



# Introduction to Statistics

Data Boot Camp  
Lesson 5.3



# Class Objectives

---

By the end of today's class, you will be able to:



Calculate summary statistics such as mean, median, mode, variance, and standard deviation using Python.



Plot, characterise, and quantify a normally distributed dataset using Python.



Qualitatively and quantitatively identify potential outliers in a dataset.



Differentiate between a sample and a population in regards to a dataset.



Define and quantify correlation between two factors.



Calculate and plot a linear regression in Python.

**Don't worry!**

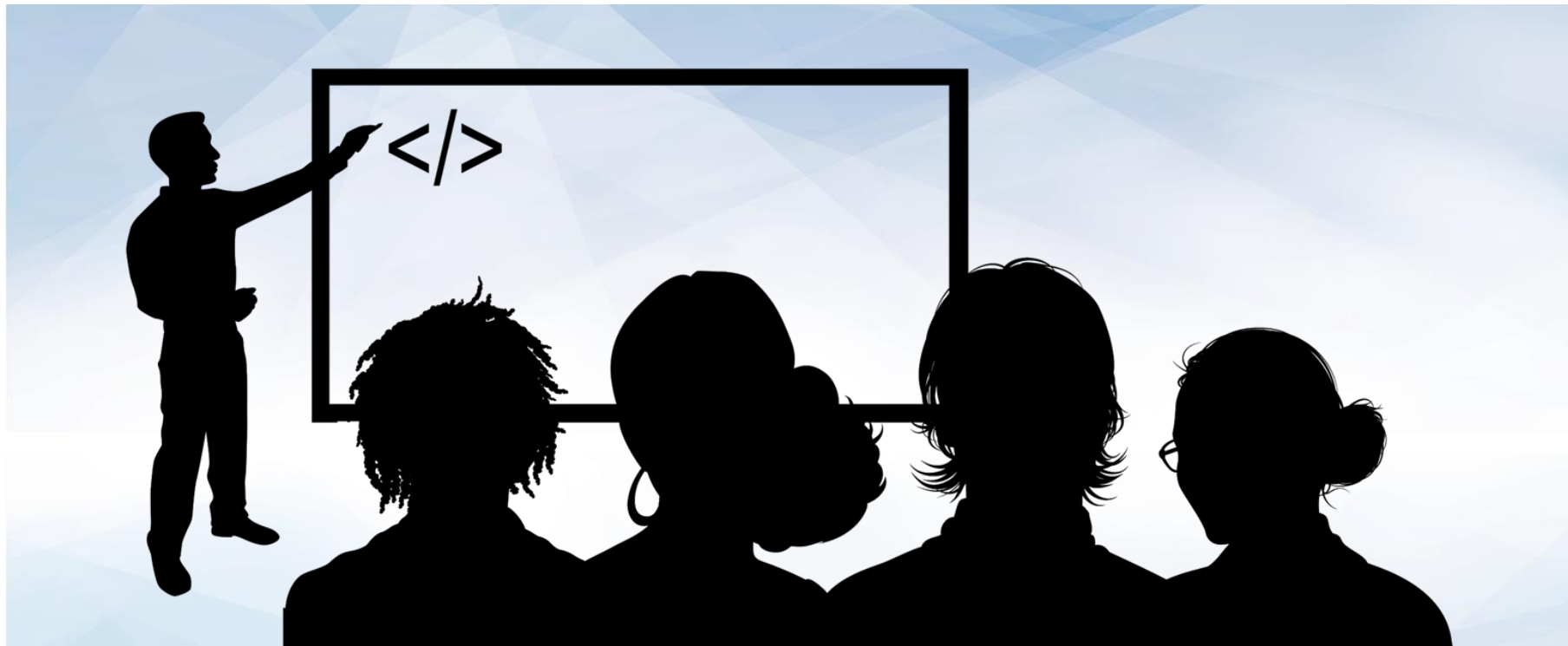
Class will not be painful.



# We Will Build on Concepts You Already Know

---





# Instructor Demonstration

## Summary Statistics in Python



What is a measure of  
**central tendency?**

# Measure of Central Tendency = Centre of a Dataset

---

Three most common measures are **mean**, **median**, and **mode**.

## Mean

**Mean** is the sum of all values divided by the number of elements in a dataset.

## Median

**Median** is the middle value in a sorted dataset.

## Mode

**Mode** is the most frequently occurring value(s) in a dataset.

# Measures of Central Tendency in Python

---

Two packages to remember when calculating statistics are **NumPy** and **SciPy**.

## Mean

Mean is calculated using **NumPy**.

## Median

Median is calculated using **NumPy**.

## Mode

Mode is calculated using **SciPy**.





When new data comes along,  
you must plot it!

# Why Plot Data?

---

01

To determine if the data is normally distributed.

02

To determine if the data is multimodal.

03

To characterise clusters in the dataset.

# What Is Normally Distributed Data?

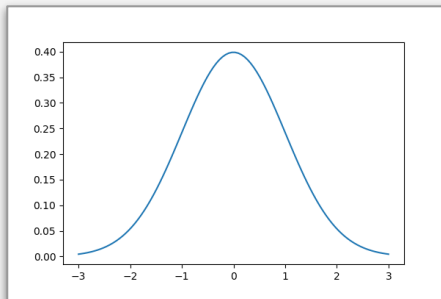
---

01

Measurements in a dataset are obtained independent of one another

02

The distribution of data follows a bell curve shape



03

We can quantitatively test if a dataset is normal using SciPy

```
stats.normaltest()
```



What are **variance** and  
**standard deviation**?

# Variance & Standard Deviation Describe Variability of Data

---



**Variance** is the measurement of how far each value is away from the mean of the dataset.



**Standard deviation** is the square root of variance.



In Python, both variance and standard deviation are calculated using the NumPy module.

# <Time to Code>





# Instructor Demonstration

## Quantiles and Outliers in Python



What are **quantiles**, **quartiles**,  
and **outliers**?



# Quantiles, Quartiles, and Outliers Describe a Dataset

---

01

**Quantiles** divide data into well-defined regions based on a sorted dataset

02

**Quartiles** are a specific type of quantile where a sorted dataset is split into four equal parts

Q1: 25% of the data  
Q2: 50% of the data  
Q3: 75% of the data

03

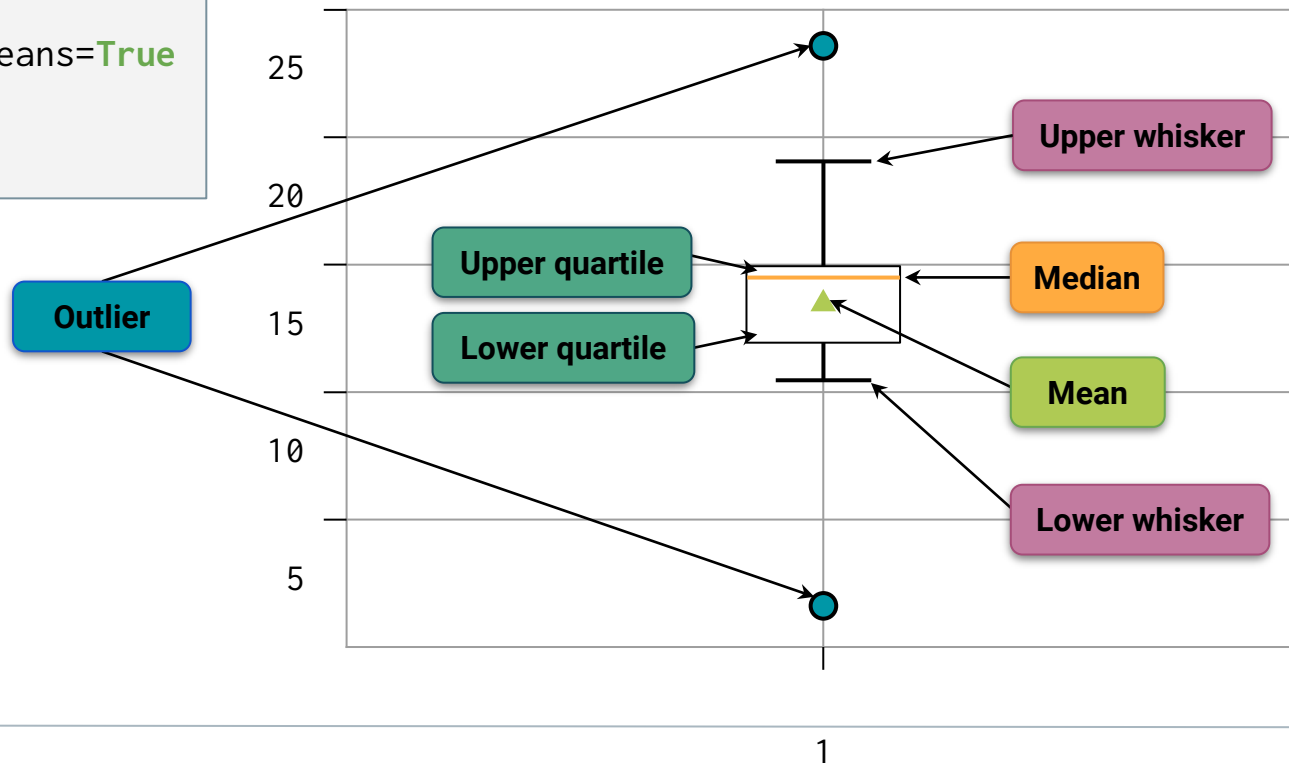
**Outliers** are an extreme value in a dataset that can skew calculations and results

# How to Identify Potential Outliers Qualitatively

Use **box and whisker plots** to visually identify potential outlier data points.

```
# Create box plot
```

```
plt.boxplot(arr, showmeans=True)  
plt.grid()  
plt.show()
```



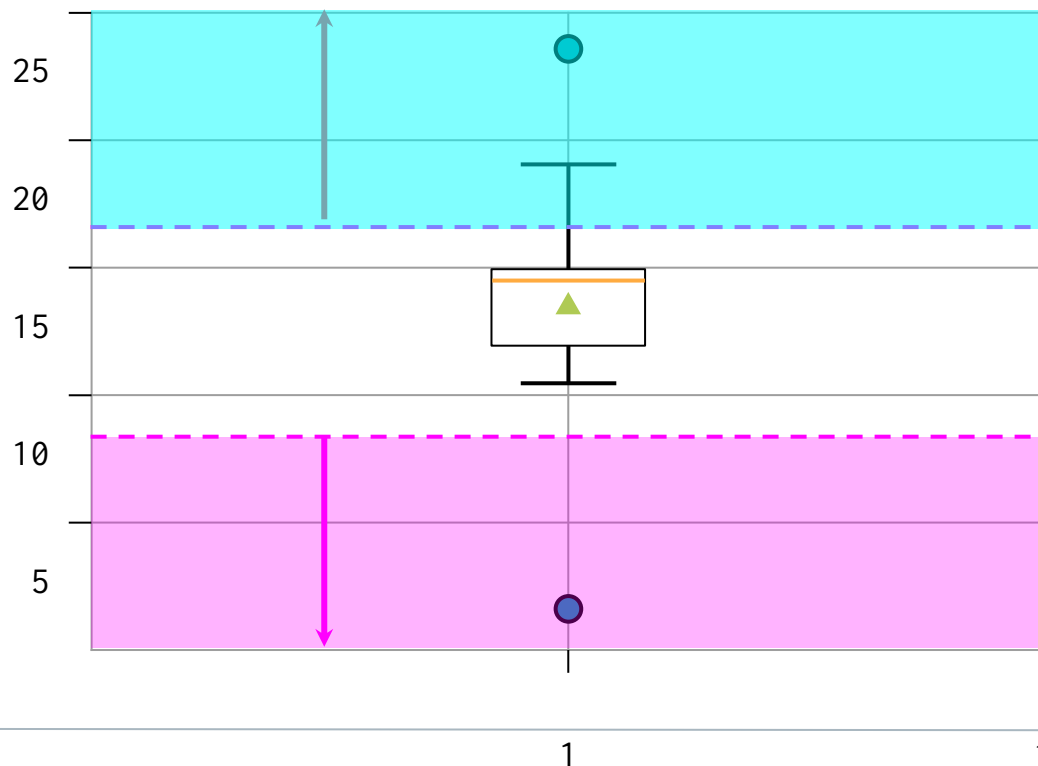
# How to Identify Potential Outliers Qualitatively

Determine the outlier boundaries in a dataset using the '1.5 IQR' rule.

IQR is the interquartile range, or the range between the 1st and 3rd quartiles.

Anything below  $Q1 - 1.5 \text{ IQR}$  could be an outlier.

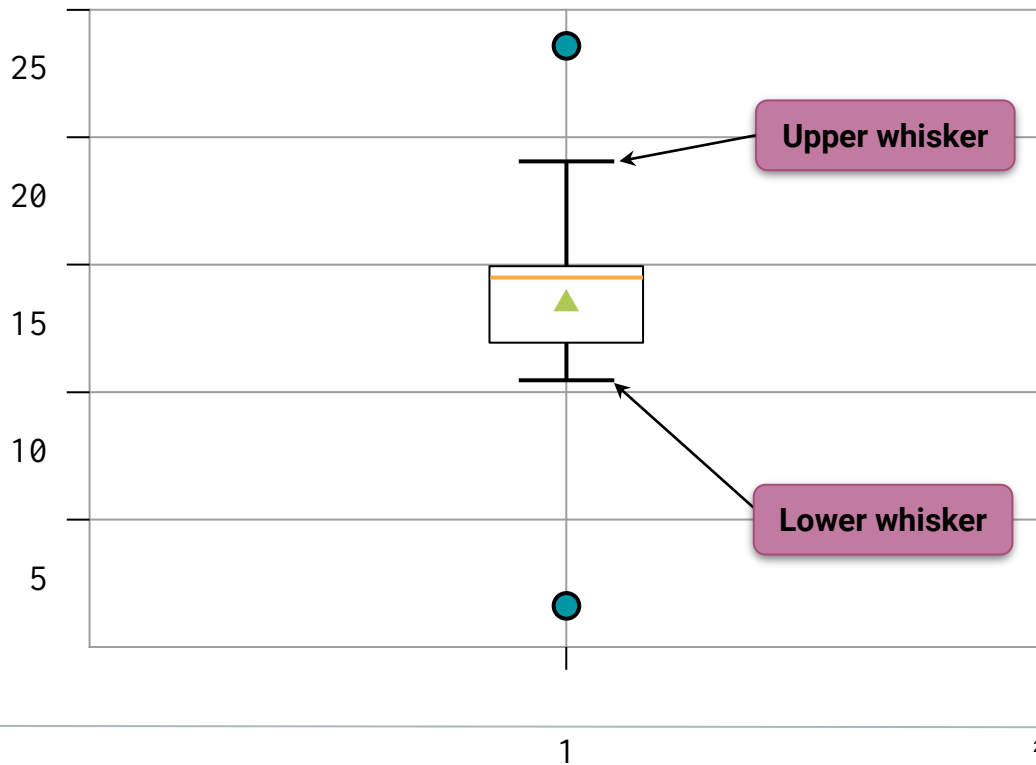
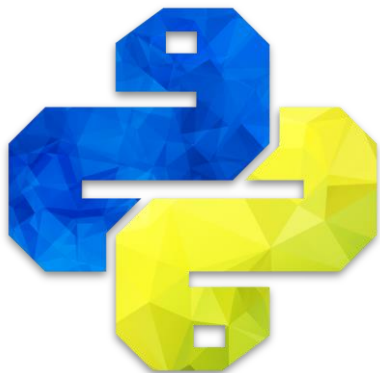
Anything above  $Q3 + 1.5 \text{ IQR}$  could be an outlier.



# How to Identify Potential Outliers in Python Qualitatively

Use Matplotlib's `pyplot.boxplot` function to plot the box and whisker.

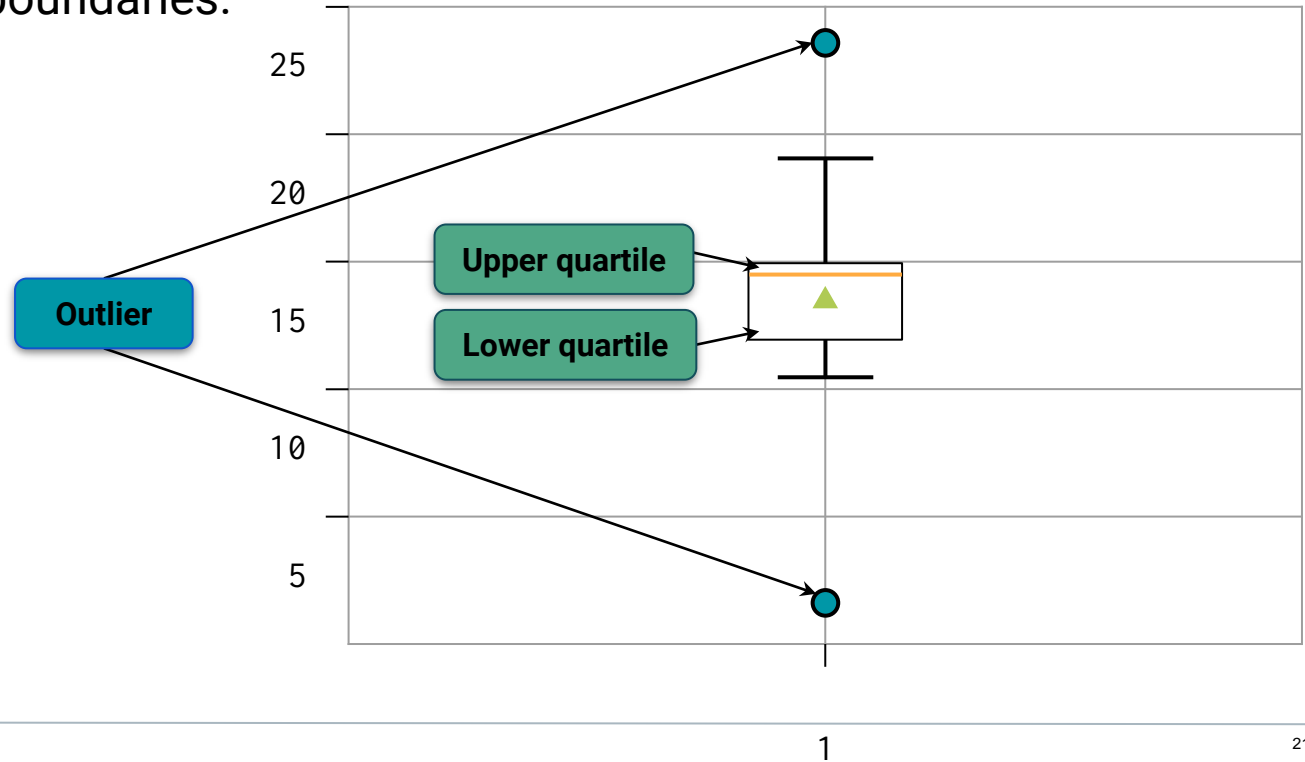
```
# Create box plot  
plt.boxplot(arr, showmeans=True)  
plt.grid()  
plt.show()
```



# How to Identify Potential Outliers in Python Qualitatively

Use Pandas' `series.quantile` function to calculate the quantile.

Calculate the outlier boundaries.



# <Time to Code>





## Activity: Summary Statistics in Python

In this activity, you will be tasked with calculating a number of summary statistics using California housing data.

Activity instructions will be sent via  **slack**

**Suggested Time:**  
20 Minutes



# Instructions: Summary Statistics in Python

---

- Using Pandas, import the California housing dataset from the Resources folder.
  - **File:** `Resources/California_Housing.csv`
- Determine the most appropriate measure of central tendency to describe the population. Calculate this value.
- Is the age of houses in California normally distributed? Use both data visualisation and quantitative measurement to find out.
- Inspect the average occupancy of housing in California and determine if there are potential outliers in the dataset.
  - **Hint:** This dataset is very large.
- If there are potential outliers in the average occupancy, what are the minimum and maximum median housing prices across the outliers?

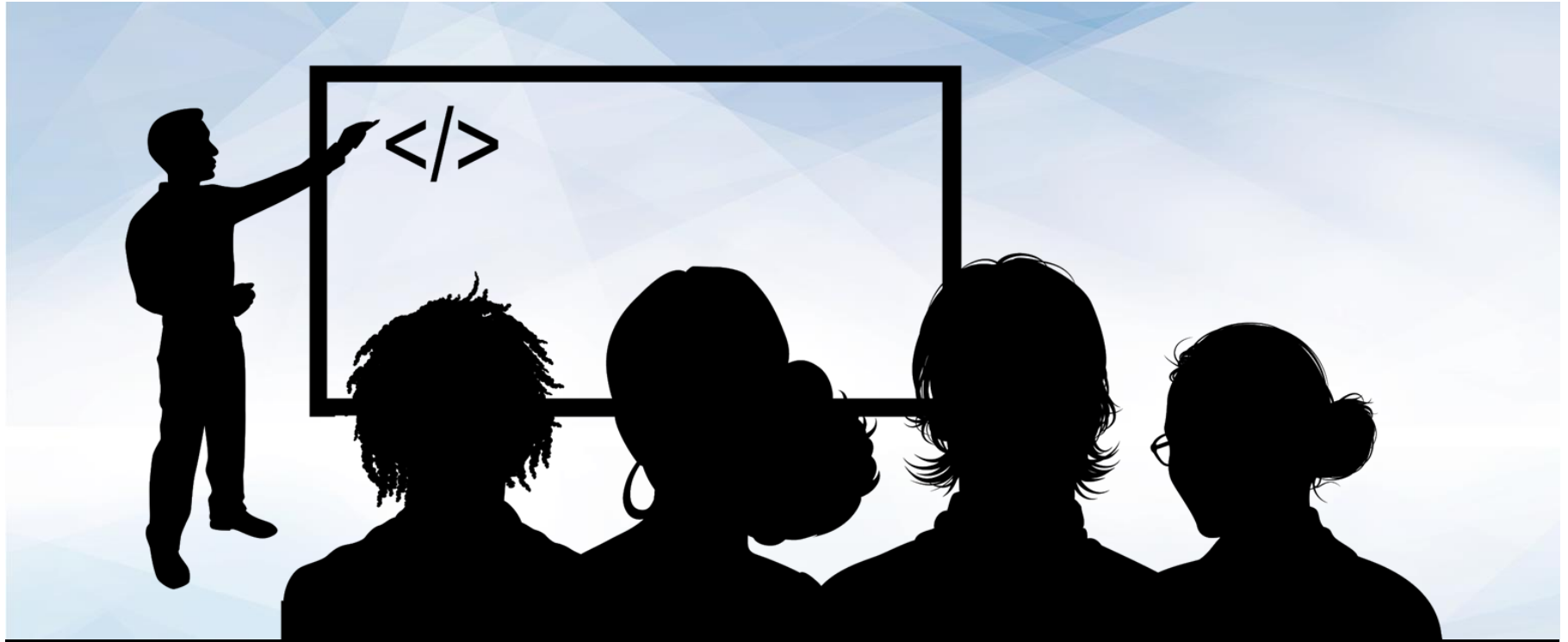
## Bonus

Plot the latitude and longitude of the California housing data using Matplotlib. Colour the data points using the median income of the block. Does any location seem to be an outlier?










# Instructor Demonstration

Sample, Population, and SEM



Let's think about  
the following  
scenario...

# Predicting the City Election

Weeks before Election Day, a local newspaper wants to predict the winner of the mayoral election. The newspaper will poll voters for their intended candidate. Consider the following:

- It would be prohibitively expensive to poll all voters.
- It is logistically impossible to know who will actually go out to vote on Election Day.
- Therefore, the newspaper must predict the outcome of the election using data from a subset of the population.





This is a **population dataset**  
versus a **sample dataset**.

# Population Dataset vs. Sample Dataset

---

## Population Dataset

- Dataset containing all possible elements of an experiment or study.
- In statistics, 'population' does not mean 'people'.
- Any complete set of data is a population dataset.

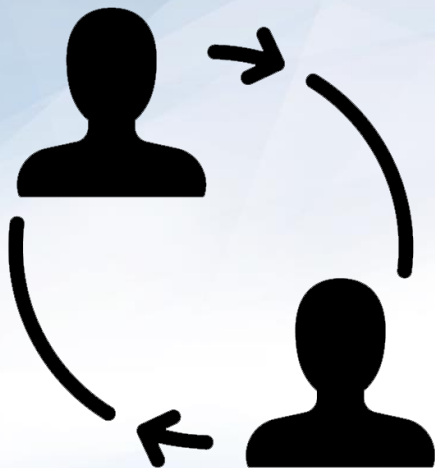
vs.

## Sample Dataset

- A subset of population data.
- A sample dataset can be selected randomly from the population or selected with bias.

# <Time to Code>





## Partner Activity: SEM and Error Bars

In this activity, you will work with a partner to characterise sample data from a Boston housing dataset. Be sure to compare your calculated values as you progress through the activity.

Take your time—this is an important statistical concept.

Activity instructions will be sent via  **slack**

**Suggested Time:**  
25 minutes





# Instructions: SEM and Error Bars

---

- Open `samples.ipynb` in the activity folder.
- Execute the starter code to import the Boston housing dataset from scikit-learn.
- Using Pandas, create a sample set of median housing prices. Be sure to create samples of size 20.
- Calculate the means and standard error for each sample.
- Create a plot displaying the means for each sample, with the standard error as error bars.
- Calculate the range of SEM values across the sample set.
- Determine which sample has the lowest standard error value.
- Compare this sample's mean to the population's mean.
- Rerun your sampling code a few times to generate new sample sets. Try changing the sample size and then rerun the sampling code.
- Discuss with your partner what changes you observe when sample size changes.

**Suggested Time:** 25 minutes





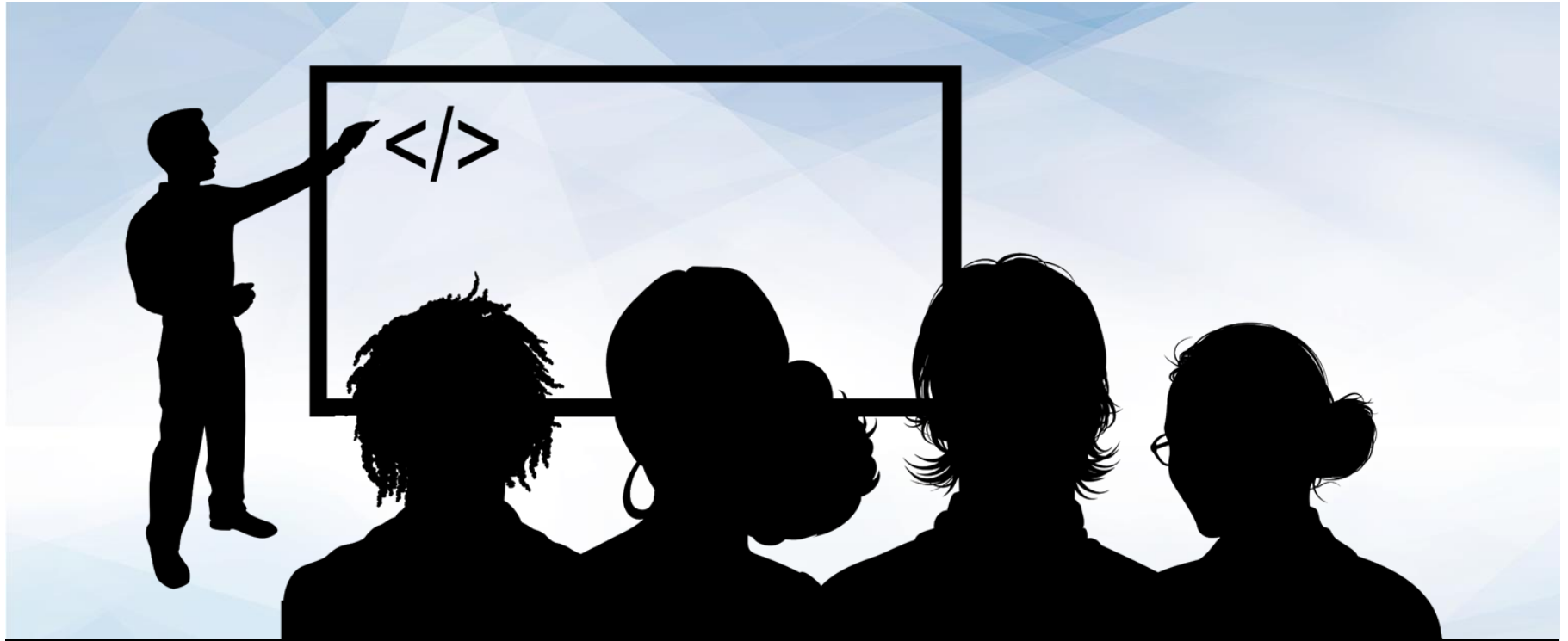


Countdown timer

**40:00**

(with alarm)





# Instructor Demonstration

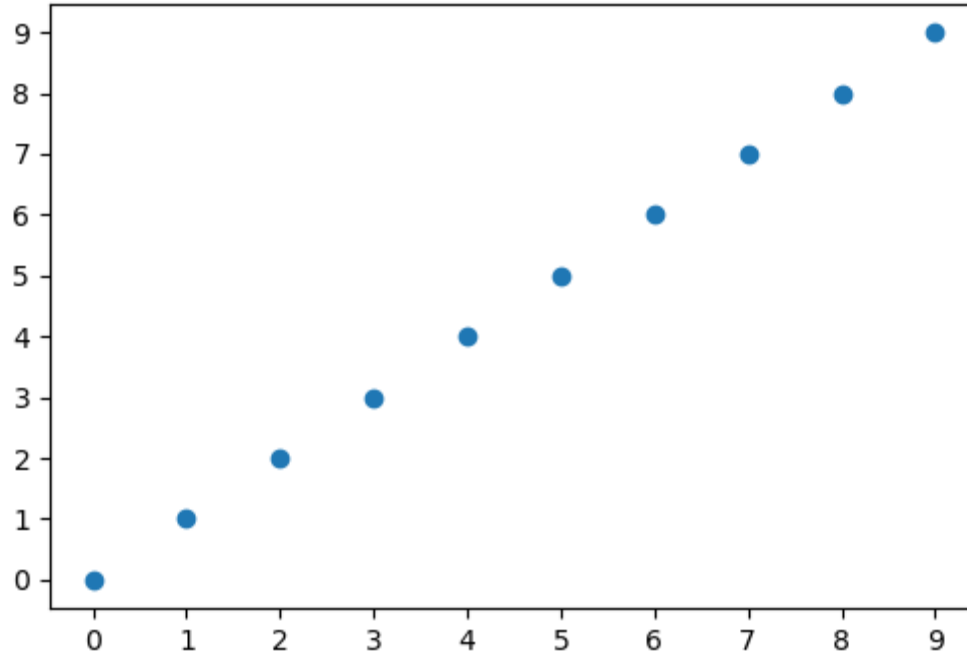
## Correlation Conundrum



**Correlation** describes the question, 'Is there a relationship between A and B?'

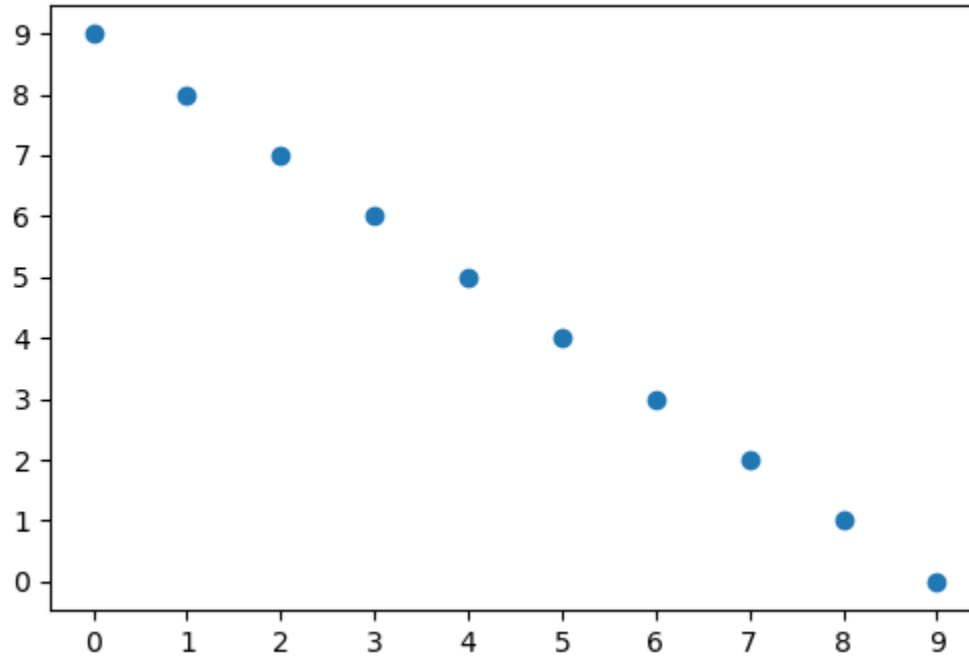
# Positive Correlation

---



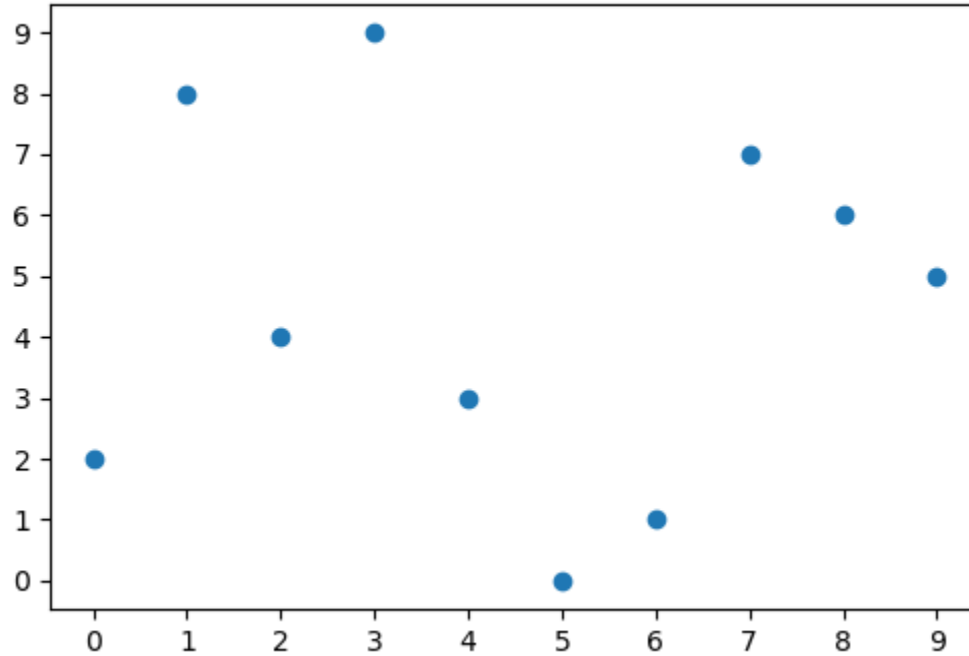
# Negative Correlation

---



# No Correlation

---





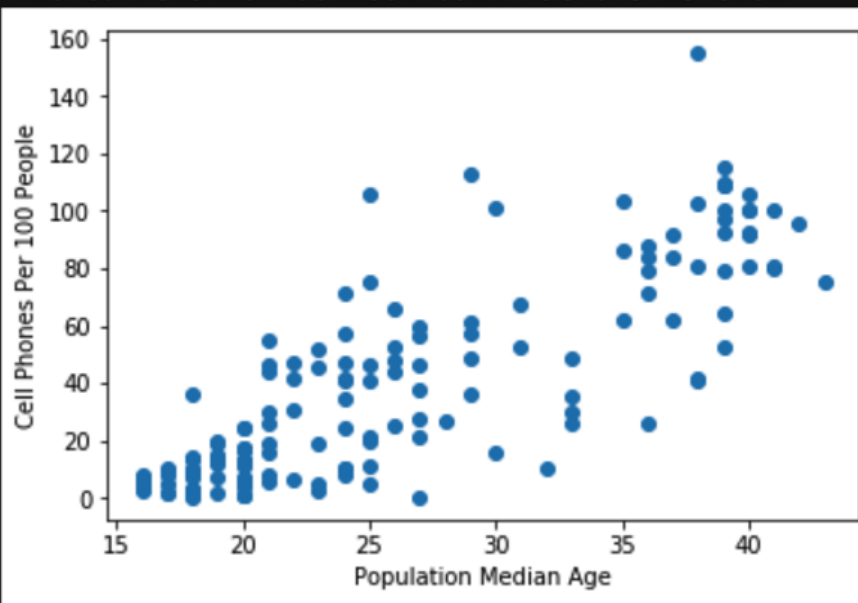
# Pearson's Correlation Coefficient

In statistics, we quantify correlation using Pearson's  $r$ .

Pearson's correlation coefficient describes the variability between two factors, denoted by the variable  $r$ .

-1	indicates perfect negative correlation
1	indicates perfect positive correlation
0	indicates no correlation

The correlation between both factors is 0.82



Real-world data is never perfect.

# <Time to Code>





## Activity: Correlation Conquerors

In this activity, you will be looking at different properties of wine to determine if wine characteristics are correlated.

Activity instructions will be sent via



**Suggested Time:**  
10 Minutes



# Instructions: Correlation Conquerors

---

- Open `correlations.ipynb` in the activity folder and execute the starter code.
- Using the dataset, plot the factors malic acid versus flavonoids on a scatter plot.
  - Is this relationship positively correlated, negatively correlated, or not correlated?
  - How strong is the correlation?
- Calculate the Pearson's correlation coefficient for malic acid versus flavonoids.
  - Compare the correlation coefficient to the 'Strength of Correlation' table.
  - Was your prediction correct?

Absolute Value of r	Strength of Correlation
$r < 0.3$	None or very weak
$0.3 \leq r < 0.5$	Weak
$0.5 \leq r < 0.7$	Moderate
$r \geq 0.7$	Strong







# Instructor Demonstration

## Fits and Regression



What is the equation of a line?



The equation of a line is:

$$y = mx + b$$



# The Equation of a Line

---

$$y = mx + b$$

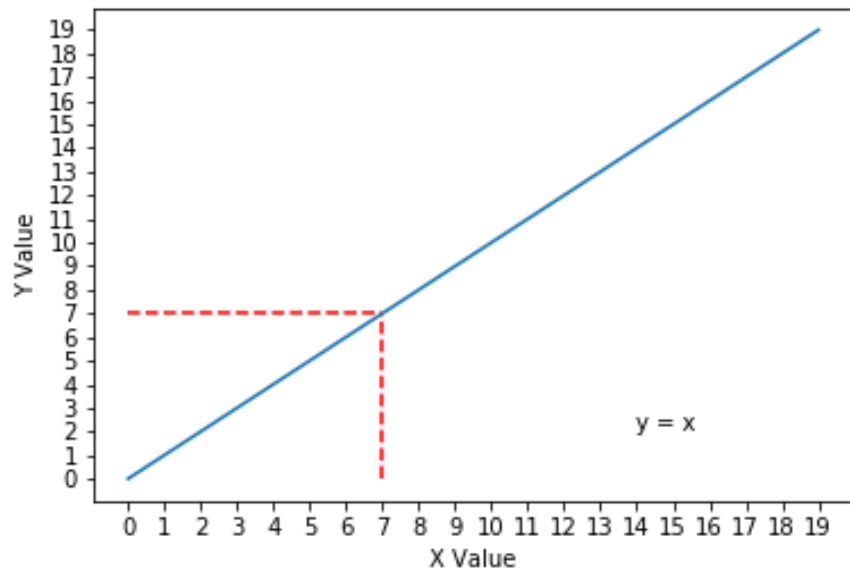
Diagram illustrating the components of the equation of a line:

- $y$ : Dependent variable (indicated by a red arrow pointing up from the label "Dependent variable")
- $m$ : Slope (indicated by a red arrow pointing up from the label "Slope")
- $x$ : Independent variable (indicated by a red arrow pointing up from the label "Independent variable")
- $b$ : y-intercept (indicated by a red arrow pointing up from the label "y-intercept")

# The Equation of a Line Determines $y$ Values, Given $x$

In this example:

- Slope = 1
- $y$ -intercept = 0
- Whatever  $x$  is, the value of  $y$  is the same.

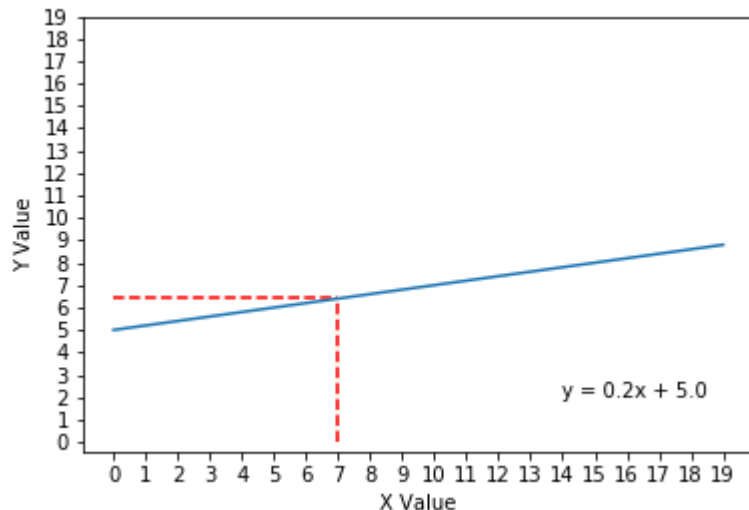


# The Equation of a Line Determines $y$ Values, Given $x$

---

In this example:

- Slope = 0.2
- $y$ -intercept = +5
- If  $x = 7$ , then  $y = 6.4$

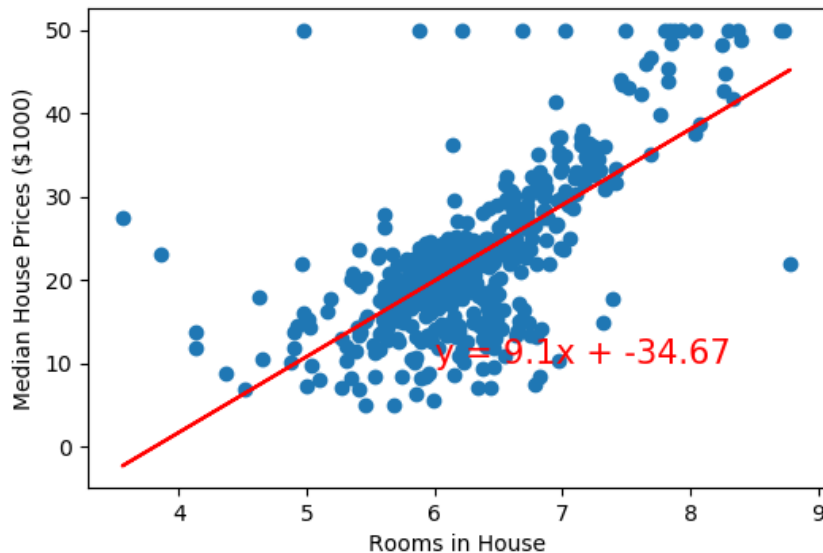


# Linear Regression Fits the Equation of a Line to Real-World Data

---

## Linear regression:

- Predicts the values of factor B, given values from factor A.
- Estimates where data points that were not measured might end up if more data was collected.
- Is used to predict housing prices, stock market, weather, etc.



# <Time to Code>





## Activity: Fits and Regressions

In this activity, you will be predicting the crime rates in 2019, using linear regression models.

Activity instructions will be sent via  **slack**

**Suggested Time:**  
20 Minutes



# Fits And Regression

---

- Open `crime.ipynb` and execute the starter code.
- Generate a scatter plot with Matplotlib using the year as the independent ( $x$ ) variable and violent crime rate as the dependent ( $y$ ) variable.
- Use `stats.linregress` to perform a linear regression with the year as the independent variable ( $x$ ) and violent crime rate as the dependent variable ( $y$ ).
- Use the information returned by `stats.linregress` to create the equation of a line from the model.
- Calculate the predicted violent crime rate of the linear model using the year as the  $x$ -value.
- Plot the linear model of year versus violent crime rate on top of your scatter plot.
- Repeat the process of generating a scatter plot, calculating the linear regression model, and plotting the regression line over the scatter plot for year versus murder rate, and year versus aggravated assault.

## Bonus

Use `pyplot.subplots` from Matplotlib to create a new figure that displays all three pairs of variables on the same plot.



