**CAP 2571-Tools for Data Science** **Solution to Homework 1 Fall 2023**

**Sample 1**

#Install the tidyverse R package and read the mpg dataset

library(tidyverse)

#Store mpg in a variable

mpg\_new <- mpg

# Plot the relationship between engine size (displ) and fuel efficiency (hwy)

ggplot( data = mpg) + geom\_point(mapping = aes(x = displ, y = hwy, color = class))

# Generate a scatterplot of **hwy** vs **cyl** in the mpg dataset

ggplot( data = mpg) + geom\_point(mapping = aes(x = hwy, y = cyl, color = class))

# Compute and display the total number of 4-, 5-, 6- and 8-cylinder vehicles in the dataset.

table(mpg$cyl)

#Display the total number of 4-, 5-, 6- and 8-cylinder vehicles

ggplot(mpg, aes(x = cyl)) + geom\_bar(fill = "steelblue") + geom\_text(stat = "count", aes(label = ..count..), vjust=1.6, color = "white", size=3.5)

# Compute and display the details (manufacturer, model, etc.) of the most fuel-efficient vehicles in the dataset.

n\_index <- which.max(mpg$hwy)

n\_index

mpg[n\_index,]

# Compute and display the details (manufacturer, model, etc.) of the least fuel-efficient vehicles in the dataset.

n\_index2 <- which.min(mpg$hwy)

n\_index2

mpg[n\_index2,]

**Sample 2**

#Loading the tidyverse and ggplot 2 libraries

library(tidyverse)

library(ggplot2)

#Reading and loading the mpg dataset to a variable

workingdata = mpg

#creating a plot for relationship between displ and hwy

#and making chart for cars grouped by number of cylinders

ggplot(data = workingdata) + geom\_point(mapping = aes(x = displ, y = hwy, color = class))

# Generate a scatterplot of **hwy** vs **cyl** in the mpg dataset

ggplot( data = mpg) + geom\_point(mapping = aes(x = cyl, y = hwy, color = class))

#Compute the total number of 4-, 5-, 6- and 8-cylinder vehicles

table(mpg$cyl)

ggplot(data = workingdata, aes(x=cyl,color=cyl)) + geom\_bar()

#grabs the row index for the highest and lowest value in cty and hwy

index\_max\_cty = which.max(workingdata$cty)

index\_max\_hwy = which.max(workingdata$hwy)

index\_min\_cty = which.min(workingdata$cty)

index\_min\_hwy = which.min(workingdata$hwy)

#prints all the details of the highest and lowest value in cty and hwy

print(workingdata[index\_max\_cty,])

print(workingdata[index\_min\_cty,])

print(workingdata[index\_max\_hwy,])

print(workingdata[index\_min\_hwy,])