots.

`bsave(filename, ...)` saves bootstrap replicates from parametric estimation in the given *filename* with specified options (that is, `replace`). `bsave()` is only allowed with parametric analysis using bootstrap.

`bfile(filename)` specifies to use the bootstrap replicates dataset for estimation replay. `bfile()` is only allowed with parametric analysis using bootstrap.

Reporting

`level(#)`; see [R] [Estimation options](#).

Options for parametric ROC estimation, using maximum likelihood

Model

`probit` fits the probit model. This option is required and implies parametric estimation.

`m1` fits the probit model by maximum likelihood estimation. This option is required and must be specified with `probit`.

`roccov(varlist)` specifies the covariates that will affect the ROC curve.

`cluster(varname)` specifies the variable used for clustering.

`ctrlcov(varlist)` specifies the covariates to be used to adjust the control population.

Reporting

`level(#)`; see [R] [Estimation options](#).

display_options: `noomit`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no1stretch`; see [R] [Estimation options](#).

Maximization

maximize_options: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used. The `technique(bhhh)` option is not allowed.

Remarks and examples

[stata.com](http://www.stata.com)

Remarks are presented under the following headings:

[Introduction](#)

[ROC statistics](#)

[Covariate-adjusted ROC curves](#)

[Parametric ROC curves: Estimating equations](#)

[Parametric ROC curves: Maximum likelihood](#)

Introduction

Receiver operating characteristic (ROC) analysis provides a quantitative measure of the accuracy of diagnostic tests to discriminate between two states or conditions. These conditions may be referred to as normal and abnormal, nondiseased and diseased, or control and case. We will use these terms interchangeably. The discriminatory accuracy of a diagnostic test is measured by its ability to correctly classify normal control and case subjects.

The analysis uses the ROC curve, a graph of the sensitivity versus $1 - \text{specificity}$ of the diagnostic test. The sensitivity is the fraction of positive cases that are correctly classified by the diagnostic test, whereas the specificity is the fraction of negative cases that are correctly classified. Thus, the sensitivity is the true-positive rate, and the specificity is the true-negative rate. We also call $1 - \text{specificity}$ the false-positive rate.

These rates are functions of the possible outcomes of the diagnostic test. At each outcome, a decision will be made by the user of the diagnostic test to classify the tested subject as either normal or abnormal. The true-positive and false-positive rates measure the probability of correct classification or incorrect classification of the subject as abnormal. Given the classification role of the diagnostic test, we will refer to it as the classifier.

Using this basic definition of the ROC curve, [Pepe \(2000\)](#) and [Pepe \(2003\)](#) describe how ROC analysis can be performed as a two-stage process. In the first stage, the control distribution of the classifier is estimated. The specificity is then determined as the percentiles of the classifier values calculated based on the control population. The false-positive rates are calculated as $1 - \text{specificity}$. In the second stage, the ROC curve is estimated as the cumulative distribution of the case population's "false-positive" rates, also known as the survival function under the case population of the previously calculated percentiles. We use the terms ROC value and true-positive value interchangeably.

This formulation of ROC curve analysis provides simple, nonparametric estimates of several ROC curve summary parameters: area under the ROC curve, partial area under the ROC curve, ROC value for a given false-positive rate, and false-positive rate (also known as invROC) for a given ROC value. In the next section, we will show how to use `rocreg` to compute these estimates with bootstrap inference. There we will also show how `rocreg` complements the other nonparametric Stata ROC commands `roctab` and `roccomp`.

Other factors beyond condition status and the diagnostic test may affect both stages of ROC analysis. For example, a test center may affect the control distribution of the diagnostic test. Disease severity may affect the distribution of the standardized diagnostic test under the case population. Our analysis of the ROC curve in these situations will be more accurate if we take these covariates into account.

In a nonparametric ROC analysis, covariates may only affect the first stage of estimation; that is, they may be used to adjust the control distribution of the classifier. In a parametric ROC analysis, it is assumed that ROC follows a normal distribution, and thus covariates may enter the model at both stages; they may be used to adjust the control distribution and to model ROC as a function of these covariates and the false-positive rate. In parametric models, both sets of covariates need not be distinct but, in fact, they are often the same.

To model covariate effects on the first stage of ROC analysis, [Janes and Pepe \(2009\)](#) propose a covariate-adjusted ROC curve. We will demonstrate the covariate adjustment capabilities of `rocreg` in [Covariate-adjusted ROC curves](#).

To account for covariate effects at the second stage, we assume a parametric model. Particularly, the ROC curve is a generalized linear model of the covariates. We will thus have a separate ROC curve for each combination of the relevant covariates. In [Parametric ROC curves: Estimating equations](#), we show how to fit the model with estimating equations and bootstrap inference using `rocreg`.

This method, documented as the “pdf” approach in [Alonzo and Pepe \(2002\)](#), works well with weak assumptions about the control distribution.

Also in *Parametric ROC curves: Estimating equations*, we show how to fit a constant-only parametric model (involving no covariates) of the ROC curve with weak assumptions about the control distribution. The constant-only model capabilities of `rocreg` in this context will be compared with those of `rocfit`. `roccomp` has the `binormal` option, which will allow it to compute area under the ROC curve according to a normal ROC curve, equivalent to that obtained by `rocfit`. We will compare this functionality with that of `rocreg`.

In *Parametric ROC curves: Maximum likelihood*, we demonstrate maximum likelihood estimation of the ROC curve model with `rocreg`. There we assume a normal linear model for the classifier on the covariates and case–control status. This method is documented in [Pepe \(2003\)](#). We will also demonstrate how to use this method with no covariates, and we will compare `rocreg` under the constant-only model with `rocfit` and `roccomp`.

The `rocregplot` command is used repeatedly in this entry. This command provides graphical output for `rocreg` and is documented in [R] [rocregplot](#).

ROC statistics

`roctab` computes the ROC curve by calculating the false-positive rate and true-positive rate empirically at every value of the input classifier. It makes no distributional assumptions about the case or control distributions. We can get identical behavior from `rocreg` by using the default option settings.

▷ Example 1: Nonparametric ROC, AUC

[Hanley and McNeil \(1982\)](#) presented data from a study in which a reviewer was asked to classify, using a five-point scale, a random sample of 109 tomographic images from patients with neurological problems. The rating scale was as follows: 1 is definitely normal, 2 is probably normal, 3 is questionable, 4 is probably abnormal, and 5 is definitely abnormal. The true disease status was normal for 58 of the patients and abnormal for the remaining 51 patients.

Here we list 9 of the 109 observations:

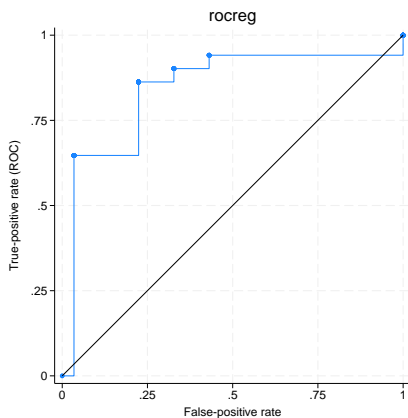
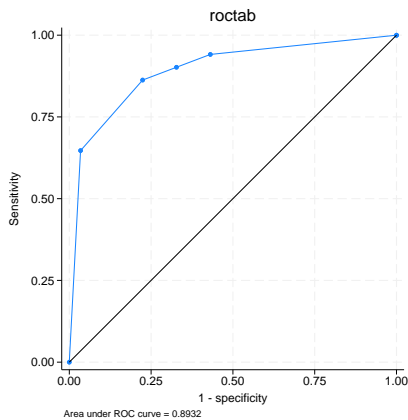
```
. use https://www.stata-press.com/data/r18/hanley
(Tomographic images)
. list disease rating in 1/9
```

	disease	rating
1.	1	5
2.	0	1
3.	1	5
4.	0	4
5.	0	1
6.	0	3
7.	1	5
8.	0	5
9.	0	1

For each observation, `disease` identifies the true disease status of the subject (0 is normal, 1 is abnormal), and `rating` contains the classification value assigned by the reviewer.

We run `roctab` on these data, specifying the `graph` option so that the ROC curve is rendered. We then calculate the false-positive and true-positive rates of the ROC curve by using `rocreg`. We graph the rates with `rocregplot`. Because we focus on `rocreg` output later, for now we use the `quietly` prefix to omit the output of `rocreg`. Both graphs are combined using `graph combine` (see [G-2] [graph combine](#)) for comparison. To ease the comparison, we specify the `aspectratio(1)` option in `roctab`; this is the default aspect ratio in `rocregplot`.

```
. roctab disease rating, graph aspectratio(1) name(a) nodraw title("roctab")
. quietly rocreg disease rating
. rocregplot, name(b) nodraw legend(off) title("rocreg")
. graph combine a b
```



Both `roctab` and `rocreg` compute the same false-positive rate and ROC values. The staircase line connection style of the graph on the right emphasizes the empirical nature of its estimates. The control distribution of the classifier is estimated using the empirical CDF estimate. Similarly, the ROC curve, the distribution of the resulting case observation false-positive rate values, is estimated using the empirical CDF. Note the footnote in the `roctab` plot. By default, `roctab` will estimate the area

under the ROC curve (AUC) using a trapezoidal approximation to the estimated false-positive rate and true-positive rate points.

The AUC can be interpreted as the probability that a randomly selected member of the case population will have a larger classifier value than a randomly selected member of the control population. It can also be viewed as the average ROC value, averaged uniformly over the (0,1) false-positive rate domain (Pepe 2003).

The nonparametric estimator of the AUC (DeLong, DeLong, and Clarke-Pearson 1988; Hanley and Hajian-Tilaki 1997) used by `rocreg` is equivalent to the sample mean of the percentile values of the case observations. Thus to calculate the nonparametric AUC estimate, we only need to calculate the percentile values of the case observations with respect to the control distribution.

This estimate can differ from the trapezoidal approximation estimate. Under discrete classification data, like we have here, there may be ties between classifier values from case to control. The trapezoidal approximation uses linear interpolation between the classifier values to correct for ties. Correcting the nonparametric estimator involves adding a correction term to each observation's percentile value, which measures the probability that the classifier is equal to (instead of less than) the observation's classifier value.

The tie-corrected nonparametric estimate (trapezoidal approximation) is used when we think the true ROC curve is smooth. This means that the classifier we measure is a discretized approximation of a true latent and a continuous classifier.

We now recompute the ROC curve of `rating` for classifying `disease` and calculate the AUC. Specifying the `tiecorrected` option allows tie correction to be used in the `rocreg` calculation. Under nonparametric estimation, `rocreg` bootstraps to obtain standard errors and confidence intervals for requested statistics. We use the default 1,000 bootstrap replications to obtain confidence intervals for our parameters. This is a reasonable lower bound to the number of replications (Mooney and Duval 1993) required for estimating percentile confidence intervals. By specifying the `summary` option in `roctab`, we will obtain output showing the trapezoidal approximation of the AUC estimate, along with standard error and confidence interval estimates for the trapezoidal approximation suggested by DeLong, DeLong, and Clarke-Pearson (1988).

```

. roctab disease rating, summary

      Obs      ROC      Std. err.      Asymptotic normal
      -----      area      -----      [95% conf. interval]
      109      0.8932      0.0307      0.83295      0.95339

. rocreg disease rating, tiecorrected bseed(29092)
(running rocregstat on estimation sample)

Bootstrap replications (1,000): .....10.....20.....30.....40....
> .....50.....60.....70.....80.....90.....100.....110...
> .....120.....130.....140.....150.....160.....170.....
> 180.....190.....200.....210.....220.....230.....240...
> .....250.....260.....270.....280.....290.....300.....
> 310.....320.....330.....340.....350.....360.....370...
> .....380.....390.....400.....410.....420.....430.....
> 440.....450.....460.....470.....480.....490.....500...
> .....510.....520.....530.....540.....550.....560.....
> 570.....580.....590.....600.....610.....620.....630...
> .....640.....650.....660.....670.....680.....690.....
> 700.....710.....720.....730.....740.....750.....760...
> .....770.....780.....790.....800.....810.....820.....
> 830.....840.....850.....860.....870.....880.....890...
> .....900.....910.....920.....930.....940.....950.....
> 960.....970.....980.....990.....1,000 done

Bootstrap results                                Number of obs = 109
                                                Replications = 1,000

Nonparametric ROC estimation
Control standardization: empirical, corrected for ties
ROC method                : empirical

Area under the ROC curve

Status      : disease
Classifier: rating

```

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
	.8931711	.0010376	.0309808	.8324498	.9538923 (N)
				.8223829	.9475383 (P)
				.8084577	.9435818 (BC)

The estimates of AUC match well. The standard error from roctab is close to the bootstrap standard error calculated by rocreg. The bootstrap standard error generalizes to the more complex models that we consider later, whereas the roctab standard-error calculation does not.



The AUC can be used to compare different classifiers. It is the most popular summary statistic for comparisons (Pepe, Longton, and Janes 2009). roccomp will compute the trapezoidal approximation of the AUC and graph the ROC curves of multiple classifiers. Using the DeLong, DeLong, and Clarke-Pearson (1988) covariance estimates for the AUC estimate, roccomp performs a Wald test of the null hypothesis that all classifier AUC values are equal. rocreg has similar capabilities.

▷ Example 2: Nonparametric ROC, AUC, multiple classifiers

Hanley and McNeil (1983) presented data from an evaluation of two computer algorithms designed to reconstruct CT images from phantoms. We will call these two algorithms modalities 1 and 2. A sample of 112 phantoms was selected; 58 phantoms were considered normal, and the remaining 54

were abnormal. Each of the two modalities was applied to each phantom, and the resulting images were rated by a reviewer using a six-point scale: 1 is definitely normal, 2 is probably normal, 3 is possibly normal, 4 is possibly abnormal, 5 is probably abnormal, and 6 is definitely abnormal. Because each modality was applied to the same sample of phantoms, the two sets of outcomes are correlated.

We list the first seven observations:

```
. use https://www.stata-press.com/data/r18/ct, clear
(Reconstruction of CT images)
. list in 1/7, sep(0)
```

	mod1	mod2	status
1.	2	1	0
2.	5	5	1
3.	2	1	0
4.	2	3	0
5.	5	6	1
6.	2	2	0
7.	3	2	0

Each observation corresponds to one phantom. The `mod1` variable identifies the rating assigned for the first modality, and the `mod2` variable identifies the rating assigned for the second modality. The true status of the phantoms is given by `status==0` if they are normal and `status==1` if they are abnormal. The observations with at least one missing rating were dropped from the analysis.

A fictitious dataset was created from this true dataset, adding a third test modality. We will use `roccomp` to compute the AUC statistic for each modality in these data and compare the AUC of the three modalities. We obtain the same behavior from `rocreg`. As before, the `tiecorrected` option is specified so that the AUC is calculated with the trapezoidal approximation.

```
. use https://www.stata-press.com/data/r18/ct2
(Reconstruction of CT images)
. roccomp status mod1 mod2 mod3, summary
```

	Obs	ROC area	Std. err.	Asymptotic normal [95% conf. interval]	
mod1	112	0.8828	0.0317	0.82067	0.94498
mod2	112	0.9302	0.0256	0.88005	0.98042
mod3	112	0.9240	0.0241	0.87670	0.97132

```
H0: area(mod1) = area(mod2) = area(mod3)
chi2(2) = 6.54 Prob>chi2 = 0.0381
```

```
. rocreg status mod1 mod2 mod3, tiecorrected bseed(38038) nodots
Bootstrap results                                     Number of obs = 112
                                                       Replications = 1,000
```

Nonparametric ROC estimation

Control standardization: empirical, corrected for ties

ROC method : empirical

Area under the ROC curve

Status : status

Classifier: mod1

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
	.8828225	-.0010192	.0318564	.820385	.94526 (N)
				.8150605	.9398384 (P)
				.8119603	.9392538 (BC)

Status : status

Classifier: mod2

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
	.9302363	.0005148	.0257043	.8798567	.9806159 (N)
				.8746504	.9769936 (P)
				.8616987	.9688995 (BC)

Status : status

Classifier: mod3

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
	.9240102	.0001857	.0240864	.8768018	.9712187 (N)
				.8727464	.9658895 (P)
				.8629984	.9621795 (BC)

H0: All classifiers have equal AUC values

Ha: At least one classifier has a different AUC value

P-value: .0339546 Test based on bootstrap (N) assumptions.

We see that the AUC estimates are equivalent, and the standard errors are quite close as well. The p -value for the tests of equal AUC under `rocreg` leads to similar inference as the p -value from `roccomp`. The Wald test performed by `rocreg` uses the joint bootstrap estimate variance matrix of the three AUC estimators rather than the [DeLong, DeLong, and Clarke-Pearson \(1988\)](#) variance estimate used by `roccomp`.

`roccomp` is used here on potentially correlated classifiers that are recorded in wide-format data. It can also be used on long-format data to compare independent classifiers. Further details can be found in [\[R\] roccomp](#).

◀

Citing the AUC's lack of clinical relevance, there is argument against using it as a key summary statistic of the ROC curve ([Pepe 2003](#); [Cook 2007](#)). [Pepe, Longton, and Janes \(2009\)](#) suggest using the estimate of the ROC curve itself at a particular point, or the estimate of the false-positive rate at a given ROC value, also known as `invROC`.

Recall from [example 1](#) how nonparametric `rocreg` graphs look, with the stairstep pattern in the ROC curve. In an ideal world, the graph would be a smooth one-to-one function, and it would be trivial to map a false-positive rate to its corresponding true-positive rate and vice versa.

However, smooth ROC curves can only be obtained by assuming a parametric model that uses linear interpolation between observed false-positive rates and between observed true-positive rates, and `rocreg` is certainly capable of that; see [example 1](#) of [\[R\] rocregplot](#). However, under nonparametric estimation, the mapping between false-positive rates and true-positive rates is not one to one, and estimates tend to be less reliable the further you are from an observed data point. This is somewhat mitigated by using tie-corrected rates (the `tiecorrected` option).

When we examine continuous data, the difference between the tie-corrected estimates and the standard estimates becomes negligible, and the empirical estimate of the ROC curve becomes close to the smooth ROC curve obtained by linear interpolation. So the nonparametric ROC and `invROC` estimates work well.

Fixing one rate value of interest can be difficult and subjective ([Pepe 2003](#)). A compromise measure is the partial area under the ROC curve (pAUC) ([McClish 1989](#); [Thompson and Zucchini 1989](#)). This is the integral of the ROC curve from 0 and above to a given false-positive rate (perhaps the largest clinically acceptable value). Like the AUC estimate, the nonparametric estimate of the pAUC can be written as a sample average of the case observation percentiles, but with an adjustment based on the prescribed maximum false-positive rate ([Dodd and Pepe 2003](#)). A tie correction may also be applied so that it reflects the trapezoidal approximation.

We cannot compare `rocreg` with `roctab` or `roccomp` on the estimation of pAUC, because pAUC is not computed by the latter two.

► Example 3: Nonparametric ROC, other statistics

To see how `rocreg` estimates ROC, `invROC`, and pAUC, we will examine a new study. [Wieand et al. \(1989\)](#) examined a pancreatic cancer study with two continuous classifiers, here called `y1` (CA 19-9) and `y2` (CA 125). This study was also examined in [Pepe, Longton, and Janes \(2009\)](#). The indicator of cancer in a subject is recorded as `d`. The study was a case–control study, stratifying participants on disease status.

We list the first five observations:

```
. use https://research.fredhutch.org/content/dam/stripe/diagnostic-biomarkers-
> statistical-center/files/wiedat2b.dta, clear
(S. Wieand - Pancreatic cancer diagnostic marker data)
. list in 1/5
```

	y1	y2	d
1.	28	13.3	no
2.	15.5	11.1	no
3.	8.2	16.7	no
4.	3.4	12.6	no
5.	17.3	7.4	no

We will estimate the ROC curves at a large value (0.7) and a small value (0.2) of the false-positive rate. These values are specified in `roc()`. The false-positive rate for ROC or sensitivity value of 0.6 will also be estimated by specifying `invroc()`. Percentile confidence intervals for these parameters are displayed in the graph obtained by `rocregplot` after `rocreg`. The pAUC statistic will be calculated for the false-positive rate of 0.5, which is specified as an argument to the `pauc()` option. Following [Pepe, Longton, and Janes \(2009\)](#), we use a stratified bootstrap, sampling separately from the case

and control populations by specifying the `bootcc` option. This reflects the case-control nature of the study.

All four statistics can be estimated simultaneously by `rocreg`. For clarity, however, we will estimate each statistic with a separate call to `rocreg`. `rocregplot` is used after estimation to graph the ROC and false-positive rate estimates. The display of the individual, observation-specific false-positive rate and ROC values will be omitted in the plot. This is accomplished by specifying `msymbol(i)` in our `plot1opts()` and `plot2opts()` options to `rocregplot`.

```
. rocreg d y1 y2, roc(.7) bseed(8378923) bootcc nodots
Bootstrap results
Number of strata = 2                                Number of obs = 141
                                                    Replications = 1,000
```

```
Nonparametric ROC estimation
Control standardization: empirical
ROC method              : empirical
```

```
ROC curve
Status      : d
Classifier: y1
```

ROC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
.7	.9222222	.0010222	.0332527	.8570482	.9873962 (N)
				.8555555	.9777778 (P)
				.8555555	.9777778 (BC)

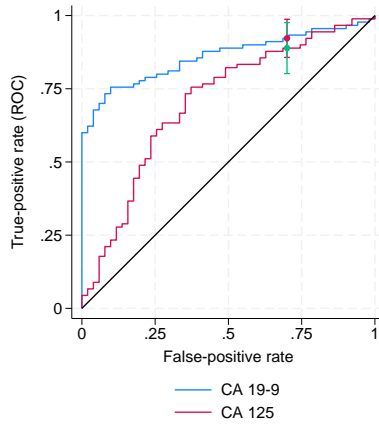
```
Status      : d
Classifier: y2
```

ROC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
.7	.8888889	-.0046556	.0444103	.8018463	.9759314 (N)
				.7833333	.9666666 (P)
				.7666667	.9555556 (BC)

```
H0: All classifiers have equal ROC values
Ha: At least one classifier has a different ROC value
Test based on bootstrap (N) assumptions
```

ROC	P-value
.7	.5537371


```
. rocregplot, plot1opts(msymbol(i)) plot2opts(msymbol(i))
```



In this study, we see that classifier y1 (CA 19-9) is a uniformly better test than is classifier y2 (CA 125) until high levels of false-positive rate and sensitivity or ROC value are reached. At the high level of false-positive rate, 0.7, the ROC value does not significantly differ between the two classifiers. This can be seen in the plot by the overlapping confidence intervals.

```
. rocreg d y1 y2, roc(.2) bseed(8378923) bootcc nodots
```

Bootstrap results

Number of strata = 2

Number of obs = 141

Replications = 1,000

Nonparametric ROC estimation

Control standardization: empirical

ROC method : empirical

ROC curve

Status : d

Classifier: y1

ROC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
.2	.7777778	.0020778	.0487666	.6821969	.8733586 (N)
				.6777778	.8722222 (P)
				.6555555	.8555555 (BC)

Status : d

Classifier: y2

ROC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
.2	.4888889	-.0054	.1348859	.2245173	.7532605 (N)
				.2222222	.6944444 (P)
				.2111111	.6777778 (BC)

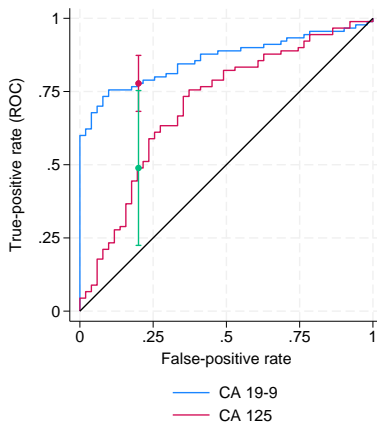
H0: All classifiers have equal ROC values

Ha: At least one classifier has a different ROC value

Test based on bootstrap (N) assumptions

ROC	P-value
.2	.0461582

```
. rocregplot, plot1opts(msymbol(i)) plot2opts(msymbol(i))
```



The sensitivity for the false-positive rate of 0.2 is found to be higher under y1 than under y2, and this difference is significant at the 0.05 level. In the plot, this is shown by the vertical confidence intervals.

```
. rocreg d y1 y2, invroc(.6) bseed(8378923) bootcc nodots
```

Bootstrap results

Number of strata = 2

Number of obs = 141

Replications = 1,000

Nonparametric ROC estimation

Control standardization: empirical

ROC method : empirical

False-positive rate

Status : d

Classifier: y1

invROC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
.6	0	.0149412	.0255885	-.0501525	.0501525 (N)
				0	.0784314 (P)
				0	.1372549 (BC)

Status : d

Classifier: y2

invROC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
.6	.254902	.0074118	.0729374	.1119474	.3978566 (N)
				.1372549	.4313726 (P)
				.1176471	.3921569 (BC)

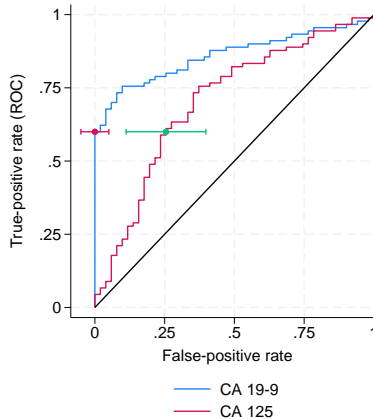
H0: All classifiers have equal invROC values

Ha: At least one classifier has a different invROC value

Test based on bootstrap (N) assumptions

invROC	P-value
.6	.0010863

```
. rocregplot, plot1opts(msymbol(i)) plot2opts(msymbol(i))
```



We find significant evidence that false-positive rates corresponding to a sensitivity of 0.6 are different from y_1 to y_2 . This is visually indicated by the horizontal confidence intervals, which are separated from each other.

```
. rocreg d y1 y2, pauc(.5) bseed(8378923) bootcc nodots
Bootstrap results
Number of strata = 2                                Number of obs = 141
                                                    Replications = 1,000
```

```
Nonparametric ROC estimation
Control standardization: empirical
ROC method           : empirical
Partial area under the ROC curve
  Status      : d
  Classifier: y1
```

pAUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
.5	.3932462	.0011971	.0219031	.3503169	.4361755 (N)
				.3489107	.4338235 (P)
				.3453159	.4315904 (BC)

```
Status      : d
Classifier: y2
```

pAUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
.5	.2496732	.0033901	.0362569	.1786109	.3207355 (N)
				.1837691	.3224946 (P)
				.1721133	.3108932 (BC)

```
H0: All classifiers have equal pAUC values
Ha: At least one classifier has a different pAUC value
Test based on bootstrap (N) assumptions
```

pAUC	P-value
.5	.001023

We also find significant evidence supporting the hypothesis that the pAUC for y_1 up to a false-positive rate of 0.5 differs from the area of the same region under the ROC curve of y_2 .



Covariate-adjusted ROC curves

When covariates affect the control distribution of the diagnostic test, thresholds for the test being classified as abnormal may be chosen that vary with the covariate values. These conditional thresholds will be more accurate than the marginal thresholds that would normally be used, because they take into account the specific distribution of the diagnostic test under the given covariate values as opposed to the marginal distribution over all covariate values.

By using these covariate-specific thresholds, we are essentially creating new classifiers for each covariate-value combination, and thus we are creating multiple ROC curves. As explained in [Pepe \(2003\)](#), when the case and control distributions of the covariates are the same, the marginal ROC curve will always be bound above by these covariate-specific ROC curves. So using conditional thresholds will never provide a less powerful test diagnostic in this case.

In the marginal ROC curve calculation, the classifiers are standardized to percentiles according to the control distribution, marginalized over the covariates. Thus, the ROC curve is the CDF of the standardized case observations. The covariate-adjusted ROC curve is the CDF of one minus the conditional control percentiles for the case observations, and the marginal ROC curve is the CDF of one minus the marginal control percentiles for the case observations (Pepe and Cai 2004). Thus, the standardization of classifier to false-positive rate value is conditioned on the specific covariate values under the covariate-adjusted ROC curve.

The covariate-adjusted ROC curve (Janes and Pepe 2009) at a given false-positive rate t is equivalent to the expected value of the covariate-specific ROC at t over all covariate combinations. When the covariates in question do not affect the case distribution of the classifier, the covariate-specific ROC will have the same value at each covariate combination. So here the covariate-adjusted ROC is equivalent to the covariate-specific ROC, regardless of covariate values.

When covariates do affect the case distribution of the classifier, users of the diagnostic test would likely want to model the covariate-specific ROC curves separately. Tools to do this can be found in the parametric modeling discussion in the following two sections. Regardless, the covariate-adjusted ROC curve can serve as a meaningful summary of covariate-adjusted accuracy.

Also note that the ROC summary statistics defined in the previous section have covariate-adjusted analogs. These analogs are estimated in a similar manner as under the marginal ROC curve (Janes, Longton, and Pepe 2009). The options for their calculation in `rocreg` are identical to those given in the previous section. Further details can be found in *Methods and formulas*.

► Example 4: Nonparametric ROC, linear covariate adjustment

Norton et al. (2000) studied data from a neonatal audiology study on three tests to identify hearing impairment in newborns. These data were also studied in Janes, Longton, and Pepe (2009). Here we list 5 of the 5,058 observations.

```
. use https://www.stata-press.com/data/r18/nnhs, clear
(Norton - neonatal audiology data)

. list in 1/5
```

	id	ear	male	currence	d	y1	y2	y3
1.	B0157	R	M	42.42	0	-3.1	-9	-1.5
2.	B0157	L	M	42.42	0	-4.5	-8.7	-2.71
3.	B0158	R	M	40.14	1	-3.2	-13.2	-2.64
4.	B0161	L	F	38.14	0	-22.1	-7.8	-2.59
5.	B0167	R	F	37	0	-10.9	-6.6	-1.42

The classifiers `y1` (DPOAE 65 at 2 kHz), `y2` (TEOAE 80 at 2 kHz), and `y3` (ABR) and the hearing impairment indicator `d` are recorded along with some relevant covariates. The infant's age is recorded in months as `currence`, and the infant's gender is indicated by `male`. Over 90% of the newborns were tested in each ear (`ear`), so we will cluster on infant ID (`id`).

Following the strategy of Janes, Longton, and Pepe (2009), we will first perform ROC analysis for the classifiers while adjusting for the covariate effects of the infant's gender and age. This is done by specifying these variables in the `ctrlcov()` option. We adjust using a linear regression rule, by specifying `ctrlmodel(linear)`. This means that when a user of the diagnostic test chooses a threshold conditional on the age and gender covariates, they assume that the diagnostic test classifier has some linear dependence on age and gender and equal variance as their levels vary. Our cluster adjustment is made by specifying the `cluster()` option.

We will focus on the first classifier. The percentile, or specificity, values are calculated empirically by default, and thus so are the false-positive rates, $(1 - \text{specificity})$. Also by default, the ROC curve values are empirically defined by the false-positive rates. To draw the ROC curve, we again use `rocregplot`.

The AUC is calculated by default. For brevity, we specify the `nobootstrap` option so that bootstrap sampling is not performed. The AUC point estimate will be sufficient for our purposes.

```
. rocreg d y1, ctrlcov(male currage) ctrlmodel(linear) cluster(id) nobootstrap
Nonparametric ROC estimation                               Number of obs = 5,056
Covariate control      : linear regression
Control variables      : male currage
Control standardization: empirical
ROC method             : empirical
Status                : d
Classifier: y1
Covariate control adjustment model:
Linear regression
                                Number of obs      =      4,907
                                F(2, 2685)         =      13.80
                                Prob > F            =      0.0000
                                R-squared           =      0.0081
                                Root MSE        =      7.7515
                                (Std. err. adjusted for 2,686 clusters in id)
```

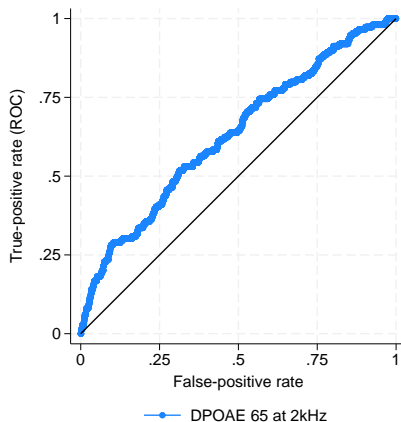
y1	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
male	.2471744	.2603598	0.95	0.343	-.2633516	.7577005
currage	-.2032456	.0389032	-5.22	0.000	-.2795288	-.1269624
_cons	-1.239484	1.487855	-0.83	0.405	-4.156942	1.677973

Area under the ROC curve

```
Status      : d
Classifier: y1
```

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
	.6293994	.	.	.	(N)
				.	(P)
				.	(BC)

```
. rocregplot
```



Our covariate control adjustment model shows that `currage` has a negative effect on `y1` (DPOAE 65 at 2 kHz) under the control population. At the 0.001 significance level, we reject that its contribution to `y1` is zero, and the point estimate has a negative sign. This result does not directly tell us about the effect of `currage` on the ROC curve of `y1` as a classifier of `d`. None of the case observations are used in the linear regression, so information on `currage` for abnormal cases is not used in the model. This result does show us how to calculate false-positive rates for tests that use thresholds conditional on a child's sex and current age. We will see how `currage` affects the ROC curve when `y1` is used as a classifier and conditional thresholds are used based on `male` and `currage` in the following section, *Parametric ROC curves: Estimating equations*.

□ Technical note

Under this nonparametric estimation, `rocreg` saved the false-positive rate for each observation's `y1` values in the utility variable `_fpr_y1`. The true-positive rates are stored in the utility variable `_roc_y1`. For other models, say with classifier `yname`, these variables would be named `_fpr_yname` and `_roc_yname`. They will also be overwritten with each call of `rocreg`. The variables `_roc_*` and `_fpr_*` are usually for internal `rocreg` use only and are overwritten with each call of `rocreg`. They are only created for nonparametric models or parametric models that do not involve ROC covariates. In these models, covariates may only affect the first stage of estimation, the control distribution, and not the ROC curve itself. In parametric models that allow ROC covariates, different covariate values would lead to different ROC curves. □

To see how the covariate-adjusted ROC curve estimate differs from the standard marginal estimate, we will reestimate the ROC curve for classifier `y1` without covariate adjustment. We rename these variables before the new estimation and then draw an overlaid `twoway line` (see [G-2] [graph twoway line](#)) plot to compare the two.

```

. rename _fpr_y1 o_fpr_y1
. rename _roc_y1 o_roc_y1
. label variable o_roc_y1 "Covariate adjusted"
. rocreg d y1, cluster(id) nobootstrap
Nonparametric ROC estimation                               Number of obs = 5,058
Control standardization: empirical
ROC method          : empirical
Area under the ROC curve
  Status           : d
  Classifier        : y1

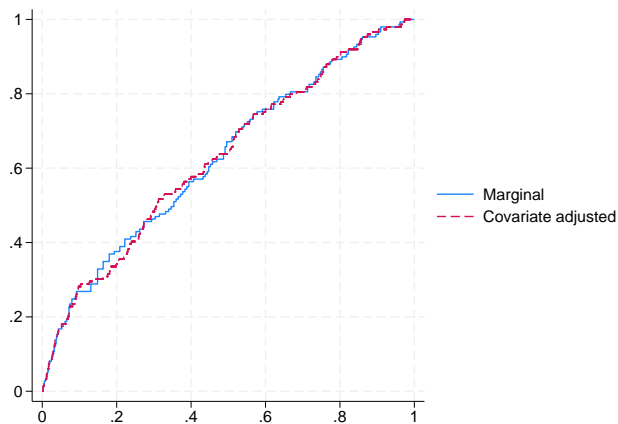
```

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]
	.6279645	.	.	. (N)
				. (P)
				. (BC)

```

. label variable _roc_y1 "Marginal"
. label variable o_fpr_y1 "False-positive rate for y1"
. twoway line _roc_y1 _fpr_y1, sort(_fpr_y1 _roc_y1) connect(J) ||
> line o_roc_y1 o_fpr_y1, sort(o_fpr_y1 o_roc_y1)
> connect(J) lpattern(dash) aspectratio(1) legend(cols(1))

```



Though they are close, particularly in AUC, there are clearly some points of difference between the estimates. So the covariate-adjusted ROC curve may be useful here.

◀

In our examples thus far, we have used the empirical CDF estimator to estimate the control distribution. `rocreg` allows some flexibility here. The `pvc(normal)` option may be specified to calculate the percentile values according to a Gaussian distribution of the control.

Covariate adjustment in `rocreg` may also be performed with stratification instead of linear regression. Under the stratification method, the unique values of the stratified covariates each define separate parameters for the control distribution of the classifier. A user of the diagnostic test chooses a threshold based on the control distribution conditioned on the unique covariate value parameters.

We will demonstrate the use of normal percentile values and covariate stratification in our next example.

► Example 5: Nonparametric ROC, covariate stratification

The hearing test study of [Stover et al. \(1996\)](#) examined the effectiveness of negative signal-to-noise ratio, `nsnr`, as a classifier of hearing loss. The test was administered under nine different settings, corresponding to different frequency, `xf`, and intensity, `x1`, combinations. Here we list 10 of the 1,848 observations.

```
. use https://www.stata-press.com/data/r18/dp, clear
(Stover - DPOAE test data)
. list in 1/10
```

	id	d	nsnr	xf	x1	xd
1.	101	1	18	10.01	5.5	3.5
2.	101	1	19	20.02	5.5	3
3.	101	1	7.6	10.01	6	3.5
4.	101	1	15	20.02	6	3
5.	101	1	16	10.01	6.5	3.5
6.	101	1	5.8	20.02	6.5	3
7.	102	0	-2.6	10.01	5.5	.
8.	102	0	-3	14.16	5.5	.
9.	102	1	10	20.02	5.5	1
10.	102	0	-5.8	10.01	6	.

Hearing loss is represented by `d`. The covariate `xd` is a measure of the degree of hearing loss. We will use this covariate in later analysis, because it only affects the case distribution of the classifier. Multiple measurements are taken for each individual, `id`, so we will cluster by individual.

We evaluate the effectiveness of `nsnr` using `xf` and `x1` as stratification covariates with `rocreg`; the default method of covariate adjustment.

As mentioned before, the default false-positive rate calculation method in `rocreg` estimates the conditional control distribution of the classifiers empirically. For comparison, we will also estimate a separate ROC curve using false-positive rates assuming the conditional control distribution is normal. This behavior is requested by specifying the `pvc(normal)` option. Using the `rocregplot` option `name()` to store the ROC plots and using the `graph combine` command, we are able to compare the Gaussian and empirical ROC curves side by side. As before, for brevity we specify the `nobootstrap` option to suppress bootstrap sampling.

```
. rocreg d nsnr, ctrlcov(xf x1) cluster(id) nobootstrap
Nonparametric ROC estimation                                Number of obs = 1,848
Covariate control      : stratification
Control variables      : xf x1
Control standardization: empirical
ROC method             : empirical

Area under the ROC curve
  Status      : d
  Classifier: nsnr
```

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]
	.9264192	.	.	. (N)
				. (P)
				. (BC)

```
. rocregplot, title(Empirical FPR) name(a) nodraw
```

```

. rocreg d nsnr, pvc(normal) ctrlcov(xf xl) cluster(id) nobootstrap
Nonparametric ROC estimation                               Number of obs = 1,848
Covariate control      : stratification
Control variables      : xf xl
Control standardization: normal
ROC method              : empirical
Area under the ROC curve
  Status      : d
  Classifier: nsnr

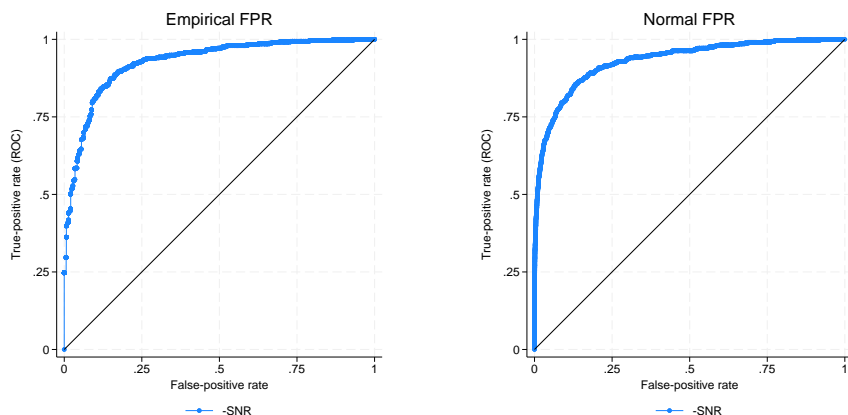
```

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]
	.9309901	.	.	. (N)
				. (P)
				. (BC)

```

. rocregplot, title(Normal FPR) name(b) nodraw
. graph combine a b, xsize(5)

```



On cursory visual inspection, we see little difference between the two curves. The AUC values are close as well. So it is sensible to assume that we have Gaussian percentile values for control standardization. ↵

Parametric ROC curves: Estimating equations

We now assume a parametric model for covariate effects on the second stage of ROC analysis. Particularly, the ROC curve is a probit model of the covariates. We will thus have a separate ROC curve for each combination of the relevant covariates.

Under weak assumptions about the control distribution of the classifier, we can fit this model by using estimating equations as described in [Alonzo and Pepe \(2002\)](#). This method can be also be used without covariate effects in the second stage, assuming a parametric model for the single (constant only) ROC curve. Covariates may still affect the first stage of estimation, so we parametrically model the single covariate-adjusted ROC curve (from the previous section). The marginal ROC curve, involving no covariates in either stage of estimation, can be fit parametrically as well.

In addition to the [Alonzo and Pepe \(2002\)](#) explanation, further details are given in [Pepe, Longton, and Janes \(2009\)](#); [Janes, Longton, and Pepe \(2009\)](#); [Pepe \(2003\)](#); and [Janes and Pepe \(2009\)](#).

The parametric models that we consider assume that the ROC curve is a cumulative distribution function g invoked with input of a linear polynomial in the corresponding quantile function invoked on the false-positive rate u . In this context, we assume that g corresponds to a standard normal cumulative distribution function, Φ . So the corresponding quantile function is Φ^{-1} . The constant intercept of the polynomial may depend on covariates, but the slope term α (the quantile coefficient) may not.

$$\text{ROC}(u) = g\{\mathbf{x}'\beta + \alpha g^{-1}(u)\}$$

The first step of the algorithm involves the choice of false-positive rates to use in the parametric fit. These are typically a set of equispaced points spanning the interval $(0,1)$. [Alonzo and Pepe \(2002\)](#) examined the effect of fitting large and small sets of points, finding that relatively small sets could be used with little loss of efficiency. Alternatively, the set can be formed by using the observed false-positive rates in the data ([Pepe 2003](#)). Further details on the algorithm are provided in [Methods and formulas](#).

Under parametric estimation, all the summary measures we defined earlier, except the AUC, are not calculated until postestimation. In models with covariates, each covariate combination would yield a different ROC curve and thus different summary parameters, so no summary parameters are initially estimated. In marginal parametric models (where there are no ROC covariates, but there are potentially control covariates), we will calculate the AUC and leave the other measures for postestimation; see [\[R\] rocreg postestimation](#). As with the other parameters, we bootstrap for standard errors and inference.

We will now demonstrate how `rocreg` performs the [Alonzo and Pepe \(2002\)](#) algorithm using the previous section's examples and others.

▷ Example 6: Parametric ROC, linear covariate adjustment

We return to the neonatal audiology study with gender and age covariates ([Norton et al. 2000](#)), which we discussed in [example 4](#). [Janes, Longton, and Pepe \(2009\)](#) suspected the current age of the infant would play a role in the case distribution of the classifier `y1` (DPOAE 65 at 2 kHz). They postulated a probit link between the ROC curve and the covariate-adjusted false-positive rates. We follow their investigation and reach similar results.

In [example 4](#), we saw the results of adjusting for the `currage` and `male` variables in the control population for classifier `y1`. Now, we see how `currage` affects the ROC curve when `y1` is used with thresholds conditioned on `male` and `currage`.

We specify the covariates that should affect the ROC curve in the `roccov()` option. By default, `rocreg` will choose 10 equally spaced false-positive rates in the $(0,1)$ interval as fitting points. The `fprpts()` option allows the user to specify more or fewer points. We specify the `bsave()` option with the `mnhs2y1` dataset so that we can use the bootstrap resamples in postestimation.

```
. use https://www.stata-press.com/data/r18/nmhs, clear
(Norton - neonatal audiology data)
. rocreg d y1, probit ctrlcov(currage male) ctrlmodel(linear) roccov(currage)
> cluster(id) bseed(56930) bsave(nmhs2y1) nodots
```

Bootstrap results Number of obs = 5,056
Replications = 1,000

Parametric ROC estimation

```
Covariate control      : linear regression
Control variables     : currage male
Control standardization: empirical
ROC method            : parametric          Link: probit
Status                : d
Classifier: y1
Covariate control adjustment model:
```

```
Linear regression      Number of obs   =    4,907
                      F(2, 2685)           =   13.80
                      Prob > F             =    0.0000
                      R-squared            =    0.0081
                      Root MSE           =    7.7515
```

(Std. err. adjusted for 2,686 clusters in id)

y1	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
currage	-.2032456	.0389032	-5.22	0.000	-.2795288	-.1269624
male	.2471744	.2603598	0.95	0.343	-.2633516	.7577005
_cons	-1.239484	1.487855	-0.83	0.405	-4.156942	1.677973

```
Status      : d
Classifier: y1
ROC Model :
```

(Replications based on 2,741 clusters in id)

y1	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]		
_cons	-1.272505	-.058656	1.157249	-3.540671	.995661	(N)
				-3.703316	.8687538	(P)
				-3.550433	1.094785	(BC)
currage	.0448228	.0015634	.0300731	-.0141194	.1037649	(N)
				-.0107322	.108762	(P)
				-.0156332	.1044122	(BC)
probit						
_cons	.9372393	.0153781	.0739921	.7922176	1.082261	(N)
				.8027433	1.108293	(P)
				.78655	1.077874	(BC)

Note how the number of clusters—here infants—changes from the covariate control adjustment model fit to the ROC model. The control fit is limited to control cases and thus fewer infants. The ROC is fit on all the data, so the variance is adjusted for all clustering on all infants.

With a 0.05 level of statistical significance, we cannot reject the null hypothesis that *currage* has no effect on the ROC curve at a given false-positive rate. This is because each of our 95% bootstrap confidence intervals contains 0. This corresponds with the finding in [Janes, Longton, and Pepe \(2009\)](#) where the reported 95% intervals each contained 0. We cannot reject that the intercept parameter β_0 , reported as *_cons* in the main table, is 0 at the 0.05 level either. The slope parameter α , reported

as `_cons` in the `probit` table, is close to 1 and cannot be rejected as being 1 at the 0.05 level. Under the assumption that the ROC coefficients except α are 0 and that $\alpha = 1$, the ROC curve at false-positive rate u is equal to u . In other words, we cannot reject that the false-positive rate is equal to the true-positive rate, and so the test is noninformative. Further investigation of the results requires postestimation; see [R] [rocreg postestimation](#).

◀

The fitting point set can be formed by using the observed false-positive rates (Pepe 2003). Our next example will illustrate this.

▷ Example 7: Parametric ROC, covariate stratification

We return to the hearing test study of Stover et al. (1996), which we discussed in [example 5](#). Pepe (2003) suspected that intensity, `xd`, would play a role in the case distribution of the negative signal-to-noise ratio (`nsnr`) classifier. A ROC regression was fit with covariate adjustment for `xf` and `x1` with stratification, and for ROC covariates `xf`, `x1`, and `xd`. There is no prohibition against the same covariate being used in the first and second stages of ROC calculation. The false-positive rate fitting point set was composed of all observed false-positive rates in the control data.

We fit the model with `rocreg` here. Using observed false-positive rates as the fitting point set can make the dataset very large, so fitting the model is computationally intensive. We demonstrate the fitting algorithm without precise confidence intervals, focusing instead on the coefficient estimates and standard errors. We will thus perform only 50 bootstrap replications, a reasonable number to obtain accurate standard error estimates (Mooney and Duval 1993). The number of replications is specified in the `breps()` option.

The ROC covariates are specified in `roccov()`. We specify that all observed false-positive rates in the control observations be used as fitting points with the `ctrlfprall` option. The `nobstrata` option specifies that the bootstrap is not stratified. The covariate stratification in the first stage of estimation does not affect the resampling. We will return to this example in postestimation, so we save the bootstrap results in the `nsnrf` dataset with the `bsave()` option.

```

. use https://www.stata-press.com/data/r18/dp
(Stover - DPOAE test data)
. rocreg d nsnr, probit ctrlcov(xf xl) roccov(xf xl xd) ctrlfprall cluster(id)
> nobstrata bseed(156385) breps(50) bsave(nsnrf)
(running rocregstat on estimation sample)
Bootstrap replications (50): .....10.....20.....30.....40.....
> ..50 done

Bootstrap results                                     Number of obs = 1,848
                                                    Replications = 50

Parametric ROC estimation
Covariate control      : stratification
Control variables      : xf xl
Control standardization: empirical
ROC method             : parametric                Link: probit
Status                : d
Classifier: nsnr
ROC Model :
                                                    (Replications based on 208 clusters in id)

```

nsnr	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
_cons	3.247872	.0868351	.8985028	1.486839	5.008905 (N)
				1.834415	5.606226 (P)
				1.834415	6.275457 (BC)
xf	.0502557	.0079289	.0290622	-.0067051	.1072166 (N)
				-.0033383	.1145611 (P)
				-.0454014	.0883843 (BC)
xl	-.4327223	-.024214	.1249467	-.6776134	-.1878313 (N)
				-.7207585	-.2425129 (P)
				-.7207585	-.1547958 (BC)
xd	.4431764	.0200785	.0875782	.2715264	.6148264 (N)
				.3388809	.6706273 (P)
				.3388809	.6706273 (BC)
probit					
_cons	1.032657	.0026243	.1287713	.7802699	1.285044 (N)
				.8308481	1.284435 (P)
				.7808038	1.284435 (BC)

We obtain results similar to those reported in [Pepe \(2003, 159\)](#). We find that the coefficients for `xl` and `xd` differ from 0 at the 0.05 level of significance. So over certain covariate combinations, we can have a variety of informative tests using `nsnr` as a classifier.

◀

As mentioned before, when there are no covariates, `rocreg` can still fit a parametric model for the ROC curve of a classifier by using the [Alonzo and Pepe \(2002\)](#) method. `roccomp` and `rocfit` can fit marginal probit models as well. We will compare the behavior of `rocreg` with that of `roccomp` and `rocfit` for probit models without covariates.

When the `binormal` option is specified, `roccomp` calculates the AUC for input classifiers according to the maximum likelihood algorithm of `rocfit`. The `rocfit` algorithm expects discrete classifiers but can slice continuous classifiers into discrete partitions. Further, the case and control distributions are both assumed normal. Actually, the observed classification values are taken as discrete indicators of the latent normally distributed classification values. This method is documented in [Dorfman and Alf \(1969\)](#).

Alonzo and Pepe (2002) compared their estimating equations probability density function method (with empirical estimation of the false-positive rates) to the maximum likelihood approach of Dorfman and Alf (1969) and found that they had similar efficiency and mean squared error. So we should expect rocfit and rocreg to give similar results when fitting a simple probit model.

► Example 8: Parametric ROC, marginal model

We return to the Hanley and McNeil (1982) data. We will fit a probit model to the ROC curve, assuming that the rating variable is a discrete indicator of an underlying latent normal random variable in both the case and control populations of disease. We invoke rocfit with the default options. rocreg is invoked with the probit option. The percentile values are calculated empirically. Because there are fewer categories than 10, there will be fewer than 10 false-positive rates that trigger a different true-positive rate value. So for efficiency, we invoke rocreg with the ctrlfprall option.

```
. use https://www.stata-press.com/data/r18/hanley
(Tomographic images)
. rocfit disease rating, nolog
Binormal model of disease on rating           Number of obs   =           109
Goodness-of-fit chi2(2) =           0.21
Prob > chi2           =           0.9006
Log likelihood        = -123.64855
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
intercept	1.656782	0.310456	5.34	0.000	1.048300	2.265265
slope (*)	0.713002	0.215882	-1.33	0.184	0.289881	1.136123
/cut1	0.169768	0.165307	1.03	0.304	-0.154227	0.493764
/cut2	0.463215	0.167235	2.77	0.006	0.135441	0.790990
/cut3	0.766860	0.174808	4.39	0.000	0.424243	1.109477
/cut4	1.797938	0.299581	6.00	0.000	1.210770	2.385106

Index	Indices from binormal fit		
	Estimate	Std. err.	[95% conf. interval]
ROC area	0.911331	0.029506	0.853501 0.969161
delta(m)	2.323671	0.502370	1.339044 3.308298
d(e)	1.934361	0.257187	1.430284 2.438438
d(a)	1.907771	0.259822	1.398530 2.417012

(*) z test for slope==1

```

. rocreg disease rating, probit ctrlfprall bseed(8574309) nodots
Bootstrap results                                     Number of obs = 109
                                                    Replications = 1,000

Parametric ROC estimation
Control standardization: empirical
ROC method      : parametric                      Link: probit
Status         : disease
Classifier      : rating
ROC Model      :

```

rating	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]		
_cons	1.635041	.0850129	.3706472	.9085857	2.361496	(N)
				1.139856	2.649876	(P)
				1.103894	2.428801	(BC)
probit						
_cons	.6951252	.0642966	.275061	.1560155	1.234235	(N)
				.3242299	1.409152	(P)
				.2721681	1.292525	(BC)
AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]		
	.9102903	-.0029679	.0300486	.8513963	.9691844	(N)
				.8448006	.9602325	(P)
				.8475004	.9607949	(BC)

We see that the intercept and slope parameter estimates are close. The intercept (`_cons` in the main table) is clearly nonzero. Under `rocreg`, the slope (`_cons` in the `probit` table) and its percentile and bias-corrected confidence intervals are close to those of `rocfite`. The area under the ROC curve for each of the `rocreg` and `rocfite` estimators also matches closely.

◀

Now, we will compare the parametric fit of `rocreg` under the constant `probit` model with `roccomp`.

▶ Example 9: Parametric ROC, marginal model, multiple classifiers

We now use the fictitious dataset generated from [Hanley and McNeil \(1983\)](#). To fit a `probit` model using `roccomp`, we specify the `binormal` option. Our specification of `rocreg` remains the same as before.

`rocregplot` is used to render the model produced by `rocreg`. We specify several graph options to both `roccomp` and `rocregplot` to ease comparison. When the `binormal` option is specified along with `graph`, `roccomp` will draw the binormal fitted lines in addition to connected line plots of the empirical false-positive and true-positive rates.

In this plot, we overlay scatterplots of the empirical false-positive rates (because percentile value calculation defaulted to `pvc(empirical)`) and the parametric true-positive rates.


```
. use https://www.stata-press.com/data/r18/ct2, clear
(Reconstruction of CT images)

. roccomp status mod1 mod2 mod3, summary binormal graph aspectratio(1)
>   plotlopts(connect(i) msymbol(o))
>   plot2opts(connect(i) msymbol(s))
>   plot3opts(connect(i) msymbol(t))
>   legend(label(1 "Modality 1") label(3 "Modality 2")
>           label(5 "Modality 3") label(2 "Modality 1 fit")
>           label(4 "Modality 2 fit") label(6 "Modality 3 fit")
>           order(1 3 5 2 4 6) pos(6))
>   title(roccomp) name(a) nodraw
Fitting binormal model for: mod1
Fitting binormal model for: mod2
Fitting binormal model for: mod3
```

	Obs	ROC area	Std. err.	[95% conf. interval]	
mod1	112	0.8945	0.0305	0.83482	0.95422
mod2	112	0.9382	0.0264	0.88647	0.99001
mod3	112	0.9376	0.0223	0.89382	0.98139

H0: area(mod1) = area(mod2) = area(mod3)
 chi2(2) = 8.27 Prob>chi2 = 0.0160

```
. rocreg status mod1 mod2 mod3, probit ctrlfprall bseed(867340912) nodots
Bootstrap results
Number of obs = 112
Replications = 999
```

Parametric ROC estimation

Control standardization: empirical
 ROC method : parametric Link: probit
 Status : status
 Classifier: mod1
 ROC Model :

mod1	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
_cons	1.726034	.164964	.5823832	.5845836	2.867484 (N)
				1.197595	3.410778 (P)
				1.154531	3.027969 (BC)
probit	.9666323	.1104948	.4635417	.0581071	1.875157 (N)
				.5102274	2.319844 (P)
				.5193889	2.319844 (BC)
AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]	
	.8927007	.000062	.0306285	.83267	.9527315 (N)
				.8297837	.946722 (P)
				.8262202	.9423347 (BC)

Status : status
 Classifier: mod2
 ROC Model :

mod2	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]		
_cons	1.696811	.0760455	.4750493	.7657314	2.627891 (N)	
				1.191126	2.854689 (P)	
				1.205256	2.916377 (BC)	
probit						
_cons	.4553828	.0245707	.304156	-.140752	1.051518 (N)	
				.0857558	1.070745 (P)	
				.1495717	1.434937 (BC)	

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]		
	.938734	-.0033942	.0268351	.8861382	.9913297 (N)	
				.875	.9774636 (P)	
				.8775983	.9777322 (BC)	

Status : status
 Classifier: mod3
 ROC Model :

mod3	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]		
_cons	2.281359	.1143008	.5773577	1.149758	3.412959 (N)	
				1.653256	3.882332 (P)	
				1.65594	3.882332 (BC)	
probit						
_cons	1.107736	.0482007	.4195496	.2854334	1.930038 (N)	
				.6128833	2.256342 (P)	
				.6514254	2.527536 (BC)	

AUC	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]		
	.9368321	-.0008781	.0226477	.8924435	.9812207 (N)	
				.887858	.9720291 (P)	
				.8866298	.971411 (BC)	

H0: All classifiers have equal AUC values

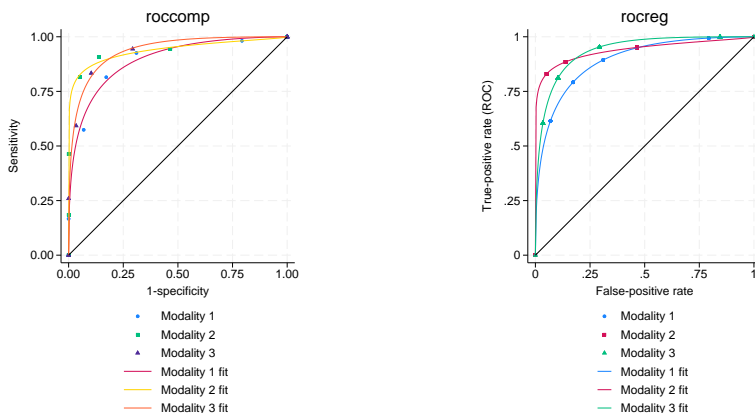
Ha: At least one classifier has a different AUC value

P-value: .0599896 Test based on bootstrap (N) assumptions.

. rocregplot, title(rocreg) nodraw name(b)

> plot1opts(msymbol(o)) plot2opts(msymbol(s)) plot3opts(msymbol(t))

```
. graph combine a b, xsize(5)
```



We see differing true-positive rate values in the scattered points, which is expected because `roccomp` gives the empirical estimate and `rocreg` gives the parametric estimate. However, the estimated curves and areas under the ROC curve look similar. Using the Wald test based on the bootstrap covariance, `rocreg` rejects the null hypothesis that each test has the same AUC at the 0.1 significance level. `roccomp` formulates the asymptotic covariance using the `rocfit` estimates of AUC. Examination of its output leads to rejection of the null hypothesis that the AUCs are equal across each test at the 0.05 significance level.



Parametric ROC curves: Maximum likelihood

The [Alonzo and Pepe \(2002\)](#) method of fitting a parametric model to the ROC curve is powerful because it can be generally applied, but that can be a limitation as well. Whenever we invoke the method and want anything other than point estimates of the parameters, we must perform bootstrap resampling.

An alternative is to use maximum likelihood inference to fit the ROC curve. This method can save computational time by avoiding the bootstrap.

`rocreg` implements maximum likelihood estimation for ROC curve analysis when both the case and control populations are normal. Particularly, the classifier is a normal linear model on certain covariates, and the covariate effect and variance of the classifier may change between the case and control populations. This model is defined in [Pepe \(2003, 145\)](#).

$$y = \mathbf{z}'\beta_0 + D\mathbf{x}'\beta_1 + \sigma(D)\epsilon$$

Our error term, ϵ , is a standard normal random variable. The variable D is our true status variable, being 1 for the case population observations and 0 for the control population observations. The variance function σ is defined as

$$\sigma(D) = \sigma_0(D = 0) + \sigma_1(D = 1)$$

This provides two variance parameters in the model and does not depend on covariate values.

Suppose a covariate x_i is present in \mathbf{z} and \mathbf{x} . The coefficient β_{1i} represents the interaction effect of the x_i and D . It is the extra effect that x_i has on classifier y under the case population, $D = 1$, beyond the main effect β_{0i} . These $\beta_{\mathbf{1}}$ coefficients are directly related to the ROC curve of y .

Under this model, the ROC curve is derived to be

$$\text{ROC}(u) = \Phi \left[\frac{1}{\sigma_1} \{ \mathbf{x}'\beta_{\mathbf{1}} + \sigma_0 \Phi^{-1}(u) \} \right]$$

For convenience, we reparameterize the model at this point, creating the parameters $\beta_i = \sigma_1^{-1} \beta_{1i}$ and $\alpha = \sigma_1^{-1} \sigma_0$. We refer to β_0 as the constant intercept, `i_cons`. The parameter α is referred to as the constant slope, `s_cons`.

$$\text{ROC}(u) = \Phi \{ \mathbf{x}'\beta + \alpha \Phi^{-1}(u) \}$$

We may interpret the final coefficients as the standardized linear effect of the ROC covariate on the classifier under the case population. The marginal effect of the covariate on the classifier in the control population is removed, and it is rescaled by the case population standard deviation of the classifier when all ROC covariate effects are removed. An appreciable effect on the classifier by a ROC covariate in this measure leads to an appreciable effect on the classifier's ROC curve by the ROC covariate.

The advantage of estimating the control coefficients β_0 is similar to the gains of estimating the covariate control models in the estimating equations ROC method and nonparametric ROC estimation. This model would similarly apply when evaluating a test that is conditioned on control covariates.

Again, we note that under parametric estimation, all the summary measures we defined earlier except the AUC are not calculated until postestimation. In models with covariates, each covariate combination would yield a different ROC curve and thus different summary parameters, so no summary parameters are estimated initially. In marginal parametric models, we will calculate the AUC and leave the other measures for postestimation. There is a simple closed-form formula for the AUC under the probit model. Using this formula, the delta method can be invoked for inference on the AUC. Details on AUC estimation for probit marginal models are found in *Methods and formulas*.

We will demonstrate the maximum likelihood method of `rocreg` by revisiting the models of the previous section.

► Example 10: Maximum likelihood ROC, single classifier

Returning to the hearing test study of [Stover et al. \(1996\)](#), we use a similar covariate grouping as before. The frequency `xf` and intensity `x1` are control covariates (\mathbf{z}), while all three covariates `xf`, `x1`, and hearing loss degree `xd` are case covariates (\mathbf{x}). In [example 7](#), we fit this model using the [Alonzo and Pepe \(2002\)](#) method. Earlier we stratified on the control covariates and estimated the conditioned control distribution of `nsnr` empirically. Now, we assume a normal linear model for `nsnr` on `xf` and `x1` under the control population.

We fit the model by specifying the control covariates in the `ctrlcov()` option and the case covariates in the `roccov()` option. The `m1` option tells `rocreg` to perform maximum likelihood estimation.

```

. use https://www.stata-press.com/data/r18/dp, clear
(Stover - DPOAE test data)

. rocreg d nsnr, ctrlcov(xf xl) roccov(xf xl xd) probit ml cluster(id) nolog
Parametric ROC estimation                                Number of obs = 112
                                                         Replications = 999

Covariate control      : linear regression
Control variables      : xf xl
Control standardization: normal
ROC method             : parametric                    Link: probit

Status      : d
Classifiers: nsnr

Classifier   : nsnr
Covariate control adjustment model:
              (Std. err. adjusted for 208 clusters in id)

```

	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
casecov						
xf	.4690907	.1408683	3.33	0.001	.192994	.7451874
xl	-3.187785	.8976521	-3.55	0.000	-4.947151	-1.42842
xd	3.042998	.3569756	8.52	0.000	2.343339	3.742657
_cons	23.48064	5.692069	4.13	0.000	12.32439	34.63689
casesd						
_cons	7.979708	.354936	22.48	0.000	7.284047	8.67537
ctrlcov						
xf	-.1447499	.0615286	-2.35	0.019	-.2653438	-.0241561
xl	-.8631348	.2871976	-3.01	0.003	-1.426032	-.3002378
_cons	1.109477	1.964004	0.56	0.572	-2.7399	4.958854
ctrlsd						
_cons	7.731203	.3406654	22.69	0.000	7.063511	8.398894

```

Status      : d
ROC Model   :
              (Std. err. adjusted for 208 clusters in id)

```

	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
nsnr						
i_cons	2.942543	.7569821	3.89	0.000	1.458885	4.426201
xf	.0587854	.0175654	3.35	0.001	.024358	.0932129
xl	-.3994865	.1171914	-3.41	0.001	-.6291775	-.1697955
xd	.381342	.0449319	8.49	0.000	.2932771	.4694068
s_cons	.9688578	.0623476	15.54	0.000	.8466587	1.091057

We find the results are similar to those of [example 7](#). Frequency (xf) and intensity (xl) have a negative effect on the classifier nsnr in the control population.

The negative control effect is mitigated for xf in the case population, but the effect for xl is even more negative there. Hearing loss severity, xd, has a positive effect on nsnr in the case population, and it is undefined in the control population.

The ROC coefficients are shown in the ROC Model table. Each are different from 0 at the 0.05 level. At this level, we also cannot conclude that the variances differ from case to control populations,

because 1 is in the 95% confidence interval for `s_cons`, the ratio of the case to control standard deviation parameters.

Both frequency (`xf`) and hearing loss severity (`xd`) make a positive contribution to the ROC curve and thus make the test more powerful. Intensity (`x1`) has a negative effect on the ROC curve and weakens the test. We previously saw in [example 5](#) that the control distribution appears to be normal, so using maximum likelihood to fit this model is a reasonable approach.

This model was also fit in [Pepe \(2003, 147\)](#). Pepe used separate least-squares estimates for the case and control samples. We obtain similar results for the coefficients, but the maximum likelihood fitting yields slightly different standard deviations by considering both case and control observations concurrently. In addition, a misprint in [Pepe \(2003, 147\)](#) reports a coefficient of -4.91 for `x1` in the case population instead of -3.19 as reported by Stata.

◀

Inference on multiple classifiers using the [Alonzo and Pepe \(2002\)](#) estimating equation method is performed by fitting each model separately and bootstrapping to determine the dependence of the estimates. Using the maximum likelihood method, we also fit each model separately. We use `suest` (see [\[R\] suest](#)) to estimate the joint variance–covariance of our parameter estimates.

For our models, we can view the score equation for each model as an estimating equation. The estimate that solves the estimating equation (that makes the score 0) is asymptotically normal with a variance matrix that can be estimated using the inverse of the squared scores. By stacking the score equations of the separate models, we can estimate the variance matrix for all the parameter estimates by using this rule. This is an informal explanation; further details can be found in [\[R\] suest](#) and in the references [Rogers \(1993\)](#); [White \(1982 and 1996\)](#).

Now, we will examine a case with multiple classification variables.

▶ Example 11: Maximum likelihood ROC, multiple classifiers

We return to the neonatal audiology study with gender and age covariates ([Norton et al. 2000](#)). In [example 6](#), we fit a model with `male` and `currage` as control covariates, and `currage` as a ROC covariate for the classifier `y1` (DPOAE 65 at 2 kHz). We will refit this model, extending it to include the classifier `y2` (TEOAE 80 at 2 kHz).

```
. use https://www.stata-press.com/data/r18/nhhs
(Norton - neonatal audiology data)
. rocreg d y1 y2, probit ml ctrlcov(currage male) roccov(currage) cluster(id)
> nolog
```

Parametric ROC estimation Number of obs = 1,848

```
Covariate control      : linear regression
Control variables     : currage male
Control standardization: normal
ROC method           : parametric          Link: probit
Status               : d
Classifiers          : y1 y2
Classifier           : y1
Covariate control adjustment model:
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
casecov						
currage	.494211	.2126672	2.32	0.020	.077391	.9110311
_cons	-15.00403	8.238094	-1.82	0.069	-31.1504	1.142338
casesd						
_cons	8.49794	.4922792	17.26	0.000	7.533091	9.46279
ctrlcov						
currage	-.2032048	.0323803	-6.28	0.000	-.266669	-.1397406
male	.2369359	.2201391	1.08	0.282	-.1945288	.6684006
_cons	-1.23534	1.252775	-0.99	0.324	-3.690734	1.220055
ctrlsd						
_cons	7.749156	.0782225	99.07	0.000	7.595843	7.902469

```
Classifier : y2
Covariate control adjustment model:
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
casecov						
currage	.5729861	.2422662	2.37	0.018	.0981532	1.047819
_cons	-18.2597	9.384968	-1.95	0.052	-36.6539	.1344949
casesd						
_cons	9.723858	.5632985	17.26	0.000	8.619813	10.8279
ctrlcov						
currage	-.1694575	.0291922	-5.80	0.000	-.2266732	-.1122419
male	.7122587	.1993805	3.57	0.000	.3214802	1.103037
_cons	-5.651728	1.129452	-5.00	0.000	-7.865415	-3.438042
ctrlsd						
_cons	6.986167	.0705206	99.07	0.000	6.84795	7.124385

```
Status      : d
ROC Model   :
              (Std. err. adjusted for 2,741 clusters in id)
```

		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
y1							
	i_cons	-1.765608	1.105393	-1.60	0.110	-3.932138	.4009225
	currage	.0581566	.0290177	2.00	0.045	.0012828	.1150303
	s_cons	.9118864	.0586884	15.54	0.000	.7968593	1.026913
y2							
	i_cons	-1.877825	.905174	-2.07	0.038	-3.651933	-.1037167
	currage	.0589258	.0235849	2.50	0.012	.0127002	.1051514
	s_cons	.7184563	.0565517	12.70	0.000	.607617	.8292957

Both classifiers have similar results. The results for y1 show the same direction as the estimating equation results in [example 6](#). However, we can now reject the null hypothesis that the ROC `currage` coefficient is 0 at the 0.05 level.

In [example 6](#), we could not reject that the slope parameter `s_cons` was 1 and that the constant intercept or ROC coefficient for current age was 0. The resulting ROC curve implied a noninformative test using y1 as a classifier. This is not the case with our current results. As `currage` increases, we expect a steeper ROC curve and thus a more powerful test, for both classifiers y1 (DPOAE 65 at 2 kHz) and y2 (TEOAE 80 at 2 kHz).

In [example 10](#), the clustering of observations within infant `id` was adjusted in the individual fit of `nsnr`. In our current example, the adjustment for the clustering of observations within `id` is performed during concurrent estimation, as opposed to during the individual classifier fits (as in [example 10](#)). This adjustment, performed by `suest`, is still accurate.

◀

Now, we will fit constant probit models and compare `rocreg` with `rocfit` and `roccomp` with the `binormal` option. Our first applications of `rocfit` and `roccomp` are taken directly from [examples 8](#) and [9](#). The [Dorfman and Alf \(1969\)](#) algorithm that `rocfit` works with uses discrete classifiers or uses slicing to make a classifier discrete. So we are applying the maximum likelihood method of `rocreg` on discrete classification data here, where it expects continuous data. We expect to see some discrepancies, but we do not find great divergence in the estimates. After revisiting [examples 8](#) and [9](#), we will fit a probit model with a continuous classifier and no covariates using `rocreg`, and we will compare the results with those from `rocfit`.

▷ Example 12: Maximum likelihood ROC, marginal model

Using the [Hanley and McNeil \(1982\)](#) data, discussed in [example 1](#) and in [example 8](#), we fit a constant probit model of the classifier `rating` with true status `disease`. `rocreg` is invoked with the `ml` option and compared with `rocfit`.


```
. use https://www.stata-press.com/data/r18/hanley, clear
(Tomographic images)
```

```
. rocfit disease rating, nolog
```

```
Binormal model of disease on rating          Number of obs   =          109
Goodness-of-fit chi2(2) =                   0.21
Prob > chi2 =                               0.9006
Log likelihood = -123.64855
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
intercept	1.656782	0.310456	5.34	0.000	1.048300	2.265265
slope (*)	0.713002	0.215882	-1.33	0.184	0.289881	1.136123
/cut1	0.169768	0.165307	1.03	0.304	-0.154227	0.493764
/cut2	0.463215	0.167235	2.77	0.006	0.135441	0.790990
/cut3	0.766860	0.174808	4.39	0.000	0.424243	1.109477
/cut4	1.797938	0.299581	6.00	0.000	1.210770	2.385106

Index	Indices from binormal fit			
	Estimate	Std. err.	[95% conf. interval]	
ROC area	0.911331	0.029506	0.853501	0.969161
delta(m)	2.323671	0.502370	1.339044	3.308298
d(e)	1.934361	0.257187	1.430284	2.438438
d(a)	1.907771	0.259822	1.398530	2.417012

(*) z test for slope==1

```
. rocreg disease rating, probit ml nolog
```

```
Binormal model of disease on rating          Number of obs   =          109
GOF chi2(0) =                               .
Log likelihood = -123.64855                  Prob > chi2     =          .
```

```
Control standardization: normal
```

```
ROC method          : parametric          Link: probit
```

```
Status      : disease
```

```
Classifiers: rating
```

```
Classifier  : rating
```

```
Covariate control adjustment model:
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
casecov						
_cons	2.3357	.2334285	10.01	0.000	1.878188	2.793211
casesd						
_cons	1.117131	.1106124	10.10	0.000	.9003344	1.333927
ctrlcov						
_cons	2.017241	.1732589	11.64	0.000	1.67766	2.356823
ctrlsd						
_cons	1.319501	.1225125	10.77	0.000	1.07938	1.559621

```
Status : disease
ROC Model :
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
<code>rating</code>						
<i>i_cons</i>	2.090802	.2941411	7.11	0.000	1.514297	2.667308
<i>s_cons</i>	1.181151	.1603263	7.37	0.000	.8669177	1.495385
auc	.9116494	.0261658	34.84	0.000	.8603654	.9629333

We compare the estimates for these models:

	<code>rocfit</code>	<code>rocreg, ml</code>
slope	0.7130	1.1812
SE of slope	0.2159	0.1603
intercept	1.6568	2.0908
SE of intercept	0.3105	0.2941
AUC	0.9113	0.9116
SE of AUC	0.0295	0.0262

We find that both the intercept and the slope are estimated as higher with the maximum likelihood method under `rocreg` than with `rocfit`. The AUC (ROC area in `rocfit`) is close for both commands. We find that the standard errors of each of these estimates is slightly lower under `rocreg` than `rocfit` as well.

Both `rocfit` and `rocreg` suggest that the slope parameter of the ROC curve (`slope` in `rocfit` and `s_cons` in `rocreg`) is not significantly different from 1. Thus, we cannot reject that the classifier has the same variance in both case and control populations. There is, however, significant evidence that the intercepts (`i_cons` in `rocreg` and `intercept` in `rocfit`) differ from 0. Because of the positive direction of the intercept estimates, the ROC curve for `rating` as a classifier of `disease` suggests that `rating` provides an informative test. This is also suggested by the high AUC, which is significantly different from 0.5, that is, a flip of a coin.

◀

▶ Example 13: Maximum likelihood ROC, marginal model, multiple classifiers

We use the fictitious dataset generated from [Hanley and McNeil \(1983\)](#), which we previously used in [example 2](#) and in [example 9](#). To fit a probit model using `roccomp`, we specify the `binormal` option. We perform parametric, maximum likelihood ROC analysis using `rocreg`. We use `rocregplot` to plot the ROC curves created by `rocreg`.

```
. use https://www.stata-press.com/data/r18/ct2, clear
(Reconstruction of CT images)

. roccomp status mod1 mod2 mod3, summary binormal graph aspectratio(1)
>   plotlopts(connect(i) msymbol(o))
>   plot2opts(connect(i) msymbol(s))
>   plot3opts(connect(i) msymbol(t))
>   legend(label(1 "Modality 1") label(3 "Modality 2")
>           label(5 "Modality 3") label(2 "Modality 1 fit")
>           label(4 "Modality 2 fit") label(6 "Modality 3 fit")
>           order(1 3 5 2 4 6) pos(6)) title(roccomp) name(a) nodraw
Fitting binormal model for: mod1
Fitting binormal model for: mod2
Fitting binormal model for: mod3
```

	Obs	ROC area	Std. err.	[95% conf. interval]	
mod1	112	0.8945	0.0305	0.83482	0.95422
mod2	112	0.9382	0.0264	0.88647	0.99001
mod3	112	0.9376	0.0223	0.89382	0.98139

H0: area(mod1) = area(mod2) = area(mod3)
chi2(2) = 8.27 Prob>chi2 = 0.0160

```
. rocreg status mod1 mod2 mod3, probit ml nolog
Parametric ROC estimation                               Number of obs = 109
Control standardization: normal
ROC method      : parametric                          Link: probit
Status          : status
Classifiers:    mod1 mod2 mod3
Classifier       : mod1
Covariate control adjustment model:
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
casecov						
_cons	2.118135	.2165905	9.78	0.000	1.693626	2.542645
casesd						
_cons	1.166078	.1122059	10.39	0.000	.9461589	1.385998
ctrlcov						
_cons	2.344828	.1474147	15.91	0.000	2.0559	2.633755
ctrlsd						
_cons	1.122677	.1042379	10.77	0.000	.9183746	1.32698

Classifier : mod2
Covariate control adjustment model:

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
casecov _cons	2.659642	.2072731	12.83	0.000	2.253395	3.06589
casesd _cons	1.288468	.1239829	10.39	0.000	1.045466	1.53147
ctrlcov _cons	1.655172	.1105379	14.97	0.000	1.438522	1.871823
ctrlsd _cons	.8418313	.0781621	10.77	0.000	.6886365	.9950262

Classifier : mod3
Covariate control adjustment model:

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
casecov _cons	2.353768	.1973549	11.93	0.000	1.966959	2.740576
casesd _cons	1.143359	.1100198	10.39	0.000	.9277243	1.358994
ctrlcov _cons	2.275862	.1214094	18.75	0.000	2.037904	2.51382
ctrlsd _cons	.9246267	.0858494	10.77	0.000	.7563649	1.092888

Status : status
ROC Model :

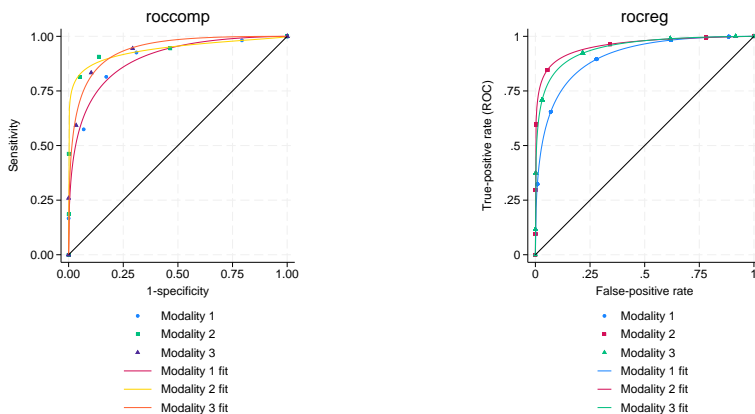
	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
mod1						
i_cons	1.81646	.3144804	5.78	0.000	1.20009	2.432831
s_cons	.9627801	.1364084	7.06	0.000	.6954245	1.230136
auc	.904657	.0343518	26.34	0.000	.8373287	.9719853
mod2						
i_cons	2.064189	.3267274	6.32	0.000	1.423815	2.704563
s_cons	.6533582	.1015043	6.44	0.000	.4544135	.8523029
auc	.9580104	.0219713	43.60	0.000	.9149473	1.001073
mod3						
i_cons	2.058643	.2890211	7.12	0.000	1.492172	2.625113
s_cons	.8086932	.1163628	6.95	0.000	.5806262	1.03676
auc	.9452805	.0236266	40.01	0.000	.8989732	.9915877

H0: All classifiers have equal AUC values

Ha: At least one classifier has a different AUC value

P-value: .0808808

```
. rocregplot, title(rocreg) nodraw name(b)
> plot1opts(msymbol(o)) plot2opts(msymbol(s)) plot3opts(msymbol(t))
. graph combine a b, xsize(5)
```



We compare the AUC estimates for these models:

	roccomp	rocreg, ml
mod1	0.8945	0.9047
mod2	0.9382	0.9580
mod3	0.9376	0.9453

Each classifier has a higher estimated AUC under `rocreg` than `roccomp`. Each curve appears to be raised and smoothed in the `rocreg` fit as compared with `roccomp`. They are different, but not drastically different. The inference on whether the curve areas are the same is similar to [example 9](#). We reject equality at the 0.10 level under `rocreg` and at the 0.05 level under `roccomp`.

Each intercept is significantly different from 0 at the 0.05 level and is estimated in a positive direction. Though all but classifier `mod2` has 1 in their slope confidence intervals, the high intercepts suggest steep ROC curves and powerful tests.

Also note that the false-positive and true-positive rate points are calculated empirically in the `roccomp` graph and parametrically in `rocreg`. In [example 9](#), the false-positive rates calculated by `rocreg` were calculated empirically, similar to `roccomp`. But in this example, the rates are calculated based on normal percentiles.



Now, we will generate an example to compare `rocfit` and `rocreg` under maximum likelihood estimation of a continuous classifier.

▷ Example 14: Maximum likelihood ROC, graphical comparison with `rocfit`

We generate 500 realizations of a population under threat of disease. One quarter of the population has the disease. A classifier `x` is measured, which has a control distribution of $N(1, 3)$ and a case distribution of $N(1 + 5, 2)$. We will invoke `rocreg` with the `ml` option on this generated data. We specify the `continuous()` option for `rocfit` and invoke it on the data as well. The `continuous()` option tells `rocfit` how many discrete slices to partition the data into before fitting.

For comparison of the two curves, we will use the `rocfit` postestimation command, `rocplot`; see [R] [rocfit postestimation](#). This command graphs the empirical false-positive and true-positive rates with an overlaid fit of the binormal curve estimated by `rocfit`. `rocplot` also supports an `addplot()` option. We use the saved variables from `rocreg` in this option to overlay a line plot of the `rocreg` fit.

```
. clear
. set seed 8675309
. set obs 500
Number of observations (_N) was 0, now 500.
. generate d = runiform() < .25
. quietly generate double epsilon = 3*invnormal(runiform()) if d == 0
. quietly replace epsilon = 2*invnormal(runiform()) if d == 1
. quietly generate double x = 1 + d*5 + epsilon
. rocreg d x, probit ml nolog
Parametric ROC estimation                               Number of obs = 112
Control standardization: normal
ROC method      : parametric                          Link: probit
Status         : d
Classifiers: x
Classifier      : x
Covariate control adjustment model:
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
casecov						
_cons	4.823931	.2305469	20.92	0.000	4.372067	5.275795
casesd						
_cons	1.926652	.1204158	16.00	0.000	1.690642	2.162663
ctrlcov						
_cons	1.14378	.155409	7.36	0.000	.8391841	1.448376
ctrlsd						
_cons	2.99742	.1098907	27.28	0.000	2.782038	3.212802

```
Status      : d
ROC Model   :
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
x						
i_cons	2.503789	.1969952	12.71	0.000	2.117686	2.889893
s_cons	1.555766	.1127296	13.80	0.000	1.33482	1.776712
auc	.912102	.0123921	73.60	0.000	.8878139	.9363902

```
. rocfit d x, continuous(10) nolog
```

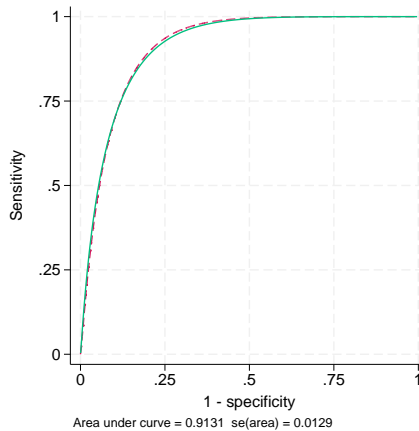
```
Binormal model of d on x                               Number of obs   =           500
Goodness-of-fit chi2(7) =                             1.33
Prob > chi2       =                                 0.9877
Log likelihood    =                               -914.15521
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
intercept	2.647297	0.277012	9.56	0.000	2.104362	3.190231
slope (*)	1.670103	0.195433	3.43	0.001	1.287062	2.053145
/cut1	-2.079091	0.153221	-13.57	0.000	-2.379398	-1.778783
/cut2	-1.383360	0.093448	-14.80	0.000	-1.566515	-1.200205
/cut3	-0.905227	0.075606	-11.97	0.000	-1.053413	-0.757041
/cut4	-0.252654	0.065679	-3.85	0.000	-0.381382	-0.123925
/cut5	0.310051	0.065913	4.70	0.000	0.180863	0.439239
/cut6	0.915048	0.072958	12.54	0.000	0.772054	1.058043
/cut7	1.512188	0.092153	16.41	0.000	1.331570	1.692805
/cut8	2.095878	0.136662	15.34	0.000	1.828026	2.363731
/cut9	2.516563	0.181939	13.83	0.000	2.159970	2.873156

Index	Indices from binormal fit			
	Estimate	Std. err.	[95% conf. interval]	
ROC area	0.913079	0.012942	0.887713	0.938445
delta(m)	1.585110	0.107531	1.374352	1.795868
d(e)	1.982917	0.121777	1.744239	2.221596
d(a)	1.923275	0.115671	1.696565	2.149985

(*) z test for slope==1

```
. rocplot, plotopts(msymbol(i)) lineopts(lpattern(dash))
> norefline addplot(line _roc_x _fpr_x, sort(_fpr_x _roc_x))
> lpattern(solid) aspectratio(1) legend(off)
```



We find that the curves are close. As before, the rocfit estimates are lower for the slope and intercept than under rocreg. The AUC estimates are close. Though the slope confidence interval contains 1, a high ROC intercept suggests a steep ROC curve and thus a powerful test.

Stored results

Nonparametric `rocreg` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_strata)</code>	number of covariate strata
<code>e(N_clust)</code>	number of clusters
<code>e(level)</code>	confidence level for bootstrap CIs
<code>e(rank)</code>	rank of <code>e(V)</code>

Macros

<code>e(cmd)</code>	<code>rocreg</code>
<code>e(cmdline)</code>	command as typed
<code>e(classvars)</code>	classification variable list
<code>e(refvar)</code>	status variable, reference variable
<code>e(ctrlmodel)</code>	covariate-adjustment specification
<code>e(ctrlcov)</code>	covariate-adjustment variables
<code>e(pvc)</code>	percentile value calculation method
<code>e(title)</code>	title in estimation output
<code>e(tiecorrected)</code>	<code>tiecorrected</code> , if specified
<code>e(nobootstrap)</code>	<code>nobootstrap</code> , if specified
<code>e(rngstate)</code>	random-number state used in bootstrap, if bootstrap was performed
<code>e(breps)</code>	number of bootstrap resamples, if bootstrap performed
<code>e(bootcc)</code>	<code>bootcc</code> , if specified
<code>e(nobstrata)</code>	<code>nobstrata</code> , if specified
<code>e(clustvar)</code>	name of cluster variable
<code>e(exp#)</code>	expression for the #th statistic
<code>e(roc)</code>	false-positive rates where ROC was estimated
<code>e(invroc)</code>	ROC values where false-positive rates were estimated
<code>e(pauc)</code>	false-positive rates where pAUC was estimated
<code>e(auc)</code>	indicates that AUC was calculated
<code>e(vce)</code>	<code>bootstrap</code>
<code>e(properties)</code>	<code>b V</code> (or <code>b</code> if bootstrap not performed)

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(b_bs)</code>	bootstrap estimates
<code>e(reps)</code>	number of nonmissing results
<code>e(bias)</code>	estimated biases
<code>e(se)</code>	estimated standard errors
<code>e(z0)</code>	median biases
<code>e(ci_normal)</code>	normal-approximation confidence intervals
<code>e(ci_percentile)</code>	percentile confidence intervals
<code>e(ci_bc)</code>	bias-corrected confidence intervals

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	----------------------------------------------------------------------------------------------------------------------------

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

Parametric, bootstrap `rocreg` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_strata)</code>	number of covariate strata
<code>e(N_clust)</code>	number of clusters
<code>e(level)</code>	confidence level for bootstrap CIs
<code>e(rank)</code>	rank of <code>e(V)</code>

Macros

<code>e(cmd)</code>	<code>rocreg</code>
<code>e(cmdline)</code>	command as typed
<code>e(classvars)</code>	classification variable list
<code>e(refvar)</code>	status variable, reference variable
<code>e(ctrlmodel)</code>	covariate-adjustment specification
<code>e(ctrlcov)</code>	covariate-adjustment variables
<code>e(pvc)</code>	percentile value calculation method
<code>e(title)</code>	title in estimation output
<code>e(tiecorrected)</code>	<code>tiecorrected</code> , if specified
<code>e(probit)</code>	<code>probit</code> , if specified
<code>e(roccov)</code>	ROC covariates
<code>e(fprpts)</code>	number of points used as false-positive rate fit points
<code>e(ctrlfprall)</code>	indicates whether all observed false-positive rates were used as fit points
<code>e(nobootstrap)</code>	<code>nobootstrap</code> , if specified
<code>e(rngstate)</code>	random-number state used in bootstrap
<code>e(breps)</code>	number of bootstrap resamples
<code>e(bootcc)</code>	<code>bootcc</code> , if specified
<code>e(nobstrata)</code>	<code>nobstrata</code> , if specified
<code>e(clustvar)</code>	name of cluster variable
<code>e(exp#)</code>	expression for the #th statistic
<code>e(vce)</code>	<code>bootstrap</code>
<code>e(properties)</code>	<code>b V</code> (or <code>b</code> if <code>nobootstrap</code> is specified)
<code>e(predict)</code>	program used to implement <code>predict</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(b_bs)</code>	bootstrap estimates
<code>e(reps)</code>	number of nonmissing results
<code>e(bias)</code>	estimated biases
<code>e(se)</code>	estimated standard errors
<code>e(z0)</code>	median biases
<code>e(ci_normal)</code>	normal-approximation confidence intervals
<code>e(ci_percentile)</code>	percentile confidence intervals
<code>e(ci_bc)</code>	bias-corrected confidence intervals

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	----------------------------------------------------------------------------------------------------------------------------

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Parametric, maximum likelihood `rocreg` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(rank)</code>	rank of <code>e(V)</code>

Macros

<code>e(cmd)</code>	<code>rocreg</code>
<code>e(cmdline)</code>	command as typed
<code>e(classvars)</code>	classification variable list
<code>e(refvar)</code>	status variable
<code>e(ctrlmodel)</code>	<code>linear</code>
<code>e(ctrlcov)</code>	control population covariates
<code>e(roccov)</code>	ROC covariates
<code>e(probit)</code>	<code>probit</code> , if specified
<code>e(pvc)</code>	<code>normal</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(vce)</code>	<code>cluster</code> if clustering used
<code>e(vcetype)</code>	<code>robust</code> if multiple classifiers or clustering used
<code>e(ml)</code>	<code>ml</code> , if specified
<code>e(predict)</code>	program used to implement <code>predict</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	-----------------------------------------------------------------------------------------------------------------------

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Assume that we applied a diagnostic test to each of N_0 control and N_1 case subjects. Further assume that the higher the outcome value of the diagnostic test, the higher the risk of the subject being abnormal. Let y_{1i} , $i = 1, 2, \dots, N_1$, and y_{0j} , $j = 1, 2, \dots, N_0$, be the values of the diagnostic test for the case and control subjects, respectively. The true status variable D identifies an observation as case $D = 1$ or control $D = 0$. The CDF of the classifier Y is F . Conditional on D , we write the CDF as F_D .

Methods and formulas are presented under the following headings:

- ROC statistics*
- Covariate-adjusted ROC curves*
- Parametric ROC curves: Estimating equations*
- Parametric ROC curves: Maximum likelihood*

ROC statistics

We obtain these definitions and their estimates from [Pepe \(2003\)](#) and [Pepe, Longton, and Janes \(2009\)](#). The false-positive and true-positive rates at cutoff c are defined as

$$\text{FPR}(y) = P(Y \geq y | D = 0)$$

$$\text{TPR}(y) = P(Y \geq y | D = 1)$$

The true-positive rate, or ROC value at false-positive rate u , is given by

$$\text{ROC}(u) = P(1 - F_0(Y) \leq u | D = 1)$$

When Y is continuous, the false-positive rate can be written as

$$\text{FPR}(y) = 1 - F_0(y)$$

The empirical CDF for the sample z_1, \dots, z_n is given by

$$\widehat{F}(z) = \sum_{i=1}^n \frac{I(z < z_i)}{n}$$

The empirical estimates $\widehat{\text{FPR}}$ and $\widehat{\text{ROC}}$ both use this empirical CDF estimator.

The area under the ROC curve is defined as

$$\text{AUC} = \int_0^1 \text{ROC}(u) d_u$$

The partial area under the ROC curve for false-positive rate a is defined as

$$\text{pAUC}(a) = \int_0^a \text{ROC}(u) d_u$$

The nonparametric estimate for the AUC is given by

$$\widehat{\text{AUC}} = \sum_{i=1}^{N_1} \frac{1 - \widehat{\text{FPR}}(y_{1i})}{N_1}$$

The nonparametric estimate of pAUC is given by

$$\widehat{\text{pAUC}}(a) = \sum_{i=1}^{N_1} \frac{\max\{1 - \widehat{\text{FPR}}(y_{1i}) - (1 - a), 0\}}{N_1}$$

For discrete classifiers, a correction term is subtracted from the false-positive rate estimate so that the $\widehat{\text{AUC}}$ and $\widehat{\text{pAUC}}$ estimates correspond with a trapezoidal approximation to the area of the ROC curve.

$$\text{FPR}^c(y) = 1 - \widehat{F}_0(y) - \frac{1}{2} \sum_{j=1}^{N_0} \frac{I(y = y_{0j})}{N_0}$$

In the nonparametric estimation of the ROC curve, all inference is performed using the `bootstrap` command (see [\[R\] bootstrap](#)). `rocreg` also allows users to calculate the ROC curve and related statistics by assuming a normal control distribution. So these formulas are updated by replacing F_0 by Φ (with adjustment of the marginal mean and variance of the control distribution).

Covariate-adjusted ROC curves

Suppose we observe covariate vector Z in addition to the classifier Y . Let $Z_{1i}, i = 1, 2, \dots, N_1$, and $Z_{0j}, j = 1, 2, \dots, N_0$, be the values of the covariates for the case and control subjects, respectively.

The covariate-adjusted ROC curve is defined by [Janes and Pepe \(2009\)](#) as

$$\text{AROC}(t) = E \{ \text{ROC}(t | Z_0) \}$$

It is calculated by replacing the marginal control CDF estimate, \hat{F}_0 , with the conditional control CDF estimate, \hat{F}_{0Z} . If we used a normal control CDF, then we would replace the marginal control mean and variance with the conditional control mean and variance. The formulas of the previous section can be updated for covariate-adjustment by making this substitution of the conditional CDF for the marginal CDF in the false-positive rate calculation.

Because the calculation of the ROC value is now performed based on the conditionally calculated false-positive rate, no further conditioning is made in its calculation under nonparametric estimation.

`rocreg` supports covariate adjustment with stratification and linear regression. Under stratification, separate parameters are estimated for the control distribution at each level of the covariates. Under linear regression, the classifier is regressed on the covariates over the control distribution, and the resulting coefficients serve as parameters for \hat{F}_{0Z} .

Parametric ROC curves: Estimating equations

Under nonparametric estimation of the ROC curve with covariate adjustment, no further conditioning occurs in the ROC curve calculation beyond the use of covariate-adjusted false-positive rates as inputs.

Under parametric estimation of the ROC curve, we can relax this restriction. We model the ROC curve as a cumulative distribution function g (standard normal Φ) invoked with input of a linear polynomial in the corresponding quantile function (here Φ^{-1}) invoked on the false-positive rate u . The constant intercept of the polynomial may depend on covariates; the slope term α (quantile coefficient) may not.

$$\text{ROC}(u) = g\{\mathbf{x}'\beta + \alpha g^{-1}(u)\}$$

[Pepe \(2003\)](#) notes that having a binormal ROC ($g = \Phi$) is equivalent to specifying that some monotone transformation of the data exists to make the case and control classifiers normally distributed. This specification applies to the marginal case and control.

Under weak assumptions about the control distribution of the classifier, we can fit this model by using estimating equations ([Alonzo and Pepe 2002](#)). The method can be used without covariate effects in the second stage, assuming a parametric model for the single ROC curve. Using the [Alonzo and Pepe \(2002\)](#) method, the covariate-adjusted ROC curve may be fit parametrically. The marginal ROC curve, involving no covariates in either stage of estimation, can be fit parametrically as well. In addition to the [Alonzo and Pepe \(2002\)](#) explanation, further details are given in [Pepe, Longton, and Janes \(2009\)](#); [Janes, Longton, and Pepe \(2009\)](#); [Pepe \(2003\)](#); and [Janes and Pepe \(2009\)](#).

The algorithm can be described as follows:

1. Estimate the false-positive rates of the classifier \mathbf{fpr} . These may be computed in any fashion outlined so far: covariate-adjusted, empirically, etc.
2. Determine a set of n_p false-positive rates to use as fitting points f_1, \dots, f_{n_p} . These may be an equispaced grid on $(0, 1)$ or the set of observed false-positive rates from part 1.

3. Expand the case observation portion of the data to include a subobservation for each fitting point. So there are now $N_1(n_p - 1)$ additional observations in the data.
4. Generate a new dummy variable `u`. For subobservation j , $u = I(\text{fpr} \leq f_j)$.
5. Generate a new variable `quant` containing the quantiles of the false-positive rate fitting points. For subobservation j , $\text{quant} = g^{-1}(f_j)$.
6. Perform a binary regression (probit, $g = \Phi$) of `fpr` on the covariates `x` and quantile variable `quant`.

The coefficients of part 6 are the coefficients of the ROC model. The coefficients of the covariates coincide naturally with estimates of β , and the α parameter is estimated by the coefficient on `quant`. Because the method is so general and makes few distributional assumptions, bootstrapping must be performed for inference. If multiple classifiers are to be fit, the algorithm is performed separately for each in each bootstrap, and the bootstrap is used to estimate covariances.

We mentioned earlier that in parametric estimation, the AUC was the only summary parameter that could be estimated initially. This is true when we fit the marginal probit model because there are no covariates in part 6 of the algorithm.

To calculate the AUC statistic under a marginal probit model, we use the formula

$$\text{AUC} = \Phi\left(\frac{\beta_0}{\sqrt{1 + \alpha^2}}\right)$$

Alternatively, the AUC for the probit model can be calculated as `pAUC(1)` in postestimation. Under both models, bootstrapping is performed for inference on the AUC.

Parametric ROC curves: Maximum likelihood

`rocreg` supports another form of parametric ROC estimation: maximum likelihood with a normally distributed classifier. This method assumes that the classifier is a normal linear model on certain covariates, and the covariate effect and variance of the classifier may change between the case and control populations. The model is defined in [Pepe \(2003, 145\)](#).

$$y = \mathbf{z}'\beta_0 + D\mathbf{x}'\beta_1 + \sigma(D)\epsilon$$

Our error term, ϵ , is a standard normal random variable. The variable D is our true status variable, being 1 for the case population observations and 0 for the control population observations. The variance function σ is defined as

$$\sigma(D) = \sigma_0(D = 0) + \sigma_1(D = 1)$$

This provides two variance parameters in the model and does not depend on covariate values.

Under this model, the ROC curve is easily derived to be

$$\text{ROC}(u) = \Phi\left[\frac{1}{\sigma_1}\{\mathbf{x}'\beta_1 + \sigma_0\Phi^{-1}(u)\}\right]$$

We reparameterize the model, creating the parameters $\beta_i = \sigma_1^{-1}\beta_{1i}$ and $\alpha = \sigma_1^{-1}\sigma_0$. We refer to β_0 as the constant intercept, `i_cons`. The parameter α is referred to as the constant slope, `s_cons`.

$$\text{ROC}(u) = \Phi\{\mathbf{x}'\beta + \alpha\Phi^{-1}(u)\}$$

The original model defining the classifier y leads to the following single observation likelihoods for $D = 0$ and $D = 1$:

$$L(\beta_0, \beta_1, \sigma_1, \sigma_0, |D = 0, y, \mathbf{z}, \mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp \frac{-(y - \mathbf{z}'\beta_0)^2}{2\sigma_0^2}$$

$$L(\beta_0, \beta_1, \sigma_1, \sigma_0, |D = 1, y, \mathbf{z}, \mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp \frac{-(y - \mathbf{z}'\beta_0 - \mathbf{x}'\beta_1)^2}{2\sigma_1^2}$$

These can be combined to yield the observation-level log likelihood:

$$\begin{aligned} \ln L(\beta_0, \beta_1, \sigma_1, \sigma_0, |D, y, \mathbf{z}, \mathbf{x}) = & -\frac{\ln 2\pi}{2} \\ & - I(D = 0) \left\{ \ln \sigma_0 + \frac{(y - \mathbf{z}'\beta_0)^2}{2\sigma_0^2} \right\} \\ & - I(D = 1) \left\{ \ln \sigma_1 + \frac{(y - \mathbf{z}'\beta_0 - \mathbf{x}'\beta_1)^2}{2\sigma_1^2} \right\} \end{aligned}$$

When there are multiple classifiers, each classifier is fit separately with maximum likelihood. Then, the results are combined by stacking the scores and using the sandwich variance estimator. For more information, see [R] **suest** and the references White (1982); Rogers (1993); and White (1996).

Acknowledgments

We thank Margaret S. Pepe, Holly Janes, and Gary Longton (retired) of the Fred Hutchinson Cancer Research Center for providing the inspiration for the **rocreg** command and for illuminating many useful datasets for its documentation.

References

- Alonzo, T. A., and M. S. Pepe. 2002. Distribution-free ROC analysis using binary regression techniques. *Biostatistics* 3: 421–432. <https://doi.org/10.1093/biostatistics/3.3.421>.
- Claves, M. A. 2002a. Comparative assessment of three common algorithms for estimating the variance of the area under the nonparametric receiver operating characteristic curve. *Stata Journal* 2: 280–289.
- . 2002b. From the help desk: Comparing areas under receiver operating characteristic curves from two or more probit or logit models. *Stata Journal* 2: 301–313.
- Cook, N. R. 2007. Use and misuse of the receiver operating characteristic curve in risk prediction. *Circulation* 115: 928–935. <https://doi.org/10.1161/CIRCULATIONAHA.106.672402>.
- DeLong, E. R., D. M. DeLong, and D. L. Clarke-Pearson. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics* 44: 837–845. <https://doi.org/10.2307/2531595>.
- Dodd, L. E., and M. S. Pepe. 2003. Partial AUC estimation and regression. *Biometrics* 59: 614–623. <https://doi.org/10.1111/1541-0420.00071>.
- Dorfman, D. D., and E. Alf, Jr. 1969. Maximum-likelihood estimation of parameters of signal-detection theory and determination of confidence intervals—rating-method data. *Journal of Mathematical Psychology* 6: 487–496. [https://doi.org/10.1016/0022-2496\(69\)90019-4](https://doi.org/10.1016/0022-2496(69)90019-4).

- Hanley, J. A., and K. O. Hajian-Tilaki. 1997. Sampling variability of nonparametric estimates of the areas under receiver operating characteristic curves: An update. *Academic Radiology* 4: 49–58. [https://doi.org/10.1016/S1076-6332\(97\)80161-4](https://doi.org/10.1016/S1076-6332(97)80161-4).
- Hanley, J. A., and B. J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143: 29–36. <https://doi.org/10.1148/radiology.143.1.7063747>.
- . 1983. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* 148: 839–843. <https://doi.org/10.1148/radiology.148.3.6878708>.
- Janes, H., G. M. Longton, and M. S. Pepe. 2009. Accommodating covariates in receiver operating characteristic analysis. *Stata Journal* 9: 17–39.
- Janes, H., and M. S. Pepe. 2009. Adjusting for covariate effects on classification accuracy using the covariate-adjusted receiver operating characteristic curve. *Biometrika* 96: 371–382. <https://doi.org/10.1093/biomet/asp002>.
- Lora, D., I. Contador, J. F. Pérez-Regadera, and A. Gómez de la Cámara. 2016. Features of the area under the receiver operating characteristic (ROC) curve. A good practice. *Stata Journal* 16: 185–196.
- McClish, D. K. 1989. Analyzing a portion of the ROC curve. *Medical Decision Making* 9: 190–195. <https://doi.org/10.1177/0272989X8900900307>.
- Mooney, C. Z., and R. D. Duval. 1993. *Bootstrapping: A Nonparametric Approach to Statistical Inference*. Newbury Park, CA: Sage.
- Norton, S. J., M. P. Gorga, J. E. Widén, R. C. Folsom, Y. Sininger, B. Cone-Wesson, B. R. Vohr, K. Mascher, and K. Fletcher. 2000. Identification of neonatal hearing impairment: Evaluation of transient evoked otoacoustic emission, distortion product otoacoustic emission, and auditory brain stem response test performance. *Ear and Hearing* 21: 508–528. <https://doi.org/10.1097/00003446-200010000-00013>.
- Pepe, M. S. 1998. Three approaches to regression analysis of receiver operating characteristic curves for continuous test results. *Biometrics* 54: 124–135. <https://doi.org/10.2307/2534001>.
- . 2000. Receiver operating characteristic methodology. *Journal of the American Statistical Association* 95: 308–311. <https://doi.org/10.2307/2669554>.
- . 2003. *The Statistical Evaluation of Medical Tests for Classification and Prediction*. New York: Oxford University Press.
- Pepe, M. S., and T. Cai. 2004. The analysis of placement values for evaluating discriminatory measures. *Biometrics* 60: 528–535. <https://doi.org/10.1111/j.0006-341X.2004.00200.x>.
- Pepe, M. S., G. M. Longton, and H. Janes. 2009. Estimation and comparison of receiver operating characteristic curves. *Stata Journal* 9: 1–16.
- Rogers, W. H. 1993. sg16.4: Comparison of nbreg and glm for negative binomial. *Stata Technical Bulletin* 16: 7. Reprinted in *Stata Technical Bulletin Reprints*, vol. 3, pp. 82–84. College Station, TX: Stata Press.
- Stover, L., M. P. Gorga, S. T. Neely, and D. Montoya. 1996. Toward optimizing the clinical utility of distortion product otoacoustic emission measurements. *Journal of the Acoustical Society of America* 100: 956–967. <https://doi.org/10.1121/1.416207>.
- Thompson, M. L., and W. Zucchini. 1989. On the statistical analysis of ROC curves. *Statistics in Medicine* 8: 1277–1290. <https://doi.org/10.1002/sim.4780081011>.
- White, H. L., Jr. 1982. Maximum likelihood estimation of misspecified models. *Econometrica* 50: 1–25. <https://doi.org/10.2307/1912526>.
- . 1996. *Estimation, Inference and Specification Analysis*. Cambridge: Cambridge University Press.
- Wieand, S., M. H. Gail, B. R. James, and K. L. James. 1989. A family of nonparametric statistics for comparing diagnostic markers with paired or unpaired data. *Biometrika* 76: 585–592. <https://doi.org/10.2307/2336123>.

Also see

[R] **rocreg postestimation** — Postestimation tools for rocreg

[R] **rocregplot** — Plot marginal and covariate-specific ROC curves after rocreg

[R] **rocfitt** — Parametric ROC models

[R] **roc** — Receiver operating characteristic (ROC) analysis

[U] **20 Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

