

Big Data with R

2021-06-30



James
Blair

Solutions Engineer @ RStudio

Overview

1. Big Data
2. dplyr
3. dplyr + friends
4. Best practices
5. Resources



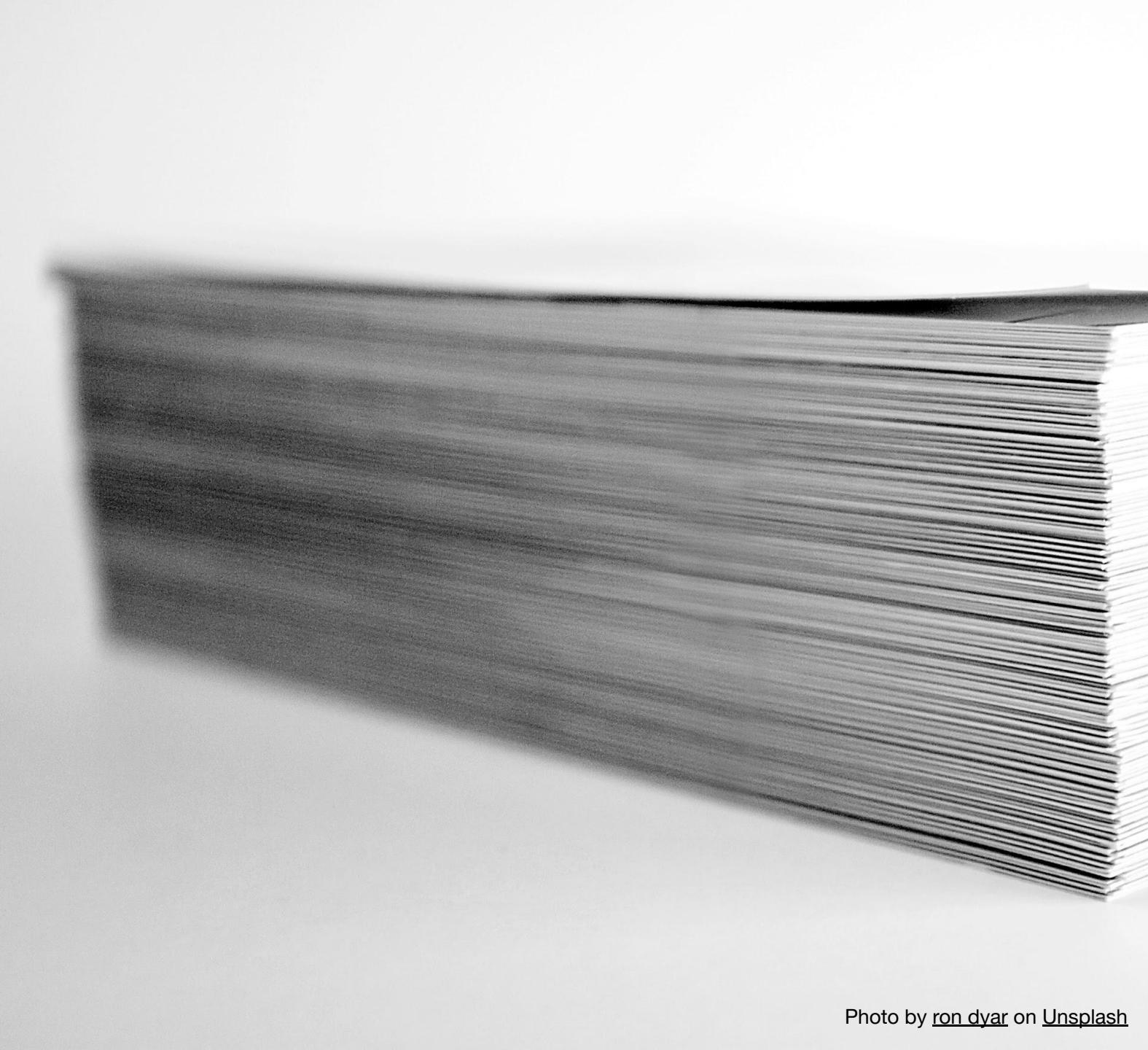
Photo by [Mat Reding](#) on [Unsplash](#)

A black and white close-up photograph of an elephant's head and trunk. The elephant's skin is highly textured with deep wrinkles and creases, particularly visible on the trunk and around the eye. Its large ears are partially visible. The lighting is dramatic, coming from the side to highlight the textures of the skin. The background is solid black, making the textured skin stand out.

Data > RAM

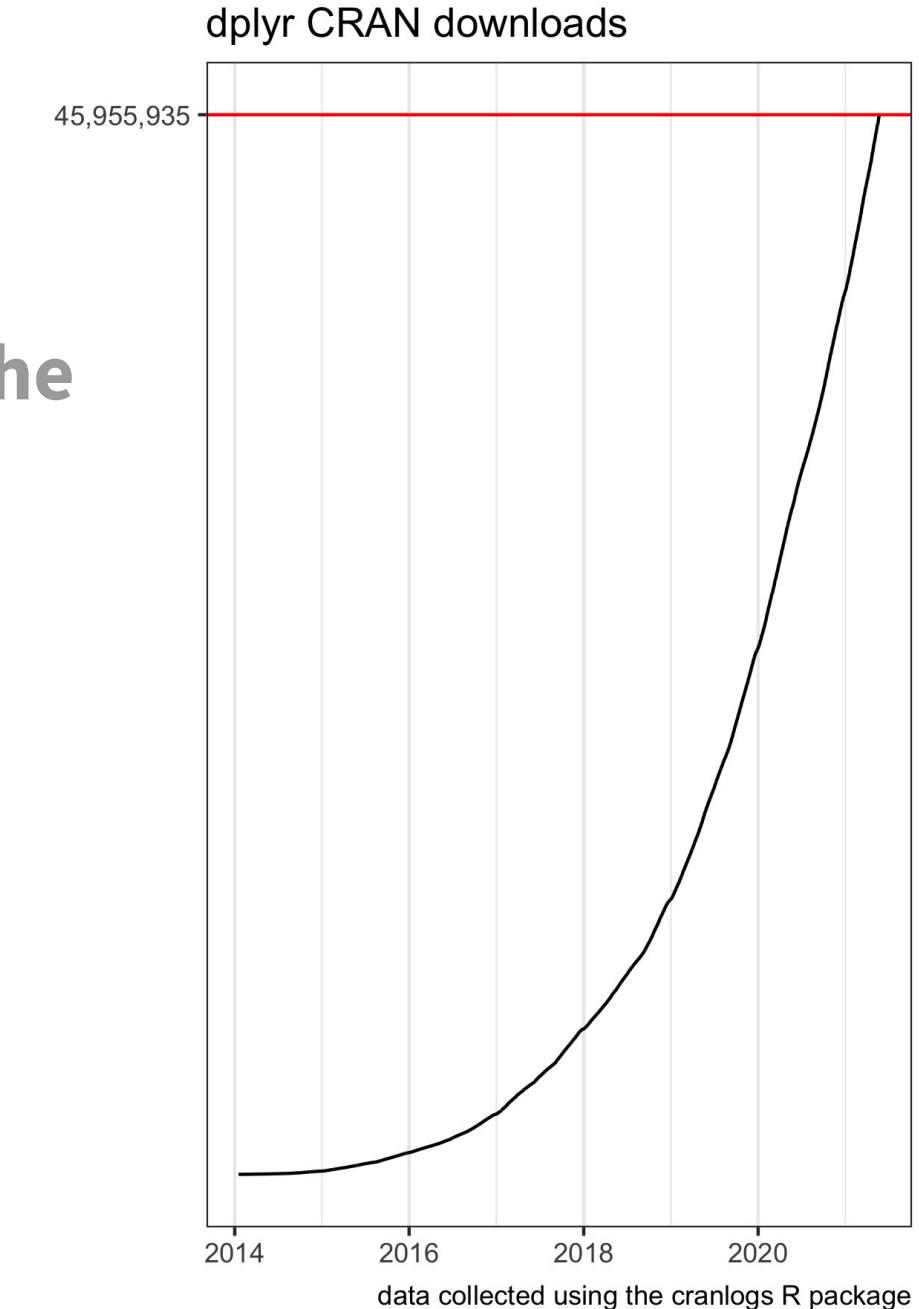
Big data forms

1. Flat file
 - a. csv
 - b. parquet
 - c. tsv
 - d. json
2. Database
3. Datalake



dplyr package

1. A grammar of data manipulation
2. Designed to **abstract over how the data is stored**
3. Consistent function interface



dplyr package



Dplyr “abstracts away how your data is stored, so that you can work with data frames, data tables and remote databases using the same set of functions.

This lets you focus on what you want to achieve, not on the logistics of data storage.

dplyr backends



Memory management

- In most cases, modifying an object in R creates a copy of that object
- Recommended to have 4x largest object of available RAM
- Solutions:
 - Minimize unnecessary object copies
 - Work with data outside of memory

Memory management

big-data - master - RStudio Pro

03-dbi.Rmd x 05-sparklyr.Rmd x 06-arrow.Rmd x Untitled > □

Go to file/function Addins Local big-data — talks

43 `arrange()`. Aggregation is not yet supported, so before you call
`summarise()`
44 or other verbs with aggregate functions, use `collect()` to pull
the selected
45 subset of the data i

46 ...{r}
47 (first_months <- ds
48 filter(date_month
49 ...

FileSystemDataset (
order_id: int64
customer_id: int64
customer_name: str

46:1 # Use `dplyr` to manip

Memory Usage Report (28% in use)

Memory Usage

Statistic Memory Source

Statistic	Memory	Source
Used by R objects	155 MiB	R
Used by session	301 MiB	MacOS System
Used by system	4,336 MiB	MacOS System
Free system memory	11,745 MiB	MacOS System
Total system memory	16,384 MiB	MacOS System

28% memory in use

OK

6) Arrow object

Viewer More C

Rdio > talks > big-data > R ...

Size Modified

1.3 MB May 28
2 KB May 28
1.1 MB May 28
3.9 KB Jun 29,
810.6 KB May 28
5.3 KB Jun 29,
832.2 KB May 28
3.1 KB Jun 29,
1.4 MB May 24
1.8 KB May 24
191 B Mar 10,

Environment History Connect 301 MiB List C

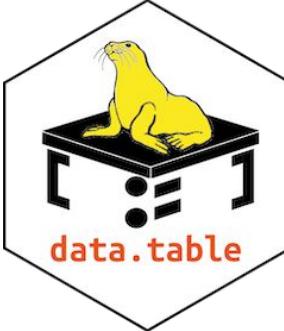
Import Dataset R Global Environment

Data

by_month List of 7
first_months List of 7
local_ds 250000 obs. of 14 variables

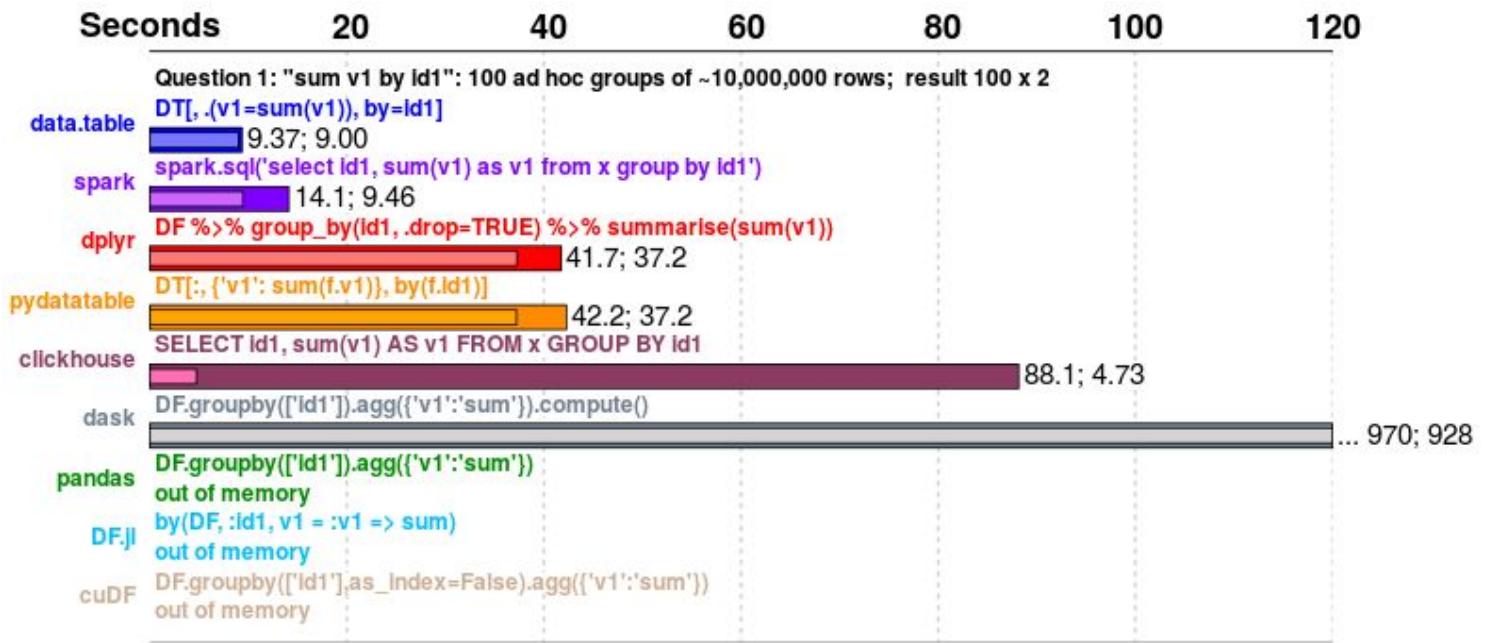
Values

The screenshot shows the RStudio interface with a dark theme. In the top-left, there's a code editor window with R code related to dplyr and arrow packages. A modal dialog titled 'Memory Usage Report' is open in the center, showing a pie chart with 28% usage and a table of memory statistics. The table includes rows for 'Used by R objects', 'Used by session', 'Used by system', 'Free system memory', and 'Total system memory'. The 'Used by system' row is notably large at 4,336 MiB. Below the modal, the RStudio environment pane shows various objects like 'by_month', 'first_months', and 'local_ds'. The bottom right corner shows a file browser with several R Markdown files and a config file.



data.table package

1. High performance version of base R data.frame
2. Fast file reader fread
3. Concise syntax DT[i, j, by]



dtplyr package



The goal of dtplyr is to allow you to write dplyr code that is automatically translated to the equivalent, but usually much faster, data.table code.

dtplyr package

1. Provides a `data.table` backend for `dplyr`
2. Combine the syntax of `dplyr` with the speed of `data.table`
3. Lazy evaluation
4. Converts `dplyr` syntax to `data.table` syntax



dplyr package

A word about copying...

In data.table parlance, all set functions change their input by reference. That is, no copy is made at all, other than temporary working memory, which is as large as one column.*

Use `lazy_dt(x, immutable = FALSE)` to prevent dplyr from making copies.



arrow package

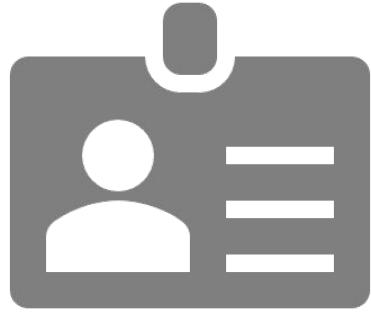


- High-performance reading and writing of data files with multiple file formats and compression codecs, including built-in support for cloud storage
- **Analyzing and manipulating bigger-than-memory data with dplyr verbs**
 - Summarization functions are not currently supported

Databases



Connection requirements



Credentials



Location



Driver

Requirement definitions



- User name & password
 - Token
-

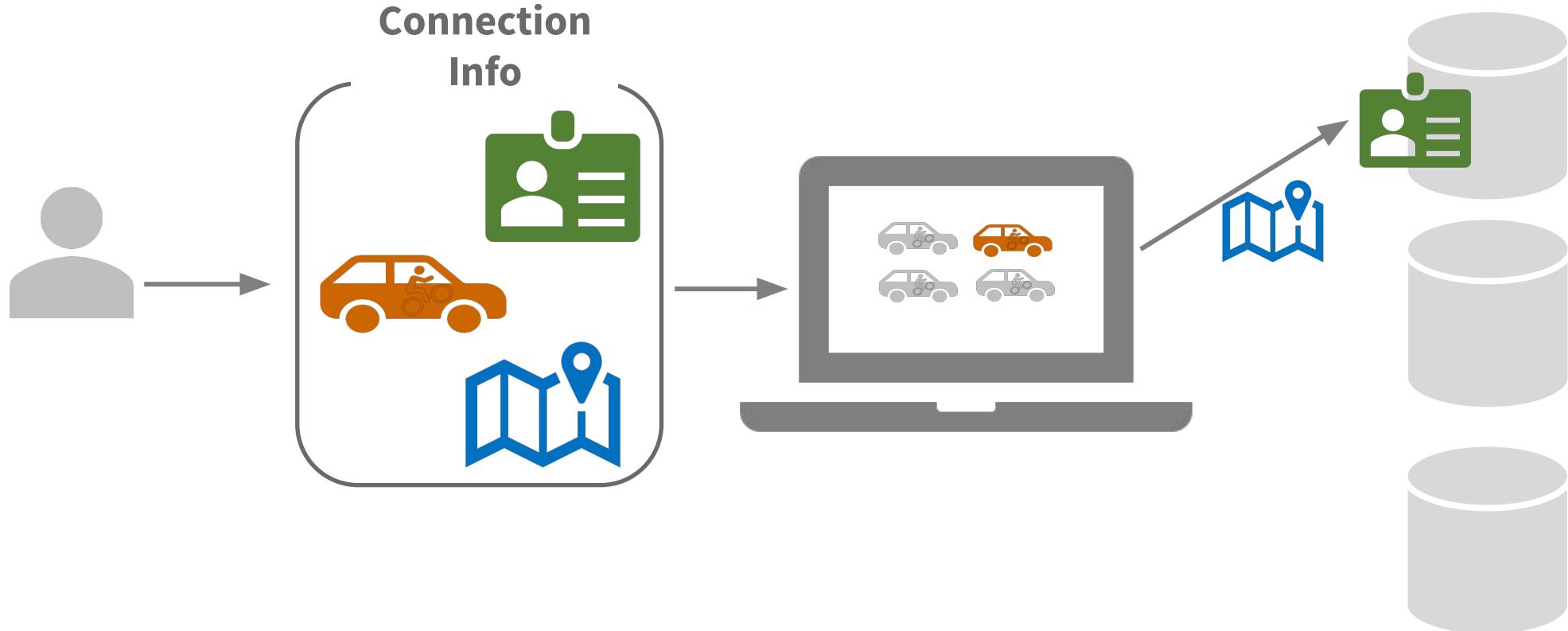


- URL
 - IP Address
-

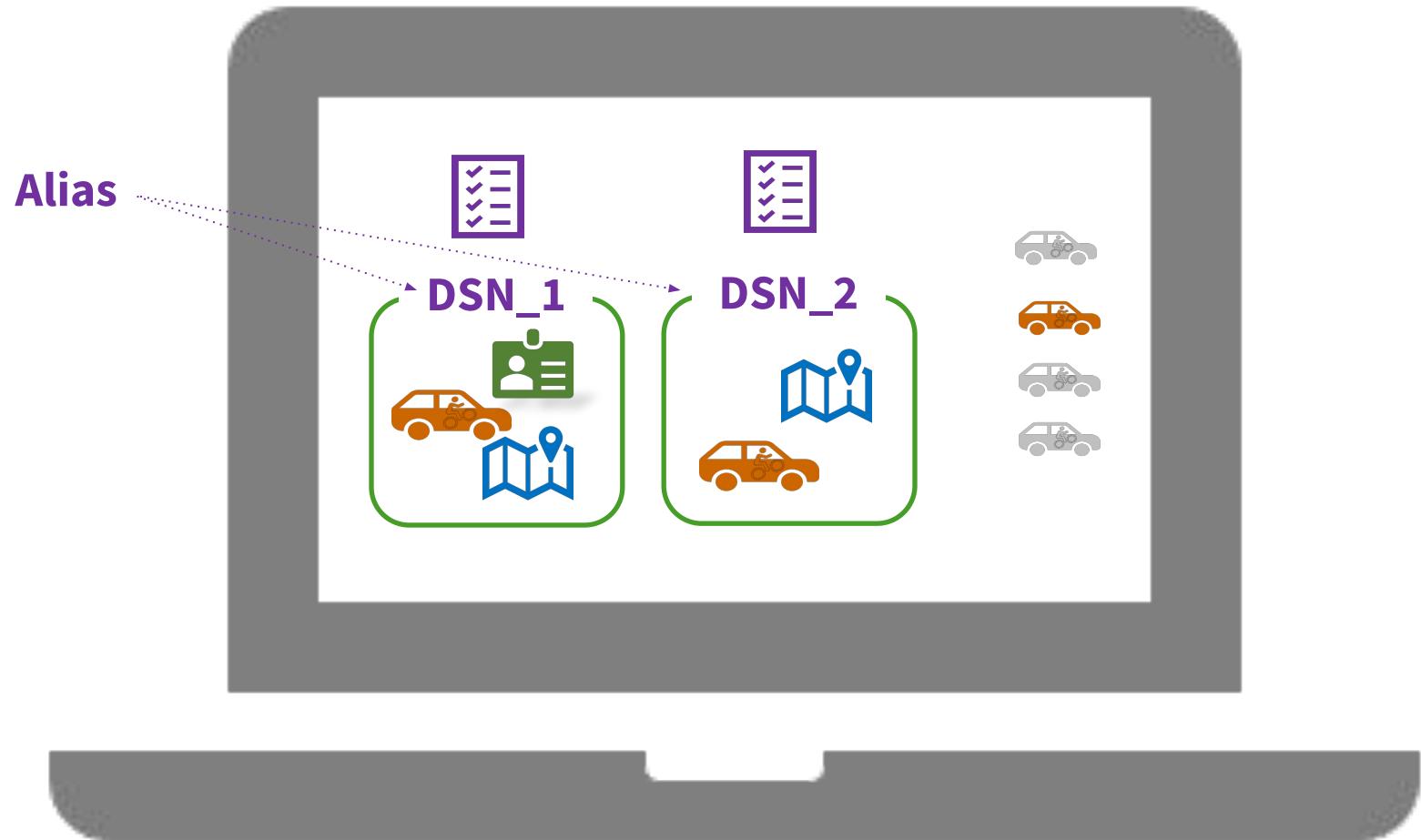


- ODBC (Used by **ADO & OLE DB**)
- JDBC

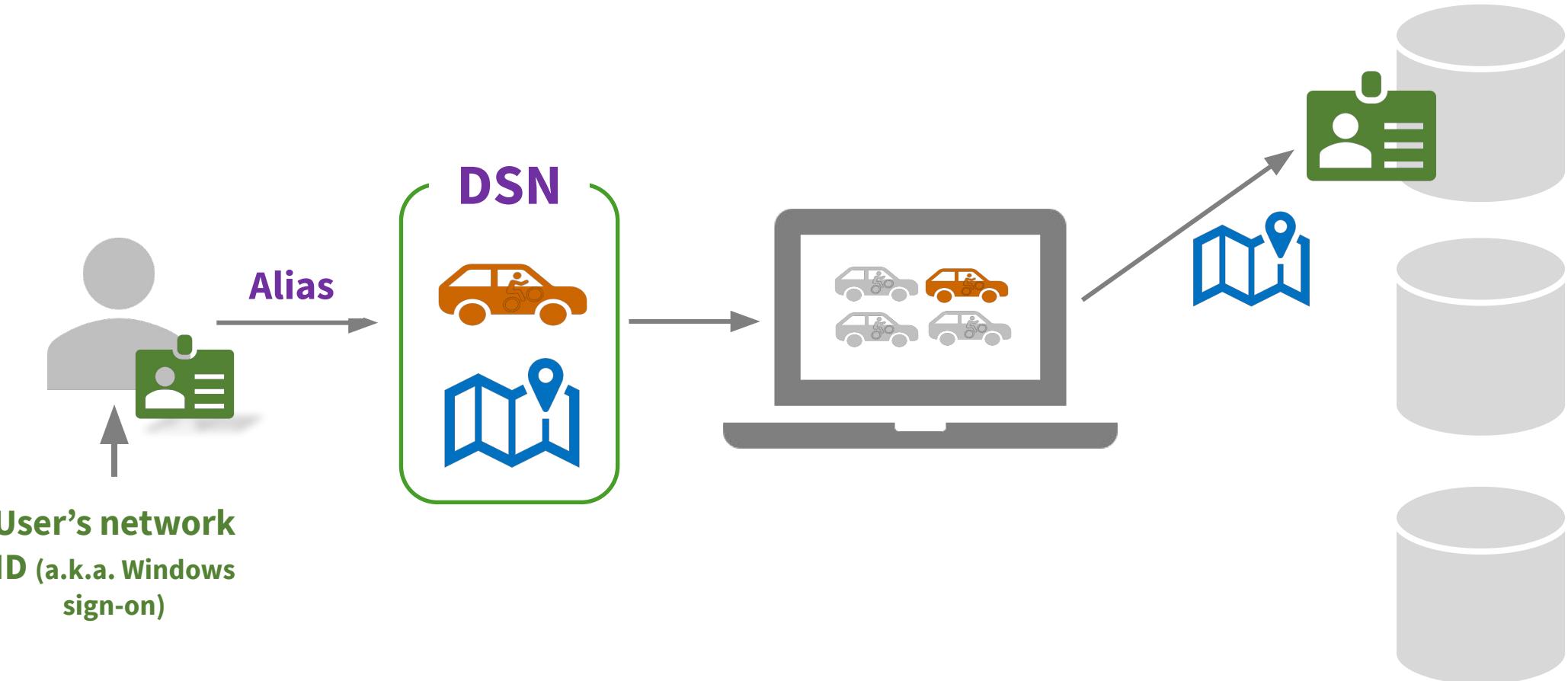
Connection info



Data Source Name (DSN)



The ideal connection



The connections pane

The screenshot shows the RStudio interface with the Connections pane highlighted by a blue rounded rectangle. The Connections tab is selected in the top navigation bar. A connection named "postgres - rstudio_dev@localhost" is listed. Below the Connections pane is a file browser showing a directory structure with files like .gitignore, .Rbuildignore, .Renviron, and several RMD files.

```
1 ````{r db-connections, include = FALSE}
2 if(Sys.getenv("GLOBAL_EVAL") != "") eval_connections <-
Sys.getenv("GLOBAL_EVAL")
3 eval_connections <- FALSE
4
5
6 ````{r, eval = eval_connections, include = FALSE}
7 library(DBI)
8 library(odbc)
9 library(config)
10 library(keyring)
11
12
13 # Introduction to database connections
14
15 ## Connect with the Connections pane
```

15:1 # Connect with the Connections pane

Name	Size	Modified
..		
.gitignore	100 B	Jan
.Rbuildignore	28 B	Jan
.Renviron	40 B	Jan
01-intro-to-vroom.Rmd	4.7 KB	Jan
02-intro-to-dtplyr.Rmd	4.8 KB	Jan
03-db-connections.Rmd	4.3 KB	Jan
04-intro-to-DBI.Rmd	4.8 KB	Jan
05-db-analysis.Rmd	3.4 KB	Jan

The screenshot shows the RStudio Connections dialog box. It has tabs for New Connection, Connection, Status, and a sidebar with options like Postgres Dev, Postgres Prod, SQL Server (DSN), Pins, Livy, Spark, Athena, and BigQuery. A help section at the bottom right provides information on Using RStudio Connections.

New Connection

Connection

Status

Postgres Dev

Postgres Prod

SQL Server (DSN)

Pins

Livy

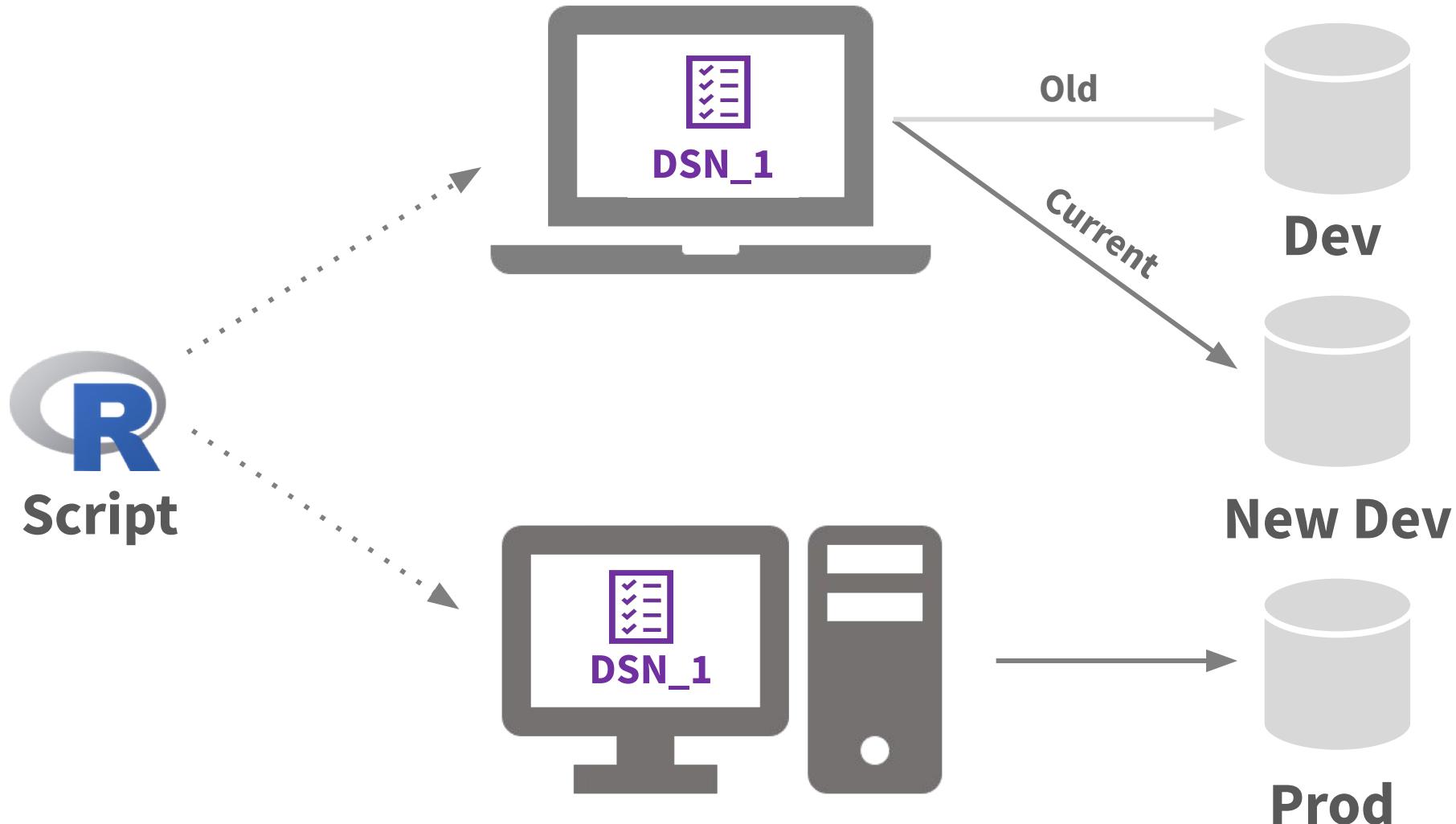
Spark

Athena

BigQuery

Using RStudio Connections

Why DSN?



Alternatives for securing connections

1. config
2. keyring
3. Environment variables
4. options()
5. Prompt for credentials

R packages

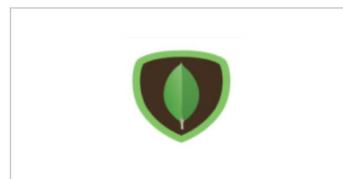
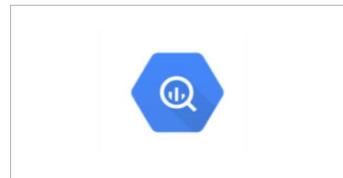
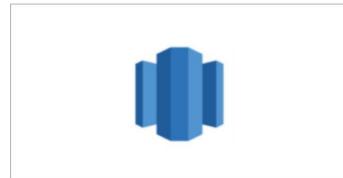
General connections

- DBI
- odbc
- connections

Specific Connections

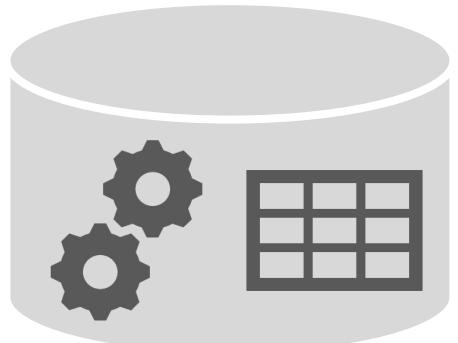
- sparklyr
- RPostgres
- RSQLite
- . . .

ODBC Drivers



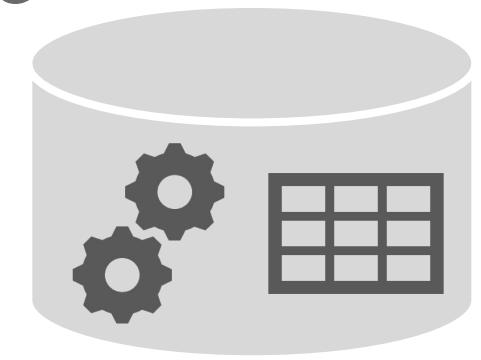
Wrangle inside the DB

Time Consuming



Extract Data

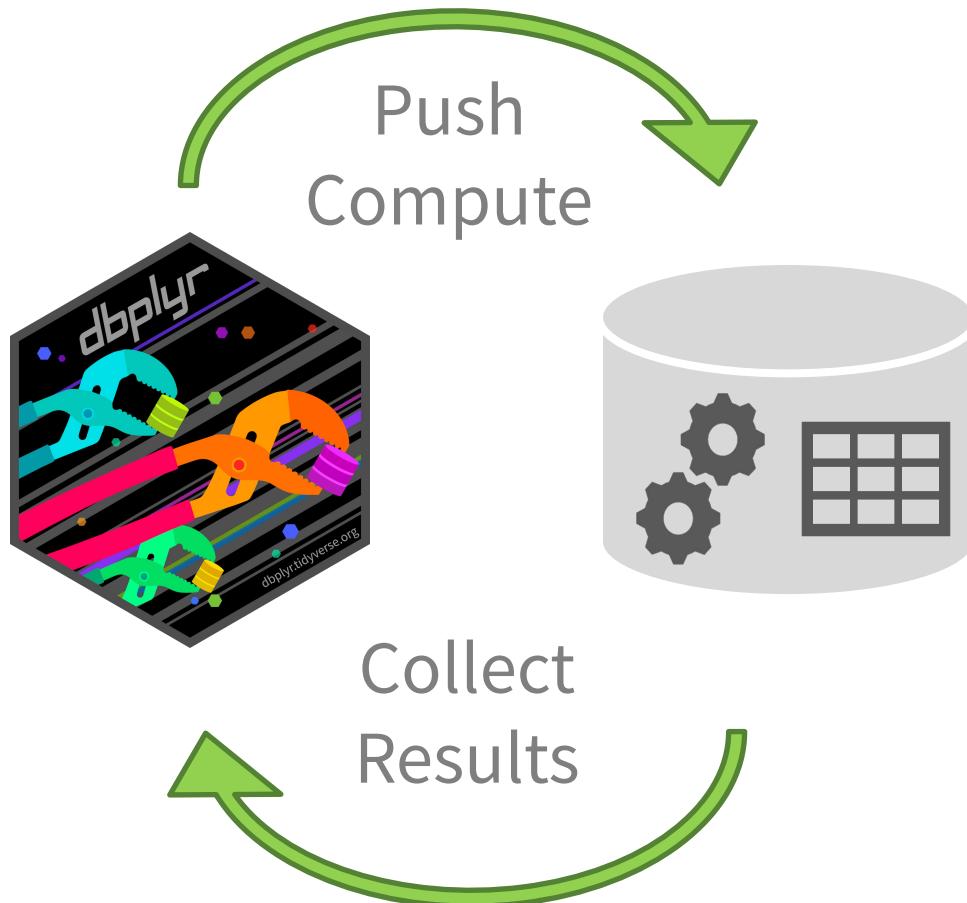
Push
Compute



Collect
Results



Advantages



1. `dplyr` translates to SQL
2. Take advantage consistent syntax
3. All your code is in R!

DBI package

1. Stands for **database interface**
2. Helps connect R to various database management systems
3. Used for connecting to and interacting with various databases
4. Execute SQL commands against the database



DBI common functions

Connecting

- dbConnect
- dbDisconnect

Queries

- dbSendQuery
- dbGetQuery
- dbExecute

Tables

- dbListTables
- dbWriteTable
- dbReadTable

Options to Push Compute

Write SQL statements

```
SELECT "customer_id",
COUNT(*) AS "n"
FROM "retail.orders"
GROUP BY "customer_id"
```

Use dplyr verbs

```
orders %>%
  count(customer_id)
```

sparklyr package

1. Leverage the power of Spark from R
2. Take advantage of distributed,
in-memory processing
3. Create extension that leverage the
Spark API



Best Practices

- Use `dplyr` for consistent syntax and tooling
- Rely on each tool for its strengths
- Leave heavy computation to tools designed for heavy computation (Databases, Spark)
- Use R for final analysis
- When possible, avoid bringing **all** data into R
- Create extracts and views to speed things up

Bookmark and check regularly

- <http://db.rstudio.com/>
- <http://spark.rstudio.com/>
- <https://www.tidyverse.org/>
- <https://arrow.apache.org/docs/r/>
- <https://rviews.rstudio.com/>
- <https://rviews.rstudio.com/categories/databases>
- <https://blog.rstudio.com/>

Join the community!

R Studio Community

Sign Up

Log In

Jobs News



RStudio Community

All things RStudio



FIND HELP



CONTRIBUTE



EXPLORE

<https://community.rstudio.com/>

Familiarize yourself with the repos

If I need to...	Check out
Report an issue or see if others are having the same problem	Issues
See if an feature exists or if it's coming up in future releases	NEWS
See the basics about the package	README

- <https://github.com/tidyverse/dplyr>
- <https://github.com/tidyverse/dbplyr>
- <https://github.com/tidyverse/ggplot2>
- <https://github.com/r-dbi/odbc>
- <https://github.com/r-dbi/DBI>
- <https://github.com/edgararuiz/dbplot>
- <https://github.com/tidymodels/tidypredict>
- <https://github.com/apache/arrow/tree/master/r>
- <https://github.com/rstudio/sparklyr>



Photo by Courtney Smith on Unsplash

Thank you

<https://bit.ly/QNB-big-data>