# MLE - Lab 8

*Andy Ballard*

*March 3, 2017*

First, let's set up our workspace

## Today

- Notes on cross-grading
- Homework 3
- Multinomial logit
- Conditional logit

## Multinomial Logit

When do we use multinomial logit instead of ordered logit or simple logit?

Multinomial logistic regression is used to model nominal outcome variables, in which the log odds of the outcomes are modeled as a linear combination of the predictor variables.

```r
#Load up some data
data(Heating); heat <- Heating; rm(Heating) #This dataset is native to the 'mlogit' package
#Yes, packages can have data already associated with them, you load these with the data() command
#Yes, you can use semicolons to run more than one command on one line
#Why might we want the data object to be 'heat' instead of 'Heating'? Shorter, no caps
```

What kind of data do we have here?

The observations consist of single-family houses in California that were newly built and had central airconditioning. The choice is among heating systems. There are five types of systems in the data:

1. gas central (gc)
2. gas room (gr)
3. electric central (ec)
4. electric room (er)
5. heat pump (hp)

The 900 observations in the dataset are characterized by the following variables:

- `idcase` gives the observation number (1-900),
- `depvar` identifies the chosen alternative (gc, gr, ec, er, hp),
- `ic.alt` is the installation cost for the 5 alternatives,
- `oc.alt` is the annual operating cost for the 5 alternatives,
- `income` is the annual income of the household,
- `agehed` is the age of the household head,
- `rooms` is the number of rooms in the house,
- `region` a factor with levels ncostl (northern coastal region), scostl (southern coastal region), mountn (mountain region), valley (central valley region).

What we'll model is the choice of heating system, given cleverly by `depvar`.

Note that there is a separate installation cost and operating cost for each of the 5 alternatives for each single-family home, so there are 5 `oc` variables and 5 `ic` variables for each row. We need more information than just the costs associated with the chosen system in order to properly model this relationship. Also because of this, we will need to reshape our data so that the cost variables are in a form that the `mlogit()` function can read. There is a built in function in the `mlogit` package that can help us with this, called `mlogit.data()`.

`mlogit.data()` takes argumkents for:

- `data`, a data frame (`heat` in this case)
- `choice`, a string specifying which variable in the data frame indicates the outcome choice (`depvar`)
- `shape`, a specification where rows are either alternatives (`long`) or observations (`wide`, as in our data)
- `varying`, an index of the variables that depend on the specific alternatives (columns [3:12])
- Other stuff we don't need to worry about right now

```r
h <- mlogit.data(heat, shape="wide", choice="depvar", varying=c(3:12))
head(h)
```

```
##      idcase depvar income agehed rooms region alt      ic     oc chid
## 1.ec      1  FALSE      7     25     6 ncostl  ec  859.90 553.34    1
## 1.er      1  FALSE      7     25     6 ncostl  er  995.76 505.60    1
## 1.gc      1   TRUE      7     25     6 ncostl  gc  866.00 199.69    1
## 1.gr      1  FALSE      7     25     6 ncostl  gr  962.64 151.72    1
## 1.hp      1  FALSE      7     25     6 ncostl  hp 1135.50 237.88    1
## 2.ec      2  FALSE      5     60     5 scostl  ec  796.82 520.24    2
```

As we can see, now our data is in a form such that each observation has 5 rows, one for each of the possible heating systems. Note: you could also accomplish this with the `melt()` command from the `reshape2` package.

Now let's run a model and see what's up

```r
m1 <- mlogit(depvar~ic+oc|0, data=h)
summary(m1)
```

```
## 
## Call:
## mlogit(formula = depvar ~ ic + oc | 0, data = h, method = "nr", 
##     print.level = 0)
## 
## Frequencies of alternatives:
##       ec       er       gc       gr       hp 
## 0.071111 0.093333 0.636667 0.143333 0.055556 
## 
## nr method
## 4 iterations, 0h:0m:0s 
## g'(-H)^-1g = 1.56E-07 
## gradient close to zero 
## 
## Coefficients :
##       Estimate  Std. Error t-value  Pr(>|t|)    
## ic -0.00623187  0.00035277 -17.665 < 2.2e-16 ***
## oc -0.00458008  0.00032216 -14.217 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Log-Likelihood: -1095.2
```

What does this model tell us? It says that as the costs (both installation and operating, because both coefficients are negative) of any particular system increase (relative to other systems), that system is less likely to be chosen. Good, this confirms either that the data are good or that people aren't too crazy, or both.

We can look at how well the predicted probabilities match up with the observed probabilities:

```
m1 %>% fitted(., outcome=F) %>% apply(., 2, mean) %>% sort()
```

```
##         er         hp         ec         gr         gc
## 0.05141477 0.08718915 0.10413057 0.24030898 0.51695653
```

```
sort(table(heat$depvar)/nrow(heat))
```

```
##
##         hp         ec         er         gr         gc
## 0.05555556 0.07111111 0.09333333 0.14333333 0.63666667
```

Yikes, not great Bob. At least the model correctly orders the top two most commonly chosen systems, `gc` and `gr`, but it misses the other 3. Also, the values are off by about 10% or greater for `gc` and `gr`.

Okay, so how could we make our model better?

One thing we can do is to include alternative-specific constraints (separate constants for each alternative, with a reference category). This will actually mathematically guarantee that the predicted probabilities equal the observed probabilites.

```
m2 <- mlogit(depvar~ic+oc, h, reflevel = 'hp')
summary(m2)
```

```
##
## Call:
## mlogit(formula = depvar ~ ic + oc, data = h, reflevel = "hp",
##     method = "nr", print.level = 0)
##
## Frequencies of alternatives:
##       hp       ec       er       gc       gr
## 0.055556 0.071111 0.093333 0.636667 0.143333
##
## nr method
## 6 iterations, 0h:0m:0s
## g'(-H)^-1g = 9.58E-06
## successive function values within tolerance limits
##
## Coefficients :
##                   Estimate  Std. Error t-value  Pr(>|t|)
## ec:(intercept)  1.65884594  0.44841936  3.6993 0.0002162 ***
## er:(intercept)  1.85343697  0.36195509  5.1206 3.045e-07 ***
## gc:(intercept)  1.71097930  0.22674214  7.5459 4.485e-14 ***
## gr:(intercept)  0.30826328  0.20659222  1.4921 0.1356640
## ic             -0.00153315  0.00062086 -2.4694 0.0135333 *
## oc             -0.00699637  0.00155408 -4.5019 6.734e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -1008.2
## McFadden R^2:  0.013691
## Likelihood ratio test : chisq = 27.99 (p.value = 8.3572e-07)
```

```
m2 %>% fitted(., outcome=F) %>% apply(., 2, mean) %>% sort()
```

```
##         hp         ec         er         gr         gc
## 0.05555556 0.07111111 0.09333333 0.14333333 0.63666667
```

```r
sort(table(heat$depvar)/nrow(heat))
```

```
##
##         hp         ec         er         gr         gc
## 0.05555556 0.07111111 0.09333333 0.14333333 0.63666667
```

Hot dang! But there are other ways we can evaluate such a model. Let's try a substantive interpretation. What we have done is estimate the type of system chosen based on the cost of the system to install and the cost to run the system. The ratio between these coefficients can give us sort of a discount rate, or how much more people are willing to pay up front to have a system that runs more efficiently. We can call this the willingness to pay (wtp).

```r
wtp2 <- coef(m2)["oc"]/coef(m2)["ic"]
wtp2
```

```
##       oc
## 4.563385
```

```r
exp(coef(m2))
```

```
## ec:(intercept) er:(intercept) gc:(intercept) gr:(intercept)             ic
##       5.253245       6.381716       5.534379       1.361059       0.998468
##             oc
##       0.993028
## attr(,"fixed")
## ec:(intercept) er:(intercept) gc:(intercept) gr:(intercept)             ic
##          FALSE          FALSE          FALSE          FALSE          FALSE
##             oc
##          FALSE
```

According to our model, households are willing to pay $4.56 up front to save $1 each year, for a discount rate of 1/4.56 = .219 Does that seem reasonable? What more information might you need to make an informed decision about this?

Now we can do some prediction, our favorite. We've actually done this above (with the `fitted()` function) to calculate the discount factors. Now we'll just show that it's doing much the same thing we have done in the past.

```r
X <- model.matrix(m2) #Why don't we need to specify a scenario here?
alt <- factor(h$alt)
chid <- factor(h$chid)
Xbeta <- X %*% coef(m2)
e.Xbeta <- Xbeta %>% exp() %>% as.numeric()
se.Xbeta <- tapply(e.Xbeta, chid, sum)
preds <- e.Xbeta / se.Xbeta[chid]
preds <- matrix(preds, ncol = 5, byrow = TRUE) #Why would it be messy to use piping for this whole expr

preds %>% apply(., 2, mean) %>% sort()
```

```
## [1] 0.05555556 0.07111111 0.09333333 0.14333334 0.63666666
```

```r
#Is it the same as before?
sort(table(heat$depvar)/nrow(heat))
```

```
##
##         hp         ec         er         gr         gc
## 0.05555556 0.07111111 0.09333333 0.14333333 0.63666667
```

```
m2 %>% fitted(., outcome=F) %>% apply(., 2, mean) %>% sort()
```

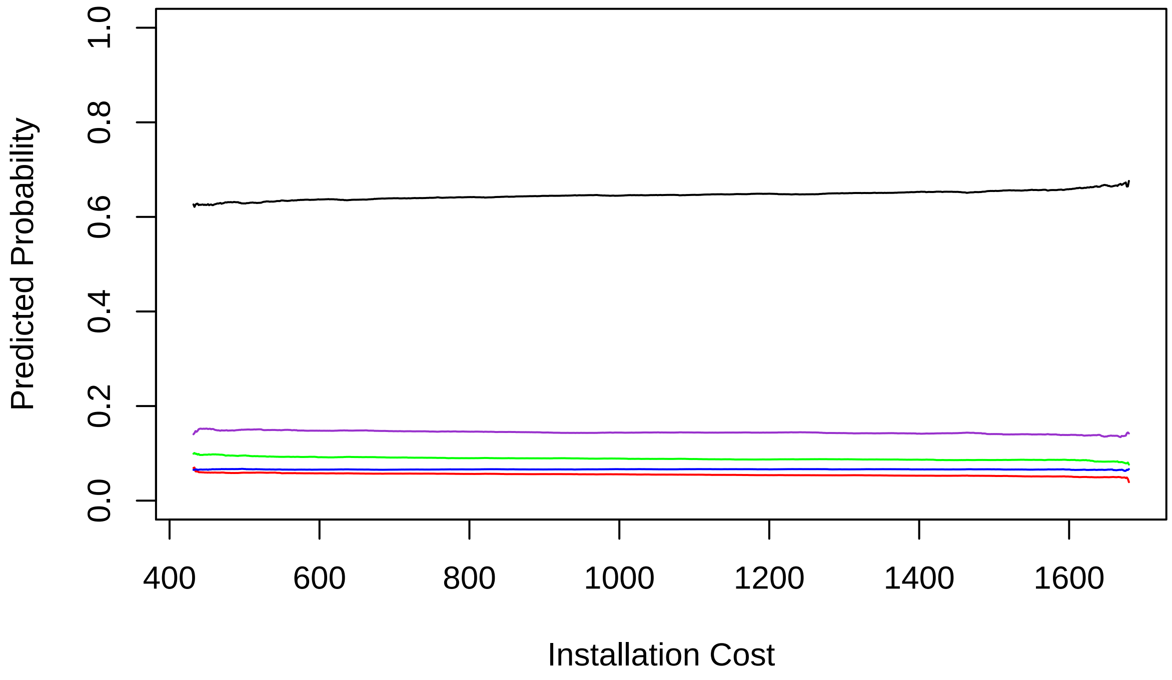
```
##          hp         ec         er         gr         gc
## 0.05555556 0.07111111 0.09333333 0.14333333 0.63666667
```

By George, it's the same! So what does that mean that the `fitted()` function is doing?

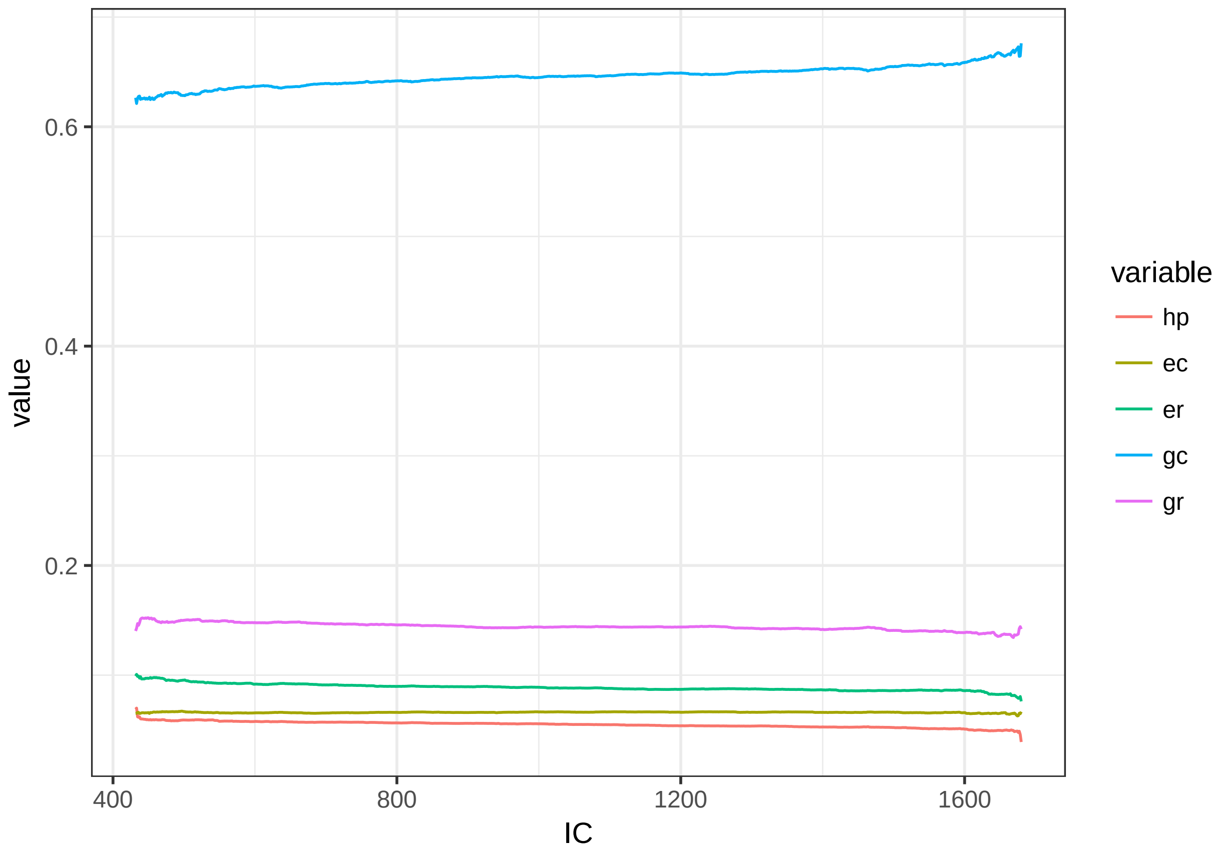
Let's look at some predictions how we're used to seeing them: plots.

```
# For this we'll have to divise a scenario, because of the way we generally plot these things
# Let's look at the effect of installation costs on the probability of choosing each system
hpred.df <- heat[,1:12]
pred.scen <- with(hpred.df, data.frame(depvar=depvar, ic.gc=sort(ic.gc), ic.gr=sort(ic.gr),
                                       ic.ec=sort(ic.ec), ic.er=sort(ic.er), ic.hp=sort(ic.hp),
                                       oc.gc=median(oc.gc), oc.gr=median(oc.gr), oc.ec=median(oc.ec),
                                       oc.er=median(oc.er), oc.hp=median(oc.hp)))
h.pred <- mlogit.data(pred.scen, shape="wide", choice="depvar", varying=c(2:11))
preds <- predict(m2,newdata=h.pred)
preds <- data.frame(cbind(preds, IC=seq(min(h$ic), max(h$ic), length.out=900)))


#Let's try using base plot for once
plot(preds$IC, preds$gc, type="l", ylim=c(0,1), xlab="Installation Cost", ylab="Predicted Probability")
lines(preds$IC, preds$hp, col="red")
lines(preds$IC, preds$ec, col="blue")
lines(preds$IC, preds$er, col="green")
lines(preds$IC, preds$gr, col="darkorchid")
```



```
#Is ggplot faster?
preds.long <- melt(preds, id="IC")
ggplot(preds.long, aes(x=IC, y=value, colour=variable)) + geom_line()
```



```
#Sure is (although you could also build the plot manually)
```

## Conditional vs. Multinomial Logit

Newsflash, we've been doing both "conditional" and "multinomial" logits this whole time. Here's the difference as described by the author of the `mlogit` package:

> "A model with only individual specific variables is sometimes called a multinomial logit model, one with only alternative specific variables a conditional logit model and one with both kind of variables a mixed logit model. This is seriously misleading : conditional logit model is also a logit model for longitudinal data in the statistical literature and mixed logit is one of the names of a logit model with random parameters. Therefore, in what follow, we'll use the name multinomial logit model for the model we've just described whatever the nature of the explanatory variables included in the model."

So which type of model is m1, and which is m2?

```
m1$call
```

```
## mlogit(formula = depvar ~ ic + oc | 0, data = h, method = "nr",
##     print.level = 0)
```

```
m2$call
```

```
## mlogit(formula = depvar ~ ic + oc, data = h, reflevel = "hp",
##     method = "nr", print.level = 0)
```