CS 438 Big Data Analytics
Spring 2023
Course Handouts
Dr. Tammy VanDeGrift

Name: _____

If found, call/email:_____

HANDOUTS – Please bring this booklet to all class sessions

For access to latest calendar and syllabus: see Moodle (learning.up.edu)

Lab Instructions

1. **Software**: See Moodle for any software installations you need to make prior to the start of the lab.

2. **Prelabs**: Prelabs are to be completed individually and are due at the start of the lab day(s) designated for the lab. Prelabs are designed to ensure you have the background knowledge of concepts from class, the textbook, and other resources.

3**. Lab work sessions**: You are expected to attend the class sessions for lab days. You will work with other students during the lab days to complete lab checkpoints. In some cases, you will work in groups of three. The lab pairings will be posted during each lab session and may be adjusted due to absences.

4. **Lab communication**: You are expected to work together on labs during class; do not divide and conquer the checkpoints.

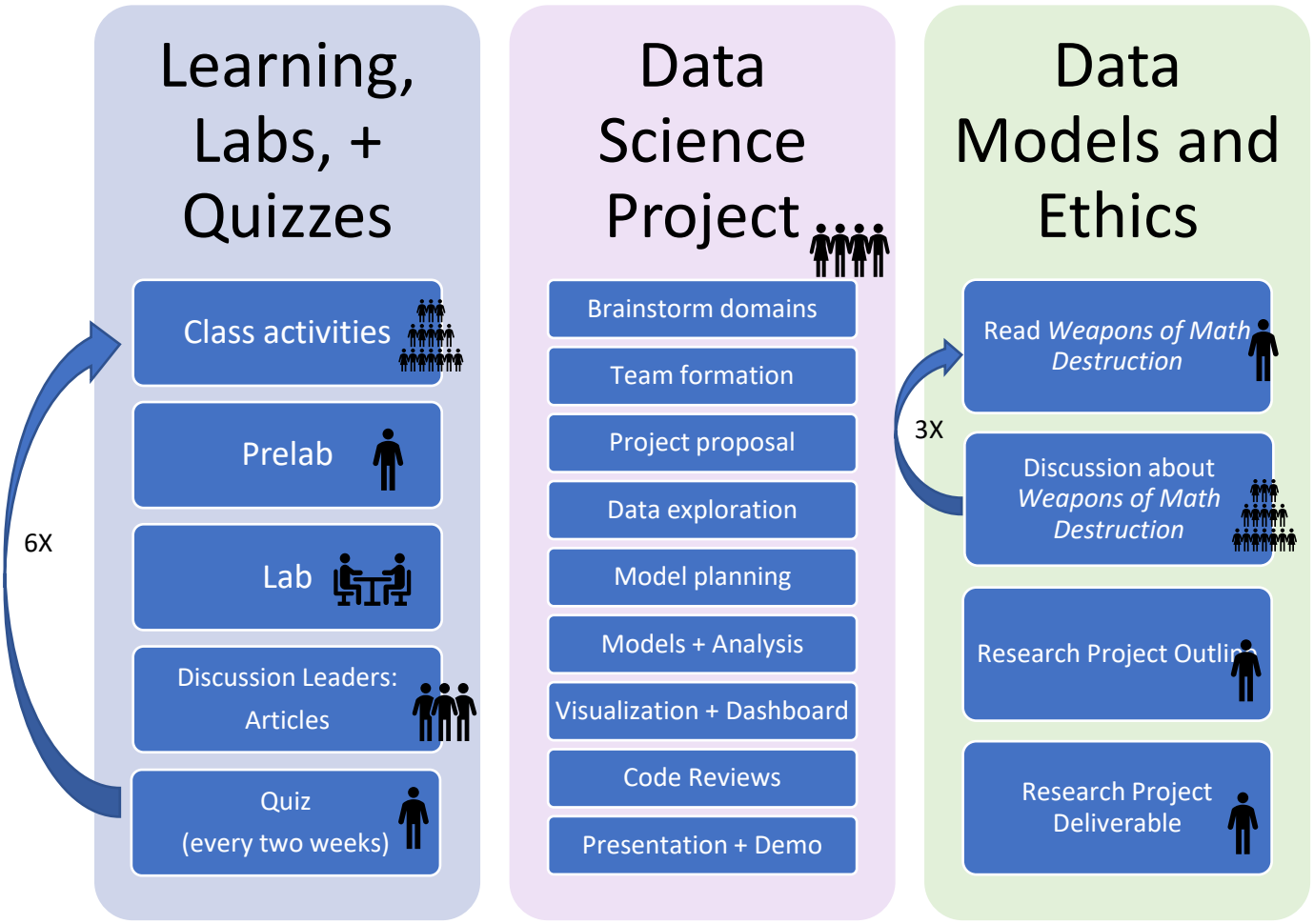
5. **Questions**: If you have a question during the lab, please ask Tammy.

6. **Checkpoints**: When you have completed a checkpoint, ask Tammy to review your work. Save all your work to your P: drive since you will submit a complete lab report and associated files for each lab.

7. **Unfinished checkpoints**: Submit any work for unfinished checkpoints to Moodle by the deadline. The expectation is that pairings will complete the work together. If it is challenging to get together to complete the lab together, be clear with your partner(s) if you will be completing the lab individually or as a pair. Be sure to indicate if unfinished checkpoints were done together or individually in your submitted work. Check Moodle for lab deadlines.

8. **Late days**: You have two free late days to submit prelabs and/or labs late. You may submit two items up to 24 hours late or submit one item up to 48 hours late. For partnered labs, all members will be "charged" late days for late work. However, if one partner has remaining late days and one partner does not, you may use the maximum late days of both partners.

Course Design for Learning



Activity 1: What is Data Science and Your Data Science Profile?

This course focuses on big data analytics. This could be an entire curriculum, so we will primarily focus on data exploration and data analysis techniques for modeling information, which is often referred to as data science. We will focus much more on the "data analytics" and "data science" rather than the "big" part of big data in this course. Below is one definition of data science by Drew Conway.

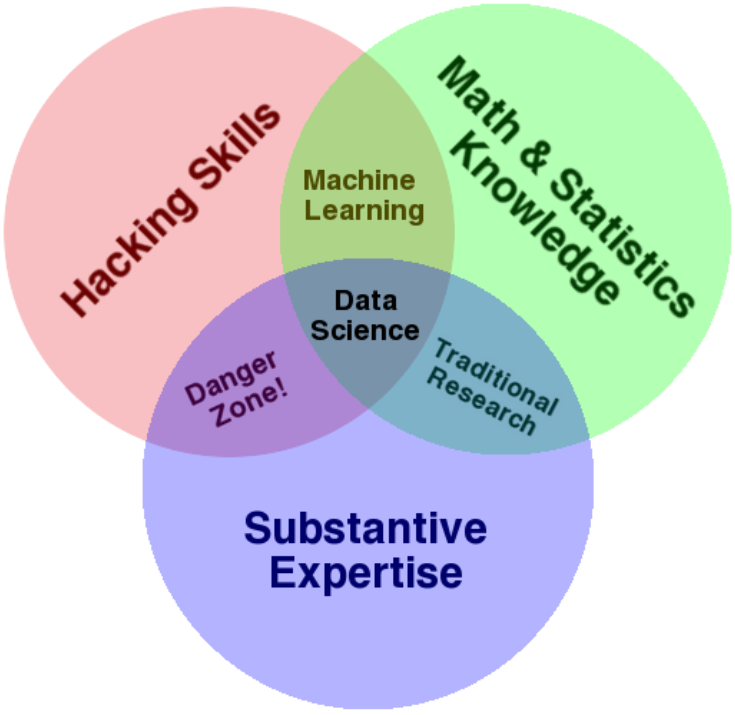


Figure: Venn Diagram of Data Science (by Drew Conway, Credit, Creative Commons: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram)

1. Which of these circles do you consider your strengths (expertise would be about the domain from which the data is collected, so let's use the domain of **economics** for this exercise)?

2. For one of the intersecting areas below, describe a project that could fall under that zone.

Machine Learning (do this one if your first name begins with A – H)

Traditional Research (do this one if your first name begins with I – P)

Danger Zone (do this one if your first name beings with R – Z)

There are many training pathways to a career in data science, but it is a field that requires the melding of many skills sets, such as computer science, mathematics, statistics, data visualization, machine learning, communication/presentation skills, and domain expertise.

Data Science is not "new" – just a resurgence due to the availability and ease of collecting and storing data. Statisticians have been doing data science for 50 years. We now have the computing capability for storing and analyzing lots of data.

Often, data scientists work within teams so the composition of the team utilizes these combined skillsets. See the next figure for how teams could form.

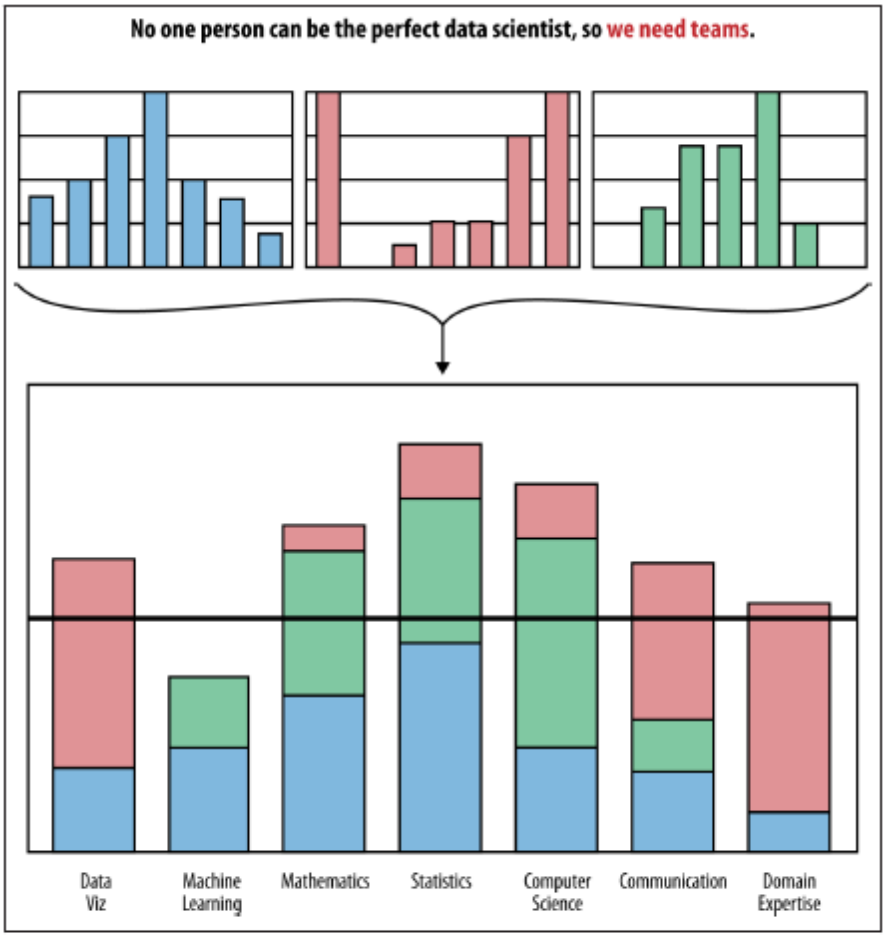


Figure 1-3. Data science team profiles can be constructed from data scientist profiles; there should be alignment between the data science team profile and the profile of the data problems they try to solve

Figure from *Doing Data Science* by Cathy O'Neil and Rachel Schutt

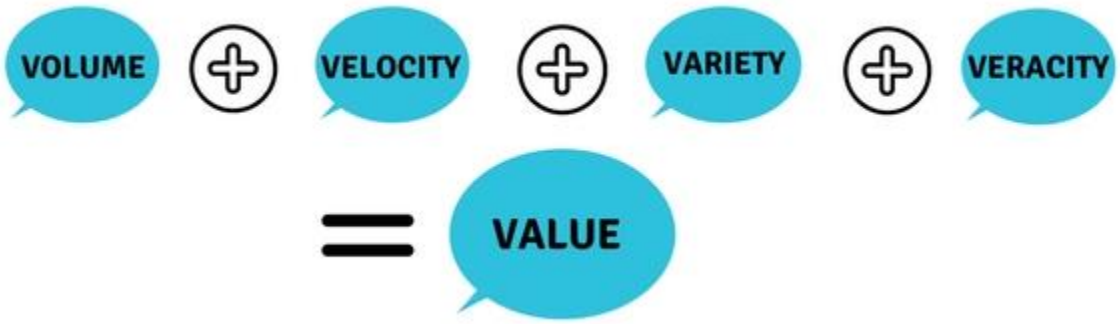
3. Draw your own bar chart about your skillsets below:

Data Viz Machine Learning Math Stats Computer Science Communication Domain Expertise Economics

Activity 2: Get to know the Vs of Big Data

Introduce yourself to other members of the group. After everyone has met, continue with the activity below.

"Big" Data is sometimes characterized by the 5 "V"s:



Look at the images on the next few pages or use google. Use them and any other searching that you want to do to summarize your understanding of the 5 Vs. Your group may be assigned one to start with and then research the others when you have completed a definition for your assigned "V".

Volume:

Velocity:

Variety:

Veracity:

Value:

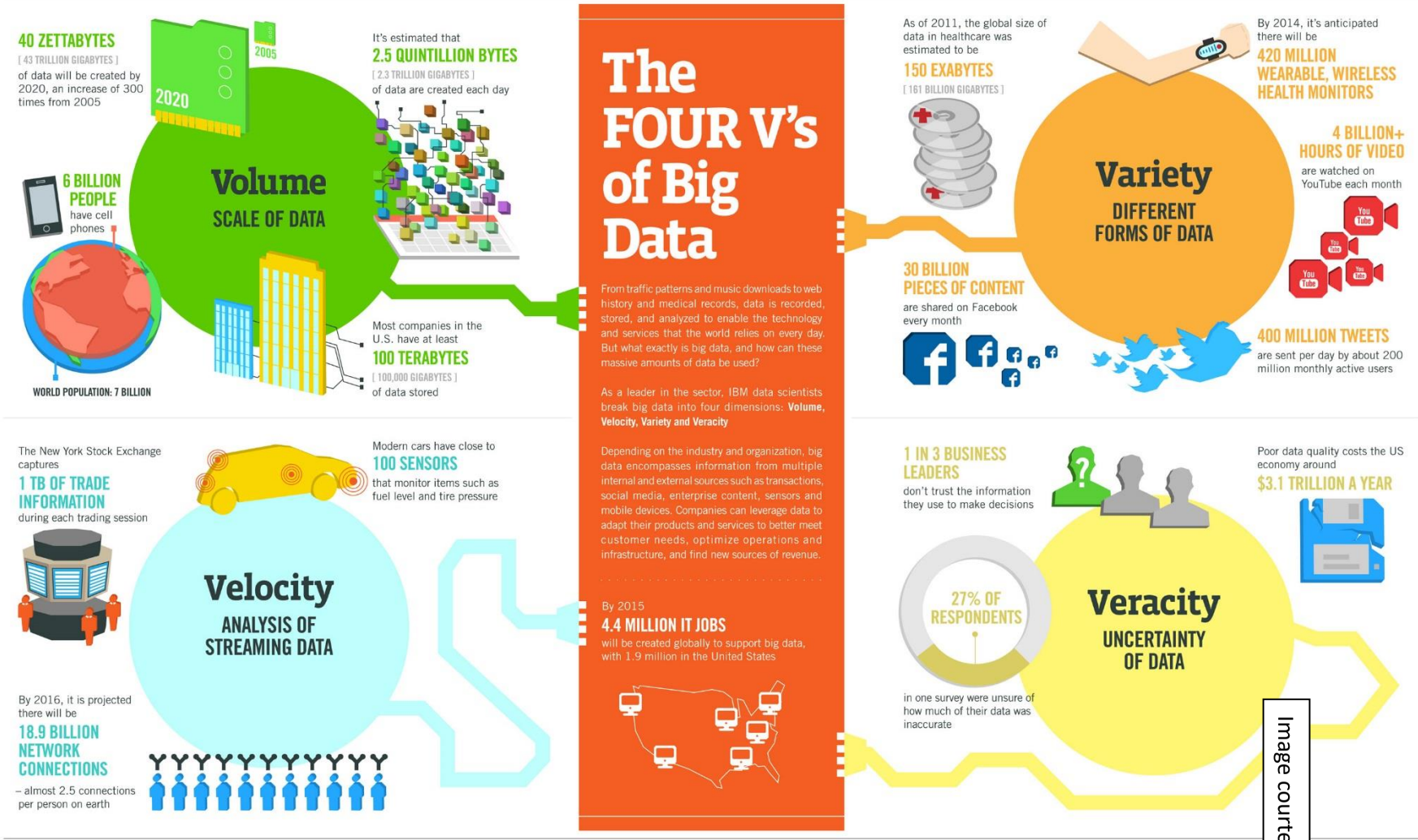
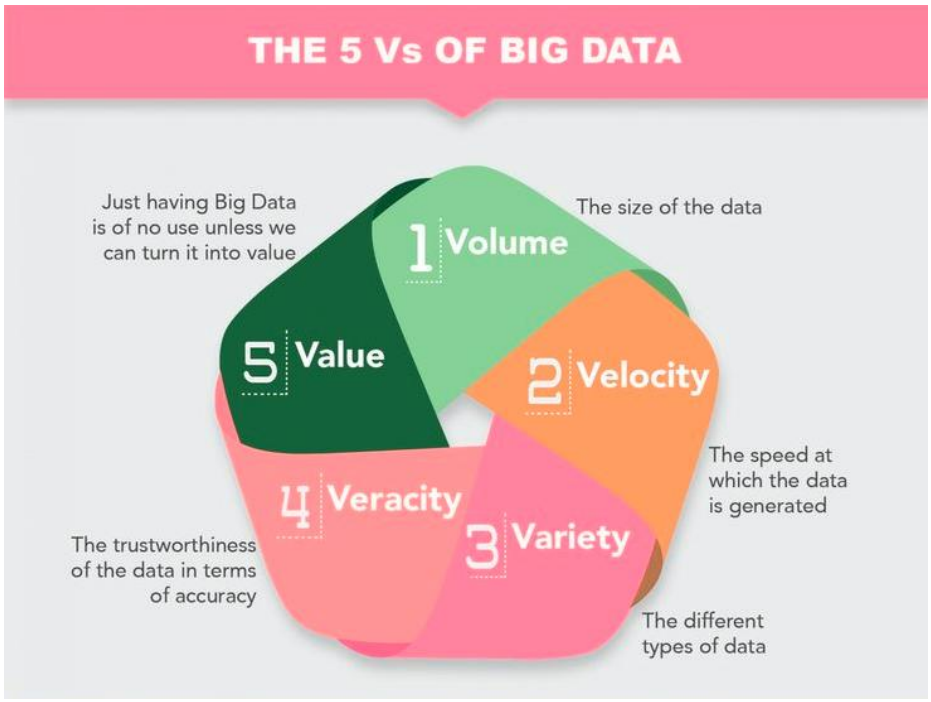
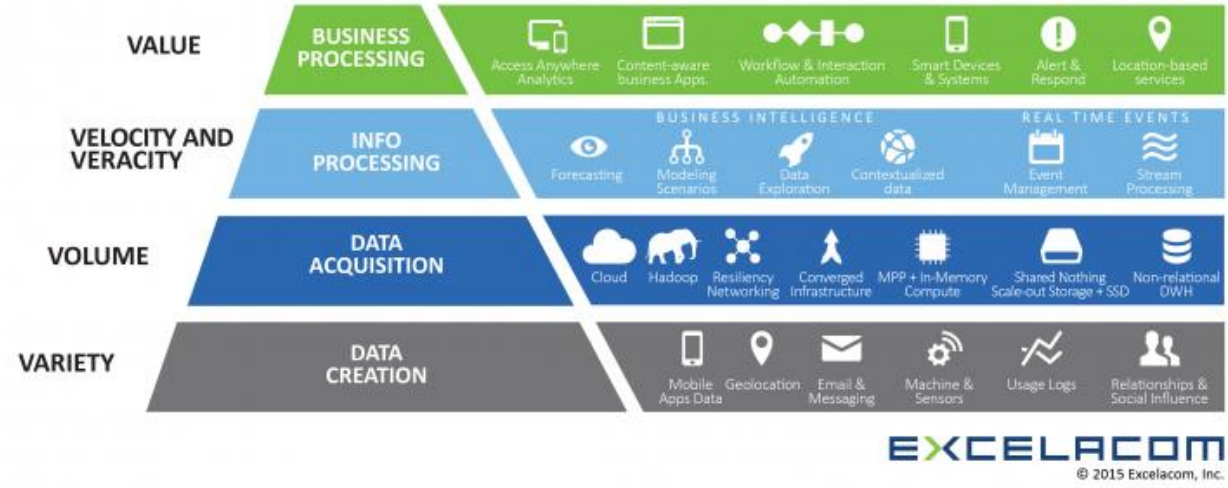


Image courtesy of IBM



Images courtesy of Excelacom



Activity 3: You and Your Expertise

1. What domains interest you other than computer science (examples: music, sports, games, chemistry, astronomy, economics, law, education, healthcare, etc.)?

2. Of your outside interests, select one to three domains in which you think you have expertise (more than just a casual observer / reader of a topic). Write those here and on post-it notes.

We will use this data in a class-wide introduction exercise to determine expertise areas for the data science project team formation.

CS 438: Topics to Review from EGR 361 or MTH 361

Data presentation:

- Histograms
- Box plots
- Time series plots
- Scatterplots

Descriptive statistics:

- Mean
- Median
- 5-number Summary and Quartiles
- Standard Deviation and Variance

Experiments:

- Population
- Sample
- Random sample
- Discrete variables
- Continuous variables

Probability distributions:

- Normal distribution
- Binomial distribution
- Poisson distribution
- Exponential distribution

Statistical inference:

- Single sample vs two sample
- Inference on mean, proportion, variance
- Analysis of variance (inference on means from more than two populations)
- Type I and Type II Errors
- Sample size
- Central Limit Theorem

Activity 4: Statistics Review

We will play a game to review concepts from EGR 361 or MTH 361. Use this space to make notes about the concepts that appear in the game.

Activity 5: Data and People

A. In your group, brainstorm some products/services that you use that only work because there is access to data.

Example: recommendation systems (Amazon thinks you would also like these books given your order/search history)

-
-
-
-
-
-

B. Choose one of the products listed above. Which did you choose? _____
Answer these questions:

1. How has the product changed your behavior?

2. How has your behavior changed the product?

3. What data does the product/service collect from you and others to make this work?

4. What do you value in terms of this product/service?

5. What does the provider/company value in terms of the data?

You may want to think of data and modeling as a continuous feedback loop.

Activity 6: Project Team Composition

Your project team is built around collective interest and expertise. In your data science project group, complete the following activities.

Rachel Schutt and Cathy O'Neil describe a set of skills (a profile) for a data scientist. Here is an example profile, similar to one from a previous activity:

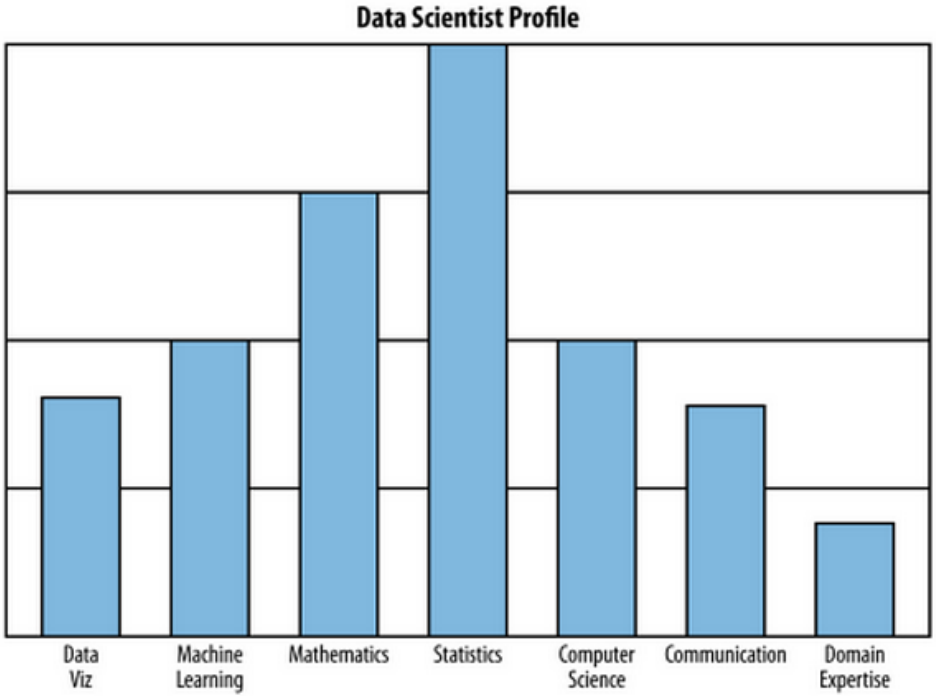
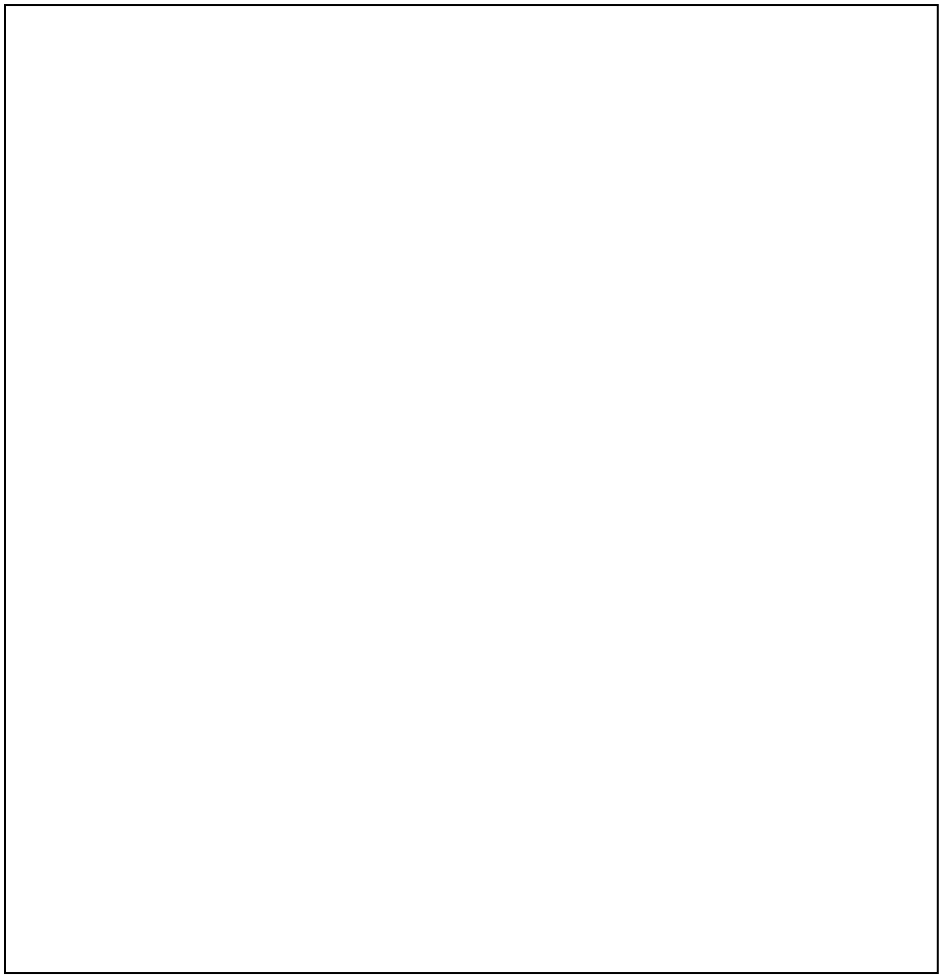


Figure: Data science Profile from *Doing Data Science* by O'Neill and Schutt

Copy your data science personal profile from the prior activity. Note that the Domain Expertise is now your project domain (write domain here): _____

With your group's individual data science profiles, put them together to make a stacked team profile. You may want to use different colors for each person, similar to the figure 1-3 in Activity 1.



Activity 7: Data Science Life Cycle and Project Planning

The book, *Data Science & Big Data Analytics* by EMC^2 presents a model of the data science life cycle. For this activity, work with your data science project team.

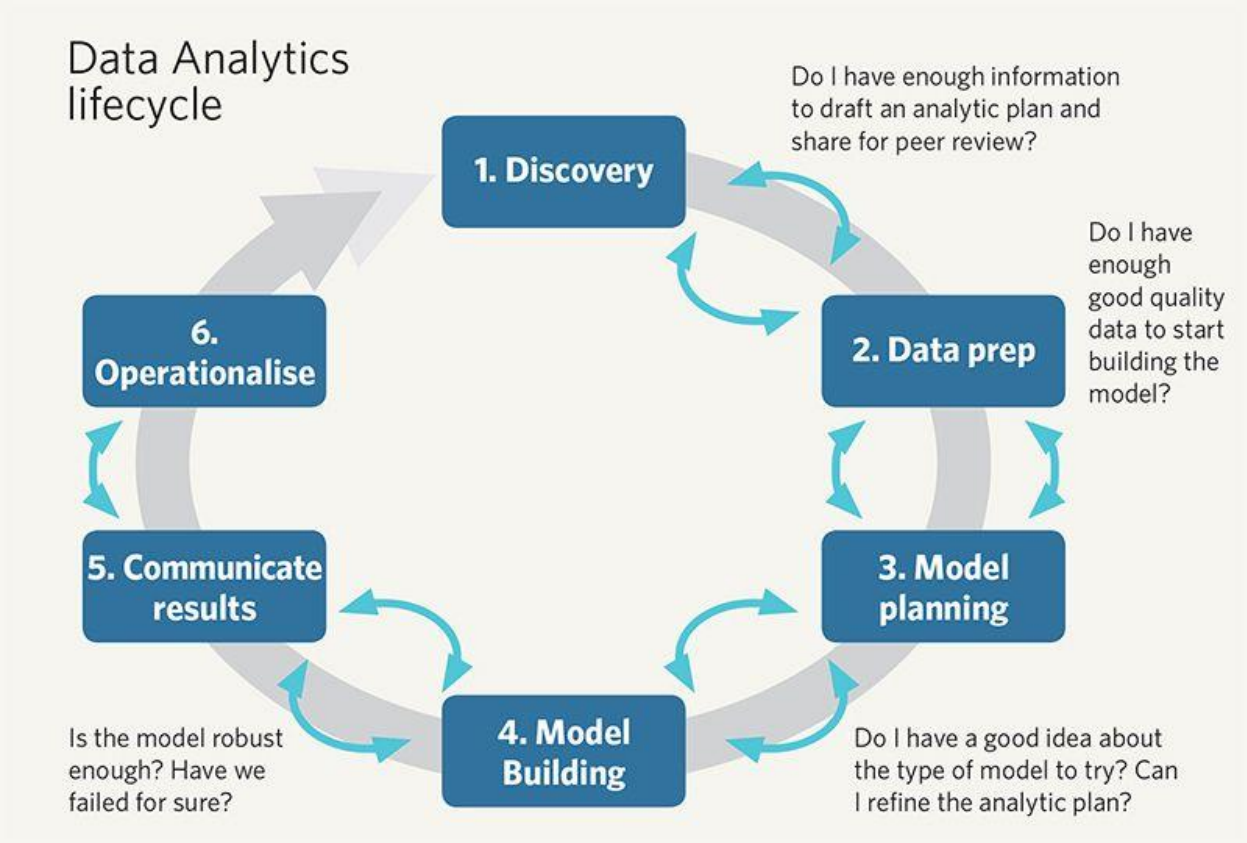


Figure: Data Analytics Life Cycle from *Data Science & Big Data Analytics* (page 29).

First, let's examine this life cycle versus traditional scientific hypothesis testing and experimental design.

Classic:

Problem and Background Research (with Questions/Hypothesis)
Design Experiment
Data
Model
Analysis
Conclusions

Data Science:

Data
Problem
Exploratory Analysis (plots, graphs, descriptive stats)
Model
Conclusions, Operationalize

Some notes about the data analytics lifecycle:

- It's not a waterfall; each phase loops to previous and next steps and may need to do some smaller cycles (more like prototyping in software engineering)
- The entire lifecycle is cyclical – no finality since the operationalize step often collects more data to discover

(**Team**) In the table below, discuss questions, goals or items to consider at each phase of the course project. Make notes about things the team should consider about the project, questions that the team might explore, the type of dataset to use, and the type of dashboard information the resulting analysis might give.

Phase	Brief Description	Team Notes, Questions, and Goals
1. Discovery **Proposal** **Due Feb 3**	Learn domain, history, and other projects; assess resources (time, people, tech, data); formulate initial questions about the data; set criteria for successful analysis outcomes; explore different sources of data or consider combining data sources	
2. Data Prep + Exploration **Data exploration** **Due Feb 24**	Data exploration; is the data good enough; do I need to clean data or get more data? get data into form for processing; initial graphs/plots/stats from exploratory data analysis	
3. Model Planning	Determine which data to use to create model; determine with model(s) are most useful; study relationships between variables	
4. Model Building **Models + analysis** **Due Mar 31**	Executes data analysis; could be training/test data; build multiple models and compare; is hardware adequate?	
5. Communicate Results **Dashboard** **Due Apr 28**	Communicate with major stakeholders, was analysis successful?, key findings, quantify business value, develop narrative to communicate with stakeholders; describe risks/bias in the data and models	
6. Operationalize **Presentation Video** **Due May 3** **Demos May 4**	Deliver final reports, code, dashboards, insights, and new tools in a production environment	

(**Team**) Establish how the team will work on the project together. You should be spending about 2 hours outside class each week on the project, and this time may be individual work time and/or team meetings.

Weekly team meeting times: _____

How will the team store documents and communicate? _____
(MS Team group, Slack, OneDrive, etc.)

Set up a github repository for the project (nominate someone to create it and add the group members). You can then post your final project dashboard live via a website on github. See https://pages.github.com/ for more information.

(**Team**) Research existing dashboards and datasets for your project domain. Track what you find as helpful resources and datasets.

Activity 8: Asking Questions Practice

You will now practice asking and generating questions given existing datasets.

Domain 1: Baseball

1. Go to https://www.baseball-reference.com/.
2. Explore some of the data you see on this website.
    a. Batting
    b. Pitching
    c. Fielding
    d. Leaders & Awards
3. Think about the "who, what, where, when, and why" questions when exploring the data.
4. Generate a list of at least three questions below (note: you do not need to solve these – this is just practice for generating questions based on data)

Example: What is the trajectory of a player's performance as they age?
Example: Does batting percentage correlate to field position?
Example: How could we quantify the value of a trade between teams?
Example: Do left-handers have longer careers than right-handers?
Example: Are weights of players increasing in the population over time?

Questions:

1.

2.

3.

4.

Domain 2: Movies

1. Go to https://www.imdb.com/
2. Explore some of the data you see on this website.
    a. Actors
    b. Films
    c. Ratings
3. Think about the "who, what, where, when, and why" questions when exploring the data.
4. Generate a list of questions below (note: you do not need to solve these – this is just practice for generating questions based on data)

Example: Can we predict how well people will like a movie?
Example: Can we predict how much money a movie will gross in the theaters?
Example: What is the age distribution of actors and actresses in film?

Questions:

1.

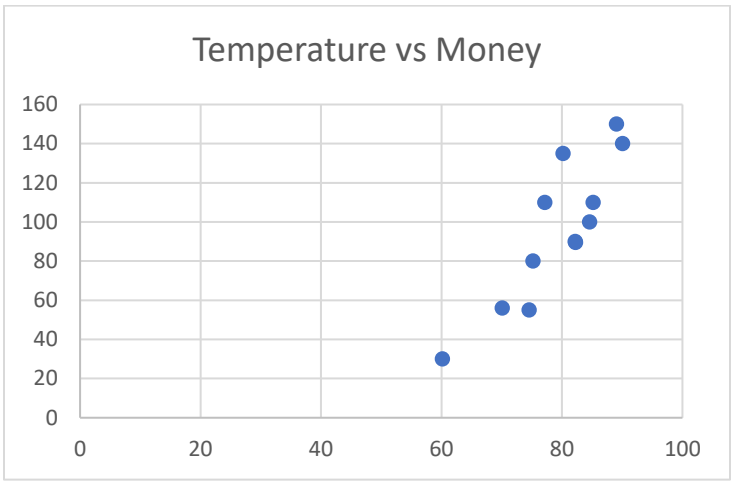
2.

3.

4.

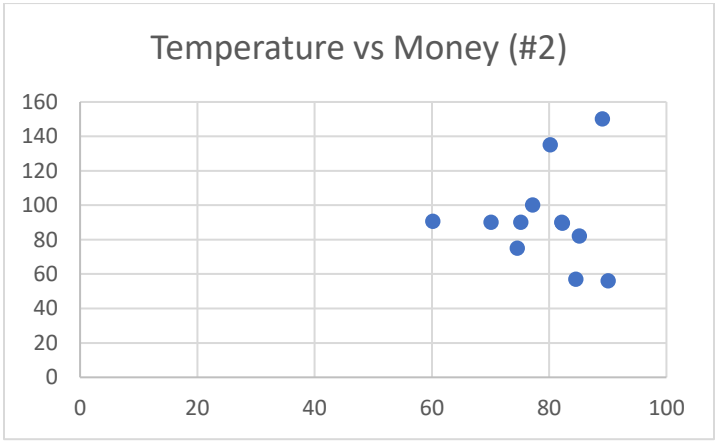
CS 438: Models, Scatterplots, and Simple Regression

Relationship between two variables:
Suppose X = temperature and Y = money made selling lemonade
We plot the temperature on the x-axis and $ on the y-axis. Suppose that plot looks like this.



1. Does money relate to temperature? Why or why not?

Suppose the plot looks like this.



2. Does money relate to temperature? Why or why not?

Correlation

What is correlation? The **sample correlation coefficient**, **r**, is defined as:

$$r = \frac{\sum[(x-\bar{x})(y-\bar{y})]}{\sqrt{[\sum(x-\bar{x})^2][\sum(y-\bar{y})^2]}}$$

Here, x and y are values for a single occurrence or observation. x bar is the average of the x values for the entire dataset. y bar is the average of y values for the entire dataset. The summation signs are over all data items in the dataset sample.

For example, X could be height and Y could be blood pressure.

Let's consider r.

When is r close to 1?	When y positively relates to x
When is r close to 0?	When y does not relate to x at all
When is r close to -1?	When y negatively relates to x

3. Draw a scatterplot of data where r is close to 0.

4. Draw a scatterplot of data where r is close to -1.

5. Draw a scatterplot of data where r is close to 0.5.

Example correlation calculation:

Blood pressure readings are collected from 6 people. The age and high number from the blood pressure are recorded. Is age correlated with blood pressure?

Subject	**Age (x)**	**Pressure (y)**
A	43	128
B	48	120
C	56	135
D	61	143
E	67	141
F	70	152
means	**57.5**	**136.5**

1. Calculate $(x - \bar{x})$ and $(y - \bar{y})$ for each sample:

x - x_bar	**y - y_bar**
-14.5	-8.5
-9.5	-16.5
-1.5	-1.5
3.5	6.5
9.5	4.5
12.5	15.5

2. Calculate the squares of each of these and the sum for each column:

(x-x_bar)^2	**(y-y_bar)^2**
210.25	72.25
90.25	272.25
2.25	2.25
12.25	42.25

	90.25	20.25
	156.25	240.25
Sum	**561.5**	**649.5**

3. Calculate $(x - \bar{x})$ * $(y - \bar{y})$ and the sum:

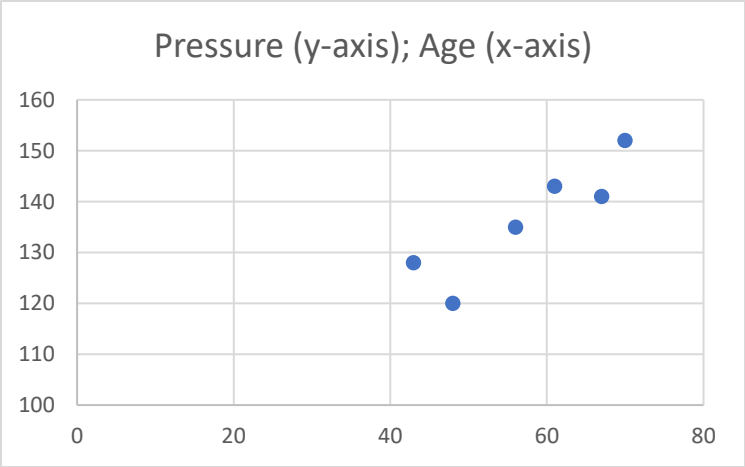
(x-x_bar)(y-y_bar)
123.25
156.75
2.25
22.75
42.75
193.75
541.5

4. Sample correlation coefficient $r = 541.5 / \sqrt{(561.5 * 649.5)}$ = 0.896673.

NOTE: CORRELATION **DOES NOT** IMPLY CAUSATION. THIS IS ONE OF THE MOST MISUNDERSTOOD CONCEPTS OF DATA ANALYSIS. IT IS SIMPLY A MEASURE OF RELATIONSHIP.

Scatterplots

A graph of dots showing the (x,y) values of the data. It is a visual representation of how two variables are related. Here is the scatterplot of (age, pressure) from the data above:

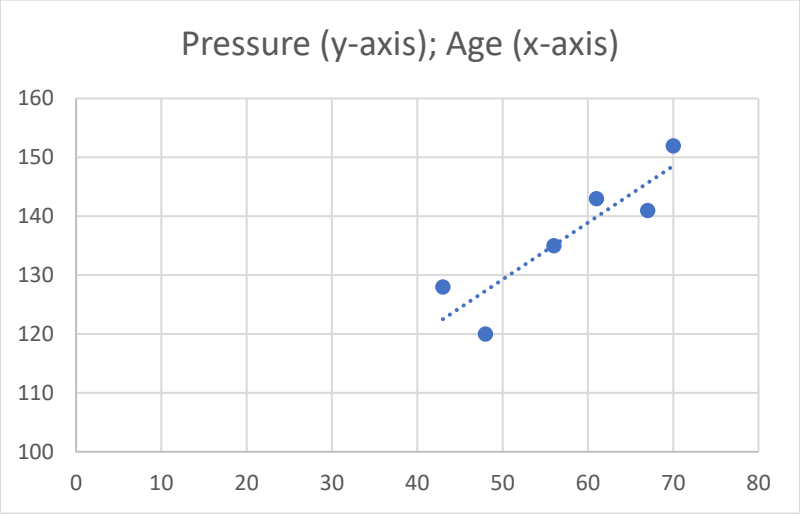


Draw a line that best "fits" these points in the graph above.

Compare your line with other students.

How should we define “best” for the possible lines? Discuss with your classmates.

Here is the same scatterplot with the regression line (trendline):



Linear Regression

Creating the trendline is called linear regression using one **predictor** variable (in this case, age) with a **response** variable (in this case, blood pressure). Why is this trendline useful?

This gives us a model built from the data. These models can then be used for prediction. For example, suppose a new patient comes to the clinic and that patient is 50 years old. What would you predict for the new patient’s blood pressure? _____

Simple linear regression builds a linear model:

Usually we say the independent variable is the ***regressor*** *variable or* ***predictor*** x
Usually we say the dependent variable is the ***response*** *variable* y

How do we determine the equation for the line?

$$Y = \beta_0 + \beta_1 X + error$$

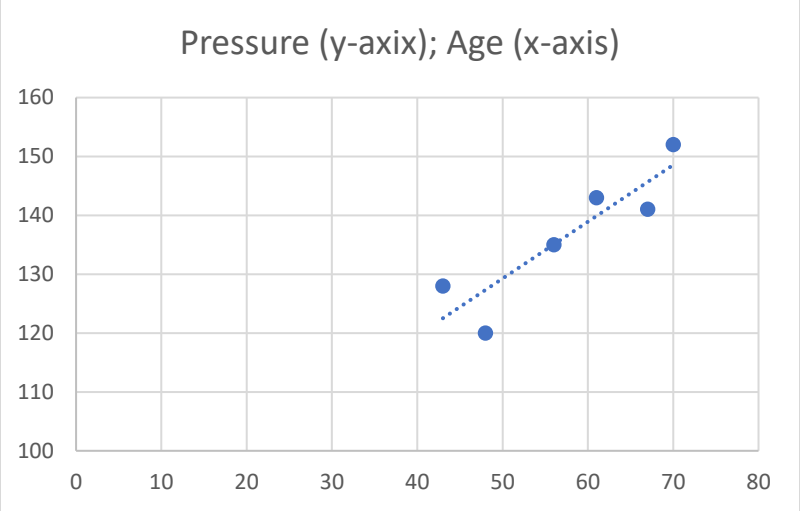
We want to minimize the error, so the predicted Y is as close to the observed y value as possible. How do we do this?

We minimize the sum of squares from the predicted Y (trendline estimate) and the observed y for all the data points. This is called *least squares estimation.*

Fortunately, we have equations to estimate the beta values that do least squares estimation.

The errors (observed_y – estimated_y) are called *residuals*. Residuals can be positive and negative. Positive residuals fall above the trendline. Negative residuals fall below the trendline. We try to minimize the sum of the squares of the residuals. We know that this minimum has to be 0 (if all points fall on the trendline).

Use the graph to estimate the residuals for the patients:



Patient A observed – predicted: _____ // patients are ordered left to right in graph
Patient B observed – predicted: _____
Patient C observed – predicted: _____
Patient D observed – predicted: _____
Patient E observed – predicted: _____
Patient F observed – predicted: _____

Here are the equations to compute the beta values:

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})\ (x_i - \bar{x})}$$

Example Calculation from the Data Above

Because β_1 is used in β_0, we calculate β_1 first, which is the slope of the line.

1. Calculate numerator. We did this above and the result is 541.5.
2. Calculate the denominator. We did this above and the result is 561.5.
3. $\beta_1 = 0.964381$

4. Calculate mean of observed y and mean of observed x. We did this above. The mean of y is 136.5. The mean of x is 57.5.
5. $\beta_0 = 81.04809$

Using R

In R, you can build a regression model using the command `lm`. Here is the output of the same data:

First, the code the build the data frame:

```
age <- c(43, 48, 56, 61, 67, 70)
pressure <- c(128, 120, 135, 143, 141, 152)
mydata <- data.frame(age, pressure)
mydata
names(mydata)
```

OK, the data looks good. Let's see the scatterplot:

```
plot(mydata$age, mydata$pressure)
```

We can create the regression model using the lm command:

```
mod <- lm(pressure ~ age, data=mydata)
summary(mod)
```

What does this print?

```
Call:
lm(formula = pressure ~ age, data = mydata)

Residuals:
       1        2        3        4        5        6
 5.48353 -7.33838 -0.05343  3.12467 -4.66162  3.44524

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  81.0481    13.8809   5.839  0.00429 **
data$age      0.9644     0.2381   4.051  0.01546 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.641 on 4 degrees of freedom
Multiple R-squared:  0.804,     Adjusted R-squared:  0.755
F-statistic: 16.41 on 1 and 4 DF,  p-value: 0.01546
```

Let's see … did the regression produce the same values as our own calculation?

Look at the residuals … did they match your estimates from above?

Look at Multiple R-squared. It is r^2 for linear regression with one predictor variable. Multiple R-squared is always between 0 and 1. Remember, r can be between -1 and 1 and indicates positive and negative relationships.

We can also use R to calculate the correlation coefficient:

```
cor(mydata$age, mydata$pressure)

This produces [1] 0.8966728, which is what we calculated earlier.
```

If we multiple the correlation coefficient by itself, we get the value 0.8040221.

We can also add the regression line to the plot:

```
abline(mod)
```

Using the Model for Prediction

Suppose a new patient comes to the clinic who is aged 53. We can predict her blood pressure:

```
predict(mod,data.frame(age=(c(45))), interval="confidence")
```

This returns:

```
       fit     lwr      upr
1 124.4452 113.998 134.8925
```

This gives us the 95% confidence interval for the predicted value, given that we are estimating the pressure mean correctly.

We can also get the prediction interval for her blood pressure:

```
predict(mod,data.frame(age=(c(45))), interval="prediction")
```

This returns:

```
       fit      lwr      upr
1 124.4452 105.6184 143.2721
```

Notice that we get the same fit value, but the interval is larger. The confidence interval assumes the data is randomly sampled from a normal distribution, so it is really estimating the mean parameter for blood pressure. A prediction interval tells you where you can expect to see the next point that is sampled (where a single value will fall versus where the sample mean will fall as in the confidence interval). Prediction intervals account for the uncertainty in knowing the true population mean plus the uncertainty in data scatter.

Because this is a new patient who is 45 years old, we can predict that her blood pressure will be between 105.6 and 143.3.

Activity 9: Practice with Scatterplots and Linear Models in R

Use R and experiment with the cars dataset. The cars dataset has speed versus stopping distance for 50 different observations.

Make a scatterplot of speed versus stopping distance.

Build a linear model to predict stopping distance.

What is the model?

Stopping distance = _____

Use the model to predict the stopping distance for speed of 50.

CS 438: Multiple Regression

Think about a scenario where you want to predict the value of something given several factors.

What are you predicting? _____

Give at least three factors/variables that could influence the value you are predicting?

1.

2.

3.

For example, you want to predict the price of an electric bill given outdoor high temperature, number of people in a household, and square footage of the household.

How would you build this model from data?

If you want to build a linear model of **multiple variables**, you may use multiple regression. For example, you are a biologist and want to predict the number of spring babies in a herd of antelope based on the current population, how much precipitation happened over the winter, and the severity of the winter.

Spring Fawn Count (/100)	Antelope Pop (/100)	Precipitation	Winter severity
2.900000095	9.199999809	13.19999981	2
2.400000095	8.699999809	11.5	3
2	7.199999809	10.80000019	4
2.299999952	8.5	12.30000019	2
3.200000048	9.6	12.60000038	3
1.899999976	6.800000191	10.60000038	5
3.400000095	9.699999809	14.10000038	1
2.099999905	7.900000095	11.19999981	3

You want to build a model:

$$FawnCount = \beta_0 + \beta_1 * pop + \beta_2 * precipitation + \beta_3 * winterSeverity$$

Notice that the model is linear and we are trying to minimize the squared error of the fawn count predictions when estimating the values for the betas.

Solving for the beta values involves a system of linear equations.

Suppose there are k variables in the model that is used to predict the value for y. We can write the data in the form of a table. Note that the top row of the table is a header column, with one header per variable.

y	x_1	x_2	x_3	…	x_k
y_1	x_{11}	x_{12}	x_{13}		x_{1k}
y_2	x_{21}	x_{22}	x_{23}		x_{2k}
y_3	x_{31}	x_{32}	x_{33}		x_{3k}
…					
y_n	x_{n1}	x_{n2}	x_{n3}		x_{nk}

Least squares equations:

$$n * \beta_0 + \beta_1 * \sum_{i=1}^{n} x_{i1} + \beta_2 * \sum_{i=1}^{n} x_{i2} + \cdots + \beta_k * \sum_{i=1}^{n} x_{ik} = \sum_{i=1}^{n} y_1$$

$$\beta_0 * \sum_{i=1}^{n} x_{i1} + \beta_1 * \sum_{i=1}^{n} x_{i1} * x_{i1} + \beta_2 * \sum_{i=1}^{n} x_{i1} * x_{i2} \ldots + \beta_k * \sum_{i=1}^{n} x_{i1} * x_{ik} = \sum_{i=1}^{n} x_{i1} y_1$$

…. … … … …

$$\beta_0 * \sum_{i=1}^{n} x_{ik} + \beta_1 * \sum_{i=1}^{n} x_{ik} * x_{i1} + \beta_2 * \sum_{i=1}^{n} x_{ik} * x_{i2} \ldots + \beta_k * \sum_{i=1}^{n} x_{ik} * x_{ik} = \sum_{i=1}^{n} x_{ik} y_i$$

There are (k+1) equations and (k+1) regression coefficients, so it can be solved with linear algebra. R does this computation for us, but now you know how R is calculating the coefficients.

Let's do the fawn count prediction using these equations (see excel file).

Now, let's see what R gives us:

```
count <- c(2.9, 2.4, 2, 2.3, 3.2, 1.9, 3.4, 2.1)
pop <- c(9.2, 8.7, 7.2, 8.5, 9.6, 6.8, 9.7, 7.9)
prec <- c(13.2, 11.5, 10.8, 12.3, 12.6, 10.6, 14.1, 11.2)
winter <- c(2, 3, 4, 2, 3, 5, 1, 3)
ant <- data.frame(count, pop, prec, winter)
ant
fit <- lm(ant$count ~ ant$pop + ant$prec + ant$winter)
fit

Call:
lm(formula = ant$count ~ ant$pop + ant$prec + ant$winter)

Coefficients:
(Intercept)      ant$pop     ant$prec   ant$winter
    -5.9220       0.3382       0.4015       0.2629
```

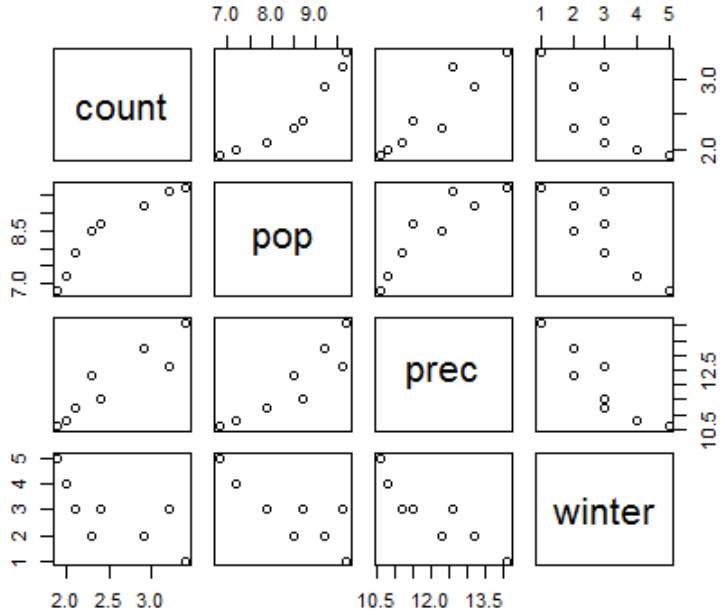
OK, that's good. We get the same coefficients as we found by hand.

What is the model?

$$FawnCount = -5.922 + 0.3382 * pop + 0.4015 * precipitation + 0.2629 * winterSeverity$$

So, we can use this equation to make predictions. Suppose the population is 8.26, the precipitation is 13.5, and the winter severity is 2. How many spring fawns would you expect? _____

Here's what the graph of all pairs scatterplots looks like:



Now, let's see how good of a model this is by applying the summary function:

```
Call:
lm(formula = ant$count ~ ant$pop + ant$prec + ant$winter)

Residuals:
       1        2        3        4        5        6
-0.11533 -0.02661  0.09882 -0.11723  0.02734 -0.04854
       7        8
 0.11715  0.06441

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.92201    1.25562  -4.716   0.0092 **
ant$pop      0.33822    0.09947   3.400   0.0273 *
ant$prec     0.40150    0.10990   3.653   0.0217 *
ant$winter   0.26295    0.08514   3.089   0.0366 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1209 on 4 degrees of freedom
Multiple R-squared:  0.9743,   Adjusted R-squared:  0.955
F-statistic: 50.52 on 3 and 4 DF,  p-value: 0.001229
```

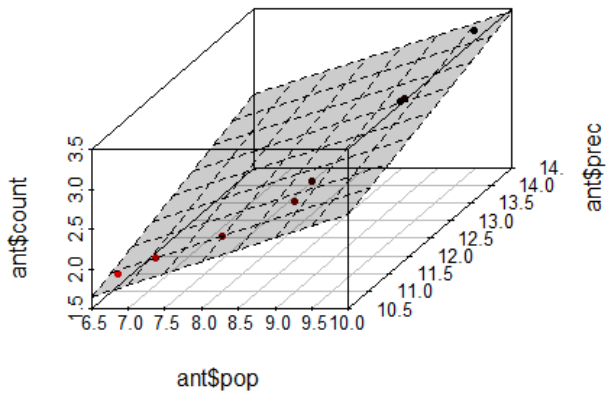
Interpretation
The residuals are, like before, the difference of the observed minus the estimated values for each observation. The coefficients are the same as before, but now we see the t-values and p-values for how much each factor is significant to the model. There are 4 degrees of freedom (N – 1 - #factors). In this case, there are (8 – 1 – 3) = 4 degrees of freedom.

The Multiple R-squared gives us an estimate of how much the variables account for the variance in the model. This is a high R-squared, so the variables together are predicting much of the fawn count. The F-statistic tells us if the regression model is better than the intercept-only model (no variables, just the mean of the response variable).

3D Scatterplots
Here, we create a linear model and plot two predictor variables (x = pop, y = prec) and the response variable (z = count).

```
install.packages("scatterplot3d")
library("scatterplot3d")
s3d <- scatterplot3d(ant$pop, ant$prec, ant$count, type="p",
highlight.3d = TRUE, pch = 20)
fit2 <- lm(count ~ pop + prec, data=ant)
s3d$plane3d(fit2, draw_polygon = TRUE, draw_lines = TRUE)
```

You can see the plane that is the predicted values based on the regression model.

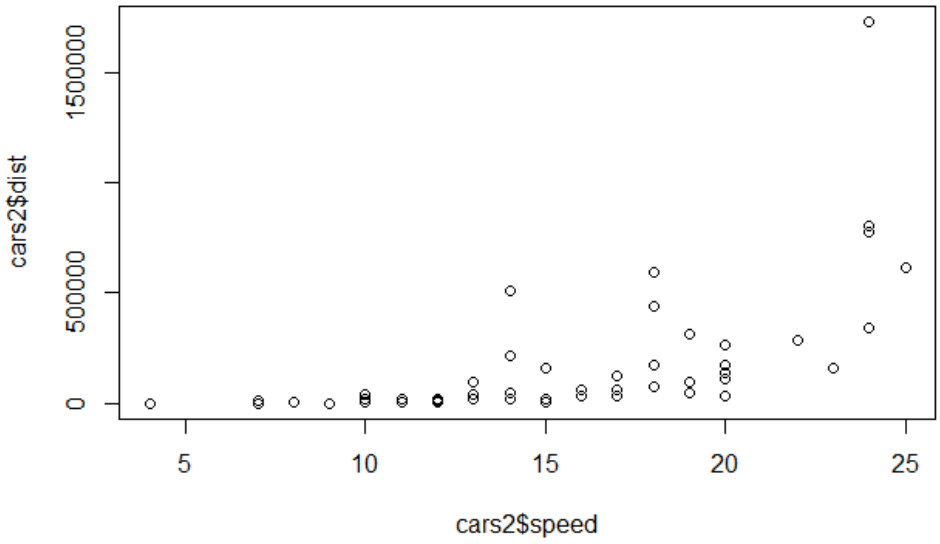


We can do multiple regression with two more predictor variables, but it is difficult to graph in 4D. Hopefully, this 3D version gives you a mental visualization of regression models.

CS 438: Variable Transformations and Polynomials for Regression

So far, we have looked at models with one or more predictor variables that predict another variable. But, the data may not relate well as a linear combination.

Consider the data below. (Speed is the `speed` of a car and `dist` is how far the car coasts before coming to a stop. This is fictitious data for the example.)



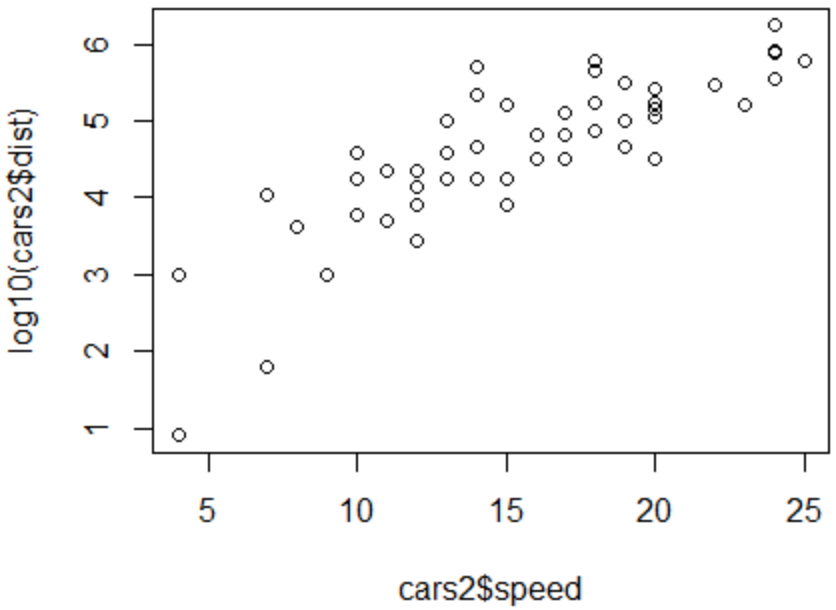
1. Draw your best fit regression line for this data on the ploy above.

2. Do the residuals get bigger as the speed increases?

3. Would a linear model be a good choice for this dataset? Why or why not?

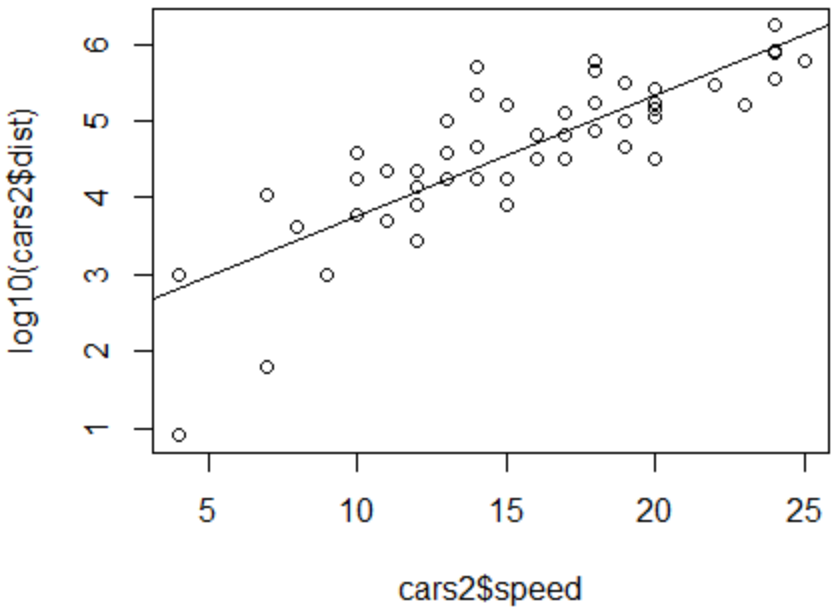
4. What could we do to the data so that a linear model would be a better fit?

Let's look at the scatterplot if we take `log10` of the distance and plot:



It's looking like a better dataset for simple linear regression. We can create the linear model using the transformed distance variable:

```
> mod_log <- lm(log10(dist) ~ speed, data=cars2)
> summary(mod_log)

Call:
lm(formula = I(log10(dist)) ~ speed, data = cars2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.91008 -0.27100 -0.02192  0.31374  1.32267

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.18379    0.25554   8.546 3.34e-11 ***
speed        0.15734    0.01571  10.015 2.41e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5815 on 48 degrees of freedom
Multiple R-squared:  0.6763,   Adjusted R-squared:  0.6696
F-statistic: 100.3 on 1 and 48 DF,  p-value: 2.413e-13

> abline(mod_log)
```



You can transform any and all variables before applying regression. That is why it is important to view the scatterplot of your data, especially if there are one or two predictor variables – that way you may look for patterns that would benefit from transformations.

Since we create a linear model that involved `log10` of our predictor variable, we would need to transform the predicted value back into the original units.

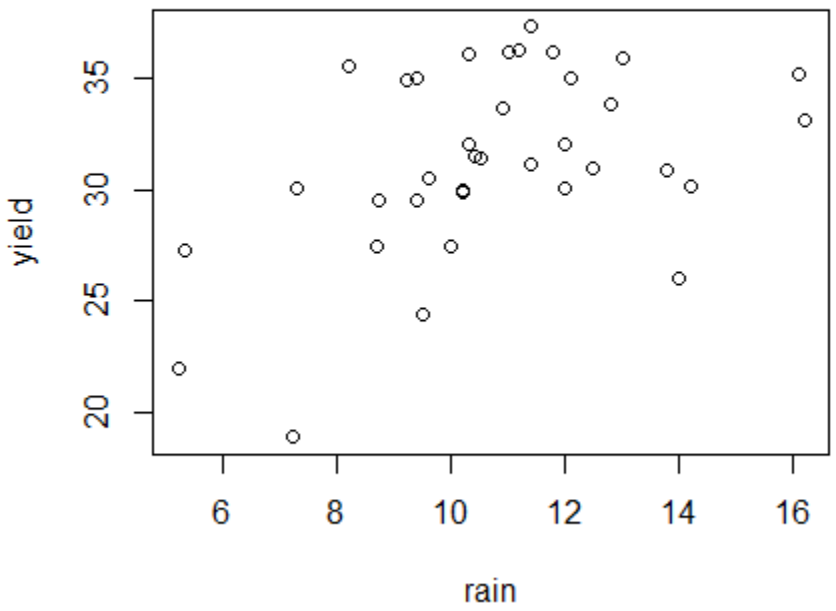
For example, looking at the graph above:

5. Suppose speed is 15. What is the predicted `log10` of the distance given the trendline? `ld` = _____

6. Now, we would need apply 10^{ld} = _____ to get the predicted distance.

Polynomials in Regression – Corn Data

Suppose rainfall and corn crop yield are measured for the past 35 years. The data is shown below.



7. Draw a best fit trend line through the data (does not need to be linear, but should be a smooth curve).

Maybe a higher-order polynomial would be better than a linear model. We can run a regression in R to fit the following function:

$$yield = \beta_0 + \beta_1 * rain + \beta_2 * rain^2$$

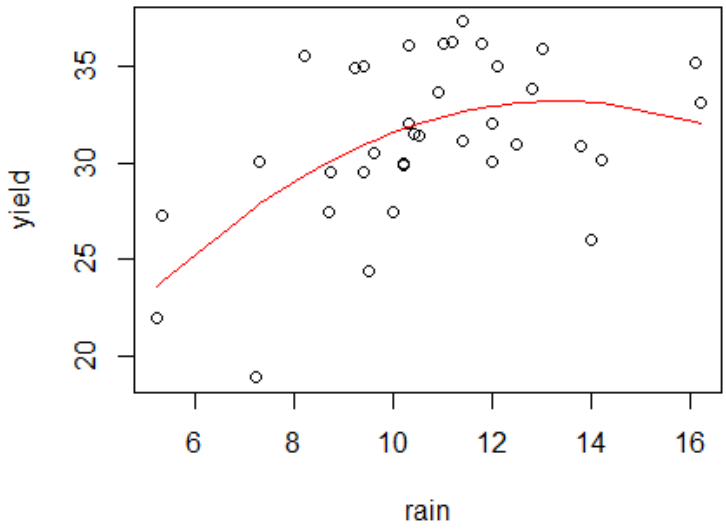
```
> mod_corn <- lm(corn$yield ~ poly(corn$rain, 2))
> summary(mod_corn)

Call:
lm(formula = corn$yield ~ poly(corn$rain, 2))

Residuals:
    Min      1Q  Median      3Q     Max
-8.8391 -1.9255 -0.5262  3.1692  6.1278

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)            31.3194     0.6022  52.005  < 2e-16 ***
poly(corn$rain, 2)1    11.1890     3.6134   3.096  0.00398 **
poly(corn$rain, 2)2    -7.9110     3.6134  -2.189  0.03575 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.613 on 33 degrees of freedom
Multiple R-squared:  0.3035,    Adjusted R-squared:  0.2613
F-statistic: 7.191 on 2 and 33 DF,  p-value: 0.002558
```

So, this creates a regression model of:

$$yield = \mathbf{31.3194} + \mathbf{11.189} * rain + (\mathbf{-7.911}) * rain^2$$

We can plot this by creating the predicted curve of this model:

```
> predicted_corn3 <- predict(mod_corn3, data.frame(x=corn$rain), interval="co
nfidence")
> lines(corn$rain, predicted_corn[,1], col="red")
```



Using this second-degree polynomial model, if the rainfall is 8 inches, what is the predicted crop yield?

_____

What is the rainfall amount for the maximum yield in this model? _____

8. How do we know if we should go to a third-order polynomial model?

We can try to add another term to the model for $rain^3$:

```
> mod_corn3 <- lm(corn$yield ~ poly(corn$rain, 3))
      Call:
      lm(formula = corn$yield ~ poly(corn$rain, 3))

      Residuals:
          Min       1Q   Median       3Q      Max
      -9.0216 -1.8982 -0.5671   3.0863   5.8962

      Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
      (Intercept)          31.3194     0.6106  51.290  < 2e-16 ***
      poly(corn$rain, 3)1  11.1890     3.6638   3.054  0.00452 **
      poly(corn$rain, 3)2  -7.9110     3.6638  -2.159  0.03843 *
```

```
poly(corn$rain, 3)3   1.1544     3.6638   0.315  0.75474
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.664 on 32 degrees of freedom
Multiple R-squared:  0.3057,    Adjusted R-squared:  0.2406
F-statistic: 4.696 on 3 and 32 DF,  p-value: 0.00792
```

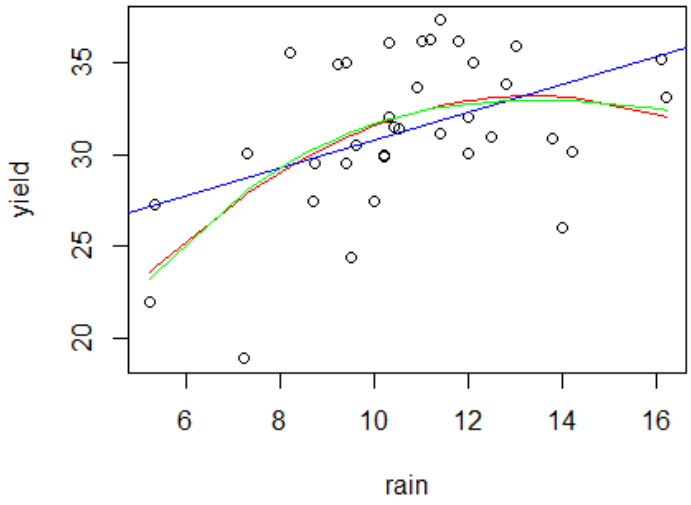
We see that the third-order term is **no longer** significant (as an independent variable) in the model.

What is the model? $Yield$ = _____

9. Let’s try to fit a third model -- simple linear regression line to our data:

```
> mod_corn1 <- lm(corn$yield ~ corn$rain)
> summary(mod_corn1)
Call:
lm(formula = corn$yield ~ corn$rain)

Residuals:
    Min      1Q  Median      3Q     Max
-9.7626 -2.2105 -0.0738  2.6972  6.0832

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  23.2325     2.8257   8.222 1.36e-09 ***
corn$rain     0.7542     0.2568   2.937  0.00591 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.81 on 34 degrees of freedom
Multiple R-squared:  0.2024,    Adjusted R-squared:  0.1789
F-statistic: 8.626 on 1 and 34 DF,  p-value: 0.00591
```

OK, now we have three different models, each using a different order of polynomial, shown below.



11. Which is the best of the three? _____

What data did you use to make your decision? _____

We can run an analysis of variance to see if the models (what they predict) are significantly different:

```
> anova(mod_corn1, mod_corn)
      Analysis of Variance Table

      Model 1: corn$yield ~ corn$rain
      Model 2: corn$yield ~ poly(corn$rain, 2)
        Res.Df    RSS Df Sum of Sq      F  Pr(>F)
      1     34 493.46
      2     33 430.88  1    62.584 4.7931 0.03575 *
      ---
      Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> anova(mod_corn1, mod_corn3)
      Analysis of Variance Table

      Model 1: corn$yield ~ corn$rain
      Model 2: corn$yield ~ poly(corn$rain, 3)
        Res.Df    RSS Df Sum of Sq      F Pr(>F)
      1     34 493.46
      2     32 429.55  2    63.916 2.3808 0.1087

> anova(mod_corn, mod_corn3)
      Analysis of Variance Table

      Model 1: corn$yield ~ poly(corn$rain, 2)
      Model 2: corn$yield ~ poly(corn$rain, 3)
        Res.Df    RSS Df Sum of Sq      F Pr(>F)
      1     33 430.88
      2     32 429.55  1    1.3326 0.0993 0.7547
```

Does this agree with your assessment of the best model of the three? _____

Measures: Multiple R-Squared and Adjusted R-Squared

The **Multiple R-squared** gives us a measure of the percentage of the variability in the data for yield is explained by rain.

When building models over several predictors, you should look at **Adjusted R-squared**. The Multiple R-squared value will go up as you add predictors to a model, even due to chance alone. A model with more terms will give us better Multiple R-squared values, so it can mislead you into concluding that a model with more predictors is always better. Plus, more predictors can lead to over-fitting the data (higher order polynomials that connect the dots in the training data really well, but would do a poor job predicting new data). Instead, look at the Adjusted R-squared. This accounts for the number of predictors in your model, so you can compare two or more models with different number of predictors.

Look at the models to see the Adjusted and Multiple R-squared values

Model	Adjusted R-Squared	Multiple R-Squared
Linear	.1789	.2024
Degree 2 Poly	.2613	.3035
Degree 3 Poly	.2413	.3057

Which has the lowest Adjusted R-squared value? _____

How are these calculated?

First, we saw the calculation for r (correlation coefficient) in an earlier lecture. Regular r^2 is the square of r.

What about **Multiple R-squared**? Now, we use the model to create predicted response levels and use the actual observed levels to estimate how much of the variability is estimated by the linear model:

$$\hat{R}^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

Here, the Y with the hat is the predicted value from the model and Y with the bar is the sample mean of the observed response values. The top of the fraction is the sum of the residuals squared. The bottom of the fraction is the sum of the residuals for a model that is a straight horizontal line for the mean of Y.

When is $\hat{R}^2$ close to 1? _____

You can see that the Multiple R-squared calculation does not take into considering of how many predictors are in the model.

The Adjusted R-squared metric takes into account the *number of parameters (beta values)* in the regression (below, this is p).

$$\hat{R}^2_{adjusted} = \hat{R}^2 - \left(1 - \hat{R}^2\right)\frac{p}{(n - p - 1)}$$

You can see that the fractional part is between 0 and 1 and $\left(1 - \hat{R}^2\right)$ is between 0 and 1, so the Adjusted R-squared value is less than the Multiple R-squared value.

You can now create many different models using regression. Remember:

- You can transform variables (using log, sqrt, etc.) to create better linear models.
- You can create higher-order terms from variables to create **non-linear** (curved) models.
- Creating models is like an art – there are several options regarding transformation of variables and polynomials
- The examples from this handout use just one predictor variable, but a model may have multiple predictor variables along with multiple orders of polynomials for those variables.

CS 438: Regression Math Summary

Now that we have been applying linear regression, this is a summary of some of the terms associated with regression.

(Review) What is a **residual**? For a particular observed value, it is the difference of the observed value y_i and the predicted value $\hat{y}_i$ based on the model.

Residual: $y_i - \hat{y}_i$

The **residual sum of squares** (error sum of squares) is defined as:

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

Take each residual, square it, and get the sum.

The **regression sum of squares** (error sum of squares) is defined as:

$$SE_R = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

Take each y value predicted by the model and subtract the mean of the observed values, square them, and sum them. NOTE that this is different than the residual sum of squares, the term we minimize to create the model.

In R:

```
library(MASS)
plot(Boston$rm, Boston$medv)
fit <- lm(medv ~ rm, data=Boston)
fit
abline(fit, col="red")
summary(fit)
anova(fit) %will report the SSE
residuals(fit)
sqEr <- residuals(fit)*residuals(fit)
sqEr
sum(sqEr)
# regression sum of squares
pred <- predict(fit, rm=Boston$rm)
regress <- pred - mean(Boston$medv)
sqErR <- regress*regress
sum(sqErR)
```

*From the anova function (22062 is the **residual sum of squares,** 20654 is the **regression sum of squares**)*

```
Analysis of Variance Table

Response: medv
           Df Sum Sq Mean Sq F value    Pr(>F)
rm          1  20654 20654.4  471.85 < 2.2e-16 ***
```

```
Residuals 504  22062    43.8
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Also, the residuals function will give you the residuals and the residual sum of squares can be computed directly. From R, the sum(sqEr) is 22061.88 and sum(sqErR) is 20654.42.

Simple Linear Regression

Simple Linear Regression: estimates of variance of the model, variance of $\boldsymbol{\beta_0}$, and variance of $\boldsymbol{\beta_1}$

For simple linear regression (one predictor variable), the variance of model is the SSE divided by the degrees of freedom (n-2), where n is the number of observations in the sample used to create the model:

$$\hat{\sigma}^2 = \frac{SS_E}{n-2}$$

$$V(\beta_0) = \hat{\sigma}^2\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)$$

$$V(\beta_1) = \frac{\hat{\sigma}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Therefore, the **standard error** of each value are the square roots of the variances. Normality is centered at the β coefficients with the standard deviations as calculated above.

Confidence intervals for $\boldsymbol{\beta_0}$, and $\boldsymbol{\beta_1}$

100(1-α)% Confidence interval for $\boldsymbol{\beta_0}$:

$$\left(\beta_0 - t_{\frac{\alpha}{2},n-2} * \sqrt{V(\beta_0)}, \quad \beta_0 + t_{\frac{\alpha}{2},n-2} * \sqrt{V(\beta_0)}\right)$$

Where the t value is from the t-distribution.

100(1-α)% Confidence interval for $\boldsymbol{\beta_1}$:

$$\left(\beta_1 - t_{\frac{\alpha}{2},n-2} * \sqrt{V(\beta_1)}, \quad \beta_1 + t_{\frac{\alpha}{2},n-2} * \sqrt{V(\beta_1)}\right)$$

In R:

```
# variance of model
denom <- length(Boston$rm) - 2
var_fit <- sum(sqEr)/denom
var_fit

# variance of beta_0
rm_mean <- mean(Boston$rm)
diff_x_mean <- Boston$rm - rm_mean
diff_x_mean
```

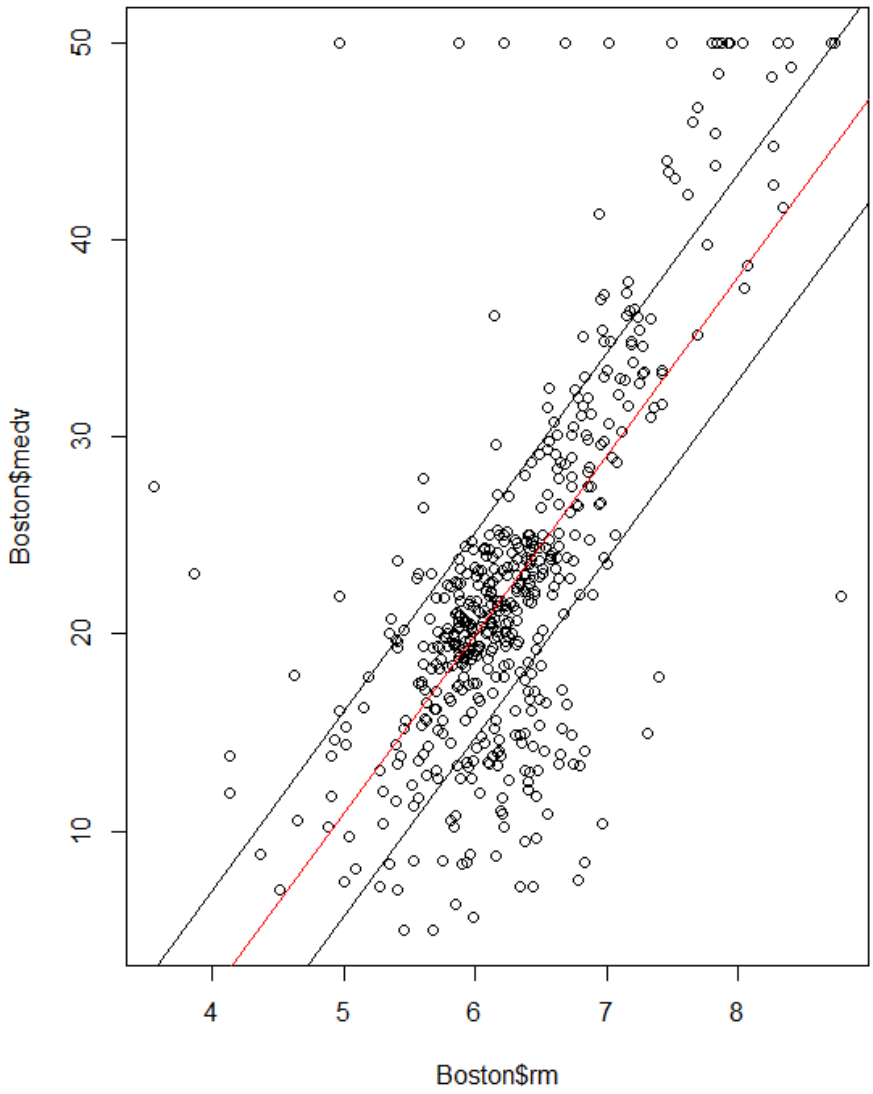
```
diff_sq <- diff_x_mean * diff_x_mean
diff_sq
var_beta0 <- var_fit*(1/length(Boston$rm) +
(rm_mean*rm_mean)/sum(diff_sq))
var_beta0

# variance of beta_1
var_beta1 <- var_fit / sum(diff_sq)
var_beta1

# confidence interval of beta_0
t_val <- 1.97
lb_beta0 <- -34.671 - t_val*sqrt(var_beta0)
lb_beta0
up_beta0 <- -34.671 + t_val*sqrt(var_beta0)
up_beta0

# confidence interval of beta_1
lb_beta1 <- 9.102 - t_val*sqrt(var_beta1)
lb_beta1
ub_beta1 <- 9.102 + t_val*sqrt(var_beta1)
ub_beta1

# with R's confidence interval
confint(fit)
```

Printout:

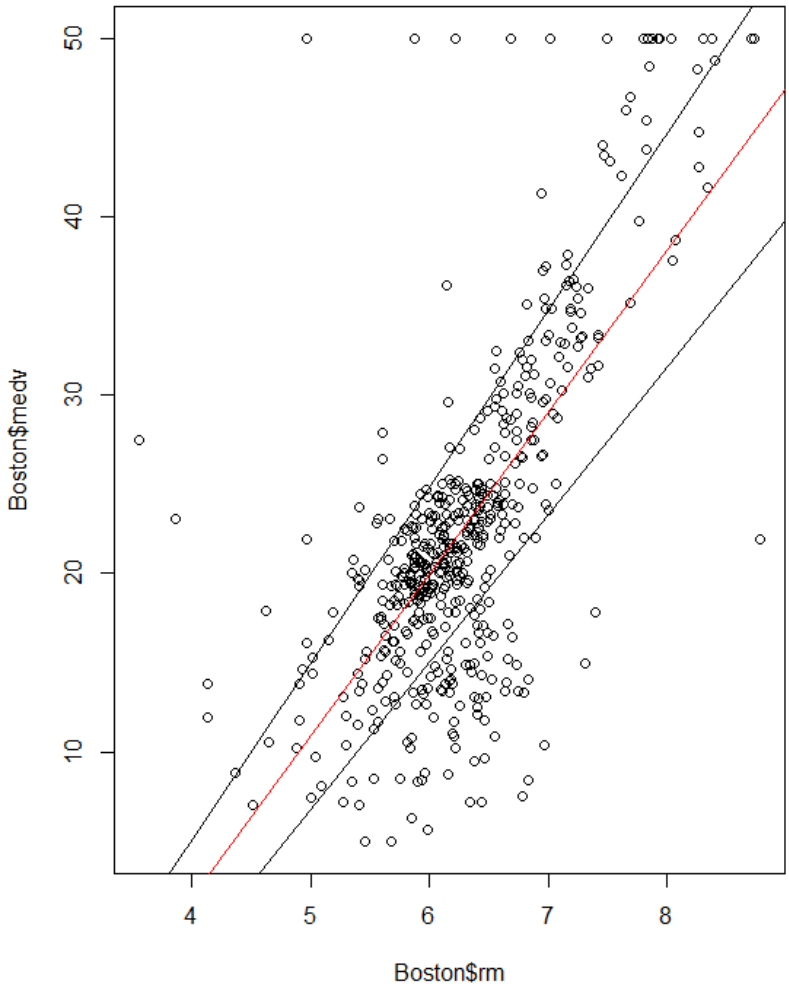
```
> confint(fit)
                2.5 %     97.5 %
(Intercept) -39.876641 -29.464601
rm            8.278855   9.925363

> lb_beta1
[1] 8.276518

> ub_beta1
[1] 9.927482
```

R has a more precise t-table value for the t-distribution, so the values are slightly different from the built-in R command for confidence interval and the direct calculations.

Visual interpretation of confidence interval for beta_0:


Boston$medv
50
40
30
20
10
4
5
6
7
8
Boston$rm

Models with the 95% confidence intervals for beta_1:



Confidence interval for the response variable given a predictor x; this is when using the confidence option in R for the predict function

100(1-α)% Confidence interval for response $E(y_0)$ for given x_0 given the linear model of:

$$\hat{y}_0 = \beta_0 + \beta_1 * x_0$$

is based on the variance of $\hat{y}_0$:

$$V(\hat{y}_0) = \hat{\sigma}^2 \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$$

Then the confidence interval for the response given input x_0 is:

$$(\hat{y}_0 - t_{\frac{\alpha}{2},n-2} * \sqrt{V(\hat{y}_0)},\ \hat{y}_0 + t_{\frac{\alpha}{2},n-2} * \sqrt{V(\hat{y}_0)})$$

Prediction interval for the response variable (future observation); this is the prediction option in R for the predict function

The variance for the prediction (new observation x_0) is:

$$V(pi_{y0}) = \hat{\sigma}^2 \left[1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$$

The 100(1-α)% Confidence interval for the predicted **new observation** is:

$$(\hat{y}_0 - t_{\frac{\alpha}{2},n-2} * \sqrt{V(pi_{y0})},\ \ \hat{y}_0 + t_{\frac{\alpha}{2},n-2} * \sqrt{V(pi_{y0})})$$

You will see that the variance of the predicted mean is smaller than the variance for the predicted new observation, given the extra 1 added to the summation in the brackets.

In R:

```
> new_data <- data.frame(rm = c(5.5))
> predict(fit, new_data, interval="confidence")
       fit      lwr      upr
1 15.39098 14.52427 16.25768
> predict(fit, new_data, interval="prediction")
       fit      lwr      upr
1 15.39098 2.363466 28.41849
```

Multiple Regression

The variance of the model:

$$\hat{\sigma}^2 = \frac{SS_E}{n - p}$$

Note the only chance from simple linear regression is the denominator that includes p, which is the number of parameters (number of betas in the model).

The confidence intervals for the parameters (betas), predicted mean, and the new observation predication interval are calculated in R. It gets more cumbersome with the equations, but they are linear combinations of the ones used in simple regression above.

R, R-squared, Multiple R-Squared, Adjusted R-Squared

These values are important when assessing relationships among variables. Remember, R can be between -1 and 1 where negative R indicates an indirect relationship and a positive R indicates a direct relationship. The closer |R| is to 1, the stronger the relationship.

R squared is what the name says.

Multiple R-Squared and Adjusted R-Squared were defined in previous lecture notes.

CS 438: Classification Introduction, Logistic Regression

Earlier, we looked at linear regression to build prediction models. The response variable in this case is continuous (can take on any numeric value). But, what about the case where the response is binary (true or false, positive or negative, will purchase or will not purchase). In this case, the response is a classification into one of two categories.

Where do you see classification problems used? (For example, spam filters classify incoming email as spam or not spam.)

Class Activity: think of the room as 2-dimensional space. Each one of you is a data point (some x1 and some x2 value. Position yourself somewhere in the room. Your location indicates your x1 and x2 position.

Each one of you is also a category (raise hand up for YES, hand down for NO). What kind of category do you want to use?

Look around you. Do you see any patterns?

Can you visualize a line that separates the data into the two classes? Is it possible to separate the classes perfectly?

Now, move around the room such that you could draw a line or curve to separate the people into two classes.

Activity 1: Consider the data below. Each observation is a patient. The red circles are those with ALL leukemia and the blue x's are those with AML leukemia. The x and y dimensions show scaled gene expression levels for two genes thought to be linked to leukemia.

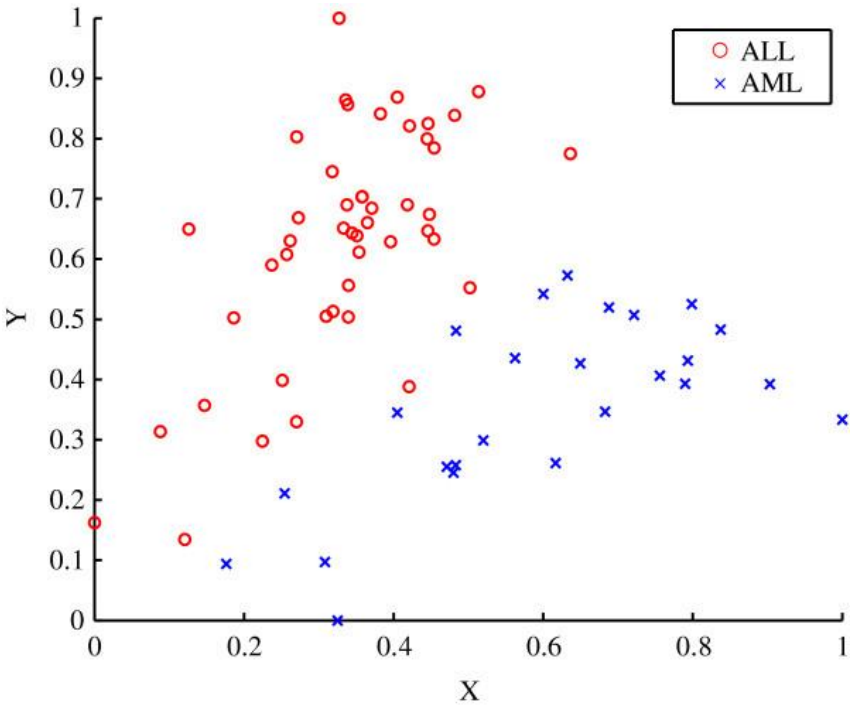
a. Suppose a new patient A has X level of 0.2 and Y level of 0.4. Given the existing data, how would you classify the new patient A? ALL or AML?

b. Suppose a new patient B has X level of 0.8 and Y level of 0.3. How would you classify the new patient? ALL or AML?

c. Suppose a new patient C has X level of .05 and Y level of 0.5. How would you classify the new patient? ALL or AML?

d. How confident are you with each of the three predicted classes?

e. How did you determine which class to choose for the patients?



Activity 2: Suppose you get one line to separate the data into two classes. Draw the line. Does it separate the patients perfectly?

Activity 3: Now suppose you get to draw a curve to separate the data into two classes. Draw the curve. Does it separate the patients perfectly?

In the plot above, there are two variables to predict a class. How is this similar to linear regression?

You can see that we can have more than two variables that can predict a class – it's just much more difficult to draw graphically in three or more dimensions.

Let's review linear regression for predicting a numerical value for antelope fawn count:

$$FawnCount = \beta_0 + \beta_1 * pop + \beta_2 * precipitation + \beta_3 * winterSeverity$$

Here, we found estimates for each of the betas to minimize the sum of squared error. FawnCount is a numerical response. In classification, the response needs to be a category or class, so we will need to somehow convert the response value to a category.

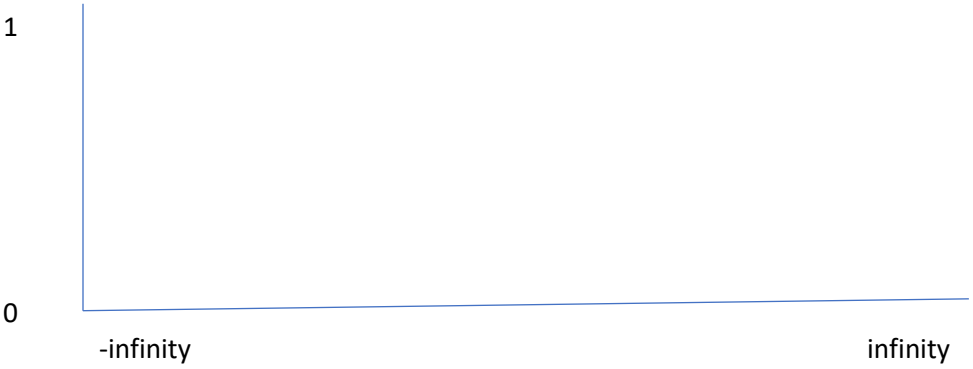
Logistic Regression: Classifying into one of two classes

We can perform a similar regression and output a probability of an item belonging to a class instead of a numerical response variable. For this to happen, the response needs to be a value between 0 and 1. Recall that probabilities must be between 0 and 1.

We can think of this as a two-step prediction process:

1. Get a response value Y based on the model
2. Convert the response value Y to a probability between 0 and 1

First, we need a function that can map any value from -infinity to infinity to [0, 1]. Draw a function below that does this.



Consider the following function:

$$P(y) = \frac{1}{1 + e^{-y}} = \frac{e^y}{e^y + 1}$$

Graph this function. Recall that e is the natural number.

When y is close to negative infinity, what is the value of P(y)? _____

When y is 0, what is the value of P(y)? _____

When y is close to infinity, what is the value of P(y)? _____

This is the function we will use to transform y to a probability.

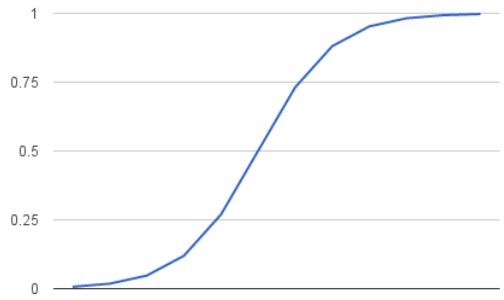


Figure of P(y): From Logistic Regression Tutorial by Jason Brownlee

Now, if we set the threshold for classes at 0.5, any returned probability that is greater than 0.5 is classified in the default class D. Otherwise, it is classified into the other class (ND = not default).

We can transform this equation through some algebra:

$$P(y) = \frac{1}{1+e^{-y}} = \frac{e^{y}}{e^{y}+1}$$

$$(e^{y}+1)P(y) = e^{y}$$

$$e^{y}P(y) + P(y) = e^{y}$$

$$P(y) + \frac{P(y)}{e^{y}} = 1$$

$$\frac{P(y)}{e^{y}} = 1 - P(y)$$

$$P(y) = e^{y}(1 - P(y))$$

$$\frac{P(y)}{1-P(y)} = e^{y}$$

This form of $P(y) \,/\, (1 - P(y))$ is called the **odds ratio**. Think of the probability of flipping heads with a fair coin. The probability of getting heads is 0.5 and the probability of getting tails is 1 – 0.5 = 0.5, so the **odds ratio** for getting heads is 1.

If the probability of winning a basketball game is 0.2, then the odds ratio for winning is 0.2/0.8 = 0.25.

When is the odds ratio large? _____

When is the odds ratio small? _____

What is the minimum odds ratio? _____

What is the maximum odds ratio? _____

So, now we can take the log of the odds ratio to get the relationship (**log odds ratio**):

$$\log\left(\frac{P(y)}{1-P(y)}\right) = y$$

Now, we just need a way to model the estimate for y. We can do this using linear parameter estimation and we will change the variable y to be x as a vector (to represent observation x).

$$\log\left(\frac{P(x = in\ class)}{1-P(x = in\ class)}\right) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \cdots \beta_k * x_k$$

So, now we think of the coefficients and variables linearly combining to result in the **log odds ratio**. Or we think about the coefficients and variables being multiplicative toward the odds ratio, as we see in the equation below.

The **logit** of p where p is a probability is $\mathbf{log}(\frac{\boldsymbol{p}}{\mathbf{1}-\boldsymbol{p}})$

Think about raising *e* to both sides of this equation. This gives us:

$$\frac{P(x)}{1-P(x)} = e^{\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \cdots \beta_k * x_k} = \prod_{j=0}^{k} e^{b_j x_j}$$

Suppose one of the coefficients for beta is 0.693. Then $e^{0.693}$ equals ~2. Suppose x is the variable for this coefficient. Then every unit increase in x **doubles the odds ratio**.

Example:
Suppose data is used to create a logistic regression classifier for the two classes: Passes_Exam and Fails_Exam. Data from students in the form of hours spent studying for the exam is collected, along with if they passed the exam. For example, student A spent 0.5 hours studying and failed the exam. Student B spent 1.75 hours studying and passed the exam.

A logistic regression is performed and the following coefficients are found:

Intercept = -4.0777
Hours = 1.5046

Log-odds of passing exam is:

$$Log\ odds\ of\ passing = -4.0777 + 1.5046 * Hours$$

Odds of passing exam:

$$Odds\ ratio\ of\ passing\ exam\ (logit) = e^{(-4.0777 + 1.5046 * Hours)}$$

Probability of passing exam from one of the first equations above:

$$Probability\ of\ passing\ exam = \frac{1}{(1 + e^{-(-4.0777+1.5046*Hours)})}$$

Hopefully, this helps you understand the various forms of these equations.

Practice:
Suppose a student studies 2 hours. What is her probability of passing the exam? _____
What is the log-odds of passing with 2 hours of studying? _____
What is the odds of passing with 2 hours of studying? _____

Suppose a student studies 4 hours. What is her probability of passing the exam? _____
What is the log-odds of passing with 4 hours of studying? _____
What is the odds of passing with 4 hours of studying? _____

What is the number of hours of studying that is the threshold for passing versus failing? ____
(Hint: set odds to 1 or set log odds to 0 or set the probability to 0.5)

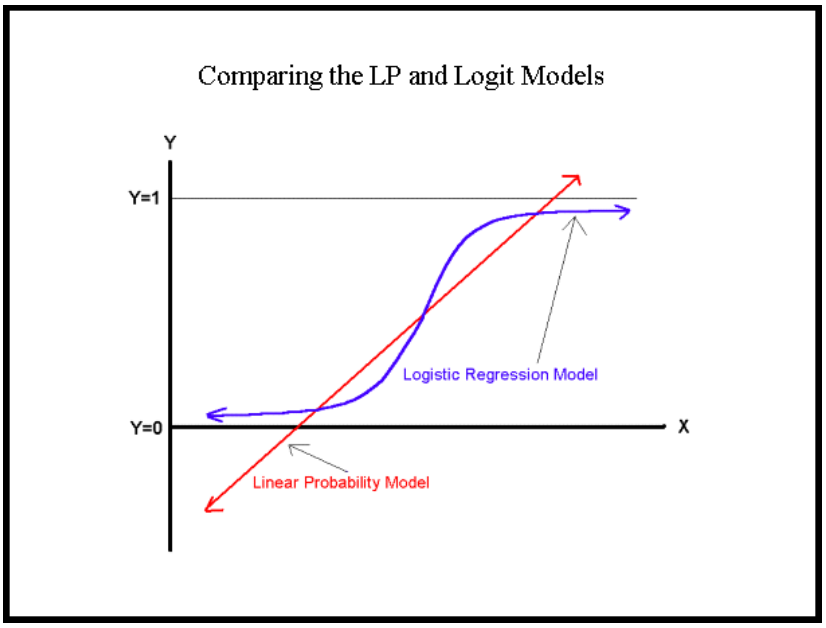
Calculating the coefficients
Finding the coefficients in logistic regression uses a process called maximizing likelihood. Below, you can see that we cannot use linear regression and do least squared error (that would give us the red line). We need coefficients to model the blue line since we are modeling the log odds. One can use maximum likelihood to estimate the coefficients. One optimization strategy is to use stochastic gradient descent to get estimates for the coefficients.

Here is the general strategy:

- Calculate predictions for observations using current coefficient estimates
- Calculate coefficients based on errors in the prediction

So, it is a back-and-forth process until the error drops to some target level or the process can run for a set number of iterations. There is a parameter, alpha, the determines the learning rate. It controls how much the coefficients can change during each run. You will do this in lab.

An example that uses stochastic gradient descent can be found here:
https://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning/



In R

R calculates a logistic model with the `glm` function. You will get a chance to create these in lab.

Example for Smarket data to predict Direction given six predictor variables:

```
glm.fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
data = Smarket, family = binomial)

summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Smarket)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.446  -1.203   1.065   1.145   1.326
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.126000   0.240736  -0.523    0.601
## Lag1        -0.073074   0.050167  -1.457    0.145
## Lag2        -0.042301   0.050086  -0.845    0.398
## Lag3         0.011085   0.049939   0.222    0.824
## Lag4         0.009359   0.049974   0.187    0.851
## Lag5         0.010313   0.049511   0.208    0.835
## Volume       0.135441   0.158360   0.855    0.392
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1731.2  on 1249  degrees of freedom
## Residual deviance: 1727.6  on 1243  degrees of freedom
```

What is the log-odds of the stock market going up given this output?

$$Log - odds\ of\ Up\ =\ -0.126 - 0.073 * lag1 - 0.042 * lag2\ +\ 0.011 * lag3\ +\ 0.009 * lag4\ +\ 0.010 * lag5\ +\ 0.135 * volume$$

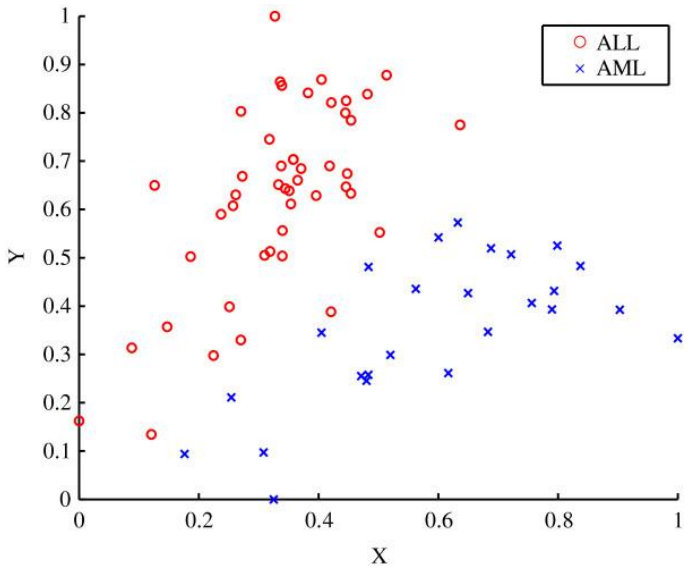
Deviance residuals are a bit like residuals we saw in linear regression, but instead of observed versus predicted, we have a measure of deviance from the classification based on the probability that is predicted.

Deviance of an observation i has two forms, depending on which class the observation is labeled:

$d_i = \sqrt{-2\ln(P_i)}\ where\ P_i\ is\ the\ predicted\ probability\ of\ default\ class, where\ observation\ is\ in\ default$
$d_i = -\sqrt{-2\ln(1-P_i)}\ where\ the\ observation\ is\ not\ default$

Binary logistic regression seeks to minimize the sum of squared deviance residuals.

Graphical Interpretation of Logistic Regression
Let's return to the cancer classification:



Draw that line to separate the two classes again. This line would be where the probability of the patient having ALL is 0.5. Take the point at x=.18, y=.65. This is far from that line, so the probability returned for this model would be closer to 1. Take the point at x=1, y=.32. This is far from the that line, so the probability returned for this model would be closer to 0.

CS 438: Classification: K Nearest Neighbors and Model Evaluation

Earlier, we looked at logistic regression to classify items into one class. There are techniques to do multinomial linear regression to build prediction models to classify items into more than one class. If there are C classes, then you create C models (one for each class) and the one that gives the highest probability in logistic regression is chosen as the predicted class.

Another approach is the do classification by finding the nearest set of K neighbors and having them take a vote, with the highest vote total winning. Of course, there can be ties if K = 3 and C = 3, there could be one vote for each class. In the case of ties, there are a couple of ways to handle them: flip a coin on a tie vote to break the tie **OR** choose the class of the closest neighbor.

Class Activity: Get up and go to a location in the room (height x distance_to_campus). Classify yourself as on-campus or off-campus. Now let's add a new data point. Use K = 7 to classify the new data point.

Add another point. Use K = 1.

Try another point. Use K = 5.

KNN probability:
Voting via math looks like this:

$$\Pr(Y = j \mid X = x_0) = \frac{1}{k} \sum_{i \in Neighbors\_k} I(y_i = j)$$

Here, k is the number of closest neighbors under consideration.

Once the probabilities are calculated (these are just fractions of each class), the class with the highest probability is chosen. With a tie, flip a coin or choose the closest point.

Distance:
There must be a notion of distance for "nearest" neighbor. A common distance metric is **Euclidean distance**.

$$D(x_1, x_2) = \sqrt{\sum (x_{i1} - x_{i2})^2}$$

This is the "tape measure" distance between the two points.

What are other way we might measure distance between two points?

Cosine similarity: think of data as vectors, two vectors that are aligned have a cosine value of 1, two vectors going in the opposite direction have a cosine value of -1

Manhatten distance: think NYC, how many blocks you have to walk to get from one intersection to another (up/down and left/right) movements.

$$D(x_1, x_2) = \sum |x_{i1} - x_{i2}|$$

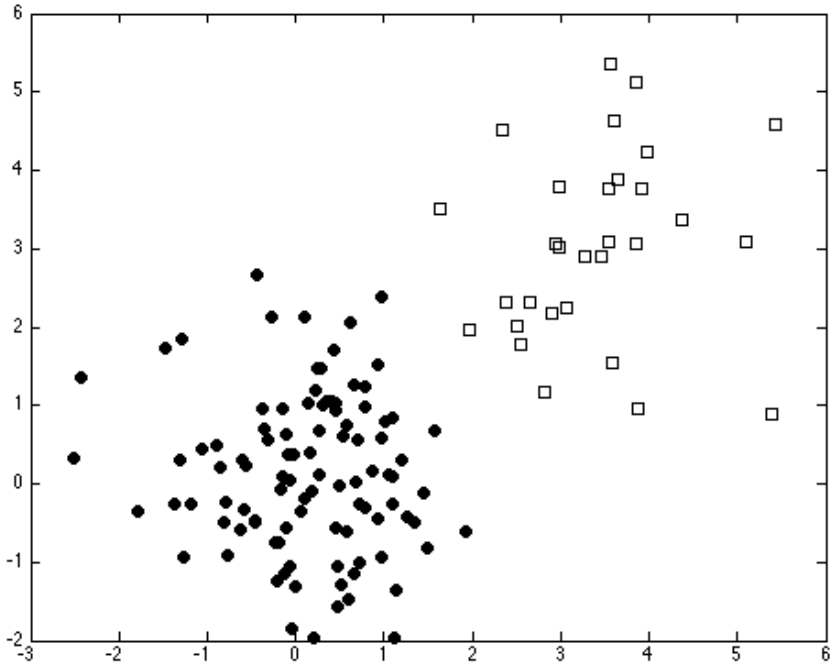
Hamming distance: how many of the same position with the vector are equal, works for data such as strings like "Tammy" and "Sammy", which have a hamming distance of 1.

Normalized distance: Take data and for each component, convert values to standard normal (mean 0 and standard deviation 1). Then, use Euclidean distance.

Why would normalized distance be preferred over regular Euclidean distance?

_____

Activity 1: Classifying with KNN
Suppose this is your dataset:



1. Suppose a new data observation is (2, 2).
   K = 1. Euclidean Distance.

   Is it a square or a circle? _____

2. Suppose a new data observation is (1.75, 1).
   K = 1. Euclidean Distance.

   Is it a square or a circle? _____

3. Suppose a new data observation is (1.5, 2).
   K = 3. Euclidean Distance.

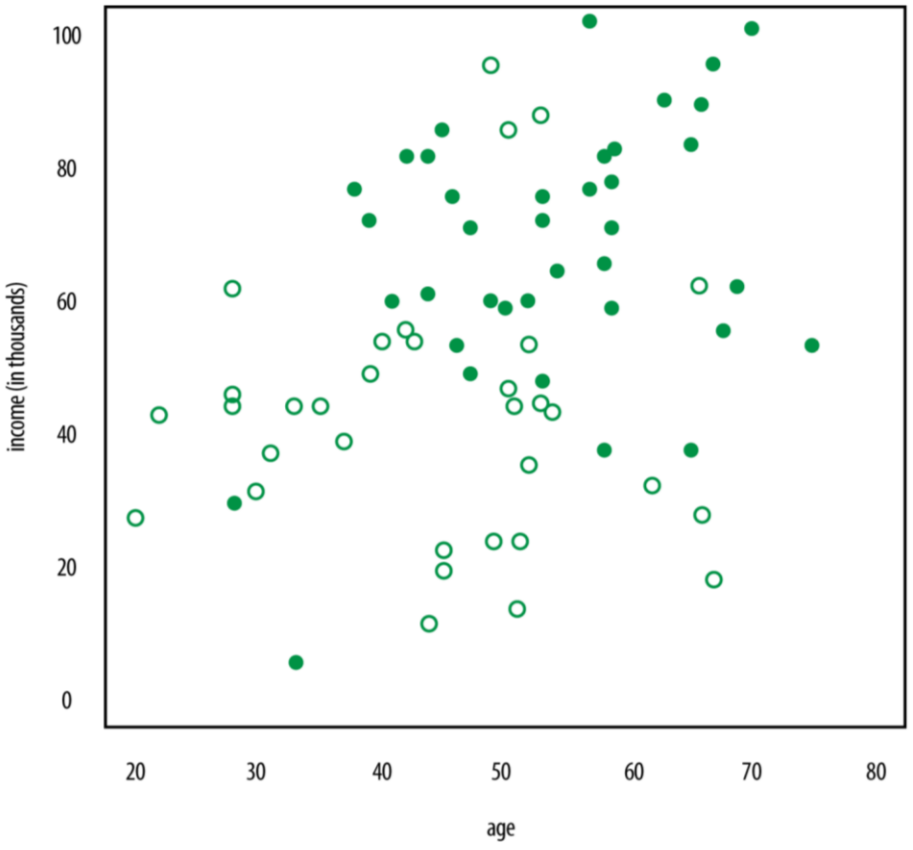
   Is it a square or a circle? _____

4. In the above figure, draw the boundary curve for classifying squares and circles based on KNN with K = 1.

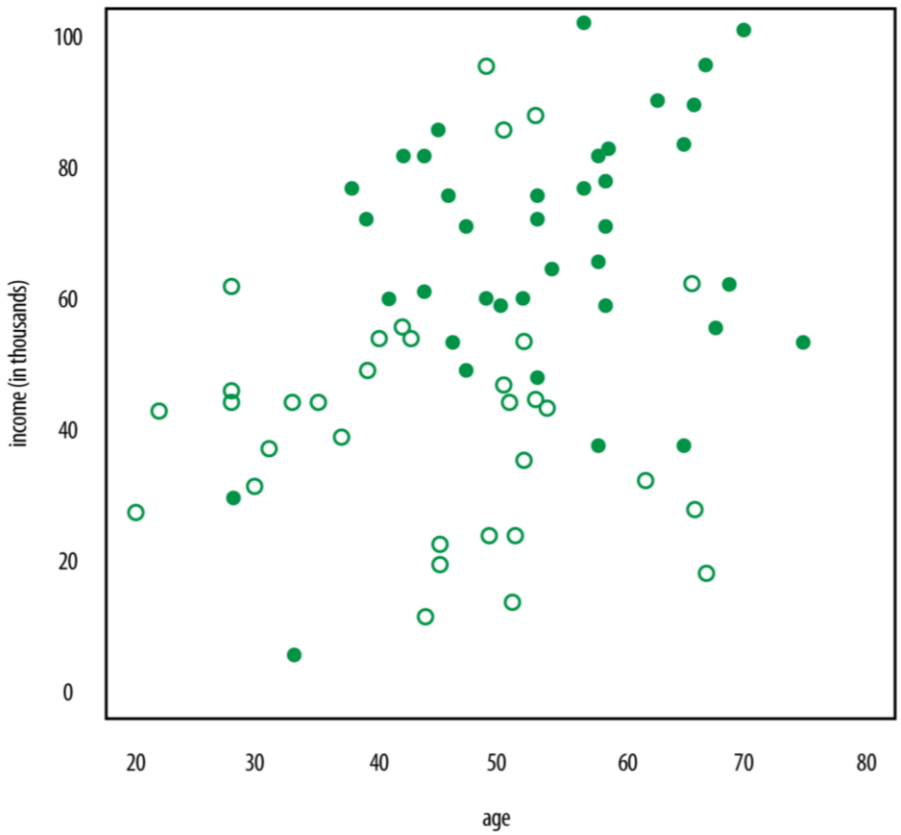
Activity 2: Creating the KNN boundaries

Suppose you are using the dataset below. Age is the age of a person and income is their annual income. Green colored in dots are people that do not default on a loan. Unfilled circles represent people who defaulted on the loan.

Use K = 1 to build the boundary for classification and use Euclidean distance. Lightly shade the area of the plot for which new observations would be classified as filled in circles.



Use k = 3 for KNN and Euclidean distance. Build the classifier boundaries. Lightly shade the area of the plot for which new observations would be classified as filled in circles.



Discussion:

1. Which takes longer to compute on a new observation? KNN or logistic regression?

2. Which would do better if the classes cluster in islands? KNN or logistic regression?

3. Does K = 1 or K = 3 have less **training error** (mistakes in predicted classes)?

4. Does KNN use more or less computer memory than logistic regression?

5. What do the boundaries look like for KNN versus logistic regression?

6. How would you decide which value of K makes the best classifier for a dataset?

Evaluation

There are many more classification methods than KNN and logistic regression, but now you have already seen there are many ways to create classifiers, even with just two models: which features from the dataset to keep in the model? (which ones in logistic regression are significant), which K to use in KNN? Which distance metric to use in KNN?

Cross-validation is a common technique to evaluate models. As in lab, you create two subsets of your original dataset:

Training set: data observations used to construct the model
Validation set: data observations used to test the model by using the model to make predictions and seeing if the predicted class is the observed class

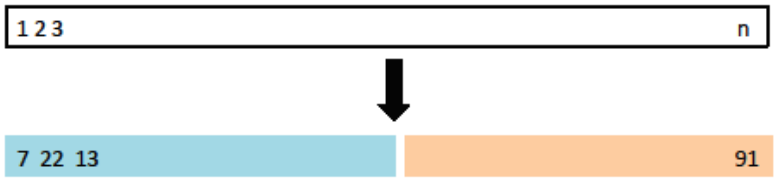


Figure 5.1 from STS; figures below also from STS

Blue is the training set. Orange is the validation set. The model is trained on the blue observations. The model is validated on the orange observations.

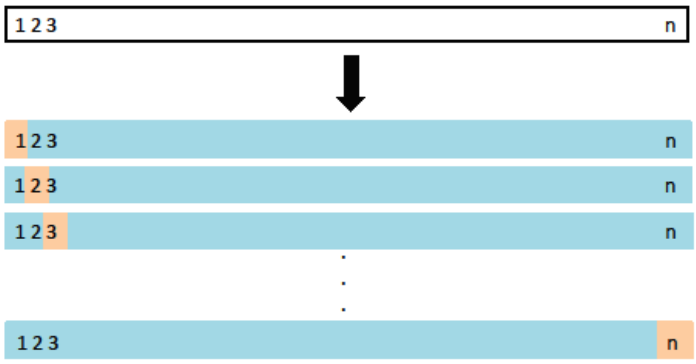
80/20 is a good starting place (80% training, 20% validation)

But, what if we used the orange set to train and the blue set to validate? We may (likely) get a different model.

What about overfitting? The model may get great training error and terrible validation error.

Leave One Out Cross-Validation (LOOCV)

We could instead make multiple models, leaving a different subset of the data as the validation set.

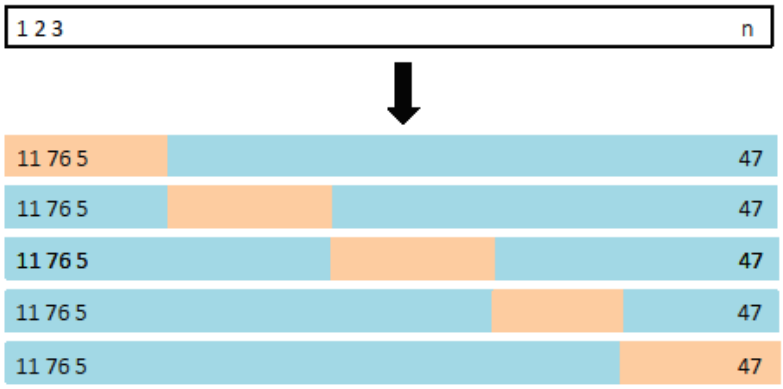


In LOOCV, there are n models created and the test error is the average of the test error across all n models.

k-Fold Cross-Validation (LOOCV)

We can create 10-fold cross-validation, in which the dataset is divided into 10 equal parts. 9/10 of the data is used to train each model. Brings down the number of models that must be built to 10 instead of n.

5-fold cross-validation is also a common technique for evaluation. Here is a picture of how this one would work: randomly split n observations into 5 disjoint sets.



Discussion:

There is a **bias-variance** tradeoff in model evaluation. **Bias** is the error caused by erroneous assumptions in the learning algorithm. High bias can cause a model to miss relevant relationships between features and classes. Bias is what causes underfitting.

Variance is the error from sensitivity to small fluctuations in the dataset. High variance causes the model to model noise in the training data. Variance is what causes overfitting.

1. Do you think LOOVC or k-fold cross-validation produces more bias (underfitting)?

2. Do you think LOOVC or k-fold cross-validation produces more variance (overfitting)?

Example:

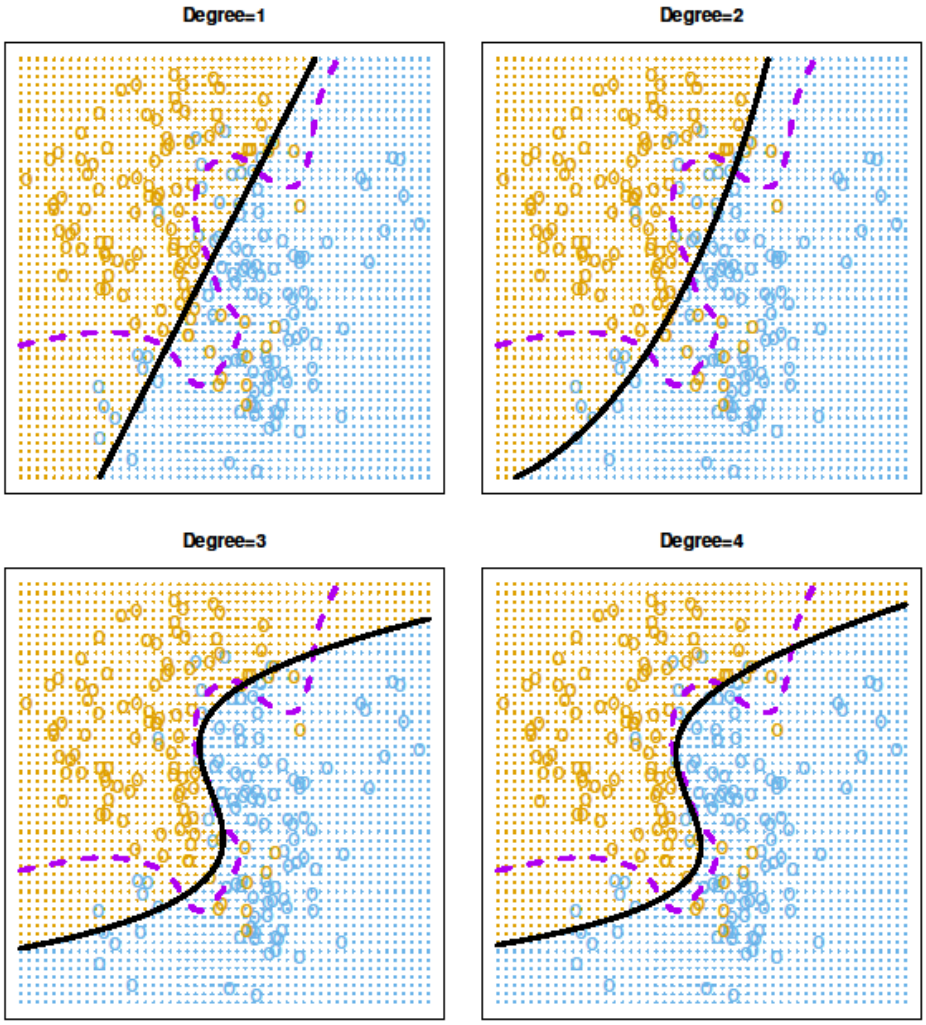
Example of logistic classification. Below are 4 models (each with a different degree of polynomial for logistic regression). Just like with linear regression, we can create a polynomial model, so the function looks like this:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_m X^m$$

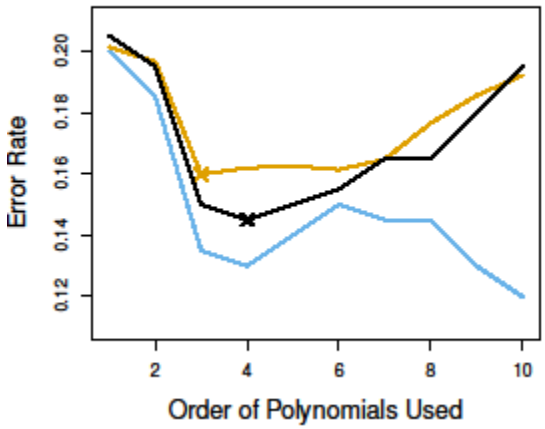
This can also be applied to multidimensional data (two or more predictor variables in the observations) and polynomials can be created. So, for a dataset with 2 predictor variables (X, Y), we could have a quadratic function of both:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 Y + \beta_4 Y^2$$

Below are the boundaries created for four different models (degree 1, degree 2, degree 3, and degree 4):



The dashed purple line is the optimal classifier. The black line shows the boundary created by the logistic regression model. We can look at the test error and training error to get the best model of the four (one that does not overfit too much or underfit too much):



Blue line (bottom line) is training error
Black line (middle line) is 10-fold CV error
Orange line (top line) is test error [available because data was simulated and have optimal classifier, but not usually available for datasets]

Note that the figure fits up to degree 10 polynomials for the dataset shown in the previous figure. The x's show the minimum value along the orange and black lines. Fourth order polynomial would be the best of this set.

Confusion Table and Equations

As you will see in lab, a confusion table shows 4 quadrants:

	Event	No_Event
Predicted Event	A	B
Predicted No_Event	C	D

$$Sensitivity = Recall = \frac{A}{A+C}$$

$$Specificity = \frac{D}{B+D}$$

$$Precision = \frac{A}{A+B}$$

$$Accuracy = \frac{Number\ Correct}{Total\ Number\ of\ predictions} = \frac{(A+D)}{(A+B+C+D)} = \frac{TP+TN}{(TP+TN+FP+FN)}$$

Practice: Calculate all four terms for the following confusion table:

	Real Yes	Real No
Predicted Yes	35	20
Predicted No	15	30

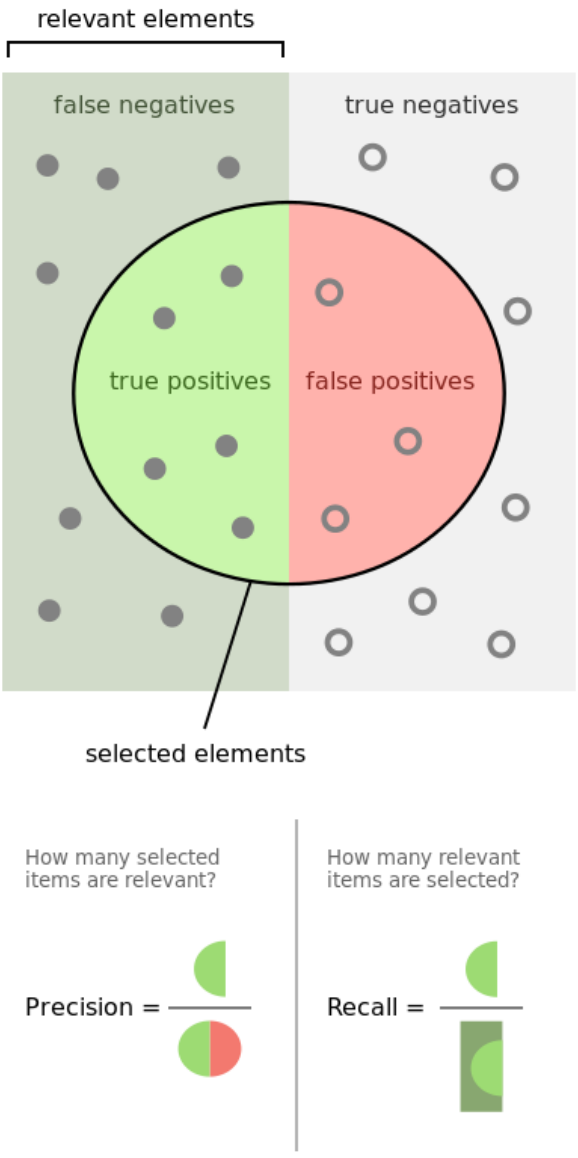
Sensitivity/Recall = _____
Specificity = _____
Precision = _____
Accuracy = _____

A visual reminder of precision and recall. What would be the confusion matrix for this dataset?

TP = True Positive
FP = False Positive
FN = False Negative
TN = True Negative

A=TP= _____ B=FP= _____

C=FN= _____ D=TN= _____



CS 438: Classification: Bayes Classifier

Earlier, we looked at logistic regression and KNN to create models to classify observations into two or more classes. A third classification method is called the Bayes Classifier (or Naïve Bayes).

First, we will look at creating a Bayes Classifier for categorical data (predictor variables are categories rather than numerical, continuous data). I think this will motivate better how the classifier works.

Example based on Jason Brownlees' tutorial:

Suppose you want to classify a new day as "go-out" or "stay-home". The decision is based on the weather and the car working properly. For weather, the categories are "sunny" and "rainy". For the car, the categories are "working" and "broken".

We will convert the categories into numerical data as follows:

Variable	**1**	**0**
Weather	Sunny = 1	Rainy = 0
Car	Working = 1	Broken = 0
Class	Go-out = 1	Stay-home = 0

Assume we have the following data

Weather	Car	Class
1	1	1
0	0	1
1	1	1
1	1	1
1	1	1
0	0	0
0	0	0
1	1	0
1	0	0
0	0	0

Just looking at the data alone, what pattern(s) do you see?

Now, let's consider classification probabilities, just like we did for logistic regression.

$$P(class = Go\_out) = \frac{Count(class = Go\_out)}{Total\_Observations}$$

$$P(class = Stay\_home) = \frac{Count(class = Stay\_home)}{Total\_Observations}$$

In our example above, since there are 5 observations for each class, the probability for each class is 0.5.

Now, let's look at conditional probabilities for weather based on the class.

$$P(weather = sunny \mid class = Go\_out)$$

$$P(weather = rainy \mid class = Go\_out)$$

$$P(weather = sunny \mid class = Stay\_home)$$

$$P(weather = rainy \mid class = Stay\_home)$$

If we had some way to calculate these probabilities given our data… How might we do this? Seems backwards – going out is predicted from the weather and not the other way around.

Bayes' Theorem:

Suppose A and B are events and the probability of B is non-zero.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A|B) is the conditional probability of A happening given that B is true.
P(B|A) is the conditional probability of B happening given that A is true.
P(A) and P(B) are the probabilities of observing A and B independently of each other.

Note that P(A|B) is also referred to as the **posterior** probability (degree of belief of A given B).
Note that P(A) is also referred to as the **prior** probability (initial degree of belief of A).
The quotient (P(B|A)/P(B)) is known as the **support** B provides for A.

Proof:
Assume A and B are events, each with at least one observation so the probabilities are non-zero. From the definition of conditional probability, we know that:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Also,

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

where P(A ∩ B) is the joint probability of both A and B being true. Since intersection is symmetric, the P(A ∩ B) is equal to P(B ∩ A).

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

from combining the first two equations.

Thus, we can isolate P(A|B), so that:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Now, we have a way to calculate the probabilities for weather and car, given our classifications.

$$P(weather = sunny \mid class = Go\_out)$$
$$P(weather = rainy \mid class = Go\_out)$$
$$P(weather = sunny \mid class = Stay\_home)$$
$$P(weather = rainy \mid class = Stay\_home)$$

P(B|A) calculations:

P(c = g | w = s) = 0.67
P(c = s | w = s) = 0.33
P(c = g | w = r) = 0.25
P(c = s | w = r) = 0.75

P(A|B) calculations using theorem:

P(w = s | c = g) = 0.67 * 0.6 / 0.5 = 0.8
P(w = r | c = g) = 0.25 * 0.4 / 0.5 = 0.2
P(w = s | c = s) = 0.33 * 0.6 / 0.5 = 0.4
P(w = r | c = s) = 0.75 * 0.4 / 0.5 = 0.6

We can do the same sets of calculations for the car variable and we get:

Car:

P(car = w | c = g) = 0.8
P(car = b | c = g) = 0.2
P(car = w | c = s) = 0.2
P(car = b | c = s) = 0.8

To Classify A New Observation

We simply multiply the probabilities together (here we are assuming the variables in the model are independent) to determine which class yields a higher number.

Suppose the new observation is weather = sunny and car = working.

$$P(c = Go_{out}) = P(w = s \mid c = Go_{out}) * P(car = w \mid c = Go_{out}) * P(c = Go_{out})$$

$$P(c = Go_{out}) = 0.8 * 0.8 * 0.5 = 0.32$$

$$P(c = Stay\_home) = P(w = s \mid c = Stay\_home) * P(car = w \mid c = Stay\_home) * P(c = Stay\_home)$$

$$(c = Stay\_home) = 0.4 * 0.2 * \ 0.5 = 0.04$$

Since 0.32 exceeds 0.04, the class is Go_out for the new observation.

Activity: Complete the calculations for the other three combinations of Weather and Car.

Outcome	**Weather**	**Car**	**Probability**
Go Out	Sunny	Working	0.32
Stay Home	Sunny	Working	0.04
Go Out	Rainy	Broken	
Stay Home	Rainy	Broken	
Go Out	Sunny	Broken	
Stay Home	Sunny	Broken	
Go Out	Rainy	Working	
Stay Home	Rainy	Working	

1. Suppose the new observation is weather = rainy and car = broken. What is the probability calculated for Bayes for the class Go_out and for Stay_home?

P(c = Go_out) = _____

P(c = Stay_home) = _____

2. Suppose the new observation is weather = sunny and car = broken. What is the probability calculated for Bayes for the class Go_out and for Stay_home?

P(c = Go_out) = _____

P(c = Stay_home) = _____

3. Suppose the new observation is weather = rainy and car = working. What is the probability calculated for Bayes for the class Go_out and for Stay_home?

P(c = Go_out) = _____

P(c = Stay_home) = _____

Note: there wasn't even an observation in the dataset with these values, yet we can still predict its classification.

Add up all the probabilities for all 8 possibilities: _____

Extension to Continuous-Valued Variables

The example above calculates the probabilities via counts or frequencies. With continuous-valued data (such as height of a person), we use the standard normal probability distribution, just as in EGR 361.

Suppose a classifier decides if someone is male or female based on the following variables:

- Height
- Weight
- Shoe Size

Based on the dataset, we find the mean and variance of each variable for men in the dataset and for women in the dataset. Suppose this is the what we find:

Gender	Height mean	Height variance	Weight mean	Weight variance	Shoe Mean	Shoe Variance
Male	5.855	.0350	176.25	122.9	11.25	.9167
Female	5.418	.0972	132.5	558.3	7.5	1.667

Suppose we want to classify a 6-foot-person, 130-pounds with shoe size 8 as male or female:

P(male) = N(6, 5.855, sqrt(0.035)) * N(130, 176.25, sqrt(122.9)) * N(8, 11.25, sqrt(.9167)) * 0.5
$=6.120 * 10^{-9}$

P(female) = N(6, 5.418, sqrt(0.0972)) * N(130, 132.5, sqrt(558.3)) * N(8, 7.5, sqrt(1.667)) * 0.5
$= 5.378 * 10^{-4}$

> Here, N(A, B, C) means the probability density function for a Normal distribution with mean B and standard deviation C.

Since the probability for being female is higher than for being male with these numbers, the classification is female.

CS 438: Decision Trees

What decisions have you made today?

Which of those decisions were binary? (yes/no, go/stay, etc.)

Class Activity:
Suppose you want to build a model to classify students at UP as those who participate in music ensembles or not.

a. Do you participate in a UP music ensemble (yes or no)? _____

b. Suppose we know the number of credits a student is taking, their gender identity, and if they commute or are residential.

For yourself, how many credits are you taking this semester? _____

What is your gender identity (female, male, non-binary)? _____

Are you a commuter or are you residential? _____

OK, we will try to build a tree classifier for the set of students in this class. We want to know which of the three variables (credits, gender, or commuter) is most meaningful. How would you define meaningful in terms of giving a good classification?

OK, let's see what happens.
1. Get up and go to two sides of the room for < 16 credits on one side and >= 16 credits on the other.

How many taking < 16 credits are in a musical ensemble? _____ Size of group? ____

How many taking >= 16 credits are in a musical ensemble? _____ Size of group? ____

2. Now, divide yourselves by gender.

How many female are in a musical ensemble? _____ Size of group? _____

How many male are in a musical ensemble? _____ Size of group? _____

How many non-binary are in a musical ensemble? _____ Size of group? _____

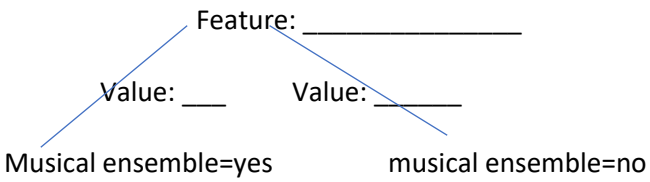
3. Now, divide yourselves by commuter.

How many commuters are in a musical ensemble? _____ Size of group? _____

How many residential students are in a musical ensemble? _____ Size of group? _____

4. Which of the three features/variables is most meaningful for classification? _____
(Which creates most homogeneous subgroups?)

OK, let's build a tree with that feature at the top:



How well did this single-feature classifier do?

Activity 1: Work in a small group to continue building this tree with other features until you have what you think is a good classifier. Then, we will compare them.

Build decision tree here:

Suppose another student taking 18 credits, is a commuter, and is female is given to your classifier. How does your tree classify the student? _____

Activity 2: Here is an example of a large decision tree to classify flowers (irises). Left branches are true. Right branches are false. Study the tree to see which features are used in the model and determine the possible classes.

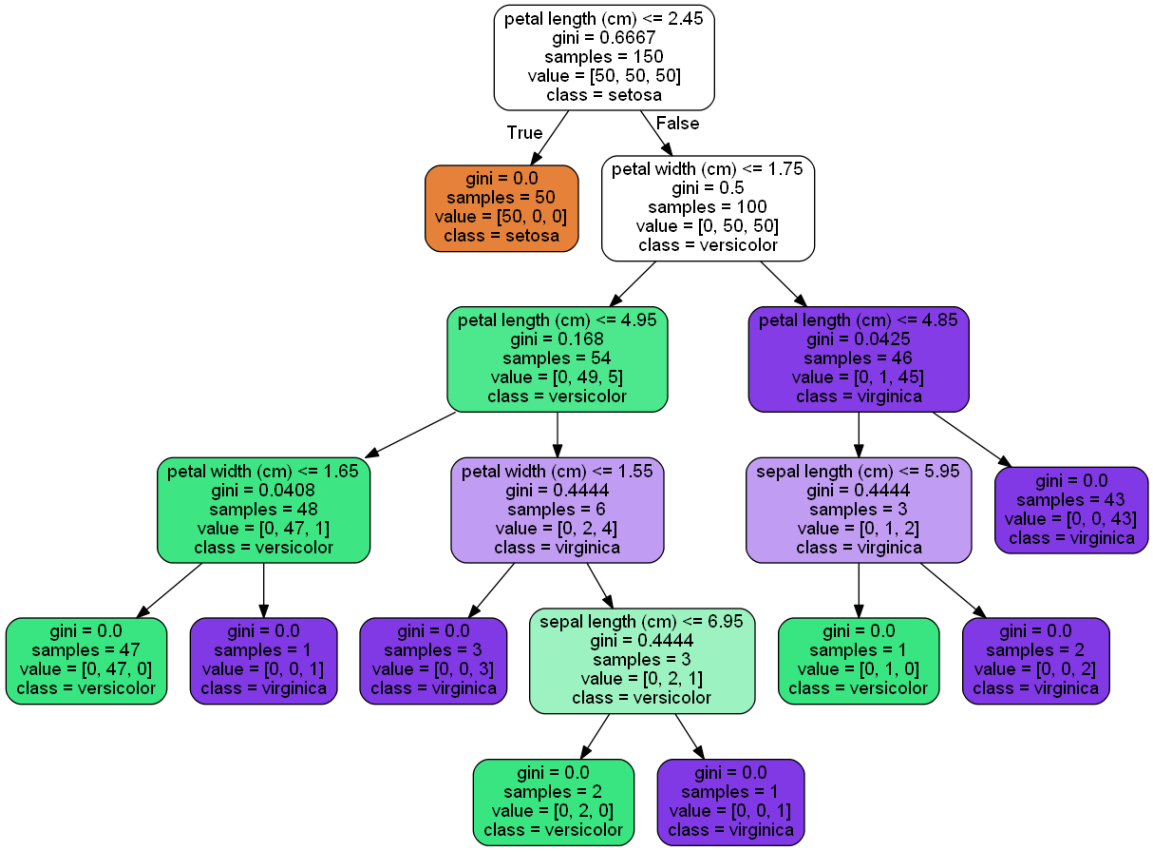


Figure from https://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html

1. What features are used to classify observations?

2. What property is true for all leaves in this tree?

3. How many samples are in the training set?

4. Do features get used more than once along a path to a leaf?

5. Which class has the shortest petal lengths?

6. Suppose a new iris is classified with this tree. The iris has petal length 3.5 cm, petal width 2.0 cm, and sepal length of 6.0 cm. What class would this decision tree predict? _____

7. Suppose an iris as petal length 5.0 cm, petal width 1.25 cm, and sepal length of 4.0 cm. What class would this decision tree predict? _____

Types of Decision Trees

If the response variable for the decision tree is a *category or class*, it is called a **classification tree**.

Examples:
The tree predicts if a customer will make a purchase.
The tree predicts type of disease.

If the response variable for the decision tree is *numerical* (similar to linear regression), it is called a **regression tree**.

Examples: The tree predicts the price of a house.
The tree predicts the recovery time after an injury.

Name an example response for a classification tree: _____

Name an example response for a regression tree: _____

Variables (Features):
Variables (features) can be numerical and/or categorical. An example of a numerical variable is a person's age. An example of a categorical variable is their occupation. Decision trees can be built for both types of variables.

For numerical variables, the split is determined based on minimizing variance of the two "split" groups. In the iris tree example above, you will see the first decision is petal length <= 2.45. The value 2.45 was determined the best threshold because it split the data into a set of 50 that are all setosa and the other 100 are versicolor or virginica.

For categorical variables, the split is by category. Some tree-building algorithms will always split binary, so if there are three categories (A, B, C), then the first split may keep A/B together and separate C, and then split A from B in the next level down. Some tree-building algorithms will create non-binary trees and make three or more children for three or more values per category.

Advantages and Limitations of Decision Trees

Activity 3: In small groups, discuss the advantages and limitations of decision trees. Think about fitting to training data, explaining the model to others, impacts of outliers, and the assumptions about types of data for variables and the distributions of data for variables.

Advantages:

1.

2.

3.

Limitations:

1.

2.

3.

Tree Construction

Trees are built in a greedy fashion, choosing the best "split" at every node. The best split is determined by which feature provides the best homogeneity among the children. This value of homogeneity is usually quantified via the gini impurity, entropy, information gain, and/or variance reduction. These metrics are defined later. They all create a measurement of how "pure" the sub-children nodes would be based on the feature selection.

The recursive greedy algorithm is as follows:

```
Create Root_Node representing the entire dataset
Build_tree(Root_Node)

Build_tree(Node):
If stopping condition is met:
      Create Leaf_Node (terminal node) with classification
Else:
      Create Decision_Node, split on feature to maximize homogeneity among children
      For all children d of Decision_Node, Build_tree(d)
```

Here is a graphical view of the types of nodes that are made:

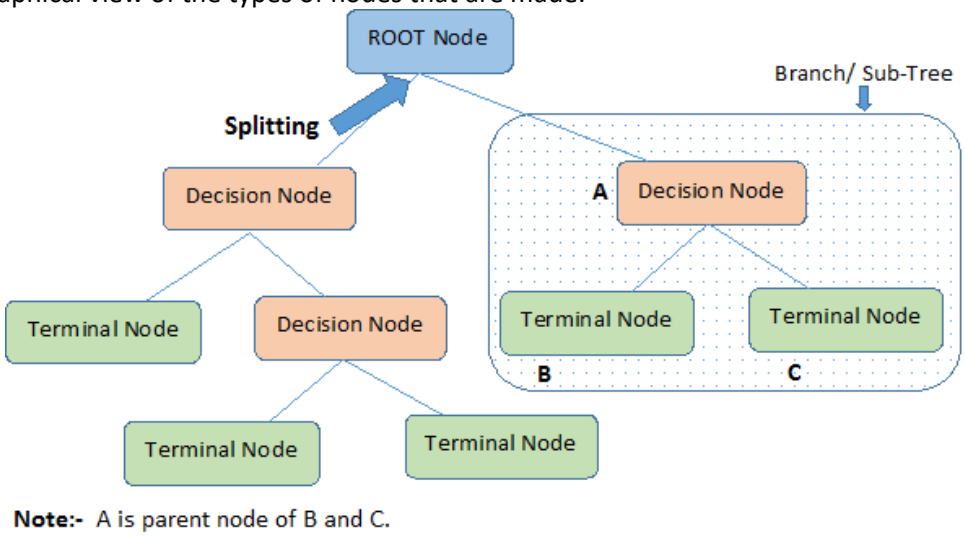


Figure from https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

We need to define the **stopping condition**. Here are some possibilities:

- Leaf is pure (all observations are in one class)
- Tree has maximum depth provided at start

- Leaf has fewer than X observations
- Run out of categorical variable splits

Which of these stopping conditions could make really tall trees? _____

Would really tall trees be subject to overfitting or underfitting? _____

One strategy for tree construction to reduce overfitting is to build tall trees and then **prune** some nodes to make the tree shorter. Other strategies include bagging, boosting, and random forests to create a set of trees that each "vote" for the overall classification to reduce overfitting.

Choosing the Best Node to Split

We will look at metrics for nodes representing categorical data first. These are gini and entropy.

Gini Impurity:
This is a fairly straightforward metric:

$$p_i = fraction\ of\ items\ labeled\ as\ class\ i\ in\ the\ set, J = number\ of\ classes$$

$$GINI(feature) = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k = \sum_{i=1}^{J} p_i\,(1 - p_i) = 1 - \sum_{i=1}^{J} p_i^2$$

1. If all observations for a feature are in the same class, what is the GINI value? ____

2. If J = 2 and half of the observations are in one class and half are in the other using the feature, what is the GINI value? _____

3. Suppose there are 1 million classes, each with one observation in the class (huge impurity). What is the GINI value? _____

Now that we have a metric for splitting, we use that metric to calculate **information gain** using that metric. Information gain is a metric that tells us how much better the children of the nodes explain the classification. At each split, we choose the feature that maximizes information gain. The information gain tells us how important that feature is in the model.

$$InfoGain(feature) = GINI(parent) - Weighted\_ave(GINI(children))$$

Apply GINI to the example data:
Suppose there are 30 students in the dataset and we are trying to classify if the students play soccer.

	Plays Soccer	Does Not Play Soccer
Sample	15 of 30	15 of 30
Gender: Male	13 of 20	7 of 20
Gender: Female	2 of 10	8 of 10
Class: Senior	9 of 16	7 of 16
Class: Junior	6 of 14	8 of 14

Attribute: Gender

$$GINI(Gender = Male) = 1 - [(0.65) * (0.65) + (0.35) * (0.35)] = 0.455$$

$$GINI(Gender = Female) = 1 - [(0.2) * (0.2) + (0.8) * (0.8)] = 0.32$$

$$GINI(Root) = 1 - [(0.5) * (0.5) + (0.5) * (0.5)] = 0.5$$

$$InfoGain(Gender) = 0.5 - \left[\left(\frac{10}{30}\right)(.32) + \left(\frac{20}{30}\right)(.455)\right] = 0.09$$

Attribute: Class

$$GINI(Class = Junior) = 1 - [(0.4285) * (0.4285) + (.5714) * (.5714)] = 0.489$$

$$GINI(Class = Senior) = 1 - [(0.5625) * (0.5625) + (.4375) * (.4375)] = 0.492$$

$$InfoGain(Class) = 0.5 - \left[\left(\frac{14}{30}\right)(.489) + \left(\frac{16}{30}\right)(.492)\right] = 0.0094$$

Since InfoGain for Gender > InfoGain for Class, Gender is chosen as the category for the first split.

```
                    Root
Gender=female              Gender=male
```

Now, we have two nodes to further split. Let's look at the `Gender=female` node. In the dataset, there are 10 females. Suppose of those 10, we have the following:

- 1 is a junior who plays soccer
- 1 is a senior who plays soccer
- 5 are juniors who do not play soccer
- 3 are seniors who do not play soccer

This is our only feature that can separate, since this group contains only women. It would end up classifying both leaves as "Do not play soccer". If our dataset had more features, then we would have more choices at each recursive split.

Note: Some implementations speed up the $InfoGain$ metric by removing the $GINI(Root)$ calculation and instead maximize $Weighted\_Ave(GINI(Children))$.

Note: The CART implementation splits using the GINI metric. CART = Classification and Regression Trees (acronym used in data science).

Entropy:
Another measure for splitting is calculated based on entropy. Entropy is also a measure of the impurity of an attribute.

$$p_i = fraction\ of\ items\ labeled\ as\ class\ i\ in\ the\ set, J = number\ of\ classes$$

$$H = Entropy(feature) = -\sum_{i=1}^{J} p_i * \lg(p_i)$$

1. Suppose all observations for the feature are in the class. What is the entropy? _____

2. Suppose there are two classes, where the feature splits it into two groups of the same size, so the fraction for each class is 0.5. What is the entropy? _____

You can see that entropy and GINI have similar behavior, just with different maximum values. As before, we have the same calculation for information gain, but use the entropy metric instead of the GINI metric.

$$InfoGain(feature) = H(parent) - Weighted\_ave(H(children))$$

Dataset:

	Plays Soccer	Does Not Play Soccer
Sample	15 of 30	15 of 30
Gender: Male	13 of 20	7 of 20
Gender: Female	2 of 10	8 of 10
Class: Senior	9 of 16	7 of 16
Class: Junior	6 of 14	8 of 14

Attribute: Gender

$$H(Parent) = -\left(\frac{15}{30}\right)\lg\left(\frac{15}{30}\right) - \left(\frac{15}{30}\right)\lg\left(\frac{15}{30}\right) = 1.0$$

$$H(Gender = Male) = -\left(\frac{13}{20}\right)\lg\left(\frac{13}{20}\right) - \left(\frac{7}{20}\right)\lg\left(\frac{7}{20}\right) = .934$$

$$H(Gender = Female) = -\left(\frac{2}{10}\right)\lg\left(\frac{2}{10}\right) - \left(\frac{8}{10}\right)\lg\left(\frac{8}{10}\right) = .722$$

Information gain for this feature is $1.0 - [(20/30) * (.934) + (10/30) * (.722)] = .137$

Attribute: Class

$$H(Class = Senior) = -\left(\frac{9}{16}\right)\lg\left(\frac{9}{16}\right) - \left(\frac{7}{16}\right)\lg\left(\frac{7}{16}\right) = .988$$

$$H(Class = Junior) = -\left(\frac{6}{14}\right)\lg\left(\frac{6}{14}\right) - \left(\frac{8}{14}\right)\lg\left(\frac{8}{14}\right) = .985$$

Information gain for this feature is $1.0 - [(16/30) * (.988) + (14/30) * (.985)] = .0134$

Thus, between the two features, Gender has higher information gain than Class, so we choose Gender for splitting first.

Note: In practice, GINI and entropy create the same tree most of the time. Because log takes some time to compute, entropy is a slightly slower calculation.

Note: ID3 (Iterative Dichotomiser 3) uses entropy for the splitting metric.

Activity:

In small groups, perform the calculations to decide which node to split on using entropy and information gain.

Here is the dataset. Play Game is the class you are deciding. The four features are Outlook, Temperature, Humidity, and Windy.

Outlook	**Temperature**	**Humidity**	**Windy**	**Play Game?**
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Each group will calculate the information gain of a different feature and then we will compare.

First, we calculate the entropy of the root node for Play_Game.

9 of 14 are Yes in the dataset

5 of 14 are No in the dataset

$$H(Root) = -\left(\frac{9}{14}\right)\lg\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right)\lg\left(\frac{5}{14}\right) = 0.94$$

Your group is assigned the feature: _____

For example: Outlook and Temperature have three categories, so there will be three children. Humidity has two categories. Windy has two categories.

Entropy(Feature=Category1) = _____

Entropy(Feature=Category2) = _____

Entropy(Feature=Category3 (if needed)) = _____

Weighted Average of Entropy of Children = _____

InformationGain(Feature) = _____

If you finish, try calculating the InfoGain for a different feature. Of the groups in the class, which has the best information gain?

CS 438: Decision Trees with Numerical Features And Regression Trees

Decision trees can be created for categorical features and for numeric features. First, we will look at building classifiers with numeric features.

Suppose the predictors are X1 and X2 for a dataset and we are outputting class blue or class red. Below is a visual about how the tree corresponds to the dataset:

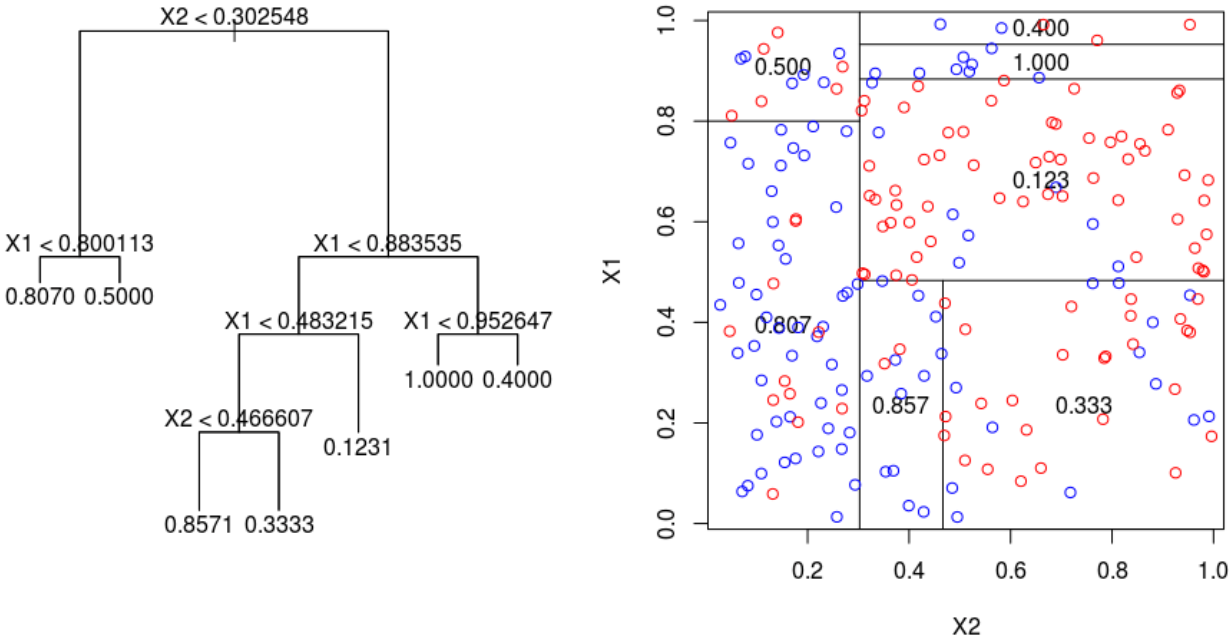


Figure from: https://www.datacamp.com/community/tutorials/decision-trees-R

How do we find the first split? Look for the line that separates the dataset into two parts (line that goes across the box vertically or horizontally).

Where is it? _____

OK, that becomes the root node.

Now, it is just a matter of seeing the two boxes as two datasets for splitting. Let's focus on the data where X2 < 0.3. In this box, there are two subsets, one above X1 > 0.8 and the other below X1 < 0.8. This is the next split, as you can see above in the tree.

The tree and dataset also show the classification probability for the leaves assigning to the blue class.

Does the classification of the scatterplot and the boxes make sense now?

Deciding Splits for Continuous Data

For continuous data, how do we decide where the splitting point is? You iterate through the possible threshold values and use each for a potential split. Then, use one of the metrics (gini, entropy, or deviance residuals) to decide which threshold split is best.

For example, if the dataset for variable age contains:

2.1
2.8
3.5
8.0
10.0
20.0
50.0
51.0

Then, the possible splits are between each value, such as:

2.45
3.15
5.75
9.0
15.0
35.0
50.5

Regression Trees

Decision trees can be created for numerical responses as well as categories. In this case, the tree is called a regression tree.

The process is the same as building a decision tree, where the splits are determined by the best feature. Since the output is numerical, we can use residual sum of squares as the metric to minimize. So, we are trying to find the boxes in the dataset to minimize RSS.

We choose the feature and splitting value that gives us the minimum SSE.

We recursively continue this process until a stopping criterion is reached, such as:

- No fewer than 5 observations in any leaf
- No fewer than some percentage of dataset in any leaf
- Max tree depth has been reached

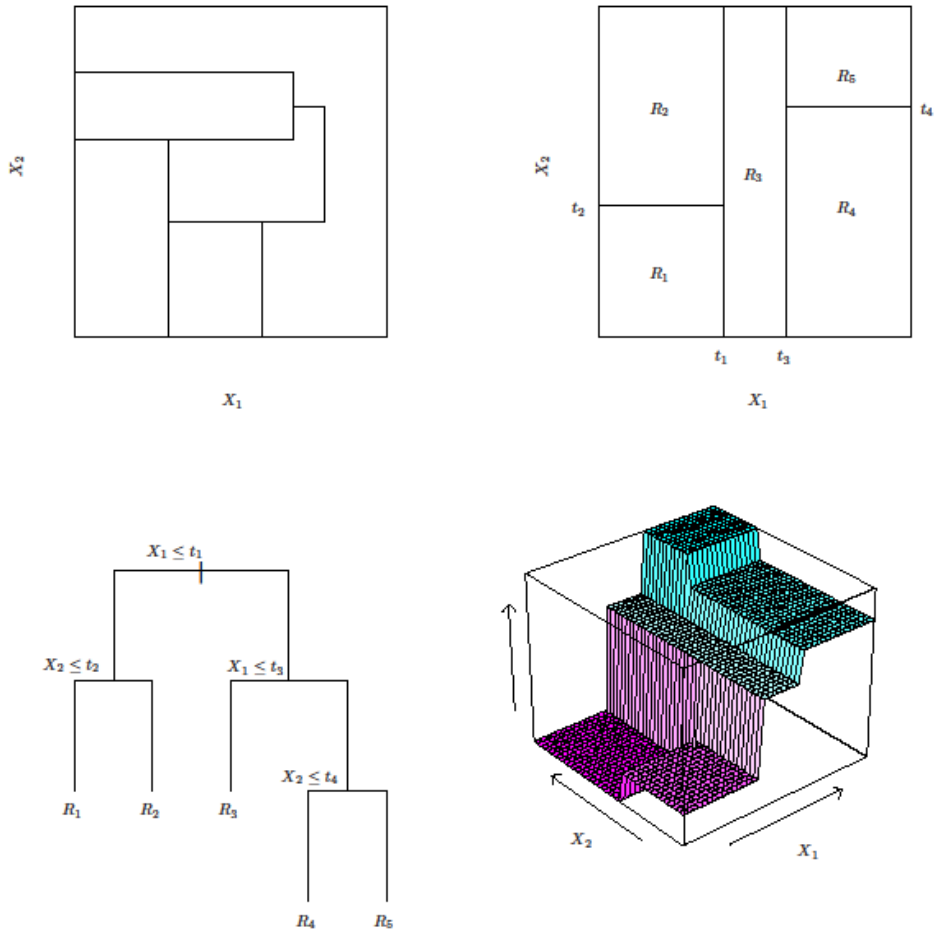


Figure 8.3 from *An Introduction to Statistical Learning*

Look at the figure above.

Top-left: Example of boundary boxes that COULD NOT be made from a tree

Top-right: Boundaries made from tree created in Lower-left

Lower-left: Regression tree

Lower-right: X1 and X2 projected onto Y (Y is the predicted output). Note that there are only 5 different output regions, corresponding to the five levels in the figure.

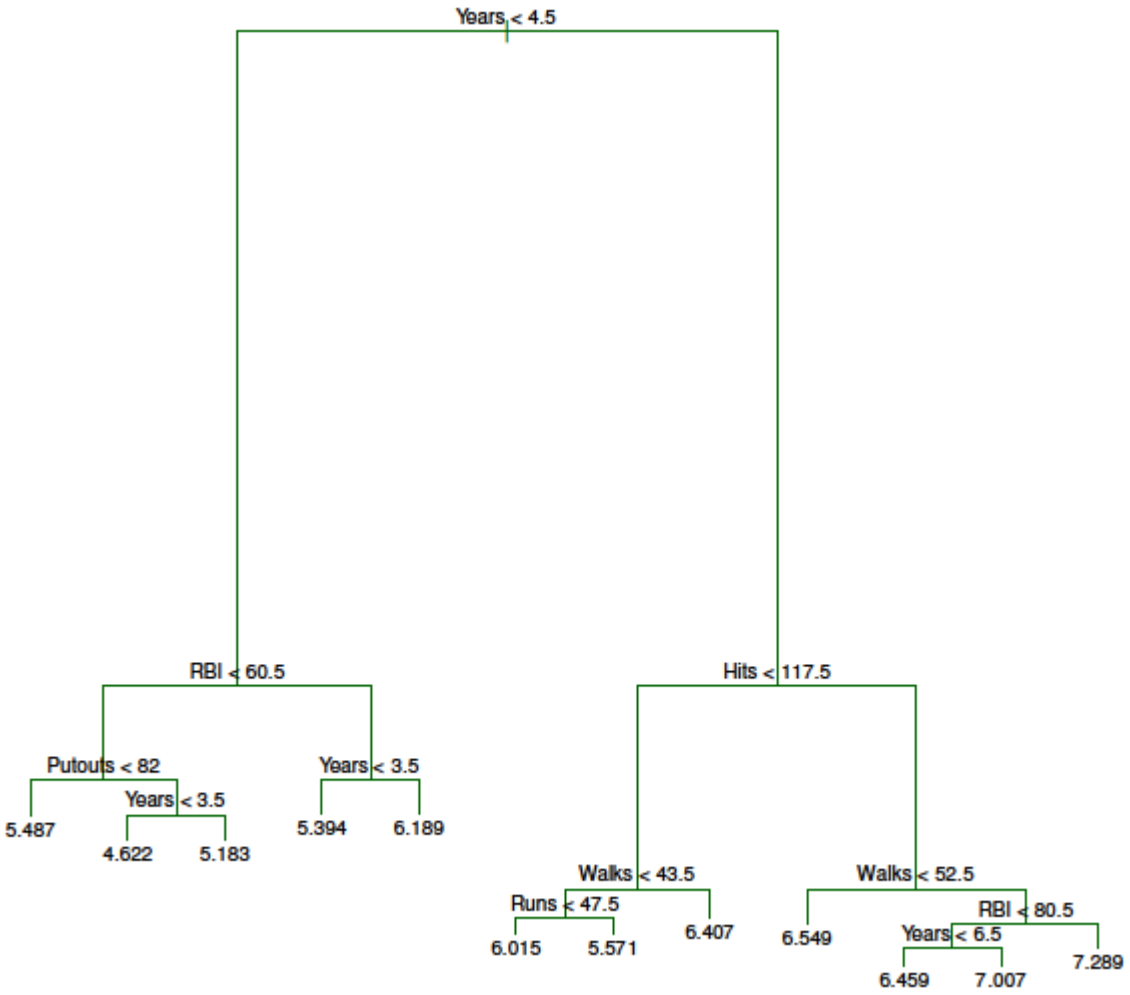


Figure 8.4 from *An Introduction to Statistical Learning*

Above is a regression tree to predict the salary of a baseball player (1000*e^(leaf value)).

Suppose a baseball player:

- Has played 5 years
- Has 120 hits
- Has 40 walks
- Has 10 runs
- Has 50 RBIs
- Has 60 Putouts

What is the predicted salary? _____

CS 438: Ensemble Methods for Trees

As you will see in lab, we can prune a tree to reduce overfitting. We can also keep a tree from getting too tall. These techniques alter a single decision tree. What if we instead create multiple trees?

Ensemble methods create and combine multiple trees to make predictions.

Activity: Each of you is a decision tree that predicts "sunny" or "rainy". Think of your prediction now.

Number who predict sunny: _____

Number who predict rainy: _____

Total number in ensemble: _____

What does the ensemble say? (sunny or rainy) _____

Now, how do we get multiple trees (aka, a forest)?

Technique 1: Bagging (short for bootstrap aggregation)

Suppose your dataset has 1000 observations. Instead of creating a single tree with all observations, we create B trees built from bootstrapped data.

Bootstrap dataset:

1. Choose size N for the sample size for S
2. For i = 1 to N:
    a. Select an observation O from the dataset D at random
    b. Add observation to S
    c. Put O back into D

1. Can a bootstrapped dataset have repeated observations? _____

2. Can a bootstrapped dataset not contain some observations from D? _____

Bagging creates B unpruned trees, each from a different bootstrapped dataset. Then, to make a prediction on a new observation x, the results are simply:

For regression trees: $\frac{1}{B}\sum_{b=1}^{B} f_b(x)$

For classification trees: majority vote (most common occurring class)

3. How does bagging reduce overfitting?

Handy validation set from bagging
Note that bagging gives us a built-in validation set to test the accuracy of our model. We do not need to do cross-validation or create a training/validation test set. Since our dataset is bootstrapped, there are observations that are not in that bootstrapped datasets that built our trees.

Out-of-bag (OOB) observation is one that was not used in the bootstrap to build the tree.

Here's an example of how we can use the OOB samples for testing the accuracy of the model.

> Suppose we have 1000 observations in D.
> Suppose N (bootstrap sample size) is 666.
> Suppose B (number of trees) is 300.
>
> Let O be one observation from D.
>
> Since the probability of O being selected to train any tree is about 0.67, there will be about a third of the trees for which O is not in the bootstrapped dataset. Let B' be the ~100 trees for which O is not in the bootstrapped dataset. We use the ensemble B' to make a prediction about O and record the error (misclassification for classifier or residual for regression).

How do we choose N?
> Can be all observations for smaller datasets
> 60% to 80% for larger datasets

How do we choose B?
> Run experiments to see which B produces better error rates
> Larger B usually works well

Technique 2: Random Forests

Another technique, called random forests, produces an ensemble of trees, but produces the trees with variations due to restrictions on the splitting features. Note that bagging produces tree variations due to variations in the training set.

Figure of Tammy's son Joel in a forest

Random forests will decorrelate trees and create a wider variety of trees.

Why would we restrict which features can be used to split the nodes?
Consider the case of a dataset where one feature has high importance (most information gain, for example). Will that feature be the top node of most of the trees for bagged trees? _____

But, if we remove that central feature from our set of choices for the root node, then another feature will become the first decision in the tree.

To create a random forest of B trees:

1. Choose m = number of predictors that will be used at each split
2. Let p = number of predictors in dataset D
3. For i = 1 to B
    a. Create bootrapped dataset S
    b. Build decision tree from D with split restriction of a random m from p predictors on S

Then, the classifier or regression output is done in the same way as bagging: for regression, it would be an average. For classification, it would be the class with the most votes.

What fraction of trees will have the central feature as the root node? _____

How do we choose m?

- Usually m = sqrt(p) is a good place to start for classification
- Usually m = p/3 is a good place to start for regression
- Can do this experimentally

How do we choose B?

- Larger B is usually better, so we experiment until we see the error rate be consistent

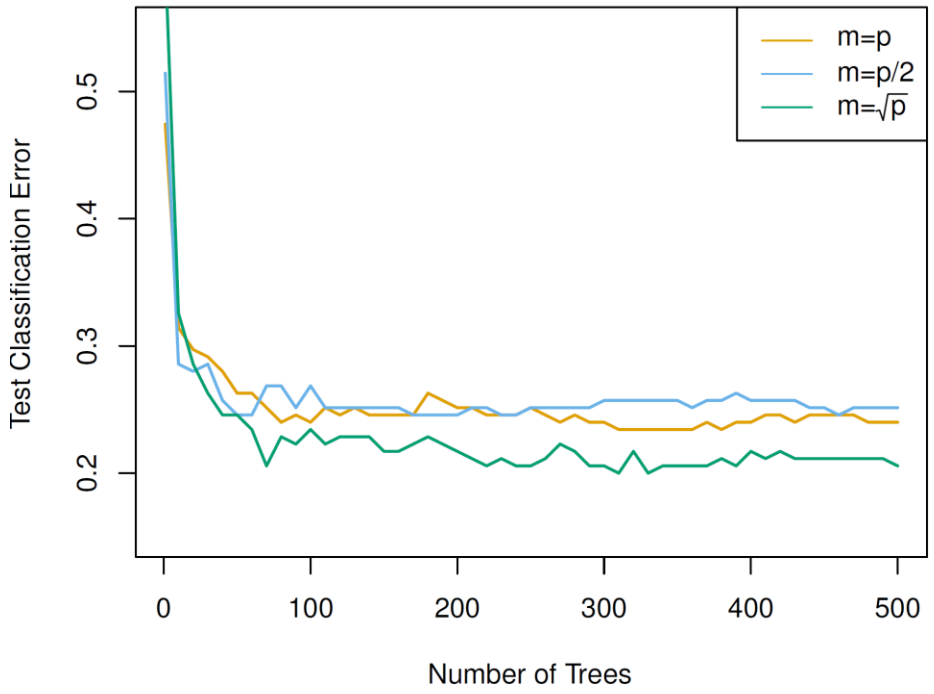


Figure 8.10 from *In Introduction to Statistical Learning*

Figure 8.10 shows the classification error for a random forest (classifying cancer type from 15 predictors). When m = p, this is equivalent to bagging. When m = sqrt(p), the error rate is the lowest. You can also see that 300 to 500 trees has similar results.

Technique 3: Boosting

This technique builds trees slowly, converting weak learners into a strong learner.

A weak learner does slightly better than random guessing.

Can we turn this weak learner into a strong leader? (Question posed in 1988)

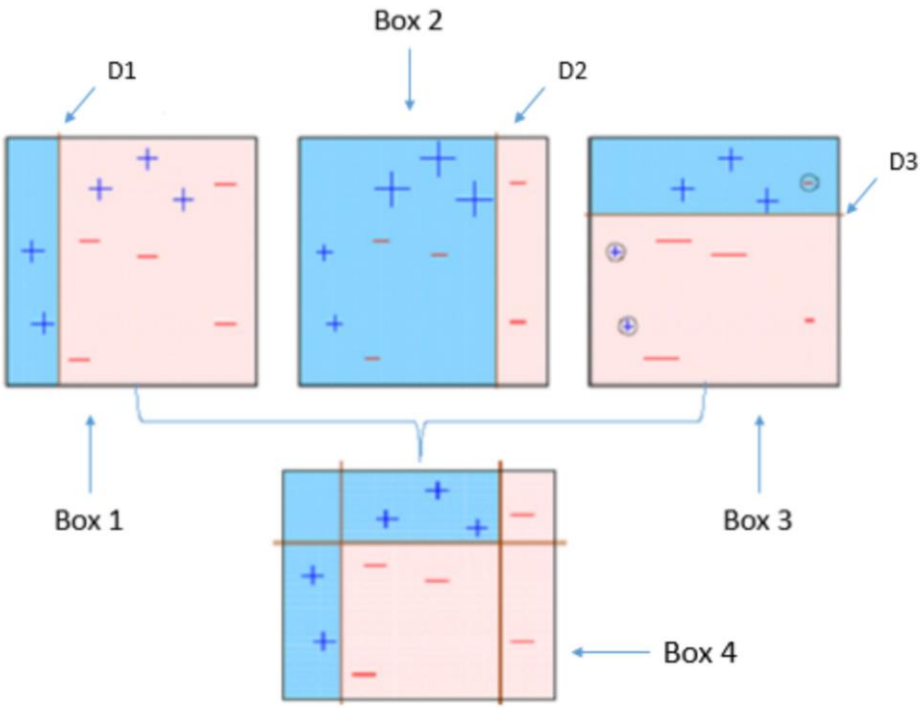
Yes, through boosting. There are many forms of boosting and many implementations, but here are the general principles:

Boosting Principles for Classification:

1. Learn slowly
2. Weight observations that are classified poorly more strongly in next tree constructed
3. Adjust model at each iteration

(here, if boosting is done with trees, then trees are built successively, instead of in parallel)

Here is an example from https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5:



Suppose X1 is the data value in the x-axis dimension and X2 is the data value in the y-axis dimension. Suppose Box 1 is our first tree constructed. Note that this would have a single root with X1 < the value of the line. It classifies the two observations in the blue zone correctly. However, it misclassifies three observations in the red zone.

The three observations that were misclassified now get a higher weight when creating the next tree. Box 2 is the next tree. Since those three +'s are now much larger, the best split is now the vertical line in Box 2. This one misclassifies three red as blue.

Box 3 is then constructed with the heavily weighted – misclassified observations as blue.

Now, we have three classifiers (short trees) that combine to make Box 4. Box 4 is the final classifier, where each of Box1, Box2, and Box3 have equally weighted votes.

Boosting Principles for Regression:

1. Fit new predictor to residual errors made by previous predictor
2. Each new tree added reduces residual error
3. Many ways to implement boosting algorithm (parameter estimation via gradient descent for example)

Algorithm 8.2 in *An Introduction to Statistical Learning* shows an algorithm for boosting for regression trees.

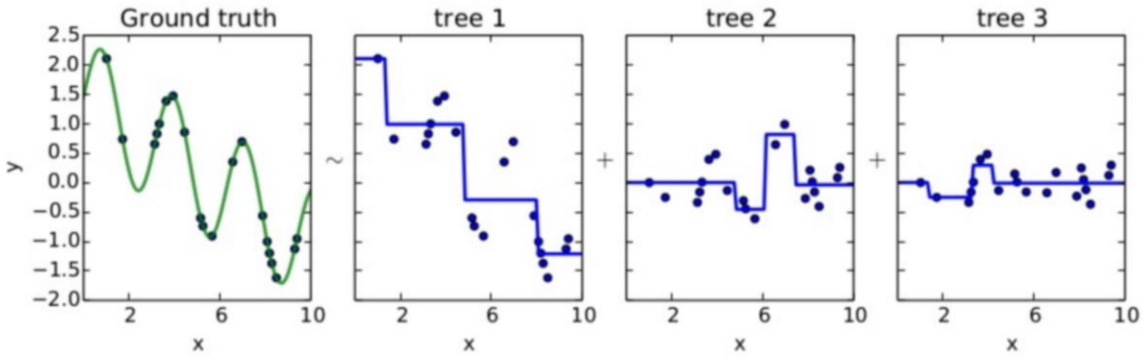


Figure from https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5

In this example, there is one predictor value x that maps to a response variable y. The left panel shows the actual function for these points.

Tree 1 is the first tree constructed (ordinary regression tree). Note that there are four leaves in tree 1 since there are four different y values that are possible. Well, let's look just at tree 1.

Which data points (x-values) are estimated well? _____ (circle them above)

Which data points (x-values) are estimated poorly? _____ (circle them above)

So, just having tree 1 would not give us a great regression tree.

We'll build tree 2 not from the original y values, but from the residuals from tree 1 (recall that the residual is the observed minus the predicted value). See the tree 2 panel. Tree 2 would have 4 leaves, since there are four different levels.

Then, tree 3 is built from the residuals of tree 2. Tree 3 has four leaves as well, since there are four different levels.

Now, we have a single prediction that is the sum of each of these trees.

Let's see how this works. Assume x = 7.

What is the output of tree 1 for x = 7? _____

What is the output of tree 2 for x = 7? _____

What is the output of tree 3 for x = 7? _____

Add these up: _____

Note that if we had just used tree 1, the predicted value would be -0.5, which isn't even close to the ground truth.

How well do the trees predict x = 2? _____

CS 438: Principal Component Analysis

Suppose we have a dataset of N observations with P features/predictors (all features in this case are numerical; note that categorical data can become numerical).

1. Suppose P is really large – 1000+. How do you decide which features to **keep** in the model?

2. Why would we want to reduce the number of features?

Reducing the number of features can give us data compression and/or remove features that are not helping the model (be it regression or classification).

One way to reduce the number of features is through feature selection. Of the P features, keep m that give the best accuracy for the model. In this case, there is no data transformation. We have done this in lab by eliminating columns in the data or not using certain columns when building models.

3. Is there another way you could reduce the number of features?

Activity 1: Suppose we have a dataset that consists of Age, Weight, and Cholesterol. We want to predict blood pressure, but we do not want to keep all the features. Give at least two methods to reduce the number of features from 3 to <= 2.

Method 1:

Method 2:

Suppose age and weight look like this:

Activity 2: How would you combine Age and Weight to give just one number to represent both for a person?

Principal Components Analysis determines the best vectors that are "closest" to the data. Another interpretation is that the principal component is the projection with highest variance.

The math relies on linear algebra (finding eigenvectors and eigenvalues), so if you have taken that course, this is an application of what you learned there. There is a tutorial on moodle if you are curious about how to do the math yourself.

Some properties of principal components:

- Each principal component is orthogonal to the others
- The 1st principal component explains the data the best (just using one number)
- The weights of the principal components are called loadings
- The principal component score is often written with Z and is the linear combination of the loadings multiplied by the (predictor_value – mean(predictor_value))
- The sum of the squares of the loadings equals one (just so we get back unique loadings, remember a vector is just a projection, so we could scale that vector and have an infinite number of the same projections)
- The # of possible principal components for an N X P matrix is min(N-1, P); usually P is smaller than N, so the # of principal components is P
- We keep the k highest principal components to reduce the dimensionality of the features
- Since a principal component is a projection, the loadings could be multiplied by -1 and we get the same projection, just with the vector pointing in the opposite direction. So, you can adjust the loadings by multiplying all by -1 if you want.

Suppose we collect data about a city's population and the total amount of advertising that is spent in that city. We have to features: Population and Ad_Spending. We want to find the principal components, so we can reduce the dimensionality from 2 to 1.

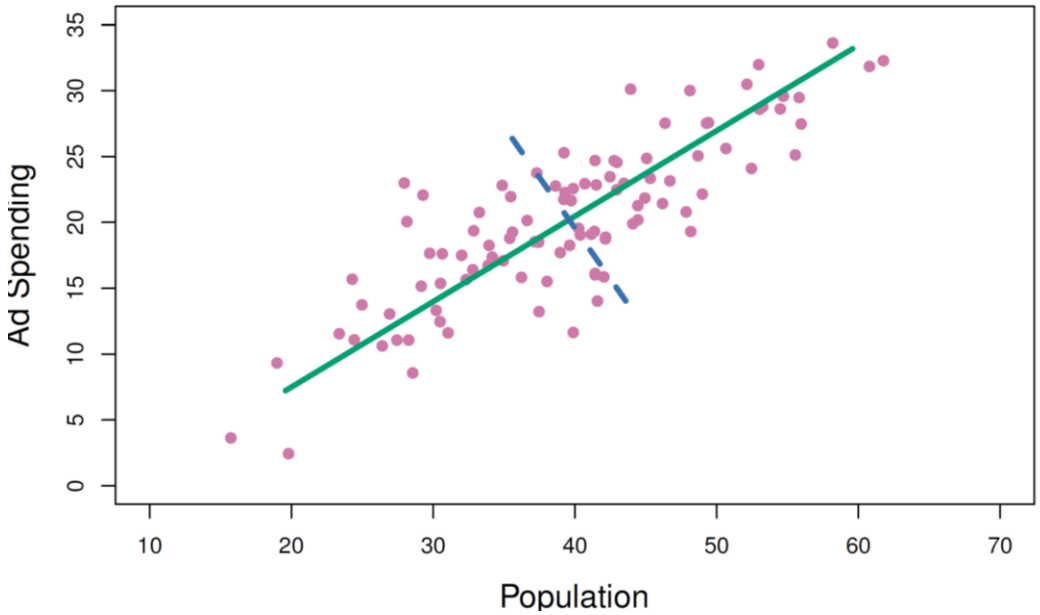


Figure 6.14 from *An Introduction to Statistical Learning*

The dots are data observations (one per city). The green solid line is the projection of the first principal component. See how it extends along the dimension of the most variability?

The dashed blue line is the second principal component. It is orthogonal to the first principal component and it projects along the second dimension with second-most variability.

If we want to combine Population with Ad_Spending into a *single* metric, we can use the first principal score for each observation.

The score Z1 for this dataset is computed with loadings 0.839 and 0.544.

First, let's check that the sum of squares of the loadings is equal to 1:

(.839)(.839) + (.544)(.544) = 0.999

$$Z_1 = 0.839 * \big(Population - Mean(Population)\big) + 0.544 * (Ad - Mean(Ad))$$

So, now we can apply this linear combination to the entire set of observations and have a single score for each observation.

Let's look at the loadings graphically now.

What is the approximate slope of that green line?

_____

What is the loading in the y-direction divided by the loading in the x-direction?

_____

For this dataset, the second principal component is shown by the blue dashed line. The loadings are 0.544 and -0.839.

$$Z_2 = 0.544 * \big(Population - Mean(Population)\big) - 0.839 * (Ad - Mean(Ad))$$

Do these loadings correspond correctly to the slope of the dashed line?

_____

How to the slopes of the two projections relate to one another?

_____

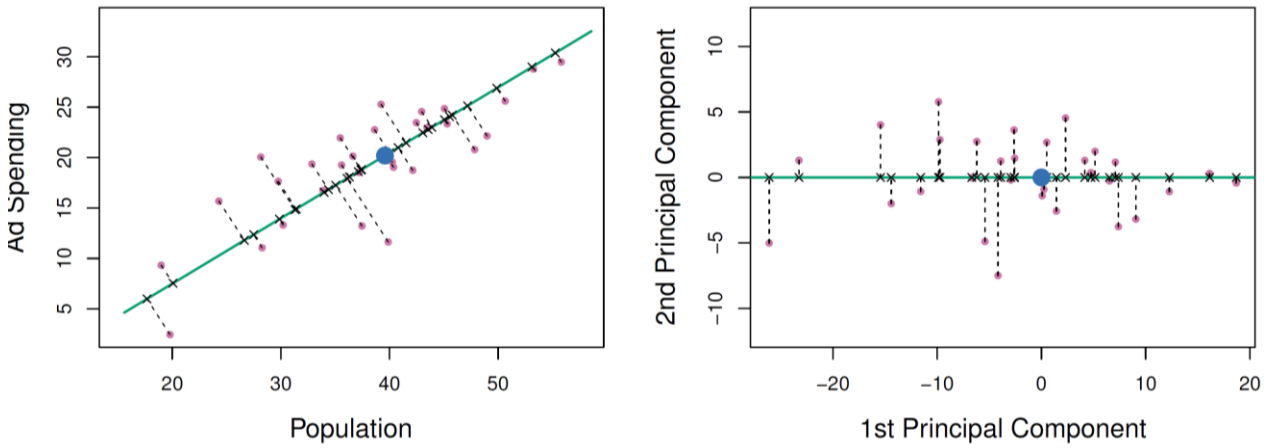


Figure 6.15 from *An Introduction to Statistical Learning:* shows the transformation from the observations to the axes of the two principal components

Another way we can think of the principal components is just a linear transformation from Population and Ad to a different set of two dimensions. The big circle is where the second principal component intersects the first principal component. We can take every observation and draw a perpendicular line to the first principal component and that is the score of the second principal component. The first principal component score is the distance from the observation to the blue dot along the green projection.

With this new graphic, we can see that if we compress Population and Ad into a single principal component score, those values are the x-values of the right figure.

Summary:

1. We can use PCA to reduce the dimensions of the feature set. We can then use the principal components for linear regression, decision trees, or clustering. It is simply a way to capture higher dimensions with lower dimensions.

2. PCA is NOT feature selection. PCA summarizes ALL the features. PCA can be used after features are selected or on the original dataset.

3. How do we know how many dimensions to use in PCA? Just like with our other techniques, we can try different numbers of dimensions and cross-validate (for regression or classification) to see where the accuracy or SSE levels off. Or for the case without a response variable, we can see how much the principal components explain of the total variance of the data.

4. When the number of observations is close to the number of features, overfitting (creating perfect models) becomes a major problem. For example, if we have two features and we have two observations, we can create a regression model (a line) that connects the points perfectly and the training error is 0. Well, if we have 100 observations and 100 features, we have the same issue (can fit a model perfectly). So, PCA gives us a tool to reduce the dimensionality of the dataset, so we are not as likely to overfit.

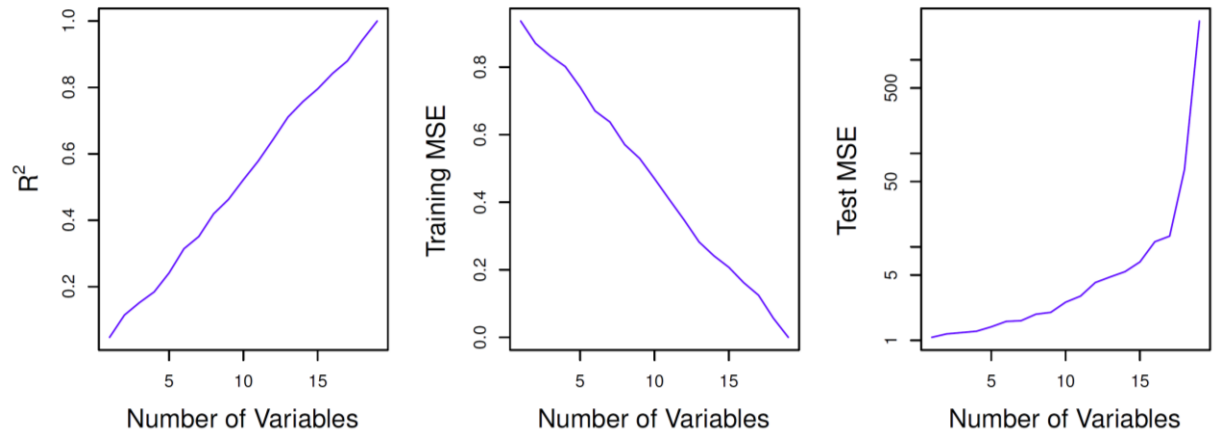


Figure 6.23 from *An Introduction to Statistical Learning:* shows what happens when the number of variables (features) becomes equal to the number of observations (in this case, N = 20). See how R squared becomes perfect, the training error becomes perfect, and the test error grows large.

5. It is important to scale the data (so mean of each feature is 0 with std dev of 1). Otherwise, the scale of the feature is going to dominate the variance and dominate the principal components. Just think back to the first example. If Ad spending is recorded in cents versus thousands of dollars, that dimension now becomes much more stretched. The figure below demonstrates this issue on a 4-dimensional dataset.

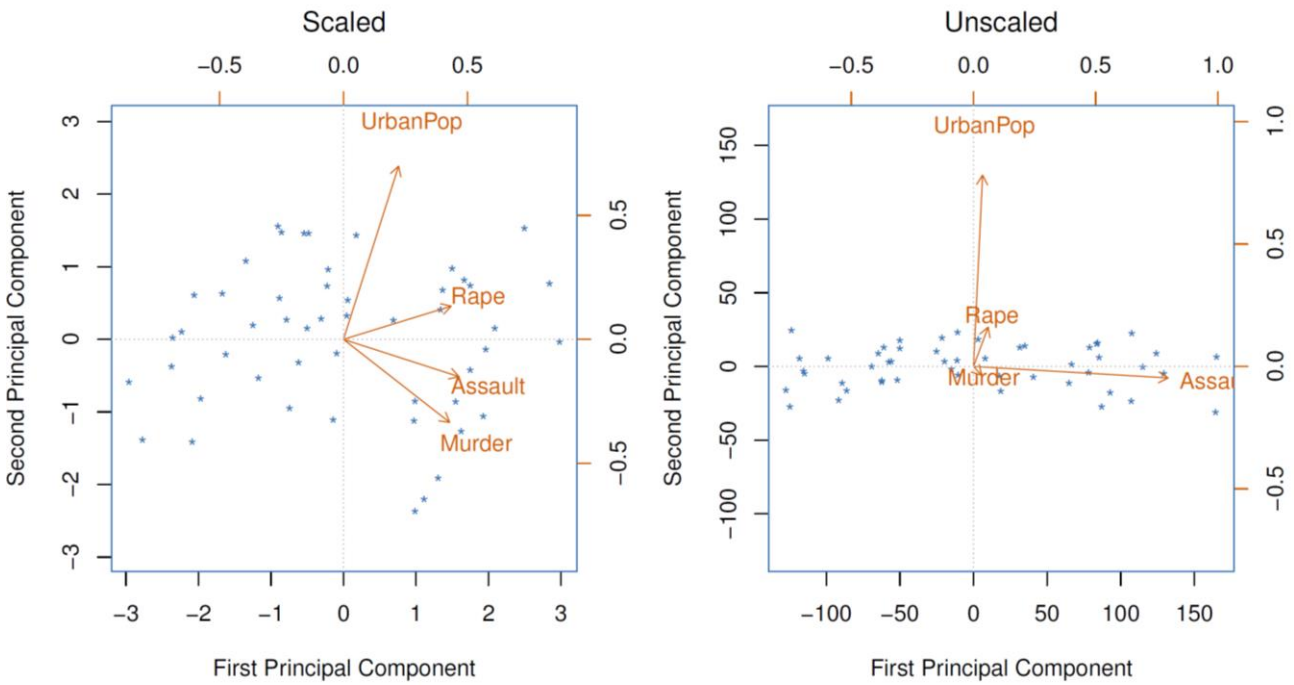


Figure 12.4 from *An Introduction to Statistical Learning:* shows what happens when the data is not scaled prior to PCA. Right figure shows that the loadings for the first two components can be very different for unscaled data. Notice the range of the principal component scores in the right figure versus the left figure.

CS 438: Clustering and k-Means

Clustering is a form of *unsupervised* learning, where we are just trying to understand how observations relate to one another. We do not have a class or response variable associated with each observation.

Activity 1: Suppose we have a two-dimensional dataset, so we can draw it below. Make a plot of 10 observations where there are three distinct clusters.

Activity 2: Suppose we have a two-dimensional dataset, so we can draw it below. Make a plot of 10 observations where there are no distinct clusters.

Activity 3: Given a set of points (2-dimensional), how would you determine where the clusters are?

Formal problem: Given N items with known distances between items and K (<= N), assign the N items into one of K clusters such that the total distance from each item to its cluster center is minimized.

Note: We can find the optimal set of clusters by trying every combination (subset) of the N items into K groups, but how much running time is this? _____

So, to create a faster algorithm, we use a greedy approach. K-means is a greedy clustering algorithm that given a dataset of features, will produce K sets of observations.

K-means clustering heuristic (Lloyd):

```
Input: n items, k = number of clusters to create

For each item, randomly assign it to one of the k clusters.
Calculate cluster centers for each cluster.
While there are new cluster assignments:
      For every item i:
            Assign i to the cluster with closest center.
      Re-compute cluster centers for each cluster.
```

To calculate the square of Euclidean distance from item c to item t:

$D(<c_1, c_2, \ldots, c_m>, <t_1, t_2, \ldots, t_m>) = \Sigma_i [(c_I - t_i)^2]$ // note that we do not need to take sqrt since all distances are positive and this reduces the number of computations

To find the center c from a set of items A:

$$\text{center}(A) = \frac{\text{vector sum of items in A}}{|\ A\ |}$$

Class Activity: Get a random colored paper. Go to spot in the classroom. You are the data points we are clustering. We will create three clusters, so K = 3.

We will simulate k-means as a class.

Discussion:

1. Is this heuristic guaranteed to stop? _____
2. Will k-means always produce the same set of clusters? _____
3. How can we make this more robust, given that the initial random assignment affects the final clusters? _____

4. What examples of datasets would be useful to cluster?

Example k-means clustering:

Suppose our dataset is two-dimensional with x1 and x2 as features. We will do k=2.

	x1	x2
obs1	0	0
obs2	0	1
obs3	1	1
obs4	1	0
obs5	.5	.5
obs6	5	5
obs7	5	6
obs8	6	6
obs9	6	5
obs10	5.5	5.5

What does the dataset look like plotted? How may clusters do you see? _____

Step 1: Randomly assign each item to one of two clusters:

C_1 = {obs1, obs4, obs5, obs8}

C_2 = {obs2, obs3, obs6, obs7, obs9, obs10}

Step 2: Calculate cluster centers:

Center(C_1) = (<0, 0> + <1, 0> + <.5, .5> + <6, 6>) / 4

= <7.5, 6.5> / 4

= <1.875, 1.625>

Center(C_2) = (<0, 1> + <1, 1> + <5, 5> + <5, 6> + <6, 5> + <5.5, 5.5>) / 6

= <22.5, 23.5> / 6

= <3.75, 3.917>

While no new cluster assignments: Calculate distance from each point to cluster center to assign point to cluster:

Item	Sq Distance to center C_1	Sq Distance to center C_2	Cluster assignment
<0, 0>	$1.875^2 + 1.625^2 = 6.15625$	$3.75^2 + 3.917^2 = 29.405$	C_1
<0, 1>	3.91	22.57	C_1
<1, 1>	1.16	16.07	C_1
<1, 0>	3.41	22.91	C_1
<.5, .5>	3.15	22.24	C_1
<5, 5>	21.16	2.74	C_2
<5, 6>	28.91	5.90	C_2
<6, 6>	36.16	9.40	C_2
<6, 5>	28.41	6.24	C_2
<5.5, 5.5>	28.16	5.57	C_2

Calculate cluster centers:

Center(C_1) = (<0, 0> + <0, 1> + <1, 1> + <1, 0> + <.5, .5>) / 5

= <.5, .5>

Center(C_2) = (<5, 5> + <5, 6> + <6, 6> + <6, 5> + <5.5, 5.5>) / 5
= <5.5, 5.5>

Calculate distance from each point to cluster center to assign point to cluster:

Item	Sq Distance to center C_1	Sq Distance to center C_2	Cluster assignment
<0, 0>	.5	60.5	C_1
<0, 1>	.5	50.5	C_1
<1, 1>	.5	40.5	C_1
<1, 0>	.5	50.5	C_1
<.5, .5>	0	50.0	C_1
<5, 5>	40.5	.5	C_2
<5, 6>	50.5	.5	C_2
<6, 6>	60.5	.5	C_2
<6, 5>	50.5	.5	C_2
<5.5, 5.5>	50.0	0	C_2

No new cluster assignments, so STOP. The final assignment is in the above table.

Another example with k = 3:

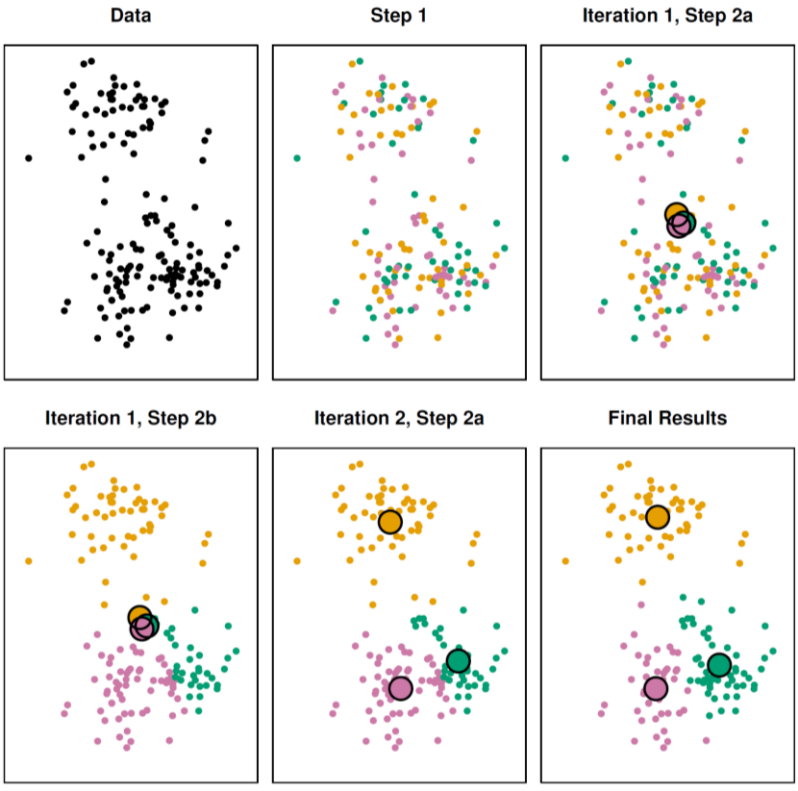


Figure 12.8 from *An Introduction to Statistical Learning*. Shows the steps of k-means.

CS 438: Hierarchical Clustering

As you read in the paper about clustering beyond k-means, there are many variations of clustering using centroids and distances to create groups. In all k-means variants, the value of k (# of clusters) is specified as an input parameter.

An alternative to clustering with a target value for k is to perform hierarchical clustering, where we create a tree to showcase distance relationships. Once we have the tree constructed, we can inspect it to see where good cut-offs would be to create groupings.

Algorithm:
Input: NxN matrix D, representing distances between all pairs of observations
Output: A tree where each leaf represents one observation

Initialization: Each observation is in its own cluster $\{C_1, C_2, C_3, \ldots C_N\}$. Let C denote the set of all clusters.
Construct T tree (called a dendrogram) as follows:

While $|C| > 1$:

a. Find the two closest clusters C_i and C_j
b. Merge C_i and C_j to form C_{ij} so it has all elements from both clusters
c. Add new merge point in T to link clusters C_i and C_j [height of merge point is the distance from C_i to C_j]
d. Remove rows and columns in D associated with C_i and C_j
e. Add new row and column for C_{ij} and input updated distances from C_{ij} to all other clusters in C

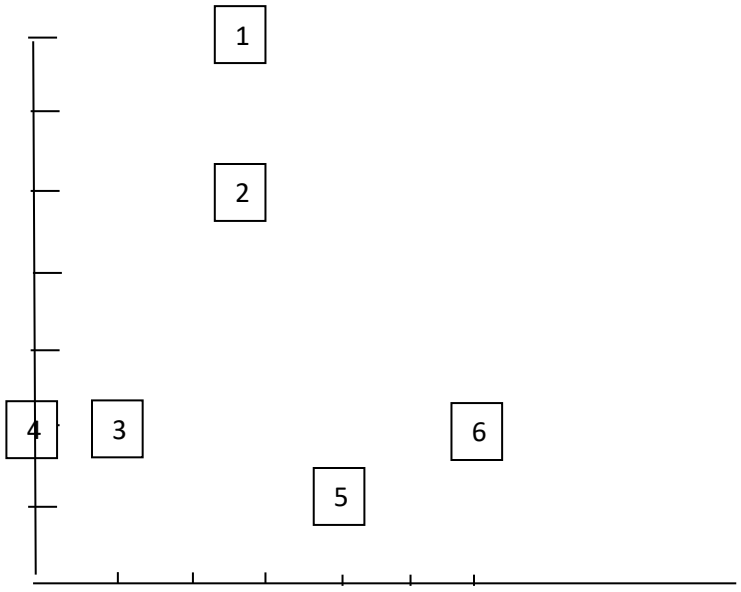
How is "closest" defined for clusters?

This can be defined by any metric, as long as distance is some numerical value.

For now, let's assume Euclidean distance is way to calculate distance and "closest" is the closest two points of the clusters

Example:
Suppose we have 6 observations:

	feature1	feature2
obs1	3	7
obs2	3	5
obs3	1	2
obs4	0	2
obs5	4	1
obs6	6	2



Need to calculate the distance matrix (input to hierarchical clustering):

Obs1	Obs2	Obs3	Obs4	Obs5	Obs6	
	2	5.39	5.83	6.08	5.83	Obs1
		3.61	4.24	3.61	4.25	Obs2
			1	3.16	5	Obs3
				4.12	6	Obs4
					2.24	Obs5
						Obs6

Use metric: closest Euclidean distance between closest elements in each cluster metric

Of the 6 clusters, which two are the closest? _____

Merge closest and update distances:

Obs1	Obs2	Obs5	Obs6	{Obs3, Obs4}	
	2	6.08	5.83	5.39	Obs1
		3.61	4.25	3.61	Obs2
			2.24	3.16	Obs5
				5	Obs6
					{Obs3, Obs4}

Of the 5 clusters, which two are the closest? _____

Merge closest and update distances:

Obs5	Obs6	{Obs3, Obs4}	{Obs1, Obs2}	
	2.24	3.16	3.61	Obs5

		5	4.25	Obs6
			3.61	{Obs3, Obs4}
				{Obs1, Obs2}

Of the 4 clusters, which two are the closest? _____

Merge closest and update distances:

{Obs3, Obs4}	{Obs1, Obs2}	{Obs5, Obs6}	
	3.61	3.16	{Obs3, Obs4}
		3.61	{Obs1, Obs2}
			{Obs5, Obs6}

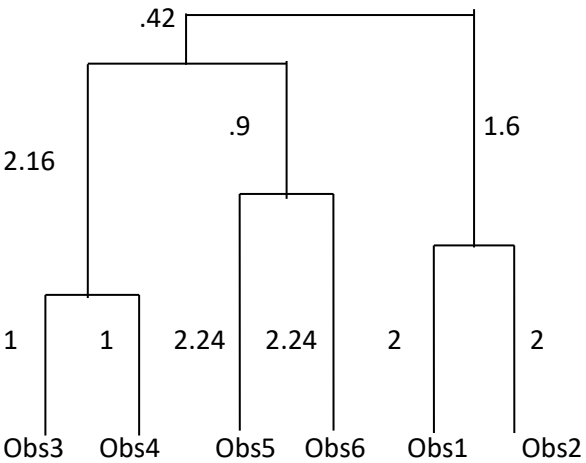
Of the 3 clusters, which two are the closest? _____

Merge closest and update distances:

{Obs1, Obs2}	{Obs3, Obs4, Obs5, Obs6}	
	3.61	{Obs1, Obs2}
		{Obs3, Obs4, Obs5, Obs6}

Of the 2 clusters, which are the closest? _____

Overall hierarchical tree:



Activity: How do we create clusters?

We choose a distance from the top and draw a horizontal line. Think of this as actually cutting the tree by branches. All the leaves from each cutting form the observations in the cluster.

Suppose we cut .2 from the top of the tree. How many clusters are created? _____

What are the elements of those clusters? _____

Suppose we cut 1.0 from the top of the tree. How many clusters are created? _____

What are the elements of those clusters? _____

Suppose we cut 2.0 from the top of the tree. How many clusters are created? _____

What are the elements of those clusters? _____

Distances (also known as dissimilarity):
There are many ways to calculate distance, as we saw earlier in the course:

Euclidean
Manhattan
Hamming
1 - Correlation

Euclidean and (1-correlation) are commonly used.

Linkages (metric for choosing closest clusters):

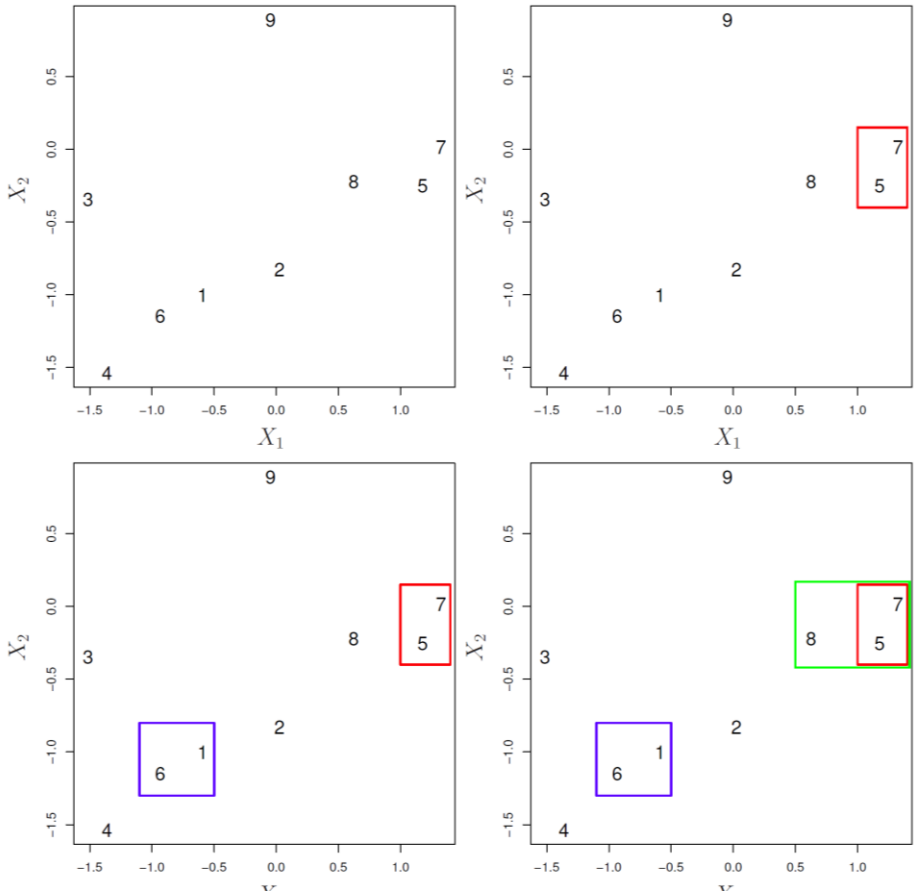
Complete	Compute all pairwise distances between observations in cluster A and cluster B; linkage is *largest* of the distances
Single	Compute all pairwise distances between observations in cluster A and cluster B; linkage is *smallest* of the distances [this is what the example above used]
Average	Compute all pairwise distances between observations in cluster A and cluster B; linkage is *average* of the distances
Centroid	Compute centroids for cluster A and cluster B; linkage is the distance between centroids [note: this can result in inversions, where merge points are below individual clusters and the tree can be difficult to interpret]

Some notes about dendrograms:

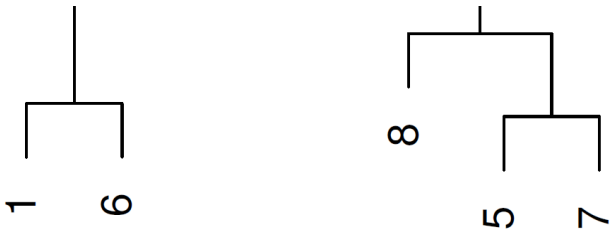
- Can rotate tree to create several equivalent trees
- Read from the leaves up to merges to understand similarity. Just because two observations are adjacent leaves in the tree does not mean they are similar.
- When reading dendrograms, the distance between observations is estimating by looking at the height of their first common merge point.
- Not the optimal set of clusters; hierarchical clustering is *greedy* (makes best choice at each step)
- Runtime is: _____

Activity: Continue creating the dendrogram for the dataset shown below. This is Figure 12.13 from an *Introduction to Statistical Learning*. The top-left part of the figure shows the original 9 observations. The top-right part of the figure shows the first merge. The bottom-left part shows the second merge. The bottom-right shows the third merge.

Use Euclidean distance and <u>complete</u> linkage.



Continue drawing tree: (for complete linkage, take the max of the pairwise distances)
The next merge is a close call. Assume distance(1,4) is smaller than distance(6,2).



Final Solution (left) and Original Dataset (right):

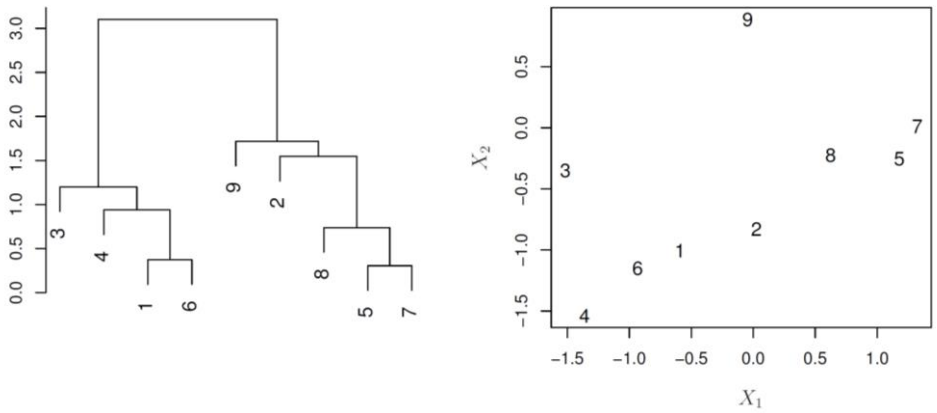


Figure 12.12 from *An Introduction to Statistical Learning with Applications in R*

If time: Build the tree for the dataset above using Euclidean distance and single linkage.

CS 438: Data Visualization

Review the best data visualizations from the New York Times:
https://www.informationisbeautifulawards.com/news/118-the-nyt-s-best-data-visualizations-of-the-year

Review different types of visualizations here:
https://www.tableau.com/learn/articles/data-visualization

1. For your project, what data models did you build?

2. Choose at least two visual representations from the list that might be helpful for to show data and models for your data science project:

A. Visualization #1 Type: _____

What kind of data does it best depict?

What are its strengths?

What are its weaknesses?

How might it showcase the project data and models?

B. Visualization #1 Type: _____

What kind of data does it best depict?

What are its strengths?

What are its weaknesses?

How might it showcase the project data and models?

CS 438: Association Rules from Sets

Review: What is a set?

Activity 1:
Suppose you have a store and record everything purchased in the same transaction. That transaction has a set of items.

Example (each set on its own line):

```
Milk    apples  carrots bread   butter
Butter  bread   yogurt  eggs
Milk    yogurt  eggs    lemon   apples  cereal
Cereal  milk
Carrots bread
Soda    peanuts chips   eggs
Cereal  milk
Yogurt  soda    eggs    cereal
Cereal  apples  butter  eggs
Apples  eggs    carrots
```

1. Do you see any patterns among the sets?

2. Could you define any of these as rules, such as Milk -> Butter? (in other words, if milk is bought, then butter is bought?)

These are called *association rules* or *market basket analysis*. The Apriori algorithm creates these rules by finding itemsets based on frequencies and then building rules from the itemsets.

Apriori Algorithm: Finding Itemsets

First, we need to define *support:*

Support(L) = percentage of transactions that contain L, where L is a set

Minimum Support = lower threshold for determining itemsets

1. Suppose bread has support of 60% in a basket. Could {bread, butter} have support > 60%? _____

```
D = transaction data
M = minimum support
N = size of largest itemset to consider

Apriori(D, M, N):
       K = 1
       Lk = {itemsets of size 1 with minimum support M}
       While Lk is non-empty and K < N:
              Generate candidate itemsets of size k+1 from Lk
              Calculate support for candidate itemsets
              Lk+1 = candidate itemsets with minimum support M
              K = K+1
       Return union of Lk for k=1 to N
```

Activity 2: Calculate the support for the 1-itemsets:

```
Milk    apples  carrots bread   butter
Butter  bread   yogurt  eggs
Milk    yogurt  eggs    lemon   apples  cereal
Cereal  milk
Carrots bread
Soda    peanuts chips   eggs
Cereal  milk
Yogurt  soda    eggs    cereal
Cereal  apples  butter  eggs
Apples  eggs    carrots
```

Support(Milk) = _____
Support(Butter) = _____
Support(Yogurt) = _____
Support(Eggs) = _____
Support(Bread) = _____
Support(Carrots) = _____
Support(Apples) = _____
Support(Cereal) = _____
Support(Lemon) = _____
Support(Soda) = _____
Support(Peanuts) = _____
Support(Chips) = _____

1. Which of the these have minimum support of 0.35?_____

Now, create 2-itemsets from these 1-itemsets:
Support(Milk,Eggs) = _____
Support(Milk, Apples) = _____
Support(Milk, Cereal) = _____
Support(Eggs, Apples) =_____
Support(Eggs, Cereal) = _____
Support(Apples, Cereal) = _____

2. Which of these have minimum support of 0.35? _____

3. Does the algorithm to generate itemsets keep going? _____

Apriori Algorithm: Creating Rules

Once we have the itemsets, we now can create rules such as X -> Y.

There are three common metrics used to generate rules:

- Confidence
- Lift
- Leverage

All use the support metric defined above.

$$Confidence(X \rightarrow Y) = \frac{Support(X \wedge Y)}{Support(X)}$$

$$Lift(X \rightarrow Y) = \frac{Support(X \wedge Y)}{Support(X) * Support(Y)}$$

$$Leverage(X \rightarrow Y) = Support(X \wedge Y) - Support(X) * Support(Y)$$

Activity 3: Let's apply these metrics.

Suppose X and Y = {bread, milk, eggs} with support 0.2
Suppose X = {bread, eggs} with support 0.2
Suppose Y = {milk} with support 0.4

1. What is the confidence of X -> Y? _____

2. What is the lift of X -> Y? _____

3. What is the leverage of X -> Y? _____

Note: Confidence does not consider the support of just the set Y in the calculation, so it cannot tell if it is coincidental. Lift and leverage do consider the support of Y.

4. What is the range for confidence? _____

Lift greater than 1 indicates usefulness for the rule.

Leverage greater than 0 indicates usefulness for the rule.

5. In which applications would this data analysis approach be useful? _____

CS 438: Big Data Tools

Research one of the following platforms/tools:

- Apache Hadoop
- Apache Storm
- Apache Spark
- Apache Cassandra
- Apache Flink
- MongoDB
- Lumify
- Kanini
- HPCC Systems
- Tableau
- Sisence
- Cloudera
- Domo
- Cloud infrastructure (Azure, AWS, Google, Oracle, …)
- Or choose another one you want to research

What platform/tool did you research?

Provide at least two concrete use cases for this tool:

1.

2.

What are the limitations (cost, size, update time, etc.) of this tool?

Be ready to share about your tool with the whole class.

CS 438: Ethics Considerations

Refer to the paper "Big Data Ethics" by Andrej Zwitter published in *Big Data & Society*, July – December 2014, pages 1 – 6. Andrej Zwitter is from the University of Groningen, in the Netherlands.

Activity 1: Consider the topics we discussed in the course. Make a list of topics related to data analytics that may require rethinking of philosophy, ethics, policy-making, or research.

Ethics Framework (from the paper):

Framework for ethics in this paper comes from *moral responsibility* of the individual. This framework is referred to as *moral agency*.

Causality: Agent can be held responsible if the ethically relevant result is outcome of its actions.

Knowledge: Agent can be blamed for result of its actions if it had knowledge of the consequences.

Choice: Agent can be blamed for the result if it had the liberty to choose an alternative without greater harm for itself.

Activity 2: Ethical Big Data Challenges Consider the concepts below (privacy, group privacy, propensity, research ethics) and answer the questions.

1. *Privacy (as individuals*): What data is collected? What every day actions are transparent?

What is an example of data that may be collected that impacts your privacy?

2. *Group Privacy*: 1) Taking a large dataset and filtering to just one individual, even though the dataset is anonymous. 2) Target people to behave in a certain way, 3) Hyper-connectivity (social media, for example) provides access to bots

What is an example of data that could impact group privacy?

3. *Propensity*: making predictions about what people are going to do (for example, commit a crime or have the demographics for which domestic violence is more likely)

What is an example of data that could make predictions that raises ethical questions? (We have seen some of these in the papers in the course)

4. *Research Ethics*: ethical codes and standards for privacy and data use, informed consent for data use, and maintaining privacy of individuals in research studies

Who should be governing data collection and use? At UP, we have an Institutional Review Board that reviews all studies involving people. Medical schools and clinics have a review board. But what about less formal research studies?

CS 438: Takeaways

1. What themes will you take away from this course?

2. How do you look at data differently now after taking this course?

3. Is data objective? Why or why not?

4. Are models objective? Why or why not?

Appendix: Quizzes

There are six quizzes in the course. Please arrive to class on time on quiz days. Each quiz will be 25 minutes in duration. You may use 1 crib sheet (one side of a regular piece of 8.5” x 11” paper or smaller) as notes during the quiz. For some quizzes, you may use a regular calculator. Otherwise, the quizzes are closed to other resources.

Quiz topic sheets can be found on Moodle. The set of material may change due to pacing of the course, so quiz topic sheets are not printed in the lecture notes. Material for quizzes will be drawn from the prelabs, labs, readings (books, articles, textbook), and lectures.