ECON-613 HW #1

Min Chul Kim

2019-01-29

Here, I am reading in the data.

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

Note

For each exercise, I am going to present 20 lines of table, if the table contains more than 20 rows. This is to prevent the assignment from being overly populated with the outputs. If more than 20 rows are needed, please run the codes provided in the R chunks.

Exercise 1

In this exercise, I report the seven statistics in three tables. The first four are in one table. The other two are presented in two tables.

```
#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()
Same_School = cbind(seq(1, nrow(Same_School), by = 1), Same_School)
colnames(Same_School) = c("Index", "Apply to the Same School?")

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

kable(head(Exercise1_df, 20), format = "markdown")
```

Num of Student	Num of School	Num of Programs	Number of Choices	Missing Test Score
340823	640	33	4974	179887

```
kable(head(Same_School, 20), format = "markdown")
```

Index	Apply to the Same School?
1	Yes
2	No
3	No
4	No
5	No
6	No
7	No
8	No
9	Yes
10	No
11	No
12	No
13	Yes
14	No
15	No
16	No
17	Yes
18	No
19	Yes
20	Yes

```
kable(head(Less_Choices, 20), format = "markdown")
```

Apply to less than 6 programs?
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes

```
#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")
#kable(Less_Choices, digits = 4, caption = "Table of Apply to less than 6 Choices")
```

Exercise 2

In this problem, I present the six variables in two tables.

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

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

kable(head(School_Level_Dataset[, 1:5], 20), format = "markdown")
```

School_Programs	schoolcode	program	cutoff	quality
100101_General Arts	100101	General Arts	198	244.1923
100101_Home Economics	100101	Home Economics	199	229.4500
100101_Technical	100101	Technical	201	235.1020
100102_Agriculture	100102	Agriculture	273	292.7816
100102_Business	100102	Business	283	303.1294
100102_General Arts	100102	General Arts	291	310.8023
100102_General Science	100102	General Science	273	298.5843
100102_Home Economics	100102	Home Economics	262	279.0000
100102_Visual Arts	100102	Visual Arts	250	273.0714
100104_General Arts	100104	General Arts	319	335.9070
100104_General Science	100104	General Science	313	333.7500
100104_Home Economics	100104	Home Economics	282	309.3556
100105_Business	100105	Business	251	267.6184
100105_General Arts	100105	General Arts	258	274.8182
100105_Home Economics	100105	Home Economics	242	257.7342

School_Programs	schoolcode	program	cutoff	quality
100106_Agriculture	100106	Agriculture	223	240.6250
100106_Business	100106	Business	238	253.5385
100106_General Arts	100106	General Arts	248	268.9750
100201_Business	100201	Business	288	314.5789
100201_General Arts	100201	General Arts	319	339.0000

```
kable(head(School_Level_Dataset[, 6:9], 20), format = "markdown")
```

size	sssdistrict	ssslat	ssslong
78	Wa Municipal	10.03062	-2.285030
40	Wa Municipal	10.03062	-2.285030
49	Wa Municipal	10.03062	-2.285030
87	Wa Municipal	10.03062	-2.285030
85	Wa Municipal	10.03062	-2.285030
86	Wa Municipal	10.03062	-2.285030
89	Wa Municipal	10.03062	-2.285030
44	Wa Municipal	10.03062	-2.285030
42	Wa Municipal	10.03062	-2.285030
43	Wa Municipal	10.03062	-2.285030
44	Wa Municipal	10.03062	-2.285030
45	Wa Municipal	10.03062	-2.285030
76	Wa Municipal	10.03062	-2.285030
77	Wa Municipal	10.03062	-2.285030
79	Wa Municipal	10.03062	-2.285030
40	Wa Municipal	10.03062	-2.285030
39	Wa Municipal	10.03062	-2.285030
40	Wa Municipal	10.03062	-2.285030
76	Lawra	10.54640	-2.800941
39	Lawra	10.54640	-2.800941

```
#kable(School_Level_Dataset[, 1:5], 4, "Table of School Level Dataset First Five Columns")
#kable(School_Level_Dataset[, 6:10], 4, "Table of School Level Dataset Last Five Columns")
```

Exercise 3

Using the supplied formula, I present a table of calculated distances.

```
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
)
```

```r
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")
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)

kable(head(dist_df[, seq(3, 13, by = 2)], 20), format = "markdown")
```

distance1	distance2	distance3	distance4	distance5	distance6
0.000000	0.000000	0.000000	0.000000	45.404991	18.08233
46.461827	0.000000	14.034819	0.000000	0.000000	14.03482
7.719788	7.719788	0.000000	0.000000	14.034819	16.57445
46.461827	14.034819	0.000000	0.000000	7.719788	14.03482
22.968725	35.613023	34.798794	22.968725	19.403603	36.38631
55.468094	19.064027	0.000000	19.064027	55.468094	39.52487
11.278269	11.278269	11.278269	11.278269	55.410029	55.41003
14.033432	0.000000	0.000000	0.000000	13.848205	13.84820
7.719788	46.461827	14.034819	7.719788	7.719788	0.00000

distance1	distance2	distance3	distance4	distance5	distance6
0.000000	13.848205	14.033432	24.016837	14.033432	33.56796
13.848205	0.000000	14.033432	24.016837	33.567956	14.03343
0.000000	0.000000	46.461827	0.000000	7.719788	39.53629
46.461827	110.841241	0.000000	0.000000	14.034819	46.46183
0.000000	0.000000	7.719788	46.461827	14.034819	46.46183
14.034819	46.461827	0.000000	0.000000	7.719788	16.57445
74.623274	14.034819	46.461827	0.000000	45.421302	46.46183
0.000000	0.000000	0.000000	0.000000	77.323477	191.40718
33.813188	33.813188	48.798298	50.398901	75.845086	115.95124
20.944477	20.944477	25.723259	25.723259	27.012692	23.84436
0.000000	33.567956	0.000000	30.217812	33.567956	0.00000

```
#kable(dist_df[, seq(3,13, by = 2)], digits = 4, "Table of Distances")
```

Exercise 4

Six tables are presented here, reporting average, sd, and test score quantiles, of each variable.

```
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()
  )
}
#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()
}

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")

kable(head(cutoff_stats, 20), format = "markdown")
```

rank	mean	sd
1	264.3742	47.32189
2	264.5785	47.44261
3	261.9684	46.01942
4	257.4536	43.50759
5	229.7480	21.75547
6	231.4267	23.18111

```
kable(head(quality_stats, 20), format = "markdown")
```

rank	mean	sd
1	288.0116	44.61923
2	283.6187	45.71187
3	281.0282	43.15453
4	275.7710	39.67396
5	246.3764	22.24052
6	248.3402	22.42373

```r
kable(head(distance_stats, 20), format = "markdown")
```

mean	sd
38.84107	51.44517
33.71148	43.99119
28.22006	38.63949
21.45510	34.02214
32.57525	26.80393
28.28762	25.91396

```r
#table(cutoff_stats, 4, "Table of Cutoff")
#table(quality_stats, 4, "Table of Quality")
#table(distance_stats, 4, "Table of Distance")
```

```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 = "")

kable(head(quantile_cutoff_df, 20), format = "markdown")
```

	rankchoice1	rankchoice2	rankchoice3	rankchoice4	rankchoice5	rankchoice6
0%	192	191	192	185	198	192
25%	227	226	225	223	216	216
50%	251	252	249	244	224	227
75%	292	292	290	284	236	239
100%	433	431	423	401	352	366

```
kable(head(quantile_quality_df, 20), format = "markdown")
```

	rankchoice1	rankchoice2	rankchoice3	rankchoice4	rankchoice5	rankchoice6
0%	200.0000	201.0000	197.0000	191.0000	198.0000	201.0000
25%	254.6550	248.4470	247.3333	245.1053	231.7500	234.0000
50%	276.0500	273.0000	270.3571	265.9048	243.0000	246.4000
75%	314.1154	311.6986	310.2000	302.8571	257.6264	259.0278

	rankchoice1	rankchoice2	rankchoice3	rankchoice4	rankchoice5	rankchoice6
100%	443.9906	431.0000	423.0000	401.0000	354.6667	366.0000

```
kable(head(quantile_distance_df, 20), format = "markdown")
```

	rankchoice1	rankchoice2	rankchoice3	rankchoice4	rankchoice5	rankchoice6
0%	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
25%	0.00000	0.00000	0.00000	0.00000	12.55811	0.00000
50%	22.41464	21.98925	17.70825	11.63369	29.82565	24.92747
75%	55.44538	48.30857	39.49381	28.48841	48.62287	43.54712
100%	418.47444	450.35435	433.23023	412.51378	352.75056	373.97420

```
#table(quantile_cutoff_df, 4, "Table of Cutoff Differentiated by Test Score Quantiles")
#table(quantile_quality_df, 4, "Table of Quality Differentiated by Test Score Quantiles")
#table(quantile_distance_df, 4,
#      "Table of Distance Differentiated by Test Score Quantiles")
```

Exericise 5

Two tables are presented here. One is with decile; the other is with quantile.

```
School_Level_Dataset2 = School_Level_Dataset %>%
  mutate(decile = ntile(cutoff, 10))

application_groups = function(var, var2){

  return(
    datstu2 %>%
      left_join(School_Level_Dataset2, by = var) %>%
      select(!!enquo(var2), decile)
  )
}

application_df = data.frame(
  application_groups(c("choice1" = "School_Programs"), choice1),
  application_groups(c("choice2" = "School_Programs"), choice2),
  application_groups(c("choice3" = "School_Programs"), choice3),
  application_groups(c("choice4" = "School_Programs"), choice4),
  application_groups(c("choice5" = "School_Programs"), choice5),
  application_groups(c("choice6" = "School_Programs"), choice6)
)
colnames(application_df) = c("choice1", "decile1", "choice2", "decile2",
                             "choice3", "decile3", "choice4", "decile4",
                             "choice5", "decile5", "choice6", "decile6"
                             )

application_df = application_df %>%
  select(choice1, choice2, choice3, choice4, choice5, choice6,
         decile1, decile2, decile3, decile4, decile5, decile6)
```

```r
application_count_df =
  apply(application_df[, 7:12], 1, FUN = unique) %>%
  map(na.omit) %>%
  map(length) %>%
  unlist() %>%
  as.data.frame()
colnames(application_count_df) = "Number of Groups"

application_df = cbind(application_df, application_count_df)

kable(head(application_count_df, 20), format = "markdown")
```

Number of Groups
4
4
6
4
3
3
5
4
4
4
4
4
5
4
5
4
5
4
5
3

```r
#table(application_count_df, 4, "Table of the Number of Groups in the Application")
```

```r
School_Level_Dataset3 = School_Level_Dataset2

student_quantile_score = quantile(datstu2$score, seq(0, 1, by = 0.25)) %>%
  as.integer()

application_quantile = function(var, var2){
  quantile = c()

  for(i in 1:2293){
  if(var2[1] < var[i] & var[i] < var2[2]){
    quantile[i] = 1
  } else if(var2[2] < var[i] & var[i] < var2[3]){
    quantile[i] = 2
  } else if(var2[3] < var[i] & var[i] < var2[4]){
    quantile[i] = 3
  } else{
```

```
    quantile[i] = 4
  }
  }
  return(quantile)
}

quantiles = application_quantile(School_Level_Dataset3$cutoff, student_quantile_score)
quantiles = c(quantiles, rep(NA, dim(School_Level_Dataset3)[1] - length(quantiles)))

School_Level_Dataset3 = cbind(School_Level_Dataset3, quantiles)

application_groups = function(var, var2){
  return(
    datstu2 %>%
      left_join(School_Level_Dataset3, by = var) %>%
      select(!!enquo(var2), quantiles)
  )
}

application_df2 = data.frame(
  application_groups(c("choice1" = "School_Programs"), choice1),
  application_groups(c("choice2" = "School_Programs"), choice2),
  application_groups(c("choice3" = "School_Programs"), choice3),
  application_groups(c("choice4" = "School_Programs"), choice4),
  application_groups(c("choice5" = "School_Programs"), choice5),
  application_groups(c("choice6" = "School_Programs"), choice6)
)
colnames(application_df2) = c("choice1", "quant1", "choice2", "quant2",
                              "choice3", "quant3", "choice4", "quant4",
                              "choice5", "quant5", "choice6", "quant6")

application_df2 = application_df2 %>%
  select(choice1, choice2, choice3, choice4, choice5, choice6,
         quant1, quant2, quant3, quant4, quant5, quant6)

application_count_df2 =
  apply(application_df2[, 7:12], 1, FUN = unique) %>%
  map(na.omit) %>%
  map(length) %>%
  unlist() %>%
  as.data.frame()
colnames(application_count_df2) = "Number of Groups"

application_df2 = cbind(application_df2, application_count_df2)

kable(head(application_count_df2, 20), format = "markdown")
```

Number of Groups
3
2
3
2
2

Number of Groups
3
4
3
3
2
2
2
4
2
3
3
2
3
3
3

```
#table(application_count_df2, 4, "Table of the Number of Groups in the Application")
```