YData: Introduction to Data Science

Class 15: review!

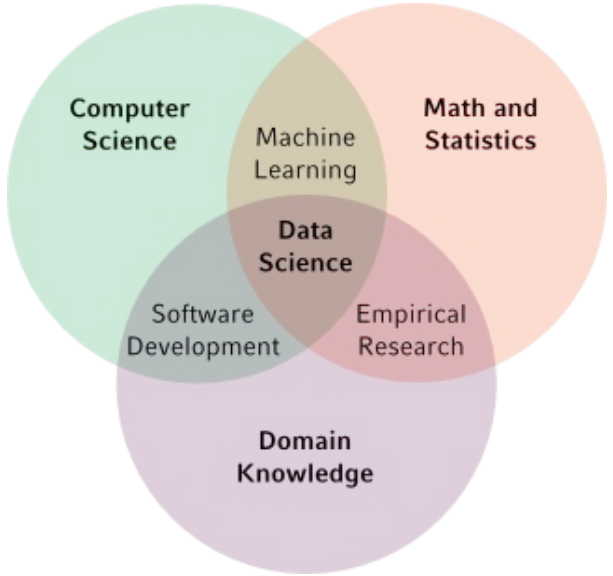
Overview

Quick overview over topics we have covered

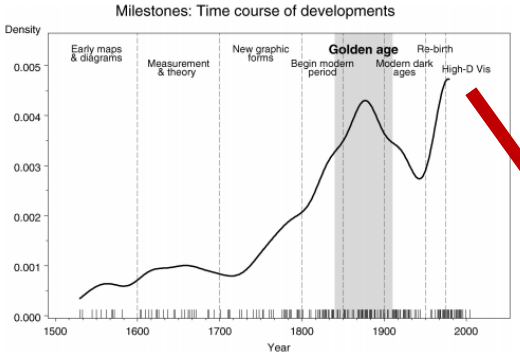
Questions you have

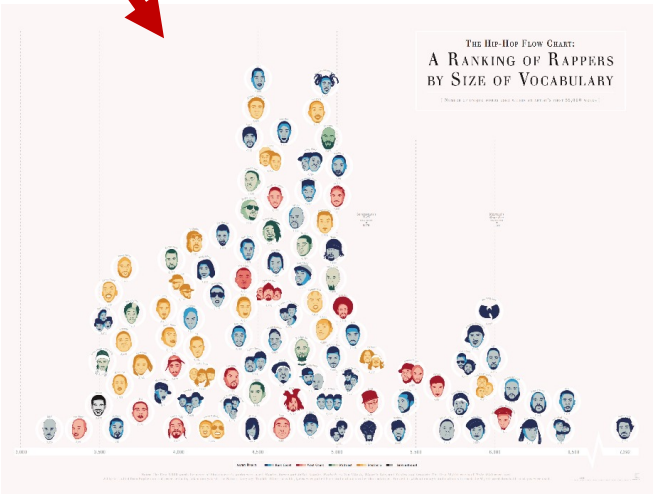
Practice problems

Quick review of what is Data Science?



Data Science is a broadening of data analyses beyond what traditional Statistical mathematical/inferential analyses to use more computation





Many other fields impacted by 'Data Science

- Making business decisions
- Predictive medicine
- Fraud detection
- Etc.

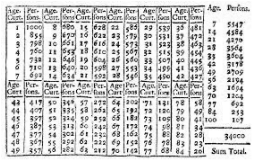
Examples:

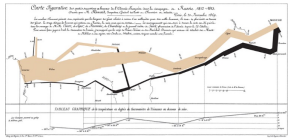
- NYC city bike visualization
- Wind map visualization

Ethical concerns around privacy, fairness and other issues

Quick review of the history of Data Science

(a very incomplete list)





Data

Ishango bone
(20,000 BCE)

Cuneiform tablets
(4,000 BCE)

Quipus in South America
(1100-1500)

Demographics
(1600's)

Golden age of data visualization
(1850-1900)

Big data
(now)

Probability

Key Take Away

Probability models dominated data analysis prior to using computational methods

Initial development
(1600's)

"Small data"

Probability in Statistics
(1820's – 1950's)

Math Stats dominates
(1900-1960's)

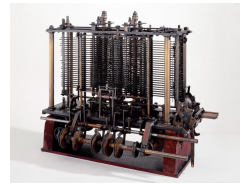
"Big data"

Computers

Abacus
(2400 BCE)

Antikythera mechanism
(100 BCE)

Analytical Engine
(1800's)



Hollerith Tabulating Machine
(1890)

Mainframes, PCs, Internet, etc.
(1950-present)

Quick review of Python basics

Expressions and type

- my_num = 2 * 3
- my_string = 'cat' + ' ' + 'hat'

List and dictionaries

- my_list = [1, 2, 3 , 4, 5, 'six'] # create a list
- my_list2 = my_list[0:3] # get the first 3 elements
- my_dict = { 'a': 7, 'b': 20} # create a dictionary

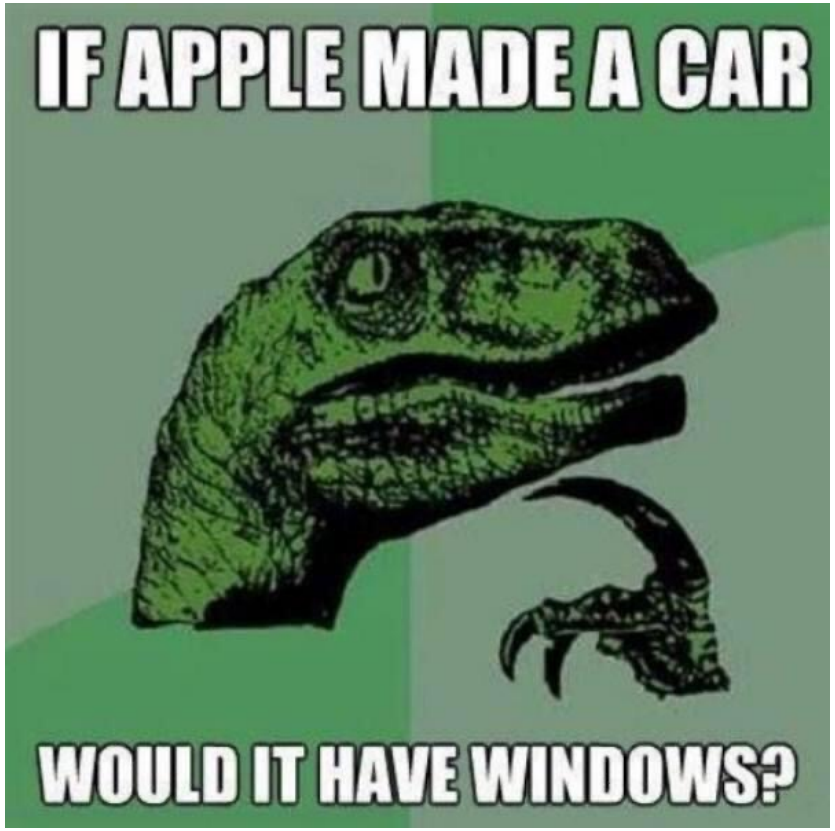
Loops

```
for i in range(10):
    print(i**3)
```

Quick review of Python basics

Conditional statements

```
num = 5
if num == 1:
    print("Monday")
elif num == 2:
    print("Tuesday")
elif num == 3:
    print("Wednesday")
elif num == 4:
    print("Thursday")
elif num == 5:
    print("Friday")
elif num == 6:
    print("Saturday")
elif num == 7:
    print("Sunday")
else:
    print("Invalid input")
```

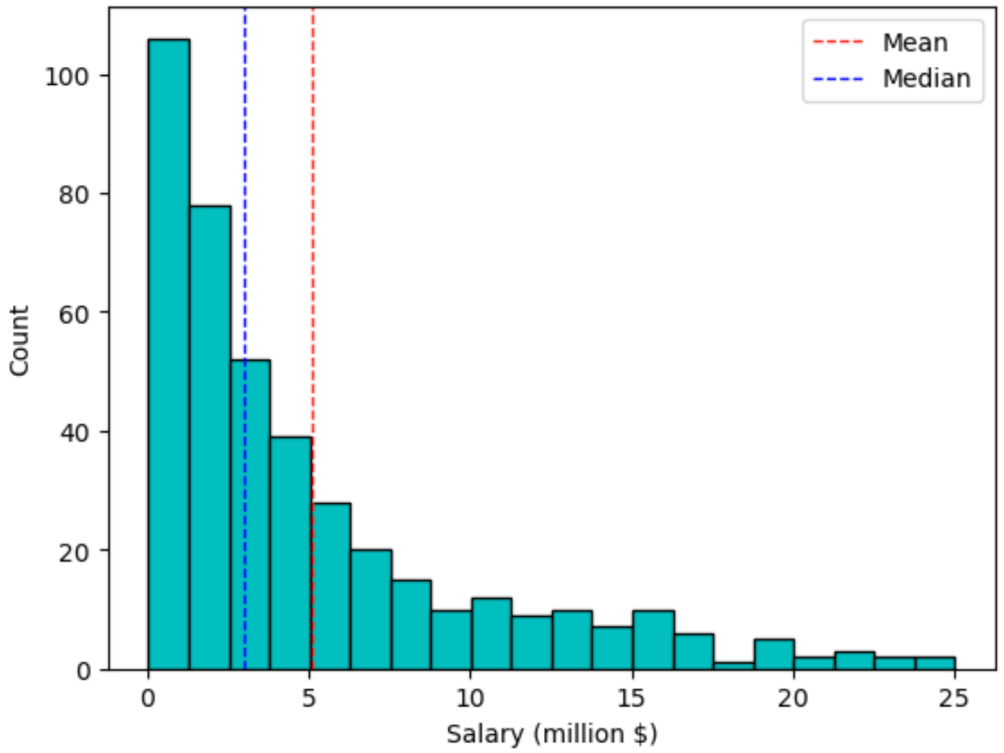


Quick review of statistics and plots

We have discussed statistics:

```
import statistics
statistics.median(data_list)
statistics.mean(data_list)
```

```
import matplotlib.pyplot as plt
plt.hist(data_list)
```



Quick review of NumPy arrays and functions

Hopefully we are comfortable with:

- Creating arrays and accessing elements: np.array()
- Getting their type and size: .shape, .dtype
- Using numeric functions: np.sum(), np.mean(), np.diff()
- Using broadcasting: my_array * 2, my_array1 - my_array
- Creating Boolean arrays: my_array < 5, my_array == "C"
- Using Boolean masks to get elements: my_array[my_array < 5]

Quick review of NumPy arrays and functions

The NumPy functions:

- np.sum()
- np.max(), np.min()
- np.mean(), np.median()
- np.diff() # takes the difference between elements
- np.cumsum() # cumulative sum

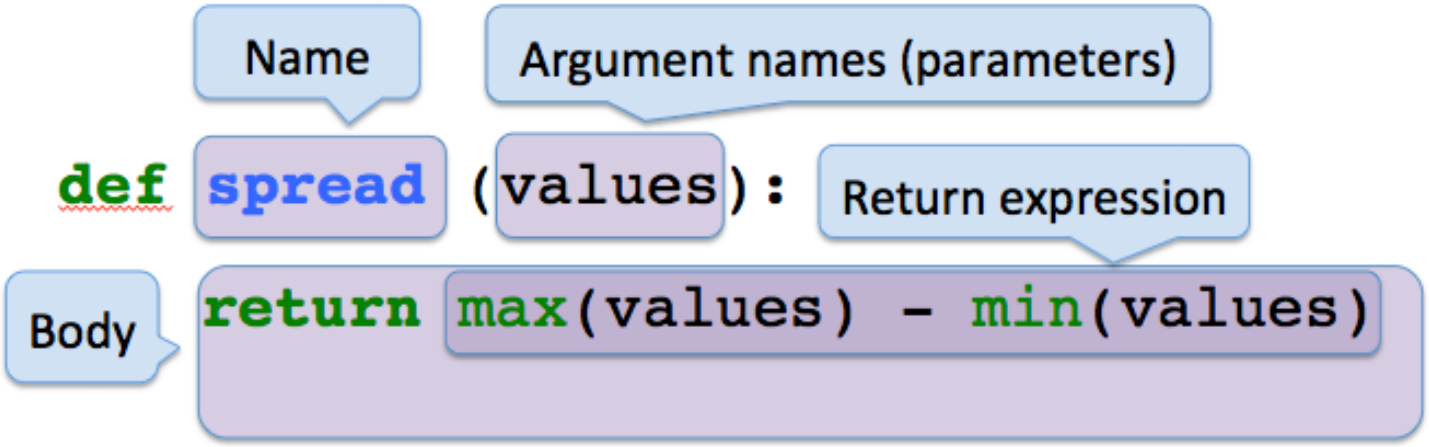
There are also "broadcast" functions that operate on all elements in an array

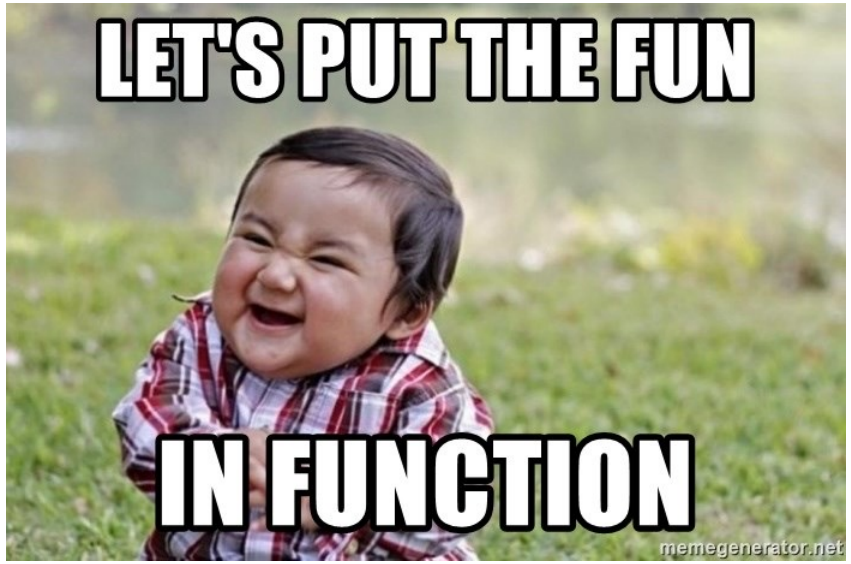
- my_array = np.array([12, 4, 6, 3, 4, 3, 7, 4])
- my_array * 2

- my_array2 = np.array([10, 9, 2, 8, 9, 3, 8, 5])
- my_array - my_array2

Quick review of writing your own functions

User-defined functions give names to blocks of code





Functions can return tuples which allow us to return multiple names

- val1, val2 = my_function()

Quick review of pandas DataFrames

PLAYER	POSITION	TEAM	SALARY
str	str	str	f64
"Paul Millsap"	"PF"	"Atlanta Hawks"	18.671659
"Al Horford"	"C"	"Atlanta Hawks"	12.0
"Tiago Splitter...	"C"	"Atlanta Hawks"	9.75625
"Jeff Teague"	"PG"	"Atlanta Hawks"	8.0
"Kyle Korver"	"SG"	"Atlanta Hawks"	5.746479

Pandas DataFrame hold Table data

Selecting columns:

- my_df[[“col1”, “col2”]].copy() # getting multiple columns using a list

Extracting rows:

- my_df.iloc[0] # getting a row by number
- my_df.loc[“index_name”] # getting a row by Index value
- my_df [my_df[“col_name”] == 7] # getting rows using a Boolean mask

Quick review of pandas DataFrames

Sorting rows of a DataFrame

```
my_df.sort_values("col_name", ascending = False)    # sort from largest to smallest
```

Adding a new:

- my_df["new_col"] = values_array

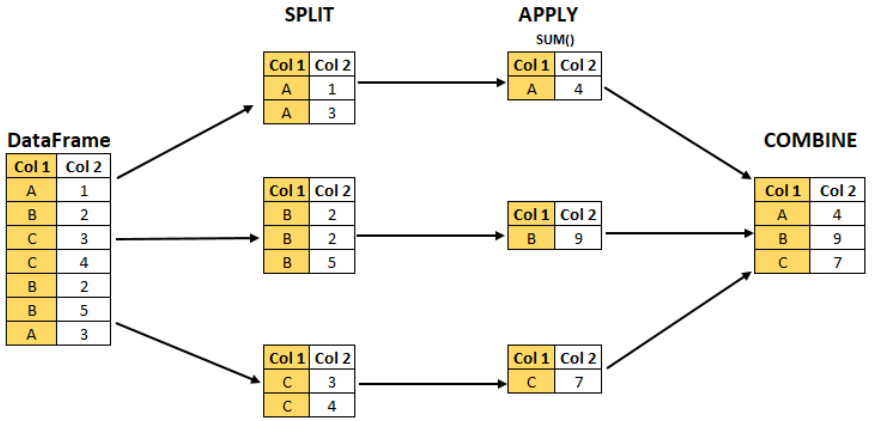
Renaming a column:

- rename_dictionary = {"old_col_name": "new_col_name"}
- my_df.rename(columns = rename_dictionary)

Quick review of pandas DataFrames

We can get statistics separately by group:

- dow.groupby("Year").agg("max")

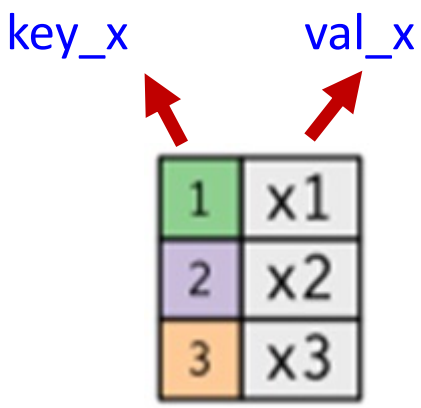



```
my_df.groupby("group_col_name").agg(
      new_col1 = ('col_name', 'statistic_name1’),
      new_col2 = ('col_name', 'statistic_name2’),
      new_col3 = ('col_name', 'statistic_name3')
)
```

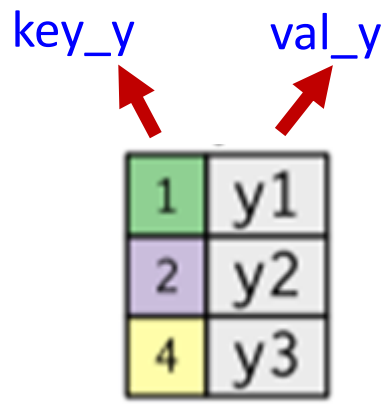
Quick review of joining data frames

Suppose we have two DataFrames (or Series) called **x_df** and **y_df**

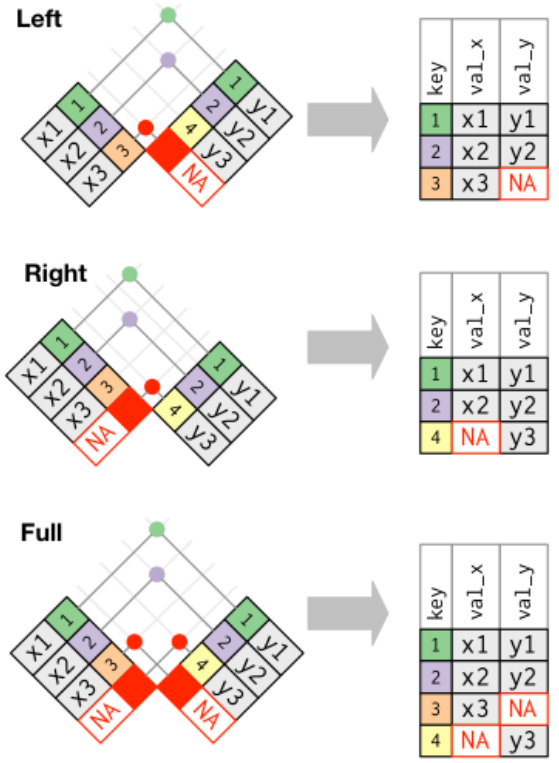
- x_df have two columns called key_x, and val_x
- y_df has two columns called key_y and val_y



DataFame: x_df



DataFrame y_df



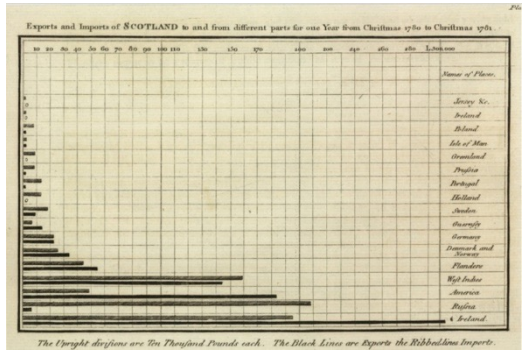
Joins have the general form:

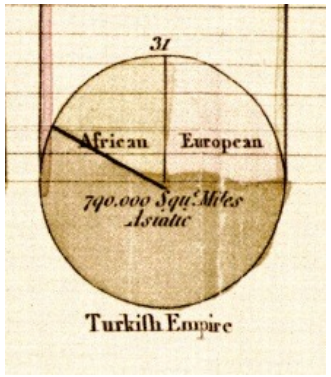
```
x_df.merge(y_df, left_on = "key_x",  right_on = "key_y")
```

Quick review of the history of data visualization

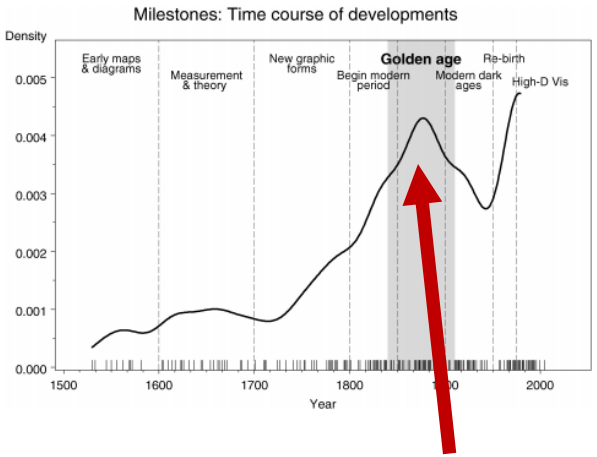
The age of modern statistical graphs began around the beginning of the 19th century

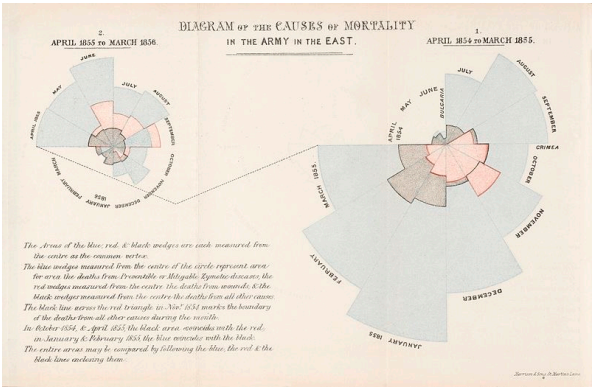
[William Playfair](#) (1759-1823)

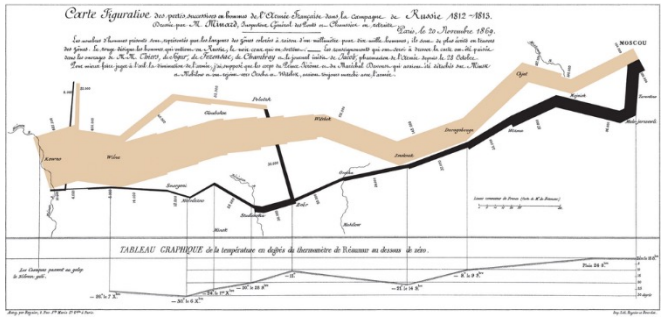


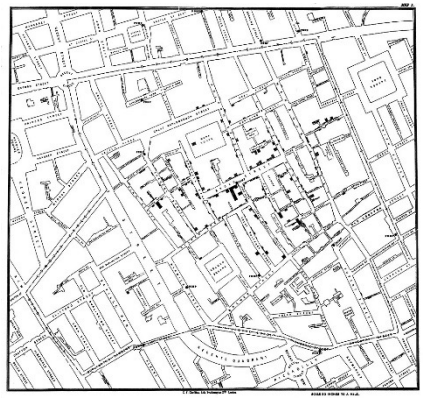


According to Friendly, statistical graphics researched its golden age between 1850-1900



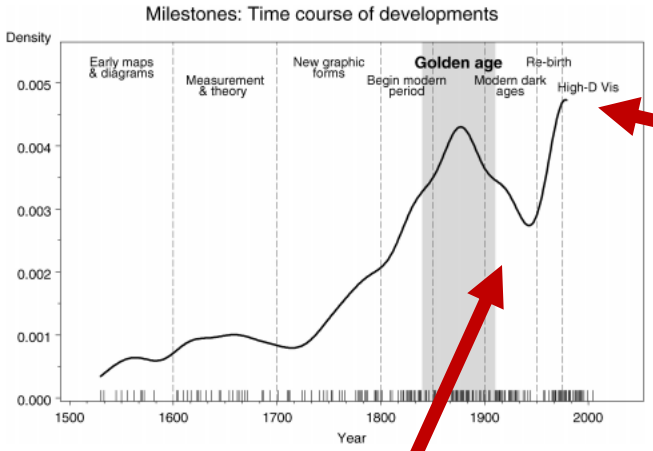


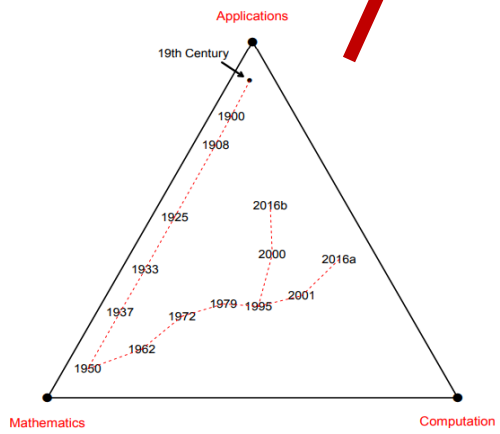


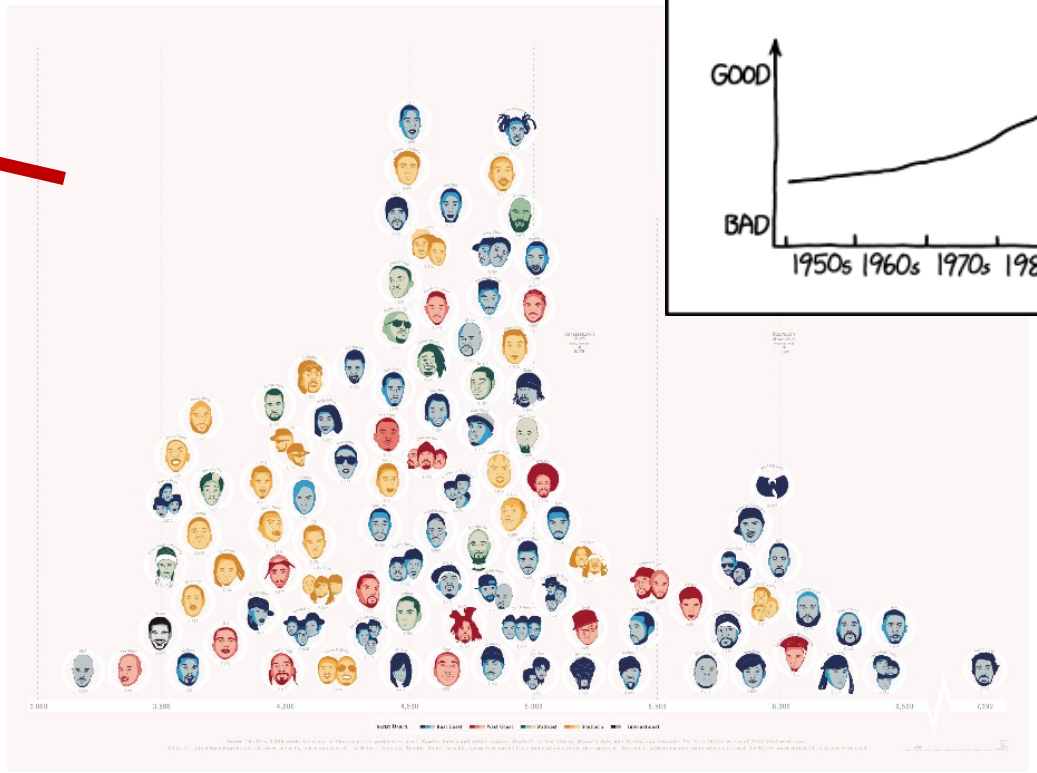


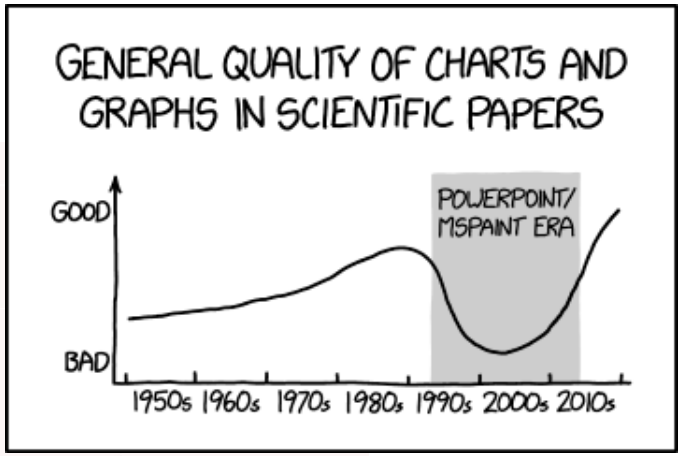
Quick review of the history of data visualization

"Graphical dark ages" around 1950









Currently undergoing a "Graphical re-birth"

Computer Age Statistical Inference, Efron and Hastie

Quick review of visualizing data with matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations.

- import matplotlib.pyplot as plt

Types of plots we have created

- plt.plot(x, y, '-o') # line plot/scatter plot
- plt.hist(data)
- plt.boxplot(data)
- plot.scatter(x, y, s = , color = , marker =)

Quick review of visualizing data with matplotlib

Make sure always label your axes:

- plt.ylabel("y label")
- plt.xlabel("x label")
- plt.title("my title")
- plt.plot(x, y, label = "blah")
- plt.legend()

We can create subplots:

- plt.subplot(1, 2, 1);
- plt.plot(x1, y1);

Quick review of seaborn

Figure level plots are grouped based on the types of variables being plotted

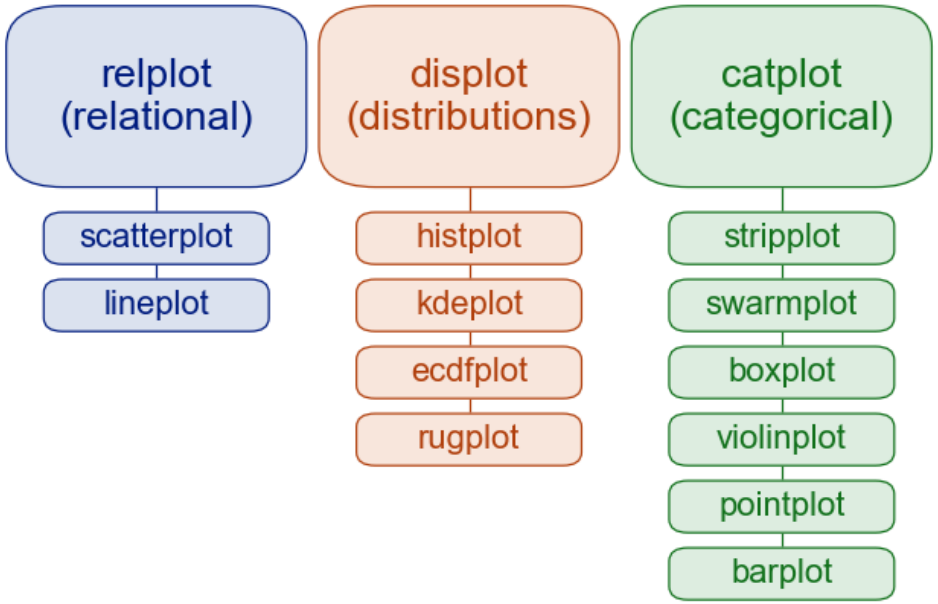
In particular, there are plots for:

1. Two quantitative variables
   - sns.relplot()

2. A single quantitative variable
   - sns.displot()

Quantitative variable compared across different categorical levels
   - sns.catplot()

Figure level plots



Quick review of text manipulation

There are many string functions including:

- isalpha(): Returns True if all characters in the string are in the alphabet
- isnumeric(): Returns True if all characters in the string are numeric
- isspace(): Returns True if all characters in the string are whitespaces
- split(separator_string): Splits the string at the specified separator, and returns a list
- splitlines(): Splits the string at line breaks and returns a list
- join(a_list): Converts the elements of an iterable into a string
- count(substring): Returns the number of times a specified value occurs in a string
- replace(original_str, replacement_str): Replace a substring with a different string.
- f"my string {value_to_fill} will be filled in"


WHEN DOES THIS HAPPEN IN THE MOVIE
NOW, EVERYTHING THAT'S HAPPENING NOW IS HAPPENING NOW
imgflip.com

Questions???

Practice questions…