MAE 5905: Introdução à Ciência de Dados - Lista 4

Leonardo Lima - 14334311 Leonardo Makoto - 7180679

2023-06-20

Questão 1

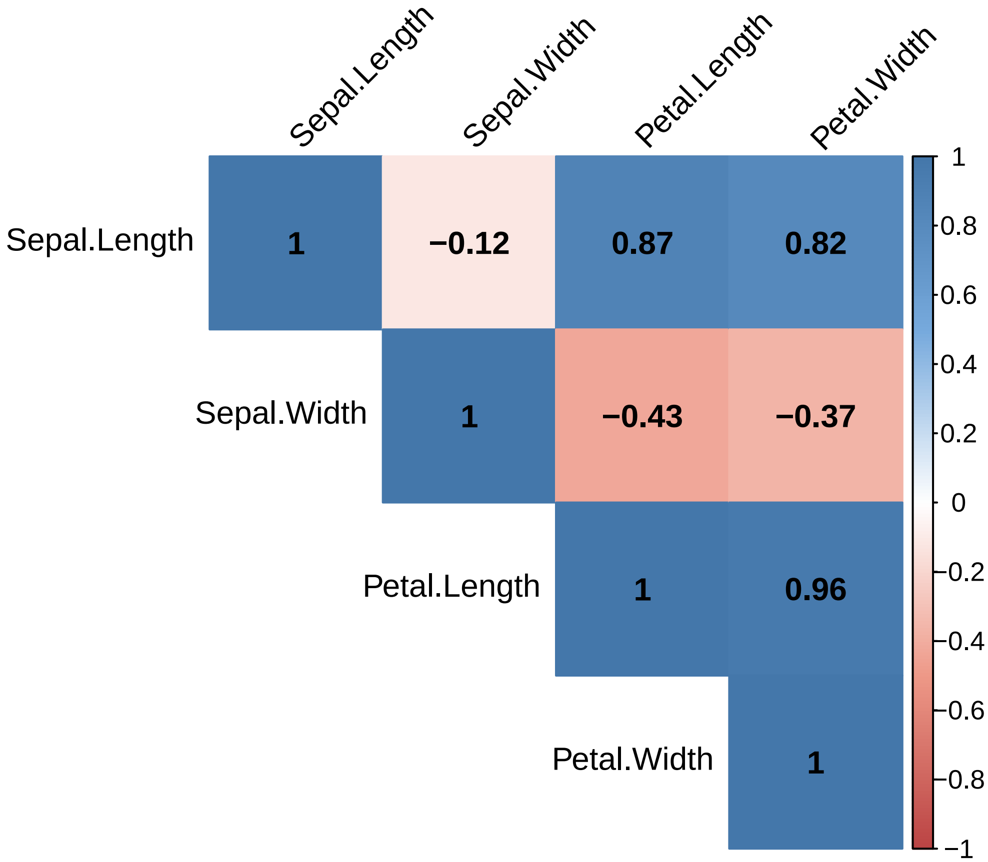
Determine as componentes principais para o conjunto de dados *iris* disponível por meio do comando `data(iris)` no pacote R.

```
# Caregando os pacotes de manipulação de dados
library(tidyverse)
# vamos carregar o pacote para produção de gráfico de correlação corrplot
library(corrplot)
# carregando a base de dados
data("iris")
```

Para os dados Iris, a variável dependente dos modelos é a factor **Species**, que contém 3 categorias: setosa, versicolor e virginica. A análise de componente principal (ACP), assim como análise fatorial (AF) e Análise de Componentes Independentes (ACI) é um método que tem o objetivo de reduzir a dimensionalidade de observações multivariadas com base em sua estrutura de dependência.

Nesse sentido, a primeira coisa a se fazer durante a aplicação do PCA é observar a correlação linear entre as variáveis explicativas do modelo que buscamos implementar:

```
# criando a correlação entre as variáveis
correlacao <- cor(iris[,1:4], method = "pearson")
# paleta de cores pasteis para usar no gráfico de correlação
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
# produzindo um gráfico para visualização
corrplot(correlacao, method = "color",
         type = "upper", col = col(200),
         addCoef.col = "black",
         tl.col="black", tl.srt=45)
```

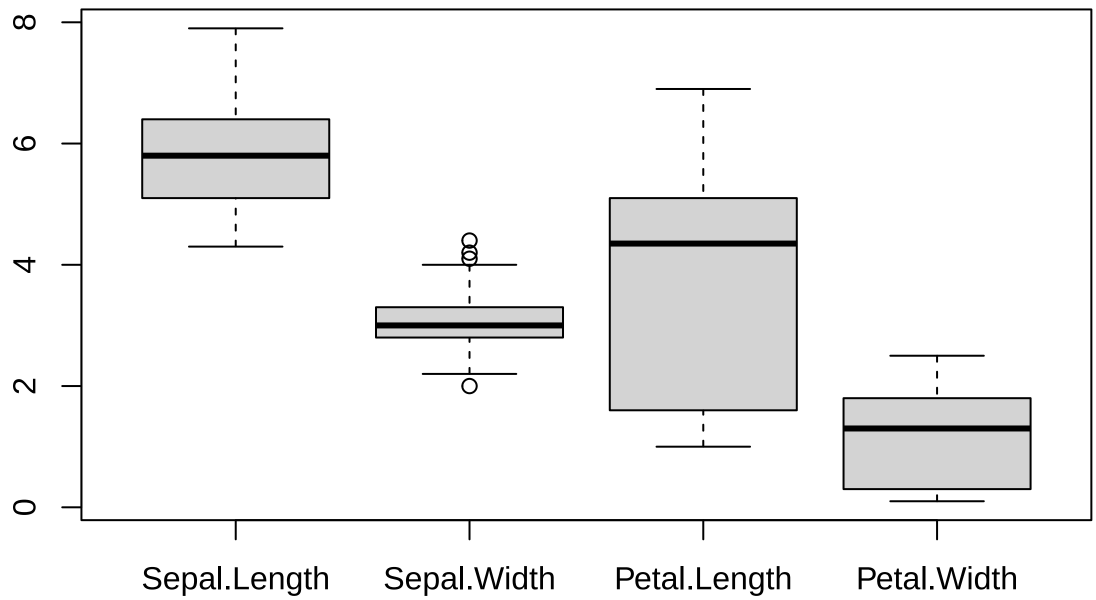



```
# observações:
# 1. method = color determina o formato do gráfico, para ser quadrados coloridos;
# 2. type = upper determina que só deve aparecer a parte superior do correlograma;
# 3. col(200) determina que o espectro de cores entre 1 e -1 tenha 200 bandas;
# 4. add.coef.col faz com que o valor da correlação seja reportado
#    junto com as cores com, além de determinar a cor do número;
# 5. tl.col r tl.srt determinam, respectivamente,
#    a cor e a inclinação do nome dos vetores.
```

De acordo com os resultados do correlograma, há uma correlação forte entre Sepal Lenght com Petal Lenght e Petal Width, assim como Petal Lenght e Petal Width. O único vetor que parece ter um comportamento significantemente distinto dos demais do ponto de vista linear é Sepal Width. No caso dessa variável, os índices de correlação são negativos com as demais e ela possuí uma correlação mais fraca.

Antes de realizar as estimativas, é preciso avaliar a dispersão dos dados para saber se é necessário padronizá-los para facilitar a interpretação dos componentes principais. Nesse caso, vamos criar um boxplot para analisar os dados:

```
# criando o boxplot
boxplot(iris[,-5])
```



Embora a dispersão dos dados não seja tão grande, há uma diferença significativa na distribuição entre Sepal Lenght e Petal Width. Nesse sentido, iremos padronizar os dados para facilitar a interpretação dos coeficientes principais. Como a questão não solicita que os dados sejam separados em diferentes amostras - para teste e treino, vamos encontrar os componentes principais utilizando todo o conjunto de dados iris.

```r
set.seed(9845)
# Como a redução de dimensionalidade é feita apenas para as variáveis independentes,
# iremos remover a variável Species do cálculo.
# calculando os componentes principais
acp <- prcomp(iris[,-5],
              center = TRUE,
              scale. = TRUE)
# observação: as opções center e scale. servem para padronizar os dados.
# 1. Center centraliza os dados ao redor de zero
# 2. scale. torna a variância das variáveis unitária
# vejamos as estimativas dos componentes principais
acp
```

```
## Standard deviations (1, .., p=4):
## [1] 1.7083611 0.9560494 0.3830886 0.1439265
##
## Rotation (n x k) = (4 x 4):
##                     PC1         PC2        PC3        PC4
## 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
```

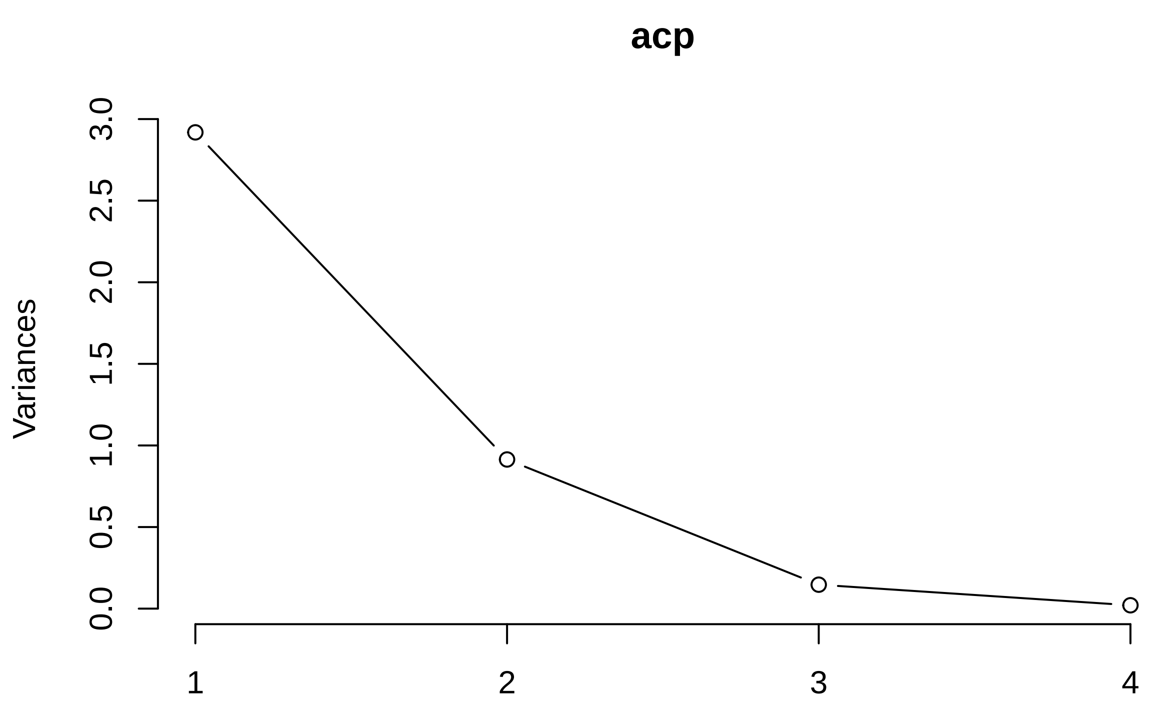
```r
summary(acp)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

Os resultados reportam 4 componentes principais. A primeira componente corresponde a aprox. 73% da variância total dos dados, enquanto a segunda corresponde a aproximadamente 23%. Em conjunto, os dois componentes respondem por aprox. 96% de toda a variabilidade das 4 variáveis, indicando que os demais seriam desnecessários, por explicarem uma parcela muito pequena dos dados.

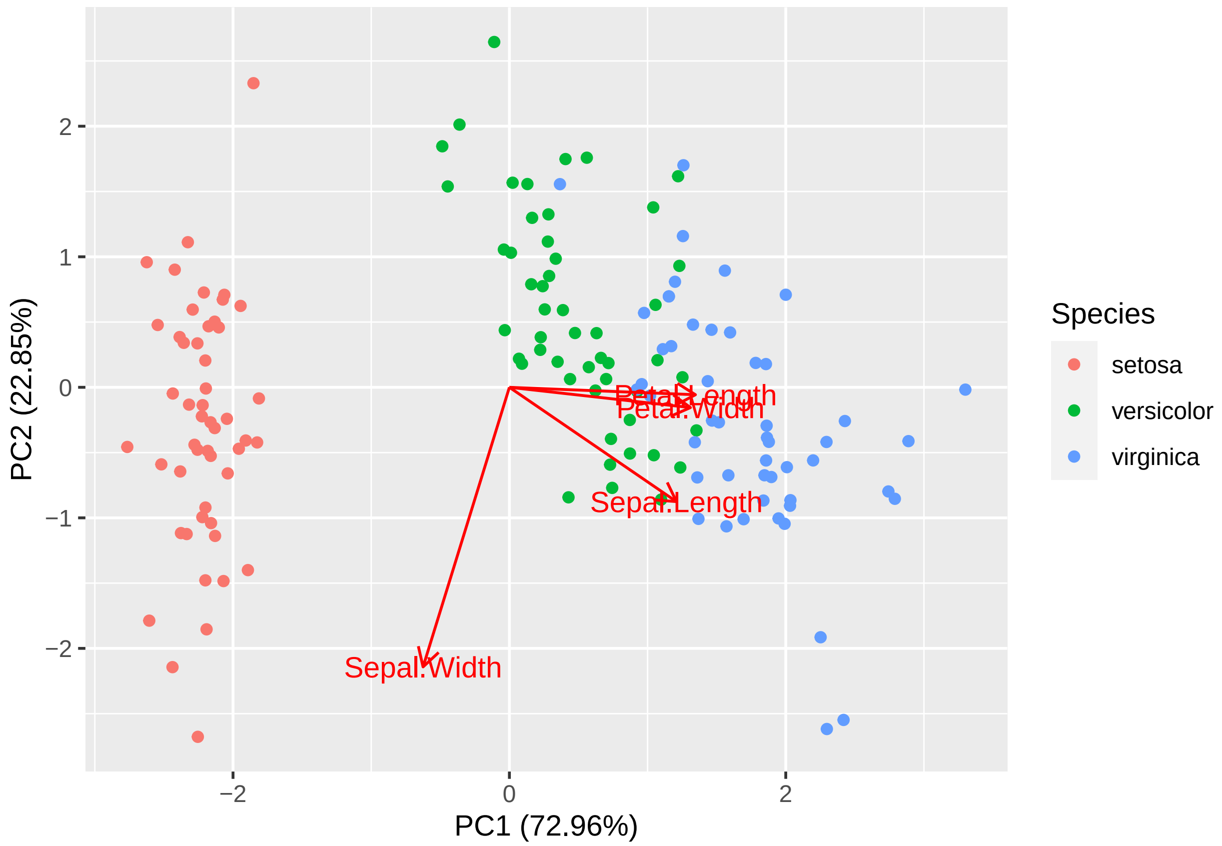
Vejamos o gráfico com os autovalores (variâncias) dos componentes principais:

```r
screeplot(acp, type = "lines")
```



Os autovalores são obtidos através do cálculo do quadrado dos coeficientes reportados como "standard deviation", anteriormente. Já a proporção da variância explicada por cada componente principal pode ser calculada através da razão entre o autovalor do componente e o somatório dos autovalores de todos os componentes. Combinando os resultados do gráfico dos autovalores (usando o Teste Scree de Cattel (1966)) com os resultados presentes na tabela anterior, confirma-se que somente os 2 primeiros componentes são necessários para o modelo. Vejamos o gráfico que mostra a relevância de cada variável em relação aos componentes:

```r
# Para isso, usaremos o pacote ggfortify,
# que permite ao ggplot interpretar os coeficientes do ACP.
library(ggfortify)
# gráfico com os autovetores e os componentes principais.
autoplot(acp, data = iris, colour = "Species",
         loadings = TRUE, loadings.label = TRUE,
         scale = 0)
```




```r
# Observações:
# 1. Loadings = TRUE determina que os autovetores devem ser reportados;
# 2. loadings.label = TRUE reporta o nome das variáveis ligadas ao vetor;
# 3. scale = 0 serve para remover a padronização dos autovetores.
```

O gráfico anterior possuí diversas características interessantes. Os valores projetados de cada vetor nos componentes principais determinam o seu nível de influência sobre aquele componente. No caso em questão, o componente principal 2 é determinado majoritariamente pelo comportamento de Sepal Width, enquanto o componente principal 1 apresenta pesos próximos para Petal Width, Lenght e Sepal Length. Além disso, o ângulo entre os vetores reportados mostra como essas variáveis são correlacionadas. Como é possível observar, Petal Lenght e Width são altamente correlacionadas e todas são pouco correlacionadas com Sepal Width.

Caso houvesse um ângulo de 90° graus entre os vetores, seria indicativo de que eles não são correlacionados. O mais próximo disso é a relação entre Sepal Lenght e Sepal Width.

Por fim, é importante destacar como o valor de cada um dos componentes principais é calculado. De acordo com os coeficientes reportados, o Componente principal 1 pode ser definido da seguinte maneira:

$$CP_1 = 0,52 * Sepal.Lenght - 0,27 * Sepal.Width + 0,58 * Petal.Lenght + 0,56 * Petal.Width$$

O segundo componente segue a mesma lógica:

$$CP_2 = -0,38 * Sepal.Lenght - 0,92 * Sepal.Width - 0,02 * Petal.Lenght - 0,07 * Petal.Width$$

Como os demais vetores explicam uma parcela insignificante da variabilidade e seguem a mesma lógica, não serão reportados.

Questão 2

Realize análise fatorial para os dados do problema anterior.

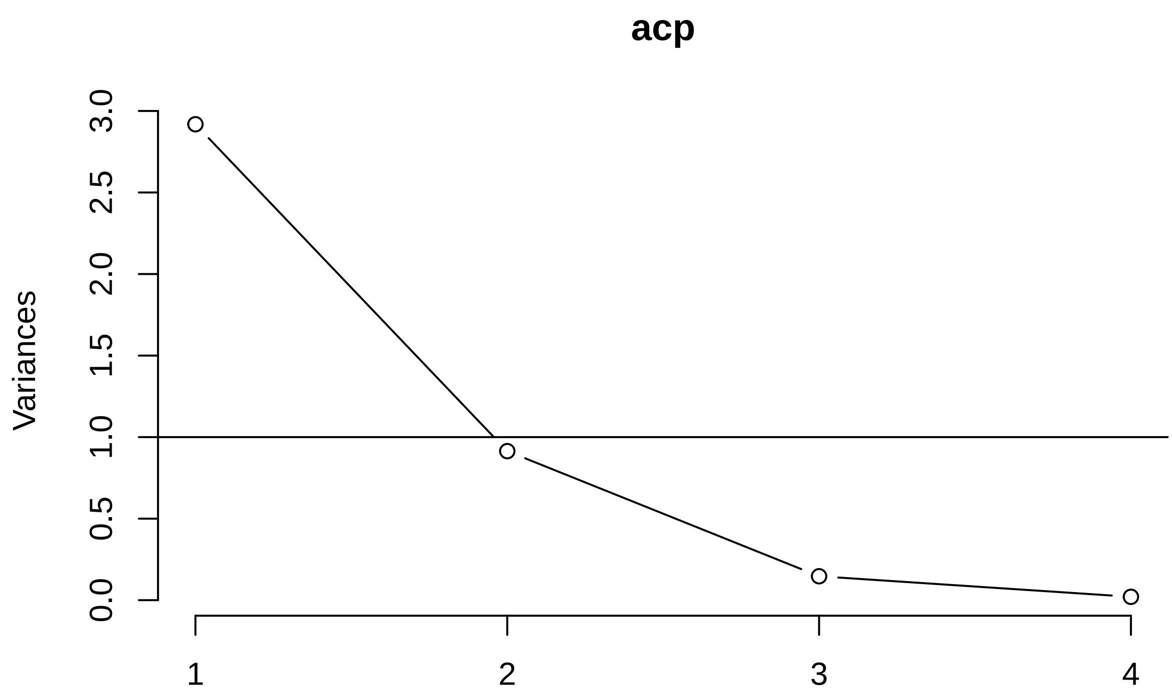
Assim como no exemplo da aula 14, para o caso da análise aplicada a **iris** não é possível realizar a análise fatorial considerando 2 fatores, pois o pacote **stats** não aceita valor superior a 1 para 4 variáveis:

```
AF <- factanal(iris[,-5], factors = 2, rotation = "varimax")
```

```
## Error in factanal(iris[, -5], factors = 2, rotation = "varimax"): 2 factors are too many for 4 varia
```

Porém, a aplicação para apenas 1 fator não é problemática, dado que a escolha do número de fatores adequada usando a Regra de Kaiser-Guttman, em que se consideram apenas os fatores com autovalores maiores que 1, indica que o número de fatores adequado é 1, divergindo da análise gráfica através do Teste Scree:

```
screeplot(acp, type = "lines")
abline(h=1)
```



Sendo assim, realizaremos as estimativas usando apenas um fator:

```r
set.seed(9845)
# Análise fatorial considerando apenas 1 fator
AF <- factanal(iris[,-5], factors = 1)
# Observações:
# 1. Como há apenas um fator, não há uma matriz de cargas fatoriais, mas apenas um vetor.
#    Assim, não é possível fazer nenhum tipo de rotação de fatores para simplificar
#    a interpretação.
AF
```

```
##
## Call:
## factanal(x = iris[, -5], factors = 1)
##
## Uniquenesses:
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        0.240        0.822        0.005        0.069
##
## Loadings:
##              Factor1
## Sepal.Length  0.872
## Sepal.Width  -0.422
## Petal.Length  0.998
## Petal.Width   0.965
##
##                Factor1
```

```
## SS loadings      2.864
## Proportion Var   0.716
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 85.51 on 2 degrees of freedom.
## The p-value is 2.7e-19
```

Há diversas informações pertinentes a serem consideradas:

1. **Uniqueness** se refere aos ruídos do modelo. É a proporção da variabilidade de cada variável (a variância específica) que não pode ser explicada pelo único fator que criamos. Nota-se que o fator explica consideravelmente bem a variabilidade de Petal Lenght e Width, além de explicar grande parte da variabilidade de Sepal Lenght. No entanto, o fator contribui menos de 20% para a variância de Sepal Width.

2. **Loadings** se refere as cargas fatoriais. Esses valores indicam a importância do fator 1 na composição de cada uma das variáveis. Valores (em módulo) próximos de 1 indicam que o fator é muito relevante para explicar a variável. Já próximos a zero, baixa. Assim como adiantado no resultado sobre Uniqueness, as cargas fatoriais são consideravelmente elevadas para as variáveis Petal Lenght, Width e Sepal Lenght, indicando que elas são bem explicadas pelo fator 1. Já Sepal Width apresenta um valor, em módulo, consideravelmente menor que os demais, indicando que ela não é bem explicada pelo fator 1.

3. **Comunalidade**: a comunalidade de cada variável não é reportada diretamente no output, mas pode ser calculada por duas maneiras: (i) através da soma dos quadrados das cargas fatoriais de cada fator; e (ii) fazendo a conta: 1 - **Uniqueness** de cada variável. A comunalidade se refere a parcela da variância da variável que é explicada pelos fatores. No caso em questão, a comunalidade será:

```r
# cálculo de comunalidade
apply(AF$loadings^2,1,sum)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.7597716    0.1781358    0.9950964    0.9306666
```

Como é possível perceber, o fator explica praticamente toda a variabilidade de Petal Lenght e Width e a maior parte de Sepal Lenght, mas explica apenas 18% de Sepal Width, indicando que não é apropriado para endereçar a variabilidade desta variável.

4. **SS Loadings e Proportion of Var**: essa parte da tabela indica a proporção da variabilidade das variáveis explicadas por cada fator. Como há apenas um, não há a linha que reporta a variabilidade cumulativa. Os resultados indicam que o fator 1 explica aproximadamente 72% da variabilidade das variáveis.

- SS Loadings é a soma dos quadrados das cargas fatoriais. Pode ser obtida através da conta:

```r
sum(AF$loadings^2)
```

```
## [1] 2.86367
```

5. A última parte do output se refere a um teste de hipótese que avalia se o número de fatores no modelo é suficiente para capturar a dimensionalidade dos dados. Com o p-valor é próximo de zero, rejeitamos a hipótese nula, o que indica que o número de fatores do modelo é pequeno demais. Esse teste só é reportado porque as estimações dos parâmetros do modelo fatorial do pacote `stats` são feitas utilizando o método de máxima verossimilhança. Podemos estimar as matrizes de covariâncias $\hat{\Sigma}$ e a residual através dos seguintes comandos:

```r
# matriz com Lambdas (cargas fatoriais)
Lambda <- AF$loadings
# matriz de ruídos
Psi <- diag(AF$uniquenesses)
# matriz de covariâncias amostral
S <- AF$correlation
# matriz de covariâncias estimada
Sigma <- Lambda %*% t(Lambda) + Psi
# Observação: t(Lambda) transpõe a matriz de cargas fatoriais
# vejamos a matriz de covariância estimada
Sigma
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000003  -0.3678893    0.8695090   0.8408888
## Sepal.Width    -0.3678893   1.0000011   -0.4210253  -0.4071671
## Petal.Length    0.8695090  -0.4210253    1.0000964   0.9623424
## Petal.Width     0.8408888  -0.4071671    0.9623424   1.0000000
```

```r
# matriz residual
mat_residual <- round(S - Sigma, 6)
mat_residual
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length     0.000000    0.250320     0.002245   -0.022948
## Sepal.Width      0.250320   -0.000001    -0.007415    0.041041
## Petal.Length     0.002245   -0.007415    -0.000096    0.000523
## Petal.Width     -0.022948    0.041041     0.000523    0.000000
```

Como é possível observar para a matriz residual, os valores que relacionam Sepal Width e Length não são próximos de zero, indicando que o modelo fatorial precisaria de um fator adicional para contemplar esta relação. Para as demais, o modelo para ser adequado. Há a possibilidade de usar 2 fatores através do pacote `psych`, mas como será apresentado abaixo, a depender do método utilizado para estimação dos parâmetros, eles produzem casos ultra-Heywood (quando a comunalidade excede 1). Um caso ultra-Heywood implica que um dos fatores únicos possuí uma variância negativa, que é um indicativo claro que algo está errado e as estimativas não são confiáveis. Abaixo segue um exemplo do resultado usando 2 fatores e o método de fatoração de minimização dos resíduos (default do pacote):

```r
library(psych)
```

```r
set.seed(9845)
# aplicação da análise com dois fatores usando minres
fa2_minres <- fa(iris[,-5], nfactors = 2, rotate = "varimax")
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect.  Try a
## different factor score estimation method.
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected.  Examine the results carefully
```

```
fa2_minres
```

```
## Factor Analysis using method =  minres
## Call: fa(r = iris[, -5], nfactors = 2, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                MR1   MR2   h2     u2 com
## Sepal.Length  0.90  0.01 0.81  0.188 1.0
## Sepal.Width  -0.14  0.97 0.97  0.031 1.0
## Petal.Length  0.96 -0.29 1.01 -0.011 1.2
## Petal.Width   0.92 -0.24 0.90  0.097 1.1
##
##                        MR1  MR2
## SS loadings           2.60 1.09
## Proportion Var        0.65 0.27
## Cumulative Var        0.65 0.92
## Proportion Explained  0.70 0.30
## Cumulative Proportion 0.70 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 factors are sufficient.
##
## df null model =  6  with the objective function =  4.81 with Chi Square =  706.96
## df of  the model are -1  and the objective function was  0.11
##
## The root mean square of the residuals (RMSR) is  0.01
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  150 with the empirical chi square  0.06  with prob <  NA
## The total n.obs was  150  with Likelihood Chi Square =  15.81  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.145
## Fit based upon off diagonal values = 1
```

Mesmo alterando a especificação do modelo com relação a forma com que os escores e cargas fatoriais são calculados o algorítmo continua chegando a uma solução do tipo Heywood:

```
# aplicação da análise com dois fatores utilizando método de fator principal
fa2_pa <- fa(iris[,-5], nfactors = 2, rotate = "varimax", fm = "pa")
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected.  Examine the results carefully
```

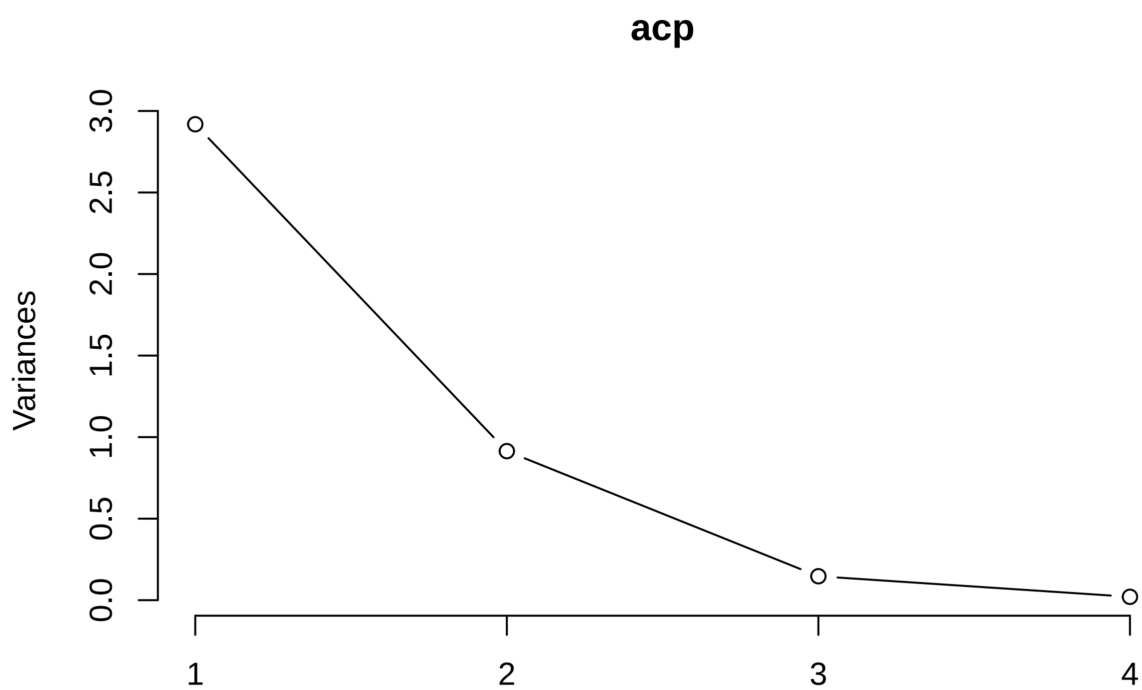
```
fa2_pa
```

```
## Factor Analysis using method =  pa
## Call: fa(r = iris[, -5], nfactors = 2, rotate = "varimax", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                PA1   PA2   h2     u2 com
## Sepal.Length  0.94 -0.01 0.88  0.120 1.0
## Sepal.Width  -0.13 -0.72 0.54  0.463 1.1
## Petal.Length  0.93  0.42 1.05 -0.046 1.4
## Petal.Width   0.88  0.35 0.89  0.111 1.3
```

```
##
##                        PA1  PA2
## SS loadings           2.54 0.82
## Proportion Var        0.63 0.20
## Cumulative Var        0.63 0.84
## Proportion Explained  0.76 0.24
## Cumulative Proportion 0.76 1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 2 factors are sufficient.
##
## df null model =  6  with the objective function =  4.81 with Chi Square =  706.96
## df of  the model are -1  and the objective function was  0
##
## The root mean square of the residuals (RMSR) is  0
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  150 with the empirical chi square  0  with prob <  NA
## The total n.obs was  150  with Likelihood Chi Square =  0.32  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.011
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                    PA1  PA2
## Correlation of (regression) scores with factors   0.99 0.94
## Multiple R square of scores with factors          0.98 0.89
## Minimum correlation of possible factor scores     0.96 0.79
```

Como a convergência dos resultados é muito dependente do método aplicado ao utilizar 2 fatores (e com base no resultado do teste de Kaiser-Guttman), optou-se por realizar a análise com base em apenas um fator, assim como apresentado anteriormente. # Questão 3 Obtenha as componentes independentes para os dados do Problema 1. ——————————————————— A Análise de Componentes Independentes transforma um conjunto de vetores em um conjunto de componentes independentes e não gaussianos. A partir do exercício 1, temos que são dois os principais componentes que maximizam a variação nas quatro variáveis do modelo são dois, como pode se observar no gráfico abaixo.

```r
screeplot(acp, type = "lines")
```



Nas tabelas abaixo, pode-se observar os coeficientes de componentes principais e a importância dos componentes. Em conjunto, os dois componentes respondem por aprox. 96% de toda a variabilidade das 4 variáveis.

```
acp$rotation
```

```
##                     PC1         PC2        PC3        PC4
## 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
```

```
summary(acp)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

Dessa forma, como são dois componentes principais, vamos assumir que são 2 os componentes independentes. Para estimar os componentes independentes, vamos usar o pacote do R fastICA.

```r
# install.packages("fastICA")
library(fastICA)
```

```
## Warning: package 'fastICA' was built under R version 4.2.3
```

```r
# install.packages("ica")
library(ica)
ica_fast <- fastICA(iris[1:4],2)
```

A matrix A é:

```r
ica_fast$A
```

```
##            [,1]       [,2]      [,3]        [,4]
## [1,]  0.7011144 -0.2112316  1.754491  0.73394756
## [2,] -0.4011070 -0.3374915 -0.106586 -0.04312812
```

Isto é,

$$X_1 = 0.70111436237236 S_1 - 0.211231632412411 S_2 + 1.75449116440755 S_3 + 0.733947559303463 S_4$$

e

$$X_2 = -0.4011069753843 84 S_1 - 0.337491549925135 S_2 - 0.106585958270931 S_3 - 0.0431281232558958 S_4$$

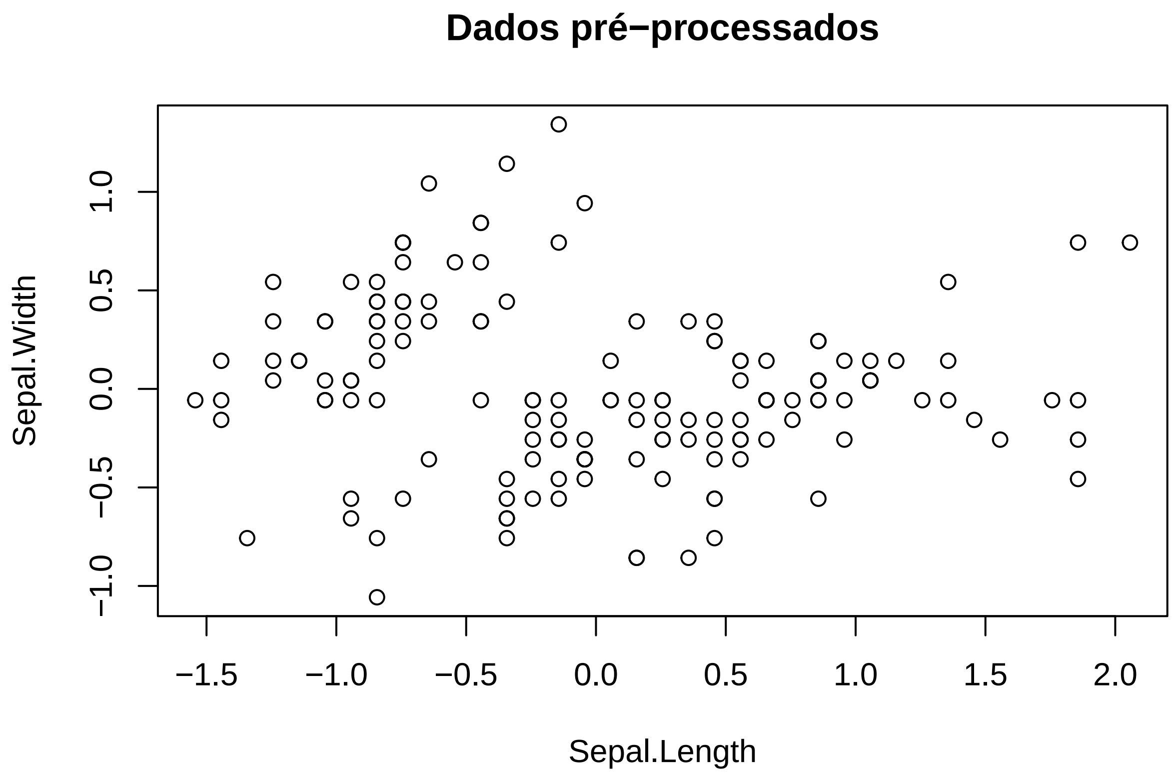
A variancia que esses dois componentes independentes explicam é a mesma variância que os componentes, isto é, 96% aproximadamente:

```r
summary(acp)
```

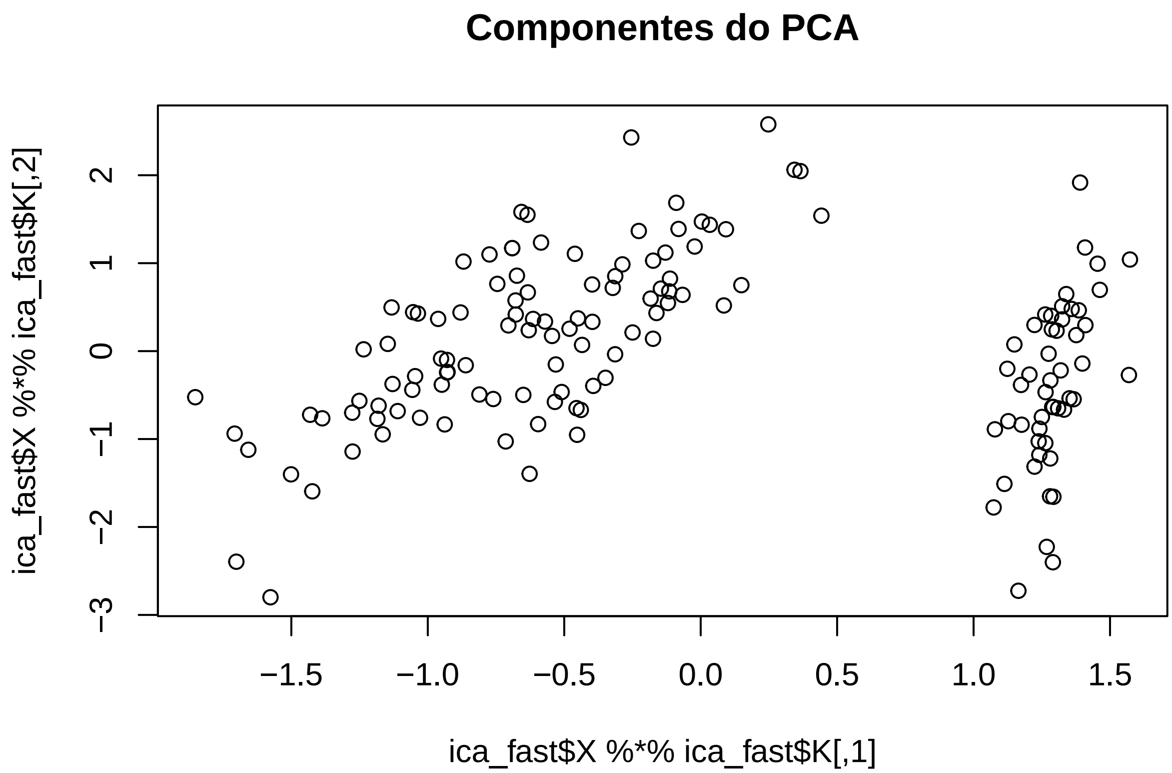
```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

Abaixo, podemos observar os gráficos

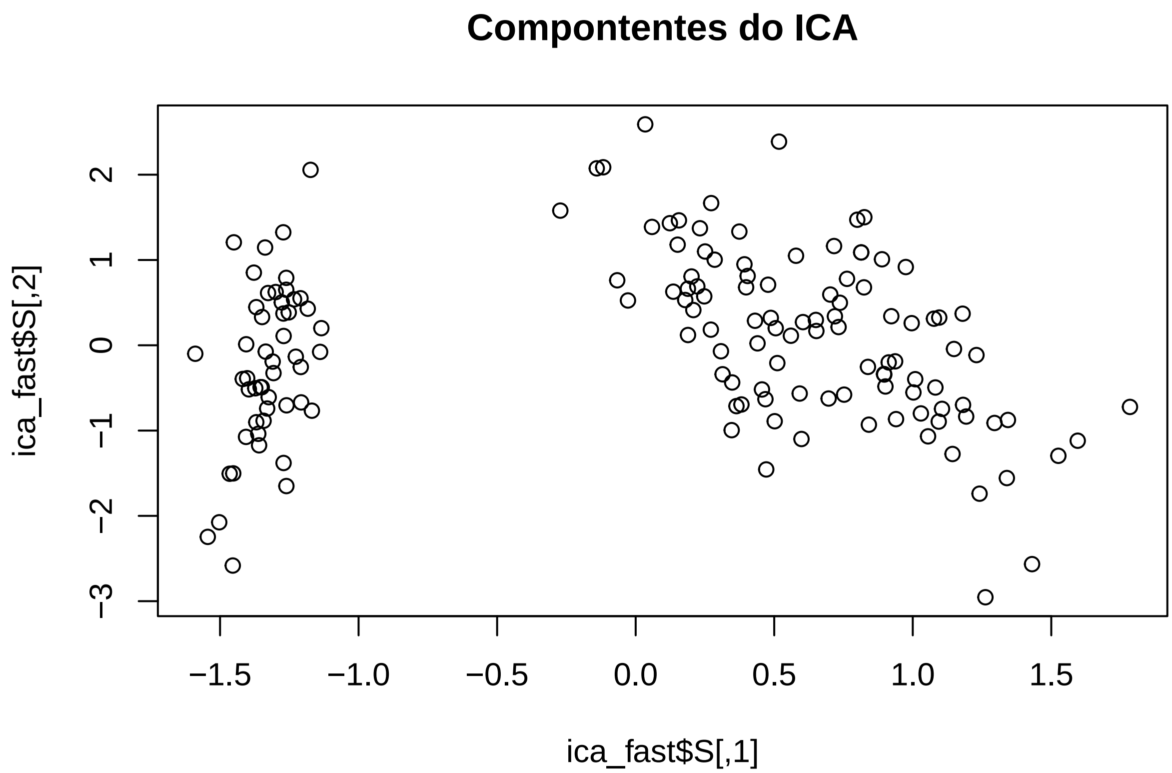
```r
plot(ica_fast$X, main = "Dados pré-processados")
```




```
plot(ica_fast$X %*% ica_fast$K, main = "Componentes do PCA")
```




```
plot(ica_fast$S, main = "Compontentes do ICA")
```



Questão 4 Considere o conjunto de dados *Boston* do pacote *ISLR*, contendo 506 amostras e 14 variáveis. Escolha variáveis que você acha que são importantes para descrever os dados. Faça uma análise de CP e uma análise fatorial e tente interpretar as componentes e os fatores. —————————————————————— Vamos extrair os dados e fazer uma análise preliminar

```
# install.packages("ISLR")
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.2.3
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
data("Boston")
boston <-  Boston
attach(boston)
head(boston)
```

```
##       crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
```

```
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```r
glimpse(boston)
```

```
## Rows: 506
## Columns: 14
## $ crim    <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.02985, 0.08829,~
## $ zn      <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5, 12.5, 12.5, 1~
## $ indus   <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87, 7.87, 7.87, 7.~
## $ chas    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ nox     <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524, 0.524, 0.524,~
## $ rm      <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012, 6.172, 5.631,~
## $ age     <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1, 100.0, 85.9, 9~
## $ dis     <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622, 5.5605, 5.9505~
## $ rad     <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 4,~
## $ tax     <dbl> 296, 242, 242, 222, 222, 222, 311, 311, 311, 311, 311, 311, 31~
## $ ptratio <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2, 15.2, 15.2, 15~
## $ black   <dbl> 396.90, 396.90, 392.83, 394.63, 396.90, 394.12, 395.60, 396.90~
## $ lstat   <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.15, 29.93, 17.10~
## $ medv    <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15~
```

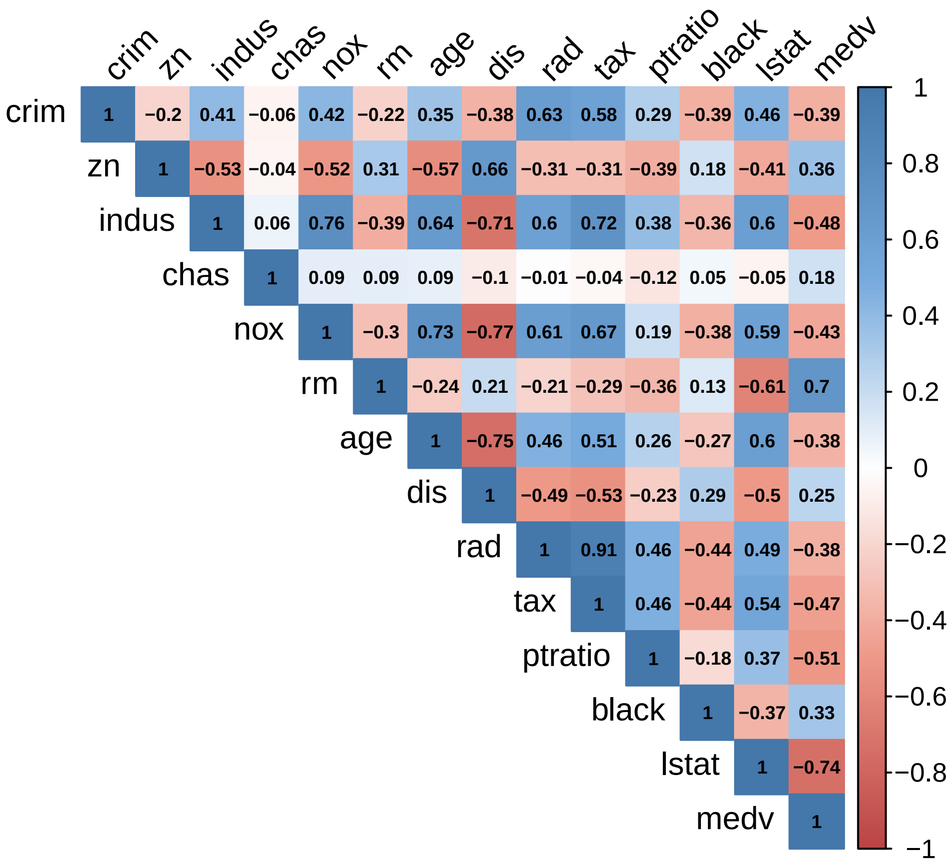
```r
summary(boston)
```

```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
```

```
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

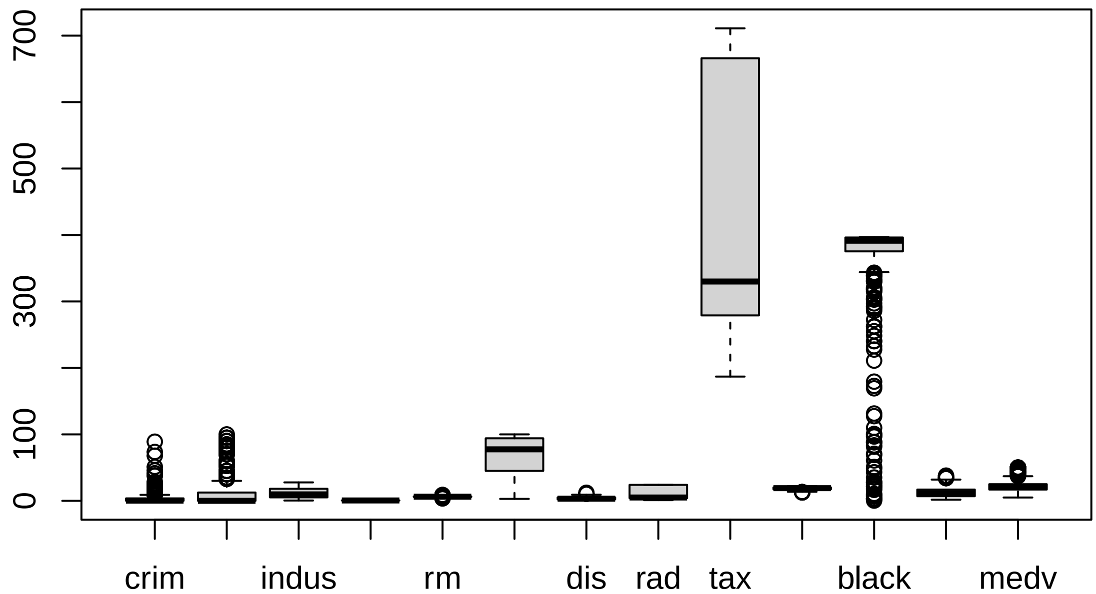
São 14 variáveis no data set: CRIM - taxa de criminalidade per capita por cidade; ZN - proporção de área residencial zoneada para lotes com mais de 25.000 pés quadrados; INDUS - proporção de acres de negócios não comerciais por cidade; CHAS - variável dummy do rio Charles (1 se o terreno faz fronteira com o rio; 0 caso contrário); NOX - concentração de óxidos nítricos (partes por 10 milhões); RM - número médio de quartos por habitação; AGE - proporção de unidades ocupadas por proprietários construídas antes de 1940; DIS - distâncias ponderadas para cinco centros de emprego de Boston; RAD - índice de acessibilidade a rodovias radiais; TAX - taxa de imposto sobre propriedade de valor total por US$ 10.000; PTRATIO - proporção aluno-professor por cidade; B - 1000(Bk - 0.63)^2 onde Bk é a proporção de pessoas negras por cidade; LSTAT - % de status social mais baixo da população MEDV - Valor médio de casas ocupadas pelos proprietários em US$ 1.000. São variáveis relacionadas ao mercado imobiliário: informações relacionads com crime, emprego, acessibilidade, característica dos imóveis estão disponíveis nessa base de dado. Provavelmente é uma base de dado para tentar explicar o preço dos imóveies, colocando medv como variável explicada. Vamos processeguir com a matrix de correlação entre as variáveias analisadas.

```r
# mudanças preliminares na base:
# exclusão do chas (variável binária)
# criando a correlação entre as variáveis
correlacao <- cor(boston , method = "pearson")
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
# produzindo um gráfico para visualização
corrplot(correlacao, method = "color",
type = "upper", col = col(200),
addCoef.col = "black",
tl.col="black", tl.srt=45,
   number.cex=0.60)
```



Como podemos ver no correlograma acima, a variável chas é pouco correlacionada com todas as outras. Ela será excluída da análise e escolheremos todas as outras para realizar o PCA. Pelo correlograma, podemos ver que a maioria das variáveis não estão correlacionadas positivamente ou negativamente com cada uma das outras. Há, no entanto, pares que são correlacionados entre si, como por exemplo, lstat (% de status social mais baixo da população) e medv (Valor médio de casas ocupadas pelos proprietários em US$ 1.000), o que faz sentido. Prosseguindo na análise: vamos avaliar a dispersão dos dados para verificar a necessidade de padronização e normalização.

```
boston <- subset(boston, select = -chas)
boxplot(boston)
```



O gráfico indica a necessidade de padronização. Vamos assim, calcular os componentes principais:

```
acp_boston <- prcomp(
  boston,
  center = TRUE,
  scale. = TRUE
)
acp_boston
```

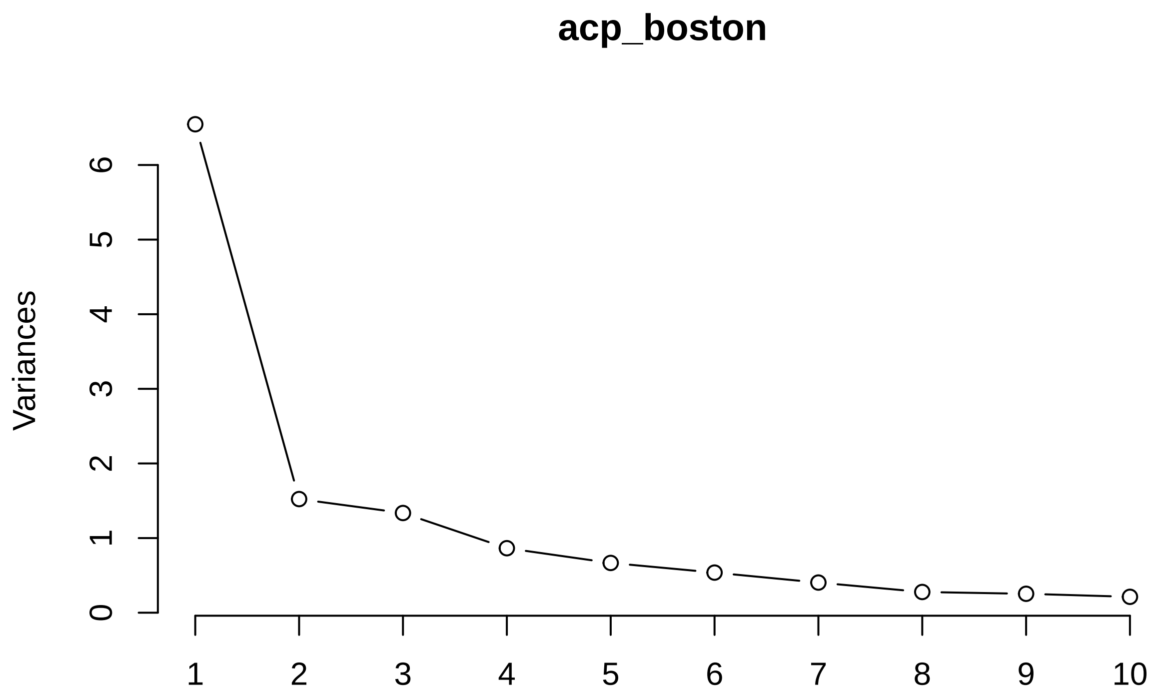
```
## Standard deviations (1, .., p=13):
##  [1] 2.5584859 1.2339618 1.1557640 0.9295180 0.8165486 0.7331145 0.6353263
##  [8] 0.5267862 0.5034334 0.4613693 0.4280941 0.3687517 0.2465631
##
## Rotation (n x k) = (13 x 13):
##                PC1         PC2         PC3         PC4         PC5          PC6
## crim     0.2422405 -0.01172081  0.40869740 -0.06251454  0.21283095 -0.778128729
## zn      -0.2454897 -0.11184069  0.43428231 -0.30142522  0.36118265  0.269786863
## indus    0.3319300  0.11604265 -0.08762068  0.01862031  0.09397546  0.340977490
## nox      0.3252950  0.25893689 -0.09797035 -0.19338669  0.13978344  0.188588552
## rm      -0.2027258  0.53305914  0.24774937  0.18533364 -0.16765587 -0.087194021
## age      0.2970743  0.25039568 -0.25847736 -0.07534470  0.03343268 -0.131023819
## dis     -0.2982844 -0.36832070  0.23985538 -0.02343807  0.02077542  0.115384862
## rad      0.3034153  0.08933238  0.41445957  0.21313034  0.15492882  0.139161884
## tax      0.3240146  0.06021256  0.34093699  0.14423562  0.20437409  0.309458894
## ptratio  0.2075682 -0.32926050  0.06369403  0.70446212 -0.25149625  0.014970256
## black   -0.1966008 -0.03079827 -0.36295651  0.40086101  0.79102571 -0.096447637
```

```
## lstat    0.3113542 -0.24579590 -0.11255495 -0.28849724  0.09599648 -0.084077459
## medv    -0.2664794  0.49289698  0.06993637  0.14317606  0.04756424  0.009466905
##                PC7         PC8         PC9        PC10        PC11
## crim     0.16401539 -0.25463951 -0.07255244 -0.06961718 -0.06618987
## zn      -0.38052605 -0.38763252  0.23476336 -0.13145433  0.22462187
## indus    0.17025070 -0.62192358 -0.26503836  0.27602049 -0.34817987
## nox      0.04219525  0.04951337 -0.21541118 -0.43648899  0.43913283
## rm      -0.44328453  0.01560175 -0.52711170  0.22659818  0.12388143
## age     -0.58897430  0.03715909  0.24765423 -0.32936842 -0.48492143
## dis     -0.12418199  0.17612377 -0.28053322 -0.10607898 -0.50732850
## rad      0.07464434  0.45977882  0.12863057  0.04311139 -0.02057250
## tax      0.06604343  0.18242776 -0.01072251  0.04156297 -0.16417705
## ptratio -0.27636063 -0.28138707  0.16095672 -0.10044686  0.22820595
## black   -0.04964524  0.06595919 -0.14866646  0.03933593  0.04204641
## lstat   -0.35493678  0.16650653  0.08016057  0.68380910  0.18038211
## medv     0.15481149 -0.08417403  0.57792859  0.23929557 -0.09725508
##                PC12         PC13
## crim     0.098582265  0.059219042
## zn      -0.130474904 -0.097971925
## indus    0.077469593 -0.231943485
## nox      0.531297138  0.093607192
## rm      -0.044655197  0.005641857
## age     -0.060955855 -0.036391972
## dis      0.554133636  0.051984305
## rad      0.002262749 -0.635285383
## tax     -0.255441158  0.696509237
## ptratio  0.194888380  0.055630243
## black   -0.021666225 -0.015721508
## lstat    0.250803309  0.085405978
## medv     0.453428439  0.144662506
```

```r
summary(acp_boston)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.5585 1.2340 1.1558 0.92952 0.81655 0.73311 0.63533
## Proportion of Variance 0.5035 0.1171 0.1027 0.06646 0.05129 0.04134 0.03105
## Cumulative Proportion  0.5035 0.6207 0.7234 0.78987 0.84116 0.88250 0.91355
##                            PC8    PC9    PC10   PC11    PC12    PC13
## Standard deviation     0.52679 0.5034 0.46137 0.4281 0.36875 0.24656
## Proportion of Variance 0.02135 0.0195 0.01637 0.0141 0.01046 0.00468
## Cumulative Proportion  0.93490 0.9544 0.97077 0.9849 0.99532 1.00000
```

```r
screeplot(acp_boston, type = "lines")
```



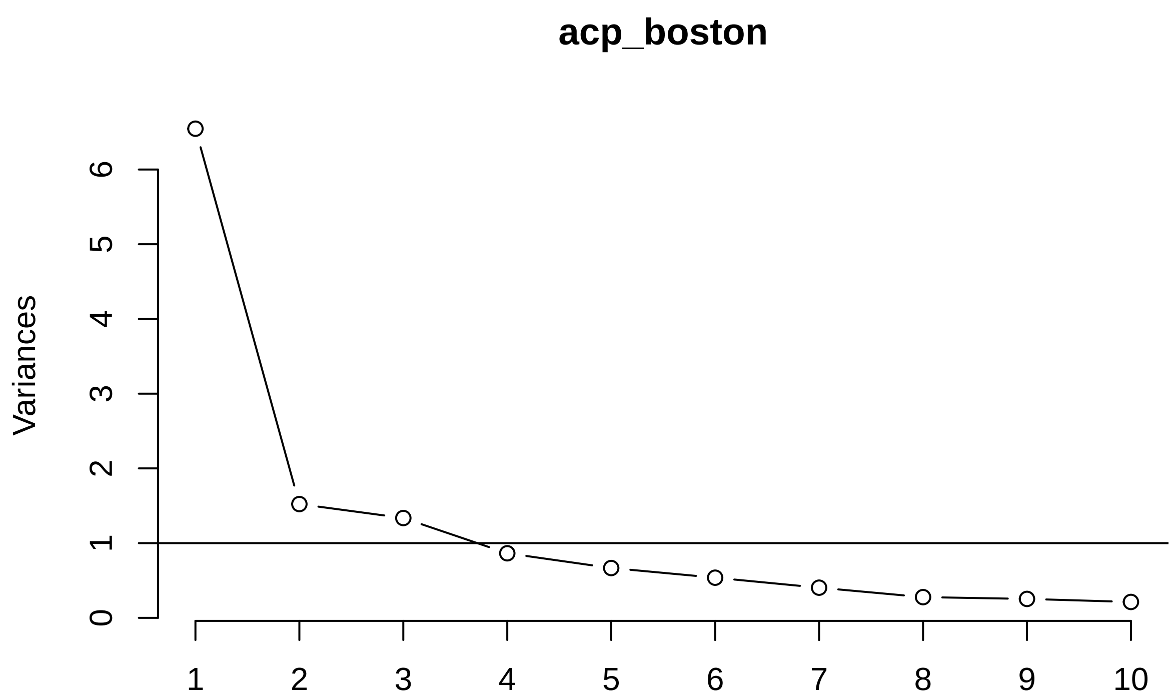
De forma análoga ao exercício 1, os autovalores são obtidos através do cálculo do quadrado dos coeficientes reportados como "standard deviation". Já a proporção da variância explicada por cada componente principal pode ser calculada através da razão entre o autovalor do componente e o somatório dos autovalores de todos os componentes. Combinando os resultados do gráfico dos autovalores com os resultados presentes na tabela anterior, confirma-se que somente os 3 primeiros componentes. Para interpretar esses três componentes, vamos analisara tabela abaixo, com os três componentes e as variáveis:

```
acp_boston$rotation[,1:3]
```

```
##                PC1         PC2         PC3
## crim     0.2422405 -0.01172081  0.40869740
## zn      -0.2454897 -0.11184069  0.43428231
## indus    0.3319300  0.11604265 -0.08762068
## nox      0.3252950  0.25893689 -0.09797035
## rm      -0.2027258  0.53305914  0.24774937
## age      0.2970743  0.25039568 -0.25847736
## dis     -0.2982844 -0.36832070  0.23985538
## rad      0.3034153  0.08933238  0.41445957
## tax      0.3240146  0.06021256  0.34093699
## ptratio  0.2075682 -0.32926050  0.06369403
## black   -0.1966008 -0.03079827 -0.36295651
## lstat    0.3113542 -0.24579590 -0.11255495
## medv    -0.2664794  0.49289698  0.06993637
```

Agora, vamos processguir com a análise fatorial. Quantos são os faotres que precisamos estimar? vamos ustilizar a Regra de Kaiser-Guttman, como na questão 2. O número de fatores adequados é 3, como pode ser anáilsado no Teste scree:

```r
screeplot(acp_boston, type = "lines")
abline(h=1)
```



Vamos prosseguir com a análise fatorial, nas especificações minres e fator principal.

```r
fa_boston_minres =  fa(boston, nfactors = 3, rotate = "varimax")
fa_boston_minres$loadings
```

```
##
## Loadings:
##         MR1    MR3    MR2
## crim     0.191  0.586  0.213
## zn      -0.637        -0.274
## indus    0.641  0.451  0.310
## nox      0.735  0.433  0.186
## rm      -0.158        -0.750
## age      0.763  0.259  0.202
## dis     -0.899 -0.276
## rad      0.257  0.922  0.130
## tax      0.325  0.863  0.219
## ptratio  0.130  0.337  0.419
## black   -0.182 -0.433 -0.161
## lstat    0.423  0.331  0.651
## medv    -0.157 -0.272 -0.890
##
##                  MR1   MR3   MR2
```

```
## SS loadings    3.233 2.971 2.339
## Proportion Var 0.249 0.229 0.180
## Cumulative Var 0.249 0.477 0.657
```

```
fa_boston_minres
```

```
## Factor Analysis using method =  minres
## Call: fa(r = boston, nfactors = 3, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##           MR1   MR3   MR2   h2    u2 com
## crim     0.19  0.59  0.21 0.42 0.575 1.5
## zn      -0.64 -0.09 -0.27 0.49 0.511 1.4
## indus    0.64  0.45  0.31 0.71 0.289 2.3
## nox      0.73  0.43  0.19 0.76 0.238 1.8
## rm      -0.16 -0.07 -0.75 0.59 0.407 1.1
## age      0.76  0.26  0.20 0.69 0.310 1.4
## dis     -0.90 -0.28 -0.05 0.89 0.114 1.2
## rad      0.26  0.92  0.13 0.93 0.067 1.2
## tax      0.32  0.86  0.22 0.90 0.102 1.4
## ptratio  0.13  0.34  0.42 0.31 0.694 2.1
## black   -0.18 -0.43 -0.16 0.25 0.754 1.7
## lstat    0.42  0.33  0.65 0.71 0.287 2.3
## medv    -0.16 -0.27 -0.89 0.89 0.110 1.3
##
##                        MR1  MR3  MR2
## SS loadings           3.23 2.97 2.34
## Proportion Var        0.25 0.23 0.18
## Cumulative Var        0.25 0.48 0.66
## Proportion Explained  0.38 0.35 0.27
## Cumulative Proportion 0.38 0.73 1.00
##
## Mean item complexity =  1.6
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  78  with the objective function =  10.18 with Chi Square =  5090.73
## df of  the model are 42  and the objective function was  0.88
##
## The root mean square of the residuals (RMSR) is  0.04
## The df corrected root mean square of the residuals is  0.05
##
## The harmonic n.obs is  506 with the empirical chi square  111.3  with prob <  3.5e-08
## The total n.obs was  506  with Likelihood Chi Square =  435.83  with prob <  6.1e-67
##
## Tucker Lewis Index of factoring reliability =  0.853
## RMSEA index =  0.136  and the 90 % confidence intervals are  0.125 0.148
## BIC =  174.31
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##                                                    MR1  MR3  MR2
## Correlation of (regression) scores with factors   0.96 0.97 0.95
## Multiple R square of scores with factors          0.92 0.94 0.90
## Minimum correlation of possible factor scores     0.83 0.88 0.80
```

```r
# install.packages("GPArotation")
library(GPArotation)
```

```
## Warning: package 'GPArotation' was built under R version 4.2.3
```

```
##
## Attaching package: 'GPArotation'
```

```
## The following objects are masked from 'package:psych':
##
##     equamax, varimin
```

```r
fa_boston_pa =  fa(boston, nfactors = 3, fm = "pa")
fa_boston_pa
```

```
## Factor Analysis using method =  pa
## Call: fa(r = boston, nfactors = 3, fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##           PA1   PA3   PA2   h2    u2 com
## crim     0.00  0.59 -0.12 0.43 0.575 1.1
## zn      -0.69  0.17  0.20 0.49 0.511 1.3
## indus    0.57  0.25 -0.18 0.71 0.289 1.6
## nox      0.71  0.22 -0.03 0.76 0.238 1.2
## rm      -0.03  0.13  0.81 0.60 0.404 1.1
## age      0.80  0.00 -0.07 0.69 0.309 1.0
## dis     -0.98 -0.01 -0.12 0.89 0.114 1.0
## rad     -0.01  0.99  0.04 0.93 0.069 1.0
## tax      0.07  0.88 -0.06 0.90 0.101 1.0
## ptratio -0.03  0.27 -0.40 0.31 0.693 1.8
## black   -0.05 -0.42  0.08 0.25 0.753 1.1
## lstat    0.28  0.11 -0.61 0.71 0.286 1.5
## medv     0.06 -0.09  0.93 0.89 0.115 1.0
##
##                        PA1  PA3  PA2
## SS loadings           3.30 2.81 2.43
## Proportion Var        0.25 0.22 0.19
## Cumulative Var        0.25 0.47 0.66
## Proportion Explained  0.39 0.33 0.28
## Cumulative Proportion 0.39 0.72 1.00
##
##  With factor correlations of
##       PA1   PA3   PA2
## PA1  1.00  0.58 -0.42
## PA3  0.58  1.00 -0.43
## PA2 -0.42 -0.43  1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  78  with the objective function =  10.18 with Chi Square =  5090.73
## df of  the model are 42  and the objective function was  0.88
##
```

```
## The root mean square of the residuals (RMSR) is  0.04
## The df corrected root mean square of the residuals is  0.05
##
## The harmonic n.obs is  506 with the empirical chi square  111.3  with prob <  3.5e-08
## The total n.obs was  506  with Likelihood Chi Square =  435.81  with prob <  6.1e-67
##
## Tucker Lewis Index of factoring reliability =  0.854
## RMSEA index =  0.136  and the 90 % confidence intervals are  0.125 0.148
## BIC =  174.29
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##                                                    PA1  PA3  PA2
## Correlation of (regression) scores with factors   0.97 0.98 0.96
## Multiple R square of scores with factors          0.94 0.97 0.92
## Minimum correlation of possible factor scores     0.89 0.93 0.85
```

```
fa_boston_pa$loadings
```

```
##
## Loadings:
##         PA1    PA3    PA2
## crim            0.591 -0.116
## zn      -0.686  0.172  0.199
## indus    0.569  0.252 -0.180
## nox      0.711  0.223
## rm              0.126  0.808
## age      0.798
## dis     -0.983        -0.120
## rad             0.987
## tax             0.875
## ptratio         0.272 -0.396
## black          -0.421
## lstat    0.282  0.113 -0.614
## medv                   0.926
##
##                  PA1   PA3   PA2
## SS loadings    2.995 2.519 2.164
## Proportion Var 0.230 0.194 0.166
## Cumulative Var 0.230 0.424 0.591
```

O primeiro componente principal pode ser intepretada como. O segundo componente principa, como. E o tericeiro componente como: