

2 - Clustering

Paolo Dalena

12/31/2020

1 - kmeans

```
library(tibble)
library(ggpubr)
library(plotly)

load("films_clus.RData")
# km1 <- kmeans(films_clus, centers = 10, nstart = 100)
```

This command returns an error, since euclidean distance can be computed only on numerical data.

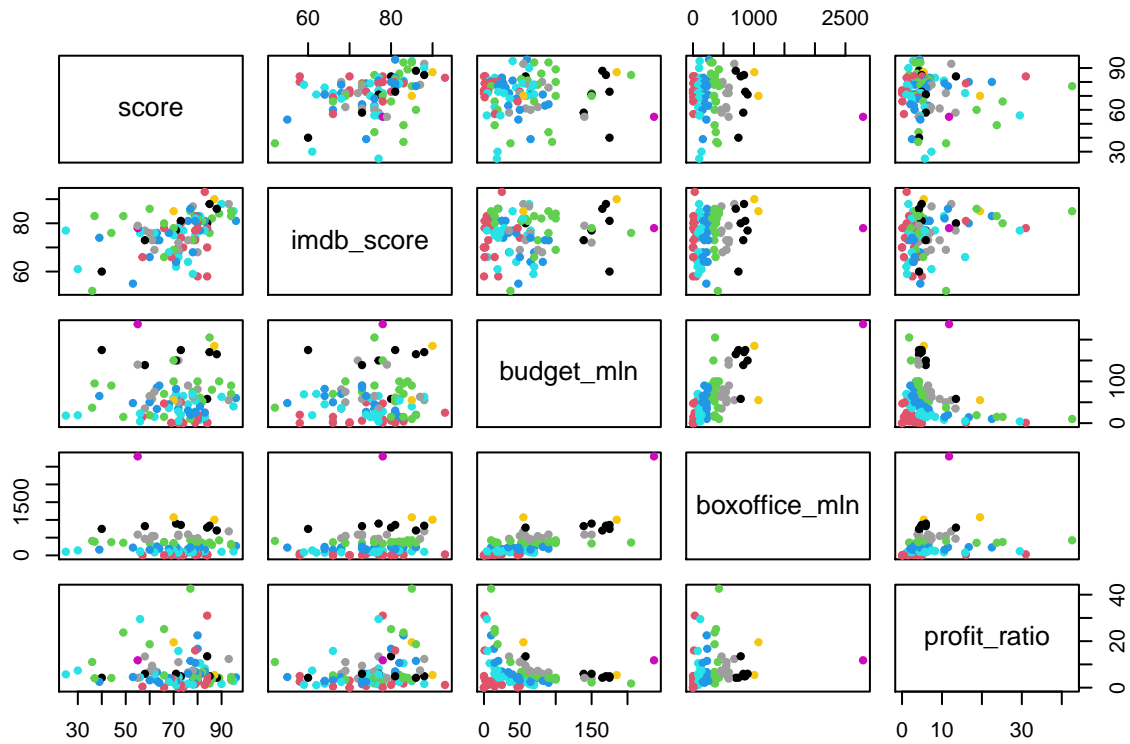
```
cbind(colnames(films_clus), 1:20)
```

```
##      [,1]      [,2]
## [1,] "score"    "1"
## [2,] "director" "2"
## [3,] "year"     "3"
## [4,] "country"  "4"
## [5,] "imdb_score" "5"
## [6,] "genre_1"   "6"
## [7,] "genre_2"   "7"
## [8,] "budget_mln" "8"
## [9,] "boxoffice_mln" "9"
## [10,] "profit_ratio" "10"
## [11,] "marcello_score" "11"
## [12,] "d_DiCaprio" "12"
## [13,] "d_Bale"    "13"
## [14,] "d_Pitt"    "14"
## [15,] "d_Damon"   "15"
## [16,] "cum_actors" "16"
## [17,] "d_frombook" "17"
## [18,] "d_truestory" "18"
## [19,] "d_rewatched" "19"
## [20,] "where"     "20"
```

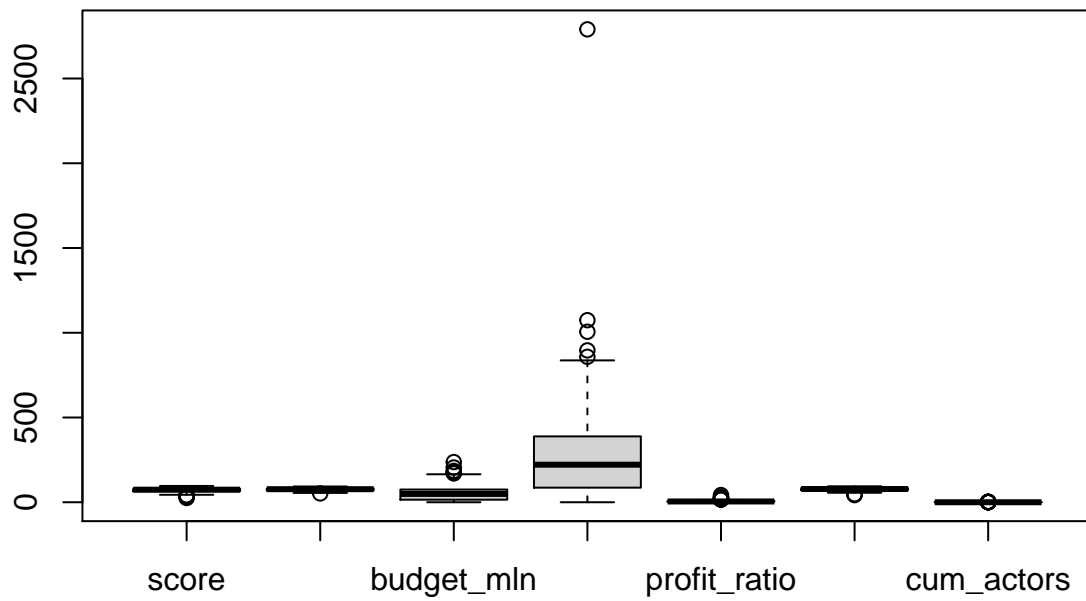
```
data_num <- films_clus[, c(1, 5, 8, 9, 10, 11, 16)]
```

Of course we're losing a lot of information.

```
km1 <- kmeans(data_num, centers = 8, nstart = 100)
pairs(data_num[1:5], col = km1$cluster, pch = 20)
```

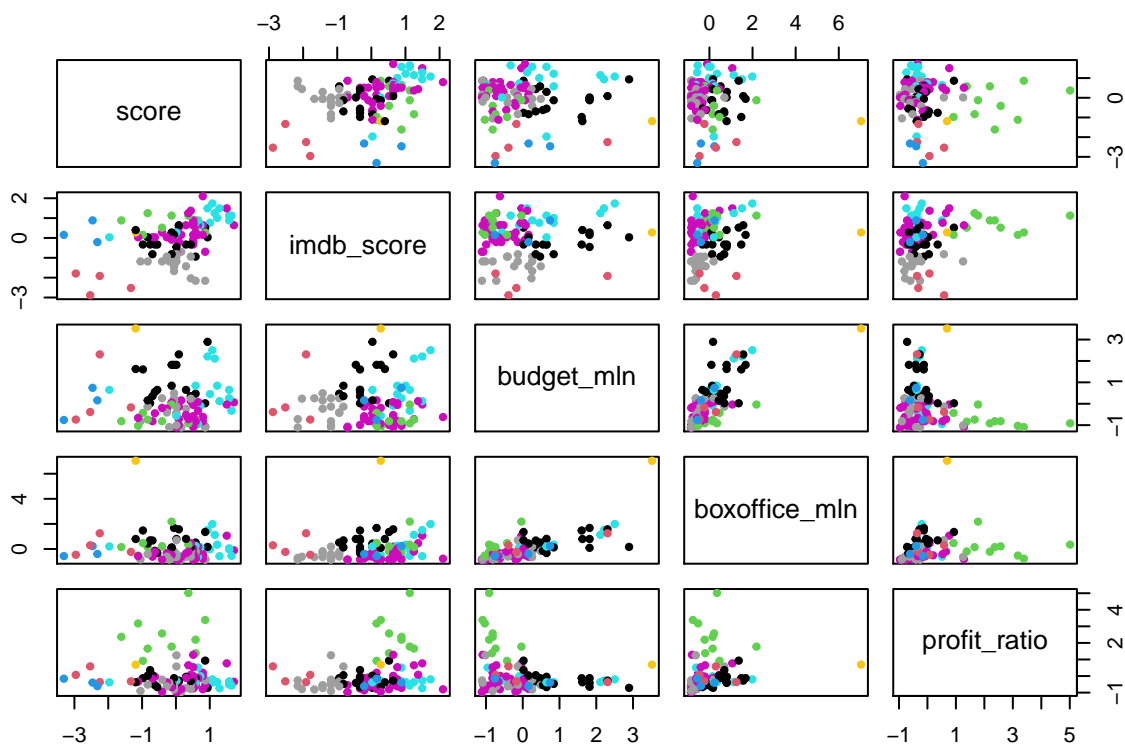


```
boxplot(data_num)
```



We have very different ranges \Rightarrow let's scale the data:

```
# standardized data
scdata <- scale(data_num)
km2 <- kmeans(scddata, centers = 8, nstart = 100)
pairs(scddata[,1:5], col = km2$cluster, pch = 20)
```



```
pc <- princomp(data_num)
summary(pc)
```

```
## Importance of components:
##
```

	Comp.1	Comp.2	Comp.3	Comp.4
## Standard deviation	353.7789497	35.392991766	15.512142888	9.4669399713
## Proportion of Variance	0.9871038	0.009879465	0.001897765	0.0007068351
## Cumulative Proportion	0.9871038	0.996983297	0.998881062	0.9995878973

```
##
```

	Comp.5	Comp.6	Comp.7
## Standard deviation	6.12824016	3.8113323302	4.133517e-01
## Proportion of Variance	0.00029619	0.0001145651	1.347530e-06
## Cumulative Proportion	0.99988409	0.9999986525	1.000000e+00

```
s_pc <- princomp(scddata)
summary(s_pc)
```

```
## Importance of components:
##
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
## Standard deviation	1.475324	1.331064	1.119814	0.8438584	0.80742929
## Proportion of Variance	0.314081	0.255661	0.180950	0.1027557	0.09407533
## Cumulative Proportion	0.314081	0.569742	0.750692	0.8534477	0.94752300

```
##
```

	Comp.6	Comp.7
## Standard deviation	0.46204299	0.38753306
## Proportion of Variance	0.03080573	0.02167127
## Cumulative Proportion	0.97832873	1.00000000

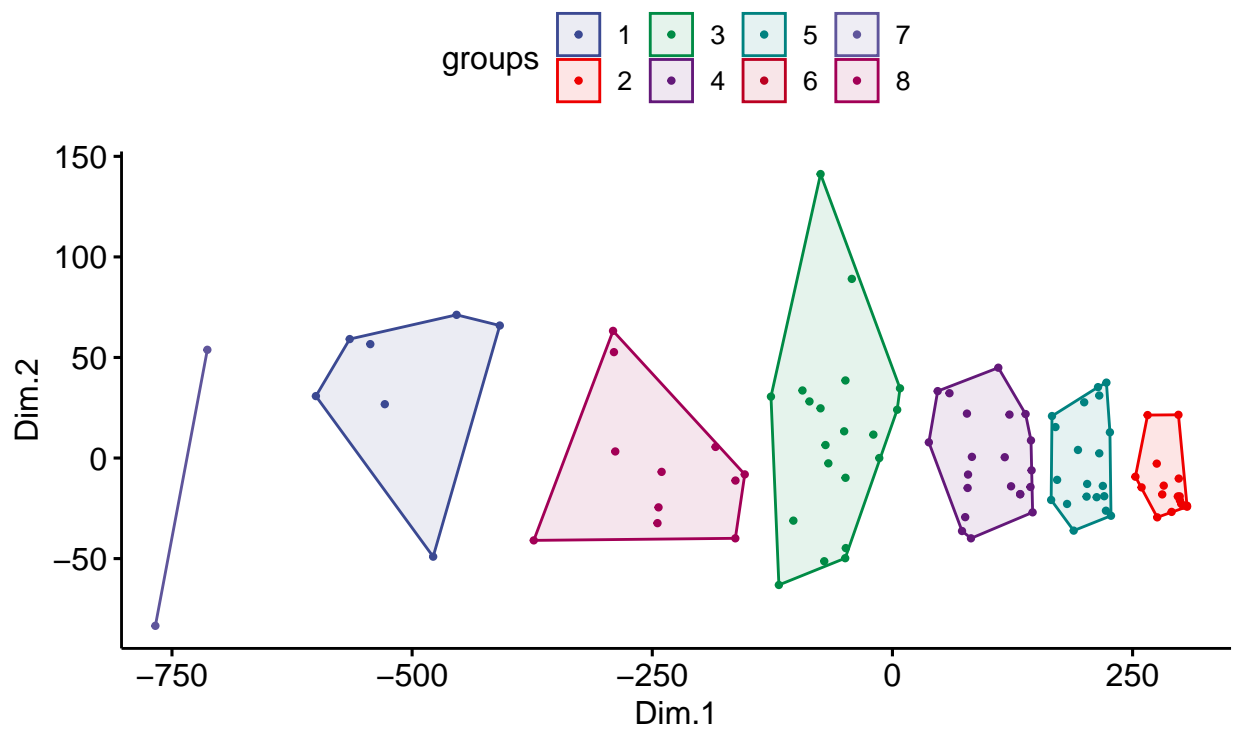
The first two principal components are not enough informative.
Let's evaluate the results on the first two anyways:

```
tbb <- tibble(
  "Dim.1" = pc$scores[, 1],
  "Dim.2" = pc$scores[, 2],
  "s_Dim.1" = s_pc$scores[, 1],
  "s_Dim.2" = s_pc$scores[, 2],
  "groups" = as.factor(km1$cluster),
  "s_groups" = as.factor(km2$cluster)
)

ggscatter(tbb,
  x = "Dim.1", y = "Dim.2",
  label = NULL,
  color = "groups",
  palette = "aaas",
  xlim= c(-750, 300),
  size = 0.8,
  ellipse = TRUE,
  ellipse.type = "convex",
  main = "Unscaled data",
  subtitle = "Problem: too much different ranges"
)
```

Unscaled data

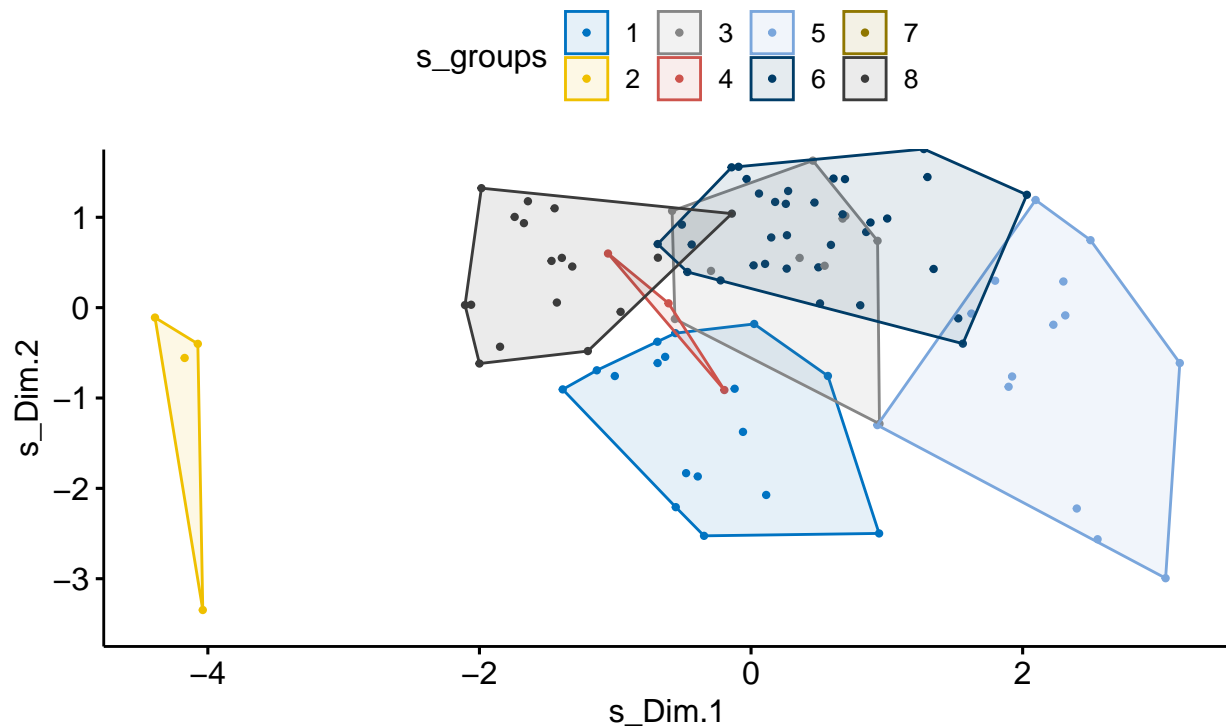
Problem: too much different ranges



```
ggscatter(tbb,
  x = "s_Dim.1", y = "s_Dim.2",
  label = NULL,
  color = "s_groups",
  palette = "jco",
  ylim = c(-3.5, 1.5),
  size = 0.8,
  ellipse = TRUE,
  ellipse.type = "convex",
  main = "Scaled data",
  subtitle = "Problem: first two princomp not enough informative"
)
```

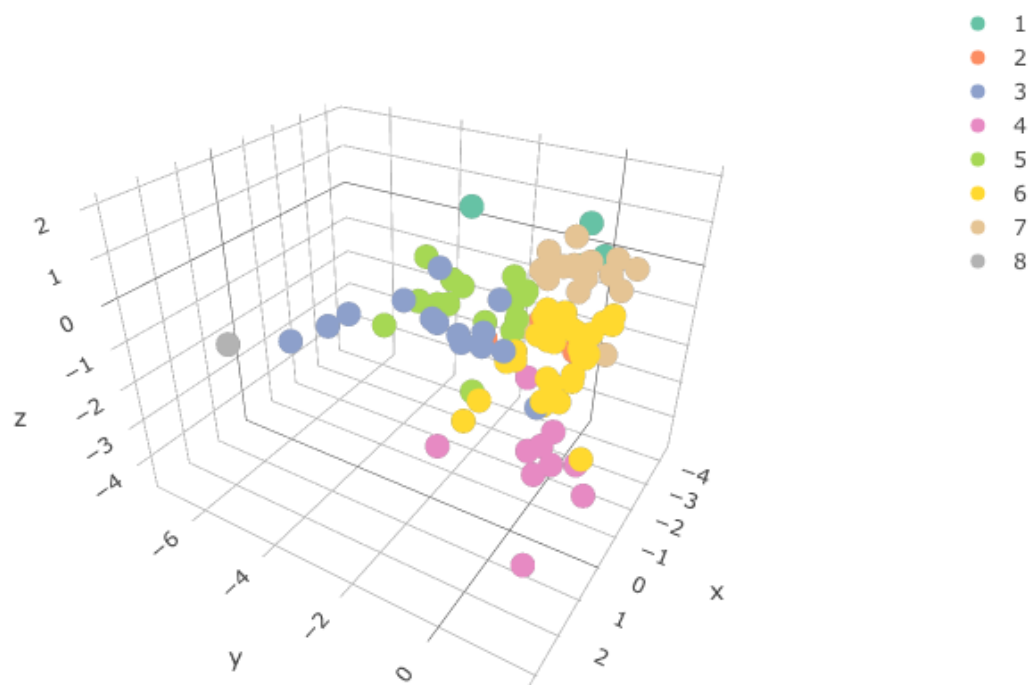
Scaled data

Problem: first two princomp not enough informative



And now let's plot the clustering on the first three principal components:

```
plot_ly(
  x = s_pc$scores[, 1],
  y = s_pc$scores[, 2],
  z = s_pc$scores[, 3],
  type = "scatter3d",
  mode = "markers",
  color = as.factor(km2$cluster)
)
```



Note that the previous plot is interactive in the html file, so you can move the axes as you want.