Databases using R

Edgar Ruiz

@theotheredgar
linkedin.com/in/edgararuiz

November 2017

RStudio

#ODSC

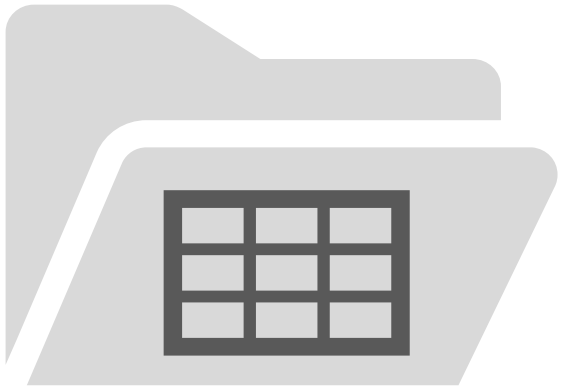
Typical DS project

Import Wrangle Learn Share

Databases vs Flat files

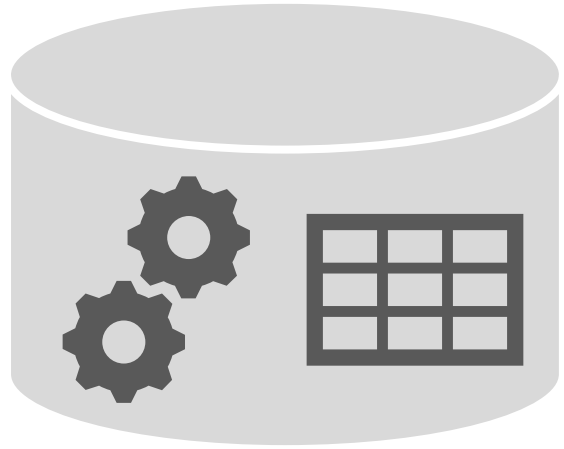
Flat files

Data only

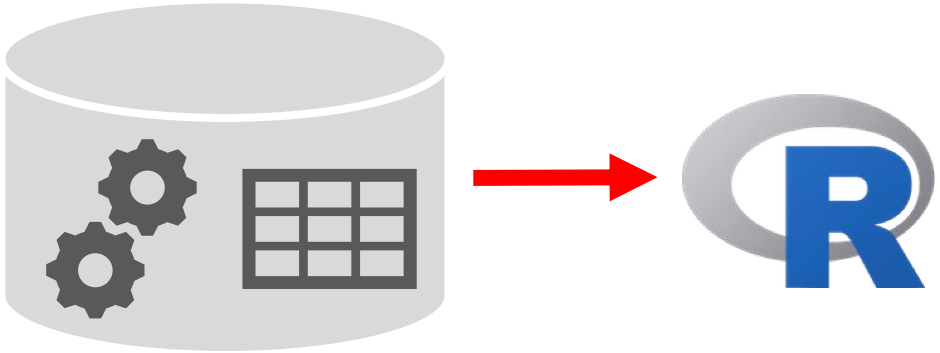


Databases

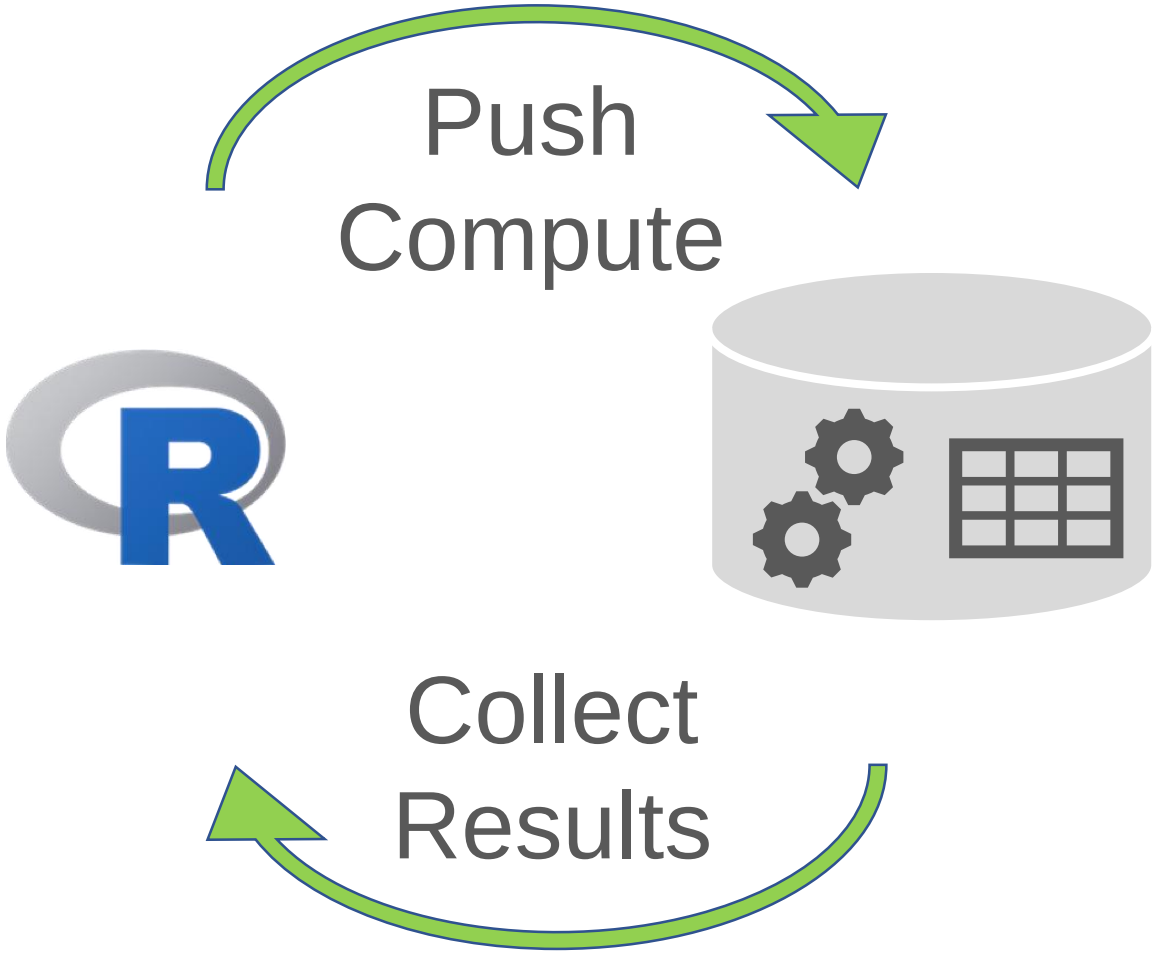
Data & SQL engine



Wrangle inside the DB



Extract
Data



Options to Push Compute

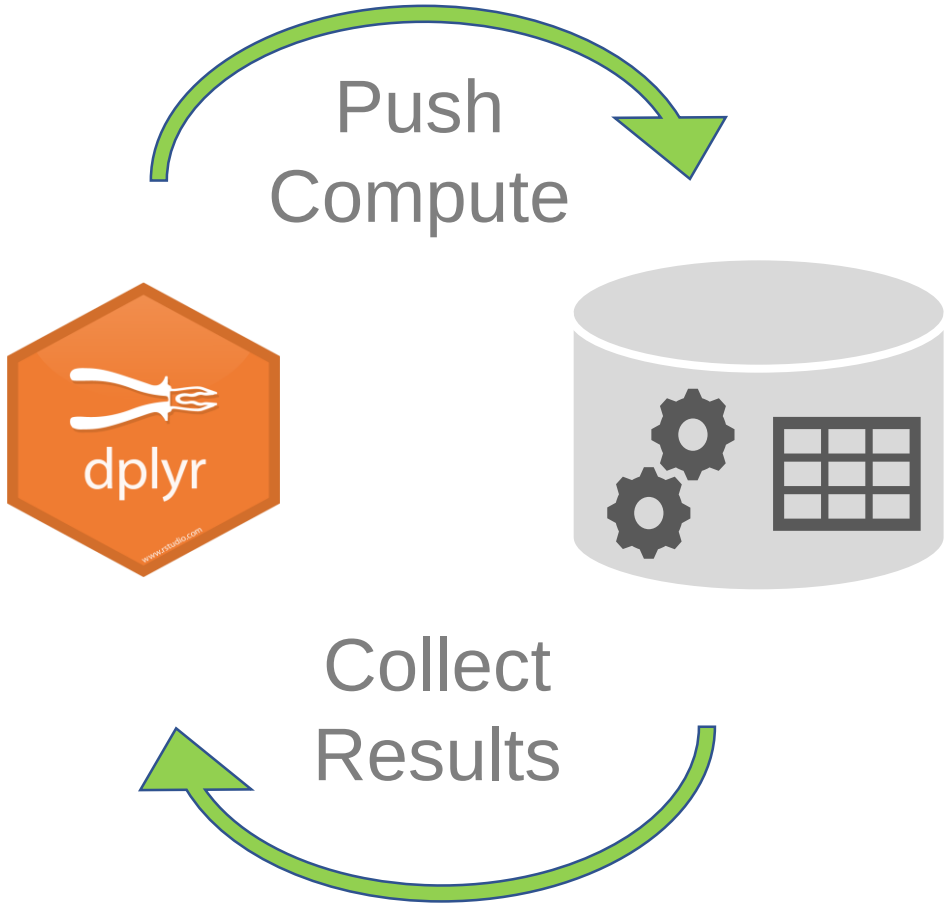
Write SQL statements

```
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!

DS project using DBs

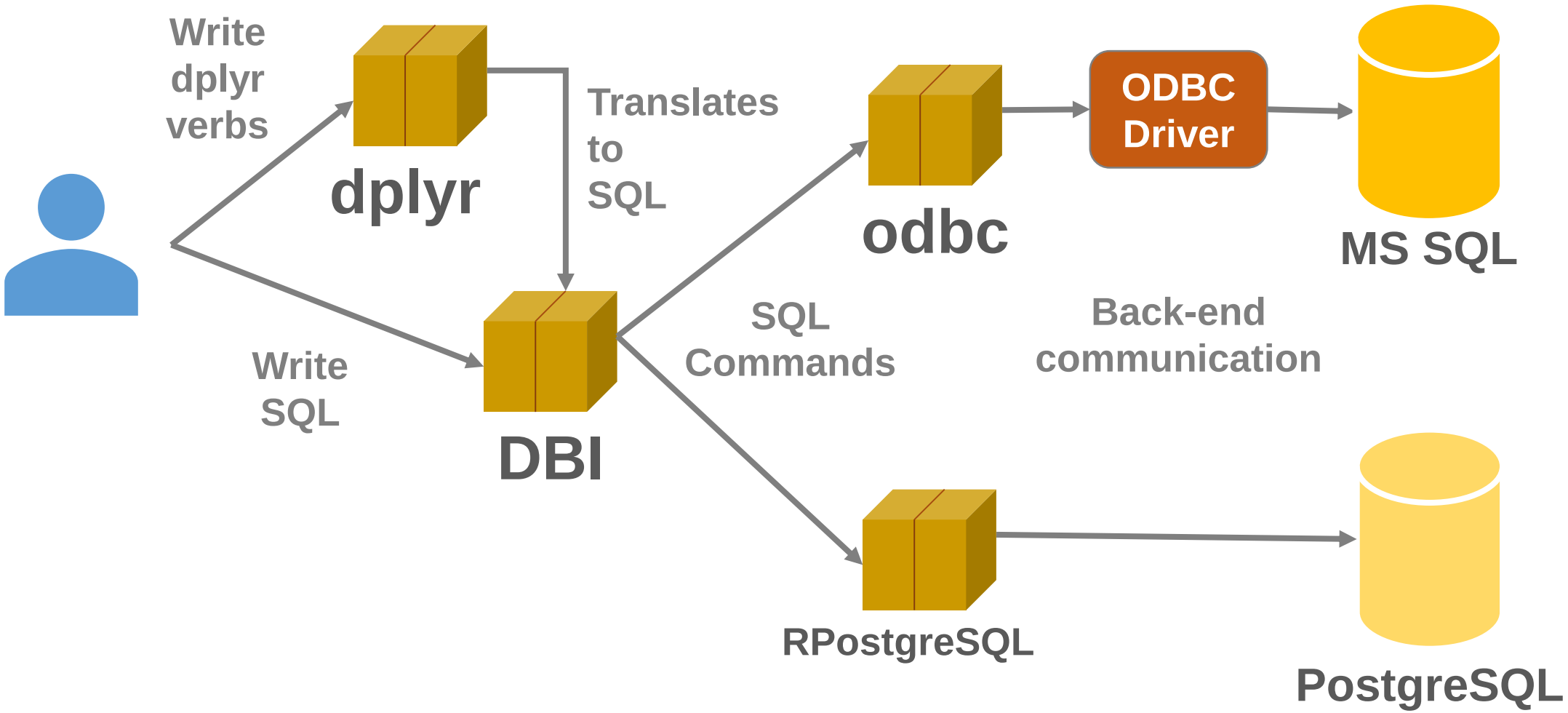
Access Wrangle Learn Share

In the database

Packages

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

Architecture



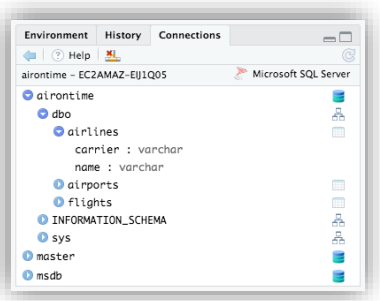
Translations available in *dbplyr*

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

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

RStudio's approach to Databases

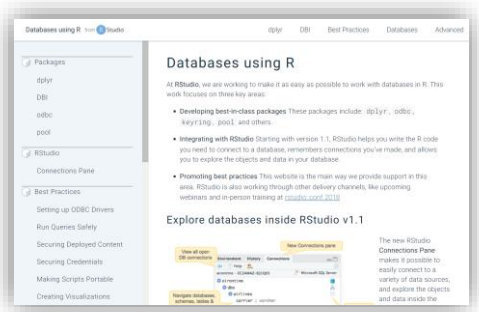


1. RStudio v1.1 Integration

- 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)

RStudio