Lab #1[1]

Purpose: This is our first lab to do something with data and visualize the results.

Before the lab, please download data AutoLab1.csv and save it to your local drive.

1. Load data AutoLab1.csv

You can use function read.table() or read.csv() to import data into R, depending on the file type. You can use help(read.csv) to find more about how to use this function. Since the file type is .csv, we use read.csv() in this case. Use the following command to load AutoLab1.csv into R and store it as an object called Auto.

```
Auto=read.csv(file.choose(),header=T)
```

In this command, file.choose() locates the file from your local drive, the option header=T (or header=TRUE) means that the first line of the file contains the variable names.

2. Data Structure

6

You can use head(Auto) to look at the first few rows. Function dim(Auto) can output the number of rows followed by the number of columns and function str (Auto) can return the data structure.

```
head(Auto)
dim(Auto)
str(Auto)
```

3. Summary Statistics

We can use function summary() to output the summary statistics of data.

```
summary(Auto)
```

4. Data Restructuring

The variable cylinders (i.e. the second column) is stored as a quantitative variable. As it only has a small number of possible values, we can convert it to a qualitative variable. Function as.factor() converts a quantitative variable into a qualitative variable. Check the summary statistics of cylinders to see if it is the same as the output in Step 3.

```
Auto[,2] =as.factor (Auto[,2])
summary(Auto[,2])
```

5. Graphs

5.1 Scatterplot

We can use function plot() to produce plots. However, simply typing the variable names does not work, because R does not know where to find those variables. You can either use the "$"sign,

```
plot(Auto$horsepower , Auto$mpg )
```

Don't need it if use attach ( ) function.

or use function attach() to tell R to make the variables available by name. Function names() lists all variable names. Then you can use the variable name directly.

[1] Acknowledgement: some of the contents are borrowed with or without modification from An Introduction to Statistical Learning, with applications in R (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R.Tibshirani.

x limit y limit (Depends on ur needs)

main title

```
attach(Auto)
names(Auto)
plot(horsepower , mpg, col ="red", xlab="Horsepower",ylab ="MPG ",xlim=c(30,250), ylim=c(5,50),
main="Horsepower vs. MPG", cex.main=1.75)
```

title size

Here the option col ="red" tells R that color data points red. The options xlab="Horsepower",ylab ="MPG ",and main="Horsepower vs. MPG", tell R the x axis title, the y axis title and the main title respectively. The options xlim and ylim tell R the range of x axis and y axis. Cex is the number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, 0.5 is 50% smaller, etc. Cex.main indicates the magnification of titles relative to cex.

There are many other optional parameters in function plot(), which we do not include in this case. You can use help(plot) or ?plot to explore more about them.

We can use function par(mfrow=c(nrows,ncols)) to combine multiple plots into one graph. For example, par(mfrow=c(3,1)) indicates that three figures will be arranged in 3 rows and 1 column. Now let's generate another two figures and arrange them in one column.

number of rows num. of columns

```
par(mfrow=c(1,2))
plot(acceleration , mpg, col ="red", xlab="Acceleration",ylab ="MPG ", main="Acceleration vs. MPG",
cex.main=1.75)
plot(weight , mpg, col ="red", xlab="Weight",ylab ="MPG ", main="Weight vs. MPG", cex.main=1.75)
```

The pairs() function creates a scatterplot for every scatterplot pair of variables. We can also produce scatterplots matrix for just a subset of the variables.

relationships between variables (if columns ≥ 10, don't suggest)

```
pairs(Auto)
pairs(Auto[c(3:5)])
```

only from third to fifth columns

5.2 Barplot

If the variable plotted on the x-axis is categorical, then boxplots will automatically be produced by function plot().

only have 1 plot

put categorical in X & continuous in Y

```
par(mfrow=c(1,1))
plot(cylinders , mpg , col ="red", varwidth =T, xlab=" Cylinders ", ylab ="MPG ", main="Cylinders vs.
MPG")
```

size of the box reflect the number of data in each category

5.3 Histogram

We can use hist() function to plot a histogram. The option "breaks=10" sets the total number of bins.

```
hist(mpg , breaks =10, col ="red", xlab ="MPG ",xlim=c(0,50),main="Histogram of MPG")
hist(horsepower , breaks =20, col ="red", xlab ="Horsepower ",xlim=c(0,250),main="Histogram of
Horsepower")
```

This concludes lab #1.

we only have 1 variable