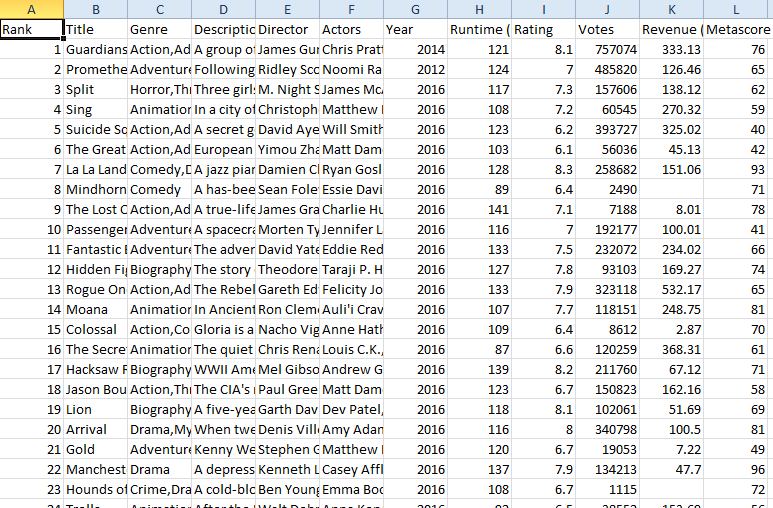
I used a data set of 1000 most popular movies on IMDB in the last 10 years. I downloaded the csv file from kaggle.com. The data fields included are: Rank, Title, Genre, Description, Director, Actors, Year, Runtime, Rating, Votes, Revenue and meta-score (Scores given by critics).

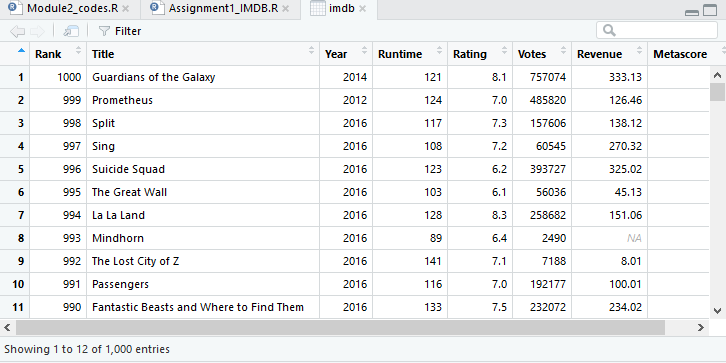


I made the below modifications in the file to keep the calculation accurate.

1. Removed few columns like Genre, Description, Director and Actors which are not needed for this calculation.
2. Flipped the Rank column in descending order so that best movie will show the highest number in the Rank column.

Let me load this file into R studio,

imdb<-read.csv("C:/ IMDB-Movie-Data.csv")



In order to bind only the below numeric columns, I used the below code which would help me to perform dimension reduction accordingly in the later steps.

1. Rank
2. Year
3. Runtime
4. Rating
5. Votes
6. Revenue
7. Metascore

X<-cbind(imdb$Rank, imdb$Year, imdb$Runtime, imdb$Rating, imdb$Votes, imdb$Revenue, imdb$Metascore)

By looking at these 7 components, I would say Rating, Votes, Revenue and Metascore are all co-related components and so can be reduced into fewer dimensions. Let me apply the princomp() function and do the principal component Analysis.

pca<-princomp(X, scores=TRUE, cor=TRUE)

Oops… there was an error.

Error in cov.wt(z) : 'x' must contain finite values only

Because few rows in the dataset had no value in the last two columns. So, I used the below code to get rid of those rows.

pca<-princomp(na.omit(X), scores=TRUE, cor=TRUE)

I executed the below code to view the standard deviation and variance values of these 7 components.

summary(pca)

|  |
| --- |
| Importance of components:  Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7  Standard deviation 1.6667318 1.1538438 1.0601504 0.8561357 0.7025444 0.5670358 0.46761554  Proportion of Variance 0.3968564 0.1901936 0.1605598 0.1047098 0.0705098 0.0459328 0.03123776  Cumulative Proportion 0.3968564 0.5870500 0.7476099 0.8523196 0.9228294 0.9687622 1.00000000 |
|  |
| |  | | --- | | > | |

I loaded the principal components with the below code.

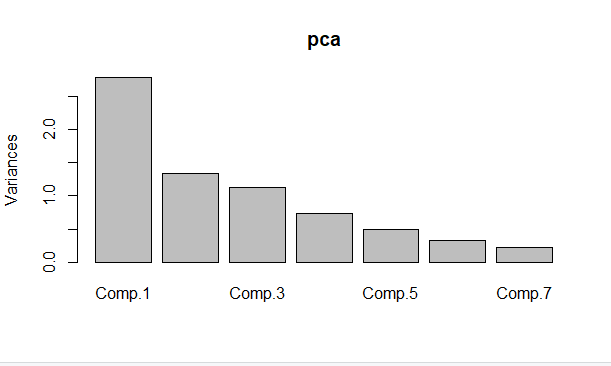
loadings(pca)

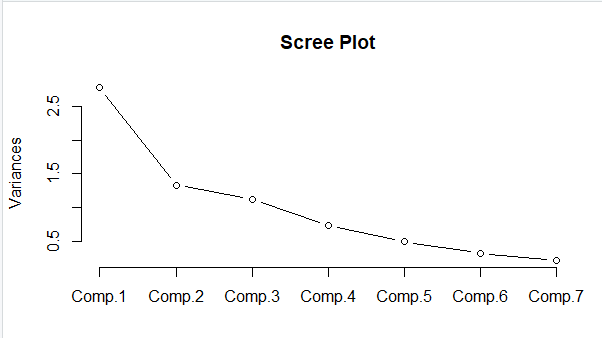
|  |
| --- |
| Loadings:  Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7  [1,] 0.276 0.611 0.227 0.656 0.199 0.158  [2,] -0.155 0.760 -0.497 -0.289 -0.257  [3,] 0.368 0.105 -0.894 -0.179 0.139  [4,] 0.466 -0.454 -0.599 0.464  [5,] 0.505 -0.188 0.253 0.172 0.126 -0.337 -0.697  [6,] 0.381 0.556 0.334 -0.489 0.191 0.394  [7,] 0.383 -0.598 0.229 -0.188 0.590 -0.226  Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7  SS loadings 1.000 1.000 1.000 1.000 1.000 1.000 1.000  Proportion Var 0.143 0.143 0.143 0.143 0.143 0.143 0.143  Cumulative Var 0.143 0.286 0.429 0.571 0.714 0.857 1.000 |
|  |
| |  | | --- | | > | |

I used plot and screeplot functions to view the variance among the 7 components.

plot(pca)

screeplot(pca, type="line", main="Scree Plot")

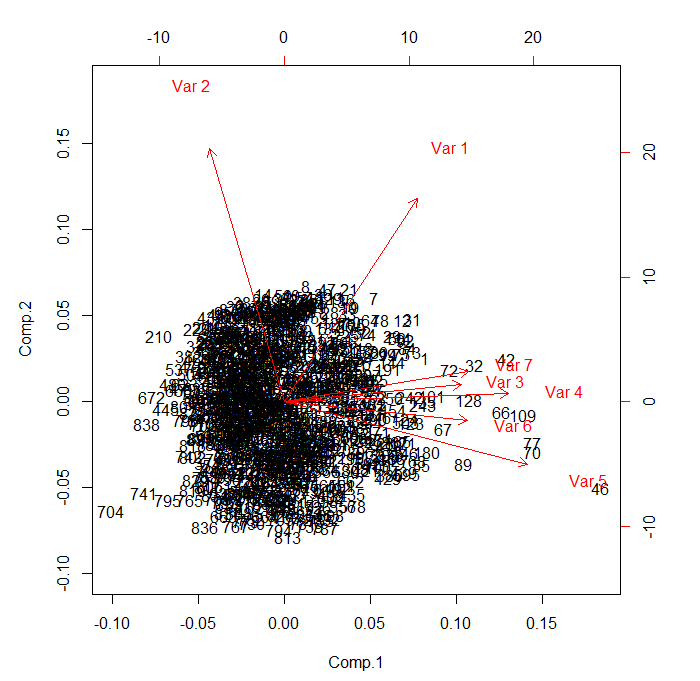




I could notice the graph flattening from the 4th component which are nothing but the co-related components - Rating, Votes, Revenue and Metascore which I discussed above.

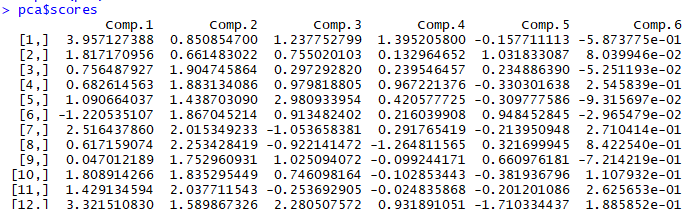
I used biplot function to view the direction and strength of eigenvectors.

biplot(pca)



Scores for each observation are given below:

pca$scores



**My inferences on this PCA analysis are as follows:**

1. The components Var 3 to Var 7 are all co-related and could be represented as one dimension or principal component representing whether the movie is a good or bad movie. That means, the runtime of the movie, rating, votes, revenue and metascore of a movie are all directly proportional and correlated to each other. I am surprised and excited to see that the long running movies are somehow good movies w.r.t this dataset.
2. The components Var 4 (Ratings) and Var 5 (Votes) are almost same direction and value. Based on this, I would say, if a movie gets more votes, then the rating will be higher. This could be due to users who liked the movie tend to vote much.
3. Var 2 (Year of the movie) has no relation to other components. That means, year of the movie cannot decide if a movie is good or bad. So, it cannot be considered for dimension reduction.