Do POC lead actors impact gross movie revenue?

Team 9 - Abhishek Baral, Joe Hsieh, Joao Mansur

Abstract for Class

In the 70s Bruce Lee auditioned to be in the show Kung Fu. Set in the American Wild West, the show would feature a Shaolin monk escaping to the US to get caught in several disputes. The part was eventually awarded to David Carradine and rumors exist that Bruce Lee was passed on because “minorities can’t carry lead roles.” This report seeks to give quantitative evidence to a *casting director at a Hollywood movie studio* to provide evidence as to whether minority (non-white) leads in movies cause changes in box office revenue. Rather than try to vindicate Bruce Lee using historical data, we will use modern data to see whether a modern movie can benefit or lose from casting a minority lead. The main method used is matching minority- and white-led movies with multiple methods and showing causality through linear regression. If making a report to an actual casting director, we would exclude code but we are keeping it here for class.

Executive Summary

Casting minority actors for lead roles in movies is beneficial for movies in the Action genre and does not cause any statistically significant loss or gain in other genres. This report proves this by scraping movie data from the IMDB and boxofficemojo.com, pruning the data by matching similar movies with POC and non-POC lead actors, then running a linear regression to find any evidence of causality. By looking at movies with similar release dates, genres, and more it was possible to isolate the effect of a lead actor’s POC status. In all cases, an actor belonging to a racial minority is not a disadvantage to a movie’s revenue and is proven to be a significant positive in Action movies.

Scraping

Our data for this report was gathered by a process called Scraping.

Web scraping or web harvesting, is the process of using a bot or web crawler to extract content and data from a website, rather than manually copying and pasting results. Web scraping extracts the HTML code and the data stored in a database, and then can recreate the website content elsewhere. Essentially, data from websites is automatically and precedurally extracted according to specification.

For our purposes, we decided to scrape 2 websites, box office mojo and IMDB. For box office mojo, we collected four years’ worth of top domestic box office films in the United States from 2017 to 2020. The relevant data that we searched for was title, gross sales, release date, revenue, and lead actor among other variables. For IMDB, we had a similar approach, but focused on ratings and movie distributor. Lastly for the race data, we were able to find lists on IMDB of actors and their ethnicity

Once the three data sets were sourced, some level of processing was needed to clean the variables in the dataset and re-casted to an appropriate data type. Lastly, we then merged the box office mojo and imdb datasets by movie title, and then merged that newly created set with the race data set by lead actor name.

The full data results from this step is available on github at

To make sure results were accurate, we checked and cleaned the data manually. Removing any incomplete or repeating data then checking whether actor-race matches were accurate. The final dataset is then sent to analysis below.

Initial Analysis

A linear regression can be an effective way to estimate the effect of a certain feature, like whether the lead actor is of a minority race or not. We can isolate the effect it can have on an outcome like a movie's gross revenue by using regression and accounting for other variables to make sure we isolate the effect. The way this happens is that through statistics, a linear regression can assign weights to variables to try to predict a movie's gross revenue the best it can. In these regressions, it is also possible to mathematically calculate whether we are sure an effect exists, a process summarized by the displayed p-score. If a p-score is above 0.05, we say that we aren't sure if the effect is statistically significant. If something is statistically significant, we are assuming that the resulting coefficient is not a random occurrence but can be attributed to a cause.

In other words, if we want to see if a racial minority lead actor affects gross revenue, then we can check whether the variable that accounts for POC status has a p-score of less than 0.05.

Let's begin by running a regression on the data we have scraped and cleaned. Firstly let's only use the Race variable:

```
summary(lm(gross~Race,data=data))
```

```
##
## Call:
## lm(formula = gross ~ Race, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -70606613 -54820638 -34249138   8923043 628442568
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 59833642    4887367  12.243   <2e-16 ***
## RacePOC     11783356   10561477   1.116    0.265
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 96390000 on 493 degrees of freedom
## Multiple R-squared:  0.002519,   Adjusted R-squared:  0.0004952
## F-statistic: 1.245 on 1 and 493 DF,  p-value: 0.2651
```

The print0out above shows that the regression thinks that an actor being a POC, despite a negative coefficient, is not statistically significant (p-score [Pr(>|t|)] above 0.05). The model considers a baseline (Intercept) then adds the coefficients to it; this makes the baseline a non-POC lead and the RacePOC coefficient being the calculated difference between them. Since the effect is not mathematically proven to not be random, thus having a low p-score, it seems that an actor being a POC has no significant effect if considered as the only variable in a regression.

However, we must also consider the many other variables we have, adding other variables could clarify the effect and isolate it further. Let's try a new regression with a few more variables.

```
summary(lm(gross~Race+year+genre +Male+opening_theaters,data=data))
```

```
##
## Call:
## lm(formula = gross ~ Race + year + genre + Male + opening_theaters,
##     data = data)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -131643531  -27851445   -8209102   13531848  552922944
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.035e+10  9.653e+09   1.072  0.28427
## RacePOC            9.910e+06  8.428e+06   1.176  0.24022
## year              -5.114e+06  4.784e+06  -1.069  0.28561
## genreAdventure    -8.386e+07  1.096e+07  -7.650 1.09e-13 ***
## genreBiography    -9.385e+07  1.567e+07  -5.988 4.13e-09 ***
## genreComedy       -8.295e+07  9.890e+06  -8.388 5.45e-16 ***
## genreDocumentary -8.115e+07  2.692e+07  -3.014  0.00271 **
## genreDrama        -8.891e+07  1.210e+07  -7.346 8.73e-13 ***
## Male               1.232e+07  7.341e+06   1.679  0.09383 .
## opening_theaters  2.375e+04  3.951e+03   6.009 3.66e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 76840000 on 485 degrees of freedom
## Multiple R-squared:  0.3765, Adjusted R-squared:  0.3649
## F-statistic: 32.54 on 9 and 485 DF,  p-value: < 2.2e-16
```

What we’re looking at here is a baseline or intercept of White female lead actors in Action movies in the year 2017 with no opening theaters (an illogical but mathematically fine assumption since we correct it later). The coefficients modify the expected gross revenue of a movie depending on their characteristics. What we see here is that there is a clearly significant difference between Action movies and other genres, where Action movies tend to make more money. This is also true for male-led movies, where we see a larger gross revenue effect. We also included the number of opening theaters for the movie, which should relate to how large the movie’s production is, and we see that movies launched on more theaters make more revenue.

In the end though, whether the lead is a POC or what year it was made does not seem to show any statistically significant difference.

However, the data isn’t well-balanced between POC and non-POC lead actors which we can see below:

```
table(data$Race)
```

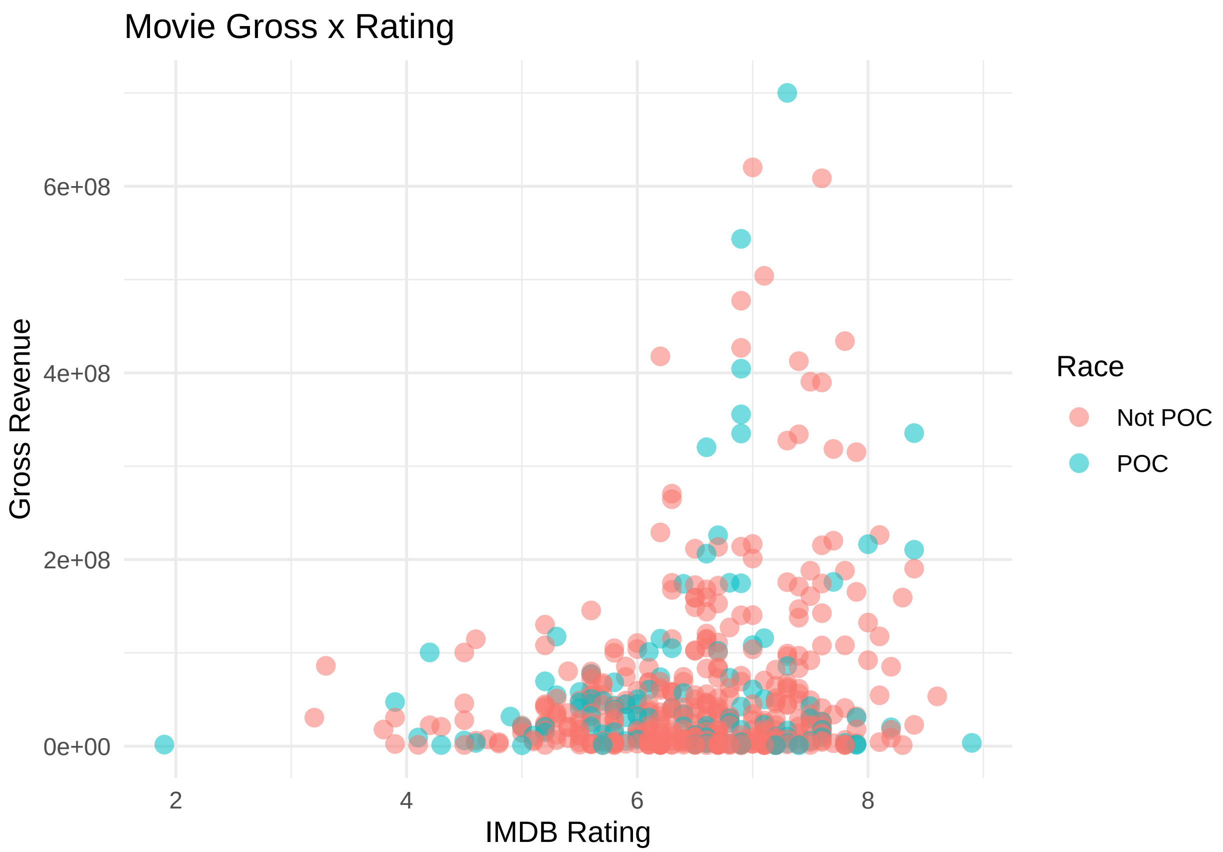
```
##
## Not POC     POC
##     389     106
```

POC-led movies are only 106 movies or around 21.4% of the data. If we plot these movies using their IMDB Ratings and their Gross Revenue, we can see that non-POC movies occupy more space.

```
library(ggplot2)
```

```
## Warning: package ’ggplot2’ was built under R version 4.0.4
```

```
p1<-ggplot(data, aes(x=imdb_rating, y=gross, color=Race))+geom_point(size=2.8,alpha=.55)+ labs(title=("
p1
```



It seems like a lot of the data is mixed and there isn't a clear separation between POC and not POC-led movies. If we prune the data to include movies that are similar to each other but only distinguishable by the race of their lead actor, we can more closely isolate the effect of a lead actor being a minority has on gross revenue.

With the data in mind, we can begin to prepare our matching strategy.

Matching

To perform matching, we must determine how movies are to be matched. We have a couple of obvious choices: the year, movie genre, film rating, and whether the lead is male. These variables are all fixed categories (they are not fractional and a movie can't occupy two at the same time). This means that we can match films based on whether they meet all criteria the same way except for whether their lead is a POC. This kind of matching is called exact matching, and it allows for the creation of groups of similar movies and the removal of ones that don't fit with another.

Some data is more granular and difficult to match exactly, like IMDB ratings that are fractional and the amount of theaters a movie opened in. These will require some strategy to allow exact matches to occur, a practice we call binning.

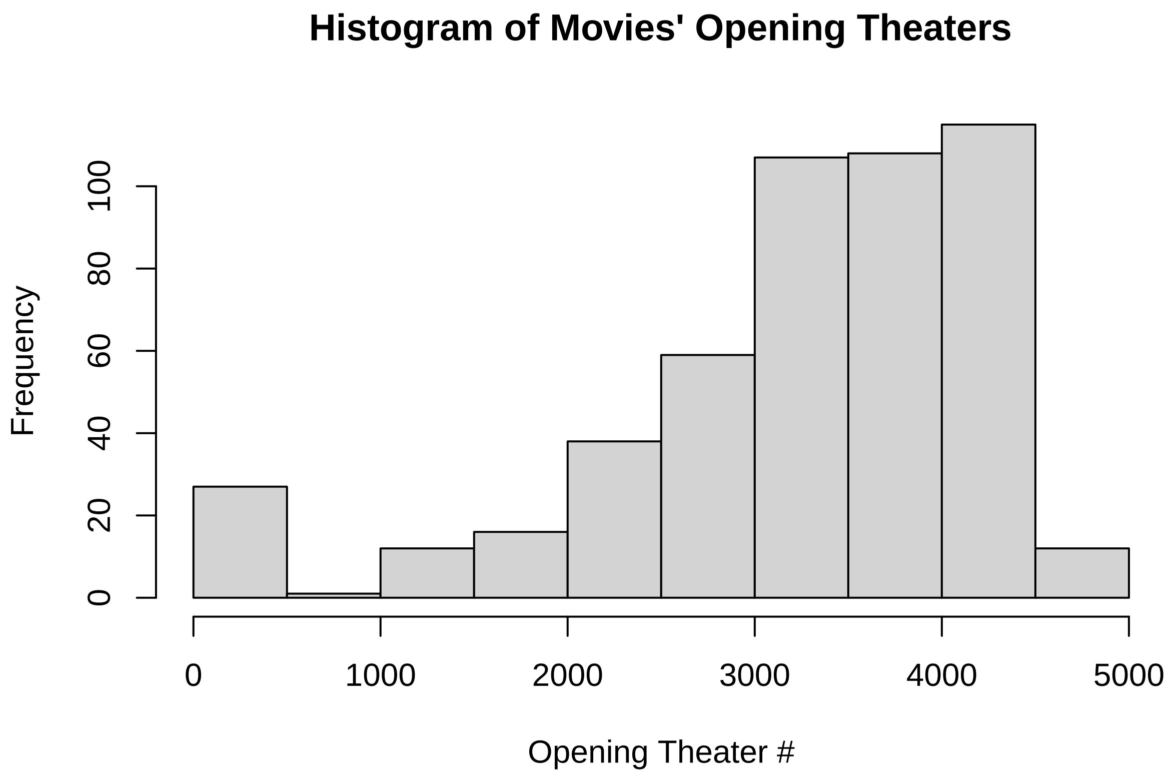
Opening Theaters

How many theaters the movies opened in is a number with a lot of variability and finding exact matches would be difficult.

However, the number of theaters a movie opened in can be really indicative of the amount of promotion that went into the movie. More indie films are clearly released in less theaters than blockbusters.

To use this feature, which we believe to be important as mentioned, we need to bin them by separating movies into groups. A histogram can help us find these bins:

```
hist(data$opening_theaters,xlab="Opening Theater #",main = paste("Histogram of Movies' Opening Theaters
```



Using a histogram we can see that we can group movies by their opening theaters. Here, the choice for the size of the bin matters where there is a trade-off between how informative a bin is and how easy it will be to find matches. We decided to use movies with under 2000, 2500, 3000 then so on in 250-sized increments. These values allow for at least 40 or so movies in each bin which allows for good matching chances without losing the predictive power of the variable.

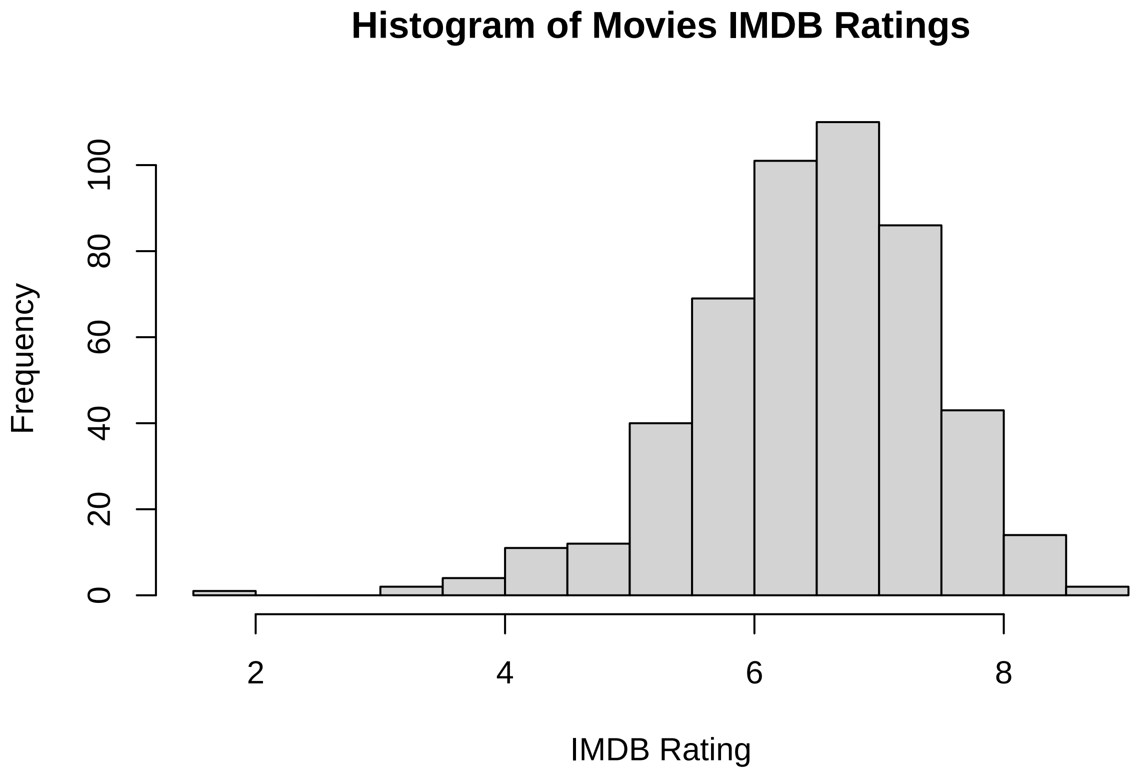
```
data$OPT<-"<1000"
data[data$opening_theaters>=4250,"OPT"]<-">4250"
for (y in c(4250,4000,3750,3500,3250,3000,2500,2000)) {
  data[data$opening_theaters<y,"OPT"]<-paste("<",y)
}

table(data$OPT)
```

```
##
## < 2000 < 2500 < 3000 < 3250 < 3500 < 3750 < 4000 < 4250  >4250
##     56     38     59     61     46     62     44     73     56
```

IMDB Ratings

We can do the same for their IMDB ratings. These ratings hopefully will account for the quality of the movie enough to be a proxy for a movie's long-term box office success.

```
hist((data$imdb_rating),xlab="IMDB Rating",main = paste("Histogram of Movies IMDB Ratings"))
```



IMDB ratings seem very close to a normal distribution around 6.5 or so. This means that if we round to whole numbers, most values will be 6 or 7 and the rest will be in tails of 8+ and less than 5. To make effective use of these, we can separate bins within one standard deviation from those beyond, thus achieving bins related to bad, medium, and great movies. One standard deviation in the IMDB Ratings is .94, meaning that movies within around 5.5 and 7.5 are within a standard deviation. To bin these movies, we can round their IMDB ratings to the enarest whole number, bin 6's and 7's, and then bin those above and beyond this range.

```
data$IMDB<-0
for (y in seq(1,10,.5)) {
  data[round(data$imdb_rating)==y,"IMDB"]<-y
}

data$IMDBbin<-"5-"
data[round(data$imdb_rating)==6,"IMDBbin"]<-"6/7"
data[round(data$imdb_rating)==7,"IMDBbin"]<-"6/7"
data[round(data$imdb_rating)>7,"IMDBbin"]<-"8+"
table(data$IMDBbin)
```

```
##
##  5- 6/7  8+
##  59 363  73
```

Using these four simple bins should be enough to isolate the effect of extraordinarily good and bad movies from the middle of the pack. 8+ movies are one standard deviation higher in rating and 5- are one lower.

Exact Matching

We can now finally start matching movies, where we can create subclasses of movies that are in all the same bins but with a difference in whether their lead actor is a POC.

To do this, we use an R Package called MatchIt. There are several ways to match data, but the method we have been referring to is called Exact matching. If there is even a slight difference then a match won't be found, this is what made binning the variables so important. MatchIt creates subclasses that contain movies with the same binned variables and drops any that don't have both movies with POC and movies with non-POC lead actors. This means that we are not matching in pairs, but rather creating groups of movies and removing any outliers.

Let's run MatchIt and see how our data set changes.

```
library(MatchIt)
data$treat<-0
data[data$Race=="POC","treat"]<-1
m.out <- matchit(treat ~ year+genre+OPT+film_rating+Male+IMDBbin,
                 data = data, method = "exact",verbose=TRUE)
```

```
## Exact matching...
## Calculating matching weights... Done.
```

```
m.data <- match.data(m.out)
```

```
nrow(data)
```

```
## [1] 495
```

```
nrow(m.data)
```

```
## [1] 126
```

```
table(m.data$Race)
```

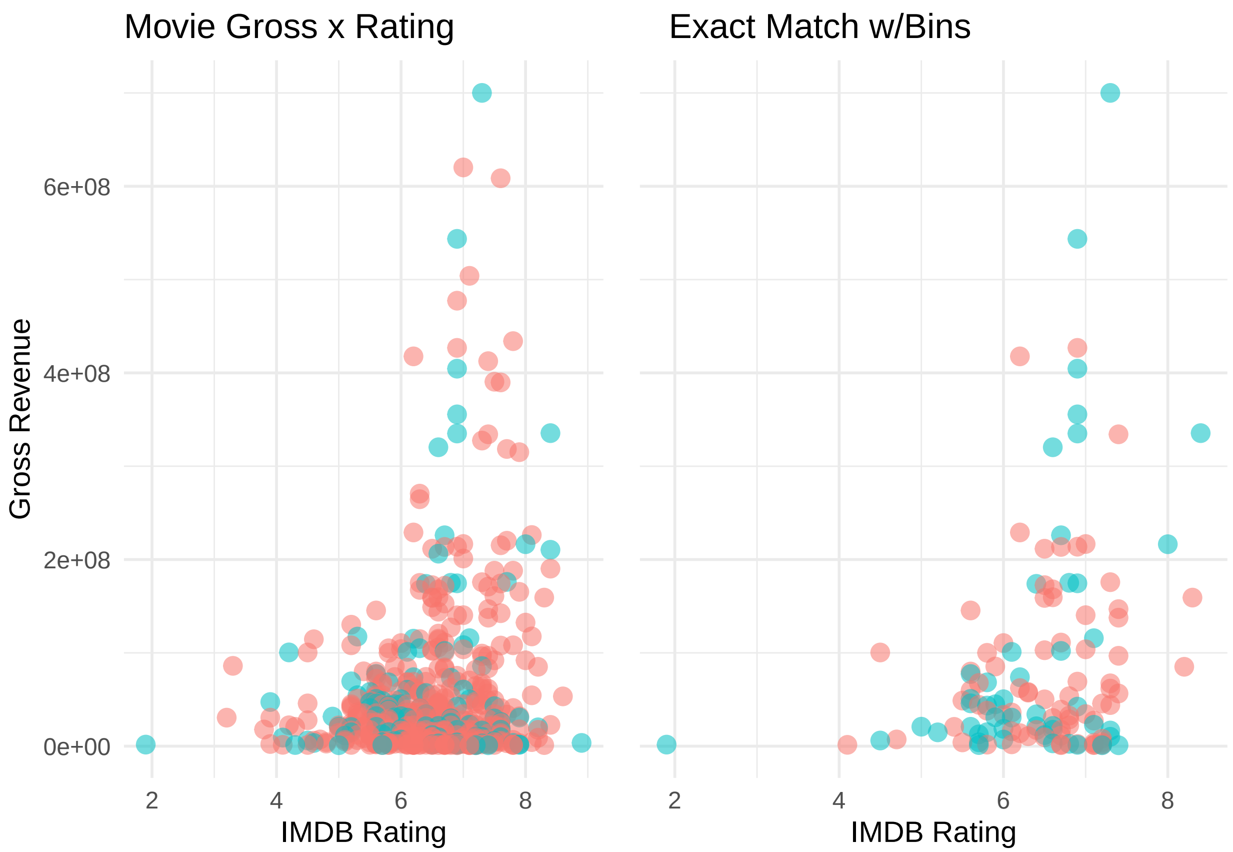
```
##
## Not POC     POC
##      76      50
```

Our original data had 495 movies pre-match, after matching we have 126 movies. While POC-led movies are still around 39.6% of the data, we have eliminated movies that don't share similarities and potentially increased the visibility of the effect we are looking for. We can visualize the dataset before and after matching in the same graph:

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.0.4
```

```
p2<-ggplot(m.data, aes(x=imdb_rating, y=gross, color=Race))+geom_point(size=2.8,alpha=.55)+ labs(title=
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank())
grid.arrange(p1+ theme(legend.position="none"),p2+ theme(legend.position="none"),ncol=2)
```



It is clear how the data set is pruned and more balanced between the turquoise POC data and the reddish non POC data. We obviously did not balance on Gross Revenue, which is our target variable to predict, but we have maintained the normal distribution of IMDB ratings which is evident by the same overall shape of both graphs.

We can now once again run our regression but this time with matched data. Note that while we matched using bins, we can go back to using un-binned variables in our regression. We will also utilize the weights from our matching. These weights are meant to give power to certain movie data that is representative of the overall population. This method means that although we pruned a lot of movies, certain movies can be given additional weight so that they represent information from movies that were lost.

With that, let's run the regression:

```
summary(lm(gross~Race+year+genre+opening_theaters+film_rating+Male+imdb_rating,data=m.data,weights=weig
```

```
##
## Call:
## lm(formula = gross ~ Race + year + genre + opening_theaters +
##     film_rating + Male + imdb_rating, data = m.data, weights = weights)
##
## Weighted Residuals:
##        Min         1Q     Median         3Q        Max
## -151423854  -41449521   -4320511   20036299  501080189
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.874e+10  2.494e+10  -0.751  0.45396
## RacePOC           3.821e+07  1.597e+07   2.392  0.01842 *
## year              9.210e+06  1.235e+07   0.745  0.45756
## genreAdventure   -4.595e+07  3.247e+07  -1.415  0.15972
## genreBiography   -6.812e+07  6.357e+07  -1.071  0.28628
## genreComedy      -7.323e+07  2.398e+07  -3.053  0.00283 **
## genreDocumentary -7.332e+07  8.182e+07  -0.896  0.37212
## genreDrama       -6.083e+07  3.379e+07  -1.800  0.07448 .
## opening_theaters  3.056e+04  1.191e+04   2.565  0.01163 *
## film_ratingPG     1.791e+07  5.822e+07   0.308  0.75896
## film_ratingPG-13 -1.410e+07  4.733e+07  -0.298  0.76640
## film_ratingR     -3.780e+07  4.779e+07  -0.791  0.43064
## Male              2.719e+07  2.017e+07   1.348  0.18028
## imdb_rating       2.409e+07  9.645e+06   2.498  0.01395 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 87260000 on 112 degrees of freedom
## Multiple R-squared:  0.4718, Adjusted R-squared:  0.4105
## F-statistic: 7.695 on 13 and 112 DF,  p-value: 1.049e-10
```

After matching, we now see that a movie’s lead actor being a POC has not only a statistically significant effect, but a positive one. As a reminder, our baseline is the gross revenue expected from White female lead actors in Action movies in the year 2017 with no opening theaters. This means that our matched data, which keeps movies that are more standard and similar to the usual release according to our data, shows that POC-led movies tend to make more money.

However, we can still create another variable to gain more insight into movies led by POC. Since a lot of movie genres are deeply impacted by race and culture, it could be a good idea to check what happens if we allow the regression to work with the interaction between race and genre. What this means is that the regression will create a variable for movies that at the same time are led by a POC actor and belong to a particular genre. These interaction terms will allow us to see the effect of having a POC lead on movies of each genre.

That regression will come to this result:

```
summary(lm(gross~Race+year+genre+genre:Race+opening_theaters+film_rating+Male+imdb_rating,data=m.data,w
```

```
##
## Call:
## lm(formula = gross ~ Race + year + genre + genre:Race + opening_theaters +
##     film_rating + Male + imdb_rating, data = m.data, weights = weights)
##
## Weighted Residuals:
##        Min         1Q     Median         3Q        Max
## -175406104  -28357173   -8078792   18715410  478171509
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1.942e+10  2.485e+10  -0.781  0.43637
## RacePOC                  7.874e+07  2.335e+07   3.372  0.00104 **
## year                     9.542e+06  1.231e+07   0.775  0.44011
## genreAdventure          -1.557e+07  3.809e+07  -0.409  0.68359
## genreBiography          -3.463e+07  7.886e+07  -0.439  0.66140
## genreComedy             -4.408e+07  2.977e+07  -1.481  0.14163
## genreDocumentary         -5.013e+07  9.119e+07  -0.550  0.58368
## genreDrama               -2.602e+07  3.846e+07  -0.677  0.50012
## opening_theaters          3.167e+04  1.209e+04   2.620  0.01008 *
## film_ratingPG             1.981e+07  5.805e+07   0.341  0.73361
## film_ratingPG-13         -1.234e+07  4.720e+07  -0.261  0.79426
## film_ratingR             -3.567e+07  4.766e+07  -0.749  0.45579
## Male                      2.817e+07  2.011e+07   1.401  0.16417
## imdb_rating               2.188e+07  9.734e+06   2.247  0.02667 *
## RacePOC:genreAdventure   -7.279e+07  5.136e+07  -1.417  0.15934
## RacePOC:genreBiography   -8.040e+07  1.147e+08  -0.701  0.48483
## RacePOC:genreComedy      -7.201e+07  4.446e+07  -1.620  0.10823
## RacePOC:genreDocumentary -4.266e+07  1.160e+08  -0.368  0.71369
## RacePOC:genreDrama       -8.386e+07  4.253e+07  -1.972  0.05120 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 86950000 on 107 degrees of freedom
## Multiple R-squared:  0.4989, Adjusted R-squared:  0.4146
## F-statistic: 5.918 on 18 and 107 DF,  p-value: 1.25e-09
```

The regression printout above now isolates the effect of POC actors in each genre, with the original RacePOC being the effect on Action movies. We still see a positive effect from POC lead actors but now with an even larger estimate and better p-value (thus more statistically significant). This is unsurprising given the success of action movies like Black Panther (Rest in peace King Boseman), Joker (Joaquin Phoenix of Puerto Rican descent) and others led by Dwayne “The Rock” Johnson, Jason Momoa, and Will Smith. However, in other genres we see that the effect of POC-leads is not statistically significant.

We can safely assume that modern Action movies can benefit from minority leads while in other genres there is no proof that minority leads have an effect on box office performance. Either way, there is no evidence that minority actors can’t carry movies as lead actors; if anything, they might be a great choice for action movies right now.

Optimal Matching

Though we decided to use Exact matching, we can test this data set by using another form of matching. The MatchIt package also offers optimal matching, which uses Mahalonobis Distances to find similar movies. These distance metrics allow for non-exact differences by mathematically finding the distance between two movies’ characteristics, and matching those closest together but different by whether the lead is a POC. MatchIt’s optimal matching system solves for pairs of points while minimizing the total distance between them, thus finding the optimal and most similar movie pairs. Exact matching, on the other hand, created subsets instead of pairs and finding pairs can create better comparisons for our target effect.

However, the fact that it minimizes the distance between points means that it relies on data a lot more. While in binning for exact matches we grouped movies following a logic (bad,medium, good movies and binning opening theater numbers to find approximations of promotion size), optimal matching minimizes distance between the metrics making small differences in features more important. This method makes certain matches viable, like matches at the end of one bin and the beginning of another, but it also overvalues relationships within bins where it may pair movies with extremely similar ratings though ratings are subjective and not informative past a certain threshold. The idea of using distances therefore has issues because it assumes that the distances between points is relevant, something we believed to not be true to this dataset.

By using exact matching, we utilized non-categorical data in a way that we tried to best preserve their utility with domain knowledge. Optimal matching, however, can produce similar or different results but it will unarguably provide pairs that are mathematically similar. Though our conclusion will rely mostly on exact matching, we believe optimal matching can provide an interesting take and likely the same overall result.

Let's try running optimal matching to see what happens.

```
#Note: optimal matching naturally attempts 1:1 matching but can be modified to do 1:n

m.out2 <- matchit(treat ~ year+genre+opening_theaters+film_rating+Male+imdb_rating,
                  data = data, method = "optimal",verbose=TRUE)
```

```
## Optimal matching...
## Calculating matching weights... Done.
```

```
m.data2 <- match.data(m.out2)
```

```
#original
paste("Original Data rows:",nrow(data))
```

```
## [1] "Original Data rows: 495"
```

```
paste("Exact-matched Data rows:",nrow(m.data))
```

```
## [1] "Exact-matched Data rows: 126"
```

```
paste("Optimal-matched Data rows:",nrow(m.data2))
```

```
## [1] "Optimal-matched Data rows: 212"
```

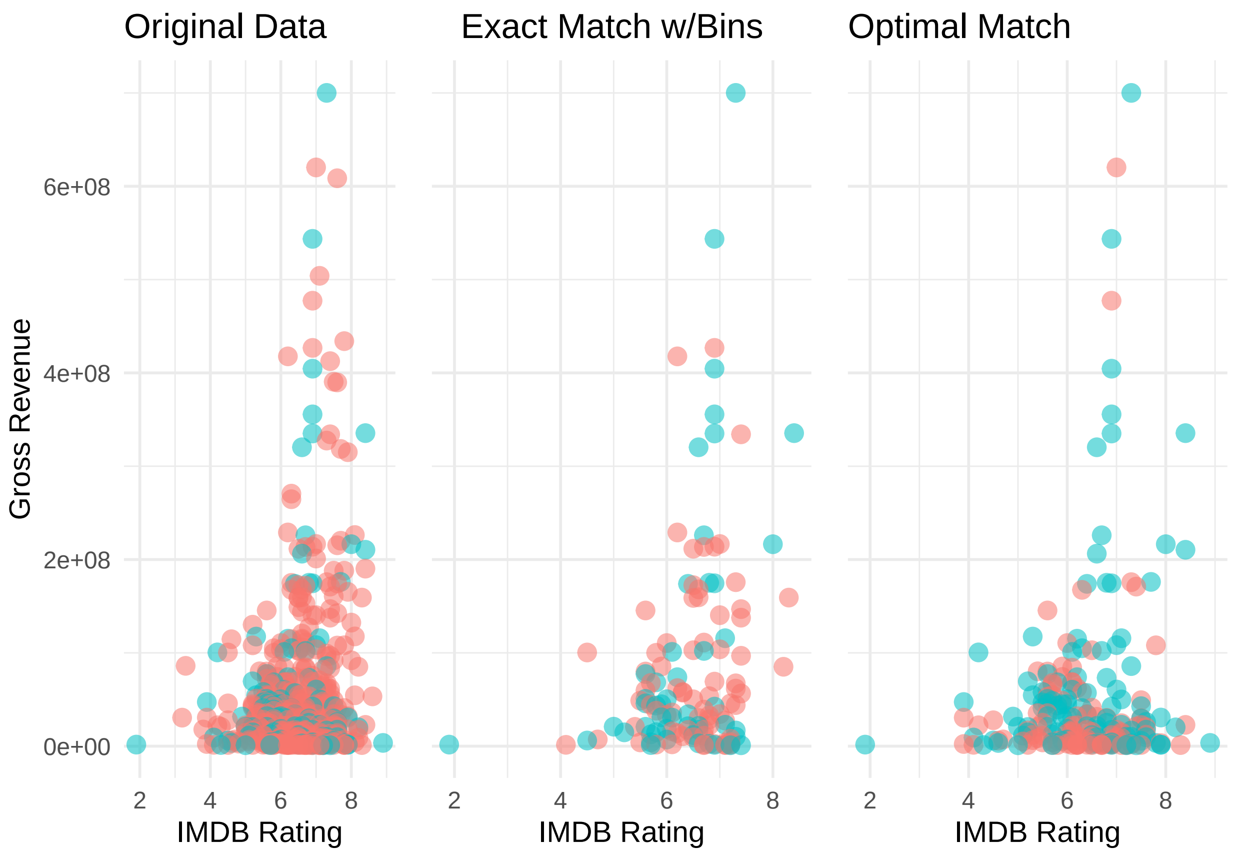
Optimal matching results in more rows than exact matching, this makes sense since exact matching is stricter in the pairs it can allow. Optimal data also balances perfectly between our treated variable (POC actors):

```
table(m.data2$Race)
```

```
##
## Not POC     POC
##     106     106
```

Let's look at the differences visually.

```
p3<-ggplot(m.data2, aes(x=imdb_rating, y=gross, color=Race))+geom_point(size=2.8,alpha=.55)+ labs(title=
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank())
grid.arrange(p1+ theme(legend.position="none")+ labs(title=("Original Data")),p2+ theme(legend.position=
```



Between Exact and Optimal there is an increase in amount of points while the overall shape is still maintained. Notably, non-POC movies matched seem to be more clustered at the bottom with two exceptions. Whether this clustering of lower-revenue non-POC movies will have an effect can be something we check with our regression.

```
#Note: Since matches were maid 1:1 there is no weight output.

summary(lm(gross~Race+year+genre+genre:Race+opening_theaters+film_rating+Male+imdb_rating,data=m.data2)
```

```
##
## Call:
## lm(formula = gross ~ Race + year + genre + genre:Race + opening_theaters +
##     film_rating + Male + imdb_rating, data = m.data2)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -139282856  -34890249   -6105185   17843582  522538080
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -4.244e+09  1.732e+10  -0.245  0.80665
## RacePOC                   5.499e+07  1.864e+07   2.950  0.00357 **
## year                      2.019e+06  8.582e+06   0.235  0.81421
## genreAdventure           -4.483e+07  2.558e+07  -1.753  0.08127 .
## genreBiography           -4.540e+07  3.409e+07  -1.332  0.18454
## genreComedy              -4.903e+07  2.322e+07  -2.111  0.03605 *
## genreDocumentary         -1.301e+07  6.504e+07  -0.200  0.84173
## genreDrama              -4.758e+07  2.539e+07  -1.874  0.06239 .
## opening_theaters         2.177e+04  6.817e+03   3.194  0.00164 **
## film_ratingNot Rated     4.960e+07  9.462e+07   0.524  0.60070
## film_ratingPG            8.747e+07  9.282e+07   0.942  0.34718
## film_ratingPG-13         7.593e+07  9.175e+07   0.828  0.40891
## film_ratingR             4.571e+07  9.152e+07   0.499  0.61807
## Male                     8.810e+06  1.282e+07   0.687  0.49279
## imdb_rating              1.616e+07  6.096e+06   2.650  0.00872 **
## RacePOC:genreAdventure   -5.411e+07  3.381e+07  -1.600  0.11120
## RacePOC:genreBiography   -5.993e+07  5.116e+07  -1.171  0.24287
## RacePOC:genreComedy      -4.058e+07  3.224e+07  -1.259  0.20974
## RacePOC:genreDocumentary -4.024e+07  7.870e+07  -0.511  0.60970
## RacePOC:genreDrama       -4.605e+07  3.331e+07  -1.383  0.16840
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 82460000 on 192 degrees of freedom
## Multiple R-squared:  0.3807, Adjusted R-squared:  0.3194
## F-statistic: 6.211 on 19 and 192 DF,  p-value: 2.986e-12
```

Once again, we see the same significant variables in the regression. Action movies with POC leads seem to generate more gross revenue while there is no statistically significant difference for the other genres.

Conclusion

When evaluating casting for a movie, there is significant evidence to prove that there is no negative causal effect from casting a racial minority in the lead role. In fact, action movies may benefit from minority leads which follows the success of several blockbuster action movies led by POC. Through the use of matching, we managed to isolate the effect of POC leads on movies by only considering similar movie groups or pairs. Our models have R-squared values between .3 and .5 showing that at least half of the variation of data is not captured, thus we should not think that these are the only factors that determine a movie's box office success. However, we have mathematically and scientifically found substantial evidence that, even outside of morality, there is no reason to discriminate against casting POC to lead movie roles.

From Team 9, we hope this has been a useful look into data and that this may fuel more representative behavior in the industry.