

Data Transformation with data.table :: CHEAT SHEET



Basics

data.table is an extremely fast and memory efficient package for transforming data in R. It works by converting R's native data frame objects into data.tables with new and enhanced functionality. The basics of working with data.tables are:

`dt[i, j, by]`

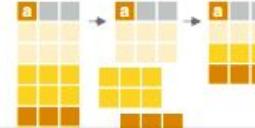
Manipulate columns with `j`

EXTRACT



`dt[, c(2)]` – extract columns by number. Prefix column numbers with “-” to drop.

Group according to `by`



`dt[, j, by = .(a)]` – group rows by values in specified columns.

`dt[, j, keyby = .(a)]` – group and simultaneously sort rows by values in specified columns.

Wrangling 4.6M Rows (375MB)

With data.table

Difficulty: **Intermediate**



Wrangling

`setDT(df*) or as.data.table(df)` – convert a data frame or a list to



`dt[, c := 1 + 2]` – compute a column based on

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 a lot 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 replies

 **Shreyas** 2 days ago

Wow! That's awesome Stephen. Congratulations!





Matt Dancho 2 days ago

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



**Congrats
Stephen!!!**



You just crushed your first .
Tableau contest.

Secret Weapon:



#BusinessScienceSuccess

The screenshot shows the RStudio interface with the following details:

- Environment** tab: Shows variables `Data_A` (426207 obs. of 25 variables) and `Data_P` (4645448 obs. of 31 variables).
- Global Environment** tab: Shows variables `Acquisitions`, `fileslocation`, `k`, `Performance`, `Performance_ColClas...`, and `Performance_Variab...`.
- Data** tab: Shows a code block defining `na.lomf` as a function.
- Files** tab: Shows a file tree under `mortgage_loans_datatable > data` with files: `..`, `.DS_Store`, `2018Q1.zip`, `Acquisition_2018Q1.txt`, and `Performance_2018Q1.txt`. The `Acquisition_2018Q1.txt` file is selected.

A large arrow points from the `Acquisition_2018Q1.txt` file in the file tree to the `Values` section in the **Data** tab, highlighting the connection between the dataset and its source file.

- **Business Case Study**

- Fannie Mae Home Loan Data
- 1 Quarter
- **4.6M Rows (375 MB)**
- 25GB Total

- **Demo**

- Wrangling 4.6M Rows

- **Large Data Strategies**

- Secret Tactics
- Learning Plan

- **Solution(s)**

- Tools
- `data.table`

- **Resources**

- Learn FAST
- DT Basics



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**

Learning Labs Pro

Community-Driven Data Science Courses

 Matt Dancho

\$19/m

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

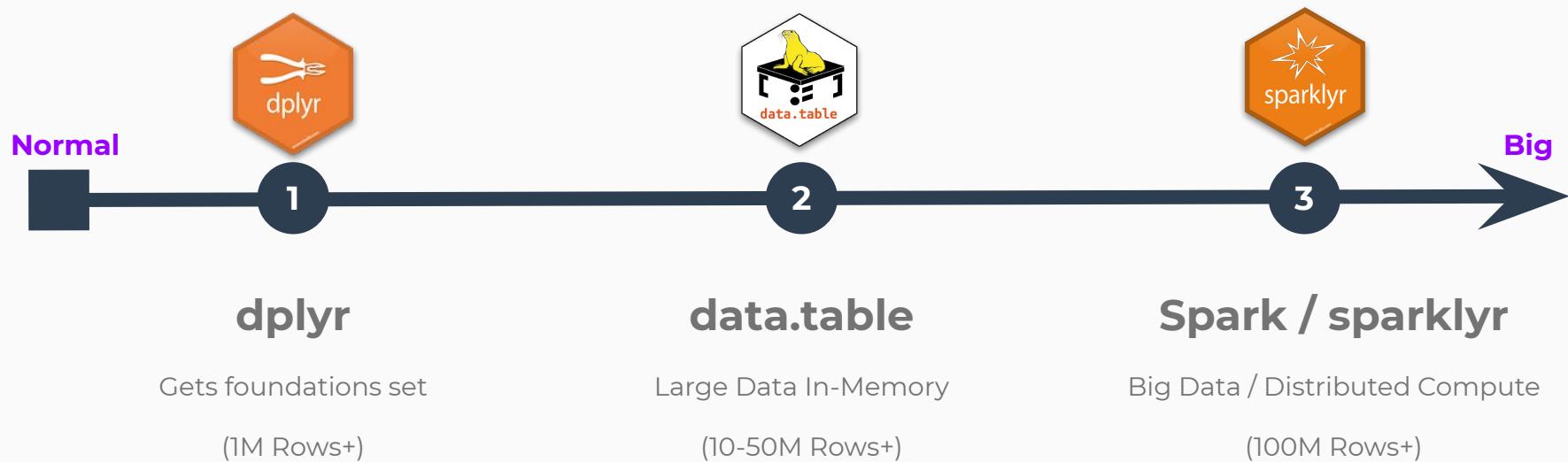
How do we analyze this **massive data set?**

Year	Quarterly Single Family Eligible Fixed Rate Mortgage Dataset			
	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

The Solution(s)

Data Wrangling Tools

by Dataset Size





What Tools Exist?

data.table

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

[Rdatatable / data.table](https://github.com/Rdatatable/data.table/wiki)

Watch 187 Unstar 1,964 Fork 765

Code Issues 710 Pull requests 13 Wiki Security Insights

Home

Matt Dowle edited this page on Apr 10 · 160 revisions

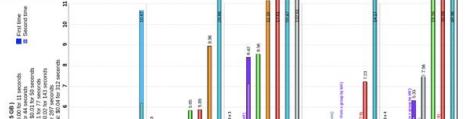
Edit New Page

 CRAN OK codecov 98% downloads 452K/month Depy 100th percentile Lin/Mac: build passing Win: build passing Extra: pipeline passed

Latest news from dev: NEWS
New presentations in 2018: click Events in sidebar menu.

data.table is one of the 13,000 add-on packages for the programming language R which is popular in [these fields](#). It provides a high-performance version of base R's `data.frame` with syntax and feature enhancements for ease of use, convenience and programming speed. As of Nov 2018, data.table was the 4th largest Stack Overflow tag about an R package with over 8,000 questions, the 10th most starred R package on GitHub and had over 650 CRAN and Bioconductor packages using it. Here are a further 6,000 accepted answers which use/mention data.table but where the question was not specifically about data.table.

We have updated the 2014 grouping benchmarks comparing data.table to pandas and dplyr, and included Spark and pydata.table. The benchmark is automated and runs regularly against the latest versions of these packages. It is a work in progress: h2oai.github.io/db-benchmark



Pages 15

Home Getting started Events: Videos & Slides Articles Installation Support Contributing ?data.table ?read ?fwrite fread for small data Benchmarks : Grouping Do's and Don'ts #rdatable @MattDowle @arun_sriniv data.table

H₂O

Clone this wiki locally

<https://github.com/Rdatatable/data.table>



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_lyr := 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_lyr_dplyr = lead(current_loan_delinquency_status, n = 3) >= 1) %>%
212     ungroup() %>%
213     filter(gt_3mo_behind_in_lyr_dplyr) |
214     toc()
215
216
```



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_lyr := 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_lyr_dplyr = lead(current_loan_delinquency_status, n = 3) >= 1) %>%
+   ungroup() %>%
+   filter(gt_3mo_behind_in_lyr_dplyr)
> toc()
8.27 sec elapsed
```

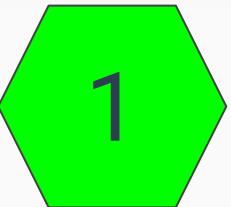
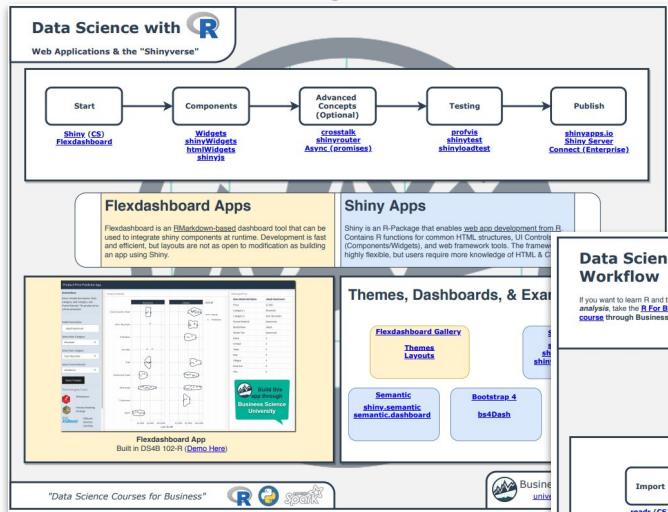
Data.Table Resources

Get up to speed FAST

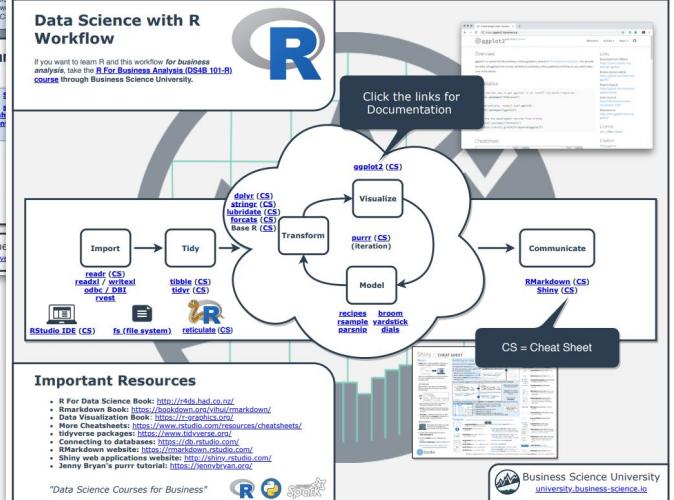


Resource #1: Ultimate R Cheat Sheet

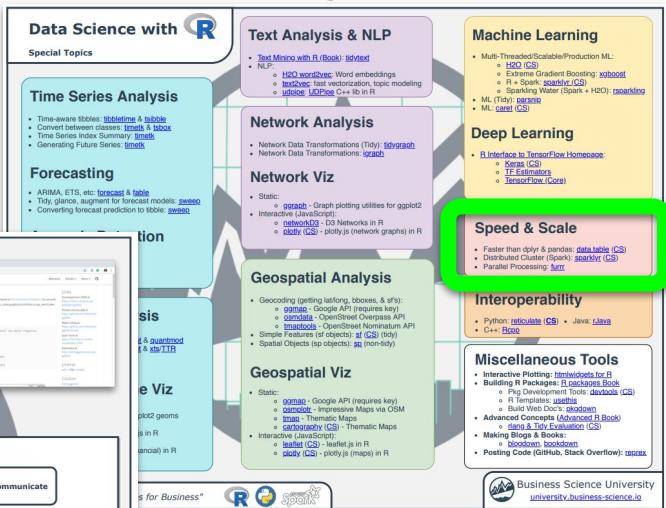
Page 2



Page 1



Page 3





Resource #2: data.table Cheat Sheet

Page 3

2

Speed & Scale

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

Interoperability

- Python: [reticulate \(CS\)](#)
- Java: [rJava](#)
- C++: [Rcpp](#)

Miscellaneous Tools

- Interactive Plotting: [htmlwidgets for R](#)
- Building R Packages: [R packages Book](#)
 - Pkg Development Tools: [devtools \(CS\)](#)
 - R Templates: [usethis](#)
 - Build Web Doc's: [pkdown](#)
- Advanced Concepts ([Advanced R Book](#))
 - [rlang & Tidy Evaluation \(CS\)](#)
- Making Blogs & Books:
 - [blogdown, bookdown](#)
- Posting Code (GitHub, Stack Overflow): [reprex](#)

 Business Science University
university.business-science.io

Data Transformation with data.table :: CHEAT SHEET

Basics

data.table is an extremely fast and memory efficient package for transforming data in R. It works by converting R's native data frame objects into data.tables with new and enhanced functionality. The basics of working with data.tables are:

dt[i, j, by]

Take data.table **dt**,
subset rows using **i**,
and manipulate columns with **j**,
grouped according to **by**.

data.tables are also data frames - functions that work with data frames therefore also work with data.tables.

Create a data.table

`data.table(a = c(1, 2), b = c("a", "b"))` – create a data.table from scratch. Analogous to data.frame().

`setDT(df)*` or `as.data.table(df)` – convert a data frame or a list to a data.table.

Subset rows using i

 `dt[1:2,]` – subset rows based on row numbers.

 `dt[a > 5,]` – subset rows based on values in one or more columns.

LOGICAL OPERATORS TO USE IN i

<	=	<code>is.na()</code>	<code>%in%</code>	<code> </code>	<code>%like%</code>
>	=	<code>is.na() !=</code>		<code>&</code>	<code>%between%</code>

Manipulate columns with j

EXTRACT

 `dt[, c(2)]` – extract columns by number. Prefix column numbers with `"."` to drop.

 `dt[, .(b, c)]` – extract columns by name.

SUMMARIZE

 `dt[, .(x = sum(a))]` – create a data.table with new columns based on the summarized values of rows.

Summary functions like `mean()`, `median()`, `min()`, `max()`, etc. can be used to summarize rows.

COMPUTE COLUMNS*

 `dt[, c := 1 + 2]` – compute a column based on an expression.

 `dt[, c == 1, c := 1 + 2]` – compute a column based on an expression but only for a subset of rows.

 `dt[, . := (c = 1, d = 2)]` – compute multiple columns based on separate expressions.

DELETE COLUMN

 `dt[, c := NULL]` – delete a column.

CONVERT COLUMN TYPE

 `dt[, b := as.integer(b)]` – convert the type of a column using `as.integer()`, `as.numeric()`, `as.character()`, `as.Date()`, etc.

Group according to by

 `dt[, j, by = .(a)]` – group rows by values in specified columns.

COMMON GROUPED OPERATIONS

`dt[, .c := sum(b), by = a]` – summarize rows within groups.

`dt[, .SD[1], by = a]` – extract first row of groups.

`dt[, .SD[N], by = a]` – extract last row of groups.

Chaining

`dt[, ...][...]` – perform a sequence of data.table operations by chaining multiple "[]".

Functions for data.tables

REORDER

 `setorder(dt, a, -b)` – reorder a data.table according to specified columns. Prefix column names with `"-"` for descending order.

* SET FUNCTIONS AND :=

data.table's functions prefixed with "set" and the operator "`:=`" work without "`<-`" to alter data without making copies in memory. E.g., the more efficient `setDT(df)*` is analogous to `df <- as.data.table(df)`.



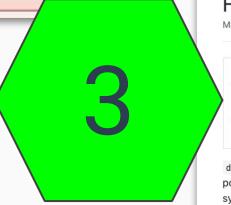
Resource #3: data.table GitHub Wiki



Page 3

Speed & Scale

- Faster than dplyr & pandas: [data.table \(CS\)](#)
- Distributed Cluster (Spark): [sparklyr \(CS\)](#)
- Parallel Processing: [furr](#)



 / data.table

 / data.table

 / data.table

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

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

# basic row subset
DT[2]
DT[2:3]
w=2:3; DT[w]
DT[order(x)]
DT[order(x), ]
DT[y>2]
DT[y>2 & v>5]
DT[1:24]
DT[-(2:4)]

# select|compute columns
DT[, v]
DT[, list(v)]
DT[, .(v)]
DT[, sum(v)]
DT[, c(sum(v))]
DT[, .(sv=sum(v))]
DT[, .(v, v*2)]

# v column (as vector)
# v column (as data.table)
# same; .() is an alias for list()
# sum of column v, returned as vector
# same but return data.table
# same but name column "sv"
# return two column data.table

# subset rows and select|compute
DT[2:3, sum(v)]
DT[2:3, .(sum(v))]
DT[2:3, .(sv=sum(v))]
DT[2:5, cat(v, "\n")]

# select columns the data.frame way
DT[, 2]
colNum = 2
DT[, ...colNum]
DT[["v"]]

# grouping operations - j and by
DT[, sum(v), by=x]
DT[, sum(v), keyby=x]
DT[, sum(v), by=x][order(x)]

# appearance order of groups preserved
# order the result by group
# same by chaining expressions together
```

Data.Table Basics

80/20 Concepts & Important
Operations



Critical Concept #1

Learn this:

DATA TABLES

The diagram shows the general form of the `data.table` syntax: **DT[i, j, by]**. Three arrows point from text boxes below to the respective parts of the formula:

- A green arrow points from the box "On which rows" to the `i` in `DT[i, j, by]`.
- A blue arrow points from the box "What to do?" to the `j` in `DT[i, j, by]`.
- An orange arrow points from the box "Grouped by what?" to the `by` in `DT[i, j, by]`.

General form: **DT[i, j, by]**

On which rows What to do? Grouped by what?



Critical Concept #2

Understand this:

Modifying In-Place

How?

`:=`

Why?

No Copies (Speed boost)

Example

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

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

Speedup

This

Modifies

This

Inplace

```
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**.

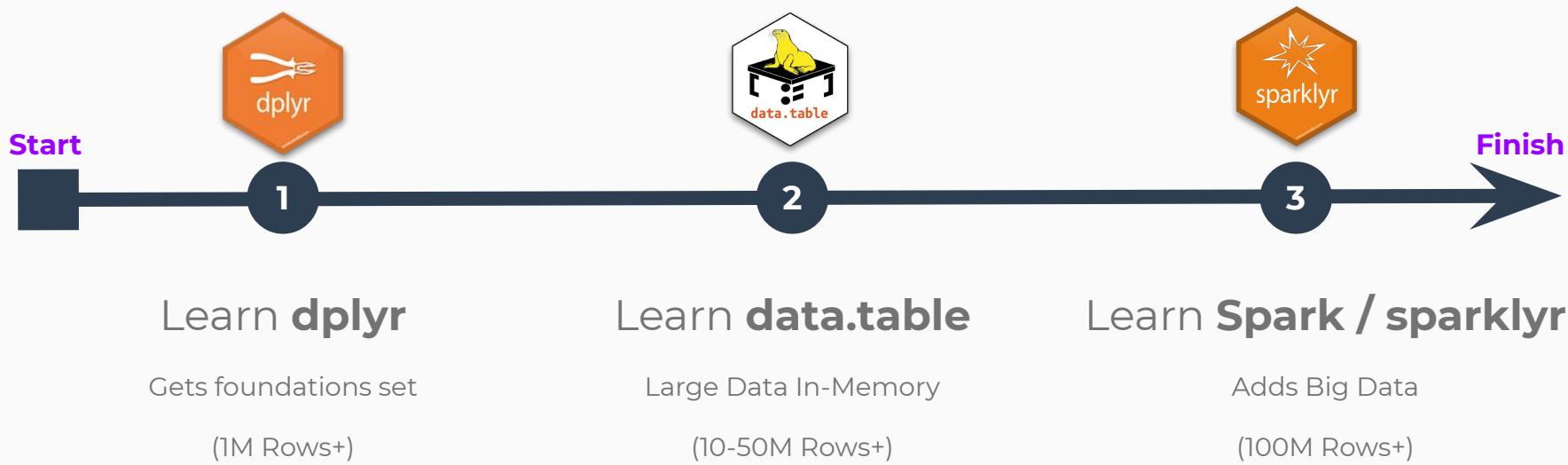
Sampling makes data **manageable**.

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



Data Wrangling **Foundations** Are The Key

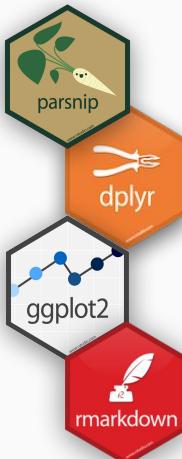


Big Data Learning Plan

Start with Foundations

35 Hours of Video Lessons

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



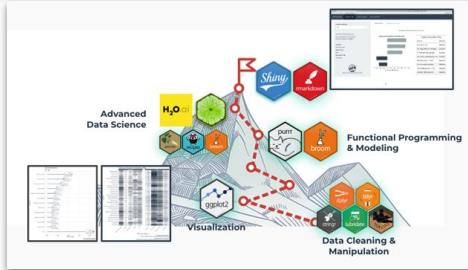
DS4B 101-R: Business Analysis With R

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

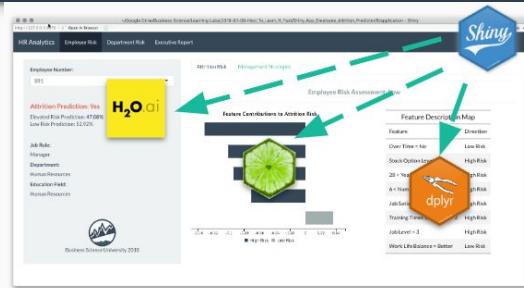
 Matt Dancho

Big Data Learning Plan

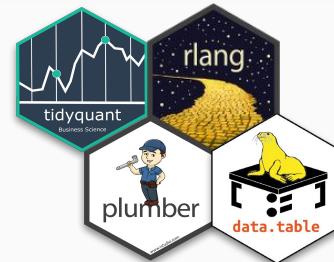
YOUR Transformation



Do Business Projects
Climb the Hill



Build Production-Ready
Web Apps



Complete 1-Hour Courses
Domain Analysis & Tool Courses

Start

1



Analysis Courses

2



App Development
Courses

Finish

3



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



DS4B 101-R: Business Analysis With R

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



Machine Learning & Business Consulting
10 Weeks



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



Web Application Development
4 Weeks



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

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

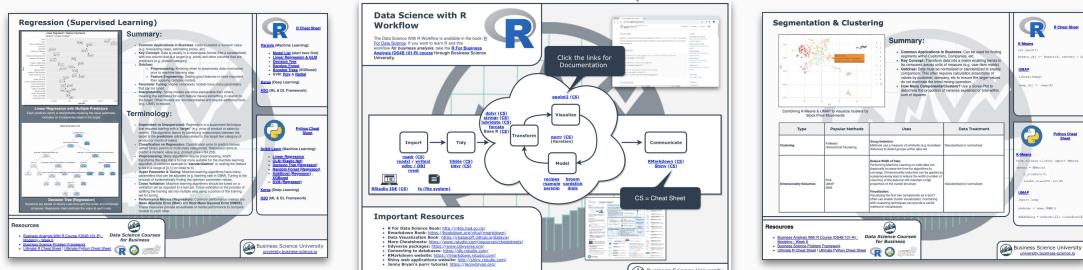


Key Benefits

- 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

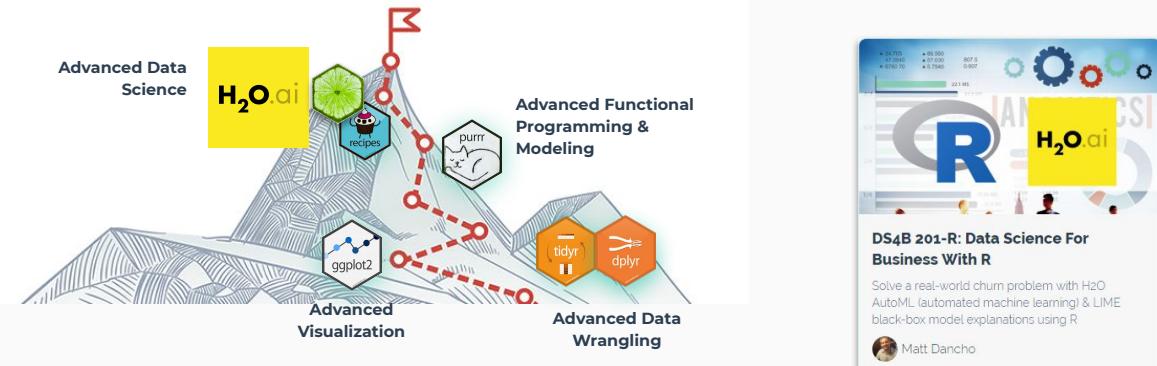
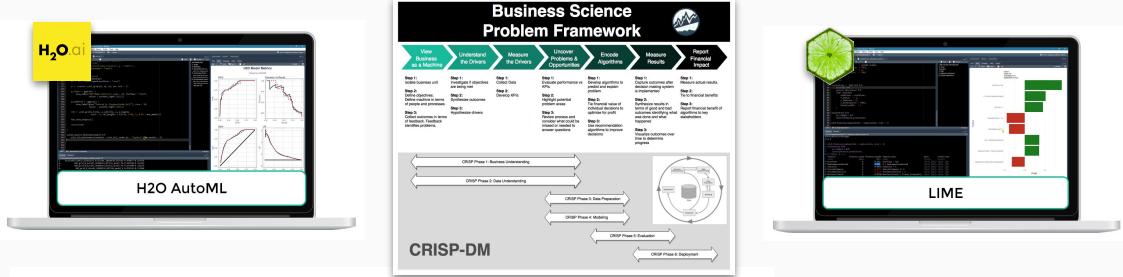
Return-On-Investment - Weeks 7, 8, 9

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

Data Science For Business

(DS4B 201-R)

Machine Learning & Business Consulting
10 Weeks



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

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.



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**

The interface features a large blue 'R' logo on the left. To its right are three hexagonal icons: a yellow one labeled 'H2O.ai', a black one labeled 'tidyverse', and a blue one labeled 'Shiny'. Below these icons is a section titled 'DS4B 101-R: Data Science For Business With R'. It includes a brief description: 'Your Data Science Journey Starts Now! Learn the fundamentals of data science for business with the tidyverse.', the author's name 'Matt Dancho', and a small profile picture. At the bottom of the screen, there is a progress bar indicating '0% COMPLETE'.

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

3 Course Bundle

0% COMPLETE

The first screenshot shows the course 'DS4B 101-R: Business Analysis With R'. The second screenshot shows the course 'DS4B 102-R: Shiny Web Applications For Business (Level 1)'. The third screenshot shows the course 'DS4B 201-R: Data Science For Business With R'. Each screenshot displays a similar layout with a large 'R' logo, hexagonal icons for H2O.ai, tidyverse, and Shiny, course titles, descriptions, authors, and profile pictures.

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

