# BIG DATA BOOTCAMP

Atlanta, GA.

OCTOBER 7th, 8th & 9th 2016

Georgia World Congress Center, 285 Andrew Young International Blvd NW, Atlanta, GA 30303.

**www.globalbigdataconference.com**

**Twitter : @bigdataconf**

# Introductory BootCamp: Turning Raw Data into a Useful Working Predictive Model

Mark Jack

*Summary*

**Frank Hasbani PMP, CSM & Klimka Szwaykowska PhD & Mark A. Jack PhD**

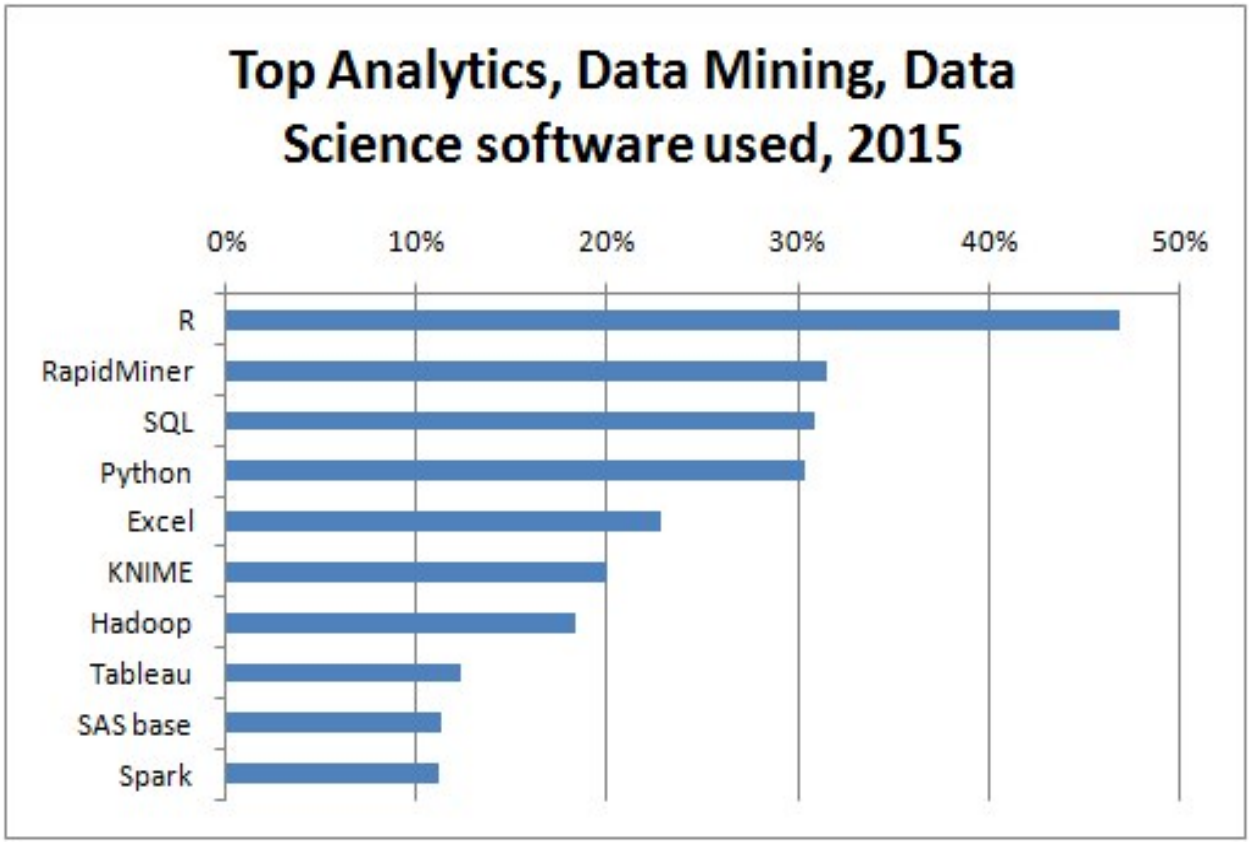
**Anova Analytics LLC, Roswell, GA – May-July 2016**

**Anovaanalytics.com**

*Topics -*

- ***Introduction to Data Science and R***
- Getting and Cleaning the Data
- Data Munging and Data Wrangling
- Visualization
- Statistical Inference
- Machine Learning – Regression and Classification
- Model Building and Estimation
- Final Project

# Introduction to Data Science and R Studio




2800 voters. The participation by region was: US/Canada (41.5%), Europe (38.4%), Asia (8.2%), Latin America (6.3%), Australia/NZ (3.1%), Africa/MidEast (2.5%).

***http://www.kdnuggets.com/2015/05/poll-r-rapidminer-python-big-data-spark.html***

# What is R?

- A free software programming language and software environment for statistical computing and graphics.
- Dialect of the S programming language with lexical scoping semantic inspired by *Scheme*.
- Provides linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, and more...
- Cross-platform: Windows, Linux, MAC OSX.
- **Hadley Wickham: Advanced R,** http://adv-r.had.co.nz
- **Grolemund & Wickham: R for Data Science,** http://r4ds.had.co.nz
- ***RStudio IDE: https://www.rstudio.com***

# Packages

- Packages extend functionality of R.

  http://cran.r-project.org/web/packages

  8825 available packages as of July 25, 2016.

- repository → installed → loaded

  ***library(help="package")***

- Datasets:

  ***data(mtcars); help(mtcars)***

- Example: Library for support vector machines (libsvm)

  ***install.packages("e1071")***

  ***library(e1071)***

  ***detach("package:e1071")***

# Getting and Cleaning the Data

# Getting and Cleaning Data Content

- Downloading data files
- Reading data
- Data formats: Excel, XML, JSON, MySQL, HDF5, Web, ...
- Merging data
- Reshaping data
- Summarizing data
- Finding and replacing data
- Data resources

# Reading Data

There are a few principal functions to read data into R:

- `*read.table*`, `*read.csv*`, for reading tabular data
- `*readLines*`, for reading lines of a text file
- `*source*`, for reading in R code files (`inverse` of `dump`)
- `*dget*`, for reading in R code files (`inverse` of `dput`)
- `*load*`, for reading in saved workspaces
- `*unserialize*`, for reading single R objects in binary form

# Data Munging and Data Wrangling

# Data Wrangling/Munging/Manipulation

- Raw vs. tidy data
- Subsetting
- Regular expression
- Ordering
- Summarizing
- Reshaping
- Merging
- Editing

# Data Wrangling/Munging/Manipulation

***R Studio Packages:***

***plyr -*** "split-apply-combine" approach to R data.

***dplyr -*** a faster and more powerful version of plyr.

***shiny*** - interactive graphical websites.

***data.table -*** a much faster version of data.frame, very well suited for larger datasets, with powerful split+apply, merge, etc. functionality; alternative to plyr.

***likert -*** visualize the results with likert-style questionnaire data.

# Data Wrangling/Munging/Manipulation

***'dplyr' functionality:***

- Five basic verbs:

  ***filter, select, arrange, mutate, summarise (+ group_by)***

- Can work with data stored in databases and data tables
- Inner join, left join, semi-join, anti-join
- Window functions for calculating ranking, offsets, and more
- Better than '***plyr***' if you're only working with data frames (though it doesn't yet duplicate all of the plyr functionality)

# Data Wrangling/Munging/Manipulation

***Example:***

Use of ***dplyr*** commands on *'hflights'* data set to subset, re-organize and manipulate data frame and compute aggregated results.
File: ***dplyrTraining.R***

Use of ***dplyr*** commands on *'mtcars'* data set
File: ***dplyrTraining2.R***

# Visualization

# Plotting Systems

Three different plotting systems:

- *base graphics*
- *lattice graphics*
- *graphical processing language (GPL)*

R's base graphics plotting command: **plot()**

A '*grammar of graphics*' - New graphing library ***ggplot2:***

- An abstraction which makes thinking, reasoning and communicating graphics easier.
- Developed by Leland Wilkinson ("The Grammar of Graphics", 1999/2005).
- High-level package for creating statistical graphics.
  A rich set of components + user friendly wrappers.

# Grammar of Graphics - ggplot2 library

Library ***ggplot2 (http://had.co.nz/ggplot2):***

- Two basic methods to create plots, **qplot()** and **ggplot().**
- **qplot():** different set of assumptions and default settings to speed the creating of graphs (lattice graphics)
- **ggplot():** layered plots with different aesthetics, facets, scales and axes, and data sources (GPL)
- Basic command line calls:

```
qplot(x, y, data = data)
ggplot(data, aes(x, y)) + geom_point()
```

# Graphical Processing Language - ggplot

***ggplot():***

***Components:***

- Data
- Geometric object (geom)
- Statistical transformation (stat)
- Scales
- Coordinate system

(+ Position adjustment, facetting)

***Plot definition:***

```
ggplot(data, mapping) +
layer(
stat = “”,
geom = “”,
position = “”,
geom_params = list(),
stat_params = list(),
)
```

# Examples – plot() and ggplot()

Plotting with base graphics in R using ‘plot()’:

File: ***baseplotting.R***

Plotting with ‘ggplot()’:

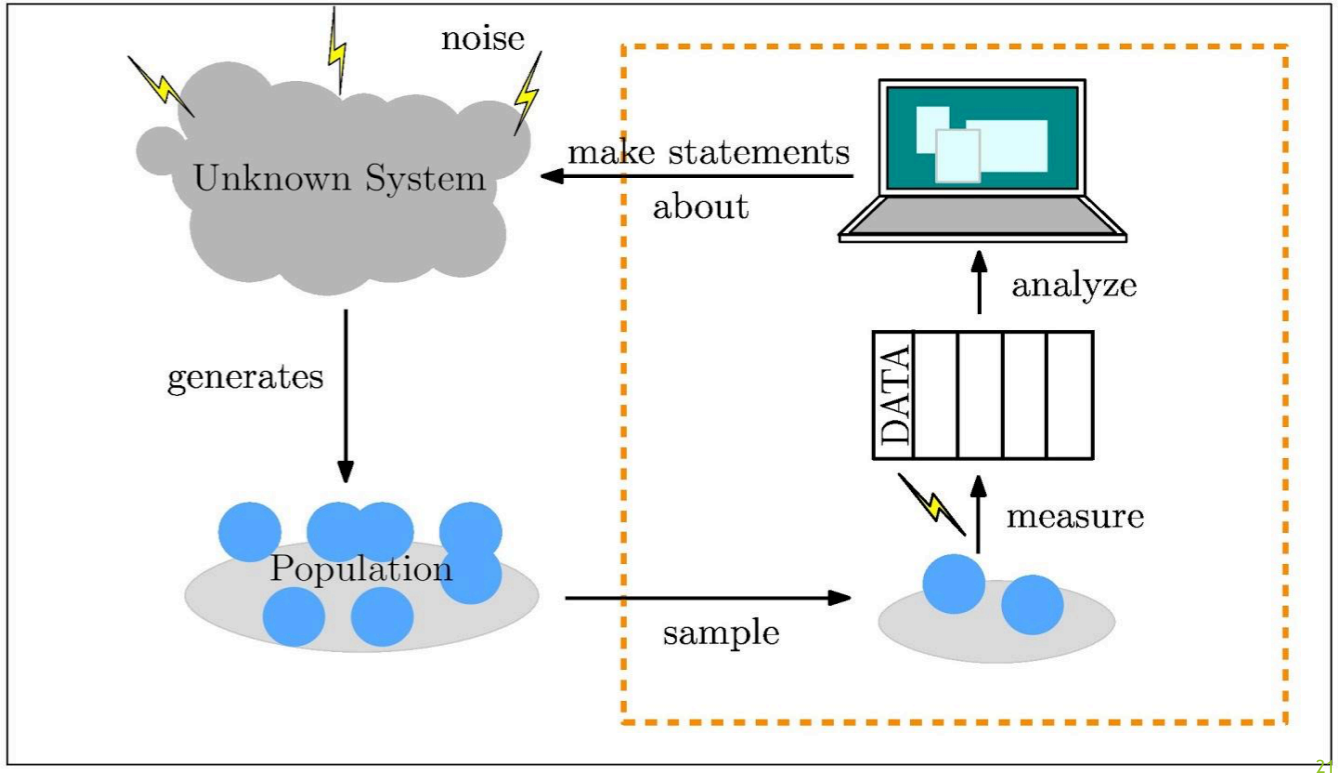
File: ***ggplot2.R***

R Studio’s ggplot2 cheat sheet:

File: ***ggplot2-cheatsheet.pdf***

# Statistical Inference

# What is Statistical Inference



# Statistical Hypothesis Testing

- Can be used to compare two statistical data sets, or to compare a data set obtained by sampling with synthetic data from an idealized model.

- Hypothesis testing allows us to distinguish--in a statistically meaningful way, between the null hypothesis and an alternative that we propose to test.

- **Example:** We want to test whether singing to a Lima bean makes it grow faster.
  - Null hypothesis: the mean heights of bean sprouts after 10 days are equal for beans that have been sung to, and a non-musical control group.
  - Alternate hypothesis: The mean height of the beans in the sung-to group is larger than that of the control group.

# Example: Coin Flip

Say we flip a coin 100 times and end up with the following results:

- The coin lands HEADS 66 times.
- The coin lands TAILS 34 times.

Can we conclude that the coin is biased?

Here, the **null hypothesis** is that we have a fair coin:

$$p(HEADS) = p(TAILS) = ½$$

And the alternate hypothesis is that the coin is **not** fair.

# Example: Coin Flip

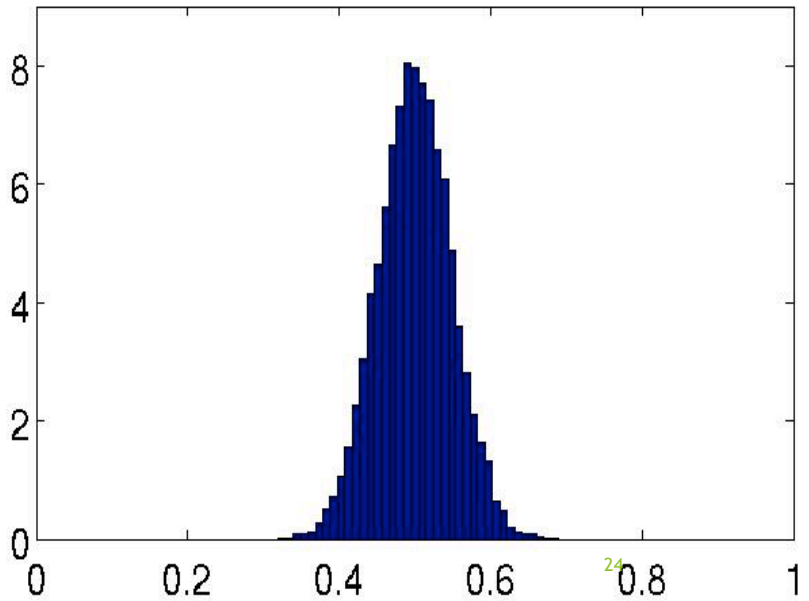
Given a known fair coin, we run 10,000 experiments of 100 coin flips. We record HEADS as 0 and TAILS as 1, and find the mean value for each experiment:

Experiment 1:
44 HEADS, 56 TAILS - mean = 0.56

Experiment 2:
62 HEADS, 38 TAILS - mean = 0.38

Experiment 3:
56 HEADS, 44 TAILS - mean = 0.44

Experiment 4:
45 HEADS, 55 TAILS - mean = 0.55



# Example: Coin Flip

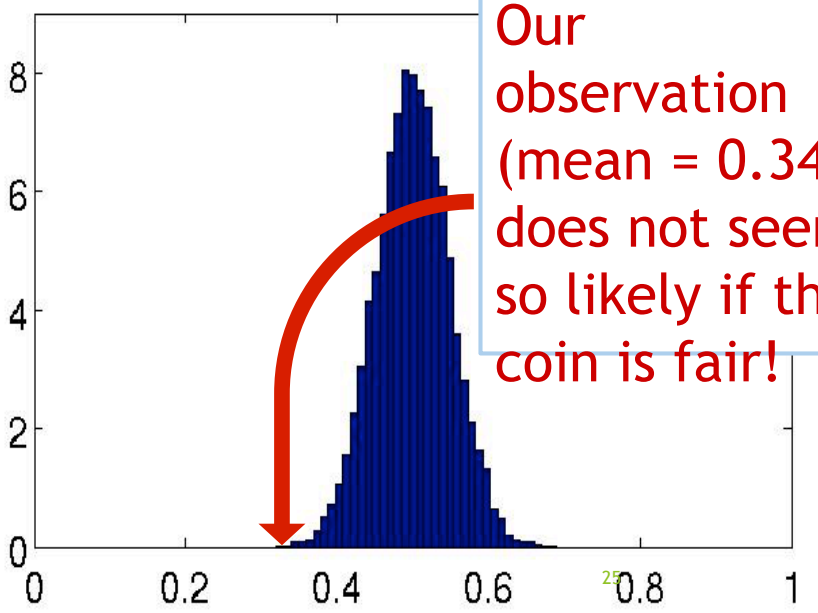
Given a known fair coin, we run 10,000 experiments of 100 coin flips. We record HEADS as 0 and TAILS as 1, and find the mean value for each experiment:

Experiment 1:
44 HEADS, 56 TAILS - mean = 0.56

Experiment 2:
62 HEADS, 38 TAILS - mean = 0.38

Experiment 3:
56 HEADS, 44 TAILS - mean = 0.44

Experiment 4:
45 HEADS, 55 TAILS - mean = 0.55




Our observation (mean = 0.34) does not seem so likely if the coin is fair!

# Machine Learning - Regression and Classification

# Machine Learning

- What is Prediction?

  Sampling from a probability distribution

  => Create training data set with a prediction function

- Components:
  - Question
  - Input Data = Training data
  - Features - predictor and outcome variables
  - Algorithm with tunable parameters

    ***(e.g. regression coefficients)***
  - Evaluation – test data set, validation data set

# Machine Learning

Functions for prediction study in the "caret" library:

- Preprocessing: *preProcess()*
- Data splitting: *createDataPartition(), createResample(), createTimeSlices()*
- Do training/testing: *train()*
- Run prediction algorithm: *predict()*
- Provide statistics / error measurements for model comparison: *confusionMatrix()*

# Preprocessing Your Data

For best performance of machine-learning algorithms, data needs to be prepared in a specific form, i.e. pre-processed:

- Instance based methods more effective if input variables have the same scale.
- Regression methods work better if input variables are standardized.

Use "caret" package in R:

Standalone: Pre-process multiple data sets.

Use functions *preProcess()* and *predict()*.

Training: Process training data during model evaluation.

Use functions *preProcess()* and *train()*.

# Preprocessing Your Data

**Examples:**

- ***zv:*** remove attributes with a zero variance (all the same value).
- ***nzv:*** remove attributes with a near zero variance (close to the same value).

- ***scale:*** divide values by standard deviation.
- ***center:*** subtract mean from values.
- ***range:*** normalize values.

- ***pca:*** transform data to the principal components.
- ***ica:*** transform data to the independent components.
- Other transforms: ***BoxCox***, ***YeoJohnson***, ***expoTrans***, etc.

# Multiple Linear Regression and Interpreting Regression Coefficients

Model to estimate outcome y by multiple predictor variables $x_i$:

Estimated value $\hat{y} = C_0 + C_1 x_1 + \ldots + C_p x_p$

$C_i$: regression coefficient for effect of $x_i$ on y with all other predictors $x_j$ held fixed $(j \neq i)$

Issues:

- Variables $x_i$ are typically correlated. Thus, changing one will also change the impact of the other on the outcome y.
- Causality is not always clear, i.e. which variable $x_i$ causes the other to change.

# Machine Learning
## - Regression and Hypothesis Testing

You can use hypothesis testing to test

null hypothesis $H_0$ – there is **NO** relationship between outcome y and predictor x (i.e. $C_1 = 0$ )

vs.

alternative hypothesis $H_A$ - there is a relationship between y and x (i.e. $C_1 \neq 0$ )

# Examples - Linear Regression

***Example 1 - Linear model for 'Old Faithful' eruptions data***

```
library(caret); data(faithful); set.seed(333)
inTrain <- createDataPartition(y=faithful$waiting, p=0.5, list=FALSE)
trainFaith <- faithful[inTrain, ]; testFaith <- faithful[!inTrain, ]
head(trainFaith)
plot(trainFaith$waiting, trainFaith$eruptions, pch=19, col="blue",
xlab="Waiting", ylab="Duration")
lm1 <- lm(eruptions ~ waiting, data=trainFaith)
summary(lm1)
plot(trainFaith$waiting, trainFaith$eruptions, pch=19, col="blue",
xlab="Waiting", ylab="Duration")
lines(trainFaith$waiting, lm1$fitted, lwd=3)
```

***Example 2 - File LinearRegressionLab.R***

# Classification

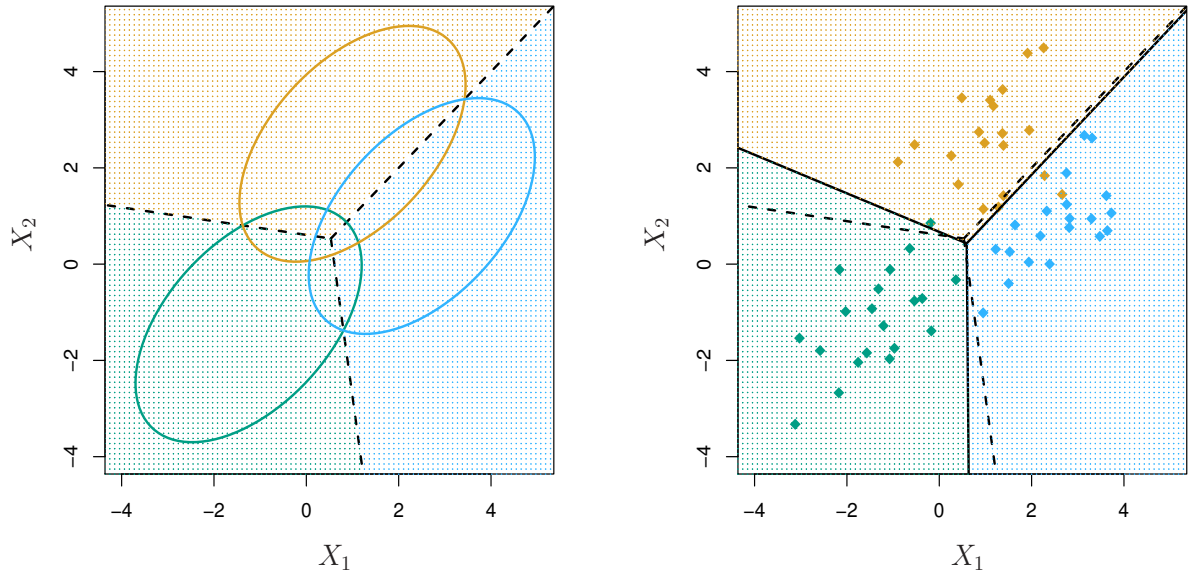
You can use Linear Regression (in principle) for classification:

- The outcome Y is binary (there are only two categories/ classes),

$$Y = \begin{cases} 0 & \text{if } X \text{ does not belong to the class} \\ 1 & \text{if } X \text{ belongs to the category} \end{cases}$$

- Make a prediction based on a threshold for the classification function, e.g. if $\hat{Y} = C_0 + \sum_i C_i x_i > 0.5$ then predict Y=1, otherwise Y=0.
- This is equivalent to linear discriminant analysis.
- Con: Probabilities may come out larger than 1 or negative.
- Alternative: Logistic regression.

# Classification
## - Linear Discriminant Analysis & Bayes Theorem



# Model Building and Estimation

# Testing and Tuning Predictive Algorithms

Testing Predictive Algorithms:

Best Algorithm for a Problem – Algorithms to Test in R

Comparing Performance of Predictive Algorithms:

- ✓ Prepare Dataset
- ✓ Train Models
- ✓ Compare Models
- ✓ Choose the Best Model

Tuning Predictive Algorithms:

- ✓ Get Better Accuracy from Top Algorithms
- ✓ Tune Predictive Algorithms
- ✓ Test Setup
- ✓ Tune Using Caret
- ✓ Craft Your Own Parameter Search

# Improving upon Models

Resampling Methods To Estimate Model Accuracy:

- ✓ Estimating Model Accuracy
- ✓ Data Split – Training Set, Test Set and Validation Set
- ✓ Bootstrap
- ✓ k-fold Cross Validation
- ✓ Repeated k-fold Cross Validation
- ✓ Leave-One-Out Cross Validation
- ✓ Tips for Evaluating Algorithms

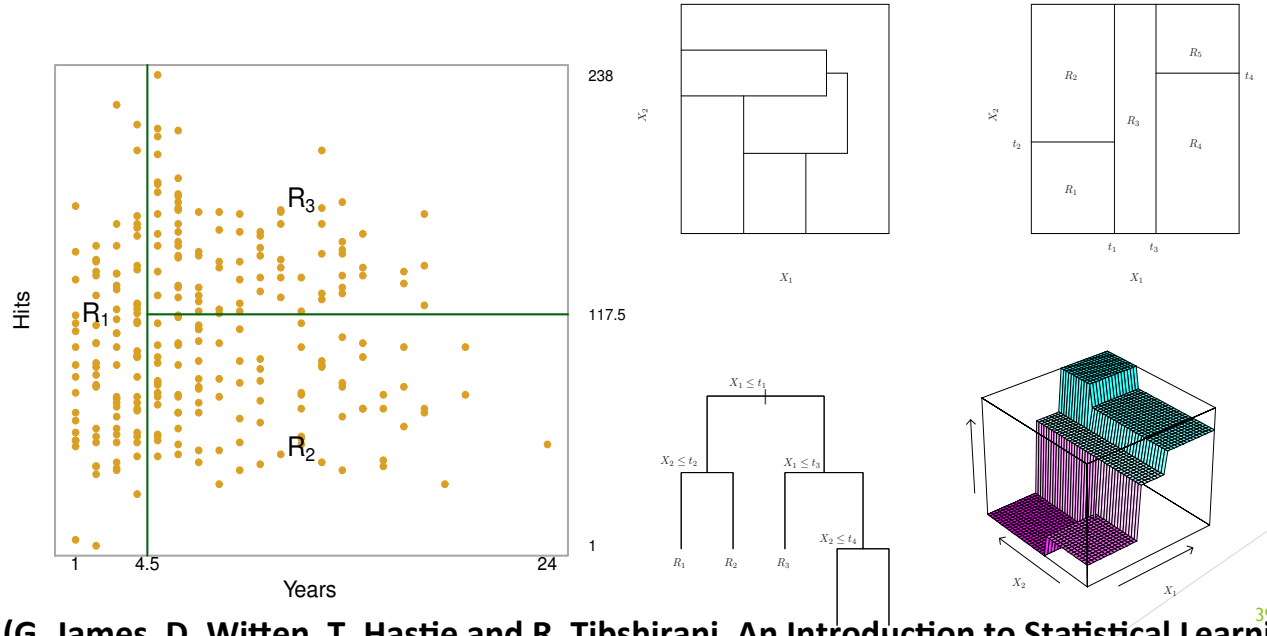
Combine Predictions from Multiple Models:

- ✓ Increase the Accuracy Of Your Models
- ✓ Ensemble Methods
- ✓ Boosting Algorithms
- ✓ Bagging Algorithms
- ✓ Stacking Algorithms

# Decision Trees, Bagging and Random Forests

***Example:***

Salaries of MLB baseball players based on number of hits and years played in the league.




**(G. James, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning with Applications in R, http://www-bcf.usc.edu/~gareth/ISL)**

# Decision Trees, Bagging and Random Forests

***Bagging (bootstrap aggregation)***

- bootstrap tree-based model e.g. random forests

Idea:

Take a set of independent observations $Z_1, \ldots, Z_n$ with variance $\sigma^2$ and then take their average, and thus reduce variance by n.

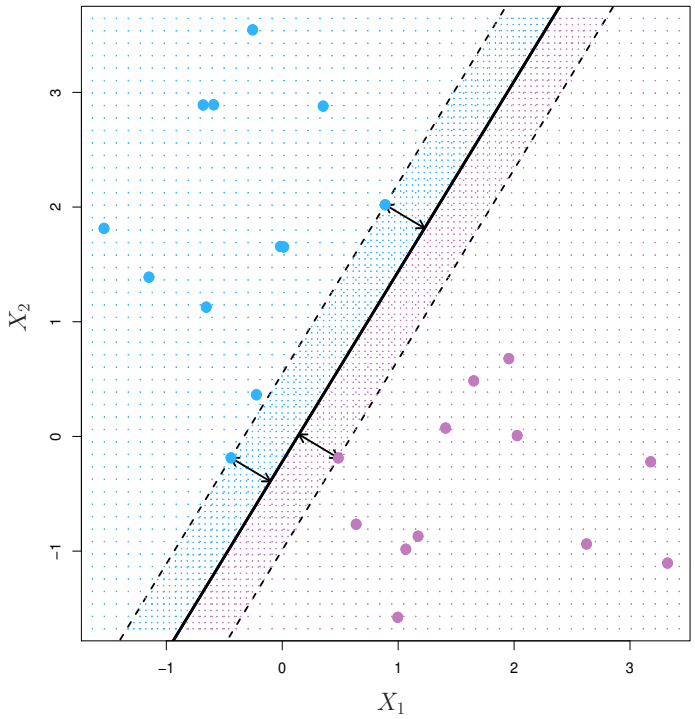
The variance of the mean $\bar{Z}$ is given as $\sigma^2/n$.

=> Take bootstrap samples and create a decision tree for each sample.

Take average predictions from all bootstrap samples = *'bagging'*.

# Support Vector Machines

***Which classifier do you pick? Maximum margin classifier (hyperplane) (function svm() in e1071 library)*** **(James et al., http://www-bcf.usc.edu/~gareth/ISL)**




Constrained optimization problem:

$$\underset{\{(\beta_0,\beta_1,\ldots,\beta_p)\}}{\text{maximize}}\ M$$

with $y_i\left(\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\ldots+\beta_p X_{ip}\right)\geq M$

subject to $\sum_{j=1}^{p}\beta_j^2=1$

for all points $i = 1,\ldots,N$

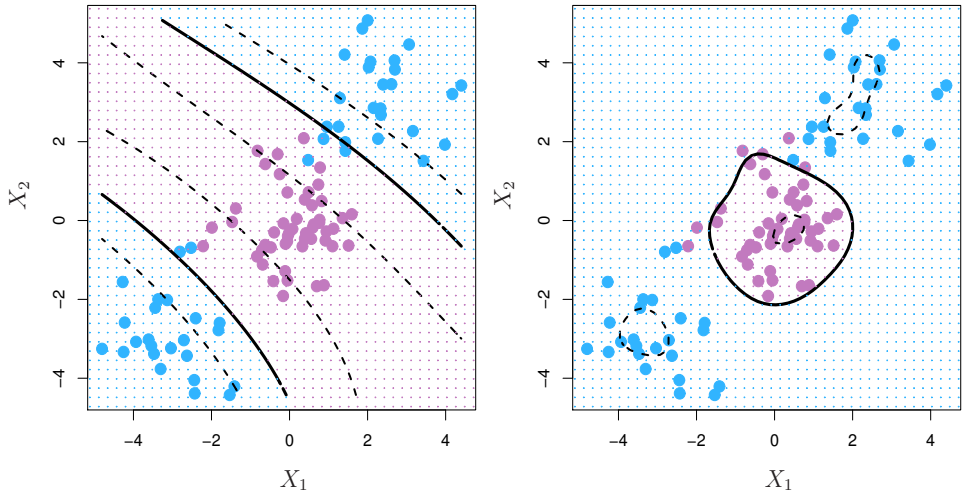
# Support Vector Machines

***Problems:***

1. Non-separable data: Typically if *N* > *p,* you might not be able to find a classifier.
2. Noisy data where a classifier would change dramatically due to an outlier.

-> '***Soft margins***' with some points on the wrong side of the margin or decision boundary.

-> ***Convex optimization problem:*** **Nonlinear decision boundaries** using a classifier in an enlargened feature space by including interaction terms $X_1^2, X_1X_2, X_2^3, \ldots$




***Classifiers and kernel functions***

***Example: radial classifier***

***(svm() in e1071)***

**(James et al., http://www-bcf.usc.edu/~gareth/ISL)**

# Final Project

# Final Project of Participants

***Examples:***

Regression model for analysis of 'mtcars' data set:

Files: - R Markdown: ***RegressionModelProject.Rmd***

- HTML output via ***knitr*** (integrated feature in R Studio):

***RegressionModelProject.html***

Model estimation for 'fitbit' human motion data:

File – ***run_analysis_ML.R***

# What Else is Out There?

- ***Presentations & Publications & Interactive Web Applications:***
  - ✓ **RMarkdown**– Create publications with integrated Latex formulas and executable R code chunks and keep it with your data: http://rmarkdown.rstudio.com
  - ✓ **Rpubs** – Make documents publicly available: https://rpubs.com
  - ✓ **Rpres** – Create interactive presentations with R code snippets: https://rpubs.com/mjgrav2001/mtcars-presentation
  - ✓ **Shinyapps** – Create interactive web applications: http://shiny.rstudio.com/gallery
- ***Big Data:*** **SparkR** – Deploy predictive analytics tools for Big Data: https://spark.apache.org
- ***Workflow Management:*** Example: KNIME on Microsoft Azure: http://knime.org

Thank You!

Questions ???

# Resources:

- Quora: List of machine learning resources

  https://www.quora.com/How-do-I-learn-machine-learning-1
- Science: List of machine learning resources
  http://www.sciencemag.org/site/feature/data/compsci/machine_learning.xhtml
- Andrew NG (Stanford, Coursera)

  https://www.coursera.org/learn/machine-learning/
- MIT Open Courseware
  http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/lecture-notes/
- Kaggle - machine learning competitions

  https://www.kaggle.com/

# Resources:

- Cheat Sheet for Linear Regression and Classification in R: http://blog.revolutionanalytics.com/2012/08/cheat-sheet-for-prediction-and-classification-models-in-r.html
- Linear classification in R: http://machinelearningmastery.com/linear-classification-in-r/
- Classification tree models in R: http://www.r-bloggers.com/classification-tree-models/
- Tbschirani & Hastie: Elements of Statistical Learning (book).

  R Library: ISLR.
- "caret" package: http://caret.r-forge.r-project.org/