UP431 Lab3: Discrete Choice Analysis (1)

Chaeyeon Han

February 18, 2021

Introduction

This R Notebook complements the class lecture on discrete choice analysis. Let's start with a simple example to explore how we might estimate a model that predicts mode choice based on just a few variables.

Exploring binary choice models

Assume that you have a small dataset that, for each person, gives you the possible travel time between two points separately for auto travel and transit travel. Also assume the dataset gives you the mode the person ultimately selected. There are two possible modes: auto and transit. The dataset is posted on Compass. Import the csv file into an object called `binchoice`.

```
library(tidyverse)
```

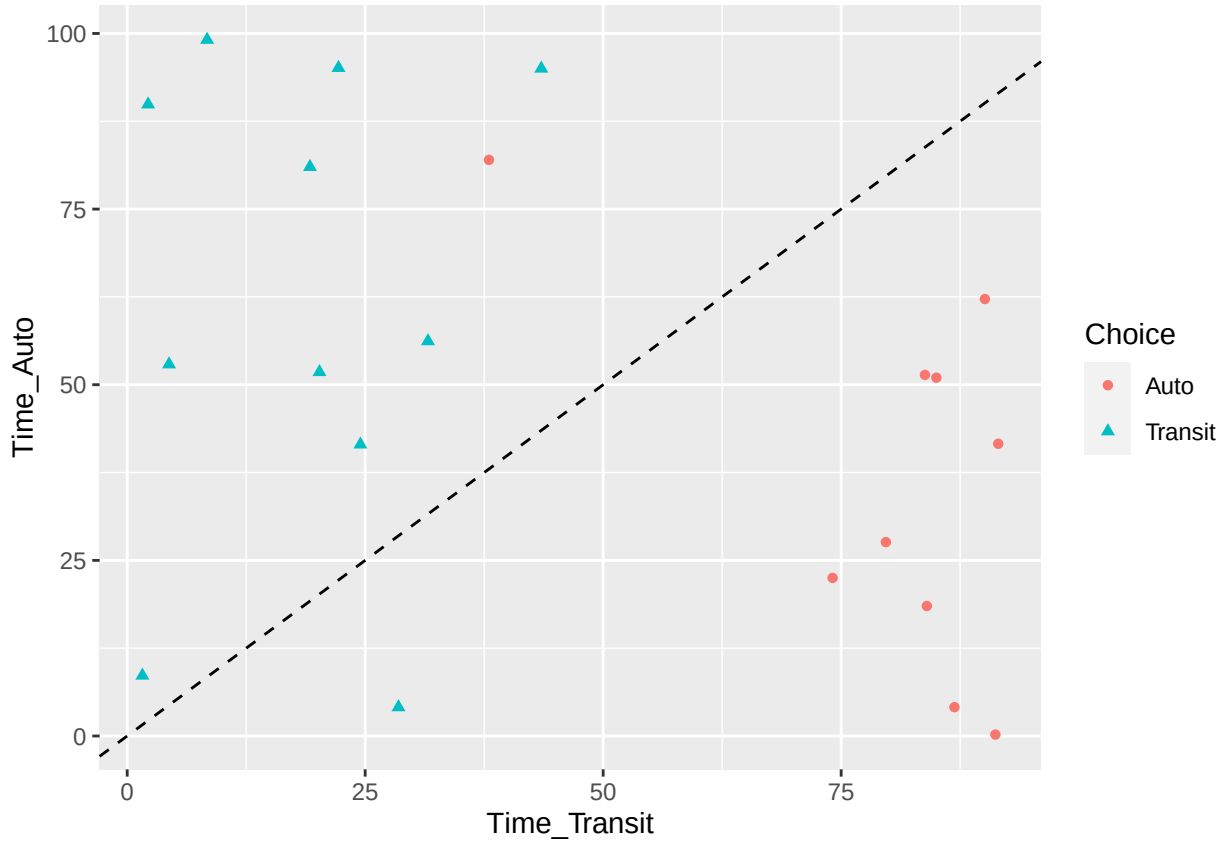
```
## -- Attaching packages ---------------------------------------------------------------------------

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## Warning: package 'readr' was built under R version 4.0.3

## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
binchoice <- read_csv("C:/Lab0/2021_UP431/Lab3/Data/simple_example2.csv")
```

```
##
## -- Column specification --------------------------------------------------------------------------
## cols(
##   Obs = col_double(),
##   Time_Auto = col_double(),
##   Time_Transit = col_double(),
##   Choice = col_character()
## )
```

How would you expect the different values for travel time to influence mode choice? Let's plot the data, with transit travel time on the x-axis and auto travel time on the y-axis, identifying the mode selected by shape and color.

```
ggplot(binchoice, aes(Time_Transit, Time_Auto, shape = Choice, color = Choice)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed")
```



It appears that transit is the preferred mode for most circumstances where the transit travel time is less than 50 minutes because in most of those instances, transit time is less than auto time. The other way to look at this is to draw a line with slope 1 and see which alternative falls on either side of the line. Again, the mode selected is usually the one which has the lower travel time.

Let's estimate a model that predicts how travel time influences whether an individual takes auto or transit. We need the `mlogit` package to estimate the model, so make sure it's installed on your machine.

```
#install.packages("mlogit")
library(mlogit)
```

```
## Warning: package 'mlogit' was built under R version 4.0.3
```

```
## Loading required package: dfidx
```

```
## Warning: package 'dfidx' was built under R version 4.0.3
```

```
##
## Attaching package: 'dfidx'

## The following object is masked from 'package:stats':
##
##     filter
```

Remember that what's going on behind the scenes is that we're estimating the utility of each mode, auto (A) and transit (T) for each traveler. The utilities we're interested in are:

$$V_{An} = \beta_1 A_A + \beta_2 T_A V_{Tn} = \beta_1 A_T + \beta_2 T_T$$

where A is alternative-specific constant indicating whether the utility is for auto (1) or transit (0), and T is the travel time for that mode. We have to estimate each β in the model to determine the utility values. But first, we need to do some data manipulation to get the data frame in a format that works with the `mlogit` package and the `mlogit` function.

```
# 1. We have to wrestle data into shape. The mlogit function wants a data frame in
# a special "long" format: one row for each ID and each possible choice. We use
# the mlogit.data function to do this. We tell it that the data is in a "wide"
# shape (one row for each observation), that the variable that contains the
# chosen alternative is called "Choice", that columns 2 and 3 vary depending
# on the choice, that those columns use the "_" character to separate the variable
# from the mode it applies to, and that the levels of the selected alternative
# are transit and auto, in that order. (Because we said the transit utility is 0.)
model_data <- mlogit.data(binchoice, shape = "wide", choice = "Choice",
                          varying = c(2, 3), sep = "_",
                          alt.levels = c("Transit", "Auto"))
model_data
```

```
## # A tibble: 42 x 6
##      Obs Choice alt      Time  chid idx$chid $alt
##  * <dbl> <lgl>  <fct>   <dbl> <int>    <int> <fct>
##  1     1 FALSE  Auto     52.9     1        1 Auto
##  2     1 TRUE   Transit   4.4     1        1 Transit
##  3     2 FALSE  Auto      4.1     2        2 Auto
##  4     2 TRUE   Transit  28.5     2        2 Transit
##  5     3 TRUE   Auto      4.1     3        3 Auto
##  6     3 FALSE  Transit  86.9     3        3 Transit
##  7     4 FALSE  Auto     56.2     4        4 Auto
##  8     4 TRUE   Transit  31.6     4        4 Transit
##  9     5 FALSE  Auto     51.8     5        5 Auto
## 10     5 TRUE   Transit  20.2     5        5 Transit
## # ... with 32 more rows
##
## ~~~ indexes ~~~~
##    chid     alt
## 1     1    Auto
## 2     1 Transit
## 3     2    Auto
## 4     2 Transit
## 5     3    Auto
## 6     3 Transit
```

```
## 7     4    Auto
## 8     4 Transit
## 9     5    Auto
## 10    5 Transit
## indexes:  1, 2
```

```r
# 2. Now we can estimate the model
# This says that Choice is a function of travel time (of both modes)
m1 <- mlogit(Choice ~ Time, model_data)

# 3. And read the summary
summary(m1)
```

```
##
## Call:
## mlogit(formula = Choice ~ Time, data = model_data, method = "nr")
##
## Frequencies of alternatives:choice
##    Auto Transit
## 0.47619 0.52381
##
## nr method
## 6 iterations, 0h:0m:0s
## g'(-H)^-1g = 1.81E-05
## successive function values within tolerance limits
##
## Coefficients :
##                       Estimate Std. Error z-value Pr(>|z|)
## (Intercept):Transit  0.237573   0.750477  0.3166  0.75158
## Time                -0.053110   0.020642 -2.5729  0.01009 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -6.166
## McFadden R^2:  0.5757
## Likelihood ratio test : chisq = 16.732 (p.value = 4.3038e-05)
```

Predicting Probabilities

Based on the result above, calculate the estimated probabilities estimated probabilities that the surveyed individual would choose either Auto or Transit. What percentage of modes chosen did the model estimate correctly?

```r
# Your code here

binchoice <- binchoice %>%
  mutate(Utility_Auto = 0.237573 * 0 + -0.053110 * Time_Auto,
         Utility_Transit = 0.237573 * 1 + -0.053110 * Time_Transit) %>%
  mutate(Prob_Auto = exp(Utility_Auto) / (exp(Utility_Auto) + exp(Utility_Transit)),
         Prob_Transit = exp(Utility_Transit) / (exp(Utility_Auto) + exp(Utility_Transit))) %>%
  mutate(Expected = if_else(Prob_Auto > Prob_Transit, "Auto", "Transit")) %>%
  mutate(Correct = if_else(Choice == as.character(Expected), TRUE, FALSE))
```

```r
prop.table(table(binchoice$Correct))
```

```
##
##     FALSE      TRUE
## 0.0952381 0.9047619
```

Multinomial Choice

In the previous examples, we looked at the case where a decision maker only had two choices. Now, we'll look at the case where there are multiple options. We'll work with an extract of the 2007–2008 CMAP Travel Tracker Survey, a household travel survey conducted in the Northeastern Illinois/Northwestern Indiana areas. The extract we're using is all the trips to work by walk, drive, carpool, or transit for people who lived in Cook County Illinois.

```r
library(tidyverse)
library(mlogit)

cook <- read_rds("C:/Lab0/2021_UP431/Lab3/Data/cook_county_work.rds")
```

Here are the variables in the dataset. Remember, you can find this out by clicking the data frame in the Data section of the environment tab, or using the `str()` function.

- `mode`: Mode choice (Walk, Drive, Carpool, or Transit)
- `tottr`: Number of people on the trip
- `trpdur`: Duration of the trip (minutes)
- `incom`: Household income ($000)
- `dist`: Straight-line distance between origin and destination (mi)
- `hhlic`: Number of household licensed drivers
- `hhveh`: Number of household vehicles
- `gend`: Gender
- `cars_per_driver`: Household vehicles per licensed driver
- `race_eth`: Race/ethnicity (Asian, Black, Hispanic, White, Unknown)

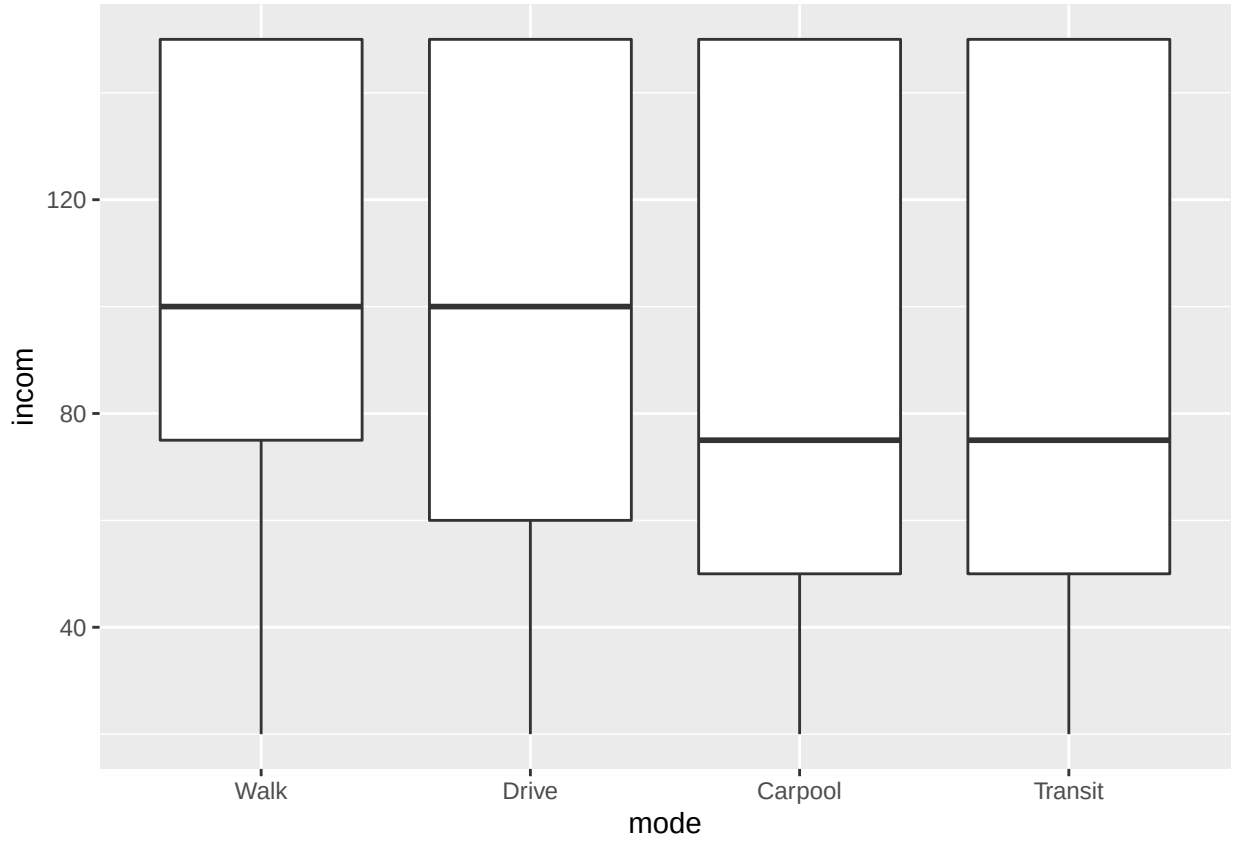
There's one key feature of this dataset: all of the variables are *individual specific* (or *choice specific*); there are no alternative-specific variables. This means that each row in the data frame represents values that apply to a decision maker rather than to the choice outcome being analyzed. This is important for how we specify the formula in the `mlogit()` function later.

Let's start simply and see what the effects of household income and gender are on mode choice. It's always a good idea to graph your data first to see if there are potential differences worth modeling. We have to graph each variable separately.

Income:

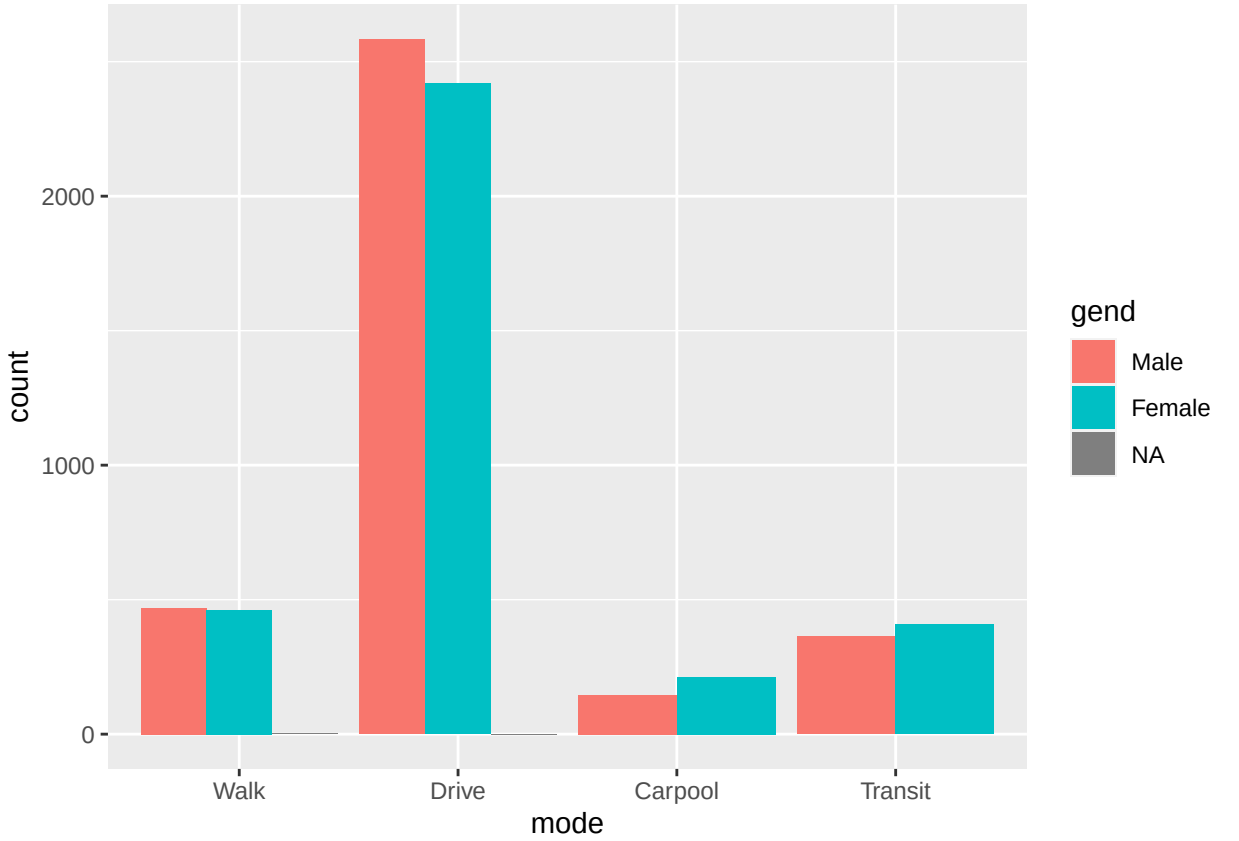
```r
ggplot(cook, aes(mode, incom)) +
  geom_boxplot()
```

```
## Warning: Removed 640 rows containing non-finite values (stat_boxplot).
```



Gender:

```
ggplot(cook, aes(mode, fill = gend, group = gend)) +
  geom_bar(position = "dodge")
```




```
cook <- cook %>% filter(!is.na(cook$incom) & !is.na(cook$gend))
```

It appears that there are differences in mode choice by income and gender, so let's continue using them as explanatory variables.

Here's how to specify the model:

```
m1 <- mlogit(mode ~ 1 | incom + gend, cook, shape = "wide")
```

Data sets can have two different shapes : • a wide shape : in this case, there is one row for each choice situation, • a long shape : in this case, there is one row for each alternative and, therefore, as many rows as there are alternatives for each choice situation.

This formula looks different from what we specified previously in the binomial models. Why? Because all the variables are individual specific, whereas our previous variables were alternative-specific. In the formula for `mlogit`, alternative-specific variables go on the left-hand side of the vertical bar (|) and the individual-specific variables go on the right-hand side. The 1 on the left-hand side of the bar indicates we want alternative-specific intercepts. We did this manually in the binomial versions of the models, but `mlogit` is smart enough to do this for us.

Note that we also didn't reshape our data before adding it to the model. When our variables are all individual-specific, we can skip this step and instead tell `mlogit` that we're using a wide dataset. Now let's look at the summary:

```
summary(m1)
```

```
##
## Call:
## mlogit(formula = mode ~ 1 | incom + gend, data = cook, shape = "wide",
##     method = "nr")
##
## Frequencies of alternatives:choice
## Carpool   Drive Transit    Walk
## 0.05028 0.70999 0.10866 0.13107
##
## nr method
## 6 iterations, 0h:0m:1s
## g'(-H)^-1g = 3.3E-06
## successive function values within tolerance limits
##
## Coefficients :
##                       Estimate Std. Error z-value  Pr(>|z|)
## (Intercept):Drive    2.4717045  0.1552602 15.9198 < 2.2e-16 ***
## (Intercept):Transit  1.0411133  0.1791043  5.8129 6.140e-09 ***
## (Intercept):Walk     0.3491169  0.1810140  1.9287  0.053771 .
## incom:Drive          0.0039033  0.0013184  2.9606  0.003071 **
## incom:Transit       -0.0015299  0.0015424 -0.9919  0.321261
## incom:Walk           0.0078839  0.0015084  5.2267 1.726e-07 ***
## gendFemale:Drive    -0.3727718  0.1171829 -3.1811  0.001467 **
## gendFemale:Transit  -0.2334349  0.1364513 -1.7108  0.087126 .
## gendFemale:Walk     -0.3370625  0.1329384 -2.5355  0.011229 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -5744.4
## McFadden R^2:  0.0075572
## Likelihood ratio test : chisq = 87.485 (p.value = < 2.22e-16)
```

What happens when we choose a different base case? Let's select driving as the reference level. We tell `mlogit()` we want to change the reference level by using the `reflevel` argument.

```
m2 <- mlogit(mode ~ 1 | incom + gend, cook, shape = "wide", reflevel = "Drive")
summary(m2)
```

```
##
## Call:
## mlogit(formula = mode ~ 1 | incom + gend, data = cook, reflevel = "Drive",
##     shape = "wide", method = "nr")
##
## Frequencies of alternatives:choice
##   Drive Carpool Transit    Walk
## 0.70999 0.05028 0.10866 0.13107
##
## nr method
## 6 iterations, 0h:0m:1s
## g'(-H)^-1g = 3.3E-06
## successive function values within tolerance limits
##
## Coefficients :
```

```
##                         Estimate  Std. Error  z-value  Pr(>|z|)
## (Intercept):Carpool -2.47170448  0.15526022 -15.9198 < 2.2e-16 ***
## (Intercept):Transit -1.43059120  0.10638942 -13.4467 < 2.2e-16 ***
## (Intercept):Walk    -2.12258755  0.10935378 -19.4103 < 2.2e-16 ***
## incom:Carpool       -0.00390329  0.00131843  -2.9606  0.003071 **
## incom:Transit       -0.00543317  0.00093617  -5.8036 6.489e-09 ***
## incom:Walk           0.00398056  0.00087428   4.5530 5.290e-06 ***
## gendFemale:Carpool   0.37277183  0.11718292   3.1811  0.001467 **
## gendFemale:Transit   0.13933690  0.08182879   1.7028  0.088608 .
## gendFemale:Walk      0.03570936  0.07528591   0.4743  0.635274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -5744.4
## McFadden R^2:  0.0075572
## Likelihood ratio test : chisq = 87.485 (p.value = < 2.22e-16)
```

Evaluate the model. X^2(9) = 87.485, p<.001, <- chisq, 9 is from the number of coefficients McFadden = Pseudo R^2

Which are significant predictors?

Let's turn again to the presentation.

Remember that we can report the coefficients as odds ratios rather than as differences in utility.

```
data.frame(exp(coef(m2)))
```

```
##                     exp.coef.m2..
## (Intercept):Carpool    0.08444081
## (Intercept):Transit    0.23916749
## (Intercept):Walk       0.11972144
## incom:Carpool          0.99610431
## incom:Transit          0.99458156
## incom:Walk             1.00398849
## gendFemale:Carpool     1.45175306
## gendFemale:Transit     1.14951131
## gendFemale:Walk        1.03635459
```

EXERCISE: Predicting Probabilities (Multinomial Choice)

Based on the result above, calculate the estimated probabilities that the surveyed individual would choose from mode choices. What percentage of the mode chosen did the model estimate correctly?

(1) Manual Calculation

This process is similar with the one we did in binomial model analysis, but it is a little bit trickier, so if you want to find an easier way, just move on to the next section.

Keep in mind that the reference level is Drive! You can make utility equations that each corresponds to Carpool, Walk, and Transit but not Drive, since it is the reference level. When calculating probability, you can think exp(Utility_Drive) as 1.

To be more specific, below is an example model where p3 is a reference level.You should compute lines in order. First calculate exponential values, then divide it by the denominator which is the sum or exponentials.

compute p1 = exp(p1) . compute p2 = exp(p2) . compute p3 = 1 .

compute denom = p1 + p2 + p3 .

compute p1 = p1 / denom . compute p2 = p2 / denom . compute p3 = p3 / denom .

```
# Your code here
# Drive is the ref level
cook2 <- cook %>%
  mutate(Utility_Carpool = -2.47170448 + -0.00390329 * incom + 0.37277183 * if_else(gend == 'Female', 1
         Utility_Walk = -2.12258755 + 0.00398056 * incom + 0.03570936 * if_else(gend == 'Female', 1, 0)
         Utility_Transit = -1.43059120 + -0.00543317 * incom + 0.13933690 * if_else(gend == 'Female', 1
  mutate(Prob_Carpool = exp(Utility_Carpool) / (exp(Utility_Carpool) + exp(Utility_Walk) + exp(Utility_
         Prob_Walk = exp(Utility_Walk) / (exp(Utility_Carpool) + exp(Utility_Walk) + exp(Utility_Transi
         Prob_Transit = exp(Utility_Transit) / (exp(Utility_Carpool) + exp(Utility_Walk) + exp(Utility_
         Prob_Drive = 1 / (exp(Utility_Carpool) + exp(Utility_Walk) + exp(Utility_Transit) + 1))
```

Check cook2. Now you have probabilities for all modes.

```
cook2
```

```
## # A tibble: 6,424 x 17
##    mode  tottr trpdur incom   dist hhlic hhveh gend  cars_per_driver race_eth
##    <fct> <dbl>  <dbl> <int>  <dbl> <dbl> <dbl> <fct>           <dbl> <fct>
##  1 Drive     1     15   150  6.38      2     3 Male              1.5 White
##  2 Drive     1     25   150  9.48      2     3 Fema~             1.5 Unknown
##  3 Drive     1      5   150  0.108     2     3 Fema~             1.5 Unknown
##  4 Drive     1     30    50  0.617     1     1 Fema~             1   White
##  5 Drive     1     50    75 18.1       2     2 Fema~             1   White
##  6 Drive     1     60    75 18.1       2     2 Fema~             1   White
##  7 Drive     1     15   100  2.61      2     2 Fema~             1   Unknown
##  8 Drive     1     15   100  2.61      2     2 Fema~             1   Unknown
##  9 Drive     1     25   150  4.12      2     2 Male              1   Unknown
## 10 Drive     1     15   100  1.69      2     2 Male              1   White
## # ... with 6,414 more rows, and 7 more variables: Utility_Carpool <dbl>,
## #   Utility_Walk <dbl>, Utility_Transit <dbl>, Prob_Carpool <dbl>,
## #   Prob_Walk <dbl>, Prob_Transit <dbl>, Prob_Drive <dbl>
```

Then to evaluate how many rows the model predicted correctly, we will add 'Expected' values and 'Correct' values as we did in the binomial model.This could be little tricky.

```
# Create a new column named Expected
# Put Mode with Largest Prob

# temporary variable to store Mode with largest prob
max <- vector(mode="numeric")

# iterate each row
for (i in 1:nrow(cook2)) {
   # input which prob is the largest - which.max brings the index of the largest column
   max[i] <- as.numeric(which.max(cook2[i,c(14:17)])[[1]])

   # input mode according to the index
   modes <- c("Carpool","Walk","Transit","Drive")
```

```
   max[i] <- modes[as.numeric(max[i])]
}

# attach to cook
cook2$Expected <- max

# delete rows with NA Expected
cook2 <- filter(cook2, !is.na(cook2$Expected))

# Create Correct col and summarise
cook2 <- mutate(cook2, Correct = if_else(mode == as.character(Expected), TRUE, FALSE))
prop.table(table(cook2$Correct))
```

```
##
##     FALSE      TRUE
## 0.2900062 0.7099938
```

(2) Using model$probabilities

When you view m2, you will find an attribute called probabilities. Use the console to view m2$probabilities. Each row contains the probability for each mode for each observation. It is the same thing as what we manually calculated in section (1). We can use this to simplify the code.

```
# store probabilities for each mode of each observation
correct <- m2$probabilities

# For each row, select the column with the largest value and extract the column name. Store it in a var
binaryCorrect <- colnames(correct)[apply(correct, 1, which.max)]

# See how it is distributed
table(cook$mode, binaryCorrect)
```

```
##          binaryCorrect
##           Drive
##   Walk      842
##   Drive    4561
##   Carpool   323
##   Transit   698
```

```
# attach it to the original dataframe
cook$Expected <- binaryCorrect
```

```
cook <- mutate(cook, Correct = if_else(mode == as.character(Expected), TRUE, FALSE))
prop.table(table(cook$Correct))
```

```
##
##     FALSE      TRUE
## 0.2900062 0.7099938
```