Understanding the defining factors of a successful actor using multidimensional scaling

Rollins, Joshua Ezra
Porluri, Saisaket
Wu, Haixia
Liu, Qiao

March 4, 2016

I. Introduction

In this study, our goal is using Multidimensional Scaling to explore the defining factors of a good actor or actress, and see if there's a difference in the professional ability of actors and actresses. Though "good" could be a rather arbitrary measurement, in this paper we define the goodness of performers by their professional achievements in the industry: the recognition they get from the professional community, the ability to create entirely different characters, and the involvement they have in the different aspects of creating a movie/TV series.

As mentioned above, the tool we will be using in our project is Classical Multidimensional Scaling (MDS). Multidimensional Scaling is a set of statistical techniques often used in visualizing similarities or dissimilarities of individual elements in the data with a specified distance measure. An MDS algorithm starts with a distance matrix that shows all item-to-item distances. Each object is represented by a point in a *k*-dimensional space, similar objects are represented by two points close together; two dissimilar objects are represented by two points that are far apart. The number of dimensions of an MDS plot, *k*, can be any integer. However, after some investigation in the later section, we decided that using a *k* of 2 best fits our purposes.

In our paper, we randomly generated 200 actors and actresses from the IMDb database and did MDS with the following defining vectors: total film count, number of director credits, number of producer credits, number of professional awards won, number of professional awards nominated, and number of roles.

Then, in order to explore the effects different factors have on the MDS, we did 3 variations with:

- Evaluating the accumulative results instead of averaging by career length;
- Instead of our original list, use the one with recent 50 Oscar winners;
- Excluding the vector for producer and director credits.

II. Background

The film industry is growing to be one of the biggest players in the broader entertainment sector; it is considered a cornerstone of the industry, and creates billions of dollars of profits each year. But there is a known pay gap between male and female performers. Our group is curious of whether there is actually a difference in professional abilities between actors and actresses, or it's mostly due to gender bias. So we will try to explore the defining features for a successful actor/actress and see if gender has any significant effects on their professional abilities by doing multidimensional scaling on 200 randomly selected movie stars.

Multidimensional Scaling (MDS) is a way of visualizing the level of similarity of individual cases of a dataset. It preserves and visualizes the distances between objects in a higher-dimensional space to a lower-dimensional space. It is a very useful tool in fields like information science, psychology, ecology and marketing. We didn't find any history of researching the film industry by MDS technique. However, there is

an article by Lee G. Cooper titled *A Review of Multidimensional Scaling in Marketing Research*.[2] In this research, Lee collected specified products from different brands, prices, personal sellers to find out the similarities and dissimilarities between those products. By mapping the products, marketing researchers found out how the products differ from each other and worked to figure out the market structure. The result shows that the selling quantity of products is essentially related to price and advertisements. In our study, the research objects are actors and actress, instead of products, but we are doing a very similar comparison.

III. Model

3.1 Data Collection

To find suitable data source for actors and actresses, we needed an easily parsed, standardized, freely available database that offers information about actors and actress as well as their career performance. We chose to use the IMDb as our source for data. Using website languages such as HTML and PHP, we were able to gain and process data from the MySQL IMDb database. However, during our search we found many amateurs with very small film counts were being included. In order to insure the baseline professionalism of our performers, we added a constraint for only actors and actresses with more than 100 film count to be included in our data. This is a valid constraint, for from the data we get we see that even young performers who have a moderately long period of production, like Natalie Portman, have 167 film counts; and a more prolific one, such as Tom Cruise, has up to 326 film counts. But keep in mind that this film count from the database is a gross add up of all the films that they are credited for. We have manually double checked their identity and found all of them to be professional actors or actresses.

We used Query to select information about the number of films those 200 actors and actresses have appeared; number of times they were credited as directors (Director Credits); number of times they were credited as producers (Producer Credits); number of times being professional awards nominees; number of times being professional awards winners; number of roles they have been performed. Professional awards include Academy Awards (Oscars), Golden Globes, Emmys and others that represent recognition from the professional community. Additionally, we needed to know actors' career length to calculate the rate they produce films, which is not given directly by IMDb. So we defined it as the time interval between the release date of their first and last movie.

To calculate the distance between each pair of performers, we used the Euclidean distance formula on the 6 factors above to compute the distance between each pair in 2-dimensional space. Using that method we created a 200 by 200 distance matrix to feed into the classical multidimensional scaling process.

3.2 Methods

By reading the data, we found that those actors and actresses with older age tend to have larger film counts, more director credits, producer credits, roles counts, award wins and nominations. For a fair comparison, we therefore decided to measure the rate they produce films, instead of the accumulative amount they've produced. We do so by dividing the original film count by their respective actor or actress's career length:

Career length = release date for their last movie - release date for their first movie

To adjust those values measured on different scales and make them comparable, we then normalized the data. To create the normalized data we applied this formula to each element:

$$normalize(x) = \frac{x \ - \ column\ mean}{column\ standard\ deviation}$$

And now, we have a new 200 by 6 matrix called "*dat*", code to create *dat* and original data are included in Appendix A and Appendix H respectively. The next step is to apply the multidimensional scaling process to

the matrix *dat*. We calculated the distance between each pairing of actors and actresses by using the Minkowski method with *p* equal to 2, also known as the classic Euclidian distance. Thus, the distance between row *i* and row *j* in category *m* will be calculated as,

$$d_{ij} = (\sum_{m} |r_{i,m} - r_{j,m}|^2)^{1/2}$$

Once we created the distance matrix for each pair of actors, we run MDS on the distance matrix to obtain our result. Multidimensional scaling takes in the distance matrix, and places each actor or actress in a *k* dimensional space. In this project, we analyzed the goodness of fit for each *k* value and decided upon *k* equal to 2. We then plotted the two-dimensional results on graphs, which visually show the clustering relationship of different actors and actresses.

IV. Solution

In order to explore our data, we wrote R codes to perform above methods. The code reads in a specially formatted csv file with information about the 6 defining vectors of 200 actors and actresses. Through the process described in section 3.2 the code produces a plot of actors and actresses in two dimensions as well as goodness of fit (all code can be found in appendix A).

Figure 1 and 2 show the resulting plots, we found that most of points lie clustered together with a few are large outliers. To read most of points clearly, we decided to zoom into the clustered portion of the graph, as shown in Figure 3 and 4.

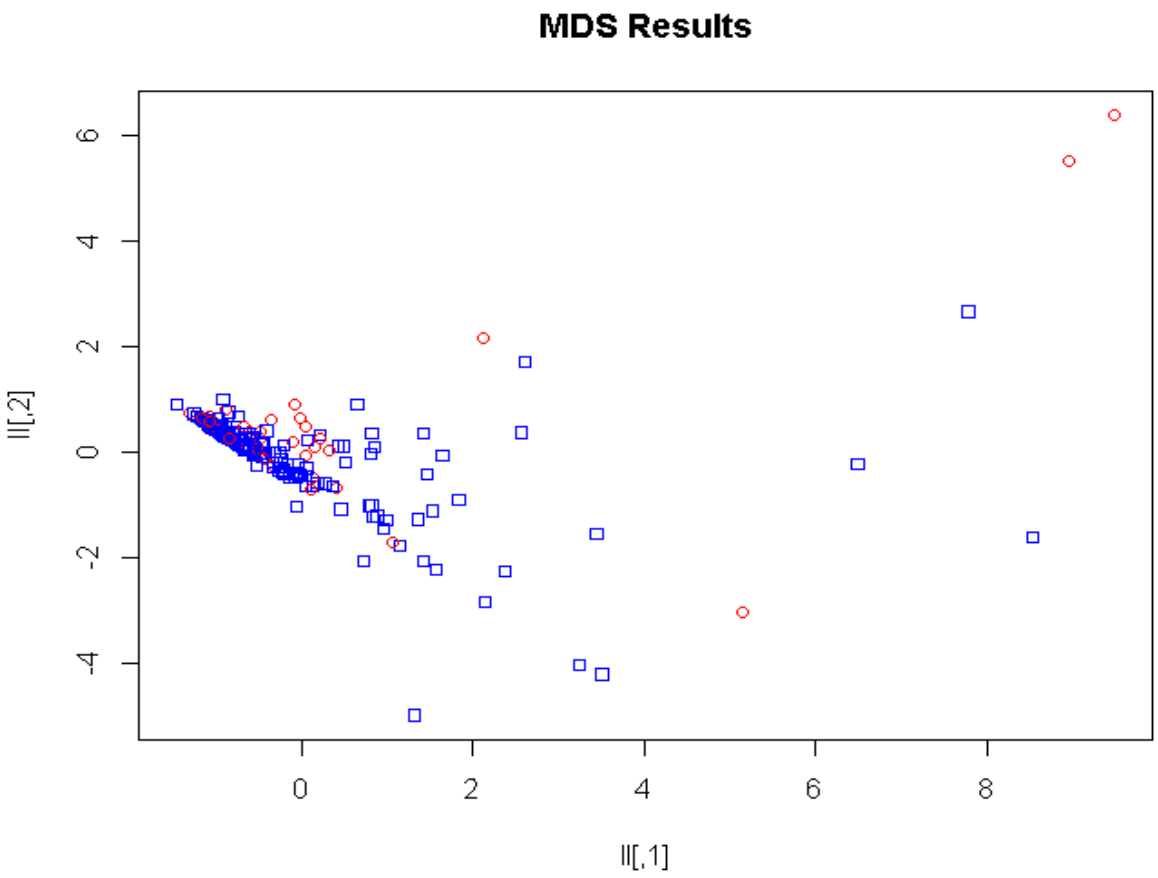

Figure 1: MDS representation of 200 actors and actresses, circles are actresses and squares are actors

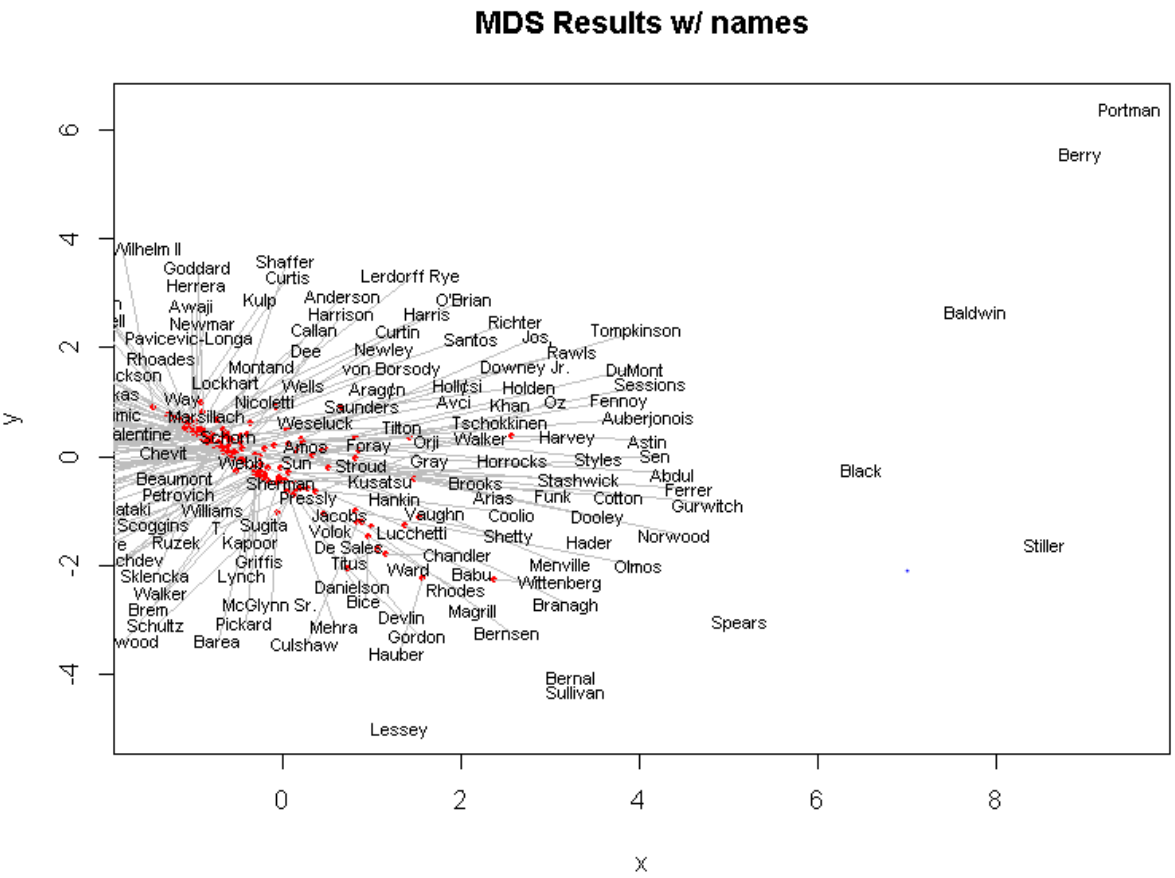

Figure 2: MDS including the last names of each individual

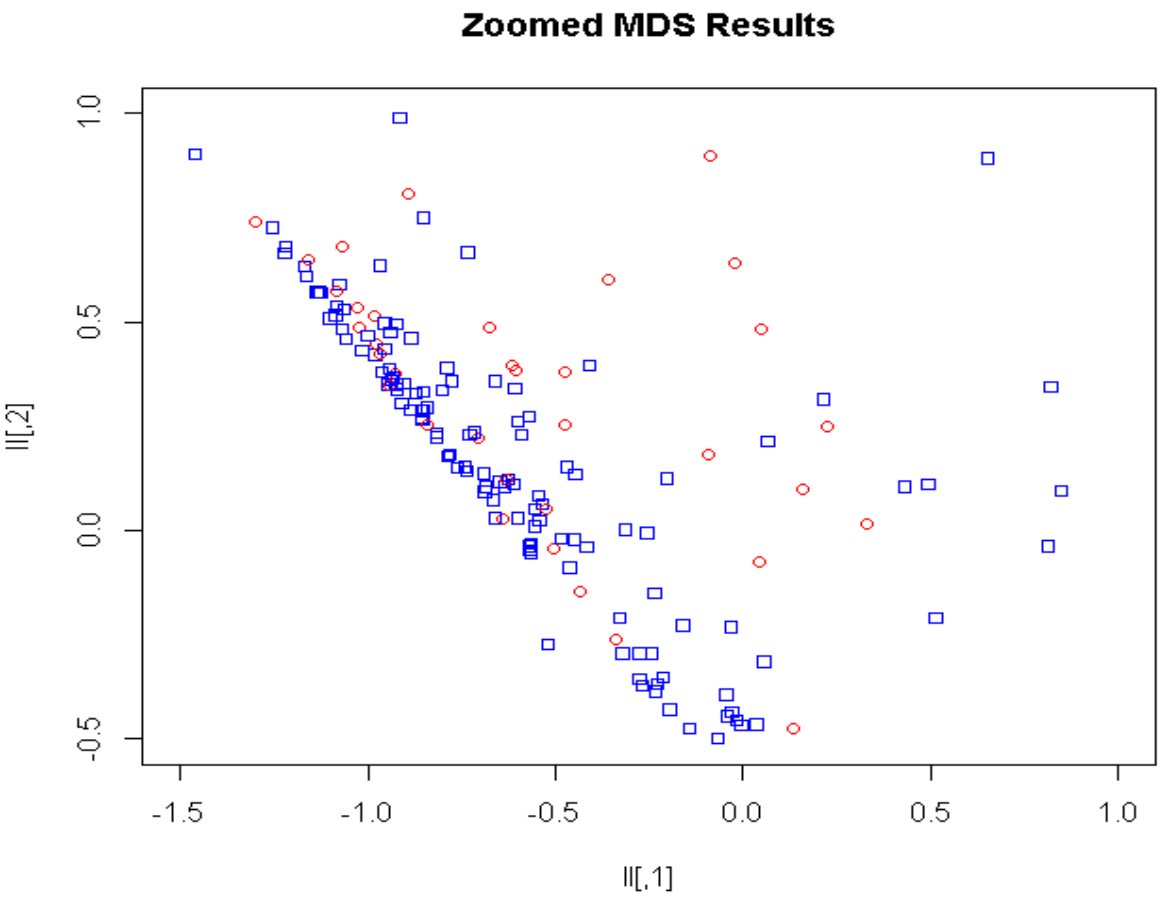

Figure 3: Zoomed MDS representation of 200 actors and actresses (squares are male circles are female)

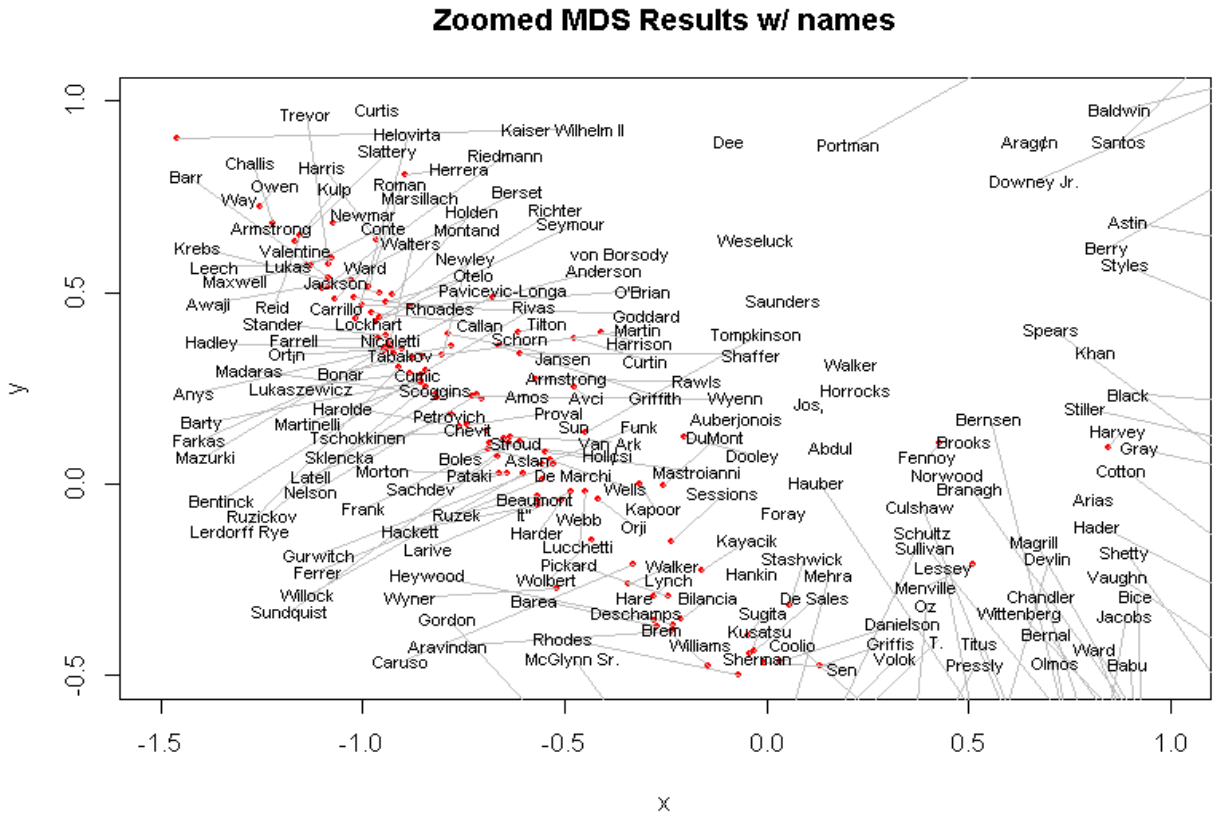

Figure 4: Zoomed MDS plot of 200 actors and actresses including last name

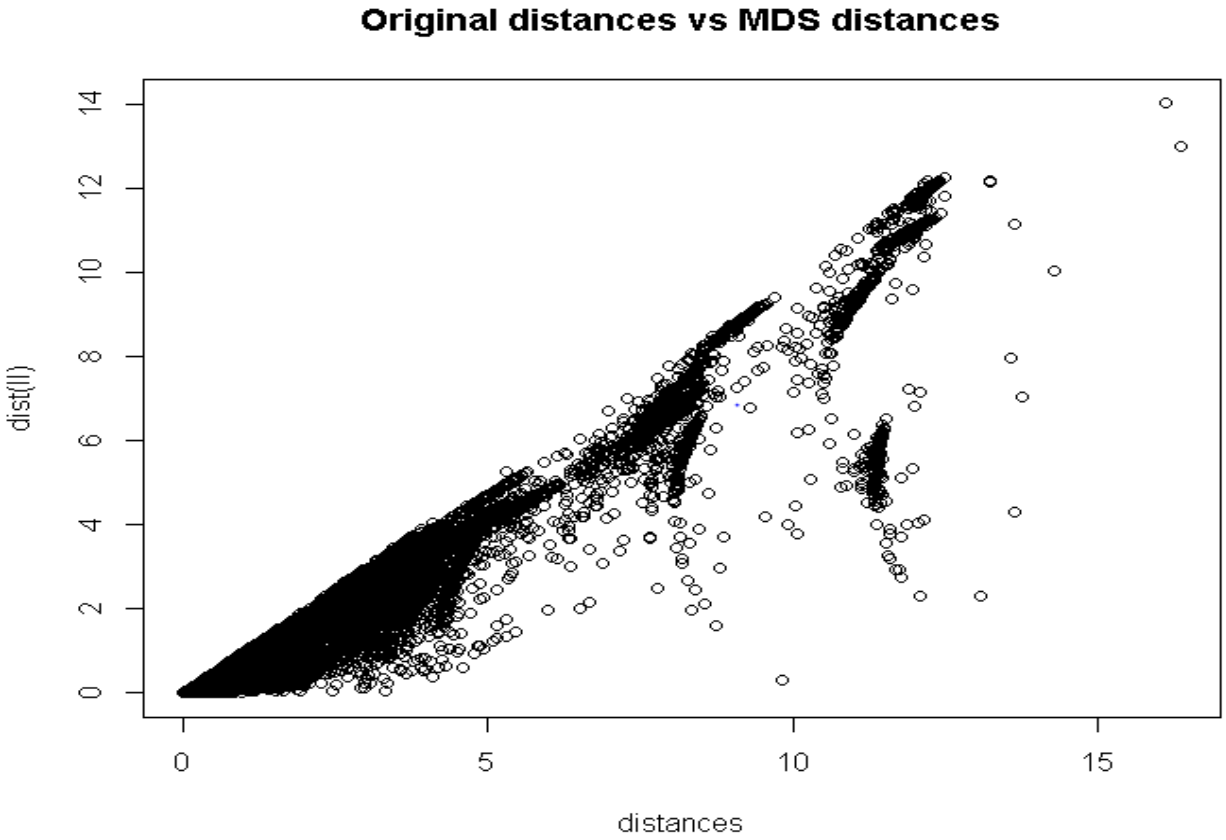

Figure 5: Distances in the MDS model compared to original distance

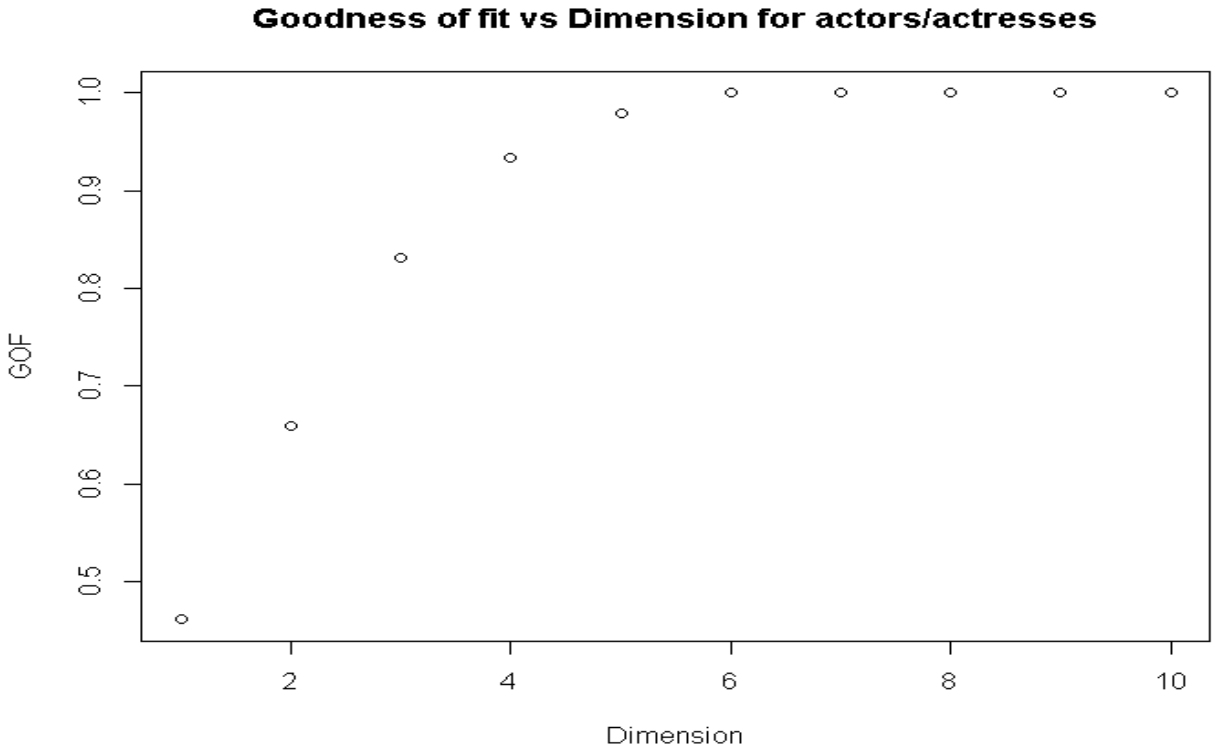

Figure 6: Goodness of fit (GOF) vs. MDS dimensions

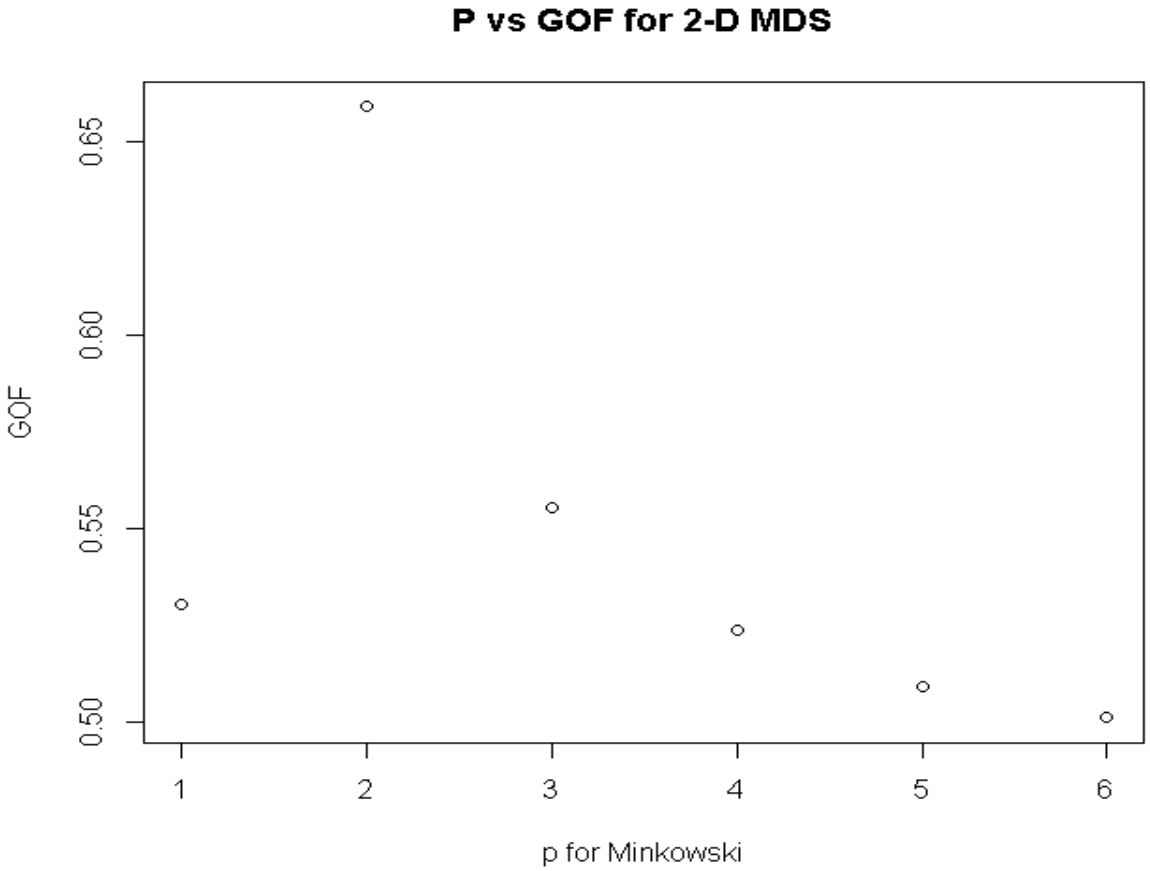

Figure 7: Visualization of value of p in Minkowski's relation to goodness of fit

Figure 5, 6 and 7 are used to check how good our model is. Figure 5 is used to check if our MDS distances are similar to the original distances, ideally all dots would be on the line of $y = x$. Figure 6 is showing how GOF relates to the number of dimensions in MDS model. And we used the following command to check the value of GOF when the MDS dimension equals 1 and 2:

```
> # Checking goodness of fit
> cmdscale(distances, k=2, eig = TRUE)$GOF
[1] 0.6590222 0.6590222
> cmdscale(distances, k=1, eig = TRUE)$GOF
[1] 0.4611713 0.4611713
```

Figure 7 is showing how GOF relates to number of p-values of various Minkowski distances.

V. Commentary on Solution

To find out how good our MDS model is, we want to see how much of the original distance is preserved by our 2D model, so we created a graph of the distances in the model versus the original distances (Figure 5). And we found that our graph aligns with the ideal case of $y = x$. Meaning we have a reasonable representation of the orignal distances.

We compared our goodness of fit (GOF) with the dimensions of the MDS in Figure 6. From which we can see that, while having decent GOF for one dimensional MDS (.4611), our GOF for two dimensions (.6590) is much better. This means that the original distance is better preserved with more than one dimension. Also considering the simplicity, we decided to use a two dimensional MDS. Comparing this value of goodness of fit to that of random data (average of .17), we can see that our model has a better represent the differences between each person.

In Figure 7, we compared the GOF with the p value of the Minkowski distance and found that the Euclidean distance($p = 2$) provides the best fit. So we will use the Euclidean distance to visualize our higher-dimensional model.

From our general knowledge of film and further research into the kinds of films the outlier performers have played a role in, we have determined the predominant deciding factor for y-axis is how artsy the performer's films are. The more overall depth and sophistication of the film, and the more of these kinds of films a person has been in, the higher he or she is on the y-axis. Analyzing more defining factors brought us to the conclusion the x-axis is determined by the commercial success of a performer's films. While performers like Ben Stiller may not have much depth to their films, commercial success of their films still allow for them to escape from the cluster.

VI. Variations

6.1 Unscaled Data

Our first variation is excluding career length as a averaging factor in our data, so we can measure the success actors/actresses have accumulated over their whole career. The process was very similar to the original case, but we do not divide each individual's data by their career length. Code for this variation can be found in Appendix B.

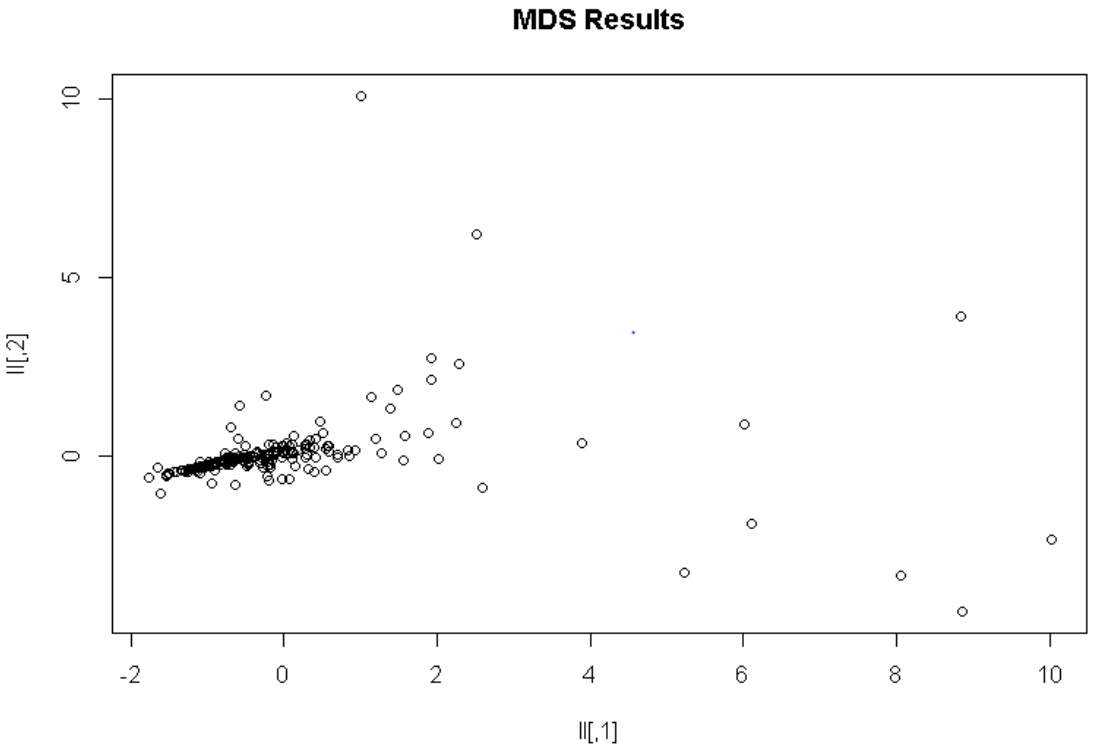

Figure 8: MDS representation of 200 actors and actresses in first variation

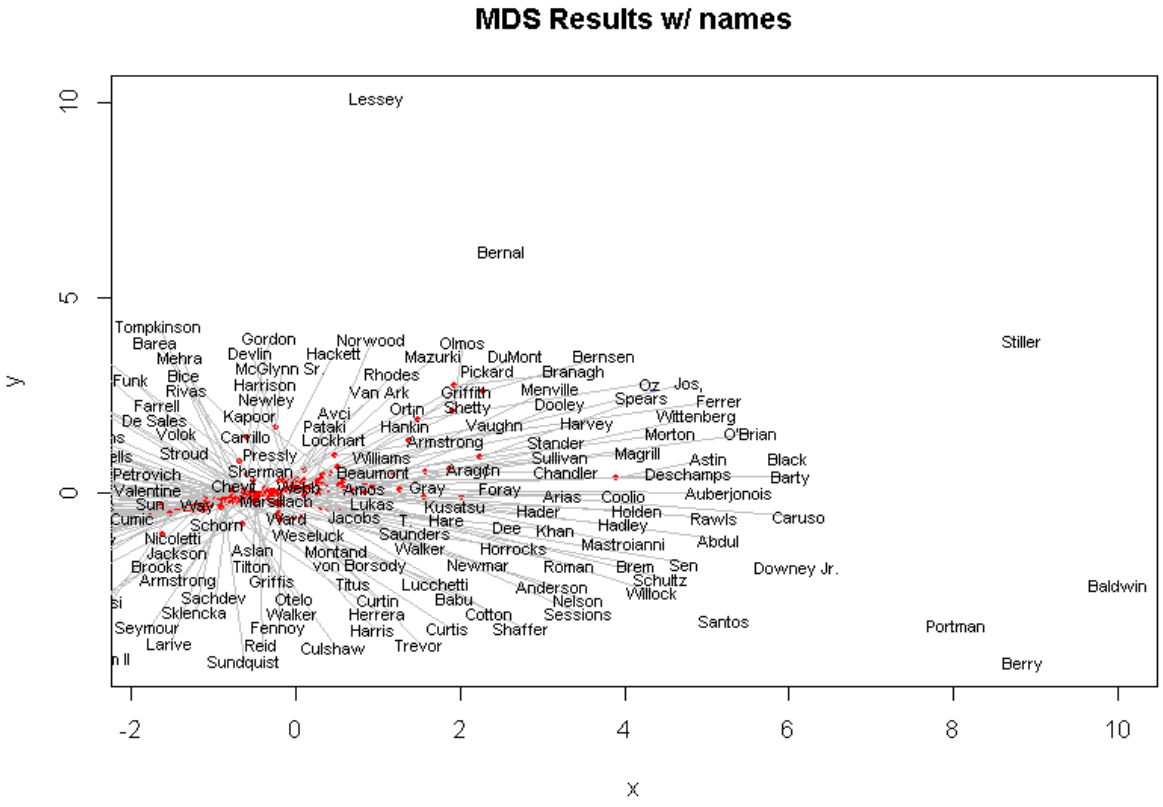

Figure 9: MDS representation with last names in first variation

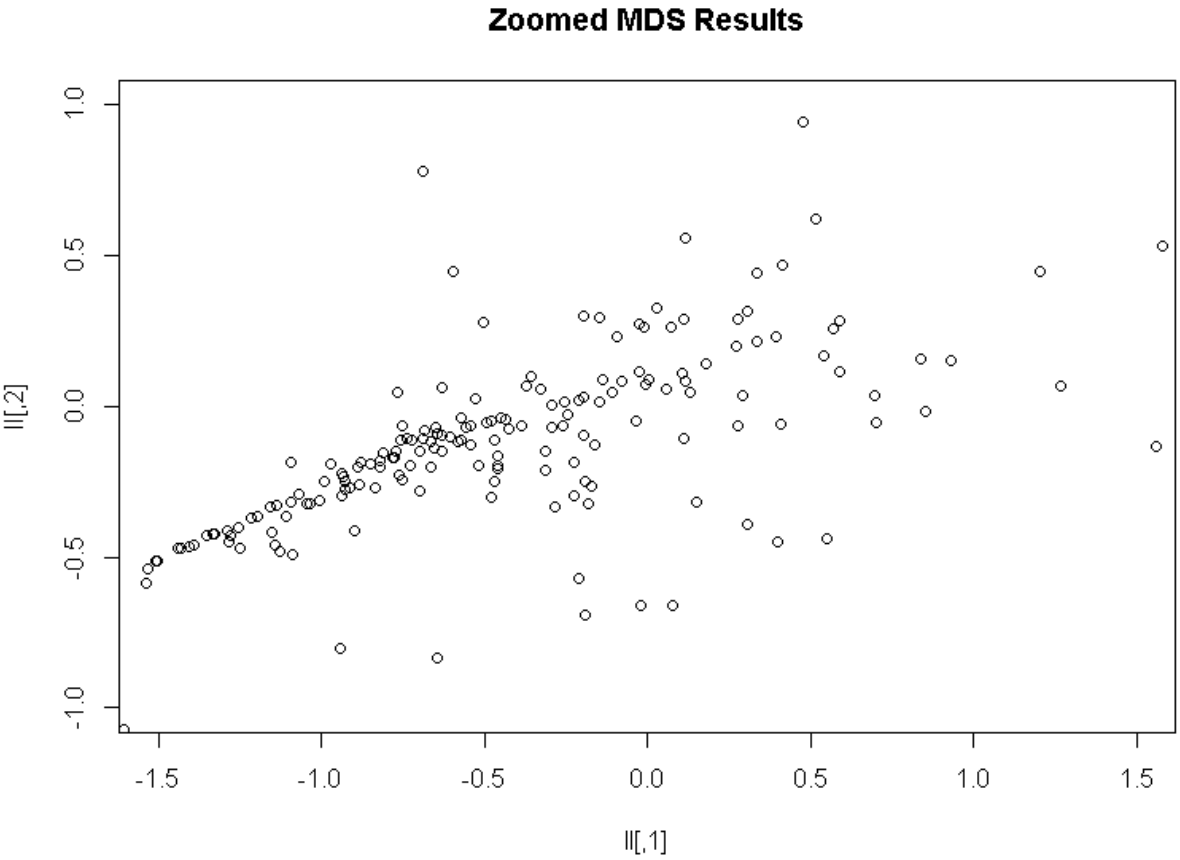

Figure 10: Zoomed MDS representation in first variation

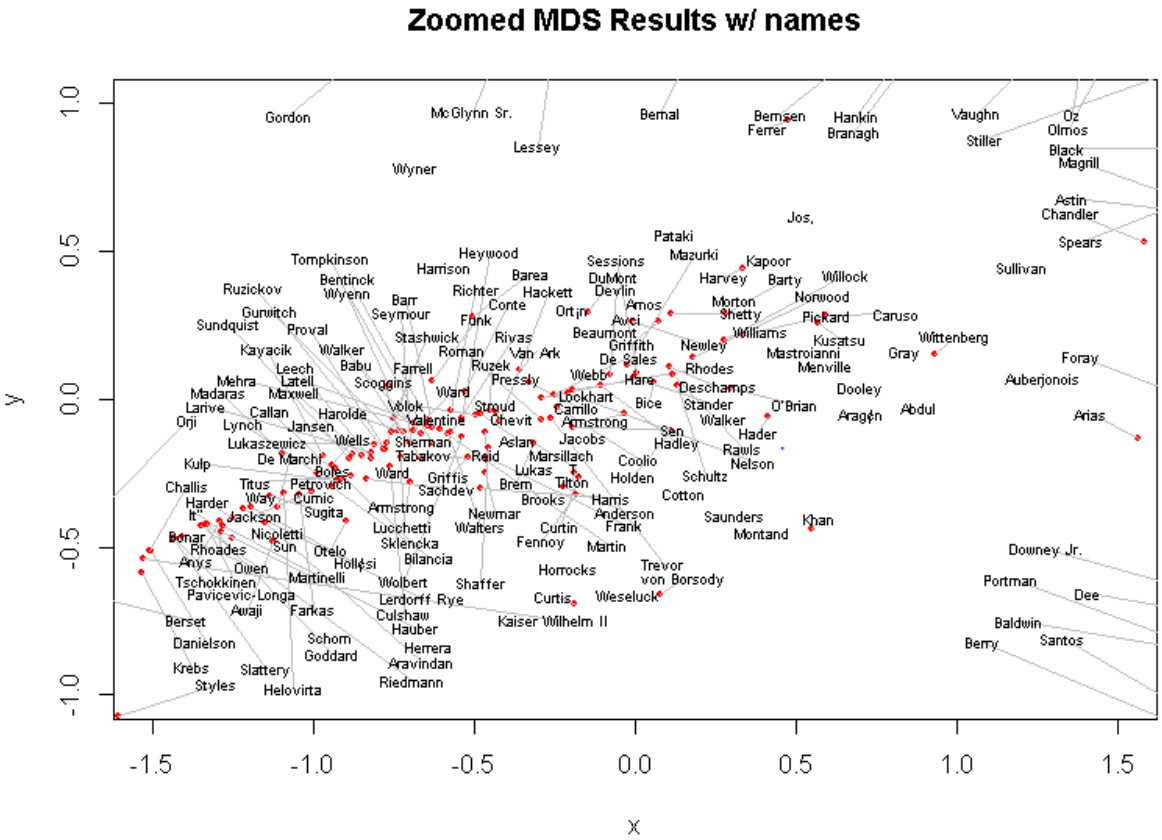

Figure 11: Zoomed MDS representation with last names in first variation

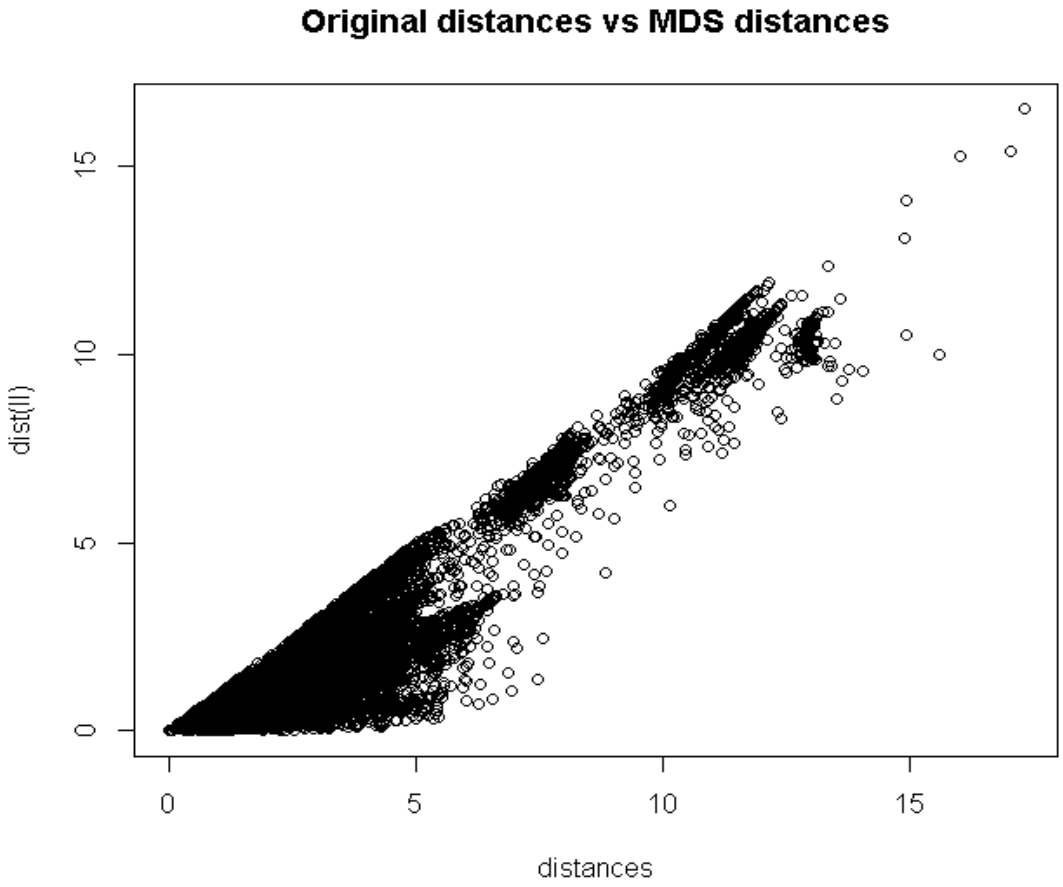

Figure 12: Distances in the MDS model compared to original distance in first variation

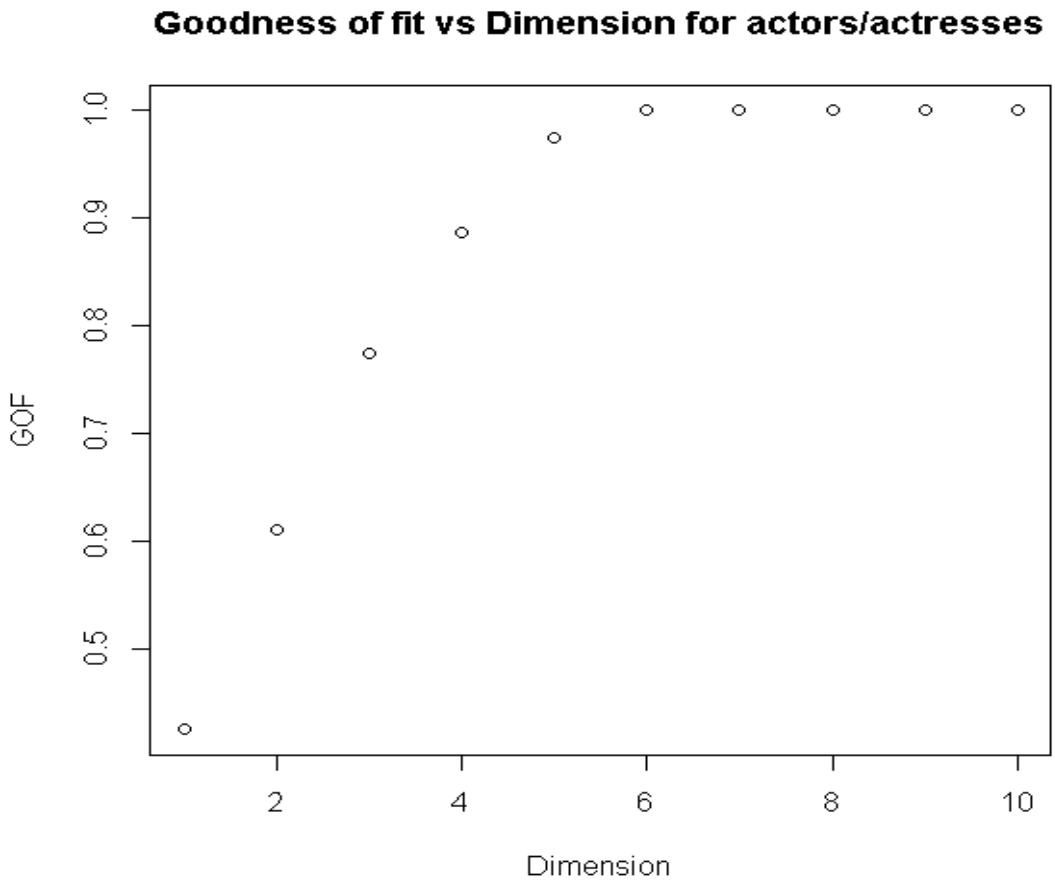

Figure 13: Goodness of fit (GOF) vs. MDS dimensions

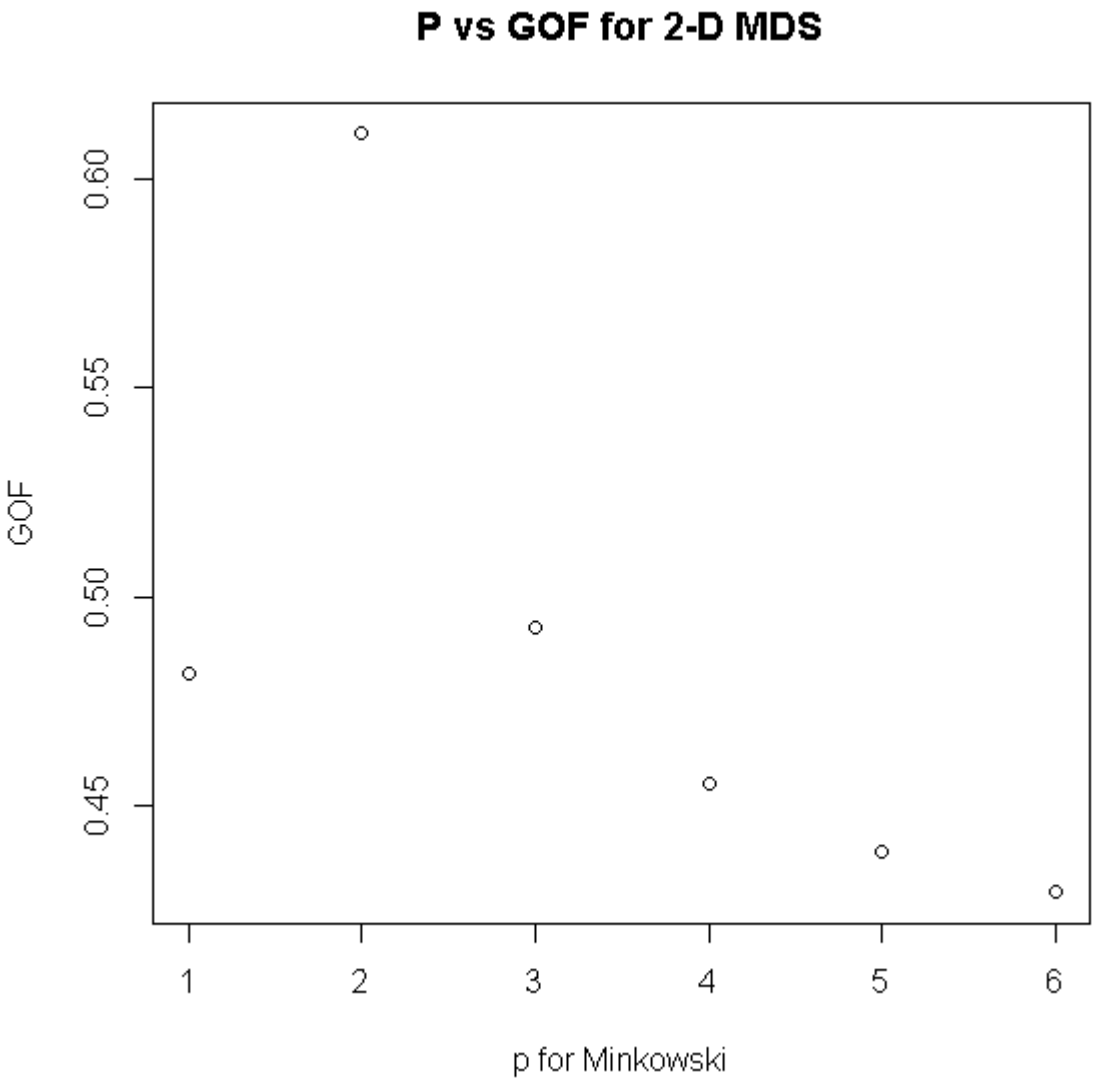

Figure 14: Visualization of value of *p* in Minkowski's relation to goodness of fit

```
> # Checking goodness of fit
> cmdscale(distances, k=2, eig = TRUE)$GOF
[1] 0.6108405 0.6108405
> cmdscale(distances, k=1, eig = TRUE)$GOF
[1] 0.4247142 0.4247142
```

With a GOF for two dimensional MDS we know the model is much better than that of a random model. However our GOF value is smaller than the original variation's GOF meaning removing the scaling by career length does decrease the accuracy of the model.

Searching for common characteristics in the data, our y-axis seems to be related to the director credits of the performer. Lessey having the most (88), followed by Bernal (40) and Stiller (16) we also see Berry down at the bottom having zero director credits.

The x-axis seems to be related to professionalism of the performer. Actors and actresses more widely known and who receive many professional awards have a higher x value.

Comparing these results to that of the original variation, we realized that when we use accumulated data instead of data per year, creativity and ventures into directing and producing seem to be rewarded greater.

6.2 Oscar Winner Variation Including Net Worth and Followers Number

There were some concerns with our initial data. The actors and actresses whom were born in the early 20th century and acted in older movies did not contain much information for three particular reasons. One is that movies were not that popular in the mid-20th century because movie theaters had not become as widespread as it is today. Another issue is that data from the movies had not always been consistently recorded or kept properly because the lack of digital technology during that era. Furthermore, many of these actors were foreign and acted in movies not in English, so the audience is more limited, resulting in fewer ratings and nominations.

Because of a lack of information on some of the actors in the original 200 people's data set, we decided to do a variation on the 50 most recent Oscar Winners. This will allow us to have a sample set of fairly recent, popular actors, which we lacked in our initial data set. Oscar nominees also tend to be fairly diverse in terms of genre of films they usually act in, age, and number of films becaues actors are given Oscars for their performance in a single role, not based on their career. We have also added two defining vectors: their net worth, indicating how much total assets they controlled, (in million) and Facebook or Twitter followers (in million). Net worth is a good indication of how successful an actor has performed in the box office. General higher net worth comes from higher paychecks and demand to act in a movie. Twitter followers indicates how popular and in-touch an actor or actress is with fans.

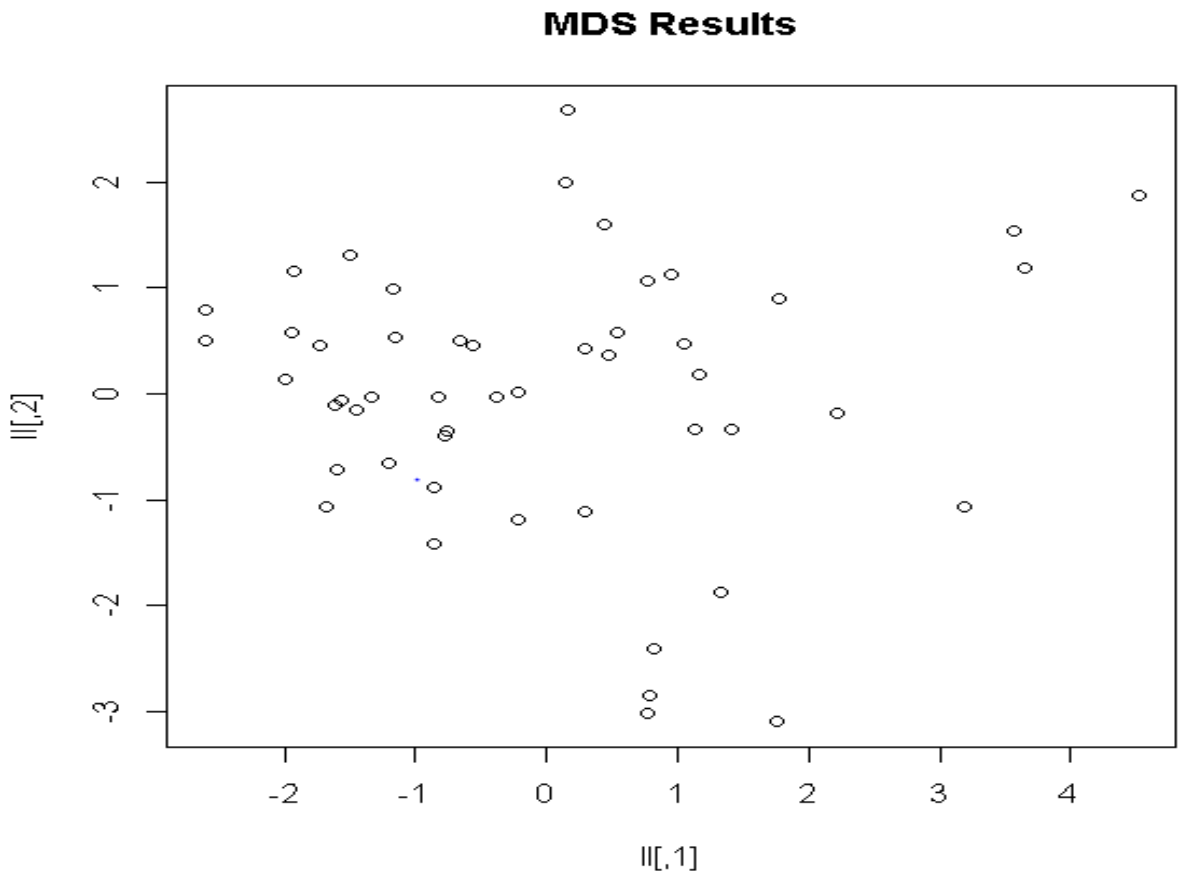

Figure 15: MDS representation of top 50 oscar winners in second variation

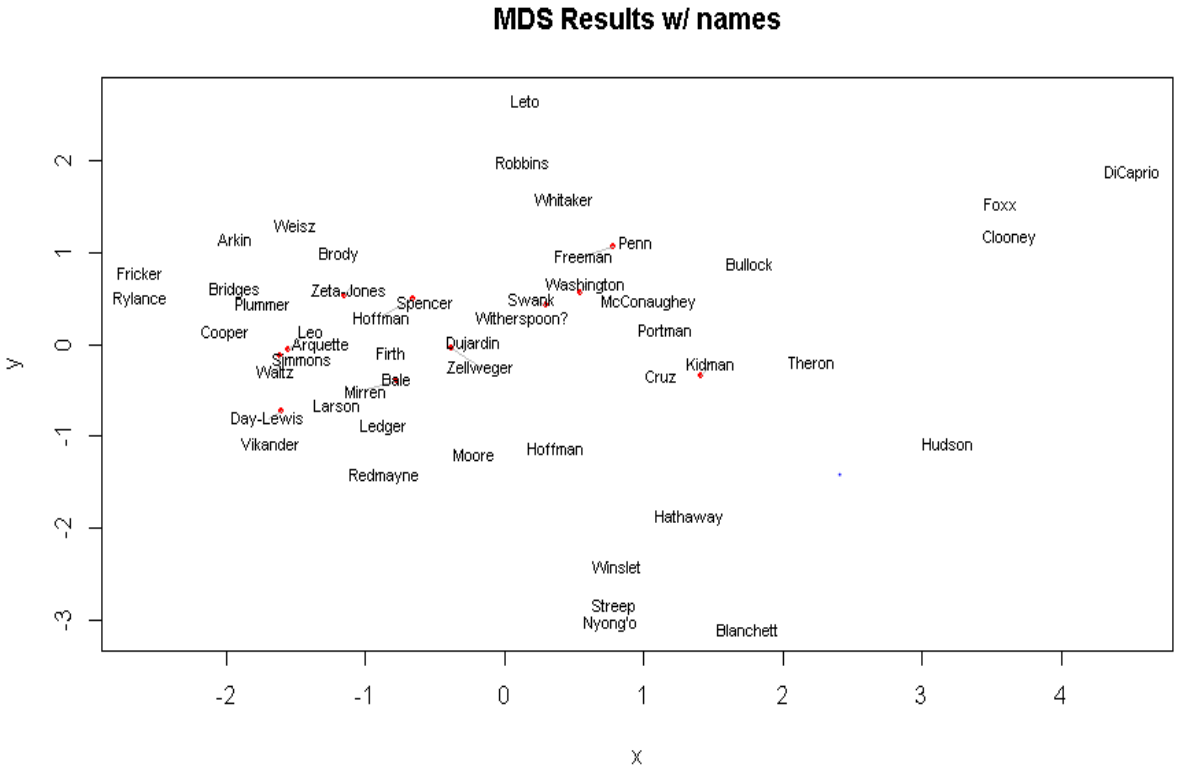

Figure 16: MDS representation of top 50 oscar winners with their last names in second variation

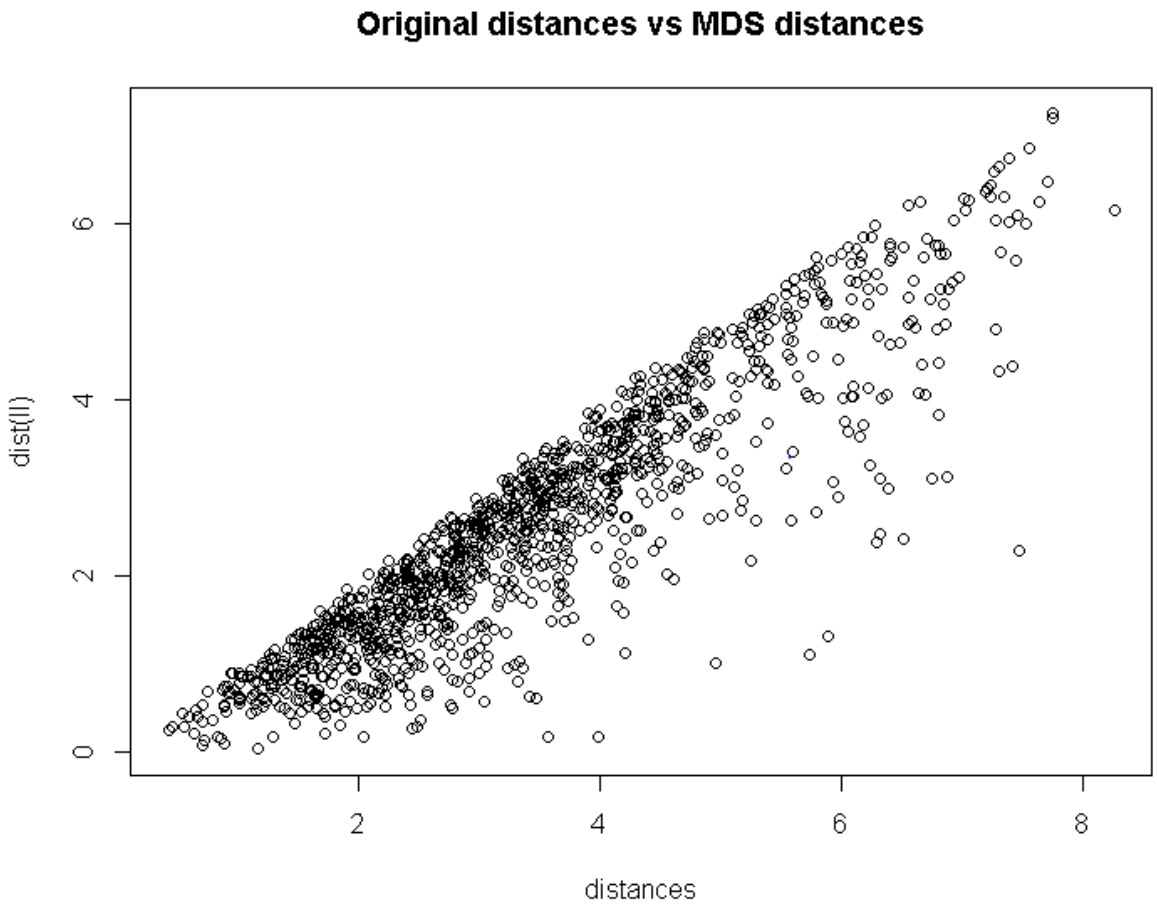

Figure 17: Distances in the MDS model compared to original distance in second variation

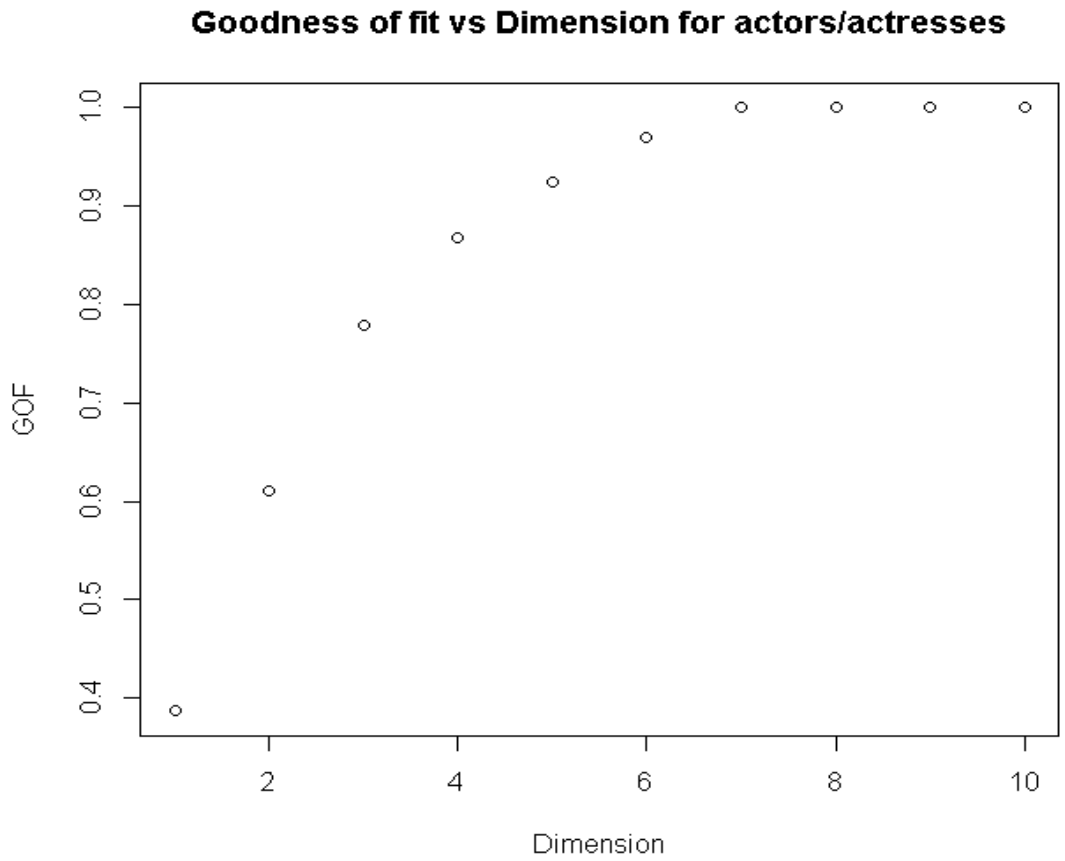

Figure 18: Goodness of fit (GOF) vs. MDS dimensions in second variation

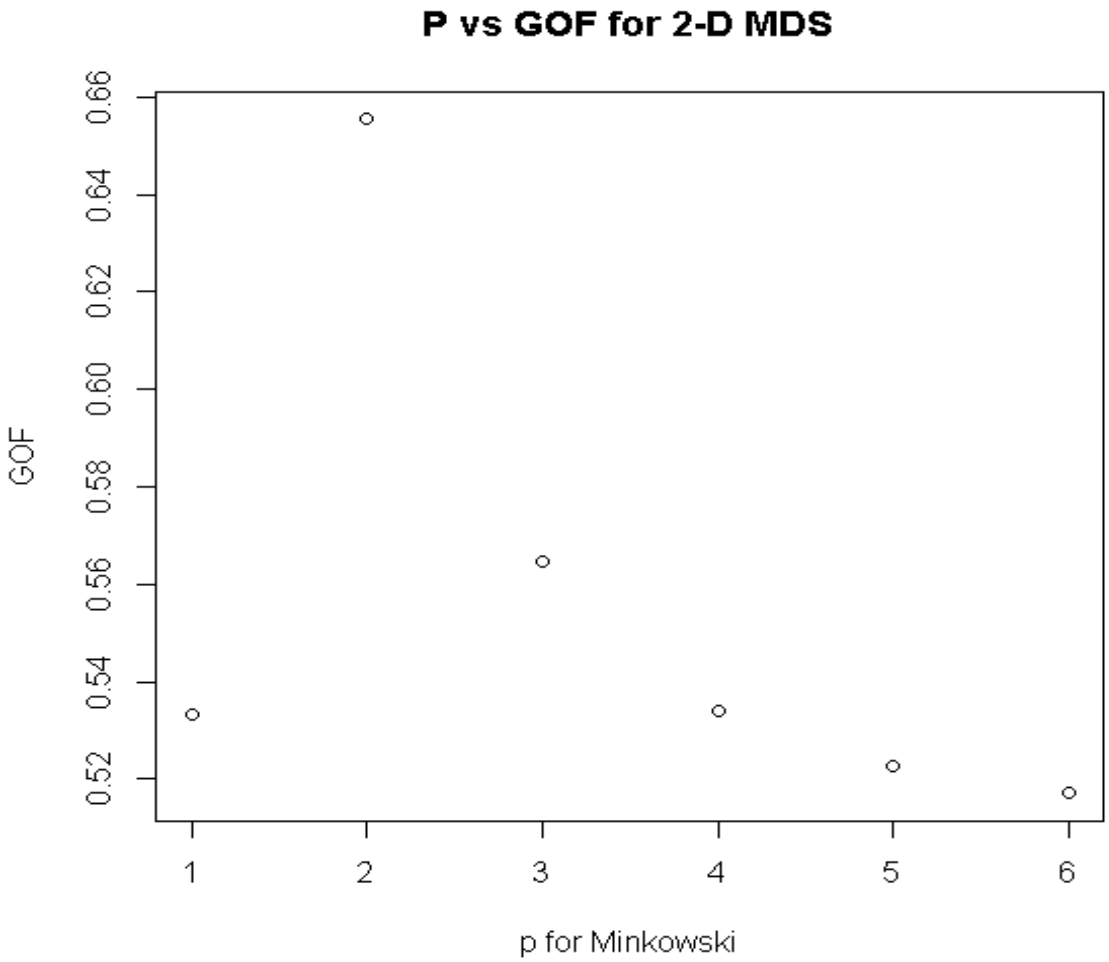

Figure 19: Visualization of value of *p* in Minkowski's relation to goodness of fit in second variation

We notice that there is a cluster of 3 actors, far from the general cluster of actors in the middle, in the top right of the MDS. These actors are Leonardo DiCaprio, Jamie Foxx, and George Clooney. All of these actors are fairly prominent, and have a high net worth when compared to the other actors. Furthermore, each one of these actors has a high number of films.

The bottom cluster of actors, including Lupita Nyong'o and Meryl Steep, happens because they have a couple of statistics that are very far away from average. For example, Meryl Streep is on the older at 66 and Lupita has acted in only 30 movies and lower net worth of $420,000. However, these actors should be pretty far away ideally from each other since Streep has a much higher net worth at $45 million and about 4.5x times more social media fans than those of Lupita. Although our model turned out well with a GOF of 0.88, there are a couple of distances between actors that are inconsistent, Lupita and Streep being one of them. We can estimate that the x-axis represents the popularity and prominence of an actor. People having low x coordinates are those who are less popular, have acted in fewer movies, have less social media follows, and lower net worth. Actors with high x-coordinates are those who are more popular, acted in many movies, have a lot of social media followers, and higher net worth. For example, Leonardo DiCaprio, a very popular actor, has a high x value because he is very popular, well-known, and successful and Mark Rylance is an example of someone who is not so popular and well known, so he has a very low x-coordinate.

We can estimate that the y-axis represents the number of awards and nominations that one has received. A high y coordinate represents fewer awards won whereas a low y coordinate represents more awards won. For example, Cate Blanchett has been nominated for 149 professional awards and won 94 professional awards, so she has one of the lowest y coordinates. On the other hand, Jared Leto only has 21 professional award nominations and won only 2 professional awards, so he has a fairly high y coordinate.

6.3 Excluding Director Producer Credits

After the above exploration, we question if certain vectors like Director and producer credits can a limiting factor. For most actors and actress do not have the financial access to a producer's position, or are not trained to be a director (though we also know that a number of actors learned the technique of directing during their time on set.) So we exclude these two categories in our third variation, to see if this is going to change the results by much.
The results we get are as follow:

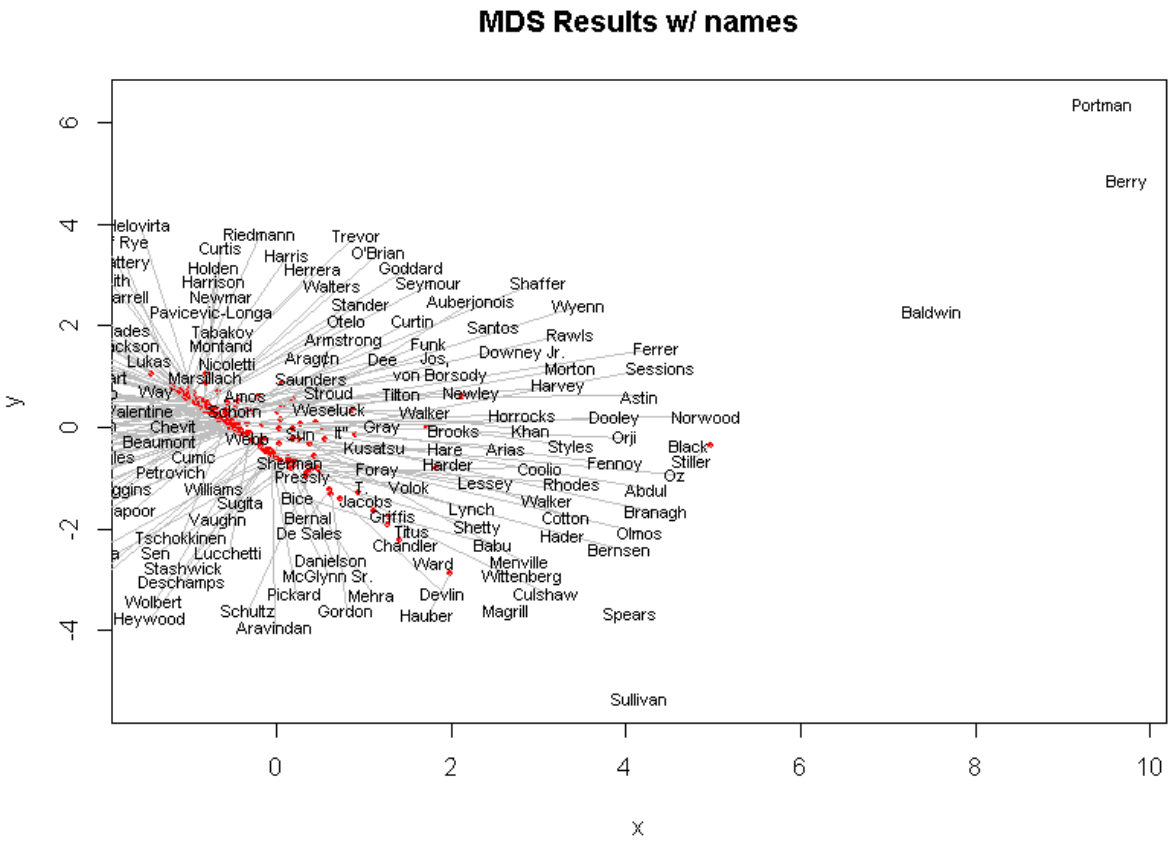

Figure 20: MDS representation of top 50 oscar winners in third variation

Figure 21: Distances in the MDS model compared to original distance in third variation

```
> # Checking goodness of fit
> cmdscale(distances, k=2, eig = TRUE)$GOF
[1] 0.8784722 0.8784722
> cmdscale(distances, k=1, eig = TRUE)$GOF
[1] 0.5897231 0.5897231
```

Excluding director and producer credit variation analysis:
When we exclude director and producer credits we get an amazing goodness of fit of .87

Immediately, we can notice two outliers consisting of Natalie Portman, Halle Berry, and, to some extent, Alex Baldwin. These actors are distance far from the general cluster of actors in the bottom left. The reason is why that all these actors have a characteristically large number of professional award nominations and wins. For example, Natalie Portman has 58 nominations and 55 wins, Alec Baldwin has 76 nominations and 37 win, and Halle Berry has 41 nominations and 76 wins. Most of the other actors have less than 10 in each category.

We postulate that the x axis represents the number of nominations that one has. With a higher x-coordinate, an actor has higher number of nominations. For example, Alec Baldwin and Paul Lucas have around the same y coordinate, but drastically different x coordinates. The difference in Alex Baldwin and Paul Lucas is that Baldwin has a much higher number of nominations won, with 76 nominations and 37 win as opposed to Paul Lucas's 3 nominations. Therefore, the x-axis seems to represent the number of nominations.

We postulate the y-axis represents information on the number of credits of an actor. An actor with a higher y coordinate represents self-credits when compared to an actor with a lower y-axis. For example, consider this piece of evidence: Anthony Newley and William Hauber have around the same x-coordinate, but Hauber's y coordinate is much smaller than that of Newley's. This is because Hauber has 0 self-credits whereas Newley has 70. This is probably because William Hauber was a rather small-time actor from the 19th century, so he was not well-known or there was not enough on the actor, leading to less number of credits.

VII. Conclusion

After investigation and thorough examination of our original MDS model, we see that the defining features of a good performer are on the level of appreciation for movie as an art, and the commercial success of a movie. There are no significant distinctions between male and female performers. So we reached the conclusion that gender is not a defining factor of one's professional ability as a performer. Additionally, for all variations and original analysis, we found the best *p*-values for Minkowski distances is 2.

Although our GOF declined in our first variation, it later becomes better and better in second variation, and third variation. For the first variation, we tried not scale the data by career length and the GOF was less than original one. Therefore, we reasonably believe that scaling by career length would be more accurate to interpret factors that have effect on professional abilities; For the third variation, we excluded director and producer credits from defining vectors, and our GOF rose to 0.8785 when k equals to 2. It means the producer and director credits of actors and actresses don't have significant effect on their professional abilities.

VIII. Bibliography

1. “All Time Top 100 Stars at the Domestic Box Office”, 2015, http://www.the-numbers.com/people/records/top-grossing-stars-in-all-roles
2. “A Review of Multidimensional Scaling in Marketing Research”, 1983, http://www.anderson.ucla.edu/faculty/lee.cooper/Pubs/A%2520Review%2520of%2520MDS%2520in%2520Marketing%2520Research.pdf
3. “The Academy Awards through the years”, 2016, http://timelines.latimes.com/academy-awards/
4. "TheRichest." TheRichest Home Comments. Web. 03 Mar. 2016, <http://www.therichest.com/>
5. "Richest Celebrities | Celebrity Net Worth." RSS. Web. 03 Mar. 2016. <http://www.celebritynetworth.com/richest-celebrities/>.
6. "Twitter." Web. 03 Mar. 2016. <https://twitter.com/>. Used for getting information about twitter counts of actors.
7."Facebook Logo." Facebook. Web. 03 Mar. 2016. <http://www.facebook.com/>. Used to obtain the number of facebook fans for an actor.

IX. Appendices

Appendix A. R Code for Original Variation

```
# Take in the data
predat <- read.csv(file.choose(), header = TRUE)

# Initialize data matrix with columns we want to include
dat <- matrix(nrow=200, ncol=6)
dat[, 1] <- predat[, 5]/predat[, 16]    # Films/Career_length
dat[, 2] <- predat[, 8]/predat[, 16]    # DirectorCred/Career_length
dat[, 3] <- predat[, 9]/predat[, 16]    # ProducerCred/Career_length
dat[, 4] <- predat[, 12]/predat[, 16]  # ProfAwardNom/Career_length
dat[, 5] <- predat[, 13]/predat[, 16]  # ProfAwardWin/Career_length
dat[, 6] <- predat[, 18]/predat[, 16]  # Roles/Career_length

# Initialize another matrix to store additional plot data
addDat <- matrix(nrow=200, ncol=2)

# Colors by gender
addDat$Color = "black"
addDat$Color[predat[, 19]==0]="red"
addDat$Color[predat[, 19]==1]="blue"

# Shapes by gender
addDat$Shape = 21
addDat$Shape[predat[, 19]==0] = 21
addDat$Shape[predat[, 19]==1] = 22

# Set up the normalizing function
normalize <- function(x, mean, sd){(x-mean)/sd}

# Apply the functions to the respective dat columns
for (i in 1:6) {
        col_mean <- mean(dat[, i])
        col_sd <- sd(dat[, i])
        dat[, i] <- normalize(dat[, i], col_mean, col_sd)
```

```
}

# Creating the distance matrix
distances = dist(dat, method="minkowski", p=2)

# Doing MDS on distance matrix
ll <- cmdscale(distances, k=2)

# Plotting the results
plot(ll, col = addDat$Color, pch = addDat$Shape, main = "MDS Results")

# Adding last names to plot
install.packages("wordcloud")
library(wordcloud)
dev.new()
textplot(ll[, 1], ll[, 2], predat[, 4], cex = .7, main = "MDS Results w/ names")

# Scaled to draw away from outliers
dev.new()
plot(ll, xlim = c(-1.5,1), col = addDat$Color, pch = addDat$Shape, ylim = c(-.5,1), main = "Zoomed MDS Results")
dev.new()
textplot(ll[, 1], ll[, 2], predat[, 5], cex=.7, xlim = c(-1.5,1), ylim = c(-.5,1), main = "Zoomed MDS Results w/ names")

# Checking goodness of fit
cmdscale(distances, k=2, eig = TRUE)$GOF
cmdscale(distances, k=1, eig = TRUE)$GOF

# Plotting original distance vs our MDS distances
dev.new()
plot(distances, dist(ll), main = "Original distances vs MDS distances")

# Plotting goodness of fit vs dimensions for our data
gof <- numeric(10)
for (i in 1:10) {
        gof[i] <- cmdscale(distances, k = i, eig = TRUE)$GOF[1]
}
dev.new()
plot(1:10, gof, xlab = "Dimension", ylab = "GOF", main = "Goodness of fit vs Dimension for actors/actresses")
```

Appendix B. R Code for Variation I (Discounting Career Length)

```
# Take in the data
predat <- read.csv(file.choose(), header = TRUE)

# Initialize data matrix with columns we want to include
dat <- matrix(nrow=200, ncol=6)
dat[, 1] <- predat[, 5]        # Films
dat[, 2] <- predat[, 8]        # DirectorCred
dat[, 3] <- predat[, 9]        # ProducerCred
dat[, 4] <- predat[, 12]# ProfAwardNom
dat[, 5] <- predat[, 13]# ProfAwardWin
dat[, 6] <- predat[, 17]# Roles

# Set up the normalizing function
normalize <- function(x, mean, sd){(x-mean)/sd}
```

```
# Apply the functions to the respective dat columns
for (i in 1:6) {
	col_mean <- mean(dat[, i])
	col_sd <- sd(dat[, i])
	dat[, i] <- normalize(dat[, i], col_mean, col_sd)
}

# Creating the distance matrix
distances = dist(dat, method="minkowski", p=2)

# Doing MDS on distance matrix
ll <- cmdscale(distances, k=2)

# Plotting the results
plot(ll, main = "MDS Results")

# Adding last names to plot
install.packages("wordcloud")
library(wordcloud)
dev.new()
textplot(ll[, 1], ll[, 2], predat[, 4], cex = .7, main = "MDS Results w/ names")

# Zoomed results
dev.new()
plot(ll, xlim = c(-1.5, 1.5), ylim = c(-1, 1), main = "Zoomed MDS Results")
dev.new()
textplot(ll[, 1], ll[, 2], predat[, 4], xlim = c(-1.5, 1.5), ylim = c(-1, 1), cex = .6, main = "Zoomed MDS Results w/ names")



# Checking goodness of fit
cmdscale(distances, k=2, eig = TRUE)$GOF
cmdscale(distances, k=1, eig = TRUE)$GOF

# Plotting original distance vs our MDS distances
dev.new()
plot(distances, dist(ll), main = "Original distances vs MDS distances")

# Plotting goodness of fit vs dimensions for our data
gof <- numeric(10)
for (i in 1:10) {
	gof[i] <- cmdscale(distances, k = i, eig = TRUE)$GOF[1]
}
dev.new()
plot(1:10, gof, xlab = "Dimension", ylab = "GOF", main = "Goodness of fit vs Dimension for actors/actresses")



x <- 1:6
y <- numeric(6)

# Creating the distance matrix
for (i in 1:6) {
	distances = dist(dat, method="minkowski", p=i)
	y[i] <- cmdscale(distances, k = 2, eig = TRUE)$GOF[1]
}
dev.new()
plot(x, y, xlab = "p for Minkowski", ylab = "GOF", main = "P vs GOF for 2-D MDS")
```

Apendix C. R Code for Variation II (Oscar Winners)

```
# Take in the data
predat <- read.csv(file.choose(), header = TRUE)

# Initialize data matrix with columns we want to include
dat <- matrix(nrow=50, ncol=7)
dat[, 1] <- predat[, 5]/predat[, 16]    # Films/Career_length
dat[, 2] <- predat[, 8]/predat[, 16]    # DirectorCred/Career_length
dat[, 3] <- predat[, 9]/predat[, 16]    # ProducerCred/Career_length
dat[, 4] <- predat[, 12]/predat[, 16]  # ProfAwardNom/Career_length
dat[, 5] <- predat[, 13]/predat[, 16]  # ProfAwardWin/Career_length
dat[, 6] <- predat[, 17]/predat[, 16]  # Net_worth/Career_length
dat[, 7] <- predat[, 18]/predat[, 16]  # Followers/Career_length

# Set up the normalizing function
normalize <- function(x, mean, sd){(x-mean)/sd}

# Apply the functions to the respective dat columns
for (i in 1:7) {
        col_mean <- mean(dat[, i])
        col_sd <- sd(dat[, i])
        dat[, i] <- normalize(dat[, i], col_mean, col_sd)
}

# Creating the distance matrix
distances = dist(dat, method="minkowski", p=2)

# Doing MDS on distance matrix
ll <- cmdscale(distances, k=2)

# Plotting the results
plot(ll, main = "Oscar MDS Results")

# Adding last names to plot
install.packages("wordcloud")
library(wordcloud)
dev.new()
textplot(ll[, 1], ll[, 2], predat[, 4], cex = .7, main = "Oscar MDS Results w/ names")

# Checking goodness of fit
cmdscale(distances, k=2, eig = TRUE)$GOF
cmdscale(distances, k=1, eig = TRUE)$GOF

# Plotting original distance vs our MDS distances
dev.new()
plot(distances, dist(ll), main = "Original distances vs Oscar MDS distances")

# Plotting goodness of fit vs dimensions for our data
gof <- numeric(10)
for (i in 1:10) {
        gof[i] <- cmdscale(distances, k = i, eig = TRUE)$GOF[1]
}
dev.new()
```

```
plot(1:10, gof, xlab = "Dimension", ylab = "GOF", main = "GOF vs Dimension for Oscar actors/actresses")


x <- 1:6
y <- numeric(6)

# Creating the distance matrix
for (i in 1:6) {
	distances = dist(dat, method="minkowski", p=i)
	y[i] <- cmdscale(distances, k = 2, eig = TRUE)$GOF[1]
}
dev.new()
plot(x, y, xlab = "p for Minkowski", ylab = "GOF", main = "P vs GOF for 2-D Oscar MDS")
```

Appendix D. R Code for Variation III (Excluding Director/Producer Credits)

```
# Take in the data
predat <- read.csv(file.choose(), header = TRUE)

# Initialize data matrix with columns we want to include
dat <- matrix(nrow=200, ncol=4)
dat[, 1] <- predat[, 5]/predat[, 16]    # Films/Career_length
dat[, 2] <- predat[, 12]/predat[, 16]  # ProfAwardNom/Career_length
dat[, 3] <- predat[, 13]/predat[, 16]  # ProfAwardWin/Career_length
dat[, 4] <- predat[, 18]/predat[, 16]  # Roles/Career_length

# Set up the normalizing function
normalize <- function(x, mean, sd){(x-mean)/sd}

# Apply the functions to the respective dat columns
for (i in 1:4) {
	col_mean <- mean(dat[, i])
	col_sd <- sd(dat[, i])
	dat[, i] <- normalize(dat[, i], col_mean, col_sd)
}

# Creating the distance matrix
distances = dist(dat, method="minkowski", p=2)

# Doing MDS on distance matrix
ll <- cmdscale(distances, k=2)

# Plotting the results
plot(ll, main = "MDS Results")

# Adding last names to plot
install.packages("wordcloud")
library(wordcloud)
dev.new()
textplot(ll[, 1], ll[, 2], predat[, 4], cex = .7, main = "MDS Results w/ names")

# Scaled to draw away from outliers
dev.new()
plot(ll, xlim = c(-1.5,1), ylim = c(-.5,1), main = "Zoomed MDS Results")
dev.new()
```

```
textplot(ll[, 1], ll[, 2], predat[, 5], cex=.7, xlim = c(-1.5,1), ylim = c(-.5,1), main = "Zoomed MDS Results w/ names")

# Checking goodness of fit
cmdscale(distances, k=2, eig = TRUE)$GOF
cmdscale(distances, k=1, eig = TRUE)$GOF

# Plotting original distance vs our MDS distances
dev.new()
plot(distances, dist(ll), main = "Original distances vs MDS distances")

# Plotting goodness of fit vs dimensions for our data
gof <- numeric(10)
for (i in 1:10) {
        gof[i] <- cmdscale(distances, k = i, eig = TRUE)$GOF[1]
}
dev.new()
plot(1:10, gof, xlab = "Dimension", ylab = "GOF", main = "Goodness of fit vs Dimension for actors/actresses")
```

Appendix E. R Code for P vs GOF Plot

```
# Take in the data
predat <- read.csv(file.choose(), header = TRUE)

# Initialize data matrix with columns we want to include
dat <- matrix(nrow=200, ncol=6)
dat[, 1] <- predat[, 5]/predat[, 16]    # Films/Career_length
dat[, 2] <- predat[, 8]/predat[, 16]    # DirectorCred/Career_length
dat[, 3] <- predat[, 9]/predat[, 16]    # ProducerCred/Career_length
dat[, 4] <- predat[, 12]/predat[, 16]  # ProfAwardNom/Career_length
dat[, 5] <- predat[, 13]/predat[, 16]  # ProfAwardWin/Career_length
dat[, 6] <- predat[, 18]/predat[, 16]  # Roles/Career_length

# Set up the normalizing function
normalize <- function(x, mean, sd){(x-mean)/sd}

# Apply the functions to the respective dat columns
for (i in 1:6) {
        col_mean <- mean(dat[, i])
        col_sd <- sd(dat[, i])
        dat[, i] <- normalize(dat[, i], col_mean, col_sd)
}

x <- 1:6
y <- numeric(6)

# Creating the distance matrix
for (i in 1:6) {
        distances = dist(dat, method="minkowski", p=i)
        y[i] <- cmdscale(distances, k = 2, eig = TRUE)$GOF[1]
}
plot(x, y, xlab = "p for Minkowski", ylab = "GOF", main = "P vs GOF for 2-D MDS")
```

Appendix F. HTML/PHP Code for Querying Database for number of films

```
<!DOCTYPE html>
<html>
	<head>
		<title>Project 2 test code</title>
		<meta charset="utf-8" />
	</head>
	<?php
		$db = new PDO("mysql:dbname=imdb;host=localhost;charset=utf8", "sapot", "pkIjut6Q9d");
		$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
		#first we want a list of 200 people
		$myFile = fopen('oscarlist.txt', 'r') or die("Fail to open name list");?>
		<!--build the table for id, movies, rankings for each, geners-->
		<h1>Table of actors</h1>
		<table>
			<tr>
				<th>ID</th>
				<th>roles</th>
				<th>movies</th>
				<th>ranking</th>
				<th>genre</th>
			</tr>
			<?=produceTable()?>
		</table>
</html> <?php
function produceTable() {
	$db = new PDO("mysql:dbname=imdb;host=localhost;charset=utf8", "sapot", "pkIjut6Q9d");
	#echo $myFile;
	$lines = file('oscarlist.txt');
  foreach($lines as $line) {
    $query = "SELECT film_count FROM actors WHERE id = $line";
    $result = $db -> query($query);
    foreach($result as $row) {
      echo $row["film_count"]."<br>";
    }
  }
}
  ?>
```

Appendix G. PHP Code for length of career

```
<!DOCTYPE html>
<html>
	<head>
		<title>Project 2 test code</title>
		<meta charset="utf-8" />
	</head>
	<?php
		$db = new PDO("mysql:dbname=imdb;host=localhost;charset=utf8", "sapot", "pkIjut6Q9d");
		$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
		#first we want a list of 200 people
		$myFile = fopen('oscarlist.txt', 'r') or die("Fail to open name list");?>
		<!--build the table for id, movies, rankings for each, geners-->
		<h1>Table of actors</h1>
		<table>
			<tr>
				<th>ID</th>
				<th>roles</th>
```

```
                                <th>movies</th>
                                <th>ranking</th>
                                <th>genre</th>
                        </tr>
                        <?=produceTable()?>
                </table>
</html> <?php
function produceTable() {
        $db = new PDO("mysql:dbname=imdb;host=localhost;charset=utf8", "sapot", "pkIjut6Q9d");
        #echo $myFile;
        $lines = file('oscarlist.txt');
        foreach($lines as $line) {
                #echo $line;
                #echo '<br>';
                $query1 = "SELECT * from roles r
JOIN movies m on r.movie_id = m.id
JOIN actors a on r.actor_id = a.id
WHERE a.id = $line
ORDER BY year asc;";


$query2 = "SELECT * from roles r
JOIN movies m on r.movie_id = m.id
JOIN actors a on r.actor_id = a.id
WHERE a.id = $line
ORDER BY year desc;";
                #echo $query;
                #initializations
                $years = 0;
                $max = 0;
                $min = 0;

                #query for movies in ascending order
                $rows = $db -> query($query1);
                #echo "number of rows ".$count."<br>";
                $current = 1;

                foreach ($rows as $row) {
                                $min = $row["year"];
                                if($min != 0) {
                                        break;
                                }
                }

                #query db for movies in descending order
                $current = 1;
                $rows2 = $db -> query($query2);
                foreach ($rows2 as $row) {
                        $max = $row["year"];
                        if($max != 0) {
                                break;
                        }
                }
                $years = $max - $min;
                echo $years."<br>";
```

```
                if(abs($years) > 1000) {
                        echo $line." ".$current." vs ".$count." ".$years."<br>";
                }
        }
        #$comps = explode($string);
        #print_r(sizeof($comps));
        #echo $string;

}?>
```

Appendix H. Data for Original, I, and III Variations

#	ID	FirstName	LastName	#Films	BirthYear	Age	DirectorCr	ProducerC	SoundCre	SelfCred	ProfAwar	ProfAwar	OtherAwa	OtherAwa	Length of	# of genre	# of roles	Gender
1	2941019	Christine	Schorn	139	1944	72	0	0	0	3	0	6	0	0	51	10	102	0
2	1153151	Adolfo	Marsillach	100	1928-2002	74	6	0	39	0	2	10	0	0	62	16	54	1
3	1948226	Danny (V)	Webb	156	1958	58	0	0	0	3	0	0	0	0	36	22	81	1
⋮																		
198	3055426	Claire	Trevor	123	1910	90	0	0	4	23	1	1	3	3	78	22	92	0
199	1982220	Dave	Willock	211	1909	81	0	0	1	6	0	0	0	1	60	22	131	1
200	1777683	Bj?rn	Sundquist	103	1948	68	0	0	0	2	0	0	3	6	35	18	80	1

Appendix I. Data for Variation II

#	ID	FirstName	LastName	#Films	BirthYear	Age	DirectorCred	ProducerCred	SoundCred	SelfCred	ProfAwardNom	ProfAward Won	OtherAwardNom	OtherAwardWin	Length of Career in years	Net worth (millions)	ok followers (millions)
1	463327	Leonardo	DiCaprio	228	1974	42	0	17	1	156	52	34	44	140	24	245	14.8
2	2601668	Brie	Larson	62	1989	27	2	0	6	45	27	47	6	20	16	2	0.104
3	1592499	Mark	Rylance	44	1960	56	0	0	0	27	19	15	10	0	30	5	0.002
4	3086854	Alicia	Vikander	31	1988	28	0	0	0	31	26	34	13	17	13	4	0.008
5	1512168	Eddie	Redmayne	44	1982	34	0	0	2	57	42	13	24	6	9	4	0.046
⋮																	
47	3152847	Catherine	Zeta-Jones	173	1969	47	0	0	12	109	23	11	8	7	25	45	0.0357
48	1942243	Denzel	Washington	248	1954	62	3	7	3	201	68	64	33	2	37	190	0.056
49	2375932	Brenda	Fricker	99	1945	71	0	0	0	11	12	4	0	0	40	0	0
50	802341	Dustin	Hoffman	340	1937	79	2	7	5	224	41	54	5	1	45	50	0.236