Qualitative Visualization

Qualitative Research Questions

1. How does genre affect voting average? - Qual
2. How does genre affect runtime?
3. How does genre affect revenue?
4. What are the ratings, revenues, and runtimes across 3 popular movie franchises?

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
```

```
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 beco
```

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

```
Attaching package: 'rvest'

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

    guess_encoding
```

```
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
```

```
library(esquisse)
```

Read in, Clean, and Wrangle Data

```
##Importing the Data----
fantasyRaw <- read_csv(
  file = "~/Desktop/STAT184/fantasy.csv"
)
```

```
Rows: 17163 Columns: 14
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (11): movie_id, movie_name, year, certificate, runtime, genre, descripti...
dbl  (3): rating, votes, gross(in $)

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.
```

```
actionRaw <- read_csv(
  file = "~/Desktop/STAT184/action.csv"
)
```

```
Rows: 52452 Columns: 14
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (11): movie_id, movie_name, year, certificate, runtime, genre, descripti...
dbl  (3): rating, votes, gross(in $)

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.
```

```
horrorRaw <- read_csv(
  file = "~/Desktop/STAT184/horror.csv"
)
```

```
Rows: 36682 Columns: 14
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (11): movie_id, movie_name, year, certificate, runtime, genre, descripti...
dbl  (3): rating, votes, gross(in $)

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.
```

```
mysteryRaw <- read_csv(
  file = "~/Desktop/STAT184/mystery.csv"
)
```

```
Rows: 18960 Columns: 14
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (11): movie_id, movie_name, year, certificate, runtime, genre, descripti...
dbl  (3): rating, votes, gross(in $)

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.
```

```
##Merging the Data----
moviesRaw <- full_join(
  x = fantasyRaw,
  y = actionRaw
) %>%
  full_join(
    y = horrorRaw
  ) %>%
  full_join(
    y = mysteryRaw
  )
```

```
Joining with `by = join_by(movie_id, movie_name, year, certificate, runtime,
genre, rating, description, director, director_id, star, star_id, votes,
```

```
`gross(in $)`)`
Joining with `by = join_by(movie_id, movie_name, year, certificate, runtime,
genre, rating, description, director, director_id, star, star_id, votes,
`gross(in $)`)`
Joining with `by = join_by(movie_id, movie_name, year, certificate, runtime,
genre, rating, description, director, director_id, star, star_id, votes,
`gross(in $)`)`
```

```r
##Cleaning the Data----
moviesCleaned <- moviesRaw %>%
  rename(revenue = `gross(in $)`
  ) %>%
  dplyr:: select(-movie_id, -description, -director_id, -star_id
  ) %>%
  drop_na() %>%
  filter(!grepl('19', year)) %>%
  filter(!duplicated(movie_name)) %>%
  mutate(runtime = readr::parse_number(runtime))

##Listing Only Relevant Movies----
relevantMovies <- moviesCleaned %>%
  separate_wider_delim(
    cols = genre,
    delim = ",",
    names = c("Genre1", "Genre2", "Genre3"),
    too_few = "align_start"
  ) %>%
  pivot_longer(
    cols = starts_with("Genre"),
    names_to = "genreNumber",
    values_to = "genre"
  ) %>%
 mutate(genre = case_match(
   genre,
   " Action" ~ "Action",
   " Mystery" ~ "Mystery",
   " Fantasy" ~ "Fantasy",
  " Horror" ~ "Horror",
  .default = genre
 )) %>%
  drop_na() %>%
  filter(
```

```
    genre == "Action" |
    genre == "Horror" |
    genre == "Mystery" |
    genre == "Fantasy") %>%
  select(-genreNumber)



##Getting Summary Statistics----
info <- list(
  Count = ~as.double(n()),
  Min = ~as.double(min(.x)),
  Q1 = ~as.double(quantile(.x,probs = 0.25, na.rm = TRUE)),
  Median = ~as.double(median(.x)),
  Avg = ~as.double(mean(.x)),
  Q3 = ~as.double(quantile(.x,probs = 0.75, na.rm = TRUE)),
  Max = ~as.double(max(.x))
)

moviesSummary <- relevantMovies %>%
  group_by(genre) %>%
  summarize(across(c(revenue,runtime), info)) %>%
  select(-runtime_Count) %>%
  drop_na() %>%
  rename(count = revenue_Count)

##Film Franchises----
harryPotterMovies <- relevantMovies %>%
  filter(grepl('Harry Potter', movie_name)) %>%
  select(-star, -genre)

harryPotterSummary <- harryPotterMovies %>%
  summarize(across(c(revenue,runtime), info)) %>%
  select(-runtime_Count) %>%
  drop_na() %>%
  rename(count = revenue_Count)

piratesMovies <- relevantMovies  %>%
  filter(grepl('Pirates of the Caribbean:', movie_name)) %>%
  select(-star, -genre) %>%
  filter(!duplicated(movie_name))
```

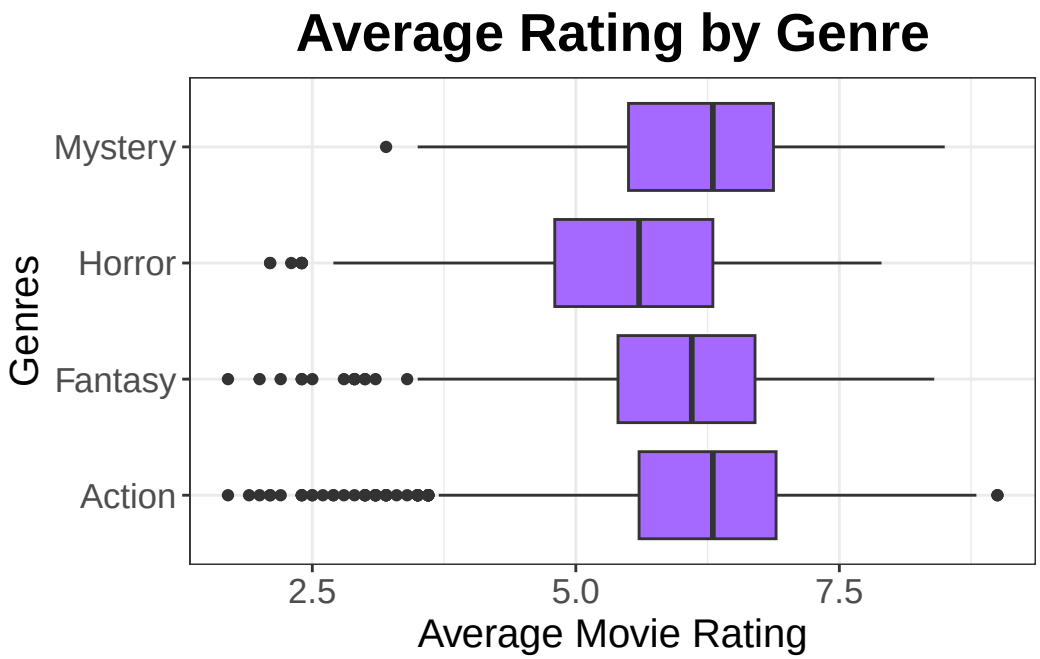
```r
piratesSummary <- piratesMovies %>%
  summarize(across(c(revenue,runtime), info)) %>%
  select(-runtime_Count) %>%
  drop_na() %>%
  rename(count = revenue_Count)

spiderMovies <- relevantMovies  %>%
  filter(grepl('Spider-Man', movie_name)) %>%
  select(-star, -genre) %>%
  filter(!duplicated(movie_name))

spiderSummary <- spiderMovies %>%
  summarize(across(c(revenue,runtime), info)) %>%
  select(-runtime_Count) %>%
  drop_na() %>%
  rename(count = revenue_Count)
```

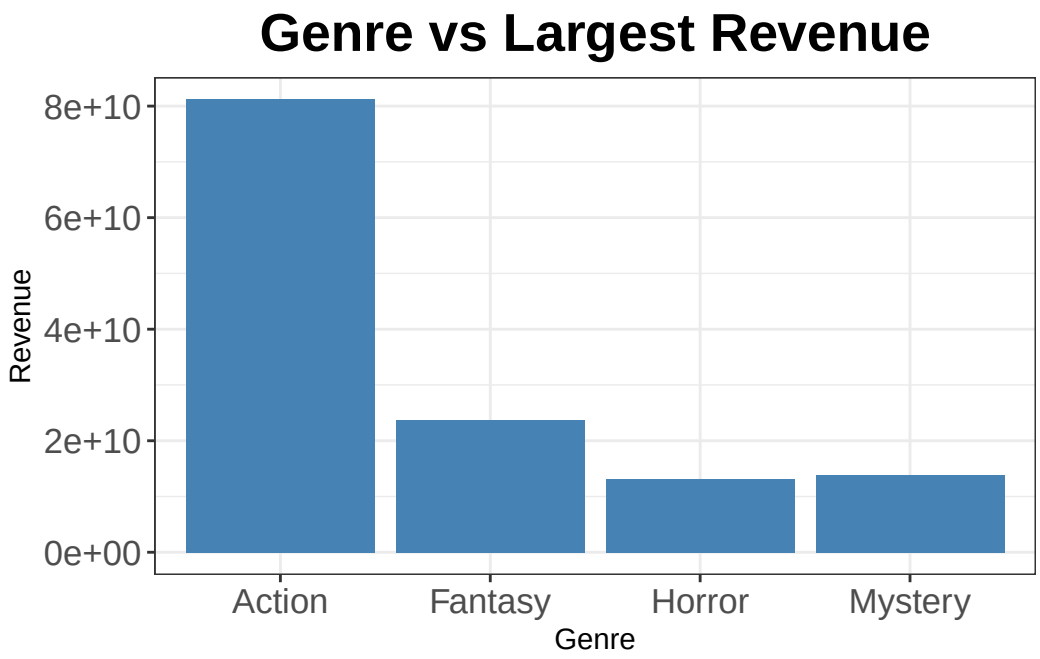
Genre and Rating

```r
ggplot(relevantMovies) +
  aes(x = rating, y = genre) +
  geom_boxplot(fill = "#A569FF") +
  labs(
    x = "Average Movie Rating",
    y = "Genres",
    title = "Average Rating by Genre"
  ) +
  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.y = element_text(size = 13L),
    axis.text.x = element_text(size = 13L)
  )
```



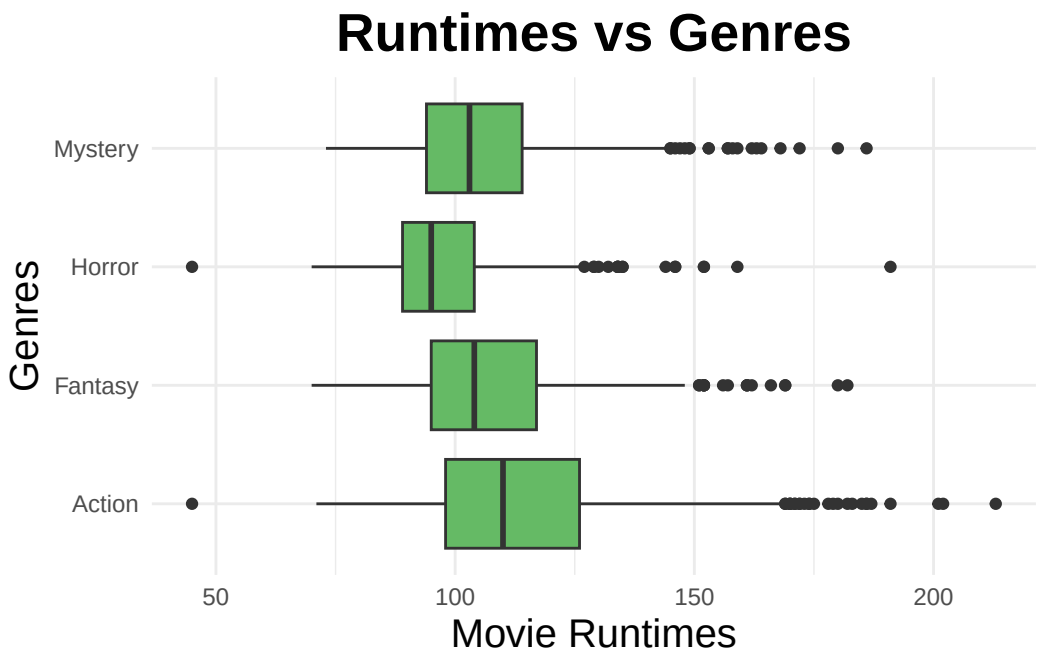
Genre and Revenue

```
ggplot(relevantMovies) +
  aes(x = genre, y = revenue) +
  geom_col(fill = "#4682B4") +
  labs(
    x = "Genre",
    y = "Revenue",
    title = "Genre vs Largest Revenue"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 20L,
    face = "bold",
    hjust = 0.5),
    axis.text.y = element_text(size = 13L),
    axis.text.x = element_text(size = 13L)
  )
```



Genre vs Runtime

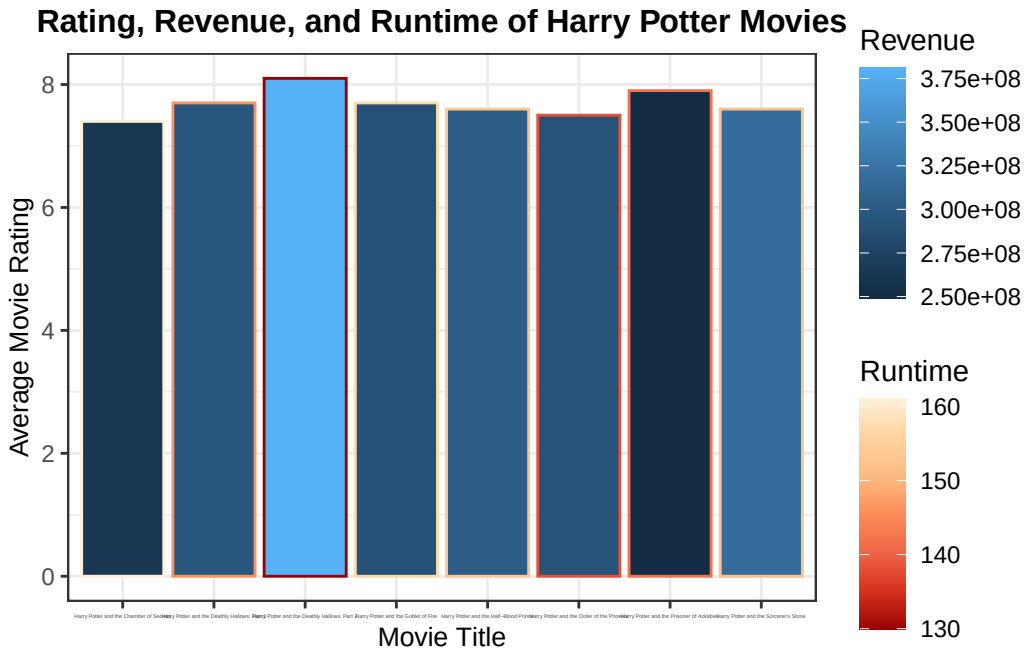
```
ggplot(relevantMovies) +
  aes(x = runtime, y = genre) +
  geom_boxplot(fill = "#65BA65") +
  labs(
    x = "Movie Runtimes",
    y = "Genres",
    title = "Runtimes vs Genres"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 20L,
    face = "bold",
    hjust = 0.5),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L)
  )
```



Harry Potter Movies

```
ggplot(harryPotterMovies) +
  aes(
    x = movie_name,
    y = rating,
    fill = revenue,
    colour = runtime
  ) +
  geom_bar(stat = "summary", fun = "sum") +
  scale_fill_gradient() +
  scale_color_distiller(palette = "OrRd") +
  labs(
    x = "Movie Title",
    y = "Average Movie Rating",
    title = "Rating, Revenue, and Runtime of Harry Potter Movies",
    fill = "Revenue",
    color = "Runtime"
  ) +
  theme_bw() +
  theme(
```

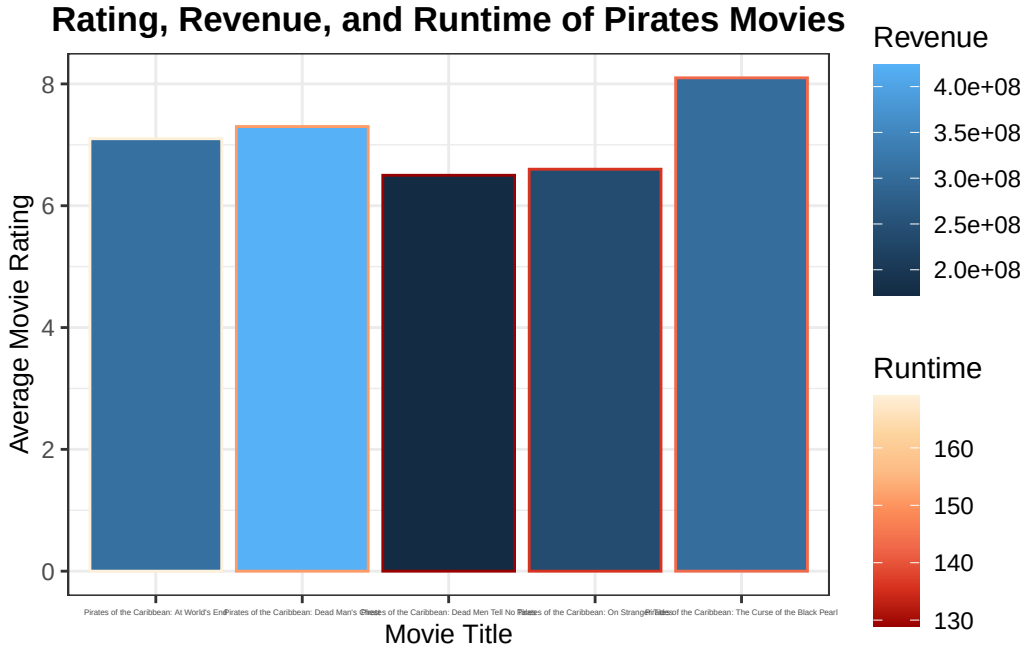
```
    plot.title = element_text(size = 12L,
    face = "bold",
    hjust = 0.5),
    axis.title.y = element_text(size = 10L),
    axis.title.x = element_text(size = 10L),
    axis.text.x = element_text(size = 2L)
  )
```



Pirates of the Caribbean Movies

```
ggplot(piratesMovies) +
  aes(
    x = movie_name,
    y = rating,
    fill = revenue,
    colour = runtime
  ) +
  geom_bar(stat = "summary", fun = "sum") +
  scale_fill_gradient() +
  scale_color_distiller(palette = "OrRd") +
```

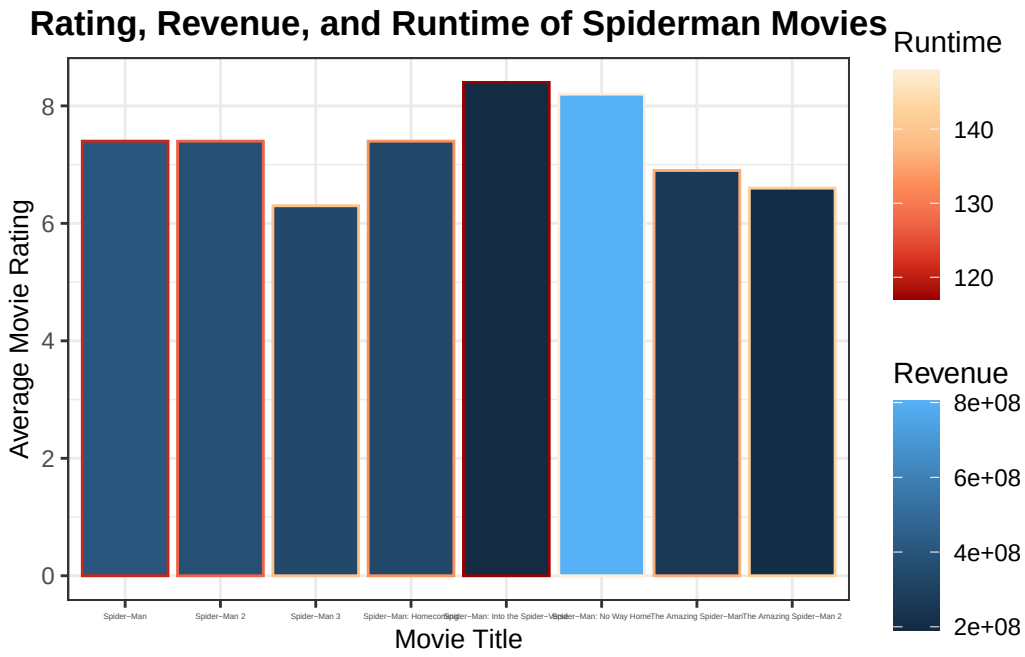
```
  labs(
    x = "Movie Title",
    y = "Average Movie Rating",
    title = "Rating, Revenue, and Runtime of Pirates Movies",
    fill = "Revenue",
    color = "Runtime"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 13L,
    face = "bold",
    hjust = 0.5),
    axis.title.y = element_text(size = 10L),
    axis.title.x = element_text(size = 10L),
    axis.text.x = element_text(size = 3L)
  )
```



Spiderman Movies

```
ggplot(spiderMovies) +
  aes(
```

```
  x = movie_name,
  y = rating,
  fill = revenue,
  colour = runtime
) +
geom_bar(stat = "summary", fun = "sum") +
scale_fill_gradient() +
scale_color_distiller(palette = "OrRd") +
labs(
  x = "Movie Title",
  y = "Average Movie Rating",
  title = "Rating, Revenue, and Runtime of Spiderman Movies",
  fill = "Revenue",
  color = "Runtime"
) +
theme_bw() +
theme(
  plot.title = element_text(size = 13L,
  face = "bold",
  hjust = 0.5),
  axis.title.y = element_text(size = 10L),
  axis.title.x = element_text(size = 10L),
  axis.text.x = element_text(size = 3L)
)
```


Rating, Revenue, and Runtime of Spiderman Movies
Average Movie Rating
0
2
4
6
8
Spider−Man
Spider−Man 2
Spider−Man 3
Spider−Man: Homecoming
Spider−Man: Into the Spider−Verse
Spider−Man: No Way Home
The Amazing Spider−Man
The Amazing Spider−Man 2
Movie Title
Runtime
140
130
120
Revenue
8e+08
6e+08
4e+08
2e+08