data_607_assignmenttwo_Kevin_Kirby

2024-09-07

Assignment Two Overview

For this assignment, I need to:

- Conduct a survey of at least 5 friends about at least 6 movies and have them rate on a scale 1-5, with movies they haven't seen left blank
- load those results into a SQL database of my choosing
  - I will use PostgreSQL (PGS)
- Connect RStudio to PGS and import the SQL table as a dataframe
- Perform data prep and explordatory data anlysis
  - Including handling missing values, which I will do in R

PGS Connection & Data Import

Table Creation in PGS

My survey of 5 friends about 6 recent movies yielded a table with seven rows and eight columns, for 27 cells total. An extremely small dataset in the grand scheme of the universe. I setup pgAdmin 4 (PGA-4) on my Macbook Pro and just used the GUI to create a database called "data_607" and then a schema and table.

This is the SQL to create the table:

```
-- Table: movie_survey_raw_results.raw_results

-- DROP TABLE IF EXISTS movie_survey_raw_results.raw_results;

CREATE TABLE IF NOT EXISTS movie_survey_raw_results.raw_results
(
    name text COLLATE pg_catalog."default" NOT NULL,
    deadpool_wolverine integer,
    dune_part_two integer,
    twisters integer,
    godzilla_kong_the_new_empire integer,
    quiet_place_day_one integer,
    inside_out_two integer,
    CONSTRAINT raw_results_pkey PRIMARY KEY (name)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS movie_survey_raw_results.raw_results
    OWNER to postgres;
```

This is the SQL to import a CSV from a local computer. I removed sensitive components of the file path name. You would need to ensure this points to where you store it.

```
COPY movie_survey_raw_results.raw_results (deadpool_wolverine, dune_part_two, twisters,
godzilla_kong_the_new_empire, quiet_place_day_one, inside_out_two)
```

```
FROM '[your_computer_folder]/607_homeworktwo_moviesurvey.csv'
DELIMITER ','
CSV HEADER;
```

I thought about making a connection to GCP to professionalize the setup a bit more but I decided against it since those charge by the hour, even when idle. Unless you're dilligent, the costs can escalate quickly.

Getting Data Ready

While I'm new to data science and to many facets of wrangling unstructured data, I'm not new to relational databases and how data should be stored in them. For instance, I saved myself some future pain by simply inputting the movie titles into my file as, for example, "deadpool_wolverine" instead of "Deadpool & Wolverine" become. This was a small survey where changing values manually was easy enough. Column names in a relational database should be lowercase and free of special characters, otherwise you'll have a bad time later on.

As has been my experience working with programmatic connections to Amazon Web Services or Google Cloud Platforms, the most difficult part of the connection and import was configuring the settings properly within PGS to ensure RStudio could access it. I also wanted to ensure the connection used RStudio global variables instead of the raw values.

This is the SQL I ran in PGA-4 to set the schema where the results live as the default. I did this after successfully establishing connection with PGS but nothing being return.

```
ALTER ROLE postgres SET search_path TO movie_survey_raw_results;
```

Connecting to PGS

I had to install the RPostgres and DBI packages as the foundation of the connection. I created global variable for the PGS username and password I need to pass in the connection request. After the connection, I tested it by looking at the value returned by dbListTables(), which should be "raw_results".

Hidden from your view (you, the reader) is the chunk I used to make a connection to PGS because it has my username and password. Here's the code I ran, with the user and password set to variables I set locally.

```
pgs_db <- dbConnect(
  RPostgres::Postgres(),
  host = "localhost",
  port = 5432,
  dbname = "data_607",
  user = pgs_username,
  password = pgs_pw
)
```

```
library(RPostgres)
library(DBI)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(purrr)
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
dbListTables(pgs_db)
```

```
## [1] "raw_results"
```

Turning SQL Table Into Dataframe

This is a clean line of code that imports all the data and assigns it to a variable. Normally running SELECT * from a table without some sort of limiter is very bad form. However, this dataset is very small and I need all the data anyway.

```
movie_results_raw_df <- dbGetQuery(pgs_db, "SELECT * FROM movie_survey_raw_results.raw_results")

kable(movie_results_raw_df) %>%
  kable_styling(full_width = FALSE, bootstrap_options = "striped")
```

name	deadpool_wolverine	dune_part_two	twisters	godzilla_kong_the_new_empire	quiet_place_day_one
MLS	NA	2	3	NA	NA
Zach	4	2	NA	3	NA
Ken	1	NA	1	NA	1
Chris	NA	NA	NA	5	2
Kellie	NA	NA	NA	NA	NA
Samuel	3	NA	4	NA	NA

We are now ready to cleanup the nulls and discuss next steps.

Lies, Damn Lies, and Nulls

There are a lot of options for dealing with null or missing values and it's highly dependent on the data you have. Here are a few with some commentary on application here:

- Mean Imputation: assinging the average rating for the movie as the value for any nulls for that movie
  - This is a basic but straightforward logic and is a good choice for the small and limited dataset I have. It's a bit of brute force and isn't tied to anything other than a small dataset for that movie.
- Collaborative Filtering: predict missing values based on participant preferences and movie patterns
  - If I had a large enough dataset and a bit more data about the user, this is the approach I would take. Taking in the context of the user and the movie to determine a value would create a better outcome.
- Custom Logic: create rules for how to assign values and program them into your system
  - Using business logic to define specific data values is very common. In my experience, the most intense conversations in the private sector are those about the definition of a metric itself.

```r
movie_results_mn <- movie_results_raw_df %>%
  mutate_all(~ ifelse(is.na(.), round(mean(., na.rm = TRUE)), .))

kable(movie_results_mn) %>%
  kable_styling(full_width = FALSE, bootstrap_options = "striped")
```

name	deadpool_wolverine	dune_part_two	twisters	godzilla_kong_the_new_empire	quiet_place_day_one
MLS	3	2	3	4	2
Zach	4	2	3	3	2
Ken	1	2	1	4	1
Chris	3	2	3	5	2
Kellie	3	2	3	4	2
Samuel	3	2	4	4	2

We now have a dataset that is full of values and could be used in further analysis.

Closing Thoughts

I had to cut for time my desire to try and "create a normalized set of tables that corresponds to the relationship between your movie viewing friends and the movies being rated." It's very rare that all the data you need is in one table and, when it is, there are usually major performance issues. A company like Meta or Google is generating petabytes of data a day and that has to placed in mutliple data models with multiple different components within it.