Movie analysis code

**Page set-up (yaml)**

---

title: "Movie ratings and box office performance"

author: "Name"

date: "Date"

output: html\_document

---

Note to change to Microsoft word use: output: word\_document

**“renv” package for reproducibility**

```{r Install renv for reproducibility, include=FALSE}

#Install renv package if needed

if(!require(renv)){install.packages('renv')}

```

```{r initializing renv, include=FALSE}

renv::init()

```

**Packages**

```{r Install and load packages, include=FALSE}  
 #Install tidyverse package if needed  
 if(!require(tidyverse)){install.packages('tidyverse')}  
 #Load tidyverse package  
 library(tidyverse)  
  
 #Install tables package if needed  
 if(!require(tables)){install.packages('tables')}  
 #Load tables package  
 library(tables)  
  
 #Install flextable package if needed  
 if(!require(flextable)){install.packages('flextable')}  
 #Load flextable package (as\_flextable)  
 library(flextable)  
  
 #Install arsenal package if needed  
 if(!require(arsenal)){install.packages('arsenal')}  
 #Load arsenal package (for tableby)  
 library(arsenal)  
  
 #Install modelsummary package if needed  
 if(!require(modelsummary)){install.packages('modelsummary')}  
 #Load modelsummary package (for datasummary\_crosstab)  
 library(modelsummary)  
  
 #Install RColorBrewer package if needed  
 if(!require(RColorBrewer)){install.packages('RColorBrewer')}  
 #Load RColorBrewer package  
 library(RColorBrewer)  
 ```

**Read data in**

```{r Read in csv file}

Movie\_db <- read\_csv("Movie\_database.csv", skip\_empty\_rows = TRUE)

```

**Data processing and labelling**

### Processing of data

```{r}

Movie\_db$Genre <- as.factor(Movie\_db$Genre)

Movie\_db$Best\_picture <- as.factor(Movie\_db$Best\_picture)

Movie\_db$Genre <- fct\_collapse(Movie\_db$Genre, Crime\_mystery = c("Crime", "Mystery"))

# head(Movie\_db) # If you would like to check your dataset, just delete the hashtag before 'head' at the beginning of this line.

```

**Create dataframe with selected variables**

```{r Create dataframe for tables}

Movies\_genre <- Movie\_db %>% dplyr::select(c(Genre, IMDB\_Rating, MetaCritic, Budget\_m, Runtime, Worldwide\_earning\_m, Best\_picture))

```

**Table 1. Movie viewer and critic ratings, runtime, budget, and earnings per genre**

```{r Genre Table}

set\_flextable\_defaults(background.color = "white")

tab <- tabular(

(Genre + 1) ~ (n = 1) + Format(digits = 1) \*

(IMDB\_Rating + MetaCritic + Runtime + Budget\_m + Worldwide\_earning\_m) \* (mean + sd),

data = Movies\_genre)

Genre\_tab <- as\_flextable(tab,) %>%

autofit()

Genre\_tab

```

*Saving table as an image*

First find dimensions:

```{r Table 1 dimensions}

dims <- flextable\_dim(Genre\_tab)

dims

```

Then use save\_as\_image to save table as a png file (this saves to the Project Folder)

```{r Table 1 save, fig.height=8.1, fig.width=4.1}

save\_as\_image(Genre\_tab, path = "Movie data by genre.png")

```

**Table 2. Movie viewer and critic ratings, runtime, budget, and earnings by best picture nomination**

```{r Best Picture Table}

set\_flextable\_defaults(background.color = "white")

tab\_bp <- tabular(

(Best\_picture + 1) ~ (n = 1) + Format(digits = 1) \*

(IMDB\_Rating + MetaCritic + Runtime + Budget\_m + Worldwide\_earning\_m) \* (mean + sd),

data = Movies\_genre)

BP\_tab <- as\_flextable(tab\_bp) %>%

autofit()

BP\_tab

```

*Save table:*

```{r Table 2 dimensions}

dims <- flextable\_dim(BP\_tab)

dims

```

```{r Table 2 save, fig.height=7.9, fig.width=2.1}

save\_as\_image(BP\_tab, path = "Movie data by best picture.png")

```

**Table 3. Best picture nominations by genre**

```{r Table of best picture nominations by genre}

options("modelsummary\_factory\_default" = "flextable")

Genre\_ctabs <- datasummary\_crosstab(Best\_picture ~ Genre, data = Movie\_db)

Genre\_ctabs

```

*Save table:*

```{r Table 3 dimensions}

dims <- flextable\_dim(Genre\_ctabs)

dims

```

```{r Table 3 save, fig.height=7.5, fig.width=1.75}

save\_as\_image(Genre\_ctabs, path = "Best picture nominations by movie genre.png")

```

Optional table (if I can fix width issue)

```{r Table, results="asis"}

mylabels <- list(Best\_picture = "Best picture", MetaCritic = "Meta critic score", IMDB\_rating = "IMDB rating", Runtime = "Run time (mins)", Budget\_m ="Budget, million $", Worldwide\_earning\_m = "Global earnings, million $")

tab1 <- tableby(Genre ~ Best\_picture + IMDB\_Rating + MetaCritic + Runtime + Budget\_m + Worldwide\_earning\_m, data=Movie\_db, test = FALSE)

summary(tab1, labelTranslations = mylabels, digits = 1)

```

**Figure 1. The relationship between critics opinions and how much a movie earns**

```{r Genre plot}

Plot\_1 <- ggplot(Movie\_db, aes(x=MetaCritic, y=Worldwide\_earning\_m, color = Genre)) +

geom\_point() +

theme\_classic()

Genre\_plot <- Plot\_1 + ggtitle("The relationship between critics opinions and how much a movie earns") +

xlab("Meta Critic Score") +

ylab("Global Earnings in millions ($)") +

scale\_color\_brewer(palette = "Set1")

Genre\_plot

```

*Save plot*

```{r Genre plot save}

ggsave("Critic ratings and movie earnings.png", plot = Genre\_plot, width = 9.0, height = 6.0, dpi = 300)

```

**Figure 2. The relationship between movie run time and movie earnings**

```{r Run time plot}

Plot\_2 <- ggplot(Movie\_db, aes(x=Runtime, y=Worldwide\_earning\_m, color = Genre)) +

geom\_point() +

theme\_classic()

Runtime\_plot <- Plot\_2 + ggtitle("The relationship between movie run time and movie earnings") +

xlab("Run time (mins)") + ylab("Global Earnings in millions ($)") +

scale\_color\_brewer(palette = "Set1")

Runtime\_plot

```

*Save:*

```{r Runtime plot save}

ggsave("Runtime and movie earnings.png", plot = Runtime\_plot, width = 9.0, height = 6.0, dpi = 300)

```

**Figure 3. Best picture nominations by genre**

```{r Barplot Best picture by genre}

p1 <- ggplot(data = Movie\_db, aes(x = Genre, fill = Best\_picture)) +

geom\_bar(position = "dodge") +

scale\_y\_continuous(limits = c(0, 30)) +

labs(subtitle = "Best picture nominations by genre") +

xlab("Genre") +

ylab("N Best picture nominations") +

theme\_classic()

p1

```

*Save:*

```{r Barplot Best picture by genre save}

ggsave("Barplot Best picture by genre.png", plot = p1, width = 7.0, height = 4.0, dpi = 300)

```

**Figure 4. Movie earnings by genre**

```{r Boxplot movie earnings by genre}

p <- ggplot(Movie\_db, aes(x=Genre, y=Worldwide\_earning\_m, color = Genre)) +

geom\_boxplot()

Box\_earnings <- p + theme\_classic() +

xlab("Movie genre") +

ylab("Worldwide earnings in millions ($)") +

stat\_summary(fun.y=mean, geom="point", shape=23, size=4, color = "red") +

scale\_fill\_brewer(palette="Set2")

Box\_earnings

```

*Save:*

```{r Boxplot movie earnings by genre save}

ggsave("Boxplot movie earnings by genre.png", plot = Box\_earnings, width = 9.0, height = 6.0, dpi = 300)

```