datasciencedojo
unleash the data scientist in you

Data Exploration, Visualization, and Feature Engineering

Supplemental Worksheet

Data Science Dojo

Important Commands

It is important to explore any dataset by using these common commands:

- **head( ) and tail ( )**
  first 6 rows of data and last 6 rows of data
- **str( )**
  Internal structure of data (number of features, types of data)
- **colnames( )**
  identify column names
- **View( )**
  view as a spreadsheet in RStudio
- **dim( )**
  rows and columns of data
- **summary( )**
  5-number summary and mean, and missing values

R Graphics

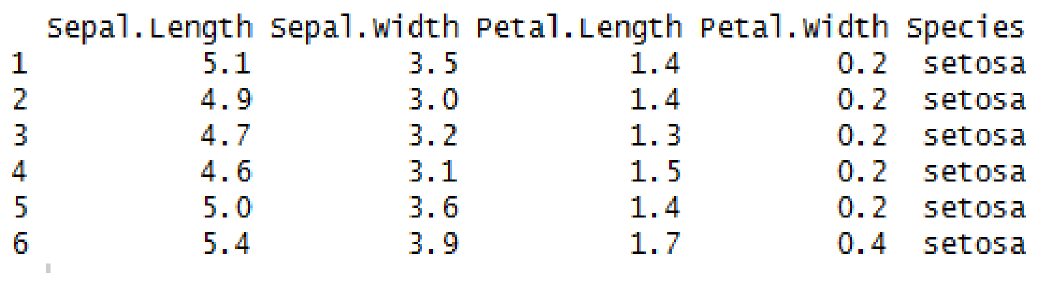
Iris Dataset

The Iris dataset is a classic dataset used for teaching data visualization and exploration, introductory classification, and machine learning. It is a very simple dataset that contains 3 classes (species) of 50 instances each, where each class refers to a type/species of iris plant – setosa, versicolor, virginica. There are 5 features/columns in the dataset – Sepal Length, Sepal Width, Petal Length, Petal Width and Species.

Load the dataset by using data(iris).

<u>Exercise:</u> Look at the first 6 rows of this dataset.

```
head(iris)
```



```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

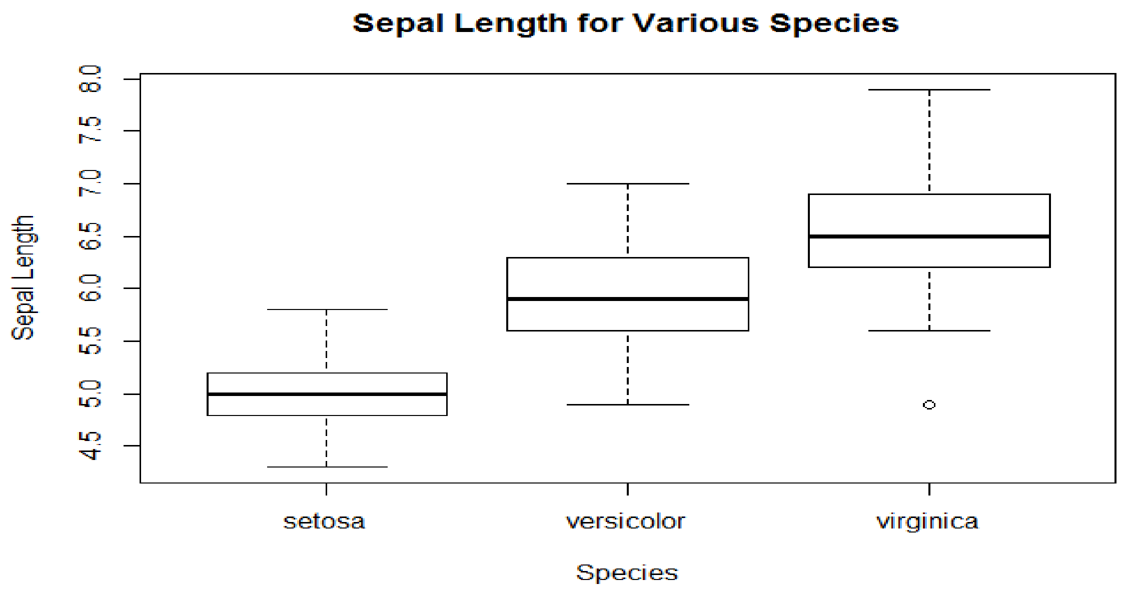
Fill in the blanks:

1. The Iris dataset has ______ rows by ______ columns.
2. There are ______missing values in the dataset.

3. What is the minimum and maximum values of the dataset?
   Minimum: ______ Maximum: ______

Boxplot

Create a boxplot of Sepal Length for all three species. Feel free to use the code in GitHub. You should have a graph like the following:



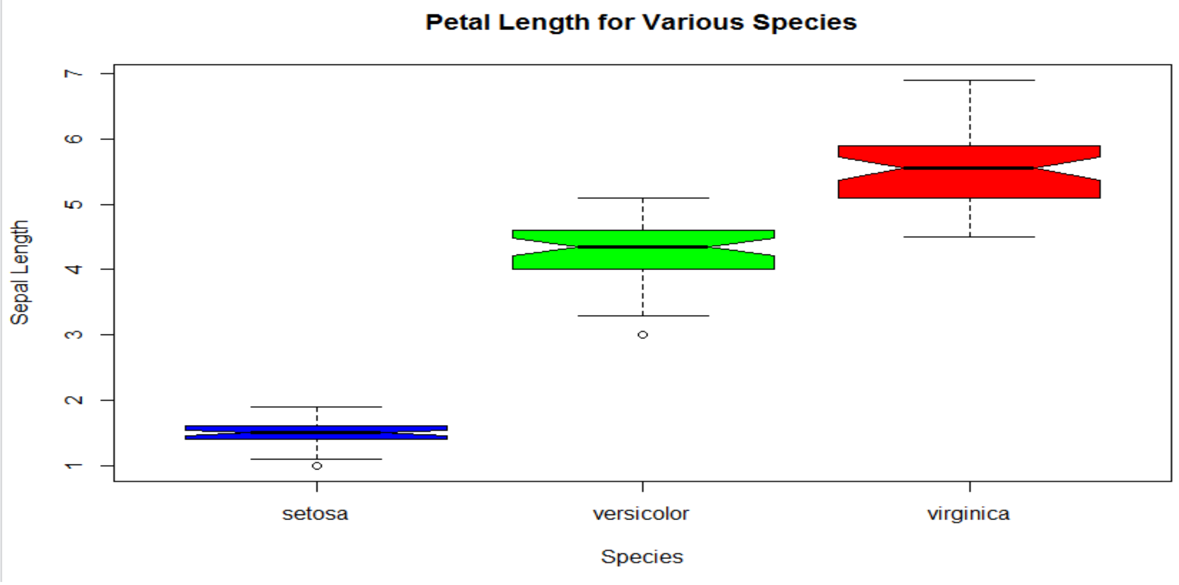
Questions:

1. What is the maximum Sepal Length for virginica? ______
2. What is the median value of Sepal Length for versicolor? ______
3. Does the boxplot have any outliers? ______

Exercise:

Create a boxplot for Petal Length for all three species.
Label the graph as "Petal Length for Various Species" using **main** function
Color the boxplots "blue", "green" and "red" using **col** function.
Create notched boxplots by using **notch = TRUE**

```
boxplot(Petal.Length ~ Species, data= iris, main= "Petal Length
for Various Species", xlab ="Species", ylab ="Sepal Length",
notch=TRUE, col = c("blue", "green", "red"))
```



Saving Plots

Save the above plot you just created.

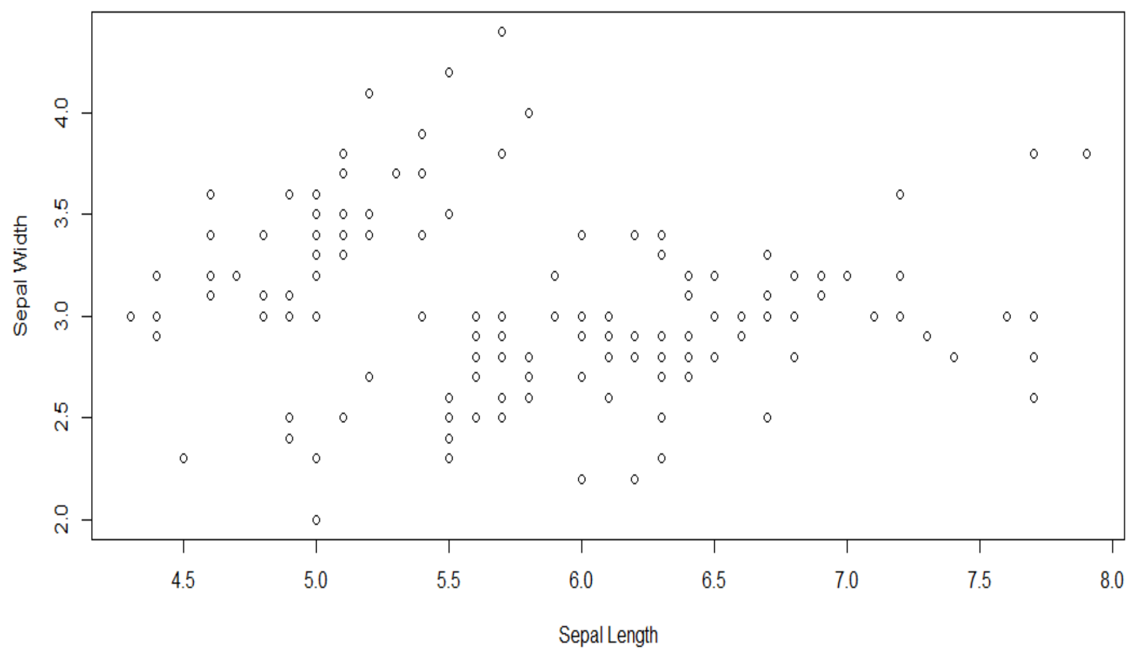
```
pdf("myplot.pdf")
boxplot(Petal.Length ~ Species, data=iris,
main="Petal Length for Various Species",
xlab= "Species", ylab="Petal Length",
notch=TRUE, col=c("blue","green","red"))
dev.off() # Returns plot to the IDE
```

R Core- Plot Function

The plot function is used to plot two numerical variables and helps in determining a relationship between them.

Using the iris dataset to create a scatterplot of Sepal Length versus Sepal Width. Make sure to label the x and y axes, and title the graph.

```
plot(Sepal.Length, Sepal.Width, xlab= “Sepal Length”, ylab=
“Sepal Width”)
```



Questions:

1. What do you notice about the plot? Is there a relationship between Sepal Width and Length?
2. What can we do to the graph to better understand the relationship?

Lattice Graphics

Some of the problems we encountered with the previous scatterplot basic function was the inability to distinguish between the different species. We will introduce another type of scatterplot that will allow us to segment using a categorical variable, in this case species of iris plants.

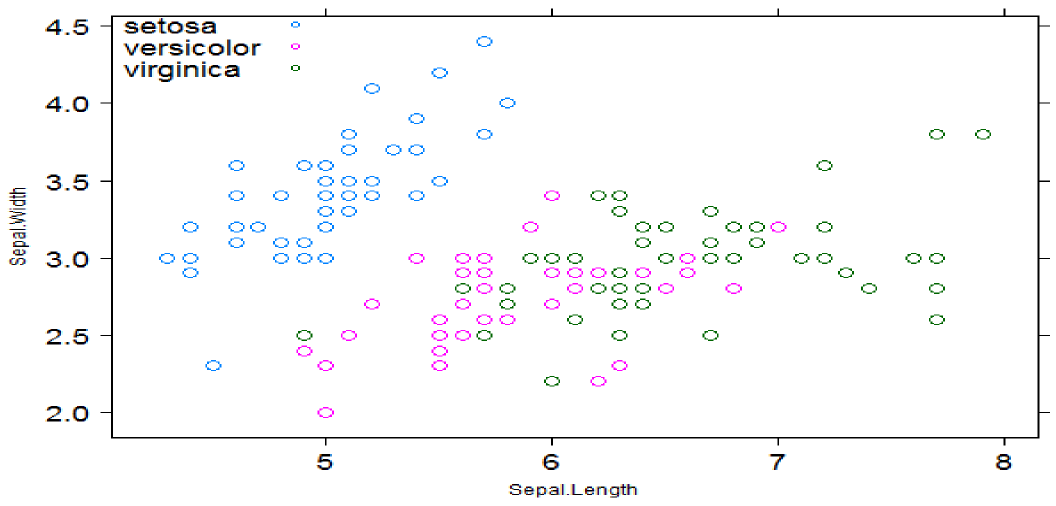
Graphical Parameters

- You can customize many features of the graph using the **par** function.
- Text and symbol size controlled by **cex** function.
- Plotting symbols controlled by **pch** function.
- Line width controlled by **lwd** function.
- Legend details are controlled by **auto.key** function.

Lattice xyplot function

The xyplot function is part of the lattice graphics and will need the lattice package to be installed. The xyplot produces bivariate scatterplots of numeric quantities. The syntax for writing the xyplot is slightly different than the plot function. The “auto.key” argument is used to automatically produce a legend.

```
# Lattice Graphics
install.packages(lattice)
library(lattice)
xyplot(Sepal.Width ~ Sepal.Length, data=iris, groups=Species,
auto.key=list(corner=c(0,0), x=0, y=0.85, cex=1.5), cex=1.5,
scales=list(cex=1.5))
```



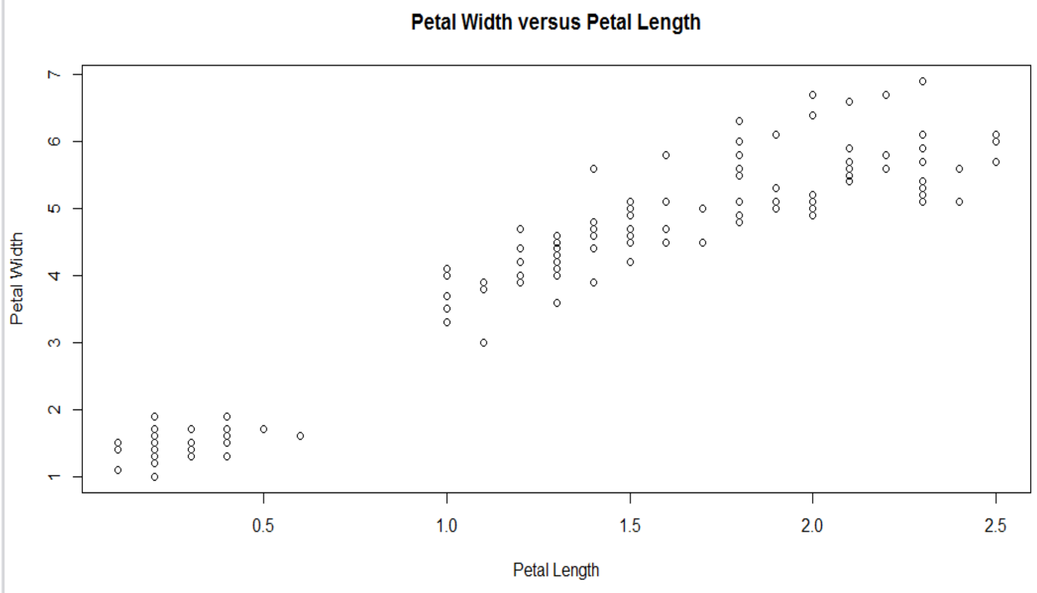
In-class Exercise 1

Compare and contrast 2D scatterplots for the iris dataset. Summarize your findings.
Create scatterplots of Petal Length versus Petal Width using the xyplot function.

Sample Solutions for Exercise 1

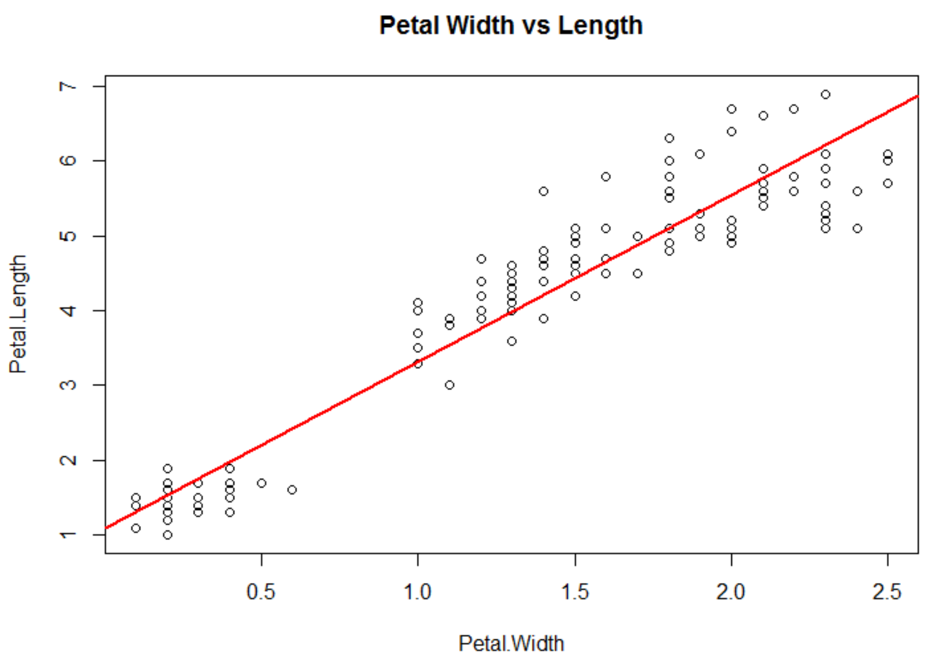
Plot of Petal Width versus Petal Length using plot( )

```
# Core Graphics
plot(Petal.Length ~ Petal.Width, data=iris)
```



Plot of Petal Length versus Petal Width using plot( ) with a regression line.

```
# Core Graphics
abline(lm(Petal.Length ~ Petal.Width,data=iris),col="red")
```

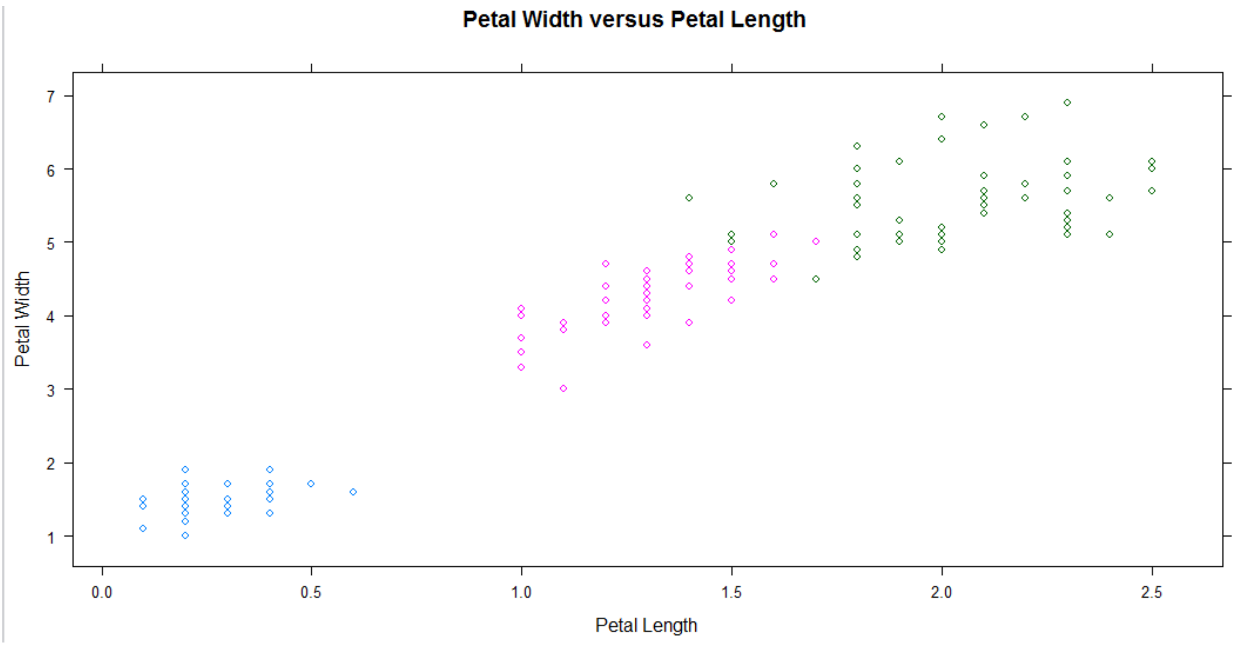


```
# Find correlation between variables
Cor(iris$Petal.Width,iris$Petal.Length)
[1] 0.96
```

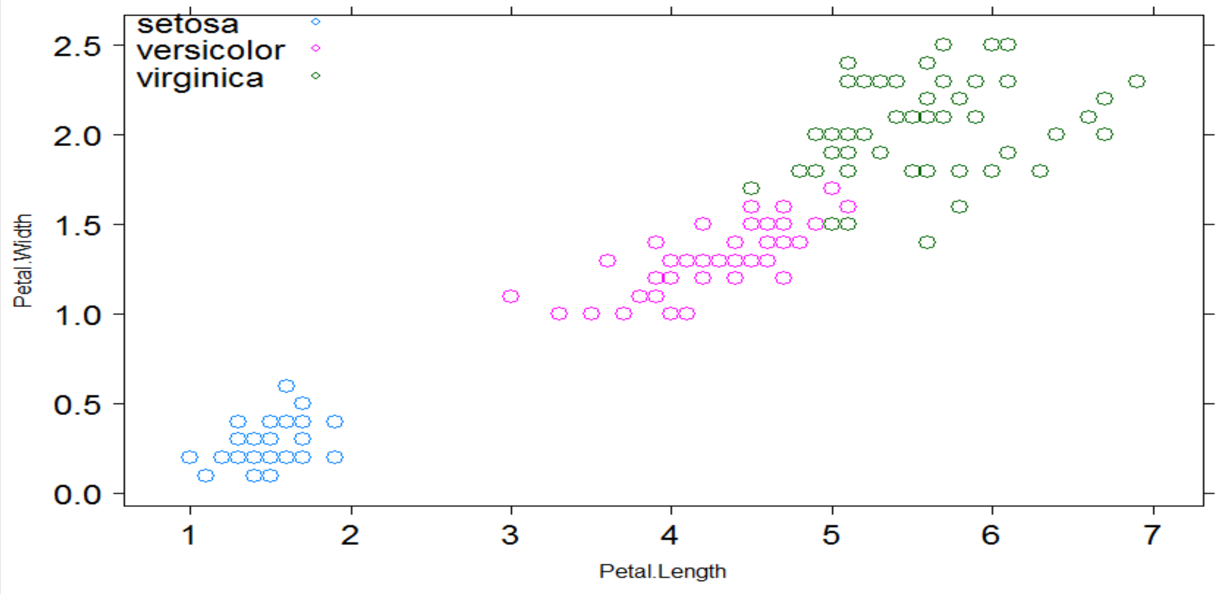
Since the value is close to 1, there is a strong correlation between two variables.

Plot of Petal Width versus Petal Length using xyplot( )

```
# Lattice Graphics
xyplot(Petal.Length ~ Petal.Width, data=iris, groups =
Species)
```



```
# Lattice Graphics
xyplot(Petal.Width ~ Petal.Length, data=iris, groups=Species,
auto.key=list(corner=c(0,0), x=0, y=0.85, cex=1.5), cex=1.5,
scales=list(cex=1.5))
```



R Core Histogram

A histogram is a graphical display of numerical data using bars of different heights to represent the frequency of that variable. It shows the frequency distribution of a quantitative variable.

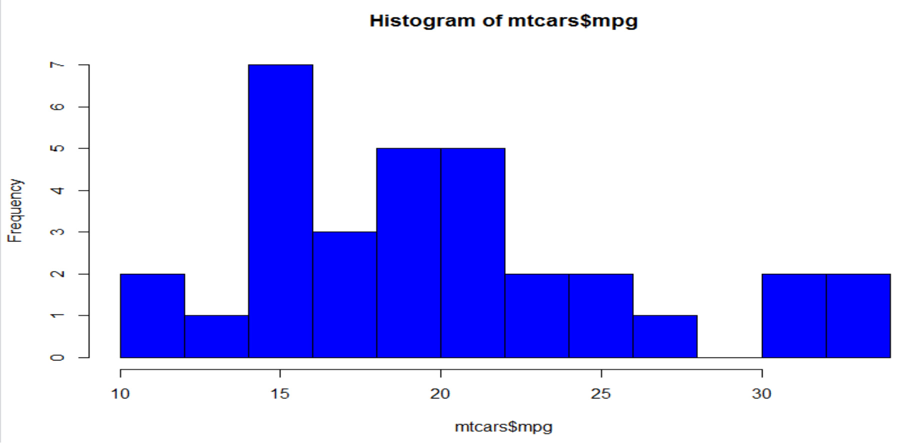
Exercise:

Create a histogram of mpg using mtcars dataset

Change the number of breaks or bins to 12 using breaks

Color the graph blue using col function.

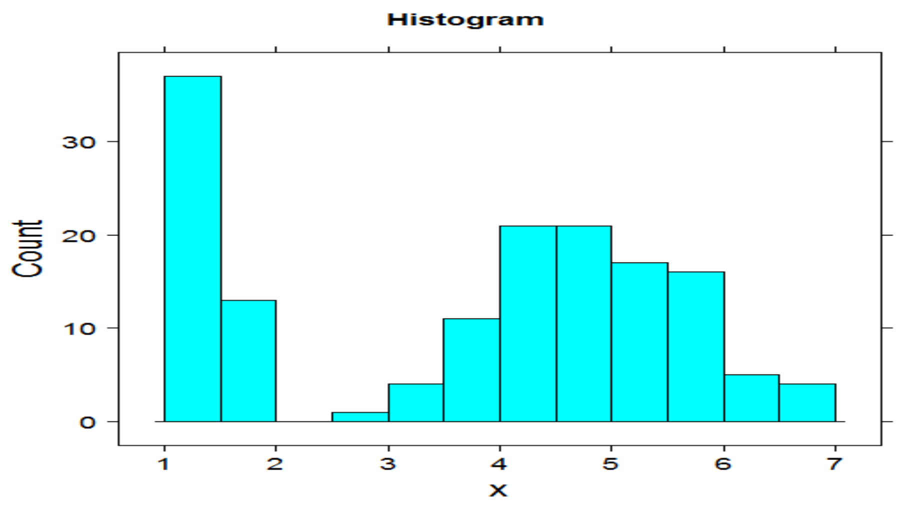
```
# Core R
Data(mtcars)
hist(mtcars$mpg,breaks=12,col="blue")
```



Lattice Histogram

We can also create better histograms using the histogram function in lattice package.

```
#Lattice Graphics
histogram(iris$Petal.Length, breaks=10, type="count",
main="Histogram")
```



Density Plot

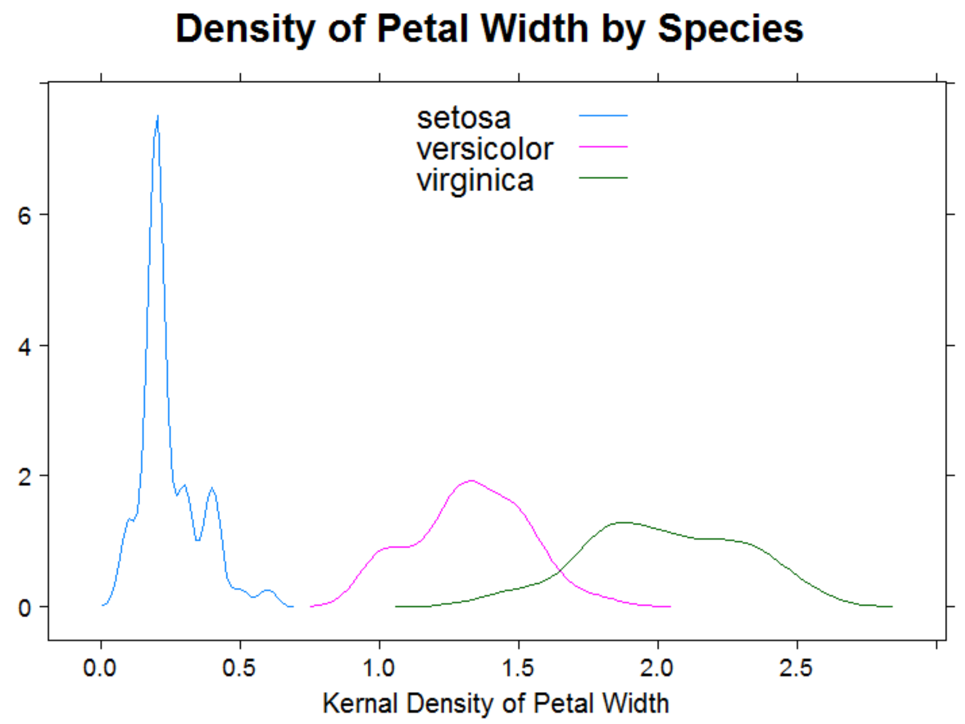
A density plot can be thought of as histogram with an infinite number of bins. It is a variation on the histogram and estimates densities for data. The idea of the total area under the curve is useful when dealing with density plots. The total area under the curve of a density plot should equal 1.

Further Exercises:
Create a density plot of mpg using mtcars dataset.
Change the cyl to a factor type. Hint: use as.factor
Create a density plot of mpg segmented by cylinders. Hint: densityplot(~mpg|cly)

Multiple Density Plot

This is used when we would like to overlay multiple graphs. Use a multiple density plot to create a graph that shows the density plots for all three species on one graph.

```
densityplot(~ Petal.Width, data=iris, groups=Species,
plot.points=F, xlab=list(label="Kernel Density of Petal
Width", fontsize=20), ylab="", main=list(label="Density of
Petal Width by Species", fontsize=24),
auto.key=list(corner=c(0,0), x=0.4, y=0.8, cex=2),
scales=list(cex=1.5))
```



Questions:

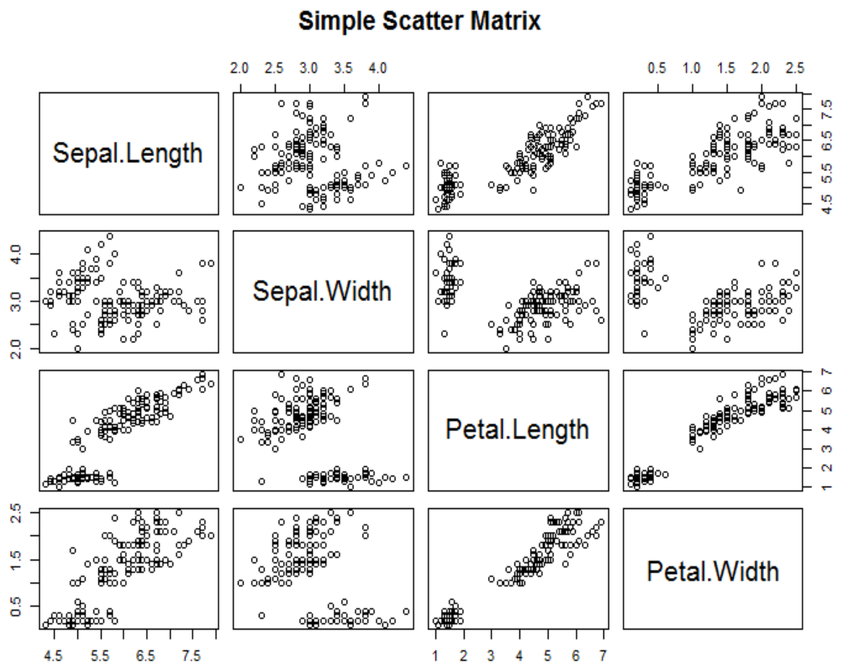
1. Is the area under the curve equal to 1 for each individual species or the whole graph?
2. Which has a higher probability of having a petal width of 1.2? Versicolor or Virginica? Why?

R Core - Scatterplot matrix

Scatterplot matrix is a scatterplot on all the numeric variable of a dataset.

Exercise: Using the iris dataset, create a scatterplot matrix.

```
pairs(~ Sepal.Length + Sepal.Width + Petal.Length +
Petal.Width, data=iris, main="Simple Scatter Matrix")
```

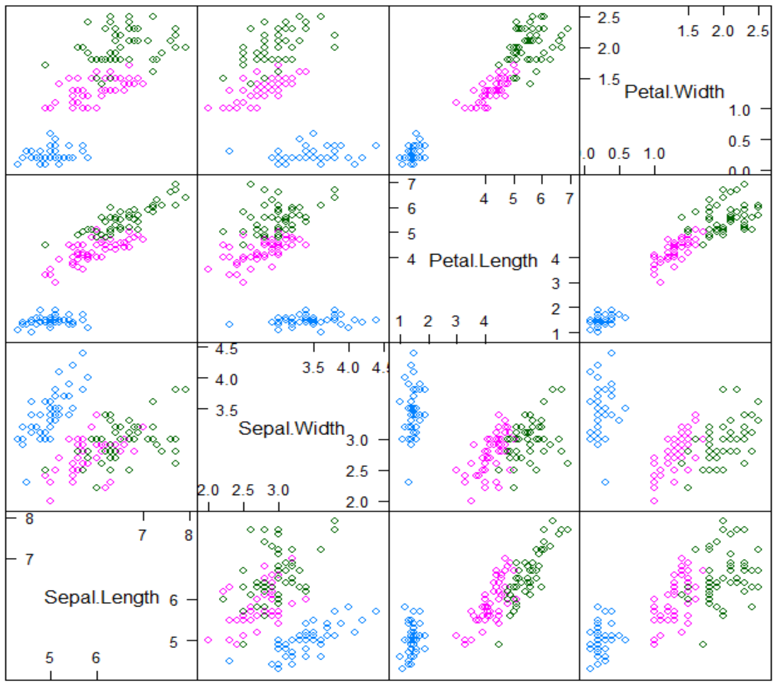


Lattice Scatterplot matrix

Lattice Scatterplot matrix is a better scatterplot.

Exercise: Using the iris dataset, create a scatterplot matrix with lattice package. Label the graph using the graphical parameters.

```
#Lattice Graphics
splom(iris[1:4], groups=iris$Species, panel=panel.superpose,
    key=list(title="Three Flower Types", columns=3,
points=list(pch=super.sym$pch[1:3],  col=super.sym$col[1:3]),
text=list(c("Setosa","Versicolor","Verginica"))))
```



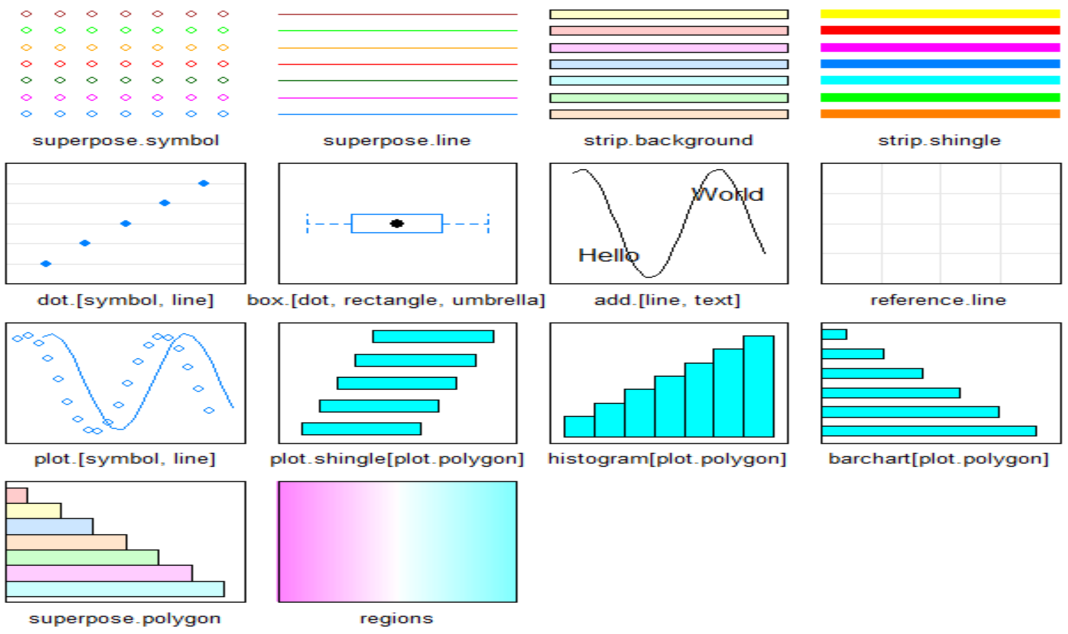
Scatter Plot Matrix

Graphical Settings

You can modify global setting when generating your reports using:

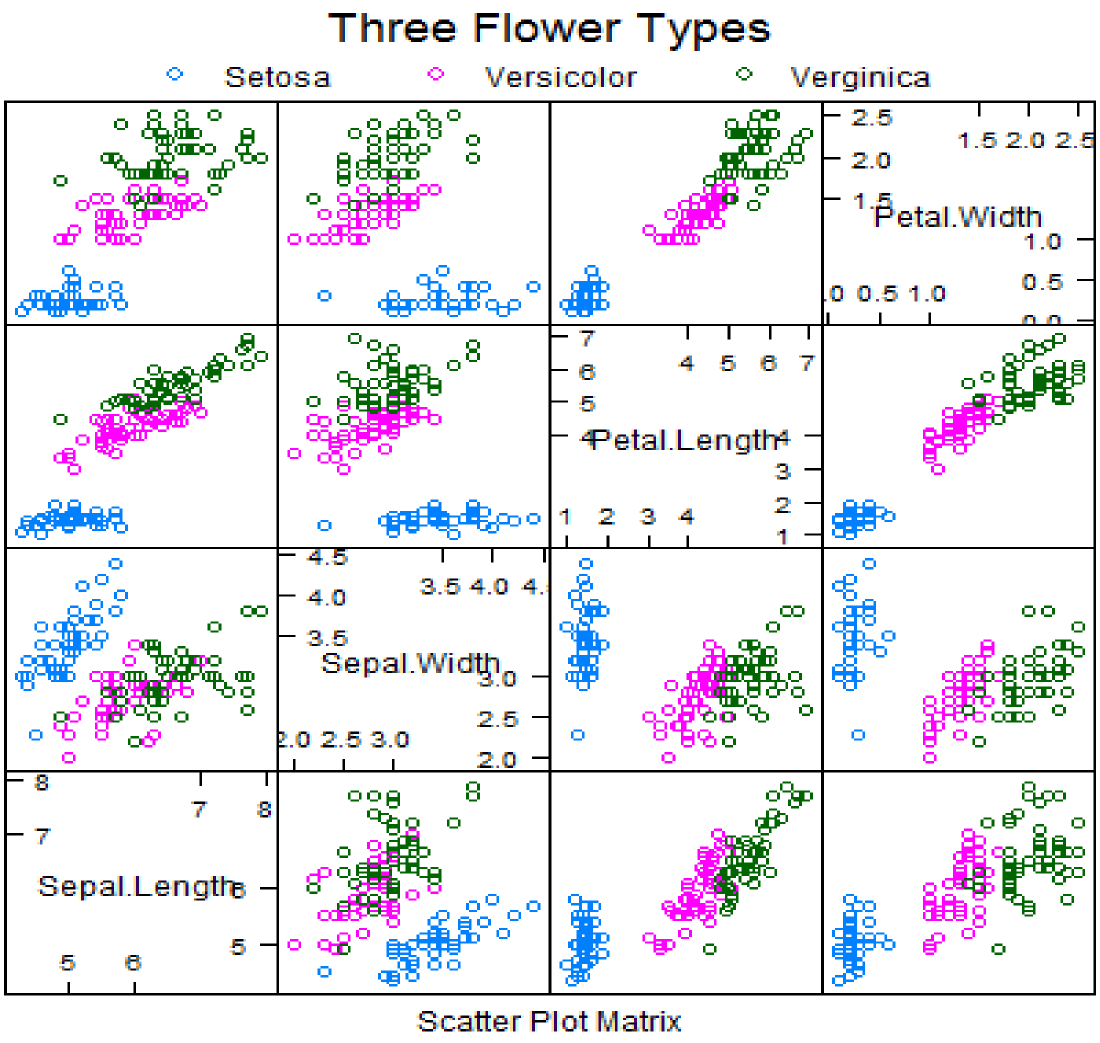
```
my.theme = trellis.par.get()
names(my.theme)
```

```
show.settings()
my.theme$fontsize$text=20
```



Scatterplot matrix with segmentation

```
# Getting settings for legend
super.sym <- trellis.par.get("superpose.symbol")
splom(iris[1:4],
groups=iris$Species,
panel=panel.superpose,
key=list(title="Three Flower Types",
columns=3,
points=list(pch=super.sym$pch[1:3],  col=super.sym$col[1:3]),
text=list(c("Setosa","Versicolor","Verginica"))))
```



Enhanced Scatterplot matrix

Enhance your graphs using GGally package.

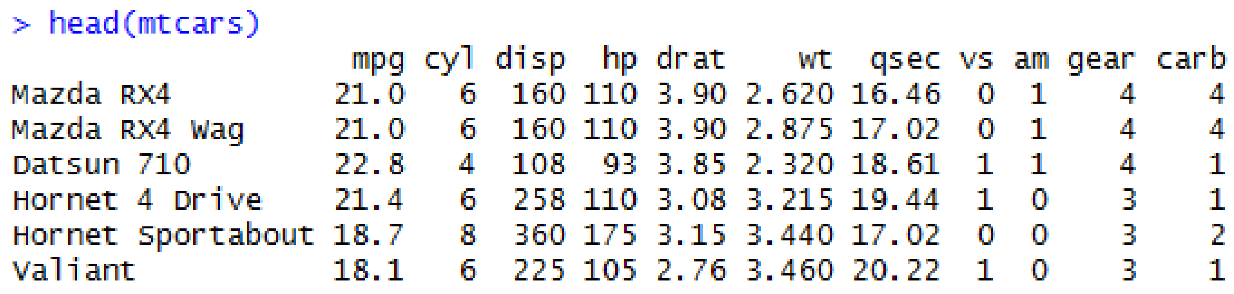
In-class Exercise 2

Using the “mtcars” dataset, predict mpg based on other columns.
Create at least 2 different plots illustrating useful relationships in data and summarize your findings. Goal: Predict mpg based on other columns.

Mtcars Dataset
The mtcars dataset was extracted from the 1947 Motor Trend US magazine and comprises fuel consumption and ten aspects of automobile design and performance for 32 automobiles. In other words the dimensions of this dataset are 32 by 10. This is common dataset used to perform regression analysis on.

Look at the dataset by using data(mtcars).

Exercise: Look at the first 6 rows of this dataset



```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Features of this dataset are:

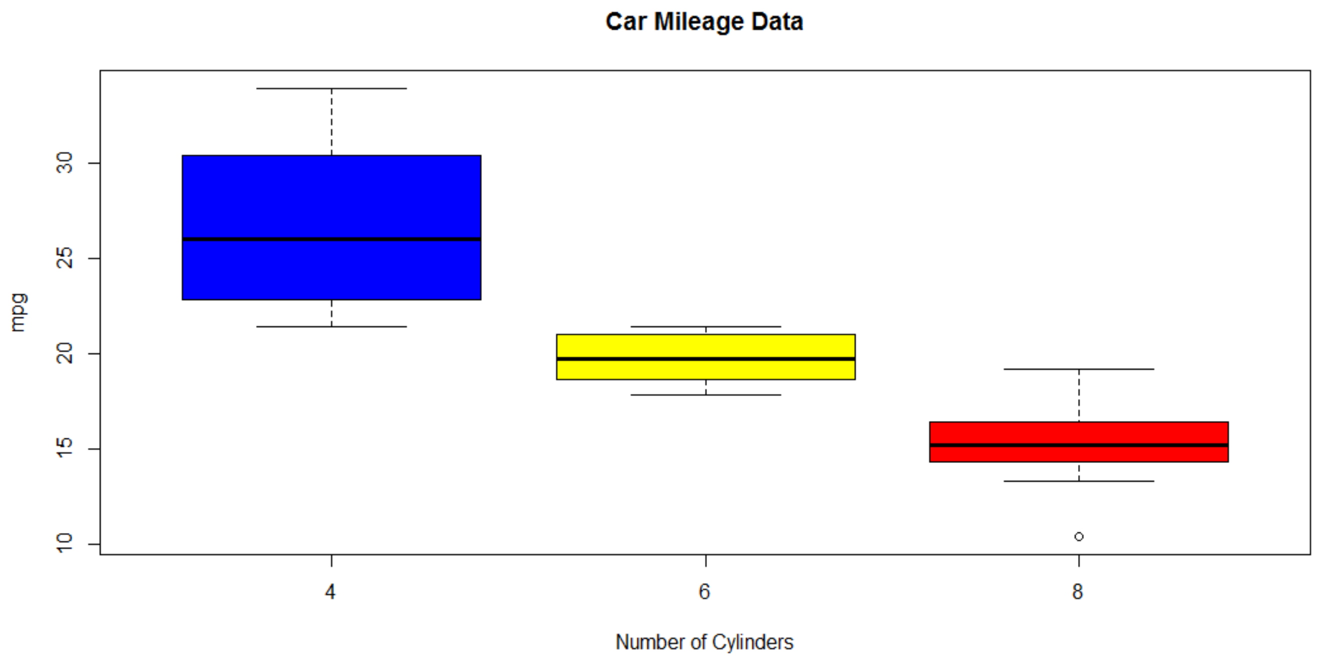
mpg = miles per gallon
cyl = number of cylinders
disp = Displacement
hp = groos horsepower
drat = rear axle ratio
wt = weight

qsec = ¼ mile time
vs = V/S
am = transmission (0 =automatic, 1=manual)
gear = number of forward gears
carb = number of carburetors

Exercise:
Create a boxplot of mpg versus cyl.
Label the graph as “Car Mileage Data” using **main** function
Label the x axis as “Number of Cylinders” using **xlab** function
Label the y-axis as “mpg” using **ylab** function
Color the boxplots “blue”, “yellow” and “red” using **col** function.

```
boxplot(mtcars$mpg ~ mtcars$cyl, data= mtcars, main="Car Mileage Data",
xlab="Number of Cylinders" ,ylab= "mpg", col = c("blue", "yellow",
"red"))
```

Plot



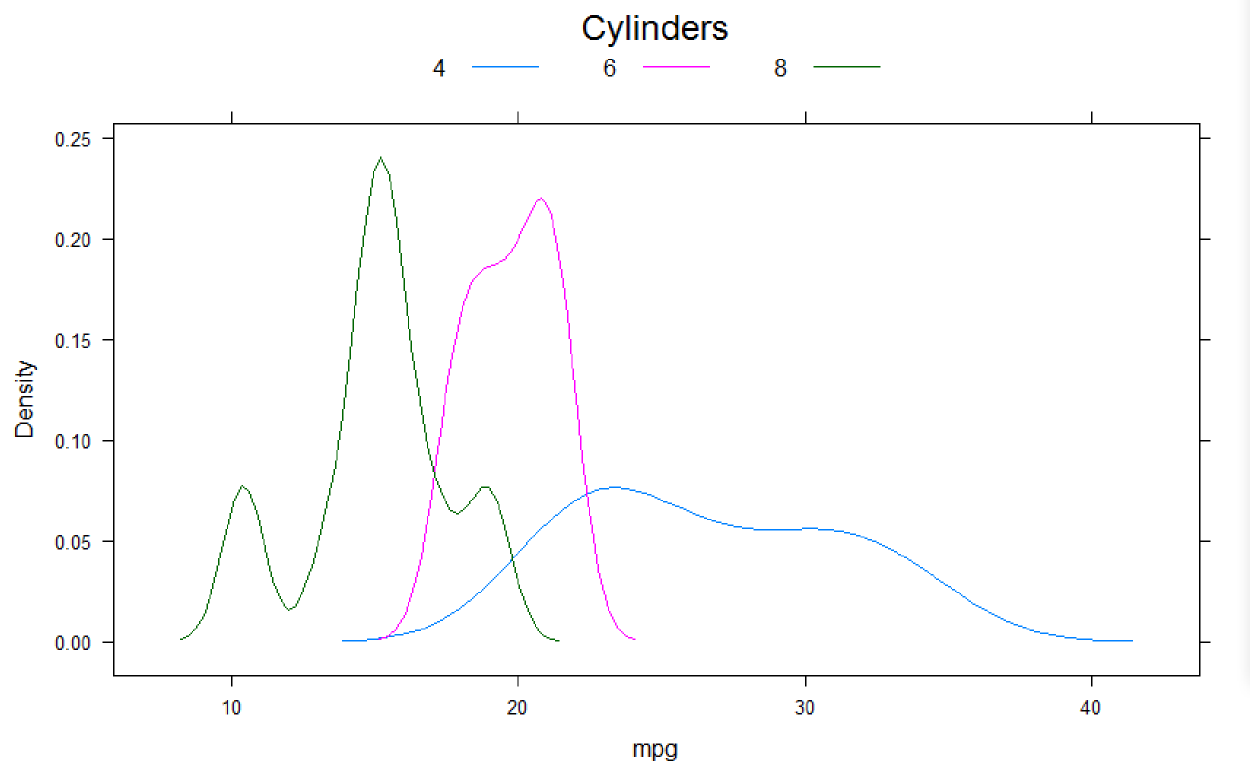
Further Exercises:
Do you think cars with manual transmission have more efficient fuel management compared to automatic transmission? How can you test this theory? Use your knowledge of the plot, xyplot, boxplot, density plot, histograms, density plot, scatterplot matrix functions (with and without segmentation) to create graphs for this dataset.

Sample Solutions for Exercise 2

Using the “mtcars” dataset, predict mpg based on other columns.
Create at least 2 different plots illustrating useful relationships in data and summarize your findings.

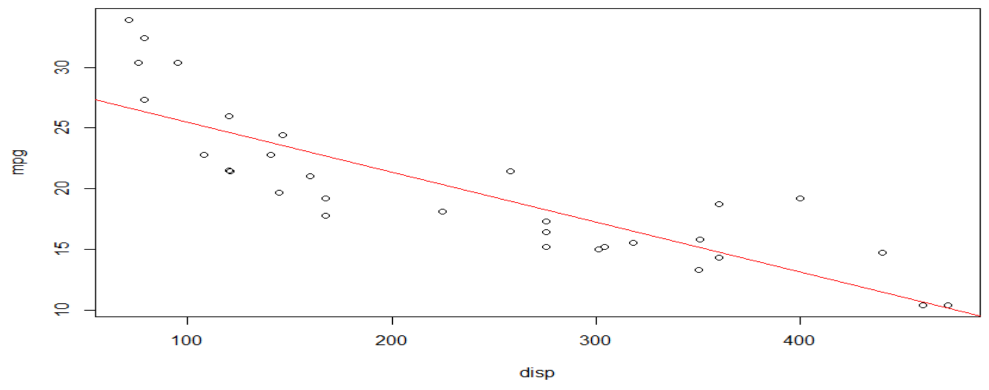
Density plot of mpg segmented by cylinders

```
#Lattice Graphics
densityplot(~ mpg,
  data=mtcars,
  groups=cyl,
  plot.points=F,
auto.key=list(columns=3, title="Cylinders"))
```



Plot of mpg versus disp using plot ( ) with a regression line

```
# Core Graphics
plot(mpg ~ disp, data=mtcars)
abline(lm(mpg ~ disp, data=mtcars), col="red")
```



Find correlation between variables

```
cor(mtcars$mpg,mtcars$disp)
[1] -0.8475514
```

GGplot2 Graphics

The ggplots2 package is the most recommended package for creating graphs. ggplot2 is a plotting system for R, based on grammar of graphics, which tries to combine lattice and core R graphs and give you the best of both worlds. It takes care of the hassle involved in adding arguments for creating legends, etc. It is an extremely powerful tool that helps with creating multi-layered graphs.

For more resources, refer to http://ggplot2.org/

In-Class Exercise 3

Exercise 3
Load the "titanic" dataset.
1. Create 2 boxplots of Age

- **Segmented by Gender**
- **Segmented by Survived**

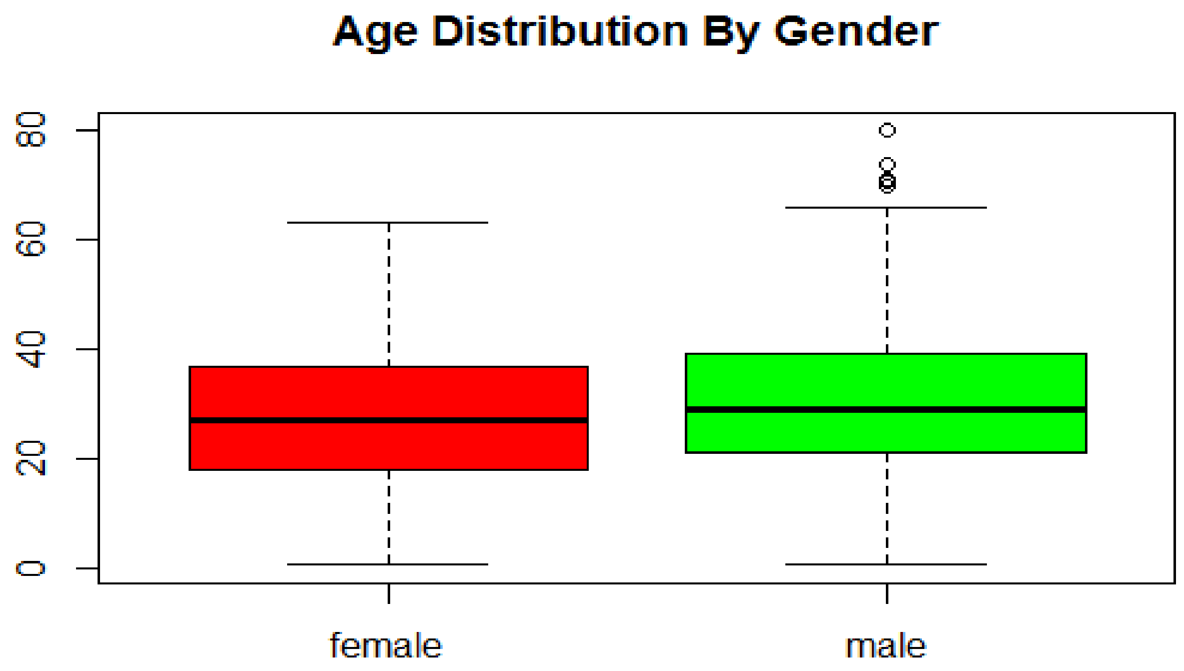
2. Create a density plot of Age

- **na.omit() may be useful**

Sample solutions for Exercise 3

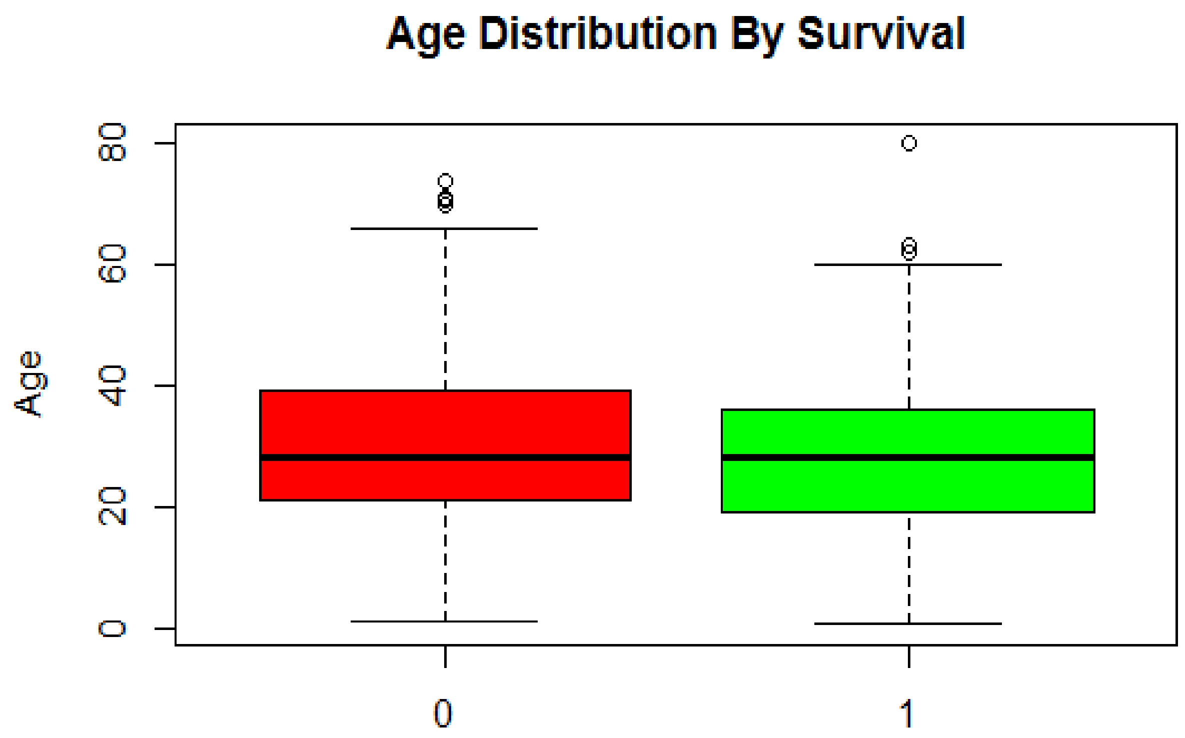
Boxplot for Age distributed by Gender

```
# Core Graphics
boxplot(Age ~ Sex, data=titanic, col=c("red","green"))
```



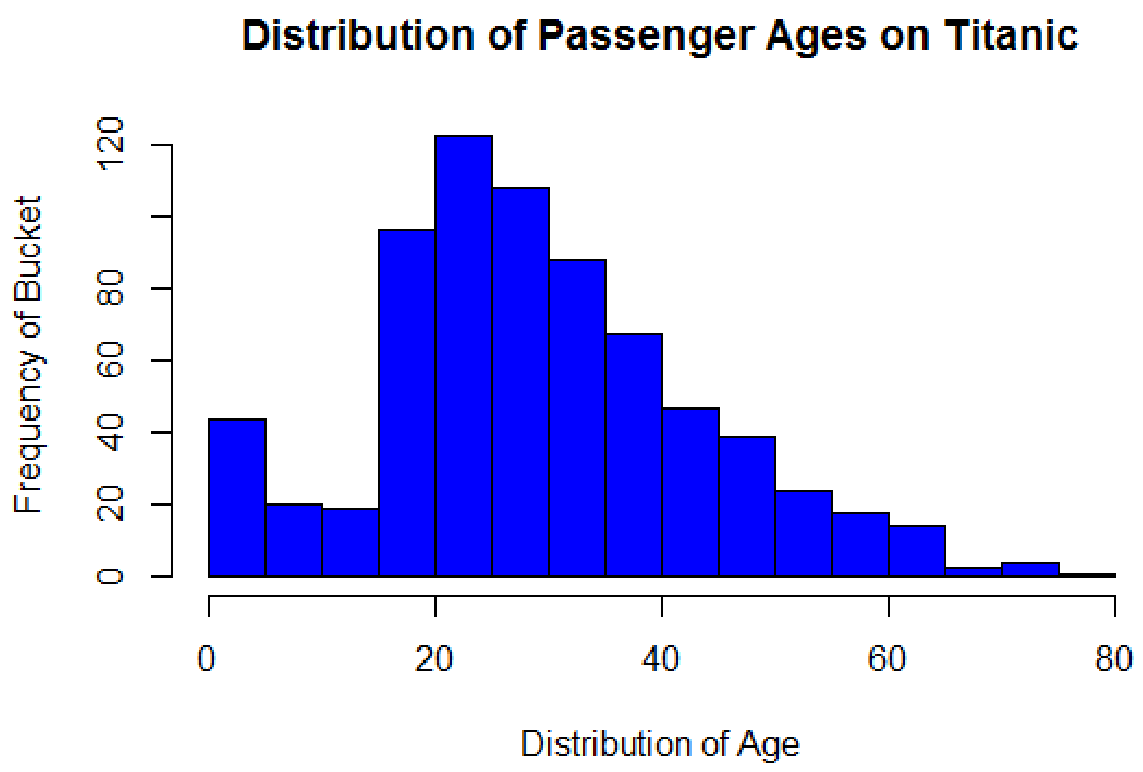
Boxplot for Age distributed by Survival

```
# Core Graphics
boxplot(Age ~ Survived, data=titanic, col=c("red","green"))
```



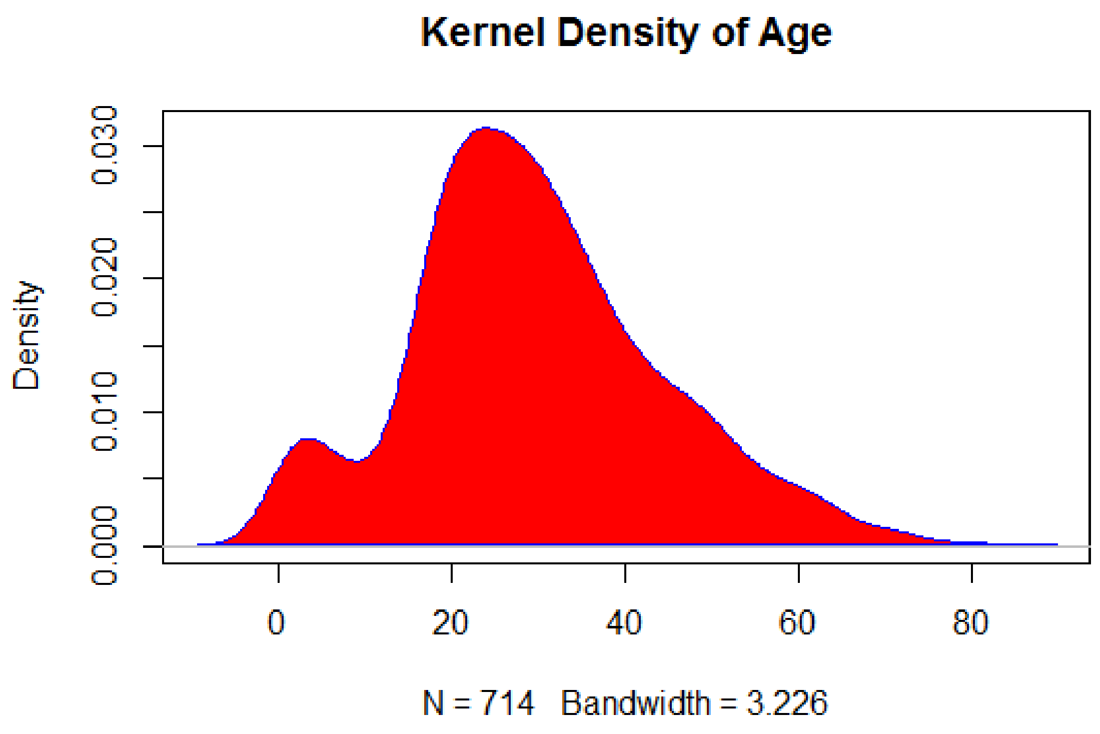
Histogram of Distribution of Passenger Ages

```
hist(titanic$Age,breaks=12,col="blue")
```

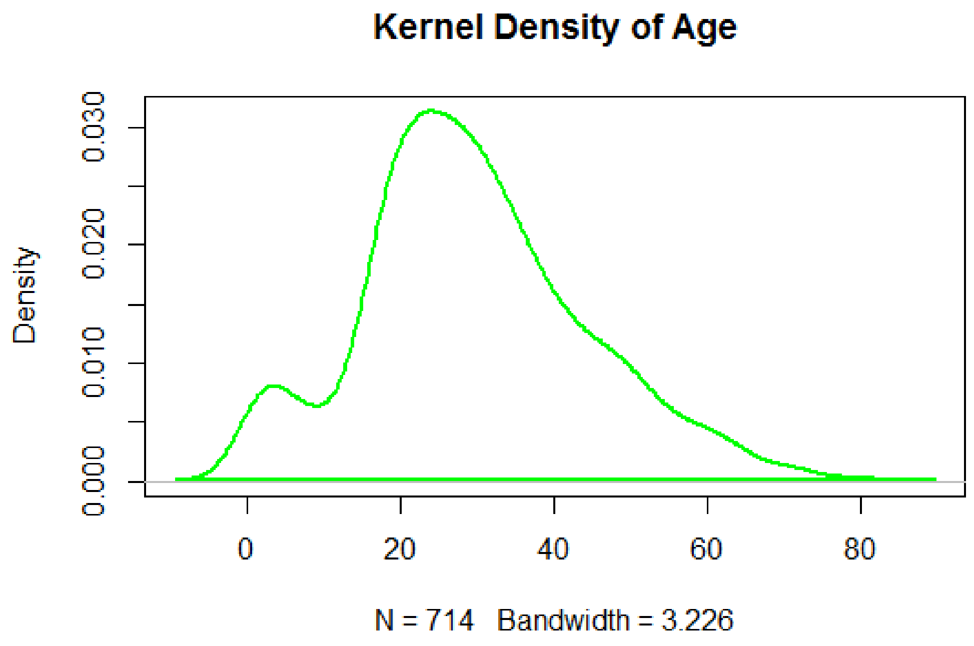


Densityplot of Distribution of Passenger Ages

```
density(titanic$Age)    #NAs prevent this
d <- density(na.omit(titanic$Age))
plot(d)
polygon(d, col="red")
```



```
density(titanic$Age)    #NAs prevent this
d <- density(na.omit(titanic$Age))
plot(d)
polygon(d,border="green")
```


Kernel Density of Age
Density
N = 714   Bandwidth = 3.226