Introdução à Análise Multivariada usando R

Marcio Nicolau

2016-10-25

Contents

Chapter 1

Pré-requisitos

Antes de iniciar o curso, faz-se nescessário instalar as ferramentas e pacotes para uso.

Será utilizado o sofware *R version 3.3.1 (2016-06-21)* disponível em CRAN e o *RStudio v1.0.44 Preview* disponível em RStudio.

A carga horária total será de 20 horas (3 dias) nos quais os seguintes tópicos de Análise Multivariada serão abordados.

- Análise de Agrupamentos (CA)
- Análise de Componentes Principais (PCA)
- Análise Fatorial (FA)

Alguns exemplos serão realizados com dados disponíveis na literatura científica e, em certos momentos, serão utilizados dados dos próprios participantes.

Espera-se que ao final do curso o aluno seja capaz de entender o uso de cada técnica e aplicá-la de forma correta em seu campo/área de pesquisa.

Cabe lembrar que este é um curso introdutório e que de forma alguma os conteúdos e aplicações serão apresentadas de forma exaustiva.

Marcio Nicolau

Estatístico / Embrapa Trigo

Palmas/TO, Outubro de 2016

Chapter 2

Introdução

As técnicas de Análise Multivariadas oferecem aplicações em diversas áreas do conhecimento no desenvolvimento científico.

Geralmente são utilizadas em fase exploratória de análise de dados, onde se busca entender melhor as relações entre as variáveis (medições físicas ou observações) de certo evento sob estudo ou de interesse científico.

Há também aplicações em conjunto com outros métodos da estatística onde é possível obter validações ou testes de carácter conclusivo.

Durante este curso e, certamente limitados pelo tempo, serão abordados somente as técnicas de caractere exploratório com a finalidade de melhor explicar as relações intrínsecas entre os dados, reduzir a dimensão, entender fontes de variabilidade, criar grupos homogêneos de indivíduos/espécies.

Pode-se dizer que a Análise Fatorial (FA), a Análise de Componentes Principais (PCA) e a Análise de Cluster (CA) são processo que tem por objetivo reduzir a complexidade dos dados observados, bem como entender o modelo estrutural presente nos dados.

No caso do FA, o objetivo é o de identificar construções poucos constructos para explicar os dados observados. No caso de PCA, pode não ser simples redução de dimensão, mas a interpretação dos componentes.

Por fim, a Análise de Cluster (CA) pode também ser usada para criar grupos de variáveis com interesse de reduzir a complexidade dos dados por meio da formação de grupos menores e homogêneos.

Tecnicamente, o problema de redução de dados pode ser resolvido como uma decomposição do valor singular (SVD) da matriz original, embora a solução mais típica seja o uso de PCA nas matrizes de covariância e/ou correlação.

2.1 Pacotes

Durante o curso serão utilizados alguns pacotes extras que necessitam ser instalados, para facilitar o processo de instalar e habilitá-los para uso, utilize a função abaixo

```r
autoLib = function(packs) {
  idx = packs %in% installed.packages()[, "Package"]
  needInstall = packs[!idx]
  if(length(needInstall) > 0) install.packages(needInstall, dep = TRUE)
  sapply(packs, require, character.only = TRUE)
}
```

Chapter 3

Análise de Agrupamentos

Nesta seção serão utilizados as seguinte bibliotecas do R.

```
autoLib(c('cluster', 'dplyr'))
```

```
## Loading required package: cluster
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## cluster   dplyr
##    TRUE    TRUE
```

```
knitr::opts_knit$set(fig.width=5, fig.height=5, fig.align='center')
```

3.1 Processo de Agrupamento

Um agrupamento pode ser construído de duas formas:

- hierarquia: funções *agnes*, *diana*, *mona* e *hclust*;
- particionamento: funções *pam*, *clara*, *fanny* e *kmeans*

3.2 Métodos de Agrupamento

Um agrupamento pode gerar os grupos utilizando algum dos métodos a seguir (mais comuns):

- average: *média* ou UPGMA (média dissimilaridade)
- single: *simple:* (vizinho mais próximo)
- complete: *completa* (vizinho mais distante)
- ward: *Ward* ou método da mínima variância.
- weighted/mcquitty: *média ponderada* ou WPGMA

3.3 Distâncias usadas para cálculo de agrupamentos

Para se calcular a distância entre os componentes, pode-se utilizar as funções a seguir (mais comuns):

- euclidian: *euclideana*, raiz da soma do quadrado das diferenças entre os pontos/observações (distância no plano cartesiano)
- mahalanobis: *Mahalanobis*, distância de cada valor em relação à média e covariância (também conhecida como distância estatística). *OBS* é capaz de trabalhar a distância para observações com repetições. (kernel da normal multivariada)
- manhattan: *Manhattan*, soma das diferenças média absoluta (L1 norm)
- maximum: *Máxima*, máxima distância entre dois componentes (supremum norm)
- canberra: *Canberra*, uso em valores não negativos (p.ex. contagem) $\sum(|x_i - y_i|/|x_i + y_i|)$
- binary: *Binária*, para valores do tipo "on"/"off" em que 0 representa desligado e números maiores que 0. A distância é a proporção de "on's".
- minkowski: *Minkowski*, P-Norm ou p-ésima raíz da soma de potência p das diferenças.

3.4 Exemplo: Força de trabalho agrícola na UE (1993)

Estes conjunto registra os dados da produção per capita e o percentual da população que trabalha na agricultura em cada país da UE em 1993.

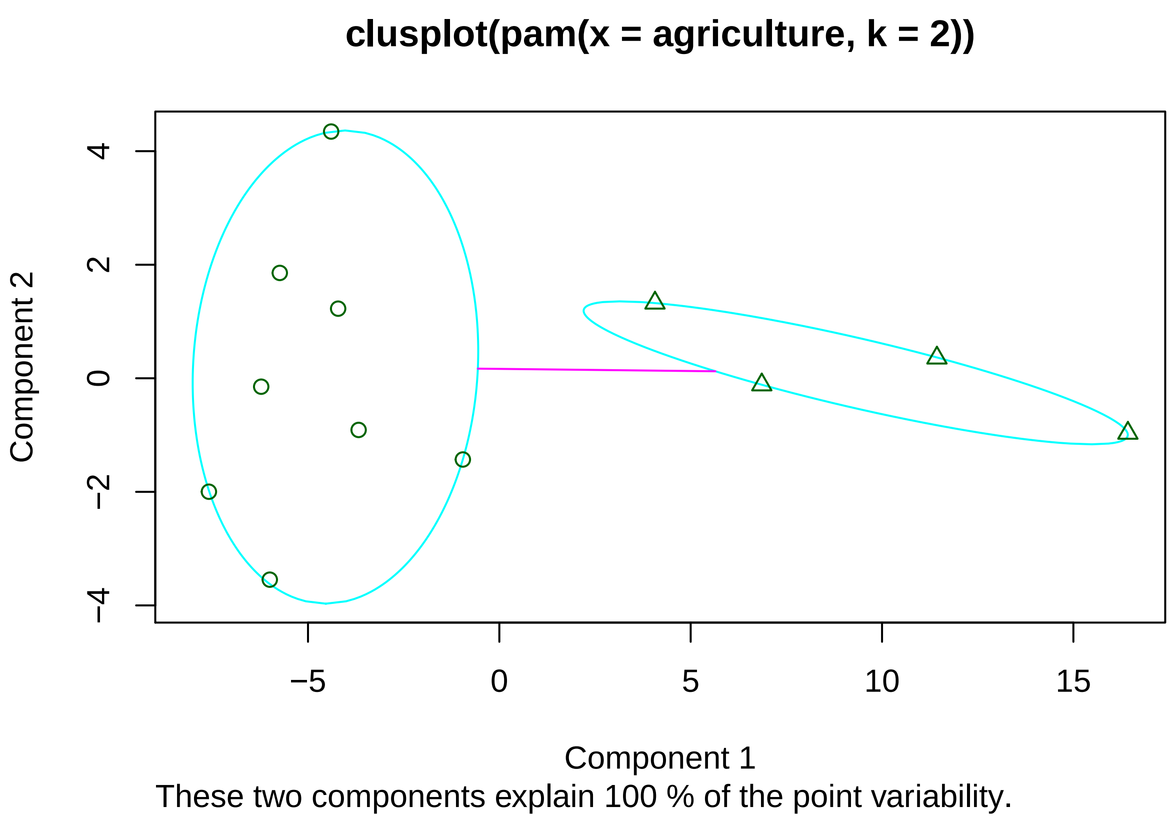
```
data(agriculture)

## Calcula matriz de dissimilaridade usando distância euclidiana
## e sem padronização das variáveis
print(daisy(agriculture, metric = "euclidean", stand = FALSE),
      digits = 2)
```

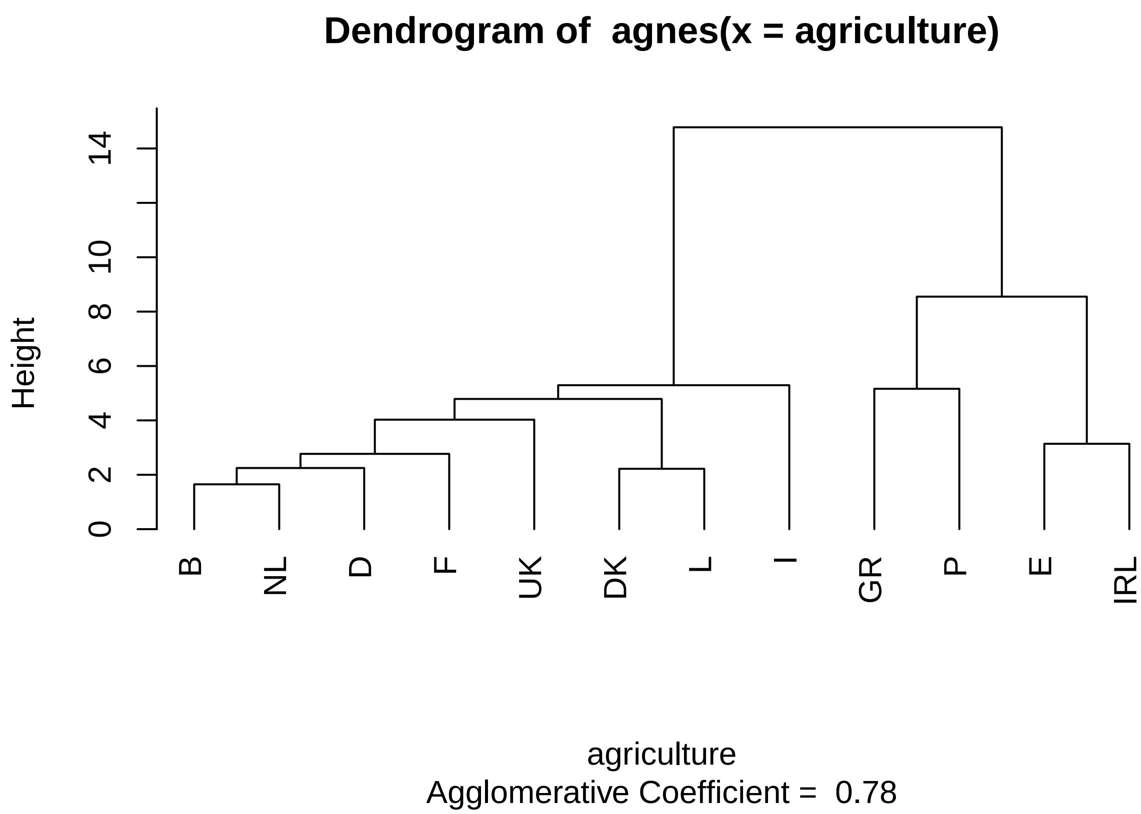
```
## Dissimilarities :
##        B   DK    D   GR    E    F  IRL    I    L   NL    P
## DK   5.4
## D    2.1  3.4
## GR  22.3 22.6 22.7
## E    9.8 11.2 10.4 12.6
## F    3.4  3.5  2.7 20.1  8.1
## IRL 12.7 13.3 13.1  9.6  3.1 10.6
## I    5.8  5.5  5.4 17.4  5.7  2.8  7.9
## L    4.3  2.2  2.3 24.0 12.1  4.1 14.6  6.7
## NL   1.6  5.1  2.4 20.8  8.3  2.2 11.2  4.2  4.7
```

```
## P   17.2 17.9 17.7  5.2  7.4 15.2  4.6 12.5 19.2 15.7
## UK   2.8  8.1  4.9 21.5  9.0  5.3 12.1  6.7  7.1  3.1 16.3
##
## Metric :  euclidean
## Number of objects : 12
```

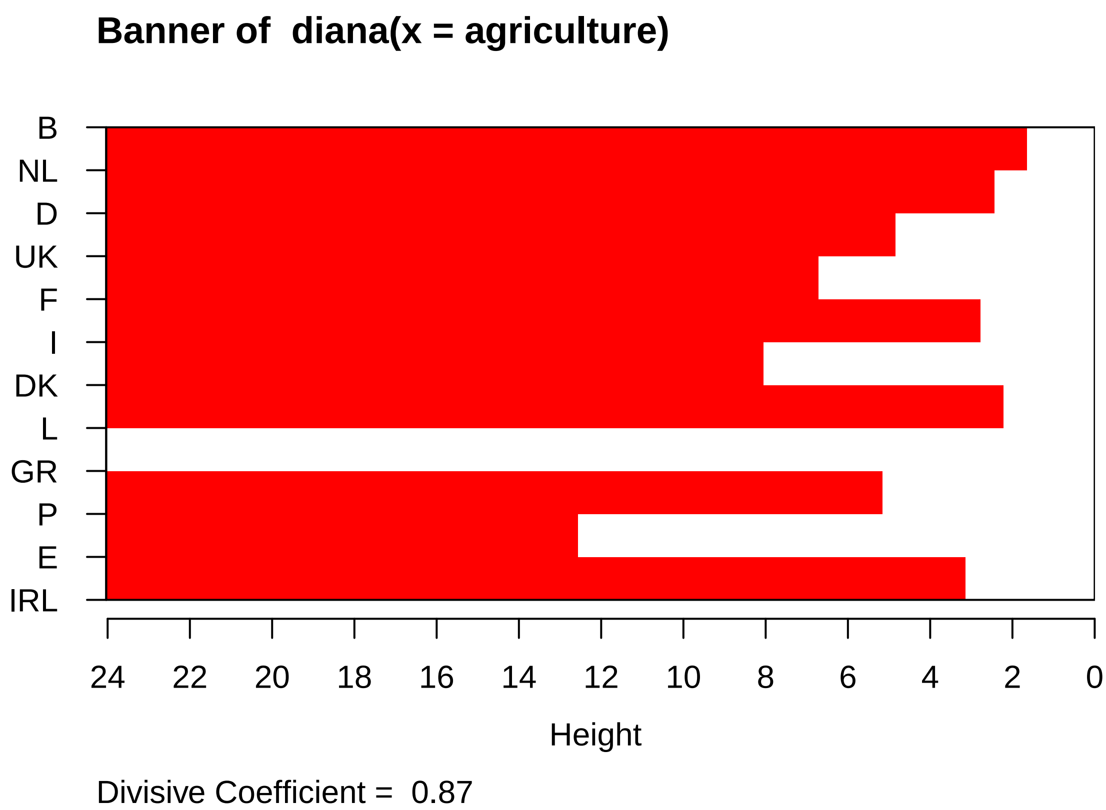
```
## Usa método de particionamento pelo meióide
## Partitioning Around Medoids (PAM)
plot(pam(agriculture, 2), which.plots = 1)
```



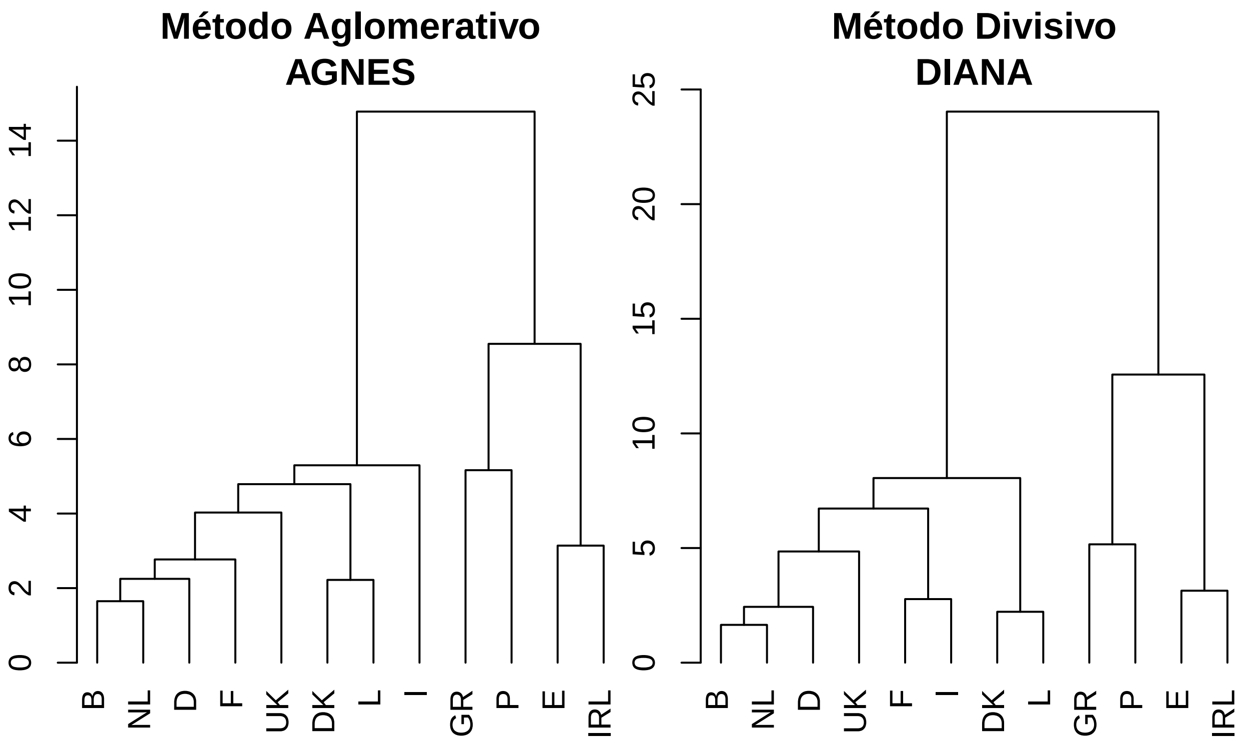

```
## Gráfico dendograma usando método aglomeração mais próximo
## agnes
plot(agnes(agriculture), which.plots = 2, hang = -1)
```




```r
## Plot dissimilaridade usando método divisivo
## diana
plot(diana(agriculture), which.plots = 1)
```




```r
## Usando agnes e diana para conjunto agricultura
par(mfrow=c(1,2), mar=c(3,2,2,0))
plot(agnes(agriculture), which.plots = 2, hang = -1,
     main = "Método Aglomerativo\nAGNES")
plot(diana(agriculture), which.plots = 2, hang = -1,
     main = "Método Divisivo\nDIANA")
```

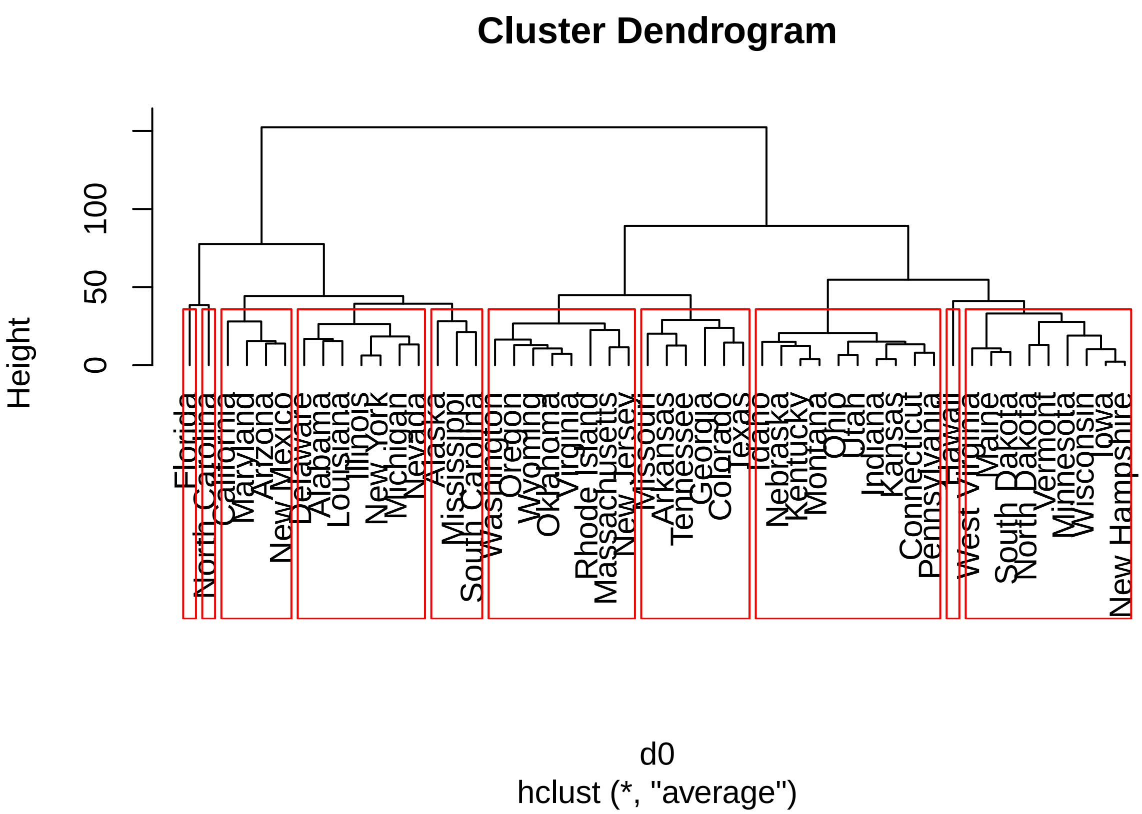


3.5 Exemplo cluster USArrest

Neste exemplo vamos utilizar o conjunto USArrest, disponível na instalação padrão do R.

```r
data(USArrests)
#?USArrests

d0 = dist(USArrests) # euclidian
hc = hclust(d0, "average")
plot(hc, hang = -1)
# Criar 10 grupos
memb = cutree(hc, k = 10)
# Anota no gráfico os 10 grupos
rect.hclust(hc, 10)
```



Como exercício, será trabalhado a seguinte situação: Gostaríamos de saber quais são os estados que compoem o grupo 1 e qual a média de assalto (variável *Assault*) para cada grupo.

```
# Cria conjunto de dados a partir da divisão inicial com
# 10 grupos, com as variáveis State e grp10
dAssault = data.frame(State = names(memb), grp10 = memb)
# Utilizando a biblioteca dplyr, transformamos o conjunto
# dAssault para criar uma nova variável grp3
dAssault = dAssault %>% mutate(grp3 = cutree(hc, k = 3), Assault = USArrests$Assault)
# calculamos a média por cada grupo grp10
dAssault %>% group_by(grp10) %>% summarize(media=mean(Assault))
```

```
## # A tibble: 10 × 2
##    grp10     media
##    <int>     <dbl>
## 1      1 247.57143
## 2      2 267.00000
## 3      3 288.75000
## 4      4 195.33333
## 5      5 112.40000
## 6      6 335.00000
## 7      7  46.00000
## 8      8  64.55556
## 9      9 156.75000
## 10    10 337.00000
```

```
# selecionamos os estados que estão associados ao grupo 1
dAssault %>% filter(grp10 == 1)
```

```
##       State grp10 grp3 Assault
## 1   Alabama     1    1     236
## 2  Delaware     1    1     238
## 3  Illinois     1    1     249
## 4 Louisiana     1    1     249
## 5  Michigan     1    1     255
## 6    Nevada     1    1     252
## 7  New York     1    1     254
```

```
# calculamos a média e o número de estados por cada grupo grp3
dAssault %>%
  group_by(grp3) %>%
  summarize(media=mean(Assault), prop = sum(Assault)/sum(USArrests$Assault),
            qtd = n()) %>% arrange(prop)
```

```
## # A tibble: 3 × 4
##    grp3    media      prop   qtd
##   <int>    <dbl>     <dbl> <int>
## 1     3  87.5500 0.2050832    20
## 2     2 173.2857 0.2841415    14
## 3     1 272.5625 0.5107754    16
```

Pode-se usa a ANOVA para avaliar o número de grupos, confira o código a seguir

```
# ANOVA para grp10
aovGrp10 = aov(Assault ~ as.factor(grp10), data = dAssault)
summary(aovGrp10)
```

```
##                  Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(grp10)  9 335715   37302   324.5 <2e-16 ***
## Residuals        40   4598     115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ANOVA para grp3
aovGrp3 = aov(Assault ~ as.factor(grp3), data = dAssault)
summary(aovGrp3)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(grp3)  2 304387  152194   199.1 <2e-16 ***
## Residuals       47  35926     764
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# OBS: todo modelo linear aumenta a explicação com a
# adição de novas variáveis ao modelo, faz-se necessário
# criar uma penalização para auxiliar na escolha entre modelos
# além de definir um critério
(mseGrp10 = sqrt(summary(aovGrp10)[[1]]$`Mean Sq`[2]))
```

```
## [1] 10.72138
```

```r
(mseGrp3 = sqrt(summary(aovGrp3)[[1]]$`Mean Sq`[2]))
```

```
## [1] 27.64738
```

```r
# critério mse - log(num fatores)
mseGrp10 - log(10)
```

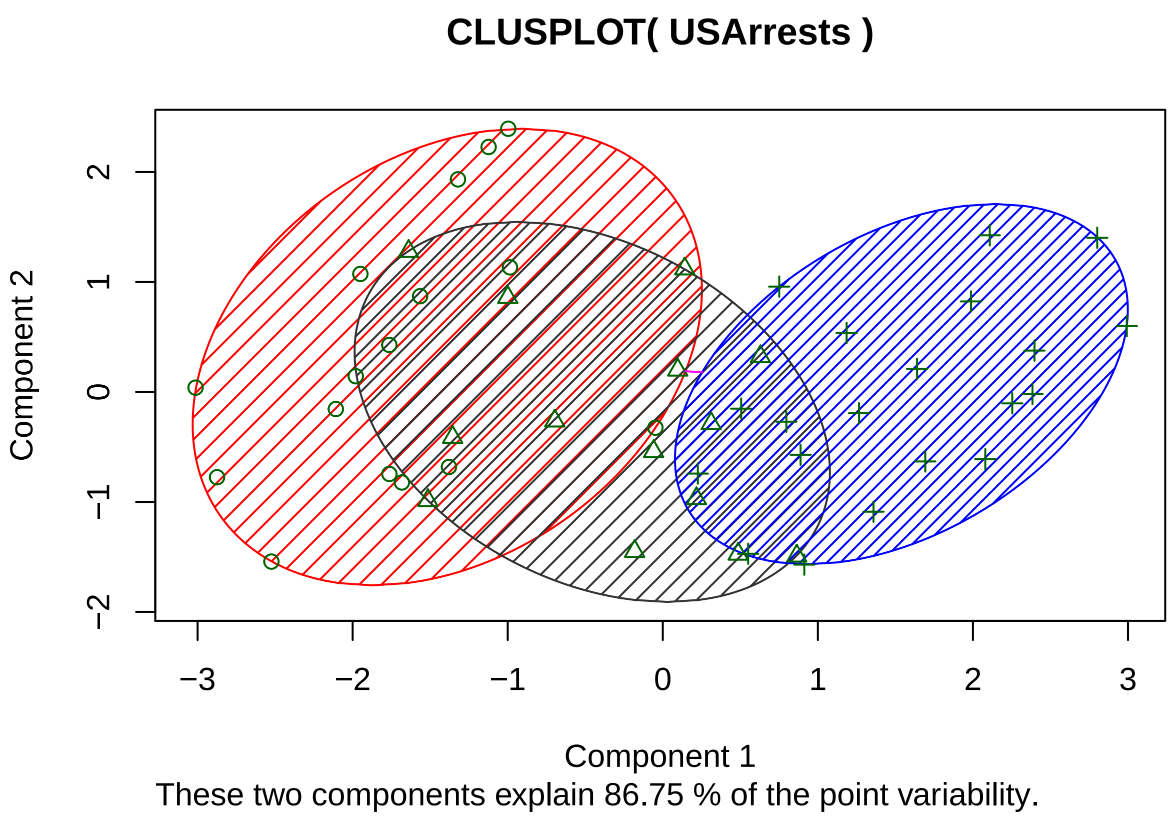
```
## [1] 8.418795
```

```r
mseGrp3 - log(3)
```

```
## [1] 26.54877
```

Por fim, a apresentação dos grupos, também pode ser feita com a função *clusplot*

```r
clusplot(USArrests, cutree(hc, k = 3), shade = TRUE, color = TRUE,
         col.clus = c("red", "grey20", "blue"))
```



Chapter 4

Análise de Componentes Principais

É uma alternativa à Análise Fatorial (FA), apesar dos objetivos serem semelhantes (PCA e FA), na PCA se busca obter o modelo descritivo dos dados enquanto na FA se busca o modelo estrutural.

Outro destaque importante é que a matriz/vetor de cargas *"loadings"* possuírem valores equivalentes, na FA estes são menores. Isto ocorre porque na PCA é ajustado um modelo para a variância completa da matriz de correlação das variáveis e na FA o processo é realizado somente para a variância comum.

Nesta seção serão utilizados as seguinte bibliotecas do R.

```r
autoLib('psych')
```

```
## Loading required package: psych
```

```
## psych
##  TRUE
```

```r
knitr::opts_knit$set(fig.width=5, fig.height=5, fig.align='center')
```

4.1 Métodos de cálculo

O cálculo de componentes principais pode ser feito, em geral, de duas formas:

- *princomp* ou *principal(psych)*: Autovalor/Autovetor
- *prcomp*: decomposição SVD

A documentação do *R* indica o uso da função prcomp por apresentar melhor precisão numérica (por causa do método). Já a função *principal(psych)* apresenta resultados mais detalhados.

De forma geral, o que ocorre é a construção de novas variáveis (fatores) que são a combinação linear entre as variáveis originais mas apresentam certas qualidades de interesse, como por exemplo a independência entre os novos fatores e que estes representam a maior variação possível.

Pode-se dizer que as novas variáveis ou fatores, formam uma rotação (ou projeção) em um novo conjunto de eixos. As direções são escolhidas com base na máxima variabilidade observada.

No final do processo, tem-se variáveis independentes entre si e que explicam uma porção da variação inicial dos dados. Geralmente, 2 ou 3 componentes são suficientes para exlicar mais que 80% da variação total.

4.2 Exemplo da função prcomp

```r
## Variáveis na escala original, inapropriado
prcomp(USArrests)
```

```
## Standard deviations:
## [1] 83.732400 14.212402  6.489426  2.482790
##
## Rotation:
##                 PC1         PC2         PC3         PC4
## Murder   0.04170432 -0.04482166  0.07989066 -0.99492173
## Assault  0.99522128 -0.05876003 -0.06756974  0.03893830
## UrbanPop 0.04633575  0.97685748 -0.20054629 -0.05816914
## Rape     0.07515550  0.20071807  0.97408059  0.07232502
```

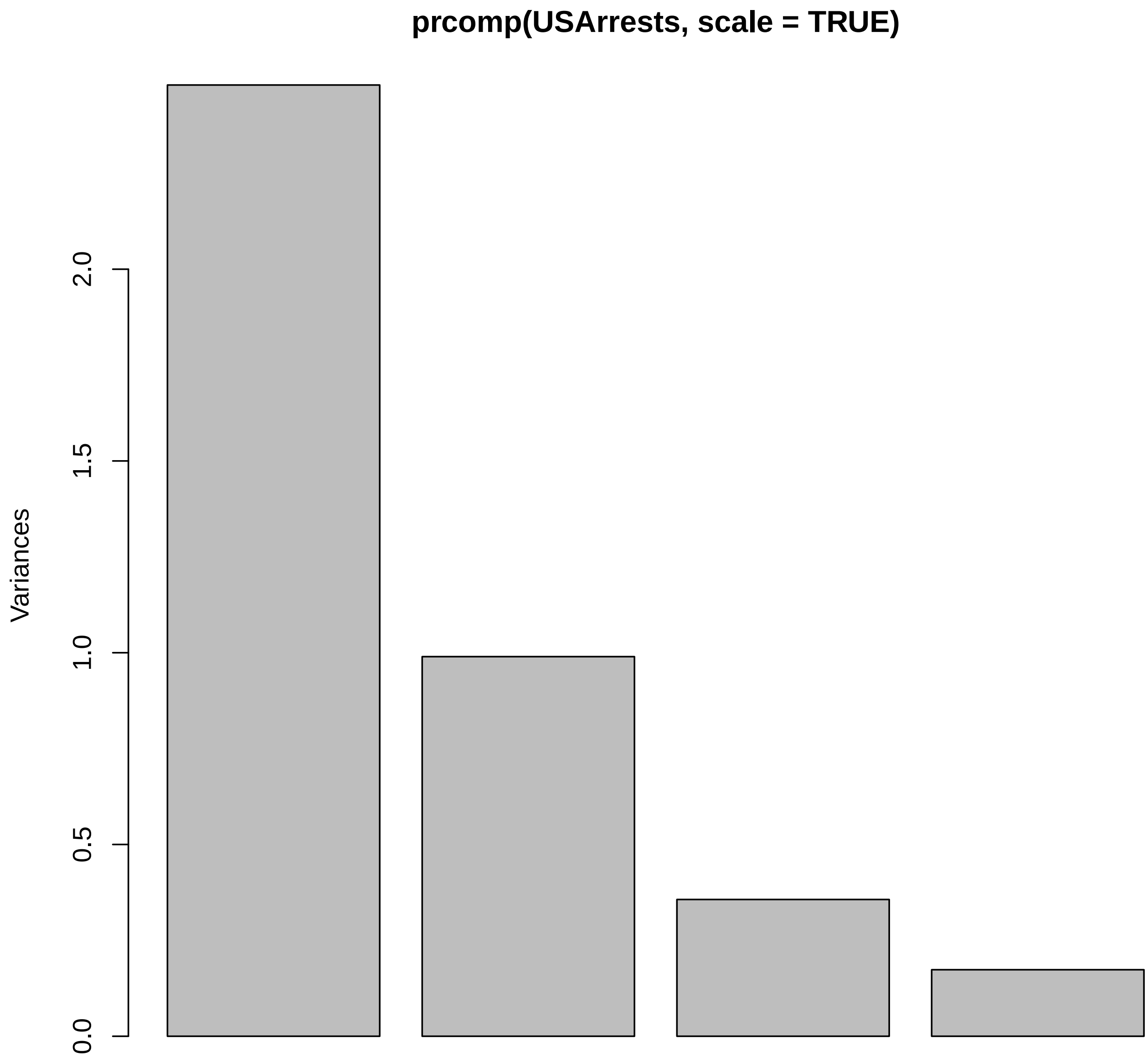
```r
## Variáveis trasnformadas para eliminar efeito de diferença
## entre escala/medida
prcomp(USArrests, scale = TRUE)
```

```
## Standard deviations:
## [1] 1.5748783 0.9948694 0.5971291 0.4164494
##
## Rotation:
##                 PC1        PC2        PC3         PC4
## Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
## Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
## UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
## Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

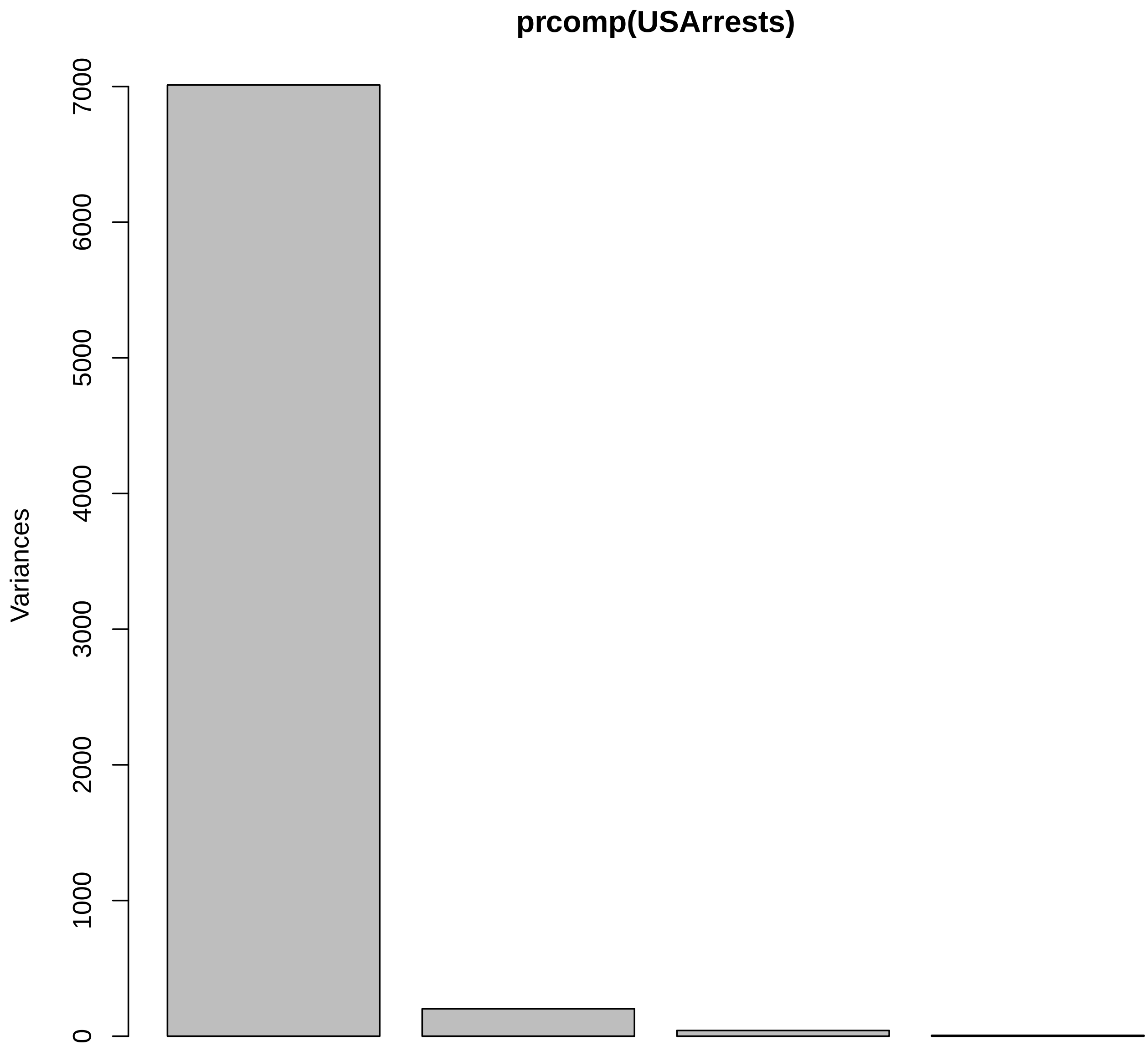
```r
## é possível escolher apenas algumas variávies de interesse
prcomp(~ Murder + Assault + Rape, data = USArrests, scale = TRUE)
```

```
## Standard deviations:
## [1] 1.5357670 0.6767949 0.4282154
##
## Rotation:
##                PC1        PC2        PC3
## Murder  -0.5826006  0.5339532 -0.6127565
## Assault -0.6079818  0.2140236  0.7645600
## Rape    -0.5393836 -0.8179779 -0.1999436
```

```r
## gráfico para escolher o número de componentes
screeplot(prcomp(USArrests, scale = TRUE))
```



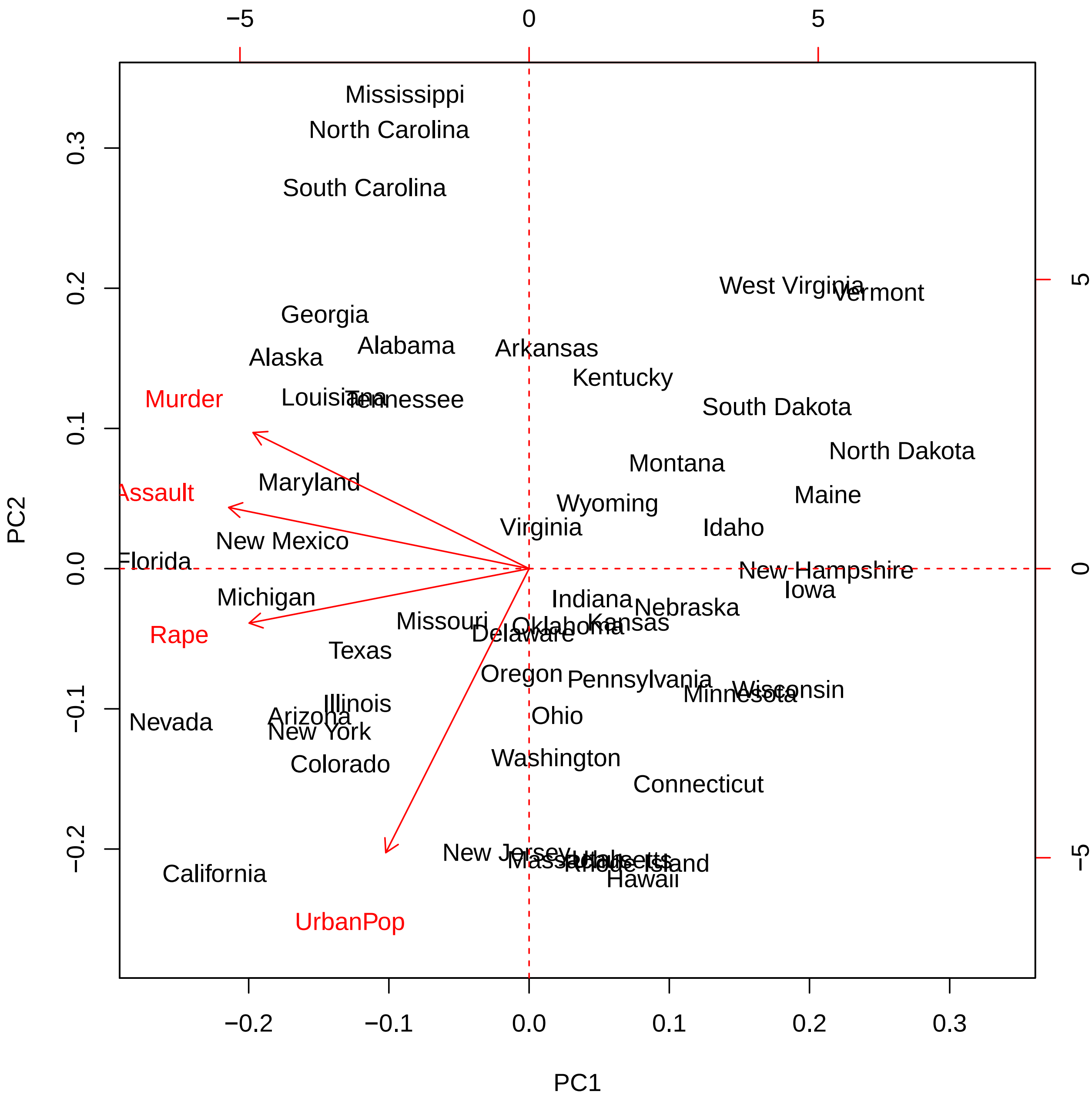

```
## outro gráfico com a representação da variância explicada por
## cada fator
plot(prcomp(USArrests))
```




```r
## sumário com os valores explicados e variância total acumulada
summary(prcomp(USArrests, scale = TRUE))
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.5749 0.9949 0.59713 0.41645
## Proportion of Variance 0.6201 0.2474 0.08914 0.04336
## Cumulative Proportion  0.6201 0.8675 0.95664 1.00000
```

```r
## gráfico biplot
biplot(prcomp(USArrests, scale = TRUE))
abline(v=0,h=0,col="red", lty=2)
```



4.3 Exemplo da função principal (psych)

```r
#Four principal components of the Harman 24 variable problem
#compare to a four factor principal axes solution using factor.congruence
# Calcula o PCA com rotação varimax
pc0 <- principal(Harman74.cor$cov,4,rotate="varimax")
# Calcula o PCA sem rotação (similar prcomp)
pc1 <- principal(Harman74.cor$cov,4,rotate="none")
# resultado do pc0
print(pc0)
```

```
## Principal Components Analysis
## Call: principal(r = Harman74.cor$cov, nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                          RC1   RC3   RC2  RC4   h2   u2 com
## VisualPerception        0.16  0.71  0.23 0.14 0.60 0.40 1.4
## Cubes                   0.09  0.59  0.08 0.03 0.37 0.63 1.1
## PaperFormBoard          0.14  0.66 -0.04 0.11 0.47 0.53 1.2
## Flags                   0.25  0.62  0.09 0.03 0.45 0.55 1.4
## GeneralInformation      0.79  0.15  0.22 0.11 0.70 0.30 1.3
## PargraphComprehension   0.81  0.18  0.07 0.21 0.73 0.27 1.2
## SentenceCompletion      0.85  0.15  0.15 0.06 0.77 0.23 1.1
## WordClassification      0.64  0.31  0.24 0.11 0.57 0.43 1.8
## WordMeaning             0.84  0.16  0.06 0.19 0.78 0.22 1.2
## Addition                0.18 -0.13  0.83 0.12 0.76 0.24 1.2
## Code                    0.18  0.05  0.63 0.37 0.57 0.43 1.8
## CountingDots            0.02  0.17  0.80 0.05 0.67 0.33 1.1
## StraightCurvedCapitals  0.18  0.41  0.62 0.03 0.59 0.41 1.9
## WordRecognition         0.23 -0.01  0.06 0.68 0.52 0.48 1.2
## NumberRecognition       0.12  0.08  0.05 0.67 0.48 0.52 1.1
## FigureRecognition       0.06  0.46  0.05 0.58 0.55 0.45 1.9
## ObjectNumber            0.14  0.01  0.24 0.68 0.54 0.46 1.4
## NumberFigure           -0.02  0.32  0.40 0.50 0.51 0.49 2.7
## FigureWord              0.14  0.25  0.20 0.42 0.30 0.70 2.4
## Deduction               0.43  0.43  0.09 0.30 0.47 0.53 2.8
## NumericalPuzzles        0.18  0.42  0.50 0.17 0.49 0.51 2.5
## ProblemReasoning        0.42  0.41  0.13 0.29 0.45 0.55 3.0
## SeriesCompletion        0.42  0.52  0.25 0.20 0.55 0.45 2.7
## ArithmeticProblems      0.40  0.14  0.55 0.26 0.55 0.45 2.5
##
##                        RC1  RC3  RC2  RC4
## SS loadings           4.16 3.31 3.22 2.74
## Proportion Var        0.17 0.14 0.13 0.11
## Cumulative Var        0.17 0.31 0.45 0.56
## Proportion Explained  0.31 0.25 0.24 0.20
## Cumulative Proportion 0.31 0.56 0.80 1.00
##
## Mean item complexity =  1.7
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.06
##
## Fit based upon off diagonal values = 0.97
```

```r
# resultado do pc1
print(pc1)
```

```
## Principal Components Analysis
## Call: principal(r = Harman74.cor$cov, nfactors = 4, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                         PC1   PC2   PC3   PC4   h2   u2 com
## VisualPerception       0.62 -0.01  0.43 -0.20 0.60 0.40 2.0
## Cubes                  0.40 -0.08  0.40 -0.20 0.37 0.63 2.5
## PaperFormBoard         0.44 -0.19  0.48 -0.11 0.47 0.53 2.4
## Flags                  0.51 -0.18  0.33 -0.22 0.45 0.55 2.4
```

```
## GeneralInformation     0.69 -0.32 -0.34 -0.05 0.70 0.30 1.9
## PargraphComprehension  0.69 -0.42 -0.27  0.08 0.73 0.27 2.0
## SentenceCompletion     0.68 -0.42 -0.36 -0.07 0.77 0.23 2.3
## WordClassification     0.69 -0.24 -0.14 -0.12 0.57 0.43 1.4
## WordMeaning            0.69 -0.45 -0.29  0.08 0.78 0.22 2.1
## Addition               0.47  0.54 -0.45 -0.20 0.76 0.24 3.2
## Code                   0.58  0.43 -0.21  0.03 0.57 0.43 2.2
## CountingDots           0.48  0.55 -0.13 -0.34 0.67 0.33 2.8
## StraightCurvedCapitals 0.62  0.28  0.04 -0.37 0.59 0.41 2.1
## WordRecognition        0.45  0.09 -0.06  0.56 0.52 0.48 2.0
## NumberRecognition      0.42  0.14  0.08  0.53 0.48 0.52 2.1
## FigureRecognition      0.53  0.09  0.39  0.33 0.55 0.45 2.6
## ObjectNumber           0.49  0.28 -0.05  0.47 0.54 0.46 2.6
## NumberFigure           0.54  0.39  0.20  0.15 0.51 0.49 2.3
## FigureWord             0.48  0.14  0.12  0.19 0.30 0.70 1.7
## Deduction              0.64 -0.19  0.13  0.07 0.47 0.53 1.3
## NumericalPuzzles       0.62  0.23  0.10 -0.20 0.49 0.51 1.6
## ProblemReasoning       0.64 -0.15  0.11  0.06 0.45 0.55 1.2
## SeriesCompletion       0.71 -0.10  0.15 -0.10 0.55 0.45 1.2
## ArithmeticProblems     0.67  0.20 -0.23 -0.06 0.55 0.45 1.4
##
##                        PC1  PC2  PC3  PC4
## SS loadings           8.14 2.10 1.69 1.50
## Proportion Var        0.34 0.09 0.07 0.06
## Cumulative Var        0.34 0.43 0.50 0.56
## Proportion Explained  0.61 0.16 0.13 0.11
## Cumulative Proportion 0.61 0.76 0.89 1.00
##
## Mean item complexity =  2.1
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.06
##
## Fit based upon off diagonal values = 0.97
```

```
# Calcula PCA para conjunto Harman.5, 2 fatores e rotação varimax
pc2 <- principal(Harman.5,2,rotate="varimax")
pc2
```

```
## Principal Components Analysis
## Call: principal(r = Harman.5, nfactors = 2, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##               RC1   RC2   h2    u2 com
## population   0.02  0.99 0.99 0.012 1.0
## schooling    0.94 -0.01 0.89 0.115 1.0
## employment   0.14  0.98 0.98 0.021 1.0
## professional 0.83  0.45 0.88 0.120 1.5
## housevalue   0.97 -0.01 0.94 0.062 1.0
##
##                        RC1  RC2
## SS loadings           2.52 2.15
## Proportion Var        0.50 0.43
## Cumulative Var        0.50 0.93
## Proportion Explained  0.54 0.46
```

```
## Cumulative Proportion 0.54 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.03
##  with the empirical chi square  0.29  with prob <  0.59
##
## Fit based upon off diagonal values = 1
```

```
# compare these correlations to the loadings
#  do it for unstandardized scores, and transform obliquely
round(cor(Harman.5,pc2$scores),2)
```

```
##               RC1   RC2
## population   0.02  0.99
## schooling    0.94 -0.01
## employment   0.14  0.98
## professional 0.83  0.45
## housevalue   0.97 -0.01
```

```
pc2o <- principal(Harman.5,2,rotate="promax",covar=TRUE)
pc2o
```

```
## Principal Components Analysis
## Call: principal(r = Harman.5, nfactors = 2, rotate = "promax", covar = TRUE)
## Unstandardized loadings (pattern matrix) based upon covariance matrix
##                 RC1     RC2      h2      u2   H2      U2
## population    -40.1 3440.30 1.2e+07 6.7e+03 1.00 5.7e-04
## schooling       1.5   -0.01 2.4e+00 8.1e-01 0.75 2.5e-01
## employment    110.3 1210.10 1.5e+06 5.4e+04 0.96 3.5e-02
## professional   87.7   48.30 1.0e+04 2.9e+03 0.78 2.2e-01
## housevalue   6368.4  -23.16 4.1e+07 2.2e+01 1.00 5.4e-07
##
##                               RC1         RC2
## SS loadings           40571924.73 13297286.79
## Proportion Var               0.75        0.25
## Cumulative Var               0.75        1.00
## Proportion Explained         0.75        0.25
## Cumulative Proportion        0.75        1.00
##
##  Standardized loadings (pattern matrix)
##              item   RC1   RC2   h2      u2
## population      1 -0.01  1.00 1.00 5.7e-04
## schooling       2  0.86 -0.01 0.75 2.5e-01
## employment      3  0.09  0.97 0.96 3.5e-02
## professional    4  0.76  0.42 0.78 2.2e-01
## housevalue      5  1.00  0.00 1.00 5.4e-07
##
##                  RC1  RC2
## SS loadings     3.76 1.23
## Proportion Var  0.75 0.25
## Cumulative Var  0.75 1.00
```

```
## Cum. factor Var 0.75 1.00
##
##  With component correlations of
##      RC1  RC2
## RC1 1.00 0.04
## RC2 0.04 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  6040.31
##  with the empirical chi square  8756488842  with prob <  0
##
## Fit based upon off diagonal values = 1
```

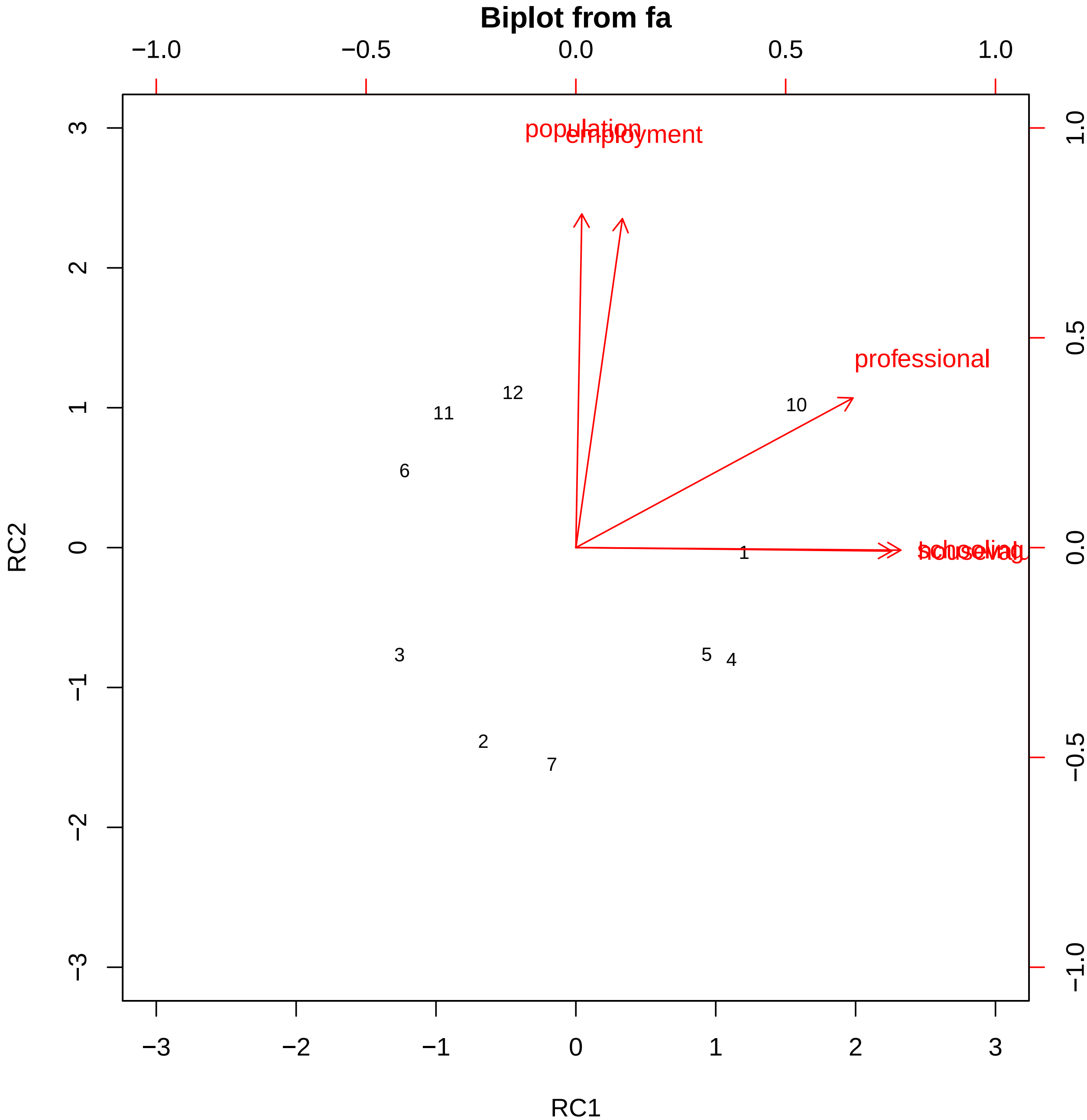
```
round(cov(Harman.5,pc2o$scores),2)
```

```
##                  RC1     RC2
## population     89.53 3438.79
## schooling       1.54    0.05
## employment    155.90 1214.25
## professional   89.56   51.60
## housevalue   6367.49  216.72
```

```
pc2o$Structure    #this matches the covariances with the scores
```

```
##                      RC1          RC2
## population     89.532324 3.438787e+03
## schooling       1.542246 4.714838e-02
## employment    155.900367 1.214255e+03
## professional   89.559504 5.160112e+01
## housevalue   6367.487506 2.167238e+02
```

```
biplot(pc2,main="Biplot of the Harman.5 socio-economic variables",labels=paste0(1:12))
```


Biplot from fa
−1.0
−0.5
0.0
0.5
1.0
population
employment
professional
schooling
housevalue
RC2
RC1
−3
−2
−1
0
1
2
3
1
2
3
4
5
6
7
10
11
12

Chapter 5

Análise Fatorial

Some *significant* applications are demonstrated in this chapter.

5.1 Example one

5.2 Example two

Bibliography