Wk 6 Data Transformation: Movies and Tv

Dirk Hartog

2023-10-08

The first step was to read in the csv file from guthub.

```
movieurl <- "https://raw.githubusercontent.com/D-hartog/DATA607/main/PROJECT2/movies_untidy.csv"
movies_tv <- read_csv(movieurl)
```

```
## Rows: 9999 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): MOVIES, YEAR, GENRE, ONE-LINE, STARS, Gross
## dbl (2): RATING, RunTime
## num (1): VOTES
##
## 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.
```

```
head(movies_tv)
```

```
## # A tibble: 6 x 9
##   MOVIES                YEAR  GENRE RATING `ONE-LINE` STARS  VOTES RunTime Gross
##   <chr>                 <chr> <chr>  <dbl> <chr>      <chr>  <dbl>   <dbl> <chr>
## 1 Blood Red Sky         (202~ "\nA~    6.1 "\nA woma~ "\n ~  21062     121 <NA>
## 2 Masters of the Unive~ (202~ "\nA~    5   "\nThe wa~ "\n ~  17870      25 <NA>
## 3 The Walking Dead      (201~ "\nD~    8.2 "\nSherif~ "\n ~ 885805      44 <NA>
## 4 Rick and Morty        (201~ "\nA~    9.2 "\nAn ani~ "\n ~ 414849      23 <NA>
## 5 Army of Thieves       (202~ "\nA~   NA   "\nA preq~ "\n ~     NA      NA <NA>
## 6 Outer Banks           (202~ "\nA~    7.6 "\nA grou~ "\n ~  25858      50 <NA>
```

```
summary(movies_tv)
```

```
##     MOVIES              YEAR              GENRE              RATING
##  Length:9999        Length:9999        Length:9999        Min.   :1.100
##  Class :character   Class :character   Class :character   1st Qu.:6.200
##  Mode  :character   Mode  :character   Mode  :character   Median :7.100
##                                                           Mean   :6.921
##                                                           3rd Qu.:7.800
##                                                           Max.   :9.900
##                                                           NA's   :1820
##    ONE-LINE            STARS               VOTES            RunTime
```

```
##  Length:9999        Length:9999        Min.   :      5   Min.   :  1.00
##  Class :character   Class :character   1st Qu.:    166   1st Qu.: 36.00
##  Mode  :character   Mode  :character   Median :    789   Median : 60.00
##                                        Mean   :  15124   Mean   : 68.69
##                                        3rd Qu.:   3772   3rd Qu.: 95.00
##                                        Max.   :1713028   Max.   :853.00
##                                        NA's   :1820      NA's   :2958
##     Gross
##  Length:9999
##  Class :character
##  Mode  :character
##
##
##
##
```

Before doing any transformations or analysis I wanted to clean up data and organize it a little differently.

1. The first column I worked on was the Year column. Since I was interested in using the release year of a movie or TV show later in my analysis, I needed to extract the first year listed and clean up the string.

2. This data set had both TV shows and Movies so I thought it would have been a good idea to try and label each observation accordingly. I created another column designating which observation was a TV show or a MOVIE based on certain criteria:

- Whether or not the original YEAR value had more than one year or if it had a hypen indicating that the program was still running.
- TV shows are usually 30 or 60 minutes long. I used the RunTime columnm to find those observations that were less than 75 minutes. I included observations longer than 60 since some TV shows do have an occasional special longer episode. (I understand that this may not have been the most accurate as this cut off was based on personal experience as a mediocore TV watcher and reading some articles on the internet).

```
movies_tv <- movies_tv %>%
  mutate(TYPE=(ifelse (str_detect(movies_tv$YEAR, "\\(\\d{4}.{2,}") | RunTime < 75, "TV", "MOVIE")))

head(movies_tv, 3)
```

```
## # A tibble: 3 x 11
##   MOVIES   YEAR  GENRE RATING `ONE-LINE` STARS  VOTES RunTime Gross RELEASE_YEAR
##   <chr>    <chr> <chr>  <dbl> <chr>      <chr>  <dbl>   <dbl> <chr>        <dbl>
## 1 Blood R~ (202~ "\nA~    6.1 "\nA woma~ "\n ~  21062     121 <NA>          2021
## 2 Masters~ (202~ "\nA~    5   "\nThe wa~ "\n ~  17870      25 <NA>          2021
## 3 The Wal~ (201~ "\nD~    8.2 "\nSherif~ "\n ~ 885805      44 <NA>          2010
## # i 1 more variable: TYPE <chr>
```

3a. There was a lot of cleaning to do in the STARS column as it listed both the actors and director in one cell. I extracted the actors from the STARS column and created a new column called ACTORS. I did the same thing for the directors listed and created a new column called DIRECTOR.

```
# Find and extract the string listing all the actors and assign it to a new column. Trim any white spac
movies_tv <- movies_tv %>%
  mutate(ACTORS = str_extract(movies_tv$STARS, "Stars:(\\n.*)+"))
movies_tv$ACTORS <- str_trim(movies_tv$ACTORS)

# I could not figure out how to clean and extract exactly what I want in one step so it was a multi-ste

movies_tv <- movies_tv %>%
  mutate(ACTORS = str_extract(movies_tv$ACTORS, "(\\n.*)+"))
movies_tv$ACTORS <- str_trim(movies_tv$ACTORS)
```

3b. At this point it made sense to transform the ACTORS column so that each actor was it's own observation transforming the data frame into a longer one.

```
movies_tv <- movies_tv %>% separate_longer_delim(ACTORS, delim = ", \n")
```

3c. As described above the director was extracted in a similar way from the original STARS column to create a new column called DIRECTOR.

```
movies_tv <- movies_tv %>%
  mutate(DIRECTOR = str_extract(movies_tv$STARS, "Director:.*\\n.*\\n"))

movies_tv <- movies_tv %>%
  mutate(DIRECTOR = str_extract(movies_tv$DIRECTOR, "\\n.*"))

movies_tv$DIRECTOR <- str_trim(movies_tv$DIRECTOR)
```

4. In order to do some analysis on the genres of programming this data set contained, a bit of cleanging and transforming need to be done. I extracted the first listed genre into the GENRE column and kept the others listed in another column.

```
movies_tv$GENRE <- str_trim(movies_tv$GENRE)

movies_tv <- movies_tv %>% mutate(GENRE = str_extract(movies_tv$GENRE, "(^[A-Z][a-z]*)"), GENRE_OTHER =

movies_tv$GENRE_OTHER <- str_trim(movies_tv$GENRE_OTHER)
```

5. I wanted to change some column names and select only the columns I needed for analysis.

```
movies_tv <- movies_tv %>% rename(RUN_TIME = RunTime) %>%
  select(MOVIES, GENRE, GENRE_OTHER, RATING, VOTES, RUN_TIME, RELEASE_YEAR, TYPE,
         ACTORS, DIRECTOR)

head(movies_tv)
```

```
## # A tibble: 6 x 10
##   MOVIES       GENRE GENRE_OTHER RATING VOTES RUN_TIME RELEASE_YEAR TYPE  ACTORS
##   <chr>        <chr> <chr>        <dbl> <dbl>    <dbl>        <dbl> <chr> <chr>
## 1 Blood Red S~ Acti~ Horror, Th~    6.1 21062      121         2021 MOVIE Peri ~
## 2 Blood Red S~ Acti~ Horror, Th~    6.1 21062      121         2021 MOVIE Carl ~
## 3 Blood Red S~ Acti~ Horror, Th~    6.1 21062      121         2021 MOVIE Alexa~
```

```
## 4 Blood Red S~ Acti~ Horror, Th~    6.1 21062      121         2021 MOVIE Kais ~
## 5 Masters of ~ Anim~ Action, Ad~    5   17870       25         2021 TV    Chris~
## 6 Masters of ~ Anim~ Action, Ad~    5   17870       25         2021 TV    Sarah~
## # i 1 more variable: DIRECTOR <chr>
```

```
# Save tidy data frame to csv
write.csv(movies_tv,file='/Users/dirkhartog/Desktop/CUNY_MSDS/DATA_607/PROJECT2/movies/movies_tv.csv', r
```

DATA ANALYSIS : The data was cleaned and transformed in order to investigate the relationships between the ratings of tv shows and/or movies and genre and actor in the data.

1a. Finding the top 15 genres with the highest average rating

```
top15avg <- movies_tv %>% filter(TYPE == "MOVIE") %>%
  drop_na(GENRE) %>%
  group_by(GENRE) %>%
  summarise(Average = round(mean(RATING, na.rm = TRUE),1),
            Max = max(RATING, na.rm = TRUE),
            Min = min(RATING, na.rm = TRUE)) %>%
  arrange(desc(Average)) %>%
  head(15)

top15avg
```

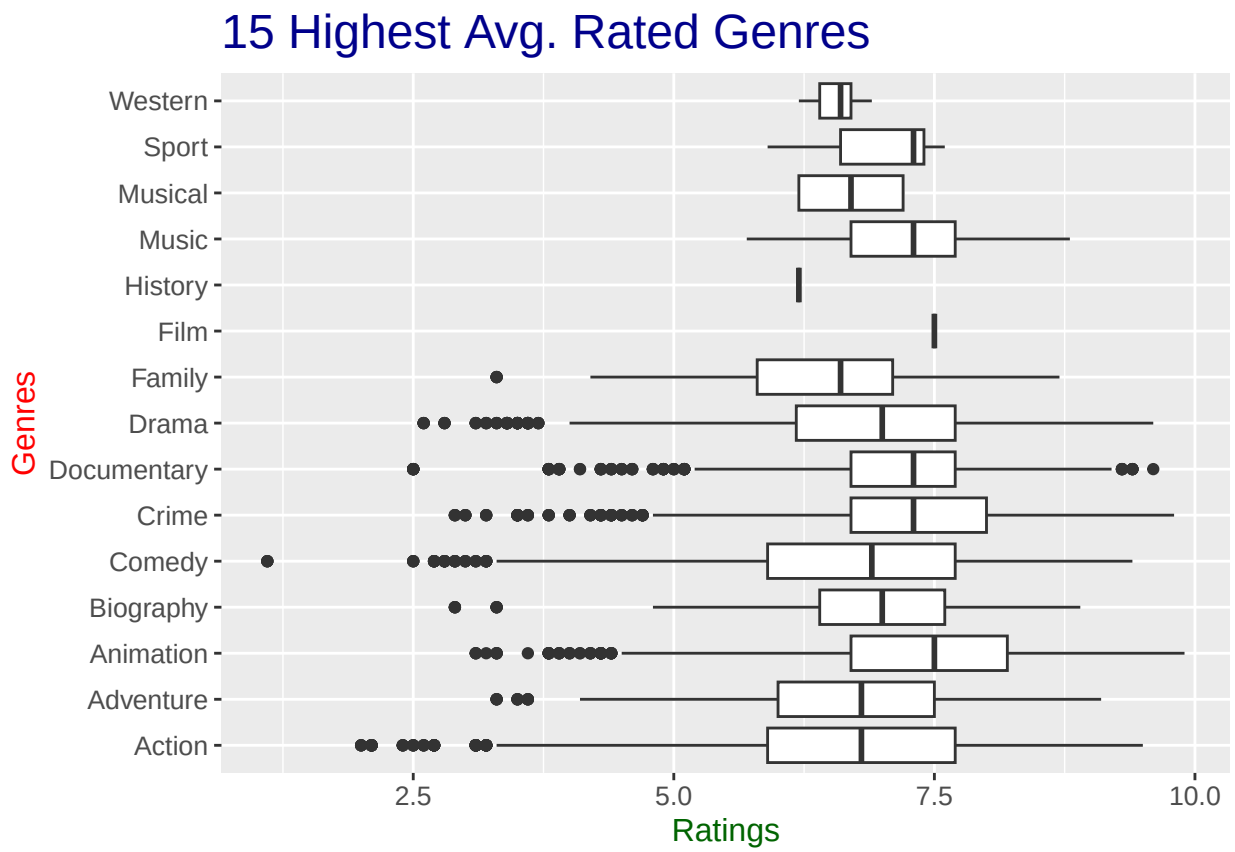
```
## # A tibble: 15 x 4
##    GENRE       Average   Max   Min
##    <chr>         <dbl> <dbl> <dbl>
##  1 Film            7.5   7.5   7.5
##  2 Music           7.4   8.3   6.5
##  3 Musical         7.2   7.2   7.2
##  4 Documentary     7.1   9.3   3.8
##  5 Animation       6.8   8.6   3.8
##  6 Biography       6.6   8.9   2.9
##  7 Western         6.6   6.9   6.2
##  8 Crime           6.3   8.6   3
##  9 Drama           6.2   8.6   2.6
## 10 History         6.2   6.2   6.2
## 11 Adventure       6.1   8.3   3.3
## 12 Family          5.9   7.8   4.2
## 13 Sport           5.9   5.9   5.9
## 14 Action          5.8   8.9   2
## 15 Comedy          5.8   8.6   2.5
```

1b. Visualize the distribution of the ratings among the top 15

```
#create a new data frame with only the top15 highest rated genres
df <- movies_tv %>% drop_na(GENRE) %>%
  filter(GENRE %in% top15avg$GENRE)

ggplot(data = df, mapping = aes(y = RATING,x = GENRE)) +
```

```
  geom_boxplot() +
  ggtitle("15 Highest Avg. Rated Genres") +
  ylab("Ratings") +
  xlab("Genres") +
  theme(axis.title.x = element_text(color="darkgreen",size=12),
        axis.title.y = element_text(color="red", size=12),
        axis.text.x = element_text(size=10),
        axis.text.y = element_text(size=10),
        plot.title = element_text(color="darkblue",
                                  size=18)) +
  coord_flip()
```

```
## Warning: Removed 4243 rows containing non-finite values (‘stat_boxplot()‘).
```



Conclusions Using box plots is helpful to visualize a lot of data among a variable and making comparisons across variables with categorical data types. In this plot we can see the distribution of ratings within the top 15 movie genres with the highest average rating. Here we can see that the median values of each genre are between 6 and 7.5. This also gives us an idea of the range of values in each genre and any outlines present. It looks like there are few and maybe even 1 value in the Film and History genres which future considerations may be looking at the most common genres listed.

2a. Find the top 10 actors or actresses who appeared the most in this data set

```
movies_only <- movies_tv %>% filter(TYPE == "MOVIE") %>%
  drop_na(ACTORS)
```

```
top10actors <- movies_only %>% group_by(ACTORS) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(10)

top10actors
```

```
## # A tibble: 10 x 2
##    ACTORS             n
##    <chr>          <int>
##  1 Adam Sandler      10
##  2 Bruce Willis      10
##  3 Liam Neeson       10
##  4 Luis Tosar         9
##  5 Mario Casas        9
##  6 Gary Oldman        8
##  7 James Franco       8
##  8 Jason Statham      8
##  9 Dwayne Johnson     7
## 10 Ian McKellen       7
```

2b. Find the average rating of the movies the actors above were in.

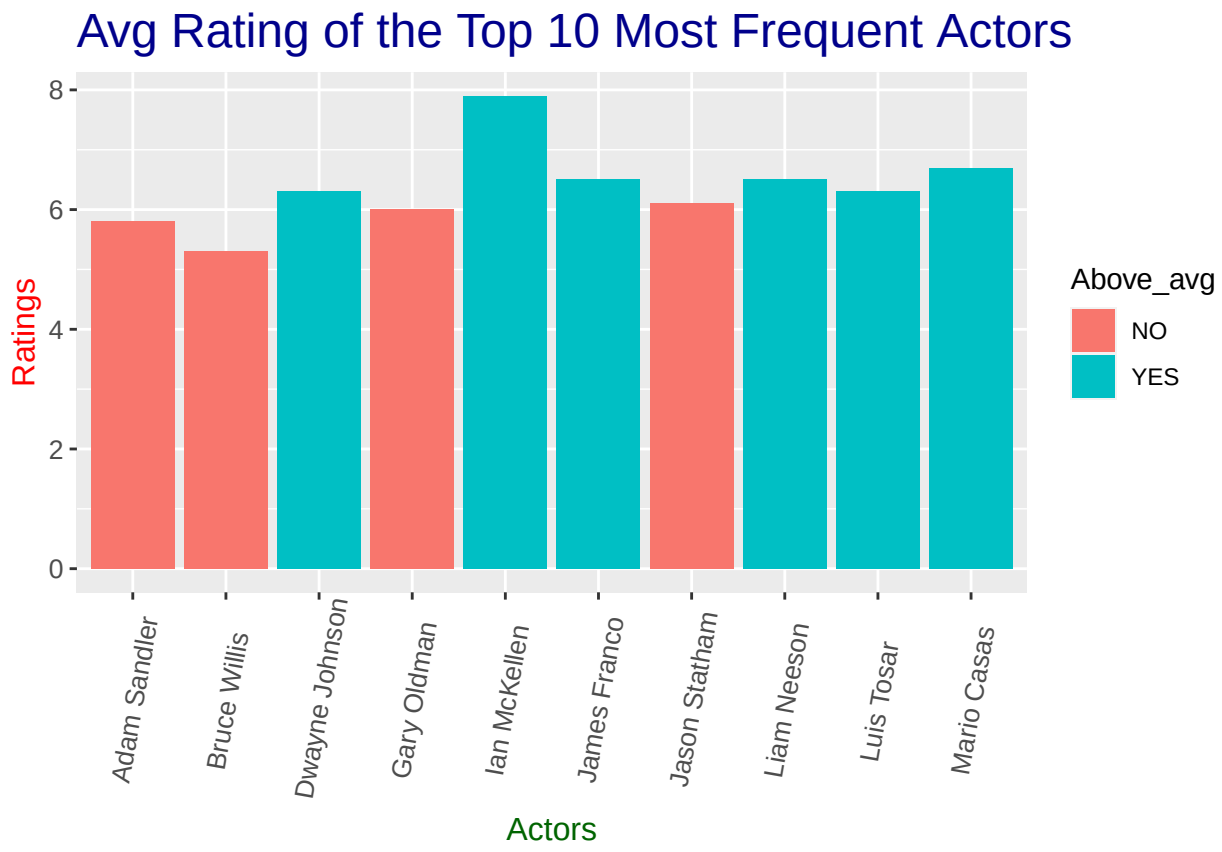
```
df2 <- movies_only %>%filter(ACTORS %in% top10actors$ACTORS)


top10avg <- df2 %>% group_by(ACTORS) %>%
  summarise(Average = round(mean(RATING, na.rm = TRUE),1),
            Max = max(RATING, na.rm = TRUE),
            Min = min(RATING, na.rm = TRUE)) %>%
  arrange(desc(Average))

top10avg
```

```
## # A tibble: 10 x 4
##    ACTORS         Average   Max   Min
##    <chr>            <dbl> <dbl> <dbl>
##  1 Ian McKellen       7.9   8.9   5.8
##  2 Mario Casas        6.7   8.1   5.6
##  3 James Franco       6.5   7.5   4.8
##  4 Liam Neeson        6.5   7.7   5.6
##  5 Dwayne Johnson     6.3   7.1   5.2
##  6 Luis Tosar         6.3   6.7   5.6
##  7 Jason Statham      6.1   7.1   3.8
##  8 Gary Oldman        6     6.9   4.8
##  9 Adam Sandler       5.8   7.4   4.8
## 10 Bruce Willis       5.3   7.8   3.1
```

2c. We want to evaluate this against the average of all movies in the data set

```r
top10avg <- top10avg %>%
  mutate(Above_avg = ifelse(top10avg$Average > mean(movies_only$RATING, na.rm = TRUE), "YES", "NO"))
```

```r
ggplot(top10avg, aes(x = ACTORS, y = Average, fill = Above_avg)) +
  geom_col() +
  ggtitle("Avg Rating of the Top 10 Most Frequent Actors") +
  ylab("Ratings") +
  xlab("Actors") +
  theme(axis.title.x = element_text(color="darkgreen",size=12),
        axis.title.y = element_text(color="red", size=12),
        axis.text.x = element_text(size=10, angle = 80, hjust = 0.45, vjust = 0.5),
        axis.text.y = element_text(size=10),
        plot.title = element_text(color="darkblue",
                                  size=18))
```



Conclusions In this plot we can see the average movie rating of each of the top 10 most most frequently listed actors in this data set. We can see that over half of the actors in this list reached above the average rating for all movies 'r mean(movies_only$RATING, na.rm = TRUE)' in the data set. I don't think we can draw strong conclusions about the influence an actor has on the raings of movies just with this plot alone. It might be interesting to expand this and look at the average movies ratings across all actors in the data set.