Title

Intro 5 — Models for discrete choices

Description

This introduction covers the commands `cmclogit`, `cmmixlogit`, `cmmprobit`, and `nlogit`. These estimation commands fit discrete choice models, that is, models in which each decision maker chooses a single alternative from a finite set of available alternatives.

Remarks and examples

Remarks are presented under the following headings:

Overview of CM commands for discrete choices

Stata has four commands designed for fitting discrete choice models. Here we give you a brief overview of the similarities and differences in the models fit by these commands.

Each of these commands allows both alternative-specific and case-specific predictors, and each one handles unbalanced choice sets properly. Each of these models can be derived as a random utility model in which each decision maker selects the alternative that provides the highest utility. See [CM] **Intro 8** for more information on the random utility model formulation of these discrete choice models.

The difference in these models largely hinges on an assumption known as independence of irrelevant alternatives (IIA). Briefly, the IIA assumption means that relative probability of selecting alternatives should not change if we introduce or eliminate another alternative. As an example, suppose that a restaurant has one chicken entree and one steak entree on the menu and that these are equally likely to be selected. If a vegetarian option is introduced, the probabilities of selecting chicken and steak will both decrease, but they should still be equal to each other if the IIA assumption holds. If the probability of selecting steak now is greater than the probability of selecting chicken, or vice versa, the IIA assumption does not hold. More technically, the IIA assumption means that the error terms cannot be correlated across alternatives. See [CM] **Intro 8** for more information on this assumption and how it applies to each choice model.

cmclogit fits McFadden's choice model using conditional logistic regression. Of the four models discussed in this entry, McFadden's choice model has the most straightforward formulation. However, it does require that you make the IIA assumption.

cmmixlogit fits a mixed logit regression for choice models. This model allows random coefficients on one or more of the alternative-specific predictors in the model. This means that the coefficients on these variables are allowed to vary across individuals. We do not estimate the coefficients for each individual. Instead, we assume that the coefficients follow a distribution such as normal distribution, and we estimate the parameters of that distribution. Through these random coefficients, the model allows correlation across alternatives. In this way, the mixed logit model relaxes the IIA assumption.

cmmprobit fits a multinomial probit choice model. Like cmclogit, this command estimates fixed coefficients for all predictors, but it relaxes the IIA assumption in another way. It directly models the correlation between the error terms for the different alternatives.

nlogit fits a nested logit choice model. With this model, similar alternatives—alternatives whose errors are likely to be correlated—can be grouped into nests. Extending our restaurant example, suppose there are now seven entrees. Three include chicken, two include steak, and two are vegetarian. The researcher could specify a nesting structure where entrees are grouped by type. The nested logit model then accounts for correlation of alternatives within the same nest and thus relaxes the IIA assumption.

Below, we provide further introductions to these models, demonstrate how to fit and interpret them using Stata, and tell you more about their relationships with each other and with other Stata estimation commands.

cmclogit: McFadden's choice model

McFadden's choice model is fit using conditional logistic regression. In Stata, this model can also be fit by the command clogit. In fact, cmclogit calls clogit to fit McFadden's choice model. However, cmclogit is designed for choice data and has features that clogit does not. cmclogit properly handles missing values for choice models, checks for errors in the alternatives variable and case-specific variables, and has appropriate postestimation commands such as the special version of margins designed for use after CM estimation.

To demonstrate cmclogit, we use the same dataset we used in [CM] **Intro 2**. We load the data, list the first three cases, and use cmset.

```
. use https://www.stata-press.com/data/r18/carchoice
(Car choice data)

. list consumerid car purchase gender income dealers if consumerid <= 3,
> sepby(consumerid) abbrev(10)

     +------------------------------------------------------------------+
     | consumerid        car   purchase   gender   income   dealers |
     |------------------------------------------------------------------|
  1. |          1   American          1     Male     46.7         9 |
  2. |          1   Japanese          0     Male     46.7        11 |
  3. |          1   European          0     Male     46.7         5 |
  4. |          1     Korean          0     Male     46.7         1 |
     |------------------------------------------------------------------|
  5. |          2   American          1     Male     26.1        10 |
  6. |          2   Japanese          0     Male     26.1         7 |
  7. |          2   European          0     Male     26.1         2 |
  8. |          2     Korean          0     Male     26.1         1 |
     |------------------------------------------------------------------|
  9. |          3   American          0     Male     32.7         8 |
 10. |          3   Japanese          1     Male     32.7         6 |
 11. |          3   European          0     Male     32.7         2 |
     +------------------------------------------------------------------+

. cmset consumerid car
note: alternatives are unbalanced across choice sets; choice sets of
      different sizes found.

     Case ID variable: consumerid
Alternatives variable: car
```

We passed `cmset` the case ID variable `consumerid` and the alternatives variable `car`, which contains possible choices of the nationality of car purchased, American, Japanese, European, or Korean.

The 0/1 variable `purchase` indicates which nationality of car was purchased. It is our dependent variable for `cmclogit`. Before we fit our model, let's run `cmtab` to see the observed choices in the data.

```
. cmtab, choice(purchase)

Tabulation of chosen alternatives (purchase = 1)

Nationality |
     of car |      Freq.     Percent        Cum.
------------+-----------------------------------
   American |        384       43.39       43.39
   Japanese |        326       36.84       80.23
   European |        135       15.25       95.48
     Korean |         40        4.52      100.00
------------+-----------------------------------
      Total |        885      100.00
```

Most of the people in these data purchased American cars (43%), followed by Japanese cars (37%) and European cars (15%). Korean cars were purchased the least (5%).

For predictors, we have the case-specific variables gender and income, and the alternative-specific variable dealers, which contains the number of dealerships of each nationality in the consumer's community. We fit the model:

```
. cmclogit purchase dealers, casevars(i.gender income)
Iteration 0:   Log likelihood = -959.21405
Iteration 1:   Log likelihood = -948.48587
Iteration 2:   Log likelihood =  -948.1217
Iteration 3:   Log likelihood = -948.12096
Iteration 4:   Log likelihood = -948.12096
Conditional logit choice model                  Number of obs      =     3,075
Case ID variable: consumerid                    Number of cases    =       862
Alternatives variable: car                      Alts per case: min =         3
                                                               avg =       3.6
                                                               max =         4
                                                   Wald chi2(7)    =     51.03
Log likelihood = -948.12096                        Prob > chi2     =    0.0000
```

purchase	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
car						
dealers	.0448082	.0262818	1.70	0.088	-.0067032	.0963196
American	(base alternative)					
Japanese						
gender						
Male	-.379326	.1712399	-2.22	0.027	-.71495	-.0437021
income	.0154978	.0065145	2.38	0.017	.0027296	.0282659
_cons	-.4787261	.331378	-1.44	0.149	-1.128215	.1707628
European						
gender						
Male	.653345	.2647694	2.47	0.014	.1344065	1.172283
income	.0343647	.0080286	4.28	0.000	.0186289	.0501006
_cons	-2.839606	.461613	-6.15	0.000	-3.744351	-1.934861
Korean						
gender						
Male	.0679233	.4464535	0.15	0.879	-.8071094	.942956
income	-.0377716	.0158434	-2.38	0.017	-.068824	-.0067191
_cons	.0511728	.8033048	0.06	0.949	-1.523276	1.625621

Note that alternative-specific variables (if any) follow the dependent variable. Case-specific variables (if any) are placed in the option casevars(). Because cmclogit requires us to specify which variables are alternative specific and which are case specific, it can verify that our data are coded as we expect. It checks whether the specified case-specific variables are truly case specific. If they are not, we get an error.

You may also see messages from cmclogit about the alternative-specific variables. For example,

```
note: variable dealers has 2 cases that are not alternative-specific; there is
      no within-case variability.
```

Alternative-specific variables can vary by alternative and by case, but they do not have to vary by alternative for every case. This message tells us that there are two cases for which the alternative-specific variable is constant within case. If an alternative-specific variable is constant within case for a large proportion of the cases, we might question how alternative specific that variable really is and be concerned about its predictive value. If a variable that is supposed to be alternative specific is in fact case specific, we will get an error.

Looking at the results from `cmclogit`, we first see that the coefficient on `dealers` is positive; based on this model, we expect the probability of purchasing a vehicle of any nationality to increase as the number of dealerships increases. However, notice that this coefficient is different from 0 at the 10% level but not at the 5% level.

American cars are chosen as the base alternative, so coefficients on the alternative-specific variables are interpreted relative to them. For instance, for the Japanese alternative, the coefficient on `Male` is negative, which indicates that males are less likely to select a Japanese car than an American car.

Looking at cases with missing values using cmsample

From the header of the `cmclogit` output, we see that our model was fit using 862 cases in our model. However, we see from the previous `cmtab` output that there are a total of 885 cases in the data. There must be missing values in one or more of the variables. Let's track down the variables and the cases with missing values using `cmsample`. First, we run `cmsample` specifying all the variables we used with `cmclogit`. The only difference is that the dependent variable goes in the `choice()` option.

```
. cmsample dealers, choice(purchase) casevars(i.gender income)
```

Reason for exclusion	Freq.	Percent	Cum.
observations included	3,075	97.31	97.31
casevars missing	85	2.69	100.00
Total	3,160	100.00	

The results tell us that the missing values are in the *casevars*, either `gender` or `income` or both. Note that the tabulation produced by `cmsample` shows counts of observations not cases.

Second, we look at `gender` alone with `cmsample`:

```
. cmsample, casevars(i.gender) generate(flag)
```

Reason for exclusion	Freq.	Percent	Cum.
observations included	3,075	97.31	97.31
casevar missing	85	2.69	100.00
Total	3,160	100.00	

These are the cases with missing values. We also specified the `generate()` option to create a variable whose nonzero values indicate cases with missing values or other problems. We list these cases:

```
. sort consumerid car
. list consumerid car gender flag if flag != 0, sepby(consumerid) abbr(10)
```

	consumerid	car	gender	flag
509.	142	American	.	casevar missing
510.	142	Japanese	Male	casevar missing
511.	142	European	Male	casevar missing
512.	142	Korean	Male	casevar missing
516.	144	American	.	casevar missing
517.	144	Japanese	Male	casevar missing
518.	144	European	Male	casevar missing

(output omitted)

We could have listed the observations with missing values of `gender` by typing `list if missing(gender)`. But using `cmsample` in this way allows us to list entire cases, potentially giving us a way to fix the problem. In this example, we could decide all the nonmissing values of `gender` are valid and fill in the missing values with the nonmissing ones for that case. However, we will not do this for the purpose of our example.

See [CM] **cmsample** and example 3 in [CM] **cmclogit** for more on missing values in choice data.

margins after CM estimation

Above, we interpreted a few of the coefficients from the `clogit` results. In [CM] **Intro 1**, we showed you that you can use `margins` to further interpret the results of your choice model. Here we demonstrate how we can apply some of `margins` special choice model features to interpret the results of this model.

First, we type `margins` without any arguments to get the average predicted probabilities for the different alternatives.

```
. margins

Predictive margins                                      Number of obs = 3,075
Model VCE: OIM

Expression: Pr(car|1 selected), predict()

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
    _outcome |
    American |   .4361949    .016755    26.03   0.000     .4033556    .4690342
    Japanese |   .3665893   .0162405    22.57   0.000     .3347585    .3984202
    European |   .1508121   .0119794    12.59   0.000     .1273328    .1742913
      Korean |   .0464037   .0069301     6.70   0.000      .032821    .0599865
------------------------------------------------------------------------------
```

Based on this model and assuming we have a random or otherwise representative sample, these are the expected proportions in the population.

`margins` can produce many types of estimates. Suppose we want to know how the probability of a person selecting a European car changes when the number of European dealerships increases. If this probability increases (as we expect it to), the increase must come at the expense of American, Japanese, or Korean cars. Which one of these is affected the most?

First, let's estimate the expected probability of purchasing each nationality of car if each community adds a new European dealership. We can use the `at(dealers=(dealers+1)` option to request this computation.

```
. margins, at(dealers=generate(dealers+1)) alternative(European)
Predictive margins                                     Number of obs = 3,075
Model VCE: OIM

Expression:  Pr(car|1 selected), predict()
Alternative: European
At: dealers = dealers+1

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
    _outcome |
    American |   .4333003   .0168164    25.77   0.000     .4003407    .4662598
    Japanese |   .3641274   .0162604    22.39   0.000     .3322577    .3959971
    European |   .1564365   .0127751    12.25   0.000     .1313978    .1814752
      Korean |   .0461358   .0068959     6.69   0.000     .0326201    .0596516
------------------------------------------------------------------------------
```

These look similar to the expected probabilities we estimated using the original number of dealerships in each community. By using the `contrast()` option, we can estimate the differences between these probabilities and the original ones. We include the `nowald` option to simplify the output.

```
. margins, at(dealers=generate(dealers)) at(dealers=generate(dealers+1))
> alternative(European) contrast(atcontrast(r) nowald)
Contrasts of predictive margins                        Number of obs = 3,075
Model VCE: OIM

Expression:  Pr(car|1 selected), predict()
Alternative: European

1._at: dealers =   dealers

2._at: dealers = dealers+1

--------------------------------------------------------------------
                    |            Delta-method
                    |   Contrast   std. err.     [95% conf. interval]
--------------------+-----------------------------------------------
        _at@_outcome |
(2 vs 1) American   |  -.0028946   .0017268     -.0062791    .0004899
(2 vs 1) Japanese   |  -.0024619   .0014701     -.0053434    .0004195
(2 vs 1) European   |   .0056244   .0033521     -.0009456    .0121944
  (2 vs 1) Korean   |  -.0002679   .0001686     -.0005983    .0000625
--------------------------------------------------------------------
```

Increasing the number of European dealerships by one increases the expected probability of selecting a European car by 0.0056. This increase comes at the expense of American cars slightly more than Japanese cars. The probability of someone purchasing an American car decreases by 0.0029, and the probability of someone purchasing a Japanese car decreases by 0.0025. The probability of buying a Korean car is barely changed, only a tiny decrease of 0.0003 in the probability. All of these changes are very small. We can look at the 95% confidence intervals to see that none of these changes in probabilities is significantly different from 0 at the 5% level.

We will ignore the lack of significance for now and explore one of `margins`'s features specific to choice models. As we mentioned before, the choice sets are unbalanced. Some consumers do not have the choice of a Korean car (corresponding to `car == 4`) as one of their available alternatives.

```
. cmchoiceset
Tabulation of choice-set possibilities
 Choice set |      Freq.     Percent        Cum.
------------+-----------------------------------
      1 2 3 |        380       42.94       42.94
    1 2 3 4 |        505       57.06      100.00
------------+-----------------------------------
      Total |        885      100.00
Note: Total is number of cases.
```

How does margins handle the fact that some persons do not have the choice of Korean cars among their alternatives? By default, margins sets the probability of buying a Korean car for these consumers to zero and keeps it fixed at zero.

If we want to look at only those consumers who have Korean in their choice set, we can use the outcome(... , altsubpop) option.

```
. margins, at(dealers=generate(dealers)) at(dealers=generate(dealers+1))
> alternative(European) contrast(atcontrast(r) nowald) outcome(Korean, altsubpop)
Contrasts of predictive margins                          Number of obs = 3,075
Model VCE: OIM
Expression:  Pr(car|1 selected), predict()
Alternative: European
Outcome:     Korean
1._at: dealers =   dealers
2._at: dealers = dealers+1

--------------------------------------------------------------
             |            Delta-method
             |   Contrast   std. err.     [95% conf. interval]
-------------+------------------------------------------------
         _at |
   (2 vs 1)  |  -.0004722   .0002972     -.0010547    .0001103
--------------------------------------------------------------
```

The probability of buying a Korean car among those who have the choice of buying a Korean decreases by 0.0005 when a European dealership is added. This change is bigger than what we estimated earlier, as we expect, because we omitted all those persons whose change was fixed at zero.

When we model these data, it seems reasonable to keep the probability of buying a Korean car fixed at zero for those consumers who do not have Korean in their choice set. The result gives a picture of the total population represented by the sample; to omit them gives a picture of only those communities with Korean dealerships. See [CM] **margins** for more examples and another discussion of this issue.

If you have not already read [CM] **Intro 1**, we recommend that you also read the examples of interpreting results of cm commands using margins that are provided in that entry. For more information on margins, see its main entry in the Stata manuals, [R] **margins**. You will also want to see the separate entry for it in this manual, [CM] **margins**, which describes the special features of this command when used after cm commands and includes lots of choice model examples.

cmmixlogit: Mixed logit choice models

cmmixlogit fits a mixed logit regression for choice data. Like cmclogit, cmmixlogit is used to model the probability that a decision maker chooses one alternative from a set of available alternatives.

In the mixed logit model, the coefficients on alternative-specific variables can be treated as fixed or random. Specifying random coefficients can model correlation of choices across alternatives, thereby relaxing the IIA property that is imposed by McFadden's choice model. In this sense, the mixed logit model fit by cmmixlogit is more general than models fit by cmclogit. McFadden and Train (2000) show that the mixed logit model can approximate a wide class of choice representations. See [CM] **Intro 8** for a description of the IIA property and how mixed logit models can fit deviations from it.

We continue with the same dataset we have been using in this introduction: consumer data on choices of nationalities of cars. The data arrangement required by cmmixlogit is exactly the same as that for cmclogit.

Mixed logit choice models can fit random coefficients for alternative-specific variables. We take dealers, the number of dealers of each nationality in each consumer's community, which is an alternative-specific variable, and fit random coefficients for it.

```
. cmmixlogit purchase, random(dealers) casevars(i.gender income)
note: alternatives are unbalanced.

Fitting fixed parameter model:

Fitting full model:

Iteration 0:  Log simulated-likelihood = -966.81349
Iteration 1:  Log simulated-likelihood = -949.54388
Iteration 2:  Log simulated-likelihood = -948.15757
Iteration 3:  Log simulated-likelihood = -948.12546
Iteration 4:  Log simulated-likelihood = -948.12106
Iteration 5:  Log simulated-likelihood = -948.12099
Iteration 6:  Log simulated-likelihood = -948.12097
Iteration 7:  Log simulated-likelihood = -948.12096

Mixed logit choice model                       Number of obs      =      3,075
Case ID variable: consumerid                   Number of cases    =        862

Alternatives variable: car                     Alts per case: min =          3
                                                              avg =        3.6
                                                              max =          4

Integration sequence:      Hammersley
Integration points:               623             Wald chi2(7)    =      51.03
Log simulated-likelihood = -948.12096             Prob > chi2     =     0.0000

------------------------------------------------------------------------------
    purchase | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
car          |
     dealers |   .0448203   .0262821     1.71   0.088    -.0066917    .0963323
-------------+----------------------------------------------------------------
/Normal      |
 sd(dealers) |   .0001994   .1981032                             .           .
-------------+----------------------------------------------------------------
American     |  (base alternative)
-------------+----------------------------------------------------------------
Japanese     |
      gender |
       Male  |  -.3793276   .1712401    -2.22   0.027     -.714952   -.0437032
      income |    .015498   .0065144     2.38   0.017       .00273    .0282661
       _cons |  -.4786729   .3313762    -1.44   0.149    -1.128158    .1708125
-------------+----------------------------------------------------------------
European     |
      gender |
       Male  |   .6533193   .2647746     2.47   0.014     .1343706    1.172268
      income |   .0343656   .0080288     4.28   0.000     .0186295    .0501017
       _cons |  -2.839604   .4616206    -6.15   0.000    -3.744363   -1.934844
-------------+----------------------------------------------------------------
Korean       |
      gender |
       Male  |   .0676844   .4464111     0.15   0.879    -.8072653    .9426341
      income |  -.0377614   .0158428    -2.38   0.017    -.0688128     -.00671
       _cons |   .0511088   .8032683     0.06   0.949    -1.523268    1.625486
------------------------------------------------------------------------------
LR test vs. fixed parameters: chibar2(01) =      0.00  Prob >= chibar2 = 0.5000
```

The estimated standard deviation for the random coefficient is small, and the likelihood-ratio test shown at the bottom of the table that compares this random-coefficients model with a fixed-coefficient model is not significant. A model with random coefficients for `dealers` is no better than one with a fixed coefficient. Note that this fixed-coefficient model is precisely the model fit earlier by `cmclogit`.

We used the default distribution for the random coefficients: a normal (Gaussian) distribution. Let's fit the model again using a lognormal distribution for the coefficient of `dealers`.

Because the lognormal distribution is only defined over positive real values, the coefficient values coming from this distribution will only be positive. This constrains the coefficient to be positive. Is this constraint okay? We believe that increasing the number of dealerships in a community of a given nationality should always increase the probability that someone in the community buys that type of car and never decrease the probability. So constraining the coefficient to be positive is what we want. (If we want to constrain the coefficient to be negative, we could create a variable equal to `-dealers` and fit a random lognormal coefficient for it.)

```
. cmmixlogit purchase, random(dealers, lognormal) casevars(i.gender income)
note: alternatives are unbalanced.

Fitting fixed parameter model:

Fitting full model:

Iteration 0:  Log simulated-likelihood = -948.13062
Iteration 1:  Log simulated-likelihood =  -948.1226
Iteration 2:  Log simulated-likelihood = -948.12155
Iteration 3:  Log simulated-likelihood = -948.12106
Iteration 4:  Log simulated-likelihood = -948.12096
Iteration 5:  Log simulated-likelihood = -948.12096

Mixed logit choice model                       Number of obs      =      3,075
Case ID variable: consumerid                   Number of cases    =        862

Alternatives variable: car                     Alts per case: min =          3
                                                              avg =        3.6
                                                              max =          4

Integration sequence:      Hammersley
Integration points:               623             Wald chi2(7)    =      79.14
Log simulated-likelihood = -948.12096             Prob > chi2     =     0.0000

------------------------------------------------------------------------------
    purchase | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
car          |
     dealers |  -3.105499   .5869861    -5.29   0.000    -4.255971   -1.955028
-------------+----------------------------------------------------------------
/Lognormal   |
 sd(dealers) |   .0036636   4.480108                             .           .
-------------+----------------------------------------------------------------
American     |  (base alternative)
-------------+----------------------------------------------------------------
Japanese     |
      gender |
       Male  |  -.3793272   .1712406    -2.22   0.027    -.7149526   -.0437018
      income |   .0154978   .0065145     2.38   0.017     .0027296    .0282661
       _cons |  -.4787181   .3313811    -1.44   0.149    -1.128213     .170777
-------------+----------------------------------------------------------------
European     |
      gender |
       Male  |   .6533465   .2647669     2.47   0.014     .1344129     1.17228
      income |   .0343648   .0080286     4.28   0.000     .0186291    .0501005
       _cons |   -2.83959   .4616219    -6.15   0.000    -3.744353   -1.934828
-------------+----------------------------------------------------------------
Korean       |
      gender |
       Male  |   .0679287   .4464459     0.15   0.879    -.8070892    .9429466
      income |  -.0377715   .0158431    -2.38   0.017    -.0688234   -.0067196
       _cons |   .0511891   .8033007     0.06   0.949    -1.523251     1.62563
------------------------------------------------------------------------------
LR test vs. fixed parameters: chibar2(01) =      0.00  Prob >= chibar2 = 0.5000
```

The random-coefficients model is still not significantly different from a fixed coefficient model.

At first glance, the requirement of including random coefficients on alternative-specific variables in this model may seem limiting. What if we do not have alternative-specific variables for which random coefficients are appropriate? Note that the constants in the model are alternative specific. They are automatically included in the model for us, but we could have equivalently typed `i.car` in the list of alternative-specific variables to include indicators for the alternatives. We can turn any of or all the constants into random intercepts. Let's do this with the constant for the European alternative. Now we need to use the factor-variable specification for the alternative-specific constants. Because we want fixed coefficients on Japanese and Korean indicators, we type `i(2 4).car` in the fixed portion of the model. To get random coefficients for the European constant, we type `random(i3.car)`. We also specify the options `noconstant` and `collinear` (or else `cmmixlogit` would drop the constants).

```
. cmmixlogit purchase dealers i(2 4).car, random(i3.car)
> casevars(i.gender income) noconstant collinear
note: alternatives are unbalanced.

Fitting fixed parameter model:

Fitting full model:

Iteration 0:  Log simulated-likelihood = -1717.8292  (not concave)
Iteration 1:  Log simulated-likelihood = -1471.6665  (not concave)
Iteration 2:  Log simulated-likelihood = -1456.0693  (not concave)
Iteration 3:  Log simulated-likelihood = -1431.4506  (not concave)
Iteration 4:  Log simulated-likelihood = -1412.2678  (not concave)
Iteration 5:  Log simulated-likelihood = -1382.4808  (not concave)
Iteration 6:  Log simulated-likelihood = -1359.4781  (not concave)
Iteration 7:  Log simulated-likelihood = -1341.5917  (not concave)
Iteration 8:  Log simulated-likelihood = -1327.6059  (not concave)
Iteration 9:  Log simulated-likelihood = -1316.6209  (not concave)
Iteration 10: Log simulated-likelihood = -1307.9616  (not concave)
Iteration 11: Log simulated-likelihood = -1294.3419  (not concave)
Iteration 12: Log simulated-likelihood =  -1155.848  (not concave)
Iteration 13: Log simulated-likelihood = -998.89495
Iteration 14: Log simulated-likelihood = -950.28922
Iteration 15: Log simulated-likelihood = -949.17489
Iteration 16: Log simulated-likelihood = -949.17151
Iteration 17: Log simulated-likelihood = -949.16844
Iteration 18: Log simulated-likelihood = -949.16776
Iteration 19: Log simulated-likelihood = -949.16759
Iteration 20: Log simulated-likelihood = -949.16755
Iteration 21: Log simulated-likelihood = -949.16754
```

```
Mixed logit choice model                          Number of obs      =      3,075
Case ID variable: consumerid                      Number of cases    =        862

Alternatives variable: car                        Alts per case: min =          3
                                                                 avg =        3.6
                                                                 max =          4

Integration sequence:      Hammersley
Integration points:               623                Wald chi2(9)    =     200.60
Log simulated-likelihood = -949.16754                Prob > chi2     =     0.0000

------------------------------------------------------------------------------
    purchase | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
car          |
     dealers |   .0502172   .0259823     1.93   0.053    -.0007071    .1011415
             |
         car |
      Korean |   .2926751   .7850917     0.37   0.709    -1.246076    1.831427
    European |  -2.591301   .4270812    -6.07   0.000    -3.428365   -1.754237
-------------+----------------------------------------------------------------
/Normal      |
    sd(3.car)|   .0001367   1.640459                             .           .
-------------+----------------------------------------------------------------
American     |  (base alternative)
-------------+----------------------------------------------------------------
Japanese     |
      gender |
        Male |  -.5113203   .1448827    -3.53   0.000    -.7952852   -.2273554
      income |    .007125   .0029512     2.41   0.016     .0013408    .0129093
-------------+----------------------------------------------------------------
European     |
      gender |
        Male |   .5843488   .2610755     2.24   0.025     .0726502    1.096047
      income |   .0304615   .0075051     4.06   0.000     .0157518    .0451712
-------------+----------------------------------------------------------------
Korean       |
      gender |
        Male |   .0012654    .444595     0.00   0.998    -.8701249    .8726556
      income |  -.0413199   .0156261    -2.64   0.008    -.0719464   -.0106934
------------------------------------------------------------------------------
LR test vs. fixed parameters: chibar2(01) =      0.00  Prob >= chibar2 = 0.5000
```

This model with a random intercept for the European alternative is not significantly different from a fixed-coefficient model. But this illustrates one of the features of `cmmixlogit`. Making the alternative-specific constants random allows us to fit models that do not satisfy IIA and test them against a fixed-coefficient model that does satisfy IIA.

See [CM] **cmmixlogit** for examples where the random-coefficients model fits better than the one with fixed coefficients. There we demonstrate how to further interpret results of these models. In addition, you can use `margins` in the same ways shown in [CM] **Intro 1** and as we did after `cmclogit` above to interpret mixed logit models.

cmmprobit: Multinomial probit choice models

`cmmprobit` fits a multinomial probit (MNP) choice model. The formulation of the utility for MNP is described in [CM] **Intro 8**. The model is similar to McFadden's choice model (`cmclogit`), except that the random-error term is modeled using a multivariate normal distribution, and you can explicitly model the covariance.

When there are no alternative-specific variables in your model, covariance parameters are not identifiable. In this case, better alternatives are mprobit, which is geared specifically toward models with only case-specific variables, or a random-intercept model fit by cmmixlogit.

The covariance parameters are set using the correlation() and stddev() options of cmmprobit. In general, there are $J(J+1)/2$ possible covariance parameters, where J is the number of possible alternatives. One of the alternatives is set as the base category, and only the relative differences among the utilities matter. This reduces the possible number of covariance parameters by J.

The scale of the utilities does not matter. Multiply the utilities for all alternatives by the same constant, and the relative differences are unchanged. This further reduces the number of covariance parameters by one. So there are a total of $J(J-1)/2-1$ covariance parameters you can fit. But you do not have to fit all of them. You can set some of them to fixed values, either zero or nonzero. Or you can constrain some of them to be equal.

When J is large, it is a good idea to initially fit just a few parameters and then gradually increase the number. If you try to fit a lot of parameters, your model may have a hard time converging because some of the parameters may not be identified. For example, the true variance for one of the alternatives may be zero, and if you try to estimate the standard deviation for the alternative, the model may not converge because zero is not part of the estimable parameter space.

See *Covariance structures* in [CM] **cmmprobit** for full details on all the choices for specifying the covariance parameters.

cmmprobit has some options for reducing the number of covariance parameters. In particular, correlation(exchangeable) fits a model in which correlations between the alternatives are all the same. Another way to reduce the number of parameters estimated is the factor(#) option. cmmprobit with factor(#) fits a covariance matrix of the form $I+\mathbf{C}'\mathbf{C}$, where the row dimension of the matrix $\mathbf{C}$ is #.

Let's fit a model using factor(1) with the data from the previous examples.

```
. cmmprobit purchase dealers, casevars(i.gender income) factor(1)
Iteration 0:  Log simulated-likelihood = -949.38598
Iteration 1:  Log simulated-likelihood = -949.08161  (backed up)
Iteration 2:  Log simulated-likelihood = -948.87143  (backed up)
Iteration 3:  Log simulated-likelihood = -948.84362  (backed up)
Iteration 4:  Log simulated-likelihood = -948.83433  (backed up)
Iteration 5:  Log simulated-likelihood = -948.53624  (backed up)
Iteration 6:  Log simulated-likelihood = -948.52521
Iteration 7:  Log simulated-likelihood = -948.42813
Iteration 8:  Log simulated-likelihood = -948.14286
Iteration 9:  Log simulated-likelihood = -948.03466
Iteration 10: Log simulated-likelihood = -948.01302
Iteration 11: Log simulated-likelihood = -947.83629
Iteration 12: Log simulated-likelihood = -947.78297
Iteration 13: Log simulated-likelihood =  -947.6765
Iteration 14: Log simulated-likelihood = -947.60503
Iteration 15: Log simulated-likelihood =  -947.5831
Iteration 16: Log simulated-likelihood = -947.55131
Iteration 17: Log simulated-likelihood = -947.50624
Iteration 18: Log simulated-likelihood = -947.46284
Iteration 19: Log simulated-likelihood = -947.44467
Iteration 20: Log simulated-likelihood = -947.40163
Iteration 21: Log simulated-likelihood = -947.32181
Iteration 22: Log simulated-likelihood = -947.29791
Iteration 23: Log simulated-likelihood = -947.23404
Iteration 24: Log simulated-likelihood = -947.17847
Iteration 25: Log simulated-likelihood = -947.13231
Iteration 26: Log simulated-likelihood = -947.08427
```

```
Iteration 27: Log simulated-likelihood = -946.83137
Iteration 28: Log simulated-likelihood = -946.73195
Iteration 29: Log simulated-likelihood = -946.44451
Iteration 30: Log simulated-likelihood = -946.37077
Iteration 31: Log simulated-likelihood = -946.34252
Iteration 32: Log simulated-likelihood = -946.32218
Iteration 33: Log simulated-likelihood = -946.31672
Iteration 34: Log simulated-likelihood = -946.31499
Iteration 35: Log simulated-likelihood = -946.31489
Iteration 36: Log simulated-likelihood = -946.31487
Iteration 37: Log simulated-likelihood = -946.31486
Iteration 38: Log simulated-likelihood =  -946.3148
Iteration 39: Log simulated-likelihood =  -946.3141
Iteration 40: Log simulated-likelihood = -946.31203
Iteration 41: Log simulated-likelihood =  -946.3114
Iteration 42: Log simulated-likelihood = -946.31114
Iteration 43: Log simulated-likelihood = -946.31109
Iteration 44: Log simulated-likelihood = -946.31109

Multinomial probit choice model                 Number of obs      =     3,075
Case ID variable: consumerid                    Number of cases    =       862

Alternatives variable: car                      Alts per case: min =         3
                                                               avg =       3.6
                                                               max =         4

Integration sequence:      Hammersley
Integration points:               704              Wald chi2(7)    =     33.07
Log simulated-likelihood = -946.31109              Prob > chi2     =    0.0000
```

purchase	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
car						
dealers	.043345	.027397	1.58	0.114	-.0103522	.0970422
American	(base alternative)					
Japanese						
gender						
Male	-.4549281	.1454853	-3.13	0.002	-.7400741	-.1697821
income	.0092406	.0054458	1.70	0.090	-.0014331	.0199142
_cons	-.419605	.2779042	-1.51	0.131	-.9642872	.1250773
European						
gender						
Male	.5630869	.4209101	1.34	0.181	-.2618817	1.388056
income	.0201237	.0102355	1.97	0.049	.0000625	.0401849
_cons	-2.273778	1.499661	-1.52	0.129	-5.21306	.6655044
Korean						
gender						
Male	.3081901	.4970798	0.62	0.535	-.6660685	1.282449
income	-.035191	.0346554	-1.02	0.310	-.1031144	.0327323
_cons	-.9509444	1.056018	-0.90	0.368	-3.020701	1.118812
/c1_2	-.8477297	1.362819	-0.62	0.534	-3.518807	1.823347
/c1_3	-1.675403	1.433511	-1.17	0.243	-4.485033	1.134226

```
(car=American is the alternative normalizing location)
(car=Japanese is the alternative normalizing scale)

. matrix b704 = e(b)
```

The estimated covariance parameters are shown in the output, but more useful is to see the estimated covariance matrix or correlation matrix. The postestimation command `estat` will display them. `estat covariance` shows the covariance matrix, and `estat correlation` shows the correlations.

```
. estat covariance
```

	Japanese	European	Korean
Japanese	2		
European	-.8477297	1.718646	
Korean	-1.675403	1.420289	3.806976

```
Note: Covariances are for alternatives differenced with American.
. estat correlation
```

	Japanese	European	Korean
Japanese	1.0000		
European	-0.4572	1.0000	
Korean	-0.6072	0.5553	1.0000

```
Note: Correlations are for alternatives differenced with American.
```

There are four alternatives in these data. But the matrices shown here are only 3×3. This is because the parameterization for the covariance matrix is, by default, differed by the base category, which in this case is the alternative American.

To see an undifferenced parameterization, we specify the `structural` option:

```
. cmmprobit purchase dealers, casevars(i.gender income) factor(1) structural
Iteration 0:  Log simulated-likelihood = -949.81324
Iteration 1:  Log simulated-likelihood = -948.95649  (backed up)
Iteration 2:  Log simulated-likelihood = -948.71164  (backed up)
Iteration 3:  Log simulated-likelihood = -948.70869  (backed up)
Iteration 4:  Log simulated-likelihood = -948.65719  (backed up)
Iteration 5:  Log simulated-likelihood = -948.52707
Iteration 6:  Log simulated-likelihood = -948.52682
Iteration 7:  Log simulated-likelihood = -948.44886
Iteration 8:  Log simulated-likelihood = -948.29451
Iteration 9:  Log simulated-likelihood = -948.22865
Iteration 10: Log simulated-likelihood = -948.14213
Iteration 11: Log simulated-likelihood = -947.96801
Iteration 12: Log simulated-likelihood = -947.95862
Iteration 13: Log simulated-likelihood = -947.85813
Iteration 14: Log simulated-likelihood = -947.84956
Iteration 15: Log simulated-likelihood =  -947.7153
Iteration 16: Log simulated-likelihood = -947.67296
Iteration 17: Log simulated-likelihood = -947.57769
Iteration 18: Log simulated-likelihood = -947.42721
Iteration 19: Log simulated-likelihood = -947.19551
Iteration 20: Log simulated-likelihood = -947.11421
Iteration 21: Log simulated-likelihood = -946.90873
Iteration 22: Log simulated-likelihood = -946.75482
Iteration 23: Log simulated-likelihood = -946.64695
Iteration 24: Log simulated-likelihood = -946.56345
Iteration 25: Log simulated-likelihood = -946.44076
Iteration 26: Log simulated-likelihood =  -946.3817
Iteration 27: Log simulated-likelihood = -946.35537
Iteration 28: Log simulated-likelihood = -946.34227
Iteration 29: Log simulated-likelihood = -946.33841
Iteration 30: Log simulated-likelihood = -946.33808
```

```
Iteration 31: Log simulated-likelihood = -946.33792
Iteration 32: Log simulated-likelihood = -946.33659
Iteration 33: Log simulated-likelihood = -946.33345
Iteration 34: Log simulated-likelihood = -946.32049
Iteration 35: Log simulated-likelihood = -946.30741
Iteration 36: Log simulated-likelihood = -946.29907
Iteration 37: Log simulated-likelihood = -946.29486
Iteration 38: Log simulated-likelihood = -946.29353
Iteration 39: Log simulated-likelihood = -946.29243
Iteration 40: Log simulated-likelihood = -946.29179
Iteration 41: Log simulated-likelihood = -946.29159
Iteration 42: Log simulated-likelihood = -946.29158
Multinomial probit choice model                 Number of obs      =      3,075
Case ID variable: consumerid                    Number of cases    =        862

Alternatives variable: car                      Alts per case: min =          3
                                                               avg =        3.6
                                                               max =          4

Integration sequence:      Hammersley
Integration points:               704              Wald chi2(7)    =      29.28
Log simulated-likelihood = -946.29158              Prob > chi2     =     0.0001

------------------------------------------------------------------------------
    purchase | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
car          |
     dealers |   .0702705   .0443614     1.58   0.113    -.0166763    .1572172
-------------+----------------------------------------------------------------
American     |  (base alternative)
-------------+----------------------------------------------------------------
Japanese     |
      gender |
       Male  |  -.5091748   .1855697    -2.74   0.006    -.8728848   -.1454648
      income |    .013291   .0067122     1.98   0.048     .0001354    .0264467
       _cons |  -.4826255   .3426393    -1.41   0.159    -1.154186    .1889352
-------------+----------------------------------------------------------------
European     |
      gender |
       Male  |    .784163   .7110074     1.10   0.270    -.6093859    2.177712
      income |   .0295921   .0163378     1.81   0.070    -.0024295    .0616136
       _cons |  -3.121519   2.487812    -1.25   0.210    -7.997541    1.754503
-------------+----------------------------------------------------------------
Korean       |
      gender |
       Male  |   .5169586   .6933728     0.75   0.456    -.8420271    1.875944
      income |  -.0271951    .021043    -1.29   0.196    -.0684387    .0140485
       _cons |  -1.150509   1.559621    -0.74   0.461     -4.20731    1.906292
-------------+----------------------------------------------------------------
       /c1_3 |  -1.407566   2.060875    -0.68   0.495    -5.446806    2.631674
       /c1_4 |  -1.709069   1.381237    -1.24   0.216    -4.416244    .9981062
------------------------------------------------------------------------------
(car=American is the alternative normalizing location)
(car=Japanese is the alternative normalizing scale)
. estat covariance

  +-------------------------------------------------------+
  |              |   American   Japanese   European     Korean |
  |--------------+--------------------------------------------|
  |     American |          1                                  |
  |     Japanese |          0          2                       |
  |     European |          0  -1.407566   2.981243            |
  |       Korean |          0  -1.709069   2.405628   3.920917 |
  +-------------------------------------------------------+
```

```
. estat correlation
```

	American	Japanese	European	Korean
American	1.0000			
Japanese	0.0000	1.0000		
European	0.0000	-0.5764	1.0000	
Korean	0.0000	-0.6103	0.7036	1.0000

When using the `structural` option, you must carefully specify the covariance parameterization because, as we described earlier, not all of $J(J+1)/2$ elements of the covariance matrix are identifiable. There are at most $J(J-1)/2-1$ estimable parameters, so either elements have to be set to fixed values or constraints need to be imposed. Specifying any desired parameterization is straightforward. It merely requires learning how to use the `correlation()` and `stddev()` options. See *Covariance structures* in [CM] **cmmprobit**.

nlogit: Nested logit choice models

`nlogit` fits nested logit choice models. Alternatives can be nested within alternatives. For example, the data could represent first-level choices of what restaurant to dine at and second-level choices of what is ordered at the restaurant. Clearly, the menu choices will depend upon the type of restaurant. The second-level alternatives are conditional on the first-level alternatives.

Although `nlogit` fits choice models, it is not a `cm` command, and you do not have to `cmset` your data. Because of the nested alternatives, `nlogit` has its own unique data requirements.

See [CM] **nlogit** for full details on nested logit choice models.

Relationships with other estimation commands

If you are familiar with conditional logistic regression or with multinomial logistic regression, you may find it helpful to see how the `cm` commands, and in particular `cmclogit`, compare with Stata's `clogit` and `mlogit` commands.

Duplicating cmclogit using clogit

Both `cmclogit` and `clogit` fit conditional logistic regression models. `cmclogit` has special handling of errors, alternative-specific and case-specific variables, and special postestimation commands that are appropriate for choice data. However, you can fit the same model with `cmclogit` and `clogit`.

Before we try to duplicate our `cmclogit` results with `clogit`, we will drop the cases with missing values using the `flag` variable that we created with our earlier `cmsample` command. We do this because `clogit` does not handle missing values the same way `cmclogit` does. By default, `cmclogit` drops the entire case when any observation in the case has a missing value. `clogit` drops only the observations that contain missing values.

```
. drop if flag != 0
(85 observations deleted)
```

To duplicate our `cmclogit` results with `clogit`, we merely have to create interactions of the case-specific variables (`gender` and `income`) with the alternatives variable `car`. To do this, we include the factor-variable terms `car##gender` and `car##c.income` in our `clogit` specification. (We use `c.income` because `income` is continuous; see [U] **11.4.3 Factor variables** for more on factor variables.) The alternative-specific variable `dealers` is included in the estimation as is.

```
. clogit purchase dealers car##gender car##c.income, group(consumerid)
note: 1.gender omitted because of no within-group variance.
note: income omitted because of no within-group variance.
Iteration 0:   Log likelihood = -959.21405
Iteration 1:   Log likelihood = -948.48587
Iteration 2:   Log likelihood =  -948.1217
Iteration 3:   Log likelihood = -948.12096
Iteration 4:   Log likelihood = -948.12096
Conditional (fixed-effects) logistic regression         Number of obs =  3,075
                                                        LR chi2(10)   = 279.12
                                                        Prob > chi2   = 0.0000
Log likelihood = -948.12096                             Pseudo R2     = 0.1283

------------------------------------------------------------------------------
    purchase | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
     dealers |   .0448082   .0262818     1.70   0.088    -.0067032    .0963196
             |
         car |
    Japanese |  -.4787261    .331378    -1.44   0.149    -1.128215    .1707628
    European |  -2.839606    .461613    -6.15   0.000    -3.744351   -1.934861
      Korean |   .0511728   .8033048     0.06   0.949    -1.523276    1.625621
             |
      gender |
        Male |          0  (omitted)
             |
  car#gender |
  Japanese # |
        Male |   -.379326   .1712399    -2.22   0.027      -.71495   -.0437021
  European # |
        Male |    .653345   .2647694     2.47   0.014     .1344065    1.172283
 Korean#Male |   .0679233   .4464535     0.15   0.879    -.8071094     .942956
             |
      income |          0  (omitted)
             |
car#c.income |
    Japanese |   .0154978   .0065145     2.38   0.017     .0027296    .0282659
    European |   .0343647   .0080286     4.28   0.000     .0186289    .0501006
      Korean |  -.0377716   .0158434    -2.38   0.017     -.068824   -.0067191
------------------------------------------------------------------------------
```

The output is in a different order, but all the coefficient estimates and their standard errors are exactly the same as the earlier results from `cmclogit`.

And they should be—because `cmclogit` calls `clogit` to do the estimation.

Multinomial logistic regression and McFadden's choice model

Multinomial logistic regression (`mlogit`) is a special case of McFadden's choice model (`cmclogit`). When there are only case-specific variables in the model and when the choice sets are balanced (that is, every case has the same alternatives), then `mlogit` gives the same results as `cmclogit`.

We can illustrate this, but the choice data we are working with are not balanced. So let's just use a subset of the dataset that is balanced. We can see the distinct choice sets using `cmchoiceset`.

```
. cmchoiceset, generate(choiceset)
Tabulation of choice-set possibilities
 Choice set |      Freq.     Percent        Cum.
------------+-----------------------------------
      1 2 3 |        373       43.27       43.27
    1 2 3 4 |        489       56.73      100.00
------------+-----------------------------------
      Total |        862      100.00
Note: Total is number of cases.
```

We included the `generate()` option to create an indicator variable `choiceset` with categories of the choice sets. We use this variable to keep only those cases that have the alternatives $\{1, 2, 3, 4\}$.

```
. keep if choiceset == "1 2 3 4":choiceset
(1,119 observations deleted)
```

(If you are not familiar with the `"1 2 3 4":choiceset` syntax, see [U] 13.11 Label values.)

Now we run `cmclogit` on this balanced sample:

```
. cmclogit purchase, casevars(i.gender income)
Iteration 0:  Log likelihood = -580.83991
Iteration 1:  Log likelihood = -575.60247
Iteration 2:  Log likelihood = -575.21416
Iteration 3:  Log likelihood = -575.21287
Iteration 4:  Log likelihood = -575.21287
Conditional logit choice model                Number of obs      =     1,956
Case ID variable: consumerid                  Number of cases    =       489
Alternatives variable: car                    Alts per case: min =         4
                                                             avg =       4.0
                                                             max =         4
                                                 Wald chi2(6)    =     41.24
Log likelihood = -575.21287                      Prob > chi2     =    0.0000

------------------------------------------------------------------------------
    purchase | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
American     |  (base alternative)
-------------+----------------------------------------------------------------
Japanese     |
      gender |
       Male  |  -.7164669   .2351233    -3.05   0.002      -1.1773   -.2556338
      income |   .0174375   .0087817     1.99   0.047     .0002257    .0346493
       _cons |  -.2370371   .4413551    -0.54   0.591    -1.102077    .6280029
-------------+----------------------------------------------------------------
European     |
      gender |
       Male  |   .2128877   .3494225     0.61   0.542    -.4719679    .8977433
      income |   .0409691   .0110817     3.70   0.000     .0192494    .0626888
       _cons |  -2.940079   .5956109    -4.94   0.000    -4.107455   -1.772703
-------------+----------------------------------------------------------------
Korean       |
      gender |
       Male  |  -.1892108   .4595242    -0.41   0.681    -1.089862      .71144
      income |  -.0361748    .016143    -2.24   0.025    -.0678145    -.004535
       _cons |  -.0367581   .8051745    -0.05   0.964    -1.614871    1.541355
------------------------------------------------------------------------------
```

To run `mlogit`, we must create a categorical dependent variable containing the chosen alternative, American, Japanese, European, or Korean. The values of the alternatives variable `car` at the observations representing the chosen alternative (`purchase` equal to one) yield a dependent variable appropriate for `mlogit`.

```
. keep if purchase == 1
(1,467 observations deleted)

. mlogit car i.gender income

Iteration 0:   Log likelihood = -596.47415
Iteration 1:   Log likelihood = -575.81328
Iteration 2:   Log likelihood = -575.21417
Iteration 3:   Log likelihood = -575.21287
Iteration 4:   Log likelihood = -575.21287

Multinomial logistic regression                         Number of obs =    489
                                                        LR chi2(6)    =  42.52
                                                        Prob > chi2   = 0.0000
Log likelihood = -575.21287                             Pseudo R2     = 0.0356
```

car	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
American	(base outcome)					
Japanese						
gender						
Male	-.7164669	.2351233	-3.05	0.002	-1.1773	-.2556338
income	.0174375	.0087817	1.99	0.047	.0002257	.0346493
_cons	-.2370371	.4413551	-0.54	0.591	-1.102077	.6280029
European						
gender						
Male	.2128877	.3494225	0.61	0.542	-.4719679	.8977433
income	.0409691	.0110817	3.70	0.000	.0192494	.0626888
_cons	-2.940079	.5956109	-4.94	0.000	-4.107455	-1.772703
Korean						
gender						
Male	-.1892108	.4595242	-0.41	0.681	-1.089862	.71144
income	-.0361748	.016143	-2.24	0.025	-.0678145	-.004535
_cons	-.0367581	.8051745	-0.05	0.964	-1.614871	1.541355

The estimates are identical.

Estimation considerations

When fitting choice models, you may need to address issues such as setting the number of integration points, lack of convergence, or data with multiple outcomes selected. Below, we provide advice on these topics.

Setting the number of integration points

In *Maximum simulated likelihood* of [CM] **Intro 8**, we describe how the estimators for `cmmixlogit`, `cmxtmixlogit`, `cmmprobit`, and `cmroprobit` all approximate integrals using Monte-Carlo simulation to compute their likelihoods. Monte-Carlo simulation creates additional variance in the estimated results, and the variance is dependent on the number of points used in the integration. More points give smaller Monte-Carlo variance. Hence, when fitting final models, it is a good idea to use the option `intpoints(#)` to increase the number of integration points and check that the coefficient and parameter estimates and their standard estimates are stable. That is, check that they do not change appreciably as the number of integration points is increased.

In the first cmmprobit example in this introduction, the default number of integration points was 704. We stored the coefficient vector from that estimation in the vector b704. Let's open a fresh copy of our data and refit the same model, specifying intpoints(2000).

```
. use https://www.stata-press.com/data/r18/carchoice, clear
(Car choice data)

. cmset consumerid car
note: alternatives are unbalanced across choice sets; choice sets of
      different sizes found.

     Case ID variable: consumerid
Alternatives variable: car

. cmmprobit purchase dealers, casevars(i.gender income) factor(1)
> intpoints(2000)
  (iteration log omitted)

Multinomial probit choice model                 Number of obs      =      3,075
Case ID variable: consumerid                    Number of cases    =        862

Alternatives variable: car                      Alts per case: min =          3
                                                               avg =        3.6
                                                               max =          4

Integration sequence:      Hammersley
Integration points:              2000              Wald chi2(7)    =      32.62
Log simulated-likelihood = -946.31243              Prob > chi2     =     0.0000

------------------------------------------------------------------------------
    purchase | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
car          |
     dealers |   .0440584   .0279595     1.58   0.115    -.0107413    .0988581
-------------+----------------------------------------------------------------
American     |  (base alternative)
-------------+----------------------------------------------------------------
Japanese     |
      gender |
       Male  |  -.4558936   .1451544    -3.14   0.002     -.740391   -.1713961
      income |   .0091975   .0054259     1.70   0.090    -.0014371     .019832
       _cons |  -.4174475   .2776516    -1.50   0.133    -.9616346    .1267396
-------------+----------------------------------------------------------------
European     |
      gender |
       Male  |     .57696   .4404936     1.31   0.190    -.2863916    1.440312
      income |   .0204216   .0108951     1.87   0.061    -.0009325    .0417757
       _cons |  -2.326064   1.586286    -1.47   0.143    -5.435127    .7829993
-------------+----------------------------------------------------------------
Korean       |
      gender |
       Male  |   .3182168   .5023253     0.63   0.526    -.6663227    1.302756
      income |  -.0345119    .033025    -1.05   0.296    -.0992397    .0302159
       _cons |  -.9586931   1.055673    -0.91   0.364    -3.027775    1.110389
-------------+----------------------------------------------------------------
       /c1_2 |   -.896706   1.400021    -0.64   0.522    -3.640697    1.847285
       /c1_3 |  -1.667291   1.366339    -1.22   0.222    -4.345266    1.010685
------------------------------------------------------------------------------
(car=American is the alternative normalizing location)
(car=Japanese is the alternative normalizing scale)

. matrix b2000 = e(b)

. display mreldif(b704, b2000)
.02582179
```

We put the coefficient vector in b2000 and compared it with the earlier results using the mreldif() function, which computes relative differences between vectors (or between matrices). We see that there is a maximum relative difference between the coefficients from the two estimations of about 3%.

We now double the number of integration points to 4000 and store the coefficient vector in b4000. We omit showing the cmmprobit results and show only the comparison of the coefficient vectors:

```
. display mreldif(b2000, b4000)
.00178383
```

The relative difference declined as intpoints() is increased. The maximum relative difference between the estimation with 2000 points and the one with 4000 points is only 0.2%.

When we look at the differences between coefficients from different runs, it is important to note the values of the coefficients relative to their standard errors. For example, we may have a variance parameter that is near zero with a big standard error (relative to the parameter estimate). The relative difference of the parameter estimate between runs with different intpoints() may not decline rapidly with increasing numbers of points because we are essentially just fitting random noise.

Convergence

Sometimes, you will try to fit a model with one of the CM commands, and the model will not converge. You might see an iteration log that goes on and on with (backed up) or (not concave) at the end of each line.

In the previous section, we showed you how increasing the number of integration points using the option intpoints(*#*) improves precision of the estimates by reducing the random variance of the Monte-Carlo integration. The randomness of the Monte-Carlo integration can affect convergence in a random way. It is possible that rerunning the command with a different random-number seed (using set seed *#* or the option intseed(*#*)) may cause a model to converge that previously did not. Increasing the number of integration points might cause a model to converge that did not when fewer points were used. It is also possible that a model may converge using the default number of integration points, but no longer converge when more integration points are used.

Our advice is when your model is not converging, first try increasing the number of integration points. If this does not help, try thinking about your model. Perhaps, this should have been the first thing to try. But this might be more painful than setting intpoints() to a big number.

Lack of convergence may be trying to tell you something about your model. Perhaps, the model is misspecified. That is, your model is not close to the true data-generating process. Or, perhaps, you simply need to collect more data.

You may want to try simplifying your model. It is best to start with a covariance parameterization with just a few parameters and then gradually increase them. For cmmprobit, using correlation(independent) and stddev(heteroskedastic) is a good model to start with. Look at the variances before trying to parameterize any correlations. Using correlation(fixed *matname*) lets you specify which elements are fixed and which are estimated. You can also fit models with just one free correlation parameter. cmroprobit, which we describe in [CM] **Intro 6**, has the same options and the same advice can be followed.

For the mixed logit models fit by cmmixlogit and cmxtmixlogit, the covariance parameterization is specified by different options, but the same general advice applies. If you are having convergence problems, start with a simple model and gradually increase the number of covariance parameters estimated.

More than one chosen alternative

What if we have data in which more than one alternative is chosen for some of or all the cases?

Well, first, we need to assess whether the data are in fact rank-ordered alternatives. If so, see [CM] **Intro 6**. There are two CM estimators for rank-ordered alternatives: `cmrologit` and `cmroprobit`.

Second, we need to assess whether the data are perhaps actually panel data and whether the choices were made at different times. For example, we might have data on how people commuted to work on a given week. Some people may have driven a car every day, but some may have driven a car some days and taken the bus on other days. Data such as these are panel data. If we have data by day of the week, we can analyze them as panel data. See [CM] **Intro 7** and example 4 in [CM] **cmclogit**.

But what if the data arose from a design in which multiple choices were allowed and not ranked? For example, suppose consumers were given four breakfast cereals and asked to pick their two favorites, without picking a single most favorite. These data are not rank-ordered data, nor are they panel data.

We note that the random utility model (see [CM] **Intro 8**) for discrete choices yields only one chosen alternative per case: that with the greatest utility. In rank-ordered models, it yields a set of ranked alternatives without any ties. Because the utility function is continuous, ties are theoretically impossible.

Train (2009, sec. 2.2) notes that the set of alternatives can always be made mutually exclusive by considering the choice of two alternatives as a separate alternative. For example, with one or two choices allowed from alternatives A, B, and C, the set of alternatives is A only, B only, C only, A and B, A and C, and B and C, a total of six alternatives. When there are only a few alternatives, this may be an appropriate way to model your data.

References

McFadden, D. L., and K. E. Train. 2000. Mixed MNL models for discrete response. *Journal of Applied Econometrics* 15: 447–470. https://doi.org/10.1002/1099-1255(200009/10)15:5⟨447::AID-JAE570⟩3.0.CO;2-1.

Train, K. E. 2009. *Discrete Choice Methods with Simulation*. 2nd ed. New York: Cambridge University Press.

Also see

[CM] **Intro 1** — Interpretation of choice models

[CM] **Intro 2** — Data layout

[CM] **Intro 3** — Descriptive statistics

[CM] **Intro 4** — Estimation commands

[CM] **cmclogit** — Conditional logit (McFadden's) choice model

[CM] **cmmixlogit** — Mixed logit choice model

[CM] **cmmprobit** — Multinomial probit choice model

[CM] **nlogit** — Nested logit regression