STAT3622 Data Visualization (Lecture 1)

Introduction to Data Science

Dr. Yiwei Fan
The University of Hong Kong

18 January 2021

R Programming

R Programming

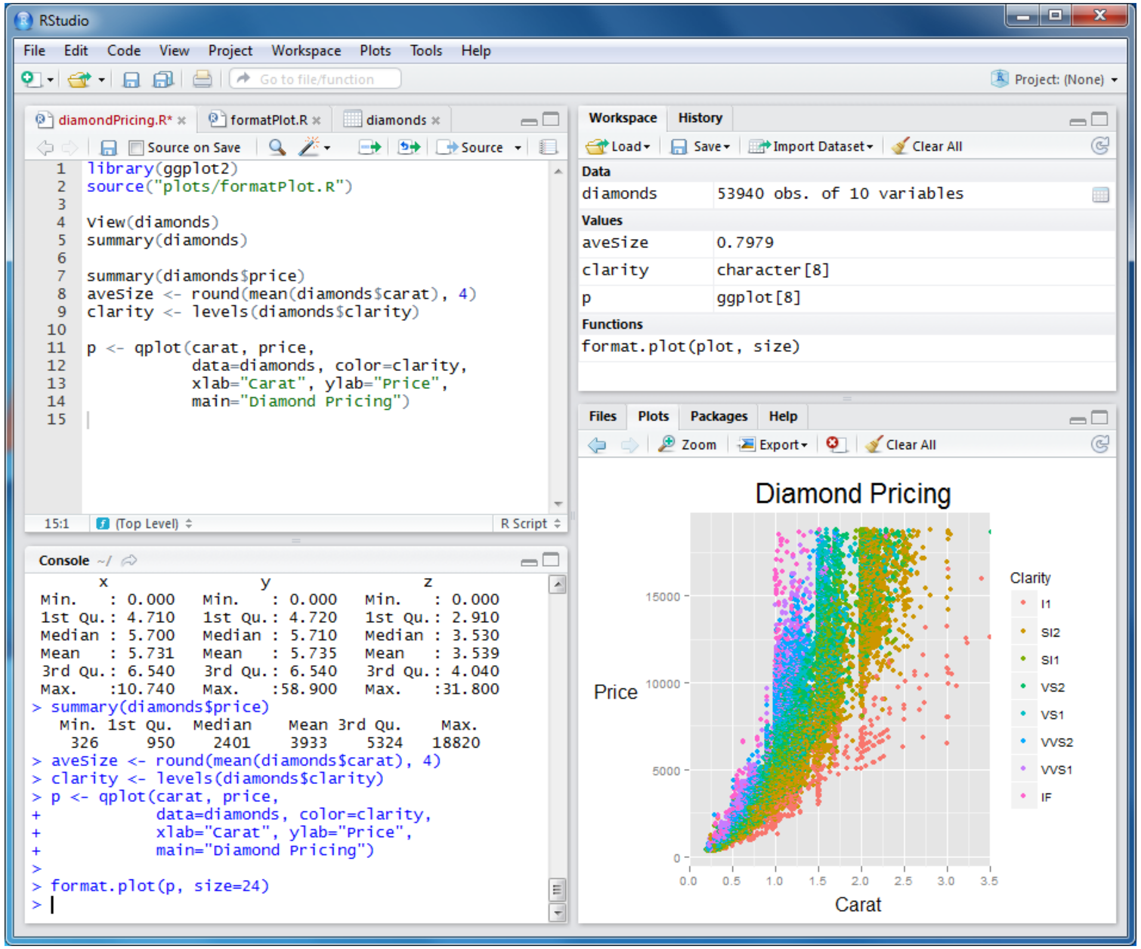
- R is a programming language and free software environment for statistical computing and graphics.
- The R language is widely used among statisticians and data miners for developing statistical software and data analysis.
- Although R has a command line interface, there are several third-party graphical user interfaces, such as RStudio, an integrated development environment.

RStudio and R Markdown

RStudio IDE

- RStudio is a popular IDE (Integrated Development Environment) for R programming
- It is a powerful editor for R coding and debugging.
- It is a powerful generator for HTML, PDF, dynamic documents and slide shows.
- RStudio can be run on both Desktop and Cloud.
- Check out more nice features of RStudio at its official website

RStudio IDE



```r
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")

```

Data

```
diamonds     53940 obs. of 10 variables
```

Values

```
aveSize      0.7979
clarity      character[8]
p            ggplot[8]
```

Functions

```
format.plot(plot, size)
```

```
       x                y                z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(p, size=24)
> |
```

R Markdown

- Click here to view a fantastic micro-video tutorial
- Browse here for a gallery of creative Rmarkdown works

R Markdown (Demonstrated)

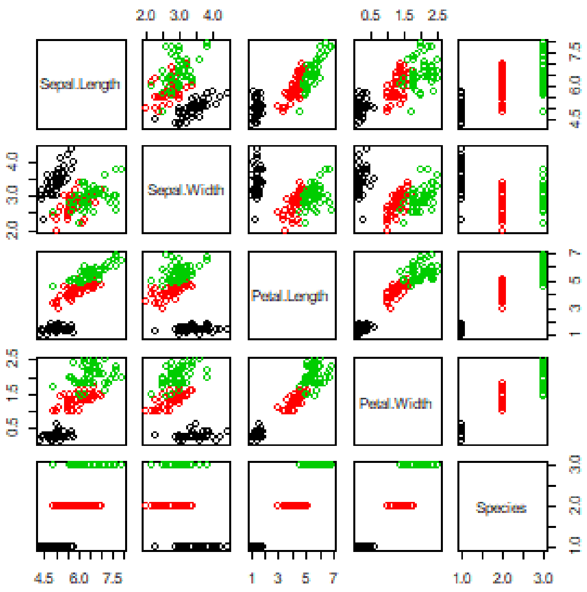
```
knitr::kable(head(iris), format = ’html’)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

- Dynamic documentation: report, table, graphics ...
- R packages by Yihui Xie: knitr, bookdown, xaringan, etc

R Markdown (Demonstrated)

```
plot(iris, col=iris$Species)
```



- Data-generated graphics that are reproducible

Exploratory Data Analysis

Exploratory Data Analysis

The EDA is a statistical approach to make sense of data by using a variety of techniques (mostly graphical). It may help us

- Assess assumption about variables distribution
- Identify relationship between variables
- Extract important variables
- Suggest use of appropriate models
- Detect problems of collected data (e.g. outliers, missing data, measurement errors)

Example: Anscombe Dataset

```
## Anscombe Dataset:
```

x1	y1	x2	y2	x3	y3	x4	y4
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.10	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.10	4	5.39	19	12.50
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

Source: Anscombe, F. J. (1973). Graphs in statistical analysis. *American Statistician*, **27**, 17-21.

Example: Anscombe Dataset (Descriptive)

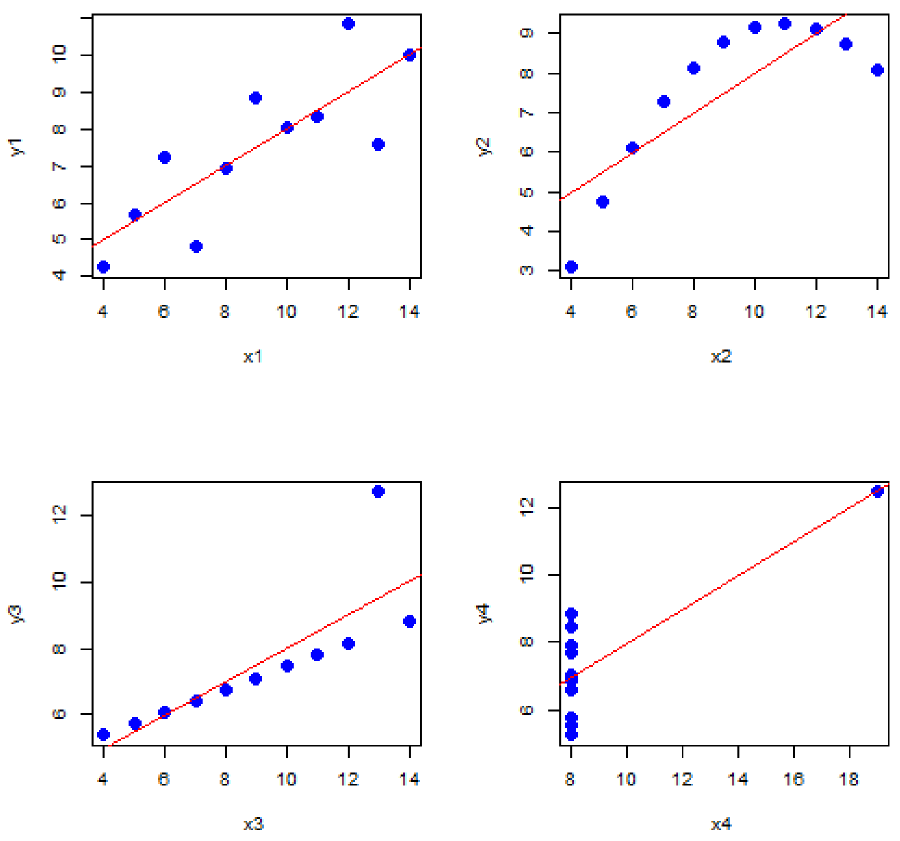
```
## Mean and standard deviation:
```

	x1	y1	x2	y2	x3	y3	x4	y4
mean	9.00	7.50	9.00	7.50	9.00	7.50	9.00	7.50
sd	3.32	2.03	3.32	2.03	3.32	2.03	3.32	2.03

```
## x-y correlation:
```

rho1	rho2	rho3	rho4
0.82	0.82	0.82	0.82

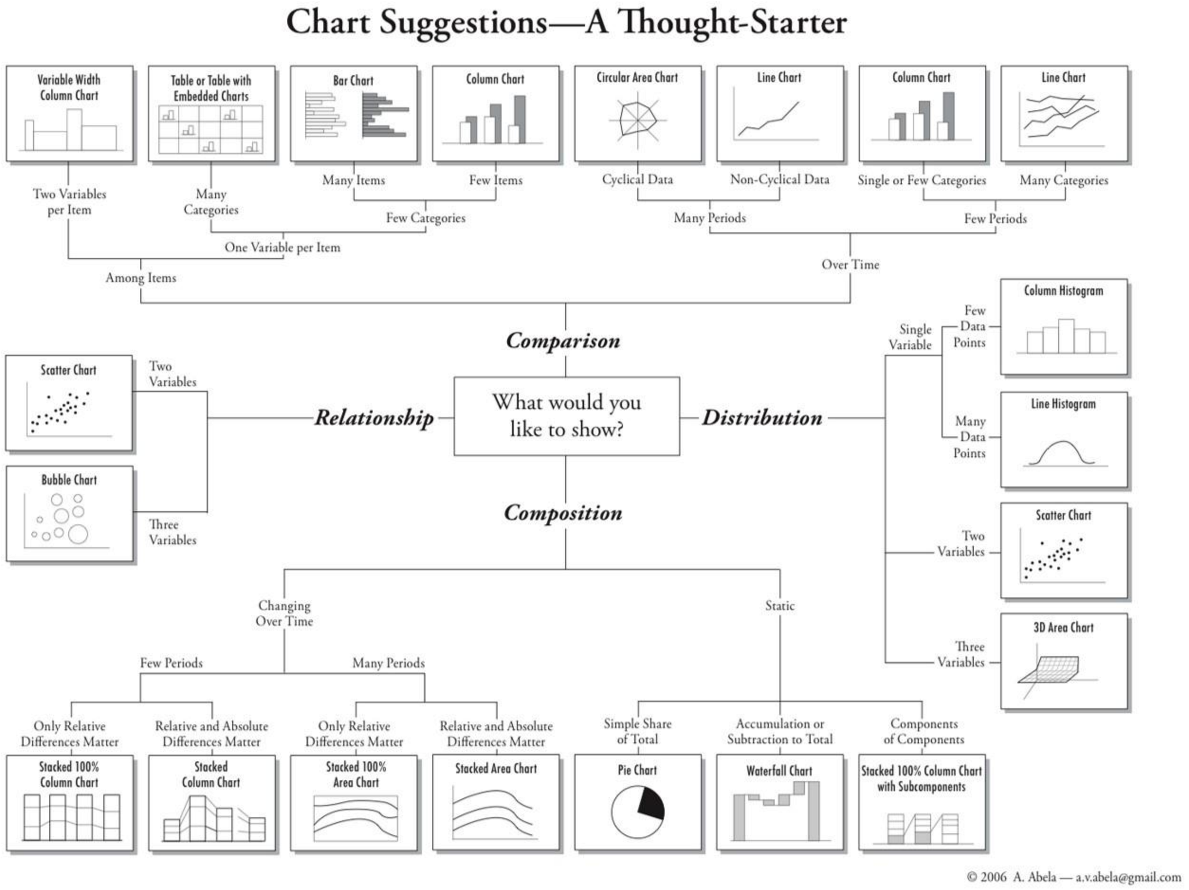
Example: Anscombe Dataset (Graphic)



Statistical Graphics

- **Univarite**
  - Histogram, Stem-and-Leaf, Dot, Q-Q, Density plots
  - Boxplot, Box-and-whisker
  - Bar, Pie, Polar, Waterfall charts
- **Bivariate**
  - XYplot, Line, Area, Scatter, Bubble charts
- **Trivariate**
  - 3D Scatter, Contour, Level/Heatmap, Surface plots

Which Chart to Use?



© 2006 A. Abela — a.v.abela@gmail.com

Simple Base Graphics

Iris Dataset

iris setosa

iris versicolor

iris virginica

```
DataX = iris   # ?iris
str(DataX)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
dim(DataX)
```

```
## [1] 150   5
```

```r
head(DataX)  # tail
```

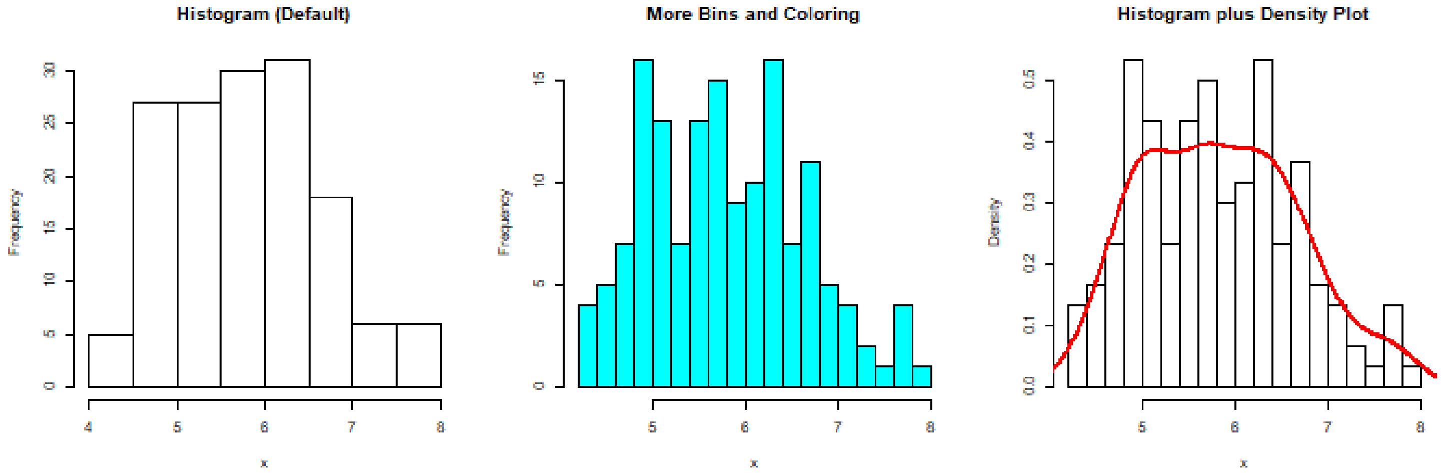
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5  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

```r
summary(DataX)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

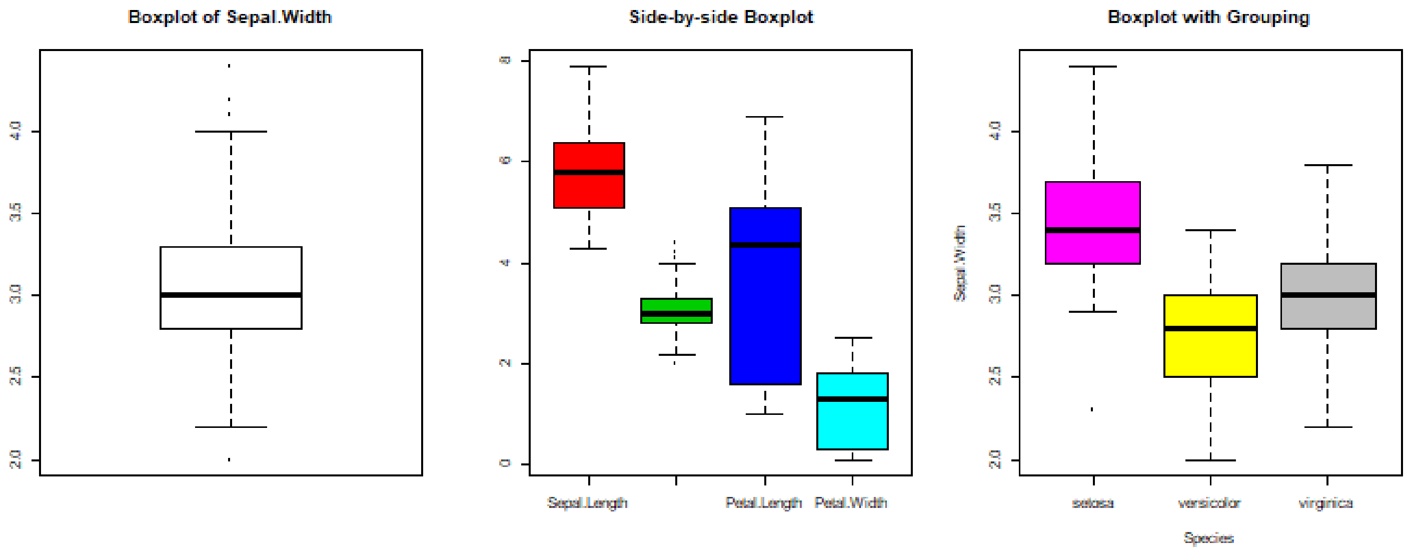
Basic R Plots: Histogram and Density Plot

```
x = DataX$Sepal.Length # a continuous variable
par(mfrow=c(1,3))
hist(x, main=’Histogram (Default)’)
hist(x, breaks=20, col=5, main=’More Bins and Coloring’)
hist(x, breaks=20, freq=F, main=’Histogram plus Density Plot’)   # using freq=FALSE
lines(density(x), col=2, lty=1, lwd=2)  #add the density curve
```



Basic R Plots: Boxplot

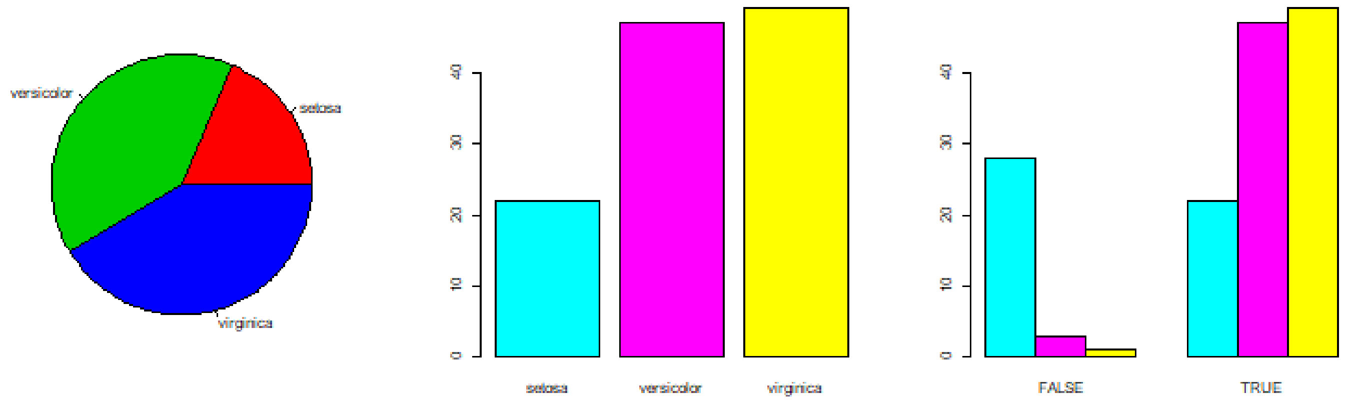
```
par(mfrow=c(1,3))
boxplot(DataX$Sepal.Width, main='Boxplot of Sepal.Width')  # Outliers
boxplot(DataX[,1:4], col=c(2,3,4,5), main='Side-by-side Boxplot')
boxplot(Sepal.Width~Species, DataX, col=c(6,7,8), main="Boxplot with Grouping")
```



- An outlier is an observation that is numerically distant from the rest of the data (e.g: outside 1.5 times the interquartile range above the upper quartile and below the lower quartile).

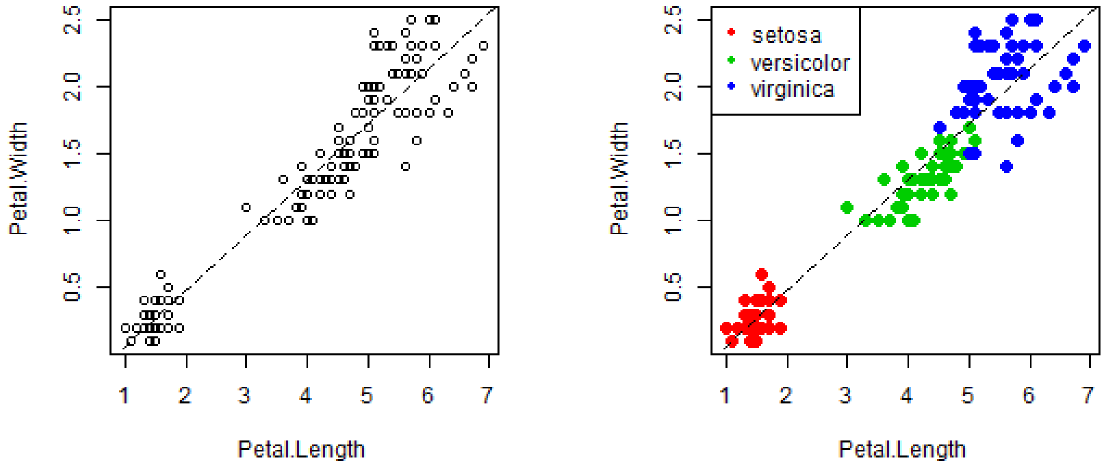
Basic R Plots: Pie and Bar Charts

```
DataX$Flag = DataX$Sepal.Length>5 # Create a binary flag
par(mfrow=c(1,3))
pie(table(DataX$Species[DataX$Flag]), col=c(2,3,4))
barplot(table(DataX$Species[DataX$Flag]), col=c(5,6,7))
barplot(table(DataX$Species, DataX$Flag), col=c(5,6,7), beside=T)
```



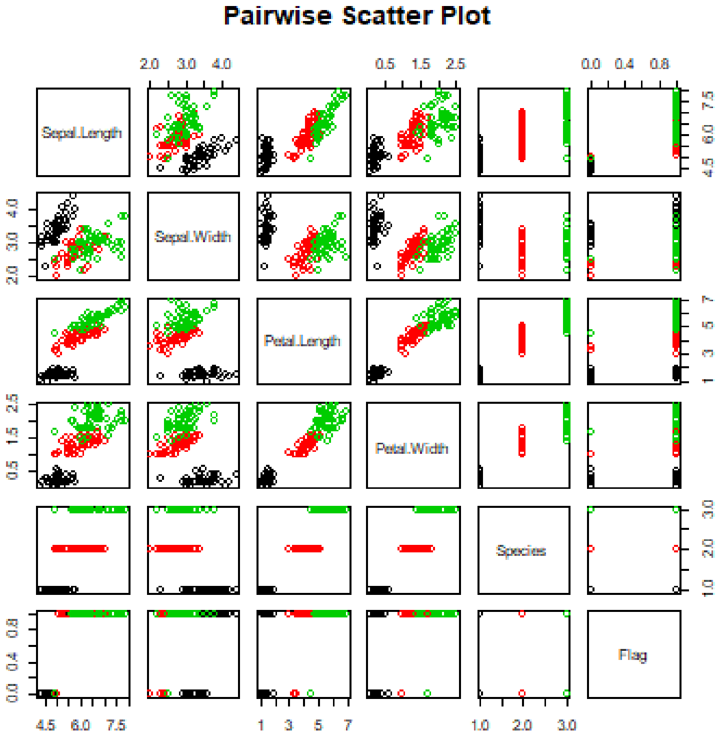
Relationship Between Variables

```
x = DataX$Petal.Length; y = DataX$Petal.Width; z = DataX$Species
par(mfrow=c(1,2)); par(mar=c(4,4,1,4))
plot(x, y, xlab="Petal.Length", ylab="Petal.Width")
abline(coef(lm(y~x)), col=1, lty=2)
plot(x, y, col=c(2,3,4)[z], pch=20, cex=2.0, xlab="Petal.Length", ylab="Petal.Width")
abline(lm(y~x), col=1, lty=2)
legend("topleft", levels(z), pch=20, col=c(2,3,4))
```

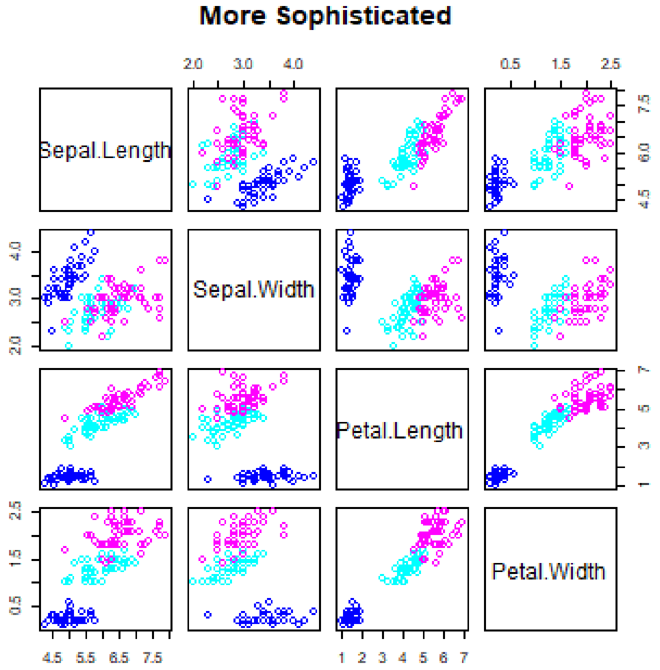


Pairwise Scatter Plot

```
plot(DataX, col=DataX$Species,
     main="Pairwise Scatter Plot")
```

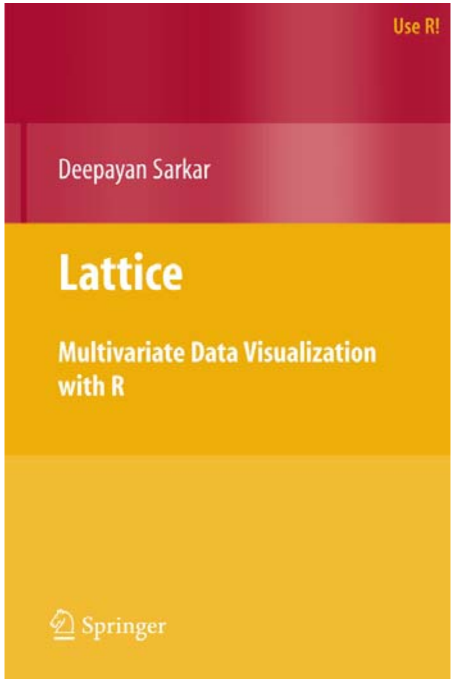


```
pairs(DataX[,1:4],
      col = c(4,5,6)[DataX$Species],
      main="More Sophisticated")
```



Using R:Lattice Package

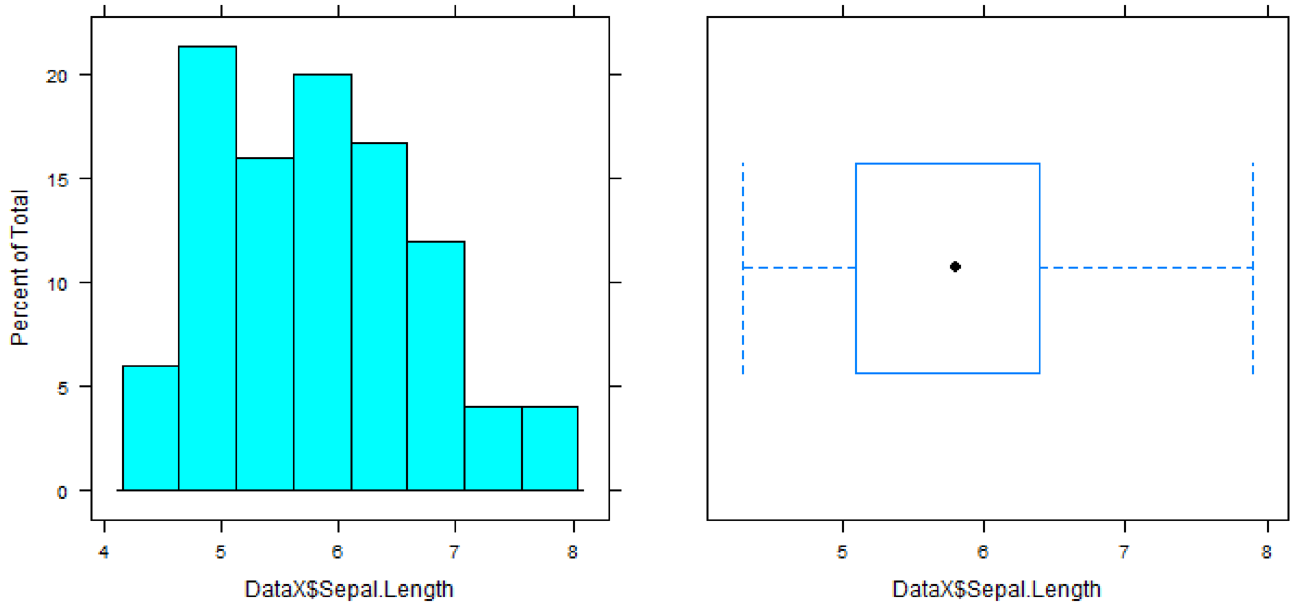
R:Lattice



- Using trellis graphs for multivariate data
- Multipanel conditioning and grouping
- Elegant high-level data visualization
- Covering most of statistical charts
- Figures and Codes can be found at http://lmdvr.r-forge.r-project.org/
- However, plot customization are not so straightforward

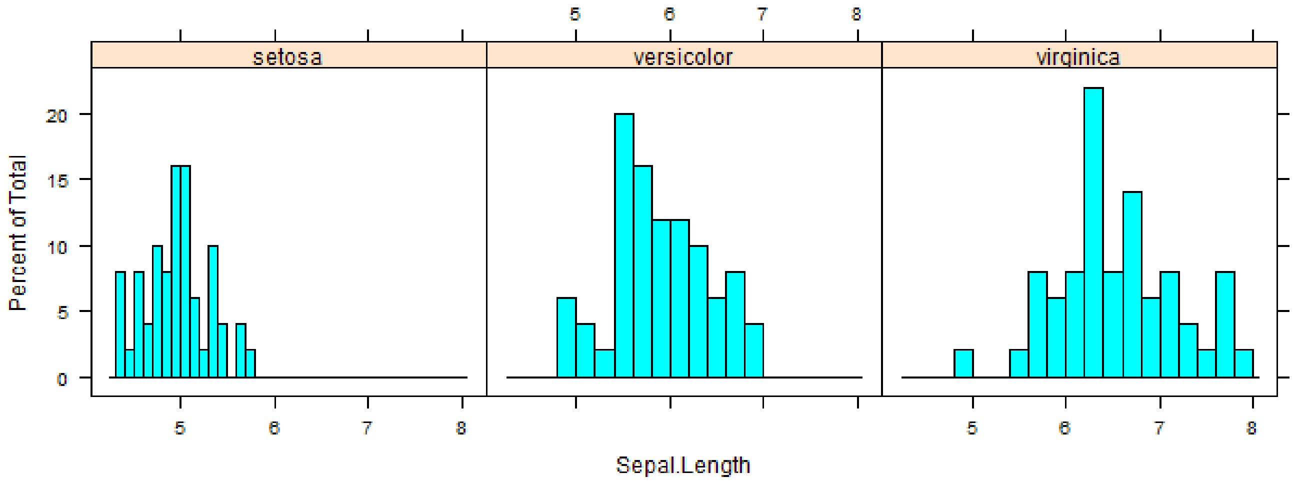
Univariate Distributions

```
library(lattice); library(gridExtra)
p1 = histogram(DataX$Sepal.Length)
p2 = bwplot(DataX$Sepal.Length)
grid.arrange(p1, p2, ncol=2)
```



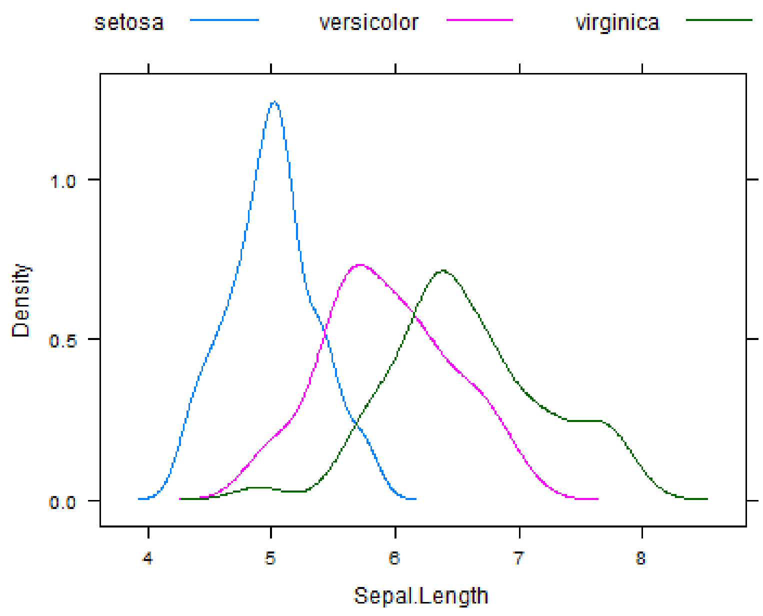
Histogram with Conditioning

```
histogram(data=DataX, ~Sepal.Length|Species, breaks=12, layout = c(3, 1))
```



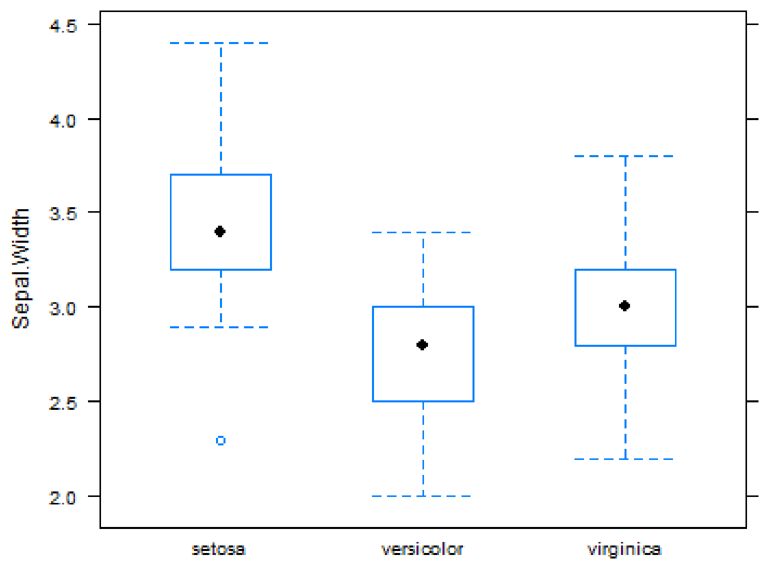
Density plot with Grouping

```
densityplot(data=DataX, ~Sepal.Length, groups=Species,
          plot.points=F, auto.key=list(space="top", columns=3))
```



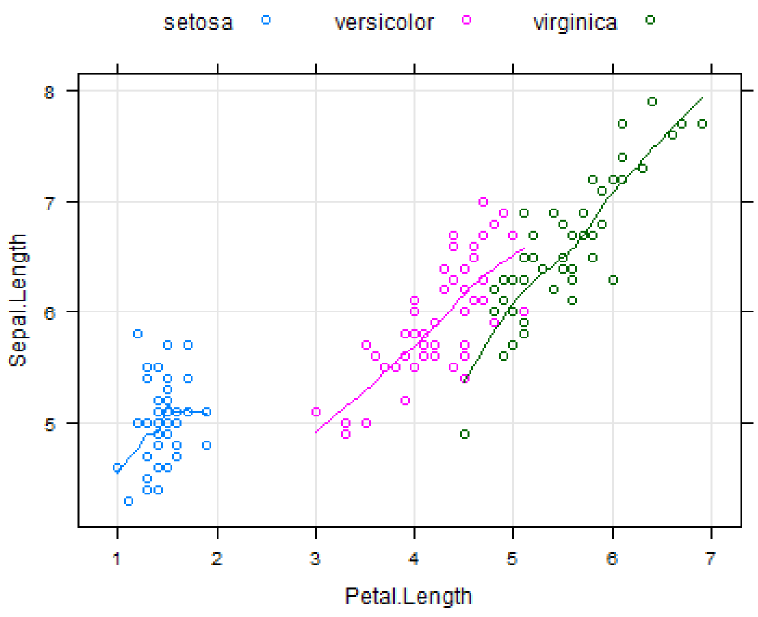
Boxplot with Grouping

```
bwplot(data=DataX, Sepal.Width~Species)
```



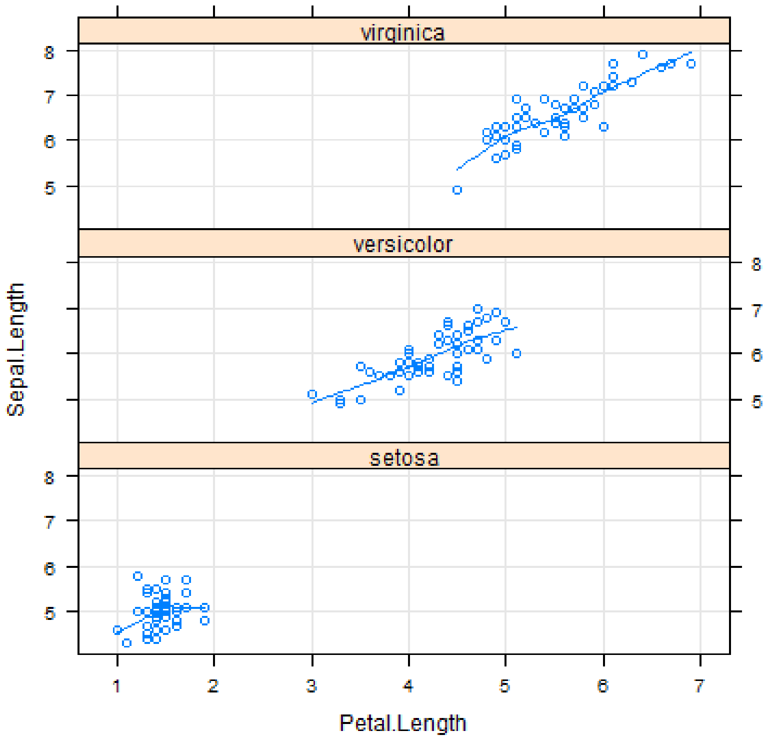
Bivariate plot with Grouping

```
xyplot(data=DataX, Sepal.Length ~ Petal.Length, groups = Species,
       type = c("p", "smooth",  "g"),
       auto.key = list(space="top", columns=3)) # grouping
```



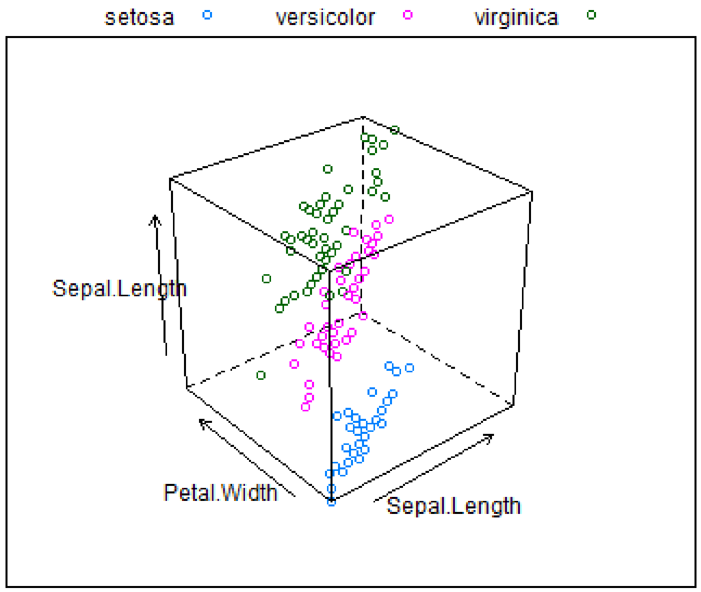
Bivariate plot with Conditioning

```
xyplot(data=DataX, Sepal.Length ~ Petal.Length|Species,
       type=c("p", "smooth",  "g"), layout=c(1,3)) # conditioning
```



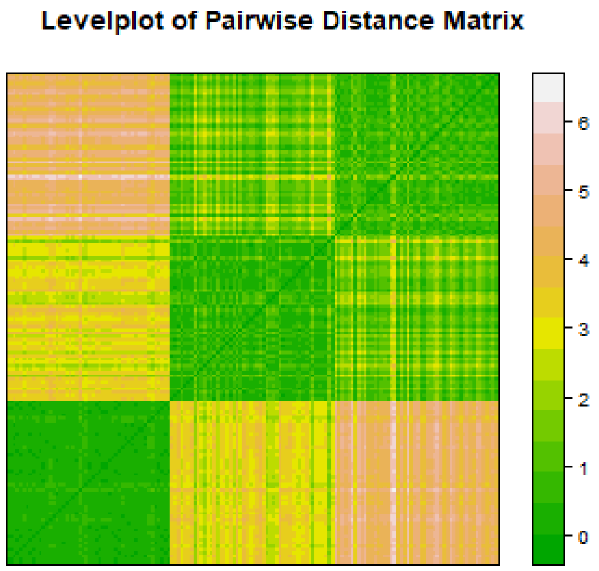
Trivariate 3D Plot

```
cloud(data=DataX, Sepal.Length ~ Sepal.Length * Petal.Width, groups = Species,
      auto.key = list(space="top", columns=3), panel.aspect = 0.8)
```



Trivariate Heatmap

```
dist = as.matrix(dist(DataX[,3:4]))
levelplot(dist, colorkey = T, col.regions = terrain.colors,
          scales = list(at=c(0,0),tck = c(0,0)),
          xlab="",ylab="",main="Levelplot of Pairwise Distance Matrix")
```



Thank you!