Qualitative Visualization

Qualitative Research Questions

1. How does genre affect voting average? - Qual
2. How does genre affect run time? - Qual
3. How does genre affect revenue? - Qual
4. What is the relationship between genre, revenue, runtime, and voting average. - Qual

Load Packages

```
library(janitor)
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
library(knitr)
library(kableExtra)
library(ggplot2)
library(dplyr)
```

```

Attaching package: 'dplyr'

The following object is masked from 'package:kableExtra':

    group_rows
```

```
The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v stringr   1.5.1
v lubridate 1.9.3     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.1
v readr     2.1.5
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter()     masks stats::filter()
x dplyr::group_rows() masks kableExtra::group_rows()
x dplyr::lag()        masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(tidyr)
library(rvest)
```

```
Attaching package: 'rvest'

The following object is masked from 'package:readr':

    guess_encoding
```

```r
library(plotly)
```

```
Attaching package: 'plotly'

The following object is masked from 'package:ggplot2':
    last_plot

The following object is masked from 'package:stats':

    filter

The following object is masked from 'package:graphics':

    layout
```

```r
library(esquisse)
```

Read in Data

```r
moviesRaw <- read_csv(
  file = "~/Desktop/STAT184/IMDBMovies.csv"
)
```

```
Rows: 683475 Columns: 29
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (18): title, status, backdrop_path, homepage, tconst, original_language...
dbl   (9): id, vote_average, vote_count, revenue, runtime, budget, popularit...
lgl   (1): adult
date  (1): release_date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

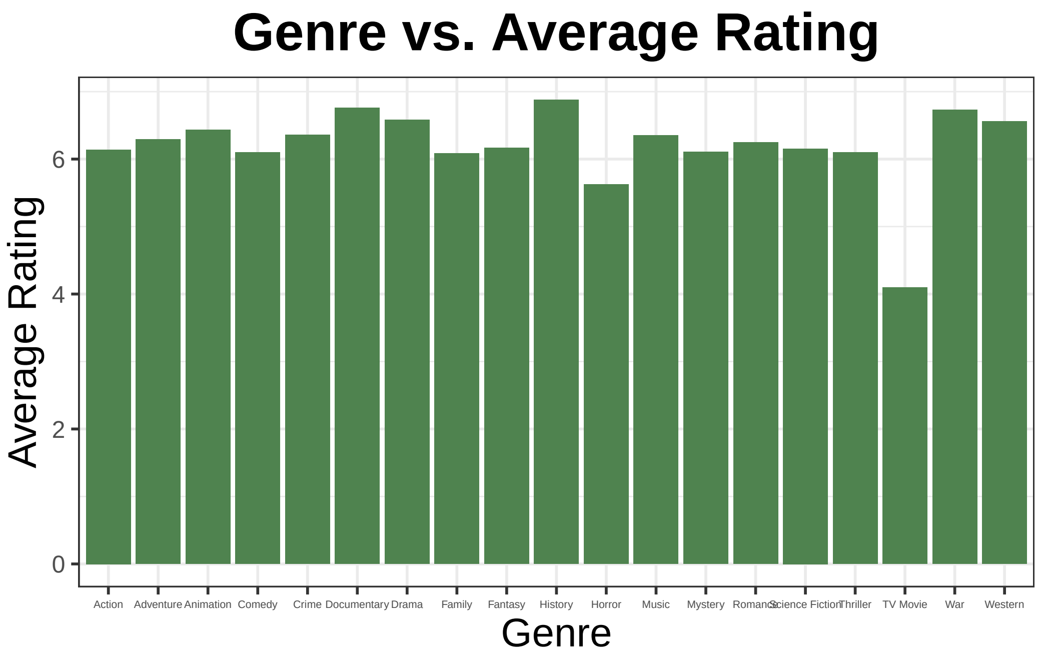
```r
moviesCleaned <- moviesRaw %>%
  dplyr:: select(-id, -vote_average, -vote_count, -overview,
  -backdrop_path, -homepage, -tconst, -poster_path, -tagline, -keywords,
  -directors, -writers, -cast, -original_title, -popularity
  ) %>%
  filter(adult == FALSE) %>%
  filter(grepl('English', spoken_languages)) %>%
  filter(original_language == "en") %>%
  dplyr:: select(-original_language, -spoken_languages, -adult) %>%
  filter(!grepl('19', release_date)) %>%
  filter(status == "Released") %>%
  filter(revenue > 1) %>%
  filter(runtime >= 30 ) %>%
  filter(runtime <= 220) %>%
  filter(budget >= 1000) %>%
  filter(numVotes >= 1000) %>%
  filter(!duplicated(title)) %>%
  filter(!grepl('UFC', title)) %>%
  drop_na()

genresWrangled <- moviesCleaned %>%
  separate_wider_delim(
    cols = genres,
    delim = ",",
    names = c("Genre1", "Genre2", "Genre3", "Genre4", "Genre5", "Genre6",
    "Genre7", "Genre8", "Genre9"),
    too_few = "align_start"
  ) %>%
  pivot_longer(
    cols = starts_with("Genre"),
    names_to = "genreNumber",
    values_to = "genre"
  ) %>%
  drop_na() %>%
  mutate(
    genre = case_match(
      .x = genre,
      " Action" ~ "Action",
      " Adventure" ~ "Adventure",
      " Crime" ~ "Crime",
      " Thriller" ~ "Thriller",
      " Science Fiction" ~ "Science Fiction",
      " Drama" ~ "Drama",
      " Comedy" ~ "Comedy",
      " TV Movie" ~ "TV Movie",
      " Family" ~ "Family",
      " Western" ~ "Western",
      " Mystery" ~ "Mystery",
      " Romance" ~ "Romance",
      " History" ~ "History",
      " War" ~ "War",
      " Fantasy" ~ "Fantasy",
      " Horror" ~ "Horror",
      " Music" ~ "Music",
      " Documentary" ~ "Documentary",
      " Animation" ~ "Animation",
      .default = genre
    )
  ) %>%
 group_by(genre) %>%
  summarize(
    minRev = min(revenue),
    Q1Rev = quantile(revenue, probs = 0.25),
    medianRev = median(revenue),
    Q3Rev = quantile(revenue, probs = 0.75),
    maxRev = max(revenue),
    avgRev = mean(revenue),
    minRating = min(averageRating),
    Q1Rating = quantile(averageRating, probs = 0.25),
    medianRating = median(averageRating),
    Q3Rating = quantile(averageRating, probs = 0.75),
    maxRating = max(averageRating),
    avgRating = mean(averageRating),
    minRun = min(runtime),
    Q1Run = quantile(runtime, probs = 0.25),
    medianRun = median(runtime),
    Q3Run = quantile(runtime, probs = 0.75),
    maxRun = max(runtime),
    avgRun = mean(runtime),
    count = n(),
    .groups = "drop"
  )

View(moviesCleaned)
```

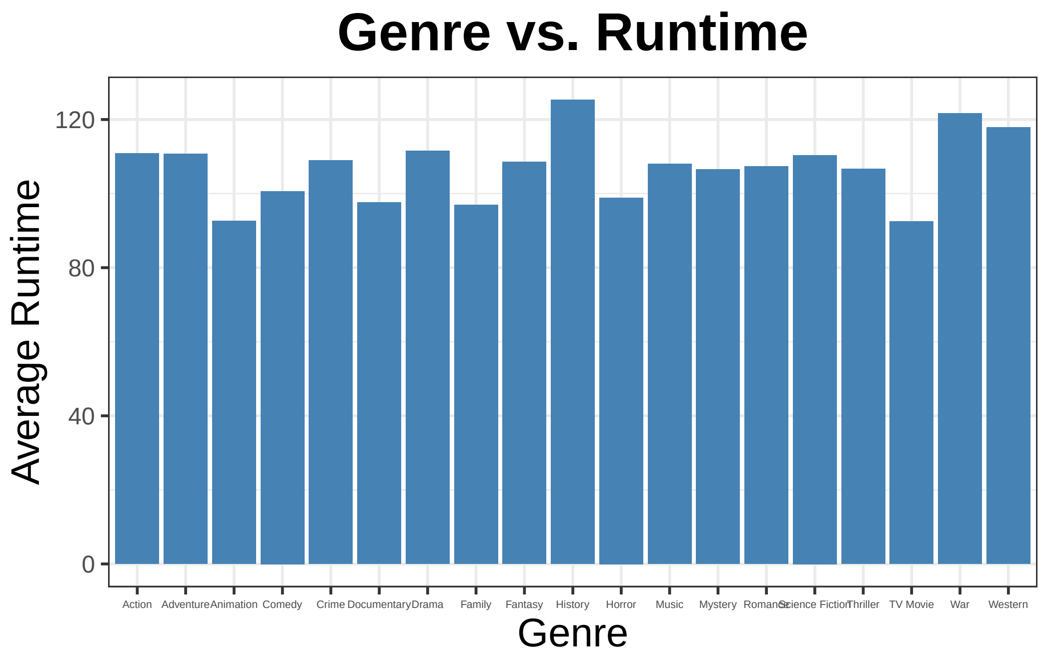
Genre and Rating

```
ggplot(genresWrangled) +
  aes(x = genre, y = avgRating) +
  geom_col(fill = "#4F834F") +
  labs(
    x = "Genre",
    y = "Average Rating",
    title = "Genre vs. Average Rating"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 20L,
    face = "bold",
    hjust = 0.5),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L),
    axis.text.x = element_text(size = 4L)
  )
```



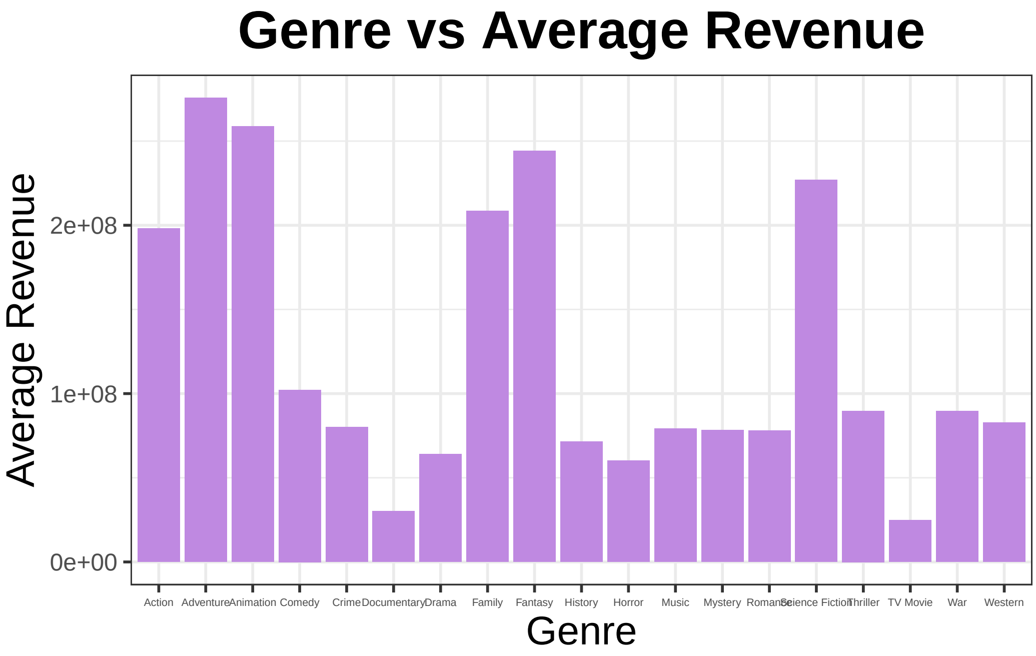
Genre and Runtime

```
ggplot(genresWrangled) +
  aes(x = genre, y = avgRun) +
  geom_col(fill = "#4682B4") +
  labs(
    x = "Genre",
    y = "Average Runtime",
    title = "Genre vs. Runtime"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 20L,
    face = "bold",
    hjust = 0.5),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L),
    axis.text.x = element_text(size = 4L)
  )
```



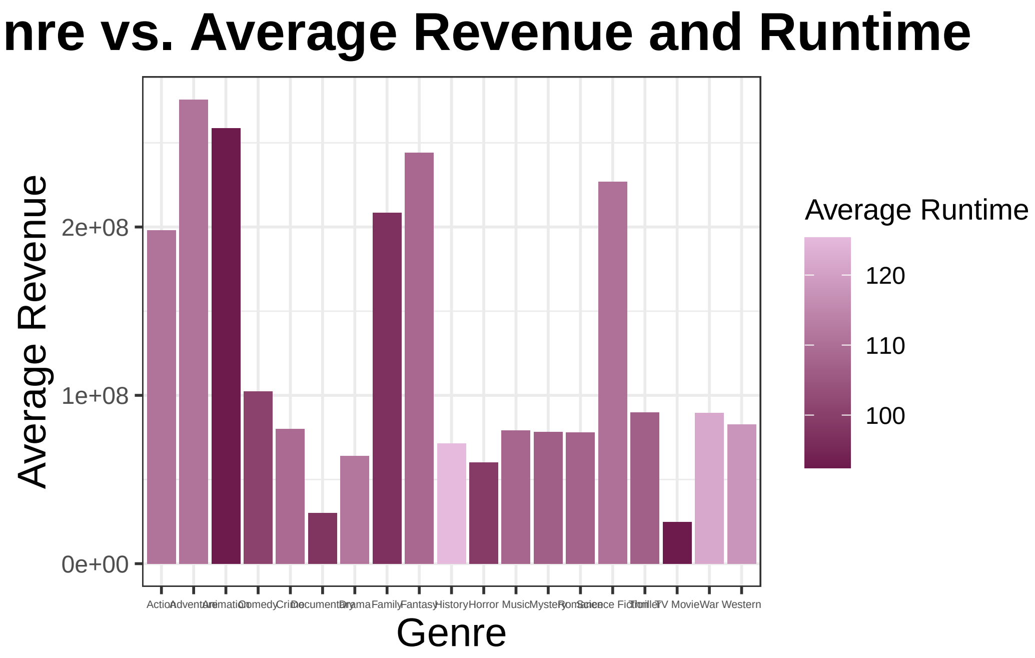
Genre and Revenue

```
ggplot(genresWrangled) +
  aes(x = genre, y = avgRev) +
  geom_col(fill = "#BF89E1") +
  labs(
    x = "Genre",
    y = "Average Revenue",
    title = "Genre vs Average Revenue"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 20L,
    face = "bold",
    hjust = 0.5),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L),
    axis.text.x = element_text(size = 4L)
  )
```



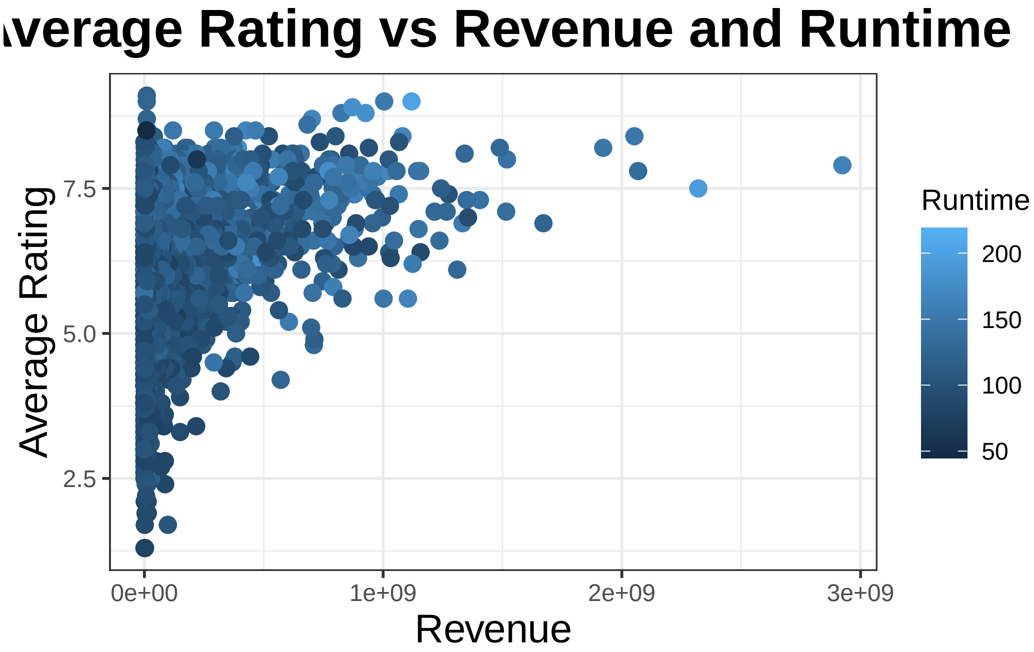
Genre vs. Average Revenue and Runtime

```
ggplot(genresWrangled) +
  aes(x = genre, y = avgRev, fill = avgRun) +
  geom_col() +
  scale_fill_gradient(low = "#6D1A4C", high = "#E5BADD") +
  labs(
    x = "Genre",
    y = "Average Revenue",
    title = "Genre vs. Average Revenue and Runtime",
    fill = "Average Runtime"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 20L,
    face = "bold",
    hjust = 0.5),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L),
    axis.text.x = element_text(size = 4L)
  )
```



Average Rating vs. Average Revenue and Runtime

```
ggplot(moviesCleaned) +
  aes(x = revenue, y = averageRating, colour = runtime) +
  geom_point(size = 2.55) +
  scale_color_gradient(low = "#132B43", high = "#56B1F7") +
  labs(
    x = "Revenue",
    y = "Average Rating",
    title = "Average Rating vs Revenue and Runtime",
    color = "Runtime"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 20L,
    face = "bold",
    hjust = 0.5),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L)
  )
```


verage Rating vs Revenue and Runtime
Average Rating
0e+00
1e+09
2e+09
3e+09
Revenue
Runtime