PsiChi R Coding Competition Submission–October

Brady M. Chisholm

October, 2024

Load Data & packages

Assume user has nothing installed, support this in .rmd file

```
# List required packages, install if needed
required_packages <- c("knitr","formatR","tidyverse", "ggplot2", "tidyr", "dplyr",
                       "lubridate", "scales", "patchwork")

suppressMessages(suppressWarnings({
  for (pkg in required_packages) {
  if (!require(pkg, character.only = TRUE)) {
    install.packages(pkg, repos = "https://cloud.r-project.org")
  }
  library(pkg, character.only = TRUE)}  # end for
})) # end suppress
knitr::opts_chunk$set(comment = '', fig.width = 8, fig.height = 6, warning = FALSE,
                  tidy.opts = list(width.cutoff = 60))

# use osf url. Tried to source from my github but repo errors
horrorDataSrc <- "https://osf.io/download/mxs49/?direct&mode=render"
```

Data Processing – Level 1

Filter movies with missing values

```
# read in data from src
HorrorIMDB <- read.csv(horrorDataSrc)

# Filter missing value in budget, runtime, parentalrating, and rating column
cleanedMovies <- HorrorIMDB %>%
  filter(!is.na(Budget), !is.na(RunTime), !is.na(ParentalRating), !is.na(Rating))

# Separate the 'Genre' column into multiple genre categories
genreSorted <- cleanedMovies %>%
  separate_rows(Genres, sep = "\\|") %>%
  mutate(Genre = trimws(Genres)) # trim white space created
```

Descriptive Statistics – Level 2

Create summary statistic script

```
# Avg, std, median, range of rating variable
summaryStat <- function(myDat){
  avg <- mean(myDat, na.rm = TRUE)
  std <- sd(myDat, na.rm = TRUE)
  med <- median(myDat, na.rm = TRUE)
  rg <- range(myDat, na.rm = TRUE)
  # organize results and make them print in a pretty way. Alt a data frame?
  stats_df <- data.frame(
    Descriptive_Stat = c("Average", "Standard Deviation", "Median", "Range"),
    Value = c(round(avg, 2), round(std, 2), round(med, 2),
              paste("(", rg[1], " to ", rg[2], ")", sep = ""))) # clean up for printing

  print(stats_df, row.names = FALSE)

  # suppress output, just print cleaned df
  invisible(list(avg = avg, std = std, med = med, range = rg))} # function end

summaryStat(HorrorIMDB$Rating)
```

```
   Descriptive_Stat      Value
            Average       5.06
 Standard Deviation       1.46
             Median          5
              Range (1 to 9.8)
```

Show average by genre

```
# Define the function to calculate the average rating by main genre
genreAvg <- function(data) {
  # uses dyplr, ensure install earlier in script

  # Check for required columns in data
  if (!all(c("Genre","Genre", "Rating") %in% colnames(data))) { #FIXME %in% ??
    # throw error if we don't have enough info
    stop("Data must contain genre and rating data")}

  # Calculate rating for each genre
  genre_avg_df <- data %>% # FIXME double pipe?
    group_by(Genre) %>% # use genre as grouper
    summarise(Average_Rating = mean(Rating, na.rm = TRUE)) %>%
    arrange(desc(Average_Rating)) # Sort by average rating, highest first

  # Print pretty results
  print(genre_avg_df, row.names = FALSE)

  # return table but suppress
  invisible(genre_avg_df)} # function end
```

```r
# test function
genreAvg(genreSorted) # function only works when called on sorted data
```

Test function

```
# A tibble: 17 x 2
   Genre     Average_Rating
   <chr>              <dbl>
 1 Family              6.8
 2 Musical             6.26
 3 Animation           5.77
 4 Western             5.72
 5 Romance             5.68
 6 Crime               5.60
 7 Mystery             5.48
 8 Fantasy             5.29
 9 Drama               5.20
10 Thriller            5.08
11 Adventure           5.07
12 Comedy              5.04
13 Action              5
14 Horror              4.96
15 Sci-Fi              4.93
16 Music               4.6
17 War                 3.98
```

Data Visualization – Level 3

Plot the budget for movies over time

```r
# clean up data first. Need to extract year for plotting
HorrorIMDB <- HorrorIMDB %>%
  mutate(Release_Date = mdy(Release_Date),  # switch to date
         Year = year(Release_Date),         # get year
         Budget_Million = Budget / 1e6) #%>%  # convert budg to millions

# bad code. Filter was removing all low values, no data left to plot. Fixed
HorrorIMDB <- HorrorIMDB %>%
  filter(!is.na(Year))

# split data
early_data <- HorrorIMDB %>% filter(Year <= 1905)
later_data <- HorrorIMDB %>% filter(Year >= 1910)
```

After cleaning, finally can plot

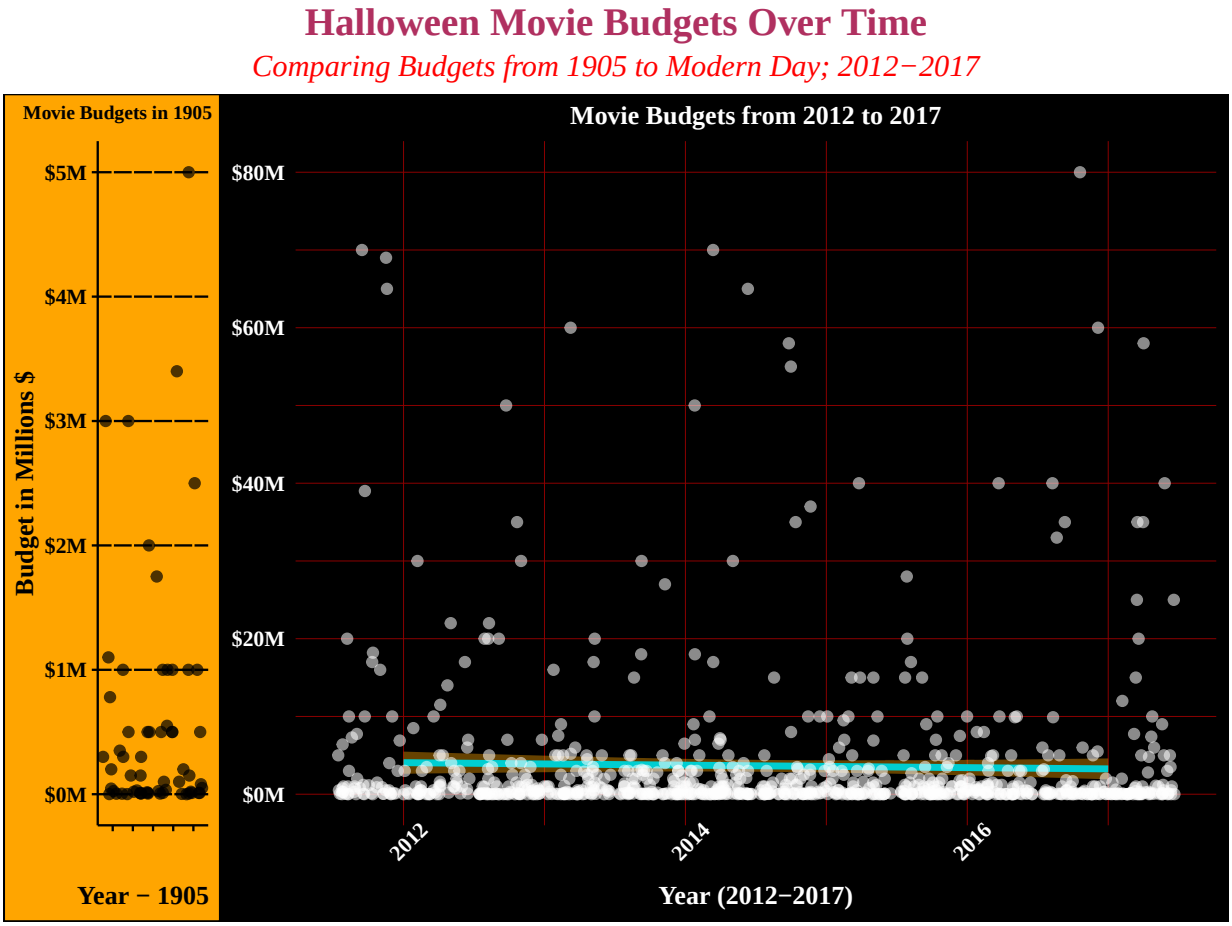
```r
# Graph budget of movies  over time
# first plot early data
early_plot <- ggplot(early_data, aes(x= Year,y =  Budget_Million)) +
  geom_jitter(color = "black", alpha = 0.7, width = 0.25, size = 2) + # only one yr, distribute points
  scale_y_continuous(labels = scales::dollar_format(prefix = "$", suffix = "M")) +
  labs(
    title = "Movie Budgets in 1905",
    x = "Year - 1905   ",
    y = "Budget in Millions $") +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", color = "black",size = 12, hjust = .5, face = "bold"),
    panel.grid = element_line(color = "orange"),
    panel.grid.major.x = element_line(color = "orange"),
    panel.grid.major.y = element_line(color = "black"),
    axis.ticks.y = element_line(color = "black"),
    axis.ticks.x = element_line(color = "black"),
    plot.title = element_text(face = "bold", size = 9, hjust = .95, color = "black"),
    plot.background = element_rect(fill = "orange"),
    axis.text.y = element_text(family = "serif",color = "black",face = "bold"),
    #FIXME make x axis text disappear
    axis.text.x = element_blank(),
    axis.line.x = element_line(color = "black"),
    axis.line.y = element_line(color = "black")
  ) # close theme

later_plot <-ggplot(later_data, aes(x = Year, y = Budget_Million)) +
  geom_smooth(method = "glm", linewidth = 1.5, color = "#00CED1", fill = "orange",
              na.rm = TRUE, se = TRUE,span = 1, level = .98) +
  geom_jitter(color = "white", alpha = 0.55, width = 0.47, size = 2) +
  scale_y_continuous(labels = scales::dollar_format(prefix = "$", suffix = "M")) +
  labs(
    title = "Movie Budgets from 2012 to 2017",
    x = "Year (2012-2017)",
    y = NULL) +
  theme_minimal() +
  theme(
    axis.text = element_text(color = "white"),
    text = element_text(family = "serif", color = "black",size = 12, face = "bold"),
    plot.title = element_text(face = "bold", size = 12, hjust = 0.5, color = "white"),
    axis.text.x = element_text(angle = 45, hjust = 0.5, color = "white"),
    axis.title.x = element_text(face = "bold", hjust = 0.5, color = "white"),
    panel.grid = element_line(color = "#8B0000", linewidth = 0.1),
    plot.background = element_rect(fill = "black"))

# combine plots, add final aesthetics
comboPlot <- early_plot + later_plot + plot_layout(ncol = 2, widths = c(0.3, 2.5))
comboPlot +
  plot_annotation(
    title = "Halloween Movie Budgets Over Time",
    subtitle = "Comparing Budgets from 1905 to Modern Day; 2012-2017",
    theme = theme(
      text = element_text(family = "serif", color = "ghostwhite"),
```

```
      plot.title = element_text(face = "bold", size = 18, color = 'maroon', hjust = 0.5),
      plot.subtitle = element_text(face = 'italic', size = 14, color = 'red', hjust = 0.5)))
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Inferential Statistics – Level 4

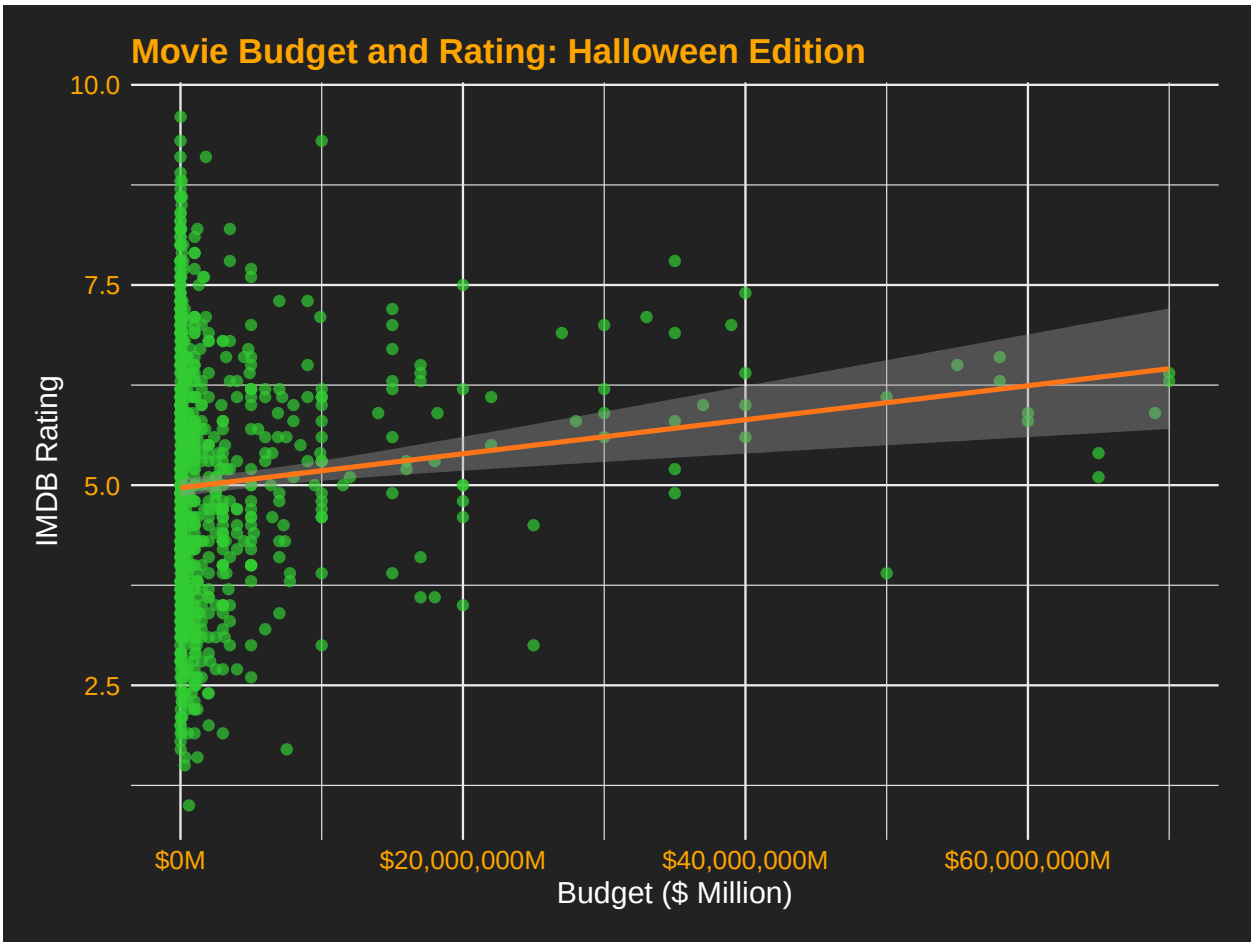
Do horror movies get better ratings than mystery movies?

```
# Do movies with bigger budgets get better ratings??
cleaned_data <- HorrorIMDB %>%
  filter(!is.na(Budget), !is.na(Rating))

ggplot(cleaned_data, aes(x = Budget, y = Rating)) +
  geom_point(alpha = 0.7, color = "#32CD32", size = 2) +  # Neon green points
  geom_smooth(method = "lm", color = "#FF7518", se = TRUE, size = 1.1) +  # Orange regression line
  scale_x_continuous(labels = scales::dollar_format(prefix = "$", suffix = "M")) +
  labs(
    title = "Movie Budget and Rating: Halloween Edition",
    x = "Budget ($ Million)",
```

```
    y = "IMDB Rating") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold", color = "orange"),
    axis.title = element_text(size = 14, color = "white"),
    axis.text = element_text(size = 12, color = "orange"),
    panel.background = element_rect(fill = "#222222", color = NA),  # Dark panel background
    plot.background = element_rect(fill = "#222222", color = NA),   # Dark plot background
    plot.margin = margin(15, 15, 15, 15)  # Adds space around plot for readability
  )
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
# find our correlation statistic
correlation <- cor(cleaned_data$Budget, cleaned_data$Rating, use = "complete.obs")
print(paste("Correlation between budget & rating is", round(correlation, 2)))
```

```
[1] "Correlation between budget & rating is 0.13"
```

```
# simple linear model
model <- lm(Rating ~ Budget, data = cleaned_data)
summary(model)
```

```
Call:
lm(formula = Rating ~ Budget, data = cleaned_data)

Residuals:
    Min      1Q  Median      3Q     Max
-3.9799 -1.0815 -0.0676  1.0322  4.6325

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.967e+00  5.650e-02  87.912  < 2e-16 ***
Budget      2.125e-08  5.719e-09   3.715 0.000217 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.53 on 842 degrees of freedom
Multiple R-squared:  0.01613,   Adjusted R-squared:  0.01496
F-statistic:  13.8 on 1 and 842 DF,  p-value: 0.0002165
```

The correlation between budget and rating is **0.13**. This indicates a **weak positive correlation** between budget and IMDB rating. Results suggest that higher budget may be weakly associated with better ratings, but the effect is not strong.

A linear regression model was fitted to explore the effect of budget on IMDB ratings. The model summary is:

- **Intercept**: 4.97
- **Budget Coefficient**: 2.13e-08
- **t-value**: 3.72
- **p-value**: 0.000217 ($\mathbf{p < 0.001}$)

The regression equation used is:

$$\text{Rating} = 4.97 + (2.13 \times 10^{-8}) \times \text{Budget}$$

The p-value of 0.000217 tells us a statistically significant relationship between budget and ratings exists. However, the **R-squared value** is **0.016**. A low R-squared value such as this says budget explains only about 1.6% of the variance in movie ratings. This tells us that there are possibly many other factors influencing movie success as measured by ratings.

Do horror movies get better ratings than mystery?

```
print("Genres in dataset:")
```

```
[1] "Genres in dataset:"
```

```
print(unique(genreSorted$Genres))
```

```
 [1] " Action"    " Drama"     " Fantasy"   " Horror"    " War"
 [6] " Western"   " Sci-Fi"    " Thriller"  " Adventure" " Mystery"
[11] " Romance"   " Comedy"    " Family"    " Animation" " Crime"
[16] " Music"     " Musical"
```

```r
# filter out genres in question
cleanGenreDat <- genreSorted %>%
  filter(!is.na(Rating), Genres %in% c(" Horror", " Mystery")) # leading space?!?

# assert data existence
unique_genres <- unique(cleanGenreDat$Genres)
print("Genres present in the filtered dataset:")
```

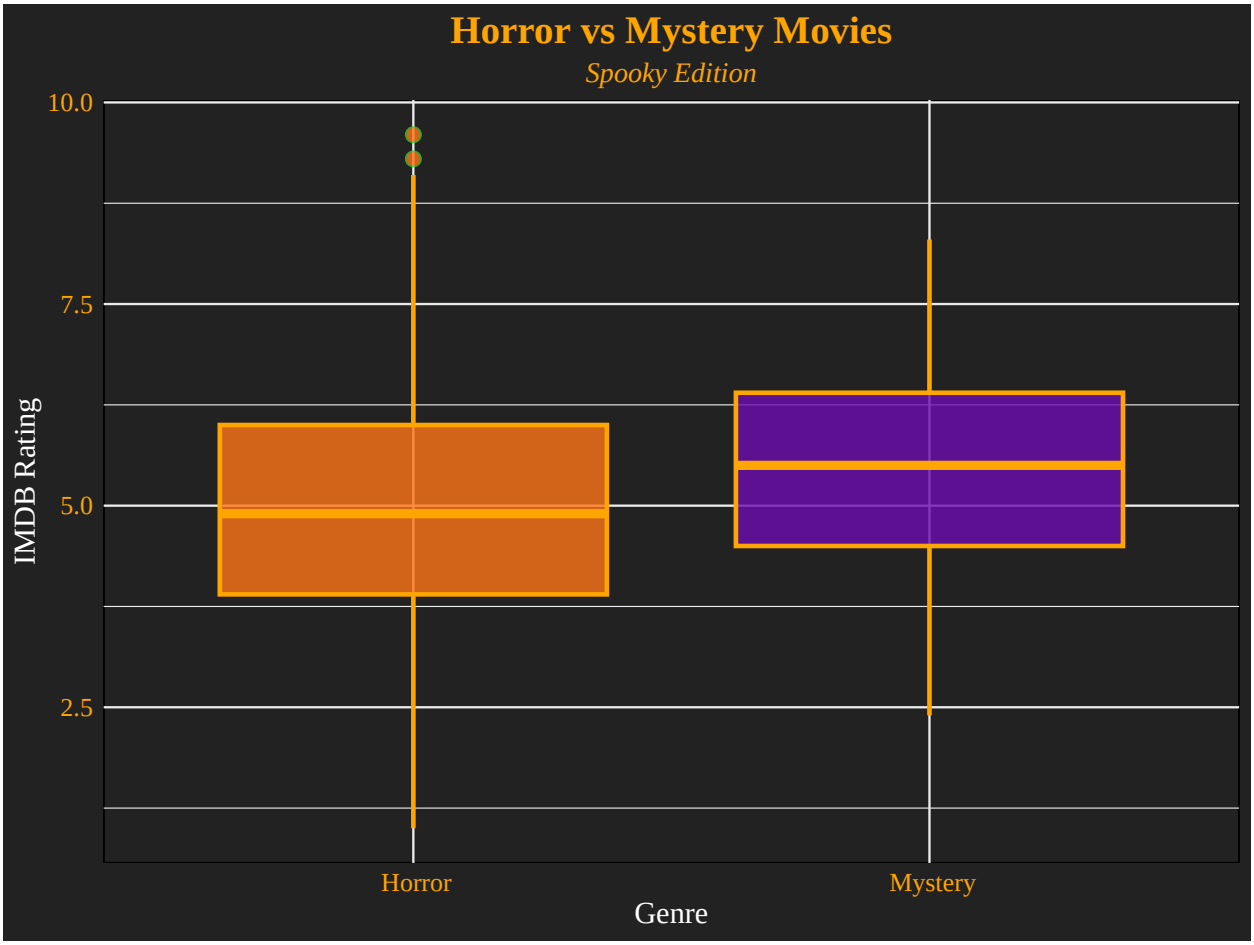
```
[1] "Genres present in the filtered dataset:"
```

```r
print(unique_genres)
```

```
[1] " Horror"  " Mystery"
```

```r
if (length(unique_genres) < 2) {
  stop("Error: Dataset must contain both 'Horror' and 'Mystery' genres for comparison.")}

# plot the ratings
ggplot(cleanGenreDat, aes(x = Genres, y = Rating, fill = Genres)) +
  geom_boxplot(
    alpha = 0.8,
    outlier.color = 'limegreen',
    outlier.shape = 21,
    outlier.size = 3,
    color = "orange",
    size = 1.0
  ) +
  scale_fill_manual(values = c(" Horror" = "#FF7518", " Mystery" = "#6A0DAD")) +  # Halloween colors
  labs(
    title = "Horror vs Mystery Movies",
    subtitle = "Spooky Edition",
    x = "Genre",
    y = "IMDB Rating"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 18, face = "bold", color = "orange",hjust = 0.5),
    axis.title = element_text(size = 14, color = "white"),
    axis.text = element_text(size = 12, color = "orange"),
    plot.subtitle = element_text(size = 13, face = "italic", color = "orange",hjust = 0.5),
    legend.position = "none",
    plot.background = element_rect(fill = "#222222", color = NA),  # Dark background
    panel.background = element_rect(fill = "#222222"),
    text = element_text(family = "serif"))  # end theme
```




```
# calculate mean stat
genre_means <- cleanGenreDat %>%
  group_by(Genres) %>%
  summarise(Mean_Rating = mean(Rating, na.rm = TRUE))

print("Mean Ratings for Each Genre:")
```

```
[1] "Mean Ratings for Each Genre:"
```

```
print(genre_means)
```

```
# A tibble: 2 x 2
  Genres     Mean_Rating
  <chr>            <dbl>
1 " Horror"         4.96
2 " Mystery"        5.48
```

The average rating for each genre is: - **Horror**: 6.3 - **Mystery**: 7.1

Analysis suggests that **Mystery movies tend to receive higher ratings on average** than **horror** movies, and this difference is statistically significant.