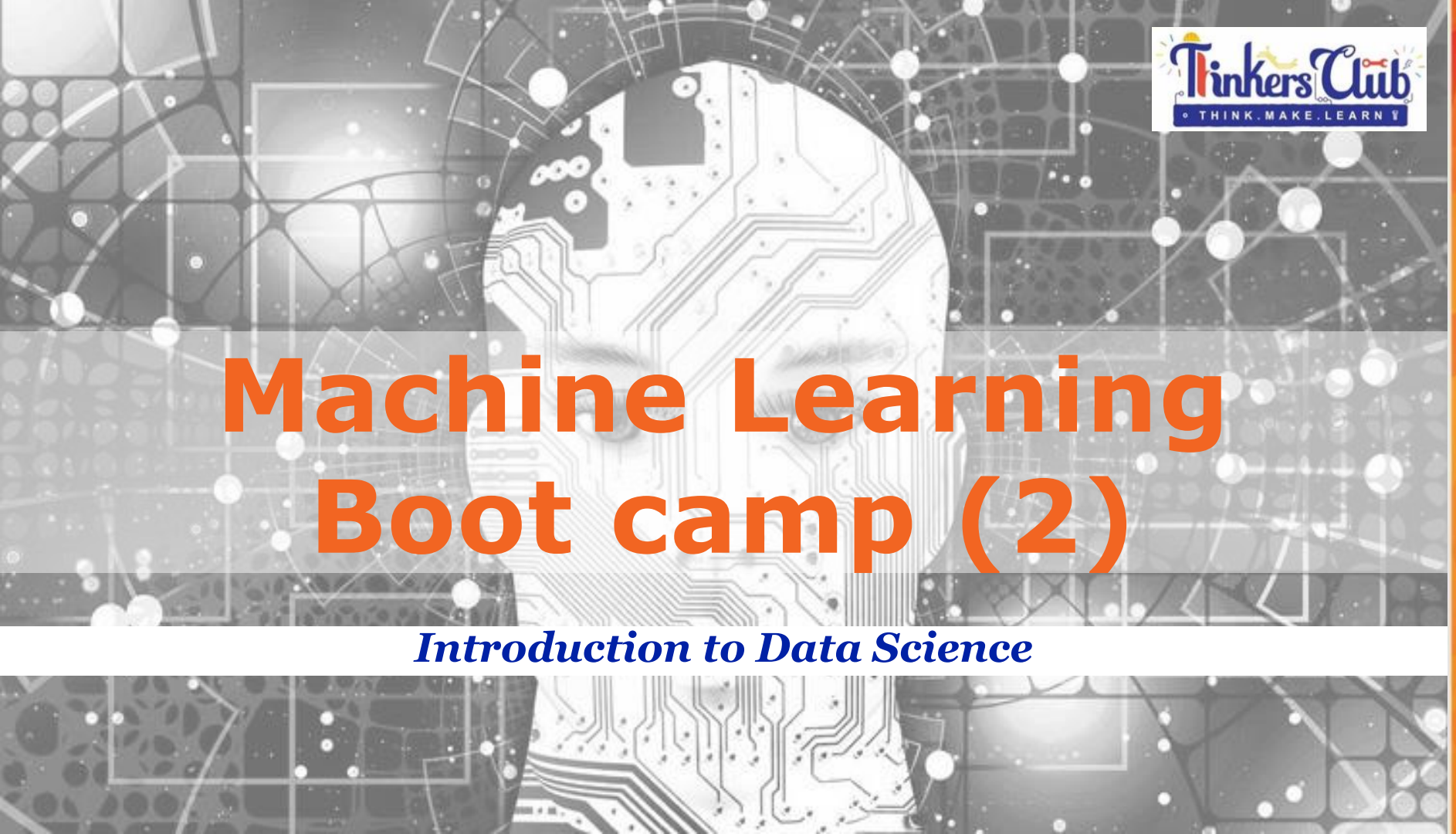
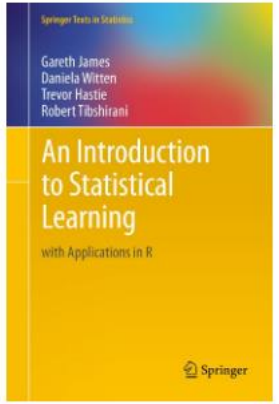


Machine Learning Boot camp (2)

Introduction to Data Science

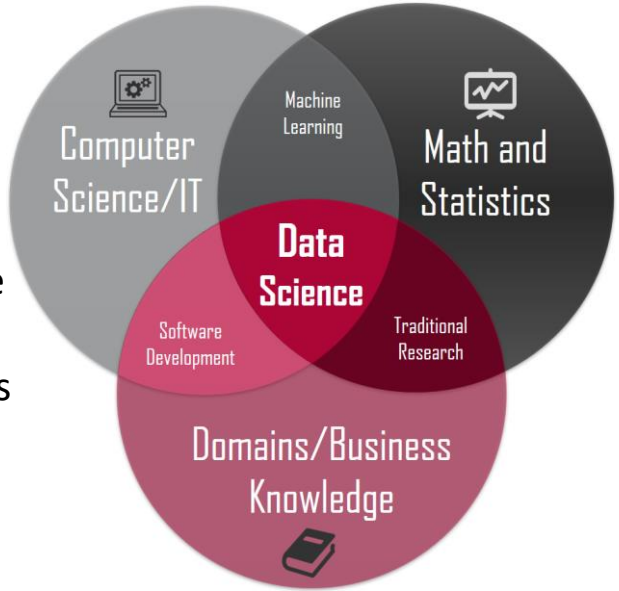
Machine Learning



http://faculty.marshall.usc.edu/gareth-james/ISL/

What is Machine Learning?

- Machine learning is essentially a form of applied statistics.
  - Machine learning is statistics scaled up to big data.
- The major difference between machine learning and statistics is their purpose.
  - Machine learning models are designed to make the most accurate predictions possible. Statistical models are designed for inference about the relationships between variables.
  - To make this slightly more explicit, there are lots of statistical models that can make predictions, but predictive accuracy is not their strength.
  - Likewise, machine learning models provide various degrees of interpretability, from the highly interpretable *lasso regression* to impenetrable *neural networks*, but they generally sacrifice interpretability for predictive power.

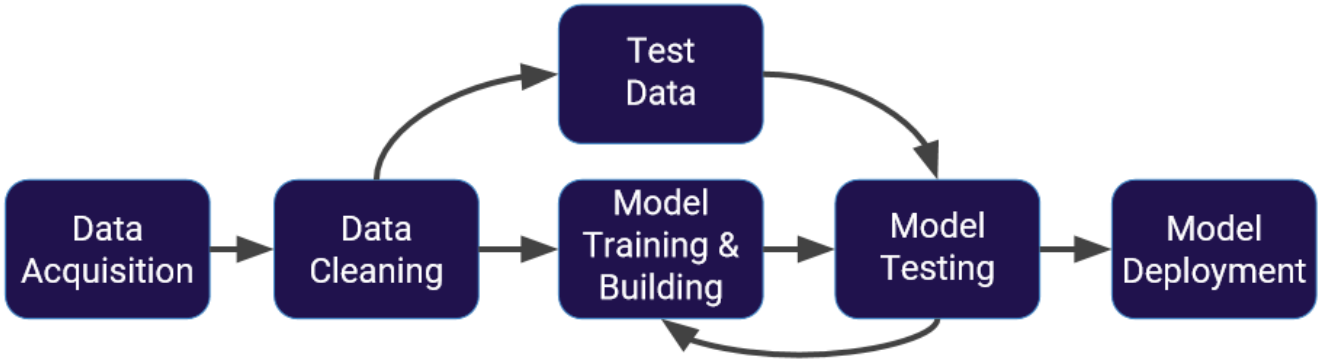


What is Machine Learning?

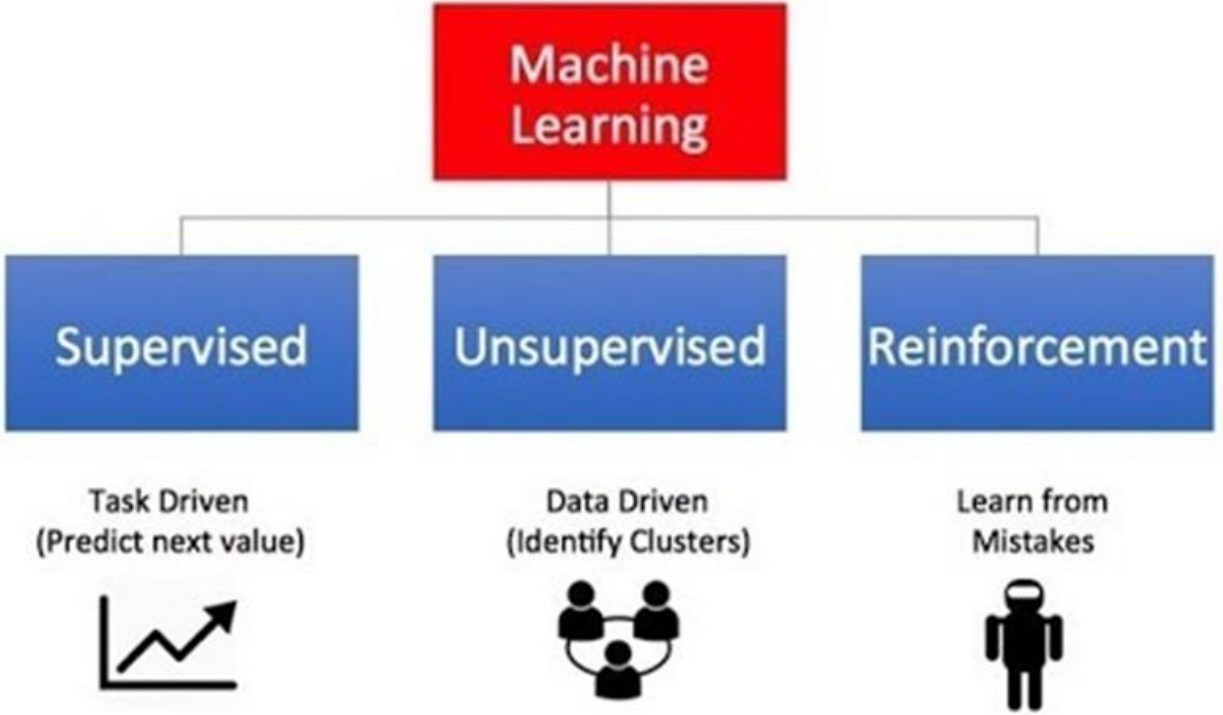
- Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look.
- What is it used for?
- ● Fraud detection. ● Web search results. ● Real-time ads on web pages ● Credit scoring and next-best offers. ● Prediction of equipment failures. ● New pricing models. ● Network intrusion detection. ● Recommendation Engines ● Customer Segmentation ● Text Sentiment Analysis ● Predicting Customer Churn ● Pattern and image recognition. ● Email spam filtering. ● Financial Modeling.

Artificial intelligence is a bigger concept to create **intelligent machines** that can simulate human thinking capability and behavior, whereas, **machine learning** is an application or subset of **AI** that allows machines to learn from data without being programmed explicitly.

Machine Learning Process



Machine Learning Types



Supervised Learning

- Supervised learning algorithms are trained using **labeled** examples, such as an input where the desired output is known.
  - For example, a piece of equipment could have data points labeled either “F” (failed) or “R” (runs).
- The learning algorithm receives a set of inputs along with the corresponding correct outputs, and the algorithm learns by comparing its actual output with correct outputs to find errors.
- It then modifies the model accordingly.
- Through methods like *classification* and *regression* etc. supervised learning uses patterns to predict the values of the label on additional unlabeled data. Supervised learning is commonly used in applications where historical data predicts likely future events.

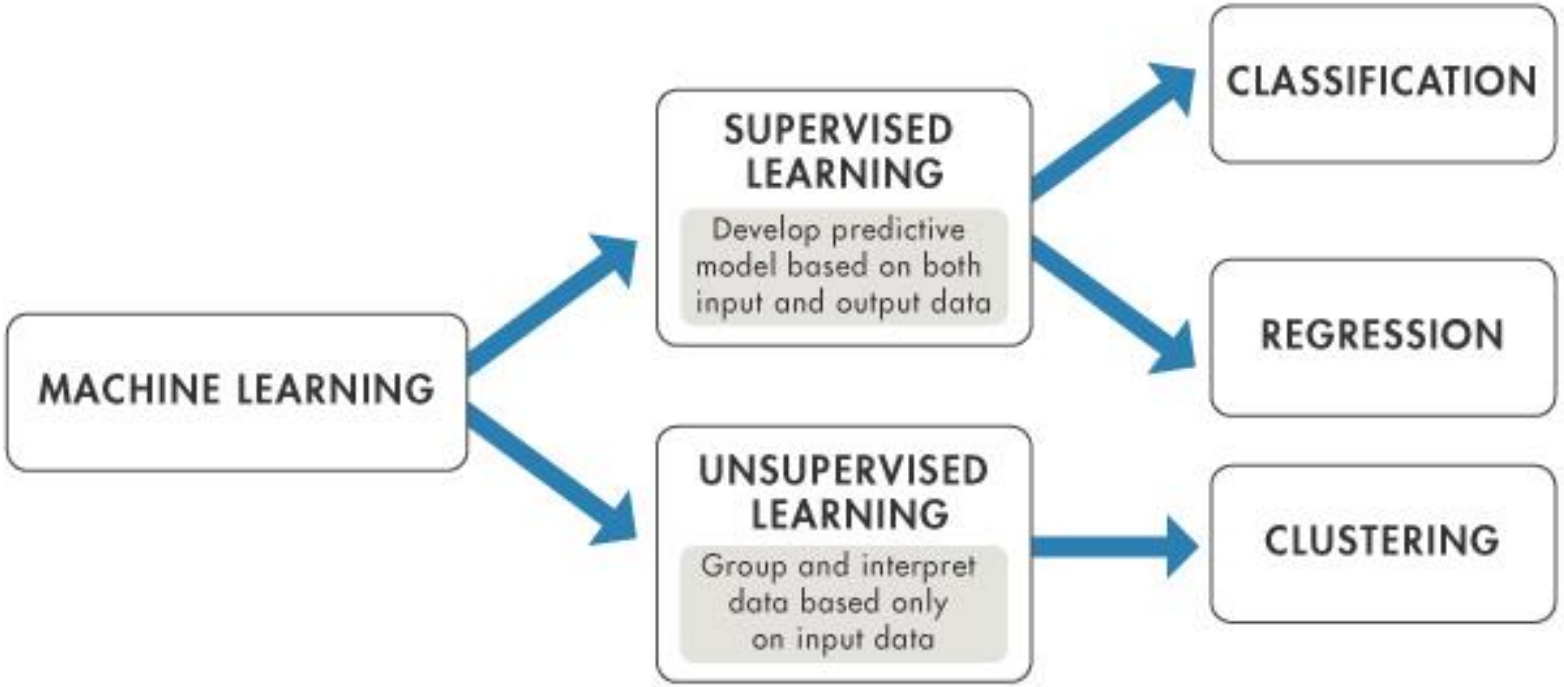
Supervised Learning

- Supervised learning is commonly used in applications where historical data predicts likely future events.
  - For example, it can anticipate when credit card transactions are likely to be fraudulent or which insurance customer is likely to file a claim.
  - Or it can attempt to predict the price of a house based on different features for houses for which we have historical price data.

Unsupervised Learning

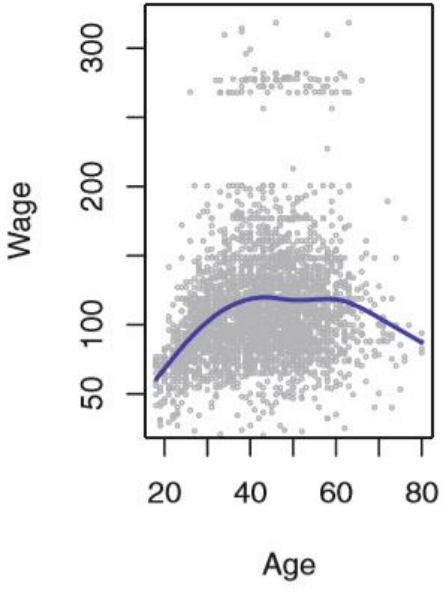
- Unsupervised learning is used against data that has no historical labels.
- The system is not told the "right answer." The algorithm must figure out what is being shown.
- The goal is to explore the data and find some structure within.
  - For example, it can find the main attributes that separate customer segments from each other.
  - These algorithms are also be used to segment recommend items and identify data outliers
- Popular techniques include self-organizing maps, nearest neighbor mapping, *k-means clustering* and singular value decomposition.

Supervised vs. Unsupervised



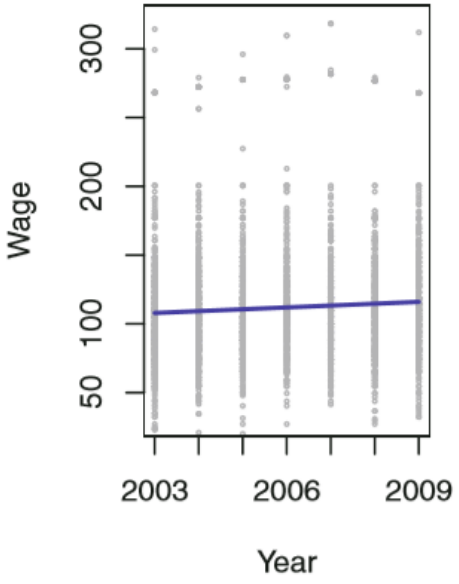
Example 1 – Wage Data

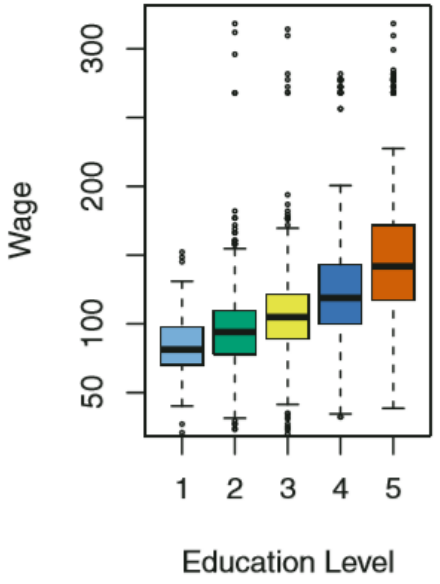
- We wish to understand the association between an **employee's age** and **education**, as well as the **calendar year**, on his **wage**.
  - Generally, **wage** increases with **age** but decreases after approximately age 60. Given an **employee's age**, we can use this curve to predict his **wage**.
  - However, it is also clear from the figure that there is a significant amount of variability associated with the average value.
  - And, so **age** alone is unlikely to provide an accurate prediction of a particular **employee's wage**.



Example 1 – Wage Data

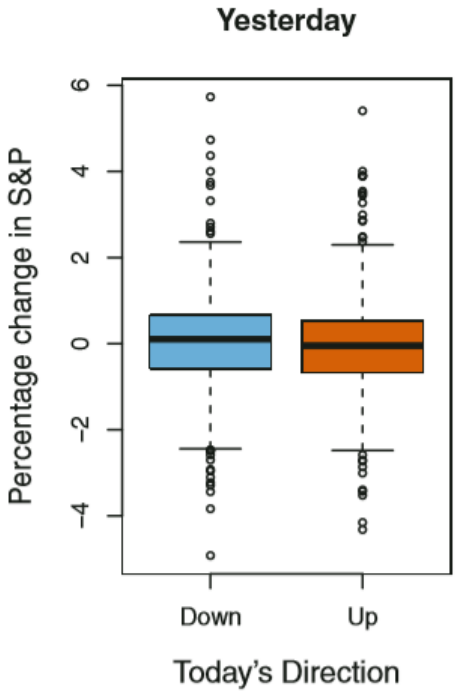
- We also have information regarding each **employee's education** level and the **year** in which the **wage** was earned.
  - Generally, **wages** increased by approximately $10,000 in a roughly linear (or straight-line) fashion, between 2003 and 2009. Though this rise is very slight relative to the variability in the data.
  - **Wages** are also typically greater for individuals with higher **education** levels.
- Clearly, the most accurate prediction of a given employee's **wage** will be obtained by combining his **age**, **education** and the **year**.





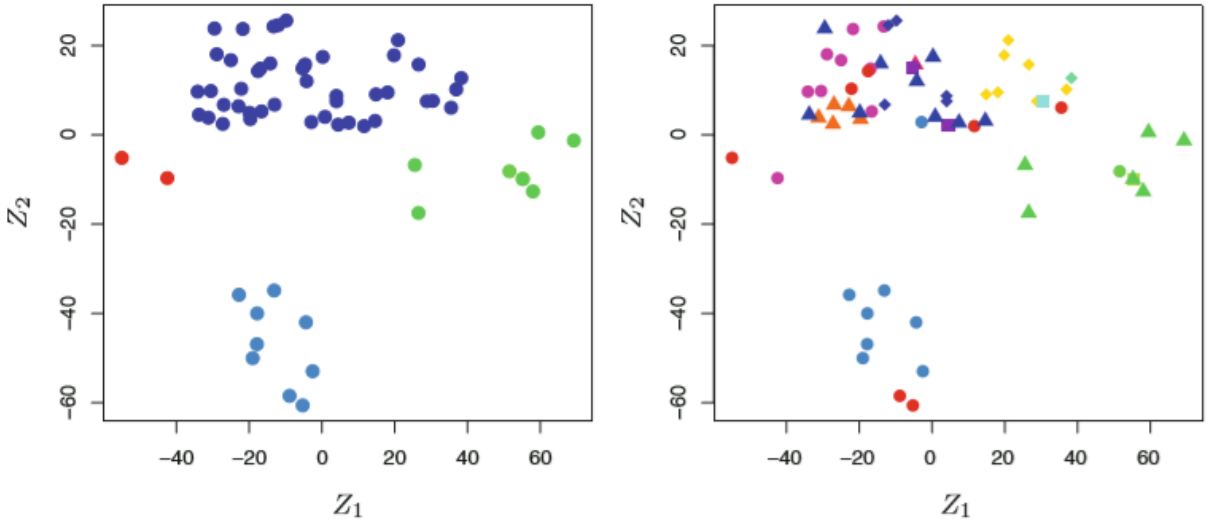
Example 2 – Stock Market Data

- The **wage** data involves a *continuous* or *quantitative* output value. This is often referred to as a *regression* problem.
- However, in certain cases we may instead with to predict a non-numerical value – i.e., a *categorical* or *qualitative* output.
  - For example, we examine a stock market data set that contains the daily movements in the Standard & Poor's 500 stock index over a 5-year period. The goal is to predict whether the index will increase or decrease on a given day – a *classification* problem.
  - A model that could accurately predict the direction of the market would be pretty useful!
  - Figure displays two boxplots of the previous day's % changes in the stock index. 648 days market up, 602 market down. Almost identical plots, suggesting that there is no simple strategy.



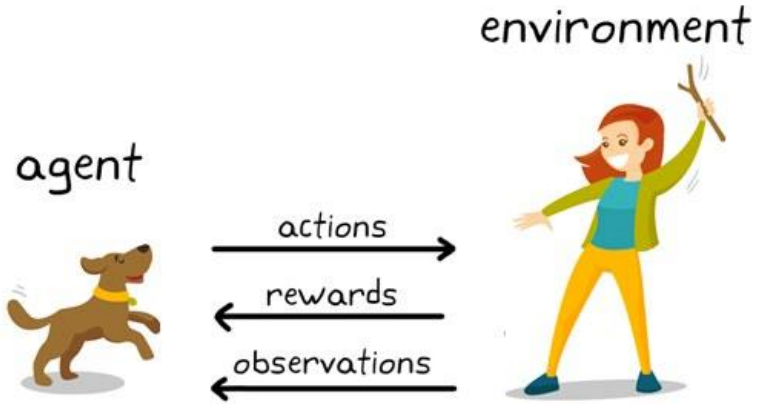
Example 3 – Gene Expression Data

- The previous 2 examples illustrate data sets with both input and output variables.
- *Clustering* problem – grouping according to the observed characteristics.
  - Our goal is to better understand the relationship between gene expression levels and cancer.
  - We examine observations within each cluster for similarities in their types of cancer.
  - *Left*: At least 4 clusters.
  - *Right*: Same as left except that the 14 distinct cancer types are shown.



Reinforcement Learning

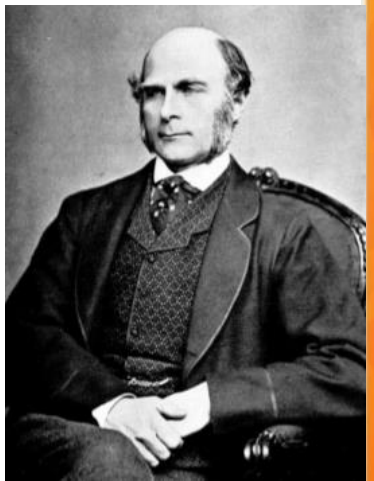
- Reinforcement learning is often used for robotics, gaming and navigation.
- With reinforcement learning, the algorithm discovers through trial and error which actions yield the greatest rewards.
- This type of learning has three primary components: the agent (the learner or decision maker), the environment (everything the agent interacts with) and actions (what the agent can do).
    - The objective is for the agent to choose actions that maximize the expected reward over a given amount of time.
    - The agent will reach the goal much faster by following a good policy. So the goal in reinforcement learning is to learn the best policy.



Supervised Learning

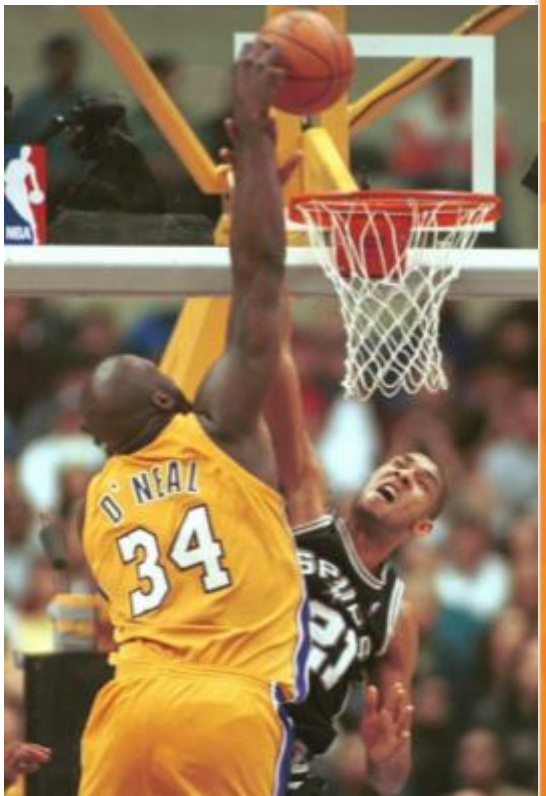
History

- This all started in the 1800s with a guy named Francis Galton.
- Galton was studying the relationship between parents and their children. In particular, he investigated the relationship between the heights of fathers and their sons.
    - What he discovered was that a man's son tended to be roughly as tall as his father. However Galton's breakthrough was that the son's height tended to be closer to the overall average height of all people.



Example

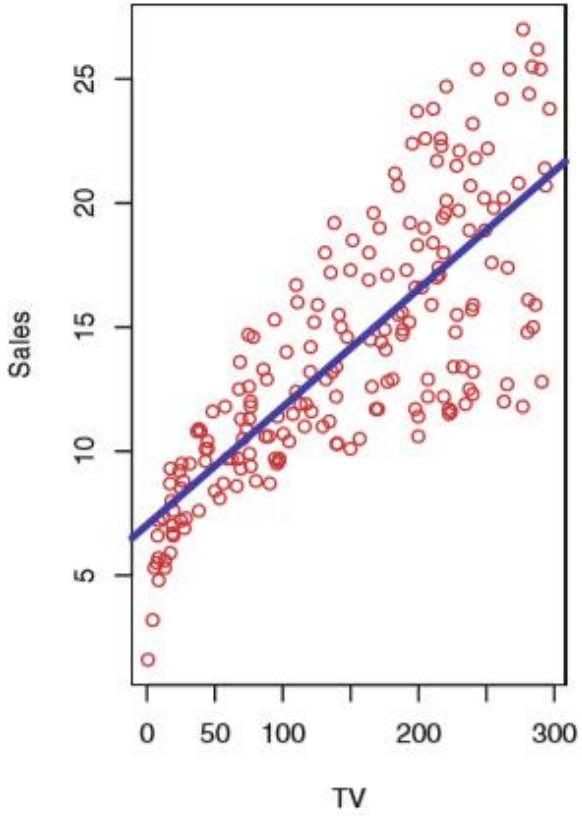
- Let's take Shaquille O'Neal as an example.
- Shaq is really tall:7ft 1in (2.2 meters). If Shaq has a son, chances are he'll be pretty tall too.
- However, Shaq is such an anomaly that there is also a very good chance that his son will be not be as tall as Shaq.
- Turns out this is the case: Shaq's son is pretty tall (6 ft 7 in), but not nearly as tall as his dad. Galton called this phenomenon **regression**, as in
    - A father's son's height tends to regress (or drift towards) the mean (average) height.

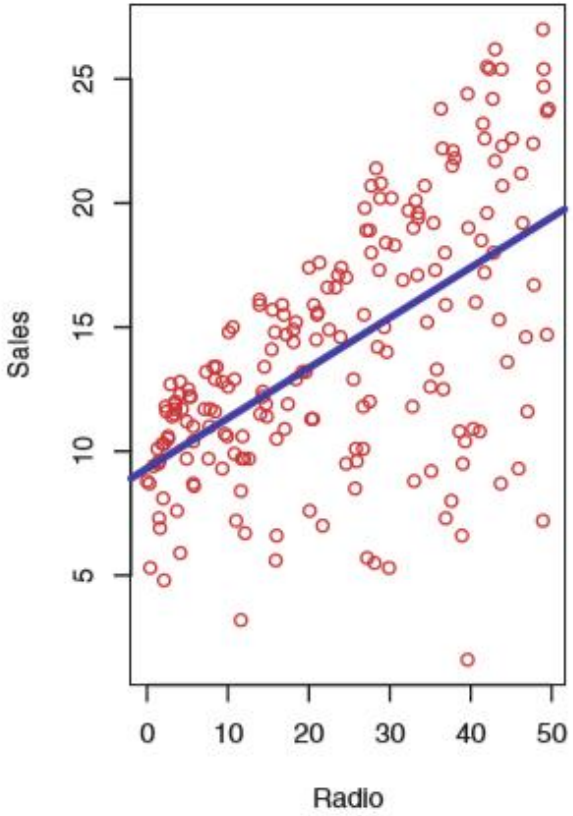


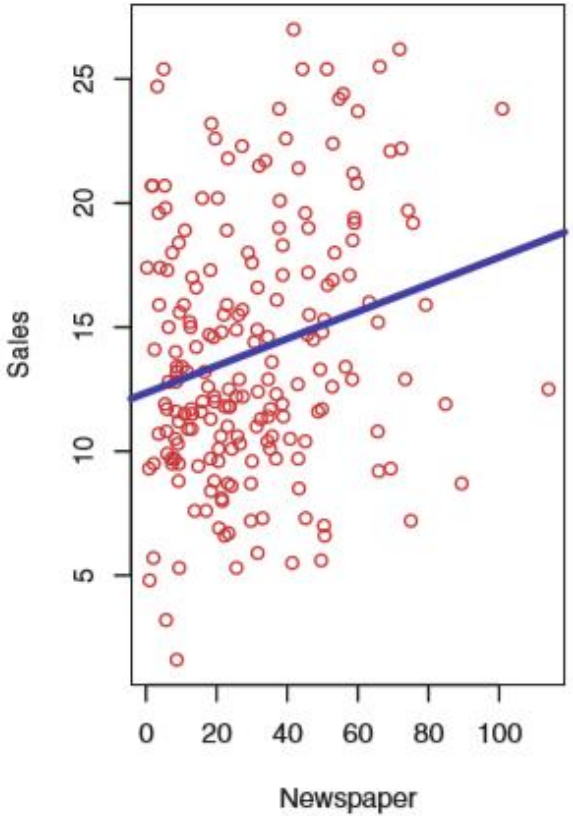
Example

- Let's look at another example.
  - Suppose, we are statistical consultants hired by a client to provide advice on how to improve sales of a particular product.
  - The **Advertising** data set consists of **sales** of the product in 200 different markets, along with advertising budgets in each of these markets for 3 different media: **TV**, **radio** and **newspaper**.
  - If we determine that there is an association between **advertising** and **sales**, then we can advice our client to adjust advertising budgets, thereby indirectly increasing **sales**.
  - In other words, our goal is to develop an accurate model that can be used to predict **sales** on the basis of the 3 media budgets.
- The *input variables,* typically denoted by *X,* are also referred to as *predictors* or *independent variables*. In our case, X_1 might be **TV** budget, X_2 the **radio** budget and X_3 the **newspaper** budget.
- The *output variable* – in this case, **sales** – is often called the *response* or *dependent variable*, typically denoted by *Y*.

Example


Sales
5
10
15
20
25
0
50
100
200
300
TV


Sales
5
10
15
20
25
0
10
20
30
40
50
Radio


Sales
5
10
15
20
25
0
20
40
60
80
100
Newspaper

Why Estimate f ?

- Our problem now boils down to estimating f, which is some unknown function representing the systematic information that $X = (X_1, X_2, X_3)$ provides about Y, i.e.
  - $Y = f(X) + e$ where e is a *random error* term.
- **Prediction**
  - f is treated as a black box, in the sense that one is not typically concerned with the exact form of f, provided that yields accurate predictions for Y.
- **Inference**
  - We are often interested in understanding the way that Y is affected as $X = (X_1, X_2, X_3)$ change. Now, f can't be treated as a black box, because we need to know its exact form.
  - Which predictors are associated with the response?
  - What is the relationship between the response and each predictor?
  - Can the relationship between Y and each predictor be adequately explained using a *linear equation*, or is the relationship more complex?

Prediction Example

- Consider a company that is interested in conducting a direct-marketing campaign.
- The goal is to identify individuals who will respond positively to a mailing, based on observations of demographic variables measured on each individual.
- In this case, the demographic variables serve as predictors, and response to the marketing campaign (either positive or negative) serves as the outcome.
- The company is not interested in obtaining a deep understanding of the relationships between each individual predictor and the response; instead, the company simply wants an accurate model to predict the response using the predictors. This is an example of modeling for *prediction*.

Inference Example

- In contrast, consider the **Advertising** data discussed in previous slides. One may be interested in answering questions such as:
  - Which **media** contribute to **sales**?
  - Which **media** generate the biggest boost in **sales**?
  - How much increase in **sales** is associated with a given increase in **TV** advertising?
- This situation falls into the *inference* paradigm.

How to Estimate f ?

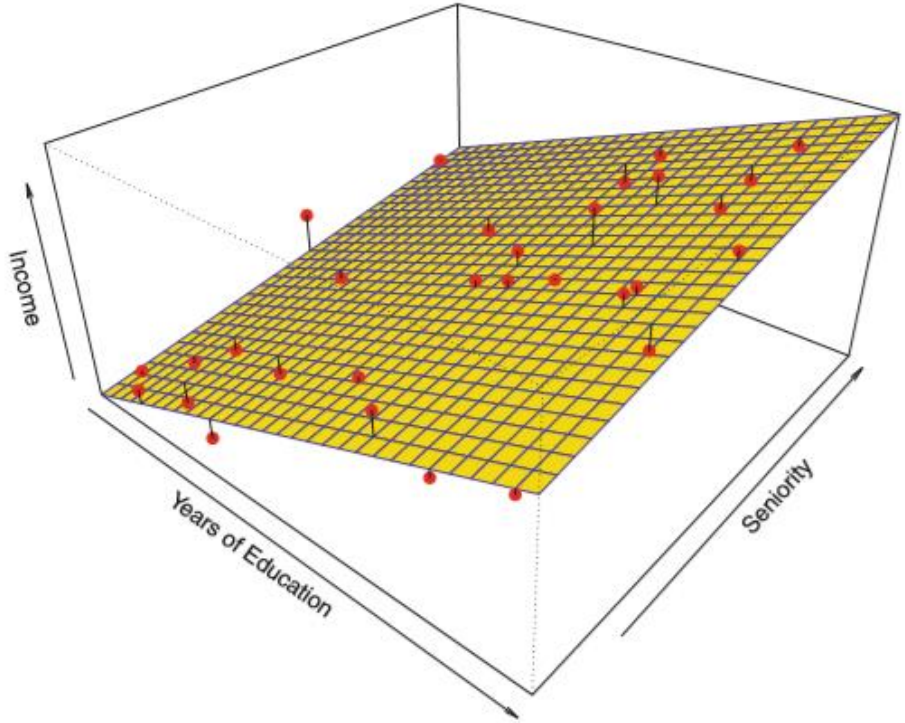
- **Parametric methods** involve a 2-step model-based approach**:**
  - First, we make an assumption about the functional form, or shape, of f. For example, one very simple assumption is that f is linear in X:
  - $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$
  - This is a linear model, which will be discussed in later slides. Once we have assumed that f is linear, the problem of estimating f is greatly simplified. One only needs to estimate the p + 1 coefficients $\beta_0, \beta_1, ..., \beta_p$.

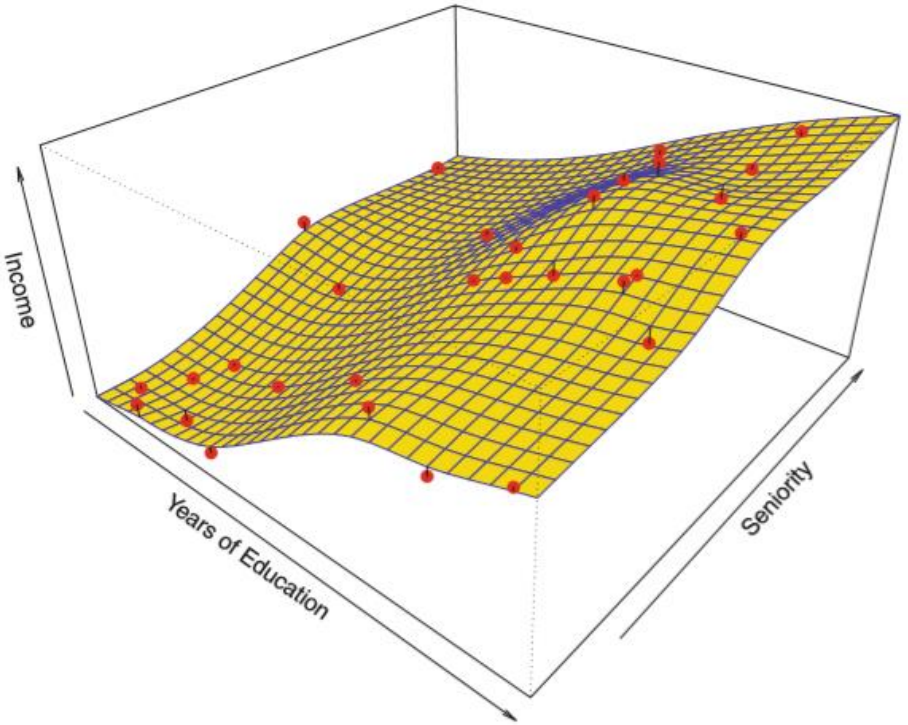
  - After a model has been selected, we need a procedure that uses the ***training data*** to fit or train the model. In the case of the linear model fit, we need to estimate the parameters $\beta_0, \beta_1, ..., \beta_p$ such that $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$.
  - The most common approach to fitting the linear model is referred to as ***least squares***, which we discuss later.

How to Estimate f ?

- **Non-Parametric methods** do not make explicit assumptions about the functional form of f. Instead they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly.
- Such approaches can have a major advantage over parametric approaches:
  - they have the potential to accurately fit a wider range of possible shapes for f.
- But non-parametric approaches do suffer from a major disadvantage:
  - since they do not reduce the problem of estimating f to a small number of parameters, a very large number of observations (far more than is typically needed for a parametric approach) is required in order to obtain an accurate estimate for f.

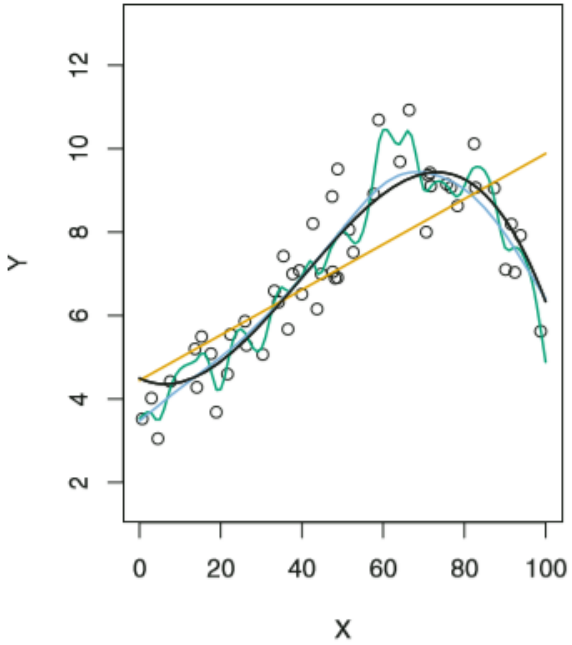
How to Estimate f ?

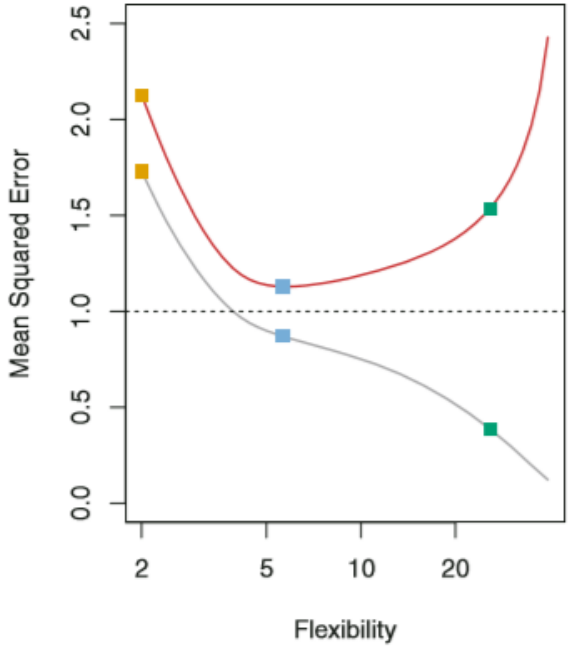




Model Accuracy

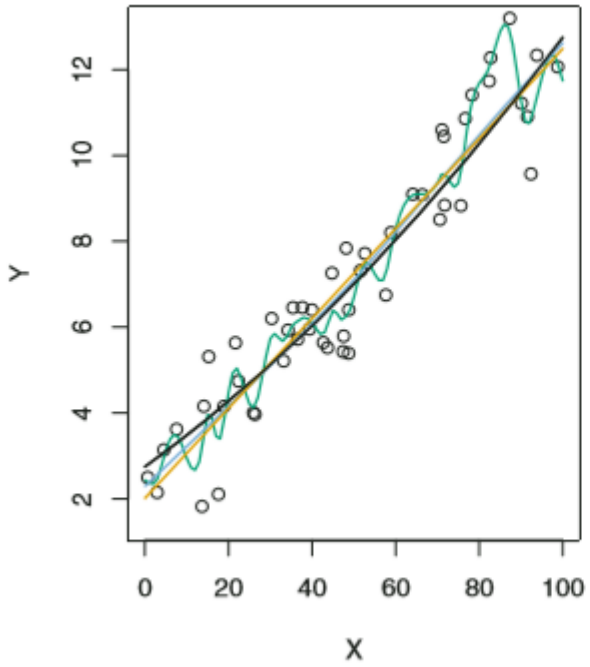
- *There is no free lunch in statistics*: no one method dominates all others over all possible data sets. Selecting the best approach can be one of the most challenging parts of performing statistical learning in practice.
  - **Measuring the Quality of Fit**: Training and test MSE (mean square of errors)
  - No guarantee that the method with the lowest training MSE will also have the lowest test MSE.
  - Increasing flexibility example wrt training data.

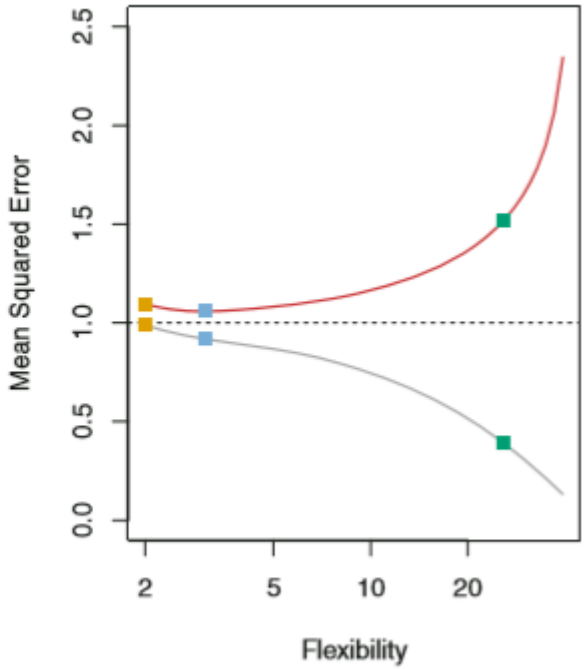




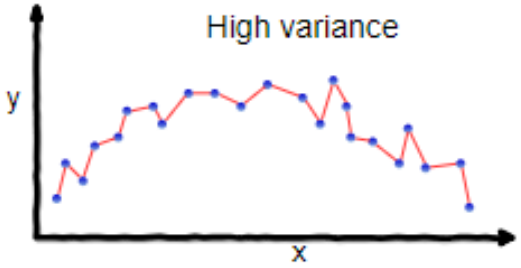
Model Accuracy

- Let's look at another example where f is much closer to being linear. In this setting, linear regression provides a very good fit to the data.
  - **Fundamental property of statistical learning**:
  - Monotone decrease in the training MSE
  - A U-shape in the test MSE
  - *Overfitting*: Small training MSE and large test MSE

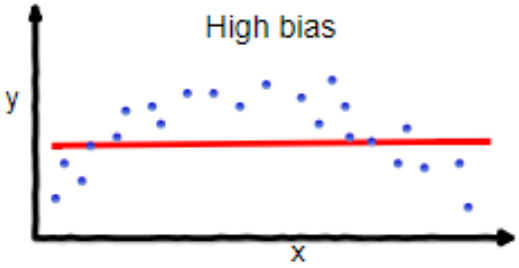




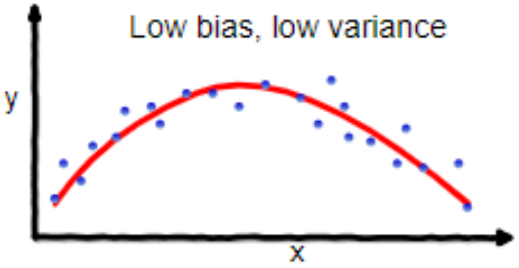
Model Accuracy



overfitting



underfitting

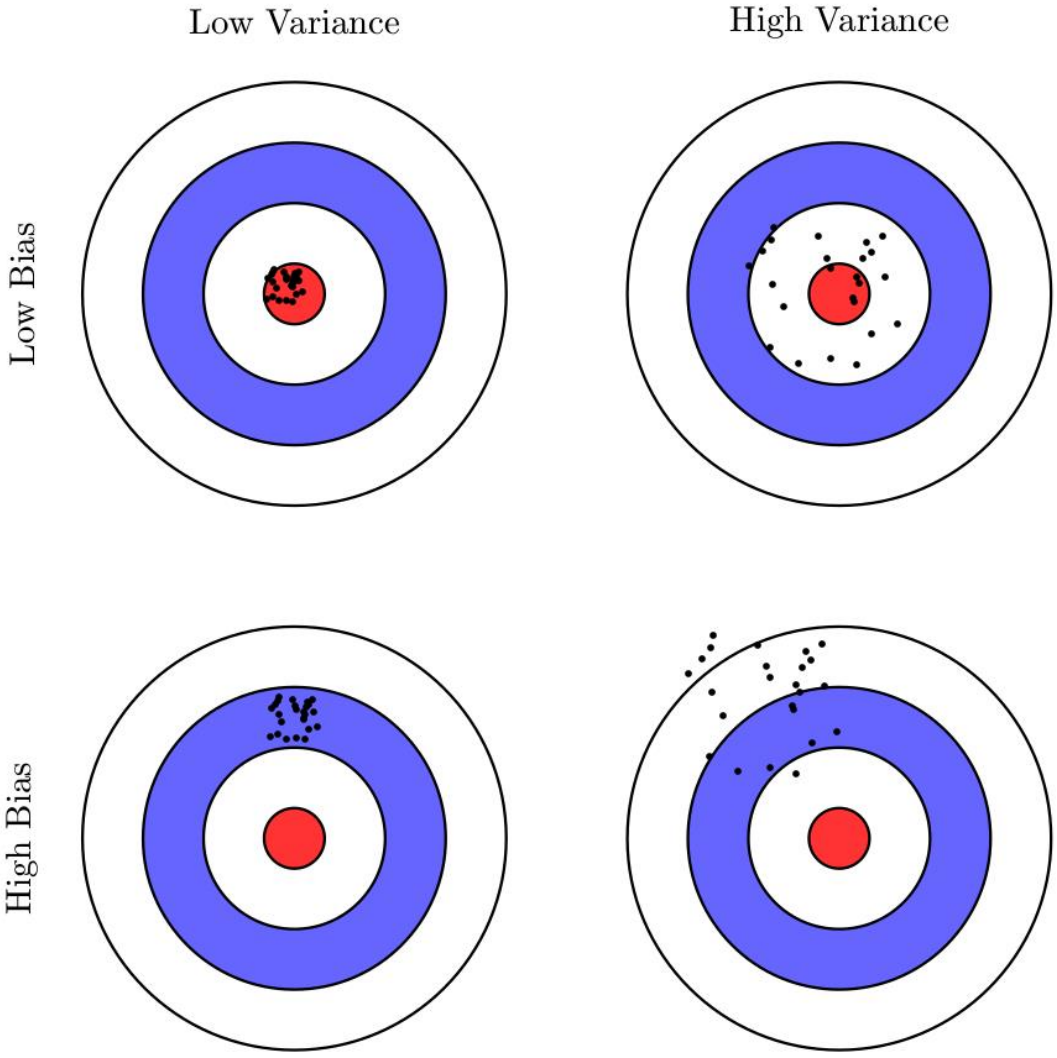


Good balance

Bias-Variance Trade-Off

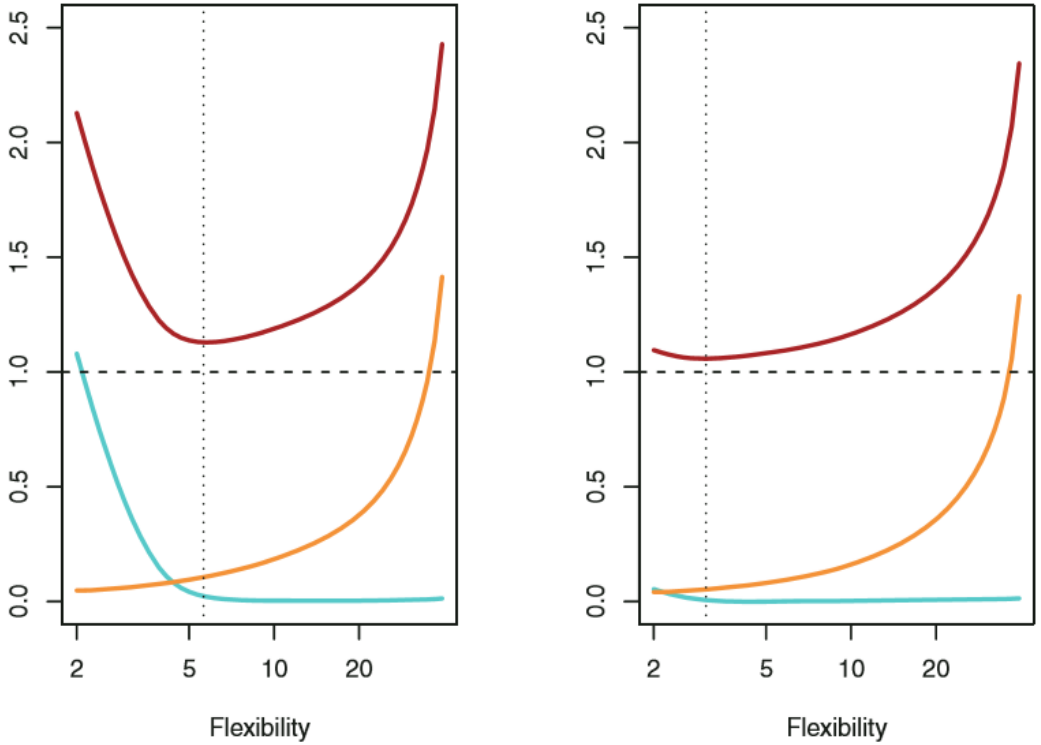
- Two competing properties: We need to select a statistical learning method that simultaneously achieves
  - **Low variance** (*variance* refers to the amount by which f would change if we estimated it using a different training data set. In general, more flexible methods have higher variance.)
  - **Low bias** (*bias* refers to the error that is introduced by approximating a real-life problem, which may be extremely complex, by a much simpler model.)

Bull's Eye Diagram



Bias-Variance Trade-Off

- Relationship between bias, variance and the test MSE is shown.
  - *Squared bias* (blue curve)
  - *Variance* (orange curve)
  - The dashed horizontal line represents the *random* or *irreducible error* Var(e).
  - *Test MSE* (red curve) is the sum of these 3 quantities.
  - In both the cases, the variance increases and the bias reduces as the method's flexibility increases.
  - Vertical dotted line: Flexibility corresponding to smallest test MSE.



Statistical Learning Exercise

Linear Regression

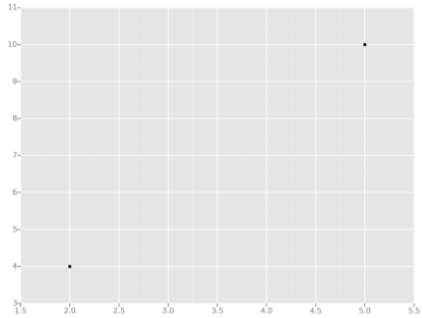
Summary

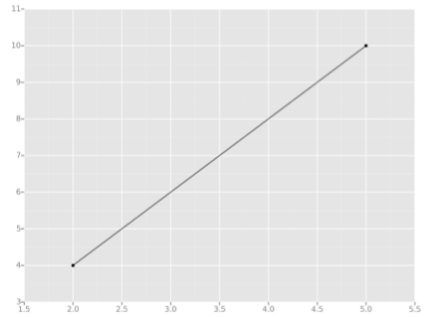
- **Regression problem**
  - Involves predicting a *continuous* or *quantitative* output value.

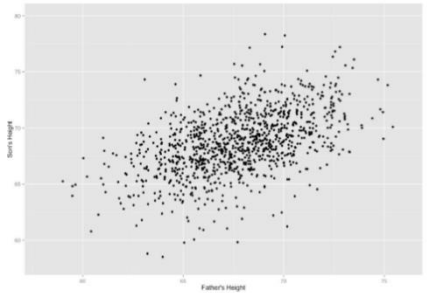
- **Classification problem**
  - Involves predicting a non-numerical value – that is, a *categorical* or *qualitative* output.

Linear Regression

- Let's take the simplest possible example: calculating a regression with only 2 data points.
- All we're trying to do when we calculate our regression line is draw a line that's as close to every dot as possible.
- Now wouldn't it be great if we could apply this same concept to a graph with more than just two data points?
  - By doing this, we could take multiple men and their son's heights and do things like tell a man how tall we expect his son to be...before he even has a son!

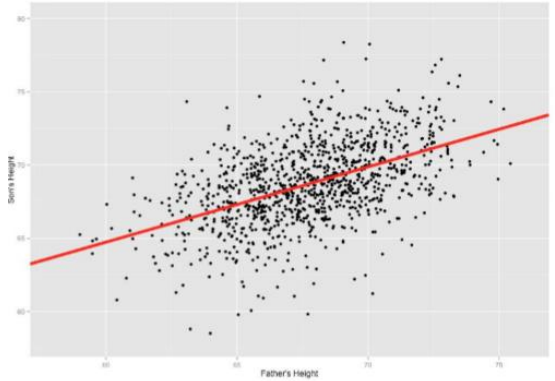






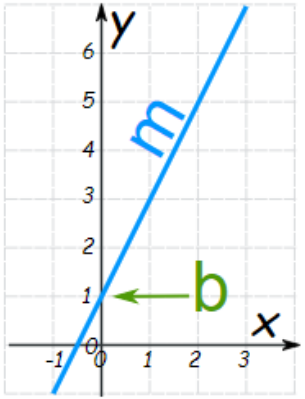
Linear Regression

- We can place the line "by eye": try to have the line as close as possible to all points, and a similar number of points above and below the line.
- But for better accuracy let's see how to calculate the line using **Least Squares** method.
- Our goal with **classic linear regression** is to minimize the vertical distance between all the data points and our line. So in determining the best line, we are attempting to minimize the distance between all the points and their distance to our line.



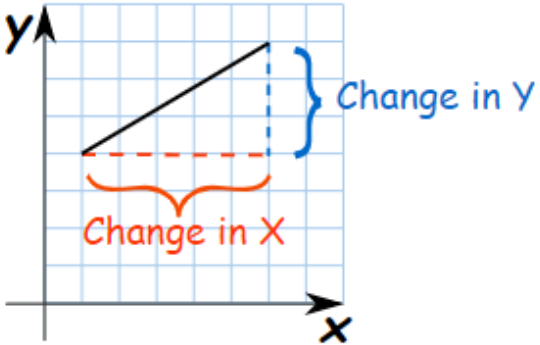
Equation of Line

$$y = mx + b$$



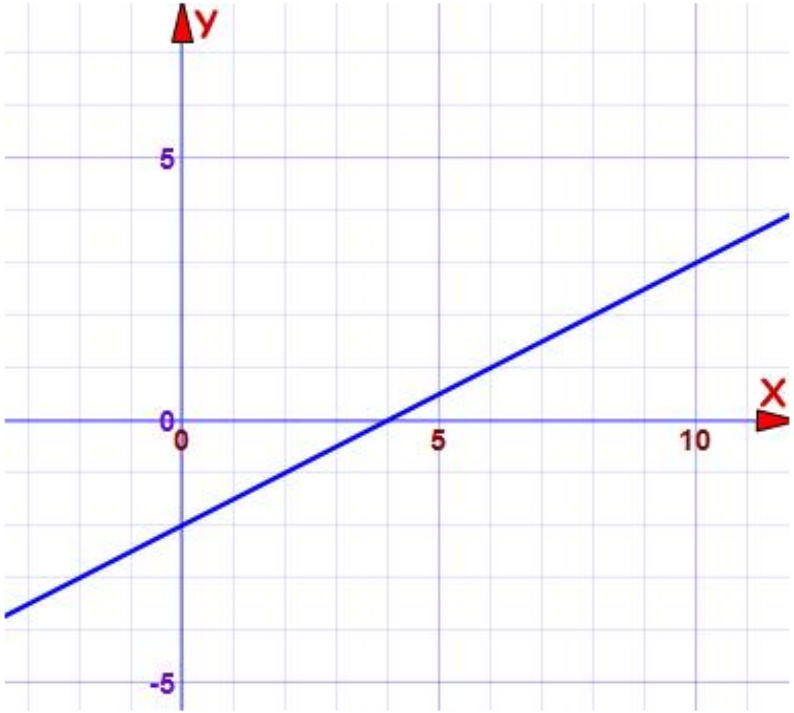
- where
  - m = Slope or Gradient (how steep the line is)
  - b = the Y Intercept (where the line crosses the Y axis)
- How do you find **m?**

$$m = \frac{\text{Change in Y}}{\text{Change in X}}$$



Q1.

What is the equation of the straight line shown in the diagram?



A $y = 2x - 2$

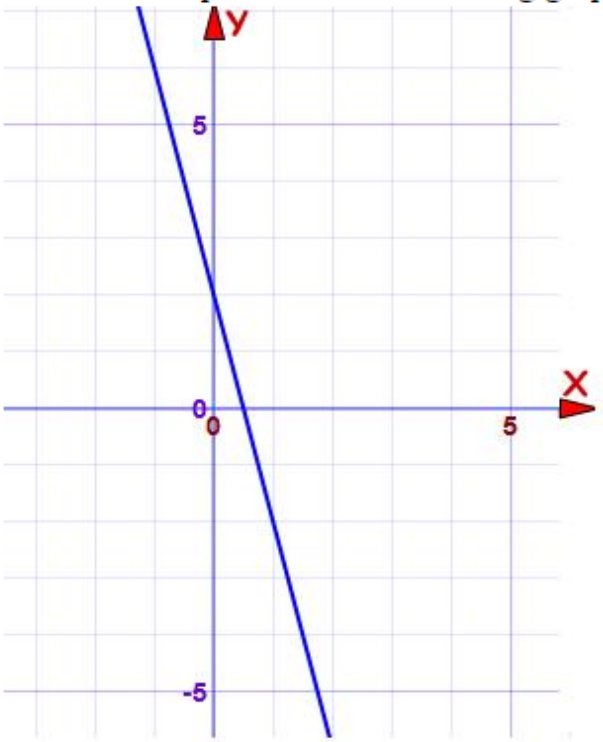
B $y = -2x + \frac{1}{2}$

C $y = -(\frac{1}{2})x + 2$

D $y = (\frac{1}{2})x - 2$

Q2.

What is the equation of the following graph?



A $y = -4x + 2$

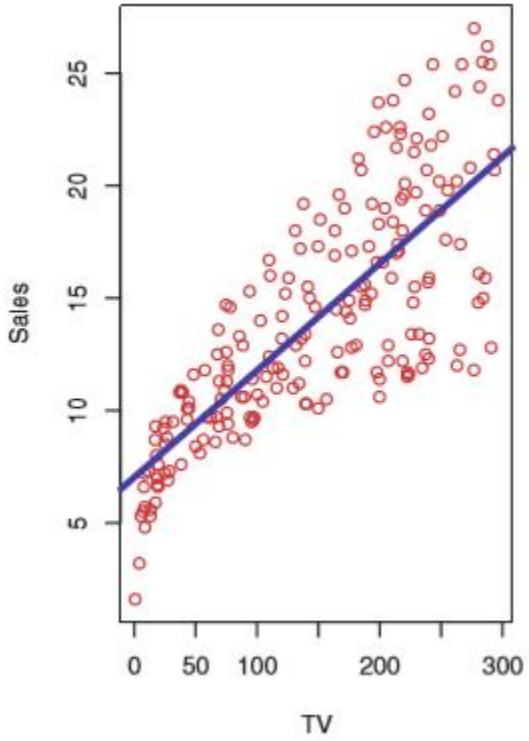
B $y = 4x + 2$

C $y = -0.25x + 2$

D $y = 0.25x + 2$

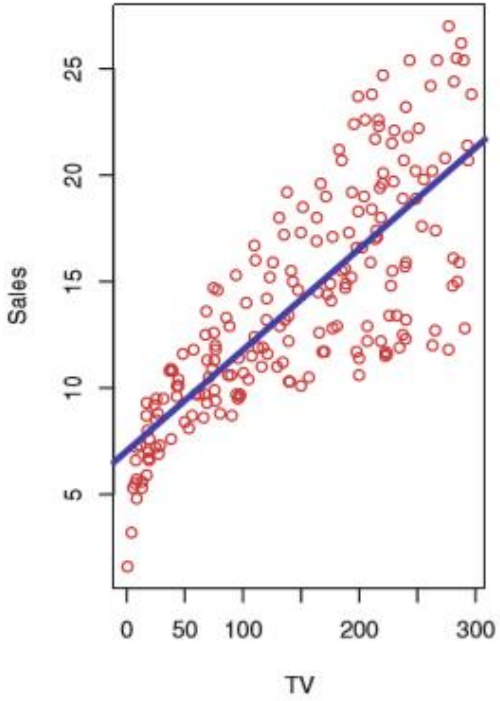
Simple Linear Regression

- It is a very straightforward approach for predicting a quantitative response Y on the basis of a single predictor X.
- It assumes a linear relationship between X and Y.
- Mathematically, we can write
  - $Y = \beta_0 + \beta_1 X_1$
  - where 2 unknown constants are intercept and slope terms.
- We describe the above equation by saying that we are *regressing Y onto X*.
  - For example, X may represent **TV advertising** and Y may represent **sales**. Then we can regress sales onto TV by fitting the model given by
  - **sales** = $\beta_0 + \beta_1$(***TV***)



Estimating the Coefficients

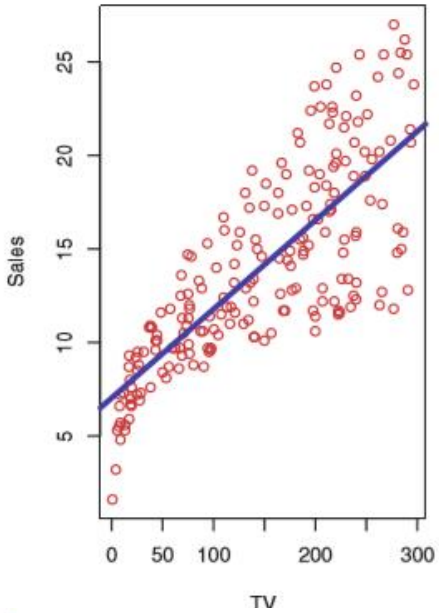
- How do we estimate the coefficients?
- We might have *n* observation pairs $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$
- We may choose the intercept and the slope values such that the resulting line is as close as possible to the *n* data points.
- There are a number of ways of measuring *closeness*.
  - Most common approach involves minimizing the *least squares criterion*.

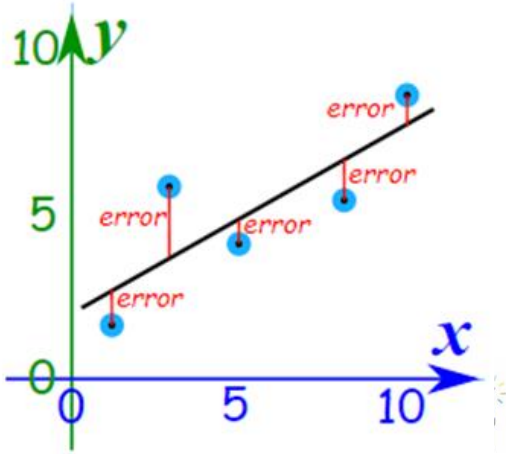


Least Square Fit

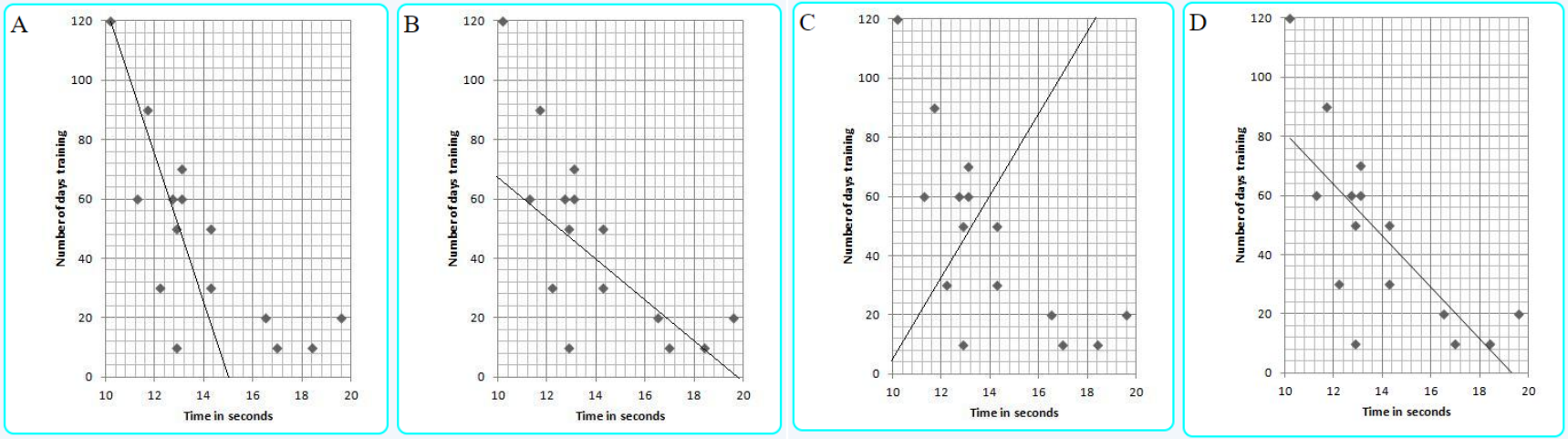
- How does it work?
  - It works by making the total of **square of errors** as small as possible. (which is why it is called *least square fit*)

- The straight line we choose minimizes the sum of squared errors. That is, when the square each of the errors and add them all up, the total is as small as possible.



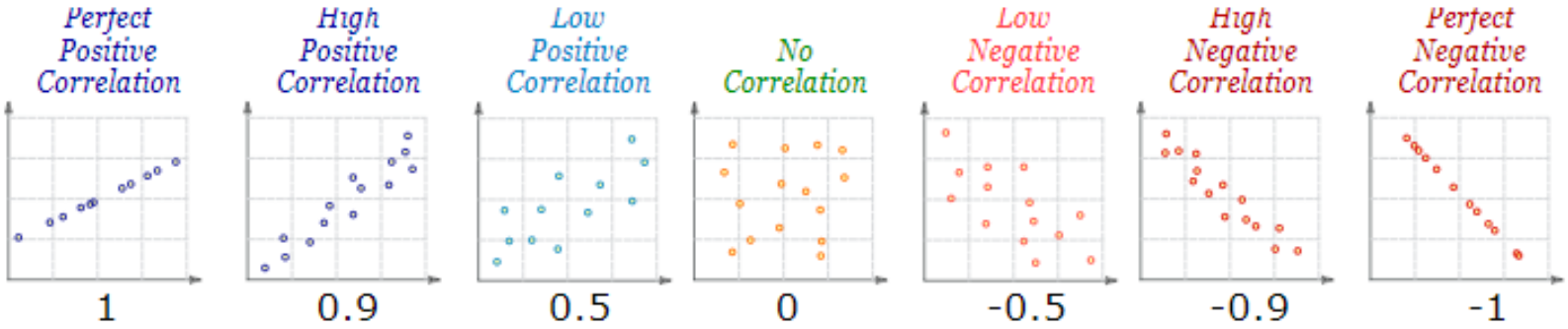


Which graph shows the most accurate "line of best fit" for the scatter plot?



- **Correlation**
  - When the two sets of data are strongly linked together we say they have a **High Correlation**.
  - Correlation is Positive when the values increase together, and it is Negative when one value decreases as the other increases.

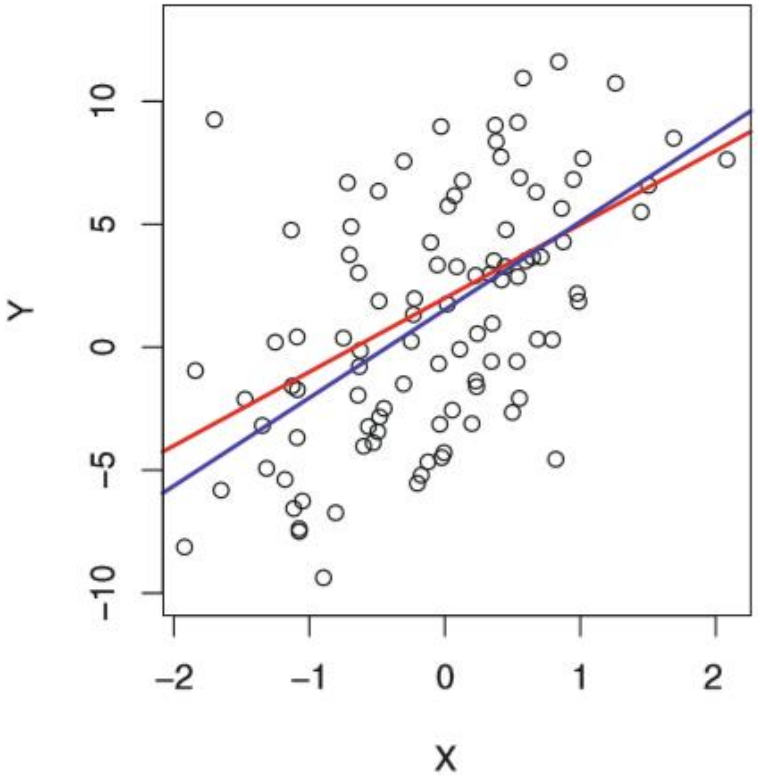
Correlation



- Correlation can have a value:
  - 1 is a perfect positive correlation
  - 0 is no correlation (the values don't seem linked at all)
  - -1 is a perfect negative correlation
- The value shows **how good the correlation is** (not how steep the line is), and if it is positive or negative.

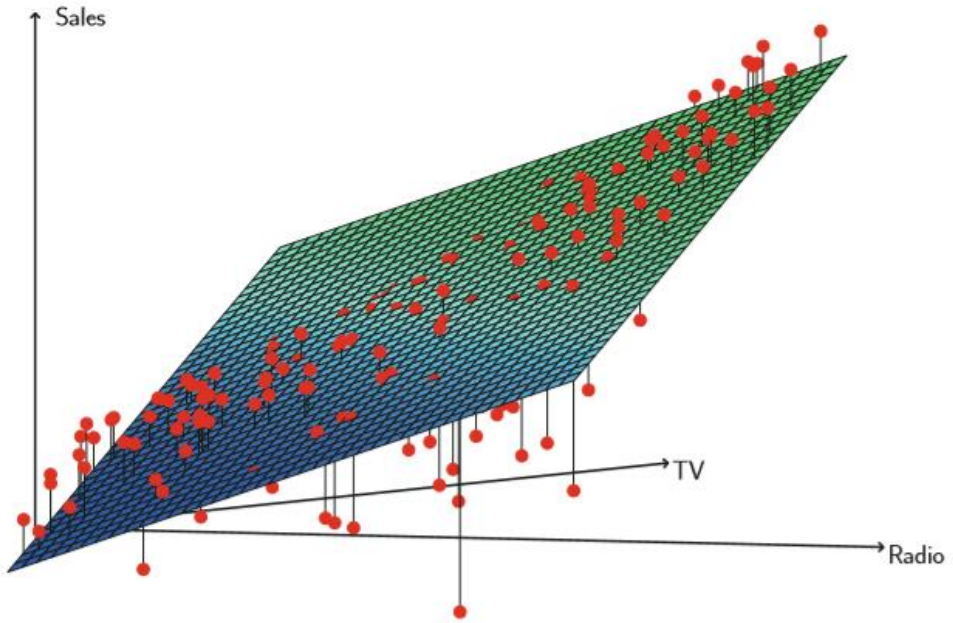
Accuracy of Coefficient Estimates

- In real applications, we have access to a limited set of observations from which we compute the **least squares line**.
- However, the true relationship is generally not known for the real data, and the **population regression line** is unobserved.
- Fundamentally, the concept of these two lines is a natural extension of the standard statistical approach of using information from a sample to estimate characteristics of a large population.



Multiple Linear Regression

- Instead of fitting a separate simple linear regression model for each predictor, a better approach is to extend the simple linear regression model so that it can directly accommodate multiple predictors.
- We can do this by giving each predictor a separate slope coefficient in a single model.
- If we have *p* predictors, the multiple regression model takes the form
  - $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$



Some Important Questions

- Is at least one of the predictors useful in predicting the response?
- Do all the predictors help to explain Y, or is only a subset of the predictors useful?
  - *Forward selection* is a greedy approach.
    - We begin with a null model – a model that contains an intercept but no predictors.
    - We then fit p simple linear regressions and add to the null model the one resulting in lowest square of errors.
    - We continue like this until some stopping rule is satisfied.
  - *Backward selection*
  - *Mixed selection*
- How well does the model fit the data?
- Given a set of predictor values, how accurate is our prediction?

Other Considerations

- Qualitative Predictors
  - So far, we assumed that all variables are quantitative. But in practice, often some predictors are qualitative.
  - For example, suppose we wish to investigate the differences in credit card balance between males and females. We may create a *dummy variable* based on the gender variable of the form

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male,} \end{cases}$$

  - and use this variable as a predictor in the regression equation. This results in the model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

  - Qualitative predictors with more than 2 levels?

Other Considerations

- Additive Assumption
  - For example, in our previous analysis of **Advertising** data, the linear models assumed that effect on **sales** of increasing one advertising medium is independent of the amount spent on the other media.
  - However, there may be some ***synergy effect*** or an ***interaction effect***.
  - One way of introducing an interaction term to allow for interaction effects between X_1 and X_2 is by computing the product of $X_1.X_2$

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

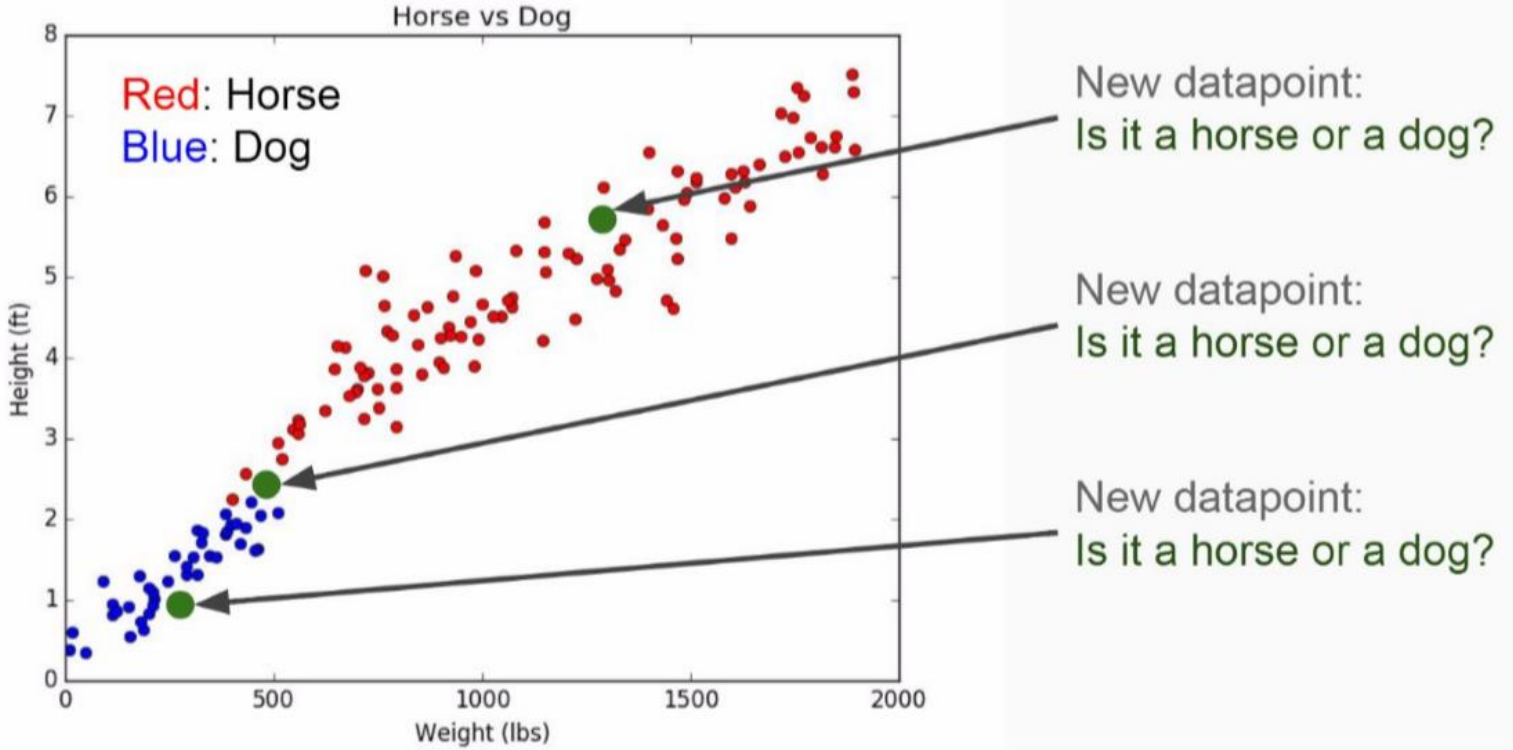
$$Y = \beta_0 + (\beta_1 + \beta_3 X_2) X_1 + \beta_2 X_2$$

Regression Exercise

Classification Problem

Example

- Imagine we had some imaginary data on Dogs and Horses, with heights and weights.

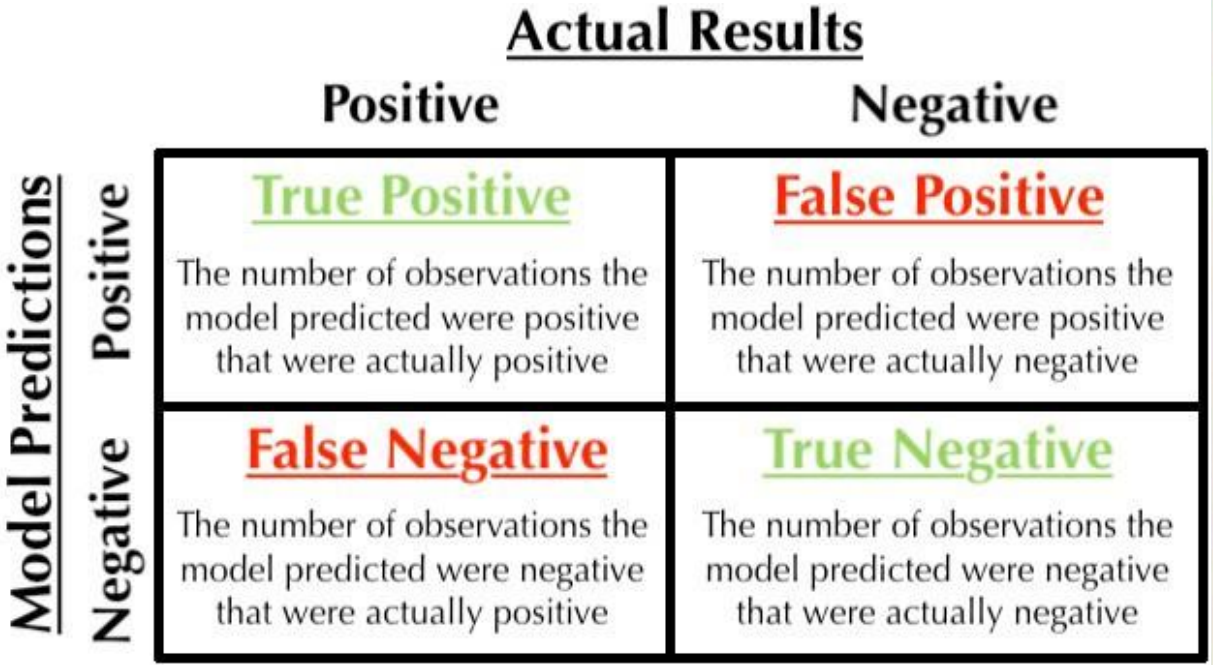


Good Classifier

- So far, the model accuracy was focused on the regression setting.
- Many of the concepts such as bias-variance trade-off also transfer to the classification setting with small modifications.
  - Due to the fact that Y is no longer numerical.
- The most common approach for quantifying the accuracy of our estimate *f* is the training or the test *error rate*, the proportion of mistakes that are made if we apply our estimate *f* to the training/test observations.
- Good classifier = small test error rate

Good Classifier – Test Error Rate

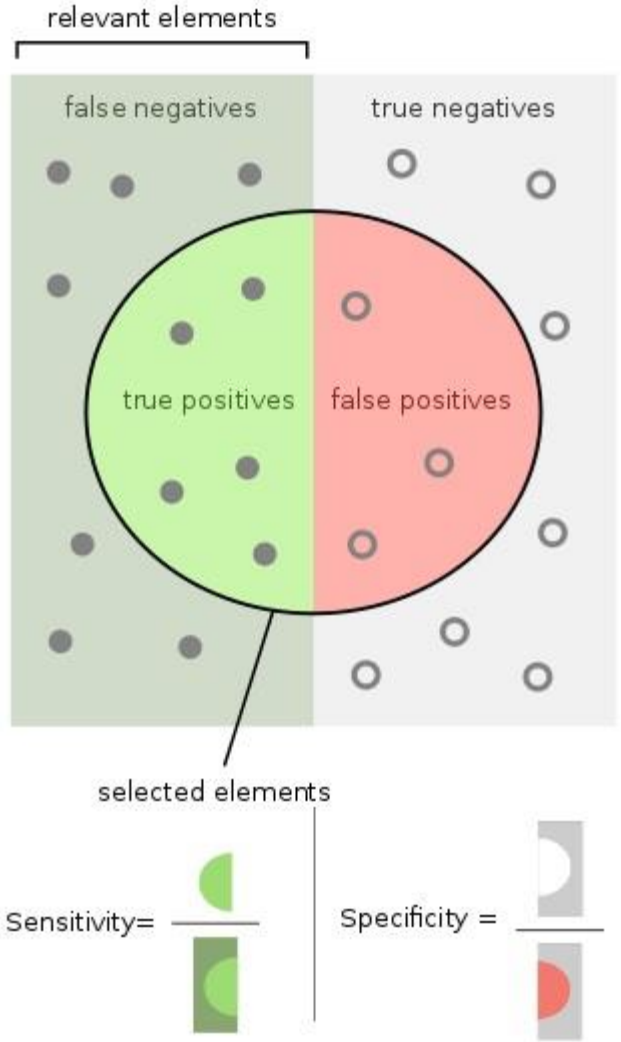
- **Confusion Matrix**
  - A convenient way to display the information
- Two kinds of error rate
  - Sensitivity
  - Specificity



Example

- COVID-19 Test Kit from China

		Actual Condition	
		Condition Positive	Condition Negative
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**
		Sensitivity = Σ True Positive / Σ Condition Positive	**Specificity** = Σ True Negative / Σ Condition Negative



- Common Cold – Medicine?

Bayes Classifier

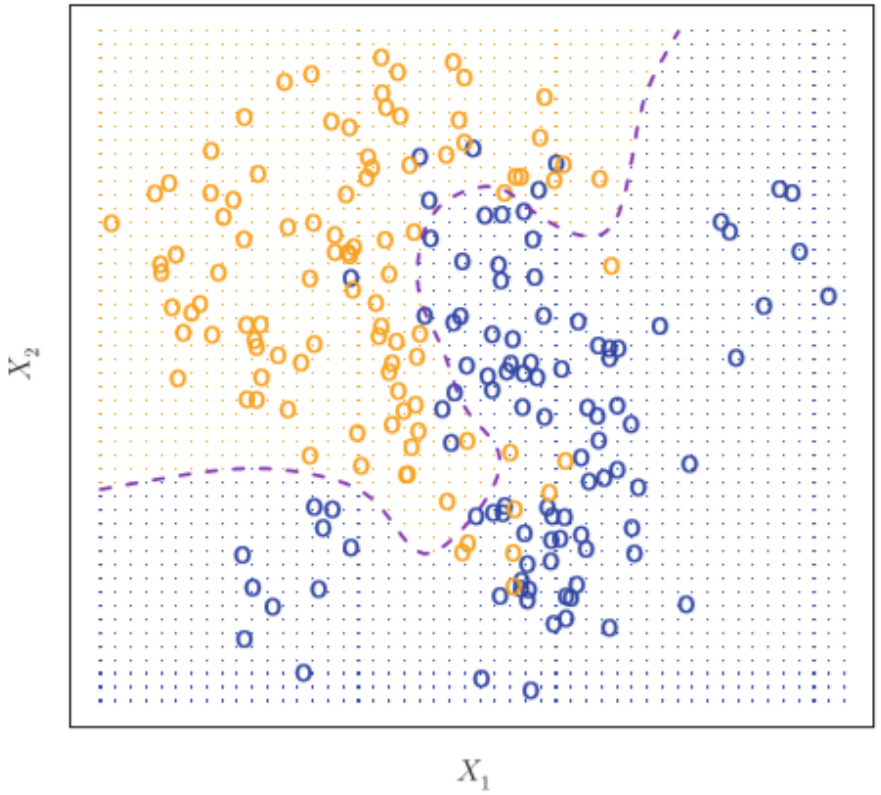
- It can be shown that the test error rate is minimized, on average, by a very simple classifier that assigns each observation to the most likely class, given its predictor values.

$$\Pr(Y = j|X = x_0)$$

- It is the conditional probability that *Y = j*, given the observed predictor vector *X = x0*.
- This simple classifier is called the *Bayes classifier*.

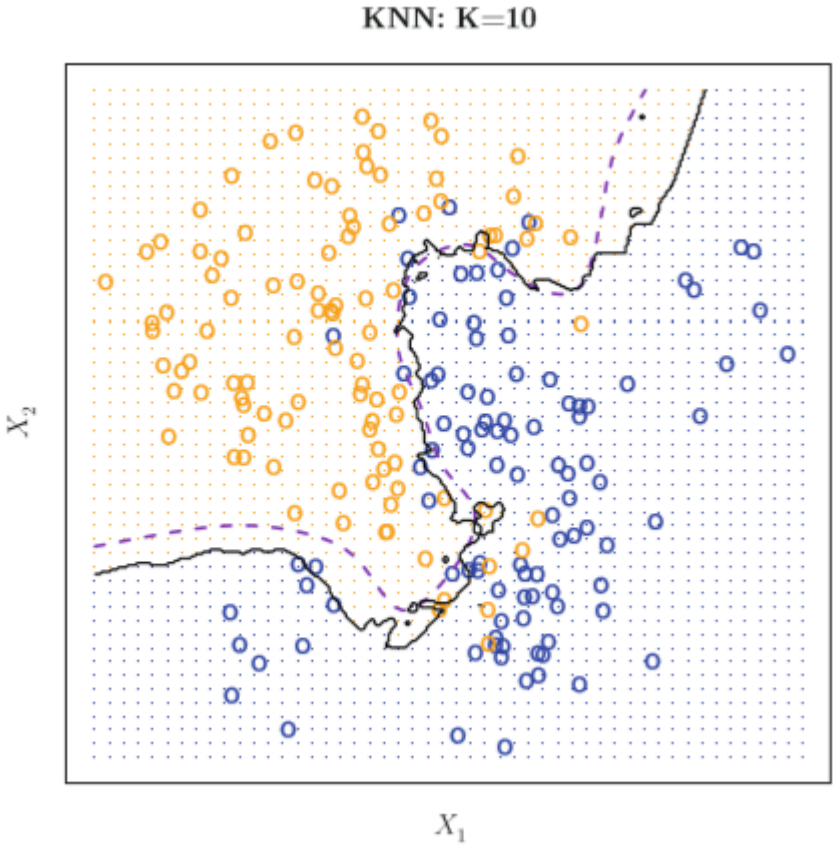
Bayes Classifier

- In a 2-class problem, there are two possible response values.
- The Bayes classifier corresponds to predicting class 1 if $Pr(Y = 1 | X = x_0) > 0.5$ and class 2 otherwise.
- For example,
  - 100 observations in each of the 2 groups
  - Dashed line represents the Bayes boundary
  - Finite Bayes error rate



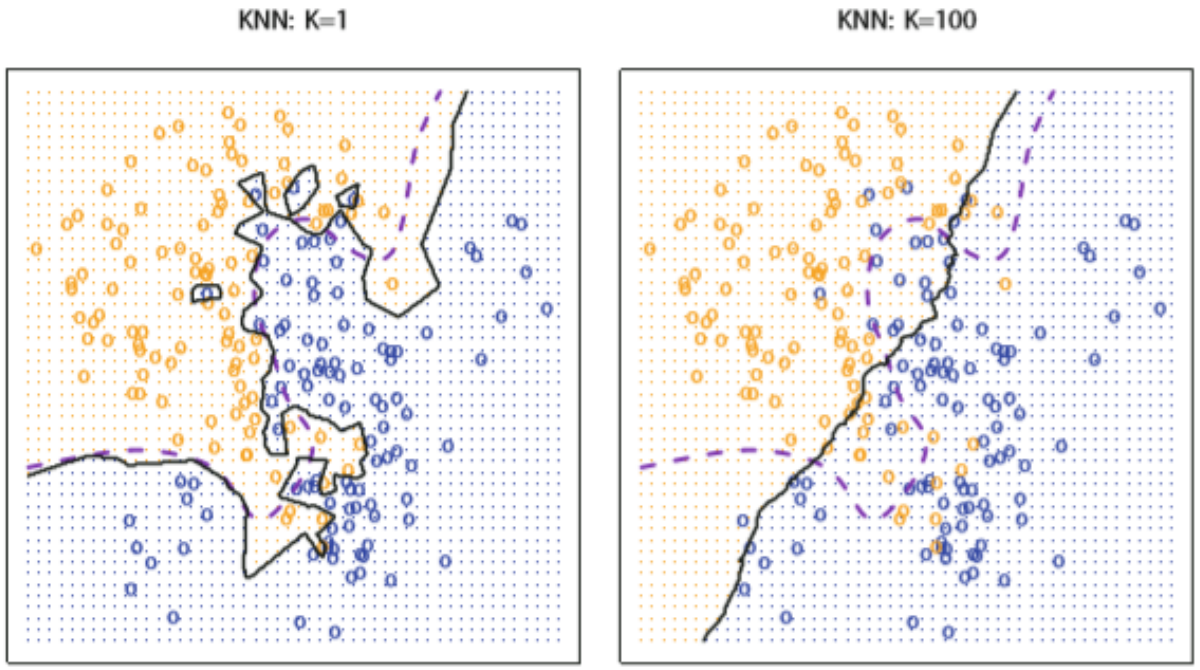
K-Nearest Neighbors

- Bayes classifier is the unattainable gold standard against which to compare other methods such as *K-nearest neighbors (KNN) classifier*.
  - Given a positive integer K and a test observation x_0, the KNN classifier first identifies K points in the training data that are closest to x_0.
  - It then estimates the conditional probability $\Pr(Y = j | X = x_0)$.



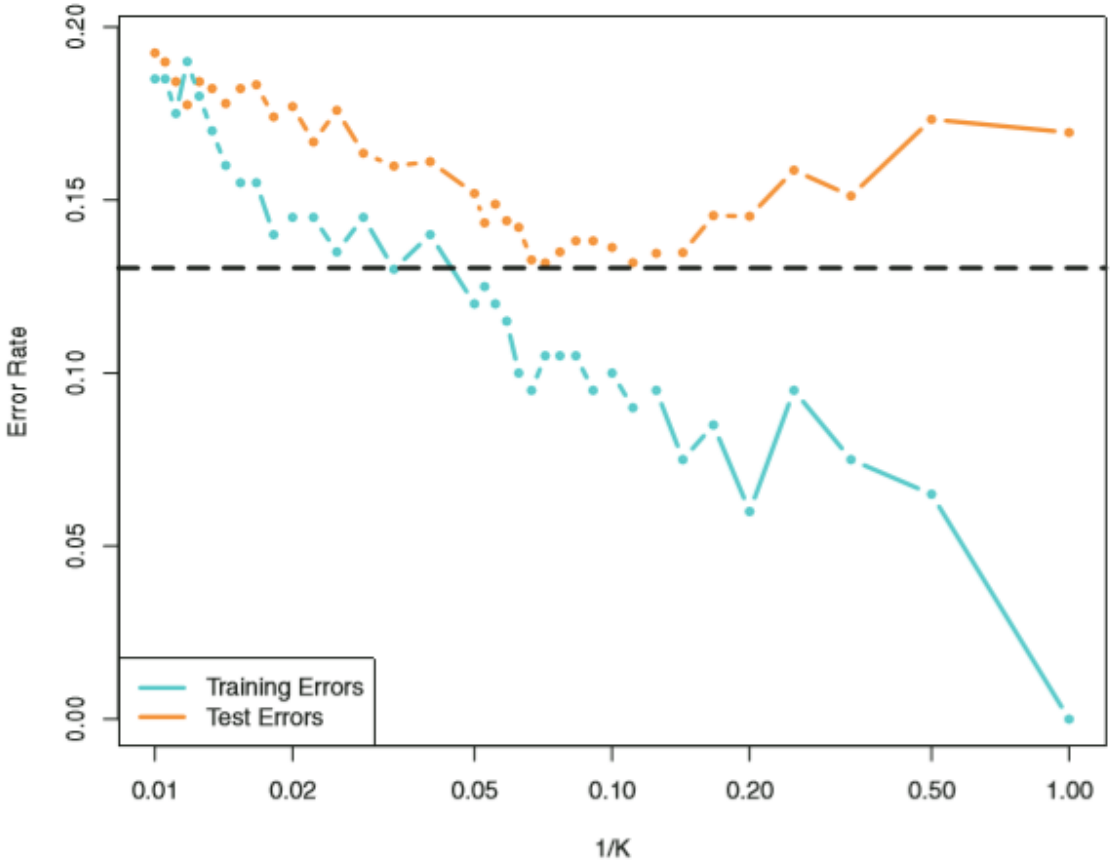
K-Nearest Neigbors

- The choice of K has a drastic effect on the KNN classifier obtained.
  - For example, for K = 1, the decision boundary is overly flexible, while with K = 100 it is not sufficiently flexible.



K-Nearest Neigbors

- As 1/K increases, the method becomes more flexible.
- As in the regression setting, the training error rate consistently declines as the flexibility increases. However, the test error exhibits a characteristic U-shape.



Classification Exercise

K-Nearest Neighbors Regression

- Linear regression is an example of a parametric approach because it assumes a linear functional form for $f(X)$.
- The simplest and best-known non-parametric method is the *KNN regression*, which is closely related to the KNN classifier.
  - For an observation x_0, it estimates $f(x_0)$ using the average of all the training responses that are closest to x_0.
- In what setting will a parametric approach such as least squares linear regression outperform a non-parametric approach such as KNN regression?
  - If the parametric form selected is close to the true form of f.
- In higher dimensions, KNN often performs worse than linear regression.
  - Curse of dimensionality.

Tree-Based Methods

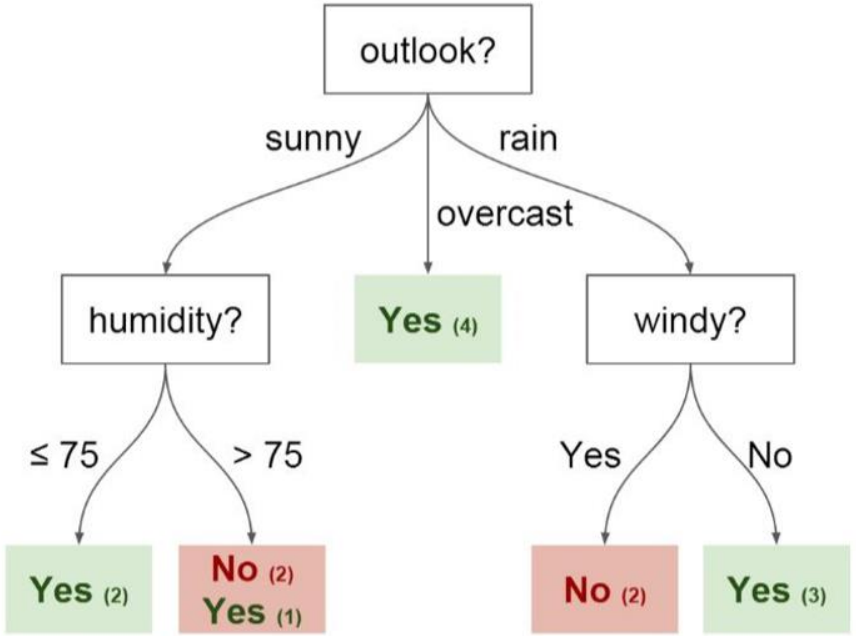
Example

- Let’s start off with a thought experiment to give some motivation behind using a decision tree.
- Imagine that you play Tennis every Saturday and you always invite a friend to come with you.
  - Sometimes your friend shows up, sometimes not.
  - For him it depends on a variety of factors, such as: weather, temperature, humidity, wind etc..
  - You start keeping track of these features and whether or not he showed up to play.

Temperature	Outlook	Humidity	Windy	Played?
Mild	Sunny	80	No	Yes
Hot	Sunny	75	Yes	**No**
Hot	Overcast	77	No	Yes
Cool	Rain	70	No	Yes
Cool	Overcast	72	Yes	Yes
Mild	Sunny	77	No	**No**
Cool	Sunny	70	No	Yes
Mild	Rain	69	No	Yes
Mild	Sunny	65	Yes	Yes
Mild	Overcast	77	Yes	Yes
Hot	Overcast	74	No	Yes
Mild	Rain	77	Yes	**No**
Cool	Rain	73	Yes	**No**
Mild	Rain	78	No	Yes

Example

- Let us use this data to predict whether or not he will show up to play.
- An intuitive way to do this is through a Decision Tree.
- In this tree we have:
  - Nodes, which are split for the value of a certain attribute
  - Edges
  - Root – the first node
  - Leaves, which are the terminal nodes that predict the outcome

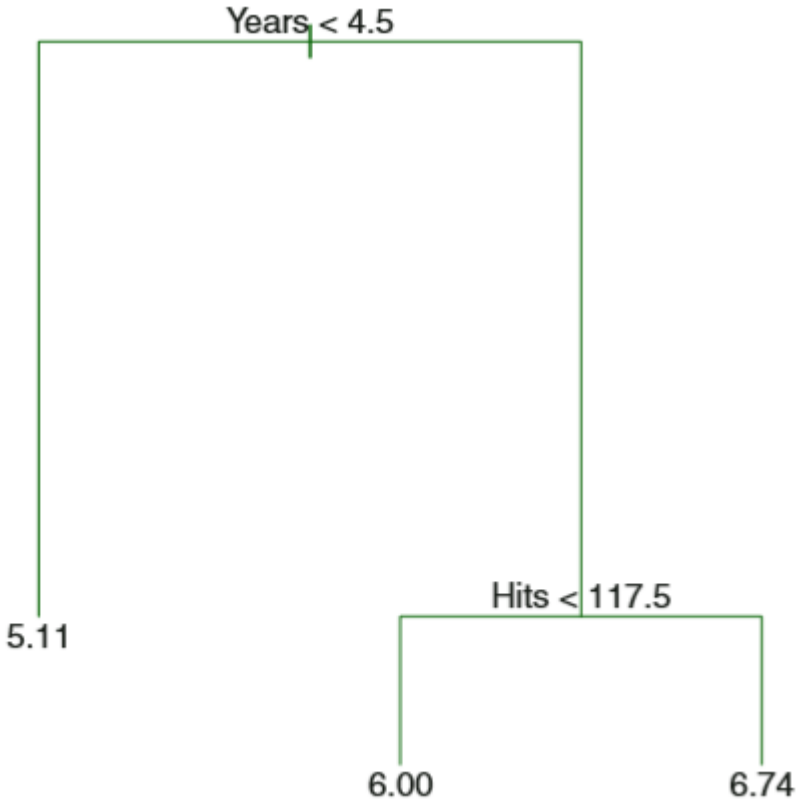


Decision Trees

- These involve segmenting the predictor space into a number of simple regions.
- Since, the set of splitting rules used to segment the predictor space can be summarized in a tree, these type of approaches are known as *decision tree methods*.
- Decision trees can be applied to both regression and classification problems.

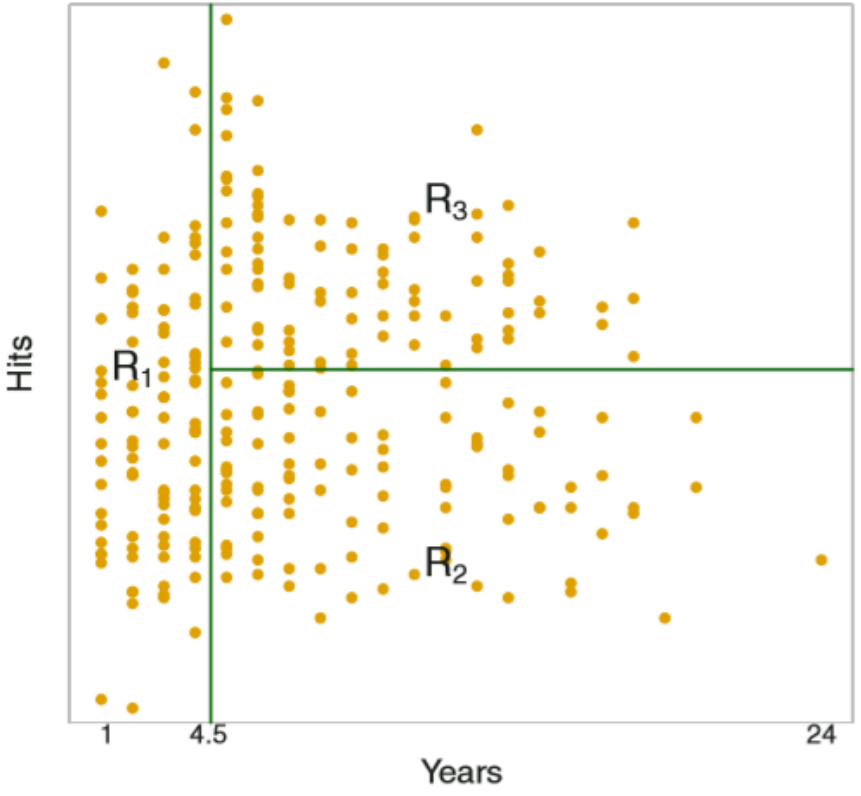
Regression Trees

- Let us begin with a simple example:
  - Predicting baseball players' salaries using regression trees
  - We use **Hitters** data set to predict a baseball player's **Salary** (in thousands of $) based on **Years** (the number of years that he has played in the leagues) and **Hits** (number of hits that he made in the previous year).
- Top split assigns observations having Years < 4.5 to the left branch.
- The predicted salary for these players is given by the mean response value for the players in the data set with Years < 4.5.



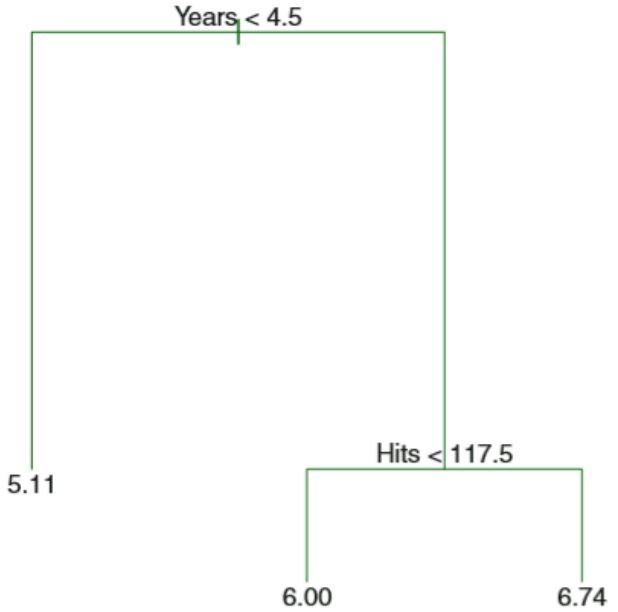
Regression Tree Example

- Overall, the tree stratifies or segments the players into 3 regions of predictor space:
  - Players who have played for 4 or fewer years
  - Players who have played for 5 or more years and make fewer than 118 hits last year
  - Players who have played for 5 or more years and make at least 118 hits last year
- Tree analogy – the 3 regions are known as *terminal nodes* or *leaves* of the tree.



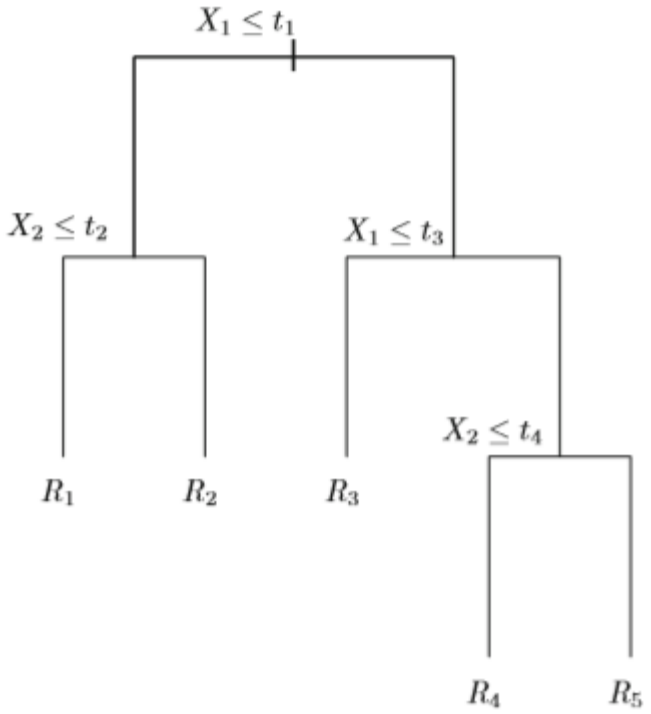
Regression Trees

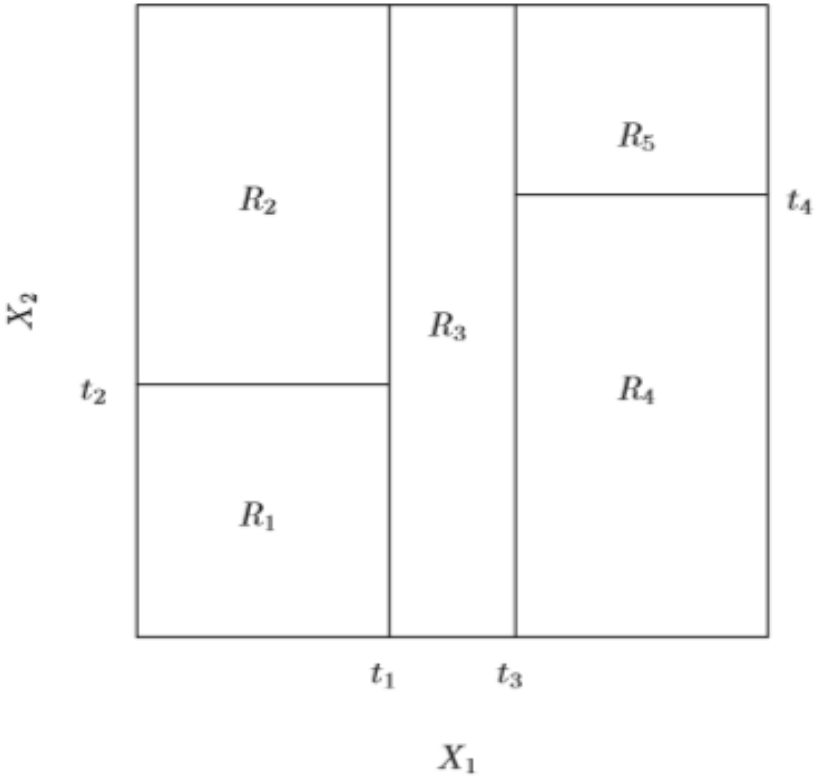
- Advantages and disadvantages:
  - Over-simplification of the true relationship between **Hits**, **Years** and **Salary**. Lack of predictive accuracy.
  - Easier to interpret
  - Nice graphical representation
- Computationally infeasible to consider every possible partition of the feature space.
- We take a top-down, greedy approach known as *recursive binary splitting*.
  - Begins at the top of the tree
  - Successively splits the predictor space
  - Each split into two branches further down the tree
  - At each step, the best split is made, rather than looking ahead



Recursive Binary Splitting

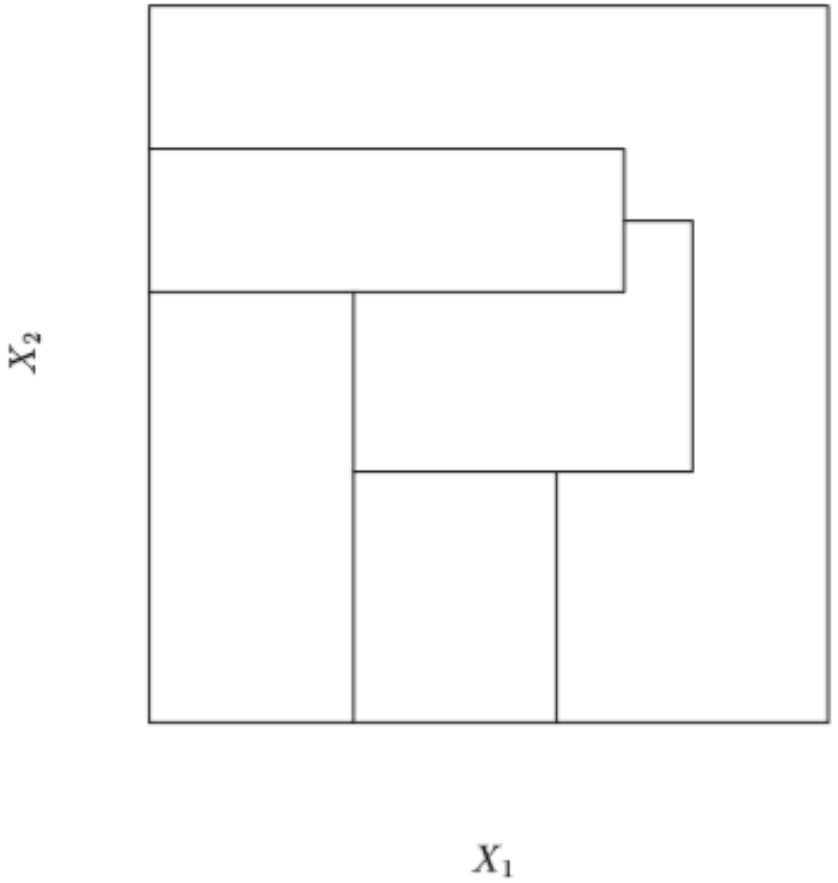
- Can the partition of two-dimensional feature space shown be a result from recursive binary splitting?

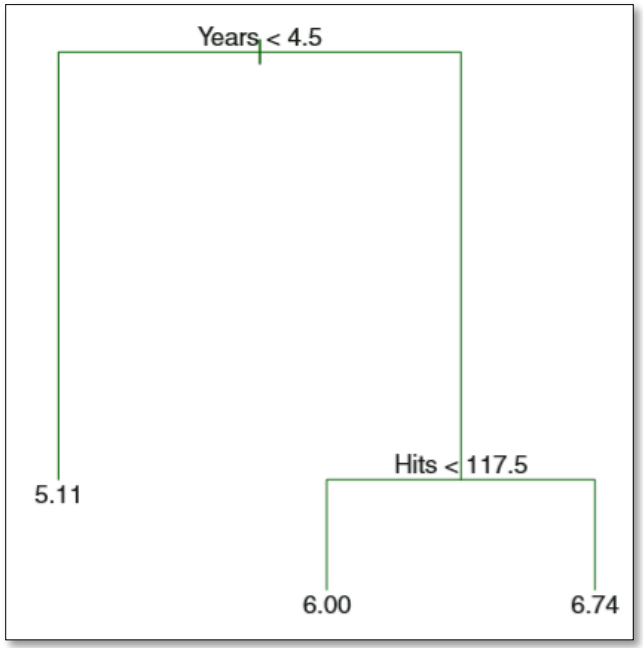


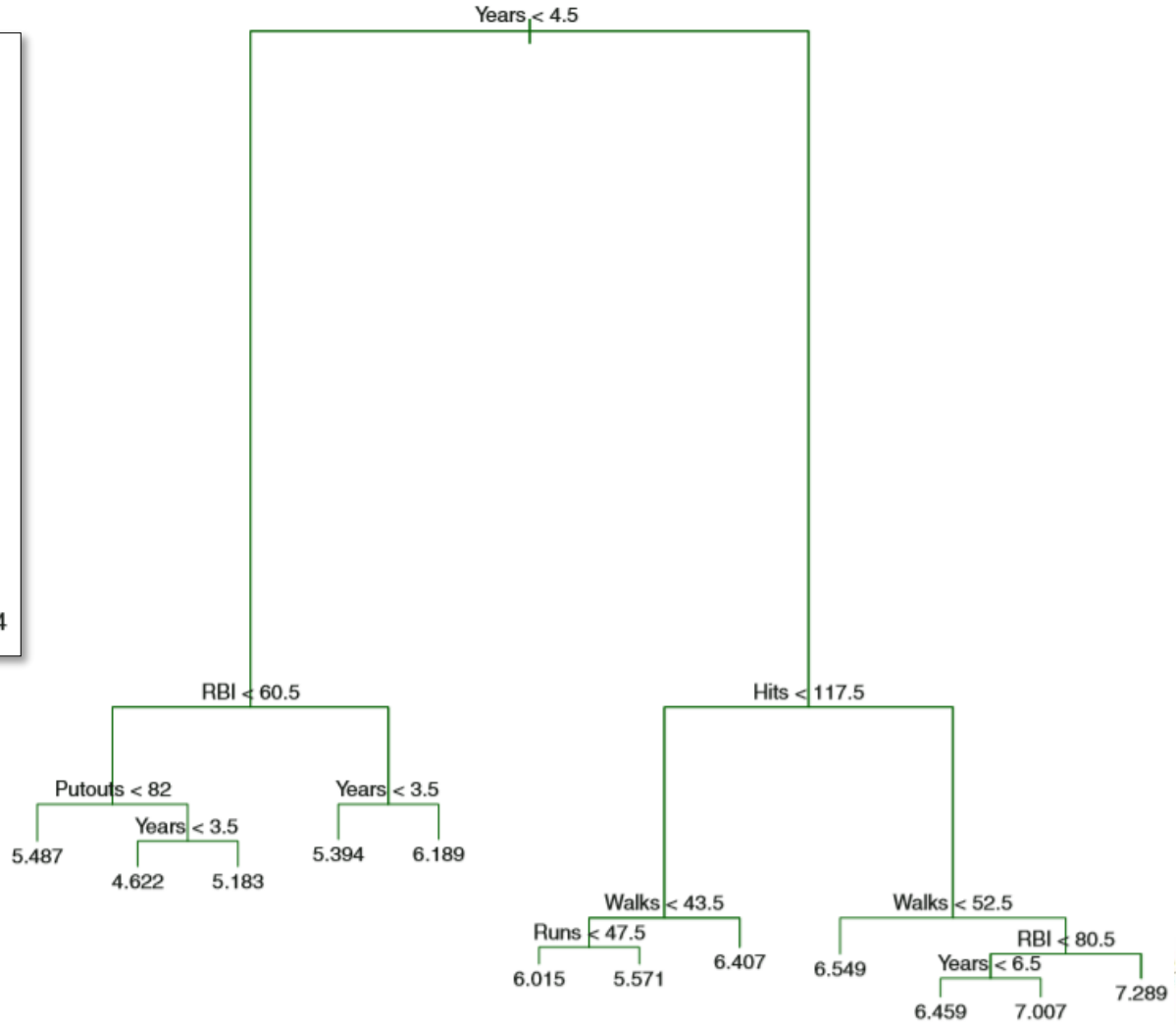


Recursive Binary Splitting

- Can the partition of two-dimensional feature space shown on the right be a result from recursive binary splitting?




Years < 4.5
5.11
Hits < 117.5
6.00
6.74

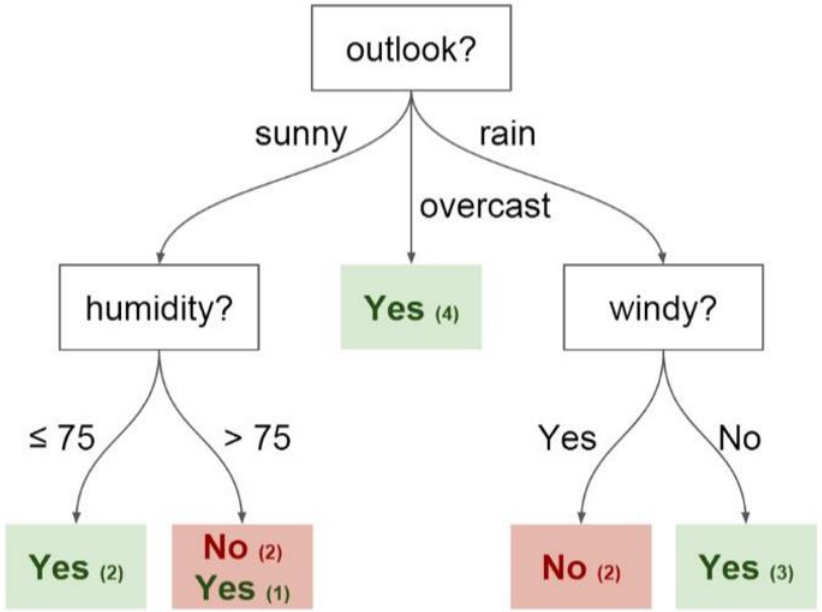

Years < 4.5
RBI < 60.5
Hits < 117.5
Putouts < 82
Years < 3.5
Years < 3.5
Walks < 43.5
Walks < 52.5
5.487
4.622
5.183
5.394
6.189
Runs < 47.5
6.407
RBI < 80.5
6.015
5.571
6.549
Years < 6.5
7.289
6.459
7.007

Tree Pruning

- One strategy is to build the tree only so long as the decrease in the RSS due to each split exceeds some threshold.
  - Smaller trees
  - Short-sighted – ? seemingly worthless split early on followed by a very good split
- Grow a large tree and then prune it, to obtain a *subtree*
  - Goal is to lower the test error rate
  - Many possible subtrees
  - *Cost complexity* or *weakest link pruning* where we consider only a sequence of subtrees

Classification Trees

- Classification tree is similar to a regression tree, except that it is used to predict a qualitative response rather than a quantitative one.
- Just as in the regression setting, we can use recursive binary splitting to grow a classification tree.
- RSS cannot be used as a criterion for making the binary splits.
  - Classification error rate
  - *Gini index* – a measure of node purity (a small value indicates that a node contains predominantly observations from a single class)
  - *Cross-entropy* – used to evaluate the quality of a particular split



Bagging

- Decision Trees Review:
  - Easier to explain to people…definitely easier than linear regression!
  - Graphical representation and easy to interpret even by a non-expert.
  - Can handle qualitative predictors without the need to create dummy variables.
  - Unfortunately, trees generally lack predictive accuracy as compared to other regression and classification approaches.
- By aggregating many decision trees, using methods like *bagging*, *random forests* and *boosting*, the predictive performance of trees can be substantially improved.
- *Bootstrap aggregation* or *boosting* is a general purpose procedure to reduce the variance of a statistical learning method.
  - And average all the predictions from B different trees.
  - We generally do not have access to multiple training sets. We generate B different bootstrapped training data sets.

Bagging

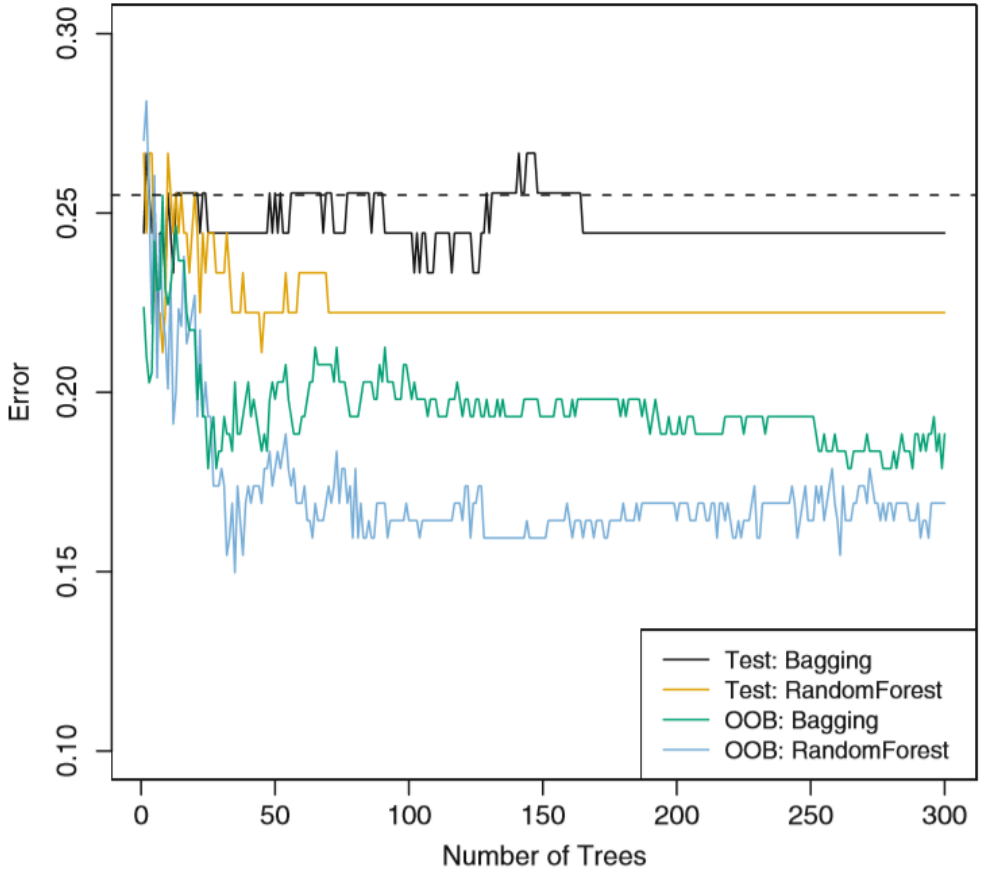
- Bagging review:
  - Typically results in improved accuracy over prediction using a single tree.
  - It is no longer clear which variables are most important to the procedure. Thus, bagging improves prediction accuracy at the expense interpretability.
- The predictions from the bagged trees is highly correlated.
  - Suppose that there is one very strong predictor.
  - Then in the collection of bagged trees, most or all of the trees will use this strong predictor in the top split.
  - Consequently, all of the bagged trees will be highly correlated.
  - Hence, the predictions from the bagged trees will be highly correlated.
- Why is it a problem?
  - Averaging many highly correlated quantities does not lead to a large reduction in variance as averaging many uncorrelated quantities.

Random Forests

- As bagging, we build a number of decision trees on bootstrapped training samples.
- Random forests overcome the bagging problem by forcing each split to consider only a random subset of the predictors.
- The main difference between bagging and random forests is the choice of predictor subset size *m*.
  - Typically, m is chosen to be the square root of p, the number of predictors.
- By randomly leaving out candidate features from each split, random forests *decorrelates* the trees, such that the averaging process can reduce the variance of the resulting model.

Random Forests

- Dashed line indicates the test error resulting from a single tree.
- Test error as a function of B, the number of bootstrapped training sets.
- OOB (Out-of-bag) error



Boosting

- Trees are grown sequentially
  - Each tree is grown using information from previously grown trees.
- Slow learning method
  - In general, statistical learning approaches that learn slowly tend to perform well.

Decision Trees Exercise

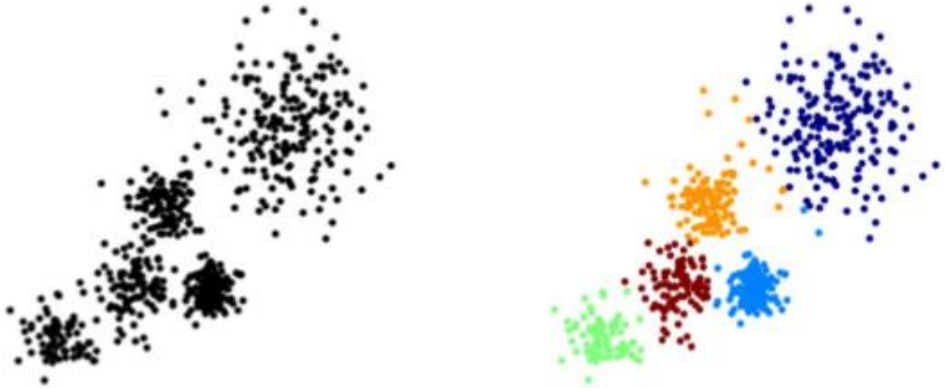
Unsupervised Learning

Unsupervised Learning

- Supervised learning is a well-understood area.
- In unsupervised learning, we only have a set of features $X = (X_1, X_2, \ldots, X_p)$ measured on n observations. We are not interested in prediction, because we do not have an associated response variable Y.
- Rather, the goal is to discover interesting things about the measurements on $X = (X_1, X_2, \ldots, X_p)$
  - *Clustering,* a broad class of methods is used for discovering unknown subgroups in data.
  - Unsupervised learning is often performed as part of an *exploratory data analysis*.
  - It is hard to access the results obtained from unsupervised learning methods.
- Examples:
  - *Online shopping* site might try to identify groups of shoppers with similar browsing and purchase histories, as well as preferences of shoppers within each group.
  - *Search engine* – search results based on click history

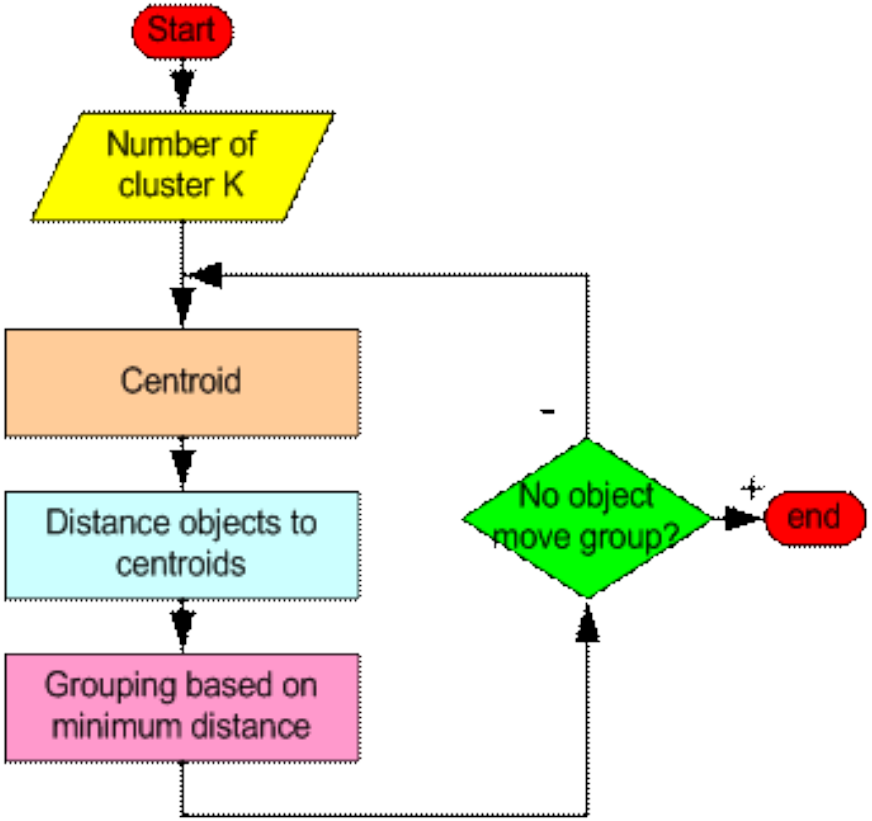
Clustering Methods

- When we cluster the observations of a data set, we seek to partition them into distinct groups so that
  - the observations within each group are quite similar to each other,
  - while observations in different groups are quite different from each other.
- We must define *similar* and *different*.
- Two best-known clustering approaches:
  - K-means clustering
  - Hierarchical clustering



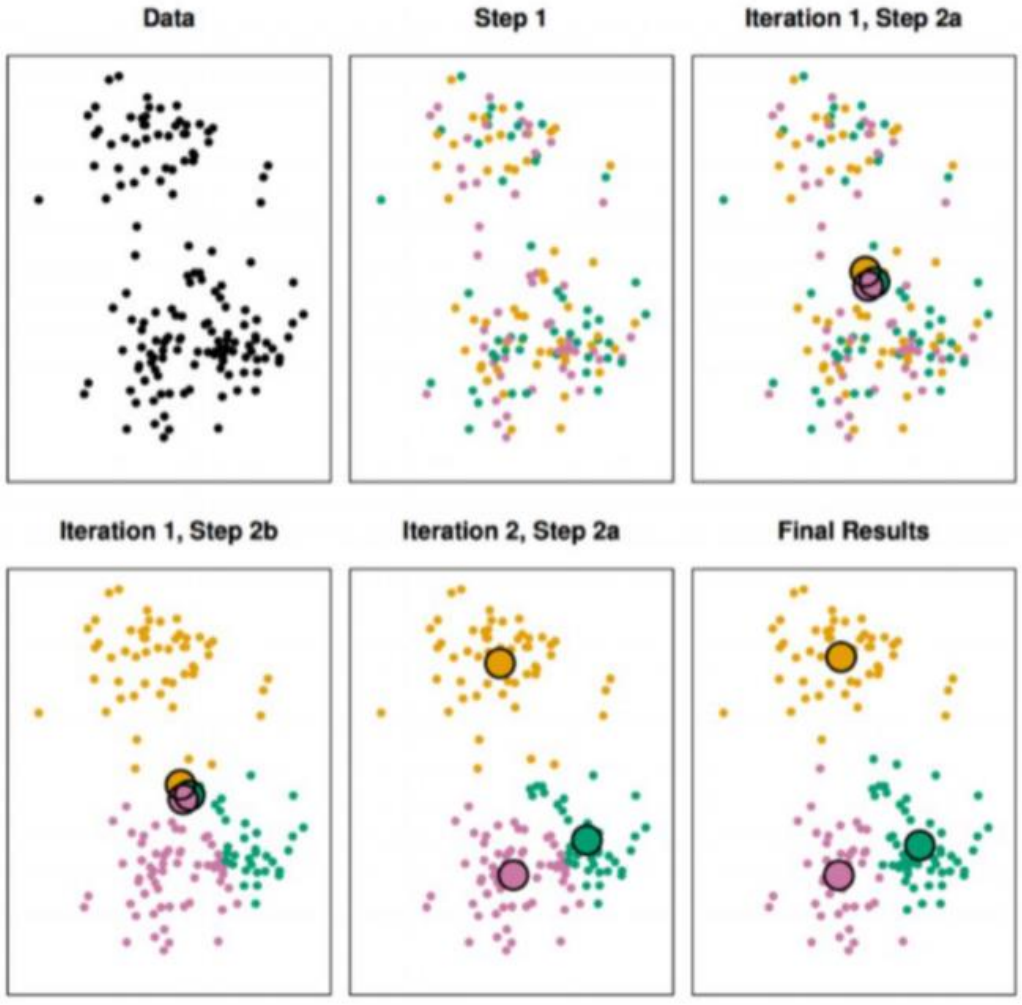
K-Means Clustering

- The K-Means Algorithm
  - Choose a number of Clusters *K*. For example *K* = 3.
  - Randomly assign each point to a cluster
  - Until clusters stop changing, repeat the following:
    - For each cluster, compute the cluster centroid.
    - Assign each data point to the cluster for which the centroid is the closest.

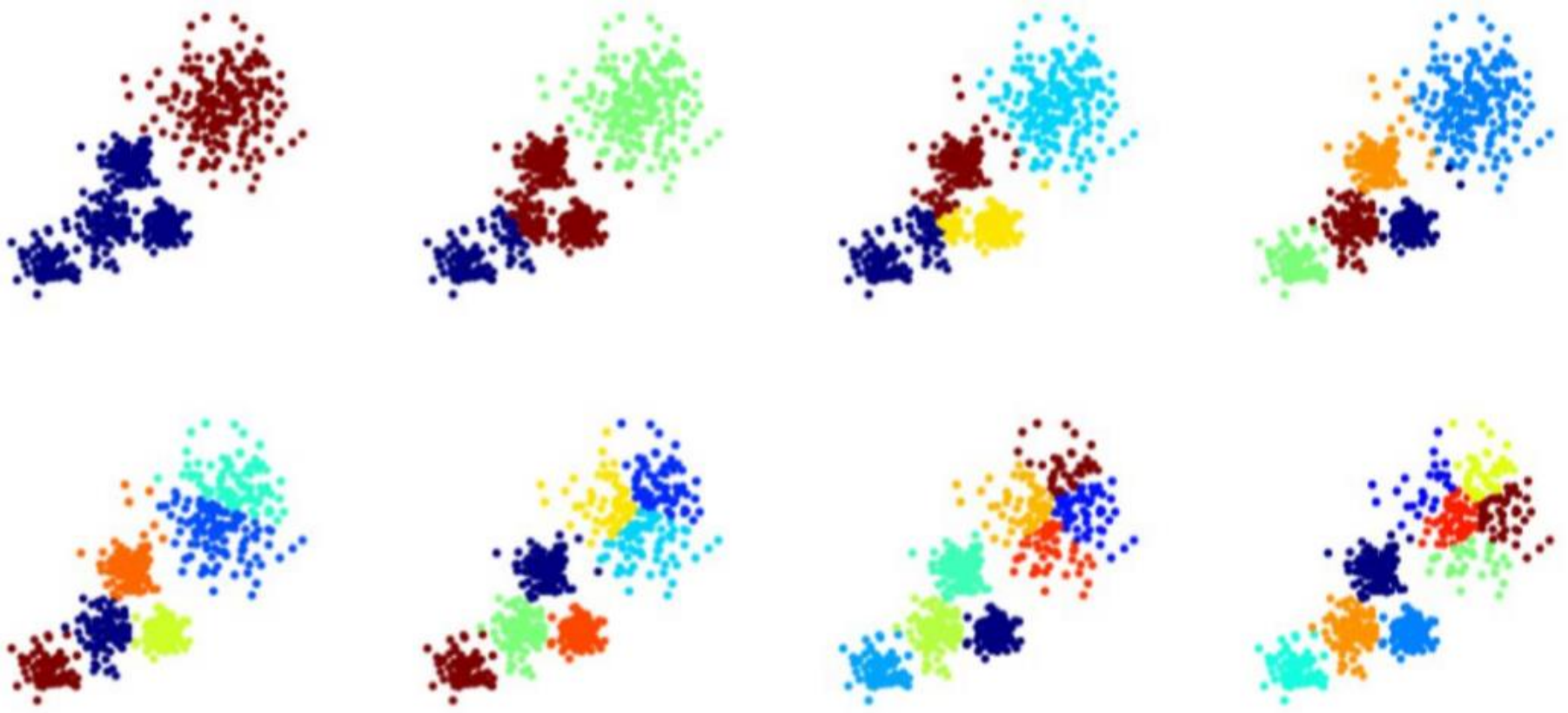


K-Means Clustering

- With K = 3
  - Step 1: Each observation is randomly assigned to a cluster.
  - Step 2a: Cluster centroids are computed. Initially, they are almost overlapping.
  - Step 2b: Each observation is assigned to the nearest centroid.
  - Step 2a (iteration 2) leads to new clusters.
  - The results obtained after 10 iterations.



Choosing a K Value

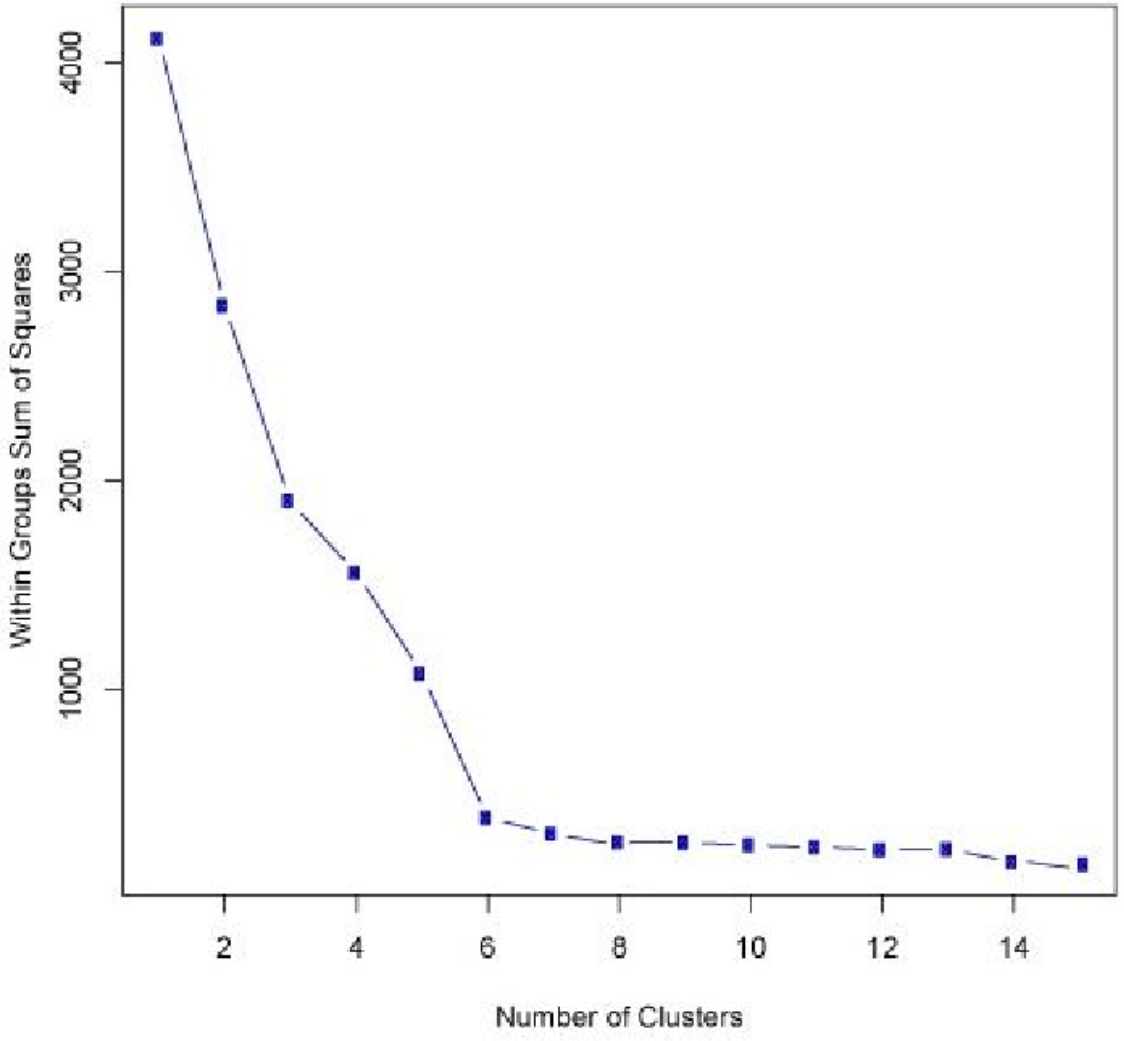


Choosing a K Value

- There is no easy answer for choosing a “best” *K* value.
- One way is the *elbow method*
  - First of all, compute the sum of squared error (SSE) for some values of *K* (for example 2, 4, 6, 8, etc.). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid.
  - If you plot *K* against the SSE, you will see that the error decreases as *K* gets larger; this is because when the number of clusters increases, they should be smaller, so distortion is also smaller.
  - The idea of the *elbow method* is to choose the *K* at which the SSE decreases abruptly.

Choosing a K Value

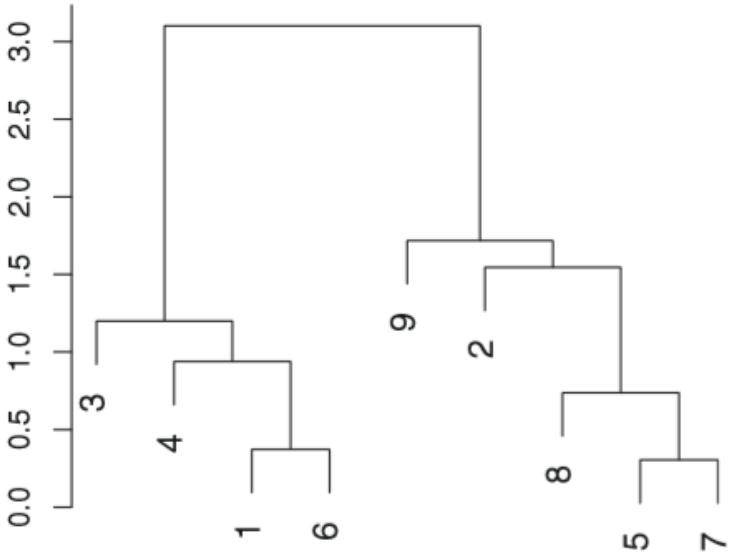
- Elbow method

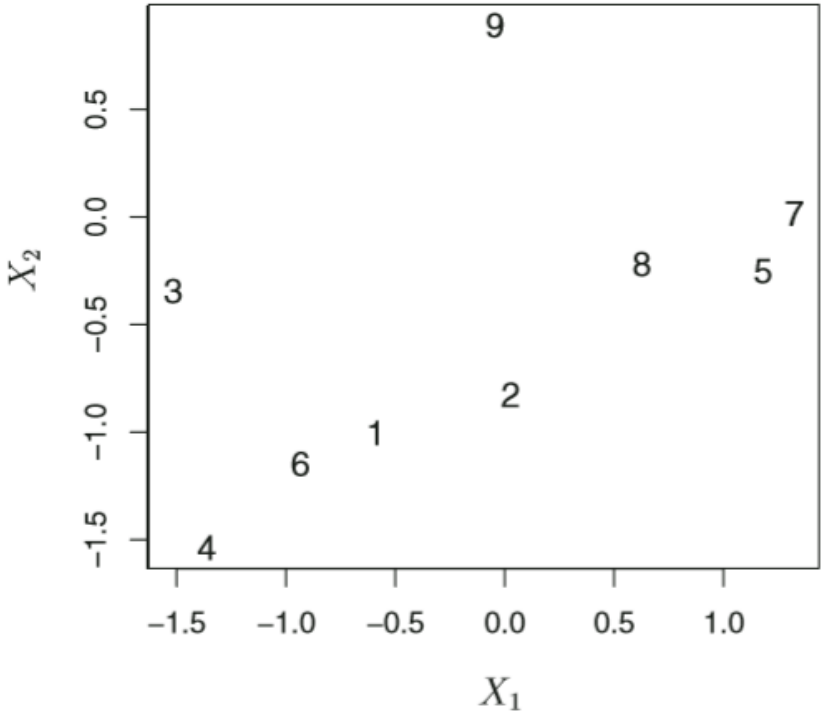


K-Means Clustering Exercise

Hierarchical Clustering

- Attractive tree-based representation of the observations, called a *dendrogram*.



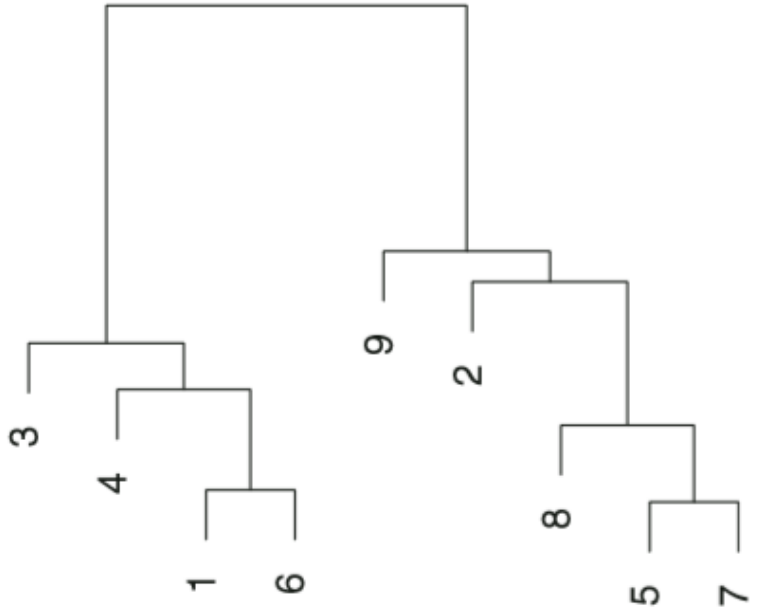


Hierarchical Clustering

- Bottom-up clustering
  - The dendrogram (generally depicted as an upside-down tree) is built starting from leaves and combining clusters up the trunk.
  - The earlier (lower in the tree) fusion occurs, the more similar the groups of observations are to each other.
  - On the other hand, observations that fuse later (near the top of the tree) can be quite different.

Dendrogram Example 1

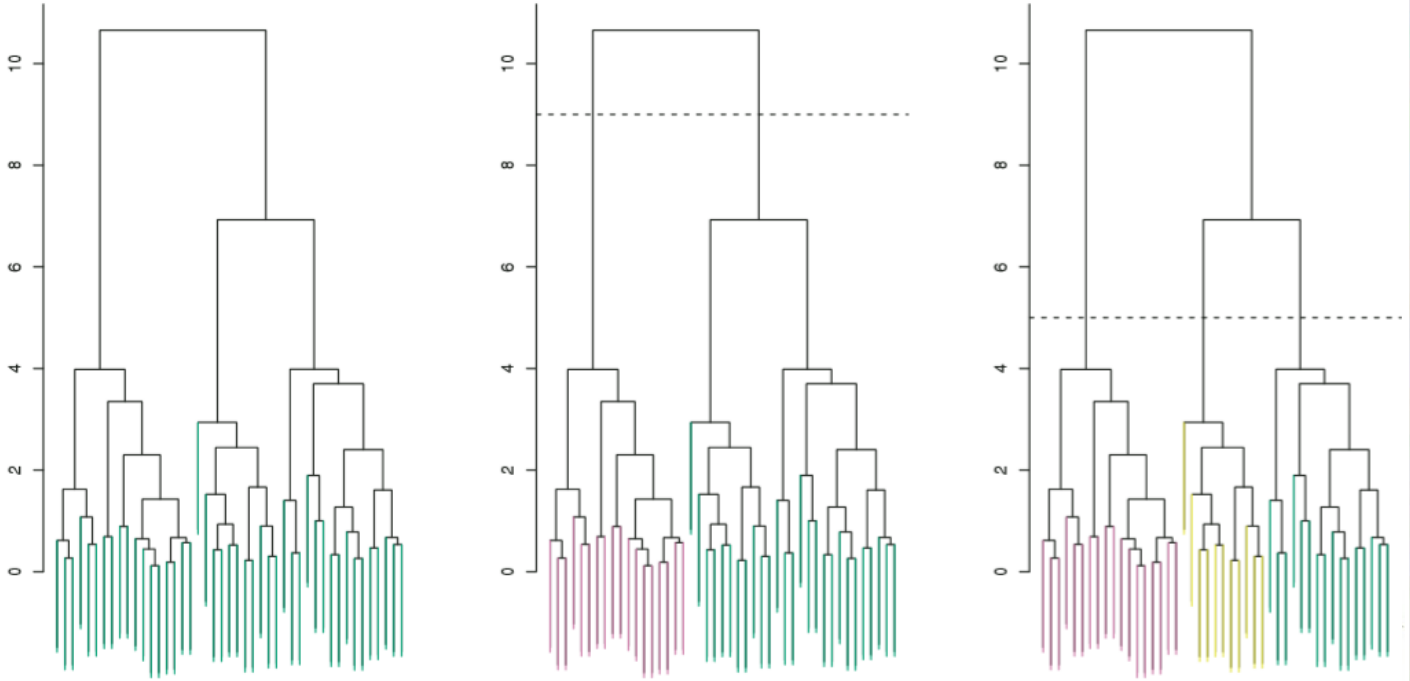
- Observations 5 & 7 are similar, as are observations 1 & 6.
- Observations 2, 8, 5 & 7 all fuse with observation 9 at the same height.
  - Therefore, observation 9 is no more similar to observation 2 than it is to observations 8, 5 & 7.
  - Even though, 9 & 2 are close together in terms of horizontal distance.



Dendrogram Example 2

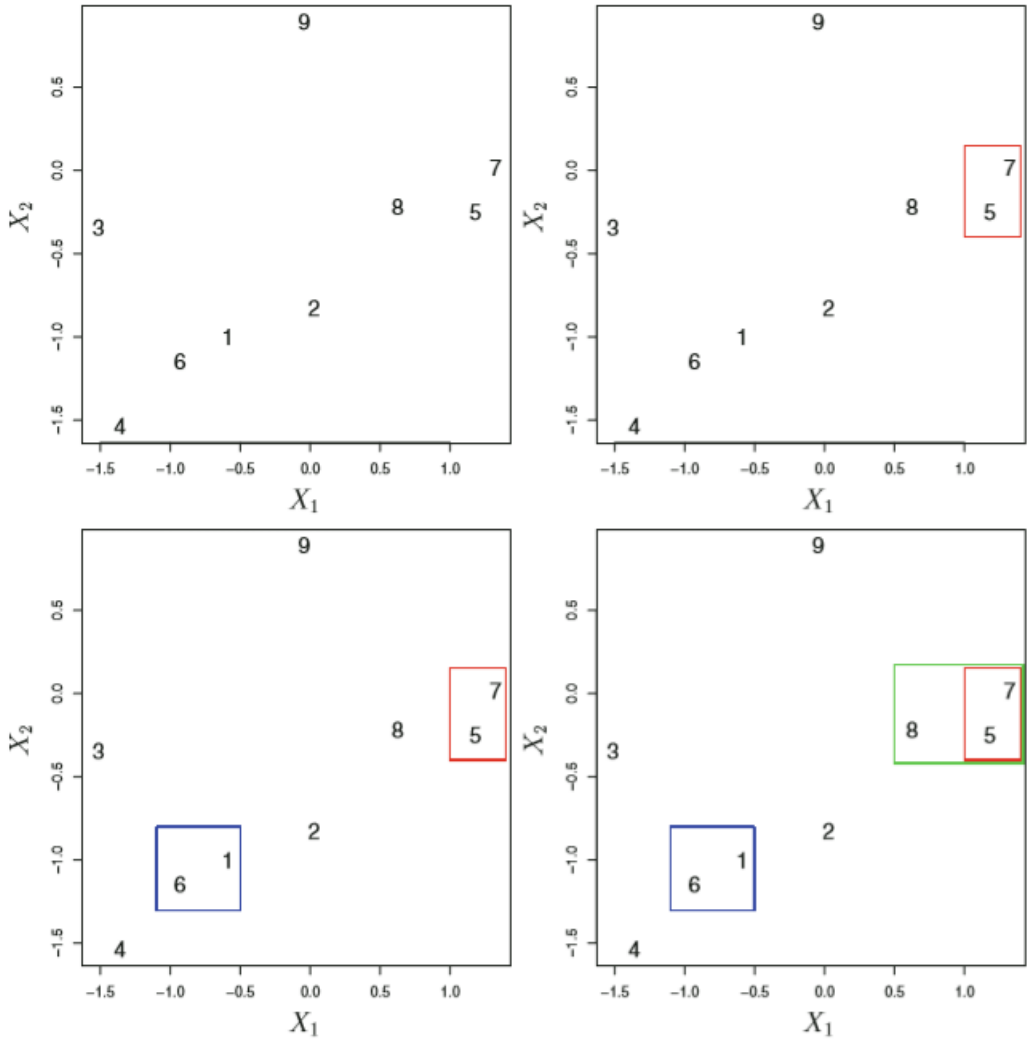
Number of Clusters

- Height of the cut to the dendrogram serves the same role as the K in K-means clustering. It controls the number of clusters obtained.



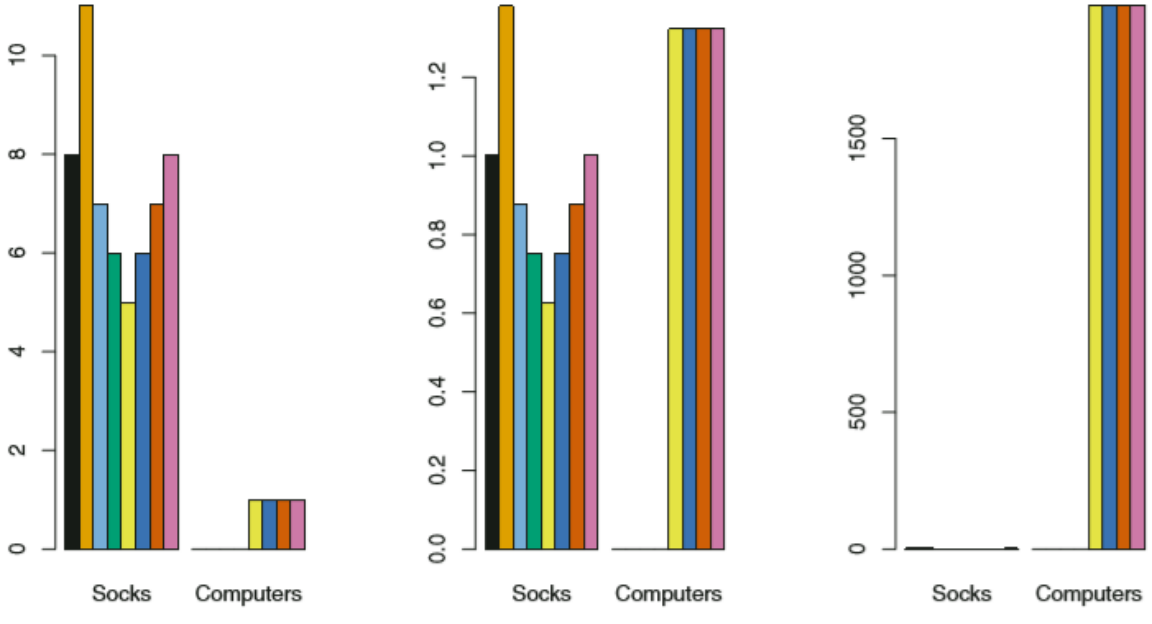
Hierarchical Clustering

- Algorithm
  - Initially, treat each observation as its own cluster.
  - Compute/measure pairwise inter-cluster dissimilarities.
  - Examine all pairwise inter-cluster dissimilarities and identify the pair of clusters that are most similar. Fuse them together in a single cluster.
  - Repeat.



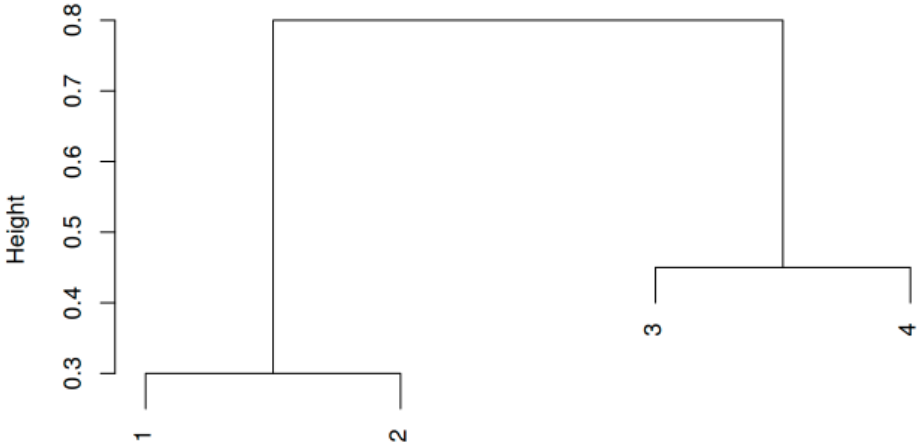
Choice of Dissimilarity Measure

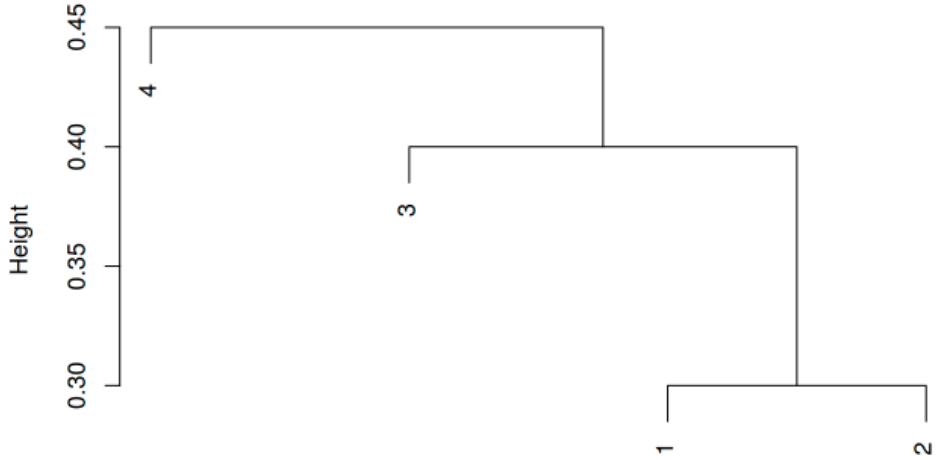
- Type of data should determine the dissimilarity measure.
  - For example, consider an online retailer interested in clustering shoppers based on their past shopping histories.
  - The goal is to identify subgroups of similar shoppers.



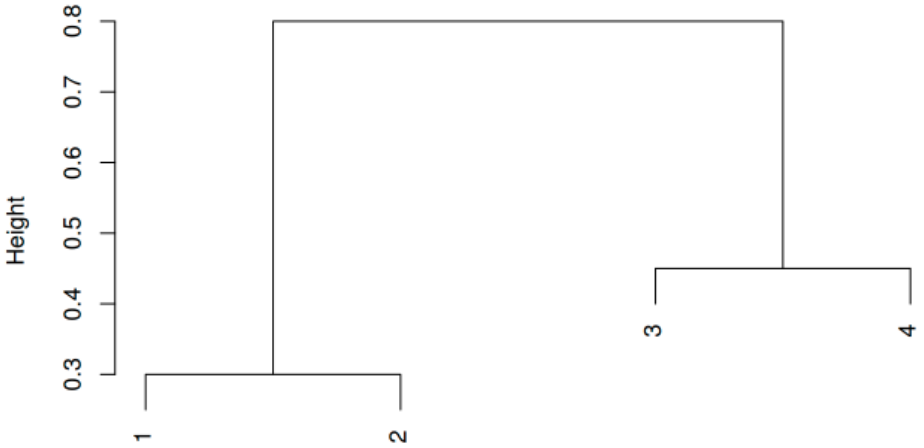
Hierarchical Clustering Exercise

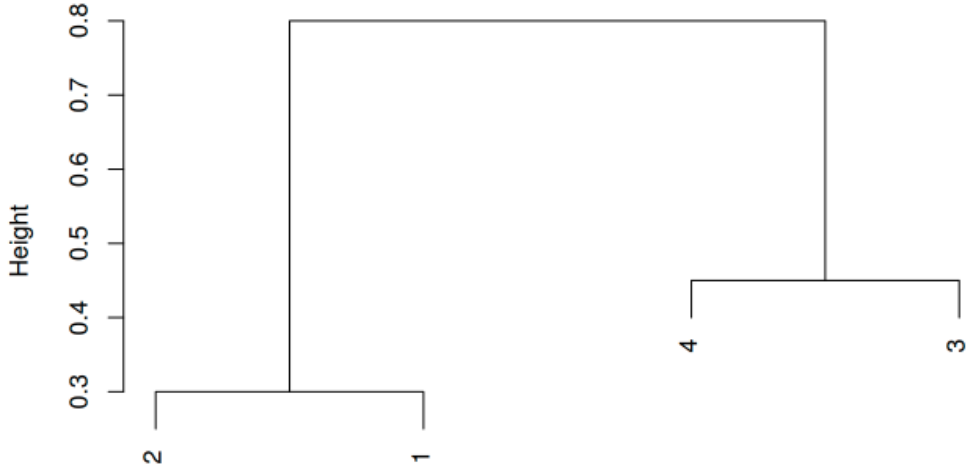
Which Dendrogram?





Which Dendrogram?





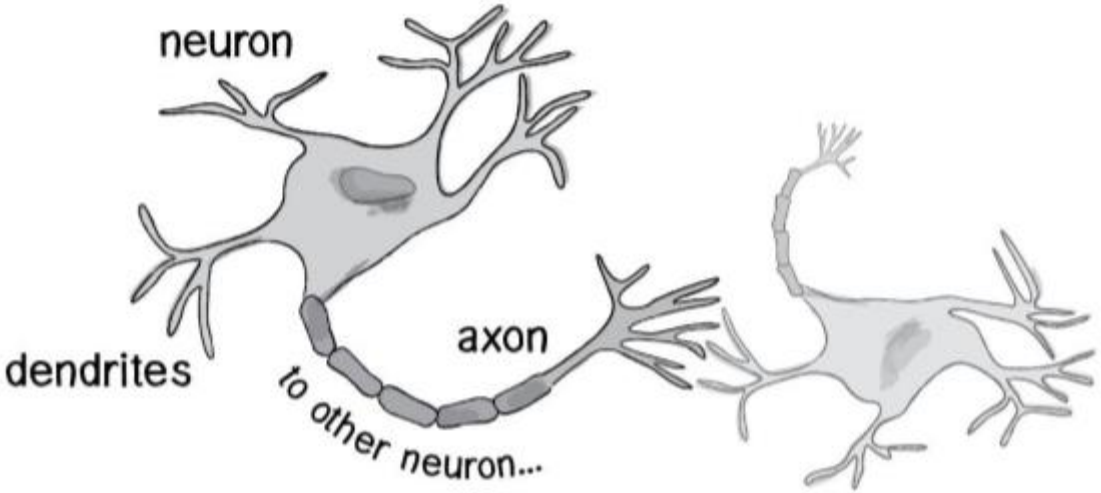
Summary

- What is data science?
- Data types – Vectors, matrices and data frames
- Programming in R
- R Packages:
  - Dplyr for **data manipulation**
  - Tidyr for **data cleaning**
  - Ggplot2 and Plotly for **data visualization**
- What is machine learning?
- Regression problem – linear regression, regression trees
- Classification problem – KNN classifier, classification trees
- Unsupervised learning – K-Means and hierarchical clustering

Reinforcement Learning

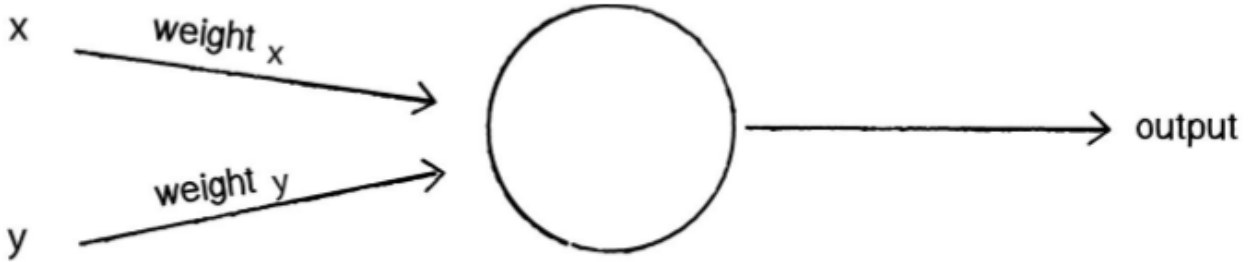
Neural Networks

- Neural networks are modeled after biological neural networks and attempt to allow computers to learn in a similar manner to humans - reinforcement learning.
- The human brain has interconnected neurons with dendrites that receive inputs, and then based on those inputs, produce an electrical signal output through the axon.



Perceptron

- Let's start by looking at the simplest Neural network possible - the perceptron.
  - A perceptron consists of one or more inputs, a processor, and a single output.
  - Each input that is sent into the neuron must first be weighted, i.e. multiplied by some value.
  - The output of a perceptron is generated by passing that sum through an activation function (Logistic, Trigonometric, Step, etc…). In the case of a simple binary output, the activation function is what tells the perceptron whether to "fire" or not.

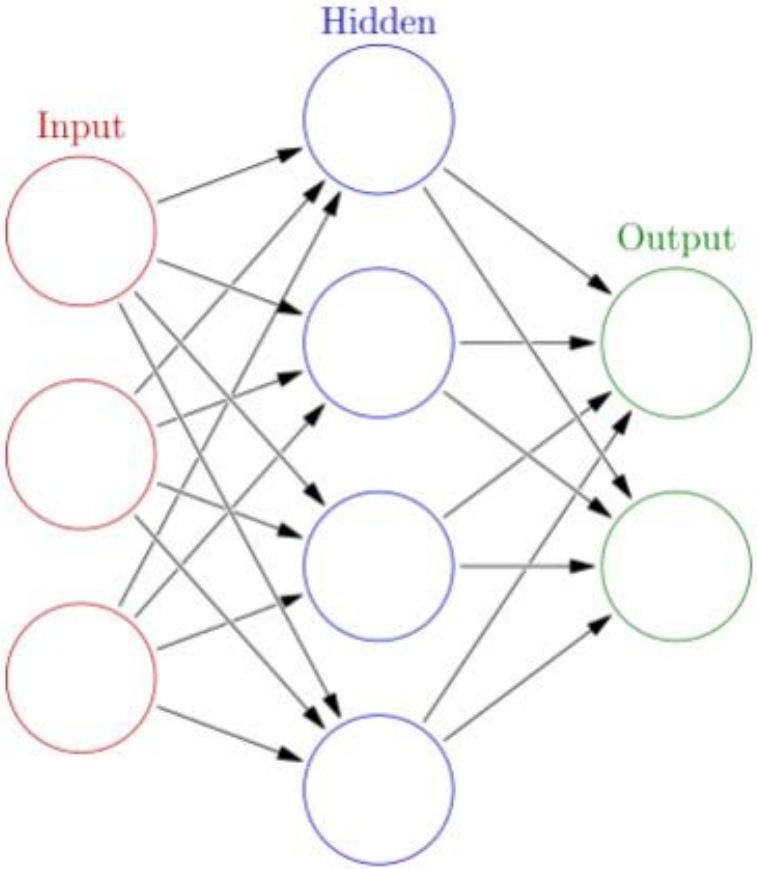


Perceptron

- To actually train the perceptron we use the following steps:
  - Provide the perceptron with inputs for which there is a known answer.
  - Ask the perceptron to guess an answer.
  - Compute the error. (How far off from the correct answer?)
  - Adjust all the weights according to the error.
  - Return to Step 1 and repeat!
- We repeat this until we reach an error we are satisfied with (we set this before hand).
- That is how a single perceptron would work. Now, to create a neural network all you have to do is link many perceptrons together in layers!

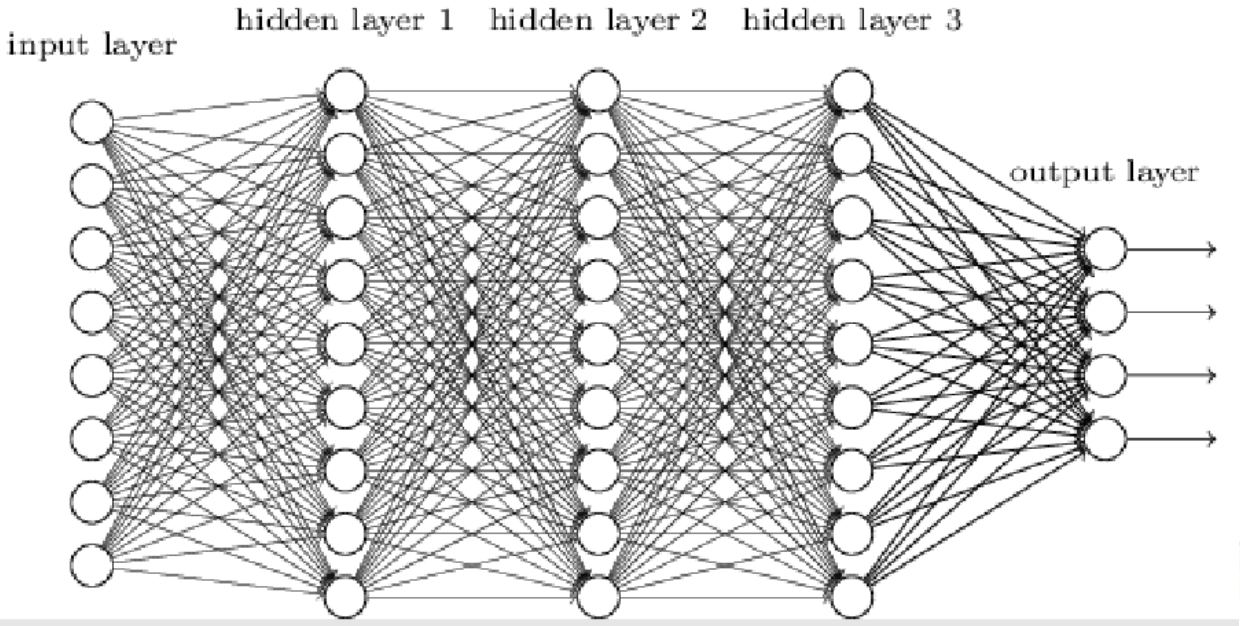
Neural Networks

- You'll have an input layer and an output layer.
- Any layers in between are known as hidden layers, because you don't directly "see" anything but the input or output.



Neural Networks

- You may have heard of the term **Deep Learning**. That's just a Neural Network with many hidden layers, causing it to be "deep". For example, Microsoft's state of the art vision recognition uses 152 layers.



Machine Learning Bootcamp

Introduction to Data Science

Tinkers Club
THINK. MAKE. LEARN