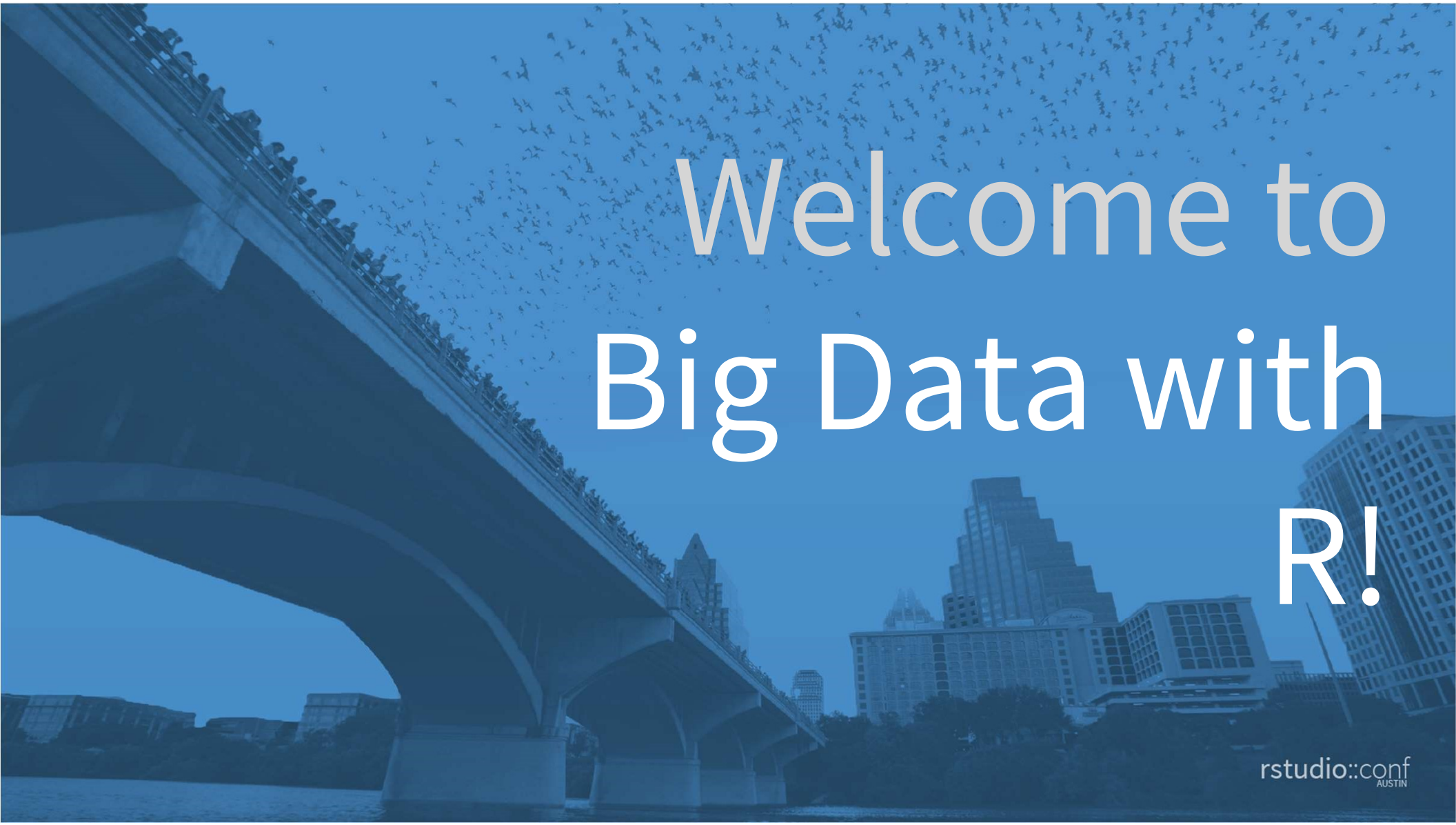

Welcome to
Big Data with
R!
rstudio::conf
AUSTIN

Housekeeping items

- Wi-fi password
- rstudio::conf app
- Access your server

Schedule

9am – 10:30am

Break (30 mins)

11am – 12:30am

Lunch (1hr)

1:30pm – 3pm

Break (30 mins)

3:30pm – 5pm

The team

Cole Arendt
Infrastructure

Mara Averick
TA

Ron Blum
TA

Javier Luraschi
Guy in the back

James Blair
Instructor

Edgar Ruiz
Instructor

Pre-class Survey Review

rstudio::conf AUSTIN

Class / material overview

- Server
- Database
- Spark
- Deck
- Exercise book

Unit 1
Accessing databases

Photo by Florian Pircher on Unsplash

Exercise 1.1 – 1.3

Connection requirements

Credentials

Location

Driver

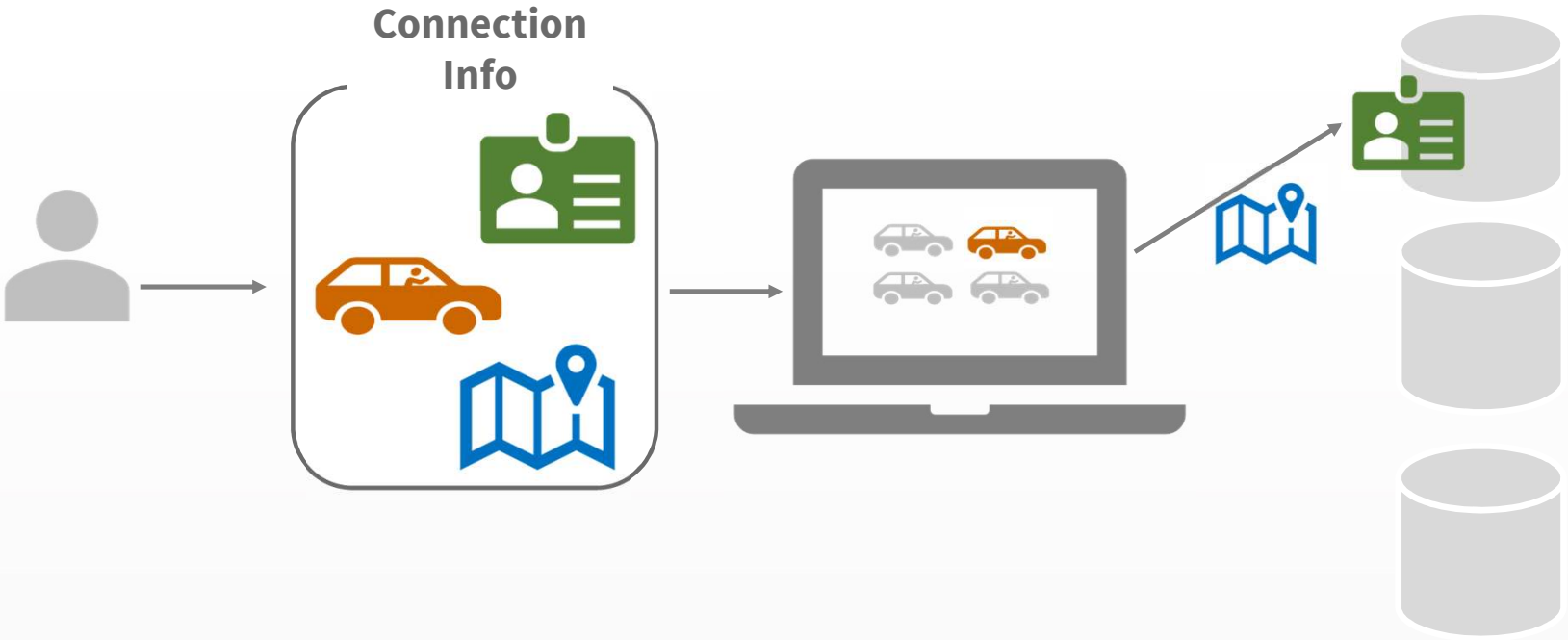
Requirement definitions

- User name & password
- Token

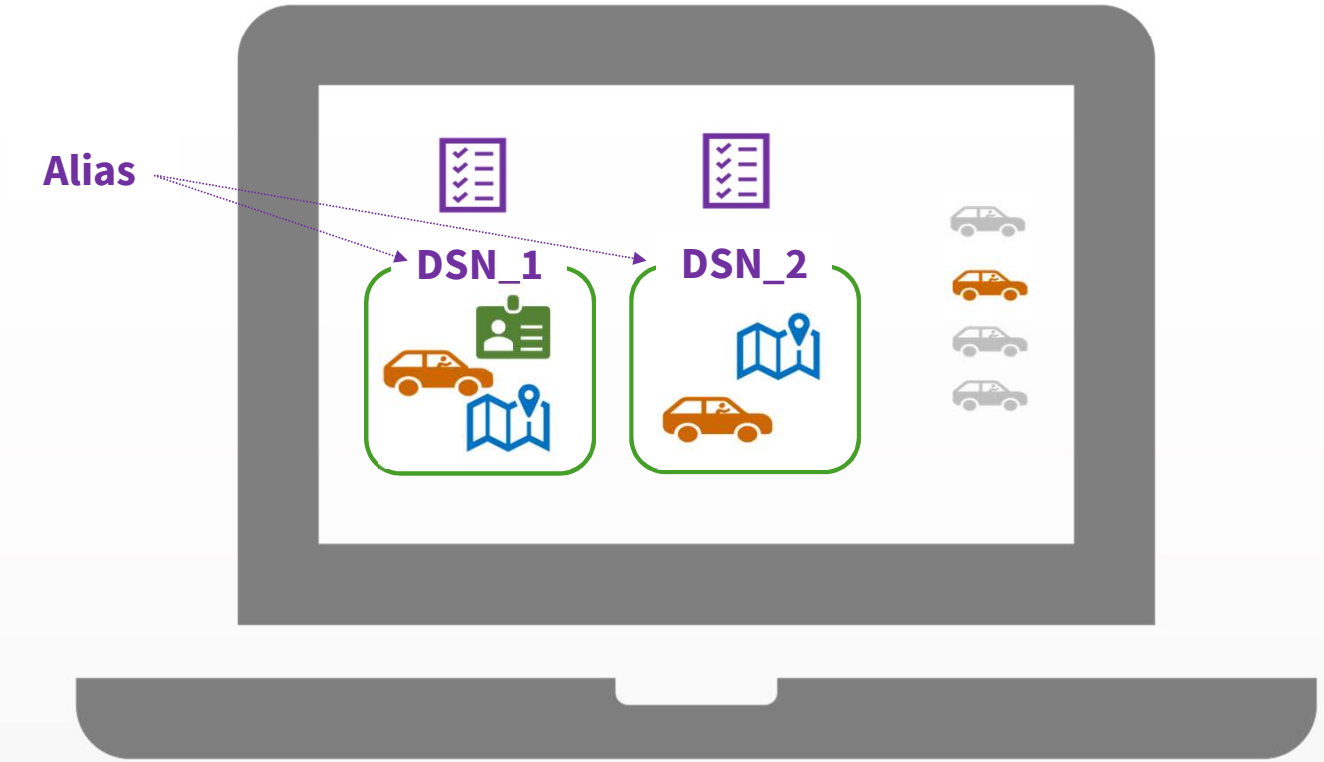
- URL
- IP Address

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

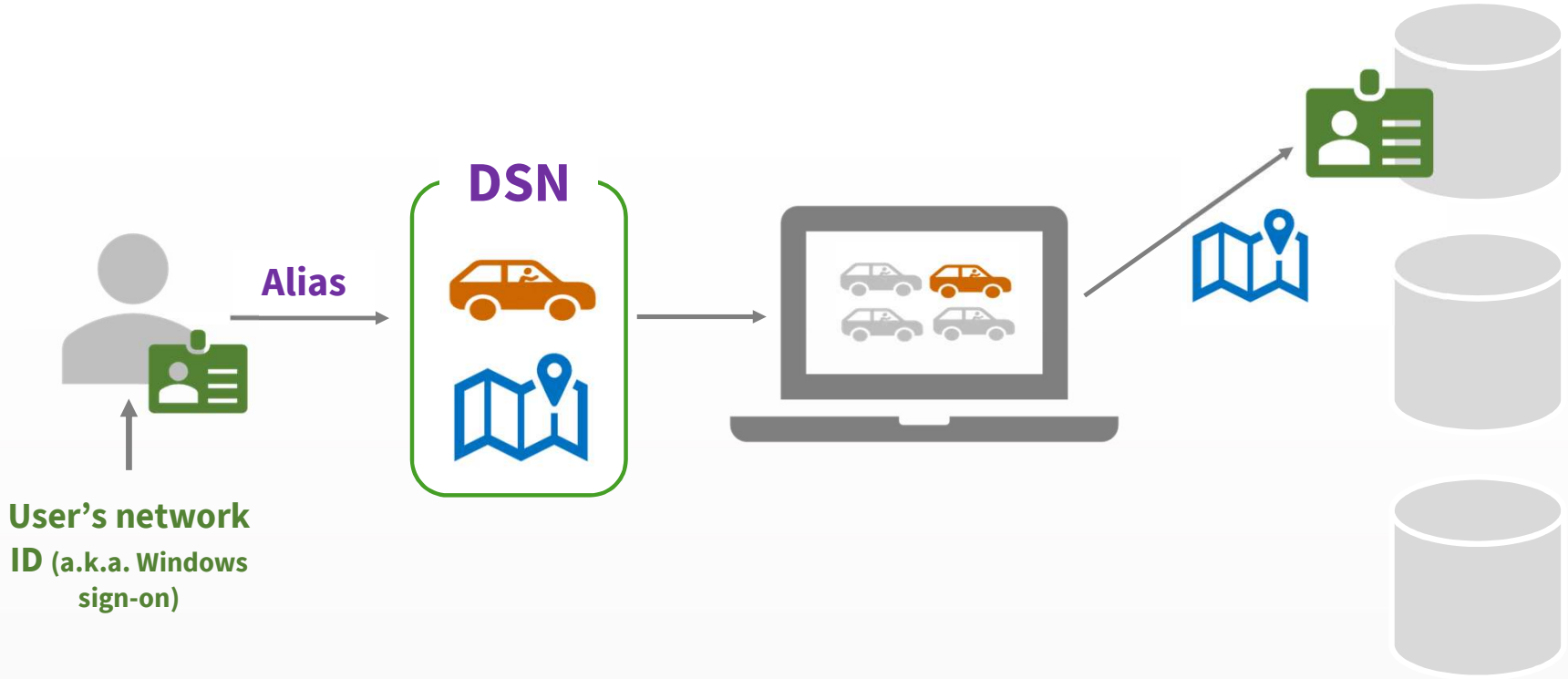
Connection info



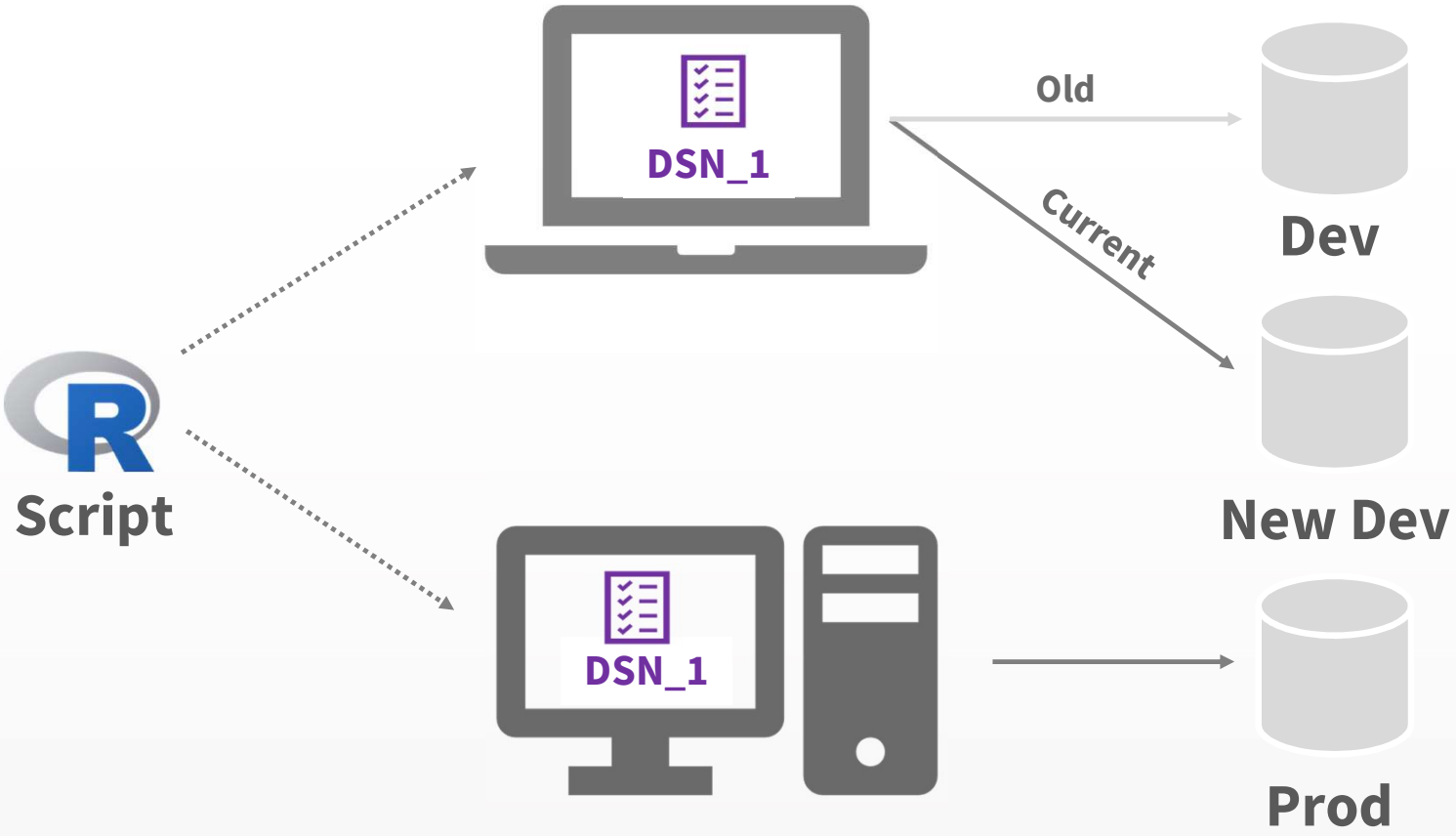
Data Source Name (DSN)



The ideal connection



Why DSN?



Exercise 1.4

Alternatives for securing connections

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

Exercise 1.5 – 1.9

Let's talk about Big Data

Photo by Chris Christensen on Unsplash

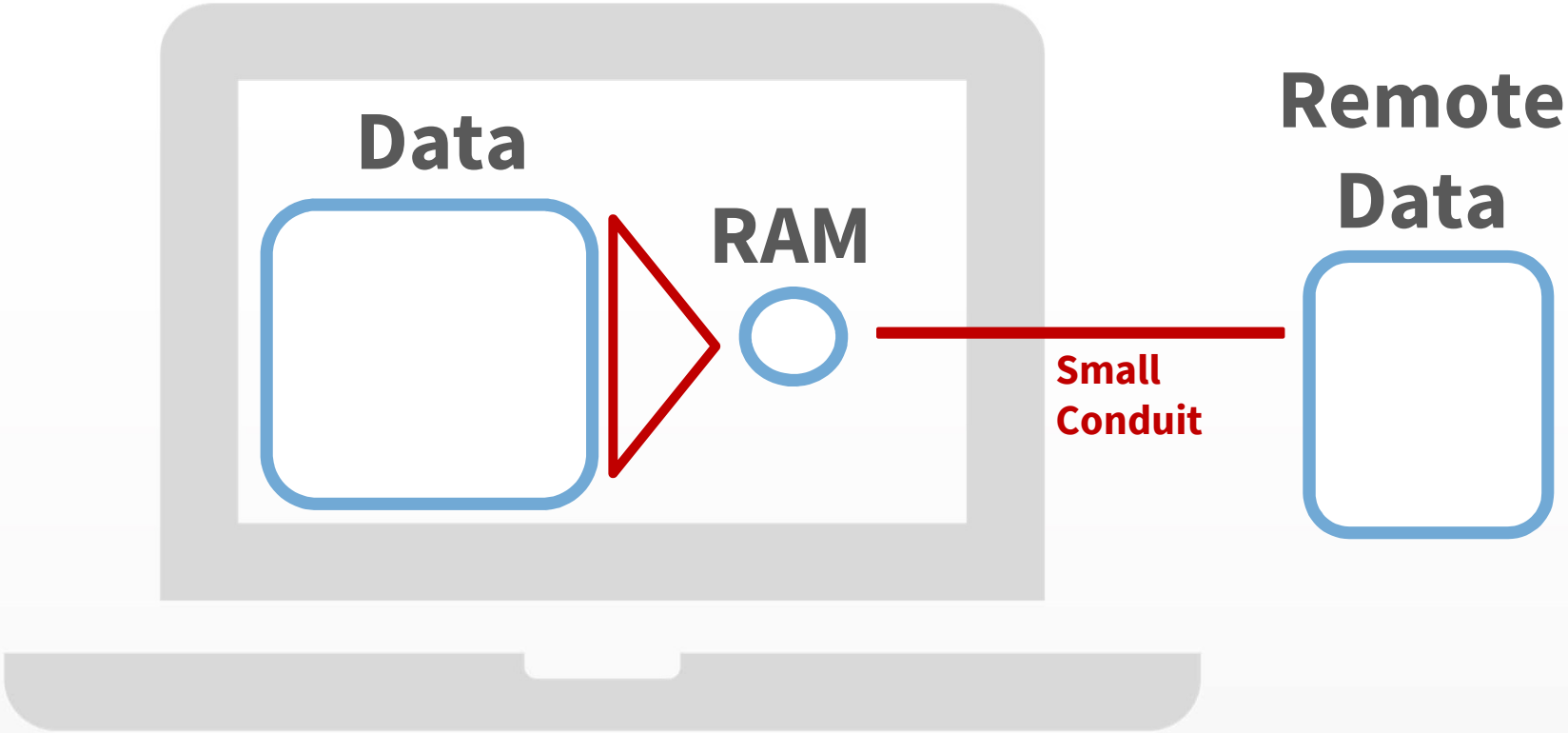
Data > RAM

Garrett Grolemund

Remote Data

Edgar Ruiz (circa 2018)

Big Data in R

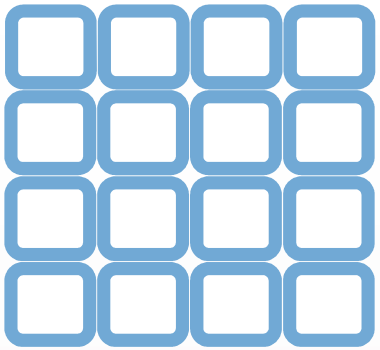


Big Data Strategies

Sample

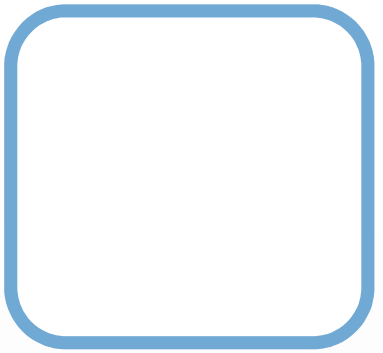
Most common approach for **modeling**

Parts



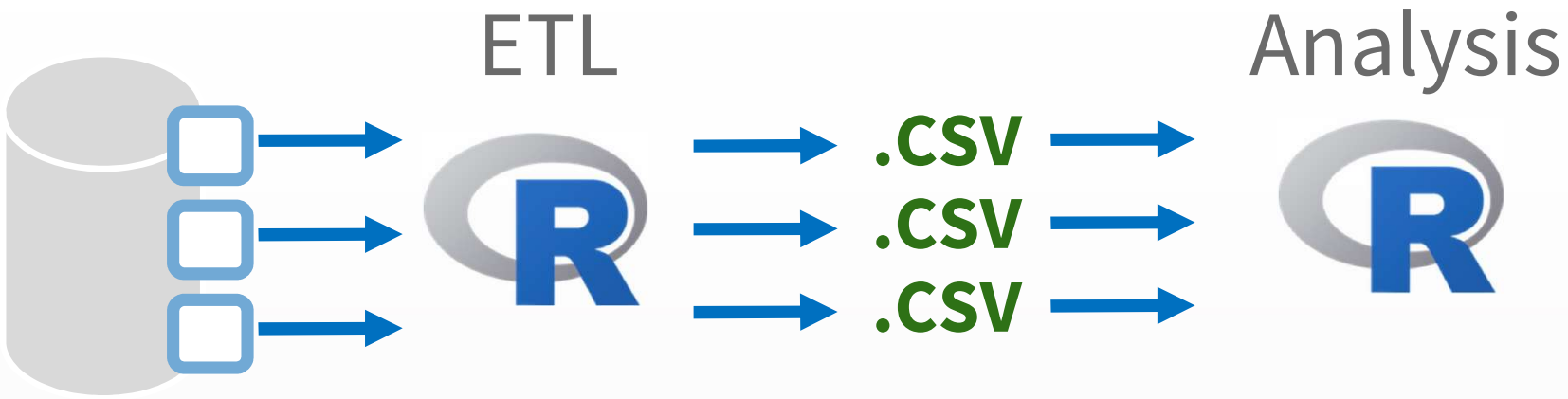
Most common approach for **general analysis**

Whole



In most cases, **the preferred approach,** it's just not feasible

Parts - “The Method”



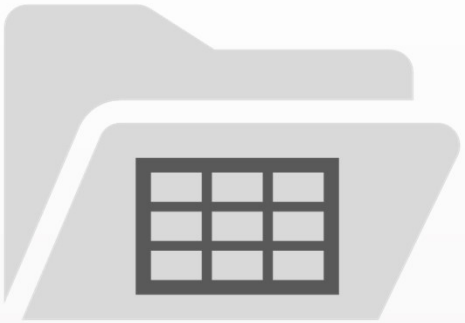
Typical DS project

Import Wrangle Learn Share

Remote Data Sources

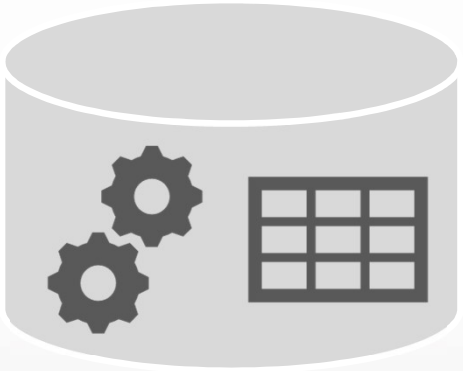
Flat Files

Only Data



Remote Sources

Data & Compute engine



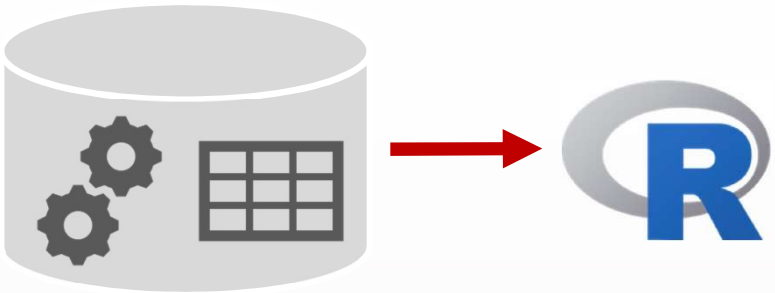
Unit 2 & 3

Using dplyr

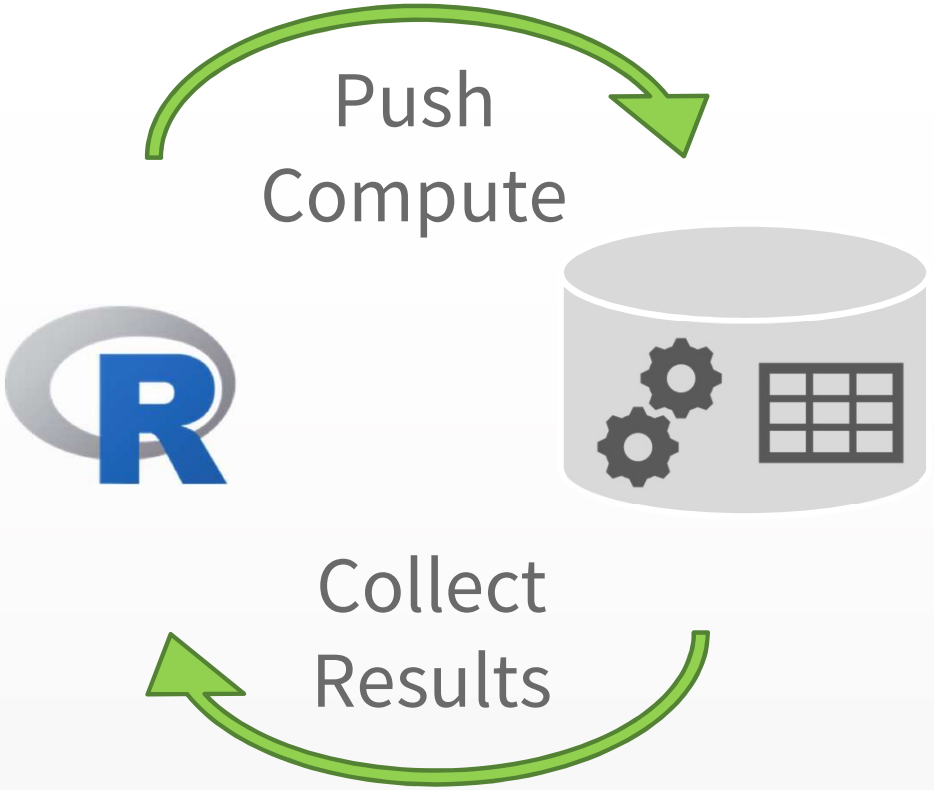
/dee-plier/

Photo by Arthur Lambillotte on Unsplash

Wrangle inside the DB



Extract Data



Options to Push Compute

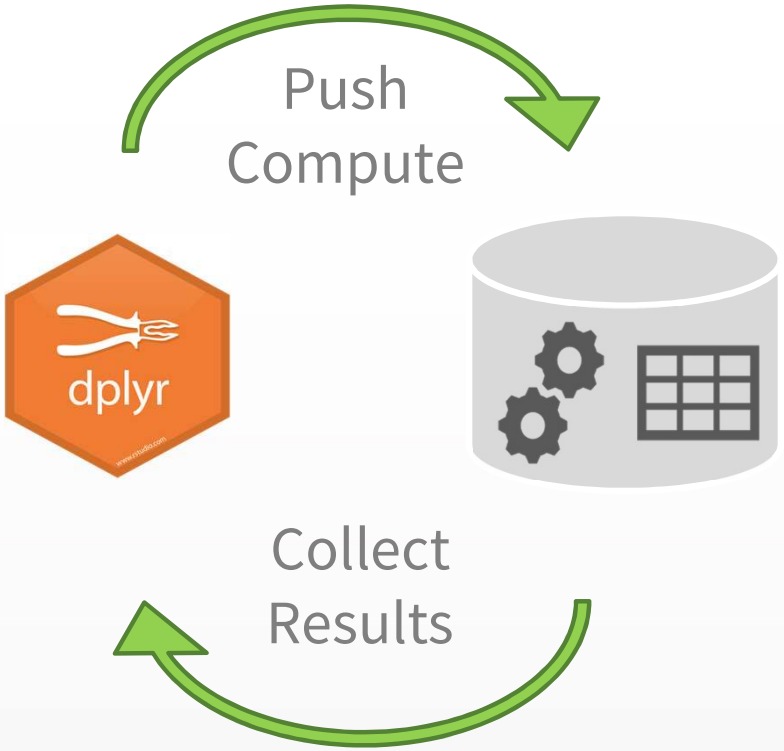
Write SQL statements

```
SELECT "name",
COUNT(*) AS "n"
FROM "vwFlights"
GROUP BY "name"
```

Use dplyr verbs

```
flights %>%
  group_by(name) %>%
  tally()
```

Advantages



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

Exercise 2.1 – 2.4

DS project using DBs

Access Wrangle Learn Share

In the database

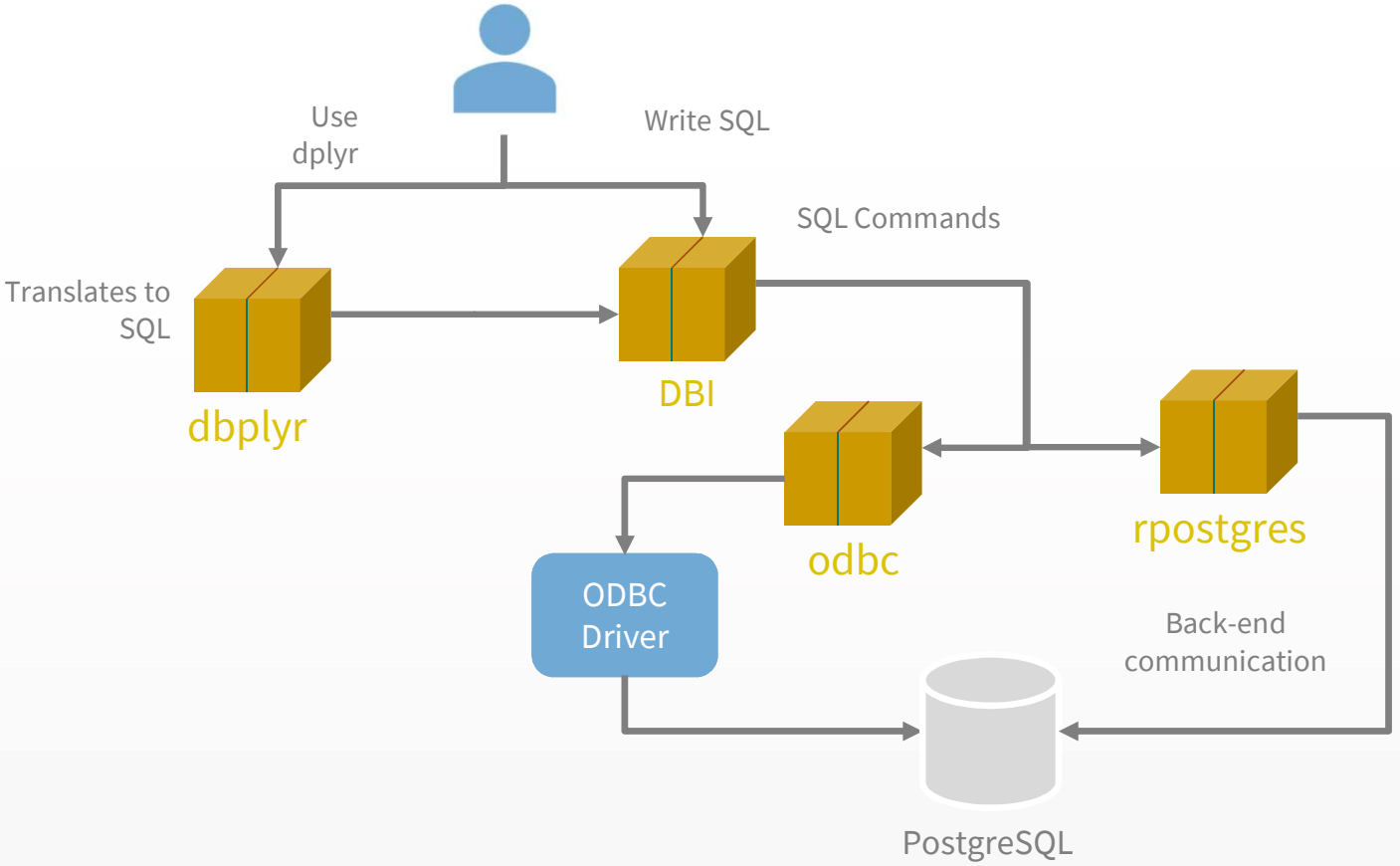
How to access a database

1. R Package – As implemented by *RPostgreSQL and others*
2. ODBC - As implemented in *odbc* package
3. JDBC - As implemented in *RJDBC* and other

Packages

1. dplyr – Simplifies data wrangling
2. dbplyr – Provides database specific translation
3. DBI – Common interface for Databases and R
4. DB R Package – Back-end interface for a specific database, such as RPostgreSQL
5. odbc – Back-end interface to a database using an ODBC driver

Architecture



Translations available in *dbplyr*

1. Microsoft SQL Server
2. Oracle
3. Apache Hive
4. Apache Impala
5. PostgreSQL
6. MS Access
7. MariaDB (MySQL)
8. SQLite
9. Amazon Redshift
10. Teradata

Exercise 3.1 – 3.5

Some advice…

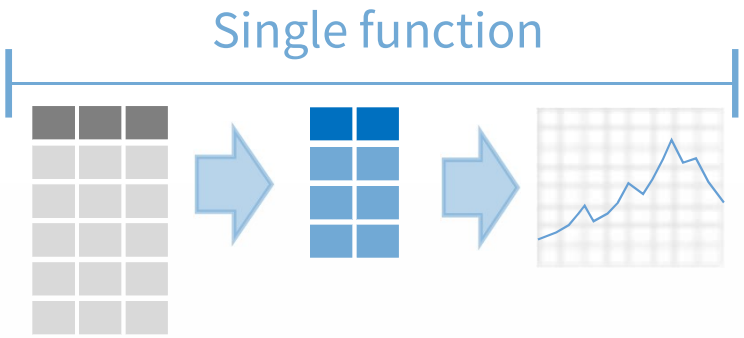
1. Think before you collect()
2. Just a bit off the top, use head()
3. Be select()ive of fields to bring back
4. tbl(con, "No SQL statements in tbl")

Unit 4
Visualizations

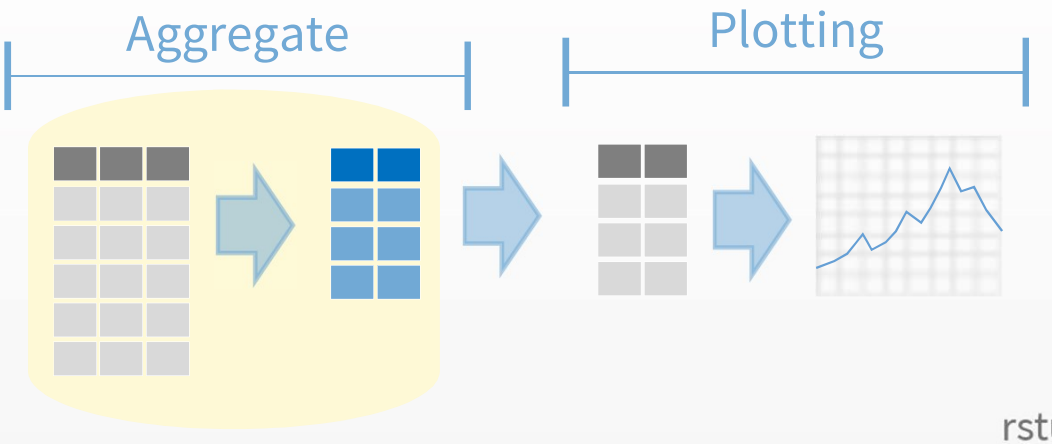
Photo by Luis Alfonso Orellana on Unsplash

Visualizations

Local data

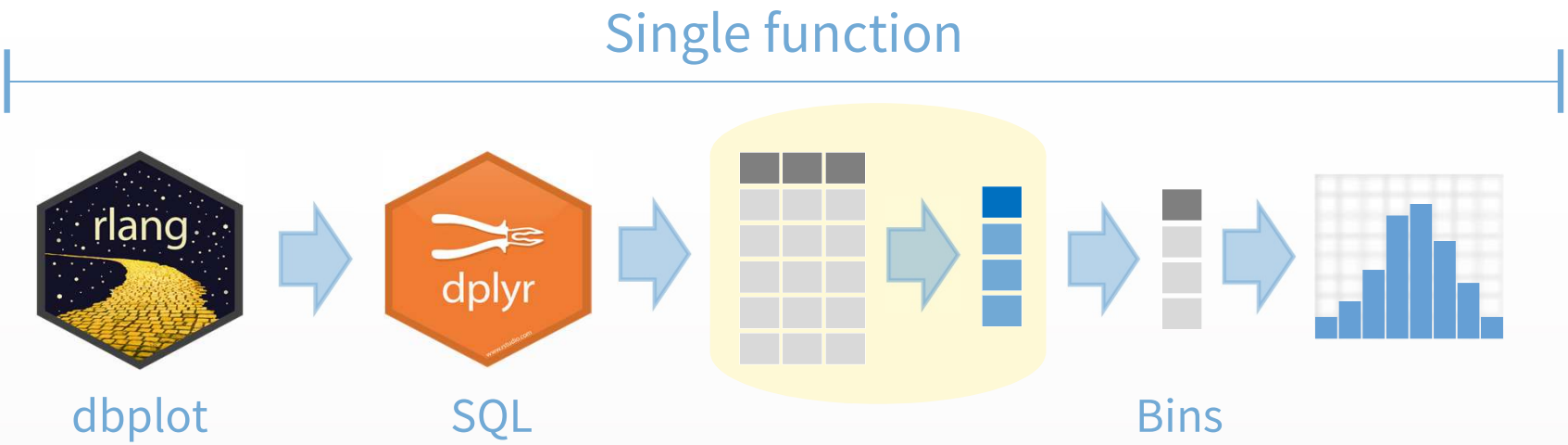


Remote data



Exercise 4.1 – 4.6

Complex plots



Exercise 4.7 – 4.10

Unit 5
Modeling

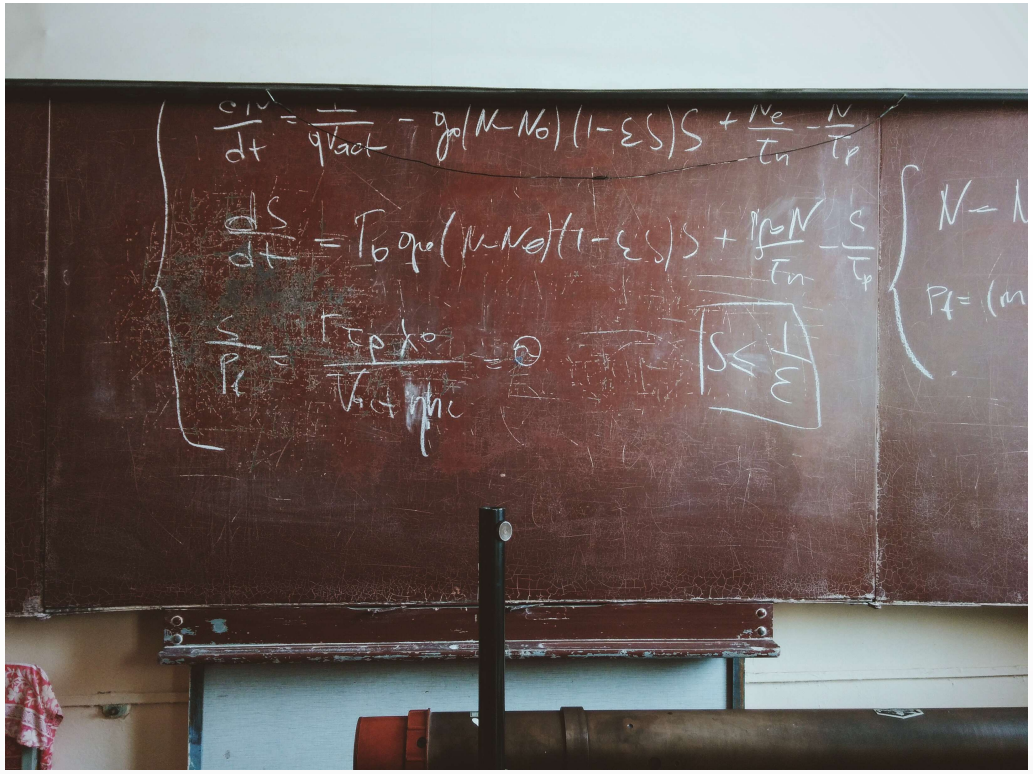
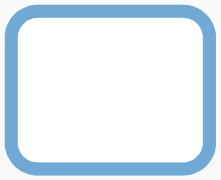
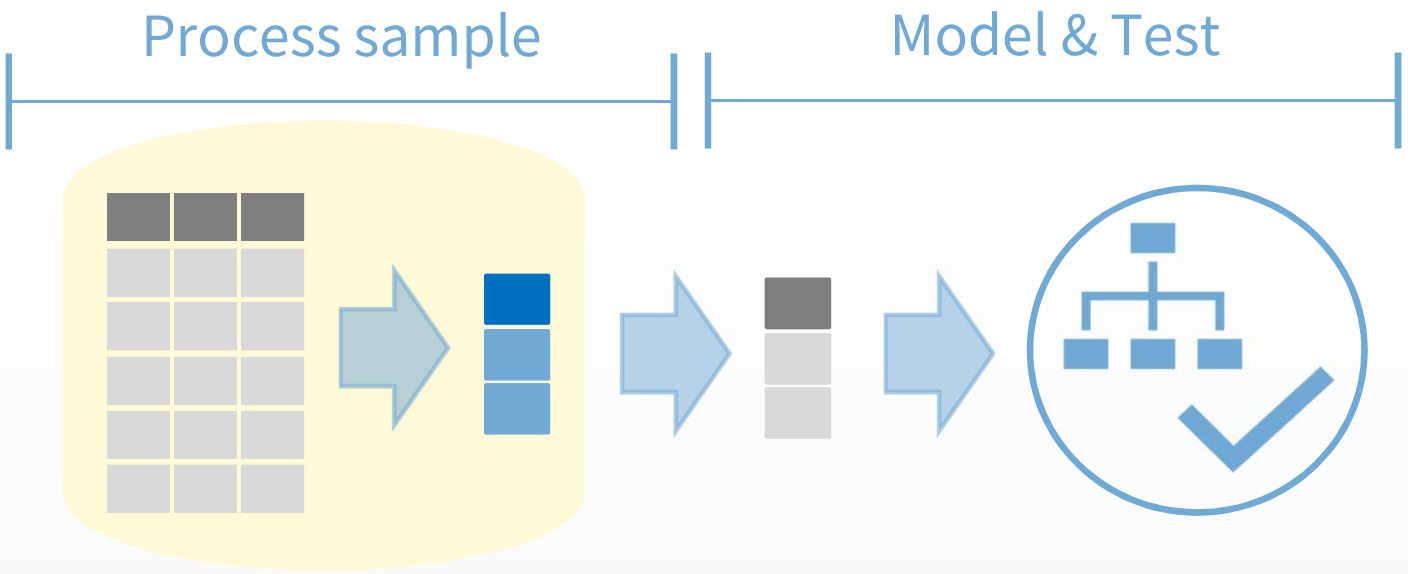


Photo by Roman Mager on Unsplash

Modeling scenario

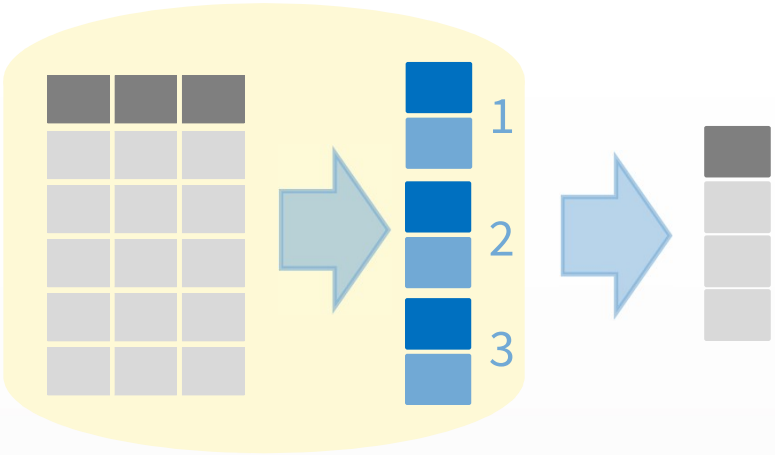
1. Training sample 
2. Model on sample 
3. Testing sample 
4. Verify model 
5. Score data 

Modeling with a Database



Exercise 5.1 – 5.2

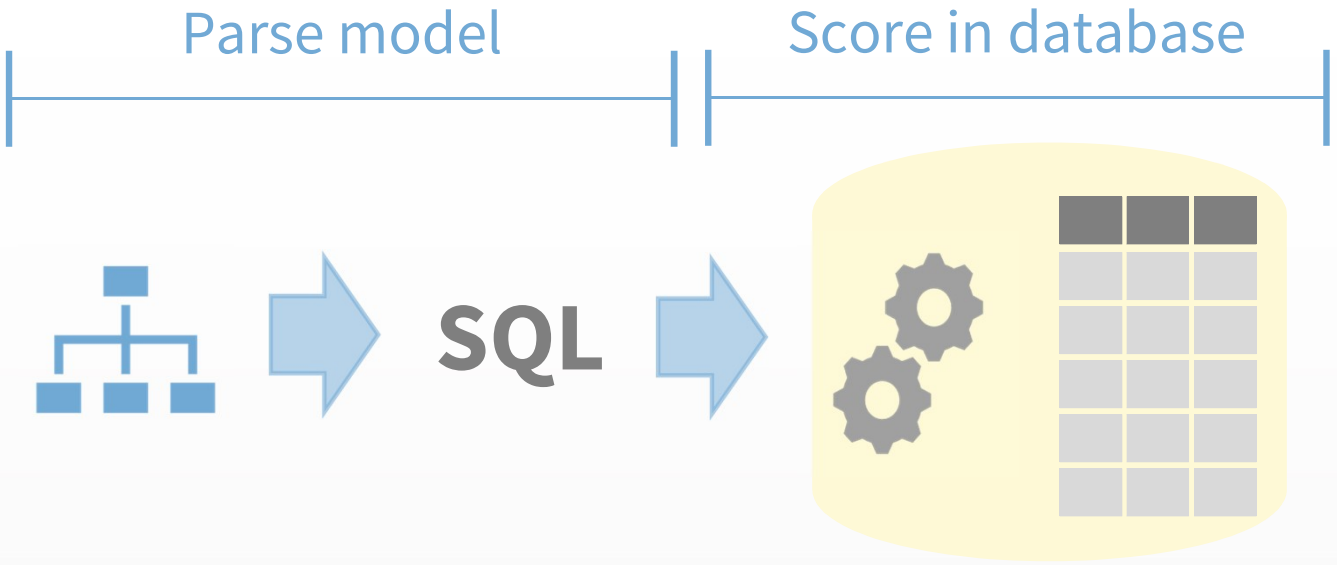
Multi-step sampling



Multiple SQL requests,
sample assemble in R

Exercise 5.2

Score inside the DB



Exercise 5.3 – 5.4

Units 7 & 6

sparklyr

/s-par-klee-r/

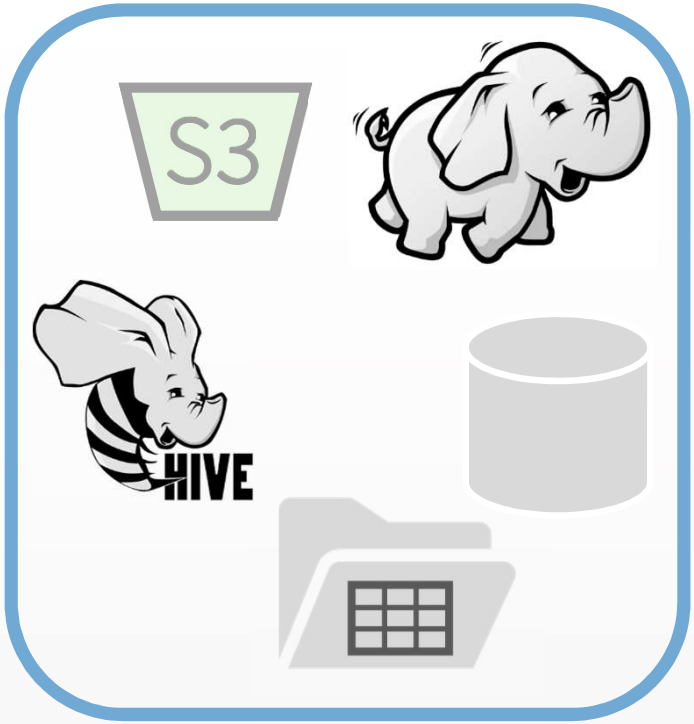
Photo by Matthew Ronder-Seid on Unsplash

What is Spark?

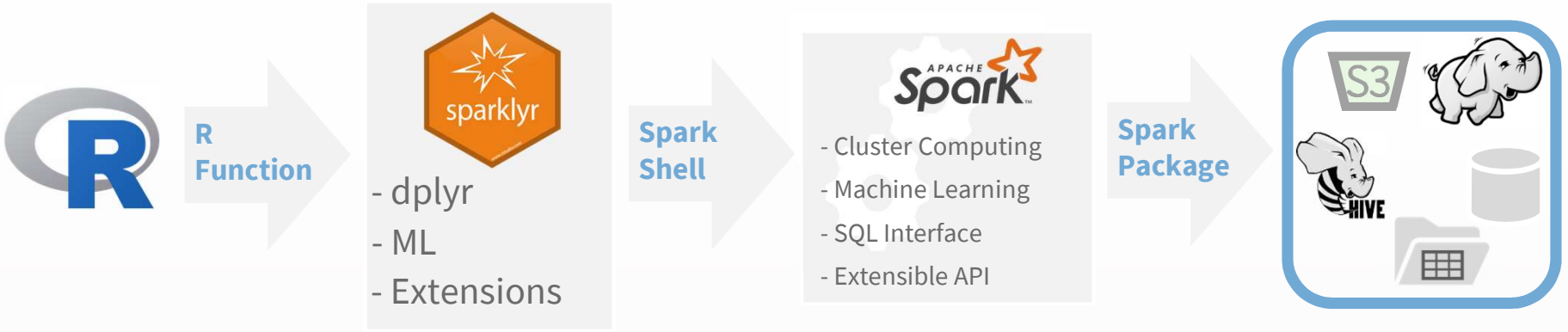
Processing

- Cluster Computing
- Machine Learning
- SQL Interface
- Extensible API

Storage



sparklyr – An R interface for Spark



Exercise 6.1 – 6.3

Working with data in Spark

Option 1

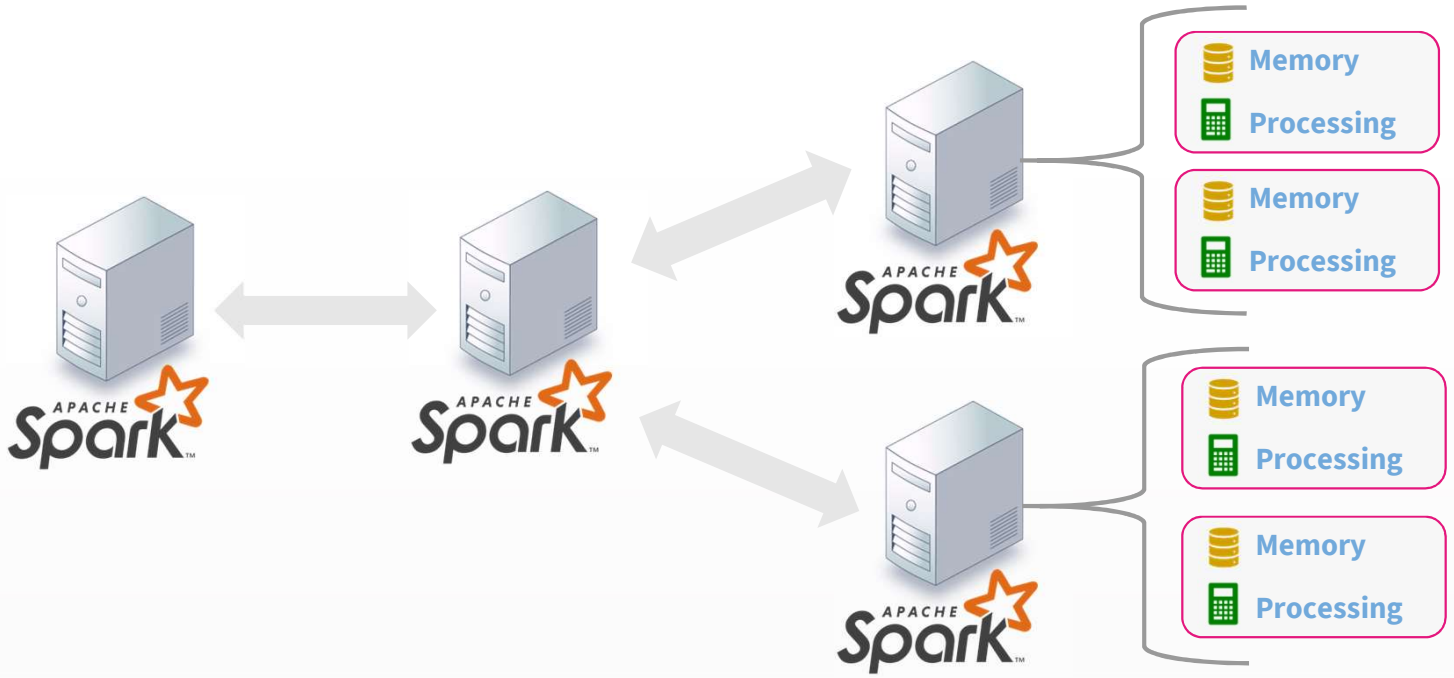
Use Spark as a pass-through for each query

Option 2

Cache the data into Spark memory & query there

Exercise 6.4 – 6.9

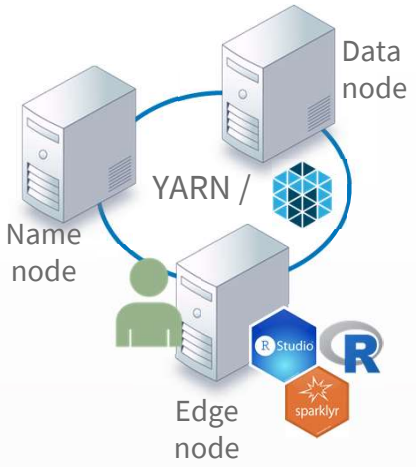
Typical architecture



Exercise 7.1 – 7.3

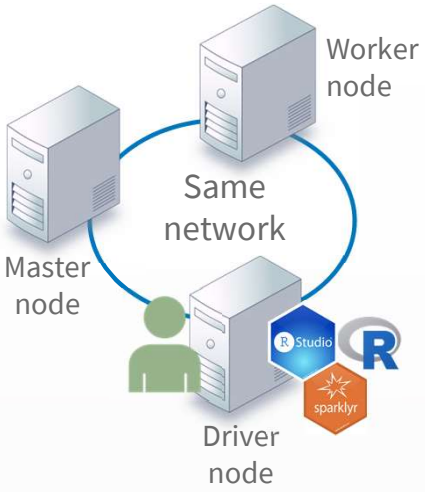
Deployment options

Managed Cluster



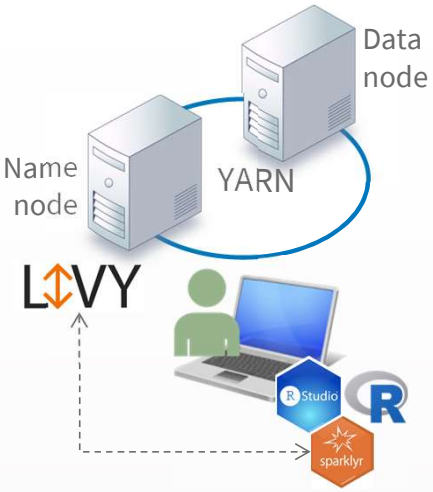
- Deployment seen at most business
- Spark version(s) available are limited to what's on the cluster

Stand Alone



- Since there's no central data repository, all data has to be either imported or connected to a common shared location (NAS, S3)

Livy



- Great for accessing a remote cluster
- Not recommended for Production deployments

Local

- Great for learning
- Works on Windows and Mac too
- Quick and easy way to access multiple cores

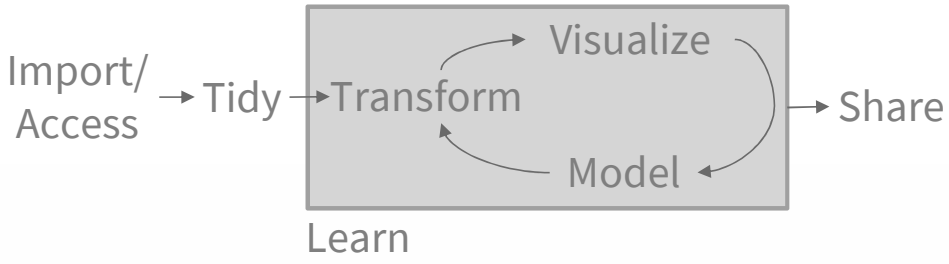
Let's talk about Data Science projects

Photo by Jo Szczepanska on Unsplash

Different deliverables

Data Science

- Deliverable: **Insights**
- Experimental
- Iterative



Production

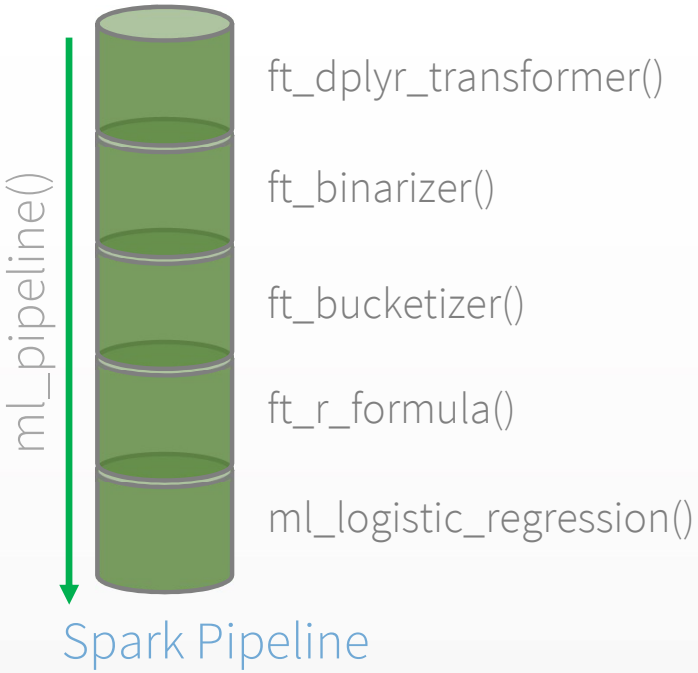
- Deliverable: **Software**
- Tested
- Automated
- Apply SDLC

Unit 8
Spark Pipelines

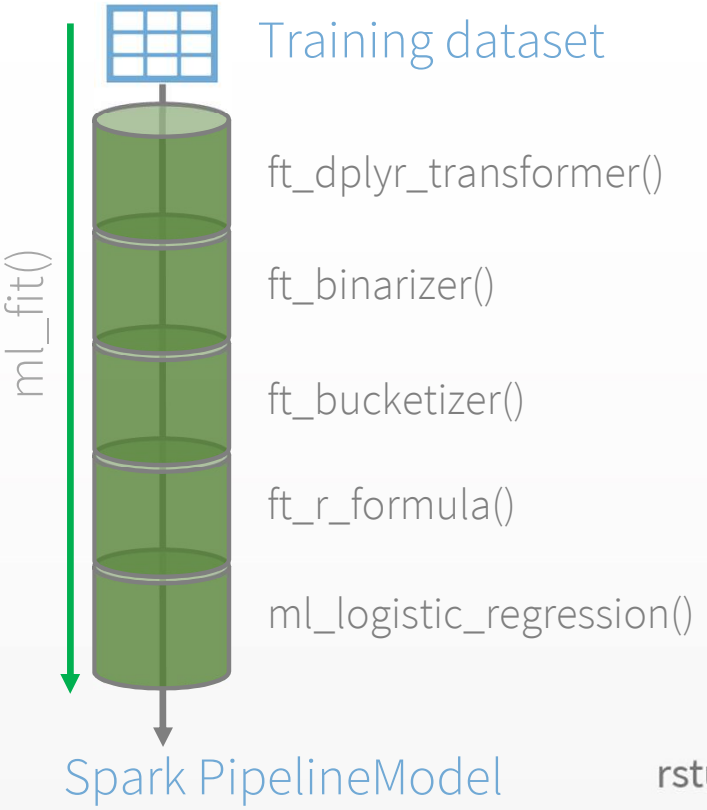
Photo by Iker Urteaga on Unsplash

Spark pipelines types

Estimator (Plan)

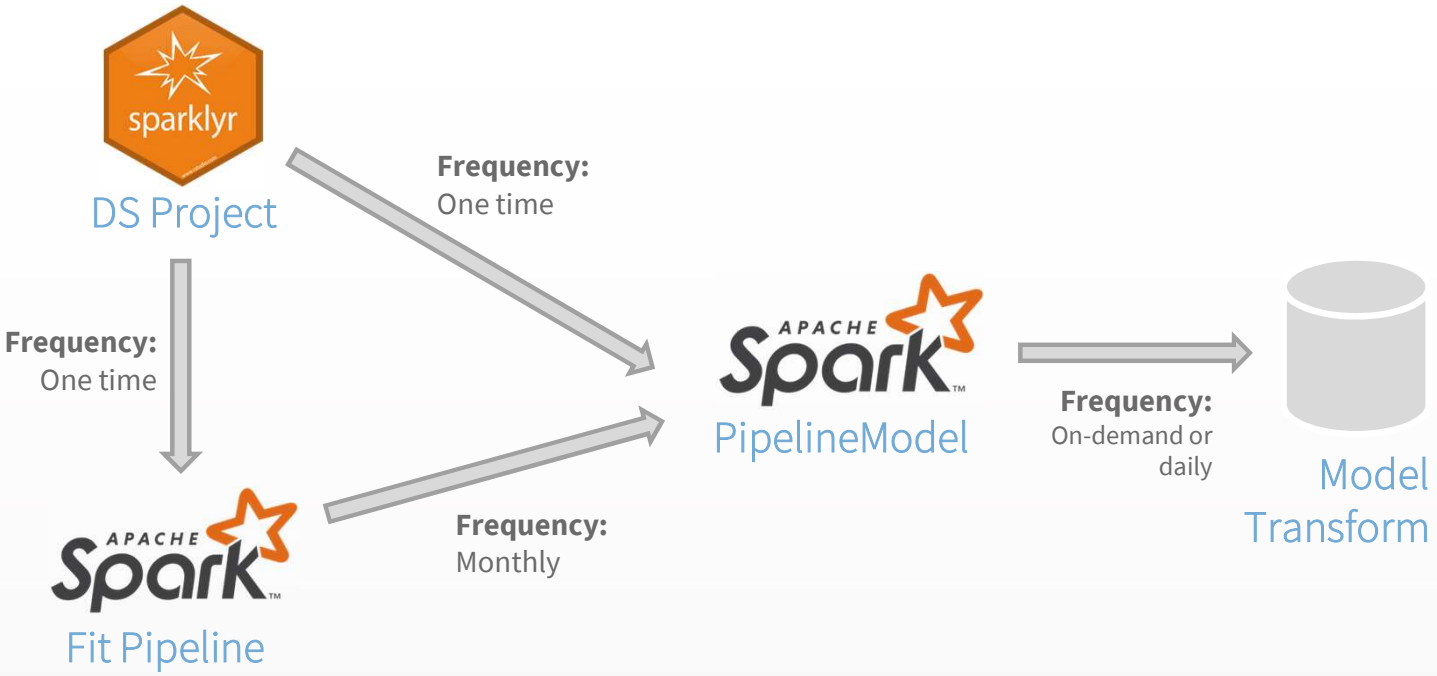


Transformer (Fit)



Exercise 8.1 – 8.4

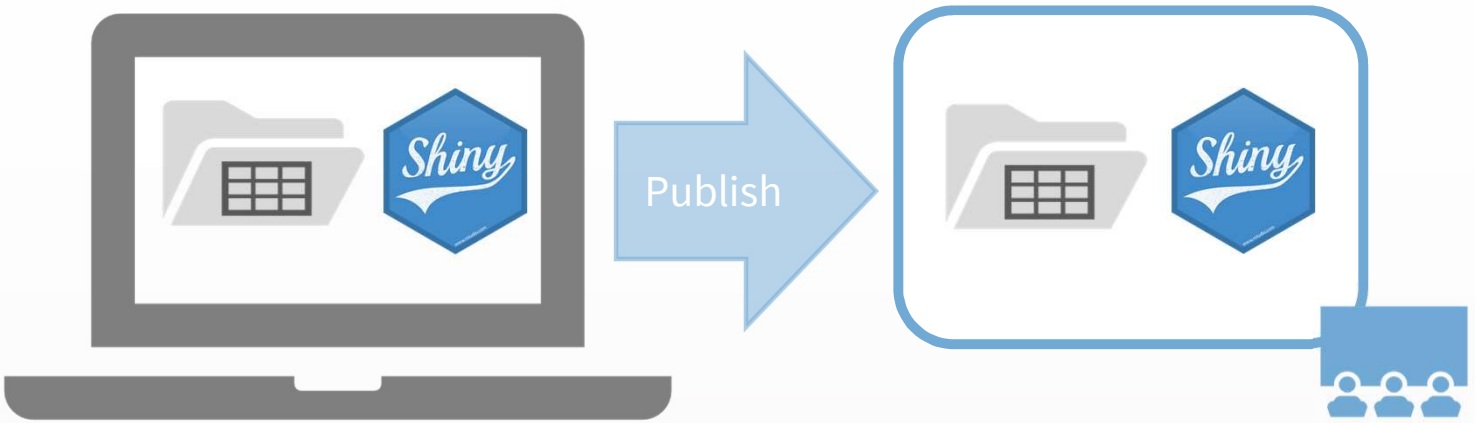
Production Implementation



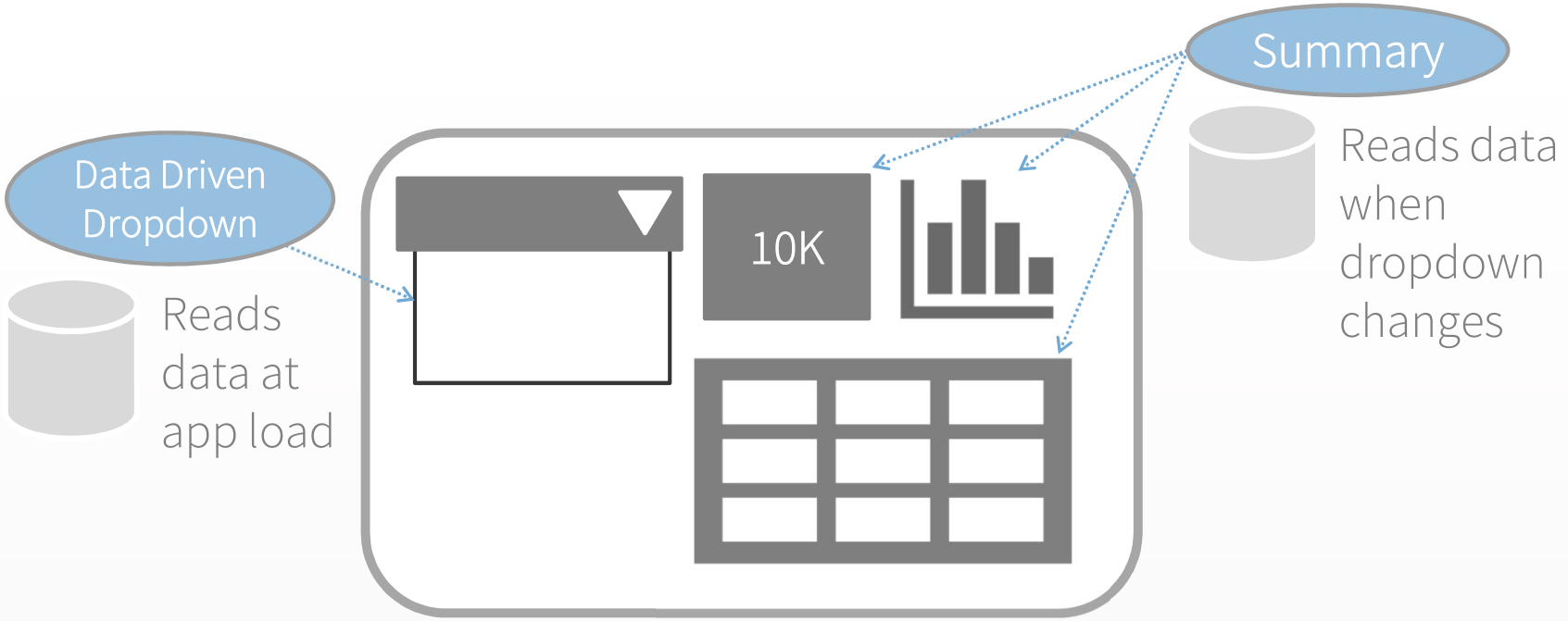
Units 9 & 10
Dashboards

Photo by Benjamin Child on Unsplash

Normal Shiny app

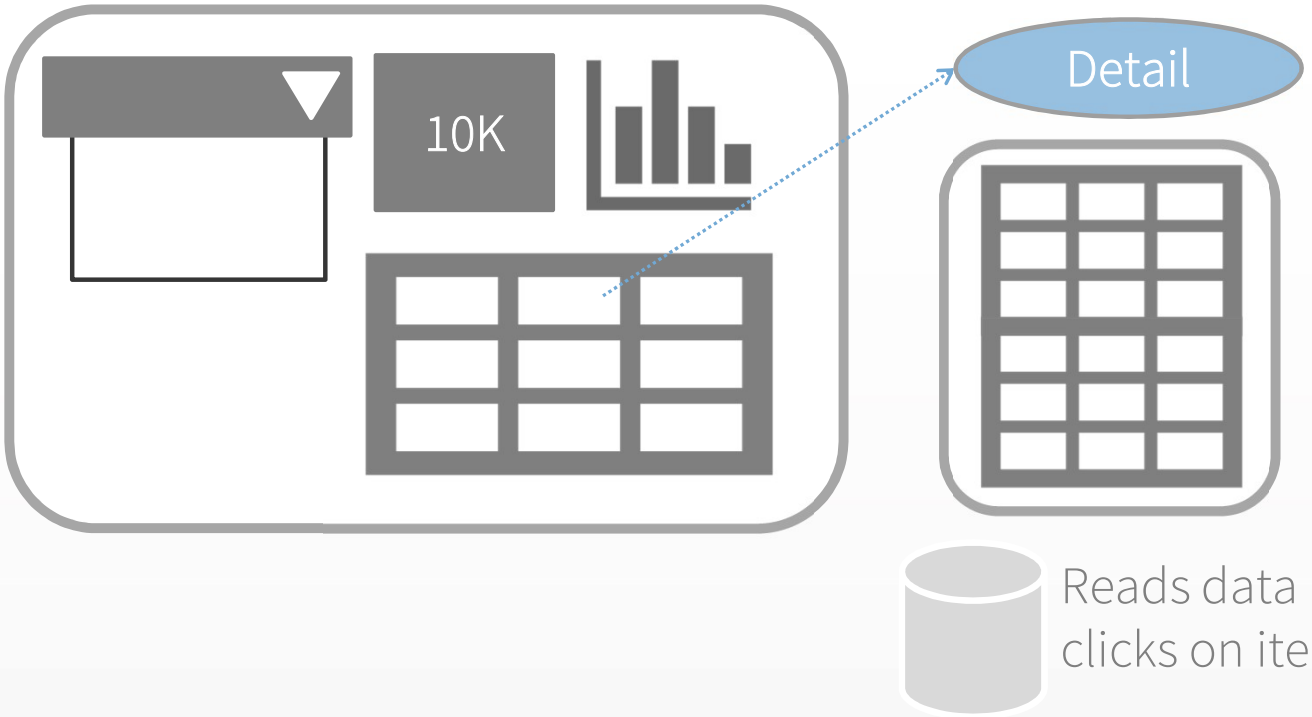


Database + Shiny Dashboard



Exercise 9.1 – 9.4

Database + Shiny Dashboard



Exercise 10.1 – 10.4

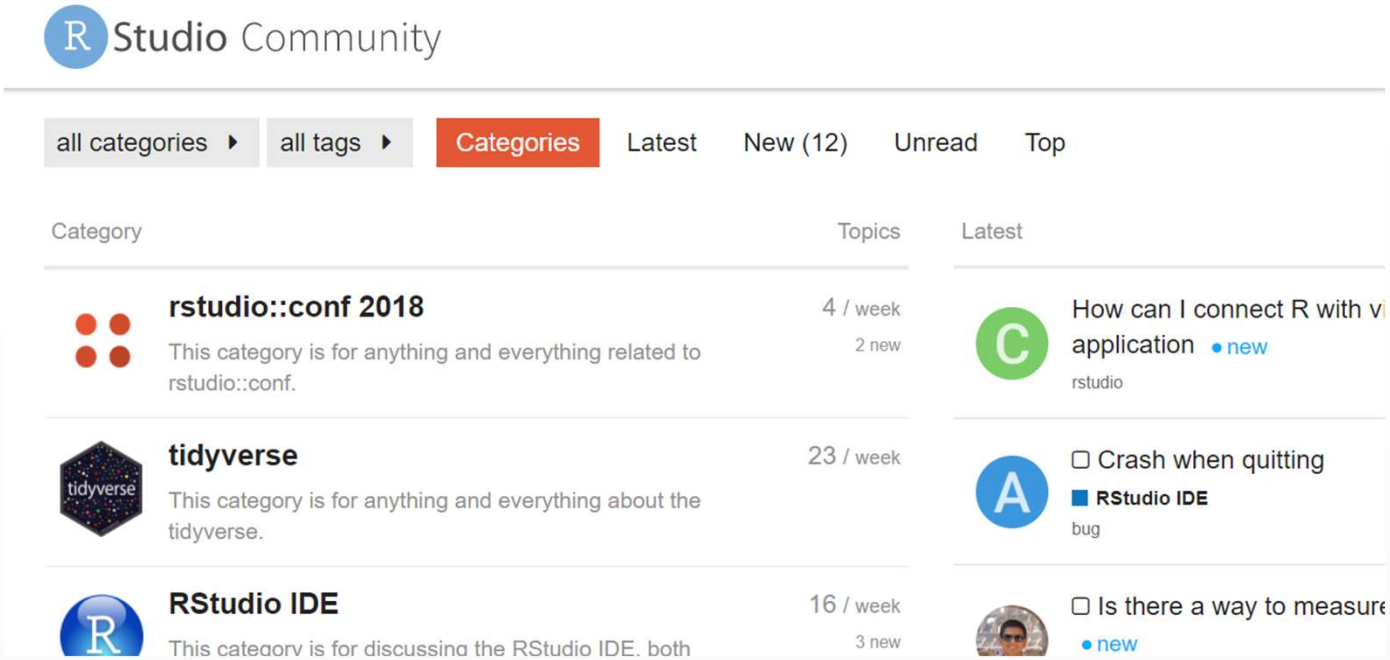
General advice

Photo by Daria Nepriakhina on Unsplash

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/

Join the community!



https://community.rstudio.com/

Familiarize yourself with the repos

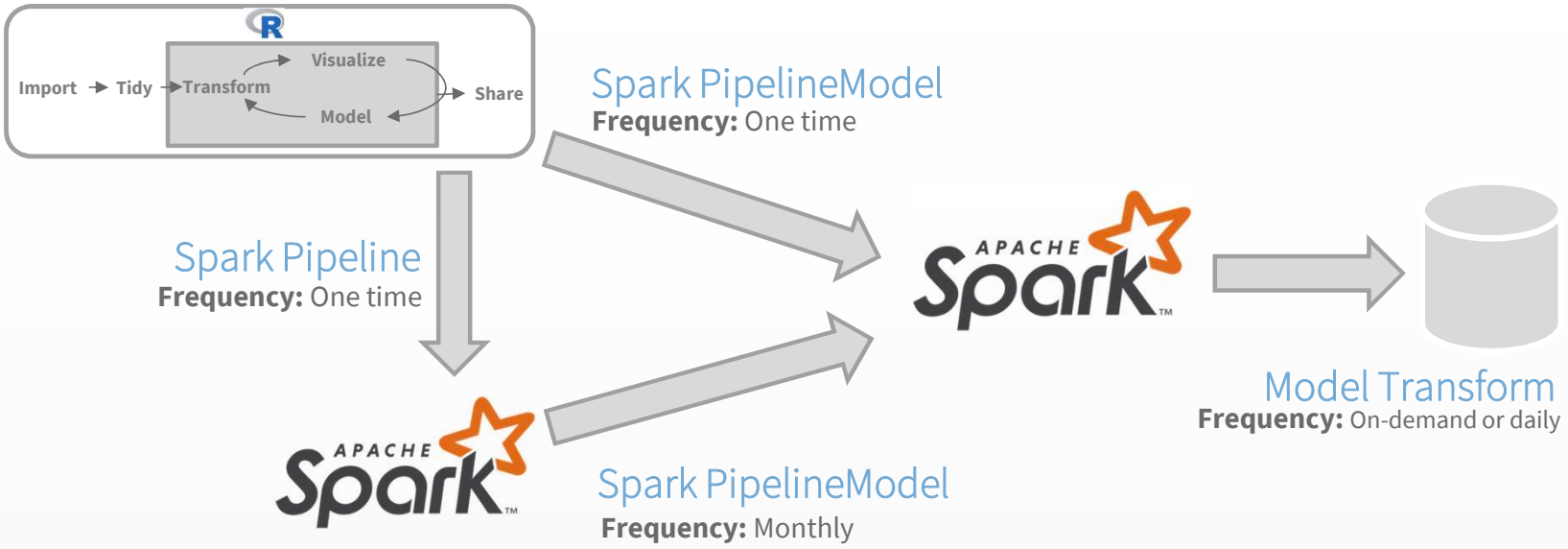
If I need to…	Check out
Report an issue or see if others are having the same problem	Issues
See if an feature exists or if it's coming up in future releases	NEWS
See the basics about the package	README

- 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/edgararuiz/tidypredict
- https://github.com/rstudio/sparklyr

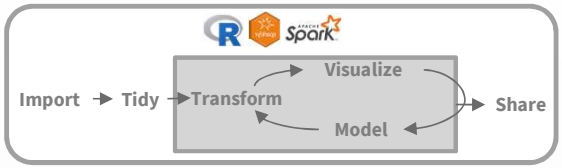
Thank you!!!!!

Photo by Gary Bendig on Unsplash

Production Implementation



Production Implementation



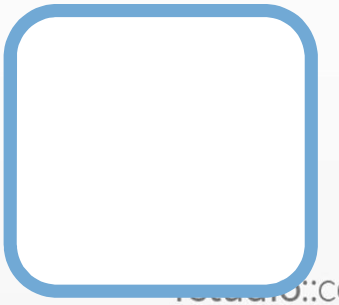
Spark Pipeline
Frequency: One time

Spark PipelineModel
Frequency: One time

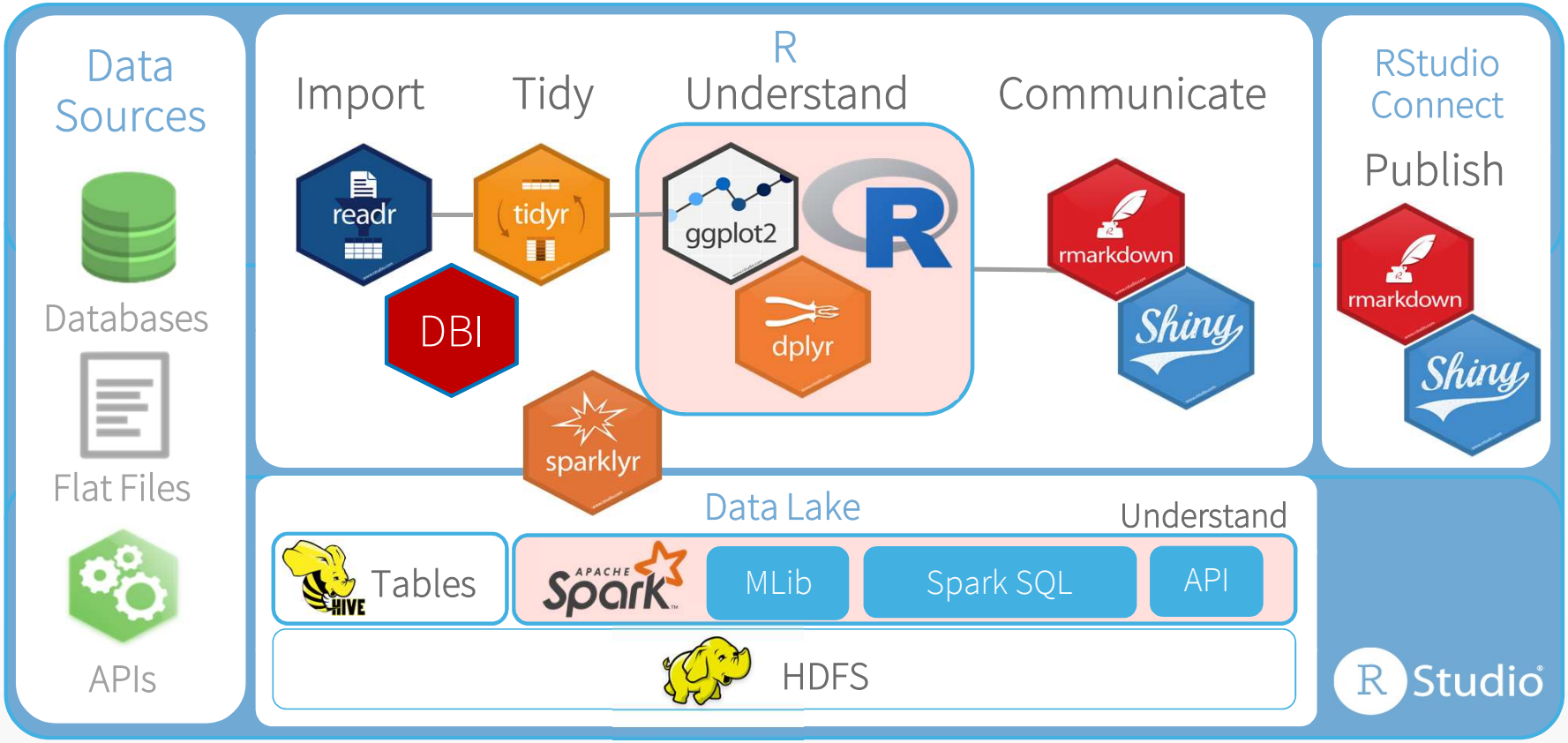
Model Transform
Frequency: On-demand or daily

Spark PipelineModel
Frequency: Monthly

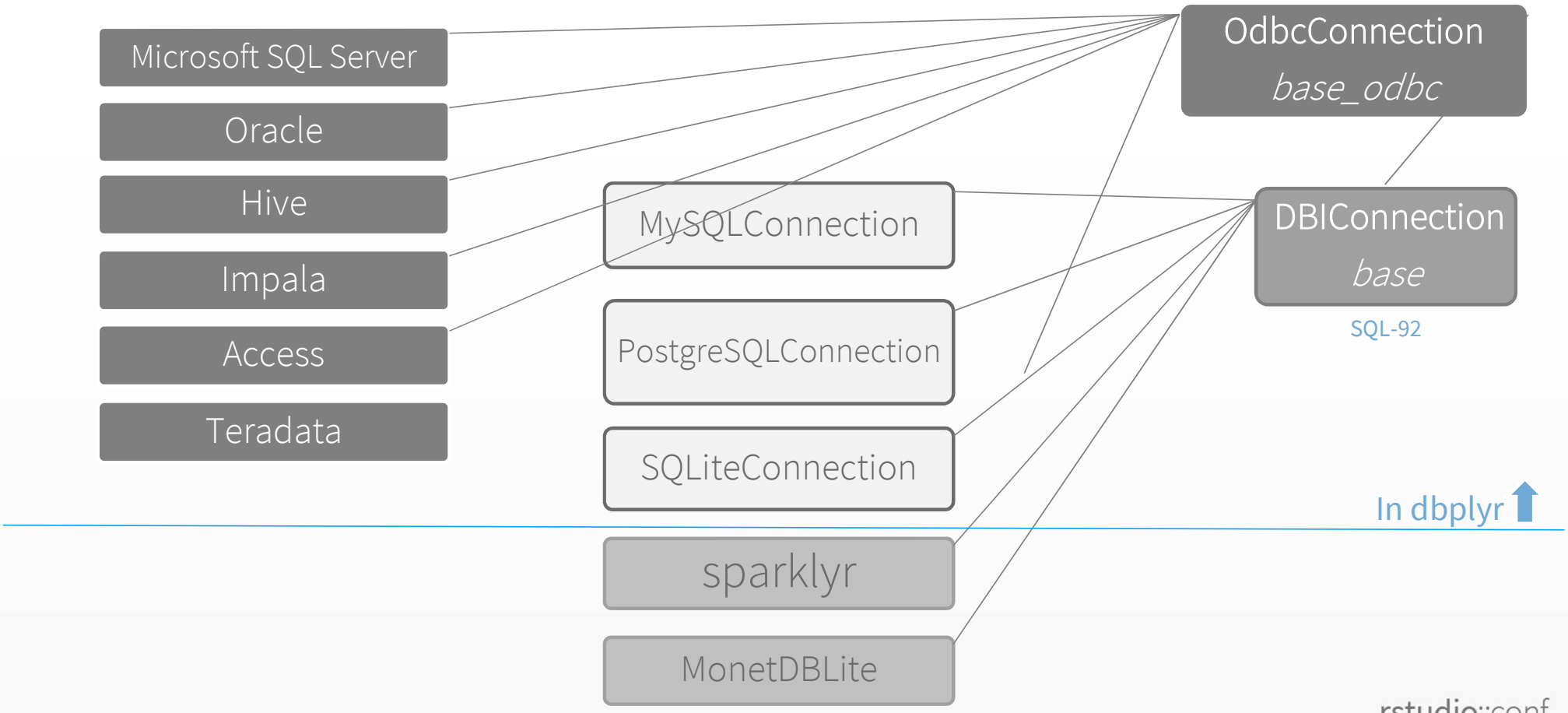
Spark pipelines



Toolchain with Spark



How translations work



Example translations

```
class(con)
[1] "Microsoft SQL Server"
attr(,"package") [1] ".GlobalEnv"
```

nchar()

```
nchar = sql_prefix("LEN")
```

```
nchar = sql_prefix("length", 1)
```

paste0()

```
paste0 = sql_prefix("CONCAT")
```

abs()

```
abs = sql_prefix("abs", 1)
```

SQL Clauses

sql_select – Orchestrates R to SQL clauses translation

```
class(con)
[1] "Microsoft SQL Server"
attr(,"package") [1] ".GlobalEnv"
```

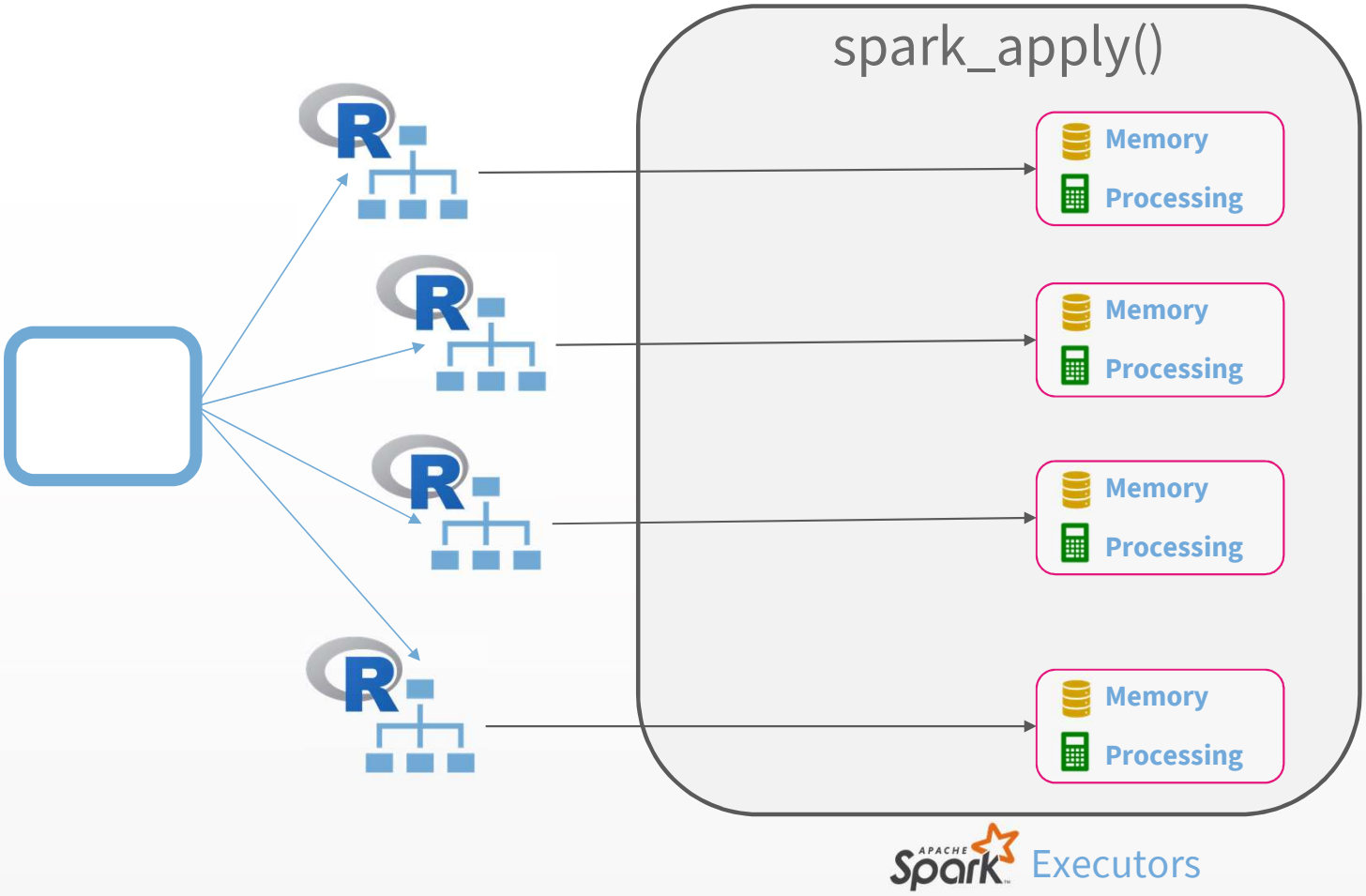
Microsoft SQL Server

↓ `sql_select.Microsoft SQL Server` ✓ SELECT TOP 10 * FROM table1

dbplyr's default

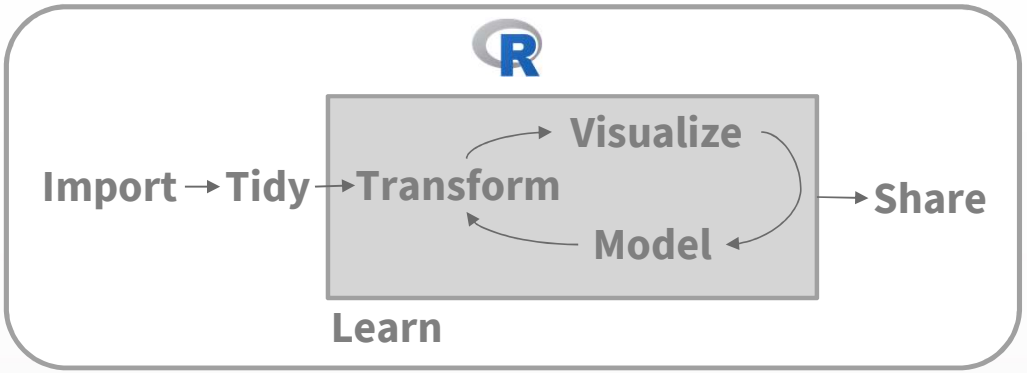
sql_select.DBIConnection SELECT * FROM table1 LIMIT 10

Parallel Problems? Distributed R



Think Big Picture

R for Data Science



Frequency: One time

R in Production

Scheduled Model Fitting

Frequency:
Monthly

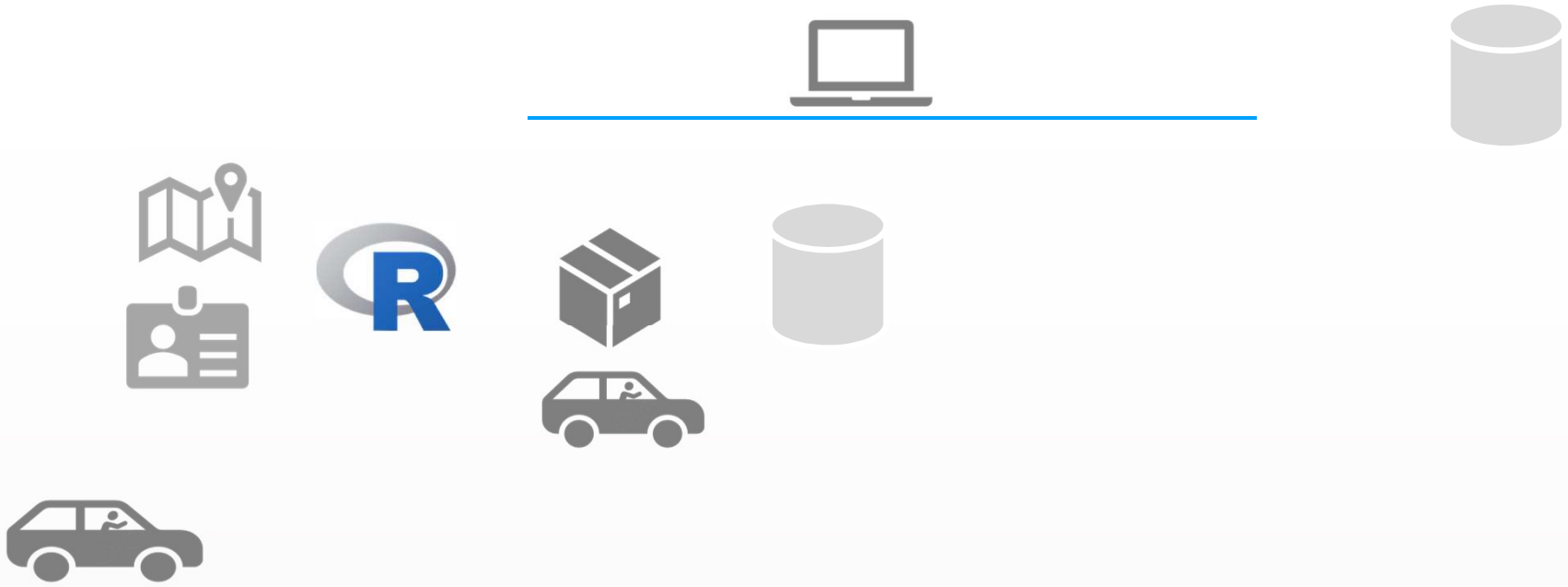
Report

DB Write back

API

Frequency:
On-demand or daily

Connecting to a Database

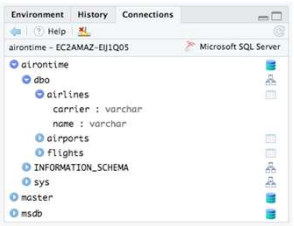


DSN

Database Drivers

R Package

RStudio's approach to Databases

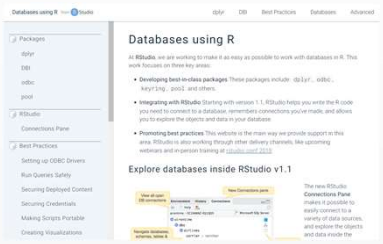


1. RStudio v1.1 Integration

- View databases, schemas, tables, and fields
- Explore data in tables or views
- Remembers connections you've made

2. Utilize best-in-class packages

- dplyr
- odbc
- DBI



3. Promoting best practices

- db.rstudio.com
- Training & presentations
- Blog posts (rviews.rstudio.com)

BIG data?

Velocity

Volume

Value

Variety

Veracity

Out-of-memory Data