# Modeling in Databases with R

Best practices for working with Big Data in R

James Blair, Solutions Engineer @ RStudio

October 2019

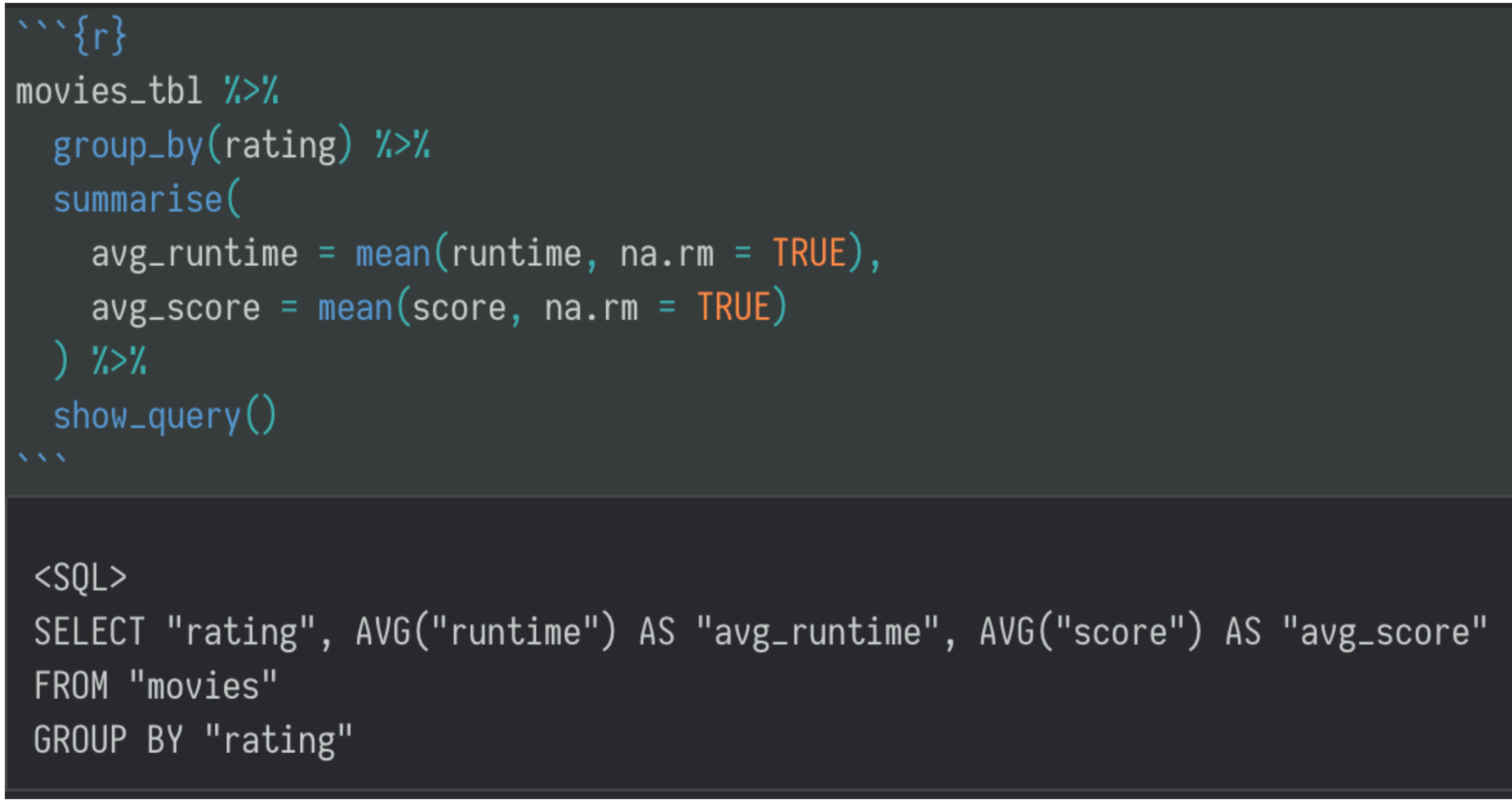
UNIVERSE

DENVER 2019

teradata.

# Agenda

- Big Data, Big Problems
- Databases with R
- R Packages
  - dbplot
  - corrr
  - tidypredict
  - modeldb
- Demo
- Resources



~~~
```{r}
movies_tbl %>%
  group_by(rating) %>%
  summarise(
    avg_runtime = mean(runtime, na.rm = TRUE),
    avg_score = mean(score, na.rm = TRUE)
  ) %>%
  show_query()
```
~~~

```
<SQL>
SELECT "rating", AVG("runtime") AS "avg_runtime", AVG("score") AS "avg_score"
FROM "movies"
GROUP BY "rating"
```

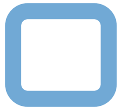
# Big Data, Big Problems

# Data > RAM

*Garrett Grolemund*

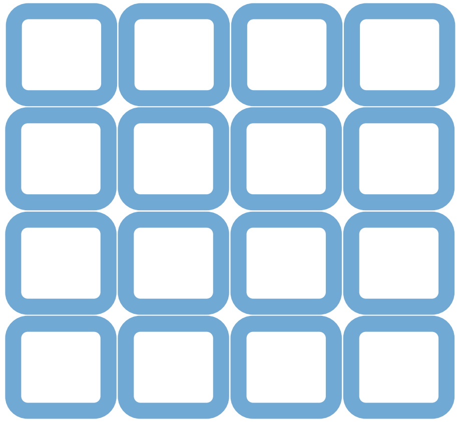
# Big Data, Big Problems

## Sample



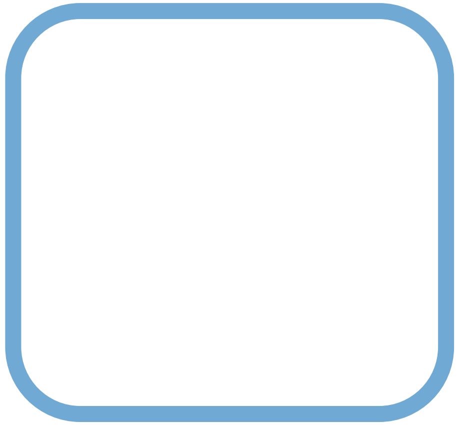
Most common approach for **modeling**

## Parts



Most common approach for **general analysis**

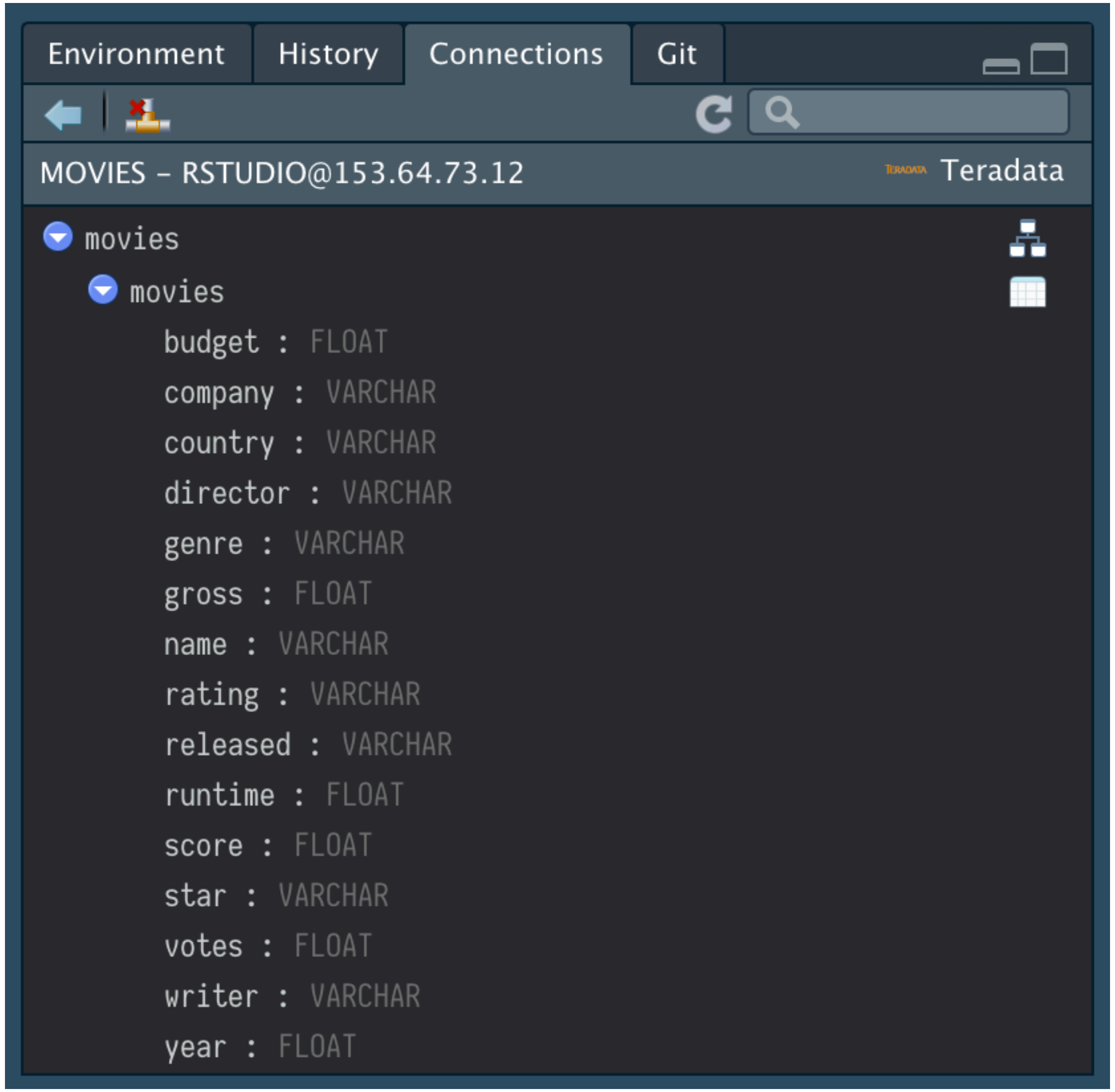
## Whole



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

# Databases with R

- Connect using ODBC
- Query with SQL
- Use common tidyverse verbs
- Preview data inside RStudio

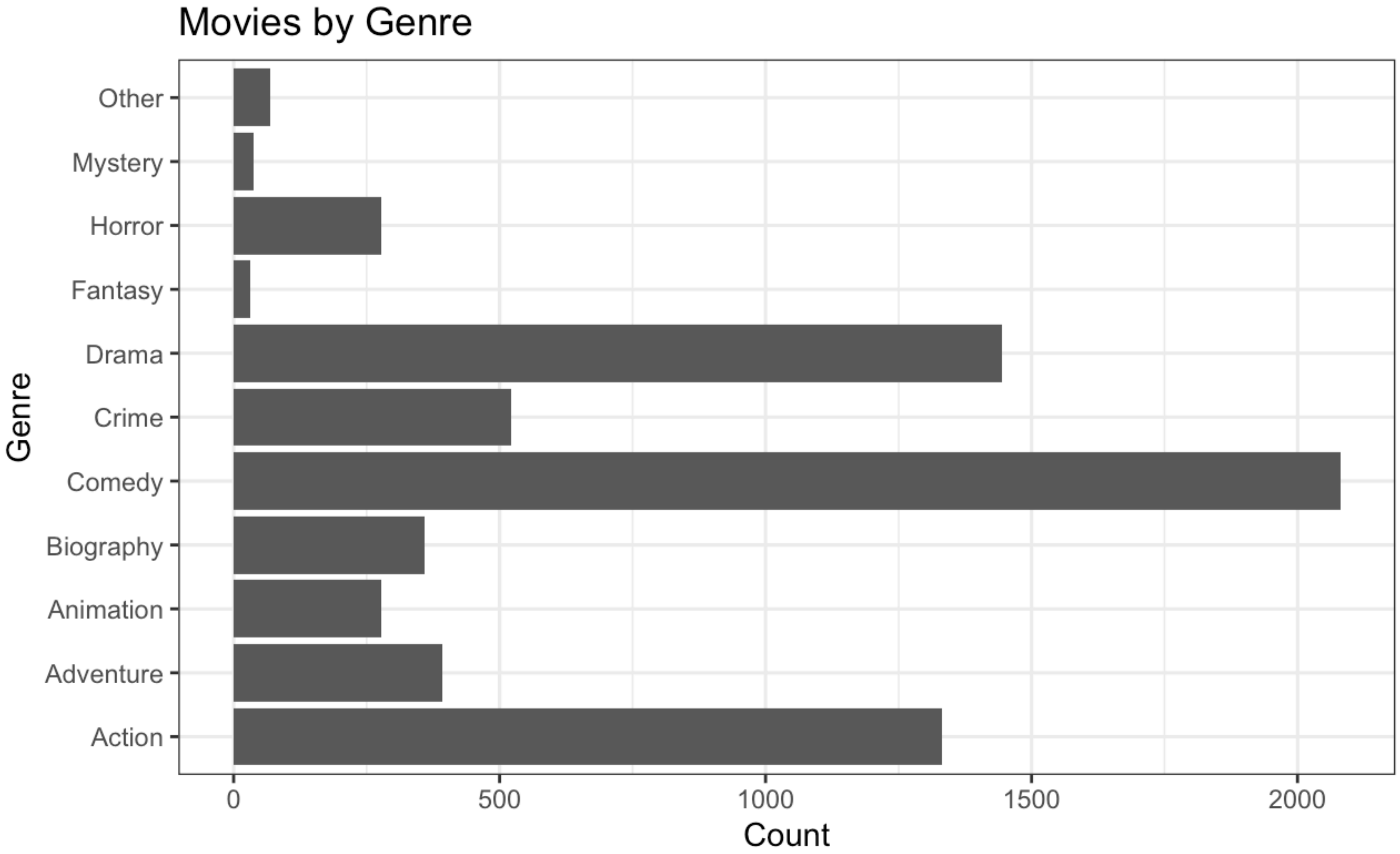


```
MOVIES – RSTUDIO@153.64.73.12
movies
  movies
    budget : FLOAT
    company : VARCHAR
    country : VARCHAR
    director : VARCHAR
    genre : VARCHAR
    gross : FLOAT
    name : VARCHAR
    rating : VARCHAR
    released : VARCHAR
    runtime : FLOAT
    score : FLOAT
    star : VARCHAR
    votes : FLOAT
    writer : VARCHAR
    year : FLOAT
```

# R packages

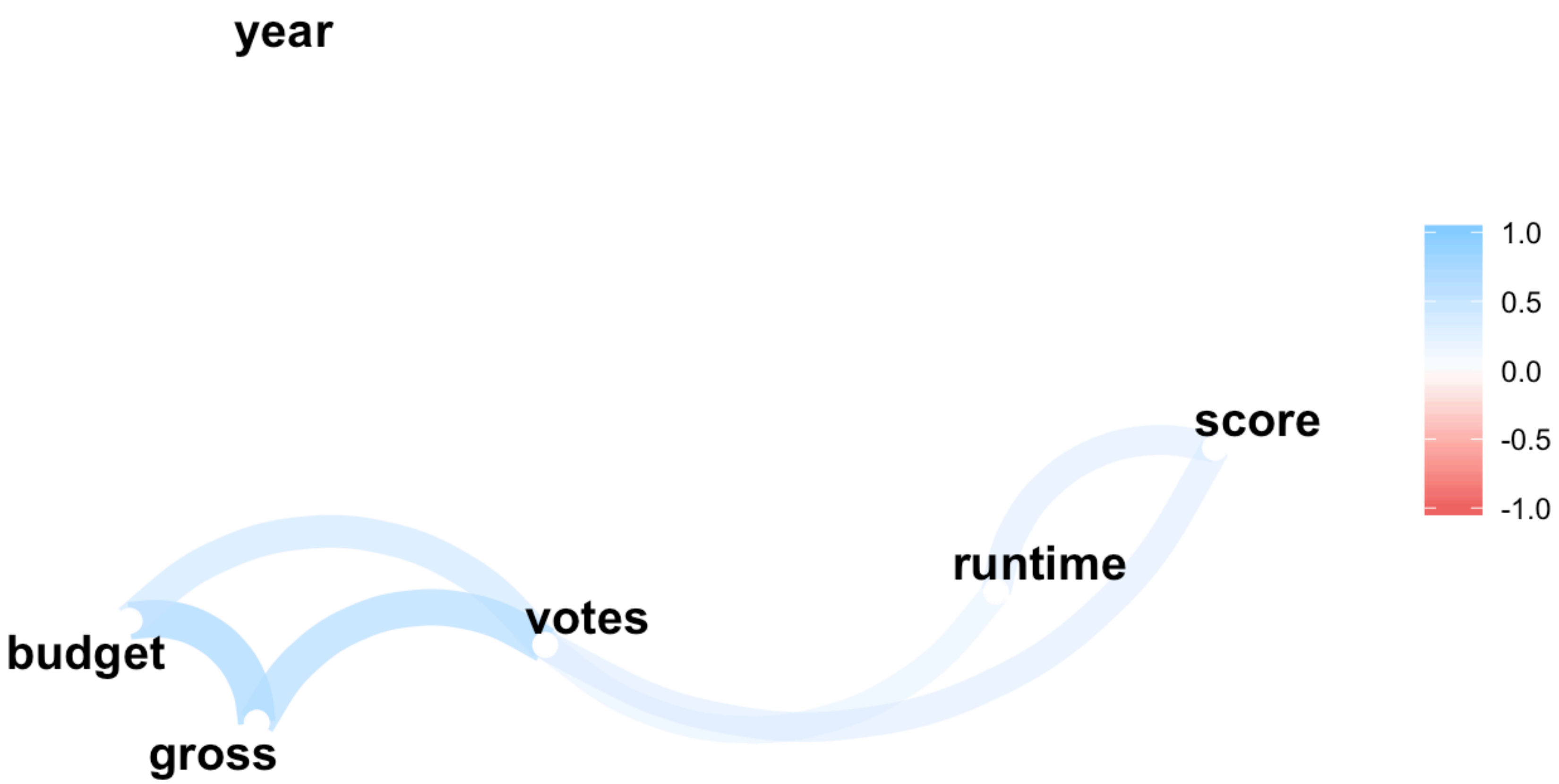
# R packages

dbplot

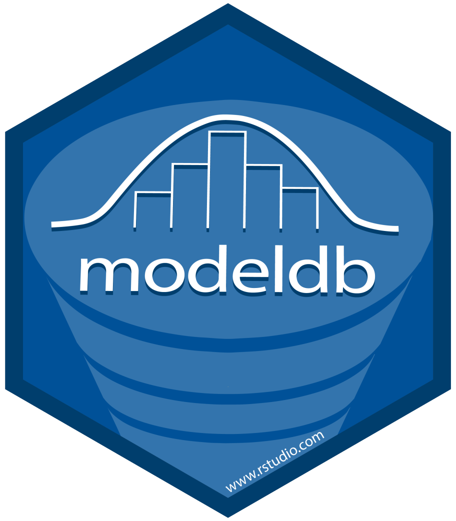

Movies by Genre
Genre
Other
Mystery
Horror
Fantasy
Drama
Crime
Comedy
Biography
Animation
Adventure
Action
0
500
1000
1500
2000
Count

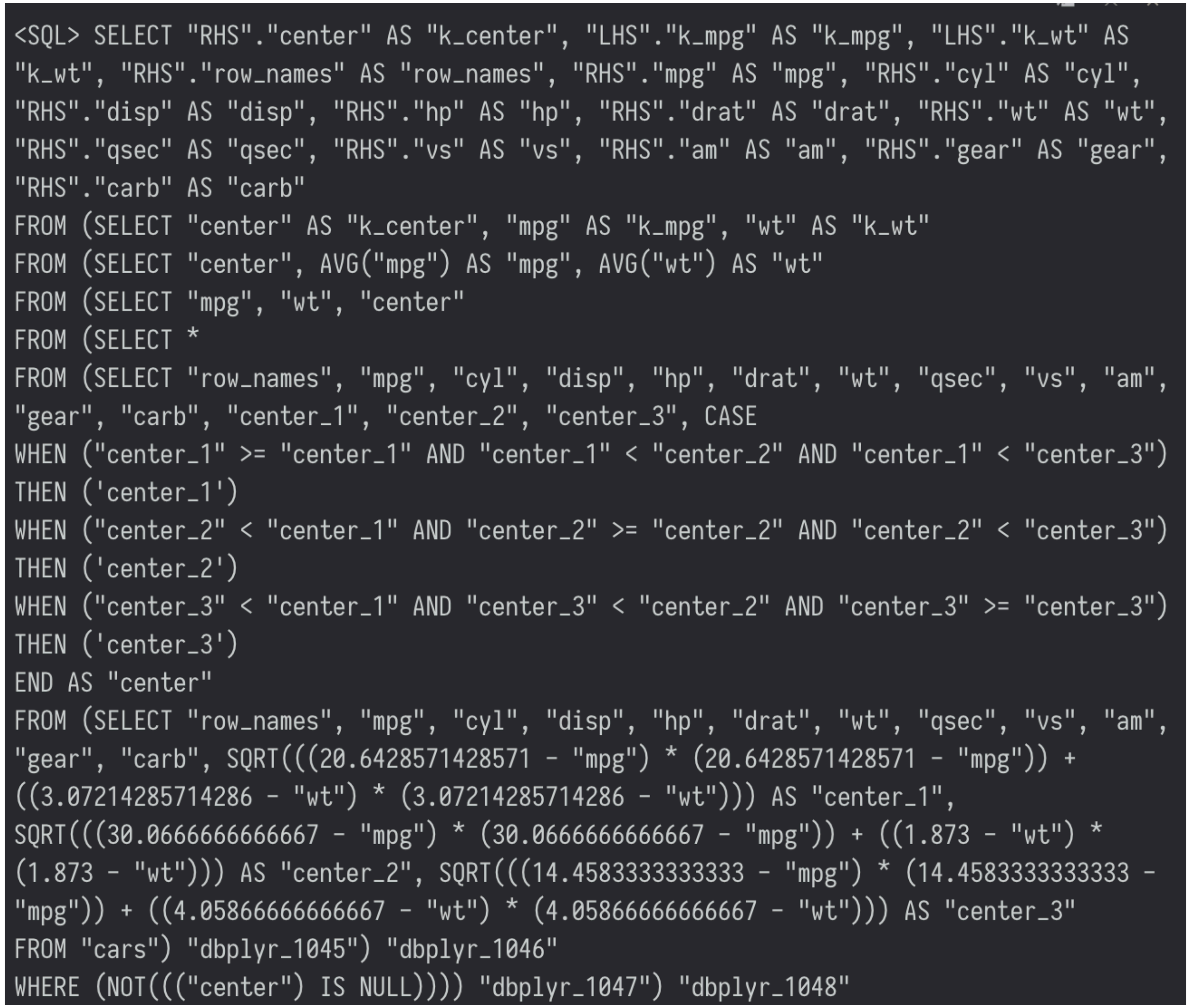
# R packages

corrr
github.com/drsimonj/corrr


year
score
runtime
votes
budget
gross
1.0
0.5
0.0
-0.5
-1.0

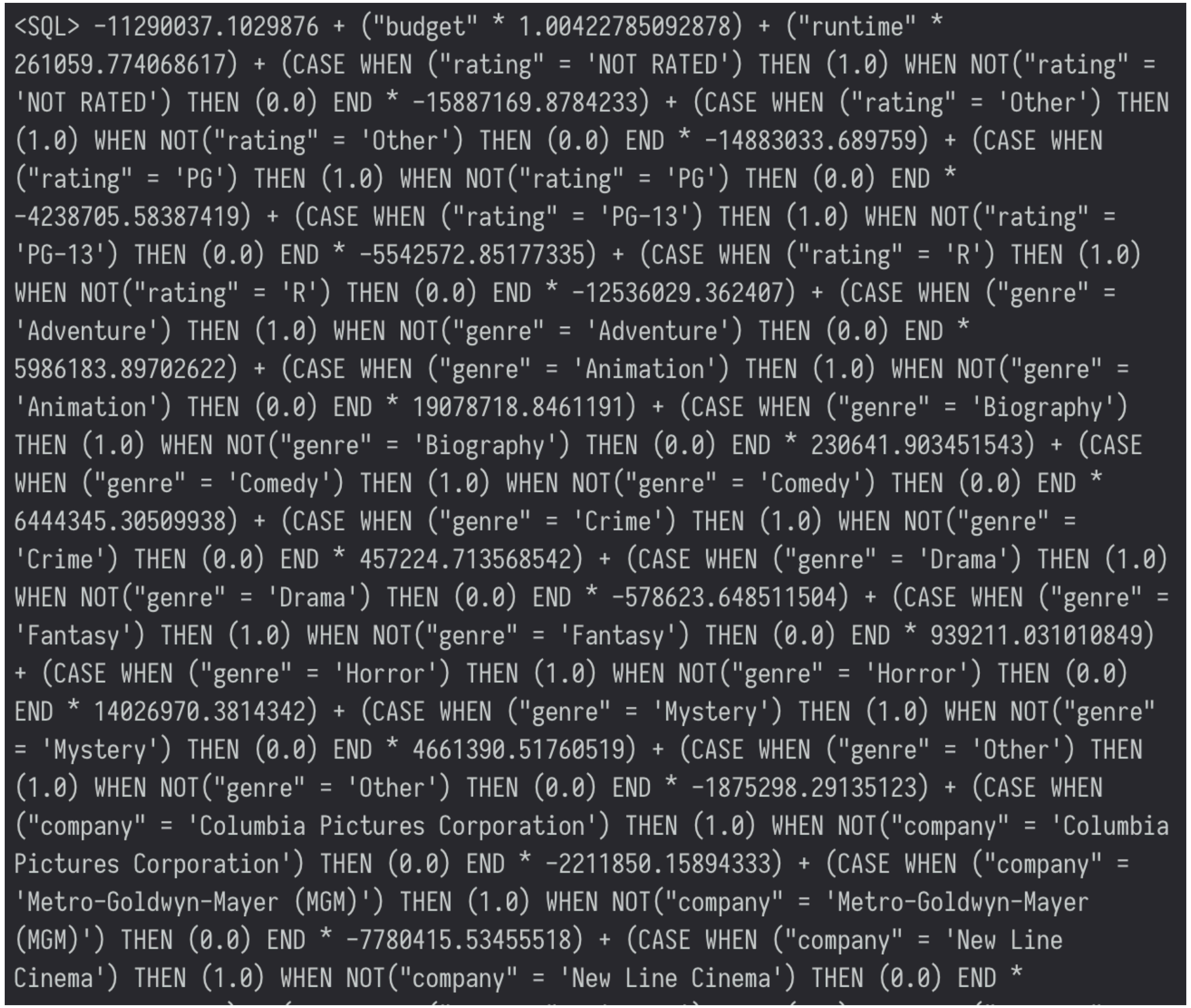
# R packages





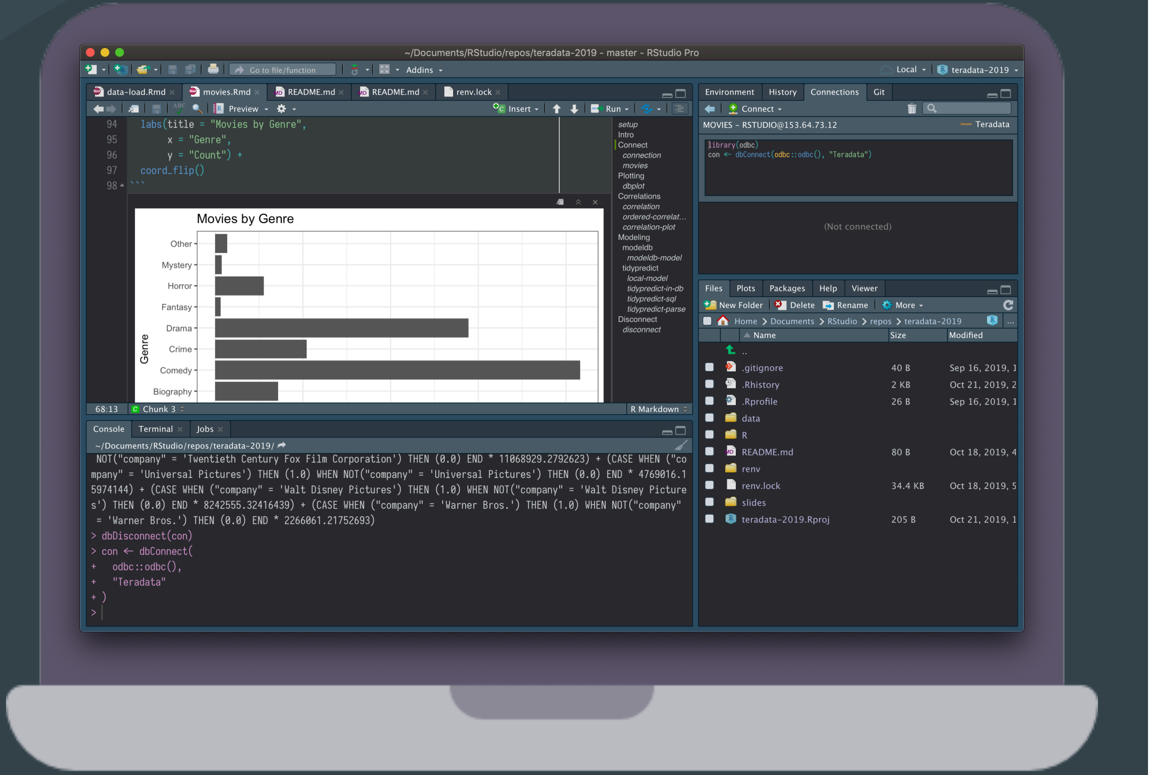
```
<SQL> SELECT "RHS"."center" AS "k_center", "LHS"."k_mpg" AS "k_mpg", "LHS"."k_wt" AS
"k_wt", "RHS"."row_names" AS "row_names", "RHS"."mpg" AS "mpg", "RHS"."cyl" AS "cyl",
"RHS"."disp" AS "disp", "RHS"."hp" AS "hp", "RHS"."drat" AS "drat", "RHS"."wt" AS "wt",
"RHS"."qsec" AS "qsec", "RHS"."vs" AS "vs", "RHS"."am" AS "am", "RHS"."gear" AS "gear",
"RHS"."carb" AS "carb"
FROM (SELECT "center" AS "k_center", "mpg" AS "k_mpg", "wt" AS "k_wt"
FROM (SELECT "center", AVG("mpg") AS "mpg", AVG("wt") AS "wt"
FROM (SELECT "mpg", "wt", "center"
FROM (SELECT *
FROM (SELECT "row_names", "mpg", "cyl", "disp", "hp", "drat", "wt", "qsec", "vs", "am",
"gear", "carb", "center_1", "center_2", "center_3", CASE
WHEN ("center_1" >= "center_1" AND "center_1" < "center_2" AND "center_1" < "center_3")
THEN ('center_1')
WHEN ("center_2" < "center_1" AND "center_2" >= "center_2" AND "center_2" < "center_3")
THEN ('center_2')
WHEN ("center_3" < "center_1" AND "center_3" < "center_2" AND "center_3" >= "center_3")
THEN ('center_3')
END AS "center"
FROM (SELECT "row_names", "mpg", "cyl", "disp", "hp", "drat", "wt", "qsec", "vs", "am",
"gear", "carb", SQRT(((20.6428571428571 - "mpg") * (20.6428571428571 - "mpg")) +
((3.07214285714286 - "wt") * (3.07214285714286 - "wt"))) AS "center_1",
SQRT(((30.0666666666667 - "mpg") * (30.0666666666667 - "mpg")) + ((1.873 - "wt") *
(1.873 - "wt"))) AS "center_2", SQRT(((14.4583333333333 - "mpg") * (14.4583333333333 -
"mpg")) + ((4.05866666666667 - "wt") * (4.05866666666667 - "wt"))) AS "center_3"
FROM "cars") "dbplyr_1045") "dbplyr_1046"
WHERE (NOT((("center") IS NULL)))) "dbplyr_1047") "dbplyr_1048"
```

# R packages



```
<SQL> -11290037.1029876 + ("budget" * 1.00422785092878) + ("runtime" *
261059.774068617) + (CASE WHEN ("rating" = 'NOT RATED') THEN (1.0) WHEN NOT("rating" =
'NOT RATED') THEN (0.0) END * -15887169.8784233) + (CASE WHEN ("rating" = 'Other') THEN
(1.0) WHEN NOT("rating" = 'Other') THEN (0.0) END * -14883033.689759) + (CASE WHEN
("rating" = 'PG') THEN (1.0) WHEN NOT("rating" = 'PG') THEN (0.0) END *
-4238705.58387419) + (CASE WHEN ("rating" = 'PG-13') THEN (1.0) WHEN NOT("rating" =
'PG-13') THEN (0.0) END * -5542572.85177335) + (CASE WHEN ("rating" = 'R') THEN (1.0)
WHEN NOT("rating" = 'R') THEN (0.0) END * -12536029.362407) + (CASE WHEN ("genre" =
'Adventure') THEN (1.0) WHEN NOT("genre" = 'Adventure') THEN (0.0) END *
5986183.89702622) + (CASE WHEN ("genre" = 'Animation') THEN (1.0) WHEN NOT("genre" =
'Animation') THEN (0.0) END * 19078718.8461191) + (CASE WHEN ("genre" = 'Biography')
THEN (1.0) WHEN NOT("genre" = 'Biography') THEN (0.0) END * 230641.903451543) + (CASE
WHEN ("genre" = 'Comedy') THEN (1.0) WHEN NOT("genre" = 'Comedy') THEN (0.0) END *
6444345.30509938) + (CASE WHEN ("genre" = 'Crime') THEN (1.0) WHEN NOT("genre" =
'Crime') THEN (0.0) END * 457224.713568542) + (CASE WHEN ("genre" = 'Drama') THEN (1.0)
WHEN NOT("genre" = 'Drama') THEN (0.0) END * -578623.648511504) + (CASE WHEN ("genre" =
'Fantasy') THEN (1.0) WHEN NOT("genre" = 'Fantasy') THEN (0.0) END * 939211.031010849)
+ (CASE WHEN ("genre" = 'Horror') THEN (1.0) WHEN NOT("genre" = 'Horror') THEN (0.0)
END * 14026970.3814342) + (CASE WHEN ("genre" = 'Mystery') THEN (1.0) WHEN NOT("genre"
= 'Mystery') THEN (0.0) END * 4661390.51760519) + (CASE WHEN ("genre" = 'Other') THEN
(1.0) WHEN NOT("genre" = 'Other') THEN (0.0) END * -1875298.29135123) + (CASE WHEN
("company" = 'Columbia Pictures Corporation') THEN (1.0) WHEN NOT("company" = 'Columbia
Pictures Corporation') THEN (0.0) END * -2211850.15894333) + (CASE WHEN ("company" =
'Metro-Goldwyn-Mayer (MGM)') THEN (1.0) WHEN NOT("company" = 'Metro-Goldwyn-Mayer
(MGM)') THEN (0.0) END * -7780415.53455518) + (CASE WHEN ("company" = 'New Line
Cinema') THEN (1.0) WHEN NOT("company" = 'New Line Cinema') THEN (0.0) END *
```

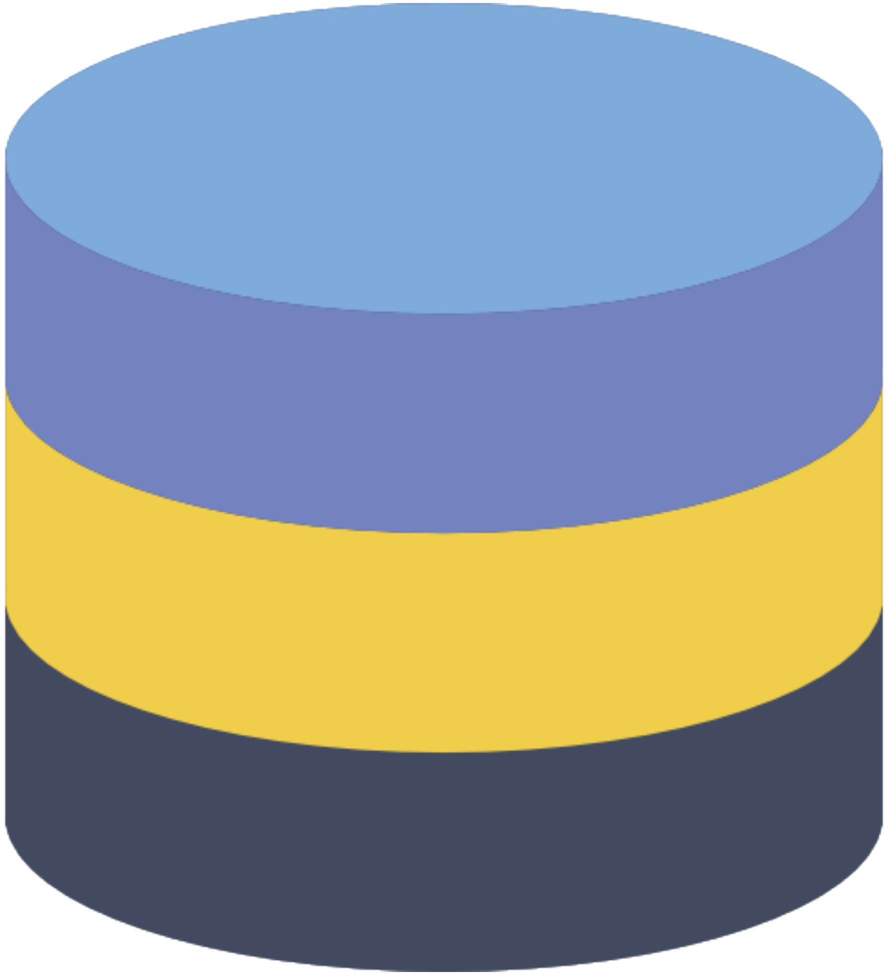
# Demo



```
  labs(title = "Movies by Genre",
       x = "Genre",
       y = "Count") +
  coord_flip()
```
```

```
library(odbc)
con <- dbConnect(odbc::odbc(), "Teradata")
```

```
~/Documents/RStudio/repos/teradata-2019/
 NOT("company" = 'Twentieth Century Fox Film Corporation') THEN (0.0) END * 11068929.2792623) + (CASE WHEN ("co
mpany" = 'Universal Pictures') THEN (1.0) WHEN NOT("company" = 'Universal Pictures') THEN (0.0) END * 4769016.1
5974144) + (CASE WHEN ("company" = 'Walt Disney Pictures') THEN (1.0) WHEN NOT("company" = 'Walt Disney Picture
s') THEN (0.0) END * 8242555.32416439) + (CASE WHEN ("company" = 'Warner Bros.') THEN (1.0) WHEN NOT("company"
 = 'Warner Bros.') THEN (0.0) END * 2266061.21752693)
> dbDisconnect(con)
> con <- dbConnect(
+   odbc::odbc(),
+   "Teradata"
+ )
>
```

| Name | Size | Modified |
|---|---|---|
| .. | | |
| .gitignore | 40 B | Sep 16, 2019, 1 |
| .Rhistory | 2 KB | Oct 21, 2019, 2 |
| .Rprofile | 26 B | Sep 16, 2019, 1 |
| data | | |
| R | | |
| README.md | 80 B | Oct 18, 2019, 4 |
| renv | | |
| renv.lock | 34.4 KB | Oct 18, 2019, 5 |
| slides | | |
| teradata-2019.Rproj | 205 B | Oct 21, 2019, 1 |
```

# Resources

- db.rstudio.com
- github.com/tidymodels
- github.com/blairj09/teradata-2019
- tdplyr



Icons from flaticon