CSCI446/946 Big Data Analytics

Week 3 Data Analytic Methods Using R

School of Computing and Information Technology
University of Wollongong Australia

Data Analytic Methods Using R

- Introduction to R
  - R, RStudio, Data I/O, Attribute and Data Types
  - Descriptive statistics
- Exploratory Data Analysis
  - Visualization before analysis
  - Visualizing single or multiple variables
- Statistical Methods for Evaluation
  - Hypothesis Testing, ANOVA

All the figures, tables and codes are from the book "Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data" unless indicated otherwise.

Data Analytic Methods Using R

- The success of a data analysis project requires a deep understanding of the data
- It requires a toolbox for mining and presenting the data
  - Basic statistical measures
  - Creation of graphs and plots
  - Identify relationships and patterns
- R: popularity and versatility

Introduction to R

- A programming language and software framework for statistical analysis and graphics
- Comprehensive R Archive Network
- An overview the basic functionality of R
- We begin with understanding the flow of a basic R script to address an analytic problem
  - Command-line interface (CLI)
  - Graphical user interface (GUI)

Introduction to R

- The first example

```
# import a csv file of the total annual sales for each
customer
sales <- read.csv("./yearly_sales.csv")

# examine the imported dataset
head(sales)
summary(sales)

# plot num_of_orders vs. sales
plot(sales$num_of_orders,sales$sales_total,
     main="Number of Orders vs. Sales")
```

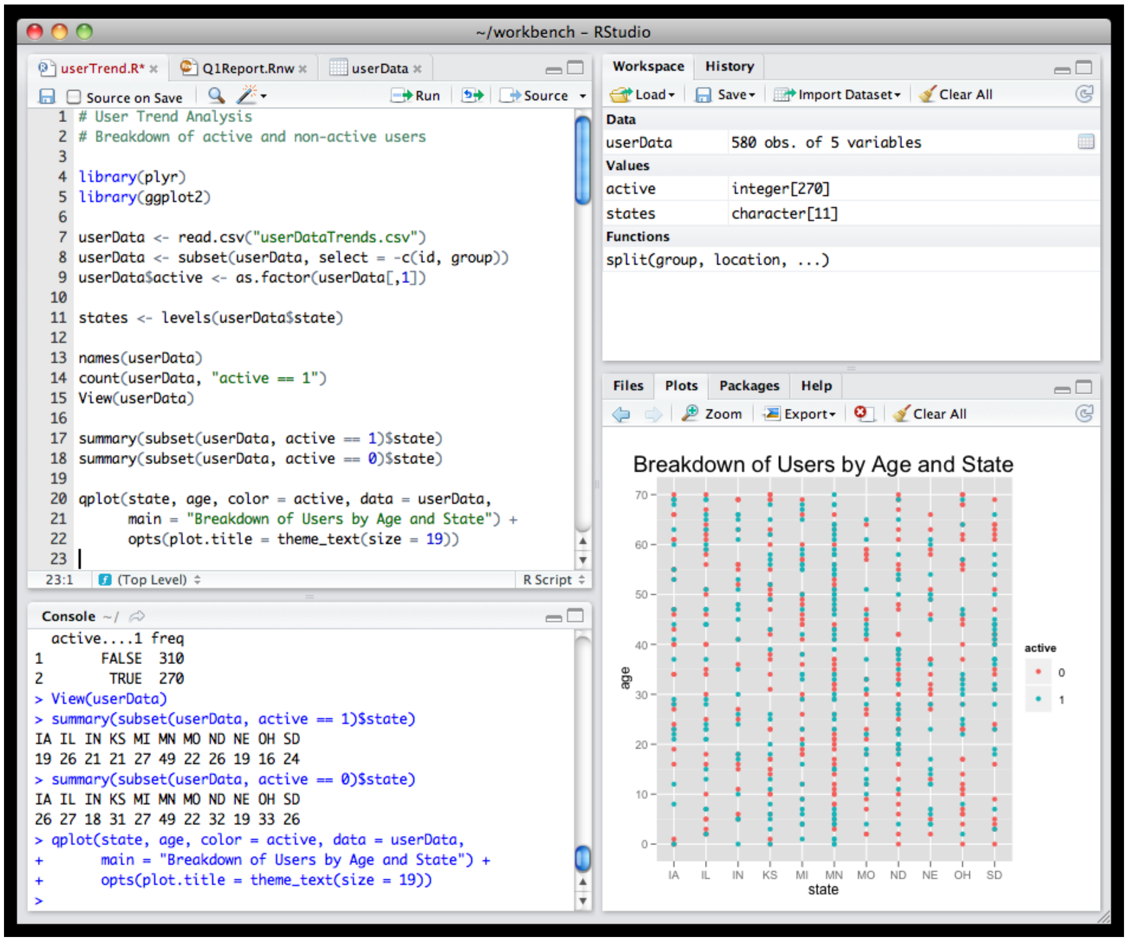
Introduction to R

- The first example

```
# perform a statistical analysis (fit a linear
regression model)
results <- lm(sales$sales_total ~ sales$num_of_orders)
results
summary(results)

# perform some diagnostics on the fitted model
# plot histogram of the residuals
hist(results$residuals, breaks = 800)
```

Introduction to R

- R Graphical User Interface (RStudio)



- Scripts
- Workspace
- Plots
- Console

Introduction to R

- Help functionality
  - Help(lm) or ?lm
- Edit() and fix()
  - Allow to update the contents of an R variable
- Save.image() function to create .Rdata file
- Load.image() function to load .Rdata file
- Please install R and RStudio to try out the R examples

Introduction to R

- Data Import and Export

```
sales <- read.csv("c:/data/yearly_sales.csv")

setwd("c:/data/")
sales <- read.csv("yearly_sales.csv")

# add a column for the average sales per order
sales$per_order <- sales$sales_total/sales$num_of_orders

# export data as tab delimited without the row names
write.table(sales,"sales_modified.txt", sep="\t",
row.names=FALSE)
```

Introduction to R

- Automatically save plots

```
# export a histogram to a jpeg
jpeg(file="c:/data/sales_hist.jpeg") # create a new jpeg
file
hist(sales$num_of_orders) # export histogram to jpeg
dev.off() # shut off the graphic device
```

- More information
  - https://cran.r-project.org/doc/manuals/r-release/R-data.html

Introduction to R

- Attribute and Data Types
- Attributes: Nominal, Ordinal, Interval, and Ratio (NOIR)

	Categorical (Qualitative)		Numeric (Quantitative)	
	Nominal	**Ordinal**	**Interval**	**Ratio**
Definition	The values represent labels that distinguish one from another.	Attributes imply a sequence.	The difference between two values is meaningful.	Both the difference and the ratio of two values are meaningful.
Examples	ZIP codes, nationality, street names, gender, employee ID numbers, `TRUE` or `FALSE`	Quality of diamonds, academic grades, magnitude of earthquakes	Temperature in Celsius or Fahrenheit, calendar dates, latitudes	Age, temperature in Kelvin, counts, length, weight
Operations	$=, \neq$	$=, \neq,$ $<, \leq, >, \geq$	$=, \neq,$ $<, \leq, >, \geq,$ $+, -$	$=, \neq,$ $<, \leq, >, \geq,$ $+, -,$ $\times, \div$

Introduction to R

- Data Types
  - Numeric, character, logical (and list)

```
i <- 1                      # create a numeric variable
sport <- "football"         # create a character variable
flag <- TRUE                # create a logical variable

class(i)                    # returns "numeric"
typeof(i)                   # returns "double"
class(sport)                # returns "character"
typeof(sport)               # returns "character"
class(flag)                 # returns "logical"
typeof(flag)                # returns "logical"
```

Introduction to R

- Vectors
  - A basic building block for data in R
  - Simple R variables are actually vectors
  - Can only consist of values in the same class

```
# Vectors
is.vector(i)                    # returns TRUE
is.vector(flag)                 # returns TRUE
is.vector(sport)                # returns TRUE
```

Introduction to R

- Vectors

```
u <- c("red", "yellow", "blue") # create a vector "red" "yellow" "blue"
u                               # returns "red" "yellow" "blue"
u[1]                            # returns "red" (1st element in u)
v <- 1:5                        # create a vector 1 2 3 4 5
v                               # returns 1 2 3 4 5
sum(v)                          # returns 15
w <- v * 2                      # create a vector 2 4 6 8 10
w                               # returns 2 4 6 8 10
w[3]                            # returns 6 (the 3rd element of w)
z <- v + w                      # sums two vectors element by element
z                               # returns 3 6 9 12 15
z > 8                           # returns FALSE FALSE TRUE TRUE TRUE
z[z > 8]                        # returns 9 12 15
z[z > 8 | z < 5]                # returns 3 9 12 15 ("|" denotes "or")
```

Introduction to R

- `vector()` function, by default, create a logical vector

```
a <- vector(length=3)               # create a logical vector of length 3
a                                   # returns FALSE FALSE FALSE
b <- vector(mode="numeric", 3)      # create a numeric vector of length 3
typeof(b)                           # returns "double"
b[2] <- 3.1                         # assign 3.1 to the 2nd element
b                                   # returns 0.0 3.1 0.0
c <- vector(mode="integer", 0)      # create an integer vector of length 0
c                                   # returns integer(0)
length(c)                           # returns 0
```

Introduction to R

- Arrays and Matrices

```
# the dimensions are 3 regions, 4 quarters, and 2 years
quarterly_sales <- array(0, dim=c(3,4,2))
quarterly_sales[2,1,1] <- 158000
quarterly_sales

sales_matrix <- matrix(0, nrow = 3, ncol = 4)
sales_matrix

install.packages("matrixcalc")  # install, if necessary
library(matrixcalc)

# build a 3x3 matrix
M <- matrix(c(1,3,3,5,0,4,3,3,3),nrow = 3,ncol = 3)
M %*% matrix.inverse(M)  # multiply M by inverse(M)
```

Introduction to R

- Data Frames
  - A structure for storing and accessing several variables of possibly different data types
  - Preferred input format for many R functions

```
sales <- read.csv("c:/data/yearly_sales.csv")
is.data.frame(sales)                  # returns TRUE

is.vector(sales$cust_id)              # returns TRUE
is.vector(sales$sales_total)          # returns TRUE
is.vector(sales$num_of_orders)        # returns TRUE
is.vector(sales$gender)               # returns FALSE
is.factor(sales$gender)               # returns TRUE
```

Introduction to R

- List: a collection of objects that can be of various types, including other lists

```
sales <- read.csv("c:/data/yearly_sales.csv")
class(sales)
typeof(sales)

# build an assorted list of a string, a numeric,
# a list, a vector, and a matrix
housing <- list("own", "rent")
assortment <- list("football", 7.5, housing, v, M)
assortment

str(assortment)
```

Introduction to R

- Factors: a categorical variable, typically with a few finite levels such as “F” and “M”
- Factors can be ordered or not ordered

```
# Factors

class(sales$gender)         # returns "factor"
is.ordered(sales$gender)    # returns FALSE
```

- Use of factors is important in R statistical modelling functions

Introduction to R

- Contingency Tables
  - A class of objects used to store the observed counts across the factors for a given dataset
  - The basis for performing a statistical test on the independence of the factors

```
# build a contingency table based on the gender and spender factors
sales_table <- table(sales$gender,sales$spender)
sales_table
```

Introduction to R

- Contingency Tables
  - A class of objects used to store the observed counts across the factors for a given dataset
  - The basis for performing a statistical test on the independence of the factors

```
class(sales_table)              # returns "table"
typeof(sales_table)             # returns "integer"
dim(sales_table)                # returns 2 3

# performs a chi-squared test
summary(sales_table)
```

Introduction to R

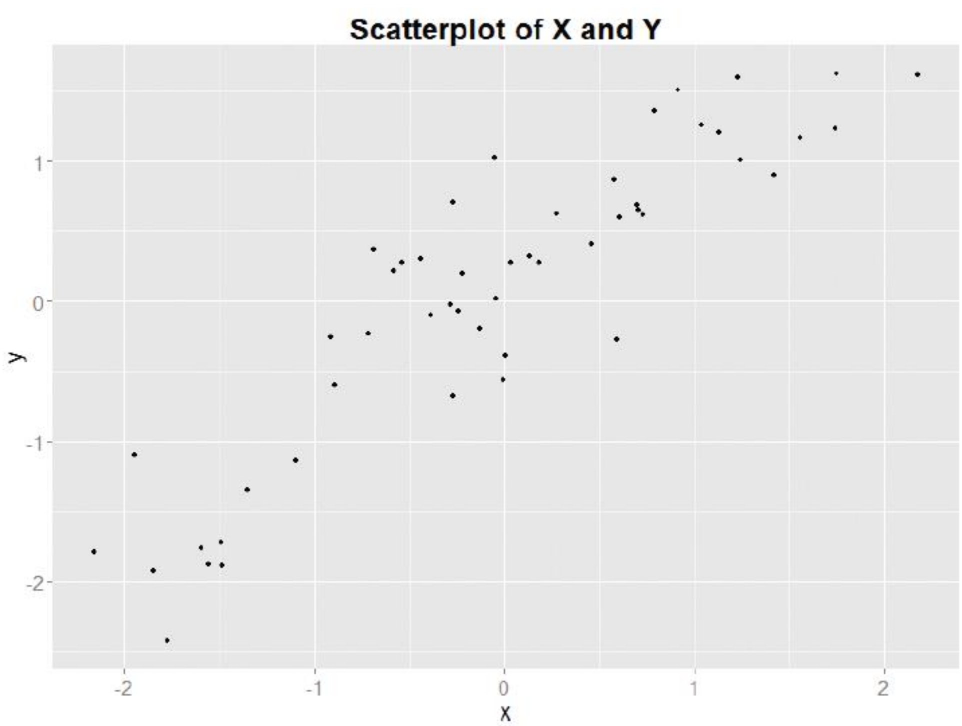
- Descriptive Statistics
  - Summary() function: mean, median, min, max
  - R functions include descriptive statistics

```
# to simplify the function calls, assign
x <- sales$sales_total
y <- sales$num_of_orders

cor(x,y)          # returns 0.7508015 (correlation)
cov(x,y)          # returns 345.2111 (covariance)
IQR(x)            # returns 215.21 (interquartile range)
mean(x)           # returns 249.4557 (mean)
median(x)         # returns 151.65 (median)
range(x)          # returns 30.02 7606.09 (min max)
sd(x)             # returns 319.0508 (std. dev.)
var(x)            # returns 101793.4 (variance)
```

Exploratory Data Analysis

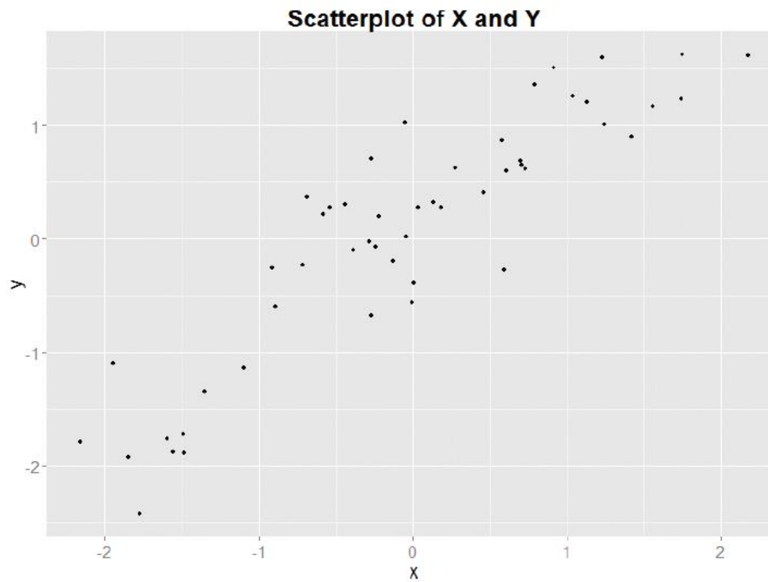
- Linear relationship and distributions are more difficult to see from descriptive statistics

```
summary(data)
       x                   y
 Min.   :-1.90483   Min.   :-2.16545
 1st Qu.:-0.66321   1st Qu.:-0.71451
 Median : 0.09367   Median :-0.03797
 Mean   : 0.02522   Mean   :-0.02153
 3rd Qu.: 0.65414   3rd Qu.: 0.55738
 Max.   : 2.18471   Max.   : 1.70199
```



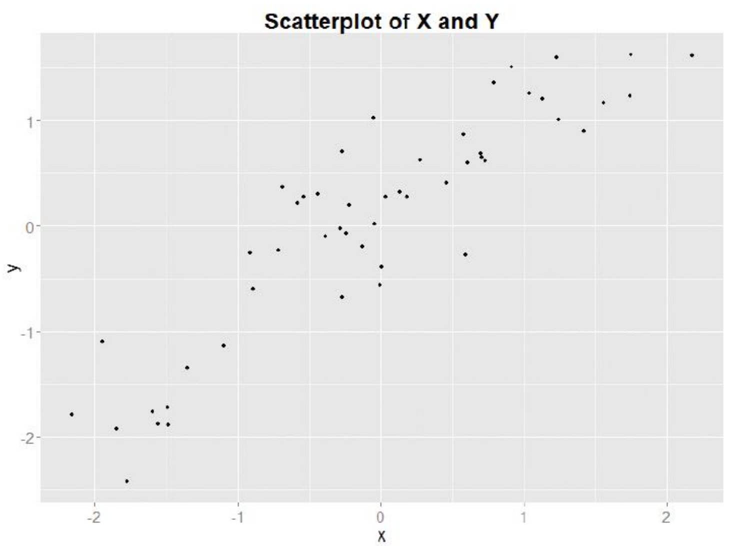
Exploratory Data Analysis

- Detect patterns and anomalies in the data
  - Through exploratory data analysis by visualization
  - Visualization gives a succinct, holistic view
  - Visualization is an important facet at the initial data exploration



Exploratory Data Analysis

```
# Figure 3-5
x <- rnorm(50)
y <- x + rnorm(50, mean=0, sd=0.5)

data <- as.data.frame(cbind(x, y))
summary(data)

library(ggplot2)
ggplot(data, aes(x=x, y=y)) +
  geom_point(size=2) +
  ggtitle("Scatterplot of X and Y") +
  theme(axis.text=element_text(size=12),
        axis.title = element_text(size=14),
        plot.title = element_text(size=20, face="bold"))
```



Exploratory Data Analysis

- Visualization Before Analysis

#1 x	#1 y	#2 x	#2 y	#3 x	#3 y	#4 x	#4 y
4	4.26	4	3.10	4	5.39	8	5.25
5	5.68	5	4.74	5	5.73	8	5.56
6	7.24	6	6.13	6	6.08	8	5.76
7	4.82	7	7.26	7	6.42	8	6.58
8	6.95	8	8.14	8	6.77	8	6.89
9	8.81	9	8.77	9	7.11	8	7.04
10	8.04	10	9.14	10	7.46	8	7.71
11	8.33	11	9.26	11	7.81	8	7.91
12	10.84	12	9.13	12	8.15	8	8.47
13	7.58	13	8.74	13	12.74	8	8.84
14	9.96	14	8.10	14	8.84	19	12.50

FIGURE 3-6 *Anscombe's quartet*

Exploratory Data Analysis

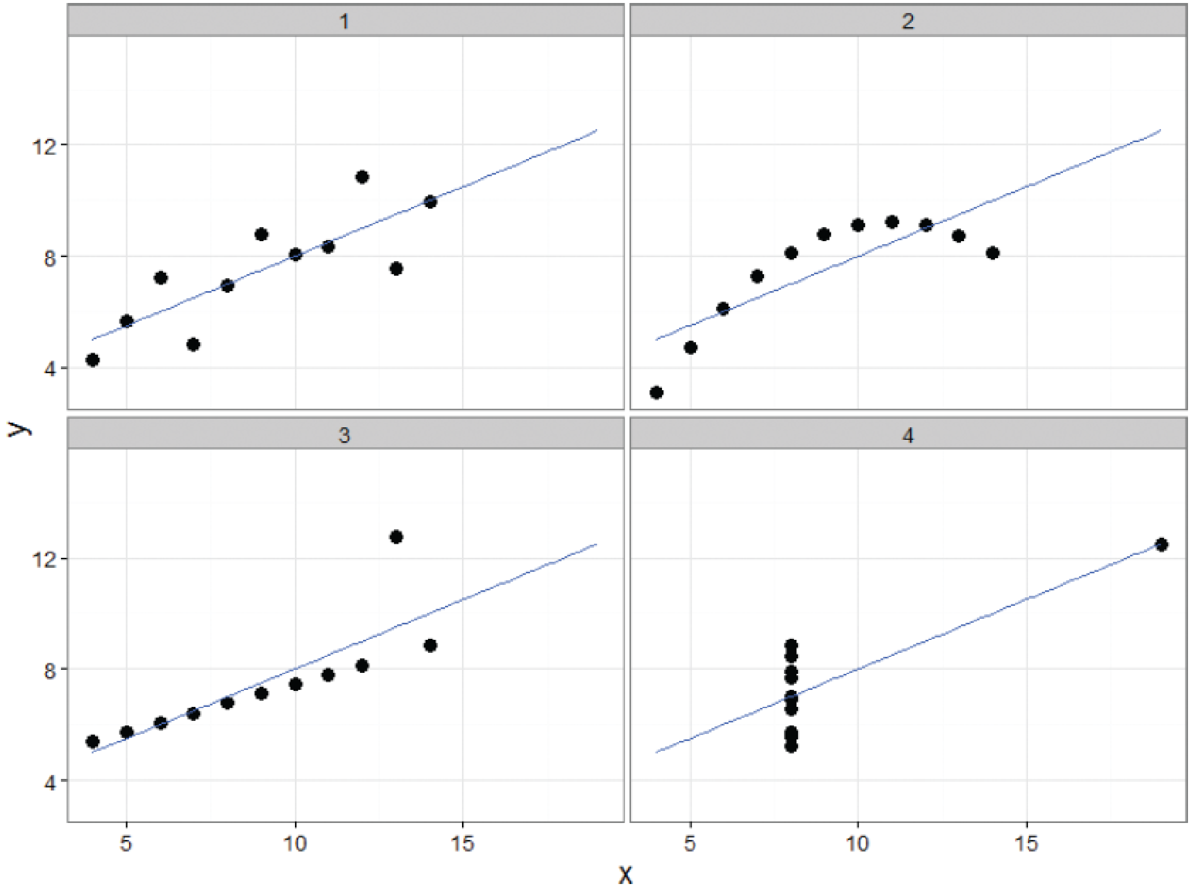
- The four data sets have nearly identical statistical properties

TABLE 3-3 *Statistical Properties of Anscombe's Quartet*

Statistical Property	Value
Mean of x	9
Variance of y	11
Mean of y	7.50 (to 2 decimal points)
Variance of y	4.12 or 4.13 (to 2 decimal points)
Correlations between x and y	0.816
Linear regression line	$y = 3.00 + 0.50x$ (to 2 decimal points)

Exploratory Data Analysis

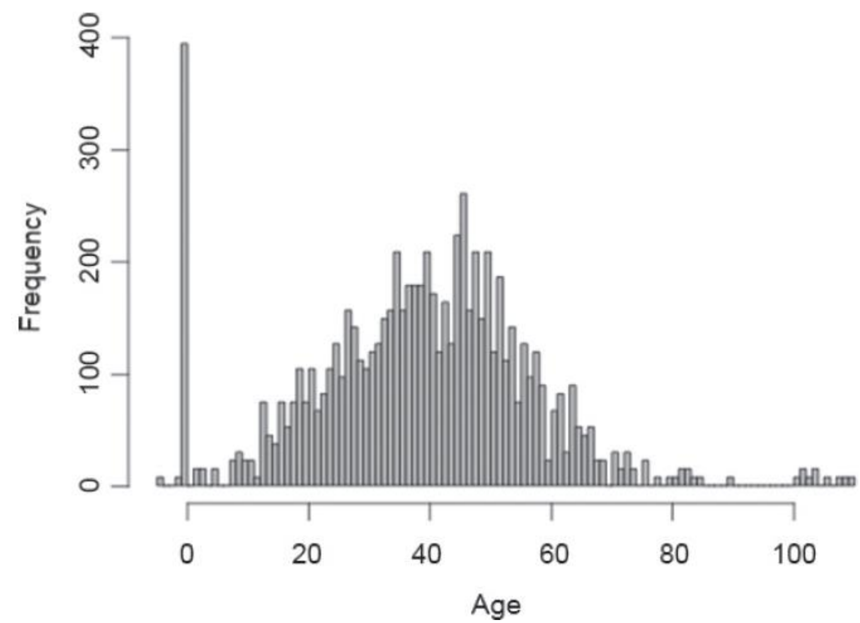
- However, the reality is a different story…



Exploratory Data Analysis

- Dirty Data
  - Detect dirty data with visualization
  - Look for anomalies, verify with domain knowledge
  - Clean the data appropriately

```
hist(age, breaks=100,
main="Age Distribution of
Account Holders",
     xlab="Age",
ylab="Frequency", col="gray")
```



Exploratory Data Analysis

- Any dirty data?


```
hist(mortgage, breaks=10, xlab="Mortgage Age", col="gray",
     main="Portfolio Distribution, Years Since Origination")
```

Exploratory Data Analysis

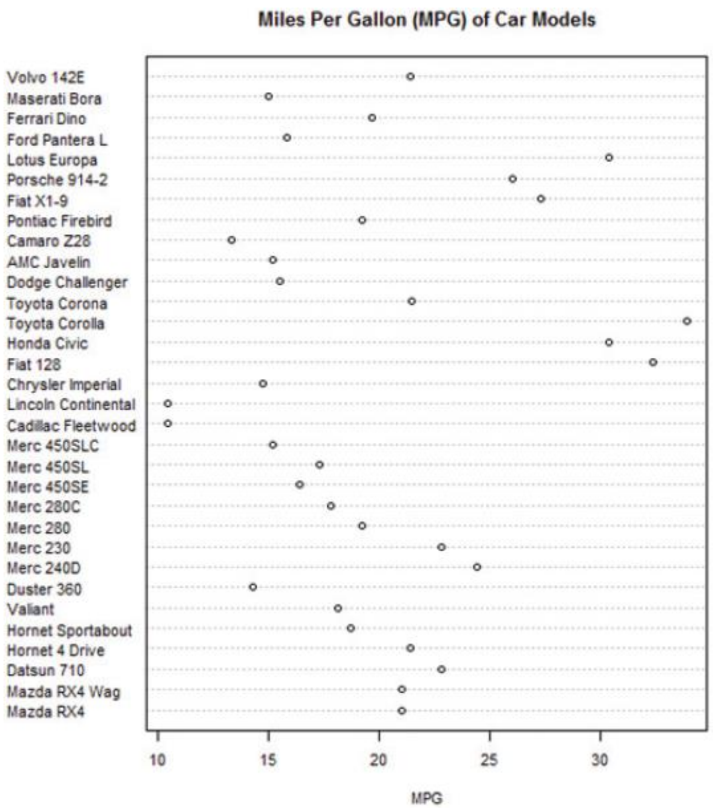
- Visualizing a Single Variable

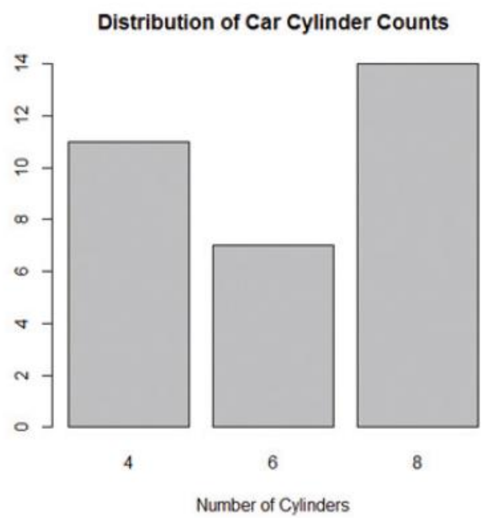
TABLE 3-4 *Example Functions for Visualizing a Single Variable*

Function	Purpose
`plot(`***data***`)`	Scatterplot where x is the index and y is the value; suitable for low-volume data
`barplot(`***data***`)`	Barplot with vertical or horizontal bars
`dotchart(`***data***`)`	Cleveland dot plot [12]
`hist(`***data***`)`	Histogram
`plot(density(`***data***`))`	Density plot (a continuous histogram)
`stem(`***data***`)`	Stem-and-leaf plot
`rug(`***data***`)`	Add a rug representation (1-d plot) of the data to an existing plot

Exploratory Data Analysis

- Visualizing a Single Variable

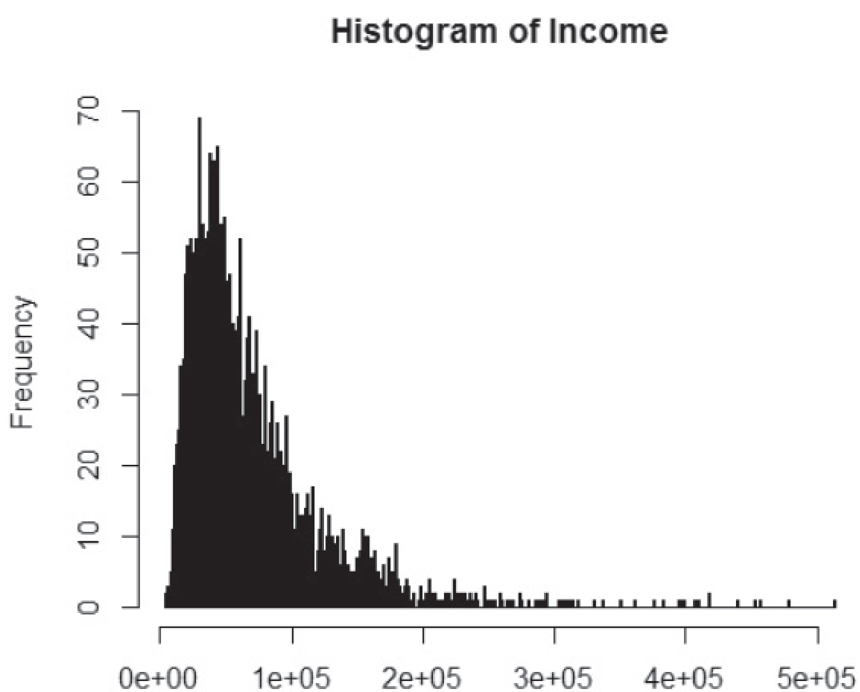


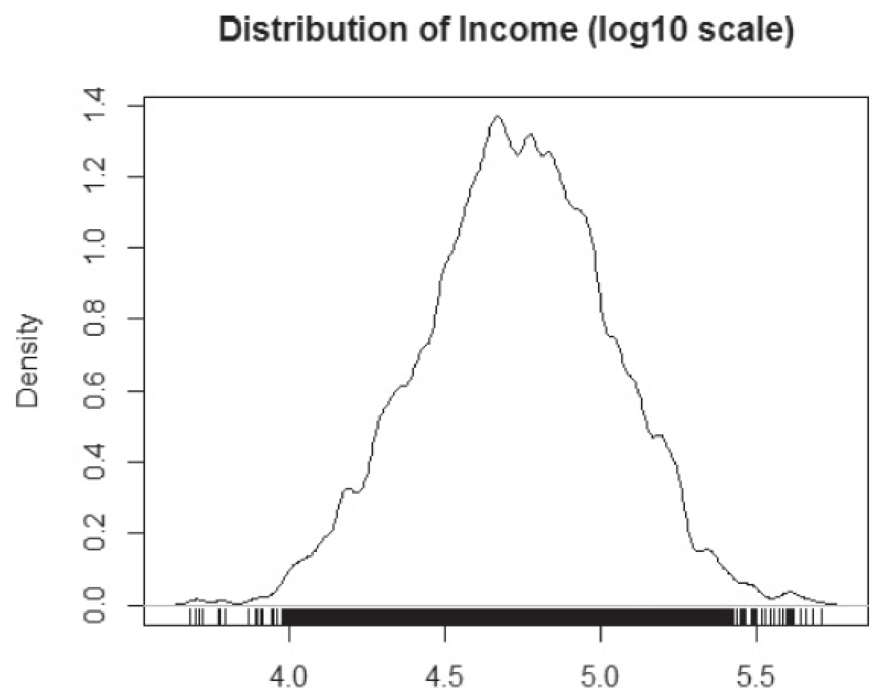



```
## Dotchart and Barplot ##
dotchart(mtcars$mpg,labels=row.names(mtcars),cex=.7, main="Miles Per Gallon (MPG) of Car Models", xlab="MPG")
barplot(table(mtcars$cyl), main="Distribution of Car Cylinder Counts", xlab="Number of Cylinders")
```

Exploratory Data Analysis

- Visualizing a Single Variable (log transformation)

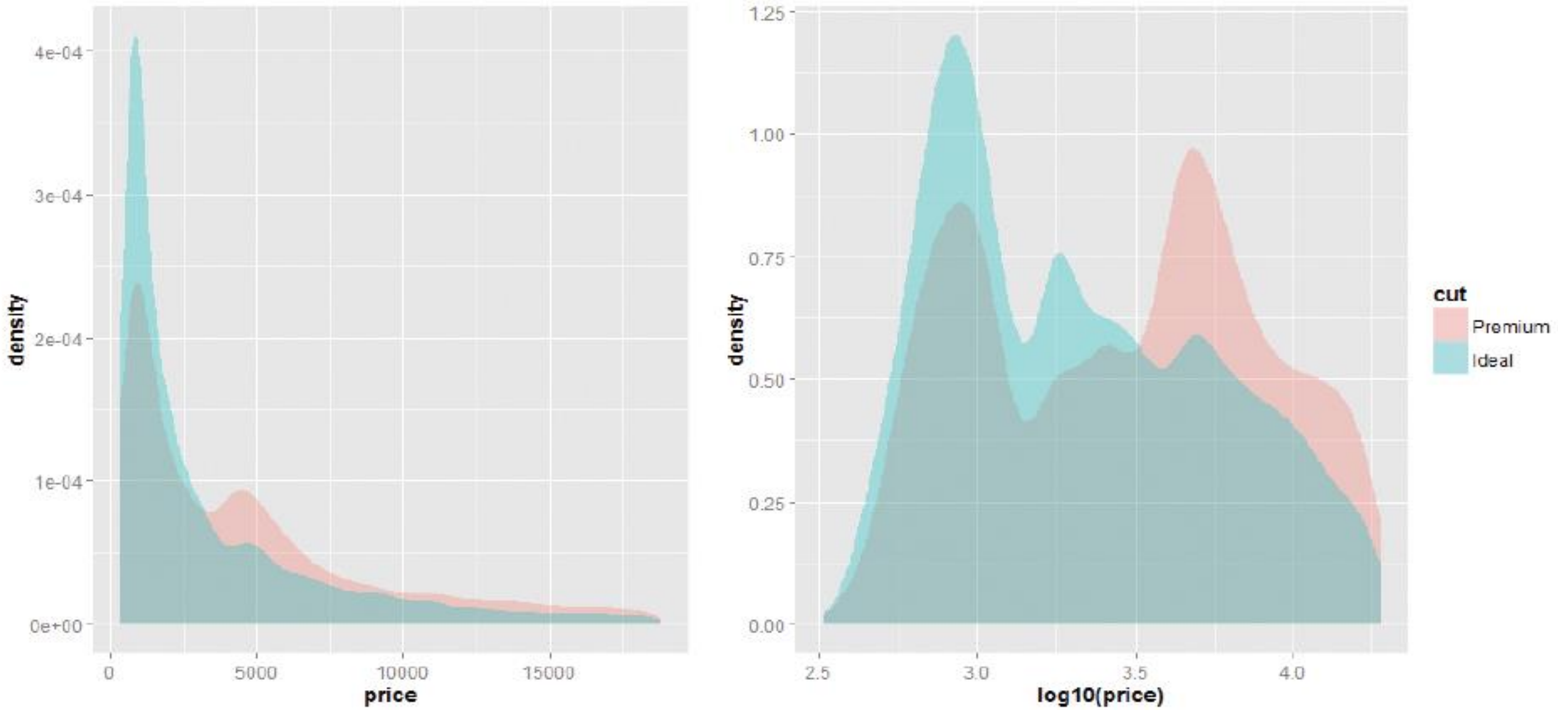





```
# plot the histogram
hist(income, breaks=500, xlab="Income", main="Histogram of Income")
# density plot
plot(density(log10(income), adjust=0.5), main="Distribution of Income (log10 scale)")
# add rug to the density plot
rug(log10(income))
```

Exploratory Data Analysis

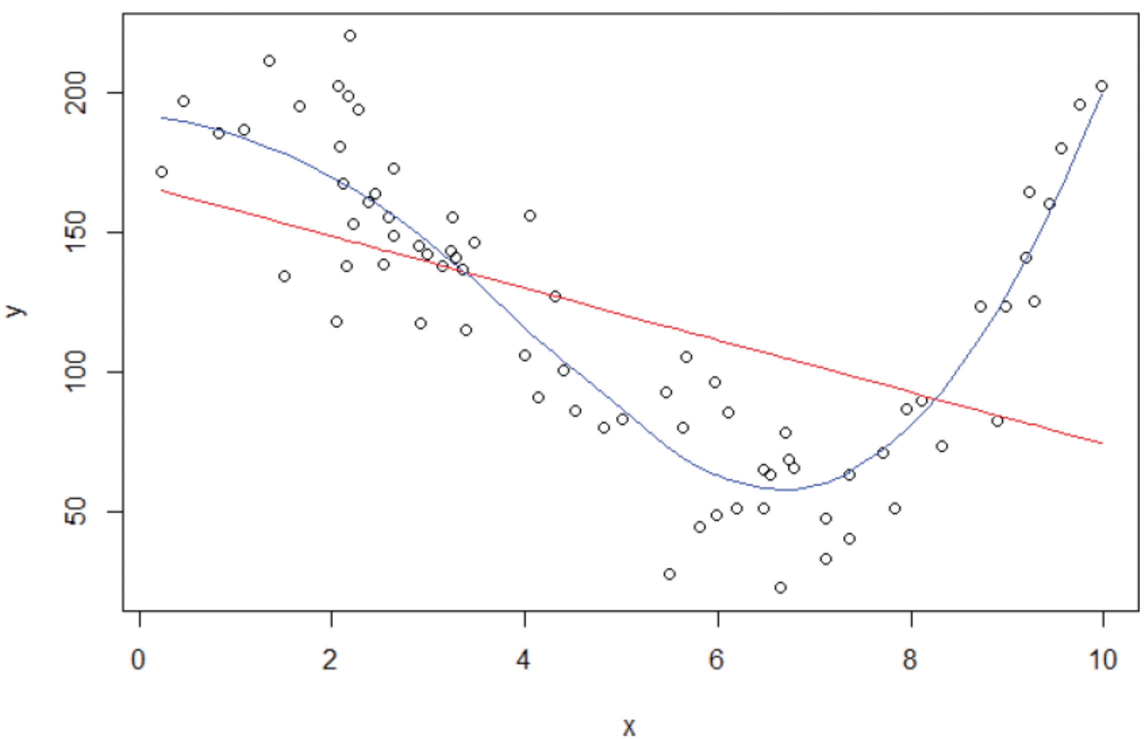
- Visualizing a Single Variable (unimodal or multimodal?)




```
# plot density plot of diamond prices
ggplot(niceDiamonds, aes(x=price, fill=cut)) +  geom_density(alpha = .3, color=NA)
# plot density plot of the log10 of diamond prices
ggplot(niceDiamonds, aes(x=log10(price), fill=cut)) +  geom_density(alpha = .3, color=NA)
```

Exploratory Data Analysis

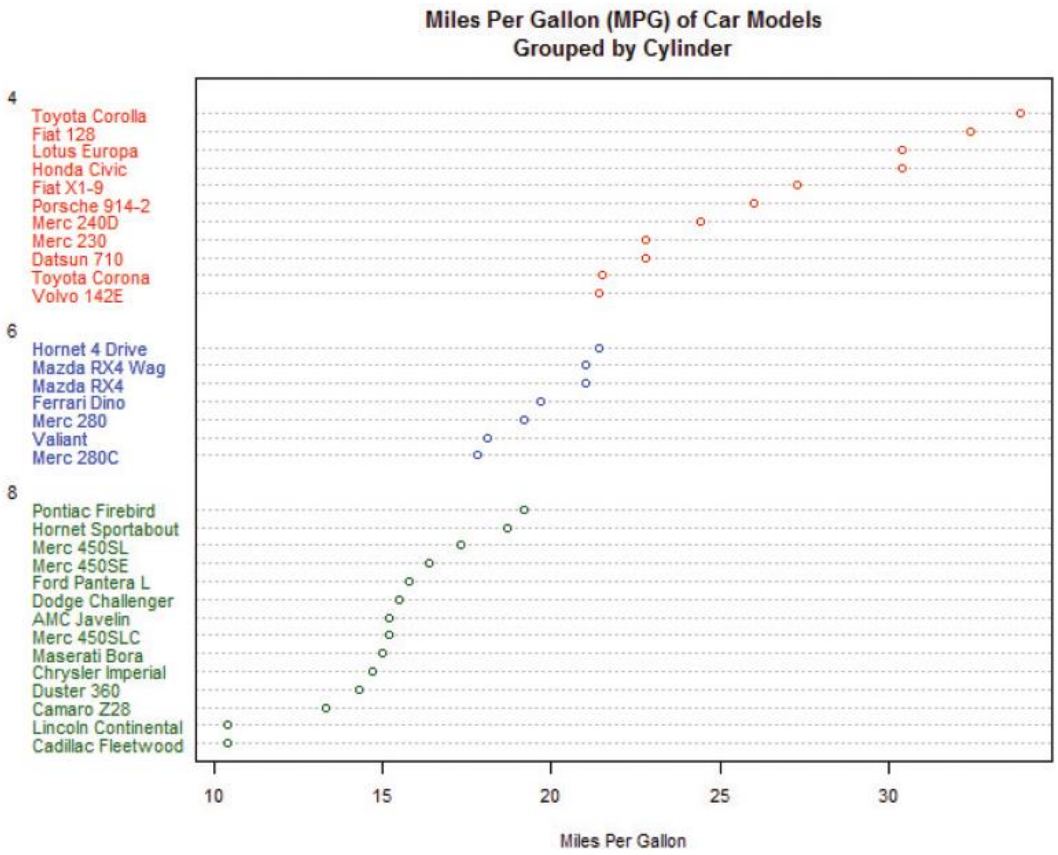
- Examining Multiple Variable




```
# 75 numbers between 0 and 10 of
uniform distribution
x <- runif(75, 0, 10)
x <- sort(x)
y <- 200 + x^3 - 10 * x^2 + x +
rnorm(75, 0, 20)
lr <- lm(y ~ x) # linear
regression
poly <- loess(y ~ x) # LOESS
fit <- predict(poly) # fit a
nonlinear line
plot(x,y)
# draw the fitted line for the
linear regression
points(x, lr$coefficients[1] +
lr$coefficients[2] * x,
       type = "l", col = 2)
# draw the fitted line with LOESS
points(x, fit, type = "l", col =
4)
```

Exploratory Data Analysis

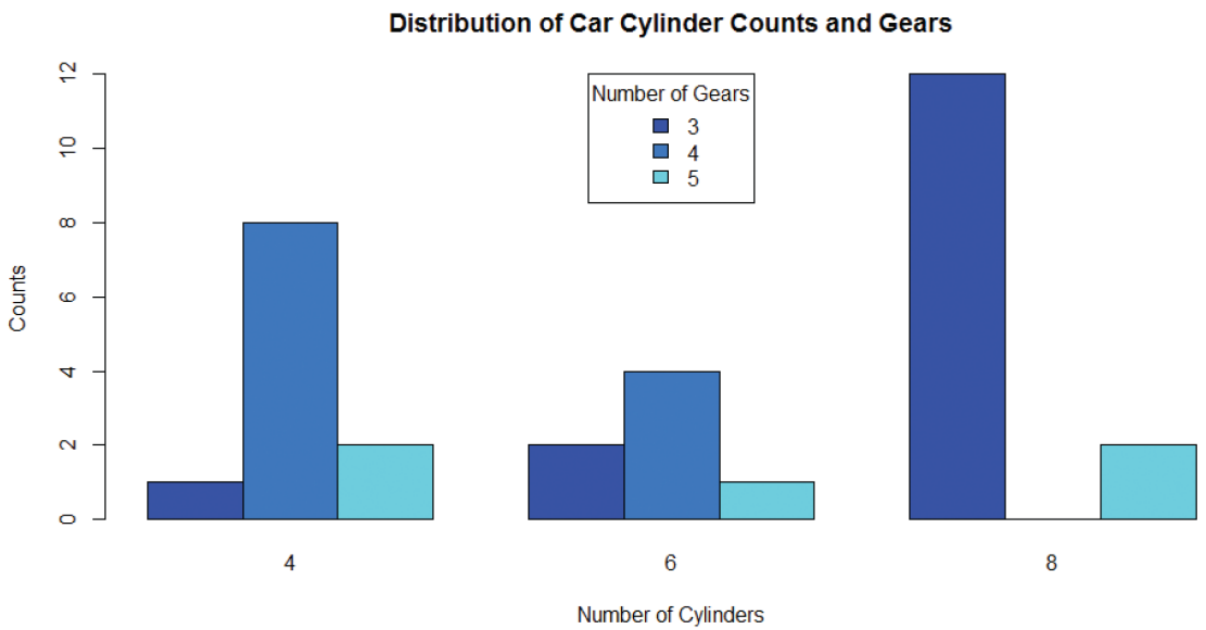
- Examining Multiple Variable




```
# sort by mpg
cars <-
mtcars[order(mtcars$mpg),]
# grouping variable must be a
factor
cars$cyl <- factor(cars$cyl)
cars$color[cars$cyl==4] <-
"red"
cars$color[cars$cyl==6] <-
"blue"
cars$color[cars$cyl==8] <-
"darkgreen"
dotchart(cars$mpg,
labels=row.names(cars),
cex=.7, groups= cars$cyl,
         main="Miles Per
Gallon (MPG) of Car
Models\nGrouped by Cylinder",
         xlab="Miles Per
Gallon", color=cars$color,
gcolor="black")
```

Exploratory Data Analysis

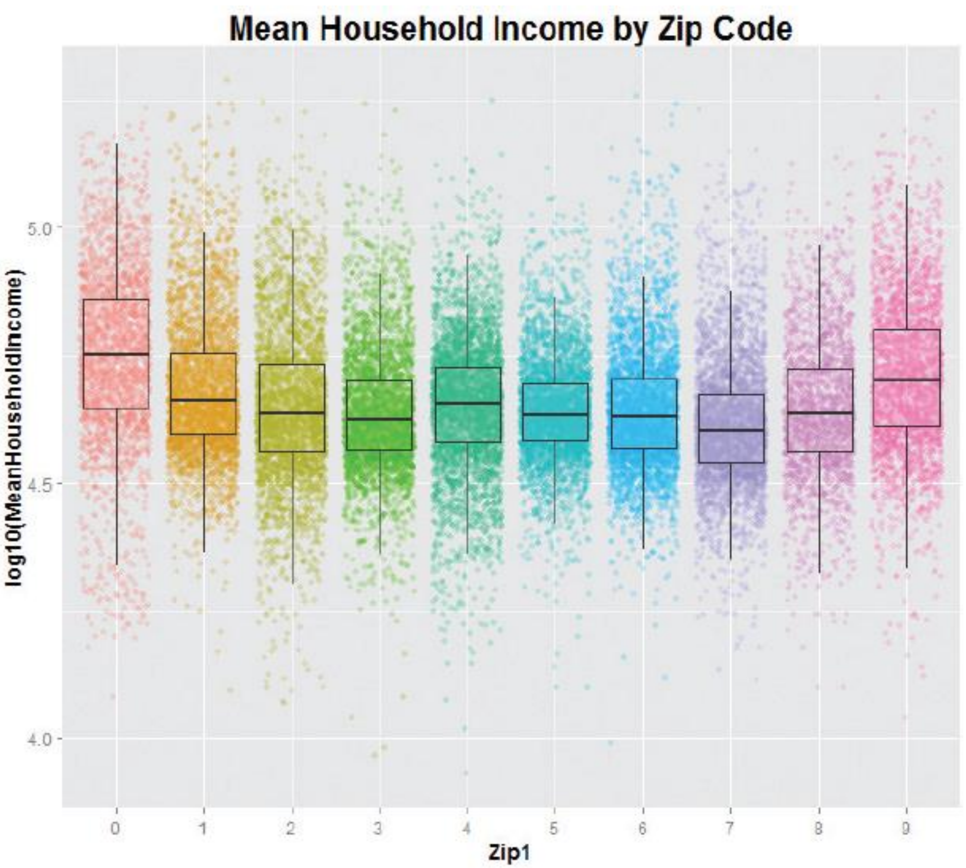
- Examining Multiple Variable




```
counts <- table(mtcars$gear, mtcars$cyl)
barplot(counts, main="Distribution of Car Cylinder Counts and Gears",
        xlab="Number of Cylinders", ylab="Counts",
        col=c("#0000FFFF", "#0080FFFF", "#00FFFFFF"),
        legend = rownames(counts), beside=TRUE,
        args.legend = list(x="top", title = "Number of Gears"))
```

Exploratory Data Analysis

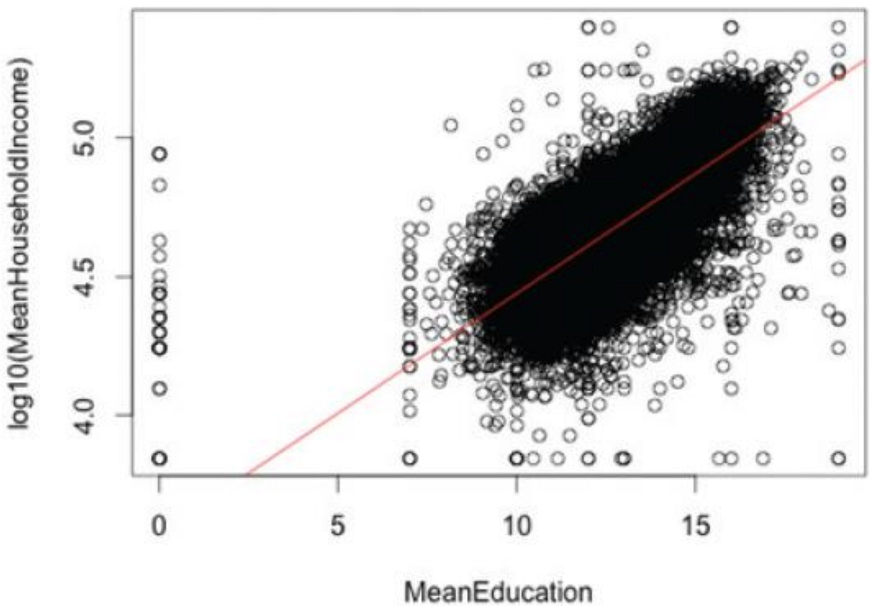
- Examining Multiple Variable (box-and-whisker plot)

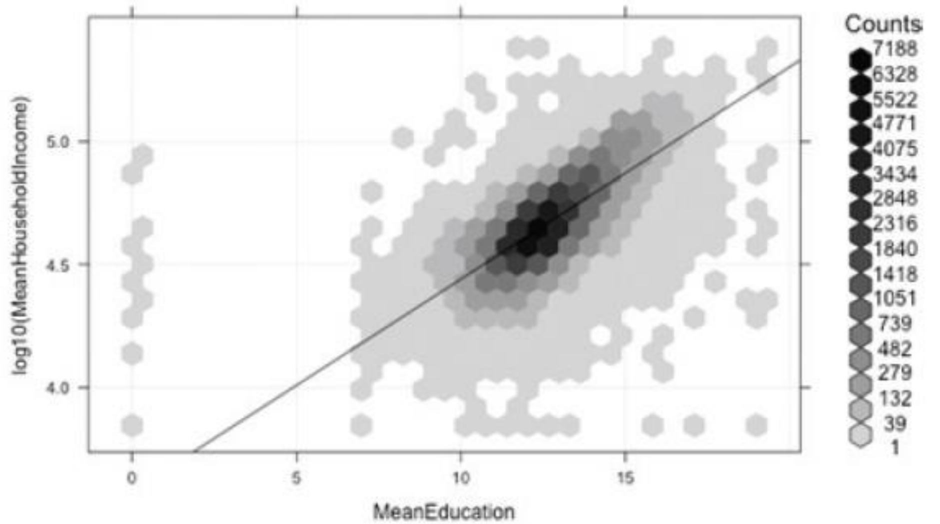



```
## Box-and-Whisker Plot ##

DF <- read.csv("c:/data/zipIncome.csv", header=TRUE,
sep=",")

# Remove outliers
DF <- subset(DF, DF$MeanHouseholdIncome > 7000 &
DF$MeanHouseholdIncome < 200000)
summary(DF)

library(ggplot2)
# plot the jittered scatterplot w/ boxplot
# color-code points with zip codes
# the outlier.size=0 prevents the boxplot from
plotting the outlier
ggplot(data=DF, aes(x=as.factor(Zip1),
y=log10(MeanHouseholdIncome))) +
  geom_point(aes(color=factor(Zip1)), alpha=0.2,
position="jitter") +
  geom_boxplot(outlier.size=0, alpha=0.1) +
  guides(colour=FALSE) +
  ggtitle ("Mean Household Income by Zip Code")

# simple boxplot
boxplot(log10(MeanHouseholdIncome) ~ Zip1, data=DF)
title ("Mean Household Income by Zip Code")
```

Exploratory Data Analysis

- Examining Multiple Variable (hexbinplot for large data)

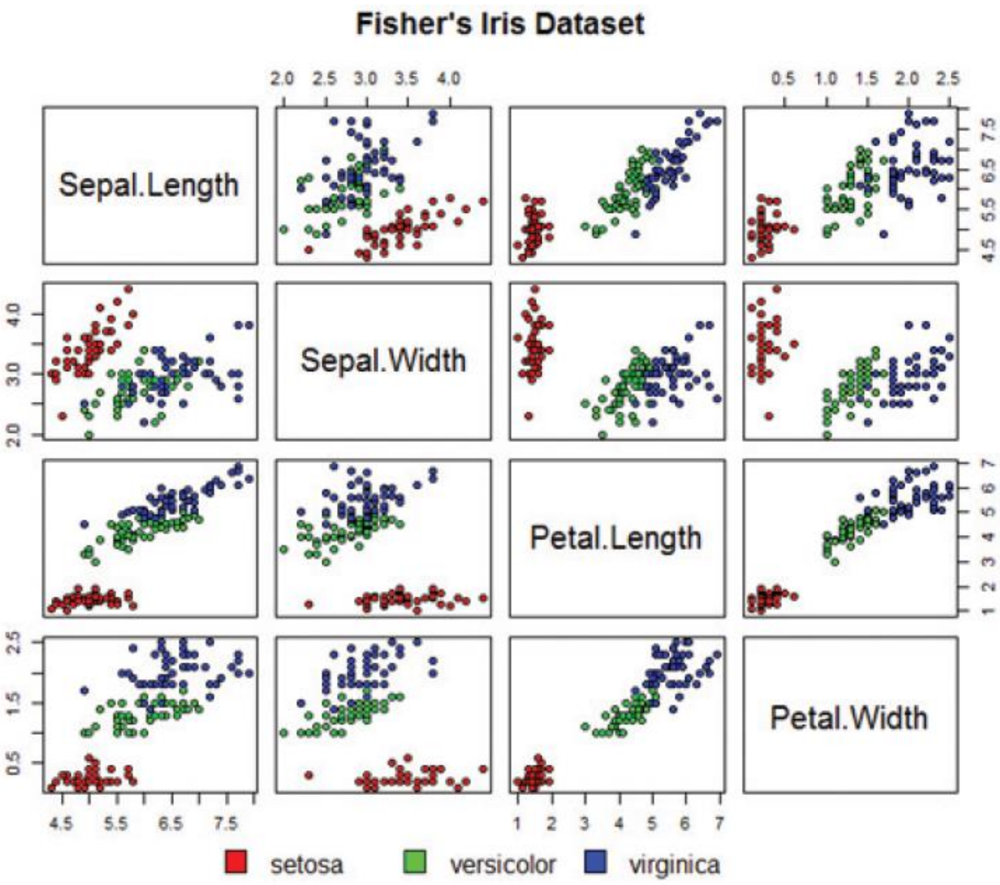





```
# plot the data points
plot(log10(MeanHouseholdIncome) ~ MeanEducation, data=DF)
# add a straight fitted line of the linear regression
abline(lm(log10(MeanHouseholdIncome) ~ MeanEducation, data=DF), col='red')
install.packages("hexbin")
library(hexbin)
# "g" adds the grid, "r" adds the regression line; sqrt transform on the count gives more dynamic range to the shading;
# inv provides the inverse transformation function of trans
hexbinplot(log10(MeanHouseholdIncome) ~ MeanEducation, data=DF, trans = sqrt, inv = function(x) x^2, type=c("g", "r"))
```

Exploratory Data Analysis

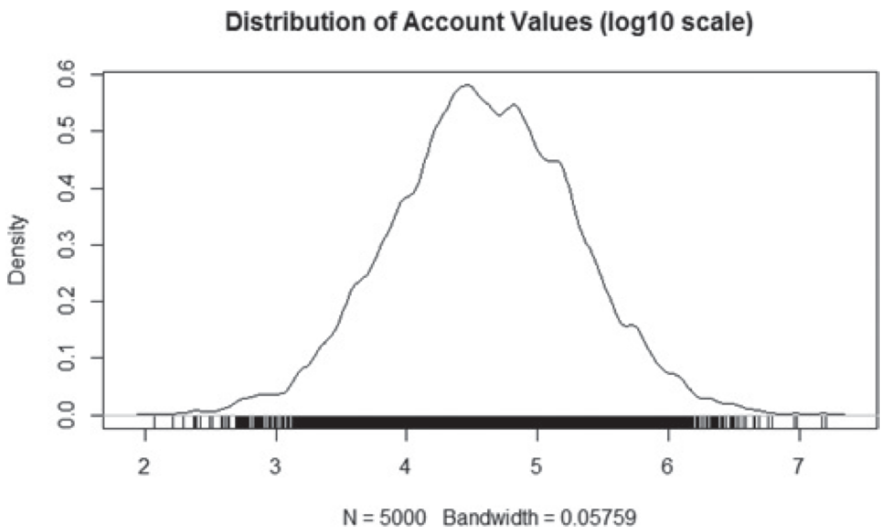
- Examining Multiple Variable (scatterplot matrix)

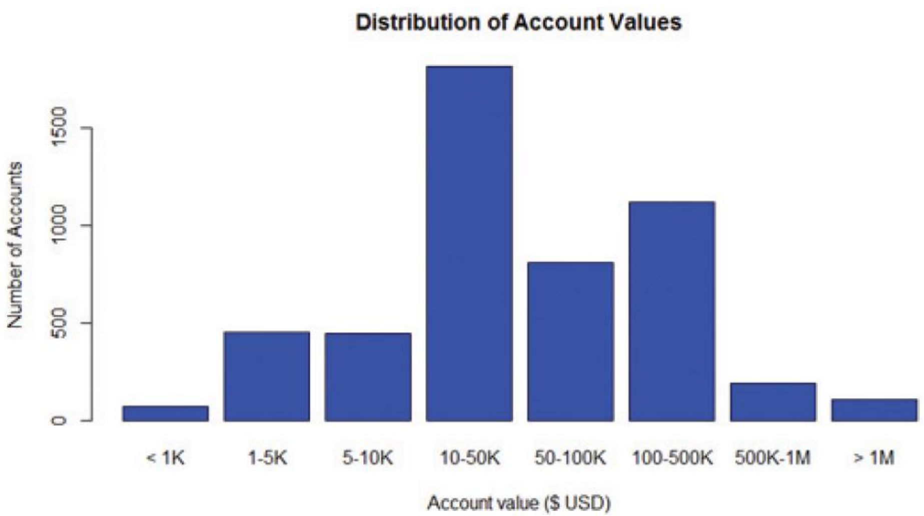



```
# define the colors
colors <- c("red", "green",
"blue")

# draw the plot matrix
pairs(iris[1:4], main = "Fisher's
Iris Dataset",
      pch = 21, bg =
colors[unclass(iris$Species)] )
# set graphical parameter to clip
plotting to the figure region
par(xpd = TRUE)
# add legend
legend(0.2, 0.02, horiz = TRUE,
as.vector(unique(iris$Species)),
       fill = colors, bty = "n")
```

Exploratory Data Analysis

- Data Exploration Versus Presentation





Presenting the same data to different audience

Statistical Methods for Evaluation

- Statistics is crucial because it may exist throughout the entire Data Analytics Lifecycle
  - Initial data exploration and data preparation
  - Model building and planning
    - Best input variables, predictability
  - Evaluation of the final models
    - Accuracy, better than guess or another one?
  - Assessment of the new models when deployed
    - Sound prediction? Have desired effect?

Statistical Methods for Evaluation

- Hypothesis Testing
  - Form an assertion and test it with data
  - Common assumption (there is no difference)
  - Null hypothesis (***H_0***)
  - Alternative hypothesis (***H_A***)
- Example: identify the effect of drug A compared to drug B on patients
  - What are the ***H_0*** and ***H_A*** ?

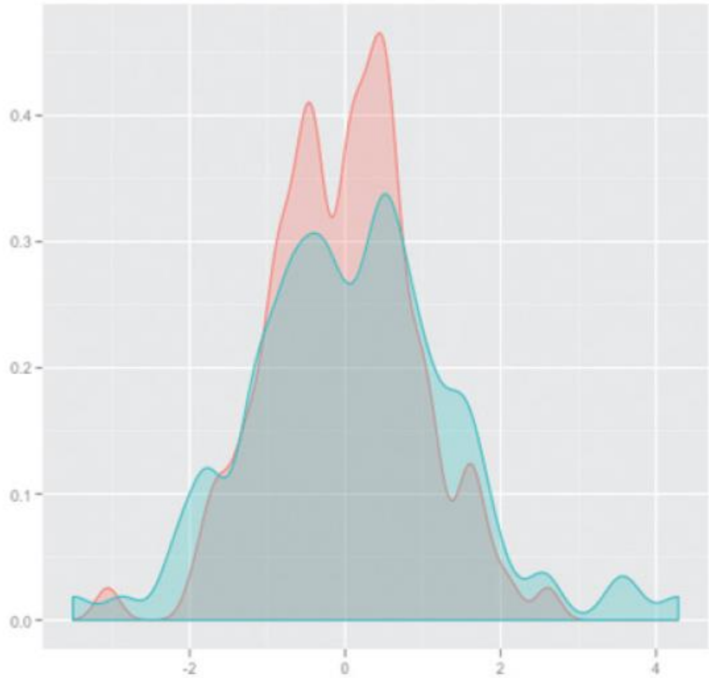
Statistical Methods for Evaluation

- Hypothesis Testing
  - Clearly state Null and Alternative hypotheses
  - **Either** reject the null hypothesis in favour of the alternative **or** not reject the null hypothesis

Application	Null Hypothesis	Alternative Hypothesis
Accuracy Forecast	Model X *does not predict* better than the existing model.	Model X *predicts* better than the existing model.
Recommendation Engine	Algorithm Y *does not produce* better recommendations than the current algorithm being used.	Algorithm Y *produces* better recommendations than the current algorithm being used.
Regression Modeling	This variable *does not affect* the outcome because its coefficient is *zero*.	This variable *affects* outcome because its coefficient is not *zero*.

Statistical Methods for Evaluation

- Difference of Means (A common hypothesis test)
  - Whether two populations are different?
  - Compare their means based on sampled data



– What are H_0 and H_A ?

Statistical Methods for Evaluation

- Student's *t*-test
  - Assume that distributions of the two populations have equal but unknown variance
  - If each population is normally distributed with the same mean and with the same variance, then

$$T = \frac{\overline{X}_1 - \overline{X}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$S_p^2 = \frac{(n_1 - 1) S_1^2 + (n_2 - 1) S_2^2}{n_1 + n_2 - 2}$$

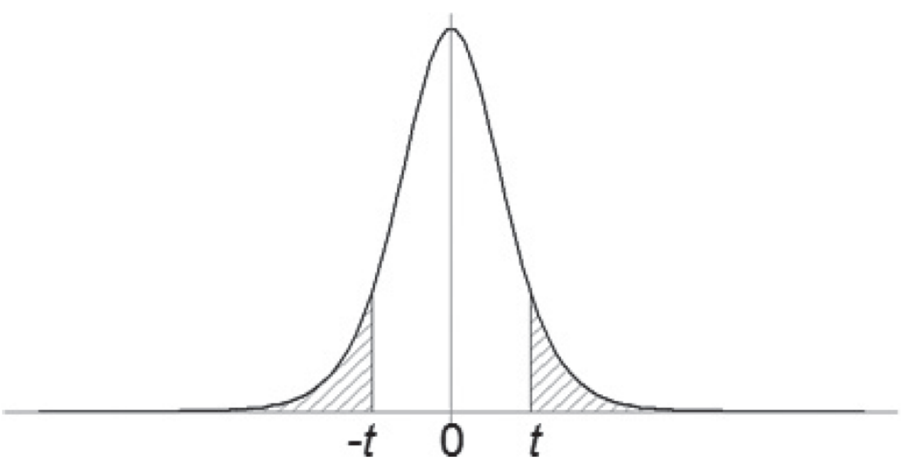
T (the *t-statistic*) follows a *t-distribution* with (n_1+n_2-2) degree of freedom

Statistical Methods for Evaluation

- Student's *t*-test
  - If the observed T is far from zero such that the probability of observing such a value of T is unlikely, one would reject the null hypothesis

$$T = \frac{\bar{X}_1 - \bar{X}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$

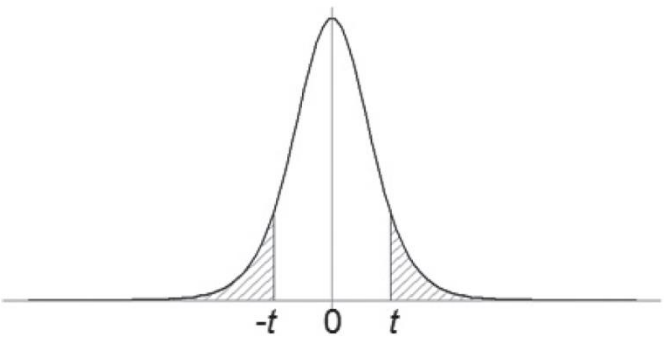


Statistical Methods for Evaluation

- Student's *t*-test
  - Significance level of the test (α): the probability of rejecting the null hypothesis, when the null hypothesis is actually TRUE
  - So, what does it mean by setting $\alpha = 0.05$?
  - Find T* such that $P(|T| \geq T^*) = \alpha$
  - Reject H_0 if $|T| \geq T^*$

$$T = \frac{\overline{X}_1 - \overline{X}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$



Statistical Methods for Evaluation

- Student's *t*-test (an example)

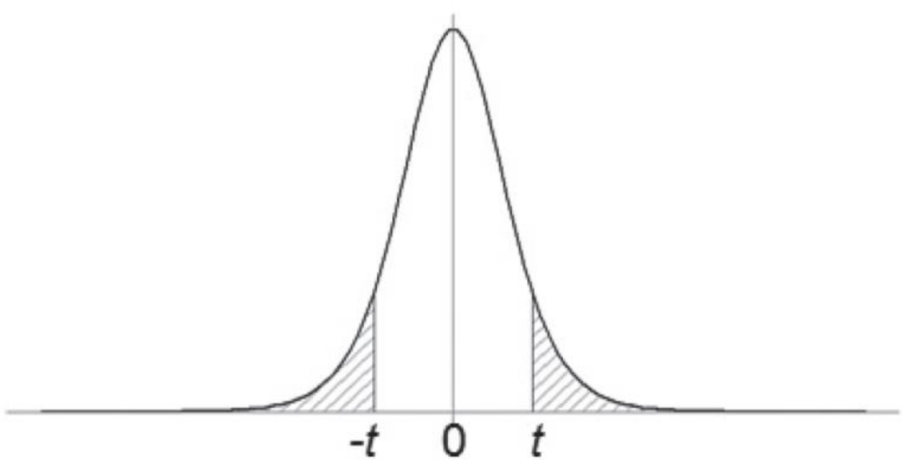
```
# generate random observations from the two populations
x <- rnorm(10, mean=100, sd=5)     # normal distribution centered at 100
y <- rnorm(20, mean=105, sd=5)     # normal distribution centered at 105

t.test(x, y, var.equal=TRUE)       # run the Student's t-test
Two Sample t-test

data:  x and y
t = -1.7828, df = 28, p-value = 0.08547
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  -6.1611557  0.4271893
sample estimates:
  mean of x mean of y
102.2136  105.0806
```

Statistical Methods for Evaluation

- Student's *t*-test (an example)

```
# obtain t value for a two-sided test at a 0.05 significance level
qt(p=0.05/2, df=28, lower.tail= FALSE)
2.048407
```

- Shall we reject or accept the null hypothesis?
- What does the "two-sided test" mean?

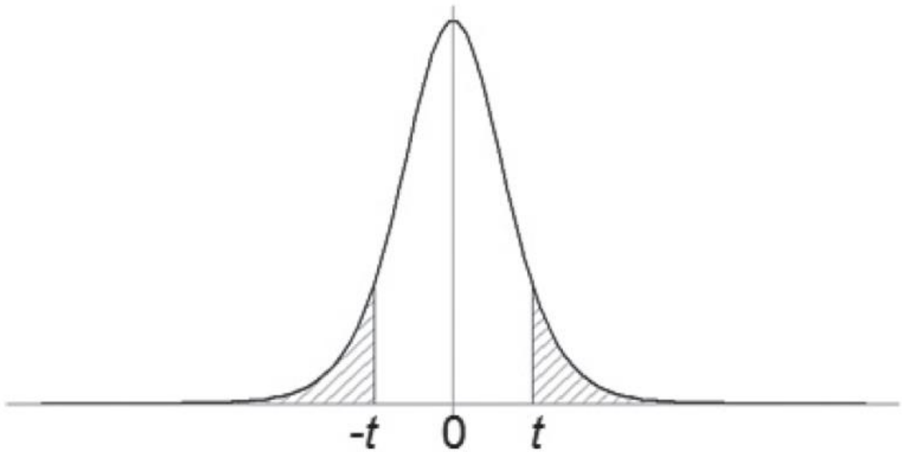


Statistical Methods for Evaluation

- Student's *t*-test (an example)
  - What does the "p-value" mean?

    ```
    t = -1.7828, df = 28, p-value = 0.08547
    ```

  - The sum of $P(T \le -t)$ and $P(T \ge t)$
  - p-value offers the probability of observing $|T| \ge t$ given the null hypothesis is TRUE



Statistical Methods for Evaluation

- Student's *t*-test (an example)
  - What is the "95 percent confidence interval"?

```
95 percent confidence interval:
 -6.1611557  0.4271893
```

  - A confidence level is an interval estimate of a population parameter based on sample data
  - The above "95 percent confidence interval" straddles the TRUE value of the difference of the population means 95% of the time

Statistical Methods for Evaluation

- Welch's *t*-test
  - Shall be used when the equal population variance assumption is NOT justified
  - It uses the sample variance for each population instead of the pooled sample variance
  - Still assume two populations are normal with the same mean

$$T_{welch} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

Statistical Methods for Evaluation

- Welch's *t*-test

```
t.test(x, y, var.equal=FALSE)        # run the Welch's t-test

Welch Two Sample t-test

data:  x and y
t = -1.6596, df = 15.118, p-value = 0.1176
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6.546629  0.812663
sample estimates:
 mean of x mean of y
102.2136  105.0806
```

Statistical Methods for Evaluation

- Wilcoxon Rank-Sum Test
  - What if the two populations are not normal?
- Parametric test
  - Makes assumptions about the population distributions from which the samples are drawn
- Nonparametric test
  - Shall be used if the populations cannot be assumed (or transformed) to be normal

Statistical Methods for Evaluation

- Wilcoxon Rank-Sum Test
  - A nonparametric test to check whether two populations are identically distributed
  - It uses “ranks” instead of numerical outcomes to avoid specific assumption about the distribution
- How to conduct the test
  - Rank two samples as if they are from one group
  - Sum assigned ranks for one population’s sample
  - Determine the significance of the rank-sums

Statistical Methods for Evaluation

- Wilcoxon Rank-Sum Test

```
wilcox.test(x, y, conf.int = TRUE)

Wilcoxon rank sum test

data:  x and y
W = 55, p-value = 0.04903
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
  -6.2596774 -0.1240618
sample estimates:
  difference in location
-3.417658
```

p-value: the probability of the rank-sums of this magnitude being observed assuming that the population distributions are identical

Statistical Methods for Evaluation

- Type I and Type II Errors
  - Type I error: the rejection of the null hypothesis when the null hypothesis is TRUE
  - The probability of type I error is denoted by α
  - Type II error: the acceptance of the null hypothesis when the null hypothesis is FALSE
  - The probability of type II error is denoted by β
- Power (statistical power)
  - The probability of correcting rejecting the null hypothesis $(1- \beta)$

Statistical Methods for Evaluation

- ANOVA (Analysis of Variance)
  - What if there are more than two populations?
  - Multiple *t*-test may not perform well now
- A generalization of the hypothesis testing
  - ANOVA tests if any of the population means differ from the other population means
  - Each population is assumed to be normal and have the same variance

Statistical Methods for Evaluation

- ANOVA (Analysis of Variance)

$$\boldsymbol{H_0}: \mu_1 = \mu_2 = \ldots = \mu_n$$

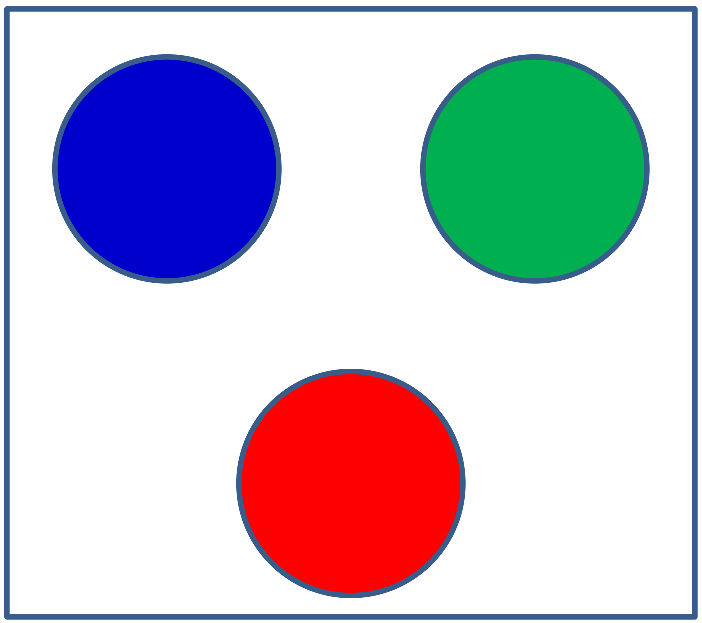
$$\boldsymbol{H_A}: \mu_i \neq \mu_j \text{ for at least one pair of } i, j$$

- Compute *F*-test statistic
  - Between-groups mean sum of squares
  - Within-groups mean sum of squares

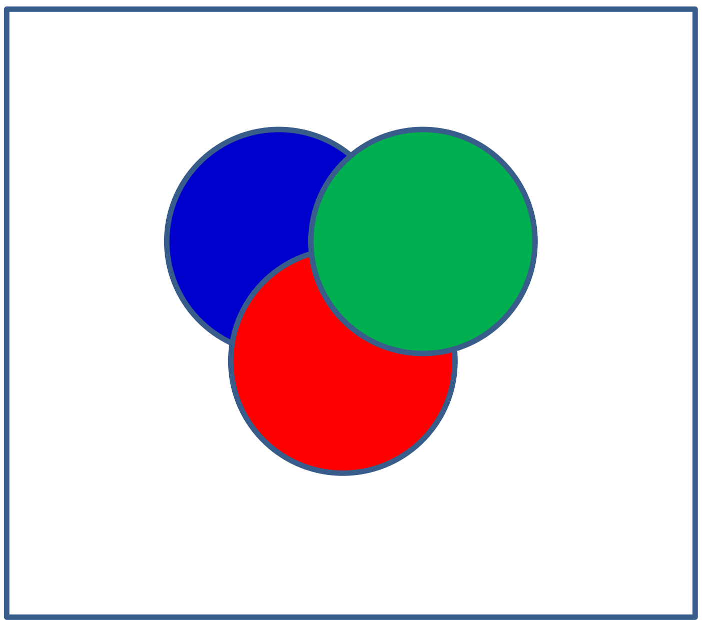
$$S_B^2 = \frac{1}{k-1}\sum_{i=1}^{k} n_i \cdot (\bar{x}_i - \bar{x}_0)^2 \qquad S_W^2 = \frac{1}{n-k}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2$$

Statistical Methods for Evaluation

- ANOVA (Analysis of Variance)



$$F = \frac{S_B^2}{S_W^2}$$



$$S_B^2 = \frac{1}{k-1}\sum_{i=1}^{k} n_i \cdot (\overline{x}_i - \overline{x}_0)^2 \qquad S_W^2 = \frac{1}{n-k}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij} - \overline{x}_i)^2$$

Statistical Methods for Evaluation

- ANOVA (Analysis of Variance)
  - Measures how different the means are relative to the variability within each group
  - The larger the *F*-test statistic, the greater the likelihood that the difference of means are due to something other than chance alone
  - The *F*-test statistic follows an [*F*-distribution]

$$F = \frac{S_B^2}{S_W^2}$$

Statistical Methods for Evaluation

- ANOVA (Analysis of Variance)

```
# fit ANOVA test
model <- aov(purchase_amt ~ offers, data=offertest)

summary(model)
             Df Sum Sq Mean Sq F value Pr(>F)
offers        2 225222  112611   130.6 <2e-16 ***
Residuals   497 428470     862
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Shall we accept or reject the null hypothesis?

Statistical Methods for Evaluation

- ANOVA (Analysis of Variance)
  - Additional tests for each pair of groups
  - Tukey's Honest Significant Difference (HSD)

```
TukeyHSD(model)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = purchase_amt ~ offers, data = offertest)

$offers
                     diff        lwr      upr     p adj
offer1-nopromo 40.961437 33.4638483 48.45903 0.0000000
offer2-nopromo 48.120286 40.5189446 55.72163 0.0000000
offer2-offer1   7.158849 -0.4315769 14.74928 0.0692895
```

Recap: Data Analytic Methods Using R

- Introduction to R
  - R, RStudio, Data I/O, Attribute and Data Types
  - Descriptive statistics
- Exploratory Data Analysis
  - Visualization before analysis
  - Visualizing single or multiple variables
- Statistical Methods for Evaluation
  - Hypothesis Testing, ANOVA

Q&A

Images Courtesy of Google Image