

Post-Estimation Commands for MLogit

These notes borrow heavily (sometimes verbatim) from Long & Freese, 2006 Regression Models for Categorical Dependent Variables Using Stata, 2nd Edition.

Many/most of the Stata & `spost9` post-estimation commands work pretty much the same way for `mlogit` as they do for `logit` and/or `ologit`. We'll therefore concentrate primarily on the commands that are somewhat unique.

Making comparisons across categories. By default, `mlogit` sets the base category to the outcome with the most observations. You can change this with the `basecategory` option. `mlogit` reports coefficients for the effect of each independent variable on each category relative to the base category. Hence, you can easily see whether, say, `yr89` significantly affects the likelihood of your being in the SD versus the SA category; but you can't easily tell whether `yr89` significantly affects the likelihood of your being in, say, SD versus D, when neither is the base. You could just keep rerunning models with different base categories; but `listcoef` makes things easier by presenting estimates for all combinations of outcome categories.

```
. use http://www.nd.edu/~rwilliam/xsoc694/long2003/ordwarm2.dta
. mlogit warm yr89 male white age ed prst, b(4)
```

```
Multinomial logistic regression          Number of obs   =          2293
                                         LR chi2(18)     =          349.54
                                         Prob > chi2     =           0.0000
Log likelihood = -2820.9982              Pseudo R2      =           0.0583
```

	warm	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
SD	yr89	-1.160197	.1810497	-6.41	0.000	-1.515048 - .8053457
	male	1.226454	.167691	7.31	0.000	.8977855 1.555122
	white	.834226	.2641771	3.16	0.002	.3164485 1.352004
	age	.0316763	.0052183	6.07	0.000	.0214487 .041904
	ed	-.1435798	.0337793	-4.25	0.000	-.209786 -.0773736
	prst	-.0041656	.0070026	-0.59	0.552	-.0178904 .0095592
	_cons	-.722168	.4928708	-1.47	0.143	-1.688177 .2438411
D	yr89	-.4255712	.1318065	-3.23	0.001	-.6839071 -.1672352
	male	1.326716	.137554	9.65	0.000	1.057115 1.596317
	white	.4126344	.1872718	2.20	0.028	.0455885 .7796804
	age	.0292275	.0042574	6.87	0.000	.0208832 .0375718
	ed	-.0513285	.0283399	-1.81	0.070	-.1068737 .0042167
	prst	-.0130318	.0055446	-2.35	0.019	-.023899 -.0021645
	_cons	-.3088357	.3938354	-0.78	0.433	-1.080739 .4630676
A	yr89	-.0625534	.1228908	-0.51	0.611	-.3034149 .1783082
	male	.8666833	.1310965	6.61	0.000	.6097389 1.123628
	white	.3002409	.1710551	1.76	0.079	-.0350211 .6355028
	age	.0066719	.0041053	1.63	0.104	-.0013744 .0147181
	ed	-.0330137	.0274376	-1.20	0.229	-.0867904 .020763
	prst	-.0017323	.0052199	-0.33	0.740	-.0119631 .0084985
	_cons	.3932277	.3740361	1.05	0.293	-.3398697 1.126325

(Outcome warm==SA is the comparison group)

```
. listcoef yr89, help
```

```
mlogit (N=2293): Factor Change in the Odds of warm
```

```
Variable: yr89 (sd=.48971781)
```

Odds comparing		b	z	P> z	e^b	e^bStdX
Group 1 vs	Group 2					
SD	-D	-0.73463	-4.434	0.000	0.4797	0.6978
SD	-A	-1.09764	-6.705	0.000	0.3337	0.5842
SD	-SA	-1.16020	-6.408	0.000	0.3134	0.5666
D	-SD	0.73463	4.434	0.000	2.0847	1.4330
D	-A	-0.36302	-3.395	0.001	0.6956	0.8371
D	-SA	-0.42557	-3.229	0.001	0.6534	0.8119
A	-SD	1.09764	6.705	0.000	2.9971	1.7118
A	-D	0.36302	3.395	0.001	1.4377	1.1946
A	-SA	-0.06255	-0.509	0.611	0.9394	0.9698
SA	-SD	1.16020	6.408	0.000	3.1906	1.7650
SA	-D	0.42557	3.229	0.001	1.5305	1.2317
SA	-A	0.06255	0.509	0.611	1.0646	1.0311

```
-----  
b = raw coefficient  
z = z-score for test of b=0  
P>|z| = p-value for z-test  
e^b = exp(b) = factor change in odds for unit increase in X  
e^bStdX = exp(b*SD of X) = change in odds for SD increase in X
```

What you are seeing, then, are the coefficients you would get if the category listed as Group 2 was the reference category. Based on the above, we see that yr89 has little effect on strongly agreeing versus agreeing. In every other contrast though, the difference is significant.

It is possible to get overwhelmed with output, at least if you do this for all variables. The `pvalue` option can limit the output to differences which are significant:

```
. listcoef , help pvalue(.01)
```

```
mlogit (N=2293): Factor Change in the Odds of warm when P>|z| < 0.01
```

```
Variable: yr89 (sd=.48971781)
```

Odds comparing		b	z	P> z	e^b	e^bStdX
Group 1 vs	Group 2					
SD	-D	-0.73463	-4.434	0.000	0.4797	0.6978
SD	-A	-1.09764	-6.705	0.000	0.3337	0.5842
SD	-SA	-1.16020	-6.408	0.000	0.3134	0.5666
D	-SD	0.73463	4.434	0.000	2.0847	1.4330
D	-A	-0.36302	-3.395	0.001	0.6956	0.8371
D	-SA	-0.42557	-3.229	0.001	0.6534	0.8119
A	-SD	1.09764	6.705	0.000	2.9971	1.7118
A	-D	0.36302	3.395	0.001	1.4377	1.1946
SA	-SD	1.16020	6.408	0.000	3.1906	1.7650
SA	-D	0.42557	3.229	0.001	1.5305	1.2317

Variable: male (sd=.49887478)

Odds comparing		b	z	P> z	e^b	e^bStdX
Group 1	vs Group 2					
SD	-SA	1.22645	7.314	0.000	3.4091	1.8438
D	-A	0.46003	4.403	0.000	1.5841	1.2580
D	-SA	1.32672	9.645	0.000	3.7686	1.9384
A	-D	-0.46003	-4.403	0.000	0.6313	0.7949
A	-SA	0.86668	6.611	0.000	2.3790	1.5409
SA	-SD	-1.22645	-7.314	0.000	0.2933	0.5423
SA	-D	-1.32672	-9.645	0.000	0.2653	0.5159
SA	-A	-0.86668	-6.611	0.000	0.4203	0.6490

Variable: white (sd=.32898941)

Odds comparing		b	z	P> z	e^b	e^bStdX
Group 1	vs Group 2					
SD	-SA	0.83423	3.158	0.002	2.3030	1.3158
SA	-SD	-0.83423	-3.158	0.002	0.4342	0.7600

Variable: age (sd=16.779034)

Odds comparing		b	z	P> z	e^b	e^bStdX
Group 1	vs Group 2					
SD	-A	0.02500	5.578	0.000	1.0253	1.5213
SD	-SA	0.03168	6.070	0.000	1.0322	1.7015
D	-A	0.02256	6.789	0.000	1.0228	1.4600
D	-SA	0.02923	6.865	0.000	1.0297	1.6330
A	-SD	-0.02500	-5.578	0.000	0.9753	0.6573
A	-D	-0.02256	-6.789	0.000	0.9777	0.6849
SA	-SD	-0.03168	-6.070	0.000	0.9688	0.5877
SA	-D	-0.02923	-6.865	0.000	0.9712	0.6124

Variable: ed (sd=3.1608267)

Odds comparing		b	z	P> z	e^b	e^bStdX
Group 1	vs Group 2					
SD	-D	-0.09225	-3.374	0.001	0.9119	0.7471
SD	-A	-0.11057	-3.945	0.000	0.8953	0.7051
SD	-SA	-0.14358	-4.251	0.000	0.8663	0.6352
D	-SD	0.09225	3.374	0.001	1.0966	1.3386
A	-SD	0.11057	3.945	0.000	1.1169	1.4183
SA	-SD	0.14358	4.251	0.000	1.1544	1.5743

Variable: prst (sd=14.492259)

Odds comparing		b	z	P> z	e^b	e^bStdX
Group 1	vs Group 2					

b = raw coefficient
z = z-score for test of b=0
P>|z| = p-value for z-test
e^b = exp(b) = factor change in odds for unit increase in X
e^bStdX = exp(b*SD of X) = change in odds for SD increase in X

Using the .01 level of significance (which may be wise given the many comparisons that are being done) we see that white only clearly distinguished between those who strongly agree and those who strongly disagree. Prst does not have any significant effects.

Using `mlogtest` for tests of the Multinomial Logistic Model.

The `mlogtest` command provides a convenient means for testing various hypotheses of interest. Incidentally, keep in mind that `mlogit` can also estimate a logistic regression model; ergo you might sometimes want to use `mlogit` instead of `logit` so you can take advantage of the `mlogtest` command.

Tests of independent variables. `mlogtest` can perform the same tests that `lrdrop1` does for logistic regression, i.e. it can provide likelihood-ratio tests for each variable in the model. To do this yourself, you would have to estimate a series of models, store the results, and then use the `lrtest` command. For example,

```
. qui mlogit warm yr89 male white age ed prst, b(4)
. est store fullmodel
. * Drop yr89 from the model
. qui mlogit warm male white age ed prst, b(4)
. est store yr89
. * Test whether yr89 has sign. effects
. lrtest yr89 fullmodel
```

```
likelihood-ratio test                               LR chi2(3) =      58.85
(Assumption: yr89 nested in fullmodel)             Prob > chi2 =      0.0000
```

We would then repeat the above process for each IV in the model. `mlogtest` can automate this process.

```
. mlogtest, lr
```

```
**** Likelihood-ratio tests for independent variables
```

```
Ho: All coefficients associated with given variable(s) are 0.
```

warm	chi2	df	P>chi2
yr89	58.853	3	0.000
male	106.199	3	0.000
white	11.152	3	0.011
age	83.119	3	0.000
ed	21.087	3	0.000
prst	8.412	3	0.038

From the above, we can see that each variable's effects are significant at the .05 level.

If you happen to have a very large data set or a very complicated model, LR tests can take a long time. It may be sufficient to simply use Wald tests in such cases. Remember, a Wald test only requires the estimation of the constrained model. In Stata, we could just do this with a series of `test` commands:

```
. qui mlogit warm yr89 male white age ed prst, b(4)
```

```
. test yr89
```

```
( 1) [SD]yr89 = 0
( 2) [D]yr89 = 0
( 3) [A]yr89 = 0

      chi2( 3) =    53.81
Prob > chi2 =    0.0000
```

```
. test male
```

```
( 1) [SD]male = 0
( 2) [D]male = 0
( 3) [A]male = 0

      chi2( 3) =    97.77
Prob > chi2 =    0.0000
```

[Repeat for white, age, ed & prst]

Again, `mlogtest`, using the `wald` parameter, can automate the process and also present results more succinctly:

```
. mlogtest, wald
```

```
**** Wald tests for independent variables
```

```
Ho: All coefficients associated with given variable(s) are 0.
```

```

warm |          chi2   df   P>chi2
-----+-----
yr89 |          53.812    3    0.000
male |          97.773    3    0.000
white |         10.783    3    0.013
age   |          79.925    3    0.000
ed    |         20.903    3    0.000
prst  |          8.369    3    0.039
-----+-----
```

Incidentally, putting the LR and Wald results side by side,

LR Tests				Wald Tests			
warm	chi2	df	P>chi2	warm	chi2	df	P>chi2
yr89	58.853	3	0.000	yr89	53.812	3	0.000
male	106.199	3	0.000	male	97.773	3	0.000
white	11.152	3	0.011	white	10.783	3	0.013
age	83.119	3	0.000	age	79.925	3	0.000
ed	21.087	3	0.000	ed	20.903	3	0.000
prst	8.412	3	0.038	prst	8.369	3	0.039

We see that both tests lead to very similar conclusions in this case. That is fairly common; it seems they are most likely to differ in borderline cases.

You can also use `mlogtest` to test sets of variables, e.g.

```
. mlogtest, lr set(white prst \ white ed \ yr89 male )
```

```
*** Likelihood-ratio tests for independent variables
```

```
Ho: All coefficients associated with given variable(s) are 0.
```

warm	chi2	df	P>chi2
yr89	58.853	3	0.000
male	106.199	3	0.000
white	11.152	3	0.011
age	83.119	3	0.000
ed	21.087	3	0.000
prst	8.412	3	0.038
set_1: white prst	19.282	6	0.004
set_2: white ed	30.334	6	0.000
set_3: yr89 male	167.621	6	0.000

Sidelight: BIC tests as an alternative. Paul Millar's `bicdrop1` routine provides a convenient means of doing BIC tests of whether individual variables should be dropped from the model. It works after several commands, including `mlogit`:

```
. bicdrop1
```

```
BIC Difference Tests: drop 1 term
```

```
mlogit regression
```

```
number of obs = 2293
```

	warm	df	-2*log ll	AIC	BICprime	BIC	BICdiff	prob
Full Model	21		5642.00	5684.00	-210.27	-11937.87		
-yr89		18	5700.85	5736.85	-174.63	-11902.23	35.6	0.000
-male		18	5748.20	5784.20	-127.28	-11854.88	83.0	0.000
-white		18	5653.15	5689.15	-222.33	-11949.93	-12.1	0.998
-age		18	5725.12	5761.12	-150.36	-11877.96	59.9	0.000
-ed		18	5663.08	5699.08	-212.39	-11939.99	-2.1	0.743
-prst		18	5650.41	5686.41	-225.07	-11952.67	-14.8	0.999

```
Terms dropped one at a time in turn.
```

A positive value for `BICdiff` indicates that the term should not be dropped, whereas a negative term suggests that perhaps it should. In this case, the BIC test considers `white`, `ed` and `prst` as candidates for deletion. The LR and Wald tests say to keep these variables, but they are the least significant ones.

Tests for combining dependent categories. If none of the IVs significantly affects the odds of outcome m versus outcome n, we say that m and n are indistinguishable with respect to the variables in the model. If two outcomes are indistinguishable with respect to the variables in the model, you can obtain more efficient estimates by combining them. Again, you can use both Stata or spost9 commands, and you can do LR or Wald tests.

```
. mlogtest, lrcomb
```

```
**** LR tests for combining outcome categories
```

```
Ho: All coefficients except intercepts associated with given pair
of outcomes are 0 (i.e., categories can be collapsed).
```

Categories tested		chi2	df	P>chi2
SD-	D	43.864	6	0.000
SD-	A	153.130	6	0.000
SD-	SA	215.033	6	0.000
D-	A	98.857	6	0.000
D-	SA	191.730	6	0.000
A-	SA	54.469	6	0.000

To do this with regular Stata commands, you first estimate the unconstrained model and store the results; then, you estimate constrained models where the coefficients for one category (except the intercept) are constrained to equal 0. For example, to test whether Agree & Strongly Agree can be combined,

```
. qui mlogit warm yr89 male white age ed prst, b(4)
. est store fullmodel
. * see if we need agree as a separate category, or if it can be combined with SA
. constraint 1 [A]
. qui mlogit warm yr89 male white age ed prst, b(4) constraint(1)
. est store no_A
. lrtest fullmodel no_A
```

```
likelihood-ratio test                                LR chi2(6) =      54.47
(Assumption: no_A nested in fullmodel)              Prob > chi2 =      0.0000
```

Based on the above, we see that no categories should be combined. Doing the same thing with the simpler Wald tests,

```
. qui mlogit warm yr89 male white age ed prst, b(4)
. mlogtest, combine
```

**** Wald tests for combining outcome categories

Ho: All coefficients except intercepts associated with given pair of outcomes are 0 (i.e., categories can be collapsed).

Categories tested		chi2	df	P>chi2
SD-	D	41.018	6	0.000
SD-	A	135.960	6	0.000
SD-	SA	183.910	6	0.000
D-	A	93.183	6	0.000
D-	SA	167.439	6	0.000
A-	SA	51.441	6	0.000

Or, doing it the hard way (I am deleting some of the less important output)

```
. test [SD=D]
```

```
( 1) [SD]yr89 - [D]yr89 = 0
( 2) [SD]male - [D]male = 0
( 3) [SD]white - [D]white = 0
( 4) [SD]age - [D]age = 0
( 5) [SD]ed - [D]ed = 0
( 6) [SD]prst - [D]prst = 0
```

```
      chi2( 6) =    41.02
Prob > chi2 =    0.0000
```

```
. test [SD=A]
```

```
      chi2( 6) =   135.96
Prob > chi2 =    0.0000
```

```
. test [SD]
```

```
( 1) [SD]yr89 = 0
( 2) [SD]male = 0
( 3) [SD]white = 0
( 4) [SD]age = 0
( 5) [SD]ed = 0
( 6) [SD]prst = 0
```

```
      chi2( 6) =   183.91
Prob > chi2 =    0.0000
```

```
. test [D=A]
```

```
      chi2( 6) =    93.18
Prob > chi2 =    0.0000
```

```
. test [D]
```

```
      chi2( 6) =   167.44
Prob > chi2 =    0.0000
```

```
. test [A]
```

```
      chi2( 6) =    51.44
Prob > chi2 =    0.0000
```

Note that, when only a single category name is given, e.g. `test [A]`, you are testing whether the given category can be combined with the reference category (in this case SA, strongly agree). If two category names are given, e.g. `test [D=A]`, you are testing whether those 2 categories can be combined.

Measures of Fit. The `fitstat` command can be used the same as before, e.g.

```
. quietly mlogit warm yr89 male white age ed prst
. quietly fitstat, save
. * Now drop prst, white & ed, the three least significant vars
. quietly mlogit warm yr89 male age
. fitstat, dif
```

Measures of Fit for mlogit of warm

	Current	Saved	Difference
Model:	mlogit	mlogit	
N:	2293	2293	0
Log-Lik Intercept Only:	-2995.770	-2995.770	0.000
Log-Lik Full Model:	-2848.592	-2820.998	-27.594
D:	5697.184(2281)	5641.996(2272)	55.188(9)
LR:	294.357(9)	349.544(18)	55.188(9)
Prob > LR:	0.000	0.000	0.000
McFadden's R2:	0.049	0.058	-0.009
McFadden's Adj R2:	0.045	0.051	-0.006
Maximum Likelihood R2:	0.120	0.141	-0.021
Cragg & Uhler's R2:	0.130	0.153	-0.023
Count R2:	0.412	0.424	-0.013
Adj Count R2:	0.061	0.081	-0.020
AIC:	2.495	2.479	0.016
AIC*n:	5721.184	5683.996	37.188
BIC:	-11952.319	-11937.868	-14.451
BIC' :	-224.718	-210.267	-14.451

Difference of 14.451 in BIC' provides very strong support for current model.

Note: p-value for difference in LR is only valid if models are nested.

Incidentally, note that the chi-square and AIC tests favor the full model; however, the BIC test prefers the model that drops the least significant variables, `prst`, `white` & `ed`. As we have seen before, the BIC test tends to lead to more parsimonious models, especially when the sample size is large.

Aids to Interpretation. These are much the same as we talked about before. Standardized coefficients, however, are a noteworthy exception:

```
. listcoef, std
option std not allowed after mlogit
r(198);
```

This is because the y^* rationale does not hold in a multinomial logit model, i.e. there is no underlying latent variable. (As we saw earlier, however, the `listcoef` command will still do X-standardization.)

Other commands, however, behave identically or almost identically to what we have seen before. For example, we can use the `predict` command to come up with predicted probabilities:

```
. quietly mlogit warm yr89 male white age ed prst, b(4)
. predict SDlogit Dlogit Alogit SAllogit
(option p assumed; predicted probabilities)

. list warm yr89 male white age ed prst SDlogit Dlogit Alogit SAllogit in 1/10, clean
```

	warm	yr89	male	white	age	ed	prst	SDlogit	Dlogit	Alogit	SAllogit
1.	SD	1977	Women	Not Whit	33	10	31	.14696	.2569168	.375222	.2209013
2.	SD	1977	Men	Not Whit	74	16	50	.1931719	.4962518	.2510405	.0595358
3.	SD	1989	Men	Not Whit	36	12	41	.074012	.3257731	.4686748	.1315401
4.	SD	1977	Women	Not Whit	73	9	36	.277139	.383207	.2358743	.1037797
5.	SD	1977	Women	Not Whit	59	11	62	.2066857	.2824558	.3317693	.1790893
6.	SD	1989	Men	Not Whit	33	4	17	.1461631	.383301	.3885765	.0819594
7.	SD	1977	Women	Not Whit	43	7	40	.2276894	.2719278	.3321202	.1682626
8.	SD	1977	Women	Not Whit	48	12	48	.1571982	.2740046	.358632	.2101651
9.	SD	1977	Men	Not Whit	27	17	69	.0970773	.259278	.4779711	.1656736
10.	SD	1977	Men	Not Whit	46	12	50	.1997817	.3800453	.3360028	.0841702

The extremes (use `findit extremes`) command helps you to see who is most likely and least likely to be predicted to strongly disagree:

```
. extremes SDlogit warm yr89 male white age ed prst
```

```
+-----+
| obs:   SDlogit  warm  yr89  male  white  age  ed  prst |
+-----+
| 1214.  .0078837   A   1989  Women  White  27  20  68 |
| 2048.  .0115555   SA  1989  Women  White  26  17  52 |
| 2241.  .0127511   SA  1989  Women  White  21  15  61 |
| 1855.  .0131329   A   1989  Women  White  25  16  36 |
| 803.   .0142298   D   1989  Women  White  30  16  60 |
+-----+

+-----+
| 612.   .4276913   D   1977  Men   Not Whit  80  5  45 |
| 171.   .4289597   SD  1977  Men   Not Whit  67  3  32 |
| 282.   .4378463   SD  1977  Men   Not Whit  68  3  37 |
| 87.    .4426529   SD  1977  Men   Not Whit  83  5  51 |
| 863.   .479314    D   1977  Men   Not Whit  54  0  40 |
+-----+
```

Based on the results, we see that fairly young white women in 1989 with high levels of education and occupational prestige were predicted to be the least likely to strongly disagree. Conversely, nonwhite elderly males in 1977 with low levels of education and generally low levels of occupational prestige had almost a 50% predicted probability of strongly disagreeing.

You can use the `prvalue` command to come up with predicted values for representative types:

Working Class Men in 1977 who are near retirement:

```
. prvalue, x(yr89 = 0 male = 1 prst =20 age = 64 ed=12) rest(mean)
```

```
mlogit: Predictions for warm
```

```
Predicted probabilities for each category:
```

```
Pr(y=SD|x):      0.2084
Pr(y=D|x):       0.5213
Pr(y=A|x):       0.2218
Pr(y=SA|x):      0.0486
```

```
          yr89      male      white      age      ed      prst
x=         0         1      .8765809      64      12      20
```

When we did the same thing after `ologit`, the percentages were 28.3%, 42.9%, 23.1%, and 5.8%. So, the `mlogit` model has this group a little more in the middle categories.

Young, highly educated women in 1989 with prestigious jobs

```
. prvalue, x(yr89=1 male = 0 prst = 80 age = 30 ed = 20) rest(mean)
```

```
mlogit: Predictions for warm
```

```
Predicted probabilities for each category:
```

```
Pr(y=SD|x):      0.0148
Pr(y=D|x):       0.0935
Pr(y=A|x):       0.4448
Pr(y=SA|x):      0.4469
```

```
          yr89      male      white      age      ed      prst
x=         1         0      .8765809      30      20      80
```

After `ologit`, the predicted values were 2.1%, 9.9%, 35.5%, and 52.5%. So again we see that `mlogit` is moving the predictions more toward the middle categories.

As before, the `prtab` command provides another way of coming up with predicted values for various combinations of outcomes. (I've rearranged the output a little bit).

```
. prtab yr89 male, novarlbl
```

```
mlogit: Predicted probabilities for warm
```

Predicted probability of outcome 1 (SD)			Predicted probability of outcome 2 (D)		
yr89	male		yr89	male	
	Women	Men		Women	Men
1977	0.1370	0.1796	1977	0.2786	0.4038
1989	0.0545	0.0785	1989	0.2311	0.3681

Predicted probability of outcome 3 (A)			Predicted probability of outcome 4 (SA)		
yr89	male		yr89	male	
	Women	Men		Women	Men
1977	0.3617	0.3309	1977	0.2228	0.0857
1989	0.4314	0.4337	1989	0.2830	0.1196

```
x=      yr89      male      white      age      ed      prst
      .39860445  .46489315  .8765809  44.935456  12.218055  39.585259
```

Throughout, you can see that men are less likely than women to believe that a working mother can have just as warm a relationship. Between 1977 and 1989, both men and women developed more positive attitudes toward working women.

The `prchange` command provides another way of illustrating how changes in the X variables affect predicted probabilities:

```
. set matsize 100
. prchange

mlogit: Changes in Predicted Probabilities for warm

yr89
      Avg|Chg|      SD      D      A      SA
0->1   .0685527  -.09277321  -.04433218  .08874959  .0483558

male
      Avg|Chg|      SD      D      A      SA
0->1   .08402095  .03498483  .13305706  -.01899102  -.14905089

white
      Avg|Chg|      SD      D      A      SA
0->1   .03699843  .0479169  .02607995  -.01173434  -.06226252

age
      Avg|Chg|      SD      D      A      SA
Min->Max .20911962  .12694021  .29129903  -.24914826  -.16909096
  +1/2   .00310007  .00181469  .00438544  -.00357869  -.00262144
  +sd/2  .05183155  .03035958  .07330352  -.05978927  -.04387383
MargEfct .0031001  .0018147  .0043855  -.00357874  -.00262146

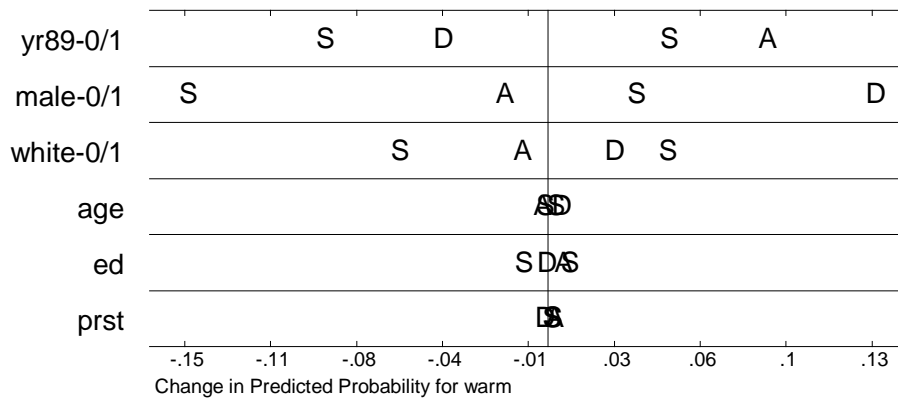
ed
      Avg|Chg|      SD      D      A      SA
Min->Max .13736561  -.27402139  -.00070983  .12796843  .1467628
  +1/2   .00640956  -.01111165  -.00170746  .00515014  .00766897
  +sd/2  .02026453  -.03517523  -.00535381  .01629332  .02423576
MargEfct .00640937  -.01110977  -.00170898  .00514963  .00766912

prst
      Avg|Chg|      SD      D      A      SA
Min->Max .08374692  .00821991  -.16749384  .09694412  .06232981
  +1/2   .00124198  .00014042  -.00248396  .00144508  .00089847
  +sd/2  .01799244  .00203484  -.03598487  .02093393  .0130161
MargEfct .00124198  .00014042  -.00248396  .00144509  .00089846

      SD      D      A      SA
Pr(y|x) .11394239  .32539049  .39422345  .16644368

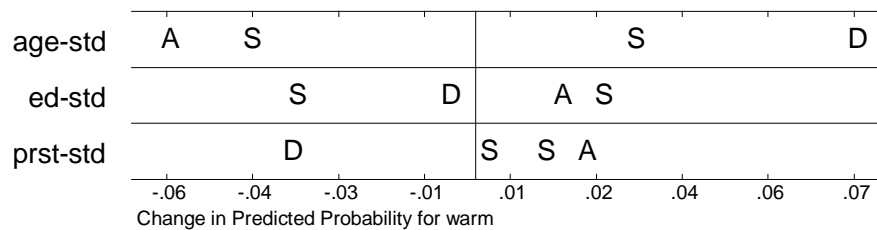
      yr89   male   white   age   ed   prst
x=   .398604  .464893  .876581  44.9355  12.2181  39.5853
sd(x)= .489718  .498875  .328989  16.779  3.16083  14.4923
```

A graphical representation of the above information can be obtained by using `mlogview`. Typing `mlogview` brings up a dialogue box, which in turn generates syntax for you. (The easiest way to learn how to use it is just to try it and see. I generated the following graphic by requesting that the amount of change produced by a one-unit change in each variable (holding other variables at their means) be plotted:



So, for example, you see that going from 1977 to 1989 reduces the probability of Strongly Disagreeing by about 9%, reduces the probability of Disagreeing by about 4%, increases the probability of strongly agreeing by 5%, and the probability of agreeing by about 9%. These are the same numbers that `prchange` produced. You also see clear effects of being male versus female and of being white versus non-white. A one-year change in age or education, or a one-unit change in prestige, has little effect on the predicted probabilities; but these are fairly small units of change.

Graphs like the above may be helpful for comparing the effects of different variables, although it may be more useful when either (a) the variables have similar scales, or (b) you look at the effect of a standard deviation change rather than a 1-unit change. For example, here I show the plot for a one standard deviation change in age, education and `prst`:



Long and Freese discuss several other possible uses of the `mlogview` command. I think you mostly just need to play around with it to see what you find useful.