PROJET R

Joseph BEASSE, Tristan GONÇALVES, Andrea MIELE

Quels indicateurs peuvent expliquer le succès d'un film ou d'une série ?

I - Introduction

Depuis quelques années, nous pouvons observer une généralisation des plateformes de streaming dans les foyers. En effet, Netflix est par exemple passé de 21 à 221 millions d'abonnés en l'espace de 10 ans [1]. Nous pouvons également noter une augmentation du nombre de plateformes depuis ces dernières années.

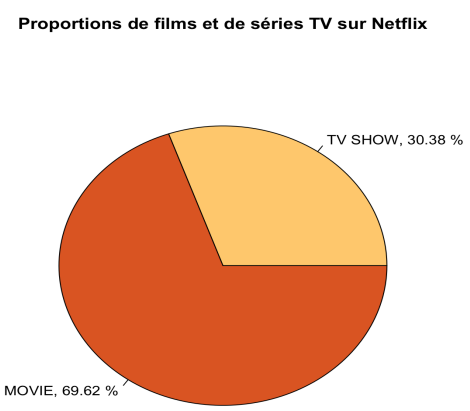
Avec ces fortes augmentations, il nous semble intéressant de nous questionner sur le contenu offert par ces différentes plateformes, afin de les comparer sur leur catalogue, et peut-être aussi nous permettre d'étudier les types de films/séries qui attirent certaines catégories d'âge.

Pour ce faire, nous avons récupéré des jeux de données des trois principales entreprises sur le marché du streaming, à savoir Netflix, Amazon Prime et Disney+.

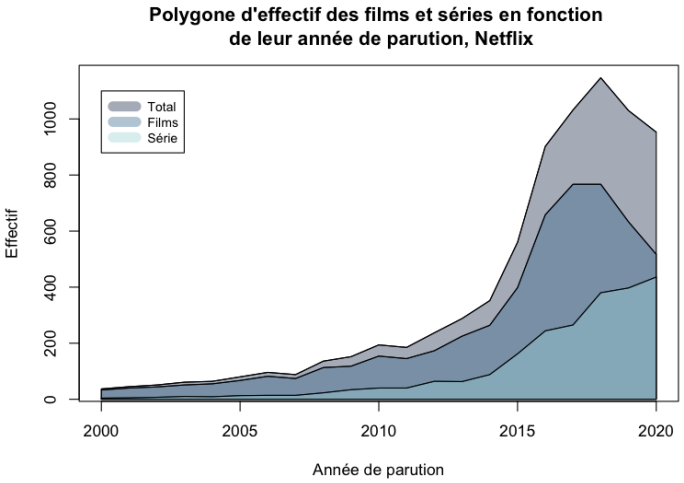
En outre, il nous paraissait intéressant d'étudier l'industrie du cinéma avec un plus grand angle, c'est pourquoi nous avons également sélectionné un autre jeu de données, comportant des informations plus chiffrées sur différents aspects de la production d'un film.
Les jeux de données sont disponibles dans cette archive : **datasets.rar**

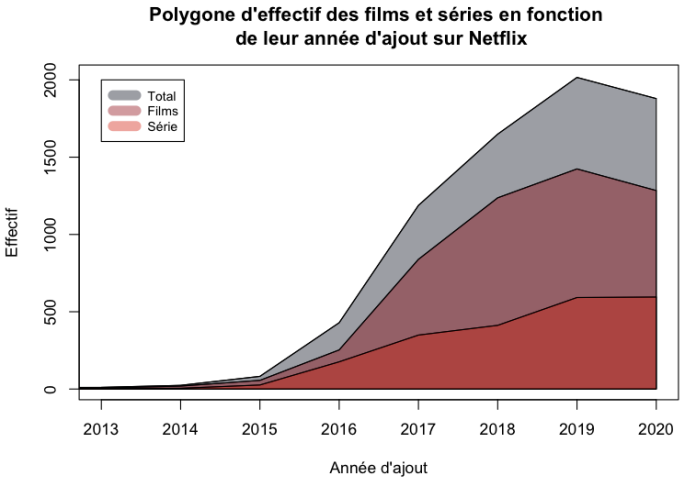
II - Statistiques descriptives

Nous avons d'abord étudié, grâce aux statistiques descriptives, la part de films et de séries pour les deux plateformes de streaming que sont Netflix et Amazon Prime.

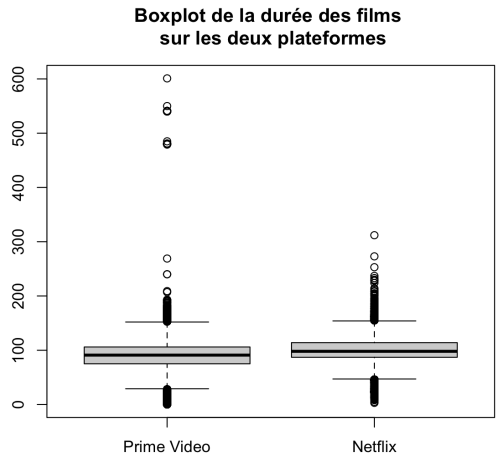


Les diagrammes présentés ci-dessus sont très similaires, avec une majorité de films (~75%) pour chacune des plateformes, et une minorité de séries et show télévisés (~25%). Suite à cette distribution, nous voulons alors étudier la hausse des effectifs au cours des années pour en déduire une tendance.
Comme le montre le graphique ci-dessous, de 2016 à 2022, le nombre de total de films et séries présents sur Netflix à connu une augmentation de 400%. Cela concorde avec le fait qu'en 2016 Netflix décide de passer d'une production dans les pays anglo-saxons à une production mondiale.

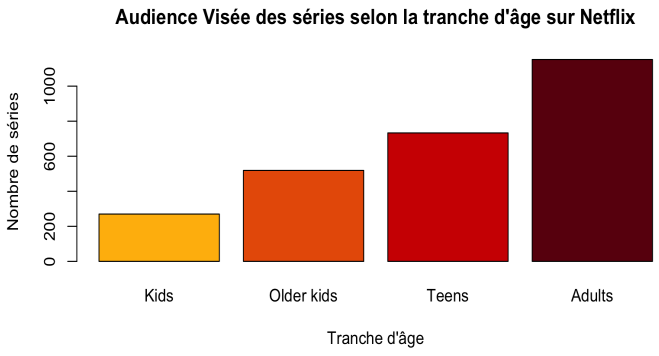


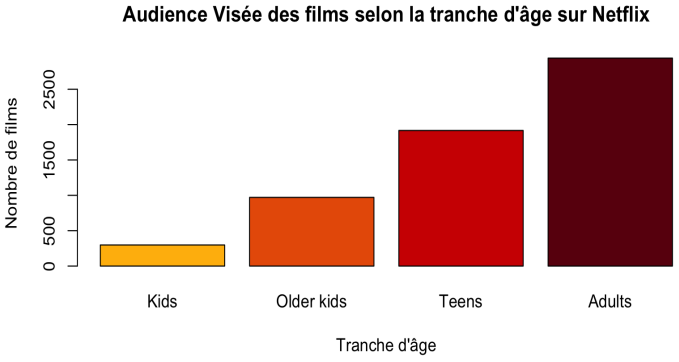


Les deux polygones montrés précédemment ont de fortes similitudes, on remarque une grande augmentation des œuvres cinématographiques vers 2015 qui coïncide entre l'année de parution des films et leur année d'ajout sur netflix. On peut également noter que Netflix a choisi d'importer plus de films que de séries sur leur catalogue et cette tendance a l'air de se confirmer au fil des années. Cependant, la courbe de parutions des séries monte fortement tandis que celle des films diminue. On peut supposer que la tendance citée précédemment s'inverse naturellement dans les prochaines années.

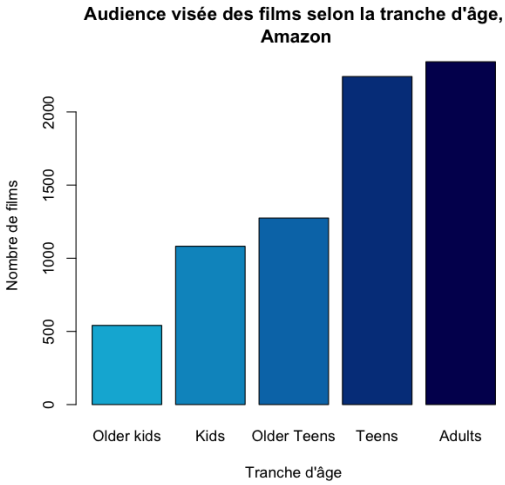


Comme nous pouvons le voir ci-dessus, Prime Video possède des valeurs extrêmes bien plus éloignées que Netflix. Après recherche, ces films de plus de 500 heures sont des films à vocation de détente ou de sommeil, par exemple 9 heures de bol chantant himalayen ou encore 9 heures d'écran noir pour la détente. Tous ces films sont du même réalisateur : Mark Knight. Si on les laisse "de côté", on peut voir que les durées des films pour Prime Video et Netflix sont sensiblement identiques, avec une moyenne tournant autour des 100 minutes.





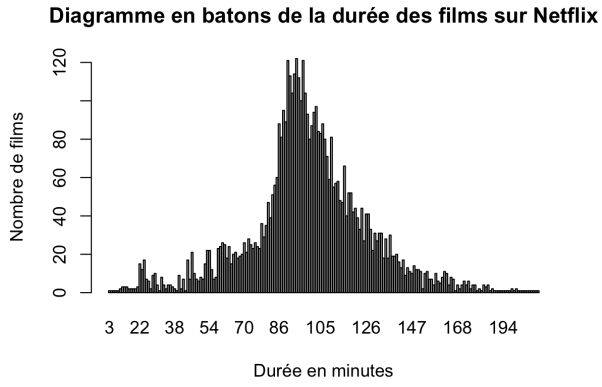
Par la suite, on a analysé l'audience visée par les séries
et les films sur le catalogue Netflix en fonction de leur nombre. On note que plus l'individu est âgé, plus le catalogue est fourni en séries et films pouvant potentiellement lui correspondre. Cela peut s'expliquer par le fait que Netflix est un service payant auquel peu d'enfants ou adolescents peuvent avoir accès sans l'aide de leurs parents.



Pour Amazon, on peut voir une plus grande précision dans la séparation des catégories d'âge, avec l'apparition de "Older Teens" correspondant aux adolescents de plus de 17 ans. La distribution ressemble à celle de Netflix, néanmoins on peut aisément dire que Amazon Prime diffuse en proportion moins de séries pour adultes que son concurrent direct.

III - Statistiques inférentielles

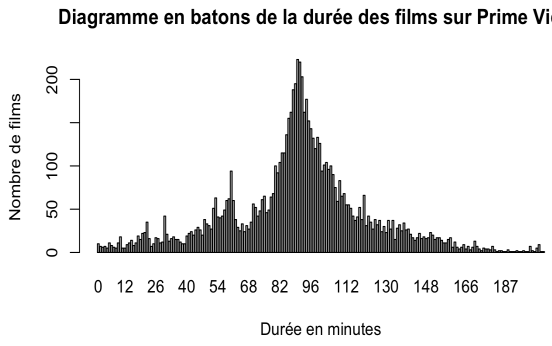
Afin de comparer les durées des films entre les deux plateformes, nous avons testé la normalité des deux échantillons grâce à un test de Shapiro-Wilk.




```r
ShapiroTest <-
netflix_movie[order(netflix_movie$duration),]
ShapiroTest <- head(ShapiroTest,-3)
shuffled_data  =  ShapiroTest[sample(1:nrow(ShapiroTest)),]
shuffled_data <- head(shuffled_data,-1200)
barplot(table(shuffled_data$duration))
# On s'attend à une loi distribuée normalement au vu du
graphique
shapiro.test(shuffled_data$duration)

        #Shapiro-Wilk normality test

#data:  shuffled_data$duration
#W = 1, p-value <2e-16
```




```r
barplot(table(amazon_movie$duration), xlab="Durée en
minutes", ylab="Nombre de films", main="Diagramme en batons
de la durée des films sur Prime Video")
shuffled_data2  =
 amazon_movie[sample(1:nrow(amazon_movie)),]
shuffled_data2 <- head(shuffled_data2,-3000)
shapiro.test(shuffled_data2$duration)

        Shapiro-Wilk normality test

data:  shuffled_data2$duration
W = 0.7, p-value <2e-16
```

Dans les deux tests, la p-value obtenue est à chaque fois inférieure à 2.10^{-16}. Ce pourcentage étant en dessous du risque de première espèce à 5%, nous ne pouvons considérer l'hypothèse de normalité valide. Il est donc impossible de réaliser un test de Student sur les deux échantillons.

Les deux datasets sont supposés indépendants (car on ne retrouve pas les mêmes films sur deux plateformes de streaming concurrentes), nous allons donc réaliser un test de Wilcoxon afin de voir si les deux échantillons possèdent la même distribution.

```
wilcox.test(amazon_movie$duration,netflix_movie$duration)

    Wilcoxon rank sum test with continuity correction

data:  amazon_movie$duration and netflix_movie$duration
W = 2e+07, p-value <2e-16
alternative hypothesis: true location shift is not equal to 0
```

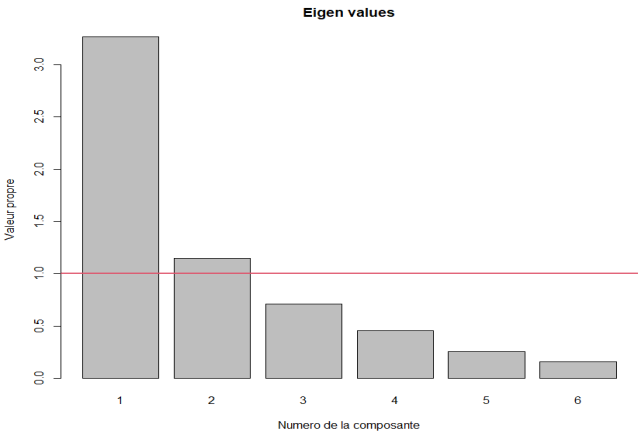
Le test de Wilcoxon nous donne un W très grand, ainsi qu'une p-value très faible (inférieure à 5%) ce qui signifie que la différence entre les deux moyennes de durée des films est très significative. Cela s'observe bien lorsqu'on regarde les deux moyennes, qui sont significativement différentes (100min pour Netflix, 91min pour Prime Vidéo).

IV - Analyse en composantes principales

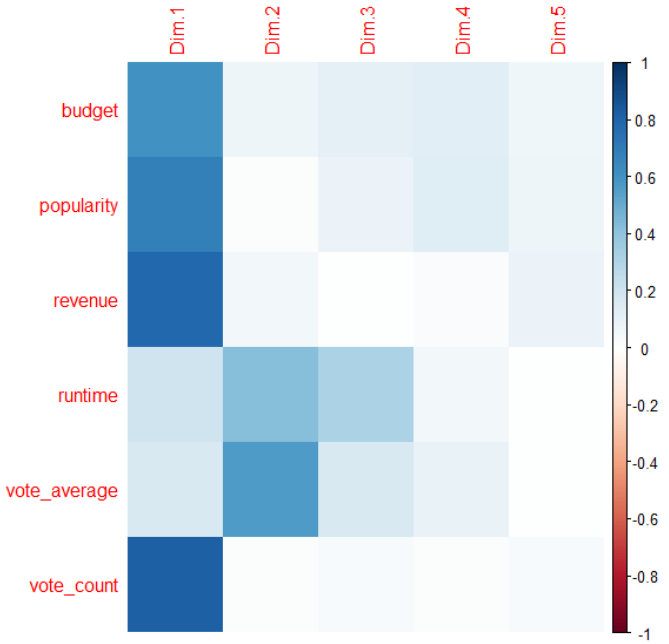
Pour l'analyse en composantes principales, on se base sur un autre dataset comprenant 5000 films notés sur "TheMovieDataBase". On récupère les données numériques intéressantes (budget, popularité, revenu, durée du film, note moyenne et le nombre de votes) auxquels on applique la méthode PCA.

```
ACP <- PCA(data.frame(tmdb_5000_movies[,c(1,9,13,14,19,20)]), graph=FALSE)
round(ACP$eig,digit=2)

       eigenvalue percentage of variance cumulative percentage of variance
comp 1       3.27                  54.4                                54
comp 2       1.15                  19.1                                74
comp 3       0.71                  11.9                                85
comp 4       0.45                   7.6                                93
comp 5       0.26                   4.3                                97
comp 6       0.16                   2.7                               100
```

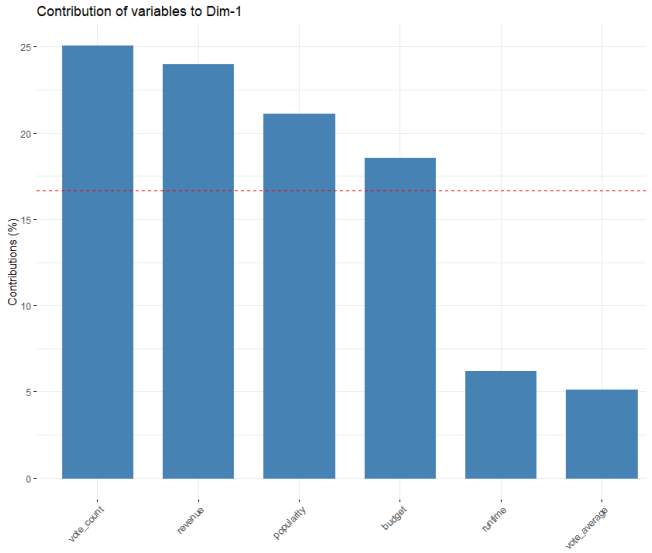


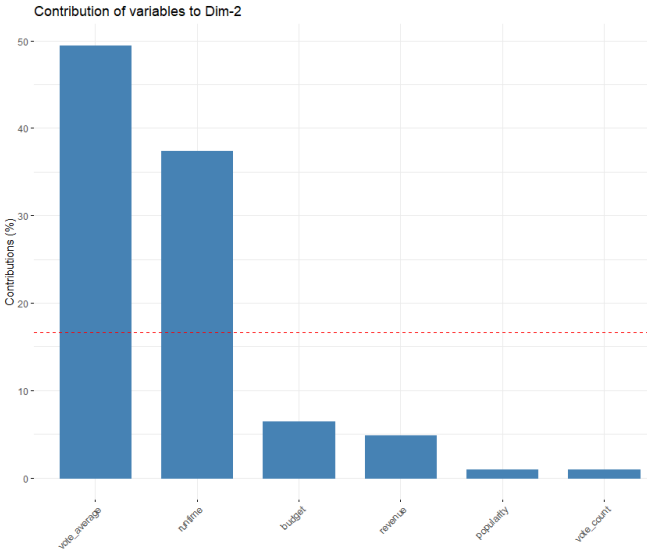
Ci-dessus apparaissent les différentes valeurs propres (*eigenvalues*) associées aux différentes composantes que nous étudions. Etant donné que l'ACP est centrée réduite (la fonction PCA de FactoMineR normalise les données automatiquement), on peut dire que d'après le critère de Kaiser, nous ne retenons pour la suite que les deux premières valeurs propres, car ce sont les seules à être supérieures à 1. De plus, ces deux axes vont respectivement restituer 54,4% et 19,1% de l'information. Un total de 73,5% de la variance totale est donc expliquée par ces deux axes, ce qui est une bonne proportion.



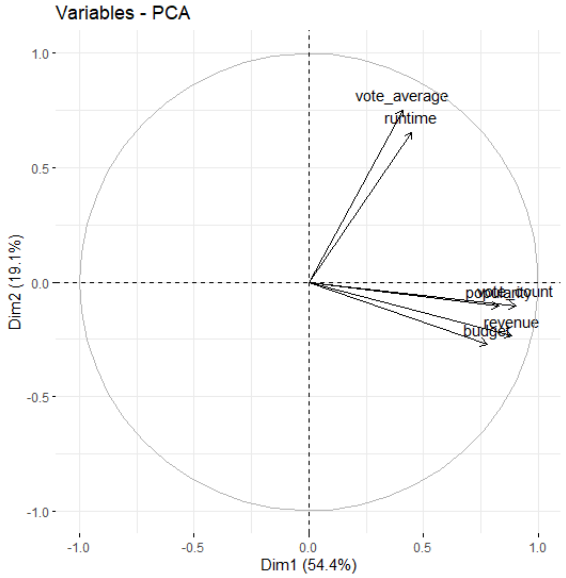

```
ACP$var$cos2
              Dim.1 Dim.2  Dim.3 Dim.4  Dim.5
budget         0.61 0.074 0.1183 0.129 0.0667
popularity     0.69 0.011 0.0803 0.132 0.0706
revenue        0.78 0.056 0.0034 0.021 0.0826
runtime        0.20 0.429 0.3116 0.057 0.0008
vote_average   0.17 0.568 0.1663 0.097 0.0016
vote_count     0.82 0.011 0.0344 0.019 0.0356
```

À gauche, on observe une heatmap reprenant les informations obtenues par les valeurs propres. En effet, en prenant la moyenne des colonnes (dim), on retrouve les 54% représentés par la dimension 1, les 19% représentés par la dimension 2. Cette nouvelle représentation confirme le choix de retenir seulement les deux premières valeurs propres, obtenue par le critère de Kaiser.



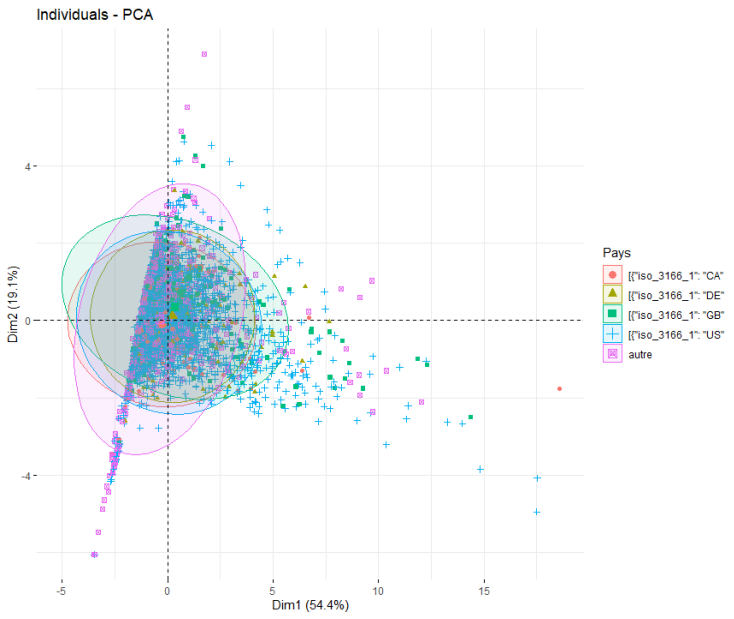


Les deux diagrammes en barres ci-dessus, représentent les pourcentages de contribution aux dimensions 1 et 2 des variables quantitatives mentionnées précédemment. Ceux-ci viennent appuyer sur les données que l'on a présenté. En effet le revenu, la popularité, le budget et la note contribuent le plus à la dimension 1, tandis que pour la dimension 2, il s'agit du nombre de votes et de la durée du film.



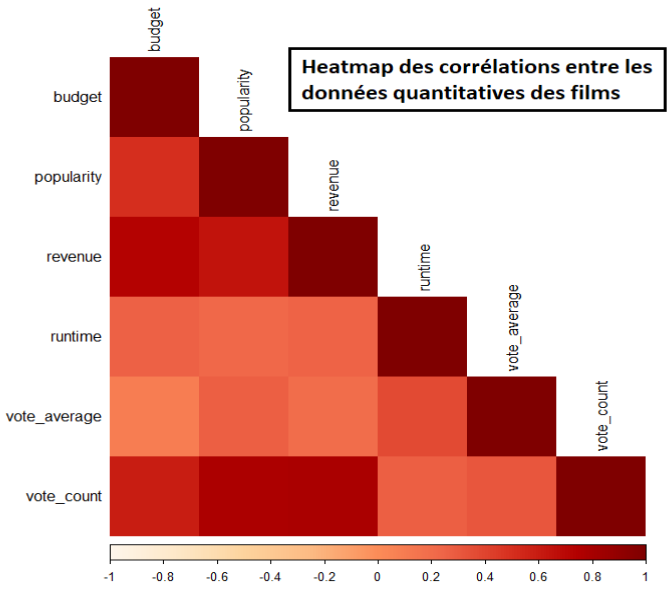
Sur ce cercle des corrélations, on peut voir que l'ensemble des variables sont bien représentées.

Nous voyons aussi que lorsque le temps du film contribue fortement à la moyenne des votes sur le film.
De même, la popularité, le nombre de votes, le revenu ainsi que le budget contribue bien entre eux, d'autant plus fortement pour la popularité avec le nombre de votes (sans surprise), que le budget et le revenu. On a donc de fortes corrélations entre le vote moyen et le temps du film, entre le nombre de votes et la popularité et enfin entre le revenu et le budget.



Sur le graphique ci-contre, on peut voir que les individus sont très concentrés autour du centre.
On observe cependant une ligne composée des films américains et d'autres nationalités, très peu représentés par l'axe numéro 2.
De plus, à l'aide des ellipses, on observe que les films des 4 premières nationalités sont plus concentrés que ceux des autres nationalités. Ils sont ainsi l'air d'être soit mal représentés par l'axe 2, soit mieux représentés par l'axe 2 que l'ensemble des films.
Une autre chose est que l'ellipse rose est tournée perpendiculairement par rapport aux autres ellipses : Ainsi les films de nationalités identifiées varient beaucoup plus selon la dimension 1, tandis que les autres varient beaucoup plus selon la dimension 2.
On peut conclure quant au fait que la nationalité influe peu sur les variables étudiées.

V - Corrélation des indicateurs quantitatifs d'un film



Sur ce graphique, nous pouvons étudier les corrélations entre les différentes variables quantitatives.
Par exemple, plus un film possède un grand budget, plus il gagne d'argent (on parle ici de revenu et non de bénéfices). Cela s'explique par le fait que plus un film a de revenus, plus il a fait d'entrées, et plus il a fait d'entrées, plus il y a de possibles notes sur le site.
Évidemment, il y a une forte corrélation entre la popularité d'un film et les revenus perçus.
Cependant étonnamment, il ne semble pas y avoir de forte corrélation entre la popularité d'un film et sa moyenne notée par les utilisateurs.

On trouve aussi des résultats surprenants : le budget d'un film, ainsi que le revenu ne sont pas corrélés à la note moyenne donnée sur le site. Ainsi, faire un blockbuster ne veut pas forcément dire faire un bon film.

VI - Conclusion

On peut voir que la distribution de la durée des films n'est pas similaire selon les plateformes de streaming, on observe aussi que la proportion de films et de séries n'est pas la même selon les plateformes de streaming. Ainsi on peut noter une différence de positionnement d'offre entre Amazon Prime et Netflix. Ces dernières plateformes ont un public visé différent, cela peut se remarquer en notant une proportion de films pour enfants supérieure sur Prime Video.
Après une analyse en composantes principales réalisée sur un échantillon de 5000 films, on note que la moyenne des notes des utilisateurs semble corrélée à la durée du film. De même, le nombre de votes, la popularité, le revenu ainsi que le budget semblent influer les uns sur les autres.
On trouve aussi des résultats surprenants, comme par exemple le fait que le revenu du film ainsi que son budget soient faiblement corrélés à la note moyenne donnée sur le site. Par conséquent, faire un film à gros budget (blockbuster) ne garantit pas sa qualité. De même, faire beaucoup d'entrées ne veut pas forcément dire que la note moyenne du film sera élevée.
Attention, ces résultats sont à prendre en compte avec beaucoup de précautions : ici est mise en évidence une corrélation et non pas une causalité.

Nous avons conscience que d'autres paramètres entrent en jeu dans la catégorisation d'un film et sa réussite et qu'ainsi notre étude n'est pas la plus complète. En effet, le jeu de données utilisé répertorie les avis provenant uniquement d'un site, qui plus est, anglophone, ce qui n'est pas vraiment représentatif de la population mondiale. Il aurait donc fallu avoir accès à des datasets d'autres sites de notations, comme par exemple Allociné ou IMDB.

Cette étude nous a cependant permis d'apprendre à choisir des données adéquates à une problématique à étudier et à les analyser, même s'il arrive d'obtenir des résultats parfois étonnants.

VII - Annexes

```
# Imports
install.packages("readr")
install.packages("dplyr")
install.packages("stringr")
install.packages("ggplot2")
install.packages("reshape2")
install.packages("PCAmixdata")
install.packages("FactoMineR")
install.packages("factoextra")
install.packages("missMDA")
install.packages("corrplot")

library(PCAmixdata)
library(readr)
library(dplyr)
library(stringr)
library(ggplot2)
library(reshape2)
library(FactoMineR)
library(factoextra)
library(missMDA)
library(corrplot)

netflix_titles <- read_csv("netflix_titles.csv",
                           col_types = cols(date_added = col_date(format = "%B %d, %Y")))

disney_plus_titles <- read_csv("disney_plus_titles.csv",
                               col_types = cols(date_added = col_date(format = "%B %d, %Y")))

amazon_prime_titles <- read_csv("amazon_prime_titles.csv",
                                col_types = cols(date_added = col_date(format = "%B %d, %Y")))

# ANALYSE NETFLIX
# Transformation des durées des films en nombre (en enlevant le " min")
netflix_movie <- subset(netflix_titles,type =="Movie")
netflix_movie$duration <- gsub(" min","",as.character(netflix_movie$duration))
netflix_movie$duration
netflix_movie$duration <- as.double(netflix_movie$duration)
summary(netflix_movie$duration)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
# 3.00   87.00   98.00   99.58  114.00  312.00       3
# Transformation des durées des séries en nombre de saisons (en enlevant le " min")

netflix_serie <- subset(netflix_titles,type =="TV Show")
netflix_serie$duration <- gsub(" Season","",as.character(netflix_serie$duration))
netflix_serie$duration <- gsub("s","",as.character(netflix_serie$duration))
netflix_serie$duration
netflix_serie$duration <- as.double(netflix_serie$duration)
summary(netflix_serie)
#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#   1.000   1.000   1.000   1.765   2.000  17.000

# On crée un camembert
options(digits=2)
percent1 <- length(netflix_serie$duration)
percent2 <- length(netflix_movie$duration)
total = length(netflix_titles$duration)
percents = round(100*c(percent1,percent2)/total,2)

lbl = c(paste('TV SHOW,',percents[1],"%"), paste('MOVIE,',percents[2],"%"))
pie(percents,labels=lbl,main="Proportions de films et de séries TV sur Netflix", col=c('#FFD07F','#E26A2C'))
attach(netflix_movie$release_year)

## --- Analyse des années de parutions
# Créer un dataframe pour chaque type de données : total, films et séries ## Remarques : On enlève 2021 car on n'a pas assez de données
df = data.frame(subset(subset(netflix_movie,release_year <=2020),release_year >=2000))
df2 = data.frame(subset(subset(netflix_serie,release_year <=2020),release_year >=2000))
df3 = data.frame(subset(subset(netflix_titles,release_year <=2020),release_year >=2000))
x1 <- 2000:2020
data <- data.frame(table(df$release_year))
data2 <- data.frame(table(df2$release_year))
data3 <- data.frame(table(df3$release_year))
plot(2000:2020,
     data3[,2],
```

```
     type = "l",
     # Set line type to line
     lwd = 1,
     xlab="Année de parution",
     ylab="Effectif",
     main="Polygone d'effectif des films et séries en fonction de parution")
lines(2000:2020,
      data2[,2],
      type = "l",
      lwd = 1)
lines(2000:2020,
      data[,2],
      type = "l",
      lwd = 1)
polygon(c(2000,2000:2020,2020) ,c(0,data3[,2],0),col = rgb(29, 53, 87,max = 255, alpha = 100))
polygon(c(2000,2000:2020,2020) ,c(0,data[,2],0),col = rgb(69, 123, 157,max = 255, alpha = 100))
polygon(c(2000,2000:2020,2020) ,c(0,data2[,2],0),col = rgb(168, 218, 220,max = 255, alpha = 100))
legend(2000,1100, legend=c( "Total","Films", "Série"),lwd=10,
       col=c(rgb(29, 53, 87,max = 255, alpha = 100), rgb(69, 123, 157,max = 255, alpha = 100), rgb(168, 218, 220,max = 255, alpha = 100)), lty=1:2,
cex=0.8)
netflix_movie$date_added =format(netflix_movie$date_added,'%Y')

## --- Analyse des années d'ajouts
# Créer un dataframe pour chaque type de données : total, films et séries
  ## Remarques : On enlève 2021 car on n'a pas assez de données
netflix_movie$date_added =format(netflix_movie$date_added,'%Y')
netflix_serie$date_added =format(netflix_serie$date_added,'%Y')
netflix_titles$date_added =format(netflix_titles$date_added,'%Y')

df = data.frame(subset(subset(netflix_movie,date_added <=2020),date_added >=2013))
df2 = data.frame(subset(subset(netflix_serie,date_added <=2020),date_added >=2013))
df3 = data.frame(subset(subset(netflix_titles,date_added <=2020),date_added >=2013))


data <- data.frame(table(df$date_added))
data2 <- data.frame(table(df2$date_added))
data3 <- data.frame(table(df3$date_added))

x1 <- 2013:2020
plot(x1,
     data3[,2],
     type = "l",
     lwd = 1,
     xlab="Année de parution",
     ylab="Effectif",
     main="Polygone d'effectif des films et séries en fonction de leur année d'ajout")
lines(x1,
      data2[,2],
      type = "l",
      lwd = 1)
lines(x1,
      data[,2],
      type = "l",
      lwd = 1)
polygon(c(2000,x1,2020) ,c(0,data3[,2],0),col = rgb(29, 53, 87,max = 255, alpha = 100))
polygon(c(2000,x1,2020) ,c(0,data[,2],0),col = rgb(69, 123, 157,max = 255, alpha = 100))
polygon(c(2000,x1,2020) ,c(0,data2[,2],0),col = rgb(168, 218, 220,max = 255, alpha = 100))
legend(2013,2000, legend=c( "Total","Films", "Série"),lwd=10,
       col=c(rgb(29, 53, 87,max = 255, alpha = 100), rgb(69, 123, 157,max = 255, alpha = 100), rgb(168, 218, 220,max = 255, alpha = 100)), lty=1:2,
cex=0.8)

hist(table(netflix_serie$rating))
netflix_serie$rating <- gsub("84 min","",as.character(netflix_serie$rating))
netflix_serie$rating <- gsub("74 min","",as.character(netflix_serie$rating))
netflix_serie$rating <- gsub("84 min","",as.character(netflix_serie$rating))
table(netflix_serie$rating)
barplot(table(netflix_serie$rating), main="Répartition des séries sur Netflix en fonction des ratings")

hist(table(netflix_movie$rating))
netflix_movie$rating <- gsub("66 min","",as.character(netflix_movie$rating))
netflix_movie$rating <- gsub("84 min","",as.character(netflix_movie$rating))
netflix_movie$rating <- gsub("74 min","",as.character(netflix_movie$rating))
table(netflix_movie$rating)
barplot(table(netflix_movie$rating), main="Répartition des films sur Netflix en fonction des ratings")

#Audience visée des films

netflixPublicCibleFilms <- netflix_movie
netflixPublicCibleFilms$rating <- gsub("TV-14|PG-13","Teens",as.character(netflixPublicCibleFilms$rating))
netflixPublicCibleFilms$rating <- gsub("TV-PG|PG|TV-Y7-FV|TV-Y7","Older kids",as.character(netflixPublicCibleFilms$rating))
netflixPublicCibleFilms$rating <- gsub("TV-Y|TV-G|G","Kids",as.character(netflixPublicCibleFilms$rating))
```

```r
netflixPublicCibleFilms$rating <- gsub("TV-MA|NR|UR|NC-17|R","Adults",as.character(netflixPublicCibleFilms$rating))

# Catégories uniques -> dans les films proposés, netflix cherche à avoir un catalogue diversifié, ayant des catégories pour tous les types de
publics
unique(na.omit(autre$rating))

barplot(main="Audience Visée des films selon la tranche d'âge",xlab="Tranche d'âge", ylab="Nombre de films",
tail(sort(table(autre$rating),decreasing=FALSE),n=4))

#Audience visée des séries

autreSerie <- netflix_serie
autreSerie$rating <- gsub("TV-14|PG-13","Teens",as.character(autreSerie$rating))
autreSerie$rating <- gsub("TV-PG|PG|TV-Y7-FV|TV-Y7","Older kids",as.character(autreSerie$rating))
autreSerie$rating <- gsub("TV-Y|TV-G|G","Kids",as.character(autreSerie$rating))
autreSerie$rating <- gsub("TV-MA|NR|UR|NC-17|R","Adults",as.character(autreSerie$rating))

# Catégories uniques -> dans les séries aussi, netflix cherche à avoir un catalogue diversifié

unique(na.omit(autreSerie$rating))
barplot(main="Audience Visée des séries selon la tranche d'âge",xlab="Tranche d'âge", ylab="Nombre de séries",
tail(sort(table(autreSerie$rating),decreasing=FALSE),n=4),col=c("green","yellow","orange","red"))

# Voir les pays avec le plus de films.

x <- netflix_titles
for (i in 1:length(netflix_titles$country))
{
  y <- str_split(netflix_titles$country[i],",", simplify = TRUE)
  netflix_titles$country[i] <- y[1,1]
}
barplot(main="TOP 10 des pays sur Netflix",ylab="Nombre de pays",xlab="Pays de
production",head(sort(table(netflix_titles$country),decreasing=TRUE),n=10))

# Test de Shapiro
# Trier les Na pour les mettre en bas du tableau, et on les supprime

ShapiroTest <- netflix_movie[order(netflix_movie$duration),]
ShapiroTest <- head(ShapiroTest,-3)

shuffled_data  =  ShapiroTest[sample(1:nrow(ShapiroTest)),]
shuffled_data <- head(shuffled_data,-1200)
barplot(table(shuffled_data$duration))
# On s'attend à une loi distribuée normalement au vu du graphique
shapiro.test(shuffled_data$duration)
# ---
# W = 1, p-value <2e-16
# Non symétrique, distribution qui s'écarte trop de la moyenne à gauche --> Distribution non normale
#=================================================================================================================================================================
================================
# ANALYSE Amazon Prime

# Transformation des durées des films en nombre (en enlevant le " min")
amazon_movie <- subset(amazon_prime_titles,type =="Movie")
amazon_movie$duration <- gsub(" min","",as.character(amazon_movie$duration))
amazon_movie$duration
amazon_movie$duration <- as.double(amazon_movie$duration)
summary(amazon_movie$duration)
#  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#   0      75      91      91      106    601

#Audience visée des films
amazonPublicCible <- amazon_movie
amazonPublicCible$rating <- gsub("13+","Teens",as.character(amazonPublicCible$rating), fixed=TRUE)
amazonPublicCible$rating <- gsub("16+","Older Teens",as.character(amazonPublicCible$rating), fixed=TRUE)
amazonPublicCible$rating <- gsub("7+","Older kids",as.character(amazonPublicCible$rating),fixed=TRUE)
amazonPublicCible$rating <- gsub("18+","Adults",as.character(amazonPublicCible$rating), fixed=TRUE)
amazonPublicCible$rating <- gsub("TV-14|PG-13","Teens",as.character(amazonPublicCible$rating))
amazonPublicCible$rating <- gsub("AGES_16_|16","Older Teens",as.character(amazonPublicCible$rating))
amazonPublicCible$rating <- gsub("TV-PG|PG|TV-Y7-FV|TV-Y7","Older kids",as.character(amazonPublicCible$rating))
amazonPublicCible$rating <- gsub("TV-MA|NR|UR|NC-17|R|UNRATED|NOT_RATE|AGES_18_|18+","Adults",as.character(amazonPublicCible$rating))
amazonPublicCible$rating <- gsub("TV-Y|TV-G|ALL_AGES|ALL|G","Kids",as.character(amazonPublicCible$rating))

# Catégories uniques ->
unique(na.omit(amazonPublicCible$rating))



barplot(main="Audience visée des films selon la tranche d'âge, Amazon",xlab="Tranche d'âge",ylab="Nombre de films",
        sort(table(amazonPublicCible$rating),
             decreasing=FALSE))
```

```r
# Test de Shapiro Amazon
barplot(table(amazon_movie$duration))
shuffled_data2  =  amazon_movie[sample(1:nrow(amazon_movie)),]
shuffled_data2 <- head(shuffled_data2,-3000)
shapiro.test(shuffled_data2$duration)
# La durée des films sur amazon ne peut suivre une loi normale, rejet de l'hypothèse
summary(amazon_movie$duration)
summary(netflix_movie$duration)

# Test de Wilcoxon #Trouver les raisons
wilcox.test(amazon_movie$duration,netflix_movie$duration)
# W = 2e+07, p-value <2e-16
var.test(amazon_movie$duration,netflix_movie$duration,alternative = "greater")

tmdb_5000_movies <- read_csv("tmdb_5000_movies.csv",col_types = cols(budget = col_number(),popularity = col_number(), release_date = col_date(format
= "%Y-%m-%d"),revenue = col_number(), runtime = col_number(), vote_average = col_number(), vote_count = col_number()))
tmdb_5000_movies$release_date =as.Date(tmdb_5000_movies$release_date,"%Y")

usef_table = tmdb_5000_movies[,c(1,9,11,13,14,19,20)]
usef_table$budget = as.numeric(usef_table$budget)
usef_table$popularity = as.numeric(usef_table$popularity)
usef_table$revenue = as.numeric(usef_table$revenue)
usef_table$runtime = as.numeric(usef_table$runtime)
usef_table$vote_average = as.numeric(usef_table$vote_average)
usef_table$vote_count= as.numeric(usef_table$vote_count)

class(usef_table$vote_average)
typeof(usef_table$vote_average)
usef_table <- data.frame(usef_table)
typeof(usef_table)

for (i in 1:length(usef_table$production_countries)){
  y <- str_split(usef_table$production_countries[i],",", simplify = TRUE)
  usef_table$production_countries[i] <- y[1,1]
}
coloration = data.frame(head(sort(table(usef_table$production_countries),decreasing=TRUE),n=4))
for (i in 1:length(usef_table$production_countries)){
  if (usef_table$production_countries[i] %in% coloration$Var1){ }else{usef_table$production_countries[i]="autre"}}

ACP <- PCA(data.frame(tmdb_5000_movies[,c(1,9,13,14,19,20)]), graph=FALSE)
round(ACP$eig,digit=2)

fviz_pca_ind(ACP, col.ind=usef_table$production_countries, label="none", legend.title="Pays", addEllipses = TRUE)


barplot(ACP$eig[,1], main ="Eigen values", names.arg=1:nrow(ACP$eig), xlab="Numero de la composante", ylab="Valeur propre")
abline(h=1,col=2, lwd=2)

# Eigenvalue Proportion Cumulative

# dim 1       3.27       54.4         54
# dim 2       1.15       19.1         74
# dim 3       0.71       11.9         85
# dim 4       0.45        7.6         93
# dim 5       0.26        4.3         97
# dim 6       0.16        2.7        100
# D'après le critère de Kaiser, seuls les axes 1 et 2 sont intéressants à retenir.
# Le premier explique 54.4% de l'inertie, tandis que le deuxième explique seulement 19.1%
#Ainsi en considérant le plan 1,2; on récupère 73.5 % de l'information.
fviz_pca_var(ACP)

# COMMENTAIRES
table(usef_table$production_countries)
var <- get_pca_var(ACP)
corrplot(var$cos2, is.corr=TRUE, method="shade")
COL1(sequential = c("Oranges", "Purples", "Reds", "Blues", "Greens","Greys", "OrRd", "YlOrRd", "YlOrBr", "YlGn"), n = 200)
mcor <- cor(na.omit(data.frame(usef_table[,c(1,2,4,5,6,7)])))

corrplot(mcor, type="lower", tl.col="black", method = "shade",col=COL1("OrRd"), title="Heatmap des corrélations entre les données quantitatives des
films")
ACP$ind
barplot(table(round(usef_table$runtime)))

fviz_contrib(ACP,choice="var", axes=1)
fviz_contrib(ACP,choice="var", axes=2)

ACP$var$cos2
```