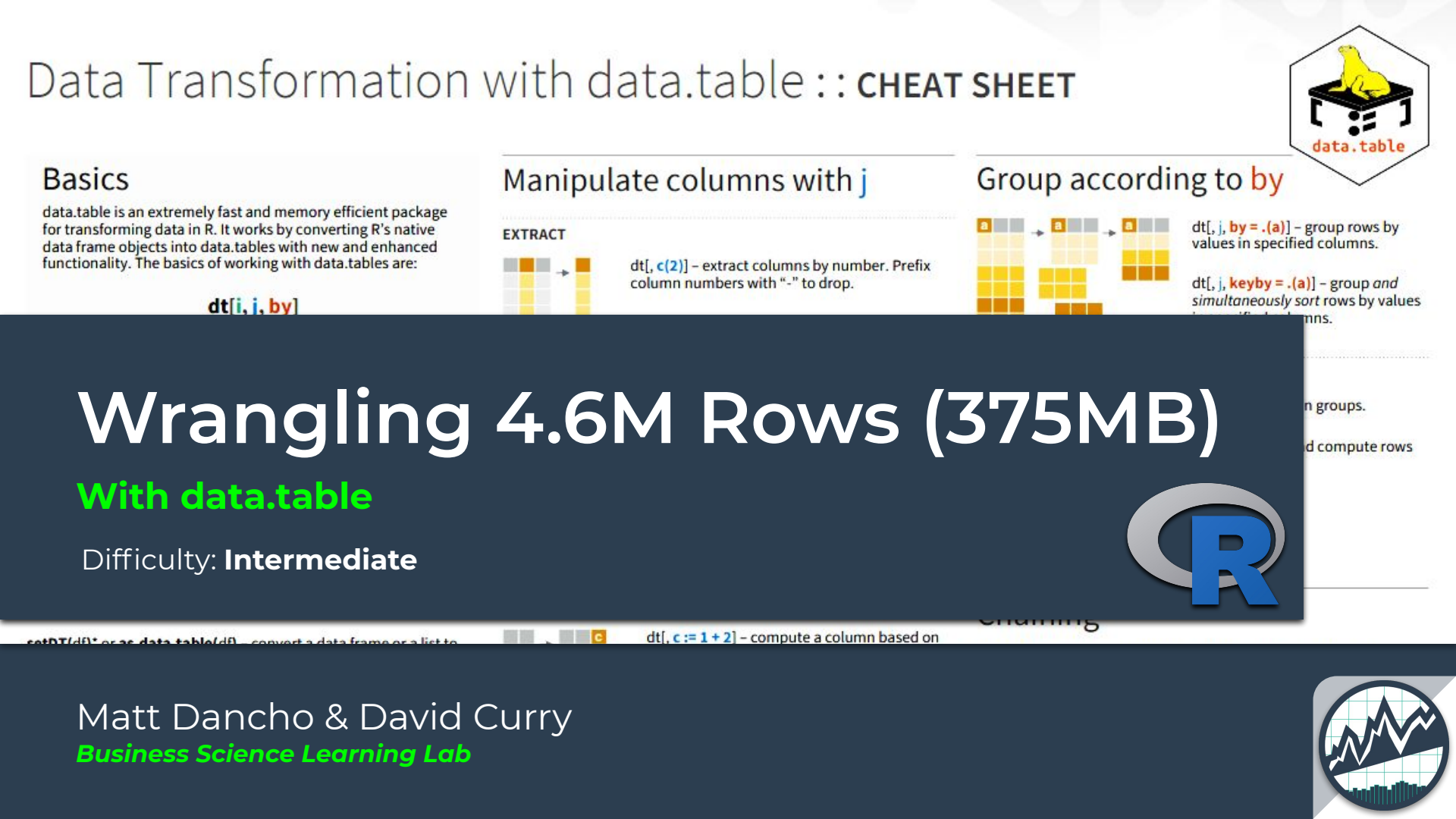


Wrangling 4.6M Rows (375MB)

With data.table

Difficulty: **Intermediate**

Matt Dancho & David Curry

Business Science Learning Lab

Learning Lab Structure

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

Your Hosts!

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

Stephen Lung

- Senior Financial Analyst at Toronto Stock Exchange
- Took DS4B 101-R
- Participated in Tableau Challenge
- Zero working knowledge of Tableau
- **Placed 3rd**
- Beat out peers with 2+ years experience with Tableau
- **Secret Weapon?**

"This is legit a milestone in my development."

Stephen Lung Jun 28th at 9:02 PM

Last week, I entered a Tableau data visualization contest. With zero working knowledge, I had used alot of the DSB101 core concepts to explore the data and even used the DataExplorer package to get a good sense of some interesting parts of the data. What blew me away was how applicable the content I learned was through the DSB courses with @Matt Dancho in a time-sensitive nature of the competition. With this I was able to iterate through multiple concepts to come up with a story for my dashboard. Through all of this, I achieved 3rd place and shockingly, I even had multiple people from the audience mention that I deserved 1st place. For me, the biggest amazement factor was that I was doing more in-depth data analysis (like correlation matrixes) than technical people who had 2 yrs of experience with the Tableau tool. Thanks for your time and stay humble, everyone!

2 2 2 4 2

2 replies

Shreyas 2 days ago

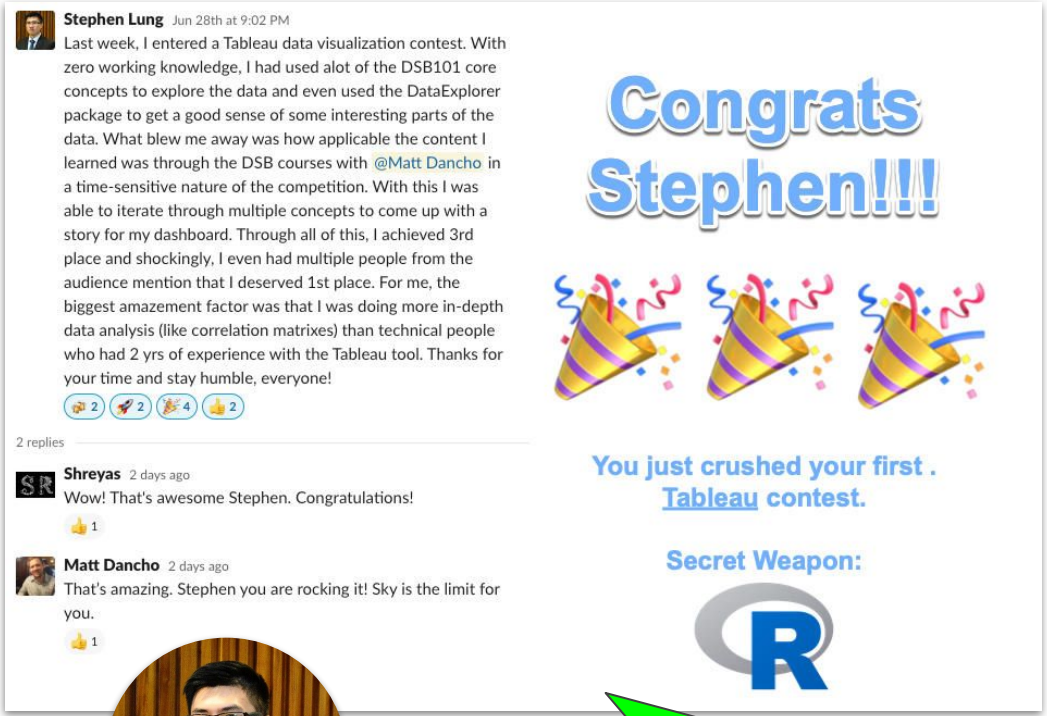
Wow! That's awesome Stephen. Congratulations!

1

Matt Dancho 2 days ago

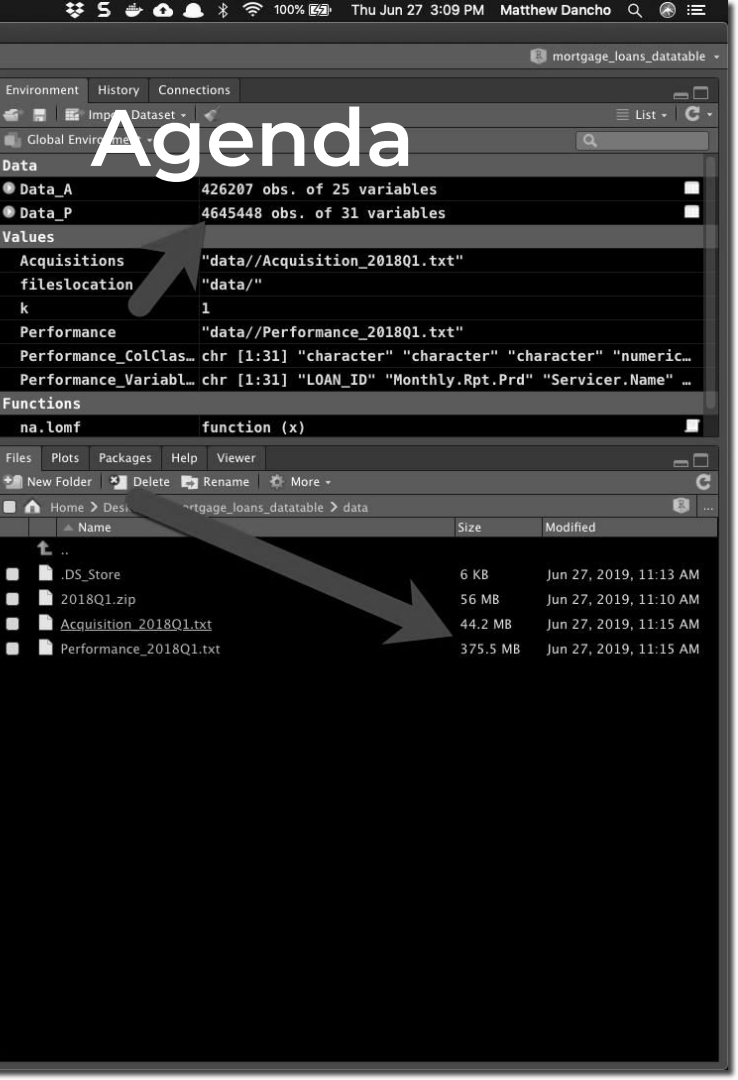
That's amazing. Stephen you are rocking it! Sky is the limit for you.

1



#BusinessScienceSuccess

Agenda



- **Business Case Study**
  - Fannie Mae Home Loan Data
  - 1 Quarter
  - **4.6M Rows (375 MB)**
  - 25GB Total
- **Solution(s)**
  - Tools
  - data.table
- **Resources**
  - Learn FAST
  - DT Basics
- **Demo**
  - Wrangling 4.6M Rows
- **Large Data Strategies**
  - Secret Tactics
  - Learning Plan

Learning Labs PRO

Every 2 Weeks

Get Code

Recordings

Slack Community

$19/month

university.business-science.io

Lab 13
Wrangling 4.6M Rows 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

Business Case Study

Analyze Loan Defaults

Bank Loan Defaults

Business Objectives

Loan defaults cost organizations multi-millions

Need to understand which people or institutions will default on loans

Large Data + Prediction

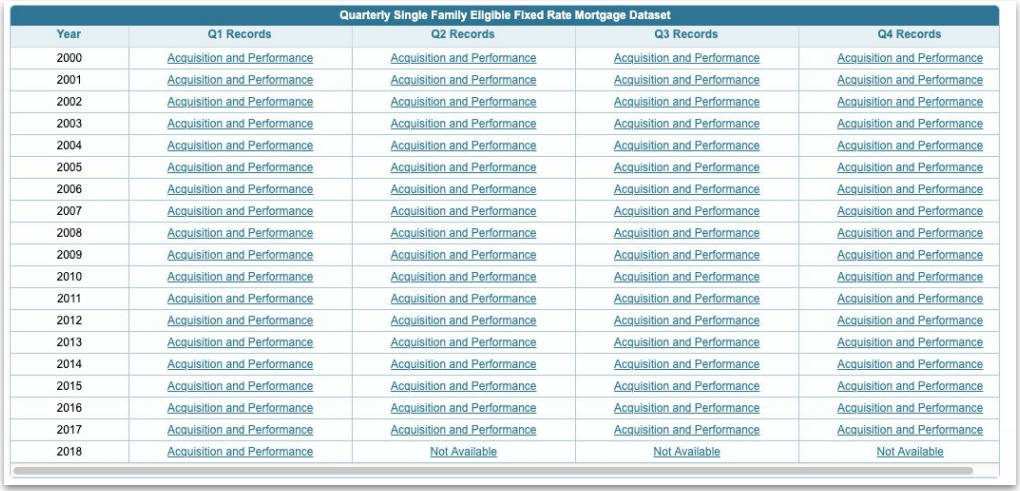
Fannie Mae Loan Data

Loan Acquisition & Performance

Each quarter = 5M Rows of Data

Since 2000 = 25GB Data

How do we analyze this **massive data set**?



Quarterly Single Family Eligible Fixed Rate Mortgage Dataset

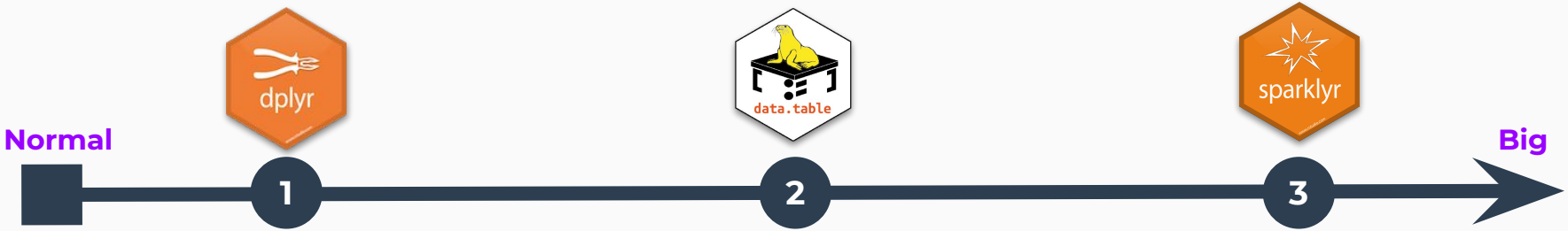
Year	Q1 Records	Q2 Records	Q3 Records	Q4 Records
2000	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2001	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2002	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2003	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2004	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2005	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2006	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2007	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2008	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2009	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2010	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2011	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2012	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2013	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2014	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2015	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2016	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2017	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance	Acquisition and Performance
2018	Acquisition and Performance	Not Available	Not Available	Not Available

https://loanperformancedata.fanniemae.com/lppub/index.html

The Solution(s)

Data Wrangling Tools

by Dataset Size



dplyr

Gets foundations set

(1M Rows+)

data.table

Large Data In-Memory

(10-50M Rows+)

Spark / sparklyr

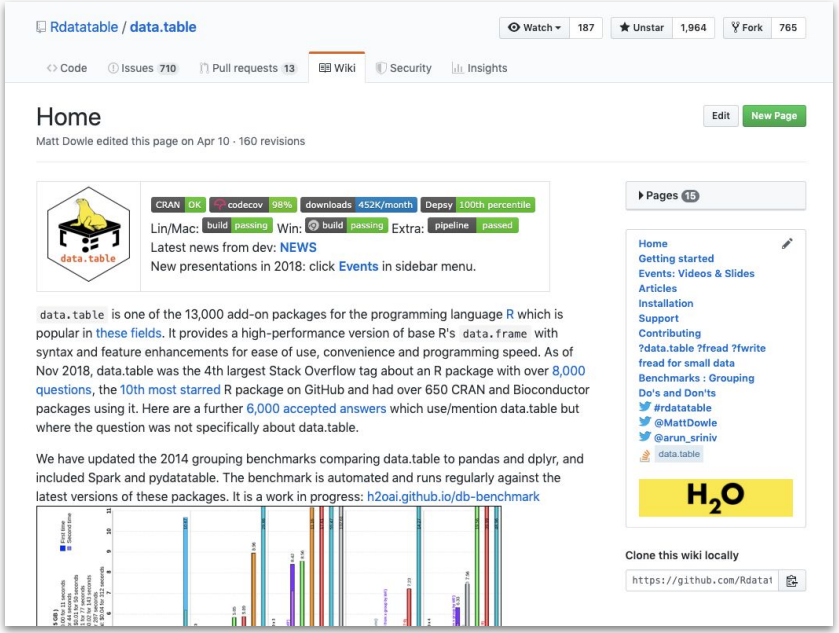
Big Data / Distributed Compute

(100M Rows+)

What Tools Exist?

data.table

High-performance version of base R's data.frame



https://github.com/Rdatatable/data.table/wiki

How does data.table help?

dplyr

Designed for **readability**.

Makes copies through the piping process.

Normally OK.

Large data is **not memory or speed efficient**.

```
199 # data.table
200 tic()
201 combined_data[, gt_3mo_behind_in_1yr := lead(current_loan_delinquency_status, n = 3) >= 1,
202                 by = loan_id]
203 toc()
204
205 combined_data
206
207 # dplyr
208 tic()
209 combined_data %>%
210     group_by(loan_id) %>%
211     mutate(gt_3mo_behind_in_1yr_dplyr = lead(current_loan_delinquency_status, n = 3) >= 1) %>%
212     ungroup() %>%
213     filter(gt_3mo_behind_in_1yr_dplyr)
214 toc()
215
```

How does data.table help?

data.table

Designed for **memory & speed efficient**.

Uses **:=** and **set** functions to **modify inplace (no copies)**

Cons

- Less readable
- Doesn't make copies

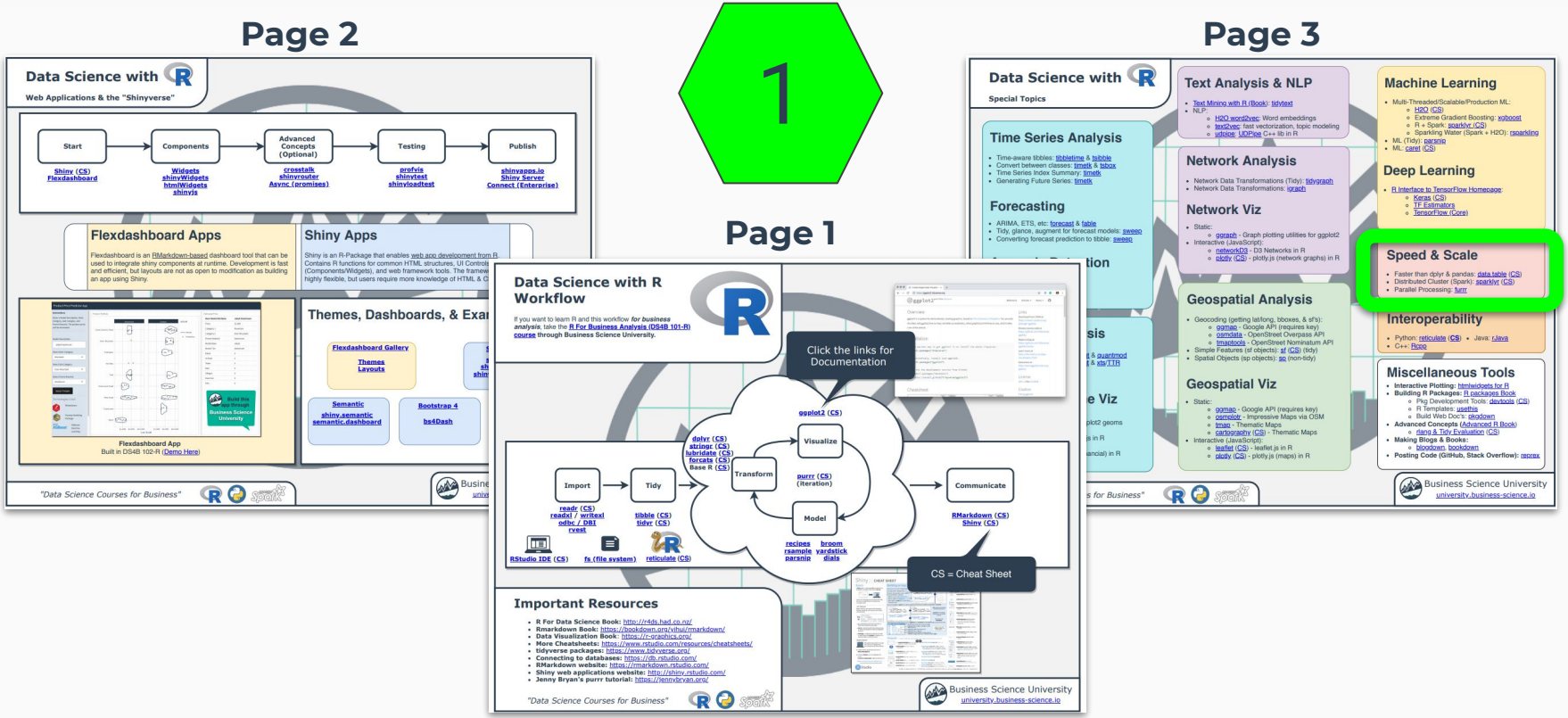
```
> tic()
> combined_data[, gt_3mo_behind_in_1yr := lead(current_loan_delinquency_status, n = 3) >= 1,
+               by = loan_id]
> toc()
4.278 sec elapsed
> tic()
> temp <- combined_data %>%
+     group_by(loan_id) %>%
+     mutate(gt_3mo_behind_in_1yr_dplyr = lead(current_loan_delinquency_status, n = 3) >= 1) %>%
+     ungroup() %>%
+     filter(gt_3mo_behind_in_1yr_dplyr)
> toc()
8.27 sec elapsed
```

https://github.com/Rdatatable/data.table/wiki

Data.Table Resources

Get up to speed FAST

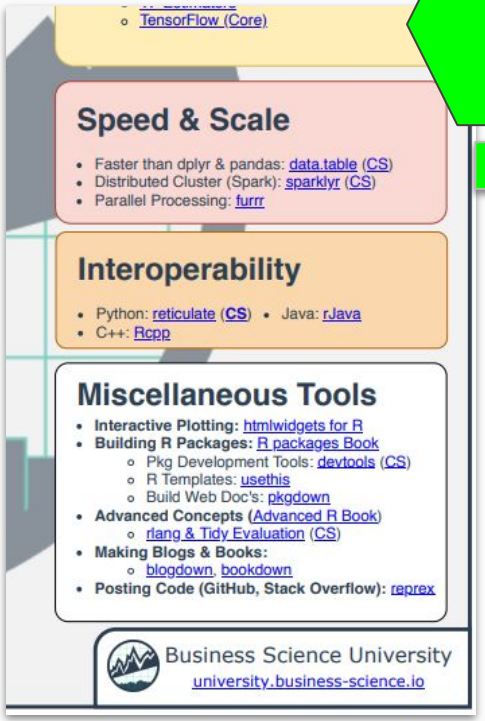
Resource #1: Ultimate R Cheat Sheet



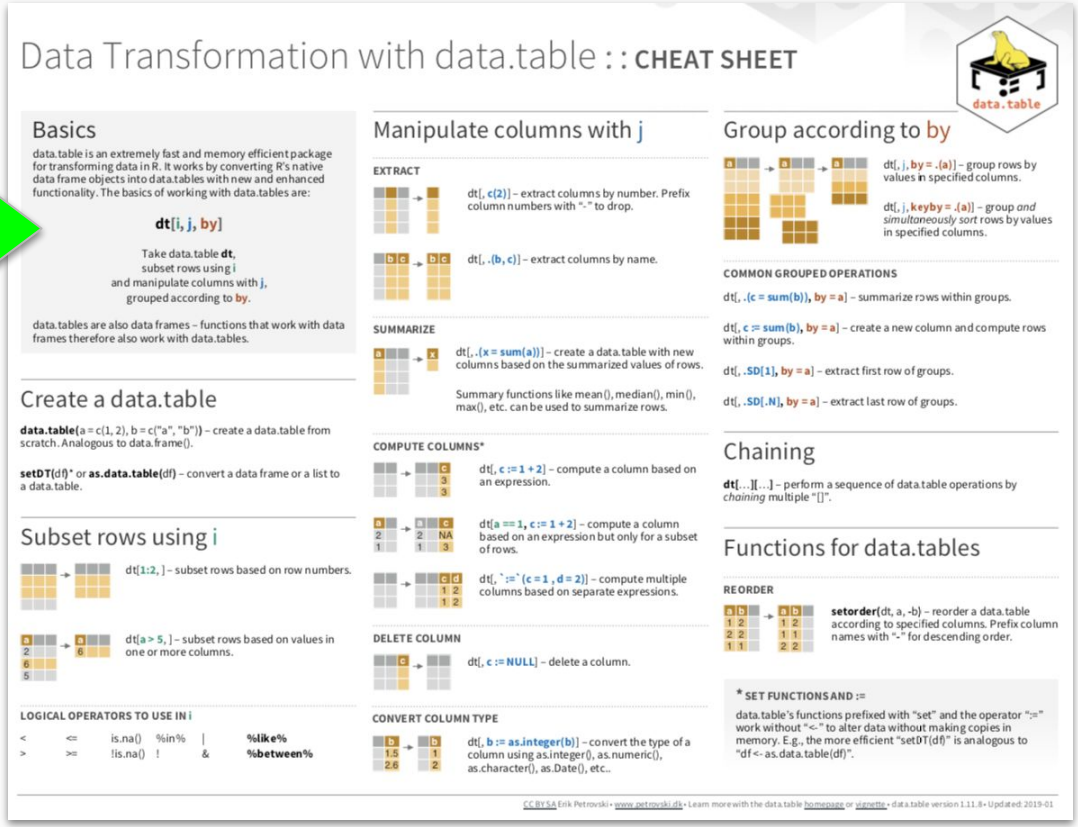
https://www.business-science.io/r-cheatsheet.html

Resource #2: data.table Cheat Sheet

Page 3



2



https://www.business-science.io/r-cheatsheet.html

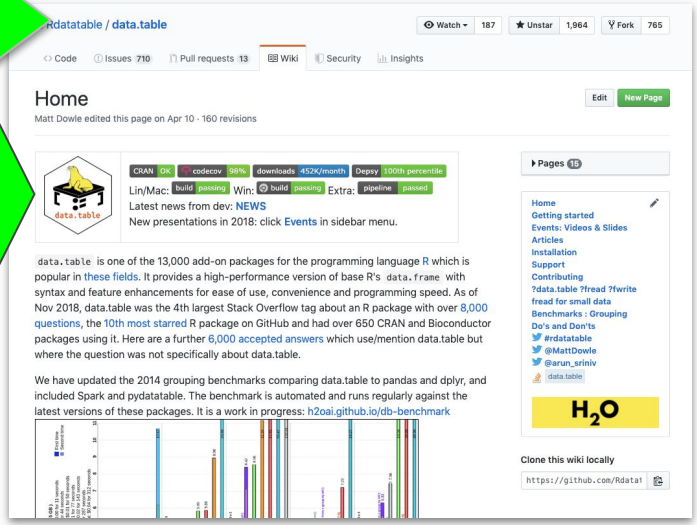
Resource #3: data.table GitHub Wiki

Page 3

Speed & Scale

- Faster than dplyr & pandas: data.table (CS)
- Distributed Cluster (Spark): sparklyr (CS)
- Parallel Processing: furrr

3




```
> require(data.table)
> example(data.table)

# basic row subset
DT[2]                                   # 2nd row
DT[2:3]                                 # 2nd and 3rd row
w=2:3; DT[w]                            # same
DT[order(x)]                            # no need for DT$ prefix on column x
DT[order(x), ]                          # same; the ',' is optional
DT[y>2]                                 # all rows where DT$y > 2
DT[y>2 & v>5]                           # compound logical expressions
DT[!2:4]                                # all rows other than 2:4
DT[-(2:4)]                              # same

# select|compute columns
DT[, v]                                 # v column (as vector)
DT[, list(v)]                           # v column (as data.table)
DT[, .(v)]                              # same; .() is an alias for list()
DT[, sum(v)]                            # sum of column v, returned as vector
DT[, .(sum(v))]                         # same but return data.table
DT[, .(sv=sum(v))]                      # same but name column "sv"
DT[, .(v, v*2)]                         # return two column data.table

# subset rows and select|compute
DT[2:3, sum(v)]                         # sum(v) over rows 2 and 3
DT[2:3, .(sum(v))]                      # same, but return data.table
DT[2:3, .(sv=sum(v))]                   # same, but name column "sv"
DT[2:5, cat(v, "\n")]                   # just for j's side effect

# select columns the data.frame way
DT[, 2]                                 # 2nd column, a data.table always
colNum = 2
DT[, ..colNum]                          # same as DT[,2]; ..var => one-up
DT[["v"]]                               # same as DT[,v] but lower overhead

# grouping operations - j and by
DT[, sum(v), by=x]                      # appearance order of groups preserved
DT[, sum(v), keyby=x]                   # order the result by group
DT[, sum(v), by=x][order(x)]            # same by chaining expressions together
```

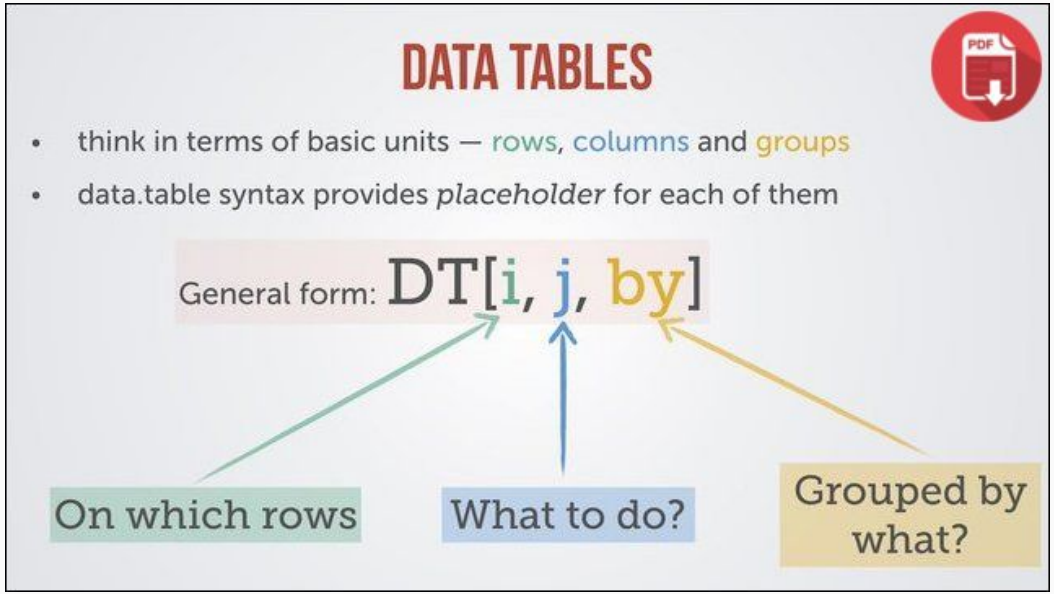
https://www.business-science.io/r-cheatsheet.html

Data.Table Basics

80/20 Concepts & Important Operations

Critical Concept #1

Learn this:



https://github.com/Rdatatable/data.table/wiki

Critical Concept #2

Understand this:

Modifying In-Place

How?
:=

Why?
No Copies (Speed boost)

Example

```
DT[, unpaid_flag  := unpaid_bal >= 1]
```

https://github.com/Rdatatable/data.table/wiki

Row Operations - Filtering & Arranging

```
# basic row subset
DT[2]                                   # 2nd row
DT[2:3]                                 # 2nd and 3rd row
w=2:3; DT[w]                            # same
DT[order(x)]                            # no need for DT$ prefix on column x
DT[order(x), ]                          # same; the ',' is optional
DT[y>2]                                 # all rows where DT$y > 2
DT[y>2 & v>5]                           # compound logical expressions
DT[!2:4]                                # all rows other than 2:4
DT[-(2:4)]                              # same
```

Similar to dplyr functions

filter()

arrange()

Column Operations - Selecting & Summarizing

```
# select|compute columns
DT[, v]                                  # v column (as vector)
DT[, list(v)]                            # v column (as data.table)
DT[, .(v)]                               # same; .() is an alias for list()
DT[, sum(v)]                             # sum of column v, returned as vector
DT[, .(sum(v))]                          # same but return data.table
DT[, .(sv=sum(v))]                       # same but name column "sv"
DT[, .(v, v*2)]                          # return two column data.table
```

Similar to dplyr functions

select()

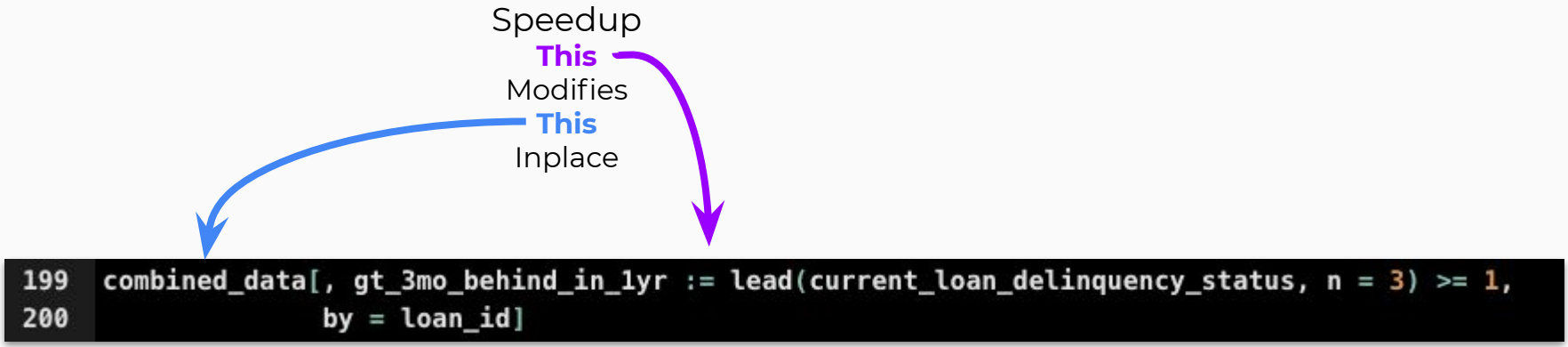
summarize()

Grouping Operations - Grouping & Summarizing

```
# grouping operations - j and by
DT[, sum(v), by=x]                          # appearance order of groups preserved
DT[, sum(v), keyby=x]                       # order the result by group
DT[, sum(v), by=x][order(x)]                # same by chaining expressions together
```

Similar to dplyr functions
group_by() + summarize()

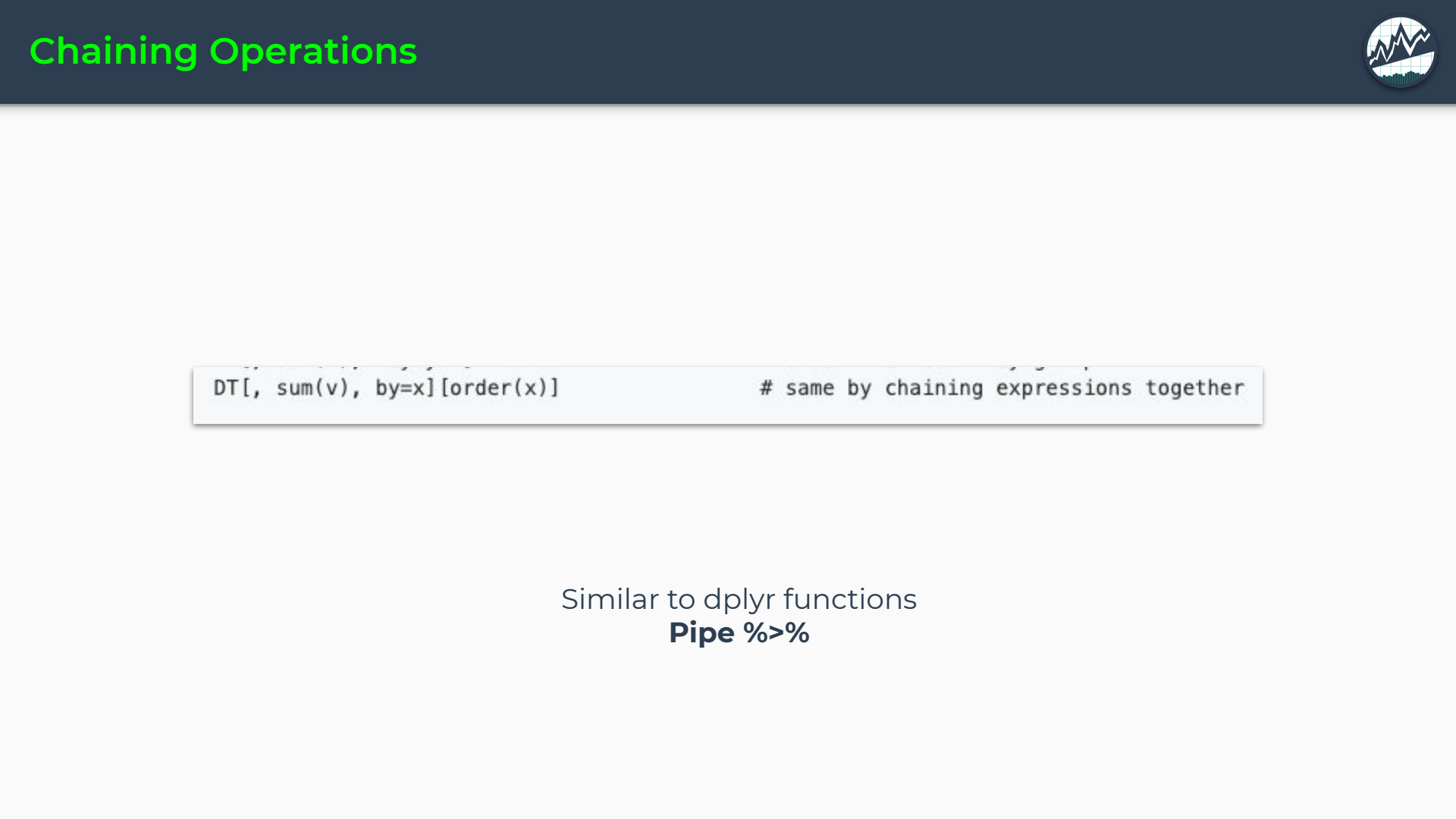
Grouping Operations - Grouping & Mutating




```
199  combined_data[, gt_3mo_behind_in_1yr := lead(current_loan_delinquency_status, n = 3) >= 1,
200                by = loan_id]
```

Similar to dplyr functions
group_by() + mutate()

Chaining Operations



```
DT[, sum(v), by=x][order(x)]                    # same by chaining expressions together
```

Similar to dplyr functions

Pipe %>%

Joining Operations

```
# joins as subsets
X = data.table(x=c("c","b"), v=8:7, foo=c(4,2))
X

DT[X, on="x"]                                  # right join
X[DT, on="x"]                                  # left join
DT[X, on="x", nomatch=0]                       # inner join
DT[!X, on="x"]                                 # not join
DT[X, on=c(y="v")]                             # join DT$y to X$v
DT[X, on="y==v"]                               # same
```

Similar to dplyr functions

left_join()
right_join()
etc

Special Symbols

```
# more on special symbols, see also ?"special-symbols"
DT[.N]                                    # last row
DT[, .N]                                  # total number of rows in DT
DT[, .N, by=x]                            # number of rows in each group
DT[, .SD, .SDcols=x:y]                    # select columns 'x' and 'y'
DT[, .SD[1]]                              # first row; same as DT[1,]
DT[, .SD[1], by=x]                        # first row of each group
DT[, c(.N, lapply(.SD, sum)), by=x]       # group size alongside sum
DT[, .I[1], by=x]                         # row number of first row of each group
DT[, grp := .GRP, by=x]                   # add a group counter column
X[, DT[.BY, y, on="x"], by=x]             # join within group to use less ram
```

Similar to dplyr functions

.N == n()

.SD == slice()

Demo

4.6M rows / 375 MB Data

data.table in action

Large Data Strategy

Secret Tactics

Big Data

Impacts Every Company

Data storage is increasing exponentially.

How can we deal with it?

Trick to Solving Big Data Problems. Make them small.

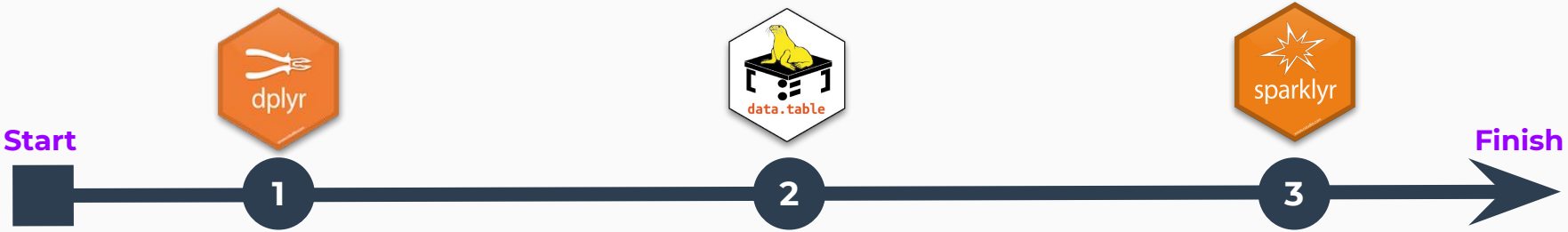
Large datasets can be **sampled**.

Sampling makes data **manageable**.

Good sampling strategy: Loss in ML accuracy is typically low.

Upgrade to **Big Data Tools** once you have a good methodology.

Big Data Learning Plan



Learn **dplyr**

Gets foundations set

(1M Rows+)

Learn **data.table**

Large Data In-Memory

(10-50M Rows+)

Learn **Spark / sparklyr**

Adds Big Data

(100M Rows+)

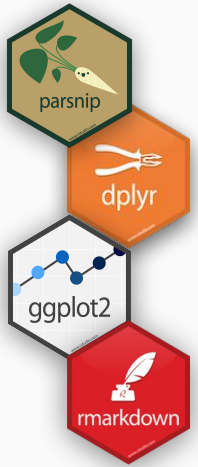
Data Wrangling **Foundations** Are The Key

Big Data Learning Plan

Start with Foundations

35 Hours of Video Lessons

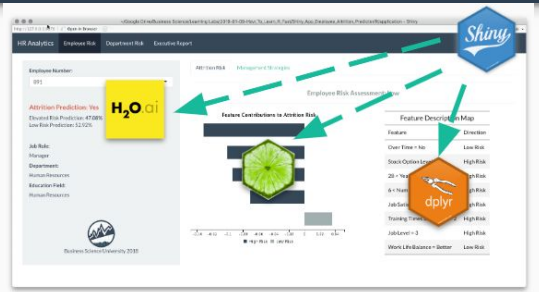
- Machine Learning (parsnip)
- **Data Manipulation (dplyr)**
- Visualization (ggplot2)
- Reporting (rmarkdown)
- More packages



Big Data Learning Plan

YOUR Transformation

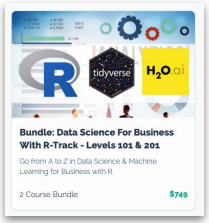
Do Business Projects
Climb the Hill



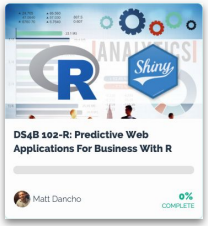
Build Production-Ready
Web Apps

Complete 1-Hour Courses
Domain Analysis & Tool Courses

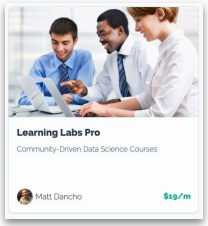
Start → 1 → 2 → 3 → **Finish**



Analysis Courses



App Development
Courses



Learning Labs PRO

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

3-Course R-Track System

Business Analysis with R
(DS4B 101-R)

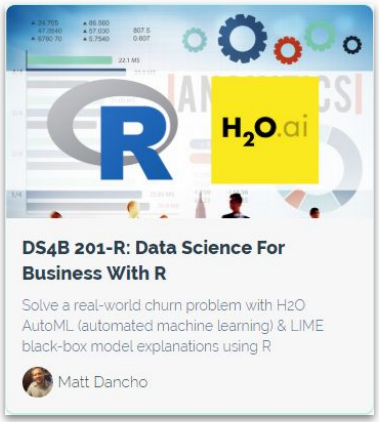
Data Science For Business with R
(DS4B 201-R)

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

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

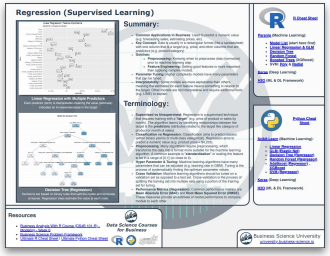
- **Machine Learning - Week 6 (8 hours of Video Lessons)**
  - **Clustering (3 hours)**
  - **Regression (5 hours)**
  - **2 Challenges**

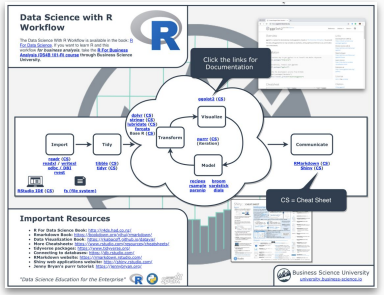
- Learn Business Reporting - Week 7
  - RMarkdown & plotly
  - 2 Project Reports:
    1. Product Pricing Algo
    2. Customer Segmentation

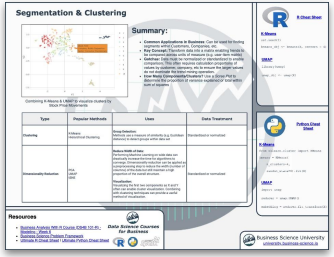
Business Analysis with R (DS4B 101-R)

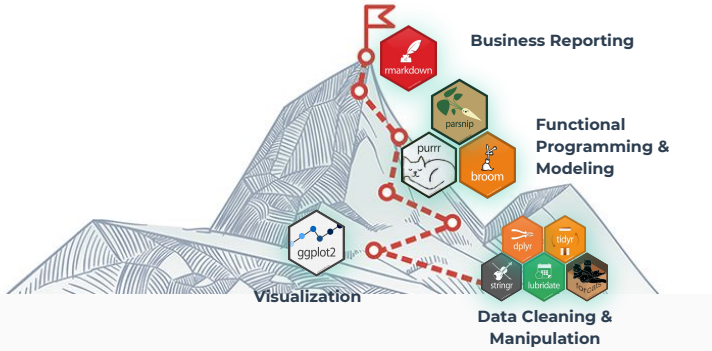
Data Science Foundations

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**

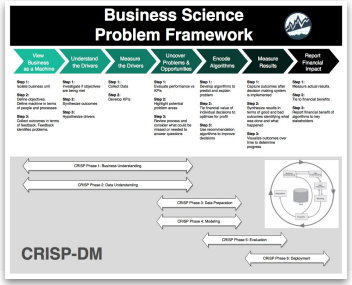
Return-On-Investment - Weeks 7, 8, 9

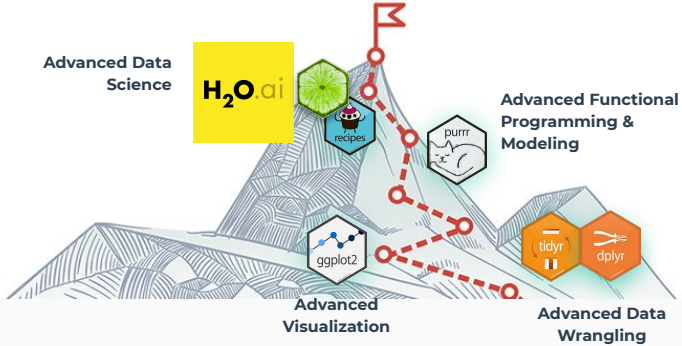
- Expected Value Framework
- Threshold Optimization
- Sensitivity Analysis
- Recommendation Algorithm

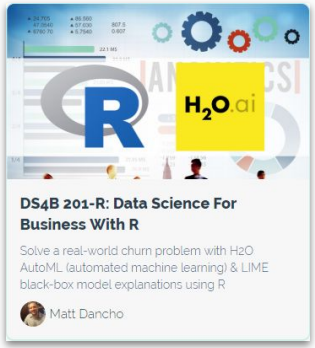
Data Science For Business (DS4B 201-R)

Machine Learning & Business Consulting

10 Weeks







Key Benefits

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

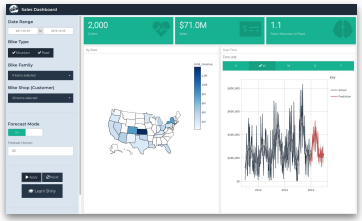
App #1: Predictive Pricing App
- Model Product Portfolio
- XGBoost Pricing Prediction
- Generate new products instantly

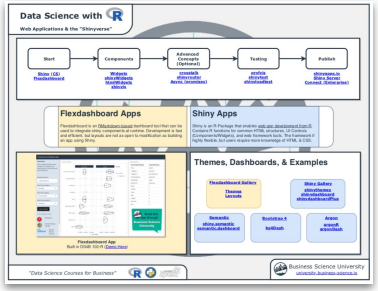
App #2: Sales Dashboard with Demand Forecasting
- Model Demand History
- Segment Forecasts by Product & Customer
- XGBoost Time Series Forecast
- Generate new forecasts instantly

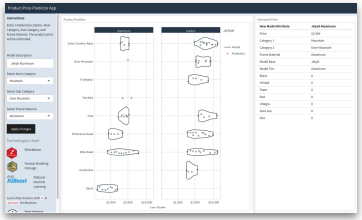
Shiny Apps for Business (DS4B 102-R)

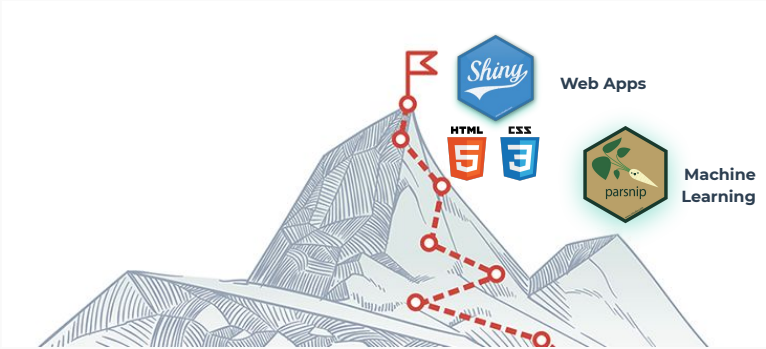
Web Application Development

4 Weeks





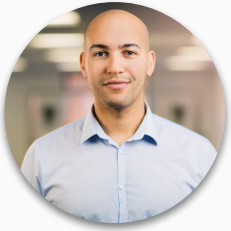




Testimonials

*“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**

PROMO Code: learninglabs

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 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

R-TRACK BUNDLE

MSRP: ~~$234/mo~~

6 Low Monthly Payments $199/mo

Save: $35/mo

Begin Learning Today

university.business-science.io