



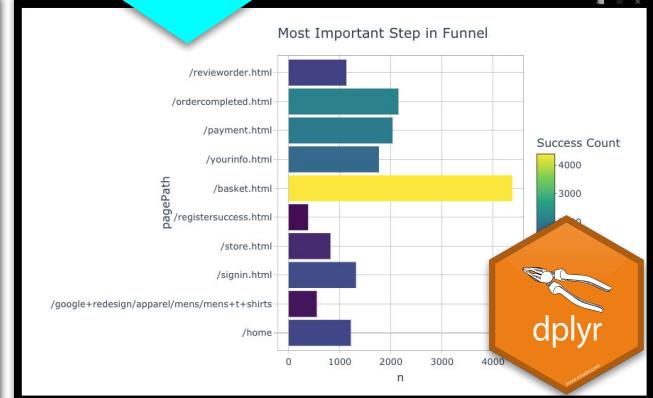
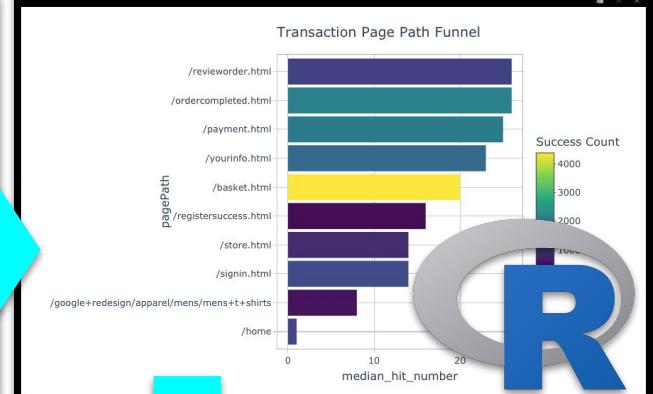
# Google BigQuery

# Google Analytics + BigQuery

## Conversion Funnel Analysis (with SQL & R)

```
336  { }
337  join_query <- str_glue(
338  "
339  SELECT * FROM ( {page_path_query} )
340  LEFT JOIN ( {transaction_visitId_query}
341  USING (visitId)
342  "
343  )
344  join_query
345  "
346

SELECT * FROM (
  SELECT fullVisitorId, visitId, visitNumber, h.hitNumber AS hitNumber, h.page.pagePath AS pagePath
  FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  LEFT JOIN UNNEST(hits) AS h
  WHERE _TABLE_SUFFIX BETWEEN '20170701' AND '20170701'
  AND h.type='PAGE'
  ORDER BY fullVisitorId, visitId, visitNumber
)
LEFT JOIN (
  SELECT visitId, SUM( totals.transactionRevenue ) AS total_transaction_revenue
  FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE _TABLE_SUFFIX BETWEEN '20170701' AND '20170701'
  GROUP BY visitId
  ORDER BY total_transaction_revenue DESC
)
USING (visitId)
```



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

Susan Welch

- Chemist/Data Analyst
- Marathon Runner!
- Took Business Science Courses
- App to Track Marathon Performance



#Business  
Science  
Success

***"Your courses gave me the skills to make this dashboard easily."***

My Marathons

Date Range

2002-11-28 to 2019-11-03

Boston Qualifier?

No     Yes

Map of My Marathons

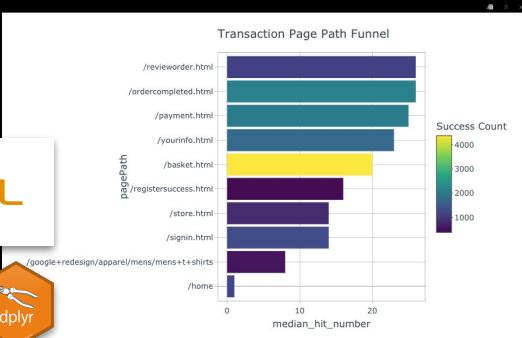
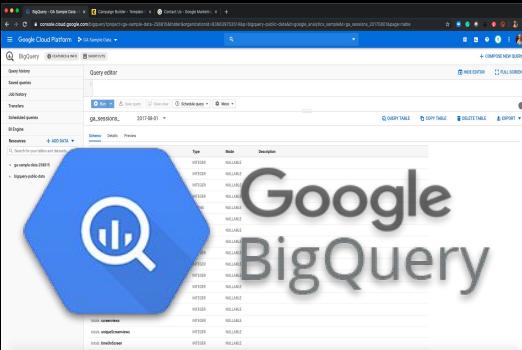
My Marathon Data

| Number | Date       | Marathon         | City        | Start | Result   | BO  |
|--------|------------|------------------|-------------|-------|----------|-----|
| 1      | 2002-11-28 | Atlanta Marathon | Atlanta, GA |       | 04:47:59 | No  |
| 2      | 2007-03-25 | Georgia Marathon | Atlanta, GA |       | 05:16:11 | No  |
| 3      | 2008-09-28 | Berlin Marathon  | Berlin, DE  |       | 04:08:43 | No  |
| 4      | 2009-09-20 | Berlin Marathon  | Berlin, DE  |       | 03:49:49 | Yes |
| 5      | 2010-04-28 | Hamburg Marathon | Hamburg, DE |       | 03:47:04 | Yes |
| 6      | 2012-05-27 | Buffalo Marathon | Buffalo, NY |       | 04:10:02 | No  |
| 7      | 2013-03-17 | Georgia Marathon | Atlanta, GA |       | 03:56:14 | No  |
| 8      | 2013-10-27 | Atlanta Marathon | Atlanta, GA |       | 03:43:58 | Yes |

Search:

Previous 1 2 3 Next

# Agenda



- **Business Case Study**
  - Google Merchandise Store
  - Conversion Funnel
- **30-Min Demo**
  - BigQuery SQL
- **Pro-Tips & Learning Guide**
  - Recap + Pro-Tips
  - Learning Plan
- **Google Analytics**
  - Terminology
- **Big Query**
  - 80/20 Data Concepts



# Learning Labs PRO

Every 2-Weeks

1-Hour Course

Recordings + Code + Slack

\$19/month

[university.business-science.io](http://university.business-science.io)

Lab 22  
**SQL for Time Series**

Lab 21  
**SQL for Data Science**

Lab 20  
**Explainable Machine Learning**

Lab 19  
**Using Customer Credit Card History for Networks Analysis**

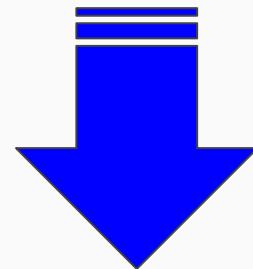
Lab 18  
**Time Series Anomaly Detection with anomalize**

Lab 17  
**Anomaly Detection with H2O Machine Learning**

Lab 16  
**R's Optimization Toolchain, Part 2**



Continuous Learning  
Jet Fuel for your Brain



Learning Labs Pro

Community-Driven Data Science Courses

 Matt Dancho

\$19/m

# Google Merchandise Store

Business Case



# Google Merchandise Store

## Google Analytics Data

Customers can purchase t-shirts, gear, etc

Google Analytics tracks every event on the website.

We can use this for **Conversion Funnel Analysis.**

<https://shop.googlemerchandise.com/>

The screenshot shows a web browser displaying the Google Merchandise Store at [shop.googlemerchandisestore.com](https://shop.googlemerchandisestore.com). The page features a header with the Google logo and 'official merchandise store'. Below the header is a promotional image of several Google t-shirts in red, green, blue, and grey, arranged on a colorful geometric background. Text on the page reads 'These tees are the bee's knees' and 'The speckled Google T-shirt is just one of the new tees this season.' A prominent blue 'Shop now' button is visible. At the bottom of the page, there are contact links for '1-855-300-2945', 'Email Us', and 'Customer Support', along with a note: 'Then go outside. We will take it.'



# Google Analytics Data

## For Google Merchandise Store

1. Google made their data public to help **Google Analytics 360** customers leverage BigQuery
2. **BigQuery** can store BILLIONS of GA Records
3. **Super granular** - Search Paths, Transactions, Hits... for every session
4. **Goal:** Analyze the **Conversion Funnel**

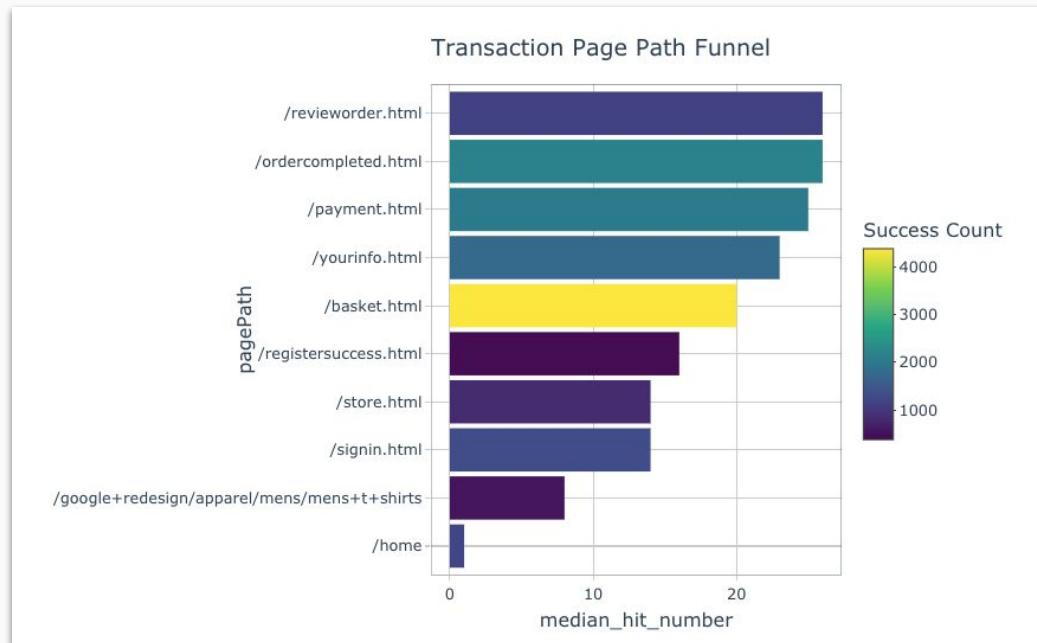
The diagram illustrates the data flow from Google Analytics Data to BigQuery. On the left, a screenshot of the Google Analytics interface shows a navigation summary chart for 'News' pages. A large blue arrow points from this interface to the right, where a screenshot of the Google Cloud Platform BigQuery interface is shown. The BigQuery interface displays a query editor with a schema for 'ga\_sessions' data, listing various fields such as visitId, visitNumber, visitStart, visitStartTime, date, visit, visitInfo, visitType, visitPageviews, visitSearches, visitTransactions, visitTransactionRevenue, visitOrderRevenue, visitOrderQuantity, visitOrderCompleted, and visitOrderStatus.

# Conversion Funnel Analysis



Understanding the **Transaction Funnel** is the most important concept for a **Digital Marketer**.

But which step(s) is/are most important?



**Key Concept** - The hit number is the sequence of events.

- Hit No. 1 is the first interaction.
- Hit No. 20 is the 20th interaction.
- Median Hit No. helps us identify the event position in the sequence.

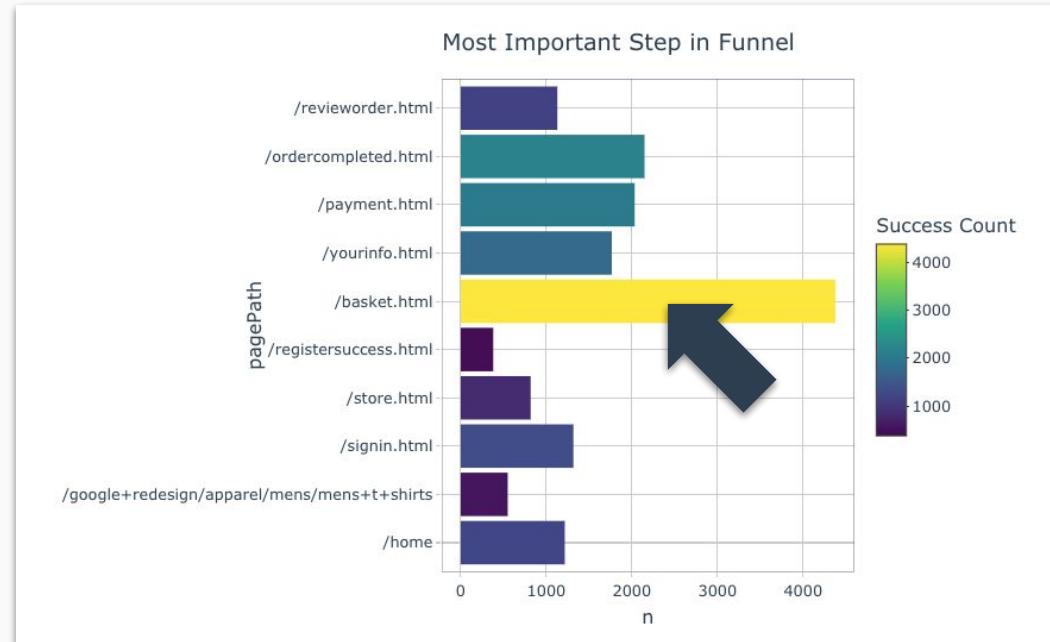


# Conversion Funnel Analysis

Understanding the **Transaction Funnel** is the most important concept for a **Digital Marketer**.

But which step(s) is/are most important?

Adding items to the **basket** is **4X more frequent in successful transactions** than visiting home; **6X more** than the store.



# Google Analytics

## Terminology



# Terminology

## Session

Record for all user interactions with a website from entry to exit.

Session Record Stores:

- Hits
- Transactions
- And more

```
110 ````{r}
111 # Tibble - Can also inspect as a nested tibble
112 # - Filter to get session info first visitID == 1470117657
113 hits_tbl <-tbl(con, "ga_sessions_20160801") %>%
114   filter(visitId == 1470117657) %>%
115   select(visitId, hits) %>%
116   collect()
117 ````
```

Visitor's Session

| ID | visitId             | hits               |
|----|---------------------|--------------------|
|    | <int><br>1470117657 | <list><br><tibble> |

1 row

Hits-Level data



# Terminology

## Hits

Tracks events that happen during the session.

- Pages the User Navigates To
- Social Interactions
- Ecommerce Hit

The screenshot shows a terminal window with R code and its output. The code is as follows:

```
125 ````{r}
126 # Drill into hits column for visitID == 1470117657
127 hits_tbl %>% pluck("hits", 1)
128
129
130 ````{r}
131 # Drill into hits > pages for visitID == 1470117657
132 hits_tbl %>%
133   pluck("hits", 1, "page") %>%
134   ...
```

The output is a data frame titled "Hits Data" with the following structure:

| hitNumber | time  | hour | minute | isSecure | isInteraction | isEntrance | isExit |
|-----------|-------|------|--------|----------|---------------|------------|--------|
| 1         | 0     | 23   | 0      | NA       | TRUE          | TRUE       | NA     |
| 2         | 30507 | 23   | 1      | NA       | TRUE          | NA         | NA     |
| 3         | 53670 | 23   | 1      | NA       | TRUE          | NA         | TRUE   |

Annotations: A red arrow points from the text "Hits Data" to the table title. Two other red arrows point from the right side of the table towards the "isExit" column header.



## Page Paths

Each page that an event occurs on has a pagePath associated with it.

GA tracks these page paths.

## Page Paths & Titles

```
130 ~ ``{r}
131 # Drill into hits > pages for visitID == 1470117657
132 hits_tbl %>%
133   pluck("hits", 1, "page") %>%
134   toJSON() %>%
135   pretty() %>%
136   fromJSON() %>%
137   unnest(cols = pagePath:pagePathLevel4)
138 ...
139
140
141 ~ ``{r}
142 # Drill into hits > transactions for visitID == 1470117657
143 hits_tbl %>%
```

| pagePath<br><chr>           | hostname<br><chr>          | pageTitle<br><chr>                    |
|-----------------------------|----------------------------|---------------------------------------|
| /home                       | shop.googlemerchandise.com | Home                                  |
| /google+redesign/gift+cards | shop.googlemerchandise.com | Gift Cards                            |
| /signin.html                | shop.googlemerchandise.com | The Google Merchandise Store - Log In |

3 rows | 1-3 of 9 columns

# BigQuery

## 80/20 Data Concepts

# BigQuery + Google Analytics 360 Integration



Google Cloud Platform • GA Sample Data ▾

BigQuery FEATURES & INFO SHORTCUTS

+ COMPOSE NEW QUERY

Query history Saved queries Job history Transfers Scheduled queries BI Engine Resources + ADD DATA ▾

ga

- ga-sample-data-258815
- bigquery-public-data
  - baseball
    - games\_post\_wide
    - games\_wide
  - eclipse\_megamovie
  - geo\_census\_tracts
    - census\_tracts\_michigan
  - geo\_us\_census\_places
    - places\_michigan
  - google\_analytics\_sample
    - ga\_sessions\_(366)
- human\_variant\_annotation
  - gnomad\_exac\_hg19\_rele...
- london\_fire\_brigade

Query editor

1

Run Save query Save view Schedule query More

ga\_sessions\_ 2017-08-01

QUERY TABLE COPY TABLE DELETE TABLE EXPORT

Schema Details Preview

| Field name                     | Type    | Mode     | Description |
|--------------------------------|---------|----------|-------------|
| visitorId                      | INTEGER | NULLABLE |             |
| visitNumber                    | INTEGER | NULLABLE |             |
| visitId                        | INTEGER | NULLABLE |             |
| visitStartTime                 | INTEGER | NULLABLE |             |
| date                           | STRING  | NULLABLE |             |
| totals                         | RECORD  | NULLABLE |             |
| totals.visits                  | INTEGER | NULLABLE |             |
| totals.hits                    | INTEGER | NULLABLE |             |
| totals.pageviews               | INTEGER | NULLABLE |             |
| totals.timeOnSite              | INTEGER | NULLABLE |             |
| totals.bounces                 | INTEGER | NULLABLE |             |
| totals.transactions            | INTEGER | NULLABLE |             |
| totals.transactionRevenue      | INTEGER | NULLABLE |             |
| totals.newVisits               | INTEGER | NULLABLE |             |
| totals.screenviews             | INTEGER | NULLABLE |             |
| totals.uniqueScreenviews       | INTEGER | NULLABLE |             |
| totals.timeOnScreen            | INTEGER | NULLABLE |             |
| totals.totalTransactionRevenue | INTEGER | NULLABLE |             |
| totals.sessionQualityDim       | INTEGER | NULLABLE |             |

Designed for Petabyte Scale

Can store Billions of Google Analytics Records

Integrates with Google Analytics 360

# Table Structure



Tables is time-based by each day

Comes in handy with  
**Wildcard Search** to  
combine tables (more  
on this later)

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, there's a sidebar with 'Recent tables' containing 'ga\_sessions\_20170101', 'ga\_sessions\_20170102', and 'ga\_sessions\_20170103'. The main area has three tabs: 'ga\_sessions\_20170101', 'ga\_sessions\_20170102', and 'ga\_sessions\_20170103'. Each tab displays a 'Query editor' window with a single row of data: 'ga\_sessions'.

ga\_sessions\_20170101

ga\_sessions\_20170102

ga\_sessions\_20170103



# Table Structure

## Data can be Nested

- Different than Redshift & Azure, which are built on PostgreSQL
- Stored as JSON, similar to mongoDB NoSQL

BigQuery uses SQL Dialect with special features:

- NEST() & UNNEST()

```
110 ````{r}
111 # Tibble - Can also inspect as a nested tibble
112 # - Filter to get session info first visitID == 1470117657
113 hits_tbl <- tbl(con, "ga_sessions_20160801") %>%
114   filter(visitId == 1470117657) %>%
115   select(visitId, hits) %>%
116   collect()
117 ````
```

Visitor's Session

| ID | visitId    | hits     |
|----|------------|----------|
|    | 1470117657 | <tibble> |

Hits-Level data

```
125 ````{r}
126 # Drill into hits column for visitID == 1470117657
127 hits_tbl %>% pluck("hits", 1)
128 ````
```

| hitNumber | time  | hour | minute | isSecure | isInteraction | isEntrance | isExit |
|-----------|-------|------|--------|----------|---------------|------------|--------|
| 1         | 0     | 23   | 0      | NA       | TRUE          | TRUE       | NA     |
| 2         | 30507 | 23   | 1      | NA       | TRUE          | NA         | NA     |
| 3         | 53670 | 23   | 1      | NA       | TRUE          | NA         | TRUE   |

Hits Data

```
129
130 ````{r}
131 # Drill into hits > pages for visitID == 1470117657
132 hits_tbl %>%
133   pluck("hits", 1, "page") %>%
```



# Table Structure

## Data can be **Nested**

- Different than Redshift & Azure, which are built on PostgreSQL
- Stored as JSON, similar to mongoDB NoSQL

BigQuery uses SQL Dialect with special features:

- NEST() & UNNEST()

| Field name                            | Type    | Mode     |
|---------------------------------------|---------|----------|
| visitorId                             | INTEGER | NULLABLE |
| visitNumber                           | INTEGER | NULLABLE |
| visitId                               | INTEGER | NULLABLE |
| visitStartTime                        | INTEGER | NULLABLE |
| date                                  | STRING  | NULLABLE |
| <b>totals</b>                         | RECORD  | NULLABLE |
| <b>totals.visits</b>                  | INTEGER | NULLABLE |
| <b>totals.hits</b>                    | INTEGER | NULLABLE |
| <b>totals.pageviews</b>               | INTEGER | NULLABLE |
| <b>totals.timeOnSite</b>              | INTEGER | NULLABLE |
| <b>totals.bounces</b>                 | INTEGER | NULLABLE |
| <b>totals.transactions</b>            | INTEGER | NULLABLE |
| <b>totals.transactionRevenue</b>      | INTEGER | NULLABLE |
| <b>totals.newVisits</b>               | INTEGER | NULLABLE |
| <b>totals.screenviews</b>             | INTEGER | NULLABLE |
| <b>totals.uniqueScreenviews</b>       | INTEGER | NULLABLE |
| <b>totals.timeOnScreen</b>            | INTEGER | NULLABLE |
| <b>totals.totalTransactionRevenue</b> | INTEGER | NULLABLE |
| <b>totals.sessionQualityDim</b>       | INTEGER | NULLABLE |

# 30-Min Demo

## Home Loan Applications

# PRO-TIPS

Yeahhhhhh!

# Pro-Tips



```
332 ````{r}
333 join_query <- str_glue(
334 "
335 SELECT * FROM ( {page_path_query} )
336 LEFT JOIN ( {transaction_visitId_query} )
337 USING (visitId)
338 "
339 )
340
341 join_query
342 ````

SELECT * FROM (
SELECT fullVisitorId, visitId, visitNumber, h.hitNumber AS hitNumber, h.page.pagePath AS pagePath
FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`
LEFT JOIN UNNEST(hits) AS h
WHERE _TABLE_SUFFIX BETWEEN '20170701' AND '20170731'
AND h.type='PAGE'
ORDER BY fullVisitorId, visitId, visitNumber, hitNumber
)
LEFT JOIN (
SELECT visitId, SUM( totals.transactionRevenue ) AS total_transaction_revenue
FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE _TABLE_SUFFIX BETWEEN '20170701' AND '20170731'
GROUP BY visitId
ORDER BY total_transaction_revenue DESC
)
USING (visitId)
```

## #1. Use dplyr if you don't need to UNNEST

dbplyr backend connects to BigQuery

## #2. If UNNEST is required, break up complex joins into smaller queries

Test sub-queries on data

## # 3. Combine with str\_glue()

Use str\_glue() to form complex joins

# **What We Just Did**

And how WE did it!



# Conversion Funnel Workflow

## Step-By-Step



### bigrquery

Connect to BigQuery database containing  
Google Analytics sessions data

### dbplyr & SQL Query

Aggregations & Joins

### dplyr & ggplot2

Visualize Conversion Funnel  
Detect Most Important Step(s)

# bigrquery



## Google BigQuery

```
50 - ````{r}
51 # Connect to BigQuery
52 bigrquery::bq_auth() # This will use your
cost to run this analysis.
53
54 con <- dbConnect(
55   bigrquery::bigquery(),
56   project = "bigquery-public-data",
57   dataset = "google_analytics_sample",
58   billing = "ga-sample-data-258815"
59 )
60
61 con
62 ````
```

A screenshot of the RStudio interface showing the 'Connections' tab selected. The main pane displays a list of BigQuery connections under the heading 'BigQueryConnection'. The 'Default' connection is selected, showing its schema. Other connections listed are 'ga\_sessions\_20160801' through 'ga\_sessions\_20160814'. The right sidebar shows a tree view of the connection structure.

# dbplyr & SQL



```
193 ##### Method 3 - Use BigQuery Wildcard and Table Suffixes
194
195 - Pros: Much more scalable - takes 14 seconds for all 365-days_
196 - Cons: For nested data, we need to know BigQuery Standard SQL
197
198 ````{r, eval=F}
199 tic()
200 unique_visitors_tbl <-tbl(con, "ga_sessions_*") %>%
201   # filter(`_TABLE_SUFFIX` %>% between("20160801", "20160810")) %>%
202   distinct(date, fullVisitorId) %>%
203   count(date) %>%
204   collect()
205 toc()
206
207 unique_visitors_tbl
208 ````
```



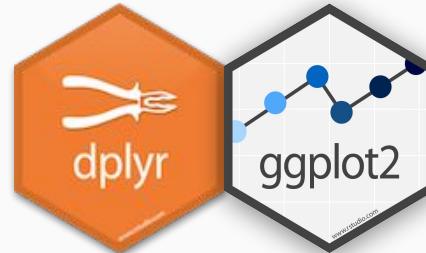
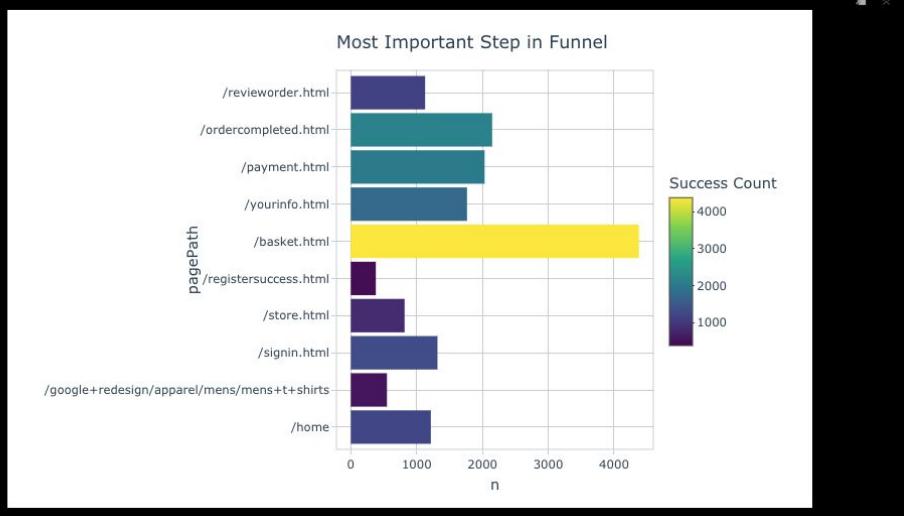
```
332 ````{r}
333 join_query <- str_glue(
334 "
335 SELECT * FROM ( {page_path_query} )
336 LEFT JOIN ( {transaction_visitId_query})
337 USING (visitId)
338 "
339 )
340
341 join_query
342 ````

SELECT * FROM (
SELECT fullVisitorId, visitId, visitNumber, h.hitNumber AS hitNumber, h.page.pagePath AS pagePath
FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`
LEFT JOIN UNNEST(hits) AS h
WHERE `_TABLE_SUFFIX` BETWEEN '20170701' AND '20170731'
AND h.type='PAGE'
ORDER BY fullVisitorId, visitId, visitNumber, hitNumber
)
LEFT JOIN (
SELECT visitId, SUM( totals.transactionRevenue ) AS total_transaction_revenue
FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE `_TABLE_SUFFIX` BETWEEN '20170701' AND '20170731'
GROUP BY visitId
ORDER BY total_transaction_revenue DESC
)
USING (visitId)
```



# dplyr & ggplot2

```
398 ~~~~{r}
399 g <- success_page_path_summary_tbl %>%
400   slice(1:10) %>%
401   arrange(median_hit_number) %>%
402   mutate(pagePath = as_factor(pagePath)) %>%
403 
404 ggplot(aes(pagePath, n, fill = n)) +
405   geom_col() +
406   coord_flip() +
407   theme_tq() +
408   scale_fill_viridis_c() +
409   labs(
410     title = "Most Important Step in Funnel",
411     fill = "Success Count"
412   )
413 
414 ggpplotly(g)
415 }
```



Adding items to the basket is **4X more frequent** in successful transactions than visiting home; **6X more** than the store.



# A lot more to learn... Advanced ML



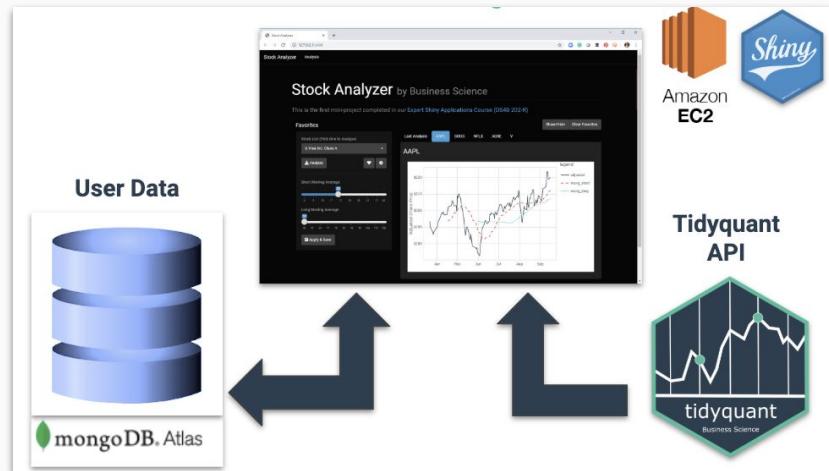
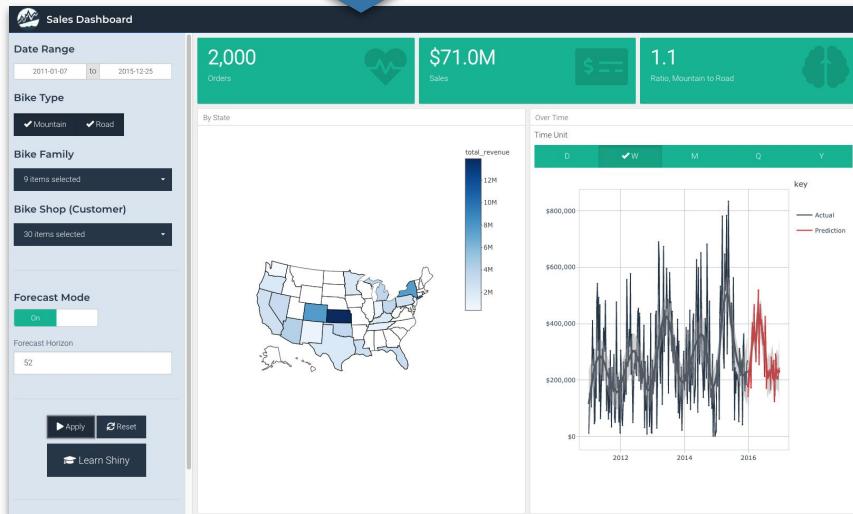
```
> h2o.predict(h2o_model, newdata = as.h2o(credit_card_group_tbl)) %>%
+   as_tibble()
# A tibble: 1,125 x 7
  predict      p1      p2      p3      p4      p5 Other
  <fct>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 Other     0.0000704 0.0228     0       0     0.977
2 Other    0.0232    0.0000717 0.0000553 0       0.00376 0.973
3 Other     0       0.0000737 0.0238     0       0.000107 0.976
4 Other    0.00643   0.0000724 0.0000558 0.00343  0.000105 0.990
5 Other     0       0.0000720 0.0000555 0       0.000104 1.000
6 3        0       0.0000704 0.909     0       0.000102 0.0909
7 3        0       0.0000761 0.995     0       0.000110 0.00491
8 1        0.984    0.0000735 0.0000567 0.00349  0.000106 0.0127
9 Other    0.195    0.0000602 0.0000464 0.00285  0.0000870 0.802
10 Other     0       0.0000737 0.0000568 0       0       1.000
# ... with 1,115 more rows
```

H2O AutoML

# A lot more to learn... Web Apps



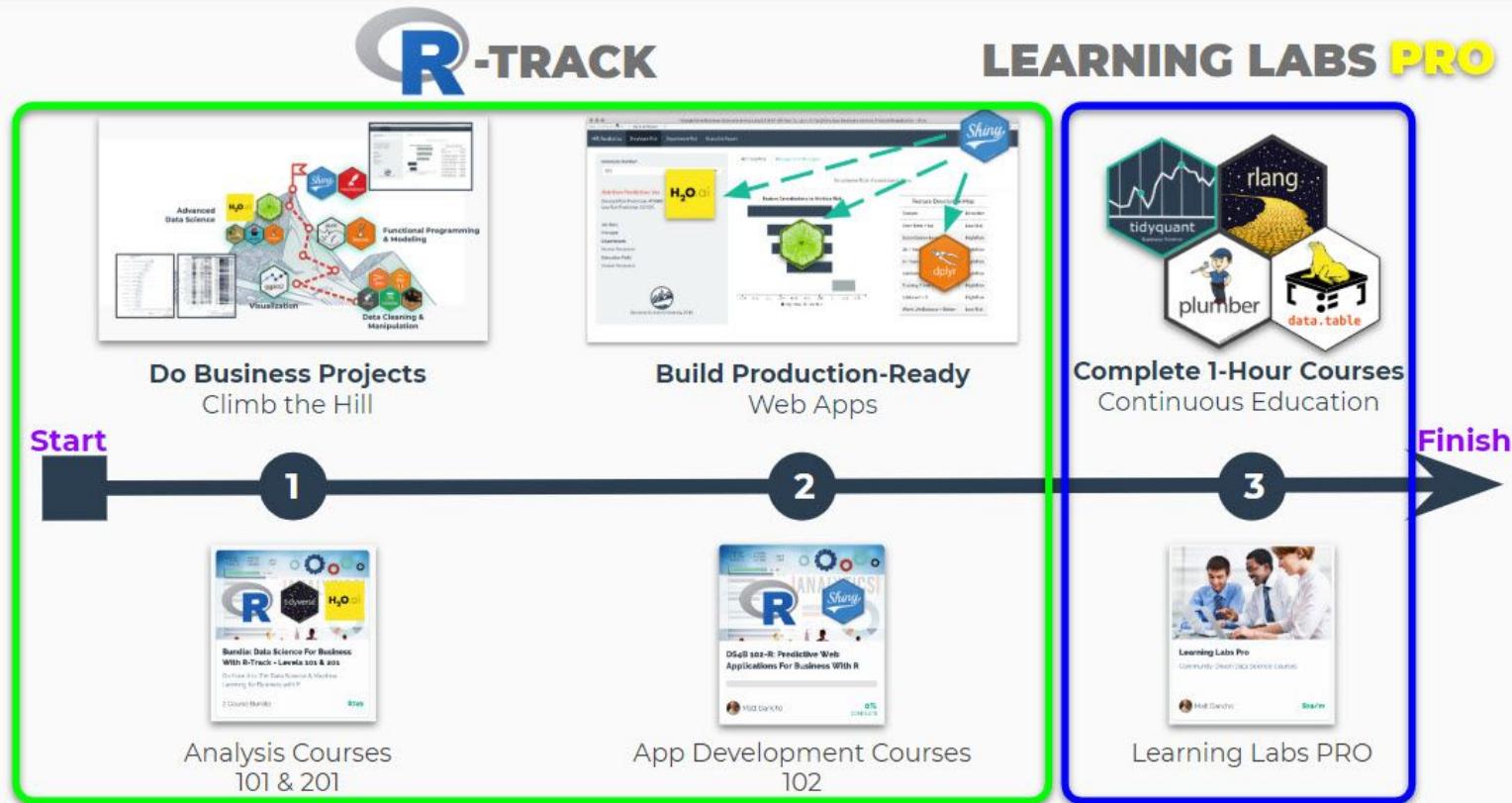
102 & 202A



# **Business Science University**

Learn Data Science for Business in 6-Months

# The program that will deliver YOUR Transformation



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

# 4-Course R-Track System



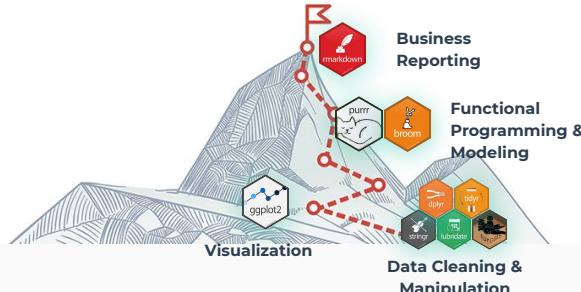
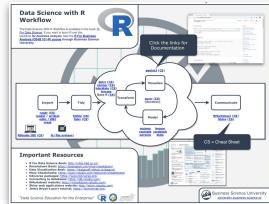
## Business Analysis with R (DS4B 101-R)

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

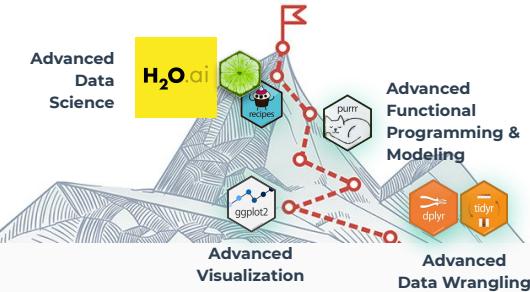
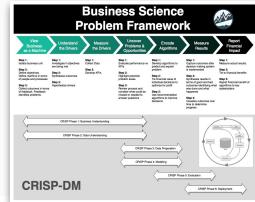
## Web Apps & Shiny Developer (DS4B 102-R + DS4B 202A-R)

### Project-Based Courses with Business Application

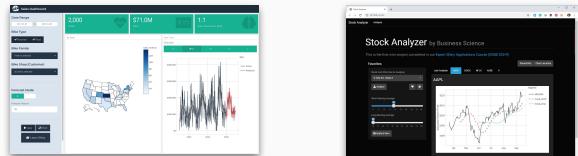
Data Science Foundations  
**7 Weeks**



Machine Learning & Business Consulting  
**10 Weeks**



Web Application Development  
**12 Weeks**

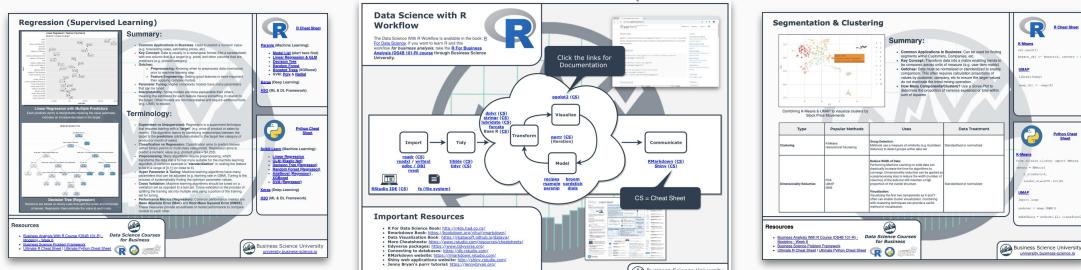


# Key Benefits

- 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

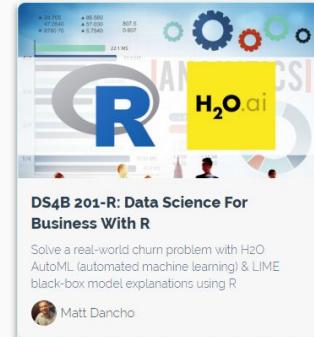
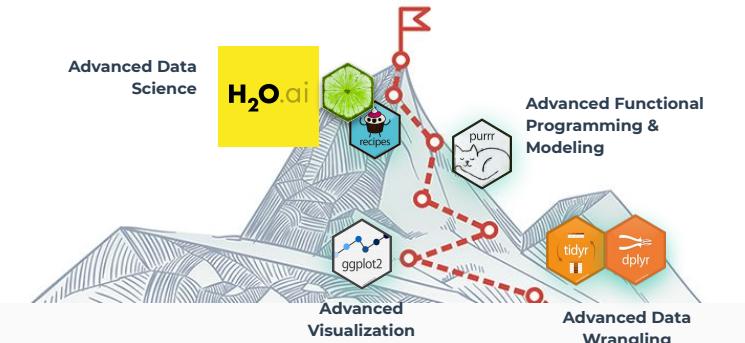
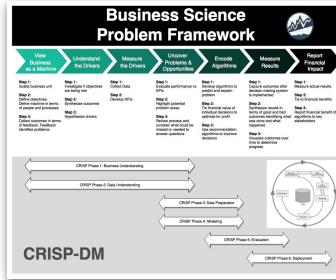
- H2O AutoML - Modeling Churn
- ML Performance
- LIME Feature Explanation

## 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

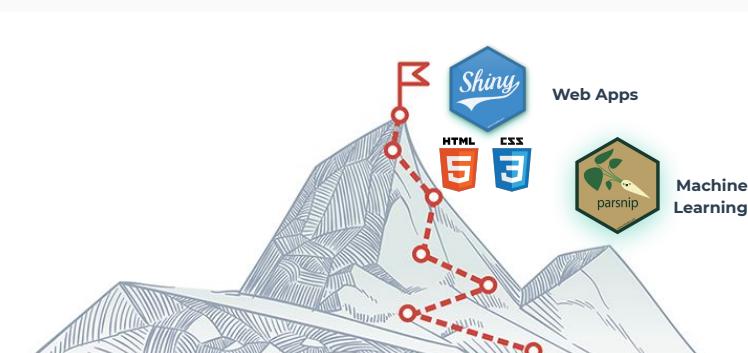
# Shiny Apps for Business (DS4B 102-R)



Web Application Development  
**4 Weeks**

The collage includes:

- A "Data Science with R" course page featuring a "Predictive Pricing App" dashboard.
- A "Flexdashboard Apps" section showing a dashboard with a map of the US and time series plots.
- A "Shiny Apps" section showing a dashboard with a scatter plot and a histogram.
- A "Themes, Dashboards, & Examples" section showing a dashboard with multiple panels and a sidebar.
- A "Business Analytics" section showing a dashboard with a map and a bar chart.
- A "Machine Learning" section showing a dashboard with a scatter plot and a sidebar.
- A "Data Science with R" course page featuring a "Sales Dashboard with Demand Forecasting" dashboard.



The collage includes:

- A "Shiny" logo and a bar chart.
- A "DATA ANALYTICS" section with a large blue "R" icon.
- A "Machine Learning" section with a green gear icon.
- A "Shiny" logo and a bar chart.
- A "DS4B 102-R: Shiny Web Applications for Business (Level 1)" course page.
- A "Build a predictive web application using Shiny, Flexdashboard, and XGBoost" section.
- A photo of Matt Dancho.

# Key Benefits

Frontend + Backend + Production Deployment

## Frontend for Shiny

- Bootstrap

## Backend for Shiny

- MongoDB
- Dynamic UI
- User Authentication
- Store & Write User Data

## Production Deployment

- AWS
- EC2 Server
- VPC Connection
- URL Routing

# Shiny Apps for Business (DS4B 202A-R)



Web Application Development  
**6 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 ~~\$1,500~~

**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

|                                  |  |   |
|----------------------------------|--|---|
| <input type="radio"/>            | <b>Paid Course</b> 15% COUPON DISCOUNT             | \$1,596<br>\$2,356.60                               |
| <input checked="" type="radio"/> | <b>12 Low Monthly Payments</b> 15% COUPON DISCOUNT | 12 payments of \$149/m<br>12 payments of \$126.65/m |

# Begin Learning Today

[university.business-science.io](http://university.business-science.io)

