INF0613 – Aprendizado de Máquina Não Supervisionado

Trabalho 3 - Técnicas de Agrupamento

Nicole Nogueira Silva Rodolfo Dalla Costa

O objetivo deste trabalho é exercitar o uso de algoritmos de agrupamento. Neste trabalho, vamos analisar diferentes atributos de carros com o objetivo de verificar se seus atributos são suficientes para indicar um valor de risco de seguro. O conjunto de dados já apresenta o risco calculado no campo `symboling` indicado na Tabela 1. Quanto mais próximo de 3, maior o risco. O conjunto de dados que deve ser usado está disponível na página do Moodle com o nome `imports-85.data`.

Atividade 0 – Configurando o ambiente

Antes de começar a implementação do seu trabalho configure o *workspace* e importe todos os pacotes e execute o preprocessamento da base:

```
# Adicione os pacotes usados neste trabalho:
library(dplyr)
library(caret)
library(NbClust)
library(factoextra)
library(dbscan)
library(knitr)

# Configure ambiente de trabalho na mesma pasta
# onde colocou a base de dados:
setwd("C:/Users/nicol/Documents/Mineração de dados/inf-0611-0612/inf0613/t3")
```

Atividade 1 – Análise e Preparação dos Dados

O conjunto de dados é composto por 205 amostras com 26 atributos cada descritos na Tabela 1. Os atributos são dos tipos `factor`, `integer` ou `numeric`. O objetivo desta etapa é a análise e preparação desses dados de forma a ser possível agrupá-los nas próximas atividades.

Implementações: Nos itens a seguir você implementará a leitura da base e aplicará tratamentos básicos.

a) *Tratamento de dados Incompletos:* Amostras incompletas deverão ser tratadas, e você deve escolher a forma que achar mais adequada. Considere como uma amostra incompleta uma linha na qual faltam dados em alguma das colunas selecionadas anteriormente. Note que, dados faltantes nas amostras podem causar uma conversão do tipo do atributo de todas as amostras e isso pode impactar no item b).

```
### Leitura da base
imports_85 <- read.table("imports-85.data",sep =",")

### Tratamento de dados faltantes
#inspeção inicial
head(imports_85,2)
```

```
##   V1 V2          V3  V4  V5  V6          V7  V8    V9  V10 V11  V12  V13  V14
## 1  3  ? alfa-romero gas std two convertible rwd front 88.6 169 64.1 48.8 2548
## 2  3  ? alfa-romero gas std two convertible rwd front 88.6 169 64.1 48.8 2548
##    V15  V16 V17  V18  V19  V20 V21 V22  V23 V24 V25   V26
## 1 dohc four 130 mpfi 3.47 2.68   9 111 5000  21  27 13495
## 2 dohc four 130 mpfi 3.47 2.68   9 111 5000  21  27 16500
```

```r
#identificação de ? como dado faltante

### Removendo dados faltantes
imports_85[imports_85 == "?"] <- NA #substituindo por NA

#verificando se não está atrelado a uma categoria de seguro específica
faltantes <- imports_85[rowSums(is.na(imports_85)) != 0, ]

table(faltantes$V1) %>%
  kable(caption = "Risco do Seguro dos Dados faltantes", booktabs = T, linesep = "", digits = 2)
```

Table 1: Risco do Seguro dos Dados faltantes

Var1	Freq
-1	2
0	19
1	8
2	3
3	14

```r
#removendo os NAs
imports_85 <- na.omit(imports_85)

#transformando os atributos em suas devidas classes
imports_85 <- imports_85 %>%
  mutate(V2 = as.numeric(imports_85$V2),
         V19 = as.numeric(imports_85$V19),
         V20 = as.numeric(imports_85$V20),
         V22 = as.numeric(imports_85$V22),
         V23 = as.numeric(imports_85$V23),
         V26 = as.numeric(imports_85$V26))

#verificando a nova estrutura
#str(imports_85)
```

b) *Seleção de Atributos:* Atributos não-numéricos não podem ser usados com as técnicas agrupamento vistas em aula. Portanto, você deve selecionar um conjunto de atributos numéricos que serão usados para o agrupamento. Além disso você deve analisar se os atributos não-numéricos são descritivos para a realização dos agrupamentos. Caso um dos atributos não numéricos seja necessário, use a técnica do *one hot encoding* para transformá-lo em numérico. **Não** aplique essa técnica nos atributos `symboling` e `make` para os agrupamentos subsequentes, eles não devem fazer parte do agrupamento.

```r
# Seleção de atributos
# Transformando os dados categóricos em numéricos através de one hot encoding
#v4
imports_85$v4gas <- as.numeric(imports_85$V4 == "gas")
imports_85$v4diesel <- as.numeric(imports_85$V4 == "diesel")
```

```r
#v5
imports_85$v5std <- as.numeric(imports_85$V5 == "std")
imports_85$v5turbo <- as.numeric(imports_85$V5 == "turbo")
#v6
imports_85$v6four <- as.numeric(imports_85$V6 == "four")
imports_85$v6two <- as.numeric(imports_85$V6 == "two")
#v7
imports_85$v7convertible <- as.numeric(imports_85$V7 == "convertible")
imports_85$v7hardtop <- as.numeric(imports_85$V7 == "hardtop")
imports_85$v7hatchback <- as.numeric(imports_85$V7 == "hatchback")
imports_85$v7sedan <- as.numeric(imports_85$V7 == "sedan")
imports_85$v7wagon <- as.numeric(imports_85$V7 == "wagon")
#v8
imports_85$v84wd <- as.numeric(imports_85$V8 == "4wd")
imports_85$v8fwd <- as.numeric(imports_85$V8 == "fwd")
imports_85$v8rwd <- as.numeric(imports_85$V8 == "rwd")
#v9
imports_85$v9front <- as.numeric(imports_85$V9 == "front")
#v15
imports_85$v15dohc <- as.numeric(imports_85$V15 == "dohc")
imports_85$v15l <- as.numeric(imports_85$V15 == "l")
imports_85$v15ohc <- as.numeric(imports_85$V15 == "ohc")
imports_85$v15ohcf <- as.numeric(imports_85$V15 == "ohcf")
imports_85$v15ohcv <- as.numeric(imports_85$V15 == "ohcv")
#v16
imports_85$V16eight <- as.numeric(imports_85$V16 == "eight")
imports_85$V16five <- as.numeric(imports_85$V16 == "five")
imports_85$V16four <- as.numeric(imports_85$V16 == "four")
imports_85$V16six <- as.numeric(imports_85$V16 == "six")
imports_85$V16three <- as.numeric(imports_85$V16 == "three")
#V18
imports_85$V181bbl <- as.numeric(imports_85$V18 == "1bbl")
imports_85$V182bbl <- as.numeric(imports_85$V18 == "2bbl")
imports_85$V18idi <- as.numeric(imports_85$V18 == "idi")
imports_85$V18mfi <- as.numeric(imports_85$V18 == "mfi")
imports_85$V18mpfi <- as.numeric(imports_85$V18 == "mpfi")
imports_85$V18spdi <- as.numeric(imports_85$V18 == "spdi")

#Matriz de correlação das variáveis
numericos <- imports_85[,c(2,c(10:14),17,19:40,42:57)] #removendo categoricos e o V9 - Front
matriz_corr <- cor(numericos)
#print(matriz_corr)

#Encontrando atributos fortemente correlacionados
altamente_corr <- findCorrelation(matriz_corr, cutoff =0.5)
print(altamente_corr)
```

```
##  [1]  6 15  4  3  7  2 14 11 13 29 28 41 37 16 17 42 18 32 24 20 21 34  9 35
```

```r
#Atributos relevantes
atributos <- names(numericos[,-altamente_corr])
imports_85sel <- imports_85 %>% select(atributos)
```

Análises

Após as implementações escreva uma análise da base de dados. Em especial, descreva o conjunto de dados inicial, relate como foi realizado o tratamento, liste quais os atributos escolhidos para manter na base e descreva a base de dados após os tratamentos listados. Explique todos os passos executados, mas sem copiar códigos na análise. Além disso justifique suas escolhas de tratamento nos dados faltantes e seleção de atributos.

Resposta:

O banco de dados é formado por informações relacionadas a características de carros. Entre os 26 atributos disponibilizados, 25 correspondem a fatores descritores dos carros inclusive a coluna *symboling*, representam o risco atrelado ao seguro do carro descrito. Ao realizar uma análise prévia, nota-se a existência de dados faltantes em 46 observações.

Antes de remover as observações, é necessário entender se os dados faltantes estavam atrelados à uma categoria específica de risco de seguro. Como observamos que isto não ocorria, decidimos remover esses registros com dados faltantes, resultando em uma base com 159 registros.

Após remover os dados faltantes, realizamos um tratamento das variáveis dado a presença de atributos categóricos. Assim, aplicamos a técnica de One-Hot-Encoding para conversão dos dados em variáveis numéricas, de forma a deixar o conjunto de dados mais preparado para a seleção. Com os dados transformados, a correlação entre todas as variáveis foi mensurada e a partir dos resultados da matriz, foi possível concluir que as variáveis, *V2, V13, V19, V21, V23, v5turbo, v7convertible, v7hardtop, v7sedan, v7wagon, v84wd, v15dohc, v15l, v15ohcf, V16five, V16six, V16three, V181bbl, V18mfi, V18mpfi* eram suficientes para contribuir para a predição do risco do seguro.

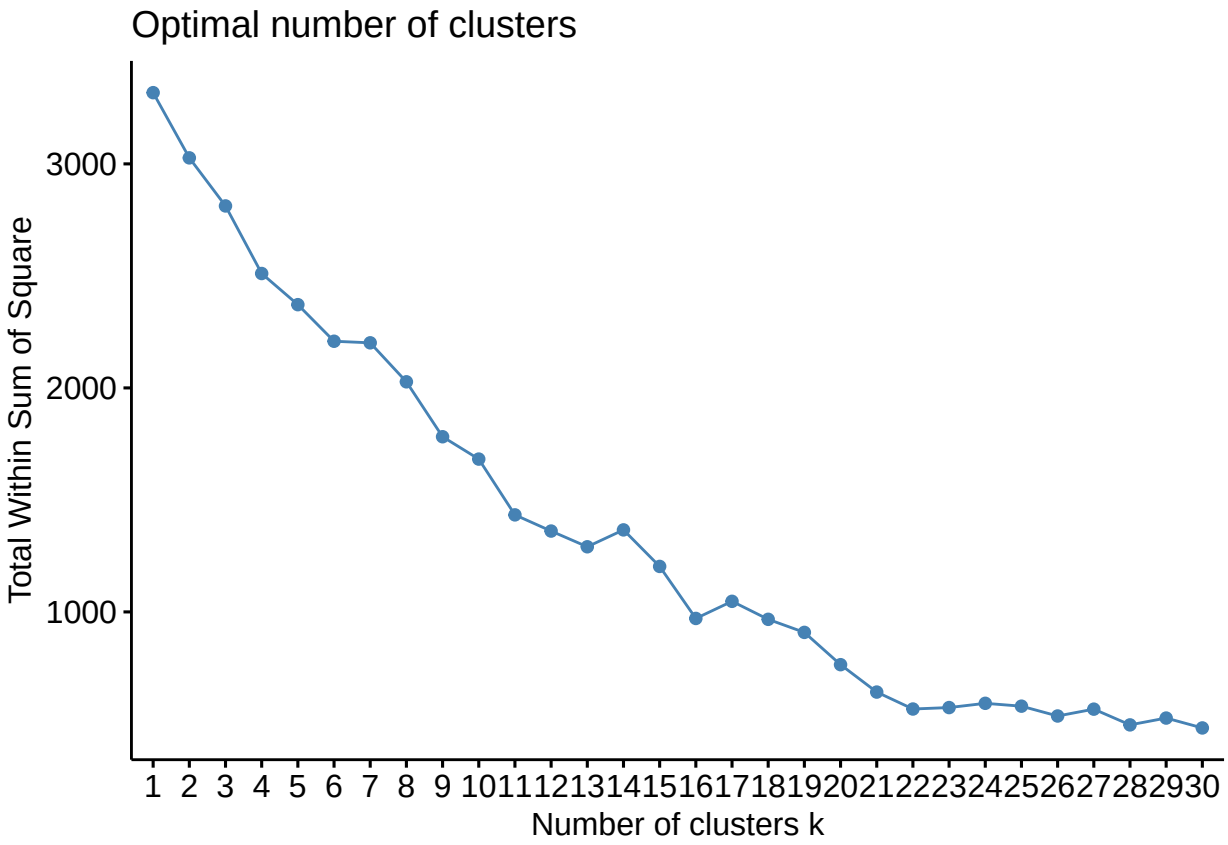
Atividade 2 – Agrupamento com o K-*means*

Nesta atividade, você deverá agrupar os dados com o algoritmo K-*means* e utilizará duas métricas básicas para a escolha do melhor K: a soma de distâncias intra-cluster e o coeficiente de silhueta.

Implementações: Nos itens a seguir você implementará a geração de gráficos para a análise das distâncias intra-cluster e do coeficiente de silhueta. Em seguida, você implementará o agrupamento dos dados processados na atividade anterior com o algoritmo K-*means* utilizando o valor de K escolhido.

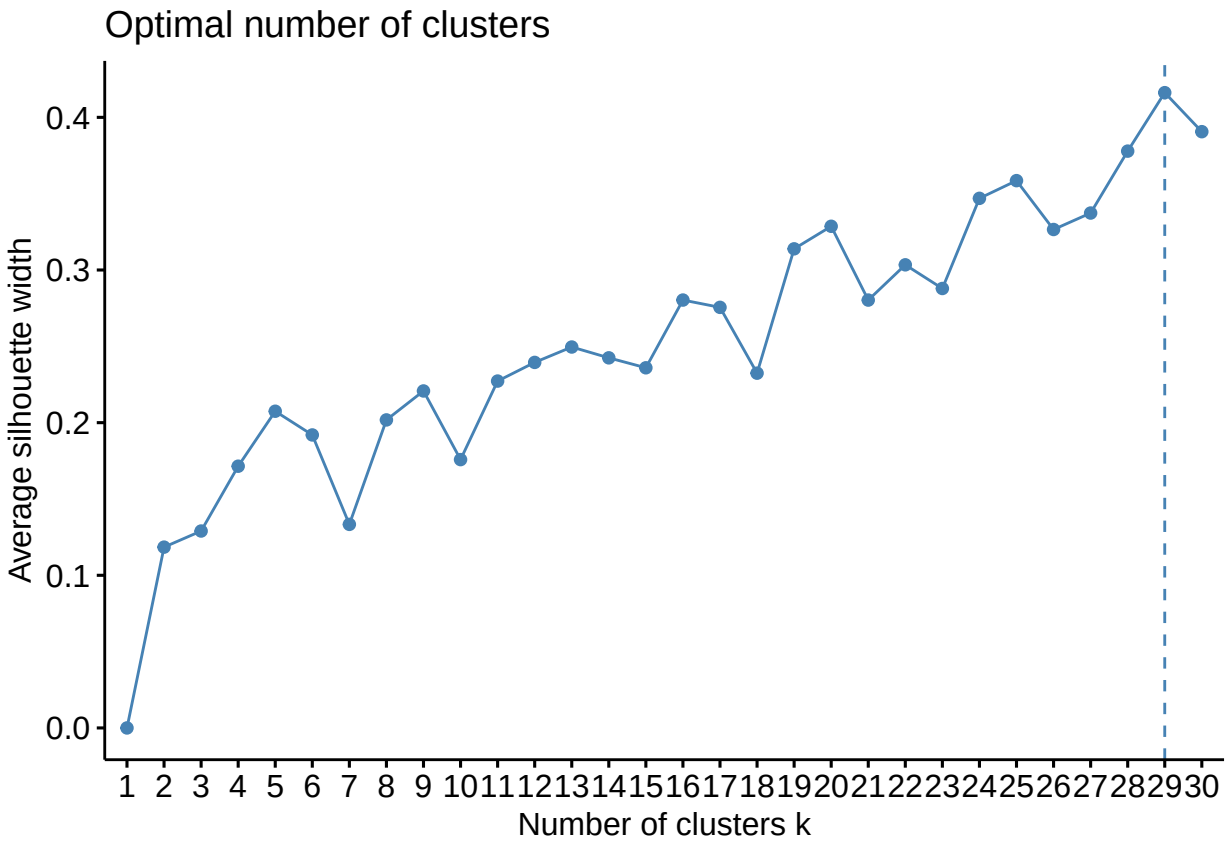
a) *Gráfico Elbow Curve:* Construa um gráfico com a soma das distâncias intra-cluster para K variando de 2 a 30.

```
# Construindo um gráfico com as distâncias intra-cluster
imports.scaled <- scale(imports_85sel)
set.seed(7)
fviz_nbclust(imports.scaled, kmeans,k.max = 30, method ="wss")
```



b) *Gráfico da Silhueta:* Construa um gráfico com o valor da silhueta para K variando de 2 a 30.

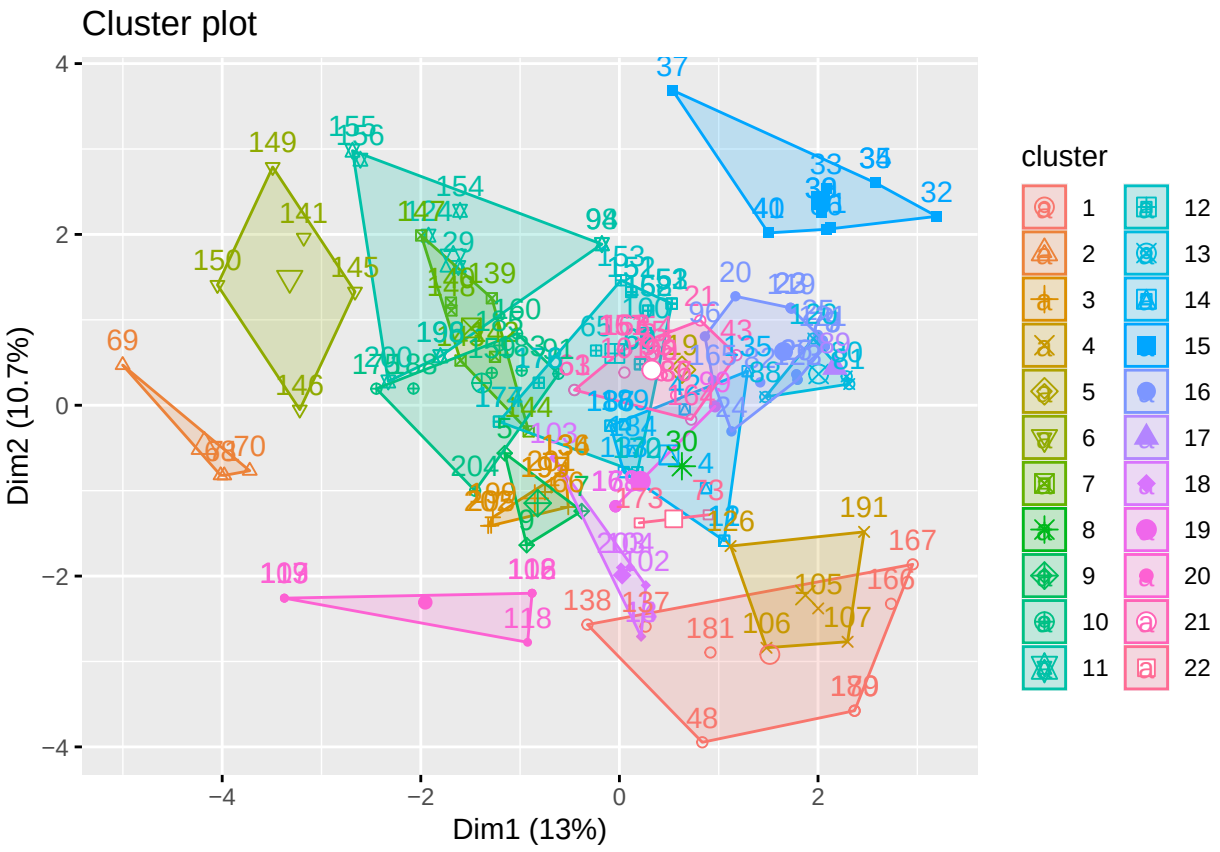
```
# Construindo um gráfico com os valores da silhueta
fviz_nbclust(imports.scaled, kmeans, method ="silhouette", k.max = 30)
```



c) *Escolha do K:* Avalie os gráficos gerados nos itens anteriores e escolha o melhor valor de K com base nas informações desses gráficos e na sua análise. Se desejar, use também a função `NbClust` para ajudar nas análises. Com o valor de K definido, utilize o rótulo obtido para cada amostra, indicando o grupo ao qual ela pertence, para gerar um gráfico de dispersão (atribuindo cores diferentes para cada grupo).

```
# Aplicando o k-means com o k escolhido
set.seed(7)
final <- kmeans(imports.scaled, centers =22)

# Construindo um gráfico de dispersão
fviz_cluster(final, data =imports.scaled)
```



Análises

Descreva cada um dos gráficos gerados nos itens acima e analise-os. Inclua na sua análise as informações mais importantes que podemos retirar desses gráficos. Discuta sobre a escolha do valor K e sobre a apresentação dos dados no gráfico de dispersão.

Resposta:

Atividade 3 – Agrupamento com o *DBscan*

Nesta atividade, você deverá agrupar os dados com o algoritmo *DBscan*. Para isso será necessário experimentar com diferentes valores de *eps* e *minPts*.

a) *Ajuste de Parâmetros:* Experimente com valores diferentes para os parâmetros *eps* e *minPts*. Verifique o impacto dos diferentes valores nos agrupamentos.

```
set.seed(7)

# Experimento com valores de eps e minPts
db <- dbscan(imports_85sel, eps =0.15 , minPts =5)
print(db)
```

```
## DBSCAN clustering for 159 objects.
## Parameters: eps = 0.15, minPts = 5
## The clustering contains 0 cluster(s) and 159 noise points.
##
##   0
```

```
## 159
##
## Available fields: cluster, eps, minPts
```

```
# Experimento com valores de eps e minPts
db <- dbscan(imports_85sel, eps =0.2 , minPts =10)
print(db)
```

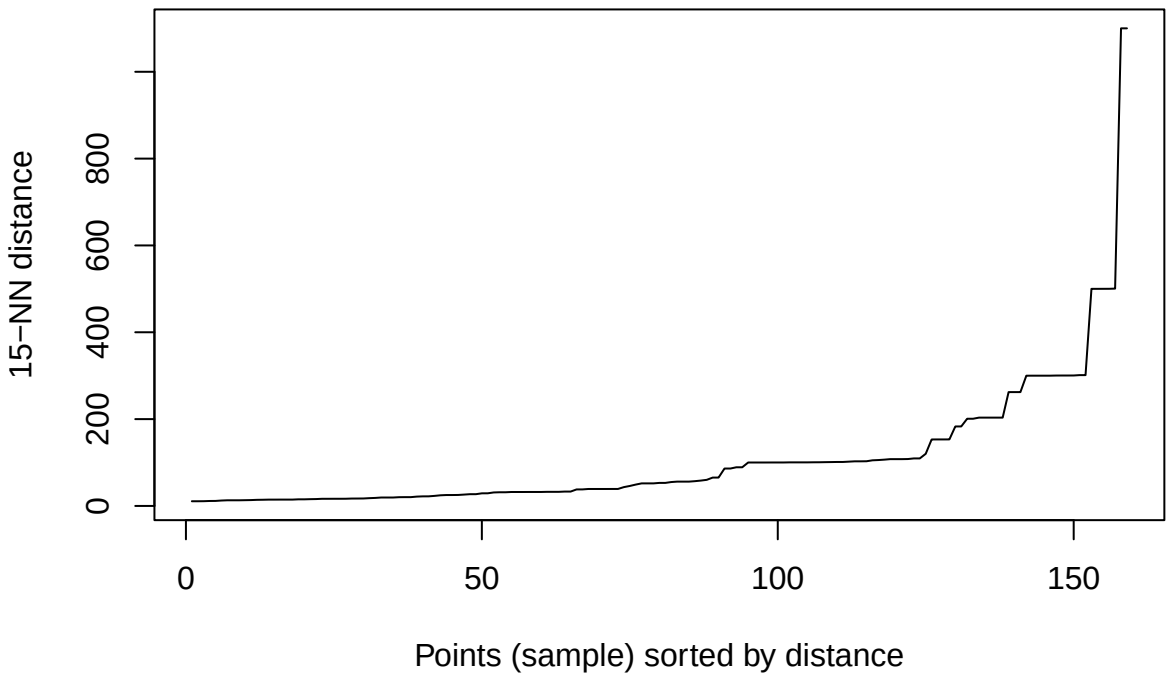
```
## DBSCAN clustering for 159 objects.
## Parameters: eps = 0.2, minPts = 10
## The clustering contains 0 cluster(s) and 159 noise points.
##
##   0
## 159
##
## Available fields: cluster, eps, minPts
```

```
# Experimento com valores de eps e minPts
db <- dbscan(imports_85sel, eps =0.3 , minPts =15)
print(db)
```

```
## DBSCAN clustering for 159 objects.
## Parameters: eps = 0.3, minPts = 15
## The clustering contains 0 cluster(s) and 159 noise points.
##
##   0
## 159
##
## Available fields: cluster, eps, minPts
```

b) *Determinando Ruídos:* Escolha o valor de *minPts* que obteve o melhor resultado no item anterior e use a função `kNNdistplot` do pacote `dbscan` para determinar o melhor valor de *eps* para esse valor de *minPts*. Lembre-se que o objetivo não é remover todos os ruídos.

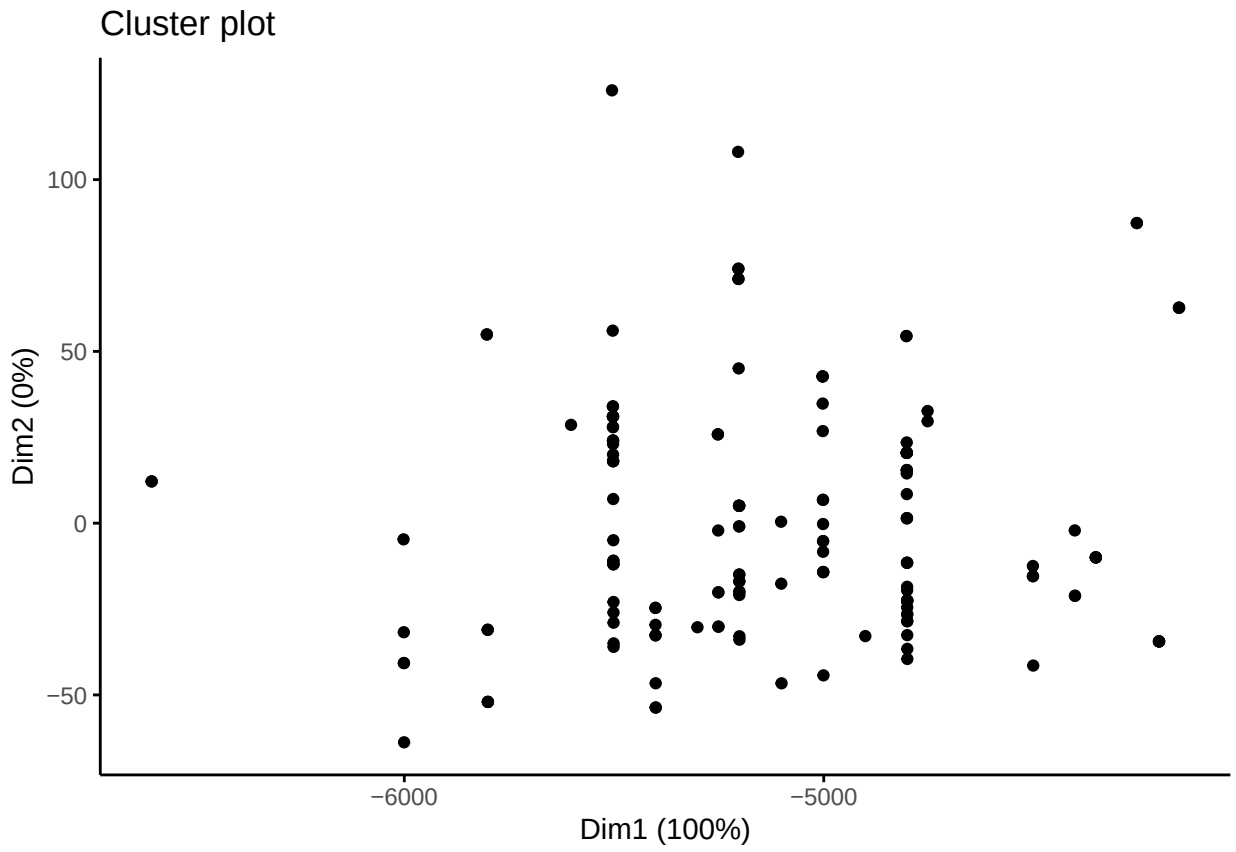
```
# Encontrando o melhor eps com o kNNdistplot
kNNdistplot(imports_85sel, k =15)
```



c) *Visualizando os Grupos:* Após a escolha dos parâmetros *eps* e *minPts*, utilize o rótulo obtido para cada amostra, indicando o grupo ao qual ela pertence, para gerar um gráfico de dispersão (atribuindo cores diferentes para cada grupo).

```
# Aplicando o DBscan com os parâmetros escolhidos
db <-dbscan(imports_85sel, eps =0.2 , minPts =15)

# Construindo um gráfico de dispersão
fviz_cluster(db, data = imports_85sel , stand = FALSE, ellipse = FALSE , show.clust.cent = FALSE,
geom ="point", palette ="jco", ggtheme = theme_classic())
```



Análises

Descreva os experimentos feitos para a escolha dos parâmetros *eps* e *minPts*. Inclua na sua análise as informações mais importantes que podemos retirar dos gráficos gerados. Justifique a escolha dos valores dos parâmetros e analise a apresentação dos dados no gráfico de dispersão.

Resposta:

Atividade 4 – Comparando os Algoritmos

Com base nas atividades anteriores, faça uma conclusão dos seus experimentos respondendo às seguintes perguntas:

a) Qual dos métodos apresentou melhores resultados? Justifique.

b) Quantos agrupamentos foram obtidos?

c) Analisando o campo `symboling` e o grupo designado para cada amostra, os agrupamentos conseguiram separar os níveis de risco?

d) Analisando o campo `make` que contém as marcas dos carros, os agrupamentos conseguiram separar as marcas?

Respostas: