Trabalho Final

Felipe da Rocha Ferreira

07/01/2023

Tarefa 1

Utilize o conjunto BreastCancer para realizar análises de reconhecimento de padrões. Para tanto suponha que a variável Class (última variável do conjunto) não é conhecida. Faça um relatório. Tal relatório deve ser estregue de maneira organizada discutindo os resultados encontrados.

O bando de dados *BreastCancer* possui 699 observações e 9 variáveis a serem analisadas:

- *Cl.thickness*: Espessura do aglomerado
- *Cell.size*: Uniformidade do tamanho da célula
- *Cell.shape*: Uniformidade da forma da célula
- *Marg.adesão*: Adesão Marginal
- *Epith.c.size*: Tamanho de célula epitelial única
- *Bare.nuclei*: Núcleos nus
- *Bl.cromatina*: Cromatina sem graça
- *Normal.nucleoli*: Nucléolos Normais
- *Mitoses*: Mitoses

Com posse dos dados e sem nenhum conhecimento prévio sobre grupos ao qual gostaríamos de classificar os dados, ou seja, não possuímos informações de categorias já existentes para então classificarmos os dados nestas, optou-se pela realização de uma análise não supervisionada.

```
library(mlbench) #banco de dados
library(dplyr)
library(tidyr)
library(factoextra)
library(cluster)
data(BreastCancer)
dados=BreastCancer%>%dplyr::select(-c(Id,Class))%>%dplyr::mutate(
  dplyr::across(.cols=everything(),~as.numeric(.)));rm(BreastCancer)
summary(dados)
```

```
##   Cl.thickness      Cell.size        Cell.shape     Marg.adhesion
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 4.000   Median : 1.000   Median : 1.000   Median : 1.000
##  Mean   : 4.418   Mean   : 3.134   Mean   : 3.207   Mean   : 2.807
##  3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000   3rd Qu.: 4.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
```

```
##
##   Epith.c.size     Bare.nuclei      Bl.cromatin     Normal.nucleoli
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000
##  Median : 2.000   Median : 1.000   Median : 3.000   Median : 1.000
##  Mean   : 3.216   Mean   : 3.545   Mean   : 3.438   Mean   : 2.867
##  3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 4.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##                   NA's   :16
##     Mitoses
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.569
##  3rd Qu.:1.000
##  Max.   :9.000
##
```

Após uma breve análise exploratória e observando que todos os dados são numéricos e não foram observados *outliers*, escolheu-se o método **k-means** de clusterização não-supervisionado.

K-means

Como passo inicial ao processo de clusterização, observou-se a presença de 16 valores faltantes na variável *Bare.nuclei* e, após a remoção das linhas com esses dados, sobraram 683 observações a serem utilizadas no modelo.

```
dados=dados%>%tidyr::drop_na();nrow(dados)
```

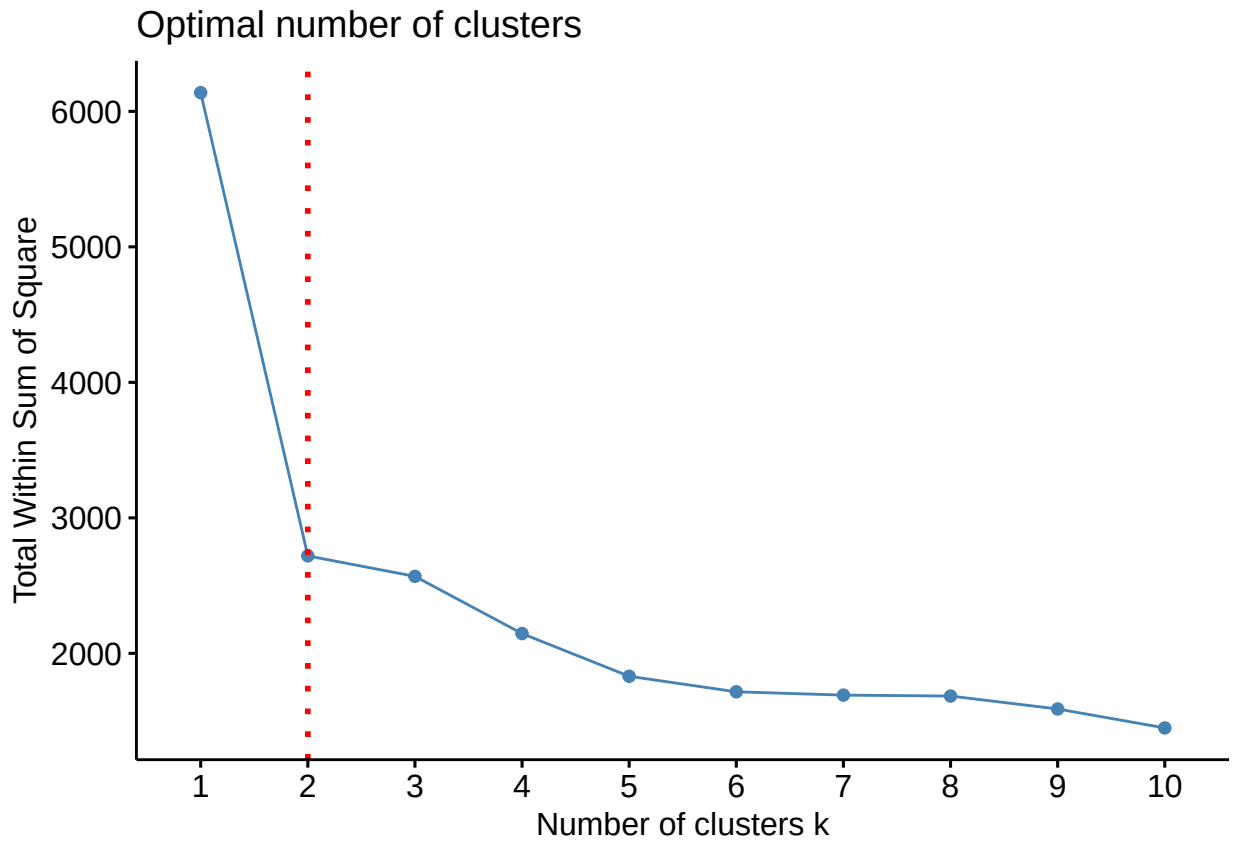
```
## [1] 683
```

Após a remoção, normalizamos os dados afim de garantir que todas as variáveis possuíssem média 0 e desvio-padrão 1.

```
dados <- scale(dados)%>%data.frame()
```

Utilizando o princípio do cotovelo, escolhemos dois como número ideal de clusters: por meio do gráfico a seguir, é possível observar que de 1 para 2, a diferença da soma dos quadrados é significativa, mas de 2 para 3 não, fazendo assim a curva e gerando o "cotovelo". Por tanto, pela parcimónia, optou-se por 2 clusters.

```
factoextra::fviz_nbclust(dados, kmeans, method = "wss")+
  geom_vline(xintercept = 2,linetype="dotted",color="red",size=1)
```



Por fim, criamos nosso modelo, que resultou em 2 clusters com 452 observações classificadas no *cluster 1* e 231 no *cluster 2*.

```
set.seed(573)#número da disciplina no PAVNET
km <- kmeans(dados, centers =2)
km$size
```

```
## [1] 452 231
```

```
paste0("(between_SS / total_SS =  ",round(km$betweenss/km$totss*100,1),"%)")
```

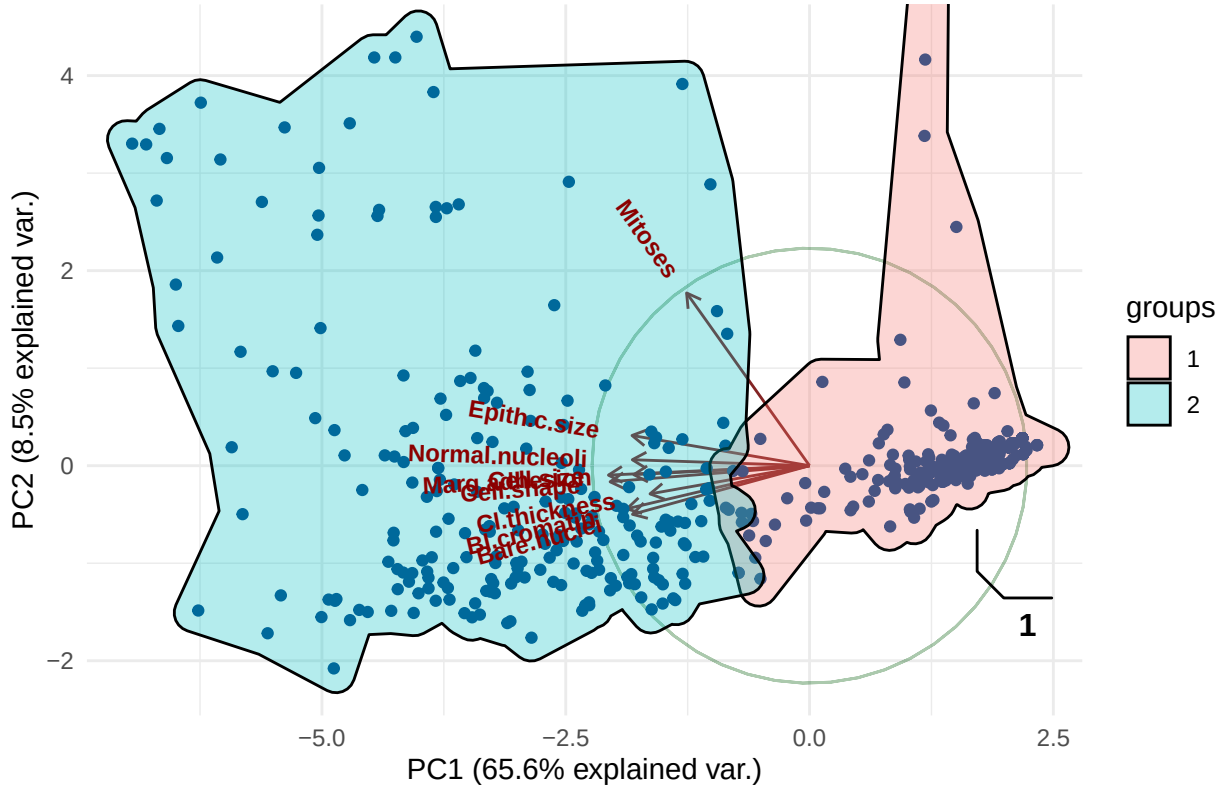
```
## [1] "(between_SS / total_SS =  55.7%)"
```

Componentes Principais (PCA)

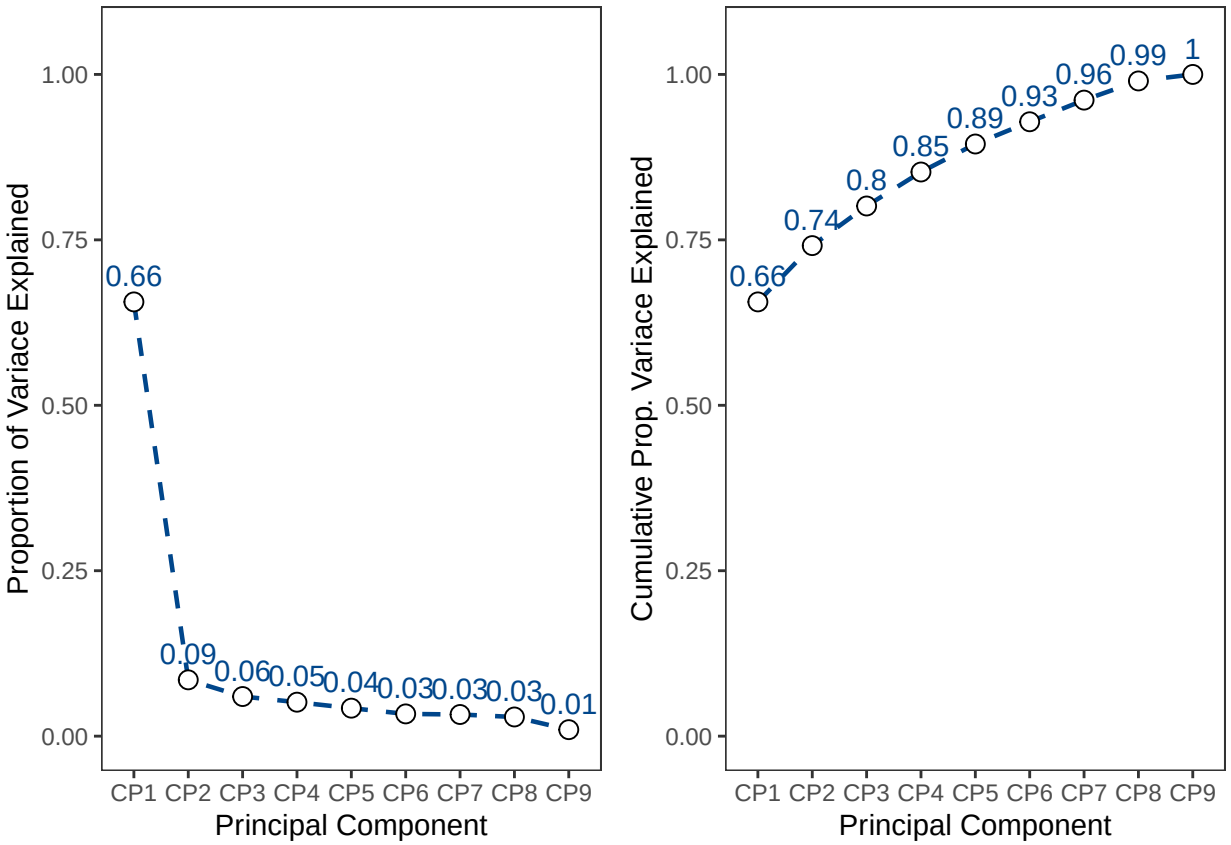
Utilizando o método dos **Componentes Principais** (PCA) para visualizar os dados, podemos observar que a variável *Mitoses* compõem uma componente (*PC2*, que explica 8,5%) e as demais compõem o *PC1* (a componente mais explicativa, 65,6%). E observa-se que o *PC1* é o que mais influência na classificação dos indivíduos: baixos valores das variáveis dessa componente resultam na classificação no *grupo 1* (vermelho), enquanto altos valores resultam na classificação em *grupo 2* (azul).

```
source("F:/Estudo/Estudo---UFV/UFV---GitHub/00.Functions/Function_PCA.R") #[1]
dados2=dados%>%dplyr::mutate(classificado=km[["cluster"]])
pca=prcomp(dados2%>%dplyr::select(-classificado), scale=TRUE)
```

```
ggbiplot(pcobj=pca,
         obs.scale = 1, var.scale = 1,
         groups = dados2$classificado%>%as.character(),ellipse = TRUE,
         circle = TRUE,labels = dados$Class ) +
  scale_color_discrete(name = '') +
  theme(legend.direction = 'horizontal', legend.position = 'top')+theme_minimal()
```




```
PCA_Prop_Var(pca)
```



Sobre a interpretação desses grupos e componentes, caberia a um especialista mais familiarizado com esses dados. Um exemplo hipotético de interpretação seria o *grupo 2* ser pessoas com alto risco de possuírem câncer de mama e o *grupo 1* pessoas com baixo risco (nesse exemplo hipotético, indivíduos com baixo valores no *componente 1*, cuja variáveis estão correlacionadas, resultariam em altas chances de possuir o câncer).

Function_PCA.R está disponível em https://github.com/felipe179971/felipe179971-UFV-Inteligencia_Artificial_e_Computacional/blob/main/00.Functions/Function_PCA.R

```
rm(dados,dados2,km,pca,pr.out)
```

Tarefa 2

Utilizando o mesmo conjunto de dados BreastCancer realize análises de classificação. Para tanto, a variável Class (última variável do conjunto) é nossa variável dependente. Utilize validação cruzada para avaliar os modelos ajustados. Faça um relatório. Tal relatório deve ser estregue de maneira organizada discutindo os resultados encontrados.

Utilizando o mesmo banco de dados anterior (683 observações), mas agora contando com a variável qualitativa *Class*, vemos que a amostra conta com 444 observações classificadas em *benign* e 239 em *malignant*.

```
library(mlbench) #banco de dados
library(MASS) #QDA Analise de discriminante quadratica e LDA Linear Discriminant Analysis
library(dplyr)
library(tidyr)
data(BreastCancer)
```

```
dados=BreastCancer;rownames(dados)=dados$id
dados=BreastCancer%>%dplyr::select(-c(Id))%>%dplyr::mutate(
  dplyr::across(.cols=colnames(.)[-which(colnames(.)=="Class")],~as.numeric(.)))%>%
  tidyr::drop_na();rm(BreastCancer);table(dados$Class)
```

```
##
##    benign malignant
##       444       239
```

LDA com validação Cruzada (leave one out)

Para realizar essa classificação, vamos utilizar o procedimento de validação cruzada, conhecida como *leave one out*. Esse procedimento é utilizado para comparar a capacidade preditiva do modelo - Por exemplo, se temos um conjunto de dados com 300 indivíduos, estimamos a função com 299 e predizemos a função para o próximo indivíduo, e assim fazemos de 1 em 1. Esse procedimento é utilizado para evitar a superestimação do poder de classificação do modelo, ou seja, para tentar fazer a predição de maneira mais independente possível.

```
r <- MASS::lda(formula = Class ~ ., #especi em funcao de todas as outras vars
data = dados, #do dados iris
prior = c(1,1)/2, #tenho 2 populacoes e defini probabilidade de
#ocorrencia de cada populacao iguais para todas (1/2)
CV=TRUE) #CV=TRUE: quero validacao cruzada (leave one out)
```

Construindo a tabela de confusão, onde o número de classificações corretas ficam na diagonal.

```
# Funcao Linear
classificacao <- r$class
cvl <- table(dados$Class,classificacao)
cvl
```

```
##            classificacao
##             benign malignant
##   benign       436         8
##   malignant     18       221
```

Com uma baixa taxa de erro aparente, o modelo se mostrou bom para classificar os dados. E, considerando o erro *Classificar como benigno quando na verdade é maligno* como o mais prejudicial ao modelo, vemos que esse tipo de erro foi o mais significativo, representando quase 70% de todos os erros, o que pode ser um ponto fraco do modelo.

```
#Taxa de Erro Aparente
TEA <- 1 - (sum(diag(cvl))/sum(cvl))
paste0(round(TEA*100,2),"%")
```

```
## [1] "3.81%"
```

```
rm(dados,r,classificacao,cvl,TEA)
```

Tarefa 3

Utilize o conjunto Boston Housing Dataset predizer o valor médio das casas ocupadas pelo proprietário (medv) por meio de técnicas e predição. Faça um relatório. Tal relatório deve ser estregue de maneira organizada discutindo os resultados encontrados.

Regressão Linear Múltipla com stepwise

Visando sempre a parcimônia do modelo, parece excessivo a utilização de dezoito variáveis para predizer uma. Por isso, utilizaremos o método de *stepwise*, que inclui as variáveis passo-a-passo e testa a permanência para determinar o modelo final.

Para a construção, o bando de dados foi dividido em dois, onde selecionou-se 80% dos dados para o treinamento e 20% para o teste. Além disso, a variável *tax* foi removida por possuir um $VIF = 9.271222 > 5$ (Teste de multicolinearidade). A princípio,*rad* também apresentou VIF alto, mas após a remoção de TAX o VIF passou de 7.753354 para 2.646155, que é menor que 5 e, portanto, permaneceu.

```
library(mlbench) #banco de dados
library(dplyr)
library(tidyr)
data("BostonHousing")
#Padronizando as variáveis
dados=BostonHousing%>%dplyr::mutate(
  dplyr::across(.cols=1:13,~scale(as.numeric(.))));rm(BostonHousing)
set.seed(573)#número da disciplina no PAVNET
sum(is.na(dados));summary(dados)
```

```
## [1] 0
```

```
##        crim.V1              zn.V1              indus.V1
##  Min.   :-0.419367   Min.   :-0.487240   Min.   :-1.5563017
##  1st Qu.:-0.410563   1st Qu.:-0.487240   1st Qu.:-0.8668328
##  Median :-0.390280   Median :-0.487240   Median :-0.2108898
##  Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.0000000
##  3rd Qu.: 0.007389   3rd Qu.: 0.048724   3rd Qu.: 1.0149946
##  Max.   : 9.924110   Max.   : 3.800473   Max.   : 2.4201701
##        chas.V1             nox.V1               rm.V1
##  Min.   :-0.272329   Min.   :-1.4644327   Min.   :-3.876413
##  1st Qu.:-0.272329   1st Qu.:-0.9121262   1st Qu.:-0.568068
##  Median :-0.272329   Median :-0.1440749   Median :-0.108358
##  Mean   : 0.000000   Mean   : 0.0000000   Mean   : 0.000000
##  3rd Qu.:-0.272329   3rd Qu.: 0.5980871   3rd Qu.: 0.482291
##  Max.   : 3.664771   Max.   : 2.7296452   Max.   : 3.551530
##         age.V1              dis.V1              rad.V1
##  Min.   :-2.3331282   Min.   :-1.265817   Min.   :-0.9818712
##  1st Qu.:-0.8366200   1st Qu.:-0.804891   1st Qu.:-0.6373311
##  Median : 0.3170678   Median :-0.279047   Median :-0.5224844
##  Mean   : 0.0000000   Mean   : 0.000000   Mean   : 0.0000000
##  3rd Qu.: 0.9059016   3rd Qu.: 0.661716   3rd Qu.: 1.6596029
##  Max.   : 1.1163897   Max.   : 3.956602   Max.   : 1.6596029
##         tax.V1            ptratio.V1             b.V1
##  Min.   :-1.3126910   Min.   :-2.7047025   Min.   :-3.903331
##  1st Qu.:-0.7668172   1st Qu.:-0.4875567   1st Qu.: 0.204869
```

```
##  Median :-0.4642132   Median : 0.2745872   Median : 0.380810
##  Mean   : 0.0000000   Mean   : 0.0000000   Mean   : 0.000000
##  3rd Qu.: 1.5294129   3rd Qu.: 0.8057784   3rd Qu.: 0.433222
##  Max.   : 1.7964164   Max.   : 1.6372081   Max.   : 0.440616
##       lstat.V1            medv
##  Min.   :-1.529613   Min.   : 5.00
##  1st Qu.:-0.798630   1st Qu.:17.02
##  Median :-0.181074   Median :21.20
##  Mean   : 0.000000   Mean   :22.53
##  3rd Qu.: 0.602423   3rd Qu.:25.00
##  Max.   : 3.545262   Max.   :50.00
```

```
index <- sample(nrow(dados), nrow(dados) * 0.80)
train <- dados[index, ]
test <- dados[-index, ]
#Definindo o modelo com todas as variáveis
all <- lm(medv ~ ., data=train)
#Verificando multicolinearidade
olsrr::ols_coll_diag(all)[["vif_t"]]
```

```
##    Variables Tolerance      VIF
## 1       crim 0.5809130 1.721428
## 2         zn 0.4104615 2.436282
## 3      indus 0.2633900 3.796651
## 4       chas 0.9281202 1.077447
## 5        nox 0.2375463 4.209706
## 6         rm 0.4881938 2.048367
## 7        age 0.3116691 3.208531
## 8        dis 0.2360509 4.236374
## 9        rad 0.1289764 7.753354
## 10       tax 0.1078606 9.271222
## 11   ptratio 0.5348115 1.869818
## 12         b 0.7648419 1.307460
## 13     lstat 0.3239421 3.086972
```

```
#Removendo tax
train=train%>%dplyr::select(-c(tax))
test=test%>%dplyr::select(-c(tax))
all <- lm(medv ~ ., data=train)
olsrr::ols_coll_diag(all)[["vif_t"]]
```

```
##    Variables Tolerance      VIF
## 1       crim 0.5810770 1.720942
## 2         zn 0.4372309 2.287121
## 3      indus 0.3155862 3.168706
## 4       chas 0.9413041 1.062356
## 5        nox 0.2390679 4.182912
## 6         rm 0.4907771 2.037585
## 7        age 0.3121164 3.203932
## 8        dis 0.2361160 4.235206
## 9        rad 0.3779068 2.646155
## 10   ptratio 0.5378890 1.859120
## 11         b 0.7662134 1.305119
## 12     lstat 0.3241307 3.085175
```

```r
#Modelo--------------------------------------
#Definindo o modelo só com o intercepto
intercept_only <- lm(medv ~ 1, data=train)
#Definindo o modelo com todas as variáveis
all <- lm(medv ~ ., data=train)
#stepwise
forward <- step(intercept_only, direction='forward', scope=formula(all), trace=0)
```

Sendo esse o modelo final

```r
forward
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm + ptratio + dis + nox + chas +
##     b + zn + crim + rad, data = train)
##
## Coefficients:
## (Intercept)        lstat           rm      ptratio          dis          nox
##     22.6144      -3.9302       2.5033      -2.1215      -3.2269      -2.3735
##        chas            b           zn         crim          rad
##      0.7927       0.8764       1.0483      -1.0104       1.3117
```

Avaliando o Modelo

Analisando a ANOVA, vemos que o modelo é significativo ($F = 100.638$ e $p - valor = 0$); possuí um $R^2 = 0.719$, indicando que $71,9\%$ da variabilidade de *medv* é explicada pelas variáveis que entraram no modelo;R^2 *ajustado* $= 0.712$ que é muito similar ao R^2, indicando não haver super ajuste do modelo (excesso de variáveis) .

```r
olsrr::ols_regress(forward)
```

```
##                         Model Summary
## ----------------------------------------------------------------
## R                       0.848       RMSE                 4.873
## R-Squared               0.719       Coef. Var           21.360
## Adj. R-Squared          0.712       MSE                 23.748
## Pred R-Squared          0.694       MAE                  3.377
## ----------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##
##                                 ANOVA
## --------------------------------------------------------------------------
##                  Sum of
##                 Squares         DF    Mean Square       F          Sig.
## --------------------------------------------------------------------------
## Regression    23899.915         10       2389.992    100.638    0.0000
## Residual       9333.109        393         23.748
## Total         33233.024        403
## --------------------------------------------------------------------------
```

```
##
##                                 Parameter Estimates
## ----------------------------------------------------------------------------------------
##       model      Beta    Std. Error    Std. Beta      t        Sig      lower     upper
## ----------------------------------------------------------------------------------------
## (Intercept)    22.614         0.243                93.028    0.000    22.136    23.092
##       lstat    -3.930         0.406       -0.429   -9.671    0.000    -4.729    -3.131
##          rm     2.503         0.343        0.271    7.306    0.000     1.830     3.177
##     ptratio    -2.121         0.321       -0.236   -6.616    0.000    -2.752    -1.491
##         dis    -3.227         0.456       -0.360   -7.079    0.000    -4.123    -2.331
##         nox    -2.373         0.444       -0.266   -5.343    0.000    -3.247    -1.500
##        chas     0.793         0.228        0.096    3.474    0.001     0.344     1.241
##           b     0.876         0.298        0.090    2.945    0.003     0.291     1.461
##          zn     1.048         0.363        0.116    2.891    0.004     0.335     1.761
##        crim    -1.010         0.299       -0.118   -3.380    0.001    -1.598    -0.423
##         rad     1.312         0.394        0.143    3.329    0.001     0.537     2.086
## ----------------------------------------------------------------------------------------
```

Predição

Agora, prevendo o Valor médio de casas ocupadas pelo proprietário em US$ 1.000 e comparando com o valor real, vemos que o modelo teve uma ótima taxa de acerto, sendo a média prevista de $21,8212$ enquanto o valor real se encontra ligeiramente abaixo, em $21,41667$.

```
previsao <- predict(forward, test)
mean(previsao)
```

```
## [1] 21.8212
```

```
mean(test$medv)
```

```
## [1] 21.41667
```

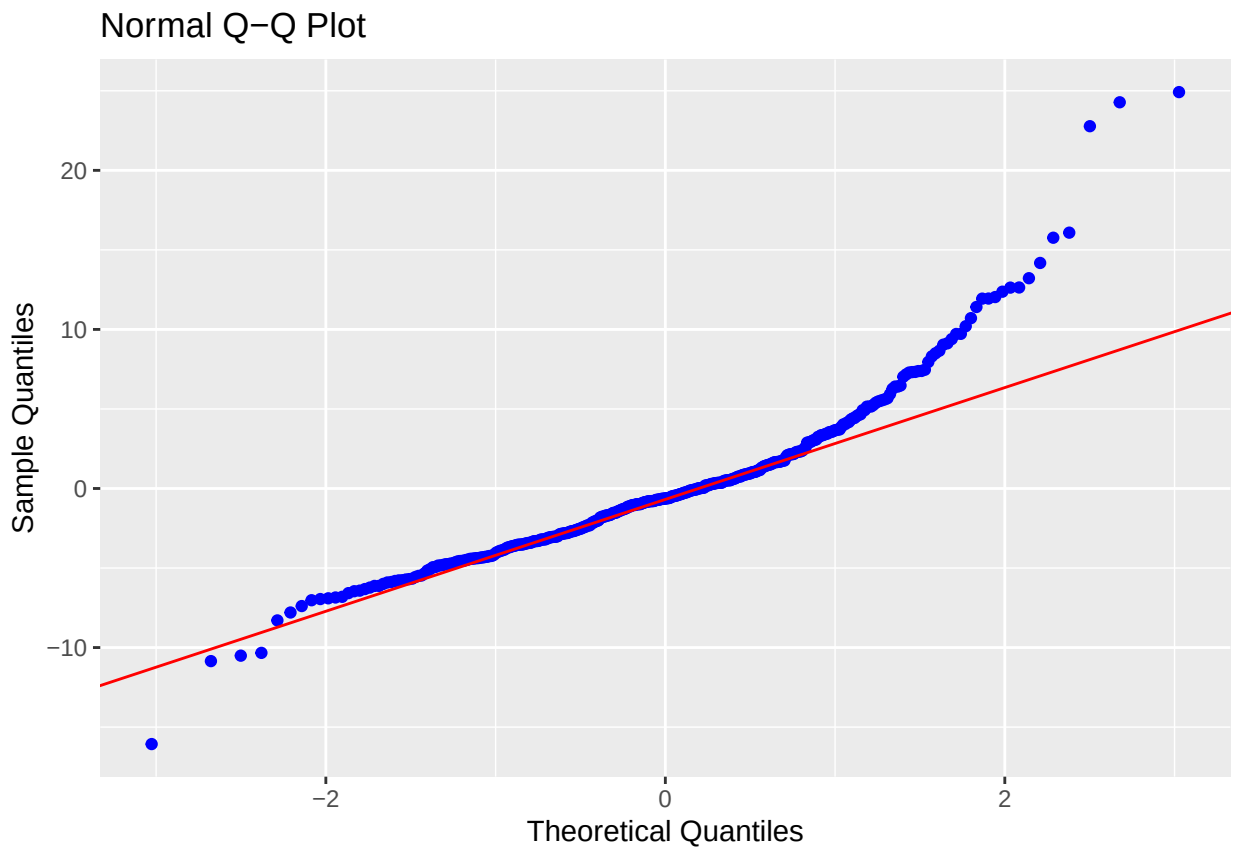
Observação

Entretanto, é importante salientar que o modelo não passou no teste de normalidade devido a sua calda pesada nem no teste de homodestaticidade. Mas passou no teste de independência e multicolinearidade. Ou seja, não cumpriu os pressupostos básicos do modelo de regressão linear múltiplo, tornando-o não confiável.

```
#Normalidade
olsrr::ols_test_normality(forward)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9061         0.0000
## Kolmogorov-Smirnov        0.1173         0.0000
## Cramer-von Mises         33.8779         0.0000
## Anderson-Darling          8.428          0.0000
## -----------------------------------------------
```

```
olsrr::ols_plot_resid_qq(forward)
```




```
#Homocedasticidade
olsrr::ols_test_breusch_pagan(forward)
```

```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##                Data
##  -------------------------------
##  Response : medv
##  Variables: fitted values of medv
##
##           Test Summary
##  -----------------------------
##  DF            =    1
##  Chi2          =    15.92754
##  Prob > Chi2   =    6.58141e-05
```

```
#Independência
car::durbinWatsonTest(forward)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.06850507      2.136673   0.132
##  Alternative hypothesis: rho != 0
```

```
#Multicolinearidade
olsrr::ols_coll_diag(forward)[["vif_t"]]
```

```
##    Variables Tolerance      VIF
## 1      lstat 0.3629521 2.755184
## 2         rm 0.5178344 1.931119
## 3    ptratio 0.5620190 1.779299
## 4        dis 0.2760825 3.622105
## 5        nox 0.2890237 3.459924
## 6       chas 0.9450217 1.058177
## 7          b 0.7694231 1.299675
## 8         zn 0.4416030 2.264477
## 9       crim 0.5827873 1.715892
## 10       rad 0.3858985 2.591355
```