ECON-613 HW #1

Min Chul Kim

2019-01-27

Here, I am calling all the packages I am going to use in this assignment.

Here, I read data.

```r
#Reading Data
datjss <- read.csv("datjss.csv")
datsss <- read.csv("datsss.csv")
datstu <- read.csv("datstu.csv")
```

Here, I am creating a table function which I will use later to present obtained statistics in a nice format.

```r
table = function(df, dig, capt){
  kable(df,
        digits = dig,
        booktabs = TRUE,
        caption = capt
        )
}
```

Exercise 1

```r
datstu = na.omit(datstu, col = c("score", "rankplace"))

#Number of School
School_Num = datstu[, paste("schoolcode", 1:6, sep = "")] %>%
  unlist() %>%
  unique() %>%
  na.omit() %>%
  length()

#Number of Program
Programs_Num = datstu[, paste("choicepgm", 1:6, sep = "")] %>%
  unlist() %>%
  unique() %>%
  na.omit() %>%
  length()

#Number of Choices
School = datstu[, paste("schoolcode", 1:6, sep = "")]
Choices = datstu[, paste("choicepgm", 1:6, sep = "")]

Total_Choices = data.frame(
  Choices1 = paste(School$schoolcode1, Choices$choicepgm1),
  Choices2 = paste(School$schoolcode2, Choices$choicepgm2),
  Choices3 = paste(School$schoolcode3, Choices$choicepgm3),
  Choices4 = paste(School$schoolcode4, Choices$choicepgm4),
  Choices5 = paste(School$schoolcode1, Choices$choicepgm5),
  Choices6 = paste(School$schoolcode1, Choices$choicepgm6)
) %>%
  unlist() %>%
  unique() %>%
  na.omit() %>%
  length()

#Missing Data
Missing_Test = datstu$score %>%
  summary() %>%
  .["NA's"] %>%
  as.integer()

#Apply to the Same School
No_NA_School = apply(School, 1, FUN = na.omit) %>%
  map(length) %>%
  unlist()

Num_School_Applied = apply(School, 1, FUN = unique) %>%
  map(na.omit) %>%
  map(length) %>%
  unlist()

Same_School = ifelse(Num_School_Applied < No_NA_School, "Yes", "No") %>%
  as.data.frame()

#Apply to less than 6 Choices
Num_Choices_Applied = apply(Choices, 1, na.omit) %>%
  map(length) %>%
  unlist()

Less_Choices = ifelse(Num_Choices_Applied < 6, "Yes", "No") %>%
  as.data.frame()
colnames(Less_Choices) = "Apply to less than 6 programs?"

#Dataframe of Number of Students, Number of Schools, Number of Programs, etc.
Exercise1_df = data_frame(
  nrow(datstu),
  School_Num,
  Programs_Num,
  Total_Choices,
  Missing_Test
)
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
colnames(Exercise1_df) = c("Num of Student", "Num of School", "Num of Programs",
                           "Number of Choices", "Missing Test Score")

#Summary Tables
#table(Exercise1_df, 4,
#      "Summary Table of Statistics Other Than Same School and Less than 6 Choices")
#table(Same_School, 4, "Table of Apply to the Same School")
```

```
#table(Less_Choices, 4, "Table of Apply to less than 6 Choices")
```

Exercise 2

```
datstu2 = datstu %>%
  mutate(
    choice1 = paste(schoolcode1, choicepgm1, sep = "_"),
    choice2 = paste(schoolcode2, choicepgm2, sep = "_"),
    choice3 = paste(schoolcode3, choicepgm3, sep = "_"),
    choice4 = paste(schoolcode4, choicepgm4, sep = "_"),
    choice5 = paste(schoolcode5, choicepgm5, sep = "_"),
    choice6 = paste(schoolcode6, choicepgm6, sep = "_"),
  )

School_Level_Dataset = data.frame(
  c(datstu2$choice1, datstu2$choice2, datstu2$choice3,
    datstu2$choice4, datstu2$choice5, datstu2$choice6)
)
colnames(School_Level_Dataset) = "School_Programs"

School_and_Program = School_Level_Dataset %>%
  separate(School_Programs, c("School", "Choices"), sep = "_")

#Cutoff
a = list()
for(i in 1:6){
 a[[i]] = full_join(School_Level_Dataset,
                     datstu2[, c(paste("choice", i, sep = ""), "rankplace", "score")],
                     by = c("School_Programs" = paste("choice", i, sep = ""))) %>%
  filter(rankplace == i) %>%
  group_by(School_Programs) %>%
  summarise(min = min(score))
}
```

```
## Warning: Column `School_Programs`/`choice1` joining factor and character
## vector, coercing into character vector

## Warning: Column `School_Programs`/`choice2` joining factor and character
## vector, coercing into character vector

## Warning: Column `School_Programs`/`choice3` joining factor and character
## vector, coercing into character vector

## Warning: Column `School_Programs`/`choice4` joining factor and character
## vector, coercing into character vector

## Warning: Column `School_Programs`/`choice5` joining factor and character
## vector, coercing into character vector

## Warning: Column `School_Programs`/`choice6` joining factor and character
## vector, coercing into character vector
```

```
b = list()
b[[1]] = full_join(a[[1]], a[[2]], by = "School_Programs")
for(i in 2:5){
```

```r
  b[[i]] = full_join(b[[i-1]], a[[i+1]], by = "School_Programs")
}

School_Level_Dataset = full_join(School_Level_Dataset, b[[5]], by = "School_Programs")
```

```
## Warning: Column `School_Programs` joining factor and character vector,
## coercing into character vector
```

```r
School_Level_Dataset = School_Level_Dataset %>%
  mutate(
    cutoff = apply(School_Level_Dataset[, names(School_Level_Dataset) != "School_Programs"],
                  1,
                  function(x) ifelse(all(is.na(x)), NA, min(x, na.rm = TRUE)))
  )

School_Level_Dataset = School_Level_Dataset %>%
  select(School_Programs, cutoff)

#Quality
e = list()
for(i in 1:6){
  e[[i]] = full_join(School_Level_Dataset,
                     datstu2[, c(paste("choice", i, sep = ""), "rankplace", "score")],
                     by = c("School_Programs" = paste("choice", i, sep = ""))) %>%
    filter(rankplace == i) %>%
    na.omit() %>%
    group_by(School_Programs) %>%
    summarise(
      Total = sum(score),
      count = n(),
      )
}

f = list()
f[[1]] = full_join(e[[1]], e[[2]], by = "School_Programs")
for(i in 2:5){
  f[[i]] = full_join(f[[i-1]], e[[i+1]], by = "School_Programs")
}
colnames(f[[5]]) = c("School_Programs", "Total1", "Count1", "Total2", "Count2",
                     "Total3", "Count3", "Total4", "Count4", "Total5", "Count5",
                     "Total6", "Count6")
f[[5]] = f[[5]] %>%
  mutate(
    quality = rowSums(f[[5]][, seq(2, 13, by = 2)], na.rm = TRUE) /
      rowSums(f[[5]][, seq(3, 13, by = 2)], na.rm = TRUE)
  )

School_Level_Dataset = full_join(School_Level_Dataset, f[[5]], by = "School_Programs")
School_Level_Dataset = School_Level_Dataset %>%
  select(School_Programs, cutoff, quality) %>%
  arrange(School_Programs) %>%
  unique() %>%
  na.omit()
```

```
#Size
h = list()
h[[1]] = datstu2 %>% filter(rankplace == 1) %>% select(choice1) %>%
  group_by(choice1) %>% summarise(count = n())
h[[2]] = datstu2 %>% filter(rankplace == 2) %>% select(choice2) %>%
  group_by(choice2) %>% summarise(count = n())
h[[3]] = datstu2 %>% filter(rankplace == 3) %>% select(choice3) %>%
  group_by(choice3) %>% summarise(count = n())
h[[4]] = datstu2 %>% filter(rankplace == 4) %>% select(choice4) %>%
  group_by(choice4) %>% summarise(count = n())
h[[5]] = datstu2 %>% filter(rankplace == 5) %>% select(choice5) %>%
  group_by(choice5) %>% summarise(count = n())
h[[6]] = datstu2 %>% filter(rankplace == 6) %>% select(choice6) %>%
  group_by(choice6) %>% summarise(count = n())
for(i in 1:6){
  colnames(h[[i]]) = c("School_Programs", "count")
}

k = list()
k[[1]] = full_join(h[[1]], h[[2]], by = "School_Programs")
for(i in 2:5){
  k[[i]] = full_join(k[[i-1]], h[[i+1]], by = "School_Programs")
}

k[[5]] = k[[5]] %>%
  mutate(
    size = apply(k[[5]][, names(k[[5]]) != "School_Programs"],
                 1,
                 function(x) ifelse(all(is.na(x)), NA, sum(x, na.rm = TRUE)))
  )

School_Level_Dataset = full_join(School_Level_Dataset, k[[5]], by = "School_Programs")
School_Level_Dataset = School_Level_Dataset %>%
  select(School_Programs, cutoff, quality, size)

School_Level_Dataset = School_Level_Dataset %>%
  separate(School_Programs, c("schoolcode", "program"), "_") %>%
  mutate(
    School_Programs = paste(schoolcode, program, sep = "_")
  )
School_Level_Dataset = School_Level_Dataset[, c("School_Programs",
                "schoolcode", "program", "cutoff", "quality", "size")]

#District, Lat, Long
catalog = datsss %>%
  select(schoolcode, sssdistrict, ssslat, ssslong) %>%
  unique() %>%
  na.omit() %>%
  arrange(schoolcode) %>%
  as.data.frame()

catalog$schoolcode = catalog$schoolcode %>% as.character()
School_Level_Dataset = full_join(School_Level_Dataset, catalog, by = "schoolcode")
```

Exercise 3

```
distance = c()
distance_list = list()

dist_df = data.frame(
  datstu2$schoolcode1 %>% as.character(),
  datstu2$schoolcode2 %>% as.character(),
  datstu2$schoolcode3 %>% as.character(),
  datstu2$schoolcode4 %>% as.character(),
  datstu2$schoolcode5 %>% as.character(),
  datstu2$schoolcode6 %>% as.character(),
  datstu2$jssdistrict
)
colnames(dist_df) = c(paste("schoolcode", 1:6, sep = ""), "jssdistrict")
colnames(datjss) = colnames(datjss) %>% as.character()

dist_df = left_join(dist_df, datjss, by = "jssdistrict")
```

```
## Warning: Column `jssdistrict` joining factors with different levels,
## coercing to character vector
```

```
dist_df = dist_df %>%
  select(-X)

dist = function(i){
  return(sqrt( (69.172 * (dist_df$ssslong[i] - dist_df$point_x[i]) *
           cos(dist_df$point_y[i] / 57.3))^2 +
          ( 69.172 *(dist_df$ssslat[i] - dist_df$point_y[i]))^2 ))
}

schoolcode_vector = paste("schoolcode", 1:6, sep = "")

for(j in 1:6){
  colnames(dist_df)[j] = "schoolcode"
  dist_df = join(dist_df, catalog, by = "schoolcode")

  for(i in 1:nrow(dist_df)){
    distance[i] = dist(i)
  }
  distance_list[[j]] = distance
  colnames(dist_df)[j] = paste("schoolcode", j, sep = "")

  dist_df = dist_df %>%
    select(-c(sssdistrict, ssslat, ssslong))
}

dist_df = dist_df %>%
  mutate(
    distance1 = distance_list[[1]] %>% unlist() %>% as.double(),
    distance2 = distance_list[[2]] %>% unlist() %>% as.double(),
    distance3 = distance_list[[3]] %>% unlist() %>% as.double(),
    distance4 = distance_list[[4]] %>% unlist() %>% as.double(),
    distance5 = distance_list[[5]] %>% unlist() %>% as.double(),
    distance6 = distance_list[[6]] %>% unlist() %>% as.double()
  ) %>%
  select(jssdistrict, schoolcode1, distance1, schoolcode2, distance2, schoolcode3, distance3,
         schoolcode4, distance4, schoolcode5, distance5, schoolcode6, distance6)
```

Exercise 4

```
mean_cutoff = function(i, var){
  return(datstu2 %>%
           filter(rankplace == i) %>%
           group_by(!!enquo(var)) %>%
           summarise(min = min(score)) %>%
           select(min) %>%
           unlist() %>%
           as.integer() %>%
           mean()
  )
}

sd_cutoff = function(i, var){
  return(datstu2 %>%
           filter(rankplace == i) %>%
           group_by(!!enquo(var)) %>%
           summarise(min = min(score)) %>%
           select(min) %>%
           unlist() %>%
           as.integer() %>%
           sd()
  )
}

mean_quality = function(i, var){
  return(datstu2 %>%
           filter(rankplace == i) %>%
           group_by(!!enquo(var)) %>%
           summarise(mean = mean(score)) %>%
           select(mean) %>%
           unlist() %>%
           as.integer() %>%
           mean()
  )
}

sd_quality = function(i, var){
  return(datstu2 %>%
           filter(rankplace == i) %>%
           group_by(!!enquo(var)) %>%
           summarise(mean = mean(score)) %>%
           select(mean) %>%
           unlist() %>%
           as.integer() %>%
           sd()
```

```r
  )
}
#Cutoff
cutoff_stats = data.frame(
  seq(1, 6),
  c(mean_cutoff(1, choice1), mean_cutoff(2, choice2), mean_cutoff(3, choice3),
    mean_cutoff(4, choice4), mean_cutoff(5, choice5), mean_cutoff(6, choice6)),
  c(sd_cutoff(1, choice1), sd_cutoff(2, choice2), sd_cutoff(3, choice3),
    sd_cutoff(4, choice4), sd_cutoff(5, choice5), sd_cutoff(6, choice6))
)
colnames(cutoff_stats) = c("rank", "mean", "sd")

#Quality
quality_stats = data.frame(
  seq(1, 6),
  c(mean_quality(1, choice1), mean_quality(2, choice2), mean_quality(3, choice3),
    mean_quality(4, choice4), mean_quality(5, choice5), mean_quality(6, choice6)),
  c(sd_quality(1, choice1), sd_quality(2, choice2), sd_quality(3, choice3),
    sd_quality(4, choice4), sd_quality(5, choice5), sd_quality(6, choice6))
)
colnames(quality_stats) = c("rank", "mean", "sd")

#Distance
dist_df$schoolcode1 = dist_df$schoolcode1 %>% as.character() %>% as.numeric()
dist_df$schoolcode2 = dist_df$schoolcode2 %>% as.character() %>% as.numeric()
dist_df$schoolcode3 = dist_df$schoolcode3 %>% as.character() %>% as.numeric()
dist_df$schoolcode4 = dist_df$schoolcode4 %>% as.character() %>% as.numeric()
dist_df$schoolcode5 = dist_df$schoolcode5 %>% as.character() %>% as.numeric()
dist_df$schoolcode6 = dist_df$schoolcode6 %>% as.character() %>% as.numeric()

mean_distance = function(i, var, var2){
  datstu2 %>%
    filter(rankplace == i) %>%
    select(!!enquo(var)) %>%
    left_join(dist_df, by = var) %>%
    select(var2) %>%
    unlist() %>%
    as.numeric() %>%
    na.omit() %>%
    mean()
}

sd_distance = function(i, var, var2){
  datstu2 %>%
    filter(rankplace == i) %>%
    select(!!enquo(var)) %>%
    left_join(dist_df, by = var) %>%
    select(var2) %>%
    unlist() %>%
    as.numeric() %>%
    na.omit() %>%
    sd()
}
```

```r
distance_stats = data.frame(
  c(
    mean_distance(1, "schoolcode1", "distance1"),
    mean_distance(2, "schoolcode2", "distance2"),
    mean_distance(3, "schoolcode3", "distance3"),
    mean_distance(4, "schoolcode4", "distance4"),
    mean_distance(5, "schoolcode5", "distance5"),
    mean_distance(6, "schoolcode6", "distance6")
  ),
  c(
    sd_distance(1, "schoolcode1", "distance1"),
    sd_distance(2, "schoolcode2", "distance2"),
    sd_distance(3, "schoolcode3", "distance3"),
    sd_distance(4, "schoolcode4", "distance4"),
    sd_distance(5, "schoolcode5", "distance5"),
    sd_distance(6, "schoolcode6", "distance6")
  )
)
colnames(distance_stats) = c("mean", "sd")
```

```r
quan_cutoff = function(i, var){
  temp = datstu2 %>%
    filter(rankplace == i) %>%
    group_by(!!enquo(var)) %>%
    summarise(min = min(score))

  quan_df = quantile(temp$min, seq(0, 1, by = 0.25)) %>%
    as.data.frame()
  colnames(quan_df) = "score"

  return(quan_df)
}

quan_quality = function(i, var){
  temp = datstu2 %>%
    filter(rankplace == i) %>%
    group_by(!!enquo(var)) %>%
    summarise(mean = mean(score))

  quan_df = quantile(temp$mean, seq(0, 1, by = 0.25)) %>%
    as.data.frame()
  colnames(quan_df) = "score"

  return(quan_df)
}

quan_distance = function(i, var, var2){
  temp = datstu2 %>%
    filter(rankplace == i) %>%
    select(!!enquo(var)) %>%
    left_join(dist_df, by = var) %>%
    select(var2) %>%
    unlist() %>%
    as.numeric() %>%
    na.omit()

  quan_df = quantile(temp, seq(0, 1, by = 0.25)) %>%
    as.data.frame()
  colnames(quan_df) = "score"

  return(quan_df)
}

quantile_cutoff_df = data.frame(
  quan_cutoff(1, choice1),
  quan_cutoff(2, choice2),
  quan_cutoff(3, choice3),
  quan_cutoff(4, choice4),
  quan_cutoff(5, choice5),
  quan_cutoff(6, choice6)
)
colnames(quantile_cutoff_df) = paste("rankchoice", 1:6, sep = "")

quantile_quality_df = data.frame(
  quan_quality(1, choice1),
  quan_quality(2, choice2),
  quan_quality(3, choice3),
  quan_quality(4, choice4),
  quan_quality(5, choice5),
  quan_quality(6, choice6)
)
colnames(quantile_quality_df) = paste("rankchoice", 1:6, sep = "")

quantile_distance_df = data.frame(
  quan_distance(1, "schoolcode1", "distance1"),
  quan_distance(2, "schoolcode2", "distance2"),
  quan_distance(3, "schoolcode3", "distance3"),
  quan_distance(4, "schoolcode4", "distance4"),
  quan_distance(5, "schoolcode5", "distance5"),
  quan_distance(6, "schoolcode6", "distance6")
)
colnames(quantile_distance_df) = paste("rankchoice", 1:6, sep = "")
```