Conteúdo

1. VISÃO GERAL E ENTENDENDO O DATASET

O objetivo deste trabalho é aplicar conhecimento de redução de dimensionalidade juntamente com o desafio de trabalhar um dataset que representando uma imagem, desta forma, dados ‘0’ representa ausência de cor (pixel) e ‘1’ representa contém pixel formando a imagem. Enunciado sobre o dataset:

Temos 200 exemplares de cada um dos dígitos, de 0 a 9, totalizando 2 mil exemplares.

Cada exemplar está contido em um arquivo, cujo nome tem o seguinte padrão: classe_id.BMP.inv.pgm.

Ex: 0_001.BMP.iv.pgm, representa o exemplar 001 do dígito 0.

Cada arquivo contém, após a terceira linha, 4096 dimensões do exemplar.

2. DESENVOLVIMENTO

2.1 ENTENDIMENTO DO DATASET

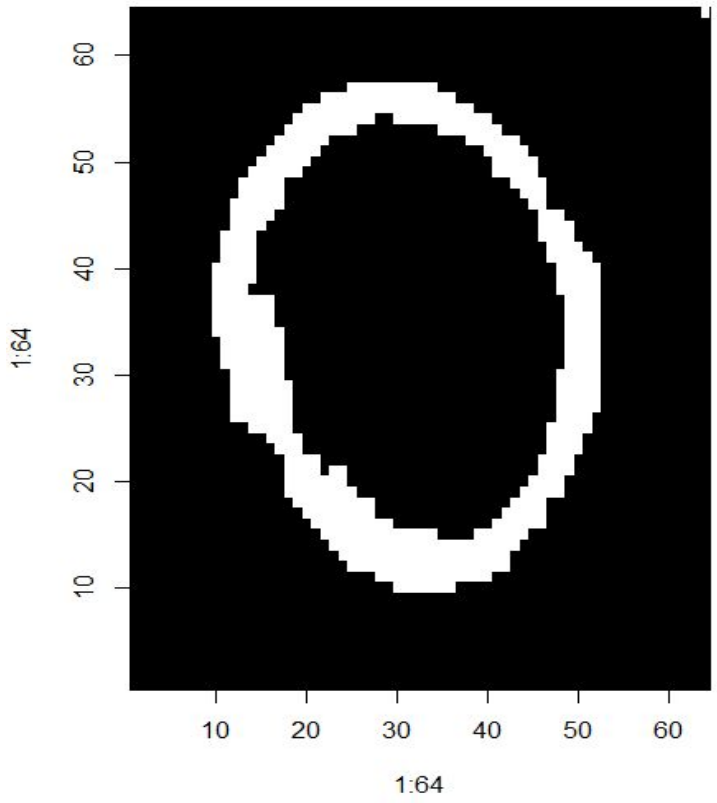
Inicialmente a primeira tarefa foi ler um arquivo e transformar em uma matriz/dataset 64x64 e usando a função image do R, conseguimos visualizar do que se tratava os exemplares, eram números escritos à mão por diversas pessoas, ‘0’ e ‘1’ representam as linhas que representam os números. Portanto tínhamos em arquivos separados números de 0 - 9 escritos a mão por diversas pessoas em diferentes formatos.

Código

```
# Leitura de um arquivo da quarta linha em diante.
file <- read.csv("./DigitosCompleto/0_001.BMP.inv.pgm", header =
FALSE, skip = 3, sep = " ")

# Removendo ultima coluna que contém apenas dados vazios
file <- file[-18]

# Exploração do arquivo
dim(file)
View(head(file, n=5))
str(file)

# Transformar em uma matrix 64x64
v<-as.vector(t(file))
NROW(v) # Temos 4097 linhas
NCOL(v) # Temos 1 coluna
v<-v[-4097]
v<-as.numeric(v)
v <- t(v)
mt<-matrix(v, byrow =F, 64,64)

# Apresentação do conteúdo do arquivo
image(1:64, 1:64, mt, col=gray((0:255)/255))
```

Resultados:

Imagem gerada pelo primeiro exemplar da classe zero.



2.2 MONTANDO O DATASET

O objetivo seria montar um dataset contendo todos os 2 mil exemplares, dividimos em duas tarefas principais

- Gerar um dataframe a partir dos arquivos lidos
- Cada arquivo será uma linha do dataframe

Também foi implementado uma classe de números na última coluna do data frame para poder obter mais informações ao data frame auxiliando em futuras análises. Essa classe foi retirada através do primeiro número do nome do arquivo que pelo padrão estabelecido representa qual imagem de número se trata o arquivo.

Nota: ao desenvolver a função como especificado, notamos um arquivo que contém mais dados que os demais portanto inicialmente o data frame gerado por nossa função continha mais colunas do que a especificação (4096 colunas e 2 mil linhas). Realizamos uma verificação onde é adicionado ao data frame apenas arquivos com dimensão esperada de 4097 colunas onde 4096 são os dígitos

binário, e a última coluna a classe (0 à 9), assim evitamos possíveis outliers com dimensão diferente em todo dataset.

Código

```
# Montar um data frame com o conteúdo de todos os arquivos

# Caminho raiz do CSV
files <- list.files(path = "./DigitosCompleto")

# Vetor com nome de todos os arquivos
vect_files <- as.vector(t(files))

# Ambiente de Teste - 100 arquivos para leitura : (descomentar)
# vect_files <- head(vect_files,100)
df <- data.frame()

for (x in vect_files) {

  filepath <- file.path(paste("./DigitosCompleto/", x ,sep=""))
  file_read <- read.csv(filepath, header = FALSE,
                        skip = 3, sep = " ")

  file_read <- file_read[-18]
  v<-as.vector(t(file_read))
  v<-v[-4097]
  v<-as.numeric(v)
  v <- t(v)

  file_name <- unlist(strsplit(x, "\\."))
  file_name <- unlist(strsplit(file_name[1], "\\_"))
  v <- cbind(v, number = file_name[1])
  v <- as.numeric(v)

  if(length(v) == 4097){
     df <- rbind(df, v)
  }

}

# Atribuindo nome para classe de dígitos (0 a 9)
colnames(df)[4097] <- "number"
```

```
View(head(file, n=5))
dim(df)
```

Também criamos um ambiente de testes que economiza o custo computacional para que pudéssemos verificar se o dataset gerado era o esperado e especificado.

```
# vect_files <- head(vect_files,100)
```

Assim apenas era retirado o comentário da linha de código acima controlando o número de linhas desejado para montar o data frame, no exemplo, amostra de 100 arquivos . Portando a fim de diminuir processamento durante o desenvolvimento a interação foi feita de acordo número de arquivos presentes em um vetor.

Como data frame final conseguimos completar todas iterações, obtendo 1999 linhas e 4097.

2.3 APLICANDO PCA

Para iniciarmos nossa atividade de redução de dimensionalidade foi feito uma análise de variância de cada uma das colunas do data frame, filtrando colunas com baixa ou nenhuma variância em relação às demais. A escolha de até qual valor considerar baixa foi olhando os resultados da média geral de variância e o primeiro quartil (25%), consideremos até a casa milesimal (0.0009) uma baixa variância.

Min	1st Quarter	Median	Mean	3rd Quarter	Max.
0.000000	0.004484	0.030069	0.063912	0.119066	8.248118

Código

```
# Análise de variância de cada uma das colunas (var).
# Alguma coluna apresenta variância muito menor do que outras? Se
sim, quantas e quais?
variance <- sapply(df, var)
summary(variance)

low_variance <- variance[unlist(variance >= 0.000000 & variance <=
0.0009)]
no_variance <- variance[unlist(variance == 0.000000)]

length(low_variance)
length(no_variance)

# Verificando as features que tem nenhuma variância, ou seja,
colunas com valores apenas em 0 ou apenas 1
no_variance <- which(apply(df, 2, var) >= 0 & apply(df, 2, var) <=
0.0009)
View(no_variance)

classe <- df[4097] # guardando a classe de números
df <- df[-4097] # removendo a classe de números para não ser
considerada como variância

# Novo dataframe sem colunas que não apresentam variância
new_df <- df[ - as.numeric(which(apply(df, 2, var) >= 0 & apply(df,
2, var) <= 0.0009))]
dim(new_df)

# Aplicando PCA para verificar a sugestão de redução de dimensão de
forma estatística
pca <- prcomp(new_df, center = TRUE, scale. = TRUE) #1999
Componentes principais

# PCA com 2 principais componentes
pca_two <- prcomp(new_df, rank. = 2 )

options(max.print=999999)
summary(pca) # A partir do 332 componente, a taxa de riqueza
acumulada dos dados se mantém estável em 90%
```

Realize a análise de variância de cada uma das colunas (var). Alguma coluna apresenta variância muito menor do que outras? Se sim, quantas e quais?

Aplique o PCA e comente sobre o método. Quantas são as dimensões resultantes? Qual a variabilidade das primeiras dimensões? Comente.

Fora gerada 1999 dimensões.

```
Standard deviation     1.31552 1.31059 1.30478 1.30237 1.29932 1.29487 1.29154 1.28847 1.28567 1.28190
Proportion of Variance 0.00046 0.00046 0.00046 0.00045 0.00045 0.00045 0.00045 0.00044 0.00044 0.00044
Cumulative Proportion  0.89095 0.89141 0.89187 0.89232 0.89277 0.89322 0.89367 0.89411 0.89455 0.89499
                         PC321   PC322   PC323   PC324   PC325   PC326   PC327   PC328   PC329   PC330
Standard deviation     1.27936 1.27437 1.27113 1.26837 1.26644 1.26430 1.26122 1.25607 1.25101 1.24684
Proportion of Variance 0.00044 0.00043 0.00043 0.00043 0.00043 0.00043 0.00043 0.00042 0.00042 0.00042
Cumulative Proportion  0.89543 0.89587 0.89630 0.89673 0.89716 0.89759 0.89801 0.89843 0.89885 0.89927
                         PC331   PC332   PC333   PC334  PC335  PC336  PC337  PC338  PC339   PC340
Standard deviation     1.24227 1.23713 1.23361 1.23170 1.2297 1.2252 1.2245 1.2185 1.2169 1.21391
Proportion of Variance 0.00041 0.00041 0.00041 0.00041 0.0004 0.0004 0.0004 0.0004 0.0004 0.00039
Cumulative Proportion  0.89968 0.90009 0.90050 0.90090 0.9013 0.9017 0.9021 0.9025 0.9029 0.90330
```

A partir do componente 328, a explicação e taxa de riqueza enquanto ao dataset original se manteve em 90%.

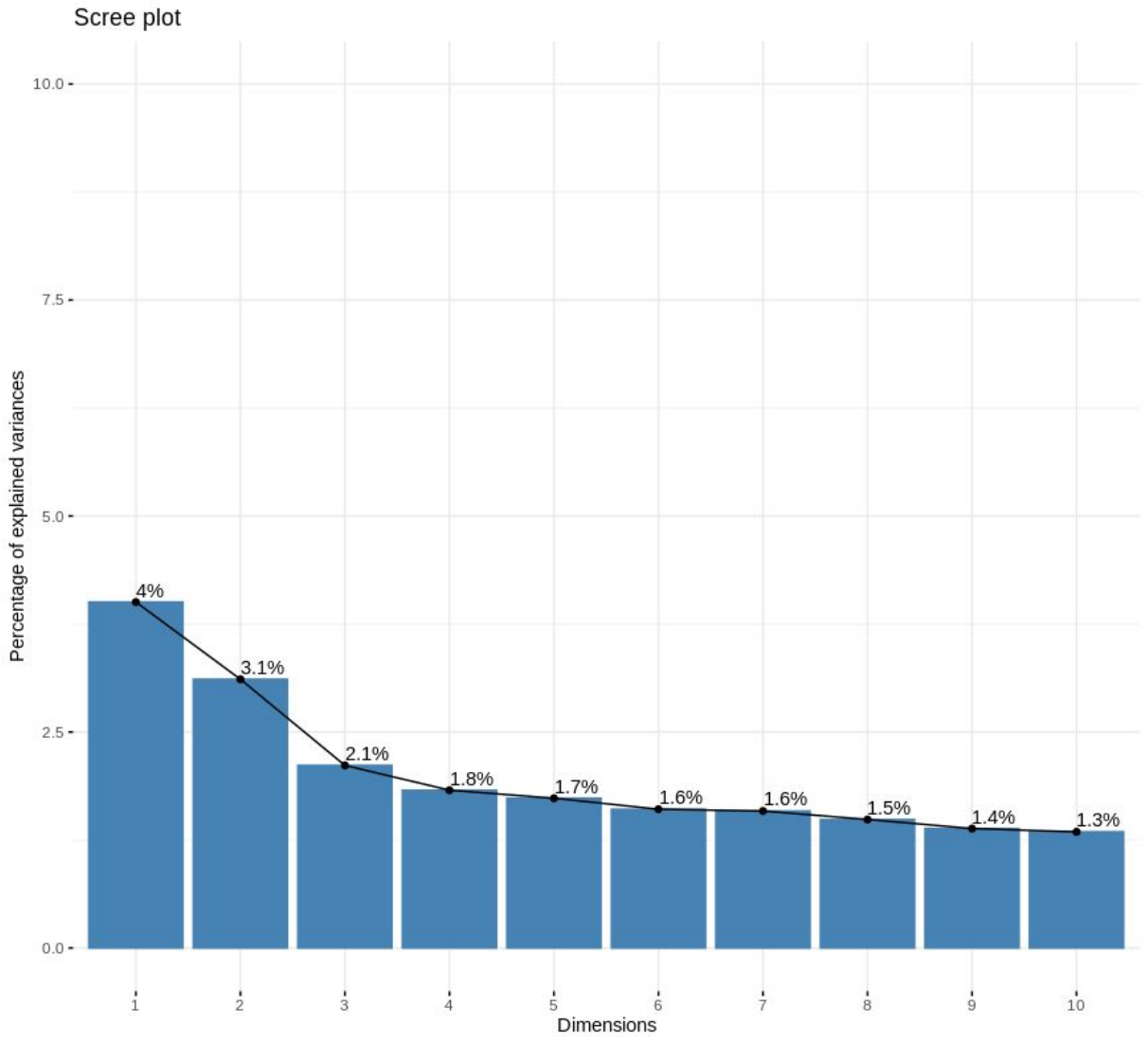
```
Importance of components:
                          PC1      PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
Standard deviation     12.23447 10.78310 8.88782 8.26263 8.04864 7.74738 7.69837 7.45549 7.18823 7.08933
Proportion of Variance  0.04004  0.03111 0.02113 0.01826 0.01733 0.01606 0.01585 0.01487 0.01382 0.01345
Cumulative Proportion   0.04004  0.07115 0.09228 0.11055 0.12788 0.14393 0.15979 0.17466 0.18848 0.20193
                         PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18    PC19    PC20
Standard deviation     6.90670 6.74007 6.71440 6.54303 6.40254 6.29897 6.22713 6.07827 6.02559 5.87201
Proportion of Variance 0.01276 0.01215 0.01206 0.01145 0.01097 0.01061 0.01037 0.00988 0.00971 0.00922
Cumulative Proportion  0.21469 0.22684 0.23890 0.25036 0.26132 0.27194 0.28231 0.29219 0.30191 0.31113
```

Os primeiros componentes são os que possuem uma maior variância em relação aos demais, a partir do PC4 essa diferença passa se estabilizar e assim se subsegue até o último componente decaindo gradativamente, com uma variância muito similar. Com o nosso conhecimento prévio obtido na questão 1 (explorando um arquivo isoladamente), supomos que os componentes principais são onde os pixels mais se encontram no plano entre os números 0 - 9, portanto o 1 (max da normalização) similar “escrito” dos números estão contidos nos componentes principais. A mesma ideia vale para os componentes que mantém a proporção de variância, onde são os ‘zeros’ comuns nas posições dos números de 0 - 9. Desta forma é muito provável que as bordas da imagem apresentem tx de variância menores do que o centro da imagem, na qual apresentam maiores chances de conter variações entre 1 e 0.

A variabilidade das primeiras dimensões são 4% e 3,1%, explicando desta forma 7% de todo dataset de forma cumulativa, sendo as duas dimensões com

maiores proporções de variância utilizamos elas para análise em gráfico de plots de duas dimensões.

Às proporção de explicação vai decaindo ao longo das demais componentes principais, principalmente após PC4, onde há uma estabilização, como podemos observar no gráfico abaixo.



Código

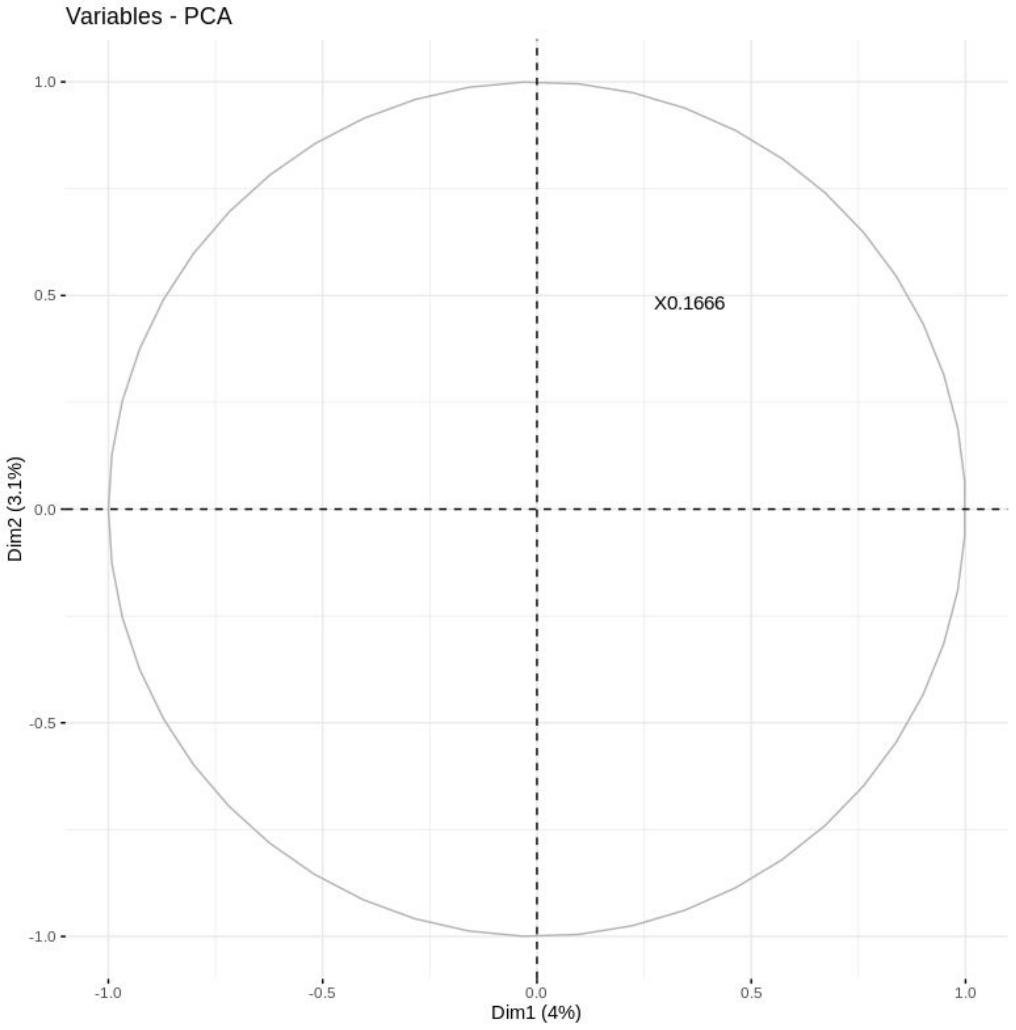
```
# plot porcentagem de explicação de variâncias por dimensões
fviz_screeplot(pca, addlabels = TRUE, ylim = c(0, 10))
```

Foi gerado gráfico 2D entre os 2 componentes principais por distribuição de indivíduos agrupados por classe dos números.

É possível notar uma massa de concentração de símbolos que representam indivíduos pertencente a uma classe de número do dataset. Essa massa sobrepõe os símbolos, indicando que muitas colunas do data set tem sua contribuição de variância em regiões próximas, assim em geral, os números foram escritos em localidade semelhantes, no caso, centralizadas. Os símbolos mais dispersos representam números que alguma característica de design que foge das regiões comuns onde a maior parte dos números foi escrita.

Plotando também a variável com maior contribuição para dataset considerando as 2 dimensões com maior proporção de variância, é possível saber em qual coluna do data frame se refere a variável. O resultado foi a variável 1666, considerando o novo dataset 3738 colunas pois foi retirado colunas sem ou baixa variância, a metade é 1869 ou seja o centro desse dataframe, assim a coluna 1666 está próximo ao centro ao lado da metade esquerda, nessa região ocorre a maior

variância considerando todos os exemplares.



Código

```
# PCA Indivíduos por classe de números
fviz_pca_ind(pca, geom="point", pointsize= 1.5, habillage =
classe$number, alpha.ind = 1)

# PCA variável com maior contribuição
fviz_pca_var(pca, geom="text", select.var = list(contrib = 1))

# Direção da variável com maior contribuição por distribuição de
indivíduos
fviz_pca_biplot(pca, select.var = list(contrib = 1),
label="none", pointsize= 1,habillage = classe$number, alpha.ind
= 1,
                col.var = "black", # Variables color
)
```

3. CONCLUSÃO

Os avanços tecnológicos no armazenamento de dados trouxe a possibilidade de uma quantidade massiva de armazenamento de dados chamado de big data, isso é armazenamento de dados como imagem, vídeo, áudio entre outros variados tipos, no entanto, muitas das vezes nem todos os dados coletados são realmentes úteis para tirar informações e conhecimento. Desta forma foi desenvolvido técnicas para reduzir a dimensionalidade de dataset deixando com informações mais relevantes para o problema, trazendo benefícios como redução de complexidade para modelos de machine learning.

A primeira redução de dimensionalidade utilizada foi muito semelhante a técnica de “Missing Values Ratio” onde inicialmente o dataset de dígitos contia a última coluna valores totalmente perdidos, desta forma removemos essa coluna, no caso 18º coluna de cada arquivo, por não existir nenhum dado, não havendo um limite de eliminação, mais sim a remoção completa da coluna em questão.

A segunda técnica utilizada foi “Low Variance Filter” onde removemos as colunas que não possuem nenhuma ou baixa proporção de variância, pois não agregam muita informação a ser trabalhada.

Então após a eliminação de colunas sem variâncias foi possível aplicar a técnica “Principal Component Analysis (PCA)” onde foi permitiu transformar colunas originais do dataset em um novo conjunto de coordenadas. Essa técnica nos proporcionou visualizar a variação em cada dimensão, no dataset em questão a maiores informações foi obtidas no centro da imagem onde os conjuntos de números mais se assemelham, no centro também é a região onde está a direção de maior espalhamento, diferente das bordas da imagem que aparecem estável.

Essas técnicas nos permitiu obter maiores informações em dataset com maiores dimensões, o que seria quase impossível a olhar humano, assim essas ferramentas poderiam ajudar em um melhor trabalho junto com técnicas de machine learning permitindo ainda mais retiradas de informações e insights.

4. CÓDIGO

```
library("tidyr")
library(dplyr)
library(ggplot2)
library(dslabs)
library(reshape2)
library(stringr)
library(class)
library(rpart)
library(rpart.plot)
library('caret')
library(plyr)
# install.packages('caret')
library(factoextra)
# install.packages("factoextra")


# ---------------------- Primeiro passo -----------------------------
# Leitura de um arquivo da quarta linha em diante.
file <- read.csv("./DigitosCompleto/0_001.BMP.inv.pgm", header = FALSE,
skip = 3, sep = " ")

# Removendo última coluna que contém apenas dados vazios
file <- file[-18]

# Exploração do arquivo
dim(file)
View(head(file, n=5))
str(file)

# Transformar em uma matriz 64x64
v<-as.vector(t(file))
NROW(v) # Temos 4097 linhas
NCOL(v) # Temos 1 coluna
v<-v[-4097]
v<-as.numeric(v)
v <- t(v)
mt<-matrix(v, byrow =F, 64,64)

# Apresentação do conteúdo do arquivo
image(1:64, 1:64, mt, col=gray((0:255)/255))
```

```r
#------------------------- Segundo Passo -----------------------------
# Montar um data frame com o conteúdo de todos os arquivos

# Caminho raiz do CSV
files <- list.files(path = "./DigitosCompleto")

# Vetor com nome de todos os arquivos
vect_files <- as.vector(t(files))

# Ambiente de Teste - 34 arquivos para leitura : (descomentar)
# vect_files <- head(vect_files,100)
df <- data.frame()

for (x in vect_files) {
 filepath <- file.path(paste("./DigitosCompleto/", x ,sep=""))
 file_read <- read.csv(filepath, header = FALSE, skip = 3, sep = " ")
 file_read <- file_read[-18]
 v<-as.vector(t(file_read))
 v<-v[-4097]
 v<-as.numeric(v)
 v <- t(v)
  file_name <- unlist(strsplit(x, "\\."))
 file_name <- unlist(strsplit(file_name[1], "\\_"))
 v <- cbind(v, number = file_name[1])
 v <- as.numeric(v)
  if(length(v) == 4097){
   df <- rbind(df, v)
 }
}

# Atribuindo nome para classe de dígitos (0 a 9)
colnames(df)[4097] <- "number"

View(head(file, n=5))
str(df[4097]) # classe de números
dim(df)




#--------------------- Terceiro passo -----------------------------

# Análise de variância de cada uma das colunas (var).
```

```r
# Alguma coluna apresenta variância muito menor do que outras? Se sim, quantas e quais?
variance <- sapply(df, var)
summary(variance)

low_variance <- variance[unlist(variance >= 0.000000 & variance <= 0.0009)]
no_variance <- variance[unlist(variance == 0.000000)]

length(low_variance)
length(no_variance)

# Verificando as features que tem nenhuma variância, ou seja, colunas com valores apenas em 0 ou apenas 1
no_variance <- which(apply(df, 2, var) >= 0 & apply(df, 2, var) <= 0.0009)
View(no_variance)

classe <- df[4097] # guardando a classe de números
df <- df[-4097] # removendo a classe de números para não ser considerada como variância

# Novo data frame sem colunas que não apresentam variância
new_df <- df[ - as.numeric(which(apply(df, 2, var) >= 0 & apply(df, 2, var) <= 0.0009))]
dim(new_df)

# Aplicando PCA para verificar a sugestão de redução de dimensão de forma estatística
pca <- prcomp(new_df, center = TRUE, scale. = TRUE) #1999 Componentes principais

# PCA com 2 principais componentes
pca_two <- prcomp(new_df, rank. = 2 )

options(max.print=999999)
summary(pca) # A partir do 332 componente, a taxa de riqueza acumulada dos dados se mantém estável em 90%

# plot porcentagem de explicação de variâncias por dimensões
fviz_screeplot(pca, addlabels = TRUE, ylim = c(0, 10))

# Resultados da análise de componentes principais para variáveis
```

```
var <- get_pca_var(pca)
var

# PCA Indivíduos por classe de números
fviz_pca_ind(pca,    geom="point",    pointsize=    1.5,    habillage    =
classe$number, alpha.ind = 1)

# PCA variável com maior contribuição
fviz_pca_var(pca, geom="text", select.var = list(contrib = 1))

#  Direção  da  variável  com  maior  contribuição  por  distribuição  de
indivíduos
fviz_pca_biplot(pca,  select.var  =  list(contrib  =  1),  label="none",
pointsize= 1,habillage = classe$number, alpha.ind = 1,
                col.var = "black", # Variables color
)
```

---

5. REFERÊNCIAS

https://www.rdocumentation.org/packages/factoextra/versions/1.0.7
