Intro to Data Mining Lecture 1b

Data Preparation and Cleaning

Created by Jon Witkowski on 12/28/2023

What is data?

- Collection of objects and their attributes
  - An attribute is a characteristic of an object and can also be known as a variable, field, dimension, feature.
  - Each data object is a row and each attribute will be a column

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Types of Attributes

Broad Category	Type	Description	Examples
Categorical Quantitative	Nominal	Equal or Not Equal	Zip Codes, Patient ID, Eye Color
	Ordinal	Order Objects	Grades, Olympic Medals
Numeric Qualitative	Interval	Measured in fixed and equal units. 0 does not mean nothing	Calendar dates, Celsius or Fahrenheit temperature
	Ratio	0 is the lack of a value	Kelvins, age, length, money, mass

Classwork

- Let’s characterize each attribute as nominal, ordinal, interval, or ratio

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25

More Classifications of Attributes

- Discrete and Continuous
  - Discrete data has a finite or countably infinite set of values
    - Number of stairs in a building,
  - Continuous has an uncountably infinite set of values
    - Height, length, weight
- Symmetric and Asymmetric
  - Data is symmetric when outcomes are equally important
    - Gender, etc.
  - Data is asymmetric when outcomes are not equally important
    - Medical test positive/negative

More Classwork

- I realized after the fact that most of those columns from the last example are ratio and we didn't have any nominal or ordinal. Here's another table:

Name (Identifier)	Gender	Favorite Color	Blood Type	General Health	Test1	Cough	High Blood Pressure
Susan	F	Blue	O-	excellent	75	N	N
Jim	M	Red	O+	good	65	N	N
Joe	M	Red	AB-	fair	64	N	Y
Jane	F	Green	A+	poor	83	Y	Y
Sam	M	Blue	A-	good	71	N	N
Michelle	F	Blue	O-	good	90	N	N

Types of Data

- In this class, we will go over 3 types of data:
  - Record Data
    - Consists of a collection of records and a fixed set a attributes
    - Will usually be tabular
  - Document Data
    - Each row is a document and each document is a vector of terms
    - Commonly used in text mining
  - Transaction Data
    - Each record has a set of items
    - Will be used in market basket analysis

Types of Data

Record Data

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Document Data

	Document 1	Document 2	Document 3
season	2	0	0
timeout	0	0	3
lost	2	3	0
win	0	0	2
game	6	0	2
score	2	1	1
ball	0	2	0
play	5	0	0
coach	0	7	1
team	3	0	0

Transaction Data

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Why do we need to preprocess?

- Garbage in, garbage out
- A good model needs good data
- Missing values, duplicates, inconsistencies, unnecessary fields, outliers, and more can ruin a good model
- Some fields are more important than others and must be treated as such
- The scaling in different fields could poorly reflect the importance and could result in a skewed or biased model
- All of these things must be considered when trying to make our data useful and usable
- In all, this task of cleaning and preparing data is estimated to take around 60% of the time that you will spend

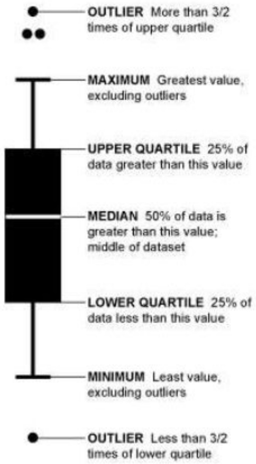
What do we do?

- We must:
  - Find outliers
  - Find missing values
  - Find duplicate rows (if any)
  - Normalize variables (if model calls for it)
  - Adjust values accordingly

Finding Outliers

- One good way to approach this graphically is with a box-and-whisker plot.
- Outliers are determined by using the Tukey Test
- This uses the difference between the upper and lower quartiles (interquartile range)
- Upper Threshold: 3^{rd} Quartile + IQR * factor
- Lower Threshold: 1^{st} Quartile - IQR * factor
- Where the factor is generally 1.5, but can be variable

Reading a Box-and-Whisker Plot



Let's say we ask 2,852 people (and they miraculously all respond) how many hamburgers they've consumed in the past week. We'll sort those responses from least to greatest and then graph them with our box-and-whisker.

Take the top 50% of the group (1,426) who ate more hamburgers; they are represented by everything above the median (the white line). Those in the top 25% of hamburger eating (713) are shown by the top "whisker" and dots. Dots represent those who ate a lot more than normal or a lot less than normal (outliers). If more than one outlier ate the same number of hamburgers, dots are placed side by side.

Why does this work?

- Statistics generally use 2 different measures to determine center of data and 2 different measures to determine spread of data.
  - These are Mean and Median and Standard Deviation and IQR respectively
  - Mean is impacted by outliers, as they can skew the data, and Standard Deviation utilizes mean in its calculation, so it too is skewed by outliers
  - Median and IQR, on the other hand, don’t necessarily use the values themselves, but the placement and occurrence of the values, so they aren’t skewed by outliers.
  - The Tukey Test uses the IQR, which is independent of Mean and Standard Deviation and will not be impacted by outliers

Normalization

- 2 types of normalization we’ll be using in class:
  - Min-Max
  - Z-Score

- Suppose we have a variable with range [100-500] and another one with range [1-10]. We normalize these values so that variable 1 doesn’t dominate the model

Normalization

Min-Max

Values range [0,1]

N = (x-min)/(max-min)

Z-Score

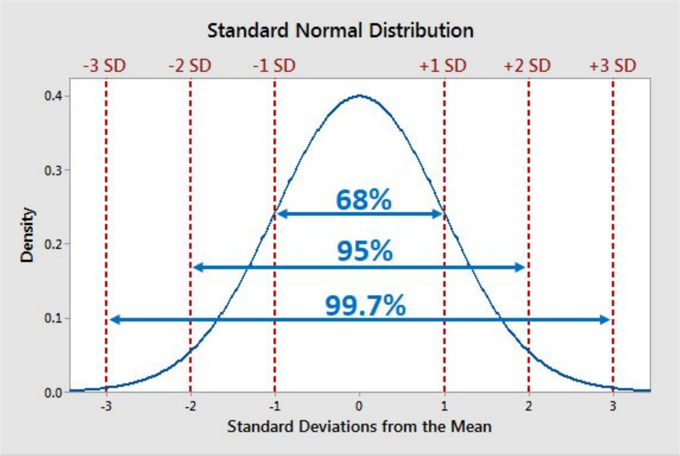
Mean of 0, SDev of 1

N = (x-mean)/SDev

Bounds should roughly be around [-4,4] due to empirical rule

Empirical Rule

- Last slide mentioned the Empirical Rule briefly
- This is also known as the 68–95–99.7 rule because (when the data is normally distributed), 68% of the data will lie within 1 standard deviation of the mean, 95% within 2, and 99.7% within 3.
- Wikipedia has a cool table showing how much data lies within x number of sd:
- https://en.wikipedia.org/wiki/68%E2%80%9395%E2%80%9399.7_rule



Classwork

- Normalize the following sets of numbers with Min-Max and Z-Score Normalization
    - **12, 19, 23, 34, 41, 43, 56, 67, 78, 90**; mean = 46.3, sd = 26.0
    - **5, 11, 18, 29, 32, 42, 47, 54, 63, 76**; mean = 37.7, sd = 23.0
    - **16, 24, 28, 35, 40, 49, 61, 73, 87, 92**; mean = 50.5, sd = 26.7
    - **14, 21, 27, 39, 45, 50, 59, 68, 72, 84**; mean = 47.9, sd = 23.1
    - **10, 17, 22, 31, 46, 58, 64, 75, 81, 93**; mean = 49.7, sd = 29.0

More Ways to Clean Data

- Discretization
  - Taking continuous numeric data and grouping it into discrete bins.
  - Basically the concept behind the x axis on a histogram. Some models or algorithms only work on completely categorical data
- Binarization
  - Taking categorical attributes and splitting them into X number of binary columns where X is the number of categories.

Discretization

- This is useful for turning numeric data into categorical data. We will use this for several types of models later in the semester, including decision trees. Here we see a table with 3 numeric fields

Customer	Savings	Home Value	Income	Credit Risk
1	7	576	75	Good
2	2	99	39	Bad
3	11	289	23	Bad
4	8	363	36	Good
5	1	432	67	Good
6	12	655	30	Good
7	4	176	32	Bad
8	6	205	86+	Good

Discretization

- Here we see the 3 numeric fields from the previous table have been discretized

Customer	Savings	Home Value	Income	Credit Risk
1	5-10	500+	67+	Good
2	Under 5	0-199	34-66	Bad
3	Over 10	200-499	0-33	Bad
4	5-10	200-499	34-66	Good
5	Under 5	200-499	67+	Good
6	Over 10	500+	0-33	Good
7	Under 5	0-199	0-33	Bad
8	5-10	200-499	67+	Good

Binarization

- Binarization is used to map categorical variables into several binary variables that we will call dummies
- Doing this concerts nominal attributes into numerous asymmetric binary attributes
- This is often used in market basket analysis

1	corn	peppers	tomatoes	beans	broccoli
2	broccoli	peppers	corn		
3	asparagus	squash	corn		
4	corn	tomatoes	beans	squash	
5	peppers	corn	tomatoes	beans	
6	beans	asparagus	broccoli		
7	squash	asparagus	beans	tomatoes	
8	tomatoes	corn			
9	broccoli	tomatoes	peppers		
10	squash	asparagus	beans		
11	beans	corn			
12	peppers	broccoli	beans	squash	
13	asparagus	beans	squash		
14	squash	corn	asparagus	beans	

- Here we have some transaction data that we'll see again in Week 10 when we go over Market Basket Analysis
- Each row represents a single transaction containing vegetables that the customers bought

Binarization

- Here we see the same transactions broken up with each category (type of vegetable) being its own column now
- The previous table was transaction data. What kind of data is it now?

	asparagus	beans	broccoli	corn	peppers	squash	tomatoes
1	0	1	1	1	1	0	1
2	0	0	1	1	1	0	0
3	1	0	0	1	0	1	0
4	0	1	0	1	0	1	1
5	0	1	0	1	1	0	1
6	1	1	1	0	0	0	0
7	1	1	0	0	0	1	1
8	0	0	0	1	0	0	1
9	0	0	1	0	1	0	1
10	1	1	0	0	0	1	0
11	0	1	0	1	0	0	0
12	0	1	1	0	1	1	0
13	1	1	0	0	0	1	0
14	1	1	0	1	0	1	0
Sum	6	10	5	8	5	7	6

Data Reduction

- Aggregation
  - Combining multiple attributes/objects into a single attribute/object. This reduces the data and uses less processing time/memory. Less variable, though can result in losing some data
- Sampling
  - Grabbing a subset of the data that is representative of the whole because processing the entire dataset can be too time consuming or taxing on the computer
- Principal Component Analysis
  - Creating new linear representations of the data that better represent the direction of the data. Only a certain number of principal components will be selected, reducing the number of columns

Aggregation

- Combine multiple attributes or objects into one.
    - This can include calculations like BMI (derived from height and weight) or maybe transforming monthly data into yearly data to reduce the rows by a factor of 12.

- Combining attributes or rows reduces the data for easier processing, can change the granularity of the data, and can make it less variable at the expense of possible insights
- See the dip and elevation a little before 5 on the x-axis on the left graph. By changing granularity from month to year, that insight is lost on the right graph.

Variation of Precipitation in Australia

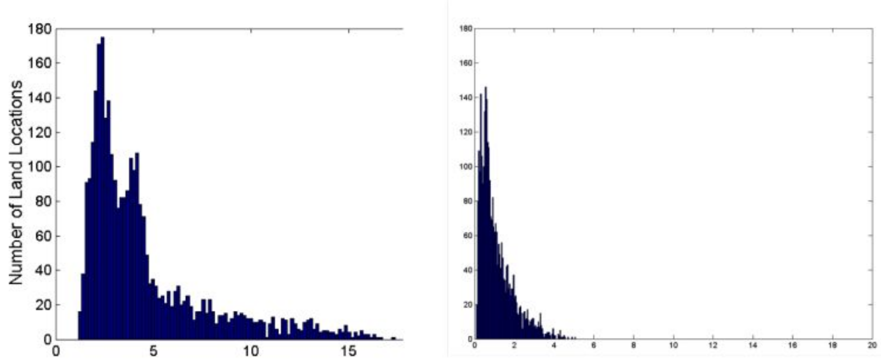


Figure: (Left) Standard deviation of average monthly precipitation; (Right) Standard deviation of average yearly precipitation

Sampling

- A sample is representative of the original set if the properties are roughly the same. Most of the time, the models we use in class will involve splitting the data into 2 subsets: training and testing.
- Ideally the training sample will be representative of the whole data.
- There are different types of sampling:
  - Simple Random Sampling
    - Sampling without replacement
    - Sampling with replacement
  - Stratified Sampling
    - Splitting the data into partitions and then drawing random samples from each partition

PCA

- Sometimes you just have too much data and your model is taking too long to run. Here, I'll briefly go into Principal Component Analysis (PCA).
- The goal of PCA is to find a projection, or set of Principal Components that captures a large amount of the variation in the data.
- Principal Components are mappings of the data generated by z-score normalizing the data and then taking the eigenvectors of the covariance matrix.
- You will then multiply your eigenvectors (in a matrix sorted by eigenvalue) by your original normalized data.
- That's a lot. Let's clarify each step.

PCA

- First step is to take the data and z-score normalize it. Remember the formula is (x-mean)/stdev
- Then we want to take the covariance matrix of this. Luckily, Python has libraries that do this so we don't have to do the math. The formula for this is: where X_i and Y_i are the i^{th} values of each column, X^ and Y^ are the means of X and Y, and n is the number of rows

$$\frac{\sum_{i=1}^{n} (X_i - \hat{X})(Y_i - \hat{Y})}{n-1}$$

- After this step, just have Python take the eigenvectors and eigenvalues. This isn't a linear algebra class.
- Then we just sort by the eigenvalues. The eigenvalues represent the % of variability in the data that the Principal Component represents. Specifically, the formula for % variability is: where lambda_i is the i^{th} eigenvalue after sorting.

$$\frac{\lambda_i}{\sum_{j=1}^{k} \lambda_j}$$

PCA Example

- We'll be using the 1990 California Census Housing Data from http://lib.stat.cmu.edu/datasets/

Median HomeVal	Median Income	Median HomeAge	Total Rooms	Total Bed Rooms	Populat ion	House holds	Lati tude	Longi tude
452600	8.3252	41	880	129	322	126	37.9	-122
358500	8.3014	21	7099	1106	2401	1138	37.9	-122
352100	7.2574	52	1467	190	496	177	37.9	-122
341300	5.6431	52	1274	235	558	219	37.9	-122
342200	3.8462	52	1627	280	565	259	37.9	-122
269700	4.0368	52	919	213	413	193	37.9	-122
299200	3.6591	52	2535	489	1094	514	37.8	-122
241400	3.12	52	3104	687	1157	647	37.8	-122
226700	2.0804	42	2555	665	1206	595	37.8	-122
261100	3.6912	52	3549	707	1551	714	37.8	-122
281500	3.2031	52	2202	434	910	402	37.9	-122
241800	3.2705	52	3503	752	1504	734	37.9	-122
213500	3.075	52	2491	474	1098	468	37.9	-122
191300	2.6736	52	696	191	345	174	37.8	-122
159200	1.9167	52	2643	626	1212	620	37.9	-122

PCA Step 1

- First we're going to normalize the data with z-score normalization

zMed HomeVal	zMed Inc	zMed HomeAge	zTot Rooms	zBed rooms	zPop	zHouse holds	zLat itude	zLong itude
2.13	2.34	0.98	-0.80	-0.97	-0.97	-0.98	1.05	-1.33
1.31	2.33	-0.61	2.05	1.35	0.86	1.67	1.04	-1.32
1.26	1.78	1.86	-0.54	-0.83	-0.82	-0.84	1.04	-1.33
1.17	0.93	1.86	-0.62	-0.72	-0.77	-0.73	1.04	-1.34
1.17	-0.01	1.86	-0.46	-0.61	-0.76	-0.63	1.04	-1.34
0.54	0.09	1.86	-0.79	-0.77	-0.89	-0.80	1.04	-1.34
0.80	-0.11	1.86	-0.05	-0.12	-0.29	0.04	1.03	-1.34
0.30	-0.40	1.86	0.21	0.35	-0.24	0.39	1.03	-1.34
0.17	-0.94	1.06	-0.04	0.30	-0.19	0.25	1.03	-1.34
0.47	-0.09	1.86	0.42	0.40	0.11	0.56	1.03	-1.34
0.65	-0.35	1.86	-0.20	-0.25	-0.46	-0.26	1.04	-1.34
0.30	-0.32	1.86	0.40	0.51	0.07	0.61	1.04	-1.34
0.06	-0.42	1.86	-0.07	-0.15	-0.29	-0.08	1.04	-1.34
-0.13	-0.63	1.86	-0.89	-0.82	-0.95	-0.85	1.03	-1.34
-0.41	-1.03	1.86	0.00	0.21	-0.19	0.32	1.04	-1.34

PCA Step 2

- Now we're going to get the covariance matrix using the formula on Slide 26. Numbers close to 1 imply colinearity and independent variables will show 0 (but 0 does not guarantee independence)

	zMedInc	zMed HomeAge	zBed rooms	zTot Rooms	zPop	zHouse holds	zLati tude	zLongi tude
zMedInc	1.00	-0.12	-0.01	0.20	0.00	0.01	-0.08	-0.02
zMed HomeAge	-0.12	1.00	-0.32	-0.36	-0.30	-0.30	0.01	-0.11
zBedrooms	-0.01	-0.32	1.00	0.93	0.88	0.98	-0.07	0.07
zTotRooms	0.20	-0.36	0.93	1.00	0.86	0.92	-0.04	0.04
zPop	0.00	-0.30	0.88	0.86	1.00	0.91	-0.11	0.10
zHouseholds	0.01	-0.30	0.98	0.92	0.91	1.00	-0.07	0.06
zLatitude	-0.08	0.01	-0.07	-0.04	-0.11	-0.07	1.00	-0.92
zLongitude	-0.02	-0.11	0.07	0.04	0.10	0.06	-0.92	1.00

PCA Steps 3 and 4

- Here, we've taken the eigenvectors of the covariance matrix and multiplied by the normalized matrix. We interpret this as:

  Component 1 = -0.05(Income) + 0.22(Age) - 0.49(Bedrooms) - 0.48(Rooms) - 0.47(Pop) - …
  Component 2 = -0.04(Income) + 0.02(Age) + 0.06(Bedrooms) + 0.07(Rooms) + 0.03(Pop) - …
  Component 3 = 0.89(Income) - 0.39(Age) - 0.12(Bedrooms) + 0.09(Rooms) - 0.12(Pop) - …
  …

	Component							
	1	2	3	4	5	6	7	8
zMedIncome	-0.05	-0.04	0.89	-0.41	-0.06	-0.06	-0.17	-0.04
zMedAge	0.22	0.02	-0.39	-0.89	0.03	0.09	-0.04	0.00
zBedrooms	-0.49	0.06	-0.12	-0.06	0.38	-0.23	-0.22	-0.70
zRooms	-0.48	0.07	0.09	-0.12	0.32	0.56	0.55	0.15
zPop	-0.47	0.03	-0.12	-0.08	-0.85	0.13	-0.02	-0.13
zHouseholds	-0.49	0.06	-0.11	-0.10	0.14	-0.40	-0.30	0.68
zLat	0.07	0.70	0.01	0.10	0.05	0.46	-0.52	0.04
zLong	-0.08	-0.70	-0.06	0.07	0.10	0.48	-0.50	0.05

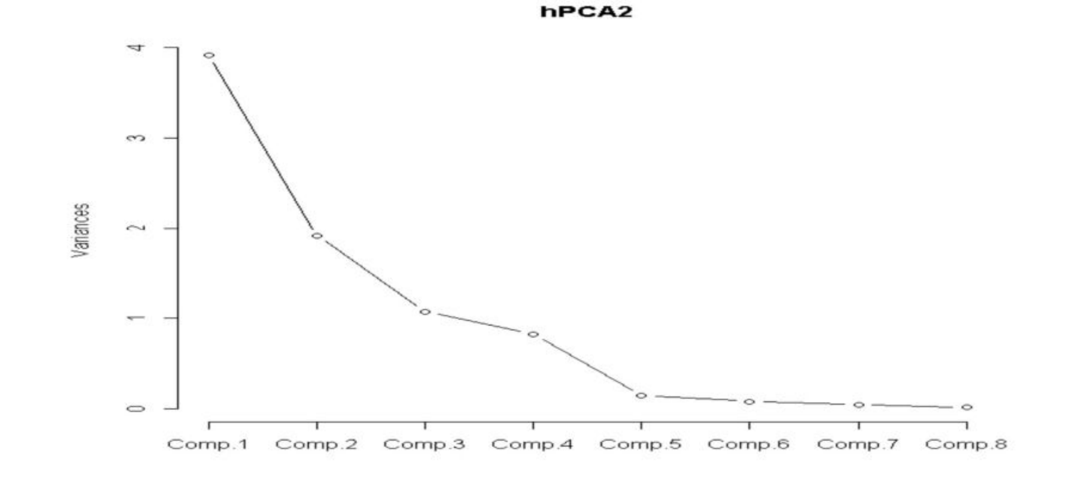
PCA Step 5

- Here we calculate % of variance using the eigenvalues. This will help us pick a number of Principal Components to use for our models.

Component	Eigenvalue	% of Variance	Cumulative %
1	3.91	48.8%	48.8%
2	1.91	23.8%	72.7%
3	1.07	13.4%	86.1%
4	0.82	10.3%	96.4%
5	0.15	1.9%	98.2%
6	0.08	1.0%	99.2%
7	0.05	0.6%	99.8%
8	0.01	0.2%	100.0%

PCA Step 6

- How do we determine how many principal components to take? Here's something called a scree plot
- Typically, you stop when the plot gets flat
- If you're having a hard time deciding where it gets flat, take multiple numbers and try models with both and compare them



PCA Code w/out Sklearn vs. w/ Sklearn

```python
import pandas as pd
import numpy as np

#importing generic data and running z-score calculation
df = pd.DataFrame(data)
z_scores = (df - df.mean()) / df.std()

#here's the covariance matrix
#pandas has a built-in function for it and so does numpy
covariance_matrix = df.cov()

#use numpy.linalg to get eigenstuff
eigenvalues, eigenvectors = np.linalg.eig(covariance_matrix)

#sort the eigenvalues and then order the eigenvectors
sorted_indices = np.argsort(eigenvalues)[::-1]
eigenvalues = eigenvalues[np.argsort(eigenvalues)[::-1]]
eigenvectors = eigenvectors[:, sorted_indices]

#we're gonna take the top 5 components
top_eigenvectors = eigenvectors[:, :5]

#now we do our multiplication and we have our principal components
#hooray
principal_components = np.dot(z_scores, top_eigenvectors)
```

```python
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

#importing our data and using standard scaler to transform the data
df = pd.DataFrame(data)
scaler = StandardScaler()
z_scores = scaler.fit_transform(df)

#scikit learn has pca built-in
pca = PCA()

#look at that. we run pca.fit using our PCA object
pca.fit(z_scores)

#here we have the components. ez
principal_components = pca.transform(z_scores)
```

Review

- What are the different types of data?
- Why do we want to preprocess it?
- What are the different ways to normalize it?
- Why is the Tukey Test effective for finding outliers?
- What methods do we use to clean data?
  - What are the methods for cleaning/transforming?
  - What are the methods for reduction?