Big Data with R

SURF Meetup | February 18, 2020

James Blair

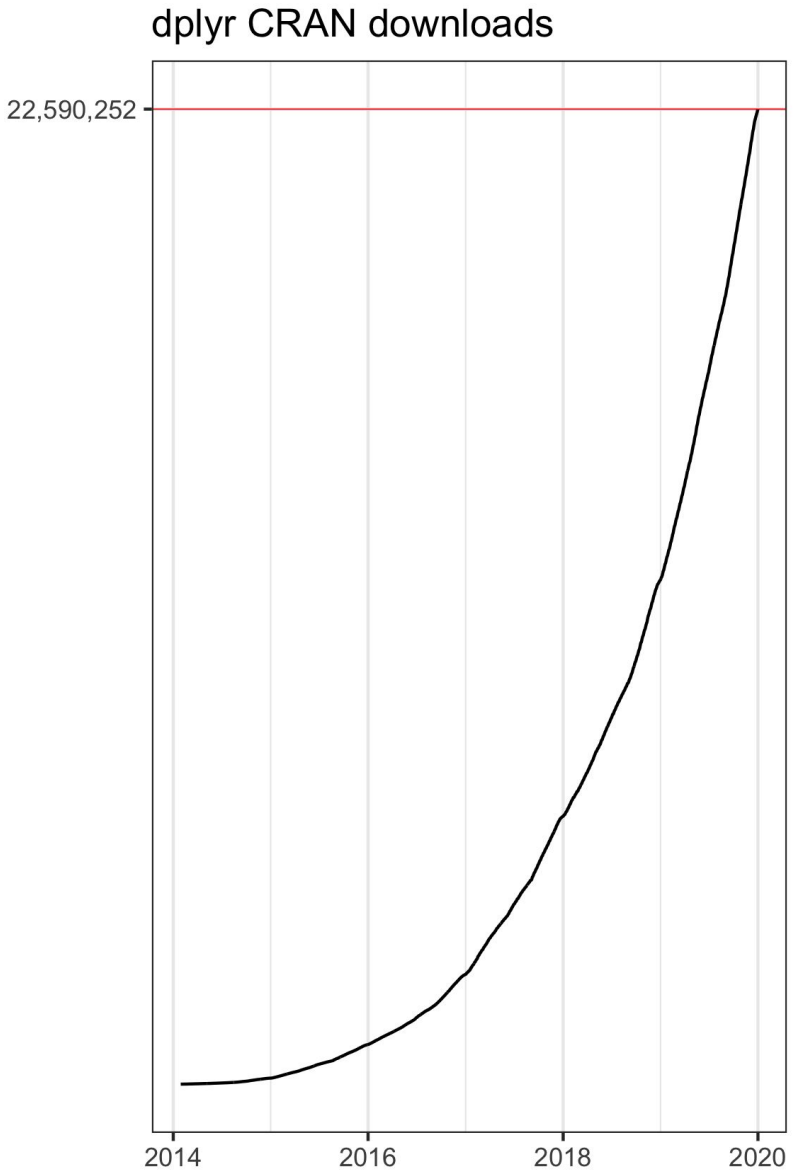
Solutions Engineer @ RStudio

Photo by Dan Freeman on Unsplash

RStudio

`dplyr` package

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



Data collected using the `cranlogs` R package

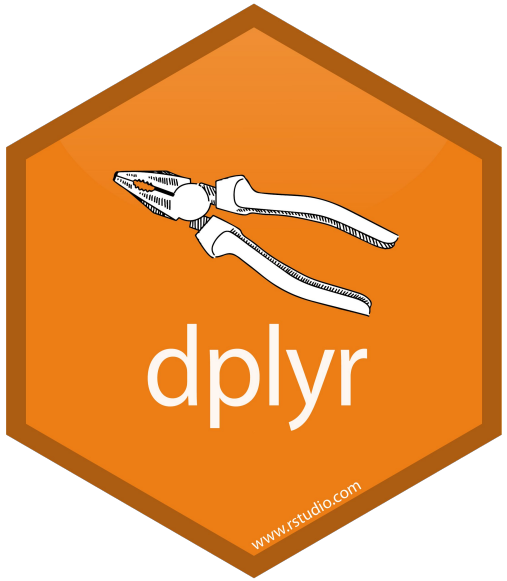
RStudio

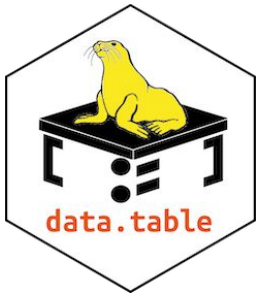
dplyr.tidyverse.org

`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





Apache Arrow

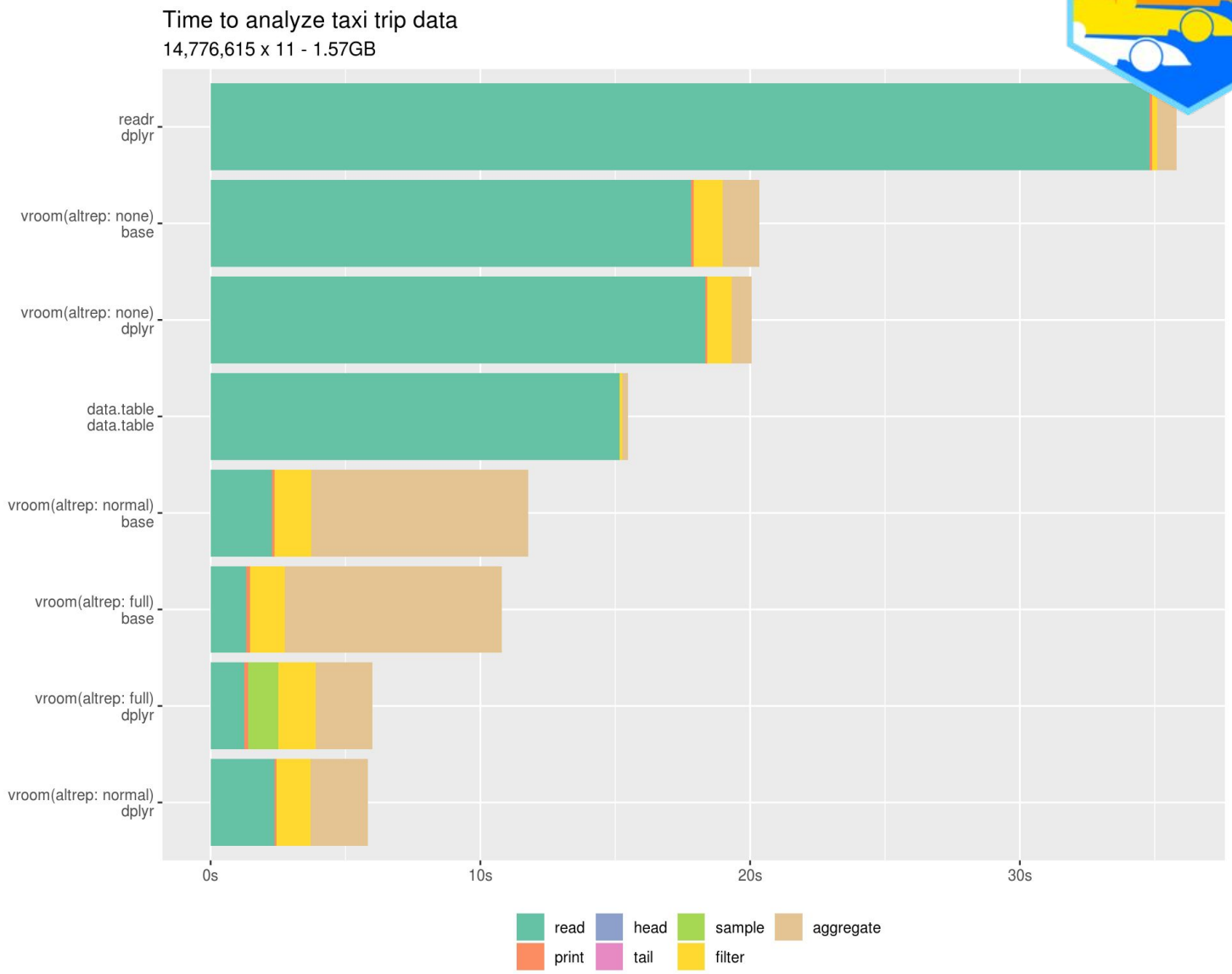
1. Cross-language platform for in-memory data
2. Exciting applications for "big data"
3. `dplyr` compliant

APACHE ARROW

http://bit.ly/arrow-exp

vroom package

1. Initially indexes data but does not read it
2. Loads the data into R only when needed
3. Super fast! 1.27 GB/sec



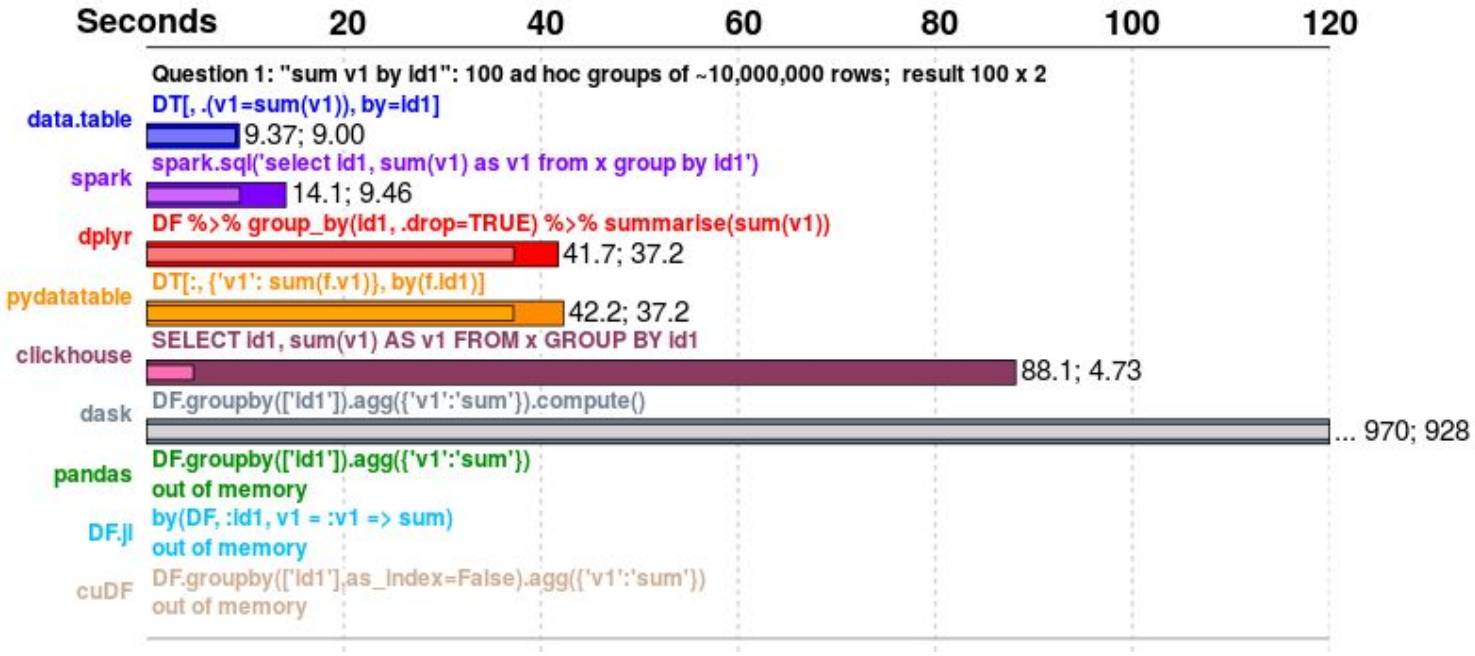
vroom.r-lib.org

`vroom` features

1. Nearly all parsing features of `readr`
2. `skip` and `n_max` arguments
3. Column selection
4. Read from multiple files or connections

`data.table` package

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



rdatatable.gitlab.io/data.table

`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

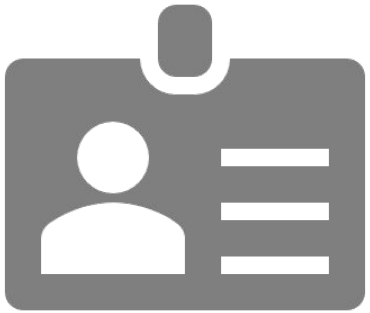
`dtplyr` package

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.

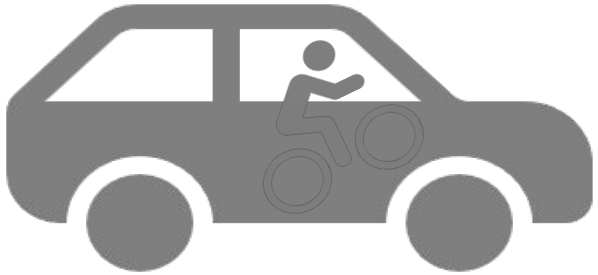
Use `lazy_dt(x, immutable = FALSE)` to prevent `dtplyr` from making copies.

Connection requirements



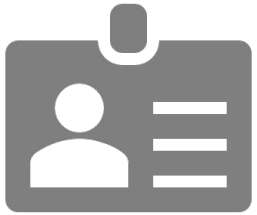
Credentials

Location



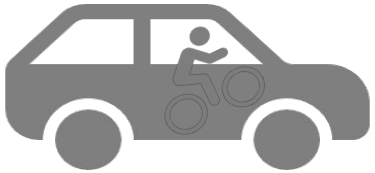
Driver

Requirement definitions



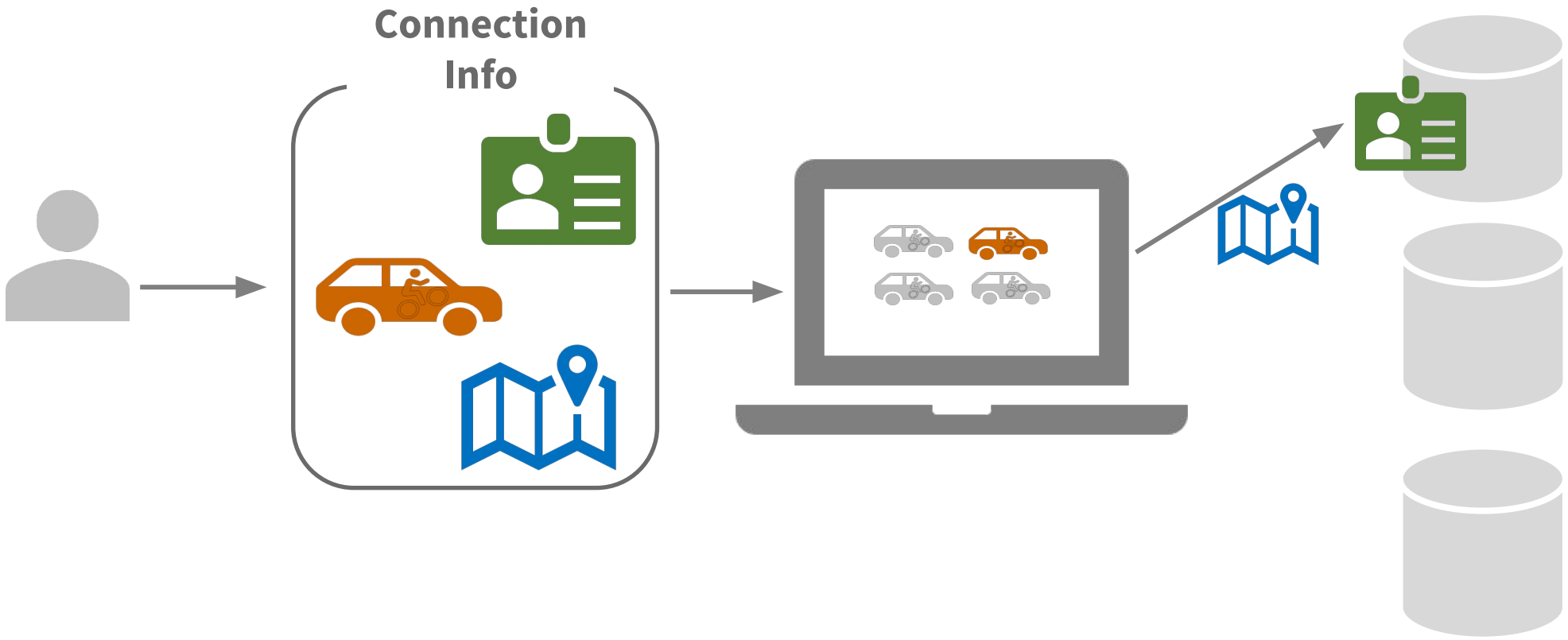
- User name & password
- Token

- URL
- IP Address

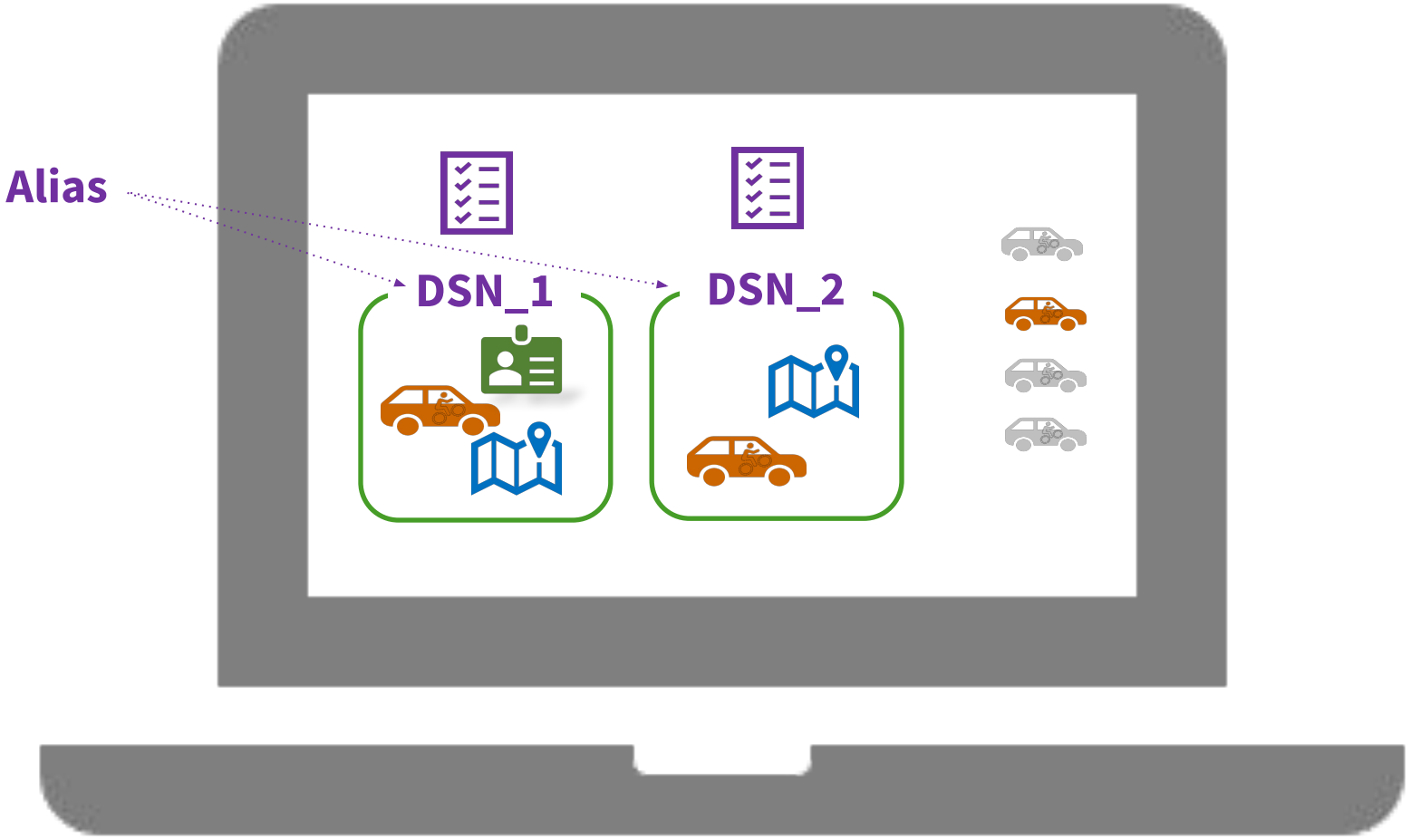


- ODBC (Used by **ADO** & **OLE DB**)
- JDBC

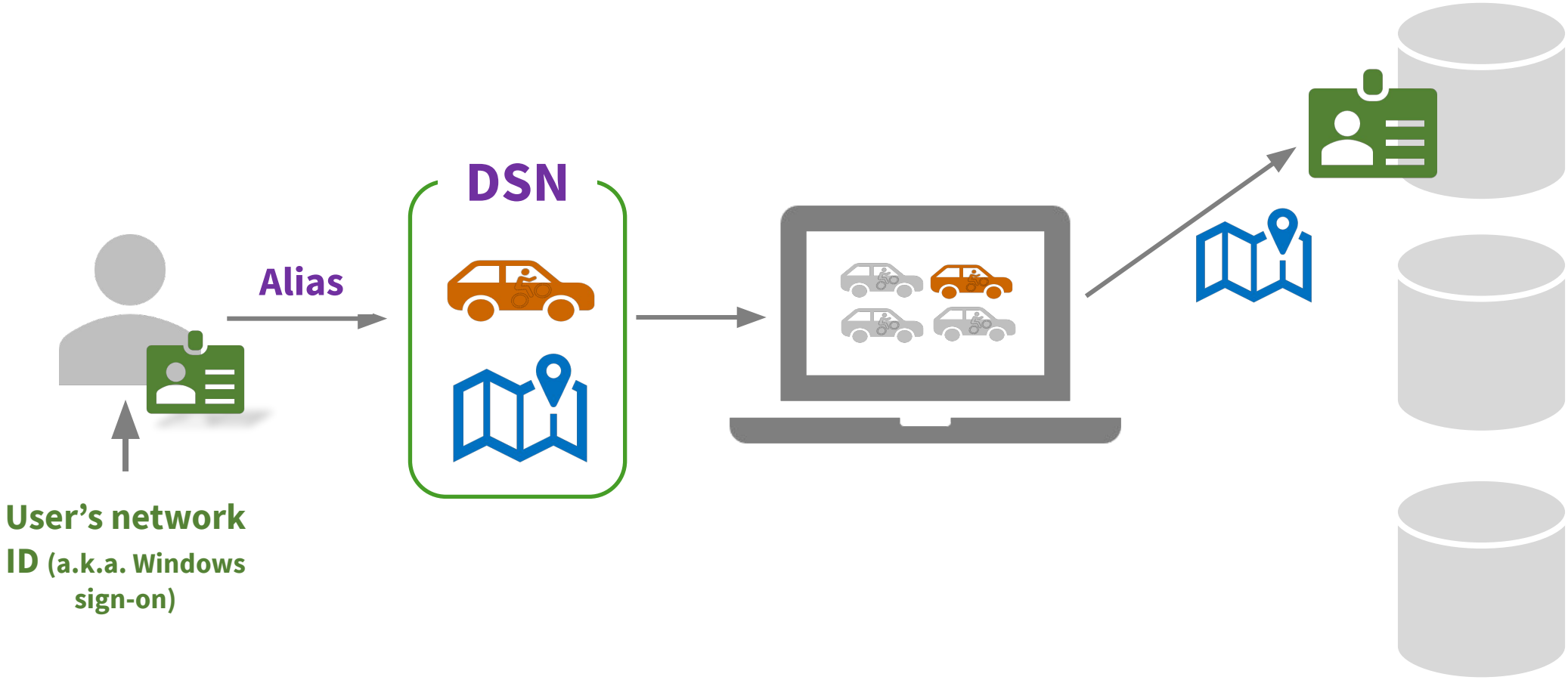
Connection info



Data Source Name (DSN)



The ideal connection



The connections pane

File Edit Code View Plots Session Build Debug Profile Tools Help

bigdata_user Sessions

Go to file/function Addins big-data R 3.6.2

03-db-connections.Rmd

Knit Insert Run

```{r db-connections, include = FALSE}
if(Sys.getenv("GLOBAL_EVAL") != "") eval_connections <- Sys.getenv("GLOBAL_EVAL")
eval_connections <- FALSE
```

```{r, eval = eval_connections, include = FALSE}
library(DBI)
library(odbc)
library(config)
library(keyring)
```

Introduction to database connections

Connect with the Connections pane

Introduction to da…
Connect with the …
Connecting via D…
Connect with a c…
Secure connectio…

15:1 Connect with the Connections pane R Markdown

Console Terminal Find in Files Launcher

~/big-data/

>

Environment History Connections Git

New Connection

Connection	Status
postgres – rstudio_dev@localhost	

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Home > big-data

Name	Size	Modif
..		
.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

Insert Run

Environment History Conn

New Connection

Connection

postgres – rstudio_dev@loc

Introduction to da…
Connect with the …
Connecting via D…
Connect with a c…
Secure connectio…

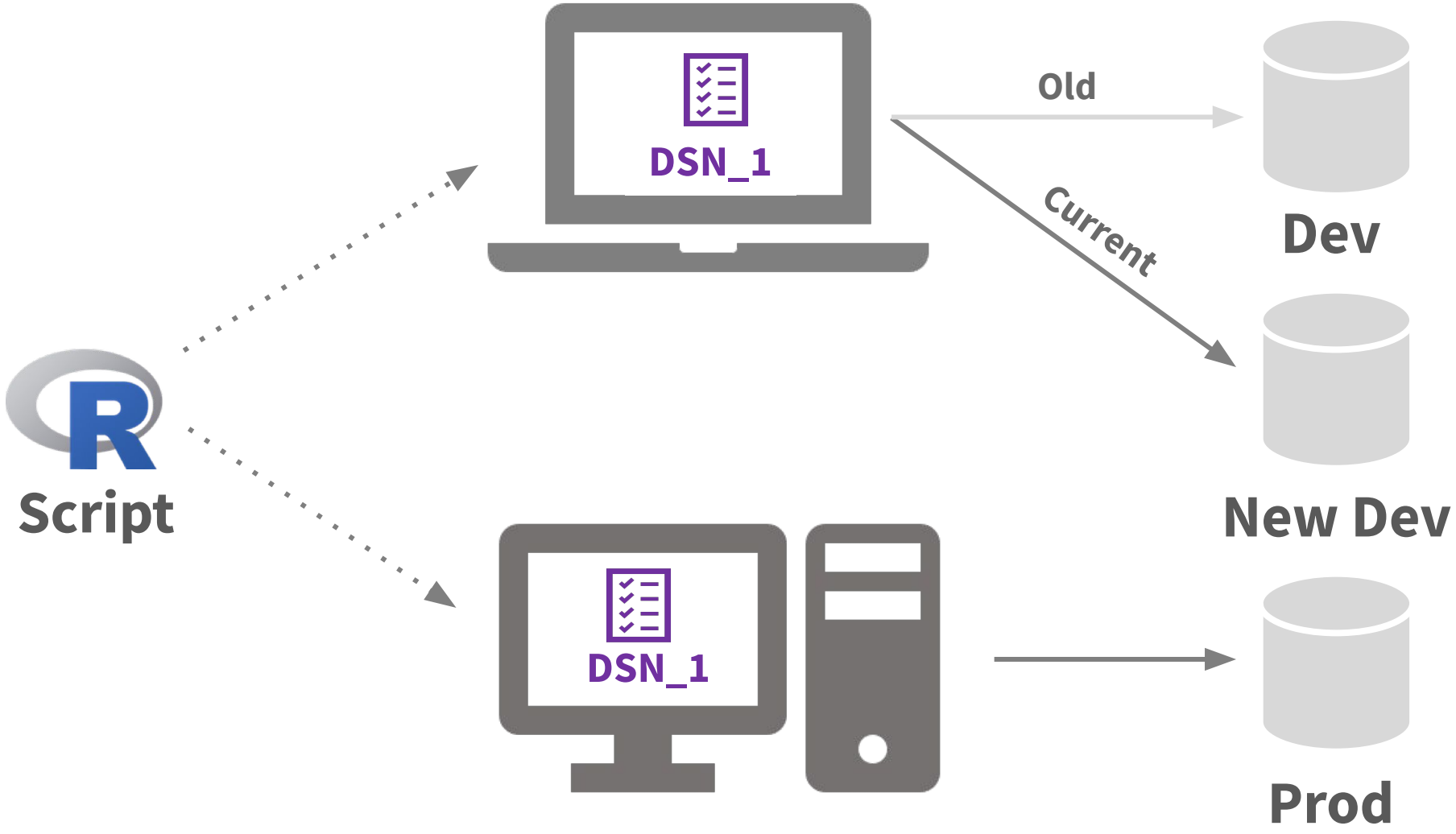
New Connection

Connect to Existing Data Sources

- Postgres Dev
- Postgres Prod
- SQL Server (DSN)
- Pins
- Livy
- Spark
- Athena
- BigQuery

Using RStudio Connections Cancel

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

- bigrquery
- RPostgres
- RSQLite
- RMariaDB
- sparklyr

github.com/r-dbi

DBI package

1. Stands for **d**ata**b**ase **i**nterface
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`

Tables

- `dbListTables`
- `dbWriteTable`
- `dbReadTable`

Queries

- `dbSendQuery`
- `dbGetQuery`
- `dbExecute`

Unit 5

Databases with dplyr

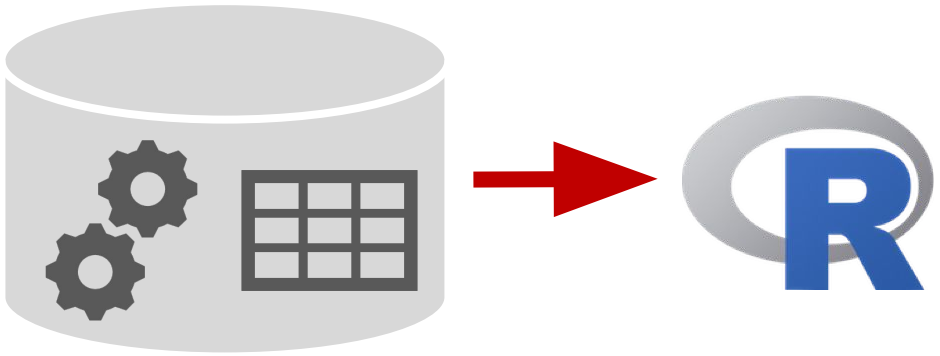
/dee-plier/

Photo by Arthur Lambillotte on Unsplash

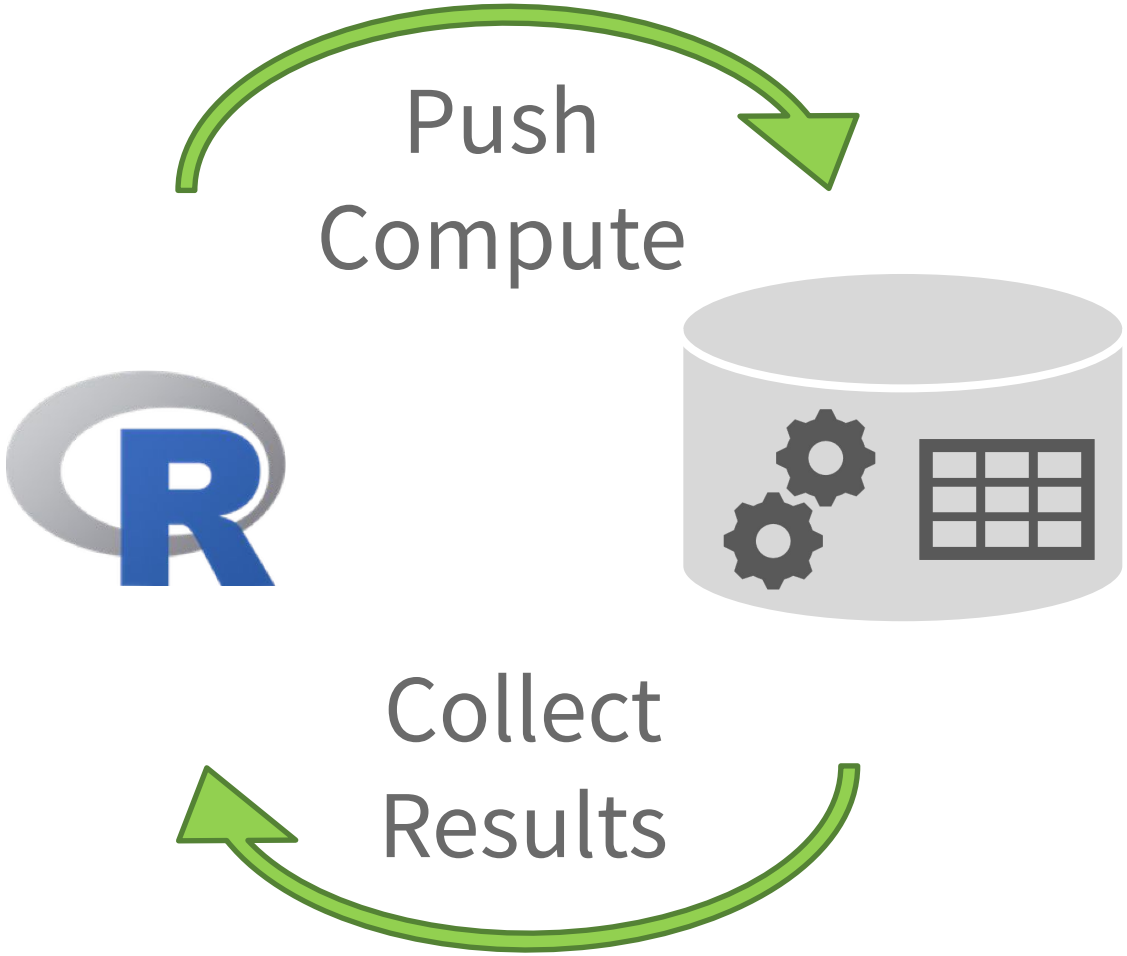
RStudio

Wrangle inside the DB

Time Consuming



Extract Data



Options to Push Compute

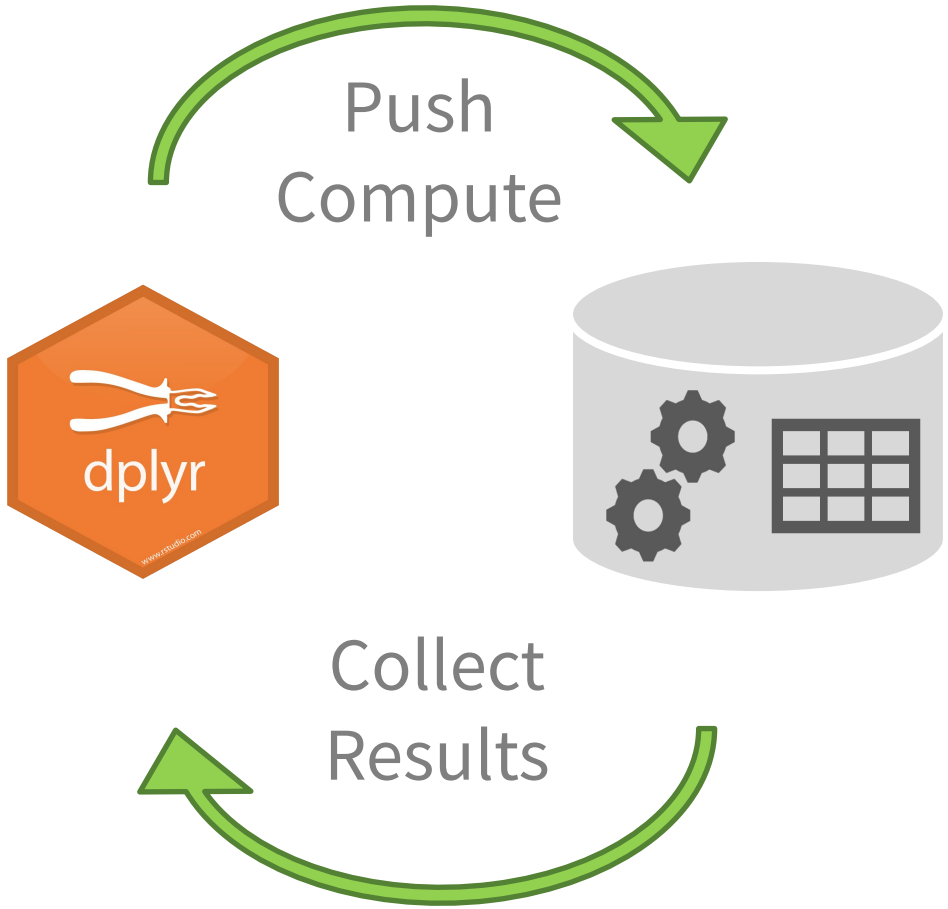
Write SQL statements

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

Use dplyr verbs

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

Advantages

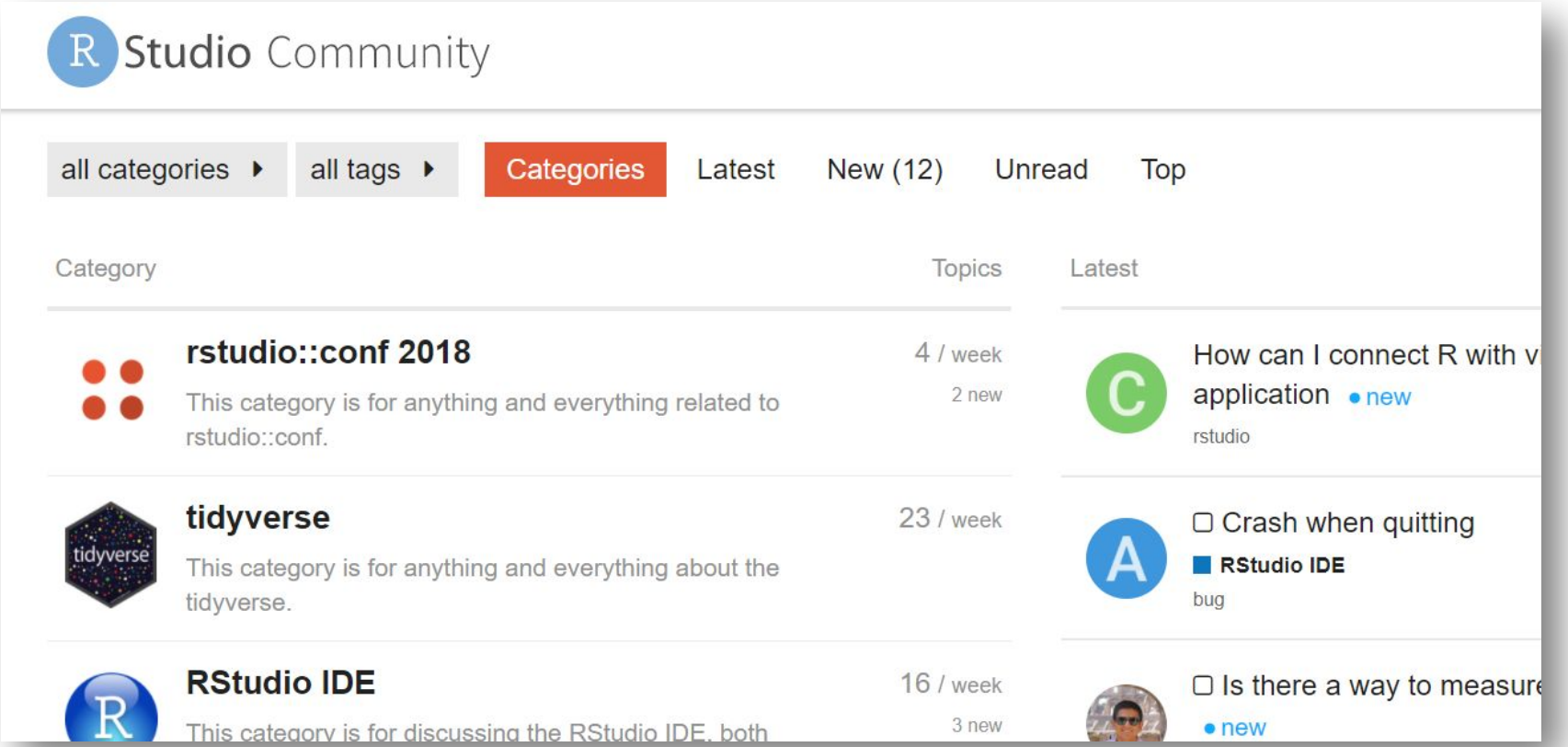


1. dplyr translates to SQL
2. Take advantage consistent syntax
3. All your code is in R!

Bookmark and check regularly

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

Join the community!



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/rstudio/sparklyr

http://bit.ly/big-data-surf

Photo by Bob Walker on Unsplash

Thank you