Practicum 1

Chenyao Xiao, Yicheng Zhang and Isabella Motha

2022-10-15

Part 1

1

```
# Create 4 variables
doctor_type <- c("PCP", "Psychiatrist", "Surgeon", "Anesthesia")
doctor_lastname <- c("Smith", "Dame", "Jones", "Zayas")
location <- c("MA", "ME", "NH", "VT")
AVG_Rating <- c(7,9,8,9)
# Join the variables to create a data frame
doctor_df <- data.frame(doctor_type, doctor_lastname, location, AVG_Rating)
doctor_df
```

```
##    doctor_type doctor_lastname location AVG_Rating
## 1          PCP           Smith       MA          7
## 2 Psychiatrist            Dame       ME          9
## 3      Surgeon           Jones       NH          8
## 4   Anesthesia           Zayas       VT          9
```

2

```
# Select row 1 in column 2
doctor_df[1,2]
```

```
## [1] "Smith"
```

```
# Select rows 2 through 4
doctor_df[2:4,]
```

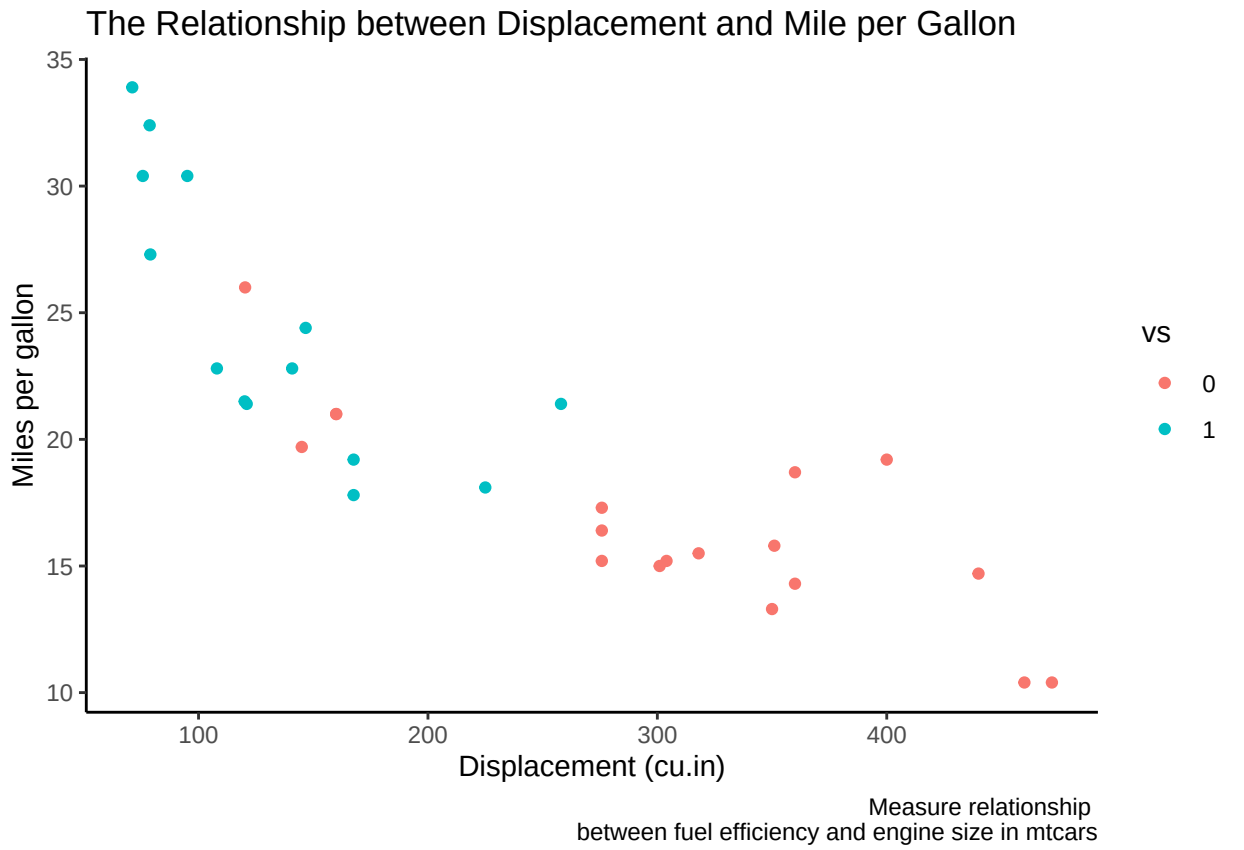
```
##    doctor_type doctor_lastname location AVG_Rating
## 2 Psychiatrist            Dame       ME          9
## 3      Surgeon           Jones       NH          8
## 4   Anesthesia           Zayas       VT          9
```

```
# Select the last column
doctor_df[, ncol(doctor_df)]
```

```
## [1] 7 9 8 9
```

3

```
mtcars$vs <- as.factor(mtcars$vs)
ggplot(data = mtcars) +
geom_point(mapping = aes(x = disp, y = mpg, color = vs))+
  labs(title= "The Relationship between Displacement and Mile per Gallon",
       x="Displacement (cu.in)",  y="Miles per gallon", caption="Measure relationship
       between fuel efficiency and engine size in mtcars")+
theme_classic()
```



Here we choose two variables: Miles per gallon to measure fuel efficiency and Displacement to measure engine size. Also, we choose a color scheme of vs(engine shape). Red spots represent V-shaped engine, while green stands for straight engines.

The scatter plot shows a negative relationship between engine displacement(cu.in) and fuel efficiency(miles per gallon). Basically, this is because bigger engine cars use more fuel and can reduce fuel efficiency. The color in this plot reveals that V-shaped engines with less efficiency use more fuel since they have bigger engine size.

4

```
# Perform a summary statistic on the dataset
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec       vs           am
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   0:18   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1:14   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71          Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85          Mean   :0.4062
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90          3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90          Max.   :1.0000
##       gear            carb
##  Min.   :3.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:2.000
##  Median :4.000   Median :2.000
##  Mean   :3.688   Mean   :2.812
##  3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :5.000   Max.   :8.000
```

```r
# calculate the pearson coefficient of the correlation R
cor(mtcars$mpg, mtcars$cyl,method = "pearson", use = "complete.obs")
```

```
## [1] -0.852162
```

Since we can get a negative correlation between displacement and mpg, we can pick other values like cylinders. More cylinders tend to need large engine size. Thus we assume that the variable cyl and mpg are negatively correlated. The correlation value -0.852 with pearson method proved that my assumption is correct. The more cylinders a car has, the lower mpg it might achieve. The purpose of this coefficient is to offer us if these two variable has association (the correct coefficient should be -1 to 1) and how is their association (positive or negative).

Part 2

1

```r
# Load the XML data directly from the URL
# it might take a while to run this chunk of code!:
fileurl <- "https://data.ny.gov/api/views/ngbt-9rwf/rows.xml"
download.file(fileurl, destfile = "rows.xml")
# parse into a data frame
rows_xml <- xmlParse("rows.xml")
ny.cd_df <- xmlToDataFrame(nodes=getNodeSet(rows_xml, "//response/row/row"))
```

2

After loading the data, we can check the structure of the data frame.

```
dim(ny.cd_df)
```

```
## [1] 99367     7
```

```
glimpse(ny.cd_df)
```

```
## Rows: 99,367
## Columns: 7
## $ year                        <chr> "2007", "2007", "2007", "2007", "2007", "20~
## $ county_of_program_location <chr> "Albany", "Albany", "Albany", "Albany", "Al~
## $ program_category            <chr> "Crisis", "Crisis", "Crisis", "Crisis", "Cr~
## $ service_type                <chr> "Medical Managed Detoxification", "Medical ~
## $ age_group                   <chr> "Under 18", "18 through 24", "18 through 24~
## $ primary_substance_group     <chr> "Heroin", "All Others", "Other Opioids", "H~
## $ admissions                  <chr> "4", "2", "6", "132", "35", "8", "1", "11",~
```

```
summary(ny.cd_df)
```

```
##      year           county_of_program_location program_category
##  Length:99367       Length:99367               Length:99367
##  Class :character   Class :character           Class :character
##  Mode  :character   Mode  :character           Mode  :character
##  service_type        age_group         primary_substance_group
##  Length:99367       Length:99367       Length:99367
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##   admissions
##  Length:99367
##  Class :character
##  Mode  :character
```

This data frame contains 7 variables and 99367 observations. According to the data dictionary, some variables needs to be cleaned as other formats. The data frame doesn't have NAs.

```
# Change year and admissions to numeric
ny.cd_df2 <-ny.cd_df %>%
  mutate(across(c(year,admissions),as.numeric))
# Find NAs
sapply(ny.cd_df2,function(x) sum(is.na(x)))
```

```
##                       year county_of_program_location
##                          0                           0
##           program_category                service_type
##                          0                           0
##                  age_group    primary_substance_group
##                          0                           0
##                 admissions
##                          0
```

Outliers increase the variability in the data, which decreases statistical power. Consequently, excluding outliers can cause our results to become statistically significant. For admission numbers here, we need to

remove outliers. When we expect our data has a normal distribution, Z-scores may help us to identify how odd the real observation is.

The degree of standard deviations around the mean decrease for each value is represented by the Z-score. The Z score equals to zero refers to the mean value, and in order to calculate it, we need to take the original admissions, calculate the mean and the use it to minus origin admission value, then divide by the SD to get the Z-score of the admission.

A huge the distance between an admission's Z-score and zero can indicate they are actually extravagant. Z $\pm$ 3 is a widely accepted cut-off point for identifying outliers, and we will apply the same logistics in our computation.

```
# Remove outliers using z score
# When z > 3, it means the data is more than
# 3 times standard deviation from the mean, which is considered an outlier
Adm <- ny.cd_df2$admissions
mean_Adm <- mean(Adm)
Std.Dev <- sd(Adm)
ny.cd_df2$z <-(mean_Adm - Adm)/Std.Dev
ny.cd_df2$z <-abs(ny.cd_df2$z)
ny.cd_df2 <- ny.cd_df2 %>%
  filter(z<=3)
```

Then we can remove column z.

```
ny.cd_df2 <- ny.cd_df2[,-8]
```

We can see that 97450 rows left after removing outliers.

3

Next we prepare four tibbles.

```
# tibble 1: county

county <- tibble(county_code = c("AL","CA","CN","DE","FR","LE","MG","ON","OL","NY",
                                 "SL","SY","SV","WR","AG","CY","CL","DU","FU","HE",
                                 "LI","NA","OD","OS","RE","SA","SE","TI","WS","NY",
                                 "CH","CO","ER","GE","JE","MA","NY","OT","OG","NY",
                                 "SC","ST","TO","WA","BM","CM","CR","ES","GR","NY",
                                 "MO","NI","OR","PU","RO","SH","SU","UL","WE","WY",
                                 "YA"),
                 county_of_program_location = c("Albany","Cattaraugus","Chenango","Delaware","Franklin"
                                                "Lewis","Montgomery","Oneida","Orleans","Queens",
                                                "Saint Lawrence","Schuyler","Sullivan","Warren",
                                                "Allegany","Cayuga","Clinton","Dutchess",
                                            "Fulton","Herkimer","Livingston","Nassau","Onondaga",
                                            "Oswego","Rensselaer","Saratoga","Seneca","Tioga",
                                            "Washington","Bronx","Chautauqua","Columbia","Erie",
                                            "Genesee","Jefferson","Madison","New York","Ontario",
                                            "Otsego","Richmond","Schenectady","Steuben","Tompkins",
                                            "Wayne","Broome","Chemung","Cortland","Essex","Greene",
                                            "Kings","Monroe","Niagara","Orange","Putnam",
```

```
                                          "Rockland","Schoharie","Suffolk","Ulster","Westchester",
                                          "Wyoming","Yates"))

# make sure they are all unique:

county %>% distinct(county_code,.keep_all = TRUE)
```

```
## # A tibble: 57 x 2
##    county_code county_of_program_location
##    <chr>       <chr>
##  1 AL          Albany
##  2 CA          Cattaraugus
##  3 CN          Chenango
##  4 DE          Delaware
##  5 FR          Franklin
##  6 LE          Lewis
##  7 MG          Montgomery
##  8 ON          Oneida
##  9 OL          Orleans
## 10 NY          Queens
## # ... with 47 more rows
```

```
# tibble 2 : program_category:

program_category <-tibble(program_code = c("CR", "IN","OTP", "RE", "OU", "SP"),
                          program_category = c("Crisis", "Inpatient", "Opioid Treatment Program",
                                               "Residential", "Outpatient", "Specialized"))
program_category %>% distinct(program_code,.keep_all = TRUE)
```

```
## # A tibble: 6 x 2
##   program_code program_category
##   <chr>        <chr>
## 1 CR           Crisis
## 2 IN           Inpatient
## 3 OTP          Opioid Treatment Program
## 4 RE           Residential
## 5 OU           Outpatient
## 6 SP           Specialized
```

```
# tibble 3:primary_substance_group

primary_substance_group <- tibble(substance_code=c("H","A","AO","C","M","N","O"),
                                  primary_substance_group= c("Heroin", "Alcohol",
                                  "All Others", "Cocaine", "Marijuana","None","Other Opioids"))
primary_substance_group %>% distinct(substance_code,.keep_all = TRUE)
```

```
## # A tibble: 7 x 2
##   substance_code primary_substance_group
##   <chr>          <chr>
## 1 H              Heroin
## 2 A              Alcohol
## 3 AO             All Others
```

```
## 4 C              Cocaine
## 5 M              Marijuana
## 6 N              None
## 7 O              Other Opioids
```

To get *admissions-data* tibble, we need to merge our tibbles for code into original data and select columns properly.

```
# tibble 4 admission_data:
# Instead include a column with their respective foreign keys
ny.cd_df3 <-left_join(ny.cd_df2,county,by="county_of_program_location")
ny.cd_df3 <-left_join(ny.cd_df3,program_category,by="program_category")
ny.cd_df3 <-left_join(ny.cd_df3,primary_substance_group,by="primary_substance_group")

# Select and rename columns we need
admissions_data <- ny.cd_df3 %>%
                 select(year,county_code, program_code, service_type, age_group,
                        substance_code,admissions) %>%
                 rename("county_of_program_location" = "county_code",
                        "program_category" = "program_code",
                       "primary_substance_group" = "substance_code")

# check our result:

str(admissions_data)
```

```
## 'data.frame':    97450 obs. of  7 variables:
##  $ year                       : num  2007 2007 2007 2007 2007 ...
##  $ county_of_program_location: chr  "AL" "AL" "AL" "AL" ...
##  $ program_category           : chr  "CR" "CR" "CR" "CR" ...
##  $ service_type               : chr  "Medical Managed Detoxification" "Medical Managed Detoxification"
##  $ age_group                  : chr  "Under 18" "18 through 24" "18 through 24" "18 through 24" ...
##  $ primary_substance_group    : chr  "H" "AO" "O" "H" ...
##  $ admissions                 : num  4 2 6 132 35 8 1 11 276 135 ...
```

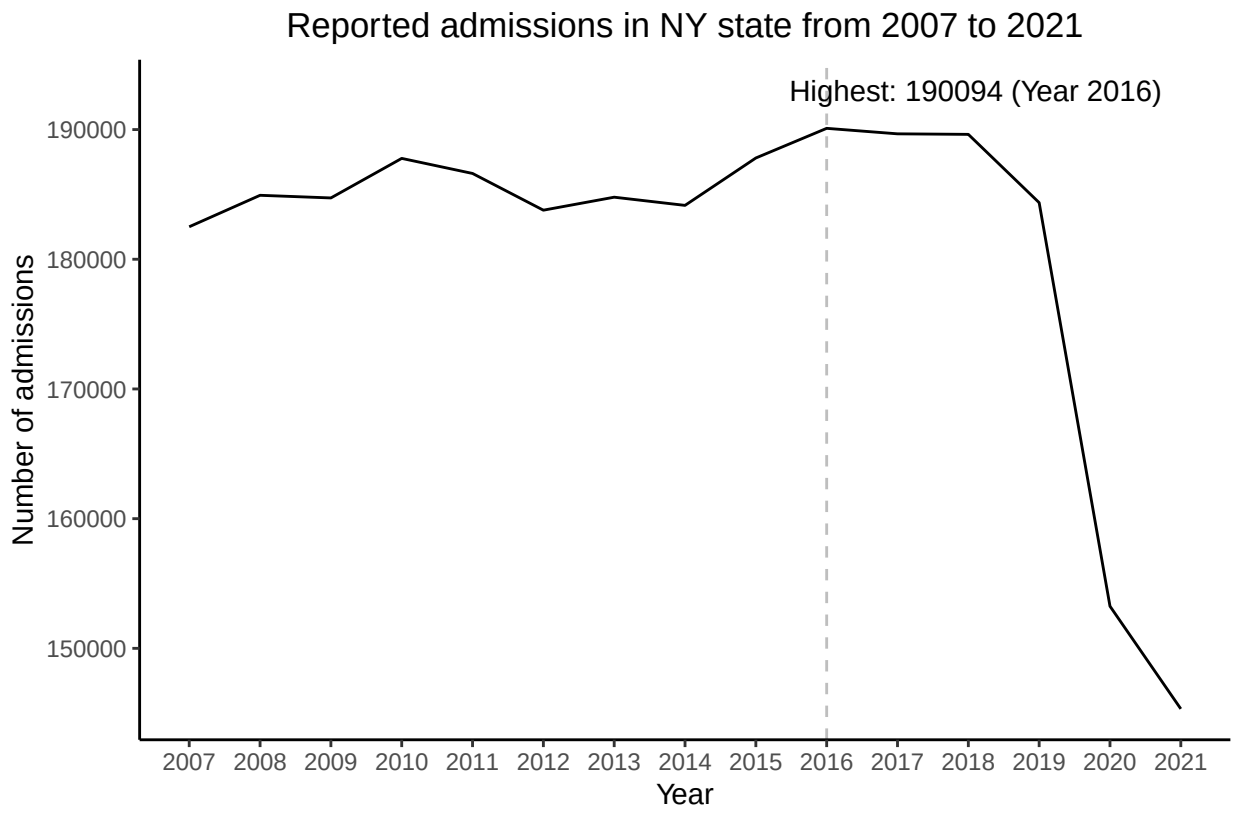
```
head(admissions_data)
```

```
##   year county_of_program_location program_category
## 1 2007                          AL               CR
## 2 2007                          AL               CR
## 3 2007                          AL               CR
## 4 2007                          AL               CR
## 5 2007                          AL               CR
## 6 2007                          AL               CR
##                     service_type     age_group primary_substance_group
## 1 Medical Managed Detoxification      Under 18                       H
## 2 Medical Managed Detoxification 18 through 24                      AO
## 3 Medical Managed Detoxification 18 through 24                       O
## 4 Medical Managed Detoxification 18 through 24                       H
## 5 Medical Managed Detoxification 18 through 24                       A
## 6 Medical Managed Detoxification 25 through 34                      AO
##   admissions
## 1          4
```

```
## 2          2
## 3          6
## 4        132
## 5         35
## 6          8
```

4

```
annualAdmissions <- function(){
    data_4 <- admissions_data %>%
    select(year,admissions)%>%
    group_by(year) %>%
    summarise(sum_adm=sum(admissions))
    ggplot(data_4, aes(x=year, y=sum_adm))+
    geom_line()+
    scale_x_continuous(breaks=seq(2007, 2021, 1))+
    geom_vline(aes(xintercept=2016),color="grey", linetype="dashed", size=0.5)+
    annotate("text",x=2016,y=193000,hjust=.1,label="Highest: 190094 (Year 2016)")+
    labs(title="Reported admissions in NY state from 2007 to 2021",
         x="Year", y="Number of admissions",
         caption = "Admission numbers in NY state reached the highest number in 2016")+
      theme_classic()+
      theme(plot.title = element_text(hjust = 0.5))
}
annualAdmissions()
```



From this result we can see that the general trend of the admission due to substance abusing is decreasing after we arrive the peak point on 2016 with 190094 individual admission cases. And then start from 2018 the trend is keeping decrease at we arrive the lowest point in 2021, which I believe we shold give credit to the improvement of the public health system in recent decades otherwise we may not have a such dramatic falling of admission numbers since 2016.

5

To ensure correct merging, we need to check the code for the county as unique. So we recode the county table here.

```
# Prepare data
# since 5 counties share the same code "NY",
# we want to rename their county code to make sure they are unique:
county_unique <- tibble(county_code = c("AL","CA","CN","DE","FR","LE","MG","ON","OL","NYQ",
                                        "SL","SY","SV","WR","AG","CY","CL","DU","FU","HE",
                                        "LI","NA","OD","OS","RE","SA","SE",
                                        "TI","WS","NYB","CH","CO","ER","GE","JE","MA","NY",
                                        "OT","OG","NYR","SC","ST","TO","WA",
                                        "BM","CM","CR","ES","GR","NYK","MO","NI","OR","PU",
                                        "RO","SH","SU","UL","WE","WY","YA"),
                        County_of_Program_Location = c("Albany","Cattaraugus","Chenango",
                                                       "Delaware","Franklin","Lewis","Montgomery",
                                                       "Oneida","Orleans","Queens","Saint Lawrence","Schuyler"
                                                     "Fulton","Herkimer","Livingston","Nassau",
                                                     "Onondaga","Oswego","Rensselaer","Saratoga",
```

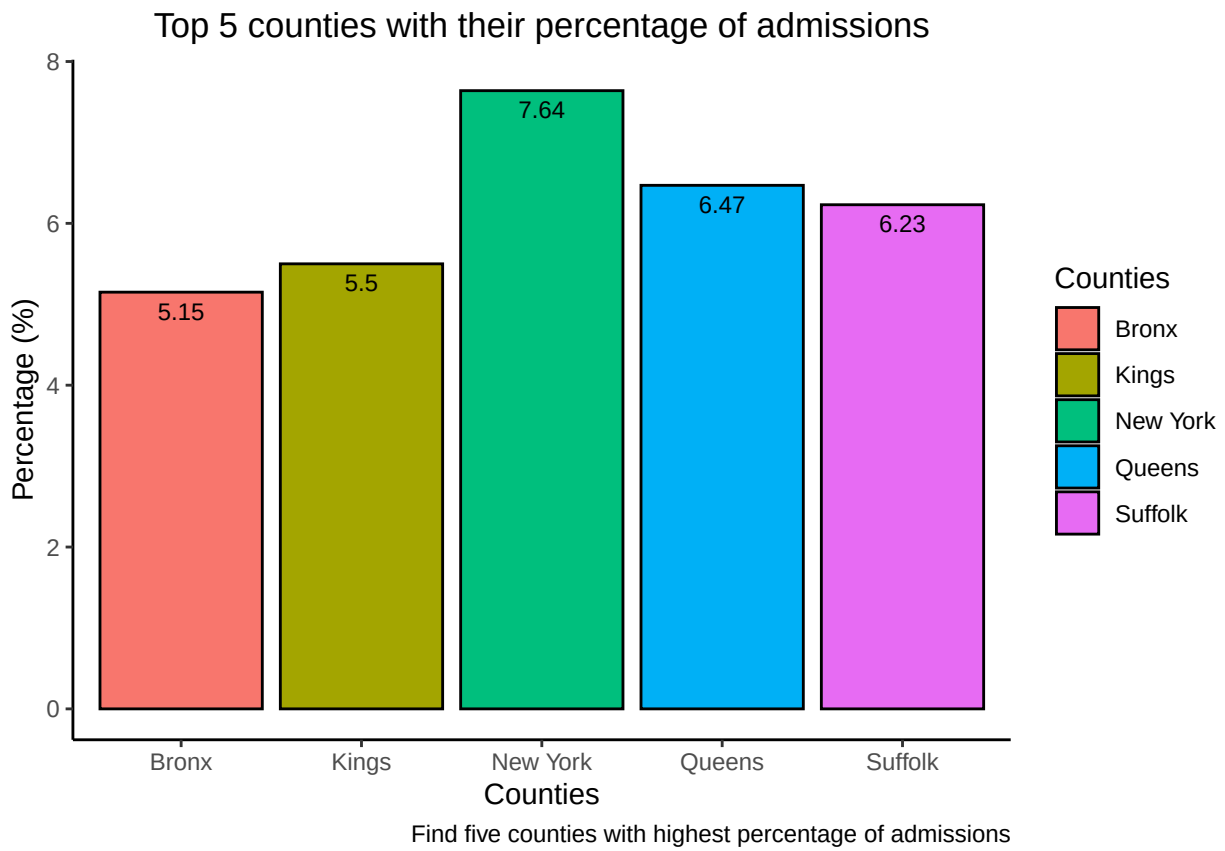
```
                                                  "Seneca","Tioga","Washington","Bronx",
                                                "Chautauqua","Columbia","Erie","Genesee",
                                                  "Jefferson","Madison","New York","Ontario","Otsego","Ri
                                                  "Chemung","Cortland","Essex","Greene",
                                                "Kings","Monroe","Niagara","Orange",
                                                  "Putnam","Rockland","Schoharie","Suffolk",
                                                "Ulster","Westchester","Wyoming","Yates"))
county_unique <- county_unique %>% distinct(county_code,.keep_all = TRUE)
# Redo admission data tibbles
admissions_data2 <- merge(x=county_unique,y=ny.cd_df2,by.x="County_of_Program_Location",
                          by.y = "county_of_program_location",all= T)
admissions_data2 <- merge(x=primary_substance_group, y=admissions_data2, by.x="primary_substance_group"
                          by.y = "primary_substance_group", all= T)
admissions_data2 <- merge(x=program_category, y=admissions_data2, by.x="program_category",
                          by.y="program_category",all = T)
# Select and rename columns we need
admissions_data2 <- admissions_data2 %>%
                  select(year,county_code, program_code, service_type, age_group,
                         substance_code,admissions) %>%
                  rename("county_of_program_location" = "county_code",
                         "program_category" = "program_code",
                        "primary_substance_group" = "substance_code")
```

Basically what we do is display question 3 in unique county codes.

```
# Merge tibbles back
colnames(admissions_data2)[2]<-"county_code"
admissions_data_new<-left_join(admissions_data2,county_unique,by="county_code")

# Group and summarise
admissions_data_new <- admissions_data_new %>%
  group_by(County_of_Program_Location) %>%
  summarise(adm_number= sum(admissions)) %>%
  mutate(percentage=round(adm_number/sum(adm_number),4)*100) %>%
  top_n(5,percentage)
```

```
# Visualize in a bar chart
ggplot(data=admissions_data_new, aes(x=County_of_Program_Location, y=percentage,
                                     fill=County_of_Program_Location)) +
  geom_bar(stat="identity",color="black")+
  geom_text(aes(label=percentage), vjust=1.6, size=3)+
  labs(title= "Top 5 counties with their percentage of admissions",
       fill= "Counties",x="Counties", y= "Percentage (%)",
       caption="Find five counties with highest percentage of admissions")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))
```



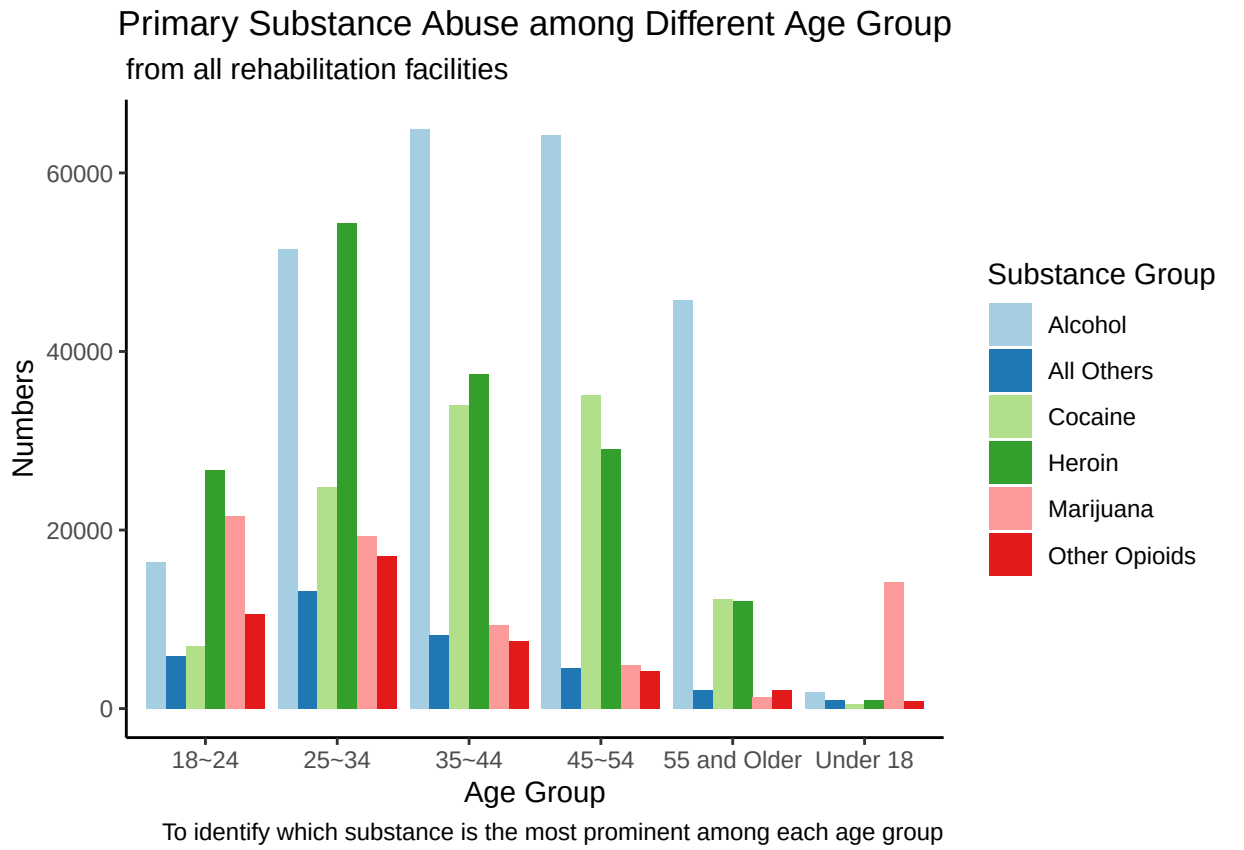
The top five counties from 2007 to 2021 are New York(city), Queens, Suffolk, Kings and Bronx. New York city has the highest admission percentage of 7.64% and it's clearly higher than other counties.From this result, we can see that the new york city has the highest admission percentage from 2008-2020 and it is way more higher than other counties. The Bronx county has the lowest in these 5 counties and it is very close to kings. This shows more admission cases in New York(city), but the New York city has large population so this is reasonable.

6

```
# extract all admissions to the various "Rehab" facilities.ny.cd_df3
substance2age_admissions_data <- ny.cd_df3 %>%
                  filter(str_detect(service_type, ".*Rehab.*"))%>%
                  group_by(age_group,primary_substance_group) %>%
                  summarise(substance_admission_total =sum(admissions))
                  # new column here refers to the total admissions number
```

```
ggplot(substance2age_admissions_data, aes(x=age_group,y=substance_admission_total,fill=primary_substanc
         geom_col(position=position_dodge())+
  scale_fill_brewer(palette="Paired")+
  labs(fill="Substance Group",
       caption="To identify which substance is the most prominent among each age group",
       x="Age Group", y="Numbers",
       title="Primary Substance Abuse among Different Age Group",
       subtitle="from all rehabilitation facilities")+
  # To display groups clearer, we reformat some x labels
```

```
scale_x_discrete("Age Group",
  labels = c(
    "18 through 24" = "18~24",
    "25 through 34" = "25~34",
    "35 through 44" = "35~44",
    "45 through 54" = "45~54"
  )
)+ theme_classic()+
theme(plot.title = element_text(hjust = 0.5))
```



To identify which substance is the most prominent among each age group

From the result, we can see that the marijuana take majority of substance using cases for under 18. Drug use is more seen in middle age groups. In short, the prominent substance for people under 18 is marijuana, and 18 to 24 is Heroin. For age between 25 to 34, the prominent substance is heroin, though alcohol also has a high abusing cases. For people from 35 to 44 and 45 to 54, the prominent substance is alcohol. People older than 55 tend to have more issues with alcohol as well. Nearly all age groups have large cases of alcohol abuses, and it could be consider as prominent as prominent in these age groups. Heroin is widely seen as well but most prominent in age 18 to 24 and 25 to 34.Marijuana is prominent for people under 18.