YData: Introduction to Data Science

Class 04: Descriptive statistics and plots

Overview

Continuation of python basics:

- Comparisons
- Quick discussion of additional string methods

Statistics and data visualizations:

- Categorical data: proportions, bar plots and pie charts
- Quantitative data: mean median and histograms
- If there is time:
  - Measures of spread: standard deviation and z-scores

Announcement: Homework 2

Homework 2 has been posted!

```
import YData

YData.download_homework(2)
```

It is due on Gradescope on Sunday September 14th at 11pm

- **Be sure to mark each question on Gradescope!**

Notes:

- There is an ~18 page reading from the book "Data and the American Dream" that you need to do, so I recommend you get started on this soon.

Review: Lists

Last class we discussed lists

- Suppose we have the list: z = [1, 2, 3, 2, 1, [8, 9, 10]]

<u>Retrieving items from a list and slicing</u>:

- z[2]
- z[1:4:2]

<u>List functions and methods:</u>

- len(z)
- z.count(2)
- new_list = z + [1, 2, 3] # returns a new list and saves it to the name new_list
- z.append(7) # modifies the list z (and returns None)

<u>Functions on lists of numbers:</u>

- sum(), max(), min()

Review: String methods

Last class we also discussed string methods:

- Suppose we have a string my_string = "Hello Yale"

String methods

- my_string.upper()
- my_string.replace("Yale", "Whale")
- string_list = my_string.split(" ")
- ", ".join(string_list)

Let's do a very quick warmup exercise in Jupyter!

Booleans and comparisons

Comparisons

We can use mathematical operators to compare numbers and strings

- Results return Boolean values True and False

Comparison	Operator	True example	False Example
Less than	<	2 < 3	2 < 2
Greater than	>	3 > 2	3 > 3
Less than or equal	<=	2 <= 2	3 <= 2
Greater or equal	>=	3 >= 3	2 >= 3
Equal	==	3 == 3	3 == 2
Not equal	!=	3 != 2	2 != 2

True is equal to 1

False is equal to 0

True + True + False
is equal to…

2

We can compare strings alphabetically

- 'a' < 'b'

Let's explore this in Jupyter!

String methods: checking string properties

There are also many functions to check properties of strings including:

- .isalnum(): Returns True if all characters in the string are alphanumeric
- .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

- .islower(): Returns True if all characters in the string are lower case
- .isupper(): Returns True if all characters in the string are upper case
- .istitle(): Returns True if the string follows the rules of a title

Let's explore this in Jupyter!

Additional string methods

String methods: string padding

Often we want to remove extra spaces (called "white space") from the front or end of a string

Conversely, sometimes we want to add extra spaces to make a set of strings the same length

- This is known as "string padding"

Python strings have a number of methods that can pad/trim strings including:

- .strip(): Returns a trimmed version of the string (i.e., with no leading or trailing white space)
  - Also, .rstrip() and .lstrip(): Returns a right/left trim version of the string
- .center(num): Returns a centered string (with equal padding on both sides)
  - Also .ljust(num) and .rjust(num): Returns a right justified version of the string
- .zfill(num): Fills the string with a specified number of 0 values at the beginning

Let's explore this in Jupyter!

String methods: filling in strings with values

There are a number of ways to fill in strings parts of a string with particular values

Perhaps the most useful is to use "f strings", which have the following syntax such as:

- value_to_fill = "my_value"
- f"my string {value_to_fill} will be filled in"

Let's explore this in Jupyter!

Brief mention: regular expressions

More complex text manipulation can be done using “regular expressions”

```
import re
bool(re.match("m.ss", "mess"))
```

We might discuss regular expression later in the semester…

A brief introduction to statistics and data visualization…

The Bechdel test

1. Has to have >= 2 women

2. Who talk to each other

3. About something other than a man

Where do samples/data come from?

Suppose we had a random sample of 1794 movies

- The ***sample size*** is 1794 (n = 1794)

The Bechdel data

Variables

Cases

title	clean_test	binary	budget	domgross	budget_2013	domgross_2013
21 & Over	notalk	FAIL	13000000	25682380.0	13000000	25682380.0
Dredd 3D	ok	PASS	45000000	13414714.0	45658735	13611086.0
12 Years a Slave	notalk	FAIL	20000000	53107035.0	20000000	53107035.0
2 Guns	notalk	FAIL	61000000	75612460.0	61000000	75612460.0
42	men	FAIL	40000000	95020213.0	40000000	95020213.0

Categorical and Quantitative Variables

Categorical Variable

Quantitative Variable

Cases

title	clean_test	binary	budget	domgross	budget_2013	domgross_2013
21 & Over	notalk	FAIL	13000000	25682380.0	13000000	25682380.0
Dredd 3D	ok	PASS	45000000	13414714.0	45658735	13611086.0
12 Years a Slave	notalk	FAIL	20000000	53107035.0	20000000	53107035.0
2 Guns	notalk	FAIL	61000000	75612460.0	61000000	75612460.0
42	men	FAIL	40000000	95020213.0	40000000	95020213.0

Categorical data

statistics

A ***statistic*** is a number computed from a sample of data

Quantitative Variable

title	clean_test	binary	budget	domgross	budget_2013	domgross_2013
21 & Over	notalk	FAIL	13000000	25682380.0	13000000	25682380.0
Dredd 3D	ok	PASS	45000000	13414714.0	45658735	13611086.0
12 Years a Slave	notalk	FAIL	20000000	53107035.0	20000000	53107035.0
2 Guns	notalk	FAIL	61000000	75612460.0	61000000	75612460.0
42	men	FAIL	40000000	95020213.0	40000000	95020213.0

95174784

Proportions

For a *single* ***categorical variable***, the main ***statistic*** of interest is the *proportion* in each category

$$\text{Proportion in a category} = \frac{\text{number in that category}}{\text{total number}}$$

Proportions

For a *single* ***categorical variable***, the main ***statistic*** of interest is the *proportion* in each category

- E.g., the proportion of movies that passed the Bechdel test

Proportion passed =

$$\frac{\text{number of movies that passed}}{\text{total number}}$$

Let's explore this in Jupyter!

Categorical Variable

title	clean_test	binary
21 & Over	notalk	FAIL
Dredd 3D	ok	PASS
12 Years a Slave	notalk	FAIL
2 Guns	notalk	FAIL
42	men	FAIL

Visualizing Categorical Data

Plotting data

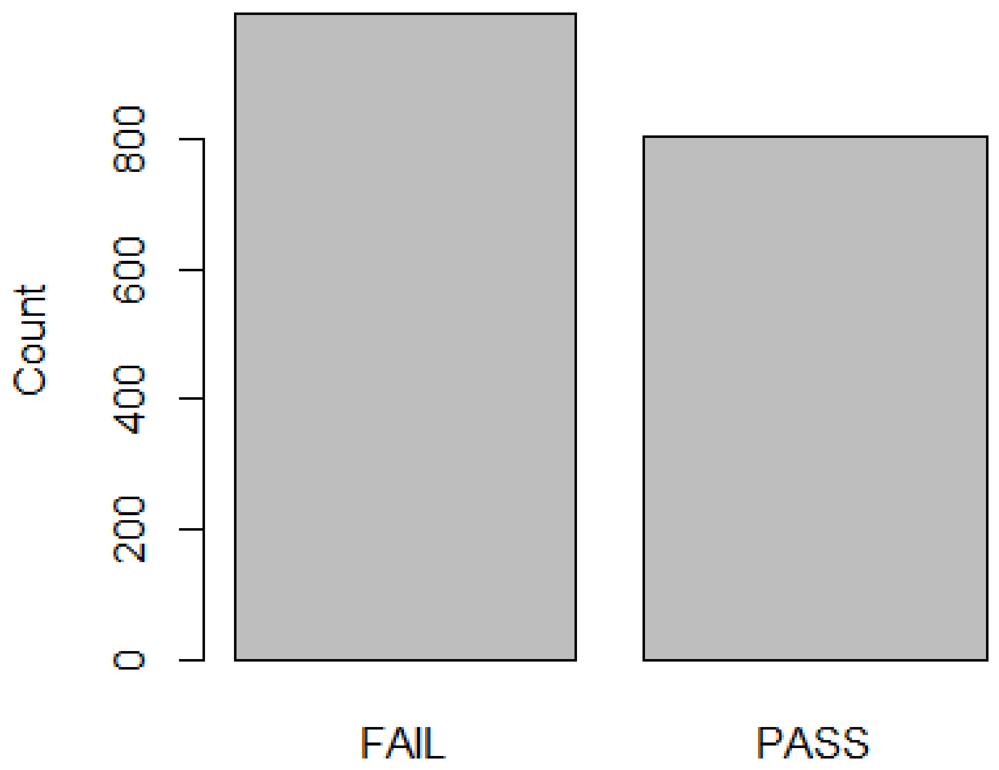
To create basic visualizations in Python we can use the matplotlib library

```
import matplotlib.pyplot as plt
```

We can then create plots using functions such as:

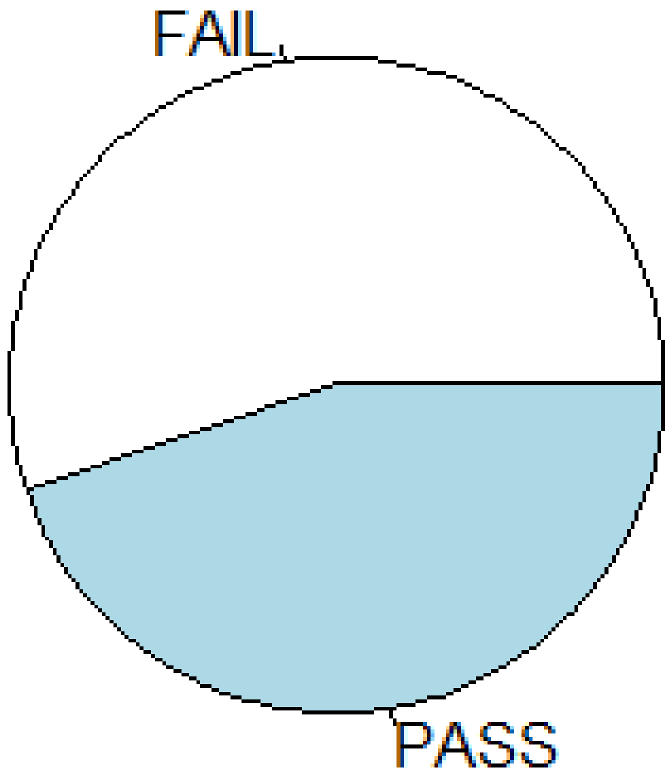
- plt.plot()
- plt.bar()
- plt.hist()
- etc.

Visualizing categorical data: The Bar Chart



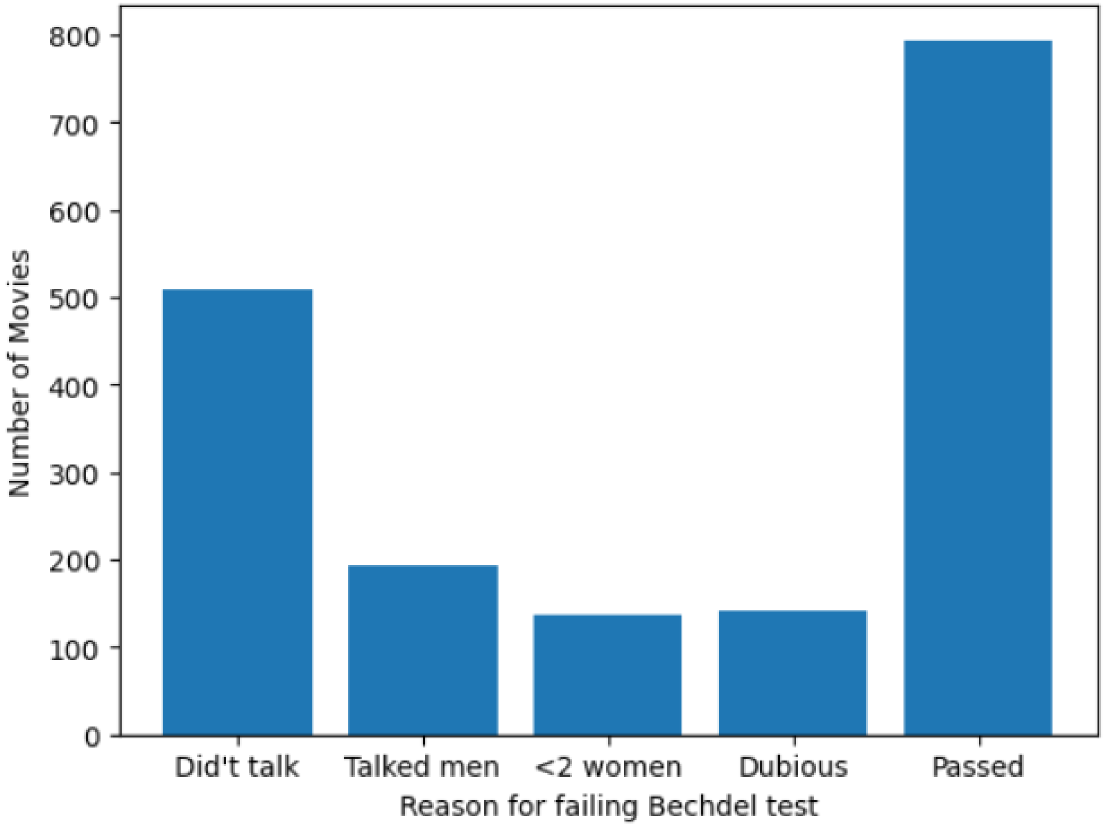
Matplotlib: `plt.bar(labels, data)`

Visualizing categorical data: The Pie Chart



Matplotlib: `plt.pie(data)`

Labeling axes!




```
plt.bar(reason_names, reason_counts);

plt.xlabel("Reason failed");

plt.ylabel("Number of movies")
```

Quantitative data

Quantitative data

To explore quantitative data, let's look at how much revenue each movie made in the United States in (2013) inflation adjusted dollars

- domgross_2013

Quantitative Variable

title	clean_test	binary	budget	domgross	budget_2013	domgross_2013
21 & Over	notalk	FAIL	13000000	25682380.0	13000000	25682380.0
Dredd 3D	ok	PASS	45000000	13414714.0	45658735	13611086.0
12 Years a Slave	notalk	FAIL	20000000	53107035.0	20000000	53107035.0
2 Guns	notalk	FAIL	61000000	75612460.0	61000000	75612460.0
42	men	FAIL	40000000	95020213.0	40000000	95020213.0

Visualizing quantitative data: histograms

Movie US revenue (in millions of dollars):

- 25.68, 13.61, 53.11, 236.84, …

To create a histogram we create a set of intervals

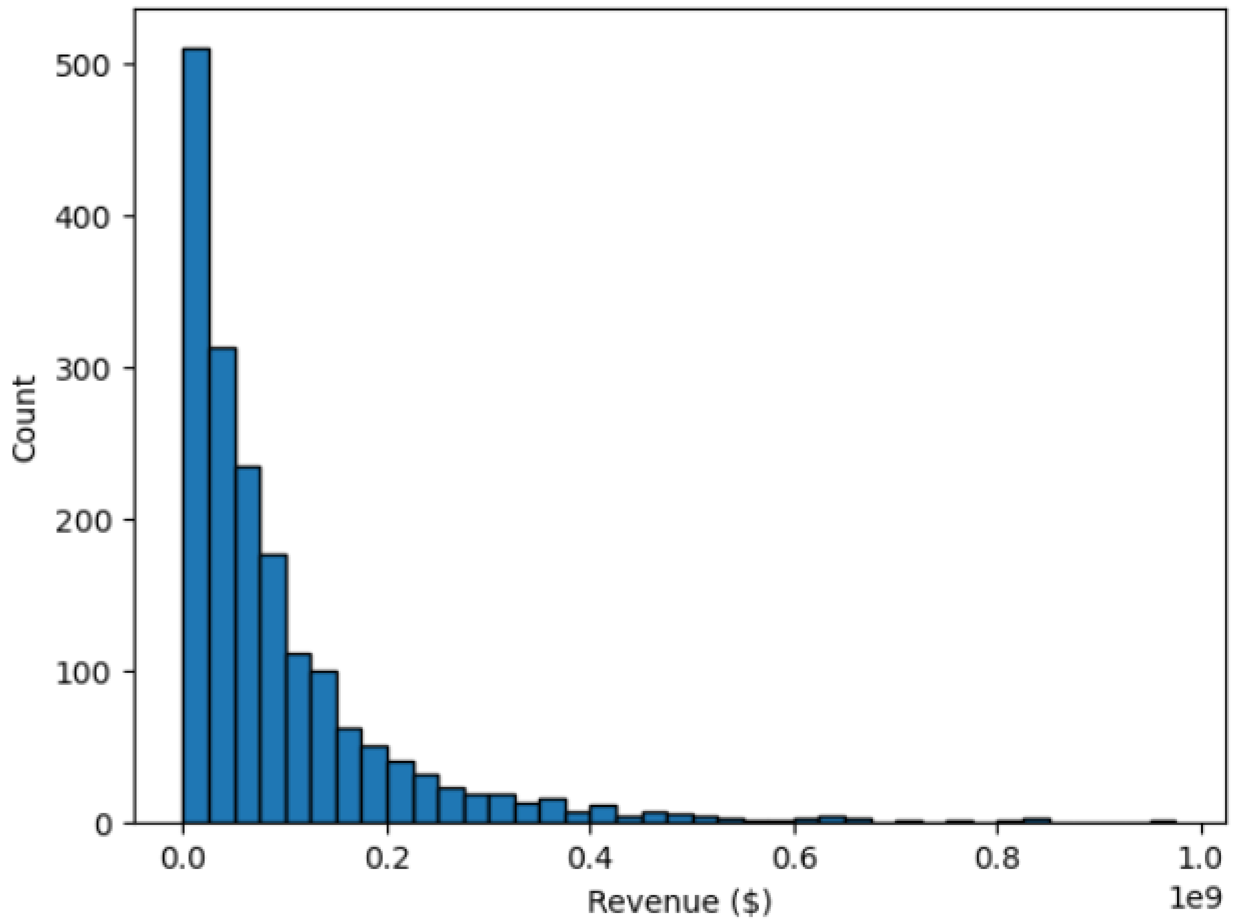
- 0-25, 25-50, 50-75, … 200-250, 250-300

We count the number of points that fall in each interval

We create a bar chart where the height of the bars is the counts in each bin

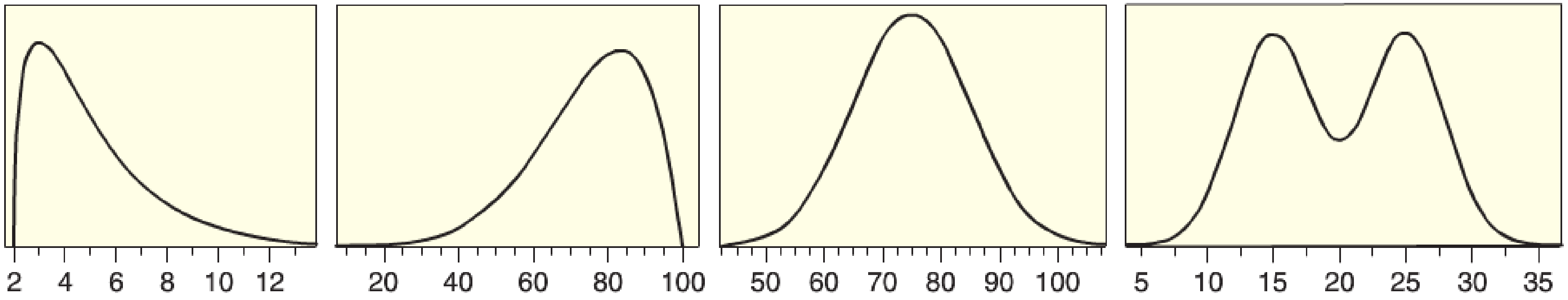
Histograms – movie US revenue

Domgross range	Frequency Count
(0 – 25]	510
(25 – 50]	312
(50 – 75]	234
(75 – 100]	176
(100 – 125]	111
(125 – 150]	99
(150 – 175]	62
(175 – 200]	51
(200 – 225]	40
(225 – 250]	32



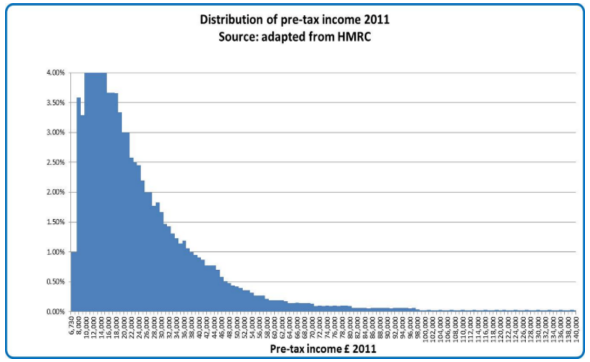
Matplotlib: `plt.hist(data)`

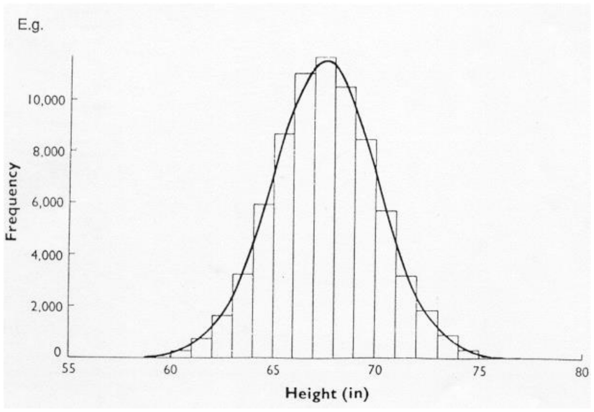
Common shapes of data distributions

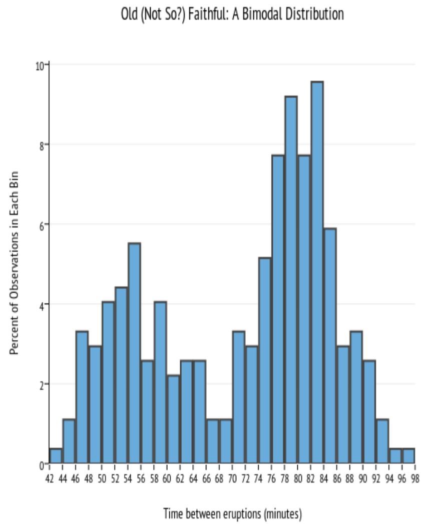


(a) Skewed to the right (b) Skewed to the left (c) Symmetric and bell-shaped (d) Symmetric but not bell-shaped

Income distribution

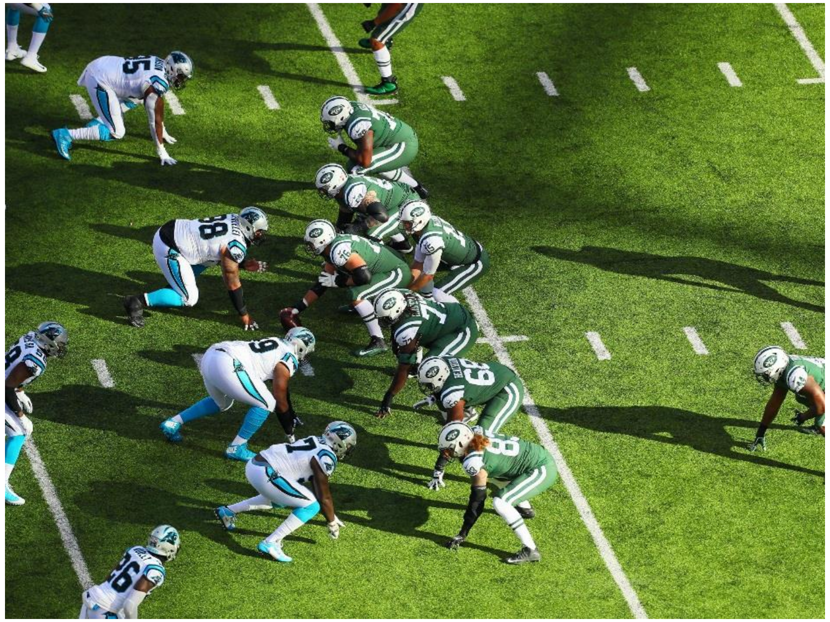


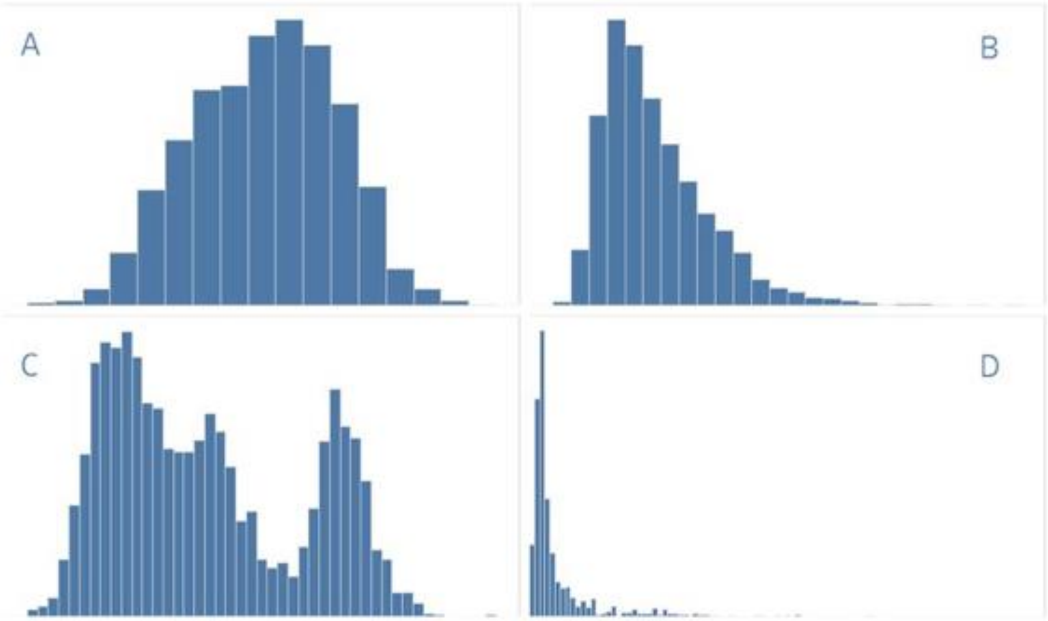




Neat facts – the average NFL player is:

- 1. **Age**: Is about 25 years old
- 2. **Height**: Is just over 6'2" in height
- 3. **Weight**: Weighs a little more than 244lbs
- 4. **Salary**: Makes slightly less than $1.5M in salary per year

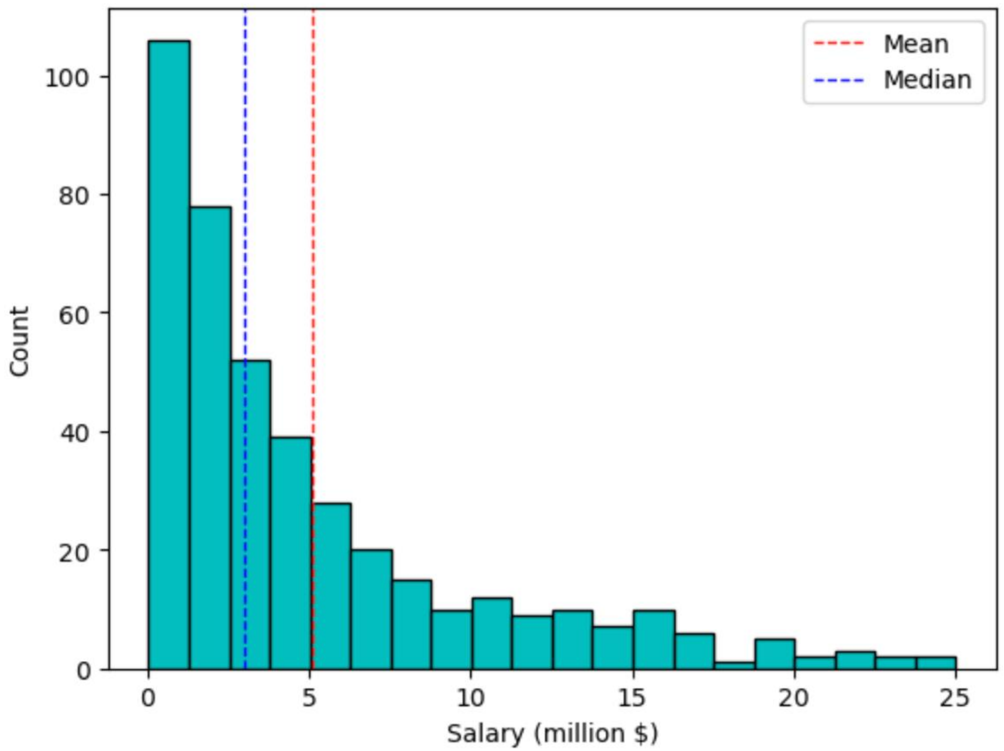




Question: Can you tell which histogram goes with which trait?

Quantitative data: statistics for central tendency

Two statistics for measuring the "central value" of a sample of quantitative data are the ***mean*** and the ***median***



The mean

$$\text{Mean} = \frac{\text{Sum of all data values}}{\text{Number of data values}}$$

$$\text{Mean} = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

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

The median

The **median** is a value that splits the data in half

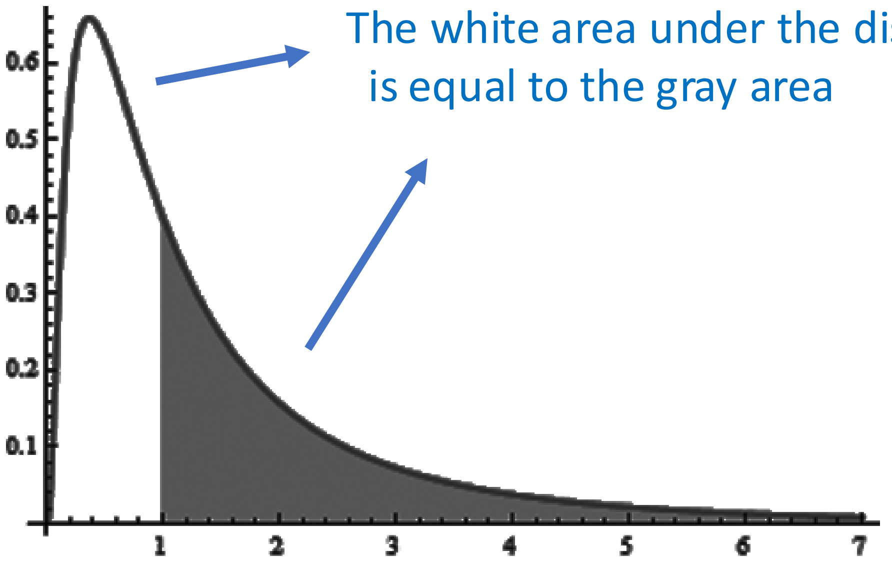
- i.e., half the values in the data are smaller than the median and half are larger

To calculate the median for a data sample of size *n*, sort the data and then:

- If n is odd: The middle value of the sorted data

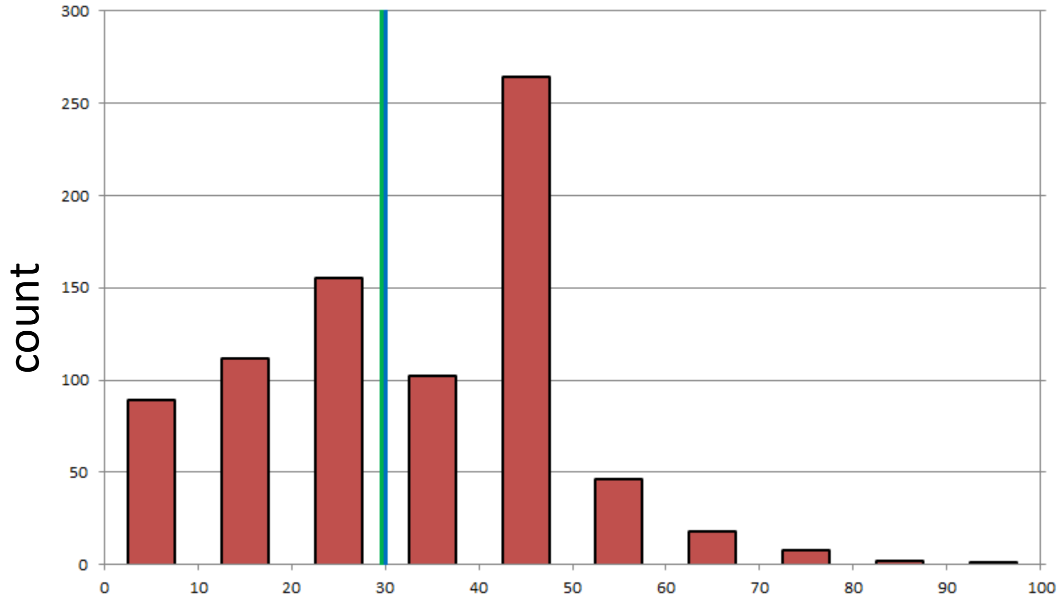
- If n is even: The average of the middle two values of the sorted data

The median



The white area under the distribution is equal to the gray area

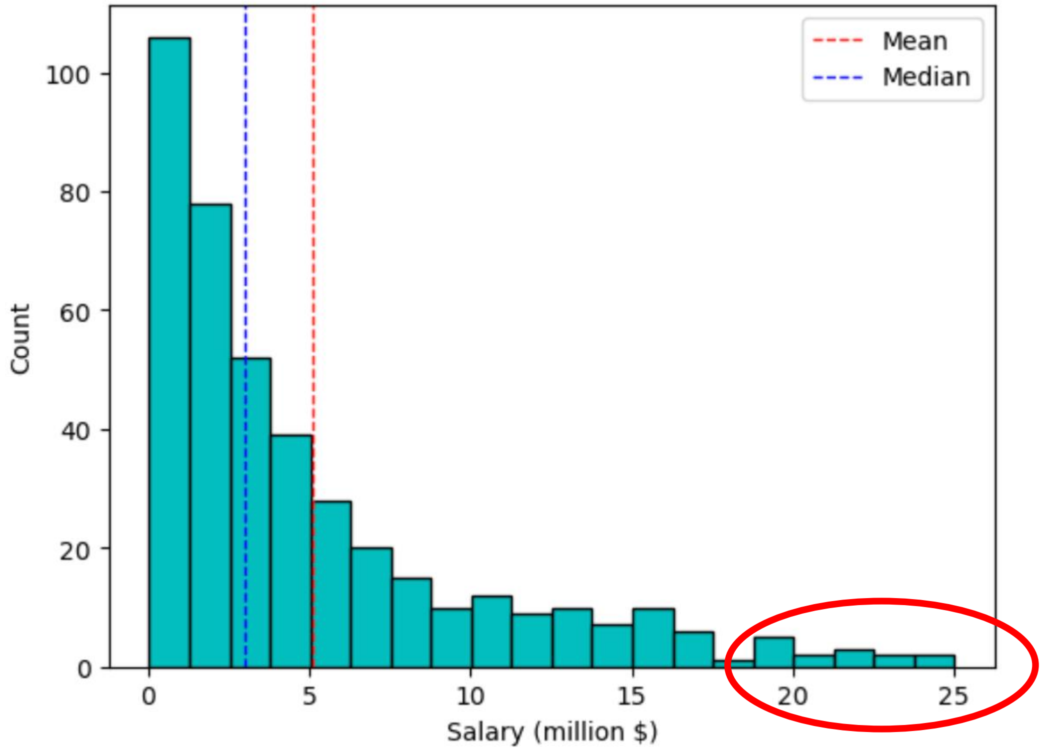
The sum of the heights of the bars on the left is equal to the sum of the heights of the bars on the right




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

Outliers

An **outlier** is an observed value that is notably distinct from the other values in a dataset by being much smaller or larger than the rest of the data.

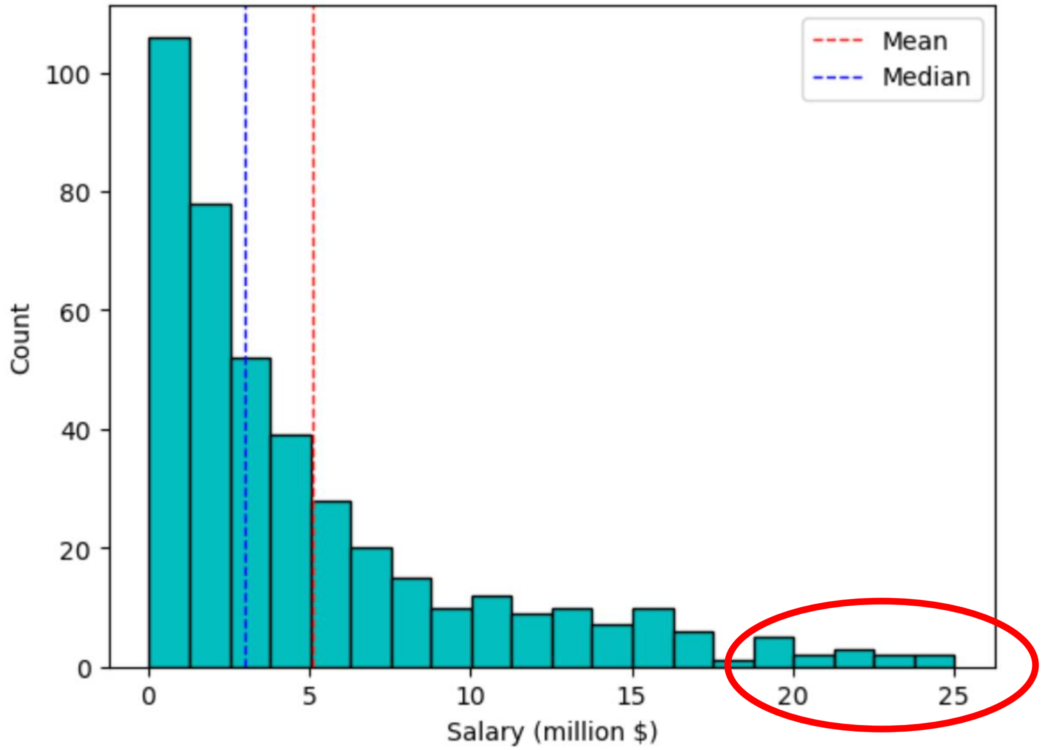


Outliers can potentially have a large influence on the statistics you calculate

One should examine outliers to understand what is causing them

- If there are due to an error, remove them
- Otherwise, need to think about how to treat them
    - Could be interesting phenomenon
    - Could restrict data to a particular range of values
    - Etc.

Outliers' impact on mean and median



The median is *resistant* to outliers

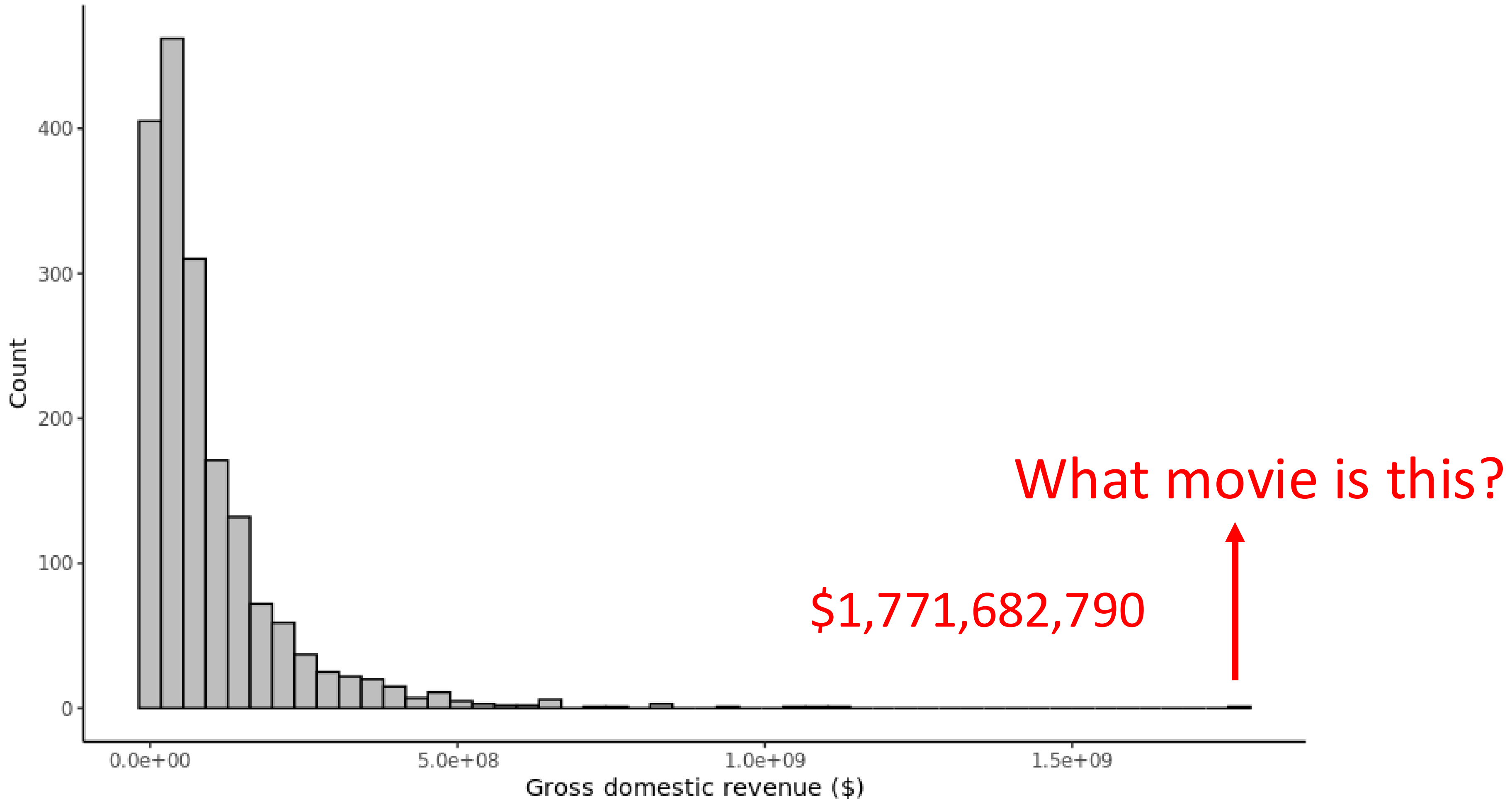
- i.e., not affected much by outliers

The mean is not resistant to outliers

What is the mean and median of the data: 1, 2, 3, 4, 990?

- Mean = 200
- Median = 3

Bechdel outliers

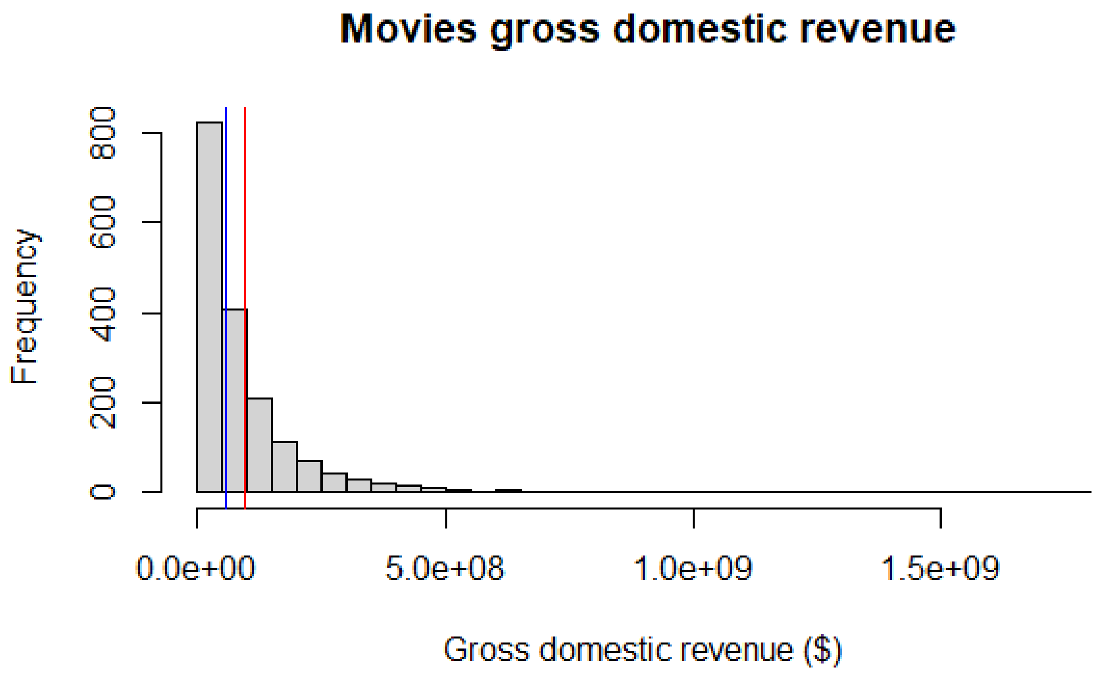



ANY
QUESTIONS?

Measures of spread

Characterizing the spread

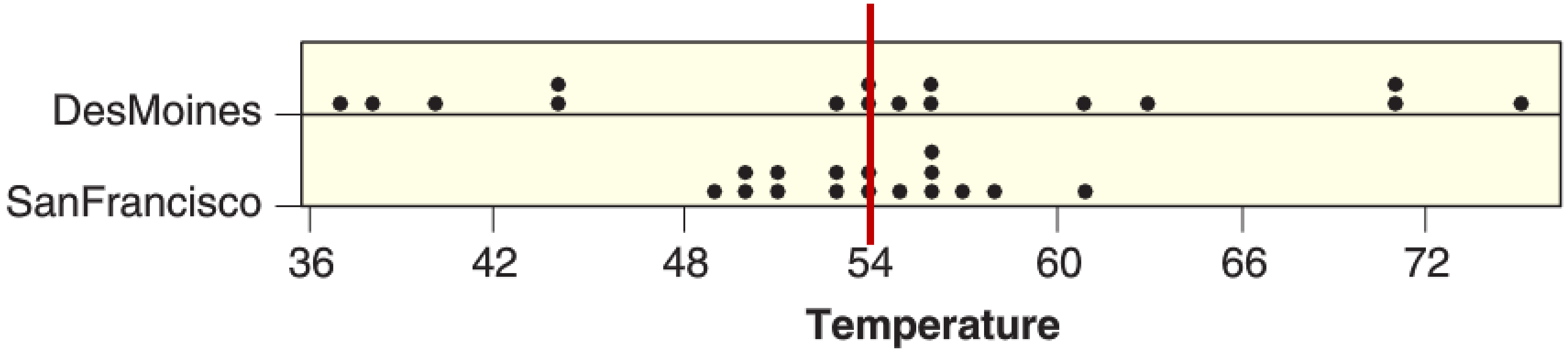
The mean and median are numbers that tell us about the center of a distribution



We can also use numbers to characterize how data is spread

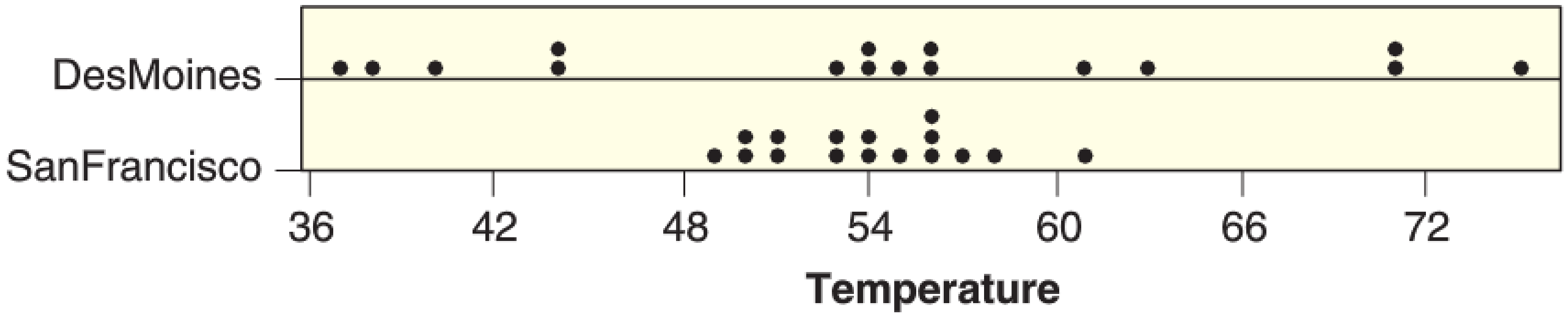
Average monthly temperature:
Des Moines vs. San Francisco

Data measured on April 14th from 1997 to 2010:



Mean temperature (°F): Des Moines = 54.49 San Fran = 54.01

Which has the larger standard deviation?



s_{DM} = 11.73 °F s_{SF} = 3.38 °F

The standard deviation

The standard deviation can be computed using the following formula:

$$s = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \overline{x})^2}$$

Standard deviation measures roughly how far the data are from their average

Example: computing the standard deviation

Suppose we had a sample with n = 4 points:

$x_1 = 8$, $x_2 = 2$, $x_3 = 6$, $x_4 = 4$,

We can compute the mean using the formula:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad = \tfrac{1}{4} \cdot (x_1 + x_2 + x_3 + x_4) \quad = \tfrac{1}{4} \cdot (8 + 2 + 6 + 4)$$

The standard deviation can be computed using the formula:

$$s = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n} (x_i - \overline{x})^2}$$

(remember order of operations!)

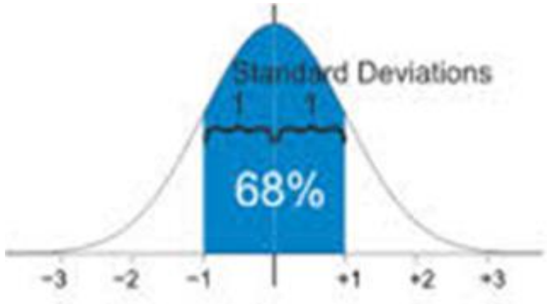
```
statistics.stdev(data_list)
```

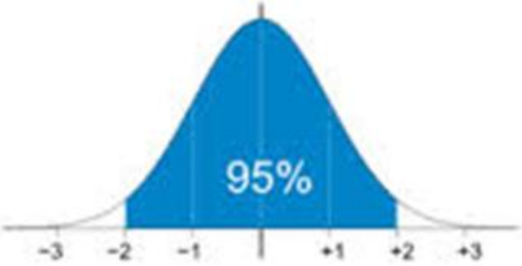
Normally distributed data

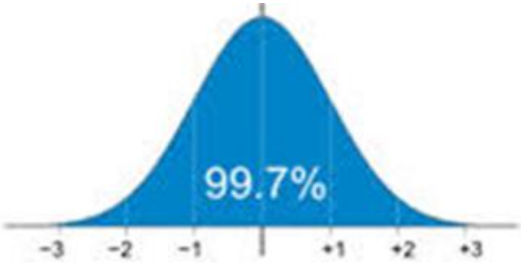
<u>No matter what the shape of the distribution</u>, the bulk of the data are in the range "average ± a few SDs"

If the data is “normally distributed” (bell shaped distribution) than the following holds:

Range	**Proportion**
Average ± 1 SDs	68% of the data
Average ± 2 SDs	95% of the data
Average ± 3 SDs	99.7% of the data







Chebyshev's Inequality

No matter what the shape of the distribution, the bulk of the data are in the range "average ± a few SDs"

Chebyshev's Inequality: No matter what the shape of the distribution, the proportion of values in the range "average ± $z \cdot$ SDs" is at least $1 - 1/z^2$

Range	**Proportion**
Average ± 2 SDs	at least 1 - 1/4 (75%)
Average ± 3 SDs	at least 1 - 1/9 (88.88...%)
Average ± 4 SDs	at least 1 - 1/16 (93.75%)
Average ± 5 SDs	at least 1 - 1/25 (96%)

Let’s briefly explore standard deviations in Jupyter!

Z-scores

Standardized units

Item in the world are often measured on very different scales

How can we create a standard scale to quantify unusual/large/impressive values?

Z-scores measure how many SDs a value is from average:

$$\text{z-score}(x_i) = \frac{x_i - \bar{x}}{s}$$

- Negative z: value below average
- Positive z: value above average
- z = 0: value equal to average

Which Accomplishment is most impressive?

LeBron James is a basketball player who had the following statistics in 2011:

- Field goal percentage (FGPct) = 0.510
- Points scored = 2111
- Assists = 554
- Steals = 124

The summary statistics of the NBA in 2011 are given below

$$\text{z-score}(x_i) = \frac{x_i - \bar{x}}{s}$$

	Mean	Standard Deviation
FGPct	0.464	0.053
Points	994	414
Assists	220	170
Steals	68.2	31.5

Question: Relative to his peers, which statistic is most and least impressive?

Relationships between two quantitative variables

Do movies with larger budgets make more money?

Q: How could we visualize the data to see if this is true?

Scatterplot

A **scatterplot** graphs the relationship between two variables

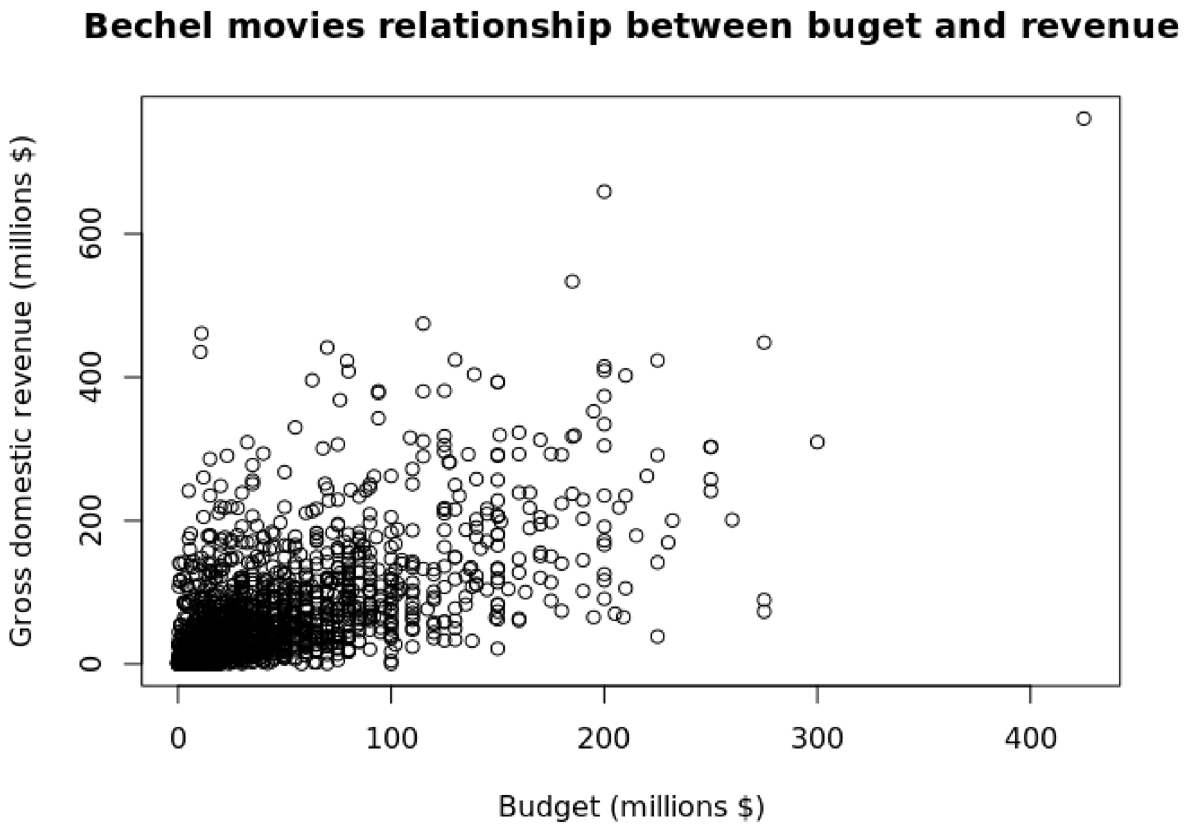
- Each axis represents the value of one variables
- Each point the plot shows the value for the two variables for a single data case

If there is an explanatory and response variable, then the explanatory variable is put on the x-axis and the response variable is put on the y-axis.

Do movies with larger budgets make more money?

Q: If we want to create a scatter plot to address our question, what variables should we use in our plot?

Relationship movie money spent and made



Do movies with larger budgets make more money?

Do most movies make money?

- How could we create a more informative scatter plot of this data?

Matplotlib: `plt.plot(x, y)`

Questions when looking at scatterplots

Do the points show a clear trend?

Does it go upward or downward?

How much scatter around the trend?

Does the trend seem be linear (follow a line) or is it curved?

Are there any outlier points?

Questions when looking at scatterplots

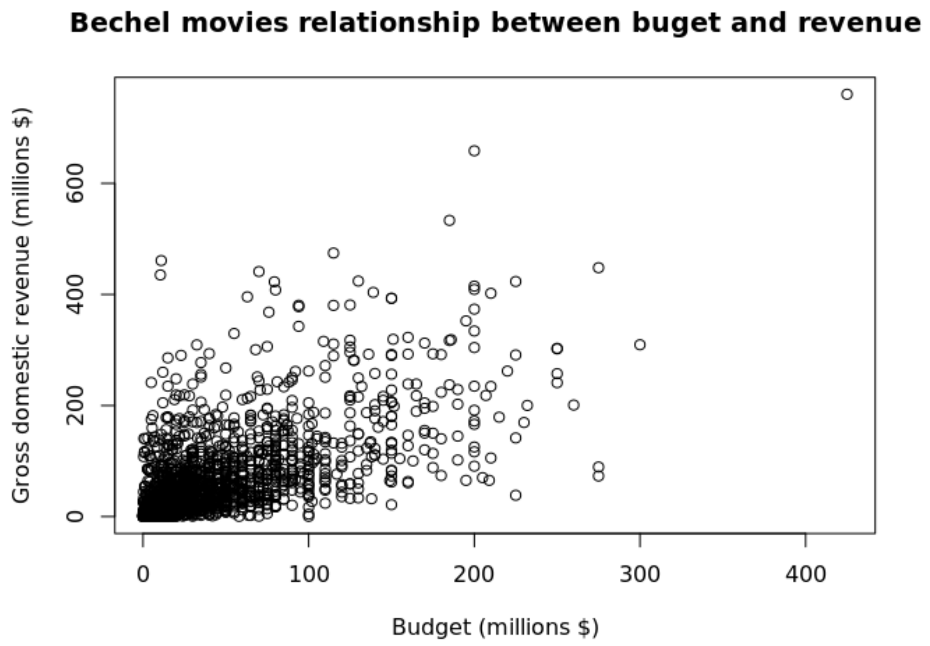
Do the points show a clear trend?

- Does it go upward or downward?
- How much scatter around the trend?

Does the trend seem be linear (follow a line) or is it curved?

Are there any outlier points?

Budget and revenue



Positive, negative, no correlation

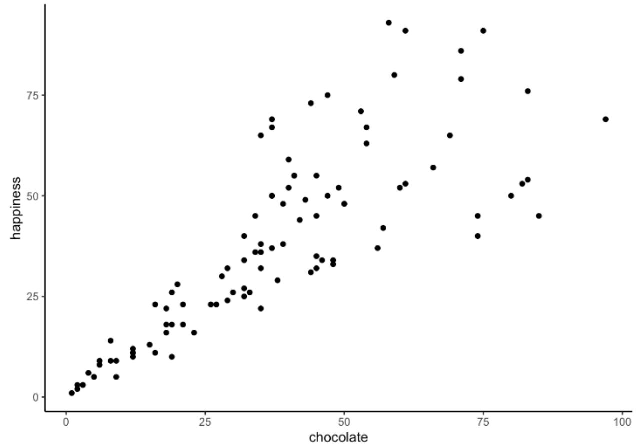
Do the points show a clear trend?

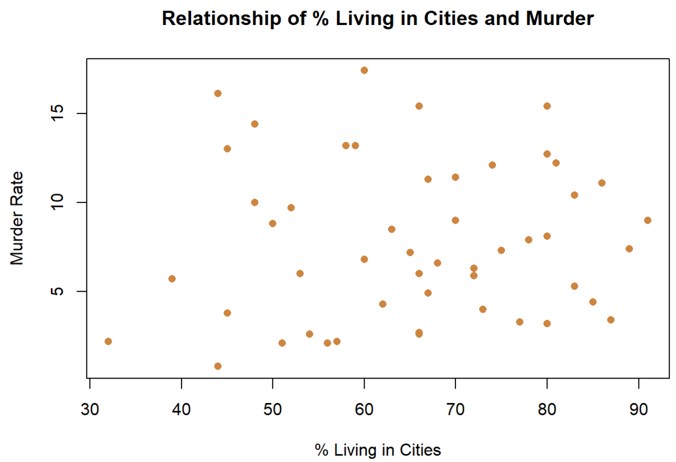
Does it go upward or downward?

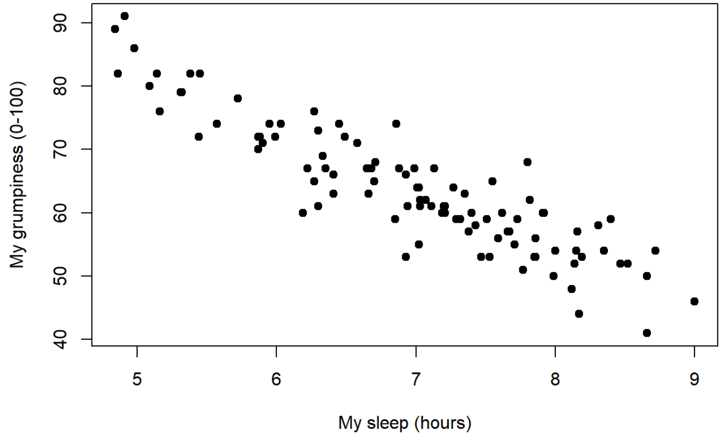
How much scatter around the trend?

Does the trend seem be linear (follow a line) or is it curved?

Are there any outlier points?







The correlation coefficient

The **correlation** is measure of the strength and direction of a <u>linear association</u> between two variables

$$r = \frac{1}{(n-1)} \sum_{i=1}^{n} \left(\frac{x_i - \overline{x}}{s_x} \right) \left(\frac{y_i - \overline{y}}{s_y} \right)$$

```
statistics.correlation(x, y)
```

Properties of the correlation

Correlation as always between -1 and 1: $-1 \leq r \leq 1$

The sign of r indicates the direction of the association

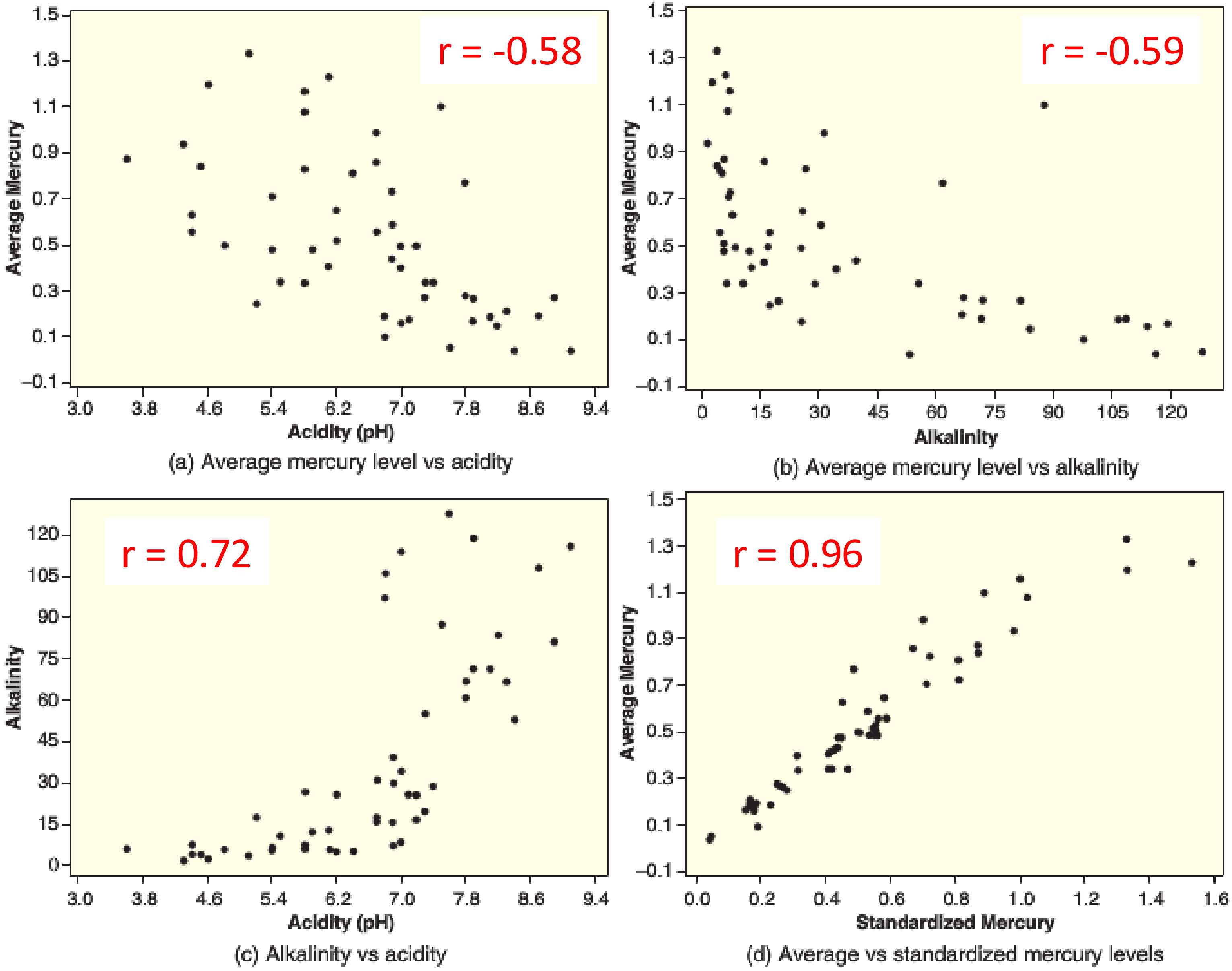
Values close to ± 1 show strong linear relationships, values close to 0 show no linear relationship

Correlation is symmetric: r = cor(x, y) = cor(y, x)

$$r = \frac{1}{(n-1)} \sum_{i=1}^{n} \left(\frac{x_i - \overline{x}}{s_x} \right) \left(\frac{y_i - \overline{y}}{s_y} \right)$$

Florida lakes

Correlation game

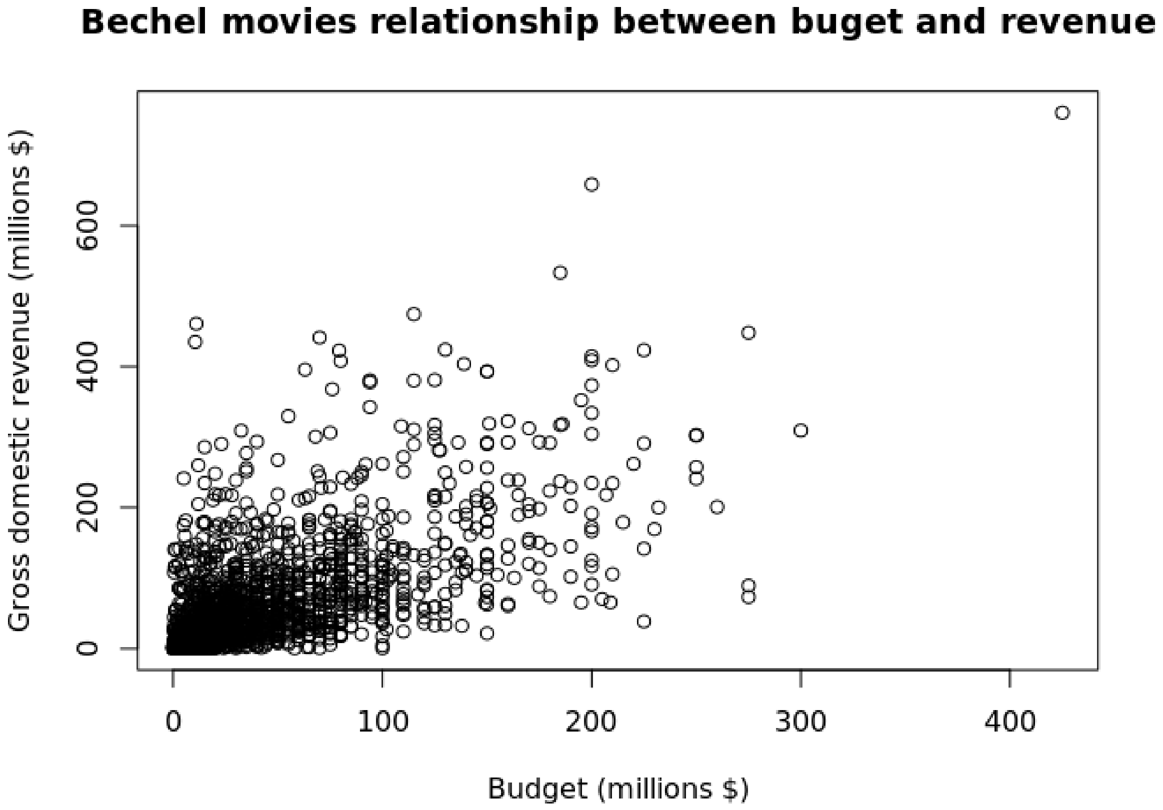


Movie budget and revenue correlation?

The **correlation** is measure of the strength and direction of a linear association between two variables

r = ?

Let's calculate the correlation in Python!



Correlation caution #1

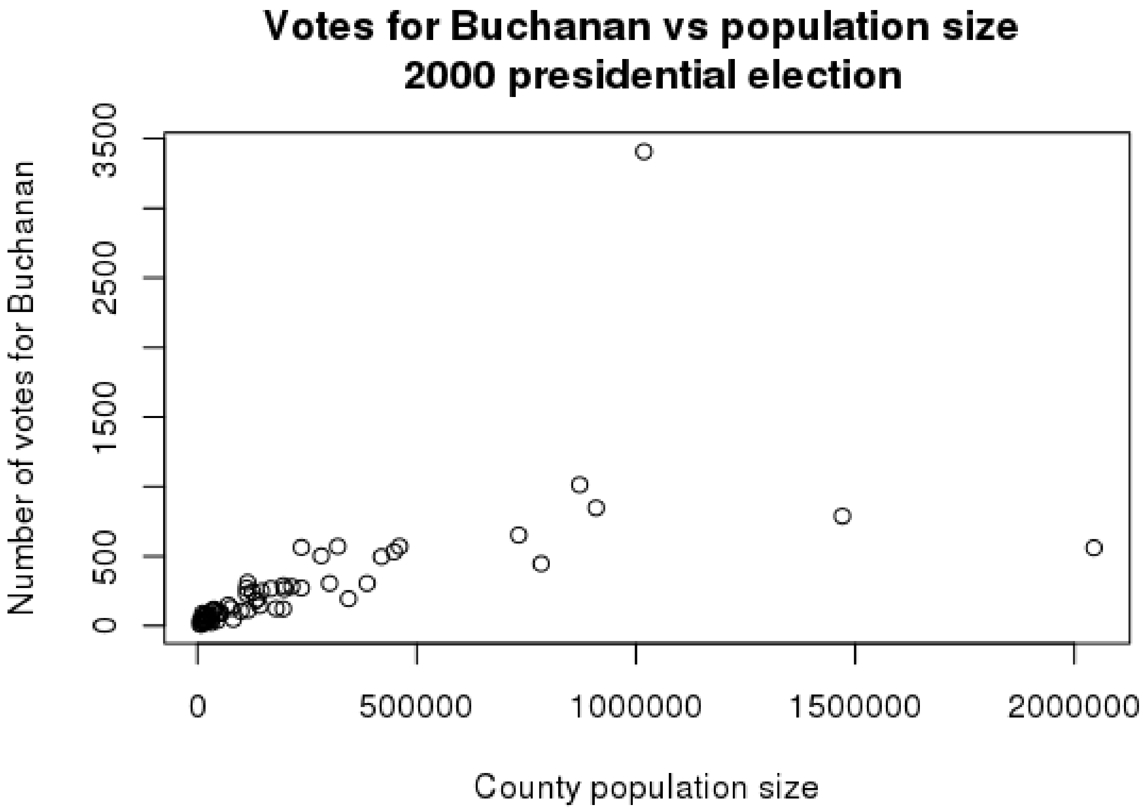
A strong positive or negative correlation does not (necessarily) imply a cause and effect relationship between two variables

Correlation caution #2

A correlation near zero does not (necessarily) mean that two variables are not associated. Correlation only measures the strength of a <u>linear</u> relationship.

Correlation caution #3

Correlation can be heavily influenced by outliers. Always plot your data!



Anscombe’s quartet (r = 0.81)