Everything You Should Already Know About Data Science.

Matt Dancho

VERSION 1.0

Table Of Contents:

Chapter 1: Which data science skills are important (To get a $50,000 increase in salary)

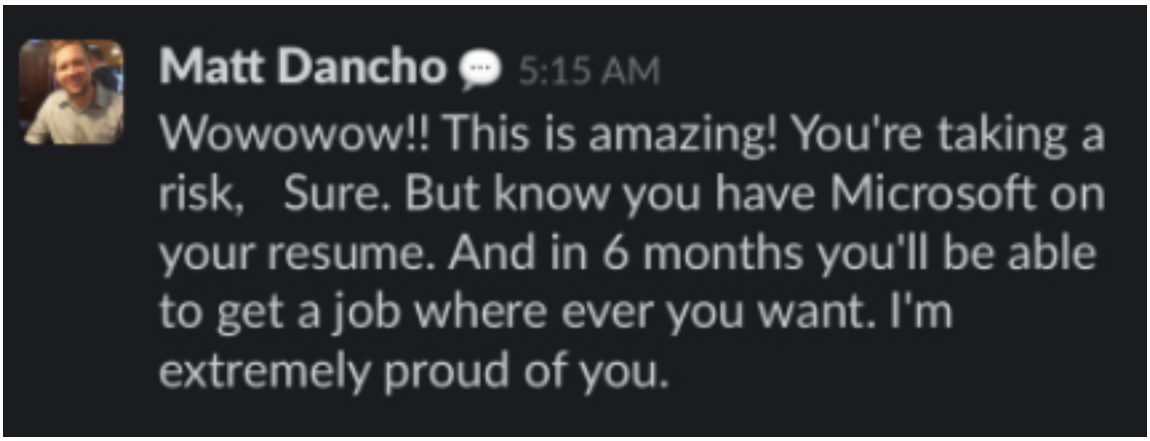
Plan	Skills
Machine Learning	Supervised Classification, Supervised Regression, Unsupervised Clustering, Dimensionality Reduction, Local Interpretable Model Explanation - H2O Automatic Machine Learning, parsnip (XGBoost, SVM, Random Forest, GLM), K-Means, UMAP, recipes, lime
Data Visualization	Interactive 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 skills needed to become a data scientist

In late September 2021, David was a Research Analyst with Texas A&M University.

In March of 2022, less than 6-months later, he has 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.

I told him that's just the beginning.



How was David able to get a job at an elite company like Microsoft?

What skills was David learning that he could transition so quickly?

The rest of this post will show you how David did it. This post includes research and 2 surveys all to answer questions like…

- What skills separate data scientists like David from everyone else
- How to pick a language (and your choice may surprise you)
- How to learn the skills
- Why this approach to learning skills works

1. The skills that separated David from everyone else

If you want to become a data scientist, you'll need to learn how to generate value for your organizations. I've written about how data scientists create value for organizations here. But, in general, you complete a process (called the data science process), which involves learning these data science skills.

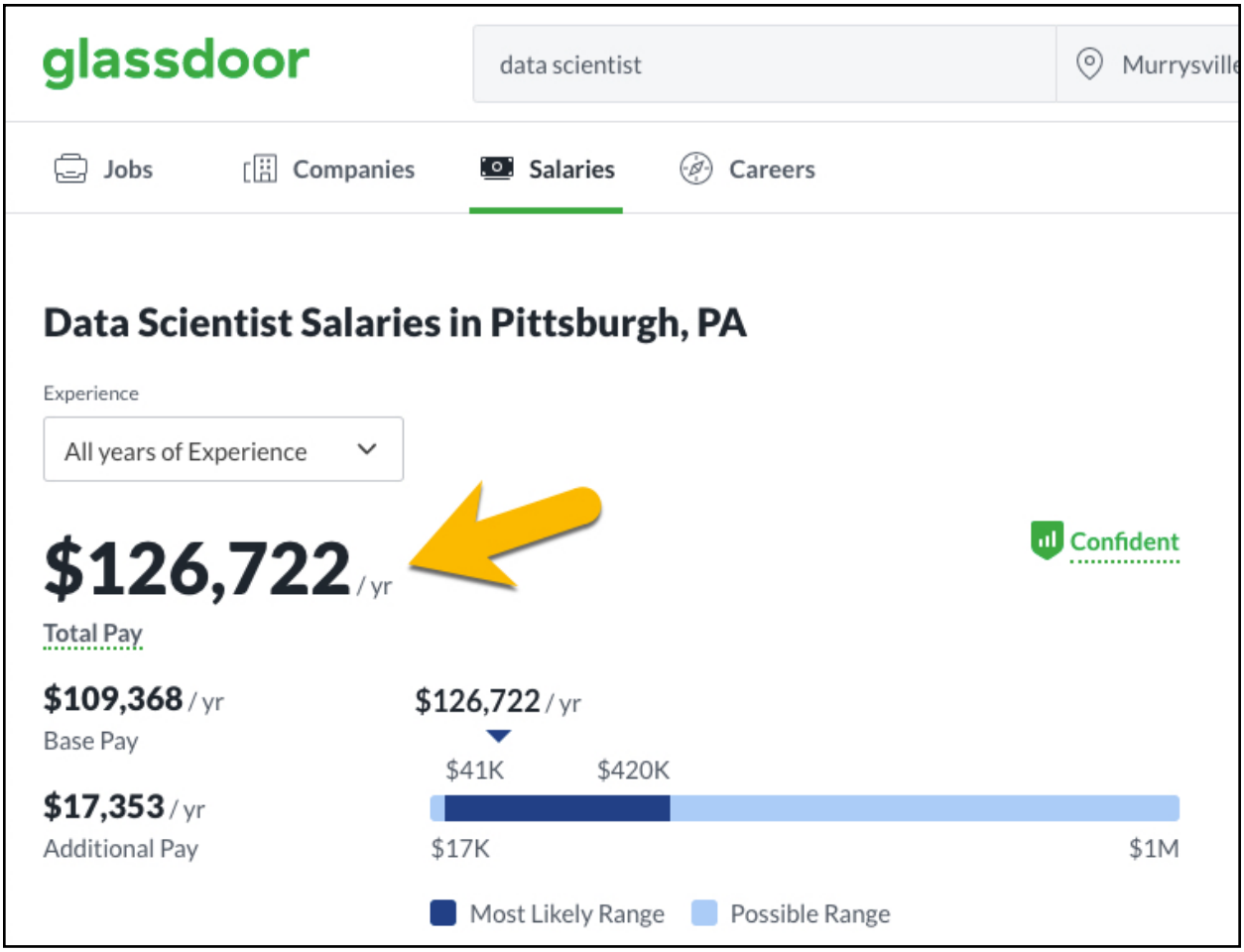
Plan	Skills
Machine Learning	Supervised Classification, Supervised Regression, Unsupervised Clustering, Dimensionality Reduction, Local Interpretable Model Explanation - H2O Automatic Machine Learning, parsnip (XGBoost, SVM, Random Forest, GLM), K-Means, UMAP, recipes, lime
Data Visualization	Interactive 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 skills needed to become a data scientist

David learned these skills and was able to convince employers to hire him. The result was an instant $50,000 increase to his salary and not to mention is in a career that he's super excited about.

The data science dream

Further to my point, according to Glassdoor, learning these data science skills can turn into a $126,722 career (if you live in Pittsburgh PA like I do, I encourage you to check your own locale).



Glassdoor: Data Scientist Earnings in Pittsburgh, PA

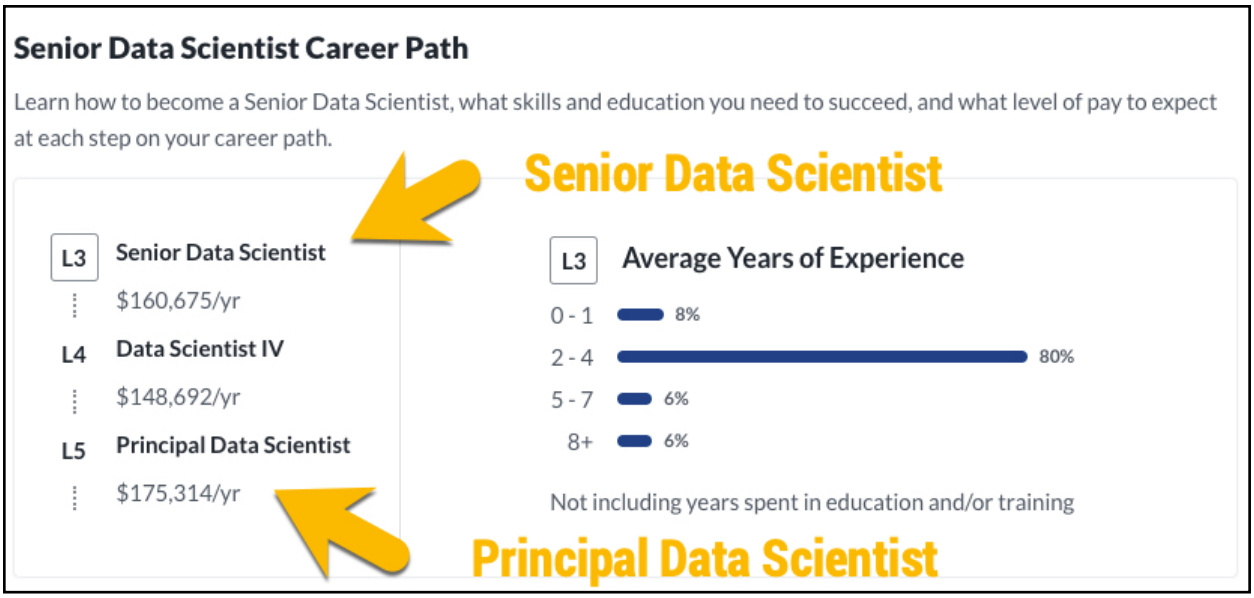
But that's just the start.

Like I told David, your career will accelerate.

What's after data scientist?

After data scientist comes senior data scientist.

Here's what the career path looks like for a Senior data scientist in Pittsburgh PA.



Glassdoor: Senior Data Scientist Earnings in Pittsburgh, PA

Now I know what you're 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 isn't hard. It just requires:

- **Motivation:** You'll need to dedicate about 10-hours per week
- **A plan:** You'll learn from David and several others in this

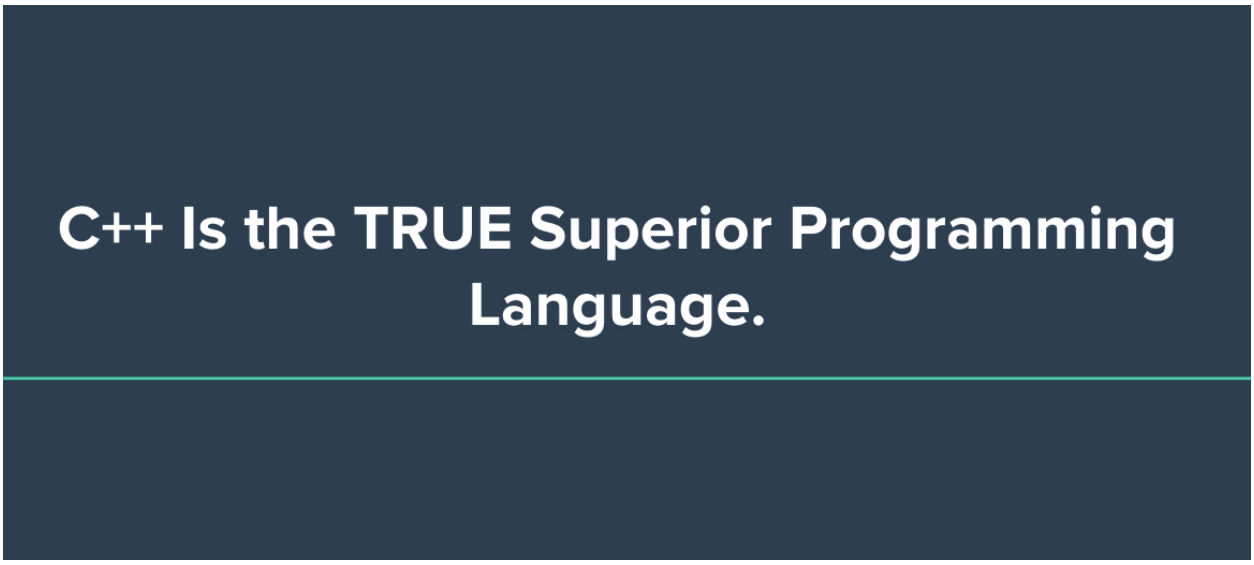
You'll need to start by picking a language.

2. How to pick a language

Who do you think is going to win this battle?

Well it's neither.

Because C++ is the true superior programming language.



I'm just kidding.

But in truth, it actually doesn't matter. You can succeed with both.

I know both.

I teach both.

But, if we want to really answer this question, we should tackle this like data scientists. You know, with data to support our decision.

So, let's tackle this like data scientists.

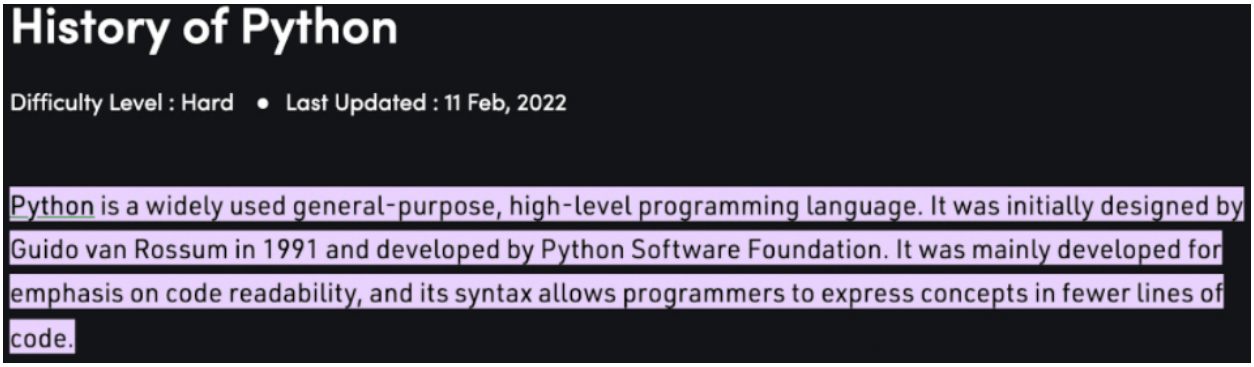
Here's how to pick a language

If I were picking a language for the first time, I would consider a few things:

1. How useful is the language for data science
2. The demand for the job market
3. The competition in the job market

How useful is the language for data science?

So 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.



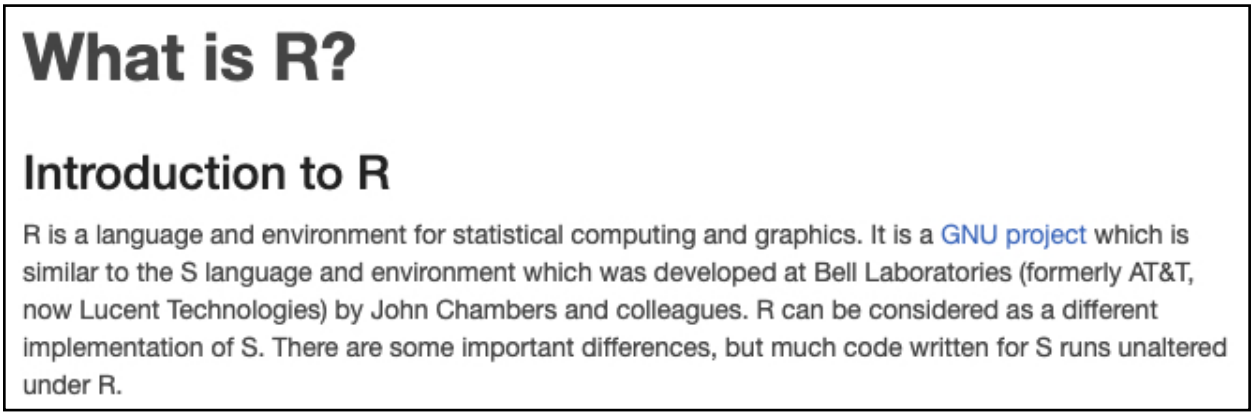
History of Python

Difficulty Level : Hard • Last Updated : 11 Feb, 2022

Python is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

GeeksforGeeks: History of Python

Meanwhile if you look at R, it was closely modeled on the S language for statistical computing and graphics.



What is R?

Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R-project: What is R?

So 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.

But I wasn't satisfied with that, so I dug a little deeper. Here's what I've found.

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

- **Python:** 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 well developed tools for business analysis and data science. Strong in everything except deep learning. But, deep learning is rarely used. And when you need deep learning or extra APIs, you can integrate R with Python.

So I'm going to give this one to R.

The demand for the job market

Next is demand for the job market for Python and R. Currently there are 21,271 Data Scientist jobs for Python. A

And, there are 8,713 Data Scientist Jobs for R.

So for every 1 R data science job there are 2.4 for Python.

I'll give this one to Python.

The competition in the job market

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

- **Python:** There are over 8,000,000 people that know python (and that number is growing fast)
- **R:** It's estimated that 250,000 to 2,000,000 people that know R and that number is also growing fast.

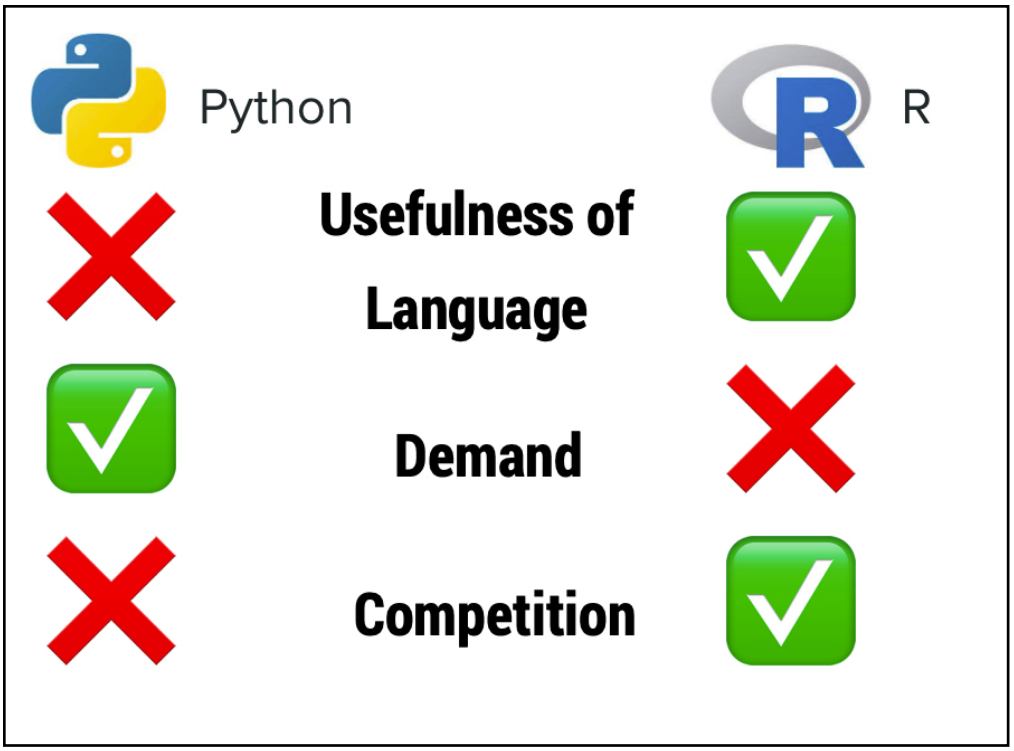
So for every 1 R user there are potentially 4 to 32 more python users.

So R positions are **less competitive by 10X or more**. Dang!

This one clearly goes to R.

R is a solid choice

R is a solid choice, and it's one of the reasons that students like David are able to quickly transition into a data science role. And keep in mind, you can always pick up Python later.



What about Excel?

At this point I always get a question, “what about excel?”

And my thought is this:

You can use any tool you'd like if it gets the organization results - R, Python, Excel, Tableau, PowerBI. All are great. BUT each has strengths and weaknesses.

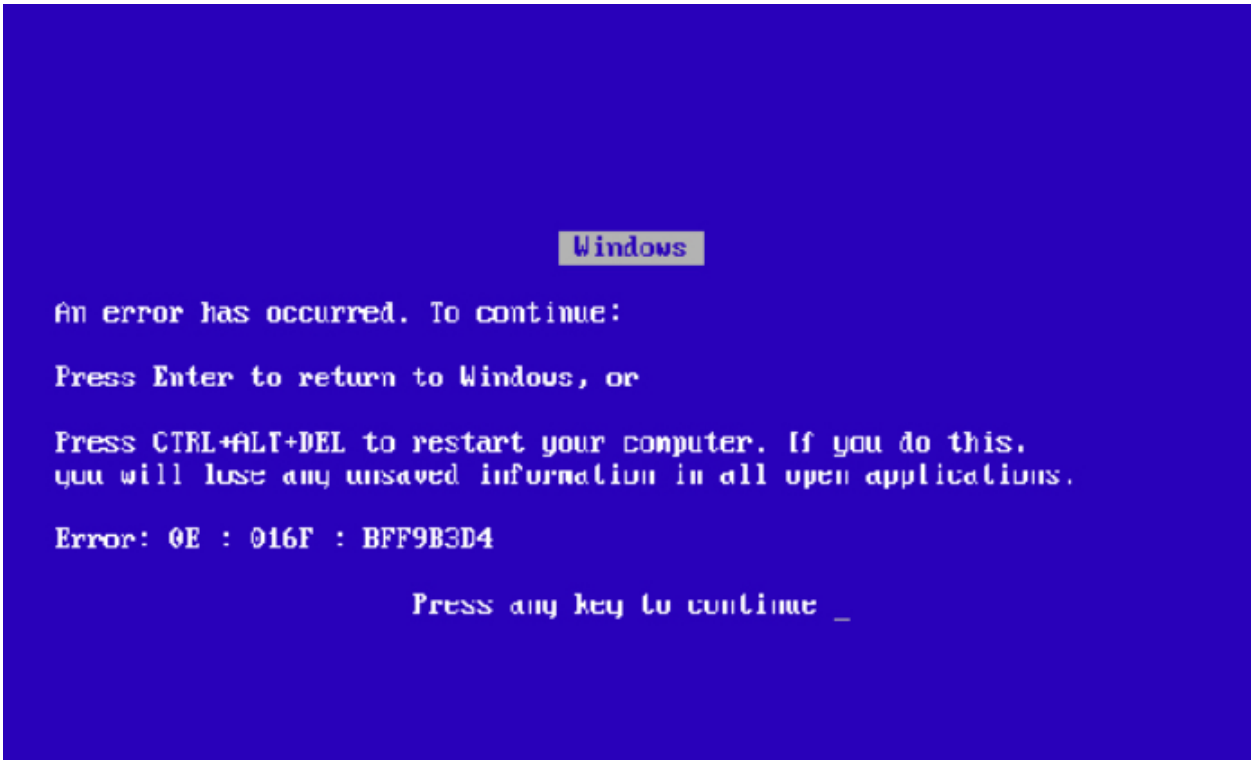
Excel is great as a communication tool:

- Everyone has it
- Business people like it.

Excel has the following limitations:

- Cannot do machine learning well. Machine learning is essential for modeling and explanations.
- Cannot handle large data well (maximum data size is 1-million rows, which is not very big)
- Functions are buried in cells, which leads to errors and difficult debugging.

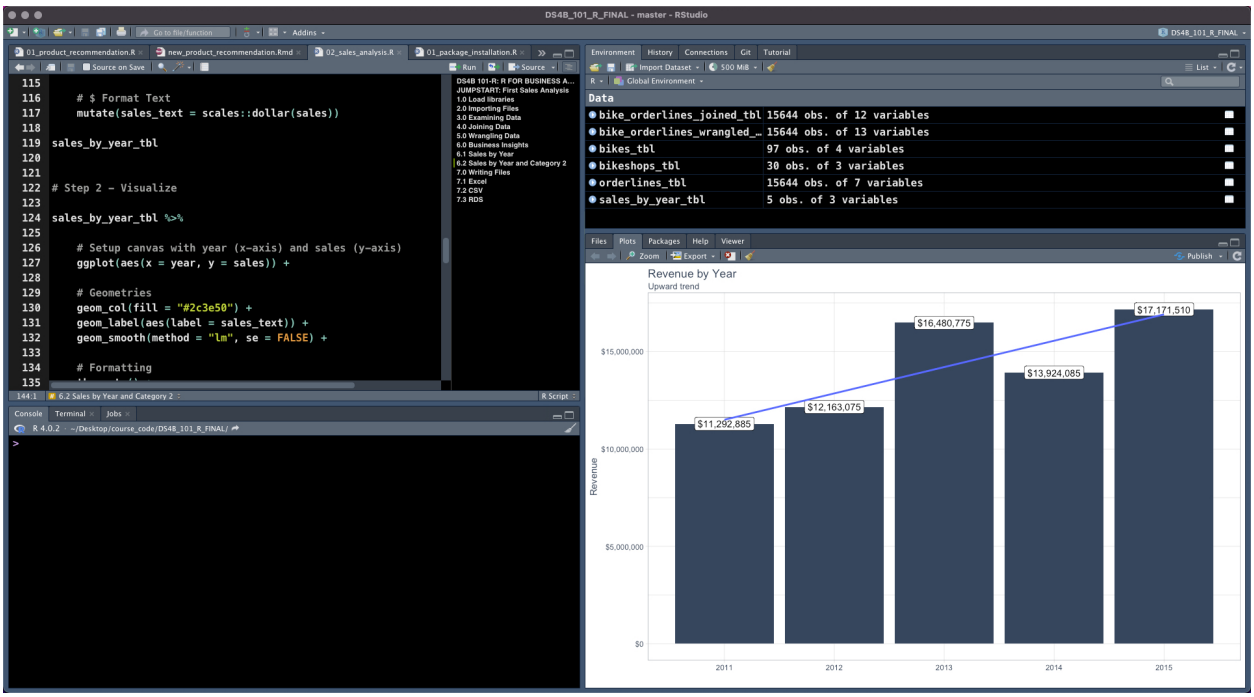
And yes, this is the **Blue Screen of Death**, and I used to get this constantly when doing data analysis in Excel.



So please use Excel wisely.

3. How to pick a development tool

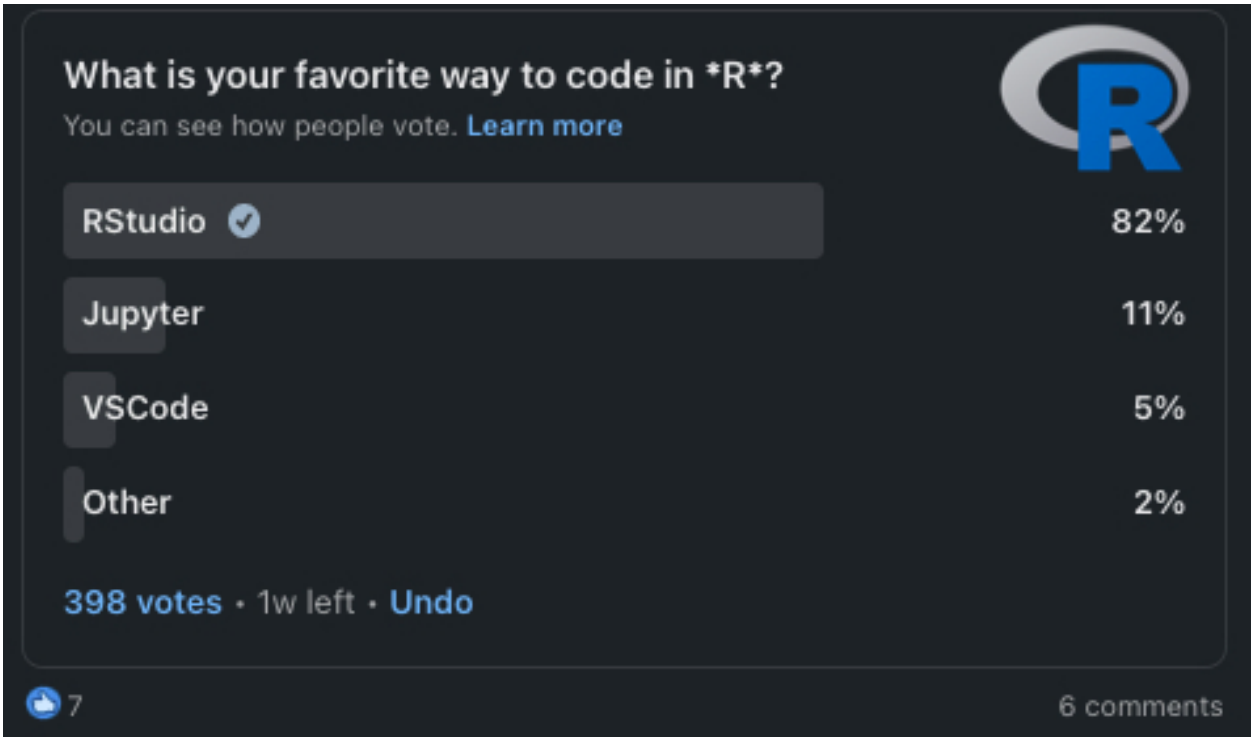
Next, it's time to pick an integrated development environment (IDE), which is just a fancy term for the thing I type code into.



The RStudio IDE: The thing I type code into

I ran a poll to see what everyone's using for R (and I did the same thing for Python too if you want to see those results).

Survey 1: What's your favorite way to code in R?



R Poll Results

Here are the results.

It's a landslide victory for RStudio.

So, if you are going to learn R, pick RStudio. Easy peasy.

Survey 2: What's your favorite way to code in Python?

Python Poll Results

I ran the same poll for python. And here's where it gets more complicated.

- About half enjoy coding in Jupyter
- A third like VSCode, and
- Some are even using RStudio to code in Python!

Keep in mind of my 61,000+ followers on LinkedIn, many are likely to be people who follow my content and therefore are interested in R programming in addition to python.

But still, it's not an easy decision for python users to pick an IDE.

In fact, I got a ton of comments for Spyder and half a dozen other random IDEs.

4. How to learn the 14 data science skills

Once you settle on a language and IDE, you're ready to begin the fun process of learning the skills to become a data scientist.

At this point you need a plan. Why?

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)

Data Science Skills

…Because your goal should be to get a data science job **as fast as possible.** The market is crazy right now. But, eventually the market will cool and you'll be outa-luck.

What about soft skills?

I always get this question at this point. I can hear it now.

"Matt, everything you've shown is technical skills. What about communication skills?"

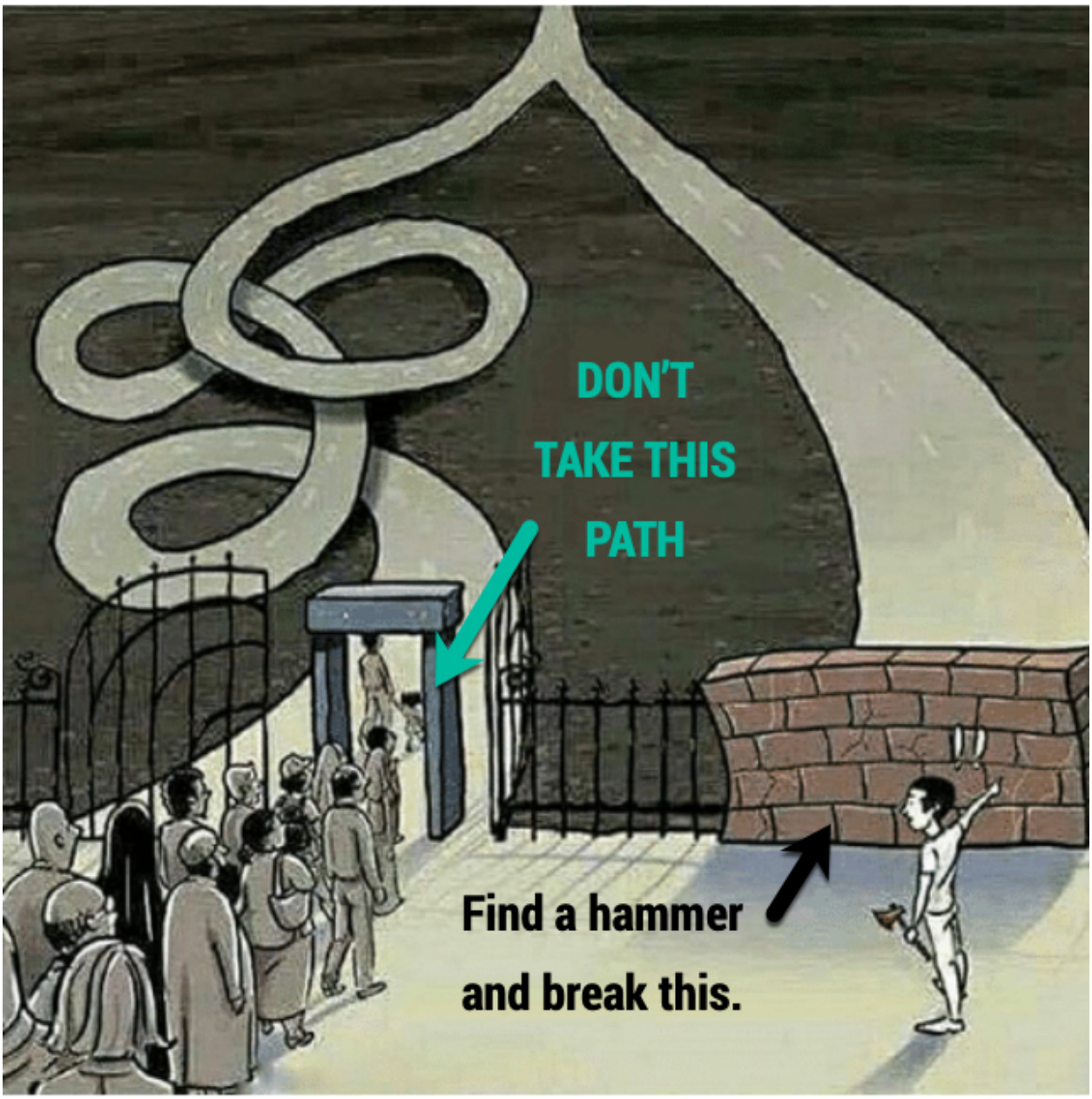
Yes - you absolutely need those too. But you've also been learning those all your life. And if you haven't, then add these 3 things to your arsenal:

1. Making a slide deck
2. Presenting your findings in a report
3. Being nice when you talk to people.

If you do those 3 things consistently, you will be promotable. And people will want to work with you.

Especially focus on #3 (Being nice).

The 3 learning paths (choose wisely)



There are 3 types of data science learning paths:

1. **Those that have 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-percent annual raise.
2. **Those that have a crappy plan.** They will take 5-years. But will eventually learn data science. They will also lose out financially because it took them sooo long to learn data science. 5-years at $125,000 per year when factoring in a low 3-percent raise = **loss of $664,000**. Ouch!

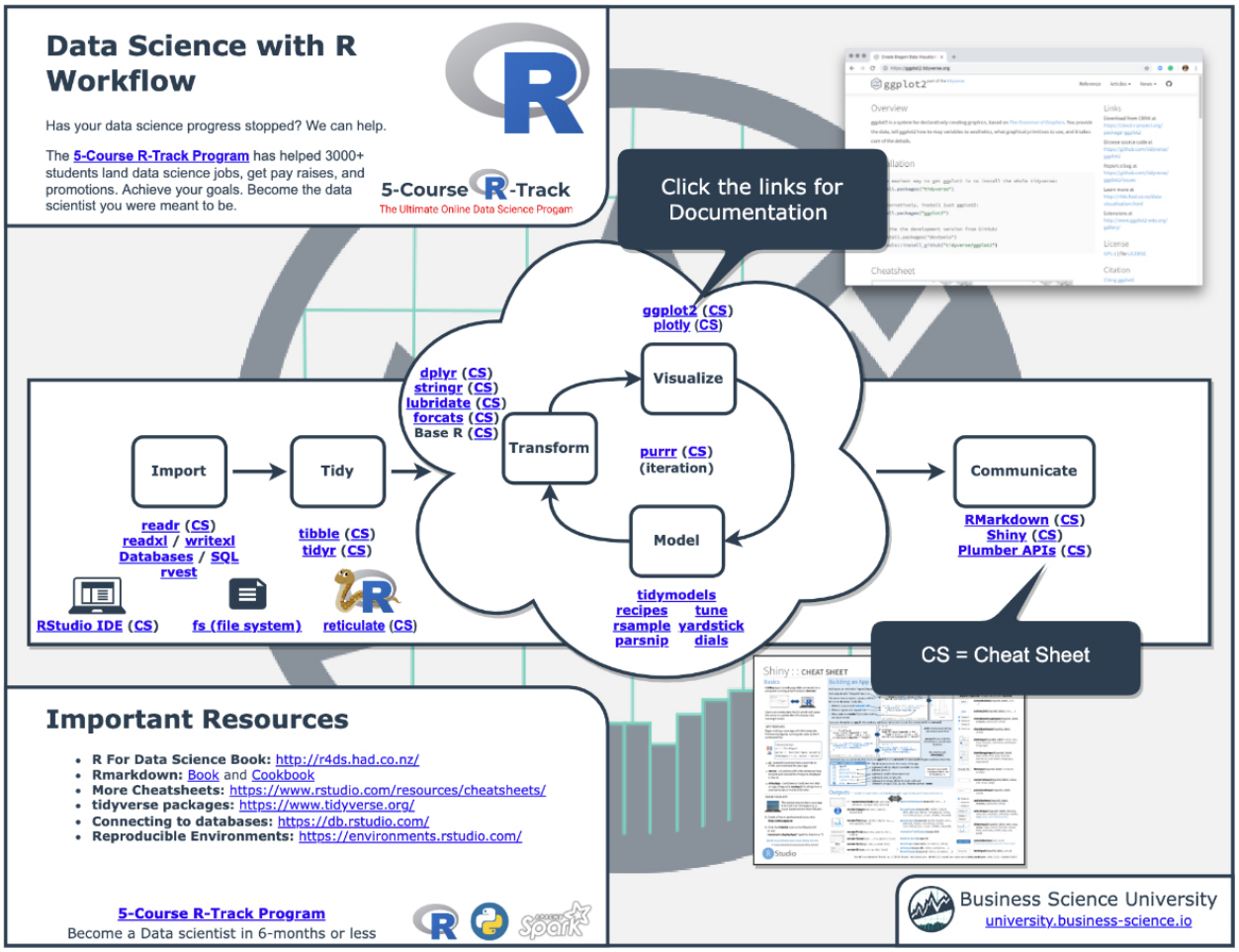
3. **Those that have an exceptional plan.** They are likely to be successful and can complete the transition in under 6-months.

Now, keep in mind, I actually had a pretty crappy plan. And it seriously took me 5-years. And it cost me a lot financially too. But whatever. At least I made it.

But, students like David have an exceptional plan. They made it in 6-months. And, it **involves cheating.**

It's OK to cheat…

And in the real world, to learn data science fast you need to cheat. What I mean is use a cheat sheet. Here's my R-Cheat Sheet that will help you learn the skills you need.



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

Here's how to cheat.

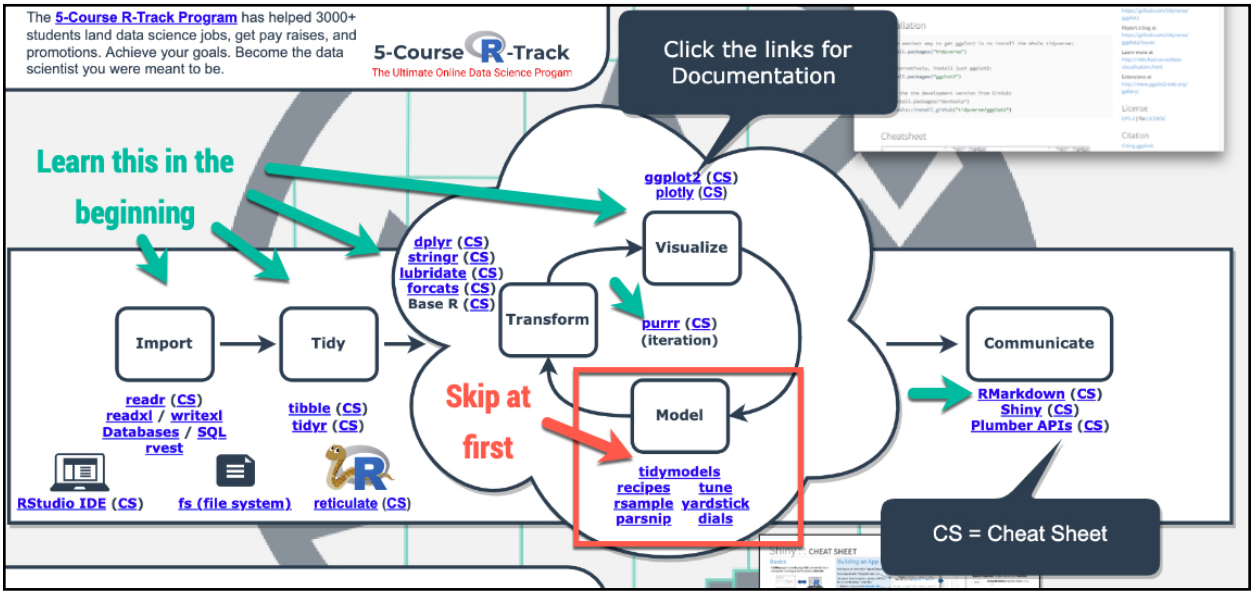
Learn the foundational skills first (save Machine learning for later)

Now I know what half of you are going to do.

You're going to jump right into Machine Learning. It's A BIG MISTAKE. Don't do that.

Instead learn these skills.

Rather, learn the foundations.



These are the skills you are going to use every day. I call them **80/20 skills.**

They are the skills that help you early on in your process.

Things like:

- **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
- **Reporting:** Making reports in interactive HTML and staticPDF formats

It' the honest truth. Listen, if you focus on these core foundational skills, it will make machine learning so much easier.

How to learn modeling (and machine learning)

Now it's time to take the training wheels off. Machine Learning!

Now you're probably thinking…

What about maths, stats, and algorithms?

At this point, a logical question is - “What about maths, stats, and algorithms?”

Here are my two cents.

The Popular Opinion: Take 5-years and study theory, maths, learn how to code algorithms from scratch.

The Smart (Fast) Way: Learn in tandem why you apply machine learning in projects

The only way I’ve ever been successful with learning new algorithms is by experimenting and applying.

I’m talking about *actually* applying data science to projects I’m working on.

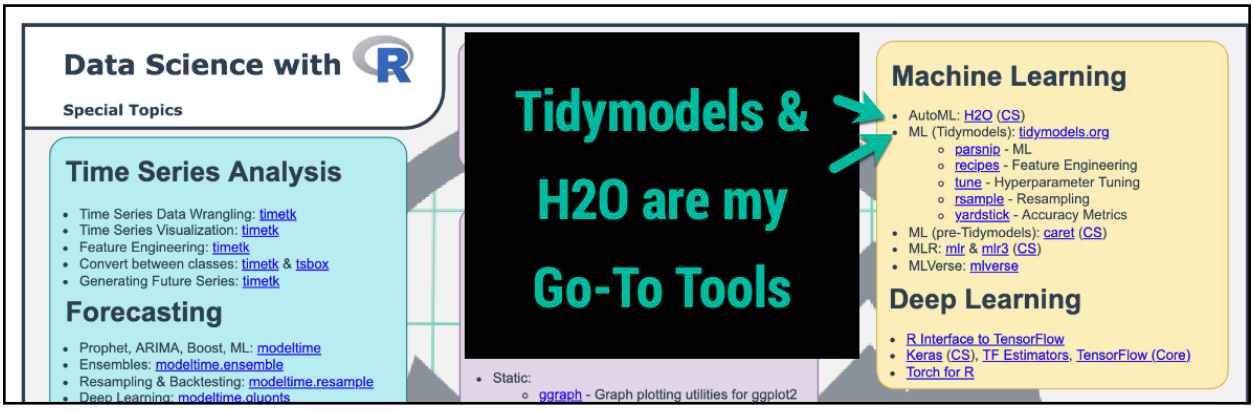
The process involves:

1. applying machine learning to problems,
2. experimenting with different algorithms, and
3. seeing the results on real applications.

If you do this on real projects then you will in fact learn maths, stats, and algorithms.

What machine learning tools should I learn?

If we head on back to my cheat sheet, on page 3 you’ll find links to my goto-machine learning tools.



I’m a big fan of two packages (or ecosystems):

1. **Tidymodels:** I use this for making adhoc models and then explaining

2. **H2O:** I use this for automatic machine learning and in production

Another (extremely important) skill is feature engineering.

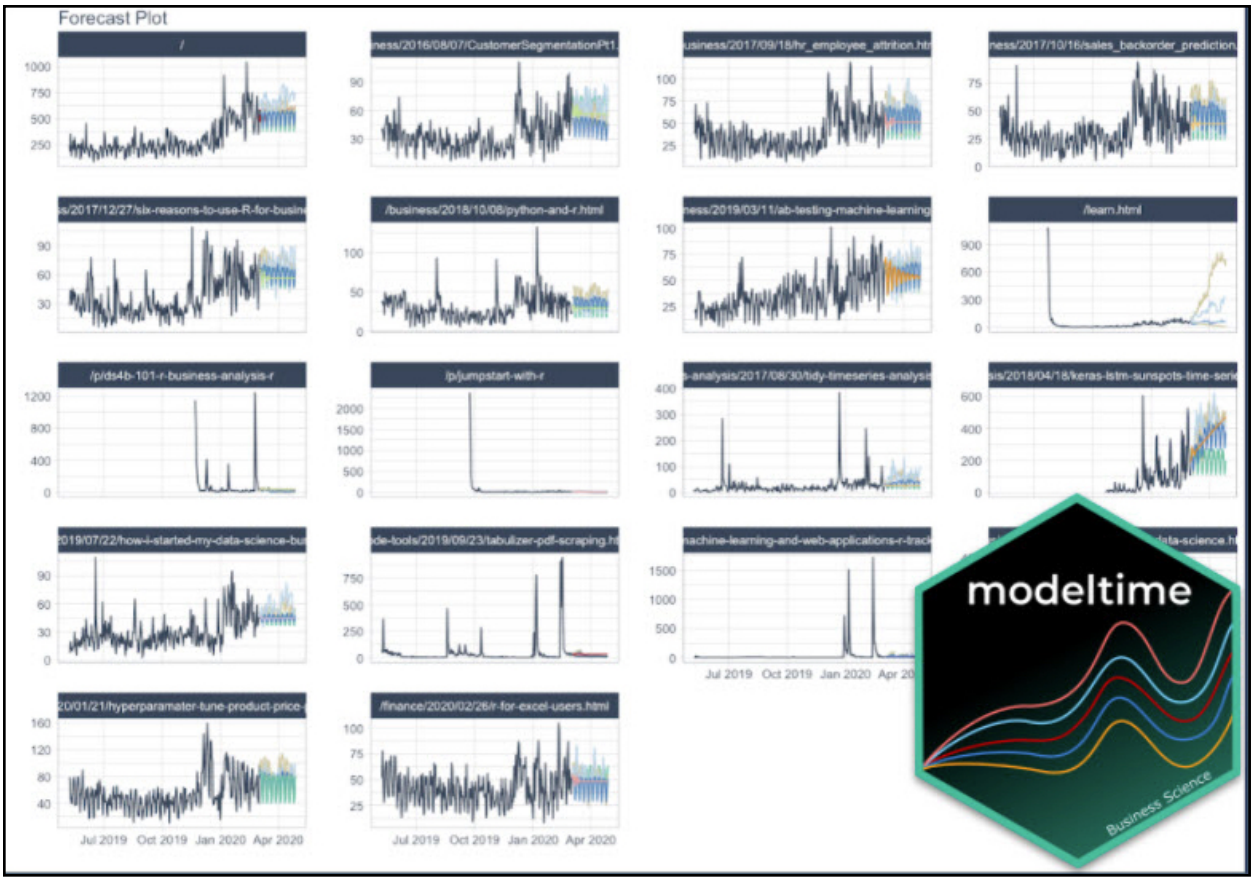
- **Recipes:** Has preprocessing tools to transform numeric data and create features from date, time, and text data.

Time series is a money saver

Next, if you are interested in becoming *insanely* valuable to your organization.

Then learn time series.

Organizations are fans of saving money. An if you can predict the future, then chances are you are going to be very valuable to your company. Why?

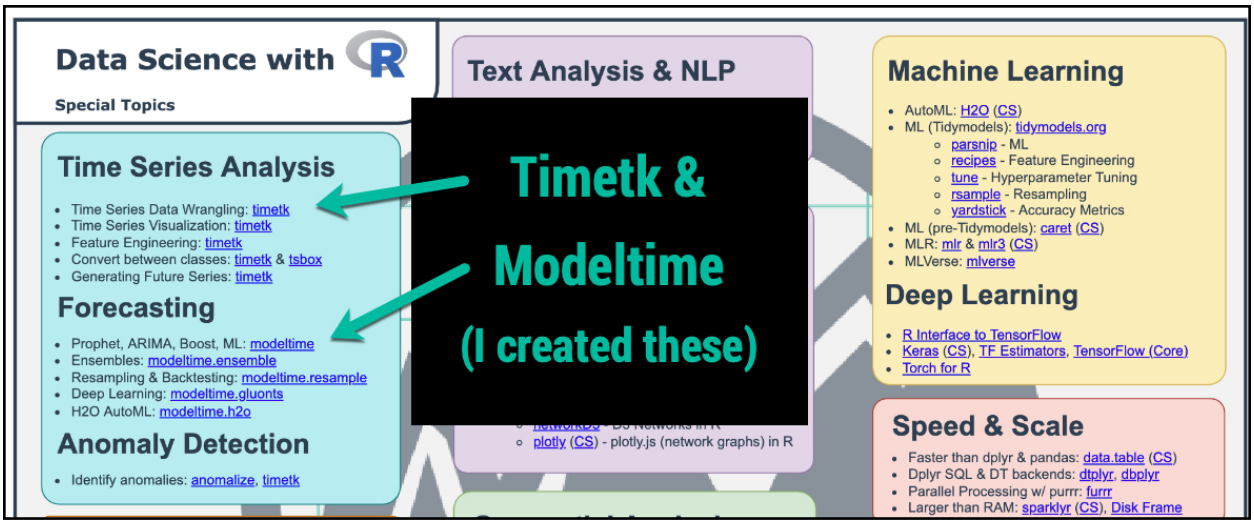


A 5% improvement in a forecast can save a company like **Walmart $50,000,000 each year**.

So Walmart will pay an arm and a leg for someone that can help them improve that area.

What time series tools should I learn?

Let's head back to the cheat sheet, and check out page 3 the "Time Series Analysis" and "Forecasting" section.



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.

How to take models into production (what the heck is production?)

Your model is worthless…

Until someone can use it to do something productive like…

Examples:

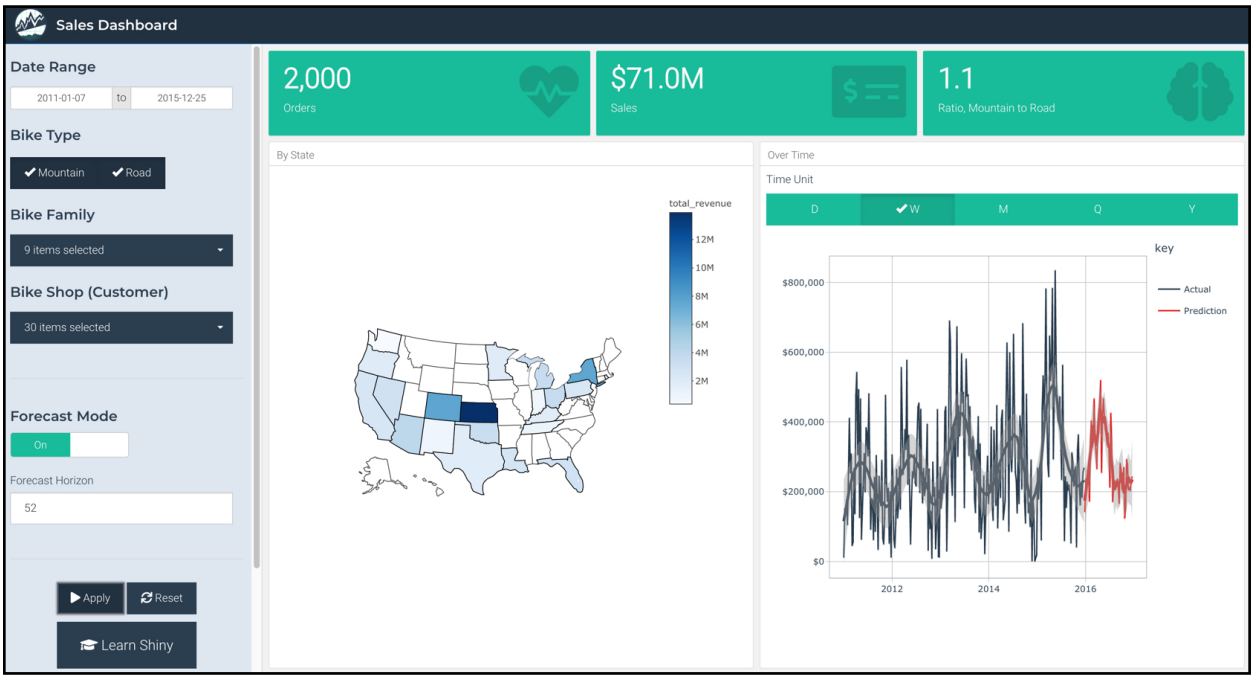
- Call a customer that is on the bleeding edge of unsubscribing because of high complaint volume
- Review more accurate forecast information before placing an $1,000,000 order for parts that could be unnecessary

It's at this point that your hard work pays off. And you provide value.

But how do you give the decision-makers the help they critically need?

This is putting applications into production.

The Application



A Shiny Application

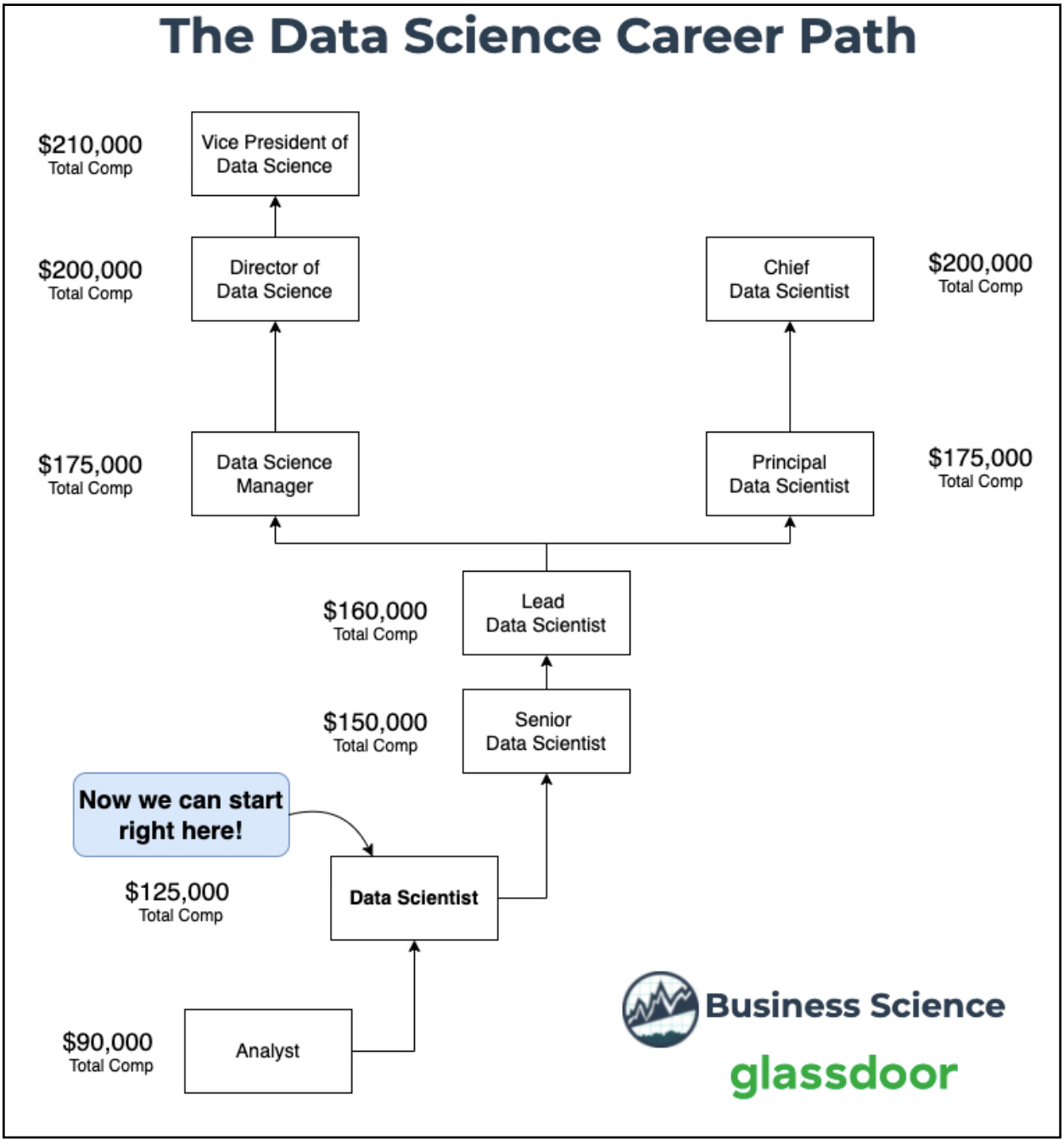
One of the truly amazing things is the ability to integrate models in to applications.

We can use applications to automate the analysis process.

And users can simply click buttons, use drop-downs, and get information, all without ever knowing that R (or Python) is truly running code behind the scenes.

The particular application shown above was made with a tool called `shiny`.

Chapter 2:
What is the Career Path for a Data Scientist? (From $75,000 to $150,000 salary in 1-year)



The Data Science Career Path

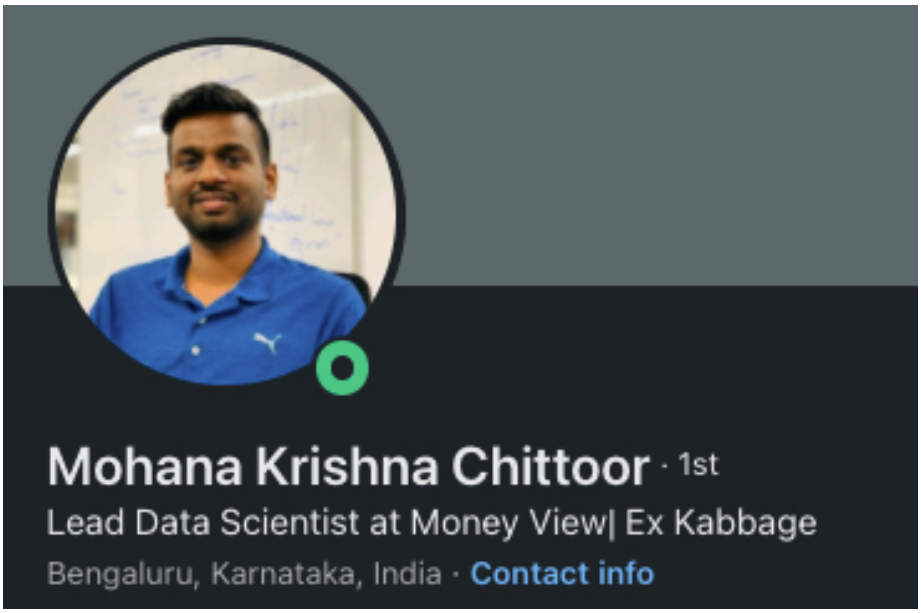
It was 2018 and Mohana was a struggling business analyst. He'd been getting a measly 3.5% raise since he joined his company.

Then in 2019 he got 1 raise (10%)…

In 6-months Mohana got another raise (this time 26%)…

And, then in just another 2 months (40% hike).

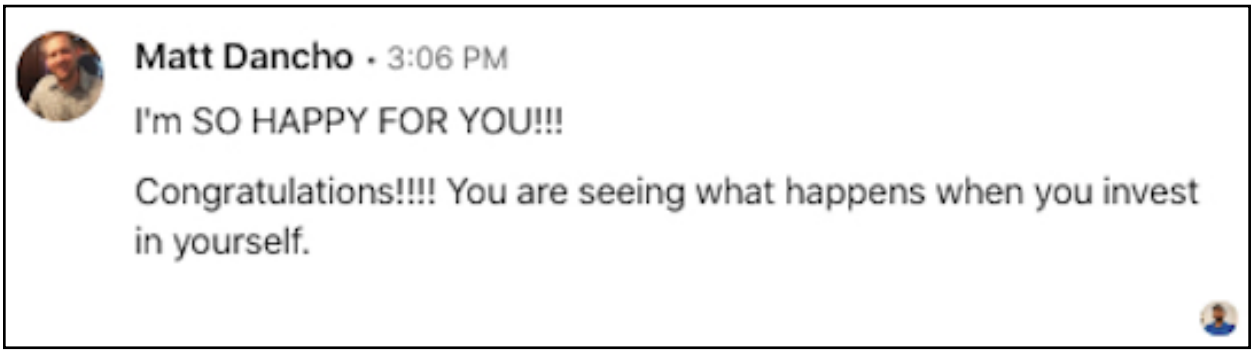
In total, in under one year Mohana got a total of 94% increase in his salary!



Today, Mohana is the Lead Data Scientist, at a Company called Money View - one of India's fastest growing startups in India 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 the move into a new phase of startup growth.



I told Mohana how happy I was of him.

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

What did he do to climb the career latter so quickly and land a job where ever he wanted?

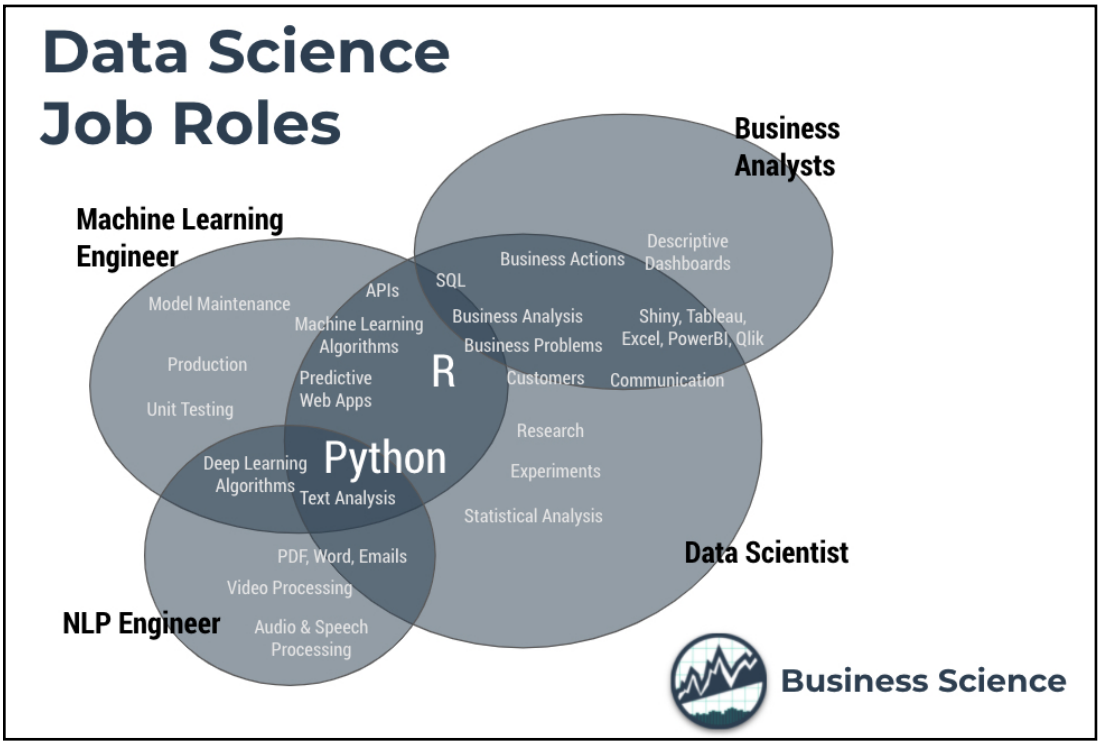
And how did he maneuver his career into to working at a leading startup, Money View, where he's now responsible for growing a team as a Lead Data Scientist?

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

This post includes career research from Glassdoor, and case studies from 2 data scientists that are growing their careers faster than I've ever seen anyone do it. In this post, we'll answer questions like:

- What data science roles exist? (and which to steer clear of)
- The career path for a data scientist (you start at $125,000)
- The skills needed to get promoted to Senior and Lead Data Scientist (you start at $150,000)
- 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)

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



Data Science and Analytics Job Roles

The first question that comes to mind when you are learning about data science or trying to figure your way through is which roles exist?

There are 4 main categories:

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)

We can see from the table that in general Data Science, Machine Learning, and NLP (Natural Language Processing) being compensated 40% more than Business Analyst positions.

So it's clear that you want to *avoid* Business Analyst (more on this in a minute).

We can also see that depending on your interest (general data science vs specialized NLP) that there tends to be *more pay for more specialization*.

But, we'll also see that pay will increase as your position changes from entry-level to senior/lead data scientist (coming shortly, hang in there).

But first, what do each of these data science roles do?

What do each of the data science roles do?

I've written extensively about the differences between each of the data science and analytics job roles here, but I'll briefly recap in this table:

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

We can see that it's more typical for the hard-core engineering disciplines to use Python versus the more business analytical disciplines to use R/ Python and Excel/PowerBI/Tableau.

If you are looking to move from Business Analysis to Data Science, we can see from the chart that you should add: R or Python to your skillset.

I explain in-depth exactly which skills are important to become a data scientist here.

What about Data Engineering?

The question I always get at this point is: *What about Data Engineering?*

Well, that's a great question. Let me explain why Data Engineering is not in the table.

Here's a typical conversation between a Business Analyst and a Data Engineer…

Typical conversation between a Business Analyst and a Data Engineer

(Day in the Life of a Data Engineer)

So, what Data Engineers do is make the jobs of the Business Analysts and Data Scientists much easier.

They do this by giving data scientists access to data in a nice tidy looking format that comes from what they call a **"data pipeline"**.

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

- Development of a data pipeline/API/microservice.
- Setup/Maintenance infrastructure
- Fixing bugs, improving code base, documentation

Data Engineers are valuable (if not essential) to data science program success

No question - Data Engineers are valuable.

But, since 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.

And quite honestly, I'm not the person to give you the pros and cons about data engineering.

This is why I'll point you to a data engineering guru like Andreas Kretz.

It's a mistake to 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.

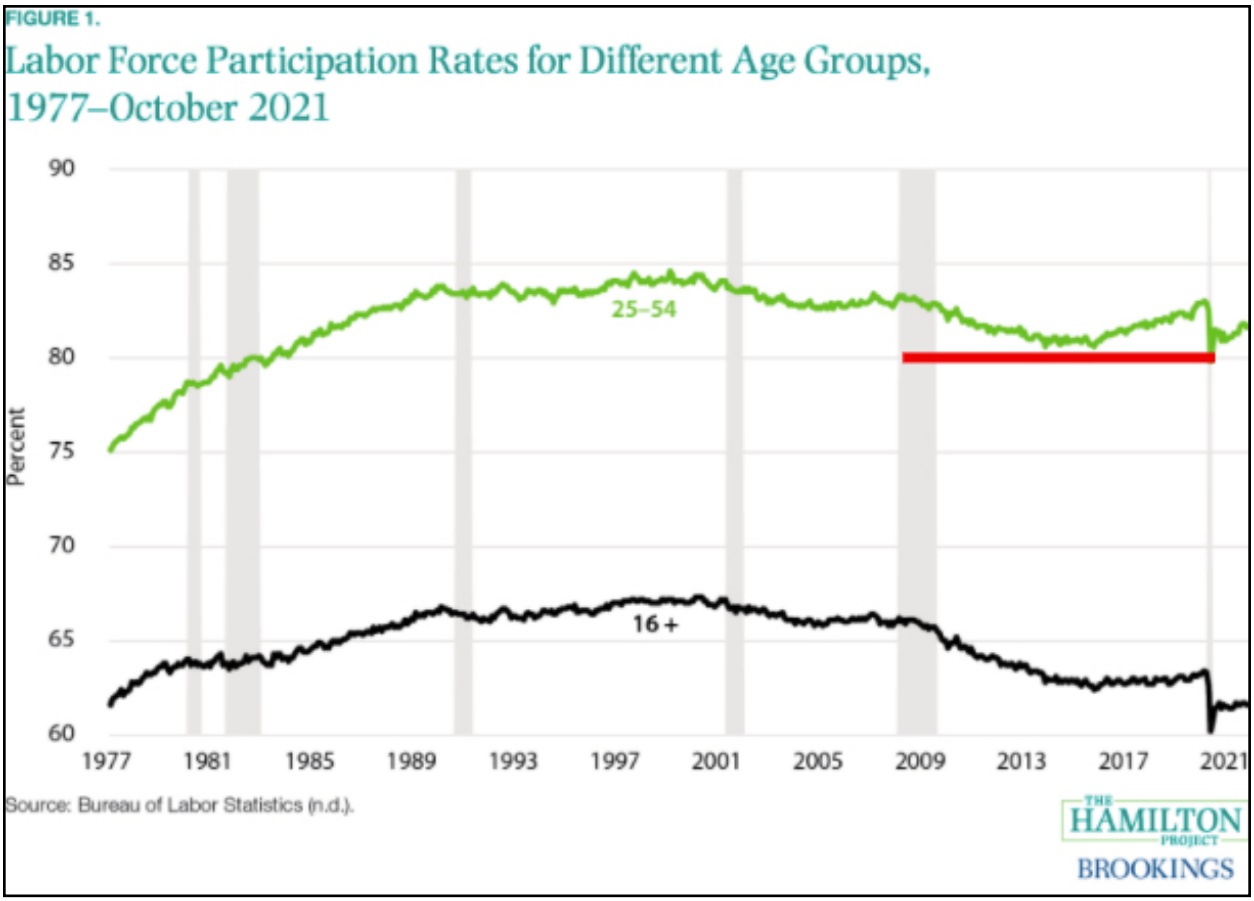
Avoid the business analyst position (RIGHT NOW!)

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.

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!

Because of COVID, there's a bullwhip-effect in the US labor market. Let me explain.

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.

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.

And the training time for most new hires is 1-2 years, so companies either had to offer higher salaries and benefits or be at risk of data scientists being poached.

Now you benefit.

Because you can SKIP the whole “business analyst -> data scientist” game and…

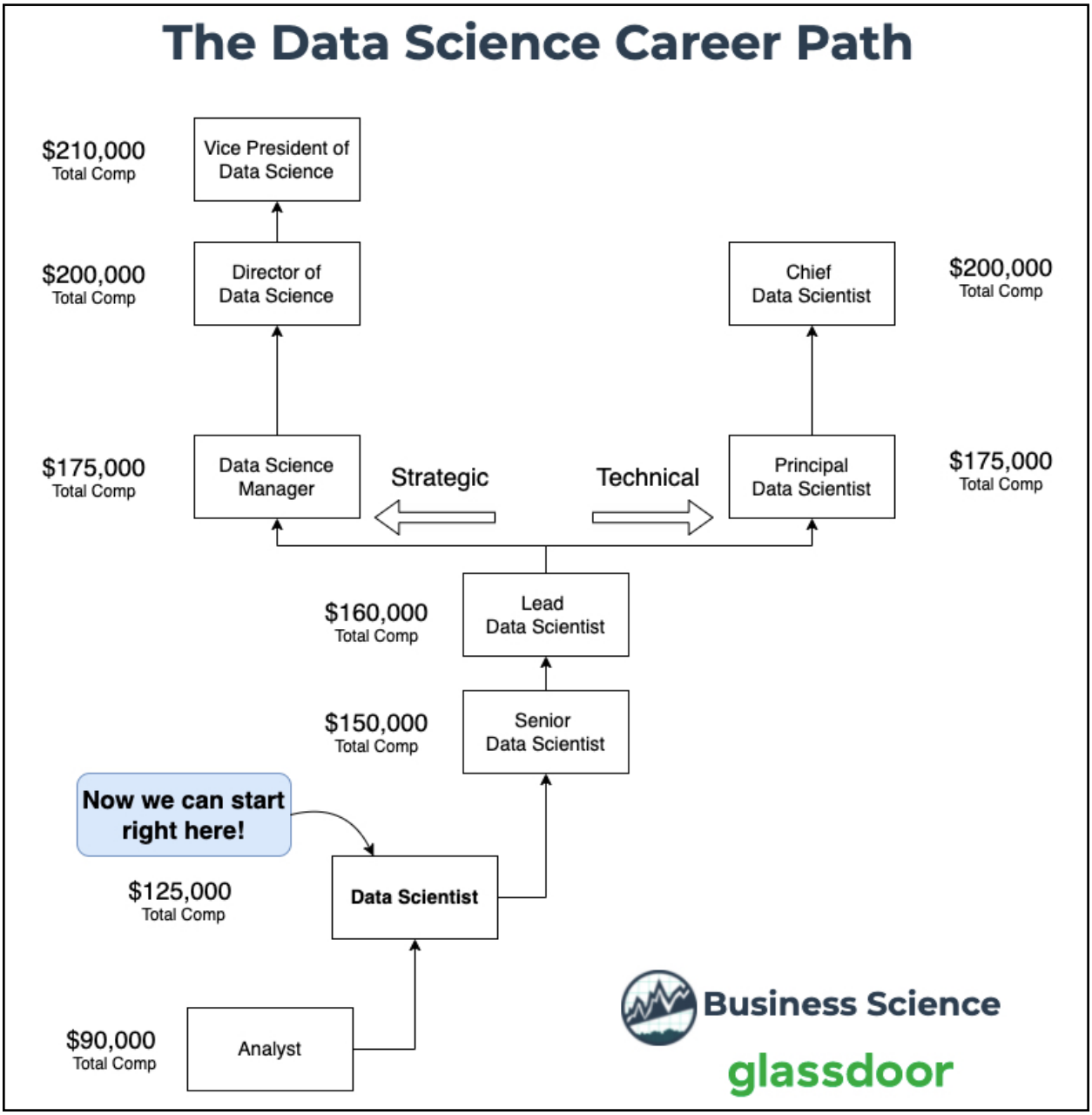
Jump right into Data Scientist roles.

How to jump right into data scientist roles

Now that you know data science is right for you, let’s show you how to get promoted to Senior and Lead Data Scientist (to make $150,000/yr).

2. The career path from Data Scientist (Start at $125,000/yr)

First, let's cover the career path for a data scientist, which for 85% of organizations 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

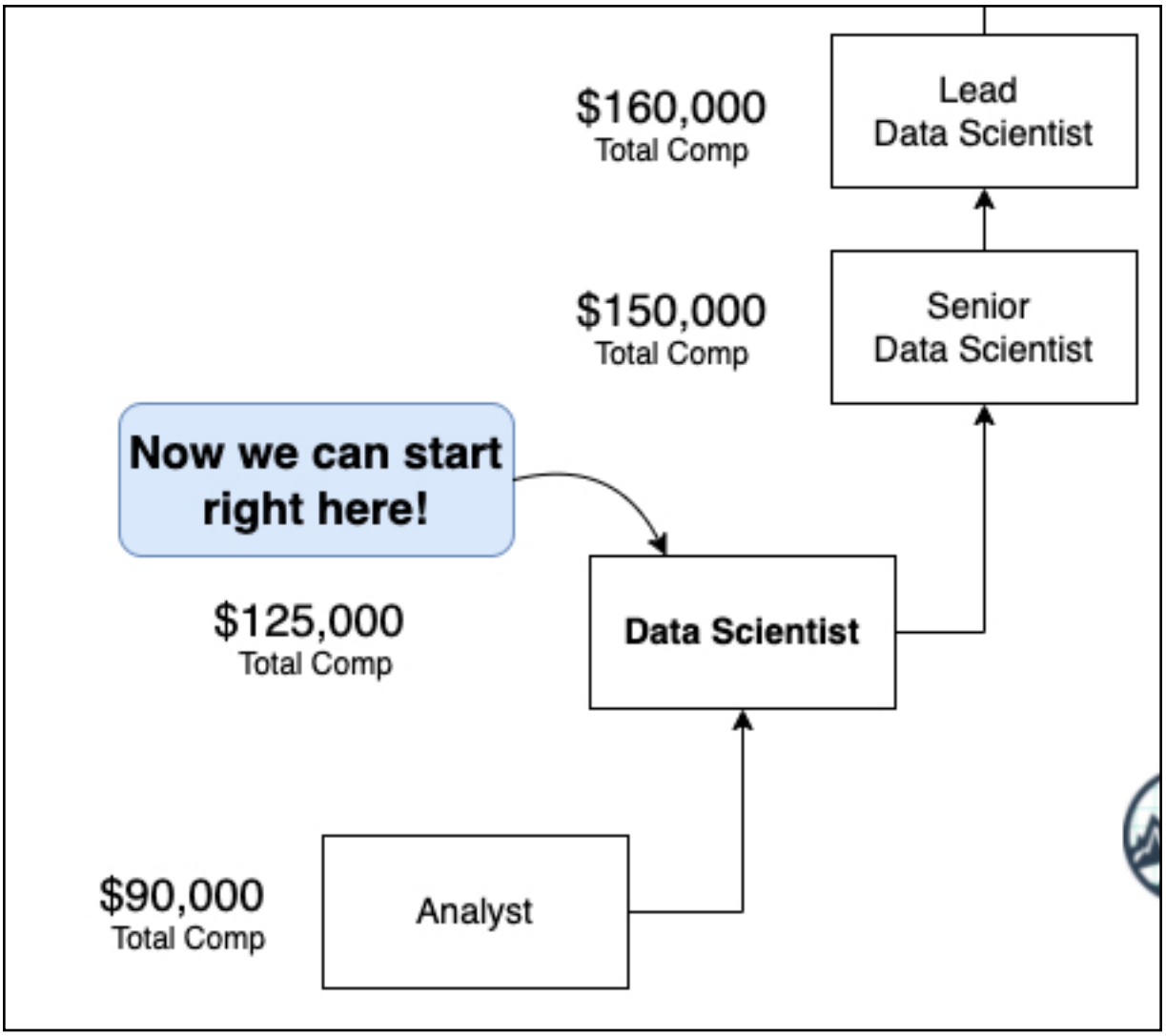
Data Science Career Path - Compensation

The things you need to think about are:

1. How to get to Senior / Lead Data Scientist as fast as possible ($150,000 - $160,000)
2. Pick a path - Strategic or Technical
3. Then keep repeating until you get to the top

The General Path

Most organizations have a general track which will take you to a Lead Data Scientist. The path looks like this:



The General Path

You start as a data scientist

You'll start at data scientist making around $125,000 per year in total compensation. All you need to do is get the skills listed here.

In fact, I even made a convenient cheat sheet to make it even easier (which you can download for free here).

And a Pro-Tip: Skip the Business Analyst position. Companies NEED YOU right now. Get the skills and go for it!

Next, you'll become a Senior Data Scientist.

These guys and gals make $150,000 per year in total compensation. And they are more experienced, probably have some big data experience, cloud experience (AWS, Docker, Git), and can do more advanced analyses when compared to the regular data scientists.

So 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

These fellas make $160,000 per year in total compensation. And 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 with others well, and build persuasive arguments.

Let's put this ALL together (comparing Senior/Lead vs Data Scientist)

If you really want to compare these 3 job general job roles, then I'll make it even simpler for you. Just learn these skills.

	Data Scientist	Senior Data Scientist	Lead Data Scientist
Predictive Analytics, Reporting, Modeling	✓	✓	✓
Advanced ML, Big Data, Web Apps, Cloud		✓	✓
Presentations, Good with Management, Persuasive			✓

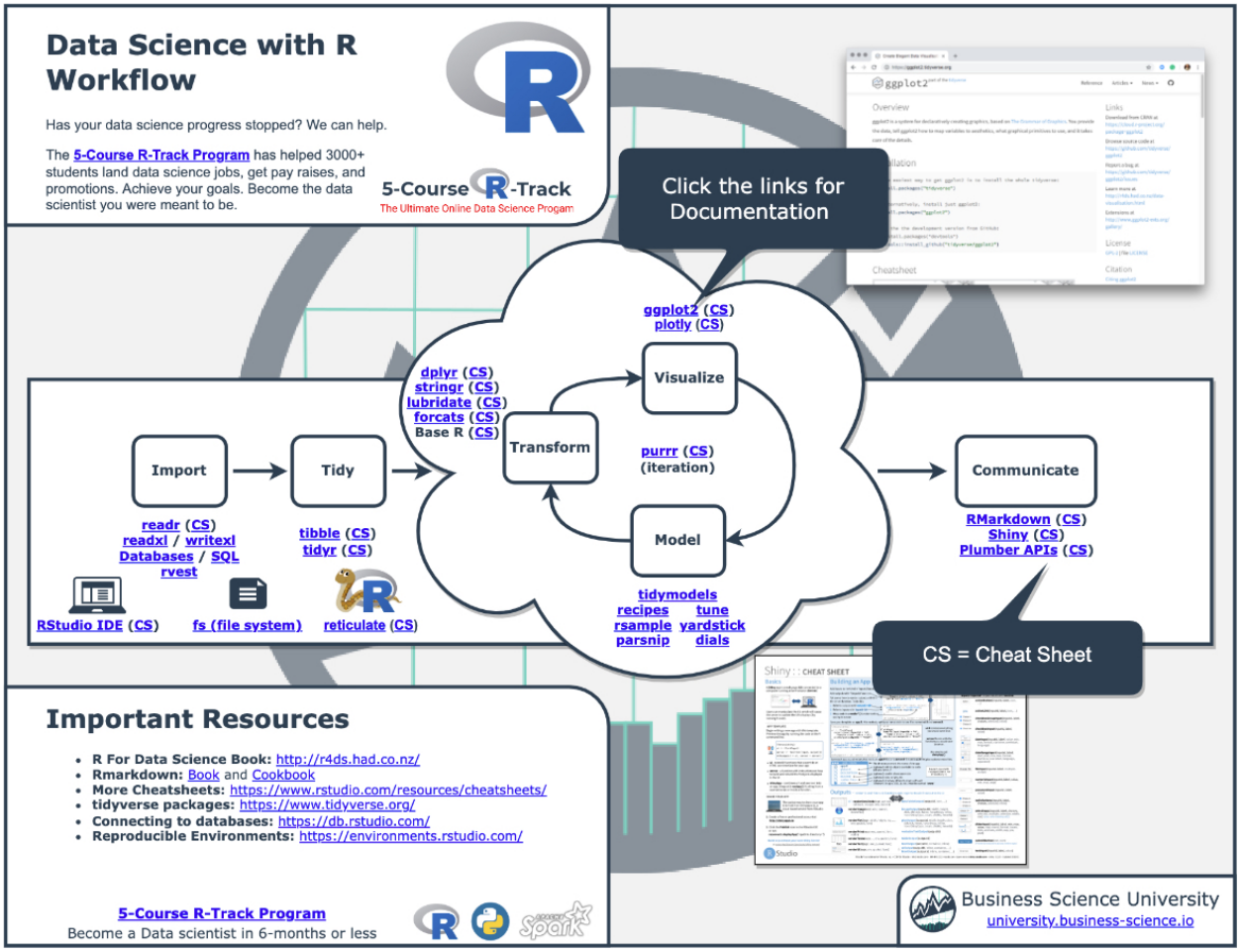
Comparing Senior/Lead vs Data Scientist

Now you are probably thinking…

3. What skills do I need to become a Senior/Lead Data Scientist ($150,000+ year)?

The easiest way is to cheat!

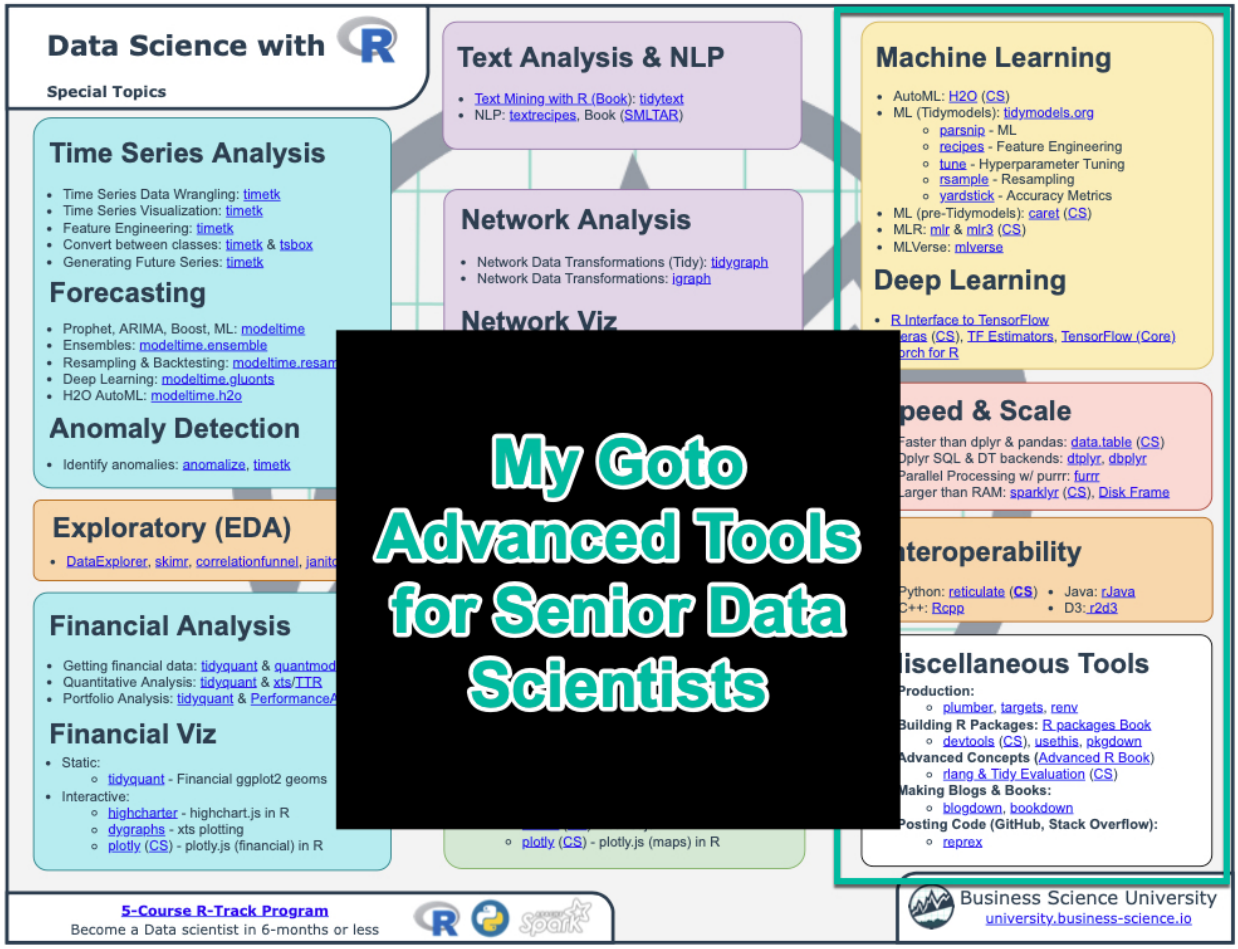
What I mean is use a cheat sheet. Here's my R-Cheat Sheet that will help you learn the skills you need to go from Data Scientist to Senior Data Scientist.



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 cheat sheet (page 3) you'll find links to my goto-advanced tools for Senior/Lead Data Scientists. (PS- Check out this article for the tools for Data Scientists if you are becoming a Data Scientist.)

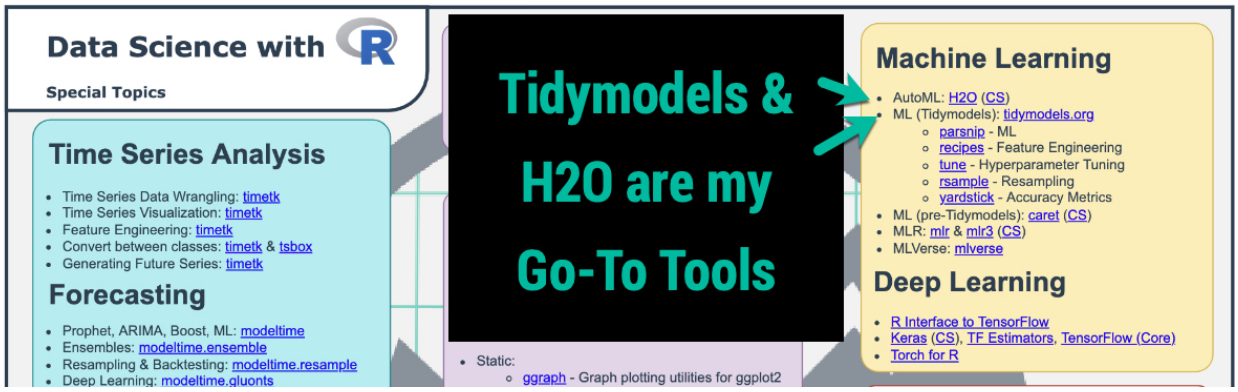


Matt's Goto Advanced Tools for Senior Data Scientists

Listen, 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

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's my personal favorites. I'm a big fan of two machine learning packages (or ecosystems):

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

Another (extremely important) skill is feature engineering. I'm always using THIS package to create features:

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

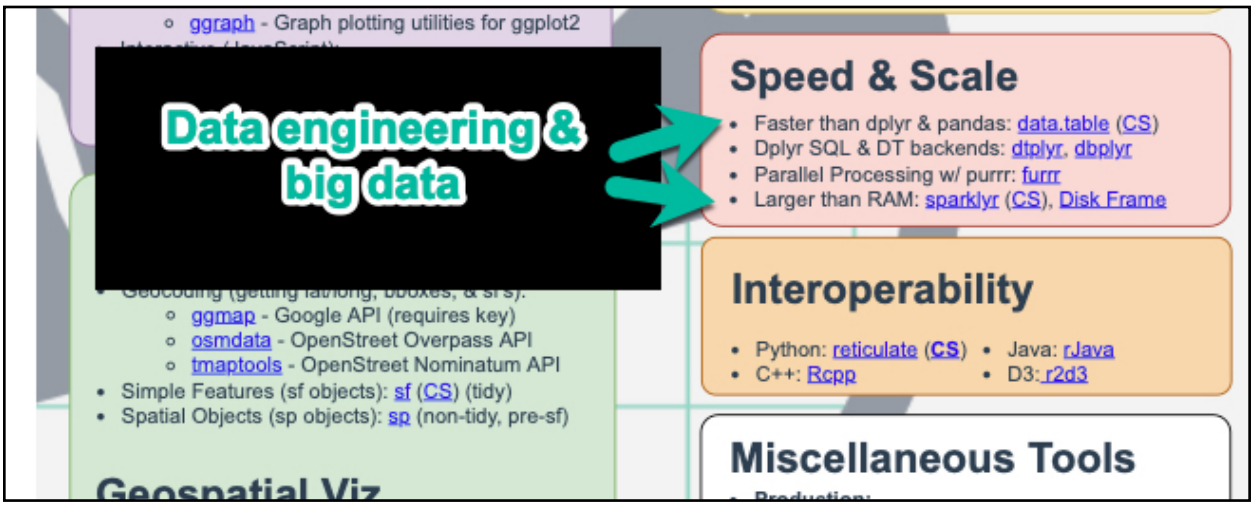
Next is hyperparameter tuning / cross validation. Here are my goto packages:

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

Data Engineering (Big Data)

Another key skill of the "big dogs" is "big data". This is where you work 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.

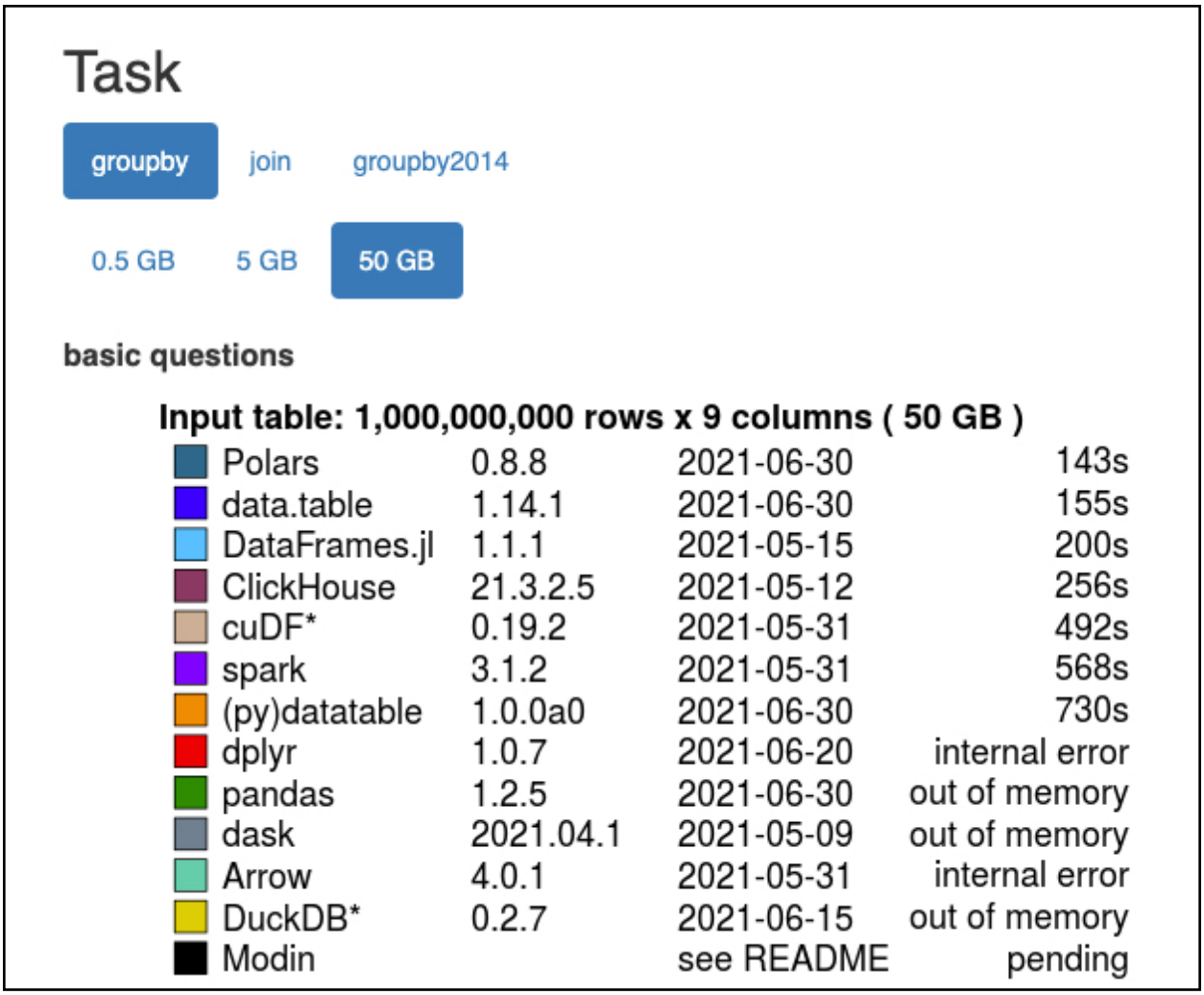


Data Engineering in R (Big Data Tools)

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".

First up is Data.Table

- **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 weird. 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.

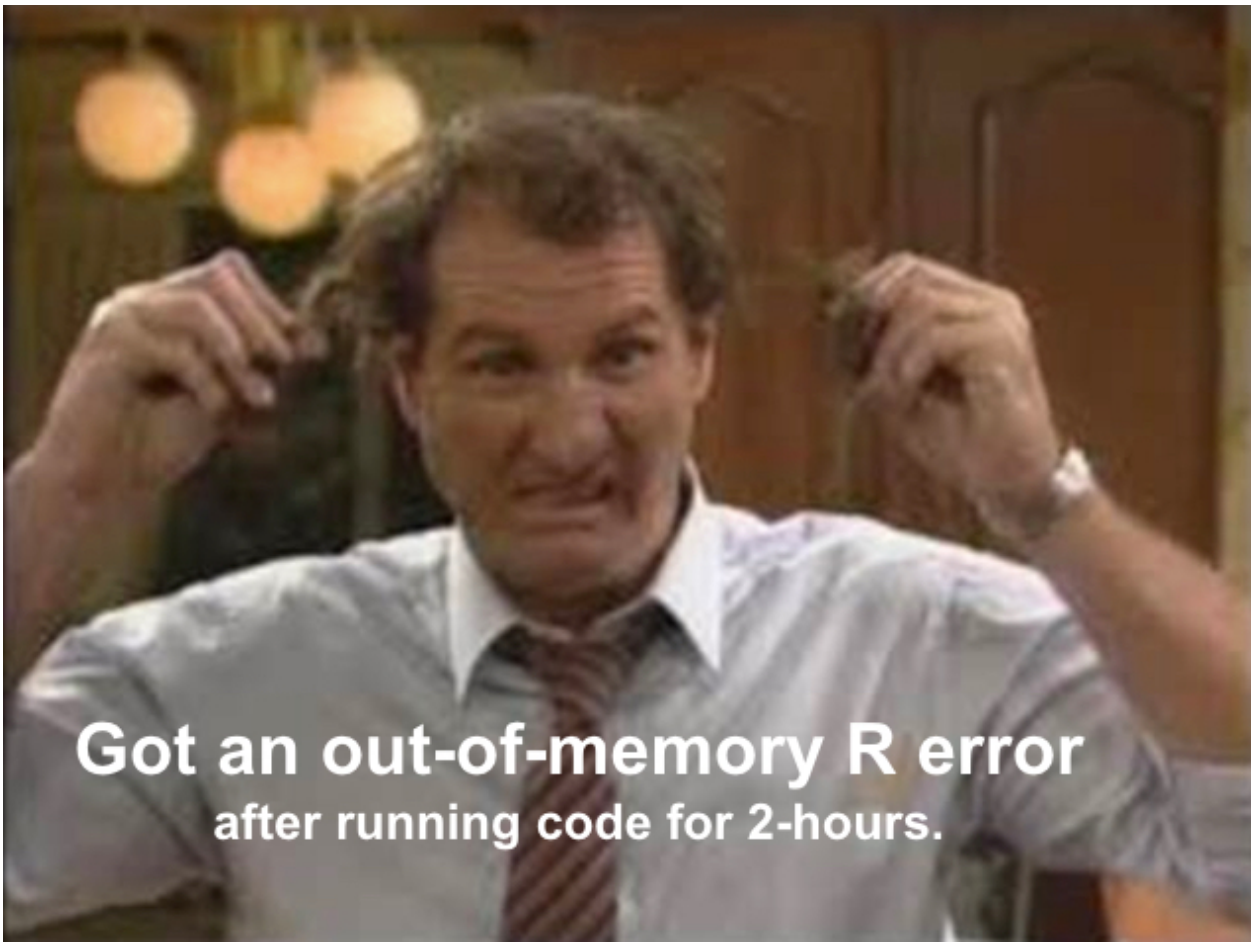
Next is databases

- **dbplyr:** This stands for “database” dplyr and allows us to run dplyr scripts on your database, which is mindblowing! Why? Because 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 😰

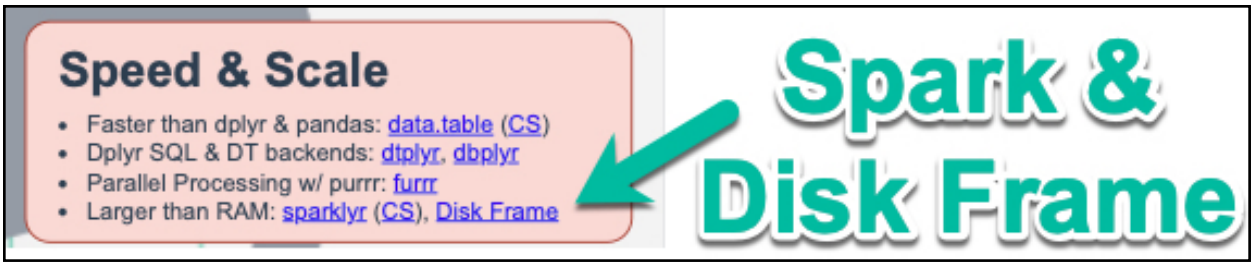
Now sometimes you're going to run out of memory right before a presentation.



This is what happened to *young* Matt. Before I knew about the next 2 package.

I'd run code for *my presentation tomorrow,* and I'd get an error 2-hours in saying something like "out-of-memory" or "vector can't be allocated." 😰

Fortunately, I'll help save your job (the way I eventually learned how to save mine). Here's how.



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.

Spark in R

- **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… I'll help. Here's my Spark in R Masterclass that I opened up for free. Normally these are only available through my Learning Labs PRO membership program, but I can't let you lose your job over an out-of-memory error. I wouldn't be able to live with myself.

Disk Frame (R's little big data secret)

Now, what happens if you don't have access to a Spark Cluster? Well, another AWESOME package is the little known `disk.frame`.

- **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.

Finally, there's Python in R

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

Wait, what?!

Yep, you CAN use Python in R. Here's how.

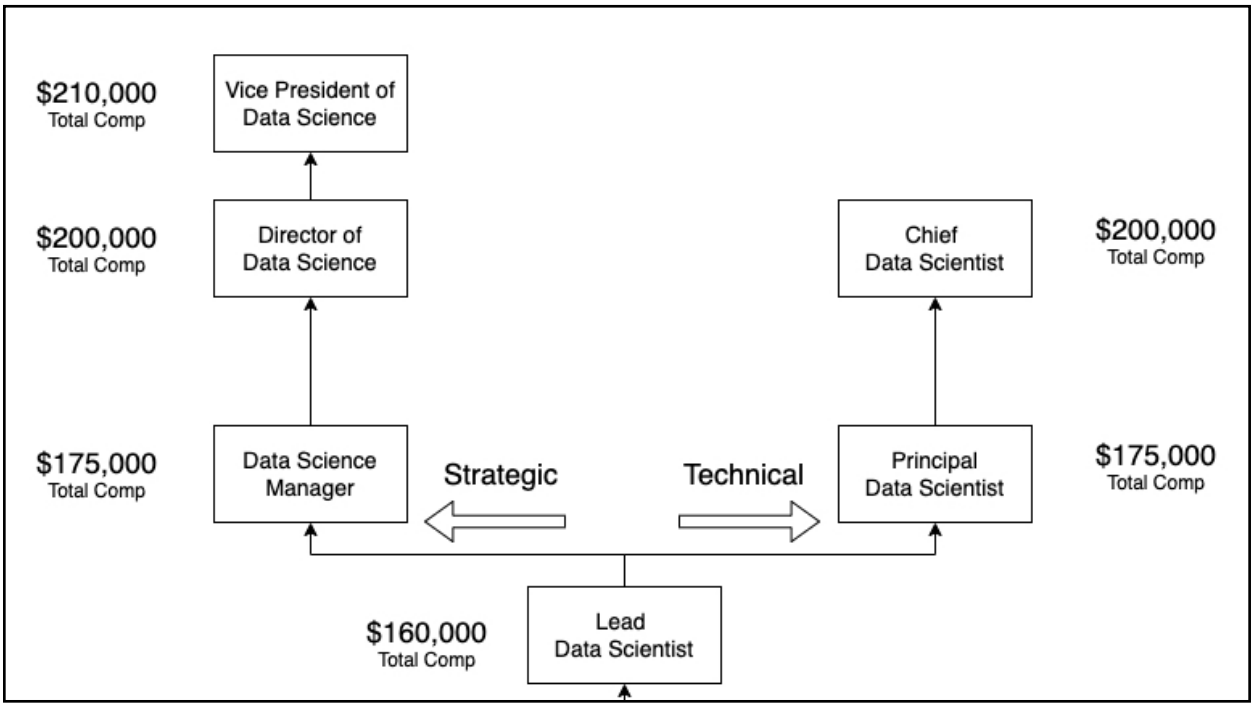
Reticulate: R's Python Connector

This is the most mind-blowing thing about R. And, it's a super-power that will:

1. **Empower you** to work collaboratively with Python teams (even though your an R user)
2. **Give you** the key ingredient to make R packages that connect to python package. Here's an R+Python Package that I created called `modeltime.gluonts` that connects to the GluonTS Python package for forecasting. Pretty sweet!!

Ok, now that you have the skills to become a Senior / Lead Data Scientist, we need to consider where you go *after* you become a Lead Data Scientist…

The Technical Path vs Strategic Path



Technical vs Strategic Career Path

You see, there are two paths… so choose wisely.

Don't worry, I'll help make this decision crystal clear.

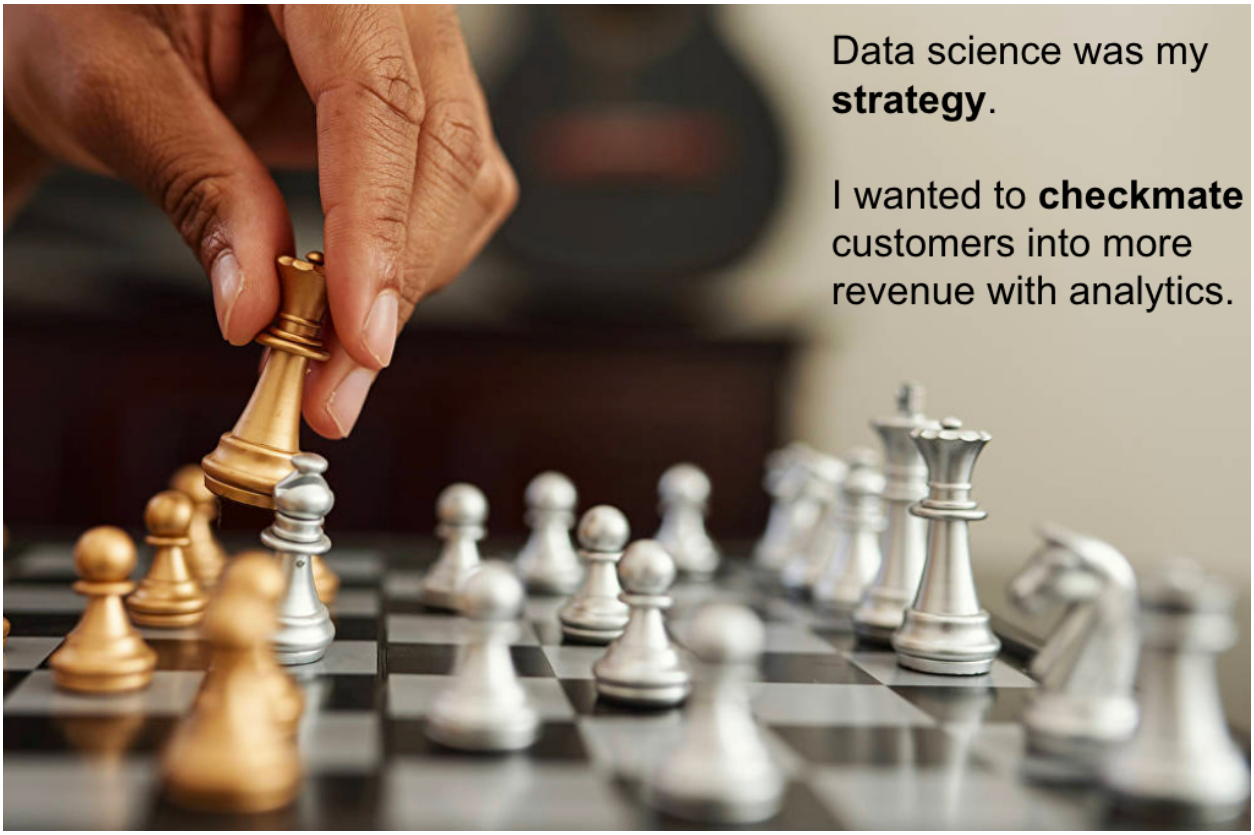
I'll share my perspective and how I chose when it was my time.

You see back in the day, before I was this amazing data science educator, I was a data scientist without a title (it was before "data scientist" existed in my previous employer).

I worked at a small company called Bonney Forge.

And, more than anything I loved the idea of influencing the direction of the company.

I was entrepreneurial, and enjoyed working with people.



Business was 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 anyone guess the path I chose?

If you guessed “STRATEGIC” you are 100% correct!

What about technical?

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.

I actually didn’t like this aspect one bit, but learned to be good with it, 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 stand personnel issues for a year or two then don't go into the strategic path.

Directors and chiefs are great, but I'm no where near that level

Listen, I get it.

But 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… capisce?!

Three ways to getting promotions (FAST)

The 3-ways to getting promotions fast are:

1. Be more productive than everyone else around you
2. Do something big!! (and repeat)
3. Job hopping

I'm a big fan of case-studies (it's what we do in MBA school), and they work. So let's cover some case studies of how to get promoted.

Note, I'm not going to discuss job-hopping. I'll have a different article soon on **how to get a job in data science** (with interview hacks and back-office secrets guaranteed to land you a job). Stay tuned.

Onto our first case-study.

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

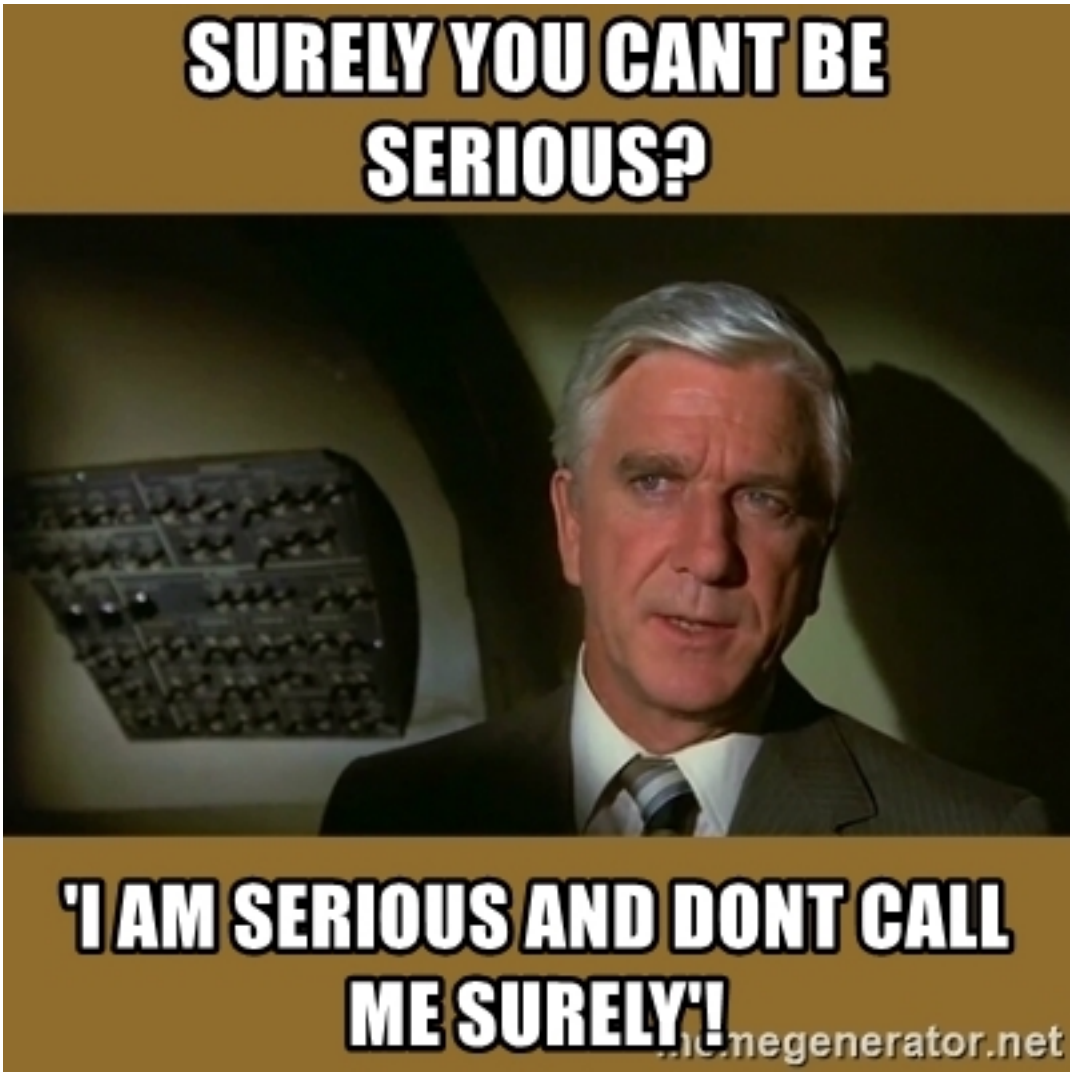
People are lazy. (I'm just going to say it.)

The simple fact is that people get comfortable.

But you don't have to. In fact, the comfort of others CAN be something you can exploit.

An edge (if you're smart).

Surely, you can't be serious?



It am serious!

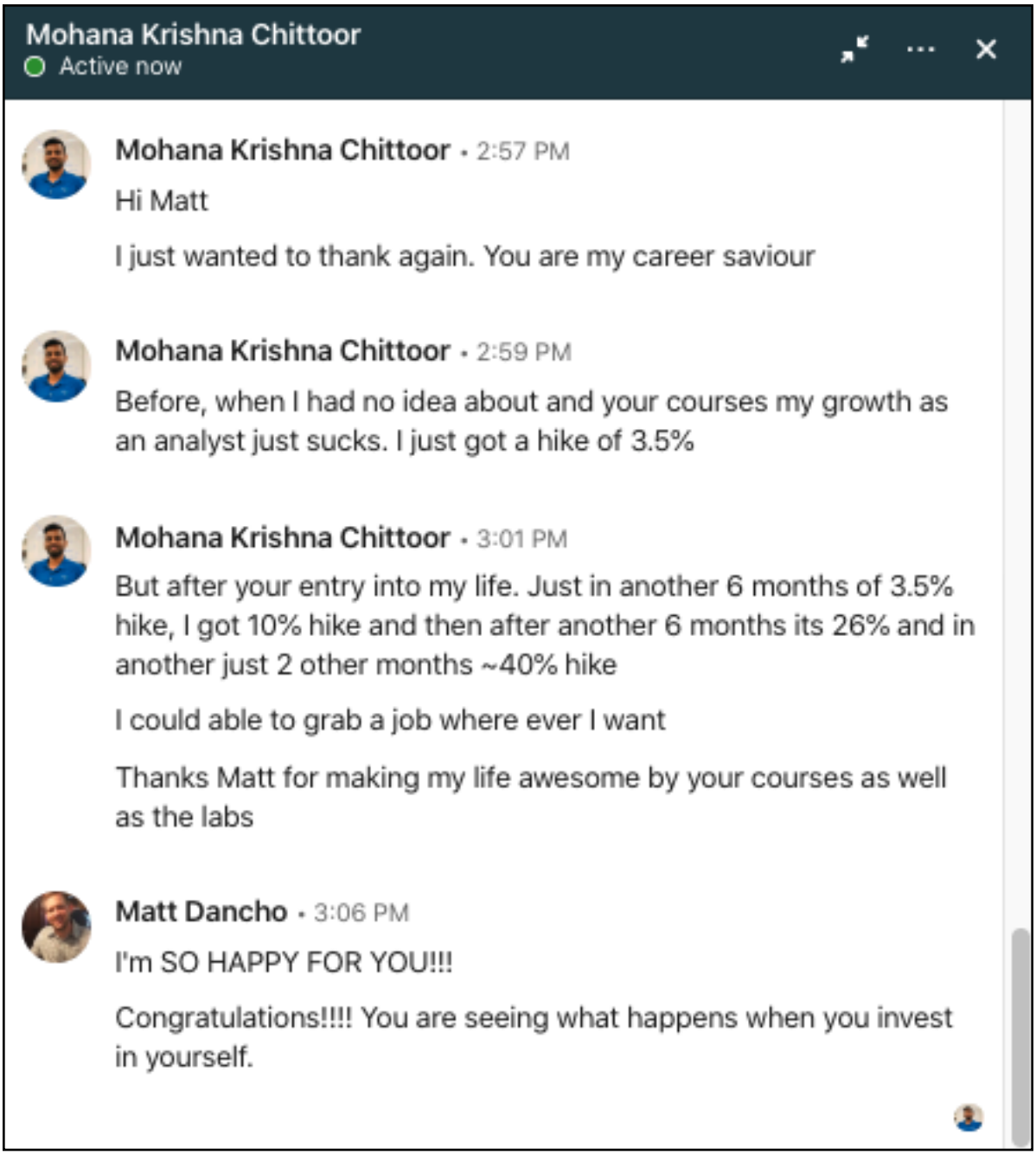
In fact, here's the story of how Mohana did it (remember Mohana from the beginning of this chapter?).

Mohana was 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 says, *"I just wanted 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]."*

Mohana exclaims, ***"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 started working with me.

That's when the flood of raises started.

Let's dive into how Mohana trippled (yes 3x-ed) his productivity.

3X-ing his productivity with my R-courses

Here's the scoop. Mohana was working with a bunch of Python coders.

These guys are slow and comfortable.

But Mohana isn't like them. He's motivated.

Mohana just needs a little edge.

And, Mohana got that when he met Matt Dancho (me). :)

You see I gave him the edge he needed to triple (yes, 3X!) his productivity versus his peers.

How did I *3X* Mohana's productivity?

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 consulting 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. You see while his python counterparts were trying to get their first app launched, Mohana already had three done.

And, 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… I mean no one had in his company.

Then, Mohana simply applied what I taught him to his business. And…

Now he's a Lead Data Scientist

Mohana kept repeating. He kept growing.

Today he's 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.

And, this is what I live for. Seeing my students succeed like this.

But that's just one case. I couldn't possibly duplicate it could I?

Let's see…

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

What if you could save your company $5,000,000 every year in perpetuity?

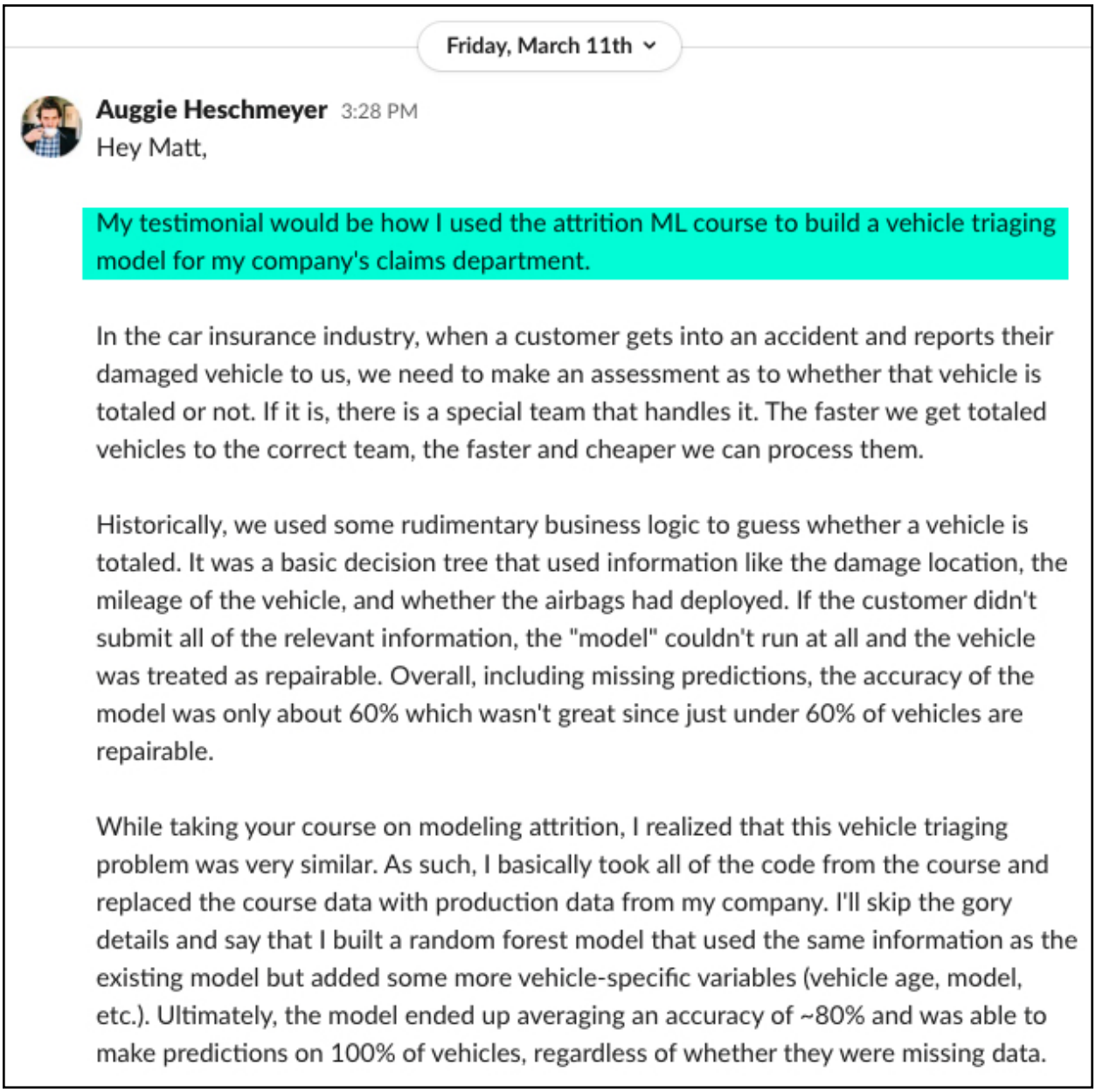
Would your company value you?

Would you be promoted?

Well, this is exactly what happened to another one of my students.

Auggie learned how to make attrition models

Here's what Auggie did…



Through my R-Track Program, Auggie learned the necessary skills to build complex attrition models.

Auggie was then able to apply the course framework to his business problem.

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.

Using my coursework, Auggie made a better model. In fact so much better that…

Auggie's model saved the organization $400,000 every month!

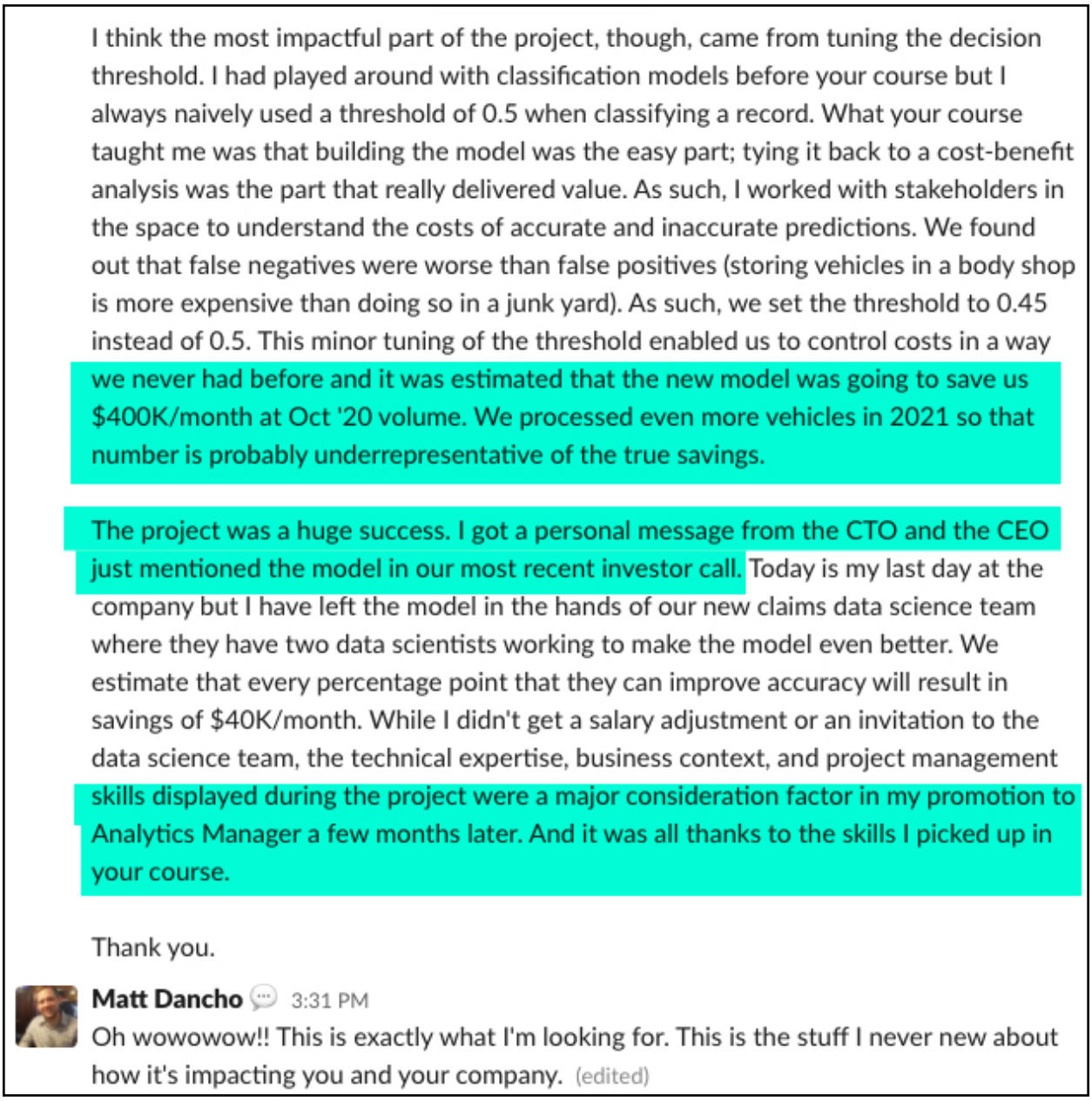
A quick math check means that Auggie saved his organization $4,800,000 per year. And these estimates may actually be low (meaning the model is likely saving more).

Auggie was recognized.

Auggie says, *"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.

He exclaims, *"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 courses."*



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)

This is why organizations everywhere will value you if you learn data science.

And, I can help.

How to go from a $75,000 to a $150,000 salary

If you've read this chapter, you now have all of the information that is needed to take you from a $75,000 salary to a $150,000 salary.

But, you still **don't** have a plan to do it fast.

It will take 2-years (or longer) on your own.

In fact, it actually took me 5-years of struggle to learn data science on my own. I took bootcamps, read books, research paper after paper, and nothing worked.

But that's why I created my R-Track Program. To help people like me, struggling to get the 6-figure career they deserve.

Imagi
ne what earning $125,000+ in 6-months from now could do for you
How amazing would it be to know you have the financial freedom to do anything you want.

You can take a vacation.

Spend more time with family.

Have financial stability and less stress.

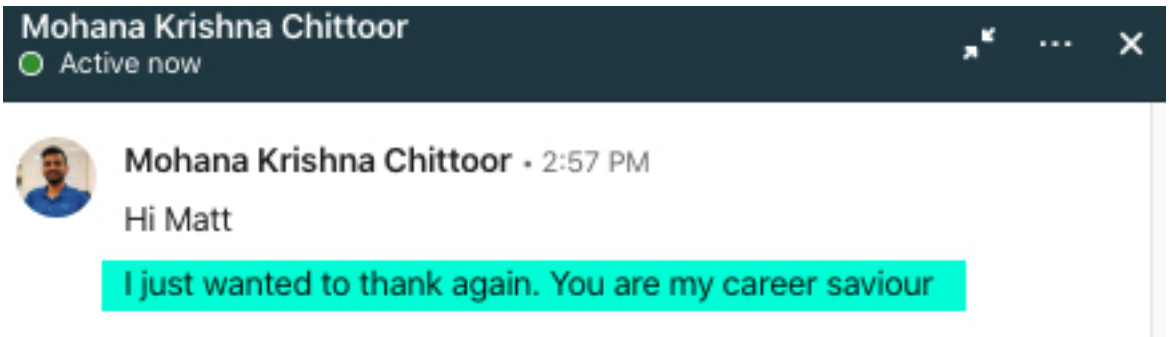
And this is why an investment in yourself will unlock those dreams.

Remember Mohana? (3.5% raise to 94% raise in under 1-year)

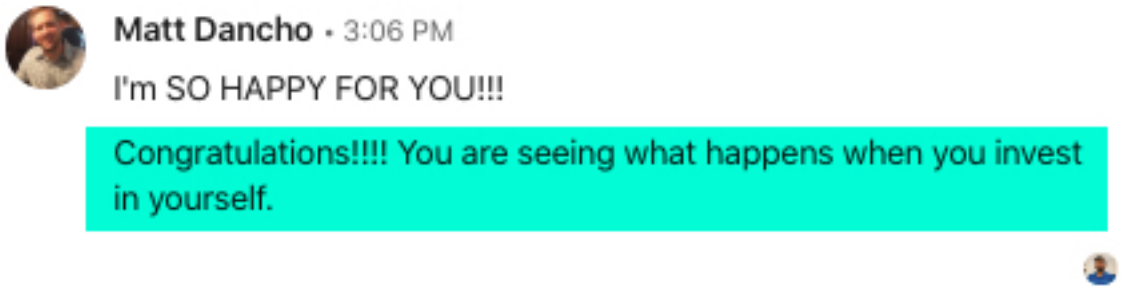
Mohana was getting 3.5% raises.

He's now a Lead Data Scientist at Money View, one of India's fastest growing start-ups.

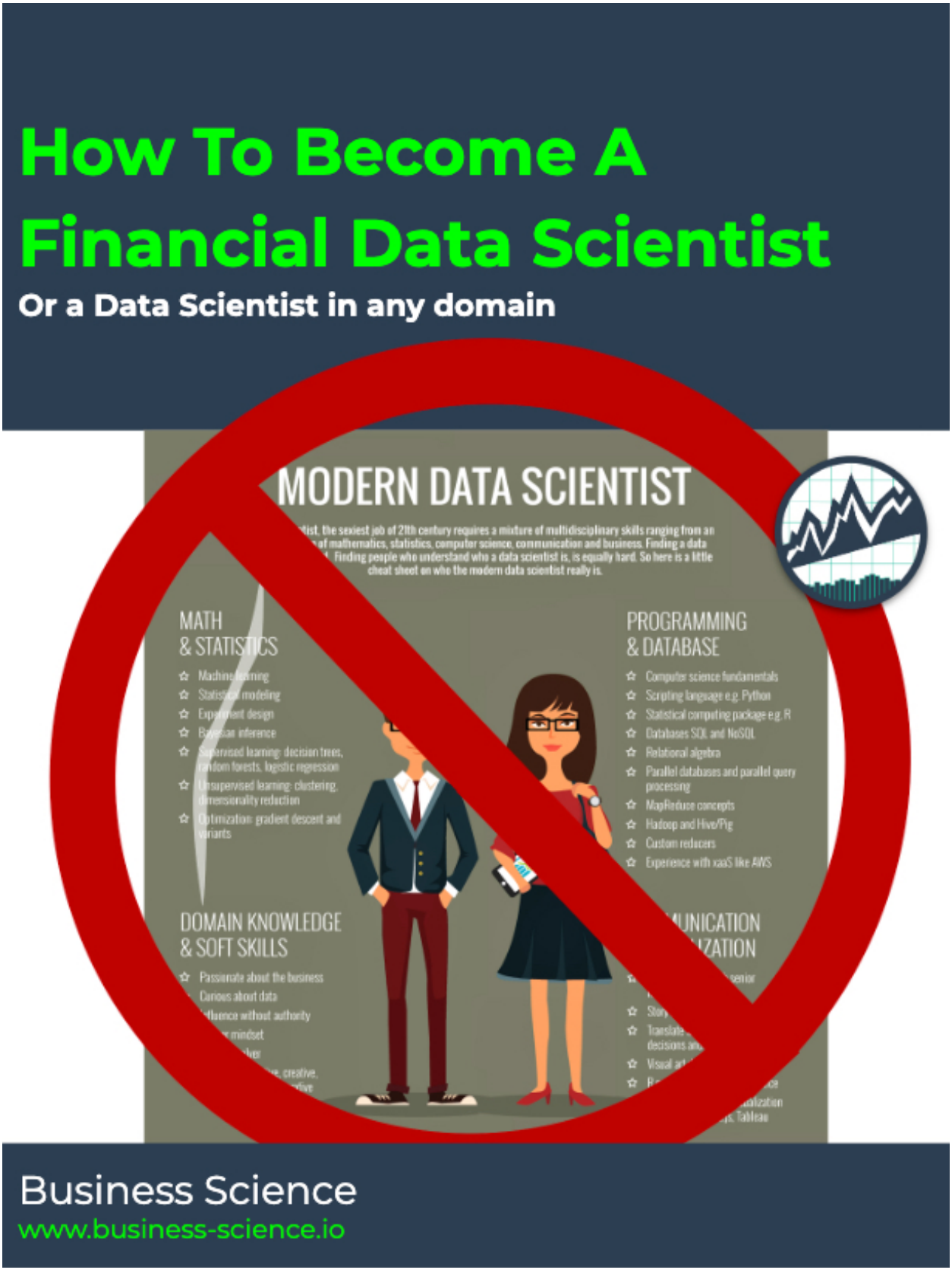
He says, "I just want to thank you again. You are my career savior."



I replied, “Congratulations. You are seeing what happens when you invest in yourself.”

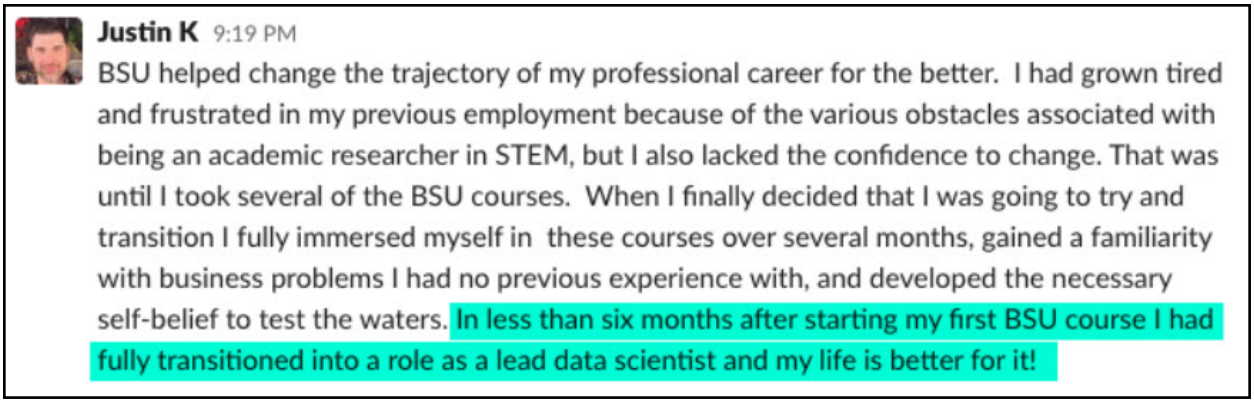


Chapter 3: How To Become A Financial Data Scientist (Or A Data Scientist In Any Domain)


How To Become A Financial Data Scientist
Or a Data Scientist in any domain
MODERN DATA SCIENTIST
MATH & STATISTICS
PROGRAMMING & DATABASE
DOMAIN KNOWLEDGE & SOFT SKILLS
Business Science
www.business-science.io

It was December of 2020. Justin was a working in academia at the University of Southern Mississippi doing sports analysis. But he was longing for more.

By June of 2021 (exactly 6 months later), Justin told me that he had just got his dream job - Lead Data Scientist at Northwestern Mutual (one of the biggest insurance firms in the US).



I was ecstatic for him. I knew he was destined for great things. But I also wanted to know more.

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:

- The 80/20 rule (how it helps you provide business value)
- Why becoming a unicorn is slowing you down
- The big mistake you're making (I made this too)
- The 3 things organizations value (and how to deliver each with data science)
- Case Study: A real world example of how to provide value to a business
- What skills you need to learn (to create value)
- How to earn a $125,000 salary (in under 6-months)

First, let's talk about the 80/20 rule.

1. The 80/20 rule for business value

As I began researching what separated the those that were becoming data scientists from those that weren't, I quickly found out a big difference.

The ones that were NOT making it were trying to learn everything. And trying to become masters.

But the ones that WERE making it were doing something different. Something unique.

They were learning by applying the **80/20 Rule**.

More specifically they were doing a 2-step process that delivered unheard of results:

1. They figured out what creates value for the business.
2. Then they applied the **80/20 Rule** to create value as fast as possible.

Do you see the difference?

The ones that WERE becoming data scientists understood that speed is critical.

Why is speed critical? (The plateau effect)

I dove a little deeper to understand why speed was critical. Here's what I discovered.

The ones that were NOT making it were never actually applying anything.

Progress was slow because they had no focus. And they plateaued.

The **plateau-effect** was costing these unsuspecting data scientists their dreams.

Even worse, their dreams were dying a **slow, agonizing death** until they simply QUIT.

And these slow learners don't realize the HUGE financial cost.

The huge financial cost

I wrote about this financial cost in the 3 paths to learning data science here. Just to recap, there are 3 types of data science learning paths:

1. **Those that have 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-percent annual raise.
2. **Those that have a crappy plan.** They will take 5-years. But will eventually learn data science. They will also lose out financially because it took them sooo long to learn data science. 5-years at $125,000 per year when factoring in a low 3-percent raise = **loss of $664,000**. Ouch!
3. **Those that have an exceptional plan.** They are likely to be successful and can complete the transition in under 6-months.

You see there's a natural phenomenon that happens when learning ANY new or complex task like learning to code, learning guitar, speaking a foreign language, etc.

We learn by first mastering a small subset of the most frequently used tools.

We then build on those over time.

Learning data science was simply no different.

So let's dive into how we can become a data scientist with the 80/20 Rule. And focus on generating business value at the same time.

How to become a data scientist

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 don't worry - if you are not a financial person, the process for creating business value is the same for other domains.

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.**

The opportunity to understand what organizations in Finance value from the data science practice.

If you understand what they 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 (or any organization).

For non-financial people

What I'm covering here is focused on finance (because I needed a specific example to drive this point home).

BUT, the same strategies I cover can be applied broadly to **ANY domain**, as wide-ranging as Marketing, Research & Development, Medicine, and more.

If you are reading this, and don't have knowledge on *Portfolio Theory* and *Risk Management*, then just replace those terms with **"Bubble Gum"** (or any silly object) and keep moving on.

The strategies for learning what organizations value are most important. Not the BUZZ WORDS or domain jargon.

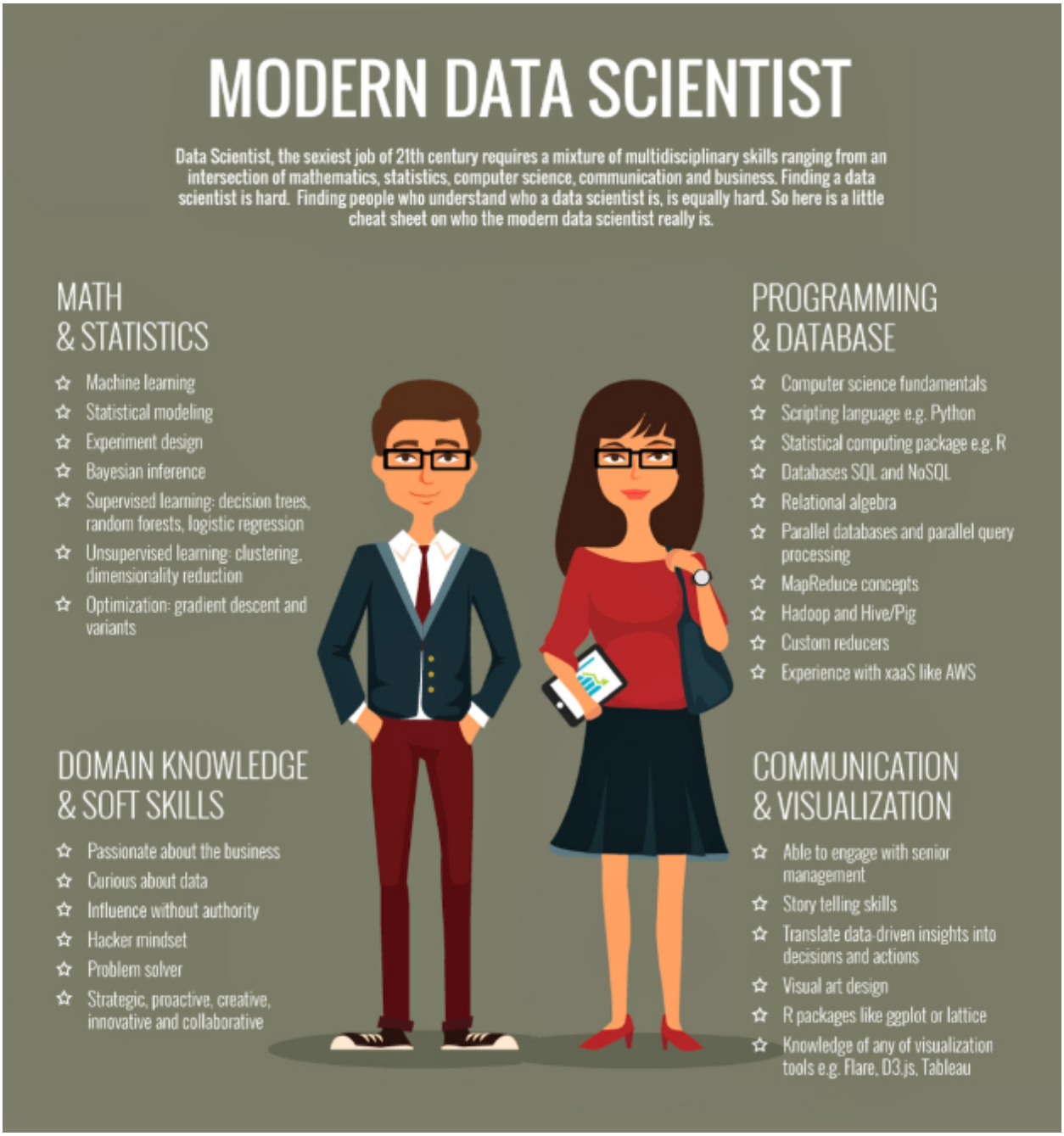
What I will show you is how you can take domain knowledge and extend it to value-creating activities for the business using data science tools.

But, first, the cold hard reality…

2. Why becoming a unicorn is slowing you down.

Most financial (or non-financial) people that want to break into Data Science make a **major a mistake** that costs them a career in data science. It starts with their first step.

They see the *"Modern Data Scientist"* infographic, and immediately feel overwhelmed. Worse, they begin down the path of learning everything. ***Learning everything is unproductive.***



"*Modern Data Scientist Infographic*" -

Everything That Is Wrong With A Learning Strategy

There is no strategy to this graphic. No foundation, no purpose, no intent. Just a smattering of skills that supposedly create a data scientist.

Worse, students starting out believe that this is the ultimate goal - ***A Unicorn Data Scientist.*** (What's that? A unicorn is mythical creature that

doesn't actually exist, but the idea spreads and people believe that they can find it, or worse… become it.)

I'm here to tell you that (fortunately) this myth is not a reality.

Here's how learning data science really happens.

We All Start At The Same Spot - Zero

When we start out learning data science, we are the most vulnerable to making missteps.

I personally remember feeling overwhelmed and directionless. It's at this moment that we are easily influenced to take the path of learning everything (and anything).

Learning *everything* is a costly strategy issue, but an easy one to make. With so many people saying different things, yet **none of the "experts" are stepping up** to give you mentorship.

It's scary - being alone on this journey.

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).

Time is our enemy.

The longer we take on this journey, the more competitive it gets and the more likely we are to fail.

Use time wisely. Be efficient. Be effective.

We Grow By Building Skills That Add Value

One thing I learned along my own journey was how to sell my value.

It was an incredibly important lesson that I learned through my experience consulting. Every initial client engagement was a sales pitch. I had to sell myself.

How did I do it?

When I'd begin any consulting engagement, I'd never go in saying I know something. Rather, I'd ask how I can help and **listen** for opportunity.

As soon as the client began talking, **I'd uncover their problems.** The more they talked, the more that problems stacked up. And, sure enough I could solve a lot of them.

This showed them value. I exposed problems they didn't even know they had, and I offered the solution. ME!

I was their bridge to value: Solving the problems they now realized they had. None of this was based on my skills. Sure, skills were needed.

BUT, skills alone don't sell. It's solutions, results, and value that sells.

3. The big mistake you're making

Quite simply skills don't sell. So why are you marketing them?

We need to change our beliefs.

Here's your mistake.

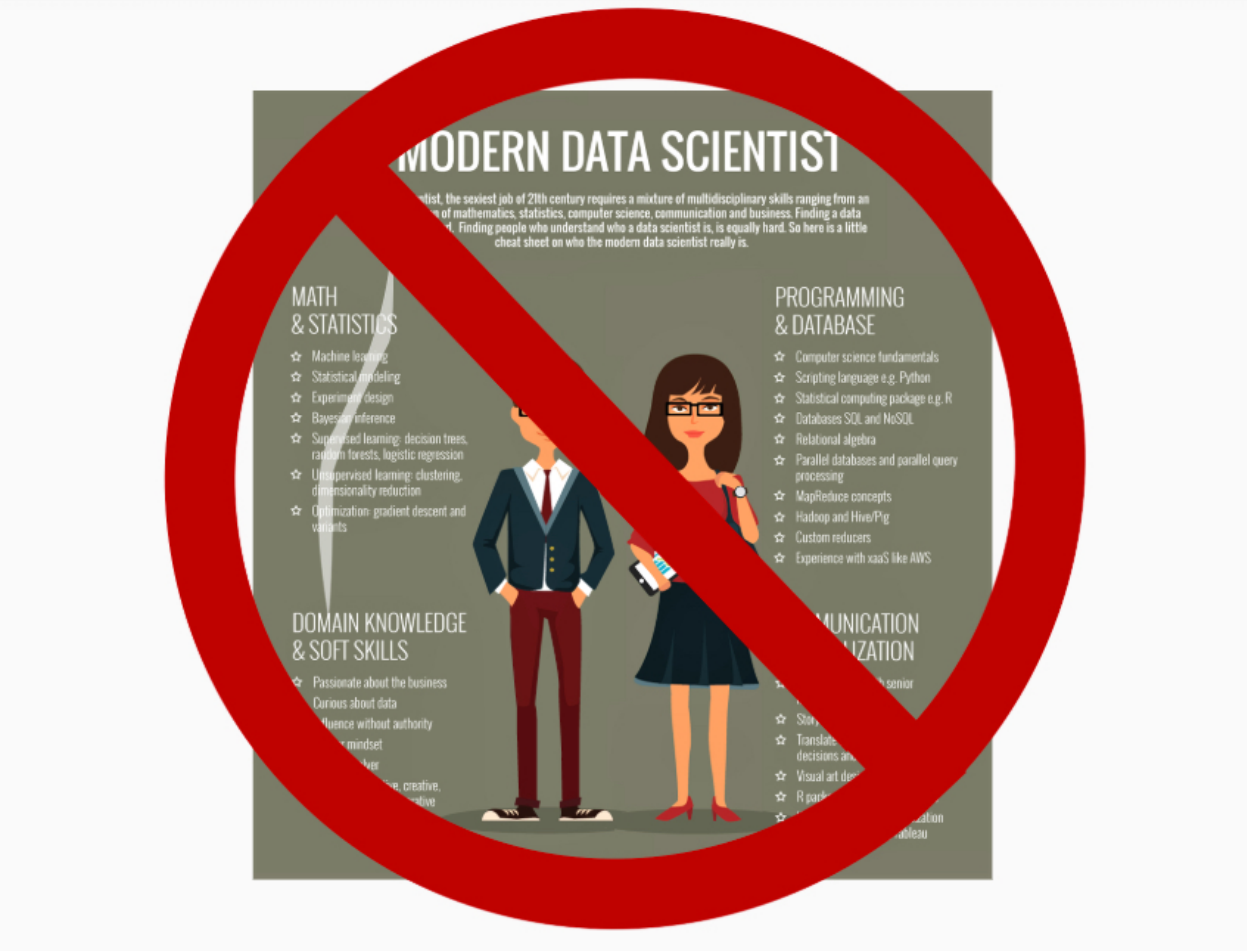
Most of us feel that to get a job in data science, you need to learn data science inside and out. Machine Learning, Deep Learning, Neural Networks, Graph Theory - the list goes on.

And then you market this smattering of skills.

Don't market skills. Market results.

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.

Remember the *"Modern Data Scientist"* infographic? Don't learn all of these skills.



Bad Strategy: Learn Skills That Every Data Scientist "Should" Have

The importance of creating value

Rather than learning everything. A better strategy is learning how to create value by incrementally adding skills to your toolkit. Focus on addressing what financial organizations want. Then learn tools that will deliver it.

Good Strategy: Learn How To Create Value

The next logical questions is…

What do organizations value?

4. The 3 things organizations value.

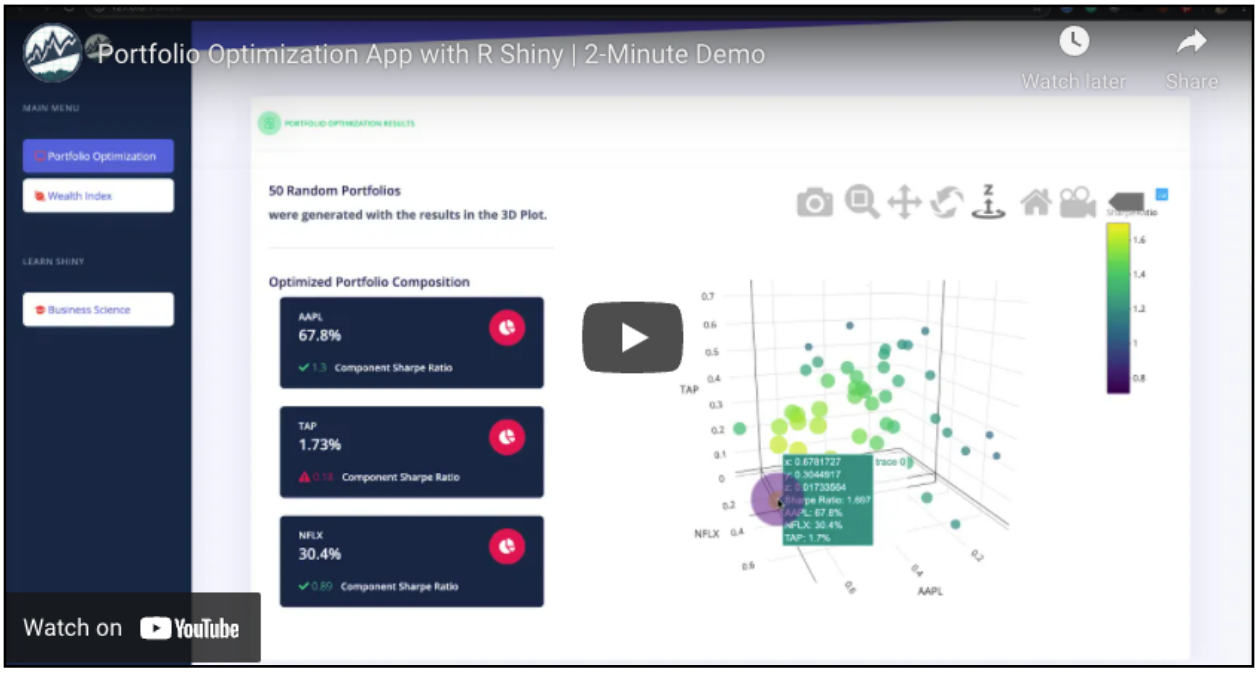
To be effective in a Financial Organization (or any company), you need to generate value for the business. You do this by:

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

Solving problems that address **KPI's (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) generate value?

It's Simple. **By taking applications into production**.



What is an application?

Every day we make decisions based on intuition. Decisions in the absence of data are WRONG more often than not.

When we use data to improve decision-making, value is generated for the organization by reducing costs, increasing revenue, and/or maximizing profit.

The application is the THING that non-data scientists (normal people) 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.

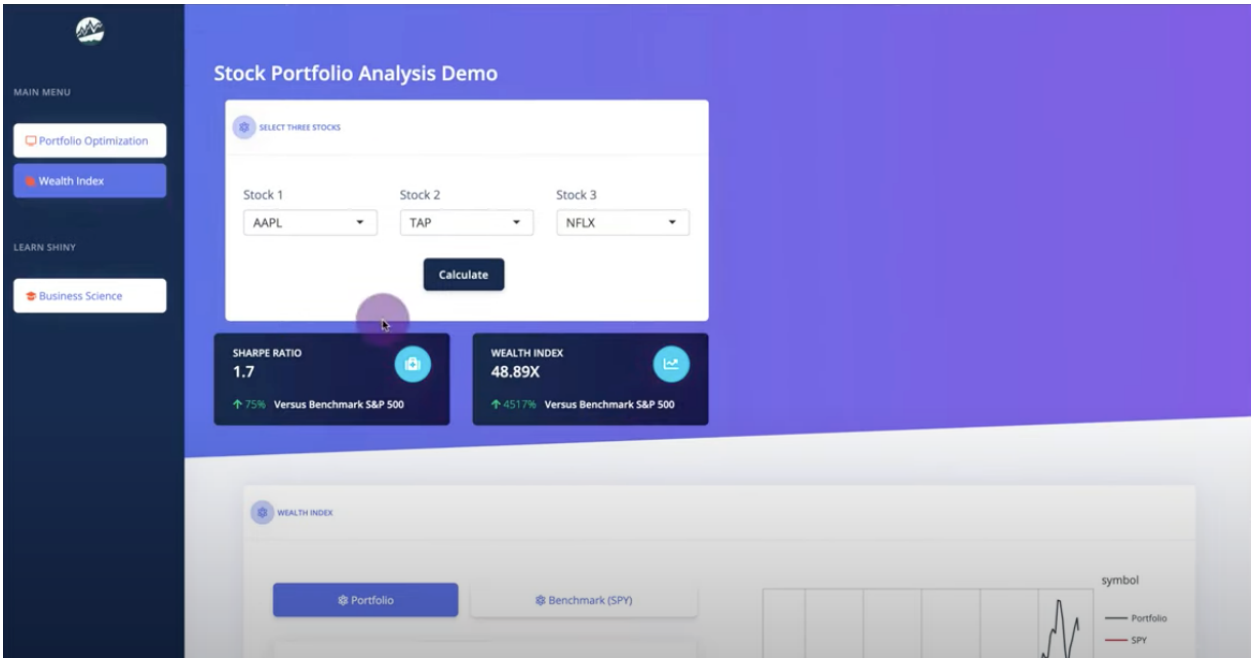
Production generates MASSIVE VALUE. In fact, applications that embed data science can save organizations $15,000,000 per year or more!

Don't believe me? Here's an expert application that I built that can easily result in *multi-million-dollar-per-year savings*.

5. Case Study (Finance): Assisting an asset manager's tactical investment allocation

The only way to make a difference is by understanding the people you seek to help.

Let's walk through a short example of this.



This is the Stock Portfolio Optimization Application (DEMO HERE) that I demonstrated at the *R/Finance 2019 Conference*.

Problem Statement

Asset Managers select stocks based on their knowledge of the company, market, and intuition of what the future holds. However, allocating an

investment among the basket is ***a time-consuming problem that is costly if the Asset Manager over-weights a risky stock***. A bad bet can result in lost Clients, costing the 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.

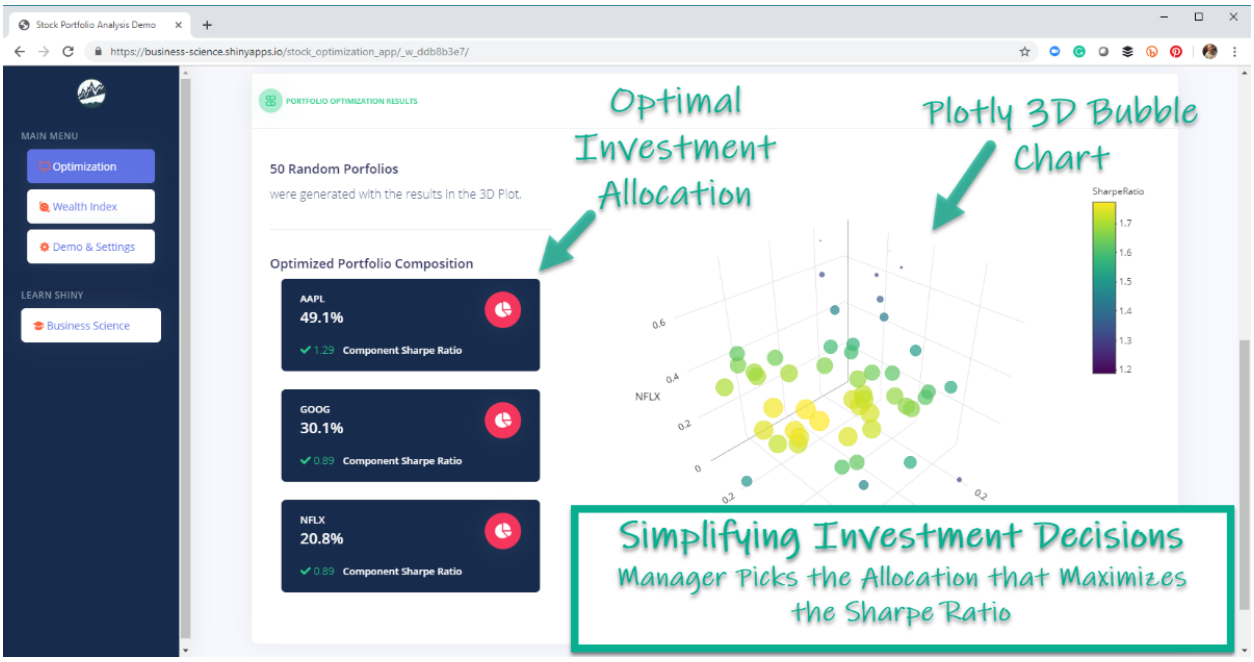
The Application

This application allows the Asset Manager to focus on his or her job of picking stocks, while the investment allocation decision becomes automated using modern portfolio theory.

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

The web application available for demo here and described in the YouTube video generates value.

The application helps an Asset Manager ***make better investment decisions that will consistently improve financial performance and thus retain clients.***



Tactical Asset Allocation - Portfolio Weights Optimized Using Sharpe Ratio - App Improves Decisions

6. What skills you need to learn to do this.

The road to go from where you are now to a data scientist in a Financial Organization can be accomplished in weeks, not years. But, you need to have a plan to ***strategically learn the right skills***.

This where I can help. I've been there. I'm willing to step up. I'm willing to guide. But, make no mistake, ***it will take serious commitment on your part.***

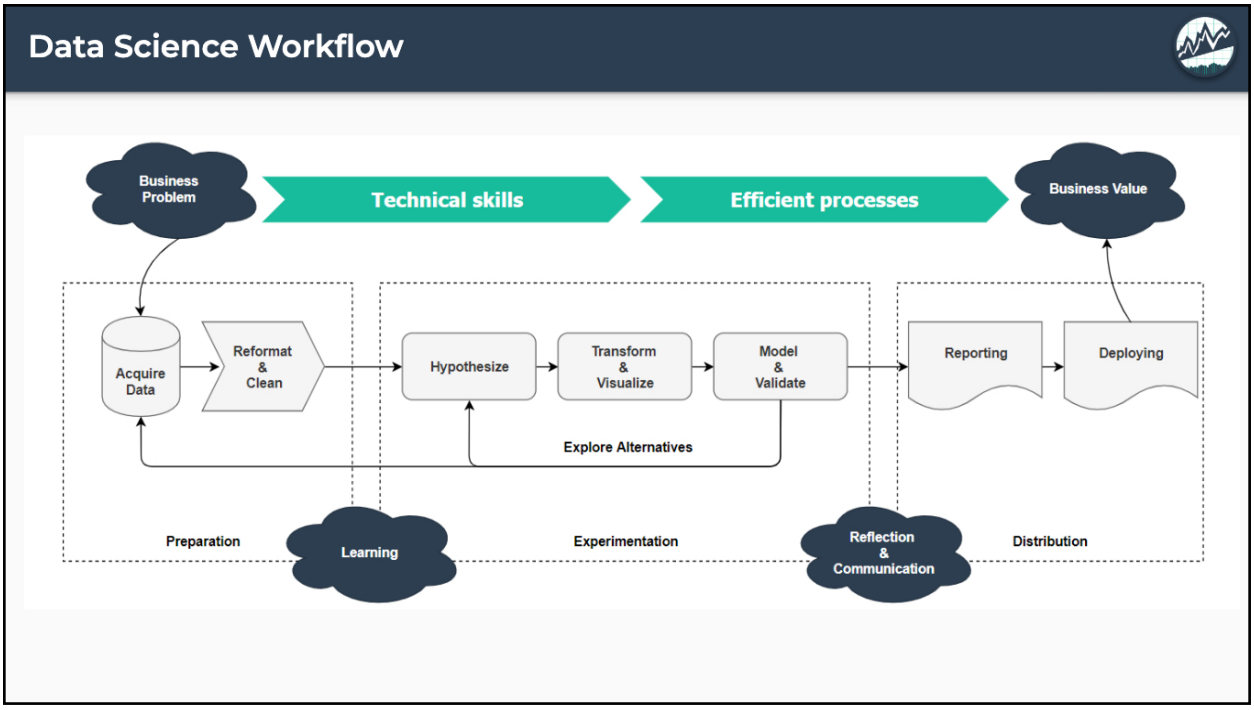
Here's what I recommend that you learn and why. It's called the *Data Science Workflow*.

The Data Science Workflow

Building applications like the Stock Portfolio Optimization App is what we call Production.

This is the end stage of your efforts. But what you don't see is the hard work that you (the data scientist) put in beforehand.

That hard work is actually a process called the **Data Science Workflow**, and it looks something like this.



The Data Science Workflow

The *"Data Science Workflow"* is the series of tasks required to go from business problem to business value. It's a time consuming process that requires:

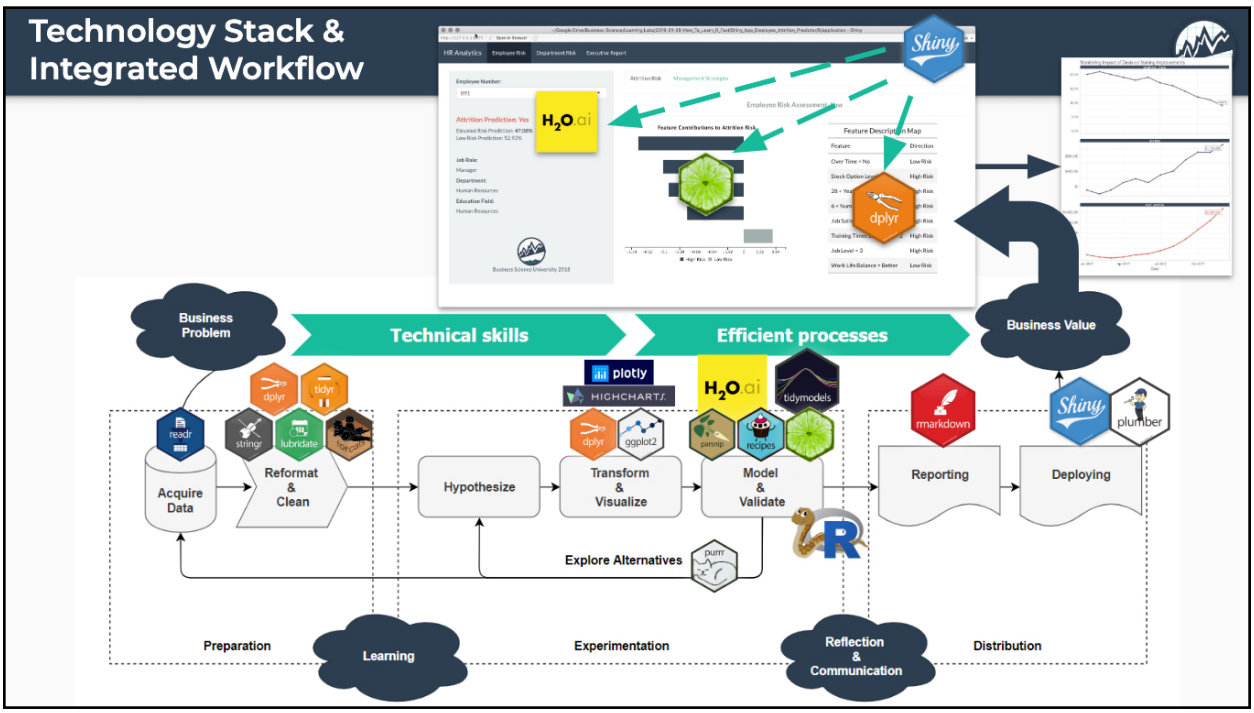
- **Business Problem Understanding**: Working with process stakeholders (e.g. Asset Managers) to understand their unique business challenges
- **Communication of Business Value**: ROI analysis of any solutions, communication with executive leadership to convey the value

In the middle is a complex series of actions that involve ***Data Science Tools*** (everything involved in going from machine learning to reporting and deploying applications).

Data Science Tools Exposed

Here is the same graphic with a set of tools that can be used as part of the *"Data Science Workflow"*. The tools integrate throughout the problem

solving and solution building process. This is how we add value to the organization.



The Data Science Workflow - With Tools Exposed

Value comes from tool integration

Value comes from using **a specific set of tools** that **incrementally add value** along the *"Data Science Workflow"*.

This allows us 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)**.

Let's break this down.

A specific set of tools…

Focusing on this specific set of tools cuts the time to learn data science dramatically. This is the ***80/20 Rule in Full Effect!***

…that incrementally add value…

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.***

…that end with a web application that delivers massive business value…

The Application is the Value-Generator. Without it, the data science team adds little value to the organization.

…and we can measure this with Return On Investment (ROI).

Any business improvement should be tracked, reported on, and measured for return on investment. Changes in KPI's converted to financial value.

7. How to earn a $125,000 salary in 6-months

With everything we've covered in this post, you now know what businesses value and what you need to deliver (an application) so they get massive value.

But you have no plan that incorporates the 80/20 philosophy.

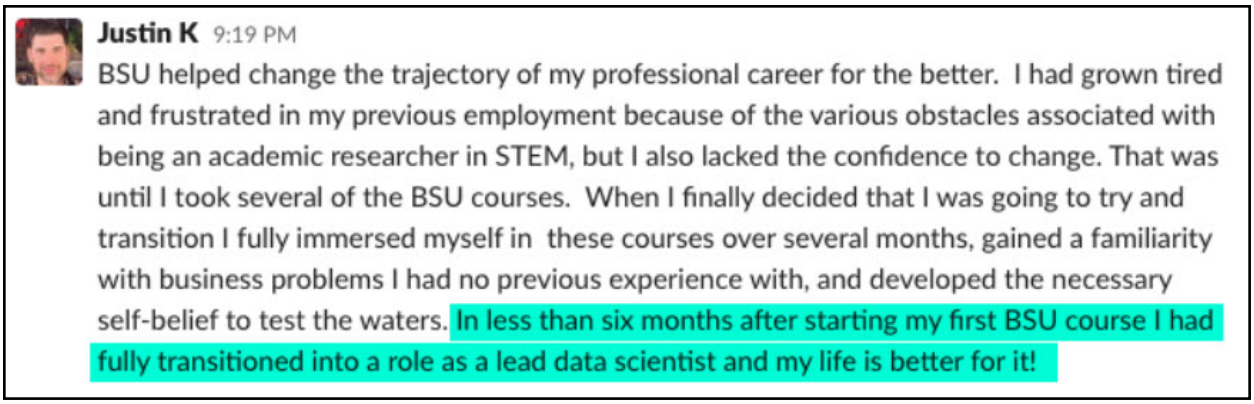
In fact it will still take you a minimum of 5 years to learn. (I know because this is how long it took me. Ugh.)

But what if you could do it 6-months?

How amazing would it be to have a 6-figure career that you love?

And in the process earn $125,000 per year or more until you retire and have the financial freedom to do the things you enjoy.

Do you remember Justin?

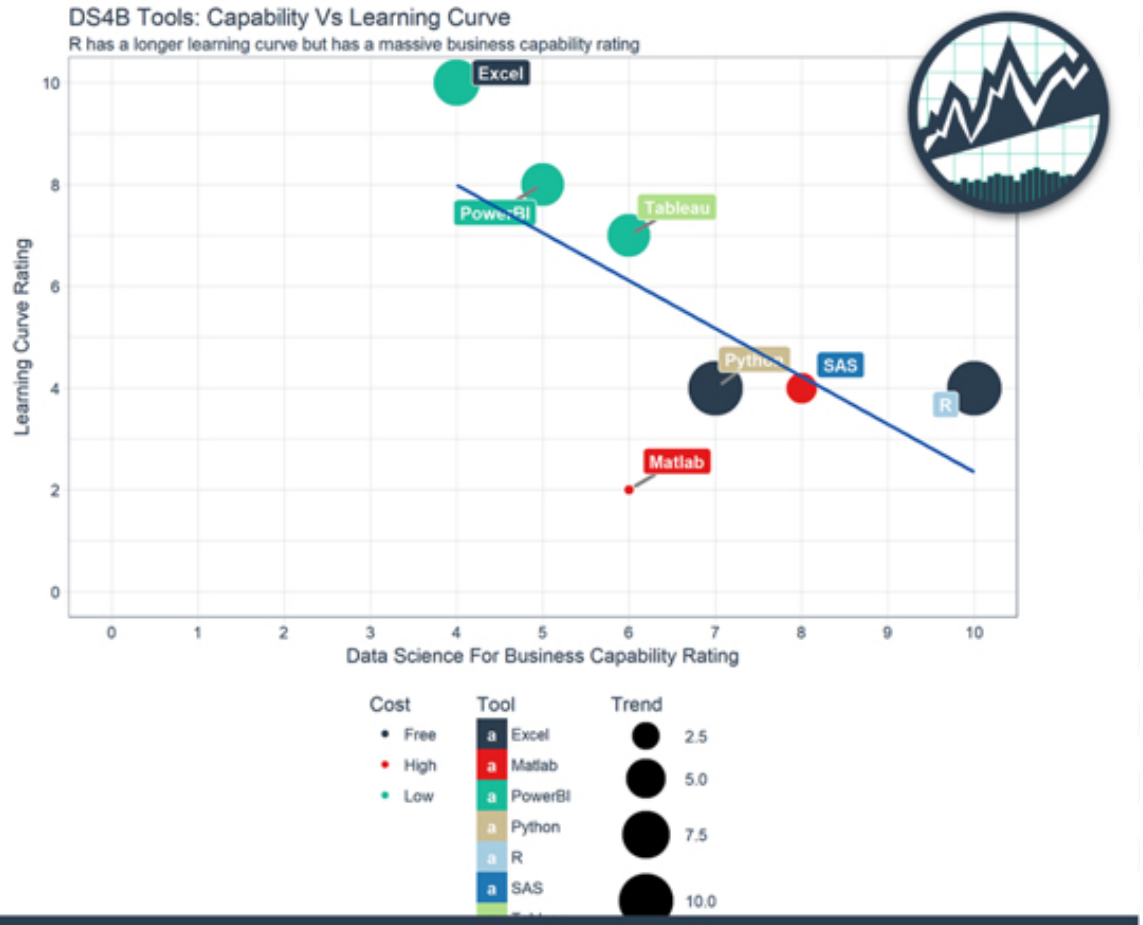


Justin was student that in less than 6-months transitioned from academia to the Lead Data Scientist of Northwestern Mutual (Top 10 Insurance Firm).

Chapter 4: 6 Reasons To Learn R For Business

6 Reasons to Learn R for Business

Why R Might Be the Right Choice for You



Business Science

www.business-science.io

Data science for business (DS4B) is the future of business analytics, yet **it is really difficult to figure out where to start**.

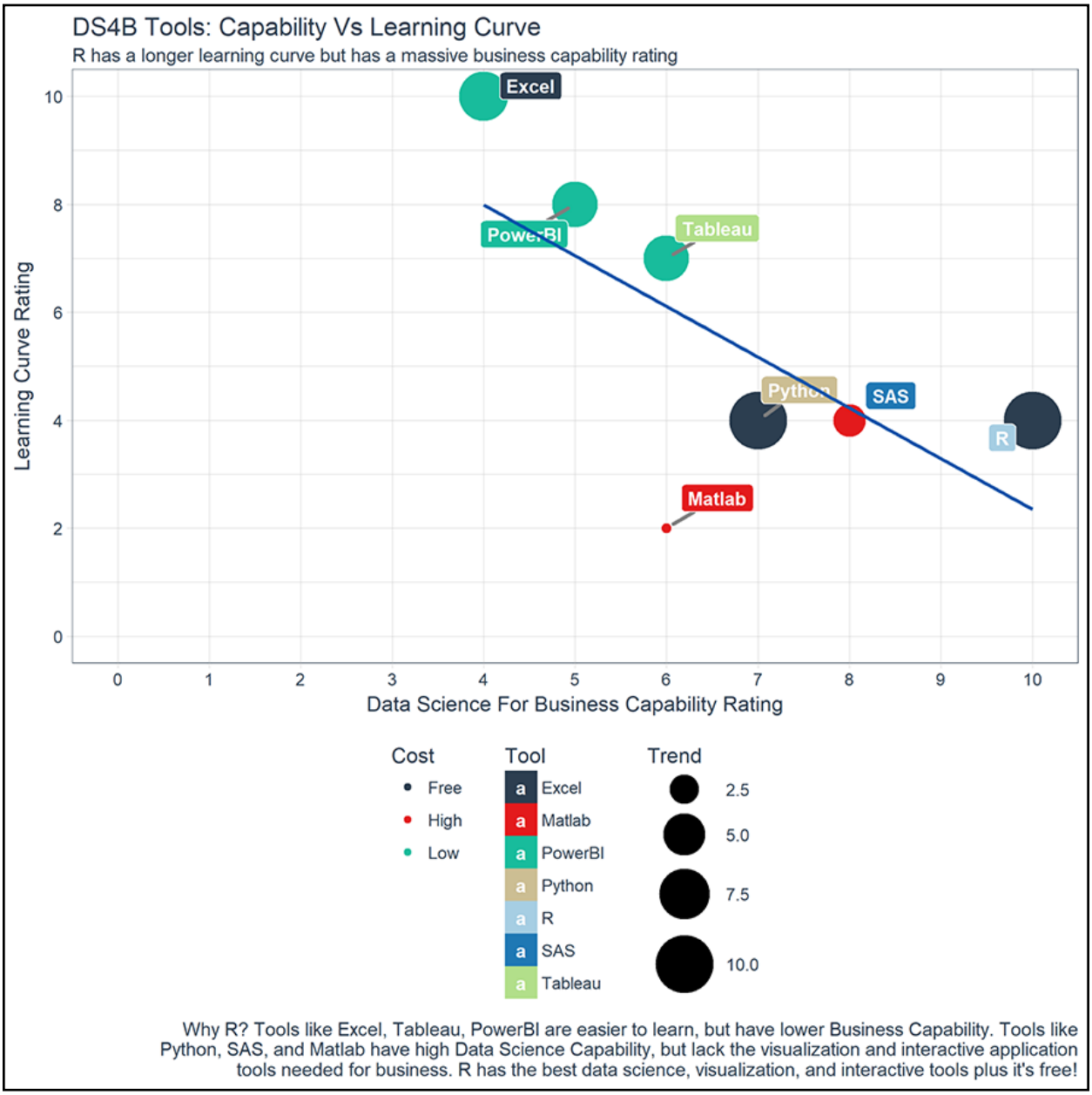
The last thing you want to do is waste time with the wrong tool.

Making effective use of your time involves two pieces: (1) selecting the right tool for the job, and (2) efficiently learning how to use the tool to return business value.

This chapter focuses on the first part, **explaining why R is the right choice in six points**.

If you'd like to tackle learning R efficiently, we have another chapter that covers the 80/20 Rule for Learning R.

Reason 1: R Has The Best Overall Qualities For Business



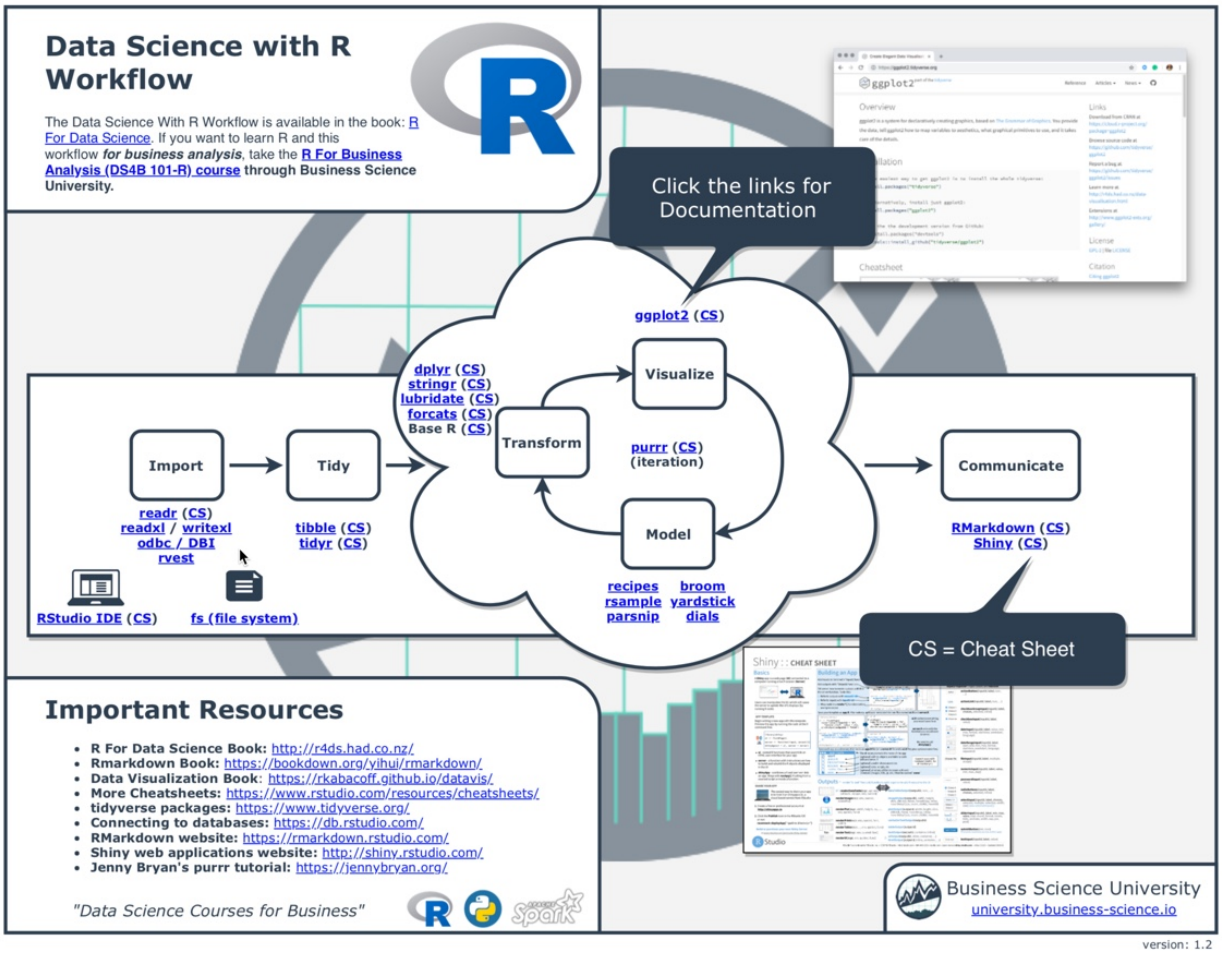
There are a number of tools available for business analysis/intelligence (with DS4B being a subset of this area). Each tool has its pros and cons, many of which are important in the business context. We can use these attributes to compare how each tool stacks up against the others! We did a qualitative assessment using several 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 we saw was particularly interesting. A trendline developed exposing a tradeoff between learning curve and DS4B 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. **Our opinion is 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 only downside is the learning curve. The Cheat Sheet below showcases the powerful libraries that are at your fingertips - Download our Ultimate R Cheat Sheet to see what libraries are available to solve specific needs.



The Ultimate 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 really have two options: **R or Python**. When starting out, you should pick one. It's a mistake to try to learn both at the same time. Your choice comes down to what's right for you. 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**. Don't understand what we mean? Let's break it down.

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 when things are not going well and to make well-informed decisions that improve the business.

Now that we recognize what's important, let's learn about the two major players in data science.

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 its communication infrastructure**. R 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?** If yes, learn Python.

- **Are you an analytics professional or mechanical/industrial/ chemical engineer looking to get into data science?** If yes, learn R.

Think about what you are trying to do:

- **Are you trying to build a self-driving car?** If yes, learn Python.

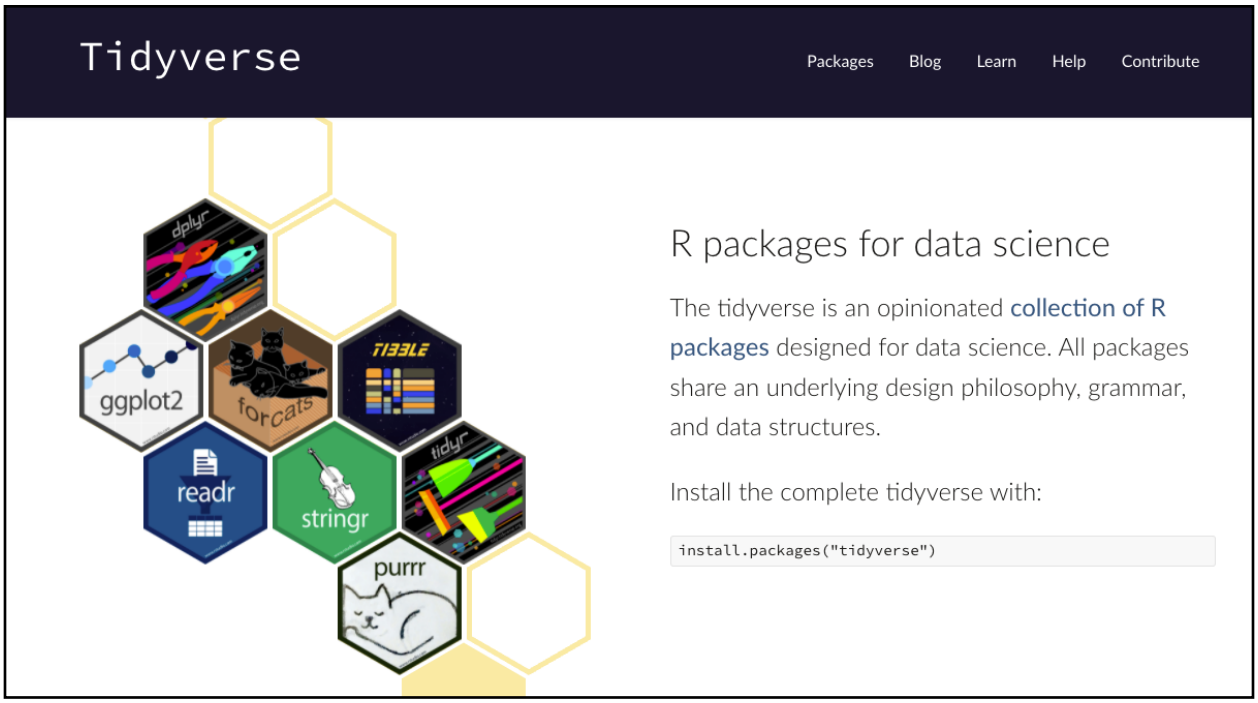
- **Are you trying to communicate business analytics throughout your organization?** If yes, learn R.

Finding it Difficult to Connect Data Science to Business?

Reason 3: Learning R Is Easy With The Tidyverse

Learning R used to be a major challenge. Base R was a complex and inconsistent programming language. Structure and formality was not the top priority as in other programming languages. This all changed with the "tidyverse", a set of packages and tools that have a consistently structured programming interface.

When tools such as `dplyr` and `ggplot2` came to fruition, it made the learning curve much easier 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, which includes **a series of commonly used packages for data manipulation, visualization, iteration, modeling, and communication**. The end result is that R is now much easier to learn - Learn R From A Master Data Scientist's Code.

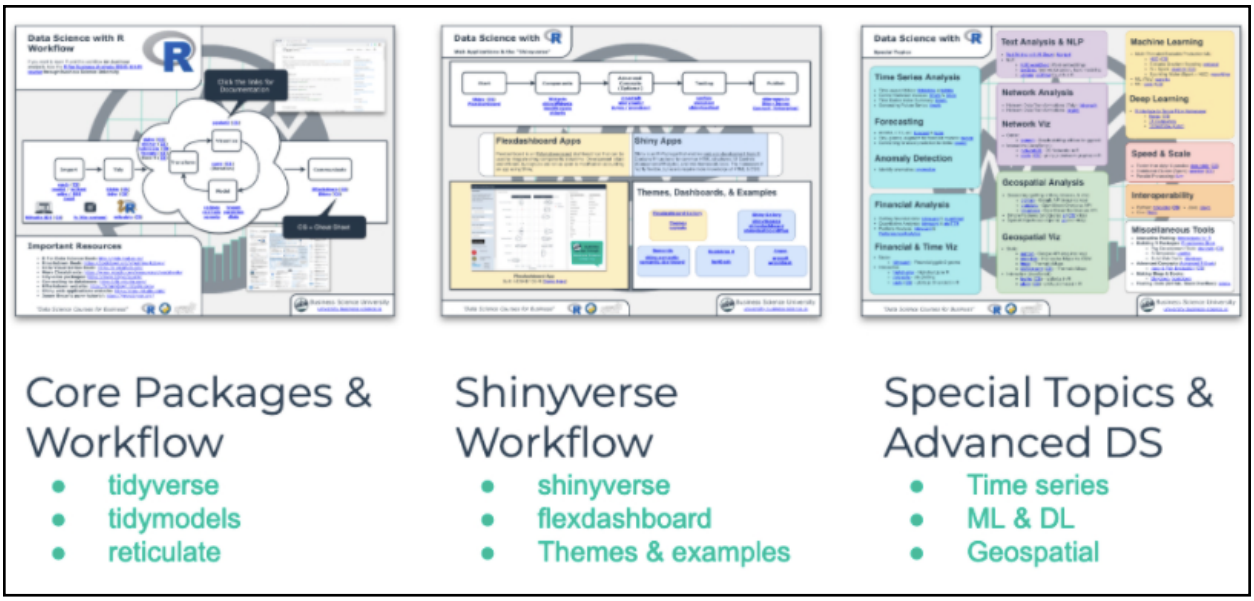


Source: tidyverse.org

R continues to evolve in a structured manner, with advanced packages that are built on top of the `tidyverse` infrastructure. A new focus is being placed on modeling and algorithms, which we are excited to see. Further, the `tidyverse` is being extended 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 2nd page of the R Cheat Sheet (FREE DOWNLOAD) links to all of the tools discussed next (and more tools beyond)!



The Ultimate R Cheat Sheet (Free Download)

🧠 R has brains

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!

These tools are used everywhere from AI products to Kaggle Competitions, and you can use them in your business analyses.

💪 R has muscle

R has powerful tools for:

- Vectorized Operations - R uses vectorized operations to make math computations lightning fast right out of the box
- Loops (`purrr`)
- Parallelizing operations (`parallel`, `future`)
- Speeding up code using C++ (`Rcpp`)
- Connecting to other languages (`rJava`, `reticulate`)

- Working With Databases - Connecting to databases (`dbplyr`, `odbc`, `bigrquery`)
- Handling Big Data - Connecting to Apache Spark (`sparklyr`)
- 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`)
- Programming (`purrr`, `tidyeval`)
- Communication (`Rmarkdown`, `shiny`)

Reason 5: R Is Built For Business

Two major advantages of learning R versus every other 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 can. The two capabilities we refer to are `rmarkdown` for report generation and `shiny` for interactive web applications.

Rmarkdown

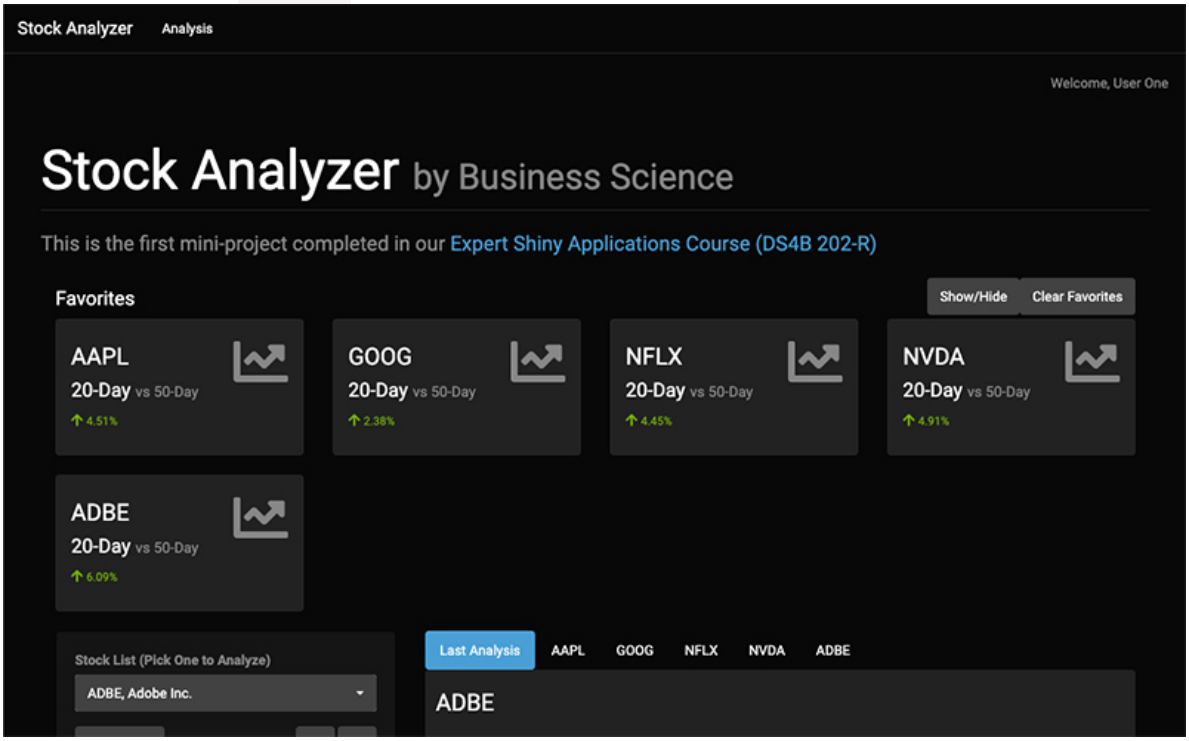
Rmarkdown is a framework for creating reproducible reports that has since been extended to building blogs, presentations, websites, books, journals, and more. It's the technology that's behind this blog, and it allows us to include the code with the text so that anyone can follow the analysis and see 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

Shiny is a framework for creating interactive web applications that are powered by R. Shiny is a major consulting area for us as 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. Here's an example of a Google Trend app built with `shiny`.



Shiny Web Apps Rule!!

Reason 6: R Community Support

Being a powerful language alone is not enough. To be successful, a language needs community support. We'll hit on two ways that R excels in this respects: CRAN and the R Community.

CRAN: Community-Provided R Packages

CRAN is like the Apple App store, except everything is free, super useful, and built for R. With over 17,000 packages, it has most everything you can possibly want 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!

Community Support

You begin learning R because of its capability, you stay with R because of its community. The R Community is the coolest part. It's tight-knit, opinionated, fun, silly, and highly knowledgeable… all of the things you want in a high performing team.

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. Here are a few that we attend and/or will be attending in the future:

- EARL - Mango Solution's conference on enterprise and business applications of R
- R/Finance - Community-hosted conference on financial asset and portfolio analytics and applied finance
- Rstudio Conf - Rstudio's technology conference
- New York R - Business and technology-focused R conference

A full list of R conferences can be found here.

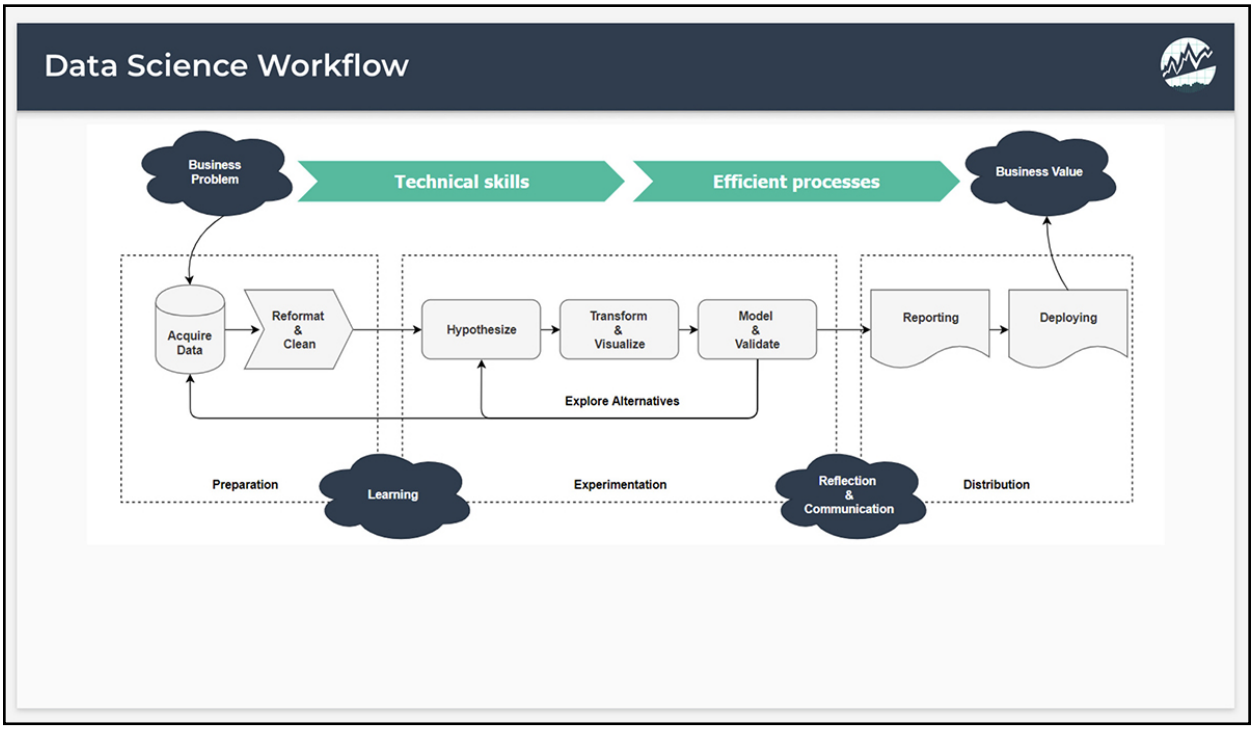
Meetups

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.

Conclusion

R has a wide range of benefits making it our obvious choice for Data Science for Business (DS4B). That's not to say that Python isn't a good choice as well, but, for the wide-range of needs for business, there's nothing that compares to R. In this chapter we saw why learning R is a great choice.

Chapter 5: Data Science Workflow - The Process for Solving Data Problems



Data Science is often misunderstood by students seeking to enter the field, business analysts seeking to add data science as a new skill, and executives seeking to implement a data science practice. This chapter aims to clear up the mystery behind data science by illustrating the sequence of steps to go from a business problem to generating business value using a data science workflow. **Once data science is understood, we can take steps to learn data science skills that will generate the most value and/or better make strategic investments in building a data science practice.**

Overview

In this chapter, you will:

- Understand what data science is
- Learn how data science generates value for an organization
- Learn how to go from business problem to business value

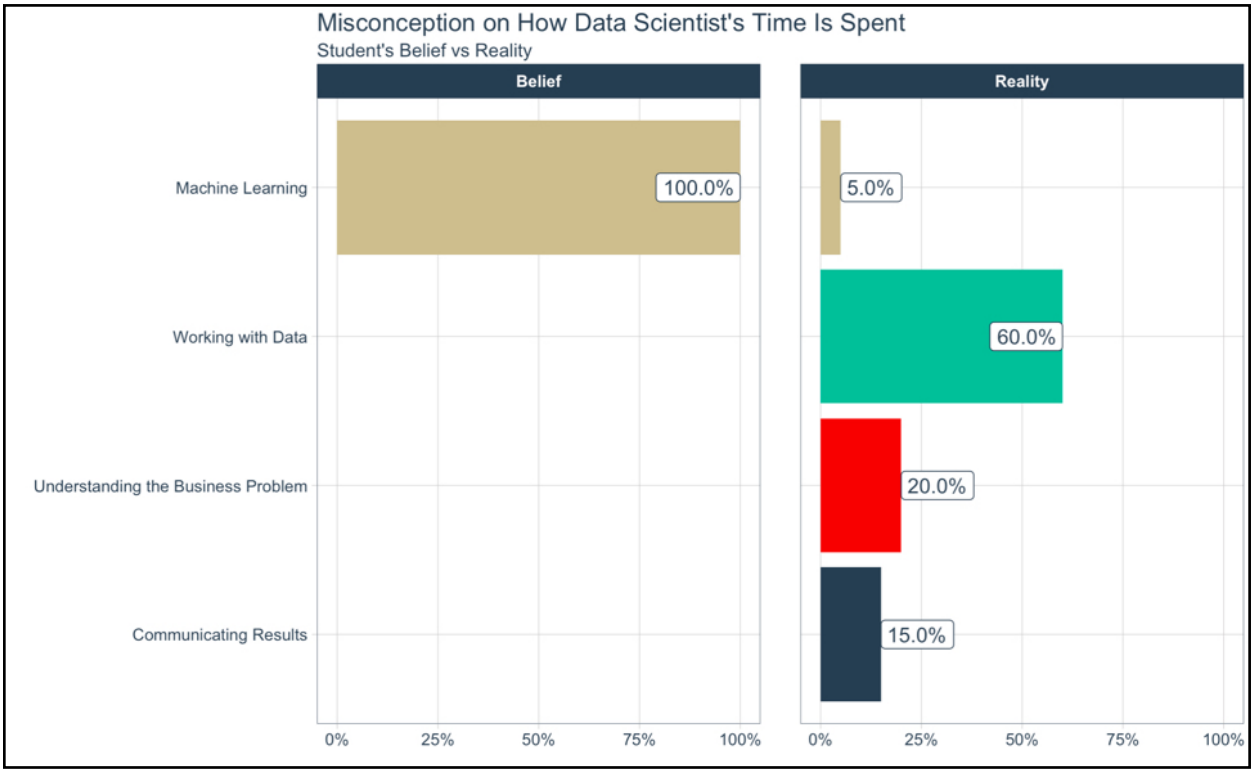
The Mystery & Confusion

Data Science is a mysterious term to many, but why?

Students & Data Enthusiasts

People excited about data science see it as machine learning - 100% of the time (this is drastically disproportionate to reality). In reality, Machine Learning (or Modeling) is about **5% of your time**. The rest of the time is spent:

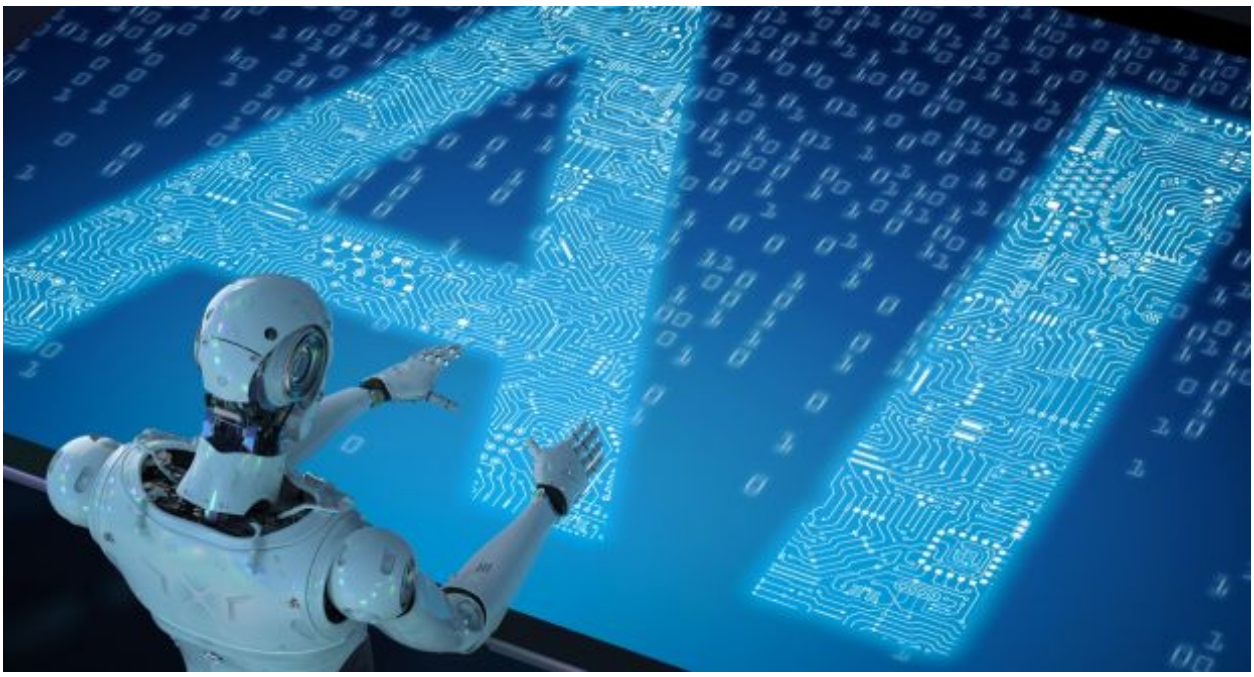
- **Understanding the Business Problem:** Communicating with Domain Experts (20%)
- **Working with Data:** Cleaning, Manipulating, Visualizing, Processing, Transforming, and Understanding (60%)
- **Communicating Results:** Reporting, Slide Decking, and Building Distributed Applications (Predictive Decision-Making Tools) (15%)



How Students View Data Science

Executives & Business Professionals

Executives and business professionals see data science as a new technology that could benefit their organization, but the connection between business problem and business value is not well understood. Data Science is often viewed as Artificial Intelligence (AI), a complex, black-box technology that is very trendy. But, the question remains - *"What Can AI Do?"*



How Executives and Business Professionals View Data Science

Reality for Everyone

Fortunately, the reality is that large businesses:

- **Have many customers** - The customers churn, generate sales, drive forecasts
- **Make many products and/or services** - The products are linked to quality, lead time, and inventory
- **Have many suppliers** - The suppliers affect lead times and serviceability
- **Have data** - The data provides a means to measure business drivers and is the fuel for data science

This combination of business-drivers - customers, products, inventory, suppliers, and more - with a wide array of internal and external data available **makes data science a competitive advantage** to organizations that can effectively implement it.

Making Better Decisions Generates Business Value

The **goal** for any Data Science Practice (Data Science Team) is to enable the rest of the organization to **make better, data-driven decisions**. Therefore, a Data Science Practice is a support role (similar to IT) that allows the organization to function better. The Data Science team can add a lot of value very quickly - through better decision making.

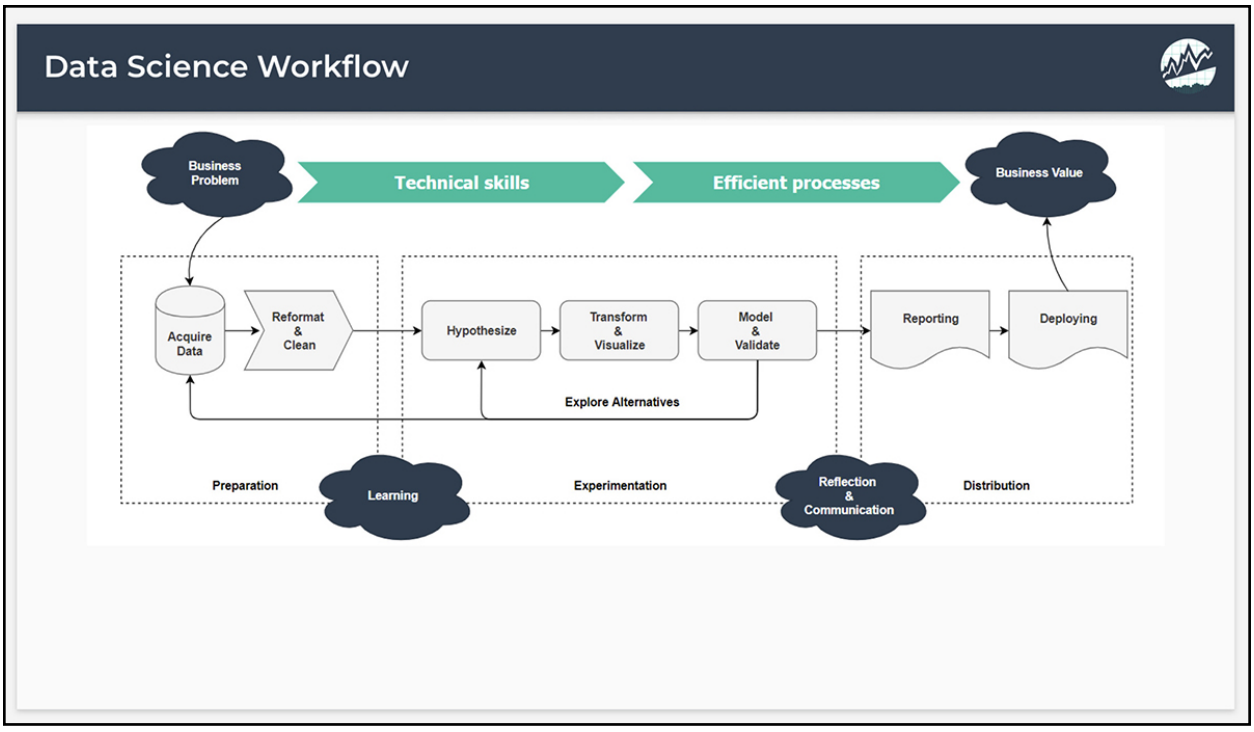
A simple example illustrates my point - **An organization that does $500M in annual revenue but has a customer churn rate of 10% loses out on $50M in revenue/year.** If a data science practice can identify the issue, predict which customers are going to churn, and implement strategies that enable the workforce to target the customers with retention strategies, the team can effectively reduce the churn rate 20%.

An organization that does $500M in annual revenue but has a customer churn rate of 10% loses out on $50M in revenue/year.

In monetary terms, **a reduction in churn of 20% equates to an annual savings of $10M**. Over 5 years, this is $50M in savings generated from the Data Science Practice working with the decision makers (e.g. Sales, Marketing, Production).

How Do We Go From Business Problem To Business Value?

The way to go from business problem to business value follows an iterative set of steps, I call the Data Science Workflow:

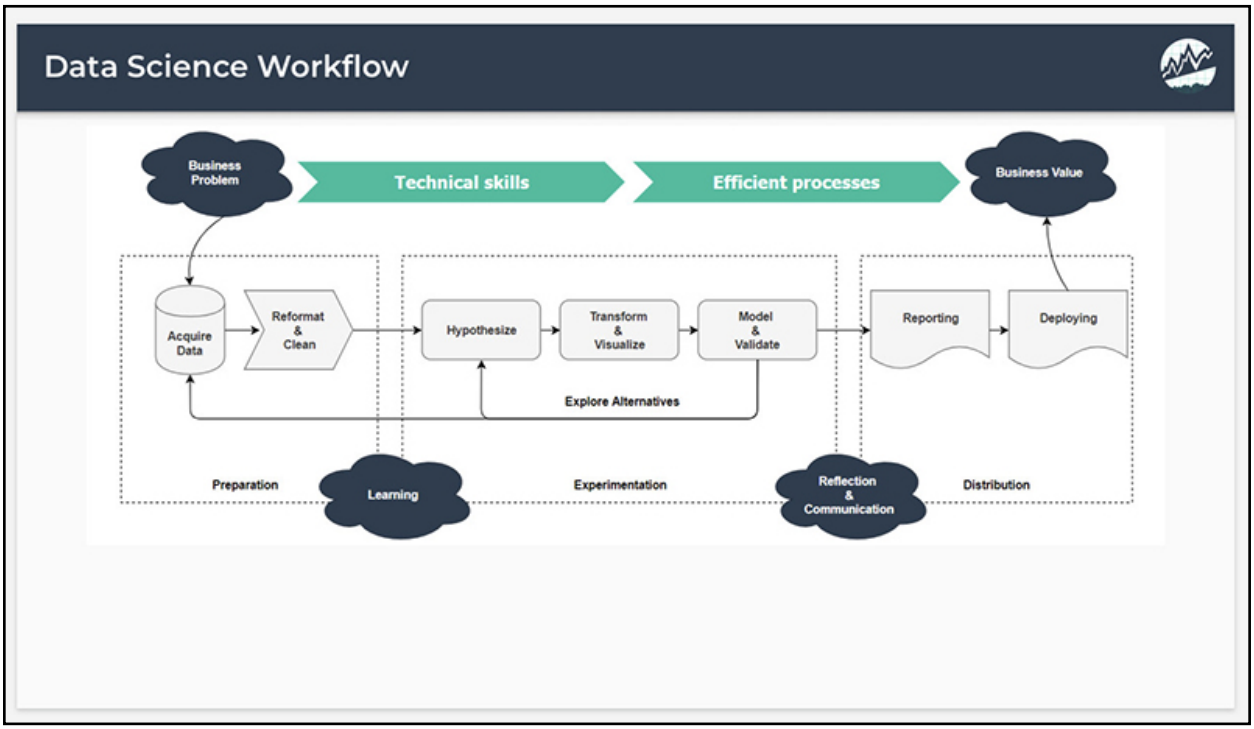


The **Data Science Workflow** has milestones (blue clouds), stages (dotted lines), and steps (gray shapes).

We begin with a **Business Problem** (milestone), where the team or organization identifies a problem that is worth solving. Typically this has a specific metric assigned to it that can be measured financially (e.g. 10% of our customers are not re-purchasing each year, this is costing the organization $50M annually).

The organization prioritizes this problem with the data science team, and they step into a project management workflow.

3 Stages of the Data Science Workflow



There are 3 stages:

1. **Preparation** - Data is collected and cleaned. This takes a significant amount of time because most data is unclean, meaning steps need to be taken to improve the quality and develop it into a format that machines can interpret and learn from.
2. **Experimentation** - This is where hypotheses are generated, data is visualized, and models are generated. This takes significantly less time than Preparation.
3. **Distribution** - Reports are generated documenting results, slide decks are created to present to management, and once management provides the go-ahead, apps are developed to implement decision making systems.

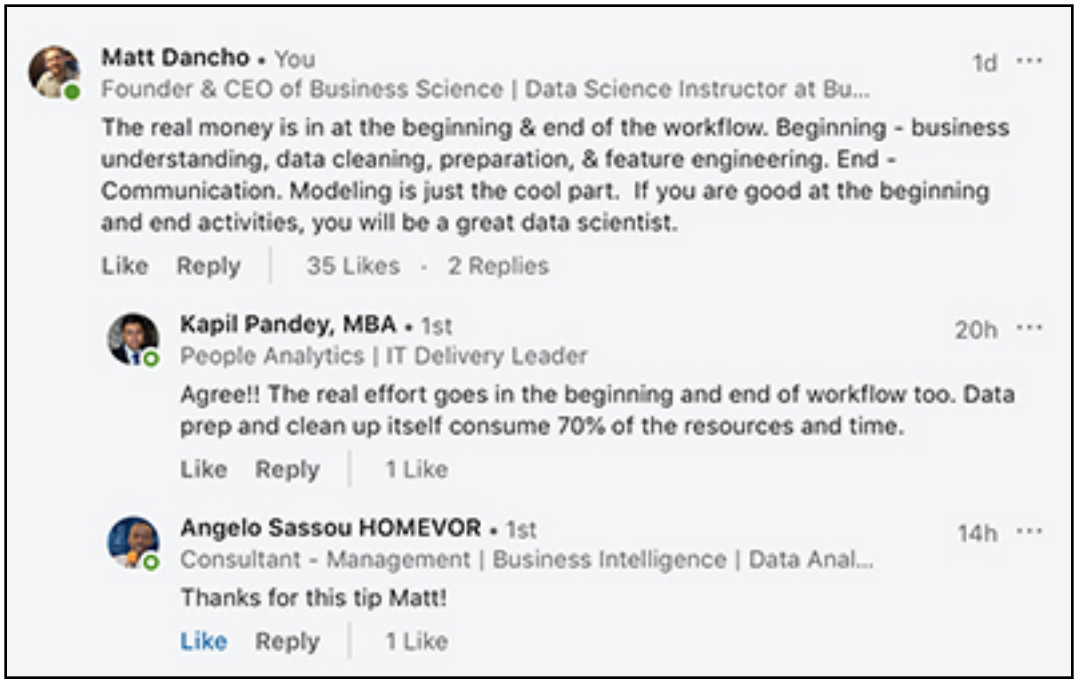
At the end of the workflow, data scientist's call this "production" or "deployment", and this is where **Business Value** (milestone) is generated.

The Best Data Science Teams Focus on These Parts

The best data science teams can iterate through this process going from problem to value very efficiently, spending little time on modeling and maximum time at the ends of the spectrum:

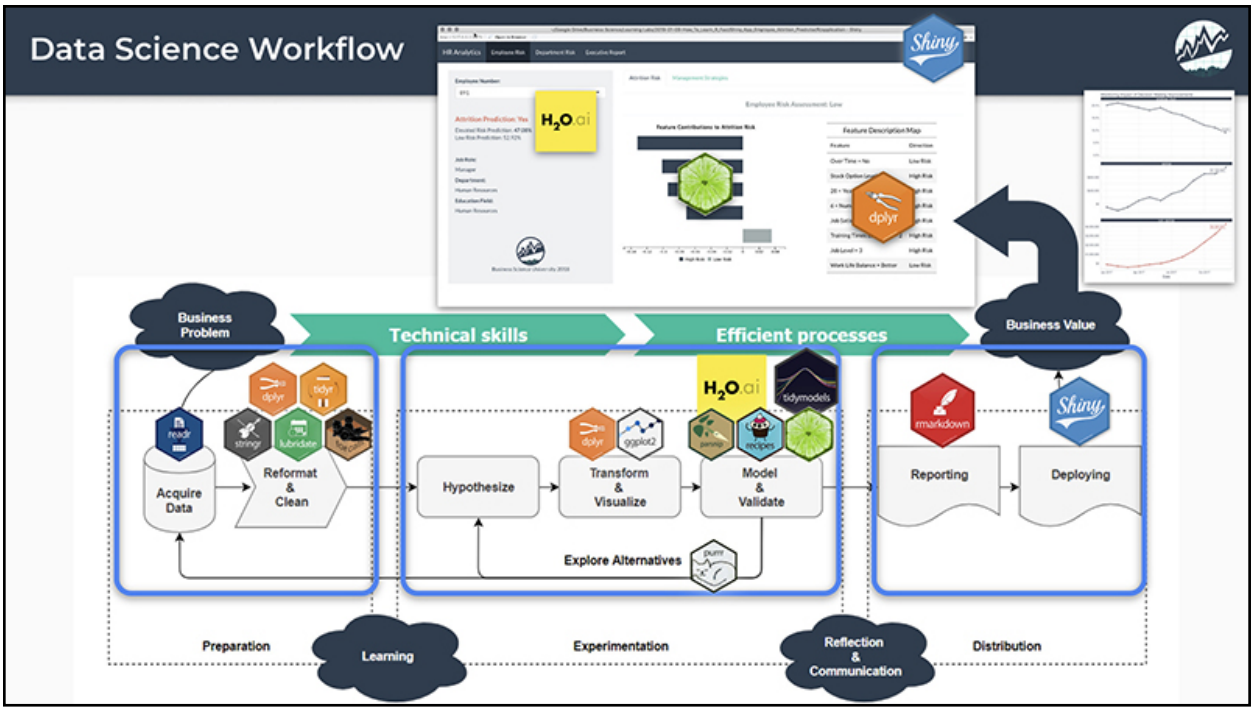
- **Beginning of Workflow:** Business Understanding / Domain Expert Communication, Data Understanding, Data Quality, and Feature Engineering are Critical

- **End of Workflow:** Communication with Project Stakeholders, Product Delivery are Critical



Learn How to Implement the Data Science Workflow

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 (I teach the *Data Science Workflow*):



Primary R Packages Overlayed on the Data Science Workflow

Data Science Methodologies you need to know

Several data science workflow and project management methodologies exist. The two that I use at *Business Science University* are:

- CRISP-DM
- BSPF Framework

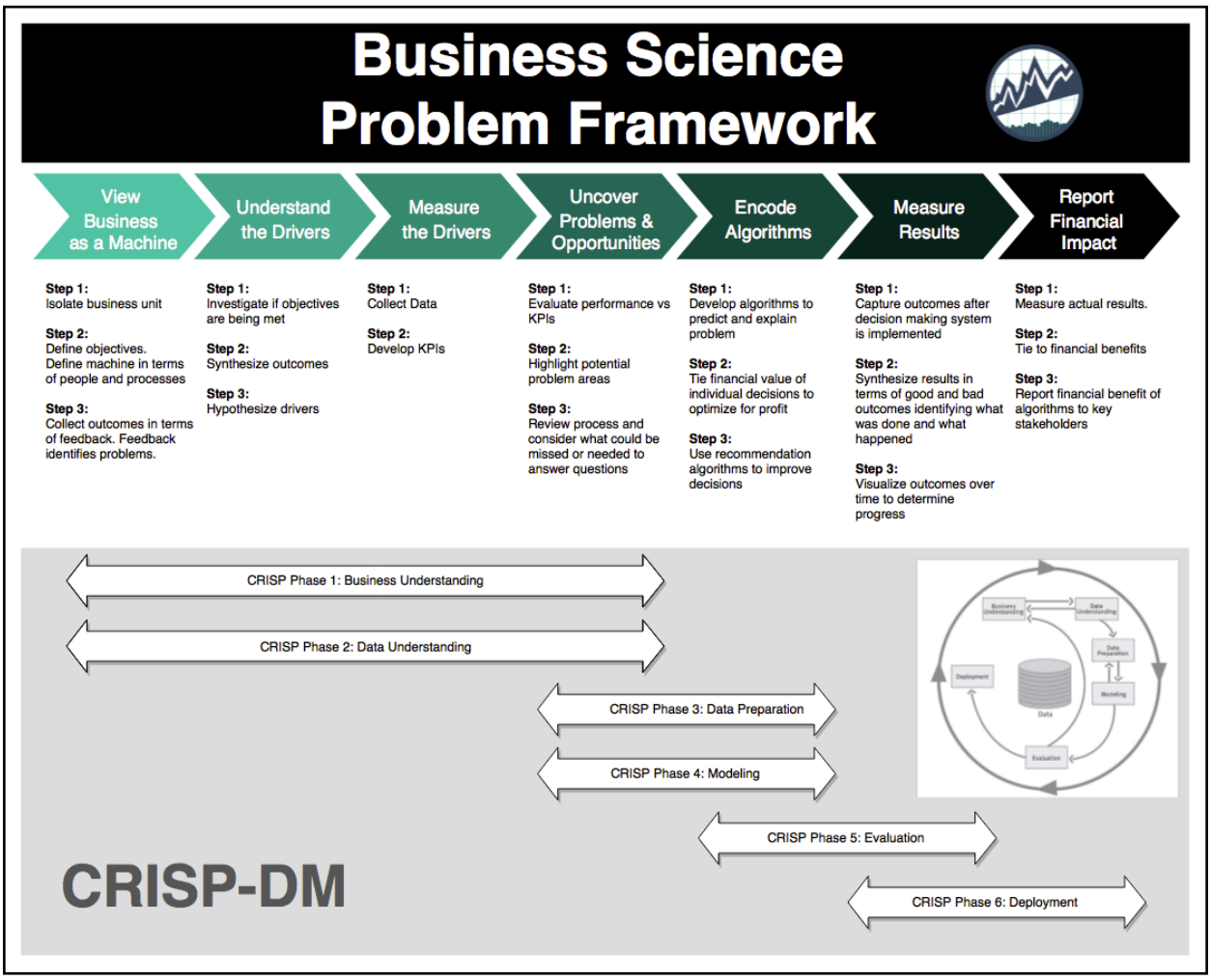
CRISP-DM

The ***CRISP-DM* - *Cross-Industry Standard Process for Data Mining*** - Is a general approach to performing data science projects. It's cross-industry, which means it is compatible with almost any data science problem. One issue is that CRISP-DM is very general, which is why I created the ***BSPF Framework (discussed next)***.

CRISP-DM Framework

BSPF

The Business Science Problem Framework is the core of solving business problems with data science. It helps us connect stakeholders, calculate costs, and show financial benefits of the project.



Business Science Problem-Solving Framework

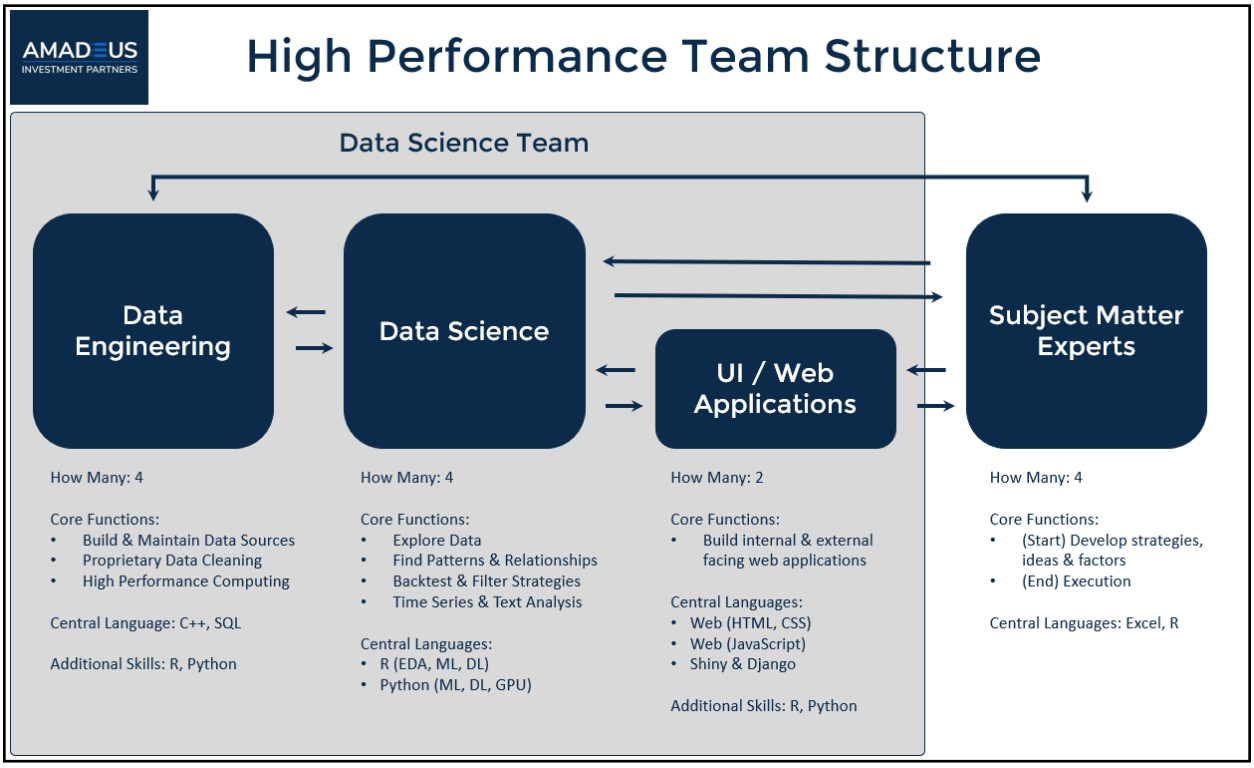
The Business Science Problem Solving Framework is also an interview secret helping my students land jobs!

"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 completing 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

Knowing how to connect data science to the business can help you land a data science job at a Fortune 500 company

Chapter 6: (Case Study) How To Build A High Performance Data Science Team

Written by Matt Dancho and Rafael Nicolas Fermin Cota.



High Performance Team Structure

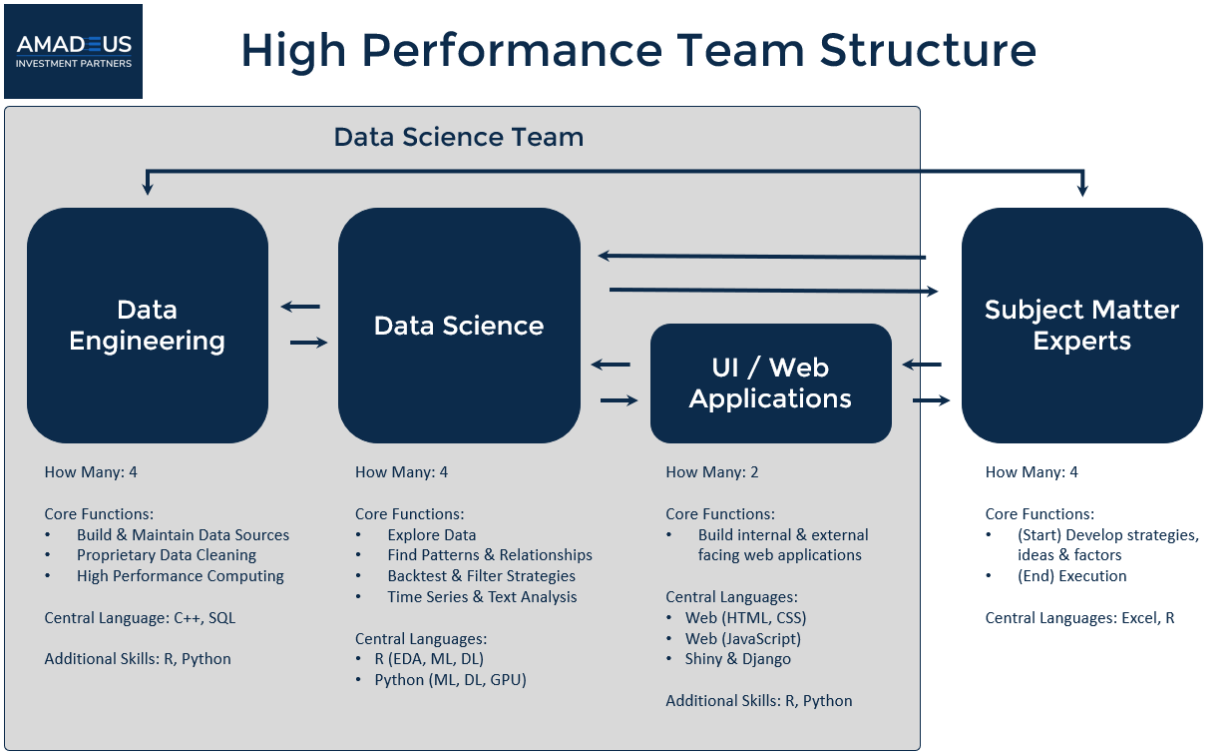
Artificial intelligence (AI) has the potential to change industries across the board, yet few organizations are able to capture its value and realize a real return-on-investment. The reality is that the transition to AI and data driven analysis is difficult and not well understood. The issue is twofold, first, the necessary technology to complete such a task has only recently become mainstream, and second, most data scientists are inexperienced in their respective industries. However, with all the uncertainty surrounding this topic, one hedge fund has managed to navigate through these challenges and accomplish what many companies are failing to do: **building a high-performing data science team that achieves real return-on-investment (ROI)**.

This is the story of an outlier

Business Science was recently invited inside the walls of *Amadeus Investment Partners* (now OneSixtyTwo Digital Capital), a hedge fund that has unlocked the power of artificial intelligence to gain superior results in one of the most competitive industries in the world: investments. Amadeus Investment Partners has spent the last five years building a high

performance data science team. What they have built is nothing short of extraordinary.

In this chapter, we will discover what makes Amadeus Investment Partners (now OneSixtyTwo Capital) an outlier and why they are unique in the data science space. We will learn the key ingredients that provide Amadeus a recipe that is driving ROI with artificial intelligence and examine what it takes to assemble a high-performance data science team.



Data Science Team Structure, Amadeus Investment Partners

We will then describe how *Business Science* is using this information to develop best-in-class data science education. We will show how we are integrating the exact same cutting-edge technology into our data science for business programs.

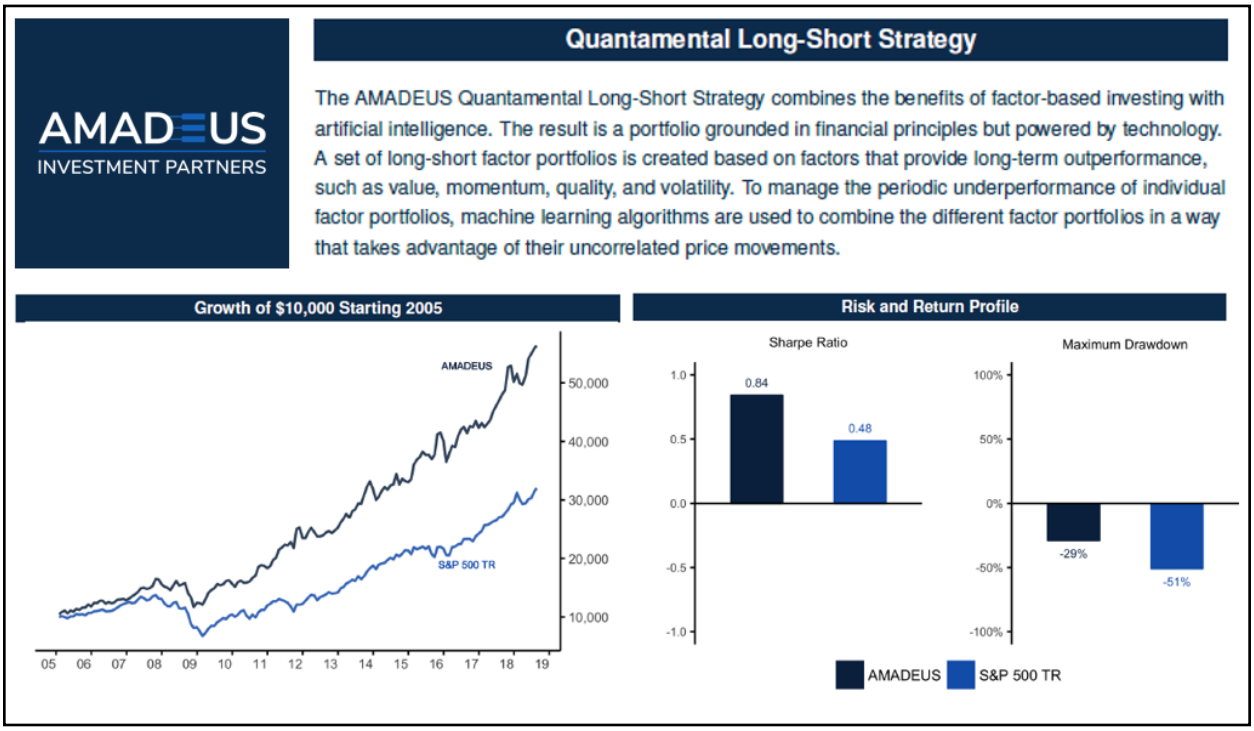
This is all aimed at one thing: ***developing a system for creating best-in-class data science teams***.

If you are interested in developing a best-in-class data science team, then read on.

Examining An Outlier

Amadeus Investment Partners (now OneSixtyTwo 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.

Amadeus' 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 Amadeus' strategy delivered "alpha", which means the strategy generated excess returns (performance) beyond the returns of the benchmark.



Risk-Return Performance, Amadeus' Quantamental Long-Short Strategy

From the ***Growth of $10,000 Starting 2005*** chart, *Amadeus* 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 Amadeus 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 Amadeus is delivering exceptional returns while taking on less risk, which is very attractive to investors.

But, how is *Amadeus* achieving these results?

A Radically Different Organization

In our meetings with Amadeus, we found ***3 key components*** to the high-performance data science team. Each of these are critically important to Amadeus' successful execution of their radically-different data-driven strategy. Amadeus:

1. Finds and trains talent in the most unlikely fashion
2. Has a well-designed team structure and culture
3. Provides access to cutting-edge technology

We will step through each of these key ingredients that make up the data-driven recipe for success.

Key 1: Finding and Training Talent in the Most Unlikely Fashion

The first key to the puzzle is finding and developing the talent to execute on the vision. That's where *Amadeus* has excelled: ***finding talent in the most unlikely places***.

Over the past several years, *Amadeus* has tactically been working with the leading educational institutions in Canada to selectively gain access to top students in…

Business Programs

Yes - Students that are top in their classes in ***Business Programs***. If you take a 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 *Amadeus*' 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. Amadeus subsequently converts these business-minded people to data scientists by augmenting their skillset with math and programming on the job.

*"Hiring people with the relevant **subject matter expertise** ensures that the right questions are asked."*

-Rafael Nicolas Fermin Cota

In terms of training the hired talent, Amadeus 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

It's this ***spirit of learning and critical thinking*** that you experience when meeting with the Amadeus data science team. What you also take away is a structured approach to this intellectual curiosity. Each member told stories of their start at Amadeus. 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.

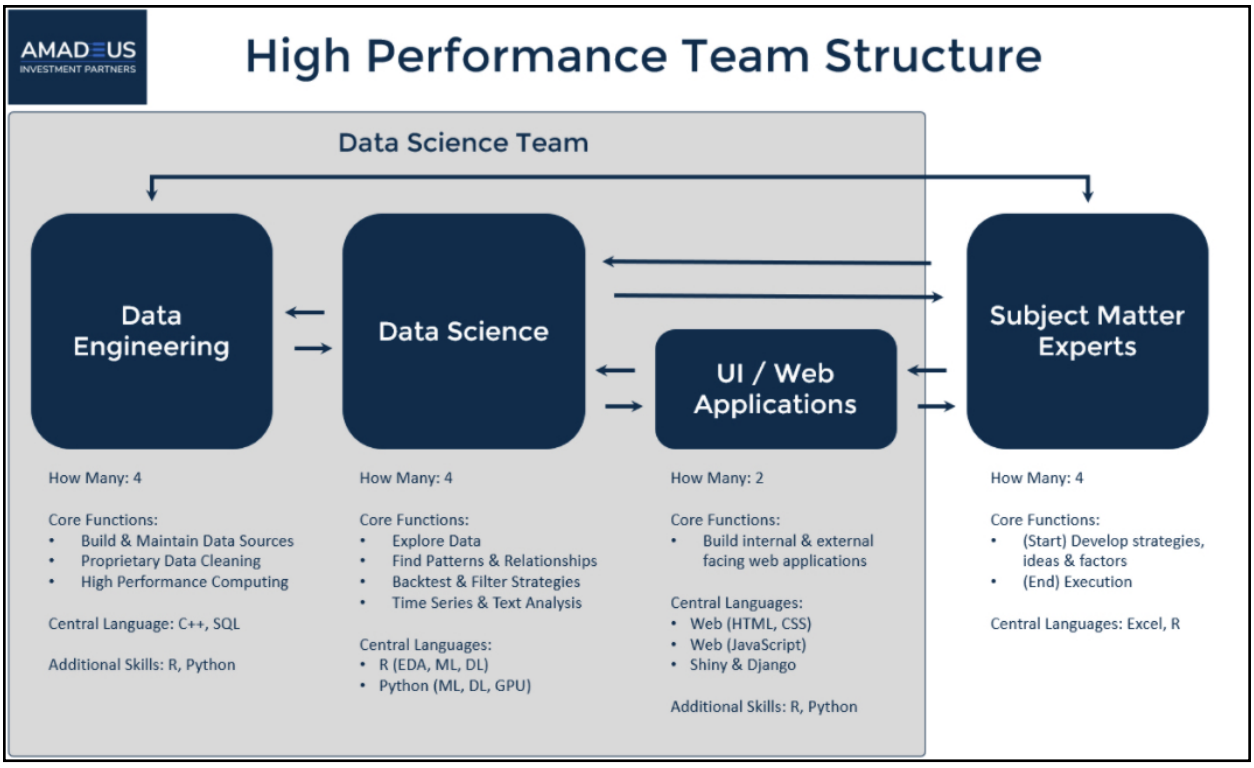
This core training ensures a common body of knowledge that team members draw from during discussions, making the communication process much more efficient.

To continue the education and professional development of the team members, everyone is free to purchase any books, courses, or other training material as needed.

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 Amadeus' 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
2. Data Engineering Experts
3. Data Science Experts
4. User Interface Experts

Subject Matter Experts (SME)

Amadeus 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 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

Data Engineering Experts (DEE)

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, Amadeus 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 Amadeus 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

Data Science Experts (DSE)

The DSEs at *Amadeus* 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.

User Interface Experts (UIE)

Amadeus 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 perform well when it comes to interactivity. 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.

Emphasis on Communication

An often overlooked part of a data science team is the team aspect, which requires communicating ideas and analyses through the workflow. For most other 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 for the detriment of the organizational performance.

At *Amadeus*, 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 functional parts described above. This way, the projects can benefit from the different perspectives of team members and the research process is streamlined without conflicts between each stage.

Also, all-hands weekly meetings are organized to keep each other up to date on individual progress and create a forum for team members to share insights and suggestions.

Key 3: Access to Cutting-Edge Technology

As mentioned above, it takes tremendous effort to find and train talent and have them work collaboratively. At this point, all of this effort would be futile if there was a technological bottleneck in the research process.

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 all team members immediate access to high-performance computational resources to minimize the time spent waiting for computations to run. This enables the team to quickly iterate through ideas.

At *Amadeus*, each team has their own computation stack 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 UIEs can use the DSEs' 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.

Amadeus 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.

Business Science witnessed *Amadeus*' data science team train a text classifier on financial news data for predicting article sentiment. The NVIDIA DGX-1 produced results in a matter of minutes, what would have taken several hours if not days on a CPU system or even a GPU system that is not optimized for deep learning.

Best-In-Class Data Science Education

Turning Insights Into Education

Business Science has gained the following insights from the Amadeus case study:

1. Hiring talent with ***subject matter expertise*** and ***subsequently educating them in data science*** has proven to be effective in building a high performance team
2. Communication among different teams is important, and ***education needs to support communication among the different teams***
3. The teams need to be ***equipped with the latest technology*** to reach full potential

Unfortunately, data science education is still in its infancy because most educational institutions don't understand what it takes to do real-world data science. Most programs focus on theory or tools. This doesn't work. Learning how to do real-world data science only comes from application and integration, and those with an understanding of the business have an advantage.

This is why Business Science is different.

I have built a best-in-class educational program that incorporates learnings:

- Through ***studying an outlier*** - A radically different data science team of the highest caliber that is successfully generating ROI for their organization.
- Through our own ***applied consulting experiences*** that have successfully generated ROI for organizations
- Through ***experience building the tools and software*** needed to solve business problems

And this is why my students get results and are consistently placed into Fortune 500 firms, fast growth-startups, in many industries around the world.