Everything You Should Already Know About Data Science.

Matt Dancho

Second Edition

Table Of Contents:

Introduction (The Way Of The Business Scientist)

It was August of 2017. I had been working a full-time job as the Director of Sales & Engineering at my previous company while simultaneously growing a small consulting firm called Business Science. I had just LLC-ed the company (a term for making Business Science a legal entity to reduce the liability if I screwed up a consulting project). Things were going really well.

My life had changed. I began learning R four years prior and my sacrifice of time and dedication had been paying dividends. In the two years prior, I had been promoted 3 times. I had gone from managing a team of 4 to a team of 60. My salary had doubled, and I was now making well over $150,000 per year (which in that time was enough money to support my family and have some fun too).

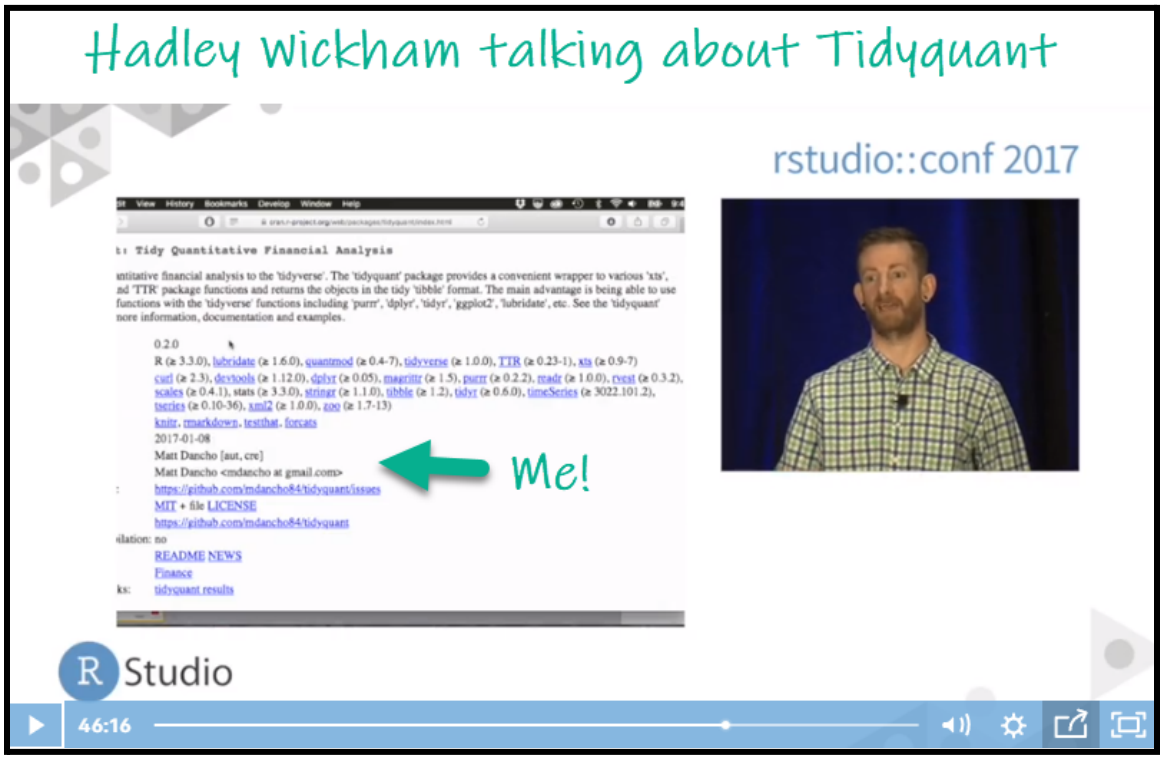
Enjoying time with my family! (My daughter caught her first fish)

But, what I was most proud of professionally was a little R package I had created the year before called `tidyquant`, which had grown to 50,000 users and had kickstarted my data science career. It was an R package that simplified financial analysis and, more importantly, made financial analysis in R accessible to the hundreds of thousands of new R users that were quickly adopting a new ecosystem of R packages called the `tidyverse`.

I had gotten the idea to build the R package while I was taking several courses on R through Coursera, the e-learning platform. The idea ate at me while I was taking these online data science courses. I hadn't been learning much through the courses, and I was worried that if I didn't jump on creating `tidyquant`, someone else would beat me to it.

In November of 2016, I sped through the last course in the Coursera program, and quickly began building `tidyquant`. I knew enough R to be dangerous, but I didn't know how to create an R package or all of the analysis that "Financial Professionals" were doing. But within 3 weeks, I figured out how to hack together enough code to build a small version of tidyquant that I was happy with. I launched `tidyquant` on December 31st, 2016. What happened next is nothing short of amazing.

My big break was the 2017 Rstudio Conference. Rstudio Conf happened the 3rd week of January, and during Hadley Wickham's Keynote Presentation, which was being viewed by 100,000+ data scientists worldwide, he mentioned my `tidyquant` package!!! That simple act created the momentum I needed to become a data scientist in my own mind. The domino-effect was in full force.



Did Hadley Wickham just bring up `tidyquant`?!!

The first domino to tumble my way was consulting. Small investment firms began reaching out asking for help in automating financial analysis. It was crazy - they wanted *my* help. I didn't have a financial degree. Heck, I was just a glorified Googler at that point with a minor in Stack Overflow.

Then the requests began getting bigger. Fortune 500 firms were reaching out to me. And by the end of 2017, I was working with some of the largest companies in the world to both train their people and build software solutions for them. All while working a 9-5 day job as Director of Sales & Engineering!

My Imposter Syndrome

What looked even more promising was when a recruiter from Facebook (now Meta) reached out to me asking if I'd be interested in working for them.

At that point, I wasn't sure where my life was going. I hadn't yet started creating my R Track Program. I just knew that data science was calling. And I simply felt that I needed to jump on this opportunity.

I updated my resume adding all of my accomplishments, from creating the R package `tidyquant` that had been downloaded 50,000 times at that point, to showing the software projects that I was building for companies, to the shiny web apps that I had been making to automate analysis for the investment firms, to even working with some of the highest profile companies (like the marketing firm MRM McCann and the financial firm S&P Global).

I responded to the recruiter, sending him my updated resume. And then I heard nothing…

A week later I messaged the recruiter. He was very professional, but the news felt like Mike Tyson punching me in the gut.

> *"Matt, I really want to thank you for applying and sending your resume. If it were up to me, I would hire you in a heartbeat. But, our Data Science Manager checked out your resume and told me he only hires PhD's in Computer Science… So, there's nothing I can do. Facebook isn't going to hire you."*

I felt like a failure. But, even worse, this rejection had caused me to question my entire future in the field I loved. It made me wonder if I'd ever be good enough to be a data scientist.

What I Learned From Getting Rejected

I sat back and thought about the situation. Facebook didn't want me. Yeah, that sucked (mind you, this was before I knew about what they *really* do with their data).

On paper, I probably shouldn't have been a data scientist. My resume was lacking a lot of "important data science stuff":

I had no degree in computer science.

I wasn't formally trained in statistics.

I didn't have a degree in advanced mathematics.

I didn't know Python.

I couldn't read mathematical notation.

I had struggled with theory.

And, I wasn't a very good coder.

But, here's what had happened.

I realized the truth. Every company *doesn't* need a PhD in computer science, statistics or mathematics. In fact, most companies need someone that's scrappy enough to hack their way to a solution that does better than whatever the current process is, which doesn't require as much coding or math skills as you'd think.

Furthermore, most companies didn't even have coding in-house. 98% of companies were stuck on Excel. So programming languages were super powers. If used, they were a massive upgrade to their current process.

And, most companies didn't care about theory either. Try talking to a Senior Vice President about models and algorithms and watch their eyes roll back into their head with boredom. Yes, I've tried and every time there is just *silence*. It's painful.

Facebook was overlooking THE most important aspect of data science.

Data science was never about algorithms, math, theory, or coding. Not even one bit. So what was data science about?

All of the companies I worked with wanted the *exact* same thing:

Business Value!!

ROI = Return on Investment, the financial measure of business value.

The most important part of data science is to provide business value. That's what companies *really* want, a return on their investment (ROI).

I had proof. I validated this "business value" argument time and time again with each and every successful consulting engagement and each and every promotion in my job. My 9-5 was rewarding me with promotion after promotion. Plus, I had been hired by multiple Fortune 500 companies that paid very well. And, they kept coming back for more because of the business value I was unlocking for them.

The day I realized what I could do for businesses with data science…

That was the day I became a Business Scientist.

Struggles You'll Face (And How A Community Can Help)

Becoming a Business Scientist alone is rife with problems. And I'm sure you've encountered one or more of these already.

First, there are a lot of "data science experts" out there that are going to tell you *"You're doing it wrong."*

You'll hear things like, *"You need to use Python. R can't do production."* Then you'll realize they started coding in python 3 months ago and they have zero experience using R. AND FYI, it's actually easier to do "production" (automating data science tasks) in R in my humble opinion.

Or you'll hear things like, *"You have to learn all of the theory and math before you can become a data scientist."* Sure math and theory are important. But, without context they are pointless. The Business Problem MUST come first.

Second, there is too much training on data science coming out of low-cost data schools. 95% of their training is a mindless regurgitation of the same tools and datasets (e.g. mtcars, titanic, boston housing, etc) without context to business problems. And, it's no wonder why students leave these Low-Cost Data Schools with zero understanding of how to actually connect data science to business.

Third, the problem magnifies as we take more and more courses. I thought that I wasn't becoming a data scientist fast because I needed to complete my Coursera program taking each course until it was done. I was wrong. Every progressive course confused me more, and led me further astray. When I finally completed the program, I was even more lost than when I started.

The courses weren't the solution. It was building something I cared about that truly made a difference. Developing `tidyquant` taught me more than 3 years of taking courses on data science and reading books on algorithms and theory.

Each of these problems cost me dearly. My confusion, my lack of confidence and my imposter syndrome are the reasons that it took me 5-years to finally call myself a "Business Scientist".

If you feel like this, then you're not alone.

But, you need to change. If you keep doing what you're doing, then you'll get the same result.

What you need to do is to commit to something bigger than you and me, a community of people striving to do one thing…

Become a Business Scientist.

And they are following the way of the Business Scientist.

The Way of the Business Scientist

The Business Scientist is a new breed of data scientist that is focused solely on **creating business value**. Nothing else matters. Not models. Not algorithms. Not tools. Only Business Value.

The Business Scientist **unlocks business value** by increasing the company's revenue, decreasing costs, and increasing profitability. The Business Scientist explores problems through data, uses proven processes and problem-solving frameworks, experiments using a scientific approach, brings together teams and stakeholders to gain buy-in through the process, and ties all results to the business problem by measuring changes in financial value (return on investment).

The Business Scientist **takes responsibility** for where he or she is at. The Business Scientist is **not a victim** of their environment. The Business scientist is in **full control** of their destiny because they know there is a path to unlocking business value. And when they stumble, there are others in the Business Science community that are willing to help and share in their successes.

The Business Scientist is willing to **put in the work** to achieve greatness. From that work, the Business Scientist **reaps the rewards.** The business scientist's career accelerates first by getting their dream job, then by earning their new promotion, and even making the job transition from where they are now to where they want to be. And, the Business Scientist **doesn't make excuses**.

If you believe in yourself. You can become a Business Scientist.

You are now on the path to becoming a

Business Scientist.

And I will be your expert guide.

Now that you know the way of the Business Scientist, and you are committed to moving forward as a Business Scientist to make your dreams become a reality, I want to help you understand the path that is in front of you.

How this book will help you.

I put this book together for you, future Business Scientist, to help you on your journey. Here's how it works.

This book contains 7 chapters that cover every phase of your journey.

Chapter 1: The 14 Data Science Skills reveals the 14 data science skills that were used to help a student of mine increase their salary by $50,000. This is the perfect starting point for someone learning data science from scratch. It will help guide you to the right choice of programming language, development tools, and help you develop a learning plan that covers the 14 most important data science skills.

Chapter 2: The Business Science Problem Framework uncovers the secret methodology I use to solve 90% of business problems successfully. This is a great place to go for both beginners and experts alike. You learn my methodical framework that will help you *unlock* business value from data science. All organizations need this. If you can do this, you will increase your value.

Chapter 3: The Career Path of a Data Scientist shows you the full career path of a data scientist. Once you learn the 14 skills and know how to unlock value for your organization with the BSPF Framework, the next step is to understand how to grow as a data scientist. This chapter is important for anyone seeking to advance their career with data science. You'll learn what separates an entry-level from lead data scientist, how to become promotable, and which of the 2 tracks to pick if you want to advance your career in a company.

Chapter 4: How to Become a Financial Data Scientist (or a data scientist in any domain) exposes the key mistakes you are making in your journey to become a data scientist. This is so important because we all make them but you don't realize it until it's too late. In fact, I've made every mistake that I cover here. But the good news is that I also show a full case-study of how you can quickly fix these mistakes and become a data scientist in finance (or any domain).

Chapter 5: Why I picked R over Python for Data Science uncovers the 6 secrets that Python users have no clue about (and why I would still pick R over python if I was learning data science from scratch). This is great for beginners that are struggling to decide, and you'll learn why many people struggle with Python. I lay it all out so you can make an informed decision.

Chapter 6: Anatomy of a Data Science Team (Case Study) reveals the secrets of a high-performance data science team. I had the opportunity to spend a week with an elite hedge fund in Toronto, Canada. I took notes documenting their team structure, the tools they used become the top 1% in their investment category. I detail everything I learned (and it's awesome).

Chapter 7: The Data Science Workflow gives you my 3-phase framework for solving ALL data science problems. In this final chapter, I send you off with a bang. I give you every piece of advice that I have to help you see what it takes to complete a data science project. I extend what you learned about teams in Chapter 6, and give you the secrets to success as a data scientist.

And I have a special surprise for you at the end of the book for my future Business Scientists.

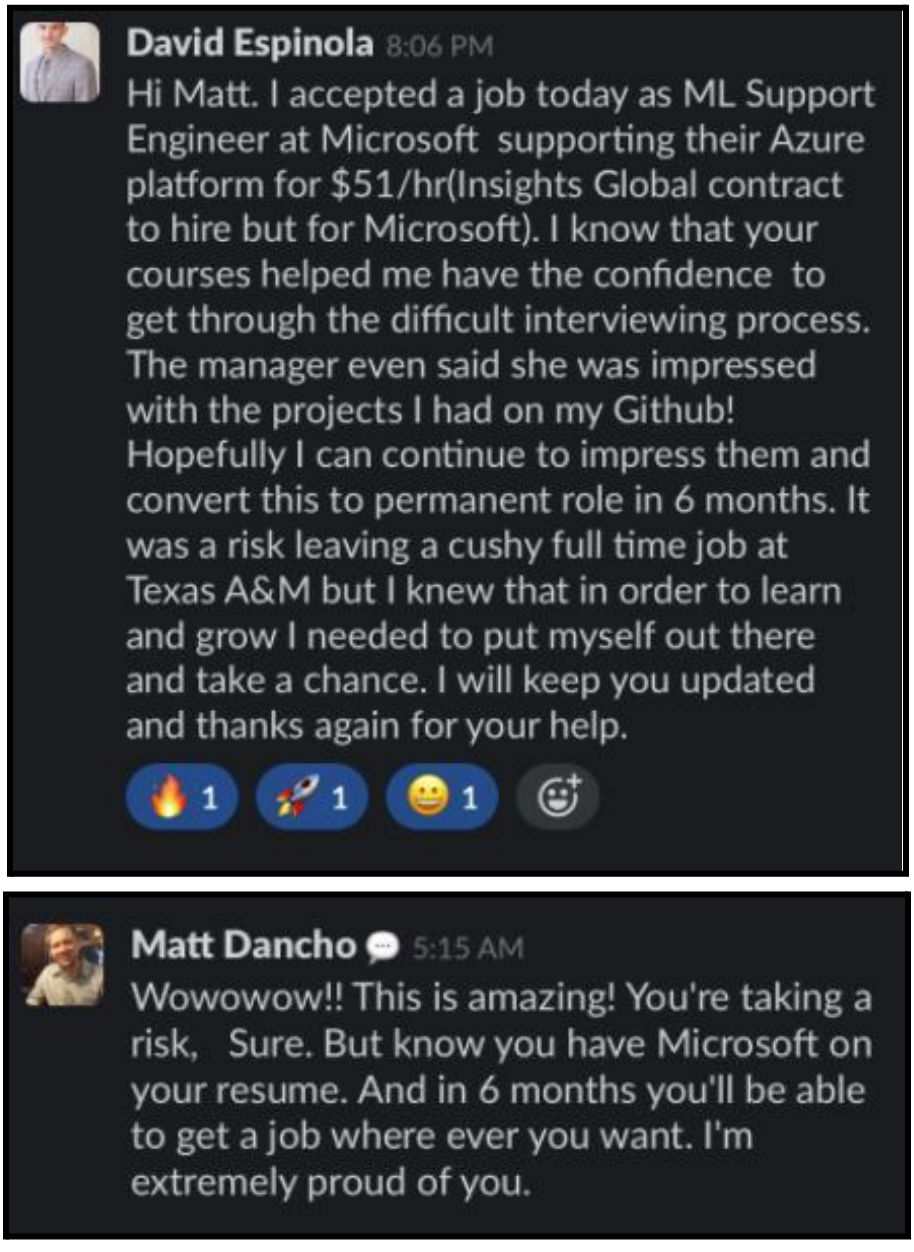
Without further ado, let's get going.

Matt Dancho

Chapter 1: The 14 Data Science Skills

That got David a $50,000 increase in salary

In late September 2021, David was a Research Analyst with Texas A&M University. In March of 2022, less than 6-months later, he accepted a position with Microsoft as a Machine Learning Support Engineer. In one of my webinars, David explained that he had just increased his salary by $50,000.



How was David able to land a job at Microsoft so quickly?

If you want to become a data scientist, you need to generate value for your organization. In general, you complete a process called the Data Science Workflow (more on this in Chapter 7), which involves learning these 14 data science skills.

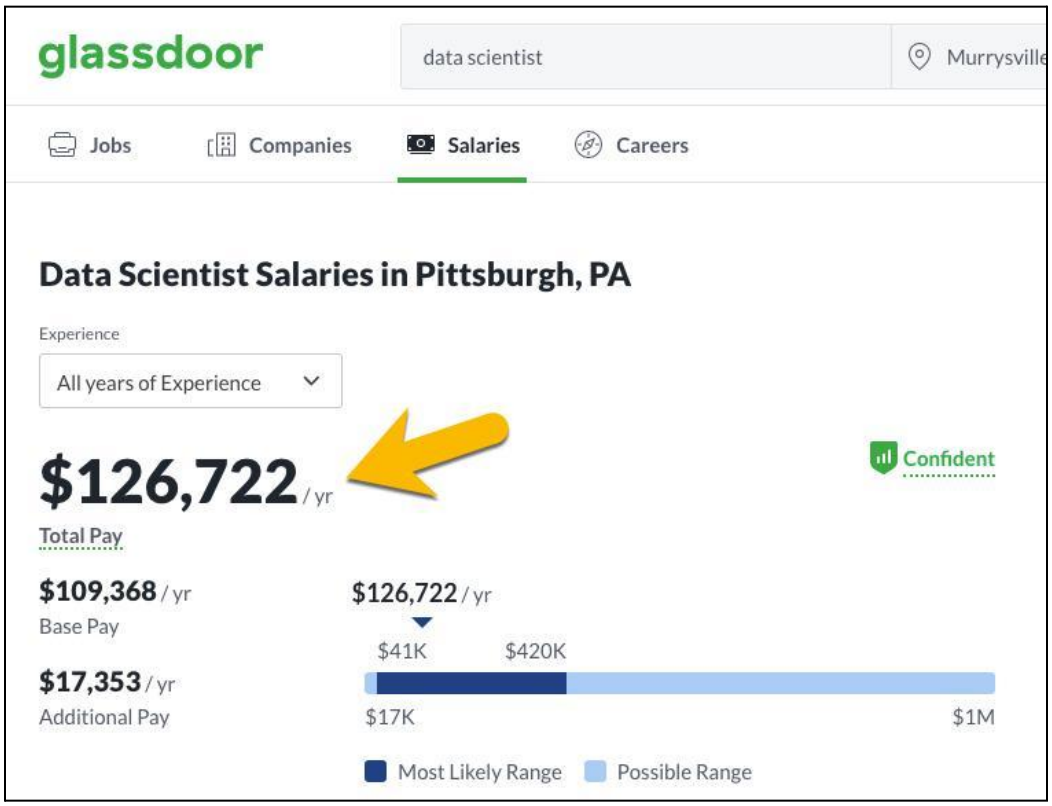
Plan	Skills
Machine Learning	Supervised Classification, Supervised Regression, Unsupervised Clustering, Dimensionality Reduction, Local Interpretable Model Explanation - H2O Automatic Machine Learning, parsnip (XGBoost, SVM, Random Forest, GLM), K-Means, UMAP, recipes, lime
Data Visualization	Interactive and Static Visualizations, ggplot2 and plotly
Data Wrangling & Cleaning	Working with outliers, missing data, reshaping data, aggregation, filtering, selecting, calculating, and many more critical operations, dplyr and tidyr packages
Data Preprocessing & Feature Engineering	Preparing data for machine learning, Engineering Features (dates, text, aggregates), Recipes package
Time Series	Working with date/datetime data, aggregating, transforming, visualizing time series, timetk package
Forecasting	ARIMA, Exponential Smoothing, Prophet, Machine Learning (XGBoost, Random Forest, GLMnet, etc), Deep Learning (GluonTS), Ensembles, Hyperparameter Tuning, Scaling to 1000s of forecasts, Modeltime package
Text	Working with text data, Stringr
NLP	Machine learning, Text Features
Functional Progamming	Making reusable functions, sourcing code
Iteration	Loops and Mapping, using Purrr package
Reporting	Rmarkdown, Interactive HTML, Static PDF
Applications	Building Shiny web applications, Flexdashboard, Bootstrap
Deployment	Cloud (AWS, Azure, GCP), Docker, Git
Databases	SQL (for data import), MongoDB (for apps)

The 14 Data Science Skills

David learned these 14 skills and was able to convince employers to hire him. The result was an instant $50,000 increase to his salary and now he is on a career path he is very excited about.

The Data Science Dream Career

According to Glassdoor, learning these data science skills can turn into a $126,722 career (if you live in Pittsburgh, PA, I encourage you to check out this information for where you live).

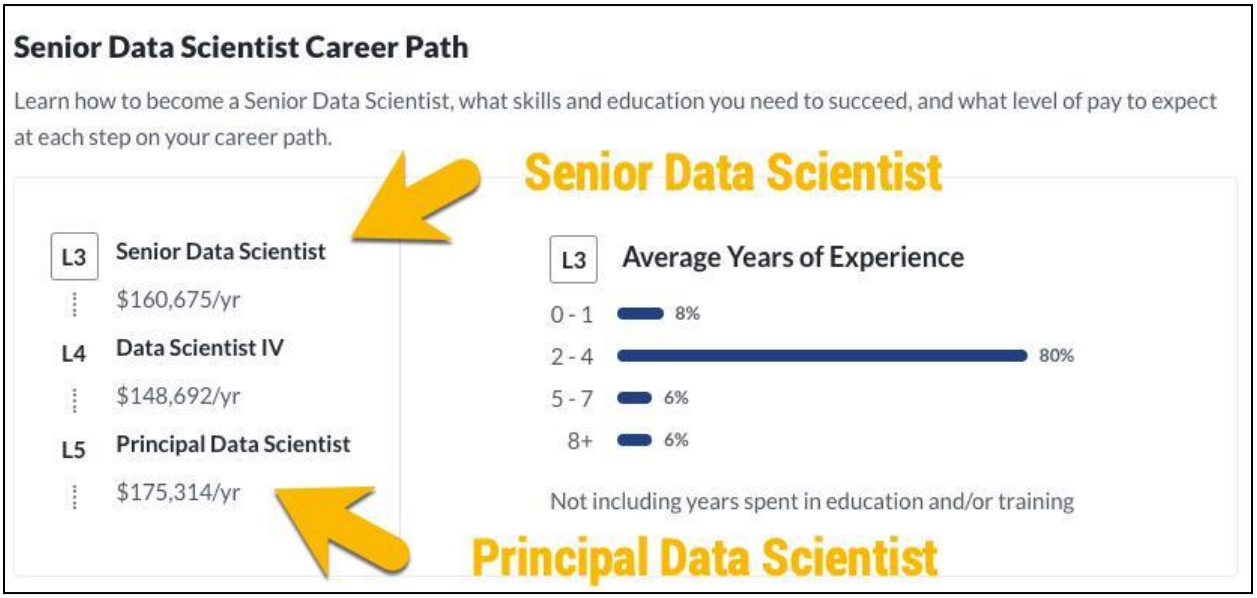


Glassdoor: Data Scientist Earnings $127,000 in Pittsburgh, PA (2022)

But that's just the start. Like I told David, your career will accelerate.

What's After 'Data Scientist'?

Here's what the career path looks like for a Senior Data Scientist in Pittsburgh, PA.



Glassdoor: Senior Data Scientist Earnings $160,000 in Pittsburgh, PA (2022)

You might be thinking: *"That salary is great. BUT, I'll never be able to master this list of skills. Especially not in 6-months."*

Actually you can.

Here's how.

How to Master Learning Data Science

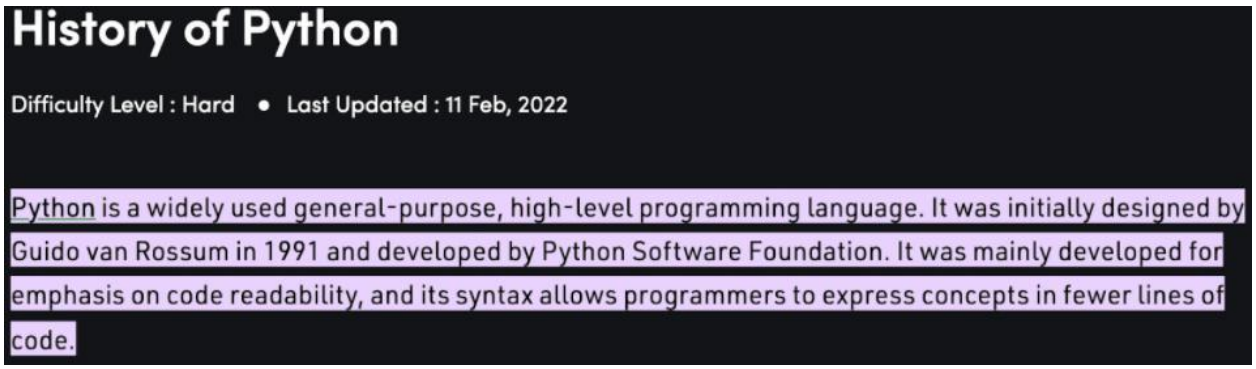
Mastering data science requires **motivation and planning.** You need to dedicate about 10-hours per week. And, you need to make a plan. Learning from people like David that have rapidly sped up their transition to data scientist can help you in your journey by speeding up the process for you too.

Step 1: Choose a Language

R or Python? Which is superior? In truth, it doesn't matter. You can succeed with either. If we want to answer this question, we should tackle this like data scientists, using data. Here are 3 things to consider: (1) how useful the language is for data science, (2) what the job demand is for the language, and (3) how much competition you will have for those job positions.

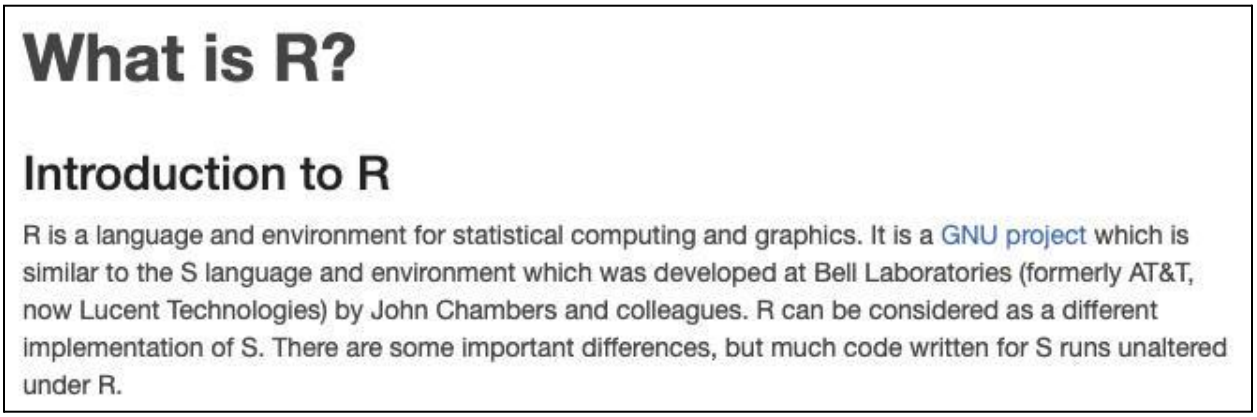
1. How useful is the language for data science?

If you look at the history of python, it clearly says it is a general purpose, high level programming language. It has an emphasis on code readability and to express concepts in fewer lines of code.



GeeksforGeeks: History of Python

Meanwhile R was closely modeled on the S language for statistical computing and graphics.



R-project: What is R?

Python is a general purpose language (but has been adapted for many tasks like data science) while R has been developed for the sole purpose of statistics. So I dug a little deeper, and here's what I found.

Strengths of the Data Science Languages

Python Strengths	R Strengths
Machine Learning	Statistics
Deep Learning	Econometrics
Apps	Statistical Modeling (& Machine Learning)
APIs	Reporting (& Communication)
	Web Apps
	APIs
	Integrates Python

Python is great for Machine Learning and Deep Learning but misses the mark on reporting (very important) and has fewer libraries for important analyses like econometrics.

R has excellent tools for business analysis and data science. R is strong in everything except deep learning. But, deep learning is rarely used in business problems. R is great for machine learning, which has much better performance for tabular data problems like those that are modeled in Excel spreadsheets. And when you need deep learning or extra APIs, you can integrate R with Python using a tool called `reticulate`.

Conclusion: R has well developed tools for business analysis and data science. Its only major weakness is deep learning, which is rarely used in business scenarios. And when you need deep learning or extra APIs, you can integrate R with Python.

I will give 1 point to R.

2. What is the job market demanding?

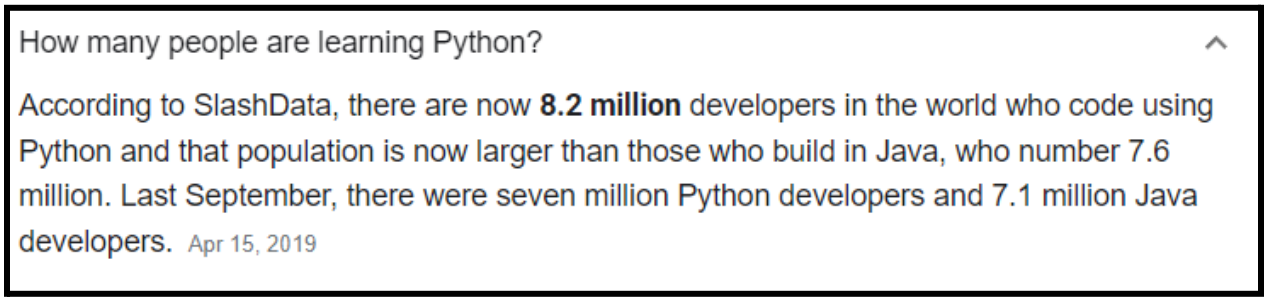
I did some research, and found that there were a total of 21,721 Python Data Science positions open, and 8,713 R Data Scientist positions open. So for every 1 R data science job there are 2.4 for Python.

Conclusion: The job market favors Python with 2.4 Python Data Scientist positions for every 1 R Data Scientist position. And keep in mind that most data science positions will accept either R or Python.

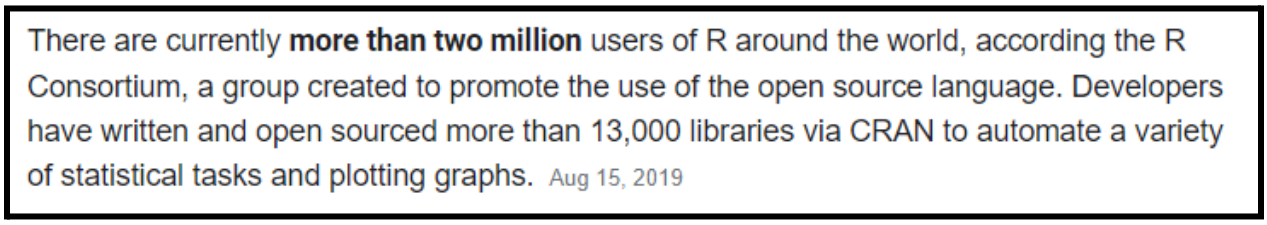
But, I will give 1 point to Python.

3. What is the job market competition?

Next, we need to consider how many people you will be competing against to get these jobs.



Source: SlashData, 2019



Source: Datanami, 2019

For every R user, there are anywhere from 4X more Python users.

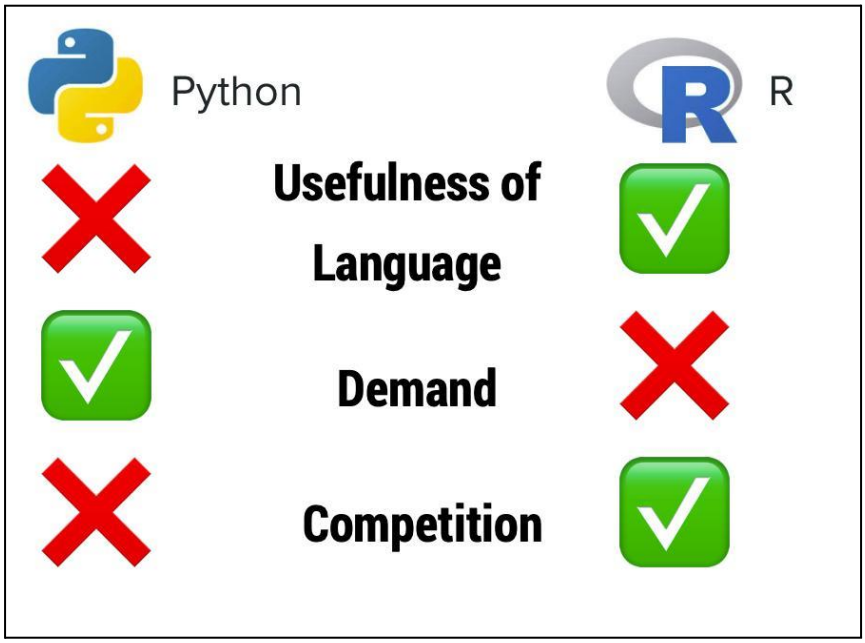
- There are over **8,000,000 people** that know Python.
- It's estimated that **2,000,000 people** know R.

Conclusion: R positions are **less competitive by 4X or more**. R makes you unique, and it's one of the reasons that students like David are able to quickly transition into a data science role. Keep in mind, you can always study Python in the future. But, you can stand out in lower competition roles with R (that pay a 6-figure data scientist salary).

The last point goes to R.

R versus Python Summary Matrix

The evaluation shows that R favors Python. R has more business-ready capabilities including easier reporting and more libraries for statistical analysis. Python has more job openings but python roles also have 4X the amount of competition. So if I were learning data science from scratch again, I'd still pick R. And this is one of the reasons students like David are able to quickly get data science positions by leveraging a more unique too, that has 4X less competition, and results in 6-figure data science careers. Your goal isn't to compete. It's to win a job. R is simply more useful and has less competitive opportunities.



What about Excel?

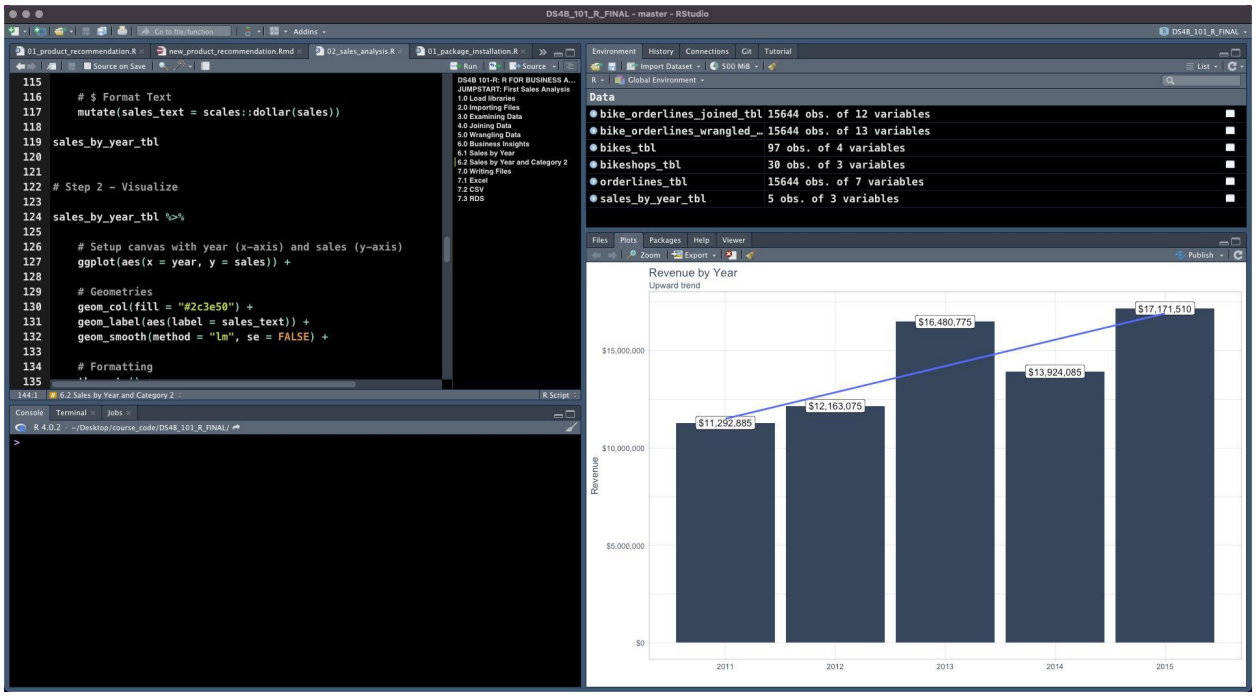
At this point I usually get the question, *"What about Excel?"*

Excel's limitations. Excel is great as a communication tool and is widely used throughout different business segments but it has limitations. First, Excel lacks machine learning which is essential for modeling business problems and getting business insights through explanations of those models. Second, Excel has a large data limitation. Excel has a maximum data size of 1-million rows, which is not very useful. Third, Excel is difficult to debug and errors go unnoticed. Functions are buried in cells, which can make it impossible to check your work.

Finally, this is the **Blue Screen of Death**, and I used to get this constantly when doing big data analysis in Excel. This is what happens when you exceed Excel's 1-million row limitation. So please use Excel wisely.

Step 2: Choose an Integrated Development Environment (IDE)

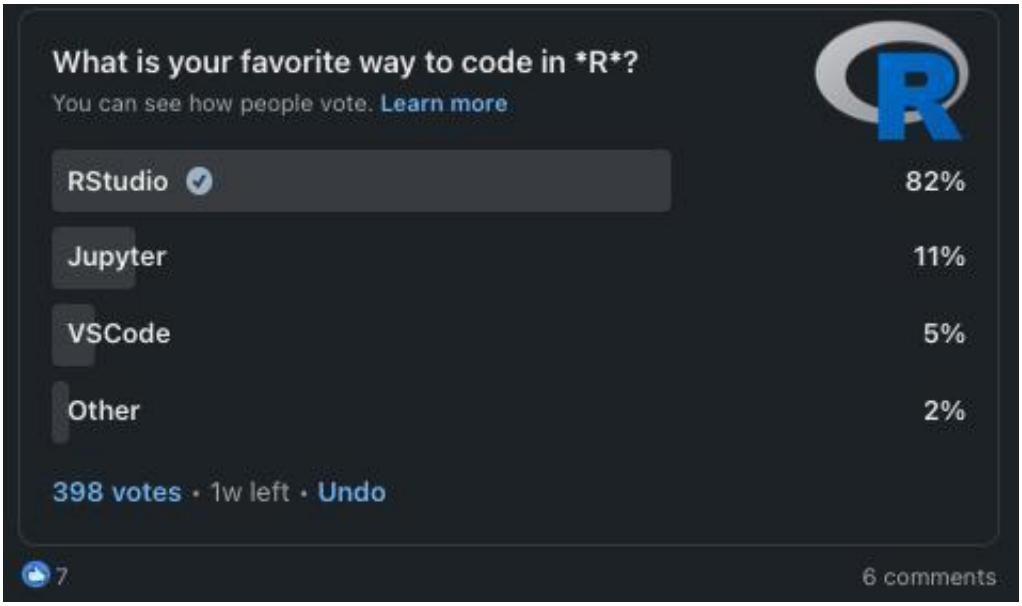
Choose an IDE or development tool, which is a fancy term for where you will type your code.



The RStudio IDE

The choice of IDE helps the community stay consistent. Lack of consistency creates problems when you try to communicate. If someone uses VSCode and has a problem. Is it Python or the IDE? If it's the IDE, then the problem may not relate to answers provided using Jupyter or RStudio.

I performed an R and Python IDE Survey of my 70,000+ LinkedIn users to see which IDE R and Python users choose most frequently.



R Poll Results

Python Poll Results

The choice for R is obvious. There is a strong preference for using **RStudio** when coding in R with **over 80%** of R-users surveyed selecting the same IDE. This often overlooked strength is a real benefit when asking questions and debugging code. If almost all users use the same IDE, the challenges you'll face are much easier as others have most likely faced and conquered them.

Python is far less straightforward. About half enjoy coding in Jupyter, a third like VSCode, and some are even using RStudio to code in Python. This makes picking an IDE just one more battle before you even get started learning Python. Keep in mind, this survey is based on my 70,000+ followers on LinkedIn who are likely interested in R programming in addition to Python. But the results are split between multiple IDEs. In fact, I got a ton of comments for Spyder and about half a dozen other IDEs that I couldn't fit into the poll.

My point is that picking Rstudio for R is a no-brainer. And, python is kind of a crap shoot. And I view simplicity as a strength.

Now once you have an IDE selected, it's time to learn the 14 data science skills.

Step 3: Learn the 14 Data Science Skill Groups

Once you settle on a language and IDE, you are ready to begin the fun process of learning the skills to become a data scientist and you need a plan. Your goal is to get a data science job **as fast as possible.**

Plan	Skills
Machine Learning	Supervised Classification, Supervised Regression, Unsupervised Clustering, Dimensionality Reduction, Local Interpretable Model Explanation - H2O Automatic Machine Learning, parsnip (XGBoost, SVM, Random Forest, GLM), K-Means, UMAP, recipes, lime
Data Visualization	Interactive and Static Visualizations, ggplot2 and plotly
Data Wrangling & Cleaning	Working with outliers, missing data, reshaping data, aggregation, filtering, selecting, calculating, and many more critical operations, dplyr and tidyr packages
Data Preprocessing & Feature Engineering	Preparing data for machine learning, Engineering Features (dates, text, aggregates), Recipes package
Time Series	Working with date/datetime data, aggregating, transforming, visualizing time series, timetk package
Forecasting	ARIMA, Exponential Smoothing, Prophet, Machine Learning (XGBoost, Random Forest, GLMnet, etc), Deep Learning (GluonTS), Ensembles, Hyperparameter Tuning, Scaling to 1000s of forecasts, Modeltime package
Text	Working with text data, Stringr
NLP	Machine learning, Text Features
Functional Progamming	Making reusable functions, sourcing code
Iteration	Loops and Mapping, using Purrr package
Reporting	Rmarkdown, Interactive HTML, Static PDF
Applications	Building Shiny web applications, Flexdashboard, Bootstrap
Deployment	Cloud (AWS, Azure, GCP), Docker, Git
Databases	SQL (for data import), MongoDB (for apps)

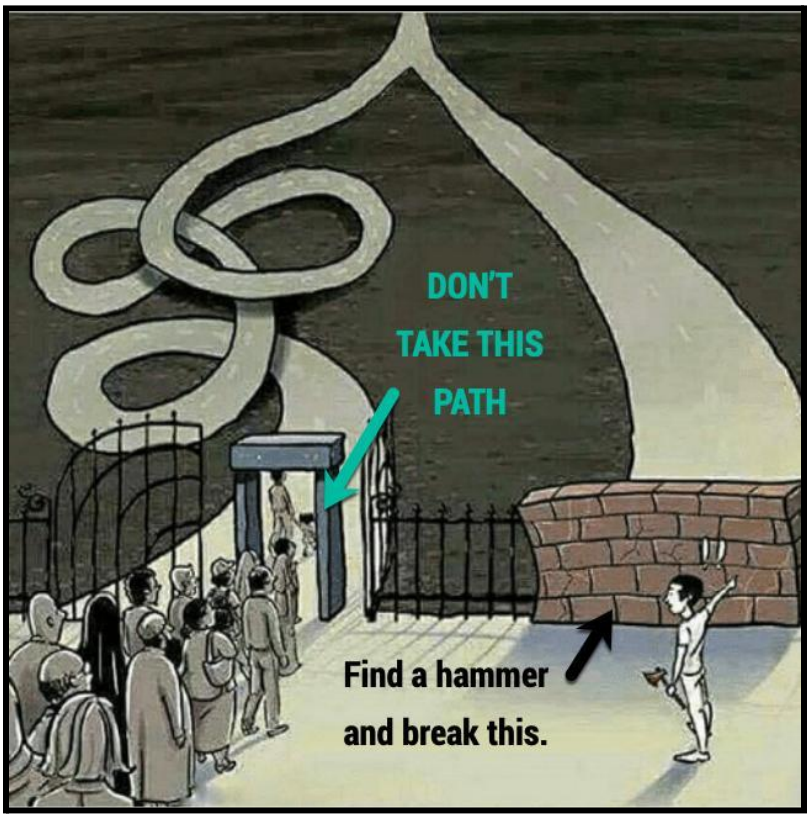
The 14 Most Important Data Science Skills (And Sub-Skills) for Your Learning Plan

Soft Skills: What about soft skills?

At this point you might be thinking, *"Matt, everything you've shown are technical skills. What about communication skills?"*

Soft skills like communication are important too. Focus on these 3 things. First, learn to create stimulating visual presentations using a slide deck. Data storytelling is very important to building persuasive arguments. Second, learn how to make concise reports. Use executive summaries to show the result and the body of the report to show how you got there. Third, be kind. People like to work with enthusiastic and friendly people. Learn to be pleasant and professional during discussions. If you do those 3 things consistently, people will want to hire you, work with you, and promote you. Now, let's switch back to the technical skills.

Technical Skills: The 3 Learning Paths (choose wisely)



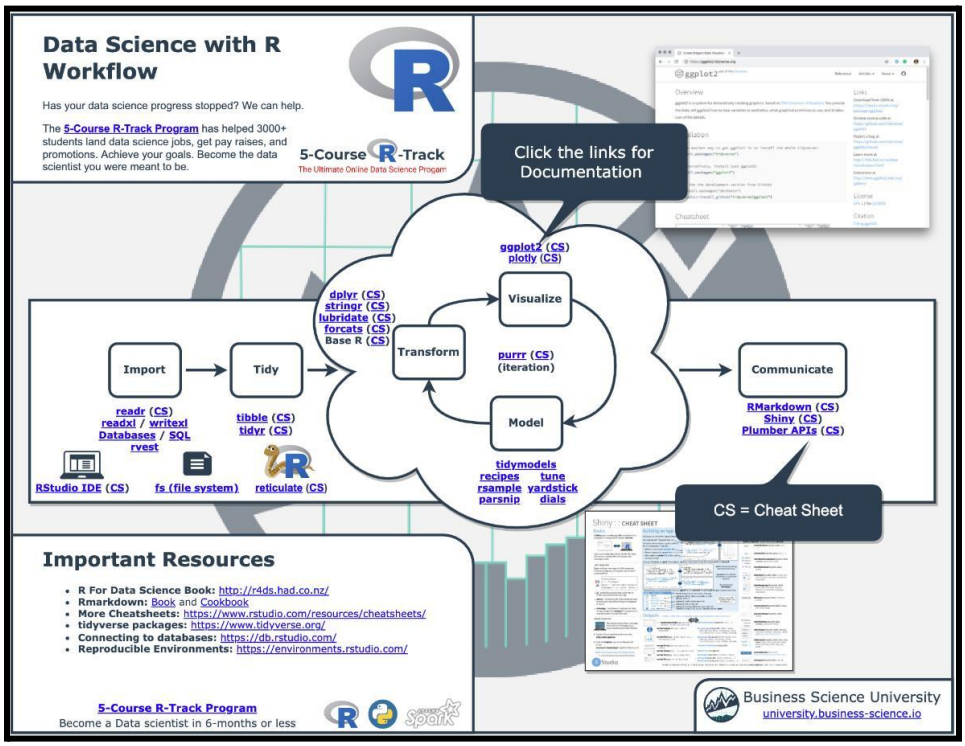
The learning paths that aspiring data scientists choose fall into 3 groups.

The first group has **no plan**. These are hobbyists. They usually quit. This costs them $8,000,000 over a 35 year career when factoring in a measly 3% annual raise. Now keep in mind, I had a bad plan and it took me over 5 years to learn and I missed out financially.

The second group has **a bad plan.** They will take 5+ years to eventually learn data science. They will also lose out financially. 5 years at $125,000 per year when factoring in a 3% raise = loss of $664,000. Ouch!

The third group has **an exceptional plan.** They are likely to be successful and can complete the transition in under 6-months. Students like David have an exceptional plan. They made it in 6-months. And, it involves cheating.

In the real world, to learn data science fast you need to learn skills and hack your way to solutions through experimentation, finding out what works and what doesn't along the way. One massive booster in this process is my R-Cheat Sheet that will help you learn the skills you need.

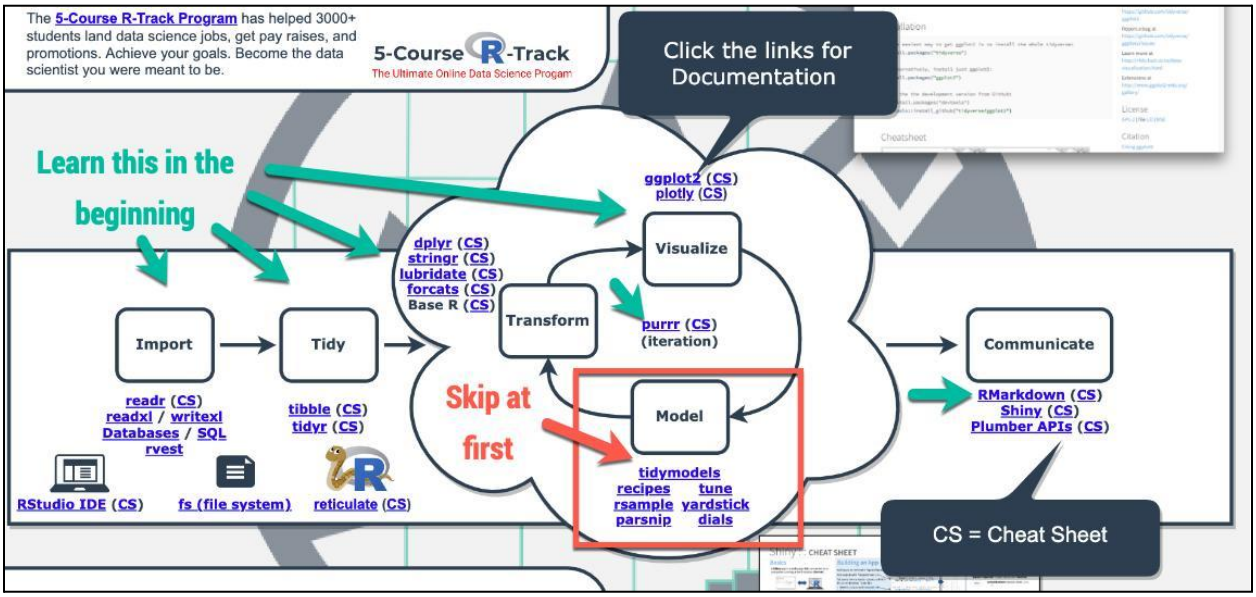


Download the Ultimate R Cheat Sheet

Here's how to use the R-Cheat Sheet to speed up learning the 14 Data Science Skills.

Step 1. Learn The Foundational Data Science Skills First

I know you are going to jump right into Machine Learning first. It's A BIG MISTAKE. Start with your foundational skills. But, which foundational skills should you learn?



We'll follow the first page of the R Cheat Sheet to learn the 80/20 Skills.

What are 80/20 skills?

These are the 20% of skills that you will use 80% of the time to solve business problems. If you focus on these core foundational skills, it will make machine learning so much easier.

Follow this table for the 80/20 skills when making your learning plan.

Task	What to learn
Importing data	Working with databases, connecting to `SQL`, `readr`, `readxl`.
Transforming data	Working with outliers, missing data, reshaping data, aggregation, filtering, selecting, calculating, and many more critical operations, `dplyr` and `tidyr` packages
Visualizing Data	Communicating through Interactive and Static Visualizations, `ggplot2` and `plotly`
Time Series	Working with date/datetime data, aggregating, transforming, visualizing time series, `timetk` package
Text	Working with text data, `stringr`
Categorical data	Working with categories, `forcats` package.
Functional Programming	Making reusable functions, sourcing code, iteration, `purrr` package
Reporting	Making reports in interactive HTML and static PDF formats, `rmarkdown` package

Step 2. Learn Machine Learning

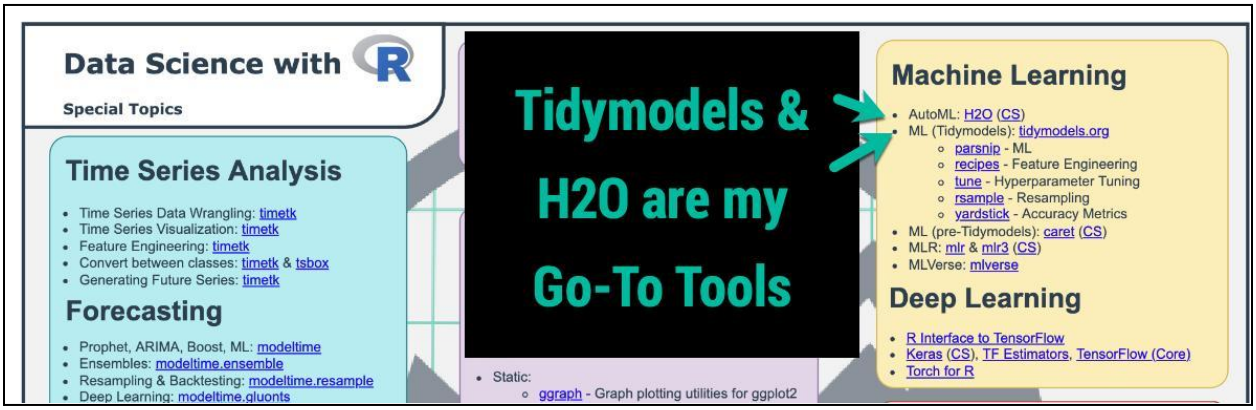
Before I jump into Machine Learning, students usually have this question: *"What about math, stats, and algorithms?"*

The Popular Opinion: Take 5-years and study theory, math, and learn how to code algorithms from scratch. I don't recommend this path. It's inefficient, and it has a heavy cost. You lose out on data science jobs that companies need right now.

Matt's Way: Learn math and stats while you apply machine learning in projects. This makes math and stats much more relevant (and you will actually remember statistical concepts because you now have experience applying them). This has been my tried and true method for speeding up my learning. And, I highly recommend it for you.

Which Machine Learning Tools Should I Learn?

On the cheat sheet, you'll find links to my go-to Machine Learning tools. Add the following tools to your learning plan: `tidymodels`, `H2O`, and `recipes`.



I'm a big fan of two machine learning ecosystems. First, `tidymodels`, which I use for making ad hoc models and then explaining them to the business to drive actions. Second `H2O`, which I use for automated machine learning (AutoML) to get high accuracy models that I use in production.

Another extremely important skill is feature engineering. The tool I use most commonly is called `recipes`**.** Recipes is a set of preprocessing tools that becomes critical in preparing data for machine learning. Garbage in = garbage out.

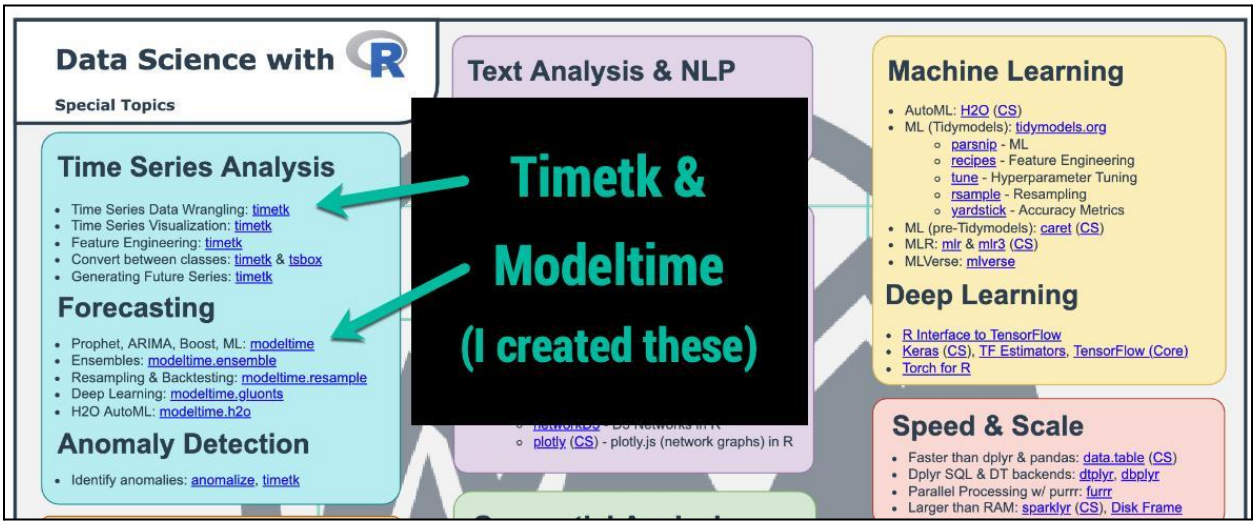
Step 3. Learn Time Series

Companies are looking for employees that can make forecast improvements.Why?

A 5% improvement in a forecast can save a company like **Walmart $50,000,000 each year**. So, if you can predict the future, chances are you are going to be very valuable to your company and your future companies.

Which Time Series Tools Should I Learn?

Back to the cheat sheet, we're going to focus on the *Time Series Analysis* and *Forecasting* sections. Add the following tools to your learning plan.



Here's what you need to learn:

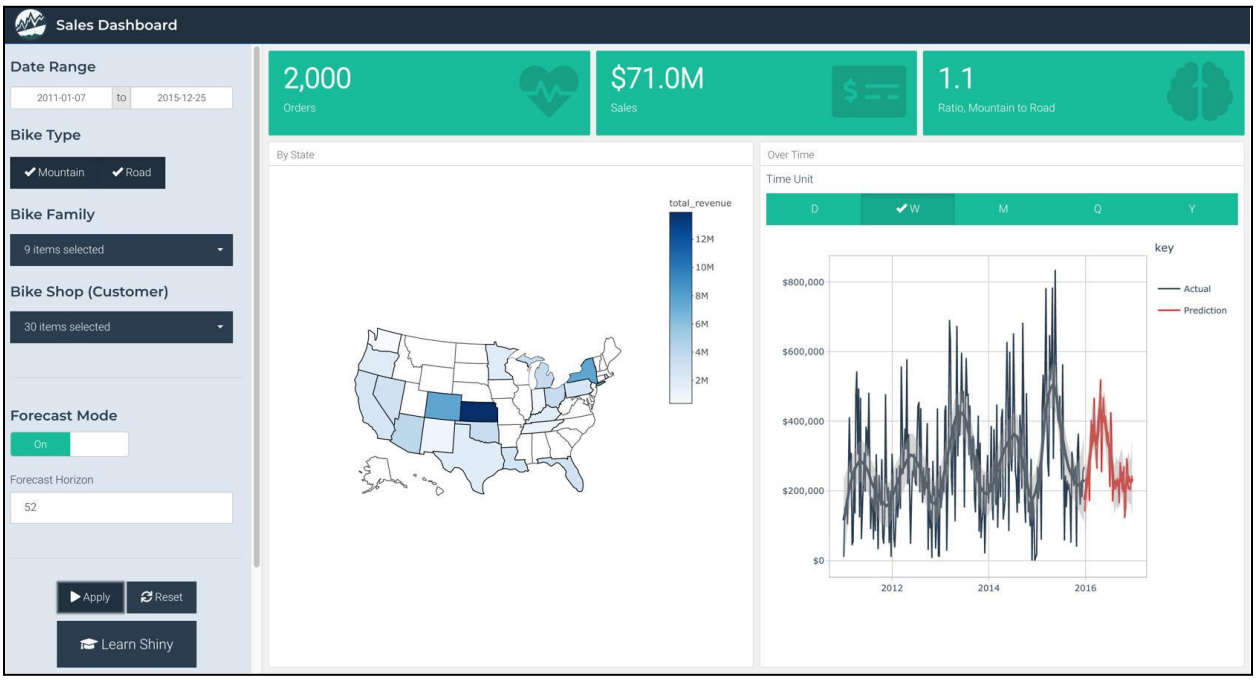
- **Time Series Analysis:** Working with date/datetime data, aggregating, transforming, visualizing time series, `timetk` package.
- **Forecasting:** ARIMA, Exponential Smoothing, Prophet, Machine Learning (XGBoost, Random Forest, GLMnet, etc), Deep Learning (GluonTS), Ensembles, Hyperparameter Tuning, Scaling to 1000s of forecasts, `modeltime` package.

Once you have those skills in the bank, then it's time to move onto production.

Step 4. Learn Production

Your model is useless until someone can use it to do something productive. Maybe your model triggers a customer service rep to call a customer that is on the edge of unsubscribing. Or maybe your model has generated more accurate forecast information before placing an $1,000,000 order for parts that could be unnecessary. Either scenario requires someone to be able to take action from your decisions.

Here is where you provide business value. You help the decision-makers by putting applications into production that improve their decision-making with data. My favorite way is through a `shiny` app.



A Shiny Application

One of the truly amazing things is the ability to integrate predictive machine learning models into applications. We can use applications to automate the analysis process and users simply click buttons, use drop-downs, and get information, all without ever knowing that R (or Python) is running code behind the scenes.

The particular application shown above was made with a tool called `shiny`. Shiny is a massively overlooked tool that I used heavily in my peak consulting days. When I showed them the shiny apps that I had built for previous projects, they instantly knew that I could help them.

Add `shiny` to your learning plan.

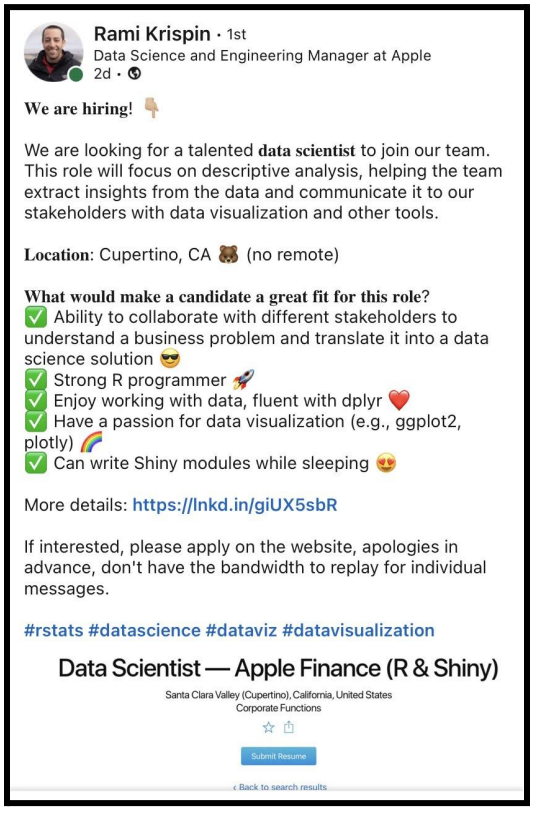
What's the job market like for R & Shiny?

The most common concern I hear for data scientists that use R and Shiny application developers is, *"Will I be able to get a job?"* I'm happy to say that my answer is a resounding yes! But, don't simply trust me. Let me show you with figures to support my emphatic YES!

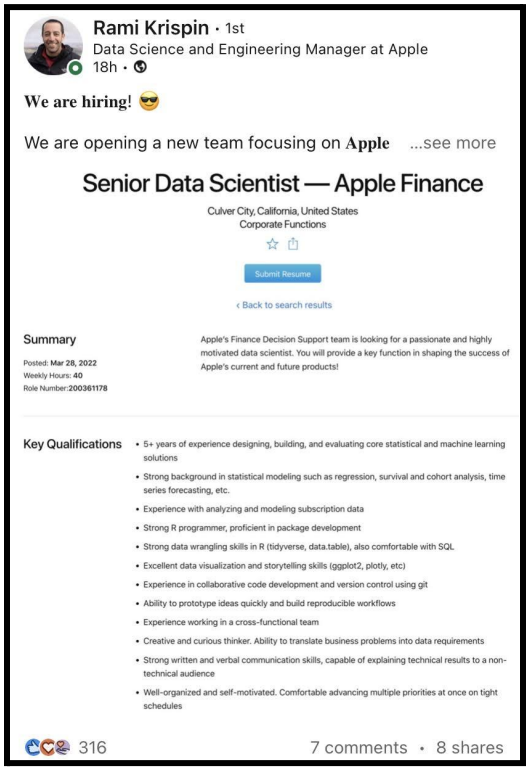
Will Fortune 500 companies hire you?

The first thing I want to say is that picking R is an amazing choice. So amazing that Fortune 500 companies like Apple are hiring data scientists with R and Shiny experience.

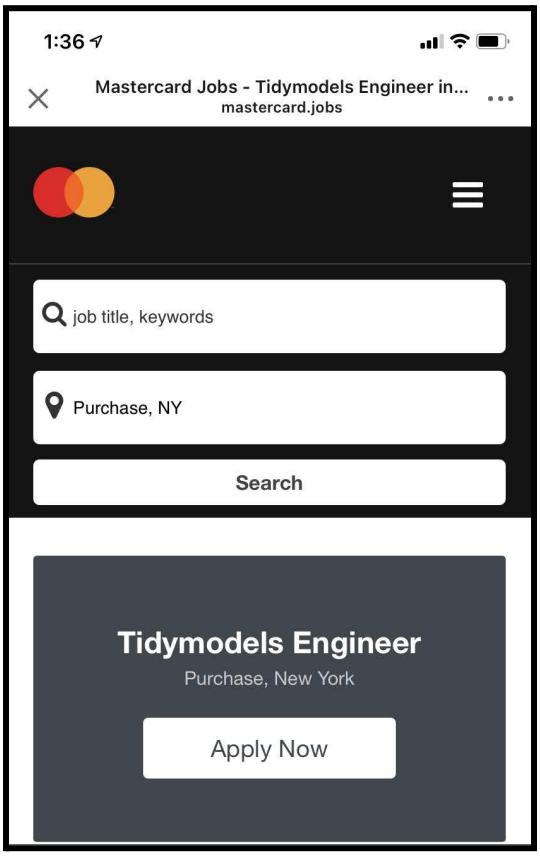
Apple (R & Shiny)



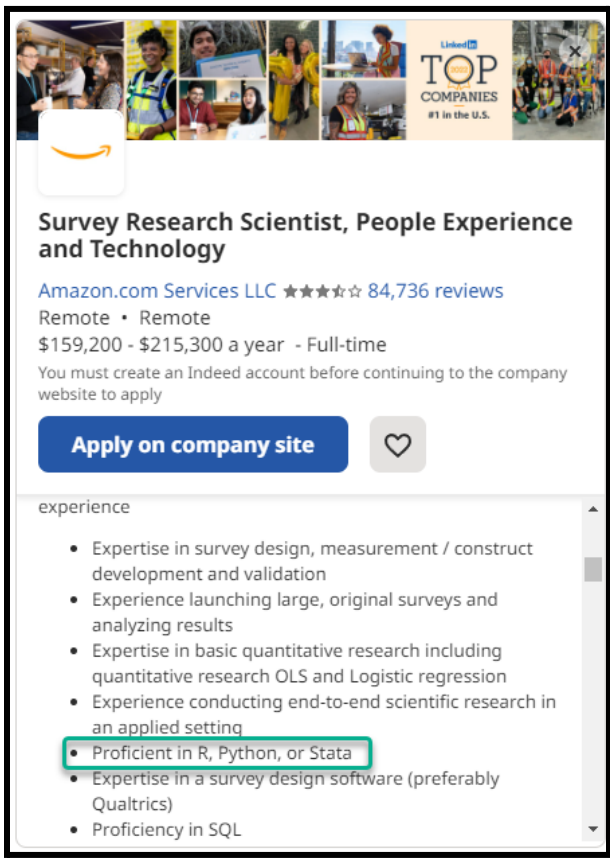
Apple (Senior Data Scientist)



MasterCard (Tidymodels Engineer)

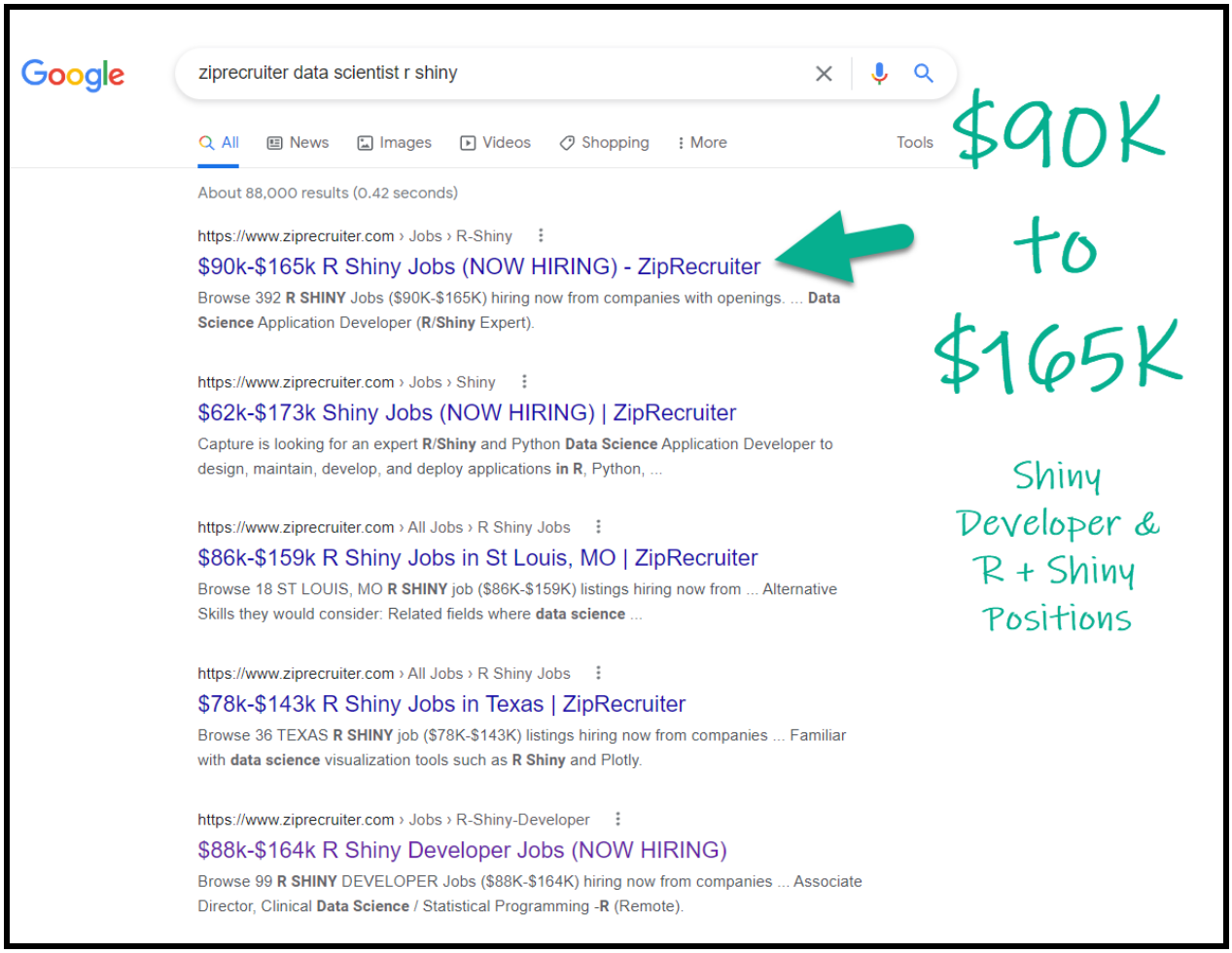


Amazon (Research Scientist)



What do data scientists with R & Shiny experience make?

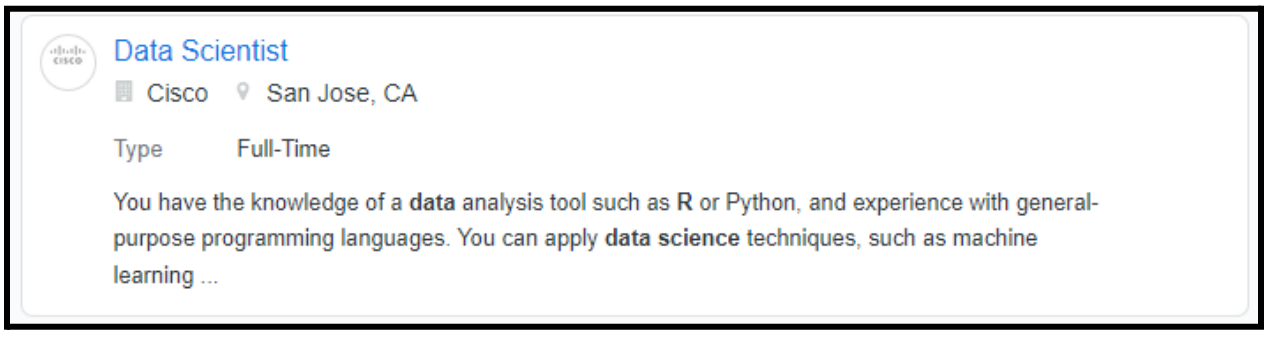
A quick search on ZipRecruiter shows a salary range between $90,000 and $165,000 currently for roles in the US. This places the salary range inline with “Data Scientist”. Further, there’s actually less competition for these roles since there are about 4X more Python programmers than R programmers.



"R Shiny Developers" & "R Data Scientists" make as much as (or more than) "Data Scientists"

Do tools even matter?

I've worked with dozens of companies. Every company is different, but what I will say is the ones that were successful have cared less about tools and more about results. In fact, many positions are open to either R or Python (tools matter not). What companies care about is **business value.**



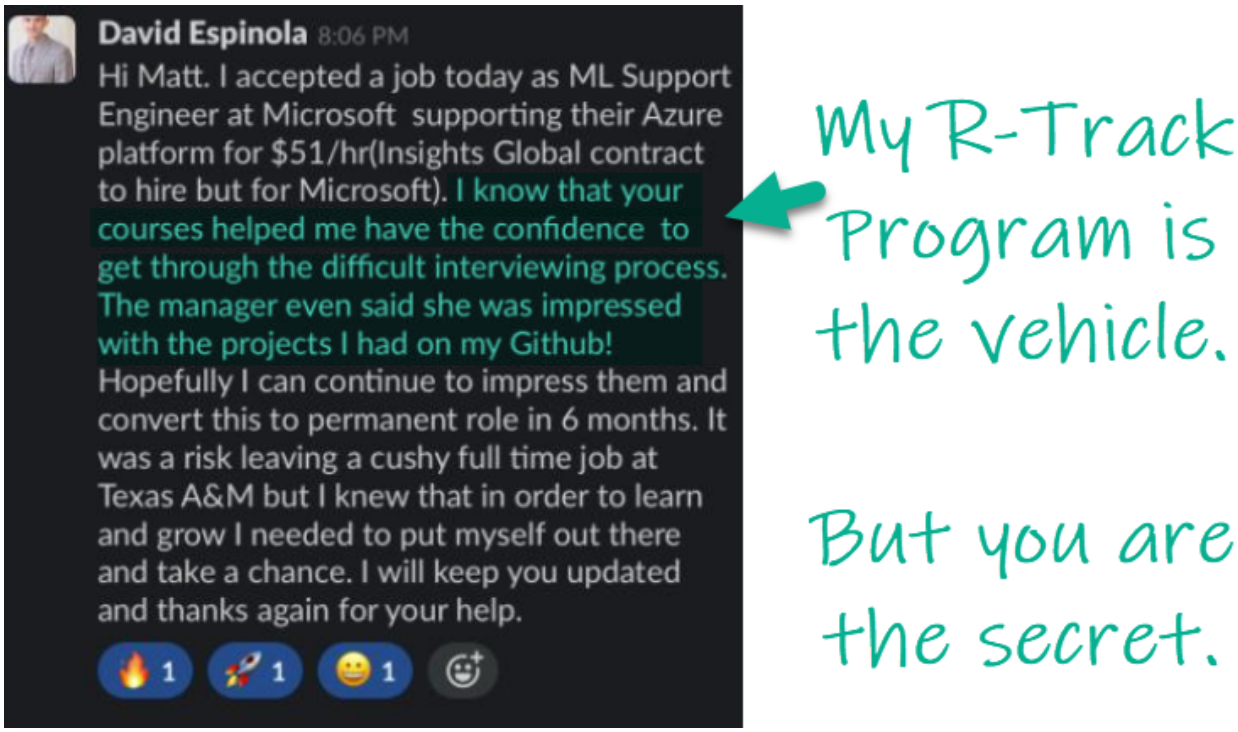
So we now know that high-quality jobs exist. But how did David learn all of this in under 6-months and get his $50,000 pay raise?

How David Got His $50,000 Pay Raise

At this point you should have a high-level understanding of which skills that David learned to increase his salary by $50,000 and land a career with Microsoft as a Machine Learning Support Engineer.

But you still don't have a plan to do it fast. This is what I call the **"Exceptional Plan"**.

David was able to achieve a transition in 6-months by following the way of the Business Scientist.



David did his research, and found me. He committed to my teaching and the way of the Business Scientist. He joined my R-Track Program. He chose wisely, took control of his destiny, and made his new data science career happen. The result was a 6-figure data science career.

...

Next, I want to introduce you to the most powerful tool in my arsenal. A special framework that can solve 90% of business problems. This framework has made me hundreds of thousands of dollars in consulting projects and has saved companies millions of dollars. And the best part is it doesn't matter whether you are a data scientist for a company, a full time business consultant, or an aspiring data scientist that is interviewing for a data science job, this framework will help you. Let's see how.

Chapter 2: The Business Science Problem Framework

How To Solve 90% of Business Problems with Data Science

This is a story of how I learned how to connect data science and business. It's embarrassing to say, but I learned more from one big failure than I did from dozens of successful projects.

My consulting failure taught me everything I needed to know about business science

It was February of 2017, and I had just founded Business Science (my company). At that point I was taking on consulting jobs in addition to working my 9-5. I had recently created Business Science as a consulting firm. We'd help financial companies create Shiny web applications using financial analysis and time series with my `tidyquant` R package that I was uniquely qualified to handle. Word began to spread, and the jobs and clients began getting bigger and bigger.

In September 2017, I received a big opportunity when a Fortune 500 company signed up with Business Science to do a quick-turn Human Resources (HR) analytics assignment.

The client provided data on Friday from their HR database, and I was to present findings on Monday so the executives could add predictive insights into a CEO presentation that was to be presented on Wednesday. I hired an HR Specialist to help me analyze their data, extract insights, and create a detailed report that documented our entire predictive analysis and findings. We created an algorithm that detected 13 or so employees that should be targeted for the "executive track" but weren't.

After working all through the weekend, on Monday I presented the predictive model that suggested 13 employees should be reviewed for executive track. And then I heard… Silence.

What you never want to hear after delivering a presentation is "silence." In fact, the silence was deafening.

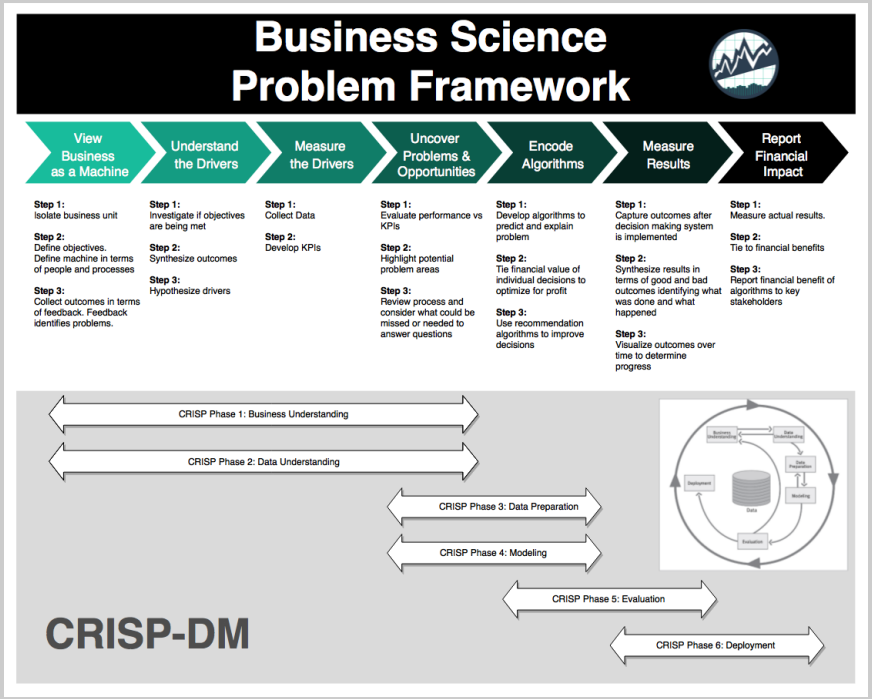
After the presentation, I got a call from my point of contact at the Fortune 500 firm. She filled me in on the disappointing news. She said, *"What we delivered was overly technical, not in a format they could easily add into their presentation. Her boss was dissatisfied."* I felt horrible.

I took the next 2-weeks off from any consulting engagements. I poured over every step of what happened. And then it hit me.

I had no plan. I needed a framework that would be the guiding light for what a successful data science project needed. The Business Science Problem Framework was born.

The Business Science Problem Framework

The Business Science Problem Framework is **my 7-step framework that can solve 90% of data science projects successfully while cutting your project times in half.**



Download the Business Science Problem Framework

A successful data science project doesn't happen by accident. It takes **communication** to effectively pitch benefits to executives, showing the results that relate to organizational goals. It takes **business understanding**, which only happens through interaction with the business stakeholders that are closest to the process or problem. And it most certainly takes **planning** to align everyone involved with the project scope and plan.

After pooling together many resources including the widely used CRISP-DM (Cross-Industry Standard Process for Data Mining) and combining with business management strategy that I

was learning from Ray Dalio's book, *"Principles"*, I developed a new strategy for managing data science problems.

The Business Science Problem Framework (BSPF) helps to manage a data science project successfully for 5 key reasons.

First, the BSPF provides a **clear plan** to discuss with executives and align their interests with a return-on-investment (ROI) oriented framework. This is what business leadership needs to see to trust you with their resources.

Second, the BSPF exposes the **key steps** in a data science project in a way that executives understand. This is critical for stakeholder buy-in.

Third, the BSPF shows that it takes **several weeks**, if not months, to complete a data science project. Having a realistic time schedule is critical to project success.

Fourth, the BSPF brings up **stakeholder questions** during the early discussion. This helps you build their questions into the project scope.

And, fifth, the BSPF gives the executive management team a **sense of confidence**.

Results from Others Using the BSPF

Before I discuss how the BSPF works, it's essential to know that since creating and deploying the framework successfully, my students have had success that exceeds my wildest expectations. **Students have been using my BSPF framework to ace the data science technical interview.**

I'd like to first introduce you to Jennifer.

"Thanks to Matt and what I've learned so far, I was able do do an in-depth analysis of Consumer Financial Protection Bureau (CFPB) data, following his Business Science Problem Framework and complete the project using RMarkdown. The polished, finished product impressed the hiring manager so much, he was willing to fast-track an offer."

-Jennifer Cooper, VP Strategic Analytics

Jennifer used the BSPF in her take-home interview (see Jennifer's testimonial). She followed the steps that she had learned in the 2nd part of my R-Track Program. JP Morgan Chase was so impressed with her take-home exam, they fast tracked an offer. Jennifer accepted the offer, landing her dream job as a Senior VP of Analytics at JP Morgan Chase.

Second is Masatake, who used the BSPF framework as part of his onsite live interview (see Masatake's testimonial).

Masatake says,

> *"In a job interview, I was able to draw the interviewer's attention because of my experiences to formulate some insight from analytics for driving business, which I had developed through his program."*

Masatake landed his dream job at Deloitte and has since grown his career to new heights becoming a data scientist at BCG Gamma.

Third is Rodrigo, who has both worked at a high-performance data science consultancy and now is co-founder of his own data science consulting firm.

Rodgrigo used the BSPF to cut project completion times in half (see Rodrigo's interview). This allowed his company to double their projects, which led to increased revenue.

Systematic Decision-Making

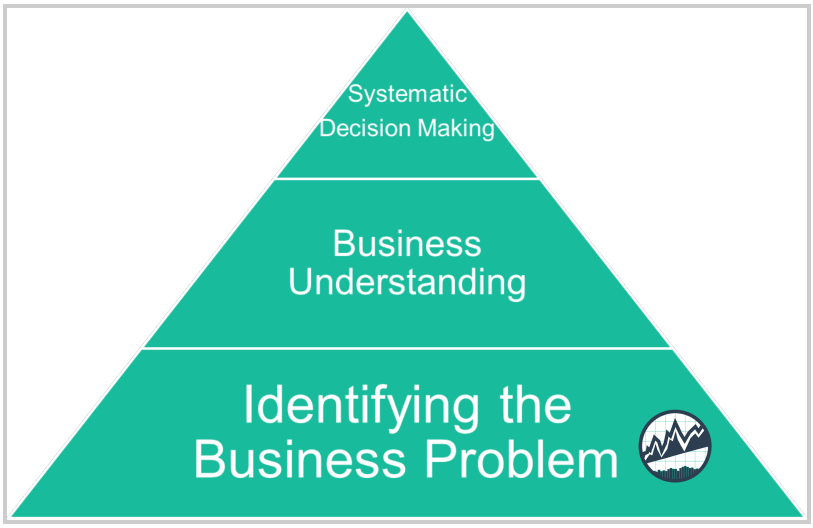
The goal is simple: to implement data science in a way that enables decision-making to follow a **systematic process**. We do this through the following equation relating measurement and analysis to improvement within a business context:

Measurement + Analysis = Improvement

Equation for organizational improvement via systematic decision making

The combination of measurement and analysis are critical for businesses that want to improve. Measurement, or collecting information typically in the form of data, combined with analysis, or digesting the information into usable insights, will lead to improvement. This *improvement* is driven by **Systematic Decision-Making**, or converting the learning that we achieve through *measurement* and *analysis* into processes that improve results.

The reality is that this equation is over simplified. Before we can implement Systematic Decision Making, we need to understand the business and identify the business problem. Thinking about this further, achieving Systematic Decision Making follows a path that can be visualized as a pyramid built on identifying drivers and understanding the business:

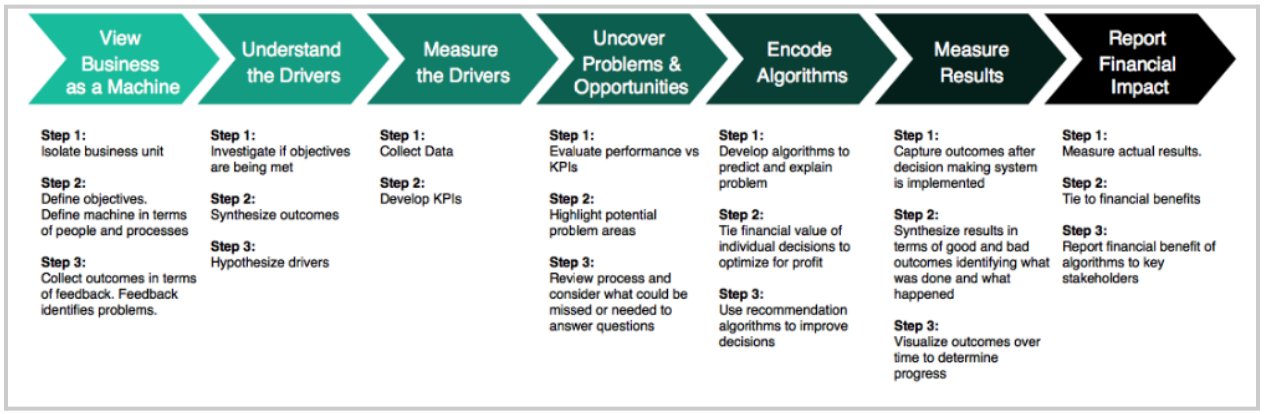


Identifying problems, understanding the business, and then converting the learning into systematic decision making is what the BSPF helps us do.

How Does the BSPF Work?

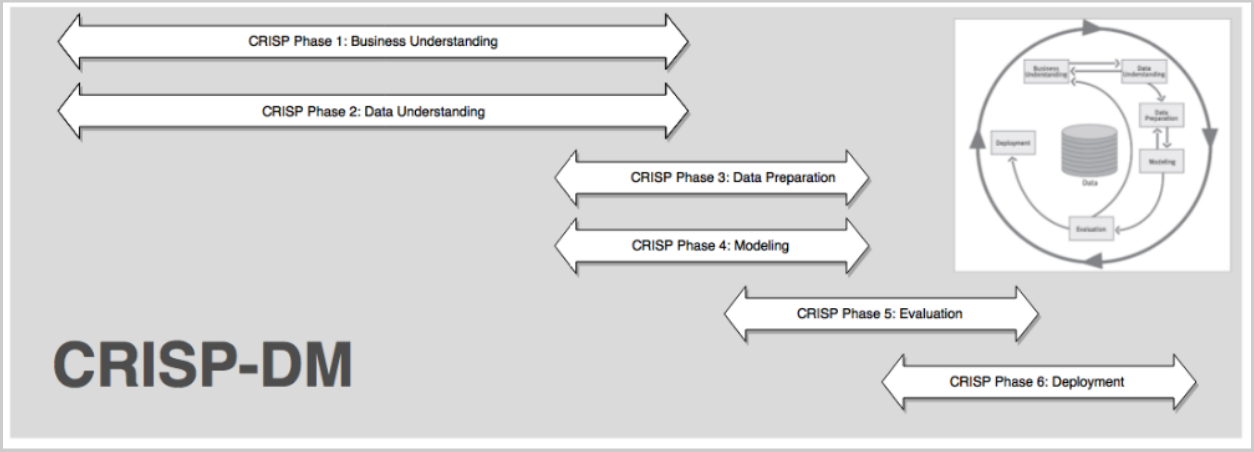
The BSPF is split into two sections: details of what to investigate (top) and high level stages of the project (bottom). The two sections together provide a complete program for managing a data science project in a business context. We get both high-level and detail in one package!

Part 1: Seven steps. The 7-Steps have actions (sub-steps) focused on understanding the problem and tying the results to Return On Investment (ROI), which is what the organization is keenly focused on:



Top Half of BSPF

Part 2: CRISP-DM. The seven BSPF steps flow along the six phases of CRISP-DM that are high-level stages for any data science problem (beyond just business):



Bottom Half of BSPF

Further, the BSPF is built on experience and best practice of business analysis. Many of the philosophies come from the writings of Ray Dalio (Refer to *Principles*) along with my experience using the BSPF with clients.

How To Use The BSPF: A Customer Churn Example

Customer churn refers to the act of customers leaving. These could be subscribers to a software or service or physical customers that choose to shop somewhere else. Customer churn often goes undiagnosed because individual customer impact is usually small, but when aggregated, the effect of churn can be large.

A good rule of thumb is that we only want to focus on problems that are $1,000,000 annually or more. The higher the cost, the more important it is to solve and the easier it is to save the organization money (for a return on investment).

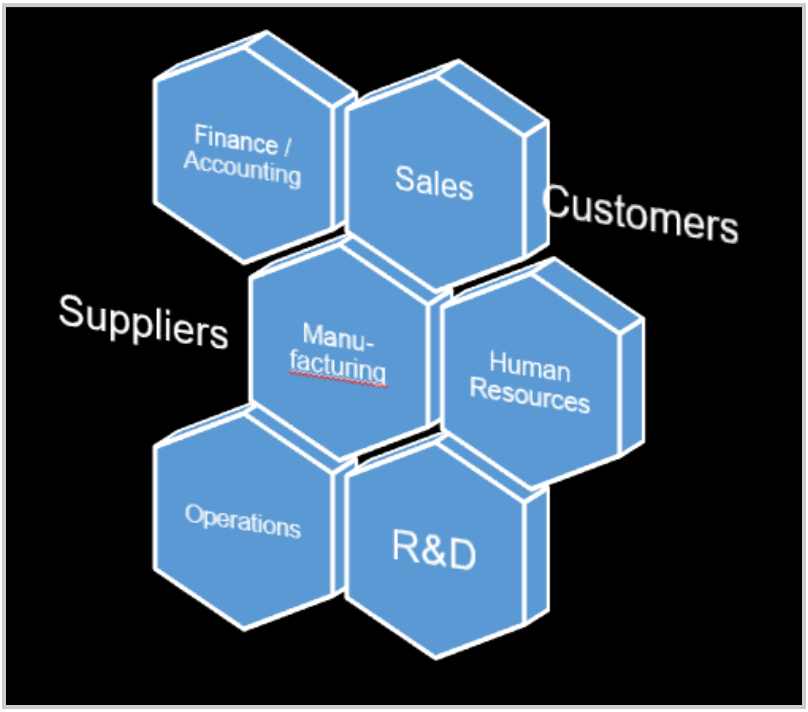
Step 1: View The Business As A Machine

The first step is viewing the business as a machine. This requires the business scientist to:

1. Isolate business units
2. Define objectives

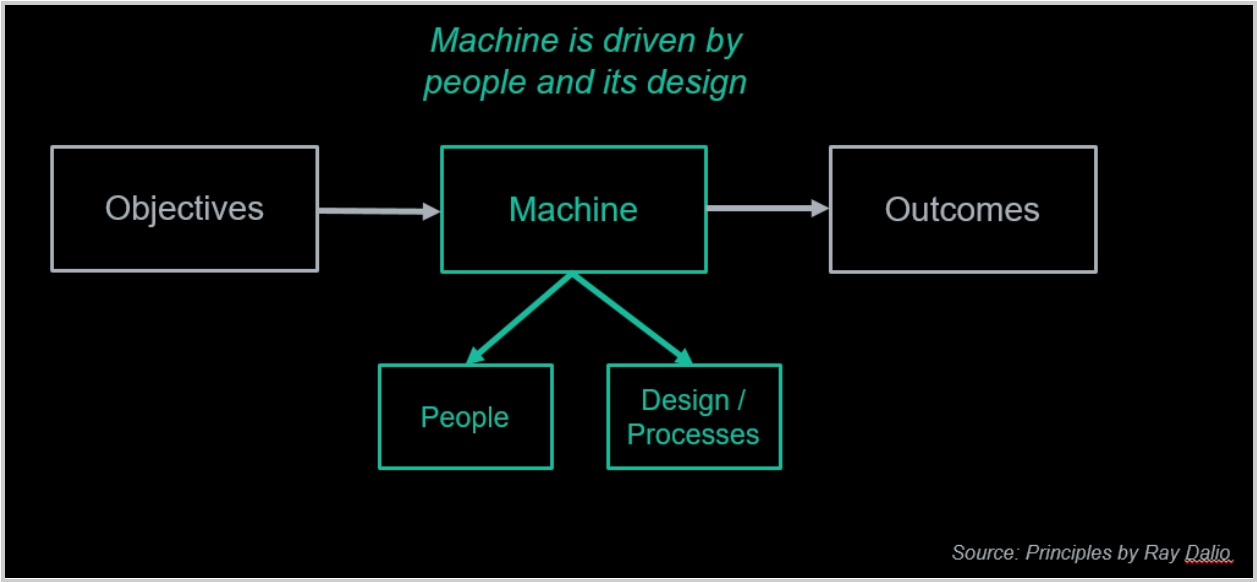
3. Collect outcomes

This involves breaking the business into internal parts (Sales, Manufacturing, Accounting, etc) and external parts (customers, suppliers) visualizing the connections.



Segmenting the business into components of the machine

We then need to visualize this interaction as a machine with goals and outcomes. The goals relate to business objectives, outcomes are what actually happens. The machine has inner workings, driven by people and processes, the process defines the setup, and the people execute the plan.



Visualizing The Business As A Machine

For the example customer churn problem, we make the following assessment:

1. **Isolate business units:** *The interaction occurs between Sales and the Customer.*
2. **Define objectives:** *Make customers happy.*
3. **Collect outcomes:** *We are slowly losing customers. It's lowering revenue for the organization by $500K per year.*

A key aspect in this stage is understanding the size of the problem. How is customer loss impacting revenue? $100, $100,000, or $1,000,000? If it's less than $100,000 then it may not be worth your time. If it's over $1,000,000 then get executives involved quickly.

Step 2: Understand The Drivers

Next, we begin the process of understanding the drivers. The key steps are:

1. Investigate if objectives are being met.
2. Synthesize outcomes.
3. Hypothesize drivers.

Start with the *business objective*: Customer Satisfaction. When customers are happy, they come back. Loss of customers indicates low satisfaction. This could be related to availability of products, poor customer service, or competition.

We need to *synthesize outcomes*. In our hypothetical example, customers are leaving for a competitor. In speaking with Sales, several customers have stated “Competition has faster delivery”.

The final step is to *hypothesize drivers*. At this stage, it’s critical to meet with subject-matter experts (SMEs). These are people in the organization that are close to the process and customers. What are the lead time drivers? Form a general equation that they help create.

$$LeadTime = f(SupplierDelivery, InventoryAvailability, Personnel, SchedulingProcess, \dots)$$

Developing a hypothesis with Subject Matter Experts (SMEs)

For the example customer churn problem, we make the following assessment:

1. **Investigate if objectives are being met:** *No, customers are unhappy.*
2. **Synthesize outcomes:** *Competitor has a faster lead time.*
3. **Hypothesize drivers:** *Lead time is related to supplier delivery, inventory availability, personnel, and the scheduling process.*

Communication is essential in this stage. As a data scientist, we know the tools really well but tools are only useful when we understand the drivers and the business problem. We need to educate ourselves by listening to SMEs.

Step 3: Measure Drivers

Now we begin the process of measuring the drivers.The key steps are:

1. Collect Data
2. Develop KPIs

First, we **collect data** related to the high level drivers. This data could be stored in databases or it may need to be collected. We could collect competitor data, supplier data, sales data (Enterprise Resource Planning or ERP data), personnel data, and more.

Collecting Data From Internal And External Sources

Second, we develop **Key Performance Indicators (KPIs)**, which are quantifiable measures that the organization uses to gauge performance.

For our customer churn example:

- **Average Lead Time:** 2-weeks, based on customer feedback on competitors.
- **Supplier Average Lead Time:** 3 weeks, based on feedback related to our competitor's suppliers.
- **Inventory Availability Percentage:** 90% based on where customers are experiencing unmet demand. This data comes from the ERP data comparing sale requests to product availability.
- **Personnel Turnover:** 15% based on the industry averages.

KPI Metric	KPI Level	KPI Basis
Avg Lead Time	2 weeks	Competitor lead time
Supplier Avg Lead Time	3 weeks	Competitor supply lead time
Inventory Availability Percentage	90%	Customer sales data
Annual Personnel Turnover in Planning Area	15%	Industry average

Developing Key Performance Indicators (KPIs)

There are two key aspects of this step. First, collecting data takes time. It may require establishing processes to collect it, but developing strategic data sources becomes a *competitive advantage*. Second, KPIs require knowledge of customers and industry norms. *Data is available outside of your organization*, learn where to access it.

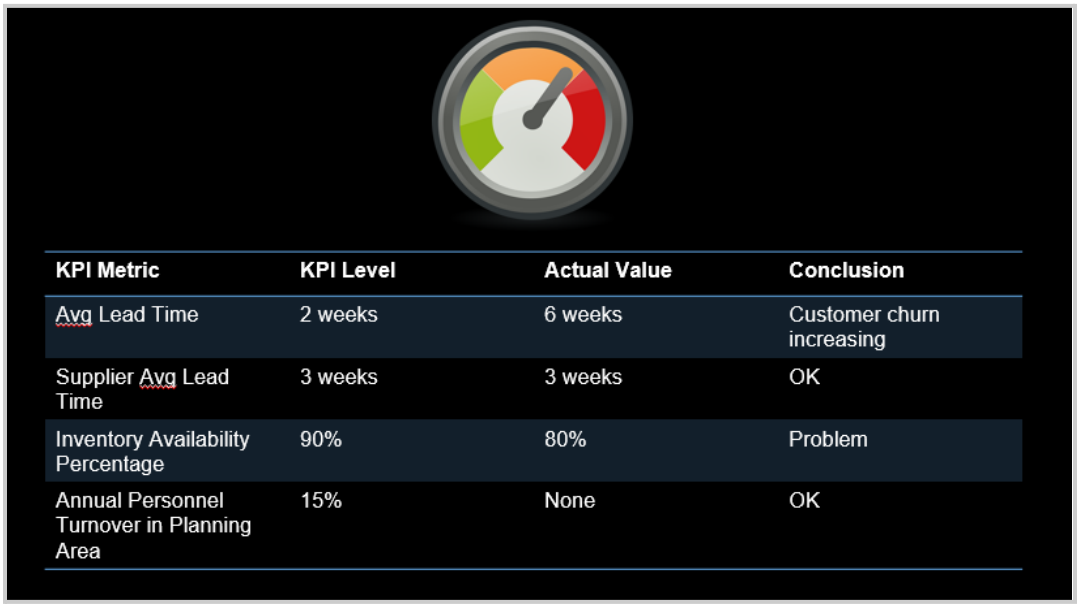
Step 4: Uncover Problems And Opportunities

In uncovering problems and opportunities we need to:

1. Evaluate performance vs KPIs
2. Highlight potential problem areas
3. Review the our project for what could have been missed

For our customer churn example, we review the results from organizational findings against the KPIs to determine where the problem areas exist. We extended the KPI table to include an Actual Value and Conclusion vs the KPI Level:

- [Concern] Lead time is 6 weeks compared to the competitor lead time of 2 weeks.
- [No Concern] Supplier lead time is on par with our competitor's.
- [Concern] Inventory percentage availability is 80%, which is too low to maintain a high customer satisfaction level.
- [No Concern] Personnel turnover in key areas is zero over the past 12 months.



KPI Metric	KPI Level	Actual Value	Conclusion
Avg Lead Time	2 weeks	6 weeks	Customer churn increasing
Supplier Avg Lead Time	3 weeks	3 weeks	OK
Inventory Availability Percentage	90%	80%	Problem
Annual Personnel Turnover in Planning Area	15%	None	OK

Performance Vs KPIs

Remember to ask questions and constantly test your assumptions. Talk with Subject Matter Experts to make sure they agree with your findings so far.

Step 5: Encode Decision Making Algorithms

The key parts in this step are:

1. Develop algorithms to predict and explain the problem
2. Optimize decisions to maximize profit
3. Use recommendation algorithms to improve decision making

First, develop algorithms using advanced tools like H2O Automated Machine Learning and LIME for black-box model explanations.

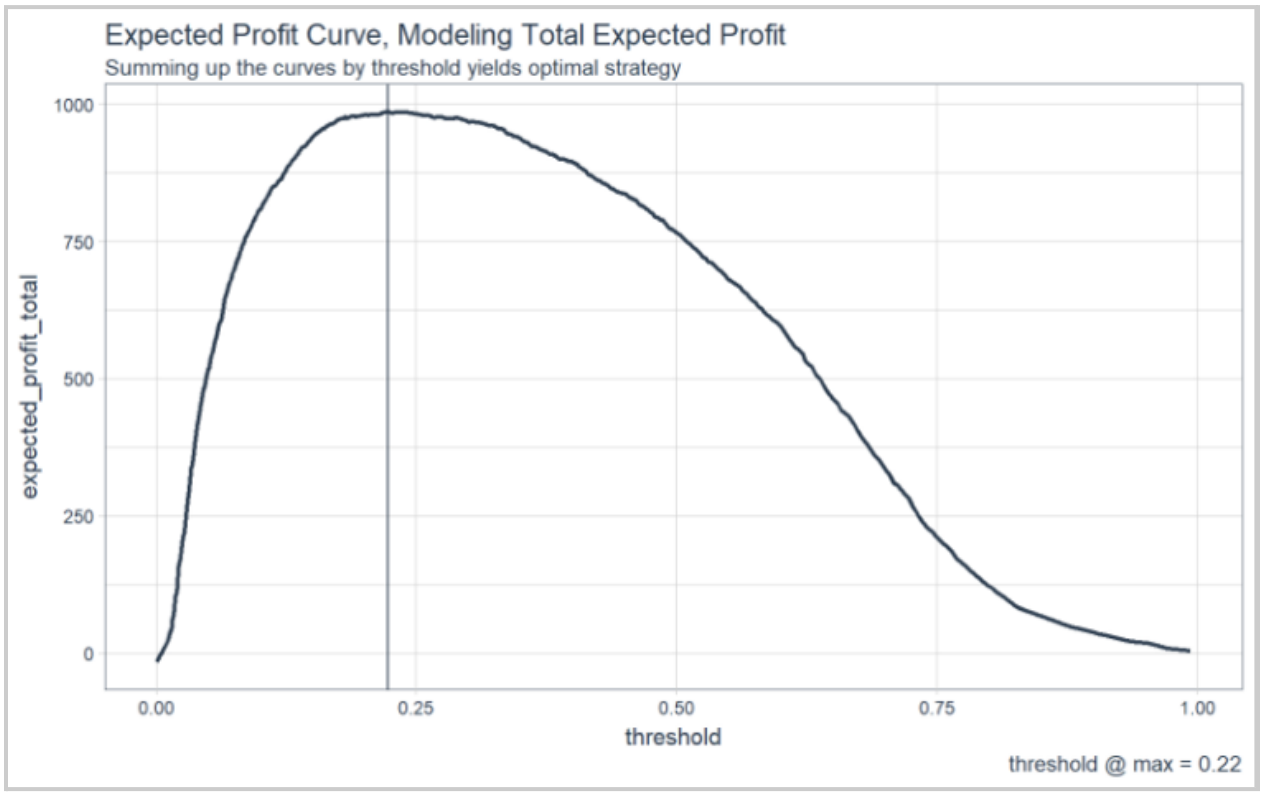
H2O is a great option because of Automated Machine Learning (AutoML). Automated machine learning is fast and develops highly accurate models, saving the data scientist time.

LIME is used to explain deep learning, random forest, and stacked ensembles, which are traditionally unexplainable.

```
h2o.init()

hr_data_bake_h2o <- as.h2o(hr_data_bake_tbl)

hr_data_split <- h2o.splitFrame(hr_data_bake_h2o, ratios = c(0.7, 0.15), seed = 1234)

train_h2o <- h2o.assign(hr_data_split[[1]], "train" ) # 70%
valid_h2o <- h2o.assign(hr_data_split[[2]], "valid" ) # 15%
test_h2o  <- h2o.assign(hr_data_split[[3]], "test" )  # 15%

y <- "Attrition"
x <- setdiff(names(train_h2o), y)

automl_models_h2o <- h2o.automl(
  x = x,
  y = y,
  training_frame    = train_h2o,
  validation_frame  = valid_h2o,
  leaderboard_frame = test_h2o,
  max_runtime_secs  = 200
)

automl_leader <- automl_models_h2o@leader

explainer <- lime::lime(
  as.data.frame(train_h2o[,-1]),
  model          = automl_leader,
  bin_continuous = FALSE
)

```

Sample R Script with H2O + LIME Machine Learning Algorithms Shown

Next, optimize decisions to maximize profit. Investigate threshold optimization for binary classification problems. Also, try sensitivity analysis to gauge which features have the largest effect on the profitability of the decisions.



Sample Threshold Optimization Visualization to Maximize Expected Return on Investment (ROI)

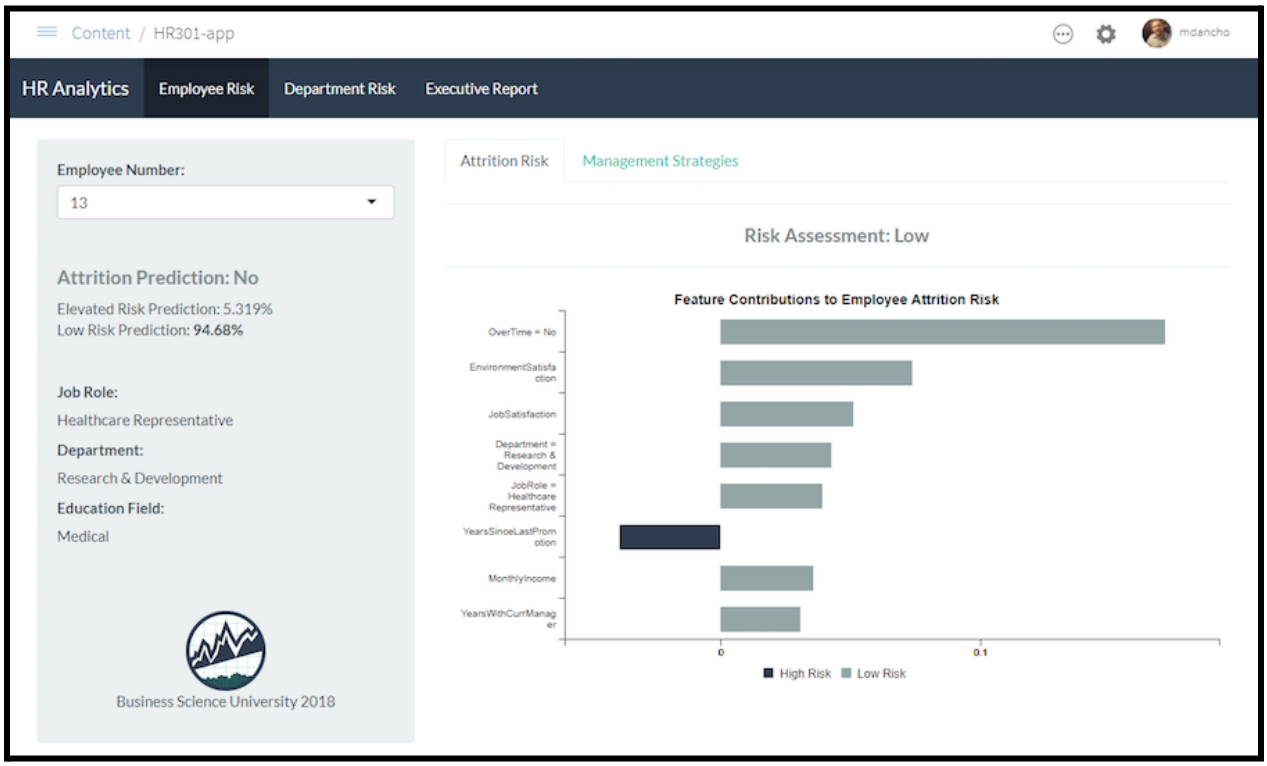
Last, build recommendation algorithms to improve decision making. Incorporate feedback from SME's along with the feature explanations from LIME (or similar feature explanation procedures).

```
pers_dev_strategy_tbl <- hr_data_raw_tbl %>%
    mutate(
        pers_dev_strategy = case_when(

            # Improve Concerns

            PerformanceRating         <= 2     ~ "Create Personal Development Program",

            # Address Working Time

            TotalWorkingYears         <= 5 |
                YearsAtCompany        <= 3     ~ "Promote Training and Formation",

            # Address Experience and Leadership Qualities
            (YearsInCurrentRole       >= 4 |
                YearsAtCompany        >= 7) &
                PerformanceRating     >= 3 &
                JobSatisfaction       == 4     ~ "Seek Mentorship Roles",

            JobInvolvement            >= 3 &
                PerformanceRating     >= 3 &
                JobSatisfaction       >= 3~ "Seek Leadership Opportunities",

            TRUE                                ~ "Retain and Maintain"
        )
    ) %>%
    select(PerformanceRating, JobInvolvement, JobSatisfaction,
           TotalWorkingYears, YearsInCurrentRole, YearsAtCompany,
           pers_dev_strategy)
```

Sample Recommendation Algorithm for Management Strategies

Step 6: Measure The Results

Once a systematic decision making algorithm is developed, it's time to deploy and measure results. A powerful tool is a Shiny Web App that can be used to predict churn and recommend management strategies.

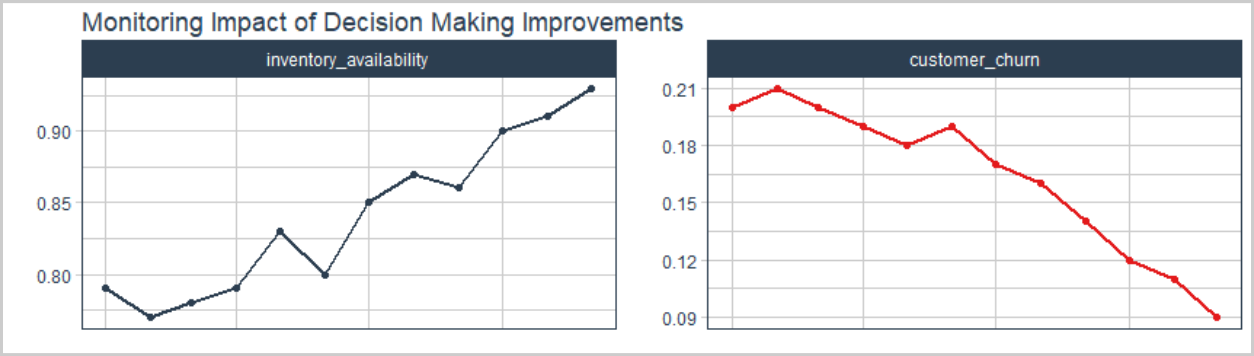


Shiny App That Predicts Churn and Recommends Management Strategies

After the web application is deployed, the results must be measured to show progress. This requires more analysis. We capture outcomes over time and synthesize results. We are looking for progress. If we have experienced good outcomes, then we need to recognize what contributed to those good outcomes.

- Were the decision makers using the tools?
- Did they follow the systematic recommendation?
- Did the model accurately predict risk?
- Were the results poor? Same questions apply.

For the customer churn example, we can make charts like these that expose the inventory availability and customer churn rate. We are seeing the inventory rise and the customer churn decline.



Visualizing Results Over Time

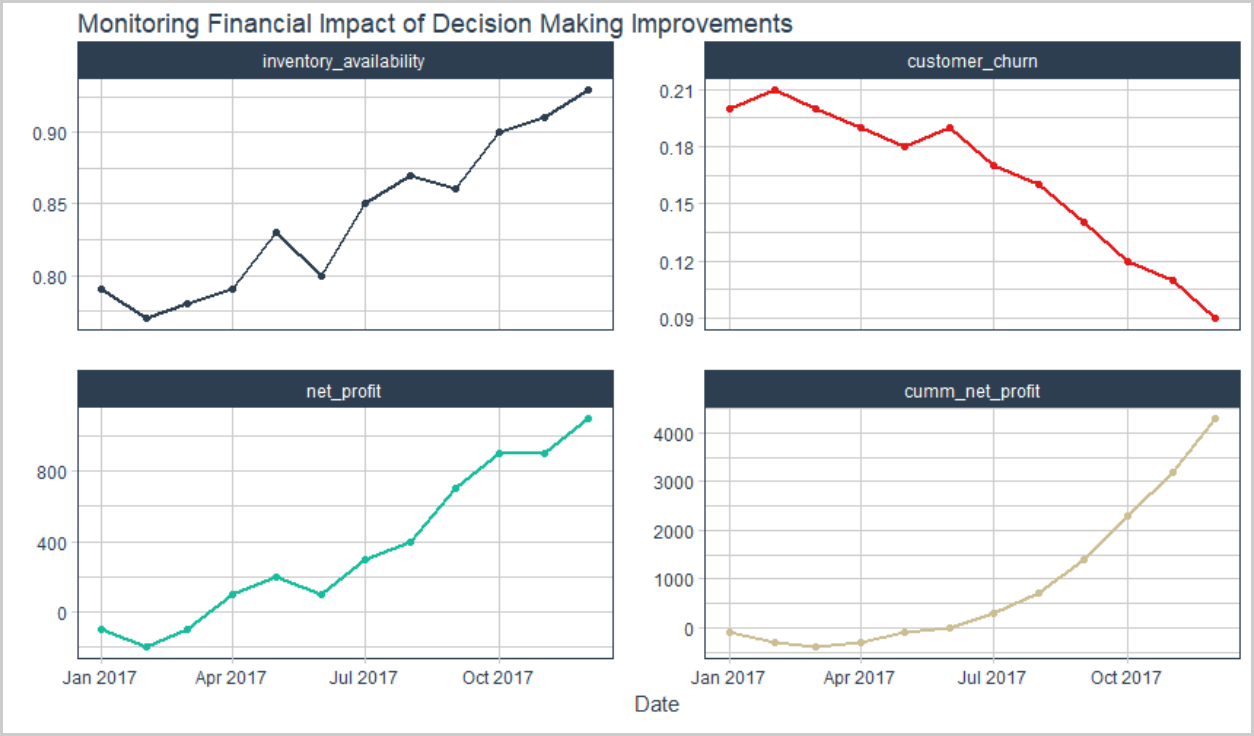
Step 7: Report Financial Impact

If we've done good data science, implemented systematic decision making, and iterated through problems, correcting along the way, we should now see positive results. Here are the steps:

1. Measure actual results
2. Tie to financial benefits
3. Report financial benefit to key stakeholders

Once results are understood, we need to show the results as financial benefits. This not only justifies our existence, but shows the organization that it is improving. The key here is that results must be conveyed in terms of financial impact.

When reporting to management, it's insufficient to say that we saved 75 employees or 75 customers. Rather, we need to say that the average cost of a lost employee or lost customer is $100,000 per year, so we just saved the organization $7.5M/year.



Measuring Return On Investment (ROI)

Example: The charts now show the net profit and cumulative net profit over time, and the accumulated savings is over $4,000,000 in under 1-year from implementation. These charts convey the success of the project and return on investment (ROI). Management can then see the relationship between the project results (increased inventory, reduced churn) and the financial results (increased profit and cumulative profit).

Next steps. How to Use the BSPF for your projects.

At this point you should have a high-level understanding of the Business Science Problem Framework, but it may not be clear how to perform a data science project inside of the BSPF Framework.

For students like Jennifer, Masatake, and Rodrigo, they learned how to apply the BSPF step-by-step to a $5,000,000 Human Resources problem to reduce employee turnover inside of my R-Track Program.

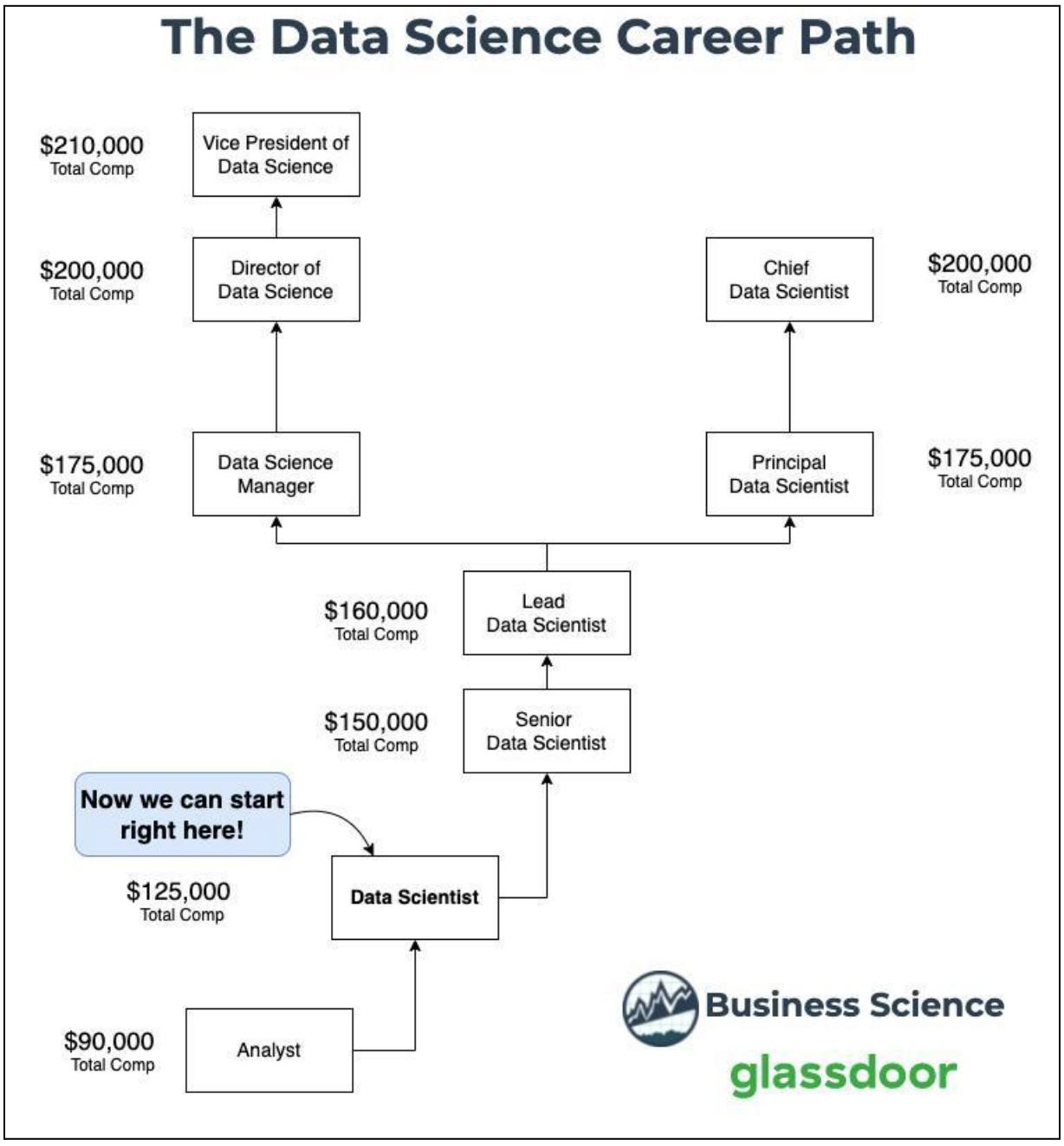
Learning the BSPF by applying it step-by-step through a large-scale business project was immensely helpful to Rodrigo, Jennifer, and Masatake. Doing a full data science project with the BSPF helped Rodrigo cut his project times in half (a big performance boost for his employer at the time). It helped Jennifer land her Senior VP of Analytics job with JP Morgan (pretty amazing, right?). And, it helped Masatake grow his data science career in data science consulting firms Deloitte and BCG Gamma. So you can see how valuable learning the BSPF by solving a project is, right? This project is available in the second course of my R-Track Program (more on this at the end of the book).

...

Next, I want to introduce you to your future career path once you become a Business Scientist. And spoiler, it doesn't need to take a long time to become a Senior Data Scientist or even a Lead Data Scientist. And did I mention that Senior and Lead Data Scientists make well over $150,000 per year? Sounds great, right? So let's learn how to carve out your future data science career path.

Chapter 3: The Career Path for a Data Scientist

From $75,000 to $150,000 salary in 1-year



The Data Science Career Path

It was 2018 and Mohana was a struggling business analyst. He'd been getting a measly 3.5% raise since he joined his company. Then in 2019 he got 1 raise (10%). And then in another 6-months Mohana got another raise. This time it was a 25% raise. And, then in just another 2 months, it happened again! This time a 40% hike.

In under one year, Mohana earned a 94% increase in his salary!

Today, Mohana is the Lead Data Scientist, at a Company called Money View, one of India's fastest growing startups that just recently closed a $75-Million Dollar Series D Round of investments.

Mohana is kicking butt, this time in a different capacity. As Lead Data Scientist, he's helping Money View grow their talented and high-productivity team as they move into a new phase of startup growth.

But how was Mohana able to double (2X) his salary in under 12-months?

What did he do to climb the career ladder so quickly and land a job wherever he wanted? And how did he maneuver his career into working at a leading startup, Money View, where he's now responsible for growing a team as a Lead Data Scientist?

The rest of this chapter will show you exactly how Mohana did it.

I'll answer questions like:

- What data science roles exist? (and which to steer clear of)
- What is the career path for a data scientist ($100,000 career)
- Which skills are needed to get promoted to Senior and Lead Data Scientist (you start at $150,000 career)
- Case Study 1: How to 2X your salary in 1-year ($75,000 → $150,000)
- Case Study 2: How to make a splash (How one data scientist saved his company $5,000,000 each year)

What data science roles exist? and which to steer clear of.

The first question I had when I was learning about data science was which roles exist and what did they make? Salary was a deal-breaker for me because I was supporting my family.

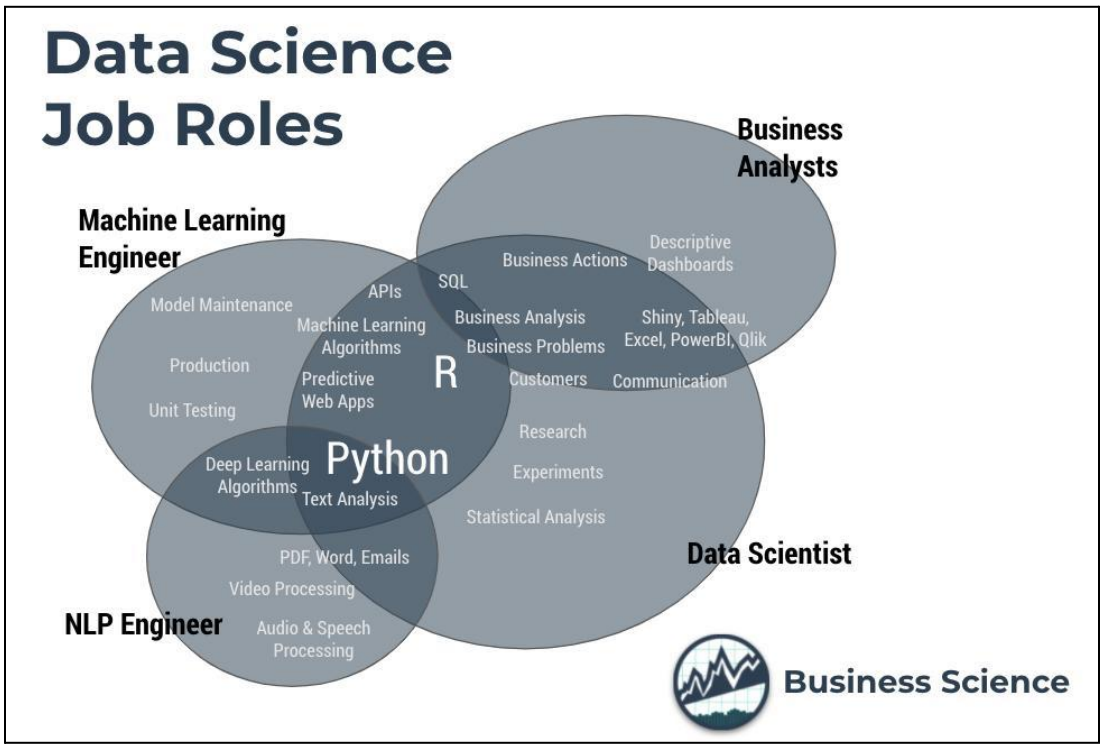
Job Role	Total Compensation (Per Year)
NLP Engineer	$129,542
Machine Learning Engineer	$122,483
Data Scientist	$121,068
Business Analyst	$90,013

Data Science Job Salaries (Glassdoor 2022)

I saw that Data Scientists and ML/NLP Engineers are being compensated 40% more than Business Analyst positions and there tends to be more pay for a specialization (Machine Learning and NLP).

Next, I wanted to know what each of these roles did.

What does a data scientist do in each of these roles?



Data Science and Analytics Job Roles

I saw that Data Scientists were more in line with what I wanted to be doing. They were doing things like research, business analysis, and applying algorithms to solve complex problems. This was really interesting to me.

Machine Learning Engineers were focused on model maintenance, production, testing, and deployment. This was interesting but somewhat advanced for my stage.

And I saw that NLP Engineers were involved in mainly video, audio, text using Deep Learning. I wasn't as interested in this because I didn't see how it could immediately help the business I was working for.

Then I dove into it further and really examined what each role did and their tools of choice.

Job Role	What they do
NLP Engineer	Specializes in natural language processing of unstructured data in the form of PDF, Word Documents, Surveys, Customer Feedback. Develops models that convert raw text into structured data for machine learning or deep learning models. Uses these models to automate insights from text. **Tool of choice:** Python code (typically).
Machine Learning Engineer	Specializes in taking models into production. Sometimes called MLOps for its close relationship with Development Operations (DevOps). **Tool of choice:** Python code (typically).
Data Scientist	Focused on experimentation, research, statistical analysis, and generating business insights through machine learning models and predictive applications. **Tool of choice:** R or Python code, Excel/PowerBI (sometimes)
Business Analyst	Least specialized. Responsible for reporting and descriptive analysis (not predictive). Analyzes customers, and develops dashboards. **Tool of choice:** Excel, Tableau/PowerBI, R (sometimes)

Data Science Job Roles Uncovered

I found that it's typical for engineering disciplines to use Python and the business analytical disciplines to use R/Python and Excel/PowerBI/Tableau.

My conclusion was that if you are looking to move from Business Analyst to Data Scientist, you should add R or Python to your skillset.

What about Data Engineering?

Why isn't Data Engineering one of the 4 main roles? Data Engineers make the jobs of the Business Analysts and Data Scientists easier by giving data scientists access to data in a format that comes from what they call a "data pipeline".

According to *"A day in the life of a data engineer"*, Data Engineers:

- Develop data pipeline/API/microservice.
- Setup/Maintain infrastructure.
- Fix bugs, improve code bases, and provide documentation.

Data Engineers are essential to the data science program success. But, as we are focused on the Data Science Career Path, it's more important to focus on downstream tasks like production and business results rather than upstream tasks like data engineering, which is why I'm excluding Data Engineering career path from the conversation.

For more information, I'll point to you to a data engineering guru like Andreas Kretz.

Mistake #1: Don't go for NLP or Machine Learning Engineering (right away)

If you want to migrate into specialized roles like ML Ops or NLP Engineering from a Data Scientist position, then I'm all for that. But, when you are just starting out, you're best served learning data science first before moving into the more specialized fields.

Remember, you can always learn more later (become specialized), but in the beginning it's important to gain general business domain and data science experience. Then make your key moves after learning the business.

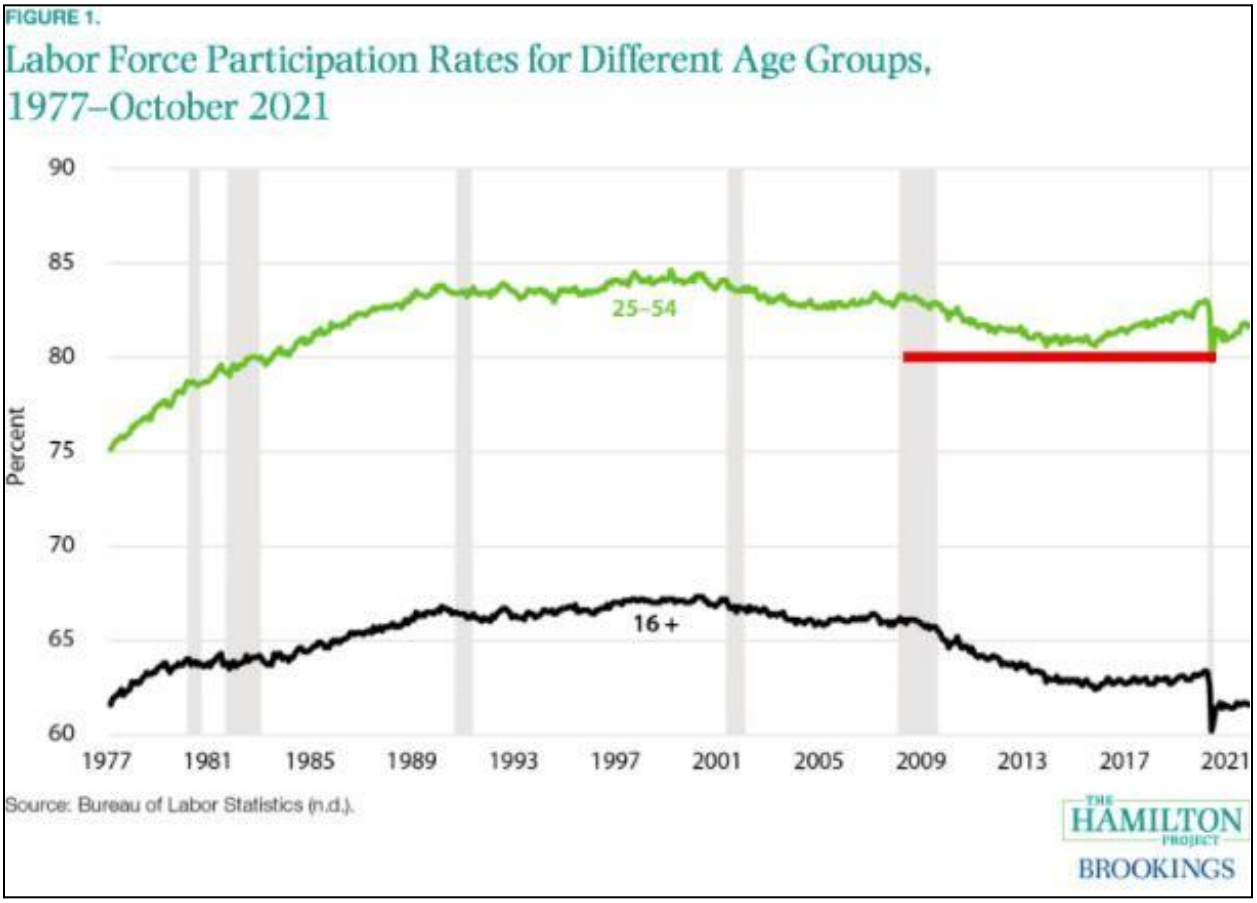
Mistake #2: Don't go for a Business Analyst Position (right now)

Avoid the business analyst position (RIGHT NOW!)

Popular Opinion: People should start as a business analyst, work there 2-4 years, and then migrated into data scientist positions.

Matt's Opinion: People are regularly getting 50% raises by snatching up lucrative data scientist positions. You should do that and here's why.

We are in a once-in-a-lifetime generational disparity between the number of data scientists available (supply) and the number of positions needed (demand).



Massive Labor Shortage = A Once-in-a-Lifetime Job Opportunity

In response to COVID, governments enforced a shutdown, forcing labor to decline swiftly and without notice. Upon reopening, not all workers came back. This created a supply imbalance

forcing companies to fill spots any way they could. In supply-and-demand theory, this is called the Bullwhip Effect. This led to…

Poaching Insanity

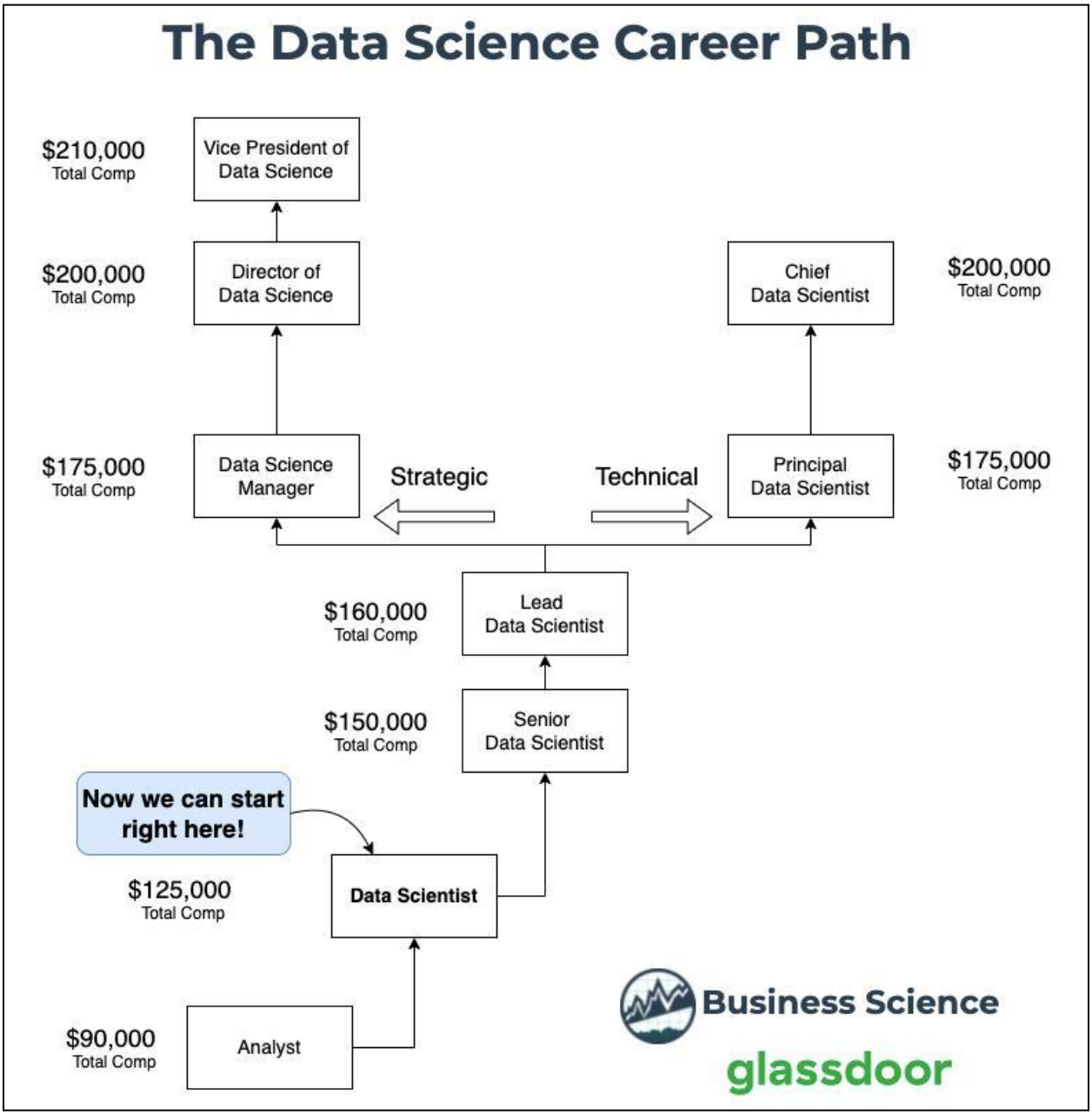
So what happened next is a once in a lifetime generational supply/demand imbalance that is working in your favor. Companies began poaching data scientists from other companies stealing their highest value assets: their employees. The training time for most new hires is 1-2 years, so companies had to offer higher salaries and benefits or be at risk of data scientists being poached.

Now you benefit because you can SKIP the "business analyst → data scientist" game and…

Jump right into Data Scientist roles RIGHT NOW.

What is the career path of a Data Scientist?

For 85% of organizations, the career path of a data scientist looks like this:



Data Science Career Path - Flow Chart

I've done the hard work of doing all the research on each of the positions. Here's what it looks like in table form:

Job Role	Total Compensation (Per Year)	Path
VP of Data Science	$210,000	Strategic
Director of Data Science	$200,000	Strategic
Chief Data Scientist	$200,000	Technical
Data Science Manager	$175,000	Strategic
Principal Data Scientist	$175,000	Technical
Lead Data Scientist	$160,000	General
Senior Data Scientist	$150,000	General
Data Scientist	$125,000	General
Business Analyst	$90,000	General

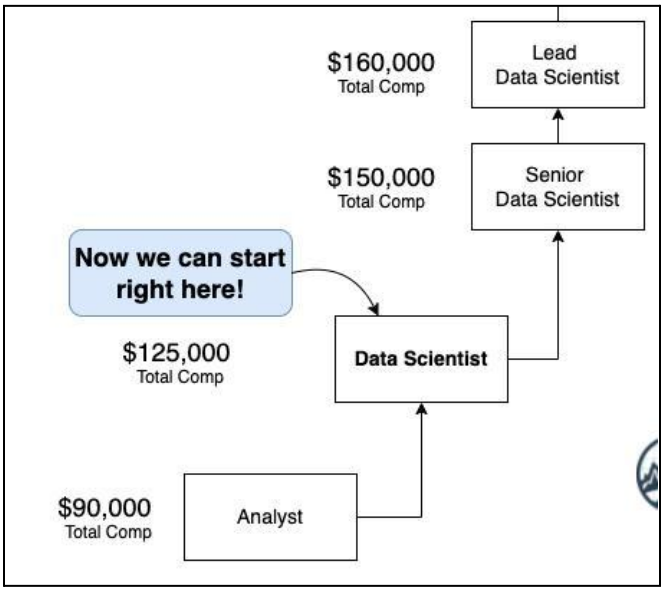
Data Science Career Path - Total Compensation (Glassdoor.com)

Things to keep in mind:

1. Get to Senior/Lead Data Scientist as fast as possible.
2. Choose a path - Strategic or Technical.
3. Stay on that path until you get to the top.

The General Data Science Career Path

Most organizations have a general track which will take you to a Lead Data Scientist (see figure to the right). You can start at Data Scientist right now, skipping the Business Analyst. Here's what you need to learn to climb the ladder.



You'll start as a data scientist making around $125,000 per year in total compensation. You will need the skills listed in *Chapter 1*. In fact, I even made a convenient cheat sheet to make it even easier (you can download the R-Cheat Sheet here).

Next, you'll become a Senior Data Scientist and make $150,000 per year in total compensation. Senior Data Scientists have more experience including big data, cloud (AWS, Docker, Git), and can do more advanced analyses when compared to entry-level data scientists. So, to become a Senior Data Scientist, learn big data and cloud, and learn to do more advanced analyses: Time Series, NLP, and Web Applications.

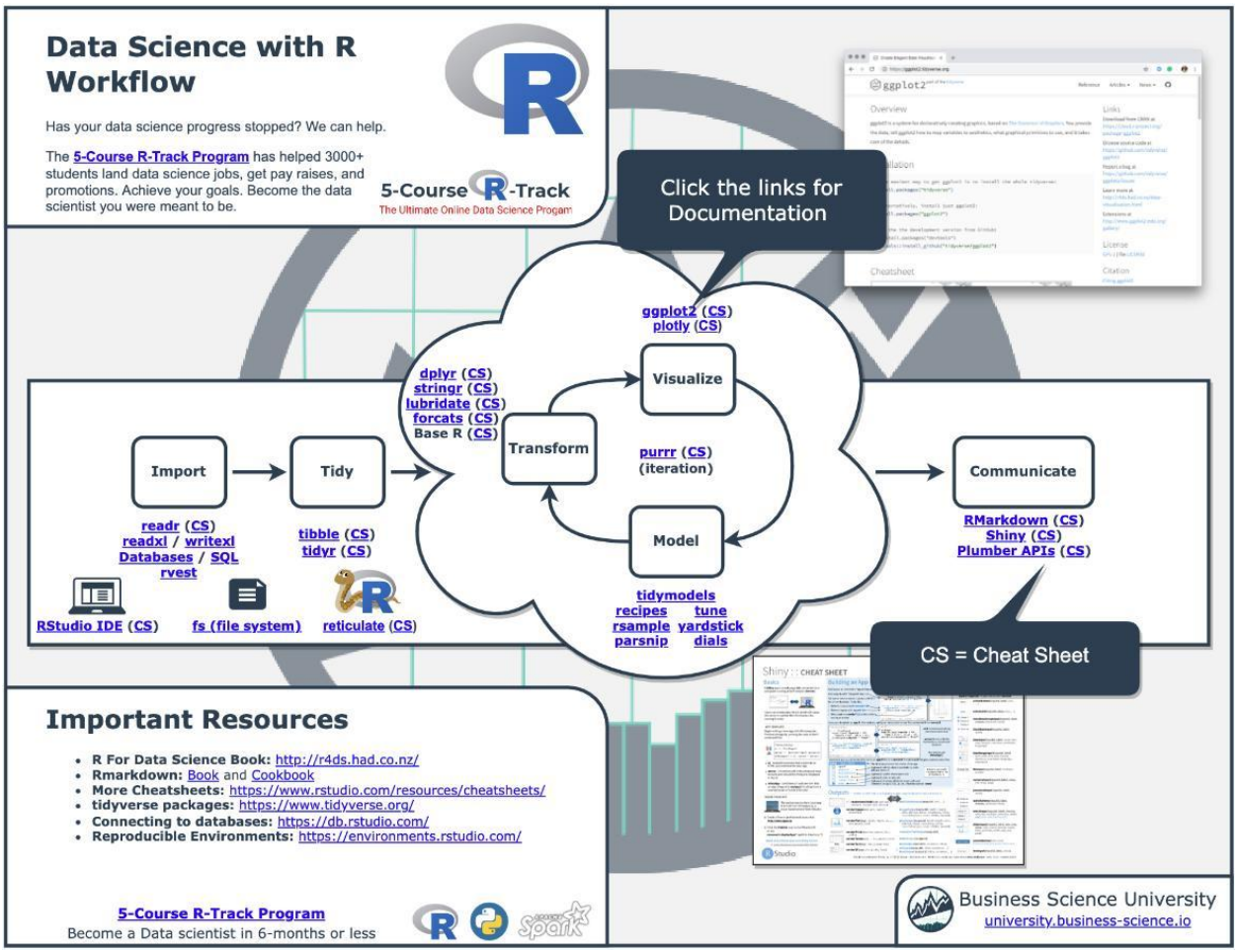
Next, you'll become a Lead Data Scientist and make $160,000 per year in total compensation. What really separates the Leads from the Seniors is their ability to work with Management, craft persuasive arguments, deliver insights (in the face of scrutiny), and they have well developed EQ (not just IQ). So learn to make and deliver presentations, work in teams, and build persuasive arguments.

If you really want to compare these three job roles, then I'll make it even simpler for you. Just learn these skills.

Comparing Senior/Lead vs Entry-Level Data Scientist

What skills do I need to become a Senior/Lead Data Scientist?

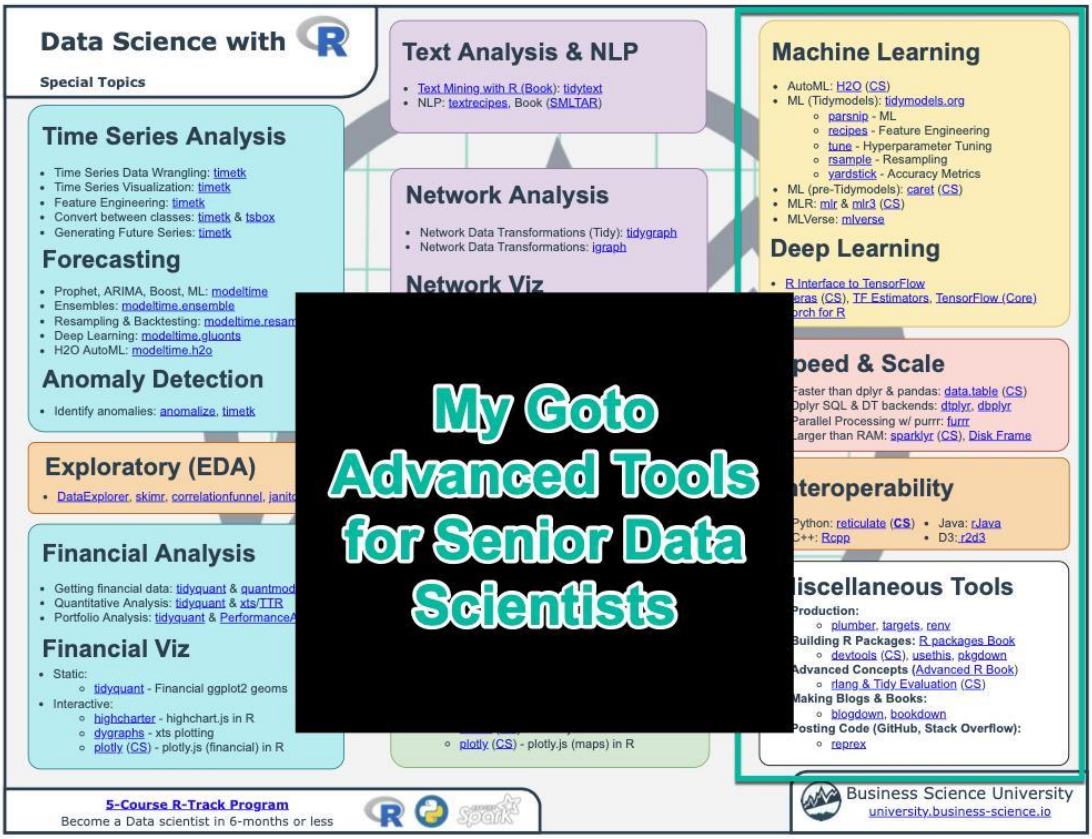
The easiest way to become a Senior/Lead Data Scientist is to cheat! What I mean is use my R-Cheat Sheet, which will help you learn the skills you need to go from Data Scientist to Senior Data Scientist.



The Ultimate R Cheat Sheet. It's OK to cheat.

How to cheat to become a Senior/Lead Data Scientist.

If we head to my R-Cheat Sheet (page 3) you'll find links to my goto-advanced tools for Senior/Lead Data Scientists.



Matt's Goto Advanced Tools for Senior Data Scientists

I'm going to give you a little secret. THIS is how the Senior and Lead Data Scientists separate themselves from the novice Data Scientists:

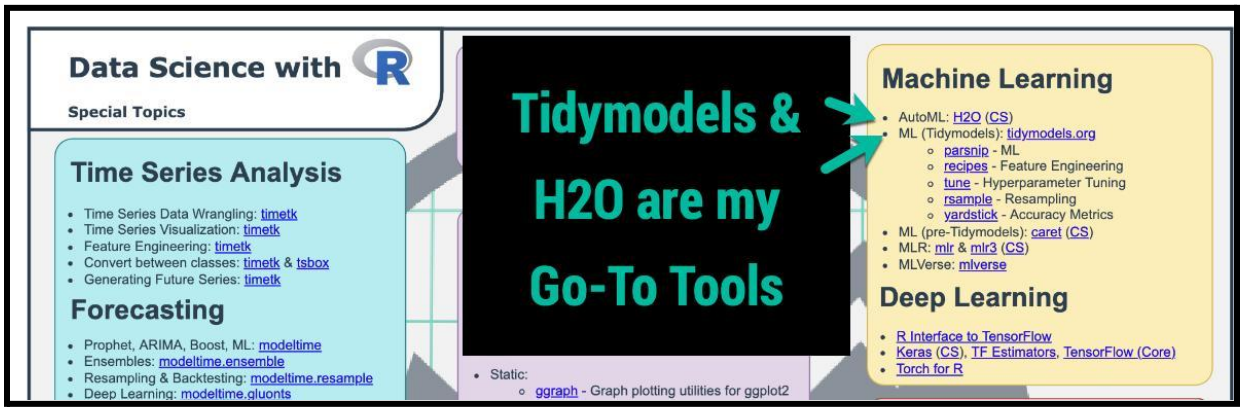
Advanced Machine Learning,

Feature Engineering, and

Cross Validation.

Advanced Machine Learning

In the section titled, “Machine Learning”, you have all of the most powerful tools used for advanced machine learning, feature engineering, and cross-validation/hyperparameter tuning. THIS is a goldmine!



Advanced Machine Learning

Here are my favorite machine learning packages (or ecosystems):

- **Tidymodels:** I use this for making adhoc models and then explaining
- **H2O:** I use this for automatic machine learning and in production

Another extremely important skill is feature engineering. I frequently use this package to create features:

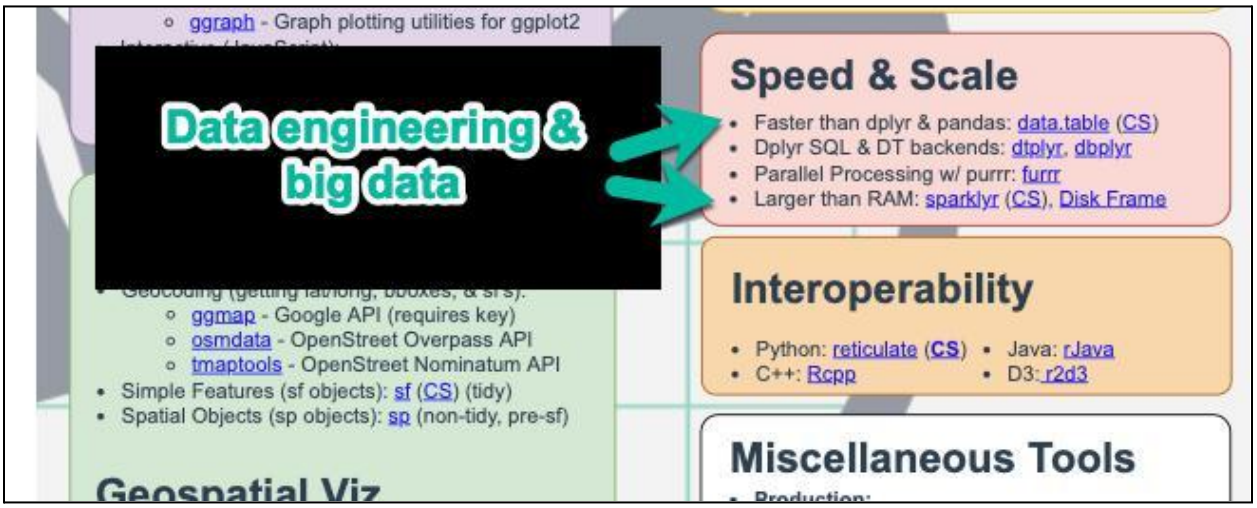
- **Recipes:** Has preprocessing tools to transform numeric data and create features from date, time, and text data.

These are my goto packages for hyperparameter tuning/cross validation:

- **Tune:** Hyperparameter tuning.
- **Rsample:** Resampling and cross-validation sets that are inputs to `tune`.
- **Yardstick:** Using pre-built accuracy metrics to minimize/maximize your loss during cross-validation.

Data Engineering (Big Data)

Another key skill is big data; working with data that is very large, sometimes SO large that it doesn't fit inside your computer's memory. But don't worry, I've got you covered here with some AMAZING packages. If we head on down a little further on Page 3 of the cheat sheet, we find a section called "Speed and Scale" and "Integrating Python".

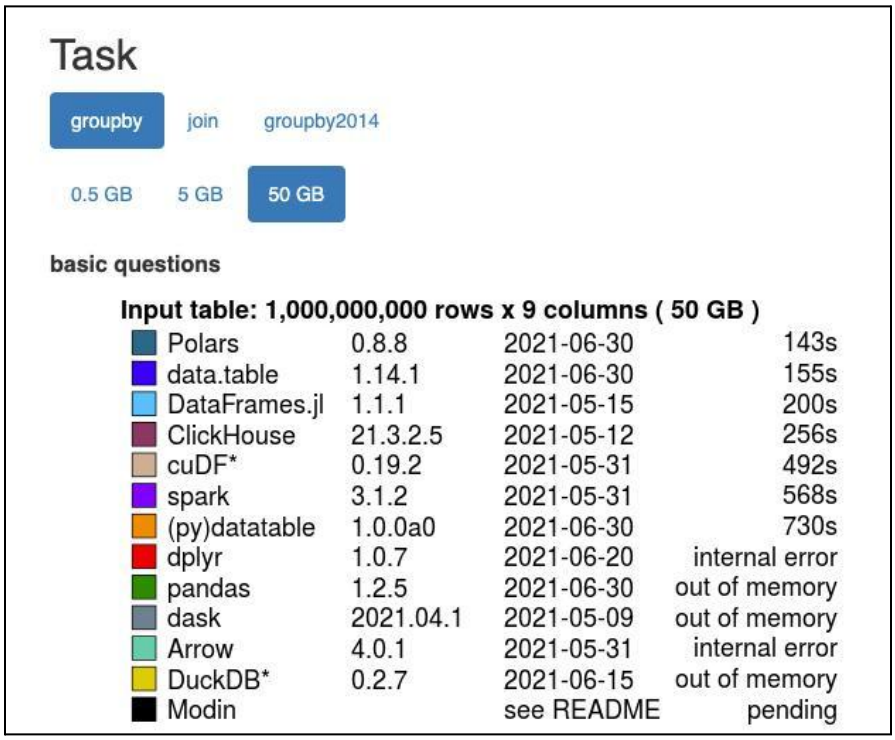


Data Engineering in R (Big Data Tools)

Let's talk about a couple of key packages: **data.table and dtplyr.**

data.table

This is the premier package for blazing speed. You can see how fast this is by exploring the Data Table Benchmarks here. It's faster than Spark, dplyr, pandas, dask, and most major data engineering and database softwares.



Data Table Speed Benchmarks

dtplyr

Now the big knock from tidyverse people (like me) that are used to dplyr is that the `data.table` syntax is strange. I eventually learned it, but people that want to skip the pain can use `dtplyr`. Dtplyr is the data table translator for dplyr. And, if you want to get up to speed quickly, I wrote a comprehensive dtplyr tutorial here.

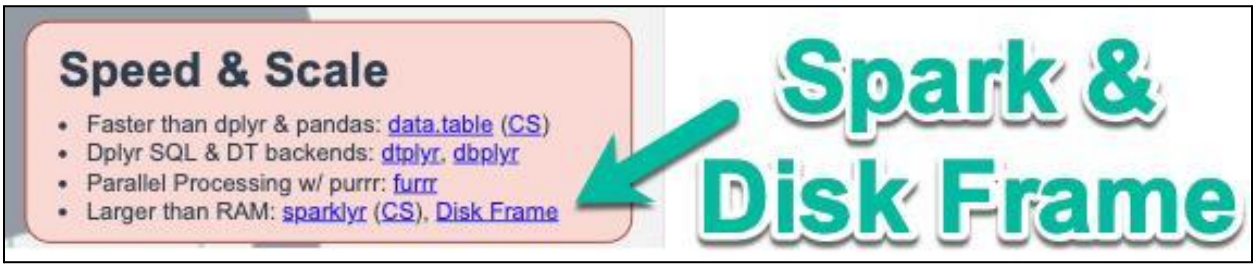
Databases

Next is databases. You'll want to learn dbplyr (not to be confused with dtplyr).

dbplyr: This stands for "database" dplyr and allows us to run dplyr scripts on your database, which is mindblowing! Databases are built for speed and scale (RAM is normally 1000X more than your puny macbook pro) and we don't need to transfer the data to our macbook until it's been chopped down, aggregated and summarized. I wanted to help you get up to speed, so I made a free dbplyr tutorial here.

Out-of-memory errors

Now sometimes you're going to run out of memory right before a presentation. This is what happened to me before I knew about these 2 packages:



Spark and Disk Frame (Fix Out of Memory Errors)

Head over to Speed and Scale (Page 3). Then click the links to sparklyr and Disk Frame.

sparklyr: Spark is a tool that runs on *cloud clusters* and allows you to do all of your big data analysis in the cloud! And even better, sparklyr allows you to run all of the computations using `dplyr` translations, which makes you **10X more productive** than your python counterparts.

But you're probably thinking, *"But Matt, I don't know how to do Spark from R. Can you help me?"* Yes! Here's my free *Spark in R Masterclass*. If you don't have access to a Spark Cluster, another AWESOME package is the little known `disk.frame`.

disk.frame: allows you to chunk your datasets into blazingly fast `fst` files, which can then be treated as a single dataset. Disk frame integrates with data.table and dplyr, meaning you can write translators no matter if you are data.table person OR a tidyverse person.

Python in R

The last thing that separates Senior/Lead Data Scientists from the entry level is the ability to use Python with R.

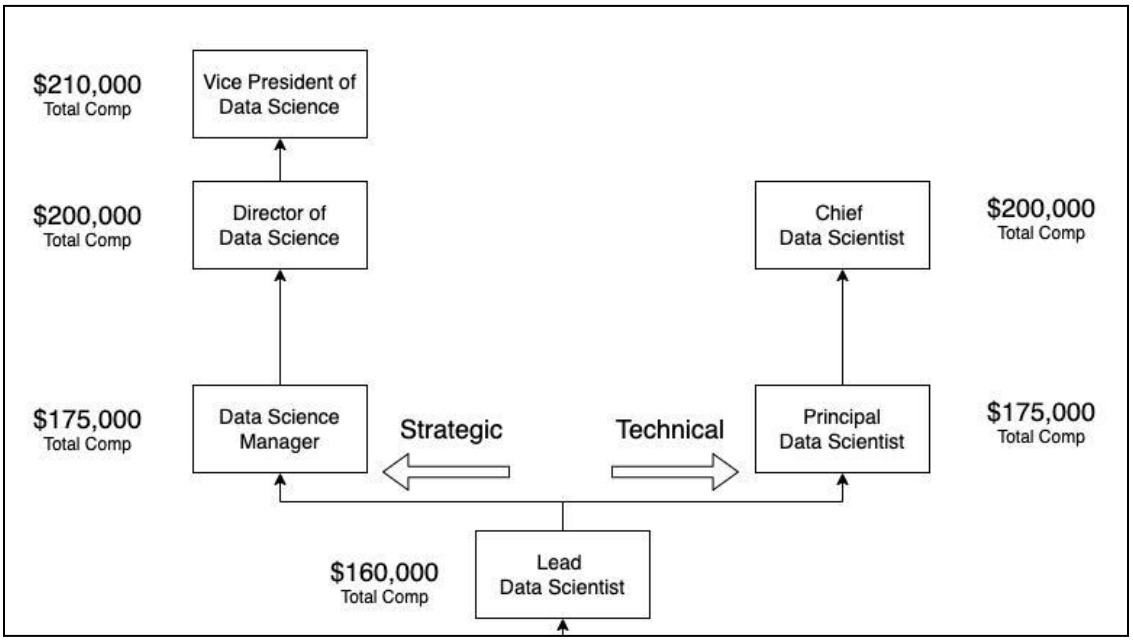
Reticulate: R's Python Connector

This is the most spectacular thing about R and will:

- **Empower you** to work collaboratively with Python teams (even though your an R user).
- **Give you** the key ingredient to make R packages that connect to Python packages. Example: An R+Python Package called `modeltime.gluonts` that connects to the GluonTS Python package for forecasting.

Now that you have the skills to become a Senior / Lead Data Scientist, what's next?

The Technical Path vs Strategic Path



Technical vs Strategic Career Path

There are 2 paths to the top: strategic and technical. So choose wisely.

How I chose my path.

I started as a data scientist without a title. 'Data Scientist' wasn't a position at my company. But I reflected on what I wanted to focus on. I liked the idea of influencing the direction of the company, I was entrepreneurial, and enjoyed working with people. I viewed business like a game of chess and I wanted to master it. My customers were my unsuspecting opponent. And I used data science to checkmate them into more revenue.

Can you guess which path I chose?

If you guessed "strategic", then in the words of Hulk Hogan, *"that's right, brother!"*

Strategic Path: Managers, Directors, and VPs of Data Science

Even though I chose the strategic path, I don't recommend it for everyone. Especially if you don't like dealing with personnel issues as a manager. But remember, it's mostly first line managers that deal with personnel issues. I actually didn't like this aspect. I learned to deal with personnel, then busted my butt to get promoted out of a line manager position as fast as possible. I eventually became a director, and my life was once again in harmony (like 38% of the time).

So what's my point?

Well, if you can't take personnel issues for a year or two then don't go into the strategic path. Go the technical path.

Technical Path: Directors and Chief Data Scientists

If you're reading this you might be saying, *"Director of Data Science and Chief of Data Science sounds great, but I'm nowhere near that level."*

I get it. But, listen, if you are reading this, you're probably also highly motivated. And guess what, those highly motivated people are the ones that eventually become directors and chiefs. So it would be a mistake not to explain to you the ins-and-outs of the entire data science career path. Not just simply how to double your salary.

So the next section will cover the secrets to getting promoted fast by sharing 2 case studies, which are the best way to learn.

How to get promoted (FAST)

There are three ways to get promoted fast. First, be more productive than everyone else around you. Simply outworking people is the first step. But it's not enough.

Second, you need to do something big!! (and repeat that performance). Doing something big gets you noticed, pads your resume, and leadership begins to notice you. The more frequently you do something big (2 or 3 times should work), then leadership will identify you for promotions and your career will take off.

Third, you can job hop. That's the quickest way to a 6-figure salary, but you also lose resume credibility if you do this too much. Be strategic in your job moves and find a position where you can make a BIG splash. Build your reputation, get promotions, and stay on the lookout for new job opportunities where you can continue this pattern.

So let's examine some real case studies of how to get promoted. Case studies are the best way to learn because you can see how people that came before you have done it.

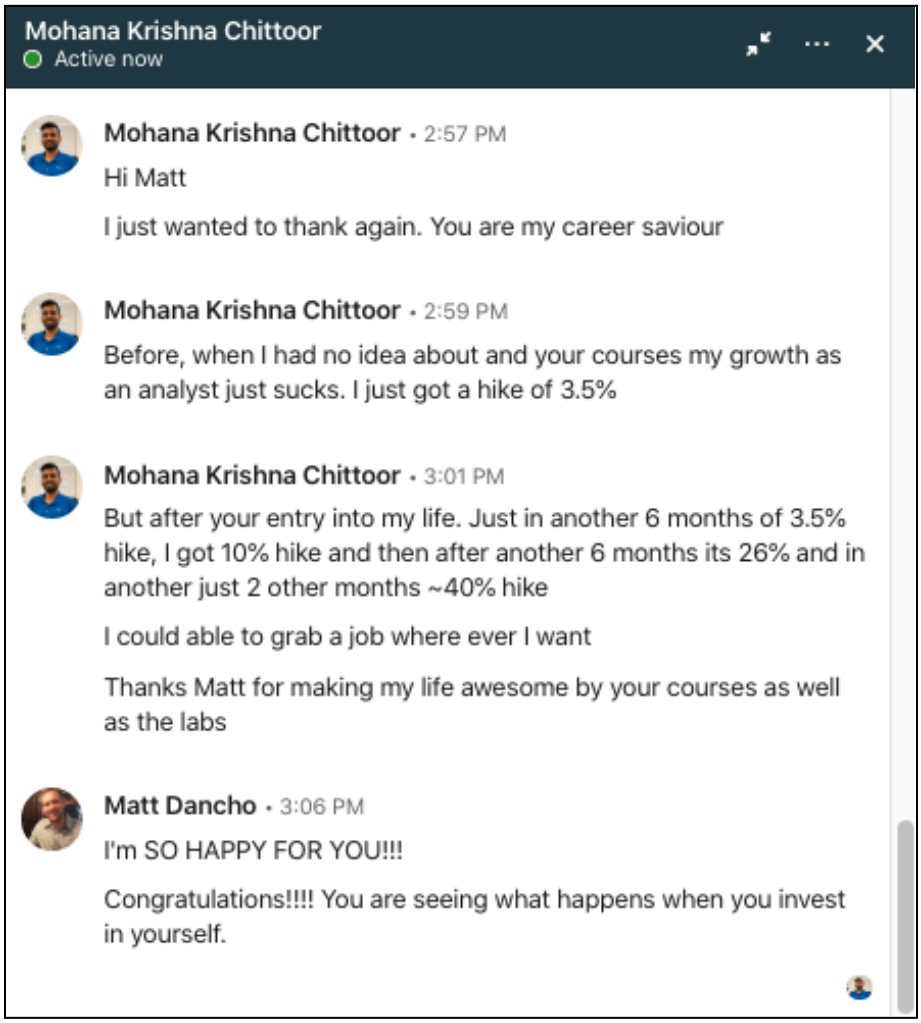
Here are 2 Case Studies of how to get promoted fast.

Case Study #1: How one data scientist 2X'ed his salary in 1-year

People are lazy. Most people aren't willing to go beyond their "job description", most people seek consistency over change, and most people get comfortable doing what they did the day before. You can exploit this.

Remember Mohana (from the beginning of this chapter)? He is the analyst that got 3 raises in the span of a year totaling a 94% increase. So if his salary was $75,000 starting out. By the end of the year his salary was $150,000. So, how did Mohana do it?

Mohana is a student of mine. One day, Mohana reached out to me on LinkedIn. Mohana said, *"I just want to thank you again. You are my career savior."*



He continues, *"Before when I had no idea about you and your courses, my growth as an analyst just sucked! I got a hike of 3.5% [per year]."After your entry into my life, I got a 10% hike, and then a 26% hike, and then a 40% hike."*

So what changed?

Mohana was working with Python coders. They were "comfortable".

Mohana wasn't like them. He's motivated. Mohana started working with me, and began the way of the Business Scientist. He committed to learning the way of the Business Scientist.

He enrolled in my R-Track Program, and it gave him the edge he needed to triple (yes, 3X!) his productivity versus his peers. What did he learn?

I taught him the way I code in R. He was able to write half the code and get twice as much done versus his python counterparts.

I taught him how to make hundreds of machine learning models in minutes. I gave him my playbook for business problem-solving with the secrets I used to spend less time on machine learning and more time on feature engineering.

I taught him the secrets to unlocking shiny web apps that his organization can use. While his Python counterparts were trying to get their first app launched, Mohana already had three done and launched all of them.

I taught him the hidden way to scale time series to 1000's of forecasts in minutes. This gave him a skill that no one else had in his company and increased his value 10-fold.

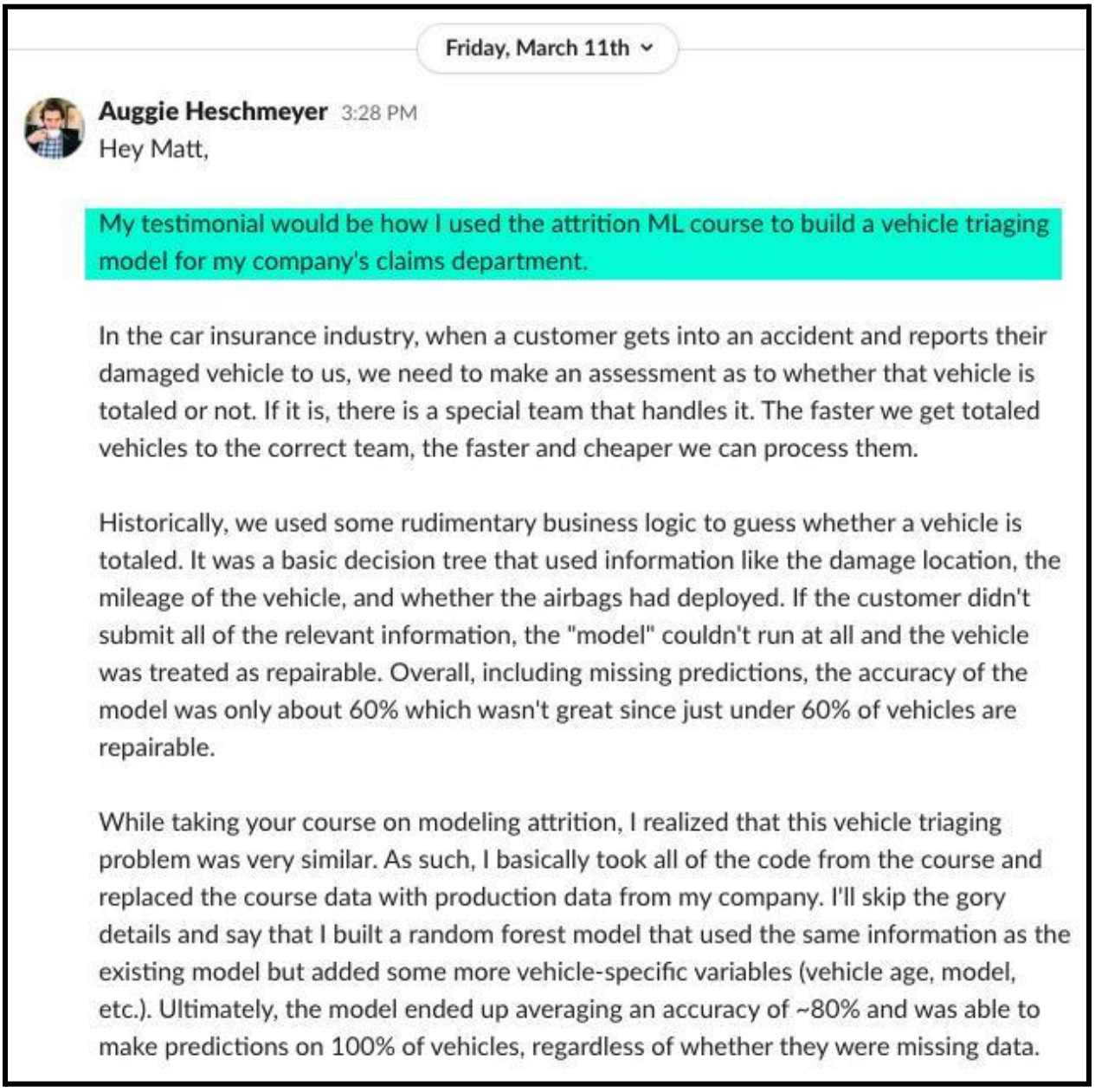
But here's the true secret.

Mohana applied what he learned to his job.

Then, Mohana applied what he learned to his business and now he's a Lead Data Scientist. Mohana kept repeating. He kept growing. He built his resume. He created a track record of success.

Today Mohana is now the Lead Data Scientist at Money View, one of the fastest growing startups in India. And they are about to grow even faster with the $75-Million Series D round of investment they just received.

Case Study #2: How a data scientist saved his company $5,000,000 per year



Friday, March 11th

Auggie Heschmeyer 3:28 PM

Hey Matt,

My testimonial would be how I used the attrition ML course to build a vehicle triaging model for my company's claims department.

In the car insurance industry, when a customer gets into an accident and reports their damaged vehicle to us, we need to make an assessment as to whether that vehicle is totaled or not. If it is, there is a special team that handles it. The faster we get totaled vehicles to the correct team, the faster and cheaper we can process them.

Historically, we used some rudimentary business logic to guess whether a vehicle is totaled. It was a basic decision tree that used information like the damage location, the mileage of the vehicle, and whether the airbags had deployed. If the customer didn't submit all of the relevant information, the "model" couldn't run at all and the vehicle was treated as repairable. Overall, including missing predictions, the accuracy of the model was only about 60% which wasn't great since just under 60% of vehicles are repairable.

While taking your course on modeling attrition, I realized that this vehicle triaging problem was very similar. As such, I basically took all of the code from the course and replaced the course data with production data from my company. I'll skip the gory details and say that I built a random forest model that used the same information as the existing model but added some more vehicle-specific variables (vehicle age, model, etc.). Ultimately, the model ended up averaging an accuracy of ~80% and was able to make predictions on 100% of vehicles, regardless of whether they were missing data.

Auggie was a data analyst for Root Inc, an insurance firm, where he specialized in car insurance analysis. In the car insurance industry, his company needs to make assessments of whether or not vehicles were totaled in collisions. An incorrect assessment can be very costly to the car insurance firm.

Through my R-Track Program, Auggie learned the necessary skills to build complex attrition models. He was able to apply the Business Science Problem Framework in my courses to his

business problem. And he realized that the processes used to *hypothetically* save millions of dollars for an employee attrition problem was actually quite similar to his company's vehicle collision assessment problem.

And Auggie was able to save Root, Inc $5,000,000 per year.

Talk about making a BIG splash!

I think the most impactful part of the project, though, came from tuning the decision threshold. I had played around with classification models before your course but I always naively used a threshold of 0.5 when classifying a record. What your course taught me was that building the model was the easy part; tying it back to a cost-benefit analysis was the part that really delivered value. As such, I worked with stakeholders in the space to understand the costs of accurate and inaccurate predictions. We found out that false negatives were worse than false positives (storing vehicles in a body shop is more expensive than doing so in a junk yard). As such, we set the threshold to 0.45 instead of 0.5. This minor tuning of the threshold enabled us to control costs in a way we never had before and it was estimated that the new model was going to save us $400K/month at Oct '20 volume. We processed even more vehicles in 2021 so that number is probably underrepresentative of the true savings.

The project was a huge success. I got a personal message from the CTO and the CEO just mentioned the model in our most recent investor call. Today is my last day at the company but I have left the model in the hands of our new claims data science team where they have two data scientists working to make the model even better. We estimate that every percentage point that they can improve accuracy will result in savings of $40K/month. While I didn't get a salary adjustment or an invitation to the data science team, the technical expertise, business context, and project management skills displayed during the project were a major consideration factor in my promotion to Analytics Manager a few months later. And it was all thanks to the skills I picked up in your course.

Thank you.

Matt Dancho 3:31 PM

Oh wowowow!! This is exactly what I'm looking for. This is the stuff I never new about how it's impacting you and your company. (edited)

Auggie reached out to me to explain his amazing accomplishment. Using what he learned through My R-Track Program, Auggie made a better model. In fact so much better that:

Auggie's model saved the organization $400,000 every month! A quick math check means that Auggie saved his organization $4,800,000 per year. And these estimates may actually be low (meaning the model is likely saving more).

Auggie was recognized. *"The project was a huge success. I got a personal message from the CTO and the CEO mentioned the model in our most recent investor call."*

Auggie was rewarded with a promotion to Analytics Manager. *"The skills displayed during the project were a major consideration factor in my promotion to Analytics Manager a few months later. And it was all thanks to the skills I picked up in your R-Track Program."*

Build your career by following our trailblazers.

Students like Mohana and Auggie are trailblazers. They have done what was previously thought of as *impossible*. Learning data science, and doubling their salaries in the process. Sounds amazing, right?

The reality is, it's *not* impossible. In fact, growing your career follows a repeatable pattern. Learn the skills by following an expert mentor that can guide you. Then look for opportunities to apply your skills. Outwork your peers. Make a BIG splash. Then get promoted.

This is what can happen to you if you follow the way of the Business Scientist. And it starts by committing to learning. It would be my honor to teach you my ways. All you have to do is commit and take control of your future. I will do everything in my power to help you succeed. I'll talk more about how I can help at the end of this book.

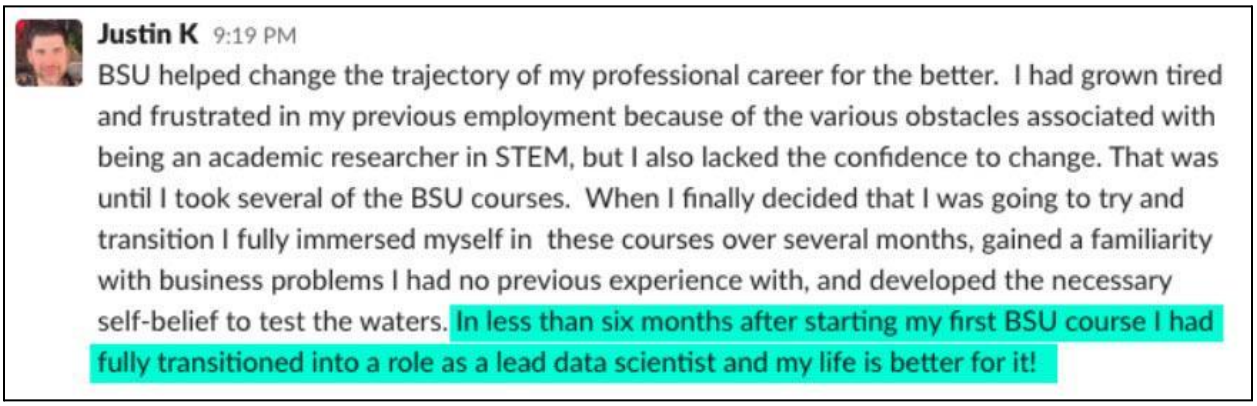
...

You now have all of the information that is needed to take you from a $75,000 salary to a $150,000 salary in under 12-months. Next, I want to cover another story of becoming a Financial Data Scientist. This will help you focus on what organizations really need. And guess what - This story doesn't just apply to "financial" data scientists. It applies to any field, industry, or domain. Let's see how you can do it.

Chapter 4: How To Become A Financial Data Scientist

Or A Data Scientist In Marketing, Manufacturing, IOT, Health Care, Pharma, … or literally any domain!

In December of 2020, Justin was working at the University of Southern Mississippi doing sports analysis. But he was not satisfied. By June of 2021, 6 months later, Justin got his dream job as the Lead Data Scientist at Northwestern Mutual (one of the biggest financial & insurance firms in the US). Just reached out to tell me the good news.



But I had one question.

How did Justin do it?

So I set out to answer exactly this question. I dove into researching his path and the paths of other data scientists that transitioned in a short period of time. In this chapter, you will learn what I found out:

- Why becoming a unicorn is slowing you down
- The 2 big mistakes you're making (I made these too)
- How to market yourself as a data scientist (The 3 things organizations value)
- Case Study: A real world example of how to provide value to a business
- How to earn a $125,000 salary (in under 6-months)

Finance is just a domain

I'm a big fan of context and case studies. So in uncovering how to become a data scientist, I analyzed a specific domain, finance. One that I'm very familiar with. But, if you are not a financial person, I have amazing news: the process for creating business value is *exactly* the same. Let me explain.

For financial people

If you are a financial professional (or any other professional) seeking to learn data science, then **this is what you've been waiting for.** If you understand what financial organizations value, you then know what skills to learn to streamline your path from where you are now to being a productive member of a Financial organization.

For non-financial people

The same strategies I cover for "finance" can be applied broadly to **ANY domain**, as wide-ranging as Marketing, Research & Development, Medicine, and more. In fact, I've used the same tactics you'll read about in Human Resources Analytics, Marketing Analytics, and more. Seriously, this is powerful and you can apply it to any business problem in domains as wide-reaching as IOT, Manufacturing, Pharmaceuticals, Biological Analytics, and even for analyzing Customer Call Centers. So please read on.

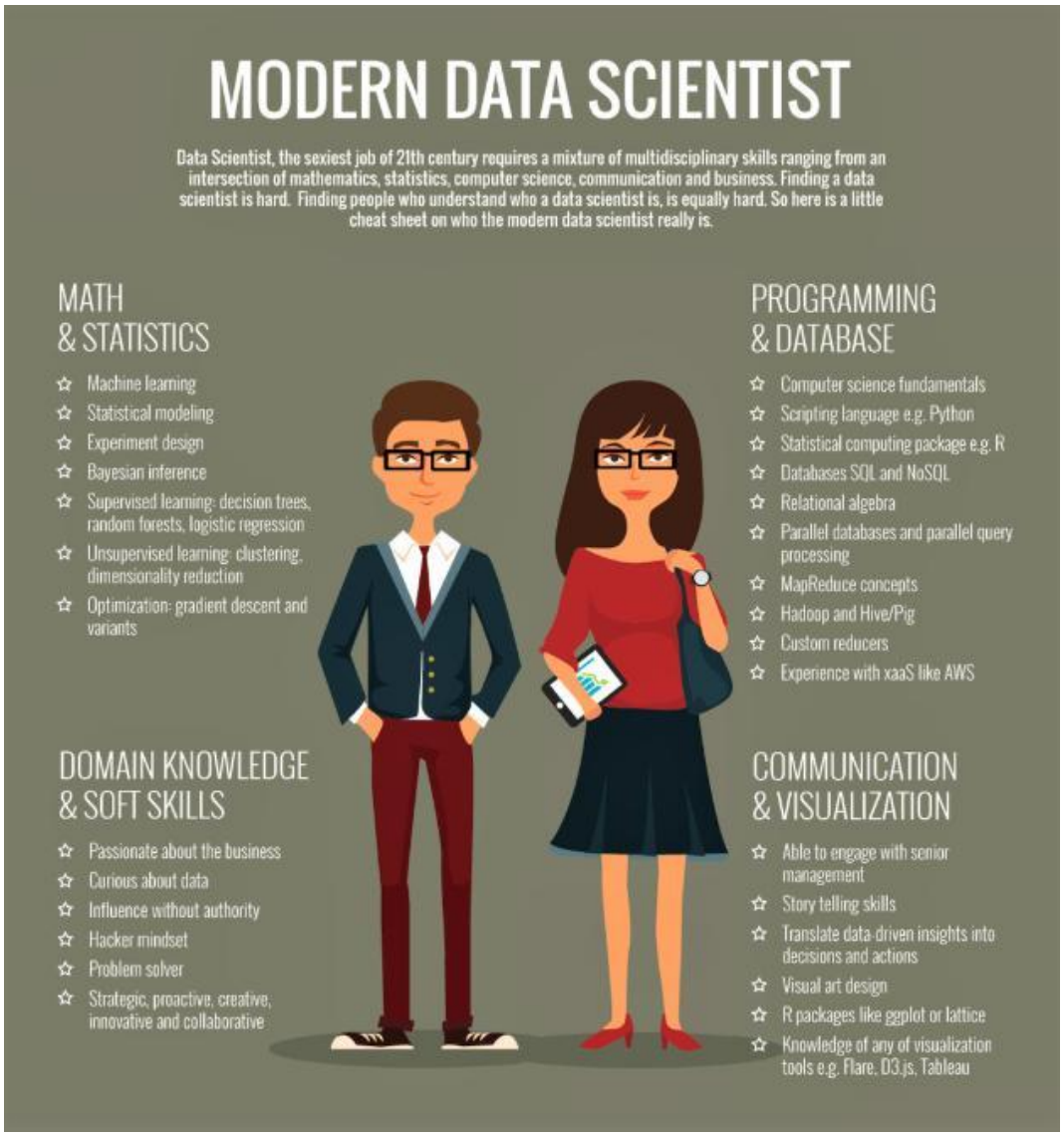
Now, one thing I need to mention - If you are reading this and don't have knowledge on *Portfolio Theory* and *Risk Management*, keep in mind the strategies for learning what organizations value are most important. Not the domain jargon (the terminology used commonly in finance like Sharpe Ratio or Beta).

What I will show you is how you can take domain knowledge and extend it to value-creating activities for the business using data science tools.

But, first, the cold hard reality…

Why becoming a unicorn is slowing you down.

Most people that get into Data Science make a **major a mistake** that costs them a career in data science: They begin down the path of learning everything and get overwhelmed. Learning *everything* is costly because everything is not useful, at least not in your current moment.



"Modern Data Scientist Infographic" - A useless smattering of skills

This graphic is <u>not</u> a strategy. It has no foundation, no purpose, no intent. It's just a useless smattering of skills that supposedly creates a modern data scientist.

The Online Data Schools promote this type of "learning plan". They tell you to, *"take 50 courses on every subject, and then we'll certify you as an official data scientist."* This is a trap.

As soon as you go down this road, you adopt the mindset of the **unicorn data scientist**. A unicorn data scientist is a mythical creature that doesn't really exist. This unicorn is a "master" in every type of project and problem. But, the truth is that the unicorn data scientist is an *impossible* goal. No one can ever be a master of everything.

When we start out learning data science, we are the most vulnerable to making missteps like **the Way of the Unicorn**. I personally remember feeling overwhelmed and directionless. It's at this moment that we are easily influenced to begin learning everything (Data Schools love this because it keeps your subscription incoming forever)... And it's a costly mistake. Especially with so much to learn. Add to it that every misstep **costs us time**, and it's easy to see why many data scientists struggle and many don't succeed.

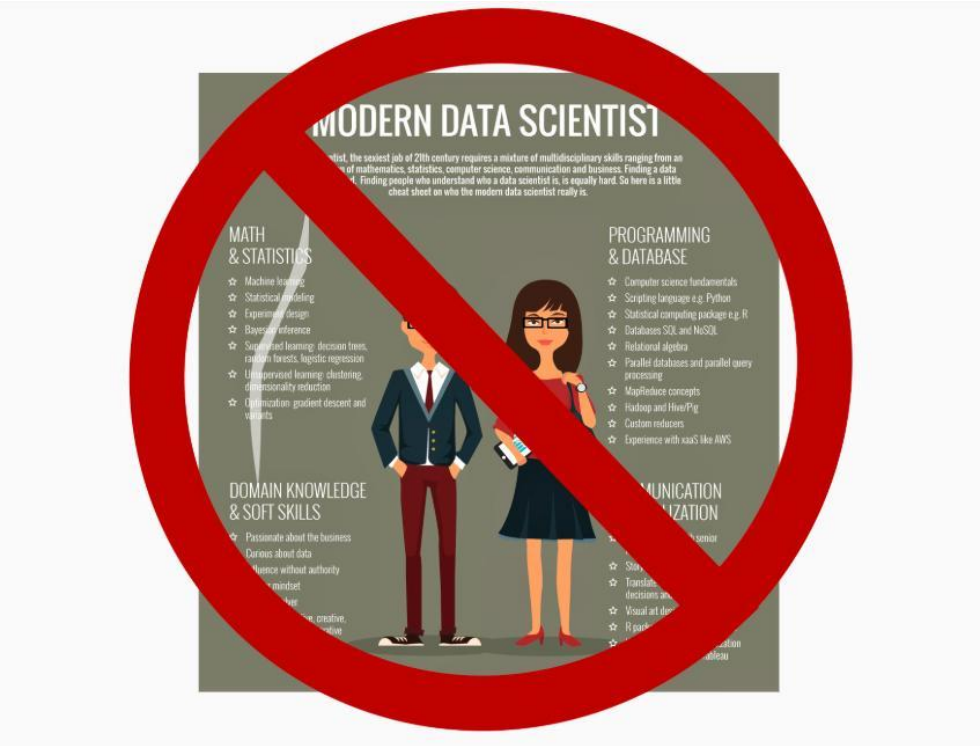
This is why we need to repel this unicorn method. We do this by following the way of the Business Scientist.

The 2 big mistakes you're making

I started my journey following the way of the unicorn. I was trying to complete every course that I thought I needed to become a data scientist. But the truth was that every additional course I took rarely got me more valuable information. Instead it further confused me, topics overlapped with things I learned previously, and it cost me more time in my journey. The end result is it took me 5-years of struggle and I wasn't much closer to my goal than Day 1 of my journey. This was heartbreaking. I felt like a failure. And, it was years later that I realized the 2 mistakes I was making.

Mistake #1: Not treating time as the enemy

In the early days, I tried learning everything. This cost me painful years of not getting anywhere. The big change was what enabled me to build a 6-figure consulting career. I began viewing time as my enemy. This focused me on only learning the essential information I needed to solve a project.



We must repel the way of the unicorn to save our most precious resource, time

In December of 2016, I pushed myself to get an early version of `tidyquant` out there in the wild where people could use it and throw stones at it. I learned only the essential skills needed to build an R package, and the minimum financial analysis tools that I was using to do a basic stock analysis.

Surprisingly, in such a short period of time, I made tidyquant, which increased my confidence, helped me move forward in my journey, and created value for others. This process taught me much more than the Data Schools taught me in 3 weeks of the time (versus 3-years of courses).

But, then I ran into my 2nd mistake.

Mistake #2: Selling skills (and not value)

If you're having a tough time in job interviews, then you'll hate consulting. Every engagement for me was like a job interview. And in the early days, I sucked! I would often fail at getting new clients. This was painful. And that pain caused me to learn. And eventually I figured out why.

The big mistake I was making early on was selling my skills, not my value. I smartened up. I learned to treat every initial client engagement as a sales pitch. I had to sell myself and my value. Mind you I wasn't a salesman, but if the value was properly presented then they'd go with me 9 times out of 10.

How did I change from selling skills to selling value? When I'd begin any consulting engagement, I'd never go in saying I know something or can do something. Rather, I'd ask what problems they were having and listen for opportunity. As soon as the client began talking, I'd uncover their problems. I knew I could solve a lot of them. I then would explain that I don't have all of the answers, but I have a high certainty that this. This showed them value. I exposed problems they didn't even know they had, and I offered the solution. ME!

Every interview was me creating a bridge from their problem to the solution they needed.

I was their bridge to value. Winning clients boiled down to being the bridge from their problem to the solution they needed. Solving the problems they now realized they had through our discovery meeting. Success was not based on my skills as a data scientist. Success was based on the idea of how I could solve a problem for them.

So here's the big mistake you're making (I was too)

Skills don't sell.

So why are we marketing them? We need to change our beliefs. Most of us feel that to get a job in data science, we need to learn data science inside and out and then market this list of skills. But, we don't.

Seek skills that lead to value.

When we adopt a new mindset, one of value over skills, we begin seeking skills that add value incrementally to the larger goal. This is a step in the right direction, one that many data scientists miss.

Sell value.

Learn how to sell your value. Here's how by following the way of the Business Scientist.

How to market yourself as a data scientist

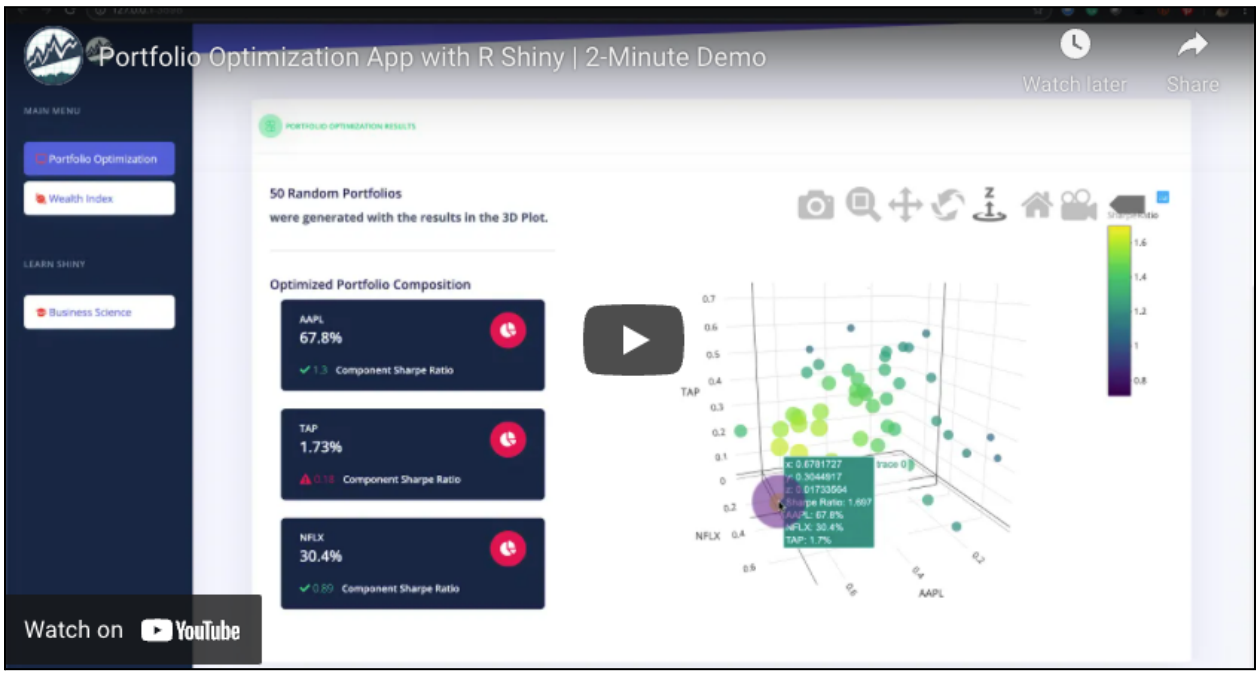
To be effective in an organization, you need to generate value for the business. You do this by:

1. Reducing Cost
2. Increasing Revenue
3. Maximizing Profit

Solving problems that address **Key Performance Indicators** is a great place to start. Anything to do with customers, quality, service, performance, and so forth.

How do you (the Data Scientist) create value?

The data scientist creates value by taking applications into production.



Demonstrating value through a web app

What is an application?

Every day we make decisions based on intuition. Decisions in the absence of data are often incorrect. When we use data to improve decision-making, value is created for the organization by reducing costs, increasing revenue, and/or maximizing profit. The application is *the thing* that non-data scientists can use to help them make better decisions.

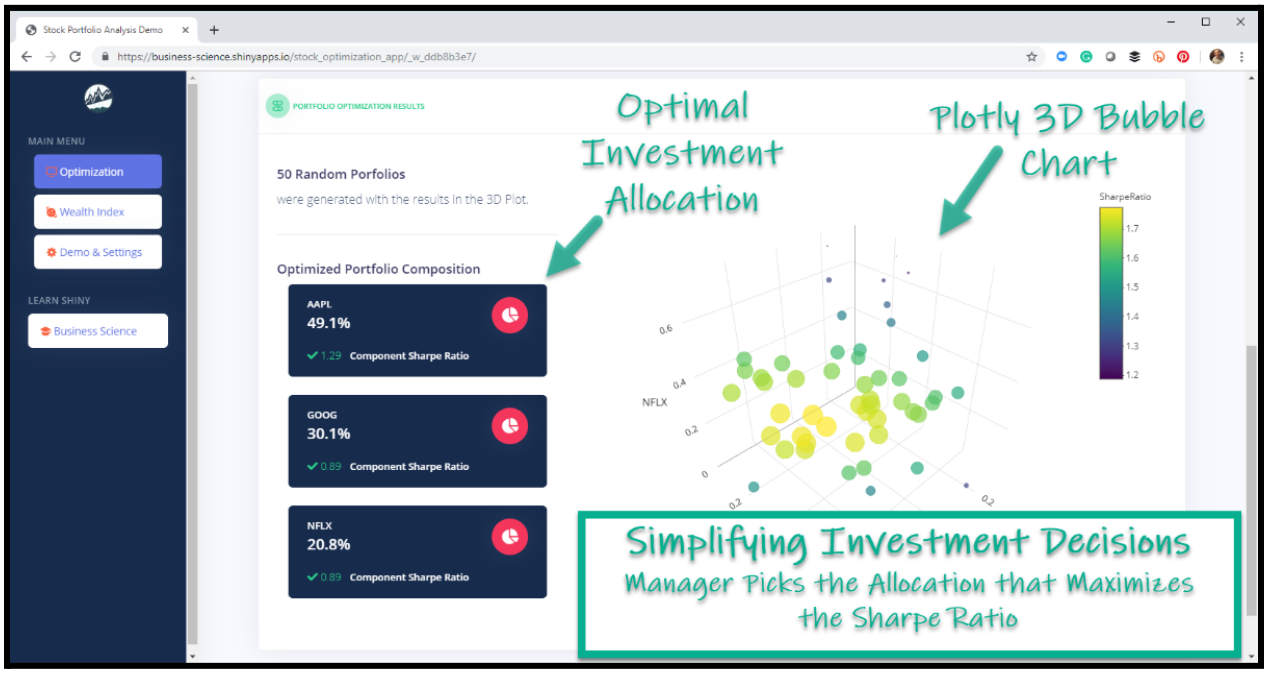
What is production?

We know that applications can help improve decision making. But applications are worthless unless people can use them. Production is the process for giving people access to your applications. Production generates massive value. In fact, applications that embed data science can save organizations $15,000,000 per year or more.

Here's a case-study of a `shiny` web application that can easily result in *multi-million-dollar-per-year savings*.

Case Study: Assisting an Asset Manager's Tactical Investment Allocation

The best way to make a difference is to understand the people you seek to help. This is the [Stock Portfolio Optimization Application] that I demonstrated at the *R/Finance 2019 Conference*. And here is a full client problem-solution statement that I would use to sell my services (yes, you can use this framework in interviews, resumes, and you should be taking notes right now).



Tactical Asset Allocation Application

Problem Statement

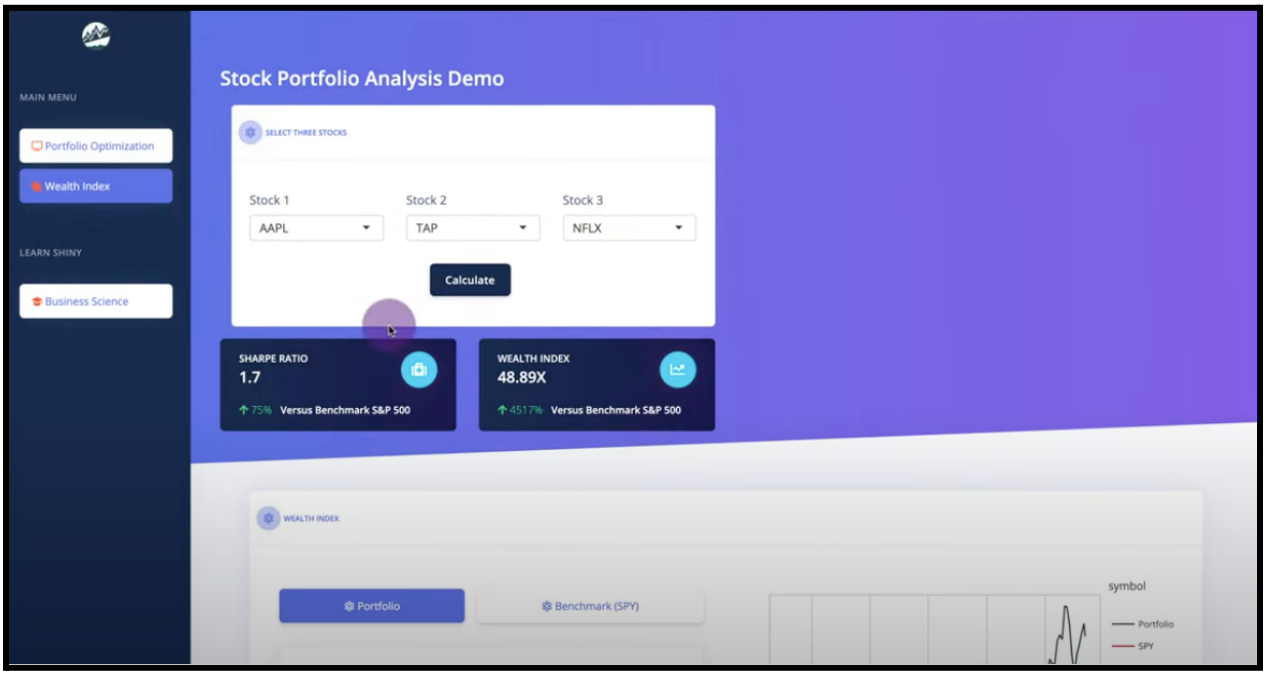
Asset Managers select stocks based on their knowledge of the company, market, and intuition of what the future holds. However, allocating investments among the basket of available stocks is ***a problem that is costly if the Asset Manager over-weights a risky stock***. A bad bet can result in lost clients, costing the organization millions in fees that would have otherwise been collected.

Solution Statement

We can use data-driven analysis to optimize the allocation of investments among the basket of stocks. Modern portfolio theory (Capital Asset Pricing Model) suggests that using the Sharpe Ratio (a metric of reward-to-risk) can reduce the riskiness of a portfolio while preserving returns. When implemented via a web application, the asset manager can select stocks (what they are good at) and then use modern portfolio theory to recommend weights that reduce risk and increase profit (what the algorithm is good at).

Proposed Web Application

The proposed web application allows the Asset Manager to focus on their job of picking stocks, while the investment allocation decision becomes automated using modern portfolio theory.



Proposed Financial Shiny Web Application

We automated the portfolio allocation process by randomly calculating portfolios and the Sharpe Ratio, and returning the tactical allocation strategy of the best portfolio.

The application helps an Asset Manager make better investment decisions that will consistently improve financial performance and thus retain clients. The web application is demonstrated here and described in a YouTube video here.

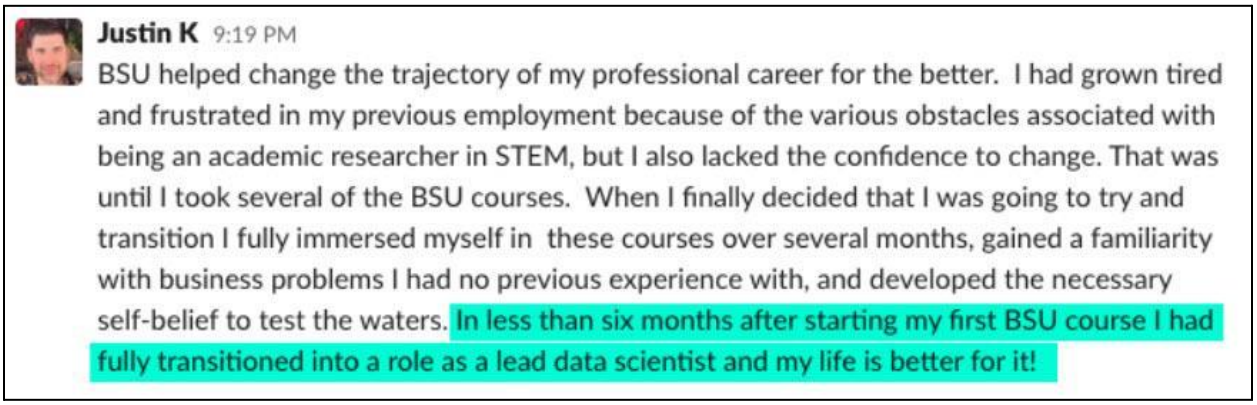
How to Become a Financial Data Scientist ($125,000 salary in 6-months)

With everything we've covered in this chapter, you now understand that businesses really want value (not skills), how you need to sell your value (not your skills), and what you can build (a web application) to demonstrate your value so they immediately see you as the bridge from their problems to the solution.

But you're probably saying, *"I don't have a plan to do this"* or *"It will take years to learn all of this stuff."* Here's a secret.

It can be done in 180-days.

Do you remember Justin (from the beginning of this chapter)? Justin was a student of mine that transitioned from academia to the Lead Data Scientist of Northwestern Mutual (Top 10 Insurance Firm). He's a financial data scientist. **And he made this transition in under 180-days**.



He told me that,

> *"In less than six months (180-days) after starting my first Business Science University course, I had fully transitioned into a role as a Lead Data Scientist [at Northwestern Mutual] and my life is better for it!"*

How did Justin do it?

Justin joined my R-Track Program and committed to the way of the Business Scientist. Justin learned the skills, frameworks, and tools to give the organization value. And he sold himself and his value in the interview to get his Lead Data Scientist position. And this had amazing benefits. It helped him and his family relocate to be closer to their extended family and get the home of their dreams. Pretty amazing, right?

It's your turn.

Imagine if in 180-days from now you were the Lead Data Scientist at Northwestern Mutual. What would this do for you and your career? How amazing would it be to finally be able to afford the dream house for your new family or take a nice vacation with your friends?

You've seen what's happened to Justin who can now afford a nice house for his family. You've seen how it helped Mohana who's the Lead Data Scientist at MoneyView, one of India's fastest growing startups. And you've heard stories from Auggie, Jennifer, Rodrigo, and Masatake who have taken control of their careers by following the way of the Business Scientist.

This is what can happen to you. But, you have to invest in yourself. And, do you honestly think there is any way you could fail with me in your corner?

...

Now you know everything you need to successfully market yourself as a data scientist. In the next chapter, we are going to dive deeper into why R is the right choice for many data scientists. And if I was learning data science over, here's why I'd learn R over Python.

Chapter 5: Why I picked R over Python for Data Science

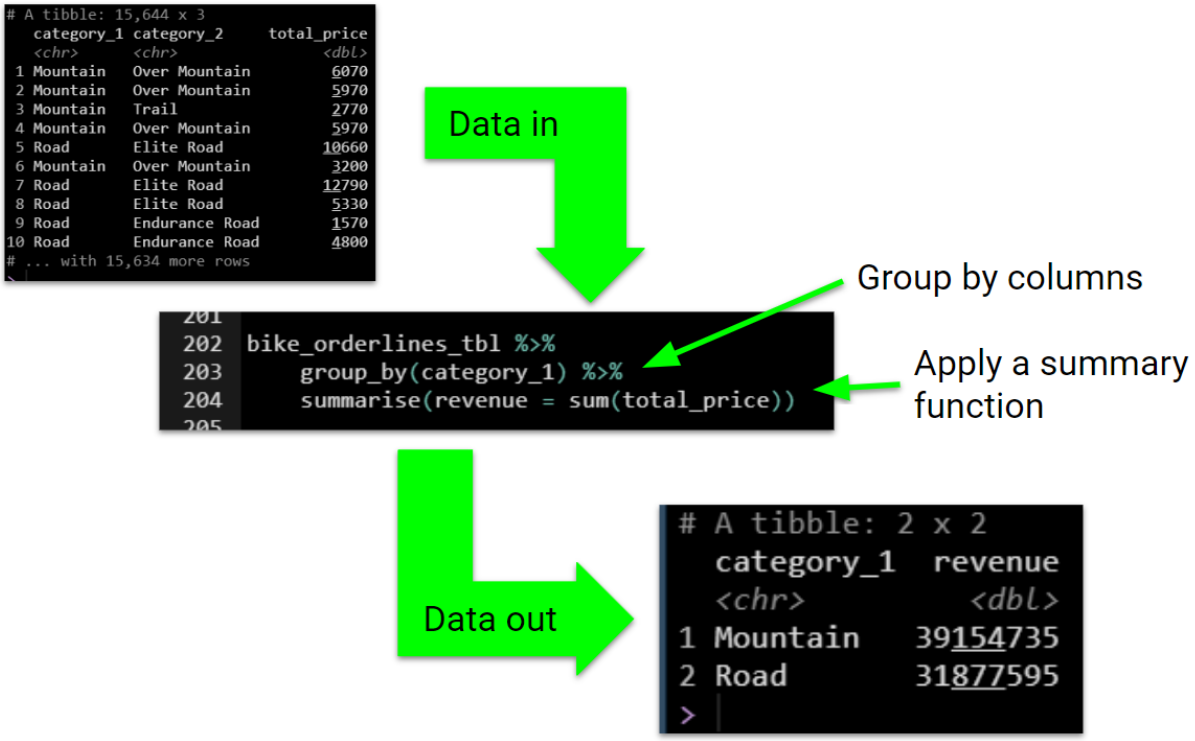
Most people don't know this, but I actually tried and failed to learn Python before I learned R. It's a true story. I began my data science learning journey in Python. I was coming from a hard-core Excel background where I had mastered the advanced Excel functionality including VBA (Excel's "builtin" programming language). So I thought I was ready for Python. Boy was I wrong.

Python was too much like coding… classes, methods, decorators. Ugh!

Python was too much like coding. And, I just didn't get it. I was 3-months into my Udemy Python Bootcamp. I was learning data wrangling. But, put an actual business dataset in front of me and I was stunned, motionless like a deer in headlights. I just simply couldn't use Python. **And after 3-months, I quit.**

It was heartbreaking. I went into a short depression because I honestly thought I'd never be good enough. A few weeks later a friend of mine recommended me to try R. Setting my ego aside, I decided to give R a try.

I started looking into R. R was a weird language, and a lot of the code used this funny looking pipe operator `%>%`. I remember wondering, *"What the hell is this %>%?"*



R's tidyverse was kind of like reading a book

As I explored R more, I found out about this cool set of tools called the tidyverse, and I loved it. The `tidyverse` code was kind of like reading a book. Each line was a "sentence" that used "Tidyverse Verbs", which were just functions that described what it was doing to the data. The pipe `%>%` connected sentences into "paragraphs". And then I could interpret what was happening just like reading a book! Pretty cool, right?

R was weird. But it clicked. And I liked it! But, then I began to experience a different problem. One that made me feel like I was a social outcast.

The lies started to flood in.

As I was learning R, I was writing about what I was learning. Things that I was surprised R could do. And, I'd post my learning on Business Science's blog and on social media.

As I would document my journey, the haters started to form. I'd read comments to my posts on social media that R couldn't do things like ***"R can't process big data"***. I found this totally wrong. I was processing 70-million rows with the tidyverse.

Then I'd hear Python people say, ***"Yeah, but I'd never put R into production."*** What?! I found the opposite to be true. R had `shiny` and `plumber`, and companies would pay me to build them applications and APIs that went into "production", which just means that they automate the process on a server so others could use it.

I'd then ask these Python people, "What's wrong with putting R into production", and they would reply ***"R can't scale"***. I'd point them to the `future` library, and ask them what they had tried to scale. That's when the holes in their arguments started to form. I'd find out they've *never* even built an application or put any code into "production" (not even Python code). They were just repeating what they've heard from others online.

The last straw was when I heard that, ***"you need to learn Python to get a data science job."*** As if R-users were some unemployable subhuman species. I'd then regularly show them companies like Apple, S&P Global, JP Morgan, and other companies that were hiring R users with fat 6-figure salaries.

These were the lies. But here's the truth.

The reality was that companies that said they were doing "data science", were really…

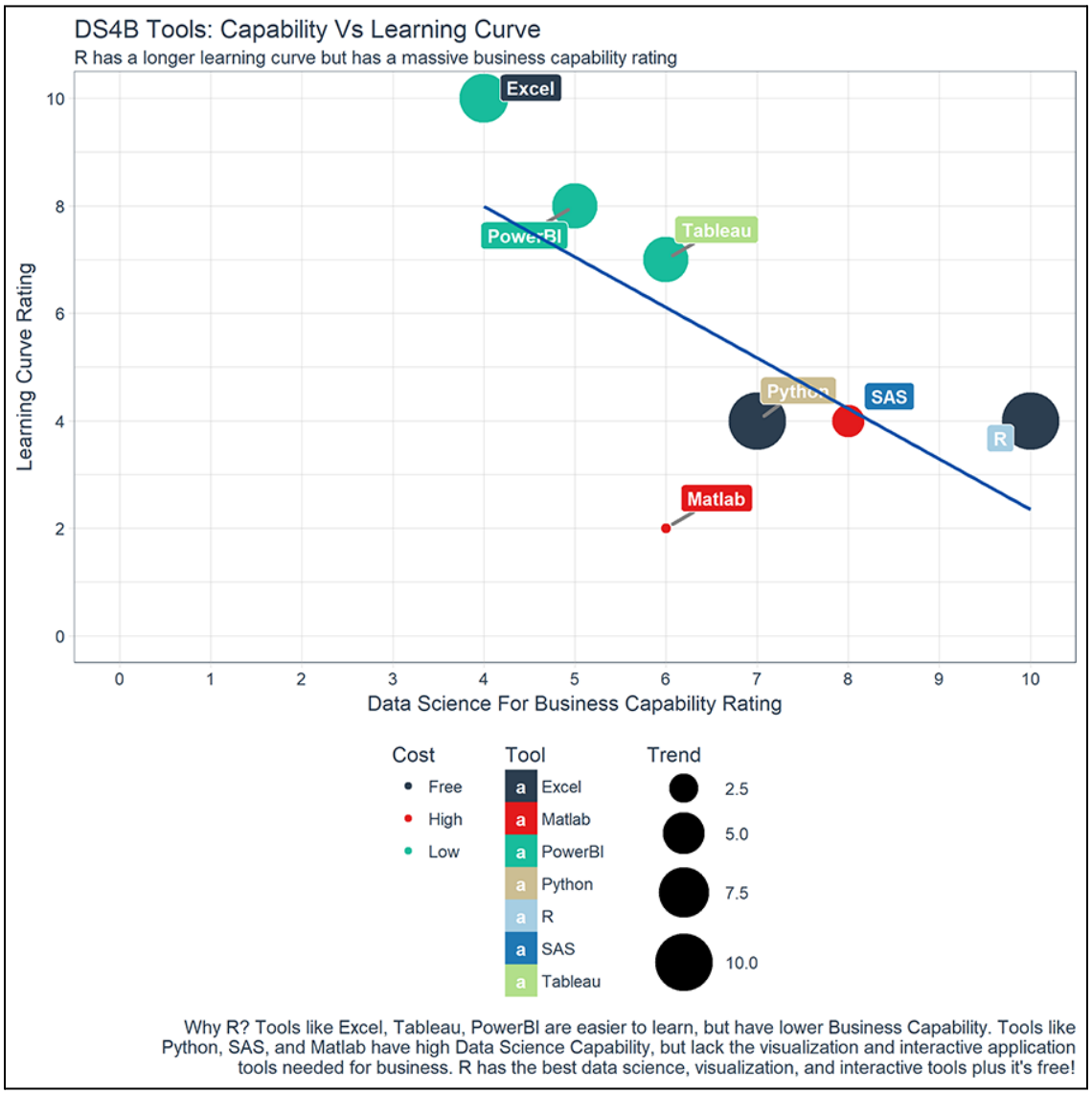
…just using Excel. And I knew I could go way beyond what Excel could do with R or Python for that matter. And, that's when it hit me.

Companies didn't care about what tool you use.

Truth is, they just want results. And that's what I was able to give them with R as my secret weapon. So that leads me to the 6 reasons that I would pick R for becoming a business data scientist if I was learning data science over again.

And this leads me to 6 of the most powerful reasons that I still would learn R to this day as my number one data science tool.

Reason 1: R Has The Best Overall Qualities For Business

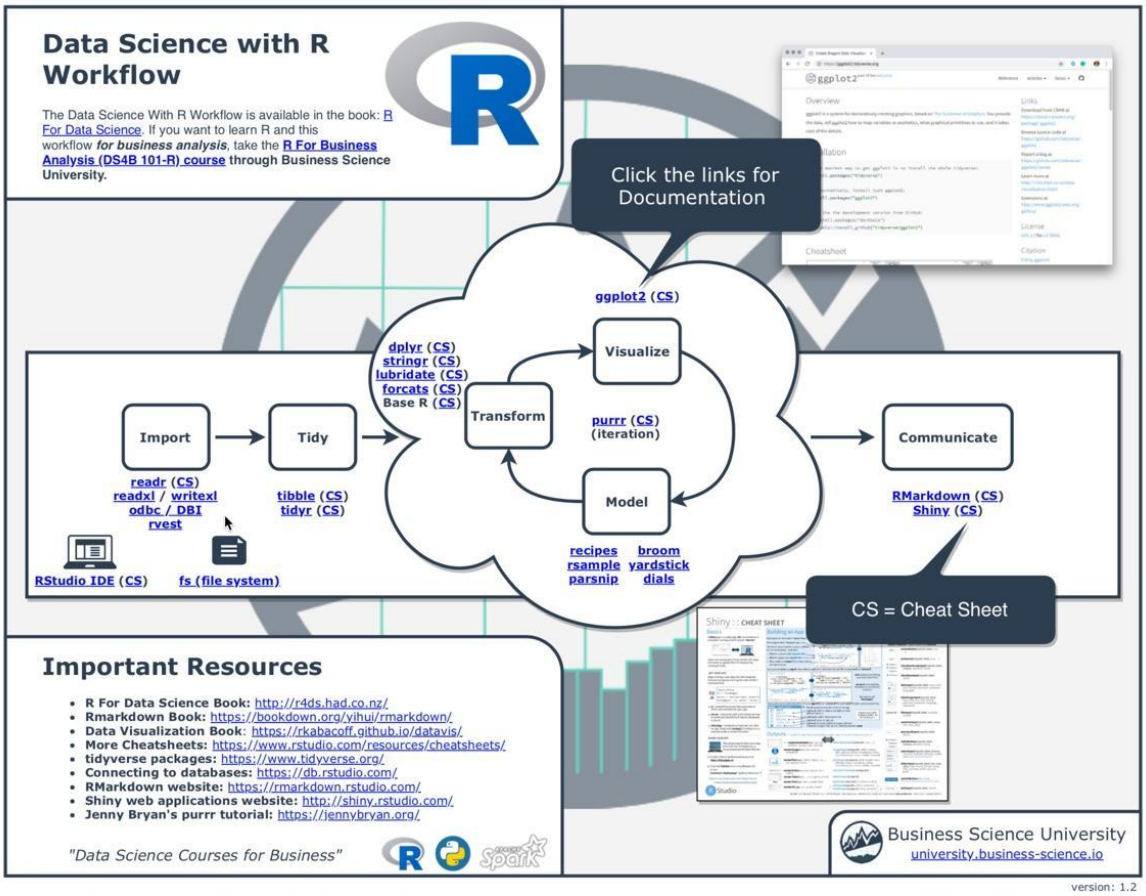


There are a number of tools available for business analysis/intelligence. Each tool has its pros and cons, many of which are important in the business context. I ranked the most common data science and business analysis tools using the following criteria:

- Business Capability (1 = Low, 10 = High)
- Ease of Learning (1 = Difficult, 10 = Easy)
- Cost (Free/Minimal, Low, High)
- Trend (0 = Fast Decline, 5 = Stable, 10 = Fast Growth)

What I saw was particularly interesting. A trendline developed exposing a tradeoff between learning curve and data science for business capability rating. The most flexible tools are more difficult to learn but tend to have higher business capability. Conversely, the “easy-to-learn” tools are often not the best long-term tools for business or data science capability. My opinion was to go for capability over ease of use.

Of the top tools in capability, R has the best mix of desirable attributes including high data science for business capability, low cost, growth, and has a massive ecosystem of powerful R libraries. The downside is the learning curve. The Cheat Sheet below showcases the powerful libraries that are at your fingertips - Download my R-Cheat Sheet to see what libraries are available to solve specific needs.



The R-Cheat Sheet showcases the massive ecosystem of powerful R packages (Free Download)

Reason 2: R Is Data Science For Non-Computer Scientists

If you are seeking high-performance data science tools, you have two options: **R or Python**. When starting out, focus on one or the other. The difference between R and Python has been described in numerous infographics and debates online, but the most overlooked reason is person-programming language fit. You might miss this, so let's break it down further.

Fact 1: Most people interested in learning data science for business are not computer scientists.

They are business professionals, non-software engineers (e.g. mechanical, chemical), and other technical-to-business converts. This is important because of where each language excels.

Fact 2: Most activities in business and finance involve communication.

This comes in the form of reports, dashboards, and interactive web applications that allow decision makers to recognize trends and to make well-informed decisions.

About Python

Python is a general service programming language developed by software engineers that has solid programming libraries for math, statistics and machine learning. Python has best-in-class tools for pure machine learning and deep learning, but lacks much of the infrastructure for subjects like econometrics and communication tools such as reporting. Because of this, Python is well-suited for computer scientists and software engineers.

About R

R is a statistical programming language developed by scientists that has open source libraries for statistics, machine learning, and data science. R lends itself well to business because of its depth of topic-specific packages and communication infrastructure.It has packages covering a wide range of topics such as econometrics, finance, and time series. R has best-in-class tools for visualization, reporting, and interactivity, which are as important to business as they are to science. Because of this, R is well-suited for scientists, engineers and business professionals.

Which Should You Learn?

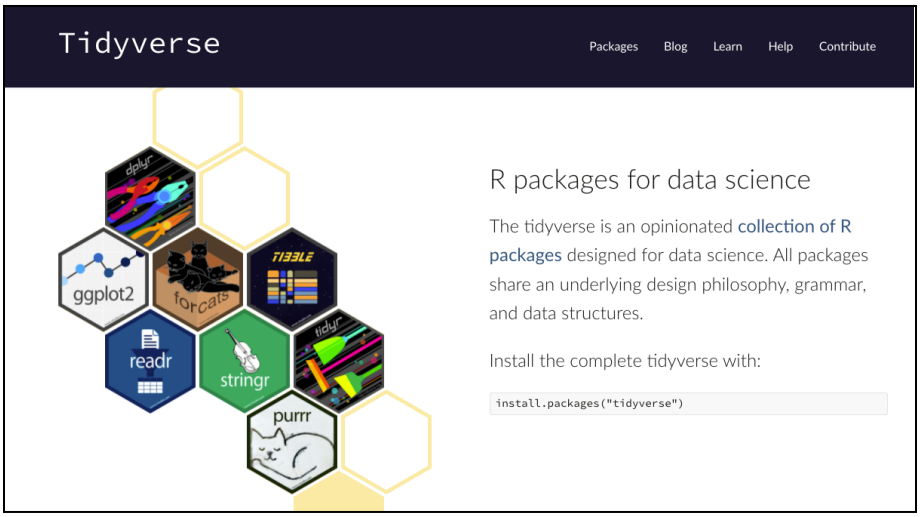
Don't make the decision tougher than what it is. Think about where you are coming from. Are you a computer scientist or software engineer that values coding over statistical analysis? If yes, learn Python. Are you an analytics professional or mechanical/industrial/chemical engineer looking to get into data science? If yes, learn R.

Next, think about what you are trying to do. Are you trying to build a self-driving car? If yes, learn Python. Are you trying to communicate business analytics throughout your organization? If yes, learn R.

Reason 3: Learning R Is Easy With The Tidyverse

Base R, is what existed 20 years ago, but the R language has since evolved with the tidyverse (it's our readable version that uses the pipe %>%). Unfortunately, Base R is a complex and inconsistent programming language that a lot of people struggled with. Those people then went to Python. And now when they talk about R, it's usually based on their experience with Base R. And I get that. But R has changed for the better.

In Base R, structure and formality was not the top priority, but this all changed with the `tidyverse`, a set of packages and tools that have a consistently structured programming interface for data manipulation, visualization, iteration, modeling, and communication.



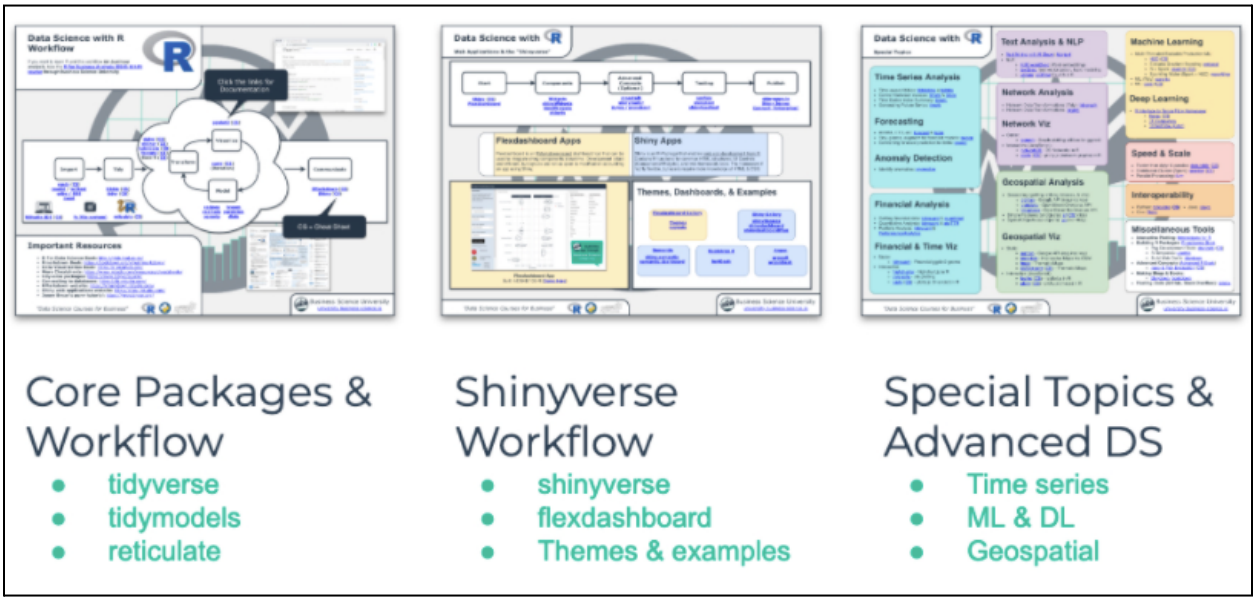
Source: tidyverse.org

Two of the core tidyverse packages, `dplyr` and `ggplot2`, changed the game, dramatically reducing the learning curve by providing a consistent and structured approach to working with data. As Hadley Wickham and many others continued to evolve R, the `tidyverse` came to be. As a result, R is now much easier to learn.

R continues to evolve in a structured manner, with advanced packages that are built on top of the `tidyverse` infrastructure. A new focus is on modeling and algorithms. Further, the `tidyverse` is extending to cover topical areas such as text (`tidytext`) and finance (`tidyquant`). For newcomers, this should give you confidence in selecting this language. R has a bright future.

Reason 4: R Has Brains, Muscle, And Heart

Saying R is powerful is actually an understatement. From the business context, R is like Excel on steroids! But more important than just muscle is the combination of what R offers: brains, muscle, and heart. The 3rd page of the R Cheat Sheet links to all of the tools discussed next (and more tools beyond)!



The Ultimate R Cheat Sheet (Free Download)

R has Brains.

These tools are used everywhere from AI products to Kaggle Competitions, and you can use them in your business analyses. R implements cutting-edge algorithms including:

- H2O (`h2o`) - High-end machine learning package
- Keras/TensorFlow (`keras`, `tensorflow`) - Go-to deep learning packages
- xgboost - Top Kaggle algorithm
- Modeltime - Time Series forecasting
- And many more!

R has Muscle.

R's constantly dogged for being a slow language, but it's not. R has powerful tools for:

- Vectorized Operations. R uses vectorized operations to make math computations lightning fast.
- Loops and iteration (`purrr`)
- Parallel and asynchronous operations (`future`)
- Speeding up code using C++ (`Rcpp`)
- Connecting to python (interoperability) (`reticulate`)
- Working With Databases (`dbplyr`, `odbc`, `bigrquery`)
- Handling Big Data (`sparklyr`, `data.table`, `dtplyr`, and `disk.frame`)
- And many more!

R has Heart.

We already talked about the infrastructure, the `tidyverse`, that enables the ecosystem of applications to be built using a consistent approach. It's this infrastructure that brings life into your data analysis. The `tidyverse` enables:

- Data manipulation (`dplyr`, `tidyr`)
- Working with data types (`stringr` for strings, `lubridate` for date/datetime, `forcats` for categorical/factors)
- Visualization (`ggplot2`, `plotly`)

- Programming (`purrr`, `rlangl`)
- Communication (`Rmarkdown`, `shiny`)

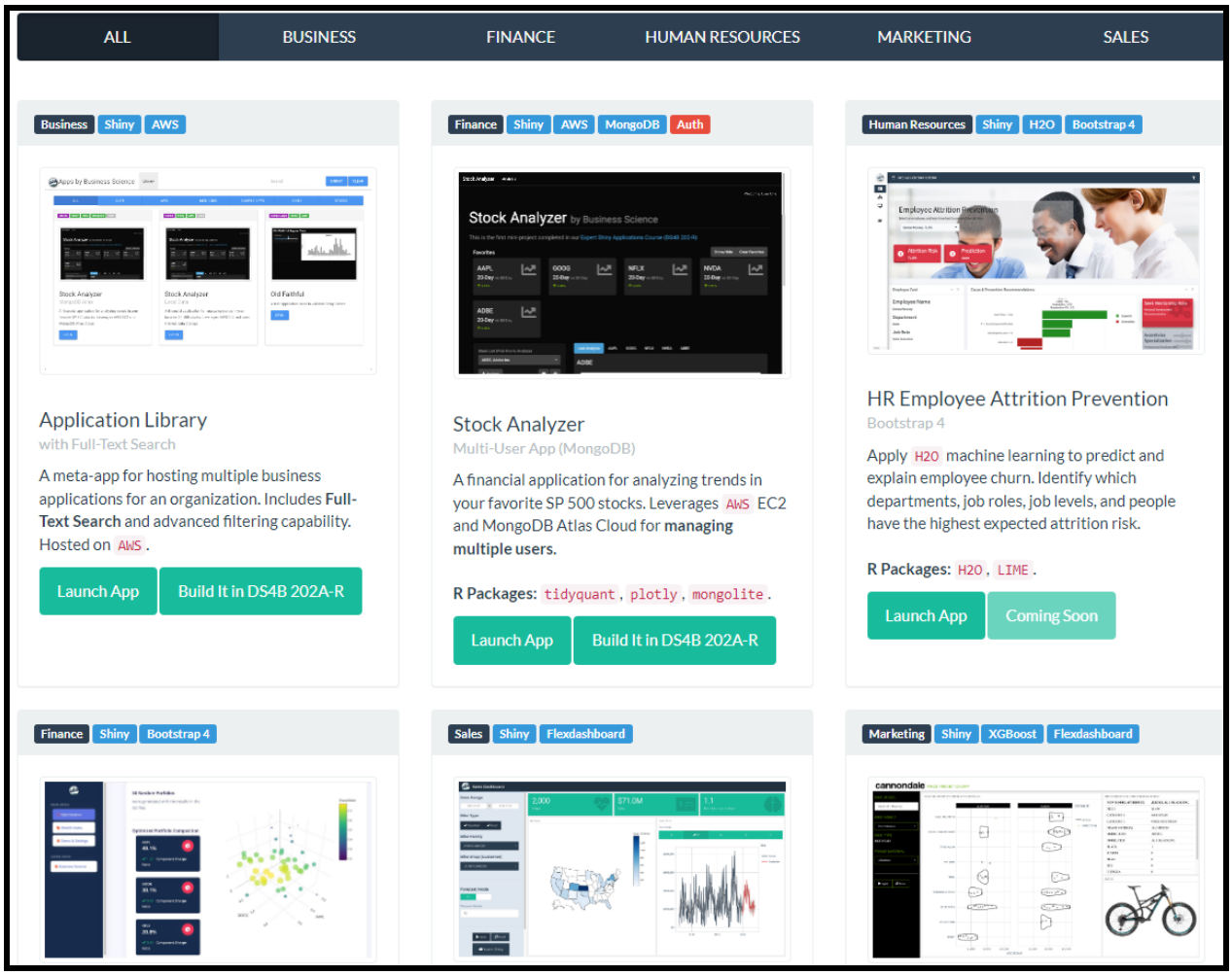
Reason 5: R Is Built For Business

The major advantages of learning R versus another programming language is that it can produce business-ready reports and machine learning-powered web applications. Neither Python or Tableau or any other tool can currently do this as efficiently as R. The two capabilities are `rmarkdown` for report generation and `shiny` for interactive web applications.

Rmarkdown is a framework for creating reproducible reports that has since been extended to building blogs, presentations, websites, books, journals, and more. I use it to include code with text so that anyone can follow the analysis and see the output right with the explanation. What's really cool is that the technology has evolved so much. Here are a few examples of its capability:

- rmarkdown for generating HTML, Word and PDF reports
- rmarkdown for generating presentations
- flexdashboard for creating web apps via the user-friendly Rmarkdown format.
- blogdown for building blogs and websites
- bookdown for creating online books
- Interactive documents
- Parameterized reports for generating custom reports (e.g. reports for a specific geographic segment, department, or segment of time)

Shiny is a framework for creating interactive web applications that are powered by R. It is a major consulting area; four of five assignments involve building a web application using `shiny`. It's not only powerful, it enables non-data scientists to gain the benefit of data science via interactive decision making tools.



Examples of a Shiny Apps

https://apps.business-science.io/

Reason 6: Community Support

You begin learning R because of its capability, you stay with R because of its community. The R Community is tight-knit, opinionated, fun, and highly knowledgeable. All of the things you want in a high-performing team.

CRAN: Community-Provided R Packages. CRAN is like the Apple App store, except everything is free, super useful, and built for R. With over 20,000 packages, it has everything from machine learning to high-performance computing to finance and econometrics! The task views cover specific areas and are one way to explore R's offerings. CRAN is community-driven, with top open source authors such as Hadley Wickham and Dirk Eddelbuettel leading the way. Package development is a great way to contribute to the community especially for those looking to showcase their coding skills and give back!

Social/Web: R users can be found all over the web. A few of the popular hangouts are:

- R-Bloggers
- #rstats on Twitter
- The R Project for Statistical Computing group on LinkedIn

Conferences: R-focused business conferences are gaining traction in a big way. A full list of R conferences can be found here. Here are a few that I attend.

- Rstudio Conf - Rstudio's technology conference.
- New York R - Business and technology-focused R conference.
- R/Finance - Community-hosted conference on financial asset and portfolio analytics and applied finance.

Meetups: A full list of R user groups can be found here. A really cool thing about R is that many major cities have a meetup nearby. Meetups are exactly what you think: a group of R-users getting together to talk R. They are usually funded by R-Consortium. You can get a full list of meetups here.

Now you know why I chose R over Python.

I didn't listen to the naysayers. I did my research and uncovered some amazing things about R that Python simply does not have (like the tidyverse, shiny apps, and an amazing R community). And, in the end learning R first helped me become a better data scientist much faster.

Now, I will say that I eventually did learn Python. But without learning R first, I would not have been nearly as successful. Once I knew R, learning Python became an extension. I knew data science with R. I just needed to add the coding-side of Python (object oriented programming, methods, etc), and that was a much easier jump than learning data science **AND** coding at the same time.

So for many of you, if you are struggling with learning Python, then give R a try. You're costing yourself time the longer you struggle. And, R won't bite. It was so much easier for me. Wouldn't you like to save time?

...

Alright, let's quickly recap. So far, you learned the 14 skills you can learn to become a data scientist, the Business Science Problem Framework (the secret framework I used to complete 90% of business projects for companies), the data science career path (plus the advanced skills) needed to grow your career beyond the entry-level data scientist position, and even why I picked R over Python in my journey. But, we haven't talked about what you're getting yourself into when you join a data science team yet. Let's learn about your part in the data science team.

Chapter 6: Anatomy of a Data Science Team (Case Study)

Written by Matt Dancho and Rafael Nicolas Fermin Cota

OneSixtyTwo Digital Capital has spent the last five years building a high performance data science team. This chapter captures my detailed notes of how their team was structured for high-performance. I wrote down everything I possibly could during my week-long visit at their headquarters in Toronto, Canada.

OneSixtyTwo Digital Capital

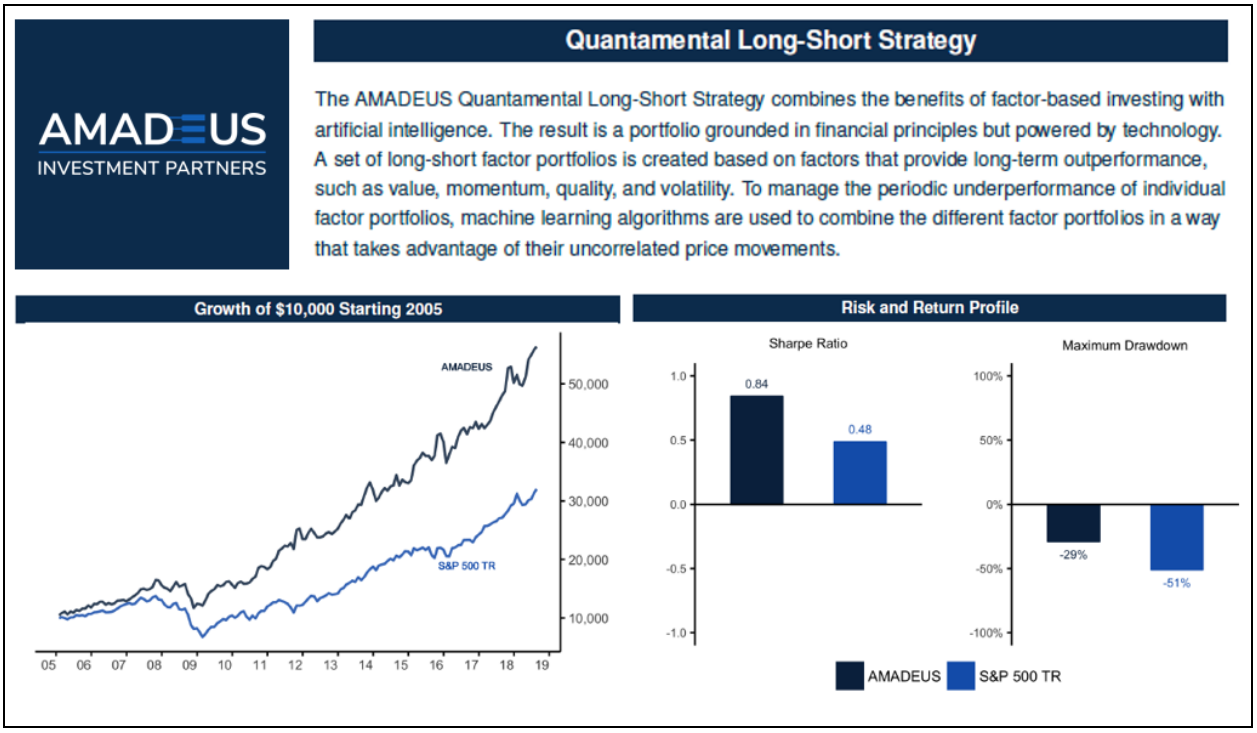
In August of 2018 I was invited inside the walls of OneSixtyTwo Digital Capital (previously Amadeus Investment Partners), a hedge fund that has achieved superior results in one of the most competitive industries in the world: stock investing.

My visit to OneSixtyTwo Digital Capital (Previously Amadeus Investment Partners)

Examining An Outlier

OneSixtyTwo Digital Capital is a hedge fund that blends traditional fundamental investment principles with cutting-edge quantitative techniques to create “Quantamental” strategies that identify assets that yield excellent returns while minimizing risk for their investors. Their goal is to provide their investors with superior risk-adjusted returns.

OneSixtyTwo’s strategy is working. Here’s an overview of backtest results from 2005 in comparison to the S&P 500, which is a difficult benchmark to outperform. Over the backtest period, we can see that OneSixtyTwo Digital Capital’ strategy delivered “alpha”, which means the strategy generated excess returns (performance) beyond the returns of the benchmark.



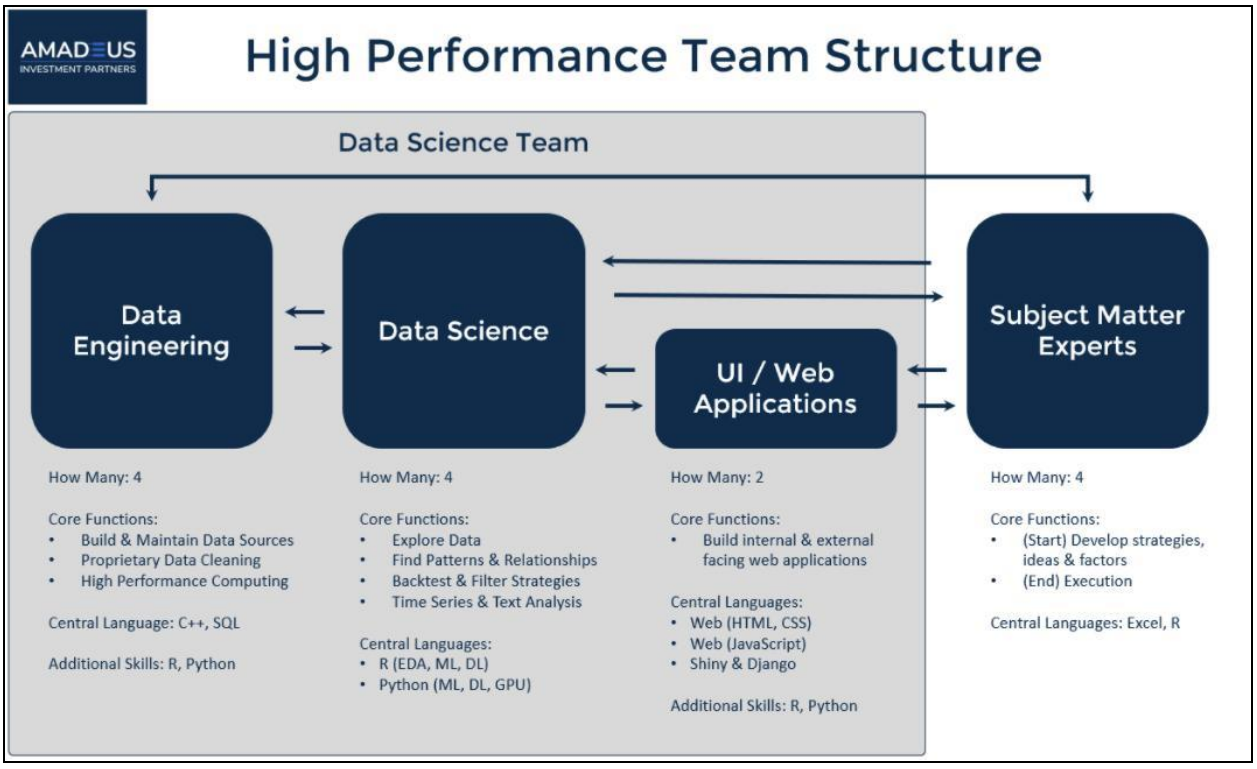
Risk-Return Performance, OneSixtyTwo (formerly Amadeus) Quantamental Long-Short Strategy

From the *Growth of $10,000 Starting 2005* chart, *OneSixtyTwo Digital Capital* appears to be a well-performing hedge fund. However, it's not until we dive into the *Risk and Return Profile*, that we begin to see the magic come to light.

The *Sharpe Ratio*, which is a ratio of reward-to-risk that is commonly referenced in investing, is almost double the S&P 500 over this time period. This means that OneSixtyTwo Digital Capital is taking less risk per unit of reward as compared to the S&P 500. Furthermore, the *Maximum Drawdown*, or the largest loss from the peak during the time frame, was about half of the S&P 500 during the same time period. Ultimately, what this means is that OneSixtyTwo Digital Capital is delivering exceptional returns while taking on less risk, which is very attractive to investors.

The 3 Key Ingredients to Forming a High-Performance Team

During the week I spent with OneSixtyTwo Digital Capital, 3 key components stood out as special ingredients to the data science team's performance. Each of these are important to successful execution of their data-driven strategy.



Summary of their high-performance data science team structure

Key 1: Finding and Training Talent in an Unlikely Fashion

The first key to the puzzle is finding and developing the talent to execute on the vision. That's where OneSixtyTwo Digital Capital has excelled. Over the past several years, OneSixtyTwo Digital Capital has tactically been working with the leading educational institutions in Canada to selectively gain access to top students from Business Programs: If you look at the demographics of their team, most don't have math or physics backgrounds. If you're familiar with the conventional data science team makeup full of math and computer science Ph.D.'s, this might come as a surprise to you.

This unusual hiring practice is founded on the belief that the **subject knowledge and the communication skills that the top business students bring** are critical advantages in OneSixtyTwo Digital Capital's data-driven approach. At the end of the day, data science is a tool that people use to answer questions that they're interested in, and hiring people with the relevant subject matter expertise will ensure that the right questions will be

asked. OneSixtyTwo Digital Capital subsequently converts these business-minded people to data scientists by augmenting their skillset with math and programming on the job.

In terms of training new hires, OneSixtyTwo Digital Capital has a distinct advantage. One of the founders, Rafael Nicolas Fermin Cota, was a professor at the Ivey Business School at Western University, one of the top schools for business in Canada. In his curriculum, he taught his students how to make business decisions using data science. He states,

> *"My work entails **teaching students how to think**. The specific course material, they may forget. But, if they learn to think, they will learn to solve the problems they face in their professional careers."*
>
> - Rafael Nicolas Fermin Cota

Each team member told stories of their start at OneSixtyTwo Digital Capital. It begins the same - learning to code, studying statistics, and getting a great deal of mentorship. It takes six months of education and training before a new employee is ready to be an integral part of the team. The core curriculum includes the following concepts:

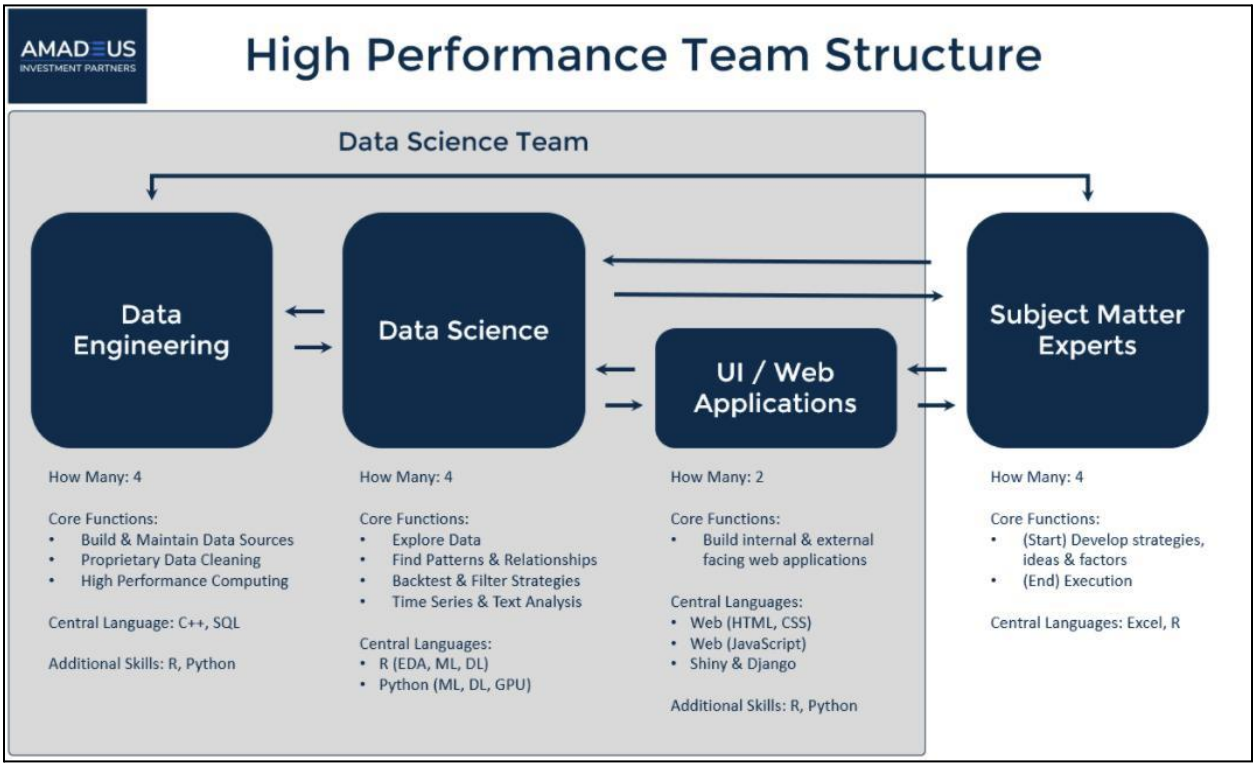
1. **Database management**: Obtaining data from various sources and storing it effectively for further access.
2. **Data manipulation**: Working with raw data (often in many different formats) and turning them into an organized dataset that can be easily analyzed.
3. **Exploratory data analysis**: Exploring the data to determine various characteristics of the dataset (NAs, mean, standard deviation, type, etc.).
4. **Predictive Modeling**: Using available data to predict the future outcome using machine learning and other artificial intelligence concepts.
5. **Visualization**: Presenting the results of the exploratory data analysis and predictive modeling to various audiences.

This core training ensures a common body of knowledge that team members draw from during discussions, making the communication process much more efficient.

Key 2: Well-Designed Team Structure and Collaborative Culture

Once the initial training is over, each new hire is ready to be integrated into a functional part of the team. Integration involves finding the role that best suits their skill sets along with OneSixtyTwo Digital Capital' needs. This approach allows the new hire to fill a position they are interested in while benefiting the organization.

The team structure was carefully designed to optimize the talent of the team members and to transparently reflect the desired interaction among the team members. Think of the High Performance Team Structure like the blueprint for success.



Data Science Team Structure, Designed for High Performance

It involves four key roles: (1) Subject Matter Experts (SMEs), (2) Data Engineers, (3) Data Scientists, and (4) UI / Web App Developers.

Role #1: Subject Matter Experts (SME)

OneSixtyTwo Digital Capital has four SMEs that are involved at both the beginning and end of the investment strategy development process. At the beginning of the process, the SMEs are responsible for generating initial ideas for new strategies. These ideas are grounded on business fundamentals and meticulously researched before being discussed with the Data Engineering and the Data Science teams. The SMEs are also responsible for the end of the process, which is the execution of the strategies. This ensures that the investment execution is in line with the original design of the strategies.

Relevant Skill-Sets:

- **Accounting and Finance:** Deep understanding of financial analysis and capital markets is required to build initial strategy ideas
- **Excel:** Excel is used to store initial strategy ideas
- **R:** `R` is used to perform data exploration and efficiently work with data

Role #2: Data Engineers

When the SMEs come up with new strategy ideas, the Data Engineering team is subsequently called to gather and make available the data required for the Data Science team to test the ideas. With petabytes of financial data at hand, the DEEs need to master programming methods that will make data delivery and computation as efficient as possible. Also, OneSixtyTwo Digital Capital has focused on data quality since further analysis is only meaningful given good quality data. The financial data is often noisy, contains many missing values, and requires timestamp joins, which is very difficult due to the size of the data and the fact that global data sources rarely align.

Relevant Skill-Sets:

- **C++**: `C++` is a high performance language at the heart of their data engineering operation. Parallelizing computations and developing distributed systems using `C++` enables OneSixtyTwo Digital Capital to take full advantage of working with big data
- **SQL**: `SQL` is the language used to directly interact with their databases
- **R**: The `data.table` package is mainly used to scale `R` for speed when taking strategies from the exploration to production

Role #3: Data Scientists

The DSEs at *OneSixtyTwo Digital Capital* are critical for exploring various properties of ideas generated by the SMEs and developing different algorithms required by the strategy, based on their expertise in statistical analysis, machine learning (supervised and unsupervised), time series analysis, and text analysis. The main challenge they face is being able to iterate through the stream of hypotheses generated by the SMEs and rapidly develop analyses. They are the ones who identify patterns or anomalies in the dataset, produce concise reports for the SMEs to allow fast interpretation of results, and determine when the ROI from a project has diminished and new projects should be started.

Relevant Skill-Sets:

- **R**: `R` is used for exploratory data analysis (EDA) and visualization because of its ease of use for exploration. The `tidyverse` is predominantly being used for quickly transforming data prior to exploration.
- **Python**: `Python` is used for advanced machine learning and deep learning with high-performance NVIDIA GPUs. All the top deep learning frameworks are available in Python and can be easily deployed through the tools provided in the NVIDIA GPU Cloud.

Role #4. Web App Developers

OneSixtyTwo Digital Capital develops interactive web applications to support internal decision-making and operations. New challenges present themselves when building dashboards. The application needs to be customized to the problem but also interact. Given these constraints, building a performant application often comes down to selecting the right tools. The UIEs use `R` + `Shiny` for lightweight applications
or `Python`, `Django` and `JavaScript` when performance and interactivity are major concerns.

Relevant Skill-Sets:

- **Databases**: Data-driven web applications start at the database. Knowledge of the appropriate query language (`SQL`, `MongoDB`, etc.) is necessary for effectively handling data.

- **Data Analysis**: `R` + `Shiny` can be used for a quick proof of concept, while `Python` + `Django` are used for production level performance.
- **Web Development**: `HTML`, `CSS`, `JavaScript` are a necessity when creating sophisticated web-based user interfaces.

The Importance of Teamwork & Communication

An often overlooked part of a data science team is the importance of operating as a team. This requires communicating ideas and analyses through the workflow. For many organizations, various departments work in silos, only interacting with each other at the senior management level. This prevents members from seeing the big picture and breeds internal competition.

At *OneSixtyTwo Digital Capital*, collaborative culture is encouraged as every project is carried out by a cross-sectional team, involving at least one person from each of the four functions described above. This way, the projects benefit from the different perspectives of team members and the research process is streamlined without conflicts between each stage.

Key 3: Access to Cutting-Edge Technology

It takes tremendous effort to find and train talent. This effort would be futile if there was a technological bottleneck in the research process.

Access to cutting edge technology (e.g. NVIDIA GPUs) is important to success

Data Science Team members have full access to computational infrastructure for both GPU intensive work (DL, NLP), and CPU intensive work (data cleaning, report generation, EDA). Their systems provide immediate access to high-performance computational resources, minimizing the time spent waiting for computations to run allowing quicker iteration through ideas.

Each team has their own computation stack so as to not interfere with the work of the other teams. This infrastructure is all connected to allow interaction between teams.

- **Data Engineering**: Systems optimized for populating and querying databases. The DEEs provide a custom API that allows all other teams immediate access to data.
- **Data Science**: High-performance CPU and GPU systems ideal for training machine learning models and performing EDA.
- **UI/Web Applications**: Systems designed specifically for hosting web applications and in-house Shiny/Django applications. The Web App Developers can use the Data Scientists' infrastructure when high-performance computations are required in the backend.
- **Subject Matter Experts**: Access to data and high-performance hardware through front-end APIs as well as hardware specifically designed for their execution needs.

OneSixtyTwo Digital Capital has partnered with *NVIDIA*, pioneers of the next generation of computational hardware for Artificial Intelligence research and deployment. The team is actively using high-performance computing with their in-house analytical technology stack that boasts the NVIDIA DGX-1, the world's fastest deep learning system. The NVIDIA DGX-1 produced results in a matter of minutes, which would have taken several hours, if not days on a CPU system or even a GPU system that is not optimized for deep learning.

...

Now you know everything you need to know about where you will fit when you are working in a high-performance team for your future company. But, up until now we haven't really talked about the process to do data science inside of the data science team. Next, I'm going to show you something I developed called The Data Science Workflow that is a 3-phase framework I developed to solve ALL data problems specifically for data science teams.

Chapter 7: The Data Science Workflow Framework
The 3-Stage Process for Solving ALL Data Problems

It was 2018, and I was creating a the most important training I've ever created. It was for my new client, a Fortune 500 company called S&P Global. They had a group of 50 data scientists and data analysts that were to be on the training. A lot was riding on this training going well because without them, Business Science could very well collapse. This training ment survival.

I was working with my point of contact, Moody, a bright data scientist that was doing well evangelizing data science at S&P. We were trying to decide what to train them on to help them apply data science to their Market Intelligence solutions like Capital IQ (CapIQ), a cool software they offer to their clients that provides access to financial data and research intelligence.

S&P Global Market Intelligence's CapIQ Software

S&P Global wanted their Market Intelligence team trained on Machine Learning so they could provide key insights to their customers from the CapIQ software. But there was a big problem with training them on Machine Learning alone.

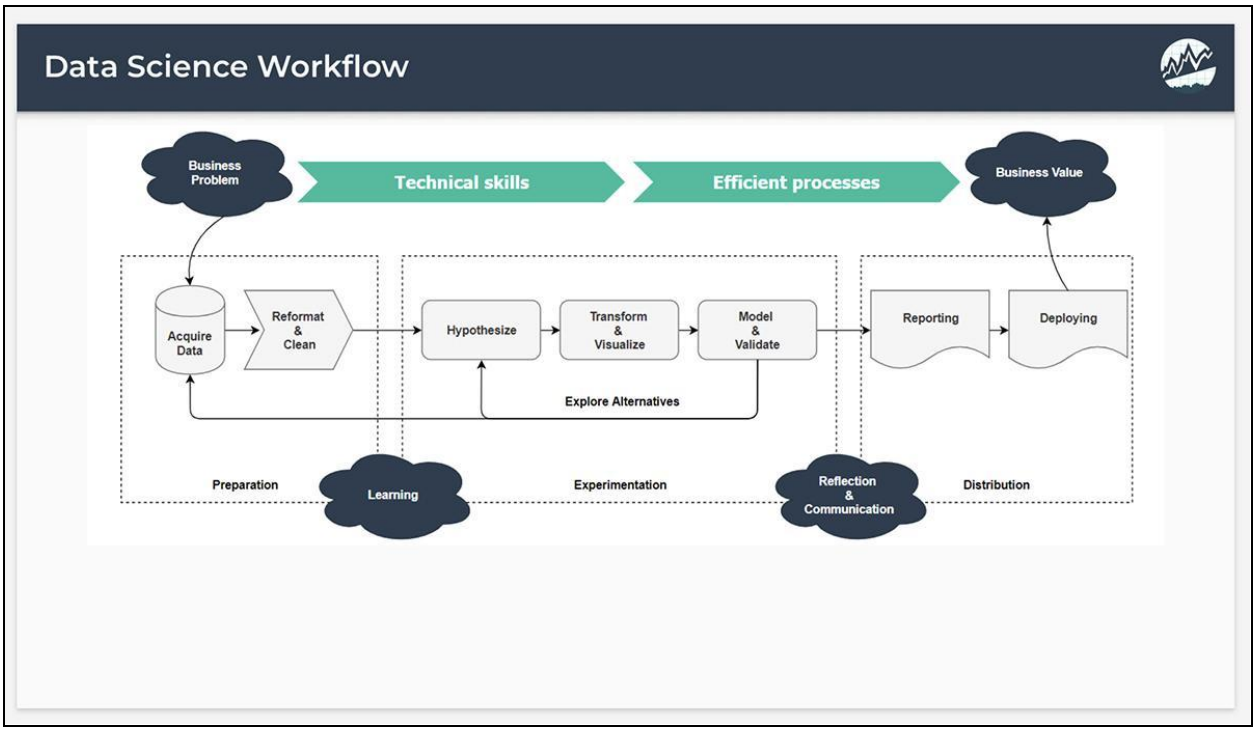
Garbage In = Garbage Out.

It wouldn't make any sense to just train them on Machine Learning in isolation (yet this is what many trainers do). The problem is, if they didn't have the right data formatted in the right way, then the machine learning model they make would not produce anything useful. And nor would they be able to apply it in their job.

Further, as we began talking more, S&P Global told me that the model wasn't really what was needed. They actually needed data products, which were just the reports and web applications that empowered S&P Global's CapIQ customers to take action.

So if we just stopped at modeling, S&P Global's team wouldn't really be able to help their customers either. We needed to do more.

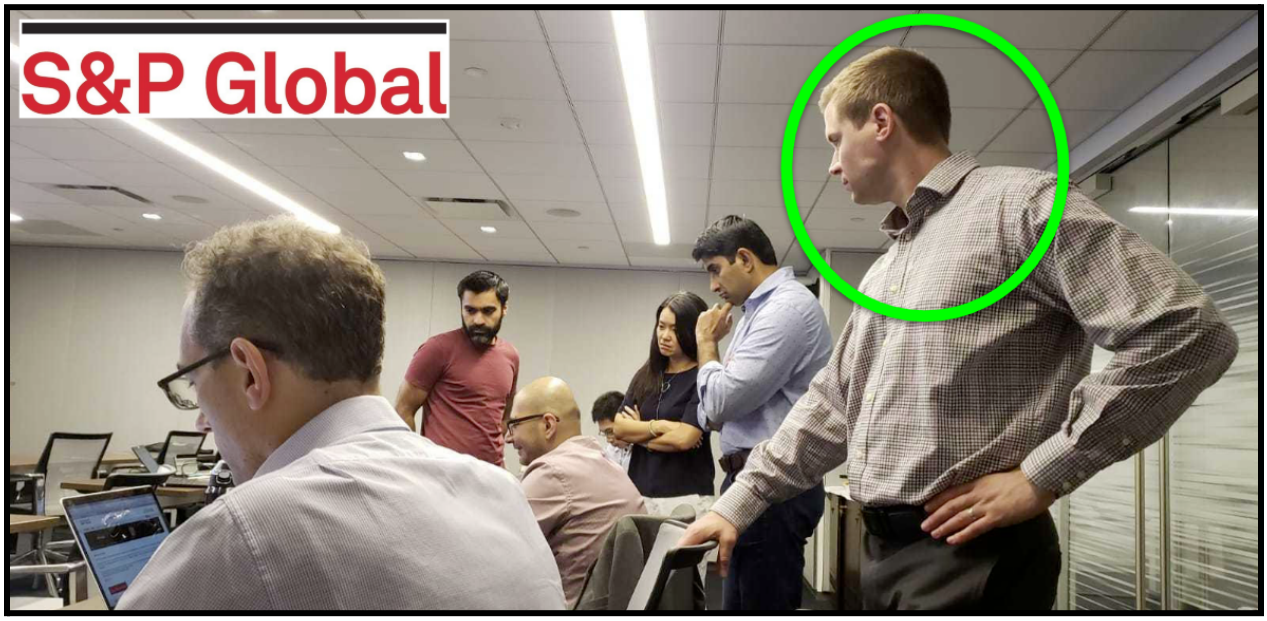
And that's when it hit me. I needed to show them my process that I use for solving ALL data problems. I began writing it down. And when I finished, I called it the Data Science Workflow.



The Data Science Workflow

The training was a massive success. The students were able to learn and apply data science using a repeatable process that would serve as the basis for future product improvements. And I was able to lead them to new heights with a streamlined framework for success.

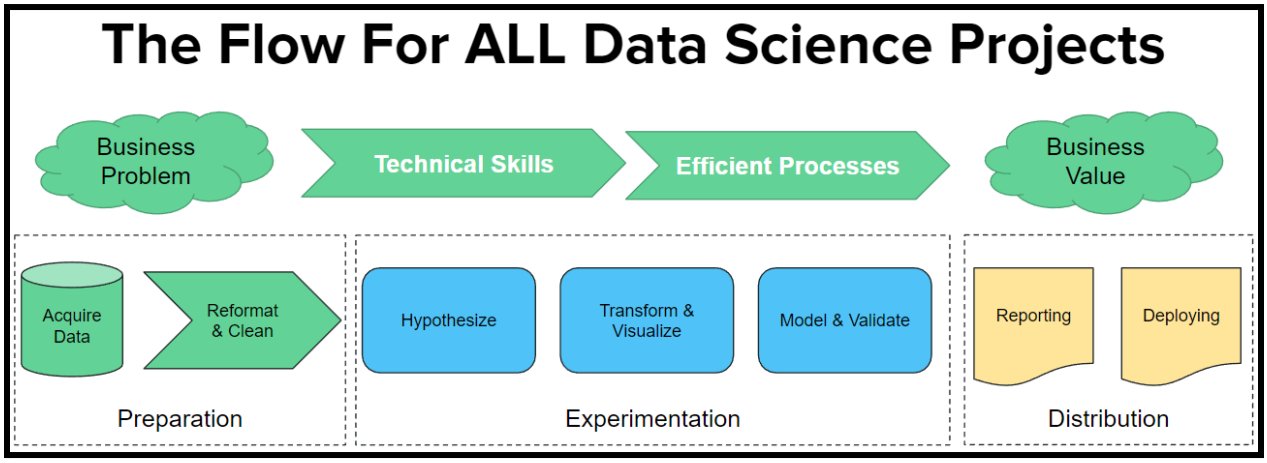
Coming out of the training, Moody, the facilitator of the training at S&P Global said that this training would serve as the basis for all future training at S&P. I was shocked and humbled.



And I knew that if S&P was getting results from my framework (many of whom had zero prior knowledge of data science), then I needed to spread the word.

The Data Science Workflow (the Flow for ALL data science projects)

Inside of a data science project, there is a 3-phase framework that I follow called the Data Science Workflow. It's designed for data scientists and data science teams that need to go from problem to value efficiently.

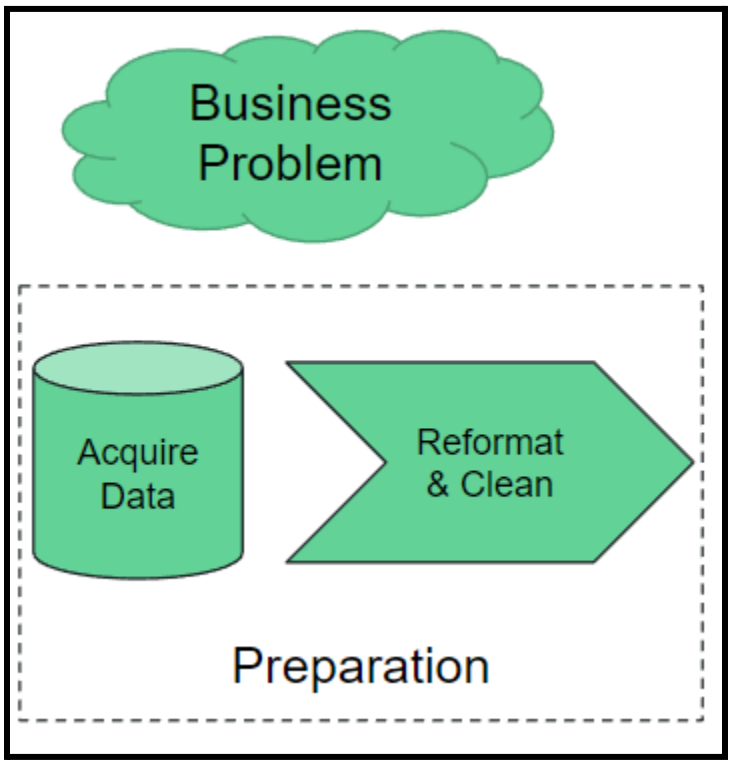


The Data Science Workflow

The Data Science Workflow has 3 key phases: (1) preparation, (2) experimentation, and (3) distribution. Let's break each of these phases down.

Phase 1: Preparation

The first phase is called preparation. This is where I acquire data and then format the data for visualization and modeling. We sometimes call this the data cleaning phase.



Phase 1: Preparation

Acquiring Data

I typically acquire data from three sources. The first data source is the internal databases of the company. This is often stored in a SQL database or inside of an Enterprise Resource Planning (ERP) database. For S&P, they have a ton of data sources they can access. But the key is to know which ones will solve the business problem.

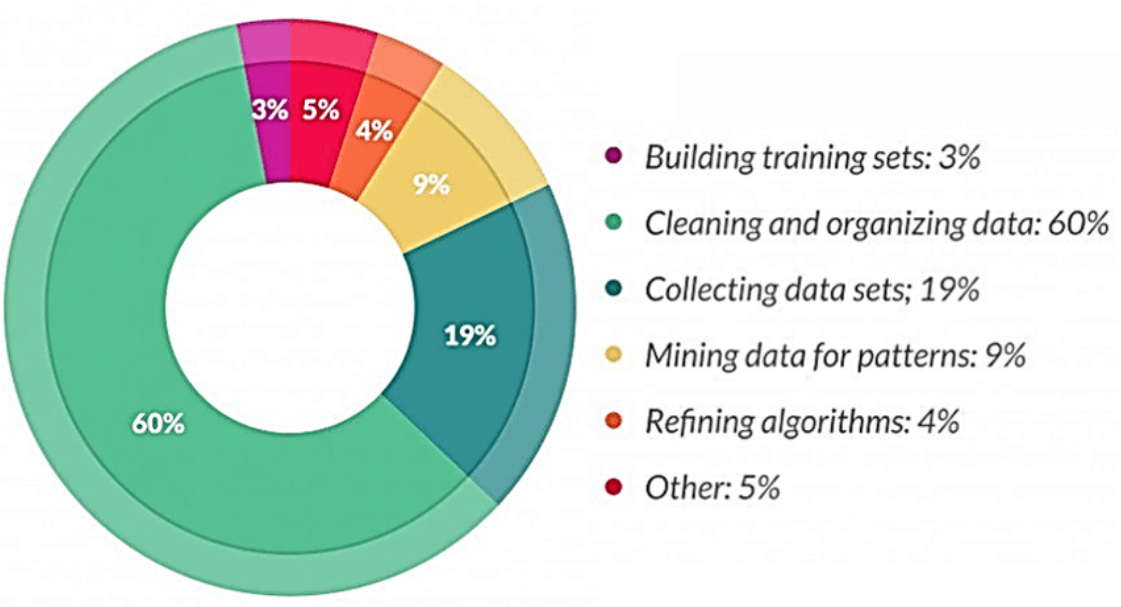
The second data source is external data that the company does not maintain. This might be data sources like stock financial data, commodity data (e.g. Energy, Materials), or housing data. Normally these are accessed through an API, which is a way to request and retrieve data from a 3rd party company. There are both free and paid data sources. A free data source might be the Energy Information Agency (EIA) database that I used to use to collect information on energy supply and demand.

The third data source is data that you create through a process called web scraping. Web scraping is a way to collect data from from the HTML of websites. I did this to collect information on my previous company's competitors, which had the product prices and descriptions of all of their product lines on their website. This formed a competitive advantage because we knew which products we had pricing power on.

Reformatting and Cleaning

Once data was collected from multiple data sources, I would evaluate the business question and then investigate the data I had available. I would go through a process of combining the data, assessing the data quality (how much was missing data or poorly formatted data), fixing the key issues.

Many data scientists don't realize that when starting out that data cleaning is incredibly time consuming. According to the following chart, data scientists actually spend 60% of their time cleaning and organizing data. Plus another 19% of their time collecting data. So in total, about 80% of a data scientist's time is spent in Phase 1.



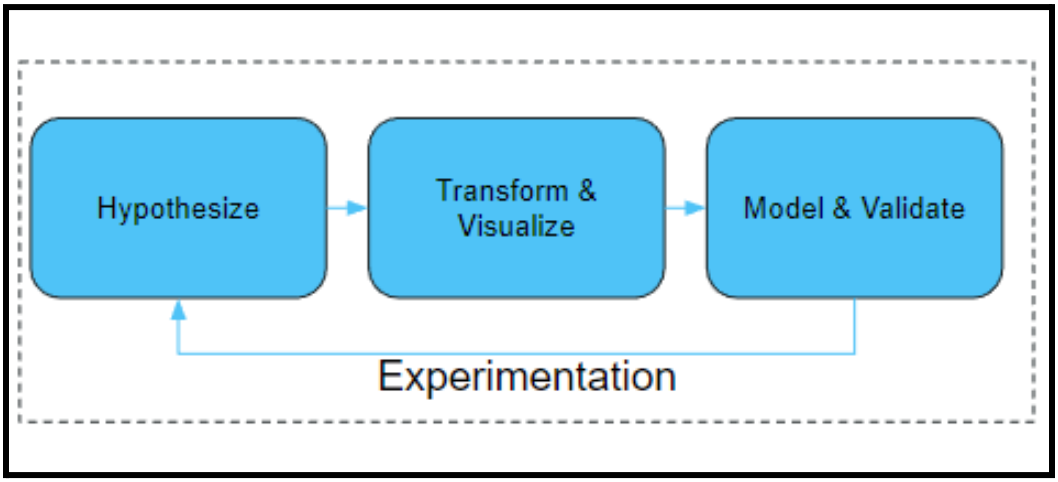
Source: [Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task]

Pro-Tip: Save your data pipeline scripts.

I found it useful to store my data formatting and cleaning scripts in a private R package for my company. This allowed me to retrieve my data cleaning scripts on-demand, saving me a ton of time for future projects. In larger companies, Data Engineering teams actually focus on this goal, but in many smaller companies you'll need to do this yourself. And it's a worthwhile skill to learn because it actually helps you get to know your data much better.

Phase 2: Experimentation

The second phase is called experimentation. This is where I hypothesize what could be causing the problem, transform and visualize the data, and model and validate.



Phase 2: Experimentation

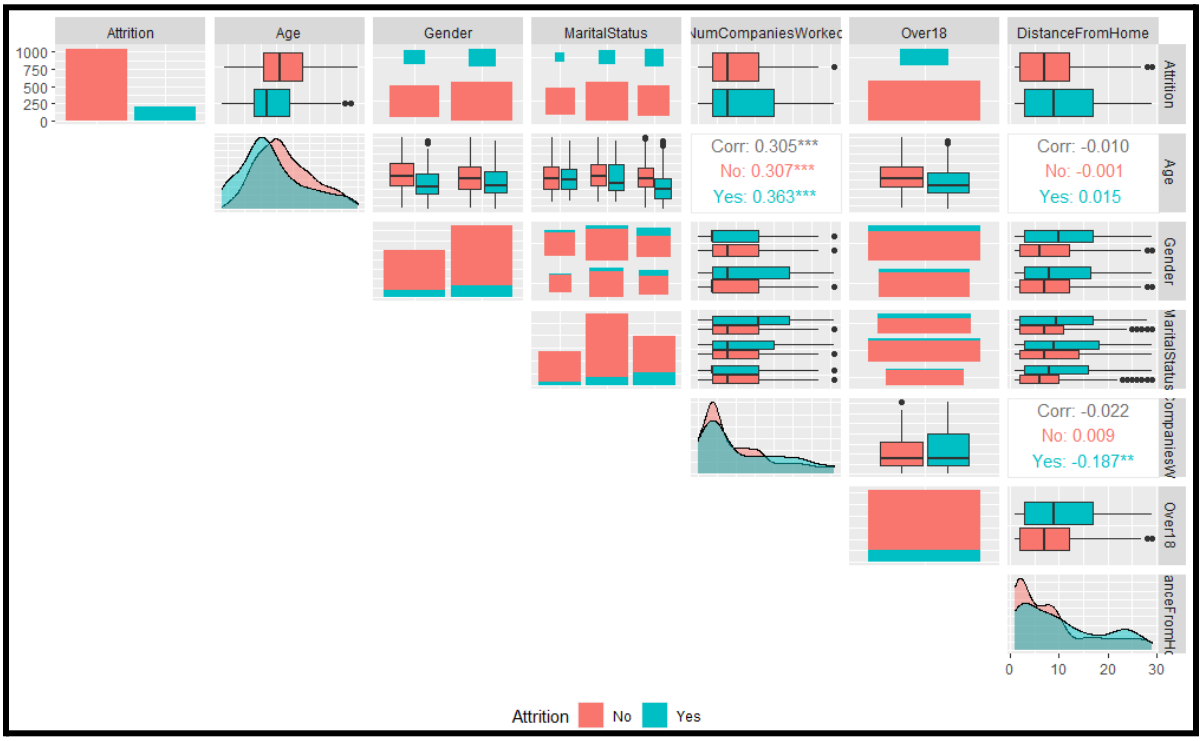
This is a loop where I might need to iterate and refine my hypothesis and improve models as I learn about the business problem through the data I've collected and processed. Let's dive into the key steps.

Hypothesize

A hypothesis is just a question that I come up with after thinking through the problem. Here's a great way to think of the different steps to creating a hypothesis. I have limited evidence, just an observation of the business problem and some data. I generate questions. Next, I hypothesize to *explain* what might be causing the issue.

Transform and Visualize

With a hypothesis in hand, I then transform and visualize the data to see what might be related to the problem. I create visualizations like this to help tell the story of what's happening.



I Visualize What Could Be Related to the Problem

Model and Validate

I then move to modeling and validation. In this step, I create models that can predict and explain why the model predicts something the way it does. I validate my models using 5-fold cross validation to make sure that they aren't bad because a bad model is no use.

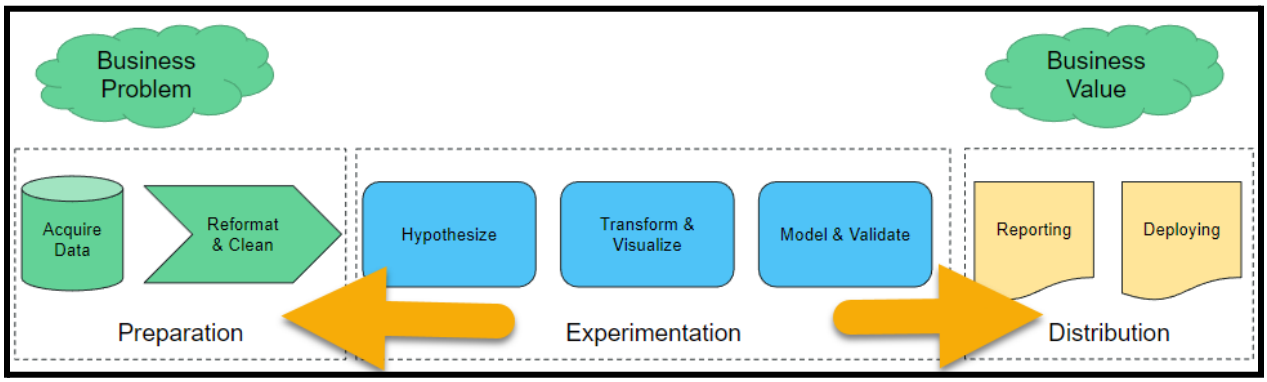
```
library(h2o)

h2o.init()

train_h2o <- as.h2o(train_tbl)
test_h2o  <- as.h2o(test_tbl)

y <- "Attrition"
x <- setdiff(names(train_h2o), y)

automl_models_h2o <- h2o.automl(
    x = x,
    y = y,
    training_frame    = train_h2o,
    max_runtime_secs = 30,
    nfolds           = 5
)
```

Modeling using H2O's Built-in 5-Fold Cross Validation

I then explain the models with special software that show which features they are using to make the prediction. This can give me insights beyond the exploratory visualizations.

Feature Importance and Model Explanations

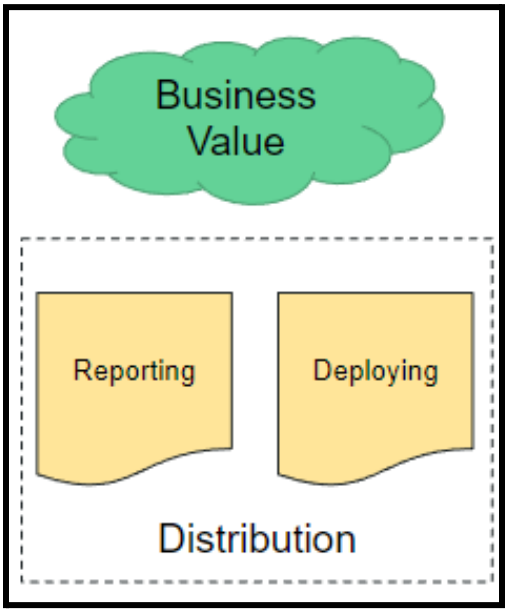
At this point I assess whether or not I've made a good model that explains the problem and can predict with high accuracy. If I do not have a good model, I head back to either the hypothesis stage or data collection stage. If I have a good model, then I will move forward to the next phase, Distribution.



I may go backwards or forwards depending on the model results

Phase 3: Distribution

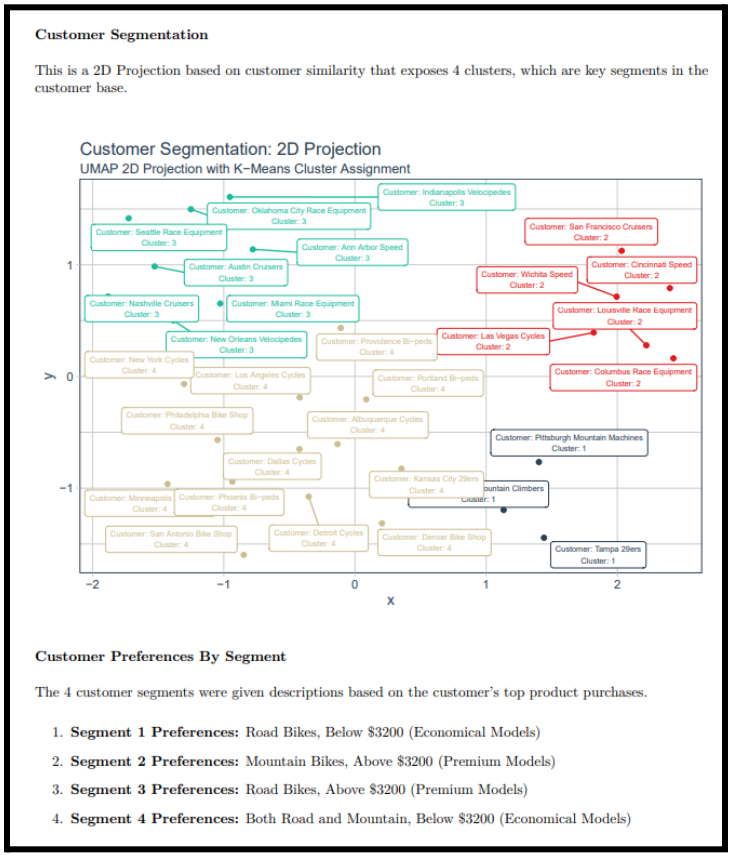
In the distribution phase, I'm developing outputs in the form of reports that management will review or applications that the organization will use to drive decisions. This stage is the most important because it's where the business gains value from all of the data science work that you've done.



Phase 3: Distribution

Reporting

The first output type is a report. I'd make reports like this Customer Segmentation Report (shown below) to document the solution, explain the story with visualizations, and to gain buy-in from leadership. Leadership reviews the report and then either accepts or rejects my proposal.



Customer Segmentation

This is a 2D Projection based on customer similarity that exposes 4 clusters, which are key segments in the customer base.

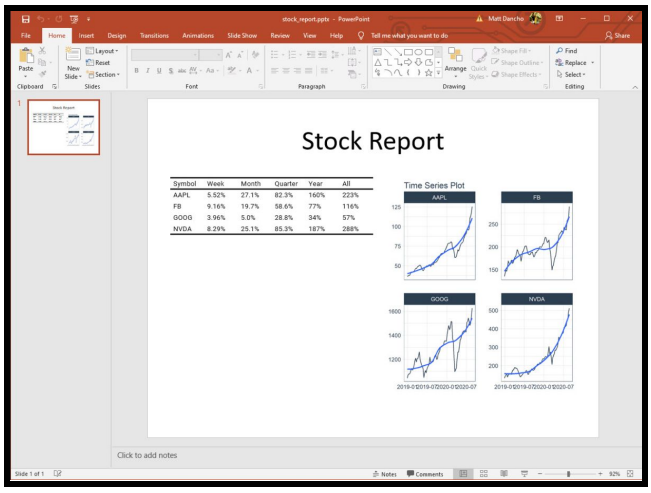
Customer Preferences By Segment

The 4 customer segments were given descriptions based on the customer's top product purchases.

1. **Segment 1 Preferences:** Road Bikes, Below $3200 (Economical Models)
2. **Segment 2 Preferences:** Mountain Bikes, Above $3200 (Premium Models)
3. **Segment 3 Preferences:** Road Bikes, Above $3200 (Premium Models)
4. **Segment 4 Preferences:** Both Road and Mountain, Below $3200 (Economical Models)

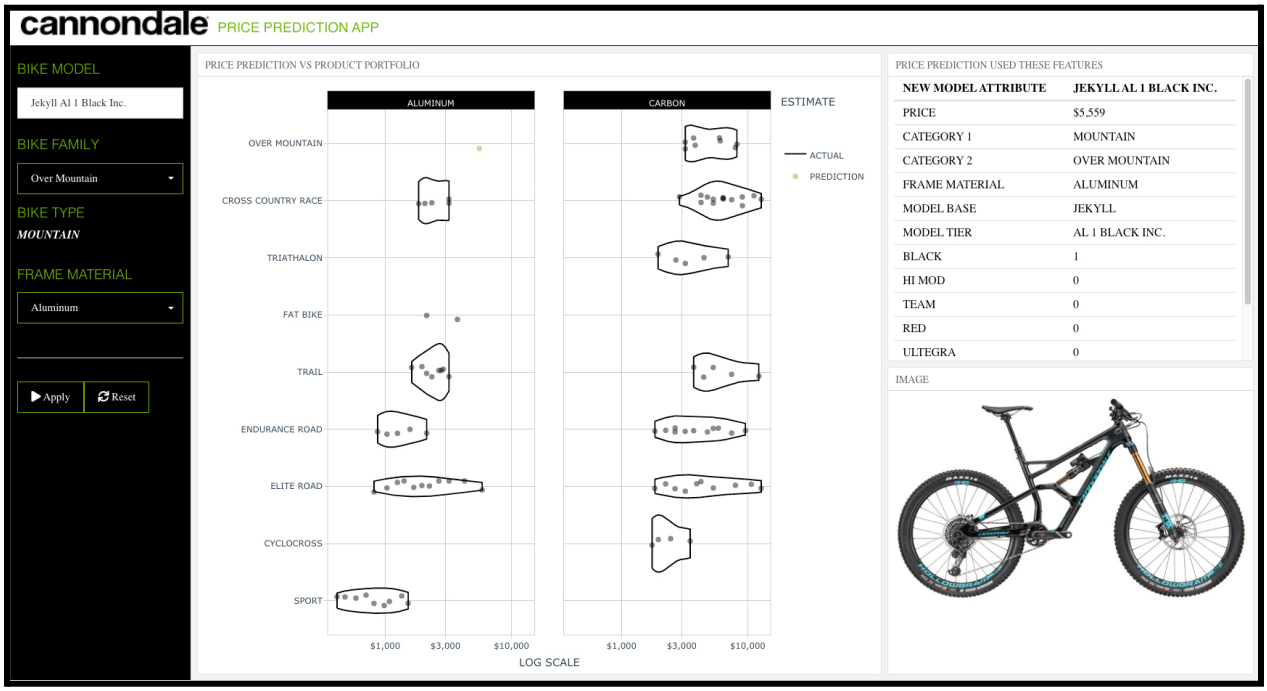
Customer Segmentation Report

PowerPoint Slide Decks. I'd commonly create slide decks using software like PowerPoint, which can be used to quickly summarize the problem, solution, and my recommendation. I discuss the importance of [powerpoint automation here](#).



Deployment

Once an organization approved my idea, the next step was to develop an application and deploy it.

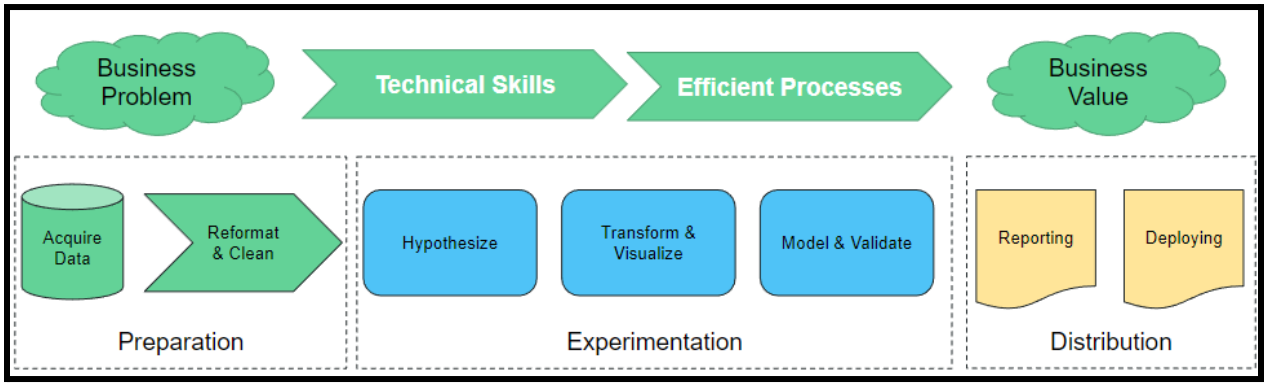


Deployed Web Application

I would make web applications like the one shown here. The applications are deployed in the cloud or on servers so the business can use them anywhere to help them use my prediction algorithms to make good decisions about customers and products.

The best data science teams spend time at the ends of the workflows

I became better as a data scientist and a consultant the faster I got at completing the process. And, I also saw this when I researched other high performing data science teams like *OneSixtyTwo Digital Capital* (which you learned about in Chapter 6).



The best data scientists spend time at the ends of the workflow
(and move fast through the process)

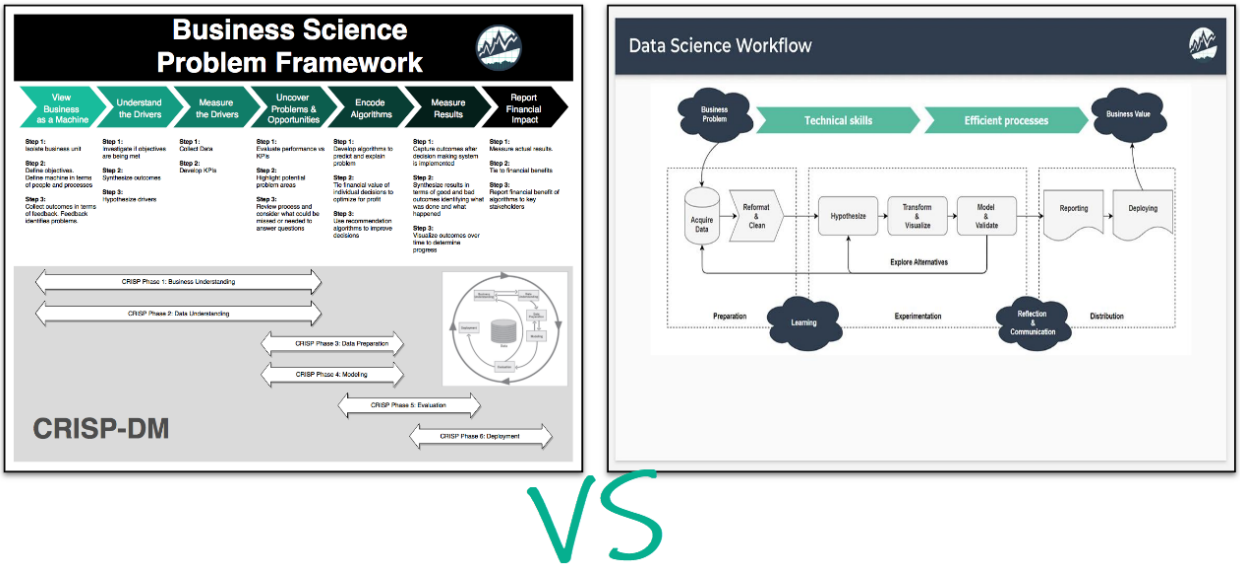
The best data science teams can iterate through this process going from business problem to business value very efficiently, spending little time on modeling and maximum time at the ends of the spectrum. By learning the technical skills and developing an efficient process I was able to increase my productivity by spending time at the ends of the workflow.

At the beginning of the workflow: I'd spend time on business understanding, working with domain experts (and understanding their problems), learning about the data (data understanding), fixing any data issues (data quality), and developing good features for modeling (feature engineering).

At the end of the workflow: I'd spend time on communication with project stakeholders and developing web applications that I knew would be useful given my in-depth knowledge of their challenges.

What about the BSPF Framework?

At this point you might be thinking, *"What about the Business Science Problem Framework?"* Here's the relationship between the two and also their key differences.



The BSPF Framework vs the Data Science Workflow

The Business Science Problem Framework is *how I take organizations* through a business project with data science. Organizations need to be brought into your project so you can work together. And they need to be brought along with your project at specific points in the process. That's what the BSPF helps with.

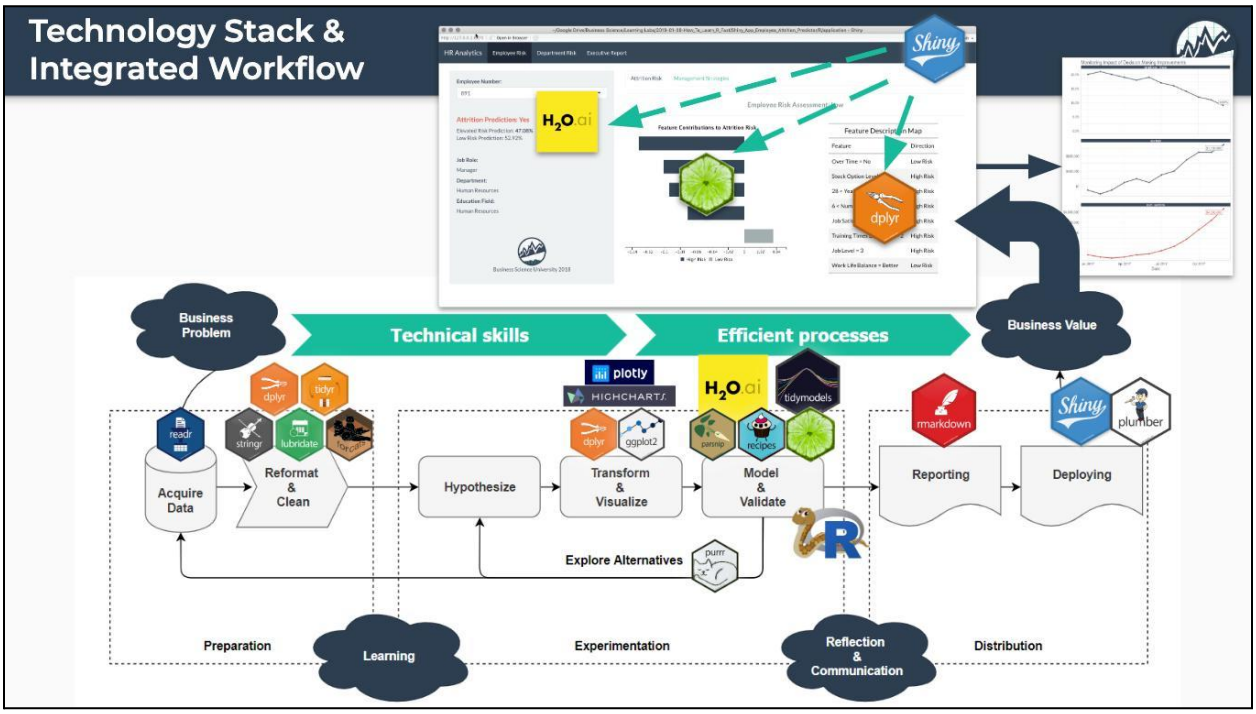
The Data Science Workflow is *how I complete* the technical aspects of the data science project. The data science workflow is performed by me (or my data science team) using the 14 data science skills (from Chapter 1) that I have developed and mastered in becoming a Business Data Scientist.

Do you see the difference?

The BSPF ensures the organization's success while the Data Science Workflow ensures my success (or the data science team's success). You need both to be successful.

Data Science Tools Exposed

Learning how to implement the Data Science Workflow requires knowing which tools to use and in what areas of the workflow they belong. Here's exactly where the R Packages fit you learned from the 14 data science skills (from Chapter 1) fit into the Data Science Workflow.



The Data Science Workflow - With Tools Exposed

Business value comes from using a specific set of R packages that incrementally add value along the Data Science Workflow. This allows us to start with a business problem and end with a web application that delivers massive business value that is tracked with reports and measured for ROI (Return on Investment).

To increase your technical skills, focusing on this specific set of R libraries cuts the time to learn data science dramatically. This is the 80/20 Rule, which states that roughly 80% of tasks can be accomplished by learning only 20% of the R ecosystem.

The tools combine into an integrated approach to solving problems. Therefore, we can't just read a book on each tool independently. We need to learn the tools together to harness their power. This is why exposing yourself to projects is important. It's through projects that you learn to integrate the tools together.

The application is the business value. Without it, the data science team adds little value to the organization. Any business improvement should be tracked, reported on, and measured for return on investment. This is where the Data Science Workflow Framework bridges the gap into the Business Science Problem Framework.

...

You now have everything you need to know what's in front of you when becoming a Business Scientist. But, I also know that habits are really hard to change. I'm a successful Business Scientist, and the information I gave you is already working for me. It's also working for many others, and I'd like to share their stories with you next. Why?

Because this is about YOU!

If you leave now, you might think you learned a lot of cool stuff, but my guess is that by tomorrow morning you'll have already slipped back into your normal routines. Right? You'd just do what you've always done. That's what most people do.

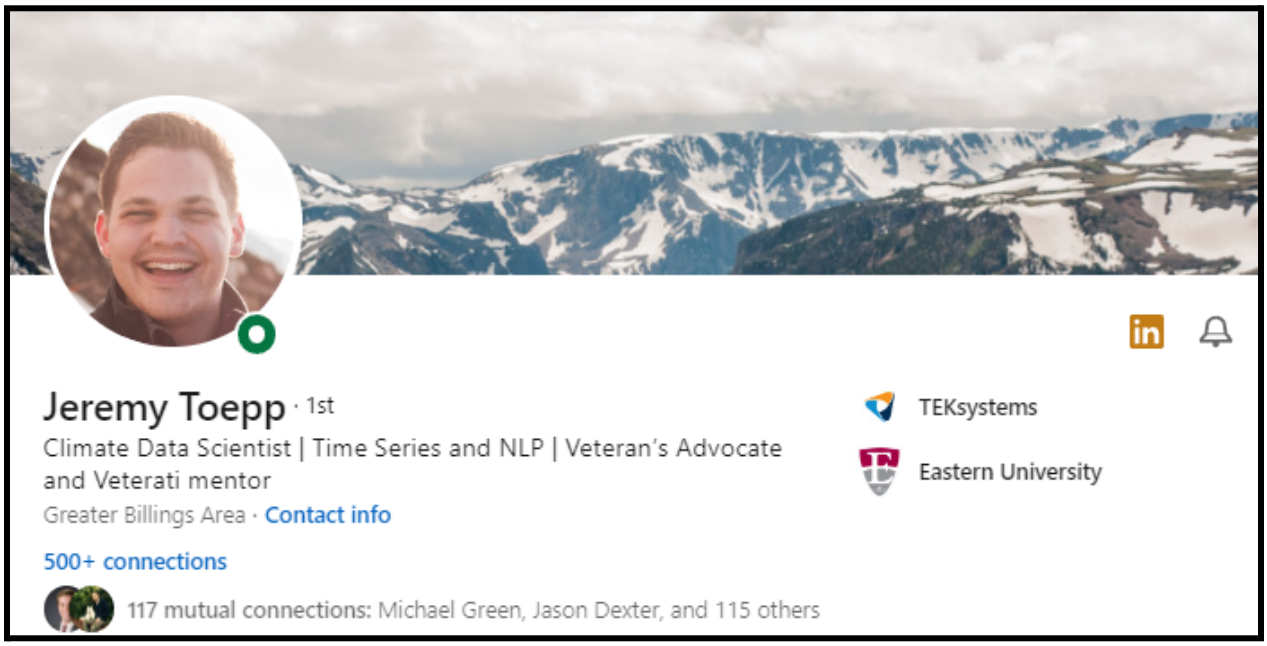
But because I'm your coach, your friend, your mentor, I'm not going to let you go back to your old habits. I'm going to make sure you're successful by breaking them. If you want real, lasting change, you need repeated exposure to what I'm about to tell you. **I will unveil the surprise next.**

Your Special Surprise!

I want to help you take control of your future, and…

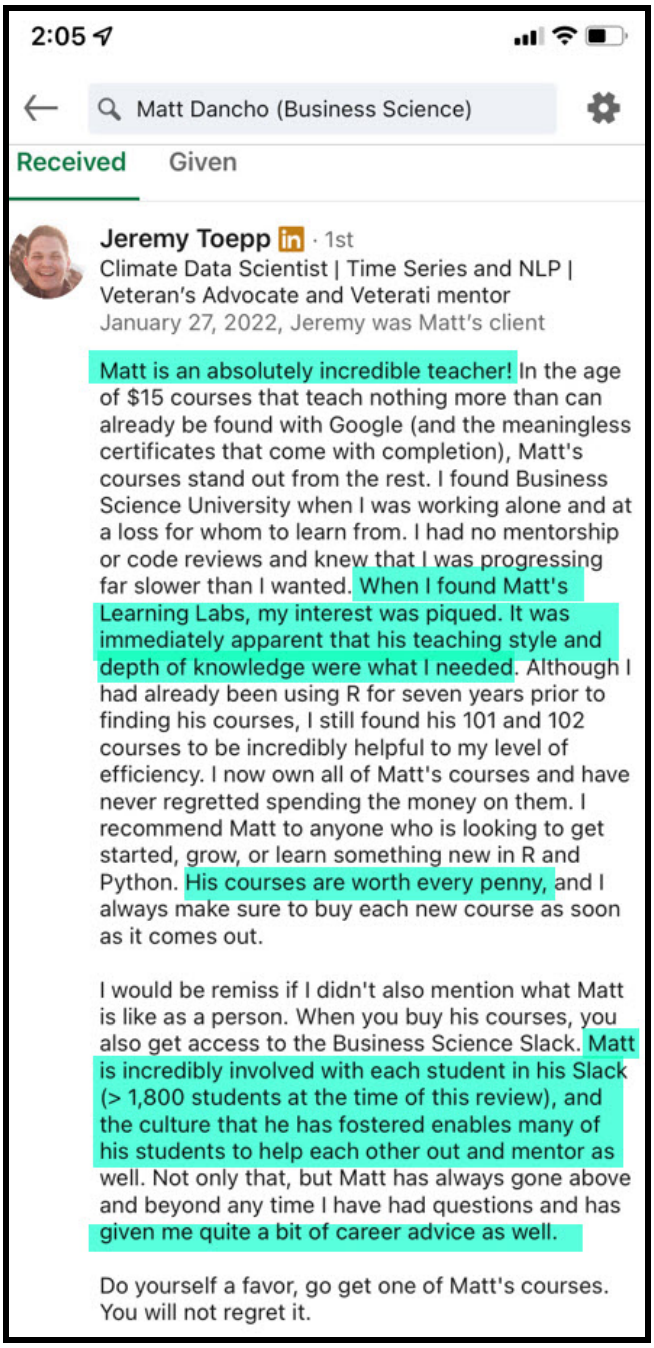
I want to help you become a Business Scientist!!

First, I'd like to introduce you to Jeremy Toepp so you can understand what this journey is like. Jeremy is a senior data scientist. More importantly, he's a Business Scientist. Here is his story.



I've had the opportunity to mentor Jeremy since June 2020 through my R program. Jeremy has been an exceptional student, putting in the hard work to learn the latest technologies and successfully apply them to his projects. And, I've watched his career grow as a result.

I've had a fun time watching his career take off as he successfully used R in production along with his growing abilities in Time Series Forecasting. I saw how his experience putting data science products into production led to a new opportunity as a Senior Data Scientist. And, I've even seen how Jeremy has connected with others in our Business Science Community Slack Channel. This networking has led to new opportunities to collaborate with other successful students. And it led to a new data science position at TEKsystems. Jeremy was kind enough to lend me a review that explains the difference between what he learned from me and



Jeremy says,

> *"I found Business Science University when I was working alone and at a loss for whom to learn from. I had no mentorship or code reviews and knew that I was progressing slower than I wanted."*

This is the all too common feeling that leads to imposter syndrome and self doubt.

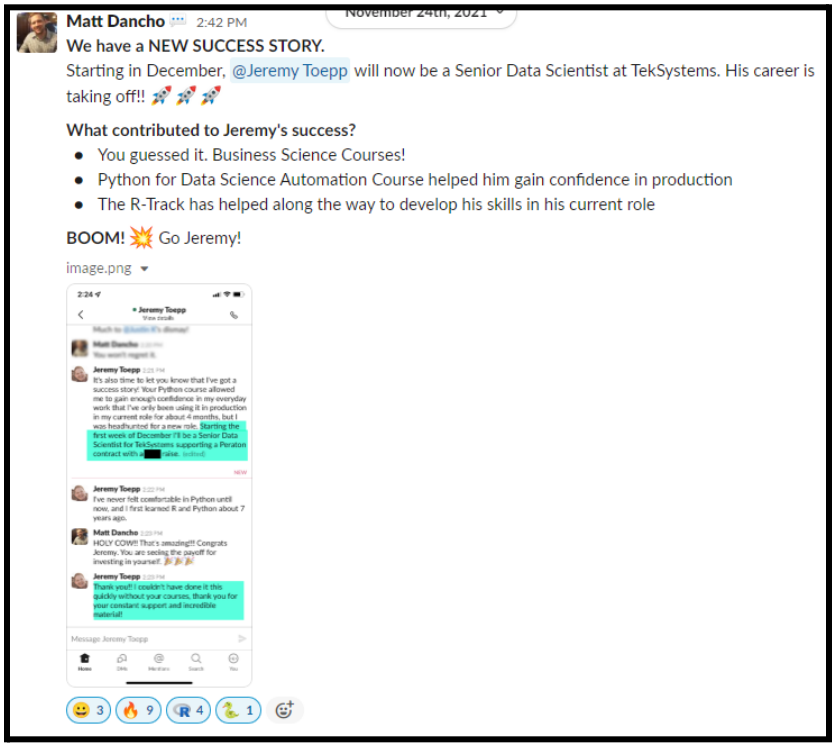
The important part is what most people don't miss when they enroll in a program, a mentor. Jeremy says,

> *"I would be remiss if I didn't mention what Matt is like as a person. Matt is incredibly involved with each student in his Slack (now > 2500 students), and the culture that he has fostered enables many of his students to help each other out and mentor as well."*

What Jeremy saw was the care that I have for every student that makes a commitment to me. I make a commitment right back to them. Jeremy continues,

> *"Matt has always gone above and beyond any time I have had questions and has given me quite a bit of career advice as well. Do yourself a favor, go get Matt's course. You will not regret it."*

This is the transformation that only happens when a student and mentor combine in a supportive environment. One where you know you have real people that are going to help you when you stumble. One where you can depend on an experienced mentor to help get you through your sticking points. One where you share in others successes and they share in yours.



Sharing in our students' successes!

Business Science is about you.

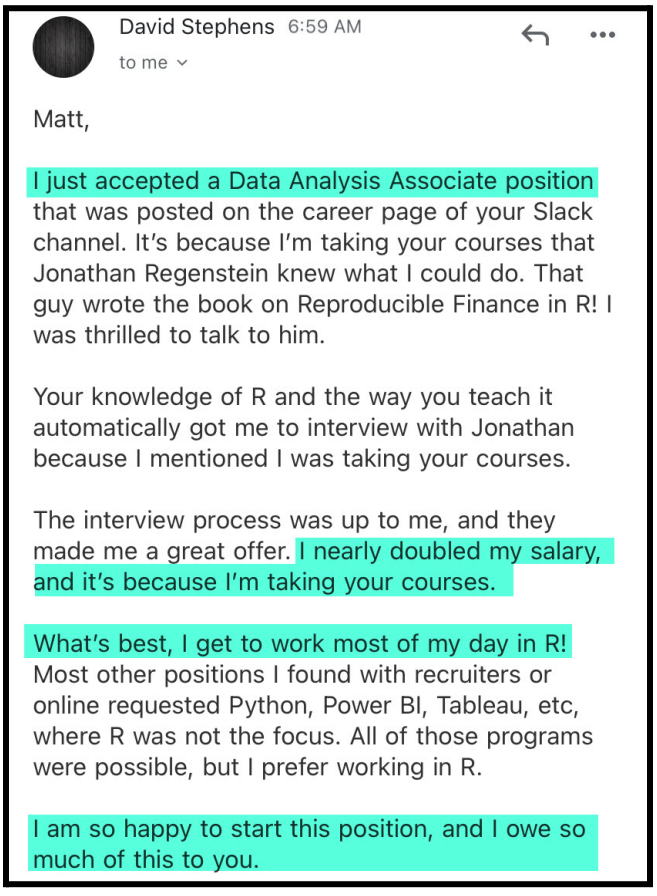
My mission is to help you take control of your career by becoming a Business Scientist. Whether your goal is to add data science to your current role, move into a new career as a data scientist, or simply start fresh as a data scientist, these are some of the results I've helped my students achieve. And, I want to do this for you. I not only want to show you that I can help you, but how my students are doing it.

First, I'd like to introduce you to David.

How David doubled his salary.

David reached out as he just landed a new role nearly doubling his salary. He told me,

> *"I just accepted a Data Analysis Associate Position that was on the career page of your Slack. I nearly doubled my salary. I'm so happy to start this position."*



David Stephens 6:59 AM

to me

Matt,

I just accepted a Data Analysis Associate position that was posted on the career page of your Slack channel. It's because I'm taking your courses that Jonathan Regenstein knew what I could do. That guy wrote the book on Reproducible Finance in R! I was thrilled to talk to him.

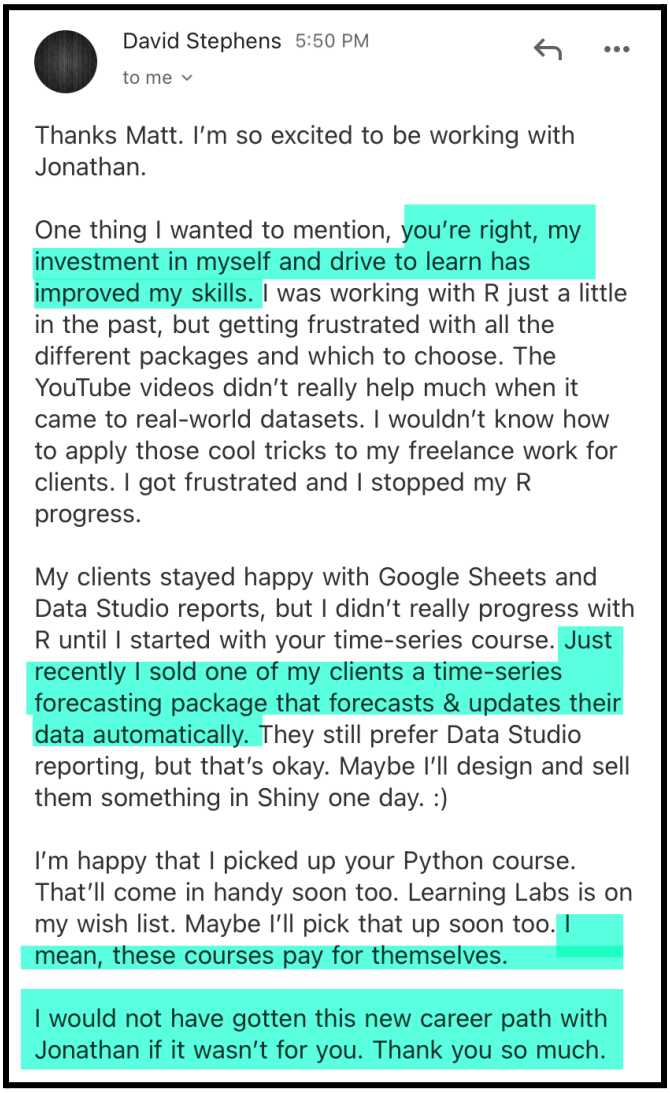
Your knowledge of R and the way you teach it automatically got me to interview with Jonathan because I mentioned I was taking your courses.

The interview process was up to me, and they made me a great offer. I nearly doubled my salary, and it's because I'm taking your courses.

What's best, I get to work most of my day in R! Most other positions I found with recruiters or online requested Python, Power BI, Tableau, etc, where R was not the focus. All of those programs were possible, but I prefer working in R.

I am so happy to start this position, and I owe so much of this to you.

I knew his new boss, Jonathan, and the next day I talked with Jonathan to find out more. After I got off a phone call with Jonathan, I knew exactly why David was hired.



David Stephens 5:50 PM

to me

Thanks Matt. I'm so excited to be working with Jonathan.

One thing I wanted to mention, you're right, my investment in myself and drive to learn has improved my skills. I was working with R just a little in the past, but getting frustrated with all the different packages and which to choose. The YouTube videos didn't really help much when it came to real-world datasets. I wouldn't know how to apply those cool tricks to my freelance work for clients. I got frustrated and I stopped my R progress.

My clients stayed happy with Google Sheets and Data Studio reports, but I didn't really progress with R until I started with your time-series course. Just recently I sold one of my clients a time-series forecasting package that forecasts & updates their data automatically. They still prefer Data Studio reporting, but that's okay. Maybe I'll design and sell them something in Shiny one day. :)

I'm happy that I picked up your Python course. That'll come in handy soon too. Learning Labs is on my wish list. Maybe I'll pick that up soon too. I mean, these courses pay for themselves.

I would not have gotten this new career path with Jonathan if it wasn't for you. Thank you so much.

David sent me a new message that confirmed everything Jonathan had told me:

> *"You were right. My investment in myself and drive to learn has improved my skills. Just recently I sold one of my clients a forecasting package [with skills learned from your courses]. I would not have gotten this new career with Jonathan if it wasn't for you. I mean, these courses pay for themselves."*

You see, it was the experience of building things that clients valued that Jonathan saw in David. And, how did David learn how to create business value, the very experience that Jonathan wanted so badly?

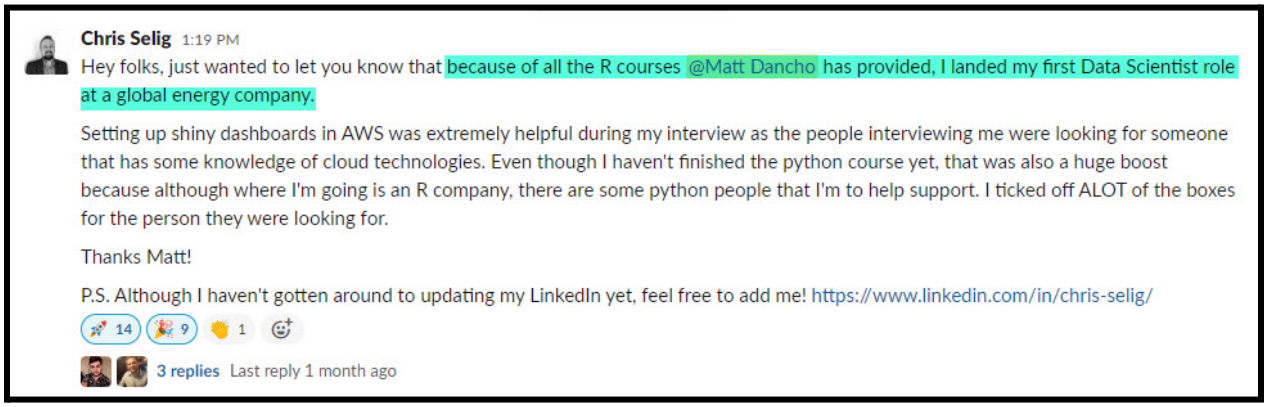
By investing in himself and learning how to build useful tools. This experience landed him his dream job which he found through my Business Science Community. So investing in yourself is the key, right?

Before we move on, think about the life-changing experience that David felt. What would doubling your salary feel like? Would that help you? Would having a career that you love be great for your well-being? That's the power of becoming a Business Scientist with me in your corner.

Next, I'd like to share Chris's powerful story of becoming a data scientist.

How Chris landed his first data scientist role

Chris has been a member of the Business Science tribe since I started my first programs in 2018. He reached out with excitement as he let us know the good news. Chris just landed his first Data Scientist role at a global energy company. What helped Chris get the job? What was it that his new employer saw in Chris?



Chris explained,

> *"Setting up shiny dashboards in AWS was extremely helpful during my interview. The people interviewing me were looking for knowledge of cloud technologies (e.g. AWS). The courses were a huge boost. I checked off a lot of the boxes that the company was looking for."*

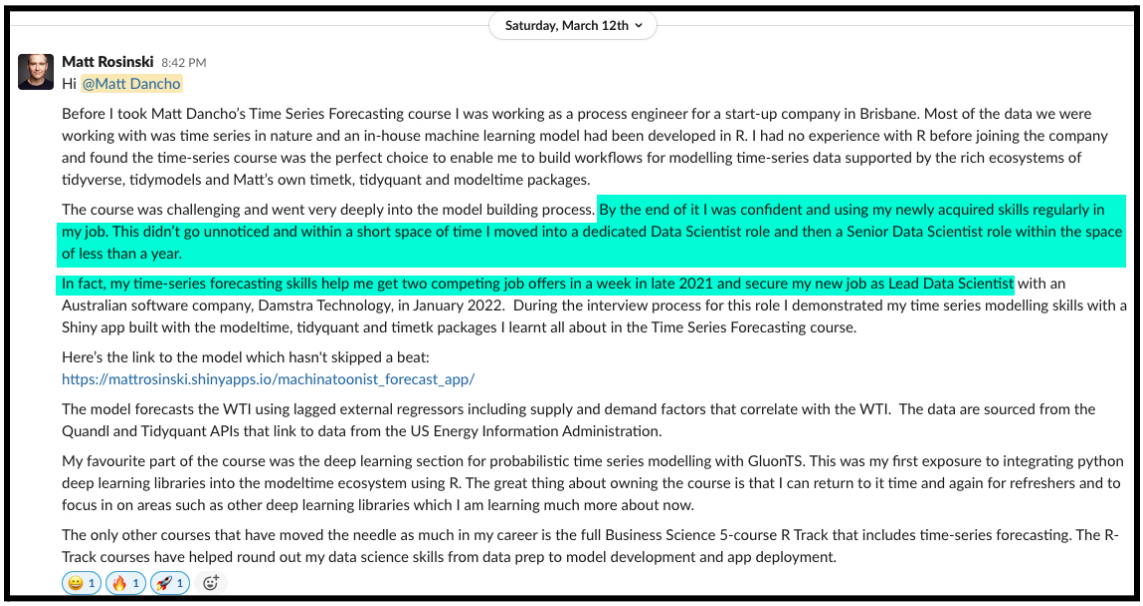
Chris invested in himself, and my program paid off in dividends. He learned the tools and skills that companies needed. And he showed that he could provide business value through web applications and knowledge of cloud technologies that he gained in the courses. As a true

Business Scientist, he took control and reaped the reward, his new Data Science job. And again, investing in himself was the key. You see that, right?

Next, I'd like to introduce you to Matt, who's now a lead data scientist.

How Matt got 2 competing offers in a week

Matt Rosinski was able to land two competing offers in a week, and secured his new job as a Lead Data Scientist. But how did Matt do this?



Saturday, March 12th

Matt Rosinski 8:42 PM

Hi @Matt Dancho

Before I took Matt Dancho's Time Series Forecasting course I was working as a process engineer for a start-up company in Brisbane. Most of the data we were working with was time series in nature and an in-house machine learning model had been developed in R. I had no experience with R before joining the company and found the time-series course was the perfect choice to enable me to build workflows for modelling time-series data supported by the rich ecosystems of tidyverse, tidymodels and Matt's own timetk, tidyquant and modeltime packages.

The course was challenging and went very deeply into the model building process. By the end of it I was confident and using my newly acquired skills regularly in my job. This didn't go unnoticed and within a short space of time I moved into a dedicated Data Scientist role and then a Senior Data Scientist role within the space of less than a year.

In fact, my time-series forecasting skills help me get two competing job offers in a week in late 2021 and secure my new job as Lead Data Scientist with an Australian software company, Damstra Technology, in January 2022. During the interview process for this role I demonstrated my time series modelling skills with a Shiny app built with the modeltime, tidyquant and timetk packages I learnt all about in the Time Series Forecasting course.

Here's the link to the model which hasn't skipped a beat:
https://mattrosinski.shinyapps.io/machinatoonist_forecast_app/

The model forecasts the WTI using lagged external regressors including supply and demand factors that correlate with the WTI. The data are sourced from the Quandl and Tidyquant APIs that link to data from the US Energy Information Administration.

My favourite part of the course was the deep learning section for probabilistic time series modelling with GluonTS. This was my first exposure to integrating python deep learning libraries into the modeltime ecosystem using R. The great thing about owning the course is that I can return to it time and again for refreshers and to focus in on areas such as other deep learning libraries which I am learning much more about now.

The only other courses that have moved the needle as much in my career is the full Business Science 5-course R Track that includes time-series forecasting. The R-Track courses have helped round out my data science skills from data prep to model development and app deployment.

😊 1 🔥 1 🚀 1

Matt has been investing in himself through my program. He learned the `tidymodels` ecosystem for machine learning, he learned time series forecasting with my `modeltime` package, and he learned how to build web applications with `shiny` through my programs. But it's how he used those skills to sell his value like a true Business Scientist.

Matt said,

> *"I was using my newly acquired skills at my job. This didn't go unnoticed. Within a short time I transitioned into Data Scientist and then Senior Data Scientist."*

But Matt didn't stop there. Matt built a web application that demonstrated what he had learned in my courses.

Matt's Shiny Forecasting Application

Matt then used the application as the basis of attracting his prospective companies and then wow-ed them in the interview. Here's what Matt says,

> *"During the interview process I demonstrated my time series forecasting skills and my shiny web application. The only courses that have moved the needle as much in my career has been the Business Science 5-Course R-Track. The R-Track courses have helped to round out my data science skills from data preparation to web application deployment."*

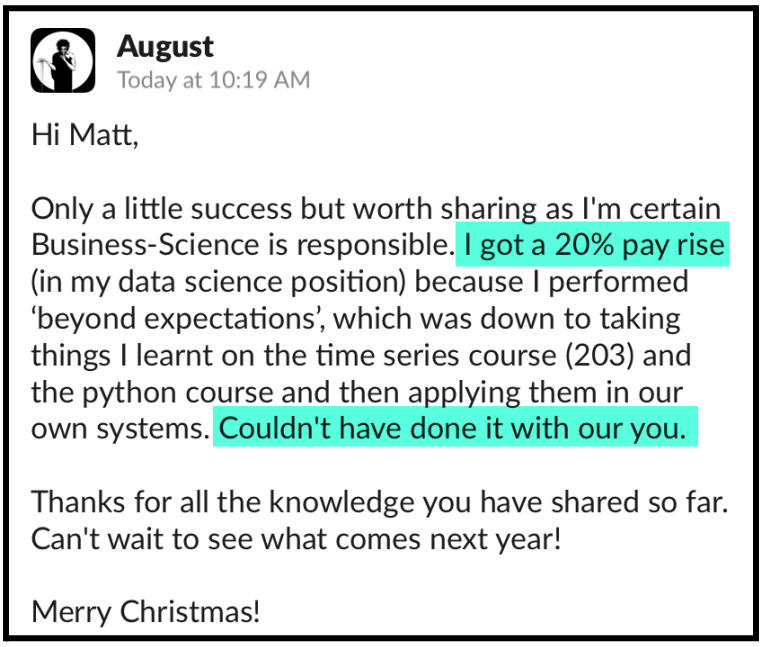
The key to Matt's success in the interview was how he presented his work. How did Matt create his work? By investing in himself through my R-Track Program. Each success story is the same, it starts with an investment. Does that make sense?

The way of the Business Scientist is the path to your new life

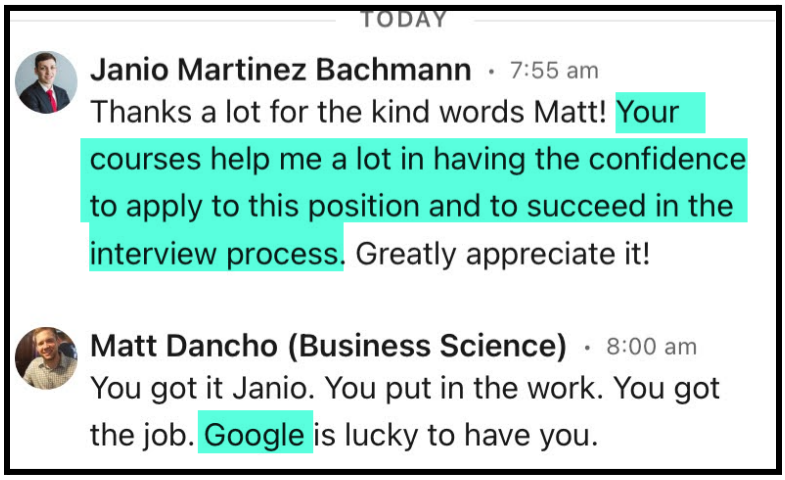
By now you've seen what happens when you invest in yourself and follow the way of the Business Scientist. The next day your life feels a little different as you take control of your destiny. A month in you feel empowered by how things that you couldn't do before seem simple now. And the complex tasks are becoming easier. And then after 90-days you begin feeling the confidence. That confidence transitions into power, and that power is what creates your new career opportunities.

What they are doing

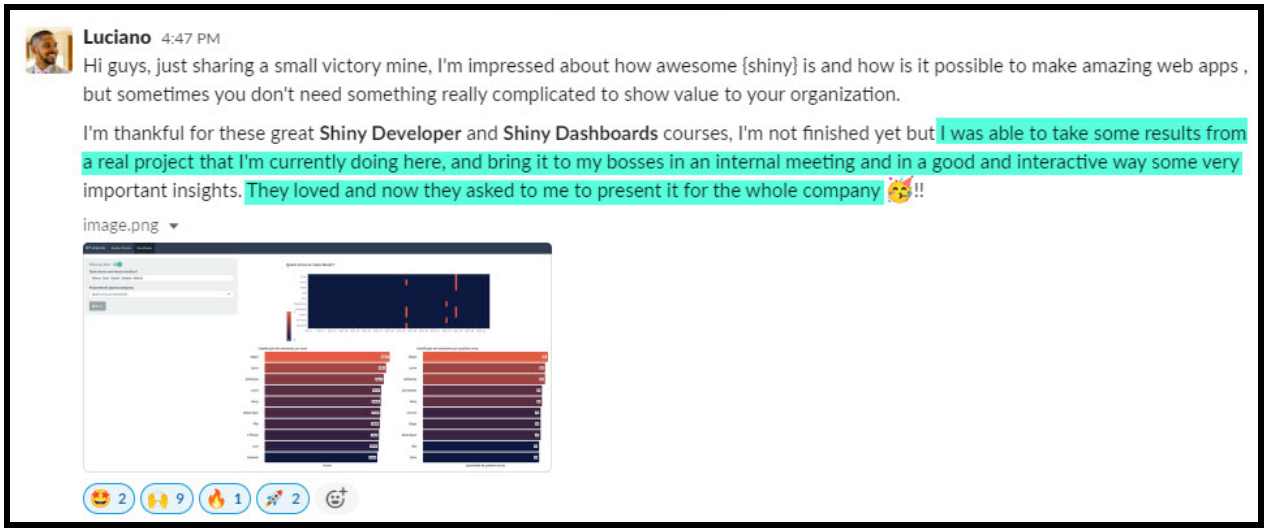
Just imagine. Wouldn't it be amazing if you could accelerate your career with data science? Here's how. The Business Science program that will change your life is what these students have adopted. Can you imagine if any of these stories happened to you? Wouldn't it be great?



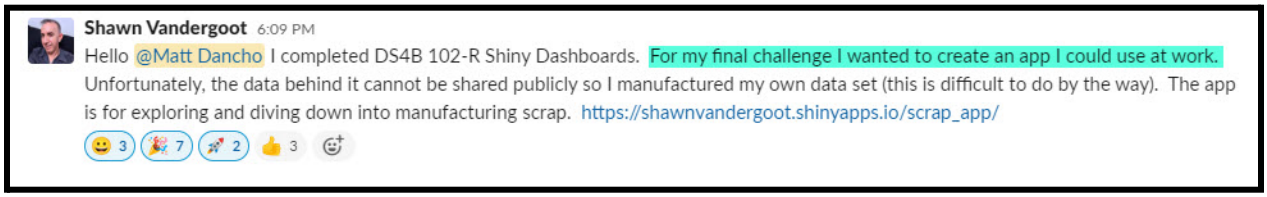
August got a 20% pay raise

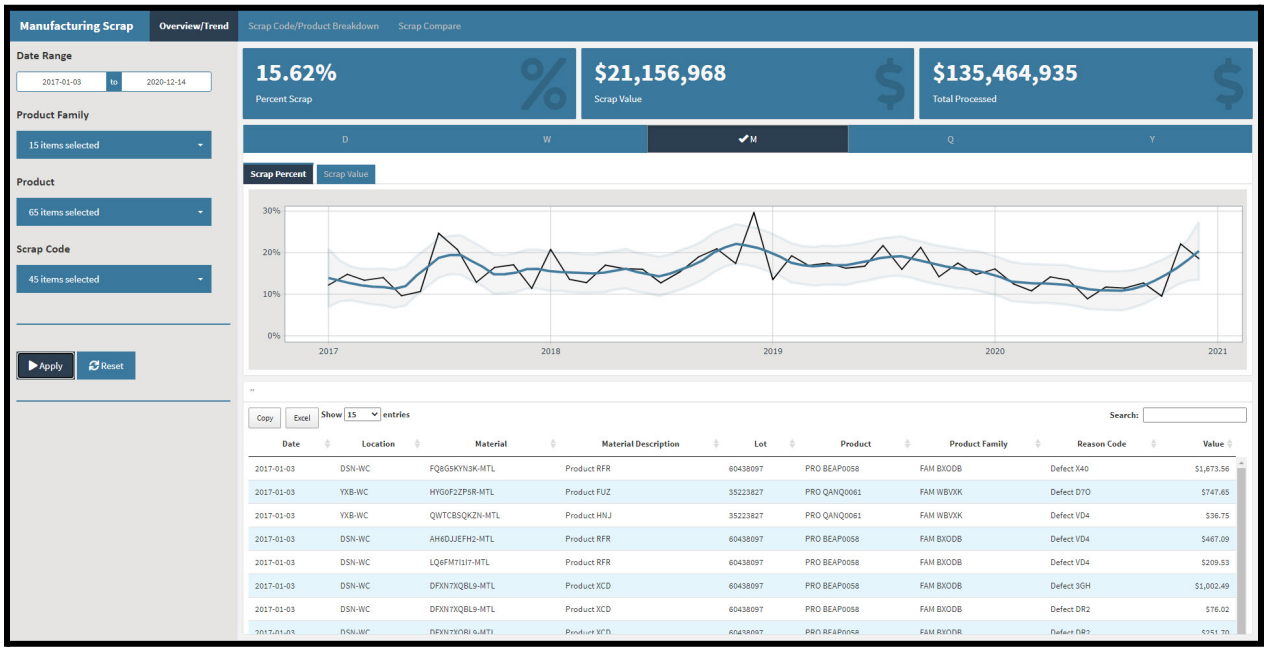


Janio got a job at Google



Luciano wowed his Management Team with a Shiny App





Shawn built a shiny app for his company

What are they doing?

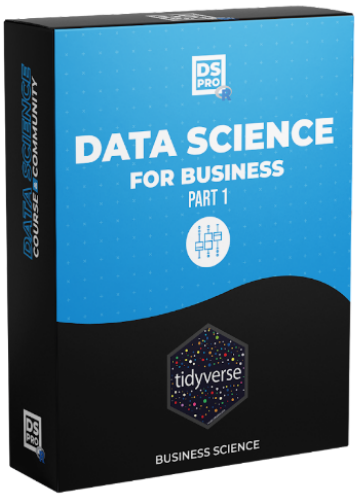
Each of these students have taken the path of the Business Scientist by enrolling and committing to a special program designed to get maximum career-changing results in minimum time.

Their results have come from the 5-Course R-Track Program. This program helps you become a data scientist by teaching exactly how to give companies business value. And if it's okay with you, I'd like to present a special offer to help you become a Business Scientist with me. Are you ready to get started?

If so, then read on.

The first course is designed to get you the foundations of data science.

Data Science For Business Part 1

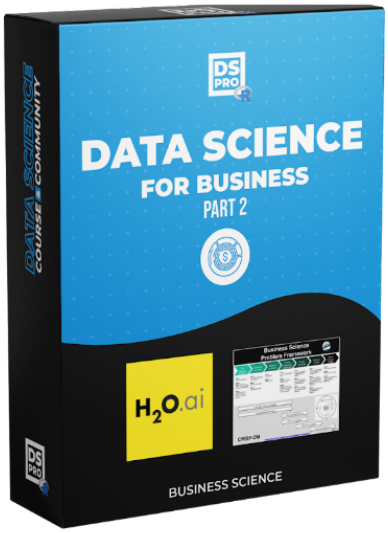


- Foundations of Data Science so you can perform basic Machine Learning
- Produce High-Quality Reports Business Can Use To Generate Insights
- Clean & Work With Data
- Visualize Data Which Produce Those Insights
- Machine Learning algorithm foundations
- These are the general foundational steps for ALL Data Scientists.

I've included this course to teach you how to perform the general foundation steps that all data scientists use to perform basic machine learning including building high-quality reports, cleaning and working data, visualizing and data storytelling, and machine learning algorithm foundations.

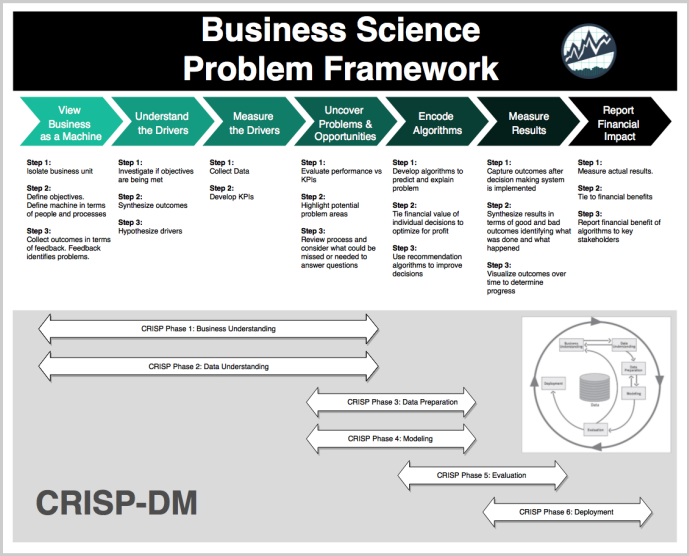
The second course is Data Science for Business Part 2.

Data Science For Business Part 2

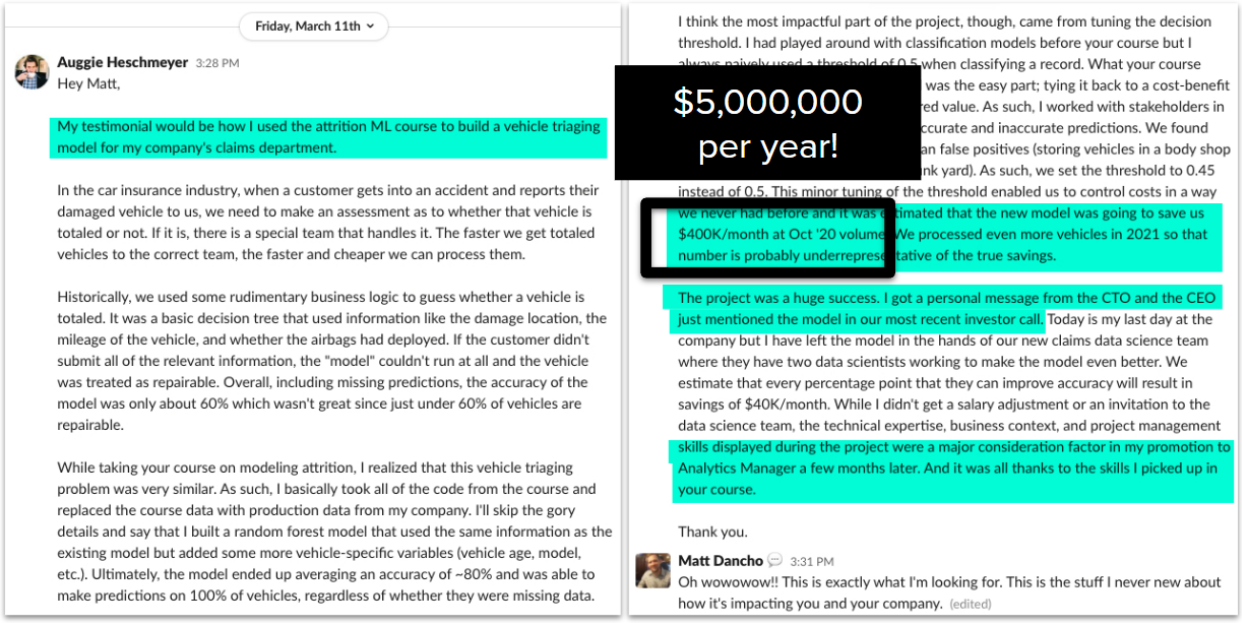


- Solve a $5M Per Year Business Problem
- Use a **repeatable framework** that can be applied to almost any business problem
- Solve the entire business problem step by step
- Use Advanced Tools: Automated Machine Learning & Explainable AI
- Develop Financial Decision Making using Return On Investment ($$$) to show organizational savings

It's in this course that you will learn how to apply the Business Science Framework (from Chapter 2) to solve a $5,000,000 business problem step-by-step.



Remember Auggie (he had an amazing case-study in Chapter 2). Auggie applied the skills he learned in this course to save his organization $5,000,000 per year. He was rewarded with a promotion to Data Analytics Manager and got a personal message from the CTO at his company. Wouldn't it be great if your career took off too? It can.



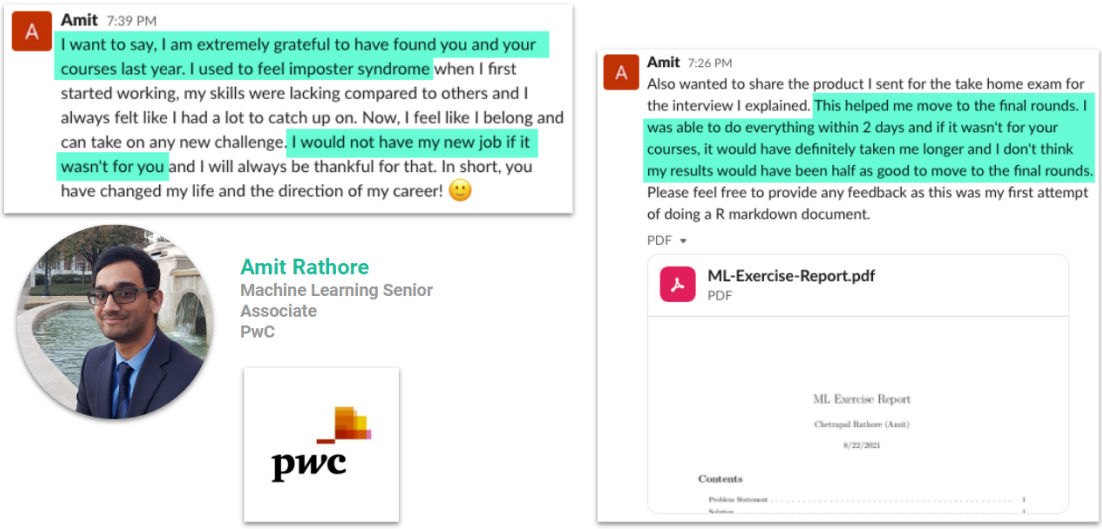
Auggie was promoted to Analytics Manager from applying the skills in this course

The third course is high-performance time series.

High-Performance Time Series

- Solve a special type of problem that costs organizations Millions of dollars per year
- Learn strategies from **4 competitions**
- Use custom software that I wrote to make forecasts at scale (1000+)
- Use Advanced Techniques: Machine Learning, Deep Learning, & Feature Engineering

In this course you learn a special type of problem called forecasting that can save organizations millions of dollars. I teach you strategies from 4 time series competitions using a custom software called `modeltime`. Amit, a student of mine, used this course to land his dream job as a Machine Learning Senior Associate at PwC. Would you like to be my next career case-study (just like Amit)? Because that's what I want for you. Do you think that with my help you could do it?



Amit landed a Machine Learning Senior Associate position with PwC by learning Time Series

Next, is the foundations of shiny web applications.

Shiny Applications Part 1

- The foundations of production for data science
- Learn how to build web applications
- Automate key business analysis & processes
- Empower your company to make data-driven decisions for years to come

I've included this course to teach you how to build basic web applications that automate key business processes and empower your company to make data-driven decisions. This is where you generate business value. Here's a shiny app that Habbee, a student with no prior experience in data science, built for her interview portfolio after she completed the first 4 courses in my system. She made it by following this course. Isn't this amazing?

Habbee build her first shiny app for her data science portfolio

Last is shiny applications part 2.

Shiny Applications Part 2

- Learn Cloud Technologies (AWS, Docker)
- Scale & deploy data science
- Add Security Authentication (Enterprise)
- Connect Backend Databases
- Customize User Interface
- Integrate an API
- Everything you need to know to deploy applications in the enterprise

I've included this course to give you the same skills that helped Chris land his first data science job. You learn cloud technologies and how to build and deploy multi-user web applications for your company. You learn how to connect to backend databases, integrate APIs, add security, all of the skills that companies need to use cloud technology. Chris said, *"I landed my first Data Scientist role. Setting up Shiny Dashboards with AWS was extremely helpful during my interview"*. Pretty exciting, right?

Setting up shiny dashboards in AWS helped land Chris his first Data Scientist job!

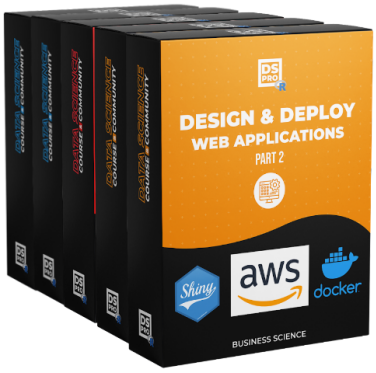
If I could say one thing to summarize, following my data science program will help you become a data scientist and more importantly a Business Scientist, setting you above your competition. These are the skills that took me over 5-years to learn and master, and you are going to be able to learn them in a fraction of the time. How wonderful is that?

- You'll learn how to apply the 14 Data Science Skills (from Chapter 1),
- You'll gain experience running a project with the BSPF Framework (from Chapter 2),
- You'll even build skills like cloud technologies (that we found out are used by Senior Data Scientists in Chapter 3),
- You'll build applications that create business value (the key to marketing yourself from Chapter 4),
- You'll learn fast by learning R with me (from Chapter 5),
- You'll gain the skills that make you immediately valuable as part of a data science team (in Chapter 6), AND
- You'll learn the most important technical skills and efficient processes to complete the Data Science Workflow fast (from Chapter 7).

Universities charge $100,000+ for programs that don't teach half of this. Data Schools take 5+ years to teach a fraction of these skills (and leave you confused in the process). Wouldn't it be great if you could become a Business Scientist in weeks for a fraction of the cost of universities?

I'm pleased to give you access to my R-Track Program, the fast-track to becoming a Business Scientist. I'm not going to charge you $100,000 for a university program. The total value to the public is $3,995 for all 5-courses, which is a bargain. But I'm not even going to charge you that.

What You're Getting Today:

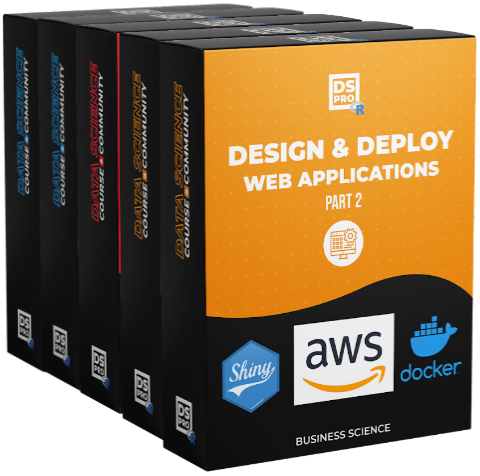


Data Science For Business 1 **($599)**
Data Science For Business 2 **($999)**
High Performance Time Series **($799)**
Shiny Web Applications 1 **($799)**
Shiny Web Applications 2 **($799)**

Total value = $3,995

When you enroll today, I'm going to slash the cost in half.

What You're Getting Today:



Data Science For Business 1 **($599)**
Data Science For Business 2 **($999)**
High Performance Time Series **($799)**
Shiny Web Applications 1 **($799)**
Shiny Web Applications 2 **($799)**

50% OFF!

Total value = ~~$3,995~~ Only $1999

My gift for investing in yourself is access to my program for 50% OFF.

Now I want you to think about something for a minute. What is an investment? A lot of people have a fear about money, and even bigger fears about spending money. But you need to understand that money is good. It's a tool that was created for exchange.

Other than that, there's no real value in money. You can't use it to stay warm or eat it. You can only trade it for something else that you want. Think for a second, everyone who exchanges money for something does it because they believe that what they are getting in exchange is greater than keeping the money for something else. At least that's what I expect when I buy something. I don't actually know for sure until I buy it, try it out, and can see the results.

But my question for you is this. Would you exchange that money for those results? If the answer is yes, then you need to get started right now. And if you have any fear that it might not be what you expected or that you might not be able to get those results, just let us know within the first 30 days, and we'll give you your money back.

I look forward to helping you take control of your career. Are you ready to get started? Then let's go!