Reduzindo mortalidade no tráfego por aprendizado não supervisionado

Grupo 5

1. O banco de dados e o seu formato

Nos Estados Unidos, a taxa de acidentes rodoviários fatais vem diminuindo constantemente desde os anos 80, porém, nos últimos dez anos, houve uma estagnação nessa redução. Juntamente com o aumento do número de milhas percorridas no país, o número total de mortes devido ao tráfego atingiu o maior valor da última década e está aumentando rapidamente.

Por pedido do Departamento de Transportes dos Estados Unidos, foi investigado como elaborar uma estratégia para reduzir a incidência de acidentes de trânsito em todo o país. Observando demograficamente as vítimas de acidentes de trânsito de cada estado dos Estados Unidos, foi descoberto que há muita variação entre os estados. Agora, procura-se verificar se existem padrões nessa variação para fornecer sugestões para um plano de ação de políticas públicas. Em particular, em vez de implementar um plano nacional financeiramente custoso, concentra-se em grupos de estados com perfis semelhantes. O objetivo é encontrar grupos de maneira estatisticamente sólida e comunicar o resultado de forma eficaz.

Para realizar essas tarefas, será utilizado manipulação de dados, análise gráfica, redução de dimensionalidade e aprendizado não supervisionado.

Os dados fornecidos foram originalmente coletados pela Administração Nacional de Segurança no Trânsito nas Rodovias e pela Associação Nacional de Comissários de Seguros. Esse conjunto de dados específico foi compilado e lançado como um arquivo CSV pelo FiveThirtyEight sob a licença CC-BY4.0.

```
# Verifica o nome do diretório atual
(current_dir <- getwd())
```

```
## [1] "C:/Users/willi/Documents/UNB/6º Semestre/Laboratório de Estatística 1"
```

```
# Lista os nomes dos arquivos presentes na pasta
(file_list <- list.files())
```

```
## [1] "car_accident.jpeg"    "Código.R"             "miles-driven.txt"
## [4] "Relatório-Final.html" "Relatório-Final.Rmd"  "Relatório Final.Rmd"
## [7] "road-accidents.txt"
```

```
# visualiza as primeiras 20 linhas de road-accidents.csv
(accidents_head <- readLines("road-accidents.txt", n=20))
```

```
##  [1] "##### LICENSE #####"
##  [2] "# This data set is modified from the original at fivethirtyeight (https://github.com/fivethirtyeight/data/tree/master/bad-drivers)"
##  [3] "# and it is released under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/)"
##  [4] "##### COLUMN ABBREVIATIONS #####"
##  [5] "# drvr_fatl_col_bmiles = Number of drivers involved in fatal collisions per billion miles (2011)"
##  [6] "# perc_fatl_speed = Percentage Of Drivers Involved In Fatal Collisions Who Were Speeding (2009)"
##  [7] "# perc_fatl_alcohol = Percentage Of Drivers Involved In Fatal Collisions Who Were Alcohol-Impaired (2011)"
##  [8] "# perc_fatl_1st_time = Percentage Of Drivers Involved In Fatal Collisions Who Had Not Been Involved In Any Previous Accidents (2011)"
##  [9] "##### DATA BEGIN #####"
## [10] "state|drvr_fatl_col_bmiles|perc_fatl_speed|perc_fatl_alcohol|perc_fatl_1st_time"
## [11] "Alabama|18.8|39|30|80"
## [12] "Alaska|18.1|41|25|94"
## [13] "Arizona|18.6|35|28|96"
## [14] "Arkansas|22.4|18|26|95"
## [15] "California|12|35|28|89"
## [16] "Colorado|13.6|37|28|95"
## [17] "Connecticut|10.8|46|36|82"
## [18] "Delaware|16.2|38|30|99"
## [19] "District of Columbia|5.9|34|27|100"
## [20] "Florida|17.9|21|29|94"
```

2. Importação e verificação da estrutura do banco de dados

O banco de dados está delimitado pela barra reta '|' e contém 5 variáveis:

- Estado nos Estados Unidos;
- Número de motoristas envolvidos em acidentes fatais por bilhão de milhas;
- Percentual de motoristas envolvidos em acidentes fatais que estavam acima da velocidade da via;
- Percentual de motoristas envolvidos em acidentes fatais que estavam embriagados;
- Percentual de motoristas envolvidos em acidentes fatais que nunca estiveram envolvidos em algum acidente previamente.

```
# Carrega a biblioteca de pacotes tidyverse
require(tidyverse)

# Importação de road-accidents.csv
car_acc <- read_delim("road-accidents.txt", comment = '#', delim = '|')

# Renomeia as variáveis
car_new <- car_acc
colnames(car_new) <- c('Estado', 'Taxa Fatal',
                       '% - Velocidade', '% - Alcoolizado',
                       '% - Não Reincidente')
```

```
# Salva o número de linhas e colunas
(rows_and_cols <- dim(car_new))
```

```
## [1] 51  5
```

```
# Verifica a estrutura do banco de dados
str(car_new)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 51 obs. of  5 variables:
##  $ Estado            : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ Taxa Fatal        : num  18.8 18.1 18.6 22.4 12 13.6 10.8 16.2 5.9 17.9 ...
##  $ % - Velocidade    : num  39 41 35 18 35 37 46 38 34 21 ...
##  $ % - Alcoolizado   : num  30 25 28 26 28 28 36 30 27 29 ...
##  $ % - Não Reincidente: num  80 94 96 95 89 95 82 99 100 94 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   state = col_character(),
##   ..   drvr_fatl_col_bmiles = col_double(),
##   ..   perc_fatl_speed = col_double(),
##   ..   perc_fatl_alcohol = col_double(),
##   ..   perc_fatl_1st_time = col_double()
##   .. )
```

```
# Visualiza as últimas 6 linhas do banco
car_new %>% tail() %>% kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

Estado	Taxa Fatal	% - Velocidade	% - Alcoolizado	% - Não Reincidente
Vermont	13.6	30	30	95
Virginia	12.7	19	27	88
Washington	10.6	42	33	86
West Virginia	23.8	34	28	87
Wisconsin	13.8	36	33	84
Wyoming	17.4	42	32	90

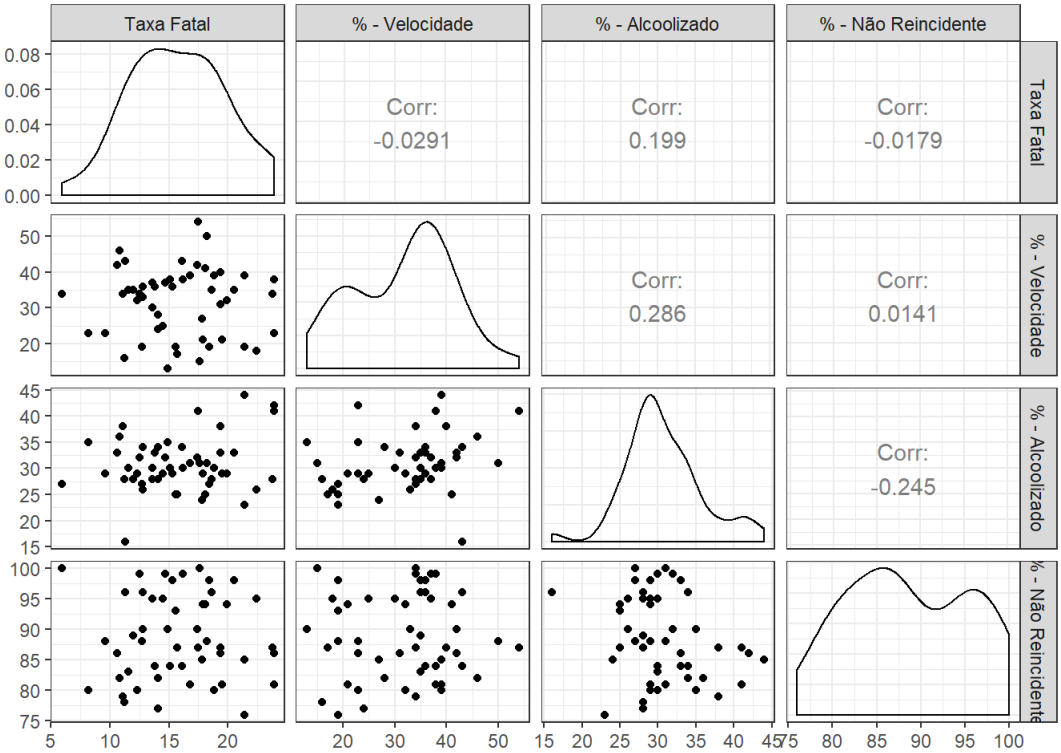
3. Resumo dos dados

Para compreender melhor o banco de dados, são calculadas medidas-resumos e gráficos são elaboradas, referentes ao banco de dados. A visualização gráfica é últil para ter um conhecimento prévio sobre a distribuição das variáveis. Geralmente, é uma boa idéia verificar a relação entre as colunas duas a duas através de um gráfico de dispersão pareado.

```
dat_summ <- summary(car_new)
kable(dat_summ) %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

Estado	Taxa Fatal	% - Velocidade	% - Alcoolizado	% - Não Reincidente
Length:51	Min. : 5.90	Min. :13.00	Min. :16.00	Min. : 76.00
Class :character	1st Qu.:12.75	1st Qu.:23.00	1st Qu.:28.00	1st Qu.: 83.50
Mode :character	Median :15.60	Median :34.00	Median :30.00	Median : 88.00
	Mean :15.79	Mean :31.73	Mean :30.69	Mean : 88.73
	3rd Qu.:18.50	3rd Qu.:38.00	3rd Qu.:33.00	3rd Qu.: 95.00
	Max. :23.90	Max. :54.00	Max. :44.00	Max. :100.00

```
# Retira a coluna de estado para a construção do gráfico de dispersão pareado
require(GGally)
car_new %>%
  select(-Estado) %>%
  ggpairs() + theme_bw()
```



4. Quantificação da associação de características e

acidentes

Já pode-se verificar algumas relações interessantes entre a variável alvo (o número de acidentes fatais) e as variáveis auxiliares (percentual de velocidade, percentual de motoristas alcoolizados e percentual de motoristas com acidentes não reincidentes).

Para quantificar as relações observadas nos gráficos de dispersão, calcula-se a matriz de coeficientes de correlação de Pearson. O coeficiente de correlação de Pearson é um dos métodos mais comuns para quantificar a correlação entre variáveis e, por convenção, os seguintes limites foram usados para o valor absoluto do mesmo:

- Até 0,2 = correlação muito fraca;
- De 0,2 até 0,5 = correlação fraca;
- De 0,5 até 0,8 = correlação moderada;
- De 0,8 até 0,9 = correlação forte;
- A partir de 0,9 = correlação muito forte.

```
# Usando pipes, remove a coluna Estado e calcula o coeficiente de correlação para todos os pares
corr_col <- car_new %>%
  select(-Estado) %>%
  cor()
corr_col %>% kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

	Taxa Fatal	% - Velocidade	% - Alcoolizado	% - Não Reincidente
Taxa Fatal	1.0000000	-0.0290801	0.1994263	-0.0179419
% - Velocidade	-0.0290801	1.0000000	0.2862442	0.0140662
% - Alcoolizado	0.1994263	0.2862442	1.0000000	-0.2454551
% - Não Reincidente	-0.0179419	0.0140662	-0.2454551	1.0000000

5. Ajuste de modelos de regressão linear

Na tabela de correlação, vê-se que a quantidade de acidentes fatais está mais correlacionada com o consumo de álcool (primeira linha). Além disso, também percebe-se que algumas das variáveis auxiliares estão correlacionados entre si, por exemplo, velocidade e consumo de álcool estão positivamente correlacionados. Portanto, existe interesse em calcular a associação da variável alvo com cada variável auxiliar, enquanto considera-se o efeito das demais variáveis auxiliares. Isso pode ser feito usando regressão linear.

Tanto a regressão linear quanto a correlação medem quanto as demais variáveis estão associadas ao resultado (acidentes fatais). Ao comparar os coeficientes de regressão com os coeficientes de correlação, ficará evidente que eles são ligeiramente diferentes. A razão para isso é que a regressão linear calcula a associação de uma variável a um resultado, dada a associação com todas as outras variáveis, o que não é considerado no cálculo dos coeficientes de correlação.

Um caso interessante é quando o coeficiente de correlação e o coeficiente de regressão das mesmas característica têm sinais opostos. Por exemplo, quando uma variável A está correlacionada positivamente com o resultado Y, mas também correlacionado positivamente com uma variável diferente B, isso tem um efeito negativo em Y, então a correlação indireta (A -> B -> Y) pode sobrecarregar a correlação direta (A- > Y). Nesse caso, o coeficiente de regressão para a variável A pode ser positivo, enquanto o coeficiente de correlação linear é negativo. Isso, às vezes, é chamado de multicolinearidade. Será estudado se a regressão múltipla pode revelar esse fenômeno.

Foram ajustados oito modelos:

- Apenas com intercepto;
- Apenas com % - Velocidade;
- Apenas com % - Alcoolizado;
- Apenas com % - Não Reincidente;
- Apenas com % - Velocidade e % - Alcoolizado;
- Apenas com % - Velocidade e % - Não Reincidente;
- Apenas com % - Alcoolizado e % - Não Reincidente;
- Com as 3 variáveis auxiliares.

```
# Use lm para ajustar os modelos de regressão linear
mod1 <- lm(`Taxa Fatal` ~ -1, data = car_new)
mod2 <- lm(`Taxa Fatal` ~ `% - Velocidade`, data = car_new)
mod3 <- lm(`Taxa Fatal` ~ `% - Alcoolizado`, data = car_new)
mod4 <- lm(`Taxa Fatal` ~ `% - Não Reincidente`, data = car_new)
mod5 <- lm(`Taxa Fatal` ~ `% - Velocidade` + `% - Alcoolizado`, data = car_new)
mod6 <- lm(`Taxa Fatal` ~ `% - Velocidade` + `% - Não Reincidente`, data = car_new)
mod7 <- lm(`Taxa Fatal` ~ `% - Alcoolizado` + `% - Não Reincidente`, data = car_new)
mod8 <- lm(`Taxa Fatal` ~ . - Estado, data = car_new)

# Coeficientes de regressão e resumo do modelo
summary(mod1);
```

```
##
## Call:
## lm(formula = `Taxa Fatal` ~ -1, data = car_new)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   5.90  12.75  15.60  18.50  23.90
##
## No Coefficients
##
## Residual standard error: 16.31 on 51 degrees of freedom
```

```
coef(mod2);summary(mod2)
```

```
##      (Intercept) `% - Velocidade`
##      16.18495487      -0.01244295
```

```
##
## Call:
## lm(formula = `Taxa Fatal` ~ `% - Velocidade`, data = car_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8619 -3.1114 -0.3485  2.7439  8.1879
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      16.18495    2.02416   7.996 1.94e-10 ***
## `% - Velocidade` -0.01244    0.06110  -0.204    0.839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.162 on 49 degrees of freedom
## Multiple R-squared:  0.0008457,  Adjusted R-squared:  -0.01955
## F-statistic: 0.04147 on 1 and 49 DF,  p-value: 0.8395
```

```
coef(mod3);summary(mod3)
```

```
##       (Intercept) `% - Alcoolizado`
##         10.8751199          0.1601718
```

```
##
## Call:
## lm(formula = `Taxa Fatal` ~ `% - Alcoolizado`, data = car_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.2998 -2.4303 -0.2201  3.1600  8.4401
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        10.8751     3.4971   3.110  0.00312 **
## `% - Alcoolizado`   0.1602     0.1124   1.425  0.16061
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.08 on 49 degrees of freedom
## Multiple R-squared:  0.03977,    Adjusted R-squared:  0.02017
## F-statistic: 2.029 on 1 and 49 DF,  p-value: 0.1606
```

```
coef(mod4);summary(mod4)
```

```
##           (Intercept) `% - Não Reincidente`
##           16.73297175           -0.01062576
```

```
##
## Call:
## lm(formula = `Taxa Fatal` ~ `% - Não Reincidente`, data = car_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7704 -3.0373 -0.1448  2.7977  8.0808
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           16.73297    7.52798   2.223   0.0309 *
## `% - Não Reincidente` -0.01063    0.08459  -0.126   0.9006
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.163 on 49 degrees of freedom
## Multiple R-squared:  0.0003219,  Adjusted R-squared:  -0.02008
## F-statistic: 0.01578 on 1 and 49 DF,  p-value: 0.9006
```

```
coef(mod5);summary(mod5)
```

```
##       (Intercept)  `% - Velocidade` `% - Alcoolizado`
##       11.48705914       -0.04015906        0.18174910
```

```
##
## Call:
## lm(formula = `Taxa Fatal` ~ `% - Velocidade` + `% - Alcoolizado`,
##     data = car_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1289 -2.6867  0.1603  3.1234  8.5894
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       11.48706    3.64665   3.150  0.00281 **
## `% - Velocidade`  -0.04016    0.06290  -0.639  0.52618
## `% - Alcoolizado`  0.18175    0.11806   1.539  0.13025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.105 on 48 degrees of freedom
## Multiple R-squared:  0.04786,    Adjusted R-squared:  0.008185
## F-statistic: 1.206 on 2 and 48 DF,  p-value: 0.3082
```

```
coef(mod6);summary(mod6)
```

```
##           (Intercept)      `% - Velocidade` `% - Não Reincidente`
##           17.10307049           -0.01233741           -0.01038556
```

```
##
## Call:
## lm(formula = `Taxa Fatal` ~ `% - Velocidade` + `% - Não Reincidente`,
##     data = car_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7450 -3.0583 -0.3028  2.7767  8.1070
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           17.10307    7.82510   2.186   0.0338 *
## `% - Velocidade`      -0.01234    0.06173  -0.200   0.8424
## `% - Não Reincidente` -0.01039    0.08544  -0.122   0.9038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.205 on 48 degrees of freedom
## Multiple R-squared:  0.001153,   Adjusted R-squared:  -0.04047
## F-statistic: 0.02771 on 2 and 48 DF,  p-value: 0.9727
```

```
coef(mod7);summary(mod7)
```

```
##          (Intercept)     `% - Alcoolizado` `% - Não Reincidente`
##           8.94168201           0.16667671           0.01954147
```

```
##
## Call:
## lm(formula = `Taxa Fatal` ~ `% - Alcoolizado` + `% - Não Reincidente`,
##     data = car_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4961 -2.3726 -0.1502  3.1349  8.4913
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           8.94168    9.24566   0.967    0.338
## `% - Alcoolizado`     0.16668    0.11712   1.423    0.161
## `% - Não Reincidente` 0.01954    0.08636   0.226    0.822
##
## Residual standard error: 4.12 on 48 degrees of freedom
## Multiple R-squared:  0.04079,    Adjusted R-squared:  0.0008271
## F-statistic: 1.021 on 2 and 48 DF,  p-value: 0.368
```

```
coef(mod8);summary(mod8)
```

```
##          (Intercept)     `% - Velocidade`     `% - Alcoolizado`
##           9.06498048          -0.04180041           0.19086404
## `% - Não Reincidente`
##           0.02473301
```

```
##
## Call:
## lm(formula = `Taxa Fatal` ~ . - Estado, data = car_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3704 -2.7142  0.2575  3.1929  8.6603
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           9.06498    9.30299   0.974    0.335
## `% - Velocidade`     -0.04180    0.06377  -0.655    0.515
## `% - Alcoolizado`     0.19086    0.12347   1.546    0.129
## `% - Não Reincidente` 0.02473    0.08724   0.284    0.778
##
## Residual standard error: 4.145 on 47 degrees of freedom
## Multiple R-squared:  0.04948,    Adjusted R-squared:  -0.01119
## F-statistic: 0.8156 on 3 and 47 DF,  p-value: 0.4917
```

6. Executando análise de componentes principais nos dados padronizados

Verificou-se que o consumo de álcool está fracamente associado ao número de acidentes fatais nos estados. Isso pode levar a concluir que o consumo de álcool deve ser o foco de novas investigações e talvez as estratégias devam dividir os estados entre alto ou baixo consumo de álcool nos acidentes. Mas também existem associações entre o consumo de álcool e os outras duas características, então pode valer a pena tentar dividir os estados de uma maneira que represente todas as três características.

Uma maneira de agrupar os dados é usar a análise de componentes principais para visualizar dados em espaço dimensional reduzido, em que pode-se tentar captar padrões. A análise de componentes principais usa a variação absoluta para calcular a variação geral explicada para cada componente principal, portanto, é importante que as características estejam em uma escala semelhante (a menos que exixsta algum motivo específico para que uma característica seja ponderada com maior peso).

As características serão padronizadas, isto é, serão transformadas para obter média zero e desvio padrão um.

```
# Centraliza e padroniza as três colunas
car_acc_standised <- car_new %>%
  mutate(`% - Velocidade` = scale(`% - Velocidade`),
         `% - Alcoolizado` = scale(`% - Alcoolizado`),
         `% - Não Reincidente` = scale(`% - Não Reincidente`))

# PCA
pca_fit <- princomp(car_acc_standised[,c("% - Velocidade", "% - Alcoolizado",
                                         "% - Não Reincidente")])

# Proporção de variância explicada por cada componente
(pr_var <- pca_fit$sdev^2)
```

```
##    Comp.1    Comp.2    Comp.3
## 1.3433259 0.9940208 0.6038298
```

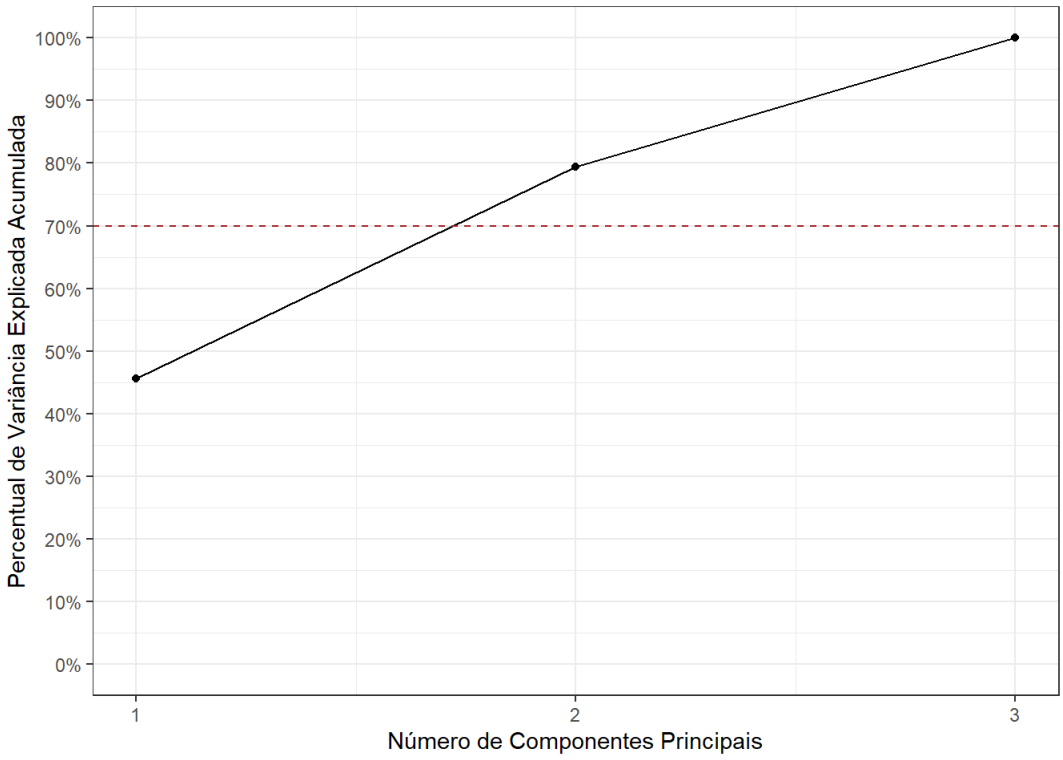
```
(pve <- pr_var / sum(pr_var))
```

```
##    Comp.1    Comp.2    Comp.3
## 0.4567308 0.3379671 0.2053021
```

```
cumsum(pve/sum(pve))
```

```
##    Comp.1    Comp.2    Comp.3
## 0.4567308 0.7946979 1.0000000
```

```
# Scree-Plot
data_frame(comp_id=1:length(pve), y=cumsum(pve/sum(pve))) %>%
  ggplot(aes(x=comp_id , y=y)) + geom_point() + geom_line() +
  coord_cartesian(ylim=c(0,1)) +
  labs(x="Número de Componentes Principais",
       y="Percentual de Variância Explicada Acumulada") + theme_bw() +
  scale_x_continuous(breaks = 1:3) +
  scale_y_continuous(breaks = seq(0,1,.1), labels = scales::percent) +
  geom_hline(yintercept = .7, linetype = 2, col = "#A11D21")
```




```
# Cálculo da proporção acumulada da variância explicada pelas componentes
# variância explicada por 2 componentes
cve <- cumsum(pve)
(cve_pc2 <- cve[2])
```

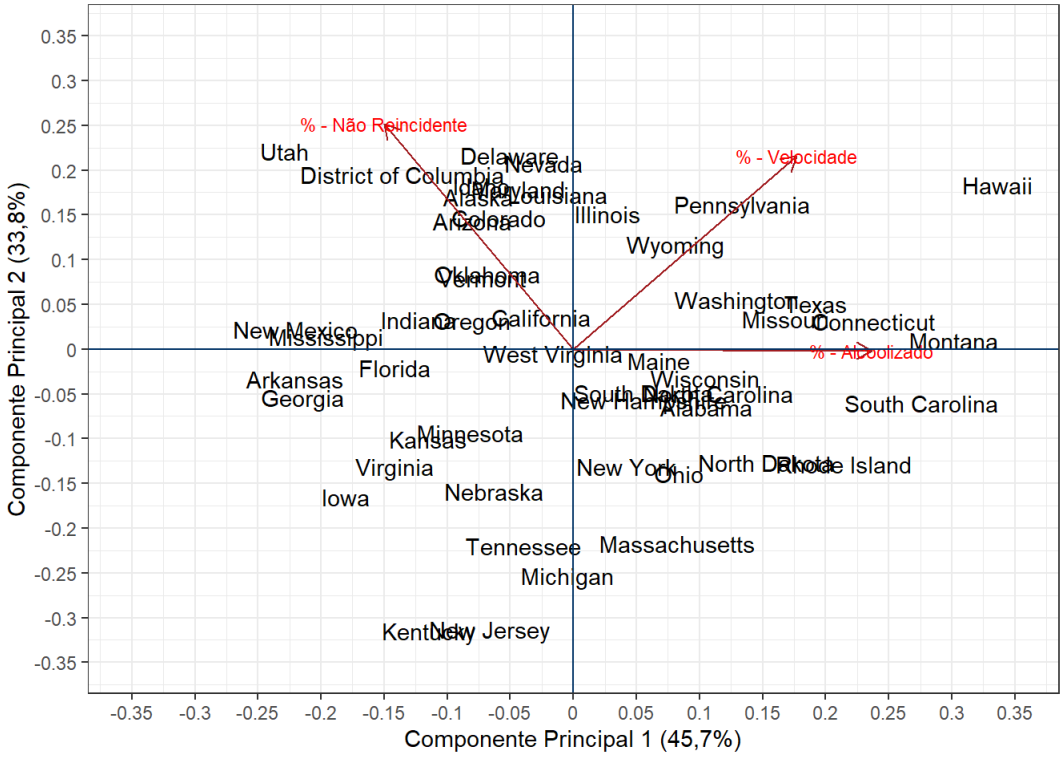
```
##    Comp.2
## 0.7946979
```

7. Visualizando as duas primeiras componentes principais

As duas primeiras componentes principais permitem a visualização dos dados em duas dimensões, capturando uma alta proporção da variação (79%) das três características: velocidade, influência do álcool e acidentes pela primeira vez. Isso permite tentar discernir padrões nos dados com o objetivo de encontrar grupos de estados semelhantes no país. Embora os algoritmos de agrupamento estejam se tornando cada vez mais eficientes, o reconhecimento humano de padrões é um método facilmente acessível e muito eficiente de avaliar padrões nos dados.

Criou-se um gráfico de dispersão das componentes principais e será explorado como os estados se agrupam nessa visualização.

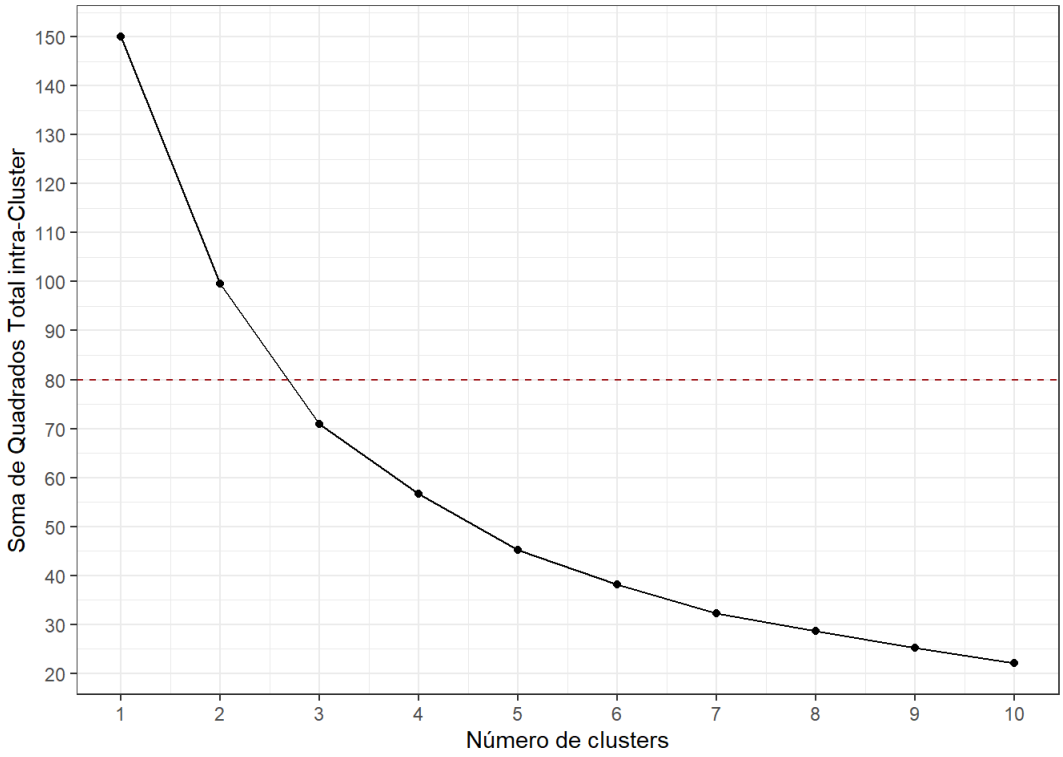
```
# Duas primeiras componentes
pcomp1 <- pca_fit$scores[,1]
pcomp2 <- pca_fit$scores[,2]

# Plotando as duas primeiras componentes com autoplot
require(ggfortify)
row.names(car_new) <- car_new$Estado
autoplot(pca_fit, data = car_new, loadings = TRUE,
         loadings.colour = "#A11D21", loadings.label = TRUE,
         loadings.label.size = 3, label = TRUE, shape = FALSE) +
  labs(x="Componente Principal 1 (45,7%)",
       y="Componente Principal 2 (33,8%)") +
  theme_bw() +
  coord_cartesian(ylim = c(-.35,.35), xlim = c(-.35,.35)) +
  scale_x_continuous(breaks = seq(-.35,.35,.05),
                     labels = paste(round(seq(-.35,.35,.05),2))) +
  scale_y_continuous(breaks = seq(-.35,.35,.05),
                     labels = paste(round(seq(-.35,.35,.05),2))) +
  geom_hline(yintercept = 0, col = "#003366") +
  geom_vline(xintercept = 0, col = "#003366")
```



8. Encontrar *clusters* de estados semelhantes pelos dados

Não ficou totalmente claro no gráfico de dispersão das componentes principais em quantos grupos os estados se agrupam. Para ajudar na identificação de um número razoável de *clusters*, pode-se usar o algoritmo *KMeans*, criando um *scree-plot* e localizando o "cotovelo" (*elbow*), que é uma indicação de quando a adição de mais clusters não adiciona muito poder explicativo.

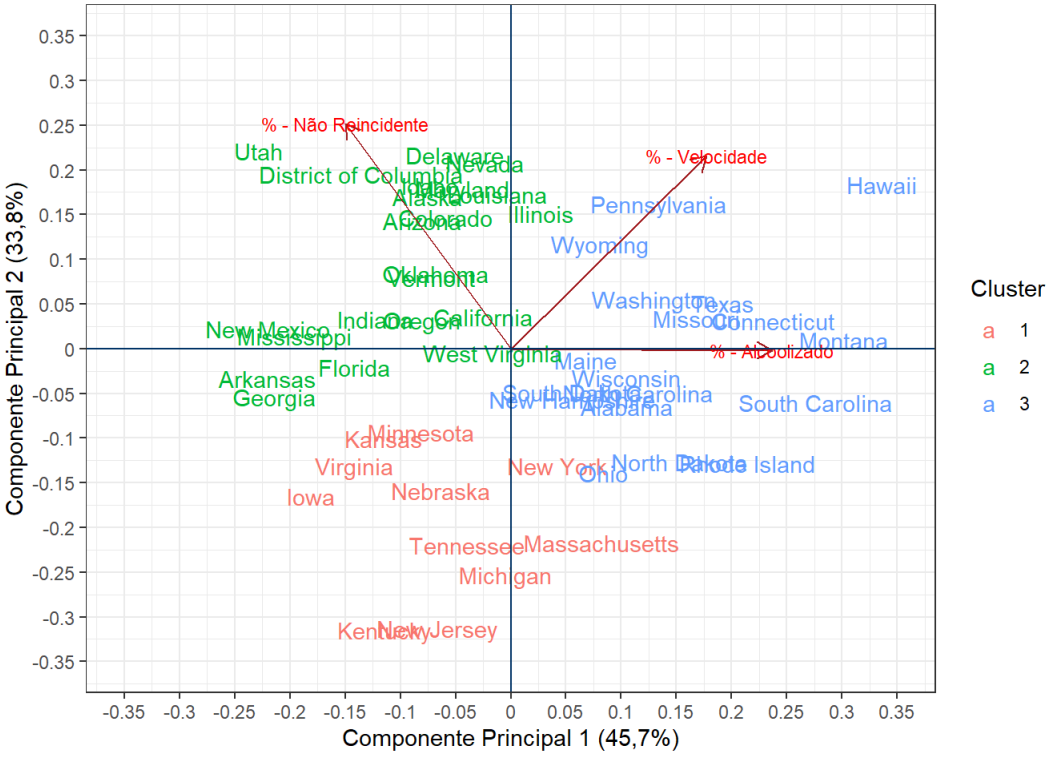
```
# Vetor de 1 a 10
k_vec <- 1:10

# Vetor de inércias
inertias <- rep(NA, length(k_vec))

# Lista para kmeans
mykm <- list()

# Kmeans é aleatório, para reproducibilidade fixa-se uma semente
set.seed(1)
for (k in k_vec) {
  # Salva o cluster em mykm
  mykm[[k]] <- kmeans(car_acc_standised[,c(3,4,5)], centers = k, nstart=50)
  # Armazena a soma de quadrados
  inertias[k] <- mykm[[k]]$tot.withinss
}

# Scree-plot
data_frame(k_vec,inertias) %>%
  ggplot(aes(k_vec, inertias) ) +
  geom_point() + geom_line() +
  geom_hline(yintercept = 80, col = "#A11D21", linetype = 2) +
  labs(x="Número de clusters",
       y="Soma de Quadrados Total intra-Cluster") +
  theme_bw() +
  scale_x_continuous(breaks = 1:10) +
  scale_y_continuous(breaks = seq(0,150,10))
```



9. *KMeans* para visualizar clusters no gráfico de dispersão de componentes principais

Como não houve um cotovelo claro no *scree-plot*, atribuir os estados em dois ou três grupos é uma escolha razoável, e a análise será feita usando três grupos. Verifica-se como fica o gráfico de dispersão de componentes principais se colorir os estados de acordo com o cluster ao qual eles estão designados.

```
# Obtenha cluster-ids de kmeans fit com k=3
cluster_id <- as.factor(mykm[[3]]$cluster)
car_new$cluster <- cluster_id

# Colorir de acordo com o cluster o gráfico de componentes principais
autoplot(pca_fit, data = car_new, loadings = TRUE,
         loadings.colour = "#A11D21", loadings.label = TRUE,
         loadings.label.size = 3, label = TRUE, shape = FALSE,
         colour = 'cluster') +
 labs(x="Componente Principal 1 (45,7%)",
      y="Componente Principal 2 (33,8%)",
      color = "Cluster") +
 theme_bw() +
 coord_cartesian(ylim = c(-.35,.35), xlim = c(-.35,.35)) +
 scale_x_continuous(breaks = seq(-.35,.35,.05),
                    labels = paste(round(seq(-.35,.35,.05),2))) +
 scale_y_continuous(breaks = seq(-.35,.35,.05),
                    labels = paste(round(seq(-.35,.35,.05),2))) +
 geom_hline(yintercept = 0, col = "#003366") +
 geom_vline(xintercept = 0, col = "#003366")
```



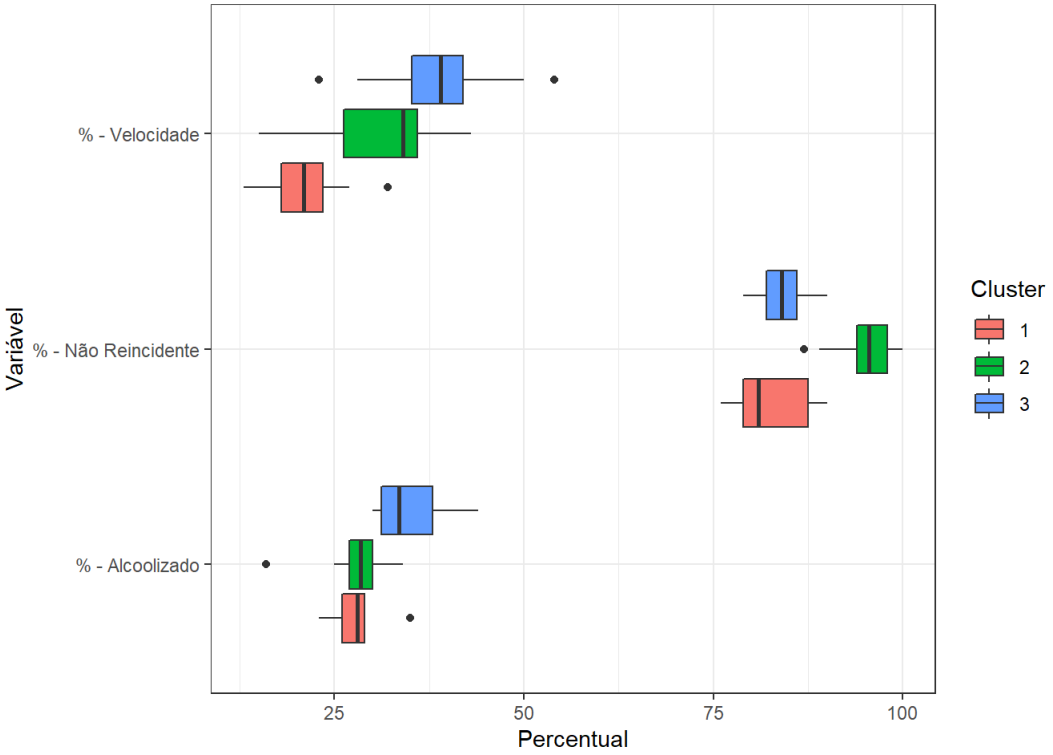
10. Visualiza as diferenças das características entre os *clusters*

Até o momento, usou-se interpretação visual dos dados e o algoritmo de clusterização *KMeans* para revelar padrões nos dados, mas o que esses padrões significam?

Lembre-se de que as informações usadas para agrupar os estados em três grupos distintos são a porcentagem de motoristas em alta velocidade, sob influência de álcool e que não foram envolvidos anteriormente em um acidente. Usa-se esses *clusters* para visualizar como os estados se agrupam ao considerar as duas primeiras componentes principais. Isso é bom para entender a estrutura dos dados, mas nem sempre é fácil de entender, especialmente se as conclusões devem ser comunicadas a um público não especialista.

Um próximo passo razoável em nossa análise é explorar como os três clusteres são diferentes em termos das três características que usou-se para o armazenamento do *cluster*. Em vez de usar os recursos iguamente dimensionados, volta-se a usar os recursos não dimensionados para ajudar a interpretar as diferenças.

```
# Transformar o banco em formato longo
car_new %>%
  select(-`Taxa Fatal`) %>%
  gather(key=feature, value=percent, -Estado, -cluster) %>%
  ggplot(aes(x=feature,y=percent, fill=cluster)) +
  geom_boxplot() +
  coord_flip() +
  labs(y = "Percentual", x = "Variável", fill = "Cluster") +
  theme_bw()
```

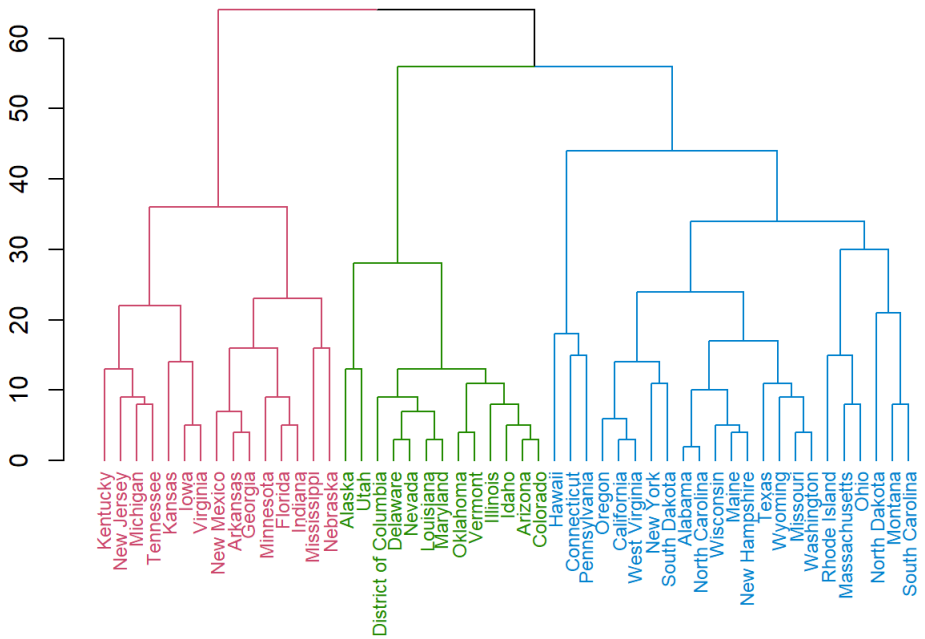


11. Método de Clusterização Hierárquica

Com o intuito de comparação, será realizado um método de clusterização hierárquico. Em seguida, os resultados serão comparados com a clusterização pelo método *KMeans*.

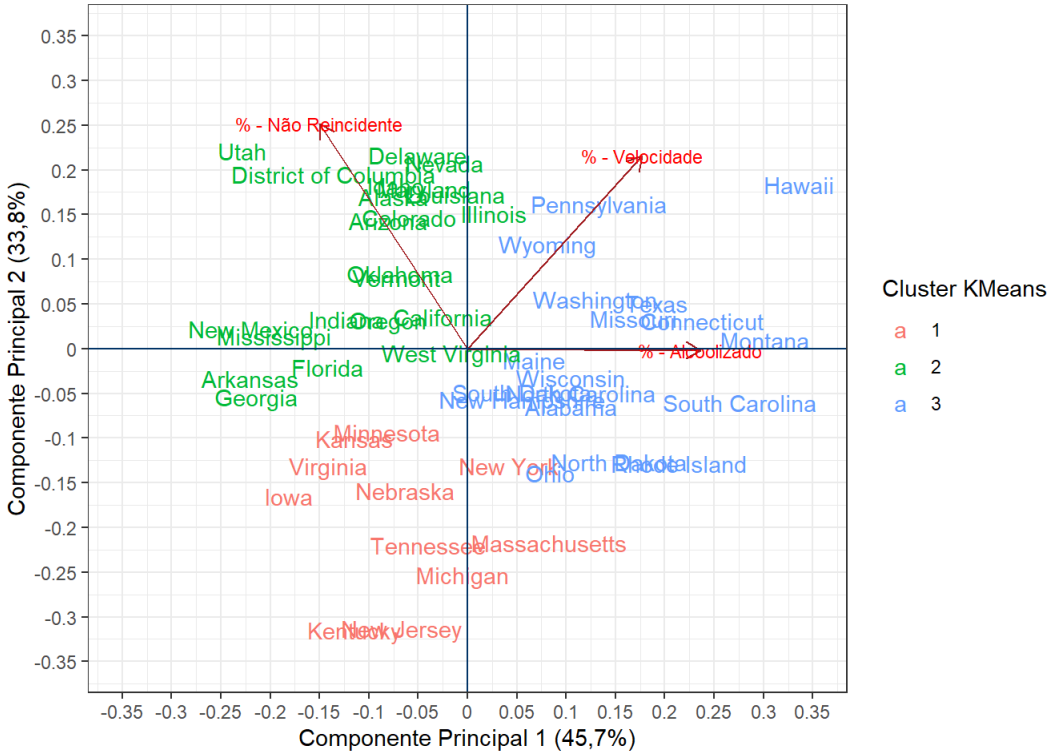
```
per_cols <- car_new %>% select(starts_with("%")) %>%
  dist(method = "manhattan")

hclust_obj <- hclust(per_cols, method = "complete")

require(dendextend)
require(colorspace)
```

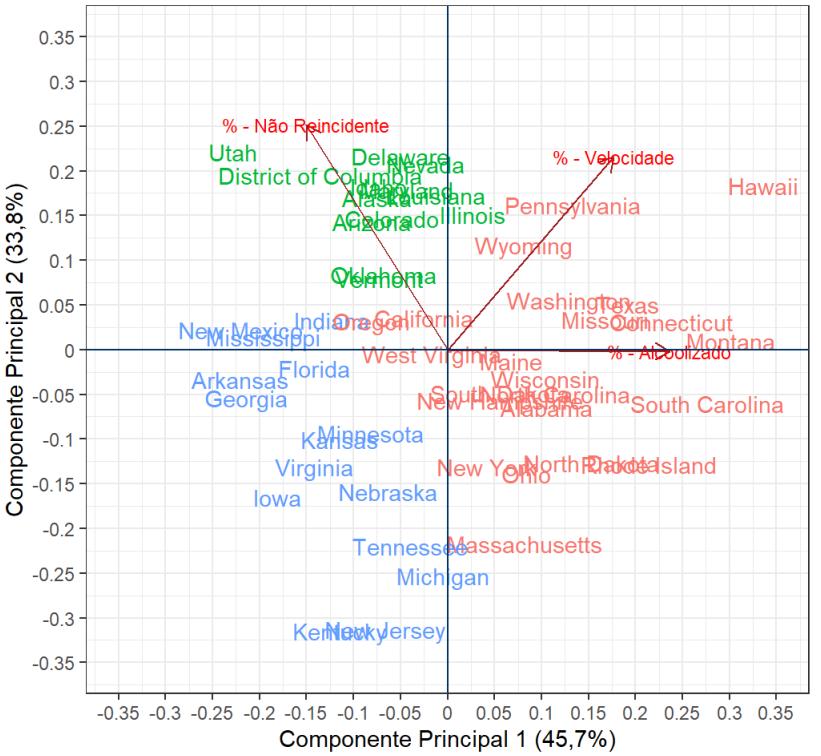
```
dend_obj <- as.dendrogram(hclust_obj)
dend_col <- color_branches(dend_obj, k = 3)
dend_obj %>% color_branches(k = 3) %>% color_labels(k = 3) %>%
  set("labels_cex",.75) %>% plot()
```



11.1. Comparação entre *KMeans* e *Cluster* Hierárquico

```
car_new$hcluster <- factor(cutree(hclust_obj, k = 3))
g1 <- autoplot(pca_fit, data = car_new, loadings = TRUE,
               loadings.colour = "#A11D21", loadings.label = TRUE,
               loadings.label.size = 3, label = TRUE, shape = FALSE,
               colour = 'cluster') +
  labs(x="Componente Principal 1 (45,7%)",
       y="Componente Principal 2 (33,8%)",
       color = "Cluster KMeans") +
  theme_bw() +
  coord_cartesian(ylim = c(-.35,.35), xlim = c(-.35,.35)) +
  scale_x_continuous(breaks = seq(-.35,.35,.05),
                     labels = paste(round(seq(-.35,.35,.05),2))) +
  scale_y_continuous(breaks = seq(-.35,.35,.05),
                     labels = paste(round(seq(-.35,.35,.05),2))) +
  geom_hline(yintercept = 0, col = "#003366") +
  geom_vline(xintercept = 0, col = "#003366")
g2 <- autoplot(pca_fit, data = car_new, loadings = TRUE,
               loadings.colour = "#A11D21", loadings.label = TRUE,
               loadings.label.size = 3, label = TRUE, shape = FALSE,
               colour = 'hcluster') +
  labs(x="Componente Principal 1 (45,7%)",
       y="Componente Principal 2 (33,8%)",
       color = "Cluster Hierárquico") +
  theme_bw() +
  coord_cartesian(ylim = c(-.35,.35), xlim = c(-.35,.35)) +
  scale_x_continuous(breaks = seq(-.35,.35,.05),
                     labels = paste(round(seq(-.35,.35,.05),2))) +
  scale_y_continuous(breaks = seq(-.35,.35,.05),
                     labels = paste(round(seq(-.35,.35,.05),2))) +
  geom_hline(yintercept = 0, col = "#003366") +
  geom_vline(xintercept = 0, col = "#003366")
g1; g2
```



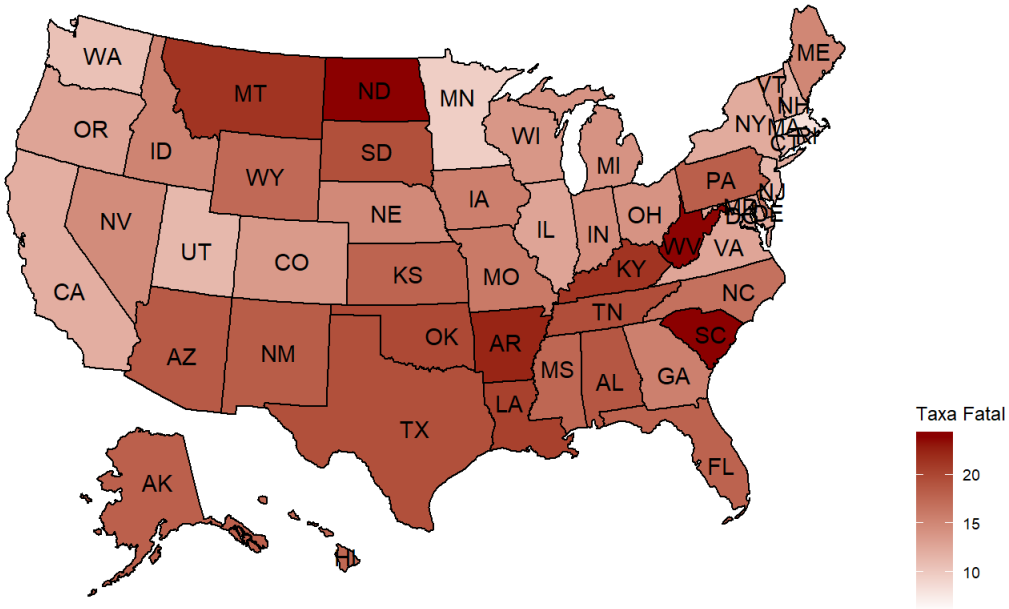


12. Mapas

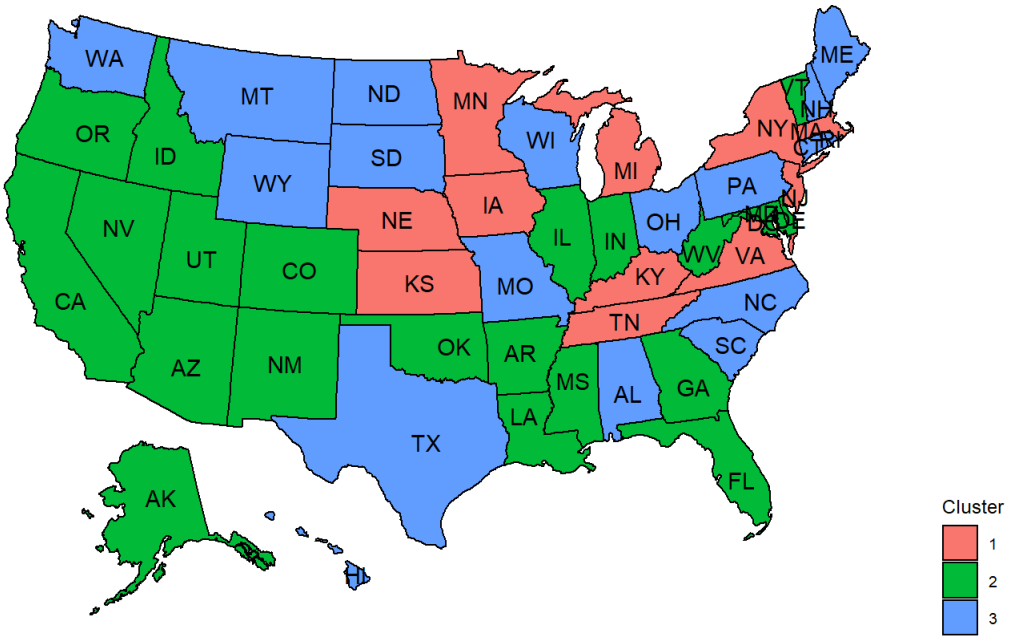
Foram elaborados 2 mapas:

- O primeiro é um mapa de calor das taxas fatais por estado;
- Já o segundo é um mapa em que cada cor representa um *cluster*.

```
require(usmap)
colnames(car_new)[1] <- 'state'
## Gráfico 1 - Taxa Fatal
plot_usmap(data = car_new, values = "Taxa Fatal", color = 'black',
           labels = TRUE) +
  scale_fill_continuous(low = 'white', high = 'darkred',
                        name = "Taxa Fatal",
                        label = scales::comma) +
  theme(legend.position = "right")
```




```
## Gráfico 2 - Taxa Fatal
plot_usmap(data = car_new, values = "cluster", color = 'black', labels = TRUE) +
  scale_fill_discrete(name = "Cluster") +
  theme(legend.position = "right")
```



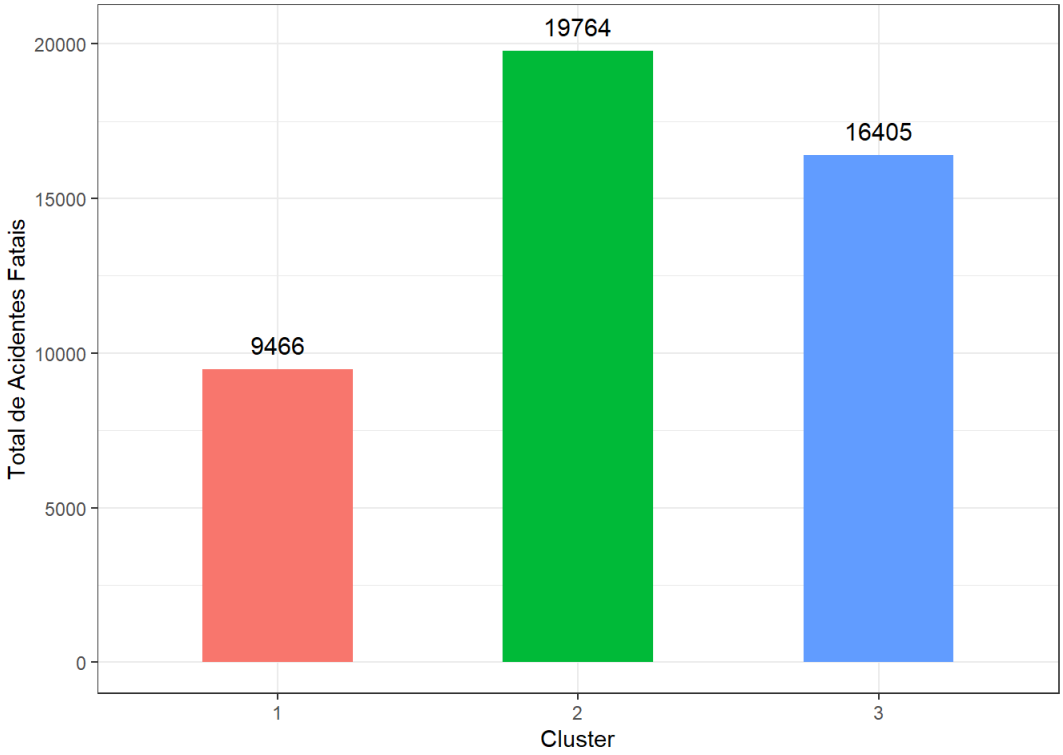
13. Calcula o número de acidentes em cada *cluster*

Agora está claro que diferentes grupos de estados podem exigir intervenções diferentes. Como os recursos e o tempo são limitados, é útil começar com uma intervenção em um dos três grupos primeiro. Que grupo seria esse? Para determinar isso, serão incluídos dados sobre quantas milhas são percorridas em cada estado, pois isso ajudará a calcular o número total de acidentes fatais em cada estado. Os dados sobre milhas percorridas estão disponíveis em outro arquivo de texto delimitado por tabulação. Atribui essas novas informações a uma coluna no quadro de dados e cria-se um *boxplot* para quantos acidentes de trânsito fatais totais existem em cada cluster de estado.

```
# Lendo o arquivo miles-driven.csv
miles_driven <- read_delim( file="miles-driven.txt", delim = '|')

# Juntar miles_driven com car_acc and add num_drvr_fatl_col
car_acc_joined <- car_acc  %>%
  left_join(miles_driven, by="state") %>%
  mutate(num_drvr_fatl_col= drvr_fatl_col_bmiles*million_miles_annually/1000)

# Agrupa o dataframe e resume os dados
car_acc_joined$cluster <- cluster_id
car_acc_joined_summ <- car_acc_joined %>%
  group_by(cluster) %>%
  select(cluster,num_drvr_fatl_col) %>%
  summarise("Quantidade de Estados"=n(),
            "Média de Acidentes" = mean(num_drvr_fatl_col),
            Soma=sum(num_drvr_fatl_col))
car_acc_joined_summ %>% kable() %>% kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

cluster	Quantidade de Estados	Média de Acidentes	Soma
1	11	860.5059	9465.565
2	22	898.3786	19764.329
3	18	911.4064	16405.316

```
# Compara o total de acidentes usando barplot
car_acc_joined_summ %>%
  ggplot(aes(x=cluster, y=Soma)) +
  geom_bar(aes(fill = cluster), stat = 'identity',
           show.legend = F, width = .5) +
  geom_text(aes(y = round(Soma) + 500, label = paste0(round(Soma))),
            vjust=0, size = 4) +
  theme_bw() + labs(x = "Cluster", y = "Total de Acidentes Fatais")
```



14. Escolha do *cluster* para aplicação da política pública inicial

Pela tabela resumo mostrada anteriormente, é possível verificar pela média que não há uma diferença marcante no número total de acidentes por cluster, porém os clusters estão agrupados de forma que os problemas relacionados aos percentuais de motoristas correndo acima da velocidade, alcoolizados e não reincidentes é próximo, logo poderia escolher o cluster com o maior número de estados e o que teve a maior quantidade total de acidentes, ou seja, os estados a serem tratados seriam os do cluster 2.