Using Stata features to interpret and visualize regression results with examples for binary models.

Isabel Canette

Senior Statistician

StataCorp LP

2014 Spanish Stata Users Group meeting
Barcelona, October 23, 2014

Introduction

The best way to present results depends on the readers we are addressing.
For example, health practitioners are usually interested in individual predictions, and, eventually, the impact of individual decisions.
Policy makers are usually interested in population predictions, and, eventually, the impact of policy decisions.

We will discuss different tools to visualize and explain results to different audiences, which may be useful also in the teaching environment.

Binary models: probabilities

The default Stata prediction for binary models are probabilities. Health practitioners would be interested in individual probabilities. In the following model, we might be interested in the predicted probability of having high blood pressure for an individual (using the `nhanes2d` data).

In-sample predictions are computed with `predict`; prediction by default is the probability (option `pr`); we can use `predictnl`, which in addition gives us the standard errors:

```
. use nhanes2d, clear // webuse if data not in directory
. logit highbp height weight age female, nolog vsquish noheader
```

highbp	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
height	-.0355632	.0036591	-9.72	0.000	-.0427348	-.0283916
weight	.0499966	.0018348	27.25	0.000	.0464004	.0535927
age	.0469231	.0014573	32.20	0.000	.0440668	.0497794
female	-.3752472	.0641992	-5.85	0.000	-.5010753	-.2494192
_cons	-.074346	.6230625	-0.12	0.905	-1.295526	1.146834

```
. predict p
(option pr assumed; Pr(highbp))
. predictnl p2 = predict(pr), se(se)
. list height weight age fem p p2 se in 1/5
```

	height	weight	age	female	p	p2	se
1.	174.598	62.48	54	0	.3484242	.3484242	.0097808
2.	152.297	48.76	41	1	.1818179	.1818179	.0079658
3.	164.098	67.25	21	1	.1258913	.1258913	.0059445
4.	162.598	94.46	63	1	.8094957	.8094957	.0093575
5.	163.098	74.28	64	1	.614661	.614661	.0093765

In-sample or out-of-sample predictions after estimation can also be computed using `margins`, which, by default, computes the same prediction as `predict`, and displays additional information, including CIs.:

```
. margins, at(height=174.598  weight=62.48  age =54 female=0)

Adjusted predictions                                  Number of obs   =      10351
Model VCE    : OIM

Expression   : Pr(highbp), predict()
at           : height          =     174.598
               weight          =       62.48
               age             =          54
               female          =           0
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	.3484242	.0097808	35.62	0.000	.3292541	.3675942

Statisticians are familiar with the importance of presenting confidence intervals together with point estimates.
Even thought the CI concept is difficult to non-statisticians, everybody has some intuitive understanding of the relationship of the length of the confidence interval with the reliability of the estimate we are presenting.

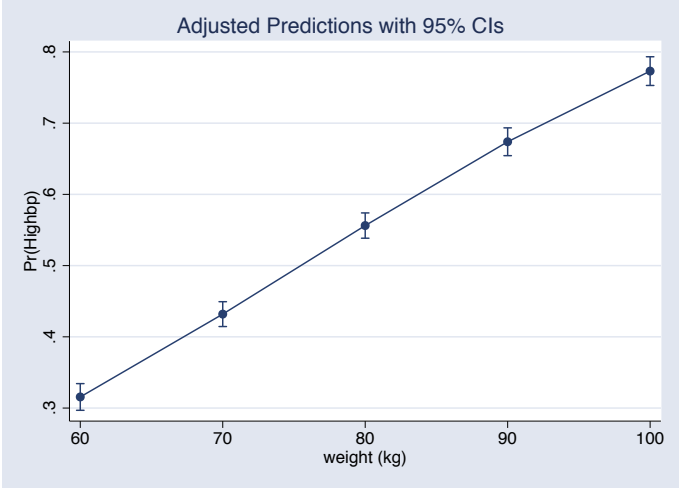
Variables not mentioned in `at()` option will be accounted by averaging results. When trying to understand the problem, performing as many plots as possible might help to get insight into it.

We can use `marginsplot` after `margins` to visualize predictions at different values of a covariate:

```
. margins, at(height = 170 age = 50 female = 0 weight = (60(10)100)) noatleg

Adjusted predictions                                  Number of obs   =      10351
Model VCE    : OIM

Expression   : Pr(highbp), predict()
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
_at						
1	.3155848	.0095498	33.05	0.000	.2968676	.334302
2	.4318833	.0088948	48.55	0.000	.4144498	.4493168
3	.55621	.0090589	61.40	0.000	.5384548	.5739652
4	.6738742	.0099572	67.68	0.000	.6543584	.6933901
5	.7730697	.0102775	75.22	0.000	.7529261	.7932133

```
. marginsplot

  Variables that uniquely identify margins: weight
```



Policy makers would be more interested in population averages of probabilities. `margins`, without `at()` option, computes averages of predictions over the sample.

```
. *vce(robust) option is required for -vce(uncondional)-
. quietly logit highbp height weight age female, vce(robust)

. margins, vce(unconditional)

Predictive margins                                Number of obs   =      10351

Expression   : Pr(highbp), predict()
```

	Unconditional Margin	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	.4227611	.0048557	87.06	0.000	.413244	.4322782

```
. quietly predict p

. quietly summ p

. display r(mean)
.42276109
```

If we want to use this measure as an estimator of the population average probability, we need to use the option `vce(unconditional)` to account for the fact that we are working on a sample.

Odds ratios

There is more than one approach to interpreting output from a logistic regression; many researchers advocate for the use of odds ratios. This is because the model itself assumes that (in the absence of interactions) those are constant over covariate patterns, and they can be computed by exponentiating the coefficients.

Example: hypothetical example for the effect of carrot consumption on the need for lenses (from the UCLA website):

```
. *use http://www.ats.ucla.edu/stat/stata/faq/eyestudy
. use eyestudy, clear
. logit lenses i.carrot i.gender latitude, nolog vsquish noheader or
```

lenses	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.carrot	.347253	.1472796	-2.49	0.013	.1512265	.7973779
2.gender	.6267289	.2630932	-1.11	0.266	.275268	1.426934
latitude	.977823	.0277312	-0.79	0.429	.9249538	1.033714
_cons	5.476334	6.237333	1.49	0.135	.5874952	51.04763

```
. quietly predict p
. list carrot gender latitude p in 1/5
```

	carrot	gender	latitude	p
1.	0	1	33	.7231932
2.	0	2	46	.5502224
3.	1	1	32	.4812792
4.	0	2	26	.6570335
5.	1	1	25	.5205055

We can see that probabilities vary across covariate patterns

Odds for an individual with a specific covariate pattern, are defined as:

$$\text{Odds for an event} = \frac{\text{probability of an event}}{1 - \text{probability of an event}}$$

which is, in our case:

$$\text{Odds for an event} = \frac{\text{probability of lenses} = 1}{1 - \text{probability of lenses} = 1}$$

Odds ratio are defined for each covariate; Usually, researchers are interested in odds ratio for the treatment variable:

$$\text{Odds for an event} = \frac{\text{Odds assuming that treatment} = 1}{\text{Odds assuming that treatment} = 0}$$

OR is the quotient for the odds for an individual assuming that undertook the treatment, and the odds for the same individual, assuming that didn't undertake the treatment.

An easy way to explain this concept is to show how to directly predict these values:

```
. *create a backup variable for carrot
. generate carrot_back = carrot

.
. *compute odds for each observation, assuming carrot = 1
. replace carrot = 1
(49 real changes made)

. predict p1
(option pr assumed; Pr(lenses))

. generate odds1 = p1/(1-p1)

.
. *compute odds for each observation, assuming carrot = 0
. replace carrot = 0
(100 real changes made)

. predict p0
(option pr assumed; Pr(lenses))

. generate odds0 = p0/(1-p0)

.
. *compute odds ratios
. generate OR_carrot = odds1/odds0

. *restore original variable for carrot
. replace carrot = carrot_back
(51 real changes made)
```

```
. list latitude gender odds1 odds0  OR_carrot in 1/5
```

	latitude	gender	odds1	odds0	OR_car~t
1.	33	1	.9072431	2.612628	.347253
2.	46	2	.4248019	1.223321	.347253
3.	32	1	.9278194	2.671883	.347253
4.	26	2	.6652452	1.915737	.347253
5.	25	1	1.08553	3.126049	.347253

```
. logit, or
```

```
Logistic regression                             Number of obs   =        100
                                                LR chi2(3)      =       7.65
                                                Prob > chi2     =     0.0538
Log likelihood = -65.308053                     Pseudo R2       =     0.0553
```

lenses	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.carrot	.347253	.1472796	-2.49	0.013	.1512265	.7973779
2.gender	.6267289	.2630932	-1.11	0.266	.275268	1.426934
latitude	.977823	.0277312	-0.79	0.429	.9249538	1.033714
_cons	5.476334	6.237333	1.49	0.135	.5874952	51.04763

Naturally, this can be used for continuous covariates also.

In short, if the treatment variable is not part of an interaction in the logit model, odds ratios are the same for all the individuals, and therefore, the same estimates work for individual level and for population level.

Note: if the treatment variable is interacted with another covariate, now odds ratios are not constant, and need to be computed either with `predictnl` or `margins`.

Risk ratios: easier to interpret, but not displayed on the command output.

In a logistic model with other covariates (in addition to the treatment), there is variation for the RR among individuals.

$$\text{Risk ratio} = \frac{\text{probability of an event assuming treatment} = 1}{\text{probability of an event assuming treatment} = 0}$$

Naturally, we can compute those manually, and we could use `nlcom` to compute confidence intervals.
If we want to choose the domain for our plots, we can use automated tools for our computations and our confidence intervals

Note: the following computations of RR are valid for any model for binary dependent variable with independent observations (e.g. probit, cloglog, etc).

ORs are sometimes interpreted as RRs, which can be misleading. Nowadays, there is not need to make such rough approximations, because we have tools to obtain what we want.

Obtaining risk-ratios by computing log-risk-ratios.

`margins, dydx()` computes derivatives of the predictions respect to a continuous covariate, or finite differences for a dummy variable. That is, if the prediction is $f(x)$, for a binary covariate x,

```
margins, dydx(x)
```

will compute $f(1) - f(0)$.
The same way,

```
margins, eydx(x)
```

will compute, for this binary covariate,

$$\log(f(1)) - \log(f(0)) = \log(f(1)/f(0))$$

In our case the default prediction is
$f(i) = pi =$ probability of positive outcome when treatment $= i$;
therefore, the computed value will be $\log(p1/p0) = \log(\text{RR})$

Example: probability of a newborn with low weight (Hosmer & Lemeshow data) ("smoke" would be our "negative treatment")

```
. use lbw, clear //webuse if not in current directory
(Hosmer & Lemeshow data)

. logit low  i.smoke age i.race, or nolog vsquish
```

```
Logistic regression                             Number of obs   =        189
                                                LR chi2(4)      =      15.81
                                                Prob > chi2     =     0.0033
Log likelihood =  -109.4311                     Pseudo R2       =     0.0674
```

low	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
smoke						
smoker	3.00582	1.118001	2.96	0.003	1.449982	6.231081
age	.9657186	.0322573	-1.04	0.296	.9045206	1.031057
race						
black	2.749483	1.356659	2.05	0.040	1.045318	7.231924
other	2.876948	1.167921	2.60	0.009	1.298314	6.375062
_cons	.365111	.3146026	-1.17	0.242	.0674491	1.976395

We use `margins` to compute log-risk-ratios for `smoke`; `post` option allows us to use those results afterwards.

```
. margins, eydx(smoke) predict(pr) at(age=(15(10)45)) over(race) post noatl vsq
> uish

Average marginal effects                        Number of obs   =        189
Model VCE    : OIM

Expression   : Pr(low), predict(pr)
ey/dx w.r.t. : 1.smoke
over         : race
```

	Delta-method ey/dx	Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.smoke						
_at#race						
1#white	.7954289	.2937747	2.71	0.007	.2196411	1.371217
1#black	.5419842	.2148224	2.52	0.012	.12094	.9630285
1#other	.5298297	.1774735	2.99	0.003	.181988	.8776714
2#white	.8649764	.3055523	2.83	0.005	.2661049	1.463848
2#black	.6349475	.2320708	2.74	0.006	.1800971	1.089798
2#other	.6230277	.1930333	3.23	0.001	.2446894	1.001366
3#white	.9223918	.3254033	2.83	0.005	.2846131	1.56017
3#black	.7233161	.2743695	2.64	0.008	.1855617	1.261071
3#other	.7122553	.2418088	2.95	0.003	.2383188	1.186192
4#white	.9680835	.3427932	2.82	0.005	.2962212	1.639946
4#black	.8029534	.3190297	2.52	0.012	.1776668	1.42824
4#other	.7932087	.2944309	2.69	0.007	.2161347	1.370283

Now, risk ratios can be obtaining by exponentiating the log-risk ratios; (because we posted our results, we can re-display them with `ereturn display`)

```
. ereturn display, eform("risk ratios") vsquish
```

	risk ratios	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.smoke						
_at#race						
1#white	2.215391	.6508257	2.71	0.007	1.24563	3.940141
1#black	1.719415	.369369	2.52	0.012	1.128557	2.619618
1#other	1.698643	.3014642	2.99	0.003	1.1996	2.405292
2#white	2.37495	.7256714	2.83	0.005	1.304872	4.32256
2#black	1.886923	.4378997	2.74	0.006	1.197334	2.973673
2#other	1.864565	.3599232	3.23	0.001	1.277225	2.721998
3#white	2.515299	.8184866	2.83	0.005	1.329248	4.759632
3#black	2.061257	.5655462	2.64	0.008	1.203894	3.529197
3#other	2.038584	.4929474	2.95	0.003	1.269114	3.274587
4#white	2.632894	.9025382	2.82	0.005	1.344768	5.154891
4#black	2.232124	.7121137	2.52	0.012	1.194427	4.171352
4#other	2.210478	.6508329	2.69	0.007	1.24127	3.936463

Another trick to compute and plot risk ratios (directly) is by using gsem. we can fit the same model twice with the same command, and then compute the quotient of predictions with treatment = 1 and treatment = 0.

```
. use lbw, clear
(Hosmer & Lemeshow data)

. keep low smoke age race

. gen obs = _n

. quietly expand 2, gen(repl)

. quietly reshape wide low smoke, i(obs) j(repl)

.
. *just show the gsem basic syntax (we add the constraints later)
. quietly gsem (low0 <- smoke0 age i.race, logit) ///
>      (low1 <- smoke1 age i.race, logit), noestimate
```

```
. *estimate the model
.  gsem (low0 <- i1.smoke0@a age@b i2.race@c2 i3.race@c3, logit) ///
>      (low1 <- i1..smoke1@a age@b i2.race@c2 i3.race@c3, logit) , nolog vsquis
> h nocnsr
```

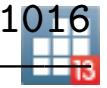
Generalized structural equation model — Number of obs = 189
Log likelihood = -218.8622

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
low0 <-						
smoke0						
smoker	1.10055	.263005	4.18	0.000	.5850701	1.616031
age	-.0348828	.023619	-1.48	0.140	-.0811752	.0114097
race						
black	1.011413	.348903	2.90	0.004	.3275756	1.69525
other	1.05673	.2870559	3.68	0.000	.494111	1.619349
_cons	-1.007554	.6201877	-1.62	0.104	-2.223099	.2079917
low1 <-						
smoke1						
smoker	1.10055	.263005	4.18	0.000	.5850701	1.616031
age	-.0348828	.023619	-1.48	0.140	-.0811752	.0114097
race						
black	1.011413	.348903	2.90	0.004	.3275756	1.69525
other	1.05673	.2870559	3.68	0.000	.494111	1.619349
_cons	-1.007554	.6201877	-1.62	0.104	-2.223099	.2079917

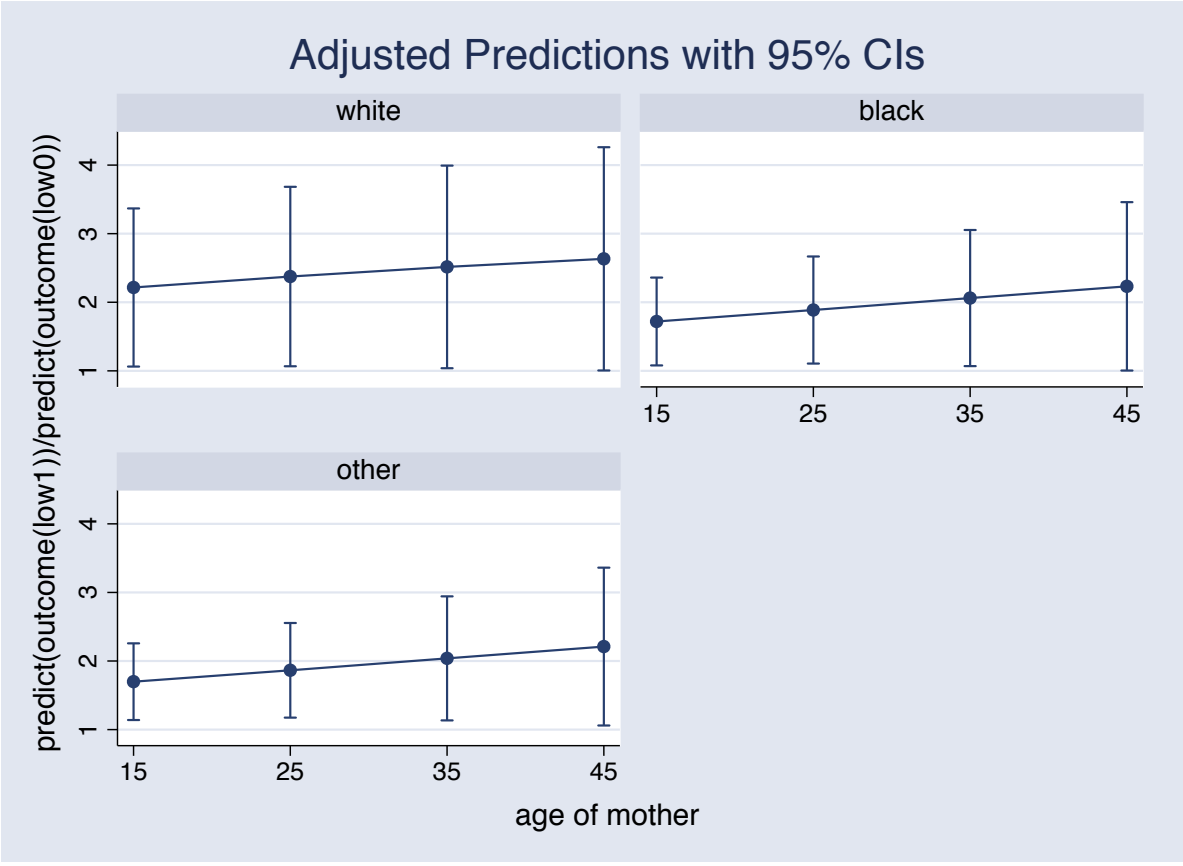
Use margins to obtain the risk ratios for variable smoke

```
. margins, expression(predict(outcome(low1))/predict(outcome(low0)) ) ///
>          at(smoke0 = 0 smoke1=1  age =(15(10)45) race=(1(1)3)) ///
>          noatlegend vsquish
Warning: prediction constant over observations.

Adjusted predictions                                  Number of obs   =        189
Model VCE    : OIM

Expression   : predict(outcome(low1))/predict(outcome(low0))
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
_at						
1	2.215391	.5881502	3.77	0.000	1.062638	3.368144
2	1.719415	.3266628	5.26	0.000	1.079168	2.359662
3	1.698643	.2853472	5.95	0.000	1.139373	2.257913
4	2.37495	.6675446	3.56	0.000	1.066587	3.683313
5	1.886923	.3982498	4.74	0.000	1.106368	2.667478
6	1.864565	.3518965	5.30	0.000	1.17486	2.554269
7	2.515299	.7537232	3.34	0.001	1.038029	3.99257
8	2.061257	.5062034	4.07	0.000	1.069117	3.053398
9	2.038584	.4614366	4.42	0.000	1.134185	2.942983
10	2.632894	.8304642	3.17	0.002	1.005214	4.260574
11	2.232124	.6263308	3.56	0.000	1.004538	3.459709
12	2.210478	.5870203	3.77	0.000	1.059939	3.361016



We can use `marginsplot` to plot the risk ratios; this time I'm using `bydimension()` option to show several plots in the same graph.



Note: Constraints are included in the previous model to estimate the correct covariance matrix (we don't want to count the same thing twice)

Notice that the two confidence intervals obtained by the two methods are not exactly the same.

The first method computes CIs based on the asymptotic normality of the log-RR; (CIs are computed for the log-RR, and then exponentiated).

The second method computes CIs based on the asymptotic normality of the RRs. Standard errors are computed used the delta methods, and these are used to obtain symmetric CIs.

Both methods are asymptotically correct.

Out-of-sample predictions (when we don't have the sample)

We can always apply a formula (manually) to compute out-of-sample predictions. However, if we have the original covariance matrix, we can use Stata to compute those predictions with CIs.
Without the original covariance, point estimates can be still computed. Also, out-of-sample validation diagnostics can be performed using measures that don't require the variance

Let's assume we have the output from the following model, and we want to compute predictions for a new individual.

```
. logit highbp c.weight#i.female c.weight##c.age c.weight#c.weight,  nolog vsqu
> ish
```

Logistic regression — Number of obs = 10351; LR chi2(5) = 2383.07; Prob > chi2 = 0.0000; Log likelihood = -5859.2282; Pseudo R2 = 0.1690

highbp	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
female# c.weight						
1	.0007826	.000647	1.21	0.226	-.0004855	.0020507
weight	.0889377	.013967	6.37	0.000	.061563	.1163125
age	.1046649	.0075367	13.89	0.000	.0898933	.1194365
c.weight# c.age	-.0007447	.0001012	-7.36	0.000	-.0009431	-.0005463
c.weight# c.weight	-.0000596	.0000756	-0.79	0.431	-.0002078	.0000886
_cons	-8.971899	.6627519	-13.54	0.000	-10.27087	-7.672929

If we had the sample, we could use `margins` as explained before. If we don't, we can use Stata to obtain predictions (without implementing the formulas manually), by posting the results.

We first create an artificial dataset to run the model; this is the easiest way to get matrices with the right labels, where we then replace the actual results.

Then, we `repost` these matrices with the actual results, so the post-estimation commands can use them for predictions.

```
. clear
. program drop _all
. set seed 1357
. set obs 100
obs was 0, now 100
. gen weight = rnormal()
. gen female = runiform()<.5
. gen age = rnormal()
. gen highbp = runiform()<.5
.
. quietly logit highbp c.weight#ib0.female c.weight##c.age c.weight#c.weight
. mat list e(b)
e(b)[1,7]
        highbp:     highbp:     highbp:     highbp:     highbp:     highbp:
     0b.female#  1.female#                             c.weight#   c.weight#
     co.weight   c.weight       weight         age       c.age     c.weight
y1           0    .273594  -.35548071    .0116049  -.10238474    .17071731

        highbp:
          _cons
y1  -.27933898
```

```
. mat b = e(b)
. mat V = e(V)
. mat b1 = [0, .0007826,  .0889377,.1046649, -.0007447 ,-.0000596,  -8.9718986]
. mat V1 = J(7,7,0)
.
. mat b[1,1] =b1
. mat list b
b[1,7]
        highbp:     highbp:     highbp:     highbp:     highbp:     highbp:
     0b.female#  1.female#                          c.weight#   c.weight#
     co.weight   c.weight      weight         age       c.age    c.weight
y1           0   .0007826   .0889377   .1046649   -.0007447   -.0000596

        highbp:

         _cons
y1  -8.9718986
. mat V[1,1] = V1
```

```
.
.   program myrepost, eclass
  1.     ereturn repost b=b V=V
  2.   end

.
. myrepost

. margins, at(weight = 80 female = 0 age = 60) noatlegend

Adjusted predictions                                  Number of obs   =        100
Model VCE    : OIM

Expression   : Pr(highbp), predict()
```

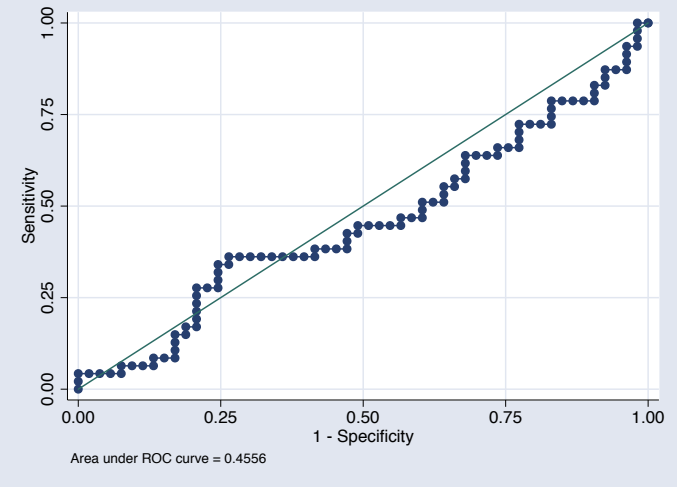
	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	.6146762	.	.	.	.	.

Notes:
ib0 notation has been used in the first logistic model to ensure that the base category is the same as in the original.
If you have a covariance matrix, you should post it. (you need it to get standard errors). If you can't obtain it, you should post zeros in e(V) to avoid misleading results.

This trick also has been used to get out-of-sample validation, that is, to assess how the original model would fit on a second dataset. (some diagnostic methods do not depend on the covariance matrix).

As an example, we will see how the original model fits to our simulated dataset (naturally, we shouldn't expect a good fit).

```
. estat gof

Logistic model for highbp, goodness-of-fit test

       number of observations =       100
 number of covariate patterns =       100
            Pearson chi2(94) =    375573.13
                 Prob > chi2 =         0.0000

. lroc

Logistic model for highbp

number of observations =      100
area under ROC curve   =   0.4556
```



Final Remarks

- There are many ways to present and visualize results from our estimations; the way we choose should be targeted to our specific audience and purposes.
- A powerful tool to interpret results (not discussed here in depth) is computing marginal effects. You might want to explore this possibility also.
- When we fit a logistic model, odds ratios are easy to compute, but not so easy to interpret. If you believe that your audience will be more comfortable with risk ratios, show those
- When computing predictions for a particular individual, it is always advisable to directly show the predictions, eventually for different scenarios, with their confidence intervals.
- The word "adjusted" has been used in many ways in the literature. If you report adjusted results, make sure that you explain what it means in your context.