Filtering and preparing the data for analysis

2023-10-04

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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
```

```r
# Read merged data csv file
# Import the dataset
csv_file <- "merged_dataset_unique.csv"
folder <- "../../gen/data-preparation"

folder_path <- file.path(folder, csv_file)

merged_dataset_unique <- read.csv2(folder_path)
```

Data preparation

Renaming and deleting unnecessary variables

There are a lot of unnecessary variables in the merged dataset that we will not be using.

```r
# Delete and rename variables
filtered_merged_dataset <- merged_dataset_unique %>%
  filter(titleType == "movie")
filtered_merged_dataset <- filtered_merged_dataset %>%
  select(-isAdult)
filtered_merged_dataset <- filtered_merged_dataset %>%
  select(-startYear)
filtered_merged_dataset <- filtered_merged_dataset %>%
  select(-tconst)
filtered_merged_dataset <- filtered_merged_dataset %>%
  select(-titleType)
filtered_merged_dataset <- filtered_merged_dataset %>%
  select(-originalTitle)
```

```
filtered_merged_dataset <- filtered_merged_dataset %>%
  select(-knownForTitles)
filtered_merged_dataset <- filtered_merged_dataset %>%
  select(-primaryName)
filtered_merged_dataset <- filtered_merged_dataset %>%
  rename(movie = name)
filtered_merged_dataset <- subset(filtered_merged_dataset, complete.cases(filtered_merged_dataset))
```

Create new variables

RuntimeMinutes dummy

We construct a dummy of the runtime per minutes where the dummy equals 1 if the runtime in minutes is above the median.

```
# Transform opening gross into a numeric variable and remove dollar signs
filtered_merged_dataset$openinggross <- as.numeric(gsub("[\\$,]", "", filtered_merged_dataset$openinggr

# Make runTimeMinutes a dummy variable by using a median split
  # Calculate median of variable 'runtimeMinutes'
filtered_merged_dataset$runtimeMinutes <- as.numeric(filtered_merged_dataset$runtimeMinutes)
median_runtime <- median(filtered_merged_dataset$runtimeMinutes)
  # Construct a dummy variable for the runtime per minutes
filtered_merged_dataset$runtime_dummy <- ifelse(filtered_merged_dataset$runtimeMinutes <= median_runtim
```

Director count

Amount of how many movies the director has directed

```
# Count how many times a director has directed a movie that is in the dataset
filtered_merged_dataset$director_count <- ave(filtered_merged_dataset$director_identifier, filtered_merg
# Calculate the median of director_count
filtered_merged_dataset$director_count <- as.numeric(filtered_merged_dataset$director_count)
median_director_count <- median(filtered_merged_dataset$director_count)
filtered_merged_dataset$directed_above_median <- ifelse(filtered_merged_dataset$director_count <= mediar
```

Converting merged_dataset_unique into a csv file

```
# Define folder for CSV file
fileplace <- "../../gen/analysis/filtered_merged_dataset.csv"

# CSV to input folder for analysis
write_csv2(filtered_merged_dataset, file = fileplace)
```