TDE #3

Filipe J. Zabala

2024-11-06

NOMES COMPLETOS: _____

Instruções

1. Entrega até **2024-11-25**.
2. Tamanho do grupo: **até 3 pessoas**.
3. **Entregue a resolução via Moodle**, indicando o(s) **nome(s) completo(s) e a turma** no cabeçalho do documento.

Questões

Q1. **(3.0)** Foi observada uma amostra de clientes, que opiniaram com notas de 0 a 10 sobre um serviço prestado. Considere $X1$: pontualidade, $X2$: conhecimento e $X3$: disponibilidade.

a. (0.5) Defina n e p a partir das informações do `BLOCO 1A`.
b. (0.5) Indique o que são as medidas intituladas `Medida 1`, `Medida 2` e `Medida 3`, calculadas no `BLOCO 1B`.
c. (0.5) Apresente o passo-a-passo de como é calculado o valor `-0.1889822` da matriz `R` e interprete-o.
d. (0.5) No `BLOCO 1C` indique os testes que estão sendo realizados, suas hipóteses H_0 e H_1 e qual a sua decisão em cada um deles considerando $\alpha = 5\%$.
e. (0.5) No `BLOCO 1D` indique qual teste está sendo realizado, quais as hipóteses H_0 e H_1 e qual sua decisão considerando $\alpha = 5\%$.
f. (0.5) Quais diferenças você identifica entre os testes realizados nos blocos `1C` e `1D`?

```
# BLOCO 1A
X <- read.table('https://filipezabala.com/data/clientes.txt',
                header = TRUE, sep = '\t')
t(X)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## X1    6   10   10    7    7    4   10    5    8     7    10     6
## X2    8   10    9    9    8   10    8   10    9     9     8    10
## X3    3    5    5    4    3    2    5    4    4     3     6     4
```

```r
# BLOCO 1B
(m <- colMeans(X))  # Medida 1
```

```
##  X1  X2  X3
## 7.5 9.0 4.0
```

```r
(S <- cov(X))       # Medida 2
```

```
##            X1         X2         X3
## X1  4.4545455 -0.7272727  2.0000000
## X2 -0.7272727  0.7272727 -0.1818182
## X3  2.0000000 -0.1818182  1.2727273
```

```r
(R <- cor(X))       # Medida 3
```

```
##            X1         X2         X3
## X1  1.0000000 -0.4040610  0.8399639
## X2 -0.4040610  1.0000000 -0.1889822
## X3  0.8399639 -0.1889822  1.0000000
```

```r
# BLOCO 1C
t.test(X$X1, mu = 8)
```

```
##
##  One Sample t-test
##
## data:  X$X1
## t = -0.82065, df = 11, p-value = 0.4293
## alternative hypothesis: true mean is not equal to 8
## 95 percent confidence interval:
##  6.159002 8.840998
## sample estimates:
## mean of x
##       7.5
```

```r
t.test(X$X2, mu = 8)
```

```
##
##  One Sample t-test
##
## data:  X$X2
## t = 4.062, df = 11, p-value = 0.001877
## alternative hypothesis: true mean is not equal to 8
## 95 percent confidence interval:
##  8.458155 9.541845
## sample estimates:
## mean of x
##         9
```

```r
t.test(X$X3, mu = 8)
```

```
##
##  One Sample t-test
##
## data:  X$X3
## t = -12.282, df = 11, p-value = 9.156e-08
## alternative hypothesis: true mean is not equal to 8
## 95 percent confidence interval:
```

```
##  3.283206 4.716794
## sample estimates:
## mean of x
##         4
```

```r
# BLOCO 1D
ICSNP::HotellingsT2(X, mu = c(8,8,8))
```

```
##
##  Hotelling's one sample T2-test
##
## data:  X
## T.2 = 158.1, df1 = 3, df2 = 9, p-value = 4.213e-08
## alternative hypothesis: true location is not equal to c(8,8,8)
```

Q2. **(2.5)** (He, Zhang, and Zhang 2016) utilizaram a fluorescência de raios-X (*ED-XRF*) para determinar a composição química de cerâmicas antigas, detalhadas no banco de dados Composição Química de Amostras Cerâmicas[1].

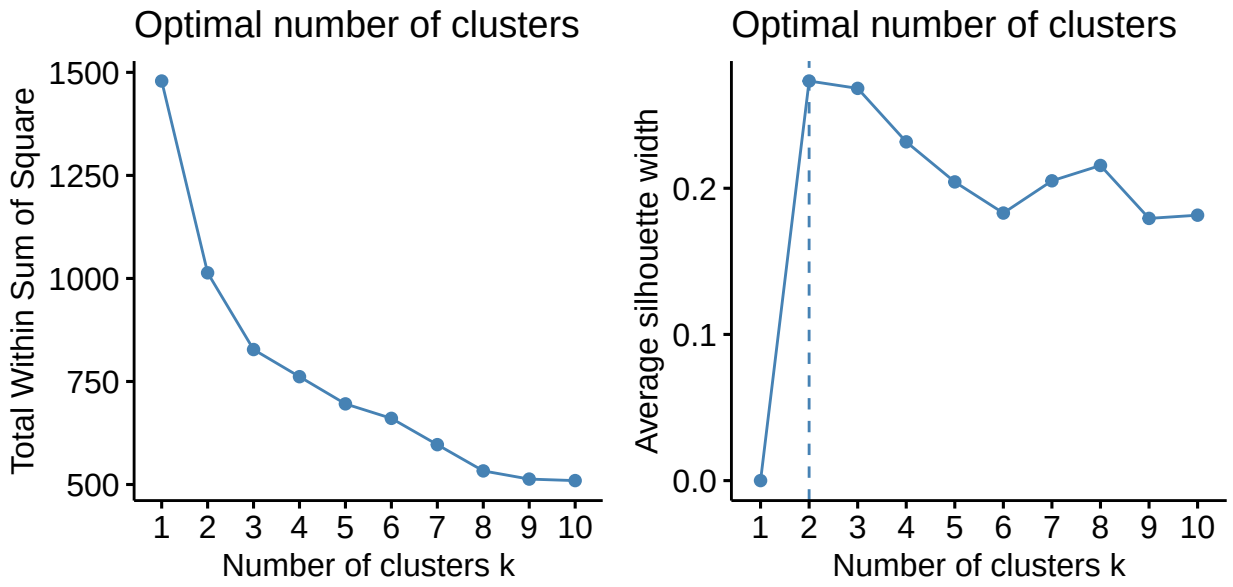
a. (0.5) Considerando os métodos 'wss' (*within sums of squares*) e 'silhouette' do `BLOCO 2B`, determine k, o número ótimo de clusters.
b. (0.5) Utilizando o resultado do item anterior, circule no `BLOCO 2C` com lápis, caneta ou virtualmente os k grupos determinados no item anterior.
c. (0.5) Explique o que ocorre nos comandos `prcomp(dat[,-(1:2)])` e `prcomp(dat[,-(1:2)], scale = TRUE)` do `BLOCO 2D`.
d. (0.5) Coloque em ordem decrescente as três variáveis com maior contribuição para a PC1, usando a saída do `BLOCO 2D`.
e. (0.5) Considerando o gráfico das duas primeira componentes principais do `BLOCO 2E`, você considera que estas duas dimensões são úteis para diferenciar as partes da cerâmica (*Body/Glaze* ou Corpo/Verniz)? Que pontos de corte você sugere em cada dimesão dos gráficos?

```
# BLOCO 2A
dat <- read.csv('https://filipezabala.com/data/ceramic.csv', head = T)
dplyr::glimpse(dat)
```

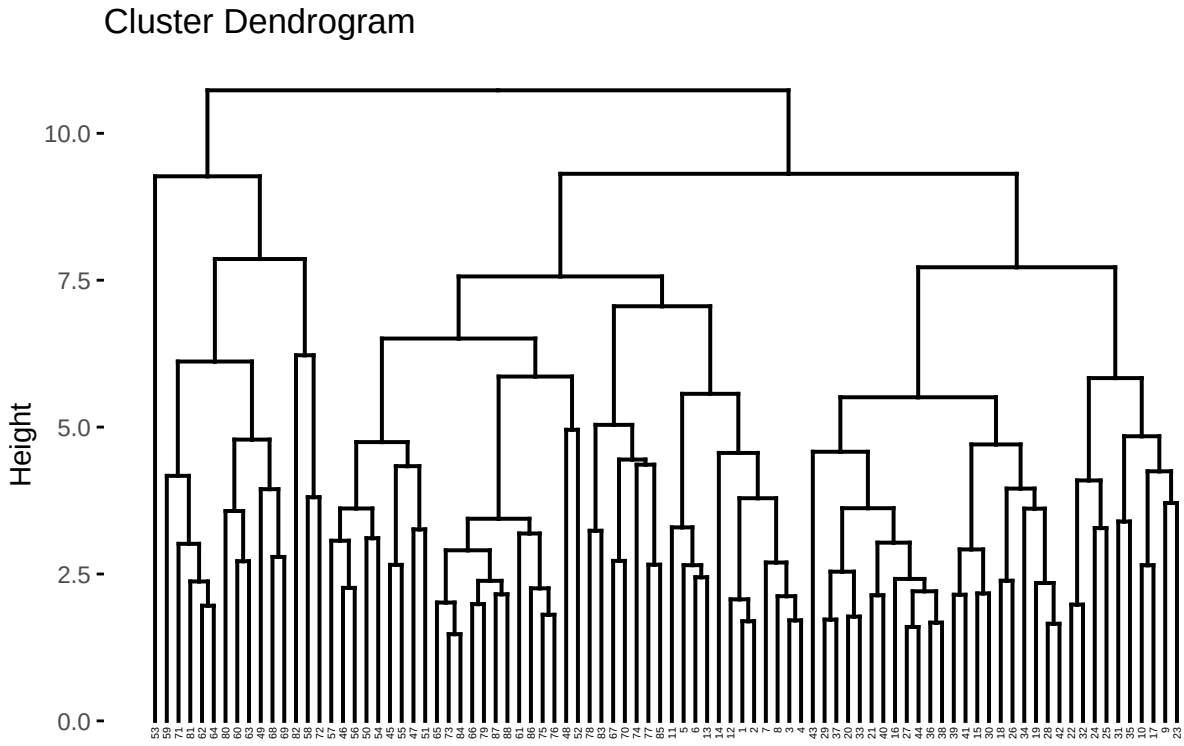
```
## Rows: 88
## Columns: 19
## $ Ceramic.Name <chr> "FLQ-1-b", "FLQ-2-b", "FLQ-3-b", "FLQ-4-b", "FLQ-5-b", "F~
## $ Part         <chr> "Body", "Body", "Body", "Body", "Body", "Body", "Body", "~
## $ Na2O         <dbl> 0.62, 0.57, 0.49, 0.89, 0.03, 0.62, 0.45, 0.59, 0.42, 0.5~
## $ MgO          <dbl> 0.38, 0.47, 0.19, 0.30, 0.36, 0.18, 0.33, 0.45, 0.53, 0.4~
## $ Al2O3        <dbl> 19.61, 21.19, 18.60, 18.01, 18.41, 18.82, 17.65, 21.42, 2~
## $ SiO2         <dbl> 71.99, 70.09, 74.70, 74.19, 73.99, 73.79, 74.99, 71.46, 6~
## $ K2O          <dbl> 4.84, 4.98, 3.47, 4.01, 4.33, 4.28, 3.53, 3.47, 3.81, 4.5~
## $ CaO          <dbl> 0.31, 0.49, 0.43, 0.27, 0.65, 0.30, 0.70, 0.35, 0.74, 0.2~
## $ TiO2         <dbl> 0.07, 0.09, 0.06, 0.09, 0.05, 0.04, 0.07, 0.05, 0.16, 0.2~
## $ Fe2O3        <dbl> 1.18, 1.12, 1.07, 1.23, 1.19, 0.96, 1.28, 1.20, 2.81, 1.1~
## $ MnO          <int> 630, 380, 420, 460, 380, 350, 650, 500, 340, 330, 320, 42~
## $ CuO          <int> 10, 20, 20, 20, 40, 20, 20, 10, 40, 20, 70, 0, 50, 0, 40,~
## $ ZnO          <int> 70, 80, 50, 70, 90, 80, 90, 70, 120, 70, 40, 90, 100, 100~
## $ PbO2         <int> 10, 40, 50, 60, 40, 10, 90, 50, 30, 20, 20, 30, 10, 90, 2~
## $ Rb2O         <int> 430, 430, 380, 380, 360, 390, 410, 380, 370, 350, 450, 43~
## $ SrO          <int> 0, -10, 40, 10, 10, 10, 30, 70, 20, 10, 10, 10, 30, 20, 1~
## $ Y2O3         <int> 40, 40, 40, 40, 30, 40, 30, 40, 30, 40, 40, 40, 30, 50, 5~
## $ ZrO2         <int> 80, 100, 80, 70, 80, 80, 90, 80, 150, 130, 120, 80, 80, 2~
## $ P2O5         <int> 90, 110, 200, 210, 150, 130, 140, 440, 180, 150, 140, 100~
```

[1] https://archive.ics.uci.edu/ml/datasets/Chemical+Composition+of+Ceramic+Samples

```r
# BLOCO 2B
dat_sc <- scale(dat[,-(1:2)])
p1 <- factoextra::fviz_nbclust(dat_sc, kmeans, method = 'wss')
p2 <- factoextra::fviz_nbclust(dat_sc, kmeans, method = 'silhouette')
gridExtra::grid.arrange(p1, p2, ncol = 2, heights = grid::unit(8, c('cm')))
```



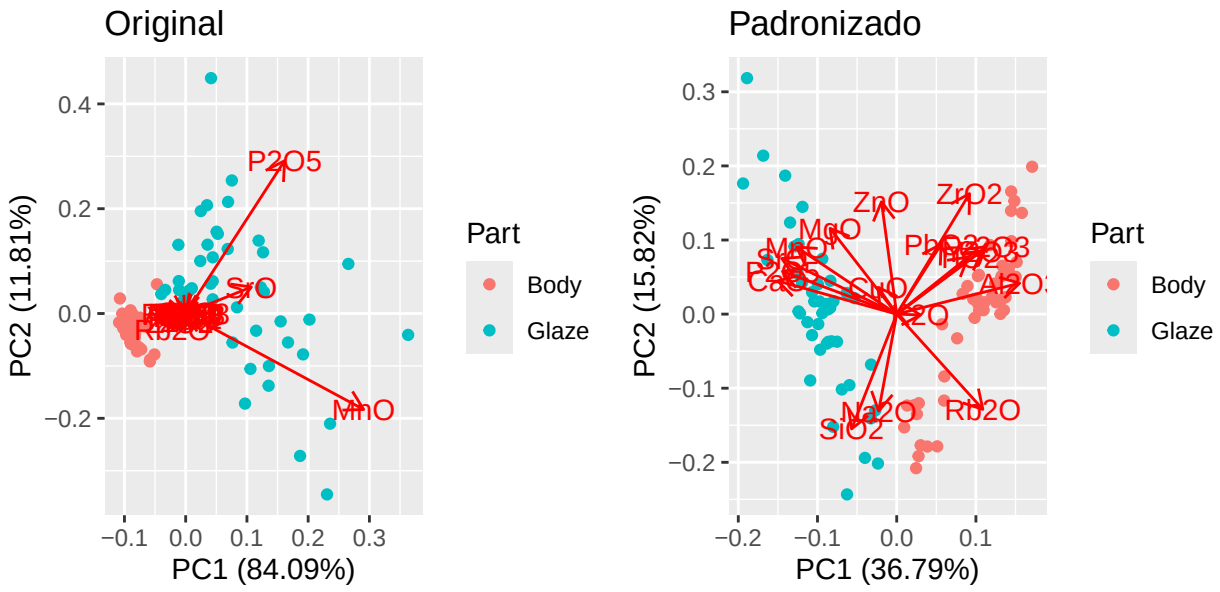

```r
# BLOCO 2C
```



```r
# BLOCO 2D
cp <- prcomp(dat[,-(1:2)])
cp_sc <- prcomp(dat[,-(1:2)], scale = TRUE)
cp_sc$rotation[,1:5]
```

```
##                PC1          PC2          PC3          PC4         PC5
## Na2O  -0.05483195 -0.314469753  0.237672805  0.503819096  0.10825114
## MgO   -0.20226122  0.278199509  0.107852777  0.089277827  0.12741566
## Al2O3  0.37313171  0.100440809 -0.004056619 -0.056817303  0.01186507
## SiO2  -0.13694588 -0.372534736 -0.231856245 -0.109454315 -0.44653673
## K2O    0.07173101 -0.002766595 -0.420006430  0.196084190  0.59864515
## CaO   -0.36096192  0.107597943  0.165020003  0.006834913  0.12518019
## TiO2   0.21661885  0.184890032  0.399860774  0.064318683 -0.15870652
## Fe2O3  0.27181505  0.218608366  0.313154202  0.276665910  0.08790755
## MnO   -0.30431777  0.217667918 -0.224758953  0.115220934  0.12826242
## CuO   -0.05636868  0.086801188 -0.048335602  0.657284732 -0.24814034
## ZnO   -0.04625579  0.364465222 -0.359421275  0.203704566 -0.35982559
## PbO2   0.13497525  0.226801712 -0.348008924  0.142393170 -0.22306849
## Rb2O   0.26154816 -0.309094340 -0.167713312  0.156814841  0.15207618
## SrO   -0.34865074  0.180487805 -0.056219645 -0.160701698  0.08200965
## Y2O3   0.25823120  0.204640640 -0.205111805 -0.096323115  0.24370097
## ZrO2   0.22114278  0.390995861  0.092597974 -0.185622680 -0.07434329
## P2O5  -0.34664031  0.138657897  0.177296428  0.050895887  0.11627896
```

```r
# BLOCO 2E
library(ggfortify)
p1 <- autoplot(cp, data = dat, colour = 'Part', main = 'Original',
               loadings = T, loadings.label = T, type = 'raw')
p2 <- autoplot(cp_sc, data = dat, colour = 'Part', main = 'Padronizado',
               loadings = T, loadings.label = T, type = 'raw')
gridExtra::grid.arrange(p1, p2, ncol = 2, heights = grid::unit(8, c('cm')))
```



Q3. **(2.5)** Ainda com os dados da Questão 2, responda os itens a seguir considerando as informações dos Blocos.

a. (0.5) Detalhe o que ocorre no `Bloco 3A`.
b. (0.5) O que você diria a respeito do modelo `fit0` apresentado no `Bloco 3B`?
c. (0.5) O que você diria a respeito da predição do `Bloco 3C`?
d. (0.5) Detalhe o que ocorre no `Bloco 3D`.
e. (0.5) O que você diria a respeito da predição do `Bloco 3E`? Compare com a predição do item **c**.

```
# BLOCO 3A
```

```
dat$y <- 1
dat$y[dat$Part == 'Glaze'] <- 0
dat$y <- (dat$y)
table(dat$Part, dat$y)
```

```
##
##          0  1
##   Body   0 44
##   Glaze 44  0
```

```
dplyr::glimpse(dat)
```

```
## Rows: 88
## Columns: 20
## $ Ceramic.Name <chr> "FLQ-1-b", "FLQ-2-b", "FLQ-3-b", "FLQ-4-b", "FLQ-5-b", "F~
## $ Part         <chr> "Body", "Body", "Body", "Body", "Body", "Body", "Body", "~
## $ Na2O         <dbl> 0.62, 0.57, 0.49, 0.89, 0.03, 0.62, 0.45, 0.59, 0.42, 0.5~
## $ MgO          <dbl> 0.38, 0.47, 0.19, 0.30, 0.36, 0.18, 0.33, 0.45, 0.53, 0.4~
## $ Al2O3        <dbl> 19.61, 21.19, 18.60, 18.01, 18.41, 18.82, 17.65, 21.42, 2~
## $ SiO2         <dbl> 71.99, 70.09, 74.70, 74.19, 73.99, 73.79, 74.99, 71.46, 6~
## $ K2O          <dbl> 4.84, 4.98, 3.47, 4.01, 4.33, 4.28, 3.53, 3.47, 3.81, 4.5~
## $ CaO          <dbl> 0.31, 0.49, 0.43, 0.27, 0.65, 0.30, 0.70, 0.35, 0.74, 0.2~
## $ TiO2         <dbl> 0.07, 0.09, 0.06, 0.09, 0.05, 0.04, 0.07, 0.05, 0.16, 0.2~
## $ Fe2O3        <dbl> 1.18, 1.12, 1.07, 1.23, 1.19, 0.96, 1.28, 1.20, 2.81, 1.1~
## $ MnO          <int> 630, 380, 420, 460, 380, 350, 650, 500, 340, 330, 320, 42~
## $ CuO          <int> 10, 20, 20, 20, 40, 20, 20, 10, 40, 20, 70, 0, 50, 0, 40,~
## $ ZnO          <int> 70, 80, 50, 70, 90, 80, 90, 70, 120, 70, 40, 90, 100, 100~
## $ PbO2         <int> 10, 40, 50, 60, 40, 10, 90, 50, 30, 20, 20, 30, 10, 90, 2~
## $ Rb2O         <int> 430, 430, 380, 380, 360, 390, 410, 380, 370, 350, 450, 43~
## $ SrO          <int> 0, -10, 40, 10, 10, 10, 30, 70, 20, 10, 10, 10, 30, 20, 1~
## $ Y2O3         <int> 40, 40, 40, 40, 30, 40, 30, 40, 30, 40, 40, 40, 30, 50, 5~
## $ ZrO2         <int> 80, 100, 80, 70, 80, 80, 90, 80, 150, 130, 120, 80, 80, 2~
## $ P2O5         <int> 90, 110, 200, 210, 150, 130, 140, 440, 180, 150, 140, 100~
## $ y            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

```
set.seed(4); itrain <- sort(sample(1:nrow(dat), floor(.6*nrow(dat))))
train <- dat[itrain,-(1:2)]
test <- dat[-itrain,-(1:2)]
```

```r
# BLOCO 3B
```

```r
fit0 <- glm(y ~ ., data = train, family = 'binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(fit0)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.273e+04  8.947e+08       0        1
## Na2O        -2.403e+02  9.098e+06       0        1
## MgO         -2.359e+02  9.209e+06       0        1
## Al2O3       -2.268e+02  9.036e+06       0        1
## SiO2        -2.295e+02  9.037e+06       0        1
## K2O         -2.359e+02  9.067e+06       0        1
## CaO         -2.338e+02  9.029e+06       0        1
## TiO2        -2.342e+02  8.875e+06       0        1
## Fe2O3       -2.298e+02  8.990e+06       0        1
## MnO          4.306e-04  3.951e+02       0        1
## CuO         -5.109e-02  3.432e+03       0        1
## ZnO          2.671e-02  2.562e+03       0        1
## PbO2         1.095e-02  5.345e+03       0        1
## Rb2O         1.576e-02  2.290e+03       0        1
## SrO         -1.143e-02  1.303e+03       0        1
## Y2O3         1.307e-01  9.269e+03       0        1
## ZrO2        -4.609e-02  4.338e+03       0        1
## P2O5         8.299e-03  3.818e+02       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7.2010e+01  on 51  degrees of freedom
## Residual deviance: 4.1743e-10  on 34  degrees of freedom
## AIC: 36
##
## Number of Fisher Scoring iterations: 25
```

```r
mctest::mctest(fit0)
```

```
##
## Call:
## omcdiag(mod = mod, Inter = TRUE, detr = detr, red = red, conf = conf,
##     theil = theil, cn = cn)
##
##
## Overall Multicollinearity Diagnostics
##
##                            MC Results detection
## Determinant |X'X|:             0.0000         1
## Farrar Chi-Square:          1152.1125         1
## Red Indicator:                 0.3692         0
## Sum of Lambda Inverse: 1120772.2819         1
```

```
## Theil's Method:              -1.2115         0
## Condition Number:         69978.2618         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

```
sort(car::vif(fit0), decreasing = TRUE)
```

```
##        Al2O3          CaO         SiO2          K2O        Fe2O3         Na2O
## 6.875856e+05 6.083616e+05 2.458227e+05 2.416777e+04 1.148067e+04 5.531171e+03
##          MgO         TiO2          SrO          MnO         ZrO2         Rb2O
## 1.376311e+03 1.067401e+02 5.109480e+01 2.552851e+01 1.975223e+01 1.528506e+01
##         PbO2         P2O5         Y2O3          ZnO          CuO
## 1.367754e+01 9.331493e+00 6.219689e+00 3.695851e+00 2.500136e+00
```

```
# BLOCO 3C
```

```
pred <- round(predict(fit0, test, type = 'response'))
(cm <- table(test$y, pred))
```

```
##    pred
##      0  1
##   0 19  0
##   1  0 17
```

```
# BLOCO 3D
```

```
pc <- prcomp(dat[,-c(1:2,20)], scale = TRUE)
train_pc <- cbind(train['y'], pc1 = pc$x[itrain,1])
train_pc <- cbind(train_pc, pc2 = pc$x[itrain,2])
test_pc <- cbind(test['y'], pc1 = pc$x[-itrain,1])
test_pc <- cbind(test_pc, pc2 = cp$x[-itrain,2])

fit1 <- glm(y ~ ., data = train_pc, family = 'binomial')
```

```
# BLOCO 3E
```

```
pred <- round(predict(fit1, test_pc, type = 'response'))
(cm <- table(test_pc$y, pred))
```

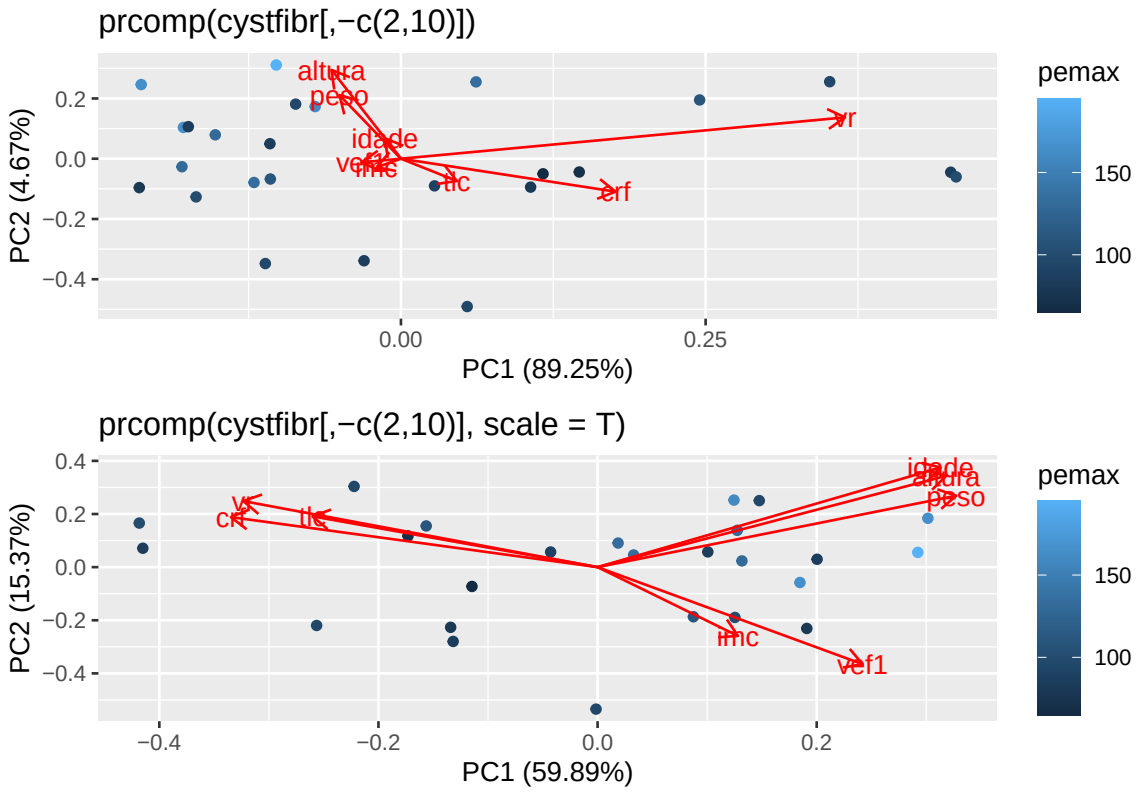
```
##    pred
##      0  1
##   0 11  8
##   1  1 16
```

Q4. **(2.0)** No Capítulo 12, Dalgaard (2008) utiliza a base de dados `cystfibr`, um banco de dados sobre capacidade respiratória discutido por Altman (1991)[2]. Abaixo estão as descrições das variáveis observadas, com volumes indicados em decilitros.

- `idade`: idade em anos
- `sexo`: 0 = masculino, 1 = feminino
- `imc`: Índice de Massa Corporal (Peso/Altura^2) como um percentual da mediana de indivíduos normais por idade
- `vef1`: Volume de Expiração Forçado em 1 segundo
- `vr`: Volume Residual, o volume restante de ar nos pulmões após uma expiração forçada
- `crf`: Capacidade Residual Funcional, o volume nos pulmões ao final da posição normal de expiração
- `cpt`: Capacidade Pulmonar Total
- `pemax`: Pressão expiratória estática máxima, variável dependente que indica a saúde do sistema respiratório (maior, melhor)

a. (0.5) Explique o que os gráficos do `BLOCO A` indicam. O que significam os percentuais nos eixos?
b. (0.5) Indique o que ocorre nos `BLOCOS B`, `C` e `D`, explicando também a relação entre eles.
c. (0.5) Aponte duas melhorias do modelo do `BLOCO E` em relação ao modelo do `BLOCO D`.
d. (0.5) Interprete os coeficientes estimados no modelo do `BLOCO E`.

```
# BLOCO A
```




```
# BLOCO B
fit0 <- lm(pemax ~ ., data = cystfibr)
summary(fit0)
```

```
##
## Call:
## lm(formula = pemax ~ ., data = cystfibr)
```

[2] *Practical Statistics for Medical Research*, Tabela 12.11, Chapman & Hall.

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.776 -17.540   3.971  14.584  36.241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.1121   185.0664  -0.055    0.957
## idade        -0.4131     4.6791  -0.088    0.931
## sexo1        -4.3718    16.4014  -0.267    0.793
## altura        0.1883     0.8511   0.221    0.828
## peso          1.2040     1.4592   0.825    0.422
## imc          -0.3796     0.5801  -0.654    0.523
## vef1          0.8295     1.1885   0.698    0.496
## vr            0.2593     0.2040   1.271    0.223
## crf          -0.4793     0.5165  -0.928    0.368
## tlc           0.5349     0.4802   1.114    0.283
##
## Residual standard error: 26.94 on 15 degrees of freedom
## Multiple R-squared:  0.599,  Adjusted R-squared:  0.3584
## F-statistic:  2.49 on 9 and 15 DF,  p-value: 0.05706
```

```
# BLOCO C
sort(car::vif(fit0), decreasing = TRUE)
```

```
##      peso     idade       crf    altura        vr      vef1      sexo       tlc
## 22.558732 18.531407 16.863994 11.073274 10.179948  5.856733  2.283452  2.195332
##       imc
##  1.848716
```

```
mctest::mctest(fit0)
```

```
##
## Call:
## omcdiag(mod = mod, Inter = TRUE, detr = detr, red = red, conf = conf,
##     theil = theil, cn = cn)
##
##
## Overall Multicollinearity Diagnostics
##
##                         MC Results detection
## Determinant |X'X|:         0.0001         1
## Farrar Chi-Square:       196.1884         1
## Red Indicator:             0.5099         1
## Sum of Lambda Inverse:    91.3916         1
## Theil's Method:            2.2567         1
## Condition Number:        125.4905         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

```
# BLOCO D
fit1 <- step(fit0, trace=0)
summary(fit1)
```

```
##
```

```
## Call:
## lm(formula = pemax ~ peso + vef1 + vr + crf + tlc, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.310 -16.724   0.722  19.260  32.688
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -32.0892    57.1098  -0.562  0.58076
## peso          1.2631     0.3626   3.484  0.00248 **
## vef1          0.9593     0.6121   1.567  0.13358
## vr            0.3113     0.1483   2.099  0.04940 *
## crf          -0.5624     0.3289  -1.710  0.10351
## tlc           0.5943     0.4234   1.404  0.17653
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.58 on 19 degrees of freedom
## Multiple R-squared:  0.5772, Adjusted R-squared:  0.4659
## F-statistic: 5.187 on 5 and 19 DF,  p-value: 0.003615
```

```r
# BLOCO E
fit2 <- lm(formula = pemax ~ peso + vef1 + vr - 1, data = cystfibr)
summary(fit2)
```

```
##
## Call:
## lm(formula = pemax ~ peso + vef1 + vr - 1, data = cystfibr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -44.71 -18.10  -0.36  19.39  41.44
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## peso  1.25825    0.30688   4.100 0.000473 ***
## vef1  0.94792    0.41104   2.306 0.030903 *
## vr    0.11159    0.03432   3.251 0.003660 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.84 on 22 degrees of freedom
## Multiple R-squared:  0.9585, Adjusted R-squared:  0.9528
## F-statistic: 169.4 on 3 and 22 DF,  p-value: 2.383e-15
```

Referências

He, Ziyang, Maolin Zhang, and Haozhe Zhang. 2016. "Data-Driven Research on Chemical Features of Jingdezhen and Longquan Celadon by Energy Dispersive x-Ray Fluorescence." *Ceramics International* 42 (4): 5123–29. https://doi.org/10.1016/j.ceramint.2015.12.030.