Academy Award Win Analysis

Andrew Norman

```
knitr::opts_chunk$set(echo = TRUE,
                      warning = FALSE,
                      message = FALSE)
library(tidyverse)
library(stringr)
library(broom)
library(patchwork)
library(scales)
library(ggrepel)
library(skimr)
library(tibble)
library(knitr)
library(caret)
library(modelsummary)
library(pROC)
```

1. First import the data after cleaning it in python.

(All data cleansing cells have been commented out since after completing the data frame it was saved and only the final data frame was called. The Python file and original base data frames are included, but takes a while to run and it will not run from start to finish without manually adding all of the films IMDB ids. We have included the file that is the step before adding the IDs and the step after adding the IDs, but there was no automation in that process. During this first step all that was done was changing the names of categories so that they are consistent throughout the years since they have changed over time.)

```
#oscar_data <- read.csv("~/Downloads/EC 422/Final_Project/oscar_data1.csv")
```

2. Convert str winner var to int win var and drop winner and winner2.

The winner variable to be an integer as opposed to a string so it can be used for analysis and get summary statistics on it if necessary and used as a factor variable for plots.

```
#filtered_oscar_data <- oscar_data %>%
  #filter(year_film >= 1970) %>%
  #mutate(winner2 = str_to_upper(winner)) %>%
  #mutate(win =
           #ifelse(winner2 == TRUE, 1, 0)) %>%
  #select(-X, -winner, -winner2)
```

3. Export as .csv and add in manual excel values, then import back into python and use Cinemagoer to append runtime, genre, box_office, and budget.

This is where most of the work the python file did is. After all of the individual IDs were added to the file, an IMDB .tsv file called `title.basics.tsv` was used to get runtime and genre information and the `Cinemagoer()` package was used to retrieve box office and budget information.

```
#write.csv(filtered_oscar_data, "oscar_data2.csv", row.names = TRUE)
```

4. Import inf_mult.csv to adjust values for inflation if necessary.

```
#inf_mult <- read.csv(("~/Downloads/EC 422/Final_Project/data/inflation_multiplier.csv"))
#inf_mult <- inf_mult %>%
  #rename(year = Year)
```

Created an inflation multiplier data frame for use if the box office and budget values were not adjusted for inflation (it turns out that they already were so it was not used in the final data frame).

5. Reimport Oscar dataframe, merge with inf_mult, then export to a new .csv and reimport to avoid redoing all these steps.

This was the cell where the final data frame was created. After all the automated work from the packages in python and the manual work of adding everything to the excel file, the data frames were merged and rewrote into a new file and then only that file was used past this point.

```
#oscar_data3 <- read.csv("~/Downloads/EC 422/Final_Project/data/oscar_data3.csv)
#oscar_data3 <- oscar_data3 %>%
  #rename(year = year_film,
         #winner = win) %>%
  #select(-X)

#left_join(oscar_data3, inf_mult, by = "year")

#write.csv(oscar_data3, "final_oscar_data.csv", row_names = TRUE)

final_od <- read.csv("~/Downloads/EC 422/Final_Project/data/final_oscar_data.csv")
```

6. Remove extra characters and convert box_office and budget to integers and drop the X column again.

```
final_od <- final_od %>%
  mutate(box_office = gsub("\\$", "", box_office),
         box_office = trimws(box_office),
```

```
         box_office = gsub(",", "", box_office),
         budget = gsub("\\$", "", budget),
         budget = gsub(",", "", budget),
         budget = gsub("\\(estimated\\)", "", budget),
         budget = trimws(budget)) %>%
  select(-X)

final_od$box_office <- ifelse(nchar(trimws(final_od$box_office)) == 0, NA, final_od$box_of)
final_od$budget <- ifelse(nchar(trimws(final_od$budget)) == 0, NA, final_od$budget)

final_od$box_office <- as.integer(final_od$box_office)
final_od$budget <- as.integer(final_od$budget)

head(final_od)
```

```
##   year year_ceremony ceremony      category
## 1 1970          1971       43  Best Picture
## 2 1970          1971       43  Best Actress
## 3 1970          1971       43 Best Director
## 4 1970          1971       43    Best Actor
## 5 1970          1971       43  Best Picture
## 6 1970          1971       43    Best Actor
##                                       name                       film
## 1                      Ross Hunter, Producer                    Airport
## 2                           Carrie Snodgress  Diary of a Mad Housewife
## 3                           Federico Fellini         Fellini Satyricon
## 4                             Jack Nicholson          Five Easy Pieces
## 5 Bob Rafelson and Richard Wechsler, Producers          Five Easy Pieces
## 6                             Melvyn Douglas I Never Sang for My Father
##   winner total_noms total_wins       ids runtime               genres
## 1      0         10          1 tt0065377     137 Action,Drama,Thriller
## 2      0          1          0 tt0065636      95          Comedy,Drama
## 3      0          1          0 tt0064940     129         Drama,Fantasy
## 4      0          4          0 tt0065724      98                 Drama
## 5      0          4          0 tt0065724      98                 Drama
## 6      0          3          0 tt0065872      92           Drama,Music
##   box_office   budget
## 1     371898 10000000
## 2    6100000  1200000
## 3         NA  3000000
## 4         NA  1600000
## 5         NA  1600000
## 6         NA       NA
```

7. View the structure of the data and skim it for general information.

```
str(final_od)
```

```
## 'data.frame':    1166 obs. of  14 variables:
```

```
##  $ year         : int  1970 1970 1970 1970 1970 1970 1970 1970 1970 1970 ...
##  $ year_ceremony: int  1971 1971 1971 1971 1971 1971 1971 1971 1971 1971 ...
##  $ ceremony     : int  43 43 43 43 43 43 43 43 43 43 ...
##  $ category     : chr  "Best Picture" "Best Actress" "Best Director" "Best Actor" ...
##  $ name         : chr  "Ross Hunter, Producer" "Carrie Snodgress" "Federico Fellini" "Jack Nicholson"
##  $ film         : chr  "Airport" "Diary of a Mad Housewife" "Fellini Satyricon" "Five Easy Pieces" .
##  $ winner       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_noms   : int  10 1 1 4 4 3 7 7 7 7 ...
##  $ total_wins   : int  1 0 0 0 0 0 1 1 1 1 ...
##  $ ids          : chr  "tt0065377" "tt0065636" "tt0064940" "tt0065724" ...
##  $ runtime      : int  137 95 129 98 98 92 100 100 100 100 ...
##  $ genres       : chr  "Action,Drama,Thriller" "Comedy,Drama" "Drama,Fantasy" "Drama" ...
##  $ box_office   : int  371898 6100000 NA NA NA NA 136400000 136400000 136400000 136400000 ...
##  $ budget       : int  10000000 1200000 3000000 1600000 1600000 NA 2200000 2200000 2200000 2200000 .
```

```
skim(final_od)
```

Table 1: Data summary

Name	final_od
Number of rows	1166
Number of columns	14
Column type frequency:	
character	5
numeric	9
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
category	0	1	10	13	0	4	0
name	0	1	4	97	0	792	0
film	0	1	2	54	0	659	0
ids	0	1	9	10	0	660	0
genres	0	1	5	26	0	118	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
year	0	1.00	1998.11	16.12	1970	1984	1999	2012	2024	
year_ceremony	0	1.00	1999.11	16.12	1971	1985	2000	2013	2025	
ceremony	0	1.00	71.11	16.12	43	57	72	85	97	
winner	0	1.00	0.19	0.39	0	0	0	0	1	
total_noms	0	1.00	5.94	3.29	1	3	6	8	15	
total_wins	0	1.00	1.63	2.04	0	0	1	2	11	
runtime	0	1.00	127.69	23.80	84	111	123	138	215	
box_office	114	0.90	132912812.02	281837062.92	67322	2600000	60738797	174600318	1488732821	
budget	85	0.93	27092899.44	35650436.43	312000	7000000	15000000	30000000	350000000	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist

8. Separate genres into 3 different variables.

```
final_od <- final_od %>%
  separate(genres, into = c("genre1", "genre2", "genre3"), sep = ",", fill = "right")
```

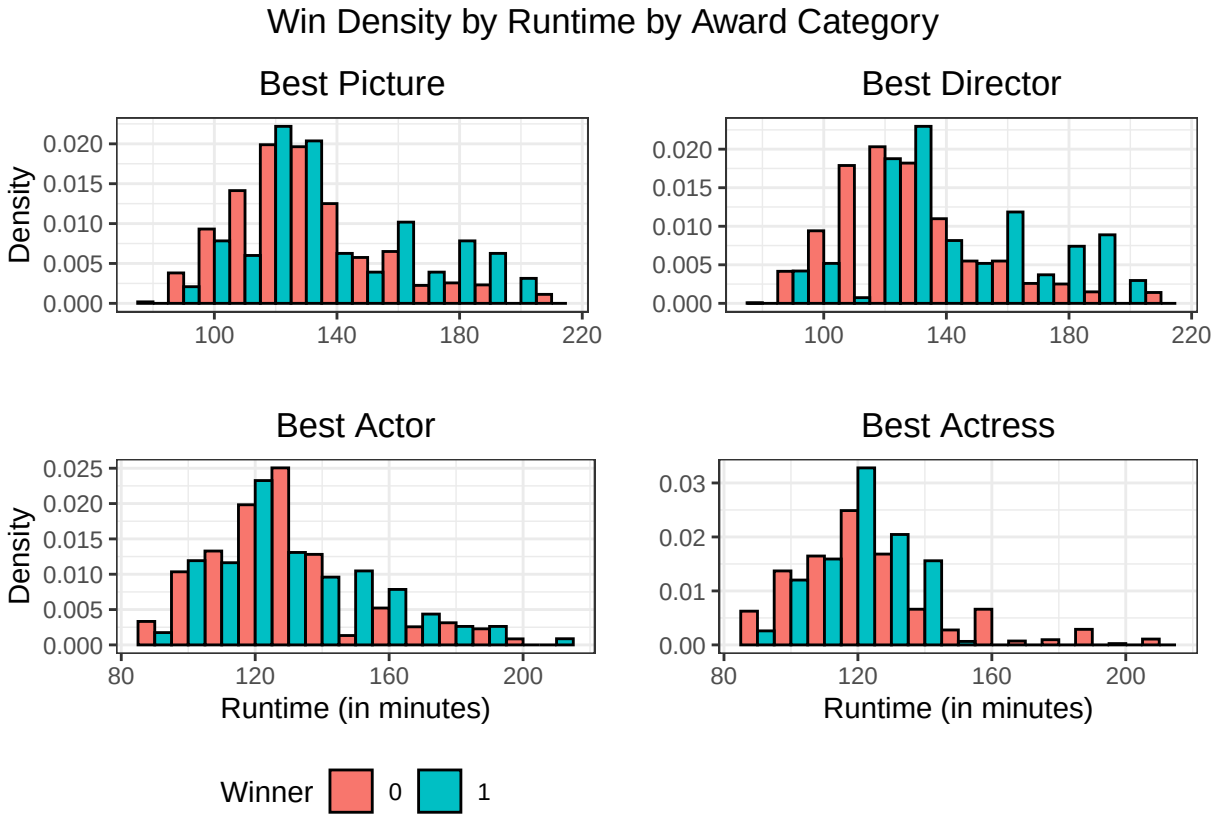
9. Create dummies for award categories and genres by creating new dataframes, merge them, and then drop repeating categories that appear after the merges.

```
final_od <- final_od %>%
  mutate(best_pic_win = ifelse(category == "Best Picture", winner, 0)) %>%
  mutate(best_actor_win = ifelse(category == "Best Actor", winner, 0)) %>%
  mutate(best_actress_win = ifelse(category == "Best Actress", winner, 0)) %>%
  mutate(best_dir_win = ifelse(category == "Best Director", winner, 0))

genre_dummies <- final_od %>%
  pivot_longer(cols = c(genre1, genre2, genre3), names_to = "genre_col", values_to = "genre", values_dr
  mutate(value = 1) %>%
  pivot_wider(names_from = genre, values_from = value, values_fill = 0)

od <- final_od %>%
  left_join(genre_dummies, by = "ids", relationship = "many-to-many") %>% select(-genre_col, -ends_with
  rename_with(~ gsub("\\.x$", "", .), ends_with(".x")) %>%
  rename(sci_fi = "Sci-Fi")
```

10. Histogram comparing the density distribution of runtime of nominees and winners by category.

```
runtime_pic_hist <- ggplot(od[od$category == "Best Picture", ],
       aes(x = runtime,
           fill = as.factor(best_pic_win))) +
  geom_histogram(binwidth = 10, color = "black",
                 aes(y = ..density..),
                 position = "dodge") +
  labs(x = "",
       y = "Density",
       title = "Best Picture", fill = "Winner") +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5),
      legend.position = "none")
```

```
runtime_dir_hist <- ggplot(od[od$category == "Best Director", ],
       aes(x = runtime,
           fill = as.factor(best_dir_win))) +
  geom_histogram(binwidth = 10, color = "black",
                 aes(y = ..density..),
                 position = "dodge") +
  labs(x = "",
       y = "",
       title = "Best Director", fill = "Winner") +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5),
      legend.position = "none")

runtime_actor_hist <- ggplot(od[od$category == "Best Actor", ],
       aes(x = runtime,
           fill = as.factor(best_actor_win))) +
  geom_histogram(binwidth = 10, color = "black",
                 aes(y = ..density..),
                 position = "dodge") +
  labs(x = "Runtime (in minutes)",
       y = "Density",
       title = "Best Actor", fill = "Winner") +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5),
      legend.position = "bottom")

runtime_actress_hist <- ggplot(od[od$category == "Best Actress", ],
       aes(x = runtime,
           fill = as.factor(best_actress_win))) +
  geom_histogram(binwidth = 10, color = "black",
                 aes(y = ..density..),
                 position = "dodge") +
  labs(x = "Runtime (in minutes)",
       y = "",
       title = "Best Actress", fill = "Winner") +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5),
      legend.position = "none")

(runtime_pic_hist | runtime_dir_hist)/
(runtime_actor_hist | runtime_actress_hist) +
  plot_annotation(title = "Win Density by Runtime by Award Category",
                  theme = theme(plot.title = element_text(hjust = .5)))
```

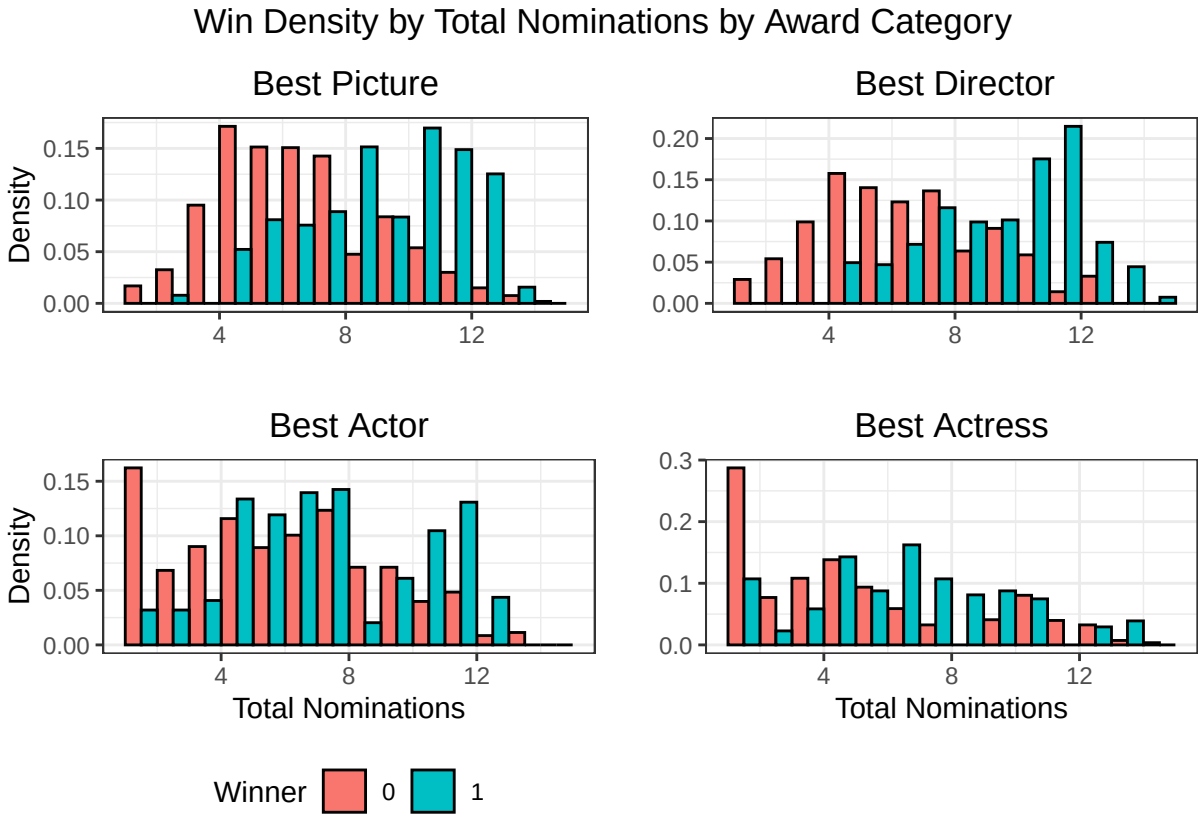


Looking at the histograms, the distribution of winners roughly mirrors that of the nominees across categories. All categories are right-skewed with a center around 120–150 minutes. In the Picture and Director categories, a noticeable spike occurs beyond the 190–minute and 3–hour marks, potentially reflecting the additional time directors have to showcase creative and technical skills. The Best Actor category exhibits a long tail for both winners and nominees, suggesting that extended runtimes may provide actors with more opportunities to demonstrate range and depth. In contrast, Best Actress stands out as an outlier, with a sparse tail beyond 160 minutes and no winners after this point. This may reflect broader industry trends, where films led by female actors tend to have shorter runtimes overall rather than a lack of nominations for longer features.

11. Histogram of density distribution of total_noms by win.

```
total_noms_pic_hist <- ggplot(od[od$category == "Best Picture", ],
        aes(x = total_noms,
            fill = as.factor(best_pic_win))) +
  geom_histogram(bins = length(unique(od$total_noms)),
                 color = "black",
                 breaks = 1:15,
                 aes(y = ..density..),
                 position = "dodge") +
  theme_bw() +
  theme(legend.position = "none",
plot.title = element_text(hjust = .5)) +
  labs(x = "",
       y = "Density",
```

```
       title = "Best Picture")

total_noms_dir_hist <- ggplot(od[od$category == "Best Director", ],
        aes(x = total_noms,
            fill = as.factor(best_dir_win))) +
  geom_histogram(bins = length(unique(od$total_noms)),
                 color = "black",
                 breaks = 1:15,
                 aes(y = ..density..),
                 position = "dodge") +
  theme_bw() +
  theme(legend.position = "none",
plot.title = element_text(hjust = .5)) +
  labs(x = "",
       y = "",
       title = "Best Director")

total_noms_actor_hist <- ggplot(od[od$category == "Best Actor", ],
        aes(x = total_noms,
            fill = as.factor(best_actor_win))) +
  geom_histogram(bins = length(unique(od$total_noms)),
                 color = "black",
                 breaks = 1:15,
                 aes(y = ..density..),
                 position = "dodge") +
  theme_bw() +
  theme(legend.position = "bottom",
plot.title = element_text(hjust = .5)) +
  labs(x = "Total Nominations",
       y = "Density",
       title = "Best Actor",
       fill = "Winner")

total_noms_actress_hist <- ggplot(od[od$category == "Best Actress", ],
        aes(x = total_noms,
            fill = as.factor(best_actress_win))) +
  geom_histogram(bins = length(unique(od$total_noms)),
                 color = "black",
                 breaks = 1:15,
                 aes(y = ..density..),
                 position = "dodge") +
  theme_bw() +
  theme(legend.position = "none",
plot.title = element_text(hjust = .5)) +
  labs(x = "Total Nominations",
       y = "",
       title = "Best Actress")

(total_noms_pic_hist | total_noms_dir_hist) /
  (total_noms_actor_hist | total_noms_actress_hist) +
  plot_annotation(title = "Win Density by Total Nominations by Award Category",
                  theme = theme(plot.title = element_text(hjust = .5)))
```

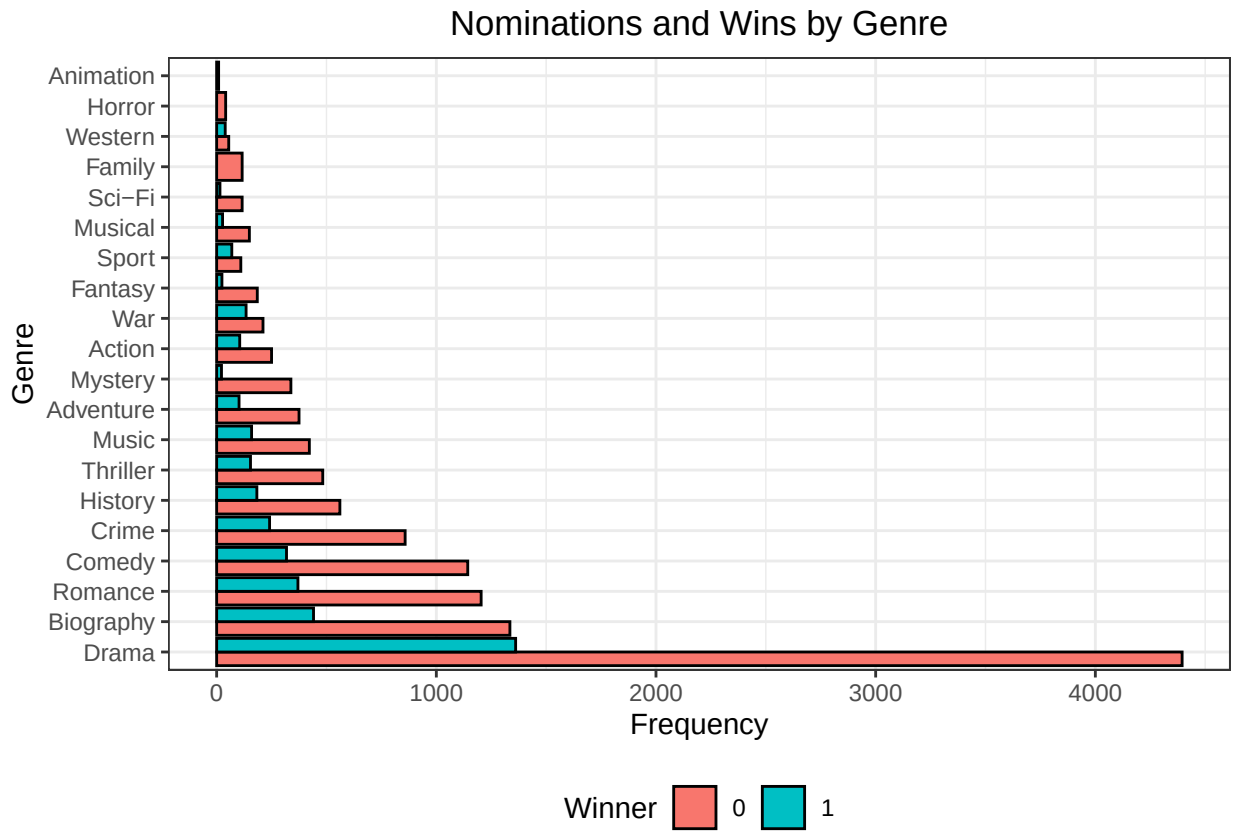


The histograms of total nominations versus wins indicate that films with fewer than four nominations are unlikely to secure awards in most categories. In the Picture and Director categories, additional nominations generally correlate with higher chances of winning, likely reflecting strong performance across multiple dimensions of filmmaking. In the Best Actor category, films with 4–8 nominations show relatively stable odds of producing a winner, with a notable increase only at the 11-nomination mark. By contrast, the Best Actress category demonstrates a different pattern, as actresses appear almost equally likely to win regardless of the total number of nominations a film receives.

12. Create a smaller genre dataframe to create a bar chart of genres and frequency of win.

```
genre_table1 <- od %>%
  select(genre1, winner, year) %>%
  rename(genre = genre1)

genre_table2 <- od %>%
  select(genre2, winner, year) %>%
  rename(genre = genre2)

genre_table3 <- od %>%
  select(genre3, winner, year) %>%
  rename(genre = genre3)
```

```r
genre_master_table <- bind_rows(genre_table1, genre_table2, genre_table3)

genre_master_table <- genre_master_table %>%
  filter(!is.na(genre))

genre_master_table <- genre_master_table %>%
  mutate(genre = fct_infreq(genre))

ggplot(genre_master_table,
       aes(x = genre,
           fill = factor(winner))) +
  geom_bar(position = "dodge",
           color = "black") +
  labs(x = "Genre",
       y = "Frequency",
       title = "Nominations and Wins by Genre",
       fill = "Winner") +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5),
        legend.position = "bottom") +
  coord_flip()
```

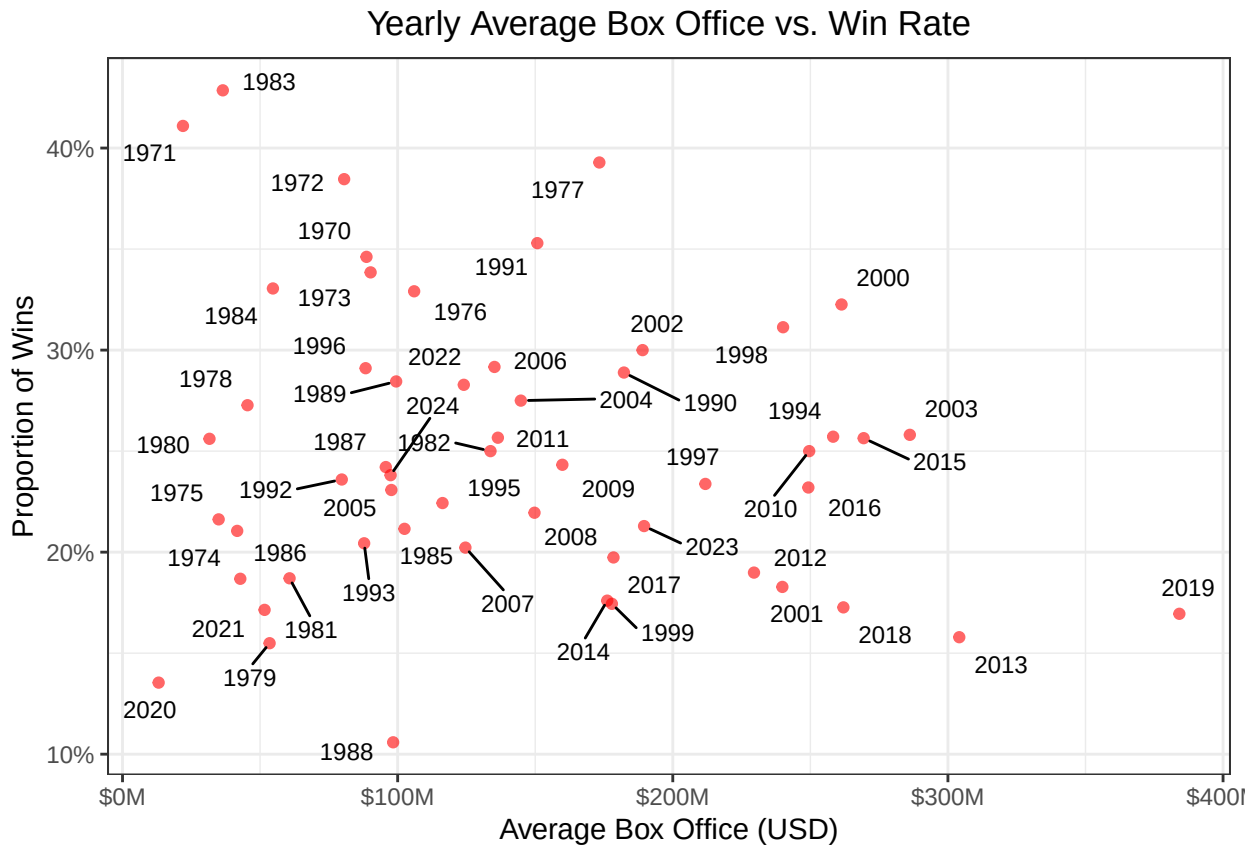


It appears that there is no genre more likely to win than the others purely because of the genre as the proportion of wins looks identical to nominations for most genres. There are some genres with no wins like family, horror, and animation, however it may just be due to incorrect applications of the genres. Most animation is not really a genre, it is a medium of filmmaking so any animated films that won may not have

animation as a genre. Also, there is a category for Best Animated Film which was left out of this analysis, so by including that results may change. Horror movies have been traditionally overlooked by the Academy over the years so that is not a surprise. As for the family genre, it makes sense as family movies are usually wide appealing movies that also have to be inclusive and interesting for children so they may not do well critically being voted on by adults.

13. Create a scatterplot for average yearly box_office vs proportion of wins.

```
 year_box_summary <- od %>%
  filter(!is.na(box_office), !is.na(winner), !is.na(year)) %>%
group_by(year) %>%
  summarise(avg_box_office = mean(box_office),
            win_rate = mean(winner),
            count = n(),
            .groups = "drop")


ggplot(year_box_summary,
       aes(x = avg_box_office,
           y = win_rate)) +
  geom_point(color = "red",
             alpha = 0.6) +
  ggrepel::geom_text_repel(
    aes(label = year),
    size = 3,
    color = "black",
    max.overlaps = 20,
    box.padding = 0.4,
    point.padding = 0.2,
    force = 2) +
  scale_y_continuous(labels = percent_format(accuracy = 1)) +
  scale_x_continuous(labels = dollar_format(scale = 1e-6, suffix = "M")) +
  labs(
    x = "Average Box Office (USD)",
    y = "Proportion of Wins",
    title = "Yearly Average Box Office vs. Win Rate") +
  theme_bw() + theme(legend.position = "bottom",
                     plot.title = element_text(hjust = .5))
```

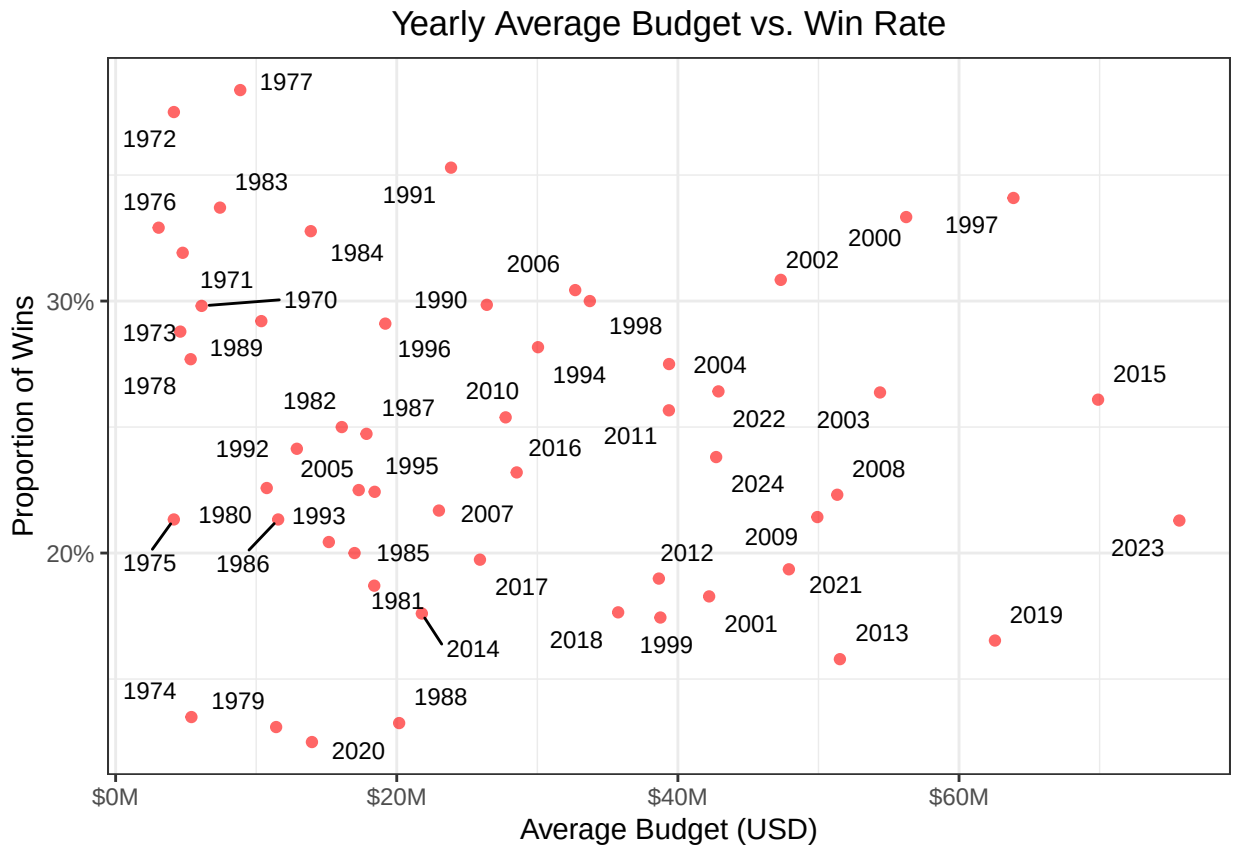


An analysis of yearly average box office returns versus the proportion of nominated films that win reveals no consistent relationship between higher spending and award success. Most data points cluster around $90–150 million in average box office and a 20–30% win rate. Variation across years is partly explained by changes in the number of Best Picture nominees, as the Academy has adjusted the slate to test for viewership impact. For example, in 2019 the larger nominee pool included more blockbusters with higher box office returns, which raised the overall average but reduced the win rate due to increased competition.

14. Create a scatterplot for average yearly budget vs proportion of wins.

```
year_bud_summary <- od %>%
  filter(!is.na(budget),
         !is.na(winner),
         !is.na(year)) %>%
  group_by(year) %>%
  summarise(
    avg_budget = mean(budget),
    win_rate = mean(winner),
    count = n(),
    .groups = "drop")

ggplot(year_bud_summary,
       aes(x = avg_budget,
```

```
          y = win_rate)) +
geom_point(#
           color = "red",
           alpha = 0.6) +
ggrepel::geom_text_repel(
  aes(label = year),
  size = 3,
  color = "black",
  max.overlaps = 20,
  box.padding = 0.4,
  point.padding = 0.2,
  force = 2) +
scale_y_continuous(labels = percent_format(accuracy = 1)) +
scale_x_continuous(labels = dollar_format(scale = 1e-6, suffix = "M")) +
labs(
  x = "Average Budget (USD)",
  y = "Proportion of Wins",
  title = "Yearly Average Budget vs. Win Rate",
  size = "Movies that Year") +
theme_bw() +
theme(legend.position = "bottom",
      plot.title = element_text(hjust = .5))
```



Shifting to looking at average yearly budget vs win proportions there is a general trend of increasing movie budgets between decades. This can be accounted for based on the fact that now more than ever films are seen as investments and less so art. Studios and distributors want to see bigger returns on movies and make

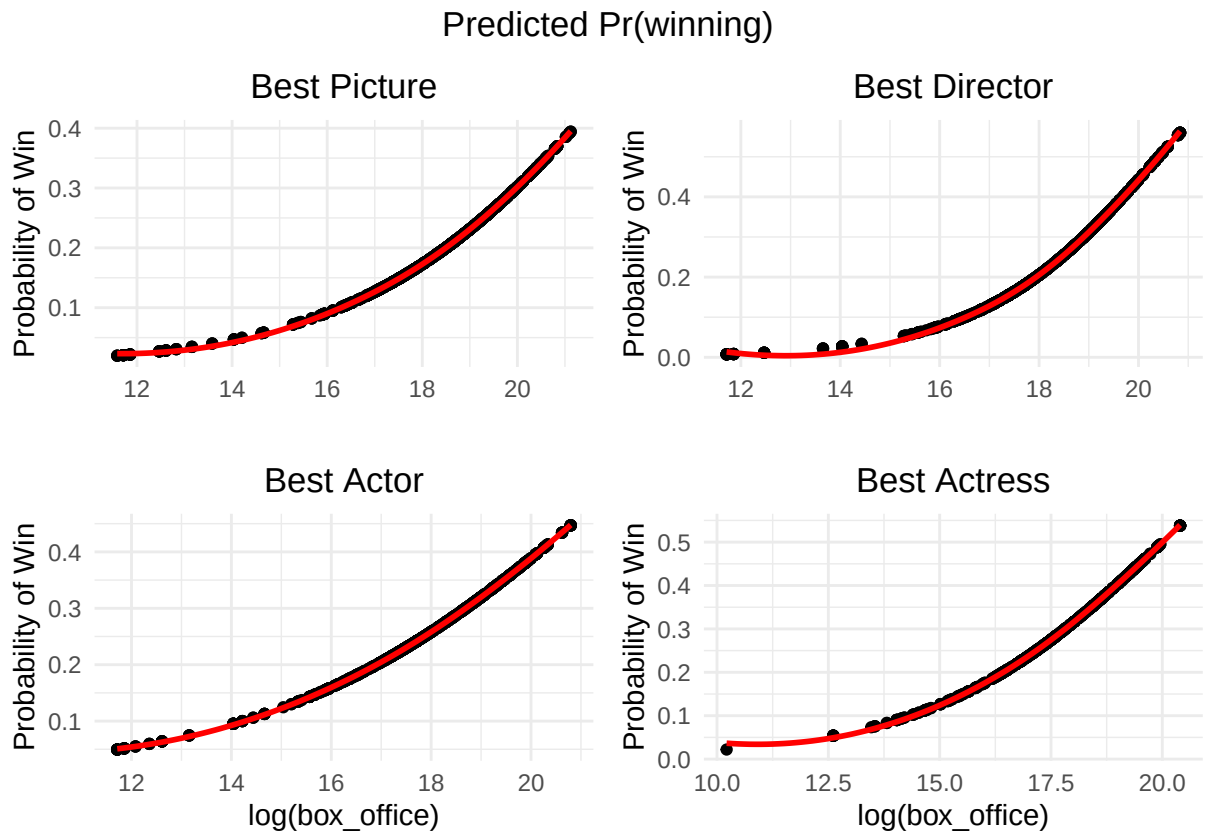
more money and to do that you may have to spend more money to get better actors, directors, set designers, and production crew members to get the attractiveness that is required to draw in the most money possible.

15. Find the predicted win probabilities for box_office for each of the four award categories and create scatterplots comparing them.

```
box_office_df <- od %>%
  filter(!is.na(box_office))

box_pic_od <- box_office_df %>%
  filter(category == "Best Picture")

box_pic_model <- glm(best_pic_win ~ log(box_office),
                     data = box_pic_od,
                     family = "binomial")

box_pic_od <- box_pic_od %>%
  mutate(
    pred_box_pic = predict(box_pic_model, type = "response")
  )

box_v_pic <- ggplot(box_pic_od,
                    aes(
                      x = log(box_office),
                      y = pred_box_pic
                    )) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(
    x = "",
    y = "Probability of Win",
    title = "Best Picture") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))

box_dir_od <- box_office_df %>%
  filter(category == "Best Director")

box_dir_model <- glm(best_dir_win ~ log(box_office),
                     data = box_dir_od,
                     family = "binomial")

box_dir_od <- box_dir_od %>%
  mutate(
    pred_box_dir = predict(box_dir_model, type = "response")
  )

box_v_dir <- ggplot(box_dir_od,
                    aes(
                      x = log(box_office),
                      y = pred_box_dir
```

```
                    )) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(
    x = "",
    y = "Probability of Win",
    title = "Best Director") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))

box_actor_od <- box_office_df %>%
  filter(category == "Best Actor")

box_actor_model <- glm(best_actor_win ~ log(box_office),
                       data = box_actor_od,
                       family = "binomial")

box_actor_od <- box_actor_od %>%
  mutate(
    pred_box_actor = predict(box_actor_model, type = "response")
  )

box_v_actor <- ggplot(box_actor_od,
                      aes(
                        x = log(box_office),
                        y = pred_box_actor
                      )) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(
    x = "log(box_office)",
    y = "Probability of Win",
    title = "Best Actor") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))

box_actress_od <- box_office_df %>%
  filter(category == "Best Actress")

box_actress_model <- glm(best_actress_win ~ log(box_office),
                       data = box_actress_od,
                       family = "binomial")

box_actress_od <- box_actress_od %>%
  mutate(
    pred_box_actress = predict(box_actress_model, type = "response")
  )

box_v_actress <- ggplot(box_actress_od,
                      aes(
                        x = log(box_office),
                        y = pred_box_actress
                      )) +
```

```
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(
    x = "log(box_office)",
    y = "Probability of Win",
    title = "Best Actress") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))

(box_v_pic | box_v_dir) /
(box_v_actor | box_v_actress) +
  plot_annotation(title = "Predicted Pr(winning)",
                  theme = theme(plot.title = element_text(hjust = .5)))
```



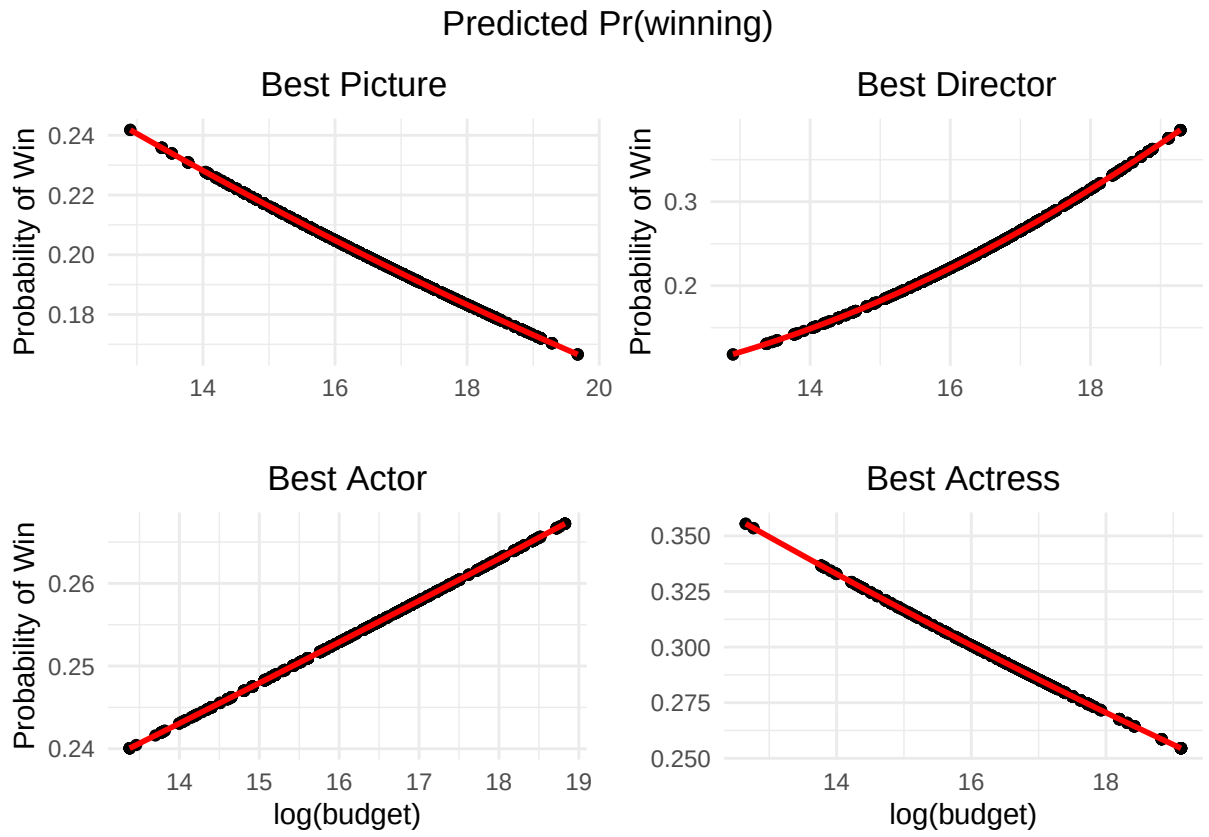
Using box office as the sole predictor in a model estimating the probability of winning awards shows a positive relationship between box office performance and all award categories. There are nuances to this outcome. This trend is understandable, as a movie attracting significant public attention is more likely to possess qualities that make it a strong candidate for awards. However, this relationship is not always straightforward, since blockbuster movies often perform best at the box office but frequently win few awards, given that they are generally not considered as artistically distinguished as other films.

16. Do the same for budget.

```
budget_df <- od %>%
  filter(!is.na(budget))

bud_pic_od <- budget_df %>%
  filter(category == "Best Picture")

bud_pic_model <- glm(best_pic_win ~ log(budget),
                     data = bud_pic_od,
                     family = "binomial")

bud_pic_od <- bud_pic_od %>%
  mutate(
    pred_bud_pic = predict(bud_pic_model, type = "response")
  )

bud_v_pic <- ggplot(bud_pic_od,
                    aes(
                      x = log(budget),
                      y = pred_bud_pic
                    )) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(
    x = "",
    y = "Probability of Win",
    title = "Best Picture") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))

bud_dir_od <- budget_df %>%
  filter(category == "Best Director")

bud_dir_model <- glm(best_dir_win ~ log(budget),
                     data = bud_dir_od,
                     family = "binomial")

bud_dir_od <- bud_dir_od %>%
  mutate(
    pred_bud_dir = predict(bud_dir_model, type = "response")
  )

bud_v_dir <- ggplot(bud_dir_od,
                    aes(
                      x = log(budget),
                      y = pred_bud_dir
                    )) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(
    x = "",
    y = "Probability of Win",
```

```
    title = "Best Director") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))

bud_actor_od <- budget_df %>%
  filter(category == "Best Actor")

bud_actor_model <- glm(best_actor_win ~ log(budget),
                       data = bud_actor_od,
                       family = "binomial")

bud_actor_od <- bud_actor_od %>%
  mutate(
    pred_bud_actor = predict(bud_actor_model, type = "response")
  )

bud_v_actor <- ggplot(bud_actor_od,
                      aes(
                        x = log(budget),
                        y = pred_bud_actor
                      )) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(
    x = "log(budget)",
    y = "Probability of Win",
    title = "Best Actor") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))

bud_actress_od <- budget_df %>%
  filter(category == "Best Actress")

bud_actress_model <- glm(best_actress_win ~ log(budget),
                       data = bud_actress_od,
                       family = "binomial")

bud_actress_od <- bud_actress_od %>%
  mutate(
    pred_bud_actress = predict(bud_actress_model, type = "response")
  )

bud_v_actress <- ggplot(bud_actress_od,
                      aes(
                        x = log(budget),
                        y = pred_bud_actress
                      )) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(
    x = "log(budget)",
    y = "",
    title = "Best Actress") +
```

```
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))

(bud_v_pic | bud_v_dir) /
(bud_v_actor | bud_v_actress) +
  plot_annotation(title = "Predicted Pr(winning)",
                  theme = theme(plot.title = element_text(hjust = .5)))
```



For the model including only budget, some interesting patterns emerge. First, Best Picture and Best Actress are less likely to win as the budget increases. This might be expected for all categories, since higher-budget films are more likely to be blockbusters, which often receive less critical acclaim. However, Best Director and Best Actor are more likely to win as the budget rises. A higher budget may allow the director greater creative freedom in shaping the film. For Best Actor, a larger budget could enable the hiring of more accomplished actors who are more likely to win awards.

17. Create our logistic regression models since our outcome variable is a probability between 0 and 1.

```
monetary_influence_pic <- glm(best_pic_win ~ log(box_office) + log(budget),
                              data = od[od$category == "Best Picture", ], family = "binomial")

monetary_influence_dir <- glm(best_dir_win ~ log(box_office) + log(budget),
```

```
                              data = od[od$category == "Best Director", ], family = "binomial")

monetary_influence_actor <- glm(best_actor_win ~ log(box_office) + log(budget),
                                data = od[od$category == "Best Actor", ], family = "binomial")

monetary_influence_actress <- glm(best_actress_win ~ log(box_office) + log(budget),
                                  data = od[od$category == "Best Actress", ], family = "binomial")

film_char_pic <- glm(best_pic_win ~ runtime + total_noms + Action + Drama + Thriller +
                     Comedy + Fantasy + Music + Romance + War + Biography + Sport +
                     Crime + Family + Musical + Comedy + History,
                     data = od[od$category == "Best Picture", ],
                     family = "binomial")

film_char_dir <- glm(best_dir_win ~ runtime + total_noms + Action + Drama + Thriller +
                     Comedy + Fantasy + Music + Romance + War + Biography + Sport +
                     Crime + Family + Musical + Comedy + History,
                     data = od[od$category == "Best Director", ],
                     family = "binomial")

film_char_actor <- glm(best_actor_win ~ runtime + total_noms + Action + Drama + Thriller +
                     Comedy + Fantasy + Music + Romance + War + Biography + Sport +
                     Crime + Family + Musical + Comedy + Mystery + History,
                     data = od[od$category == "Best Actor", ],
                     family = "binomial")

film_char_actress <- glm(best_actress_win ~ runtime + total_noms + Action + Drama + Thriller +
                     Comedy + Fantasy + Music + Romance + War + Biography + Sport +
                     Crime + Family + Musical + Comedy + Mystery + History,
                     data = od[od$category == "Best Actress", ],
                     family = "binomial")




full_model_pic <- glm(best_pic_win ~ runtime + log(box_office) + log(budget) + total_noms +
                            Action + Drama + Thriller + Comedy + Fantasy + Music +
                            Romance + War + Biography + Sport + Crime + Family +
                            Musical + Comedy + Mystery + History,
                        data = od[od$category == "Best Picture", ], family = "binomial")

full_model_dir <- glm(best_dir_win ~ runtime + log(box_office) + log(budget) + total_noms +
                            Action + Drama + Thriller + Comedy + Fantasy + Music +
                            Romance + War + Biography + Sport + Crime + Family +
                            Musical + Comedy + Mystery + History,
                        data = od[od$category == "Best Director", ], family = "binomial")

full_model_actor <- glm(best_actor_win ~ runtime + log(box_office) + log(budget) + total_noms +
                            Action + Drama + Thriller + Comedy + Fantasy + Music +
                            Romance + War + Biography + Sport + Crime + Family +
                            Musical + Comedy + Mystery + History,
                          data = od[od$category == "Best Actor", ], family = "binomial")
```

```
full_model_actress <- glm(best_actress_win ~ runtime + log(box_office) + log(budget) + total_noms +
                          Action + Drama + Thriller + Comedy + Fantasy + Music +
                          Romance + War + Biography + Sport + Crime + Family +
                          Musical + Comedy + Mystery + History,
                          data = od[od$category == "Best Actress", ], family = "binomial")

mon_influence_table <- modelsummary(
  list("Best Picture" = monetary_influence_pic,
       "Best Director" = monetary_influence_dir,
       "Best Actor" = monetary_influence_actor,
       "Best Actress" = monetary_influence_actress),
  stars = c("*" = .1, "**" = .05, "***" = .01),,
  statistic = "p.value",
  fmt = 5,
  gof_omit = "BIC|Log.Lik",
  title = "Monetary Influence Only")

film_char_table <- modelsummary(
  list("Best Picture" = film_char_pic,
       "Best Director" = film_char_dir,
       "Best Actor" = film_char_actor,
       "Best Actress" = film_char_actress),
  stars = c("*" = .1, "**" = .05, "***" = .01),
  statistic = "p.value",
  fmt = 5,
  gof_omit = "BIC|Log.Lik",
  title = "Filmmaking Characteristics Only")

full_model_table <- modelsummary(
  list("Best Picture" = full_model_pic,
       "Best Director" = full_model_dir,
       "Best Actor" = full_model_actor,
       "Best Actress" = full_model_actress),
  stars = c("*" = .1, "**" = .05, "***" = .01),
  statistic = "p.value",
  fmt = 5,
  gof_omit = "BIC|Log.Lik",
  title = "Full Model")

mon_influence_table
```

```
film_char_table
```

```
full_model_table
```

For the monetary influences logistic regression, some interesting patterns emerge. When box office and budget are combined in the same regression, the results differ from the analyses of each variable individually in the earlier logit models. Previously, some budget coefficients were positive, but in the combined regression, budget shows a negative effect across all award categories, while box office remains positive.

In the filmmaking characteristics logistic regression model, several variables stand out. Total nominations are a significant predictor for all four award categories, with greater significance for Best Picture and Best

Table 4: Monetary Influence Only

	Best Picture	Best Director	Best Actor	Best Actress
(Intercept)	−5.321 08***	−10.119 45***	−4.294 82***	−4.127 45***
	($<1 \times 10^{-5}$)	($<1 \times 10^{-5}$)	(1×10^{-4})	(0.001 85)
log(box_office)	0.652 99***	0.647 74***	0.425 93***	0.661 24***
	($<1 \times 10^{-5}$)	($<1 \times 10^{-5}$)	($<1 \times 10^{-5}$)	($<1 \times 10^{-5}$)
log(budget)	−0.488 53***	−0.175 32***	−0.268 75***	−0.522 36***
	($<1 \times 10^{-5}$)	(0.006 73)	(9×10^{-5})	($<1 \times 10^{-5}$)
Num.Obs.	1888	1564	1262	938
AIC	1774.7	1640.7	1411.7	1087.6
F	46.059	48.322	22.968	31.195
RMSE	0.39	0.42	0.43	0.44

* p <0.1, ** p <0.05, *** p <0.01

Director than for Best Actor and Best Actress. Runtime is also a significant predictor for all categories except Best Picture, which is notable given that Best Director is often judged similarly and would typically show comparable variable relevance. Various genre variables show significance for some award categories but not others. For instance, the drama, thriller, romance, war, sport, and comedy genres are significant for Best Picture, Best Director, and Best Actress, but not for Best Actor.

Finally, in the model combining monetary influences with filmmaking characteristics, some changes are observed. Runtime becomes significant for Best Picture. Beyond that, coefficients for all variables in the model are adjusted, and the four main predictors—box office, budget, total nominations, and runtime—are all statistically significant at the 1% level, except for runtime, which is significant at the 5% level for the Best Actor category.

Table 5: Filmmaking Characteristics Only

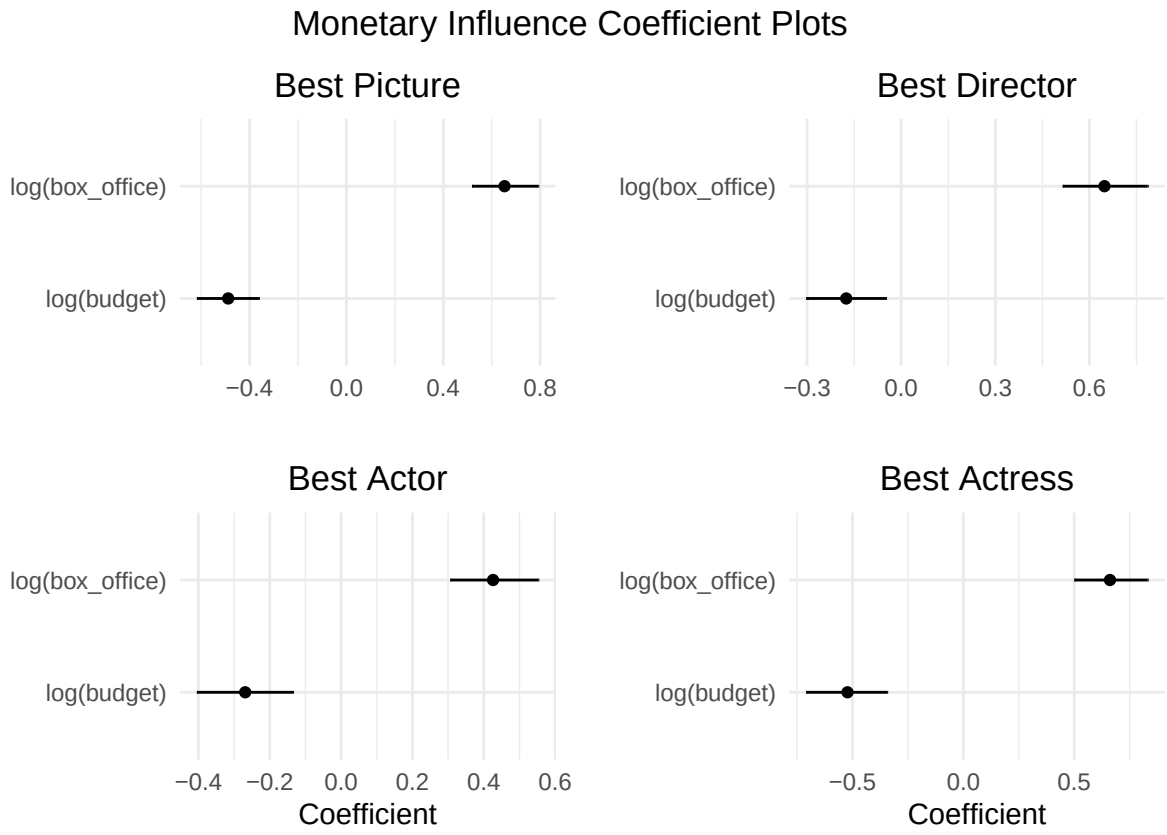
	Best Picture	Best Director	Best Actor	Best Actress
(Intercept)	−6.221 08***	−6.920 77***	−1.903 23***	−0.229 46
	(<1 × 10^{-5})	(<1 × 10^{-5})	(0.001 25)	(0.726 04)
runtime	0.002 54	0.009 39***	−0.009 25***	−0.022 63***
	(0.335 34)	(0.000 54)	(0.004 91)	(<1 × 10^{-5})
total_noms	0.413 32***	0.448 39***	0.194 01***	0.186 47***
	(<1 × 10^{-5})	(<1 × 10^{-5})	(<1 × 10^{-5})	(<1 × 10^{-5})
Action	0.579 39	0.167 35	2.097 80***	2.161 93**
	(0.200 44)	(0.717 58)	(0.001 90)	(0.041 19)
Drama	1.069 53***	0.801 07***	0.725 20	0.834 36*
	(0.000 25)	(0.003 23)	(0.101 17)	(0.084 93)
Thriller	1.490 62***	1.439 34***	−0.016 25	1.313 22**
	(0.000 31)	(0.000 34)	(0.978 45)	(0.025 10)
Comedy	1.280 23***	0.585 28*	−0.364 43	1.151 00**
	(0.000 26)	(0.098 36)	(0.490 02)	(0.024 86)
Fantasy	−0.339 34	−0.173 65	−15.205 67	−14.129 95
	(0.633 97)	(0.783 79)	(0.984 20)	(0.973 91)
Music	−0.254 02	0.995 77**	1.034 20**	0.980 35*
	(0.656 28)	(0.027 84)	(0.047 14)	(0.086 12)
Romance	0.835 54**	0.721 38**	0.334 41	0.998 06**
	(0.017 50)	(0.035 84)	(0.493 29)	(0.046 84)
War	1.562 33***	1.706 84***	0.799 84	1.414 63*
	(0.000 43)	(0.000 07)	(0.179 54)	(0.082 29)
Biography	1.108 65***	0.665 72**	0.904 27*	1.060 09*
	(0.001 08)	(0.044 37)	(0.050 34)	(0.050 25)
Sport	2.827 48***	2.438 29***	0.909 14	1.715 86**
	(<1 × 10^{-5})	(0.000 06)	(0.168 02)	(0.021 94)
Crime	1.328 54***	0.245 38	0.667 27	0.770 45
	(0.000 25)	(0.504 21)	(0.183 79)	(0.168 80)
Family	−12.993 21	−13.920 89	−14.654 38	−13.549 68
	(0.967 11)	(0.975 96)	(0.986 13)	(0.980 25)
Musical	−0.233 02	−0.711 80	−14.828 14	−0.146 00
	(0.743 84)	(0.321 33)	(0.984 97)	(0.860 68)
History	1.049 49***	0.212 81	0.881 24*	−0.217 39
	(0.007 82)	(0.591 98)	(0.097 07)	(0.782 44)
Mystery			0.589 70	−1.023 60
			(0.337 71)	(0.369 12)
Num.Obs.	1982	1680	1398	1140
AIC	1608.6	1437.4	1475.9	1241.1
F	17.644	19.237	5.321	5.630
RMSE	0.35	0.37	0.41	0.42

* p <0.1, ** p <0.05, *** p <0.01

Table 6: Full Model

	Best Picture	Best Director	Best Actor	Best Actress
(Intercept)	−3.505 76***	−7.161 48***	−3.218 04**	−1.962 85
	(0.009 81)	(<1 × 10^{-5})	(0.015 72)	(0.216 80)
runtime	0.015 44***	0.019 90***	−0.007 40**	−0.020 44***
	(<1 × 10^{-5})	(<1 × 10^{-5})	(0.043 75)	(0.000 05)
log(box_office)	0.664 74***	0.578 26***	0.343 64***	0.430 88***
	(<1 × 10^{-5})	(<1 × 10^{-5})	(<1 × 10^{-5})	(<1 × 10^{-5})
log(budget)	−1.012 44***	−0.697 50***	−0.287 29***	−0.377 28***
	(<1 × 10^{-5})	(<1 × 10^{-5})	(0.000 28)	(0.000 39)
total_noms	0.442 26***	0.424 35***	0.143 39***	0.160 89***
	(<1 × 10^{-5})	(<1 × 10^{-5})	(<1 × 10^{-5})	(<1 × 10^{-5})
Action	0.623 72	0.207 11	2.248 53***	2.621 04**
	(0.214 41)	(0.685 15)	(0.001 83)	(0.037 38)
Drama	1.020 70***	0.863 84***	0.800 83*	1.089 85**
	(0.002 50)	(0.007 42)	(0.075 17)	(0.029 05)
Thriller	1.389 33***	1.362 54***	−0.103 70	1.395 72**
	(0.001 75)	(0.001 58)	(0.864 85)	(0.021 36)
Comedy	1.343 97***	0.790 19**	−0.303 93	1.207 93**
	(0.000 61)	(0.045 19)	(0.570 57)	(0.022 36)
Fantasy	−0.540 99	−0.104 87	−15.230 52	−14.325 56
	(0.474 76)	(0.875 16)	(0.984 28)	(0.975 89)
Music	−0.275 38	1.469 60***	1.207 41**	2.125 23***
	(0.661 55)	(0.004 32)	(0.023 58)	(0.000 76)
Romance	0.717 41*	0.649 94*	0.307 01	1.168 66**
	(0.070 01)	(0.098 57)	(0.535 58)	(0.024 91)
War	1.417 80***	1.750 44***	0.830 00	1.588 89*
	(0.003 99)	(0.000 26)	(0.172 65)	(0.067 18)
Biography	1.197 02***	0.819 84**	0.972 53**	1.490 58***
	(0.001 92)	(0.030 42)	(0.038 63)	(0.008 52)
Sport	3.192 59***	2.426 28***	0.790 20	1.621 60*
	(<1 × 10^{-5})	(0.000 88)	(0.260 03)	(0.065 60)
Crime	1.194 91***	0.286 66	0.525 68	0.870 18
	(0.003 30)	(0.485 93)	(0.302 90)	(0.132 01)
Family	−14.983 81	−14.534 70	−14.028 78	−13.136 40
	(0.986 19)	(0.983 14)	(0.988 26)	(0.985 57)
Musical	0.064 96	−0.167 98	−14.410 34	0.045 32
	(0.930 32)	(0.823 23)	(0.984 32)	(0.957 08)
Mystery	−14.695 57	−14.029 62	0.252 44	−0.784 52
	(0.980 02)	(0.969 57)	(0.707 10)	(0.495 59)
History	1.077 92**	0.244 54	0.993 46*	0.128 85
	(0.016 46)	(0.580 97)	(0.067 34)	(0.875 35)
Num.Obs.	1888	1764	1262	938
AIC	1408.5	1303.7	1365.6	1045.0
F	16.556	15.766	5.280	5.900

18. Coefficient plots for monetary influence models.



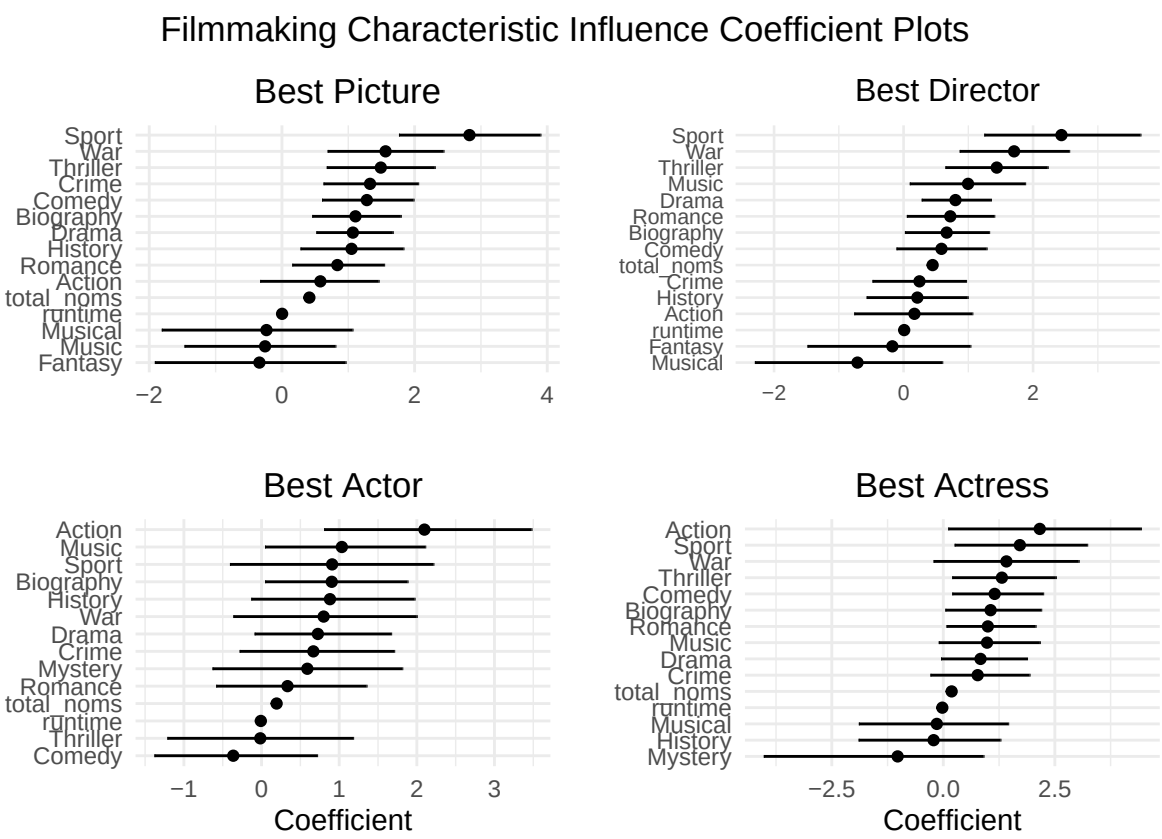
The same patterns observed in the previous regressions are evident here. All coefficients are statistically different from zero. Box office retains the same positive coefficient seen in the logit plots, while budget is now negative across all categories, rather than only some, once box office is included in the model.

19. Coefficient plots for filmmaking characteristic models.

```
film_char_pic_tidy <- tidy(film_char_pic, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  filter(abs(conf.high - conf.low) < 20)
film_char_dir_tidy <- tidy(film_char_dir, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  filter(abs(conf.high - conf.low) < 20)
film_char_actor_tidy <- tidy(film_char_actor, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  filter(abs(conf.high - conf.low) < 20)
film_char_actress_tidy <- tidy(film_char_actress, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  filter(abs(conf.high - conf.low) < 20)

film_char_pic_coef_plot <- ggplot(film_char_pic_tidy,
       aes(
```

```
        x = estimate,
        y = reorder(term, estimate)
      )) +
  geom_point() +
  geom_errorbarh(aes(
    xmin = conf.low,
    xmax = conf.high,
    height = 0
  )) +
  theme_minimal() +
  labs(
    x = "",
    y = "",
    title = "Best Picture") +
  theme(plot.title = element_text(hjust = .5))

film_char_dir_coef_plot <- ggplot(film_char_dir_tidy,
       aes(
         x = estimate,
         y = reorder(term, estimate)
       )) +
  geom_point() +
  geom_errorbarh(aes(
    xmin = conf.low,
    xmax = conf.high,
    height = 0
  )) +
  theme_minimal(base_size = 10) +
  labs(
    x = "",
    y = "",
    title = "Best Director") +
  theme(plot.title = element_text(hjust = .5))

film_char_actor_coef_plot <- ggplot(film_char_actor_tidy,
       aes(
         x = estimate,
         y = reorder(term, estimate)
       )) +
  geom_point() +
  geom_errorbarh(aes(
    xmin = conf.low,
    xmax = conf.high,
    height = 0
  )) +
  theme_minimal() +
  labs(
    x = "Coefficient",
    y = "",
    title = "Best Actor") +
  theme(plot.title = element_text(hjust = .5))

film_char_actress_coef_plot <- ggplot(film_char_actress_tidy,
```

```
      aes(
        x = estimate,
        y = reorder(term, estimate)
      )) +
  geom_point() +
  geom_errorbarh(aes(
    xmin = conf.low,
    xmax = conf.high,
    height = 0
  )) +
  theme_minimal() +
  labs(
    x = "Coefficient",
    y = "",
    title = "Best Actress") +
  theme(plot.title = element_text(hjust = .5))

(film_char_pic_coef_plot | film_char_dir_coef_plot) /
(film_char_actor_coef_plot | film_char_actress_coef_plot) +
  plot_annotation(title = "Filmmaking Characteristic Influence Coefficient Plots",
                  theme = theme(plot.title = element_text(hjust = .5)))
```



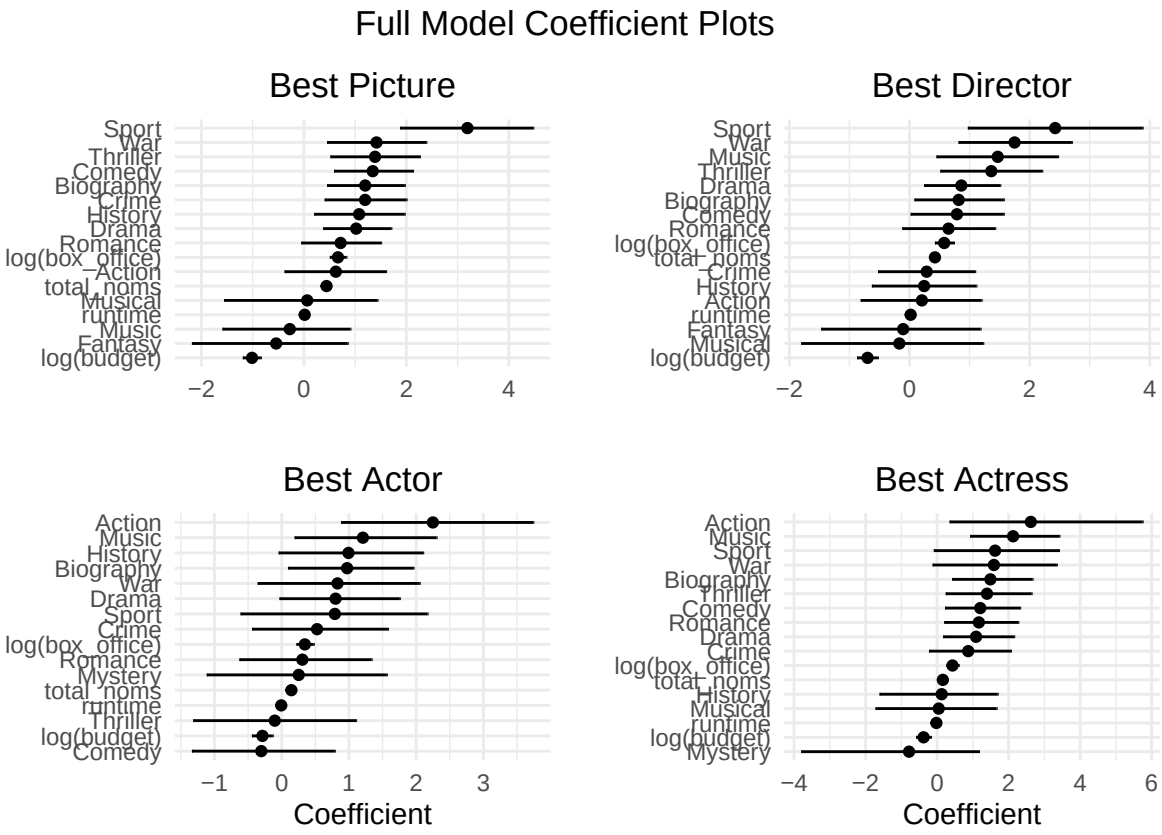
Examining the coefficient plot instead of the regression table reveals that genre has minimal impact for actors and actresses pursuing awards, as most confidence intervals for the coefficients barely extend beyond zero, or do not at all, with the exception of action for actors. Directors, however, show stronger associations between genre and win probabilities, with some genres, such as sport and war, clearly exhibiting coefficients

well outside the range of zero for Best Director and Best Picture.

20. Coefficient plots for full models.

```
full_model_pic_tidy <- tidy(full_model_pic, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  filter(abs(conf.high - conf.low) < 20)
full_model_dir_tidy <- tidy(full_model_dir, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  filter(abs(conf.high - conf.low) < 20)
full_model_actor_tidy <- tidy(full_model_actor, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  filter(abs(conf.high - conf.low) < 20)
full_model_actress_tidy <- tidy(full_model_actress, conf.int = TRUE) %>%
  filter(term != "(Intercept)") %>%
  filter(abs(conf.high - conf.low) < 20)

full_model_pic_coef_plot <- ggplot(full_model_pic_tidy,
       aes(
         x = estimate,
         y = reorder(term, estimate)
       )) +
  geom_point() +
  geom_errorbarh(aes(
    xmin = conf.low,
    xmax = conf.high,
    height = 0
  )) +
  theme_minimal() +
  labs(
    x = "",
    y = "",
    title = "Best Picture") +
  theme(plot.title = element_text(hjust = .5))

full_model_dir_coef_plot <- ggplot(full_model_dir_tidy,
       aes(
         x = estimate,
         y = reorder(term, estimate)
       )) +
  geom_point() +
  geom_errorbarh(aes(
    xmin = conf.low,
    xmax = conf.high,
    height = 0
  )) +
  theme_minimal() +
  labs(
    x = "",
    y = "",
    title = "Best Director") +
```

```
  theme(plot.title = element_text(hjust = .5))

full_model_actor_coef_plot <- ggplot(full_model_actor_tidy,
       aes(
         x = estimate,
         y = reorder(term, estimate)
       )) +
  geom_point() +
  geom_errorbarh(aes(
    xmin = conf.low,
    xmax = conf.high,
    height = 0
  )) +
  theme_minimal() +
  labs(
    x = "Coefficient",
    y = "",
    title = "Best Actor") +
  theme(plot.title = element_text(hjust = .5))

full_model_actress_coef_plot <- ggplot(full_model_actress_tidy,
       aes(
         x = estimate,
         y = reorder(term, estimate)
       )) +
  geom_point() +
  geom_errorbarh(aes(
    xmin = conf.low,
    xmax = conf.high,
    height = 0
  )) +
  theme_minimal() +
  labs(
    x = "Coefficient",
    y = "",
    title = "Best Actress") +
  theme(plot.title = element_text(hjust = .5))

(full_model_pic_coef_plot | full_model_dir_coef_plot) /
(full_model_actor_coef_plot | full_model_actress_coef_plot) +
  plot_annotation(title = "Full Model Coefficient Plots",
                  theme = theme(plot.title = element_text(hjust = .5)))
```



Finally, examining the coefficient plot for the entire model brings together visually all the patterns observed in the logistic regression table. Actors show few statistically significant genre predictors, but the previously identified major predictors remain significant. Actresses exhibit a similar pattern, with slightly more significant genre predictors than actors. For Best Picture and Best Director, the graphs are quite similar, with minor differences among genre predictors, while all four main predictors remain positive and statistically significant.

21. Test our models with prediction accuracy.

```
set.seed(10)

pred_od <- od %>%
  filter(!is.na(budget), !is.na(box_office))

pred_pic_od <- pred_od %>%
  filter(category == "Best Picture")

pic_indices <- sample(1:nrow(pred_pic_od), size = nrow(pred_pic_od) * .8, replace = FALSE)
train_pic <- pred_pic_od[pic_indices,]
test_pic <- pred_pic_od[-pic_indices,]

pred_dir_od <- pred_od %>%
  filter(category == "Best Director")
```

```
dir_indices <- sample(1:nrow(pred_dir_od), size = nrow(pred_dir_od) * .8, replace = FALSE)
train_dir <- pred_dir_od[dir_indices,]
test_dir <- pred_dir_od[-dir_indices,]

pred_actor_od <- pred_od %>%
  filter(category == "Best Actor")

actor_indices <- sample(1:nrow(pred_actor_od), size = nrow(pred_actor_od) * .8, replace = FALSE)
train_actor <- pred_actor_od[actor_indices,]
test_actor <- pred_actor_od[-actor_indices,]

pred_actress_od <- pred_od %>%
  filter(category == "Best Actress")

actress_indices <- sample(1:nrow(pred_actress_od), size = nrow(pred_actress_od) * .8, replace = FALSE)
train_actress <- pred_actress_od[actress_indices,]
test_actress <- pred_actress_od[-actress_indices,]

pred_model_pic <- glm(best_pic_win ~ runtime + box_office + budget + total_noms +
                        Action + Drama + Thriller + Comedy + Fantasy + Music +
                        Romance + War + Biography + Sport + Crime + Family +
                        Musical + Comedy + Mystery + History + Horror + sci_fi, data = train_pic)

pred_pic <- predict(pred_model_pic, newdata = test_pic, type = "response")
roc_pic <- roc(response = test_pic$best_pic_win, predictor = pred_pic)
optimal_pic_coords <- coords(roc_pic, x = "best",
                        best.method = "youden",
                        ret = "threshold")

pic_cutoff <- optimal_pic_coords[,1]
pred_pic <- factor(ifelse(pred_pic > pic_cutoff, 1, 0), levels = c(0, 1))
actual_pic <- factor(test_pic$best_pic_win, levels = c(0, 1))

cm_pic <- confusionMatrix(pred_pic, actual_pic)

pred_model_dir <- glm(best_dir_win ~ runtime + box_office + budget + total_noms +
                        Action + Drama + Thriller + Comedy + Fantasy + Music +
                        Romance + War + Biography + Sport + Crime + Family +
                        Musical + Comedy + Mystery + History + Horror + sci_fi, data = train_dir)

pred_dir <- predict(pred_model_dir, newdata = test_dir, type = "response")
roc_dir <- roc(response = test_dir$best_dir_win, predictor = pred_dir)
optimal_dir_coords <- coords(roc_dir, x = "best",
                        best.method = "youden",
                        ret = "threshold")

dir_cutoff <- optimal_dir_coords[,1]
pred_dir <- factor(ifelse(pred_dir < dir_cutoff, 0, 1), levels = c(0, 1))
actual_dir <- factor(test_dir$best_dir_win, levels = c(0, 1))

cm_dir <- confusionMatrix(pred_dir, actual_dir)

pred_model_actor <- glm(best_actor_win ~ runtime + box_office + budget + total_noms +
```

```
                          Action + Drama + Thriller + Comedy + Fantasy + Music +
                          Romance + War + Biography + Sport + Crime + Family +
                          Musical + Comedy + Mystery + History + Horror + sci_fi, data = train_actor)

pred_actor <- predict(pred_model_actor, newdata = test_actor, type = "response")
roc_actor <- roc(response = test_actor$best_actor_win, predictor = pred_actor)
optimal_actor_coords <- coords(roc_actor, x = "best",
                         best.method = "youden",
                         ret = "threshold")

actor_cutoff <- optimal_actor_coords[,1]
pred_actor <- factor(ifelse(pred_actor < actor_cutoff, 0, 1), levels = c(0, 1))
actual_actor <- factor(test_actor$best_actor_win, levels = c(0, 1))

cm_actor <- confusionMatrix(pred_actor, actual_actor)

pred_model_actress <- glm(best_actress_win ~ runtime + box_office + budget + total_noms +
                          Action + Drama + Thriller + Comedy + Fantasy + Music +
                          Romance + War + Biography + Sport + Crime + Family +
                          Musical + Comedy + Mystery + History + Horror + sci_fi, data = train_actress)

pred_actress <- predict(pred_model_actress, newdata = test_actress, type = "response")
roc_actress <- roc(response = test_actress$best_actress_win, predictor = pred_actress)
optimal_actress_coords <- coords(roc_actress, x = "best",
                         best.method = "youden",
                         ret = "threshold")

actress_cutoff <- optimal_actress_coords[,1]
pred_actress <- factor(ifelse(pred_actress < actress_cutoff, 0, 1), levels = c(0, 1))
actual_actress <- factor(test_actress$best_actress_win, levels = c(0, 1))

cm_actress <- confusionMatrix(pred_actress, actual_actress)
```

```
extract_cm_summary <- function(cm, name) {
  tibble::tibble(
    Outcome = name,
    Accuracy = cm$overall["Accuracy"],
    Sensitivity = cm$byClass["Sensitivity"],
    Specificity = cm$byClass["Specificity"],
    ConfusionMatrix = paste0(
      "TP: ", cm$table[2,2], ", ",
      "FP: ", cm$table[1,2], ", ",
      "TN: ", cm$table[1,1], ", ",
      "FN: ", cm$table[2,1]
    )
  )
}

cm_summary <- bind_rows(
  extract_cm_summary(cm_pic, "Best Picture"),
  extract_cm_summary(cm_dir, "Best Director"),
  extract_cm_summary(cm_actor, "Best Actor"),
  extract_cm_summary(cm_actress, "Best Actress"))
```

```
cm_summary
```

```
## # A tibble: 4 x 5
##   Outcome       Accuracy Sensitivity Specificity ConfusionMatrix
##   <chr>            <dbl>       <dbl>       <dbl> <chr>
## 1 Best Picture     0.804       0.829       0.691 TP: 47, FP: 21, TN: 257, FN: 53
## 2 Best Director    0.796       0.812       0.740 TP: 54, FP: 19, TN: 195, FN: 45
## 3 Best Actor       0.593       0.484       0.884 TP: 61, FP: 8, TN: 89, FN: 95
## 4 Best Actress     0.755       0.848       0.5   TP: 25, FP: 25, TN: 117, FN: 21
```

The confusion matrix containing the main results of the predictive model shows moderately promising performance. The Best Picture category has an accuracy of 80.4%, with sensitivity and specificity of 82.9% and 69.1%, respectively. This indicates that 82.9% of true positives are correctly identified, while 69.1% of true negatives are correctly classified. Higher specificity would be desirable, but optimizing for it would reduce the true positive rate. Best Director and Best Actress exhibit relatively similar percentages across all three measures, although Best Actress has a notably low specificity of only 50%. The category with the lowest overall accuracy is Best Actor, with 59.3% accuracy, correctly identifying true positives 48.4% of the time and true negatives 88.4% of the time. While this represents the lowest performance in overall accuracy and true positives, it achieves the highest true negative rate. A person familiar with all nominated films and actors might predict outcomes better than the model, but for blind predictions, the model is likely to perform better.