DATA VISUALIZATION using R

Sachin Kumar Manjhi

Agenda

- R Graphics
- Plots in Base Package
    - Scatter Plot
    - Histogram
    - Box Plot
    - Pie Chart
- Parameters
- Low-level Functions
- Lattice Graphics
- ggPlot2
- Case1 – Diamonds for ggPlot2
- Case 2 – Credit Default for Data Visualization leading to Predictive Modeling
- Other Graphics Packages
- References

R Graphics

One of the main reasons data scientist turn to R is for its strong graphic capabilities.

```
> demo(graphics)
```

R Graphics

The graphics package is part of the standard distribution and contains many useful function for creating a variety of graphic displays.

There are three basic plotting functions in R: high-level functions, low-level functions, and the layout command par.

- High-level graphics functions, like plot() creates a complete plot and allow to add more afterwards. Example: plot(), hist(), boxplot() etc.
- Lower-level functions add elements to an existing plot created by high level command. Example: like points(), lines() , text() etc.
- The par command controls the layout of the graphics device

Graphics are an incredibly deep and broad area, and a session like this cannot explore the depths.

Instead, the goal here will be to demonstrate some useful high-level functions and show how to change some of the key options that affect the results.

Scatter Plot

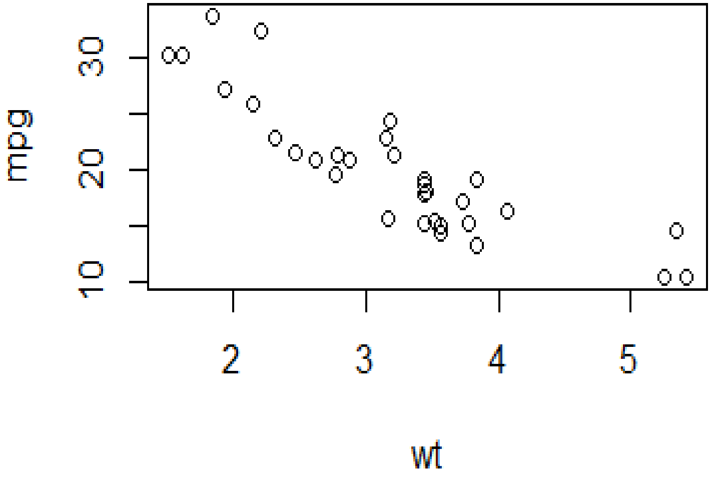
The most used plotting function in R programming is the plot() function. It is a generic function, meaning, it has many methods which are called according to the type of object passed to plot().

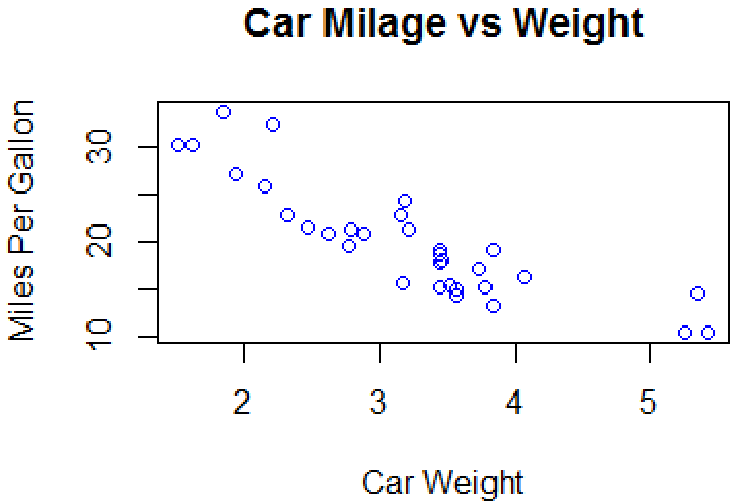
```
# plot(x,y) creates a basic
# scatterplot.

attach(mtcars)
plot(wt, mpg)
```

```
#Adding axes, title and color

attach(mtcars)
plot(wt, mpg,
   main="Car Milage vs Weight",
   xlab="Car Weight",
   ylab="Miles Per Gallon",
   col="blue")
```

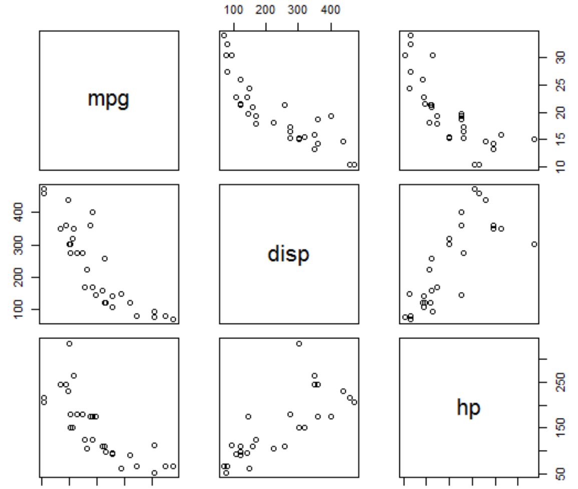




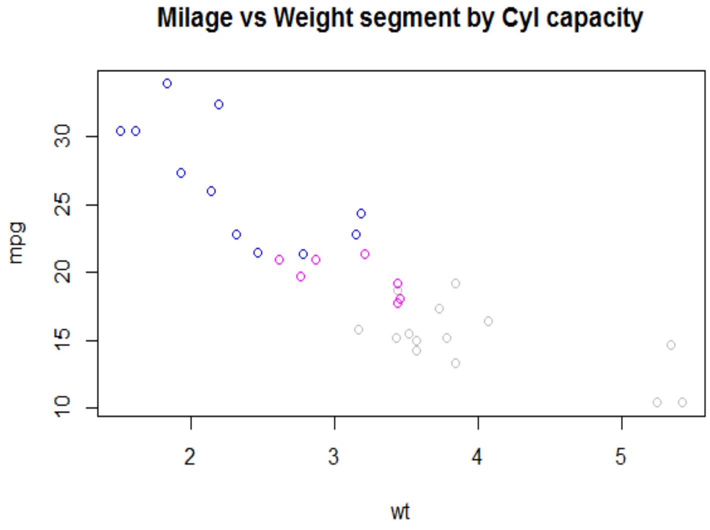
Scatter Plot

- If a dataframe is passed to plot(), the function assumes we want to create a scatterplot for each combination of numeric column vectors present.
- Providing a vector of numeric to col options will print a different color for each value of the variable.

```
cars <- subset(mtcars,,
        select= c(mpg,disp,hp))
plot(cars)
```



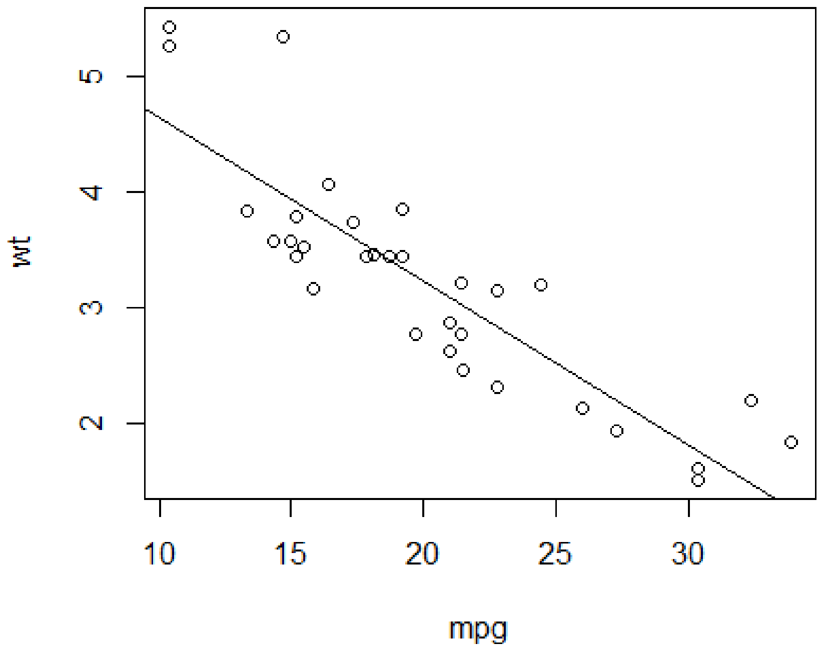
```
plot(wt, mpg,
     main="Milage vs Weight segment by Cyl capacity",
     col = as.integer(cyl))
```



Scatter Plot

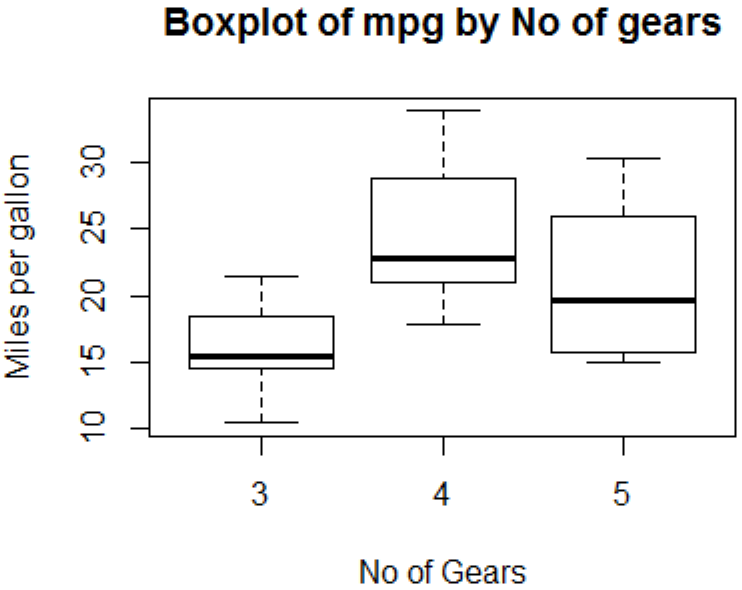
Plotting a regression line of a scatter plot.

```
# create a model object and then plot
the model object
lm <- lm(wt ~ mpg)
plot(wt ~ mpg)
abline(lm)
```



Providing a factor and numeric vector will result in a box plot.

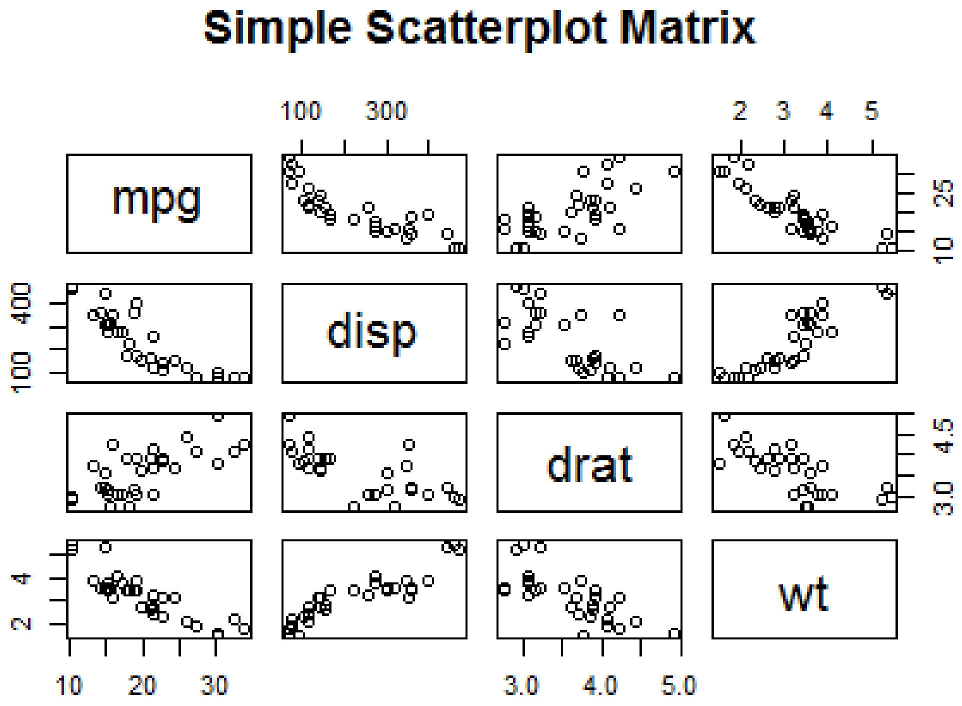
```
plot(as.factor(gear), mpg,
     main = "Boxplot of mpg by
No of gears",
     xlab = "No of Gears",
     ylab = "Miles per gallon")
```



Scatter Plot

Apart from Plot function, there are many other functions to create scatter plot.

```
#Scatterplot Matrix with multiple variables
pairs(~mpg+disp+drat+wt,data=mtcars,
   main="Simple Scatterplot Matrix")
```



More examples: http://www.statmethods.net/graphs/scatterplot.html

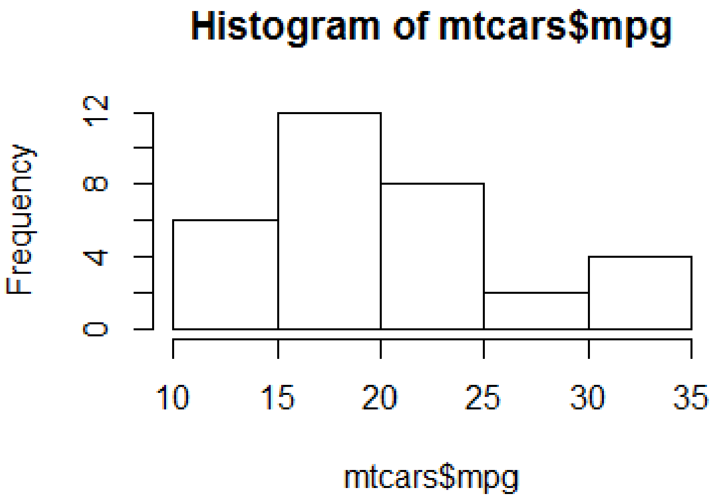
Histograms

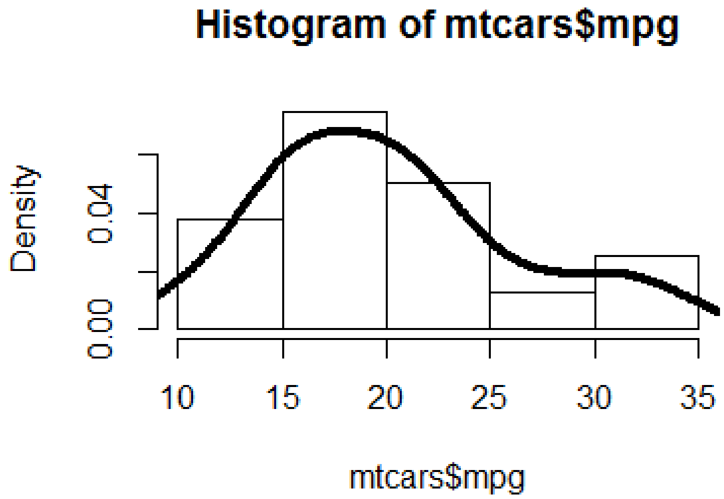
We can create histograms with the function **hist(*x*)** where *x* is a numeric vector of values to be plotted.

- The option **freq=FALSE** plots probability densities instead of frequencies.
- The option **breaks=** controls the number of bins.

```
#Histogram for a continuous var.
hist(mtcars$mpg)
```

```
# Histogram with empirical pdf
lines(density(AirPassengers), lwd=4)
```



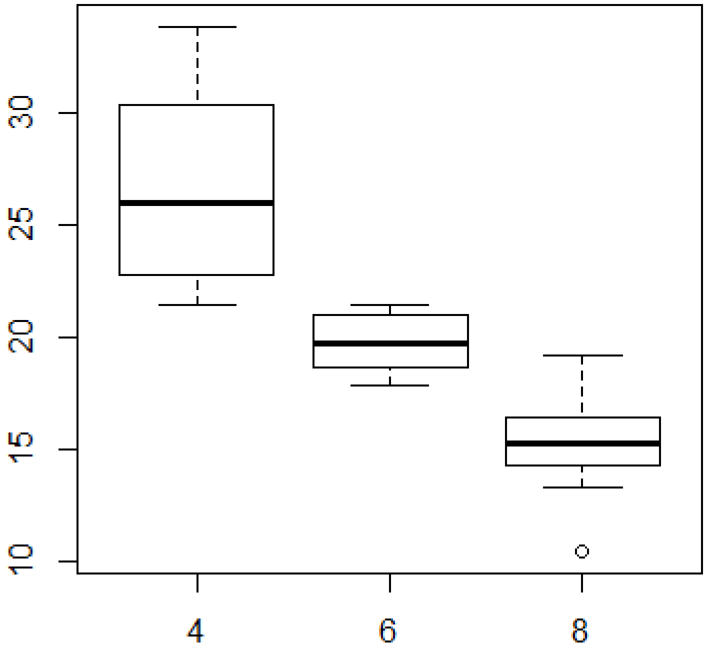


Boxplot

Boxplots can be created for individual variables or for variables by group.

- The format is **boxplot(*x*, data=)**, where *x* is a formula and **data=** denotes the data frame providing the data. Use with() to wrap the data frame.

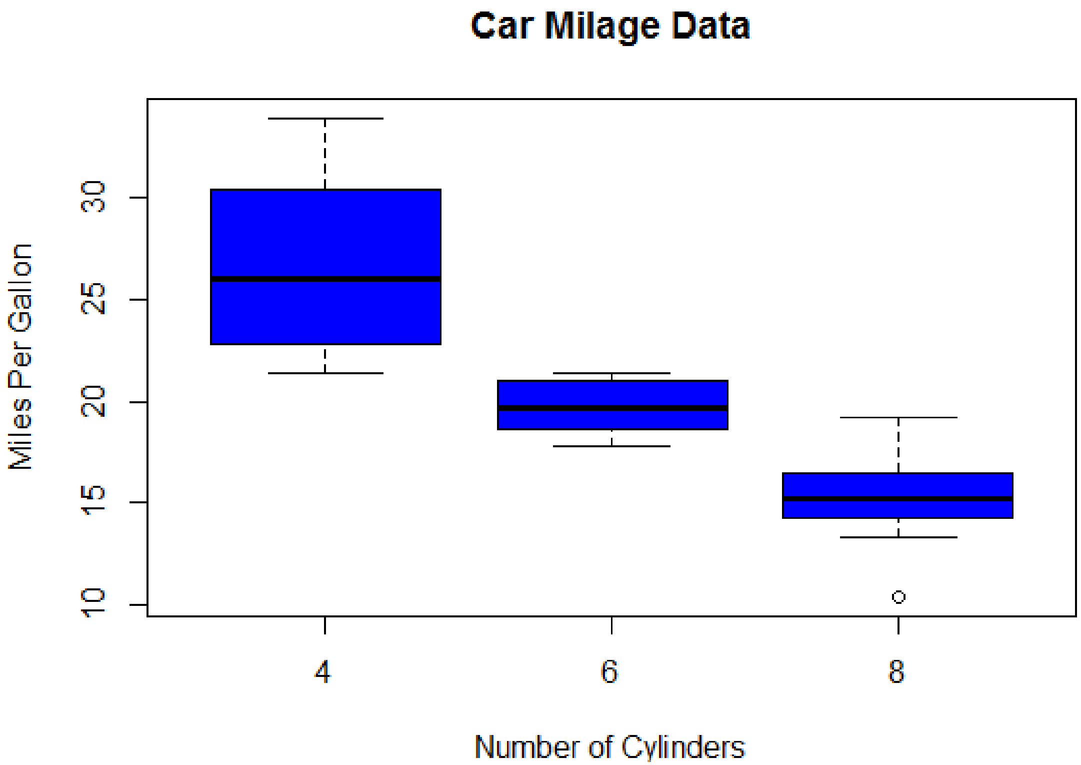
```
boxplot(mpg~cyl,data=mtcars)
```



Boxplot 2

Providing colors and Title to the boxplot.

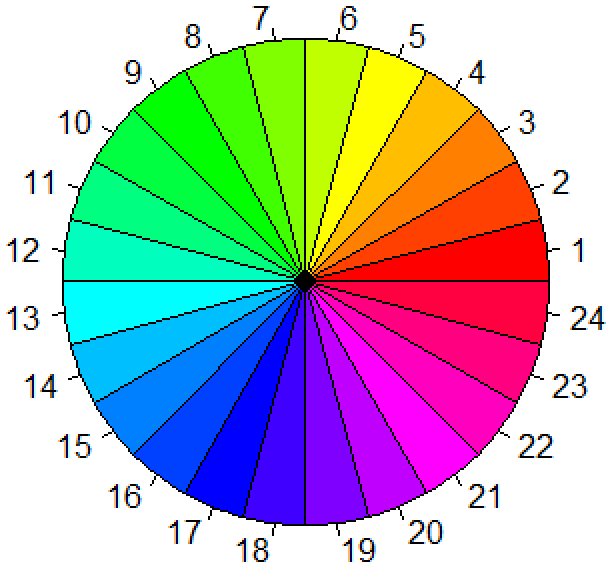
```
boxplot(mpg~cyl,data=mtcars,
           main="Car Milage Data",
           xlab="Number of Cylinders",
           ylab="Miles Per Gallon“,
           col=“blue”)
```



Pie Chart

Creating a pie chart

```
pie(rep(1,24), col = rainbow(24), radius = 0.9)
```



Lattice Graphics

The lattice package contains a wholly distinct way of creating graphics. In some ways it's more powerful than the "traditional" graphics.

Lattice attempts to improve on base R graphics by providing better defaults and the ability to easily display multivariate relationships.

The following display types are available in lattice.

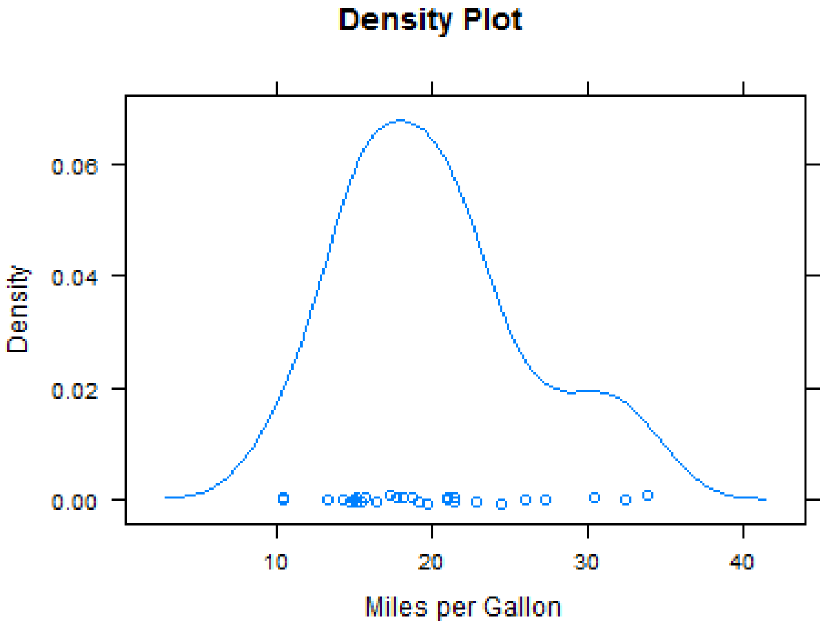
Function	Default Display
histogram()	Histogram
densityplot()	Kernel Density Plot
qqmath()	Theoretical Quantile Plot
qq()	Two-sample Quantile Plot
stripplot()	Stripchart (Comparative 1-D Scatterplots)
bwplot()	Comparative Box-and-Whisker Plots
dotplot()	Cleveland Dot Plot
barchart()	Bar Plot
xyplot()	Scatterplot
splom()	Scatterplot Matrix
contourplot()	Contour Plot of Surfaces
levelplot()	False Colour Level Plot of Surfaces
wireframe()	Three-dimensional Perspective Plot of Surfaces
cloud()	Three-dimensional Scatterplot
parallel()	Parallel Coordinates Plot

Lattice Example (1 of 3)

```
library(lattice)
attach(mtcars)

# kernel density plot

densityplot(~mpg, main="Density Plot", xlab="Miles per Gallon")
```

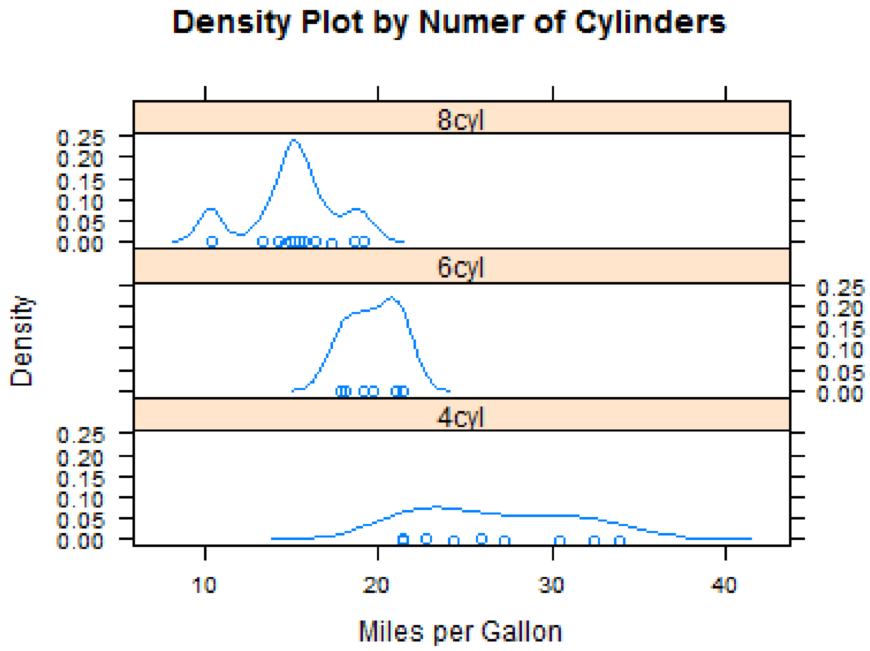


Lattice Example (2 of 3)

```
# kernel density plots by factor level

# create factors with value labels
cyl.f <-factor(cyl,levels=c(4,6,8),
     labels=c("4cyl","6cyl","8cyl"))

densityplot(~mpg|cyl.f,
          main="Density Plot by Numer of Cylinders",
          xlab="Miles per Gallon",
          layout=c(1,3))
```

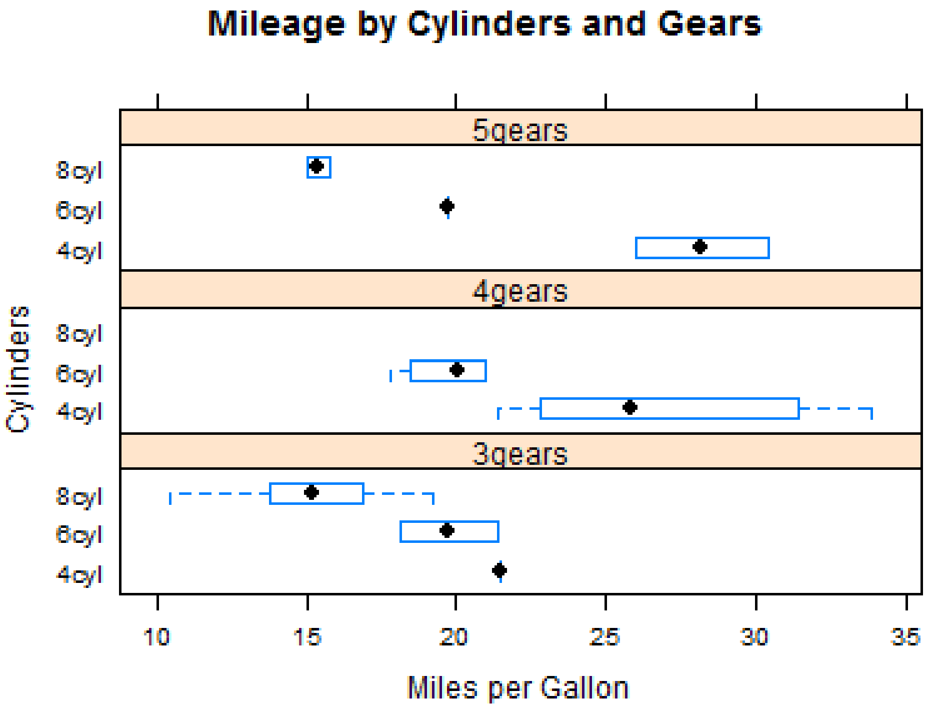


Lattice Example (3 of 3)

```
# boxplots for each combination of two factors

gear.f<-factor(gear,levels=c(3,4,5),
     labels=c("3gears","4gears","5gears"))


bwplot(cyl.f~mpg|gear.f,
       ylab="Cylinders", xlab="Miles per Gallon",
       main="Mileage by Cylinders and Gears",
       layout=(c(1,3)))
```



Parameters

Parameters help customize many features of your graphs (fonts, colors, axes, titles) through graphic options:

- The graphics environment is determined by values in the par() object.
- Parameters can be set by specifying them as arguments to par in tag = value form, or by passing them as a list of tagged values.
- You can see what they are with par()...and change them withpar(name=value). For example, you can get a 2X2 matrix of plots with par(mfrow = c(2,2))
- See help(par)

Lower-level Functions: Text and Data

TEXT FUNCTIONS:

- text(x,y,"text") to add text
- title(main="text", sub="text" for titles and subtitles
- mtext("text", side) to write in the margins
- axis(side) to add an axis

Points & Lines :

- points(x,y) to add points to an existing plot
- lines(x0=,y0=,x1=,y1=) to add lines between points
- abline() to add a single line, or an href or vref, quickly
- arrows() to add arrows to the plot

GGPlot2

ggplot2 is a data visualization package for R. Created by Hadley Wickham in 2005, ggplot2 is an implementation of Leland Wilkinson's ***Grammar of Graphics***

Advantages of ggplot2:

- consistent underlying “grammar of graphics”
- plot specification at a high level of abstraction
- very flexible
- theme system for polishing plot appearance
- mature and complete graphics system

Disadvantages:

- No support for 3-dimensional graphics (see the rgl package)
- Can’t create Graph-theory type graphs (nodes/edges layout; see the igraph package)
- No Interactive graphics (see the ggvis package)

Compared to base graphics, ggplot2

- is more verbose for simple / canned graphics
- is less verbose for complex / custom graphics
- does not have methods (data should always be in a data.frame)
- uses a different system for adding plot elements

ggPlot2 - Overview

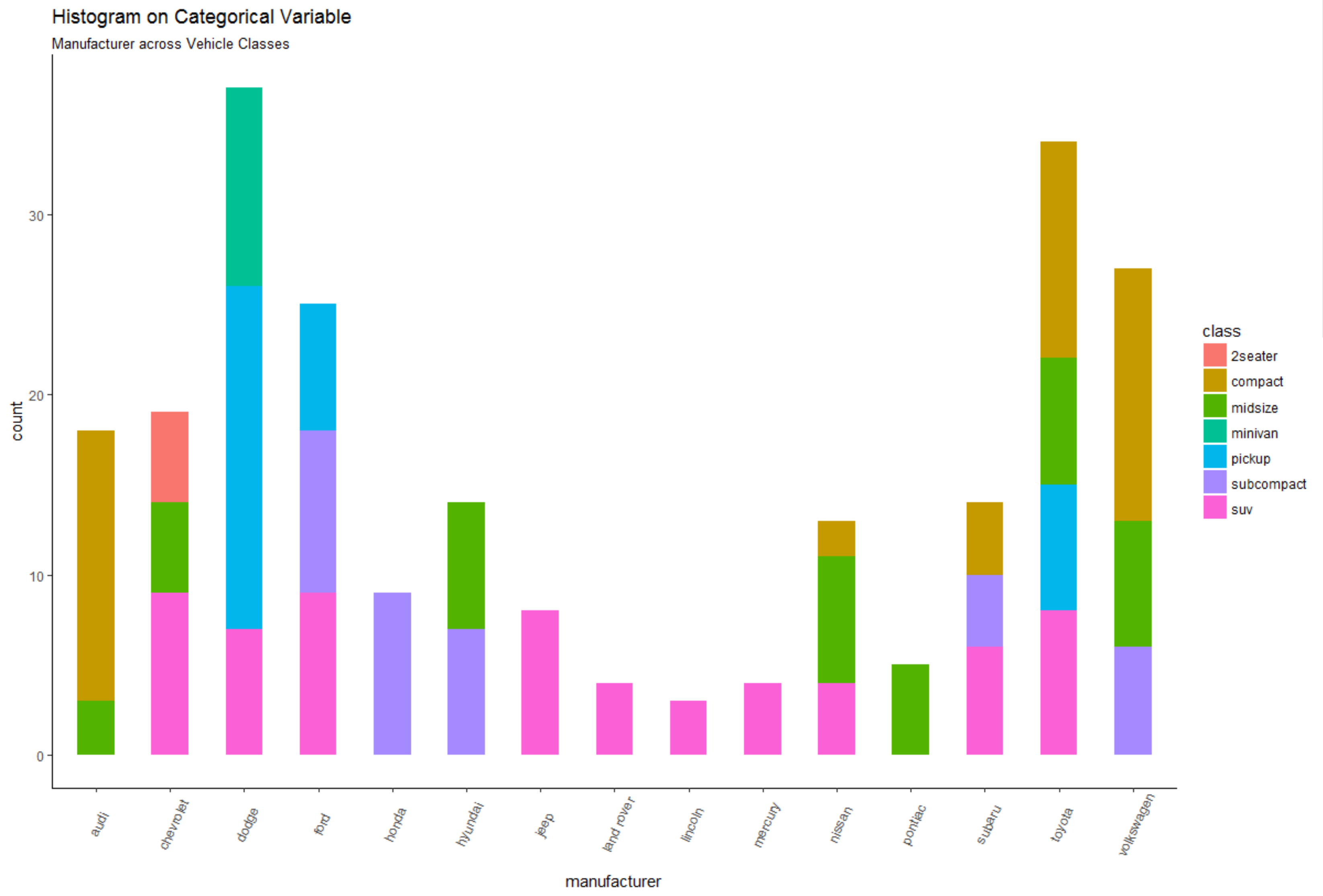
```
# Code Template
   ggplot( data = <DATA>)         +
   <GEOM_FUNCTION> (
       mapping = aes(<MAPPINGS>,
       stat = <STAT>,
      position = <POSITION>
   )                              +
   <COORDINATE_FUNCTION>          +
   <FACET_FUNCTION>
```

- **Geoms:** these are the geometric objects (bars, points, lines). Over 30 geom objects are available in ggplot2.
- **Aesthetics:** these are the roles that the variables play in each graph. A variable may control where points appear, the color or shape of a point, the height of a bar and so on.
- **Statistics:** these are the functions like count or linear regression line.
- **Position:** specify where to place the chart elements (bars to be stacked vs place on side.)
- **Coordinate**: specify type of coordinate system or rotate the charts.
- **Facets**: repeat a chart by groups in your data. Faceting by gender would cause the graph to repeat for the two genders.

ggPlot2 - Example

```
library(ggplot2)
the

# H
g <
g +
  t
  l
```



Source: http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

Case Example

DIAMONDS

Case Example

CREDIT DEFAULT

Exercise 1

Dataset: **PlantGrowth**

Dataset contains results from an experiment to compare yields (as measured by dried weight of plants) obtained under a control and two different treatment conditions.

This is a standard dataset that comes with R.

- Weight– dried weight of plant
- Group– control and two different conditions

Questions:

1. Create a boxplot for the three different groups. Do you see a difference in the treatments from the graph?
2. Find the mean and standard deviation by each group. This may require you to use a “plyr” package for data frame manipulation.
3. We want to be sure that there is a difference in the dried weight per treatment. To investigate whether the treatments are different to the control group, run an ANOVA test on the data. Report your findings.

Exercise 2

Dataset: HSAUR2: Usmelanoma

The dataset consists of USA mortality rates for white males due to malignant melanoma 1950-1969.
A data frame with 48 observations on the following 5 variables.

- Mortality – number of white males died due to malignant melanoma 1950-1969 per one million inhabitants.
- Latitude – latitude of the geographic centre of the state.
- Longitude – longitude of the geographic centre of each state.
- Ocean – a binary variable indicating contiguity to an ocean at levels no or yes.

Questions:

1. How do the mortality rates compare for ocean and non-ocean states?
2. How are mortality rates affected by latitude and longitude?
3. Compare the distribution of Coastal States against Land Slides.
4. How mortality rates are related to the geographic location of a state as represented by the latitude and longitude of the centre of the state?

Other Popular Graphics Packages

- googleVis: Let's you use Google Chart tools to visualize data in R. Google Chart tools used to be called Gapminder, the graphing software Hans Rosling made famous in his TED talk.
    - Example: https://cran.r-project.org/web/packages/googleVis/vignettes/googleVis_examples.html

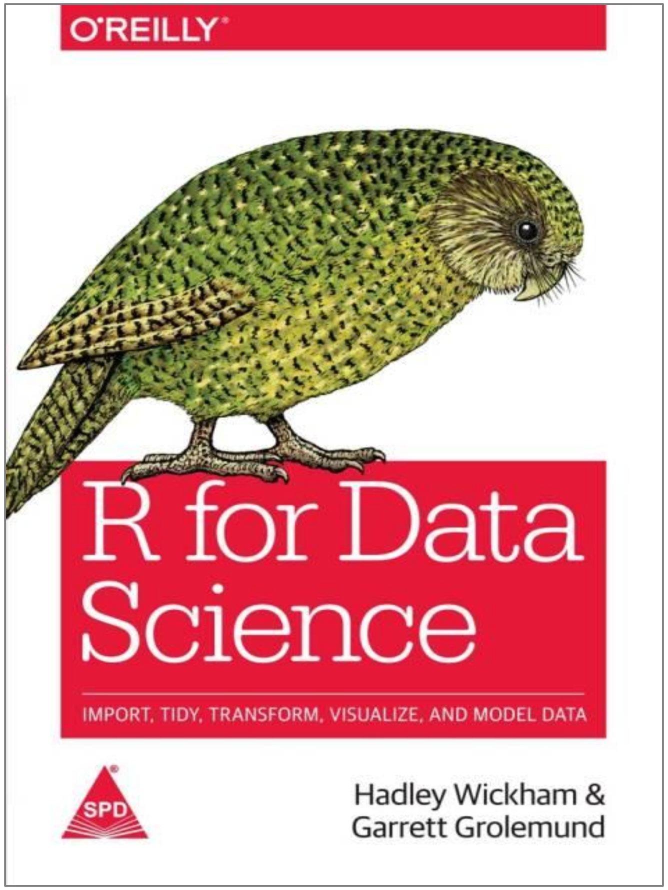
- Plotly: Plotly for R is an interactive, browser-based charting library built on the open source JavaScript graphing library plotly.js.

- ggvis: Interactive, web based graphics built with the grammar of graphics.
    - http://ggvis.rstudio.com/ggvis-basics.html

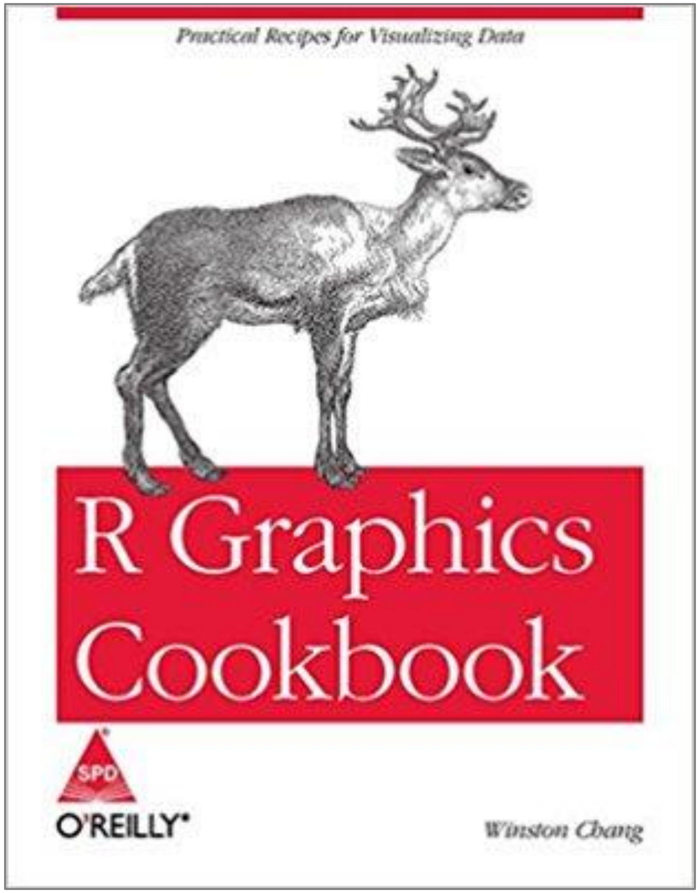
- rgl: Interactive 3D visualizations with R
    - http://rgl.neoscientists.org/about.shtml

- Shiny: *Shiny* is an *R* package that makes it easy to build interactive web apps straight from *R*
    - http://shiny.rstudio.com/articles/

Reference



http://r4ds.had.co.nz/



http://www.cookbook-r.com/Graphs/

THANK YOU!