LEDE Algorithms

Richard Dunks

Chase Davis

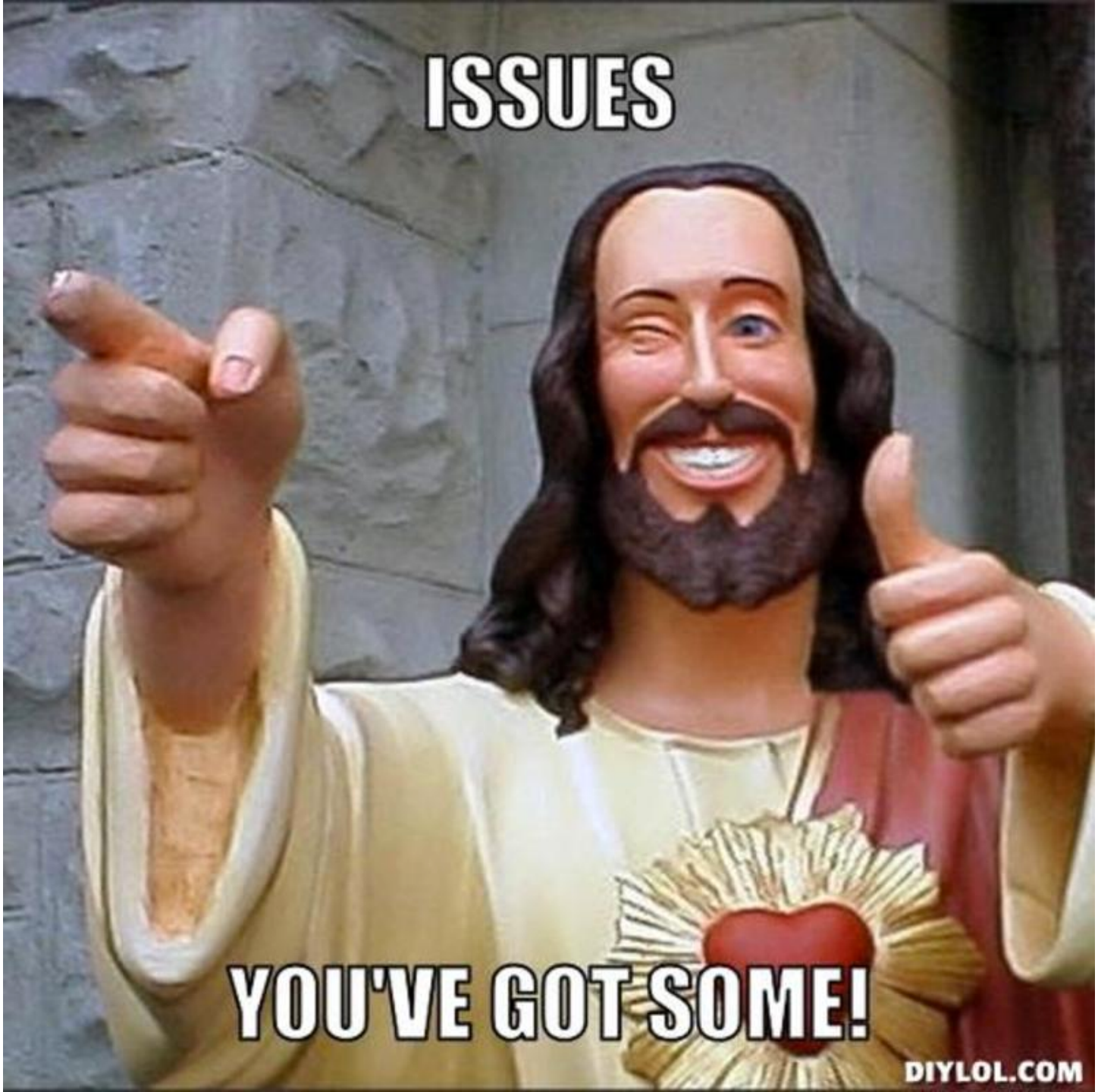
WEEK 1
CLASS 2

Goals for today

- Review exercises from last class
- Review basic descriptive statistics
- Review and practice calculating coefficient of correlation and coefficient of determination
- Review linear regression (if time)
- Practice performing linear regression in Python (if time)

Outcomes for today

- You will be more familiar with the basics of coding in Python
- You will have reviewed the fundamentals of descriptive statistical analysis
- You will be practices in calculating the coefficients of correlation and determination
- You will have practice creating a linear regression model in Python (if time)


ISSUES
YOU'VE GOT SOME!
DIYLOL.COM

EXERCISES FROM LAST CLASS

Exercises

1. Write a function that takes in a list of numbers and outputs the mean of the numbers using the formula for mean. Do this without any built-in functions like sum(), len(), and, of course, mean()

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{\sum x_i}{n}$$

Exercise 1 - Key Challenge

- Implement key features without using built-in functions

Exercise 1 - Possible Answer

```
def my_mean(input_list):
    list_sum = 0
    list_count = 0
    for el in input_list:
        list_sum += el
        list_count += 1
    return list_sum / float(list_count) # cast list_count to float
```

Testing Function

```
tl = []
```

```
my_mean(tl)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-5-a4122f107495> in <module>()
----> 1 my_mean(tl)

<ipython-input-1-c0d0c9ea90a9> in my_mean(input_list)
      5         list_sum += el
      6         list_count += 1
----> 7     return list_sum / float(list_count) # cast list_count to float

ZeroDivisionError: float division by zero
```

Comparing to Numpy

```
import numpy as np
```

```
tl = []
```

```
np.mean(tl)
```

```
/Users/richarddunks/anaconda/lib/python2.7/site-packages/numpy/core/_methods.p
y:59: RuntimeWarning: Mean of empty slice.
  warnings.warn("Mean of empty slice.", RuntimeWarning)
/Users/richarddunks/anaconda/lib/python2.7/site-packages/numpy/core/_methods.p
y:71: RuntimeWarning: invalid value encountered in double_scalars
  ret = ret.dtype.type(ret / rcount)
```

```
nan
```

A Better Possible Answer

```
def my_mean(input_list):
    if input_list:
        list_sum = 0
        list_count = 0
        for el in input_list:
            list_sum += el
            list_count += 1
        return list_sum / float(list_count) # cast list_count to float
    else:
        print "List is empty"
        return
```

Exercises

2. Create your own version of the Mayoral Excuse Machine (http://dnain.fo/1CCHKmI) in Python that takes in a name and location, selects an excuse at random and prints an excuse ("Sorry, Richard, I was late to City Hall to meet you, I had a very rough night and woke up sluggish").

   Use the "excuses.csv" in the Github repository. Extra credit if you print the link to the story as well.

Exercise 2 - Key Challenges

- Read in CSV input
- Get user input
- Randomly select excuse from CSV input and combine with user input

Exercise 2 - Possible Solution

```
import random
```

```
import csv
```

```
person = raw_input('Enter your name: ')
```

```
Enter your name: Richard
```

```
place = raw_input('Enter your destination: ')
```

```
Enter your destination: Chelsea
```

```
r = random.randrange(0,11) # generate random number between 0 and 10
```

```
excuse_list = []
inputReader = csv.DictReader(open('../lede_algorithms/class1_1/exercise/excuse.csv','rU'))
for line in inputReader:
    excuse_list.append(line)
```

```
print "Sorry, " + person + " I was late to " + place + ", " + excuse_list[r]['excuse']
print 'From the story "' + excuse_list[r]['headline'] + '"'
print excuse_list[r]['hyperlink']
```

```
Sorry, Richard I was late to Chelsea, breakfast began a little later than expected
From the story "De Blasio 15 Minutes Late to St. Patrick's Day Mass, Blames Breakfast"
http://www.dnainfo.com/new-york/20150317/midtown/de-blasio-15-minutes-late-st-patricks-day-mass-blames-breakfast
```

Methods of Reading Files

```
excuse_list = []
inputReader = csv.DictReader(open('../lede_algorithms/class1_1/exercise/excuse.csv','rU'))
for line in inputReader:
    excuse_list.append(line)
```

```
inputFile = open('../lede_algorithms/class1_1/exercise/excuse.csv','rU')
header = next(inputFile) # skip the first line of the file
excuse_list = []
for line in inputFile:
    line = line.split(',')
    excuse_list.append(line[0])
inputFile.close() # close connection to the file
```

```
with open('../lede_algorithms/class1_1/exercise/excuse.csv','rU') as inputFile:
    header = next(inputFile) # skip the first line of the file
    excuse_list = []
    for line in inputFile:
        line = line.split(',')
        excuse_list.append(line[0])
    #file connection is close at end of the indented code
```

Exercises

3. Modify the code below (in Exercise3.ipynb) that prints every prime number between 1 and 100 to only print every other prime number. Extra credit if you can modify the code to speed it up.

```
for num in range(1,101):
    prime = True
    for i in range(2,num):
        if (num%i==0):
            prime = False
    if prime:
        print num
```

Exercise 3 - Key Challenge

- Think through the execution of code and modify it slightly to achieve a desired result

Exercise 3 - Possible Answer

```
j = 0 # add check counter outside the for-loop so it doesn't get reset
for num in range(1,101):
    prime = True
    for i in range(2,num):
        if (num%i==0):
            prime = False
    if prime:
        if j%2 == 0: # test the check counter for being even and if so, then print the number
            print num
        j += 1 # increment the check counter each time a prime is found
```

```
1
3
7
13
19
29
37
43
53
61
71
79
89
```

Exercise 3 - Extra Credit

```
j = 0
for num in range(1,1001):
    prime = True
    for i in range(2,num):
        if (num%i==0):
            prime = False
            break
            # once the number has already been shown to be false,
            # there's no reason to keep checking
    if prime:
        if j%2 == 0:
            print num
        j += 1
```

Exercises

4. The code in Exercise4.ipynb is meant to search for New York Times articles on gay marriage and look at the mean and median word count, but the code has some problems. Follow the instructions in the notebook to fix the code and submit your fixed code.

Exercise 4 - Key Challenges

- Interpret error messages and fix errors
- Think through the execution of code to identify logic issues
- Fix logic issues based on an understanding of what the code is intended to do

Exercise 4 - Fixing url assignment

```
api_key = "ffaf60d7d82258e112dd4fb2b5e4e2d6:3:72421680"
```

```
url = "http://api.nytimes.com/svc/search/v2/articlesearch.json?q=gay+marriage&api-key=%s" % API_key
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-8aa99419c0cf> in <module>()
----> 1 url = "http://api.nytimes.com/svc/search/v2/articlesearch.json?q=gay+marriage&api-key=%s" % API_key

NameError: name 'API_key' is not defined
```

```
url = "http://api.nytimes.com/svc/search/v2/articlesearch.json?q=gay+marriage&api-key=%s" % api_key
```

Exercise 4 - Fixing for-loop

```
wc_list = []
for article in r.json()['response']['docs']:
    wc_list.append(article['word_count'])
```

```
my_mean(wc_list)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-10-0d341f097dad> in <module>()
----> 1 my_mean(wc_list)

<ipython-input-9-27d6b5b627b7> in my_mean(input_list)
      3     list_count = 0
      4     for el in input_list:
----> 5         list_sum += el
      6         list_count += 1
      7     return list_sum / list_count

TypeError: unsupported operand type(s) for +=: 'int' and 'unicode'
```

Exercise 4 - Fixing for-loop

```
wc_list = []
for article in r.json()['response']['docs']:
    wc_list.append(article['word_count'])
```

```
my_mean(wc_list)
```

```
wc_list = []
for article in r.json()['response']['docs']:
    wc_list.append(int(article['word_count']))
```

Could also fix in function

Exercise 4 - Fix my_mean()

```
def my_mean(input_list):
    list_sum = 0
    list_count = 0
    for el in input_list:
        list_sum += el
        list_count += 1
    return list_sum / list_count
```

```
def my_mean(input_list):
    list_sum = 0
    list_count = 0
    for el in input_list:
        list_sum += el
        list_count += 1
    return list_sum / float(list_count) # cast list_count to float
```

Exercise 4 - Fixing my_median()

```
def my_median(input_list):
    list_length = len(input_list)
    return input_list[list_length/2]
```

```
def my_median(input_list):
    input_list.sort() # sort the list
    list_length = len(input_list) # get length so it doesn't need to be recalculated

    # test for even length and take len/2 and len/2 -1 divided over 2.0 for float division
    if list_length %2 == 0:
        return (input_list[list_length/2] + input_list[(list_length/2) - 1]) / 2.0
    else:
        return input_list[list_length/2]
```

Exercises

5. Watch this video on how Yelp determines whether to recommend a review: https://youtu.be/PniMEnM89iY

   Based on the video, think about the features necessary for the algorithm to determine whether to recommend a review and write a short blogpost on the class Tumblr discussing what features you think Yelp is using and how they might quantifying what they're trying to measure

Key Things to Take Away from These Exercises

- I don’t hate you, I promise
- This more than one way to accomplish a task in code
- Some solutions are more elegant than others
- Elegance isn’t a requirement, but becomes important when working with lots of data (or many operations)
- The built-in functions are generally going to be more efficient and robust
- Seriously, I don’t hate you

10 MIN BREAK

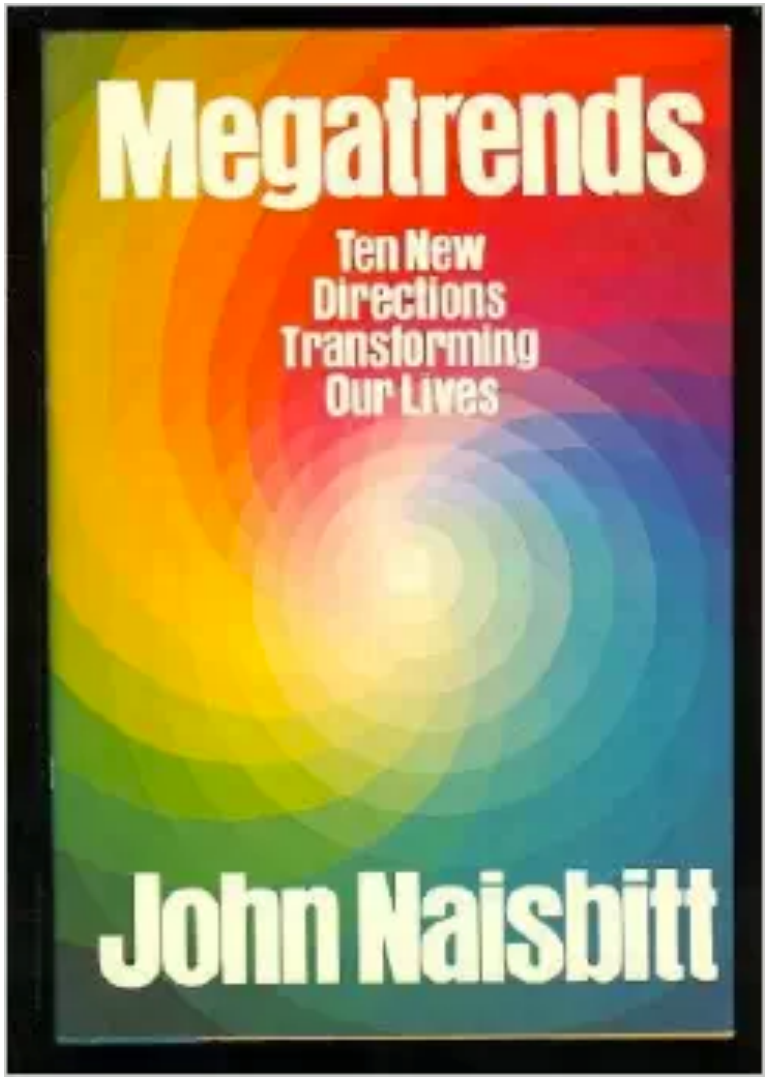
Fill out this Google Doc: http://bit.ly/1f59Fki

Name

height (in inches)

age

siblings (not including you)



We are drowning in information but starved for knowledge.

- John Naisbitt

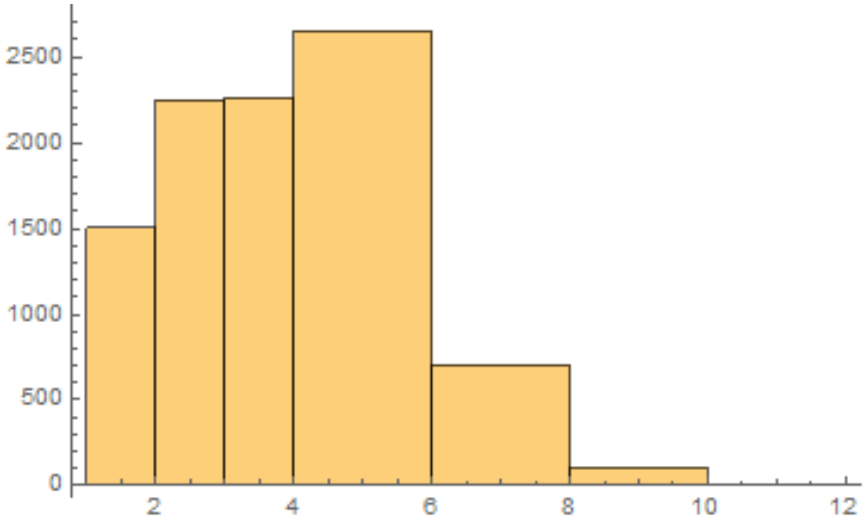
Why statistics?

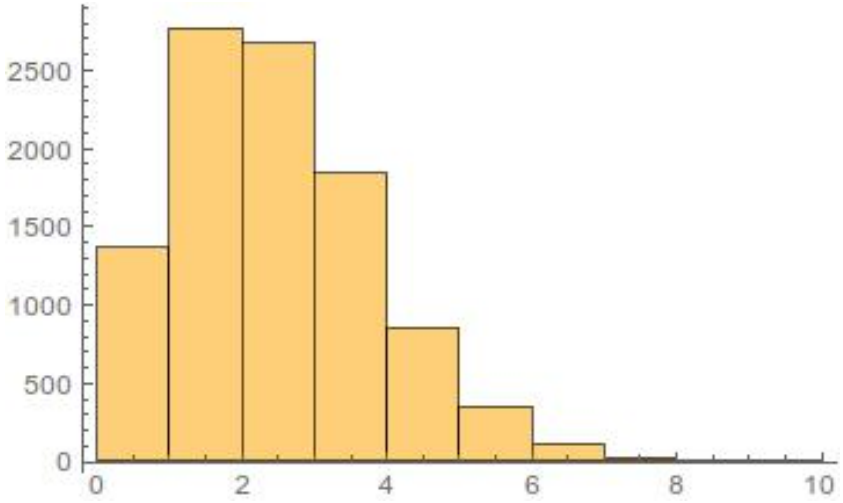
- Tools for extracting meaning from data
- Commonly understood ways of communicating meaning to others

DATA DISTRIBUTIONS

Histogram

- Charts the frequency of instances in the data
- Shows the frequency distribution
- Values are grouped into class intervals
- Best to have a consistent size to class intervals

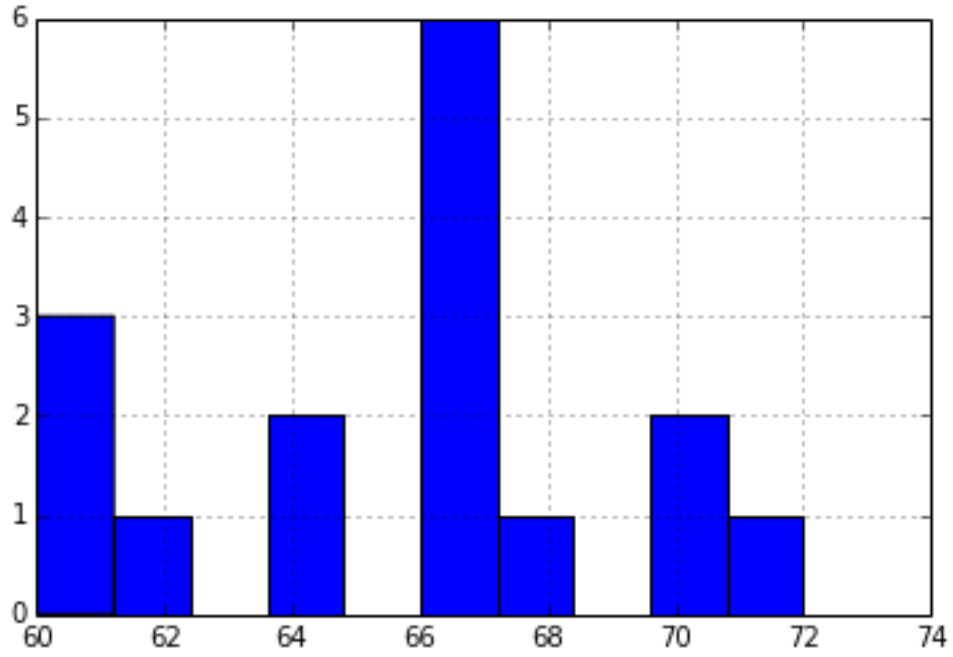




http://mathematica.stackexchange.com/questions/59520/histogram-with-variable-bin-size

Histogram in Python

```
df['Height'].hist()
```



HOLY COW, BATMAN!
WE'D BETTER CHECK THE DOCUMENTATION!

Parameters:

data *: DataFrame*

column *: string or sequence*

If passed, will be used to limit data to a subset of columns

by *: object, optional*

If passed, then used to form histograms for separate groups

grid *: boolean, default True*

Whether to show axis grid lines

xlabelsize *: int, default None*

If specified changes the x-axis label size

xrot *: float, default None*

rotation of x axis labels

ylabelsize *: int, default None*

If specified changes the y-axis label size

yrot *: float, default None*

rotation of y axis labels

ax *: matplotlib axes object, default None*

sharex *: boolean, default True if ax is None else False*

In case subplots=True, share x axis and set some x axis labels to invisible; defaults to True if ax is None otherwise False if an ax is passed in; Be aware, that passing in both an ax and sharex=True will alter all x axis labels for all subplots in a figure!

sharey *: boolean, default False*

In case subplots=True, share y axis and set some y axis labels to invisible

figsize *: tuple*

The size of the figure to create in inches by default

layout: (optional) a tuple (rows, columns) for the layout of the histograms

bins: integer, default 10

Number of histogram bins to be used

kwds *: other plotting keyword arguments*

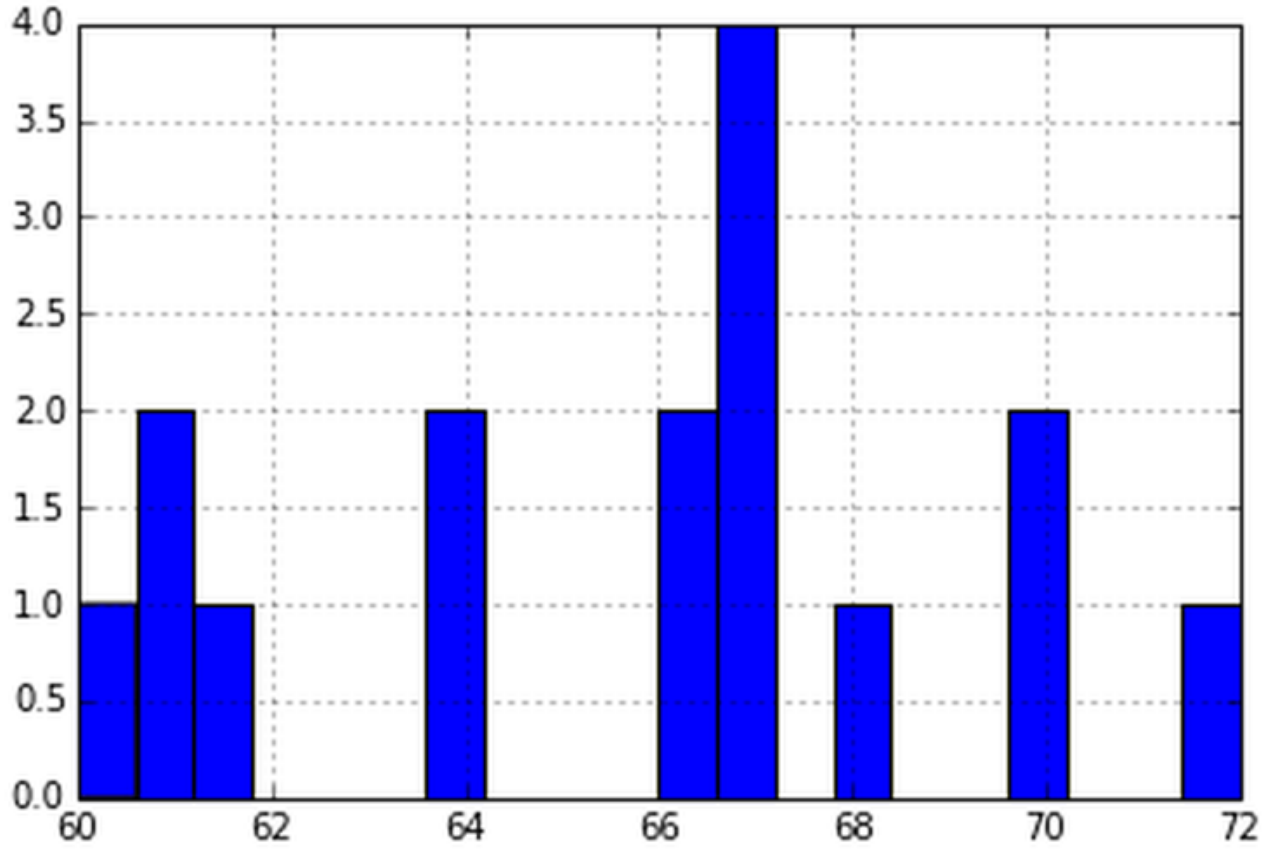
To be passed to hist function

http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.hist.html

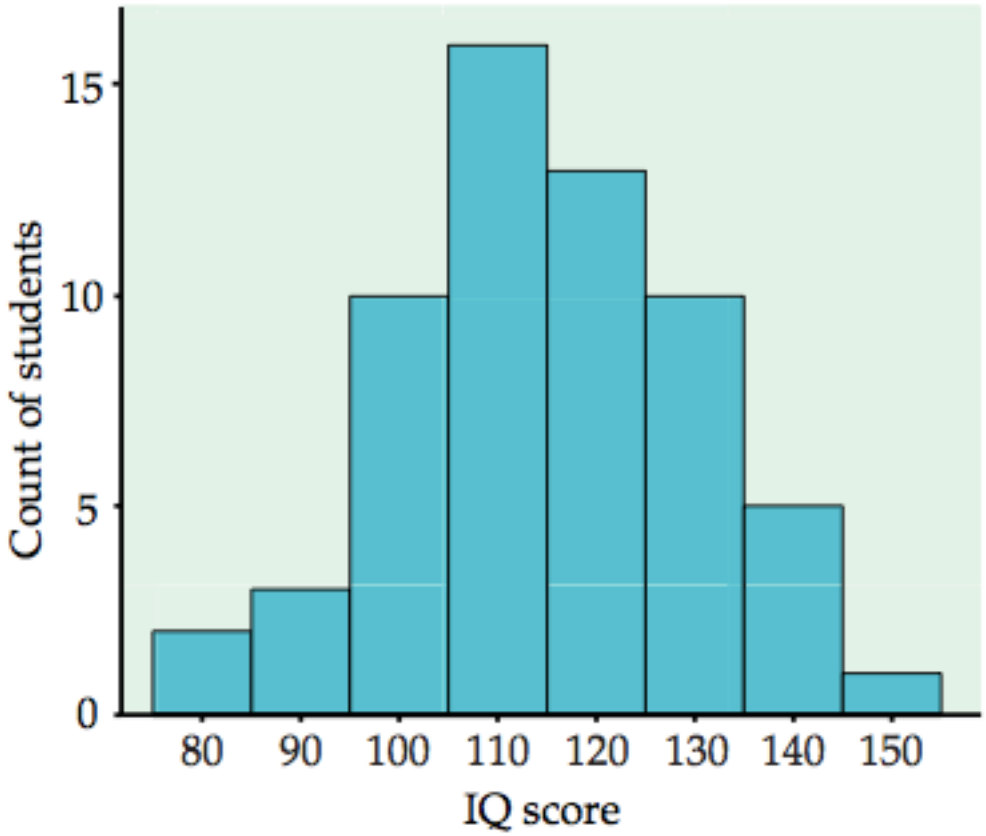
Histogram in Python

```
df['Height'].hist(bins=20)
```

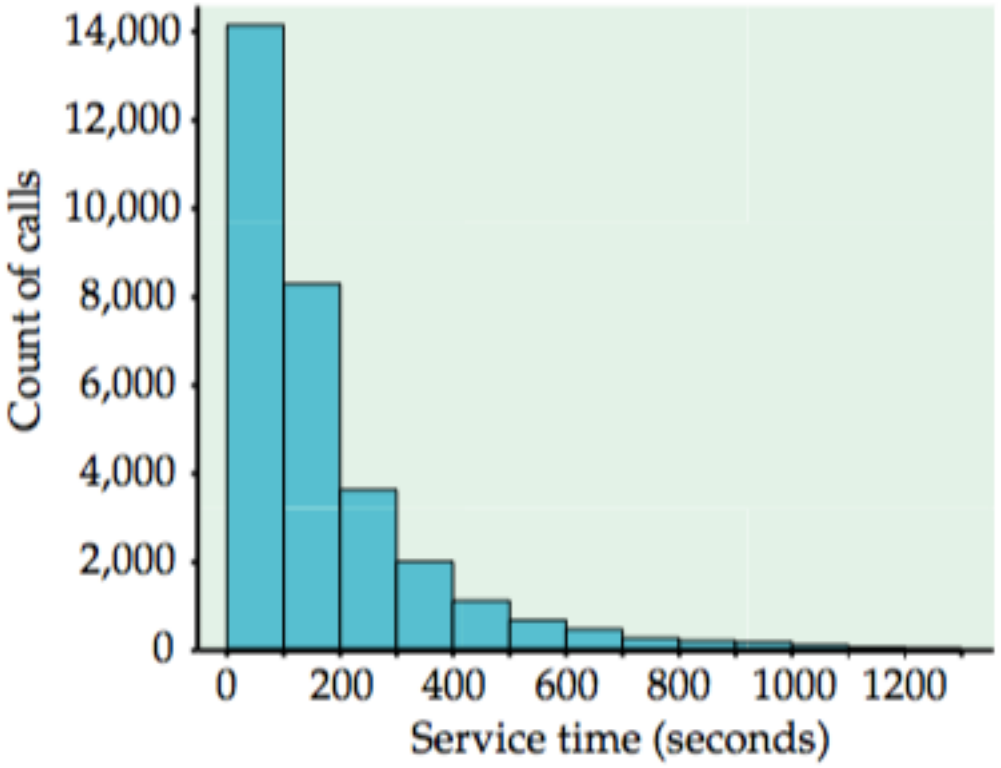
```
<matplotlib.axes.AxesSubplot at 0x10d8b01d0>
```



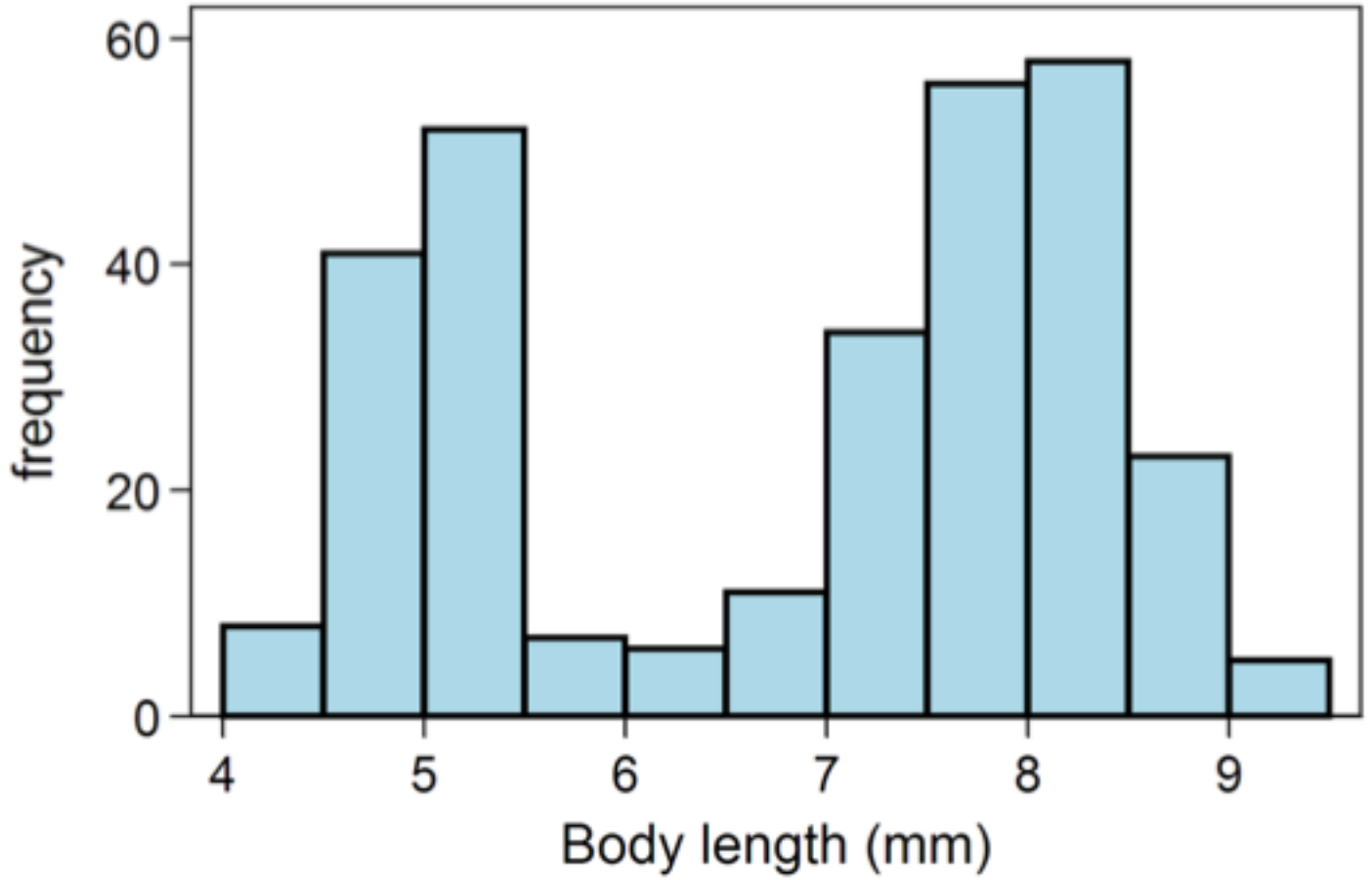
Normal Distribution



Long-tail Distribution



Bi-modal



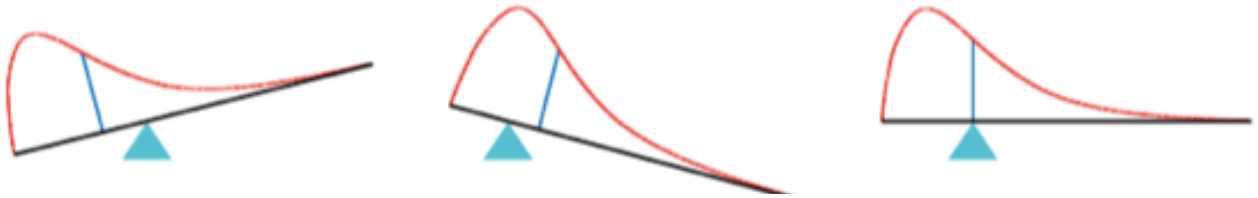
Mean

- A representative value for the data
- Usually what people mean by “average”
- Calculate by adding all the values together and dividing by the number instances
- Sensitive to extremes

```
df.mean()
```

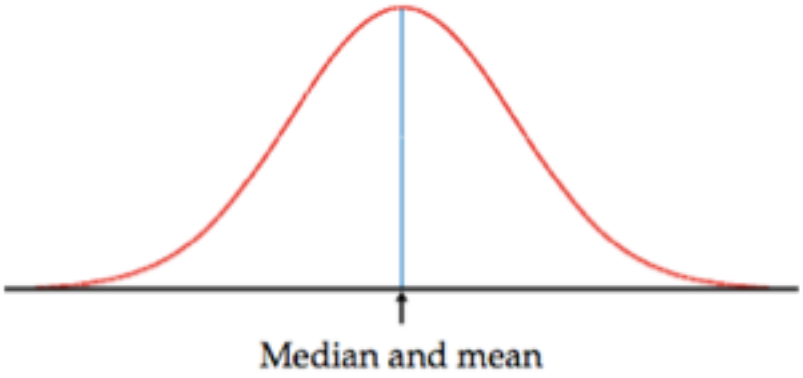
Median

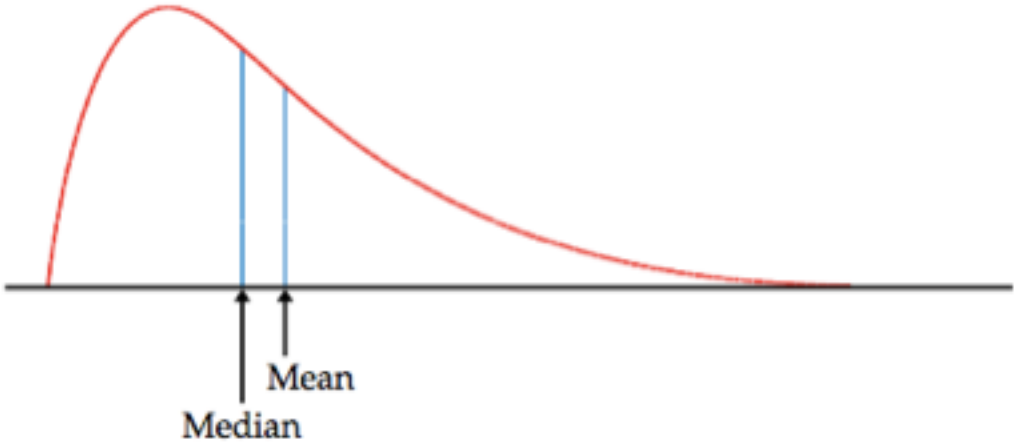
- The “middle” value of a data set
  - Center value of a data set with an odd number of values
  - Sum of two middle values divided by 2 if the number of items in a data set is even
- Resistant to extreme values



```
df.median()
```

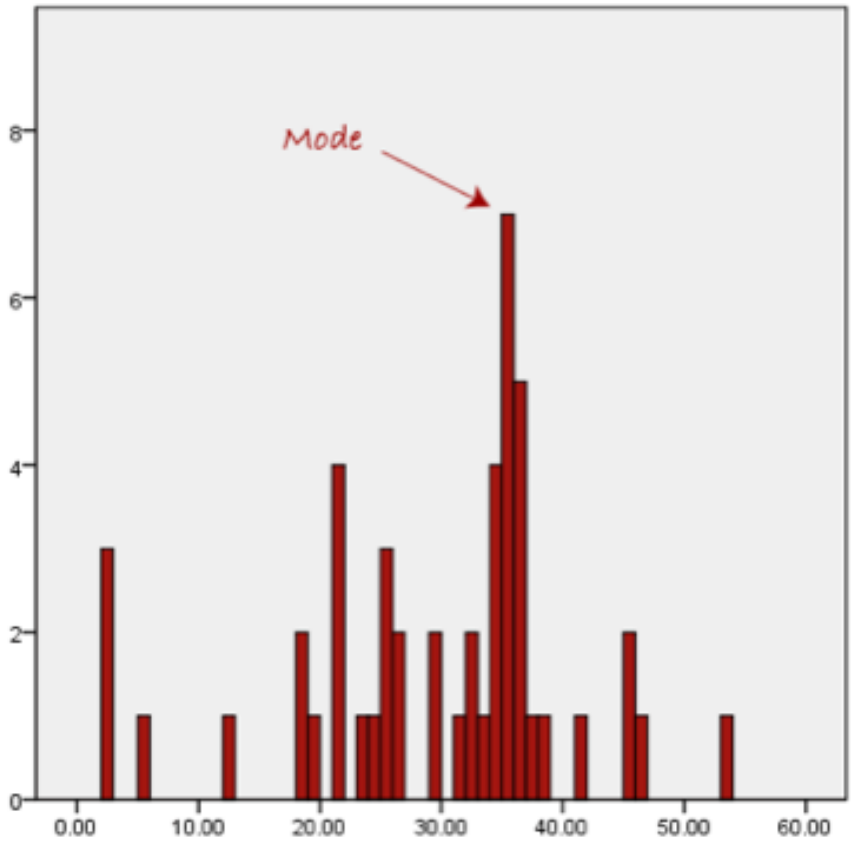
Mean vs Median





Mode

- The most frequent value in a dataset
- Often used for categorical data




```
df.mode()
```

Measures of Central Tendency

- Quantitative data tends to cluster around some central value
- Contrasts with the spread of data around that center (i.e. the variability in the data)
- Measurements
  - Mean is a more precise measure and more often used
  - Median is better when there are extreme outliers
  - Mode is used when the data is categorical (as opposed to numeric)

Why are these Important?

- Help us understand our data
- Measures of central tendency are often used as inputs to algorithms

HOW DO WE MEASURE VARIABILITY IN A DATA SET?

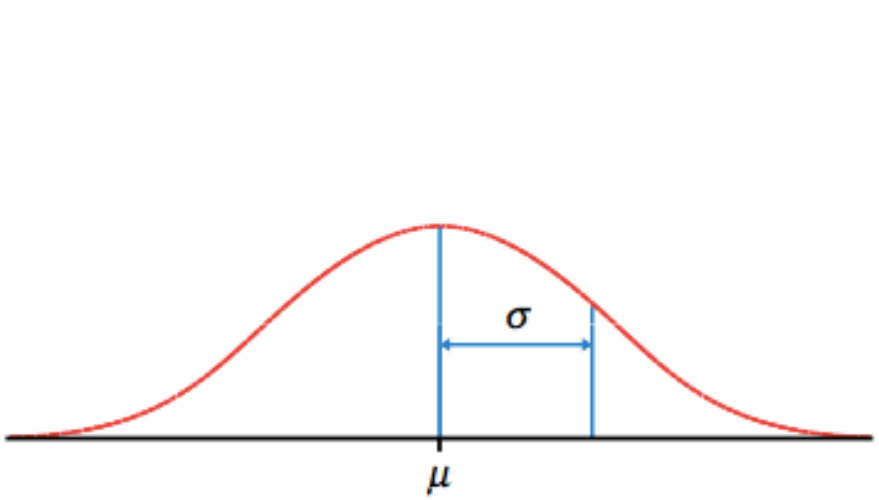
Range

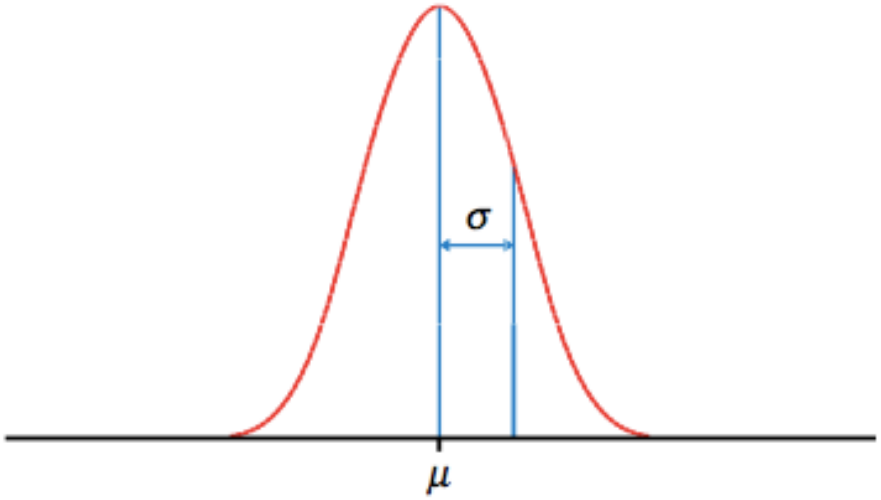
- The gap between the minimum value and the maximum value
- Calculated by subtracting the minimum from the maximum
- What's the range in the data we collected today?

```
df['Height'].max() - df['Height'].min()
```

Standard Deviation

- The average distance of each data point from the mean
- Larger the standard deviation, the greater the spread





Formula for Standard Deviation

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}} \qquad \sigma = \sqrt{\frac{\sum (x - \mu)^2}{N}}$$

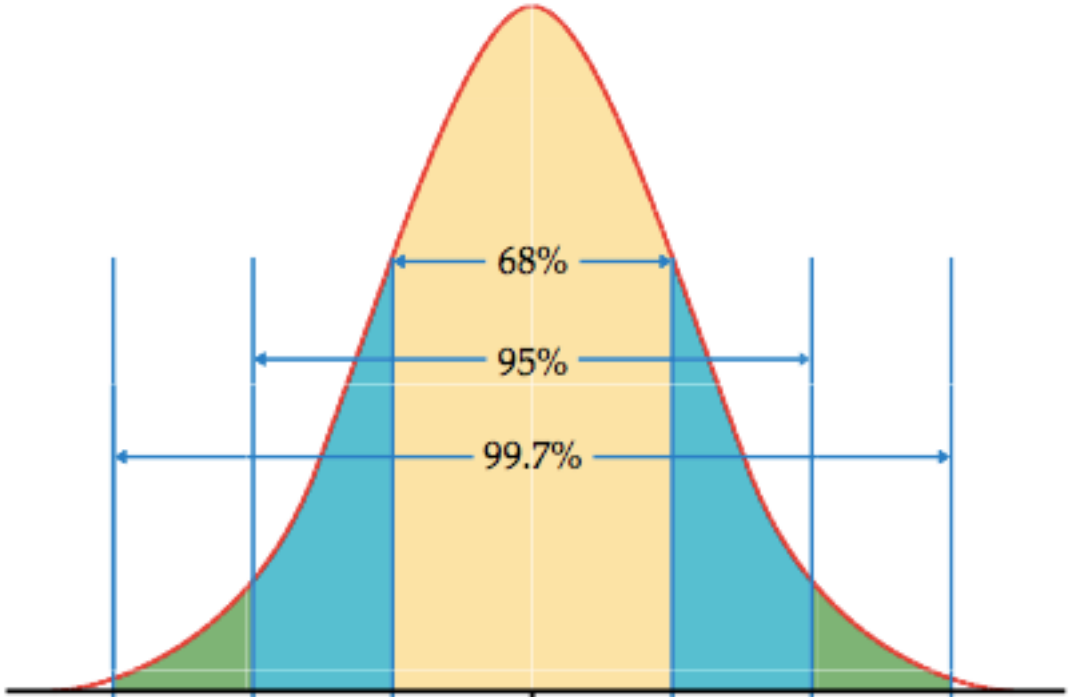
Calculating the Standard Deviation

1. Subtract the mean from each data point
2. Square the result
3. Sum them together
4. Divide by the number of instances (minus 1)
5. Take the square root

Do this for the data we collected at the beginning of class

```
df.std()
```

Standard Deviation



Calculating the Variance

1. Subtract the mean from each data point
2. Square the result
3. Sum them together
4. Divide by the number of instances
5. ~~Take the square root~~

Measures of Variability

- Describe the distribution of our data
- Measures
  - Range (Maximum – Minimum)
  - Standard Deviation
    - Useful for comparing across different data sets
  - Variance
    - Less useful -> can't meaningful compare the data

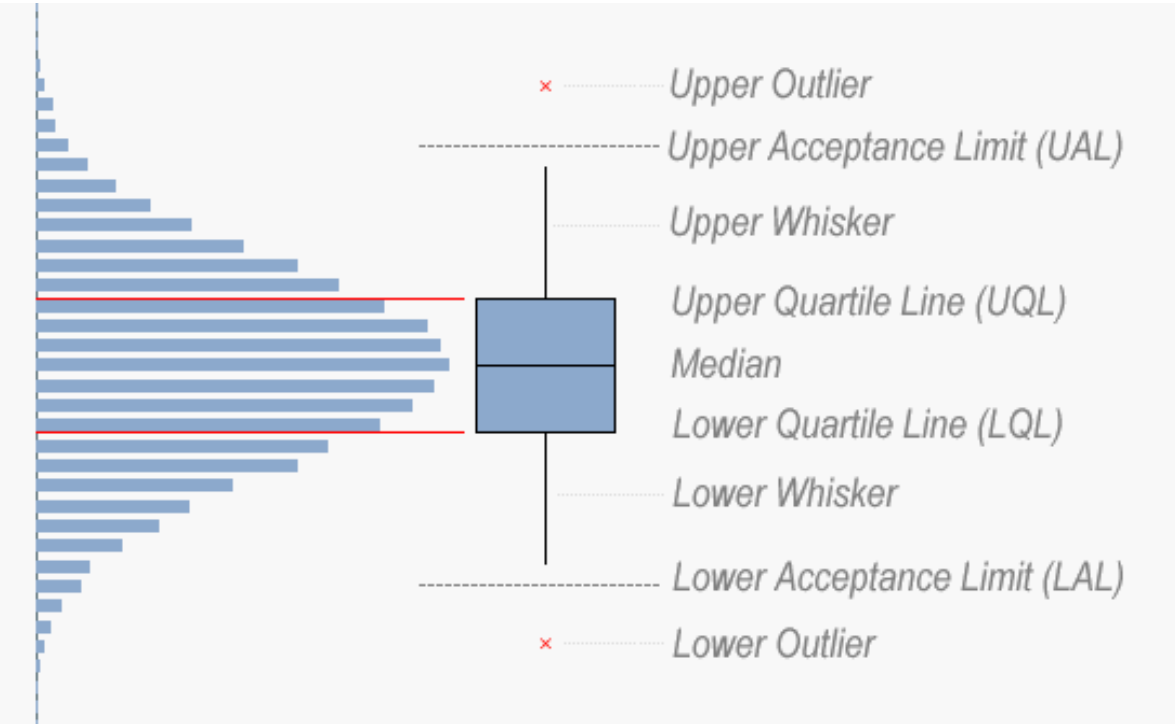
Quartiles

- Median splits the data set into two equal groups
- Quartiles split the data into four equal groups
- First quartile is 0-25% of the data
- Second quartile is 25-50% of the data
- Third quartile is 50-75% of the data
- Fourth quartile is 75-100% of the data

```
df.quantile(q=0.25)
```

Inter-Quartile Range

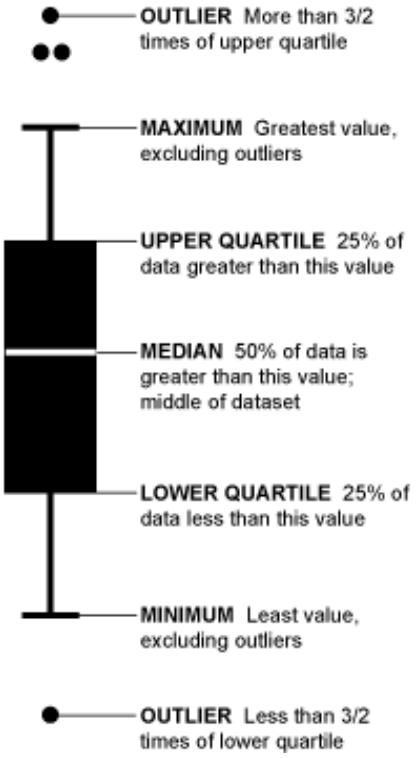
- "Middle" 50% of data (between 1st Quartile and 3rd Quartile)



https://community.qlik.com/blogs/qlikviewdesignblog/2014/08/18/recipe-for-a-box-plot

Outliers

- Anything that lies 1.5 times the IQR



http://flowingdata.com/2008/02/15/how-to-read-and-use-a-box-and-whisker-plot/

```
df.boxplot(column='Height')
```

Measures of Variability

- Describe the distribution of our data
- Measures
  - Range (Maximum – Minimum)
  - Standard Deviation
  - Variance
  - Inter-quartile Range

Descriptive Statistics

- Quantitatively describe the main features of a dataset
- Help distinguish distributions and make them comparable

DESCRIBING WITH STATISTICS

http://simplystatistics.org/2012/11/26/the-statisticians-at-fox-news-use-classic-and-novel-graphical-techniques-to-lead-with-data/

Group Exercise

- Break into groups and record the height, age, and number of siblings for the group
- How does your group compare to the class as a whole?

WRAP-UP

Exercises

1. Write code necessary to analyze the relationship between median income and recycling rate in New York City Community Boards (using 2013_NYC_CD_MedianIncome_Recycle.xlsx). Calculate:
   - coefficient of correlation
   - coefficient of determination
2. What is the relationship between these two variables? Write a short Tumblr post outlining the relationship based on your findings

Exercises

3. Using data from the FiveThirtyEight post http://53eig.ht/1e2aV6U, write code to calculate the correlation of the responses from the poll.
4. Write a short Tumblr post describing the results of your analysis