Unit 2

Basic Mathematics

Basic Statistics

- **Basic Mathematics**:
- Basic symbols and terminology - Vectors and matrices - Arithmetic symbols - Summation –
- Proportional –
- Dot product –
- Graphs - Logarithms/exponents - Set theory –
- Linear algebra –
- Matrix multiplication

- **Basic Statistics:**
- Obtaining data –
- Observational - Experimental –
- Sampling data –
- Probability sampling - Random sampling –
- Unequal probability sampling –
- Measuring statistics - Measures of center - Measures of variation - Measures of relative standing - The Empirical rule

Mathematics as a discipline

- Mathematics, as a science, is one of the oldest known forms of logical thinking by mankind.
- Since ancient Mesopotamia and likely before (3,000 BCE), humans have been relying on arithmetic and more challenging forms of math to answer life's biggest questions.

Vectors and Matrices

- A vector is defined as an object with both magnitude and direction.
- This definition, however, is a bit complicated for our use. For our purpose, a vector is simply a 1-dimensional array representing a series of numbers.
- Put in another way, a vector is a list of numbers. It is generally represented using an arrow or bold font, as shown: $\vec{x}$ or $\boldsymbol{x}$

- Vectors are broken into components, which are individual members of the vector.
- We use index notations to denote the element that we are referring to, as illustrated:
- $$\text{If} \quad \vec{x} = \begin{pmatrix} 3 \\ 6 \\ 8 \end{pmatrix} \quad \text{then } x_1 = 3$$

- In Python, we can represent arrays in many ways. We could simply use a Python list to represent the preceding array:
- x = [3, 6, 8]

- However, it is better to use the numpy array type to represent arrays, as shown, because it gives us much more utility when performing vector operations:

```
import numpy as np
x = np.array([3, 6, 8])
```

- Regardless of the Python representation, vectors give us a simple way of storing multiple dimensions of a single data point/observation.
- Consider that we measure the average satisfaction rating (0-100) of employees for three departments of a company as being 57 for HR, 89 for engineering, and 94 for management. We can represent this as a vector with the following formula:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 57 \\ 89 \\ 94 \end{pmatrix}$$

- This vector holds three different bits of information about our data. This is the perfect use of a vector in data science.

- We can extend our notion of an array to move beyond a single dimension and represent data in multiple dimensions.

Matrix

- A matrix is a 2-dimensional representation of arrays of numbers.
- Matrices (plural) have two main characteristics that we need to be aware of.
- The dimension of the matrix, denoted as n x m (n by m), tells us that the matrix has n rows and m columns.
- Matrices are generally denoted using a capital, bold-faced letter, such as X. Consider the following example:

  3 4
- 8 5

  5 9

- This is a 3 x 2 (3 by 2) matrix because it has three rows and two columns.
- The matrix is our generalization of the Pandas Dataframe.

- It is arguably one of the most important mathematical objects in our toolkit. It is used to hold organized information, in our case, data.
- Revisiting our previous example, let's say we have three offices in different locations,each with the same three departments: HR, engineering, and management.
- We could make three different vectors, each holding a different office's satisfaction scores, as shown:

$$x = \begin{pmatrix} 57 \\ 89 \\ 94 \end{pmatrix}, y = \begin{pmatrix} 67 \\ 87 \\ 84 \end{pmatrix}, z = \begin{pmatrix} 65 \\ 98 \\ 60 \end{pmatrix}$$

Arithmetic Symbols: Summation

- The uppercase sigma $\sum$ symbol is a universal symbol for addition.
- Whatever is to the right of the sigma symbol is usually something iterable, meaning that we can go over it one by one (for example, a vector).
- For example, let's create the representation of a vector:
- $X = [1, 2, 3, 4, 5]$
- To find the sum of the content, we can use the following formula:
- $\sum X_i = 15$

- In Python, we can use the following formula:
- sum(x) # == 15
- For example, the formula for calculating the mean of a series of numbers is quite common.
- If we have a vector (x) of length n, the mean of the vector can be calculated as follows:

$$mean = \frac{1}{n} \sum x_i$$

Proportional

- The lowercase alpha symbol α represents values that are proportional to each other.
- This means that as one value changes, so does the other.
- The direction in which the values move depends on how the values are proportional. Values can either vary directly or indirectly.
- If values vary <u>directly</u>, they both move in the same direction (as one goes up, so does the other).
- If they vary <u>indirectly</u>, they move in opposite directions (if one goes down, the other goes up).
- Consider the following examples:

• The sales of a company vary directly with the number of customers. This can be written as *Sales α Customers*

• Gas prices vary (usually) indirectly with oil availability, meaning that as the availability of oil goes down (it's more scarce), gas prices will go up. This can be denoted as *Gas α 1 / Oil Availability* .

Dot product

- The dot product is an operator like addition and multiplication. It is used to combine two vectors, as shown:

$$\begin{pmatrix} 3 \\ 7 \end{pmatrix} \cdot \begin{pmatrix} 9 \\ 5 \end{pmatrix} = 3 * 9 + 7 * 5 = 62$$

- So, what does this mean? Let's say we have a vector that represents a customer's sentiments towards three genres of movies—comedy, romantic, and action.
- Consider that, on a scale of 1-5, a customer loves comedies, hates romantic movies, and is alright with action movies.

- We might represent this as follows: $\begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix}$
- Here:
- • 5 denotes the love for comedies,
- • 1 is the hatred for romantic
- • 3 is the indifference of action
- Now, let's assume that we have two new movies, one of which is a romantic comedy and the other is a funny action movie.
- The movies would hav ctor of qualities, as shown: $m_1 = \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix}$ and $m_2 = \begin{pmatrix} 5 \\ 1 \\ 5 \end{pmatrix}$

- Here,
- *m1*, is our romantic comedy and
- *m2*, is our funny action movie.
- In order to make a recommendation, we will apply the dot product between the customer's preferences for each movie. The higher value will win and, therefore, will be recommended to the user.
- Let's compute the recommendation score for each movie. For movie 1, we want to compute:

Customer: M_1

$$\begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix} = \begin{array}{ll} (5.4) \longrightarrow & \text{user loves comedies and this move is funny} \\ + & \\ (1.5) \longrightarrow & \text{user hates romance but this move is romantic} \\ + & \\ (3.1) \longrightarrow & \text{user doesn't mind action and the move is not action packed} \\ \hline 28 & \end{array}$$

- The answer we obtain is 28, but what does this number mean? On what scale is it?
- Well, the best score anyone can ever get is when all values are 5, making the outcome as follows:

$$\begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix} . \begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix} = 5^2 + 5^2 + 5^2 = 75$$

- The lowest possible score is when all values are 1, as shown:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 1^2 + 1^2 + 1^2 = 3$$

- So, we must think about 28 on a scale from 3-75. To do this, imagine a number line from 3 to 75 and where 28 would be on it. This is illustrated as follows

- Not that far. Let's try for movie 2:

$$\begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 1 \\ 5 \end{pmatrix} = (5*5)+(1*1)+(3*5) = 41$$

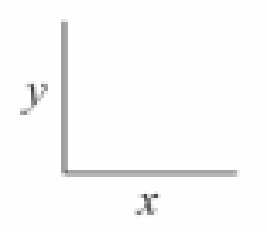
- This is higher than 28! Putting this number on the same timeline as before, we can also visually observe that it is a much better score, as shown:

- So, between movie 1 and movie 2, we will definitely recommend movie 2 to our user.
- This is, in essence, how most movie prediction engines work.

- They build a customer profile, which is represented as a vector.
- They then take a vector representation of each movie they have to offer, combine them with the customer profile (perhaps with a dot product), and make recommendations from there.
- Of course, most companies must do this on a much larger scale, which is where a particular field of mathematics, called linear algebra, can be very useful.

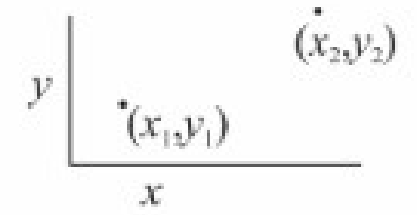
Graphs

- No doubt you have encountered dozens, if not hundreds, of graphs in your life so far. I'd like to mostly talk about conventions with regard to graphs and notations.



- This is a basic Cartesian graph (x and y coordinate). The x and y notation are very standard but sometimes do not entirely explain the big picture.
- We sometimes refer to the x variable as being the independent variable and the y as the dependent variable.
- This is because when we write functions, we tend to speak about them as being y is a function of x, meaning that the value of y is dependent on the value of x.
- This is what a graph is trying to show.

- Suppose we have two points on a graph, as shown:



- We refer to the points as (x_1, y_1) and (x_2, y_2)
- The slope between these two points is defined as follows:
- Slope = m = $\frac{y2-y1}{x2-x1}$
- You have probably seen this formula before, but it is worth mentioning if not for its significance.
- <u>The slope defines the rate of change between the two points.</u>
- Rates of change can be very important in data science, specifically in areas involving differential equations and calculus.

- Rates of change are a way of representing how variables move together and to what degree.
- Consider that we are modeling the temperature of your coffee in relation to the time that it has been sitting out.
- Perhaps we have a rate of change as follows:
- $\frac{-2\ degrees\ F}{1\ minute}$
- This rate of change is telling us that for every single minute, our coffee's temperature is dropping by two degrees Fahrenheit.

Logarithms/exponents

- An exponent tells you how many times you have to multiplv a number to itself, as illustrated:

exponent

$$2^4 = 2 \cdot 2 \cdot 2 \cdot 2 = 16$$

base

- A logarithm is the number that answers the question: "what exponent gets me from the base to this other number?" This can be denoted as follows:

$$\log_2(16) = 4$$

base logarithm

- If these two concepts seem similar, then you are correct! Exponents and logarithms are heavily related.
- In fact, the words exponent and logarithm actually mean the same thing!
- A logarithm is an exponent. The preceding two equations are actually two versions of the same thing.
- The basic idea is that 2 times 2 times 2 times 2 is 16.
- The following is a depiction of how we can use both versions to say the same thing.
- Note how I use arrows to move from the log formula to the exponent formula:

$$\log_2(16) = 4 \leftrightarrow 2^4 = 16$$

- Consider the following examples:
- $\text{Log}_3\ 81 = 4$ because $3^4 = 81$
- $\text{Log}_5\ 125 = 3$ because $5^3 = 125$
- Note something interesting, if we rewrite the first equation to be:
- $$\text{Log}_3\ 81 = 4$$
- We then replace 81 with the equivalent statement, 3^4, as follows:
- $$\text{Log}_3\ 3^4 = 4$$
- Exponents and logarithms are most important when dealing with growth.
- More often than not, if some quantity is growing (or declining in growth), an exponent/logarithm can help model this behavior.

Set Theory

- The set theory involves mathematical operations at a set level.
- It is sometimes thought of as a basic fundamental group of theorems that governs the rest of mathematics.
- For our purpose, we use the set theory in order to manipulate groups of elements.
- A set is a collection of distinct objects.
- That's it! A set can be thought of as a list in Python, but with no repeat objects.
- In fact, there even exists a set of objects in Python:
- s = set()
- s = set([1, 2, 2, 3, 2, 1, 2, 2, 3, 2])
- # will remove duplicates from a list
- s == {1, 2, 3}

- dict = {"dog": "human's best friend", "cat": "destroyer of world"}
- dict["dog"] # == "human's best friend"
- len(dict["cat"]) # == 18
- # but if we try to create a pair with the same key as an existing key
- dict["dog"] = "Arf"
- dict
- {"dog": "Arf", "cat": "destroyer of world"}
- # It will override the previous value
- # dictionaries cannot have two values for one key.
- They share this notation because they share a quality in that sets cannot have duplicate elements, just as dictionaries cannot have duplicate keys.
- The magnitude of a set is the number of elements in the set and is represented as follows:
- $|A| = magnitute\ of\ A$

- s # == {1,2,3}
- len(s) == 3 # magnitude of s

- If we wish to denote that an element is within a set, we use the epsilon notation, as shown:
- $2 \epsilon \{1,2,3\}$
- This means that the element 2 exists in the set of 1, 2, and 3.
- If one set is entirely inside another set, we say that it is a subset of its larger counterpart.
- $A = \{1,5,6\}$, $B = \{1,5,6,7,8\}$
- $A \subseteq B$
- So, A is a subset of B and B is called the superset of A. If A is a subset of B but A does not equal B (meaning that there is at least one element in B that is not in A), then A is called a proper subset of B.
- Consider the following examples:
- • A set of even numbers is a subset of all integers
- • Every set is a subset, but not a proper subset, of itself
- • A set of all tweets is a superset of English tweets
- In data science, we use sets (and lists) to represent a list of objects and, often, to generalize the behavior of consumers. It is common to reduce a customer to a set of characteristics.

- Consider that we are a marketing firm trying to predict where a person wants to
- shop for clothes. We are given a set of clothing brands the user has previously
- visited, and our goal is to predict a new store that they would also enjoy.
- Suppose a specific user has previously shopped at the following stores:

- user1 = {"Target","Banana Republic","Old Navy"}
- # note that we use {} notation to create a set
- # compare that to using [] to make a list

- So, user1 has previously shopped at Target, Banana Republic, and Old Navy.
- Let's also look at a different user, called user2, as shown:

- user2 = {"Banana Republic","Gap","Kohl's"}

- Suppose we are wondering how similar these users are. With the limited information we have, one way to define similarity is to see how many stores there are that they both shop at.
- This is called an intersection.
- The intersection of two sets is a set whose elements appear in both the sets. It is
- denoted using the symbol , as shown:
- User1 $\cap User\ 2\ = \{Banana\ Republic\}$
- $|User1\ \cap User\ 2\ | = 1$

- The intersection of the two users is just one store. So, right away that doesn't seem great. However, each user only has three elements in their set, so having 1/3 does not seem as bad.
- Suppose we are curious about how many stores are represented between the two of them; this is called a union.
- The union of two sets is a set whose elements appear in either set. It is denoted using the symbol ∪, as shown:
- User1 ∪ User2 = {Banana republic, Target, Old navy, Gap, Kohl's}
- |User1 ∪ User2|= 5
- When looking at the similarity between user1 and user2, we should use a combination of the union and the intersection of their sets.
- user1 and user2 have one element in common out of a total of five distinct elements between them. So, we can define the similarity between the two users as follows:
- $\frac{|User1 \cap User\ 2|}{|\text{User1} \cup \text{User2}|} = \frac{1}{5} = 0.2$
- In fact, this has a name in the set theory. It is called the **jaccard measure**.
- In general, for the sets A and B, the jaccard measure (jaccard similarity) between the two sets is defined as follows:
- $\text{JS(A, B)} = \frac{|A \cap B|}{|\text{A} \cup B|}$
- It can also be defined as the magnitude of the intersection of the two sets divided by the magnitude of the union of the two sets.
- This gives us a way to quantify similarities between elements represented with sets.

- Intuitively, the jaccard measure is a number between 0 and 1, such that when the number is closer to 0, the people are more dissimilar and when the measure is closer to 1, the people are considered similar to each other.
- If we think about the definition, then it actually makes sense. Take a look at the
- measure once more:
- JS (A, B) = $\frac{Number\ of\ stores\ they\ share\ in\ common}{Unique\ No.of\ store\ they\ like\ combined}$
- Here, the numerator represents the number of stores that the users have in common (in the sense that they like shopping there), while the denominator represents the unique number of stores that they like put together.
- We can represent this in Python using some simple code, as shown:

- user1 = {"Target","Banana Republic","Old Navy"}
- user2 = {"Banana Republic","Gap","Kohl's"}
- def jaccard(user1, user2):
- stores_in_common = len(user1 & user2)
- stores_all_together = len(user1 | user2)
- return stores / float(stores_all_together)

- # I cast stores_all_together as a float to return a decimal answer instead of python's default integer division # so
- jaccard(user1, user2) == # 0.2 or 1/5

- The set theory becomes highly prevalent when we enter the world of probability and also when dealing with high-dimensional data.
- We will use sets to represent real-world events taking place and probability becomes set theory with vocabulary on top of it.

Linear algebra
Matrix multiplication

- Like numbers, we can multiple matrices together. Multiplying matrices is, in essence, a mass produced way of taking several dot products at once. Let's, for example, try to multiple the following matrices:
- $\begin{matrix} 1 & 5 \\ 5 & 8 \\ 7 & 8 \end{matrix} \cdot \begin{matrix} 3 & 4 \\ 2 & 5 \end{matrix}$
- Unlike numbers, multiplication is not commutative, meaning that the order in which you multiply matrices matters a great deal.
- In order to multiply matrices, their dimensions must match up.
- This means that the first matrix must have the same number of columns as the second matrix has rows.
- To remember this, write out the dimensions of the matrices. In this case, we have a 3 x 2 times a 2 x 2 matrix. You can multiple matrices together if the second number in the first dimension pair is the same as the first number in the second dimension pair.

$$3 \times \boxed{2 \cdot 2} \times 2$$

How to multiply matrices

- From the previous example:

  13 29
- 31 60

  37 68
- Way to go! Let's come back to the movie recommendation example. Recall the user's movie genre preferences of comedy, romance, and action, which are illustrated a

$$U = user\ prefs = \begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix}$$

- Now suppose we have 10,000 movies, all with a rating for these three categories. To make a recommendation, we need to take the dot product of the preference vector with each of the 10,000 movies. We can use matrix multiplication to represent this.
- Instead of writing them all out, let's express it using the matrix notation.
- We already have U, defined here as the user's preference vector (it can also be thought of as a 3 x 1 matrix), and we also need a movie matrix:

- M = movies = 3 * 10,000 dimension matrix
- So, now we have two matrices, one is 3 x 1 and the other is 3 x 10,000. We can't multiply these matrices as they are because the dimensions do not work out.
- We will have to change U a bit. We can take the transpose of the matrix (turning all rows into columns and columns into rows). This will switch the dimensions around:
- *U^T = transpose of U = (5 1 3)*
- So, now we have two matrices that can be multiplied together. To visualize what this looks like:

$$\begin{pmatrix} 5 & 1 & 3 \end{pmatrix} \cdot \begin{pmatrix} 4 & 5 & 2 & \cdots \\ & & & \\ 1 & 5 & 1 & \end{pmatrix}$$

$$1 \times \boxed{3 \qquad 3} \times 10000$$

- The resulting matrix will be a 1 x 1,000 matrix (a vector) of 10,000 predictions for each individual movie. Let's try this out in Python!

- import numpy as np
- user_pref = np.array([5, 1, 3]) # create user preferences
- # create a random movie matrix of 10,000 movies
- movies = np.random.randint(5,size=(3,1000))+1
- # Note that the randint will make random integers from 0-4
- # so I added a 1 at the end to increase the scale from 1-5

- We are using the numpy array function to create our matrices. We will have both a user_pref and a movies matrix to represent our data.
- To check our dimensions, we can use the numpy shape variable, as shown:
- print user_pref.shape # (1, 3)
- print movies.shape # (3, 1000)
- This checks out. Last but not least, let's use the matrix multiplication method of numpy (called dot) to perform the operation, as illustrated:
- # np.dot does both dot products and matrix multiplication
- np.dot(user_pref, movies)

- The result is an array of integers that represents the recommendations of each movie.
- For a quick extension of this, let's run some code that predicts across more than 10,000 movies, as shown:

```
import time
for num_movies in (10000, 100000, 1000000, 10000000, 100000000):
movies = np.random.randint(5,size=(3, movies))+1
now = time.time()
np.dot(user_pref, movies)
print (time.time() - now), "seconds to run", movies, "movies“
```

- 0.000160932540894 seconds to run 10000 movies
- 0.00121188163757 seconds to run 100000 movies
- 0.0105860233307 seconds to run 1000000 movies
- 0.096577167511 seconds to run 10000000 movies
- 4.16197991371 seconds to run 100000000 movies
- It took only a bit over 4 seconds to run through 100,000,000 movies using matrix multiplication.

Basic Statistics

- In this chapter, we will look at the following topics:
- • How to obtain and sample data
- • The measures of center, variance, and relative standing
- • Normalization of data using the z-score
- • The Empirical rule

What are statistics?

- To answer this question, we need to back up for a minute and talk about why we even measure them in the first place.
- The goal of this field is to try to explain and model the world around us. To do that, we have to take a look at the population.
- We can define a population as the entire pool of subjects of an experiment or a model.
- If you are trying to test if smoking leads to heart disease, your population would be the smokers of the world.
- If you are trying to study teenage drinking problems, your population would be all teenagers.
- Now, consider that you want to ask a question about your population, for example, if your population is all of your employees (assume that you have over 1,000 employees), perhaps you want to know what percentage of them use illicit drugs.
- The question is called a parameter.

- We can define a parameter as a numerical measurement describing a characteristic of a population.
- For example, if you ask all 1,000 employees and 100 of them are using drugs, the rate of drug use is 10%. The parameter here is 10%.
- However, let's get real, you probably can't ask every single employee whether they are using drugs.
- What if you have over 10,000 employees? It would be very
- difficult to track everyone down in order to get your answer.
- When this happens, it's impossible to figure out this parameter.
- In this case, we can estimate the parameter.
- First, we will take a sample of the population.
- We can define a <u>sample</u> of a population as a subset (random not required) of the population.

- So, we perhaps ask 200 of the 1,000 employees you have. Of these 200, suppose 26 use drugs, making the drug use rate 13%.
- Here, 13% is not a parameter because we didn't get a chance to ask everyone. This 13% is an estimate of a parameter.
- Do you know what that's called?
- That's right, a statistic!
- **We can define a statistic as a numerical measurement describing a characteristic of a sample of a population.**
- A statistic is just an estimation of a parameter. It is a number that attempts to describe an entire population by describing a subset of that population.
- This is necessary because you can never hope to give a survey to every single teenager or to every single smoker in the world.
- That's what the field of statistics is all about—taking samples of populations and running tests on these samples.
- So, the next time you are given a statistic, just remember, that number only represents a sample of that population, not the entire pool of subjects.

How do we obtain and sample data?

- <u>Obtaining data:</u>
- There are two main ways of collecting data for our analysis: observational and experimentation. Both these ways have their pros and cons, of course.
- <u>Observational:</u>
- We might obtain data through observational means, which consists of measuring specific characteristics but not attempting to modify the subjects being studied.
- For example, you have a tracking software on your website that observes users' behavior on the website, such as length of time spent on certain pages and the rate of clicking on ads, all the while not affecting the user's experience, then that would be an observational study.
- This is one of the most common ways to get data because it's just plain easy. All you have to do is observe and collect data.
- Observational studies are also limited in the types of data you may collect.
- This is because the observer (you) is not in control of the environment. You may only watch and collect natural behavior.
- If you are looking to induce a certain type of behavior, an observational study would not be useful.

- <u>Experimental:</u>
- An experiment consists of a treatment and the observation of its effect on the subjects.
- Subjects in an experiment are called experimental units.
- This is usually how most scientific labs collect data. They will put people into two or more groups (usually just two) and call them the control and the experimental group.
- The control group is exposed to a certain environment and then observed.
- The experimental group is then exposed to a different environment and then observed.
- The experimenter then aggregates data from both the groups and makes a decision about which environment was more favorable (favorable is a quality that the experimenter gets to decide).

- In a marketing example, consider that we expose half of our users to a certain landing page with certain images and a certain style (website A), and we measure whether or not they sign up for the service.
- Then, we expose the other half to a different landing page, different images, and different styles (website B) and again measure whether or not they sign up.
- We can then decide which of the two sites performed better and should be used going further.
- This, specifically, is called an A/B test.
- Let's see an example in Python! Let's suppose we run the preceding test and obtain the following results as a list of lists:
- results = [['A', 1], ['B', 1], ['A', 0], ['A', 0] …]
- Here, each object in the list result represents a subject (person). Each person then has the following two attributes:

- Once we create these two lists that will eventually hold each individual conversion Boolean (0 or 1), we will iterate all of our results of the test and add them to the appropriate list, as shown:
- for website, converted in results:
- # iterate through the results
- # will look something like website == 'A' and converted == 0
- if website == 'A':
- users_exposed_to_A.append(converted)
- elif website == 'B':
- users_exposed_to_B.append(converted)
- Now, each list contains a series of 1s and 0s.
- To get the total number of people exposed to website A, we can use the len() feature
- in Python, as illustrated:
- len(users_exposed_to_A) == 188 #number of people exposed to website A
- len(users_exposed_to_B) == 158 #number of people exposed to website B

- To count the number of people who converted, we can use the sum() of the list, as shown:
- sum(users_exposed_to_A) == 54 # people converted from website A
- sum(users_exposed_to_B) == 48 # people converted from website B
- If we subtract the length of the lists and the sum of the list, we are left with the number of people who did not convert for each site, as illustrated:
- len(users_exposed_to_A) - sum(users_exposed_to_A) == 134 # did not convert from website A
- len(users_exposed_to_B) - sum(users_exposed_to_B) == 110 # did not convert from website B
- We can aggregate and summarize our results in the following table that represents our experim testing:

	Did not sign up	Signed up
Website A	134	54
Website B	110	48

The results of our A/B test

- We can quickly drum up some descriptive statistics. We can say that the website conversion rates for the two websites are as follows:
  - Conversion for website A: $\frac{54}{134+54} = .288$
  - Conversion for website B: $\frac{48}{110+48} = .3$

- Not much difference, but different nonetheless.
- Even though B has the higher conversion rate, can we really say that the version B significantly converts better?
- Not yet. To test the statistical significance of such a result, a hypothesis test should be used.
- These tests will be covered in depth in the next chapter, where we will revisit this exact same example and finish it using the proper statistical test.

Sampling data

- Remember how statistics are the result of measuring a sample of a population.
- Well, we should talk about two very common ways to decide who gets the honor of being in the sample that we measure.
- We will discuss the main type of sampling, called random sampling, which is the most common way to decide our sample sizes and our sample members.
- <u>Probability sampling:</u>
- Probability sampling is a way of sampling from a population, in which every person has a known probability of being chosen but that number might be a different probability than another user.
- The simplest (and probably the most common) probability sampling method is random sampling.

Random sampling

- Suppose that we are running an A/B test and we need to figure out who will be in group A and who will be in group B.
- There are the following three suggestions from your data team:
- 1, Separate users based on location: Users on the west coast are placed in group A, while users on the east coast are placed in group B
- 2, Separate users based on the time of day they visit the site: Users who visit between 7 p.m. and 4 a.m. get site A, while the rest are placed in group B
- 3, Make it completely random: Every new user has a 50/50 chance of being placed in either group

- The first two are valid options for choosing samples and are fairly simple to implement, but they both have one fundamental flaw: they are both at risk of introducing a sampling bias.
- A sampling bias occurs when the way the sample is obtained systemically favors some outcome over the target outcome.
- It is not difficult to see why choosing option 1 or option 2 might introduce bias. If we chose our groups based on where they live or what time they log in, we are priming our experiment incorrectly and, now, we have much less control over the results.
- Specifically, we are at risk of introducing a confounding factor into our analysis, which is bad news.
- A confounding factor is a variable that we are not directly measuring but connects the variables that are being measured.
- Basically, a confounding factor is like the missing element in our analysis that is invisible but affects our results

- In this case, option 1 is not taking into account the potential confounding factor of geographical taste. For example, if website A is unappealing, in general, to the west coast users, it will affect your results drastically.
- Similarly, option 2 might introduce a temporal (time-based) confounding factor. What if website B is better viewed in a nighttime environment (which was reserved for A), and users are turned off to the style purely because of what time it is. These are both factors that we want to avoid, so, we should go with option 3, which is a random sample

- A <u>random sample</u> is chosen such that every single member of a population has an equal chance of being chosen as any other member.
- This is probably one of the easiest and most convenient ways to decide who will be a part of your sample.
- Everyone has the exact same chance of being in any particular group.
- Random sampling is an effective way of reducing the impact of confounding factors

Unequal probability sampling

- Suppose we are interested in measuring the happiness level of our employees.
- We already know that we can't ask every single person on the staff because that would be silly and exhausting. So, we need to take a sample.
- Our data team suggests random sampling and at first everyone high fives because they feel very smart and statistical.
- But then someone asks a seemingly harmless question— does anyone know the percentage of men/women who work here? The high fives stop and the room goes silent
- This question is extremely important because Gender is likely to be a confounding factor. The team looks into it and discovers a split of 75% men and 25% women in the company

- This means that if we introduce a random sample, our sample will likely have a similar split and, thus, favor the results for men and not women.
- To combat this, we can favor including more women than men in our survey in order to make the split of our sample less favored for men.
- At first glance, introducing a favoring system in our random sampling seems like a bad idea, however, alleviating <u>unequal sampling</u> and, therefore, working to remove systematic bias among gender, race, disability, and so on is much more pertinent.
- A simple random sample, where everyone has the same chance as everyone else, is very likely to drown out the voices and opinions of minority population members.
- Therefore, it can be okay to introduce such a favoring system in your sampling techniques

How do we measure statistics?

- Once we have our sample, it's time to quantify our results.
- Suppose we wish to generalize the happiness of our employees or we want to figure out whether salaries in the company are very different from person to person.
- These are some common ways of measuring our results.

Measures of center

- Measures of center are how we define the middle, or center, of a dataset.
- We do this because sometimes we wish to make generalizations about data values.
- For example, perhaps we're curious about what the average rainfall in Seattle is or what the median height for European males is.
- It's a way to generalize a large set of data so that it's easier to convey to someone.
- A measure of center is a value in the "middle" of a dataset.
- However, this can mean different things to different people.
- Who's to say where the middle of a dataset is?
- There are so many different ways of defining the center of data. Let's take a look at a few.
- The arithmetic mean of a dataset is found by adding up all of the values and then dividing it by the number of data values.

- This is likely the most common way to define the center of data, but can be flawed!
- Suppose we wish to find the mean of the following numbers:
- import numpy as np
- np.mean([11, 15, 17, 14]) == 14.25
- Simple enough, our average is 14.25 and all of our values are fairly close to it.
- But what if we introduce a new value: 31?
- np.mean([11, 15, 17, 14, 31]) == 17.6
- This greatly affects the mean because the arithmetic mean is sensitive to <u>outliers</u>.
- The new value, 31, is almost twice as large as the rest of the numbers and, therefore, skews the mean.
- Another, and sometimes better, measure of center is the <u>median</u>

- The **median** is the number found in the middle of the dataset when it is sorted in order,
- as shown: np.median([11, 15, 17, 14]) == 14.5
- np.median([11, 15, 17, 14, 31]) == 15
- Note how the introduction of 31 using the median did not affect the median of the dataset greatly.
- This is because the median is less sensitive to outliers.
- When working with datasets with many outliers, it is sometimes more useful to use the median of the dataset, while if your data does not have many outliers and the data points are mostly close to one another, then the mean is likely a better option.
- But how can we tell if the data is spread out? Well, we will have to introduce a new type of statistic

Measures of variation

- Measures of center are used to quantify the middle of the data, but now we will explore ways of measuring how "spread out" the data we collect is.
- This is a useful way to identify if our data has many outliers lurking inside.
- Let's start with an example. Consider that we take a random sample of 24 of our friends on Facebook and wrote down how many friends that they had on Facebook.
- Here's the list:
- friends = [109, 1017, 1127, 418, 625, 957, 89, 950, 946, 797, 981, 125, 455, 731, 1640, 485, 1309, 472, 1132, 1773, 906, 531, 742, 621]

- np.mean(friends) == 789.1

- The average of this list is just over 789. So, we could say that according to this sample, the average Facebook friend has 789 friends.
- But what about the person who only has 89 friends or the person who has over 1,600 friends?
- In fact, not a lot of these numbers are really that close to 789.
- Well, how about we use the median, as shown, because the median generally is not as affected by outliers:
- np.median(friends) == 769.5
- The median is 769.5, which is fairly close to the mean.
- Hmm, good thought, but still, it doesn't really account for how drastically different a lot of these data points are to one another.
- This is what statisticians call measuring the variation of data.
- Let's start by introducing the most basic measure of variation: <u>the range</u>.
- The range is simply the maximum value minus the minimum value, as illustrated:
- np.max(friends) – np.min(friends) == 1684

- The range tells us how far away the two most extreme values are. Now, typically, the range isn't widely used but it does have its use in application.
- Sometimes we wish to just know how spread apart the outliers are.
- This is most useful in scientific measurements or safety measurements.
- Suppose a car company wants to measure how long it takes for an air bag to deploy.
- Knowing the average of that time is nice, but they also really want to know how spread apart the slowest time is versus the fastest time.
- This literally could be the difference between life and death.
- Shifting back to the Facebook example, 1,684 is our range, but I'm not quite sure it's saying too much about our data.
- Now, let's take a look at the most commonly used measure of variation, the <u>standard deviation</u>.
- In essence, standard deviation, denoted by s when we are working with a sample of a population, measures how much data values deviate from the arithmetic mean.
- It's basically a way to see how spread out a general formula to calculate the standard deviation is:

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

- • s is our sample standard deviation
- • x is each individual data point.
- • $\bar{x}$ is the mean of the data
- • n is the number of data points

- For each value in the sample, we will take that value, subtract the arithmetic mean from it, square the difference, and, once we've added up every single point this way, we will divide the entire thing by n, the number of points in the sample.
- Finally, we take a square root of everything. Without going into an in-depth analysis of the formula, think about it this way: it's basically derived from the distance formula.
- **Essentially, what the standard deviation is calculating is a sort of average distance of how far the data values are from the arithmetic mean**

- If you take a closer look at the formula, you will see that it actually makes sense:
- For example, by squaring the number of friends minus the mean, we changed our units to friends square, which makes no sense. Taking the square root puts our units back to just "friends".
- Let's go back to our Facebook example for a visualization and further explanation of this. Let's begin to calculate the standard deviation.
- So, we'll start calculating a few of them. Recall that the arithmetic mean of the data was just about 789, so, we'll use 789 as the mean.
- We start by taking the difference between each data value and the mean, squaring it, adding them all up, dividing it by one less than the number of values, and then taking its square root. This would look as follows:

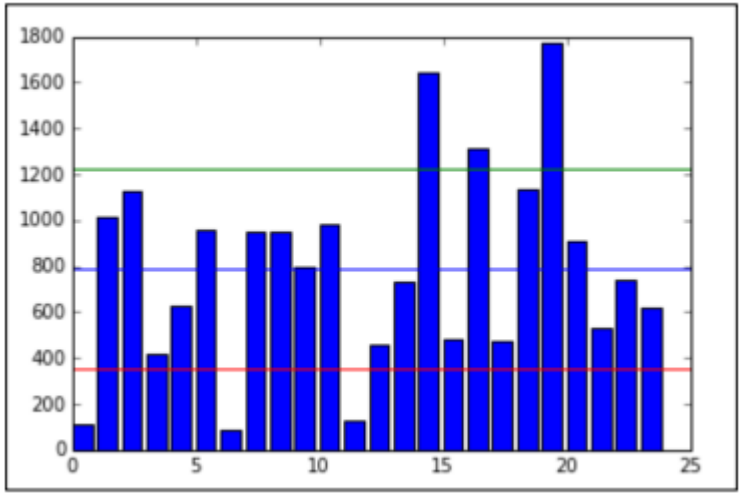
$$s = \sqrt{\frac{(109-789)^2 + (1017-789)^2 + \cdots + (621-789)^2}{24}}$$

- On the other hand, we can take the Python approach and do all this programmatically (which is usually preferred).
- np.std(friends) # == 425.2
- What the number 425 represents is the spread of data.
- You could say that 425 is a kind of average distance the data values are from the mean.
- What this means, in simple words, is that this data is pretty spread out.
- So, our standard deviation is about 425.
- This means that the number of friends that these people have on Facebook doesn't seem to be close to a single number and that's quite evident when we plot the data in a bar graph and also graph the mean as well as the visualizations of the standard deviation.

- In the following plot, every person will be represented by a single bar in the bar chart, and the height of the bars represent the number of friends that the individuals have:
- import matplotlib.pyplot as plt
- %matplotlib inline
- y_pos = range(len(friends))

- plt.bar(y_pos, friends)
- plt.plot((0, 25), (789, 789), 'b-')
- plt.plot((0, 25), (789+425, 789+425), 'g-')
- plt.plot((0, 25), (789-425, 789-425), 'r-')

- The blue line in the center is drawn at the mean (789), the red line on the bottom is drawn at the mean minus the standard deviation (789-425 = 364), and, finally, the green line towards the top is drawn at the mean plus the standard deviation (789+425 = 1,214).
- Note how most of the data lives between the green and the red lines while the outliers live outside the lines.
- Namely, there are three people who have friend counts below the red line and three people who have a friend count above the green line.
- It's important to mention that the units for standard deviation are, in fact, the same units as the data's units. So, in this example, we would say that the standard deviation is 425 friends on Facebook



- So, now we know that the standard deviation and variance is good for checking how spread out our data is, and that we can use it along with the mean to create a kind of range that a lot of our data lies in.
- But what if we want to compare the spread of two different datasets, maybe even with completely different units?
- That's where the <u>coefficient of variation</u> comes into play

Coefficient of variation

- **The coefficient of variation is defined as the ratio of the data's standard deviation to its mean.**
- This ratio (which, by the way, is only helpful if we're working in the ratio level of measurement, where division is allowed and is meaningful) is a way to standardize the standard deviation, which makes it easier to compare across datasets.
- We use this measure frequently when attempting to compare means, and it spreads across populations that exist at different scales

Example – employee salaries

- If we look at the mean and standard deviation of employees' salaries in the same company but among different departments, we see that, at first glance, it may be tough to compare variations

Salaries of Company XYZ

Department	Mean Salary	SD	CoV
Mailroom	$25,000	$2,000	8.0%
Human Resources	$52,000	$7,000	13.5%
Executive	$124,000	$42,000	33.9%

- With measures of variation, we can begin to answer big questions, such as how spread out this data is or how we can come up with a good range that most of the data falls in

Measures of Relative Standing

- We can combine both the measures of centers and variations to create measures of relative standings.
- **Measures of variation** measure where particular data values are positioned, relative to the entire dataset.
- Let's begin by learning a very important value in statistics, the z-score.
- The **z-score** is a way of telling us how far away a single data value is from the mean.
- The z-score of a x data value is as follows:

- Where: $$z = \frac{x - \bar{x}}{s}$$

- • x is the data point
- • $\bar{x}$ is the mean
- • s is the standard deviation.

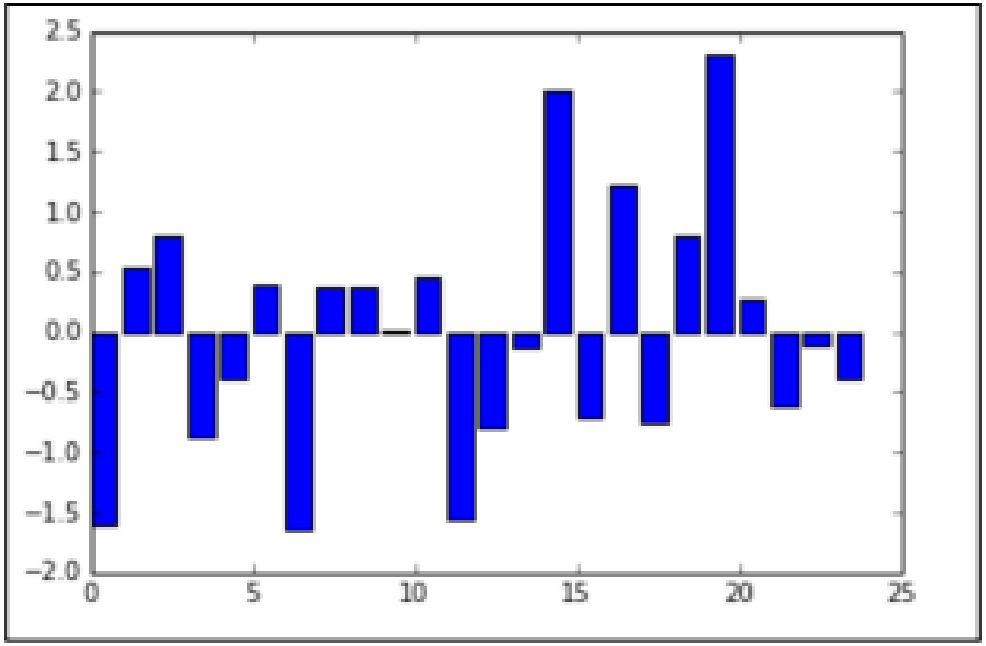
- Remember that the standard deviation was (sort of) an average distance that the data is from the mean, and, now,
- the z-score is an individualized value for each particular data point.
- We can find the z-score of a data value by subtracting it from the mean and dividing it by the standard deviation.
- The output will be the standardized distance a value is from a mean.
- We use the z-score all over statistics. It is a very effective way of normalizing data that exists on very different scales, and also to put data in context of their mean

- Let's take our previous data on the number of friends on Facebook and standardize the data to the z-score.
- For each data point, we will find its z-score by applying the preceding formula. We will take each individual, subtract the average friends from the value, and divide that by the standard deviation, as shown:
- z_scores = []
- m = np.mean(friends) # average friends on Facebook
- s = np.std(friends) # standard deviation friends on Facebook

- for friend in friends:
- z = (friend - m)/s # z-score
- z_scores.append(z) # make a list of the scores for plotting

- Now, let's plot these z-scores on a bar chart.
- The following chart shows the same individuals from our previous example using friends on Facebook, but, instead of the bar height revealing the raw number of friends, now each bar is the z-score of the number of friends they have on Facebook.
- If we graph the z-scores, we'll notice a few things:
- plt.bar(y_pos, z_scores)

- We have negative values (meaning that the data point is below the mean)
- • The bars' lengths no longer represent the raw number of friends, but the degree to which that friend count differs from the mean.



- This chart makes it very easy to pick out the individuals with much lower and higher friends on an average.
- For example, the individual at index 0 has fewer friends on an average (they had 109 friends where the average was 789).
- What if we want to graph the standard deviations?
- Recall that we earlier graphed three horizontal lines:
- one at the mean,
- one at the mean plus the standard deviation (x + s),
- and one at the mean minus the standard deviation (x - s).
- If we plug in these values into the formula for the z-score, we will get:

$$\text{Z-score of } (\bar{x}\,x) = \frac{x-\bar{x}}{s} = \frac{0}{s} = 0$$

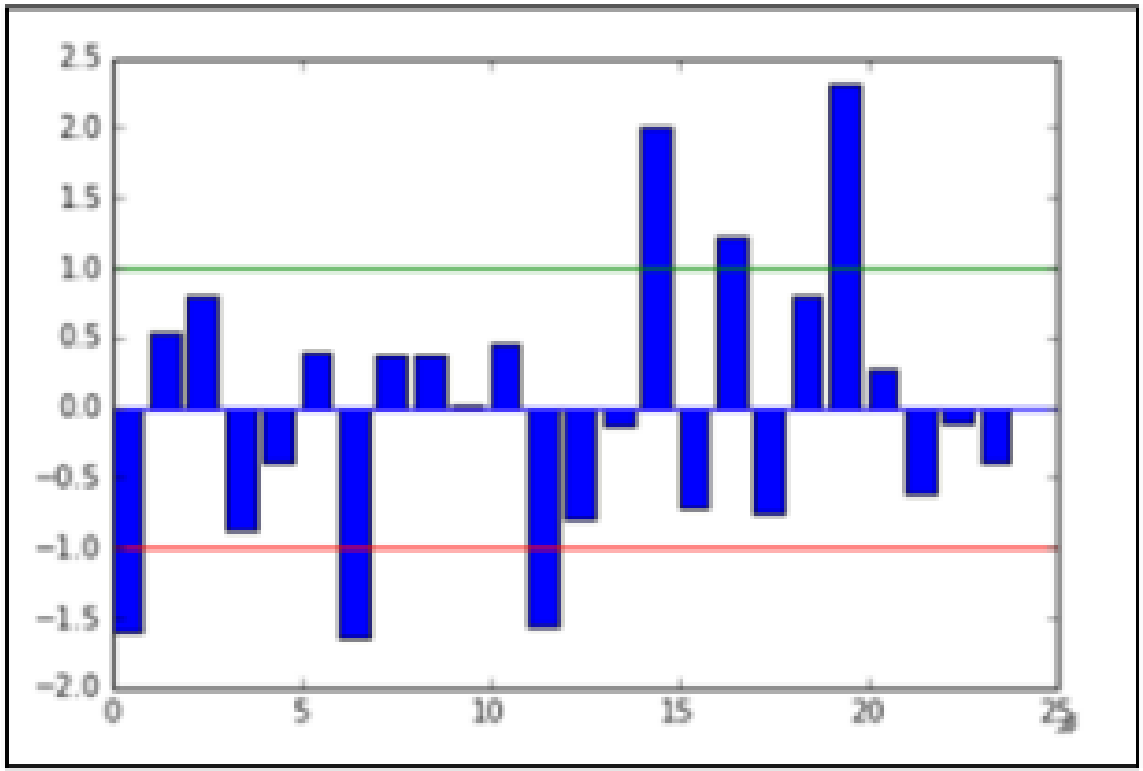
$$\text{Z-score of } (x+s) = \frac{(\bar{x}+s)-\bar{x}}{s} = \frac{s}{s} = 1$$

$$\text{Z-score of } (x-s)\ \frac{(\bar{x}-s)-\bar{x}}{s} = \frac{-s}{s} = -1$$

- This is no coincidence! When we standardize the data using the z-score, our standard deviations become the metric of choice.
- Let's see our new graph with the standard deviations plotted:
- plt.bar(y_pos, z_scores)
- plt.plot((0, 25), (1, 1), 'g-')
- plt.plot((0, 25), (0, 0), 'b-')
- plt.plot((0, 25), (-1, -1), 'r-')

- The preceding code is adding in the following three lines:
- • A blue line at y = 0 that represents zero standard deviations away from the mean (which is on the x axis)
- • A green line that represents one standard deviation above the mean
- • A red line that represents one standard deviation below the mean.
- The colors of the lines match up with the lines drawn in the earlier graph of the raw friend count.
- If you look carefully, the same people still fall outside of the green and the red lines.
- Namely, the same three people still fall below the red (lower) line, and the same three people fall above the green (upper) line

- Under this scaling, we can also use statements as follows:
- This data point is over one standard deviation away from the mean:
- This person has a friend count within one standard deviation from the mean:



- Z-scores are an effective way to standardize data.
- This means that we can put the entire set on the same scale.
- For example, if we also measure each person's general happiness scale (which is between 0 and 1), we might have a dataset similar to the following dataset:

- friends = [109, 1017, 1127, 418, 625, 957, 89, 950, 946, 797, 981, 125, 455, 731, 1640, 485, 1309, 472, 1132, 1773, 906, 531, 742, 621]
- happiness = [.8, .6, .3, .6, .6, .4, .8, .5, .4, .3, .3, .6, .2, .8, 1, .6, .2, .7, .5, .3, .1, 0, .3, 1]

- import pandas as pd
- df = pd.DataFrame({'friends':friends, 'happiness':happiness})
- df.head()

- These data points are on two different dimensions, each with a very different scale.
- The friend count can be in the thousands while our happiness score is stuck between 0 and 1.

	friends	happiness
0	109	0.8
1	1017	0.6
2	1127	0.3
3	418	0.6
4	625	0.6

- To remedy this (and for some statistical/machine learning modeling, this concept will become essential), we can simply standardize the dataset using a prebuilt standardization package in *scikit-learn*, as follows:
- from sklearn import preprocessing
- df_scaled = pd.DataFrame(preprocessing.scale(df), columns = ['friends_ scaled', 'happiness_scaled'])
- df_scaled.head()

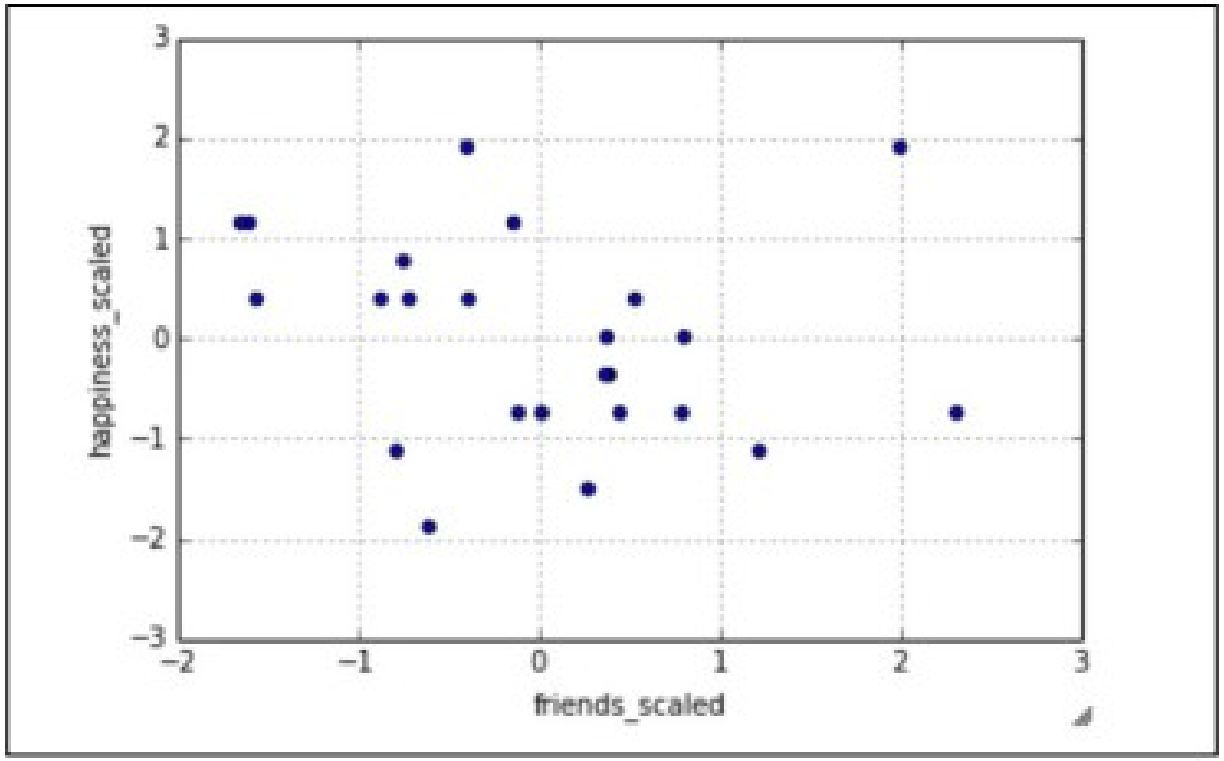
- This code will scale both the friends and happiness columns simultaneously, thus revealing the z-score for each column.
- It is important to note that by doing this, the preprocessing module in *sklearn* is doing the following things separately for each column:
- • Finding the mean of the column
- • Finding the standard deviation of the column
- • Applying the z-score function to each element in the column

- The result is two columns, as shown, that exist on the same scale as each other even if they were not previously:

	friends_scaled	happiness_scaled
0	-1.599495	1.153223
1	0.536040	0.394939
2	0.794750	-0.742486
3	-0.872755	0.394939
4	-0.385909	0.394939

- Now, we can plot friends and happiness on the same scale and the graph will at least be readable

- Now, we can plot friends and happiness on the same scale and the graph will at least be readable.
- df_scaled.plot(kind='scatter', x = 'friends_scaled', y = 'happiness_ scaled')



- Now our data is standardized to the z-score and this scatter plot is fairly easily interpretable!

The insightful part – correlations in data

- Having data is only one step to a successful data science operation.
- Being able to obtain, clean, and plot data helps to tell the story that the data has to offer but cannot reveal the moral.
- In order to take this entire example one step further, we will look at the relationship between having friends on Facebook and happiness.
- The question of the day here is—can we find a relationship between the number of friends on Facebook and overall happiness?
- Given the nature of our data, we really only have the following three options for a hypothesis:
- • There is a positive association between the number of online friends and happiness (as one goes up, so does the other)
- • There is a negative association between them (as the number of friends goes up, your happiness goes down)
- • There is no association between the variables (as one changes, the other doesn't really change that much)

- Can we use basic statistics to form a hypothesis about this question? I say we can!
- But first, we must introduce a concept called <u>correlation</u>.
- **Correlation coefficients are a quantitative measure that describe the strength of association/relationship between two variables.**
- The correlation between two sets of data tells us about how they move together.
- Would changing one help us predict the other?
- This concept is not only interesting in this case, but it is one of the core assumptions that many machine learning models make on data.
- For many prediction algorithms to work, they rely on the fact that there is some sort of relationship between the variables that we are looking at.
- The learning algorithms then exploit this relationship in order to make accurate predictions.

- A few things to note about a correlation coefficient are as follows:
- • It will lie between -1 and 1
- • The greater the absolute value (closer to -1 or 1), the stronger the relationship between the variables: ° The strongest correlation is a -1 or a 1 ° The weakest correlation is a 0
- • A positive correlation means that as one variable increases, the other one tends to increase as well
- • A negative correlation means that as one variable increases, the other one tends to decrease
- We can use Pandas to quickly show us correlation coefficients between every feature and every other feature in the Dataframe, as illustrated:

- # correlation between variables
- df.corr()

- This table shows the correlation between friends and happiness.
- Note the first two things, shown as follows:
- • The diagonal of the matrix is filled with positive is. This is because they represent the correlation between the variable and itself, which, of course, forms a perfect line, making the correlation perfectly positive!
- • The matrix is symmetrical across the diagonal. This is true for any correlation matrix made in Panda

	friends	happiness
friends	1.000000	-0.216199
happiness	-0.216199	1.000000

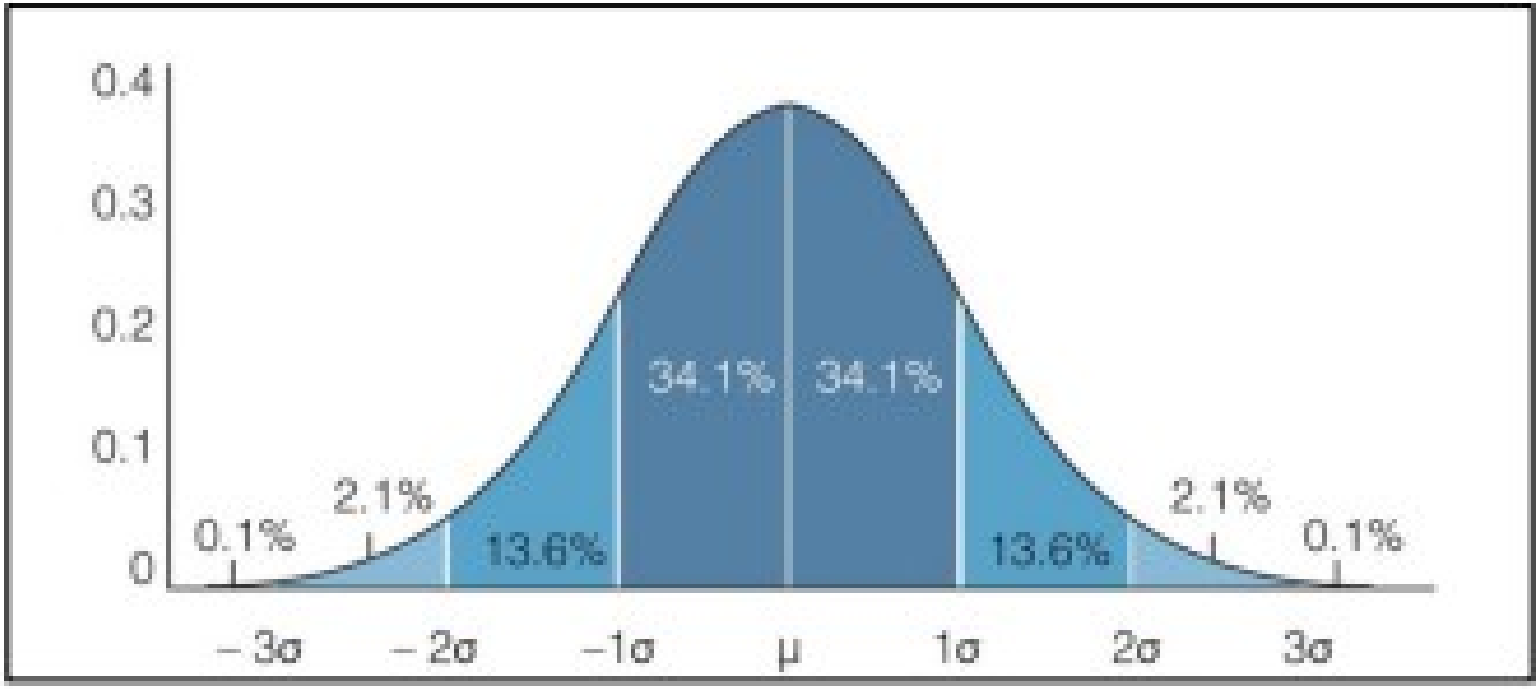
- In general, a correlation will attempt to measure a *linear relationship* between variables.
- This means that if there is no visible correlation revealed by this measure, it does not mean that there is no relationship between the variables, only that there is no line of best fit that goes through the lines easily.
- There might be a *non-linear* relationship that defines the two variables.
- **It is important to realize that causation is not implied by correlation.**
- Just because there is a weak negative correlation between these two variables does not necessarily mean that your overall happiness decreases as the number of friends you keep on Facebook goes up.
- This causation must be tested further. To sum up, we can use correlation to make hypotheses about the relationship between variables, but we will need to use more sophisticated statistical methods and machine learning algorithms to solidify these assumptions and hypotheses.

The Empirical rule

- Recall that a normal distribution is defined as having a specific probability distribution that resembles a bell curve.
- In statistics, we love it when our data behaves normally.
- For example, if we have data that resembles a normal distribution, like so:
- The Empirical rule states that we can expect a certain amount of data to live between sets of standard deviations.

- Specifically, the Empirical rule states for data that is distributed normally:

- • about 68% of the data fall within 1 standard deviation
- • about 95% of the data fall within 2 standard deviations
- • about 99.7% of the data fall within 3 standard deviations


0.4
0.3
0.2
0.1
0
0.1%
2.1%
13.6%
34.1%
34.1%
13.6%
2.1%
0.1%
– 3σ
– 2σ
–1σ
μ
1σ
2σ
3σ

- For example, let's see if our Facebook friends' data holds up to this.
- Let's use our Dataframe to find the percentage of people that fall within 1, 2, and 3 standard deviations of the mean, as shown:

- # finding the percentage of people within one standard deviation of the mean
- within_1_std = df_scaled[(df_scaled['friends_scaled'] <= 1) & (df_scaled['friends_scaled'] >= -1)].shape[0]
- within_1_std / float(df_scaled.shape[0])
- # 0.75

- # finding the percentage of people within two standard deviations of the mean w
- ithin_2_std = df_scaled[(df_scaled['friends_scaled'] <= 2) & (df_scaled['friends_scaled'] >= -2)].shape[0] within_2_std / float(df_scaled.shape[0])
- # 0.916

- # finding the percentage of people within three standard deviations of the
- mean within_3_std = df_scaled[(df_scaled['friends_scaled'] <= 3) & (df_ scaled['friends_scaled'] >= -3)].shape[0] within_3_std / float(df_scaled.shape[0])
- # 1.0

- We can see that our data does seem to follow the Empirical rule.
- About 75% of the people are within a single standard deviation of the mean.
- About 92% of the people are within two standard deviations, and all of them are within three standard deviations.

Example – exam scores

- Let's say that we're measuring the scores of an exam and the scores generally have a bell-shaped normal distribution.
- The average of the exam was 84% and the standard deviation was 6%.
- We can say, with approximate certainty, that:
- • About 68% of the class scored between 78% and 90% because 78 is 6 units below 84, and 90 is 6 units above 84
- • If we were asked what percentage of the class scored between 72 and 96%, we would notice that 72 is 2 standard deviations below the mean, and 96 is 2 standard deviations above the mean, so, the Empirical rule tells us that about 95% of the class scored in that range.
- However, not all data is normally distributed, so, we can't always use the Empirical rule.
- We have another theorem that helps us analyze any kind of distribution. In the next chapter, we will go in depth about when we can assume the normal distribution. This is because many statistical tests and hypotheses require the underlying data to come from a normally distributed population