Assignment 1

Xiaoshu Gui

1/24/2019

Setup

```
# Load packages
library(dplyr)
library(purrr)
library(magrittr)
library(tidyr)
library(tibble)
library(stringr)
library(reshape2)
# Read data
jss <- read.csv("datjss.csv", header=T, sep=",")
sss <- read.csv("datsss.csv", header=T, sep=",")
stu <- read.csv("datstu.csv", header=T, sep=",", na.strings=c("","."," ","NA"))
# Note that there are empty cells in stu, replace them with NAs first.

# Reshape school choices from wide to long.
school <- stu %>%
  select(X:schoolcode6, rankplace) %>%
  gather(key = 'school', 'schoolcode', -c(X:male, rankplace)) %>%
  mutate(choice = str_match(school, "[1-6]") %>% as.numeric())

# Reshape program choices from wide to long
pgm <- stu %>%
  select(X, choicepgm1:rankplace) %>%
  gather(key = 'program', 'choicepgm', -c(X, jssdistrict, rankplace)) %>%
  mutate(choice = str_match(program, "[1-6]") %>% as.numeric())

# Create a school_program level dataframe: df_schpgm
df_schpgm <- left_join(school, pgm, by = c("X", "choice", "rankplace")) %>%
   mutate(choice_sp = paste(schoolcode, choicepgm, sep = ",")) # id var is choice of school_pgm

# clean datsss
ss <- sss %>%
  select(-X) %>%
  mutate(schoolname = str_remove_all(schoolname, "\\d")) %>% # clean school names
  mutate(schoolname = str_remove_all(schoolname, "\\W")) %>%
  distinct(schoolcode, .keep_all = T) # Note that some school names do not match their
# school code, I will only use school codes as id variable.
```

Exercise 1

- Number of students: 340823

```
length(unique(stu$X))
```

```
## [1] 340823
```

- Number of schools: 898

```
length(unique(sss$schoolcode))    # or length(unique(ss$schoolcode))
```

```
## [1] 898
```

- Number of programs: 32

```
length(unique(pgm$choicepgm))
```

```
## [1] 33
```

Note that there's NA in unique choicepgms, so the number of programs is: 33 - 1 = 32

- Number of choices (school, program):

```
# drop NAs in school and program choices
stu1 <- df_schpgm %>%
  drop_na(., c(schoolcode,choicepgm))

length(stu1$choice_sp) # number of all choices of school and program
```

```
## [1] 2006470
```

```
length(unique(stu1$choice_sp)) # number of unique choices of school and program
```

```
## [1] 2773
```

- Missing test score: 179887 (number of NAs in "score" in datstu)

```
summary(stu$score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   158.0   252.0   283.0   291.1   324.0   469.0  179887
```

- Apply to the same school (different programs): *stu2* shows the number of students apply to the same school.

```
stu2 <- school %>%
  drop_na(schoolcode) %>%
  group_by(schoolcode) %>%
  summarise(count= n())
```

- Apply to less than 6 choices: 20988

```
# rule out NAs in choicepgm, leaving us students who apply 6 choices
stu3 <- drop_na(stu, starts_with("choice"))
# Then the number of students who apply less than 6 choices should be
# total number of students minus those who apply 6 choices:
340823 - count(stu3)
```

```
##       n
## 1 20988
```

Excercise 2 Data

The school level dataset is given by *stu5*

```
df <- stu1 %>%
  select(X, choice_sp, schoolcode, score, rankplace)
```

```r
stu4 <- left_join(df, ss, by = "schoolcode") # merge ss with df

admin <- stu4 %>%
  filter(!is.na(rankplace)) %>%
  filter(rankplace < 7) %>%  # get a subset of admitted students
  group_by(choice_sp) %>%
  summarise(cutoff = min(score), quality = mean(score), size = n())

stu5 <- left_join(stu4, admin, by = "choice_sp") %>%
  select(choice_sp, ssdistrict:size) %>%
  distinct()
```

Excercise 3 Distance

The distance between junior high school and senior high school is given by the variable *dis_sss_jsss* in dataset *distance*.

```r
# create an individual level dataset where each row is a pair of individual and school choice
id_ss <- stu %>%
  select(X, schoolcode1:schoolcode6, jssdistrict, rankplace, score) %>%
  gather(key = 'seniorhigh', 'schoolcode', -X, -jssdistrict, -rankplace, -score) %>%
  mutate(id = paste(X, schoolcode, sep = ",")) # id variable is individual_ss

# merge jss and sss with id_ss to get the coordinates of junior and senior high schools
js <- jss %>% select(-X)
colnames(js)[2:3] <- c("jsslong", "jsslat")

distance <- left_join(id_ss, js, by = "jssdistrict") %>%
  left_join(., ss, by = "schoolcode") %>%
  mutate(dist_sss_jss = sqrt((69.172*(ssslong - jsslong)*cos(jsslat/57.3))^2
                             + (69.172*(ssslat - jsslat))^2)) # calculate distance using the formula
```

Exercise 4 Descriptive Characteristics

avg_sd is the table for average and sd of cutoff, quality and distance by ranked choice.

```r
# get cutoff and quality from exercise 2
stu6 <- left_join(stu4, admin, by = "choice_sp") %>%
  select(choice_sp, schoolcode, ssdistrict:size, rankplace, score) %>%
  filter(!is.na(rankplace) & rankplace < 7) %>% # add rankplace as the id variable to stu5 in exercise
  distinct()

cutoff <- stu6 %>%
  select(choice_sp, cutoff, rankplace) %>%
  group_by(rankplace) %>%
  #spread(., choice_sp, cutoff) %>%
  summarise(mean_cutoff = mean(cutoff), sd_cutoff = sd(cutoff))

quality <- stu6 %>%
  select(choice_sp, quality, rankplace) %>%
  group_by(rankplace) %>%
  summarise(mean_quality = mean(quality), sd_quality = sd(quality))
```

```
# get distance from exercise 3
dist <- distance %>%
  select(id, dist_sss_jss, rankplace) %>%
  filter(!is.na(rankplace) & !is.na(dist_sss_jss)) %>% # drop NAs in distance and rankplacce
  filter(rankplace < 7) %>%
  group_by(rankplace) %>%
  summarise(mean_distance = mean(dist_sss_jss), sd_distance = sd(dist_sss_jss))

avg_sd <- left_join(cutoff, quality, by = 'rankplace') %>%
  left_join(., dist, by = 'rankplace')
```

Redo this part, differentiating by student test score quantiles. *avg_sd_quantile* returns the result.

```
# divide the whole sample into 4 subsamples by score quantiles.
stu7 <- stu6 %>%
  filter(!is.na(score)) %>%
  mutate(quantile = ntile(score, 4)) # adding a score quantile variable

cutoff_q <- stu7 %>%
  select(choice_sp, cutoff, quantile, rankplace) %>%
  group_by(quantile, rankplace) %>%
  summarise(mean_cutoff = mean(cutoff), sd_cutoff = sd(cutoff))

quality_q <- stu7 %>%
  select(choice_sp, quality, rankplace, quantile) %>%
  group_by(quantile, rankplace) %>%
  summarise(mean_quality = mean(quality), sd_quality = sd(quality))

dist_q <- distance %>%
  filter(!is.na(score)) %>%
  mutate(quantile = ntile(score, 4)) %>%
  select(id, dist_sss_jss, rankplace, quantile) %>%
  filter(!is.na(rankplace) & !is.na(dist_sss_jss) & rankplace < 7) %>%
  group_by(quantile, rankplace) %>%
  summarise(mean_distance = mean(dist_sss_jss), sd_distance = sd(dist_sss_jss))

avg_sd_quantile <- left_join(cutoff_q, quality_q, by = c("quantile", "rankplace")) %>%
  left_join(., dist_q, by = c("quantile","rankplace"))
```

Excercise 5

stu9 groups schools by decile of selectivity. **num_group** in *group_school* presents the number of groups in each invidual's application.

```
stu8 <- school %>%
   filter(!is.na(rankplace) & !is.na(score) & rankplace < 7)

stu9 <- stu8 %>%
  group_by(schoolcode) %>%
  summarise(cutoff = min(score)) %>%
  inner_join(stu8, by = 'schoolcode') %>%
  mutate(decile = ntile(cutoff, 10)) %>% # decile of cutoffs
  group_by(decile) # group schools by decile of cutoffs.
```

```
group_school <- stu9 %>%
   select(X, school, decile) %>%
   spread(school, decile) %>%
   mutate(num_group = apply(., 1, function(x) length(unique(x))-1))# Note that X also constitutes
  # a unqiue number-- minus one to get the number of groups in the application.
```

Redo this part by student test score quantiles:

```
stu10 <- stu %>%
  filter(!is.na(score)) %>%
  mutate(quantile = ntile(score, 4)) %>%
  select(X:schoolcode6, rankplace, quantile) %>%
  gather(key = 'school', 'schoolcode', -c(X:male, rankplace, quantile)) %>%
  mutate(choice = str_match(school, "[1-6]") %>% as.numeric()) %>%
  filter(!is.na(rankplace) & !is.na(score) & rankplace < 7) %>%
  group_by(quantile, schoolcode) %>%
  summarise(cutoff = min(score)) %>%
  inner_join(stu8, by = 'schoolcode') %>%
  mutate(decile = ntile(cutoff, 10)) %>% # decile of cutoffs
  group_by(decile)

group_school_d <- stu10 %>%
  select(X, school, decile, quantile) %>%
  spread(school, decile) %>%
  mutate(num_group = apply(., 1, function(x) length(unique(x))-1)) %>%
  group_by(quantile)
```