Descriptive Statistics for Exploratory Data Analysis

Ivan Corneillet

Data Scientist

GENERAL ASSEMBLY

Learning Objectives

After this lesson, you should be able to:

- ID variable types
- Use the *pandas* (and *NumPy*) libraries to analyze datasets using basic summary statistics: mean, median, mode, max, min, quartile, inter-quartile range, variance, standard deviation, and correlation
- Create data visualizations – including boxplots, histograms, and scatter plots – to discern characteristics and trends in a dataset

Announcements and Exit Tickets

Q & A

Git and GitHub

Take 2

Practice #1

Fork the course repository; clone your fork (one-time setup)

- Using the GitHub web interface:
  - Open https://github.com/ga-students/DS-SF-23; click on the Fork button on the top right; your fork is https://github.com/paspeur/DS-SF-23
- Using your terminal:
  - `git clone https://github.com/paspeur/DS-SF-23`
  - `cd DS-SF-23`
  - `git remote add upstream https://github.com/ga-students/DS-SF-23`

Update your clone and fork (in that order) (recurring)

- `git fetch upstream`
- `git merge upstream/master`
- `git commit –m "Merged commits from ga-students/SF-DAT-23 up to xxx"`
  - (if the merge was “Fast-forward”, i.e., trivial, there is no commit to do)
- `git push`
  - (Git might ask you your GitHub credentials the first time around)

Practice #2

Clone the course repository (one-time setup)

- Using your terminal:
  - `git clone https://github.com/ga-students/DS-SF-23`
  - `cd DS-SF-23`

Update your clone (recurring)

- `git pull`
  - `git pull` combines `git fetch` and `git merge` in one operation
- `git commit –m "Merged commits from ga-students/SF-DAT-23 up to xxx"`
  - (if the merge was “Fast-forward”, i.e., trivial, there is no commit to perform)

Working on Unit Project #1 and committing it to Git/GitHub

Assumptions

- Your clones DS-SF-23 and DS-SF-23-work are the at same level (i.e., you were in the same directory/folder when you cloned these repositories; e.g., the root of your home directory)
- If you decided to use DS-SF-23 to submit your assignments, replace “DS-SF-23-work” with “DS-SF-23” in these slides

Initial commit

- `cd DS-SF-23-work`
- `cp ../DS-SF-23/unit-projects/1/code/unit-project-1-starter-code.ipynb unit-project-1-ivan.ipynb`
- `git add unit-project-1-ivan.ipynb`
- `git commit -m "Ivan's Unit Project #1 (unmodified from the course repository)"`
- `git push`

Working on Unit Project #1 and committing it to Git/GitHub (cont.)

Subsequent edits and commits

- Edit your iPython Notebook
  - `jupyter notebook`
    - Save as often as needed
- Commit early and often
  - `git add unit-project-1-ivan.ipynb`
  - `git commit -m "Updated Unit Project #1"`
  - `git push`

If you can't push, you need to merge

- If you have multiple clones of the same GitHub repositories (or you have multiple people working of the same GitHub repository), the local clone you are trying to push from the git pull might be "behind" (it doesn't have all the commits from the origin/GitHub repository)
- `git fetch`
- `git merge`
- `git commit –m "Merged commits from ga-students/SF-DAT-23-work"`
  - (if the merge was "Fast-forward", i.e., trivial, there is no commit to perform)

Working on Unit Project #1 and committing it to Git/GitHub (cont.)

If you can't merge. E.g., *unit-project-1-ivan.ipynb*

- You have conflict(s). For example, you are trying to push changes that modify the same cell that a previous commit on the origin/GitHub repository also changed. Somehow, Git cannot resolve the conflict because both commits aren't compatible with each other so it errors out and let you resolve the merge manually)
- You won't be able to resolve the conflict with iPython Notebook as Git annotated the file with the merge conflict that broke the structure of the notebook

If you can't merge. E.g., *unit-project-1-ivan.ipynb* (cont.)

- `git diff unit-project-1-ivan.ipynb`
  - The output will tell you in plain text where the conflicts occurs. Make notes of them
- `git reset --hard`
  - Undo the merge for the time being
- `cp unit-project-1-ivan.ipynb unit-project-1-ivan-pre-merge.ipynb`
  - Make a copy of your pre-merge changes because you will apply them manually

Working on Unit Project #1 and committing it to Git/GitHub (cont.)

If you can't merge. E.g., *unit-project-1-ivan.ipynb* (cont.)

- `git merge`
  - Try to merge again; of course, you'll get the same conflicts
- `git checkout --theirs unit-project-1-ivan.ipynb`
  - Checkout the copy from the origin/GitHub repository
- Now you can use iPython notebook to open both *unit-project-1-ivan.ipynb* (the last version in the origin/GitHub repository) and *unit-project-1-ivan-pre-merge.ipynb* (the file you wanted to push)

If you can't merge. E.g., *unit-project-1-ivan.ipynb* (cont.)

- Copy and paste the relevant sections from *unit-project-1-ivan-pre-merge.ipynb* back to *unit-project-1-ivan.ipynb*
- `git add unit-project-1-ivan.ipynb`
  - Basically to tell Git you resolve the conflicts
- `git commit -m "Updated Unit Project #1"`
- `git push`

Guest Speaker

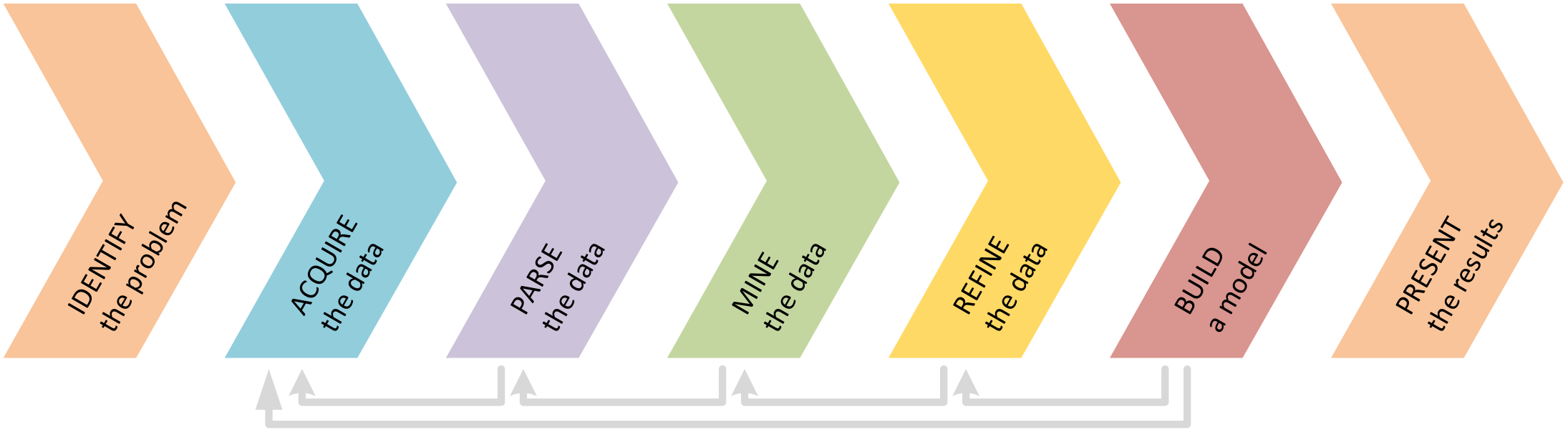
Michael Lin, General Assembly Data Science Alumnus

Review

Review

Data Science Workflow

And step ❼ is PRESENT the results



Review

❶ *IDENTIFY the problem*

The SMART Framework for Data Science

❶ IDENTIFY the problem | The SMART Framework for Data Science: (cont.)

SPECIFIC	The dataset and key variables are clearly defined
MEASURABLE	The type of analysis and major assumptions are articulated
ATTAINABLE	The question you are asking is feasible for your dataset and is not likely to be biased
REPRODUCIBLE	Another person (or you in 6 months!) can read your state and understand exactly how your analysis is performed
TIME-BOUND	You clearly state the time period and population for which this analysis will pertain

Trends often change over time and vary by the population of source of your data. It is important to clearly define who/what you included in your analysis as well as the time period for the analysis

Review

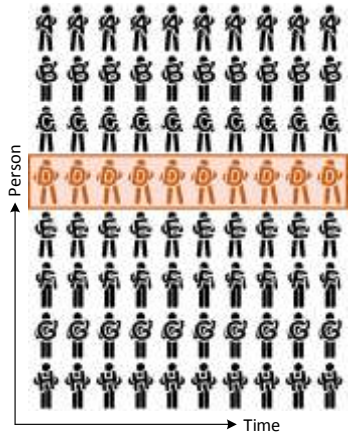
❷ *ACQUIRE the Data*

Data Types

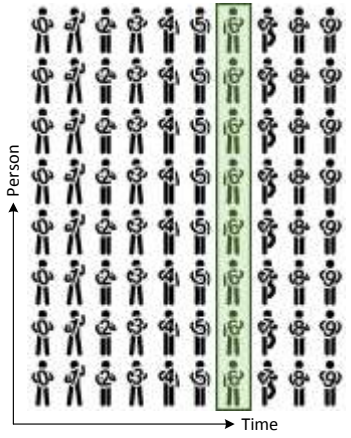
❷ ACQUIRE the Data | Data Types | Unstructured/ structured data; longitudinal/cross-sectional data

Bundit Chuangboonsri © 123RF.com

milosb © 123RF.com



Khoon Lay Gan © 123RF.com



Khoon Lay Gan © 123RF.com

Review

❸ *Parse the Data*

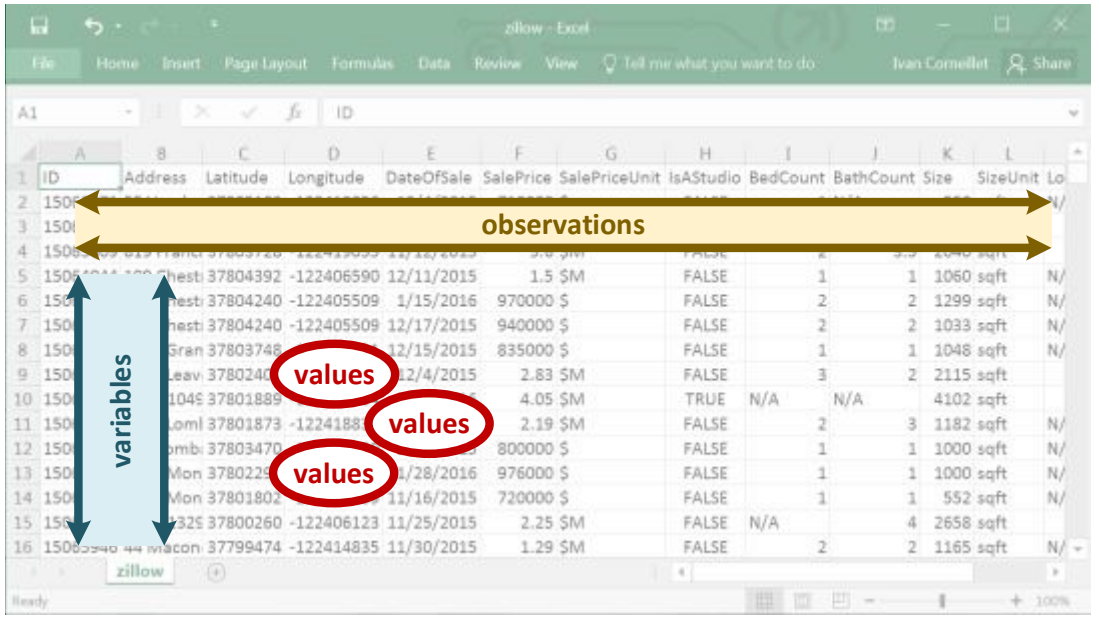
Tidy Data and pandas

Codealong | Introduction to pandas

Codealong | Tidying up (more) the SF housing dataset

❷ PARSE the Data | Tidy Data: a tabular format suitable for *pandas* and machine learning algorithms

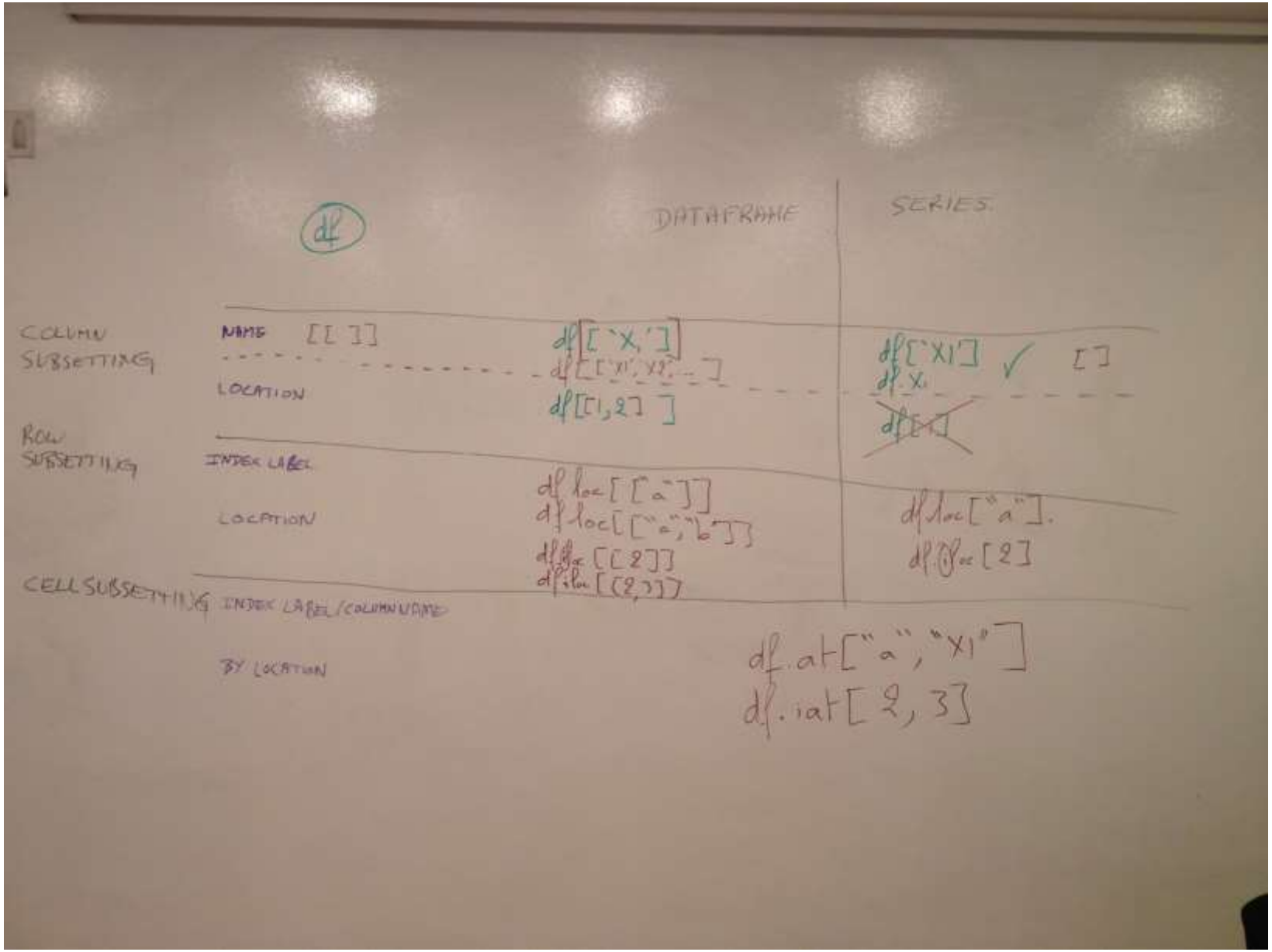
- The three rules of tidy data:
  - Each observation is placed in its own row
  - Each variable in the dataset is placed in its own column
  - Each value is placed in its own cell



Activity | Subsetting with *pandas* (5 minutes)

EXERCISE

	DataFrame	Series
Column subsetting		
by name	?	?
by location	?	?
Row subsetting		
by index label	?	?
by location	?	?
Cell subsetting/scalar lookup		
By index label/column name	?	
By location	?	


df
DATAFRAME
SERIES
COLUMN SUBSETTING
NAME [[ ]]
df[['X']]
df[['X1','X2',...]]
df[[1,2]]
df['X1']
df.X1
[]
LOCATION
ROW SUBSETTING
INDEX LABEL
df.loc[['a']]
df.loc[['a','b']]
df.loc['a'].
df.loc[2]
LOCATION
CELL SUBSETTING
INDEX LABEL/COLUMN NAME
df.at['a','X1']
BY LOCATION
df.iat[2,3]

	DataFrame	Series
Column subsetting		

by name
 	
(Columns names are stored in df.columns)
(df.columns.get_loc('X1') returns X1's column index)
	`# New DataFrame with column named X1`
	`df[ ['X1'] ]`
	
	`# 2+ columns (in the order listed)`
	`df[ ['X1', 'X2', …] ]`
		`df['X1']`
		
		`df.X1`

by location	`# New DataFrame with column at location column_i (numbering starts at 0)`
	`df[ [column_i] ]`
	
	`# 2+ columns (in the order listed)`
	`df[ [column_i, column_j, …] ]`	

Row subsetting		

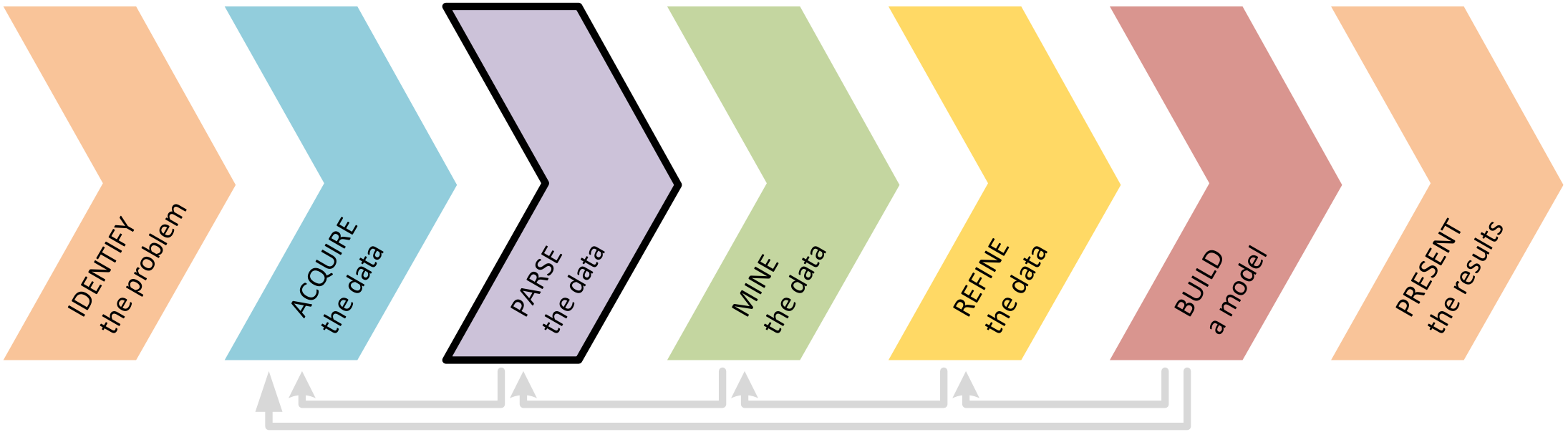
by index label	`df.loc[ [index_label_i] ]`
	`df.loc[ [index_label_i, index_label_j, …] ]`
	
	`# Can use a range if the index is made of numbers (rows "a" to "b" included)`
	`df.loc[ index_label_a : index_label_b ]`
		`df.loc[index_label_i]`

by location	`df.loc[ [row_i] ]`
	`df.loc[ [row_i, row_j, …] ]`
	
	`# (rows "a" to "b' excluded)`
	`df.iloc[row_a : row_b ] or df[row_a : row_b ]`
		`df.iloc[location_i]`

Cell subsetting/scalar lookup		
By index label/column name	`df.at[index_label, 'X1']`	
By location	`df.iat[row_i, column_j]`	

Today

Today we'll keep our focus on PARSE the data



Today, we are covering Research Design and introducing the *pandas* library

Research Design and Data Analysis	Research Design	Data Visualization in *pandas*	Statistics	Exploratory Data Analysis in *pandas*
Foundations of Modeling	Linear Regression	Classification Models	Evaluating Model Fit	Presenting Insights from Data Models
Data Science in the Real World	Decision Trees and Random Forests	Time Series Data	Natural Language Processing	Databases

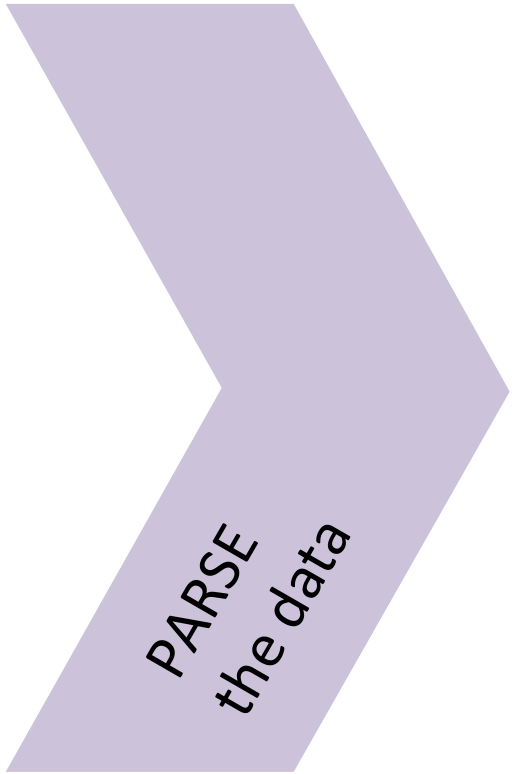
Here's what happening today:

- Unit Project 1 due today
- Announcements and Exit Tickets
- Guest Speaker
- Review
- ❸ Parse the Data
  - Types of Data and Types of Measurement Scales
  - Populations and Samples; Descriptive vs. Inferential Statistics
  - Measures of Central Tendency and Measures of Dispersion
  - Boxplots
  - Outliers
  - Histograms
  - Measurement Errors
  - Correlation
- Review
- Unit Project 2 (due in 1 week)

Q & A

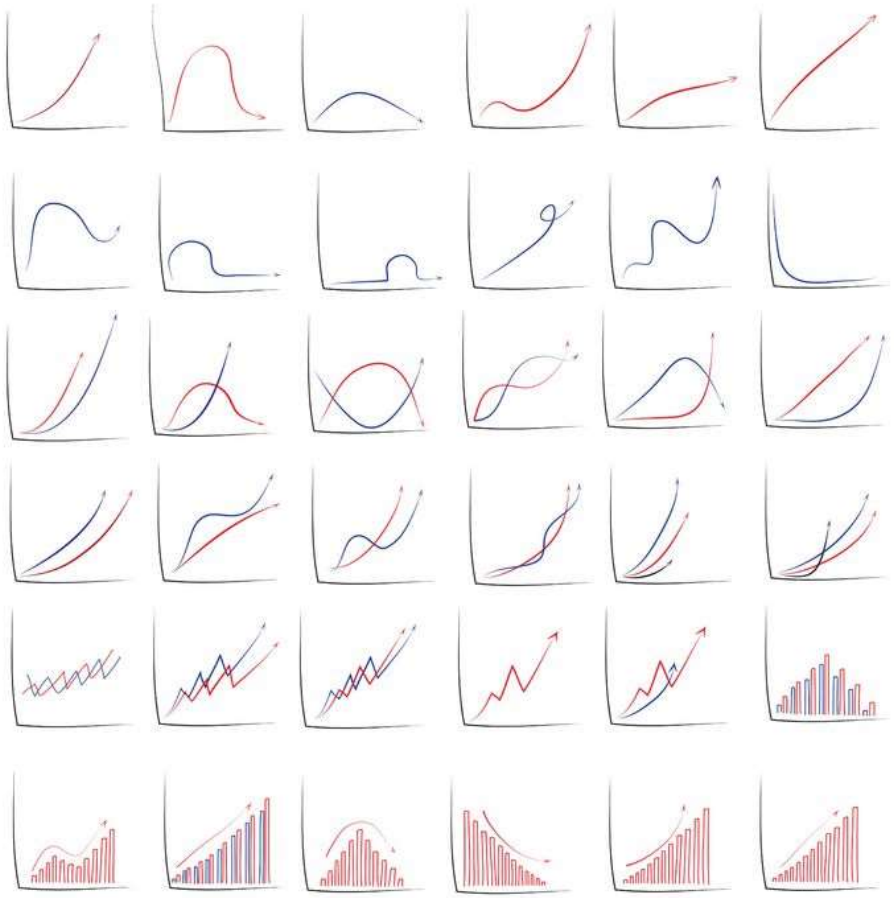
❸ PARSE the Data

❸ Parse the Data



- Parse the Data
  - *Read any documentation provided with the data (session 2)*
  - **Perform exploratory data analysis (session 3)**
    - *Verify the quality of the data (sessions 2/3)*

The main theme today is to have enough statistics knowledge to perform Exploratory Data Analysis



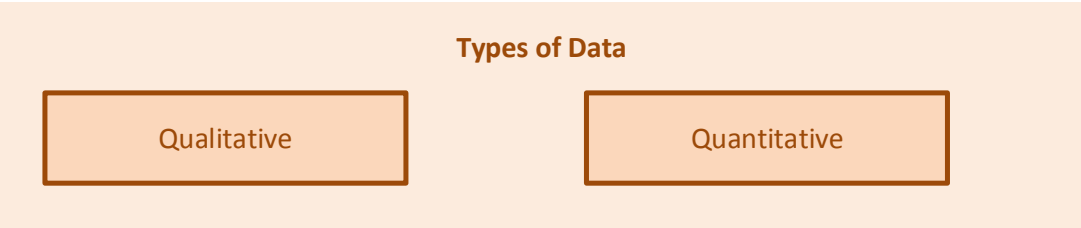
Napat Polchoke © 123RF.com

- Types of Data and Types of Measurement Scales
- Populations and Samples; Descriptive vs. Inferential Statistics
- Measures of Central Tendency and Measures of Dispersion
- Boxplots
- Outliers
- Histograms
- Measurement Errors
- Correlation

❸ PARSE the Data

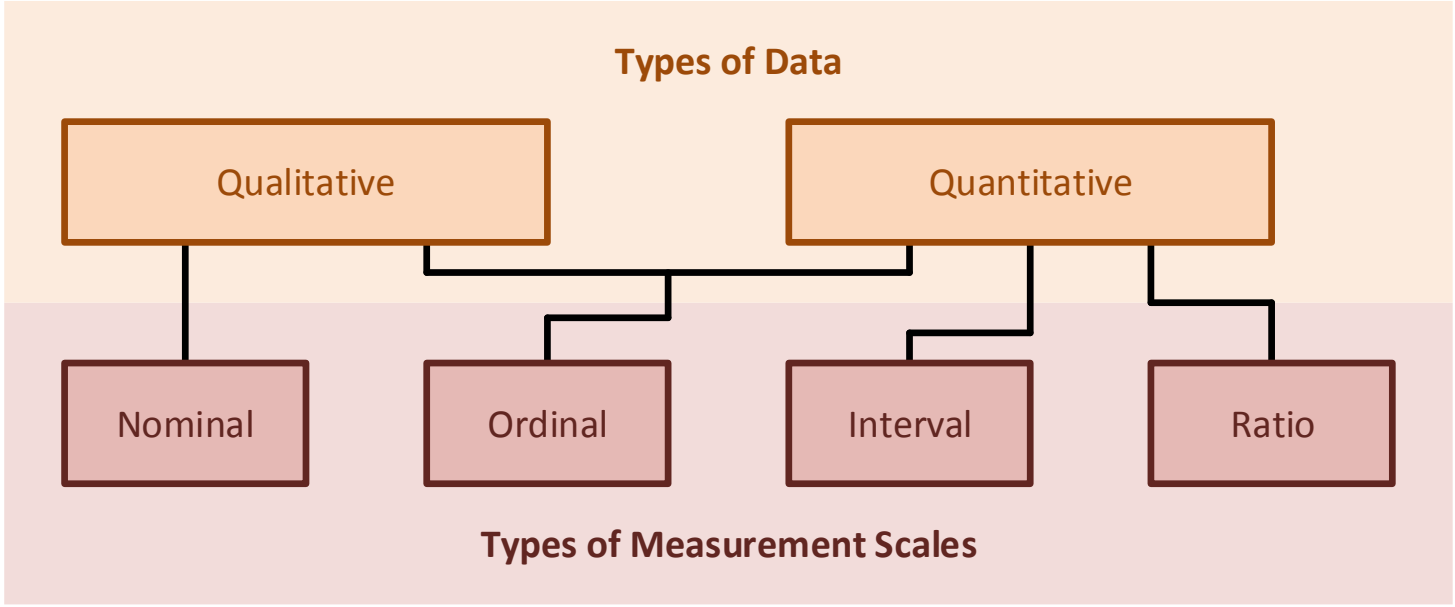
Types of Data and
Types of Measurement Scales

Types of Data



- Qualitative Data
  - Uses descriptive terms to measure or classify something of interest, e.g., education level
- Quantitative Data
  - Uses numerical values to describe something of interest, e.g., age

Types of Measurement Scales



Types of Measurement Scales (cont.)

	Nominal	Ordinal	Interval	Ratio
e.g.	Gender	Movie ratings	Temperature	Salary
Categorize?	✓ (male, female)	✓	✓	✓
Rank-order?	✘	✓ (★<2★<3★<4★)	✓	✓
Add and subtract?	✘	✘ (4★−3★≠★)	✓ (75°C is 50°C warmer than 25°C)	✓
Multiply and divide?	✘	✘ (4★ not 4× better than 1★)	✘ (75°C not 3× as warm as 25°C) (0°C doesn't mean no temperature!)	✓ (Salary of $200K is 2× that of $100K) ($0 means no salary ☹)

❸ PARSE the Data

Activity | Knowledge Check

Activity | Knowledge Check

EXERCISE

DIRECTIONS (5 minutes)

1. What type of data are the columns in the Zillow dataset?
    a. Zillow ID
    b. Address
    c. Date of Sale
    d. Sale Price
    e. Whether it is a Studio
    f. Number of beds
    g. Number of baths
    h. Size
    i. Lot Size
    j. Year it was built
2. When finished, share your answers with your table

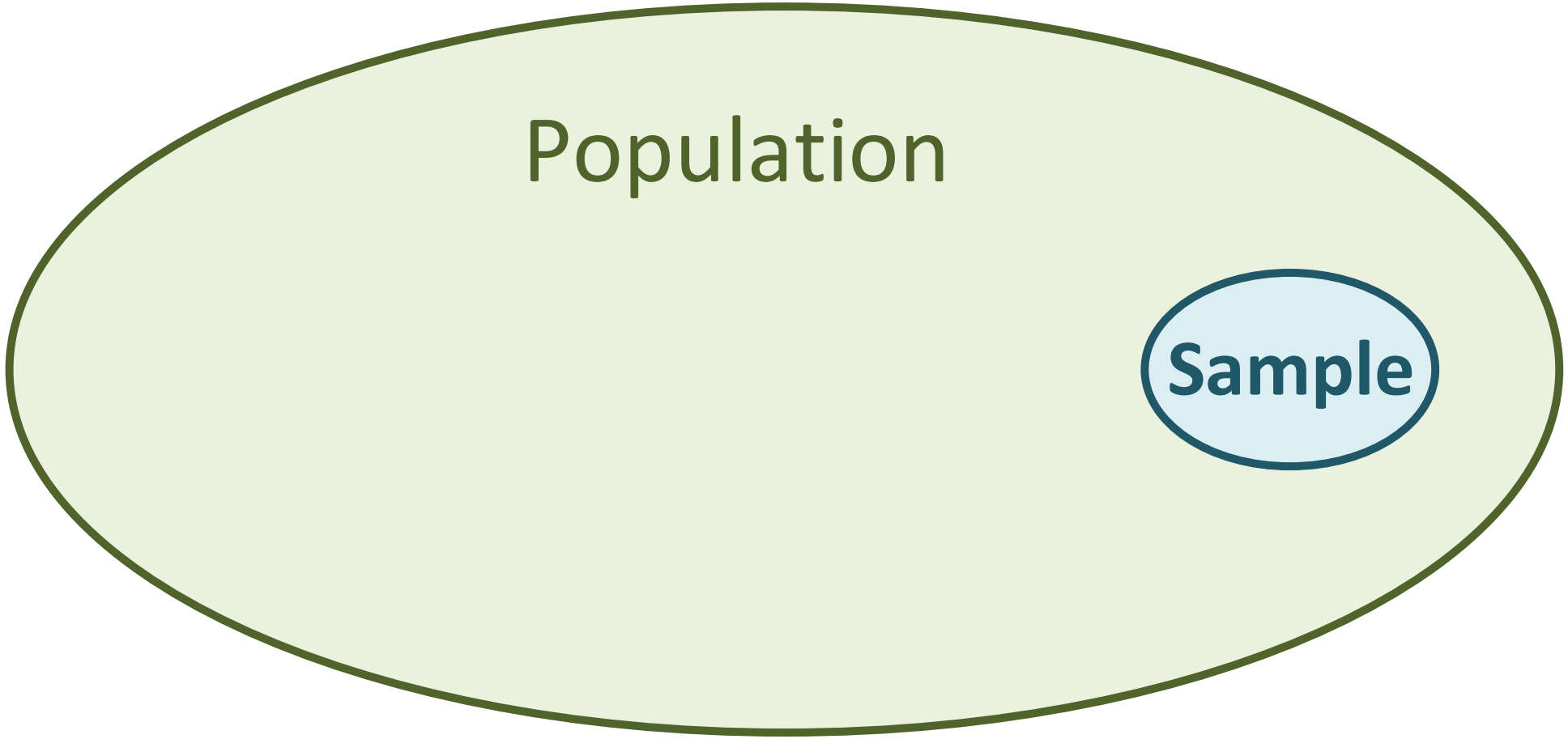
DELIVERABLE

Answers to the above questions

❸ PARSE the Data

Populations and Samples

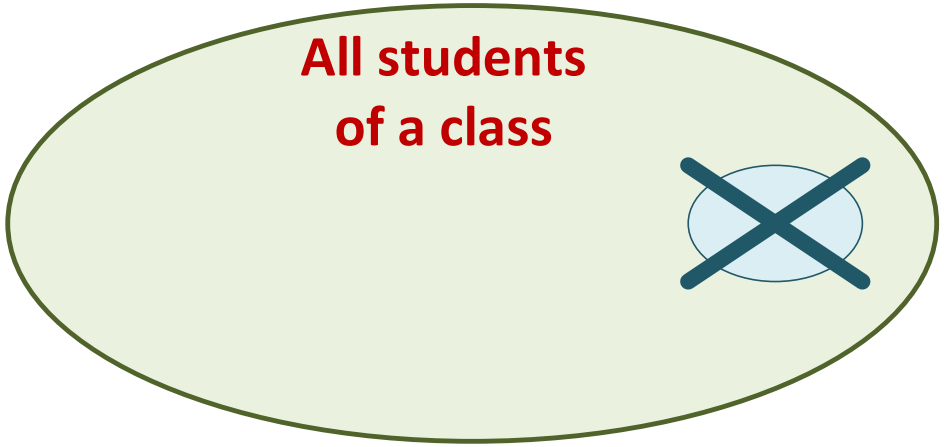
Populations and Samples



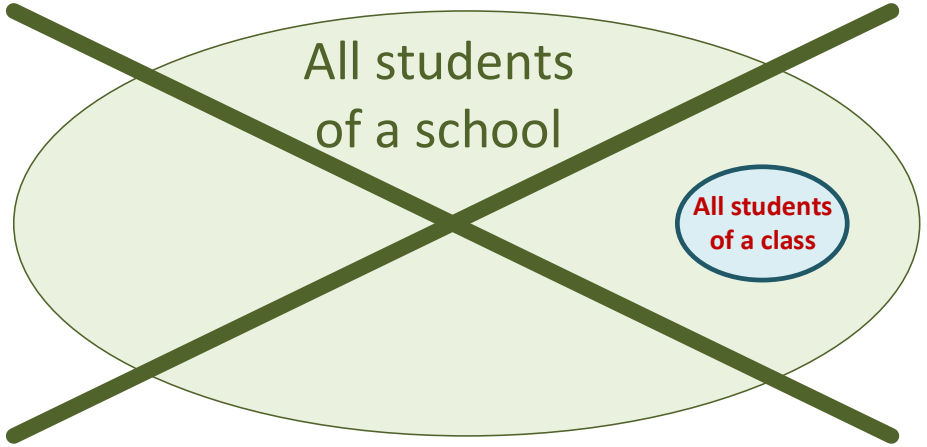
A dataset may be considered either as a population or a sample, depending on the reason for its collection and analysis

- Students of a class are a population if the analysis describes the distribution of scores in that class
- But they are a sample the analysis infers from their scores the scores of other students (e.g., all students from that school)

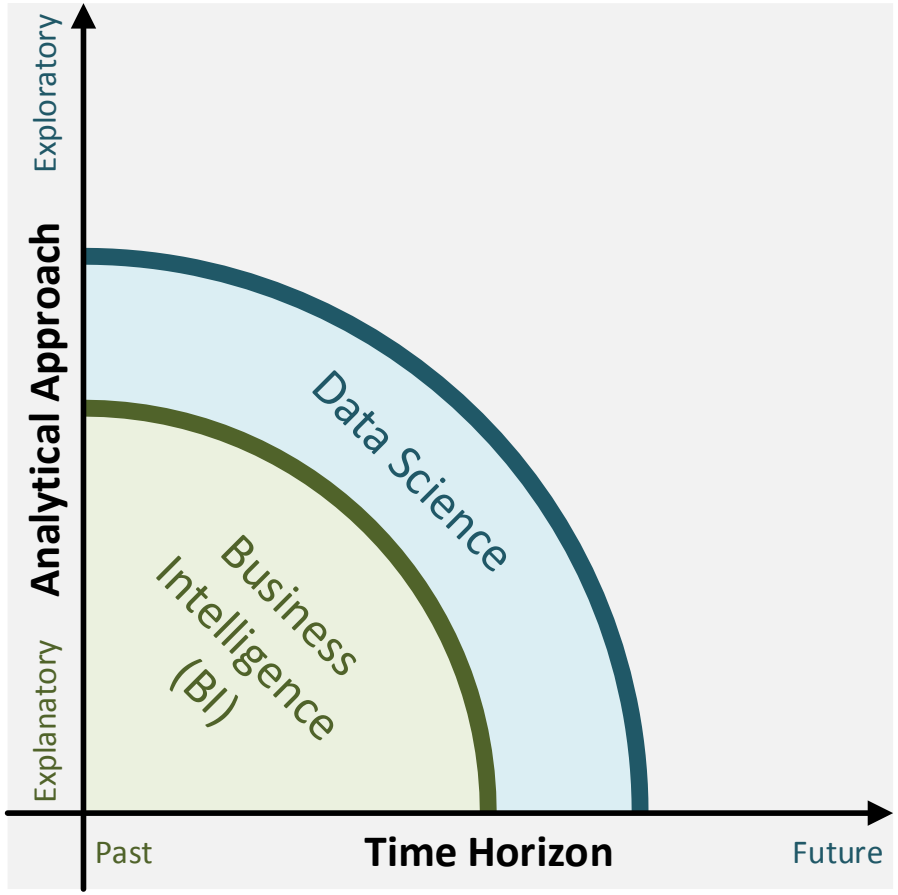
Descriptive Statistics



Inferential Statistics



Data Science and Business Intelligence; Population and Sample; Which is which?

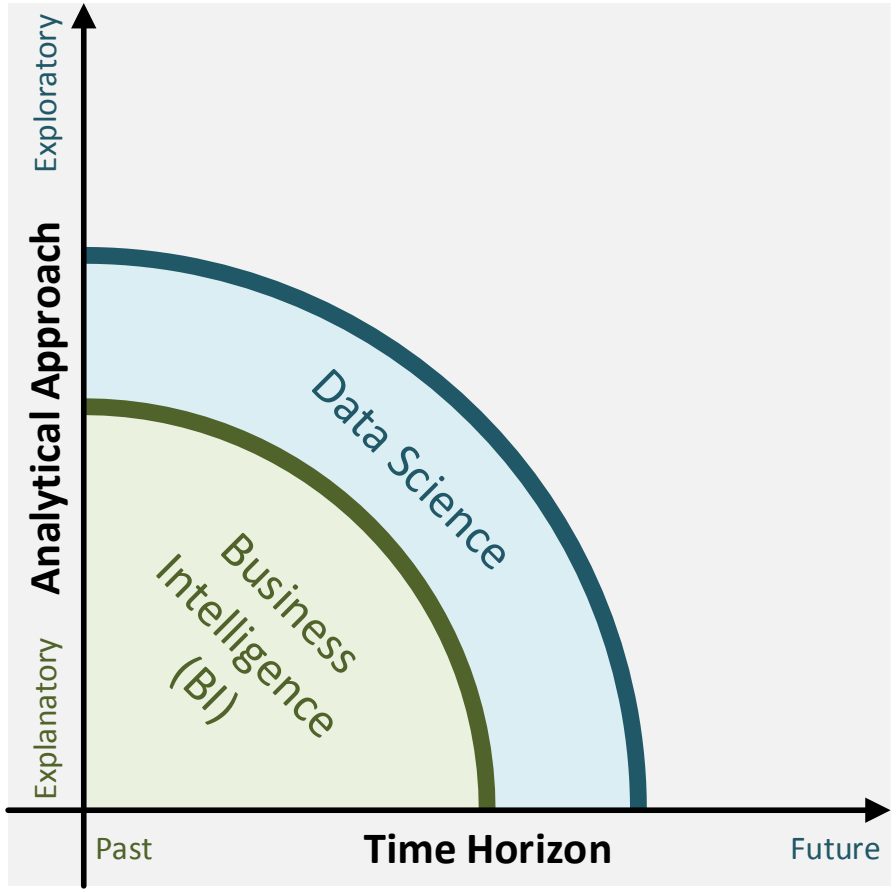


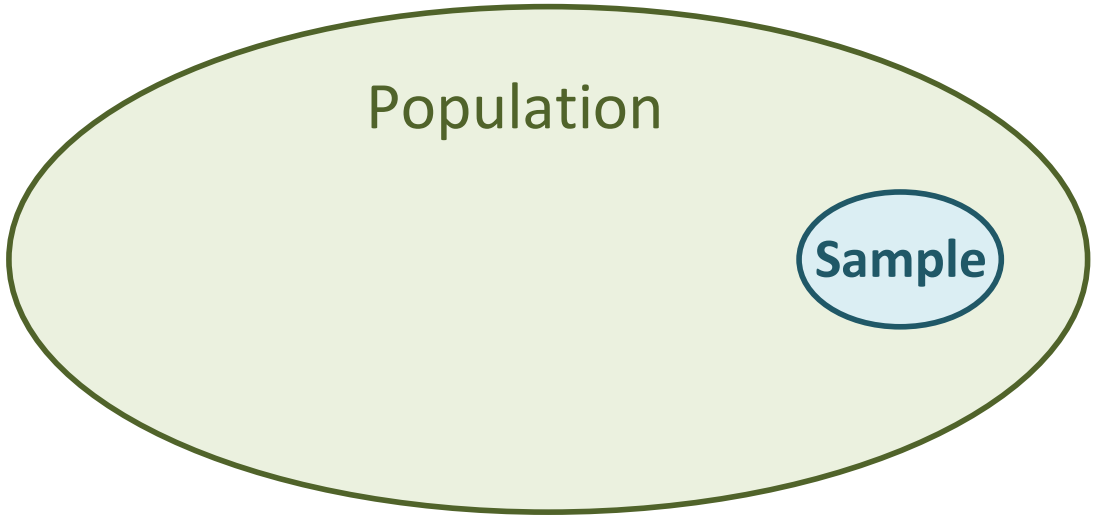
Data Science (Data Mining and Predictive Analytics)	
Common questions	• What if …? • What's the optimal scenario for our business? • What will happen next? What if these trends continue? Why is it happening?

Business Intelligence (BI)	
Common questions	• What happened last quarter? • How many units were sold? • Where is the problem? In which situations?

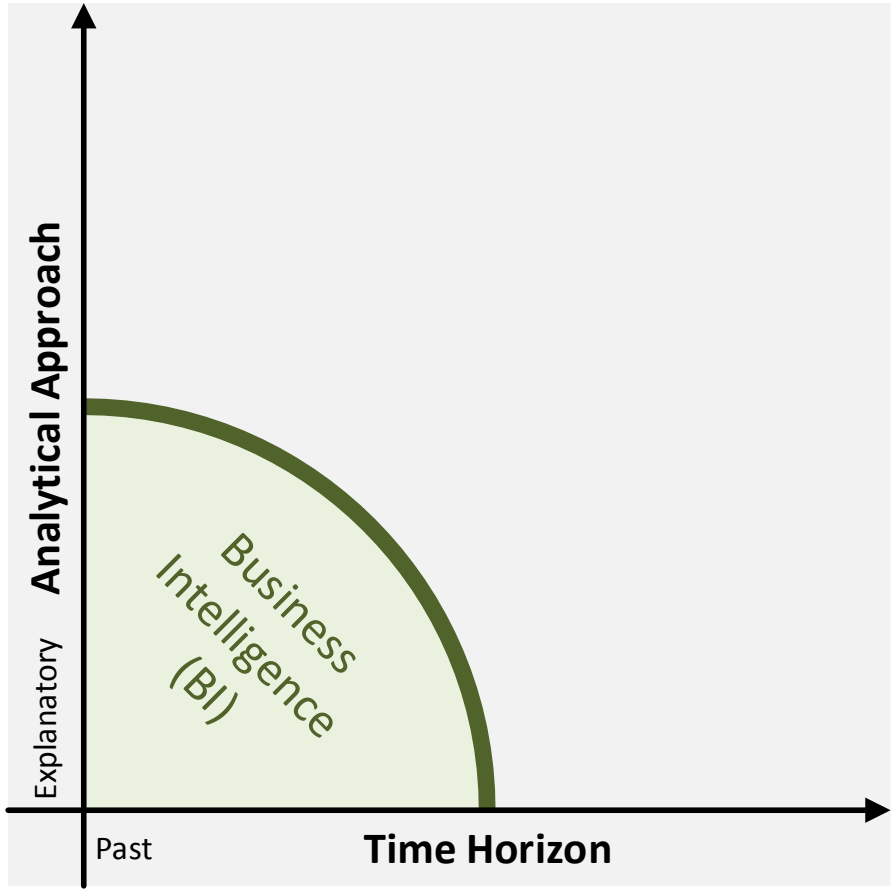
Source: Data Science and Big Data Analytics

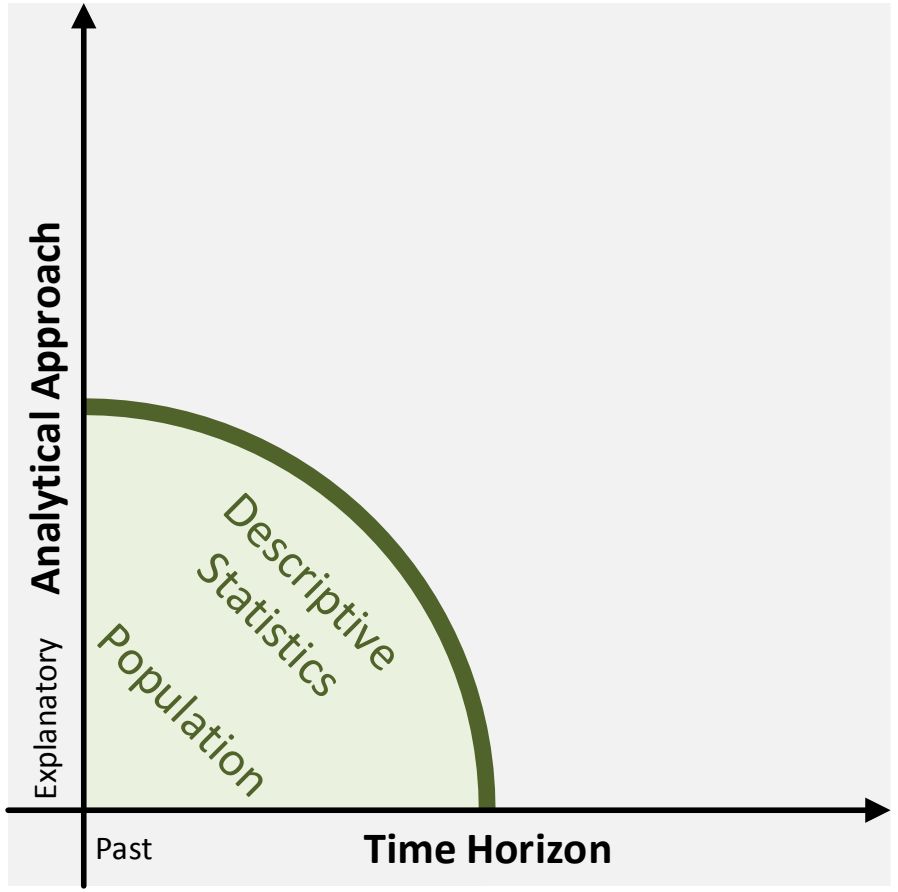
Data Science and Business Intelligence; Population and Sample; Which is which? (cont.)



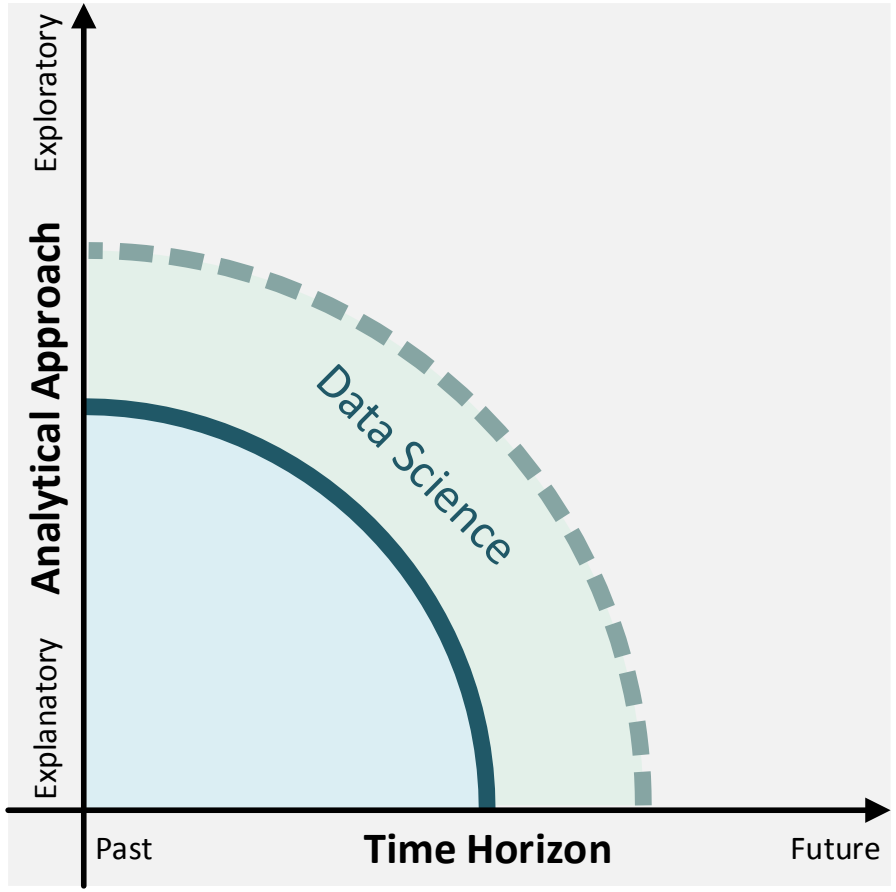


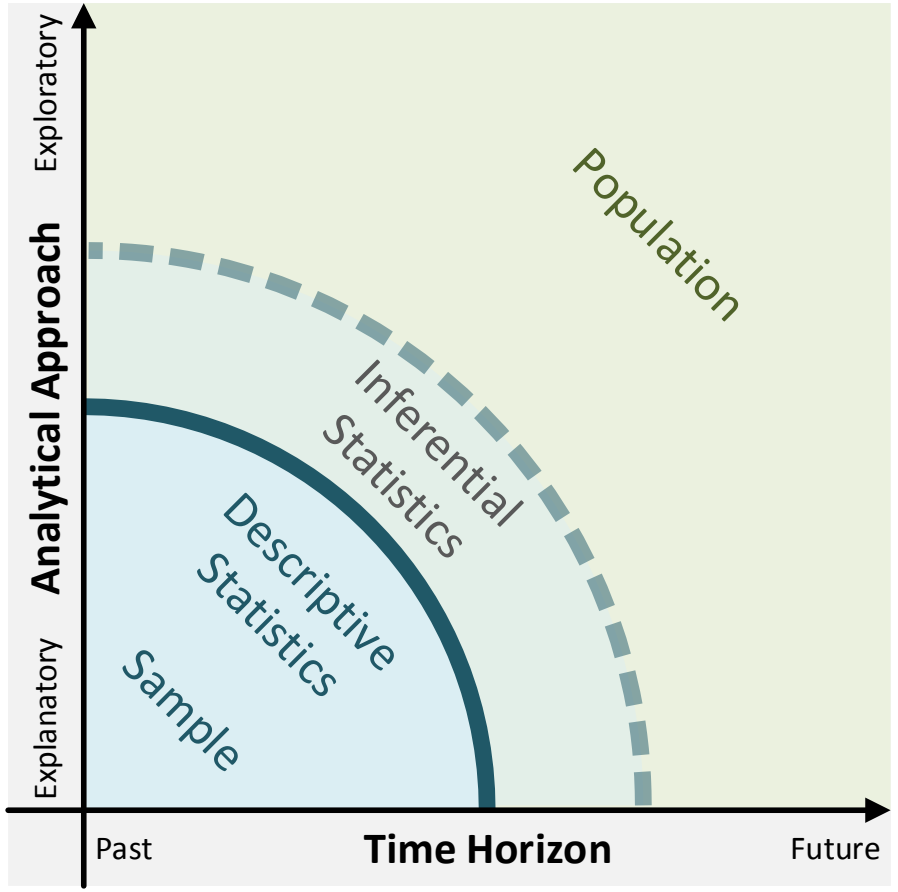
Business Intelligence is concerned with descriptive statistics (e.g., “what happened last quarter?” and “how many units were sold?”)





Data Science concerns itself with inferential statistics (e.g., “what if …?”, “what will happen next?”, and “what if these trends continue?”)



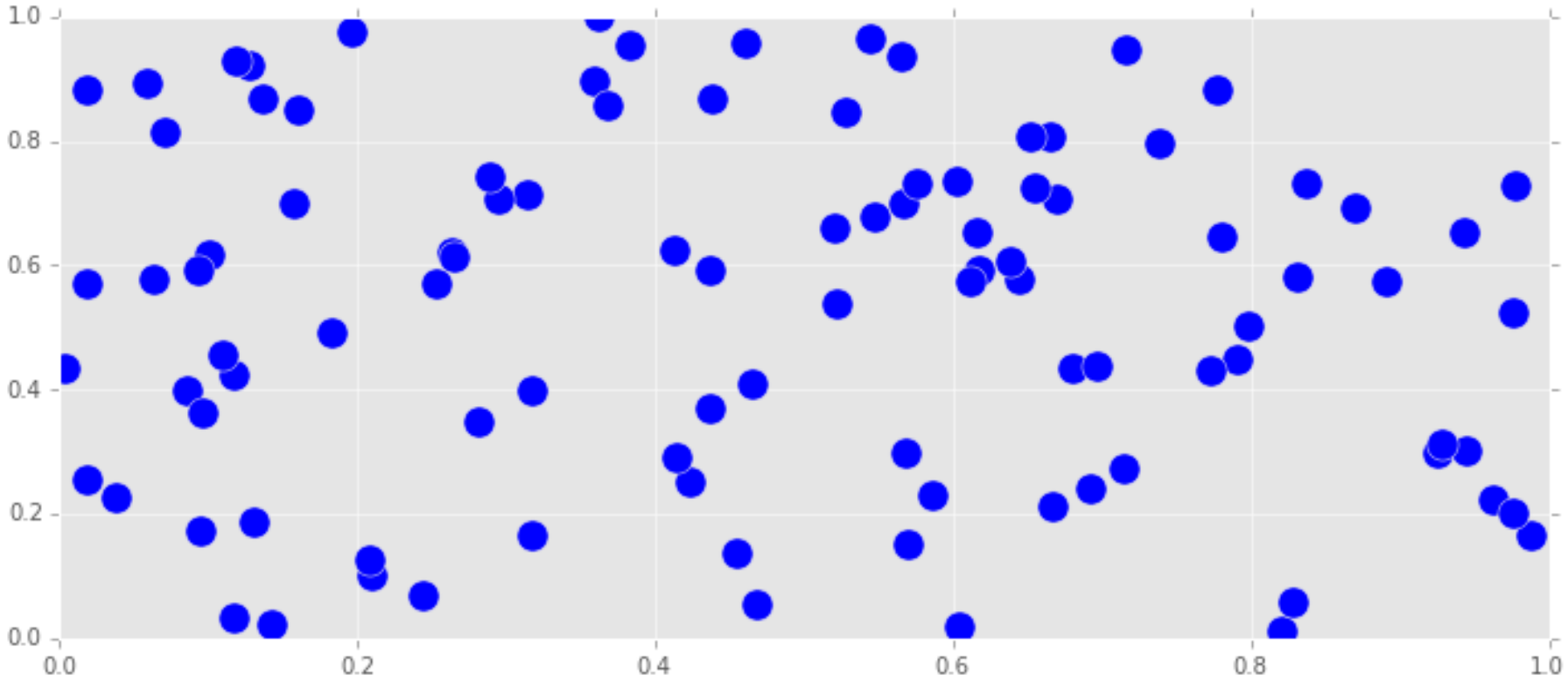


❸ PARSE the Data

Activity | Summarizing Data

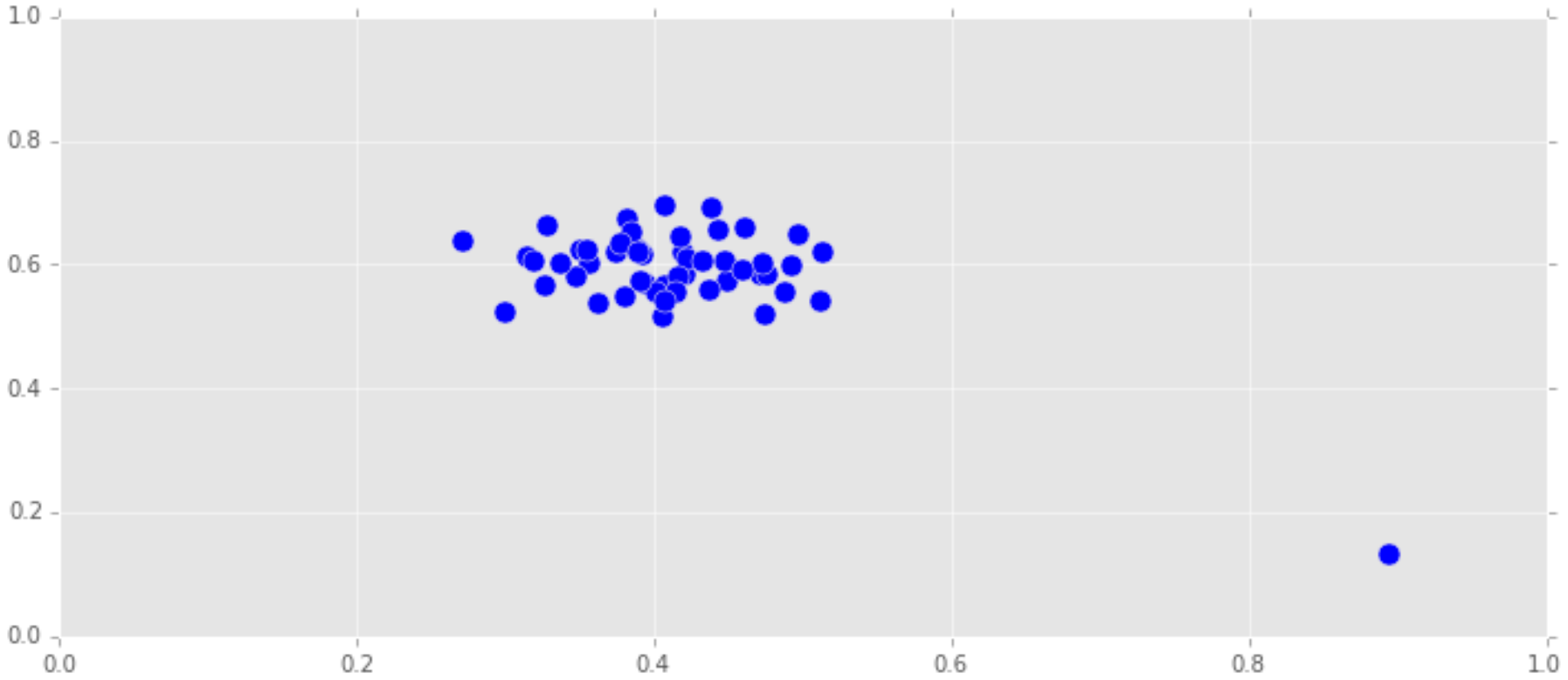
Activity | How would you summarize this data?

EXERCISE



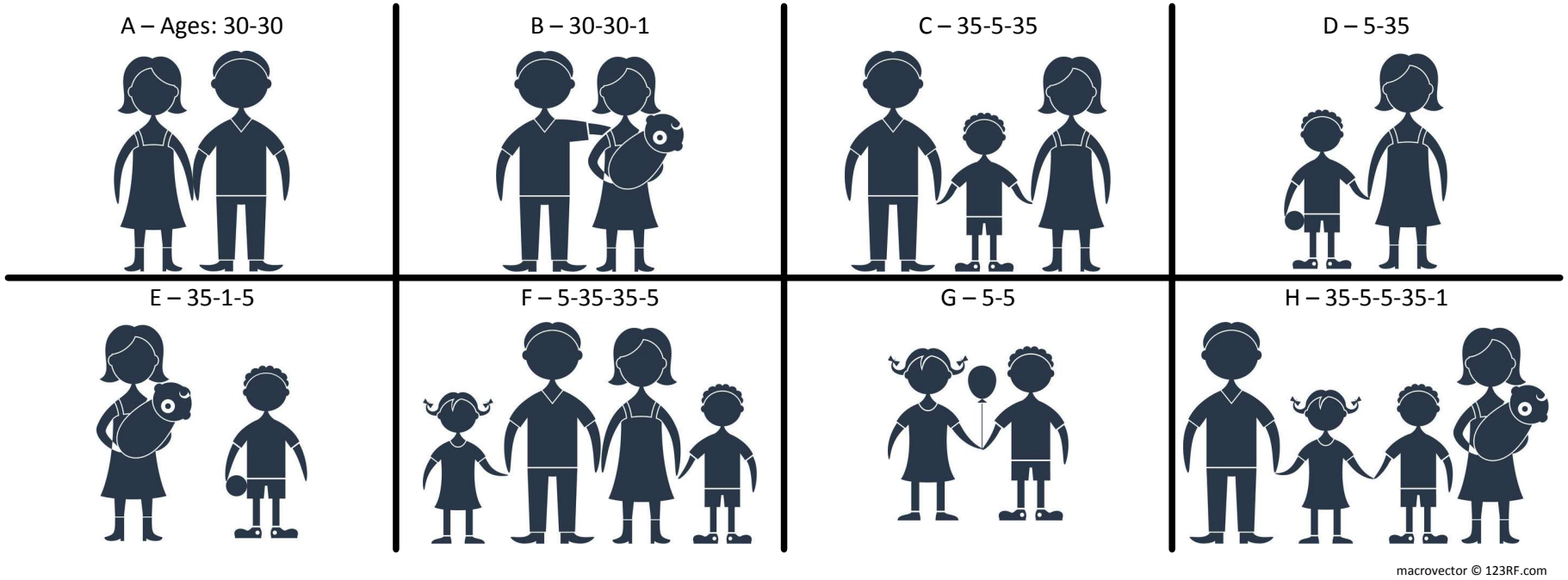
Activity | How would you summarize this data? (cont.)

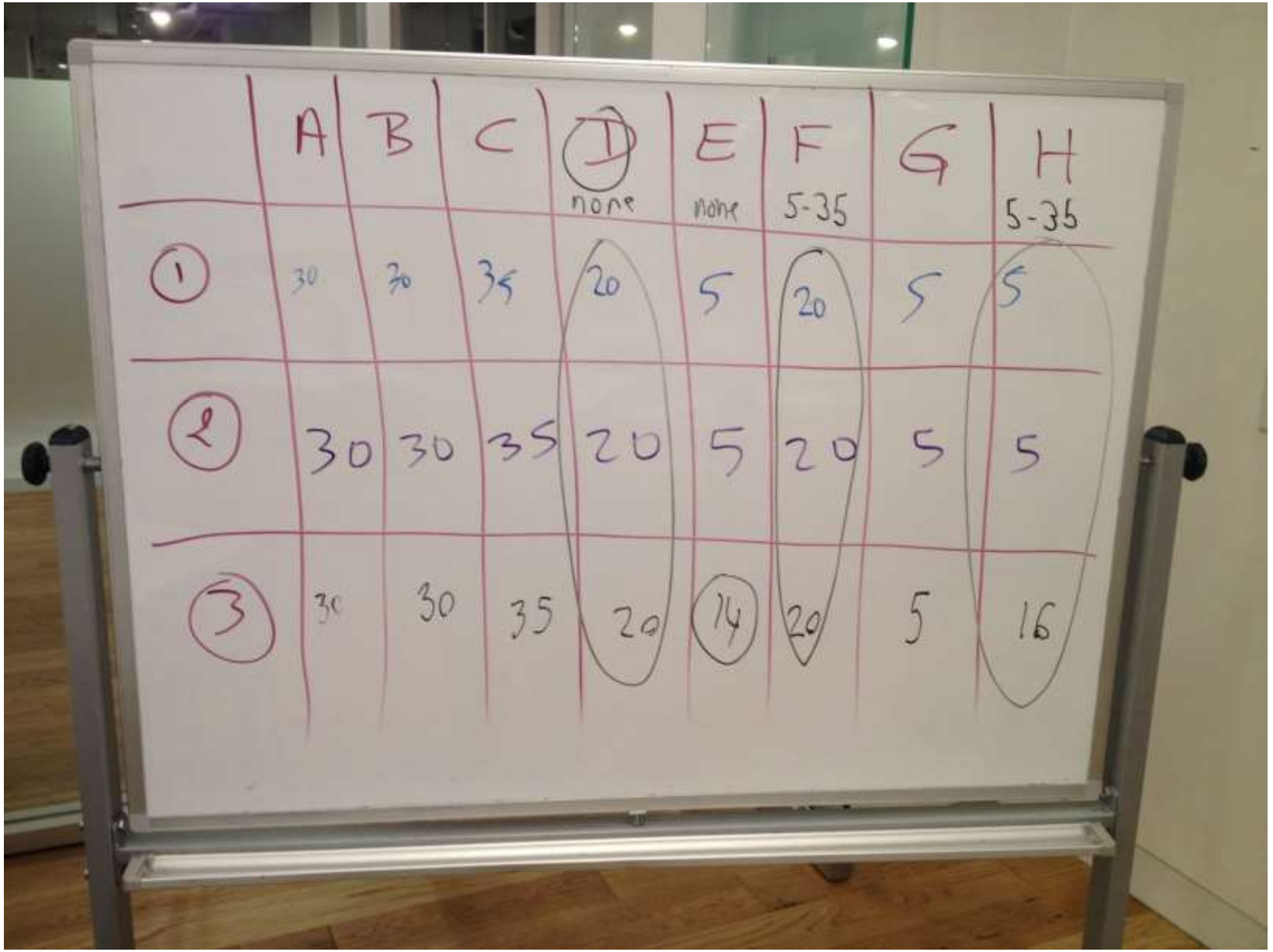
EXERCISE



Activity | Measures of Central Tendency. What is the typical age for these 8 groups of people? (5 minutes)

EXERCISE




A
B
C
D
none
E
none
F
5-35
G
H
5-35
1
30
30
35
20
5
20
5
5
2
30
30
35
20
5
20
5
5
3
30
30
35
20
14
20
5
16

Activity | What is the typical age for these 8 groups of people? (cont.)

Group	Mean	Median	Mode
A (30-30)	30[1]	30[1]	30[1]
B (30-30-1)	20.3[2] (i.e., no 20-year-olds in the group)	30[3]	30[3]
C (35-5-35)	25[2]	35[3]	35[3]
D (5-35)	20[2]	20[2]	None[4]
E (35-1-5)	13.6[2]	5[2]	None[4]
F (5-35-35-5)	20[2]	20[2]	5 and 35[5]
G (5-5)	5[1]	5[1]	5[1]
H (35-5-5-35-1)	16.2[2]	5[6]	5 and 35[5]

[1] All values are equal [2] Value not representative [3] Follow the "majority" [4] All values are different [5] Follow the "majorities" [6] Partially correct

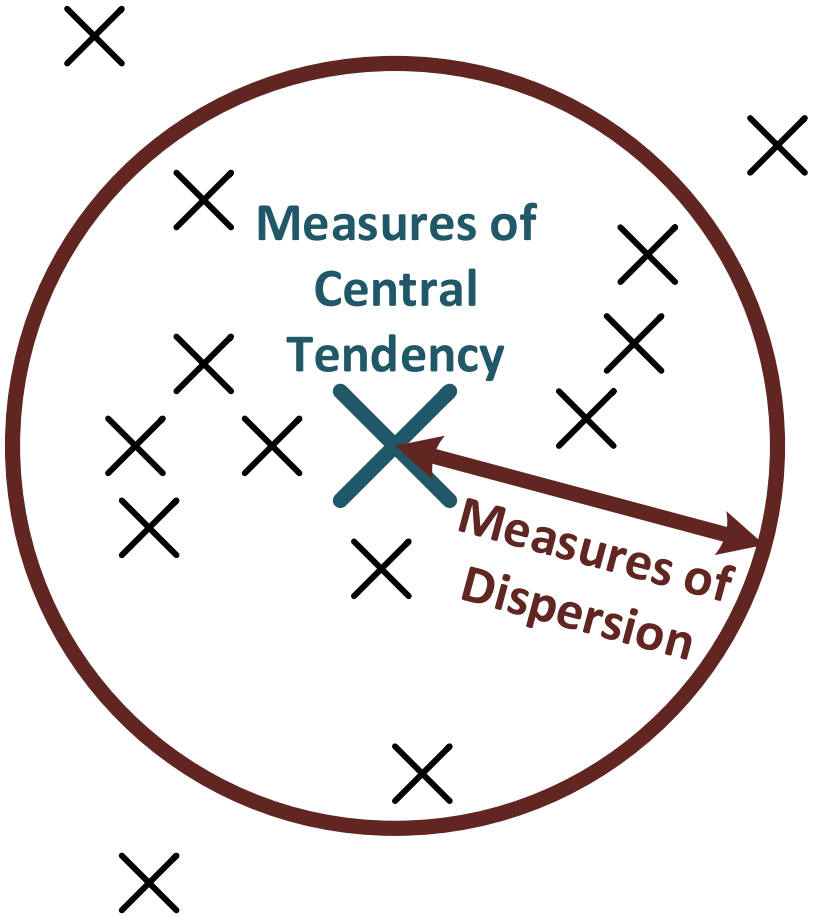
Mean, Median, and Mode: There is no "Winner-Take-All"

	Value is in the dataset	Value is easy to compute	Value is resistant to outliers	Corresponding measure of Dispersion	Used extensively by mathematical models
Mean	☹ (Unlikely)	☺	☹	☺ (Variance, Standard Deviation)	☺
Median	😐 (50% chance)	😐 (need to rank the values)	☺	☺ (Interquartile Range)	☹
Mode	☺ (Always)	☹ (Need to count and rank the count)	☺	☹ (Not really)	☹ (Mode might not be defined or you might have multiple values)

❸ PARSE the Data

Measures of Central Tendency and Measures of Dispersion

Measures of Central Tendency and Measures of Dispersion



- Measures of Central Tendency
  - (Or measures of location)
  - Answer the question: “What’s the typical or common value for a variable?”
  - Mean, Median, Mode
- Measures of Dispersion
  - (Or measures of variability/spread)
  - Answer the question: “How far do values stray from the typical value?”
  - Variance, Standard Deviation, Range, Interquartile Range (IQR)

(Arithmetic) Mean, Variance, and Standard Deviation

Ordinal ✘	Nominal ✘	Interval ✓	Ratio ✓

	Population	Sample
(Arithmetic) Mean *(a.k.a., the first moment)* (Mean has unit of X:$[X]$)	$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i = E[X^1]$ (mu)	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (x-bar)
Variance *(a.k.a., the second moment)* $[X^2]$	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$ $= E[(X-\mu)^2]$ (sigma-squared)	$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$
Standard Deviation $[X]$	$\sigma = \sqrt{\sigma^2}$ (sigma)	$s = \sqrt{s^2}$

(mean, variance, and standard deviations are based on the values of x_i)

❸ PARSE the Data

Codealong – Part A

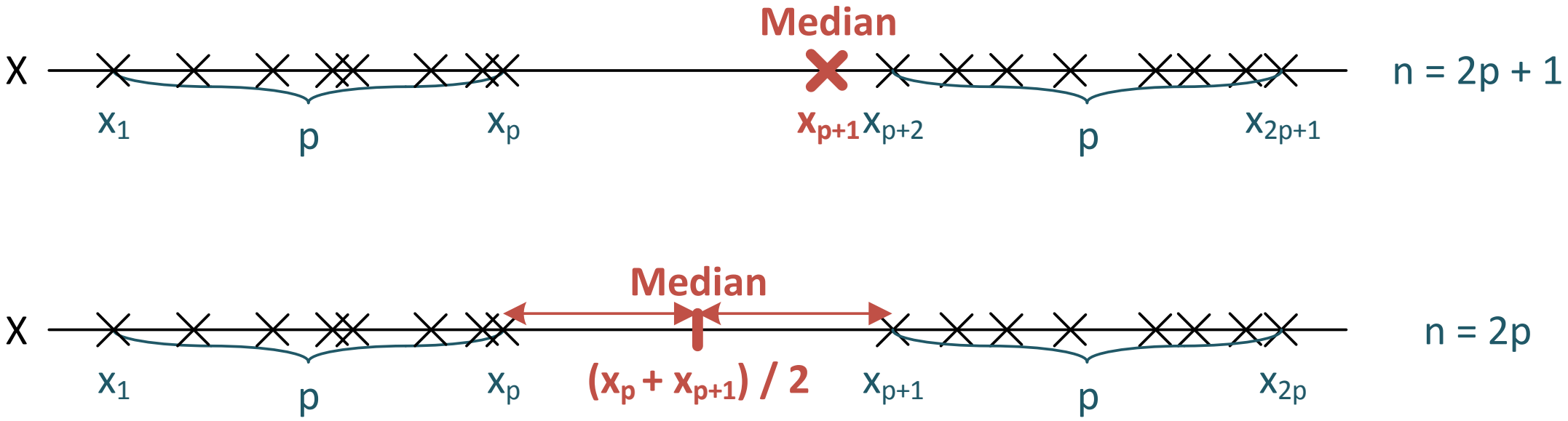
`.mean()`

`.var(), .std()`

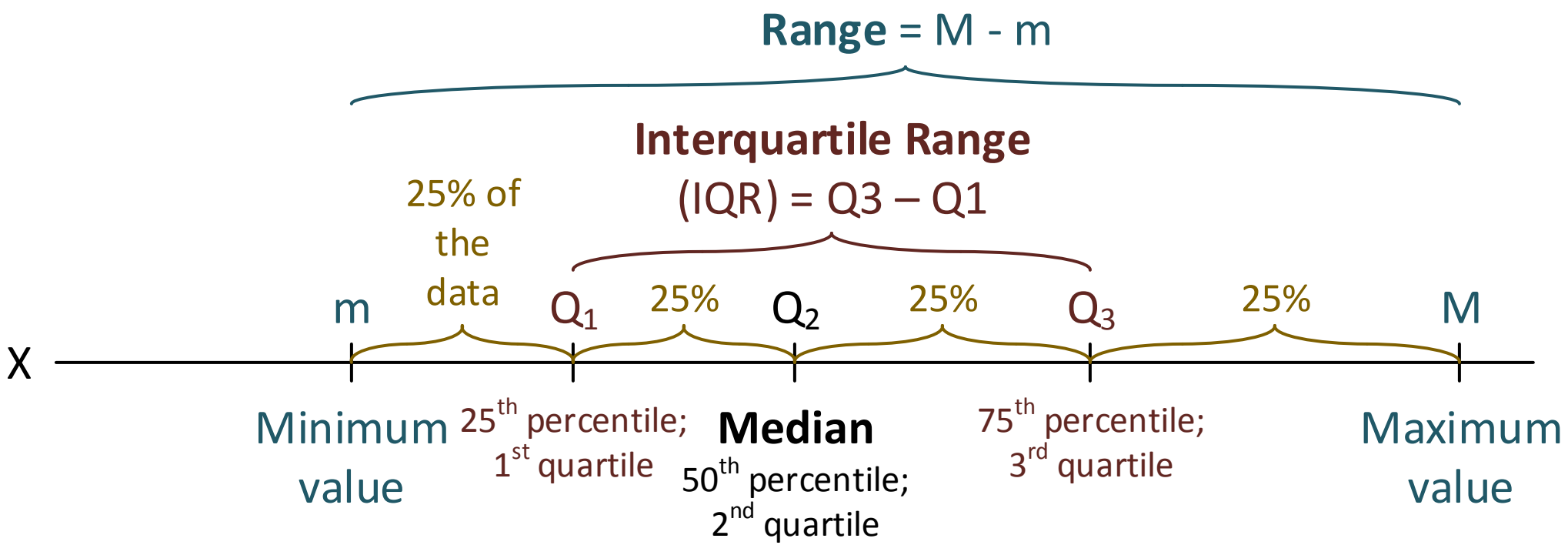
❸ PARSE the Data

Median, Range, and Interquartile Range

Median


Median
x
n = 2p + 1
x1
p
xp
xp+1 xp+2
p
x2p+1
Median
x
n = 2p
x1
p
xp
(xp + xp+1) / 2
xp+1
p
x2p

Median, Range, and Interquartile Range



Median, Range, and Interquartile Range (cont.)

Nominal ✘	Ordinal ✘	Interval ✓	Ratio ✓

Median	$median = \begin{cases} x_{p+1} \; if \; n = 2p+1 \\ \dfrac{x_p + x_{p+1}}{2} \; if \; n = 2p \end{cases}$
Range	$range = x_n - x_1$
Percentile	$q_k = \begin{cases} x_{\lceil p \rceil} \; if \; p = \dfrac{nk}{100} \; not \; integer \\ \dfrac{x_p + x_{p+1}}{2} \; otherwise \end{cases}$
Quartile	$Q_1 = q_{25}; Q_3 = q_{75}$
Interquartile Range	$IQR = Q_3 - Q_1$

(median, range, and interquartile range are based on the ranks of x_i; x_i ranked from smallest to largest)

❸ PARSE the Data

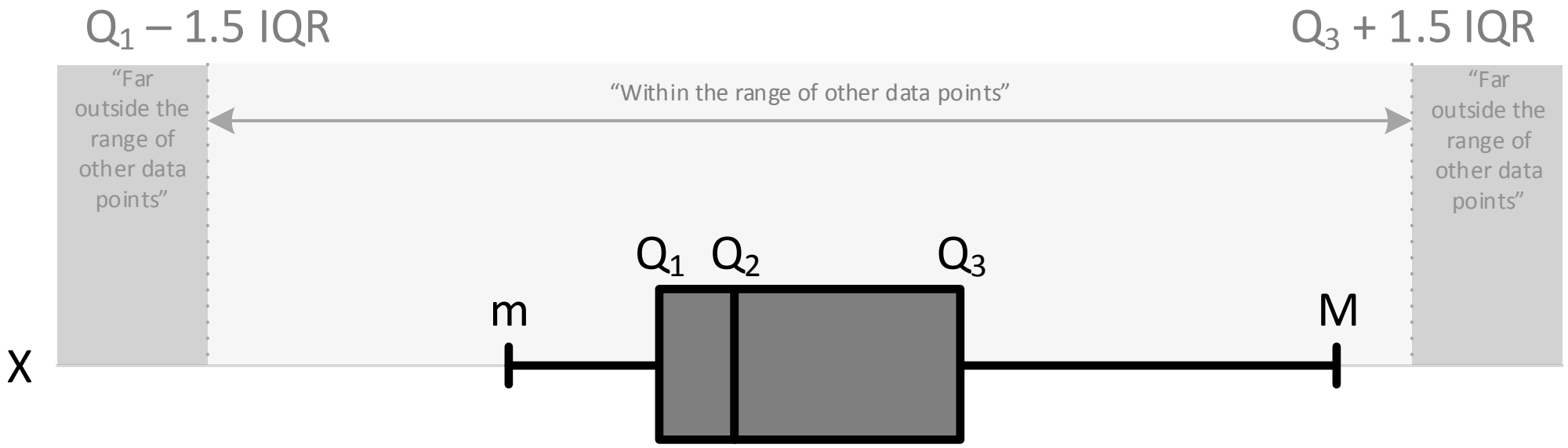
Codealong – Part B

.mean(), .median()
.count(), .dropna(), .isnull()
.min(), .max()
.quantile()
.describe()

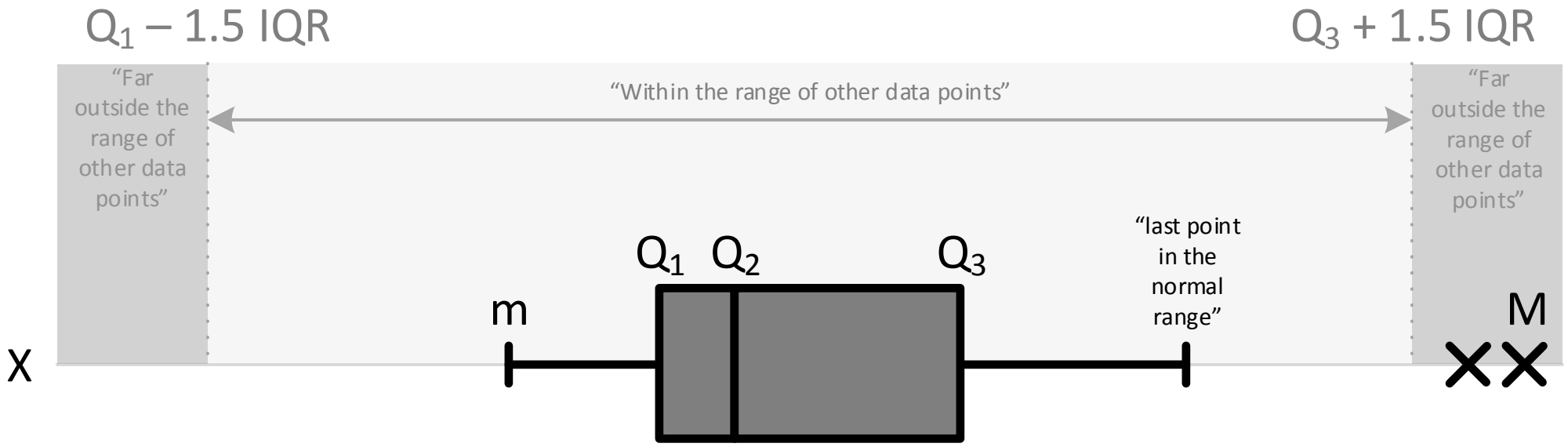
❸ PARSE the Data

Median, Range, Interquartile Range, and Boxplots

Boxplot #1 | Median, Range, Interquartile Range, and no Outliers



Boxplot #2 | Median, Range, Interquartile Range, and Outliers



❸ PARSE the Data

Codealong – Part C

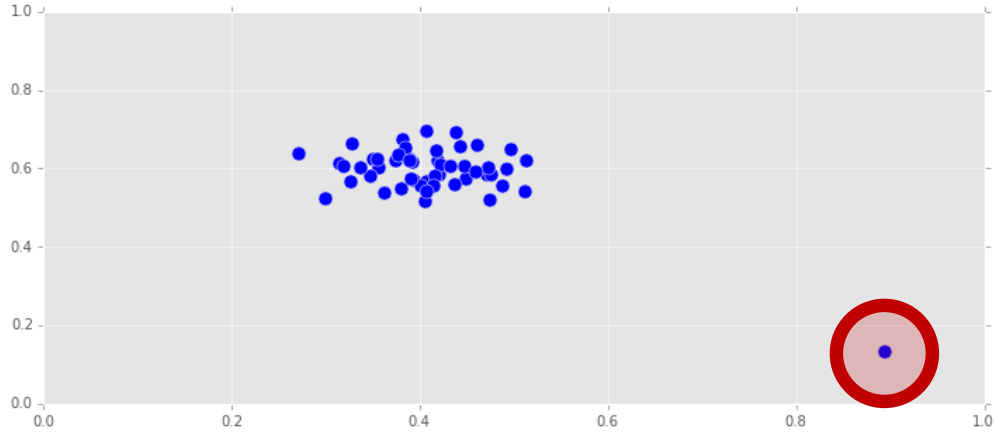
Boxplots

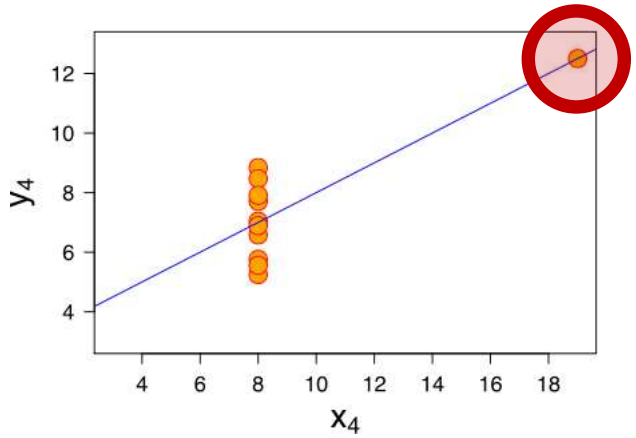
❸ PARSE the Data

Outliers

Think twice before discarding outliers; they might be the most important points

- Outliers are values that are “far” from the central tendency
- No formal definition among statisticians on how to define outliers (how do you define “far”?)
- However, general agreement that they be identified and dealt with appropriately (e.g., keep or discard)
  - They might be the most important points of your dataset

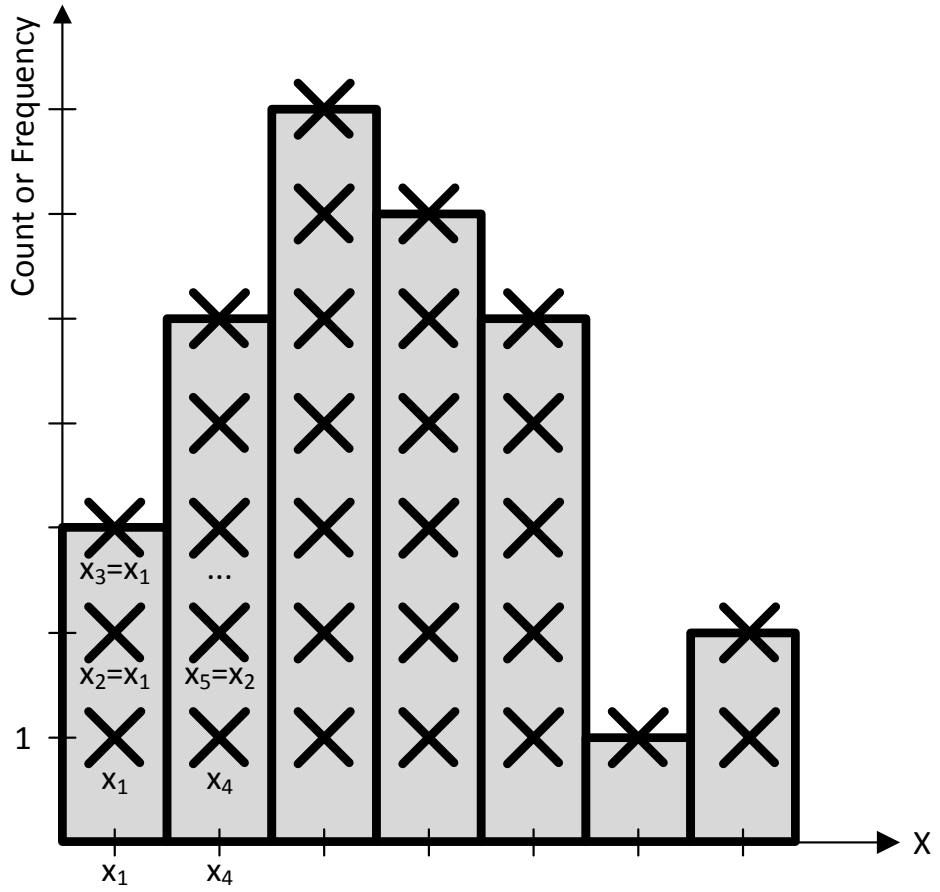




❸ PARSE the Data

Histograms

Histograms. $x_1 = x_2 = x_3 < x_4 = x_5 \ldots$



❸ PARSE the Data

Codealong – Part D

Histograms

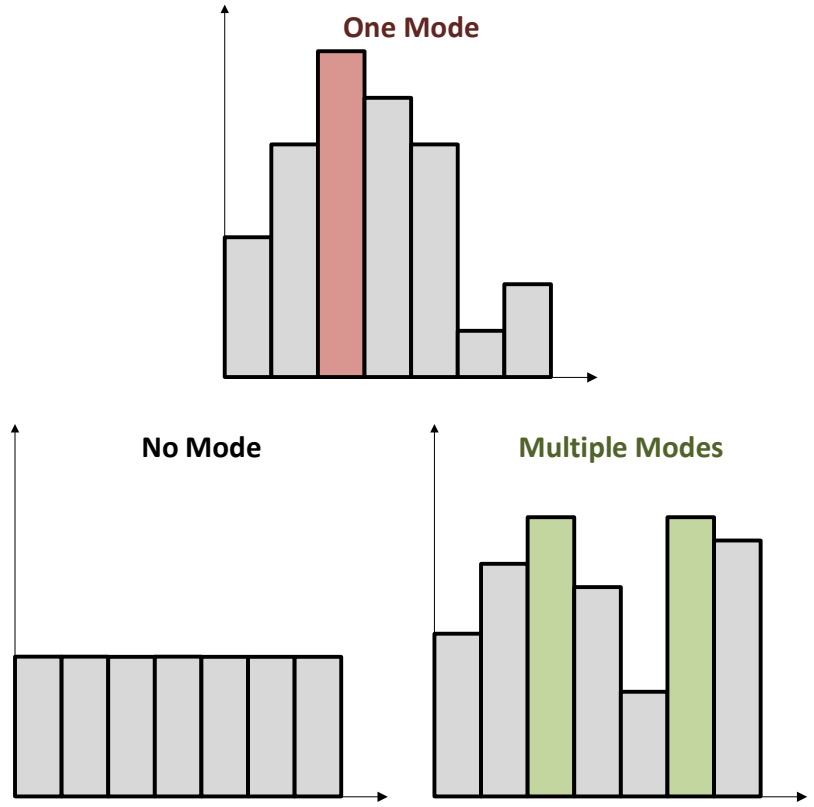
❸ PARSE the Data

Mode

Modes and Histograms

Nominal ✓	Ordinal ✓	Interval ✓	Ratio ✓

- The Mode is the value(s) that occur(s) most often



❸ PARSE the Data

Codealong – Part E

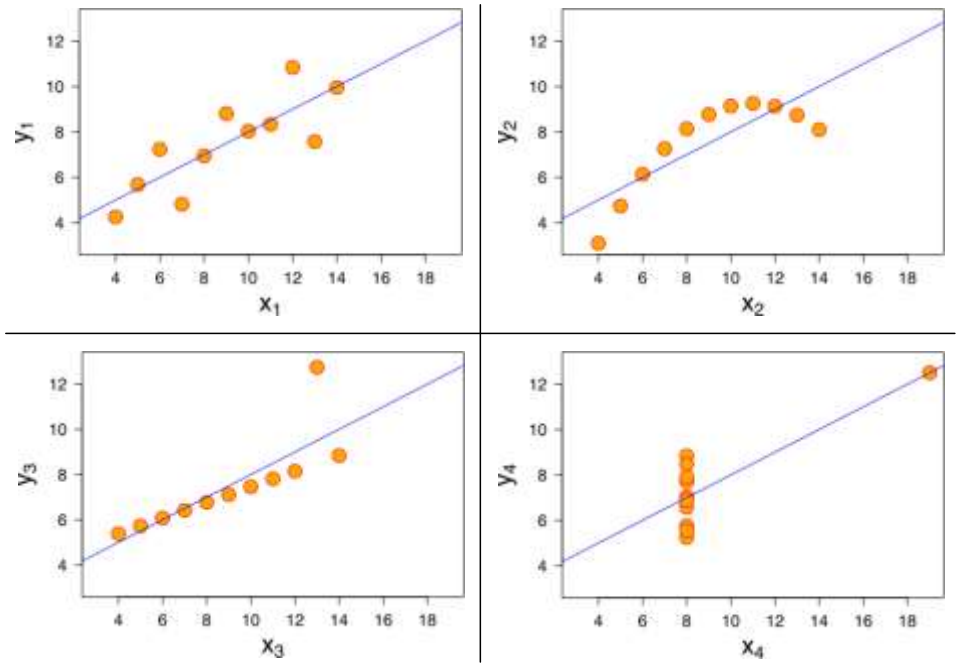
`.mode()`

❸ PARSE the Data

Plot the Data!

Don’t rely on basic statistic properties and **plot the data!** 4 datasets (Anscombe’s quartet) that have nearly identical simple statistical properties, yet are very different

Scatter plot appears to be a simple linear relationship, corresponding to two variables correlated and following the assumption of normality.



Not distributed normally; while an obvious relationship between the two variables can be observed, it is not linear, and the linear correlation is not relevant.

Distribution is linear, but with a different regression line, which is offset by the one outlier which exerts enough influence to alter the regression line.

Example when one outlier is enough to produce a high correlation coefficient, even though the relationship between the two variables is not linear.

Property	Value
Mean of x_i	9
Sample variance of x_i	11
Mean of y_i	7.50
Sample variance of y_i	4.122 or 4.127
Correlation between x_i and y_i	0.816
Linear regression line in each case	$y_i = 3.00 + 0.500\ x_i$

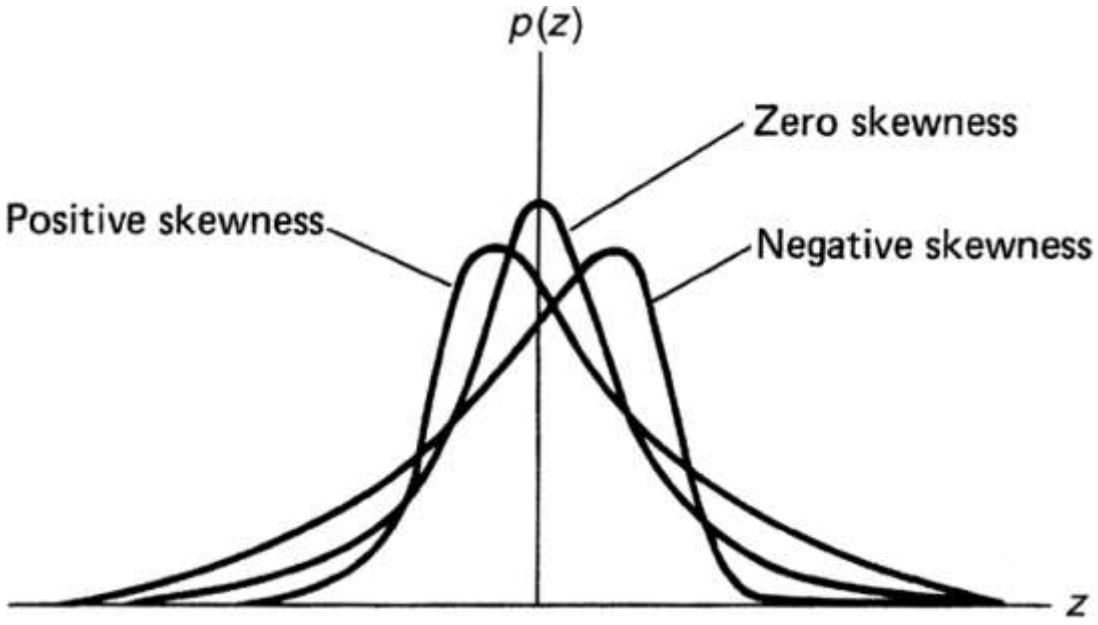
❸ PARSE the Data

Third and Fourth Moments

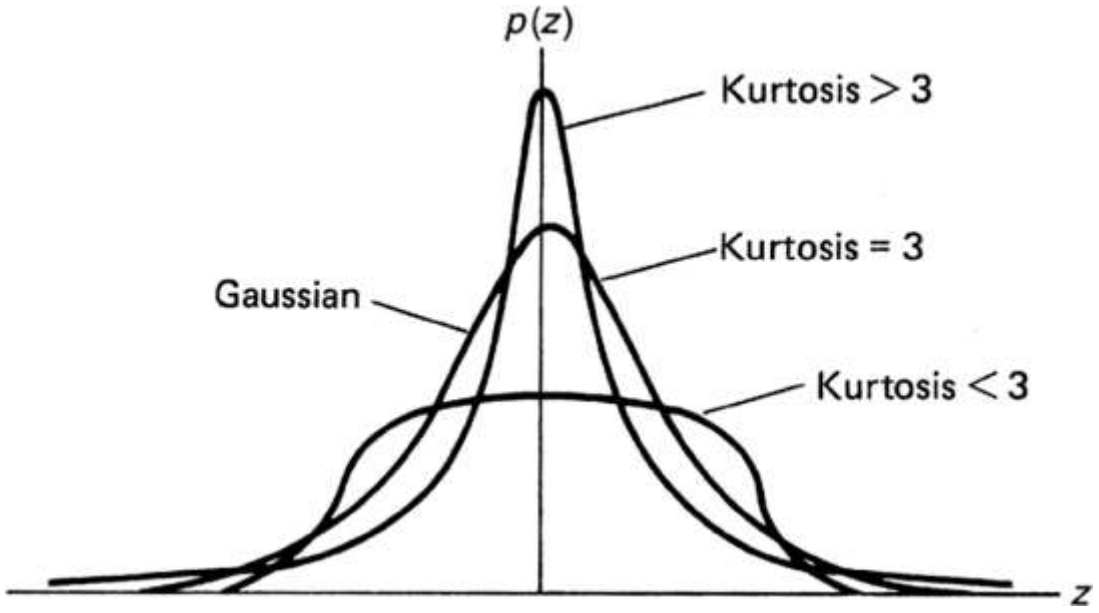
Skewness

- Skewness measure lack of symmetry. A dataset is symmetric if it looks the same to the left and right of the center point
- a.k.a., the third moment

$$Skew[X] = E[(X - \mu)^3]$$



Kurtosis



- Kurtosis measures whether the dataset is heavy-tailed (high kurtosis) or light-tailed (low kurtosis) relative to a normal distribution
- Heavy tails signals the presence of outliers
- Light tails the absence of outliers
- a.k.a., the fourth moment

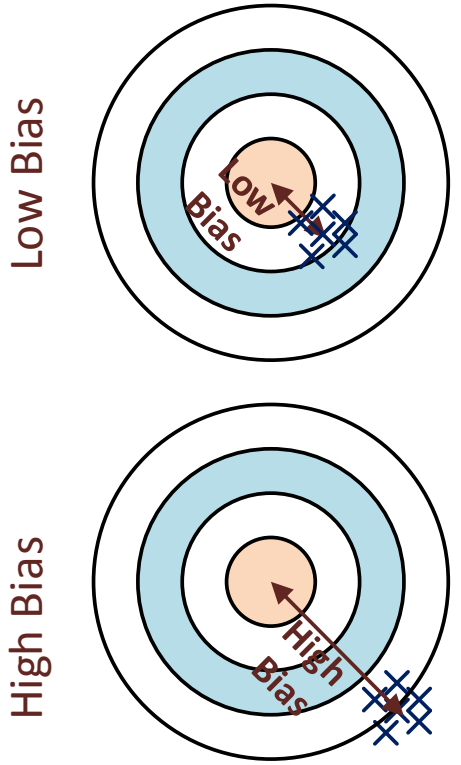
$$Kurt[X] = E[(X - \mu)^4]$$

❸ PARSE the Data

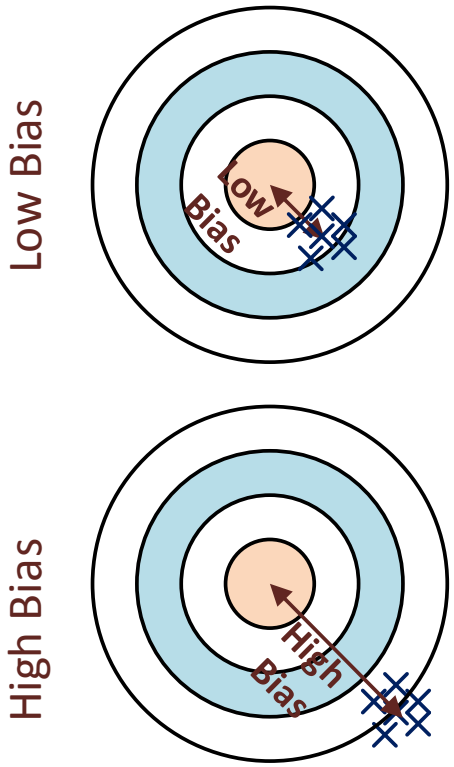
Measurement Errors

Bias

- Source of *systematic* rather than *random* error
- Can lead to false conclusion despite the application of correct statistical procedures and techniques



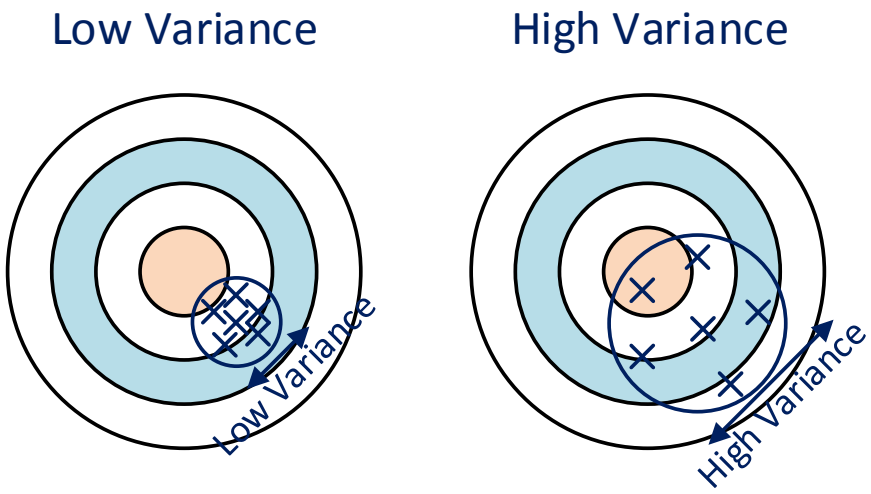
Bias (cont.)



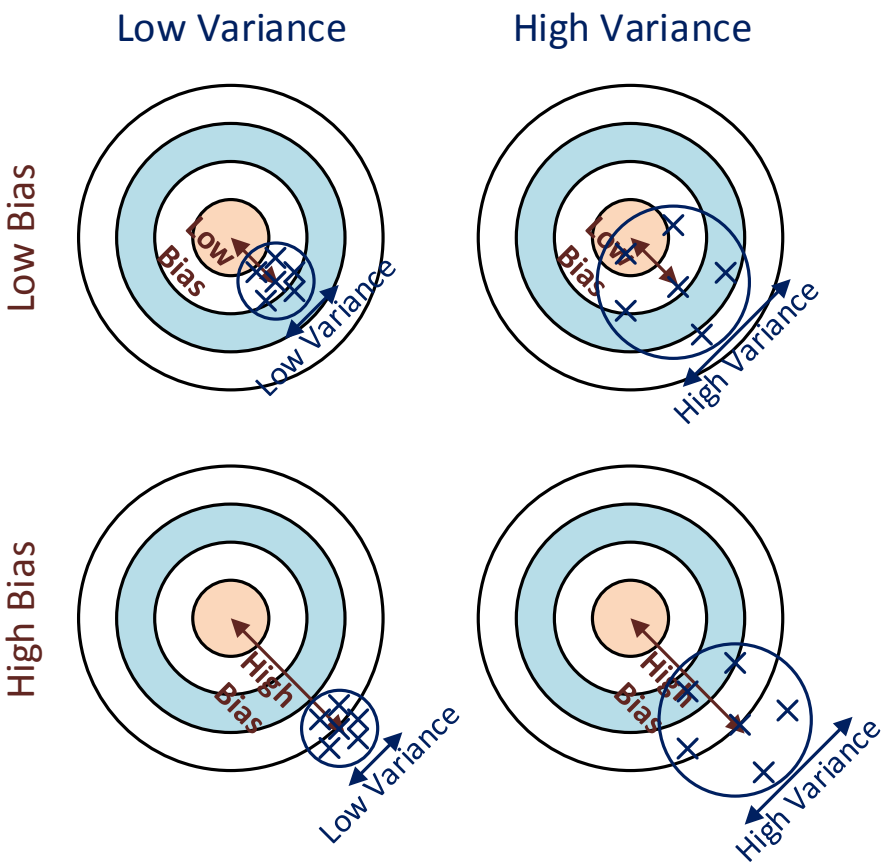
- Selection bias
- Volunteer bias
- Nonresponse bias
- Survival bias

Variance

- Source of *random* rather than *systematic* error



Bias vs. Variance, a.k.a., *Systematic* vs. *Random* errors



❸ PARSE the Data

(Linear) Correlation

Correlation

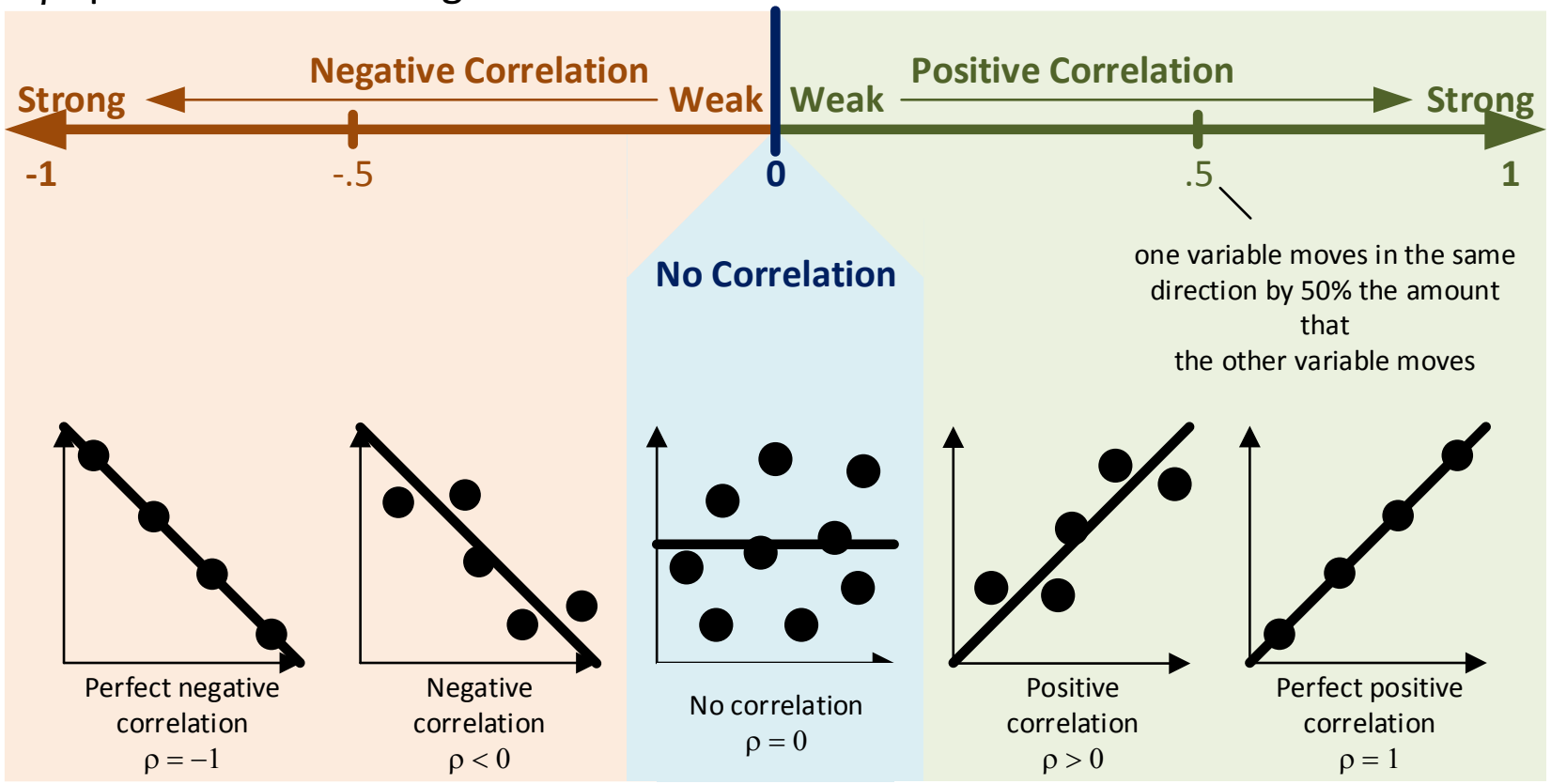
- A measure of strength and direction for a **linear association** between two random variables

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

- $\rho = 0$ means that the two variables don't have a linear association
  - It doesn't imply that they are independent!

Correlation (cont.)

ρ quantifies the strength and direction of movements of two random variables

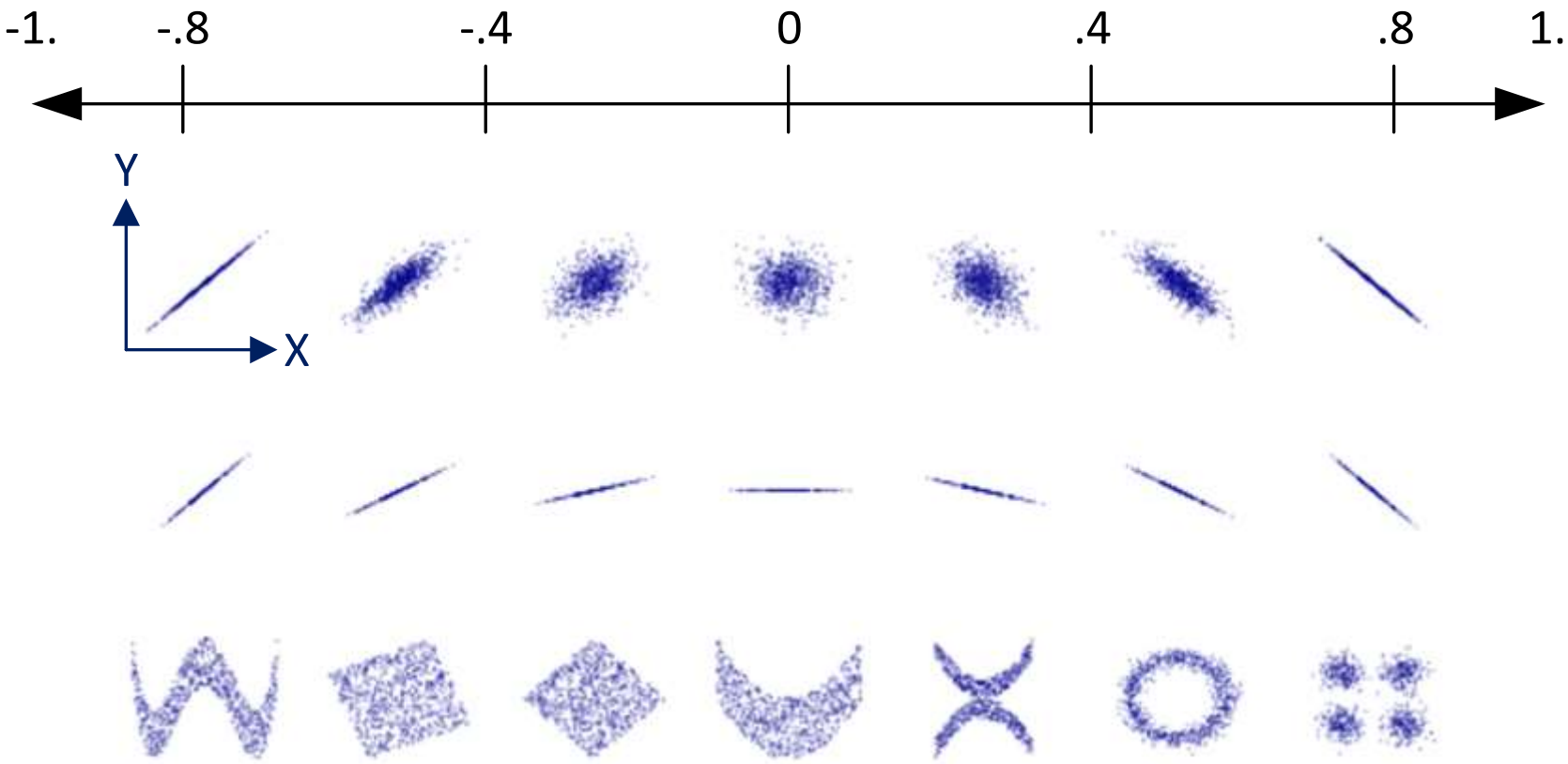


❸ PARSE the Data

Activity | Correlations and Scatter Plots

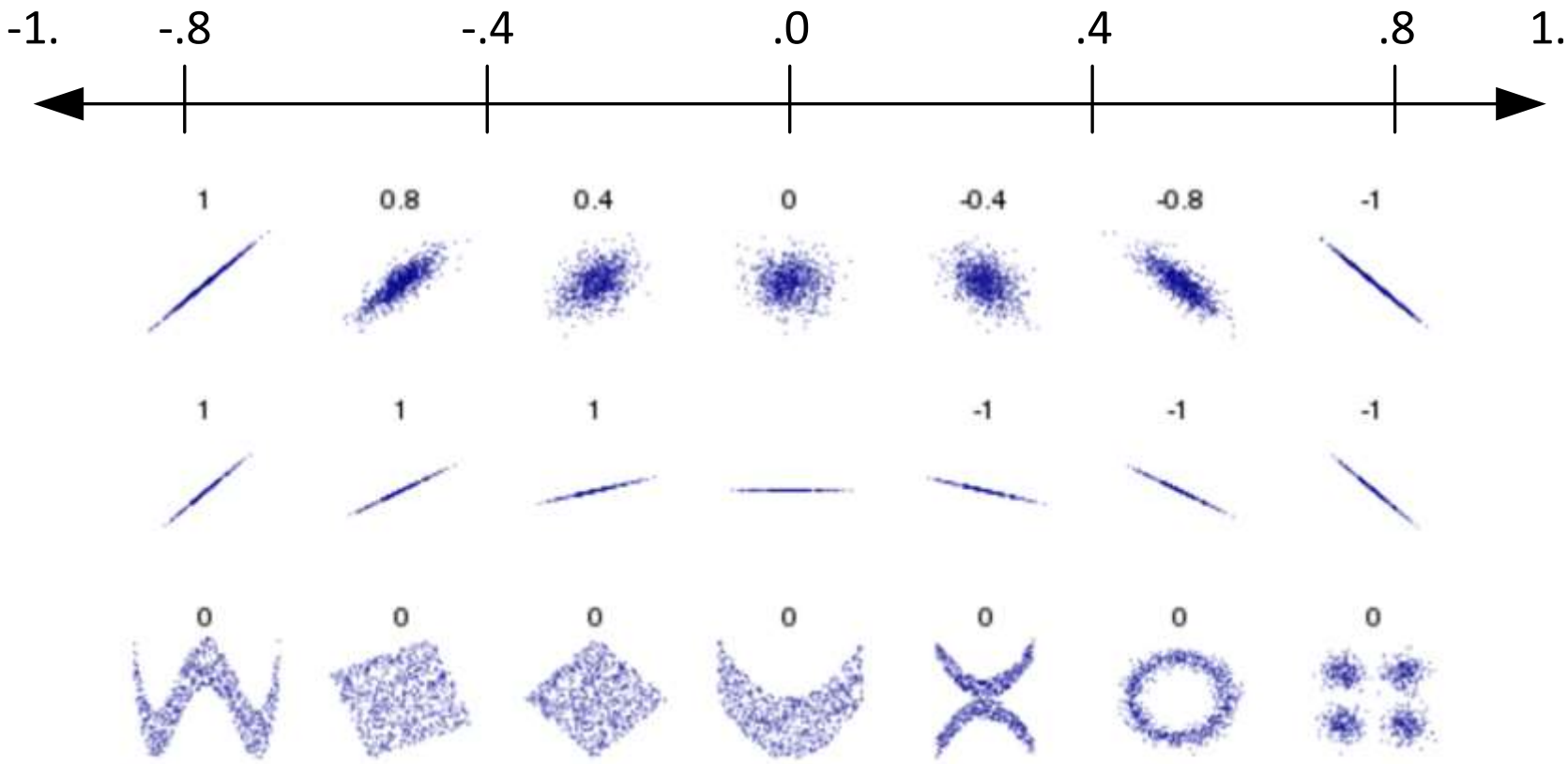
Activity: What's the correlations for the following scatter plots (5 minutes)

EXERCISE



Activity: What's the correlations for the following scatter plots (cont.)

EXERCISE



Source: Wikipedia

❸ PARSE the Data

Codealong – Part E

.corr()

Heatmaps
Scatter plots and matrices

Lab

Exploratory Data Analysis with pandas

Review

Review

You should now be able to:

- ID variable types
- Use the *pandas* (and *NumPy*) libraries to analyze datasets using basic summary statistics: mean, median, mode, max, min, quartile, inter-quartile range, variance, standard deviation, and correlation
- Create data visualizations – including boxplots, histograms, and scatter plots – to discern characteristics and trends in a dataset

Q & A

Before Next Class

Before Next Class

- Projects
  - Unit Project 2 (due a week from now on 5/19)

Next Class

Flexible Class Session #1 | Exploratory Data Analysis

Exit Ticket

Don’t forget to fill out your exit ticket here

Slides © 2016 Ivan Corneillet Where Applicable
Do Not Reproduce Without Permission

Airline Tweets Sentiment Analysis

General Assembly

Data Science

by Michael Lin

DATA SETS

Where, when and how?

FOUR MAJOR AIRLINES

@AmericanAir

AA

#americanairlines

#americanair

@delta

DELTA

#deltaairlines

#deltaair

@southwestair

SOUTHWEST

#southwestairlines

#southwestair

@united

UNITED

#unitedairlines

#unitedair

Sentiment Analysis

SOURCES

80,121 Tweets

TWITTER API

Using Python and **twython** to retrieve tweets through Twitter's API during 7 days period.

2,400 rated sentiment

14,640 Tweets

KAGGLE

Reformatted/cleaned tweets with graded sentiment of Major Airlines from Feb 2015

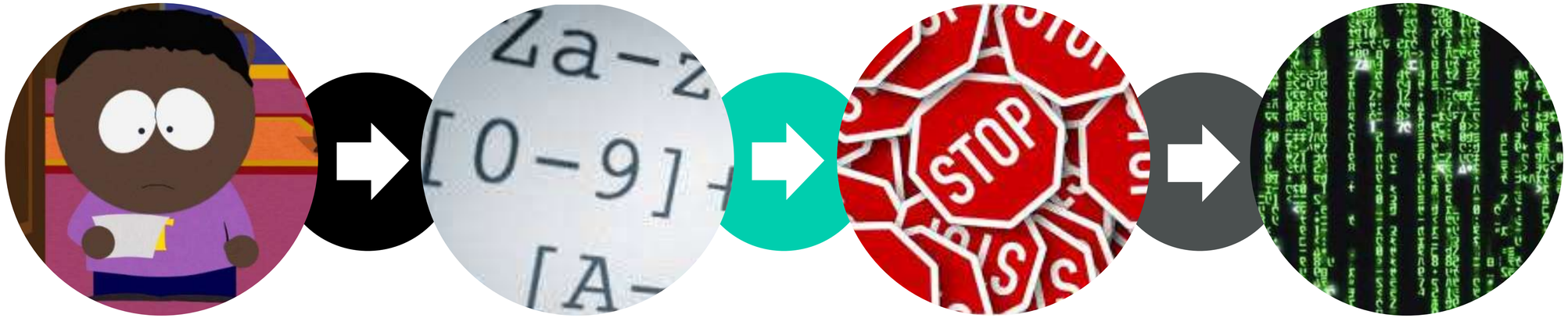
All with rated sentiment

DATA PRE-PROCESSING

Natural language processes to prepare and transform the tweet content for various classification models and sentiment analysis

NATURAL LANGUGE PROCESSING

Major processing steps



1 Tokenization

Tokenize all tweet contents

2 RegEx

Perform regular expression

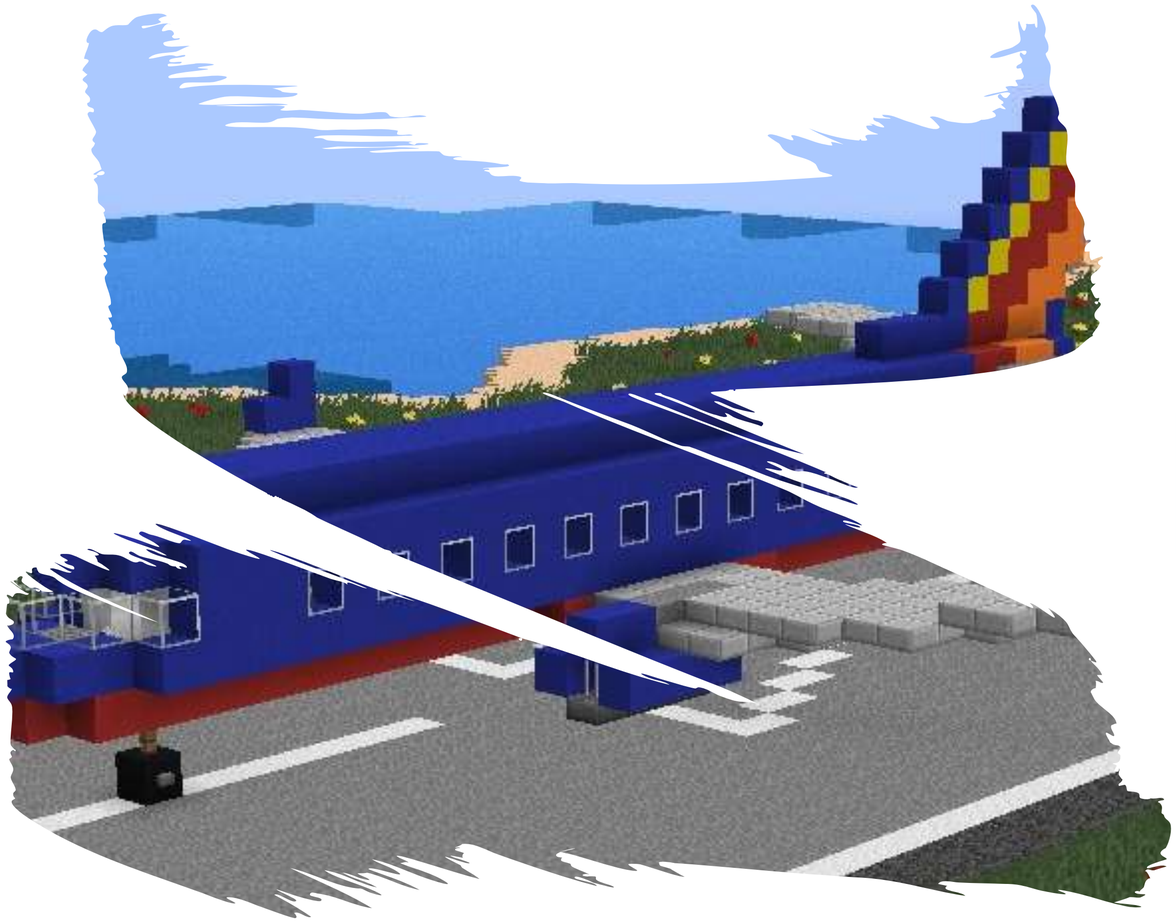
3 Stop Words

Remove all the English stop words

4 Vectorizer

For machine learning

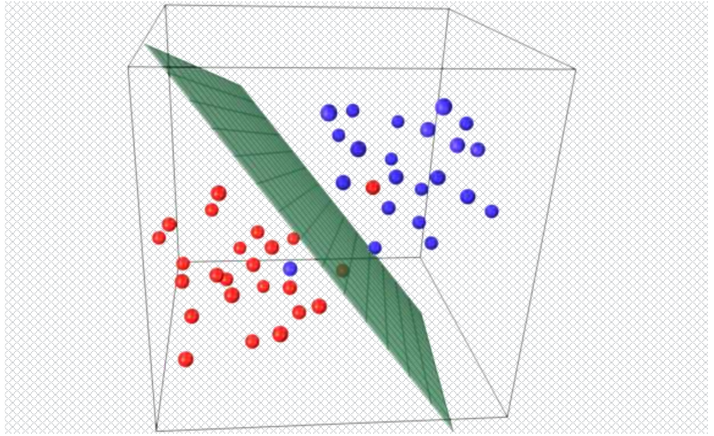
Vector → Matrix… get it?



MACHINE LEARNING

Part I – **Binary** Classifications

MACHINE LEARNING I



1 Data Prep

Remove All Neutral Sentiment

2 Random Forest

Perform Random Forest Model

3 Linear SVM

Perform Support Vector Classifier

“Hyperplane”

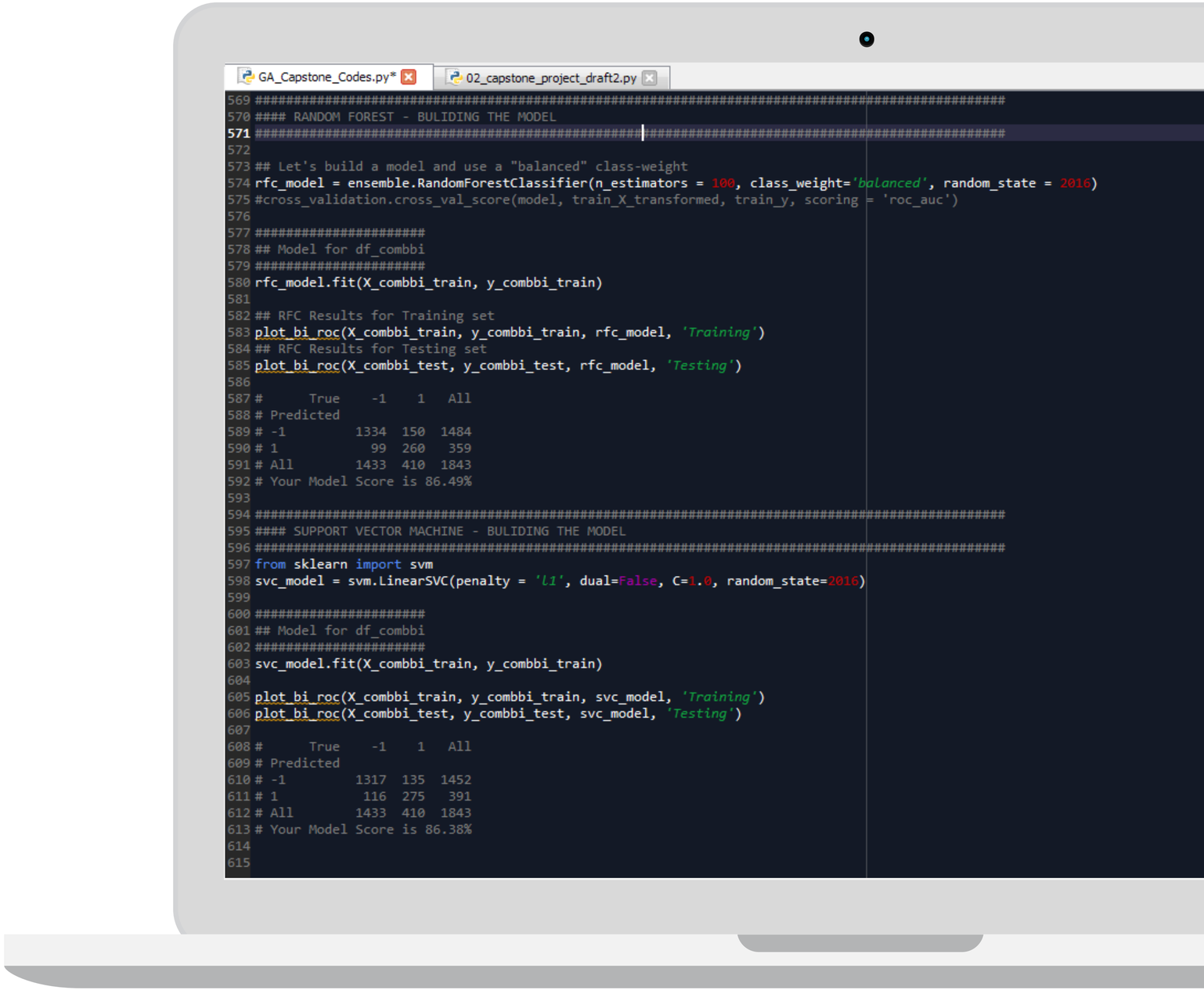
Accuracy

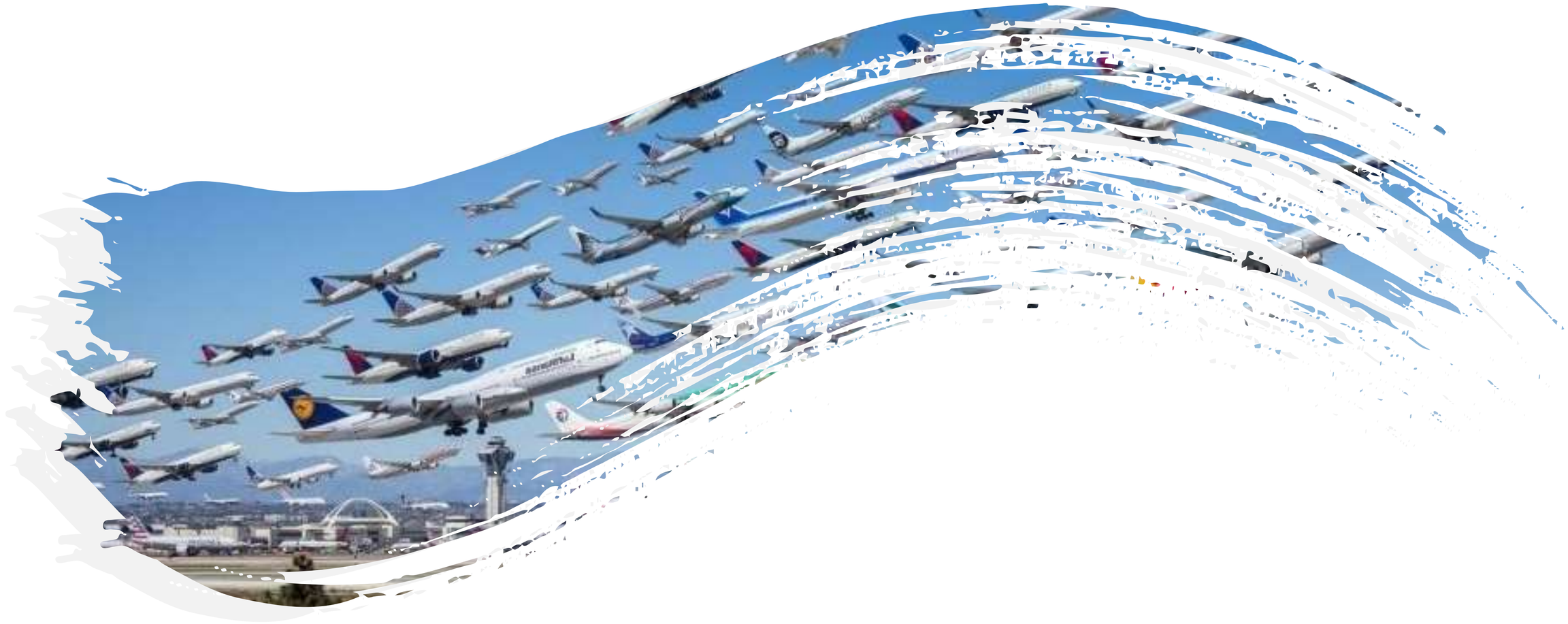
RANDOM FOREST

86.49 %

SUPPORT VECTOR CLASSIFIER

86.38 %





MACHINE LEARING

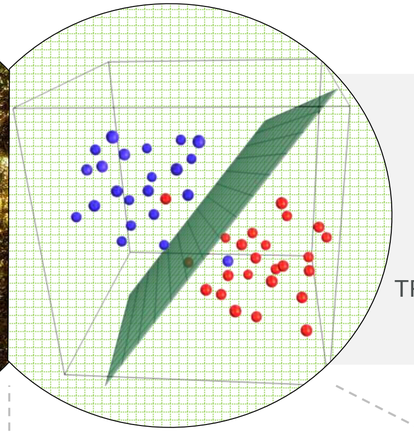
Part II – **Multi-Class** Classifications

MACHINE LEARNING II

TRAINING: 2,040

TESTING: 360

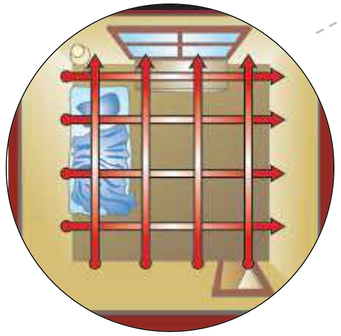
AdaBoost



SVC

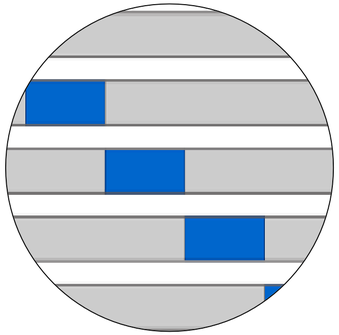
TRAINING: 14,640

TESTING: 2,400



Grid Search Cross-Validation

GridSearchCV ( )



5-Fold Cross Validation

StratifiedKFold ( )

Apply Model on Testing

model.fit ( )

CONCLUSION

Support Vector Classifier

Model Parameters: decision_fun='ovo', kernel='linear', cost=0.4, gamma=0

Use Entire Kaggle Dataset as Training Set (14,640 data points)

Apply Model to Tweets of Unknown Sentiment

IPython console

Console 1/A

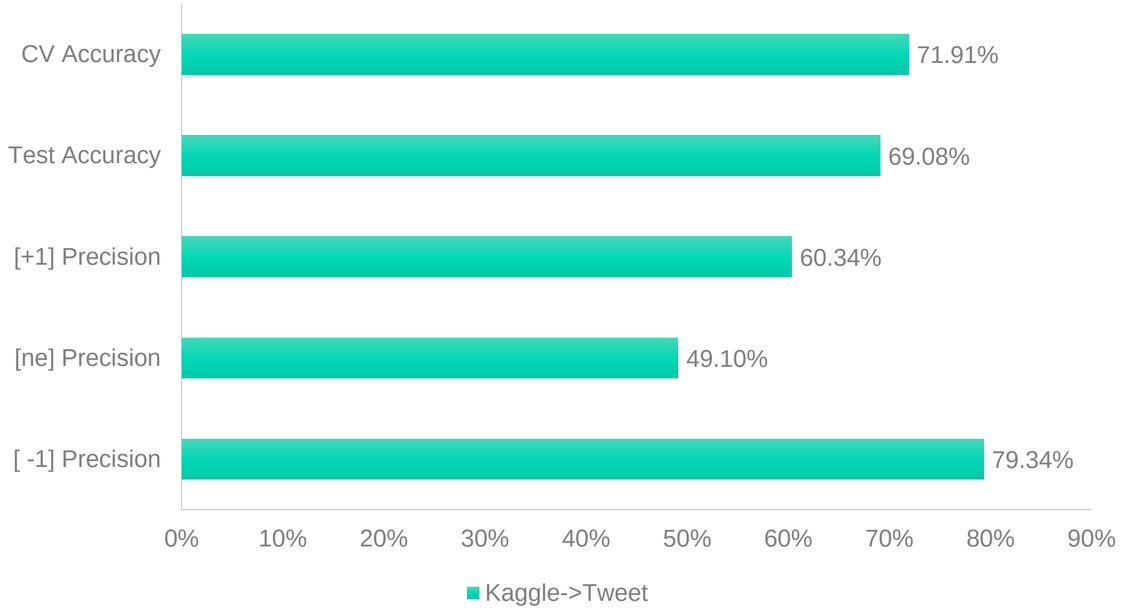
```
In [47]: print('The Average Score is {:2.4}%'.format(np.average(accu_cv2)*100))
    ...: multi_class_outcome(X_2400_test, y_2400_test, svm_model2)
    ...:
The Average Score is 66.39%
     True   -1   0   1  All
Predicted
-1        210  57  32  299
0           6  10   2   18
1           9   3  31   43
All       225  70  65  360
69.7222222222
Your Model Score is 69.72%

In [48]: print('The Average Score is {:2.4}%'.format(np.average(accu_cv)*100))
    ...: multi_class_outcome(X_comb_test, y_comb_test, svm_model)
    ...:
The Average Score is 71.91%
     True    -1    0    1   All
Predicted
-1        1137  253   97  1487
0          217  273   66   556
1           79   30  248   357
All       1433  556  411  2400
69.0833333333
Your Model Score is 69.08%

In [49]: ## Here we Correlation between the truth and prediction
    ...: TrueLabel = list(itertools.chain(*true_cv))
    ...: PredictedLabel = list(itertools.chain(*pred_cv))
    ...: print ('Correlation between the actual and prediction is:', pearsonr(TrueLabel
    ...:        'with p-value',  ("%2.2f" % pearsonr(TrueLabel, PredictedLabel)[1]))
    ...:
Correlation between the actual and prediction is: 0.576245965274 with p-value 0.00

In [50]: ## Here we plot out the confusion matrix of the Cross-Validation Results
    ...: cm = confusion_matrix(PredictedLabel, TrueLabel)
    ...: fig, ax = plt.subplots()
    ...: im = ax.matshow(cm)
    ...: for (i, j), z in np.ndenumerate(cm):
    ...:     ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center',
    ...:             bbox=dict(boxstyle='round', facecolor='white', edgecolor='0.3'))
    ...: plt.title('Confusion matrix')
    ...: fig.colorbar(im)
```

SVC
MODEL PERFORMANCE



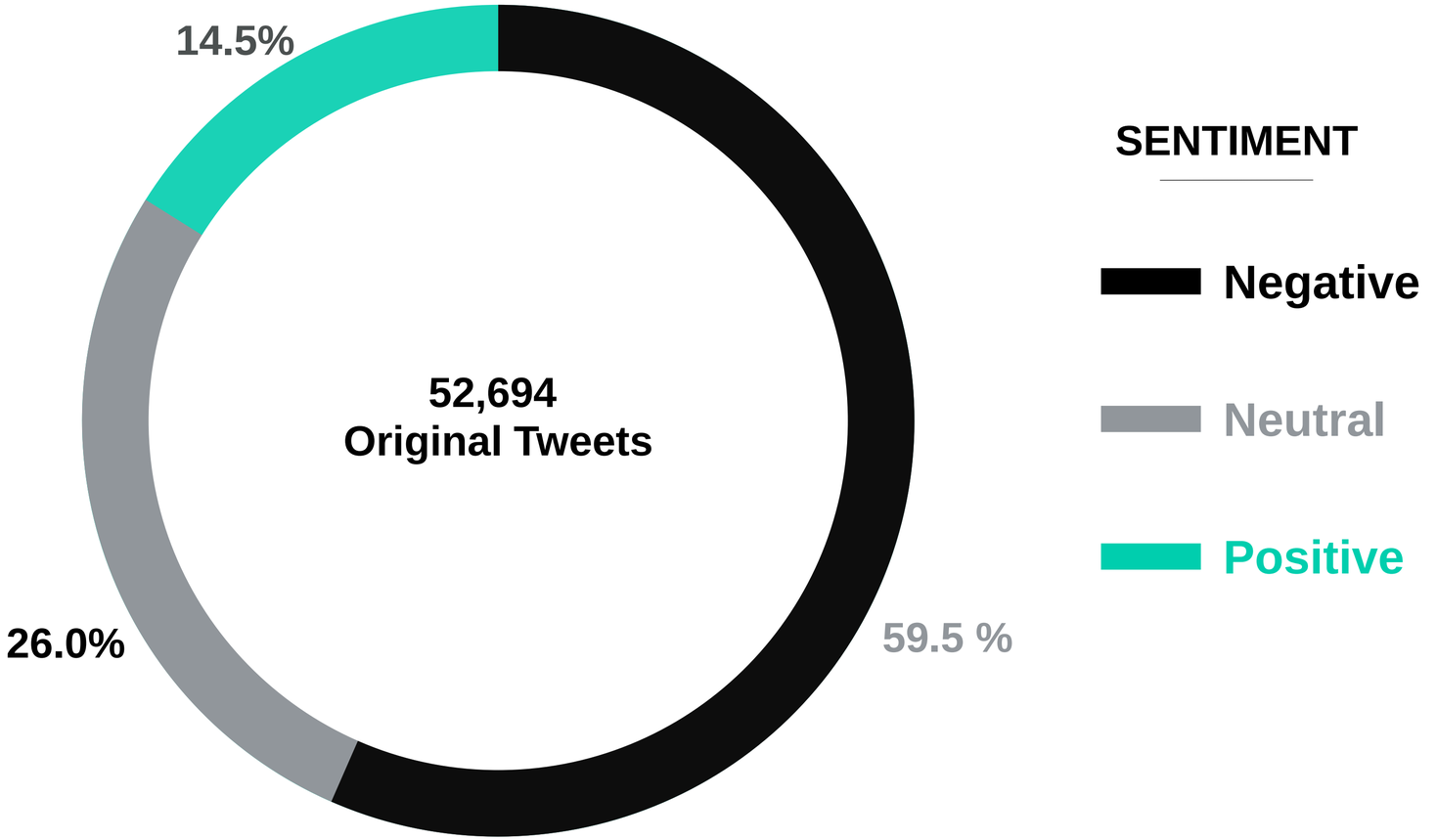
Again, while we see the precision of the twitter data is superior, the Kaggle data yields a more balanced results for all three sentiments. Because the overall test accuracy is about the same, we chose to use Kaggle data as training due to its superior CV accuracy as well as its large sum of data points (~8X).

STATISTICS

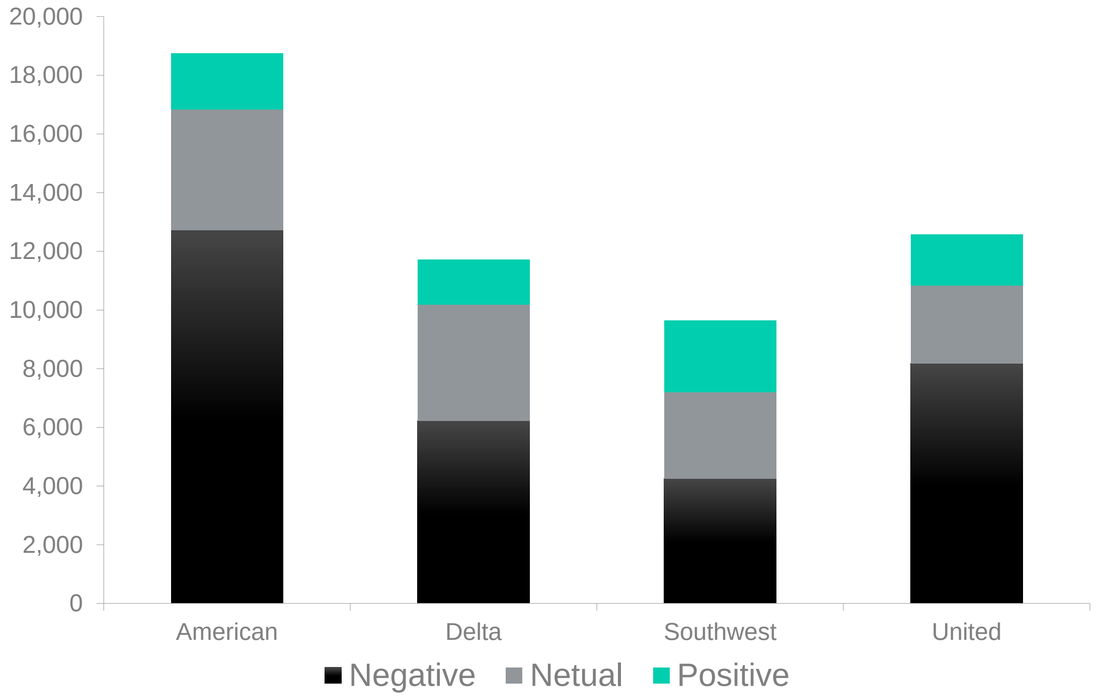
Now we have the model, let’s take a look at the results of collected tweets!

Sentiment Analysis

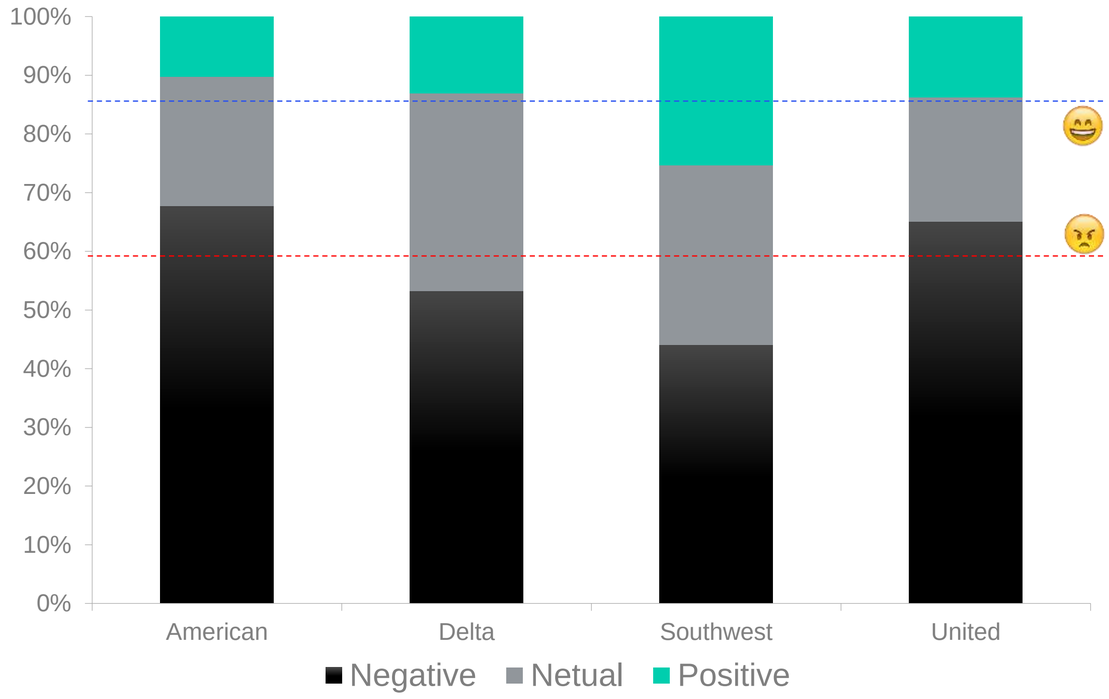
OVERALL SENTIMENT



SENTIMENT PER AIRLINE



Number of Original Tweets



Percentage of Original Tweets

SOUTHWEST AIRLINE

Negative Sentiment

“Dalton Rapattoni”

Positive Sentiment

“Dalton Rapattoni”

UNITED AIRLINE

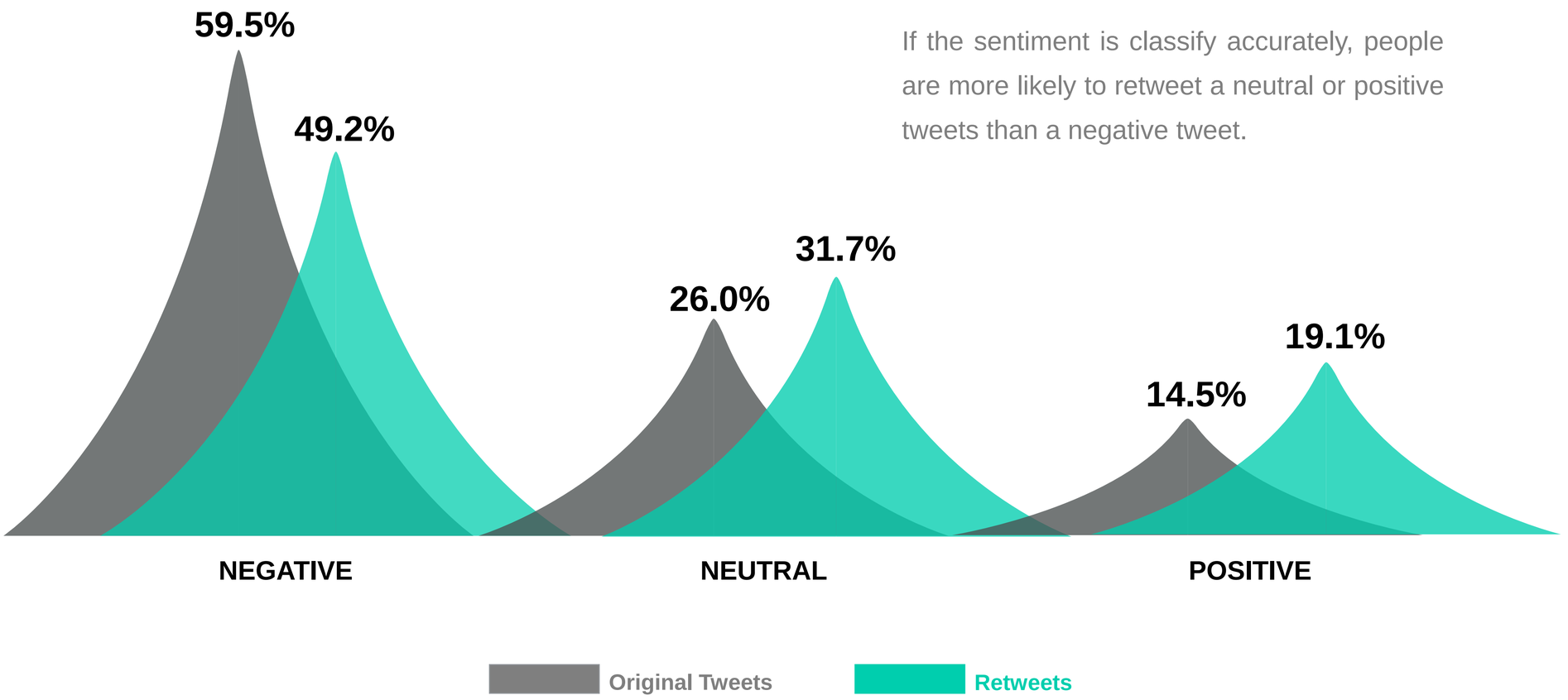
Negative Sentiment

“Muslim”, “Shame”

Positive Sentiment

“Happy Birthday”, “90th"

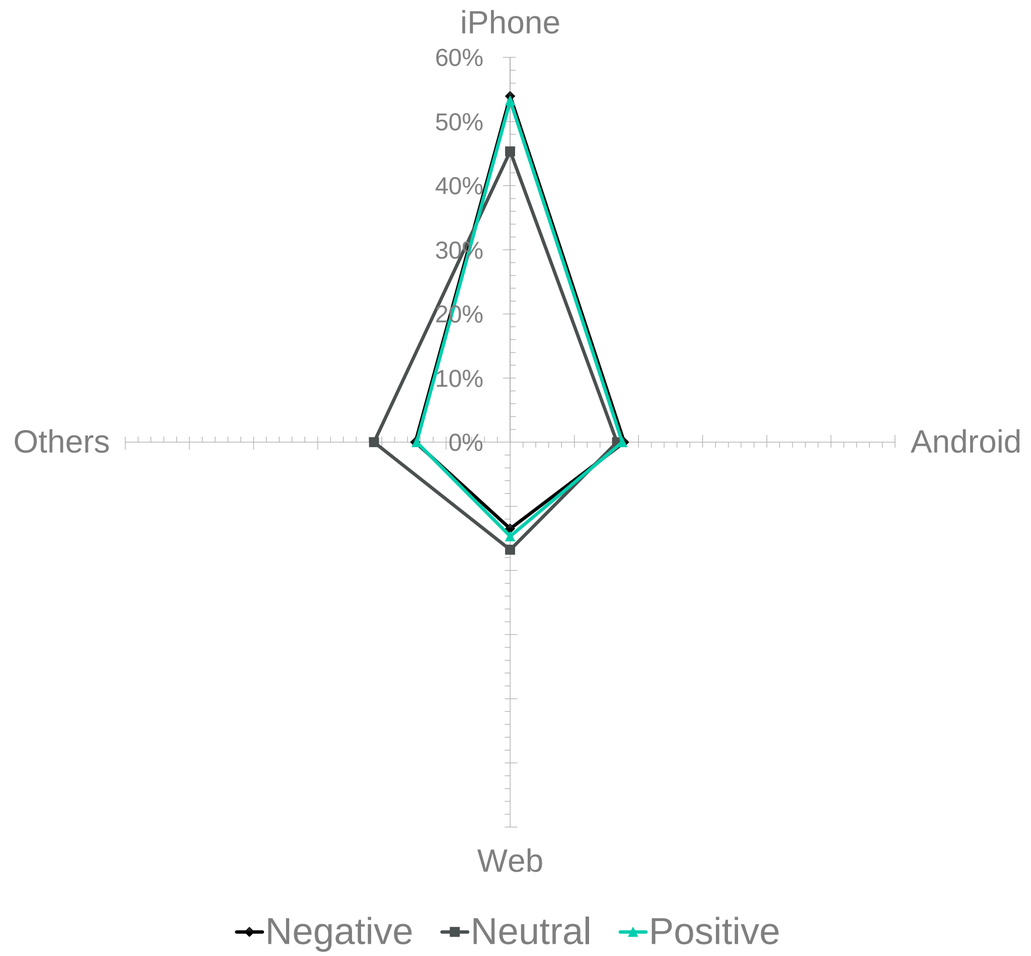
RETWEET PER SENTIMENT



If the sentiment is classify accurately, people are more likely to retweet a neutral or positive tweets than a negative tweet.

TWEET SOURCES

The majority of the tweets are from iPhone. There is also almost 20% of tweets from Android devices. The rest from the Web client and other applications. There is little difference between the source of negative and positive tweets; but there seems to be more alternative sources for the neutral sentiment.



THANK YOU

Q&A

https://github.com/**michaelucsb**/Python_Twitter-Sentiment_GA

SUGGESTIONS

Setting up for final project success!

SET UP FOR PROJECT SUCCESS!

1. Find a topic that interests you

Don’t forget, 1/3 of the data science skill relies on domain knowledge. This will help keeping you engage and spend more effort on writing codes and data understanding.

2. Obtain and review dataset contents asap

I mean like… now! Many of my classmates ended up having to switch project last minutes because they didn’t realize their data was not useable.

3. Don’t try to solve the world’s hunger problem

I just mean don’t go overboard – try to put some hard constrains on what you want to do and how much data you want to utilize.

4. Utilizes functions

If you have to use the same code more than twice, write a function. It also reduces the chances of mistakes and save you time.

5. Explore visualizations for your data

Very useful for your final presentations and keep your audience engaged.

6. Be patient – like visiting Disney Land in summertime

Be prepare to go back and forth between building different models and multiple iterations of data cleaning.