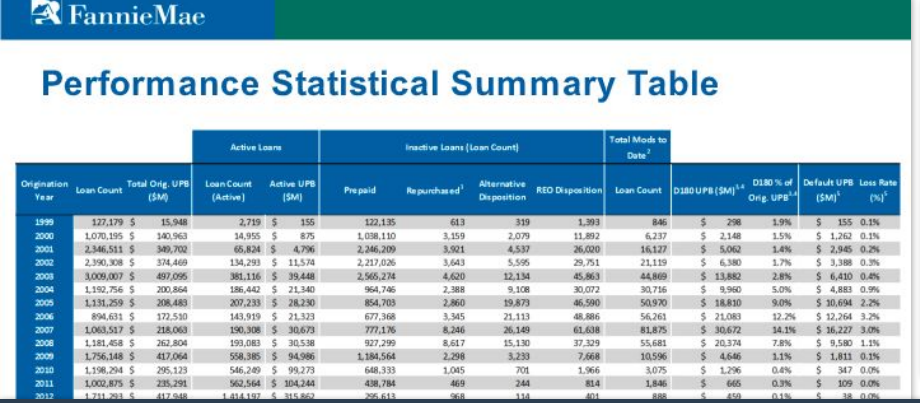
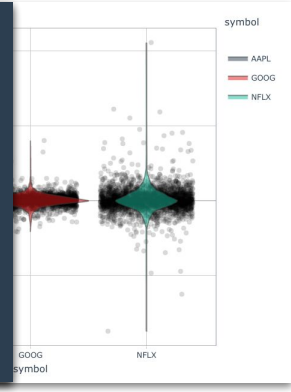


SQL for Time Series

Time Series SQL Wrangling

Difficulty: **Intermediate**



Matt Dancho & David Curry

Business Science Learning Lab

Learning Lab Structure

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

Matt Dancho

Founder of Business Science, Matt designs and executes educational courses and workshops that deliver immediate value to organizations. His passion is **up-leveling future data scientists** coming from **untraditional backgrounds**.

David Curry

Founder of Sure Optimize, David works with businesses to help improve website performance and SEO using data science. His passion is **ethical Machine Learning initiatives**.

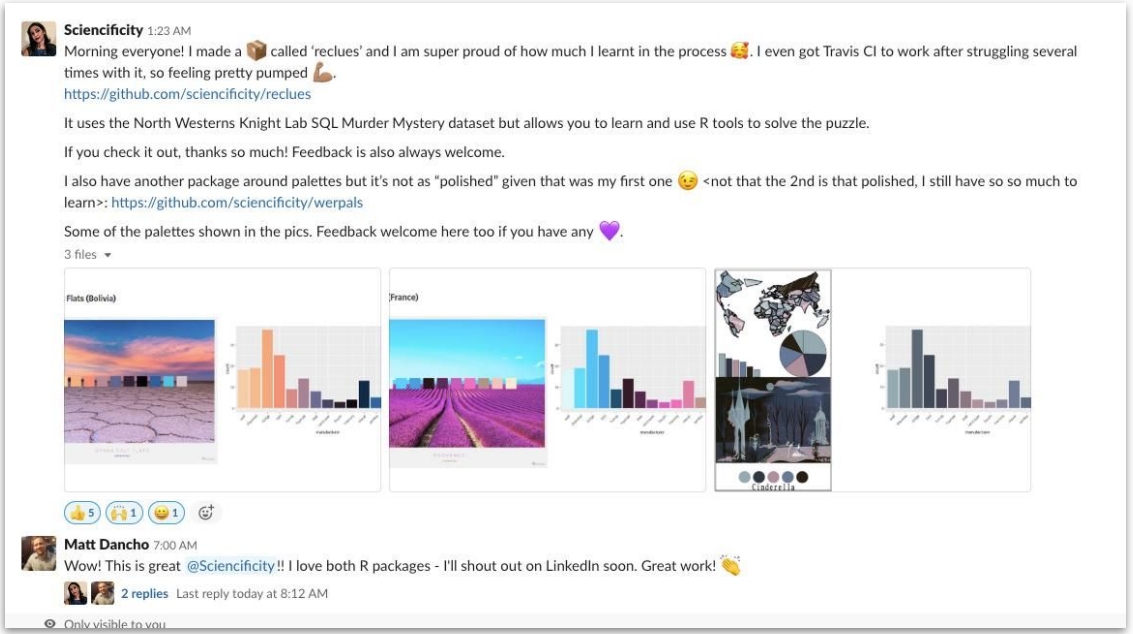
Success Story

Vebashini Naidoo

- General Manager: Data Analytics at rain South Africa
- Student in R-Track
- **Just created her first 2 R Packages**

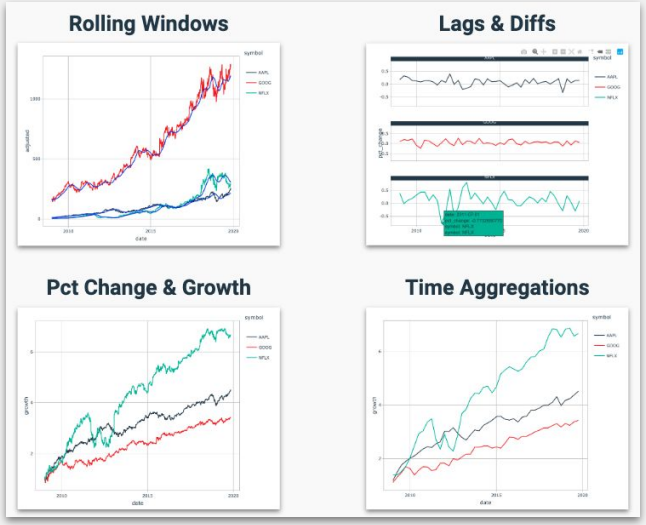
#Business Science Success

"I am super proud of how much I learnt in the process. 🥰"



reclues: https://github.com/sciencificity/reclues
werpals: https://github.com/sciencificity/werpals

Agenda



- **Business Case Study**
  - Mortgage Data
  - Delinquency prevention
- **Time Series Op's**
  - Rolling Windows
  - Lags & Diffs
  - Pct Change & Growth
  - Time Aggregations
- **SQL Concept Recap**
  - Feature Engineering
  - SQL Speed + R Data Science Power
- **30-Min Demo**
  - Stock Analysis
  - Mortgage Analysis
    - 65M Rows
    - 7GB
- **SQL Time Series Pro-Tips**
- **2020 Data Science Learning Guide**

Learning Labs PRO

Every 2-Weeks

1-Hour Course

Recordings + Code + Slack

$19/month

university.business-science.io

Lab 21
SQL for Data Science

Lab 20
Explainable Machine Learning

Lab 19
Using Customer Credit Card History for Networks Analysis

Lab 18
Time Series Anomaly Detection with anomalize

Lab 17
Anomaly Detection with H2O Machine Learning

Lab 16
R's Optimization Toolchain, Part 2 - Nonlinear Programming

Lab 15

Continuous Learning
Jet Fuel for your Brain

Mortgage Loan Delinquency Data

Business Case

Loan Delinquency Costs Billions

Mortgages

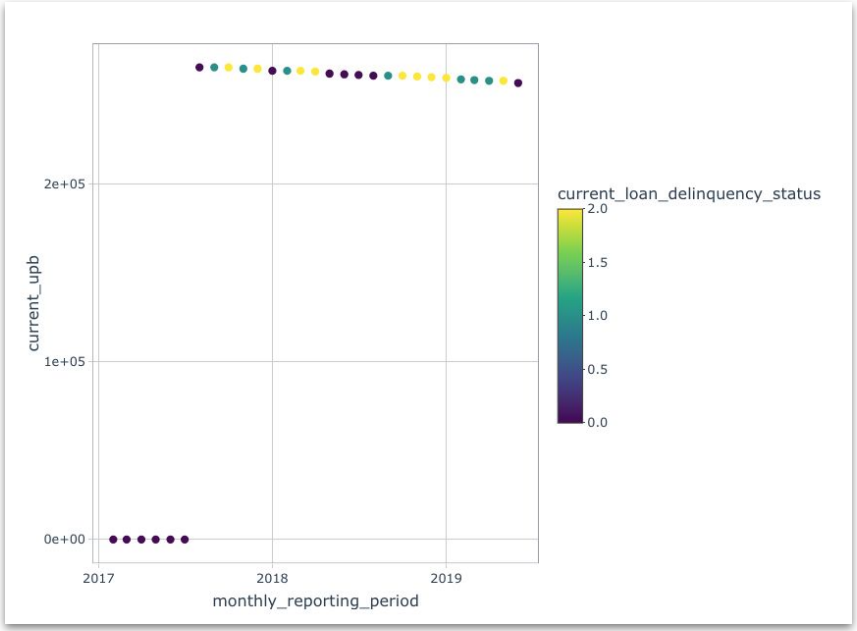
1. Banks lend **Billions**
2. **Delinquency -** When a customer falls behind in payments
3. If we can better **predict repeat patterns** of delinquency, we can **save** billions

Delinquency

How far does the customer fall behind?

Loan ID:
100670358700

Unpaid Balance (UPB)



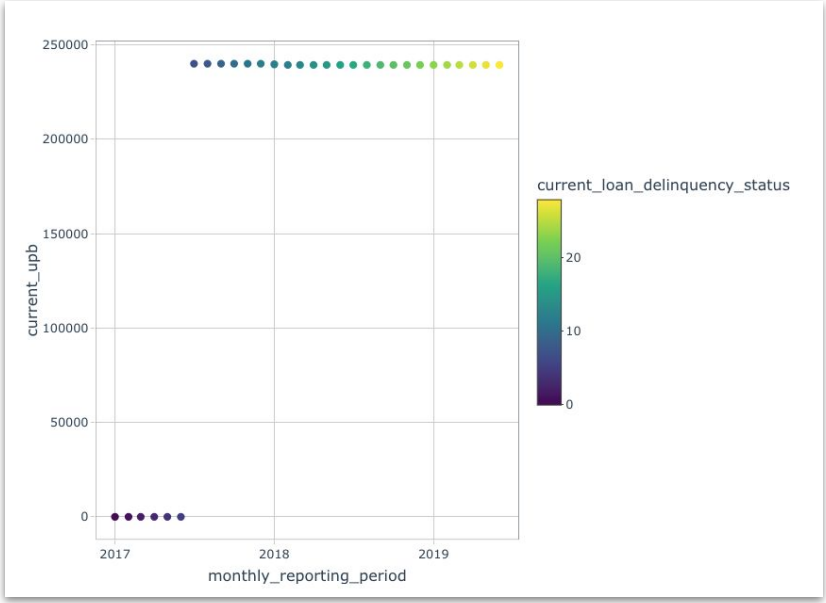
Delinquency Status

Delinquency

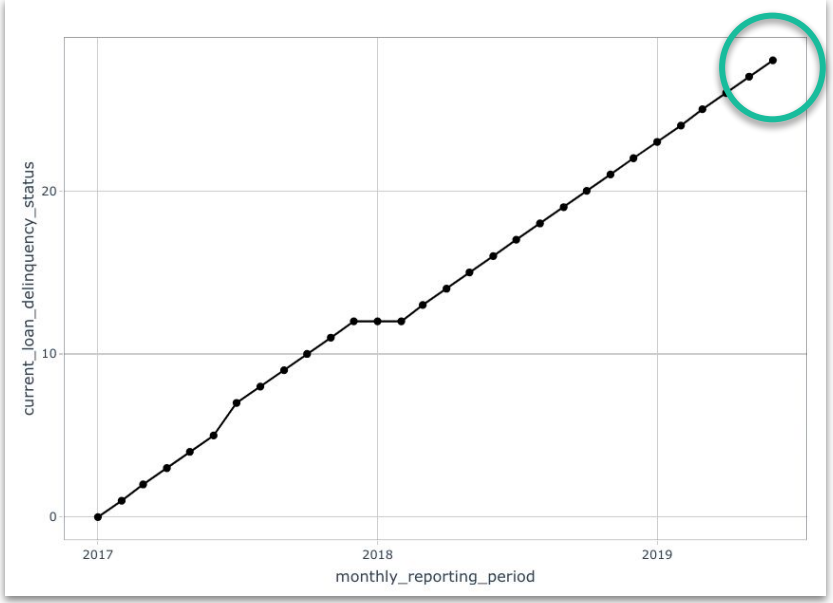
How far does the customer fall behind?

Loan ID:
138315939282

Unpaid Balance (UPB)



Delinquency Status



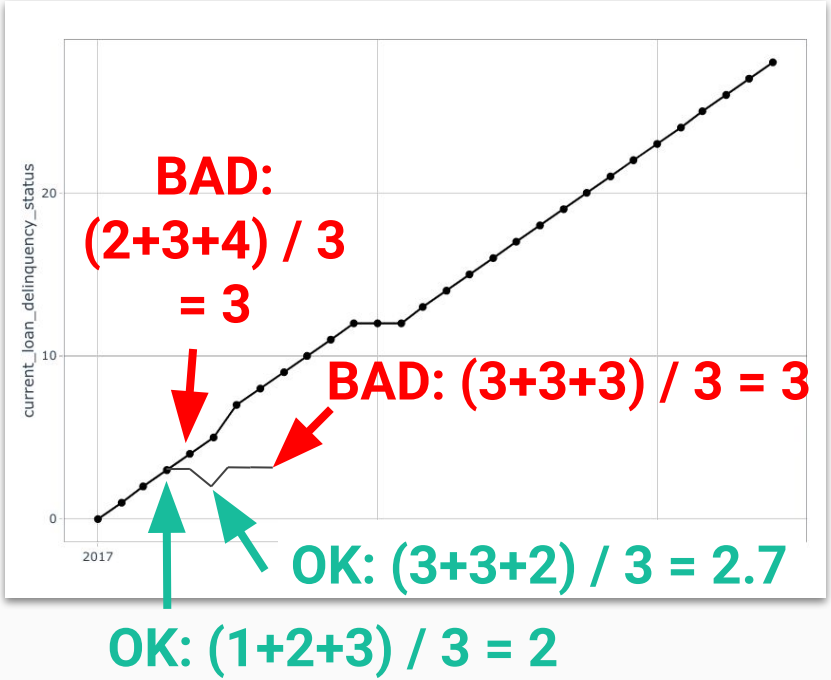
Using Moving Averages
To Make Machine Learning Targets

Use 3 Month Rolling Average
Want to give latitude to customers that can improve their trends.

Target Logic
If Customer has a pattern of **3 consecutive months** averaging more than 3 payments behind.

If max(rolling_avg_3) >= 3, Loan is Failure
If max(rolling_avg_3) < 3, Loan is OK

Prevent Trends from Getting Out of Control



Time Series Operations

80/20 Concepts

Time Series Operations

Rolling Windows

Lags & Diffs

Pct Change & Growth

Time Aggregations

Time Series Operations

3 Month Rolling Average

- Window Calculation
- Apply an aggregation function (e.g. mean())
- To sliding windows
- Allows us to monitor trend

Month	Sales	Moving Average
Jan-08	280	
Feb-08	356	
Mar-08	486	374
Apr-08	603	482
May-08	737	609
Jun-08	815	718
Jul-08	882	811
Aug-08	907	868
Sep-08	952	914
Oct-08	1004	954
Nov-08	1087	1014
Dec-08	1090	1060

Rolling Average

Time Series Operations

Lags & Differences

- Lags are offsets
- Differences are changes between previous and current
- Allows us to calculate Percent Change

Date	Value	$Value_{t-1}$	$Value_{t-2}$
1/1/2017	200	NA	NA
1/2/2017	220	200	NA
1/3/2017	215	220	200
1/4/2017	230	215	220
1/5/2017	235	230	215
1/6/2017	225	235	230
1/7/2017	220	225	235
1/8/2017	225	220	225
1/9/2017	240	225	220
1/10/2017	245	240	225

Percent Change

(Value[t] - Value[t-1]) / Value[t-1]

Time Series Operations

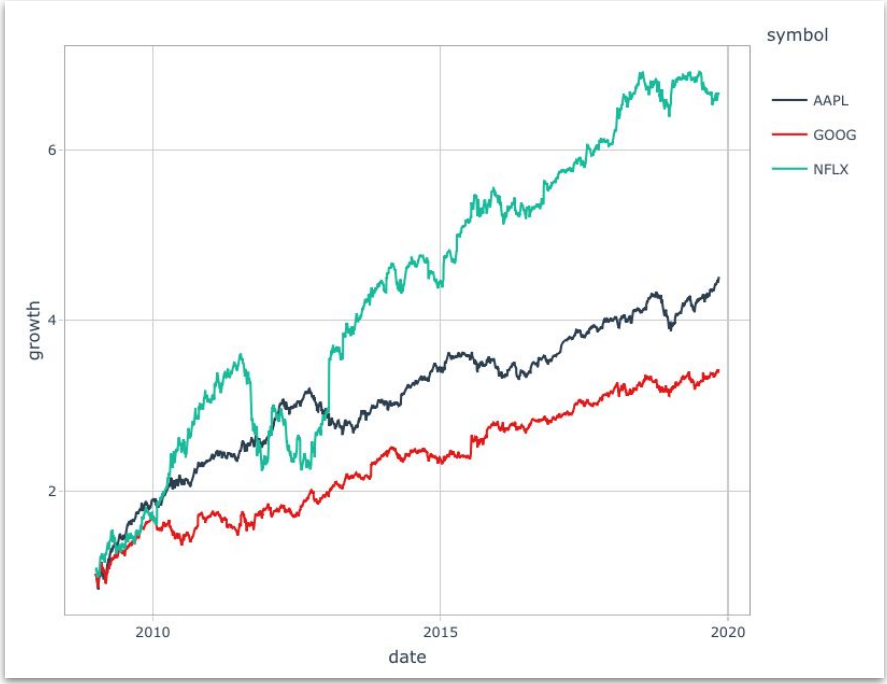
Lags & Differences

- Start with Pct Change = (Value[t] - Value[t-1]) / Value[t-1]
- Growth = cumsum(Pct Change) + C

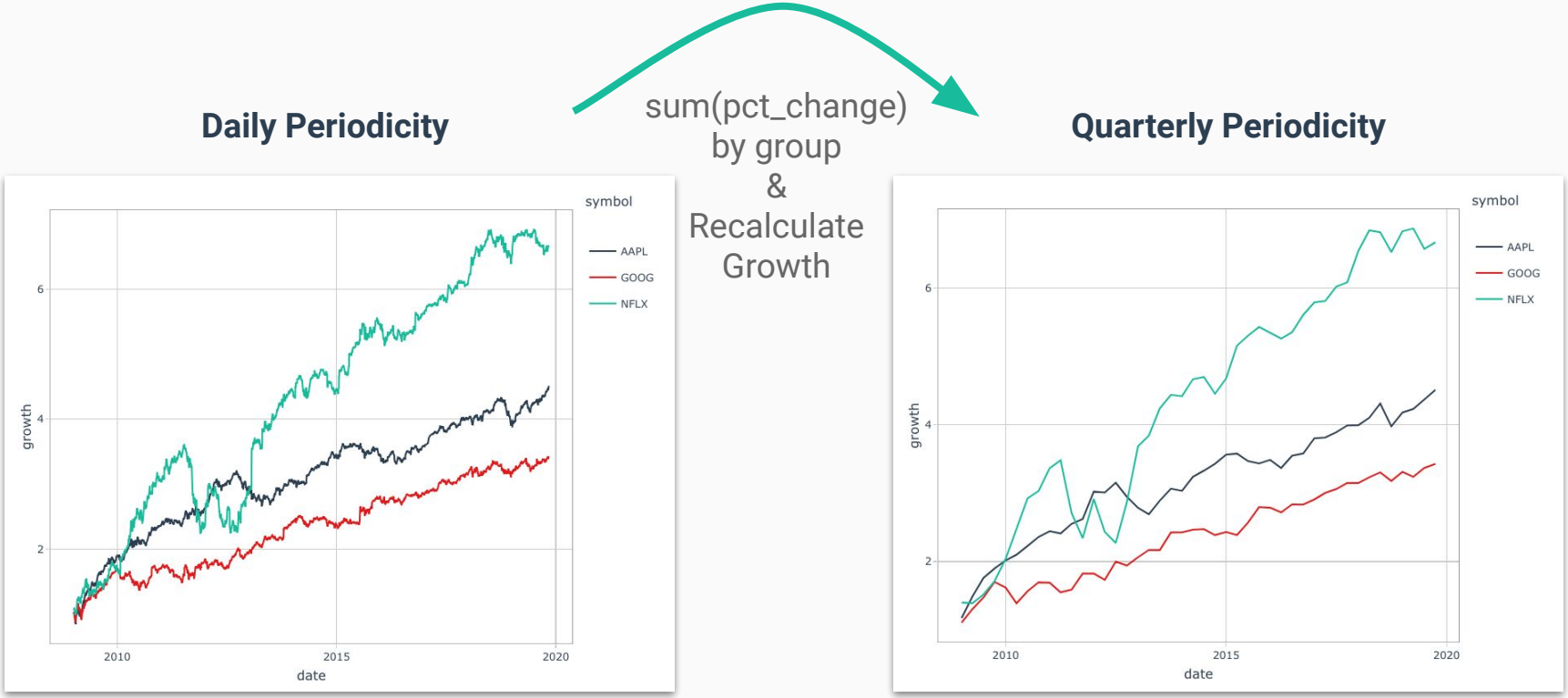
Date	Value	$Value_{t-1}$	$Value_{t-2}$
1/1/2017	200	NA	NA
1/2/2017	220	200	NA
1/3/2017	215	220	200
1/4/2017	230	215	220
1/5/2017	235	230	215
1/6/2017	225	235	230
1/7/2017	220	225	235
1/8/2017	225	220	225
1/9/2017	240	225	220
1/10/2017	245	240	225

Growth of $1

cumsum(pct_change) + 1



Time Series Operations



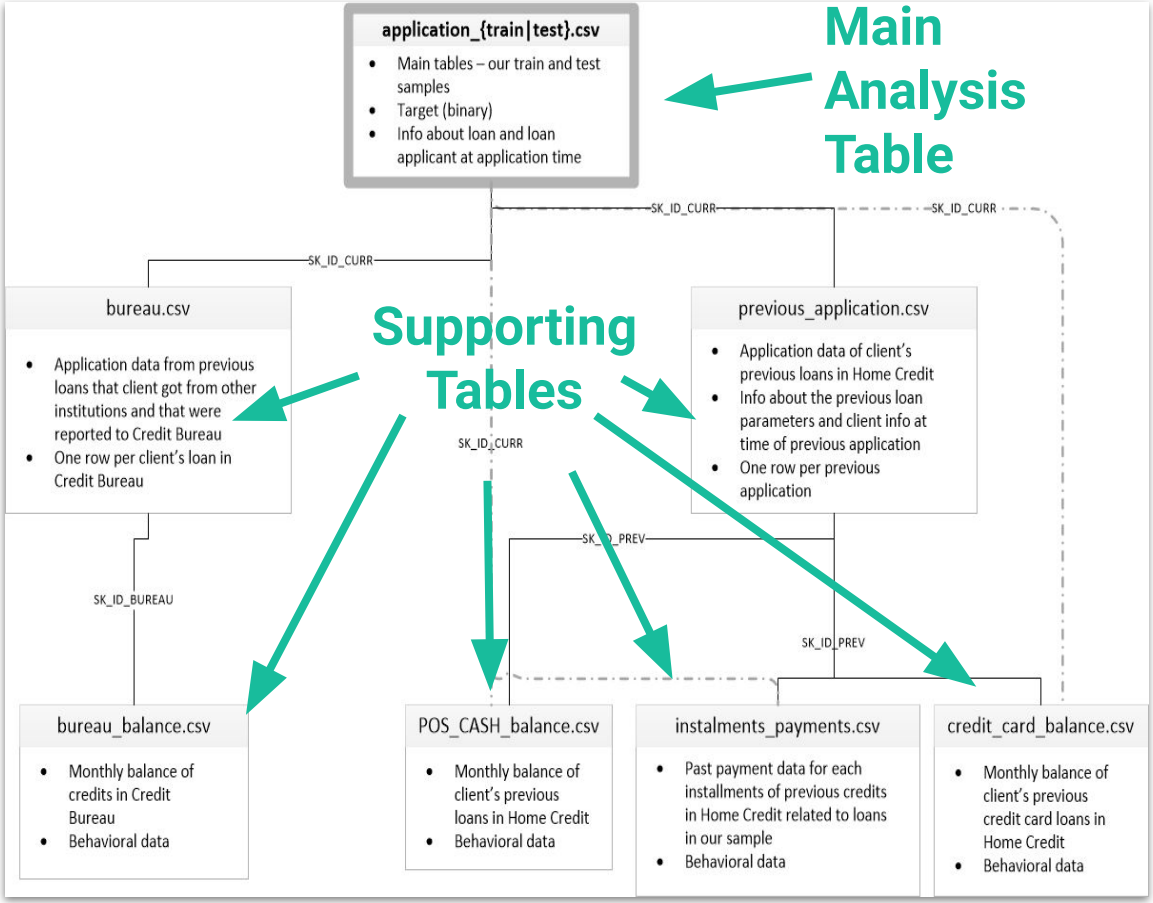
Feature Engineering w/ SQL

Recap from
Lab 21 - SQL for Data Science

SQL Database

1. Data stored in **SQL Tables**
2. **Relationships** between tables linked with common field called an ID (Primary Key)
3. Data Scientists can use tables to **generate features** & model business problem

Key Point - Can use relationships between tables to **generate critical features** to higher level data model



Relational Database (SQL)

Loan ID

01_stock_time_series_analysis.R | 02_fannie_time_series_analysis.R | acquisitions | 00_setup_stocks_sqlite.R

(Displaying up to 1,000 records)

loan_id	original_channel	seller_name	original_interest_rate	original_upb	original_loan_term
100002130634	R	QUICKEN LOANS INC.	4.375	159000	360
100003080256	B	UNITED SHORE FINANCIAL SERVICES, LLC D/B/A UNIT...	4.875	94000	360
100003722619	R	QUICKEN LOANS INC.	3.875	113000	240
100004837443	R	OTHER	3.500	198000	360
100007361597	C	OTHER	3.625	94000	360
100014264039	C	AMERIHOME MORTGAGE COMPANY, LLC	4.500	114000	360
100014590941	C	OTHER	4.750	90000	360
100015242204	R	OTHER	3.125	60000	180
100015715149	C	SUNTRUST BANK	4.250	165000	360
100019520450	C	SUNTRUST BANK	4.250	233000	360
100021015676	C	SUNTRUST BANK	3.000	251000	180
100021519010	B	OTHER	3.500	262000	360
100022856839	R	FREEDOM MORTGAGE CORP.	4.375	114000	180
100023490738	C	FLAGSTAR BANK, FSB	4.375	380000	360
100025374033	R	WELLS FARGO BANK, N.A.	3.000	215000	180
100027040138	R	OTHER	5.250	122000	360

Showing 1 to 17 of 1,000 entries, 26 total columns

Main Analysis Table

Loan Acquisition Data

- Each loan has 1 row
- Time Independent

01_stock_time_series_analysis.R | 02_fannie_time_series_analysis.R | acquisitions | performance | 00_setup_stocks_s

(Displaying up to 1,000 records)

loan_id	monthly_reporting_period	servicer_name	current_interest_rate	current_upb	loan_age	remaining_months_t
100002130634	2017-02-01	QUICKEN LOANS INC.	4.375	NA	1	359
100002130634	2017-03-01		4.375	NA	2	358
100002130634	2017-04-01		4.375	NA	3	357
100002130634	2017-05-01		4.375	NA	4	356
100002130634	2017-06-01		4.375	NA	5	355
100002130634	2017-07-01		4.375	NA	6	354
100002130634	2017-08-01		4.375	157167.59	7	353
100002130634	2017-09-01		4.375	156947.22	8	352
100002130634	2017-10-01		4.375	96507.29	9	351
100002130634	2017-11-01		4.375	96065.76	10	350
100002130634	2017-12-01		4.375	95556.57	11	349
100002130634	2018-01-01		4.375	95111.57	12	348
100002130634	2018-02-01		4.375	94664.95	13	347
100002130634	2018-03-01		4.375	94664.95	14	346
100002130634	2018-04-01		4.375	93766.82	15	345
100002130634	2018-05-01		4.375	93766.82	16	344

Showing 1 to 17 of 1,000 entries, 32 total columns

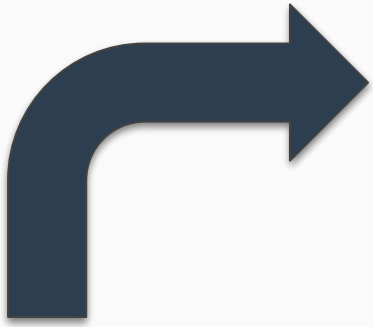
Supporting Table

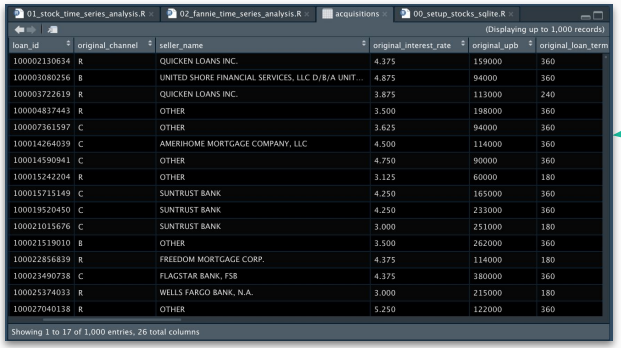
Loan Performance Data

- Each loan is a Time Series
- Time Dependence

Relational Database (SQL)

Compute Rolling Average
& Identify Loans with Bad Trends
Becomes our Target Variable



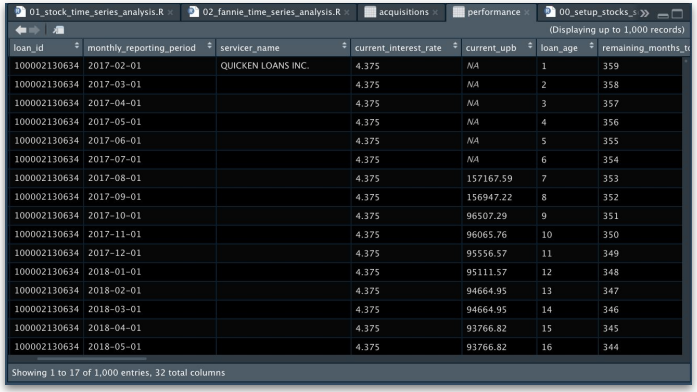


loan_id	original_channel	seller_name	original_interest_rate	original_upb	original_loan_term
100002130634	R	QUICKEN LOANS INC.	4.375	159000	360
100003080256	B	UNITED SHORE FINANCIAL SERVICES, LLC D/B/A UNIT...	4.875	94000	360
100003722619	R	QUICKEN LOANS INC.	3.875	113000	240
100004837443	R	OTHER	3.500	198000	360
100007361597	C	OTHER	3.625	94000	360
100014264039	C	AMERIHOME MORTGAGE COMPANY, LLC	4.500	114000	360
100014590941	C	OTHER	4.750	90000	360
100015242204	R	OTHER	3.125	60000	180
100015715149	C	SUNTRUST BANK	4.250	165000	360
100019520450	C	SUNTRUST BANK	4.250	233000	360
100021015676	C	SUNTRUST BANK	3.000	251000	180
100021519010	B	OTHER	3.500	262000	360
100022856839	R	FREEDOM MORTGAGE CORP.	4.375	114000	180
100023490738	C	FLAGSTAR BANK, FSB	4.375	380000	360
100025374033	R	WELLS FARGO BANK, N.A.	3.000	215000	180
100027040138	R	OTHER	5.250	122000	360

Main Analysis Table

Loan Acquisition Data

- Each loan has 1 row
- Time Independent



loan_id	monthly_reporting_period	servicer_name	current_interest_rate	current_upb	loan_age	remaining_months_t
100002130634	2017-02-01	QUICKEN LOANS INC.	4.375	NA	1	359
100002130634	2017-03-01		4.375	NA	2	358
100002130634	2017-04-01		4.375	NA	3	357
100002130634	2017-05-01		4.375	NA	4	356
100002130634	2017-06-01		4.375	NA	5	355
100002130634	2017-07-01		4.375	NA	6	354
100002130634	2017-08-01		4.375	157167.59	7	353
100002130634	2017-09-01		4.375	156947.22	8	352
100002130634	2017-10-01		4.375	96507.29	9	351
100002130634	2017-11-01		4.375	96065.76	10	350
100002130634	2017-12-01		4.375	95556.57	11	349
100002130634	2018-01-01		4.375	95111.57	12	348
100002130634	2018-02-01		4.375	94664.95	13	347
100002130634	2018-03-01		4.375	94664.95	14	346
100002130634	2018-04-01		4.375	93766.82	15	345
100002130634	2018-05-01		4.375	93766.82	16	344

Supporting Table

Loan Performance Data

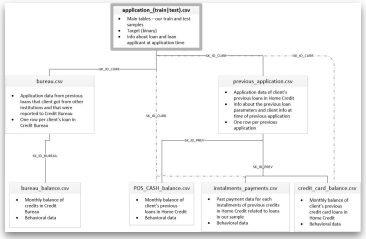
- Each loan is a Time Series
- Time Dependence

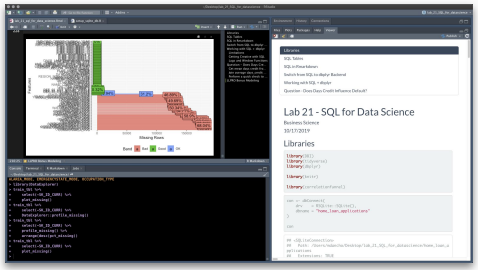
Databases are Fast

SQL & NoSQL are **optimized** to handle data & perform aggregations, filtering, etc.

Data takes a long time to **transfer** between machines.

SQL for Data Science Workflow

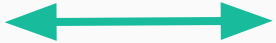




Maximize
Simple but
Expensive
Operations

Home Loans
Database
(SQL Database)

Minimize
Data
Transfer via
Aggregation



Matt's Computer
(MacBook Pro)

Perform
Complex
Data Science

SQL is painful

It costs you **time** to write SQL,

it's prone to **errors** & **difficult** to learn

SQL Translation with dplyr

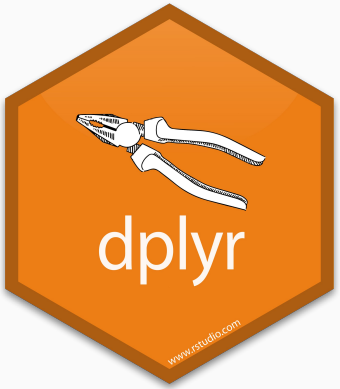
Database Table Connection

```
days_credit_query <- tbl(con, "bureau") %>%

    # Selet columns
    select(SK_ID_CURR, DAYS_CREDIT) %>%

    # Group by SK_ID_CURR and calculate average days credit
    group_by(SK_ID_CURR) %>%
    summarise(mean_days_credit = mean(DAYS_CREDIT, na.rm = T)) %>%
    ungroup() %>%

    # Arrange Descending by mean
    arrange(desc(mean_days_credit))
```

dplyr operations

dplyr translates to SQL

```
<SQL>
SELECT `SK_ID_CURR`, AVG(`DAYS_CREDIT`) AS `mean_days_credit`
FROM (SELECT `SK_ID_CURR`, `DAYS_CREDIT`
FROM `bureau`)
GROUP BY `SK_ID_CURR`
ORDER BY `mean_days_credit` DESC
```



30-Min Demo

Stock Analysis

Mortgage Loan Delinquency

Time series PRO-TIPS

Yeahhhhhh!

Time Series Pro-Tips

```
85  rolling_window_query <- tbl(con_stocks, "stock_history") %>%
86
87      select(symbol, date, adjusted) %>%
88
89      group_by(symbol) %>%
90
91      window_frame(from = -90, to = 0) %>%
92      window_order(date) %>%
93
94      mutate(roll_avg = mean(adjusted, na.rm = TRUE))  %>%
95
96      ungroup()
97
98  rolling_window_query %>% show_query()
99
```

```
> rolling_window_query %>% show_query()
<SQL>
SELECT `symbol`, `date`, `adjusted`, AVG(`adjusted`) OVER (PARTITION BY `symbol`
 ORDER BY `date` ROWS 90 PRECEDING) AS `roll_avg`
FROM (SELECT `symbol`, `date`, `adjusted`
FROM `stock_history`)
>
```

#1. Use group_by()

To partition you SQL database

#2. Use window_frame() & window_order()

To specify rolling windows within groups (partitions)

3. Apply aggregations using mutate()

To easily calculate rolling aggregations

2020 Data Science Guide

Solving NEW Business Needs

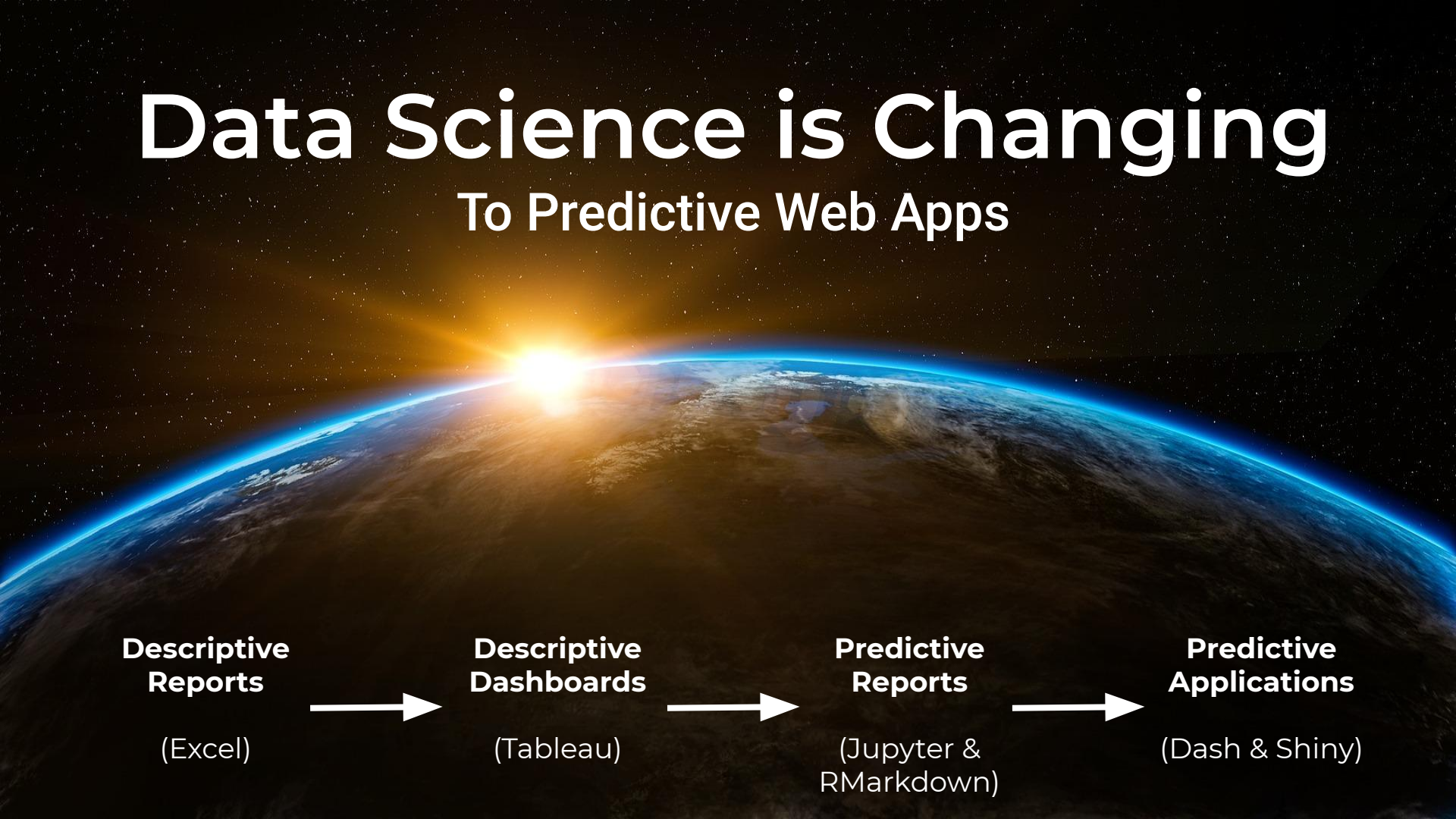
Business now need apps + cloud

2015
Reports + Servers

2020
Apps + Cloud

Data Science is Changing

To Predictive Web Apps



ML-App that identifies anomalous payments

SAFEAUTO

Payments Dashboard

Download full dataset

Data current through 27 Apr 2016

Data sourced from AutoPilot_Staging

Payment Type:

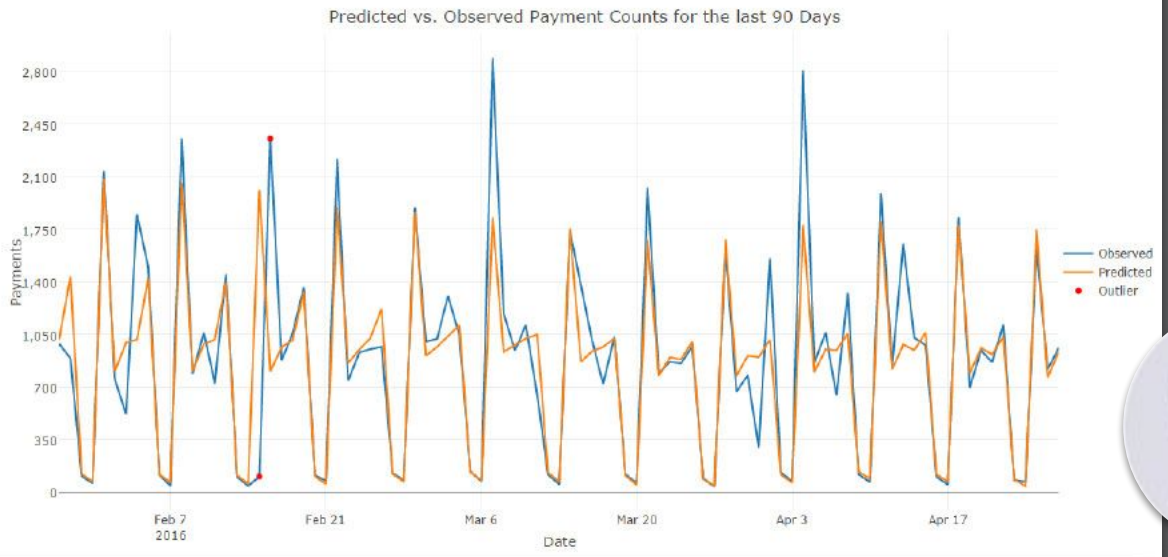
Cash

Payment Channel:

- All
- Bill Consolidator
- Lock Box
- Phone Rep
- Vendor Payment

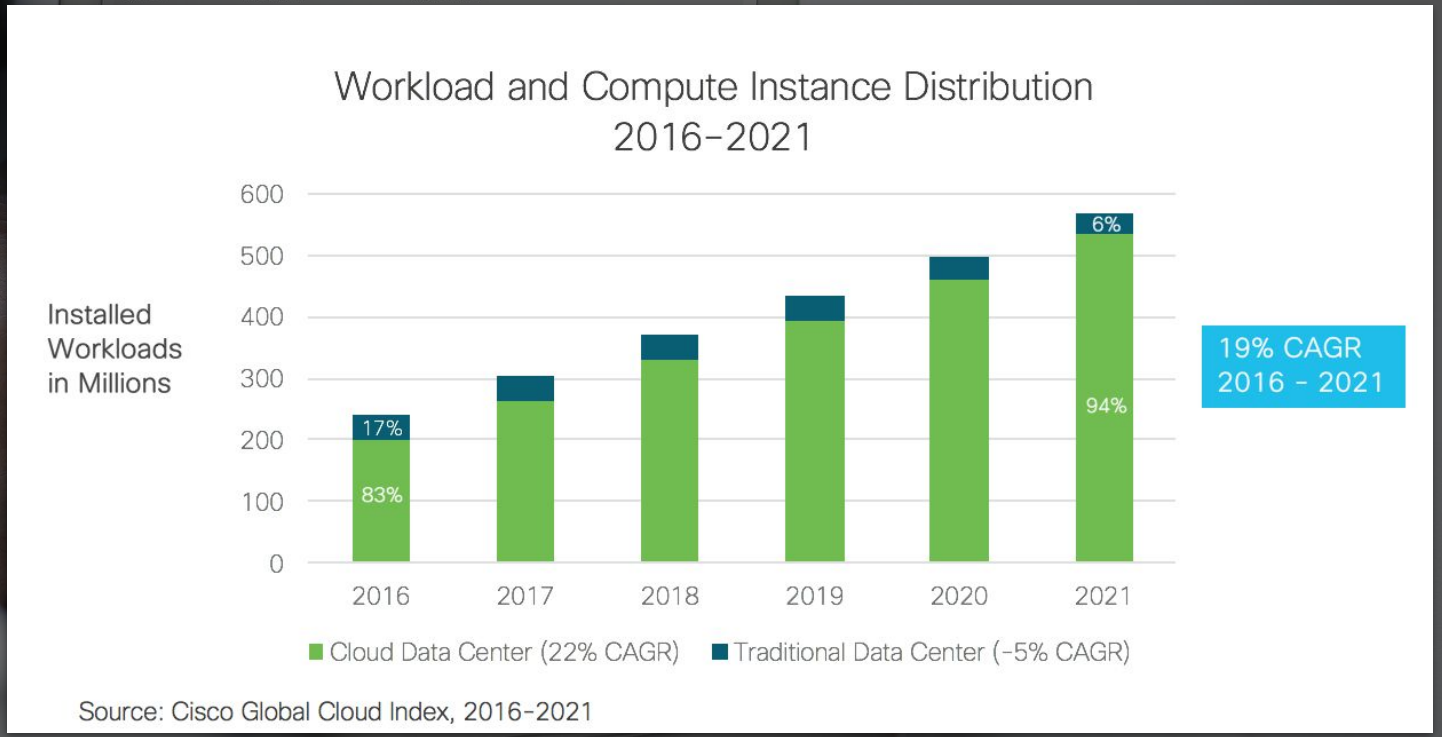
Trailing Payments | Payment Analytics | F.A.Q.

Predictions | Summary | Analytics F.A.Q.



Global Cloud Index

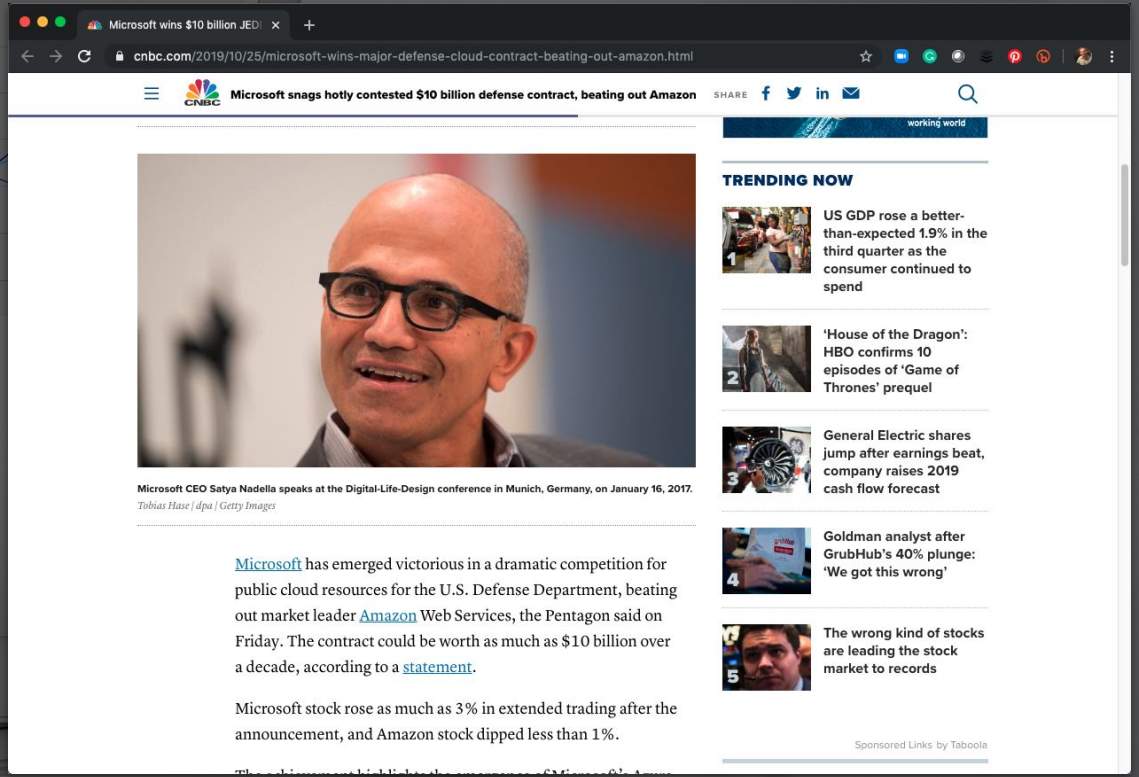
22% Annual Growth Rate vs -5% Data Center Growth



Microsoft snags hotly contested $10 Billion defense contract, beating out Amazon

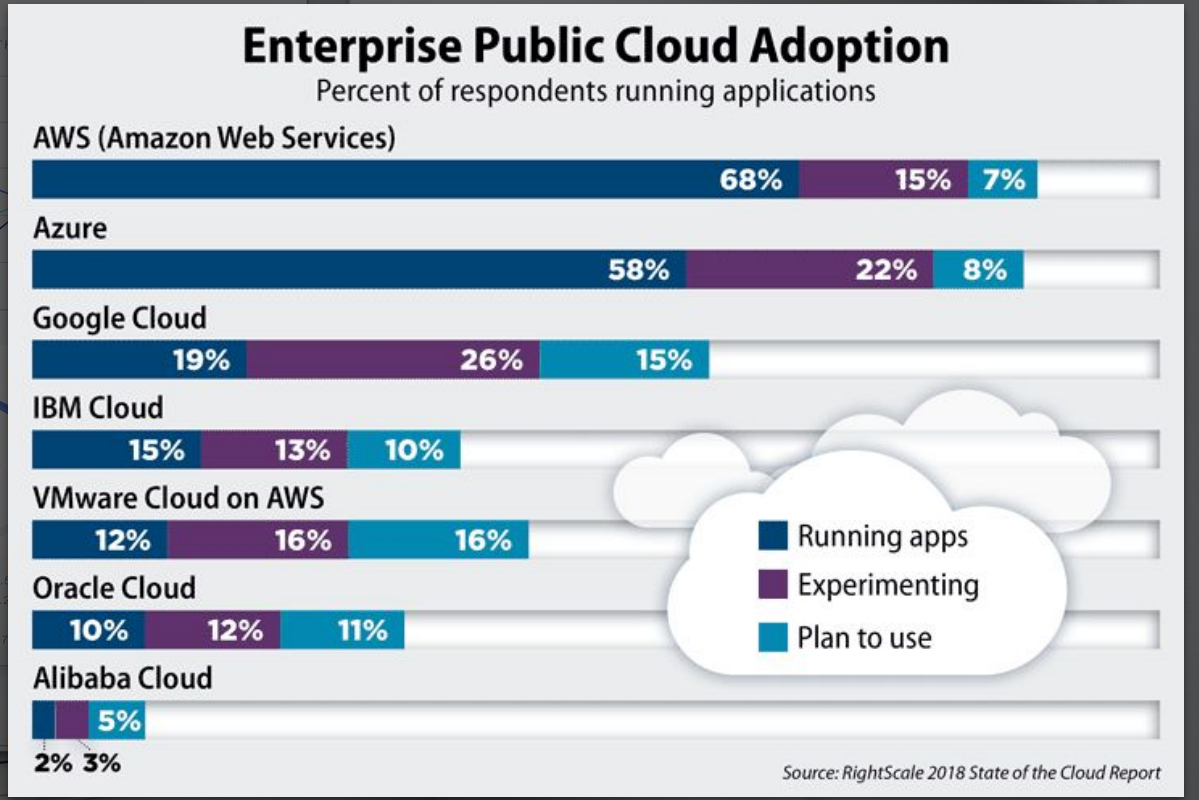
JEDI contract worth up to $10B over 10 years

*"The contract will provide the Pentagon with **cloud services** for basic storage and power all the way up to **artificial intelligence, machine learning, and the ability to process mission-critical workloads.**"*



Cloud Players at a Glance

- **Amazon Web Services (AWS)** - The market leader in enterprise & beyond
- **Microsoft Azure** - 2nd in Popularity; Popular with Enterprise
- **Google Cloud Platform (GCP)** - Popular with Digital Marketing because of integration with Google Analytics



A lot to learn

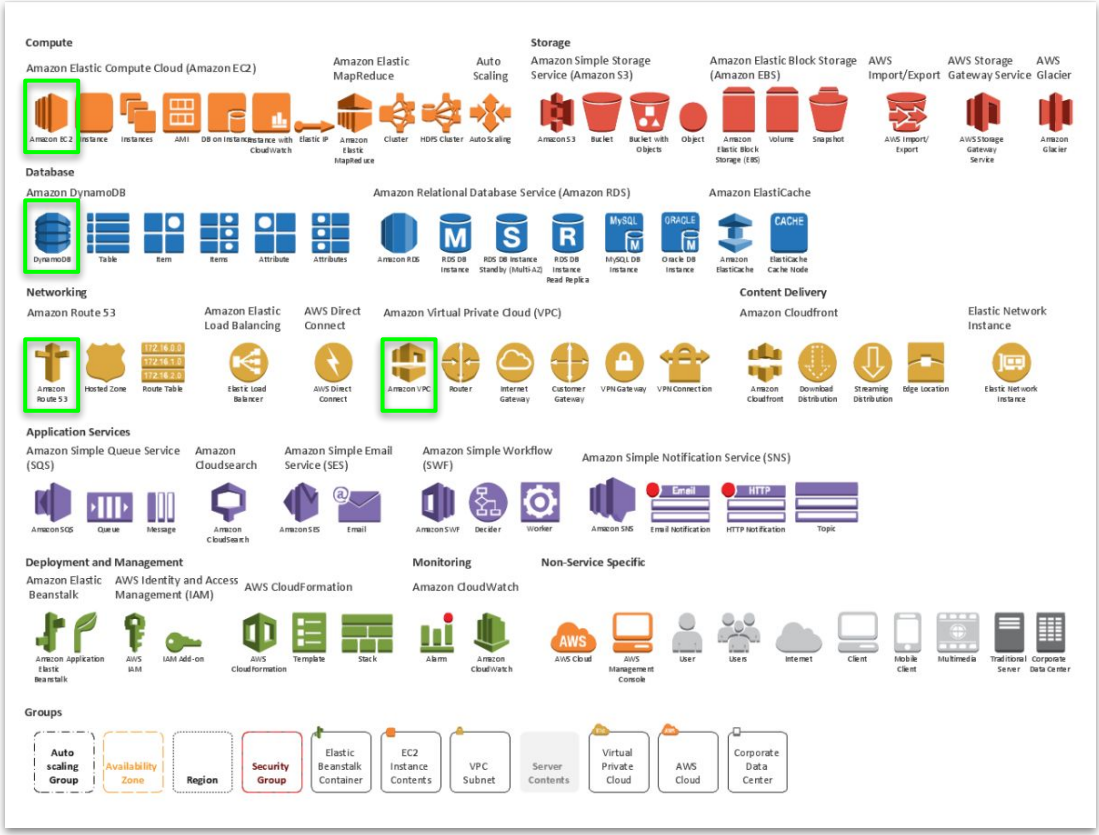
aws Cloud

Bootstrap

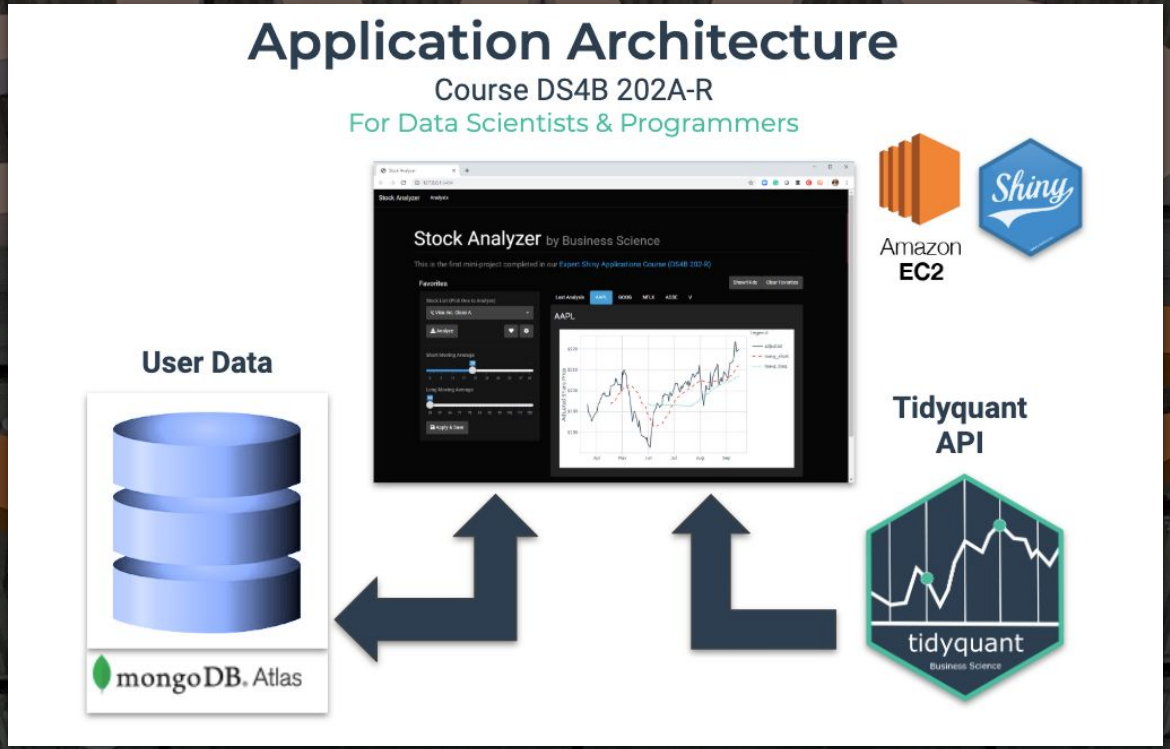
mongoDB.

Shiny Server

docker



Learn by building & deploying



4-Course R-Track System

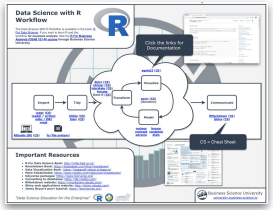
Business Analysis with R
(DS4B 101-R)

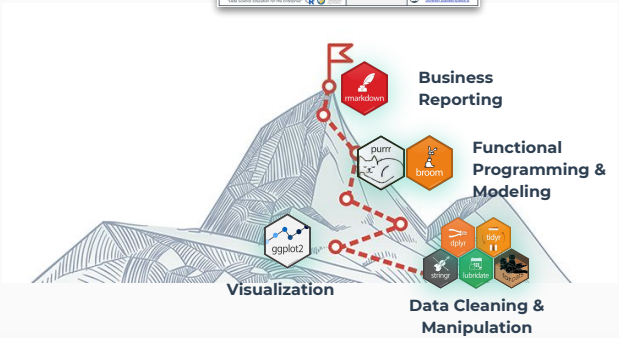
Data Science For Business with R
(DS4B 201-R)

Web Apps & Shiny Developer
(DS4B 102-R + DS4B 202A-R)

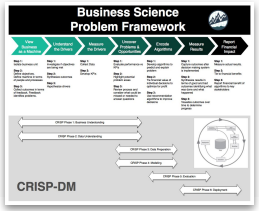
Project-Based Courses with Business Application

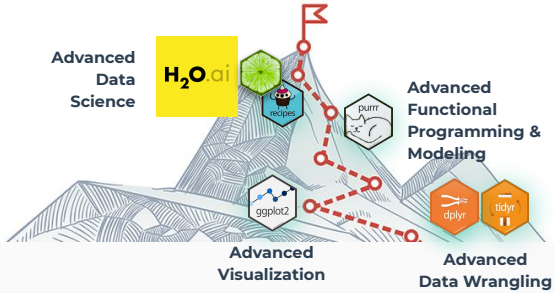
Data Science Foundations
7 Weeks



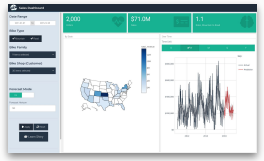


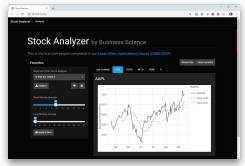
Machine Learning & Business Consulting
10 Weeks





Web Application Development
12 Weeks





15% OFF PROMO Code: learninglabs

R-TRACK BUNDLE

4-Course Bundle - Machine Learning + Expert Web Applications (R-Track)

Go from Beginner to Expert Data Scientist & Shiny Developer in Under 6-Months

4 Course Bundle ~~$1,596~~

$127/mo
Get started today!

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

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

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 202A-R: Expert Shiny Developer with AWS

Learn how to build Scalable Data Science Applications using R, Shiny, and AWS Cloud Technology.

Matt Dancho

○	**Paid Course** 15% COUPON DISCOUNT	~~$1,596~~ **$1,356.60**
◉	**12 Low Monthly Payments** 15% COUPON DISCOUNT 12X Payment Plan	~~12 payments of $149/m~~ **12 payments of $126.65/m**