IST 5535: Machine Learning Algorithms and Applications

Langtao Chen, Spring 2021

1. Introduction to Machine Learning

Reading

- Book Chapters 1, 2 (sections 2.1, 2.2)

- Online Article: Statistics – Understanding the Levels of Measurement
  - http://www.kdnuggets.com/2015/08/statistics-understanding-levels-measurement.html

Learning Objectives

- Explain important concepts related to machine learning
- Understand dataset and be able to distinguish among different scales of measurement
- Explain methods used to assess model accuracy
- Explain bias-variance trade-off

OUTLINE

- (I) Overview of machine learning (ML)
  1. What is learning?
  2. Practical definition of ML
  3. ML model estimation methods: parametric, nonparametric
  4. Types of ML

- (II) Scale of measurement

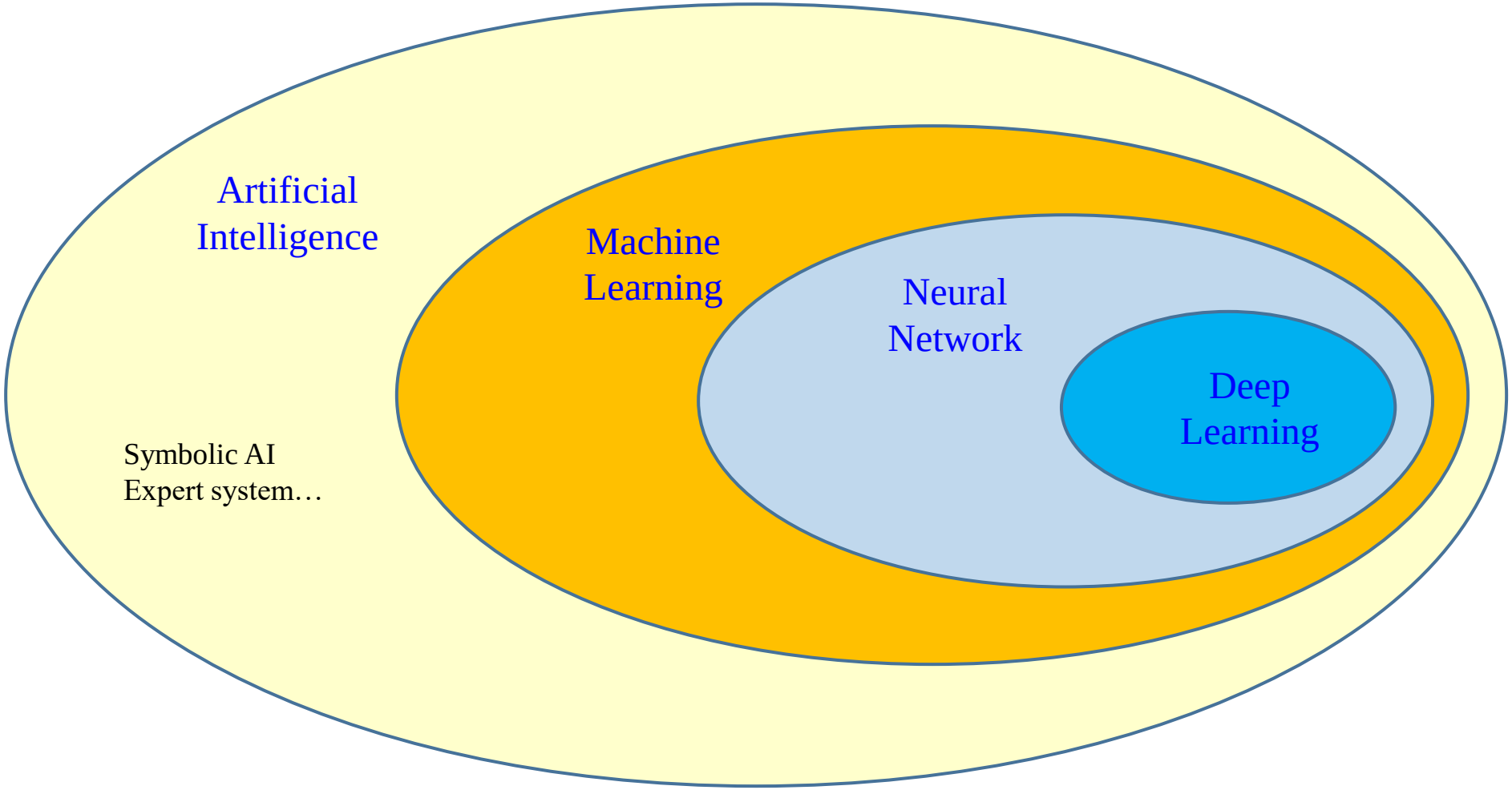
  Nominal, ordinal, interval, ratio

- (III) Model accuracy
  1. Regression setting: MSE, training MSE, test MSE
  2. Classification setting: Error rate
  3. Bias variance tradeoff

Agenda

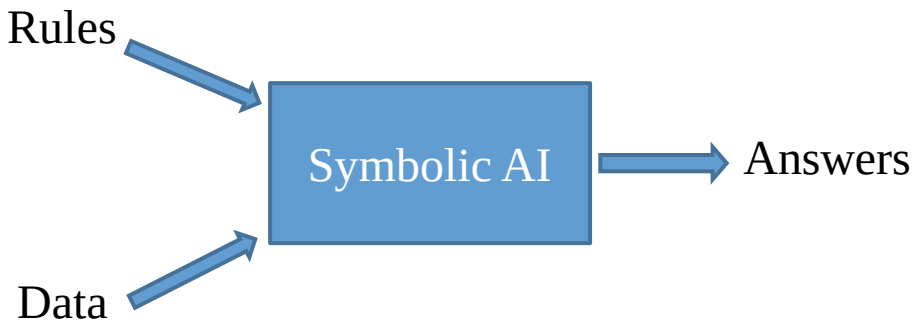
- Overview of Machine Learning
- Dataset and Scales of Measurement
- Assessing Model Accuracy

AI, Machine Learning, Neural Network, and Deep Learning

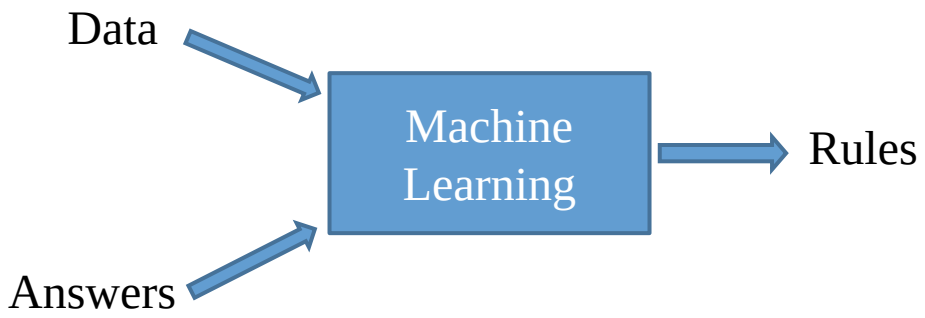


What is a Learning?

- Symbolic AI



- Machine Learning



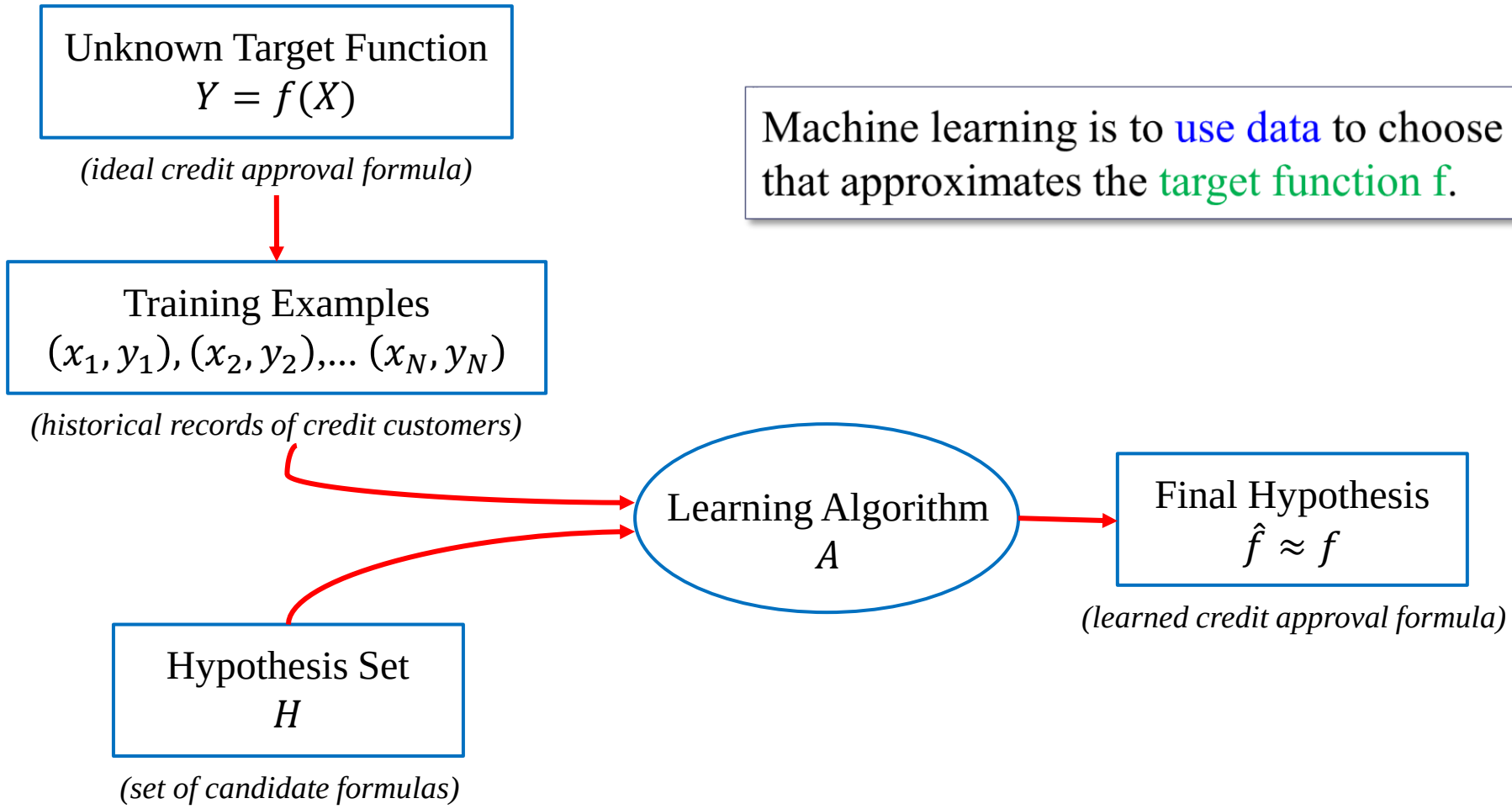
Central Research Questions of Machine Learning

- How can we build computer systems that automatically improve with experience?
- What are the fundamental laws that govern all learning processes?

> “Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed”.
>
> ----Wikipedia

Practical Definition of Machine Learning

Basic Setup of the learning problem (adapted from Abu-Mostafa et al 2012)



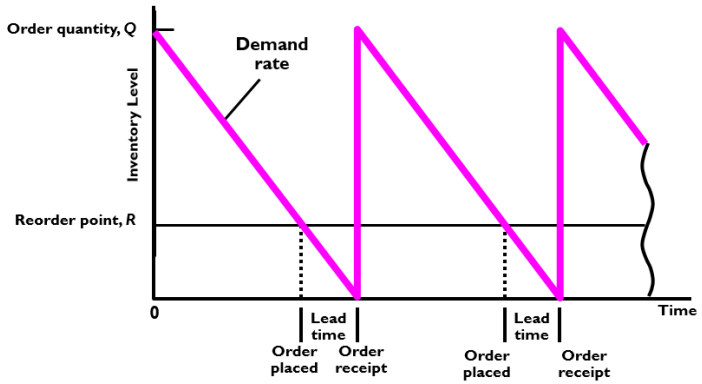
Machine learning is to use data to choose a hypothesis $\hat{f}$ that approximates the target function f.

When do We Need Machine Learning?

Some problems have analytic solutions

- What is the optimal ordering quantity in order to minimize the total inventory cost?

EOQ model: $Q_{opt} = \sqrt{\frac{2C_oD}{C_c}}$



Only empirical solutions are feasible

- How can we classify an email as either spam or ham?

When there is no analytic solution but we do have a lot of data, we can use machine learning methods to construct an empirical solution from the data.

Learning = Representation + Evaluation + Optimization

- Representation: Formal language used to represent a learning algorithm
- Evaluation: Assess the performance of algorithms
- Optimization: Search the optimal solutions

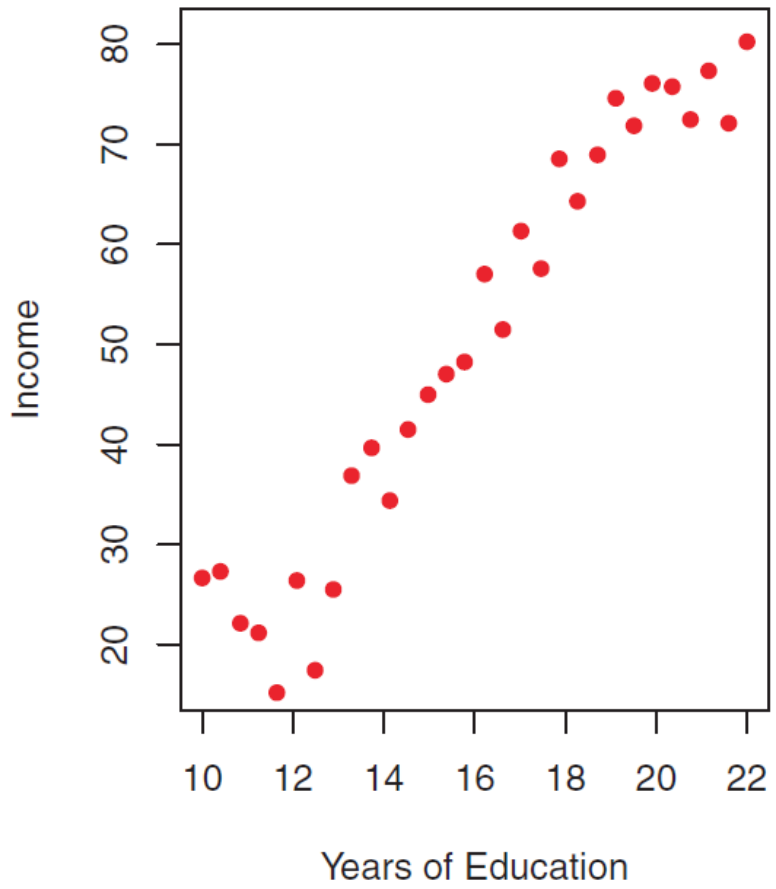
Table 1: The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Source: Pedro Domingos, "A Few Useful Things to Know about Machine Learning"

An Example of Machine Learning

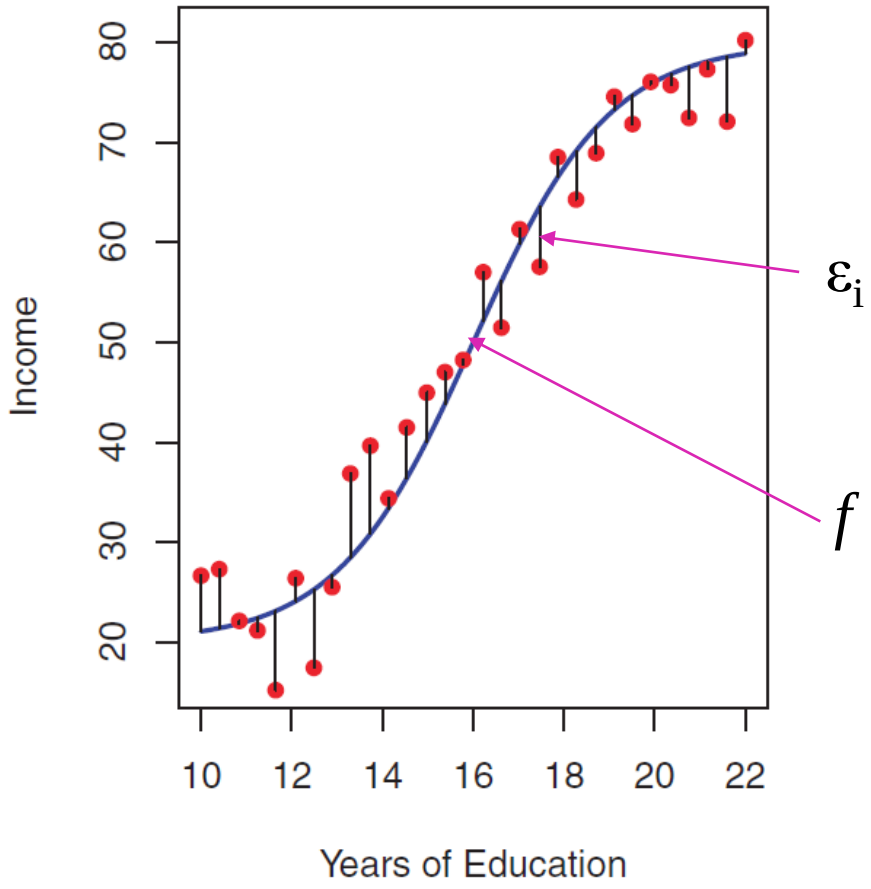
- Observed pattern



Machine Learning

$\hat{f} \approx f$

- True underlying relationship



Why Do We Estimate f ?

- Machine learning is all about estimating the unknown function f .

- Two major reasons for estimating f : $\hat{Y} = \hat{f}(X)$
  - Prediction
    - If $\hat{f}$ approximates f well, we can accurately predict Y based on new value of X.
    - $\hat{f}$ is often treated as a black box.
  - Inference
    - We are interested in understanding the relationship between X and Y.
    - $\hat{f}$ should be a white box. We need to know its exact form.

How Do We Estimate *f* ?

- Parametric methods: Reduce the problem of estimating f down to one of estimating a set of parameters.
- A two-step model-based approach
  - Step 1: Make assumption about the functional form, or shape, of *f*

    For example, assume linear relationships (linear model)

    $$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

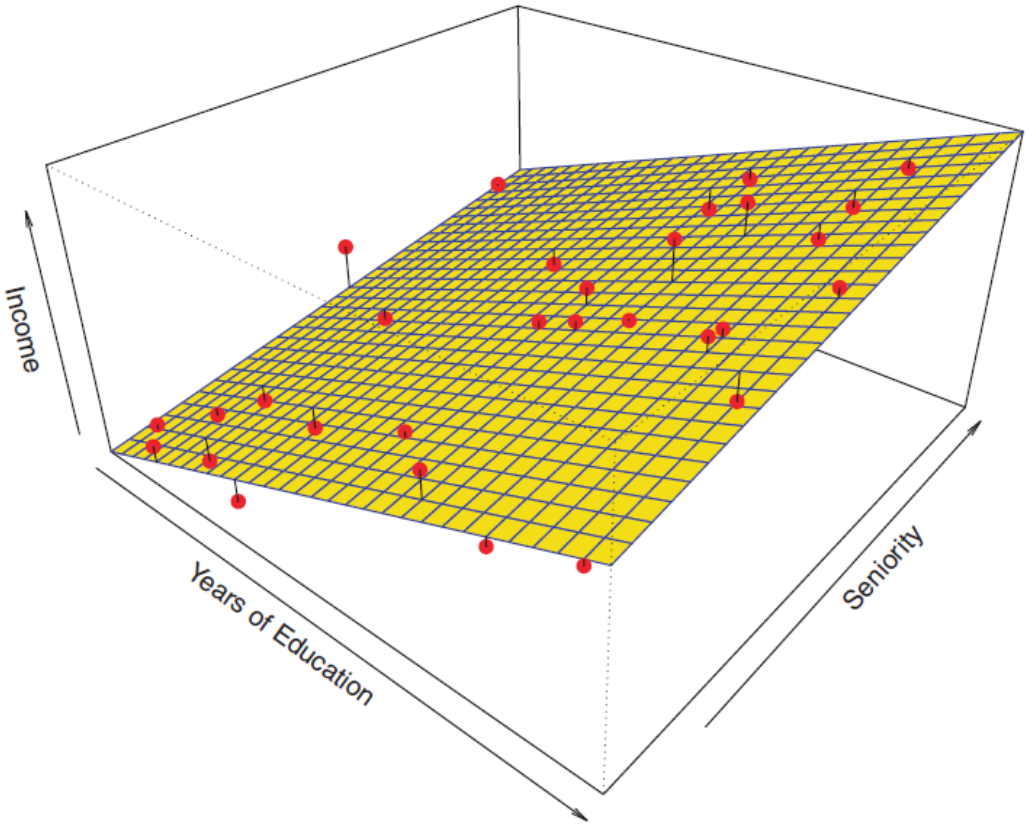
  - Step 2: Use a procedure that uses the training data to *fit* or *train* the model

    For example, use ordinary least square (OLS) or maximum likelihood (ML) to estimate the parameters $\beta_0, \beta_1, \beta_2, \ldots \beta_p$

An Example of Parametric Method

- A linear model fit by OLS to the income data

$$Income \approx \beta_0 + \beta_1 \times education + \beta_2 \times seniority$$



The true f has some curvature that is not captured in the linear fit

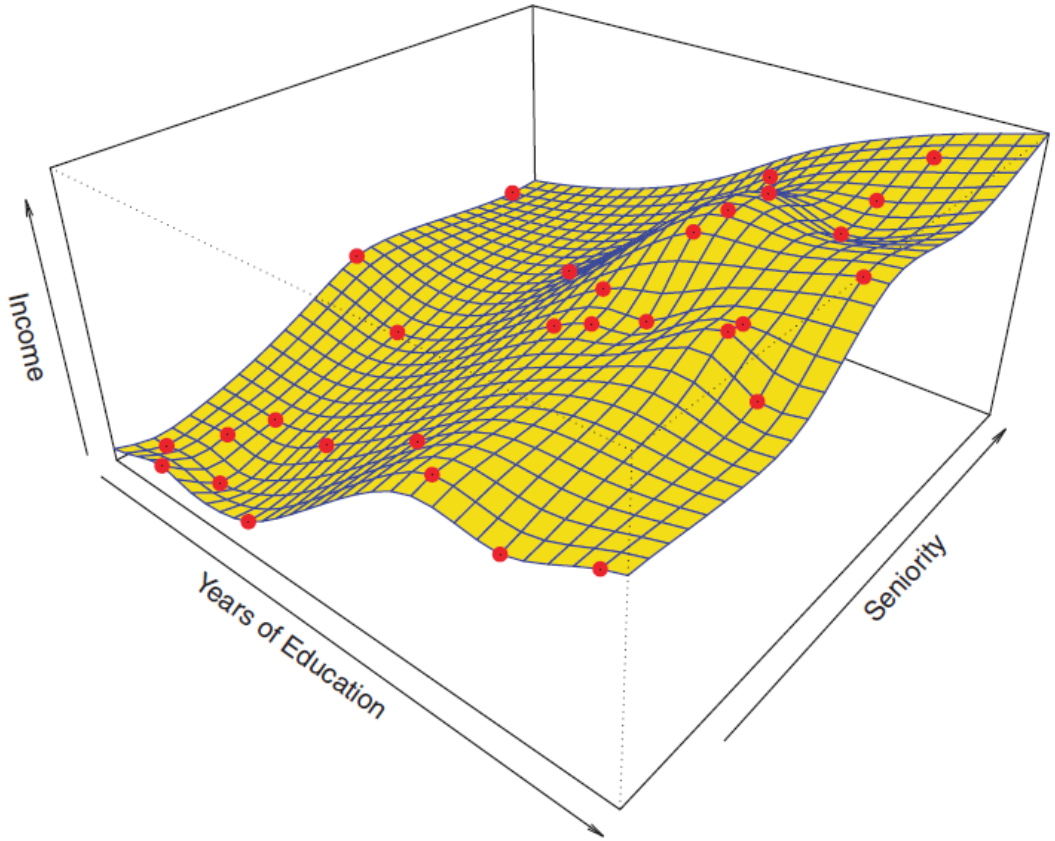
How Do We Estimate *f* ?

- Non-parametric methods
  - No explicit assumptions about the functional form of *f*
  - Advantage: Have the potential to accurately fit a wide range of possible shapes of *f*
  - Disadvantage: A large number of observations is required in order to obtain an accurate estimate of *f*

An Example of Non-parametric Method

- A rough thin-plate spline fit to the income data

This fit makes zero errors in the training data



However, there is likely an overfitting issue: The fitted model will not yield accurate estimates of the response on new data.

Why are there so many machine learning algorithms?

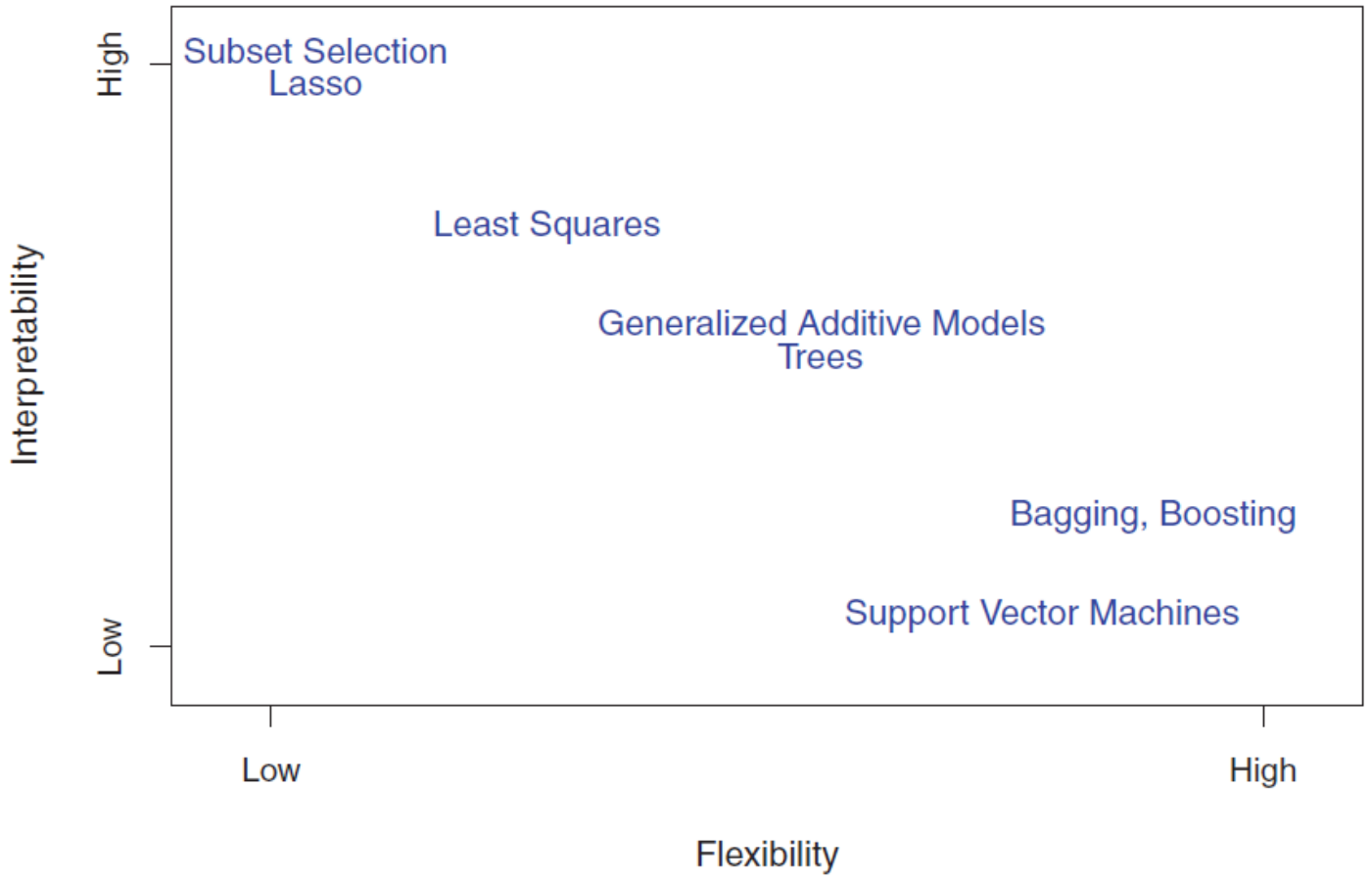
- “No free lunch theorem”

  There is no such a single algorithm that is uniquely better for all problems.

- So we’ll learn a couple of important machine learning algorithms in this class.

Tradeoff between Prediction Accuracy and Model Interpretability

- In general, as the flexibility of a method increases, its interpretability decreases



Tradeoff between Prediction Accuracy and Model Interpretability

- Why would we prefer a more restrictive model over a very flexible model?
  - If we are mainly interested in inference, restrictive models are much more interpretable;
  - Even when inference is NOT the goal, less flexible models are not likely to overfit the data, thus often providing more accurate estimate.

Types of Learning

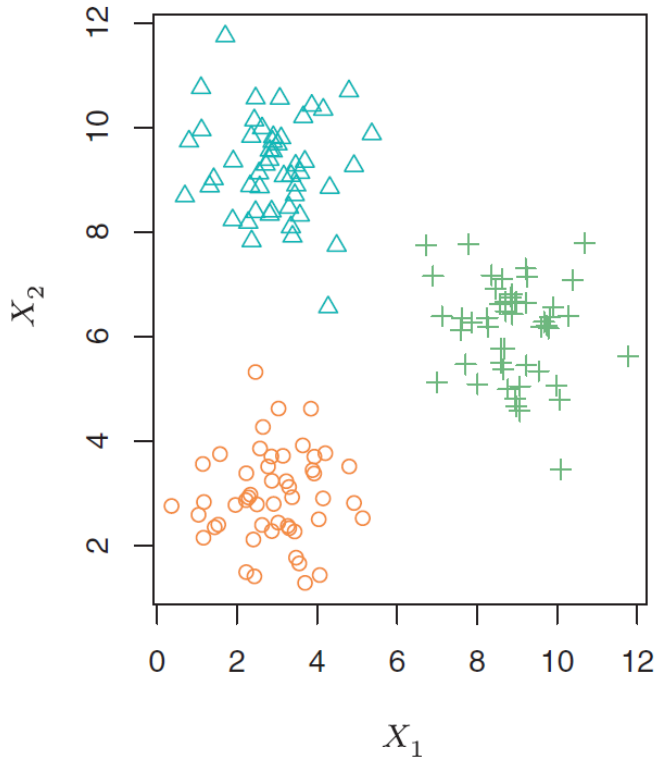
- Supervised Learning
  - Supervised learning algorithms are used for prediction and classification.
  - We need to supervise the learning of the algorithm by using training data to train the algorithm.
  - Data are labeled with correct output: (X_i, y_i), i=1...N
  - The most studied type of learning

Types of Learning

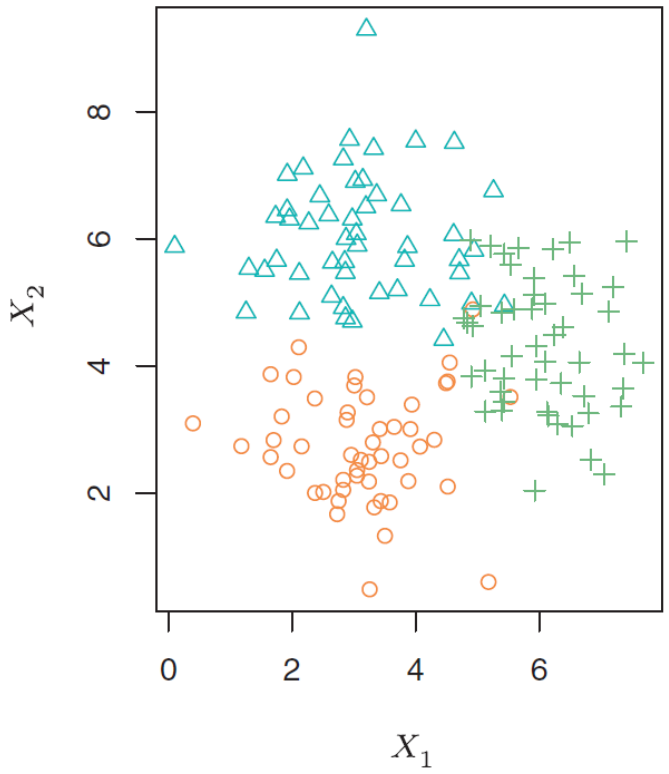
- Unsupervised Learning
  - Unsupervised learning algorithms are used when there is no outcome variable to predict or classify. We simply learn something from the inputs by themselves.
  - Data are unlabeled: only $X_i (i=1...N)$ are observed
  - There is no training-testing partition of the dataset.
  - Popular scenarios include association rules and clustering.

Types of Learning

- Unsupervised Learning Example



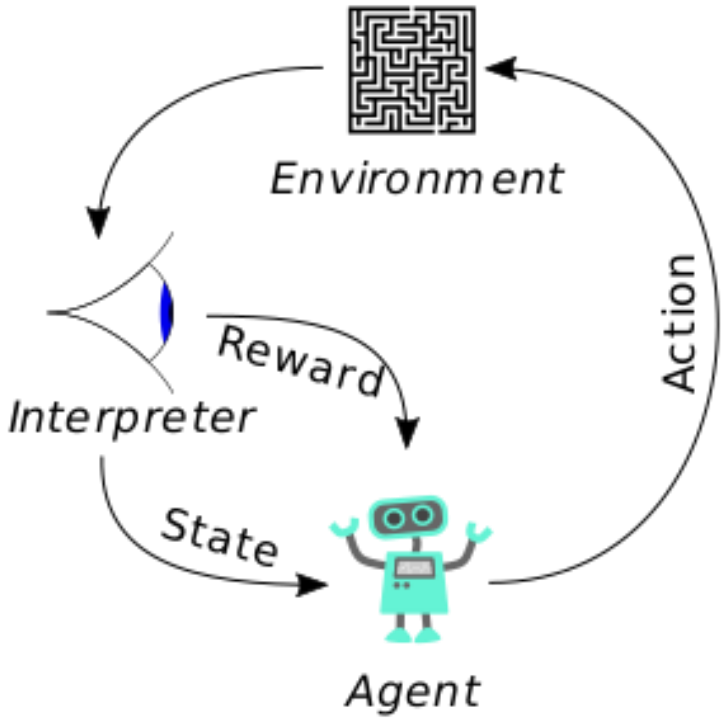
3 groups are well-separated



Overlap among 3 groups.
Clustering is more challenging.

Types of Learning

- Reinforcement Learning
  - To let computer agent learn like people, without godlike “supervisor” providing correct output.
  - Data is unlabeled, but contain some possible outputs with their goodness scores.
  - The agent learns from experience through trial-and-error.
  - The goal is to maximize long-term reward.



An agent takes actions in an environment, which is interpreted into a reward and a representation of the state, which are fed back into the agent.

Applications of Reinforcement Learning

- Game play: AlphaGo trumped a human Go campion in 2016

Photo: Google

Applications of Reinforcement Learning

- Self-driving cars

Model-Based Learning Vs. Instance-Based Learning

Model-Based Learning

- Model-based learning tries to build a model y=f(x) from training data and then use the model to generalize to new problem.
- Usually model training is computationally intensive, while prediction is easy and simple.
- Examples: regression, neural network, hidden Markov model…

Instance-Based Learning

- Instance-based learning compares new problem instances with the instances seen in the training data.
- Classification or prediction is postponed when the new instance needs to be evaluated. Usually prediction stage is computationally intensive. Sometime called as lazy learning.
- Examples: k-nearest neighbors, support vector machines…

Regression Versus Classification Problems

- Supervised machine learning problems can be categorized as regression or classification problems.

- Regression problems: when the response variable is quantitative.
  - What is the customer demand of product ABC in the next month?

- Classification problems: response variable is qualitative or categorical.
  - Will a customer churn her service? (yes or no, binary response)
  - Will a customer default on a debit? (yes or no, binary response)

Agenda

- Overview of Machine Learning
- Dataset and Scales of Measurement
- Assessing Model Accuracy

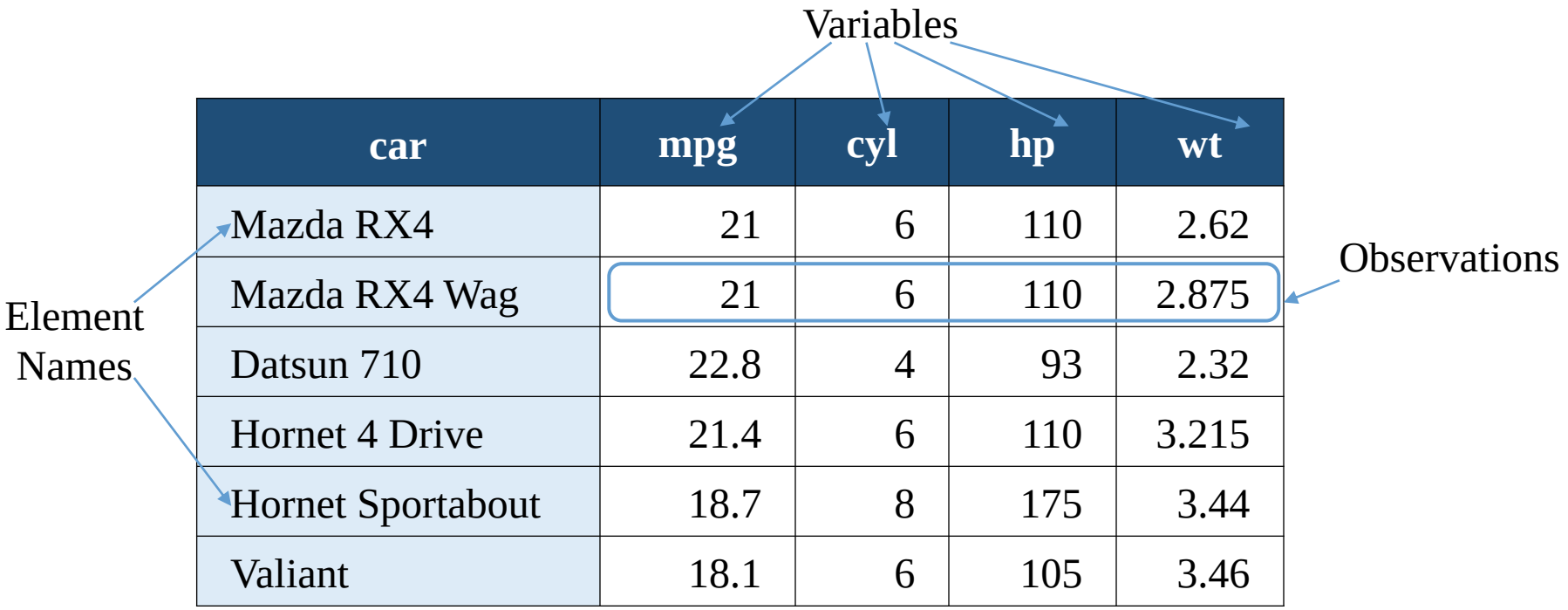
Data and Data Set

- Data are the facts collected, analyzed, and interpreted.

- The data collected in a particular data science project are commonly referred to as a data set.

Data Set: Elements, Variables, and Observations

- Elements/subjects: entities of interest
- Variables: characteristic of elements
- Observation: the set of measurements obtained for an element



car	mpg	cyl	hp	wt
Mazda RX4	21	6	110	2.62
Mazda RX4 Wag	21	6	110	2.875
Datsun 710	22.8	4	93	2.32
Hornet 4 Drive	21.4	6	110	3.215
Hornet Sportabout	18.7	8	175	3.44
Valiant	18.1	6	105	3.46

Scales of Measurement

- Scale/level of measurement determines:
  - the amount of information contained in data
  - data summarization and analysis methods that are appropriate

- Four types of scales
  - Nominal
  - Ordinal
  - Interval
  - Ratio

Qualitative / categorical data

Quantitative data

Nominal Scale

- Numerical values are just names or labels of the attribute
  - Ordering of these values is meaningless
  - No mathematical calculation (+, -, *, /) applicable

- For example:
  - Gender (1 = “Male”, 0 = “Female”)
  - Student ID (1,2,3...)
  - Department (1 = “BIT”, 2 = “CS”...)
  - Zip code (65401, 65402...)

Ordinal Scale

- Attributes can be ranked/ordered.

- For example:
  - Football team rank (1st, 2nd, 3rd...)
  - Customer rating (1 = “Bad”, 2 = “OK”, 3 = “Excellent”)

Interval Scale

- Have all characteristics of ordinal scale
- Distance between attributes does have meaning.
- Ratios are not meaningful.

- For example:
  - Temperature
    - The distance from 40 – 60 is same as the distance from 60 – 80
    - 80 cannot be said as twice hot as 40
  - SAT Score
  - GMAT Score

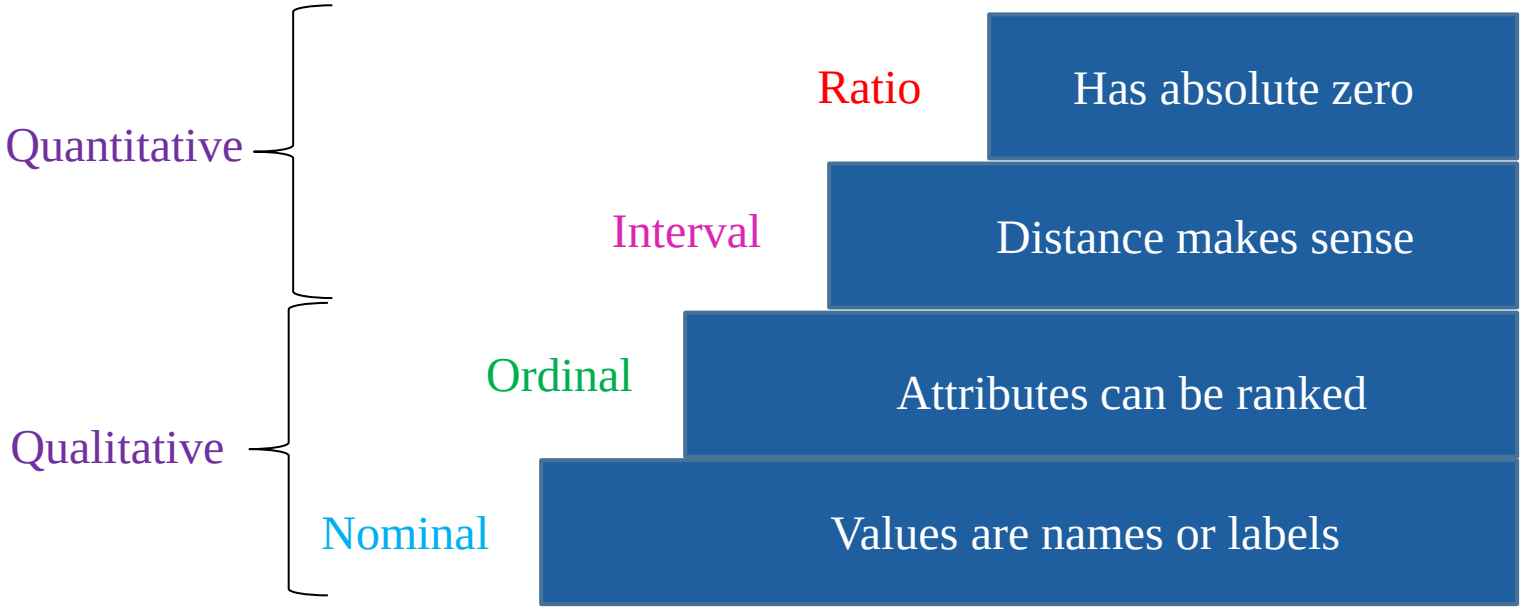
Ratio Scale

- Have all characteristics of interval scale
- A ratio of two values is meaningful.
- An absolute zero is meaningful.

- For example:
  - Weight
  - Height
  - Distance
  - Number of visits
  - Credit hours earned

Hierarchy of Measurement Scales

- A higher level scale contains all properties of its lower scale.
- From lower to higher levels, analysis tends to be more comprehensive. Improper use of lower level scales suffers information loss in the data
- In general, we prefer a higher scale of measurement than a lower one.



Exercise

- Decide the scales of measurement for the following columns:

A sales summary of two stores by operating hour

Store#	City	Hour	Sale
101	Rolla, MO	9	$1,000
101	Rolla, MO	10	$1,100
101	Rolla, MO	11	$1,200
102	St. Louis, MO	9	$3,000
102	St. Louis, MO	10	$3,300
102	St. Louis, MO	11	$4,000

Note: In the hour column, 9 means time between 9AM and 10AM.

Things Could be Tricky: Measurement Scale for Year

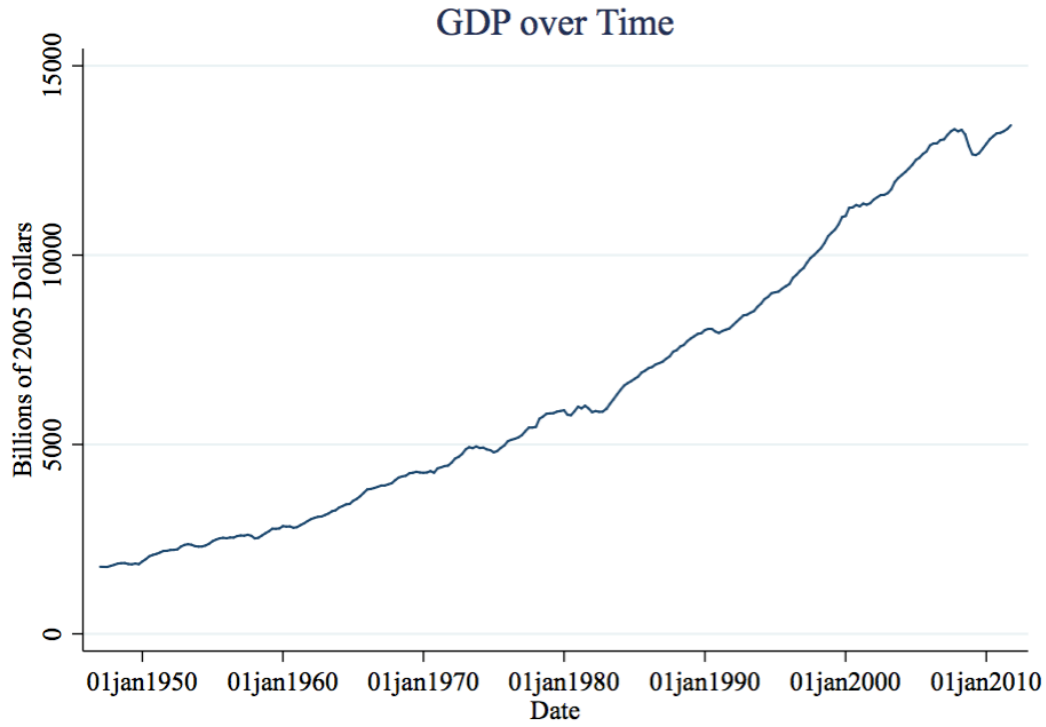
- GDP has an increasing trend over time

- We want to predict world GDP in 2020 using the following regression model:

$$GDP_{year} = \beta_0 + \beta_1 * year$$

What is the scale of measurement for *year*?

Ratio



Things Could be Tricky: Measurement Scale for Year

- It seems a consumer's spending is partly determined by his/her income.
- We collected a dataset of annual spending and revenue from 100 consumers across a 5-year period from 2011 to 2015.
- We want to estimate the effect of annual income on annual spending by controlling for possible time effect.

$$Spend_{i,t} = \beta_0 + \beta_1 * Income_{i,t} + \theta * year\_dummies$$ => A panel regression

where t=1,2,...5, i = 1, 2,..., 100

What is the scale of measurement for *year*?

Nominal

Agenda

- Overview of Machine Learning
- Dataset and Scales of Measurement
- Assessing Model Accuracy

Why Do We Need to Assess Model Accuracy?

- There are so many different statistical learning approaches.
- *There is no free lunch in statistics*: no one method dominates all others overall all possible dataset.
- On a particular dataset, one specific method may work best.
- Selecting the best approach can be a challenge in practice

Assessing Model Accuracy

- Measuring the quality of fit
- The classification setting
- The bias-variance trade-off

Measuring the Quality of Fit

- In regression setting, one commonly used measure is the *mean squared error* (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the prediction that a learning method gives for the *i*th observation in the training data

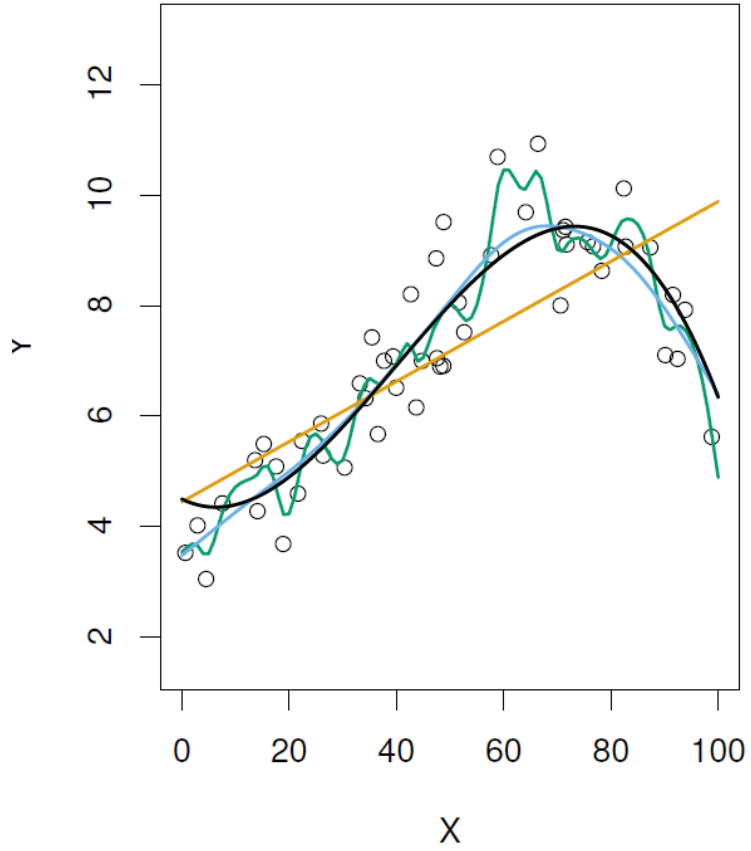
- More accurately, this is the training MSE.

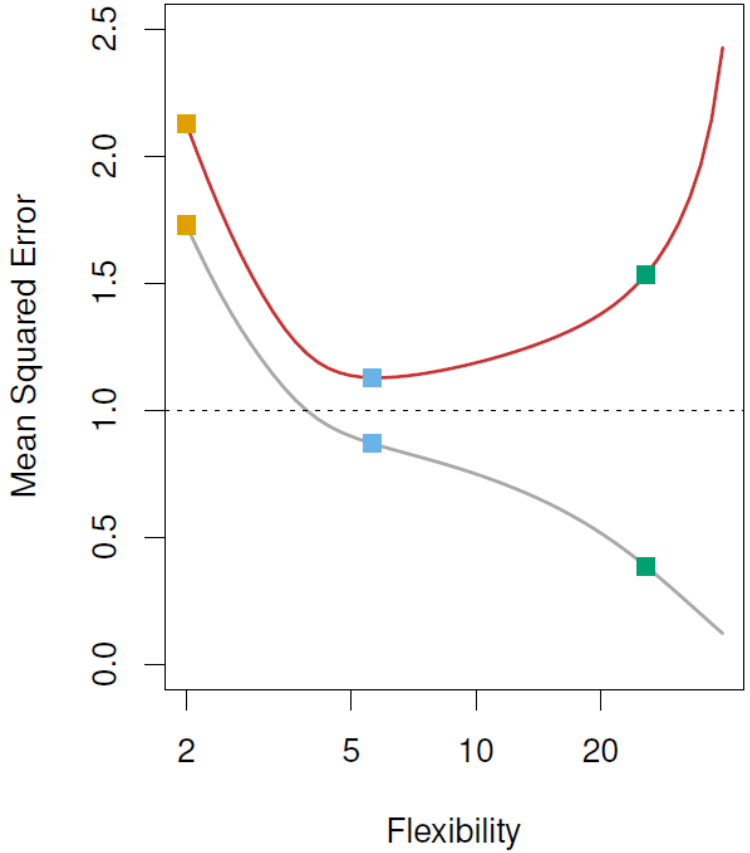
- MSE is small if the predicted responses are very close to the true responses

Training MSE Vs Test MSE

- Statistical learning methods are trying to minimize MSE on the training data.
  - For example, OLS (ordinary least squares) minimizes the MSE. But this does not guarantee OLS to be the best method for prediction.
- What we really care is how well the method performs on previously unseen test data.
  - Stock price prediction: We don't really care how well our method predicts last week's stock price. Instead, we care about how it will predict tomorrow's price.
- Smallest training MSE does not guarantee smallest test MSE.
- Thus, we should use test MSE to select models. The best model is the one that has the smallest test MSE.

Example of Training and Test MSEs





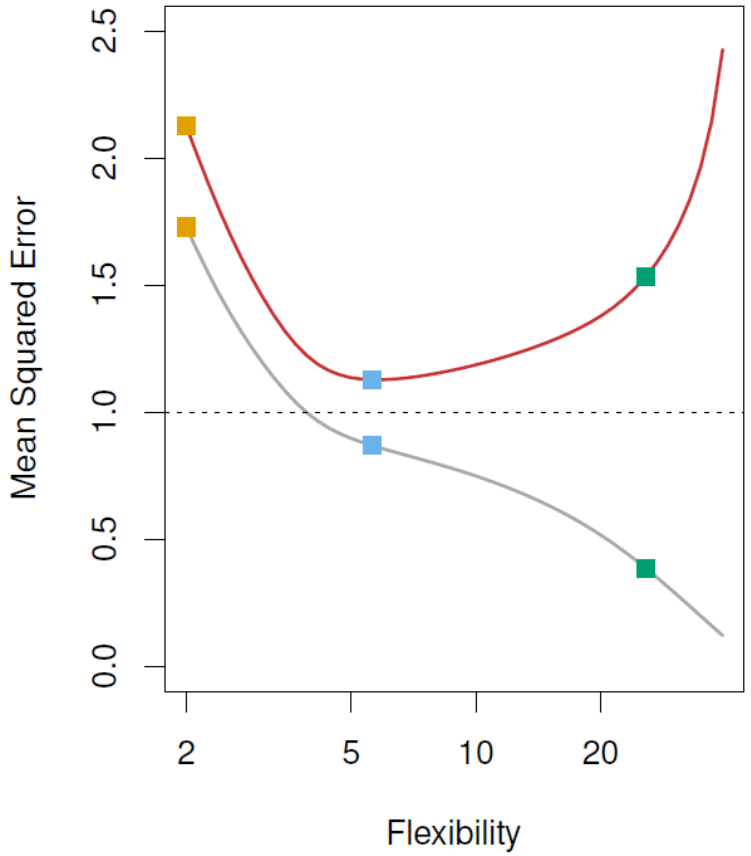
Left:

- Black: truth f(x)
- Orange: linear regression
- Blue: smoothing spline (less flexible)
- Green: smoothing spline (more flexible)

Right:

- Red: Test MSE
- Gray: Training MSE
- Dashed line: minimum possible test MSE

Example of Training and Test MSEs



- Red: Test MSE
- Gray: Training MSE
- Dashed line: minimum possible test MSE

- As flexibility (degrees of freedom) increases:
  - Training MSE declines monotonically;
  - Test MSE follows a U-shape.
- Overfitting: When a method yields a small training MSE but a large test MSE.
- Underfitting: When a method yields both a large training MSE and a large test MSE.
- Thus, our objective is to find a method with proper flexibility that fits the data just right.

The Classification Setting

- For classification problems, we can use *error rate* to assess the model accuracy

$$Error\ Rate = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

where $I(y_i \neq \hat{y}_i)$ is an indicator function

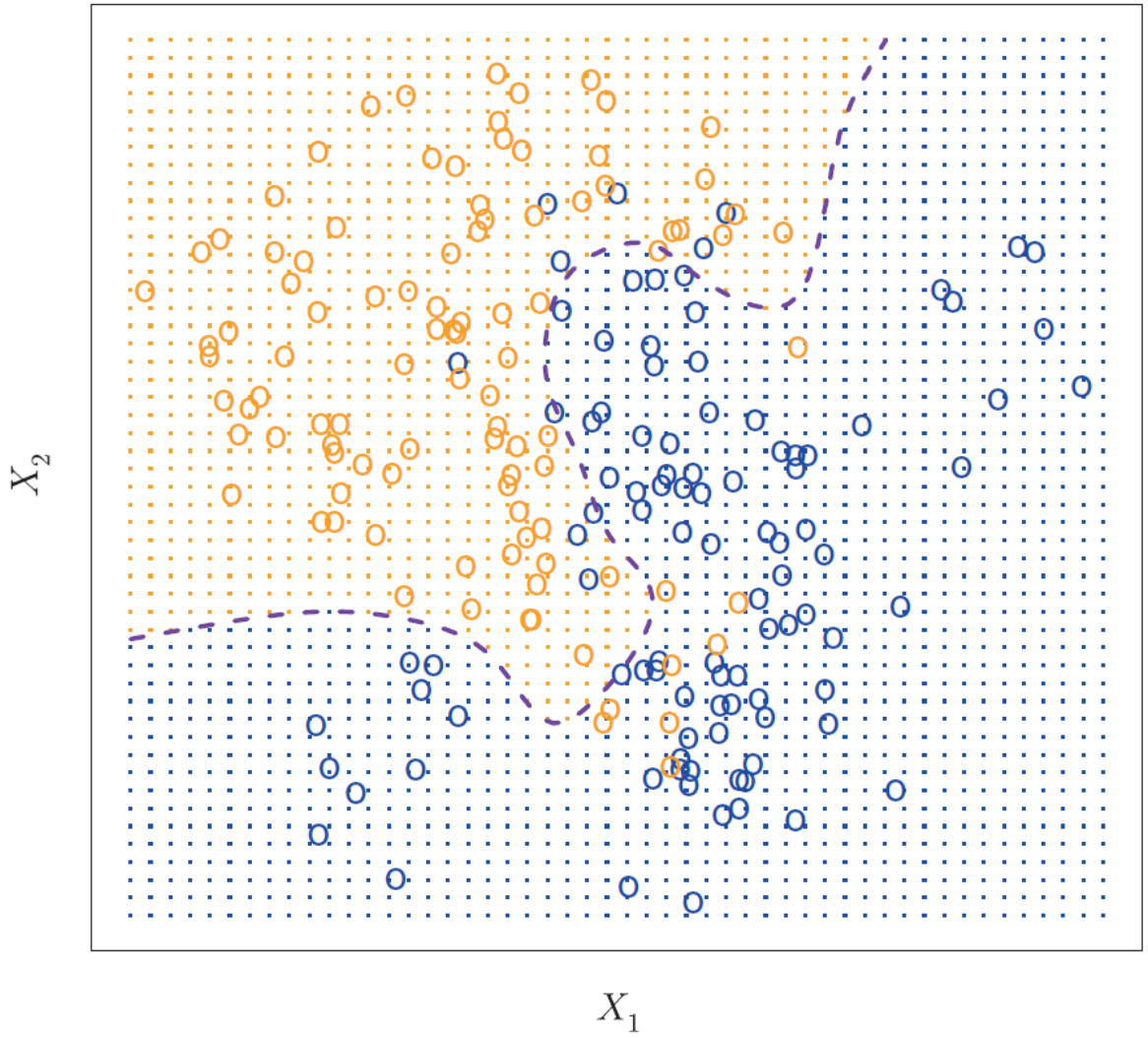
$$I(y_i \neq \hat{y}_i) = \begin{cases} 1, if\ the\ condition(y_i \neq \hat{y}_i)\ is\ true \\ 0, if\ the\ condition(y_i \neq \hat{y}_i)\ is\ false \end{cases}$$

The Bayes Classifier

- In order to minimize test error rate, on average, a classifier can assign each observation to the most likely class given its predictor values.

- Bayes classifier works in a simple way:
  - First, calculates conditional probability $\Pr(Y = j|X = x_0)$;
  - Then, assign the class j for which the conditional probability is largest.

Bayes Optimal Classifier – Unattainable Gold Standard

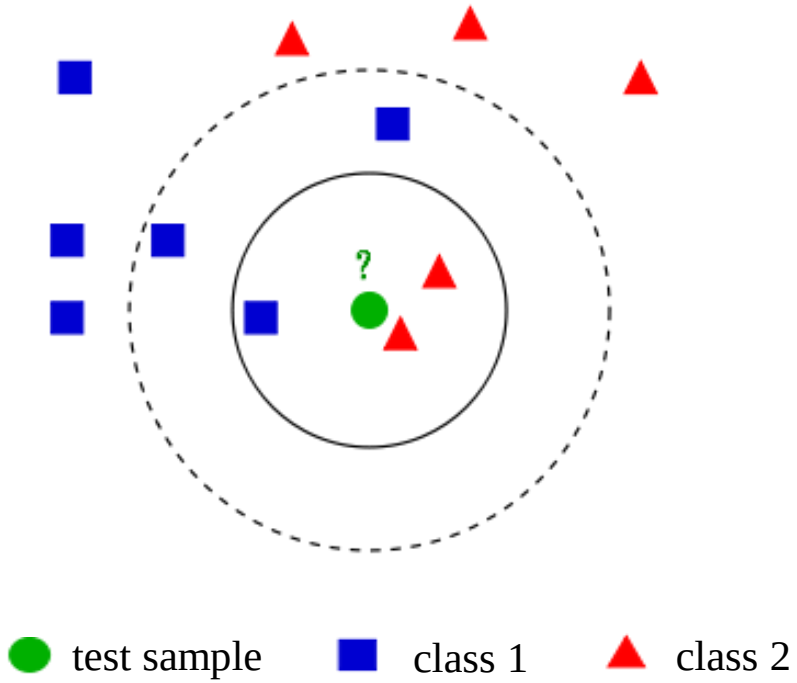


- Two predictors X_1 and X_2
- Two classes: Orange and Blue
- Dashed line: Bayes decision boundary

For real data, conditional probability is unknown, so that it's impossible to implement Bayes classifier.

K-Nearest Neighbors (KNN)

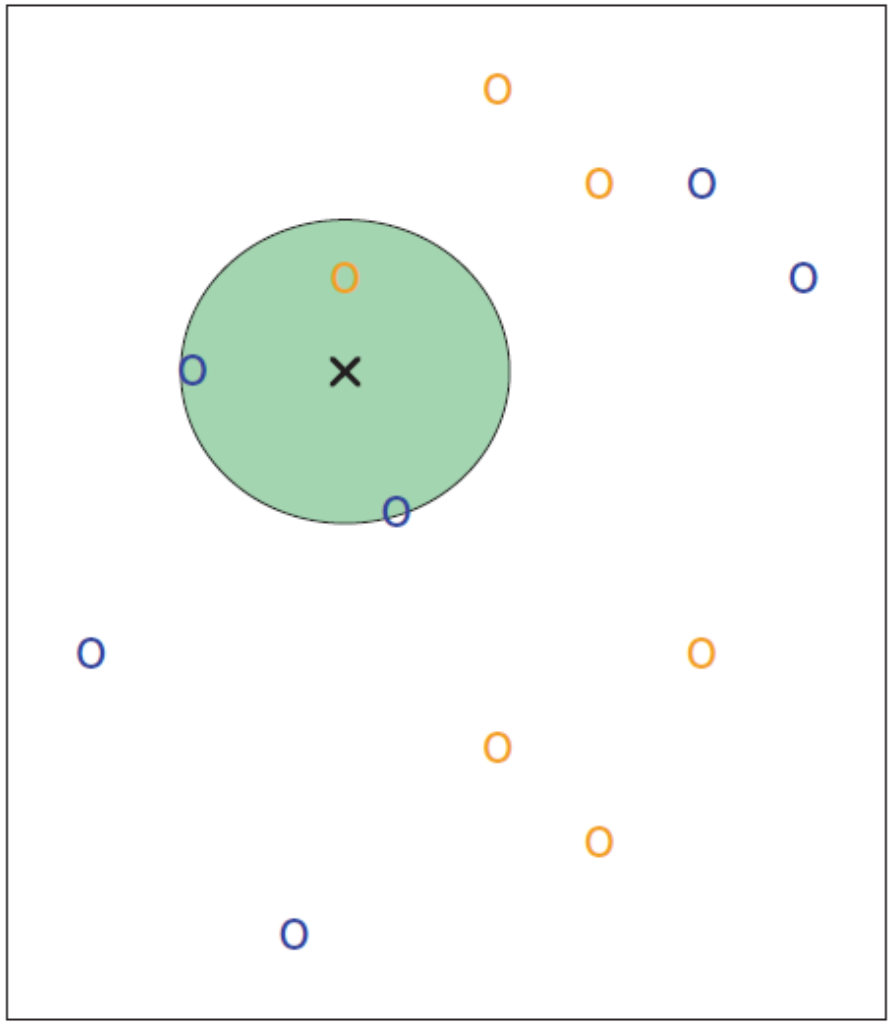
- Many approaches including KNN tries to estimate the conditional distribution of Y given X.
- KNN contains a parameter k
  - Large values of k reduce the effect of noise, but make boundaries between classes less distinct.

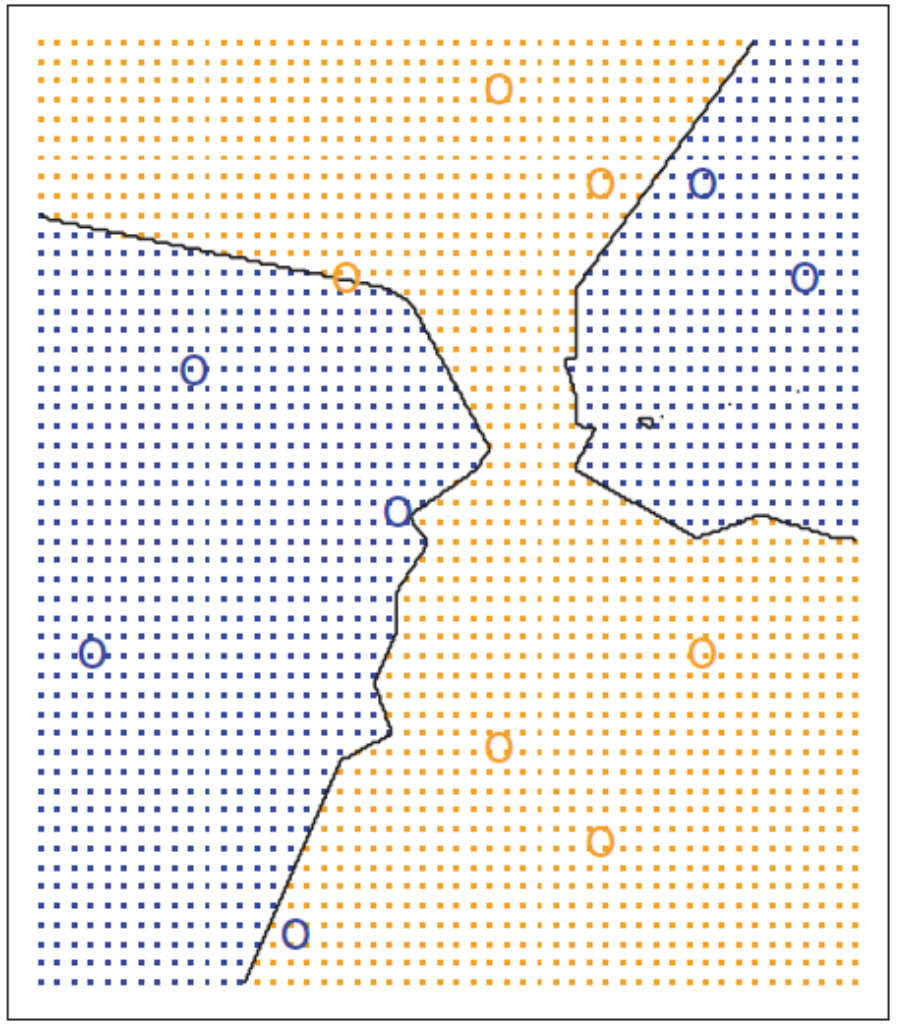


If k = 3, the test sample is classified as class 2;

If k = 5, class 1 is assigned.

KNN Example: k=3



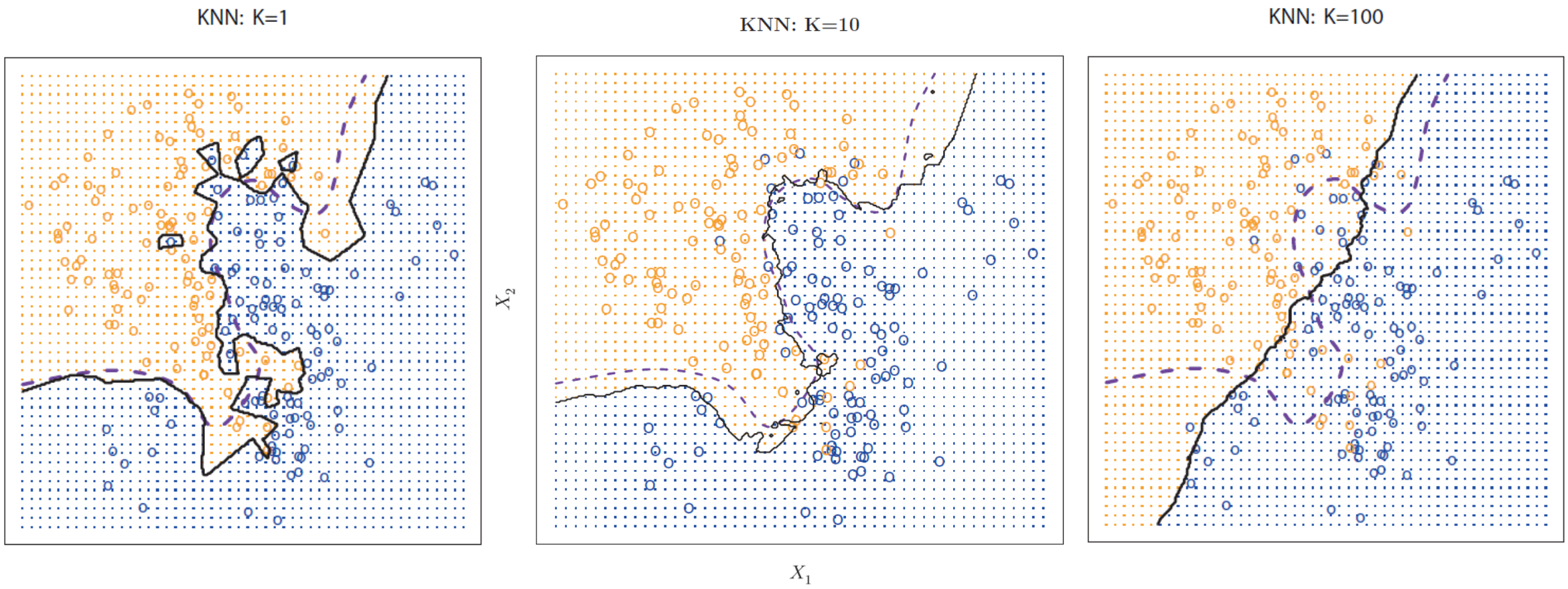


KNN Simulated Data

Overly flexible

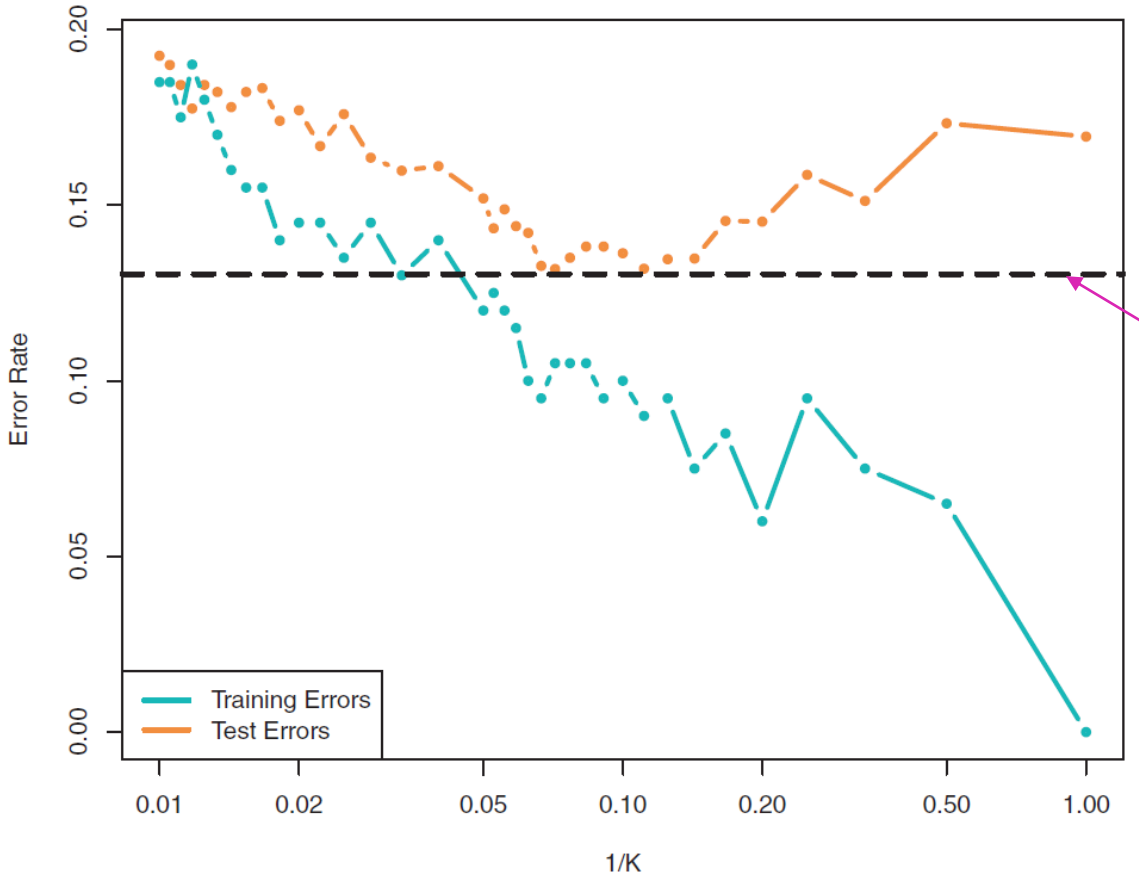
Looks right flexible

Insufficiently flexible



- Purple dashed line: Bayes decision boundary
- Black curve: KNN decision boundary

KNN Training and Test Error Rates



Choosing the correct level of flexibility is critical to the success of any statistical learning method.

The bias-variance tradeoff, and the resulting U-shape in the test error, can make this a difficult task.

Select the "Optimal" Model: Bias-Variance Tradeoff

- Bias is an error from improper assumptions in the learning algorithm.

$$Bias = E\left[\hat{f}(x)\right] - f(x)$$

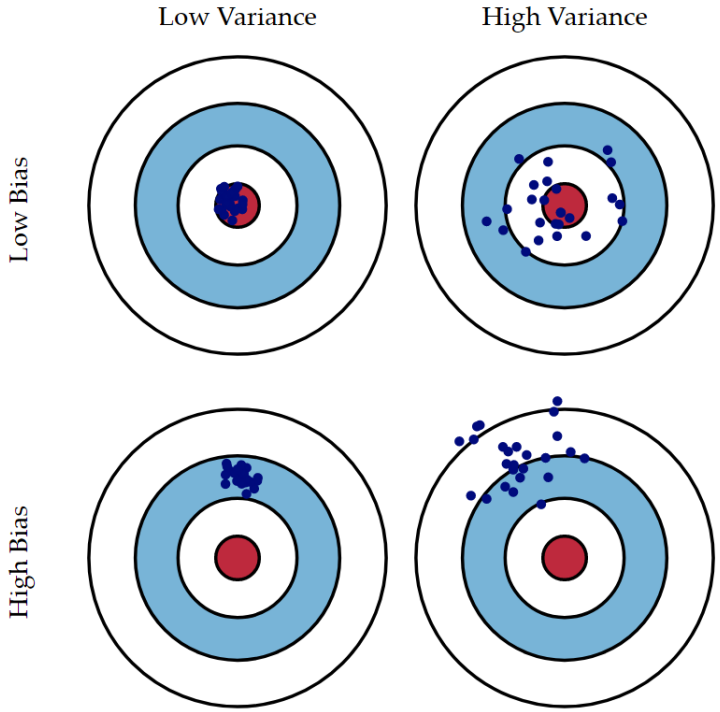
- Variance is an error from sensitivity to small fluctuations in the training set.

$$Variance = E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right]$$

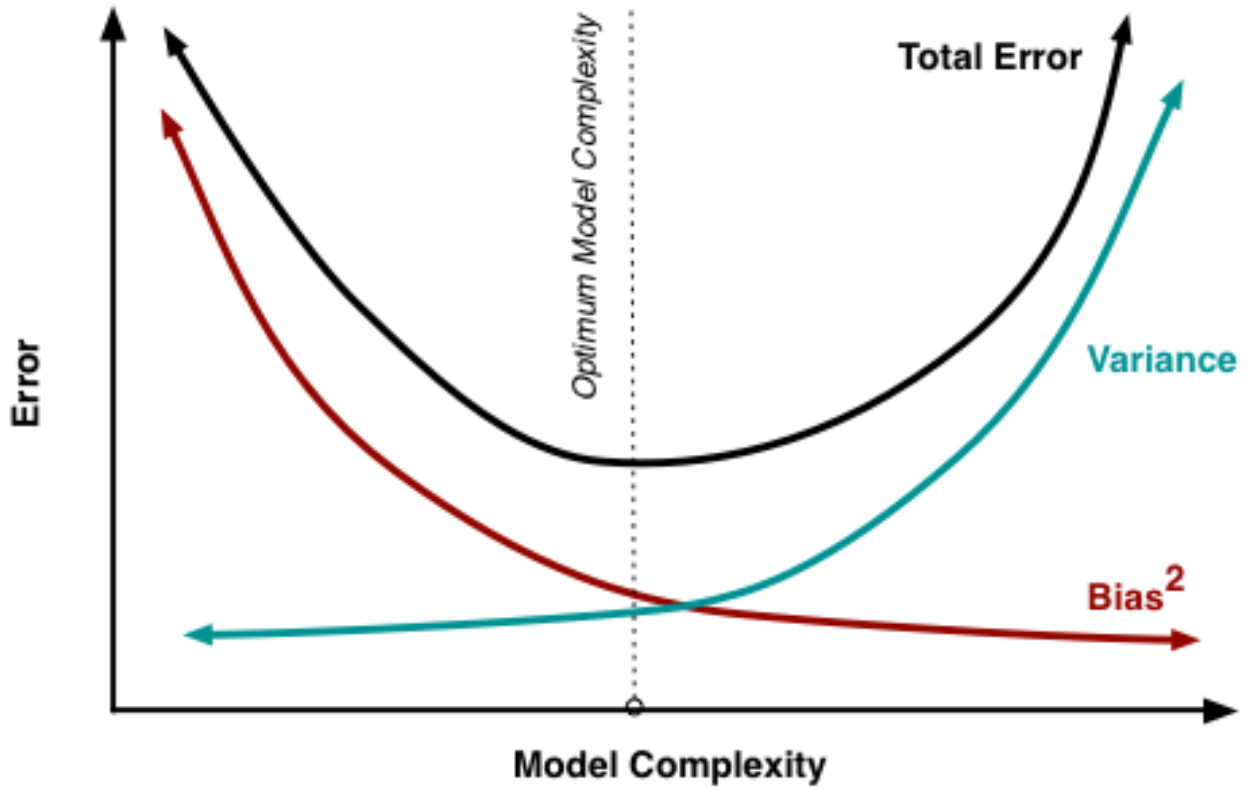
- Squared estimation error can be decomposed as:

$$\begin{aligned}
Error(x) &= E\left[\left(Y - \hat{f}(x)\right)^2\right] \\
&= E\left[\left(f(x) + \varepsilon - \hat{f}(x)\right)^2\right] = E\left[\left(f(x) - \hat{f}(x)\right)^2\right] + E(\varepsilon^2) \\
&= E\left[f(x)^2 - 2f(x)\hat{f}(x) + \hat{f}(x)^2\right] + \sigma_\varepsilon^2 \\
&= f(x)^2 - 2f(x)E\left[\hat{f}(x)\right] + E\left[\hat{f}(x)^2\right] + \sigma_\varepsilon^2 \\
&= \left(f(x) - E\left[\hat{f}(x)\right]\right)^2 + \left(E\left[\hat{f}(x)^2\right] - E\left[\hat{f}(x)\right]^2\right) + \sigma_\varepsilon^2 \\
&= \left(f(x) - E\left[\hat{f}(x)\right]\right)^2 + E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right] + \sigma_\varepsilon^2
\end{aligned}$$

$$Error(x) = \underbrace{Bias^2 + Variance}_{Reducible\ Error} + Irreducible\ Error$$



Select the "Optimal" Model: Bias-Variance Tradeoff



Under-fitting: high bias, low variance

Over-fitting: low bias, high variance

Given imperfect models and finite data, there is a tradeoff between minimizing bias and minimizing variance.

More Flexible Model vs. Less Flexible Model

- A more flexible model can better fit non-linear relationship, thus decreasing bias;
- But a more flexible model may also fit the noise (rather than signal) too closely, thus increasing variance;
- Also the results of a more flexible model are more difficulty to explain.

- A more flexible model tends to be better when:
  - n is very large, p is small;
  - Non-linear relationship between predictors and response;
  - Emphasis on prediction rather than interpretation.

RECAP: OUTLINE

- (I) Overview of machine learning (ML)
  1. What is learning?
  2. Practical definition of ML
  3. ML model estimation methods: parametric, nonparametric
  4. Types of ML

- (II) Scale of measurement

  Nominal, ordinal, interval, ratio

- (III) Model accuracy
  1. Regression setting: MSE, training MSE, test MSE
  2. Classification setting: Error rate
  3. Bias variance tradeoff

Q & A

Assignments

- Homework 1 (Due Jan 24)
- Reading (Due Jan 25)
  - Book Chapter 2 Section 2.3 and Try the Code
  - "An Introduction to R" Chapters 1, 2, 3, 4, 5, 6, 9,10; pg 2-29, 40-50
- Install R and RStudio to your PC (Due Jan 25)

IST 5535: Machine Learning Algorithms and Applications

Langtao Chen, Spring 2021

2. Getting Started with R

Reading

- Book Section 2.3 “Lab: Introduction to R”
- An Introduction to R (Chapters 1, 2, 3, 4, 5, 6, 9,10; pg 2-29, 40-50)
  - https://cran.r-project.org/doc/manuals/R-intro.pdf
- Data Wrangling with dplyr and tidyr Cheat Sheet
  - https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf
- RStudio IDE Cheat Sheet
  - https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf
- Base R Cheat Sheet
  - http://github.com/rstudio/cheatsheets/raw/master/base-r.pdf

Learning Objectives

- Learn basic R programming knowledge
- Get familiar with RStudio, be able to use it for BA and ML projects
- Be able to apply basic data structures in R
- Understand the concepts of control structures and be able to use them in R programming
- Be able to define functions for code reuse

Agenda

- What is R?
- Program with RStudio
- Introduction to R Markdown
- Data Structures in R
- R Functions
- Control Structures (Selection and Loop)

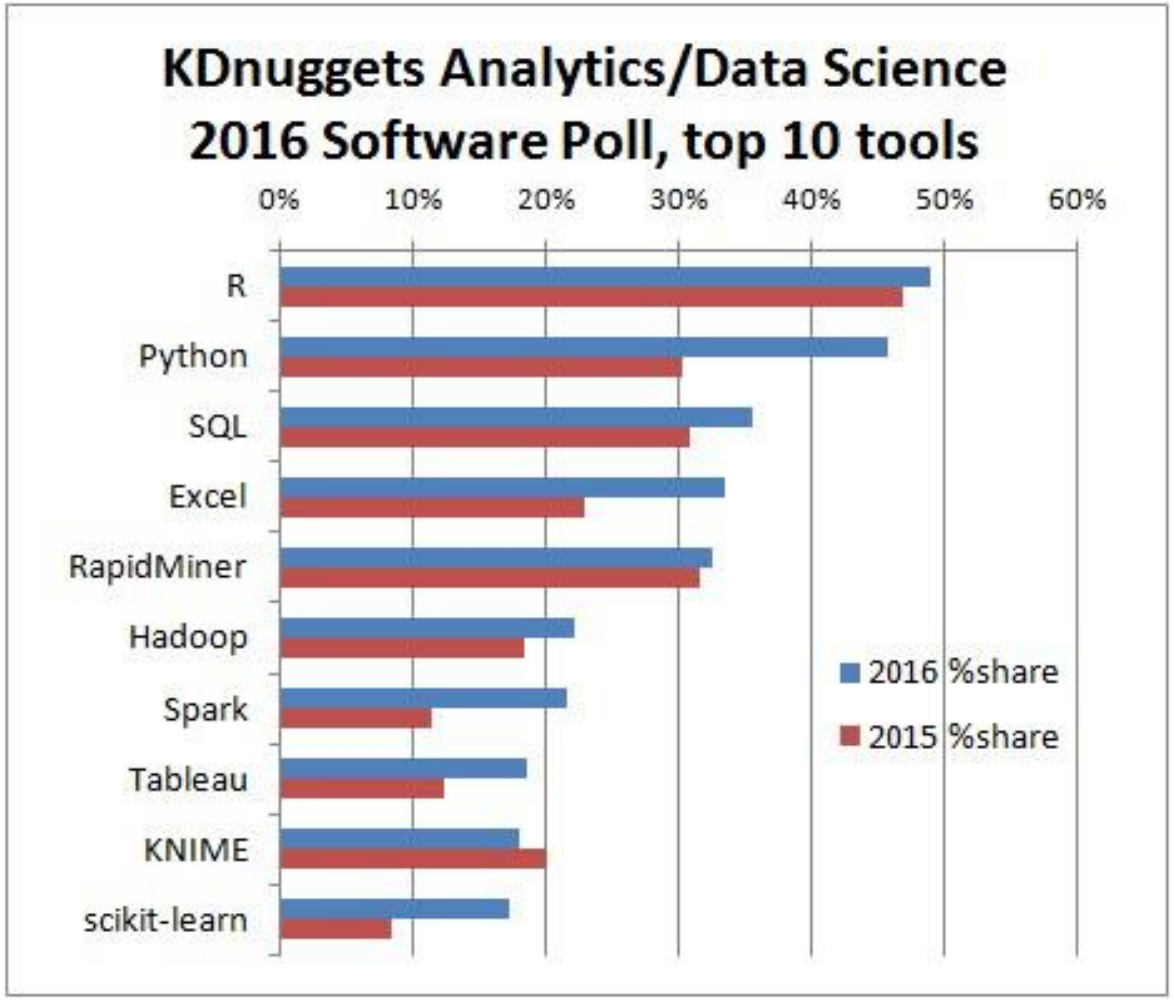
R

- A free, open-source programming language for statistical computing
- An interpreted language (executed directly, no compilation)
- R supports matrix arithmetic (like Matlab)
- R supports both procedural programming and object-oriented programming

CRAN: Comprehensive R Archive Network

- Capability extended through a packaging system on CRAN, the Comprehensive R Archive Network
  - http://cran.r-project.org/
- So many useful packages available on CRAN
- You can contribute to CRAN by uploading your own package!

R is popular; Don’t get left behind.



http://www.kdnuggets.com/2016/06/r-python-top-analytics-data-mining-data-science-software.html

Steep Learning Curve for R

- The “weird” syntax of R

“The best thing about R is that it was developed by statisticians. The worst thing about R is that ... it was developed by statisticians.”

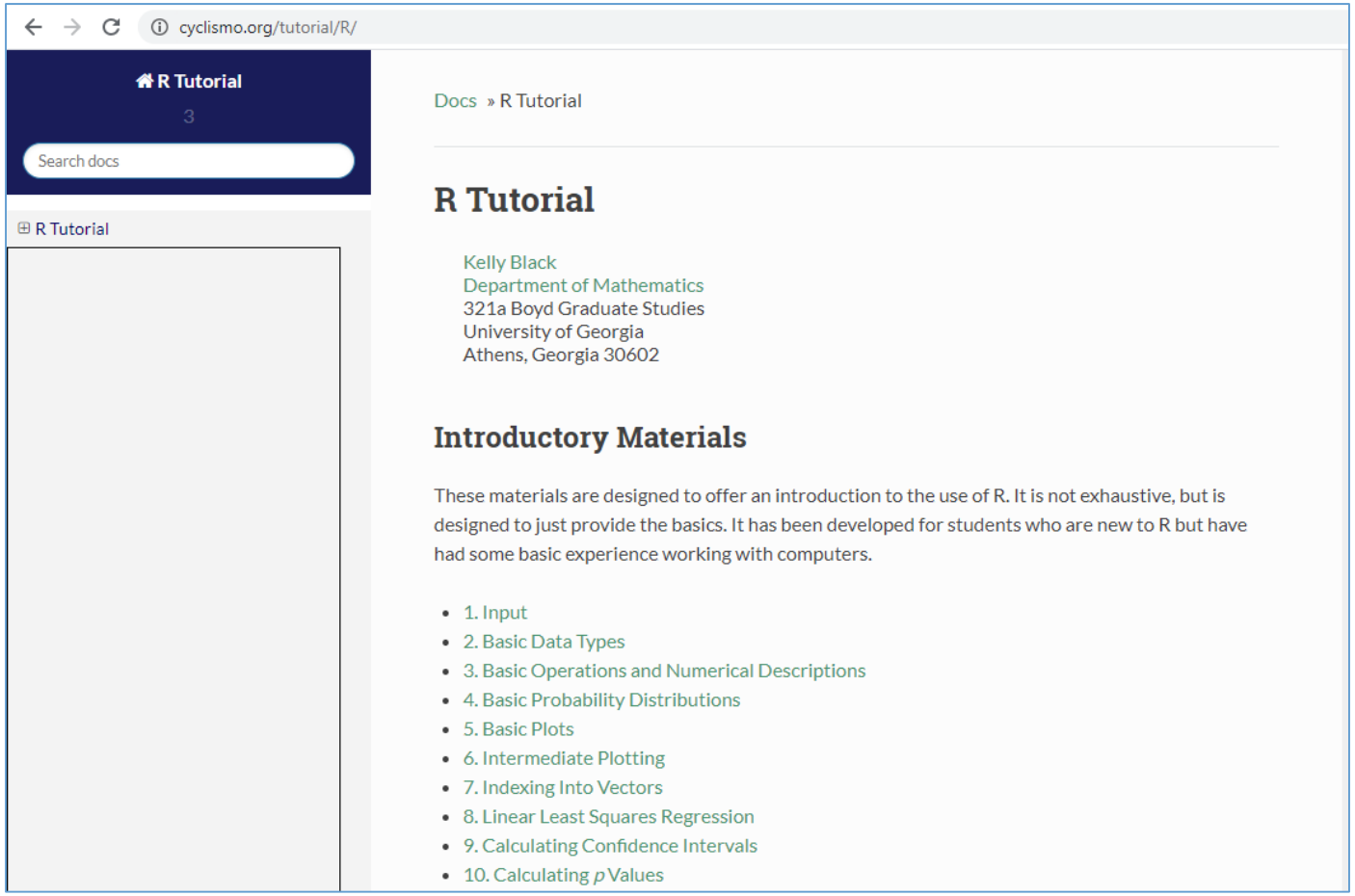
-- Bo Cowgill, Google

“Unlike other high-level scripting languages, such as Python or Ruby, R has a unique and somewhat prickly syntax and tends to have a steeper learning curve than other languages.”

-- Drew Conway & John White, “Machine Learning for Hackers” P2.

To be familiar with R

https://www.cyclismo.org/tutorial/R/



Other Resources for Learning R

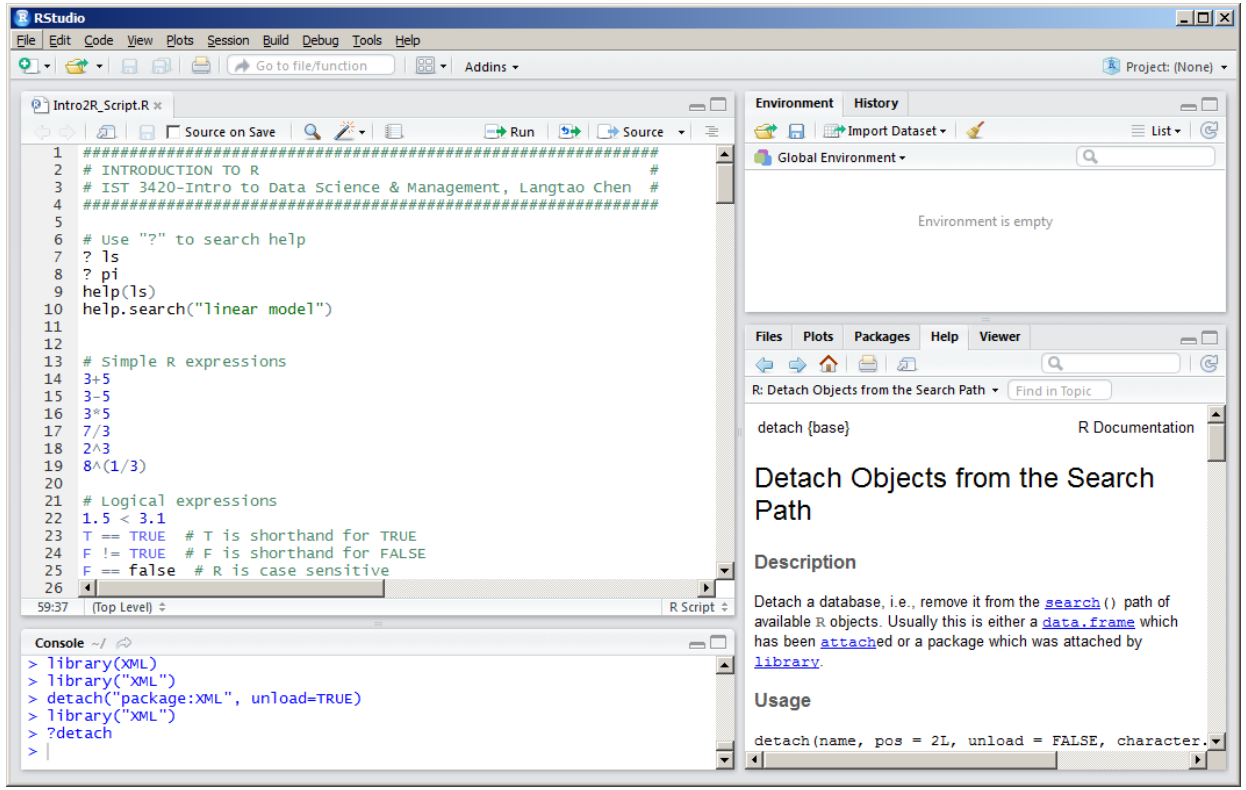
- Remember the Rseek (search engine for R language)!
  - http://rseek.org/
- “An Introduction to R”
  - https://cran.r-project.org/doc/manuals/R-intro.pdf
- R Language Definition
  - https://cran.r-project.org/doc/manuals/r-release/R-lang.pdf
- “R Reference Card” – quick reference for important tasks
  - https://cran.r-project.org/doc/contrib/Short-refcard.pdf
- A Step-by-Step R Tutorial
  - http://www.cyclismo.org/tutorial/R/
- Stack Overflow Q&A Site
  - http://stackoverflow.com/questions/tagged/r
- Commonly Used R Packages
  - https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages

Agenda

- What is R?
- Program with RStudio
- Introduction to R Markdown
- Data Structures in R
- R Functions
- Control Structures (Selection and Loop)

RStudio

- An open-source IDE for R
- Since version 1.2, RStudio started to support Python
- Install the RStudio Desktop (open source edition) from https://www.rstudio.com/products/rstudio/download/



Try RStudio

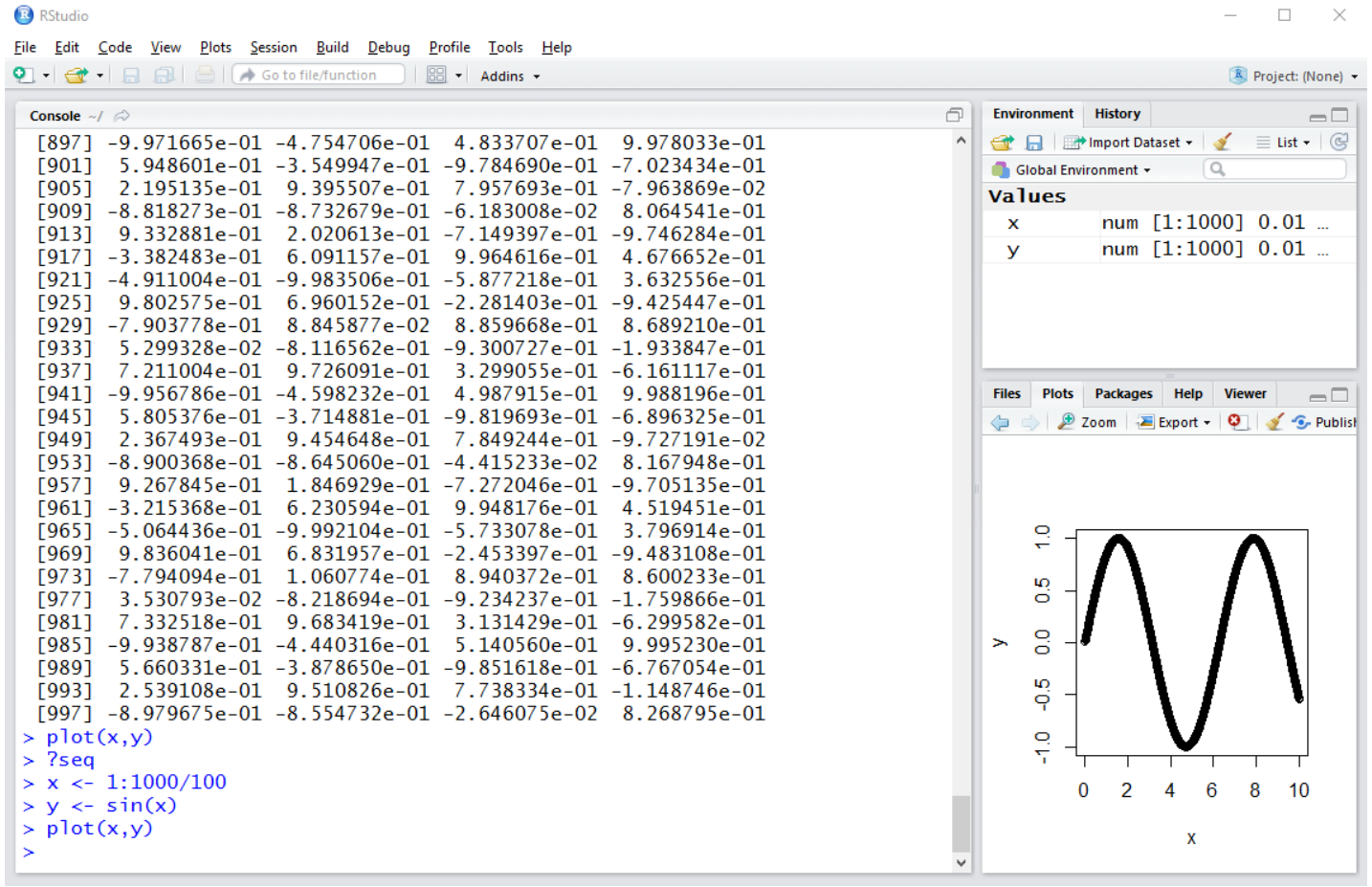
Use Shortcuts to Improve Coding Efficiency

- For a complete list, refer to https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts

Function	***Windows & Linux***	***Mac***
Move cursor to Source Editor	Ctrl + 1	Ctrl + 1
Move cursor to Console	Ctrl + 2	Ctrl + 2
Interrupt currently executing command	Esc	Esc
Navigate command history	Up/Down	Up/Down
Run current line/selection	Ctrl + Enter	Command + Enter
Save active document	Ctrl + S	Command + S

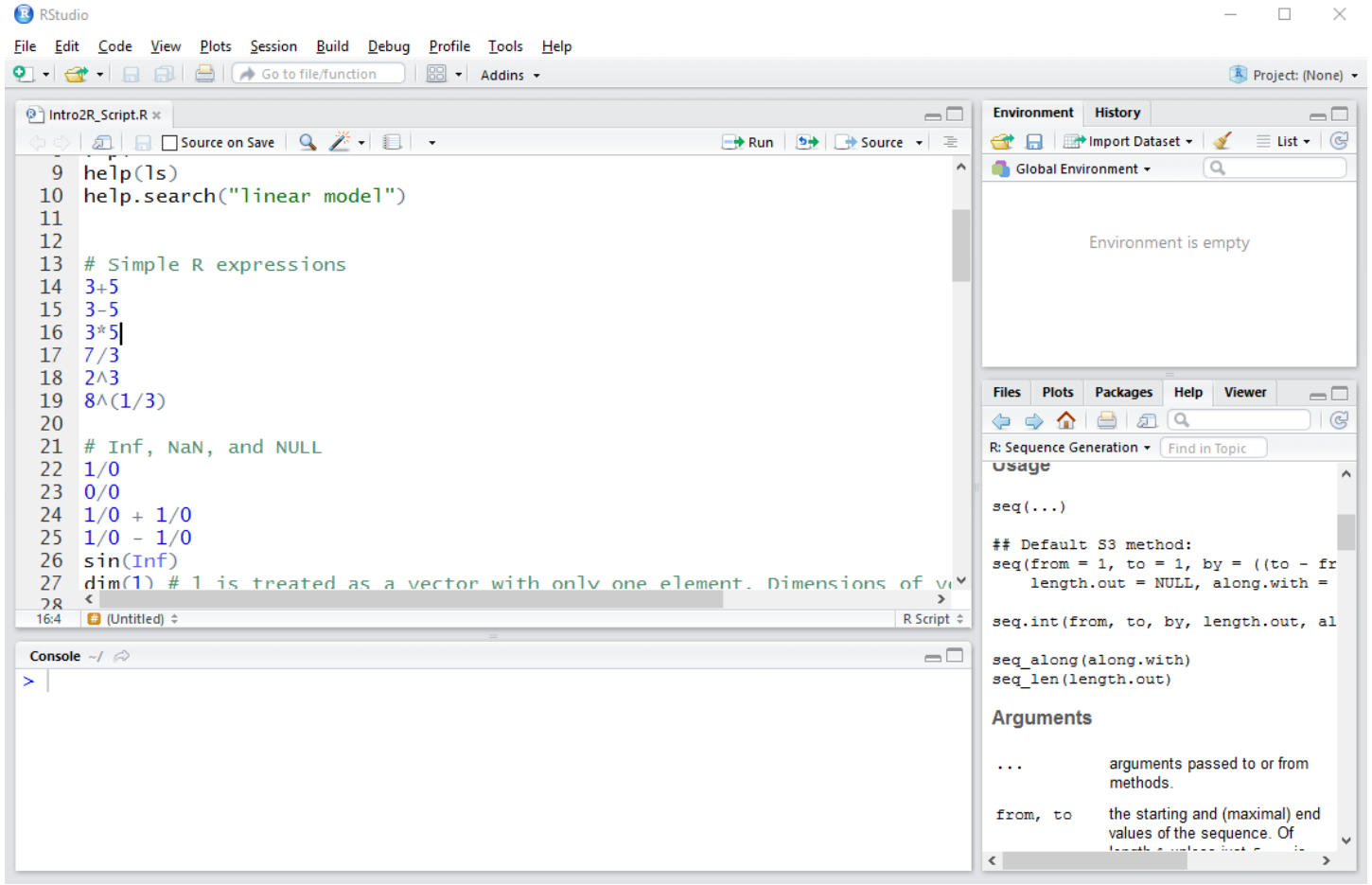
Three Ways of R Programming for Data Science

- 1: Type and execute R command in Console window line by line => Avoid



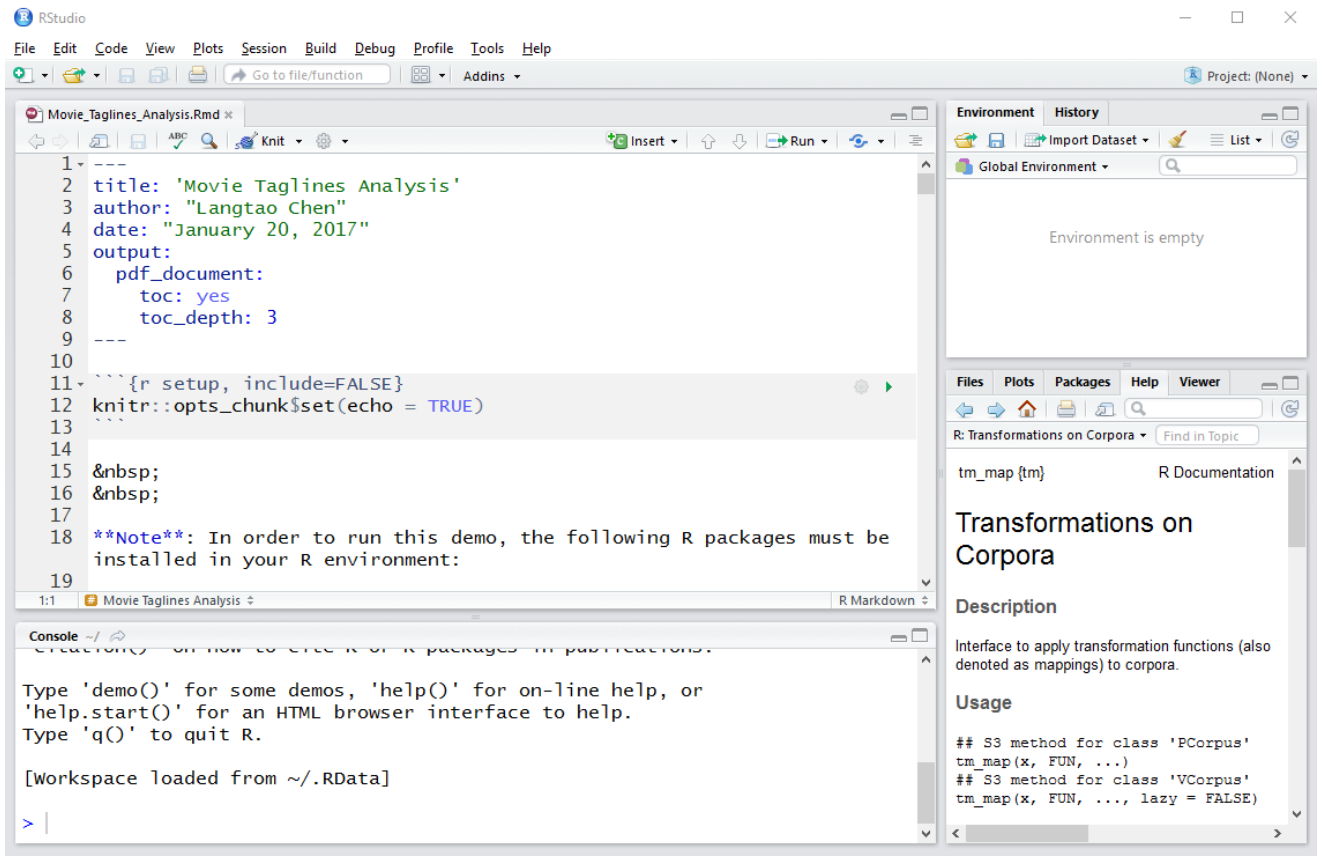
Three Ways of R Programming for Data Science

- 2: Program in R script => Acceptable



Three Ways of R Programming for Data Science

- 3: Program in R Markdown + Results + Full explanation => Preferred



Fundamental Principles

- Everything that exists in R is an object.
- Everything that happens in R is a function call.
- Interfaces to other software are part of R.

Source: Chambers, John M. *Extending R*. CRC Press, 2016.

Attributes of an Object

- names
- dimnames
- dim
- class
- attributes (contain metadata)
- length (works on vectors and lists)
- nchar (number of characters in a string)

Basic Operations

- R is case sensitive!
- Use “?” to search help
- Constants and symbols
  - Any number typed directly is a constant.
  - The name of a variable is a symbol.
- Two assignment operators
  - Left assignment <- (for example, a <- 4)
  - Right assignment -> (for example, 4 -> b)
- List indexing: $

Atomic Data Types

- Character
  - “a”, “hello”
- Logical
  - TRUE, FALSE
- Integer
  - x <- 5L # Must add L at the end to explicitly denote integer
- Double
  - 4, 13.48
- Complex
  - 2 + 3i

R Basic Operators

- Arithmetic Operators

Operator	Meaning	Unary or Binary
+	Plus	Both
-	Minus	Both
*	Multiplication	Binary
/	Division	Binary
^	Exponentiation	Binary
%%	Modulus	Binary
%/%	Integer division	Binary
%*%	Matrix product	Binary
%o%	Outer product	Binary

(cont.)

- Comparison Operators

Operator	Meaning	Unary or Binary	Example (a is 4)	Result
<	Less than	Binary	a < 0	FALSE
>	Greater than	Binary	a > 0	TRUE
==	Equal to	Binary	a == 3	FALSE
>=	Greater than or equal to	Binary	a >= 0	TRUE
<=	Less than or equal to	Binary	a <= 0	FALSE
!=	Not equal to	Binary	a !=3	TRUE

(cont.)

- Logic Operators

Operator	Meaning	Unary or Binary	Example (a is TRUE, b is FALSE)	Result
&	And, vectorized	Binary	a & b	FALSE
|	Or, vectorized	Binary	a | b	TRUE
&&	And, not vectorized	Binary	a && b	FALSE
||	Or, not vectorized	Binary	a || b	TRUE
!	Not	Unary	!a	TRUE
xor	Exclusive or	Binary	xor(a,b)	TRUE
isTrue()	Test if true	Unary	isTRUE(a)	FALSE

Agenda

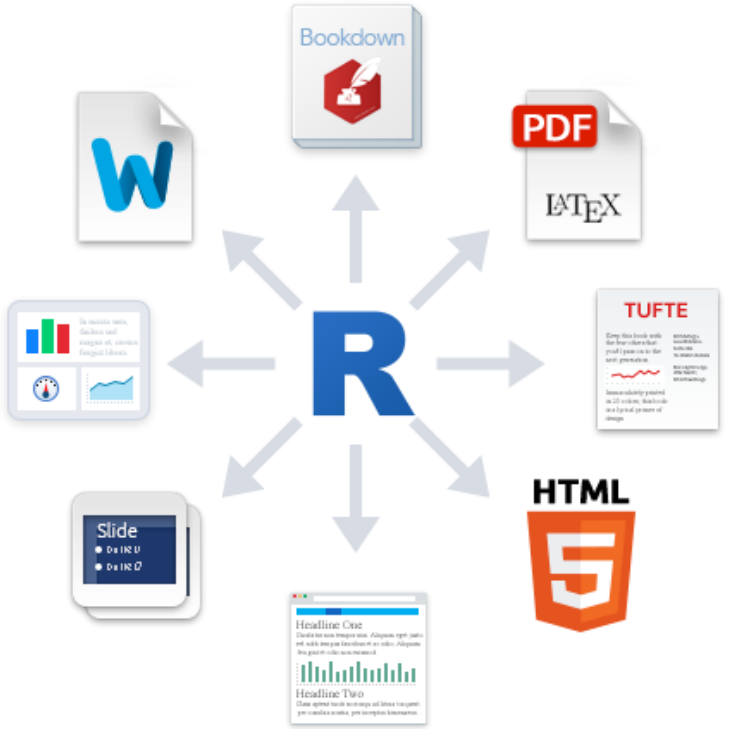
- What is R?
- Program with RStudio
- Introduction to R Markdown
- Data Structures in R
- R Functions
- Control Structures (Selection and Loop)

Dynamic Documents in R

- "R Markdown is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R".

- R code embedded in text
  - You can write R code in plain text and generate data analysis reports in various formats such as HTML, PDF, Word, HTML5 slides.

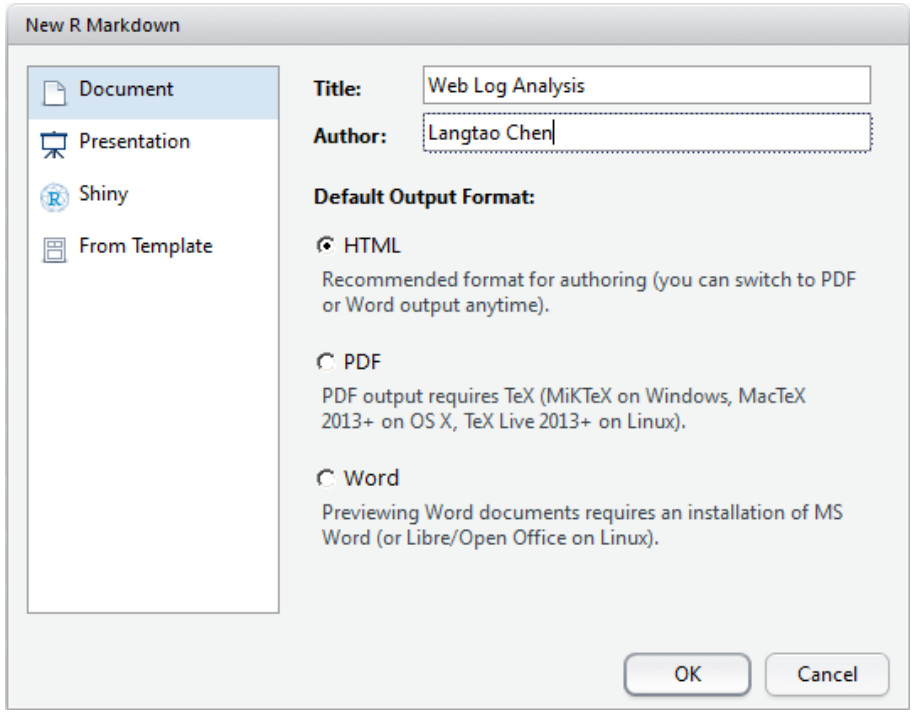
- Reproducible analysis
  - You can easily reproduce the data analysis results after the data and/or code change.



Source: http://rmarkdown.rstudio.com/

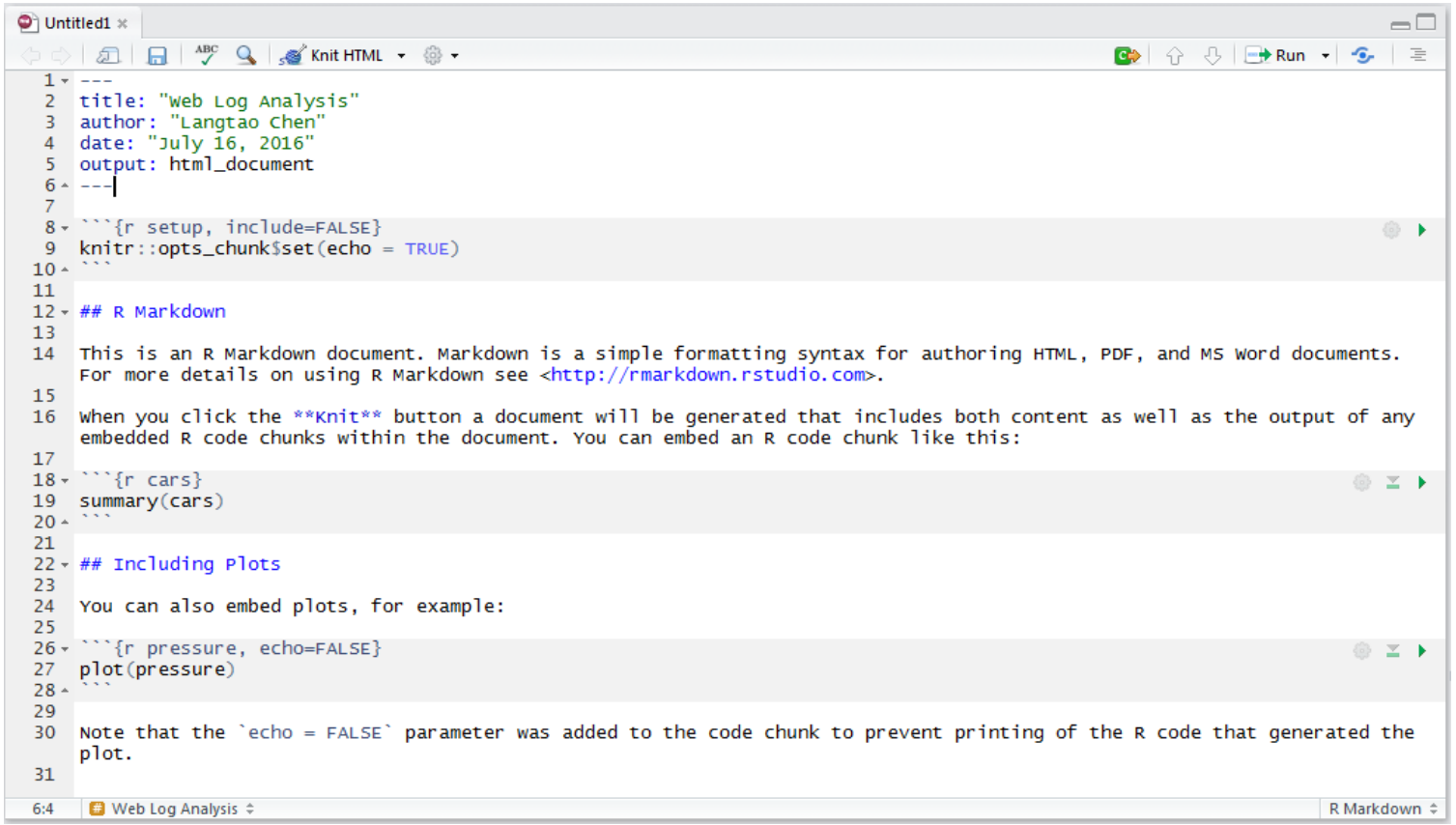
Use R Markdown

- Install R markdown package
  - install.packages("rmarkdown")
- In Rstudio, click “File -> New File -> R Markdown...” menu
- In the popup window, input header information, then click “OK” button



(cont.)

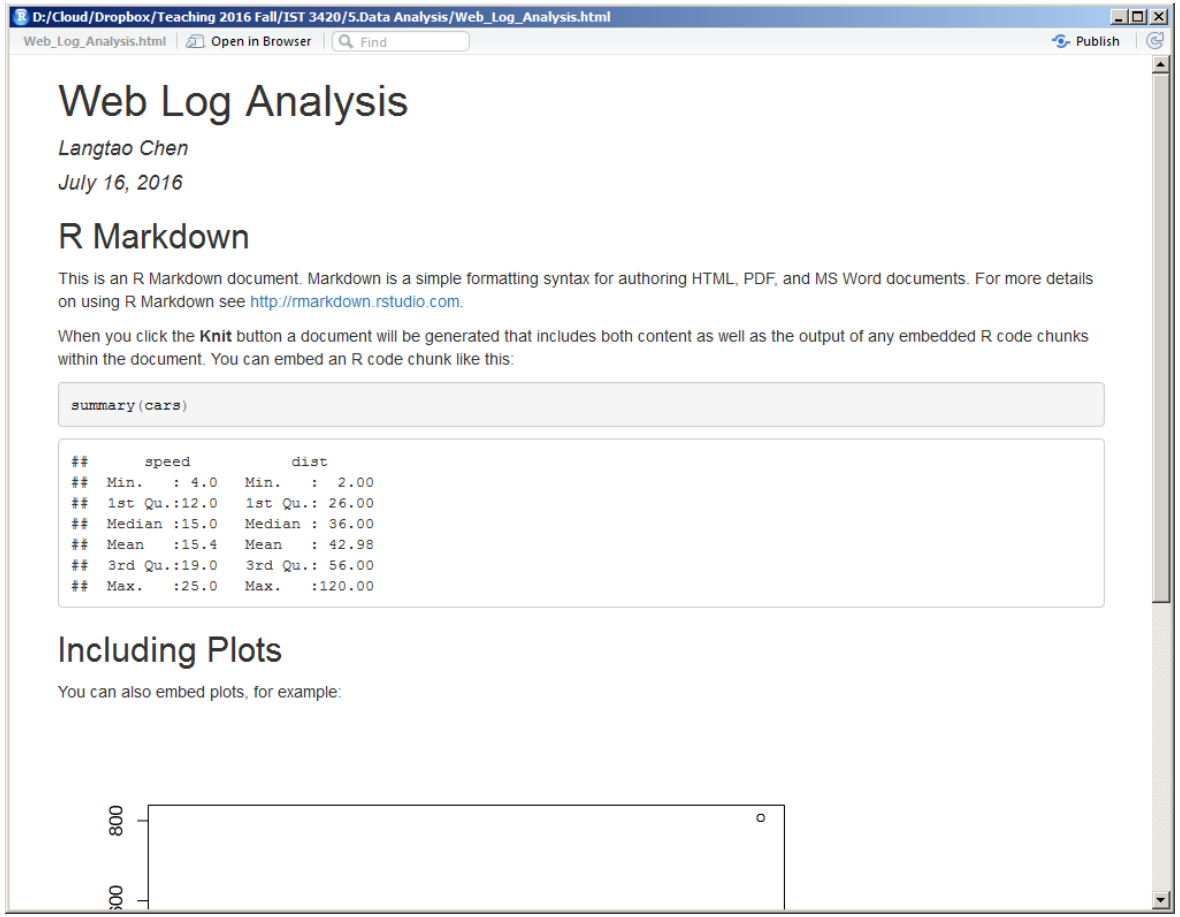
- RStudio generates a sample R markdown (.Rmd) file for you



(cont.)

- Click the “Knit HTML” button 

on the toolbar to generate the HTML report



(cont.)

- Follow the R markdown syntax demonstrated in the sample file, write your own data analysis by editing the Rmd template.

R Markdown Syntax Summary

- YAML Header (key: value pairs)
  - At the beginning of Rmd file
  - Between lines of ---
- Plain Text Format
  - Headers: Begin with #
  - Lists: Begin with -
  - LaTex or MathML equations: Enclosed within $
- Embedded R Code
  - R Code Chunks: Begin with ```{r} and end with ```
  - Inline R Code: Begin with `r and end with `

An Example

- R Markdown File

Manage_Weblog_Data.Rmd

- PDF Output

Manage_Weblog_Data.pdf

Reference

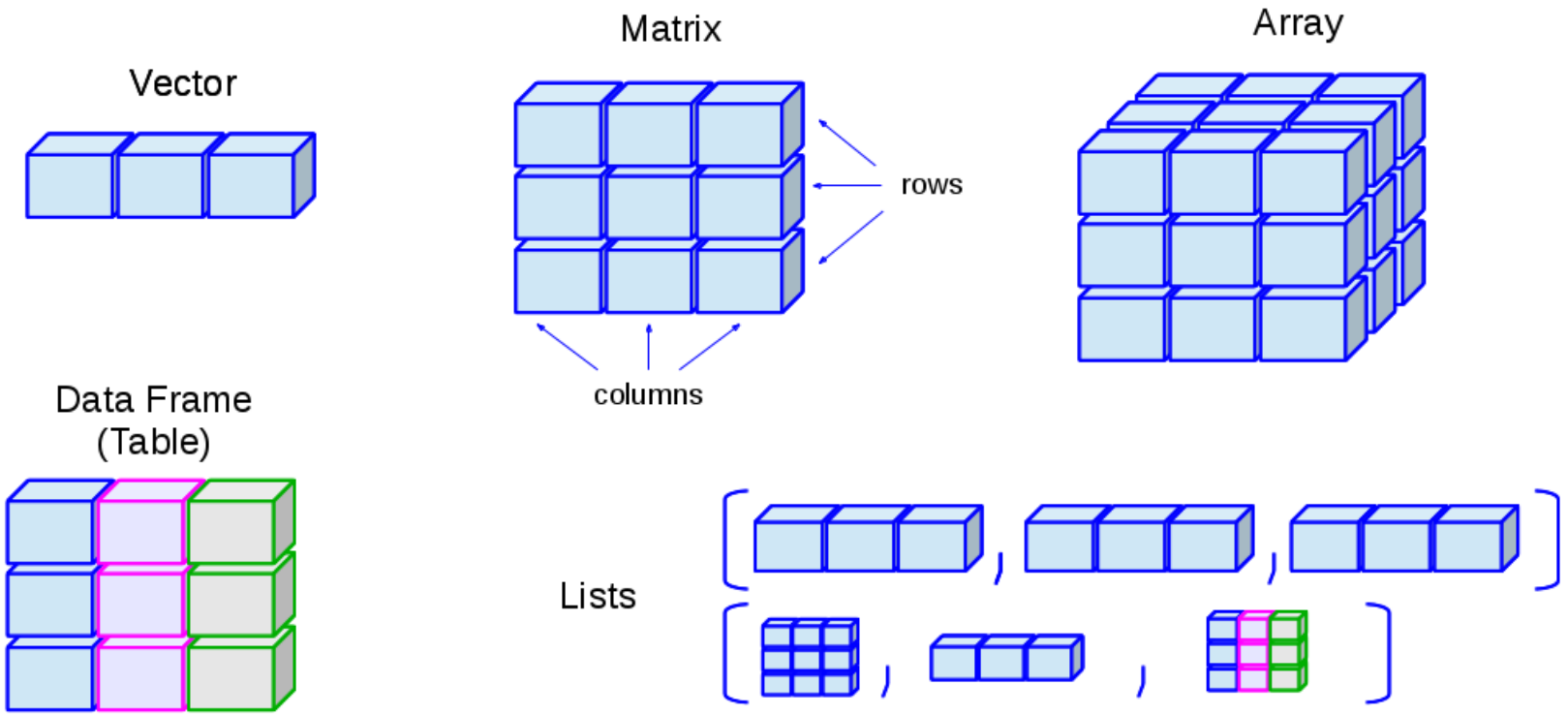
- R Markdown Cheat Sheet
  - http://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf
- R Markdown Reference Guide
  - http://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf

Agenda

- What is R?
- Program with RStudio
- Introduction to R Markdown
- Data Structures in R
- R Functions
- Control Structures (Selection and Loop)

R Data Structures

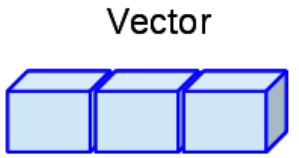
- Vectors, matrices, arrays, data frames (like tables in a RDBMS), and lists



There is no scalar data structure in R. We simply use a vector of length 1 to represent scalar.

Image source: http://venus.ifca.unican.es/Rintro/dataStruct.html

Vectors



- An ordered collection of elements
- Create a vector of numbers
  - v1 <- c(1,2,3,4)
- Use [] to access vector elements
- Create a vector of strings
  - v2 <- c("a","b","c")
- Elements in a vector should be of the same type
  - v3 <- c(1, "a")
  - mode(v3) # Check the type of storage mode
    [1] "character"

c function: c means “combine”

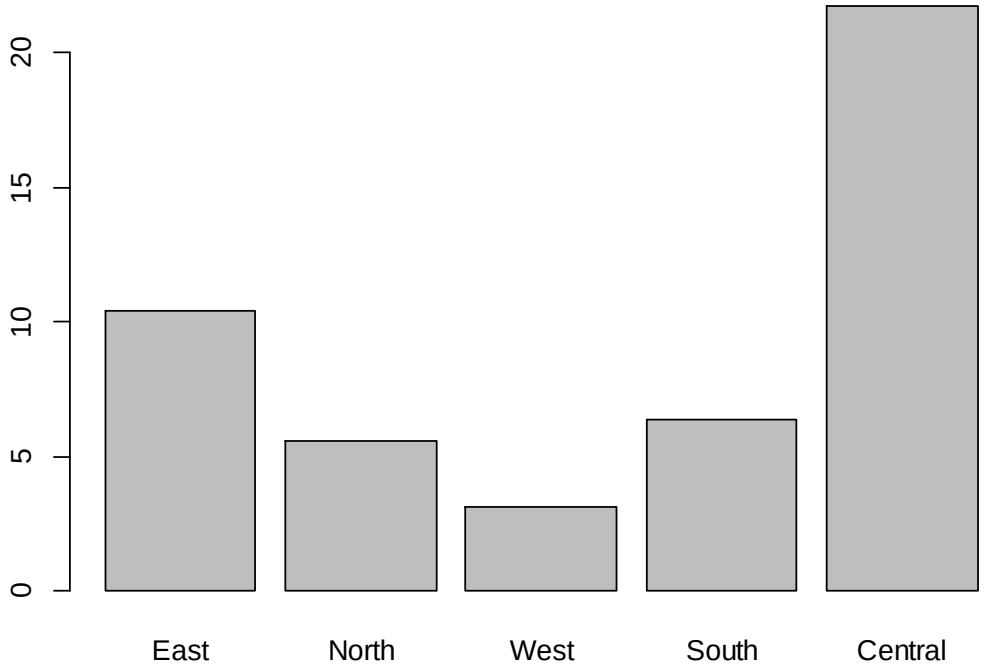
Names of Vectors

- Use names() to set or get names of an object

```
> v4
[1] 10.4 5.6 3.1 6.4 21.7
> names(v4) <- c("East","North","West","South","Central")
# To set vector name
> v4
  East North West South Central
  10.4 5.6 3.1 6.4 21.7
> names(v4) # To get vector name
[1] "East" "North" "West" "South" "Central"
```

Bar Plot

- barplot(v4)



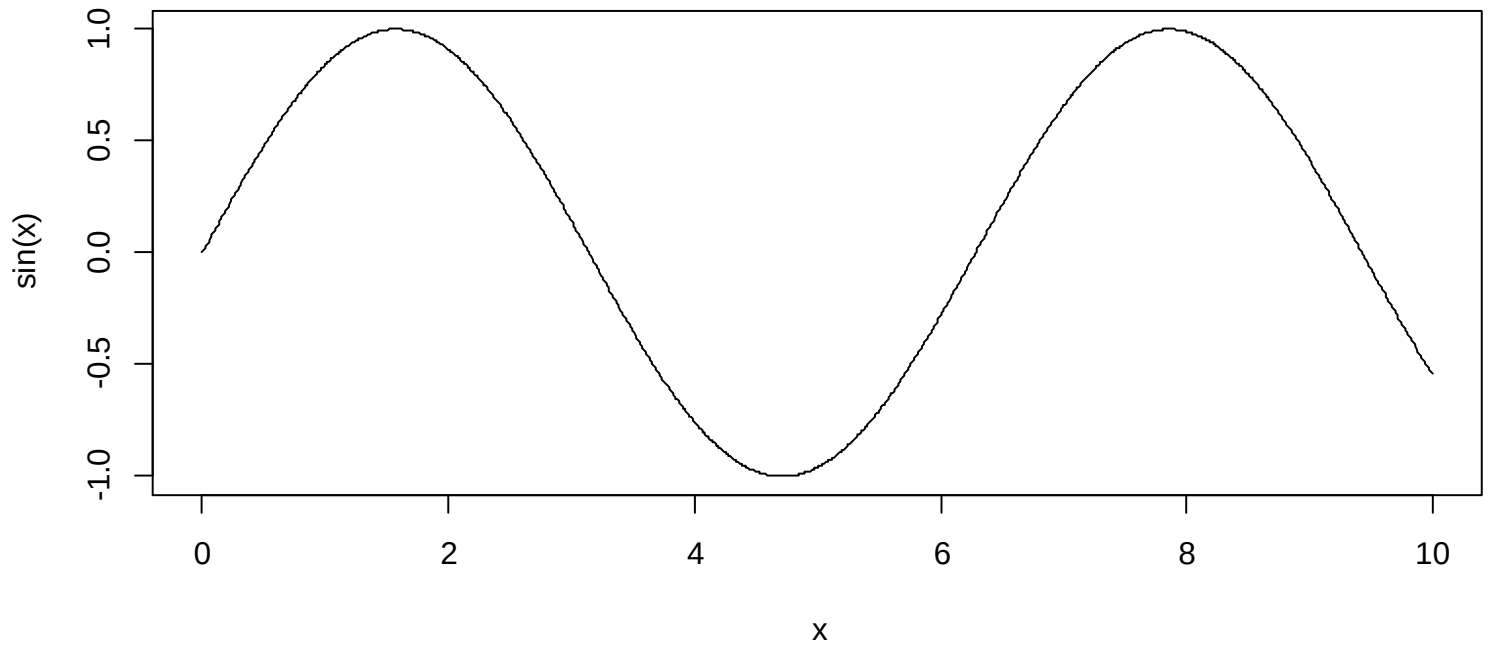
Sequences

- Use colon :
- Use seq() function

```
> 5:9
[1] 5 6 7 8 9
> seq(5,9)
[1] 5 6 7 8 9
> seq(5,9,by = 1)
[1] 5 6 7 8 9
> seq(5,9,by = 0.5)
[1] 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0
> seq(from = 5, to = 9, by = 0.4)
[1] 5.0 5.4 5.8 6.2 6.6 7.0 7.4 7.8 8.2 8.6 9.0
> seq(0, 1, length.out = 11)
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

Plot

```
> x <- seq(0,10, by=0.01)
> plot(x,sin(x),type ="l")
```



Repetitions

- Use rep( ) function

```
> rep(1:4, 3)
[1] 1 2 3 4 1 2 3 4 1 2 3 4
> rep(1:4, each = 3)
[1] 1 1 1 2 2 2 3 3 3 4 4 4
> rep(1:4, c(3,3,3,3))
[1] 1 1 1 2 2 2 3 3 3 4 4 4
> rep(1:4, c(1,2,3,4))
[1] 1 2 2 3 3 3 4 4 4 4
```

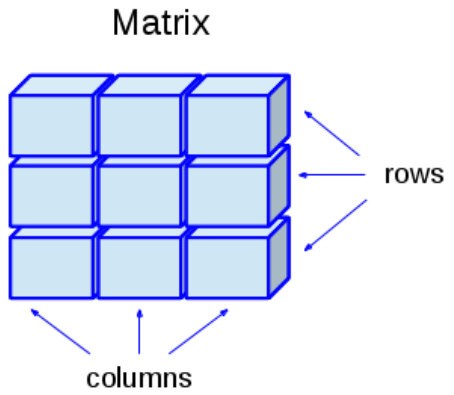
Vector Math

- Most arithmetic operations work as well

```
> a <- c(1,2,3,4)
> b <- a + 2
> a*2
[1] 2 4 6 8
> a/3
[1] 0.3333333 0.6666667 1.0000000 1.3333333
> a^2
[1] 1 4 9 16
> a<b
[1] TRUE TRUE TRUE TRUE
> sin(b)
[1] 0.1411200 -0.7568025 -0.9589243 -0.2794155
```

Matrices

- A matrix is a bi-dimensional array
  - Rows
  - Columns



Colum Names and Row Names

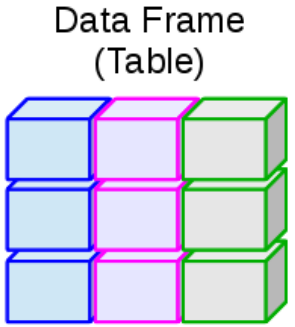
- Use colnames and rownames to retrieve or set the row and column names of a matrix-like object.

```
> m2 <- matrix(1:12,ncol = 4, byrow = TRUE)
> m2
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
> colnames(m2) <- c("a","b","c","d")
> rownames(m2) <- c("i","j","k")
> m2
  a  b  c  d
i 1  2  3  4
j 5  6  7  8
k 9 10 11 12
> colnames(m2)
[1] "a" "b" "c" "d"
> rownames(m2)
[1] "i" "j" "k"
```

Data Frames

- Data frame is a list of vectors of equal length.
- Each column should be of the same type.
- Similar to tables in RDBMS, or data set in SAS or SPSS, i.e. a “cases by variables” matrix of data.

```
> id <- c(11,12,13)
> name <- c("Lily","Jim","Tom")
> credit <- c(710,700,680)
> df <- data.frame(id,name,credit)
> df
  id name credit
1 11 Lily    710
2 12  Jim    700
3 13  Tom    680
> df["name"]  # Show the name column
  name
1 Lily
2  Jim
3  Tom
> df[["name"]]
[1] Lily Jim  Tom
Levels: Jim Lily Tom
```



Motor Trend Data Built in R

- The mtcars is a built-in data frame which comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

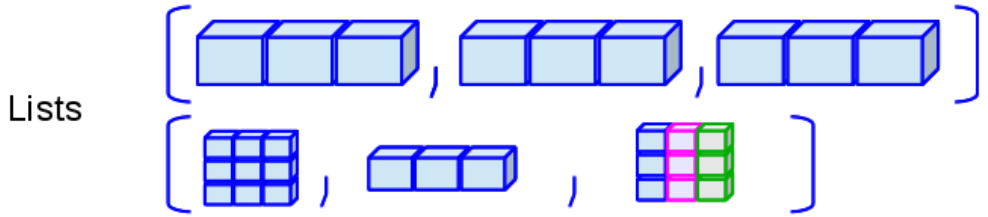
```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> head(mtcars,n=3)
               mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4     21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710    22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
> tail(mtcars)
                 mpg cyl  disp  hp drat    wt qsec vs am gear carb
Porsche 914-2   26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
Lotus Europa    30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
Ford Pantera L  15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
Ferrari Dino    19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
Maserati Bora   15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
Volvo 142E      21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

Summary of All Variables(Columns)

```
> summary(mtcars)  # Summary of all variables(columns)
      mpg             cyl             disp             hp             drat      
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0   Min.   :2.760  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080  
 Median :19.20   Median :6.000   Median :196.3   Median :123.0   Median :3.695  
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7   Mean   :3.597  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920  
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0   Max.   :4.930  
       wt             qsec             vs              am       
 Min.   :1.513   Min.   :14.50   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :3.325   Median :17.71   Median :0.0000   Median :0.0000  
 Mean   :3.217   Mean   :17.85   Mean   :0.4375   Mean   :0.4062  
 3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :5.424   Max.   :22.90   Max.   :1.0000   Max.   :1.0000  
      gear            carb      
 Min.   :3.000   Min.   :1.000  
 1st Qu.:3.000   1st Qu.:2.000  
 Median :4.000   Median :2.000  
 Mean   :3.688   Mean   :2.812  
 3rd Qu.:4.000   3rd Qu.:4.000  
 Max.   :5.000   Max.   :8.000  
```

Lists

- A list is a special type of vector. Elements can be of different types.
- Use lists act as containers.



Agenda

- What is R?
- Program with RStudio
- Introduction to R Markdown
- Data Structures in R
- R Functions
- Control Structures (Selection and Loop)

Functions as Building Blocks of Software

- Functions are reusable pieces of programs.
- They allow you to give a name to a block of statements, then run that block using the specified name anywhere in your program and any number of times.
- R has a rich set of built-in functions such as length(), summary().
- You can define your own function and call it in other places.

Functions: Closure Type Objects

- So many built-in functions available
- You can define your own functions
  - Function name
  - Input (argument list)
  - Output

```
> f2c <- function(f){
+ # Fahrenheit to Celsius conversion
+ c <- (f-32)*5/9
+ return(c)
+ }
> f2c(90)
[1] 32.22222
> f2c(32)
[1] 0
> typeof(f2c)
[1] "closure"
```

Writing Your Own Functions

- Syntax

  function (arglist) body

  - The keyword function indicates that you want to create a function.
  - An argument list is a comma separated list of formal arguments. A formal argument can be a symbol, a statement of the form 'symbol = expression', or the special formal argument '...'.
  - The body can be any valid R expression. Generally, the body is a group of expressions contained in curly braces ('{' and '}') called block.
  - Generally functions are assigned to symbols but they don't need to be (anonymous functions).

Source: https://cran.r-project.org/doc/manuals/r-release/R-lang.html

(cont.)

- Formal arguments define the variables whose values will be supplied at the time the function is invoked. The names of these arguments can be used within the function body.

- Default values for arguments can be specified using the special form ‘name = expression’. In this case, if the user does not specify a value for the argument when the function is invoked the expression will be associated with the corresponding symbol.

Source: https://cran.r-project.org/doc/manuals/r-release/R-lang.html

Programming Style and Documentation

- Programming style is important
  - Good programming style makes a program more readable
  - Good programming style helps reduce programming errors

- Several guidelines
  - Appropriate Comments
  - Naming Conventions
  - Proper Indentation and Spacing Lines

Google's R Style Guide

- https://google.github.io/styleguide/Rguide.xml

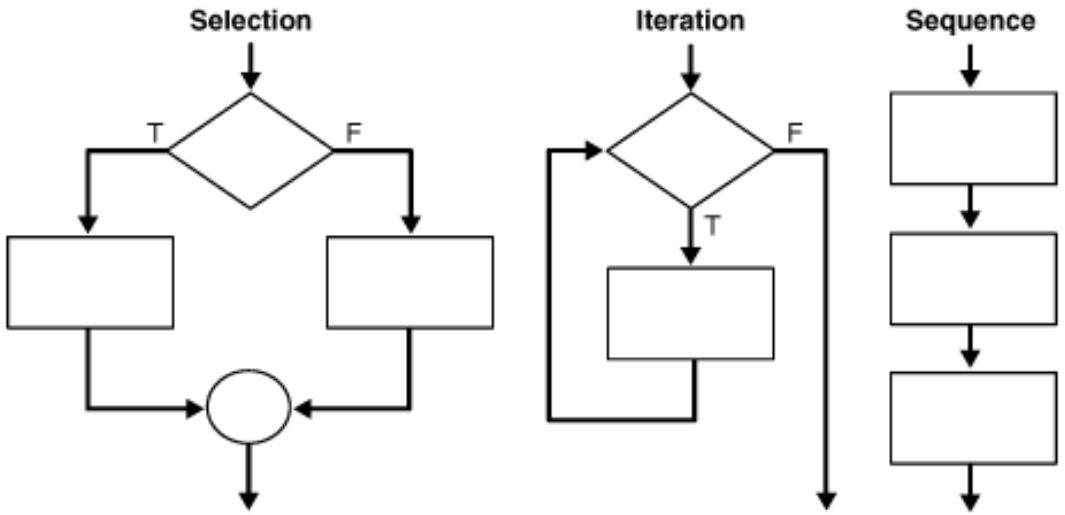
1. File Names: end in `.R`
2. Identifiers: `variable.name` (or `variableName`), `FunctionName`, `kConstantName`
3. Line Length: maximum 80 characters
4. Indentation: two spaces, no tabs
5. Spacing
6. Curly Braces: first on same line, last on own line
7. else: Surround else with braces
8. Assignment: use `<-`, not `=`
9. Semicolons: don't use them
10. General Layout and Ordering
11. Commenting Guidelines: all comments begin with `#` followed by a space; inline comments need two spaces before the `#`
12. Function Definitions and Calls
13. Function Documentation
14. Example Function
15. TODO Style: `TODO(username)`

Agenda

- What is R?
- Program with RStudio
- Introduction to R Markdown
- Data Structures in R
- R Functions
- Control Structures (Selection and Loop)

Structure Theorem

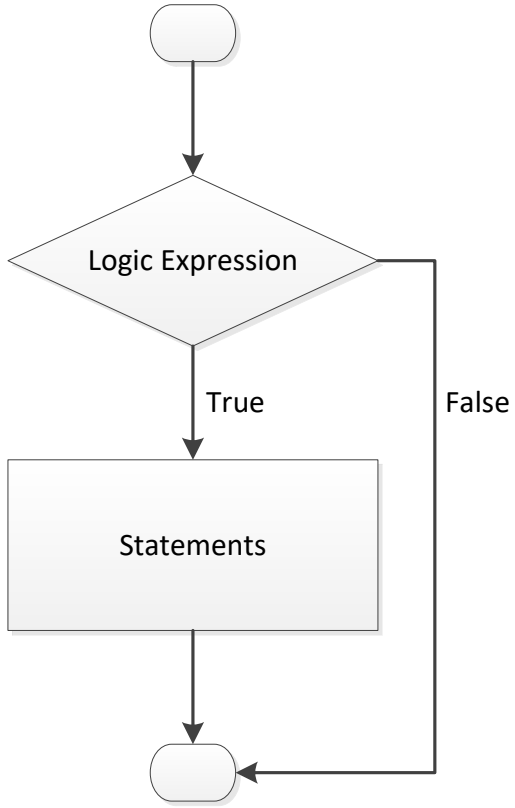
- According to the *structure theorem*, any computable program can be written using three basic control structures:
  - Sequence: executing one subprogram, and then another subprogram
  - Selection: executing one of two subprograms according to the value of a boolean expression
  - Iteration (loop): executing a subprogram until a boolean expression is true



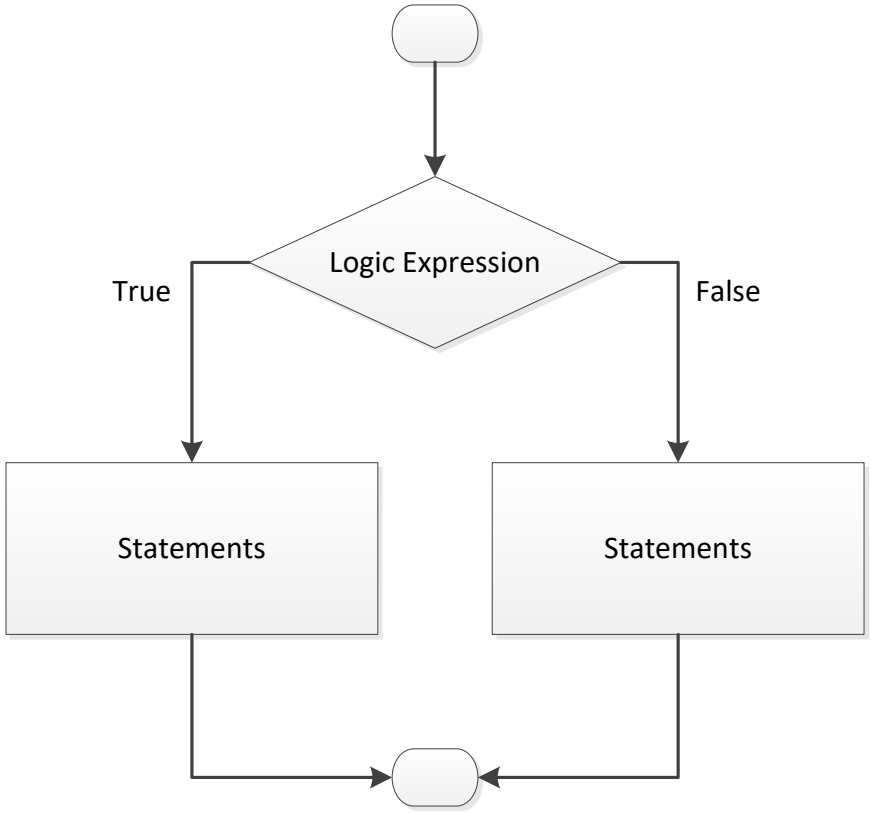
Reading: http://en.wikipedia.org/wiki/Structured_program_theorem

Selection Structure

One-way selection structure



Two-way selection structure



One-Way Selection Structure in R

- Syntax

  if(logic expression) {...}

Function

```
is.even <- function(x){
  if(x%%2==0){
    return(TRUE)
  }
}
```

Test

```
> is.even(24)
[1] TRUE
> is.even(23)
> is.even(10.5)
```

Two-Way Selection Structure in R

- Syntax

if(logic expression) {...} else {...}

Function

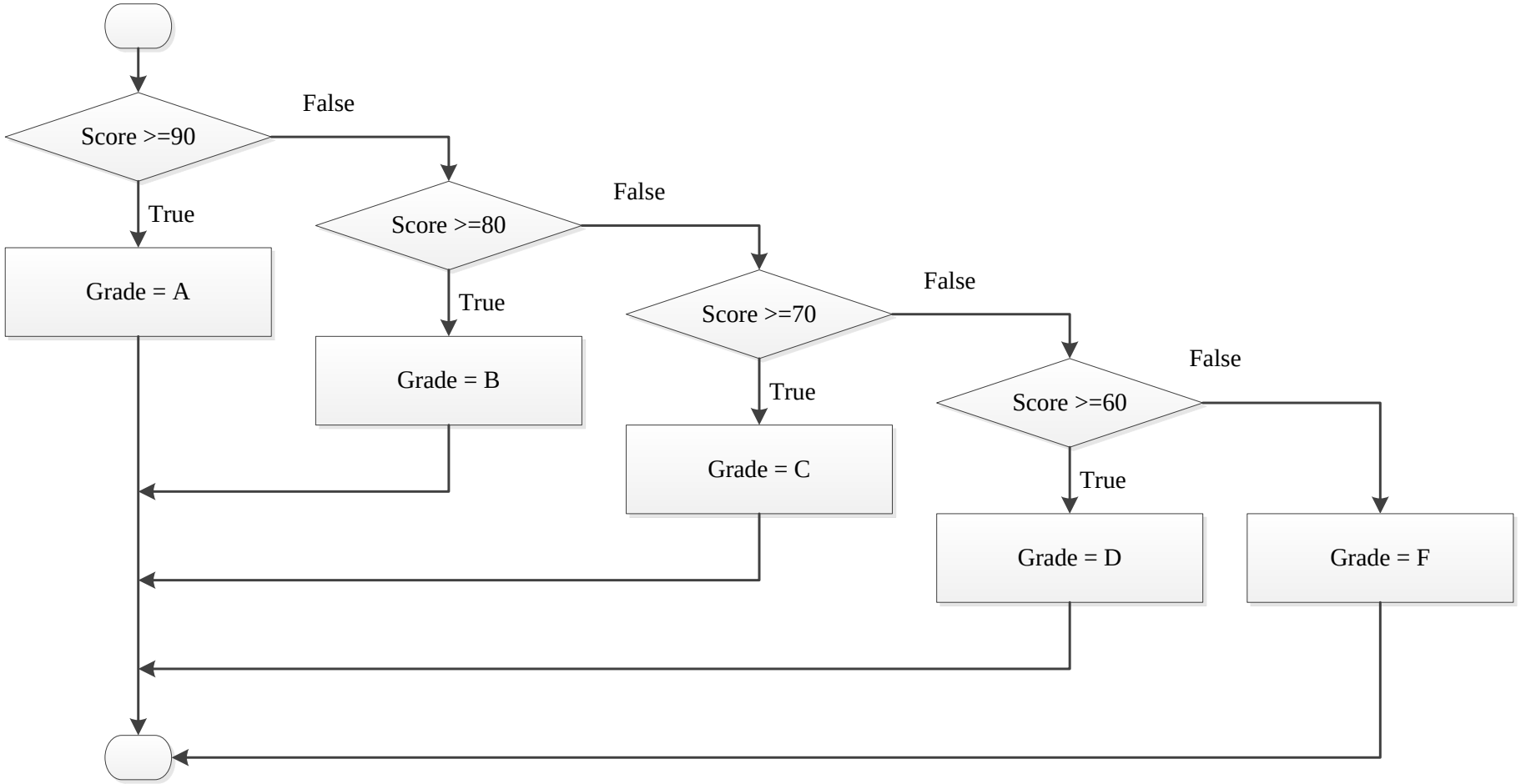
```
is.even2 <- function(x){
  if(x%%2==0){
    return(TRUE)
  } else{
    return(FALSE)
  }
}
```

Test

```
> is.even2(24)
[1] TRUE
> is.even2(23)
[1] FALSE
> is.even2(10.5)
[1] FALSE
```

Multi-Way Selection Structure

- Example: convert score to letter grade



Multi-Way Selection Structure in R

Function

```
score2grade <- function(score){
  if(score >= 90) return("A")
  else if (score >=80) return("B")
        else if (score >= 70) return("C")
              else if (score >= 60) return("D")
                    else return("F")
}
```

Test

```
> score2grade(99)
[1] "A“
> score2grade(90.1)
[1] "A“
> score2grade(89.9)
[1] "B“
> score2grade(70.1)
[1] "C“
> score2grade(68.6)
[1] "D“
> score2grade(57)
[1] "F"
```

The Nearest Rule (if else ambiguity)

The else clause matches the nearest preceding if clause in the same block.

Loop Structure in R

- R provides three statements to support looping
  - for statement
  - while statement
  - repeat statement
- Two statements used to explicitly control looping
  - break statement
  - next statement

for Loop

- Syntax

for (*name* in *vector*)
    statement

```
# Generate random scores for 100 students
score_v <- sample(50:100, 100, replace=T)
print(score_v)

# Use for loop
grade_v <- NULL  # Initiate a grade vector
for (i in 1:100)
    grade_v[i] = score2grade(score_v[i])
print(grade_v)  # Show the grades calculated
```

while Loop

- Syntax

```
while (logic expression)
    statement
```

```
# Use while loop
grade_v <- NULL
i <- 1
while (i <= 100){
  grade_v[i] = score2grade(score_v[i])
  i <- i + 1
}
print(grade_v)  # Show the grades calculated
```

repeat Loop

- Syntax

repeat statement

```
# Use repeat loop
grade_v <- NULL
i <- 1
repeat {
  grade_v[i] = score2grade(score_v[i])
  i <- i + 1
  if (i == 101) break
}
print(grade_v)  # Show the grades calculated
```

Which Loop to Use?

- The three forms of loop statements, for, while, and repeat, are expressively equivalent.
- You can write a loop in any of these three forms.

Guidelines for Choosing Loop Structures

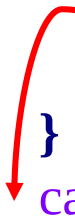
- Use the one that is most intuitive and comfortable for you.
- In general, a for loop may be used if the number of repetitions is known, as, for example, when you need to print a message 100 times.
- A while loop may be used if the number of repetitions is not known, as in the case of reading the numbers until the input is 0.
- A repeat loop can be used to replace a while loop if the loop body has to be executed before testing the continuation condition.

Using break and next

- The break and next keywords provide additional controls in a loop.
- break statement breaks out of the loop.
- continue statement bypasses the current iteration.

break Statement

```
sum <- 0
 i <- 0
 while(i < 20){
    i <- i +1
    if (sum >= 100)
       break
    sum <- sum + i
 }
 cat("The i is",i,"\n")
 cat("The sum is",sum,"\n")
```



break statement breaks out of the loop.

sum = 1 + 2 + 3 + ...+14 = 105

next Statement

```
sum <- 0
    i <- 0
    while(i < 20){
       i <- i + 1
       if (i == 10 | i ==11 )
          next
       sum <- sum + i
    }
    cat("The i is",i,"\n")
    cat("The sum is",sum,"\n")
```

next statement bypasses the current iteration.

sum = 1 + 2 + ...+ 8 + 9 + 12 + 13 + ... + 20 = 189

Q & A

IST 5535: Machine Learning Algorithms and Applications

Langtao Chen, Spring 2021

3. Linear Regression

Reading

- Book Chapter 3

Learning Objectives

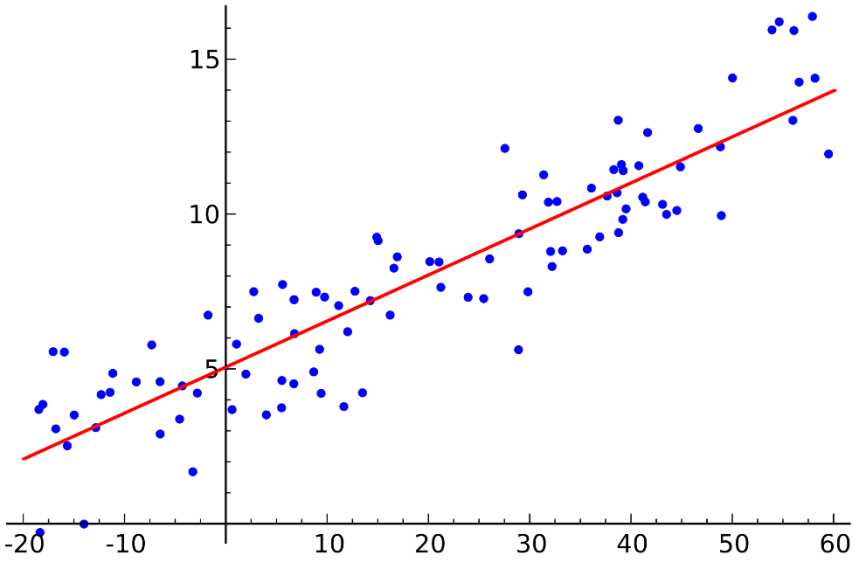
- Understand linear regression coefficient estimation and the ways of assessing the accuracy of coefficient estimates and the accuracy of the model.
- Understand methods dealing with qualitative predictors in linear regression.
- Understand interaction terms in linear regression.
- Understand non-linear relationship fit using polynomial regression.
- Understand potential problems of linear regression.
- Understand the comparison between linear regression and KNN regression.
- Be able to use R to conduct linear regression analysis and use diagnostic plots to check potential issues in linear regression.

Agenda

- Linear Regression
  - Estimating the Coefficients
  - Assessing the Accuracy of Coefficient Estimates
  - Assessing the Accuracy of the Model
- Other Considerations in Regression Model
  - Qualitative Predictors
  - Extensions of the Linear Model
  - Potential Problems
- Linear Regression vs. KNN

What is Regression?

- Regression is about estimating relationships between variables.
- Regression is a statistical technique that attempts to build a function of independent variables (regressors, input variables, or predictors) to predict or explain a dependent variable (response, or outcome).
- Regression intends to summarize observed data as simply and usefully as possible.



Major Objectives of Regression Analysis

- Explanatory modeling
  - The purpose is to explain or quantify the effect of independent variables on dependent variable
  - The classical statistical approach
  - Focus on unveiling the underlying relationship between variables
  - Use the entire dataset to fit the model with the data

- Predictive modeling
  - Predict the outcome value for new records, given value(s) of their input variable(s)
  - Focus on predictive performance rather than coefficients (beta)
  - Train the model on a training dataset and evaluate its performance on a test dataset

Linear Regression Model

- Linear regression model is a special case of the parametric model

$$y = X\beta + \varepsilon$$

where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

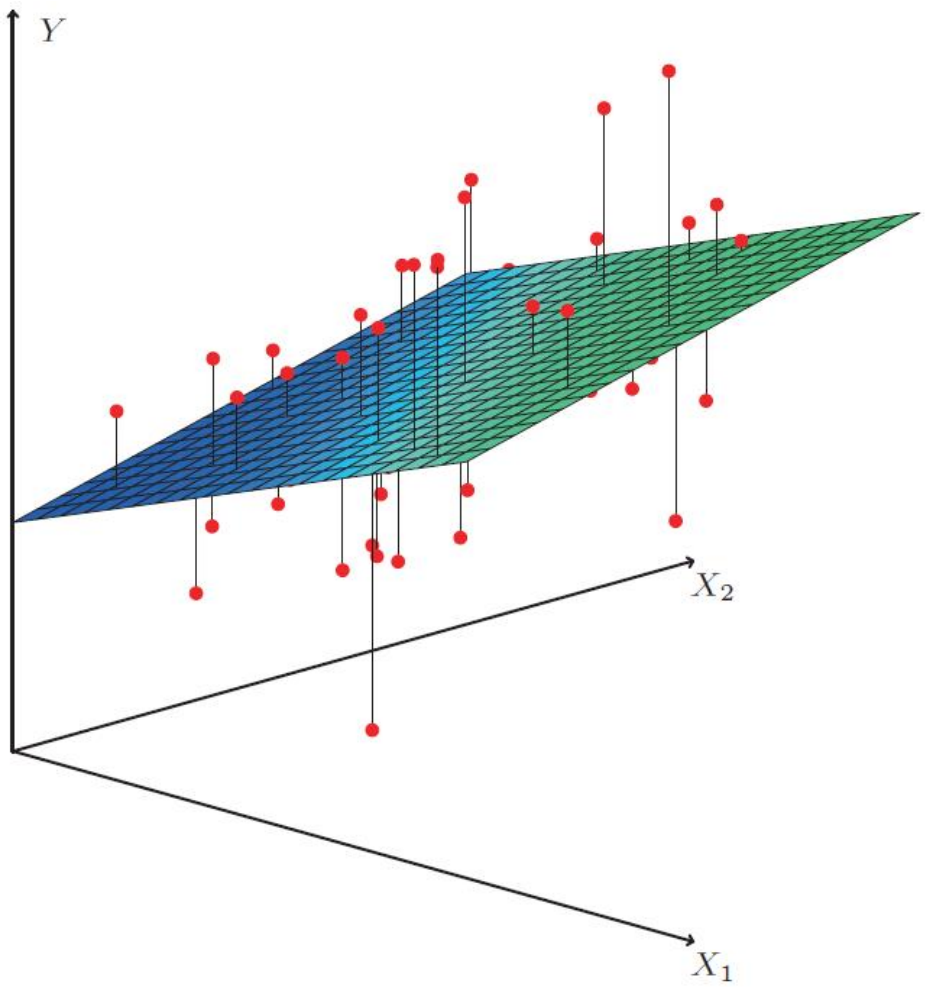
- Benefits
  - The simple linear functional form is easy to estimate
  - Interpretation is straightforward

Estimate Linear Regression Parameters

- Use ordinary least squares (OLS) to find $\hat{\beta}$ that minimize MSE

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\beta}X\right)^2$$



Formally Define OLS: Optimization Problem

Set $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$ $X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$ $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$

Then $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = (y - X\beta)^T (y - X\beta)$

To find the best model is to find the parameters $\hat{\beta}$ that minimize the SSE (error sum of squares).

Formal definition:

$$\hat{\beta} = arg \min_{\beta} [(y - X\beta)^T (y - X\beta)]$$

Ordinary Least Squares (OLS) Estimator

Target: $\hat{\beta} = arg\min_{\beta}[(y - X\beta)^T(y - X\beta)]$

According to optimization theory, the optimal parameter $\hat{\beta}$ satisfies the following conditions:

1. First order condition (F.O.C.):

$$\frac{\partial[(y - X\beta)^T(y - X\beta)]}{\partial\beta} = 0 \Leftrightarrow \frac{\partial[y^Ty - 2\beta^TX^Ty + \beta^TX^TX\beta]}{\partial\beta} = 0 \Leftrightarrow X^TX\hat{\beta} - X^Ty = 0$$

$$\Leftrightarrow X^TX\hat{\beta} = X^Ty$$

If X^TX is invertible, $\hat{\beta} = (X^TX)^{-1}X^Ty$.

2. Second order condition (S.O.C.): $\frac{\partial^2[(y - X\beta)^T(y - X\beta)]}{\partial\beta\partial\beta^T} = X^TX \geqslant 0$ (positive semi definite)

Thus, $\hat{\beta} = (X^TX)^{-1}X^Ty$ minimizes the SSE. This is called the OLS estimator (closed form solution).

Simple Linear Regression

$$y = \beta_0 + \beta_1 x$$

- One dependent variable (y): the one to predict or explain
- One independent variable (x): explanatory variable/predictor
- β_0: intercept
  - When x equals to zero, what is the value of y.
- β_1: slope
  - Increase x by one unit, how much would y change.

Multiple Linear Regression

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ \ldots\ldots + \beta_p x_p$$

- One dependent variable (y): the one to predict or explain
- Multiple independent variables (x_1, x_2, … x_n): explanatory
- β_0 : intercept
  - When all explanatory variables are zero, what is the value of y.
- β_i : slope (i >= 1)
  - Increase x_i by one unit, how much would y change after controlling for other factors.

Inference in Regression

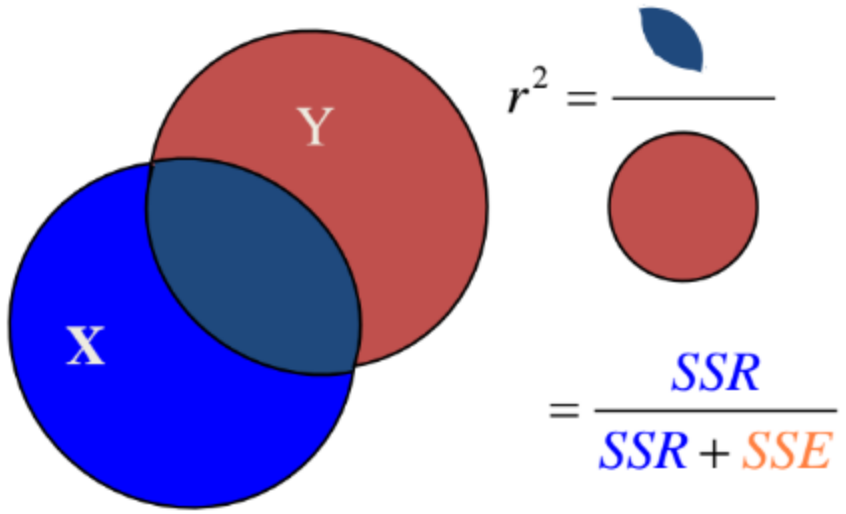
- How well does the regression model fit the data?
- What is the relationship between X and Y?
- What is the expected value of Y given an X value?

Measure of Fit: R^2

- The proportion of variation in Y that is explained by the independent variable X in the regression model.

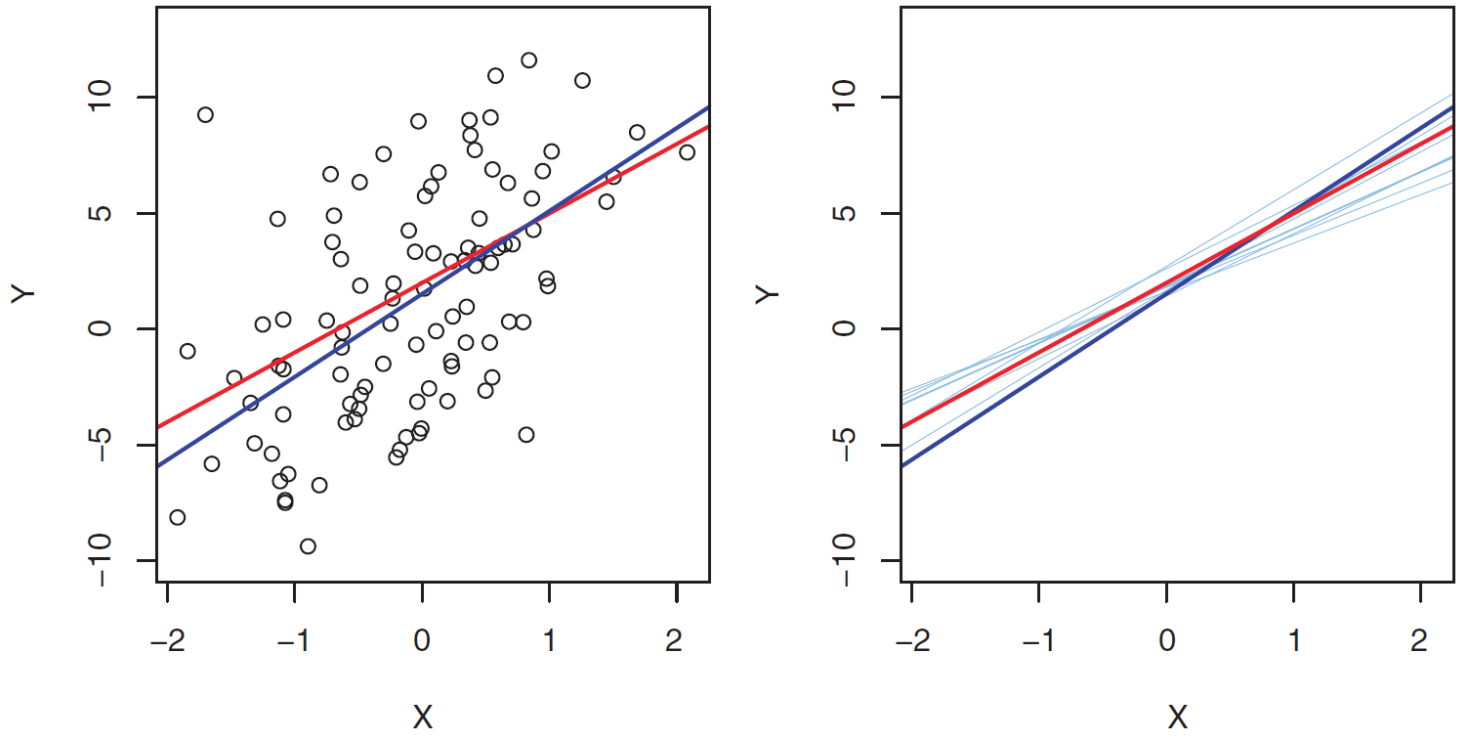
$$R^2 = 1 - \frac{SSE}{\sum(Y_i - \bar{Y})^2} = 1 - \frac{residual\ sum\ of\ squares}{total\ variance\ of\ Y}$$

- $R^2 \in [0, 1]$
  - $R^2 = 0$: X does not explain any variance of Y
  - $R^2 = 1$: X fully explains variance of Y (perfect fit)



Assessing the Accuracy of the Coefficient Estimates

- Different datasets result in slightly different OLS estimates.



- Red: population regression line (Y = 2 +3X) which is unknown in real dataset
- Dark Blue: OLS line estimated from observed data
- Light Blue: OLS line estimated from a random set of observations

Assessing the Accuracy of the Coefficient Estimates

- Population regression line (unknown)

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ \ldots\ldots + \beta_p x_p$$

- OLS line

$$\hat{y} = \widehat{\beta_0} + \widehat{\beta_1} x_1 + \widehat{\beta_2} x_2 + \ \ldots\ldots + \widehat{\beta_p} x_p$$

Statistical Inference is all about using an estimate from observed data to guess the true parameters.

We can calculate 95% confidence interval for β_j:

$$\left[\widehat{\beta_j} - 2 * SE\left(\widehat{\beta_j}\right), \widehat{\beta_j} + 2 * SE\left(\widehat{\beta_j}\right)\right]$$

The range contains the true value of the parameter with 95% probability.

Hypothesis Tests on Coefficients

- Null hypothesis

  H_0: There is no relationship between X_j and Y

- Alternative hypothesis

  H_a: There is some relationship between X_j and Y

Mathematically,

$$H_0: \beta_j = 0$$
$$H_a: \beta_j \neq 0$$

- In practice, we conduct a t-test to assess whether there is a relationship between X_j and Y:

$$t = \frac{\hat{\beta}_j - 0}{SE(\hat{\beta}_j)}$$

Example

```
> lm.fit=lm(sales ~ TV + radio + newspaper, data = Advertising)
> summary(lm.fit)

Call:
lm(formula = sales ~ TV + radio + newspaper, data = Advertising)

Residuals:
    Min      1Q  Median      3Q     Max
-8.8277 -0.8908  0.2418  1.1893  2.8292

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.938889   0.311908   9.422   <2e-16 ***
TV           0.045765   0.001395  32.809   <2e-16 ***
radio        0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.686 on 196 degrees of freedom
Multiple R-squared:  0.8972,  Adjusted R-squared:  0.8956
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

1. How well does the model fit the data?

```
> stargazer::stargazer(lm.fit, type = "text")
================================
                  Dependent variable:
                      sales
--------------------------------
TV                   0.046***
                    (0.001)
radio                0.189***
                    (0.009)
newspaper           -0.001
                    (0.006)
Constant             2.939***
                    (0.312)
--------------------------------
Observations          200
R2                    0.897
Adjusted R2           0.896
Residual Std. Error   1.686 (df = 196)
F Statistic           570.271*** (df = 3; 196)
================================
Note:           *p<0.1; **p<0.05; ***p<0.01
```

89.7% of the variance of sales can be explained by TV, radio, and newspaper advertising budgets.

2. What is the relationship between X and Y?

```
> stargazer::stargazer(lm.fit, type = "text")
================================
                  Dependent variable:
                  sales
--------------------------------
TV                0.046***
                  (0.001)
radio             0.189***
                  (0.009)
newspaper         -0.001
                  (0.006)
Constant          2.939***
                  (0.312)
--------------------------------
Observations      200
R2                0.897
Adjusted R2       0.896
Residual Std. Error  1.686 (df = 196)
F Statistic       570.271*** (df = 3; 196)
================================
Note:             *p<0.1; **p<0.05; ***p<0.01
```

- P-value <0.001
  X and Y are statistically significantly related at an alpha level of 0.001
- P-value <0.01
  X and Y are statistically significantly related at an alpha level of 0.01
- P-value <0.05
  X and Y are statistically significantly related at an alpha level of 0.05

TV budget and sales are statistically significantly related at an alpha level of 0.01, controlling for other factors.

Radio budget and sales are statistically significantly related at an alpha level of 0.01, controlling for other factors.

Newspaper budget and sales are NOT statistically significantly related, controlling for other factors.

3. What is the expected value of Y given an X value?

```
> stargazer::stargazer(lm.fit, type = "text")
================================
                  Dependent variable:
                      sales
--------------------------------
TV                    0.046***
                      (0.001)
radio                 0.189***
                      (0.009)
newspaper             -0.001
                      (0.006)
Constant              2.939***
                      (0.312)
--------------------------------
Observations            200
R2                      0.897
Adjusted R2             0.896
Residual Std. Error     1.686 (df = 196)
F Statistic         570.271*** (df = 3; 196)
================================
Note:               *p<0.1; **p<0.05; ***p<0.01
```

$$\widehat{sales} = 2.939 + 0.046 * TV + 0.189 * radio - 0.001 * newspaper$$

Predict sales when allocating all \$300k budgets to TV

$$\widehat{sales} = 2.939 + 0.046 * 300 + 0.189 * 0 - 0.001 * 0$$
$$= 16.739 \text{ (thousand units)}$$

Agenda

- Linear Regression
  - Estimating the Coefficients
  - Assessing the Accuracy of Coefficient Estimates
  - Assessing the Accuracy of the Model
- Other Considerations in Regression Model
  - Qualitative Predictors
  - Extensions of the Linear Model
  - Potential Problems
- Linear Regression vs. KNN

Qualitative Predictors (a.k.a. Factors)

- From the Credit dataset, we want to investigate differences in credit card balance between males and females.

```
> str(Credit)
'data.frame': 400 obs. of  12 variables:
 $ ID        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Income    : num  14.9 106 104.6 148.9 55.9 ...
 $ Limit     : int  3606 6645 7075 9504 4897 8047 3388 7114 3300 ...
 $ Rating    : int  283 483 514 681 357 569 259 512 266 491 ...
 $ Cards     : int  2 3 4 3 2 4 2 2 5 3 ...
 $ Age       : int  34 82 71 36 68 77 37 87 66 41 ...
 $ Education: int  11 15 11 11 16 10 12 9 13 19 ...
 $ Gender    : Factor w/ 2 levels " Male","Female": 1 2 1 2 1 1 2  ...
 $ Student   : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 2 ...
 $ Married   : Factor w/ 2 levels "No","Yes": 2 2 1 1 2 1 1 1 1 2 ...
 $ Ethnicity: Factor w/ 3 levels "African American",..: 3 2 2 2 3 ...
 $ Balance   : int  333 903 580 964 331 1151 203 872 279 1350 ...
```

Code Factors as Dummy Variables or Indicator Variables

- Qualitative Predictors with Two Levels
  - For example gender

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

- Qualitative Predictors with More than Two Levels
  - For example ethnicity
  - Create multiple dummy variables

$$ethnicity\_asian_i = \begin{cases} 1 & \text{if the } i\text{th person is Asian} \\ 0 & \text{if the } i\text{th person is not Asian} \end{cases}$$

$$ethnicity\_caucasian_i = \begin{cases} 1 & \text{if the } i\text{th person is Caucasian} \\ 0 & \text{if the } i\text{th person is not Caucasian} \end{cases}$$

Extension of the Linear Model

- Two basic assumptions of linear model
  - Additive assumption: the effect of X_j on Y is independent of other predictors.
  - Linear assumption: the change in Y due to a one-unit change of X_j is constant, regardless of the value of X_j.

Removing the Additive Assumption

- Interaction effect: the effect of X_j on Y is dependent of another predictor X_k
- For example, perhaps spending $50,000 on television advertising and $50,000 on radio advertising results in more sales than allocating $100,000 to either television or radio individually.
- In marketing, this is called a ***synergy*** effect.

Interaction in Advertising

$$sales = \beta_0 + \beta_1 * TV + \beta_2 * Radio + \beta_3 * TV * Radio + \epsilon$$

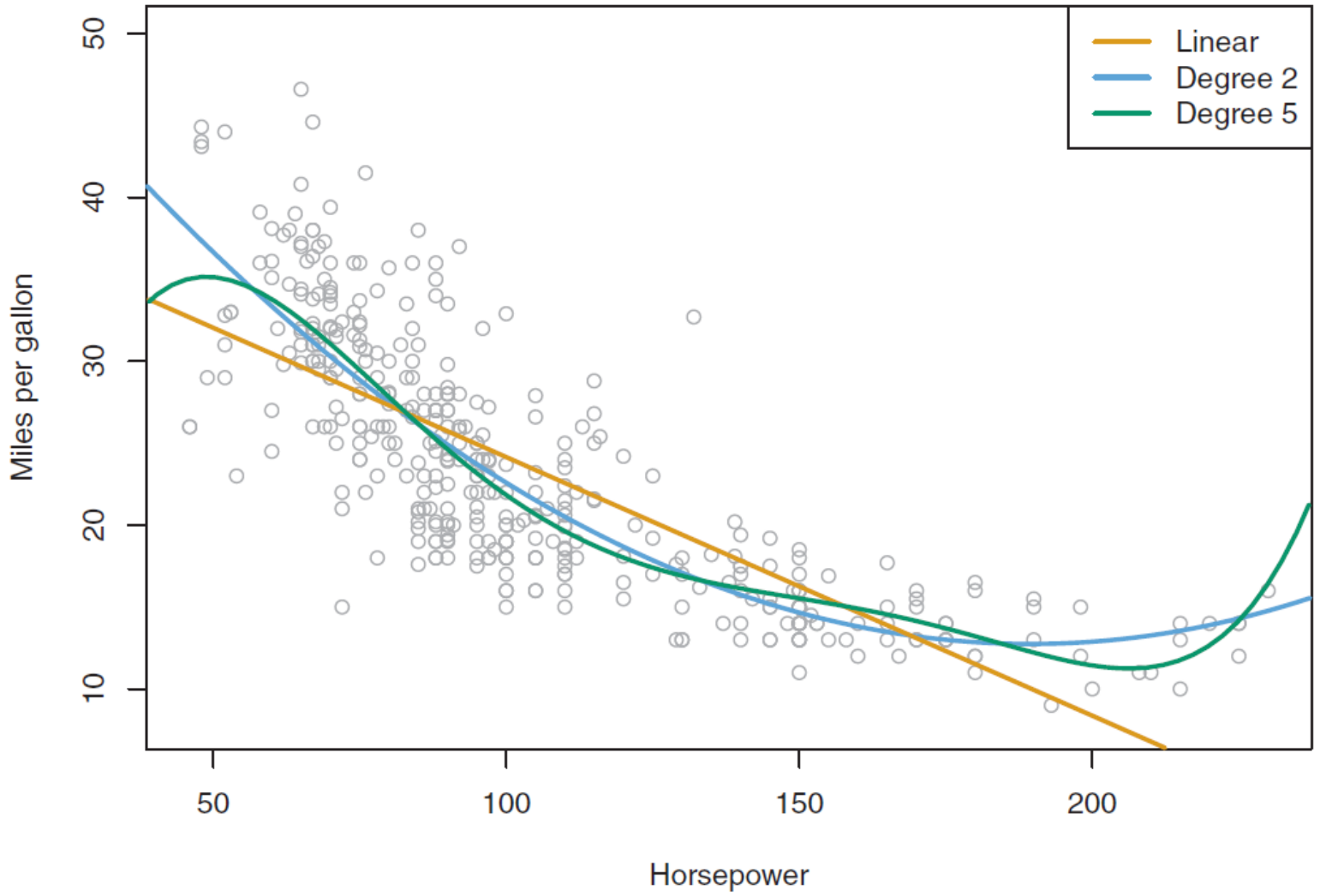
	Coefficient	Std. error	t-statistic	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
radio	0.0289	0.009	3.24	0.0014
TV×radio	0.0011	0.000	20.73	< 0.0001

$$sales = \beta_0 + (\beta_1 + \beta_3 * Radio) * TV + \beta_2 * Radio + \epsilon$$

The effect of TV on sales depends on radio advertising:

$$\tilde{\beta}_1 = \beta_1 + \beta_3 * Radio$$

Non-Linear Relationships



Use Linear Model to Fit Nonlinear Relationship

- Dependent variable y is modeled as an *h-th* degree polynomial of x:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots\ldots + \beta_h x^h + \varepsilon$$

- Relationships when the degree h is low:
  - h=2: quadratic
  - h=3: cubic
  - h=4: quartic

- Polynomial regression is a linear model, since the outcome y is a linear combination of coefficients $\beta_i (i = 1, 2, \ldots h)$

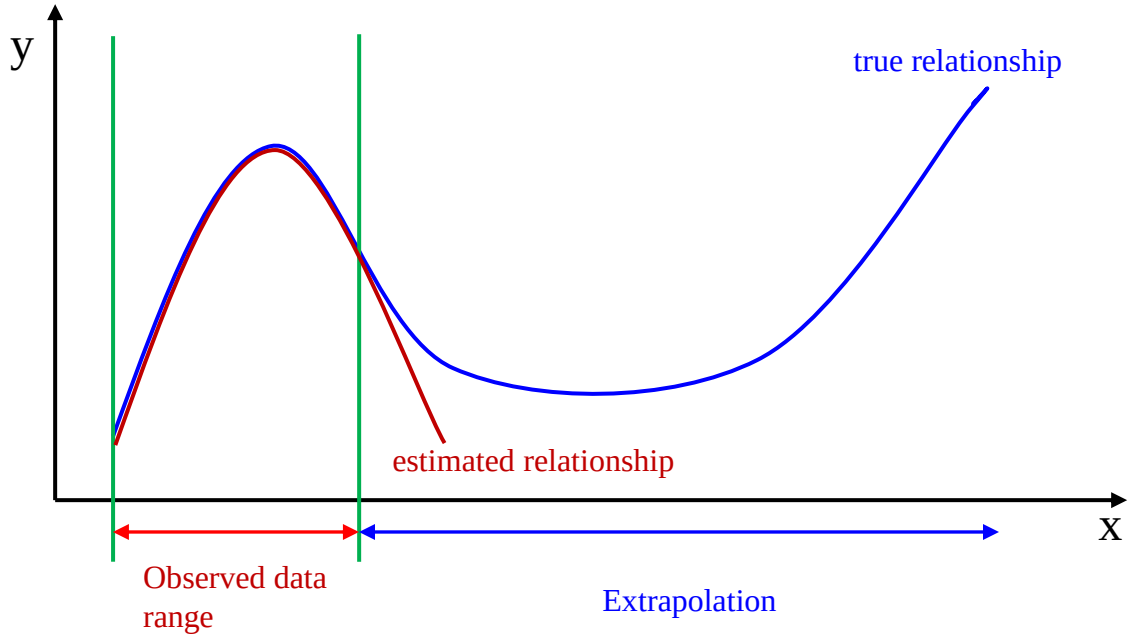
The Hierarchy Principle:

If the polynomial regression model contains x^h and its coefficient is significant, then the model should also include all lower-degree terms $x^j (j < h)$, no matter those x^j are significant or not.

Be Cautious of the Overfitting Issue

- Polynomial regression may be misleading if you don't have a large dataset.

- Do NOT extrapolate beyond your observed data range.



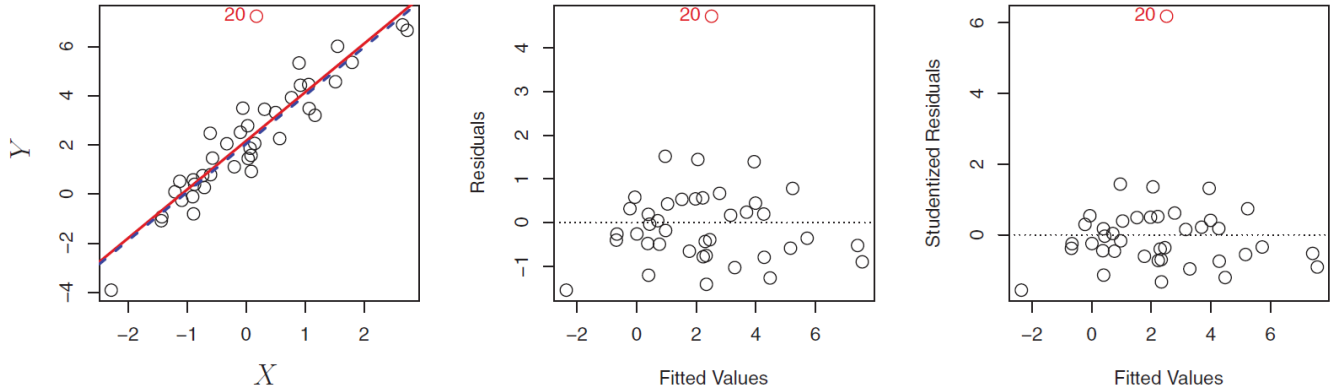
Outside the observed data range, inference or prediction is not reliable.

Potential Fit Problems

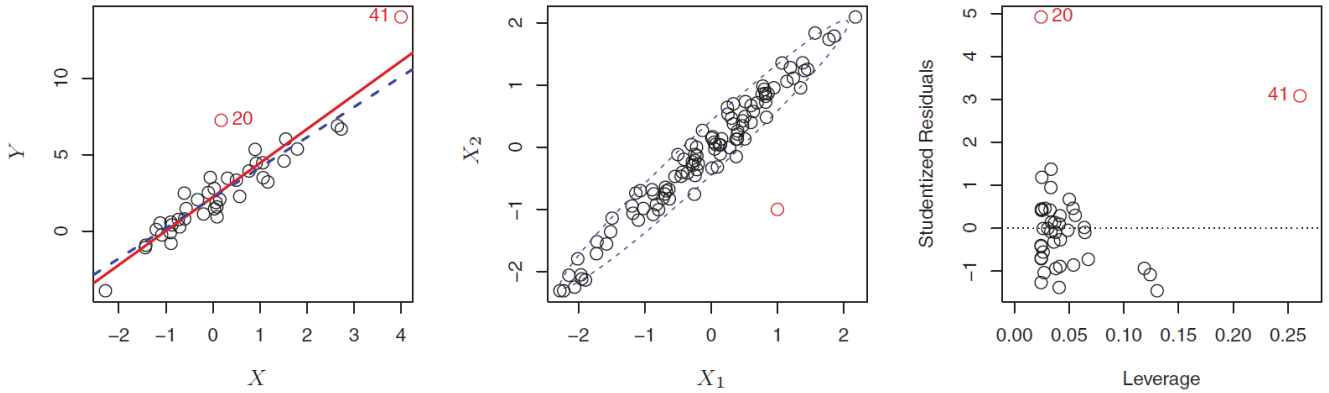
- When we fit a linear regression model to a particular data set, many problems may occur. Most common among these are:
  - 1. *Non-linearity of the response-predictor relationships*
  - 2. *Correlation of error terms*
  - 3. *Non-constant variance of error terms (heteroscedasticity)*
  - 4. *Outliers*
  - 5. *High-leverage points*
  - 6. *Collinearity*

Influential Points

- Outliers: data points with unusually large/small response values



- High leverage points: data points with unusually large/small independent values



$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2}$$

Agenda

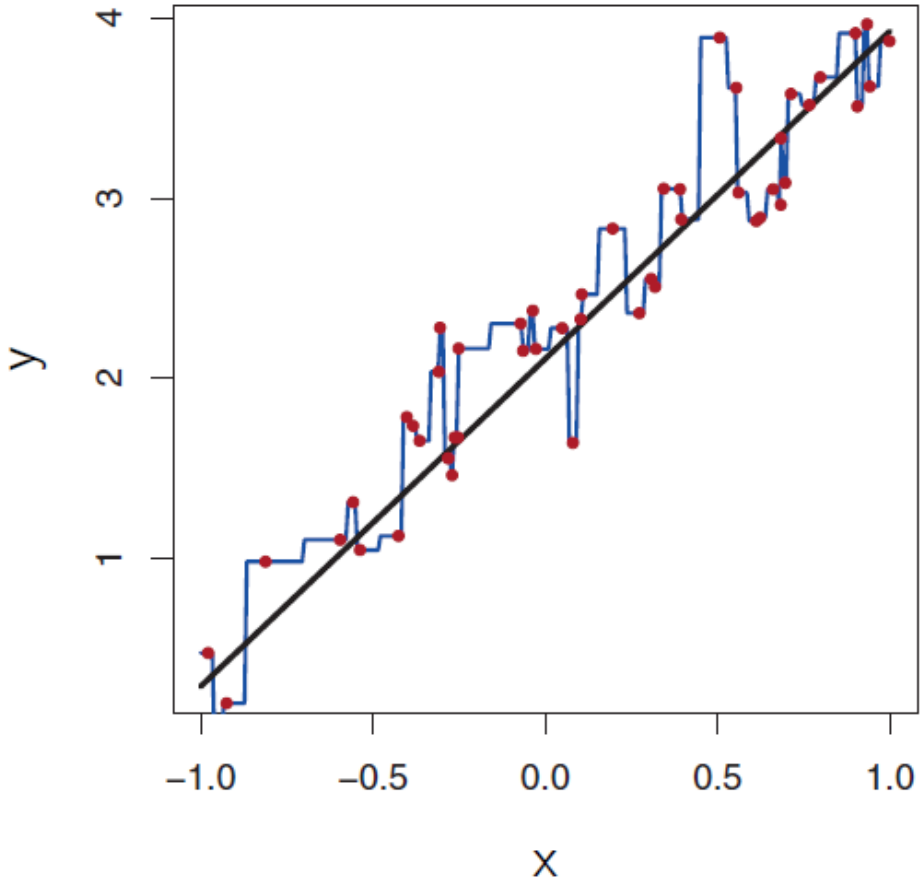
- Linear Regression
  - Estimating the Coefficients
  - Assessing the Accuracy of Coefficient Estimates
  - Assessing the Accuracy of the Model
- Other Considerations in Regression Model
  - Qualitative Predictors
  - Extensions of the Linear Model
  - Potential Problems
- Linear Regression vs. KNN

KNN Regression

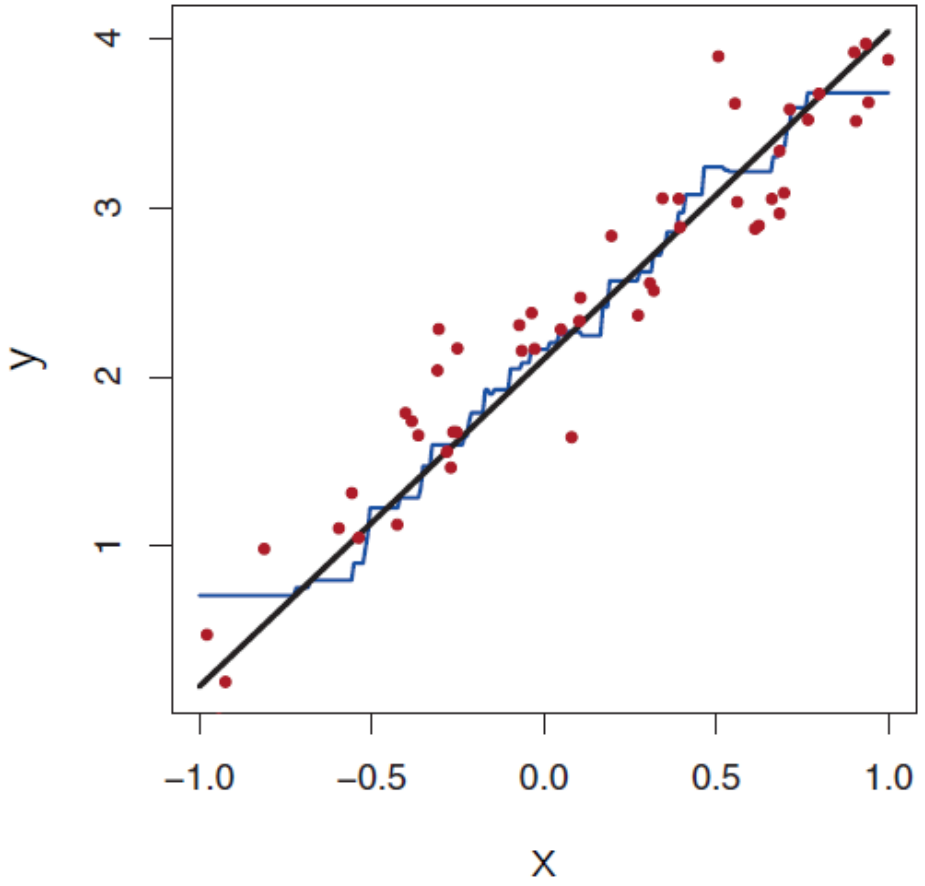
- KNN regression works similar to KNN classification
  - Step 1: Find *K* nearest neighbors;
  - Step 2: Predict the response as the average of *K* neighbors:

$$\hat{f}\,(x_0) = \frac{1}{K} \sum_{x_0 \in N_0} y_i$$

Larger K results in smoother fit



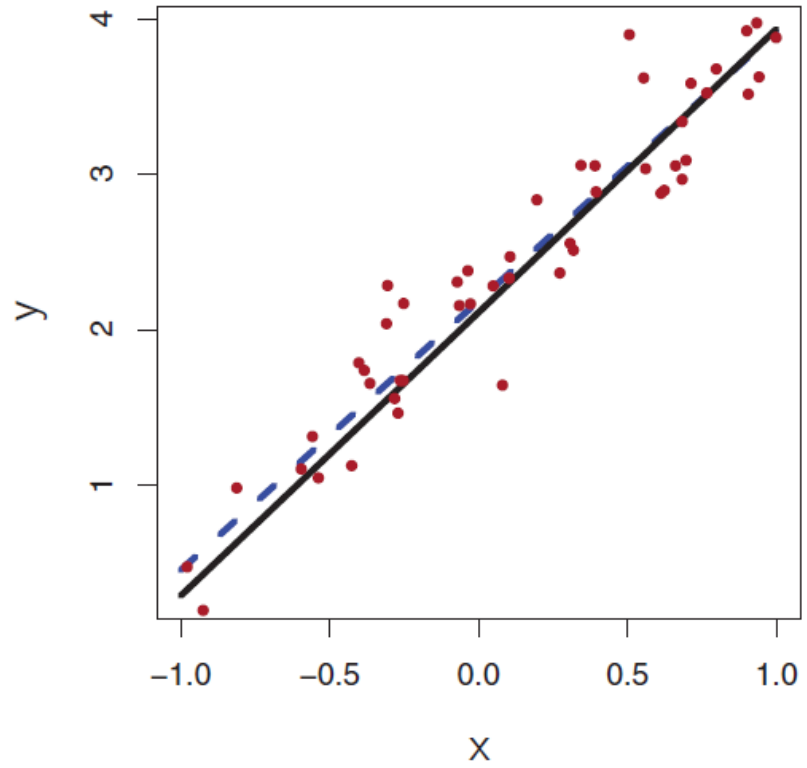
K=1

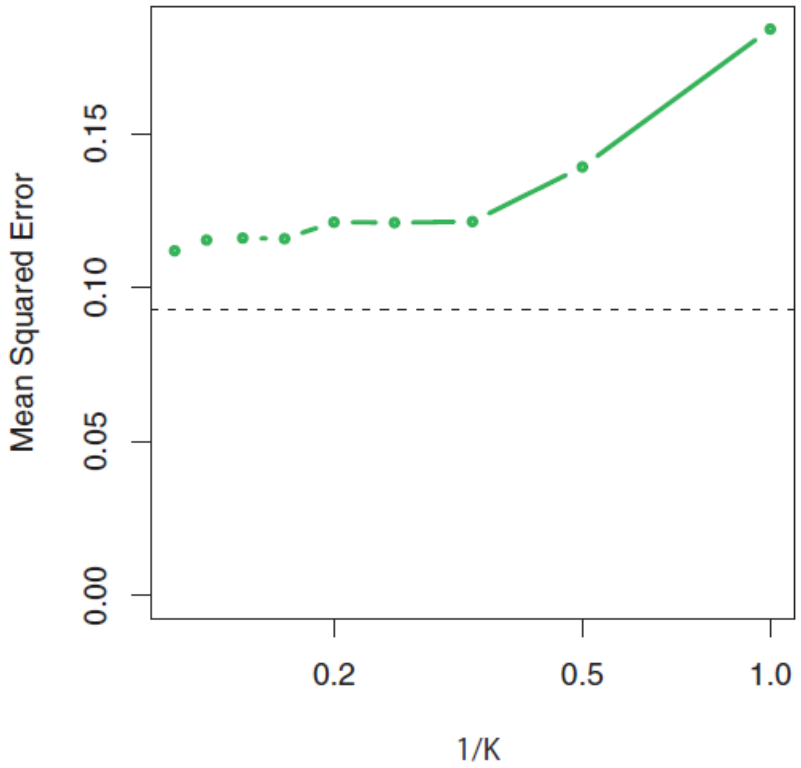


K=9

True Linear Relation, One Dimension (p=1)

- When K is large, KNN performs only a little worse than linear regression;
- When K is small, KNN performs far worse.



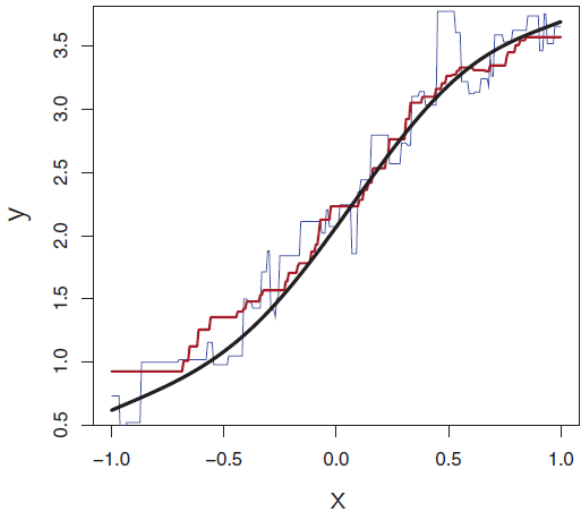


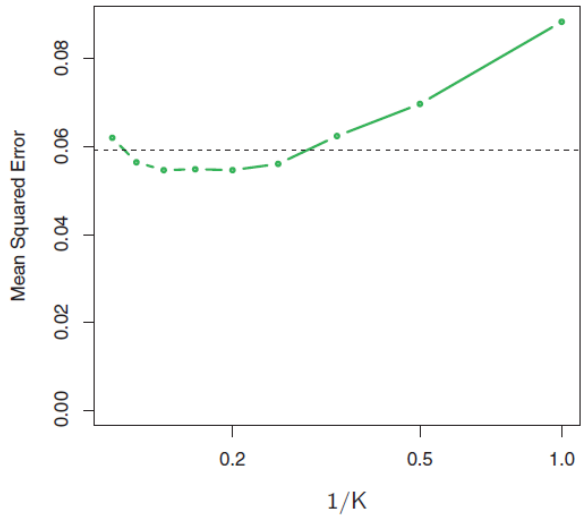
KNN regression

Linear regression

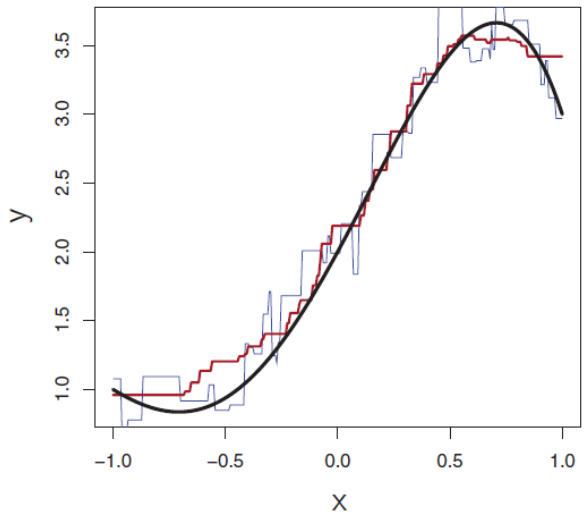
True Nonlinear Relation, One Dimension (p=1)

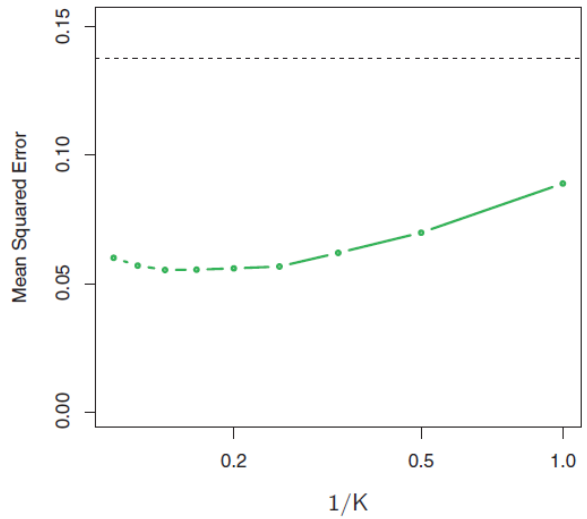
- Slightly non-linear relationship





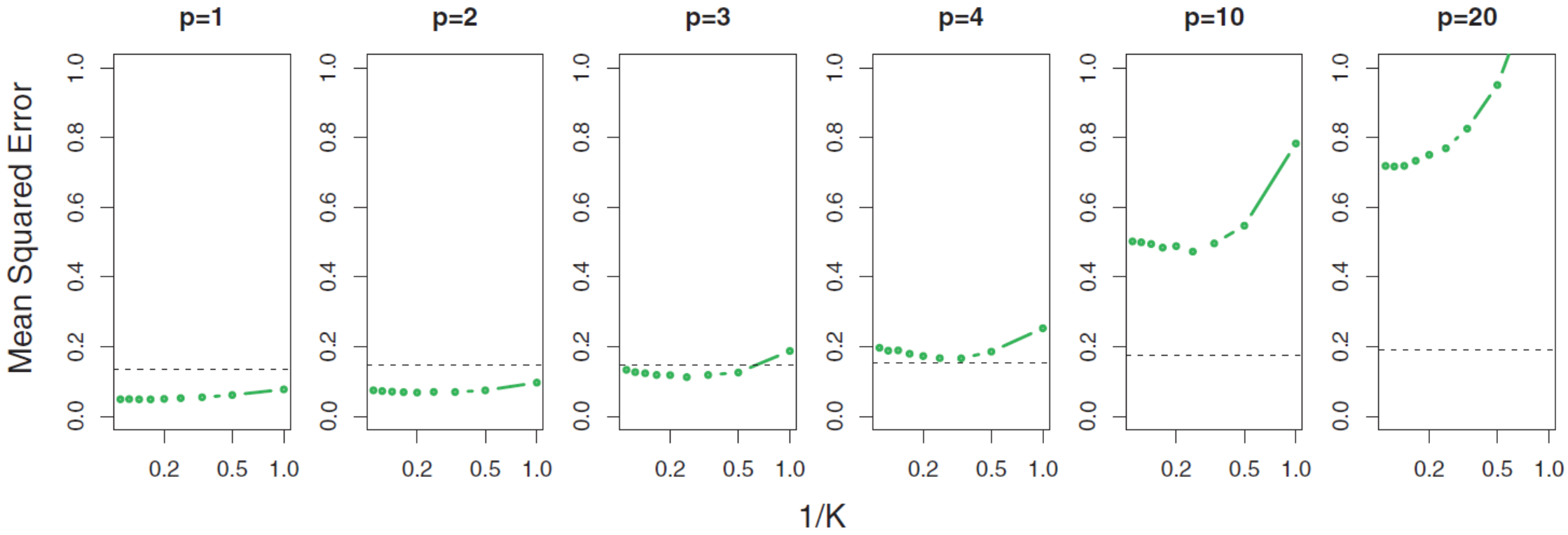
- Strongly non-linear relationship





True Nonlinear Relation, Different Dimensions

- KNN suffers from the curse of dimensionality.



Review: Learning Objectives

- Understand linear regression coefficient estimation and the ways of assessing the accuracy of coefficient estimates and the accuracy of the model.
- Understand methods dealing with qualitative predictors in linear regression.
- Understand interaction terms in linear regression.
- Understand non-linear relationship fit using polynomial regression.
- Understand potential problems of linear regression.
- Understand the comparison between linear regression and KNN regression.
- Be able to use R to conduct linear regression analysis and use diagnostic plots to check potential issues in linear regression.

Q & A

IST 5535: Machine Learning Algorithms and Applications

Langtao Chen, Spring 2021

4. Classification

Reading

- Book Chapter 4

Learning Objectives

- Understand logistic regression, linear discriminant analysis, and quadratic discriminant analysis.
- Understand performance measures including sensitivity, specificity, false positive rate, false negative rate, and AUC.
- Understand the impact of prediction threshold on performance measures.
- Be able to compare logistic regression, linear discriminant analysis, quadratic discriminant analysis, and KNN.
- Be able to use R to conduct logistic regression, linear discriminant analysis, and quadratic discriminant analysis.

Agenda

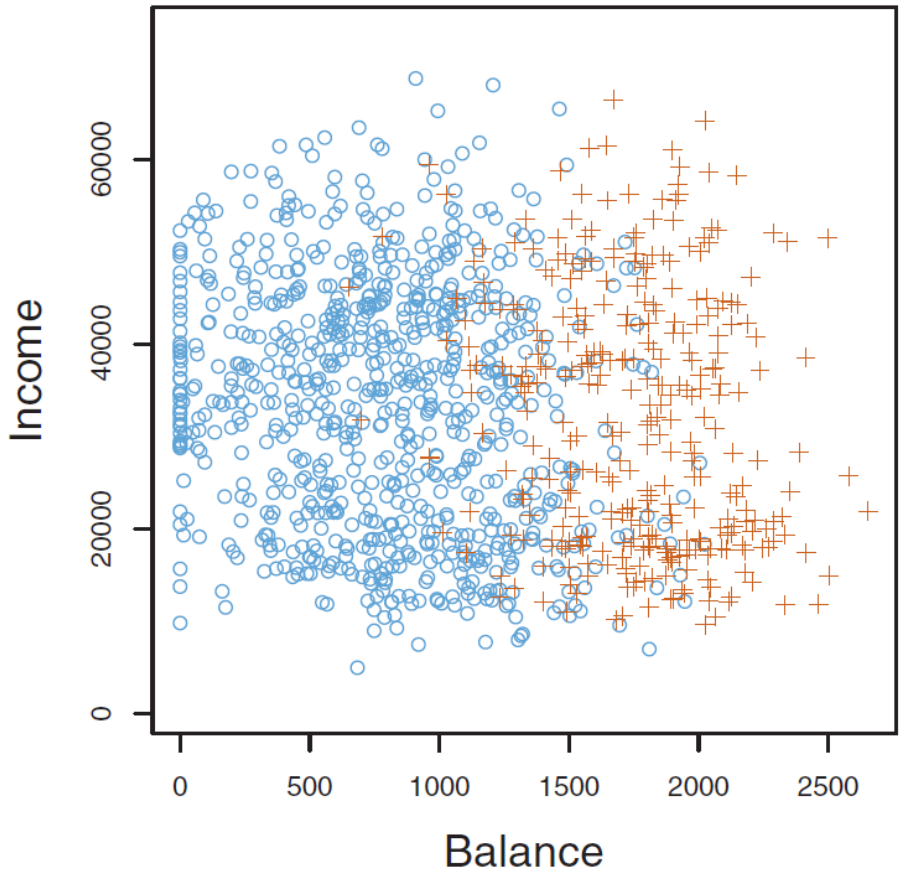
- Logistic Regression
- Linear Discriminant Analysis (LDA)
- More Performance Measures
- Quadratic Discriminant Analysis (QDA)
- A Comparison of Classification Methods

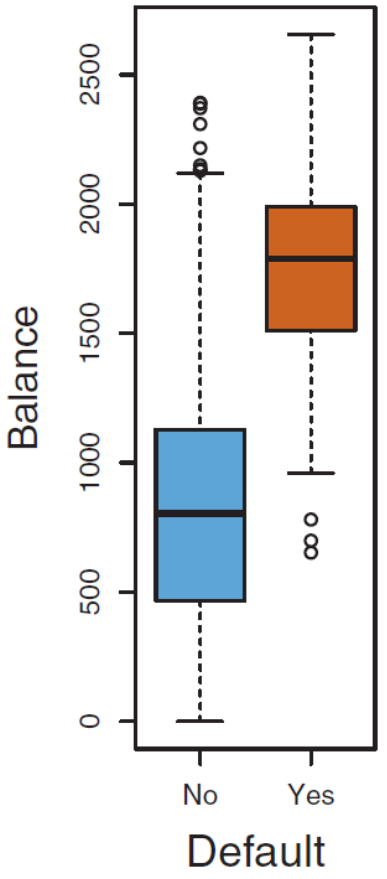
Classification

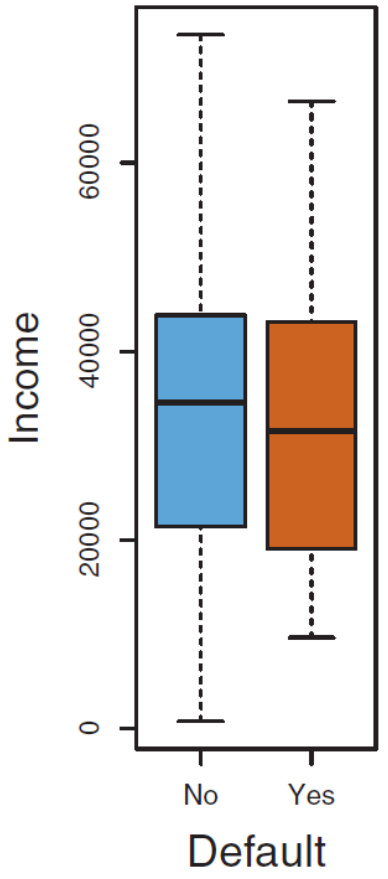
- In many cases, the response variable is qualitative or categorical:
  - Will the account holder pay off or default on the loan? $\text{default} \in \{\text{yes}, \text{no}\}$
  - Is the email spam or ham? $\text{email} \in \{\text{spam}, \text{ham}\}$
  - Is this bank transaction true or fraudulent? $\text{transaction} \in \{\text{true}, \text{fraudulent}\}$
  - ……
- Usually, we are interested in estimating the probabilities of Y belonging to each class.
- In this section, we'll discuss three classifiers:
  - Logistic regression
  - Linear discriminant analysis
  - Quadratic discriminant analysis

Default Credit Card Payment

- Predict whether a customer will default on his or her credit card payment, on the basis of annual income and monthly credit card balance.







Can We Use Linear Regression?

- For a qualitative response variable with more than 2 levels, there is no way to code this qualitative variable as a continuous variable.

- We may consider coding the response as follows:

$$y = \begin{cases} 1 & \text{if stroke;} \\ 2 & \text{if drug overdose;} \\ 3 & \text{if epileptic seizure.} \end{cases}$$

Ordinal

Nominal

Can We Use Linear Regression?

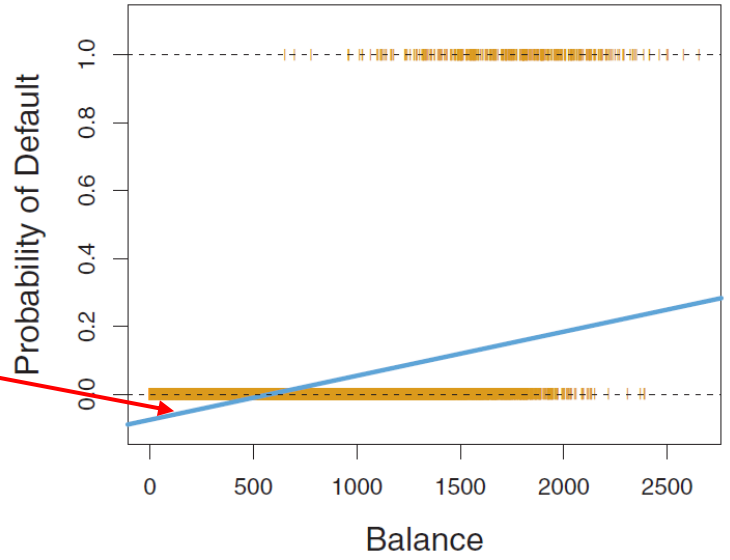
- For a qualitative response variable with 2 levels, linear regression could be used:

$$y = \begin{cases} 1, \text{if Yes;} \\ 0, \text{if No.} \end{cases}$$

=> Linear Probability Model

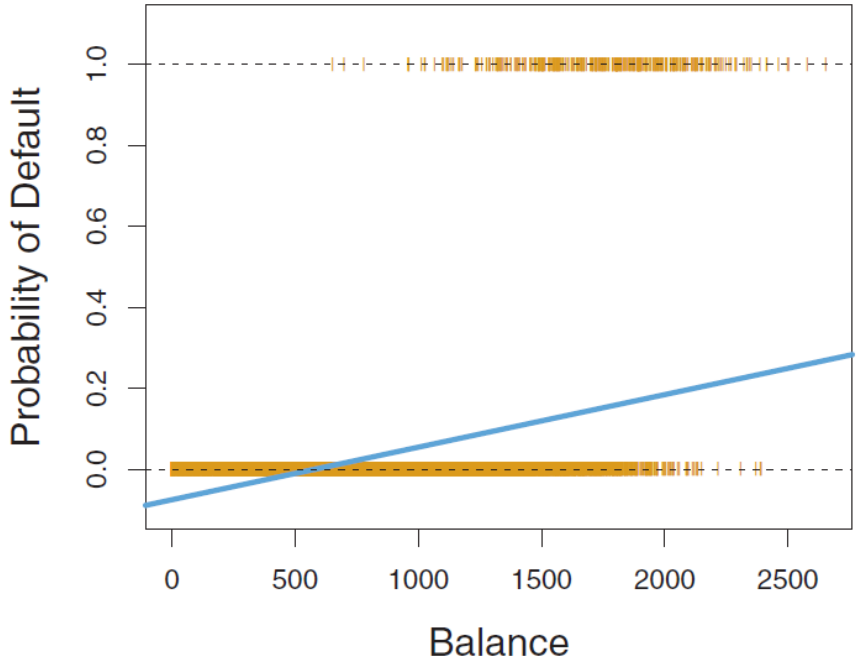
- We simply perform a linear regression of y on X and classify as Yes if $\hat{y} > 0.5$
- However, the predicted values can be outside the [0, 1] interval, making them hard to interpret.

When balance < 500, Pr(Default) is negative.



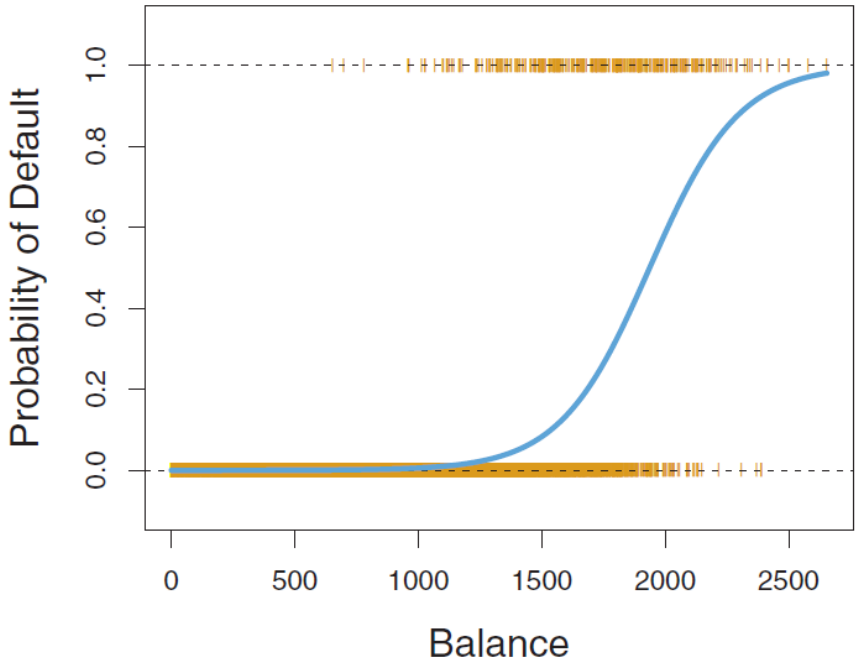
Can We Use Linear Regression?

- In such case of binary response, logistic regression is preferred than linear regression



Linear Regression

$\text{Default} = \beta_0 + \beta_0 * \text{Balance}$



Logistic Regression

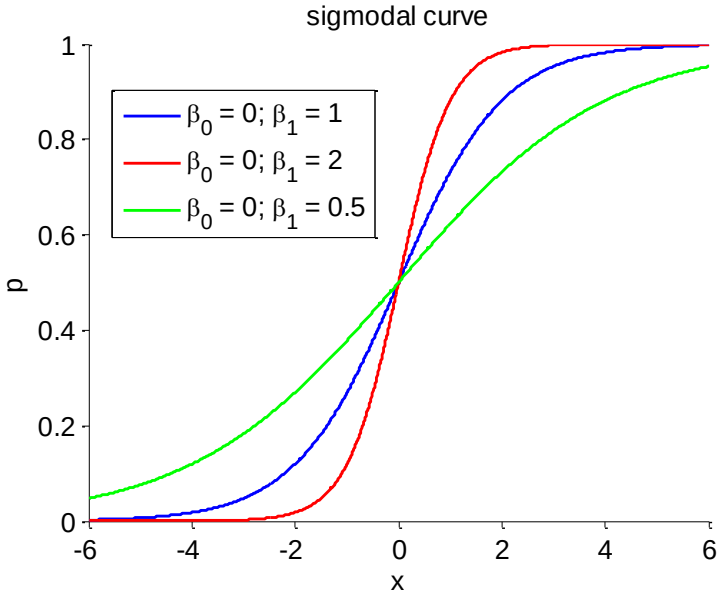
$Logit\ (\text{Default} = \text{Yes}) = \beta_0 + \beta_0 * \text{Balance}$

Fitting a Probability

- Logistic regression model maps a linear combination $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_n x_n$ from $(-\infty, +\infty)$ to [0,1] by using a probability function

$$p(y|X) = \frac{\exp(X\beta)}{1+\exp(X\beta)} = \frac{exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_n x_n)}{1+exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_n x_n)}$$

- We can fit the distribution of y with a Logistic Curve



- The intercept basically just 'scale' the input variable
- Large regression coefficient => risk factor strongly influences the probability

Transform Logistic to Linear Model

$$P(y|X) = \frac{e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_n x_n)}}{1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_n x_n)}}$$

- Step 1: Specify a probability as odds
  - $odds = \frac{P(y|X)}{1 - P(y|X)} = e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_n x_n)}$
- Step 2: Calculate the logit function
  - $\text{Logit} = \ln(odds) = \ln\left(\frac{P(y|X)}{1 - P(y|X)}\right)$

$$= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_n x_n$$

Implement Logistic Regression in R

- Use glm() function to fit generalized linear models;
- Need to specify the family as the binomial distribution, or else it would be a linear regression model;
- Then call summary() function to report the logistic regression results;
- An alternative approach is to use the stargazer package to report the result.

```
model <- glm(default ~ balance + income + student,
             family=binomial(link='logit'),data = Default)
summary(model)
```

Interpreting Logistic Regression Result

```
Logistic Regression
=========================================
                      Dependent variable:
                     ---------------------
                           default
-----------------------------------------
balance                     0.0057***
                            (0.0002)

income                      0.000003
                            (0.00001)

studentYes                  -0.6468**
                            (0.2363)

Constant                    -10.8690***
                            (0.4923)

-----------------------------------------
Observations                  10,000
Log Likelihood               -785.7724
Akaike Inf. Crit.          1,579.5450
=========================================
Note:    *p<0.05; **p<0.01; ***p<0.001
```

- Balance has a positive and significant effect on default (p-value < 0.001). A unit increase in balance increases the log odds by 0.0057 after controlling for other factors.
- Income does not have a statistically significant on default.
- Being a student has a negative and significant effect on default (p-value < 0.01), keeping all other factors constant. Being student reduces the log odds by 0.6468 after controlling for other factors.

Confounding

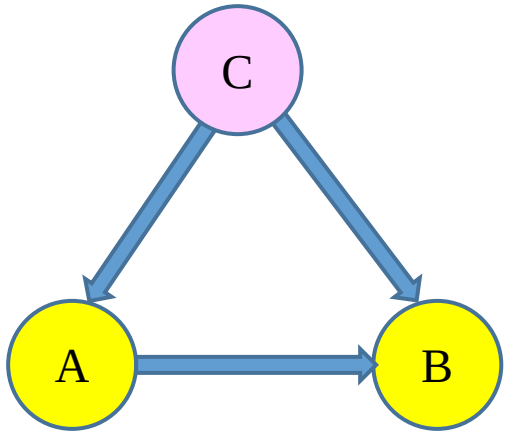
- Confounding due to high correlation between student and balance. Balance is a confounder or confounding variable in this case.

```
Logistic Regression
================================================
                        Dependent variable:
                    ----------------------------
                              default
                          (1)             (2)
------------------------------------------------
balance                                0.0057***
                                        (0.0002)

income                                  0.000003
                                       (0.00001)

studentYes                0.4049***     -0.6468**
                          (0.1150)       (0.2363)

Constant                 -3.5041***    -10.8690***
                          (0.0707)       (0.4923)

------------------------------------------------
Observations               10,000         10,000
Log Likelihood           -1,454.3410     -785.7724
Akaike Inf. Crit.        2,912.6830     1,579.5450
================================================
Note:                *p<0.05; **p<0.01; ***p<0.001
```

- If only student is included, the effect is positive and significant;
- If all predictors are included, the effect of student is negative and significant.

Confounding

- As ice cream sales increase, the rate of drowning deaths increases sharply. Therefore, ice cream consumption causes drowning.



Third Factor C Causes both A and B

C is called a confounder or confounding variable

Confounding

- Confounders can distort the relationship between the predictor and the response.
- Dealing with confounding:
  - Experimental design: random assignment, within subject design
  - Observational design: statistical control

Agenda

- Logistic Regression
- Linear Discriminant Analysis (LDA)
- More Performance Measures
- Quadratic Discriminant Analysis (QDA)
- A Comparison of Classification Methods

When we have more than 2 response classes

- The regular logistic regression model can only deal with a binary response;
- Logistic regression can be extended to handle response variables with more than 2 classes;
- In practice, we often use linear discriminant analysis (LDA) for multi-class classification.

Why Not Logistic Regression?

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. LDA does not suffer from this problem.

- LDA is more stable than logistic regression when:
  - n is small, and
  - the distribution of the predictors X is approximately normal in each of the classes.

- LDA is popular when we have more than two response classes.

Using Bayes Theorem for Classification

- The famous Bayes theorem:

$$Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

- Re-write it as:

$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

  - $f_k(x) = \Pr(X = x|Y = k)$: *density* for X in class k
  - $\pi_k = \Pr(Y = k)$: marginal or *prior* probability for class k

Bayes Classifier

- Bayes Classifier is the gold standard, but unattainable since the density is unknown.

$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- Recap: Does a Bayes classifier lead to perfect prediction (zero error rate)?

- However, if we can find a way to estimate the density, then we can develop a classifier that approximates the Bayes classifier.

Linear Discriminant Analysis when p =1

- Assume normal/Gaussian density

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-u_k}{\sigma_k}\right)^2}$$

- Further assume variances are the same, i.e., $\sigma_k = \sigma$
- Then, we get

$$p_k(x) = Pr(Y = k|X = x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-u_k}{\sigma}\right)^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-u_l}{\sigma}\right)^2}}$$

Discriminant Functions

- Classification is to find the class k when X=x for which $p_k(x) = Pr(Y = k|X = x)$ is the largest.

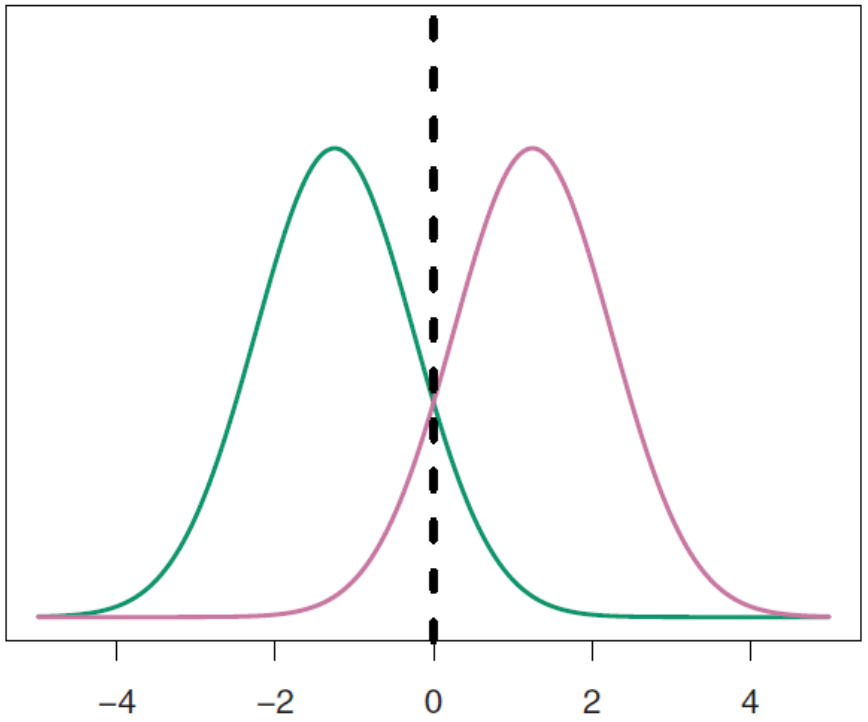
- After log-transform the previous formula and discard terms not depending on k, this is equivalent to assigning k to the class with the largest discriminant score:

$$\delta_k(x) = x \cdot \frac{u_k}{\sigma^2} - \frac{u_k^2}{2\sigma^2} + \log(\pi_k)$$

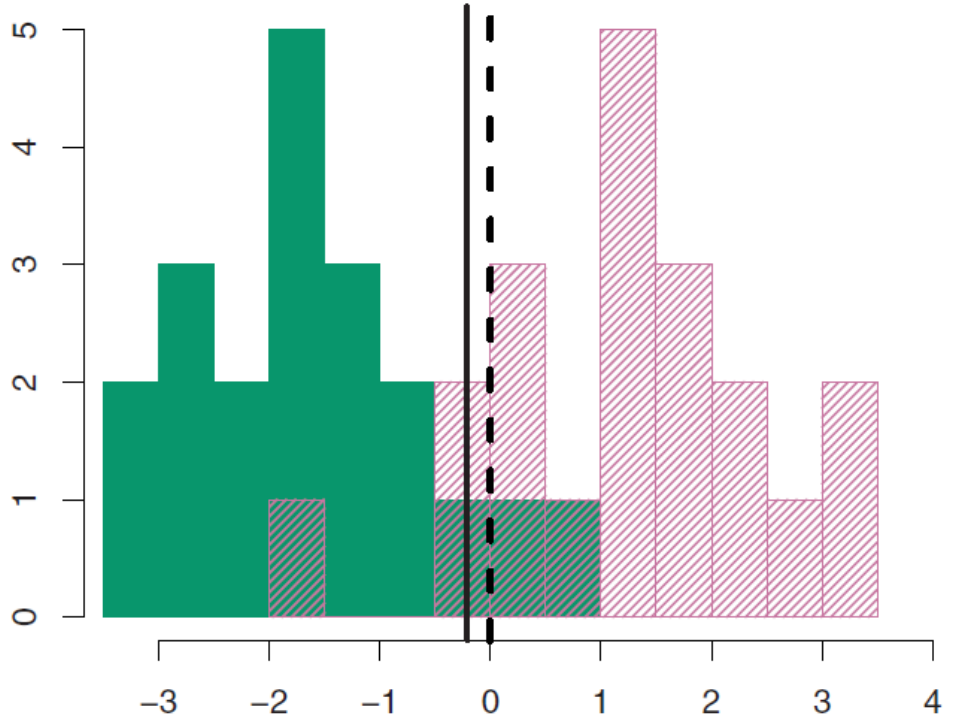
The word "linear" in LDA stems from the fact that the discriminant function is a linear function of x.

LDA Example

- Two classes K=2, u_1=-1.25, u_2=1.25, $\pi_1=\pi_2$=0.5, σ^2=1



Dashed vertical line: Bayes decision boundary



Solid vertical line: LDA decision boundary estimated from training data

Estimating the Parameters

- In practice, we don't know the parameters in normal distribution.
- LDA approximates the Bayes classifier by simply estimating the parameters and plugging them into the discriminant function.

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

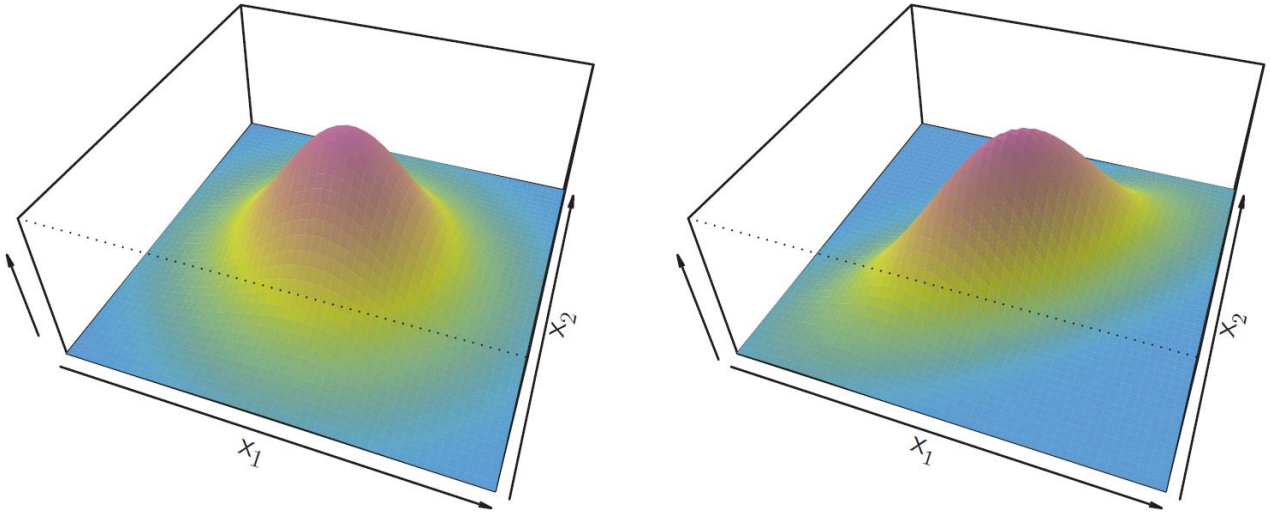
$$\hat{\pi}_k = n_k/n$$

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

Linear Discriminant Analysis when p >1

- When X contains multiple predictors, the similar approach is applied by using a multivariate density function.

Examples: Two multivariate Gaussian density functions



X_1 and X_2 are uncorrelated

Corr(X_1, X_2) = 0.7

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \mathbf{\Sigma}^{-1} \mu_k + \log \pi_k$$

A General Process of LDA

- Extract discriminant functions
  - Number of LD = min(number of predictors, number of classes -1)
  - $LD_m = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$
- Use linear discriminants to classify response

Summary of LDA Procedure

- Assume that observations from each class are drawn from a Gaussian distribution;
- Estimate parameters (means, variance/covariance) in the Gaussian distribution density from the data;
- Plug parameter estimates into Bayes theorem to calculate $p_k(x)$;
- Assign the class that has the largest probability.

LDA on the Default Data

- A confusion matrix comparing LDA predictions to true statuses

		True default status		
		No	Yes	Total
Predicted default status	No	9,645	254	9,899
	Yes	22	79	101
	Total	9,667	333	10,000

- The overall prediction accuracy seems good

$$Accuracy = \frac{9{,}645 + 79}{10{,}000} = 97.24\%$$

- A null classifier (always classifying the response as the majority class) yields to 96.7% accuracy!

- However, if our purpose is trying to identify high-risk customer, this model performs not that well:
  - This model only detect 79 out of 333 true default customers;
  - 254 customers who default are incorrectly predicted by the model as no default customers.

Agenda

- Logistic Regression
- Linear Discriminant Analysis (LDA)
- More Performance Measures
- Quadratic Discriminant Analysis (QDA)
- A Comparison of Classification Methods

Sensitivity and Specificity

- *Sensitivity* (*true positive rate*, *recall*, or *hit rate*): the percentage of true positive observations that are correctly identified.

$$Sensitivity = \frac{79}{333} = 23.7\%$$

- *Specificity* (or *true negative rate*): the percentage of true negative observations that are correctly identified.

$$Specificity = \frac{9645}{9667} = 99.8\%$$

		True default status		
		No	Yes	Total
Predicted default status	No	9,645	254	9,899
	Yes	22	79	101
	Total	9,667	333	10,000

Why does LDA have such a low sensitivity?

False Positive Rate and False Negative Rate

- False positive rate: The fraction of negative observations that are classified as positive.

$$False\ positive\ rate = \frac{22}{9667} = 0.2\%$$

- False negative rate: The fraction of positive observations that are classified as negative.

$$False\ negative\ rate = \frac{254}{333} = 76.3\%$$

		True default status		
		No	Yes	Total
Predicted default status	No	9,645	254	9,899
	Yes	22	79	101
	Total	9,667	333	10,000

$False\ positive\ rate = 1 - specificity$

$False\ negative\ rate = 1 - sensitivity$

The Problem of Imbalanced Dataset

- A dataset is unbalanced when it has uneven class distribution.
  - In a customer loan dataset, only 5 out of 100 customers have bad credit.
  - Suppose we need to train a classifier to classify whether a customer has good credit.
- The accuracy paradox
  - A "dumb" algorithm (null classifier) is to always predict the majority class for new data;
  - Such an algorithm does not learn the underlying patterns from the data, but it "really" performs very good: accuracy = 95%.

Pred. Class \ True Class	Yes	No
Yes	95	5
No	0	0

$$Accuracy = \frac{95+0}{95+5+0+0} = 95\% \qquad Sensitivity = \frac{95}{95+0} = 100\%$$

However, $Specificity = \frac{0}{0+5} = 0\%$

This algorithm fails to detect risky customers.

The problem: machine learning algorithms tends to bias towards the majority class.

Deal with Imbalanced Dataset

- Plot the confusion matrix, understand problems in your analysis.
- Use the right performance metrics
  - Not rely on accuracy, choose other metrics such as balanced accuracy, recall, specificity, AUC, precision, f1, etc.

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2}$$

- Resample the training dataset
  - Over-sample the minority class
  - Under-sample the majority class
- Use different threshold for prediction
- Customize the cost function to assign larger penalty to the misclassified minority class

Use 0.2 as Threshold for Prediction

- $Pr(default = yes | X = x) > threshold$

Threshold = 0.5

		True default status		
		No	Yes	Total
Predicted default status	No	9,645	254	9,899
	Yes	22	79	104
	Total	9,667	333	10,000

$$False\ positive\ rate = \frac{22}{9667} = 0.2\%$$

$$False\ negative\ rate = \frac{254}{333} = 76.3\%$$

Threshold = 0.2

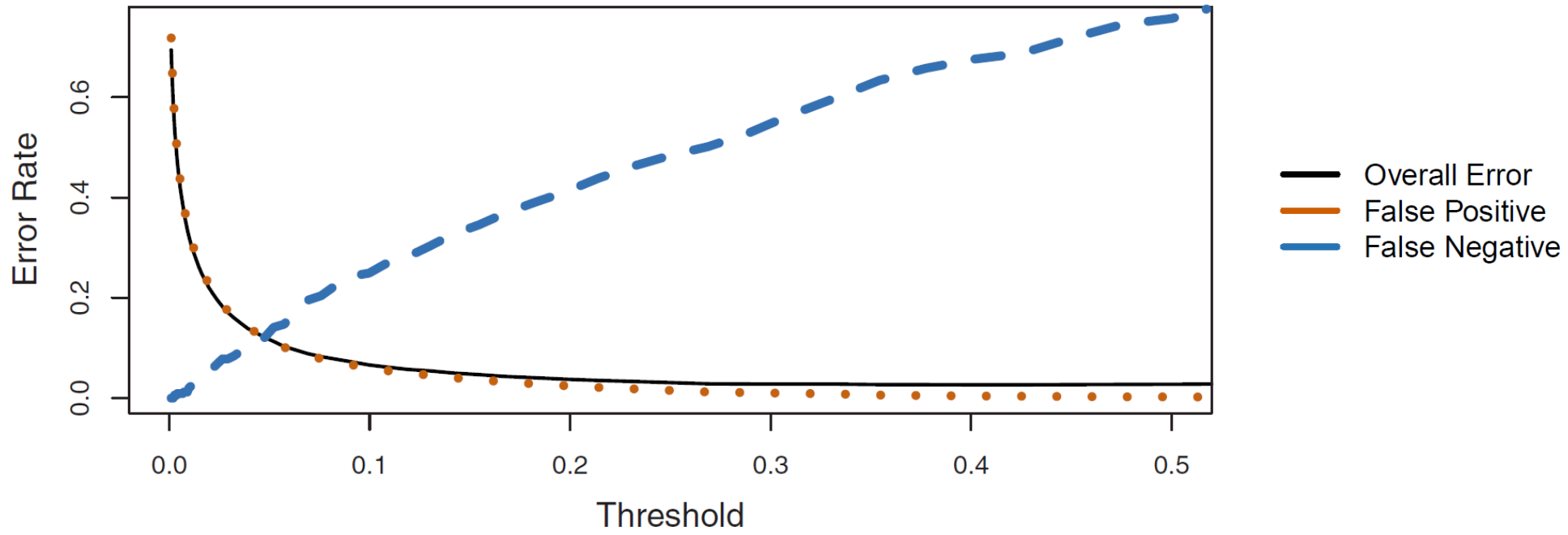
		True default status		
		No	Yes	Total
Predicted default status	No	9,435	140	9,570
	Yes	232	193	430
	Total	9,667	333	10,000

$$False\ positive\ rate = \frac{232}{9667} = 2.4\%$$ ↗

$$False\ negative\ rate = \frac{140}{333} = 42.0\%$$ ↘

Error Rates as a Function of Threshold

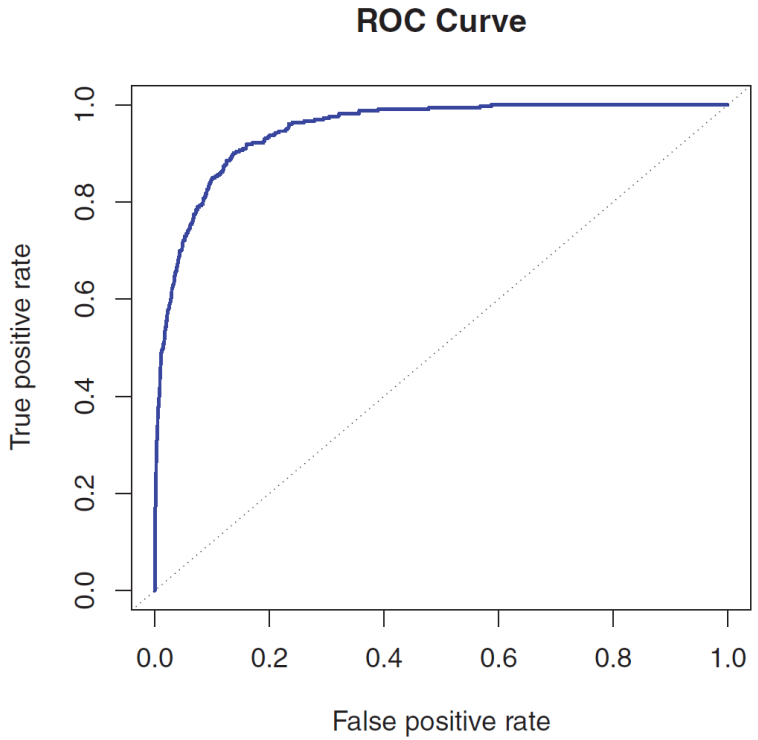
- We can change the two error rates using different thresholds in [0, 1].
- The best threshold should be based on domain knowledge.



In order to further reduce the false negative rate (or increase sensitivity), we may want to reduce the threshold to 0.1 or less.

ROC (receiver operating characteristics) Curve

- ROC plot displays both false positive rate and true positive rate simultaneously with varying thresholds.
- We can use the *AUC* (area under the curve) to summarize the overall performance.
- Good classifier has large area under curve (AUC).



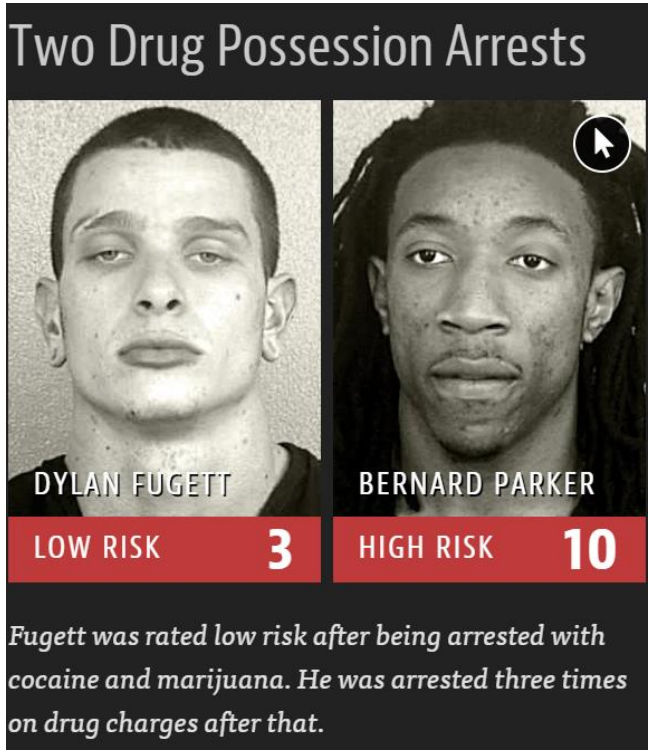
General guide

.90-1 = excellent (A)
.80-.90 = good (B)
.70-.80 = fair (C)
.60-.70 = poor (D)
.50-.60 = fail (F)

More Fundamental Bias in Machine Learning

- Garbage in, garbage out
  - Biased data => biased machine learning models
- *Pro Publica* found machine learning algorithms falsely flagged black defendants as future criminals, wrongly labeling them at almost twice the rate of white defendants.

DYLAN FUGETT
Prior Offense:
1 attempted burglary
Subsequent Offenses:
3 drug possessions



BERNARD PARKER
Prior Offense:
1 resisting arrest without violence
Subsequent Offenses:
None

Source: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Further Reading

- Chouldechova, A., and Roth, A. 2020. "A Snapshot of the Frontiers of Fairness in Machine Learning," *Communications of the ACM* (63:5), pp. 82–89.

Agenda

- Logistic Regression
- Linear Discriminant Analysis (LDA)
- More Performance Measures
- Quadratic Discriminant Analysis (QDA)
- A Comparison of Classification Methods

Quadratic Discriminant Analysis (QDA)

- Issues of LDA:
  - LDA assumes the same variance/covariance for each class;
  - LDA may perform poorly due to this strong assumption.

- QDA assumes each class has its own covariance matrix. Then the discriminant function would be:

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log\pi_k \\ &= -\frac{1}{2}x^T\mathbf{\Sigma}_k^{-1}x + x^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log\pi_k\end{aligned}$$

"Quadratic" in QDA: the discriminant function is a quadratic function of x.

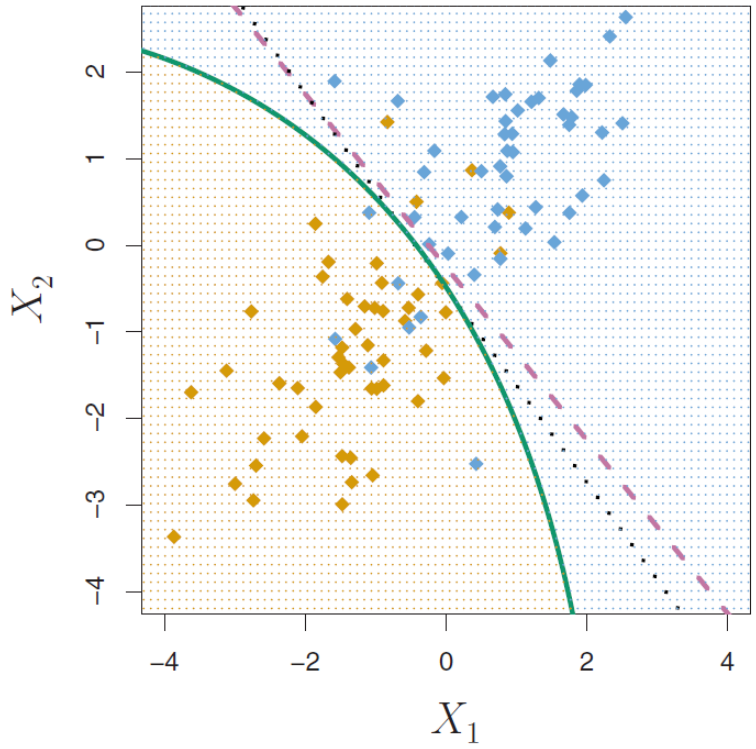
Which One to Choose? LDA or QDA?

- The answer lies in the bias-variance trade-off
  - QDA allows a separate covariance matrix for each class, thus QDA is more flexible than LDA.
  - QDA may reduce the bias, but its variance might be higher.

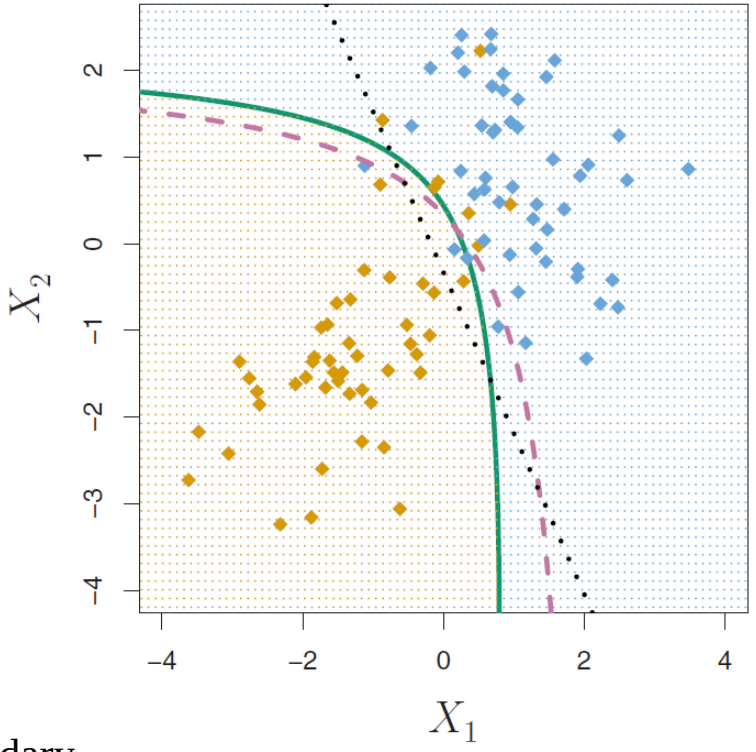
- In general,
  - LDA tends to be better if there are relatively few training observations and so reducing variance is crucial.
  - QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable.

LDA vs. QDA

Covariances of the X are equal across classes
Bayes decision boundary is linear
LDA is a better approximate



Covariances of the X are not equal across classes
Bayes decision boundary is quadratic
QDA is a better approximate



Black dotted: LDA boundary
Purple dashed: Bayes' boundary
Green solid: QDA boundary

Agenda

- Logistic Regression
- Linear Discriminant Analysis (LDA)
- More Performance Measures
- Quadratic Discriminant Analysis (QDA)
- A Comparison of Classification Methods

Logistic Regression vs. LDA

- For a two-class setting with p=1, the LDA discriminant function is

$$p_k(x) = Pr(Y = k|X = x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-u_k}{\sigma}\right)^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-u_l}{\sigma}\right)^2}}$$

- Then, we can get:

$$\log\left(\frac{p_1(x)}{1-p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1 x$$

- Thus, LDA has the same form as logistic regression.
- The difference is how the parameters are estimated.
- In practice the results are often very similar.

Logistic Regression vs. LDA

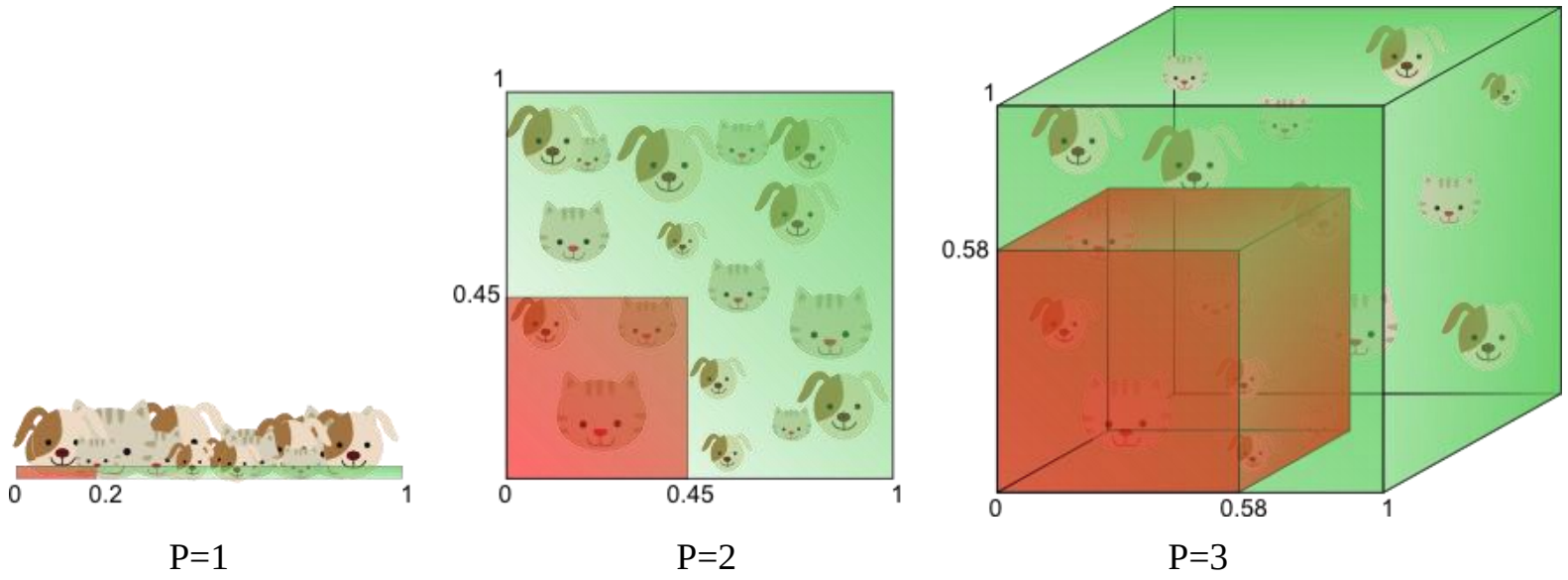
- LDA assumes that the observations are drawn from a Gaussian distribution with a common covariance matrix in each class, and so can provide some improvements over logistic regression when this assumption approximately holds.
- Conversely, logistic regression can outperform LDA if these Gaussian assumptions are not met.

KNN vs. LDA and Logit

- KNN takes a completely different approach.
- KNN is a completely non-parametric approach: no assumptions are made about the shape of the decision boundary.
- Therefore, we can expect this approach to dominate LDA and logistic regression when the decision boundary is highly non-linear.
- However, KNN does not tell us which predictors are important; we don’t get a table of coefficients.

KNN Suffers from the Curse of Dimensionality

- As variables are added, the data space becomes increasingly sparse.
- Prediction and classification models fail due to insufficient data for a useful model across so many variables.



P=1 P=2 P=3

The amount of training data needed to cover 20% of the feature range grows exponentially with the number of dimensions.

Image: http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

QDA vs. KNN, LDA, and Logit

- QDA serves as a compromise between the non-parametric KNN method and the linear LDA and logistic regression approaches.
- Since QDA assumes a quadratic decision boundary, it can accurately model a wider range of problems than can the linear methods.
- No one single method dominates in all situations:
  - True decision boundary is linear: LDA and Logit perform well;
  - True decision boundary is moderately non-linear: QDA performs better;
  - True decision boundary is more complicated: non-parametric KNN is superior.

Q & A