Lista 4 - Análise Multivariada

Francisco Rosa Dias de Miranda - 4402962

Dez 2022

Contents

Observação: Foi considerado um nível de significância de 5%, exceto quando especificado.

Lista 4: Exercício 2

A continuação dos resultados de um experimento planejado envolvendo uma reação química. As variáveis de entrada (independentes) são:

- x_1: temperatura
- x_2: concentração
- x_3: tempo

As variáveis de rendimento (dependentes) são:

- y_1 : porcentagem de material de partida inalterado
- y_2 : porcentagem convertida no produto desejado
- y_3 : porcentagem de subprodutos indesejados

```
reacao<-read_delim("data/chemical_reaction.csv",delim = ";", show_col_types = F )

kable(reacao[1:10,],caption = 'Recorte dos Resultados do Experimentos')
```

Table 1: Recorte dos Resultados do Experimentos

experiment	y1	y2	y3	x1	x2	x3
1	41.5	45.9	11.2	162	23.0	3.0
2	33.8	53.3	11.2	162	23.0	8.0
3	27.7	57.5	12.7	162	30.0	5.0
4	21.7	58.8	16.0	162	30.0	8.0
5	19.9	60.6	16.2	172	25.0	5.0
6	15.0	58.0	22.6	172	25.0	8.0
7	12.2	58.6	24.5	172	30.0	5.0
8	4.3	52.4	38.0	172	30.0	8.0
9	19.3	56.9	21.3	167	27.5	6.5
10	6.4	55.4	30.8	177	27.5	6.5

(a) Estimativa de quadrados mínimos dos coeficientes de regressão

As estimativas de mínimos quadrados são dadas por

$$\boldsymbol{\beta} = (\boldsymbol{Z}^T \cdot \boldsymbol{Z})^{-1} \cdot (\boldsymbol{Z}^T \cdot \boldsymbol{Y})$$

Por conveniência, podemos usar a implementação em linguagem R através do método `lm`.

```
Y <- as.matrix(reacao[,c('y1','y2','y3')])
fit1 <- lm(Y ~ x1 + x2 + x3, data=reacao)
fit1
```

```
##
## Call:
## lm(formula = Y ~ x1 + x2 + x3, data = reacao)
##
## Coefficients:
```

```
##                    y1          y2          y3
## (Intercept)   332.1110   -26.0353   -164.0789
## x1             -1.5460     0.4046      0.9139
## x2             -1.4246     0.2930      0.8995
## x3             -2.2374     1.0338      1.1535
```

O método fornece-nos como saída a matriz dos coeficientes de regressão para cada um dos Y_i dado X_j de nosso modelo.

(b) Suposições do modelo de regressão

Suposições:

- O erro tem média zero e variância σ^2 desconhecida.
- Erros são não-correlacionados
- Os erros têm distribuição normal
- As variáveis explicativas $X_1, ..., X_n$ são controladas pelo experimentador e medidas com erro insignificante.

Primeiramente, obtenhamos as estimativas para a média dos resíduos de cada um dos Y_i.

```
1:3 |> map_dbl(~mean(fit1$residuals[,.]))
```

```
## [1] -2.191176e-17  2.031452e-16 -2.629476e-17
```

Conforme o esperado quando o modelo faz um bom ajuste aos dados, temos as médias dos resíduos de cada uma das variáveis resposta próximas de zero.

```
cor(fit1$residuals)
```

```
##             y1          y2          y3
## y1  1.0000000 -0.1534677 -0.4842038
## y2 -0.1534677  1.0000000 -0.7474143
## y3 -0.4842038 -0.7474143  1.0000000
```

Com as estimativas dos coeficientes de correlação de Pearson, vemos que Y_2 e Y_3 possuem alta correlação inversa (-0.74).

Contudo, as correlações entre (Y_1, Y_2) e (Y_1, Y_3) são menores, o que favorece a suposição inicial das variáveis não serem correlacionadas.

(c) Significância da regressão geral

Aqui, nossa hipótese nula para os quatro testes é que a matriz de coeficientes do modelo é a matriz nula.

```
anova(fit1, test='Wilks')
```

```
## Analysis of Variance Table
##
##             Df    Wilks approx F num Df den Df    Pr(>F)
## (Intercept)  1 0.000113    38492      3     13 < 2.2e-16 ***
## x1           1 0.071886       56      3     13 1.089e-07 ***
## x2           1 0.088692       45      3     13 4.233e-07 ***
## x3           1 0.236067       14      3     13 0.0002279 ***
## Residuals   15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit1, test='Pillai')
```

```
## Analysis of Variance Table
##
##             Df  Pillai approx F num Df den Df    Pr(>F)
## (Intercept)  1 0.99989    38492      3     13 < 2.2e-16 ***
## x1           1 0.92811       56      3     13 1.089e-07 ***
## x2           1 0.91131       45      3     13 4.233e-07 ***
## x3           1 0.76393       14      3     13 0.0002279 ***
## Residuals   15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit1, test='Hotelling-Lawley')
```

```
## Analysis of Variance Table
##
##             Df Hotelling-Lawley approx F num Df den Df    Pr(>F)
## (Intercept)  1           8882.7    38492      3     13 < 2.2e-16 ***
## x1           1             12.9       56      3     13 1.089e-07 ***
## x2           1             10.3       45      3     13 4.233e-07 ***
## x3           1              3.2       14      3     13 0.0002279 ***
## Residuals   15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit1, test='Roy')
```

```
## Analysis of Variance Table
##
##             Df    Roy approx F num Df den Df    Pr(>F)
## (Intercept)  1 8882.7    38492      3     13 < 2.2e-16 ***
## x1           1   12.9       56      3     13 1.089e-07 ***
## x2           1   10.3       45      3     13 4.233e-07 ***
## x3           1    3.2       14      3     13 0.0002279 ***
## Residuals   15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Os quatro testes rejeitam H_0 a um nível de significância de 5% para todos os coeficientes testados.

Dessa forma, todas as covariáveis que testamos são importantes para explicar fontes de variação dos dados, de acordo com as diferentes metodologias utilizadas..

Lista 5: Exercício 5

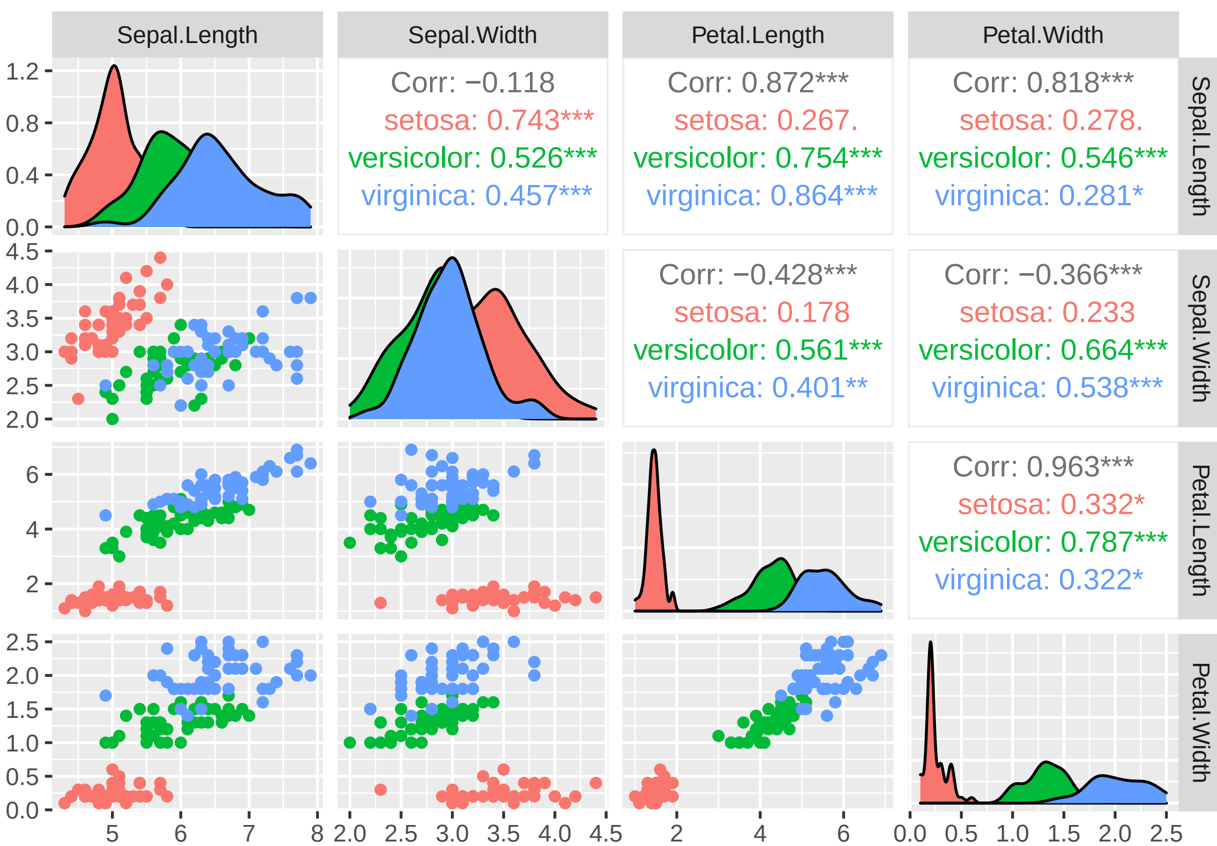
O conjunto de dados de flores Iris (originalmente apresentados em Fisher, R. A.1936), fornece as medidas em centímetros das variáveis: comprimento e largura da sépala e comprimento e largura da pétala, para 50 flores de cada uma das 3 espécies de íris.

As espécies são íris setosa, versicolor e virgínica. Vamos realizar uma análise de componentes principais.

```
data = iris[,1:4]
head(data)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          5.1         3.5          1.4         0.2
## 2          4.9         3.0          1.4         0.2
## 3          4.7         3.2          1.3         0.2
## 4          4.6         3.1          1.5         0.2
## 5          5.0         3.6          1.4         0.2
## 6          5.4         3.9          1.7         0.4
```

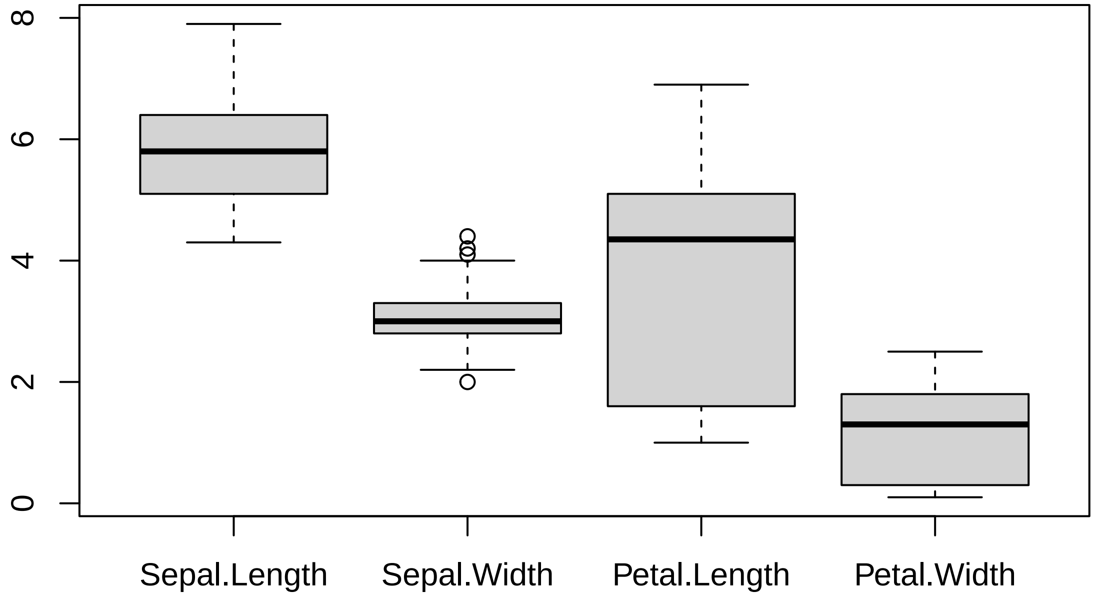
```
ggpairs(iris, columns = 1:4, aes(color = Species))
```




```
apply(data,2,sd)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.8280661    0.4358663    1.7652982    0.7622377
```

```r
boxplot(data)
```



As variáveis têm variâncias muito diferentes. Dessa forma, padronizamos os dados para que tenham média 0 e variância 1.

A utilização da mesma escala também permite que comparemos diretamente cada um dos atributos presentes em nosso conjunto de dados.

```r
data.scaled = scale(data, center = T, scale = T)
apply(data.scaled, 2, sd)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##            1            1            1            1
```

```r
Sigma = cov(data.scaled)
round(Sigma,2)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length         1.00       -0.12         0.87        0.82
## Sepal.Width         -0.12        1.00        -0.43       -0.37
## Petal.Length         0.87       -0.43         1.00        0.96
## Petal.Width          0.82       -0.37         0.96        1.00
```

```r
Eigen = eigen(Sigma)
Eigenvectors <- Eigen$vectors
colnames(Eigenvectors) <- paste0("CP", 1:4)
rownames(Eigenvectors) <- colnames(data)
Eigenvectors
```

```
##                     CP1         CP2        CP3        CP4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

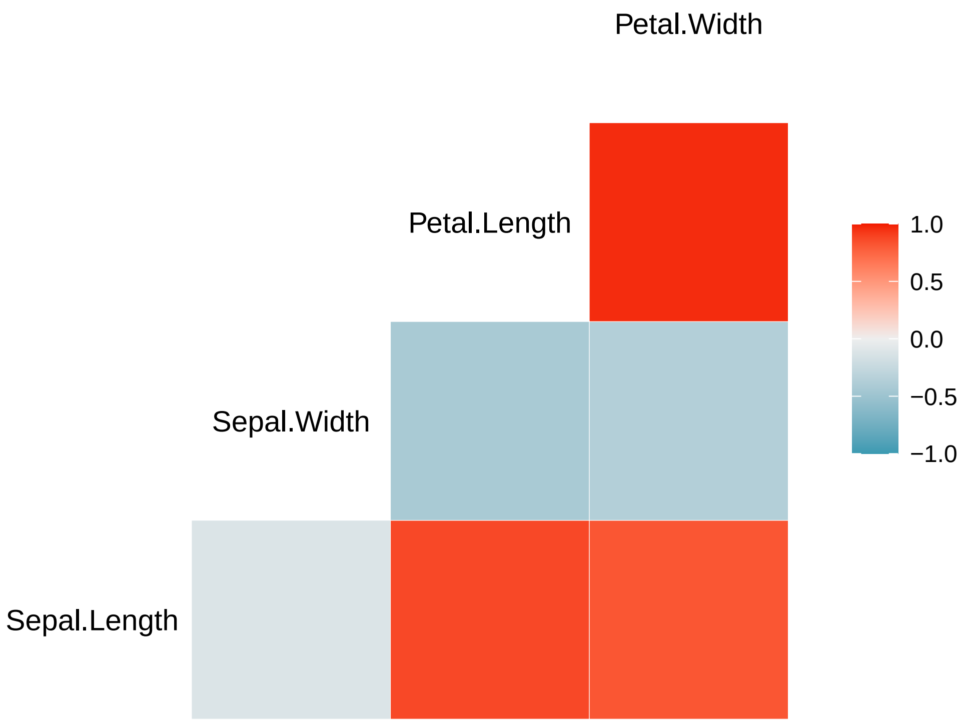
(a) Componentes principais

Primeiramente, analisamos o gráfico de componentes principais a fim de escolher o número de componentes principais.

```
pca_fit <- PCA(data, scale.unit = TRUE, graph =T, ncp = 2)
```

Vemos que as duas primeiras dimensões resumem 95% da inércia total (a inércia é a variância total do dataset i.e. o traço da matriz de correlação).

```
ggcorr(data)
```

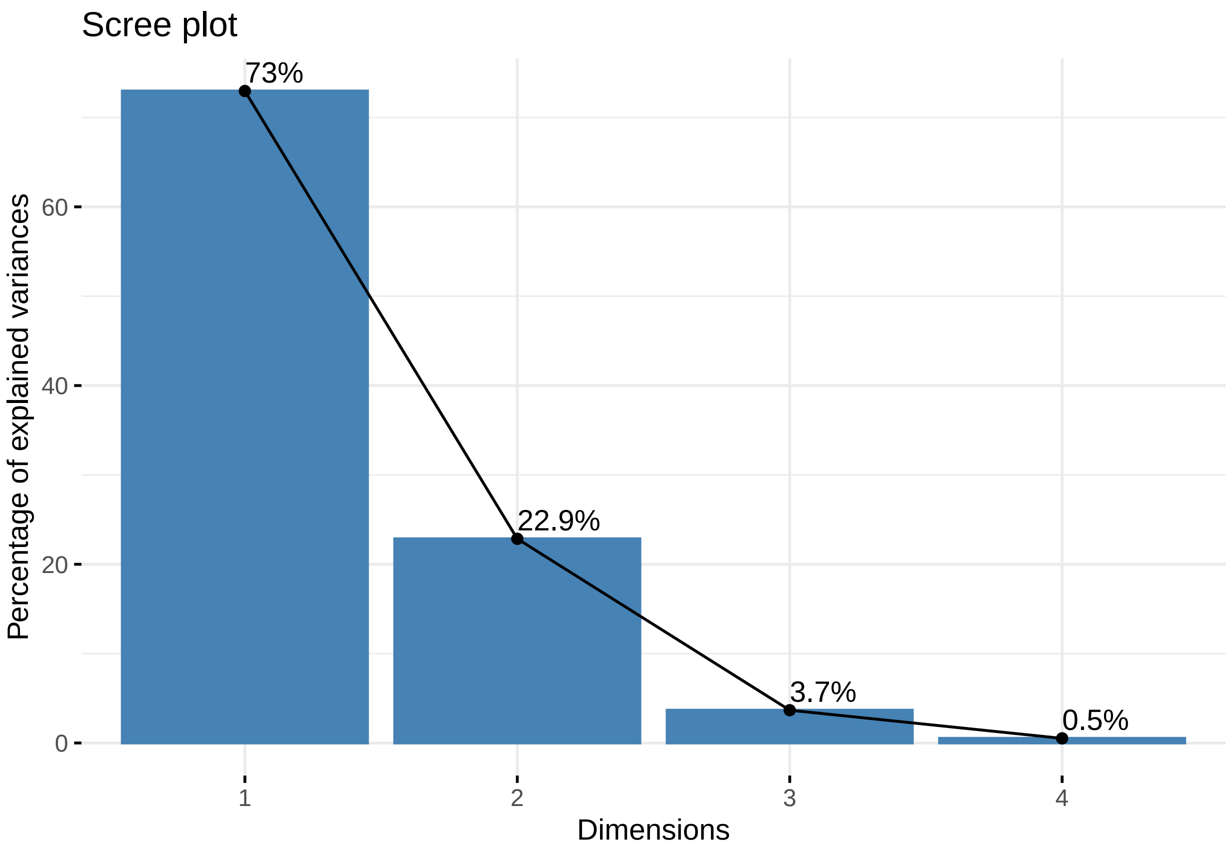


É interessante notar também que três das covariáveis são altamente correlacionadas, o que contribui para explicar o fato de que podemos descartar algumas delas, por não agregarem mais explicações ao modelo.

```
get_eigenvalue(pca_fit)
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1 2.91849782       72.9624454                    72.96245
## Dim.2 0.91403047       22.8507618                    95.81321
## Dim.3 0.14675688        3.6689219                    99.48213
## Dim.4 0.02071484        0.5178709                   100.00000
```

```
fviz_eig(pca_fit, addlabels = TRUE)
```



A função `dimdesc()` calcula o coeficiente de correlação entre uma variável em uma dimensão e faz um teste de significância.

```
dimdesc(pca_fit, axes=c(1,2))
```

```
## $Dim.1
##
## Link between the variable and the continuous variables (R-square)
## =================================================================================
##              correlation       p.value
## Petal.Length   0.9915552 3.369916e-133
## Petal.Width    0.9649790  6.609632e-88
## Sepal.Length   0.8901688  2.190813e-52
## Sepal.Width   -0.4601427  3.139724e-09
##
## $Dim.2
##
## Link between the variable and the continuous variables (R-square)
## =================================================================================
##              correlation      p.value
## Sepal.Width    0.8827163 2.123801e-50
## Sepal.Length   0.3608299 5.731933e-06
```

Escolhemos somente às duas primeiras componentes principais, que são responsáveis por explicar cerca de 95% da variância.

(b) Interprete a(s) componente(s) obtida(s)

Os autovetores representam as direções dos eixos das componentes principais, que definem a contribuição de cada combinação linear da variável para a componente principal.

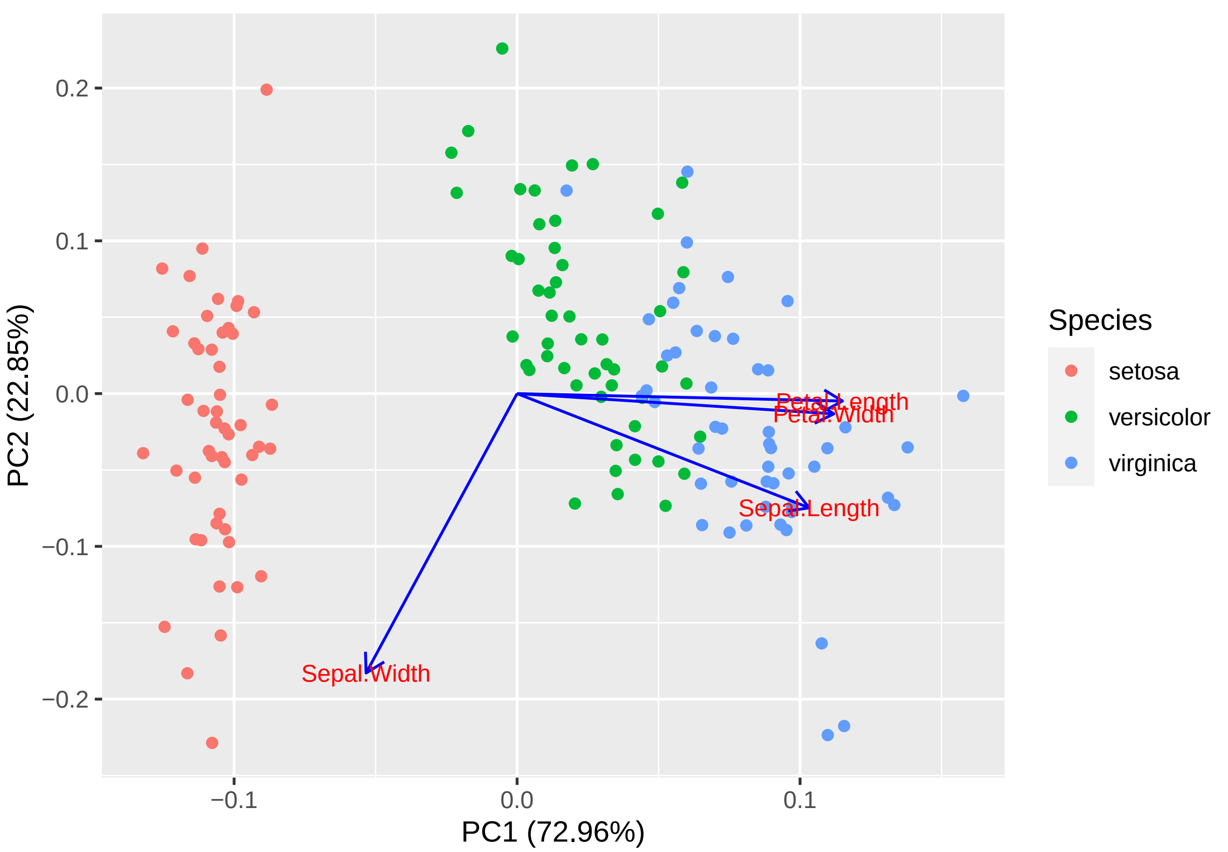
```
Eigenvectors
```

```
##                      CP1         CP2        CP3        CP4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

(c) Grupo entre as variáveis

A partir da representação gráfica do escore da primeira CP em relação escore da segunda CP e veja se podemos descobrir algum grupo entre as espécies.

```
library(ggfortify)
pca_res <- prcomp(data, scale. = TRUE)
autoplot(pca_res, data = iris, colour = 'Species',
         loadings = TRUE, loadings.colour = 'blue',
         loadings.label = TRUE, loadings.label.size = 3)
```



Lista 6: Exercício 1

Os dados a continuação são referentes a estimativas do consumo médio de proteínas de diferentes fontes de alimentos para os habitantes de 25 países europeus como publicado por Weber (1973).

```
protein <- read.csv("data/protein.csv")
kable(head(protein),caption = 'Protein Dataset')
```

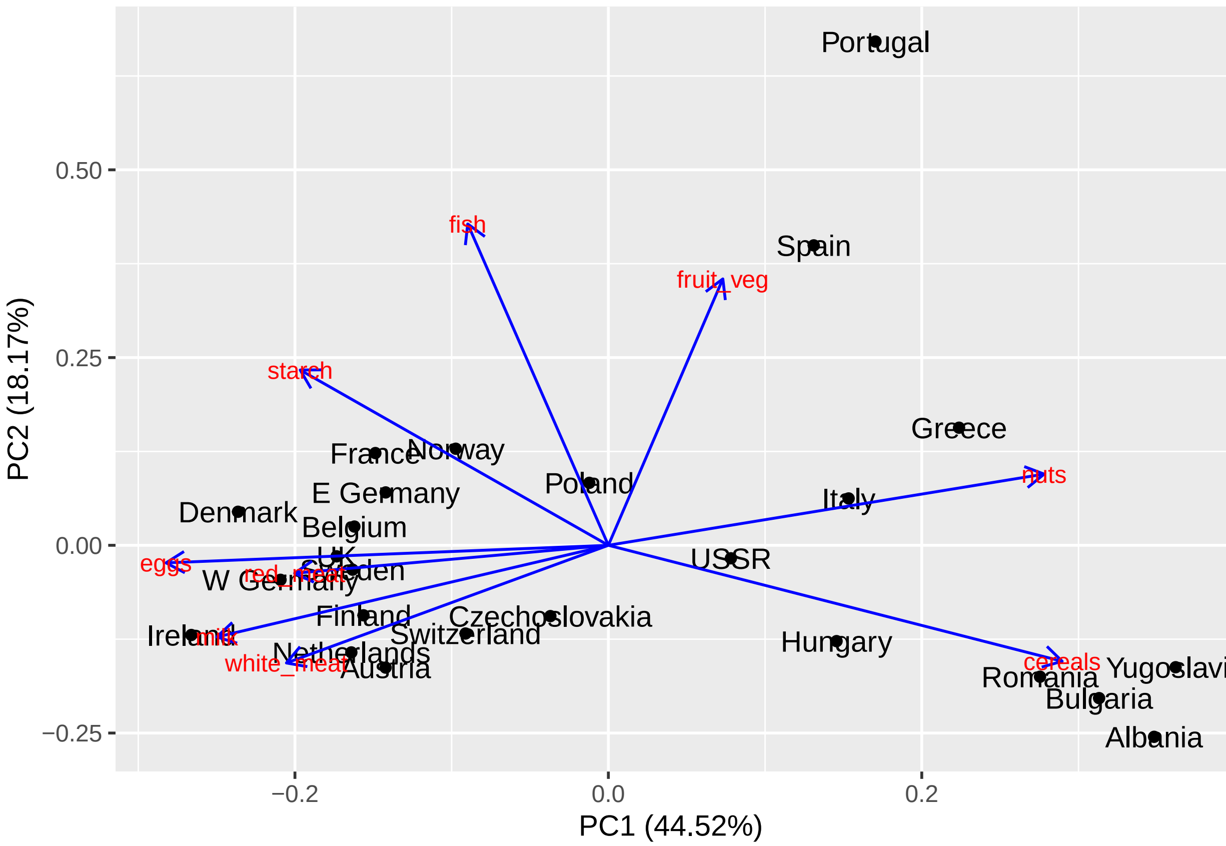
Table 2: Protein Dataset

country	red_meat	white_meat	eggs	milk	fish	cereals	starch	nuts	fruit_veg
Albania	10.1	1.4	0.5	8.9	0.2	42.3	0.6	5.5	1.7
Austria	8.9	14.0	4.3	19.9	2.1	28.0	3.6	1.3	4.3
Belgium	13.5	9.3	4.1	17.5	4.5	26.6	5.7	2.1	4.0
Bulgaria	7.8	6.0	1.6	8.3	1.2	56.7	1.1	3.7	4.2
Czechoslovakia	9.7	11.4	2.8	12.5	2.0	34.3	5.0	1.1	4.0
Denmark	10.6	10.8	3.7	25.0	9.9	21.9	4.8	0.7	2.4

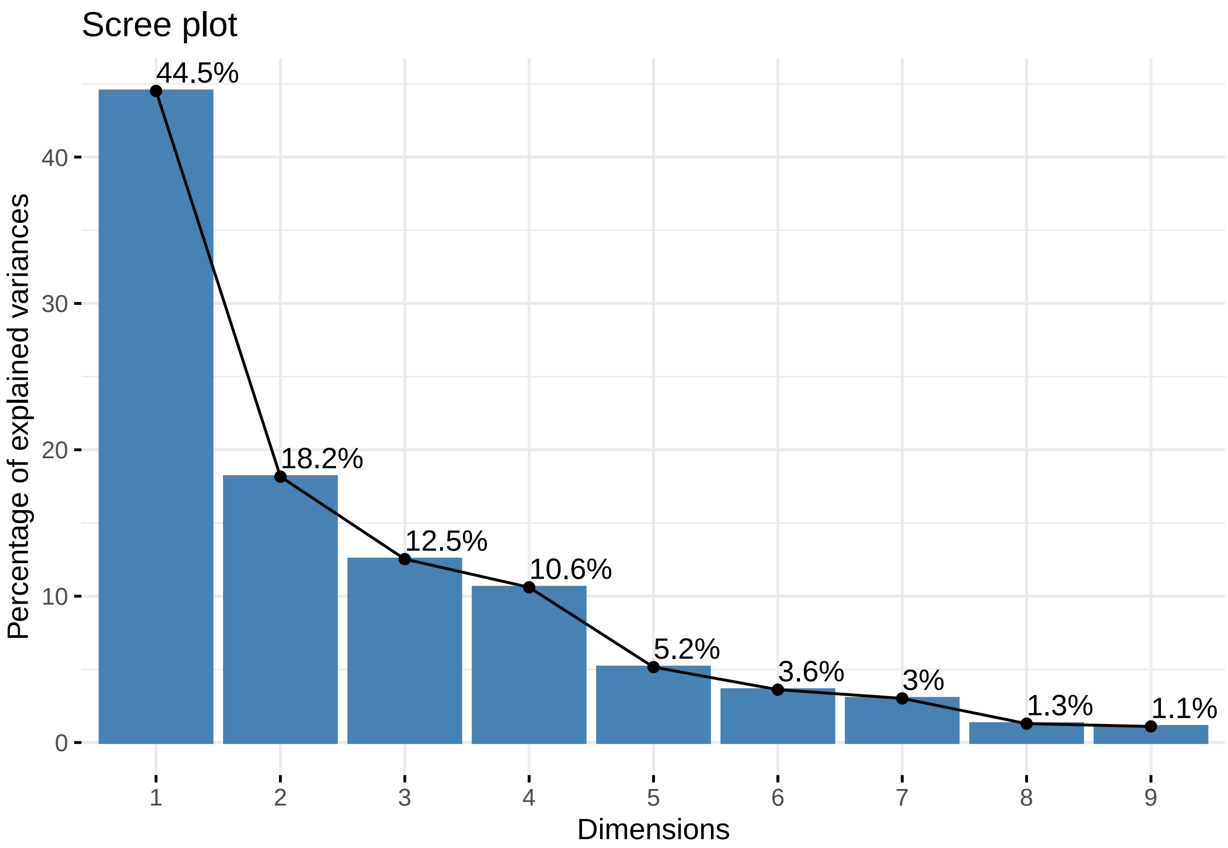
(a) Componentes pricipais

Utilizando um conjunto de dados sobre consumo de proteína de 10 diferentes de alimento para os habitantes de 25 países europeus, vamos realizar uma análise de componentes principais para investigar o relacionamento entre os países com base nestas variáveis.

```
data <- protein[,2:10]
pca_res <- prcomp(data, scale. = TRUE)
autoplot(pca_res, data = protein,
         loadings = TRUE, loadings.colour = 'blue', label = TRUE,
         loadings.label = TRUE, loadings.label.size = 3, label.label='country')
```




```
pca_fit <- PCA(data, scale.unit = TRUE, graph =F)
fviz_eig(pca_fit, addlabels = TRUE)
```



Em nossa análise, buscamos identificar fatores importantes descritos pelas variáveis observadas, para investigar o relacionamento entre os países com respeito aos fatores.

Lista 7: Exercício 11.24

Dados financeiros anuais são coletados para firmas em falência e financeiramente estáveis, por aproximadamente 2 anos.

- X1 = CF/TD = (cash flow)/(total debt)
- X2 = NI/TA = (net income)/(total assets)
- X3 = CA/CL = (current assets)/(current liabilities)
- X4 = CA/NS = (current assets)/(net sales)
- pop:
  - 0: bankrupt firms
  - 1: non bankrupt firms.

```
bank<-read.csv("data/bankruptcy.csv")
kable(bank[1:5,],caption = 'Recorte bankruptcy Dataset')
```

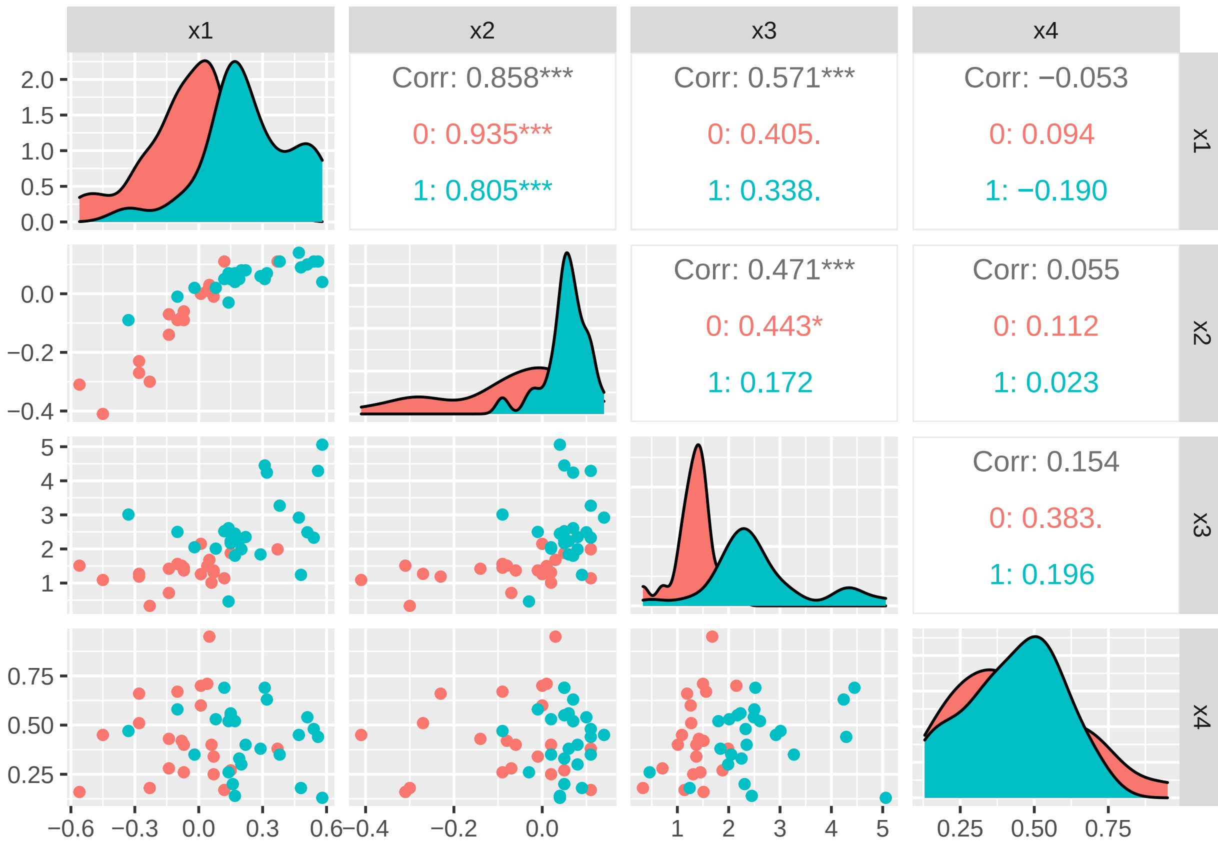
Table 3: Recorte bankruptcy Dataset

population	x1	x2	x3	x4
0	-0.45	-0.41	1.09	0.45
0	-0.56	-0.31	1.51	0.16
0	0.06	0.02	1.01	0.40
0	-0.07	-0.09	1.45	0.26
0	-0.10	-0.09	1.56	0.67

(a) Plot dos pares

Utilizando um símbolo diferente para Using a different symbol for each group, plot the data for the pairs of observations (xl,x2), (x1,x3) and (x1,x4). Does it appear as if the data are approximately bivariate normal for any of these pairs of variables?

```
ggpairs(bank,aes(color = as_factor(population)), 2:5)
```



(b) Vetor de médias amostrais

Vamos agora calcular os vetores de médias e covariâncias observados, com os $n_1 = 21$ pares de observação (x1, x2) para empresas falidas e $n_2 = 25$ pares (x1, x2) de empresas que não faliram.

```
# vetor de medias
bank |> group_by(population) |>
  summarise_all(mean)
```

```
## # A tibble: 2 x 5
##   population      x1      x2    x3    x4
##        <int>   <dbl>   <dbl> <dbl> <dbl>
## 1          0 -0.0690 -0.0814  1.37 0.438
## 2          1  0.235   0.0556  2.59 0.427
```

```
# matrizes de covariancia
# populacao = 0
bank[1:21,2:5] |> cov()
```

```
##             x1          x2         x3          x4
## x1 0.044129048 0.028476429 0.03449333 0.004147381
## x2 0.028476429 0.021002857 0.02602000 0.003441429
## x3 0.034493333 0.026020000 0.16430333 0.032781667
## x4 0.004147381 0.003441429 0.03278167 0.044579048
```

```r
# populacao == 1
bank[22:44,2:5] |> cov()
```

```
##              x1            x2         x3            x4
## x1  0.045576680  0.0094960474 0.04137806 -0.0031887352
## x2  0.009496047  0.0025675889 0.01115375 -0.0002476285
## x3  0.041378063  0.0111537549 0.85382925  0.0696804348
## x4 -0.003188735 -0.0002476285 0.06968043  0.0203632411
```