

Kotlin for Data Science



Thomas Nield

[@thomasnield9727](https://twitter.com/thomasnield9727)

Agenda

Kotlin for Data Science

- What is Data Science?
- Challenges in Data Science
- Why Kotlin for Data Science?
- Example Applications
- Getting Involved



Thomas Nield

Business Consultant at Southwest Airlines

Author

- *Getting Started with SQL* by O'Reilly
- *Learning RxJava* by Packt

Trainer and content developer at O'Reilly Media

OSS Maintainer/Collaborator

RxKotlin

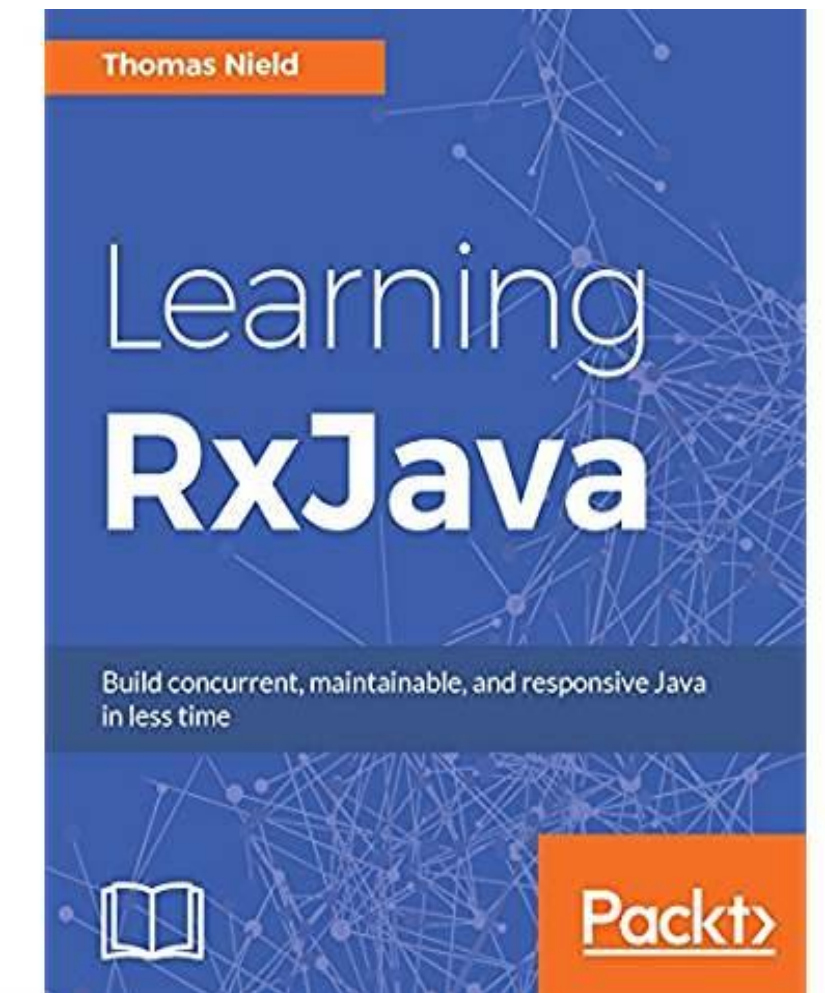
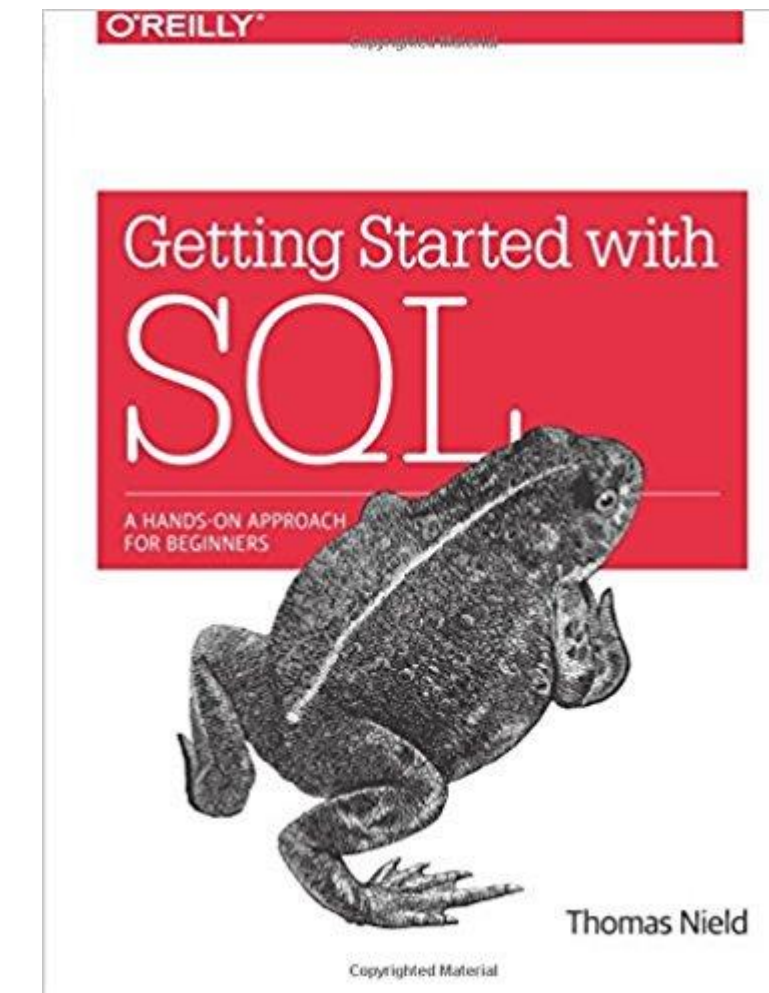
TornadoFX

RxJavaFX

Kotlin-Statistics

RxKotlinFX

RxPy



 [thomasnield9727](https://twitter.com/thomasnield9727)

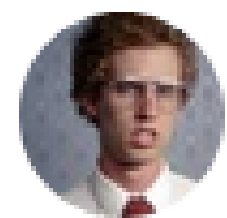
 <https://github.com/thomasnield>



What is Data Science?

A Quick Overview

Not Data Science



I Am Devloper

@iamdevloper

You say: "We added AI to our product"
I hear: "We added a bunch more IF
statements to our codebase"

9:07 AM - 10 Feb 2017

3,451 Retweets 5,730 Likes



46 3.5K 5.7K



Thomas Nield

@thomasnield9727

Replying to @iamdevloper

"The AI is smart enough to recognize several
variants of a text keyword". I used a regular
expression.

8:36 PM - 10 Feb 2017

3 Retweets 23 Likes

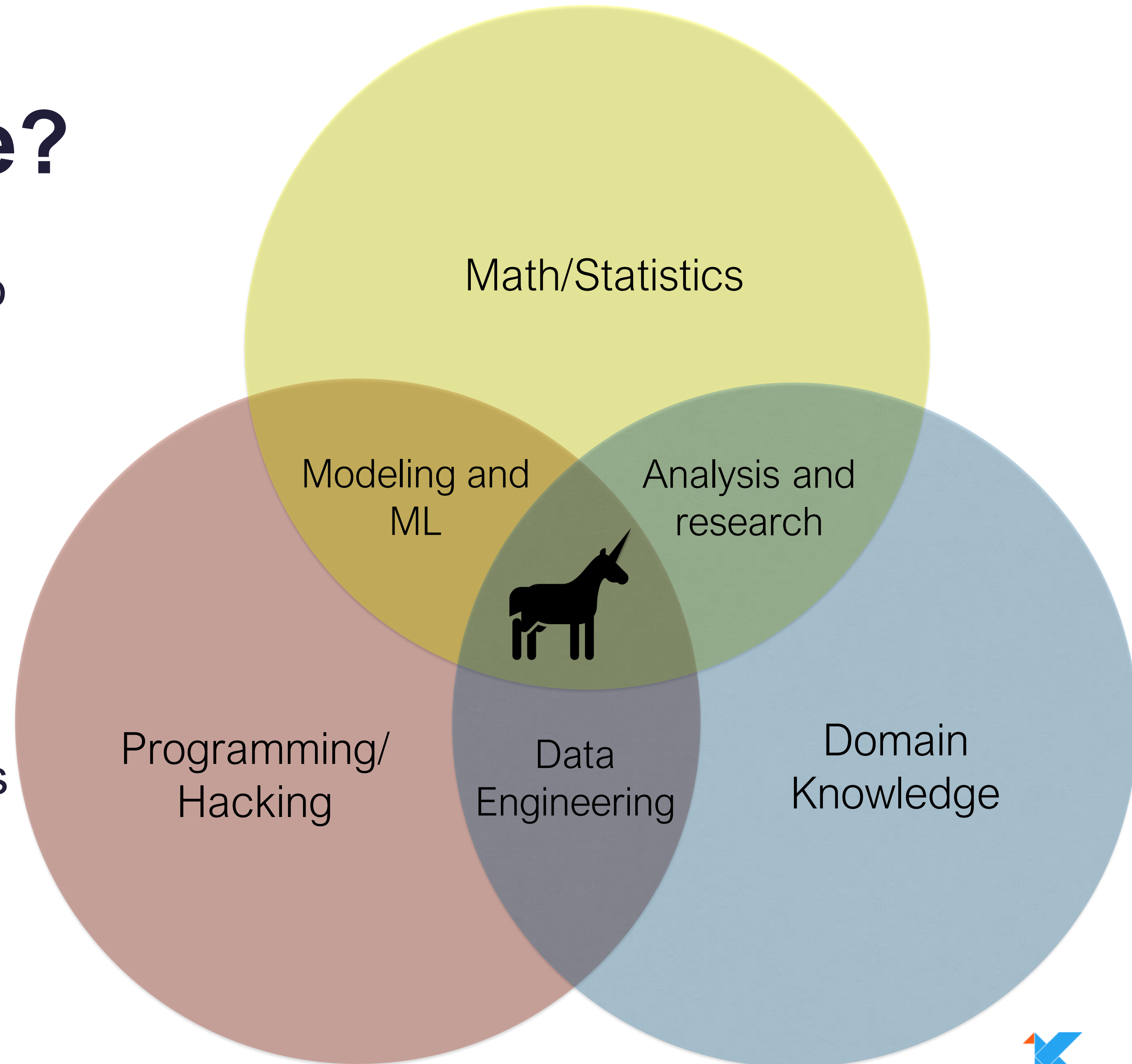


3 23

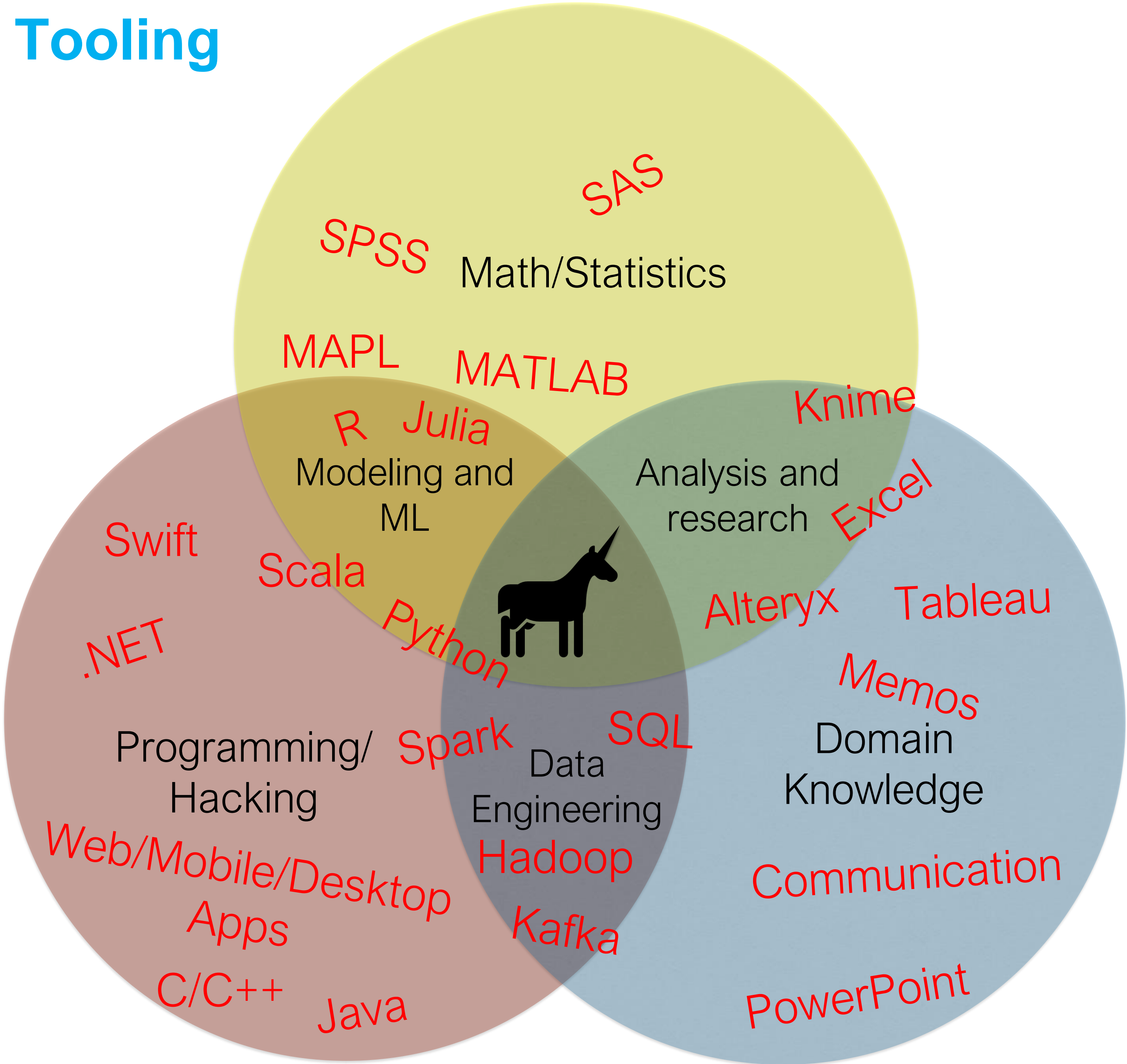


What is Data Science?

- Data science attempts to turn data into insight.
- Insight can then be used to aid business decisions or create data-driven products.
- A strong data science professional has some mix of programming/hacking, math/statistics, and business domain knowledge.



Data Science Tooling



Data Scientist Archetypes

A Subjective Categorization

The Statistician – Summarizes data using classic statistical methods and probability metrics.

The Mathematician – The individual who solves a problem by converting it into sea of numbers, often in the form of vectors and matrices.

The Data Engineer – An architect of “big data” solutions who can create reusable pipelines of data transformations and share it through reusable API's.



Data Scientist Archetypes

A Subjective Categorization

The ML Scientist – A more advanced mathematician who leverages machine learning, neural networks, and other forms of AI modeling.

The Programmer – A trained software developer who likely knows Scala, Java, or Python, and often creates code from scratch tailored to specific business problems.

The Bard – The person who crafts communications about data findings with leaders and stakeholders, often telling stories with memos, charts, PowerPoints, infographics, spreadsheets, and other visual tools.



What is a Model?

Concoction of Math and Code

What is a model? – A code representation of a problem, often mathematical in nature, that offers a solution in some form.

Examples of models:

- A linear programming system that finds optimal values for business decision variables.
- Machine learning model that clusters customers based on their attributes.
- AI that parses, interprets, and links legal documents.
- Neural network that identifies or categorizes images or natural language.



Data Science Challenges



MODEL \neq PRODUCT



Data Science Challenges

Models Are Not Products

A current struggle in data science is putting models into production.

- A model is often a hacky Python or R script that simply does not plug into a large enterprise technology ecosystem (which is often built on Java or .NET).
- Models often use dynamically typed languages with tabular data structures and procedural code which is difficult to modularize, test, evolve, and refactor.
- If a model starts to break down and produce errors, it can bring into question the data scientist's credibility.



Data Science Challenges

Models Are Not Products

Models often need to be rewritten from scratch as software:

- Software engineers often need to rewrite a model from Python or R to Java.
- The model needs to be “opened up” so its inner workings can be presented in frontend software.
- The engineer may even have to introduce production data to the model, as the model may only have been tested with dummy data.
- The production code also needs to be architected for scalability, refactorability, code reuse, and testing.



Twitter

“There was only one problem—all of my work was done in my local machine in R. People appreciate my efforts but they don’t know how to consume my model because it was not “*productionized*” and the infrastructure cannot talk to my local model. **Hard lesson learned!**”

- Robert Chang, Data Scientist at Airbnb (formerly Twitter)



Stitch Fix

“Data scientists are often frustrated that engineers are slow to put their ideas into production and that work cycles, road maps, and motivations are not aligned. By the time version 1 of their ideas are put into *[production]*, they already have versions 2 and 3 queued up. Their frustration is completely justified.”

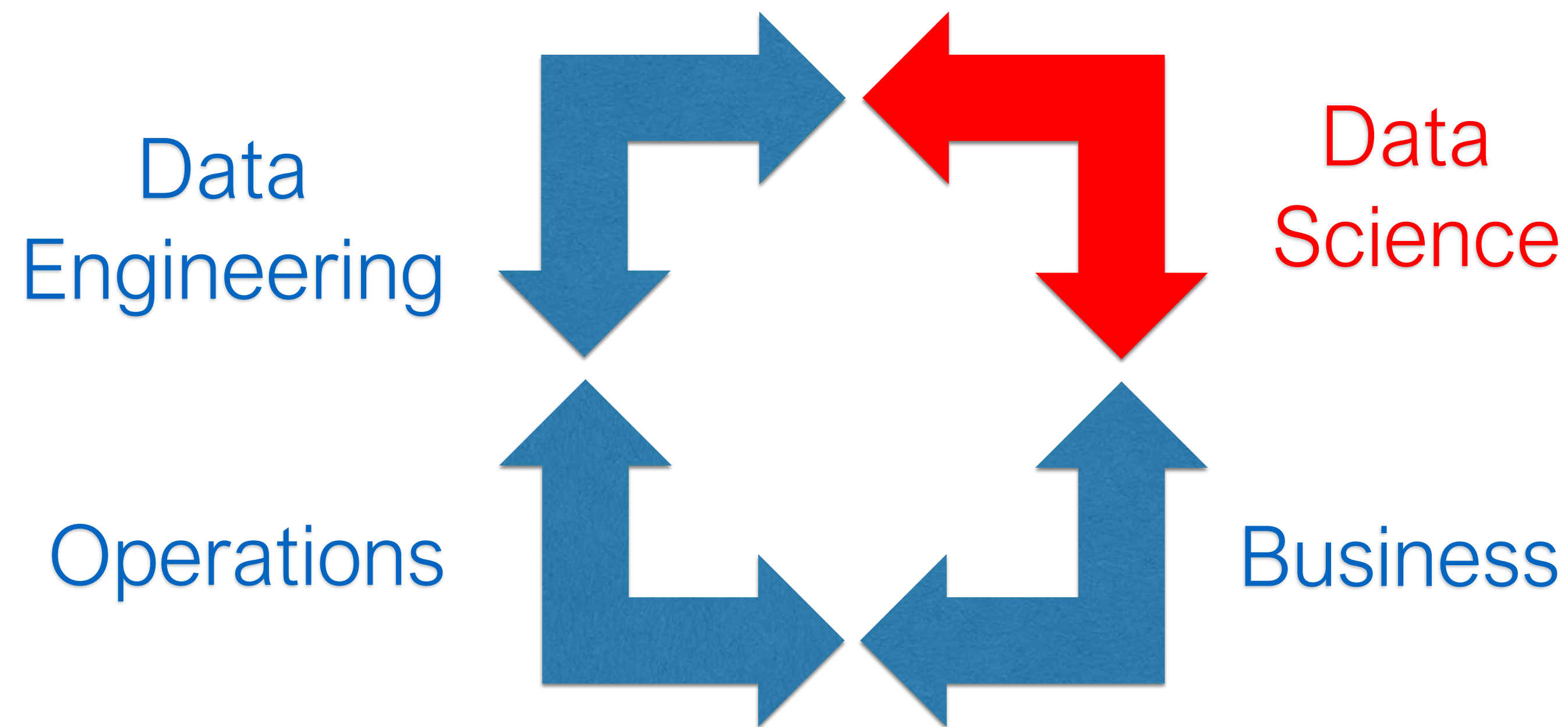
- Jeff Magnusson, Director of Data at Stitch Fix



Slack

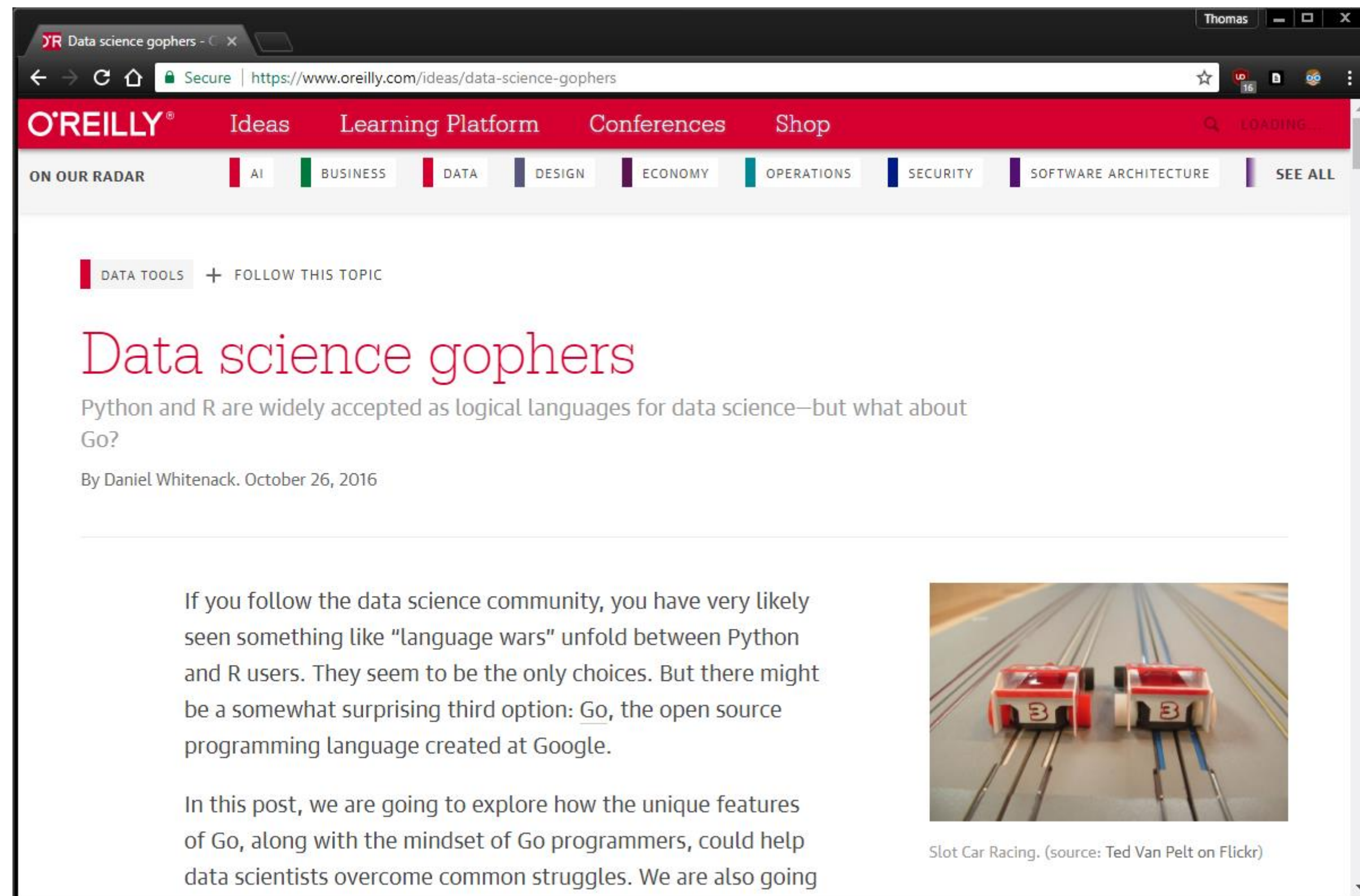
“The infinite loop of sadness.”

- **Josh Wills**, Director of Data Engineering



Recommended Reading

Data Science Gophers



<https://www.oreilly.com/ideas/data-science-gophers>



Why Kotlin for Data Science?



What is the Solution

Kotlin, of course!



Data scientists who code often need the following:

- Rapid turnaround, quick iterative development
- Easy to learn, flexible code language
- Mathematical and machine learning libraries

Experienced software engineers often want the following:

- Static typing and object-oriented programming
- Production-grade architecture and support
- Refactorability, reusability, concurrency, and scaling

Kotlin encompasses all the qualities above, and can provide a common platform to close the gap between data science, data engineering, and software engineering.

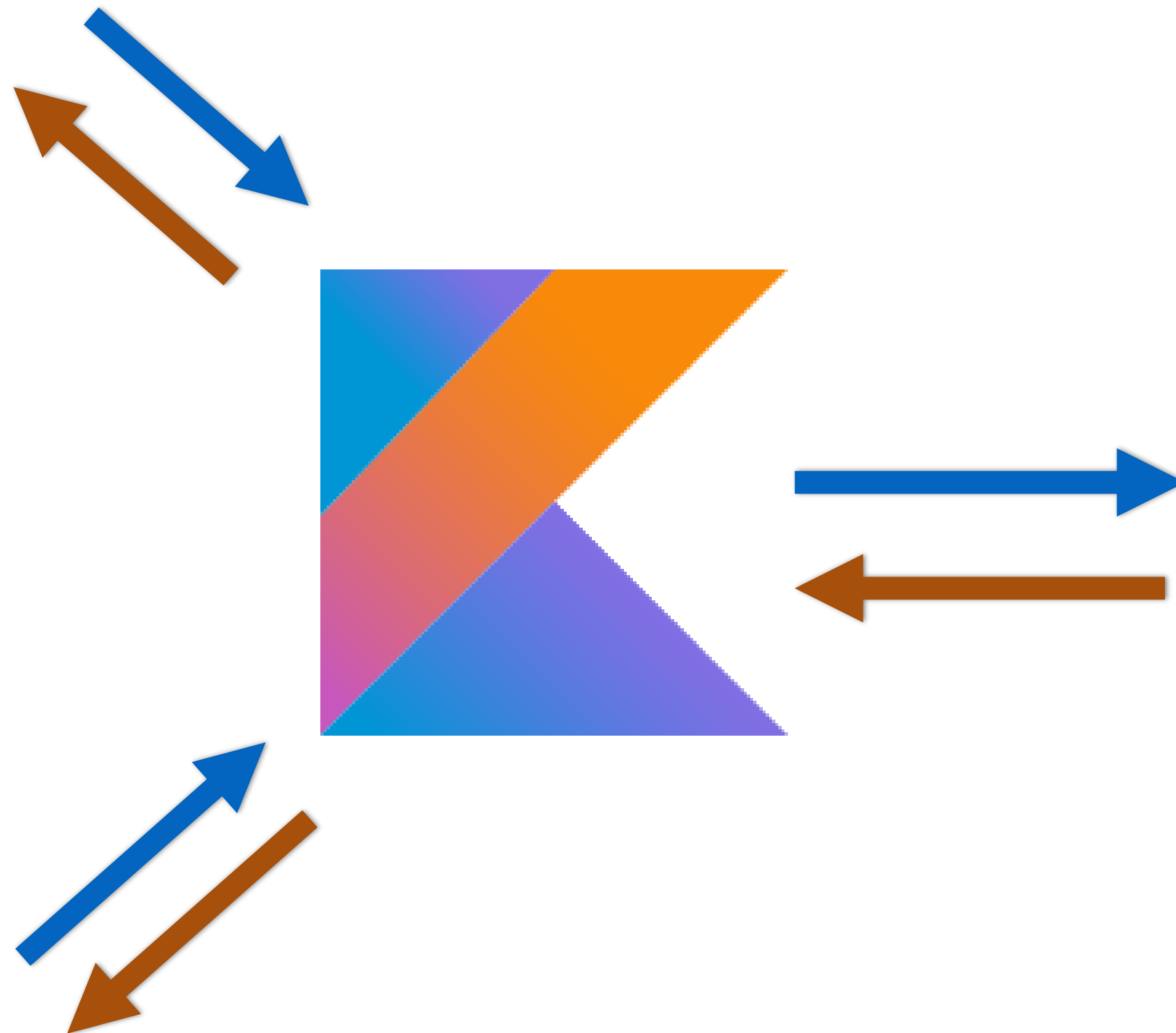


One language, One Codebase, One Platform

Data Engineer

Data Scientist

Software Engineering/
Dev Ops



Kotlin vs Python

Static vs Dynamic

Python is a powerful, flexible platform with a simple syntax and rich ecosystem of libraries.

Dynamic typing makes Python flexible for ad hoc analysis, but it is challenging to use in production.

- Dynamic types allow improvised data structures to be defined at runtime.
- Dynamic typing can quickly create difficulties in maintaining, testing, and debugging codebases, especially as the codebase grows large.



Kotlin vs Python

Static vs Dynamic

Kotlin, like Scala, embraces immutability and static typing.

- Data structures are explicitly defined and enforced at compile time, not runtime.
- While static typing is traditionally verbose, Kotlin manages to make it concise in a Pythonic manner.

Kotlin may not have as many mainstream data science libraries like Python, but it has comparable ones in the Java ecosystem:

Apache Spark
Apache Hadoop
TensorFlow
Apache Kafka

ND4J
Weka
Java-ML
Krangl

DeepLearning4J
Apache Commons Math
Kotlin Statistics
Komputation

ojAlgo!
Koma
H2O
EJML



Kotlin vs Scala

Pragmatism vs Features

Scala has seen success in adoption on the data science domain, arguably due to Apache Spark and other “big data” solutions.

However, Scala *might* have some challenges going forward.

- Apache Spark is being interfaced in other languages like Python and R to make it accessible.
- Computation engines and libraries are increasingly moving back to C/C++, and away from JVM.
- Plethora of features = Good or overwhelming?



Kotlin vs Scala

Pragmatism vs Features

Scala not taking significant share from Python may present an opportunity for Kotlin.

- Kotlin might be able to finish what Scala started, establishing an engineering-grade coding platform for data science.
- Compared to Scala, Kotlin has easier interoperability with Java.
- Kotlin encompasses many of the best ideas from Scala, but strives to be simpler in its features and be more accessible (e.g. “Pythonic”).
- While computation engines are unlikely to be dominated by Kotlin implementations, Kotlin can be effective in interfacing with them.



Weaknesses of Kotlin

For Data Science

Platform Drawbacks

- **Not Dynamically Typed** – Data structures have to be explicitly defined, which can add additional steps in working with data.
- **Numerical Efficiency** – Boxing of numbers might hurt performance without ND4J or other low-level computation libraries.

Libraries and Tooling

- **Ad Hoc Analysis** – Casually exploring data without a clear objective may be challenging without data frame libraries like Krangl.
- **Libraries** – Breadth of data science libraries, while decent, does not match Python or R.
- **Documentation** – Java libraries use Java (not Kotlin) in their documentation.



Strengths of Kotlin

For Data Science

Platform Strengths

- **Accessibility** – Easy to learn and intuitive, few esoteric features.
- **Minimal boilerplate, fast turnaround** – “Pythonic” productivity
- **Interoperability with Java** – Plugs into enterprise Java ecosystems

Language Features

- **Data classes** – No more tuples or improvised data structures at runtime.
- **DSL** – Create streamlined languages for domain-specific logic.
- **Static Typing** – Benefits of OOP and static typing, without the verbosity.
- **Nullable Types** – Helpful asset in data wrangling.
- **Function Syntax** – Flexible, expressive function features including extensions.
- **Lambdas and Pipelines** – Practical functional programming constructs.



Example Applications

Linear Programming

A Word Problem

You have three drivers who charge the following rates:

- Driver 1: \$10 / hr
- Driver 2: \$12 / hr
- Driver 3: \$15 / hr

From 6:00 to 22:00, schedule one driver at a time to provide coverage, and minimize cost.

Each driver must work 4-6 hours a day. Driver 2 cannot work after 11:00.



Stay Calm

Math = Powerful Apps

S_i = Shift start time for each i driver

E_i = Shift end time for each i driver

R_i = Hourly rate for each i driver

δ_{ij} = Binary (1,0) between two ij drivers

M = Length of planning window

Minimize:

$$\sum R_i(E_i - S_i)$$

Constraints:

$$4 \leq E_i - S_i \leq 6$$

$$16 = \sum E_i - S_i$$

$$E_2 \leq 11$$

$$S_i \geq E_j - M\delta_{ij}$$

$$S_j \geq E_i - M(1 - \delta_{ij})$$



Data-Driven Apps

Endless Possibilities



Just the subject of linear programming alone opens up a large domain of apps:

- Schedule generation (e.g. classrooms, transportation, staff)
- Operations and resource planning (e.g. construction, factory planning)
- Blending problems (e.g. financial portfolios, food/drink ingredients)

Kotlin makes it easier than ever to make a model a polished product.

Kotlin is capable of solving a wide array of problems for many data science topics.



Getting Involved

Getting Involved

Help Bring Kotlin to Data Science

To help bring Kotlin into the data science domain, learn the area(s) that interest you.

Apache Hadoop/Spark
Mathematical Models
Statistical Models

Graphing/visualizations
Machine Learning
Linear programming

Data mining
Data wrangling
Optimization

Create some data-driven Kotlin projects and share them!

OSS Libraries

Blog articles

Apps



Getting Involved

Help Bring Kotlin to Data Science

Never stop researching, learning, and advocating

- Although it is incredibly difficult to achieve, never stop striving for that “unicorn” status.
- Keep struggling to learn math, statistics, machine learning, etc... and find ways to make what you learn useful.
- Introduce data-driven features into your apps, and share how you did it.
- If you work on a data science team, propose using Kotlin as a possible solution especially when production needs arise.



Practical Advice

Using Kotlin for Data Science

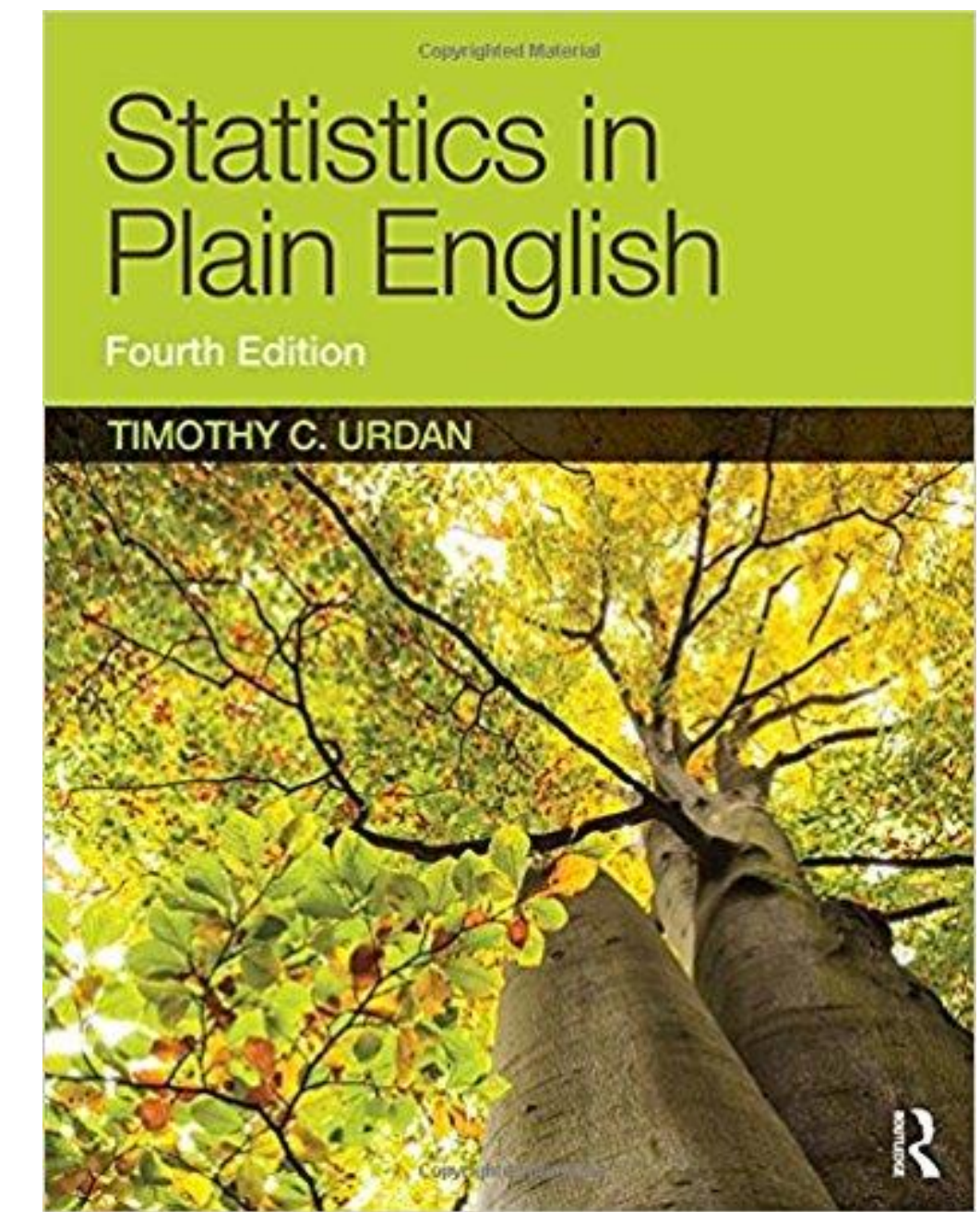
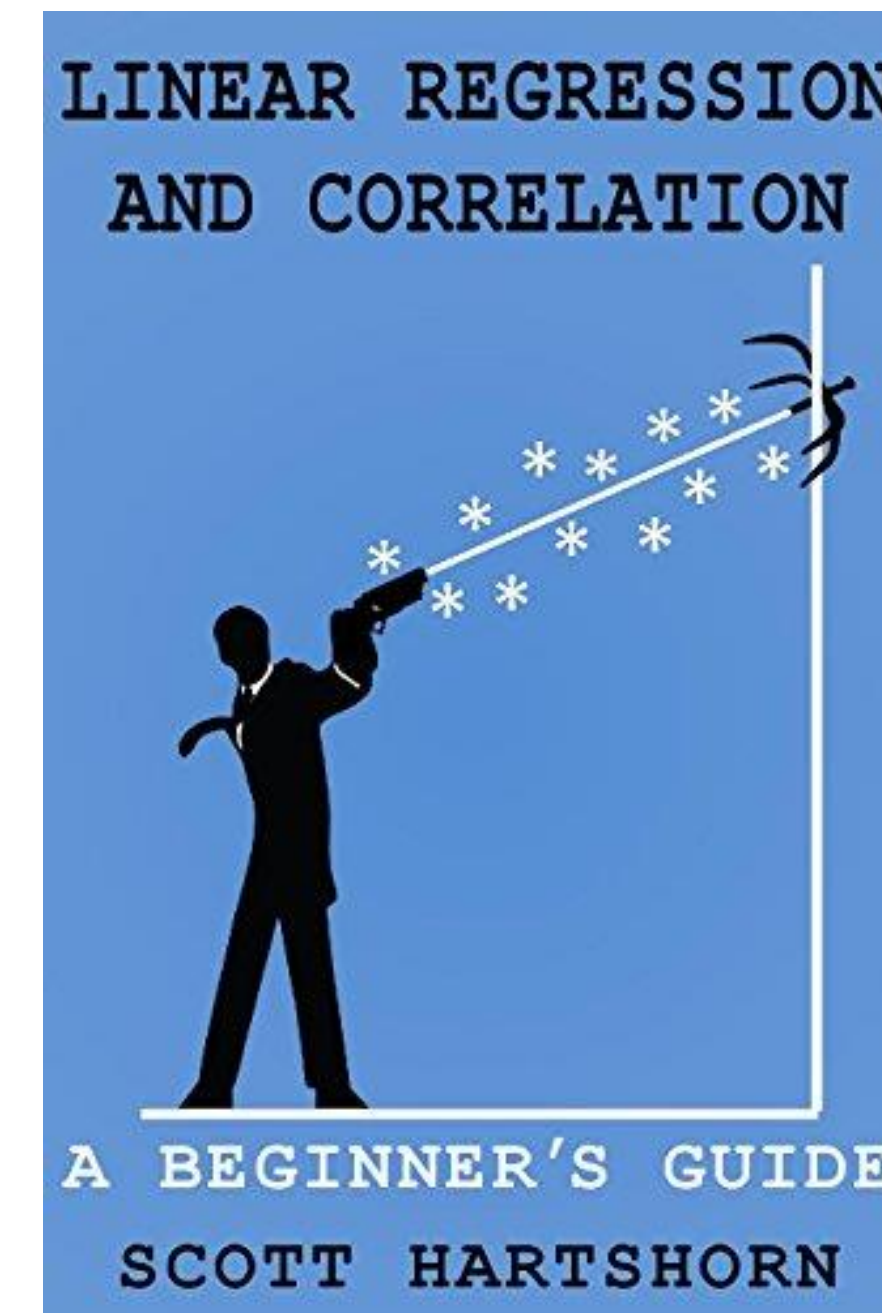
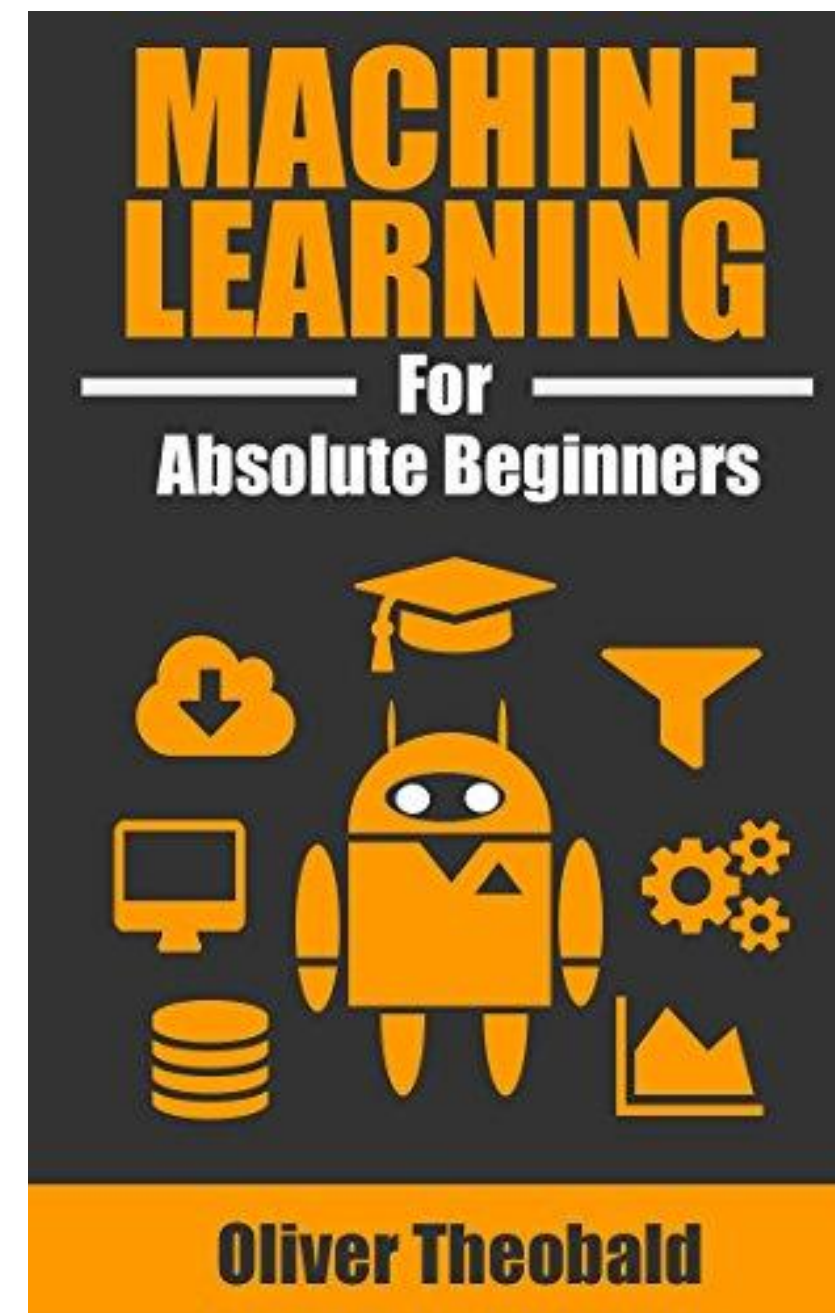
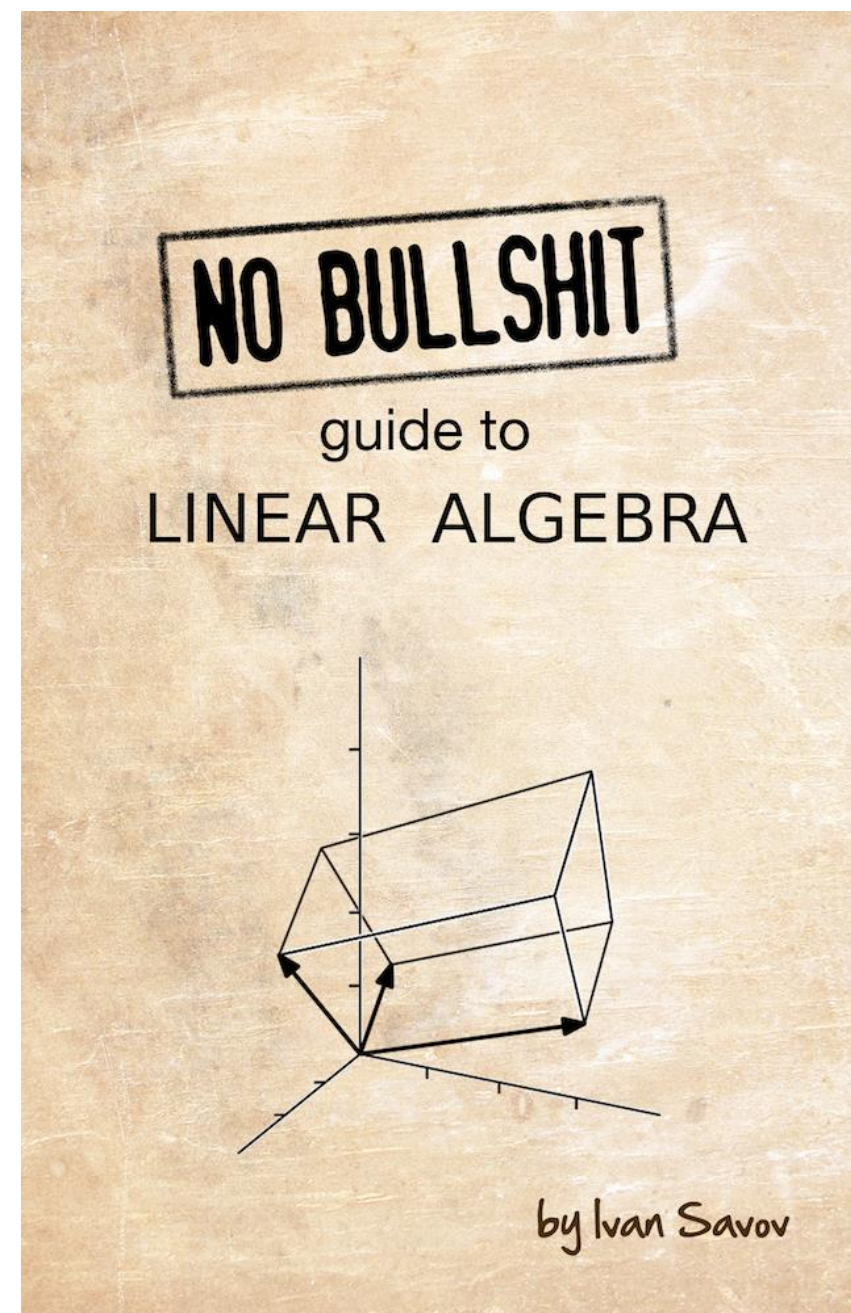
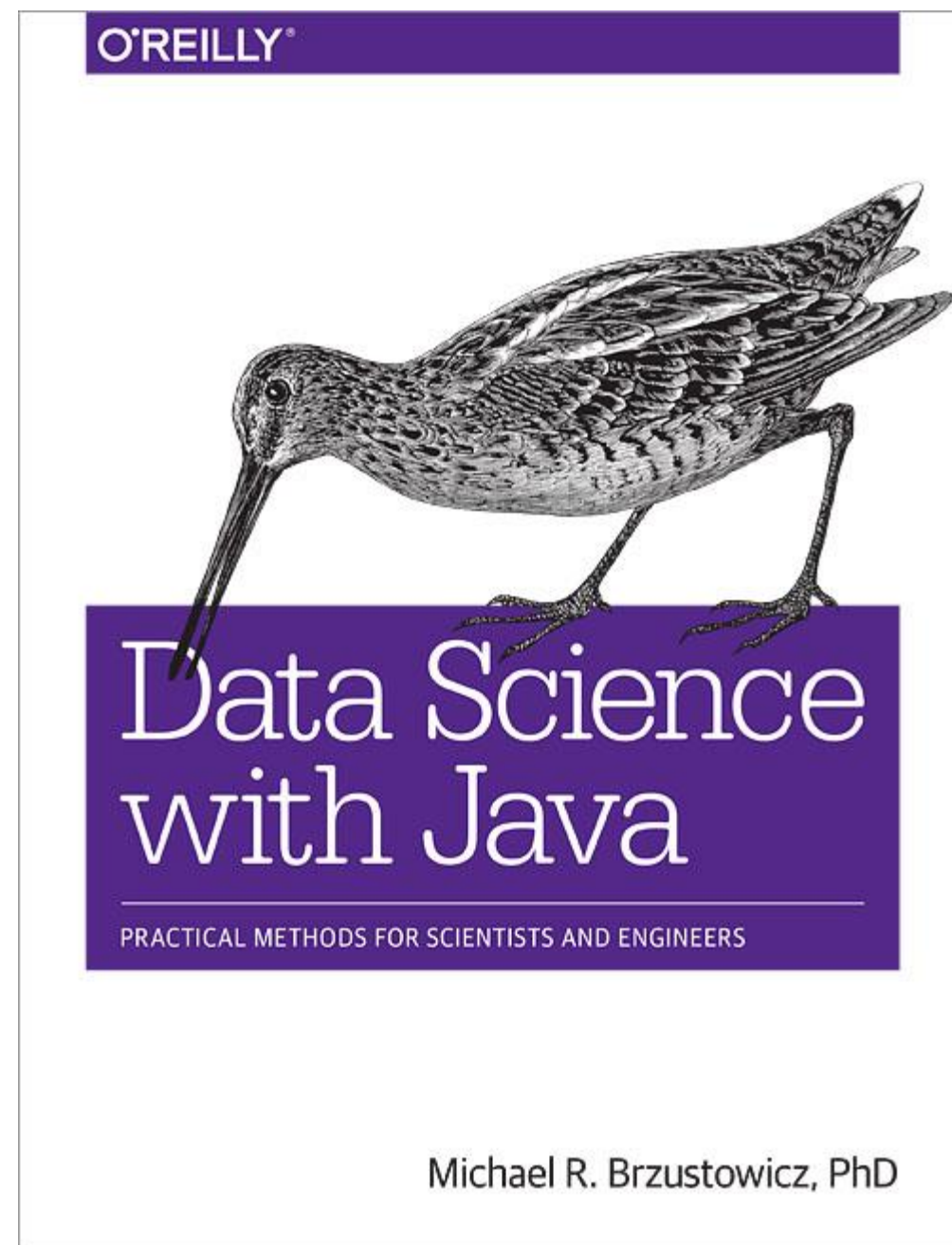
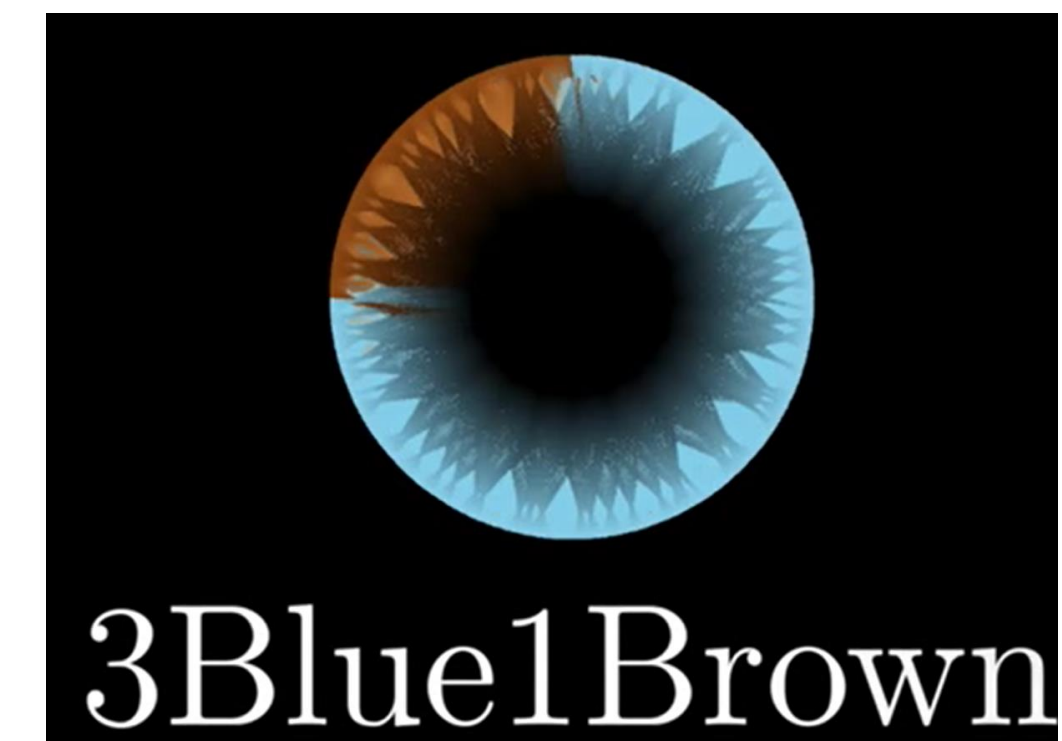
Utilize object-oriented programming, functional programming, and DSL's when doing modeling.

- Rather than working exclusively with matrices, data frames, and piles of numbers, use classes and functional pipelines to keep things organized and refactorable.
- Avoid getting procedural and have a well-planned domain of classes, functions, and DSL's to feed numbers and functions into your modeling library.



Resources To Learn Data Science

Excellent YouTube Channel!



Never rely on one resource!

