Preparação

Para preparar o ambiente para a execução do código, primeiro deve-se instalar alguns pacotes que permitem o acesso a métodos e execução de gráficos.

São eles:

```
if(!require(factoextra)) install.packages("factoextra")
if(!require(readxl)) install.packages("readxl")
if(!require(dplyr)) install.packages("dplyr")
if(!require(car)) install.packages("car")
if(!require(psych)) install.packages("psych")
if(!require(RVAideMemoire)) install.packages("RVAideMemoire")
if(!require(pacman)) install.packages("pacman")
if(!require(pacman)) install.packages("cluster")

library(factoextra)
library(cluster)
library(readxl)
library(dplyr)
library(car)
library(psych)
library(RVAideMemoire)
library(pacman)
pacman :: p_load(dplyr, ggplot2, car, rstatix, lmtest, ggpubr, ggpmisc, psych,
MASS, DescTools, QuantPsyc)
```

Base de dados

A base de dados utilizada foi do próprio R, importada da seguinte forma:

```
data('USArrests')
dados <- USArrests
```

Nessa base de dados, é apresentado o perfil de ocorrência de determinados tipos de crime por estados dos estados unidos. A base também trás a informação da população urbana em cada um dos estados.

O perfil dos dados é:

	Murder	**Assault**	**UrbanPop**	**Rape**
Min.	0.800	45.0	32.00	7.30
1st Qu.	4.075	109.0	54.50	15.07
Median	7.250	159.0	66.00	20.10

	Murder	Assault	UrbanPop	Rape
Mean	7.788	170.8	65.54	21.23
3rd Qu.	11.250	249.0	77.75	26.18
Max.	17.400	337.0	91.00	46.00

Como a clusterização é uma avaliação realizada por meio da comparação entre distâncias, a dimensão dos atributos importa para o correto resultado da análise. Dessa forma, foi necessário colocar os dados em uma mesma escala, de forma que uma variável com um valor médio maior não tenha mais influência na minha clusterização em detrimento de uma outra variável com um valor médio menor mas que explique melhor meu modelo. Em suma, colocando as variáveis em uma mesma escala, eu retiro a ponderação destas feita pela escala.

Para isso, o código realizado foi:

```
dados.p <- scale(dados)
```

Clusterização Hierárquica

A clusterização é uma análise não supervisionada, isto é, o algoritmo realiza o agrupamento por parâmetros matemáticos e não pelo nome da amostra.

A matemática por trás de uma clusterização avalia a distância entre cada um dos pontos, distância essa que pode ser calculada de várias formas diferentes. O R permite as seguintes para clusterização hierárquica:

Distâncias
euclidean
maximum
manhattan
canberra
binary
minkowski
pearson
spearman
kendall

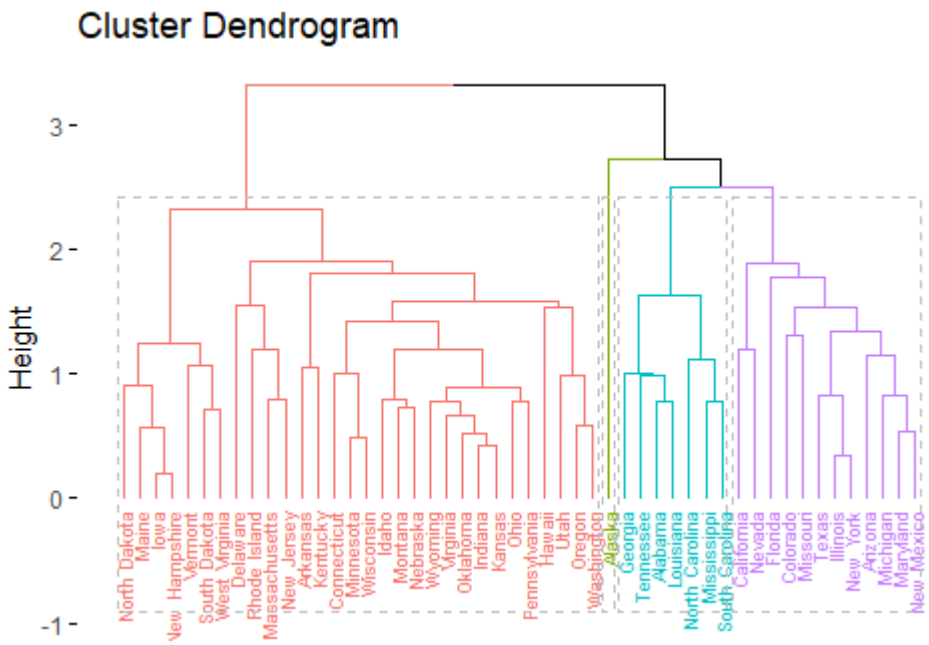
O código que permite a clusterização hierárquica é demonstrado a seguir:

```
d.eucl <- dist(dados.p, method = 'euclidean')
hc.m <- hclust(d.manh, method = 'average')
```

Por meio desse código, é utilizada a distância euclideana com o método da média para o cálculo dos clusteres hierárquicos.

Uma boa forma de visualizar uma clusterização hierárquica é por meio do dendograma. Um diagrama que demonstra como acontece a separação dos grupos e qual a hierarquia dessa separação.

O dendograma gerado é demonstrado a seguir:



Por meio do diagrama é possível observar quais grupos são formados com quais componentes cortanto na altura 2.5 formando 4 grupos de componentes e como se dá a relação entre cada um dos grupos. A altura de um dendograma representa a distância entre dois grupos.

Nesse dendograma, os grupos estão separados por cores e por caixas. O gráfico foi gerado pelo seguinte comando:

```
fviz_dend(hc.m, cex = 0.5, k = groups, color_labels_by_k = TRUE, rect = TRUE)
```

Clusterização Não Hierárquica

Para realizar a clusterização não hierárquica, foi realizado o método kmeans. Este algoritmo visa encontrar o máximo local em cada iteração onde k é a quantidade de grupos e means são as médias entre esses grupos.

A divisão dos grupos será realizada com base, nesse caso, entre as médias dos componentes desse grupo

O algoritmo é realizado em 6 passos.

1. Especificação do número de agrupamentos
2. Atribuição aleatória de dados a um cluster
3. Posicionamento de centróides
4. Reatribuição de dados a um cluster, de forma paramétrica
5. Re-posicionamento de centróides
6. Repetição dos passos 4 e 5 para encontrar máximos globais para o modelo.

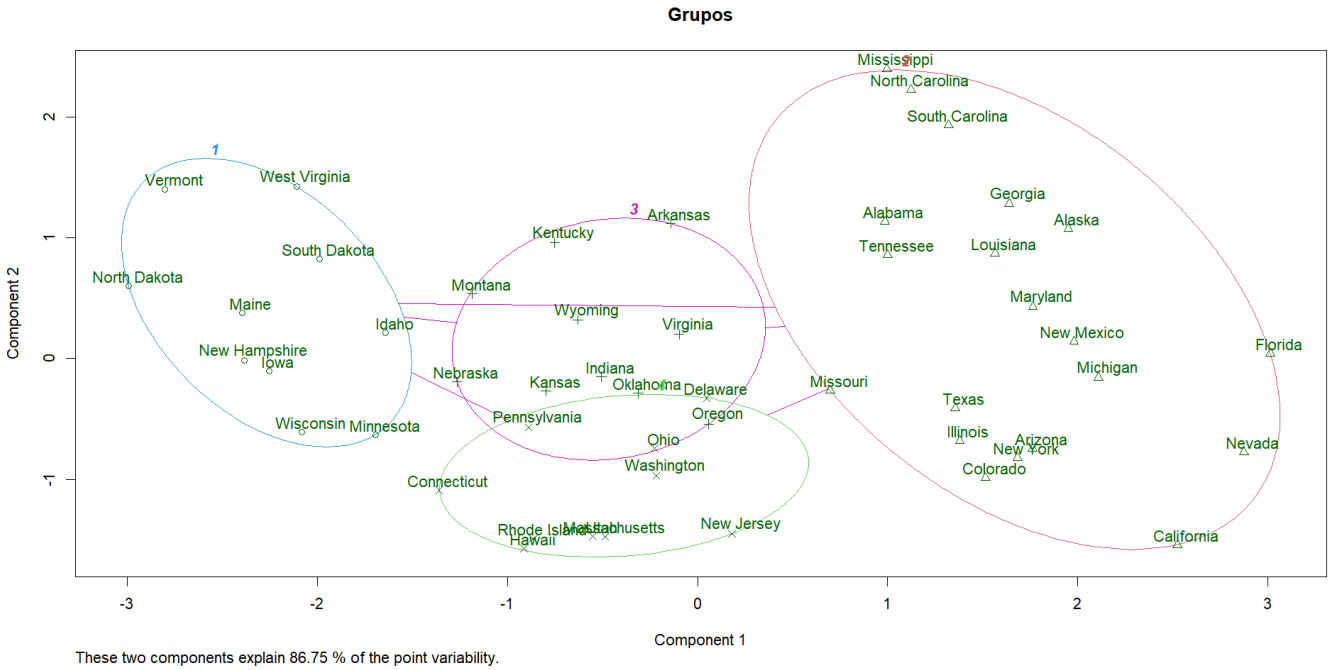
Para a clusterização não hierárquica, os clusteres são passados para o algoritmo.

A melhor quantidade de clusteres é aquela que aumenta a homogeneidade entre os pontos internos e a heterogeneidade entre os clusteres de forma parcimoniosa.

Para gerar os clusteres, bem como o diagrama, o seguinte código foi utilizado:

```
k_means<-kmeans(dados.p, centers = groups, nstart = 1)
clusplot(dados.p, k_means$cluster, color = T, labels = 2, main = 'Grupos')
```

O gráfico clusplot vai me mostrar como o modelo separou os grupos com base nos componentes que ele criou para a avaliação e quais os componentes de cada grupo.



Código:

```
if(!require(factoextra)) install.packages("factoextra")
if(!require(readxl)) install.packages("readxl")
if(!require(dplyr)) install.packages("dplyr")
if(!require(car)) install.packages("car")
if(!require(psych)) install.packages("psych")
if(!require(RVAideMemoire)) install.packages("RVAideMemoire")
if(!require(pacman)) install.packages("pacman")
if(!require(pacman)) install.packages("cluster")
```

```
library(factoextra)
library(cluster)
library(readxl)
library(dplyr)
library(car)
library(psych)
library(RVAideMemoire)
library(pacman)
pacman :: p_load(dplyr, ggplot2, car, rstatix, lmtest, ggpubr, ggpmisc, psych,
MASS, DescTools, QuantPsyc)

data('USArrests')
dados <- USArrests

summary(dados)

p.cov <- var(scale(dados))
p.mean <- apply(dados, 2, mean)
p.mah <- mahalanobis(dados, p.mean, p.cov)
View(p.mah)

dados.p <- scale(dados)

#Modelo Hierárquico

d.eucl <- dist(dados.p, method = 'euclidean')
round(as.matrix(d.eucl)[1:5,1:5],1)
hc.m <- hclust(d.eucl, method = 'average')

groups <- 4

fviz_dend(hc.m, cex = 0.5, k = groups, color_labels_by_k = TRUE, rect = TRUE)


#Modelo não hierárquico (kmeans)

k_means<-kmeans(dados.p, centers = groups, nstart = 1)

plot(x=dados.p[,4], y=dados.p[,3], col=k_means$cluster)
points(k_means$centers, pch=3, cex=1)

clusplot(dados.p, k_means$cluster, color = T, labels = 2, main = 'Grupos')
```