INF0613 – Aprendizado de Máquina Não Supervisionado

Trabalho 3 - Técnicas de Agrupamento

Leonardo Cesr Silva dos Santos Fernando Augusto Cardoso Candalaft

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(factoextra)
library(dbscan)
library(ggrepel)
library(ggplot2)

# Configure ambiente de trabalho na mesma pasta
# onde colocou a base de dados:
setwd("~/workspace/mdc/04_aprendizado_nao_supervisionado_I/test03")
```

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
cars_df <- read.csv("./imports-85.data", header=F)
head(cars_df)
```

```
##   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
## 3  1   ? alfa-romero gas std  two   hatchback rwd front 94.5 171 65.5 52.4 2823
## 4  2 164        audi gas std four       sedan fwd front 99.8 177 66.2 54.3 2337
## 5  2 164        audi gas std four       sedan 4wd front 99.4 177 66.4 54.3 2824
## 6  2   ?        audi gas std  two       sedan fwd front 99.8 177 66.3 53.1 2507
##    V15  V16 V17  V18  V19  V20  V21 V22  V23 V24 V25   V26
## 1 dohc four 130 mpfi 3.47 2.68  9.0 111 5000  21  27 13495
## 2 dohc four 130 mpfi 3.47 2.68  9.0 111 5000  21  27 16500
## 3 ohcv  six 152 mpfi 2.68 3.47  9.0 154 5000  19  26 16500
## 4  ohc four 109 mpfi 3.19 3.40 10.0 102 5500  24  30 13950
## 5  ohc five 136 mpfi 3.19 3.40  8.0 115 5500  18  22 17450
## 6  ohc five 136 mpfi 3.19 3.40  8.5 110 5500  19  25 15250
```

```
# Tratamento de dados faltantes
summary(cars_df)
```

```
##        V1              V2                V3                V4
##  Min.   :-2.000   Length:205         Length:205         Length:205
##  1st Qu.: 0.000   Class :character   Class :character   Class :character
##  Median : 1.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 0.834
##  3rd Qu.: 2.000
##  Max.   : 3.000
##       V5                 V6                V7                V8
##  Length:205         Length:205         Length:205         Length:205
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       V9                 V10              V11           V12            V13
##  Length:205         Min.   : 86.6   Min.   :141   Min.   :60.3   Min.   :47.8
##  Class :character   1st Qu.: 94.5   1st Qu.:166   1st Qu.:64.1   1st Qu.:52.0
##  Mode  :character   Median : 97.0   Median :173   Median :65.5   Median :54.1
##                     Mean   : 98.8   Mean   :174   Mean   :65.9   Mean   :53.7
##                     3rd Qu.:102.4   3rd Qu.:183   3rd Qu.:66.9   3rd Qu.:55.5
##                     Max.   :120.9   Max.   :208   Max.   :72.3   Max.   :59.8
##       V14           V15                V16                V17
##  Min.   :1488   Length:205         Length:205         Min.   : 61
##  1st Qu.:2145   Class :character   Class :character   1st Qu.: 97
##  Median :2414   Mode  :character   Mode  :character   Median :120
##  Mean   :2556                                         Mean   :127
##  3rd Qu.:2935                                         3rd Qu.:141
##  Max.   :4066                                         Max.   :326
##      V18                V19                V20                V21
##  Length:205         Length:205         Length:205         Min.   : 7.0
##  Class :character   Class :character   Class :character   1st Qu.: 8.6
##  Mode  :character   Mode  :character   Mode  :character   Median : 9.0
##                                                           Mean   :10.1
##                                                           3rd Qu.: 9.4
##                                                           Max.   :23.0
##      V22                V23                V24            V25
```

```
##  Length:205         Length:205         Min.   :13.0   Min.   :16.0
##  Class :character   Class :character   1st Qu.:19.0   1st Qu.:25.0
##  Mode  :character   Mode  :character   Median :24.0   Median :30.0
##                                        Mean   :25.2   Mean   :30.8
##                                        3rd Qu.:30.0   3rd Qu.:34.0
##                                        Max.   :49.0   Max.   :54.0
##      V26
##  Length:205
##  Class :character
##  Mode  :character
##
##
##
```

```
for(col in names(cars_df)) {
  print(paste("Column name:", col))
  print(paste("NA values?: ", any(is.na(cars_df[, col]) ) ) )
  print("Unique values")
  print(unique(cars_df[, col]) )
  print("----------------------------")
}
```

```
## [1] "Column name: V1"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1]  3  1  2  0 -1 -2
## [1] "----------------------------"
## [1] "Column name: V2"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] "?"   "164" "158" "192" "188" "121" "98"  "81"  "118" "148" "110" "145"
## [13] "137" "101" "78"  "106" "85"  "107" "104" "113" "150" "129" "115" "93"
## [25] "142" "161" "153" "125" "128" "122" "103" "168" "108" "194" "231" "119"
## [37] "154" "74"  "186" "83"  "102" "89"  "87"  "77"  "91"  "134" "65"  "197"
## [49] "90"  "94"  "256" "95"
## [1] "----------------------------"
## [1] "Column name: V3"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] "alfa-romero"   "audi"          "bmw"           "chevrolet"
##  [5] "dodge"         "honda"         "isuzu"         "jaguar"
##  [9] "mazda"         "mercedes-benz" "mercury"       "mitsubishi"
## [13] "nissan"        "peugot"        "plymouth"      "porsche"
## [17] "renault"       "saab"          "subaru"        "toyota"
## [21] "volkswagen"    "volvo"
## [1] "----------------------------"
## [1] "Column name: V4"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1] "gas"    "diesel"
## [1] "----------------------------"
## [1] "Column name: V5"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
```

```
## [1] "std"   "turbo"
## [1] "----------------------------"
## [1] "Column name: V6"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1] "two"  "four" "?"
## [1] "----------------------------"
## [1] "Column name: V7"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1] "convertible" "hatchback"   "sedan"       "wagon"       "hardtop"
## [1] "----------------------------"
## [1] "Column name: V8"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1] "rwd" "fwd" "4wd"
## [1] "----------------------------"
## [1] "Column name: V9"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1] "front" "rear"
## [1] "----------------------------"
## [1] "Column name: V10"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1]  88.6  94.5  99.8  99.4 105.8  99.5 101.2 103.5 110.0  88.4  93.7 103.3
## [13]  95.9  86.6  96.5  94.3  96.0 113.0 102.0  93.1  95.3  98.8 104.9 106.7
## [25] 115.6  96.6 120.9 112.0 102.7  93.0  96.3  95.1  97.2 100.4  91.3  99.2
## [37] 107.9 114.2 108.0  89.5  98.4  96.1  99.1  93.3  97.0  96.9  95.7 102.4
## [49] 102.9 104.5  97.3 104.3 109.1
## [1] "----------------------------"
## [1] "Column name: V11"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] 169 171 177 177 193 178 177 189 194 197 141 156 159 157 175 173 145 150 163
## [20] 157 168 175 169 171 173 200 192 159 167 169 178 175 191 188 203 180 208 199
## [39] 178 173 172 165 170 166 162 173 182 185 178 187 199 167 169 176 182 187 157
## [58] 158 172 174 174 159 170 166 169 176 176 184 188 172 159 166 180 183 189
## [1] "----------------------------"
## [1] "Column name: V12"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] 64.1 65.5 66.2 66.4 66.3 71.4 67.9 64.8 66.9 70.9 60.3 63.6 63.8 64.6 63.9
## [16] 64.0 65.2 62.5 66.0 61.8 69.6 70.6 64.2 65.7 66.5 66.1 70.3 71.7 70.5 72.0
## [31] 68.0 64.4 65.4 68.4 68.3 65.0 72.3 66.6 63.4 65.6 67.7 67.2 68.9 68.8
## [1] "----------------------------"
## [1] "Column name: V13"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] 48.8 52.4 54.3 53.1 55.7 55.9 52.0 53.7 56.3 53.2 50.8 50.6 59.8 50.2 52.6
## [16] 54.5 58.3 53.3 54.1 51.0 53.5 51.4 52.8 47.8 49.6 55.5 54.4 56.5 58.7 54.9
## [31] 56.7 55.4 54.8 49.4 51.6 54.7 55.1 56.1 49.7 56.0 50.5 55.2 52.5 53.0 59.1
## [46] 53.9 55.6 56.2 57.5
## [1] "----------------------------"
```

```
## [1] "Column name: V14"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##   [1] 2548 2823 2337 2824 2507 2844 2954 3086 3053 2395 2710 2765 3055 3230 3380
##  [16] 3505 1488 1874 1909 1876 2128 1967 1989 2191 2535 2811 1713 1819 1837 1940
##  [31] 1956 2010 2024 2236 2289 2304 2372 2465 2293 2734 4066 3950 1890 1900 1905
##  [46] 1945 1950 2380 2385 2500 2410 2443 2425 2670 2700 3515 3750 3495 3770 3740
##  [61] 3685 3900 3715 2910 1918 1944 2004 2145 2370 2328 2833 2921 2926 2365 2405
##  [76] 2403 1889 2017 1938 1951 2028 1971 2037 2008 2324 2302 3095 3296 3060 3071
##  [91] 3139 3020 3197 3430 3075 3252 3285 3485 3130 2818 2778 2756 2800 3366 2579
## [106] 2460 2658 2695 2707 2758 2808 2847 2050 2120 2240 2190 2340 2510 2290 2455
## [121] 2420 2650 1985 2040 2015 2280 3110 2081 2109 2275 2094 2122 2140 2169 2204
## [136] 2265 2300 2540 2536 2551 2679 2714 2975 2326 2480 2414 2458 2976 3016 3131
## [151] 3151 2261 2209 2264 2212 2319 2254 2221 2661 2563 2912 3034 2935 3042 3045
## [166] 3157 2952 3049 3012 3217 3062
## [1] "----------------------------"
## [1] "Column name: V15"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1] "dohc"  "ohcv"  "ohc"   "l"     "rotor" "ohcf"  "dohcv"
## [1] "----------------------------"
## [1] "Column name: V16"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1] "four"   "six"    "five"   "three"  "twelve" "two"    "eight"
## [1] "----------------------------"
## [1] "Column name: V17"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] 130 152 109 136 131 108 164 209  61  90  98 122 156  92  79 110 111 119 258
## [20] 326  91  70  80 140 134 183 234 308 304  97 103 120 181 151 194 203 132 121
## [39] 146 171 161 141 173 145
## [1] "----------------------------"
## [1] "Column name: V18"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
## [1] "mpfi" "2bbl" "mfi"  "1bbl" "spfi" "4bbl" "idi"  "spdi"
## [1] "----------------------------"
## [1] "Column name: V19"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] "3.47" "2.68" "3.19" "3.13" "3.50" "3.31" "3.62" "2.91" "3.03" "2.97"
## [11] "3.34" "3.60" "2.92" "3.15" "3.43" "3.63" "3.54" "3.08" "?"    "3.39"
## [21] "3.76" "3.58" "3.46" "3.80" "3.78" "3.17" "3.35" "3.59" "2.99" "3.33"
## [31] "3.70" "3.61" "3.94" "3.74" "2.54" "3.05" "3.27" "3.24" "3.01"
## [1] "----------------------------"
## [1] "Column name: V20"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] "2.68" "3.47" "3.40" "2.80" "3.19" "3.39" "3.03" "3.11" "3.23" "3.46"
## [11] "3.90" "3.41" "3.07" "3.58" "4.17" "2.76" "3.15" "?"    "3.16" "3.64"
## [21] "3.10" "3.35" "3.12" "3.86" "3.29" "3.27" "3.52" "2.19" "3.21" "2.90"
## [31] "2.07" "2.36" "2.64" "3.08" "3.50" "3.54" "2.87"
## [1] "----------------------------"
```

```
## [1] "Column name: V21"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1]  9.00 10.00  8.00  8.50  8.30  7.00  8.80  9.50  9.60  9.41  9.40  7.60
## [13]  9.20 10.10  9.10  8.10 11.50  8.60 22.70 22.00 21.50  7.50 21.90  7.80
## [25]  8.40 21.00  8.70  9.31  9.30  7.70 22.50 23.00
## [1] "----------------------------"
## [1] "Column name: V22"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] "111" "154" "102" "115" "110" "140" "160" "101" "121" "182" "48"  "70"
## [13] "68"  "88"  "145" "58"  "76"  "60"  "86"  "100" "78"  "90"  "176" "262"
## [25] "135" "84"  "64"  "120" "72"  "123" "155" "184" "175" "116" "69"  "55"
## [37] "97"  "152" "200" "95"  "142" "143" "207" "288" "?"   "73"  "82"  "94"
## [49] "62"  "56"  "112" "92"  "161" "156" "52"  "85"  "114" "162" "134" "106"
## [1] "----------------------------"
## [1] "Column name: V23"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] "5000" "5500" "5800" "4250" "5400" "5100" "4800" "6000" "4750" "4650"
## [11] "4200" "4350" "4500" "5200" "4150" "5600" "5900" "5750" "?"    "5250"
## [21] "4900" "4400" "6600" "5300"
## [1] "----------------------------"
## [1] "Column name: V24"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] 21 19 24 18 17 16 23 20 15 47 38 37 31 49 30 27 25 13 26 36 22 14 45 28 32
## [26] 35 34 29 33
## [1] "----------------------------"
## [1] "Column name: V25"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##  [1] 27 26 30 22 25 20 29 28 53 43 41 38 24 54 42 34 33 31 19 17 23 32 39 18 16
## [26] 37 50 36 47 46
## [1] "----------------------------"
## [1] "Column name: V26"
## [1] "NA values?:  FALSE"
## [1] "Unique values"
##   [1] "13495" "16500" "13950" "17450" "15250" "17710" "18920" "23875" "?"
##  [10] "16430" "16925" "20970" "21105" "24565" "30760" "41315" "36880" "5151"
##  [19] "6295"  "6575"  "5572"  "6377"  "7957"  "6229"  "6692"  "7609"  "8558"
##  [28] "8921"  "12964" "6479"  "6855"  "5399"  "6529"  "7129"  "7295"  "7895"
##  [37] "9095"  "8845"  "10295" "12945" "10345" "6785"  "11048" "32250" "35550"
##  [46] "36000" "5195"  "6095"  "6795"  "6695"  "7395"  "10945" "11845" "13645"
##  [55] "15645" "8495"  "10595" "10245" "10795" "11245" "18280" "18344" "25552"
##  [64] "28248" "28176" "31600" "34184" "35056" "40960" "45400" "16503" "5389"
##  [73] "6189"  "6669"  "7689"  "9959"  "8499"  "12629" "14869" "14489" "6989"
##  [82] "8189"  "9279"  "5499"  "7099"  "6649"  "6849"  "7349"  "7299"  "7799"
##  [91] "7499"  "7999"  "8249"  "8949"  "9549"  "13499" "14399" "17199" "19699"
## [100] "18399" "11900" "13200" "12440" "13860" "15580" "16900" "16695" "17075"
## [109] "16630" "17950" "18150" "12764" "22018" "32528" "34028" "37028" "9295"
## [118] "9895"  "11850" "12170" "15040" "15510" "18620" "5118"  "7053"  "7603"
## [127] "7126"  "7775"  "9960"  "9233"  "11259" "7463"  "10198" "8013"  "11694"
## [136] "5348"  "6338"  "6488"  "6918"  "7898"  "8778"  "6938"  "7198"  "7788"
```

```
## [145] "7738"  "8358"  "9258"  "8058"  "8238"  "9298"  "9538"  "8449"  "9639"
## [154] "9989"  "11199" "11549" "17669" "8948"  "10698" "9988"  "10898" "11248"
## [163] "16558" "15998" "15690" "15750" "7975"  "7995"  "8195"  "9495"  "9995"
## [172] "11595" "9980"  "13295" "13845" "12290" "12940" "13415" "15985" "16515"
## [181] "18420" "18950" "16845" "19045" "21485" "22470" "22625"
## [1] "----------------------------"
```

```
# Os valores faltantes estao representados na base pelo caracter "?"
cols_with_missing_values <- c()
for(col in names(cars_df)) {
  if("?" %in% unique(cars_df[, col])) {
    cols_with_missing_values <- c(cols_with_missing_values, col)
  }
}

cols_with_missing_values # "V2"  "V6"  "V19" "V20" "V22" "V23" "V26"
```

```
## [1] "V2"  "V6"  "V19" "V20" "V22" "V23" "V26"
```

```
unique(cars_df[, "V6"])
```

```
## [1] "two"  "four" "?"
```

```
for(col in cols_with_missing_values) {
  cars_df[cars_df[, col] == "?", col] <- NA
}
unique(cars_df[, "V6"])
```

```
## [1] "two"  "four" NA
```

```
# Corrigindo o tipo dos dados
cars_df[, "V2"] <- as.numeric(cars_df[, "V2"])
cars_df[, "V19"] <- as.numeric(cars_df[, "V19"])
cars_df[, "V20"] <- as.numeric(cars_df[, "V20"])
cars_df[, "V22"] <- as.numeric(cars_df[, "V22"])
cars_df[, "V23"] <- as.numeric(cars_df[, "V23"])
cars_df[, "V26"] <- as.numeric(cars_df[, "V26"])

summary(cars_df) # Checando se as correcoes foram aplicadas
```

```
##        V1                V2           V3                 V4
##  Min.   :-2.000   Min.   : 65   Length:205         Length:205
##  1st Qu.: 0.000   1st Qu.: 94   Class :character   Class :character
##  Median : 1.000   Median :115   Mode  :character   Mode  :character
##  Mean   : 0.834   Mean   :122
##  3rd Qu.: 2.000   3rd Qu.:150
##  Max.   : 3.000   Max.   :256
##                   NA's   :41
##       V5                 V6                 V7                 V8
##  Length:205         Length:205         Length:205         Length:205
##  Class :character   Class :character   Class :character   Class :character
```

```
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       V9                 V10             V11           V12            V13
##  Length:205         Min.   : 86.6   Min.   :141   Min.   :60.3   Min.   :47.8
##  Class :character   1st Qu.: 94.5   1st Qu.:166   1st Qu.:64.1   1st Qu.:52.0
##  Mode  :character   Median : 97.0   Median :173   Median :65.5   Median :54.1
##                     Mean   : 98.8   Mean   :174   Mean   :65.9   Mean   :53.7
##                     3rd Qu.:102.4   3rd Qu.:183   3rd Qu.:66.9   3rd Qu.:55.5
##                     Max.   :120.9   Max.   :208   Max.   :72.3   Max.   :59.8
##
##       V14           V15                V16                 V17
##  Min.   :1488   Length:205         Length:205         Min.   : 61
##  1st Qu.:2145   Class :character   Class :character   1st Qu.: 97
##  Median :2414   Mode  :character   Mode  :character   Median :120
##  Mean   :2556                                         Mean   :127
##  3rd Qu.:2935                                         3rd Qu.:141
##  Max.   :4066                                         Max.   :326
##
##      V18                V19             V20             V21            V22
##  Length:205         Min.   :2.54   Min.   :2.07   Min.   : 7.0   Min.   : 48
##  Class :character   1st Qu.:3.15   1st Qu.:3.11   1st Qu.: 8.6   1st Qu.: 70
##  Mode  :character   Median :3.31   Median :3.29   Median : 9.0   Median : 95
##                     Mean   :3.33   Mean   :3.26   Mean   :10.1   Mean   :104
##                     3rd Qu.:3.59   3rd Qu.:3.41   3rd Qu.: 9.4   3rd Qu.:116
##                     Max.   :3.94   Max.   :4.17   Max.   :23.0   Max.   :288
##                     NA's   :4      NA's   :4                     NA's   :2
##       V23            V24             V25            V26
##  Min.   :4150   Min.   :13.0   Min.   :16.0   Min.   : 5118
##  1st Qu.:4800   1st Qu.:19.0   1st Qu.:25.0   1st Qu.: 7775
##  Median :5200   Median :24.0   Median :30.0   Median :10295
##  Mean   :5125   Mean   :25.2   Mean   :30.8   Mean   :13207
##  3rd Qu.:5500   3rd Qu.:30.0   3rd Qu.:34.0   3rd Qu.:16500
##  Max.   :6600   Max.   :49.0   Max.   :54.0   Max.   :45400
##  NA's   :2                                    NA's   :4
```

```r
# Substituindo valores NAs
# Mediana para dados numericos
for(col in c("V2", "V19", "V20", "V22", "V23", "V26")) {
  cars_df[is.na(cars_df[, col]), col] <- median(cars_df[, col], na.rm=TRUE)
}

# Moda para dados categoricos
cars_df$V6[is.na(cars_df$V6)] <- names(sort(table(cars_df$V6), decreasing=T))[1]

any(is.na(cars_df)) # FALSE: Ajustes ok
```

```
## [1] FALSE
```

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.

```
# Seleção de atributos

# Separando as variaveis categoricas que nao serao usadas para agrupamento
symboling <- "V1"
make <- "V3"

unique(cars_df[, symboling])
```

```
## [1]  3  1  2  0 -1 -2
```

```
unique(cars_df[, make])
```

```
##  [1] "alfa-romero"   "audi"          "bmw"           "chevrolet"
##  [5] "dodge"         "honda"         "isuzu"         "jaguar"
##  [9] "mazda"         "mercedes-benz" "mercury"       "mitsubishi"
## [13] "nissan"        "peugot"        "plymouth"      "porsche"
## [17] "renault"       "saab"          "subaru"        "toyota"
## [21] "volkswagen"    "volvo"
```

```
# Selecionando as colunas numericas
num_cars_df <- cars_df[, sapply(cars_df, is.numeric)]
num_cars_df[, symboling] <- NULL
num_cars_df[, make] <- NULL

names(num_cars_df)
```

```
##  [1] "V2"  "V10" "V11" "V12" "V13" "V14" "V17" "V19" "V20" "V21" "V22" "V23"
## [13] "V24" "V25" "V26"
```

```
# Selecionando as colunas categoricas
cat_cars_df <- cars_df[, sapply(cars_df, is.character)]
cat_cars_df[, symboling] <- NULL
cat_cars_df[, make] <- NULL

names(cat_cars_df)
```

```
## [1] "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V15" "V16" "V18"
```

```
head(cat_cars_df)
```

```
##    V4  V5   V6          V7  V8    V9  V15  V16  V18
## 1 gas std  two convertible rwd front dohc four mpfi
## 2 gas std  two convertible rwd front dohc four mpfi
## 3 gas std  two   hatchback rwd front ohcv  six mpfi
## 4 gas std four       sedan fwd front  ohc four mpfi
## 5 gas std four       sedan 4wd front  ohc five mpfi
## 6 gas std  two       sedan fwd front  ohc five mpfi
```

```r
# Como estamos trabalhando com algoritmos baseados em distancia vamos normalizar os # nossos dados de m
dim(num_cars_df)
```

```
## [1] 205  15
```

```r
# num_cars_df <- cbind(scale(num_cars_df[, 1:15]), num_cars_df[, c(16, 17)])
num_cars_df <- as.data.frame(scale(num_cars_df))

# Analisando as colunas categoricas a mais descritiva eh a V4, que informa se o carro eh movido a gas o
# Vamos aplicar entao a tecnica one-hot encoding nessa coluna
num_cars_df$gas <- as.numeric(cat_cars_df$V4 == "gas")
num_cars_df$diesel <- as.numeric(cat_cars_df$V4 == "diesel")

head(num_cars_df)
```

```
##       V2    V10    V11    V12    V13     V14     V17    V19    V20     V21
## 1 -0.176 -1.687 -0.425 -0.843 -2.015 -0.0145  0.0743  0.519 -1.837 -0.2876
## 2 -0.176 -1.687 -0.425 -0.843 -2.015 -0.0145  0.0743  0.519 -1.837 -0.2876
## 3 -0.176 -0.707 -0.231 -0.190 -0.542  0.5136  0.6026 -2.397  0.682 -0.2876
## 4  1.365  0.173  0.207  0.136  0.235 -0.4198 -0.4300 -0.515  0.459 -0.0359
## 5  1.365  0.107  0.207  0.229  0.235  0.5155  0.2184 -0.515  0.459 -0.5394
## 6 -0.176  0.173  0.263  0.183 -0.256 -0.0933  0.2184 -0.515  0.459 -0.4135
##       V22    V23    V24    V25    V26 gas diesel
## 1  0.1729 -0.264 -0.645 -0.545 0.0437   1      0
## 2  0.1729 -0.264 -0.645 -0.545 0.4251   1      0
## 3  1.2607 -0.264 -0.951 -0.690 0.4251   1      0
## 4 -0.0548  0.784 -0.186 -0.109 0.1015   1      0
## 5  0.2741  0.784 -1.104 -1.271 0.5457   1      0
## 6  0.1476  0.784 -0.951 -0.835 0.2665   1      0
```

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: A base de dados contem informacoes sobre carros e suas respectivas chances de risco de seguro. A base em questao possui valores faltantes os quais estavam representados pelo caracter "?". Tais valores faltantes influenciaram de modo negativo o tipo dos dados de modo que um tratamento adequado foi necessario.

Para tal tratamento, primeiro substituimos todos os valores "?" por NA de modo que fazer a coersão dos dados para o tipo numérico fosse mais simples. Como método de substituição de valores ausentes optamos por utilizar a mediana para os valores numéricos de modo a garantir que não iríamos alterar a distribuição dos dados. Já para os valores categóricos optamos por utilizar a moda, já que esta toma o valor mais frequente.

Os atributos escolhidos para serem usados na etapa de agrupamento foram os atributos numéricos, exceto as colunas symboling e a coluna make.

Também normalizamos os dados numéricos uma vez que os algoritmos de agrupamento são baseados em cálculos de distância.

Por fim, fizemos o processo de one hot encoding da coluna fuel-type, uma vez que essa possui informacoes relevantes sobre o tipo do carro, no caso se o carro é movido a gás ou diesel. Note que não normalizamos as colunas obtidas via one hot enconding.

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.

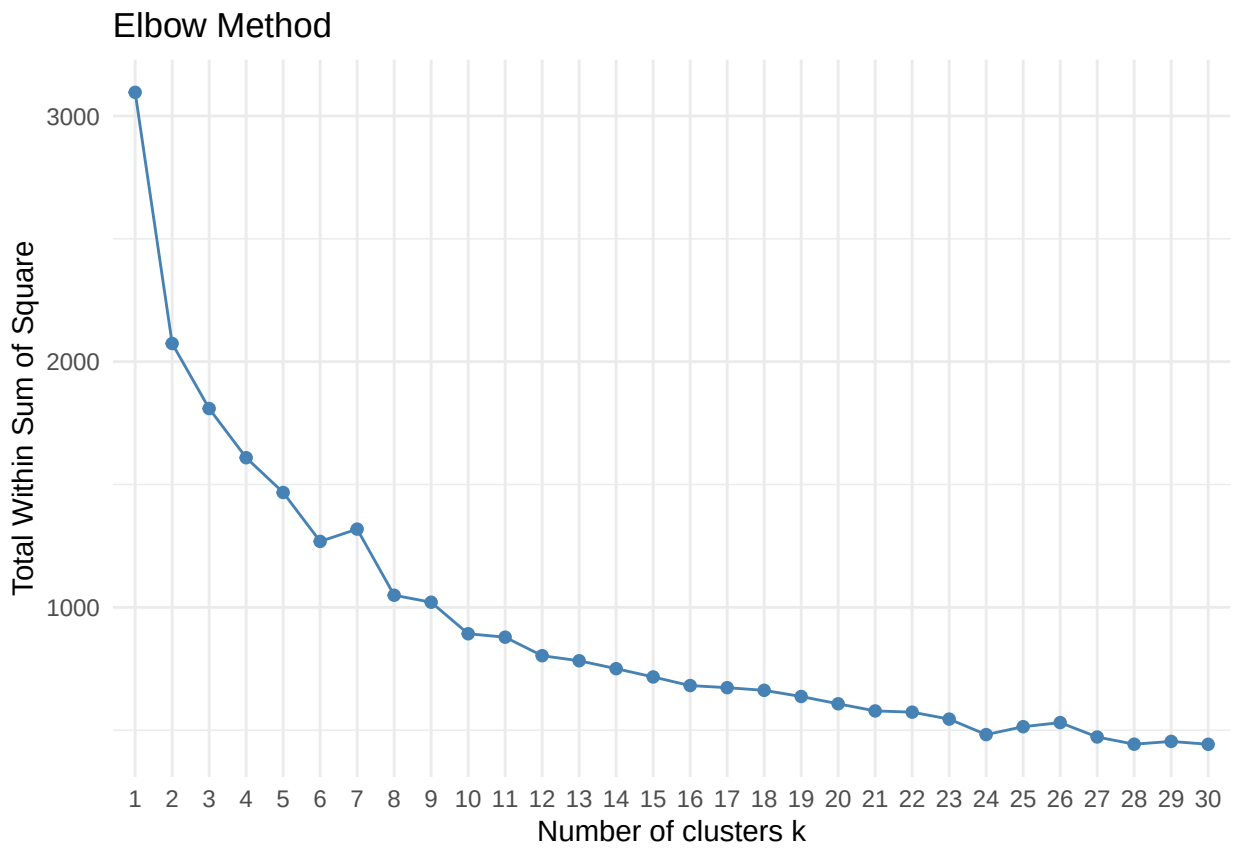
Implementações: Nos itens a seguir você implementará a geração de gráficos para a análise das distâncias intra-cluster e do coeficiente de silhueta. Em seguida, você implementará o agrupamento dos dados processados na atividade anterior com o algoritmo *K-means* utilizando o valor de K escolhido.

a) *Gráfico Elbow Curve:* Construa um gráfico com a soma das distâncias intra-cluster para K variando de 1 a 30.

```
# Construindo um gráfico com as distâncias intra-cluster
head(num_cars_df)
```

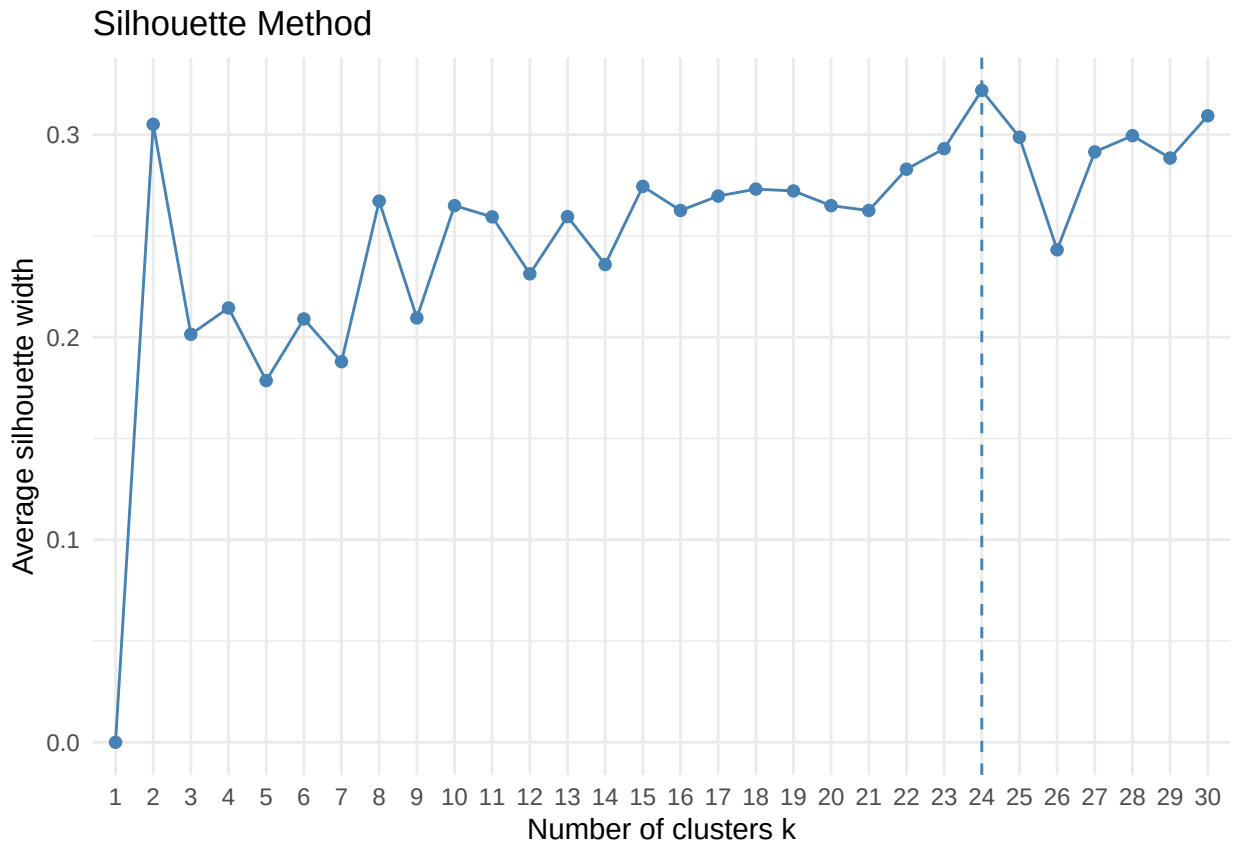
```
##       V2    V10    V11    V12    V13     V14     V17    V19    V20     V21
## 1 -0.176 -1.687 -0.425 -0.843 -2.015 -0.0145  0.0743  0.519 -1.837 -0.2876
## 2 -0.176 -1.687 -0.425 -0.843 -2.015 -0.0145  0.0743  0.519 -1.837 -0.2876
## 3 -0.176 -0.707 -0.231 -0.190 -0.542  0.5136  0.6026 -2.397  0.682 -0.2876
## 4  1.365  0.173  0.207  0.136  0.235 -0.4198 -0.4300 -0.515  0.459 -0.0359
## 5  1.365  0.107  0.207  0.229  0.235  0.5155  0.2184 -0.515  0.459 -0.5394
## 6 -0.176  0.173  0.263  0.183 -0.256 -0.0933  0.2184 -0.515  0.459 -0.4135
##       V22    V23    V24    V25    V26 gas diesel
## 1  0.1729 -0.264 -0.645 -0.545 0.0437   1      0
## 2  0.1729 -0.264 -0.645 -0.545 0.4251   1      0
## 3  1.2607 -0.264 -0.951 -0.690 0.4251   1      0
## 4 -0.0548  0.784 -0.186 -0.109 0.1015   1      0
## 5  0.2741  0.784 -1.104 -1.271 0.5457   1      0
## 6  0.1476  0.784 -0.951 -0.835 0.2665   1      0
```

```
fviz_nbclust(
  num_cars_df,
  FUNcluster = kmeans,
  method = "wss",
  k.max = 30
) + theme_minimal() + ggtitle("Elbow Method")
```



b) *Gráfico da Silhueta:* Construa um gráfico com o valor da silhueta para K variando de 1 a 30.

```
# Construindo um gráfico com os valores da silhueta
fviz_nbclust(
  num_cars_df,
  FUNcluster = kmeans,
  method = "silhouette",
  k.max = 30
) + theme_minimal() + ggtitle("Silhouette Method")
```



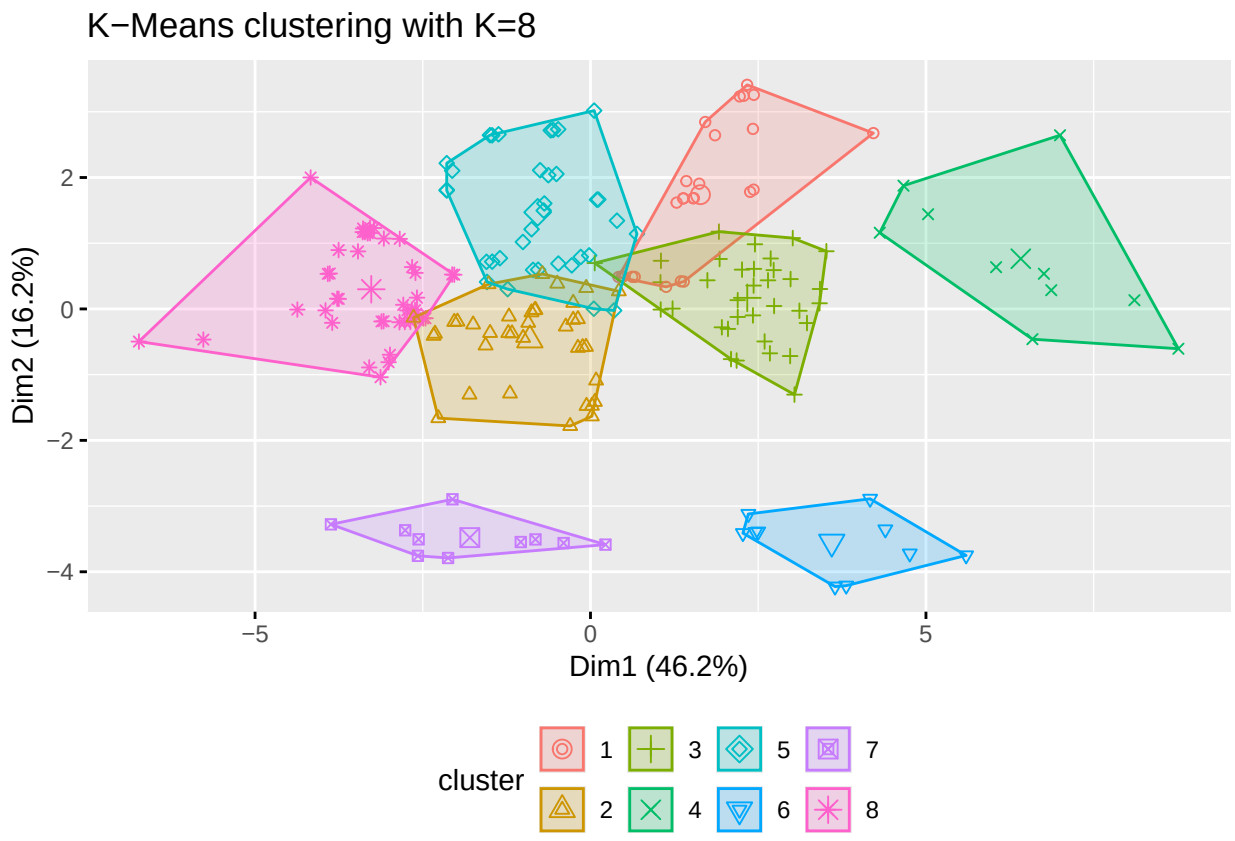
c) *Escolha do K:* Avalie os gráficos gerados nos itens anteriores e escolha o melhor valor de K com base nas informações desses gráficos e na sua análise. Se desejar, use também a função `NbClust` para ajudar nas análises. Com o valor de K definido, utilize o rótulo obtido para cada amostra, indicando 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
kmeans_cluster <- kmeans(num_cars_df, centers=8, nstart=30)
kmeans_cluster
```

```
## K-means clustering with 8 clusters of sizes 25, 37, 34, 10, 35, 10, 10, 44
##
## Cluster means:
##         V2     V10     V11    V12     V13    V14    V17    V19     V20    V21
## 1  0.89042 -0.3236  0.0602  0.298 -0.9105  0.581  0.864  0.784  0.452 -0.376
## 2 -0.81935 -0.1880 -0.2066 -0.364  0.4721 -0.414 -0.398  0.177 -0.523 -0.335
## 3 -0.10209  0.9684  1.1007  0.846  0.9190  0.820  0.299  0.609 -0.500 -0.348
## 4  0.00314  1.7027  1.7865  1.973  0.0512  2.243  3.074  1.066  0.513 -0.431
## 5  0.44198 -0.4938 -0.2963 -0.398 -0.7421 -0.332 -0.429 -0.349  0.329 -0.362
## 6 -0.08175  1.9252  1.4437  1.605  1.3158  1.692  0.884  0.936  0.956  2.834
## 7 -0.55337 -0.0542 -0.1531 -0.185  0.4113 -0.373 -0.461 -0.603  0.507  3.139
## 8  0.05402 -0.8258 -1.1747 -0.972 -0.4036 -1.161 -0.840 -1.105 -0.143 -0.207
##       V22      V23     V24    V25    V26 gas diesel
## 1  1.2425  0.00818 -0.8223 -0.684  0.513   1      0
## 2 -0.5211 -0.65243  0.2350  0.217 -0.550   1      0
## 3  0.5933  0.24432 -0.8608 -0.827  0.440   1      0
```

```
## 4  2.0145 -0.43204 -1.5621 -1.750  3.006   1      0
## 5 -0.0584  0.91557 -0.3218 -0.234 -0.263   1      0
## 6  0.0793 -1.74221 -0.0641 -0.458  1.060   0      1
## 7 -1.0768 -1.15525  1.6173  1.619 -0.378   0      1
## 8 -0.9109  0.38361  1.1928  1.165 -0.798   1      0
##
## Clustering vector:
##   [1] 5 5 5 5 1 5 3 3 3 1 5 5 1 1 3 4 4 4 8 8 8 8 8 5 8 8 8 5 2 1 8 8 8 8 8 8 8
##  [38] 5 5 5 5 5 5 2 8 8 5 4 4 4 8 8 8 8 8 5 5 5 5 2 2 2 2 7 2 3 7 6 6 6 6 4 4 4
##  [75] 4 3 8 8 8 5 5 5 1 1 1 5 5 5 5 8 7 8 8 8 8 8 8 8 8 2 2 3 3 3 1 1 1 3 6 3 6
## [112] 3 6 3 6 3 6 3 8 5 8 8 8 2 1 1 1 1 1 1 5 5 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2
## [149] 2 2 8 8 8 2 2 2 2 2 7 7 8 2 2 8 8 5 5 1 1 1 1 1 1 2 7 2 2 2 1 1 3 3 7 2 7
## [186] 2 2 7 2 5 5 3 7 5 3 3 3 3 3 3 3 3 3 6 3
##
## Within cluster sum of squares by cluster:
## [1] 193.0 168.9 176.6  83.9 186.8  41.0  30.9 149.0
##  (between_SS / total_SS =  66.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# Construindo um gráfico de dispersão
fviz_cluster_obj <- fviz_cluster(kmeans_cluster, data=num_cars_df, geom="point", stand = FALSE) + ggtitl
fviz_cluster_obj
```



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: Sabemos que pelo método Elbow o objetivo é selecionar um valor de K que minimize o valor SSE intra-cluster. Já no método Slhouette queremos selecionar K que maximize as médias intra-clusters e entre-clusters. Deste modo, faz sentido selecionarmos um valor de K intermediário entre os dois métodos. Sendo assim, selecionamos o valor de $K = 8$, uma vez que para este valor temos um ponto de mudança no método Elbow e no método Silhouette. Vale ressaltar que o método da Silhueta apresenta uma robustez maior com relação ao método Elbow, principalmente quando temos dados com formatos irregulares.

O gráfico de dispersão com o resultado dos agrupamentos parecem agrupar bem os valores posicionados na região central superior, porém os grupos formados na região central inferior não possuem uma distância entre-grupos ideal, o que talvez indique que o valor de K selecionado talvez não tenha sido ideal ou talvez a complexidade dos dados é demais para o algoritmo K-Means, que como visto em aula, possui suas limitações.

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.

```
# Experimento com valores de eps e minPts
db1 <- dbscan::dbscan(num_cars_df, eps=1.6, minPts=4)
print(db1)
```

```
## DBSCAN clustering for 205 objects.
## Parameters: eps = 1.6, minPts = 4
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 9 cluster(s) and 88 noise points.
##
##  0  1  2  3  4  5  6  7  8  9
## 88 64  4  5  4  5 16  9  6  4
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

```
# Experimento com valores de eps e minPts
db2 <- dbscan::dbscan(num_cars_df, eps=1.2, minPts=8)
print(db2)
```

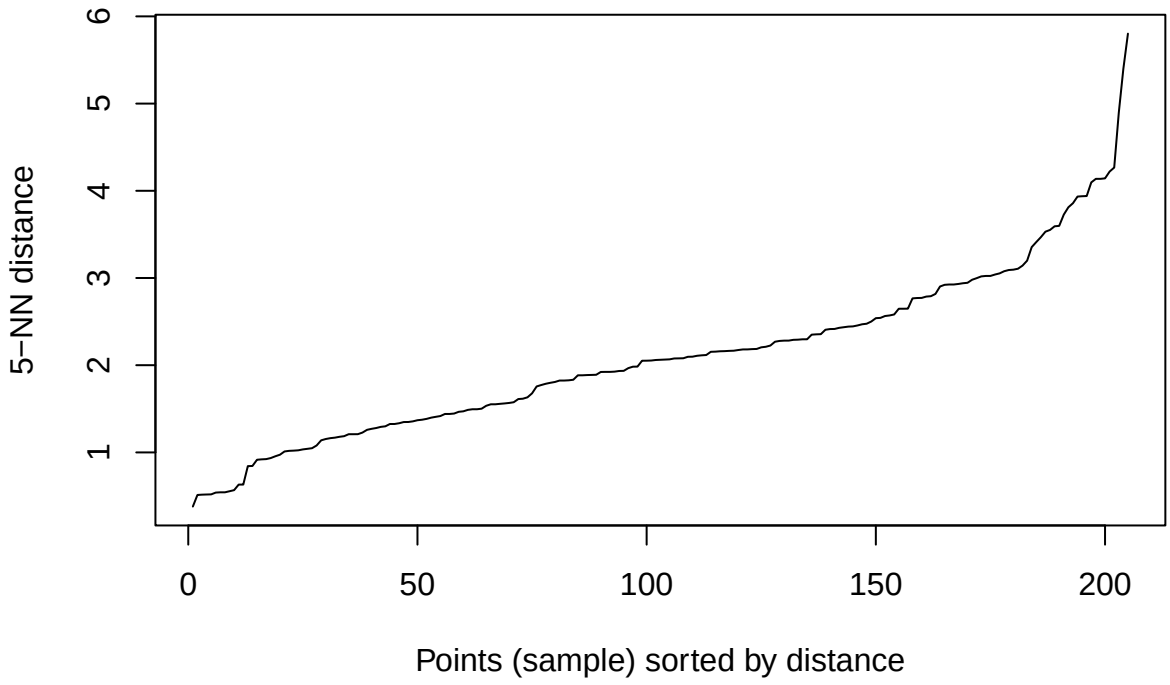
```
## DBSCAN clustering for 205 objects.
## Parameters: eps = 1.2, minPts = 8
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 2 cluster(s) and 180 noise points.
##
##   0   1   2
## 180  10  15
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

```r
# Experimento com valores de eps e minPts
db3 <- dbscan::dbscan(num_cars_df, eps=2.1, minPts=5)
print(db3)
```

```
## DBSCAN clustering for 205 objects.
## Parameters: eps = 2.1, minPts = 5
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 4 cluster(s) and 68 noise points.
##
##   0   1   2   3   4
##  68 112   5   8  12
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

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.

```r
# Encontrando o melhor eps com o kNNdistplot
minPts = 3 # Dado que para esse valor tivemos o menor valor de outliers
dbscan::kNNdistplot(num_cars_df, k=5)
```



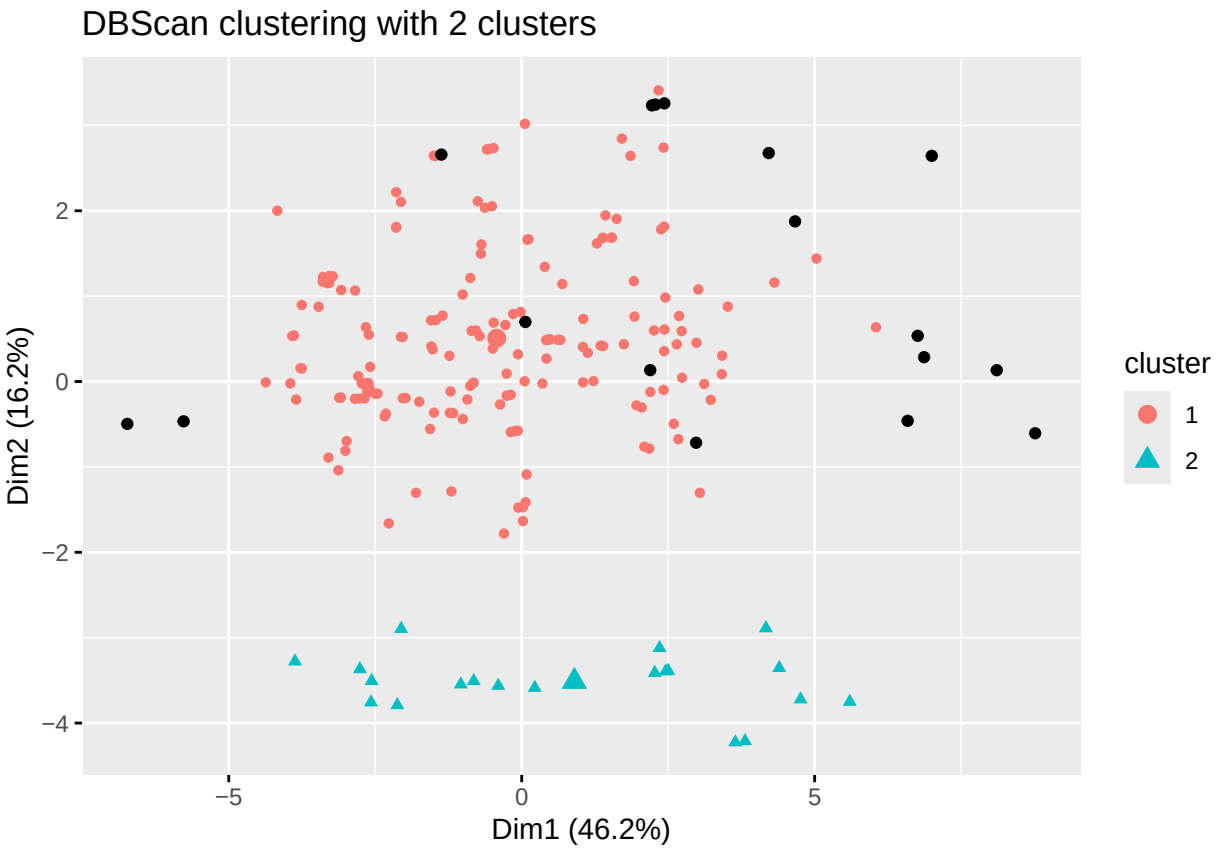
```
# Valor otimo de distancia eh aproximadamente 3.1, que eh aonde temos um ponto de inflexao
```

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::dbscan(num_cars_df, eps=3.1, minPts=5)
print(db)
```

```
## DBSCAN clustering for 205 objects.
## Parameters: eps = 3.1, minPts = 5
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 2 cluster(s) and 17 noise points.
##
##   0   1   2
##  17 168  20
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

```
# Construindo um gráfico de dispersão
fviz_cluster(db,
             data=num_cars_df,
             stand=F,
             ellipse=F,
             show.clust.cent=T,
             geom="point") + ggtitle("DBScan clustering with 2 clusters")
```



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: Analisamos 3 possibilidades para os valores de minPts, no caso 4, 8 e 5 e alguns valores de distancia. Com base nos resultados obtidos selecionamos o valor de minPts com a menor quantidade de outliers. Após isso, usamos o gráfico *kNNDisplot* para avaliar quando ocorria o ponto de inflexão no gráfico de distâncias de modo a selecionar o melhor valor para *eps*, que no caso foi aproximadamente 3.1. Feito isso, obtivemos uma separação razoável e interpretável entre os dados, na qual temos uma concentração dos dados na região central e na região central inferior, cada uma representando um agrupamento, e alguns pontos identificados como outliers, não fazendo estes parte de nenhum grupo.

Atividade 4 – Comparando os Algoritmos

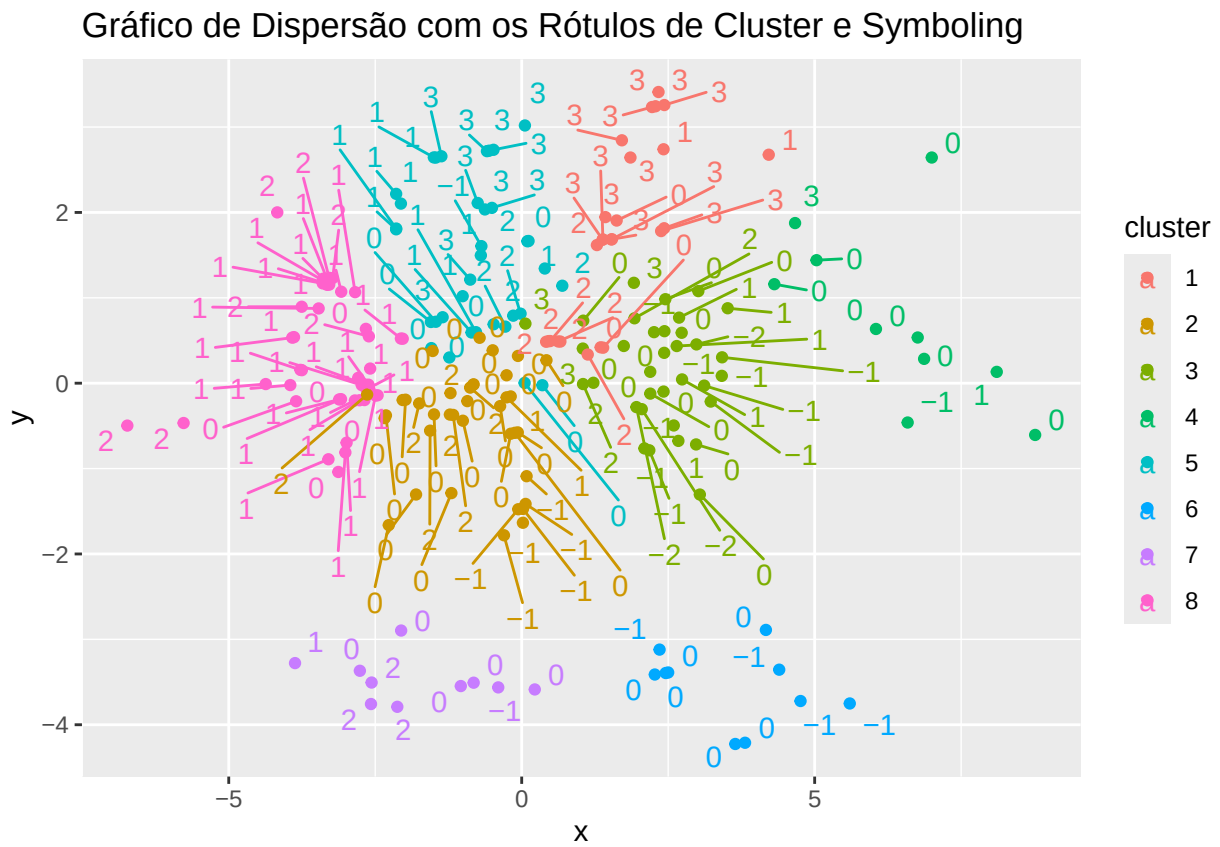
```
cluster_results <- as.factor(kmeans_cluster$cluster)

# Adicionando as classes ao dataframe
num_cars_df <- cbind(num_cars_df, cluster=cluster_results, symboling=cars_df[, symboling], make=cars_df
dim(num_cars_df)
```

```
## [1] 205  20
```

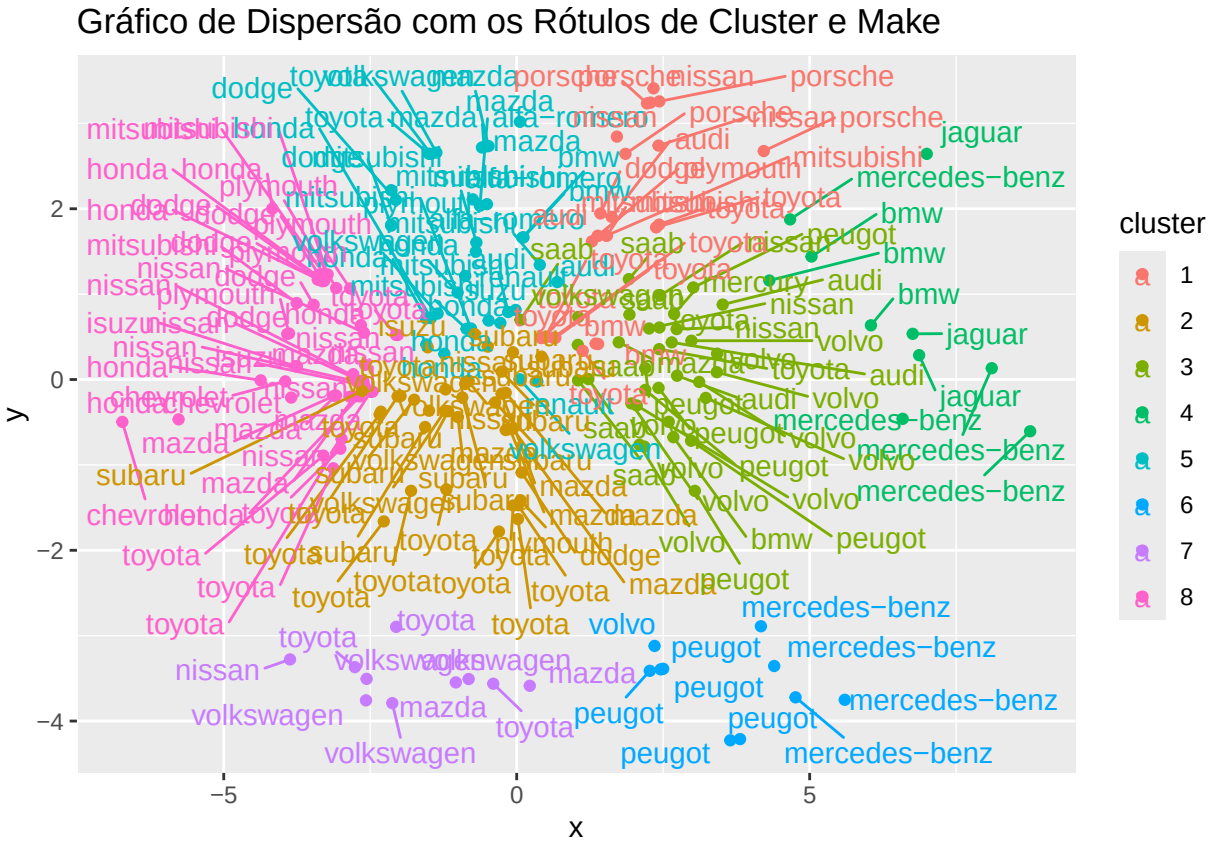
```
kmeans_fviz_df <- fviz_cluster_obj$data
kmeans_fviz_df$symboling <- cars_df[, symboling]
kmeans_fviz_df$make <- cars_df[, make]

head(kmeans_fviz_df)
```

```
##   name      x     y coord cluster symboling        make
## 1    1 -0.629 2.035 4.537       5         3 alfa-romero
## 2    2 -0.508 2.054 4.475       5         3 alfa-romero
## 3    3  0.394 1.343 1.960       5         1 alfa-romero
## 4    4 -0.145 0.791 0.646       5         2        audi
## 5    5  1.284 1.618 4.265       1         2        audi
## 6    6  0.693 1.141 1.782       5         2        audi
```

```
ggplot(kmeans_fviz_df, aes(x = x, y = y, color = cluster, label = symboling)) +
  geom_point() +
  geom_text_repel(max.overlaps=40) +  # Adiciona os rótulos sem sobreposição
  ggtitle("Gráfico de Dispersão com os Rótulos de Cluster e Symboling")
```




```
ggplot(kmeans_fviz_df, aes(x = x, y = y, color = cluster, label = make)) +
  geom_point() +
  geom_text_repel(max.overlaps=70) +  # Adiciona os rótulos sem sobreposição
  ggtitle("Gráfico de Dispersão com os Rótulos de Cluster e Make")
```



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 para cada método?

c) Analisando o campo `symboling` e o grupo designado para cada amostra, os agrupamentos conseguiram separar os níveis de risco? Dica: utilize o comando `table` para visualizar os dados de cada grupo em relação ao campo `symboling`.

d) Analisando o campo `make` que contém as marcas dos carros, os agrupamentos conseguiram separar as marcas? Dica: utilize o comando `table` para visualizar os dados de cada grupo em relação ao campo `make`.

Respostas: Nas configurações apresentadas acima o algoritmo K-Means se adaptou melhor ao problema, uma vez que retornou uma quantidade mais adequada de clusters se comparado ao método DBScan, isto é, o nível de informação do K-Means foi mais assertivo.

O método K-Means retornou 8 clusters e o métod DBScan retornou 2 clusters.

Analisando o campo Symboling podemos perceber que há uma confusão do algoritmo de agrupamento com relação as classes reais. Há alguns acertos, porem também há casos de overlap. A mesma situação ocorre quando olhamos para a coluna Make (marca dos carros). Tal fato pode estar ligado ao número não ideal de clusters, uma vez que o mesmo não é aderente ao número de valores possíveis na coluna Symboling ou Make. Também podemos estar obtendo resultados não tão bons devido ao fato de estarmos usando todas as colunas numéricas, ao invés de usar um subconjunto das mesmas. Todos esses pontos são factíveis de investigação.