Midterm

Chang An Le Harry Jr

3/2/2022

```
library(jsonlite)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.7     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()  masks stats::filter()
## x purrr::flatten() masks jsonlite::flatten()
## x dplyr::lag()     masks stats::lag()
```

Question 1

```
college_data = read.csv("college.csv")
head(college_data) #reading csv file
```

```
##   rank                                   major major_category major_stem
## 1    1                   Petroleum Engineering    Engineering       STEM
## 2    2             Mining And Mineral Engineering    Engineering       STEM
## 3    3                Metallurgical Engineering    Engineering       STEM
## 4    4 Naval Architecture And Marine Engineering    Engineering       STEM
## 5    5                    Chemical Engineering    Engineering       STEM
## 6    6                     Nuclear Engineering    Engineering       STEM
##   total sharewomen unemployment_rate median
## 1  2339  0.1205643        0.01838053 110000
## 2   756  0.1018519        0.11724138  75000
## 3   856  0.1530374        0.02409639  73000
## 4  1258  0.1073132        0.05012531  70000
## 5 32260  0.3416305        0.06109771  65000
## 6  2573  0.1449670        0.17722641  65000
```

1. Which major has the highest unemployment rate? In your output, display a table that lists the top five majors with the highest unemployment rate.

```
college_q1 = college_data %>%
  arrange(desc(unemployment_rate)) #arranging data based on unemployment rate, in desc order

college_q1a = college_q1 %>%
  select(major, unemployment_rate) #selecting only the relevant columns

head(college_q1a, n = 5) #show top 5 majors based on highest unemployment rate only
```

```
##                                         major unemployment_rate
## 1                         Nuclear Engineering         0.1772264
## 2                       Public Administration         0.1594906
## 3 Computer Networking And Telecommunications         0.1518498
## 4                         Clinical Psychology         0.1490482
## 5                               Public Policy         0.1284263
```

From the output table above, the top 5 majors with the highest unemployment rates are:
1. Nuclear Engineering
2. Public Administration
3. Computer Networking And Telecommunications
4. Clinical Psychology
5. Public Policy

2. Which major has the lowest percentage of women? In your output, display a table of three columns: rank, major, sharewomen. Round sharewomen to 2 decimal places.

```
college_q2 = college_data %>%
  arrange(sharewomen) #arranging data based on sharewomen, in asc order

college_q2$sharewomen = round(college_q2$sharewomen, digit = 2) #round off sharewomen to 2dp

college_q2a = college_q2 %>%
  select(rank, major, sharewomen) #selecting only the relevant columns

head(college_q2a, n = 5) #show top 5 majors based on lowest sharewomen only
```

```
##   rank                                       major sharewomen
## 1   74                       Military Technologies       0.00
## 2   67 Mechanical Engineering Related Technologies       0.08
## 3   27                       Construction Services       0.09
## 4    2              Mining And Mineral Engineering       0.10
## 5    4   Naval Architecture And Marine Engineering       0.11
```

From the output table above, the major Military Technologies has the lowest percentage of women.

3. Which major has the highest percentage of women? Summarize in words of the patterns you found in part 2 and part 3.

```
college_q3 = college_data %>%
  arrange(desc(sharewomen)) #arranging data based on sharewomen, in desc order

college_q3a = college_q3 %>%
  select(major, sharewomen) #selecting only the relevant columns

head(college_q3a, n = 5) #show top 5 majors based on highest sharewomen only
```

```
##                                          major sharewomen
## 1                     Early Childhood Education  0.9689537
## 2 Communication Disorders Sciences And Services  0.9679981
## 3                    Medical Assisting Services  0.9278072
## 4                          Elementary Education  0.9237455
## 5                 Family And Consumer Sciences  0.9109326
```

From the output table above, the major Early Childhood Education has the highest percentage of women.

Based on the patterns derived from parts 2 and 3, we can deduce that the majors with highest percentages of women are mostly non-STEM majors. On the other hand, the majors with the lowest percentages of women comprise of a mixture of STEM and non-STEM majors (although still mostly dominated by the STEM majors).

I believe that the trends from this dataset tallies with the overall trend that men are still mostly dominating the STEM majors in college such as computing and engineering while women are mostly dominating the non-STEM majors, such as the social sciences, humanities, health and education.

4. Explore the dataset, and answer one question you find interesting about the data. Include the code you used, and summarize (in words) what you found.

For this question, I'd like to find out - what is the relationship between the three variables: shareofwomen, unemployment_rate and median amongst STEM and non-STEM majors?

Before we proceed with the exploration, we would need to omit rows with NA values in any of the columns from the dataset.

```
college_q4 = na.omit(college_data) #only 1 observation removed (Food Science)
```

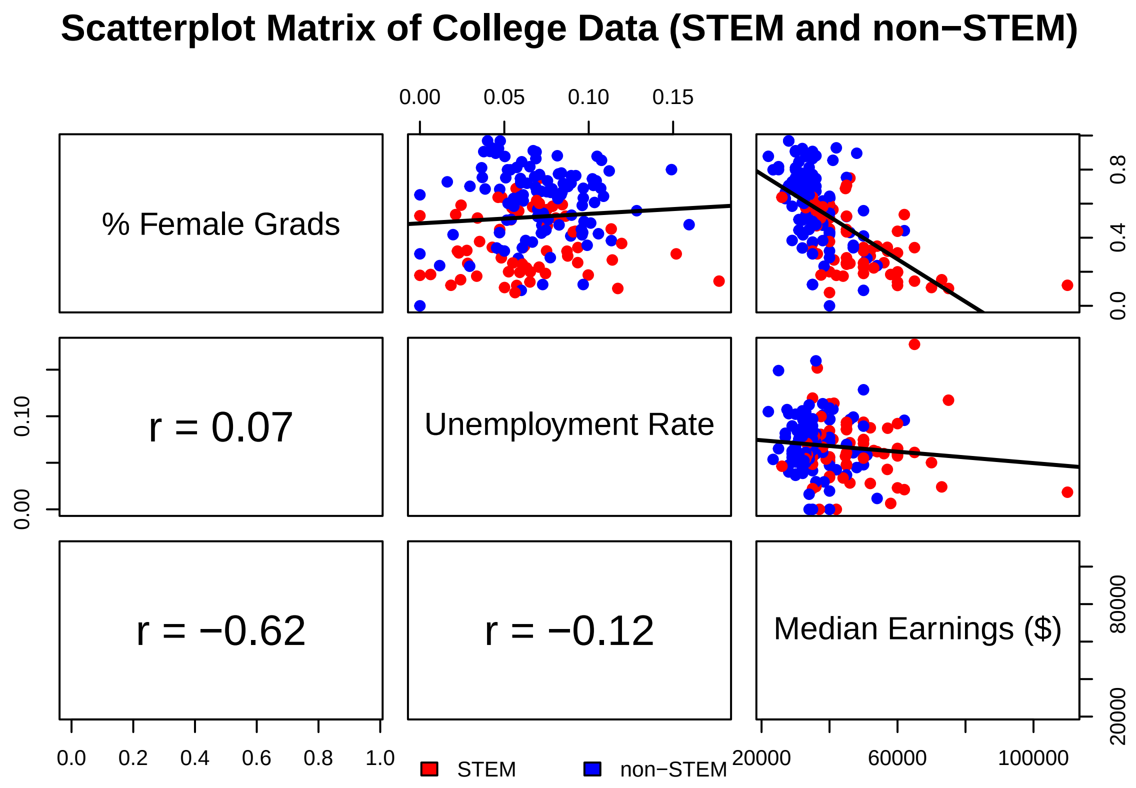
Next, we can plot a scatterplot matrix to investigate the relationship between the 3 variables:

```
par(mar=c(1,1,1,1))
options(scipen = 999) #disable scientific notation

#correlation panel to calculate respective correlation coefficients between any 2 variables
panel.cor = function(x, y){
    usr = par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r = round(cor(x, y), digits=2)
    txt = paste0("r = ", r)
    text(0.5, 0.5, txt, cex = 2)
```

```r
}

my.text.panel = function(labels) {
  function(x, y, lbl, ...) {
    if (lbl %in% names(labels)) lbl <- labels[[lbl]]
    text(x, y, lbl, ...)
  }
}

panel.lm = function (x, y, col = par("col"), bg = NA, pch = par("pch"),
                      cex = 1, col.smooth = "black", ...) {
  points(x, y, pch = 19, col = ifelse(college_q4$major_stem == "STEM", "red", "blue"), bg = bg, cex = c
  abline(stats::lm(y ~ x),  col = col.smooth, lwd = 2)
}

pairs(college_q4[,6:8],
      main = "Scatterplot Matrix of College Data (STEM and non-STEM)",
      text.panel = my.text.panel(c(sharewomen = "% Female Grads",
                                   unemployment_rate = "Unemployment Rate",
                                   median = "Median Earnings ($)")),
      lower.panel = panel.cor,
      upper.panel = panel.lm)
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```

```r
 legend("bottom",inset = -0.02, horiz = TRUE, legend=c("STEM", "non-STEM"), fill = c("red","blue"), cex
```



For the scatterplot matrix above, I have included regression lines and correlation coefficients to visually interpret the various relationships between any 2 variables. I have also renamed the variable names (i.e sharewomen -> % Female Grads, unemployment_rate -> Unemployment Rate, median -> Median Earnings) in the plot above for easier reference.

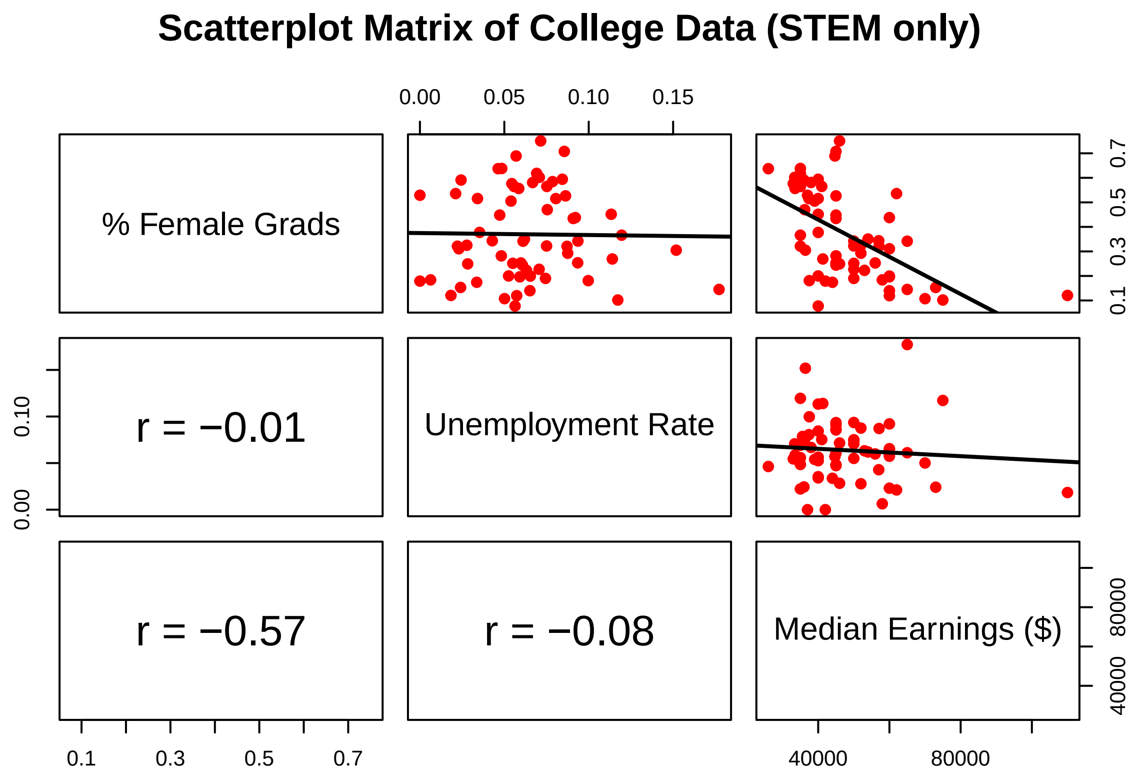
Given the very low values for the correlation coefficients of sharewomen vs unemployment_rate (r = 0.07) and unemployment_rate vs median (r = - 0.12), we can only determine that there is a moderately negative correlation between the sharewomen and median variables (r = -0.62). This suggests that with less women graduating from a particular major, the median annual earnings of a graduate from that same major generally increases. In other words, this means that graduates from a male-dominant major such as Computing and Engineering will generally have higher paying jobs based on the value of median annual earnings.

But what if we were to further divide the observations based on major_stem (STEM vs non-STEM)? As seen in the above plot, the STEM observations were colour-coded as red dots while the non-STEM observations were colour-coded as blue dots. With possibly varying correlation coefficients based on major_stem, we can create additional scatterplot matrices using smaller datasets categorised based on STEM and non-STEM.

```
college_q4_stem = subset(college_q4, major_stem == "STEM",
                         select = c("sharewomen", "unemployment_rate", "median")) #plot for STEM only

options(scipen = 999) #disable scientific notation

#correlation panel to calculate respective correlation coefficients between any 2 variables
panel.cor = function(x, y){
    usr = par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r = round(cor(x, y), digits=2)
```

```r
    txt = paste0("r = ", r)
    text(0.5, 0.5, txt, cex = 2)
}

my.text.panel = function(labels) {
  function(x, y, lbl, ...) {
    if (lbl %in% names(labels)) lbl <- labels[[lbl]]
    text(x, y, lbl, ...)
  }
}

panel.lm = function (x, y, col = par("col"), bg = NA, pch = par("pch"),
                      cex = 1, col.smooth = "black", ...) {
  points(x, y, pch = 19, col = "red", bg = bg, cex = cex)
  abline(stats::lm(y ~ x),  col = col.smooth, lwd = 2)
}

pairs(college_q4_stem,
      main = "Scatterplot Matrix of College Data (STEM only)",
      text.panel = my.text.panel(c(sharewomen = "% Female Grads",
                                   unemployment_rate = "Unemployment Rate",
                                   median = "Median Earnings ($)")),
      lower.panel = panel.cor,
      upper.panel = panel.lm)
```

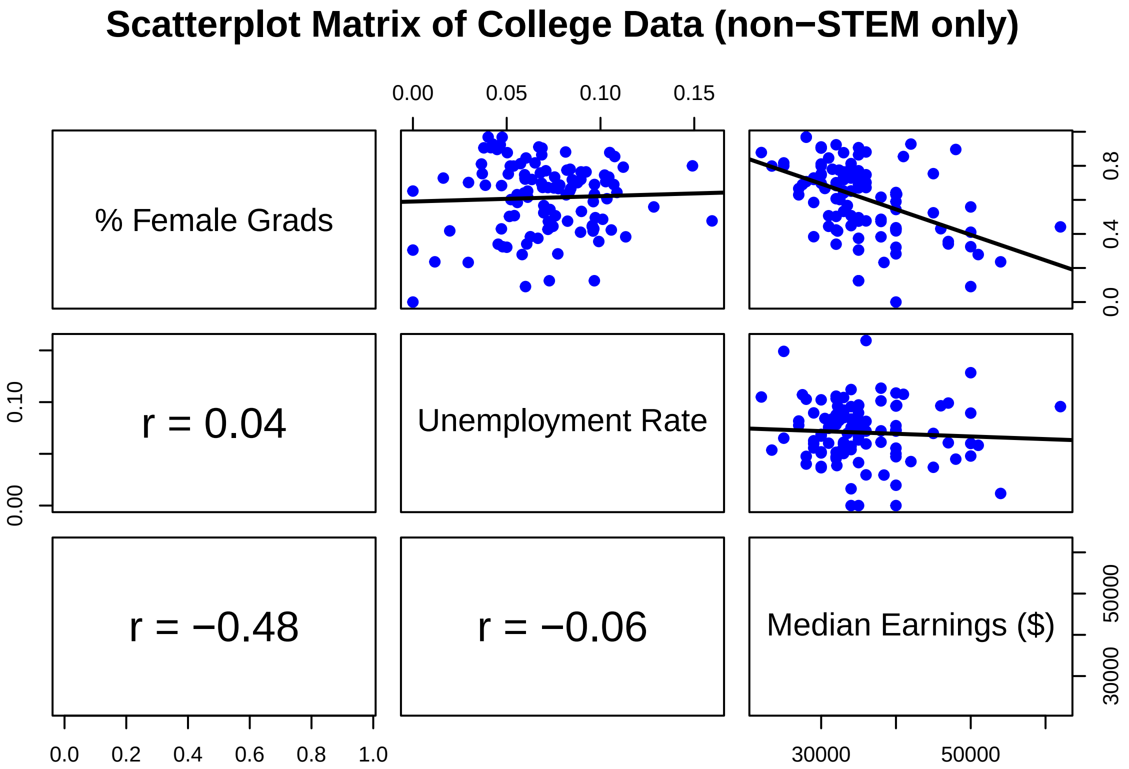
```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```



By comparing the STEM-only matrix and the combined scatterplot matrix, there is generally not much difference in the derived insights from the various relationships established. Looking at the STEM-only matrix, here is still a moderate negative correlation between sharewomen and media (r = -0.57) while there is a similarly weak/negligible correlation between unemployment_rate and median (r = -0.08). Interestingly, the correlation coefficient when comparing the sharewomen and unemployment_rate is now negative (r = -0.01) as compared to the combined matrix, which had a value of 0.07 when investigating the same 2 variables.

```
college_q4_nonstem = subset(college_q4, major_stem == "non-STEM",
                         select = c("sharewomen", "unemployment_rate", "median")) #plot for non-STEM on

options(scipen = 999) #disable scientific notation

#correlation panel to calculate respective correlation coefficients between any 2 variables
panel.cor = function(x, y){
    usr = par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r = round(cor(x, y), digits=2)
    txt = paste0("r = ", r)
    text(0.5, 0.5, txt, cex = 2)
}

my.text.panel = function(labels) {
  function(x, y, lbl, ...) {
    if (lbl %in% names(labels)) lbl <- labels[[lbl]]
    text(x, y, lbl, ...)
  }
}

panel.lm = function (x, y, col = par("col"), bg = NA, pch = par("pch"),
                      cex = 1, col.smooth = "black", ...) {
  points(x, y, pch = 19, col = "blue", bg = bg, cex = cex)
  abline(stats::lm(y ~ x),  col = col.smooth, lwd = 2)
}

pairs(college_q4_nonstem,
      main = "Scatterplot Matrix of College Data (non-STEM only)",
      text.panel = my.text.panel(c(sharewomen = "% Female Grads",
                                   unemployment_rate = "Unemployment Rate",
                                   median = "Median Earnings ($)")),
      lower.panel = panel.cor,
      upper.panel = panel.lm)
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```



Moving onto the non-STEM-only matrix, the insights are also similar to those of the combined scatterplot matrix, although the magnitude of the respective coefficient correlation values have been reduced, signifying weaker relationships between the respective pairs of variables when considering only non-STEM observations.

In conclusion, the only meaningful relationship that could be established from my analysis would be the moderately negative correlation between the sharewomen and median variables. And with further analysis based on the major_stem variable, it seems like there is a stronger negative relationship between sharewomen and median amongst STEM majors (r = -0.57) as compared to that same relationship amongst non-STEM majors (r = -0.48). This suggests that within the STEM majors, a major which has a higher composition of men within its cohort would be more likely to experience higher annual earnings compared to non-STEM majors.

Question 2

Use the Data API link on the web page to find the resource id of this data. Download the full dataset, and name it as rents.

```r
root_url = "https://data.gov.sg"
url1 = paste(root_url,
             "/api/action/datastore_search?",
             "resource_id=4384e22c-234f-4196-9df8-1941cd41c667",
             sep = "")
rents_json = fromJSON(url1)
str(rents_json)
```

```
## List of 3
##  $ help   : chr "https://data.gov.sg/api/3/action/help_show?name=datastore_search"
##  $ success: logi TRUE
##  $ result :List of 5
##   ..$ resource_id: chr "4384e22c-234f-4196-9df8-1941cd41c667"
##   ..$ fields     :'data.frame':  8 obs. of  2 variables:
##   .. ..$ type: chr [1:8] "int4" "text" "text" "numeric" ...
##   .. ..$ id  : chr [1:8] "_id" "qtr" "project_name" "postal_district" ...
##   ..$ records    :'data.frame':  100 obs. of  8 variables:
##   .. ..$ project_name    : chr [1:100] "10 SHELFORD" "18 WOODSVILLE" "26 NEWTON" "368 THOMSON" ...
##   .. ..$ qtr             : chr [1:100] "2022-Q4" "2022-Q4" "2022-Q4" "2022-Q4" ...
##   .. ..$ 75th_percentile : chr [1:100] "7.43" "5.25" "7.47" "5.44" ...
##   .. ..$ median          : chr [1:100] "6.48" "4.65" "7.02" "4.59" ...
##   .. ..$ 25th_percentile : chr [1:100] "5.75" "3.73" "6.01" "4.15" ...
##   .. ..$ postal_district : chr [1:100] "11" "13" "11" "11" ...
##   .. ..$ _id             : int [1:100] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ rental_contracts: chr [1:100] "13" "11" "28" "19" ...
##   ..$ _links     :List of 2
##   .. ..$ start: chr "/api/action/datastore_search?resource_id=4384e22c-234f-4196-9df8-1941cd41c667"
##   .. ..$ next : chr "/api/action/datastore_search?offset=100&resource_id=4384e22c-234f-4196-9df8-194
##   ..$ total      : int 581
```

```r
rents = rents_json$result$records
total_records = rents_json$result$total

times = floor(total_records/100)
for (i in 1:times) {
  url = paste(root_url,
              "/api/action/datastore_search?",
              "offset=", i,
               "00&",
               "resource_id=4384e22c-234f-4196-9df8-1941cd41c667",
               sep = "")
  rents_json = fromJSON(url)
  rents = rbind(rents, rents_json$result$records)
}
dim(rents) #check if dimensions are correct
```

```
## [1] 581   8
```

Since the number of records (651) tallies with what is presented on the webpage, all we need to do is to remove the variable _id and check the dataset again:

```
rents = rents[,-(7)] #drop _id column
head(rents) #check dataset
```

```
##    project_name     qtr 75th_percentile median 25th_percentile postal_district
## 1   10 SHELFORD 2022-Q4            7.43   6.48            5.75              11
## 2 18 WOODSVILLE 2022-Q4            5.25   4.65            3.73              13
## 3     26 NEWTON 2022-Q4            7.47   7.02            6.01              11
## 4   368 THOMSON 2022-Q4            5.44   4.59            4.15              11
## 5   38 I SUITES 2022-Q4            5.53   4.95            4.65              15
## 6  6 DERBYSHIRE 2022-Q4            6.82   5.91            5.46              11
##   rental_contracts
## 1               13
## 2               11
## 3               28
## 4               19
## 5               15
## 6               23
```

1. Read the data description on the website and briefly describe the dataset and the variables it includes.

Based on the data description on the website, this dataset contains summary statistics of major non-landed private residential projects with at least 10 rental contracts signed in a quarter. A major project is defined as one with at least 100 residential units. The coverage ov the dataset ranges from Oct 1, 2021 to Dec 31, 2021, also referring to the 4th quarter of 2021.

The dataset contains 7 different variables, which include:
a. qtr - quarter in which data was collected (listed as YYYY-QX)
b. project_name - name of project
c. postal_district - postal district of which each project is located at
d. 25th_percentile - value of 25th percentile of rental contracts signed for each project (S$ per square meter, per month)
e. median - value of median of rental contracts signed for each project (S$ per square meter, per month)
f. 75th_percentile - value of 75th percentile of rental contracts signed for each project (S$ per square meter, per month)
7. rental_contracts - number of rental contracts signed for each project

2. How many observations and variables are there? Display the first ten rows of your dataset.

```r
dim(rents) #check dimensions again
```

```
## [1] 581   7
```

As shown above, there are 651 observations and 7 variables, which tallies with the dataset presented on the given webpage.

```r
head(rents, n = 10) #check first 10 rows of dataset
```

```
##        project_name     qtr 75th_percentile median 25th_percentile
## 1       10 SHELFORD 2022-Q4            7.43   6.48            5.75
## 2     18 WOODSVILLE 2022-Q4            5.25   4.65            3.73
## 3         26 NEWTON 2022-Q4            7.47   7.02            6.01
## 4       368 THOMSON 2022-Q4            5.44   4.59            4.15
## 5       38 I SUITES 2022-Q4            5.53   4.95            4.65
## 6      6 DERBYSHIRE 2022-Q4            6.82   5.91            5.46
## 7        76 SHENTON 2022-Q4            7.52   7.05            6.58
## 8   77 @ EAST COAST 2022-Q4            6.61   6.22            6.14
## 9  8 @ MOUNT SOPHIA 2022-Q4            5.41   4.93            4.37
## 10        8 BASSEIN 2022-Q4             7.2    6.5            5.86
##    postal_district rental_contracts
## 1               11               13
## 2               13               11
## 3               11               28
## 4               11               19
## 5               15               15
## 6               11               23
## 7                2               31
## 8               15               11
## 9                9               25
## 10              11               13
```

3. Display the top ten non-landed private residential project that had the most rental contracts signed in 2021-Q4. Summarize in words of the patterns you found.

```r
rents_q3 = rents %>%
  arrange(desc(rental_contracts))

head(rents_q3, n = 10)
```

```
##                 project_name     qtr 75th_percentile median 25th_percentile
## 1     CITY SQUARE RESIDENCES 2022-Q4            5.78   5.11            4.53
## 2              BAYSHORE PARK 2022-Q4            3.78   3.39               3
## 3                QUEENS PEAK 2022-Q4             7.7   7.06            6.34
## 4        COMMONWEALTH TOWERS 2022-Q4            8.21   7.28            6.64
## 5       HIGH PARK RESIDENCES 2022-Q4            5.46   4.95            4.28
## 6                       ICON 2022-Q4            7.23   6.43            6.02
## 7                THE BAYSHORE 2022-Q4            4.22   3.76             3.29
## 8  REFLECTIONS AT KEPPEL BAY 2022-Q4            6.14   5.53             5.15
## 9                   EUHABITAT 2022-Q4            5.67   4.94             4.18
## 10                  J GATEWAY 2022-Q4            7.41   6.74             6.06
##    postal_district rental_contracts
## 1                8               89
## 2               16               86
## 3                3               82
## 4                3               81
## 5               28               81
## 6                2               81
## 7               16               77
## 8                4               76
## 9               14               74
## 10              22               72
```

The output table above presents the top 10 non-landed private residential projects with the most rental contracts signed in 2021-Q4, with REFLECTIONS AT KEPPEL BAY having the highest number of rental contracts signed (98).

Among the top 10 projects listed above, 7 out of the 10 projects are located in the East, based on their postal district identification (postal districts 14, 16, 18, 19). This may be an indication that the East may be the most popular region by demand amongst potential non-landed private housing residents in Singapore.

Source for postal district identification: https://www.ura.gov.sg/realEstateIIWeb/resources/misc/list__of__postal__districts.htm

4. Explore the dataset, and answer one question you find interesting about the data. Include the code you used, and summarize (in words) what you found.

For this particular dataset, I would like to find out - how many rental contacts were signed in total based on the 6 different regions (i.e North, South, East, West, Central, City) in Singapore? And based on this question, which are currently the most highly demanded regions for non-landed private residential housing in Singapore?

But first, we need to categorise the 28 postal districts into the different regions. However, there are many online interpretations of Singapore's postal district map, with certain districts still being disputed as to which region they really belong to. For instance, it is debatable that postal district 14 (Eunos, Geylang) can be considered as either part of the Central region or the East region, depending on different perspectives.

Thus, for the purpose of this analysis, I will be making reference to the link below when categorising the 28 districts into the 6 different regions: https://sharonanngoh.com/useful-info/singapore-district-guide/

We first need to create a new column to indicate the region in which each postal district belongs to:

```
rents_q4 = rents %>% #creating region column
  mutate(postal_district = as.numeric(postal_district)) %>%
  mutate(region = case_when(
    postal_district >= 13 & postal_district <= 18 ~ "East",
    postal_district >= 21 & postal_district <= 24 | postal_district == 5 ~ "West",
    postal_district >= 25 & postal_district <= 28 | postal_district == 19 | postal_district == 20 ~ "No
    postal_district == 3 | postal_district == 4 ~ "South",
    postal_district >= 8 & postal_district <= 12 ~ "Central",
    postal_district == 1 | postal_district == 2 | postal_district == 6 | postal_district == 7 ~ "City"))

head(rents_q4, n = 10) #to check if new column has successfully been created
```

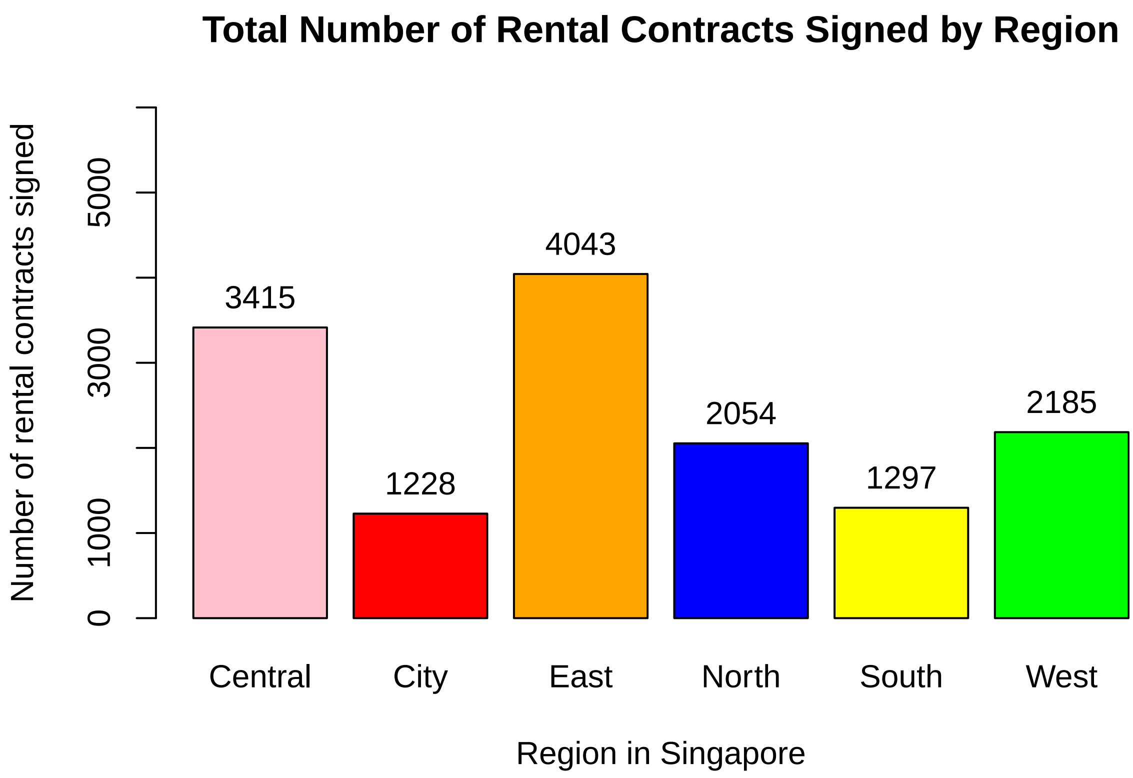
```
##        project_name     qtr 75th_percentile median 25th_percentile
## 1       10 SHELFORD 2022-Q4            7.43   6.48            5.75
## 2     18 WOODSVILLE 2022-Q4            5.25   4.65            3.73
## 3         26 NEWTON 2022-Q4            7.47   7.02            6.01
## 4       368 THOMSON 2022-Q4            5.44   4.59            4.15
## 5       38 I SUITES 2022-Q4            5.53   4.95            4.65
## 6      6 DERBYSHIRE 2022-Q4            6.82   5.91            5.46
## 7        76 SHENTON 2022-Q4            7.52   7.05            6.58
## 8   77 @ EAST COAST 2022-Q4            6.61   6.22            6.14
## 9  8 @ MOUNT SOPHIA 2022-Q4            5.41   4.93            4.37
## 10        8 BASSEIN 2022-Q4             7.2    6.5            5.86
##    postal_district rental_contracts  region
## 1               11               13 Central
## 2               13               11    East
## 3               11               28 Central
## 4               11               19 Central
## 5               15               15    East
## 6               11               23 Central
## 7                2               31    City
## 8               15               11    East
## 9                9               25 Central
## 10              11               13 Central
```

Next, we can create an aggregated dataset to sum up the total number of rental contracts signed based on the different regions:

```
rents_q4a = rents_q4 %>%
  mutate(rental_contracts = as.numeric(rental_contracts)) %>%
  group_by(region) %>%
  summarise(total_rental_contracts = sum(rental_contracts)) %>%
  arrange(region)

rents_q4a #check new dataset
```

```
## # A tibble: 6 x 2
##   region  total_rental_contracts
##   <chr>                    <dbl>
## 1 Central                   3415
## 2 City                      1228
## 3 East                      4043
## 4 North                     2054
## 5 South                     1297
## 6 West                      2185
```

```
a = barplot(total_rental_contracts ~ region,
        data = rents_q4a,
        xlab = "Region in Singapore", ylab = "Number of rental contracts signed",
        main = "Total Number of Rental Contracts Signed by Region",
        ylim = c(0, 6000), cex.axis = 1,
        col = c("pink", "red", "orange", "blue", "yellow", "green"))

text(x = a, y = rents_q4a$total_rental_contracts, label = rents_q4a$total_rental_contracts, pos = 3, ce
```



Based on the barplot above, we can deduce that the top 3 most popular regions amongst residents in Singapore to rent a residential unit in a non-landed residential project are Central, East and West.

The Central region consists of locations such as Orchard, River Valley, Novena and Toa Payoh, all which are surrounded by various shopping facilities and top schools. Generally, this region would be convenient to live in particularly for foreign expatriates, who may intend to work in Singapore for a long period of time, while being able to send their children to study in Singapore's top schools as well. Also, another point to note is that the Central region is geographically one of the smaller regions among the six main regions, which further boasts its status as being the most popular region to rent a non-landed residential housing unit in based on demand.

The East region consists of locations such as Bedok, Pasir Ris, Tampines and Paya Lebar, which consists of a wide variety of shopping facilities and office buildings that caters to both foreign expatriates and local citizens. These individuals would similarly find it convenient to travel back and forth between their rented homes and their workplaces as well.

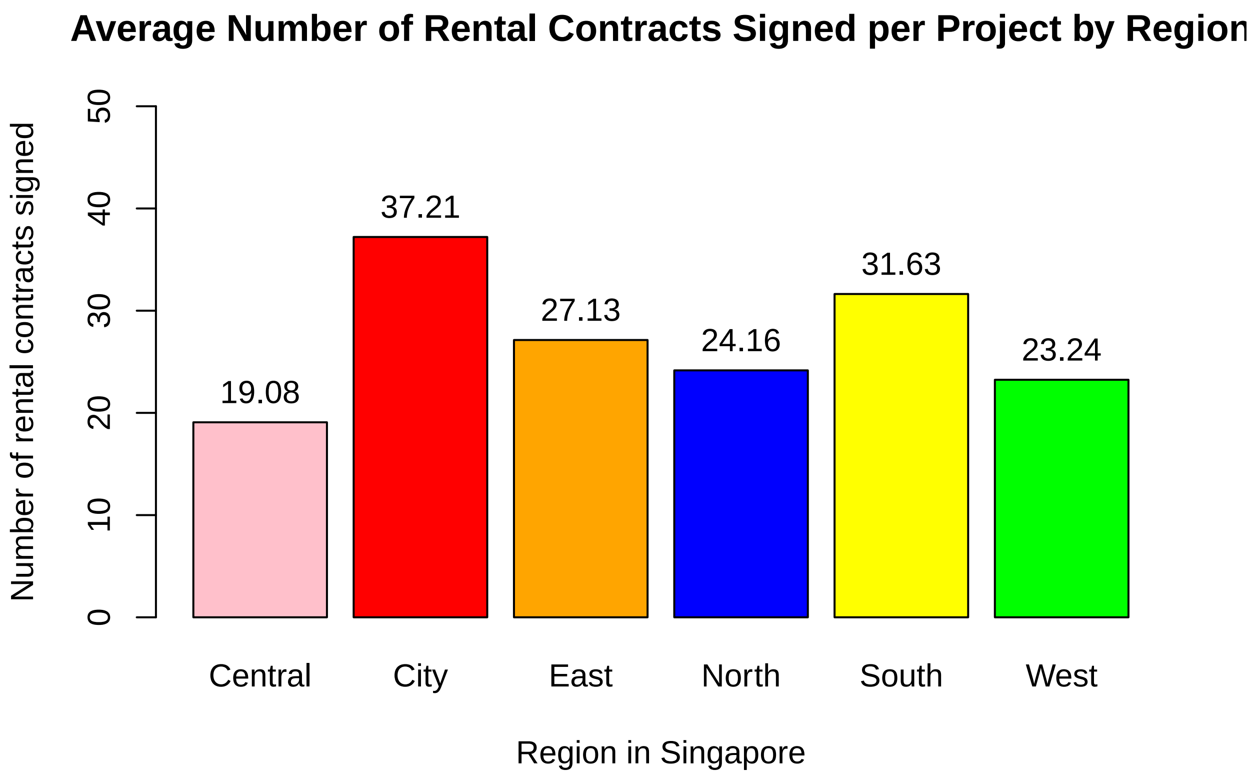
The West region comprises locations such as Jurong and Tuas, which mainly host industrial facilities and office buildings. Renting a non-landed residential unit in the West would mostly be beneficial for those working in industries such as pharmaceuticals and energy, where their offices and power plants (if applicable) would have to be located in more remote areas in the West.

To further enhance this analysis, we can also compare the average number of rental contracts signed for each property based on the 6 different regions:

```
rents_q4b = rents_q4 %>%
  mutate(rental_contracts = as.numeric(rental_contracts)) %>%
  group_by(region) %>%
  summarise(mean_rental_contracts = round(mean(rental_contracts),2)) %>%
  arrange(region)

rents_q4b #check new dataset
```

```
## # A tibble: 6 x 2
##   region  mean_rental_contracts
##   <chr>                   <dbl>
## 1 Central                  19.1
## 2 City                     37.2
## 3 East                     27.1
## 4 North                    24.2
## 5 South                    31.6
## 6 West                     23.2
```

```
b = barplot(mean_rental_contracts ~ region,
        data = rents_q4b,
        xlab = "Region in Singapore", ylab = "Number of rental contracts signed",
        main = "Average Number of Rental Contracts Signed per Project by Region",
        ylim = c(0,50), cex.axis = 1,
        col = c("pink", "red", "orange", "blue", "yellow", "green"))
text(x = b, y = rents_q4b$mean_rental_contracts, label = rents_q4b$mean_rental_contracts, pos = 3, cex=
```



Based on the barplot above, we can deduce that City region has the highest number of rental contracts signed per residential project on average. This may be an indication that each non-landed private residential project in the City region generally contains more residential units compared to the residential projects in other regions. With the City region having the 2nd lowest number of total contracts signed, this may suggest that renting a non-landed private residential unit in the City region is very expensive indeed, and will only cater to the richest individuals living in Singapore. This may be due to the relatively higher value of each unit, as well as the convenient access to various landmarks and facilities that each project's location offers.

A similar analysis can be offered for residential projects in the South region as well, since it has the second highest number of rental contracts signed per project on average, as well as having the lowest total number of rental contracts signed across all of its properties compared to other regions. This is no surprise as the South region is located next to City region according to the district map, with the former boasting popular locations such as Tiong Bahru, Alexandra, Keppel and Sentosa - all of which contain properties of higher relative value.

Conversely, in the East and West regions, the average numbers of rental contracts signed per project are lower. Given that the East and West regions were amongst the top 3 in terms of the total number of rental contracts signed, this may suggest that each non-landed private residential unit is relatively cheaper to rent due to the smaller size of each residential project (less units per project on average) in the 2 regions. The locations of each project may not be as ideal as well (e.g project not located as close to MRT station or shopping facilities for convenience).

Question 3

```
gifts_df = read_rds("gifts_retail.rds") #read dataset
```

```
head(gifts_df, n = 10) #check dataset
```

```
## # A tibble: 10 x 8
##    InvoiceNo StockCode Description        Quantity InvoiceDate         UnitPrice
##    <chr>     <chr>     <chr>                 <dbl> <dttm>                  <dbl>
##  1 536365    85123A    WHITE HANGING HEA~        6 2010-12-01 08:26:00      2.55
##  2 536365    71053     WHITE METAL LANTE~        6 2010-12-01 08:26:00      3.39
##  3 536365    84406B    CREAM CUPID HEART~        8 2010-12-01 08:26:00      2.75
##  4 536365    84029G    KNITTED UNION FLA~        6 2010-12-01 08:26:00      3.39
##  5 536365    84029E    RED WOOLLY HOTTIE~        6 2010-12-01 08:26:00      3.39
##  6 536365    22752     SET 7 BABUSHKA NE~        2 2010-12-01 08:26:00      7.65
##  7 536366    22633     HAND WARMER UNION~        6 2010-12-01 08:28:00      1.85
##  8 536367    84879     ASSORTED COLOUR B~       32 2010-12-01 08:34:00      1.69
##  9 536367    22745     POPPY'S PLAYHOUSE~        6 2010-12-01 08:34:00      2.1
## 10 536367    22748     POPPY'S PLAYHOUSE~        6 2010-12-01 08:34:00      2.1
## # ... with 2 more variables: CustomerID <dbl>, Country <chr>
```

1. Extract all rows from the one customer who spent the most in the dataset. What is the customer id for this person? How much did he/she spend in total?

```
q1_df = gifts_df %>%
  mutate(subtotal_spent = Quantity * UnitPrice) %>% #need to find out total spent on each particular it
  group_by(CustomerID) %>%
  summarise(total_spent = sum(subtotal_spent)) %>% #aggregate (sum) total spent by each customer
  arrange(desc(total_spent)) #arranging in desc order to identify top spenders

head(q1_df)
```

```
## # A tibble: 6 x 2
##   CustomerID total_spent
##        <dbl>       <dbl>
## 1      14646     278742.
## 2      18102     259657.
## 3      17450     189736.
## 4      14911     128927.
## 5      12415     123638.
## 6      14156     113859.
```

Based on the above, the customer with the CustomerID 14646 spent the most according to the dataset, with a total expenditure of £278742.00.

To have an in-depth look at the items purchased by this customer, we can filter the dataset based on his CustomerID:

```
filter(gifts_df, CustomerID == 14646)
```

```
## # A tibble: 2,058 x 8
##    InvoiceNo StockCode Description        Quantity InvoiceDate         UnitPrice
##    <chr>     <chr>     <chr>                 <dbl> <dttm>                  <dbl>
##  1 539491    21981     PACK OF 12 WOODLA~       12 2010-12-20 10:09:00      0.29
##  2 539491    21986     PACK OF 12 PINK P~       12 2010-12-20 10:09:00      0.29
##  3 539491    22720     SET OF 3 CAKE TIN~        2 2010-12-20 10:09:00      4.95
##  4 539491    21931     JUMBO STORAGE BAG~        1 2010-12-20 10:09:00      1.95
##  5 539491    22613     PACK OF 20 SPACEB~        2 2010-12-20 10:09:00      0.85
##  6 539491    20751     FUNKY WASHING UP ~        1 2010-12-20 10:09:00      2.1
##  7 539491    21246     RED RETROSPOT BIG~        2 2010-12-20 10:09:00      4.95
##  8 539491    22960     JAM MAKING SET WI~        1 2010-12-20 10:09:00      4.25
##  9 539491    22355     CHARLOTTE BAG SUK~        2 2010-12-20 10:09:00      0.85
## 10 539491    21123     SET/10 IVORY POLK~        2 2010-12-20 10:09:00      1.25
## # ... with 2,048 more rows, and 2 more variables: CustomerID <dbl>,
## #   Country <chr>
```

2. Add a new column to the gifts_df object, year_month, which corresponds to the year and month from the Invoice Date. Count the number of unique invoices in each year-month combination and then arrange them in descending order in a data frame (or tibble) called q2_df. Display the first six rows of q2_df.

```
q2_df_pre = gifts_df %>%
  mutate(gifts_df, year_month = format(InvoiceDate, "%Y-%m"))
```

```
q2_df = q2_df_pre %>%
  count(InvoiceNo, year_month, sort = TRUE) %>%
  rename(count = n)

head(q2_df, n = 6)
```

```
## # A tibble: 6 x 3
##   InvoiceNo year_month count
##   <chr>     <chr>      <int>
## 1 576339    2011-11      541
## 2 579196    2011-11      532
## 3 580727    2011-12      528
## 4 578270    2011-11      441
## 5 573576    2011-10      434
## 6 567656    2011-09      420
```

3. Some items have different unit prices, even within the same invoice. Create a new object q3_df containing only those stock codes with non-unique prices over the period. It should indicate the maximum and minimum prices for that item. Display the first six rows of q3_df.

```
q3_df_pre = distinct(gifts_df, StockCode, UnitPrice)

q3_df_pre_a = q3_df_pre %>%
  group_by(StockCode) %>%
  mutate(count = n()) %>%
  filter(count > 1) #filtering out only stock codes with non-unique prices
```

```
q3_df = q3_df_pre_a %>%
  group_by(StockCode) %>%
  summarise( #creating max_price and min_price columns
    max_price = max(UnitPrice),
    min_price = min(UnitPrice)
  ) %>%
  arrange(StockCode)

head(q3_df, n = 6) #display first 6 rows of final dataset
```

```
## # A tibble: 6 x 3
##   StockCode max_price min_price
##   <chr>         <dbl>     <dbl>
## 1 10080          0.85      0.39
## 2 10125          0.85      0.42
## 3 10133          0.85      0.42
## 4 10135          2.46      0.25
## 5 11001          3.29      1.27
## 6 15034          0.14      0.07
```

4. The columns InvoiceNo and StockCode are of class character. Why is this so? Can they be safely converted to numeric columns?

InvoiceNo and StockCode are used as unique identifiers of each transaction, which may be a main reason as to ensure they are character columns.

For InvoiceNo, the column class is character possibly to prevent unnecessary computation of summary statistics. For instance, there is no point calculating the sum, mean or median of all the unique invoice numbers, as no useful insights can be derived from them. InvoiceNo is better used as a categorical variable instead to perform analyses based on the unique invoice numbers.

For StockCode, there are certain values that contain both integers and alphabets (e.g "85123A", "85099B"), which may be why the class of this column should remain as character instead of making it purely numeric.

To test if the 2 columns can be safely converted to numeric, we can try the following:

```r
q4_df = gifts_df %>%
  mutate(InvoiceNo = as.numeric(InvoiceNo),
         StockCode = as.numeric(StockCode))
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘StockCode = as.numeric(StockCode)‘.
## Caused by warning:
## ! NAs introduced by coercion
```

As seen in the above error code, the stock codes containing both integers and alphabets will be transformed into NA values upon conversion of the StockCode column from character to numeric. On the other hand, there is no issue converting the InvoiceNo column from character to numeric, since all its values contain numbers only.