Project 1

2022-10-30

Marvel versus DC: Ratings and Academy Awards

Authors: Doan Nguyen, Ethan Chang, Natleigh Burns, Rhean Palencia

1.Introduction:

In today’s society debates are inevitable, this is caused by everyone’s will to be right. This can be said for the Marvel vs DC Cinematic universe, which consists of some die-hard fans on both sides, advocating for why people should support one as opposed to the other. Personal bias does play a heavy role in the vouching from each side, but when doesn’t it? To gain a better outlook on the debate we decided to obtain data that would exploit the success obtained from each universe based on the movie.

The data sets we have selected to be utilized in our project were obtained from a website known as “Kaggle.” This online community platform homes a multitude of data sets for users to upload, manipulate and use to conduct research. The data sets we deemed best for the research we wanted to conduct are the “*MARVEL vs. DC - IMDB & ROTTEN TOMATOES*”(2) and “*The Oscar Award, 1927 - 2020.*”(1)

The ” *MARVEL vs. DC - IMDB & ROTTEN TOMATOES*” data set consists of 19 variables, two of which are shared with the “*The Oscar Award, 1927 - 2020*” data set, which contains a total of 7 variables. The two variables in common within each data set are Title and Year**/**Year_film.

- **Expectation**:

We expect that Marvel movies would be more successful in terms of popular opinion than DC, but DC would garner more favor from critics than Marvel. As such, we expect Marvel to have a higher tomato rating/score and IMDb rating, also we do expect DC to have more Oscar’s nominations.

Loading Data

```
#We loaded in 2 of the data that we are using for this project
oscar_award <- read.csv("the_oscar_award.csv")
marvelvsdc <- read.csv("marvelvsdc.csv")
```

2. Tidying:

```
#Tidying the data, renaming column and adjust some value of category column in oscar_award data
 so that it would fit better later.
oscar_award<-oscar_award%>%rename(title=film)
oscar_award<-oscar_award%>%
  mutate(category = tolower(category))%>%
  mutate(category = str_to_title(category))
```

First, we are using `rename` function to rename column “film” as “title,” so that both data set would have a matching column. Then we are using `mutate` to adjust the variable names as we need.

Note: We used `pivot_longer` and `pivot_wider` later to rearrange summary statistics in part 4.Wrangling. Our data are not exactly tidy but we kept it this way so it’s easier to manipulated it later.

3.Joining/Merging

Here, we are first joining 2 data sets marvelvsdc and oscar_award by common variable “title”.

```
#Joining two data set by title
joineddata<-left_join(marvelvsdc, oscar_award, by ="title")
#Check the number observations each data set.
count(oscar_award)
```

```
##       n
## 1 10395
```

```
count(marvelvsdc)
```

```
##    n
## 1 90
```

```
#Check the number of variable in each data set.
length(variable.names(oscar_award))
```

```
## [1] 7
```

```
length(variable.names(marvelvsdc))
```

```
## [1] 19
```

```
#Create a new data set that retain only rows in both set so we can figure out which variables in
one data set but not the other and the number of variables in common.
MD_Oscar<-inner_join(oscar_award, marvelvsdc)
```

```
## Joining, by = "title"
```

```
length(variable.names(MD_Oscar))
```

```
## [1] 25
```

First, we used `left_join` function to merge 2 data set by “title” to create a new data set, named it joineddata.

To learn how many observations in each data set, we used `count` function and applied it in each data set. Then to learn how many variable there are in each data set, we used `variable.names` to get all variable names, then used `length` function to counts the number of characters in string and returns the number.

Lastly, to get the number of variables in one data set but not the other and the number of variables in common, we used `inner_join` function to create a data set that only have variable in one data set but not the other. Then used `variable.names` to get all the variable names and used `length` to get the number of variables in common.

Overall, what we learned about these two data sets and the joined data set are:

- Total observations in each data set: The oscar_award data set has 10395 observation and the marvelvsdc data set has 90 observations.
- Variables in each data set: The oscar_award data set has 7 variables while the marvelvsdc datatset has 19 variables.
- Variables in one data set but not the other and the number of variables in common
    - We have 2 variables in common: title/film, year/year_film.
    - marvelvsdc data set has 17 variables that are not in oscar_award while oscar_award has 5 variables that are not in common with the marvelvsdc data set.
- After joining, the joined data set has 125 rows/observations with 25 variables.
- In total, we dropped 10270 rows/observations from oscar_award and added 35 observations into the marvelvsdc data set.

Potential issues: duplicated movies and missing some data, column or rows can contains \`NA\` which could lead to difficult in calculating and computing.

4.Wrangling data

```
#Created a new data set that called joined, that only containt
joined<-
  joineddata%>%
  select(title, imdb_rating,imdb_votes, imdb_gross,tomato_meter, tom_aud_score, entity, id, cate
gory, winner)
#We created another data set called joined2 from the first data set. This data set we grouped it
by title and added one new column called nominated to indecate if the movie was nominated for Os
car Award or not, then added another column to count how many nominations each movie have. Then
 pick out all column except category and winner that only contain distinct movies that has the i
mdb gross is not 0 and has more than 100000.
joined2<-joined%>%
  group_by(title)%>%
  mutate(nominated = as.numeric(!is.na(category)))%>%
  mutate(num_nominations = sum(nominated))%>%
  select(-category, -winner)%>%
  unique()%>%
  filter(imdb_gross!=0, imdb_votes >100000)
#We then created 2 tables that each of them in descending order for tomato audience score and nu
mber of nominations.
tomatoscore<-joined2%>%
  select(title,entity,tom_aud_score)%>%
  arrange(desc(tom_aud_score))%>%
  head()%>%
  kable()
numnominations<-joined2%>%
  select(title,entity,num_nominations)%>%
  arrange(desc(num_nominations))%>%
  head()%>%
  kable()
```

In this part, we first used `select` to pick all the columns we want to use (which are title, imdb_rating,imdb_votes, imdb_gross,tomato_meter, tom_aud_score, entity, id, category, winner) from joineddata data set and named this data set joined.

We then created another data set called joined2, using `group_by` function and group them by "title" then using `mutate` to add a new column to tell us whether or not the movie was nominated or not. To do this, we checked if the column category of each row is "NA" or not and return a numeric value. We named this column "nominated." After that, we used `mutate` again to create another column as the number of nomination that movie get using `sum` , and named it as "num_nominations".

We now interested in other columns but not category and winner, so we used `select` to pick it out. Since there are multiple duplicate of movies in this data set, to get rid of it, we used `unique` to get only one row for each movie. The last step is that using `filter` to get the movie that has has the gross profit not 0 and has the imdb_votes above 100000 so that the data would not be bias.

To achieved what we want which is top 6 movies in tomato audience score and number of nominations, we used `select` to pick out only title, entity, and tom_aud_score/num_nominations then used `arrange` on tom_aud_score/num_nominations by descendant order using `desc` . Then make each of them into a table using `kable`

```
tomatoscore
```

title	entity	tom_aud_score
Spider-Man: Far from Home	MARVEL	95
Batman Begins	DC	94
The Dark Knight	DC	94
Captain America: The Winter Soldier	MARVEL	92
Guardians of the Galaxy	MARVEL	92
Iron Man	MARVEL	91

```
numnominations
```

title	entity	num_nominations
Joker	DC	11
The Dark Knight	DC	8
Black Panther	MARVEL	7
Superman	DC	5
Spider-Man 2	MARVEL	3
Batman Forever	DC	3

Spider-Man: Far from Home from Marvel has the highest tomato audience score with 95/100 score, while Joker from DC has the most number of Oscar’s nominations with 11 nominations.

First categorical variables summary stats

Marvel vs DC: Proportion of tomato rating in term of its freshness.

```
#Create a new data set name tomato from joined2 dataset, then added another column to indecated
 if the movie is FRESH or ROTTEN based on their's tomato_meter score.
tomato<-joined2%>%
  mutate(tomato_rating=if_else(tomato_meter>=60,"FRESH","ROTTEN"))%>%
  group_by(entity, tomato_rating)

#Create a proportion table for fresh vs rotten from previous data set.
tomato%>%
  summarise(n=n())%>%
  pivot_wider(names_from = tomato_rating,
              values_from = n)%>%
  mutate(Proportion = FRESH/(ROTTEN+FRESH))%>%
  kable()
```

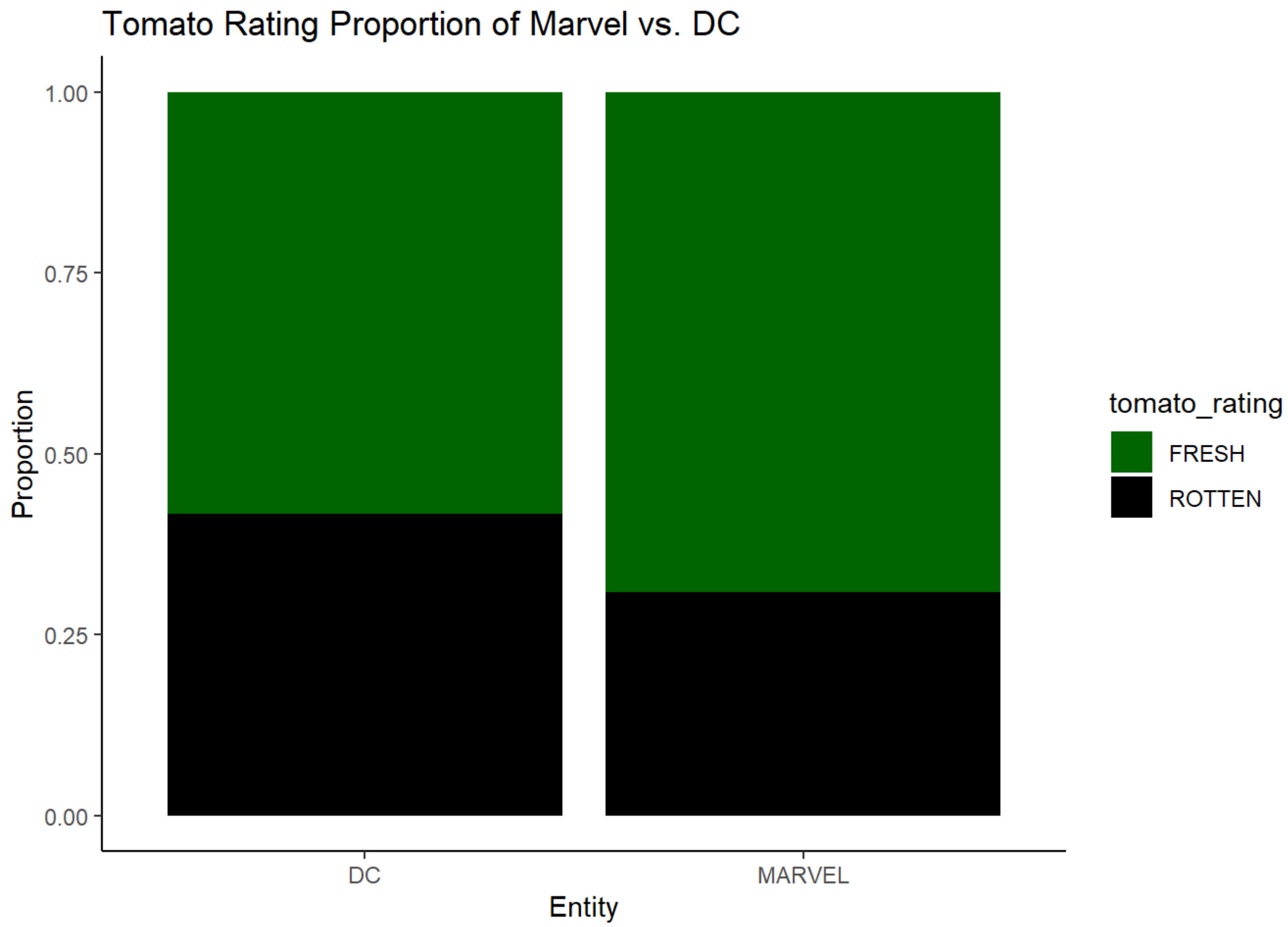
```
## `summarise()` has grouped output by 'entity'. You can override using the
## `.groups` argument.
```

entity	FRESH	ROTTEN	Proportion
DC	14	10	0.5833333
MARVEL	36	16	0.6923077

First, we `mutate` to create a new column, called “tomato_rating” that indicated whether the movie is FRESH or ROTTEN using their “tomato_meter”.Using `group_by` by “entity” and “tomato_rating” then using `summarise` to create a data set that calculate the sum of each FRESH/ROTTEN for Marvel/DC seperately. Next, using `pivot_wider` to make FRESH/ROTTEN become separate column and then using \`mutate\` to create a new column named “Proportion,” calculate the proportion of FRESH rating of each entity. Then, we`re using `kable` to make it into a table.

Visualization

```
#Create a bar plot or proportion with x-axis as entity (Marvel/DC) and added fill layer with tom
ato rating to it. Adjust the theme, labs names and scale.
tomato%>%
  group_by(entity,tomato_rating)%>%
  ggplot(aes(x= entity, fill = tomato_rating))+
  geom_bar(position = "fill")+
  labs(x = "Entity", y = "Proportion", title = "Tomato Rating Proportion of Marvel vs. DC ")+
  theme_classic()+
  scale_fill_manual(values = c("darkgreen", "black"))
```



To create the barplot that show the proportion of freshness for each franchise, we first used `group_by` and group the data by entity and tomato_rating. After that, we used `ggplot` and `geom_bar` then plot the bar plot with entity as x-axis, and fill by tomato_rating to create a proportion in each bar. We then used `labs` to adjust the name of x-axis and y-axis and its title, we also used `theme_classic` and `scale_fill_manual` to adjust the theme and scale to make it easier to read.

Analyzing:

- According to our table and bar graph from above, 58.3% of DC movies had FRESH tomato ratings, which is less than the 69.2% FRESH tomato rating percentage for Marvel films. If we set aside other considerations like the number of films Marvel has produced overall or the demographics of each franchise, we can claim that the general public favors Marvel’s films slightly more than DC’s.

Second categorical variables summary stats

Marvel vs DC: Target Audience.

We are interested in the target audience of each franchise and its proportion within each type of audience.

```
#Create a table of proportion of mpa_rating for each franchise that only take movies from 1952 t
ill now and
joineddata%>%
  filter(year >1952)%>%
  group_by(mpa_rating, entity)%>%
  summarise(n=n())%>%
  pivot_wider(names_from = entity, values_from = n)%>%
  mutate(Proportion = DC/(DC+MARVEL))%>%
  kable()
```

```
## `summarise()` has grouped output by 'mpa_rating'. You can override using the
## `.groups` argument.
```

mpa_rating	DC	MARVEL	Proportion
PG	11	2	0.8461538
PG-13	31	57	0.3522727
R	14	8	0.6363636

So to able to get the proportion of the rating for each franchise, we first used `filter` the joineddata by years that are greater than 1952. After that, we used `group_by` to group the data by mpa_rating and entity. We then create a summary data set using `summarize` that count the total movies in each category of mpa_rating by each entity (Marvel or DC).

The data set after that was a bit hard to read and calculate, so then `pivot_wider` was used to increase the number of columns, create column “MARVEL” and “DC” that in each of the column will contain the number of movies in each type of rating. Last step, we used `mutate` to create a new column, named “Proportion,” then got the value of proportion of the ratings by calculating DC divided by DC + Marvel together. We then make it into a table using `kable` function.

Analyzing:

- Based on our table and calculation, we found the proportion of movies that has rating PG is 84.6% DC and 15.4% Marvel. The proportion of movies with a rating of PG-13 is 35.2% DC and 64.8% is Marvel. Lastly, the proportion of movies with a rating of R is 63.6% DC whereas 36.4% is Marvel.
- Looking at our results, we can see that a majority of DC movies cater to the more mature audiences (Rated R) whereas Marvel has a smaller proportion of rated R movies. In contrast, Marvel has a larger proportion of movies that cater towards a younger audience(PG-13). This can explain of why Marvel is slightly bit popular than DC in general since more people grow up with it and carry on. It tend to make audience be more fonder of its franchise.

First numerical summary stat

Marvel versus DC: IMDb RATING

We are now interesting in each franchise’s statistic number in term of IMDb rating.

```
#We created a table of summary stats including the mean and standard deviation in term of imdb r
ating for each franchise.
imdbmean<-joined2%>%
  group_by(entity)%>%
  summarise(mean = mean(imdb_rating),
            standard_deviation = sd(imdb_rating))
imdbmean%>%kable()
```

entity	mean	standard_deviation
DC	6.645833	1.3516429
MARVEL	6.946154	0.9479517

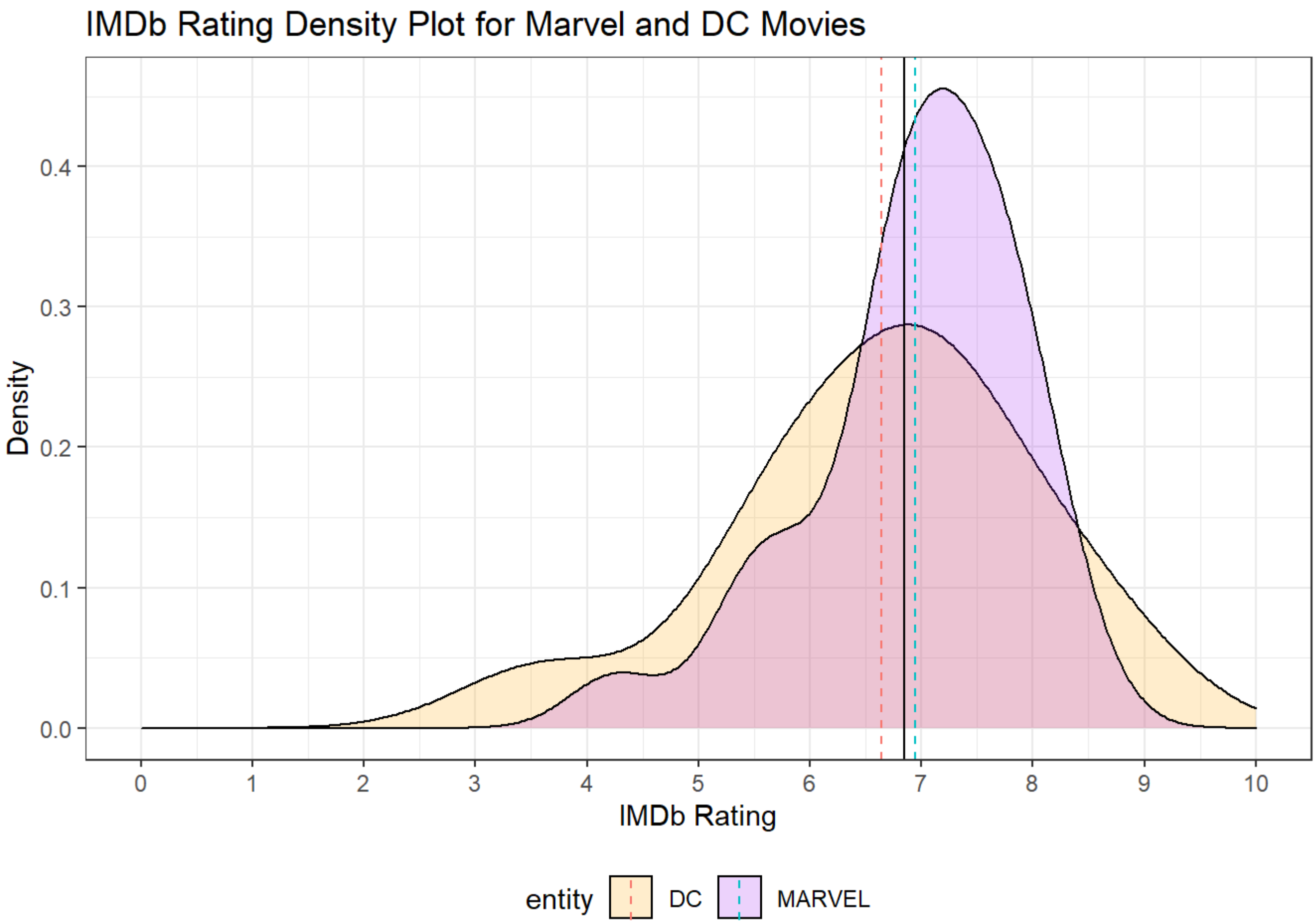
To create a summary statistic table of IMDb rating, we first used `group_by` to group the data set by entity then using `summarise` to calculate the mean by using `mean` function and standard_deviation using `sd` function for IMDb rating. We made it into a table as well using `kable`.

Analyzing:

For our first numerical summary statistics, we compared the IMDb ratings of movies appearing in Marvel and DC franchises. From the table produced, it can be seen that Marvel movies tend to have a slightly higher IMDb rating on average compared to DC movies. Marvel movies also have a lower standard deviation compared to DC movies, meaning most of the IMDb ratings for Marvel movies will be more concentrated around the mean compared to those of DC movies, which we expect to be more spread out.

Visualization:

```
#We created a density graph for each franchise with x as IMDb rating. The density graph includes
the mean of the rating for each franchise and overall mean so we can compare it.
joined2%>%
  group_by(entity)%>%
  ggplot(aes(x = imdb_rating, fill = entity))+
  geom_density(alpha = 0.2, adjust = 1.4)+
  geom_vline(data=imdbmean, aes(xintercept=mean, color=entity), linetype="dashed")+
  geom_vline(aes(xintercept=mean(imdb_rating)))+
  scale_fill_manual(values = c("orange", "purple"))+
  theme_bw()+
  theme(legend.position="bottom") +
  scale_x_continuous(breaks = 0:10, limits = c(0, 10)) +
  labs(x = "IMDb Rating", y = "Density", title = "IMDb Rating Density Plot for Marvel and DC Mov
ies")
```



To create the visualization for this, we used `group_by` function to group the data by entity, and start plotting a density graph using `ggplot` and `geom_density` function with x as imdb_rating and layer them with `fill = entity` to create two density graph. We added dashed lines to represent the mean of IMDb rating for each franchise and a normal line for overall mean, using `geom_vline` . After that, we used `scale_fill_manual` to change the color of each graph, change the x scale by `scale_x_continuous` to extend the limit of x-axis. We also changed the theme by `theme_bw` and `theme` to change the position of legend. Lastly, we changed the name of title and the axis by using `labs` .

Analyzing:

Now, when looking at the plot, the summary statistics and expectations can be clearly seen. The dashed lines depicting the mean of DC and Marvel movies show that Marvel movies' IMDb ratings, on average, tend to be higher than those of DC. The density plot also shows how an overwhelming majority of Marvel movies tend to do fairly well, slightly better than most DC movies, hovering around the 7-8 range. The greater variability of DC movie ratings due to the higher standard deviation can also be seen in the plot as DC movies tend to overtake Marvel in density when looking further below or above the mean. These trends seem to suggest that, in terms of IMDb movie ratings, Marvel will provide consistently better movies on average compared to DC.

Second + Third numerical summary stat

Marvel vs. DC: CRITIC VS PUBLIC SCORE

Next, we are interested in how critic and public perceived these movies from Marvel and DC franchise.

```
#Create a summary statistic table that include the mean and standard deviation for tomato audience score for each franchise.
audience<-tomato%>%
  group_by(entity)%>%
  summarise(audience_score=mean(tom_aud_score), aud_st_dev = sd(tom_aud_score))
#Create a summary statistic table that include the mean and standard deviation for tomato critic score for each franchise.
critic<-tomato%>%
  group_by(entity)%>%
  summarise(critic_score = mean(tomato_meter), crit_st_dev = sd(tomato_meter))
#Merge 2 statistic table together
tomato_score<-full_join(audience, critic)
```

```
## Joining, by = "entity"
```

```
tomato_score%>%kable()
```

entity	audience_score	aud_st_dev	critic_score	crit_st_dev
DC	69.00000	21.59106	61.29167	27.02733
MARVEL	72.36538	18.67895	68.69231	24.98434

To create a table of summary statistic for tomato audience score and critic score, we first used `group_by` function to group the data by “entity” then using `summarise` to calculate the mean by using `mean` function and standard_deviation using `sd` function for tomato audience score and save it as “audience_score” and “aud_st_dev.” This gave us the average of tomato audience score and its standard deviation by each entity. We repeated the previous step but with tomato_meter, this gave us the average tomato score from critic and its standard deviation.

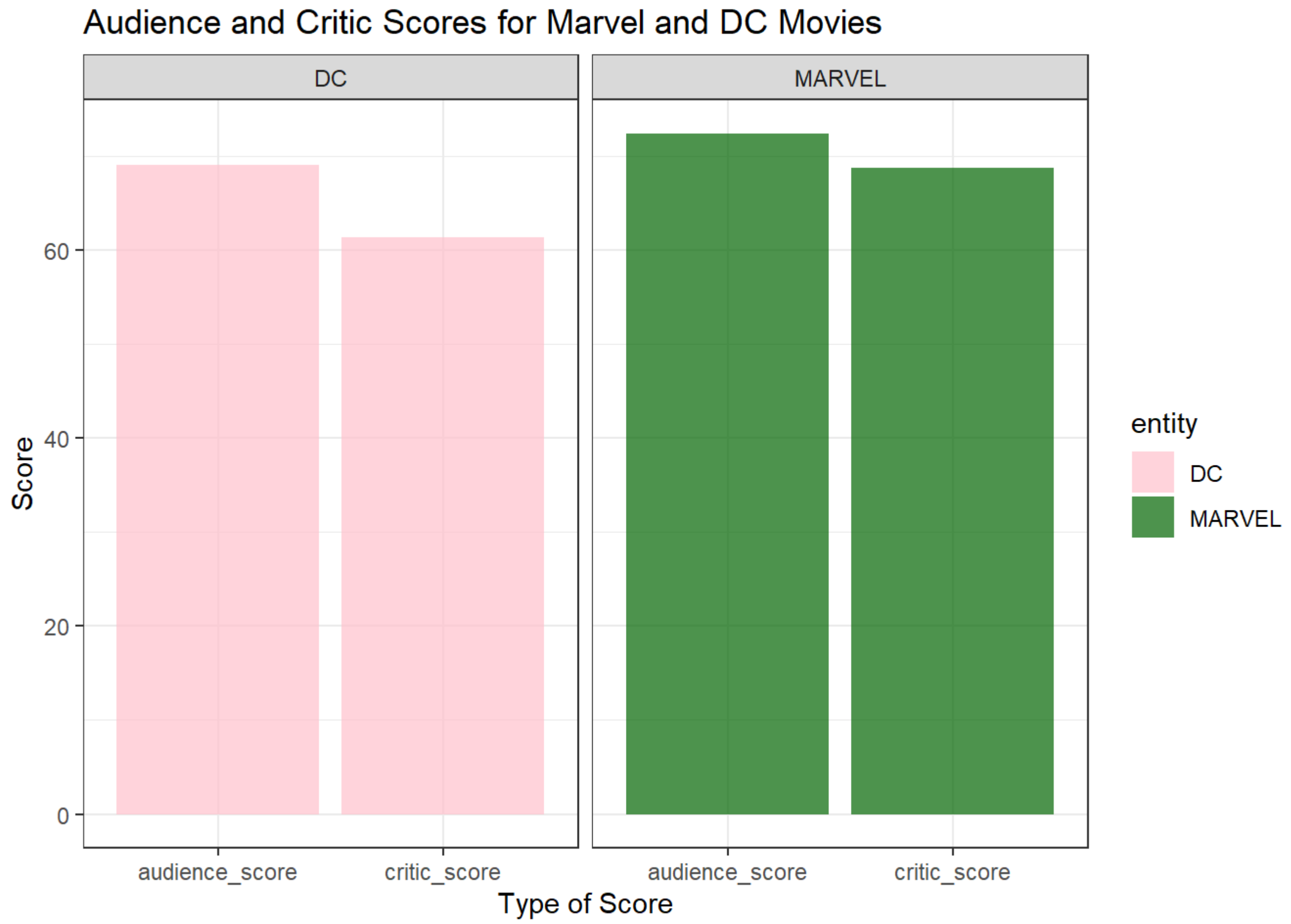
Next, we used `full_join` on critic and audience data set, we created a new data set that have data from both of the data set, which is the tomato score and its standard deviation of audience and critic. Then we used `kable` to make it into a table

Analyzing:

For our next numerical summary statistics, we looked at both the critic and audience Rotten Tomatoes ratings for Marvel and DC movies. When looking at the summary statistics alone, it can be seen that for both the audience and critic Rotten Tomatoes scores, Marvel movies tend to have a higher score on average, whereas DC movies tend to have higher standard deviations. These statistics match the relationship seen in the previous numerical summary statistics when we looked at IMDb ratings, suggesting that density plots for each should result in plots depicting similar trends and inferences. When specifically comparing these statistics between audience and critic ratings, it can be seen that, on average, audience members and the general public seem to rate movies higher than critics do as the average audience scores, for both DC and Marvel movies, is higher than the average critic scores. However, the standard deviations of critic scores tend to be higher, for both DC and Marvel movies, compared to the standard deviation of audience scores, suggesting that critics tend to be more strict and extreme in the scoring, whereas the audience tends to be more laid back and lenient in their scoring.

Visualization

```
#Adjust the tomato score summary statistics table so that we can see what type of score of each
 stats value is and what their score.
tomato_score<-tomato_score%>%
  pivot_longer(cols = c(audience_score, critic_score),
               names_to = "type_of_score",
               values_to = "Score")
#Create a side by side bar plot for Marvel and DC with x as type of score and y as score.
tomato_score%>%
  ggplot(aes(x = type_of_score, y =Score , fill = entity))+
  stat_identity(geom ="bar", alpha = 0.7)+
  facet_grid(~entity)+
  theme_bw()+
  scale_fill_manual(values = c("pink", "darkgreen")) +
  labs(x = "Type of Score", title = "Audience and Critic Scores for Marvel and DC Movies")
```



While the table of summary statistic for critic and normal audience was clear, we wanted to see the type of score and their score into 2 different column, so we used `pivot_longer` to make it into separate column. After that, we created a bar plot by using `stat_identity` function that geom layer is `bar` . We wanted to see the data side by side so that it’s easier to compare and contrast to we facet it by entity using `facet_grid` . We then proceeded to change the theme and scale by using `theme_bw` and `scale_fill_manual` , added the title and axis names using `labs` .

Analyzing:

In addition to our previous analysis, the produced bar plot clearly depicts these suggested trends as it can be seen that for both Marvel and DC movies, audience scores are always higher on average, and Marvel movie scores are always higher on average.

Third numerical summary stats

We are interested in the relationship between general rating and the number of Oscar’s nominations in each franchise.

```
#We create a new data set that has the number of Oscar's nomination, the IMDb rating/tomato audi
ence score's mean and standard deviation
third<-joined2%>%
  select(title, entity, nominated, num_nominations, imdb_rating, tom_aud_score)%>%
  group_by(entity, nominated)%>%
  summarise(n=n(),imdb_mean=mean(imdb_rating), tomato_mean= mean(tom_aud_score), imdb_sd =sd(imd
b_rating),tomato_sd = sd(tom_aud_score))%>%
  mutate(nominated = as.logical(nominated))
```

```
## `summarise()` has grouped output by 'entity'. You can override using the
## `.groups` argument.
```

```
#We then made it into a table
third%>%kable()
```

entity	nominated	n	imdb_mean	tomato_mean	imdb_sd	tomato_sd
DC	FALSE	15	6.320000	65.66667	1.3518242	22.170336
DC	TRUE	9	7.188889	74.55556	1.2343464	20.604072
MARVEL	FALSE	39	6.687180	68.07692	0.9325064	19.263252
MARVEL	TRUE	13	7.723077	85.23077	0.4399883	8.288082

To create the table of numerical stats for the relationship between the number of Oscar’s nominations and general rating for each franchise, we first used `select` to pick out the column that we want to used from the joined2 data set. Then we used `group_by` function to group the data by entity and nominated. After that, we started to calculate the mean and standard deviation by using `mean` and `sd` in `summarise` , we also count the total number of movies that was nominated/not nominated in each franchise here. Finally, we used `mutate` to turn value of nominated into a logical type of data instead of numerical and then used `kable` to make it into a table.

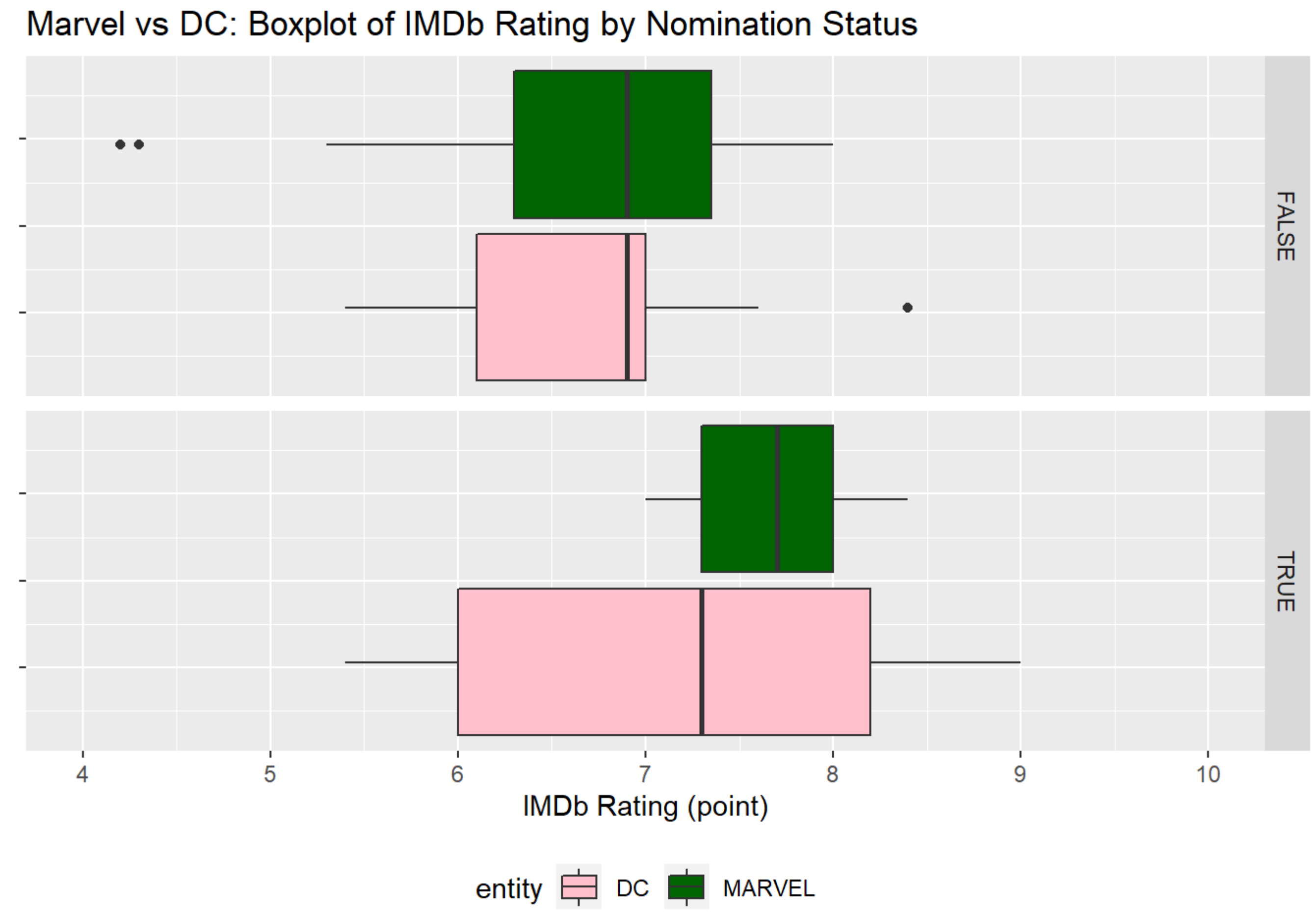
Analyzing: For our final numerical summary statistic, we went over the IMDb and Rotten Tomatoes ratings for each entity depending on whether it was nominated for an Oscar award or not. From the summary statistics, it can be seen that for each entity, the mean rating from both IMDb and Rotten Tomatoes was higher when a movie was nominated for an Oscar award by approximately 10%. This makes sense as movies that are nominated for an

Oscar in the first place tend to be good, and thus rated relatively high. It can also be seen that the standard deviations of movies for each entity is lower if they were nominated for an Oscar. This also makes sense as nominated films tend to all be relatively good and rated high so they should have ratings near the same range. Those that aren’t could still be really good or really bad, making its range a lot larger and explaining the larger standard deviation. Additionally, the means displayed show that the ratings given by the audience in Rotten Tomatoes tends to be higher than those from IMDb as they appear to be consistently higher throughout. For each nomination, it can be seen that Marvel movies also tend to do better, on average, than DC movies based on ratings. Marvel movies also have smaller standard deviations than DC movies here, consistent with the summary stats analyzed previously. Overall, these stats suggest that regardless of entity, movies that have been nominated for an Oscar will generally be better in terms of ratings.

Visualization:

```
#Created a boxplot facet by nomination status for IMDb rating for each franchise
joined2%>%
  mutate(nominated = as.logical(nominated))%>%
  ggplot(aes(x= imdb_rating, fill = entity))+
  geom_boxplot()+
  facet_grid(nominated~.)+
  theme_gray()+
  theme(legend.position="bottom", axis.text.y =element_blank()) +
  scale_fill_manual(values = c("pink", "darkgreen")) +
  scale_x_continuous(breaks = 4:10, limits = c(4, 10))+
  labs(x = "IMDb Rating (point)", title = "Marvel vs DC: Boxplot of IMDb Rating by Nomination St
atus")
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```



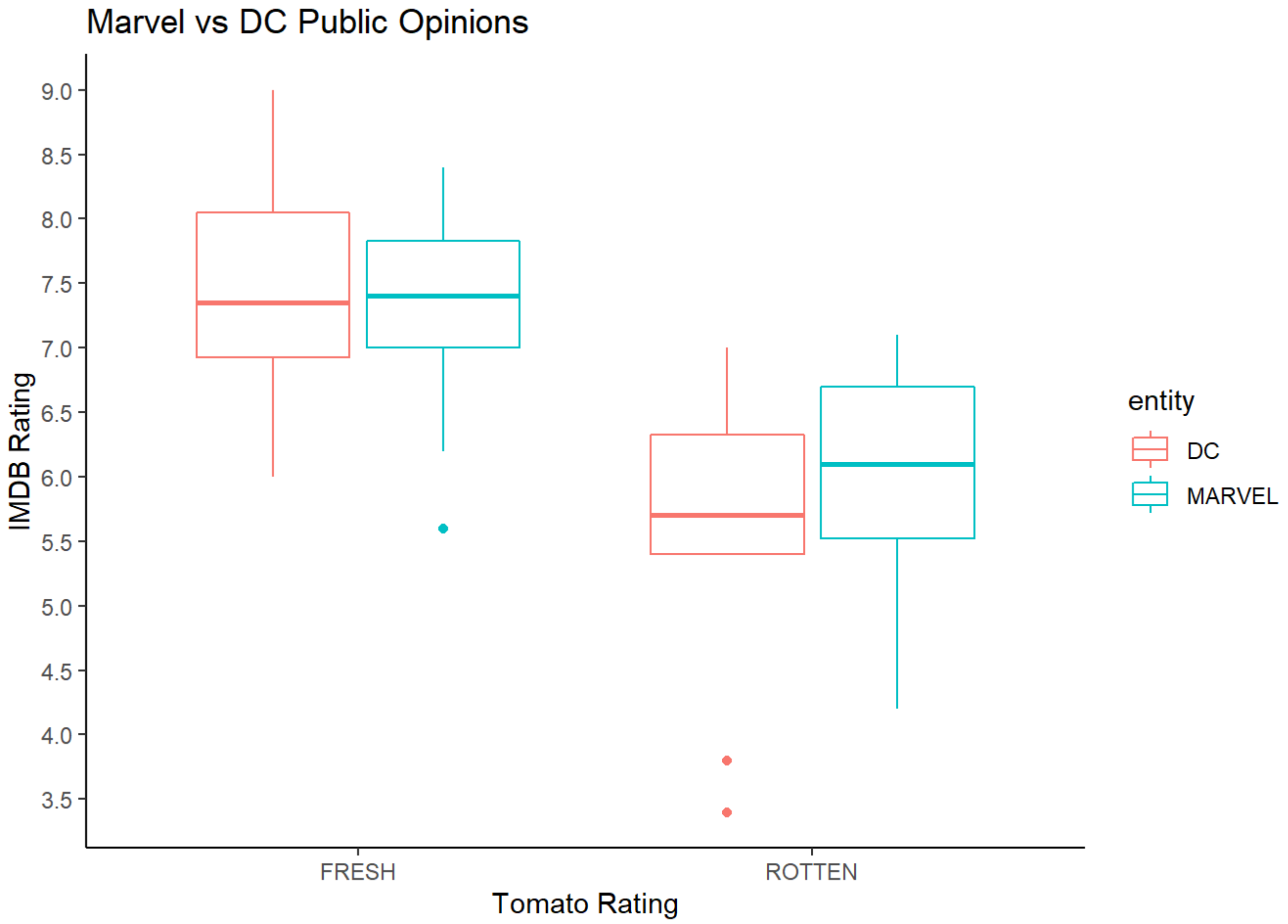
To create the visualization for what we want, we first used `mutate` to make the nominated variable into a logical variable instead of numerical type. We then proceeded to create a box plot using `ggplot` and `geom_boxplot` that has x as IMDb rating and then added layer `fill=entity` . After that, we faceted by nominated it by using `facet_grid` and adjust the graph. We adjusted the theme of the graph by `theme_gray` and `theme,` and scale the x axis with `scale_x_continuous` , we also changed the color using `scale_fill_manual` . Finally, we rename the title and axis by using `labs` function.

Analyzing: We created a box plot to represent IMDB Rating by Nomination Status for both Marvel and DC. The box plot shows the five number summary and its outliers for the IMDB rating and whether or not the entity was nominated. When observing the box plot for Marvel with no nominations the minimum IMDB rating is ~5.3 points, the median rating is ~6.8 points, and the maximum rating is ~8.0 points. Marvel has 2 outliers closer to a rating of 4.0 points whereas DC has an outlier of approximately 8.4 points. When looking at DC with no nominations, the minimum IMDB rating is ~5.4 points, the median rating is ~6.8 points, and the maximum rating is ~7.6 points. In contrast, for Marvel with nominations the minimum IMDB rating is 7.0 points, median rating is ~7.7 points, and maximum rating is ~8.4 points. For DC with nominations the minimum IMDB rating is ~5.4 points, median rating is ~7.4 points, and maximum rating is 9.0 points. Looking at the box plot overall, we can see that the higher the median IMDB score points is, the more likely the entity will get nominated.

5.Visualization

Marvel vs DC: Public Opinions.

```
#Reorganize the tomato data by entity and tomato rating, then using that to make a boxplot for tomato rating vs imdb rating that showing by entity. Then adjust name of each labs and change the layer and scale to make it look more aesthetic pleasing
tomato%>%
  group_by(entity,tomato_rating)%>%
  ggplot(aes(x= tomato_rating, y = imdb_rating, color = entity))+
  geom_boxplot()+
  labs(x = "Tomato Rating", y = "IMDB Rating",
 title ="Marvel vs DC Public Opinions")+
  theme_classic()+
  scale_y_continuous(breaks=seq(1, 10,by = 0.5))
```



We first used `group_by` function to group the tomato data by entity and tomato_rating. Then created a boxplot by using `ggplot` and `geom_boxplot` with x as tomato_rating, y as imdb rating and then add color layer by `color = entity`. After that, we changed the title and axis name using `labs` and adjust the theme with `theme_classic` while scaling the y axis using `scale_y_continuous`.

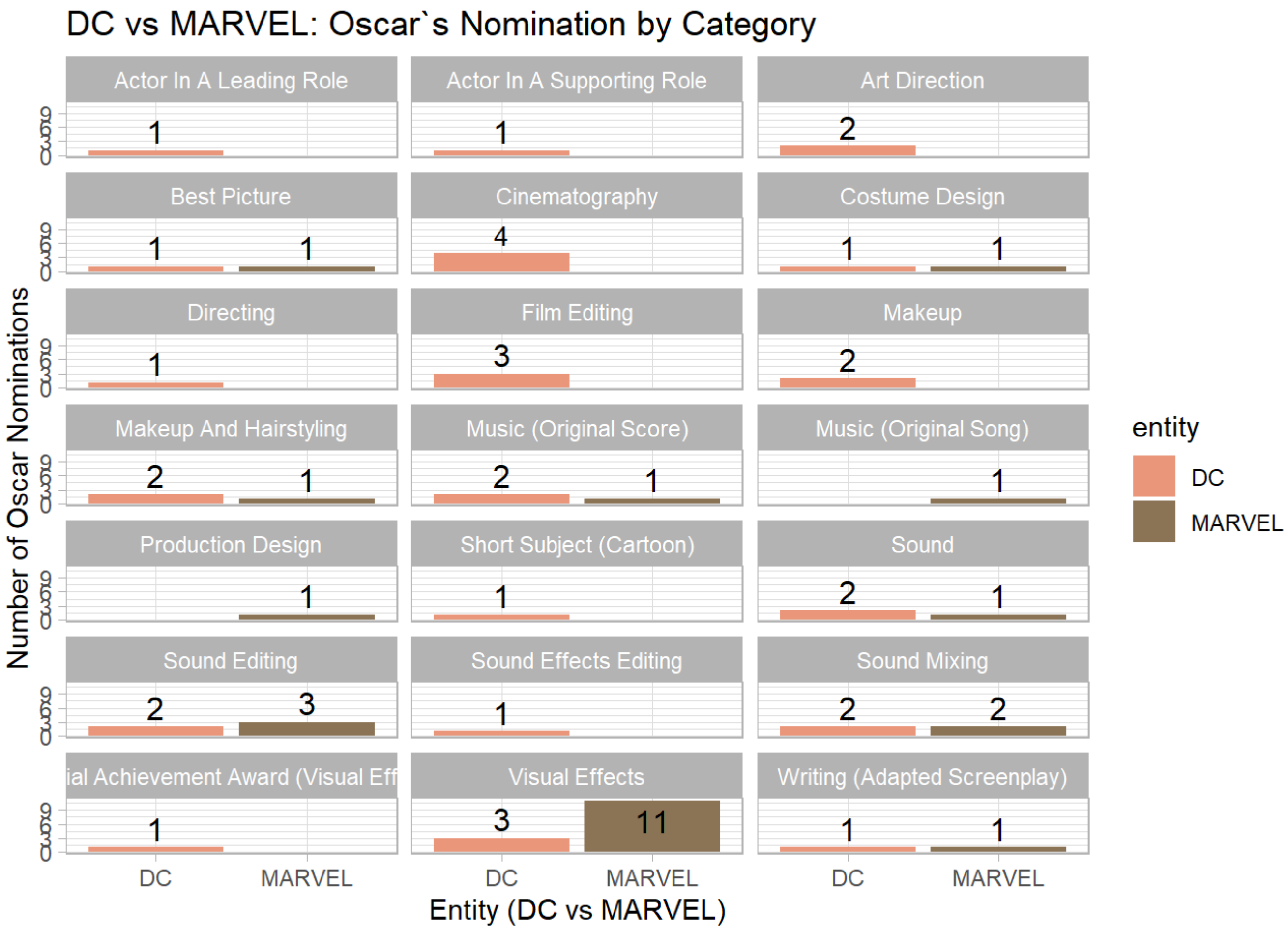
Analyzing:

We created a Marvel vs DC Public Opinions box plot. It shows the five number summary and its outliers for both DC and Marvel. We observe that the entity DC has a median rating of 7.3 for its “fresh” rating. Similarly, Marvel has a median rating of 7.3 for its “fresh” rating. DC has a maximum rating of ~8.9 and a minimum of 6.0. Whereas Marvel has a maximum of 8.4 and a minimum 6.3. Marvel has an outlier of a rating of 5.5 while DC does not have an outlier. When comparing the rotten scores, we can see that DC has a median rating of 5.8 whereas Marvel has a higher median rating of 6.3. The maximum for DC has a rating of 7.0 with 2 outliers of 4.0 and 3.5. In comparison, Marvel has a similar maximum rating score of 7.0 but a different minimum score of 4.3. Looking at the statistics, we can observe that DC and Marvel have similar public opinions ratings but DC has a slightly more negative public opinion compared to Marvel mainly due to the outliers. Other than that, there isn’t any significant difference between the public opinions.

Bar plot for each Category of Nomination for each Franchise

```
#We created a bar plot for each category of nomination for each franchise.
joined%>%
  group_by(entity,category)%>%
  summarise(num_nominate=n())%>%
  mutate(num_nominate=as.numeric(num_nominate))%>%
  filter(category !="NA")%>%
  ggplot(aes(x = entity, y = num_nominate, fill = entity))+
  geom_bar(stat = "identity")+
  scale_fill_manual(values = c("darksalmon", "burlywood4"))+
  facet_wrap(.~category, ncol = 3)+
  theme_light()+
  geom_bar_text()+
  labs(x = "Entity (DC vs MARVEL)",y = "Number of Oscar Nominations", title = "DC vs MARVEL: Osc
ar`s Nomination by Category")
```

```
## `summarise()` has grouped output by 'entity'. You can override using the
## `.groups` argument.
```



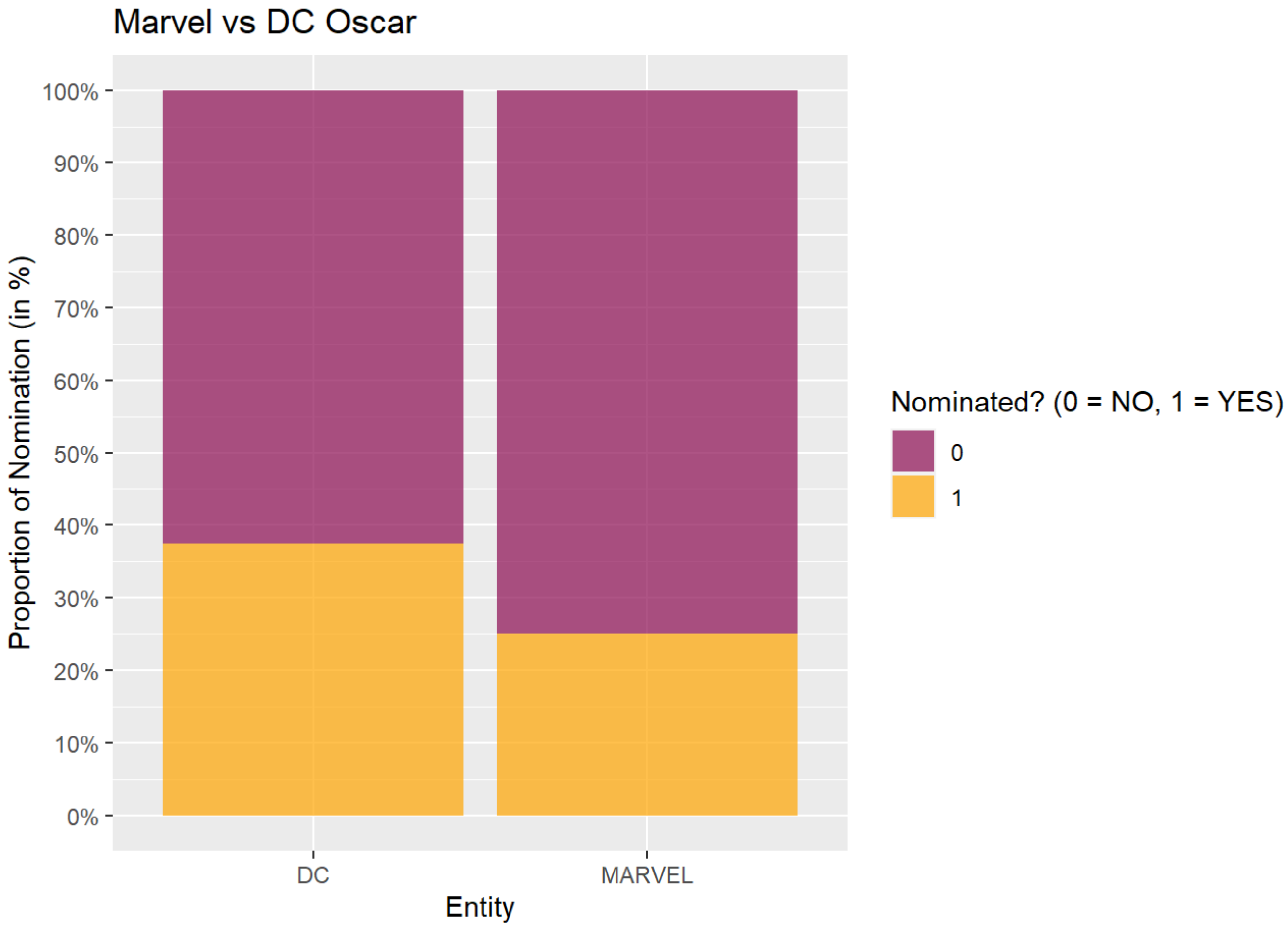
To create a bar plot of number of nomination for each category for each franchise, we first create a new data set by using `group_by` to group the joined data set by entity and category, then used `summarise` to find the number of nomination for each movie and named it “num_nominate.” We then used `mutate` to make num_nominate become a numerical type of data. After that, we used `filter` to filter out the category that has “NA” as name of value.

We then started to create our plot. We used `ggplot` and `geom_bar` to create a bar plot with x axis as entity, and y as the number of nomination. We added layer `fill = entity` then faceted it by category using `facet_wrap`. After that, we adjusted the graph using `scale_fill_manual` and `theme_light` to change the theme and scale color, and using `geom_bar_text` to show value of each category for each franchise. Finally we changed the title and axis using `labs` function.

Analyzing: We now looked at the types and number of Oscar nominations received by Marvel and DC films. The produced plot depicts the number of Oscar nominations for each entity, but split up into multiple plots by the type of Oscar nomination. When looking at these plots, it can be seen that besides Visual Effects, Sound Editing, Production Design, and Music, DC got at least as many awards compared to Marvel in all of the other categories. When adding up the number of nominations, it can be seen that DC actually has more Oscar nominations than Marvel by a decent amount, which lines up with our expectations. While looking at the previous plots may make a plot like this a little surprising, it could also be seen that DC movies had a higher standard deviation due to more DC movies having a rating of 10 than Marvel movies. This small detail is likely the reason why DC has more Oscar nominations as those movies could have been so good that they received multiple Oscar awards for them. This can be seen when looking at the data set itself as Joker, a DC movie, received 11 nominations. All of this suggests that while Marvel may produce better movies, in terms of ratings, on average, DC has produced some of the best movies, even better than Marvel, individually.

We now look at the proportion of nominated status for each franchise.

```
#We created a proportion barplot by percentage of nomination status for each franchise.
joined2%>%
  group_by(entity,nominated)%>%
  ggplot(aes(x= entity, fill = as.character(nominated)))+
  geom_bar(position = "fill", alpha = 0.7)+
  scale_fill_manual(values = c("deeppink4", "orange"))+
  labs(x = "Entity",
       y = "Proportion of Nomination (in %)",
       title ="Marvel vs DC Oscar",
       fill="Nominated? (0 = NO, 1 = YES)")+
  theme_gray()+
  scale_y_continuous(labels =scales::percent_format(accuracy = 1), breaks=seq(0, 1,by = 0.1))
```



We used `group_by` to group the variables “entity” and “nominated”. We then created a barplot using `ggplot` and `eom_bar` with x as “entity” and layer it with `fill= as.character(nominated)`. We customized the bar plot using `scale_fill_manual` and `theme_gray`. Then we used `labs` to change the title and axis name, and added fill layer so it would display whether the entity was nominated or not with 1 being “yes” and 0 being “no.” We then used `scale_y_continuous` to change the label of y-axis into percentage.

Based on the bar plot of nominated proportion, we can see that almost 40% of DC movies was nominated for Oscar’s Award while only 25% of Marvel was nominated. So we can say that in term of academy aspect, DC are doing better than Marvel.

Contribution:

Doan Nguyen: Worked on the code, tidying data, commentary and organized the report.

Ethan Chang: Worked on the code, analyzing and commentary on the tables and some visualizations.

Natleigh Burns: Found the data sets, did the introduction, helped with writing the commentary and the code, and organizing the presentation.

Rhean Palencia: Worked on the analyzing for visualizations and commentary and helped with the code.

Reference

1. Fontes, R. (2020). The Oscar Award, 1927 - 2020. Retrieved October 2022, from https://www.kaggle.com/datasets/unanimad/the-oscar-award/code. (https://www.kaggle.com/datasets/unanimad/the-oscar-award/code.)

2. Kraggle. (2021). MARVEL vs. DC - IMDB & ROTTEN TOMATOES. Retrieved October 2022, from https://www.kaggle.com/datasets/jcraggy/marvel-vs-dc-imdb-rotten-tomatoes (https://www.kaggle.com/datasets/jcraggy/marvel-vs-dc-imdb-rotten-tomatoes).