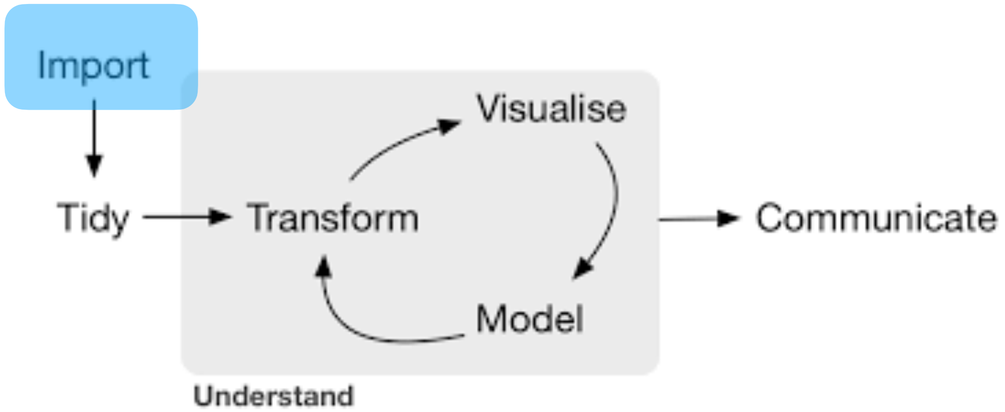
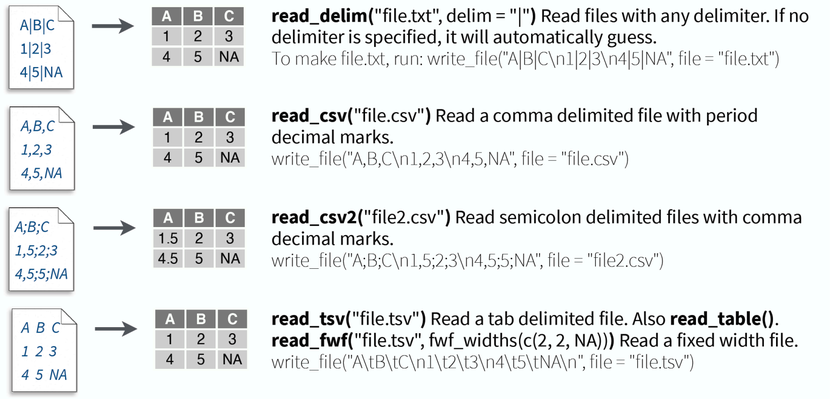
dbcooper: Turn any database into an R package

David Robinson

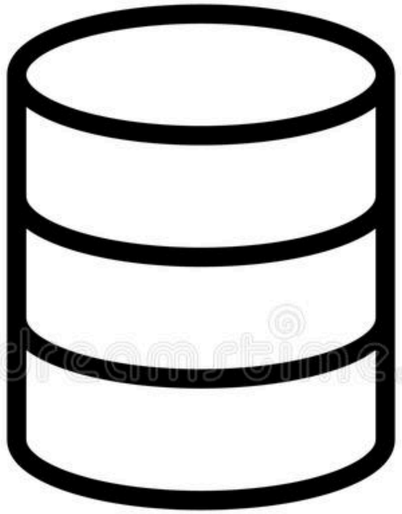
2021-09-10

A lot of the EDA flow is built around reading data into memory

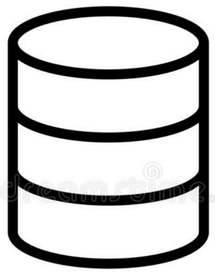




Most organizations work with data in a database



Working with a db should be as easy for your team as possible



Build a Corporate R Package for Pleasure and Profit

Brad Lindblad Mar 5, 2019 · 9 min read

Using R packages and education to scale Data Science at Airbnb

AirbnbEng Follow
Mar 29, 2016 · 9 min read

By Ricardo Bion

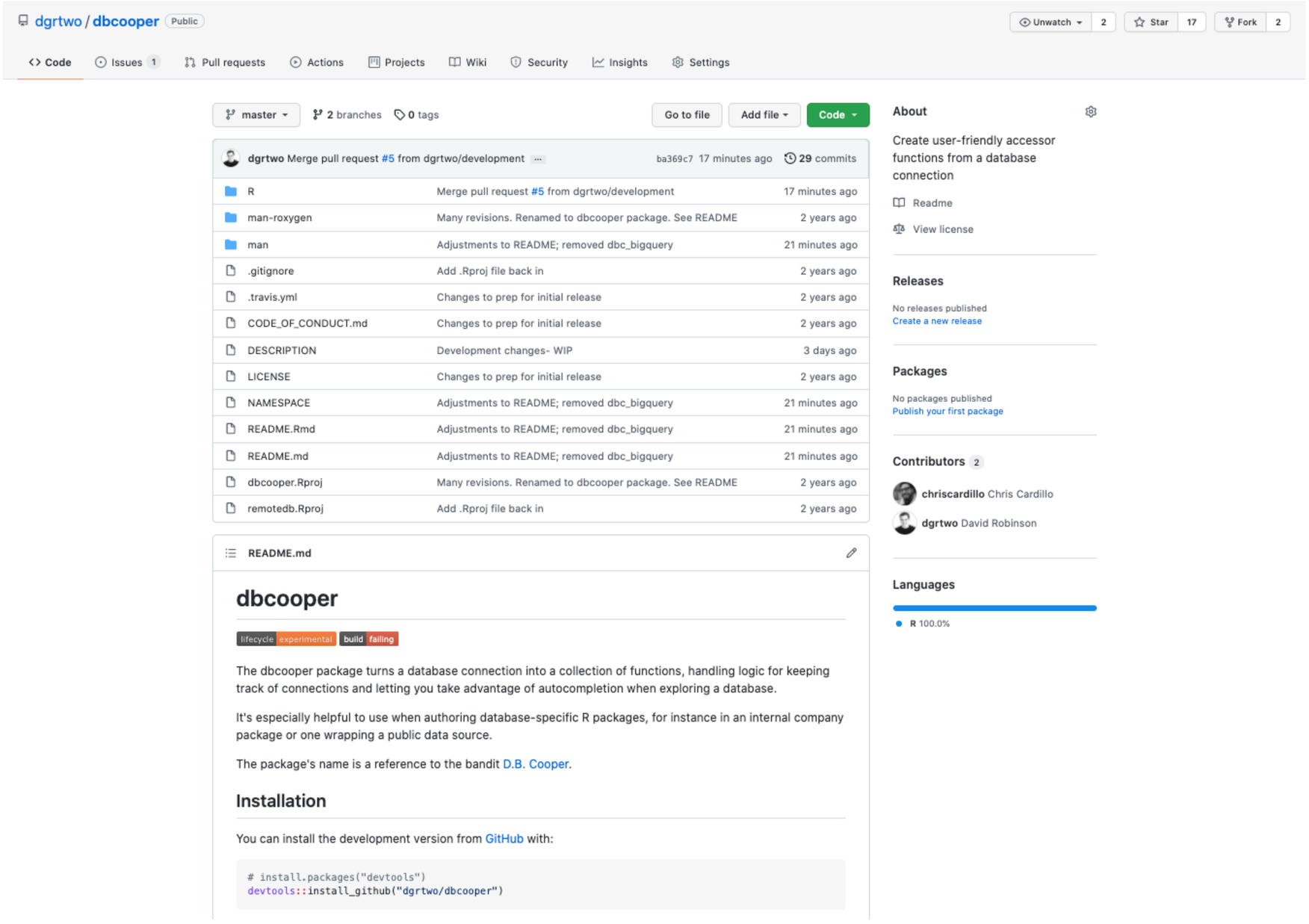
oRganization

How to make internal R packages a part of your team

Emily Riederer @emilyriederer emily.rbind.io

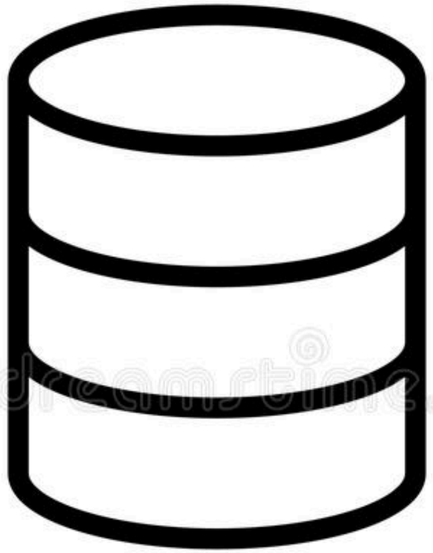
I first started appreciating internal packages when I was working as a data scientist at Stack Overflow

```
library(sqlstackr)
```



https://github.com/dgrtwo/dbcooper

The goal of dbcooper: one step to a database-specific package



dbcooper

Widgets Inc
Database

widgetr
Package

Problem:
What's it like to work with a database?

What do you want to do when analyzing a database?

- List tables
- Work with tables using dplyr
- Run SQL queries
- Execute commands

Example: Creating a Connection

```
library(dplyr)
library(dbplyr)

lahman_db <- lahman_sqlite()

lahman_db
#> <SQLiteConnection>
#>   Path: /var/folders/8p/xzrrqphx2qb3d2s_fgqrk5xr0000gn/T//Rtmp9OcGbM/lahman.sqlite
#>   Extensions: TRUE
```

Example: List tables

```
DBI::dbListTables(lahman_db)
#>  [1] "AllstarFull"          "Appearances"          "AwardsManagers"
#>  [4] "AwardsPlayers"        "AwardsShareManagers" "AwardsSharePlayers"
#>  [7] "Batting"              "BattingPost"          "CollegePlaying"
#> [10] "Fielding"             "FieldingOF"           "FieldingOFsplit"
#> [13] "FieldingPost"         "HallOfFame"           "HomeGames"
#> [16] "LahmanData"           "Managers"             "ManagersHalf"
#> [19] "Master"               "Parks"                "People"
#> [22] "Pitching"             "PitchingPost"         "Salaries"
#> [25] "Schools"              "SeriesPost"           "Teams"
#> [28] "TeamsFranchises"      "TeamsHalf"            "sqlite_stat1"
#> [31] "sqlite_stat4"
```

Example: Get a remote table

```
tbl(lahman_db, "Batting")
#> # Source:   table<Batting> [?? x 22]
#> # Database: sqlite 3.36.0
#> #   [/var/folders/8p/xzrrqphx2qb3d2s_fgqrk5xr0000gn/lahman.sqlite]
#>    playerID  yearID stint teamID lgID      G    AB     R     H   X2B   X3B    HR
#>    <chr>      <int> <int> <chr>  <chr> <int> <int> <int> <int> <int> <int> <int>
#>  1 abercda01   1871     1 TRO    NA        1     4     0     0     0     0     0
#>  2 addybo01    1871     1 RC1    NA       25   118    30    32     6     0     0
#>  3 allisar01   1871     1 CL1    NA       29   137    28    40     4     5     0
#>  4 allisdo01   1871     1 WS3    NA       27   133    28    44    10     2     2
#>  5 ansonca01   1871     1 RC1    NA       25   120    29    39    11     3     0
#>  6 armstbo01   1871     1 FW1    NA       12    49     9    11     2     1     0
#>  7 barkeal01   1871     1 RC1    NA        1     4     0     1     0     0     0
#>  8 barnero01   1871     1 BS1    NA       31   157    66    63    10     9     0
#>  9 barrebi01   1871     1 FW1    NA        1     5     1     1     1     0     0
#> 10 barrofr01   1871     1 BS1    NA       18    86    13    13     2     1     0
#> # … with more rows, and 10 more variables: RBI <int>, SB <int>, CS <int>,
#> #   BB <int>, SO <int>, IBB <int>, HBP <int>, SH <int>, SF <int>, GIDP <int>
```

Example: Run a SQL query

```
tbl(lahman_db, sql("SELECT playerID, COUNT(*) FROM Batting GROUP BY playerID"))
#> # Source:   SQL [?? x 2]
#> # Database: sqlite 3.36.0
#> #   [/var/folders/8p/xzrrqphx2qb3d2s_fgqrk5xr0000gn/lahman.sqlite]
#>    playerID  `COUNT(*)`
#>    <chr>          <int>
#>  1 aardsda01          9
#>  2 aaronha01         23
#>  3 aaronto01          7
#>  4 aasedo01          13
#>  5 abadan01           3
#>  6 abadfe01          10
#>  7 abadijo01          2
#>  8 abbated01         10
#>  9 abbeybe01          6
#> 10 abbeych01          5
#> # … with more rows
```

Example: Execute a command

```
DBI::dbExecute(lahman_db, "DROP TABLE Batting")
#> [1] 0
```

```r
DBI::dbListTables(lahman_db)
```

```r
tbl(lahman_db, sql("Batting"))
```

```r
tbl(lahman_db, sql("SELECT playerID, COUNT(*)
                    FROM Batting
                    GROUP BY playerID"))
```

```r
DBI::dbExecute(lahman_db, "DROP TABLE Batting")
```

```r
DBI::dbListTables(lahman_db)
```

```r
tbl(lahman_db, sql("Batting"))
```

```r
tbl(lahman_db, sql("SELECT playerID, COUNT(*)
                    FROM Batting
                    GROUP BY playerID"))
```

```r
DBI::dbExecute(lahman_db, "DROP TABLE Batting")
```

examples.R

```
1  library(dplyr)
2  library(dbplyr)
3
4  # Create a connection
5  lahman_db <- lahman_sqlite()
6
7  DBI::dbListTables(lahman_db)
8
9  tbl(lahman_db, sql("Batting"))
10
11 tbl(lahman_db, sql("SELECT playerID, COUNT(*)
12                      FROM Batting
13                      GROUP BY playerID"))
14
15 DBI::dbExecute(lahman_db, "DROP TABLE Batting")
```

15:48 (Top Level) R Script

Console Terminal Jobs

R 4.1.0 · ~/Repositories/rpackages/stackbigquery/

```
n.sqlite]
   playerID  `COUNT(*)`
   <chr>          <int>
 1 aardsda01          9
 2 aaronha01         23
 3 aaronto01          7
 4 aasedo01          13
 5 abadan01           3
 6 abadfe01          10
 7 abadijo01          2
 8 abbated01         10
 9 abbeybe01          6
10 abbeych01          5
# … with more rows
>
> DBI::dbExecute(lahman_db, "DROP TABLE Batting")
[1] 0
>
```

Environment History Connections Build Git

Import Dataset 287 MiB List

R Global Environment

Data

lahman_db	Formal class SQLiteConnection

Files Plots Packages Help Viewer

Refresh Help Topic

R: Cache and retrieve an 'src_sqlite' of the Lahman baseball... Find in Topic

lahman {dbplyr} R Documentation

Cache and retrieve an `src_sqlite` of the Lahman baseball database.

Description

This creates an interesting database using data from the Lahman baseball data source, provided by Sean Lahman at http://www.seanlahman.com/baseball-archive/statistics/, and made easily available in R through the **Lahman** package by Michael Friendly, Dennis Murphy and Martin Monkman. See the documentation for that package for documentation of the individual tables.

Usage

```
lahman_sqlite(path = NULL)

lahman_postgres(dbname = "lahman", host = "localhost", ...)
```

examples.R

```r
library(dplyr)
library(dbplyr)

# Create a connection
lahman_db <- lahman_sqlite()

DBI::dbListTables(lahman_db)

tbl(lahman_db, sql("Batting"))

tbl(lahman_db, sql("SELECT playerID, COUNT(*)
                    FROM Batting
                    GROUP BY playerID"))

DBI::dbExecute(lahman_db, "DROP TABLE Batting")
```

15:48 (Top Level) R Script

Console Terminal Jobs

R 4.1.0 · ~/Repositories/rpackages/stackbigquery/

```
n.sqlite]
   playerID  `COUNT(*)`
   <chr>          <int>
 1 aardsda01          9
 2 aaronha01         23
 3 aaronto01          7
 4 aasedo01          13
 5 abadan01           3
 6 abadfe01          10
 7 abadijo01          2
 8 abbated01         10
 9 abbeybe01          6
10 abbeych01          5
# … with more rows
>
> DBI::dbExecute(lahman_db, "DROP TABLE Batting")
[1] 0
>
```

Environment History Connections Build Git

Import Dataset 287 MiB List

R Global Environment

Data

lahman_db	Formal class SQLiteConnection

Files Plots Packages Help Viewer

Refresh Help Topic

R: Cache and retrieve an 'src_sqlite' of the Lahman baseball... Find in Topic

lahman {dbplyr} R Documentation

Cache and retrieve an `src_sqlite` of the Lahman baseball database.

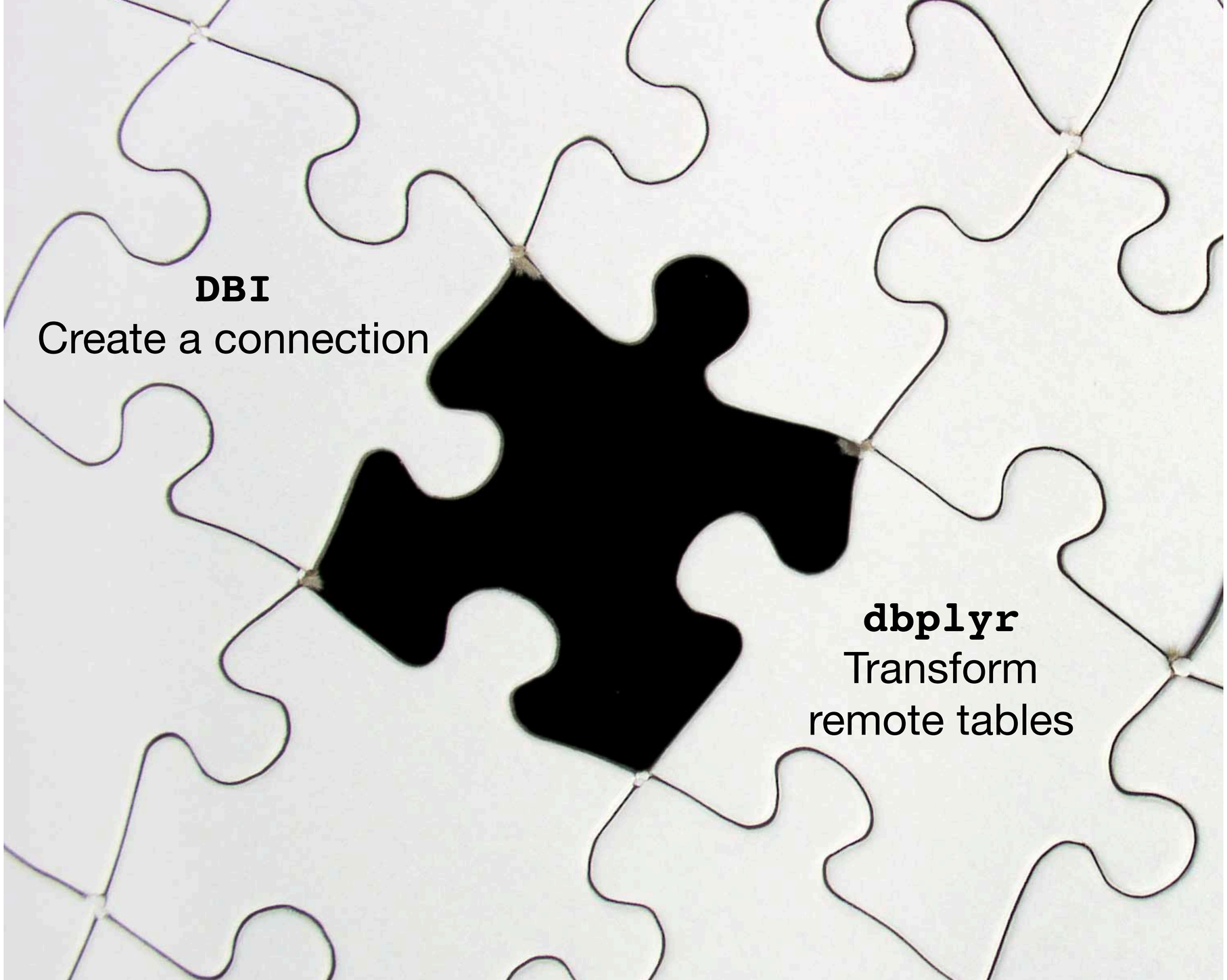
Description

This creates an interesting database using data from the Lahman baseball data source, provided by Sean Lahman at http://www.seanlahman.com/baseball-archive/statistics/, and made easily available in R through the **Lahman** package by Michael Friendly, Dennis Murphy and Martin Monkman. See the documentation for that package for documentation of the individual tables.

Usage

```
lahman_sqlite(path = NULL)

lahman_postgres(dbname = "lahman", host = "localhost", ...)
```


DBI
Create a connection
dbplyr
Transform
remote tables

Solution
dbcooper turns a database into a collection of functions

```
dbcooper::dbc_init(lahman_db, "lahman")
```

- Stores and manages the connection globally
- Generates prefixed functions for tasks
- Generates autocomplete-friendly accessors for tables

dbcooper creates prefixed functions for common tasks

```
dbcooper::dbc_init(lahman_db, "lahman")
```

`DBI::dbListTables(lahman_db)`	`lahman_list()`
`tbl(lahman_db, sql("Batting"))`	`lahman_tbl("Batting")`
`tbl(lahman_db, sql("SELECT playerID, COUNT(*) FROM Batting GROUP BY playerID"))`	`lahman_query("SELECT playerID, COUNT(*) FROM Batting GROUP BY playerID")`
`DBI::dbExecute(lahman_db, "DROP TABLE Batting")`	`lahman_execute("DROP TABLE Batting")`

Access tables with autocomplete

```
lahman_tbl("Batting")
lahman_tbl("Fielding")
lahman_tbl("Teams")
```

```
lahman_batting()
lahman_fielding()
lahman_teams()
```

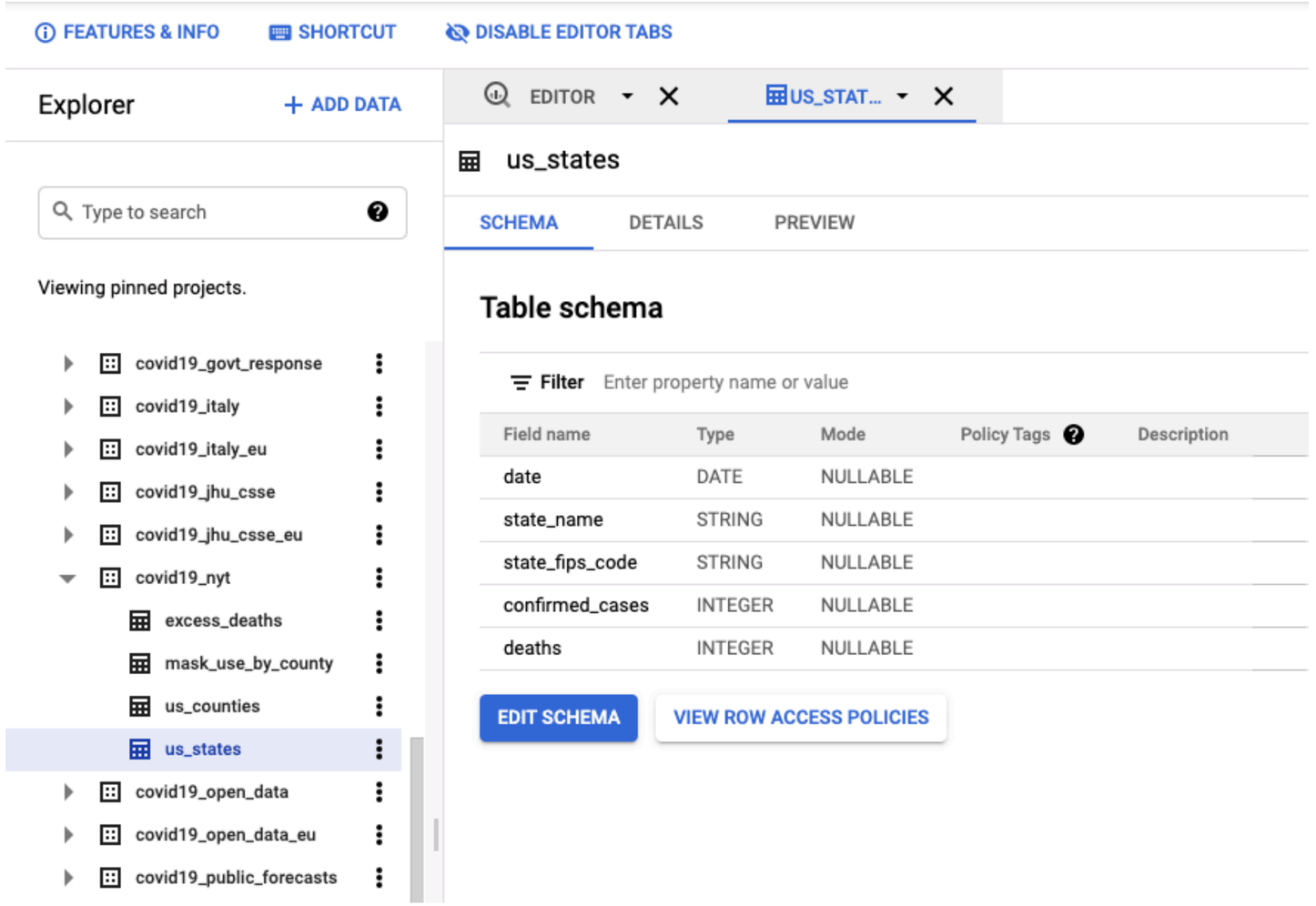
Access tables with autocomplete

```
tbl(lahman_db, "Batting") %>%
  inner_join(tbl(lahman_db, "Fielding"), by = "playerID") %>%
  left_join(tbl(lahman_db, "Salaries"), by = c("playerID", "yearID"))
```

```
lahman_batting() %>%
  inner_join(lahman_fielding(), by = "playerID") %>%
  left_join(lahman_salaries(), by = c("playerID", "yearID"))
```

Example:
NYT COVID-19 Database

BigQuery Public Databases



Learn more: BigQuery Public Databases

Example: NYT COVID-19 Data

```
library(dbcooper)

# Use any approach to construct the connection
con <- DBI::dbConnect(bigrquery::bigquery(),
                      project = "bigquery-public-data",
                      dataset = "covid19_nyt",
                      billing = Sys.getenv("BIGQUERY_BILLING_PROJECT"))
```

Example: NYT COVID-19 Data

```
library(dbcooper)

# Use any approach to construct the connection
con <- DBI::dbConnect(bigrquery::bigquery(),
                      project = "bigquery-public-data",
                      dataset = "covid19_nyt",
                      billing = Sys.getenv("BIGQUERY_BILLING_PROJECT"))

# Create functions with prefix covid_
dbc_init(con, "covid")
```

Exploring COVID-19 Data

```
# Create functions with prefix covid_
dbc_init(con, “covid")
```

Exploring COVID-19 Data

```
# Create functions with prefix covid_
dbc_init(con, “covid")

covid_list()
#> [1] "excess_deaths"      "mask_use_by_county" "us_counties"
#> [4] "us_states"
```

Exploring COVID-19 Data

```
# Create functions with prefix covid_
dbc_init(con, “covid")

covid_list()
#> [1] "excess_deaths"      "mask_use_by_county" "us_counties"
#> [4] "us_states"

covid_us_states()
#> # Source:   SQL [?? x 5]
#> # Database: BigQueryConnection
#>    date       state_name state_fips_code confirmed_cases deaths
#>    <date>     <chr>      <chr>                     <int>  <int>
#>  1 2020-03-15 Guam       66                            3      0
#>  2 2020-03-16 Guam       66                            3      0
#>  3 2020-03-17 Guam       66                            3      0
#>  4 2020-03-18 Guam       66                            8      0
#>  5 2020-03-19 Guam       66                           12      0
#>  6 2020-03-20 Guam       66                           14      0
#>  7 2020-03-21 Guam       66                           15      0
#>  8 2020-03-22 Guam       66                           27      1
#>  9 2020-03-23 Guam       66                           29      1
#> 10 2020-03-24 Guam       66                           32      1
#> # … with more rows
```

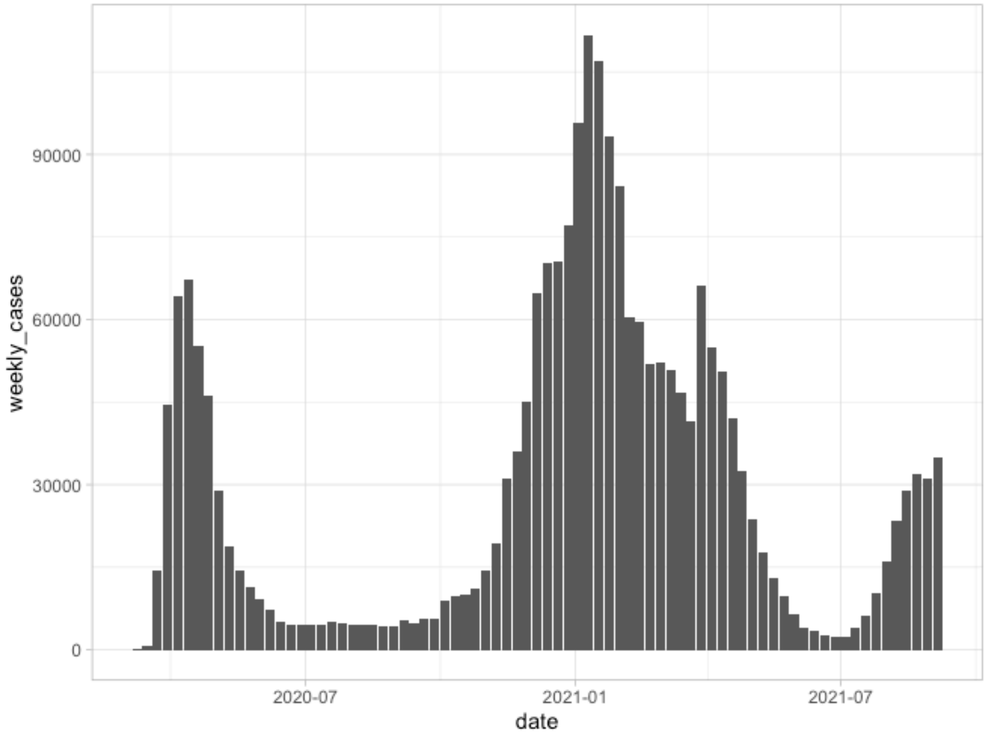
Exploring COVID-19 Data

```
covid_us_states() %>%
    filter(date == max(date)) %>%
    arrange(desc(confirmed_cases))
```

```
#> # Source:     lazy query [?? x 5]
#> # Database:   BigQueryConnection
#> # Ordered by: desc(confirmed_cases)
#>    date       state_name     state_fips_code confirmed_cases deaths
#>    <date>     <chr>          <chr>                     <int>  <int>
#>  1 2021-09-08 California     06                      4545832  66722
#>  2 2021-09-08 Texas          48                      3752776  59090
#>  3 2021-09-08 Florida        12                      3378772  46977
#>  4 2021-09-08 New York       36                      2310837  54088
#>  5 2021-09-08 Illinois       17                      1558768  26769
#>  6 2021-09-08 Georgia        13                      1436172  22639
#>  7 2021-09-08 Pennsylvania   42                      1329111  28446
#>  8 2021-09-08 North Carolina 37                      1269208  14925
#>  9 2021-09-08 Ohio           39                      1268841  21020
#> 10 2021-09-08 New Jersey     34                      1108291  27007
#> # … with more rows
```

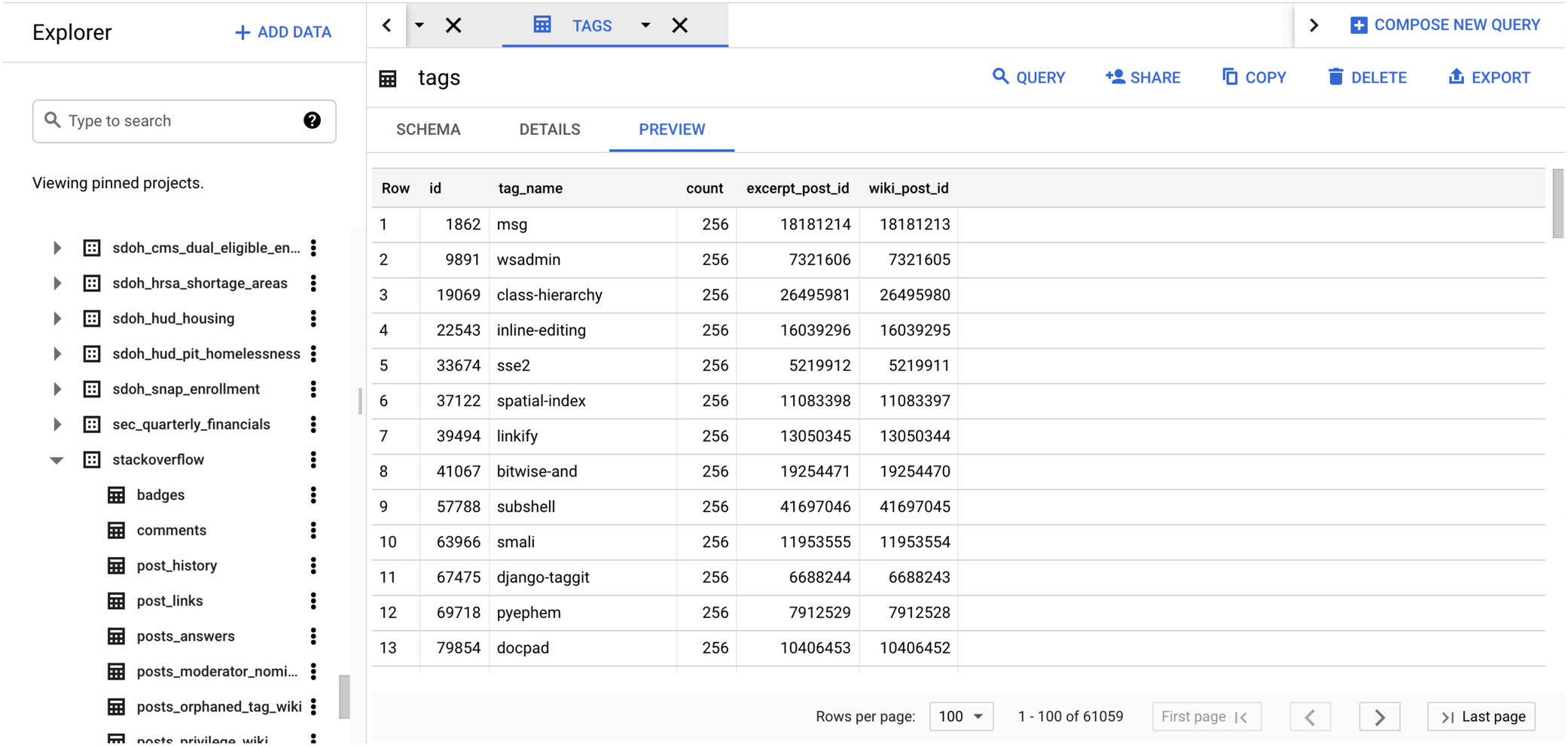
Exploring COVID-19 Data: New York Cases

```
covid_us_states() %>%
    filter(state_name == "New York",
           EXTRACT(DAYOFWEEK %FROM% date) == 1) %>%
    arrange(date) %>%
    mutate(weekly_cases = confirmed_cases - lag(confirmed_cases)) %>%
    ggplot(aes(date, weekly_cases)) +
    geom_col()
```



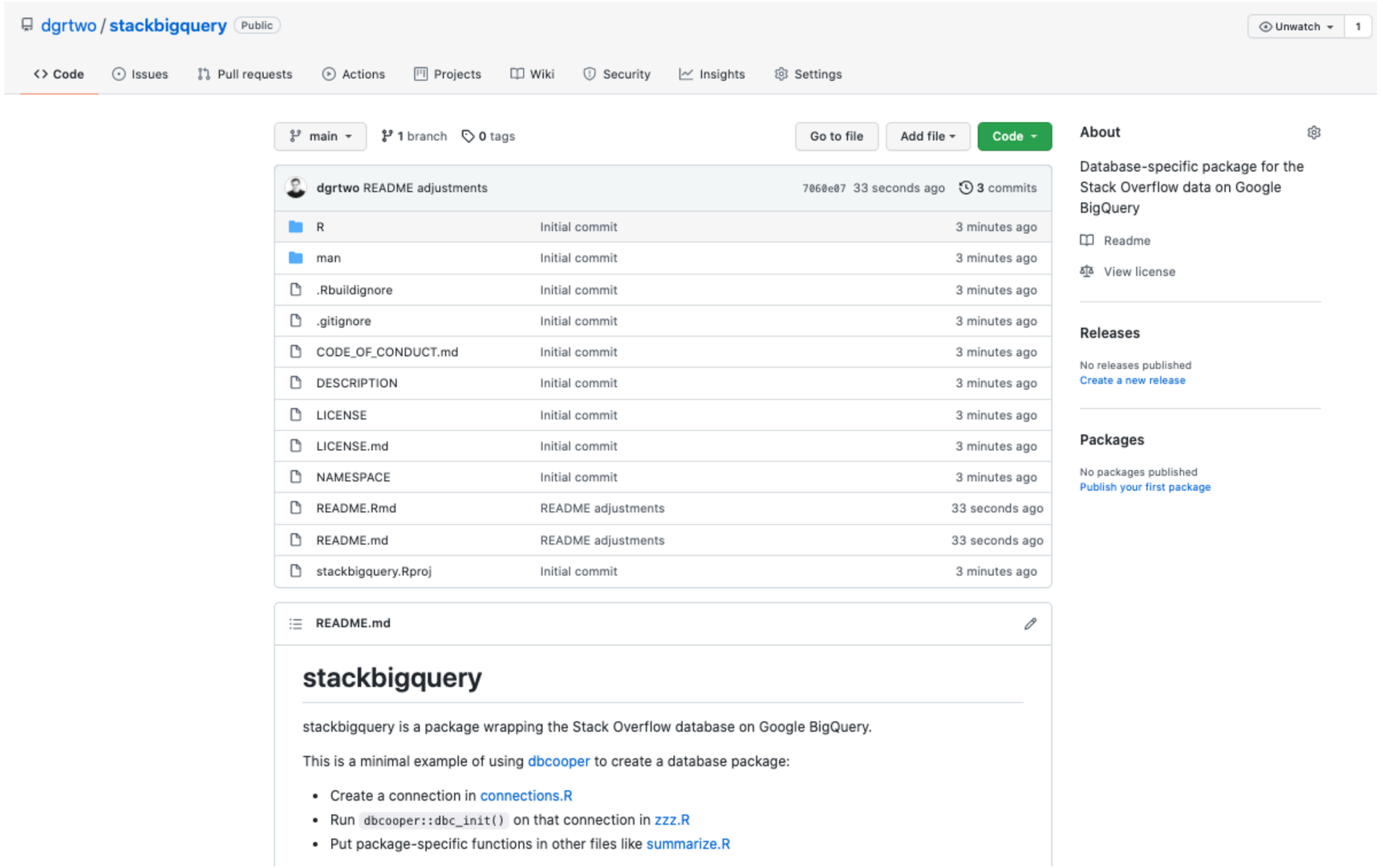
Creating a package

BigQuery Database: Stack Overflow



Learn more: Stack Overflow on BigQuery

Example: stackbigquery package



Link

1. Write database-specific connection code

connections.R

Source on Save

```r
#' Create a connection object
#'
#' If bigquery_billing_project isn't set, raises an error.
stack_create_connection <- function() {
    DBI::dbConnect(bigrquery::bigquery(),
                   project = "bigquery-public-data",
                   dataset = "stackoverflow",
                   billing = bigquery_billing_project())
}
```

2. Add code to zzz.R to run dbc_init()

zzz.R

Source on Save

```r
.onLoad <- function(libname, pkgname) {
    # Use the internal function to create a db connection (see connections.R)
    con <- stack_create_connection()

    # Add tbl, query, execute, etc functions to the package
    package_env <- parent.env(environment())
    dbcooper::dbc_init(con, "stack", env = package_env)
}

.onUnload <- function(libpath){
    dbcooper::dbc_clear_connections()
}
```

3. Add any database-specific verbs

summarize.R

Source on Save

```r
#' Given a grouped remote table of Stack Overflow posts, summarize the number of
#' questions in selected tags within each group
#'
#' @param tbl A remote grouped table based on stack_post_questions
#' @param tags Optionally, a vector of tags to summarize
#' @param question_id Which column represents the question ID?
#'
#' @importFrom dplyr %>%
#'
#' @export
summarize_tags <- function(tbl, tags = NULL, question_id = "id") {
    g <- dplyr::group_vars(tbl)

    tag_tbl <- stack_question_tags()
    if (!is.null(tags)) {
        tag_tbl <- tag_tbl %>%
            dplyr::filter(tag %in% tags)
    }

    by <- stats::setNames("question_id", question_id)
    numerator <- tbl %>%
        dplyr::inner_join(tag_tbl, by = by) %>%
        dplyr::group_by(tag, .add = TRUE) %>%
        dplyr::summarize(number = n()) %>%
        dplyr::collect()

    totals <- tbl %>%
        dplyr::summarize(total = n()) %>%
```

And you have a package!

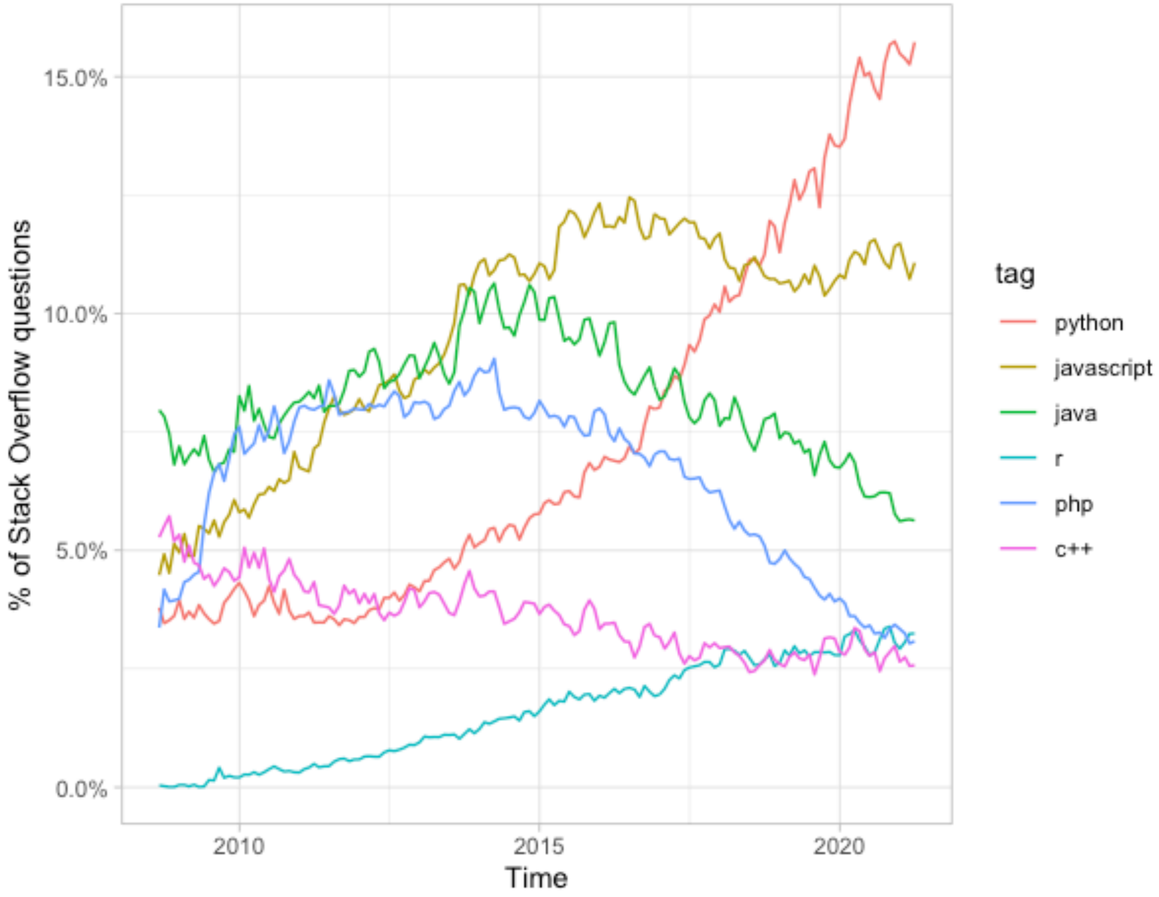
```
library(stackbigquery)
stack_posts_questions()
#> # Source:   SQL [?? x 20]
#> # Database: BigQueryConnection
#>          id title        body         accepted_answer… answer_count comment_count
#>       <int> <chr>        <chr>                   <int>        <int>          <int>
#>  1  3.58e7 Camera with… "<p>current…               NA            2              0
#>  2  3.59e7 Configure a… "<p>How can…               NA            2              0
#>  3  3.59e7 How does Ex… "<p>I notic…               NA            1              0
#>  4  3.59e7 CXF with OD… "<p>Could a…               NA            1              0
#>  5  3.59e7 Why does ch… "<p>So I'm …               NA            1              0
#>  6  3.59e7 Algolia sea… "<p>I'm usi…               NA            1              0
#>  7  3.60e7 How to use … "<p>We are …               NA            4              0
#>  8  3.60e7 How to prin… "<p>Im work…               NA            1              0
#>  9  3.60e7 Can I write… "<p>I'm try…         35978083            1              0
#> 10  3.60e7 XQuery join… "<p>I have …         35988035            2              0
#> # … with more rows, and 14 more variables: community_owned_date <dttm>,
#> #   creation_date <dttm>, favorite_count <int>, last_activity_date <dttm>,
#> #   last_edit_date <dttm>, last_editor_display_name <chr>,
#> #   last_editor_user_id <int>, owner_display_name <chr>, owner_user_id <int>,
#> #   parent_id <chr>, post_type_id <int>, score <int>, tags <chr>,
#> #   view_count <int>
```

Exploring Stack Overflow: Tags over Time

```
library(tidyverse)
library(stackbigquery)

by_month_tag <- stack_posts_questions() %>%
  group_by(month = DATE_TRUNC(DATE(creation_date), MONTH)) %>%
  summarize_tags(c("javascript", "java", “python",
                   "php", "c++", "r"))

by_month_tag %>%
  filter(month != max(month),
         month != min(month)) %>%
  arrange(month) %>%
  mutate(tag = fct_reorder(tag, -percent, last)) %>%
  ggplot(aes(month, percent, color = tag)) +
  geom_line() +
  scale_y_continuous(labels = scales::percent_format()) +
  expand_limits(y = 0) +
  labs(x = “Time",
       y = "% of Stack Overflow questions")
```



- Don’t underestimate the value of small usability improvements that affect you every day
- A good database package lets you forget about the package completely

examples.R

```
1  library(tidyverse)
2  library(dbcooper)
3  theme_set(theme_light())
4
5  # Create the connection the same way you always would
6  con <- DBI::dbConnect(bigrquery::bigquery(),
7                        project = "bigquery-public-data",
8                        dataset = "covid19_nyt",
9                        billing = Sys.getenv("BIGQUERY_BILLING_PROJECT"))
10
11 # Create functions with prefix covid_
12 dbc_init(con, "covid")
13
14 covid_us_states() %>%
15     filter(state_name == "New York",
16            EXTRACT(DAYOFWEEK %FROM% date) == 1) %>%
17     arrange(date) %>%
18     mutate(weekly_cases = confirmed_cases - lag(confirmed_cases)) %>%
19     ggplot(aes(date, weekly_cases)) +
20     geom_col()
```

22:1 (Top Level) R Script

Console | Terminal | Find in Files | Jobs

R 4.1.0 · ~/Repositories/rpackages/stackbigquery/

```
+                        project = "bigquery-public-data",
+                        dataset = "covid19_nyt",
+                        billing = Sys.getenv("BIGQUERY_BILLING_PROJECT"))
> # Create functions with prefix covid_
> dbc_init(con, "covid")
> covid_us_states() %>%
+     filter(state_name == "New York",
+            EXTRACT(DAYOFWEEK %FROM% date) == 1) %>%
+     arrange(date) %>%
+     mutate(weekly_cases = confirmed_cases - lag(confirmed_cases)) %>%
+     ggplot(aes(date, weekly_cases)) +
+     geom_col()
Complete
Billed: 0 B
Downloading first chunk of data.
First chunk includes all requested rows.
Warning message:
Removed 1 rows containing missing values (position_stack).
>
```

Environment | History | Connections | Build | Git

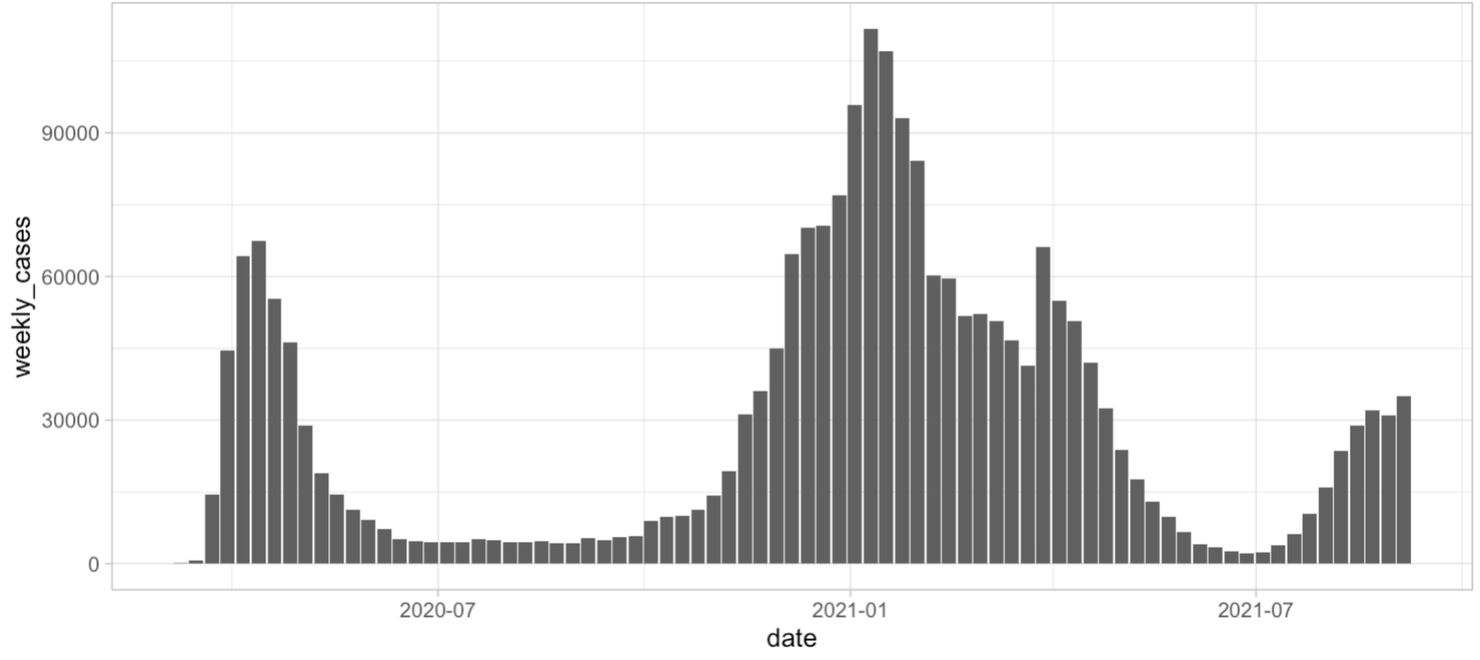
Import Dataset · 219 MiB · List

R · Global Environment

Data	
con	Formal class BigQueryConnection
Functions	
covid_excess_deaths	function ()
covid_execute	function (query)
covid_list	function (query)
covid_mask_use_by_county	function ()
covid_query	function (query)
covid_src	function ()
covid_tbl	function (table_name = NULL)
covid_us_counties	function ()
covid_us_states	function ()

Files | Plots | Packages | Help | Viewer

Zoom · Export · Publish



examples.R

```r
library(tidyverse)
library(dbcooper)
theme_set(theme_light())

# Create the connection the same way you always would
con <- DBI::dbConnect(bigrquery::bigquery(),
                      project = "bigquery-public-data",
                      dataset = "covid19_nyt",
                      billing = Sys.getenv("BIGQUERY_BILLING_PROJECT"))

# Create functions with prefix covid_
dbc_init(con, "covid")

covid_us_states() %>%
    filter(state_name == "New York",
           EXTRACT(DAYOFWEEK %FROM% date) == 1) %>%
    arrange(date) %>%
    mutate(weekly_cases = confirmed_cases - lag(confirmed_cases)) %>%
    ggplot(aes(date, weekly_cases)) +
    geom_col()
```

22:1 (Top Level) R Script

Console Terminal Find in Files Jobs

R 4.1.0 · ~/Repositories/rpackages/stackbigquery/

```
+                      project = "bigquery-public-data",
+                      dataset = "covid19_nyt",
+                      billing = Sys.getenv("BIGQUERY_BILLING_PROJECT"))
> # Create functions with prefix covid_
> dbc_init(con, "covid")
> covid_us_states() %>%
+     filter(state_name == "New York",
+            EXTRACT(DAYOFWEEK %FROM% date) == 1) %>%
+     arrange(date) %>%
+     mutate(weekly_cases = confirmed_cases - lag(confirmed_cases)) %>%
+     ggplot(aes(date, weekly_cases)) +
+     geom_col()
Complete
Billed: 0 B
Downloading first chunk of data.
First chunk includes all requested rows.
Warning message:
Removed 1 rows containing missing values (position_stack).
>
```

Environment History Connections Build Git

Import Dataset 219 MiB List

R Global Environment

Data

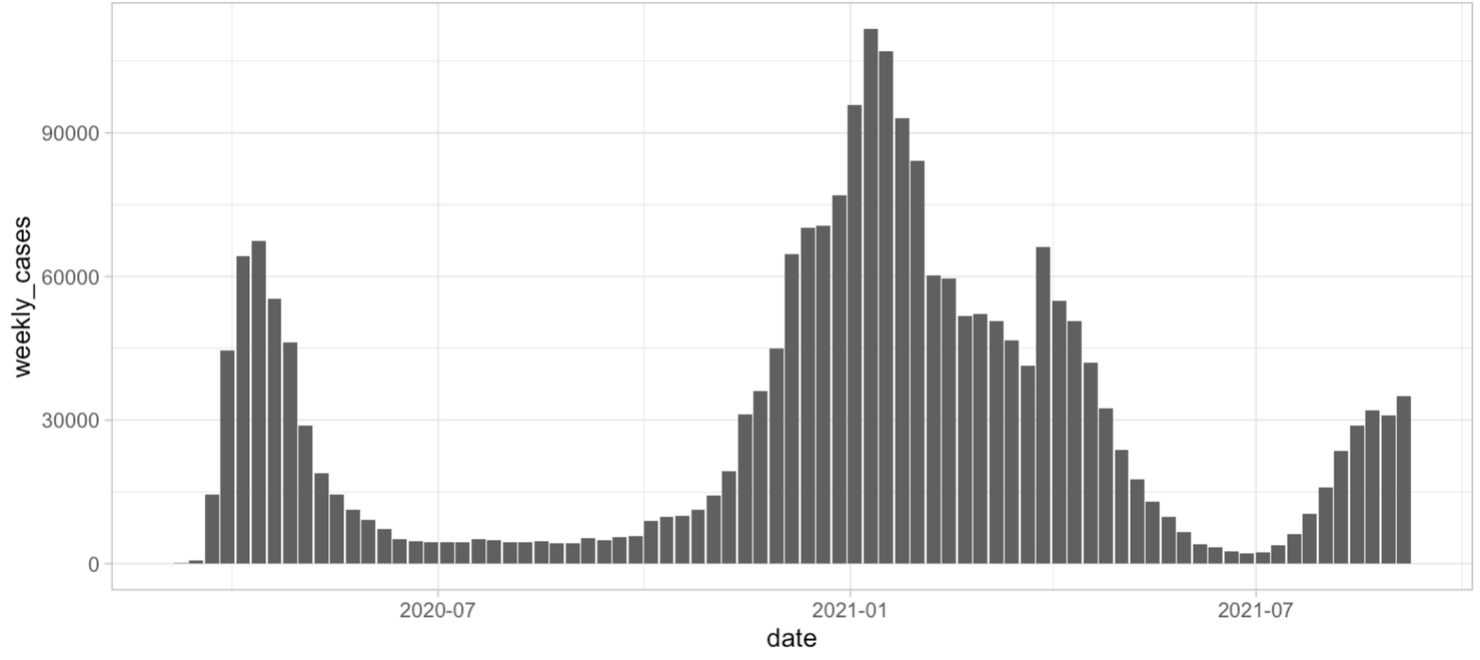
con	Formal class BigQueryConnection

Functions

covid_excess_deaths	function ()
covid_execute	function (query)
covid_list	function (query)
covid_mask_use_by_county	function ()
covid_query	function (query)
covid_src	function ()
covid_tbl	function (table_name = NULL)
covid_us_counties	function ()
covid_us_states	function ()

Files Plots Packages Help Viewer

Zoom Export Publish



Thank you

- **Lander Analytics**
  - Jared Lander
  - Nicole DelGiudice
- **Collaborators**
  - Chris Cardillo
  - Ramnath Vaidyanathan

@drob

www.varianceexplained.org