assignment_3a

Table of contents

Global Baseline Estimates

Using the information you collected on movie ratings, implement a Global Baseline Estimate recommendation system in R. The attached spreadsheet provides the implementation algorithm.

Most recommender systems use personalized algorithms like "content management" and "item-item collaborative filtering." Sometimes non-personalized recommenders are also useful or necessary. One of the best non-personalized recommender system algorithms is the "Global Baseline Estimate.

The job here is to use the survey data collected and write the R code that makes a movie recommendation using the Global Baseline Estimate algorithm. Please see the attached spreadsheet for implementation details.

Movie Ratings XLSX"

Approach

Review

I'll start by reviewing the excel.

There's 4 sheets:

MovieRatings

- Survey of the list of movies

Problem Statement

- Just seems like survey
- No movie title keys
- Keys for names of people taking the survey

MeanCenteredMovieRatings

- First table takes the mean rating per person based on the movies they rated
- small subset of all the critics
- Second table takes the deviation from the mean per rating per person

Global Baseline

- user average
    - average(movie rating) per row
    - last row is the average of all movies
- user average - mean
    - user average - total movie average
- total movie average
    - Takes the average per row, ignores NA
- Movie avg
    - average rating per movie
- movie avg - mean movie
    - movie average - total average
- How would Param rate Pitch Perfect 2?
    - Global Baseline Estimate =
        - Mean movie rating +
        - Pitch Perfect 2's rating relative to average +
        - Param's rating relative to average

So, outside of the movie references, the Global Baseline Estimates are:

- Expected value = Grand Mean + Row Effect + Column Effect
    - Expected value = F10

        - Value we are trying to predict.
    - Grand Mean = H18
        - Overall effect
    - Row effect = I10
        - Group A effect
    - Column effect = F19
        - Group B effect

It's pretty interesting, apparently *it's just variance decomposition.* It's pretty intuitive, you are predicting a cell, so you take the variance from the row, column, and the entire table to inform that prediction. The model is applied all over the place, because it's a general pattern structure that separates systematic structure (predictable patterns) from randomness.

ANOVA

$$SS_{Total} = SS_{Rows} + SS_{Columns} + SS_{Residual}$$

—

Implementation

- Import rating data from PGSQL as df

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

load_dot_env()

con <- dbConnect(
  RPostgres::Postgres(),
  dbname = Sys.getenv("DB_NAME"),
  host = Sys.getenv("DB_HOST"),
  port = Sys.getenv("DB_PORT"),
  user = Sys.getenv("DB_USER"),
  password = Sys.getenv("DB_PASSWORD")
)
```

Connection Test

```
dbGetQuery(con, "SELECT version();")
```

```
                                                                   version
1 PostgreSQL 17.6 on x86_64-windows, compiled by msvc-19.44.35213, 64-bit
```

Creating csv from df

```
query <- "SELECT * FROM popular_movies.v_ratings_raw"
df <- dbGetQuery(con, query) |>
  as_tibble()

df |> select(name, title, rating)
```

```
# A tibble: 30 x 3
   name  title                    rating
   <chr> <chr>                     <int>
 1 Alex  One Battle After Another      5
 2 Alex  Begonia                       4
 3 Alex  Wicked for Good               4
 4 Alex  The Materialist               3
 5 Alex  Sinners                      NA
 6 Bri   One Battle After Another      4
 7 Bri   Begonia                       3
 8 Bri   Wicked for Good               5
 9 Bri   The Materialist              NA
10 Bri   Sinners                       4
# i 20 more rows
```

```
write.csv(df, "movie_ratings.csv", row.names = FALSE)
```

So the data now lives in the folder. I'll just clean up the df.

```
df <- read.csv("movie_ratings.csv")
df <- df |> select(name, title, rating)
df
```

```
   name                    title rating
1  Alex One Battle After Another      5
2  Alex                  Begonia      4
```

```
3  Alex          Wicked for Good      4
4  Alex          The Materialist      3
5  Alex                  Sinners     NA
6   Bri One Battle After Another      4
7   Bri                  Begonia      3
8   Bri          Wicked for Good      5
9   Bri          The Materialist     NA
10  Bri                  Sinners      4
11 Chen One Battle After Another     NA
12 Chen                  Begonia      5
13 Chen          Wicked for Good      4
14 Chen          The Materialist      3
15 Chen                  Sinners      4
16 Devi One Battle After Another      3
17 Devi                  Begonia     NA
18 Devi          Wicked for Good      3
19 Devi          The Materialist      4
20 Devi                  Sinners      5
21  Eli One Battle After Another      4
22  Eli                  Begonia      3
23  Eli          Wicked for Good     NA
24  Eli          The Materialist      5
25  Eli                  Sinners      3
26 Fran One Battle After Another      5
27 Fran                  Begonia      3
28 Fran          Wicked for Good      4
29 Fran          The Materialist      4
30 Fran                  Sinners     NA
```

So I have name, title, and rating. I want to create a function called global_baseline_estimate

```
#global_baseline_estimate() <- function(df){}
```

So, it would need to do the following:

- create summarization by name (df_name), get mean rating (na.rm = TRUE) per name (n_mean)
- create summarization by title (df_title), get mean rating (na.rm = TRUE) per title (t_mean)
- create variable for the mean rating of all titles (x)
- mutate name summarization (df_name) to calculate effect (n_effect) = (n_mean - x)
- mutate title summarization (df_title) to calculate effect (t_effect) = (t_mean - x)

- join df_name$n_effect by name
- join df_title$t_effect by title
- mutate df (gbe) by rating: if na then x + n_effect + t_effect else rating

That should get me a completed dataset where na values are filled with ratings from a global baseline estimate. Pretty neat.

Codebase