Andrew N Carr

Everyday Data Science

Optimize Your Life

ANDREW N CARR

EVERYDAY DATA SCIENCE

Dedicated to AnnaLisa, For putting up with my clicky keyboard

Contents

1 Everyday Data Science

Life is full of decisions. We, as people, have the remarkable ability to make decisions in the face of uncertainty. We have recently developed the ability to use computers to process vast amount of data to improve our decision making. This has led to the development of the field of Data Science.

This book is written to give tools and inspiration to aspiring decision makers. You make decisions daily and the methodology of data science can help.

1.1 What is Data Science?

This question is challenging, and might be better answered by asking "what isn't Data Science?". It's not magic. Nor is it a cure for all of societies ills[1].

[1] Although, there are some people who will try and tell you otherwise

It is simply a set of tools that are useful when you are trying to make decisions using data. In general, however, data science is the study of decision making informed by data. It requires questions, a quality source of data, analysis, and communication of results. There are many different kinds of data. You can have a table with rows, like a spreadsheet, an picture with friends, a tree of relationships, or written text.

I've collected a number of fun case studies where data is used in *everyday* situations. I've also included some interesting math and tricks you can use in your day-to-day life.

1.2 How is Data Science?

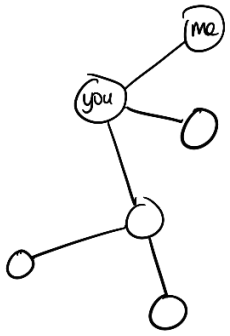
Or rather, "How is Data Science done?". Data science happens when someone uses methods from programming, stats, and mathematics to analyze some data. The analysis usually takes the form of graphs, charts, and numbers. The numbers can summarize important data[2] and the graphs can show important trends.

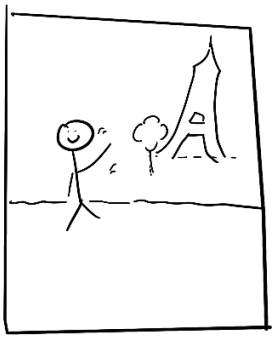
[2] For example, how many steps you've taken in the day

To show how this works, let's look at the question "Why hasn't there been a new Pirates of the Caribbean movie recently?". I remember the series fondly and was wondering about this very question myself.

DATA

Name	Age	Friends
me	~	[you]
you	~	[me]
them	~	~





"I have lots of friends and we like travel"

So I started the Data Science process by first framing the question *Why hasn't there been a sequel to Pirates of the Caribbean recently?*.

The process of making decisions, especially with data, must begin with a proper hypothesis, a theory about why something happens that you can test. There are many who just "see what they can find" in the data, but you need to know what you're looking for.

In this case, I had a theory I wanted to test. You only make a sequel if the most recent movie was successful and well received. A proxy of success can be audience score from a movie reviewing site like Rotten Tomatoes.

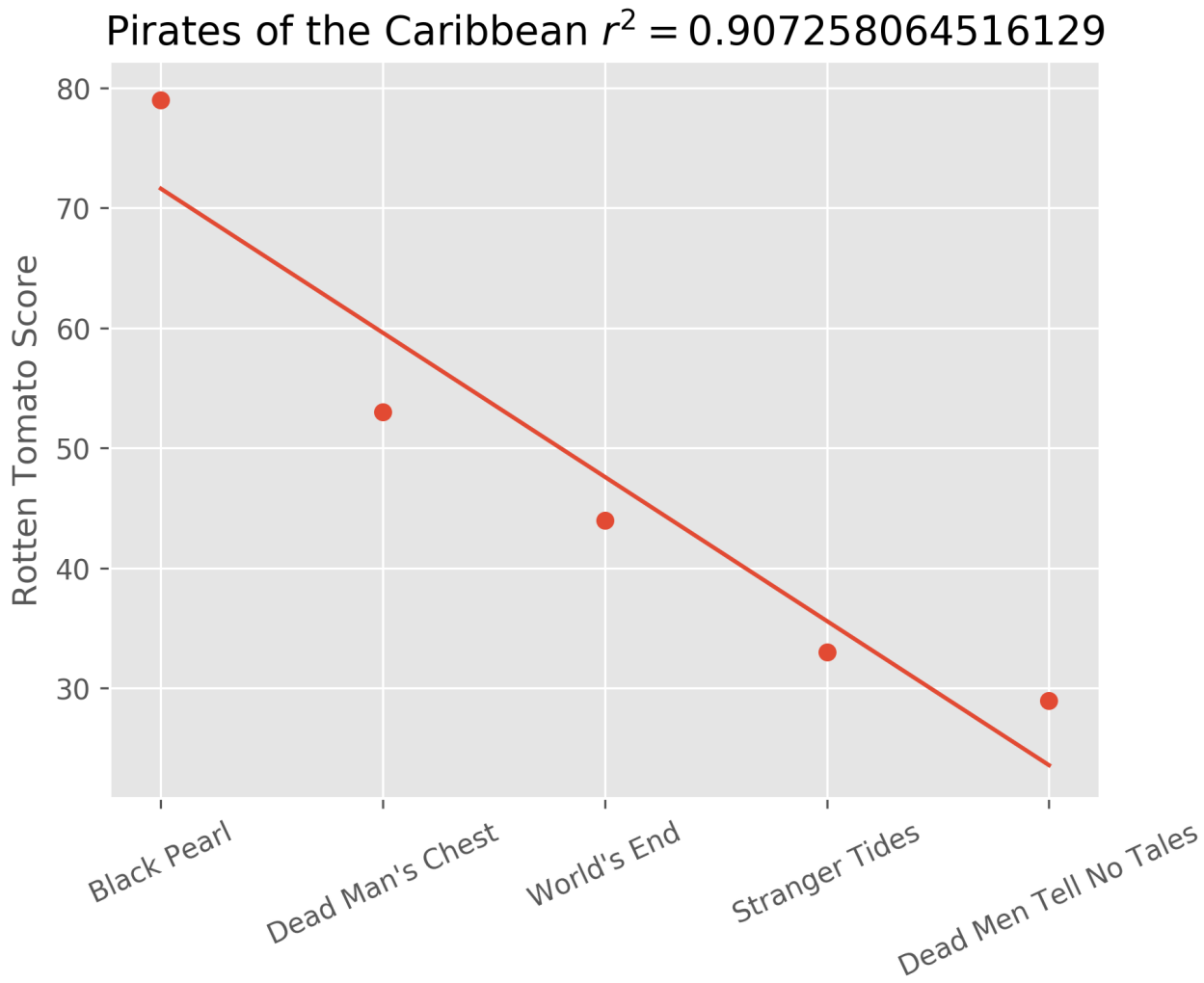
I collected the data by simply going to the website for each movie and writing down the score. I then plotted those scores to see if there was an interesting trend. As you can see in the graph on the next page, the Rotten Tomatoes score dropped rapidly with each subsequent movie.

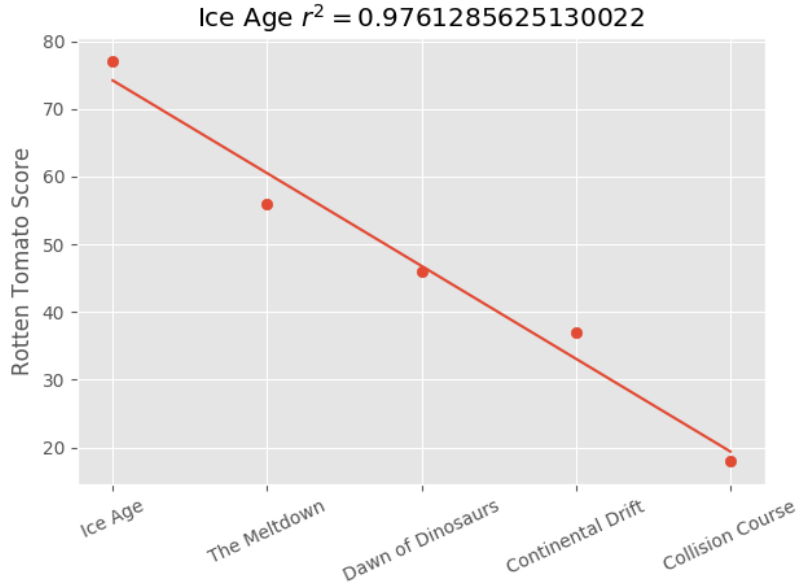
The number at the top is the r^2 score, or goodness rating, of the line on the graph. It measures how much of the audience score can be explained by a *negative linear relationship*. A negative linear relationship means that we think the score will drop by the same amount between each movie. The r^2 score tells us that this relationship can explain 90% of the change in scores. An even better model would curve like the dots do, perhaps what is called a *decreasing exponential model*, which would also keep the scores from going negative if they made more movies[3].

[3] Although, maybe people would have truly tried to give it a negative score if they made any more movies

This trend seems to be pretty consistent across movies with a large number of sequels.

Although, if you look at the profit of the two movies, we see that


Pirates of the Caribbean $r^2 = 0.907258064516129$
Rotten Tomato Score
80
70
60
50
40
30
Black Pearl
Dead Man's Chest
World's End
Stranger Tides
Dead Men Tell No Tales



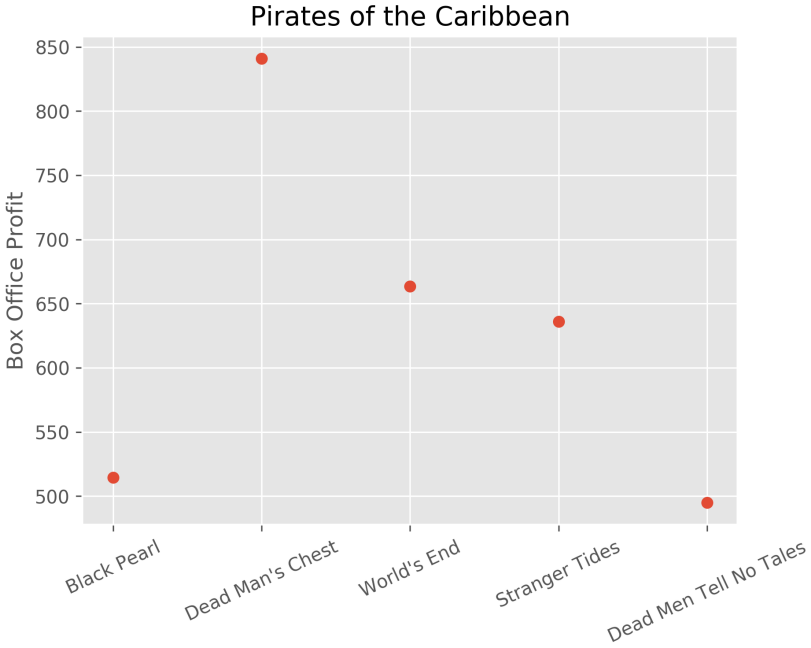
the points aren't in a straight line, so a linear model wouldn't work as well. But we do observe that profit rises and then falls as more movies get made.

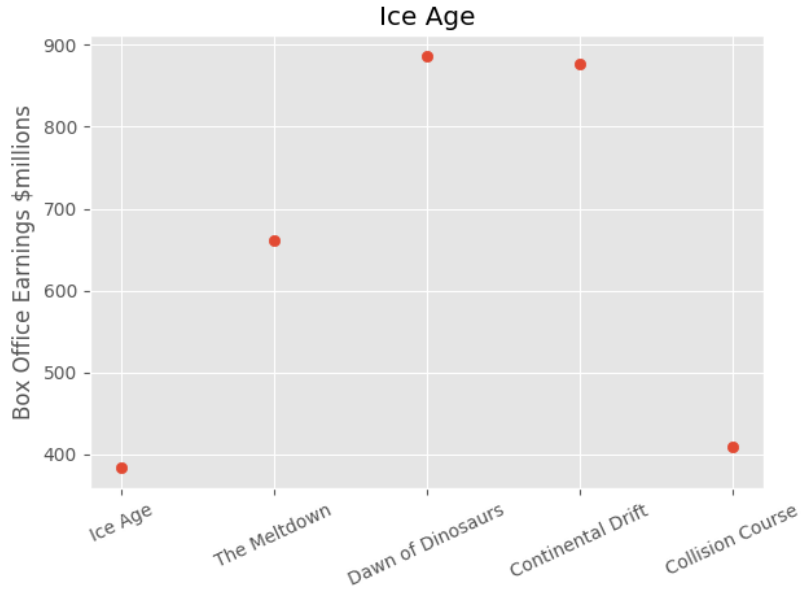
Interestingly, the fall starts to happen fairly quickly for Pirates of the Caribbean while it takes longer to start falling for Ice Age. The last movie for each came out around the same time.

In the end, we see here a really basic example of data science. If we were in charge of giving the green light for another movie in either of these franchises we would be better able to see the trend in audience sentiment and earnings from previous iterations.

1.3 *Why is Data Science?*

Anyone who needs to make regular decisions should be familiar with methods in data science. It is estimated that more than half of all


Pirates of the Caribbean
Box Office Profit
850
800
750
700
650
600
550
500
Black Pearl
Dead Man's Chest
World's End
Stranger Tides
Dead Men Tell No Tales


Ice Age
Box Office Earnings $millions
900
800
700
600
500
400
Ice Age
The Meltdown
Dawn of Dinosaurs
Continental Drift
Collision Course

organizations will suffer from data illiteracy, meaning that those in the organization won't understand what the data means, and how to use it to make decisions. This number grows in many decision making positions such as managers and company leaders.

Data is everywhere and contains interesting insights that can help us make better decisions as a society. However, in this book, we'll mostly focus on how data can improve our daily lives and where we can apply these methods.

1.4 *Who is Data Science?*

You

1.5 *When is Data Science?*

Everyday

1.6 *Terms to know*

Like many in technical fields, data scientists enjoy jargon. While I tried to keep it to a minimum, I still do use jargon several times in this book.

So, I'll put a few terms here that get used commonly in data science.

- Model: A piece of math or stats that is used to explain or predict something.

- Fit: The process of showing data to a model to the model get better at explaining or predicting something.
- Graph: A visual representation of data
- Function: A relationship between a group of two or more things[4].
- Object: Catch-all term for anything that follows a predetermined list of rules.

[4] There is a function between a movie and its Rotten Tomato score

1.7 *Reading this book*

Each chapter is a self contained case study designed to explain some overarching principle. There is plenty of mathematics in this book, but also plenty of images, so you can understand the ideas without diving deep into the equations[5].

[5] Although, there are lots of interesting ideas in the mathematics

You can read the chapters in whichever order you choose and skip around as you see fit, as there is no overarching story to follow between chapters.

This is not meant to be a textbook or reference for data science techniques. Neither is it meant to be an exhaustive example of ways you can use data in your daily life. It is simply a collection of interesting use cases that is designed to serve as an inspiration to you. If you find yourself interested in the ideas presented, feel free to reach out[6] to me or explore other books more deeply to gain a deeper understanding of the principles of Data Science.

[6] My contact information is at the end of the book.

2 *When Life Gives You Lemons*

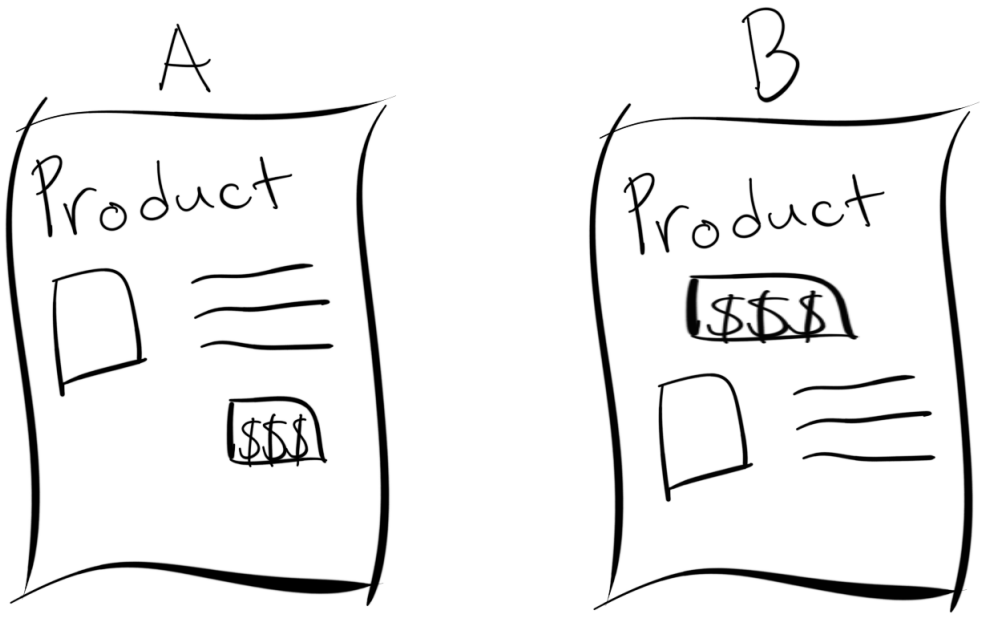
An Everyday Look At A/B Testing

One of the greatest pleasures in life is a cool glass of lemonade after a hot day laboring over your GPU. You've spent the afternoon tuning hyperparameters using the age-old method of "guess and check", and now you deserve a break. You meander over to the fridge, chuckling to yourself as you remember a colleague who used logistic regression instead of a resnet-51. To your utmost horror, you find that you drank the last of the lemonade yesterday, now all you have in the fridge is water, sugar, and whole lemons.

"What now?" you ask yourself. That batch of lemonade had the perfect ratio of ingredients, but you've forgotten the numbers. As an intrepid data scientist, you jump into action to design an A/B test which will determine the perfect ratio of water, sugar, and lemons.

2.1 *What is A/B testing?*

Imagine you are a data scientist for a large shopping website named after a massive rainforest. Congo. Your website sells tons of goods everyday. However, you believe changes to the product page can increase sales, and profits, even further.



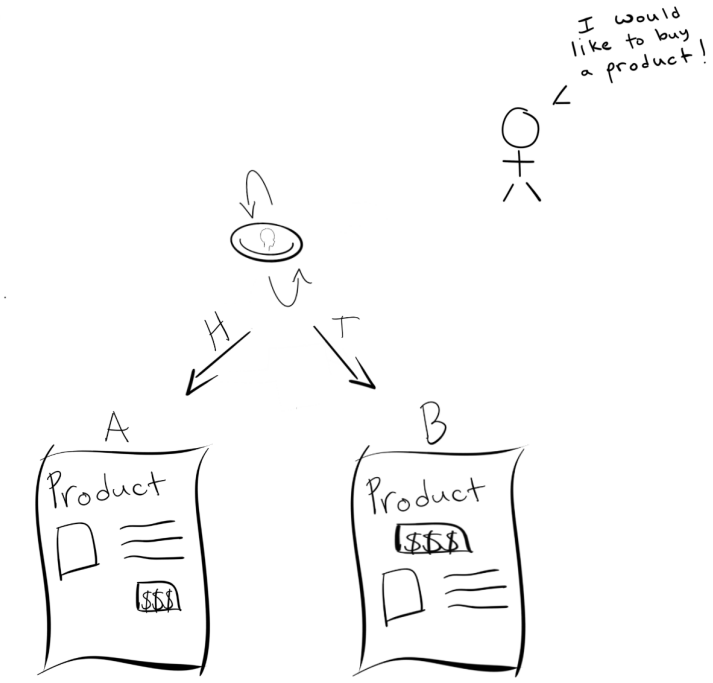
Our current plan, shown here as Plan B, has the *Buy Button* (shown here with the [$$$] box) before the information about what we are selling. We think that customers would be better informed, and would then end up buying our product, if the button came after the product's information. A/B testing is a way to test that hypothesis and determine for sure which product page is best for the customer and business.

There are many ways to run a good A/B test. Many of them can

get very complicated as you are working with more and more options. However, as an everyday data scientist, your toolkit looks different. So, let's explore the simplest type of A/B test, before taking a peek at more complicated versions.

2.1.1 *The simple way*

The simple way may be the easiest way initially. Continuing with the product page example, you get the go ahead from management to run your test over the next two weeks. You work with engineering to build a system that routes people to one of the two pages, randomly, based on a coin flip.



After the 2 weeks are up, you start combing through the data to see what you can find.

Page A	0	1	1	1	1	1	1	1	0	.....
Page B	0	1	1	0	1	1	1	0	1	.....

	0	1
Counts A	1725	3275
Counts B	2198	2802

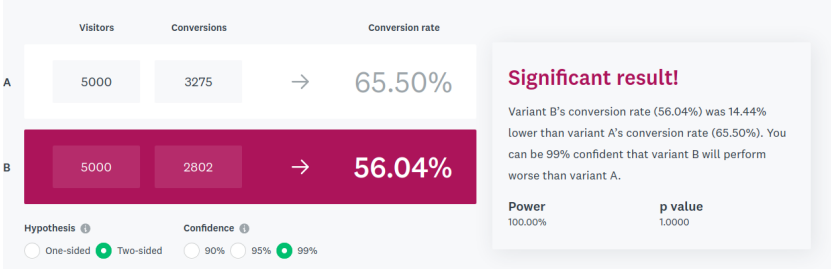
In the lines above, a **1** means that a customer was directed to the page, and then bought the item. A **0** means they left without buying anything. We're making some assumptions about the customer behavior here that are important. For example, we ignore the case where a customer comes back later and buys the item. This is similar to the real world lemonade example we'll work through later.

Using the simple method[1], we take the data and count the number of 0's and the number of 1's. If we look at these sorted counts of our method, we find that page A has more 1 values than page B while they both have 5000 recorded data points.

[1] 0 means a customer saw the page, and chose not to buy. While 1 means the customer bought the product.

Page A has a 66% conversion rate, while page B only has a 56% conversion rate. This seems to be a meaningful difference. But to be sure, we run a quick p-test[2]. Sure enough, we find our results to be statistically significant.

[2] A statistics test that tells us whether the results are just random chance, or if they actually mean something



survey monkey A/B significance testing

At the end of the day this works fine, however there are some potential problems. The first of which is that we needed 10,000 visitors and 2 weeks to gather these results. If we want to test another change to the page, we have to spend a bunch more time on another round of A/B testing. Additionally, that 14% difference is

actual lost revenue for the company. Most importantly, if we want to run an A/B test to determine the best lemonade combination, we likely don't have 10,000 friends willing to try our concoctions.

2.1.2 *A better way*

It turns out that A/B testing can be modeled as a multi-armed bandit problem.

The name "one-armed bandit" is often used to describe slot machines. These machines have a single lever which you pull that produces a random pay out, and you get whatever you get. In the multi-armed version, each arm has a random payout, but some have a higher average payout than others, so we want to pull those levers once we find them. In the case of our lemonade dilemma, we are trying to figure out which arm (combination of ingredients) results in the highest score when pulled (tastes the best).

Picture a slot machine with three levers. The first gives 10 when pulled, the second 1, and the third 25. Obviously, once we figure this out, we'd want to pull Lever Three all the time (pick number 3 m'lord!), and since the numbers here are fixed, it is easy to pull the levers and figure out which is best.

However, in our everyday life, we need to test our lemonade on multiple people who all have slightly different preferences. Since the types of lemonade illicit different reactions from different people, the pulling of levers is now probabilistic[3]. We would need to give a lot of people each combination to figure out which lemonade is scoring best, and so we need to pull the levers in a clever way to figure out which lever is best. Or, in other words, this means we need to present our friends with cups of lemonade in a clever way to figure out which recipe is best.

[3] Someone might rate the worst lemonade as the best, and another might rate the best one as the worst, but when you take all of the scores together, the best lemonade would have a higher average score than the rest

In general, you want to take as few tries to find the best option as possible, as then you can focus on pulling the right lever as often as you can. Since you only have a limited number of friends to try your lemonade, you will need to use the information you gain during the test to make it more efficient while you are testing.

Ok, how do we do that? The first idea might be a greedy™ approach. Once you try all the levers, you always pull the lever that gives maximum reward. This is fine in the case of a few options (2 or 3) but when the number of levers / combinations (also known as the state space) is huge then it quickly becomes impossible to try them all. Also, even with a few arms, a single pull might give you misleading results. If one person rates a bad lemonade recipe as great during the initial round of tests, you could end up stuck with that bad combination when more pulls would have revealed a better

recipe for your lemonade.

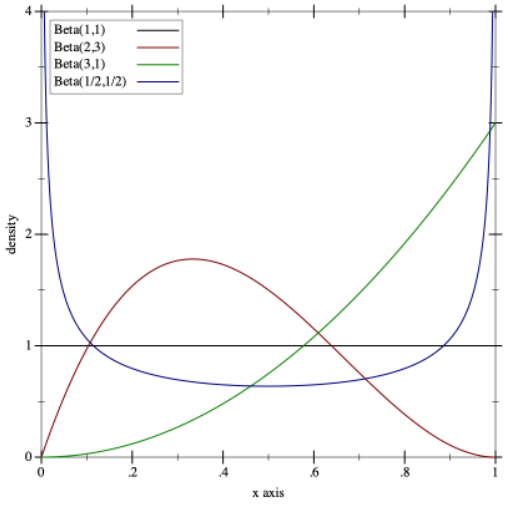
It turns out there is a simple solution called Thompson sampling. Thompson sampling is an algorithm for online decision making where after you give one person a random sample to try, you use their response to change how 'random' the next sample will be. So, if you get several people that like Option A, then it becomes more likely that you'll give future testers Option A. It was first introduced in 1933, but was pretty much ignored by the academic community until decades later. Then at the beginning of the 2010's, it was shown to have very strong practical applications which has led to a widespread adoption of the method.

The idea can be complicated to understand but simple to implement and incredibly powerful. In no time, you can impress your boss with the perfect glass of lemonade. To set up the problem to solve with Thompson sampling for lemonade we're going to simplify slightly and only consider two potential recipes. We can treat each recipe (parts sugar, water, and lemon) as an arm on the bandit.

The reason Thompson sampling works is because people's preferences have a certain mathematical *shape*, called a **distribution**. In our earlier testing, we were using a coin-flip to determine which recipe to let people try, which is modeled using what is called a *Bernoulli distribution*, which simply randomly flips between two options. For Thompson sampling, however, we will use what is called a *Beta distribution* which has a pair of variables alpha and beta that represent prior successes and failures. The next paragraph gives a deeper dive, but you can move on if you're not interested in the mathematics.

There is a certain underlying probability about people's preference for each lemonade recipe. We cleverly choose a cup to present to our friend and they can choose to drink it all or leave it after a sip. Our friend's preference can be modeled with a probability similar to that of a coin flip. The coin flip distribution is called the Bernoulli distribution $p(1-p)$ if $k = 1$. Thompson sampling will try to figure out the probabilities of this distribution. The key piece of genius is which distribution to use as a prior which will be filled in as we guess. Thompson sampling uses a Beta distribution[4] which has two parameters which represent success and failure α, β.

[4] $\frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$, with $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha\beta)}$



Because Thompson sampling updates itself after each new piece of information, we don't have to wait 2 weeks for our results. Instead, each person we test will make the model better as we test, letting us use just our small friend group, rather than 10,000 people. It is far more efficient than the naive method we tried first, and our friends will get good lemonade much faster, resulting in less time lost drinking bad lemonade.

2.2 *How to build it*

When we start the experiment, each lemonade recipe has two values: The number of wins, or the number of times our friends finished the full glass of lemonade, and the number of losses, or the number of times they stopped after just a few sips. The two parameters of the Beta distribution α, β represent the number of wins and losses respectively.

Here's the clever part. We choose which lemonade recipe to present to our friends based on the Beta distribution itself. We start with Beta(1,1)[5] for both lemonades, representing an equal number of wins and losses and pick a random number using the Beta distribution. This gives us a number between zero and one for each recipe. If lemonade A's number is larger than that of lemonade B then we give our friend lemonade A to try. Otherwise we give them lemonade B. When they try the lemonade we then record whether they drink the whole glass or not in the α and β parameters for each lemonade. So, if the first person drank all of Lemonade A, then the distribution for Lemonade A would change to Beta(2,1). If they only took a sip, it would change to Beta(1,2) instead. As the number of wins for a lemonade gets higher, the numbers between 0 and 1 picked for that lemonade will be higher as well, making that lemonade more likely to be picked moving forward. The reverse is also true as the losses get higher, with the lemonade being less likely to get picked. After testing out these two recipes against a few friends we may have Beta(8,2)[6] for lemonade A and Beta(1, 6) for lemonade B. This means that our friends preferred lemonade A since the α value for A is higher than that of lemonade B[7] and our algorithm will now suggest lemonade A to our friends more often so

[5] $\frac{x^{1-1}(1-x)^{1-1}}{B(1,1)}$, with $B(1,1) = \frac{\Gamma(1)\Gamma(1)}{\Gamma(1)}$

[6] $\frac{x^{8-1}(1-x)^{2-1}}{B(8,2)}$, with $B(8,2) = \frac{\Gamma(8)\Gamma(2)}{\Gamma(16)}$

[7] $8 > 1$

they don't miss out on the tastiest recipe.

2.3 *Wrapping up*

In the end, we have a few lines of decision making code based on sound probabilistic principles that is guaranteed to converge. We can increase the satisfaction of our friends and quickly discover which lemonade is the best. In this case, according to our experiments, the answer is 1:1:5 which is 1 cup lemon juice, 1 cup sugar, and 5 cups of water. We didn't need to test out our method against 10,000 friends to be statistically confident. And in the end, we have a foolproof method that we can apply to any number of culinary needs.

3 Your Body

An Everyday Look at Populations

How well do you fit in with the average? Average height, weight, intelligence? Averages are used to describe a population and are thereby applied frequently to individuals of that population. It can be dangerous to live by averages because what if you are an outlier? In health care especially, experts use well tuned averages to make a diagnosis. But again, what if you are different?

In this chapter we have a story of a man whose life was saved because he didn't fit the average and spoke up.

3.1 What is a marker?

A Marker is a value, often a single digit real number, used to indicate the presence of a disease. A common marker in the western hemisphere is the change in size and shape of a mole on the skin. When the mole changes by a certain threshold, relative to itself and moles in general, it is often an indicator of some form of skin cancer or Melanoma. If you are careful in checking these moles regularly, you can often catch Melanoma before it becomes an issue. This is the story with most markers. They exist as a warning sign.

Prostate-specific antigen, or PSA, is a protein produced by normal and malignant cells in the prostate. The testing of PSA was introduced in 1987 and resulted in a spike in the number of reported cases of prostate cancer. While the test is merely a soft marker (its results are only one of many that could be used to diagnose prostate cancer) it is still an easy way for men over 50 to protect their health. Prostate cancer is insidious. Easy to treat early but tragically difficult in the later stages.

As with all markers, however, the value of the PSA test depends on the person. The national average used by many primary care physicians is 4.0 ng/ml[1]. If your PSA number is above this mark, then further invasive testing is performed. This means a sample of tissue is taken from the prostate and examine for cancerous cells. The PSA value was standardized because when men with PSA values between 2.5 and 4.0 ng/ml were tested, they were only found to have prostate cancer between 12 and 23% of the time which was considered a low enough risk.

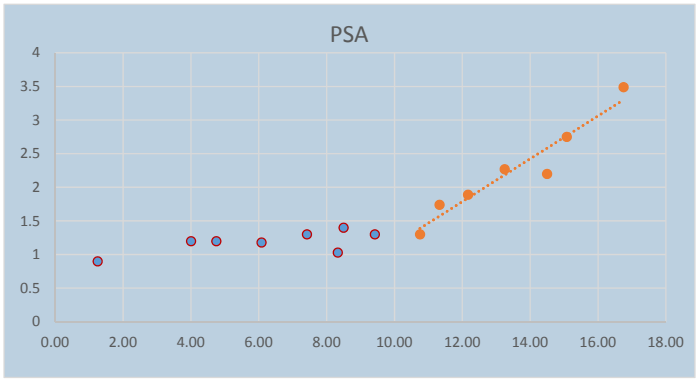
[1] nanograms per milliliter

With this background in place, we move to the story and see the

power of data science for our everyday health. Our friend tracked his PSA numbers for years (along with a number of other personal health test results). He kept a log handy where he recorded the results of every medical test and lab result he had. Over time, he had seen the PSA numbers steadily increase. They started under 1.0 ng/ml and moved to 1.5 ng/ml within a few years. His PSA peaked near 2.5 ng/ml and he was concerned.

Even though his personal PSA values were much lower than the threshold used by physicians, it was the trend he found worrying[2]. As a result, he went to his primary care physician for a regular physical where he got another PSA test. This one was over 3.0 ng/ml. The man told his doctor that he was greatly concerned with this result and that he wanted to do a biopsy to test for prostate cancer. The doctor, of course, told him there was nothing to worry about. His test was still much lower than the threshold of 4.0 ng/ml. But the man showed him this graph.

[2] The derivative matters



The doctor sat for a moment and then said, "Maybe you need to see a urologist." The man was shuffled quickly to another

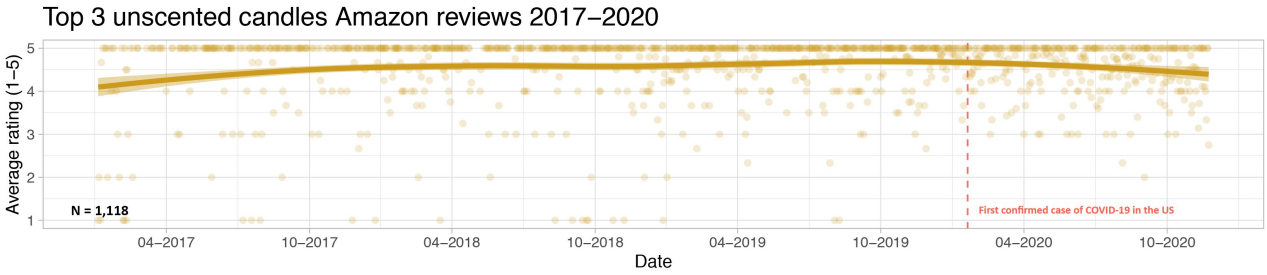
department. When the urologist saw this plot, he got genuinely excited (maybe not the best bedside manners). The doctor called in his assistant and asked, "How can we get this data for everyone?"

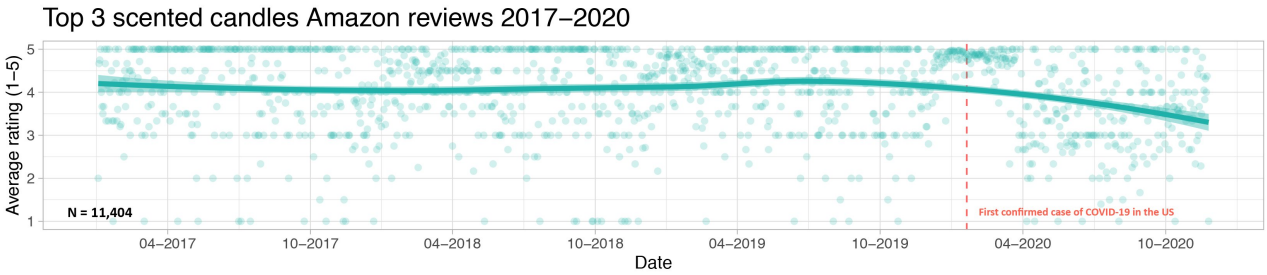
3.2 *Your markers vs the World*

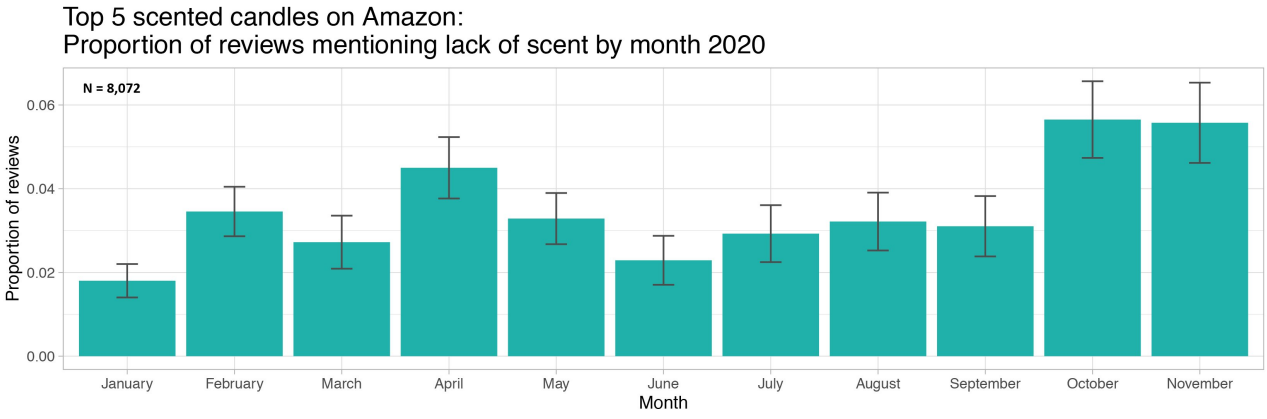
Let's briefly pause our story to talk about a little bit of statistics that we have glossed over during this discussion. In statistics, when a hypothesis is put forth, it is often framed with respect to some population. Many times this population is actually made up of people, but not always. For example, you could have a population of cars if you were trying to determine how long, on average, until the next oil change. Or, you may be able to detect an outbreak of a deadly virus by looking at changes in item purchases or other behaviors that would indicate the presence of the disease.

To illustrate that second example, Kate Petrova looked at the average reviews of scented candles on Amazon. When compared against unscented candle reviews, we see a large dip in the average rating on the scented candles. The timeline for this dip matches up with the progression of the recent pandemic. In other words, one of the markers for infection is the loss of smell. So, as the infection rate increases, we see a trend in dissatisfied people who can no longer smell their scented candles.

While this is an interesting anecdote, and the data is visually quite convincing, there is still much more you can do with this sort of analysis. If you had access to the location of reviews over time, you could quickly identify pockets of new infections and order tests or vaccines to an area. Interestingly, loss of smell doesn't mean you


Top 3 unscented candles Amazon reviews 2017–2020
Average rating (1–5)
N = 1,118
First confirmed case of COVID-19 in the US
04–2017
10–2017
04–2018
10–2018
04–2019
10–2019
04–2020
10–2020
Date


Top 3 scented candles Amazon reviews 2017–2020
Average rating (1–5)
N = 11,404
First confirmed case of COVID-19 in the US
04–2017
10–2017
04–2018
10–2018
04–2019
10–2019
04–2020
10–2020
Date


Top 5 scented candles on Amazon:
Proportion of reviews mentioning lack of scent by month 2020
N = 8,072
Proportion of reviews
January
February
March
April
May
June
July
August
September
October
November
Month

have been infected, but it is a good personal indicator for your own information.

When you are ready to do the actual number crunching analysis, it's unlikely you will have access to the entire population. You typically don't have the budget to test every car on the road. So instead, you take a sample from the population. You do this by selecting randomly, taking some cars from Utah, some from California, some from New York, etc., and you look at enough of them to ensure that the sampling of cars is representative of the population as a whole.

Analyzing the relation between populations and samples is important, and one way to do this to by using the *central tendency* that we will discuss. The one most people are familiar with is the **mean** or average of the data, but the average can sometimes be misleading. If you know which measure was used when a population statistic is given, then you can determine how your own health relates.

3.2.1 *Mean*

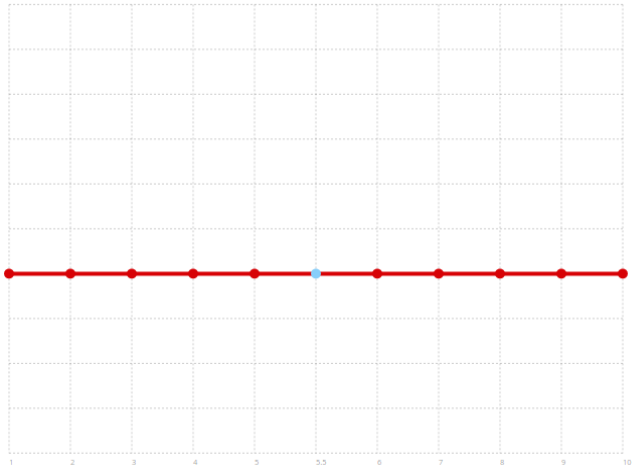
The mean is the measure of central tendency that is used most often. It's often called the average. The technical term is the arithmetic mean. It takes the form of the following equation:

$$\hat{\mu} = \frac{1}{n} \sum_i x_i$$

where you add up all the data points (x_i), and then divide by the number of them there are (n). It is the expected value for some distribution. You can see that for a uniform distribution, the mean is smack dab in the middle (blue dot). However, the problem with the

mean is that it is heavily effected[3] by the presence of outliers. If you were to take the mean of the net worth of you, me, and Warren Buffet, it would put our average net worth in the hundreds of millions of dollars, well above what you or I make, I think.

[3] skewed



3.2.2 *Median*

The median is much better in the presence of Buffet sized outliers. When you calculate the median, you sort the data from smallest to largest and choose the value in the exact center. If your data has a large number of 0 values and extreme large values, then the median still captures the center values well. One big example is financial well being of the US population. Many people are struggling financially and couldn't cover an unexpected $500 expense. However, according to the Federal Government Insurance website, the average American family's savings account balance is $41,700.

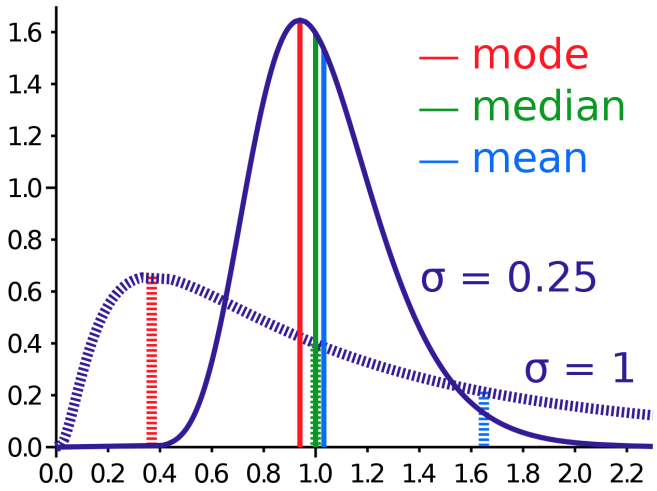
Those two data points don't make sense. How can the average American struggle to cover a $500 expense but also have $41,700 in savings? You got it. Because the median savings account balance is

only $5,300 (as of 2019). The wealthy outliers drag up the average while the median better represents the financial struggles many of us face.

3.2.3 *Mode*

Another measure of centrality for a population or sample is the Mode. It is simply the most common element in your dataset. So, if 50% of the population has 1 car, 40% have 2 or more cars, and 10% have no cars, the mode would have to be 1 car. This is the value with the highest count.

```
from collections import Counter
mode = Counter(data).most_common(1)
```



In this figure, from Wikipedia, we have a visual representation of

the different values. When your data is skewed one way or another the measures of center can be quite different. This means that in the slightly dashed line, the mode is far to the left while the median is in the center at 1.0 and the mean is far to the right.

3.2.4 *Geometric Mean*

The final measure of center you should be aware of for your personal health data is called the geometric mean. It is analogous to the arithmetic mean we saw earlier, except it uses multiplication instead of addition

$$\mu = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

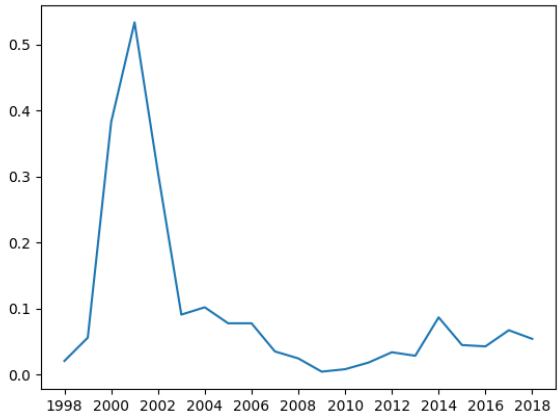
. This measure is often used when you want to compare items that are on different scales. It's important to use it on positive numbers, otherwise weird things might happen. Let's take a company's financial data as a measure of health. We'll use this as a final example before continuing our story.

Let's say you are interested in giving a single marker of your company's health. We have information about long term debt, revenue, and the size of the company (market cap). When we make measures that combines debt and revenue together, we will want to *normalize* based on the size of the company, because while a big company may make more money than a smaller one, the big company can still be under performing compared to its size.

$$\frac{geom\_mean(debt\_lt, revenue)}{mean(mkt\_cap)}$$

This measure serves as a warning sign. If the value gets large then

our company is less healthy but if the value is small than we're doing well. Let's look again at data from Congo.



This measure is not perfect for company health because it also can measure some idea of growth vs profit oriented organizations. Either way, we were able to use simple measures of center to analyze a trend of a single entity and not a whole population.

3.3 *Health Matters*

Our friend was sitting with the urologist. The urologist wanted to have everyday data for all his patients. As a result of their conversation, our friend decided to get a biopsy and they found out that his PSA levels were right and he had pancreatic cancer. They were lower than the population threshold of 4.0 ng/ml, but they trended up in a worrying manner. Our friend had his pancreas removed and they found that the cancer had spread throughout the organ but had not left it.

This means they had caught the cancer in time, with not a moment

to lose. It was a savvy bit of personal statistics by our friend to track his health over time (which he does for a number of lab results).

Your personal baseline for health tests may not reflect the population as a whole. Which is why personal statistics are so important. If you know yourself, through lab work and doctor's visits, you know when something is off. By tracking personal data and looking at it regularly, you can catch issues before you might have been able to otherwise. Sometimes the trend is a more reliable indicator than a comparison against the population.

4 Walking The Dog

An Everyday Look At Graphs

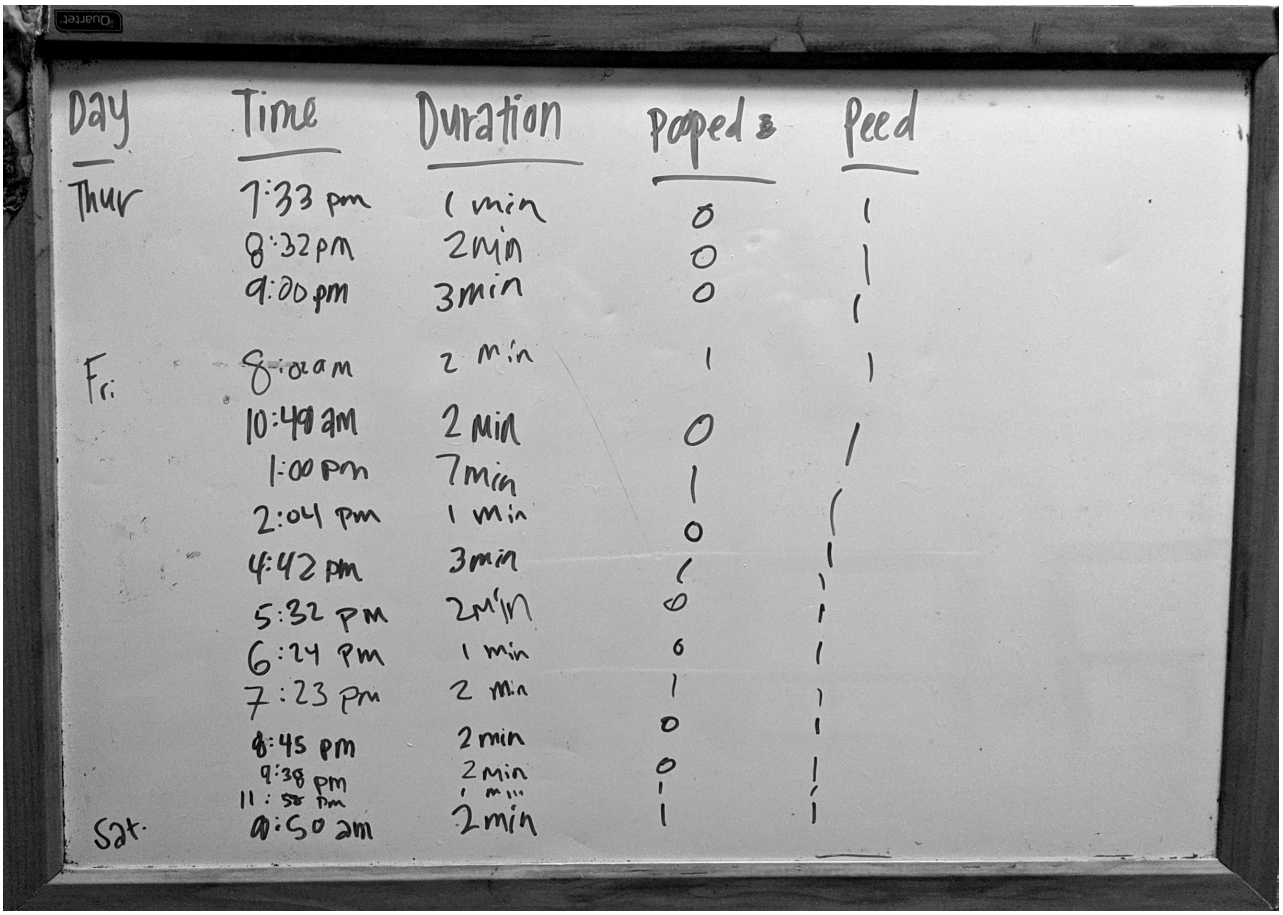
"Again? I just took you out?"

This was a very common phrase with our senior cocker spaniel named Lady. She loved to sniff leaves, put her head in the snow, and just stare at people.

We didn't have a fenced-in yard and so we took her out often to use the potty or just explore. We took her out so often, however, that we started to wonder if there was rhyme or reason to her asking.

My wife and I meticulously recorded her potty habits for two weeks to see if we could find anything interesting that might help us help her have a more enjoyable time.

4.1 *Data Collection*

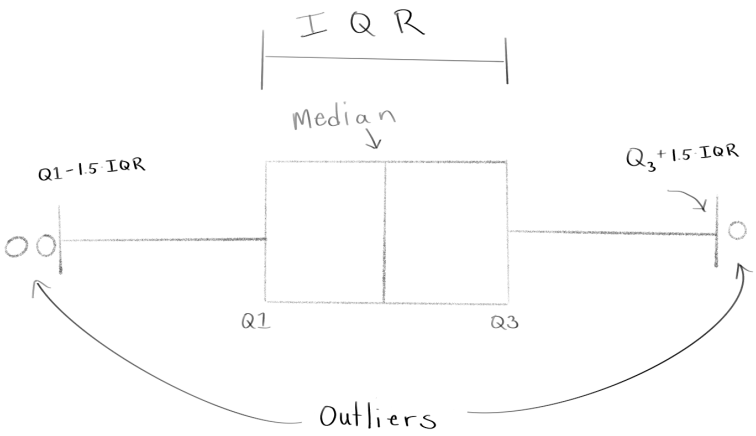


Sometimes the simple solutions are the best solutions. Added complexity often gets in the way of the process. Instead of using an app, spreadsheet, or computer vision model we decided to simply use a whiteboard to collect the data about our dog. The whiteboard sat right next to the door and we would offload to a CSV when the board was full. As you can see, we recorded fairly granular data which allowed us to analyze the results in a number of ways. First, we wanted to know if time of day had any effect on how long she wanted to go walking.

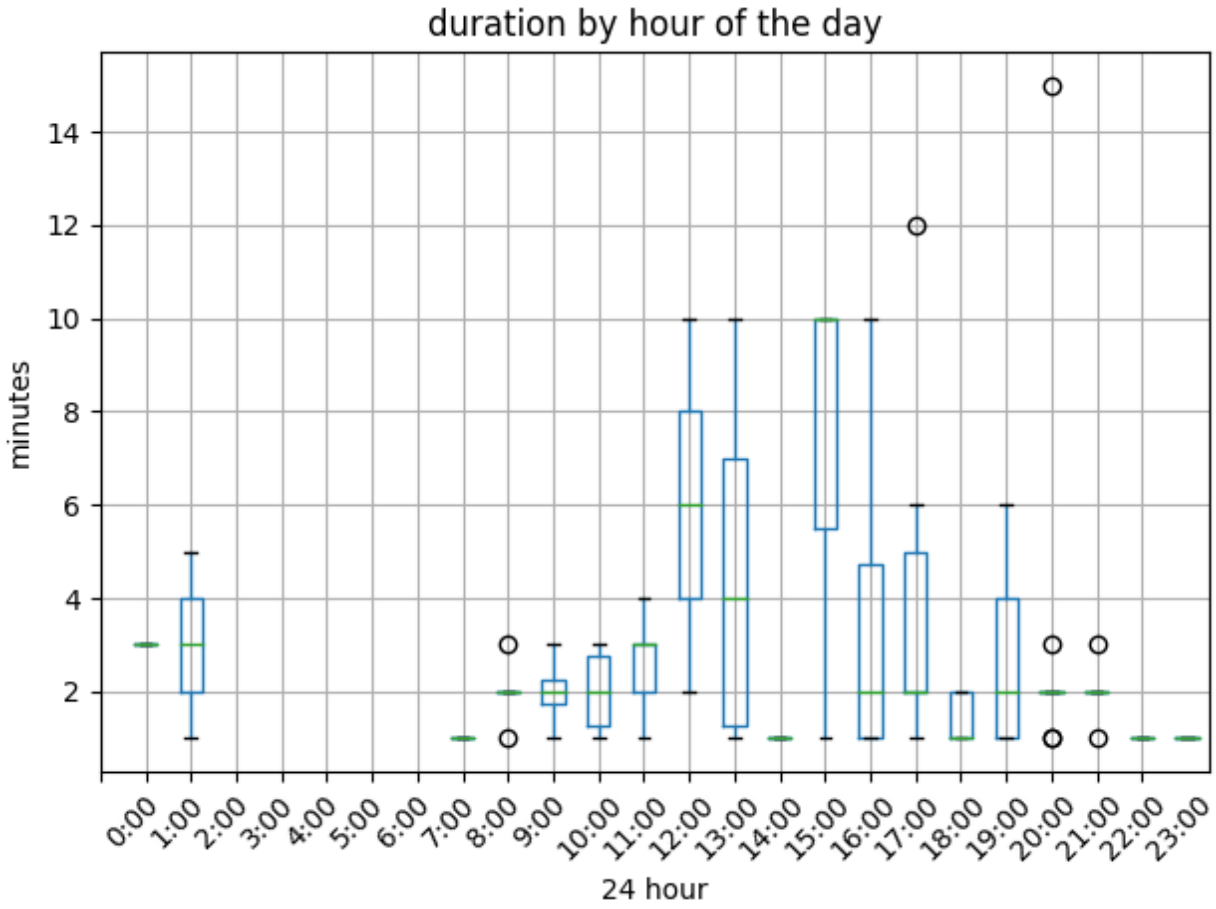
However, as you can see in the image of our whiteboard, we have to clean up our data a bit first. To do so, we came up with 24 sections, each representing an hour, and then assigned each potty trip to the hour section that it fell into. When graphing, we can then put the length of the walk in minutes on the vertical y-axis, and the hour of the day on the horizontal x-axis. Now, before we dive into the analysis; we need to talk briefly about a boxplot.

4.2 *Data Visualization*

A boxplot is a standardized way to show the distribution of discrete data points. It uses lines, circles, and boxes to represent central tendency and outliers. We therefore create a diagram, starting with the median as the line in the center. We then make a box around that, extending from Q1, the value that is halfway between the minimum of our data and the median to Q3. The lines that extend off the sides of the box stretch out to any points in the data that are up to 1.5 times the distance between Q1 and Q3, the value that is halfway between the maximum of our data and the median. Any points beyond these lines are considered to be outliers, and are marked with circles.


I Q R
Median
Q1 - 1.5 IQR
Q3 + 1.5 IQR
Q1
Q3
Outliers

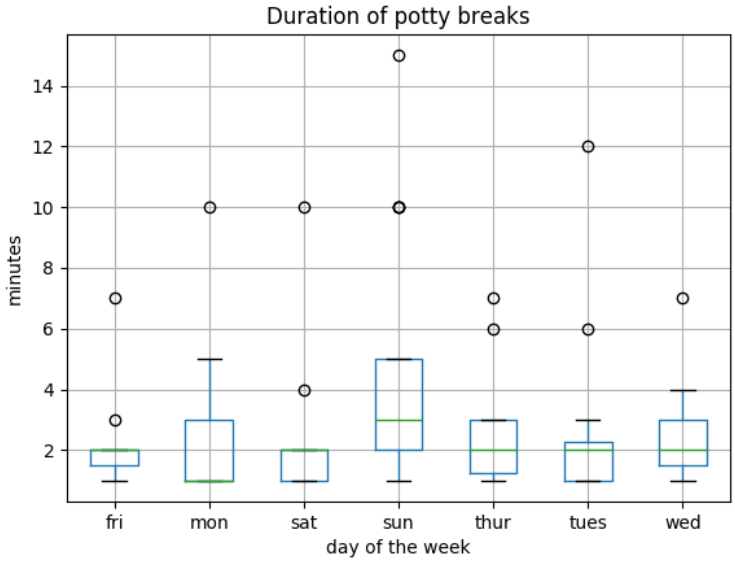
With that in mind, we can stack the boxplots vertically and observe trends of data distribution over time and across bins.



There are a few interesting tidbits to glean from this graph. First of which, yes · · · we had to take her out in the middle of the night (0 - 2), as shown by the boxplot information over on the far left side of the graph. While we miss her dearly, our sleep has definitely improved since she passed on. Secondly, there is a distinct bimodality to the data, meaning that there are two peaks, with a gap in between, seen right around 12 Noon on the graph. There are a few possible explanations for this. First, she liked to nap around that time

and often would go out before and after a nap. However, a more likely explanation is that we would be eating lunch. She was glued to our side when we were eating, and so didn't ask to go out. We would take her for long walks after lunch and after work which shows the large skew and outliers towards the end of the day.

After the success of the first analysis, let's now look and see if there are any differences across days of the week. We group the data by day of the week and again plot the duration, in minutes, on the y axis. It is gratifying to see our Sunday family walks reflected so clearly in the data. The higher median, and larger distance to Q3 imply longer walks in general. Somewhat surprisingly, Saturday had shorter walks than most week days.

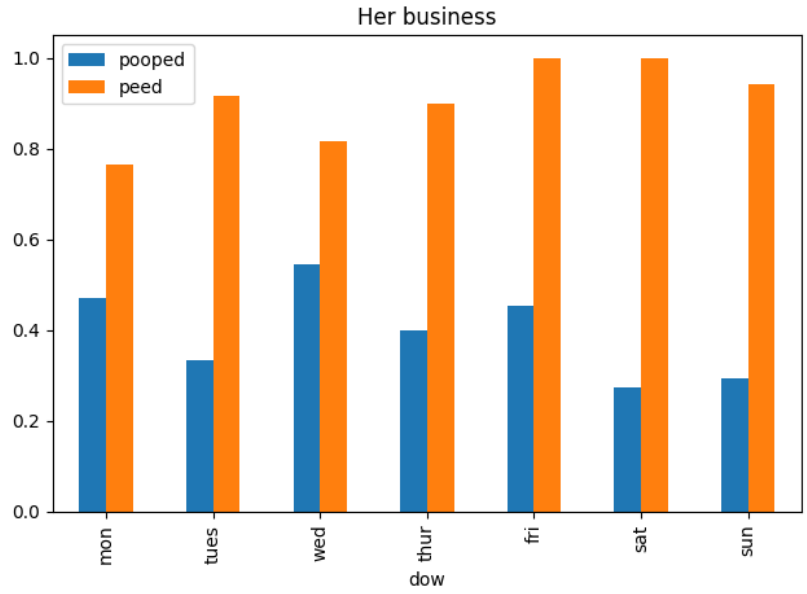


It's also interesting to note that there are a number of outliers on the longer end of walks. Meaning most of the walks we took were short, while only a few (one or two per day) were much longer. She was also 14 years old, and so she didn't quite have the stamina she

used to so 15 minutes was about as long as she could go.

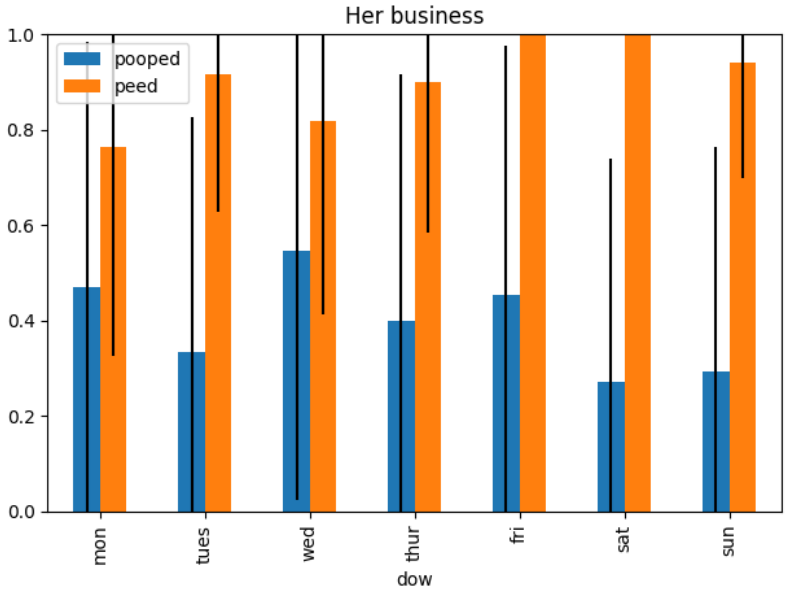
4.2.1 *Her Business*

For this next section, we'll be talking about her actually doing her business. Feel free to skip ahead. My wife and I also dutifully recorded what Lady did on each of her outings.

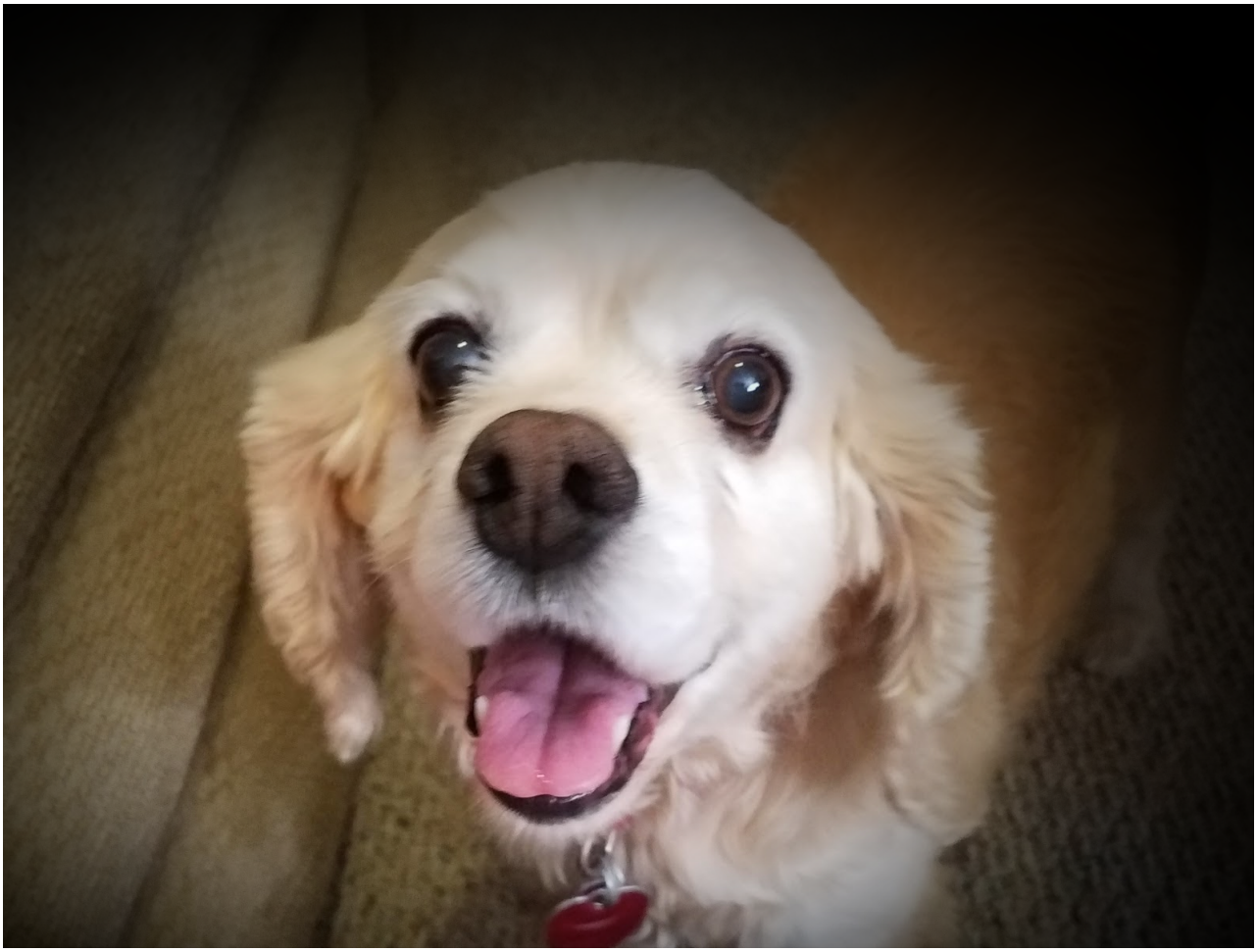


Sometimes she would just sniff, other times she would go 1 and 2. If we break that information up by day of the week, we see an interesting trend. She initially seemed slightly less likely to go number 2 on the weekend. Which seemed quite odd. However, the problem we see here is that we are looking at the average value with no respect to the standard deviation. We aren't seeing the whole picture.

Once we add the standard deviation to the mean value, we see that there is not really any trend across the days and she went



whenever she pleased. In the end, Lady was a wonderful companion and we were able to preemptively take her out after lunch so she was happy and comfortable. While we miss her dearly, in a way she lives on through this analysis and now you have her as part of your own story.



5 ODEs On A Diet

An Everyday Look At DiffEq

ODEs, or Ordinary Differential Equations, are mathematical tools used to relate how something changes with the thing itself. So you could calculate how hot your car tires get when accelerating from a stop light, or how many new bunnies will be born on your rabbit farm.

"Now wait just a moment" you might say. "Differential equations are just for engineers and physicists, not for data scientists and statisticians!" While this has been true in the mainstream tutorials, blogs, and curriculum; it could not be further from the truth in general.

To show what I mean by that, we'll consider the problem of diets. We, as a society, often talk about the 2,000 calorie per day diet as if it were a universal truth. Where does that number come from, and how does it relate to our personal weight loss journey?

5.1 *ODEs for data*

As we know, the purpose of data science is to make decisions based on information gathered from data. ODEs on the other hand allow you to make decisions when you may not have data. You can write an ODE to describe a process that occurs in nature, then use the results you calculate to make decisions about what may happen in the future.

Differential equations are useful for describing continuous things. These can be anything from chemical reactions to the motion of the planets. You can use them to describe how things deform when put under pressure or to describe how populations vary over time.

It is also the case that people who study statistics typically aren't the same people that study differential equations. However, many of the same skills are useful in both fields. You can quantify uncertainty, talk about long term behavior, and in the end you generally need to make a lot of simplifying assumptions.

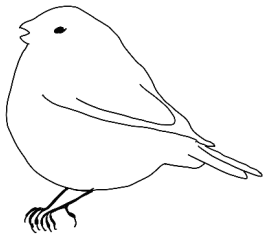
5.2 *A simple example*

Imagine you have a hobby of bird-watching. You enjoy spending time out in nature with your binoculars and picnic lunch. As part of this hobby, you have kept diligent notes on the species that live nearby. You know there are two main bird groups that spend time in the local area: Advers and Midgens.

Advers

You're ready to take your hobby to the next level. You decide you want to figure out the population of the various groups and see how

they interact with one another. One group, the Midgens, seems to be less frequent than they were before. While the Advers have been more and more noisy with larger groups than you've ever seen.



Midgen

You initially try to count the birds you see on each day. After just a few hours of counting, you realize you are double counting often and can't seem to get a reliable estimate.

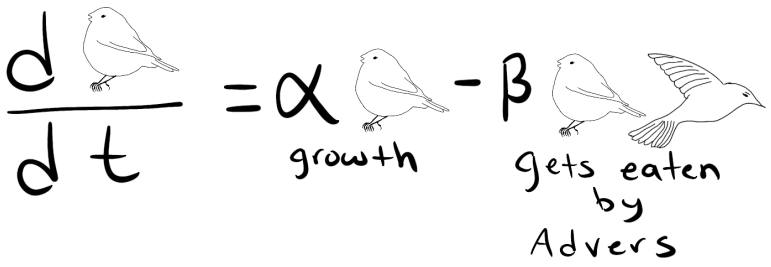
Thankfully, there is an ODE model called the predator-prey model which can help in this situation. It was originally designed by Lotka and Volterra and is an ODE that describes how populations of interacting predators and prey grows and shrinks over time.

Without any competition, the Midgens will grow unchecked. However, luckily they are kept in check by the Advers. So the change in the Midgen population is governed by their natural growth rate, and the hunting rate between the Midgens and Advers.

Secondly, the Advers growth is determined by how much food they get from the Midgens. However, the Advers are territorial and fight over the best nesting space. This means that their population decreases when there are more Advers around due to infighting and quarrels.

With these sketches in mind, you can write your first ordinary differential equation: The predator-prey model.

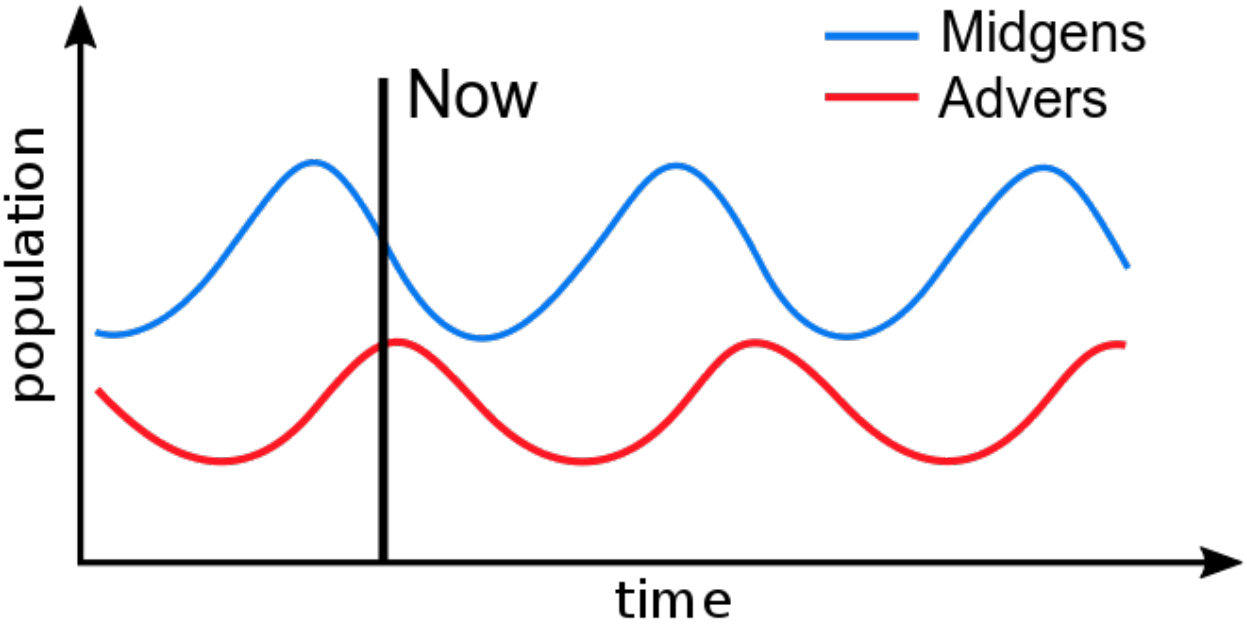
In this ODE, there are 4 parameters to which we need to assign values. α represents the natural growth rate of the Midgens while β represents how often Midgens and Advers are found in contact with one another. These parameters typically take on values between 0 and 1 where 1 represents 100% interaction or growth. The next parameter delta δ represents how much the Advers' population


growth
gets eaten by Advers


increase from food
decrease with competition

grows by hunting the Midgens. Finally, the gamma γ parameter is used to represent how the Advers die off from fighting one another.

After you record some values for $\alpha, \beta, \delta, \gamma$ which is much easier than counting the entire population, you simulate the model on the computer with a few lines of code (mostly ODE solver import statements) and you produce a beautiful graph.



You realize your careful bird-watching habit is showing you the cyclical nature of the two populations. You are seen a dip in the Midgen group and a growth from the Advers. With very little additional effort, your ODE model can be applied into the future.

5.3 *From Birds to Weight*

This section contains a deeper mathematical dive into a particular ODE that will allow you to calculate the amount of calories you

personally should consume to hit your goal weight. It is based off a number of bodily factors and uses the change in energy stored in your body, over time, to determine how your weight will change. This section is involved, so feel free to move on, but the pay off is worth it for those interested readers.

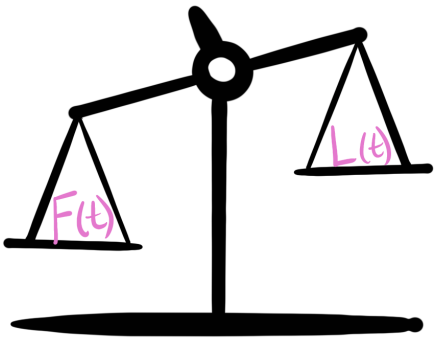
We often read that 2,000 calories per day is a healthy diet for adults. Which is generally true. However, as we learned previously, each person is different in their own health. The caloric intake we need to gain or lose weight can be determined personally from a 'weight change' ODE.

This section will be math heavy, feel free to move on if this isn't interesting to you.

While we are talking about losing weight here, you could easily apply this model to gain weight (which I did successfully).

To lose weight you need to move more and eat less. This means your energy needs to be balanced in a certain way. The Energy Balance (EB) is the relationship between your Intake and Expense.

$$EB = EI - EE$$



If you have a positive EB then you will gain weight. If, however, you EB is negative than you will lose weight. Easy? Maybe.

We first need to build out a model of your own body by using just a few data points. Your body weight can be expressed as how much Fat tissue you have and how much Lean tissue you have at any one time. $BW(t) = F(t) + L(t)$. We then use the energy balance and body weight to determine how our fatty tissue and lean tissue change over

time.

$$\frac{dL}{dt} = \frac{p(t)EB(t)}{\rho_L}$$
$$\frac{dF}{dt} = \frac{(1-p(t))EB(t)}{\rho_F}$$

In this equation, we use $p(t)$ to determine what proportion of the energy balance changes the fatty or lean tissue. We then also have two energy density constants for fatty and lean tissue[1] $\rho_F = 9400$ kcal/kg, $\rho_L = 1800$ kcal/kg.

[1] I guess we can see why bears store fat to hibernate for the winter.

The solution of $p(t)$ is given by the so called Forbes' equation.[2]

[2] $C = 10.4\frac{\rho_L}{\rho_F}$

$$p(t) = \frac{C}{C + F(t)}$$

We then need to find a closed form expression for our EB. After we have the energy balance we can use our change in fatty and lean tissue to determine our energy intake. Calculating the energy expenditure is done using a combination of your physical activity level (A) and your personal resting metabolic rate (Mr).
$EE = A \times Mr$.

Many tests have you choose your activity level based on how much you exercise. They usually do so based on days per week. In this case, the activity level comes from an appropriately scaled $[1.40, 2.40]$ interval.[3]

[3] 1.40 - 1.69 is sedentary with little exercise, 1.70 - 1.99 is for active with regular exercise, 2.00 - 2.40 is for strenuous work or multiple hours of daily exercise.

Then, for the grand finale, the following can be used for the resting metabolic rate calculation. It is a combination of contributions from fatty tissue, lean tissue, energy intake, and changes in body composition. We use a number of standard coefficient weights, the

interested reader can peruse "Modeling weight-loss maintenance to help prevent body weight regain" and "Quantification of the effect of energy imbalance on bodyweight".[4]

[4] $\gamma_F = 3.2\text{kcal/kg/d}$, $\gamma_L = 22\text{kcal/kg/d}$, $\eta_F = 180\text{kcal/kg}$, $\eta_L = 230\text{kcal/kg}$, $\beta_{at} = 0.14$

$$Mr = \frac{EE}{A} = K + \gamma_F F(t) + \gamma_L L(t) + \eta_F \frac{dF}{dt} + \eta_L \frac{dL}{dt} + \beta_{at} EI$$

I will spare you the algebraic manipulations. To determine how many calories you need to intake to gain or lose weight at a determined rate, you can use the solution for EB.[5]

[5] you also solve this for K

$$EB(t) = \frac{(\frac{1}{A} - \beta_{at} EI - K - \gamma_F F(t) - \gamma_L L(t))}{\frac{\eta_F}{\rho_F}(1 - p(t)) + \frac{\eta_L}{\rho_L} p(t) + \frac{1}{A}}$$

Then, by using your BMI, age, an ODE solver, and personal factors, you can solve for your own weight loss journey.

I solved for 3,000 calories a day in my personal, aggressive, weight gain plan. I was able to follow it (although it was quite challenging) and move much closer to a healthy weight.

In the end, your personal caloric intake is determined by many personal factors such has height, weight, age, activity level, and more. By using this math, you can figure out how to make a healthy weight. It takes a lot of effort to maintain your health. By using all of the tools at your disposal, you can give yourself the best chance to meet your goals. You also have an unlikely partner in the humble ODE.

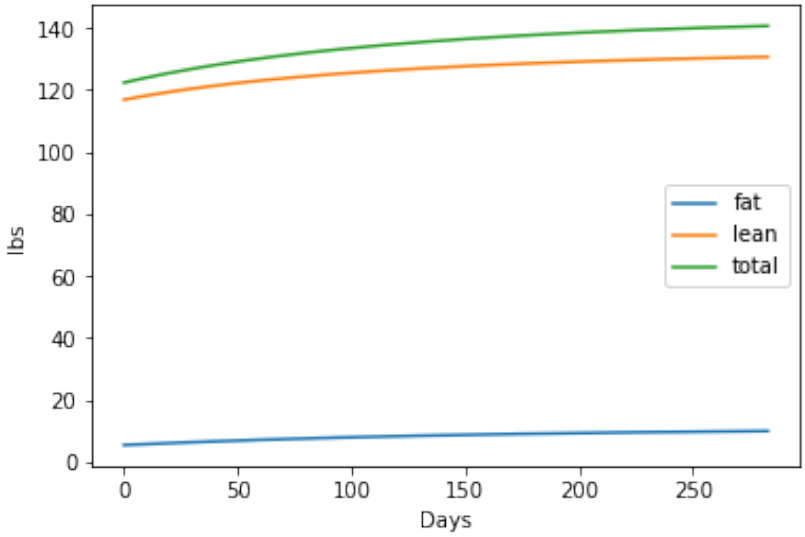
```
import numpy as np

def forbes(F):
    C1 = C * rho_L / rho_F
    return C1 / (C1 + F)

def energy_balance(F, L, EI, A):
    p = forbes(F)
    a1 = (1. / A - beta_AT) * EI - K - gamma_F * F - gamma_L * L
    a2 = (1 - p) * eta_F / rho_F + p * eta_L / rho_L + 1. / A
    return a1 / a2

def weight_odesystem(t, y, EI, A):
    F, L = y[0], y[1]
    p, EB = forbes(F), energy_balance(F, L, EI, A)
    return np.array([(1 - p) * EB / rho_F , p * EB / rho_L])

def fat_mass(BW, age, H, sex):
    BMI = BW / H**2.
    if sex == 'male':
        return BW * (-103.91 + 37.31 * log(BMI) + 0.14 * age) / 100
    else:
        return BW * (-102.01 + 39.96 * log(BMI) + 0.14 * age) / 100
```


140
120
100
80
60
40
20
0
lbs
0
50
100
150
200
250
Days
fat
lean
total

```
from scipy.integrate import ode
from scipy.integrate import odeint
import numpy as np
from math import log

# constants
rho_F = 9400.
rho_L = 1800.
gamma_F = 3.2
gamma_L = 22.
eta_F = 180.
eta_L = 230.
C = 10.4 # Forbes constant
beta_AT = 0.14 # Adaptive Thermogenesis
beta_TEF = 0.1 # Thermic Effect of Feeding
K = 0

gender = 'male'
age = 28
height = 1.8288 # in meters
weight = 55.610425 # in kg
A = 1.7
days = 283 # goal
EI = 3000 # test on a single calorie in value
F0 = fat_mass(weight, age, height, gender)
L0 = weight - F0
y0 = [F0, L0]
tf = days

t = np.linspace(0,tf,tf)
y = np.zeros((len(t), len(y0)))

y[0,:] = y0

weight_loss_ode = lambda t, y:weight_odesystem(t, y, EI, A)
w_l_solver = ode(weight_loss_ode).set_integrator('dopri5')
w_l_solver.set_initial_value(y0, 0)

for j in range(1, len(t)):
    y[j,:] = w_l_solver.integrate((t[j]))

np.save('weight_goal.npy',(y[:,0]+y[:,1])*2.2) # save the data

plt.plot(t, y[:,0]*2.2, label='fat')
plt.plot(t, y[:,1]*2.2, label='lean')
plt.plot(t, (y[:,0]+y[:,1])*2.2, label='total')
plt.legend()
plt.xlabel('Days')
plt.ylabel('lbs')
plt.show()
```

You can see your final weight following this plan in y and test different caloric EI values to determine how much you need to eat to hit your goal weight.

6 The Way You Do That Walk

An Everyday Look At Time Series

With the advent of smart devices in the home that monitor audio signals, many people are more aware of their privacy. People have begun to clean up their online data and take back ownership of many key services.

However, there is a lesser known privacy issue that follows you around wherever you go. In this chapter, we talk about how I can tell who you are just by how you walk with your phone in your pocket.

6.1 Timeseries gone by

A time series is a collection of data points which are ordered in time. Time is an important feature of the data. Typically, time series are interesting because we often try to predict the future. When we forecast[1], we are hoping to predict the value of something over time.

In much of everyday data science, our time series data will be discrete[2]. Meaning the data will be generated by measurements at potentially irregular intervals.

There are a couple of important pieces of information that need to be considered with respect to time series.

- Stationarity
- Seasonality
- Autocorrelation

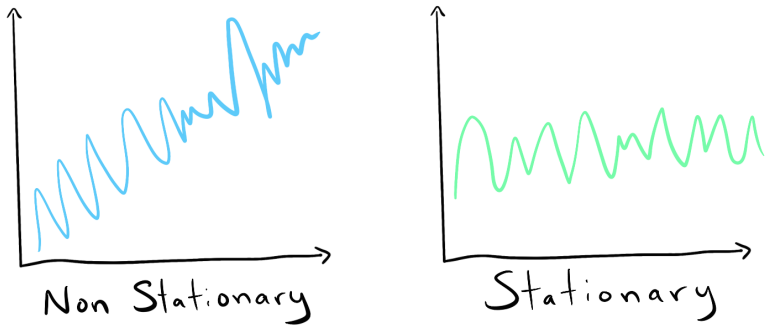
[1] **Forecasting** (4-caast) *Noun* • A conjecture as to something in the future

[2] Two random variables were talking at a restaurant. They thought they were being discrete, but I could hear their chatter continuously.

6.1.1 Stationarity

As we have discussed previously you can describe data using some statistics. Mean, variance, median, etc. A time series is said to be stationary if the mean and variance don't change over time.

This aspect of a time series is important for many modeling scenarios. Many time series aren't stationary, the world is constantly changing and as a result the values we measure change over time as well. However, there are a number of transforms we can do that will make them stationary and suitable for our techniques.

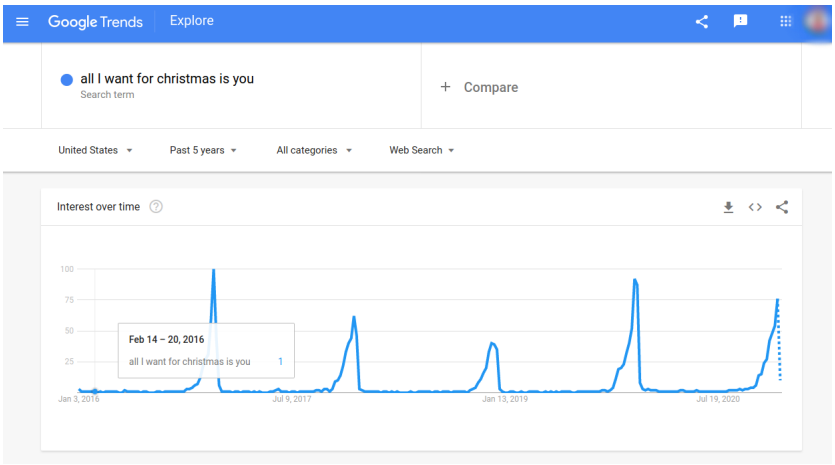


6.1.2 *Seasonality*

Sometimes known as periodicity. A seasonal time series has repeating fluctuations over time. I imagine a sine wave as the prototypical example for a seasonal time series. Another example is online shopping, which peaks during the holiday seasons. As a fun example, if you look at the trends for the smash hit "All I Want For Christmas Is You" by Mariah Carey, you'll see a perfect example of a seasonal time series[3].

[3] There is just one thing I need

The spikes occur towards the end of every year and come with a vengeance. This trend is the bane of retail workers around the world, but illustrates the seasonality well. This is a hard feature of time series to deal with because the spikes interfere with many of the simpler prediction methods and require more sophisticated tools.



6.1.3 *Autocorrelation*

Autocorrelation is the correlation, or connection, of a time series with some delayed copy of itself. It is measured as the distance, or similarity, between data points in the time series as a function of how far the points from one another in time. A simple illustrative example might be temperature. If there is a high temperature now, it is likely that tomorrow's temperature will be similarly high, while temperatures further in the future are unknown and less related to the current temperature. We won't focus much on autocorrelation here, but if your prediction quality for a time series problem is poor, you should explore whether you have an autocorrelation issue.

6.2 *Models*

The time series modeling community has a huge volume of work on different models that are all used in special scenarios. These models all have acronyms[4] which capture some information about the set up and assumptions.

[4] see "The ARMA alphabet soup: A tour of ARMA model variants"

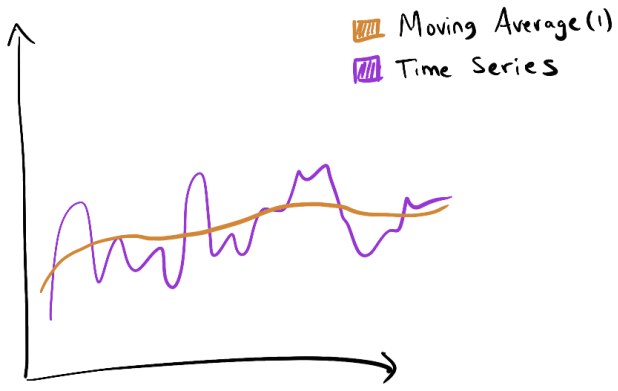
The first acronym, which is also the simplest model, is the moving average (MA) model. The model is relatively simple, the next prediction is the mean of the past observations.

$$x_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

This can be a surprisingly robust model of a time series' long term behavior. μ is the mean of the time series thus far, while the θ terms are called the parameters of the model. The ϵ values are error terms. In a simple example, if it rained 2 inches Sunday, 4 inches Monday, and 7 inches Tuesday we can predict that $\mu = 4.33$ which can be used along with a random error sample as our naive prediction for the amount of rain for Wednesday[5].

[5] It would be between 3 and 5 inches. We have a high range because of the small number of samples which have a high variance between them

The MA model has one hyperparameter to choose q. This is referred to as the *order* of the model. It is fairly common to use MA(1).



The second acronym is for auto regressive (AR) models. Instead of

the prediction being the mean of previous observed values. The AR model says the prediction is a linear combination of its own previous values. The difference is subtle, but the equation shows the change.

$$x_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t$$

This model also shows surprisingly strong performance in a number of cases. Again, we have ϕ_i as the model parameters and ϵ_t is the noise. We also introduce a constant c and the order of the AR model p. This model can capture shocks and outliers slightly better than the MA model.

However, the real power comes when you combine the two into an ARMA model which is one of the first models you should reach for in your time series tool kit.

6.3 *Where do you keep your ARMAs?*

With a stationary model that exhibits low autocorrelation and seasonality, the ARMA(p,q)[6] model performs extremely well on a number of time series prediction tasks. There are a number of different ways to write this model, but the most common ARMA(1,1) model can be written simply as follows.

[6] Other variants include ARIMA, SARMA, PARMA, ARFIMA, VARMA, INARMA, etc

$$X_t = \mu + \phi X_{t-1} + \epsilon_t + \theta \epsilon_{t-1}$$

Now, to fit the parameters of an ARMA model is fairly advanced, and we don't cover the details here. However, we fit the parameters of an ARMA model using the exact maximum likelihood with a

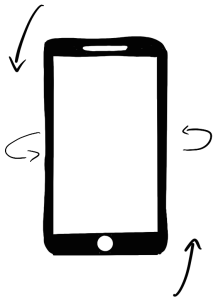
Kalman filter to find the best parameters according to the Akaike information criterion. This seems complex, but we are bascially finding the range of values that best fit the data we've seen so far, which we can them project forward with the ARMA model to get a prediction.

The standard way to actually fit an ARMA model is to use the statsmodels.tsa.arima_model.ARMA.fit function.

6.4 *The walking prediction*

Your cellphone is a super computer that could guide 120,000,000 Apollo space crafts to the moon simultaneously. It is equipped with professional grade cameras, lightning fast processors, and a whole array of sensors.

One of these sensors is called an accelerometer and it is used to measure acceleration forces, movement, and vibrations. One of the primary uses is for automatic screen rotation when you orient your phone or tablet to watch a movie.



The data generated by an accelerometer can be formed into a time series with a time value and an acceleration value (aT). The acceleration is a single value that you can read from the

accelerometer. There are also recorded values for rotations and vibrations, however, as we'll see here, the acceleration has such a high informational density that you can use it as a sole feature in the task to follow.

	Andrew-aT	Mitch-aT
0	3.485	8.872
1	3.156	8.690
2	2.819	8.290
3	2.418	7.867
4	1.946	7.465

This data was collected by putting a phone into our pocket and walking around for just a minute or so. The data can be exported remotely, or by plugging in to a computer. We took turns with the phone, gathering data, and collected it into a single folder.

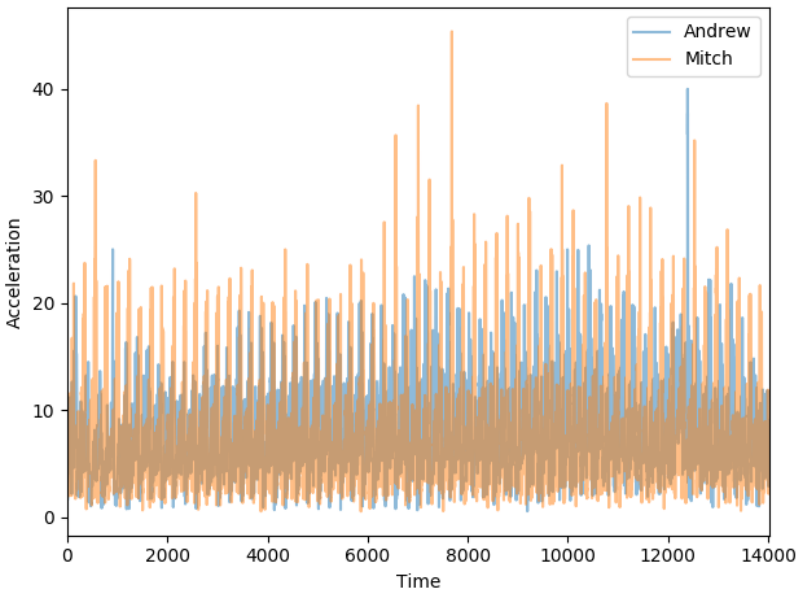
We have acceleration data for 26 friends, I've shown the data for myself and Mitchell Probst. Looking at descriptive statistics of that information, it may appear that the acceleration values are similar to one another, but they are actually sufficiently different which allows us to predict whose gait is whose.

The difference becomes more stark and easier to see when you plot the acceleration values over time. You can see, in blue, that my time series has a different overall shape than my friend Mitch's time series. The background vibrations may be similar, but when we take steps it is captured differently in the values of the acceleration.

This difference implies a key fact. We can take the accelerometer data from your phone and determine who you are. Your walk has enough information to extract. This is a fascinating piece of mathematics and a scary bit of privacy. What this implies is that it is

```
>> df['Andrew-aT'].describe()
count    14014.000000
mean         7.992693
std          3.781012
min          0.560000
25%          5.024250
50%          7.676000
75%         10.643250
max         39.997000

>> df['Mitch-aT'].describe()
count    14014.000000
mean         7.763417
std          4.711492
min          0.544000
25%          4.067000
50%          7.156000
75%         10.181750
max         45.370000
```



possible to know who is carrying your phone at any time.

How does that work?

We can extend the idea of time series prediction with an ARMA(p,q) model to the idea of time series classification with the same model. This means that, instead of worrying about what happens after the observed data, we care much more about the parameters we learned during the process, and seeing if the time series information has a similar shape to time series that we have recorded before.

For each person's data you get $X = (\phi, \theta, \mu, \sigma)$. You can then do any of your favorite classification techniques to determine who is carrying the phone at any one time. In my case, I did a simple distance comparison $match = \arg\min_{them} ||X_{me} - X_{them}||^2$ where I find the argument (them) that minimizes the distance to the person I'm trying to classify (me). This is done with a single for loop over the learned model parameters and achieves 96% accuracy.

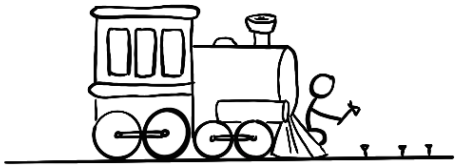
6.5 Conclusion

While the privacy implications for this are scary, there are many other cool applications of time series analysis that you can do in your everyday life.

If you look into how long you spend shopping for groceries or your monthly budget, you can see trends emerge which you can use to predict what will happen in the future.

However, a word of warning, there are many time series which are extremely hard to predict and can be frustrating if you're not

prepared. It sometimes feels like time series prediction and classification is like laying down the tracks while on the front of a speeding train.



7 *Your Resumé Lives in* $\mathbb{R}^{300}$

An Everyday Look At Vectors

During k12 education, we learn that math is memorization and algorithmic computation. Because of this, many people who aren't computationalists generalize to say they aren't "math people".

Math, in reality, is the study of relationships and conclusions. Distances and results. It's a beautiful interplay between what we know, and what we hope is true.

A huge recent advance in natural language processing is the ability to do math with words. Imagine if you could write the following equation

$$(\text{king} - \text{man}) + \text{woman}$$

and you would get the result of **Queen**. This is the power of word vectors and in this chapter we'll see how to use them to improve your resumé.

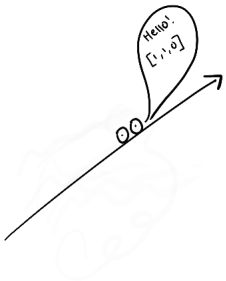
7.1 A bit of background - Math

A fundamental idea in mathematics is the idea of distance. Mathematicians are extremely interested in being able to tell how far apart two *things* are from one another. The idea is so important in mathematics they gave it a special[1] name.

[1] jargon

A metric is something you use to measure distances in math. It's typically represented with a d. So if you wanted to measure the distance between two things a and b you could write $d(a, b)$.

To be a bit more concrete. Let's first introduce the idea of a vector.



A vector can be thought of as a list of numbers of a certain length. The length of the vector is called the **dimension**. So a vector that has 3 numbers is a 3 dimensional vector. As a result, if all the numbers are decimal valued, this vector would be an element of $\mathbb{R}^3$, which is all possible lists of 3 decimal numbers.

Because mathematicians sometimes don't get enough human contact, we often anthropomorphize objects. Vectors are no exception. So you will regularly see the phrase "lives in" with reference to a vector and it's larger space. The vector of 3 numbers is also said to live in $\mathbb{R}^3$.

It turns out that vectors are much more than just a list of numbers, they are a fascinating abstract object, see Foundations of Applied Mathematics Volume 1, but they also have a geometric interpretation.

Let's take a 3d vector[2] [1,1,0] and plot it inside of a cube.

[2] a vector that lives in $\mathbb{R}^3$

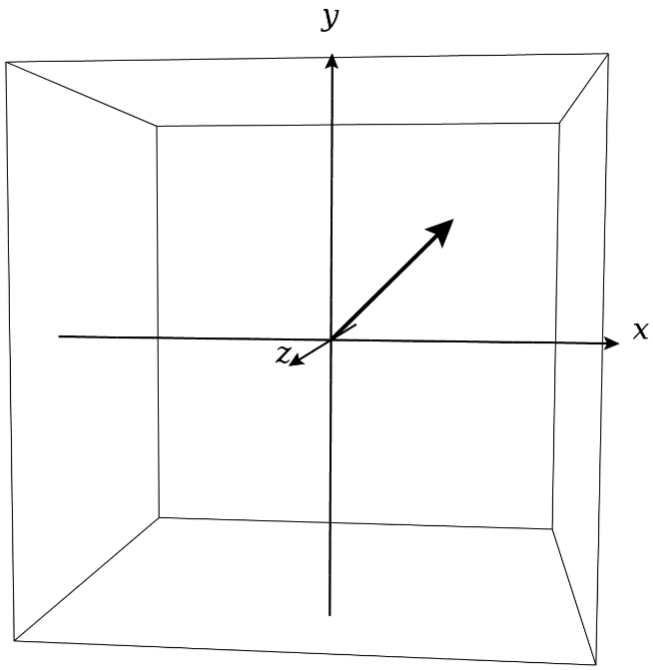
We go 1 step in x, 1 step in y, and 0 steps in z (out from the page).

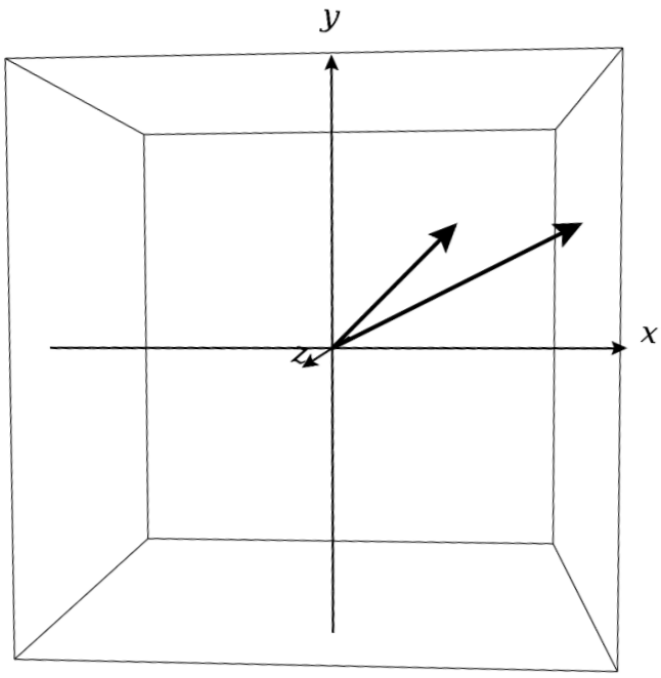
It turns out, you can use the results from our metric (i.e., Inner Product and Norm) to measure how similar this initial vector is with another vector [2,1,0].[3]

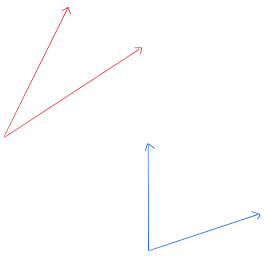
[3] The similarity between these two is 0.948683 which is pretty high since these distance values are between [0,1] in this case

To do so, we first realize that some vectors are "more similar" than others. As we can see the red vectors seem to be closer together in some sense. The blue ones seem less similar. Our intuition is correct. There are many ways to measure this similarity, a neat and intuitive one is called **Cosine Similarity** and measures the angle between the two vectors.

The larger the angle, like the blue vector pair, the less similar the two vectors are to one another. It's as simple as that.


y
x
z


y
x
z



$$\cos\theta = \frac{<a,b>}{||a|| \cdot ||b||} = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i a_i}\sqrt{\sum_{i=1}^{n} b_i b_i}}$$

The cosine angle between two vectors is calculated by finding the inner product between the first and second vectors and dividing by the length.

7.2 A bit of background - Word Vectors

Now that we can measure the distance between two vectors as their cosine similarity, we're ready for a big idea.

a word is characterized by the company it keeps

- Firth 1957

You could imagine rephrasing this quote to

the distance between words is a function of their neighbors

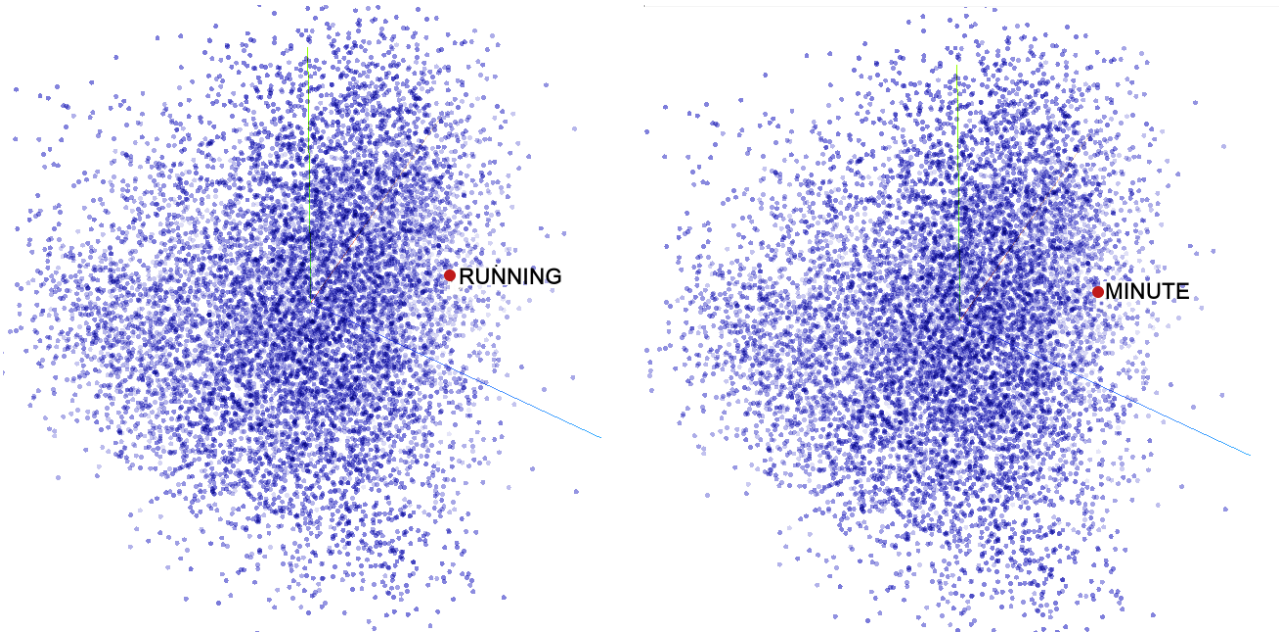
- Andrew 2021

While this is a bit silly of a translation, it turned out to be a pivotal discovery around 2013 and continues with work such as GPT-3 today.

If you take in a bunch of text, and clump the text together based on co-occurrence, then do some neural network magic, you get a vector space of words.

The distance between words is now semantically meaningful. For example, you can see that running and minute are close together in this 3d representation of the space[4]. This is because you often talk

[4] Calculated using PCA, you can also use TSNE or UMAP do visualize the space



about how long you run in terms of minutes.

Interestingly, there are many different word spaces that have slightly different properties. Some popular ones are Word2Vec, GloVe, BERT, ELMO, and more. The community really likes Sesame street.

For our purposes, we will use GloVe embeddings because they are high quality and readily available[5].

[5] we will be using the Spacy python package

7.2.1 *Deep dive into word similarity*

There is something amazing about seeing word vectors in action. They are relatively simple to work with because of some amazing open source packaging. If we start with a spacy object nlp = spacy.load('en_core_web_lg') which loads in a 300 dimensional GloVe space, we can then create and compare sentence similarities as if they

were vectors, it defaults to cosine similarity.

```
>> d1 = nlp("I use data science everyday")


>> d2 = nlp("after studying this book, I feel comfortable
applying data science methods in my day to day life")


>> d1.similarity(d2)
0.8566970196518339

>> d3 = nlp("dogs and cats have fun")

>> d1.similarity(d3)
0.5820495527582082
```

The astute reader will notice that we are comparing whole sentences, and not just individual words. It turns out, that in 300 dimensional space you can simply add up all the word vectors and get a pretty accurate representation of the entire sentence.

7·3 *Legalease demystified (detour)*

As an example of something we all encounter in our day to day life, let's look at **Terms and Conditions**. In truth, there are very few people who read through all of the terms and conditions before clicking accept[6]. Often times, we break land speed records by

[6] only 22% according to our surveys

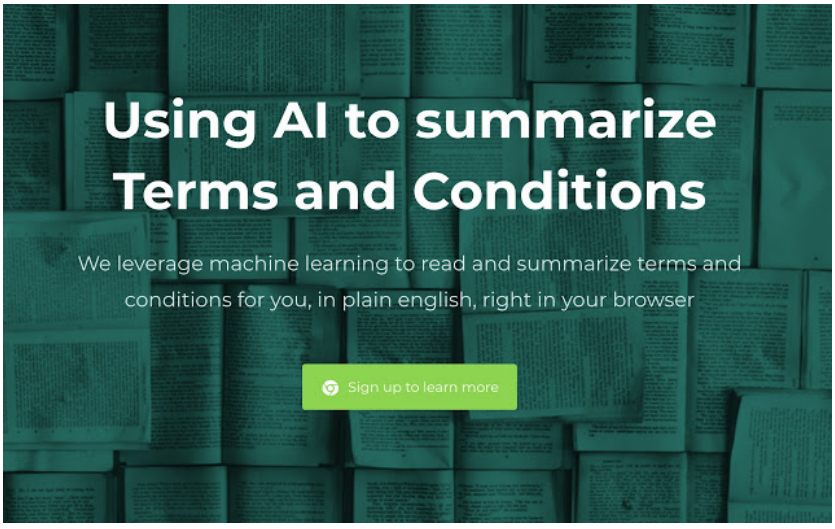
scrolling to the bottom so quickly.

In truth, it's not a big deal most of the time, however, there are some terms and conditions you really want to be aware of. There are services that provide human written summaries for some of the big services online, but this is an expensive process and can only cover a fraction of the online experiences we have daily.

It turns out, that word vectors can be used to summarize large bodies of text. Now, it's important to realize that this is an open problem and there are many people working to solve it. However, the technology is mature enough that you can get pretty good results.

In 2018, I started a company called Legal Leaf which used word embeddings to summarize the terms and conditions of any website you visit[7].

[7] and yes, I'm guilty of using the AI buzzword to generate hype



The idea was simple, and it worked as a chrome extension on any website. We had good performance and quickly grew to 2,000 users.

The entire code base is only $\sim$ 200 lines of code.

Over the course of the next 8 months, we grew the product and were acquired. It was a phenomenal journey and an application of data science to solve a problem I experienced regularly.

Over the years, I've used word embeddings on a number of legal tech projects with great success. It's a wide open frontier for applications.

7·4 *Job hunt*

They say a recruiter only looks at your resumé for 7 seconds before making a decision. This 7 seconds is a crucial time for you to make an impression. Some people use fancy designs to impress while others use their fancy credentials. In any case, you should do your best to make your resumé as close to perfect as possible.

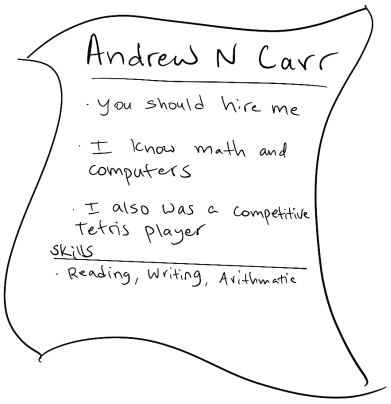
In 1510, Plato was teaching at the school in Athens. During his time there he proposed a theory of Forms. The idea, which is now widely known as the Platonic Ideal[8], is that there is a true or perfect Form for everything existing thing. He suggested that when we think about a horse, we are actually imagining the Form of this horse. It is this Form that allows us to classify objectsaround us.

[8] are CNNs just Platonic Form recognizers?

Some people believe there is an Ideal Platonic Resumé. They claim that if you can craft your resumé to be as close to this Form as possible, you have a much better chance at getting an interview.

I put that idea to the test. I searched for examples of technical resumés online that were touted as "perfect examples" of what a resumé should include.

One example, from Monster.com, for a mid-level computer



programmer shows how to properly explain your impact in a clear and concise way.

Developed and enhanced programs using Java, C and C++, contributing to solutions that streamlined processes, increased accuracy and lowered costs. Developed back-office application that saved client more than $125K annually.

So, let's take that as our Platonic Ideal of an entry on a resume. Part of what makes it an ideal is that it follows what is called the S.T.A.R principle (see below). Now, let's compare this to a bullet point from my own resumé:

I wrote code and made people money

This, of course, doesn't seem as good or eye-catching as the Platonic Ideal, but we can use the principle of semantic similarity to measure just how far away from the perfect bullet point we are. In this case, the similarity score is .77 with 1.0 being most similar. Let's make some edits, shown below, and you can see how the similarity between my bullet and the *best* bullet improves.

I started by trying to quantify my impact better. Often when

writing a resumé, a bullet point will follow the S.T.A.R principle.

S. T. A. R.

Each bullet point should start by outlining the situation and setting. Then, you explain the task you have to accomplish. Followed by the action you took and the final result.

In our example bullet point, we don't have an explicit task, but we can see all the pieces of our S. T. A. R. included.

Developed and enhanced programs using Java, C and C++, contributing to solutions that streamlined processes, increased accuracy and lowered costs. Developed back-office application that saved client more than $125K annually.

So in the case of my bullet point, I wrote that I had increased the speed of the system by 3× which was much faster.

This change brought my example closer in line with the ideal example. I continued writing my bullet point in line with the STAR principle and compared it against the ideal example I had found.

The progression of my bullet point quality, as measured by similarity to a 'great example', was steady and positive. By the end, I had a bullet point I am proud of and use currently in my resumé.

7·5 *Conclusion*

Distances are fundamental to mathematics. It turns out, they are really important in natural language processing as well. By using the distance between sentences, you can summarize, classify, and generalize.

Text	Similarity Score
I wrote code and made people money	.77
I wrote code that increased the speed of the system by 3x	.83
I wrote code that increased the speed of the system by 3x and reduced hardware costs by 63%	.88
Increased speed of system 3x resulting in a 63% reduction in hardware costs while handling 3 million daily requests by engineering asynchronous API using parallel processing and high performance computing techniques	.94

Using word vectors to give your resumé a *quality score* by comparing it to some perfect example is a great way to iteratively improve your chances of landing that dream job interview.

There are thousands of applications of distances and word vectors. Many companies are using them to great effect. Now you have the basic understand you need to use them to solve problems in your daily life.

Also, feel free to reach out if you want resumé help, I'm happy to offer advice and assistance.

8 The Olympics is Calling

An Everyday Look At Goals

As a recent graduate with a Master's degree in statitics, Jared Ward's thesis on optimal pacing strategies for the 2013 St George Marathon served as a guide to help him prepare and eventually quality for the 2016 Rio Olympic games.

Jared rigorously recorded data about his training over the course of years. In this chapter, we look at how we can use data science to meet our fitness goals and how Jared use stats to qualify for the Olympics.

8.1 *The power of goals*

Before diving into the analysis on our Olympian friend's build up to the games, we're going to take a detour and look at some data that supports the powerful effect of goal setting.

Every year, thousands of people set New Year's resolutions. They sign up for gym memberships and by exercise equipment. However, as you may have experienced, many of them call it quits in the first three months of exercise.

For many of us, it is a continuous cycle of starting and stopping.

There are, however, many people who set goals at the beginning of the year and build a new life long habit of fitness. What is the difference between these groups of people?

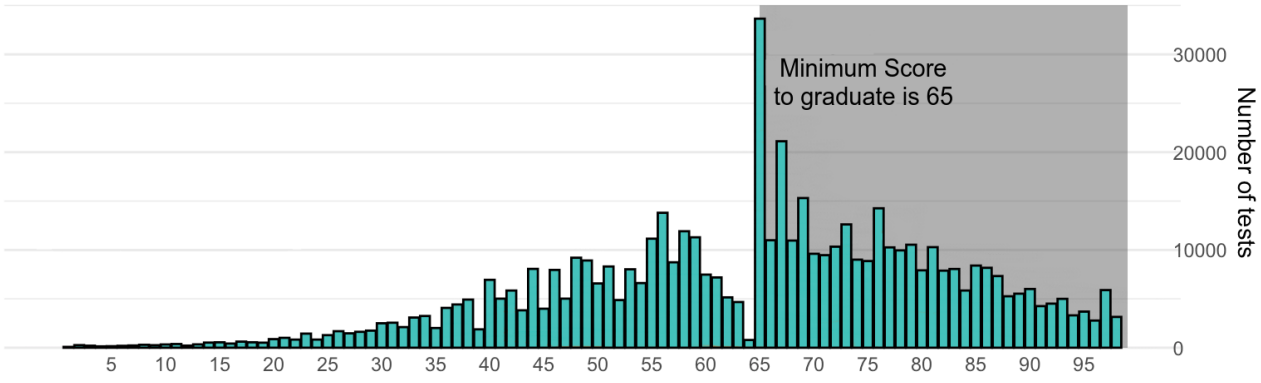
There are as many hypotheses as there are people. One constant seems to be shame[1]. If you are worried about being embarrassed in front of others, even when that embarrassment is mild, then you are more likely to accomplish your goals.

[1] often framed better as accountability

This can be expanded into a general rule. Having another person who is invested in your success helps with accomplishing your goals. In 2010, New York gathered data about average test scores across 5 tests required for a Regent diploma. Many students hope to pass these tests and get the credentials they've been working towards. By extension, the teachers are also invested in the students' success.

An interesting trend emerges in the data. It can be interpreted a number of ways with respect to goals and educational measurement, but it shows an astounding number of students barely passing the

exams and qualifying for the diploma.



Data from the New York Times

This can mean a number of things. The first being the more cynical interpretation that the teachers are passing people who barely make the cut off. Another could be that students put in the minimal required effort to pass.

If that is the case, then the power of the number 65 is motivating students to a certain level of performance.

In our own lives, we have personal thresholds for success. Sometimes they are set by others, often times they are set by ourselves as personal goals and benchmarks. By setting and recording[2] goals you can hit your threshold for success.

[2] and analyzing

8.2 *Optimal pacing strategies*

To return to Jared's journey to the 2016 Olympics we start in 2013. Jared was working towards a Master's degree in statistics. His Thesis analyzed the pacing strategy of marathon runners in the the St George Utah marathon.

The analysis consisted of "Bayesian multivariate regression [analyzing] posterior distributions of effect sizes on gender, age, and ability as well as first order interactions, on the relative pace strategy of participants". They had 5,819 participants and a number of features such as *elite* status or *boston qualification*. They fed this data into their model and fit a number of parameters to form a prior distribution[3].

[3] in a Bayes sense

They found an interesting difference between those that qualified for the Boston marathon and those that did not. Those that didn't qualify started the first half of the marathon much faster than the second. While those who hit the qualifying time ran an even pace and often ran faster in the later stages of the race.

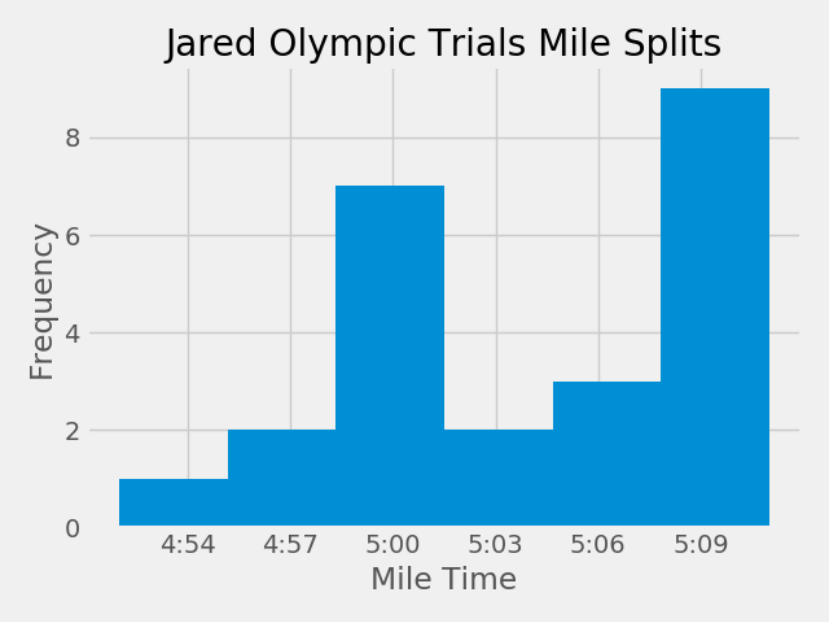
This is not causal, but it could be interpreted to mean that steady pacing matters more than a fast start, which can lead to burnout. The conclusions they drew served as a guideline for Jared's own qualification at the USA Olympic Trials where he ran consistent mile times throughout and was able to qualify.

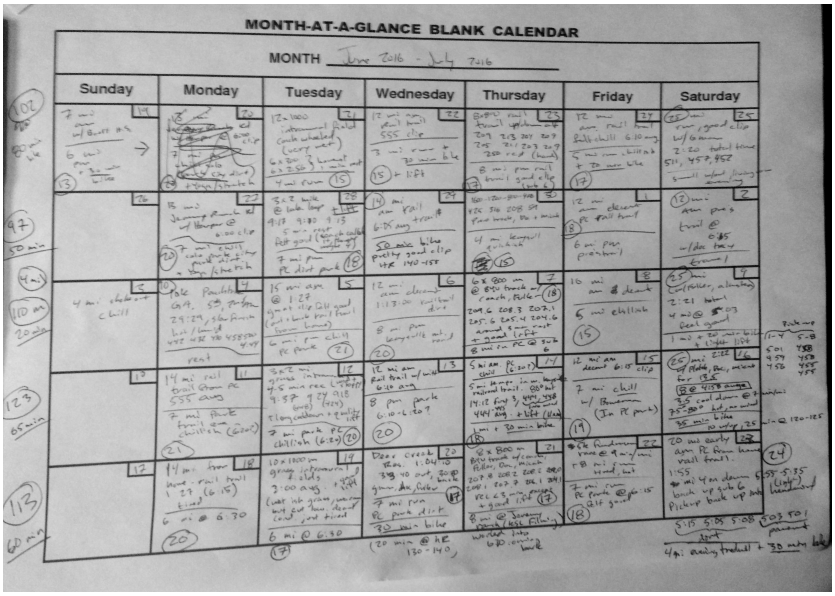
The next step was the build up in training for Rio.

8.3 *Recording the data*

Jared recorded daily workouts on calendars where he was often running twice per day and would often run over 100 miles per week.

As part of the data, he would also include a desired pace / feeling for the workout. He often used words like "chillish" to describe a reasonable pace but not too hard.


Jared Olympic Trials Mile Splits
Frequency
0
2
4
6
8
4:54
4:57
5:00
5:03
5:06
5:09
Mile Time


MONTH-AT-A-GLANCE BLANK CALENDAR
MONTH
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

8.4 2016 Rio Build up

To train your body properly, you need a mixture of intensities in the workouts you do. There are many ways to measure this intensity. One way that is slightly more precise than others is that of heart rate[4] zoning. This is where you partition your heart rate into zones according to distance from max heart rate. In other words, **Z1** is easy effort and **Z3** is hard effort.

[4] effort is measured by how your body responds, so workouts are at your current fitness level

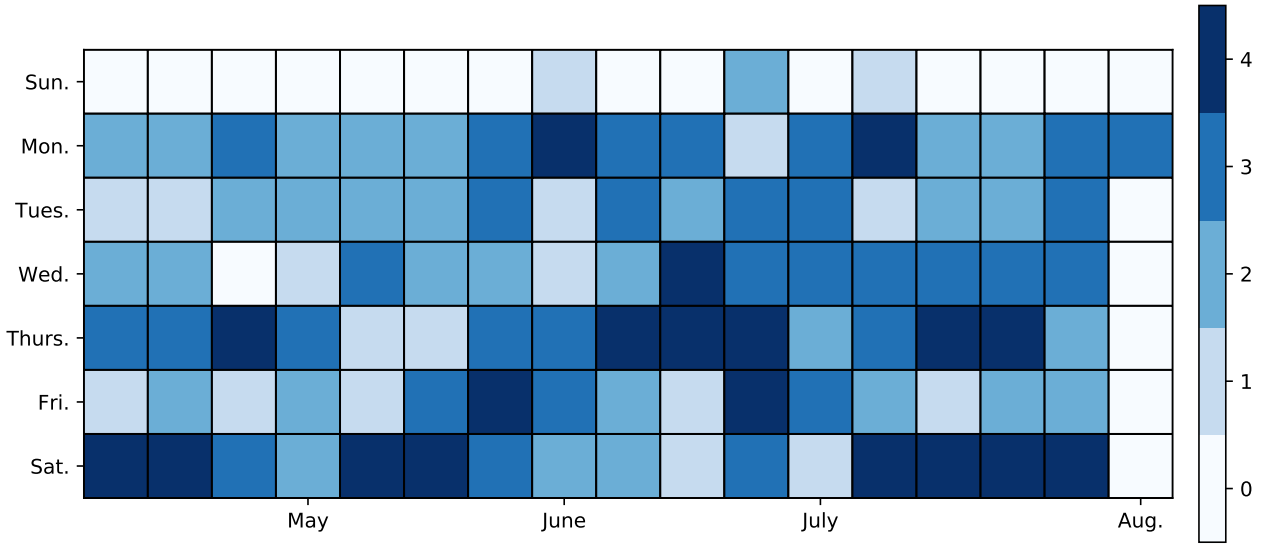
In Jared's case, he has a self constructed rating system, mostly likely accidental, that emerged from his training.

- No Workout (0)
- Chill (1)
- Chillish (2)
- Descent (3)
- Great Clip (4)

By looking at the recorded data, and then assigning a category for each intensity level, we can look at how hard Jared trained in the months leading up to the Olympic games.

There are lots of fun little tidbits here. My favorite is if you look top to bottom you can see his recovery days and higher intensity days align well with days of the week. Similarly, if you look from left to right you can see a higher overall intensity as the Olympics approached.

In running it is also important to taper, meaning you reduce the intensity of your exercises as a race grows closer. Jared's taper isn't



shown here. He continued running in Brazil for the entire month of August which is when he started to rest his body more. Over all, by using statistics of pacing, and keeping track of his workouts over a long period of time[5], Jared placed 6^{th} in the Olympic marathon.

[5] and with insane personal dedication

8.5 *Its your turn*

Throughout this book, we've talked about various interesting ways that people have used math, stats, and data science techniques in their daily lives. There are so many opportunities to use a little bit of data to have a large impact on your decision making.

Now that you've read this book, I hope you're inspired to start your own personal data projects. Measure things about your life and run some analysis.

I'd love to see the results, feel free to share them with me and the broader community with **#everydaydata**.