R Optimization for Finance

With ROI Package (Part 2: Nonlinear Optimization)

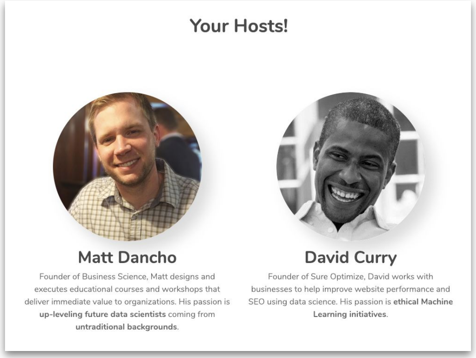
Difficulty: **Intermediate**

Matt Dancho & David Curry

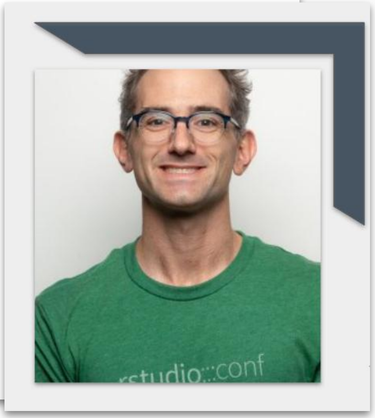
Business Science Learning Lab

Learning Lab Structure

- **Presentation**
  (20 min)
- **Demo's**
  (20 min)
- **Pro-Tips**
  (20 mins)



Reproducible Finance with R



Success Story

Luis Francisco Gomez Lopez

- Economist & Student
- Started courses 2 months ago
- Was using Excel
- Completely switched to R
- Latest report & homework assignment
- **TOTAL TRANSFORMATION!**

"Without the BSU courses, I couldn't do what I'm doing."

Gross Domestic Product (GDP)

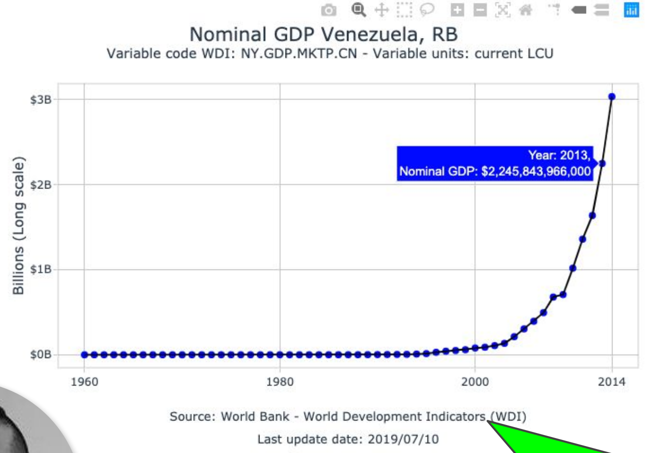
Real and Nominal GDP

- Enter into the World Bank's Human Development Indicators database using the link:

<https://databank.worldbank.org/source/world-development-indicators>

- In *Country* select *Venezuela, RB*
- In *Series* select *GDP (current LCU)*
- In *Time* select *VIEW RECENT YEARS: 50*
- Apply changes and download the data using *Download options* > *Excel* at the upper right corner of the platform

1. Make a plot of the **Nominal GDP** of *Venezuela, BR* where the x-axis corresponds to the years and y-axis corrsponds to the value of the **Nominal GDP**. **(4 points)**



#BusinessScienceSuccess

Agenda

2 Parts

Part 1 (Lab 15)

Product Mix Problem

Linear Programming

If you missed this, join Learning Labs PRO

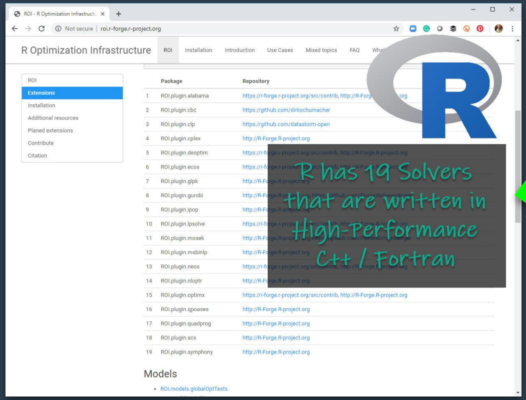
	A	B	C	D	E	F	G	H	I	J
1	**Assembling and Testing Macbooks**									
2										
3	Cost per labor hour assembling	$14								
4	Cost per labor hour testing, line 1	$22								
5	Cost per labor hour testing, line 2	$19								
6										
7	Inputs for assembling and testing a Macbook									
8		Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	
9	Labor hours for assembly	4	5	5	5	5.5	5.5	5.5	6	
10	Labor hours for testing, line 1	1.5	2	2	2	2.5	2.5	2.5	3	
11	Labor hours for testing, line 2	2	2.5	2.5	2.5	3	3	3.5	3.5	
12	Cost of component parts	$900	$1,350	$1,350	$1,350	$1,500	$1,500	$1,500	$1,800	
13	Selling price	$2,100	$2,700	$2,760	$2,820	$3,000	$3,150	$3,180	$3,600	
14	Unit margin, tested on line 1	$1,111	$1,236	$1,296	$1,356	$1,368	$1,518	$1,548	$1,650	
15	Unit margin, tested on line 2	$1,106	$1,233	$1,293	$1,353	$1,366	$1,516	$1,537	$1,650	
16										
17	Assembling, testing plan (# of Macbooks)									
18		Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	
19	Number tested on line 1	1500	0	0	0	0	100	1000	0	
20	Number tested on line 2	0	0	0	0	0	900	0	500	
21	Total computers produced	1500	0	0	0	0	1000	1000	500	
22		<=	<=	<=	<=	<=	<=	<=	<=	
23	Maximum sales	1500	1250	1250	1250	1000	1000	1000	800	
24										
25	Constraints (hours per month)	Hours used		Hours available						
26	Labor availability for assembling	20000	<=	20000						
27	Labor availability for testing, line 1	5000	<=	5000						
28	Labor availability for testing, line 2	4450	<=	6000						
29										
30	Net profit ($ per month)	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Totals
31	Tested on line 1	$1,666,500	$0	$0	$0	$0	$151,800	$1,548,000	$0	$3,366,300
32	Tested on line 2	$0	$0	$0	$0	$0	$1,364,400	$0	$824,750	$2,189,150
33										$5,555,450

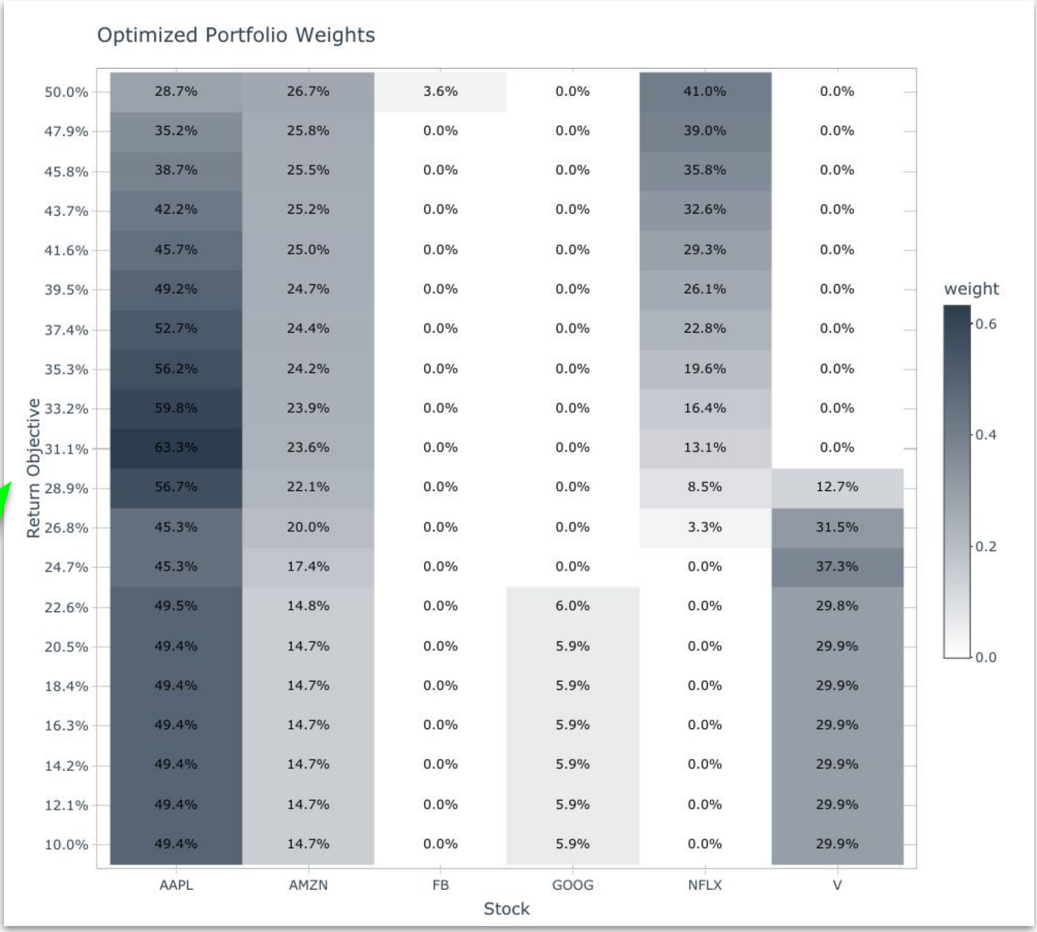
```r
model_2 <- MIPModel() %>%

    add_variable(macbooks_tested_line_1[i], i = 1:n_models, type = "integer", lb = 0) %>%
    add_variable(macbooks_tested_line_2[i], i = 1:n_models, type = "integer", lb = 0) %>%

    add_constraint(macbooks_tested_line_1[i] + macbooks_tested_line_2[i] <= max_sales[i], i = 1:n_models) %>%

    add_constraint(sum_expr((macbooks_tested_line_1[i] + macbooks_tested_line_2[i]) * labor_hours_for_assembly[i], i = 1:n_models) <= max_labor[1]) %>%
    add_constraint(sum_expr((macbooks_tested_line_1[i]) * labor_hours_for_testing_line_1[i], i = 1:n_models) <= max_labor[2]) %>%
    add_constraint(sum_expr((macbooks_tested_line_2[i]) * labor_hours_for_testing_line_2[i], i = 1:n_models) <= max_labor[3]) %>%

    set_objective(
        sum_expr(macbooks_tested_line_1[i] * unit_margin_line_1[i] + macbooks_tested_line_2[i] * unit_margin_line_2[i], i = 1:n_models),
        sense = "max") %>%

    solve_model(with_ROI(solver = "glpk"))
```

Part 2 (Today)

Stock Portfolio Optimization

Nonlinear Programming





Learning Labs PRO

Every 2-Weeks

1-Hour Course

Recordings + Code + Slack

$19/month

university.business-science.io

Lab 15
R's Optimization Toolchain, Part 1

Lab 14
Customer Churn Survival Analysis

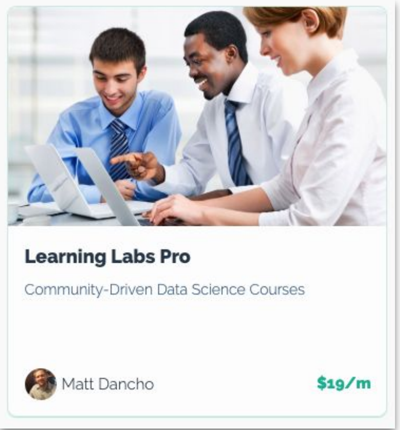
Lab 13
Wrangling 4.6M Rows of Financial Data w/ data.table

Lab 12
How I built anomalize

Lab 11
Market Basket Analysis w/ recommenderLab

Lab 10
Building API's with plumber & postman

Lab 9
Finance in R with tidyquant

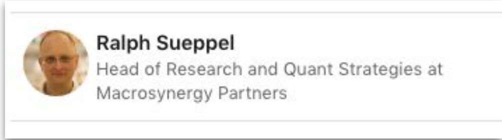


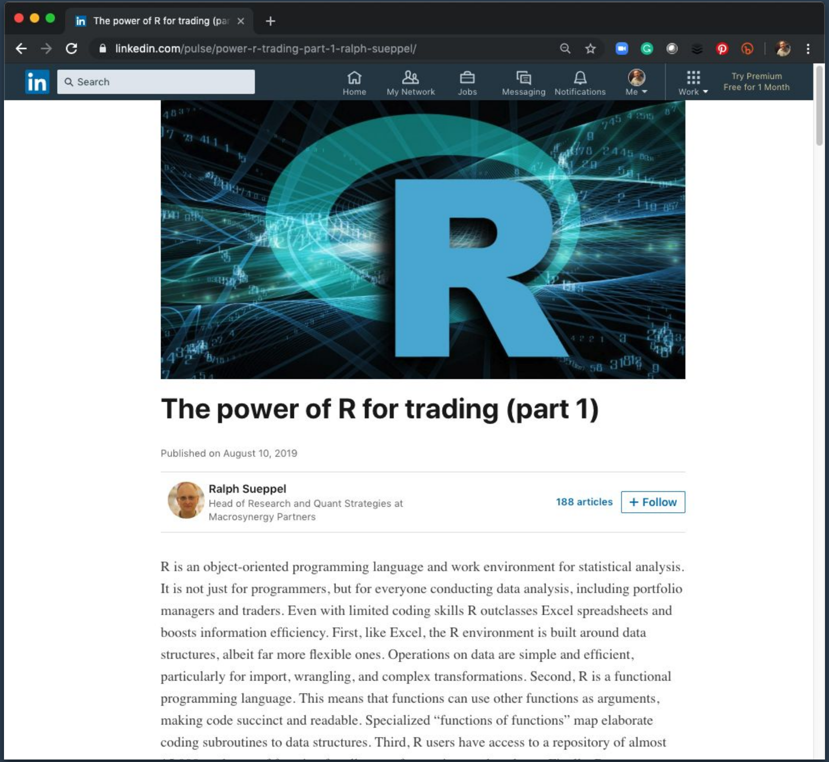
R in Industry

The power of R for trading

The power of R for Trading

R is the best language in the world for doing **rapid financial analysis**





The power of R for trading (part 1)

Published on August 10, 2019

Ralph Sueppel
Head of Research and Quant Strategies at
Macrosynergy Partners

188 articles + Follow

R is an object-oriented programming language and work environment for statistical analysis. It is not just for programmers, but for everyone conducting data analysis, including portfolio managers and traders. Even with limited coding skills R outclasses Excel spreadsheets and boosts information efficiency. First, like Excel, the R environment is built around data structures, albeit far more flexible ones. Operations on data are simple and efficient, particularly for import, wrangling, and complex transformations. Second, R is a functional programming language. This means that functions can use other functions as arguments, making code succinct and readable. Specialized "functions of functions" map elaborate coding subroutines to data structures. Third, R users have access to a repository of almost

https://www.linkedin.com/pulse/power-r-trading-part-1-ralph-sueppel/

Financial Case Study

Optimize Stock Portfolio

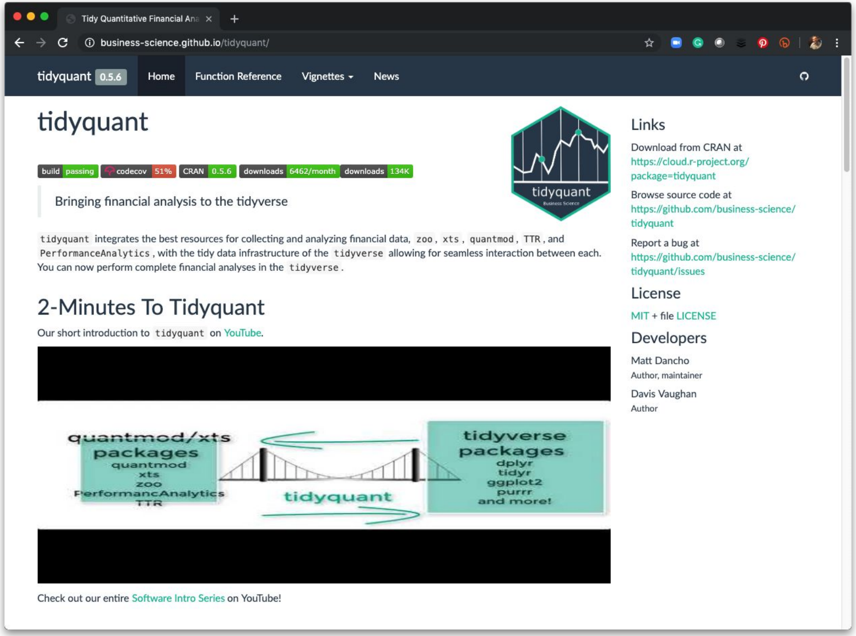
Building a Financial Portfolio

Historical Stock Prices

We analyze the historical stock prices for **5 Stocks**:

AAPL, AMZN, FB, GOOG, & NFLX

How do we determine an **optimal portfolio mix**?



https://business-science.github.io/tidyquant/

Modern Portfolio Theory

Minimize Variance

Harry Markowitz - Nobel Prize

What is the **lowest portfolio variance** that I can achieve a given **return objective**?

$$\min_{x \in \mathbb{R}^N} x^\top Q x$$

$$Q = Cov(r)$$

x

Stock input data					
	AAPL	AMZN	FB	GOOG	NFLX
Mean return	0.18	0.39	0.37	0.19	0.81

Q

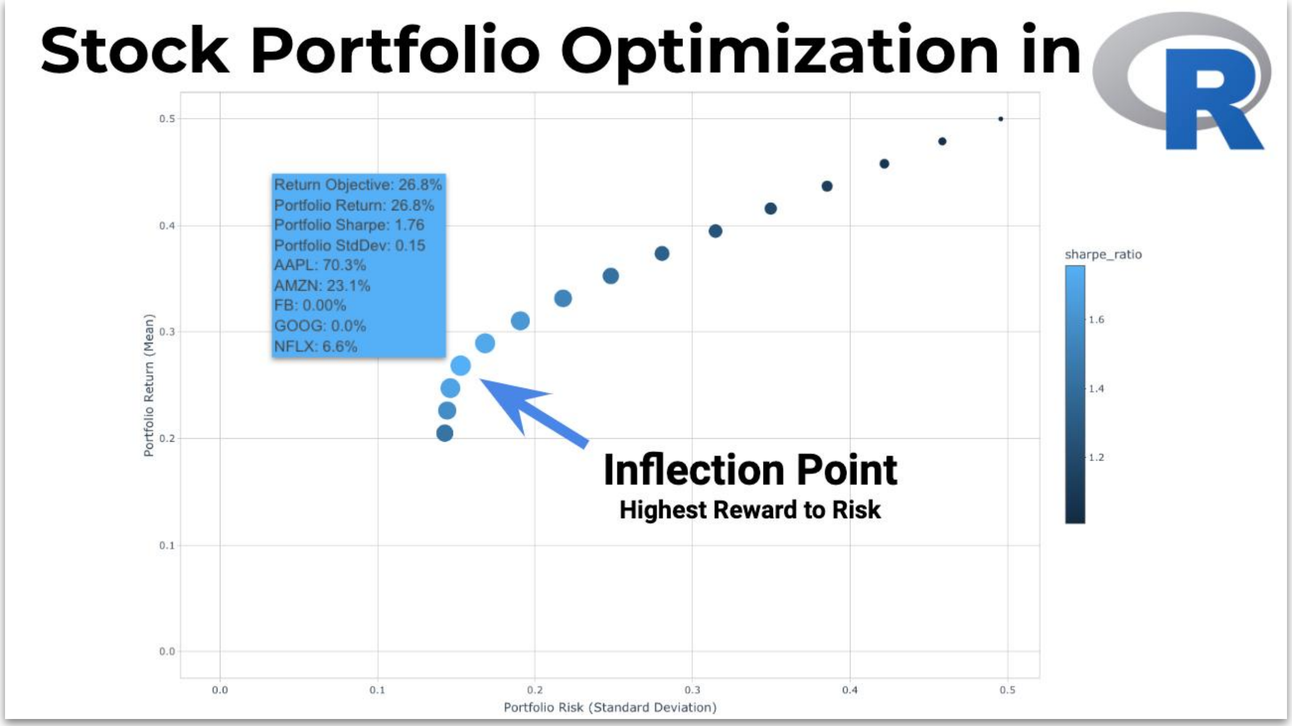
Covariances	AAPL	AMZN	FB	GOOG	NFLX
AAPL	0.0451	-0.0407	0.0270	-0.0106	-0.0968
AMZN	-0.0407	0.1926	0.0320	0.0880	0.2619
FB	0.0270	0.0320	0.1411	0.0620	0.2632
GOOG	-0.0106	0.0880	0.0620	0.0626	0.2291
NFLX	-0.0968	0.2619	0.2632	0.2291	1.1377

Efficient Frontier - What happens when we minimize risk

Efficient Frontier

What happens when we adjust the **Portfolio Return Objective**?

Quantitative Decision Making + Optimization



Optimization (Recap)

80/20 Concepts

Terminology

1

Objectives

The **goal** of the model

Always Maximize or Minimize.

2

Constraints

Decision Rules for the model

Restricts the decision variables to boundaries

3

Decision Variables

Your problem's **Frame of Reference**

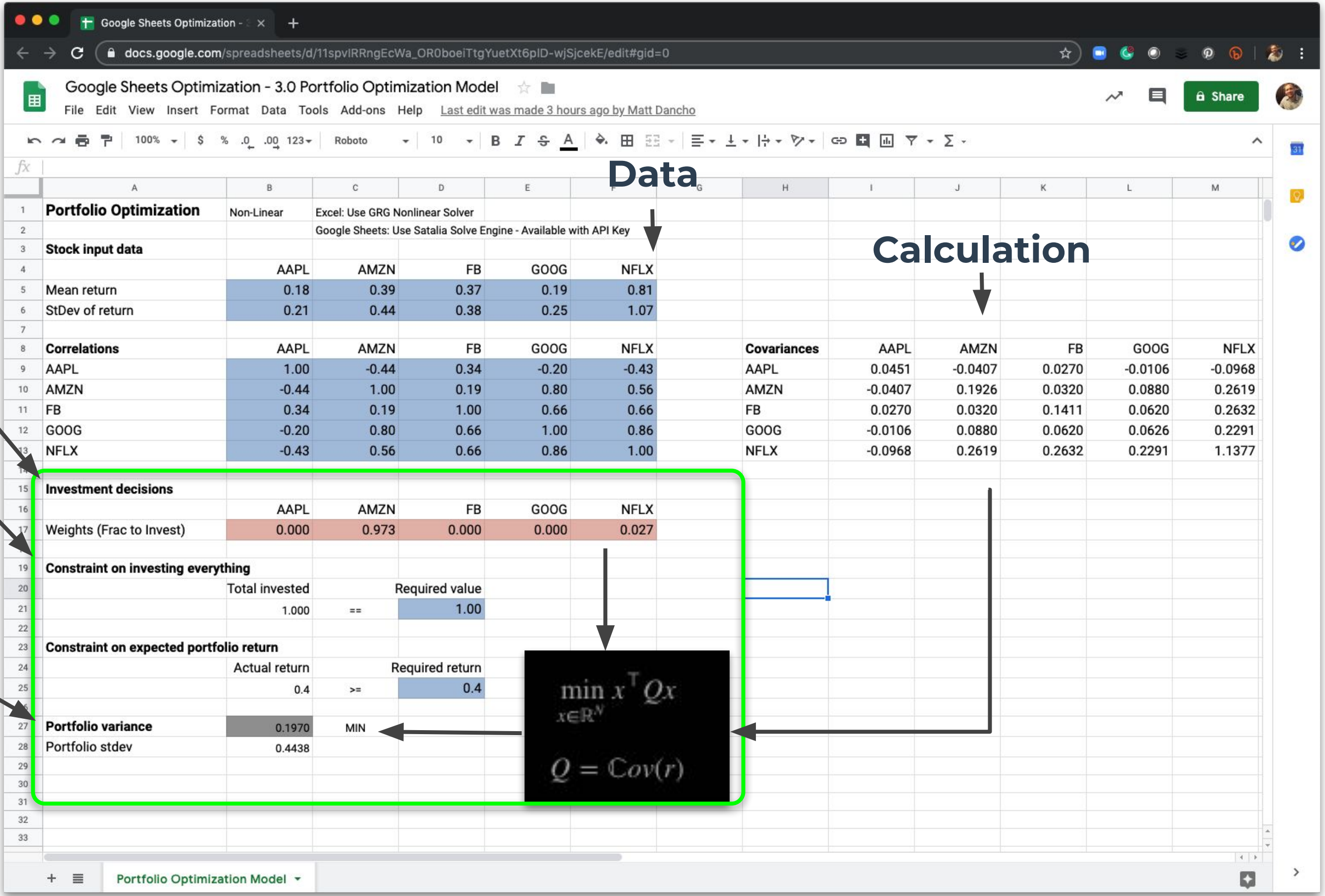
The parameters that your **optimization model adjusts**

Excel

Decision Variables (adjusted values)

Constraints

Goal



Types of Optimization Models

Linear

- Fastest solutions
- Easy to conceptualize
- Analysis limited to aggregations, constant multiplications, etc
- Many problems don’t fit this mold (e.g. take correlation of something)

Quadratic

- Fast solutions
- Difficult to Conceptualize
- Requires formulation as a quadratic function

Nonlinear

- Easy to conceptualize
- Super Flexible
- Cannot use linear solvers
- Slower solutions
- Can get suboptimal results (local vs global maxima)

Types of Optimization Models

Linear

- Fastest solutions
- Easy to conceptualize
- Analysis limited to aggregations, constant multiplications, etc
- Many problems don't fit this mold (e.g. take correlation of something)

Part 1

~~Quadratic~~

- ~~Fast solutions~~
- ~~Difficult to Conceptualize~~
- ~~Requires formulation as a quadratic function~~

Nonlinear

- Easy to conceptualize
- Super Flexible
- Cannot use linear solvers
- Slower solutions
- Can get suboptimal results (local vs global maxima)

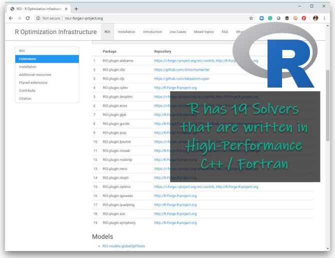
Part 2

Process & Tools

What we need to know

Financial Optimization Modeling

Step-By-Step



Start — **1** — **2** — **3** — **Finish**

1. tidyquant & dplyr

Retrieve & Manipulate Data

2. ROI

Nonlinear Optimization

3. purrr & ggplot2

Iteration & Visualization

Financial Optimization Modeling

Step-By-Step

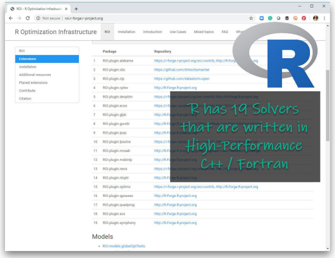
Start

1

tidyquant & dplyr

Retrieve & Manipulate Data

101 & Lab 13



2

ROI

Nonlinear Optimization

NEW

3

purrr & ggplot2

Iteration & Visualization

101 & 201

Finish

ROI Package

80/20 Concepts & Important Operations

ROI

Conversion process is a bit trickier than what we learned in **Part 1**

Need to think in terms of **Matrix Calculations**

Nonlinear programming has 4 Super Powers

Using ROI

Decision Variables (adjusted variables)

Asset Weights for each stock

[w_aapl, w_amzn, w_fb, w_goog, w_nflx]

```r
model_nlp <- OP(
    objective   = F_objective(F = calc_portfolio_variance, n = n_assets, names = assets),
    constraints = rbind(
        F_constraint(F = calc_portfolio_return, dir = ">=", rhs = 0.40),
        L_constraint(diag(n_assets), rep(">=", n_assets), rep(0, n_assets)),
        L_constraint(diag(n_assets), rep("<=", n_assets), rep(1, n_assets)),
        L_constraint(rep(1, n_assets), "==", 1)
    ),
    maximum = FALSE
)

tic()
sol <- ROI_solve(model_nlp, solver = "alabama", start = rep(1/n_assets, n_assets))
toc()

```

- **F_objective**
    - Minimize portfolio variance calculation
- **F_constraint** - Functional Constraint
    - Inputs asset weights, outputs port
    - Return objective set to 40% per year
- **L_constraint** - Linear Constraint
    - Bounding the weights

Nonlinear Super Power #1

Functional Objective

```r
str(F_objective)
```

```
## function (F, n, G = NULL, H = NULL, names = NULL)
```

$$x_1^2 + x_2^2$$

```r
fo <- F_objective(F = function(x) sum(x^2), n = 2,
                  G = function(x) 2*x, names = c("x_1", "x_2"))
```

http://roi.r-forge.r-project.org/introduction.html

Nonlinear Super Power #2

Functional Constraints

```
str(F_constraint)
```

```
## function (F, dir, rhs, J = NULL, names = NULL)
```

$$\begin{aligned} x^2 &\leq 2 \\ y^2 &\leq 4 \end{aligned}$$

```
fc1 <- F_constraint(F = function(x) x^2, dir = c("<=", "<="), rhs = c(2, 4),
                    J = function(x) diag(x = 2, nrow = 2) * x,
                    names = c("x", "y"))
```

http://roi.r-forge.r-project.org/introduction.html

Nonlinear Super Power #3

Linear Constraints

```
str(L_constraint)
```

```
## function (L, dir, rhs, names = NULL)
```

$$
\begin{array}{ccccccc}
3x & + & 4y & + & 1z & \leq & 90 \\
1x & + & 0y & + & 2z & \geq & 5 \\
1x & + & 1y & + & 0z & = & 2
\end{array}
$$

```
lc <- L_constraint(L = rbind(c(3, 4, 1), c(1, 0, 2), c(1, 1, 0)),
                   dir = c("<=", ">=", "=="), rhs = c(90, 5, 2),
                   names = c("x", "y", "z"))
```

http://roi.r-forge.r-project.org/introduction.html

Nonlinear Super Power #4

Put it all together

Can combine linear (matrix) & nonlinear programming (functions)

```r
model_nlp <- OP(
    objective   = F_objective(F = calc_portfolio_variance, n = n_assets, names = assets),
    constraints = rbind(
        F_constraint(F = calc_portfolio_return, dir = ">=", rhs = 0.40),
        L_constraint(diag(n_assets), rep(">=", n_assets), rep(0, n_assets)),
        L_constraint(diag(n_assets), rep("<=", n_assets), rep(1, n_assets)),
        L_constraint(rep(1, n_assets), "==", 1)
    ),
    maximum = FALSE
)

tic()
sol <- ROI_solve(model_nlp, solver = "alabama", start = rep(1/n_assets, n_assets))
toc()
```

Demo

Optimizing a Stock Portfolio

AAPL, AMZN, FB, GOOG, NFLX

Optimization

Secret Tactics & Pro Tips

People get **frustrated** with

Optimization

Because they **fail to start simple**

Pro Tip

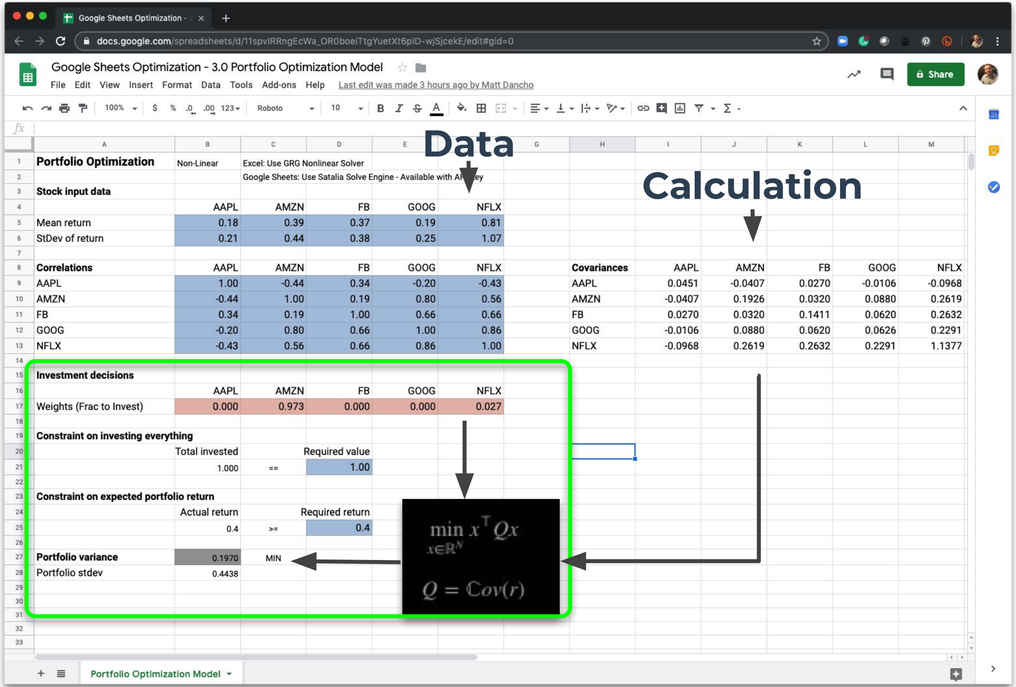
Design in Excel, then Convert to R

When starting out...

Pros:
Sometimes Excel is **easier** to conceptualize

Cons:

- Excel <u>cannot scale</u> (try to do **iterative analysis**)
- Excel <u>cannot scale</u> (try to add **1 more stock**)
- Excel has difficulties with **nonlinear** problems



Pro Tip

Design in Excel, then Convert to R

When starting out...

Cons:
Sometimes R is **more difficult** to conceptualize

Pros:

- R can scale (try to do **iterative analysis**)
- R can scale (try to add **1 more stock**)
- R is really good with **nonlinear** problems

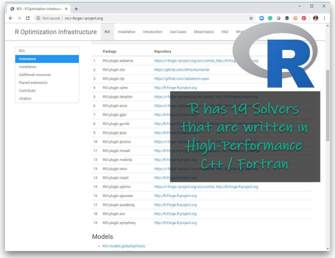
```r
model_nlp <- OP(
    objective   = F_objective(F = calc_portfolio_variance, n = n_assets, names = assets),
    constraints = rbind(
        F_constraint(F = calc_portfolio_return, dir = ">=", rhs = 0.40),
        L_constraint(diag(n_assets), rep(">=", n_assets), rep(0, n_assets)),
        L_constraint(diag(n_assets), rep("<=", n_assets), rep(1, n_assets)),
        L_constraint(rep(1, n_assets), "==", 1)
    ),
    maximum = FALSE
)

tic()
sol <- ROI_solve(model_nlp, solver = "alabama", start = rep(1/n_assets, n_assets))
toc()
```

Data Science + Decision Science

Learning Plan

Financial Optimization Modeling

Step-By-Step



Start

1

2

3

Finish

tidyquant & dplyr

Retrieve & Manipulate Data

101 & Lab 13

ROI

Nonlinear Optimization

Lab 16

purrr & ggplot2

Iteration & Visualization

101 & 201

dplyr + tidyquant

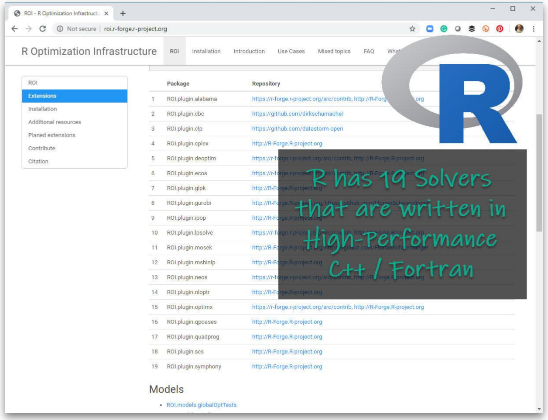
```r
# 2.0 DATA ----
assets <- c("FB", "AMZN", "AAPL", "GOOG", "NFLX") %>% sort()

stock_prices_tbl <- tq_get(assets, from = "2013-01-01", to = "2019-07-01")

stock_returns_tbl <- stock_prices_tbl %>%
    select(symbol, date, adjusted) %>%
    group_by(symbol) %>%
    tq_transmute(adjusted, mutate_fun = periodReturn, period = "yearly", col_rename = "returns")

returns_matrix_tbl <- stock_returns_tbl %>%
    spread(symbol, returns) %>%
    select(assets)

returns_matrix_tbl
```

Lab 13

```r
# 4.3.1 Heat Map ----
plot_heatmap <- function(data) {

    data_transformed_tbl <- data %>%
        mutate(sharpe_ratio = portfolio_return / portfolio_stdev) %>%
        mutate(portfolio_id = row_number()) %>%
        gather(key = stock, value = weight,
               -sharpe_ratio, -portfolio_return, -portfolio_stdev,
               -portfolio_id, -return_constraint,
               factor_key = TRUE) %>%
        mutate(return_objective = scales::percent(return_constraint)) %>%
        mutate(label_text = str_glue("Return Objective: {scales::percent(return_constraint)}
                                     Portfolio Return: {scales::percent(portfolio_return)}
                                     Portfolio Sharpe: {round(sharpe_ratio, 2)}
                                     Portfolio StdDev: {round(portfolio_stdev, 2)}"))

```

101

ROI




```r
model_nlp <- OP(
    objective   = F_objective(F = calc_portfolio_variance, n = n_assets, names = assets),
    constraints = rbind(
        F_constraint(F = calc_portfolio_return, dir = ">=", rhs = 0.40),
        L_constraint(diag(n_assets), rep(">=", n_assets), rep(0, n_assets)),
        L_constraint(diag(n_assets), rep("<=", n_assets), rep(1, n_assets)),
        L_constraint(rep(1, n_assets), "==", 1)
    ),
    maximum = FALSE
)

tic()
sol <- ROI_solve(model_nlp, solver = "alabama", start = rep(1/n_assets, n_assets))
toc()
```

Lab 16

ggplot2 & purrr

```r
    g <- data_transformed_tbl %>%
        ggplot(aes(stock, y = return_objective, fill = weight)) +
        geom_tile() +
        geom_point(aes(text = label_text), size = 0.1, alpha = 0) +
        scale_fill_gradient(low = "#FFFFFF", high = "#2c3e50") +
        geom_text(aes(label = scales::percent(weight)), size = 3) +
        theme_tq() +
        labs(title = "Optimized Portfolio Weights", x = "Stock", y = "Return Objective")

    ggplotly(g, tooltip = "text")

}

portfolio_sim_results_tbl %>% plot_heatmap()
```

101 + 201

```r
# 4.2 Map (Simulation) ----
tic()
portfolio_sim_results_tbl <- seq(0.10, 0.50, length.out = 20) %>%
    map_dfr(optimize_portfolio)
toc()

portfolio_sim_results_tbl
```

101 + 201

Business Science University

Our program that will <u>TRANSFORM YOU</u> in weeks, not years.

The program that will deliver YOUR Transformation

R-TRACK

LEARNING LABS PRO

Do Business Projects
Climb the Hill

Build Production-Ready
Web Apps

Complete 1-Hour Courses
Continuous Education

Start — 1 — 2 — 3 — Finish

Analysis Courses
101 & 201

App Development Courses
102

Learning Labs PRO

Everything is **Taken Care of** For You in Our Platform

R-TRACK

3-Course R-Track System

Business Analysis with R
(DS4B 101-R)

Data Science For Business with R
(DS4B 201-R)

R Shiny Web Apps For Business
(DS4B 102-R)

Project-Based Courses with Business Application

Data Science Foundations
7 Weeks

Machine Learning & Business Consulting
10 Weeks

Web Application Development
4 Weeks

DS4B 101-R: Business Analysis With R

Your Data Science Journey Starts Now! Learn the fundamentals of data science for business with the tidyverse.

Matt Dancho

DS4B 201-R: Data Science For Business With R

Solve a real-world churn problem with H2O AutoML (automated machine learning) & LIME black-box model explanations using R

Matt Dancho

DS4B 102-R: Shiny Web Applications For Business (Level 1)

Build a predictive web application using Shiny, Flexdashboard, and XGBoost

Matt Dancho

Key Benefits

- Fundamentals - Weeks 1-5 (25 hours of Video Lessons)
    - Data Manipulation (dplyr)
    - Time series (lubridate)
    - Text (stringr)
    - Categorical (forcats)
    - Visualization (ggplot2)
    - Programming & Iteration (purrr)
    - 3 Challenges

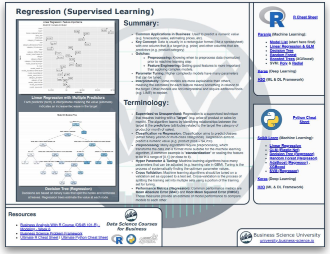
- **Machine Learning - Week 6 (8 hours of Video Lessons)**
    - **Clustering (3 hours)**
    - **Regression (5 hours)**
    - **2 Challenges**

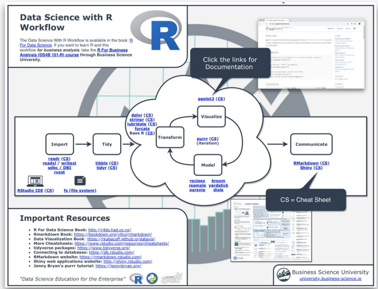
- Learn Business Reporting - Week 7
    - RMarkdown & plotly
    - 2 Project Reports:
        1. Product Pricing Algo
        2. Customer Segmentation

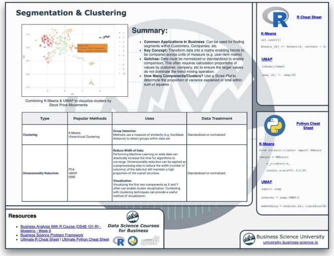
Business Analysis with R (DS4B 101-R)

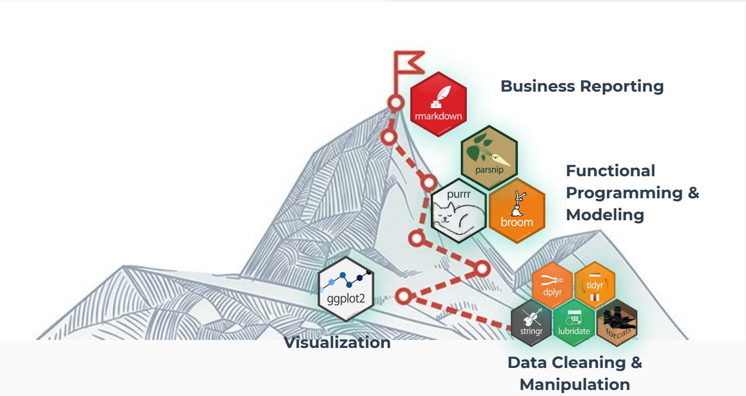
Data Science Foundations

7 Weeks









Key Benefits

End-to-End Churn Project

Understanding the Problem & Preparing Data - Weeks 1-4

- Project Setup & Framework
- Business Understanding / Sizing Problem
- Tidy Evaluation - rlang
- EDA - Exploring Data -GGally, skimr
- Data Preparation - recipes
- Correlation Analysis
- 3 Challenges

Machine Learning - Weeks 5, 6, 7

- **H2O AutoML - Modeling Churn**
- **ML Performance**
- **LIME Feature Explanation**

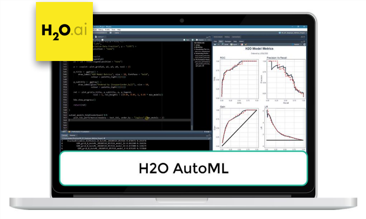
Return-On-Investment - Weeks 7, 8, 9

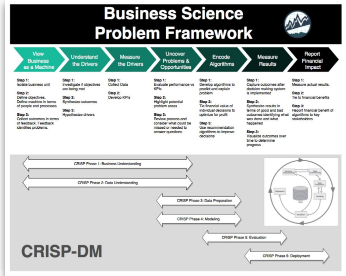
- **Expected Value Framework**
- **Threshold Optimization**
- **Sensitivity Analysis**
- **Recommendation Algorithm**

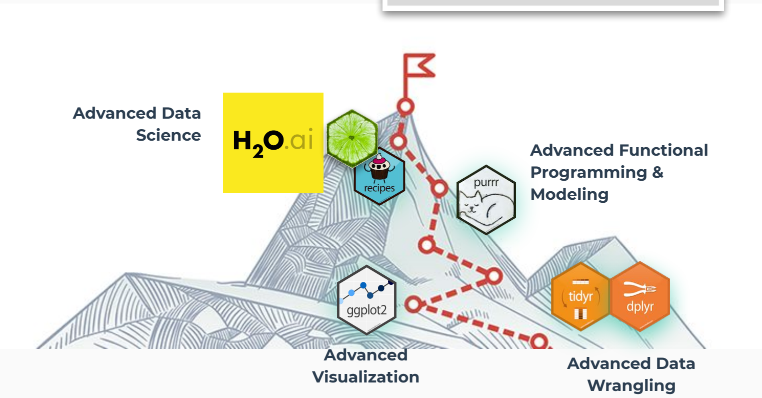
Data Science For Business (DS4B 201-R)

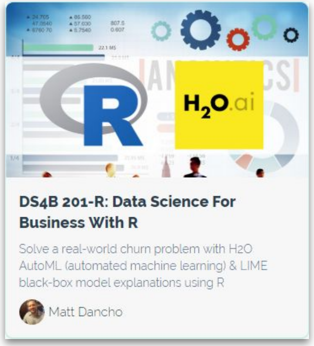
Machine Learning & Business Consulting

10 Weeks









Key Benefits

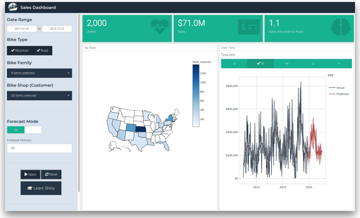
Learn Shiny & Flexdashboard
- Build Applications
- Learn Reactive Programming
- Integrate Machine Learning

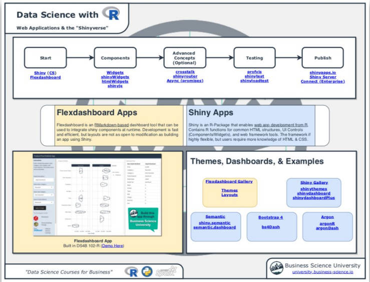
App #1: Predictive Pricing App
- Model Product Portfolio
- XGBoost Pricing Prediction
- Generate new products instantly

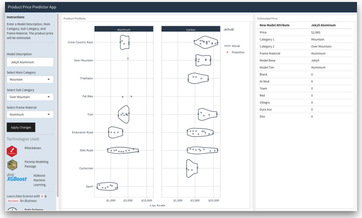
App #2: Sales Dashboard with Demand Forecasting
- Model Demand History
- Segment Forecasts by Product & Customer
- XGBoost Time Series Forecast
- Generate new forecasts instantly

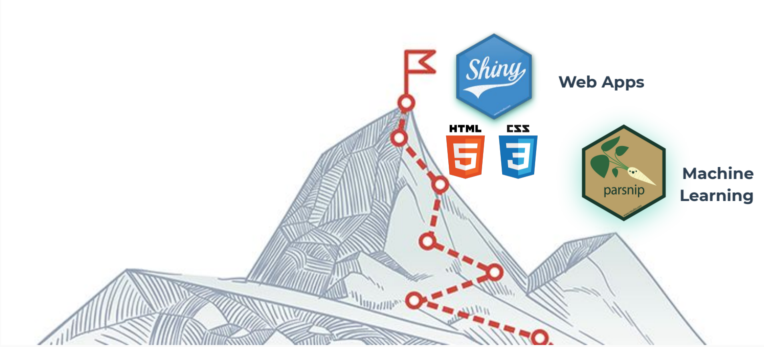
Shiny Apps for Business (DS4B 102-R)

Web Application Development
4 Weeks









Testimonials

*"Your program allowed me to cut down to **50% of the time** to deliver solutions to my clients."*

-Rodrigo Prado, Managing Partner Big Data Analytics & Strategy at Genesis Partners

*"I can already **apply** a lot of the early gains from the course to current working projects."*

-Adam Mitchell, Data Analyst with Eurostar

*"My work became **10X easier**. I can spend quality time asking questions rather than wasting time trying to figure out syntax."*

-Mohana Chittor, Data Scientist with Kabbage, Inc

Achieve **Results** that Matter to the **Business**

15% OFF PROMO Code: learninglabs

R-TRACK BUNDLE

Bundle - DS For Business + Web Apps (Level 1): R-Track - Courses 101, 102,

3 Course Bundle

0%
COMPLETE

DS4B 101-R: Business Analysis With R

Your Data Science Journey Starts Now! Learn the fundamentals of data science for business with the tidyverse.

Matt Dancho

DS4B 201-R: Data Science For Business With R

Solve a real-world churn problem with H2O AutoML (automated machine learning) & LIME black-box model explanations using R

Matt Dancho

DS4B 102-R: Shiny Web Applications For Business (Level 1)

Build a predictive web application using Shiny, Flexdashboard, and XGBoost

Matt Dancho

Get started now!

◉	**Paid Course** 15% COUPON DISCOUNT	~~$1,149~~ $976.65 Enroll
○	**3 Monthly Payments** 15% COUPON DISCOUNT 3X Monthly	~~3 payments of $449/m~~ 3 payments of $381.65/m
○	**6 Low Monthly Payments** 15% COUPON DISCOUNT 6X Payment Plan	~~6 payments of $234/m~~ 6 payments of $198.90/m
○	**12 Low Monthly Payments** 15% COUPON DISCOUNT 12X Plan	~~12 payments of $125/m~~ 12 payments of $106.25/m

Begin Learning Today

university.business-science.io