Homework 3: Data and multinomial choices

Tianpei Zhu

2022-03-18

Table of Contents

Data Import: Definitions

```
library(tidyverse)
datjss <- read.csv("datjss.csv", row.names=1)
# head(datjss)
datsss <- read.csv("datsss.csv", row.names=1)
# head(datsss)
datstu <- read.csv("datstu_v2-1.csv", row.names=1,
                   na.strings = c(" ", "NA", ""))
# head(datstu)
```

Exercise 1 Basic Statistics

Number of students, schools, programs

Number of students

```
# Exercise 1 Basic Statistics -----------------------------------------
----
# Number of students, schools, programs
number_of_students <- nrow(datstu)
number_of_students

## [1] 2198
```

Number of schools

```
number_of_schools <- datstu %>%
  select(contains("school")) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = !id, names_to = "Program") %>%
  distinct(value) %>% count()
number_of_schools$n

## [1] 550
```

Number of programs

```
# Number of programs
number_of_programs <- datstu %>%
  select(contains("pgm")) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = !id, names_to = "Program") %>%
  distinct(value) %>%
  na.omit() %>% count()
number_of_programs$n

## [1] 28
```

Number of choices (school, program) (Hint: Create a matrix of school, programs. Convert data from Wide to Long)

```
programs_only <- datstu %>%
  select(contains("pgm")) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = !id, names_to = "ProgramNo",
               values_to = "Program")

schools_only <- datstu %>%
  select(contains("school")) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = !id, names_to = "SchoolNo",
               values_to = "School")
#
number_of_choices <- cbind.data.frame(schools_only, programs_only) %>%
  select(School, Program) %>%
  group_by(School, Program) %>%
  count() %>%
  pivot_wider(names_from = Program, values_from = n, values_fill = 0)
number_of_choices
```

```
## # A tibble: 550 x 30
## # Groups:   School [550]
##    School Agriculture Business `General Arts` `General Science` `Hom
e Economics`
##     <int>       <int>    <int>          <int>             <int>
       <int>
##  1  10101           1        9              9                 3
           7
##  2  10102           0        0              2                 0
           0
##  3  10103           0        9              7                 1
           7
##  4  10104           0        0              0                 1
           0
##  5  10106           0        5              6                 0
           4
##  6  10107           0        5              5                 2
           9
##  7  10108           0        6              8                 2
           7
##  8  10109           0        0              0                 0
          17
##  9  10110           0        0              2                 0
           0
## 10  10112           0        1              1                 0
           0
## # ... with 540 more rows, and 24 more variables: `Visual Arts` <int>,
## #   Technical <int>, `NA` <int>, `Auto Body Works` <int>,
## #   `Mech. Eng. Craft Pract.` <int>, `Plumbing & Gas Fitting` <int>,
## #   `Small Eng. Repairs` <int>, `Welding & Fabrication` <int>,
## #   `Carpentry & Joinery` <int>, `Fashion Design` <int>,
## #   `Radio, TV & Electronics` <int>, `Electrical Installation Works`
 <int>,
## #   `Motor Vehicle Mech.` <int>, Catering <int>, Printing <int>, ...
```

Number of students applying to at least one senior high schools in the same district to home (Suppose students live in the same district to their junior high schools)

```
# Number of students applying to at least one senior high schools in
#the same district to home (Suppose students live in the same district
# to their junior high schools)
school_jss <- datstu %>%
  select(contains("school"), jssdistrict) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = schoolcode1:schoolcode6, names_to = "SchoolNo",
               values_to = "schoolcode")
school_jss_datsss <- merge(x = school_jss, datsss, by = 'schoolcode')
#
live_same_senior_junior_apply <- school_jss_datsss %>%
  select(jssdistrict, sssdistrict) %>%
  mutate(loc = ifelse( # Partial String Matching
    grepl(sssdistrict, jssdistrict, ignore.case = T),1,0)) %>%
  summarise(sum = sum(loc))
live_same_senior_junior_apply$sum

## [1] 7507
```

Number of students each senior high school admitted

I kept on requesting for the better data with score and rankplace columns that have values and nothing was done. The two columns have missing values all through. Its just NA from observation 1 to observation last. Anyway, i tried filling up with random numbers for the purppose of writing code.

```
# Delete this chunk if your data has the score and rankplace
# Number of students each senior high school admitted
set.seed(12)
# Had to fill up data with random values as not provided for score and
rankplace
datstu$score <- sample.int(n = 100, size = dim(datstu)[1], replace = T)
datstu$rankplace <- sample.int(n = 2, size = dim(datstu)[1], replace =
T)-1

school_score_rank <- datstu %>%
  select(contains("school"), rankplace, score) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = schoolcode1:schoolcode6, names_to = "SchoolNo",
               values_to = "School") %>%
  filter(rankplace==1)
senior_highschool_admitted <- school_score_rank %>%
  group_by(School) %>%
  summarise(n = n())
senior_highschool_admitted
```

```
## # A tibble: 521 x 2
##    School     n
##     <int> <int>
##  1  10101    15
##  2  10102     2
##  3  10103    14
##  4  10104     4
##  5  10106     9
##  6  10107    15
##  7  10108    11
##  8  10109    24
##  9  10110     2
## 10  10112     1
## # ... with 511 more rows
```

The cutoff of senior high schools (the lowest score to be admitted)

```
# The cutoff of senior high schools (the lowest score to be admitted)
senior_highschool_cutoff_low <- min(school_score_rank$score)
senior_highschool_cutoff_low
```

```
## [1] 1
```

The quality of senior high schools (the average score of students admitted)

```
# The quality of senior high schools (the average score of students adm
itted)
senior_highschool_cutoff_high <- mean(school_score_rank$score)
senior_highschool_cutoff_high
```

```
## [1] 51.29167
```

Exercice 2: data

```
# Exercice 2: data ---------------------------------------------------
----
#
programs_score_only <- datstu %>%
  select(contains("pgm"), score, jssdistrict, rankplace) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = choicepgm1:choicepgm6, names_to = "ProgramNo",
               values_to = "Program")

schools_only <- datstu %>%
  select(contains("school")) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = !id, names_to = "SchoolNo",
               values_to = "School")
#
school_data <- cbind.data.frame(
  schools_only, programs_score_only)
school_data$id <- NULL
school_data <- school_data %>%
  filter(rankplace == 1)
#
Q2_data <- merge(datjss, school_data, by = "jssdistrict")
#
school_cutoff_quality_size <- school_score_rank %>%
  group_by(School, ) %>%
  summarise(
    cutoff = min(score),
    quality = mean(score),
    size = n()
  )
#
Q2_data_1 <- merge(x = Q2_data, y = school_cutoff_quality_size,
                   by = "School")
head(Q2_data_1)
```

```
##   School        jssdistrict    point_x  point_y    SchoolNo score ra
nkplace
## 1  10101  Ga East (Abokobi) -0.2411459 5.721143 schoolcode1    10
      1
## 2  10101        Bia (Essam) -3.0435438 6.737386 schoolcode2    88
      1
## 3  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4    18
      1
## 4  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4    53
      1
## 5  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4   100
      1
## 6  10101 Ga West (Amasaman) -0.3975105 5.664688 schoolcode1    47
      1
##     id  ProgramNo     Program cutoff  quality size
## 1 1528 choicepgm1 Visual Arts      6 53.86667   15
## 2  246 choicepgm2    Business      6 53.86667   15
## 3 1279 choicepgm4    Business      6 53.86667   15
## 4  391 choicepgm4 Agriculture      6 53.86667   15
## 5 1193 choicepgm4    Business      6 53.86667   15
## 6 1799 choicepgm1 Visual Arts      6 53.86667   15
```

Exercise 3 Distance

```
# Exercise 3 ----------------------------------------------------------
----
#
```

```
Q3_data <- merge(x = Q2_data_1, y = datsss,
                 by.x = "School", by.y = "schoolcode")
Q3_data$dist_sss_jss = sqrt(
  (69.172*(Q3_data$ssslong- Q3_data$point_x)*cos(Q3_data$point_y/57.3))
^2 +
    (69.172*(Q3_data$ssslat = Q3_data$point_y))^2)
head(Q3_data)
```

```
##   School        jssdistrict    point_x  point_y    SchoolNo score ra
nkplace
## 1  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4     6
      1
## 2  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4     6
      1
## 3  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4     6
      1
## 4  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4     6
      1
## 5  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4     6
      1
## 6  10101 Accra Metropolitan -0.1971153 5.607396 schoolcode4     6
      1
##     id  ProgramNo      Program cutoff  quality size
## 1 1197 choicepgm4 General Arts      6 53.86667   15
## 2 1197 choicepgm4 General Arts      6 53.86667   15
## 3 1197 choicepgm4 General Arts      6 53.86667   15
## 4 1197 choicepgm4 General Arts      6 53.86667   15
## 5 1197 choicepgm4 General Arts      6 53.86667   15
## 6 1197 choicepgm4 General Arts      6 53.86667   15
##                                schoolname        sssdistrict    ssslo
ng   ssslat
## 1                                               Accra Metro
NA 5.607396
## 2                                               Accra Metro
NA 5.607396
## 3 EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN Accra Metropolitan -0.19711
53 5.607396
## 4 EBENEZER SENIOR HIGH. SCHOOL, DANSOMAN Accra Metropolitan -0.19711
53 5.607396
## 5                                               Accra Metro
NA 5.607396
## 6                                               Accra Metro
NA 5.607396
##   dist_sss_jss
## 1           NA
## 2           NA
## 3     387.8748
## 4     387.8748
## 5           NA
## 6           NA
```

Exercise 4 Dimensionality Reduction

```
# Exercice 4: Dimensionality Reduction --------------------------------
----
Q4_data <- Q3_data
```

Recode the schoolcode into its frst three digits (substr). Call this new variable scode rev.

```
# Recode the schoolcode into its frst three digits (substr).
# Call this new variable scode rev.
Q4_data$scode_rev <- str_sub(Q4_data$School, 1,4)
```

Recode the program variable into 4 categories: arts (general arts and visual arts), economics (business and home economics), science (general science) and others. Call this new variable pgm rev.

```
#
Q4_data$Program <- factor(
  Q4_data$Program,
  levels =
    c("Accounting","Agric. Mechanics","Agriculture",
      "Auto Body Works","Block Laying & Concreting","Business",
      "Carpentry & Joinery","Catering","Electrical Installation Works",
      "Electrical Mach. Rew.","Fashion Design","Furniture Craft",
      "General Arts","General Science","Home Economics",
      "Industrial Mechanics","Mech. Eng. Craft Pract.",
      "Motor Vehicle Mech.","Painting & Decorating",
      "Plumbing & Gas Fitting","Printing","Refrigeration & Air Cond.",
      "Small Eng. Repairs","Technical","Visual Arts","Welding & Fabrica
tion"),

  labels =
    c("Others","Others","Others", "Others","Others","economics",
  "Others","Others","Others", "Others","Others","Others","Arts","Scienc
e",
  "Economics","Others","Others","Others","Others","Others",
  "Others","Others","Others","Others","Arts","Others"))
Q4_data$pgm_rev <- Q4_data$Program
```

Create a new choice variable choice rev.

```
# Create a new choice variable choice rev.
Q4_data$choice_rev <- str_sub(Q4_data$ProgramNo, -1)
```

Recalculate the cutoff and the quality for each recoded choice.

```
# Recalculate the cutoff and the quality for each recoded choice.
recoded_choices_data <- Q4_data %>%
  group_by(scode_rev, pgm_rev, choice_rev) %>%
  summarise(
    cutoff = min(score),
    quality = mean(score),
    size = n()
  )
#
head(recoded_choices_data)

## # A tibble: 6 x 6
## # Groups:   scode_rev, pgm_rev [2]
##   scode_rev pgm_rev   choice_rev cutoff quality  size
##   <chr>     <fct>     <chr>       <int>   <dbl> <int>
## 1 1001      Others    1              24    28.9    24
## 2 1001      Others    2              30    44.8    12
## 3 1001      Others    3              12    41.5    18
## 4 1001      Others    4              33    73.2    18
## 5 1001      economics 1              19    43.5    33
## 6 1001      economics 2               7    46.3     9

new_data <- merge(
  Q4_data, recoded_choices_data,
  by = c("scode_rev","pgm_rev","choice_rev"))
```

Exercise 5: First Model

```
# Exercise 5: First Model ---------------------------------------------
----
Q5_data <- Q4_data[complete.cases(Q4_data),]
Q5_data <- distinct(Q5_data)
new_data <- Q5_data
new_data$choice_rev <- factor(
  new_data$choice_rev,
  levels = c("1","2","3","4","5","6"),
  ordered = T)

# individual identifier: id
# Decision variable: choice_rev
# choice_rev alternatives: "4" "1" "2" "3" "6" "5"
# Independent variable: score
# For student i, being assigned to choice_rev j
# choice_rev_{ij} = c_j + \gamma score + \epsilon_{ij}
# In order to run the ordinal logistic regression model,
# we need the polr function from the MASS package
library(MASS)
library(margins)
library(effects)

# Build ordinal logistic regression model
OLRmodel_5 <- polr(choice_rev ~ score , data = new_data, Hess = T)
#summary(OLRmodel_5)
```

Estimate parameters and compute the marginal effect of the proposed model

```
# We can use the coef() function to check the parameter estimates
(OLRestimates_5 <- coef(summary(OLRmodel_5)))

##                 Value   Std. Error      t value
## score  4.835238e-05 0.0007525413   0.06425212
## 1|2   -1.573612e+00 0.0504167284 -31.21210903
## 2|3   -6.566960e-01 0.0464690555 -14.13189866
## 3|4    5.276640e-02 0.0457483818   1.15340466
## 4|5    7.715760e-01 0.0467263509  16.51265221
## 5|6    1.650020e+00 0.0509182894  32.40525132

# Marginal Effects
summary(margins(OLRmodel_5))

##  factor     AME     SE    z      p   lower   upper
##   score -0.0000 0.0000 -Inf 0.0000 -0.0000 -0.0000
```

Exercise 6 Second Model

```
# Question 6 ----------------------------------------------------------
----
new_data$Program <- as.factor(new_data$Program)

# Build ordinal logistic regression model
OLRmodel_6 <- polr(
  choice_rev ~ quality + dist_sss_jss + cutoff + size + Program,
  data = new_data, Hess = T)
# summary(OLRmodel_6)
# We can use the coef() function to check the parameter estimates
(OLRestimates_6 <- coef(summary(OLRmodel_6)))

##                           Value   Std. Error    t value
## quality            0.0044151572 0.0027036252  1.6330508
## dist_sss_jss       0.0008504690 0.0001957446  4.3447897
## cutoff            -0.0002441114 0.0023814613 -0.1025049
## size               0.0191126698 0.0015565284 12.2790371
## Prograeconomics    0.0175270027 0.0695687342  0.2519379
## ProgramArts       -0.0250974983 0.0561861502 -0.4466848
## ProgramScience    -0.5264465151 0.1318539470 -3.9926489
## ProgramEconomics  -0.0127388438 0.0681722315 -0.1868626
## 1|2               -0.5901117518 0.1767874858 -3.3379724
## 2|3                0.3357347881 0.1759956877  1.9076308
## 3|4                1.0564024132 0.1761774231  5.9962417
## 4|5                1.7945051703 0.1770961359 10.1329437
## 5|6                2.6985458998 0.1792356230 15.0558569
```

```
#
# marginal_effects(OLRmodel_6)
summary(margins(OLRmodel_6))

##             factor     AME     SE       z      p   lower   upper
##             cutoff  0.0000 0.0003  0.1027 0.9182 -0.0006  0.0007
##       dist_sss_jss -0.0001 0.0000 -5.5466 0.0000 -0.0002 -0.0001
##        ProgramArts  0.0035 0.0079  0.4421 0.6584 -0.0120  0.0189
##   Programeconomics -0.0024 0.0095 -0.2536 0.7998 -0.0210  0.0162
##   ProgramEconomics  0.0018 0.0095  0.1861 0.8524 -0.0168  0.0203
##     ProgramScience  0.0852 0.0258  3.3010 0.0010  0.0346  0.1358
##            quality -0.0006 0.0003 -1.8693 0.0616 -0.0013  0.0000
##               size -0.0027 0.0003 -8.1280 0.0000 -0.0033 -0.0020
```

Exercise 7 Counterfactual simulations

We recommend model 2 given that the predictors are not only limited to the test scores but include information with regard to the quality of school, the distance, programs offered and the availability of the spaces during selection.

Calculate choice probabilities under the appropriate model

```
probabilitie_s <- predict(OLRmodel_6, type = "probs")
head(probabilitie_s)

##           1         2         3         4         5          6
## 1 0.1949840 0.1844154 0.1774955 0.1675608 0.1420851 0.13345927
## 2 0.1883803 0.1810358 0.1769367 0.1695134 0.1456677 0.13846617
## 3 0.1944231 0.1841344 0.1774545 0.1677291 0.1423854 0.13387347
## 4 0.2849227 0.2164957 0.1725835 0.1382095 0.1021809 0.08560766
## 5 0.1735724 0.1728797 0.1750338 0.1736199 0.1540685 0.15082565
## 6 0.1883803 0.1810358 0.1769367 0.1695134 0.1456677 0.13846617
```

Simulate how these choice probabilities change when these choices are excluded.

```
new_data <- subset(new_data, Program != "Others")
new_data$Program <- as.factor(new_data$Program)

# Build ordinal logistic regression model
OLRmodel_7 <- polr(
  choice_rev ~ quality + dist_sss_jss + cutoff + size + Program,
  data = new_data, Hess = T)
# summary(OLRmodel_6)
probabilitie_s2 <- predict(OLRmodel_7, type = "probs")
head(probabilitie_s2)

##           1         2         3         4         5          6
## 1 0.1919330 0.1836431 0.1723034 0.1596571 0.1517027 0.14076063
## 2 0.1847264 0.1798588 0.1715908 0.1615095 0.1557541 0.14656040
## 3 0.1916475 0.1834967 0.1722790 0.1597321 0.1518611 0.14098355
## 4 0.2787167 0.2158570 0.1688902 0.1339344 0.1111177 0.09148395
## 5 0.1771005 0.1756464 0.1706082 0.1633654 0.1601635 0.15311590
## 6 0.1847264 0.1798588 0.1715908 0.1615095 0.1557541 0.14656040
```