Title

rocreg postestimation — Postestimation tools for rocreg

Postestimation commands

The following commands are of special interest after `rocreg`:

Command	Description
`estat nproc`	nonparametric ROC curve estimation, keeping fit information from `rocreg`
`rocregplot`	plot marginal and covariate-specific ROC curves

The following standard postestimation commands are also available:

Command	Description
`estimates`	cataloging estimation results
`etable`	table of estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	roc curve predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

Use of predict after fitting a parametric model with rocreg allows calculation of all the ROC curve summary statistics for covariate-specific ROC curves. The ROC values for given false-positive rates, false-positive rate for given ROC values, area under the ROC curve (AUC), and partial areas under the ROC curve (pAUC) for a given false-positive rate can all be calculated.

Menu for predict

Statistics > Postestimation

Syntax for predict

predict [*type*] *newvar* [*if*] [*in*] [, *statistic options*]

statistic	Description
Main	
at(*varname*)	input variable for statistic
auc	total area under the ROC curve; the default
roc	ROC values for given false-positive rates in at()
invroc	false-positive rate for given ROC values in at()
pauc	partial area under the ROC curve up to each false-positive rate in at()
classvar(*varname*)	statistic for given classifier

options	Description
Options	
intpts(*#*)	points in numeric integration of pAUC calculation
se(*newvar*)	predict standard errors
ci(*stubname*)	produce confidence intervals, stored as variables with prefix *stubname* and suffixes _l and _u
level(*#*)	set confidence level; default is level(95)
*bfile(*filename*, ...)	load dataset containing bootstrap replicates from rocreg
*btype(n | p | bc)	produce normal-based (n), percentile (p), or bias-corrected (bc) confidence intervals; default is btype(n)

*bfile() and btype() are only allowed with parametric analysis using bootstrap inference.

Options for predict

Main

at(*varname*) records the variable to be used as input for the above predictions.

auc predicts the total area under the ROC curve defined by the covariate values in the data. This is the default statistic.

roc predicts the ROC values for false-positive rates stored in *varname* specified in at().

invroc predicts the false-positive rates for given ROC values stored in *varname* specified in at().

pauc predicts the partial area under the ROC curve up to each false-positive rate stored in *varname* specified in at().

classvar(*varname*) performs the prediction for the specified classifier.

Options

intpts(#) specifies that # points be used in the pAUC calculation.

se(*newvar*) specifies that standard errors be produced and stored in *newvar*.

ci(*stubname*) requests that confidence intervals be produced and the lower and upper bounds be stored in *stubname*_l and *stubname*_u, respectively.

level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is level(95) or as set by set level; see **[U] 20.8 Specifying the width of confidence intervals**.

bfile(*filename*, ...) uses bootstrap replicates of parameters from rocreg stored in *filename* to estimate standard errors and confidence intervals of predictions.

btype(n | p | bc) specifies whether to produce normal-based (n), percentile (p), or bias-corrected (bc) confidence intervals. The default is btype(n).

estat

Description for estat

`estat nproc` allows calculation of all the ROC curve summary statistics for covariate-specific ROC curves, as well as for a nonparametric ROC estimation. Under nonparametric estimation, a single ROC curve is estimated by `rocreg`. Covariates can affect this estimation, but there are no separate covariate-specific ROC curves. Thus, the input arguments for `estat nproc` are taken in the command line rather than from the data as variable values.

Menu for estat

Statistics > Postestimation

Syntax for estat nproc

`estat nproc` [, *estat_nproc_options*]

estat_nproc_options	Description
Main	
`auc`	estimate total area under the ROC curve
`roc(`*numlist*`)`	estimate ROC values for given false-positive rates
`invroc(`*numlist*`)`	estimate false-positive rate for given ROC values
`pauc(`*numlist*`)`	estimate partial area under the ROC curve up to each false-positive rate

At least one option must be specified.

Options for estat nproc

Main

`auc` estimates the total area under the ROC curve.

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

`invroc(`*numlist*`)` estimates the false-positive rate for each of the ROC values in *numlist*. The values in *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 in *numlist* must be in the range (0,1].

Remarks and examples

stata.com

Remarks are presented under the following headings:

Using predict after rocreg

predict, after parametric rocreg, predicts the AUC, the ROC value, the false-positive rate (invROC), or the pAUC value. The default is auc.

We begin by estimating the area under the ROC curve for each of the three age-specific ROC curves in example 1 of [R] **rocregplot**: 30, 40, and 50 months.

▷ Example 1: Parametric ROC, AUC

In example 6 of [R] **rocreg**, a probit ROC model was fit to audiology test data from Norton et al. (2000). The estimating equations method of Alonzo and Pepe (2002) was used to fit the model. Gender and age were covariates that affected the control distribution of the classifier y1 (DPOAE 65 at 2 kHz). Age was a ROC covariate for the model, so we fit separate ROC curves at each age.

Following Janes, Longton, and Pepe (2009), we drew the ROC curves for ages 30, 40, and 50 months in example 1 of [R] **rocregplot**. Now, we use predict to estimate the AUC for the ROC curve at each of those ages.

The bootstrap dataset saved by rocreg in example 6 of [R] **rocreg**, nnhs2y1.dta, is used in the bfile() option.

We will store the AUC prediction in the new variable predAUC. We specify the se() option with the new variable name seAUC to produce an estimate of the prediction's standard error. By specifying the stubname cin in ci(), we tell predict to create normal-based confidence intervals (the default) as new variables cin_l and cin_u.

```
. use https://www.stata-press.com/data/r18/nnhs
(Norton - neonatal audiology data)
. rocreg d y1, probit ctrlcov(currage male) ctrlmodel(linear) roccov(currage)
> cluster(id) bseed(56930) bsave(nnhs2y1)
  (output omitted )
. set obs 5061
Number of observations (_N) was 5,058, now 5,061.
. quietly replace currage = 30 in 5059
. quietly replace currage = 40 in 5060
. quietly replace currage = 50 in 5061
. predict predAUC in 5059/5061, auc se(seAUC) ci(cin) bfile(nnhs2y1)
. list currage predAUC seAUC cin* in 5059/5061
```

	currage	predAUC	seAUC	cin_l	cin_u
5059.	30	.5209999	.076013	.3720171	.6699827
5060.	40	.6479176	.0284218	.592212	.7036232
5061.	50	.7601378	.0794346	.6044489	.9158267

As expected, we find the AUC to increase with age.

Essentially, we have a stored bootstrap sample of ROC covariate coefficient estimates in nnhs2y1.dta. We calculate the AUC using each set of coefficient estimates, resulting in a sample of AUC estimates. Then, the bootstrap standard error and confidence intervals are calculated based on this AUC sample. Further details of the computation of the standard error and percentile confidence intervals can be found in *Methods and formulas* and in [R] **bootstrap**.

We can also produce percentile or bias-corrected confidence intervals by specifying btype(p) or btype(bc), which we now demonstrate.

```
. drop *AUC*
. predict predAUC in 5059/5061, auc se(seAUC) ci(cip) bfile(nnhs2y1) btype(p)
. list currage predAUC cip* in 5059/5061
```

	currage	predAUC	cip_l	cip_u
5059.	30	.5209999	.3625608	.6615624
5060.	40	.6479176	.5874361	.702619
5061.	50	.7601378	.5836248	.8910214

```
. drop *AUC*
. predict predAUC in 5059/5061, auc se(seAUC) ci(cibc) bfile(nnhs2y1) btype(bc)
. list currage predAUC cibc* in 5059/5061
```

	currage	predAUC	cibc_l	cibc_u
5059.	30	.5209999	.3733315	.6669934
5060.	40	.6479176	.5901812	.7038136
5061.	50	.7601378	.5738162	.8882028

◁

predict can also estimate the ROC value and the false-positive rate (invROC).

▷ Example 2: Parametric ROC, invROC, and ROC value

In example 7 of [R] **rocreg**, we fit the ROC curve for status variable hearing loss (d) and classifier negative signal-to-noise ratio nsnr with ROC covariates frequency (xf), intensity (xl), and hearing loss severity (xd). The data were obtained from Stover et al. (1996). The model fit was probit with bootstrap resampling. We saved 50 bootstrap replications in the dataset nsnrf.dta.

The covariate value combinations xf = 10.01, xl = 5.5, and xd = .5, and xf = 10.01, xl = 6.5, and xd = 4 are of interest. In example 3 of [R] **rocregplot**, we estimated the ROC values for false-positive rates 0.2 and 0.7 and the false-positive rate for a ROC value of 0.5 by using rocregplot. We will use predict to replicate the estimation.

We begin by appending observations with our desired covariate combinations to the data. We also create two new variables: rocinp, which contains the ROC values for which we wish to predict the corresponding invROC values, and invrocinp, which contains the invROC values corresponding to the ROC values we wish to predict.

```
. clear
. input xf xl xd rocinp invrocinp
              xf          xl          xd      rocinp   invrocinp
  1. 10.01 5.5 .5 .2 .
  2. 10.01 6.5 4  .2 .
  3. 10.01 5.5 .5 .7 .5
  4. 10.01 6.5 4  .7 .5
  5. end
. save newdata
file newdata.dta saved
. use https://www.stata-press.com/data/r18/dp
(Stover - DPOAE test data)
. quietly rocreg d nsnr, ctrlcov(xf xl) roccov(xf xl xd) probit cluster(id)
> nobstrata ctrlfprall bseed(156385) breps(50) ctrlmodel(strata) bsave(nsnrf)
. append using newdata
. list xf xl xd invrocinp rocinp in 1849/1852
```

	xf	xl	xd	invroc~p	rocinp
1849.	10.01	5.5	.5	.	.2
1850.	10.01	6.5	4	.	.2
1851.	10.01	5.5	.5	.5	.7
1852.	10.01	6.5	4	.5	.7

Now, we will use `predict` to estimate the ROC value for the false-positive rates stored in `rocinp`. We specify the `roc` option, and we specify `rocinp` in the `at()` option. The other options, `se()` and `ci()`, are used to obtain standard errors and confidence intervals, respectively. The dataset of bootstrap samples, `nsnrf.dta`, is specified in `bfile()`. After prediction, we list the point estimates and standard errors.

```
. predict rocit in 1849/1852, roc at(rocinp) se(seroc) ci(cin) bfile(nsnrf)
. list xf xl xd rocinp rocit seroc if !missing(rocit)
```

	xf	xl	xd	rocinp	rocit	seroc
1849.	10.01	5.5	.5	.2	.7652956	.0624187
1850.	10.01	6.5	4	.2	.9672505	.0162785
1851.	10.01	5.5	.5	.7	.9835816	.0133583
1852.	10.01	6.5	4	.7	.999428	.0007784

These results match example 3 of [R] **rocregplot**. We list the confidence intervals next. These also conform to the `rocregplot` results from example 3 in [R] **rocregplot**. We begin with the confidence intervals for ROC under the covariate values `xf=10.01`, `xl=5.5`, and `xd=.5`.

```
. list xf xl xd rocinp rocit cin* if inlist(_n, 1849, 1851)
```

	xf	xl	xd	rocinp	rocit	cin_l	cin_u
1849.	10.01	5.5	.5	.2	.7652956	.6429572	.887634
1851.	10.01	5.5	.5	.7	.9835816	.9573998	1.009763

Now, we list the ROC confidence intervals under the covariate values `xf=10.01`, `xl=6.5`, and `xd=4`.

```
. list xf xl xd rocinp rocit cin* if inlist(_n, 1850, 1852)
```

	xf	xl	xd	rocinp	rocit	cin_l	cin_u
1850.	10.01	6.5	4	.2	.9672505	.9353452	.9991558
1852.	10.01	6.5	4	.7	.999428	.9979024	1.000954

Now, we will predict the false-positive rate for a ROC value by specifying the `invroc` option. We pass the `invrocinp` variable as an argument to the `at()` option. Again, we list the point estimates and standard errors first.

```
. drop ci*
. predict invrocit in 1849/1852, invroc at(invrocinp) se(serocinv) ci(cin)
> bfile(nsnrf)
. list xf xl xd invrocinp invrocit serocinv if !missing(invrocit)
```

	xf	xl	xd	invroc~p	invrocit	serocinv
1851.	10.01	5.5	.5	.5	.0615144	.0209516
1852.	10.01	6.5	4	.5	.0043298	.003835

These also match those of example 3 of [R] **rocregplot**. Listing the confidence intervals shows identical results as well. First, we list the confidence intervals under the covariate values `xf=10.01`, `xl=5.5`, and `xd=.5`.

```
. list xf xl xd invrocinp invrocit cin* in 1851
```

	xf	xl	xd	invroc~p	invrocit	cin_l	cin_u
1851.	10.01	5.5	.5	.5	.0615144	.0204499	.1025789

Now, we list the confidence intervals for false-positive rate under the covariate values `xf=10.01`, `xl=6.5`, and `xd=4`.

```
. list xf xl xd invrocinp invrocit cin* in 1852
```

	xf	xl	xd	invroc~p	invrocit	cin_l	cin_u
1852.	10.01	6.5	4	.5	.0043298	-.0031867	.0118463

◁

The `predict` command can also be used after a maximum-likelihood ROC model is fit.

▷ Example 3: Maximum likelihood ROC, invROC, and ROC value

In the previous example, we revisited the estimating equations fit of a probit model with ROC covariates frequency (`xf`), intensity (`xl`), and hearing loss severity (`xd`) to the Stover et al. (1996) audiology study data. A maximum likelihood fit of the same model was performed in example 10 of [R] **rocreg**. In example 2 of [R] **rocregplot**, we used `rocregplot` to estimate ROC values and false-positive rates for this model under two covariate configurations. We will use `predict` to obtain the same estimates. We will also estimate the partial area under the ROC curve.

We append the data as in the previous example. This leads to the following four final observations in the data.

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

. rocreg d nsnr, probit ctrlcov(xf xl) roccov(xf xl xd) ml cluster(id)
  (output omitted)

. append using newdata

. list xf xl xd invrocinp rocinp in 1849/1852
```

	xf	xl	xd	invroc~p	rocinp
1849.	10.01	5.5	.5	.	.2
1850.	10.01	6.5	4	.	.2
1851.	10.01	5.5	.5	.5	.7
1852.	10.01	6.5	4	.5	.7

Now, we predict the ROC value for false-positive rates of 0.2 and 0.7. Under maximum likelihood prediction, only Wald-type confidence intervals are produced. We specify a new variable name for the standard error in the `se()` option and a stubname for the confidence interval variables in the `ci()` option.

```
. predict rocit in 1849/1852, roc at(rocinp) se(seroc) ci(ci)

. list xf xl xd rocinp rocit seroc ci_l ci_u if !missing(rocit), noobs
```

xf	xl	xd	rocinp	rocit	seroc	ci_l	ci_u
10.01	5.5	.5	.2	.7608593	.0510501	.660803	.8609157
10.01	6.5	4	.2	.9499408	.0179824	.914696	.9851856
10.01	5.5	.5	.7	.978951	.0097382	.9598644	.9980376
10.01	6.5	4	.7	.9985001	.0009657	.9966073	1.000393

These results match our estimates in example 2 of [R] **rocregplot**. We also match example 2 of [R] **rocregplot** when we estimate the false-positive rate for a ROC value of 0.5.

```
. drop ci*

. predict invrocit in 1851/1852, invroc at(invrocinp) se(serocinv) ci(ci)

. list xf xl xd invrocinp invrocit serocinv ci_l ci_u if !missing(invrocit),
> noobs
```

xf	xl	xd	invroc~p	invrocit	serocinv	ci_l	ci_u
10.01	5.5	.5	.5	.0578036	.0198626	.0188736	.0967336
10.01	6.5	4	.5	.0055624	.0032645	-.0008359	.0119607

◁

Example 4: Maximum likelihood ROC, pAUC, and ROC value

In example 13 of [R] **rocreg**, we fit a maximum-likelihood marginal probit model to each classifier of the fictitious dataset generated from Hanley and McNeil (1983). In example 5 of [R] **rocregplot**, `rocregplot` was used to draw the ROC for the `mod1` and `mod3` classifiers. Estimates of the ROC value and false-positive rate were also obtained with Wald-type confidence intervals.

We return to this example, this time using `predict` to estimate the ROC value and false-positive rate. We will also estimate the pAUC for the false-positive rates of 0.3 and 0.8.

First, we add the input variables to the data. The variable `paucinp` will hold the 0.3 and 0.8 false-positive rates that we will input to pAUC. The variable `invrocinp` holds the ROC value of 0.8 for which we will estimate the false-positive rate. Finally, the variable `rocinp` holds the false-positive rates of 0.15 and 0.75 for which we will estimate the ROC value.

```
. use https://www.stata-press.com/data/r18/ct2, clear
(Reconstruction of CT images)
. rocreg status mod1 mod2 mod3, probit ml
  (output omitted)
. quietly generate paucinp = .3 in 111
. quietly replace paucinp = .8 in 112
. quietly generate invrocinp = .8 in 112
. quietly generate rocinp = .15 in 111
. quietly replace rocinp = .75 in 112
```

Then, we estimate the ROC value for false-positive rates 0.15 and 0.75 under classifier `mod1`. The point estimate is stored in `roc1`. Wald confidence intervals and standard errors are also estimated. We find that these results match those of example 5 of [R] **rocregplot**.

```
. predict roc1 in 111/112, classvar(mod1) roc at(rocinp) se(sr1) ci(cir1)
. list rocinp roc1 sr1 cir1* in 111/112
```

	rocinp	roc1	sr1	cir1_l	cir1_u
111.	.15	.7934935	.0801363	.6364293	.9505578
112.	.75	.9931655	.0069689	.9795067	1.006824

Now, we perform the same estimation under the classifier `mod3`.

```
. predict roc3 in 111/112, classvar(mod3) roc at(roci) se(sr3) ci(cir3)
. list rocinp roc3 sr3 cir3* in 111/112
```

	rocinp	roc3	sr3	cir3_l	cir3_u
111.	.15	.8888596	.0520118	.7869184	.9908009
112.	.75	.9953942	.0043435	.9868811	1.003907

Next, we estimate the false-positive rate for the ROC value of 0.8. These results also match example 5 of [R] **rocregplot**.

```
. predict invroc1 in 112, classvar(mod1) invroc at(invrocinp) se(sir1) ci(ciir1)
. list invrocinp invroc1 sir1 ciir1* in 112
```

	invroc~p	invroc1	sir1	ciir1_l	ciir1_u
112.	.8	.1556435	.069699	.0190361	.292251

```
. predict invroc3 in 112, classvar(mod3) invroc at(invrocinp) se(sir3) ci(ciir3)
. list invrocinp invroc3 sir3 ciir3* in 112
```

	invroc~p	invroc3	sir3	ciir3_l	ciir3_u
112.	.8	.0661719	.045316	-.0226458	.1549896

Finally, we estimate the pAUC for false-positive rates of 0.3 and 0.8. The point estimate is calculated by numeric integration. Wald confidence intervals are obtained with the delta method. Further details are presented in *Methods and formulas*.

```
. predict pauc1 in 111/112, classvar(mod1) pauc at(paucinp) se(sp1) ci(cip1)
. list paucinp pauc1 sp1 cip1* in 111/112
```

	paucinp	pauc1	sp1	cip1_l	cip1_u
111.	.3	.221409	.0240351	.174301	.268517
112.	.8	.7033338	.0334766	.6377209	.7689466

```
. predict pauc3 in 111/112, classvar(mod3) pauc at(paucinp) se(sp3) ci(cip3)
. list paucinp pauc3 sp3 cip3* in 111/112
```

	paucinp	pauc3	sp3	cip3_l	cip3_u
111.	.3	.2540215	.0173474	.2200213	.2880217
112.	.8	.7420408	.0225192	.6979041	.7861776

◁

Using estat nproc

When you initially use `rocreg` to fit a nonparametric ROC curve, you can obtain bootstrap estimates of a ROC value, false-positive rate, area under the ROC curve, and partial area under the ROC curve. The `estat nproc` command allows the user to estimate these parameters after `rocreg` has originally been used.

The seed and resampling settings used by `rocreg` are used by `estat nproc`. So the results for these new statistics are identical to what they would be if they had been initially estimated in the `rocreg` command. These new statistics, together with those previously estimated in `rocreg`, are returned in `r()`.

We demonstrate with an example.

▷ Example 5: Nonparametric ROC, invROC, and pAUC

In example 3 of [R] **rocreg**, we examined data from a pancreatic cancer study (Wieand et al. 1989). Two continuous classifiers, `y1` (CA 19-9) and `y2` (CA 125), were used for the true status variable `d`. In that example, we estimated various quantities including the false-positive rate for a ROC value of 0.6 and the pAUC for a false-positive rate of 0.5. Here we replicate that estimation with a call to `rocreg` to estimate the former and follow that with a call to `estat nproc` to estimate the latter. For simplicity, we restrict estimation to classifier `y1` (CA 19-9).

We start by executing `rocreg`, estimating the false-positive rate for a ROC value of 0.6. This value is specified in `invroc()`. Case–control resampling is used by specifying the `bootcc` option.

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

. rocreg d y1, 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

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

Now, we will estimate the pAUC for the false-positive rate of 0.5 using `estat nproc` and the `pauc()` option.

```
. matrix list e(b)

symmetric e(b)[1,1]
          y1:
    invroc_1
y1         0

. estat nproc, pauc(.5)

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

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

Partial area under the ROC curve
   Status    : d
   Classifier: y1

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

```
. matrix list r(b)

r(b)[1,2]
               y1:         y1:
         invroc_1      pauc_1
y1              0   .39324619

. matrix list e(b)

symmetric e(b)[1,1]
             y1:
       invroc_1
y1            0

. matrix list r(V)

symmetric r(V)[2,2]
                        y1:          y1:
                  invroc_1       pauc_1
y1:invroc_1      .00065477
  y1:pauc_1     -.00033586    .00047975

. matrix list e(V)

symmetric e(V)[1,1]
                       y1:
                 invroc_1
y1:invroc_1     .00065477
```

◁

The advantages of using `estat nproc` are twofold. First, you can estimate additional parameters of interest without having to respecify the bootstrap settings you did with `rocreg`; instead `estat nproc` uses the bootstrap settings that were stored by `rocreg`. Second, parameters estimated with `estat nproc` are added to those parameters estimated by `rocreg` and returned in the matrices `r(b)` (parameter estimates) and `r(V)` (variance–covariance matrix). Thus, you can also obtain correlations between any quantities you wish to estimate.

Stored results

`estat nproc` stores the following in `r()`:

Matrices

`r(b)`	coefficient vector
`r(V)`	variance–covariance matrix of the estimators
`r(ci_normal)`	normal-approximation confidence intervals
`r(ci_percentile)`	percentile confidence intervals
`r(ci_bc)`	bias-corrected confidence intervals

Methods and formulas

Details on computation of the nonparametric ROC curve and the estimation of the parametric ROC curve model coefficients can be found in [R] **rocreg**. Here we describe how to estimate the ROC curve summary statistics for a parametric model. The cumulative distribution function, g, can be the standard normal cumulative distribution function, Φ.

Methods and formulas are presented under the following headings:

Parametric model: Summary parameter definition
Maximum likelihood estimation
Estimating equations estimation

Parametric model: Summary parameter definition

Conditioning on covariates $\mathbf{x}$, we have the following ROC curve model:

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

$\mathbf{x}$ can be constant, and $\boldsymbol{\beta} = \beta_0$, the constant intercept.

We can solve this equation to obtain the false-positive rate value u for a ROC value of r:

$$u = g\left[\{g^{-1}\,(r) - \mathbf{x}'\boldsymbol{\beta}\}\alpha^{-1}\right]$$

The partial area under the ROC curve for the false-positive rate u is defined by

$$\text{pAUC}\,(u) = \int_o^u g\{\mathbf{x}'\boldsymbol{\beta} + \alpha g^{-1}\,(t)\}dt$$

The area under the ROC curve is defined by

$$\text{AUC} = \int_o^1 g\{\mathbf{x}'\boldsymbol{\beta} + \alpha g^{-1}\,(t)\}dt$$

When g is the standard normal cumulative distribution function Φ, we can express the AUC as

$$\text{AUC} = \Phi\left(\frac{\mathbf{x}'\boldsymbol{\beta}}{\sqrt{1+\alpha^2}}\right)$$

Maximum likelihood estimation

We allow maximum likelihood estimation under probit parametric models, so $g = \Phi$. The ROC value, false-positive rate, and AUC parameters all have closed-form expressions in terms of the covariate values $\mathbf{x}$, coefficient vector $\boldsymbol{\beta}$, and slope parameter α. So to estimate these three types of summary parameters, we use the delta method (Oehlert 1992; Phillips and Park 1988). Particularly, we use the `nlcom` command (see [R] **nlcom**) to implement the delta method.

To estimate the partial area under the ROC curve for false-positive rate u, we use numeric integration. A trapezoidal approximation is used in calculating the integrals. A numeric integral of the $\text{ROC}(t)$ function conditioned on the covariate values $\mathbf{x}$, coefficient vector estimate $\widehat{\boldsymbol{\beta}}$, and slope parameter estimate $\widehat{\alpha}$ is computed over the range $t = [0, u]$. This gives us the point estimate of $\text{pAUC}(u)$.

To calculate the standard error and confidence intervals for the point estimate of $\text{pAUC}(u)$, we again use the delta method. Details on the delta method algorithm can be found in *Methods and formulas* of [R] **nlcom** and the earlier mentioned references.

Under maximum likelihood estimation, the coefficient estimates $\widehat{\boldsymbol{\beta}}$ and slope estimate $\widehat{\alpha}$ are asymptotically normal with variance matrix $\mathbf{V}$. For convenience, we rename the parameter vector $[\boldsymbol{\beta}', \alpha]$ to the k-parameter vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_k]$. We will also explicitly refer to the conditioning of the ROC curve by $\boldsymbol{\theta}$ in its mention as $\text{ROC}(t, \boldsymbol{\theta})$.

Under the delta method, the continuous scalar function of the estimate $\widehat{\boldsymbol{\theta}}$, $f(\widehat{\boldsymbol{\theta}})$ has asymptotic mean $f(\boldsymbol{\theta})$ and asymptotic covariance

$$\widehat{\text{Var}}\left\{f(\widehat{\boldsymbol{\theta}})\right\} = \mathbf{f}\mathbf{V}\mathbf{f}'$$

where $\mathbf{f}$ is the $1 \times k$ matrix of derivatives for which

$$\mathbf{f}_{1j} = \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_j} \qquad j = 1, \ldots, k$$

The asymptotic covariance of $f(\widehat{\boldsymbol{\theta}})$ is estimated and then used in conjunction with $f(\widehat{\boldsymbol{\theta}})$ for further inference, including Wald confidence intervals, standard errors, and hypothesis testing.

In the case of pAUC(u) estimation, our $f(\widehat{\boldsymbol{\theta}})$ is the aforementioned numeric integral of the ROC curve. It estimates $f(\boldsymbol{\theta})$, the true integral of the ROC curve on the $[0, u]$ range. The $\mathbf{V}$ variance matrix is estimated using the likelihood information that `rocreg` calculated, and the estimation is performed by `rocreg` itself.

The partial derivatives of $f(\boldsymbol{\theta})$ can be determined by using Leibnitz's rule (Weisstein 2011):

$$\mathbf{f}_{1j} = \frac{\partial}{\partial \theta_j} \int_0^u \text{ROC}(t, \boldsymbol{\theta}) dt = \int_0^u \frac{\partial}{\partial \theta_j} \text{ROC}(t, \boldsymbol{\theta}) dt \qquad j = 1, \ldots, k$$

When θ_j corresponds with the slope parameter α, we obtain the following partial derivative:

$$\frac{\partial}{\partial \alpha} \text{pAUC}(u) = \int_0^u \phi\{\mathbf{x}'\boldsymbol{\beta} + \alpha \Phi^{-1}(t)\} \Phi^{-1}(t)\, dt$$

The partial derivative of $f(\boldsymbol{\theta})$ [pAUC(u)] for β_0 is the following:

$$\frac{\partial}{\partial \beta_0} \text{pAUC}(u) = \int_0^u \phi\{\mathbf{x}'\boldsymbol{\beta} + \alpha \Phi^{-1}(t)\} dt$$

For a nonintercept coefficient, we obtain the following:

$$\frac{\partial}{\partial \beta_i} \text{pAUC}(u) = \int_0^u x_i \phi\{\mathbf{x}'\boldsymbol{\beta} + \alpha \Phi^{-1}(t)\} dt$$

We can estimate each of these integrals by numeric integration, plugging in the estimates $\widehat{\boldsymbol{\beta}}$ and $\widehat{\alpha}$ for the parameters. This, together with the previously calculated estimate $\widehat{\mathbf{V}}$, provides an estimate of the asymptotic covariance of $f(\widehat{\boldsymbol{\theta}}) = \widehat{\text{pAUC}}(u)$, which allows us to perform further statistical inference on pAUC(u).

Estimating equations estimation

When we fit a model using the Alonzo and Pepe (2002) estimating equations method, we use the bootstrap to perform inference on the ROC curve summary parameters. Each bootstrap sample provides a sample of the coefficient estimates $\boldsymbol{\beta}$ and the slope estimates α. Using the formulas in *Parametric model: Summary parameter definition* under *Methods and formulas*, we can obtain an estimate of the ROC, false-positive rate, or AUC for each resample. Using numeric integration (with the trapezoidal approximation), we can also estimate the pAUC of the resample.

By making these calculations, we obtain a bootstrap sample of our summary parameter estimate. We then obtain bootstrap standard errors, normal approximation confidence intervals, percentile confidence intervals, and bias-corrected confidence intervals using this bootstrap sample. Further details can be found in [R] **bootstrap**.

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.

Choi, B. C. K. 1998. Slopes of a receiver operating characteristic curve and likelihood ratios for a diagnostic test. *American Journal of Epidemiology* 148: 1127–1132. https://doi.org/10.1093/oxfordjournals.aje.a009592.

Hanley, J. A., and B. J. McNeil. 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.

Norton, S. J., M. P. Gorga, J. E. Widen, 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.

Oehlert, G. W. 1992. A note on the delta method. *American Statistician* 46: 27–29. https://doi.org/10.2307/2684406.

Phillips, P. C. B., and J. Y. Park. 1988. On the formulation of Wald tests of nonlinear restrictions. *Econometrica* 56: 1065–1083. https://doi.org/10.2307/1911359.

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.

Weisstein, E. W. 2011. Leibniz integral rule. From *Mathworld*—A Wolfram Web Resource. http://mathworld.wolfram.com/LeibnizIntegralRule.html.

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** — Receiver operating characteristic (ROC) regression

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

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