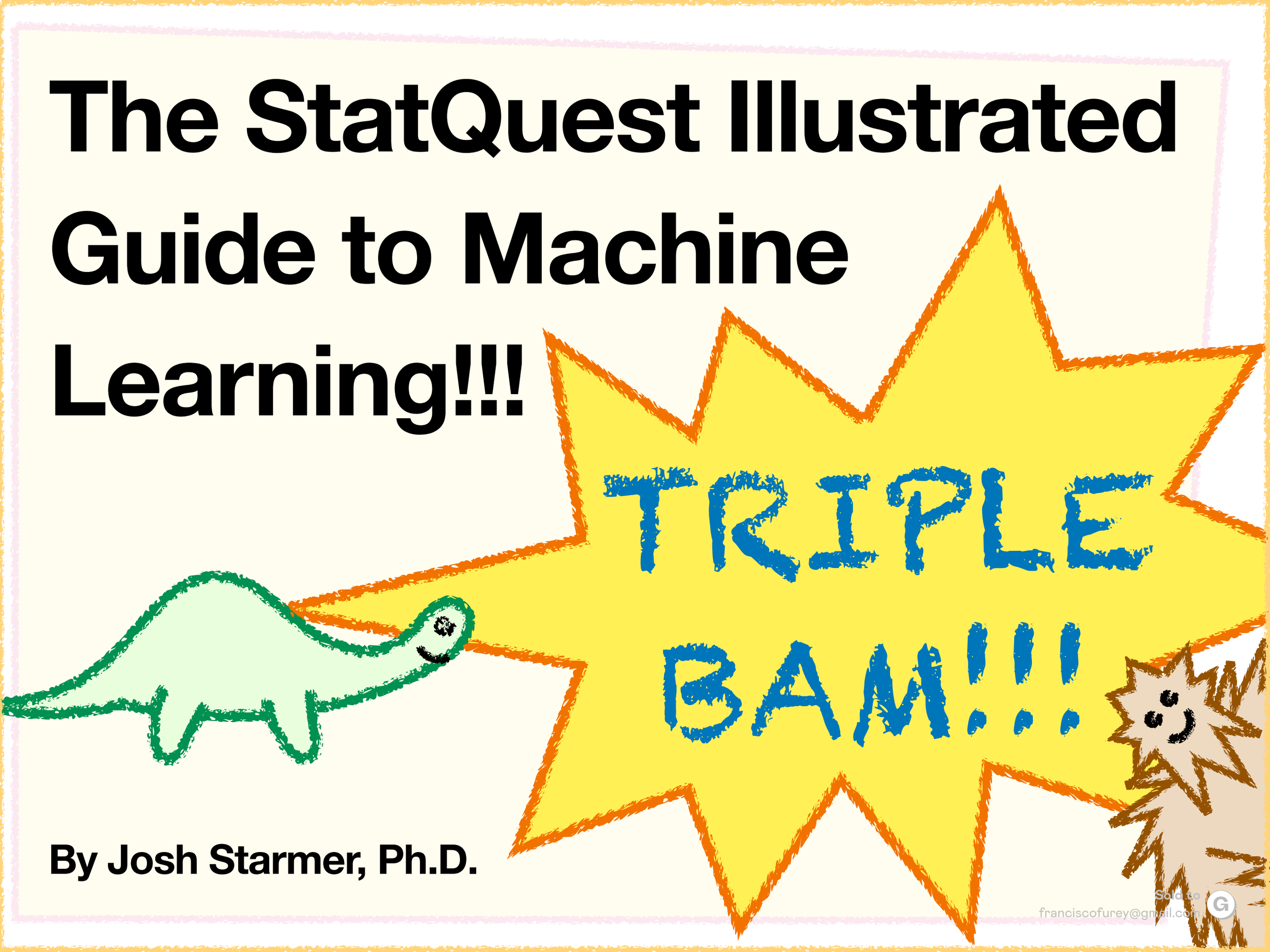
The StatQuest Illustrated Guide to Machine Learning!!!



By Josh Starmer, Ph.D.

www.statquest.org

Thank you for buying

The StatQuest Illustrated Guide to Machine Learning!!!

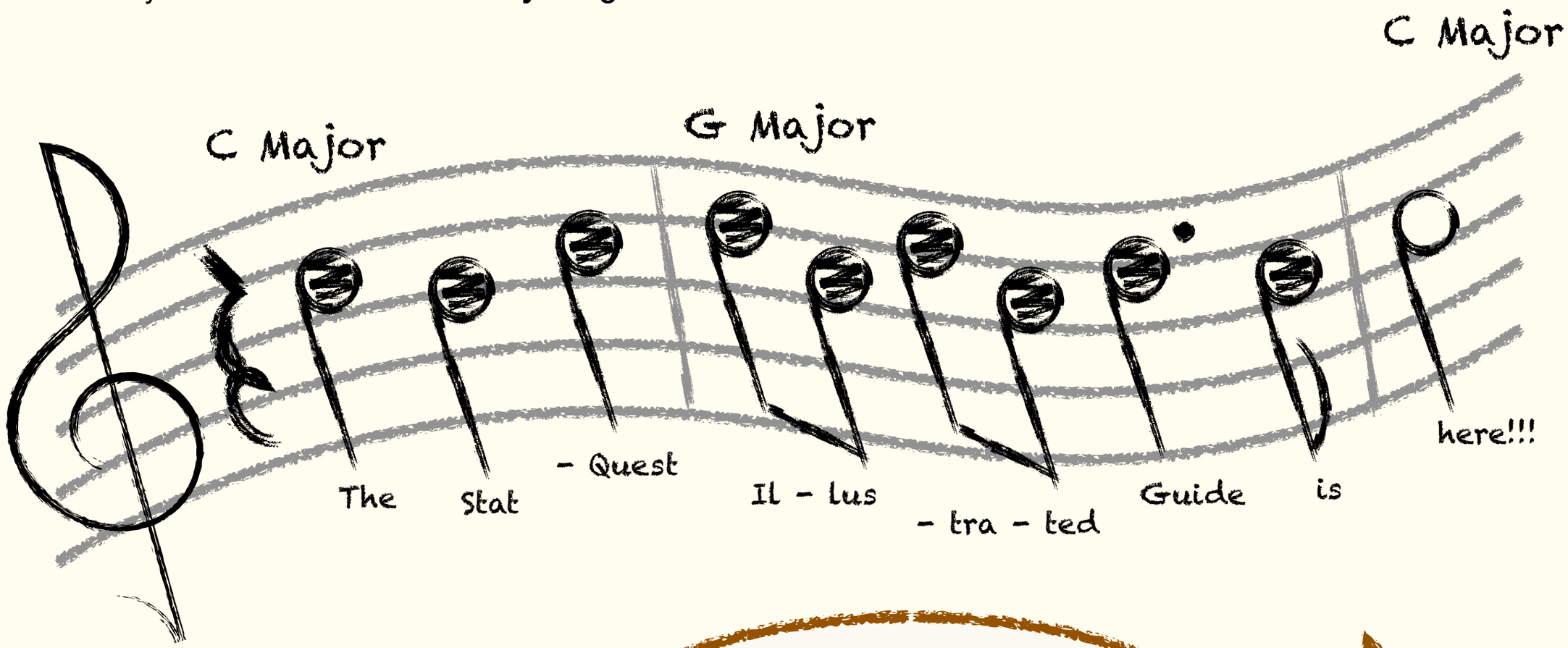
Every penny of your purchase goes to supporting **StatQuest** and helps create new videos, books, and webinars. Thanks to you, **StatQuest** videos and webinars are free on YouTube for everyone in the whole wide world to enjoy.

NOTE: If you're borrowing this copy of **The StatQuest Illustrated Guide to Machine Learning** from a friend, feel free to enjoy it. However, if you find it helpful, consider buying your own copy or supporting **StatQuest** however you can at https://statquest.org/support-statquest/.

Scan, click, or tap this QR code to visit StatQuest.org!!!

For **F** and **B**, for teaching me to think differently,
T and **D**, for making it possible for me to think differently,
and for **A**, for everything.

Since every **StatQuest** starts with a **Silly Song**…



Scan, click, or tap this QR code to hear the **Silly Song**!!!

Hello!!!

I'm Josh Starmer, and welcome to **The StatQuest Illustrated Guide to Machine Learning!!!** In this book, we'll talk about everything, from the very basics to advanced topics like **Neural Networks**. All concepts will be clearly illustrated, and we'll go through them one step at a time.

Table of Contents

How This Book Works

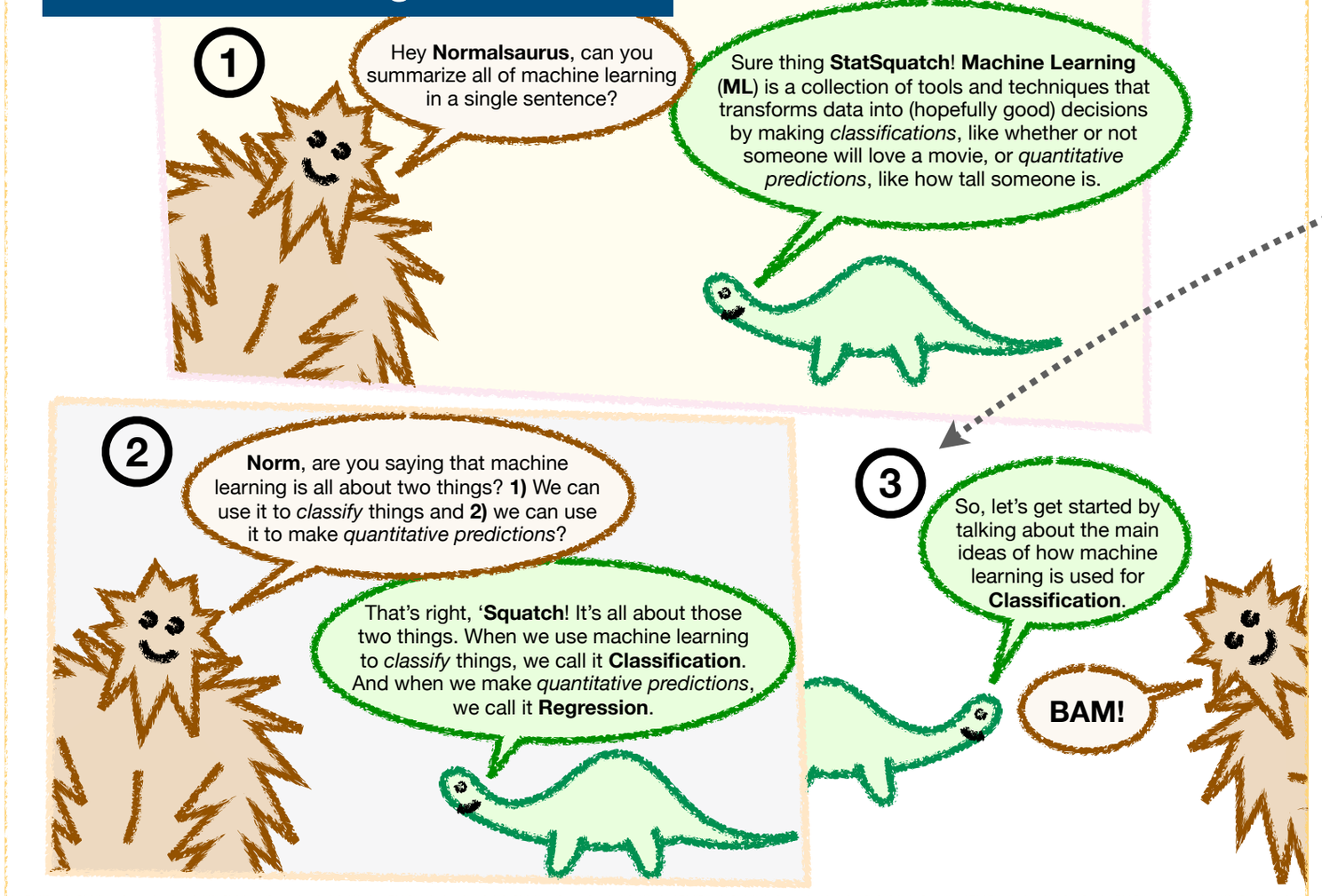
1 **NOTE:** Before we get started, let's talk a little bit about how this book works by looking at a sample page.

2 Each page starts with a header that tells you exactly what concept we're focusing on.

3 Throughout each page, you'll see circled numbers like these…

…that go from low to high, and all you have to do is follow them in order for each concept to be clearly explained.

4 **BAM!!** Now that you know how this book works, let's get started!!!



Chapter 01

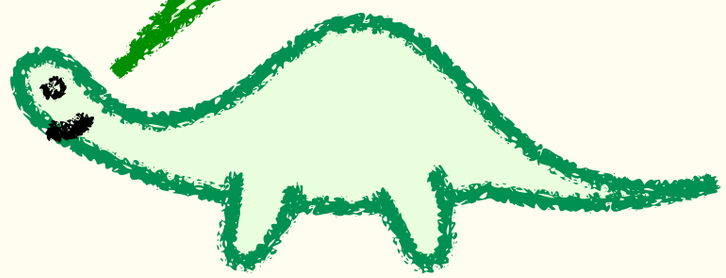
Fundamental Concepts in Machine Learning!!!

Machine Learning: Main Ideas

1

Hey **Normalsaurus**, can you summarize all of machine learning in a single sentence?

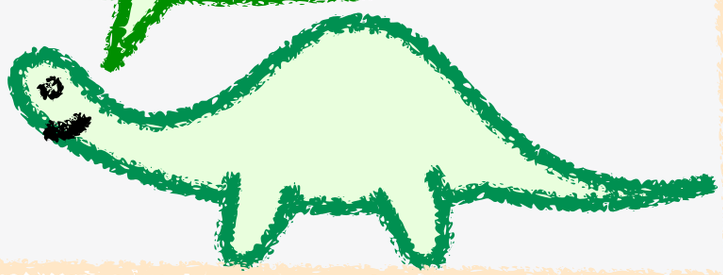
Sure thing **StatSquatch**! **Machine Learning** (**ML**) is a collection of tools and techniques that transforms data into (hopefully good) decisions by making *classifications*, like whether or not someone will love a movie, or *quantitative predictions*, like how tall someone is.



2

Norm, are you saying that machine learning is all about two things? **1)** We can use it to *classify* things and **2)** we can use it to make *quantitative predictions*?

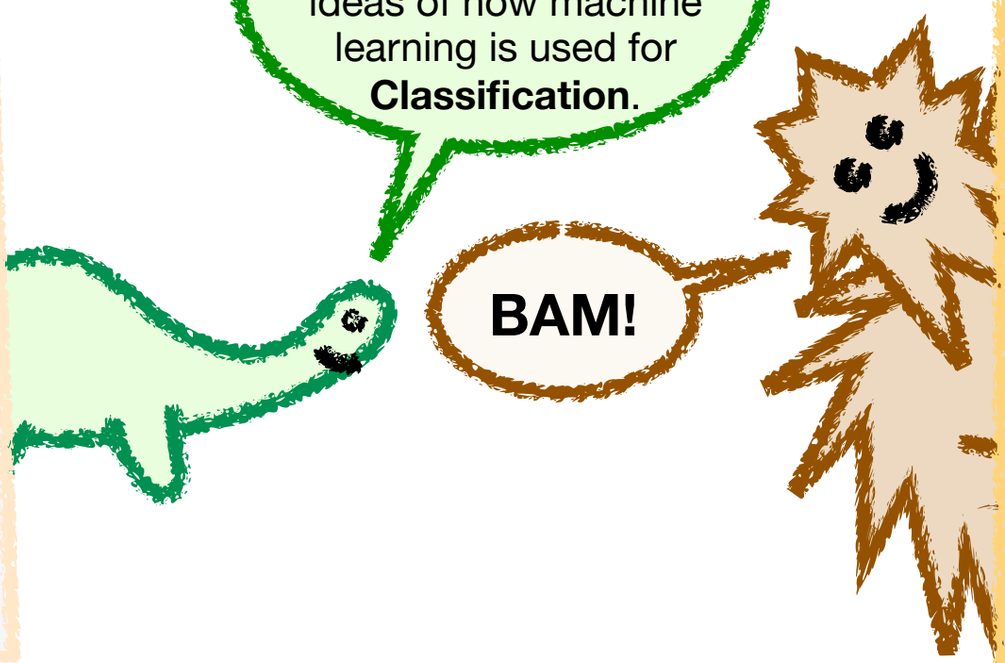
That's right, '**Squatch**! It's all about those two things. When we use machine learning to *classify* things, we call it **Classification**. And when we make *quantitative predictions*, we call it **Regression**.



3

So, let's get started by talking about the main ideas of how machine learning is used for **Classification**.

BAM!



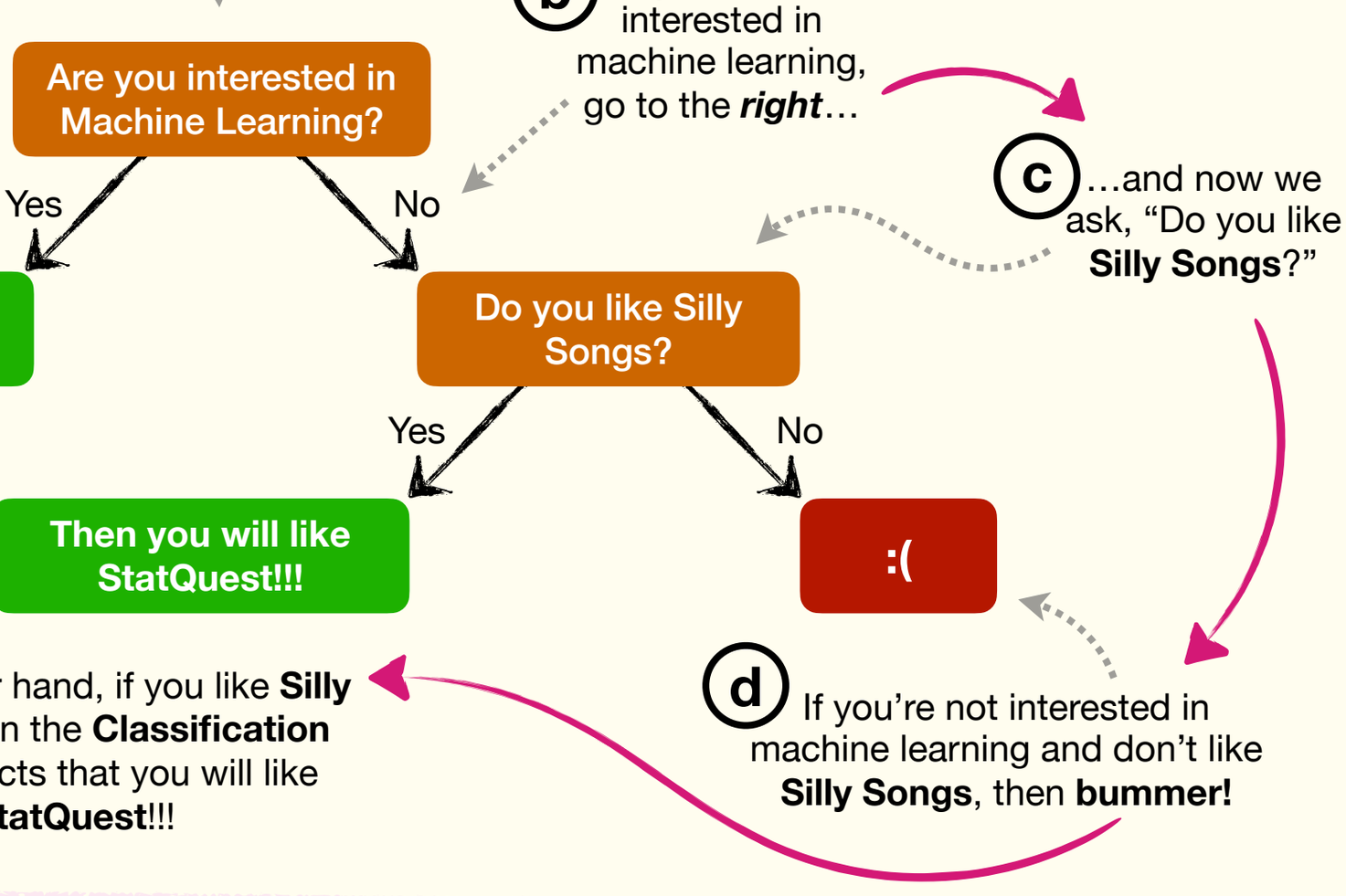
Machine Learning Classification: Main Ideas

1 **The Problem:** We have a big pile of data, and we want to use it to make *classifications*.

For example, we meet this person and want to **Classify** them as someone who will like **StatQuest** or not.

2 **A Solution:** We can use our data to build a **Classification Tree** (for details, see **Chapter 10**) to **Classify** a person as someone who will like **StatQuest** or not.

a Once the **Classification Tree** is built, we can use it to make **Classifications** by starting at the top and asking the question, "Are you interested in machine learning?"



b If you're *not* interested in machine learning, go to the ***right***…

c …and now we ask, "Do you like **Silly Songs**?"

d If you're not interested in machine learning and don't like **Silly Songs**, then **bummer!**

e On the other hand, if you like **Silly Songs**, then the **Classification Tree** predicts that you will like **StatQuest**!!!

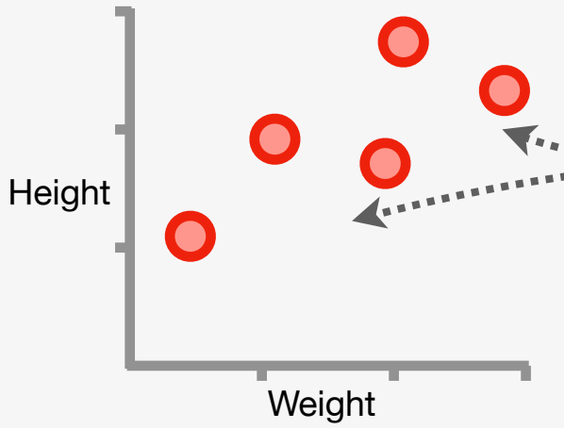
f And if you are interested in machine learning, then the **Classification Tree** predicts that you will like **StatQuest**!!!

g **BAM!!!**

Now let's learn the main ideas of how machine learning is used for **Regression**.

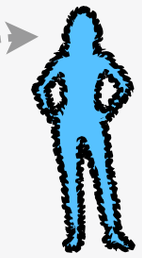
Machine Learning Regression: Main Ideas

1 **The Problem:** We have another pile of data, and we want to use it to make *quantitative predictions*, which means that we want to use machine learning to do **Regression**.

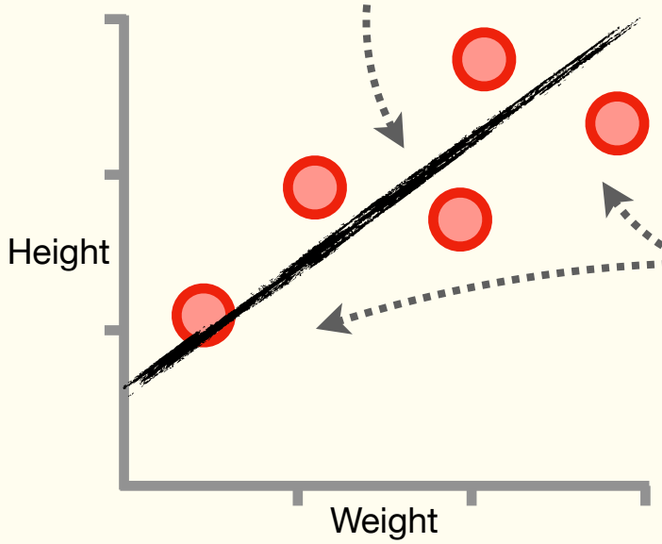


For example, here we measured the **Heights** and **Weights** of **5** different people. Because we can see a trend in the data—the larger the value for Weight, the taller the person—it seems reasonable to predict Height using Weight.

Thus, when someone new shows up and tells us their Weight, we would like to use that information to predict their Height.



2 **A Solution:** Using a method called **Linear Regression** (for details, see **Chapter 4**), we can fit a **line** to the original data we collected and use that line to make quantitative predictions.

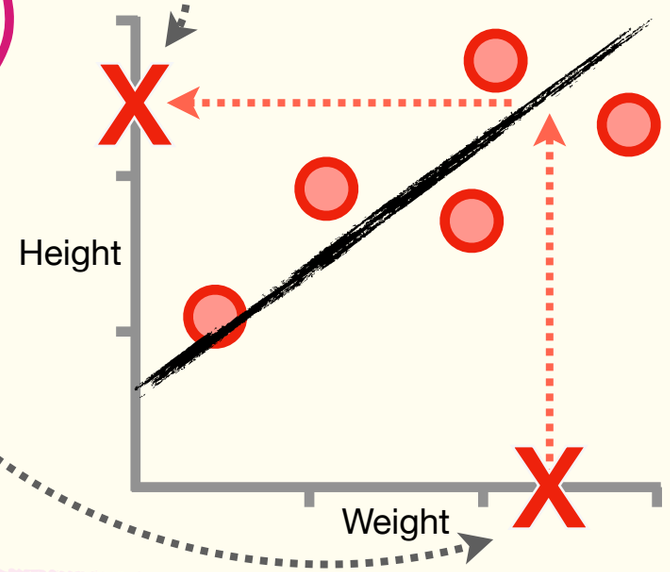


The **line**, which goes up as the value for Weight increases, summarizes the trend we saw in the data: as a person's Weight increases, generally speaking, so does their Height.

Now, if you told me that this was your Weight…

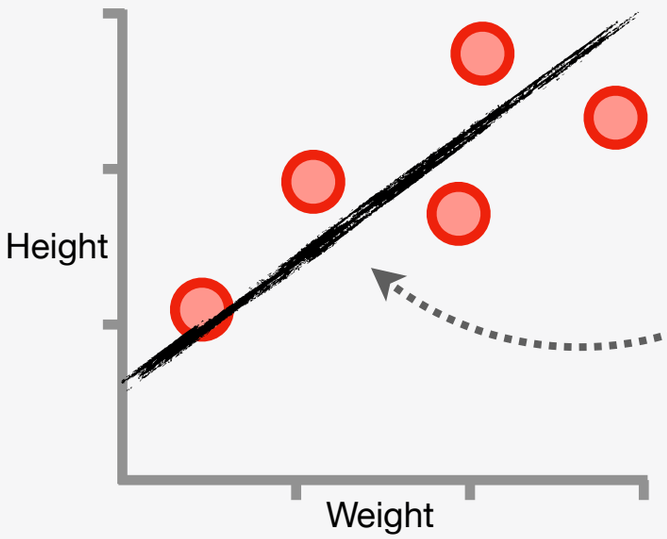
…then we could use the **line** to predict that this is your Height. **BAM!!!**

Because there are lots of machine learning methods to choose from, let's talk about how to pick the best one for our problem.



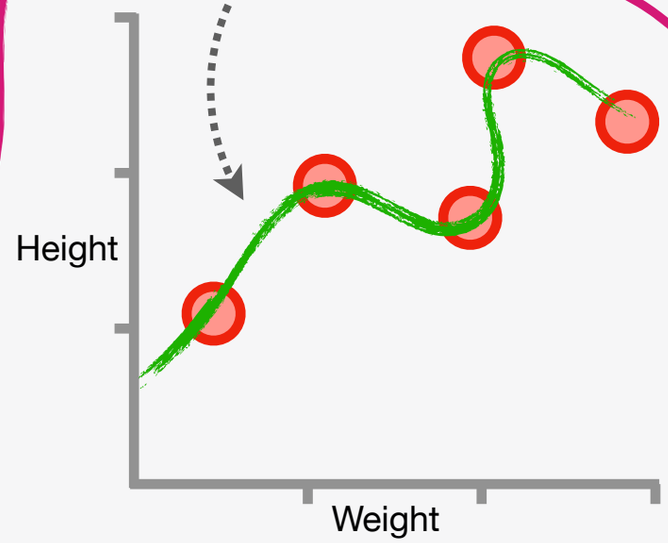
Comparing Machine Learning Methods: Main Ideas

1 **The Problem:** As we'll learn in this book, machine learning consists of a lot of different methods that allow us to make **Classifications** or **Quantitative Predictions**. How do we choose which one to use?



For example, we could use this **black line** to predict Height from Weight…

…or we could use this **green squiggle** to predict Height from Weight.

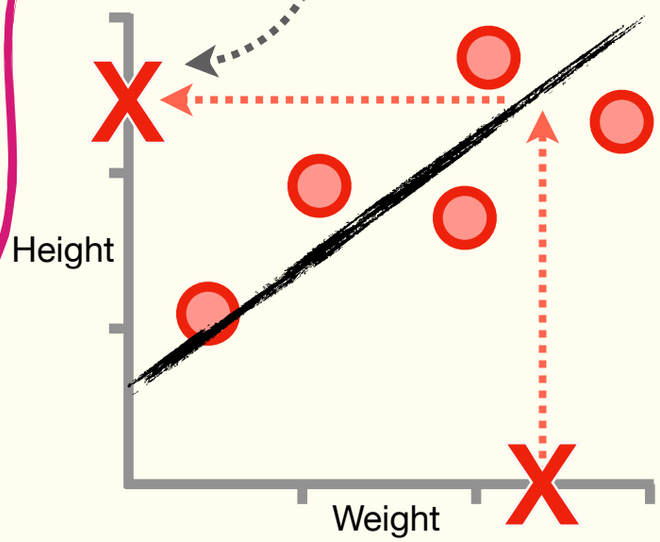


How do we decide to use the **black line** or the **green squiggle**?

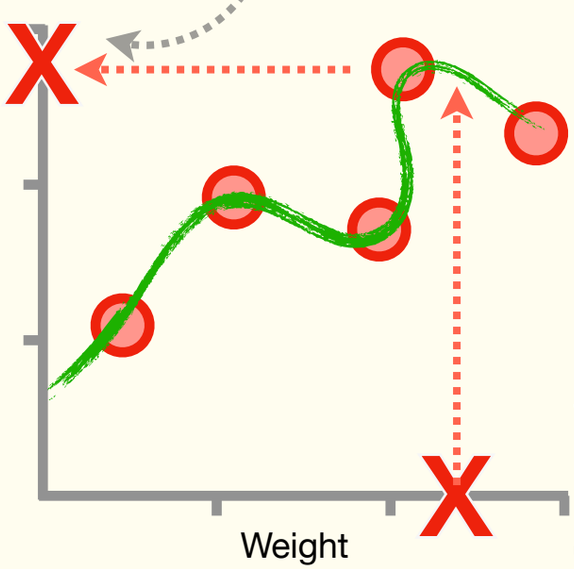
2 **A Solution:** In machine learning, deciding which method to use often means just trying it and seeing how well it performs.

For example, given this person's Weight…

…the **black line** predicts this Height.



In contrast, the **green squiggle** predicts that the person will be *slightly taller*.



We can compare those two predictions to the person's *actual* Height to determine the quality of each prediction.

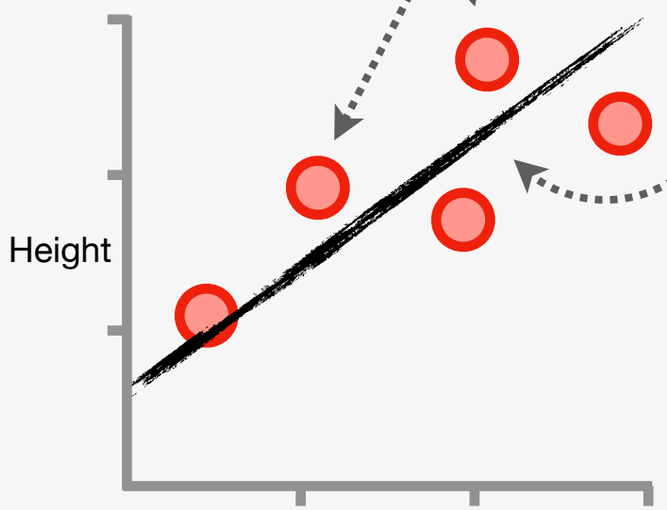
BAM!!!

Now that we understand the **Main Ideas** of how to compare machine learning methods, let's get a better sense of how we do this in practice.

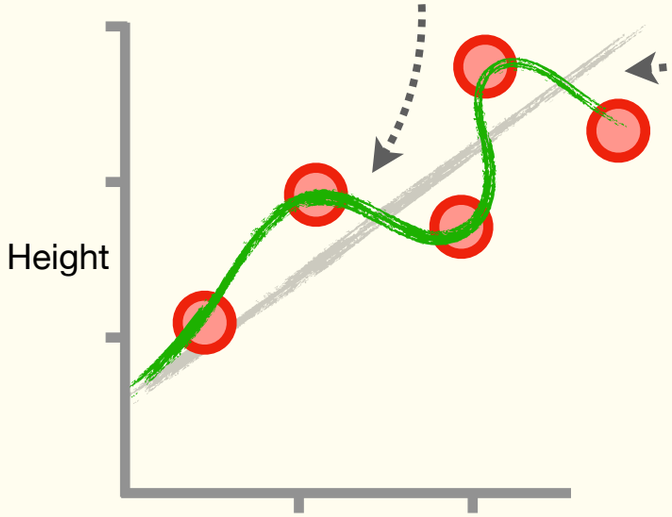
Comparing Machine Learning Methods: Intuition Part 1

1 The original data that we use to observe the trend and fit the **line** is called **Training Data**.

In other words, the **black line** is ***fit*** to the **Training Data**.

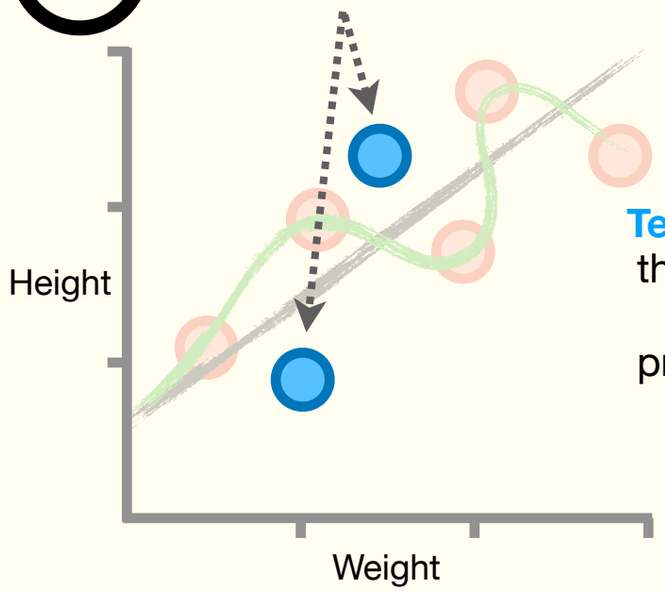


2 Alternatively, we could have fit a **green squiggle** to the **Training Data**.



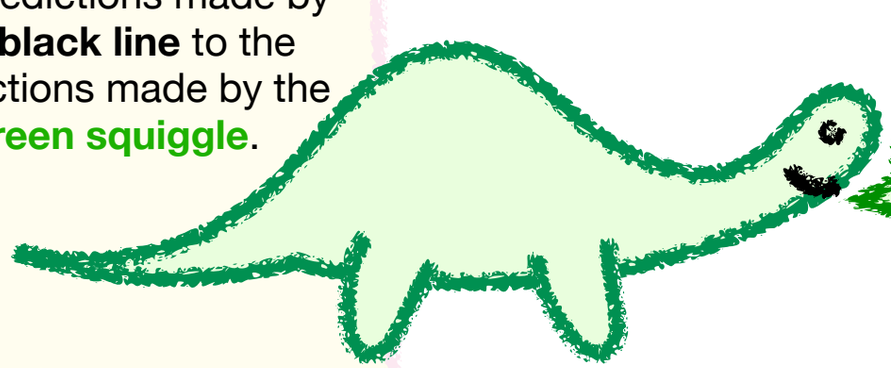
The **green squiggle** fits the **Training Data** better than the **black line**, but remember the goal of machine learning is to make ***predictions***, so we need a way to determine if the **black line** or the **green squiggle** makes better predictions.

3 So, we collect more data, called **Testing Data**…



…and we use the **Testing Data** to compare the predictions made by the **black line** to the predictions made by the **green squiggle**.

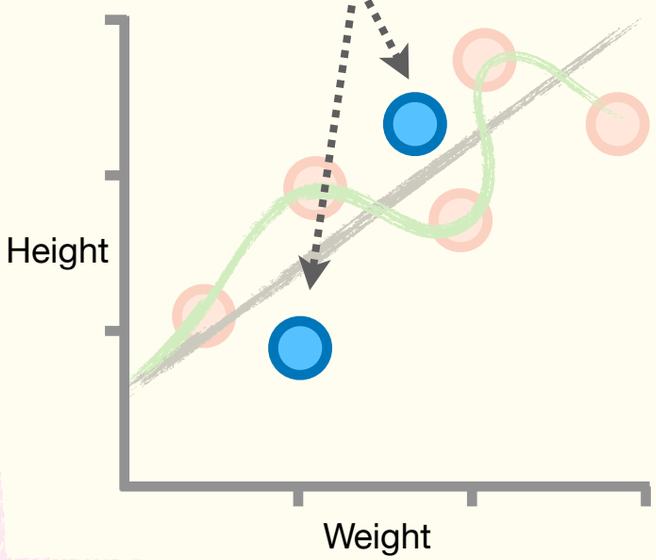
Hey **Normalsaurus**, don't you wish we'd get a warning when new terminology, like **Training Data** and **Testing Data**, is introduced?



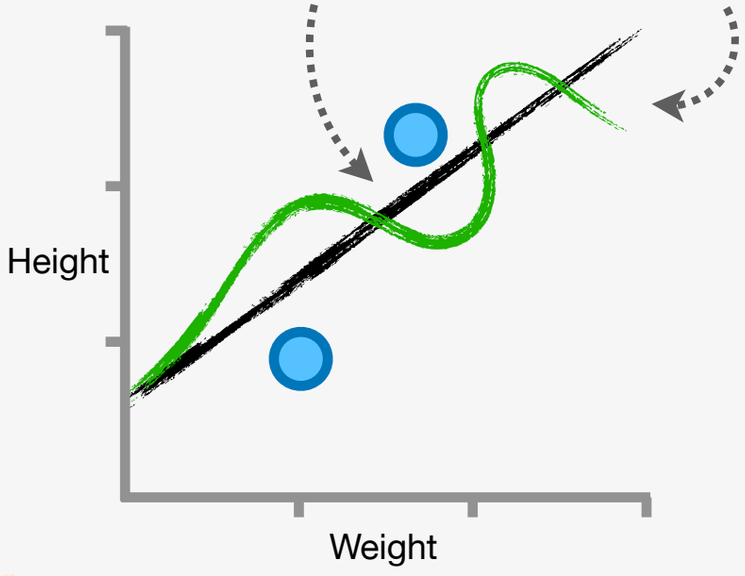
I sure would, **StatSquatch**! So, from this point forward, look for the dreaded **Terminology Alert!!!**

Comparing Machine Learning Methods: Intuition Part 2

4 Now, if these **blue dots** are the **Testing Data**…

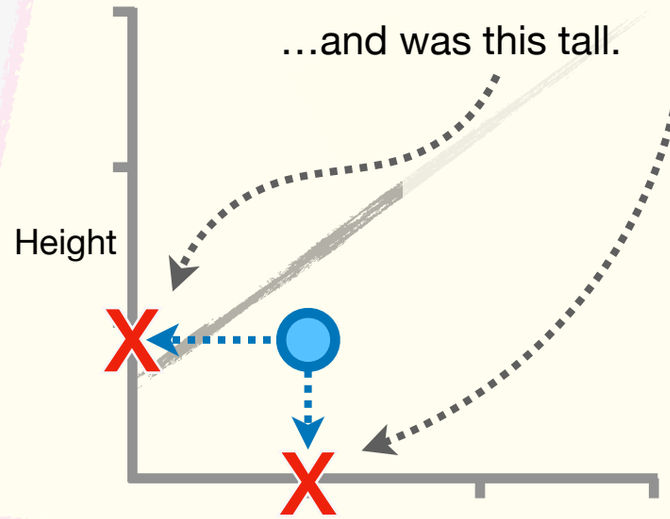


5 …then we can compare their **Observed** Heights to the Heights **Predicted** by the **black line** and the **green squiggle**.



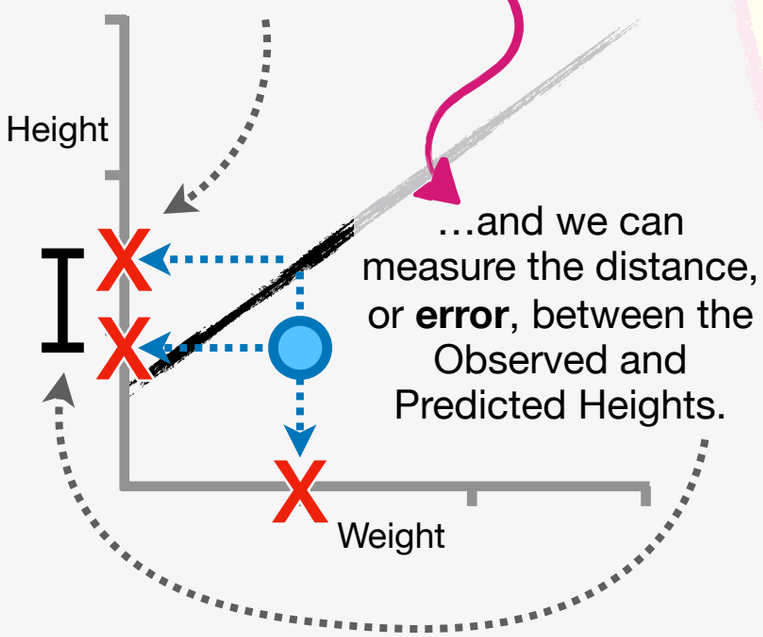
6 The first person in the **Testing Data** had this Weight…

…and was this tall.

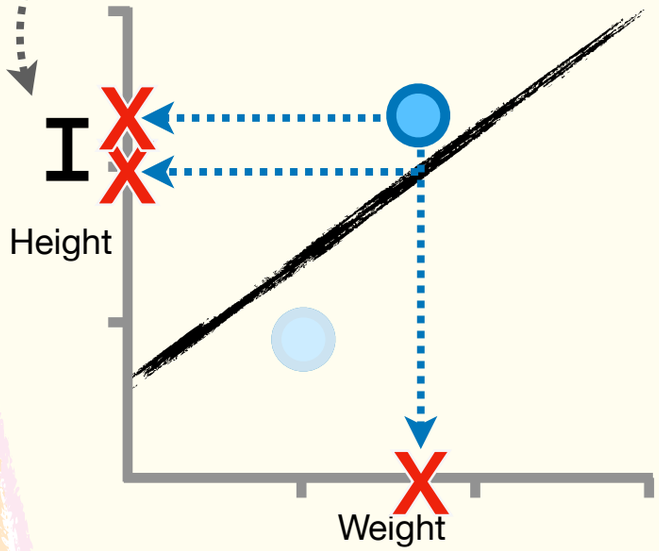


7 However, the **black line** predicts that they are taller…

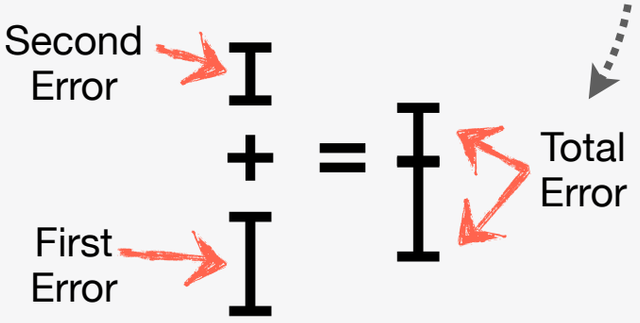
…and we can measure the distance, or **error**, between the Observed and Predicted Heights.



8 Likewise, we measure the **error** between the Observed and Predicted values for the second person in the **Testing Data**.

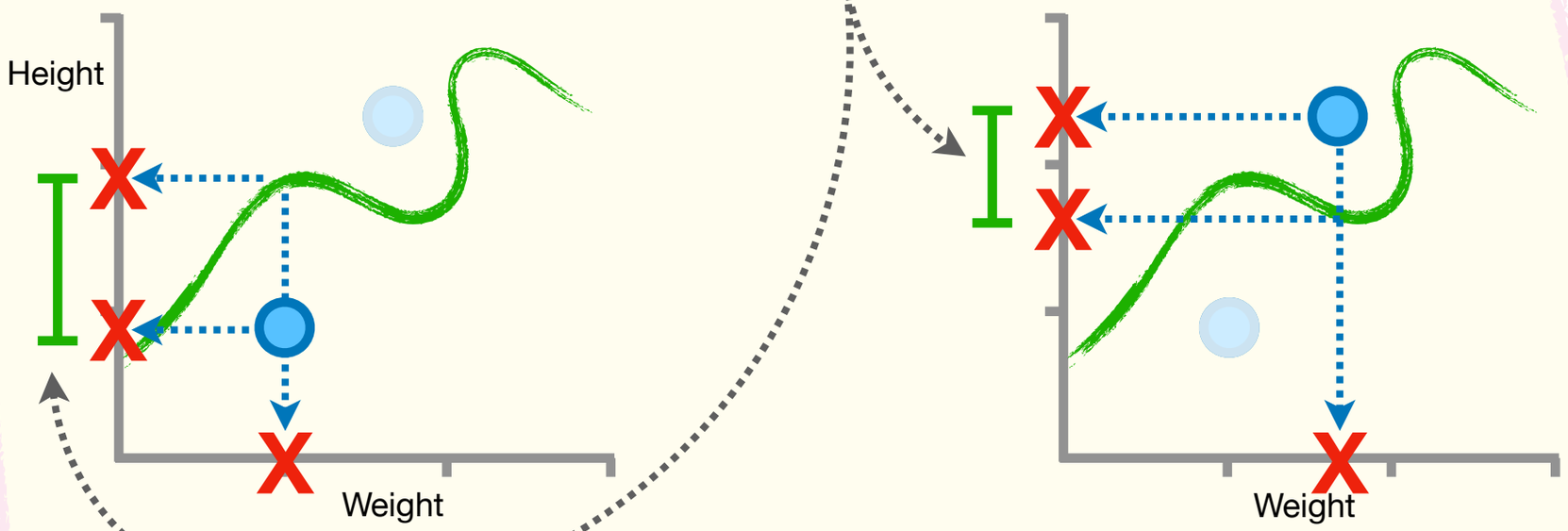


9 We can then add the two **errors** together to get a sense of how close the two Predictions are to the Observed values for the **black line**.

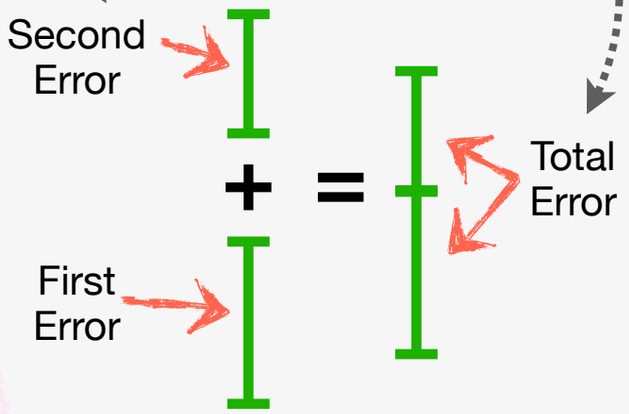


Comparing Machine Learning Methods: Intuition Part 3

10 Likewise, we can measure the distances, or **errors**, between the Observed Heights and the Heights Predicted by the **green squiggle**.



11 We can then add the two errors together to get a sense of how close the Predictions are to the Observed values for the **green squiggle**.



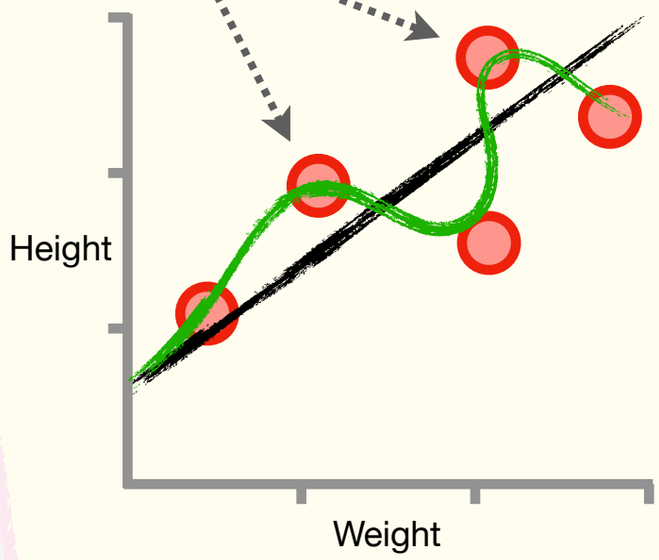
12 Now we can compare the predictions made by the **black line** to the predictions made by the **green squiggle** by comparing the sums of the **errors**.

Total Green Squiggle Errors

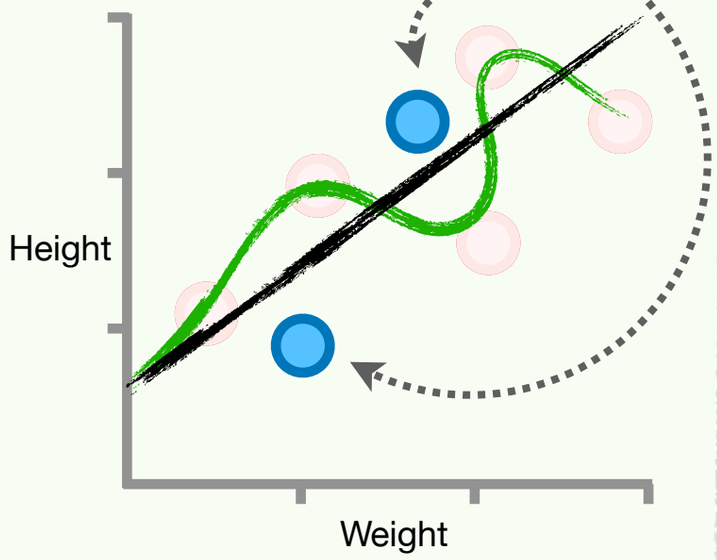
Total Black Line Errors

And we see that the sum of the **errors** for the **black line** is shorter, suggesting that it did a better job making predictions.

13 In other words, even though the **green squiggle** fit the **Training Data** way better than the **black line**…

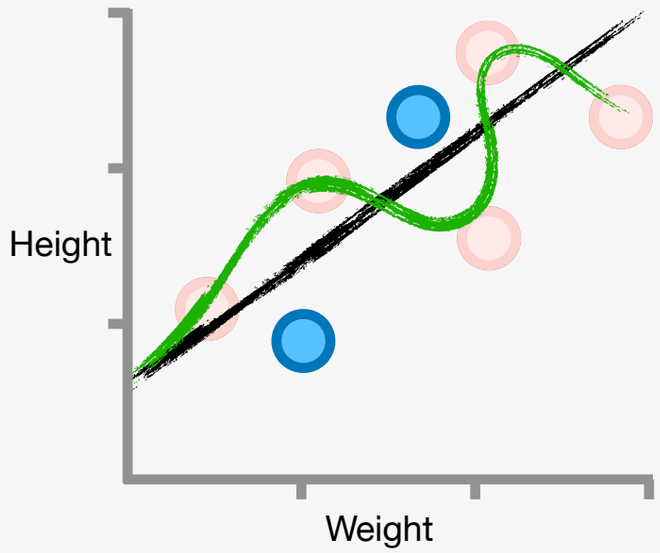


14 …the **black line** did a better job predicting **Height** with the **Testing Data**.

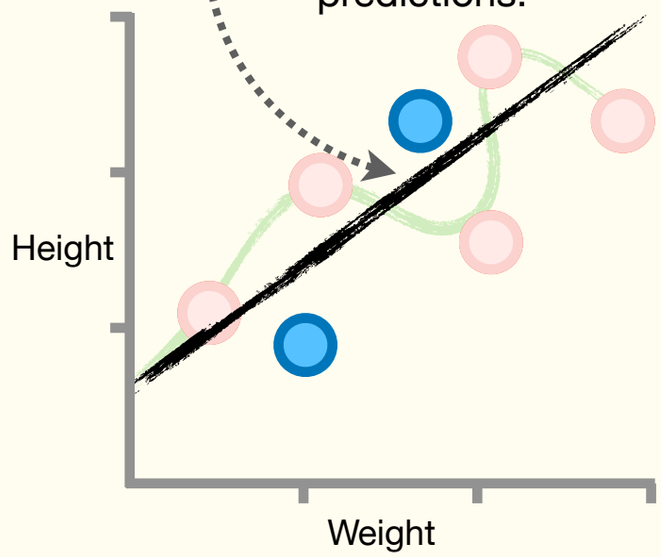


Comparing Machine Learning Methods: Intuition Part 4

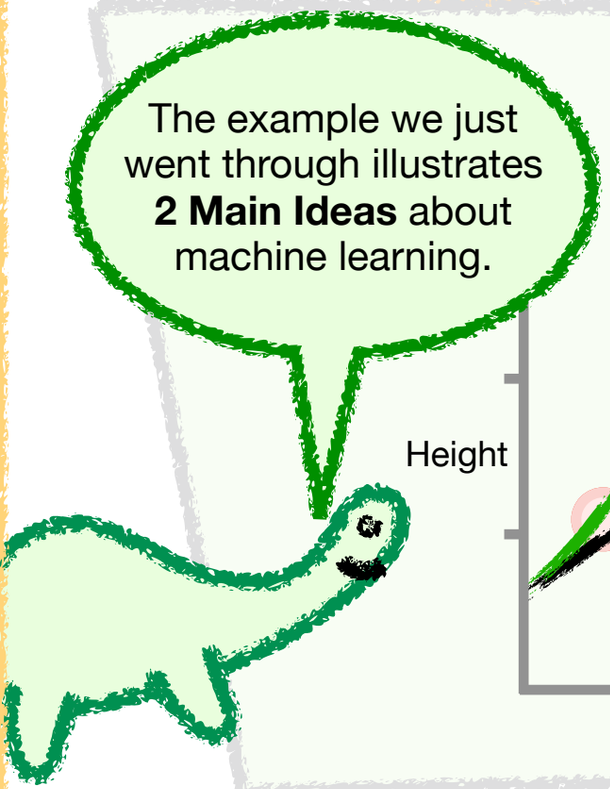
15 So, if we had to choose between using the **black line** or the **green squiggle** to make predictions…

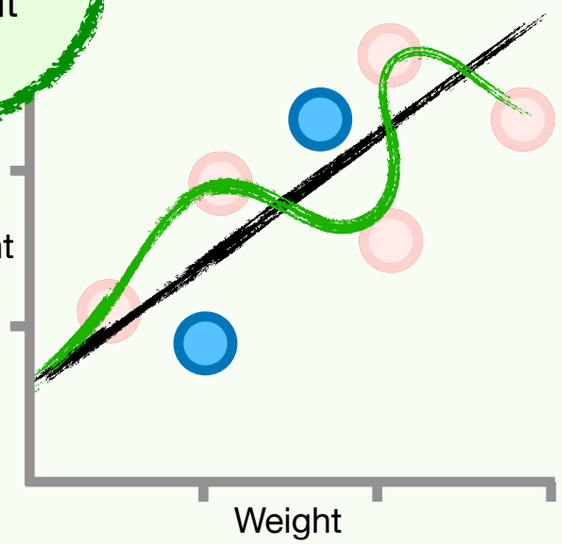


16 …we would choose the **black line** because it makes better predictions.



BAM!!!





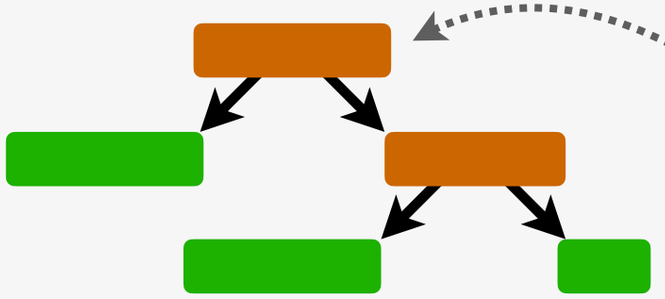
First, we use **Testing Data** to evaluate machine learning methods.

Second, just because a machine learning method fits the **Training Data** well, it doesn't mean it will perform well with the **Testing Data**.

TERMINOLOGY ALERT!!!

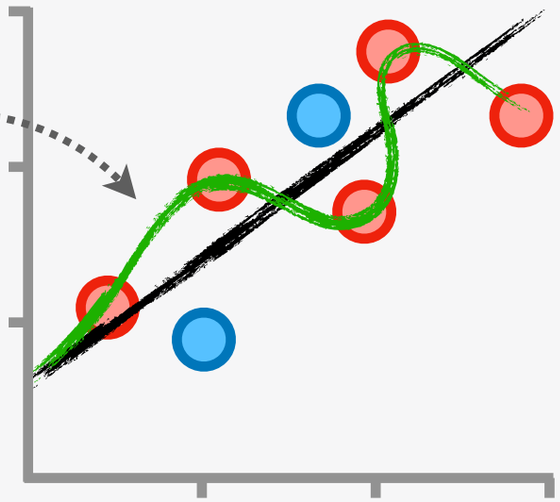
When a machine learning method fits the **Training Data** *really well* but makes *poor predictions*, we say that it is **Overfit** to the **Training Data**. **Overfitting** a machine learning method is related to something called the **Bias-Variance Tradeoff**, and we'll talk more about that later.

The Main Ideas of Machine Learning: Summary



Now, you may be wondering why we started this book with a super simple **Decision Tree**…

…and a simple **black line** and a silly **green squiggle** instead of a…



…**Deep Learning Convolutional Neural Network**
or a
[insert newest, fanciest machine learning method here].

There are lots of cool machine learning methods. In this book, we'll learn about…

Regression

Logistic Regression

Naive Bayes

Classification Trees

Regression Trees

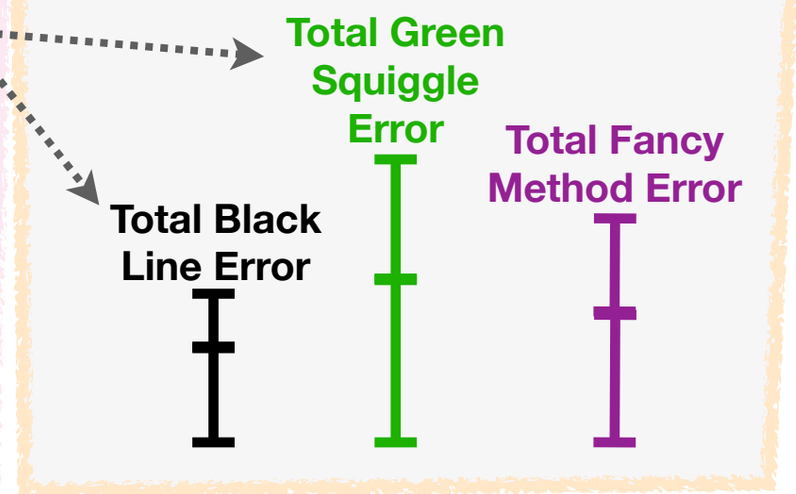
Support Vector Machines

Neural Networks

There are tons of fancy-sounding machine learning methods, like **Deep Learning Convolutional Neural Networks**, and each year something new and exciting comes along, but regardless of what you use, the most important thing is how it performs with the **Testing Data**.

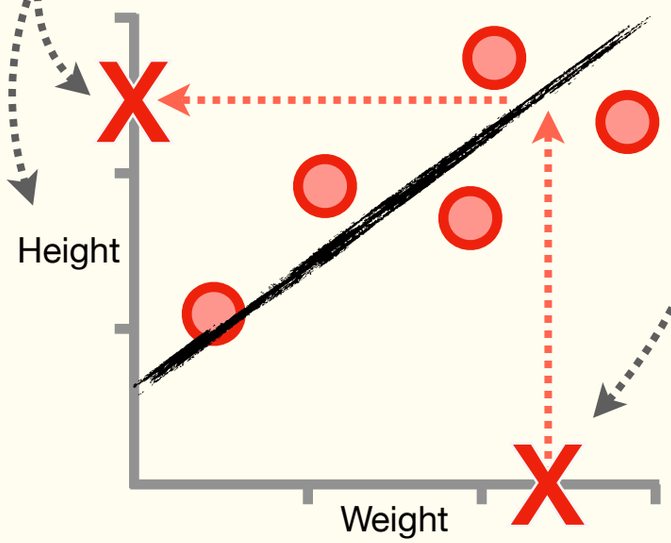
BAM!!!

Now that we understand some of the main ideas of machine learning, let's learn some fancy terminology so we can sound smart when we talk about this stuff at dance parties.



Terminology Alert!!! Independent and Dependent Variables

1 So far, we've been predicting Height… …from Weight measurements… …and the data have all been displayed on a nice graph. However, we can also organize the data in a nice table.



Now, regardless of whether we look at the data in the graph or in the table, we can see that Weight *varies* from person to person, and thus, Weight is called a **Variable**.

Likewise, Height *varies* from person to person, so Height is also called a **Variable**.

Weight	Height
0.4	1.1
1.2	1.9
1.9	1.7
2.0	2.8
2.8	2.3

2 That being said, we can be more specific about the types of **Variables** that Height and Weight represent.

Because our Height predictions *depend* on Weight measurements, we call Height a ***Dependent Variable***.

In contrast, because we're not predicting Weight, and thus, Weight does not depend on Height, we call Weight an ***Independent Variable***. Alternatively, Weight can be called a ***Feature***.

3 So far in our examples, we have only used Weight, a single **Independent Variable**, or **Feature**, to predict Height. However, it's very common to use multiple **Independent Variables**, or **Features**, to make predictions. For example, we might use Weight, Shoe Size and Favorite Color to predict Height.

Weight	Shoe Size	Favorite Color	Height
0.4	3	Blue	1.1
1.2	3.5	Green	1.9
1.9	4	Green	1.7
2.0	4	Pink	2.8
2.8	4.5	Blue	2.3

Bam.

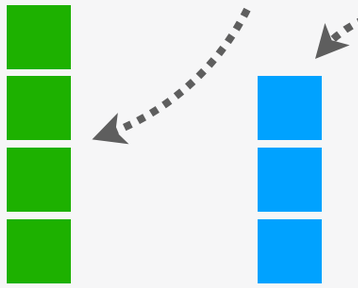
Now, as we can see in the table, Weight is a *numeric measurement* and Favorite Color is a *discrete category*, so we have different types of data. Read on to learn more about these types!!!

Terminology Alert!!! Discrete and Continuous Data

1 Discrete Data…

…is ***countable*** and only takes *specific* values.

2 For example, we can count the number of people that love the color **green** or love the color **blue**.

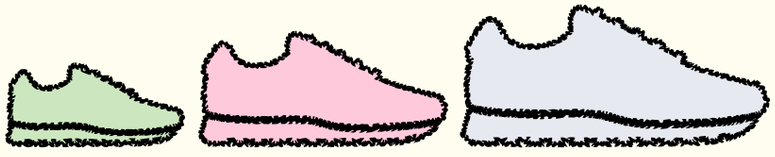


4 people love **green**

3 people love **blue**

Because we are counting individual people, and the totals can only be whole numbers, the data are **Discrete**.

3 American shoe sizes are **Discrete** because even though there are half sizes, like **8 1/2**, shoes sizes are never **8 7/36** or **9 5/18**.

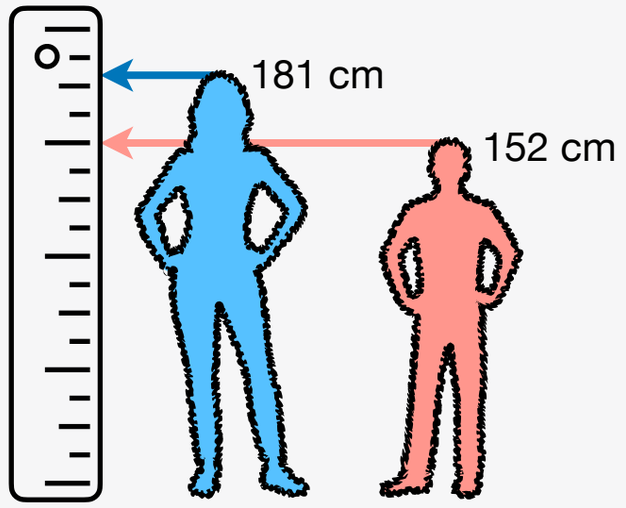


4 Rankings and other orderings are also **Discrete**. There is no award for coming in **1.68** place. Total bummer!

5 Continuous Data…

…is ***measurable*** and can take *any* numeric value within a range.

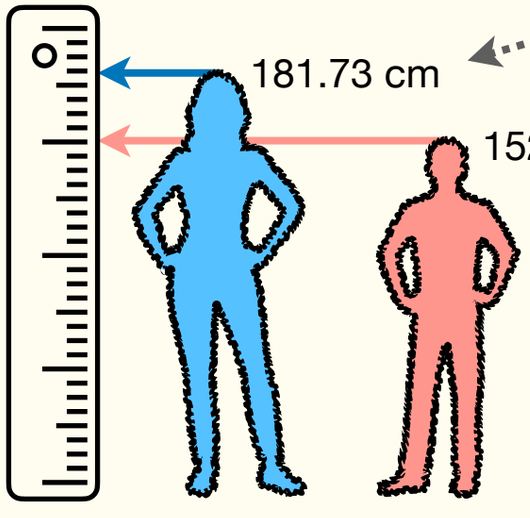
6 For example, Height measurements are **Continuous** data.



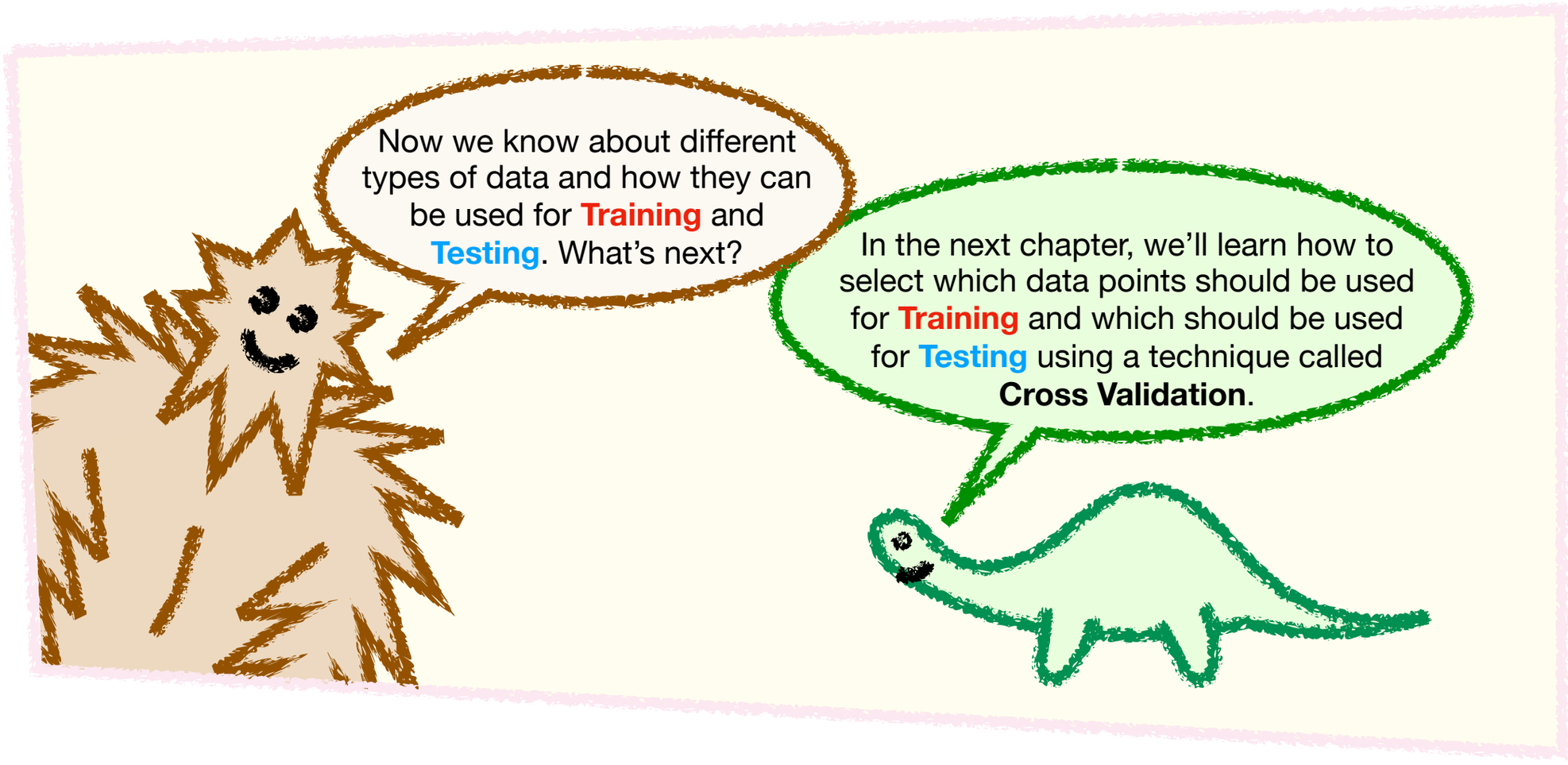
Height measurements can be any number between **0** and the height of the tallest person on the planet.

7 **NOTE:** If we get a more precise ruler…

…then the measurements get more precise.



So the precision of **Continuous** measurements is only limited by the tools we use.


Now we know about different types of data and how they can be used for Training and Testing. What's next?
In the next chapter, we'll learn how to select which data points should be used for Training and which should be used for Testing using a technique called Cross Validation.

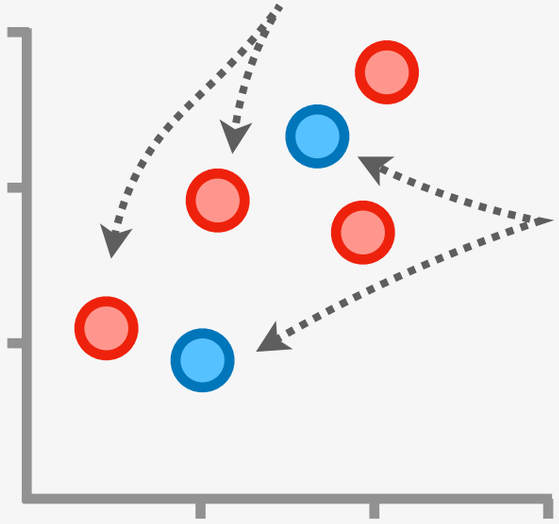
Chapter 02

Cross Validation!!!

Cross Validation: Main Ideas

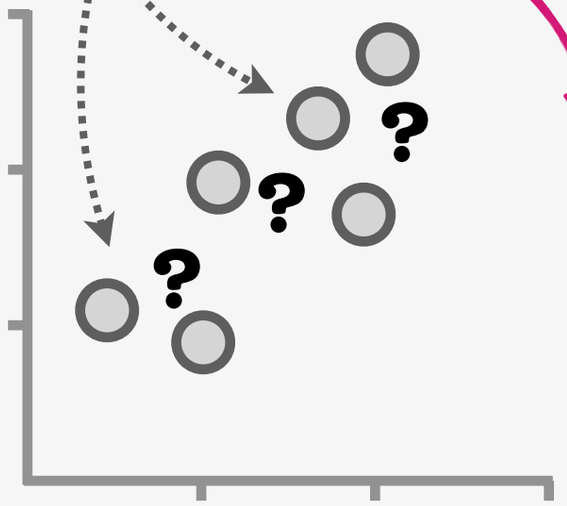
1

The Problem: So far, we've simply been told which points are the **Training Data**…



…and which points are the **Testing Data**.

However, usually no one tells us what is for **Training** and what is for **Testing**.



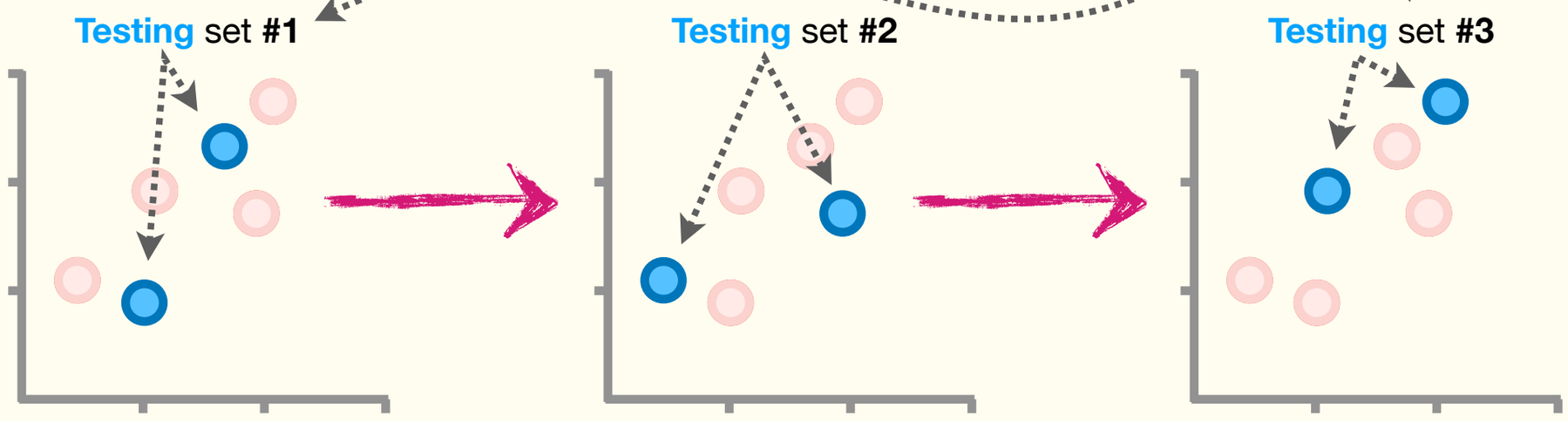
How do we pick the best points for **Training** and the best points for **Testing**?

2

A Solution: When we're not told which data should be used for **Training** and for **Testing**, we can use **Cross Validation** to figure out which is which in an unbiased way.

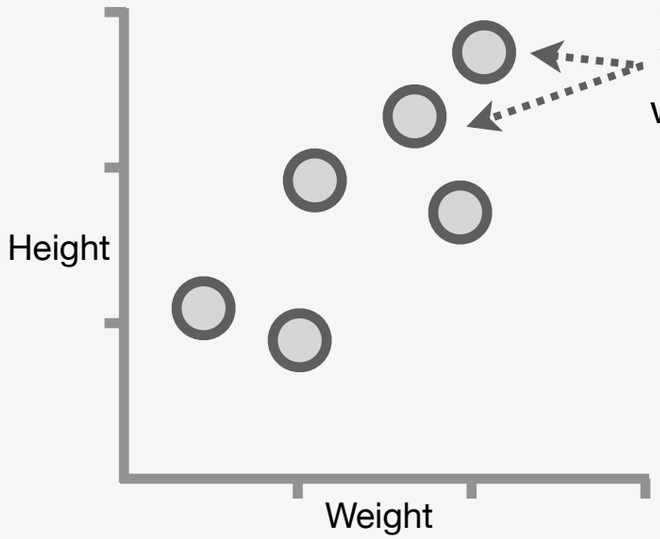
Rather than worry too much about which specific points are best for **Training** and best for **Testing**, **Cross Validation** uses *all* points for both in an *iterative* way, meaning that we use them in steps.

BAM!!!



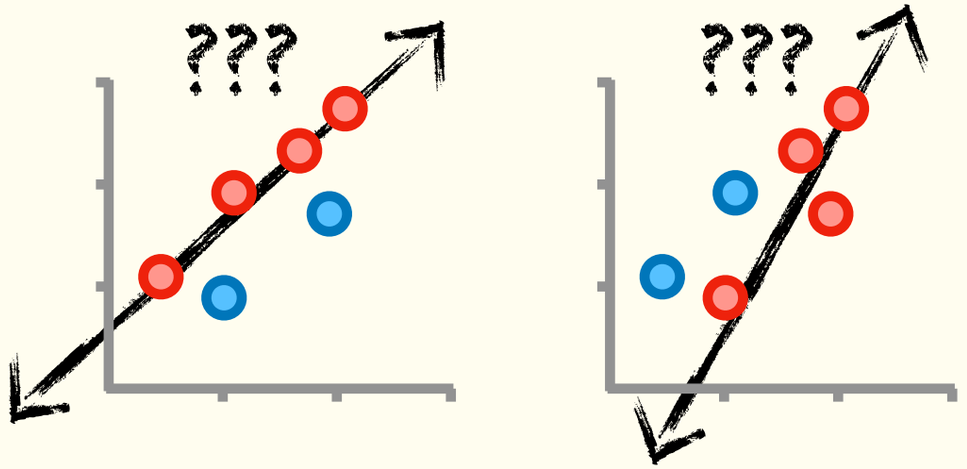
Cross Validation: Details Part 1

1 Imagine we gathered these **6** pairs of Weight and Height measurements…

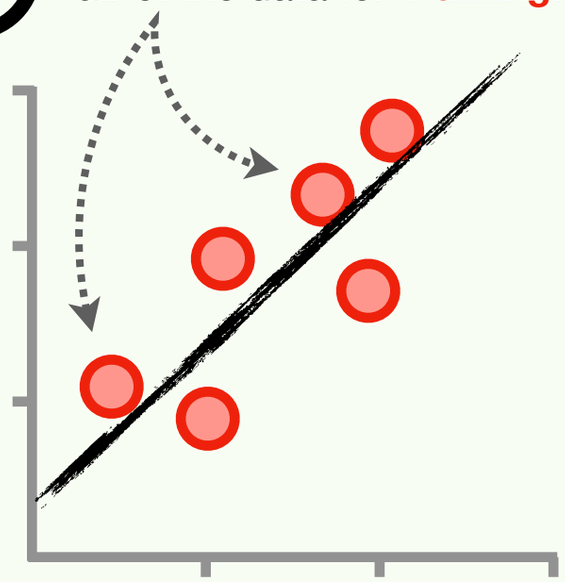


…and because we see a trend that people with larger Weights tend to be taller, we want to use Weight to predict Height…

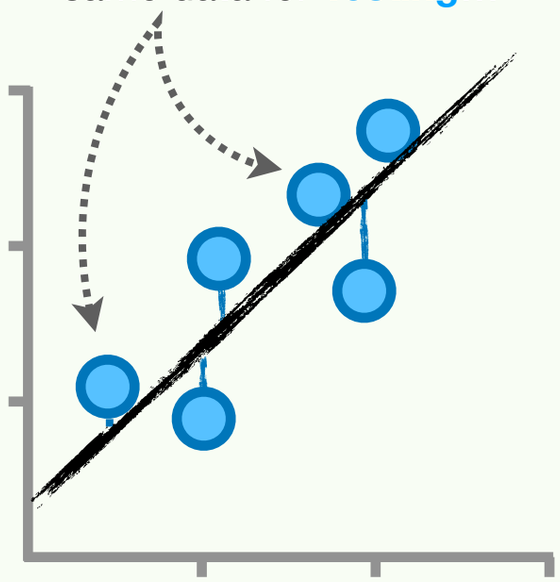
2 …so we decide to fit a **line** to the data with **Linear Regression** (for details, see **Chapter 4**). However, we don't know which points to use for **Training** and which to use for **Testing**.



3 A terrible idea would be to use ***all*** of the data for **Training**…



…and then reuse the exact same data for **Testing**…



…because the only way to determine if a machine learning method has been **Overfit** to the **Training Data** is to try it on new data that it hasn't seen before.

TERMINOLOGY ALERT!!!

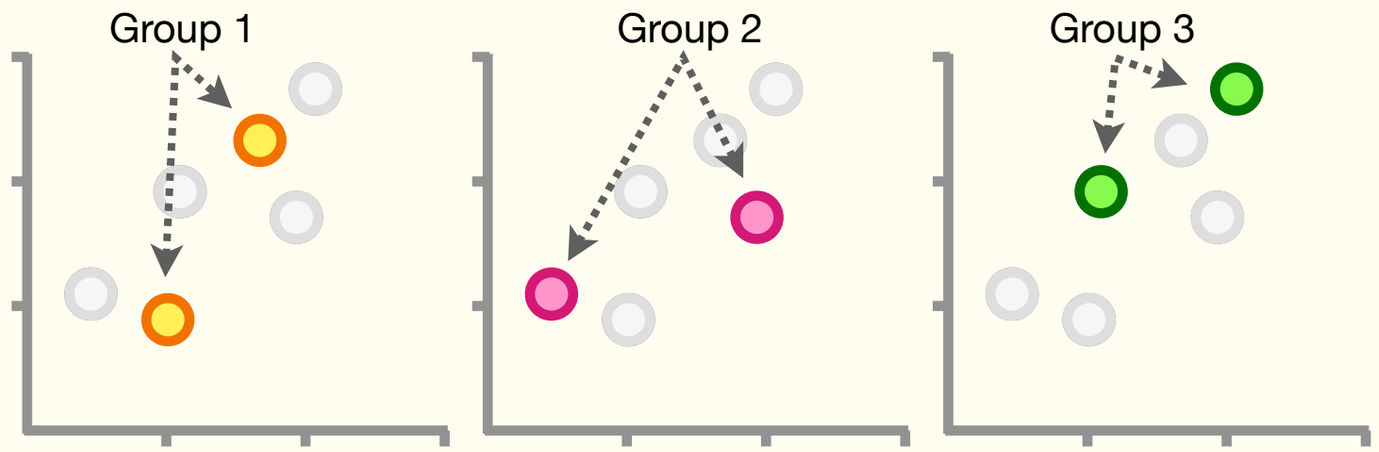
Reusing the same data for **Training** and **Testing** is called **Data Leakage**, and it usually results in you believing the machine learning method will perform better than it does because it is **Overfit**.

Cross Validation: Details Part 2

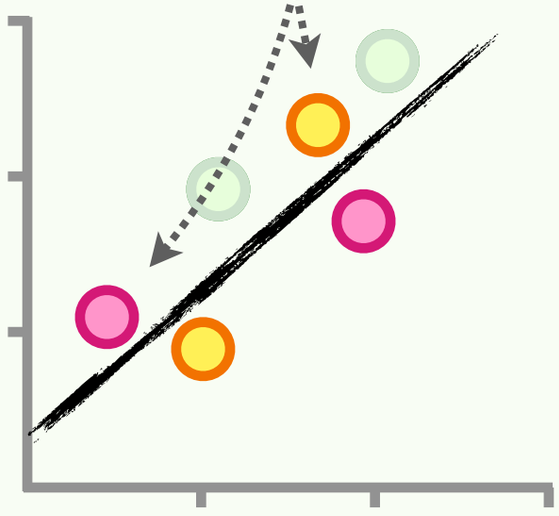
4 A slightly better idea is to randomly select some data to use *only* for **Testing** and select and use the rest for **Training**.

This avoids **Data Leakage**, but how do we know we selected the best data for **Testing**?

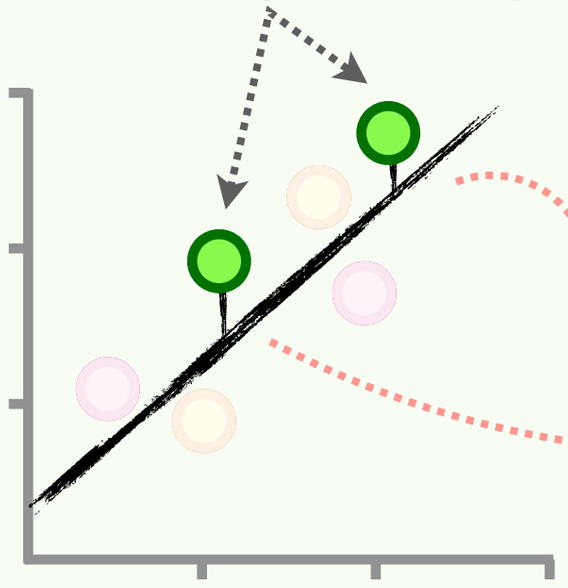
5 **Cross Validation** solves the problem of not knowing which points are the best for **Testing** by using them *all* in an *iterative* way. The first step is to randomly assign the data to different groups. In this example, we divide the data into **3** groups, where each group consists of **2** points.



6 Now, in the first **Cross Validation** iteration, we'll use **Groups 1** and **2** for **Training**…



…and **Group 3** for **Testing**.



7 Then, just like before, we can measure the errors for each point in the **Testing Data**…

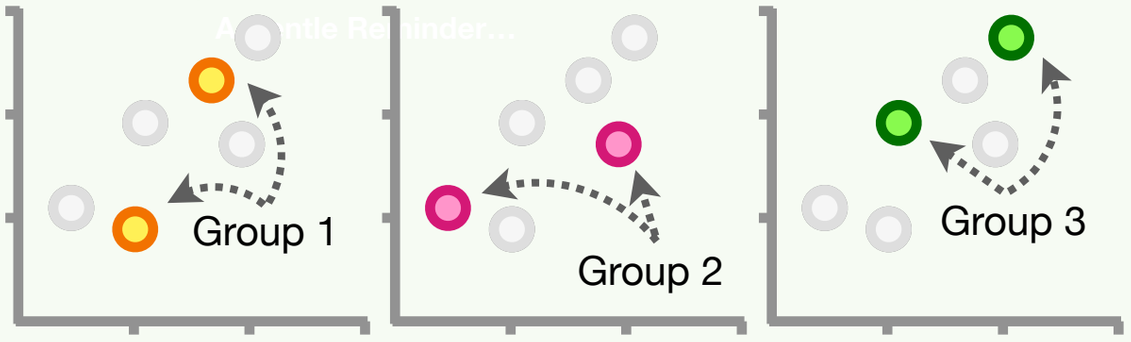
Iteration #1: Black Line Errors

…however, unlike before, we don't stop here. Instead, we keep iterating so that **Groups 1** and **2** can each get a turn to be used for **Testing**.

Cross Validation: Details Part 3

8 Because we have **3** groups of data points, we'll do **3** iterations, which ensures that each group is used for **Testing**. The number of iterations are also called **Folds**, so this is called **3-Fold Cross Validation**.

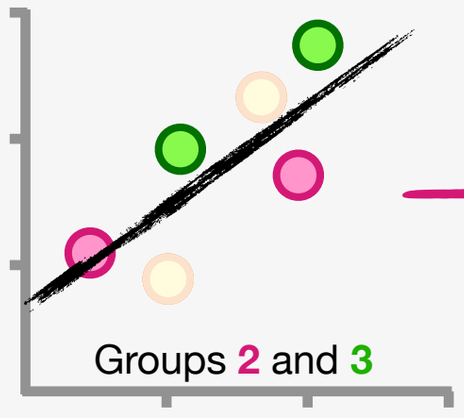
Gentle Reminder: These are the original **3** groups.



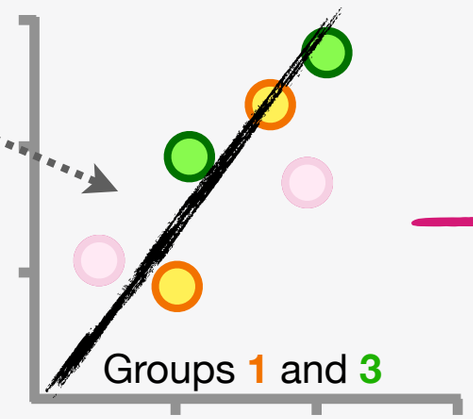
9 So, these are the **3** iterations of **Training**…

NOTE: Because each iteration uses a different combination of data for **Training**, each iteration results in a slightly different fitted **line**.

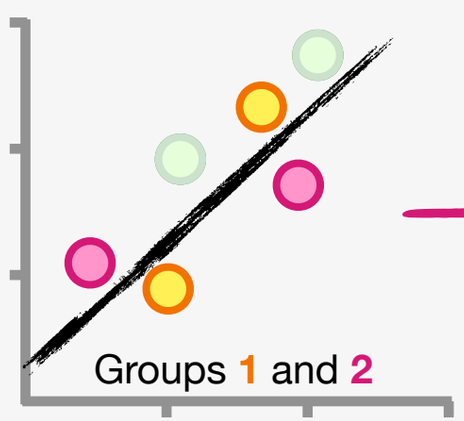
Iteration #1

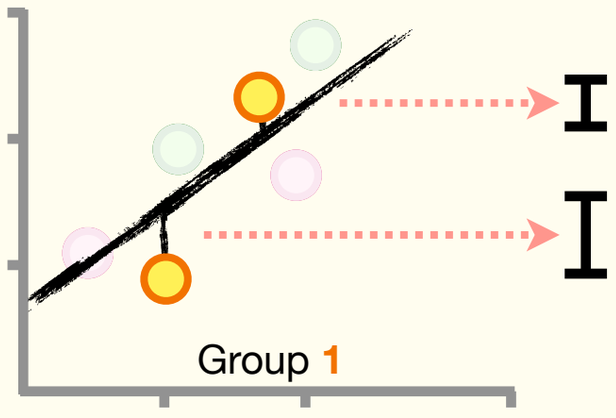


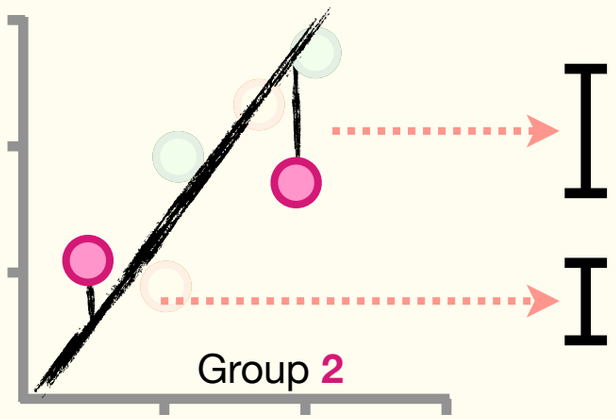
Iteration #2

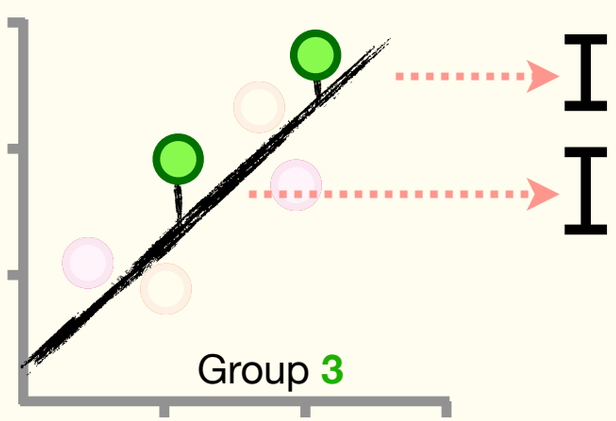


Iteration #3









10 …and these are the **3** iterations of **Testing**.

A different fitted line combined with using different data for **Testing** results in each iteration giving us different prediction errors.

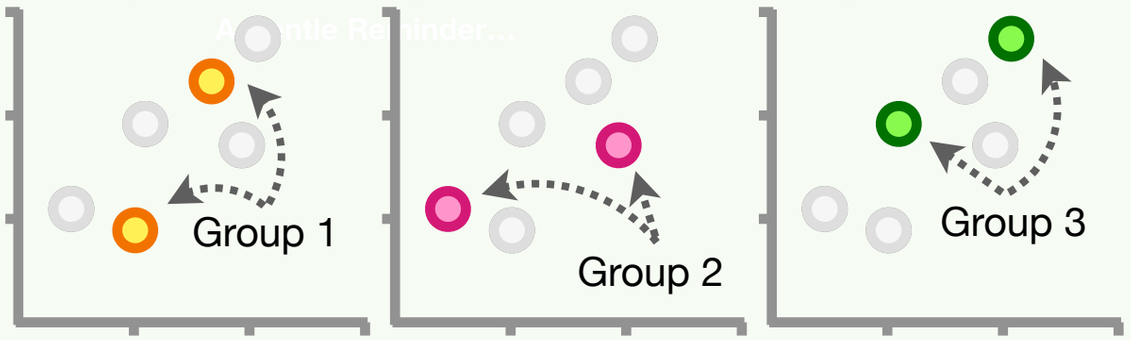
We can average these errors to get a general sense of how well this model will perform with future data…

…or we can compare these errors to errors made by another method.

Cross Validation: Details Part 4

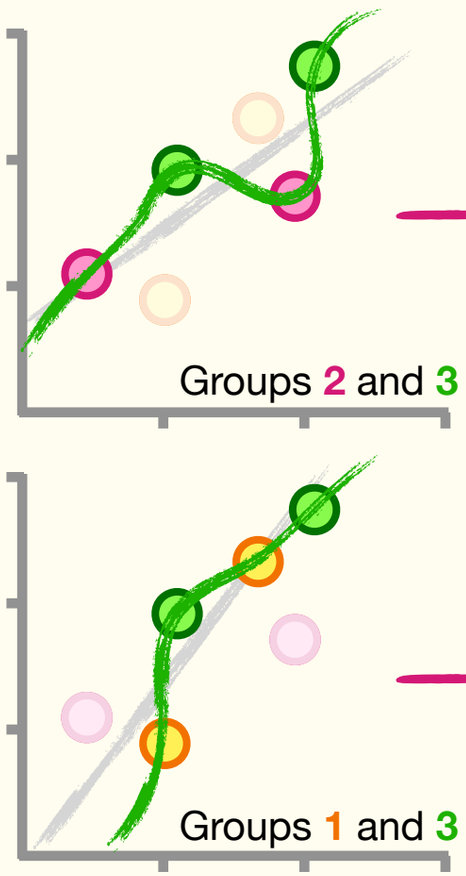
11 For example, we could use **3-Fold Cross Validation** to compare the errors from the **black line** to the errors from the **green squiggle**.

Gentle Reminder: These are the original **3** groups.



12 Training

Iteration #1

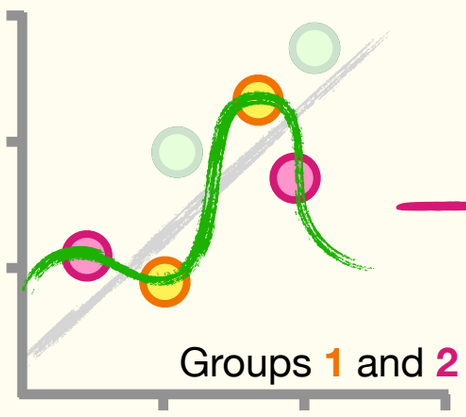


Again, because each iteration uses a different combination of data for **Training**…

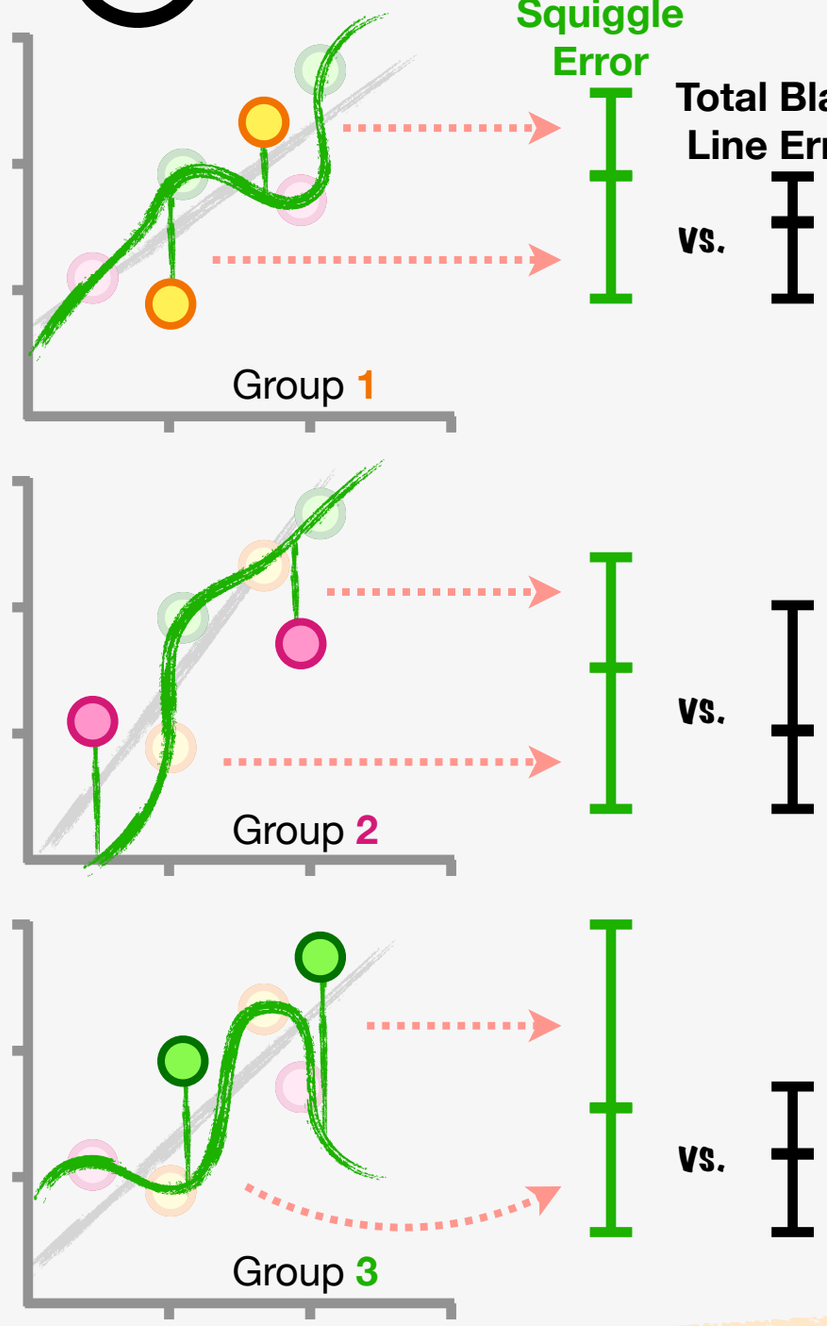
Iteration #2

…each iteration results in a slightly different fitted **line** and fitted **squiggle**.

Iteration #3



13 Testing



In this case, all **3** iterations of the **3-Fold Cross Validation** show that the **black line** does a better job making predictions than the **green squiggle**.

By using **Cross Validation**, we can be more confident that the **black line** will perform better with new data without having to worry about whether or not we selected the best data for **Training** and the best data for **Testing**.

BAM!!!

NOTE: In this example, the **black line** consistently performed better than the **green squiggle**, but this isn't usually the case. We'll talk more about this later.

Cross Validation: Details Part 5

14 When we have a lot of data, **10-Fold Cross Validation** is commonly used.

Imagine that this gray column represents many rows of data.

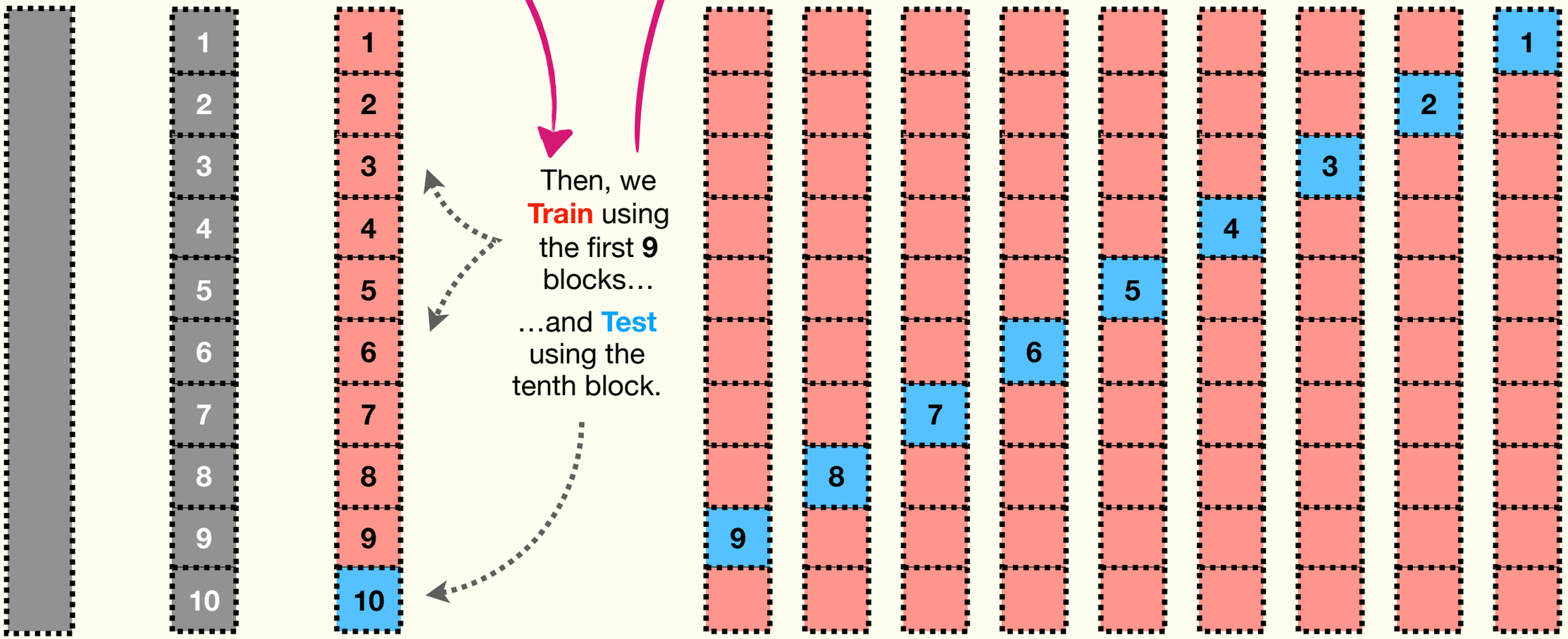
To perform **10-Fold Cross Validation**, we first randomize the order of the data and divide the randomized data into **10** equal-sized blocks.

Then, we **Train** using the first **9** blocks…

…and **Test** using the tenth block.

Then, we iterate so that each block is used for **Testing**.

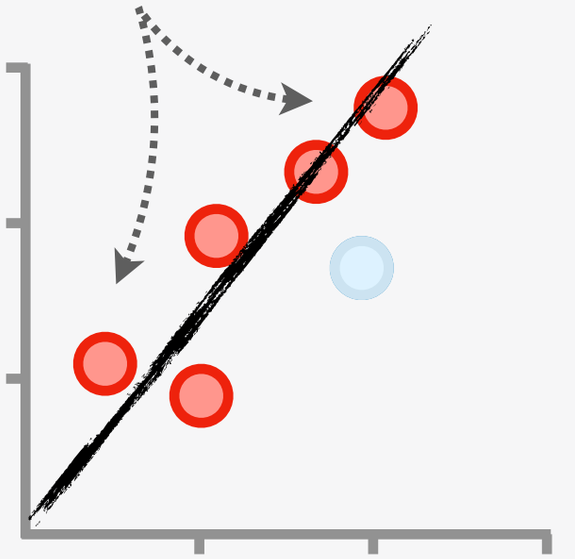
DOUBLE BAM!!!



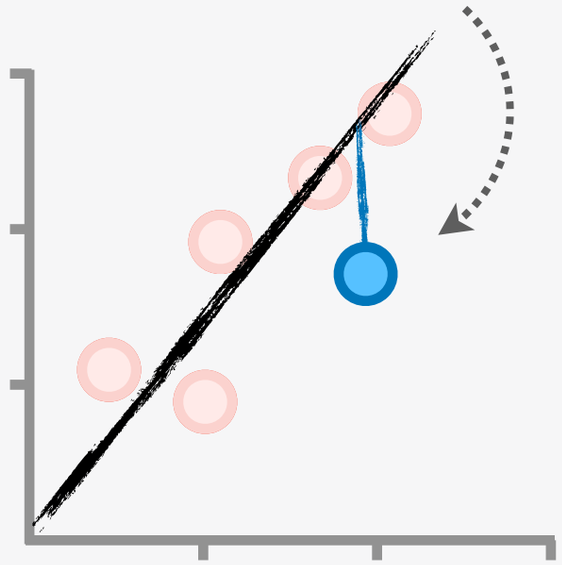
Cross Validation: Details Part 6

15 Another commonly used form of **Cross Validation** is called **Leave-One-Out**.

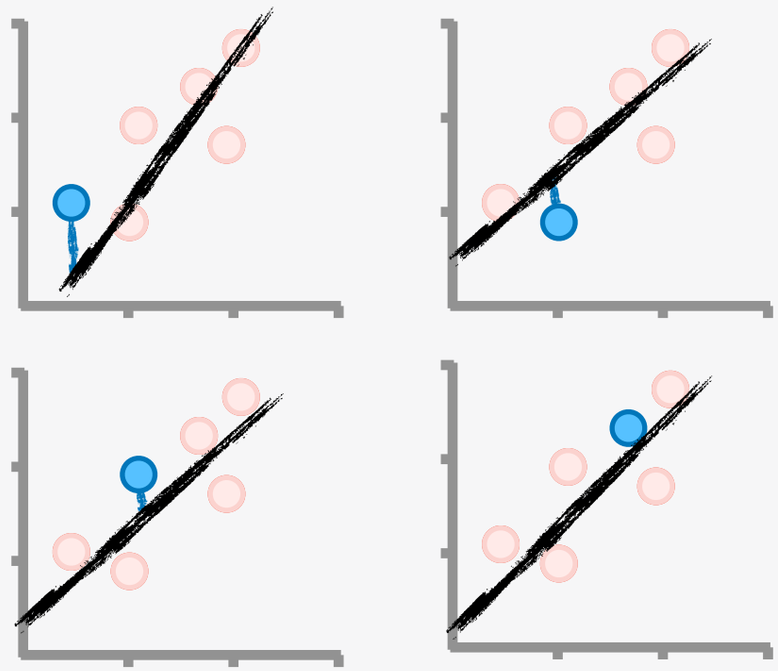
Leave-One-Out Cross Validation uses all but one point for **Training**…

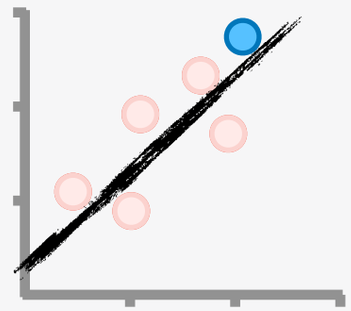


…and uses the one remaining point for **Testing**…



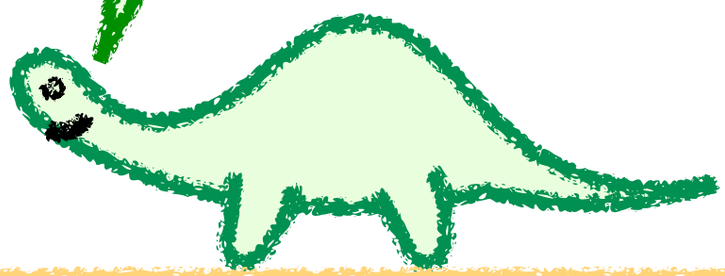
…and then iterates until every single point has been used for **Testing**.





Hey **Norm**, how do I decide if I should use **10-Fold** or **Leave-One-Out Cross Validation**?

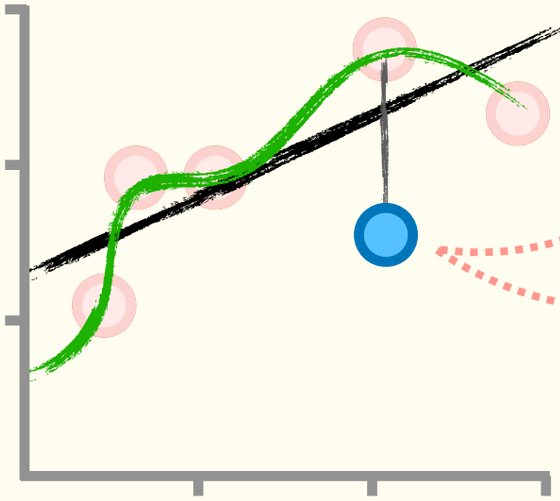
Some experts say that when the dataset is large, use **10-Fold Cross Validation**, and when the dataset is very small, use **Leave-One-Out**.

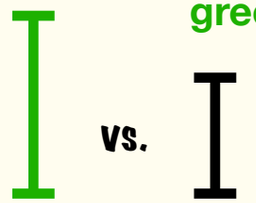


Cross Validation: Details Part 7

16 When we use **Cross Validation** to compare machine learning methods, for example, if we wanted to compare a **black line** to a **green squiggle**…

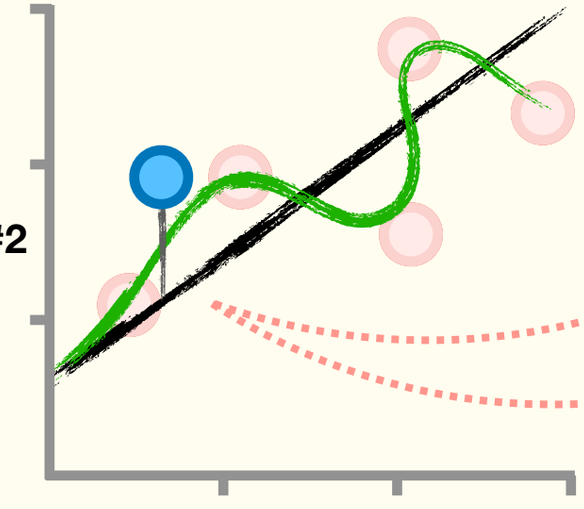
Iteration #1





…sometimes the **black line** will perform *better* than the **green squiggle**…

Iteration #2

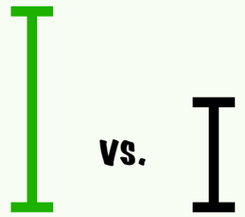


…and sometimes the **black line** will perform *worse* than the **green squiggle**.

And, after doing all of the iterations, we're left with a variety of results, some showing that the **black line** is better…

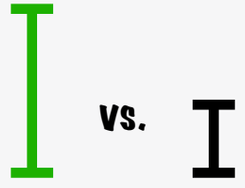
Iteration #3

Iteration #4



Iteration #5

Iteration #6



…and some showing that the **green squiggle** is better.

When the results are mixed, how do we decide which method, if any, is better? Well, one way to answer that question is to use **Statistics**, and that's what we'll talk about in the next chapter.

TRIPLE BAM!!!

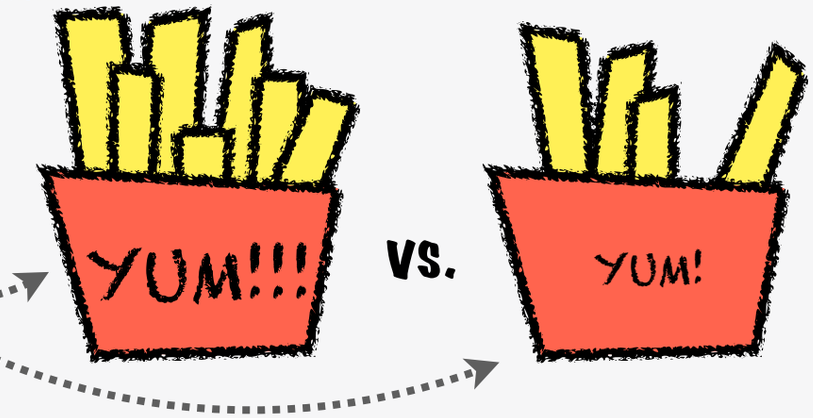
Chapter 03

Fundamental Concepts in Statistics!!!

Statistics: Main Ideas

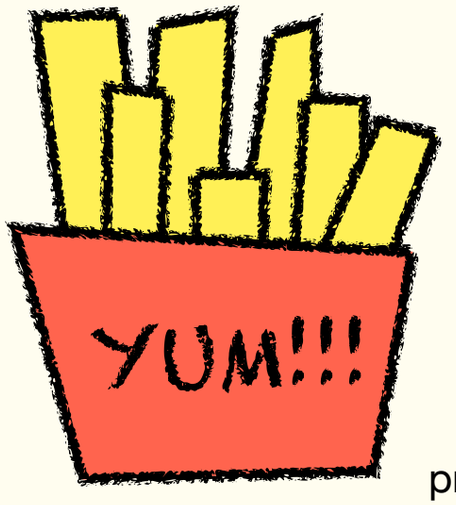
1 **The Problem:** The world is an interesting place, and things are not always the same.

For example, every time we order french fries, we don't always get the exact same number of fries.



2 **A Solution: Statistics** provides us with a set of tools to quantify the variation that we find in everything and, for the purposes of machine learning, helps us make predictions and quantify how confident we should be in those predictions.

For example, once we notice that we don't always get the exact same number of fries, we can keep track of the number of fries we get each day…



Fry Diary

Monday: **21** fries

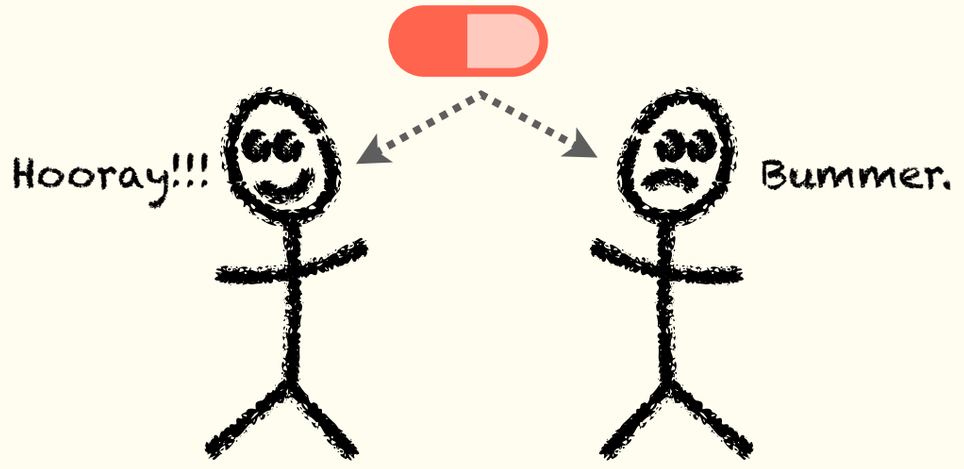
Tuesday: **24** fries

Wednesday: **19** fries

Thursday: ???

…and statistics can help us predict how many fries we'll get the next time we order them, and it tells us how confident we should be in that prediction.

Alternatively, if we have a new medicine that helps some people but hurts others…



…statistics can help us predict who will be helped by the medicine and who will be hurt, and it tells us how confident we should be in that prediction. This information can help us make decisions about how to treat people.

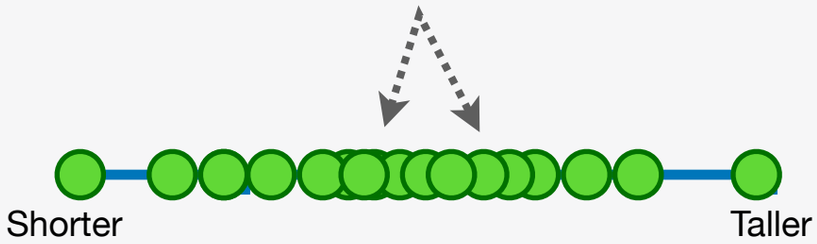
For example, if we predict that the medicine will help, but we're not very confident in that prediction, we might not recommend the medicine and use a different therapy to help the patient.

3 The first step in making predictions is to identify trends in the data that we've collected, so let's talk about how to do that with a **Histogram**.

Histograms: Main Ideas

1 **The Problem:** We have a lot of measurements and want to gain insights into their hidden trends.

For example, imagine we measured the Heights of so many people that the data, represented by **green dots**, overlap, and some **green dots** are completely hidden.



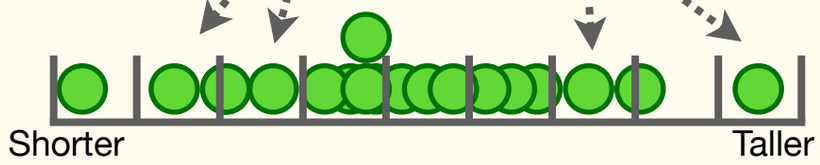
We could try to make it easier to see the hidden measurements by stacking any that are exactly the same…

…but measurements that are *exactly* the same are rare, and a lot of the **green dots** are still hidden.

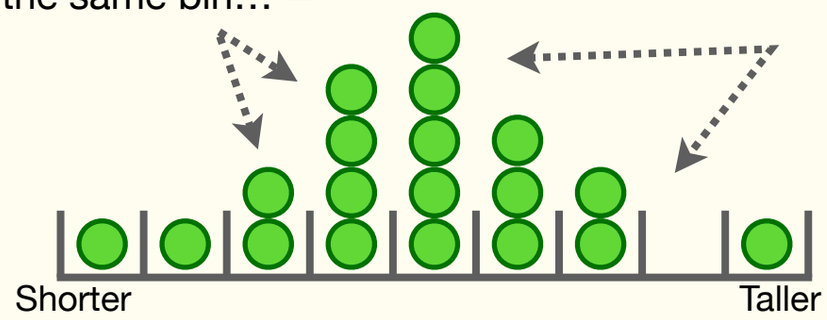
:(

2 **A Solution: Histograms** are one of the most basic, but surprisingly useful, statistical tools that we can use to gain insights into data.

Instead of stacking measurements that are *exactly* the same, we divide the range of values into bins…



…and stack the measurements that fall in the same bin…



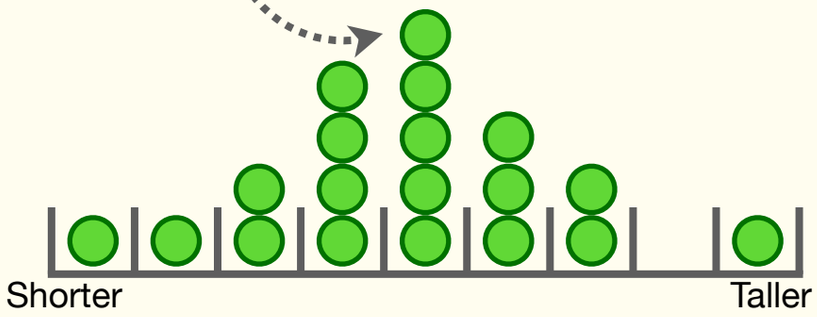
…and we end up with a **histogram!!!**

The **histogram** makes it easy to see trends in the data. In this case, we see that most people had close to average heights.

BAM!!!

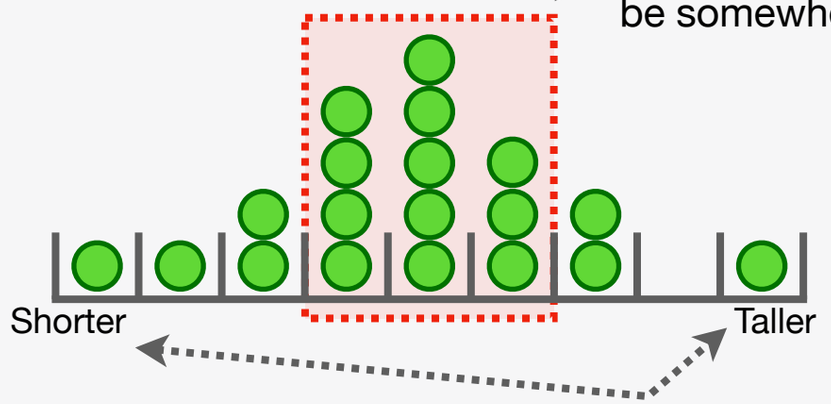
Histograms: Details

1 The taller the stack within a bin, the more measurements we made that fall into that bin.



2 We can use the histogram to estimate the probability of getting future measurements.

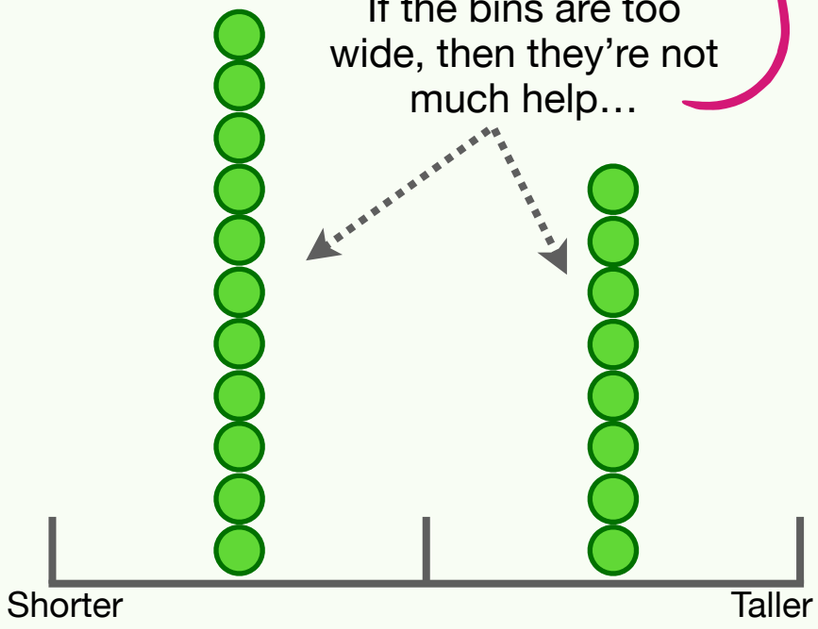
Because most of the measurements are inside this **red box**, we might be willing to bet that the next measurement we make will be somewhere in this range.



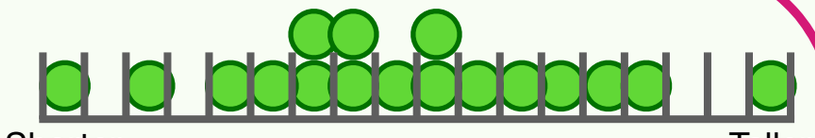
Extremely short or tall measurements are rarer and less likely to happen in the future.

3 **NOTE:** Figuring out how wide to make the bins can be tricky.

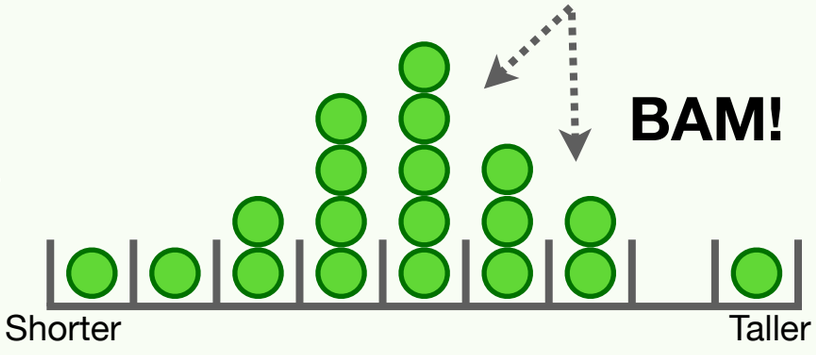
If the bins are too wide, then they're not much help…



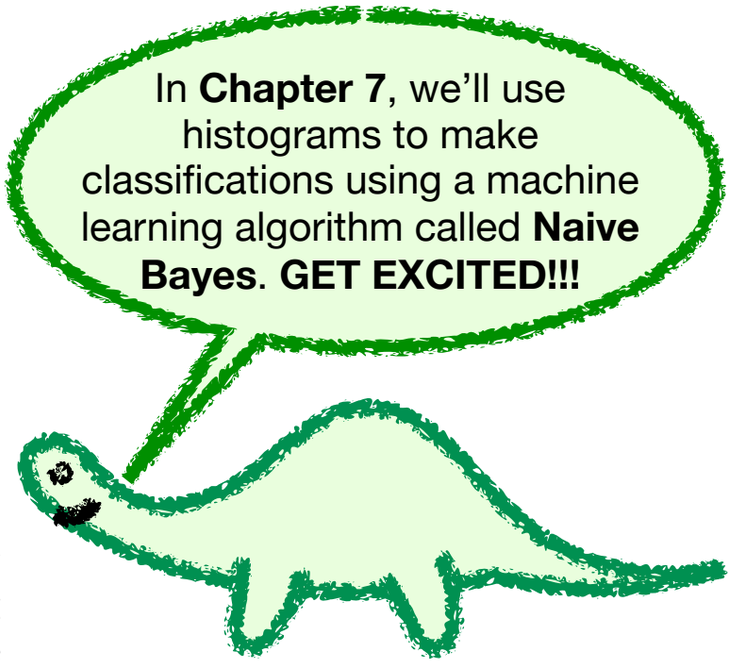
…and if the bins are too narrow, then they're not much help…



…so, sometimes you have to try a bunch of different bin widths to get a clear picture.

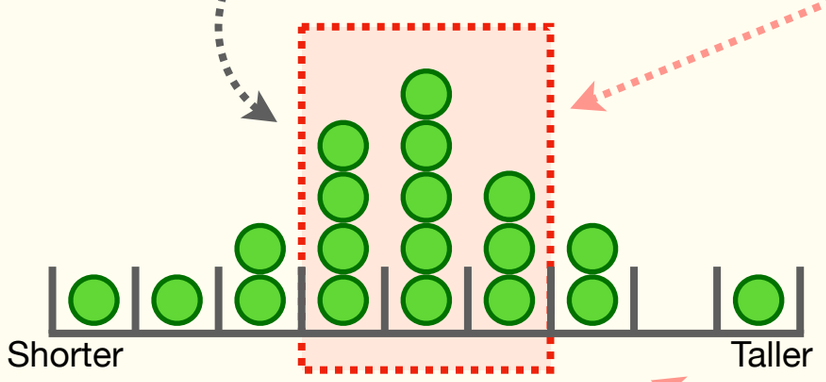


In **Chapter 7**, we'll use histograms to make classifications using a machine learning algorithm called **Naive Bayes**. **GET EXCITED!!!**



Histograms: Calculating Probabilities Step-by-Step

1 If we want to estimate the probability that the next measurement will be in this **red box**…



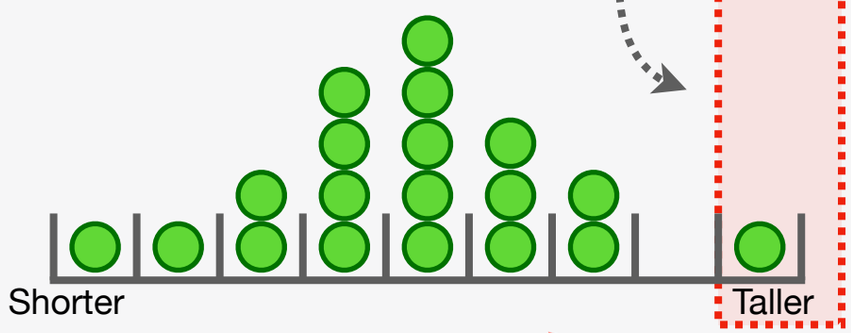
…we count the number of measurements, or observations, *in the box* and get **12**…

…and divide by the *total* number of measurements, **19**…

$$\frac{12}{19} = 0.63$$

…and we get **0.63**. In theory, this means that **63%** of the time we'll get a measurement in the **red box**. However, the confidence we have in this estimate depends on the number of measurements. Generally speaking, the more measurements you have, the more confidence you can have in the estimate.

2 To estimate the probability that the next measurement will be in this **red box**, which only contains the tallest person we measured…



…we count the number of measurements in the box and get **1**…

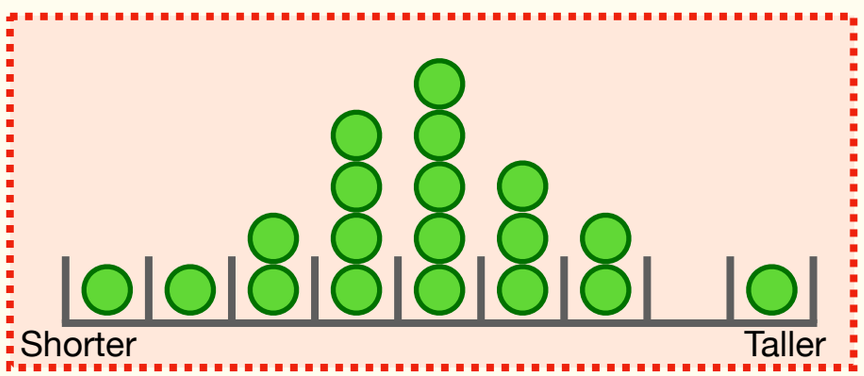
…and divide by the total number of measurements, **19**…

$$\frac{1}{19} = 0.05$$

…and the result, **0.05**, tells us that, in theory, there's a **5%** chance that the next measurement will fall within the box. In other words, it's fairly rare to measure someone who is really tall.

Histograms: Calculating Probabilities Step-by-Step

3 To estimate the probability that the next measurement will be in a **red box** that spans all of the data…



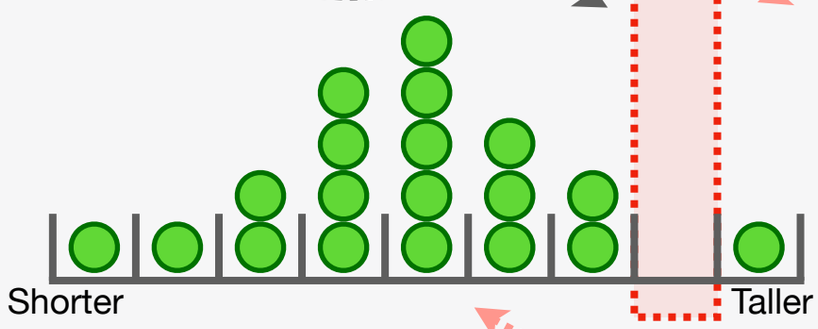
…we count the number of measurements in the box, **19**…

…and divide by the total number of measurements, **19**…

$$\frac{19}{19} = 1$$

…and the result, **1**, tells us that there's a **100%** chance that the next measurement will fall within the box. In other words, the maximum probability is **1**.

4 If we want to estimate the probability that the next measurement will be in this **red box**…



…we count the number of measurements in the box, **0**…

…and divide by the total number of measurements, **19**…

$$\frac{0}{19} = 0$$

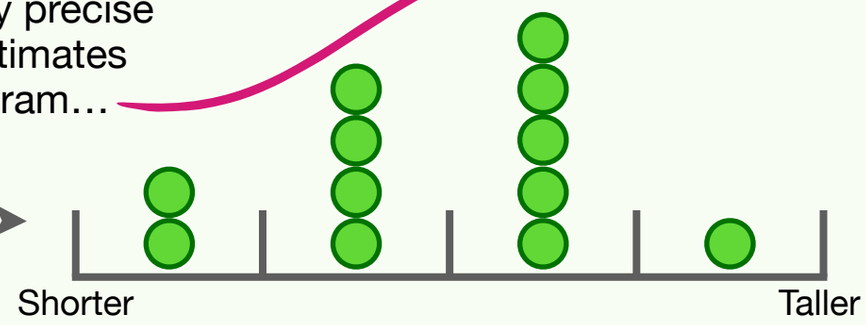
…and we get **0**. This is the minimum probability and, in theory, it means that we'll ***never*** get a measurement in this box. However, it could be that the only reason the box was empty is that we simply did not measure enough people.

If we measure more people, we may either find someone who fits in this bin or become more confident that it should be empty. However, sometimes getting more measurements can be expensive, or take a lot of time, or both. This is a problem!!!

The good news is that we can solve this problem with a **Probability Distribution**. **Bam!**

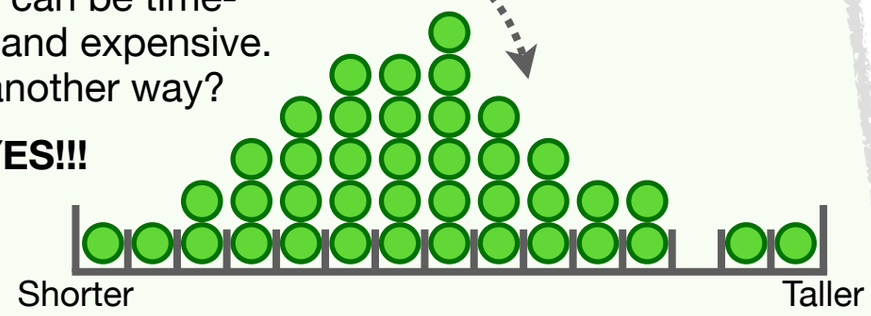
Probability Distributions: Main Ideas

1 **The Problem:** If we don't have much data, then we can't make very precise probability estimates with a histogram…

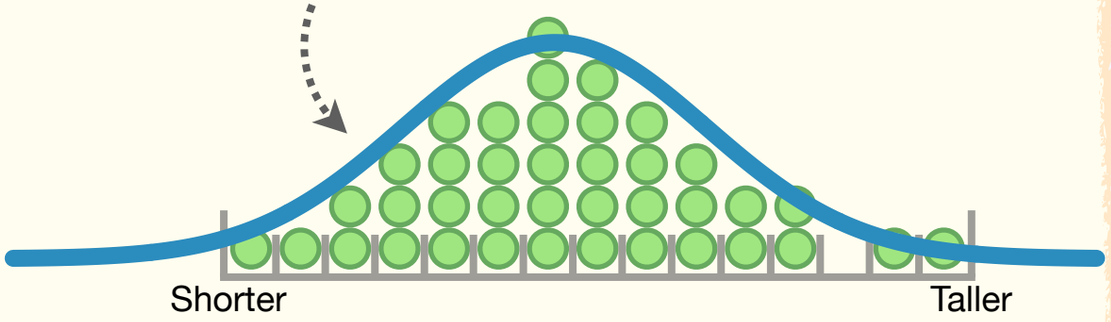


…however, collecting tons of data to make precise estimates can be time-consuming and expensive. Is there another way?

YES!!!

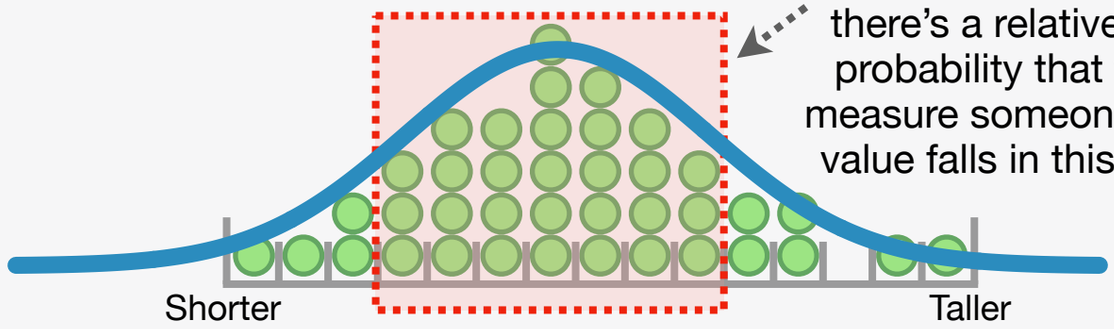


2 **A Solution:** We can use a **Probability Distribution**, which, in this example, is represented by a **blue, bell-shaped curve**, to approximate a histogram.



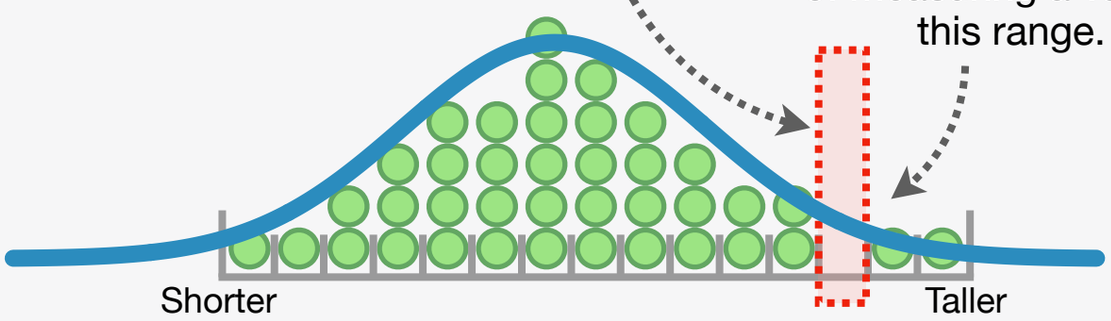
3 This **blue, bell-shaped curve** tells us the same types of things that the histogram tells us.

For example, the relatively large amount of area under the curve in this **red box** tells us that there's a relatively high probability that we will measure someone whose value falls in this region.

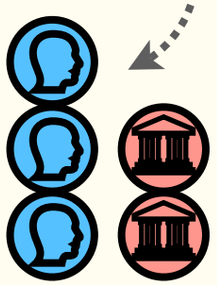


4 Now, even though we never measured someone who's value fell in this range…

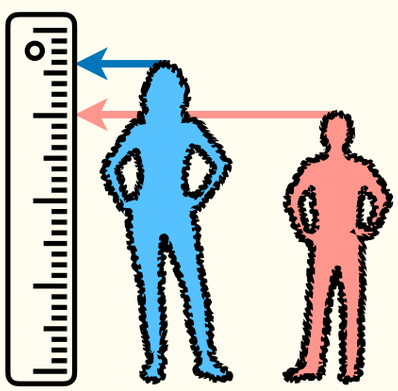
…we can use the area under the curve to estimate the probability of measuring a value in this range.



5 **NOTE:** Because we have **Discrete** and **Continuous** data…



…there are **Discrete** and **Continuous Probability Distributions**.



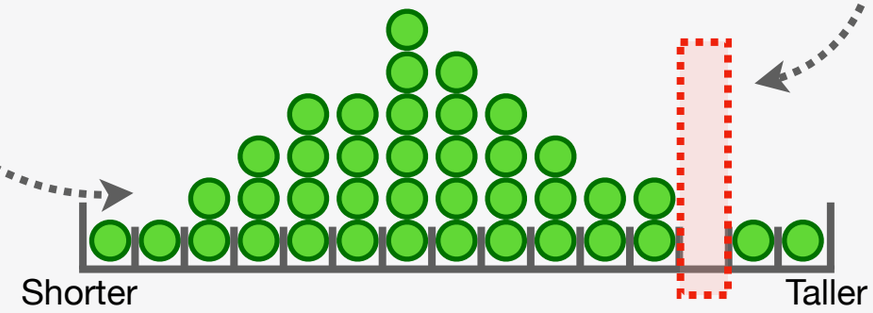
So let's start by learning about **Discrete Probability Distributions**.

Discrete Probability Distributions: Main Ideas

1

The Problem: Although, technically speaking, histograms are **Discrete Distributions**, meaning data can be put into discrete bins and we can use those to estimate probabilities…

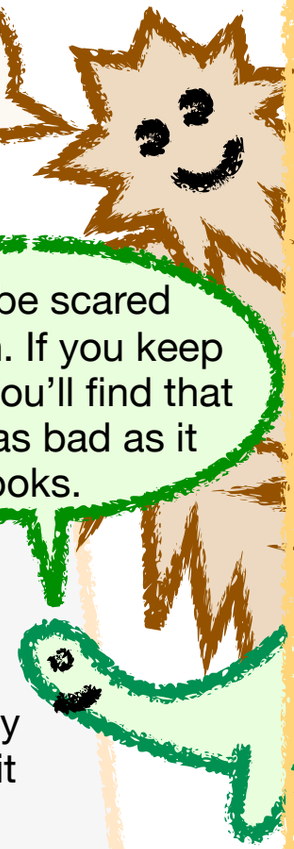
…they require that we collect a lot of data, and it's not always clear what we should do with blank spaces in the histograms.



2

A Solution: When we have discrete data, instead of collecting a ton of data to make a histogram and then worrying about blank spaces when calculating probabilities, we can let ***mathematical equations*** do all of the hard work for us.

The **Binomial Distribution** makes me want to run away and hide.



Don't be scared **'Squatch**. If you keep reading, you'll find that it's not as bad as it looks.

3

One of the most commonly used **Discrete Probability Distributions** is the **Binomial Distribution**.

As you can see, it's a mathematical equation, so it doesn't depend on collecting tons of data, but, at least to **StatSquatch**, it looks *super scary!!!*

$$p(x \mid n, p) = \left(\frac{n!}{x!(n-x)!} \right) p^x (1-p)^{n-x}$$

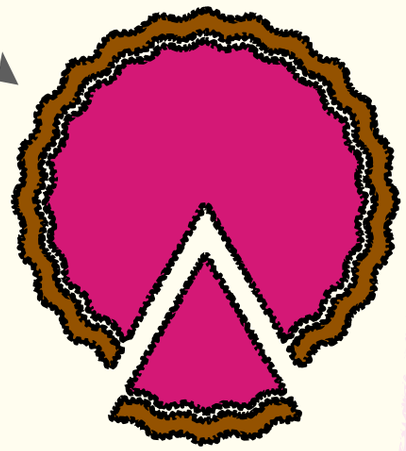
The good news is that, deep down inside, the **Binomial Distribution** is really simple. However, before we go through it one step a time, let's try to understand the main ideas of what makes the equation so useful.

The Binomial Distribution: Main Ideas Part 1

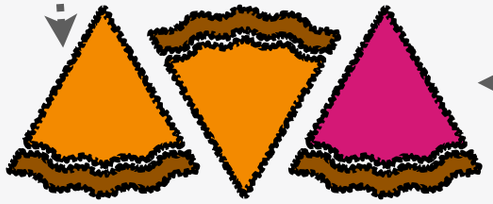
1 First, let's imagine we're walking down the street in **StatLand** and we ask the first **3** people we meet if they prefer pumpkin pie or blueberry pie…

Pumpkin Pie

Blueberry Pie



2 …and the first **2** people say they prefer pumpkin pie…

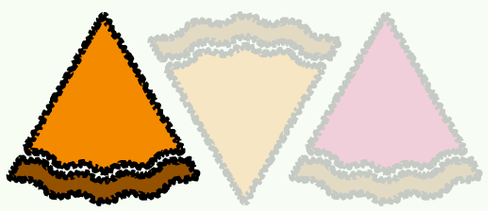


…and the last person says they prefer blueberry pie.

Based on our extensive experience judging pie contests in **StatLand**, we know that **70%** of people prefer pumpkin pie, while **30%** prefer blueberry pie. So now let's calculate the probability of observing that the first two people prefer pumpkin pie and the third person prefers blueberry.

3 The probability that the first person will prefer pumpkin pie is **0.7**…

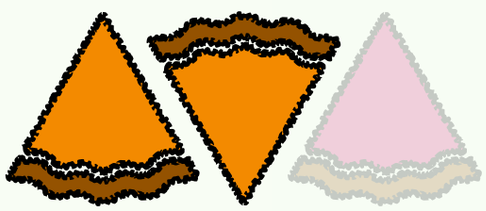
0.7



…and the probability that the first two people will prefer pumpkin pie is **0.49**…

(Psst! If this math is blowing your mind, check out **Appendix A**.)

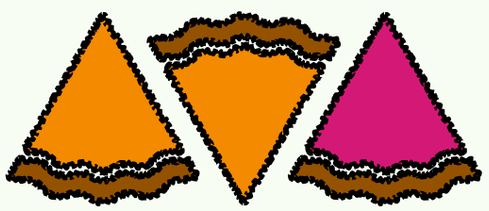
0.7 x **0.7** = **0.49**



…and the probability that the first two people will prefer pumpkin pie and the third person prefers blueberry is **0.147**.

(Again, if this math is blowing your mind, check out **Appendix A**.)

0.7 x **0.7** x **0.3** = **0.147**



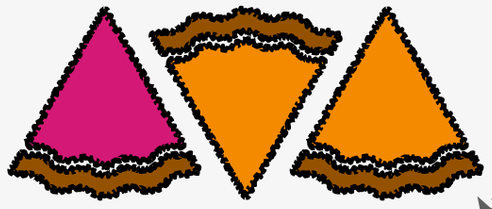
NOTE: 0.147 is the probability of observing that the first two people prefer pumpkin pie and the third person prefers blueberry…

…it is *not* the probability that **2** out of **3** people prefer pumpkin pie.

Let's find out why on the next page!!!

The Binomial Distribution: Main Ideas Part 2

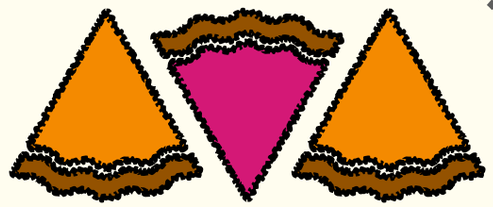
4 It could have just as easily been the case that the first person said they prefer blueberry and the last two said they prefer pumpkin.



In this case, we would multiply the numbers together in a different order, but the probability would still be **0.147** (see **Appendix A** for details).

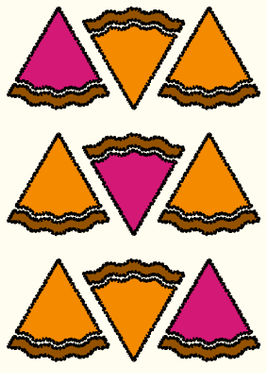
0.3 x **0.7** x **0.7** = **0.147**

5 Likewise, if only the second person said they prefer blueberry, we would multiply the numbers together in a different order and still get **0.147**.



0.7 x **0.3** x **0.7** = **0.147**

6 So, we see that all three combinations are equally probable…

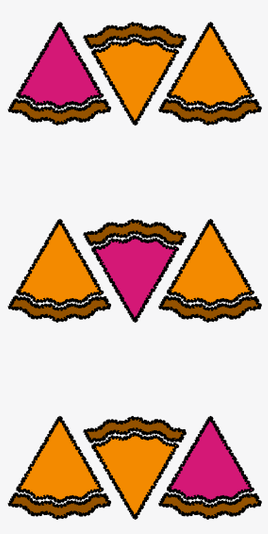


0.3 x **0.7** x **0.7** = **0.147**

0.7 x **0.3** x **0.7** = **0.147**

0.7 x **0.7** x **0.5** = **0.147**

7 …and that means that the probability of observing that **2** out of **3** people prefer pumpkin pie is the ***sum*** of the **3** possible arrangements of people's pie preferences, **0.441**.



0.3 x **0.7** x **0.7** = **0.147**

+

0.7 x **0.3** x **0.7** = **0.147**

+

0.7 x **0.7** x **0.5** = **0.147**

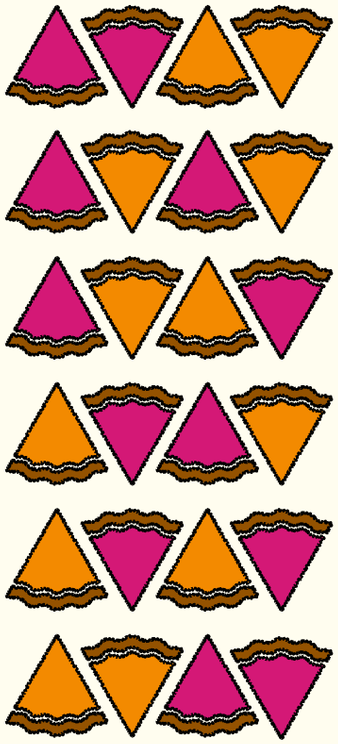
= 0.441

NOTE: Calculating by hand the probability of observing that **2** out of **3** people prefer pumpkin pie was not that bad. All we did was draw the **3** different ways **2** out of **3** people might prefer pumpkin pie, calculate the probability of each way, and add up the probabilities.

Bam.

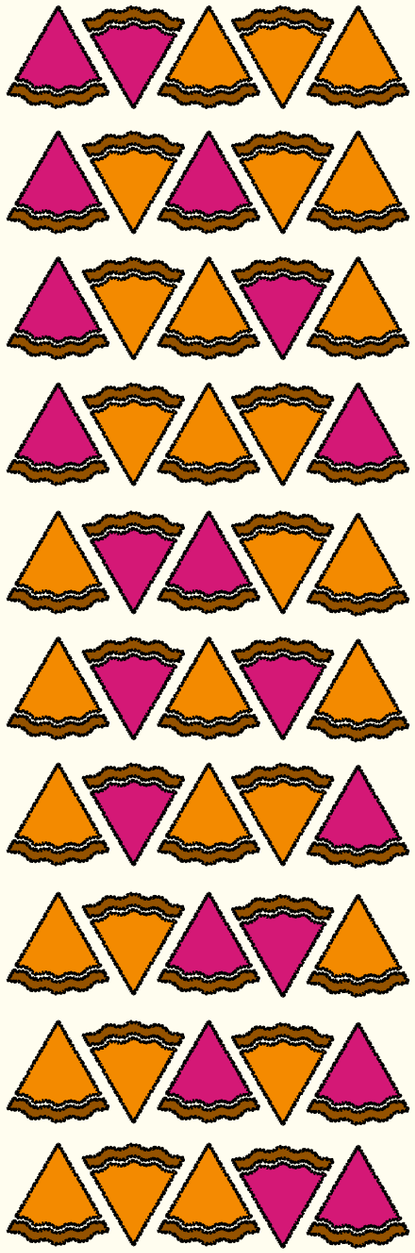
The Binomial Distribution: Main Ideas Part 3

8 However, things quickly get tedious when we start asking more people which pie they prefer.



For example, if we wanted to calculate the probability of observing that **2** out of **4** people prefer pumpkin pie, we have to calculate and sum the individual probabilities from **6** different arrangements…

…and there are **10** ways to arrange **3** out of **5** people who prefer pumpkin pie.



UGH!!! Drawing all of these slices of delicious pie is super tedious.

:(

9 So, instead of drawing out different arrangements of pie slices, we can use the equation for the **Binomial Distribution** to calculate the probabilities directly.

$$p(x \mid n, p) = \left(\frac{n!}{x!(n-x)!} \right) p^x (1-p)^{n-x}$$

BAM!!!

In the next pages, we'll use the **Binomial Distribution** to calculate the probabilities of pie preference among **3** people, but it works in any situation that has binary outcomes, like wins and losses, yeses and noes, or successes and failures.

Now that we understand why the equation for the **Binomial Distribution** is so useful, let's walk through, one step at a time, how the equation calculates the probability of observing **2** out of **3** people who prefer pumpkin pie.

0.3 x **0.7** x **0.7** = **0.147**

+

0.7 x **0.3** x **0.7** = **0.147**

+

0.7 x **0.7** x **0.3** = **0.147**

= 0.441