Lab Session 8

1. mtcars dataset

a. Check to see if you have the mtcars dataset by entering the command

```
mtcars.
```

b. What class is mtcars? `class(mtcars)`

c. How many observations (rows) and variables (columns) are in the mtcars dataset?

```
dim(mtcars)
nrow(mtcars)
ncol(mtcars)
```

d. Copy mtcars into an object called cars and rename mpg in cars to MPG. Use rename().

```
cars=mtcars
cars=rename(cars, MPG=mpg)
```

e. Convert all the column names of cars to upper case. Use rename_all, and the toupper command (or colnames).

```
Mt_upper=rename_all(cars,toupper)
toupper(colnames(cars))
```

f. Convert the rownames of cars to a column called car using rownames_to_column. Subset the columns from cars that end in "p" and call it pvars using ends_with().

```
cars = tibble::rownames_to_column(mtcars, var = "car")
head(cars)
pvars = select(cars, ends_with("p"))
head(pvars)
```

g. Create a subset cars that only contains the columns: wt, qsec, and hp and assign this object to carsSub. What are the dimensions of carsSub? (Use select() and dim().)

```
carsSub = select(cars, wt, qsec, hp)
dim(carsSub)
```

h. Convert the column names of carsSub to all upper case. Use rename_all(), and toupper() (or colnames()).

```
carsSub = rename_all(carsSub, toupper)
```

i. Subset the rows of cars that get more than 20 miles per gallon (mpg) of fuel efficiency. How many are there? (Use filter().)

```
cars_mpg = filter(cars, mpg > 20)
select(cars_mpg,mpg,hp)
```

j. Subset the rows that get less than 16 miles per gallon (mpg) of fuel efficiency and have more than 100 horsepower (hp). How many are there? (Use filter().)

```
nrow(filter(cars, mpg < 16 & hp > 100))
```

k. Create a subset of the cars data that only contains the columns: wt, qsec, and hp for cars with 8 cylinders (cyl) and reassign this object to carsSub. What are the dimensions of this dataset?

```
carsSub = filter(cars, cyl == 8)
carsSub = select(carsSub, wt, qsec, hp, car)
dim(carsSub)
```

l. Re-order the rows of carsSub by weight (wt) in increasing order. (Use arrange().)

```
carsSub = arrange(carsSub, wt)
```

m. Create a new variable in carsSub called wt2, which is equal to wt^2, using mutate() and piping %>%.

```
carsSub %>% mutate(wt2 = wt^2)
```

2. Bike_Lane dataset

```
bike = read.csv("http://johnmuschelli.com/intro_to_r/data/Bike_Lanes.csv")

bike
```

1. How many bike “lanes” are currently in Baltimore? You can assume each observation/row is a different bike “lane”

```
dim(bike)
```

2. How many (a) feet and (b) miles of bike “lanes” are currently in Baltimore?

```
sum(bike$length)/5280
```

3. How many types of bike lanes are there? Which type has (a) the most number of and (b) longest average bike lane length?

```
colnames(bike)

bike$length

length(unique(bike$type))


m1=bike %>%

  group_by(type) %>%

  summarise(number_of_rows = n(),

            mean = mean(length)) %>%

  arrange(desc(mean))

m1[1,]
```

4. How many different projects (project) do the bike lanes fall into? Which project category has the longest average bike lane length?

```
length(unique(bike$project))

avg = bike %>%

  group_by(project) %>%

  summarize(mn = mean(length, na.rm = TRUE)) %>%

  filter(mn == max(mn))

avg
```

5. What was the average bike lane length per year that they were installed? (Be sure to first set dateInstalled to NA if it is equal to zero.)

```
bike = bike %>% mutate(
  dateInstalled = ifelse(
    dateInstalled == 0,
    NA,
    dateInstalled)
)
print(mean(bike$length[ !is.na(bike$dateInstalled)]))
```

6. (a)Numerically and (b) graphically describe the distribution of bike lane lengths (length).

```
# Numeric summary

quantile(bike$length)

hist(bike$length)

boxplot(bike$length~bike$type)
```