Screeplot, Biplot e Aplicações de Componentes Principais

Guilherme Ludwig

September 16, 2019

Componentes Principais

Seja $\hat{\mathbf{\Sigma}} = \mathbf{S}$, com decomposição espectral $\mathbf{S} = \hat{\mathbf{Q}}\hat{\mathbf{\Lambda}}\hat{\mathbf{Q}}^t$. Seja $\mathbf{A} = (n-1)^{-1/2}(\mathbf{X} - \mathbf{1}_{n\times 1}\bar{\mathbf{X}}_{1\times p})$ com decomposição SVD $\mathbf{A} = \mathbf{UDV}^t$.

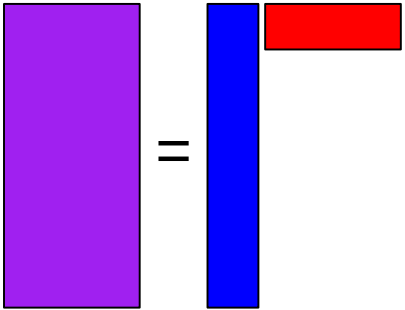
Naturalmente, $\mathbf{A}^t\mathbf{A} = \mathbf{S}$, logo $\mathbf{VD}^2\mathbf{V}^t = \hat{\mathbf{Q}}\hat{\mathbf{\Lambda}}\hat{\mathbf{Q}}^t$. É possível fazer a mesma decomposição em $\mathbf{R}$, a matriz de correlações amostrais (os resultados podem ser bem diferentes, examinaremos um caso em breve).

Para deixar claro:

$$\mathbf{Z} = \underbrace{\mathbf{UD}}_{\text{Scores}}\ \underbrace{\mathbf{V}^t}_{\text{Loadings}}$$

Alguns softwares fazem $\mathbf{Z} = \underbrace{\mathbf{UD}^{\delta}}_{\text{Scores}}\underbrace{\mathbf{D}^{1-\delta}\mathbf{V}^t}_{\text{Loadings}}$ para algum $\delta \in [0,1]$, onde $\mathbf{D}^{\delta}$ é um abuso de notação para a matriz com entradas d_{ij}^{δ}.

Esparsidade



Componentes Principais

Suponha que $\mathbf{X}_{n\times p}$ está padronizada. Note que

$$\mathbf{X}_{n\times p}\mathbf{V}_{p\times p} = \mathbf{U}_{n\times n}\mathbf{D}_{n\times p},$$

em que

$$\mathbf{D}_{n\times p} = \begin{pmatrix} d_{11} & 0 & 0 & \cdots & 0 \\ 0 & d_{22} & 0 & \cdots & 0 \\ 0 & 0 & d_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_{pp} \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

onde $d_{11} \geq d_{22} \geq \cdots \geq d_{pp} \geq 0$, assumindo, é claro, $p < n$.

Componentes Principais

Se $\mathrm{Var}(\mathrm{vec}(\mathbf{X})) = \mathbf{\Sigma}_{p\times p} \otimes \mathbf{I}_{n\times n}$, então

$$\begin{aligned}
\mathrm{Var}(\mathrm{vec}(\mathbf{XV})) &= \mathrm{Var}(\mathrm{vec}(\mathbf{I}_{n\times n}\mathbf{XV})) \\
&= \mathrm{Var}([\mathbf{V}^t \otimes \mathbf{I}_{n\times n}]\mathrm{vec}(\mathbf{X})) \\
&= [\mathbf{V}^t \otimes \mathbf{I}_{n\times n}]\mathrm{Var}(\mathrm{vec}(\mathbf{X}))[\mathbf{V}^t \otimes \mathbf{I}_{n\times n}]^t \\
&= [\mathbf{V}^t \otimes \mathbf{I}_{n\times n}][\mathbf{\Sigma} \otimes \mathbf{I}_{n\times n}][\mathbf{V} \otimes \mathbf{I}_{n\times n}] \\
&= [\mathbf{V}^t \otimes \mathbf{I}_{n\times n}][\mathbf{\Sigma} \otimes \mathbf{I}_{n\times n}][\mathbf{V} \otimes \mathbf{I}_{n\times n}] \\
&= [\mathbf{V}^t\mathbf{\Sigma}\mathbf{V}] \otimes \mathbf{I}_{n\times n} = \mathbf{\Lambda}_{p\times p} \otimes \mathbf{I}_{n\times n}
\end{aligned}$$

Naturalmente: sua estimativa de $\mathbf{V}$, com base em $\mathbf{S}$, pode não ser muito boa (n pequeno, problemas de mau condicionamento de $\mathbf{X}$ etc.).

ABC da decomposição espectral

Considere a matriz

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Então λ_1, λ_2 são soluções de

$$(1-\lambda)^2 - \rho^2 = 0,$$

ou simplesmente $\lambda_1 = 1+\rho$, $\lambda_2 = 1-\rho$, com auto-vetores (normalizados)

$$\mathbf{v}_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}.$$

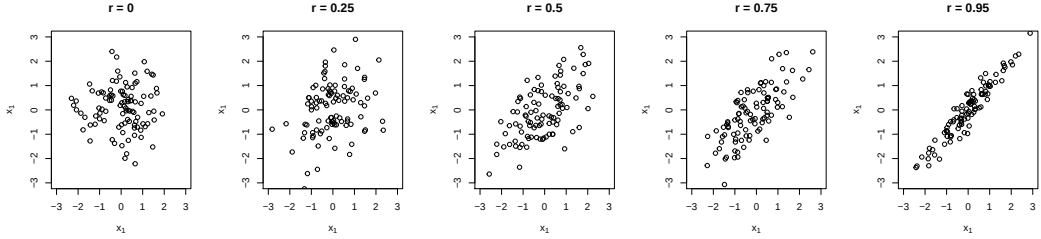
Note que

$$\mathbf{\Sigma} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^t = \sum_{i=1}^{2} \lambda_i \mathbf{v}_i \mathbf{v}_i^t = \frac{(1+\rho)}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} + \frac{(1-\rho)}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

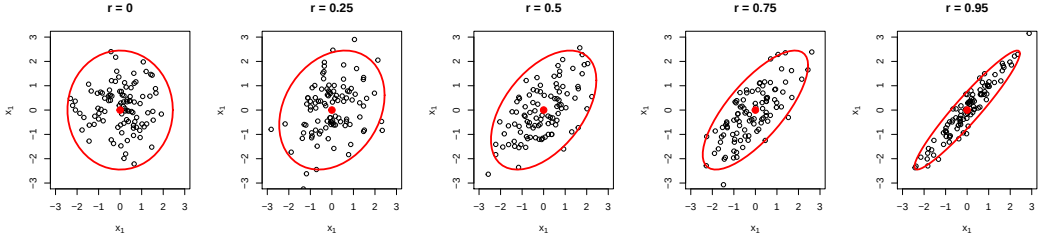
Exemplo simulado 1

```
r <- c(0, .25, .5, .75, .95)
# List of 5 matrices, differnt r
R0.5 <- lapply(r, function(r) {
  Sigma <- matrix(c(1, r, r, 1), ncol = 2)
  ST <- eigen(Sigma, symmetric = TRUE)
  # V = ST$vectors; L = diag(ST$values)
  # all.equal(Sigma, V %*% L %*% t(V))
  return(diag(ST$values^0.5) %*% t(ST$vectors))
  })
set.seed(1)
X <- lapply(R0.5, function(DVt) matrix(rnorm(100*2), ncol = 2) %*% DVt)
```

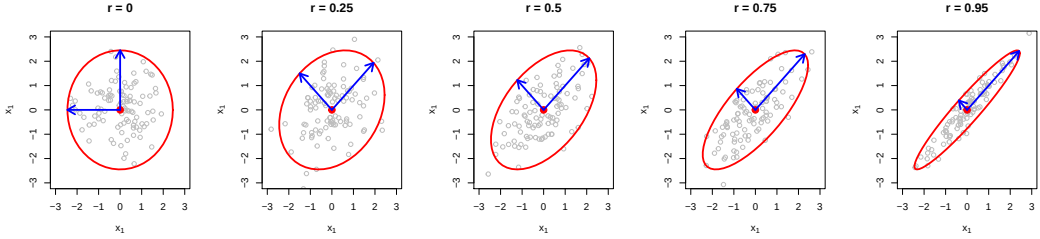
Exemplo simulado 1



Elipses $\mathbb{P}(\mathbf{X}_i^t \mathbf{\Sigma}^{-1} \mathbf{X}_i \leq \chi_p^2) = 0.95$



Autopares $(\sqrt{\chi^2_2(0.95)}\lambda_i, \mathbf{v}_i)$



Variância explicada e redução de dimensão

É comum definir medidas como $|\mathbf{\Sigma}|$ ou $\mathrm{tr}(\mathbf{\Sigma})$ como medidas de **variância total**. A idéia de usar análise de componentes principais para reduzir a dimensão de um problema é simples: escolhemos um número q de componentes tal que a variabilidade explicada pelas primeiras q componentes seja próxima da variabilidade total.

Note que $\lambda_j = \mathrm{Var}(\mathbf{X}\mathbf{v}_j)$ é a variância do j-ésimo componente $\mathbf{X}\mathbf{v}_j$.

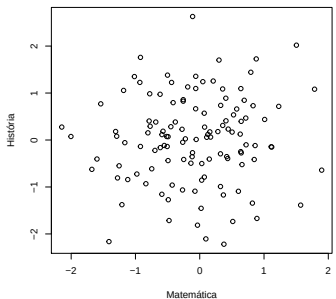
Defina a proporção da variância explicada pelo componente j por

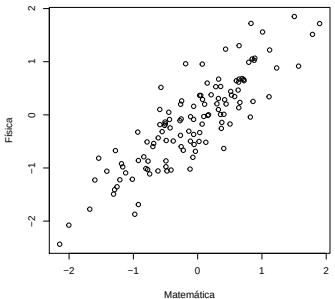
$$c_j = \frac{\lambda_j}{\lambda_1 + \cdots + \lambda_p} = \frac{\lambda_j}{\mathrm{tr}(\mathbf{\Sigma})}.$$

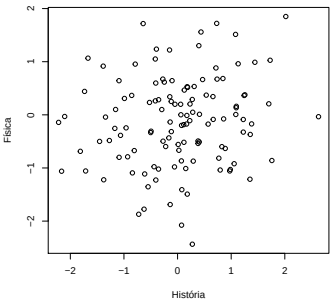
No exemplo anterior, quando $\rho = 0$, temos autovalores $\lambda_1 = 1, \lambda_2 = 1$; logo $c_1 = c_2 = 1/2$. Mas se $\rho = 0.95$, então $c_1 = (1 + 0.95)/(1 + 0.95 + 1 - 0.95) = 0.975$. Ou seja, o primeiro componente explica 97.5% da variabilidade total.

Exemplo simulado 2

Por exemplo, notas de 120 alunos (independentes entre si) em três matérias, padronizadas:







Exemplo simulado 2

```
summary(modelPCA <- princomp(Notas))
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3
## Standard deviation     1.1183738 0.9117907 0.29706365
## Proportion of Variance 0.5762891 0.3830511 0.04065982
## Cumulative Proportion  0.5762891 0.9593402 1.00000000
```

Note: `prcomp()` usa decomposição SVD em $\mathbf{S}$, enquanto `princomp()` usa `eigen()`. Os resultados devem ser parecidos exceto por álgebra de ponto flutuante e rotações nos eixos (lembre-se que $\mathbf{v}_j\mathbf{v}_j^t = (-\mathbf{v}_j)(-\mathbf{v}_j)^t$ para todo j).

Exemplo simulado 2

Há 200 anos o output de `loadings()` mostra uma proporção incorreta de variância explicada; ignorem a segunda parte do output:

```
loadings(modelPCA)
```

```
##
## Loadings:
##             Comp.1 Comp.2 Comp.3
## Matemática   0.641  0.240  0.729
## História     0.288 -0.956
## Física       0.711  0.170 -0.682
##
##                Comp.1 Comp.2 Comp.3
## SS loadings     1.000  1.000  1.000
## Proportion Var  0.333  0.333  0.333
## Cumulative Var  0.333  0.667  1.000
```

Screeplot

Uma maneira de exibir a % da variabilidade explicada por cada componente é através de um screeplot. Neste caso, o gráfico é um gráfico de barras para cada j em

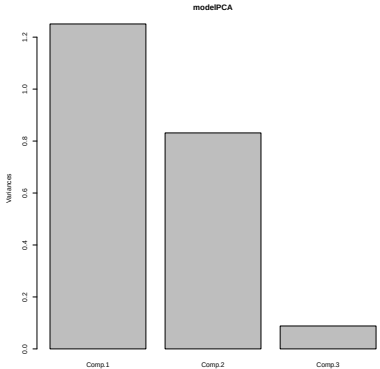
$$c_j = \frac{\lambda_j}{\lambda_1 + \cdots + \lambda_p} = \frac{\lambda_j}{\text{tr}(\mathbf{\Sigma})}.$$

No `R`, obtém-se o screeplot através do método `plot.princomp()` (mas esse método plota λ_j, não c_j).

- Comumente, dizemos que a PCA explica $100\alpha\%$ da variância se $q := \arg\min_q \{\sum_{j \leq q} c_j \geq \alpha\}$.
- No exemplo simulado, $c_1 \approx 0.606$, $c_2 \approx 0.346$ e $c_3 \approx 0.048$. Consequentemente, duas componentes explicam 95.2% da variabilidade total.

Screeplot

```
plot(modelPCA)
```



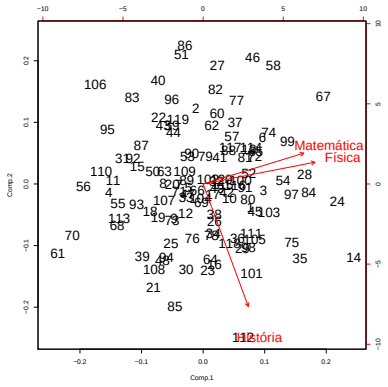
Biplot

O biplot é uma ferramenta para visualizar a conexão entre os componentes principais (ou scores) $\mathbf{U}_{n\times q}\mathbf{D}^{\delta}_{p\times q} = \mathbf{X}_{n\times p}\mathbf{D}^{1-\delta}_{p\times q}\mathbf{V}_{q\times q}$ e os loadings $\mathbf{D}^{1-\delta}_{p\times q}\mathbf{V}_{q\times q}$

- O biplot é feito com um par de componentes principais. Em geral, a primeira e a segunda (pois os autovalores estão em ordem decrescente), mas é preciso examinar todas as que forem importantes.
- São colocados eixos com as direções de variação, correspondendo às primeiras q colunas de $\mathbf{V}$. Em geral softwares estatísticos escolhem δ que coloque os dados em escalas comparáveis (o R faz dois eixos diferentes).
- Variáveis positivamente correlacionadas têm eixos sobrepostos, e negativamente correlacionadas tem eixos opostos. Eixos ortogonais implicam em independência, se os dados forem normais. Em tese é possível interpretar o ângulo dos eixos como medida de correlação, mas só consideraremos isso visualmente.

Biplot

```
biplot(modelPCA, cex = 2)
```



Exemplo Johnson & Wichern (2007), p. 477

O exemplo a seguir coleciona os recordes masculinos para cada país que disputou a *2005 World Championships in Athletics*, Helsinki, Finlândia (fonte: *IAAF/ATES Track and Field Statistics Handbook*). As primeiras três medidas estão em segundos, as outras em minutos.

```
runner <- read.delim("data/T8-6.DAT", header = FALSE,
                     row.names = 1, sep = " ")
colnames(runner) <- c("100m", "200m", "400m", "800m", "1500m",
                      "5000m", "10000m", "Marathon")
```

Exemplo Johnson & Wichern (2007), p. 477

```
head(runner)
```

```
##             100m  200m  400m 800m 1500m 5000m 10000m Marathon
## Argentina 10.23 20.37 46.18 1.77  3.68 13.33  27.65   129.57
## Australia  9.93 20.06 44.38 1.74  3.53 12.93  27.53   127.51
## Austria   10.15 20.45 45.80 1.77  3.58 13.26  27.72   132.22
## Belgium   10.14 20.19 45.02 1.73  3.57 12.83  26.87   127.20
## Bermuda   10.27 20.30 45.26 1.79  3.70 14.64  30.49   146.37
## Brazil    10.00 19.89 44.29 1.70  3.57 13.48  28.13   126.05
```

A sugestão do Johnson & Wichern é fazer a análise em metros por segundo.

Metros por segundo

```r
# Different units
distances <- c(100, 200, 400, 800/60,
               1500/60, 5000/60, 10000/60, 42195/60)
speed <- sweep(runner, 2, distances, FUN = function(x,y) y/x)
round(head(speed), 2)
```

```
##             100m  200m 400m 800m 1500m 5000m 10000m Marathon
## Argentina   9.78  9.82 8.66 7.53  6.79  6.25   6.03     5.43
## Australia  10.07  9.97 9.01 7.66  7.08  6.44   6.05     5.52
## Austria     9.85  9.78 8.73 7.53  6.98  6.28   6.01     5.32
## Belgium     9.86  9.91 8.88 7.71  7.00  6.50   6.20     5.53
## Bermuda     9.74  9.85 8.84 7.45  6.76  5.69   5.47     4.80
## Brazil     10.00 10.06 9.03 7.84  7.00  6.18   5.92     5.58
```

Centralização

```
round(head(scale(speed, center = TRUE, scale = FALSE)), 2)
```

```
##             100m 200m  400m  800m 1500m 5000m 10000m Marathon
## Argentina -0.02 0.08 -0.07 -0.01 -0.06  0.12   0.17     0.14
## Australia  0.28 0.23  0.28  0.12  0.23  0.31   0.20     0.23
## Austria    0.06 0.04  0.00 -0.01  0.13  0.15   0.15     0.03
## Belgium    0.07 0.16  0.15  0.16  0.15  0.36   0.34     0.24
## Bermuda   -0.06 0.11  0.10 -0.10 -0.10 -0.44  -0.39    -0.48
## Brazil     0.21 0.31  0.30  0.30  0.15  0.05   0.07     0.29
```

Padronização

```
round(head(scale(speed)),2) # Default: scale = TRUE
```

```
##             100m 200m  400m  800m 1500m 5000m 10000m Marathon
## Argentina -0.08 0.30 -0.28 -0.07 -0.22  0.37   0.55     0.44
## Australia  1.33 0.89  1.06  0.53  0.84  1.00   0.64     0.72
## Austria    0.29 0.14 -0.01 -0.07  0.48  0.48   0.50     0.09
## Belgium    0.33 0.64  0.57  0.74  0.55  1.16   1.12     0.77
## Bermuda   -0.26 0.43  0.39 -0.45 -0.36 -1.43  -1.28    -1.55
## Brazil     1.00 1.23  1.13  1.37  0.55  0.15   0.22     0.93
```

Análise de componentes principais

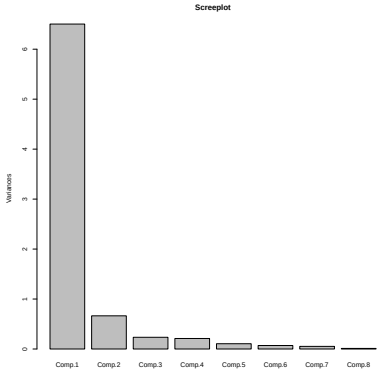
```
summary(modelPCArun <- princomp(scale(speed)))
```

```
## Importance of components:
##                            Comp.1     Comp.2     Comp.3     Comp.4
## Standard deviation     2.5501082 0.81487080 0.48424490 0.45877527
## Proportion of Variance 0.8282189 0.08456788 0.02986469 0.02680575
## Cumulative Proportion  0.8282189 0.91278676 0.94265145 0.96945720
##                             Comp.5      Comp.6      Comp.7      Comp.8
## Standard deviation     0.32500667 0.264508706 0.229187818 0.108149450
## Proportion of Variance 0.01345279 0.008910618 0.006689767 0.001489624
## Cumulative Proportion  0.98290999 0.991820610 0.998510376 1.000000000
```

Dois componentes explicam pelo menos 90% da variabilidade. O primeiro componente explica 82%.

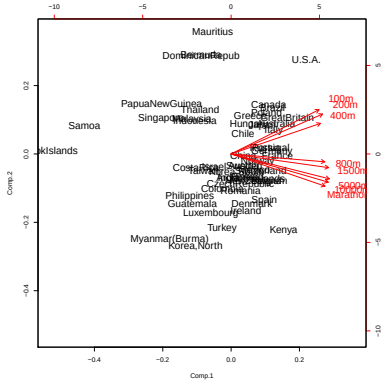
Screeplot

```
plot(modelPCArun, main = "Screeplot")
```



Biplot

```
biplot(modelPCArun, cex = 1.5)
```



Interpretação

- Primeiro loading dá pesos iguais a todas as características. Em outras palavras: a dimensão dos dados que mais explica a variabilidade entre países é a velocidade média dos atletas (entre todos os tipos de corrida).
- Segundo loading diz que a segunda maior variabilidade entre países está entre países com alta velocidade em corridas curtas (e.g. USA: 10.22m/s em corridas de 100m, 10.35m/s em corridas de 200m) contra países com alta velocidade em corridas longas (e.g. Quênia: 6.30m/s em 10000m, 5.65m/s em maratonas).

Usando covariâncias ao invés de correlações

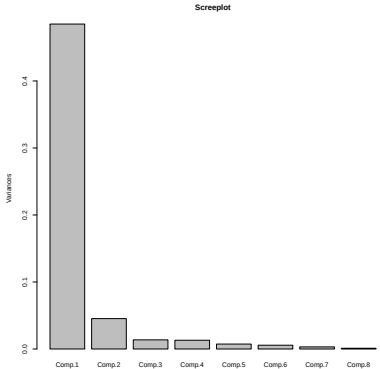
```
summary(modelPCArunC <- princomp(scale(speed, scale = FALSE)))
```

```
## Importance of components:
##                            Comp.1     Comp.2     Comp.3     Comp.4
## Standard deviation     0.6963482 0.21299720 0.11685311 0.11434212
## Proportion of Variance 0.8443571 0.07899888 0.02377682 0.02276594
## Cumulative Proportion  0.8443571 0.92335597 0.94713279 0.96989873
##                            Comp.5      Comp.6      Comp.7      Comp.8
## Standard deviation     0.08592579 0.075118210 0.056220447 0.033165590
## Proportion of Variance 0.01285643 0.009825703 0.005503789 0.001915352
## Cumulative Proportion  0.98275516 0.992580859 0.998084648 1.000000000
```

Resultados mudam pouco se escalas (desvio-padrão das dimensões) são parecidas.

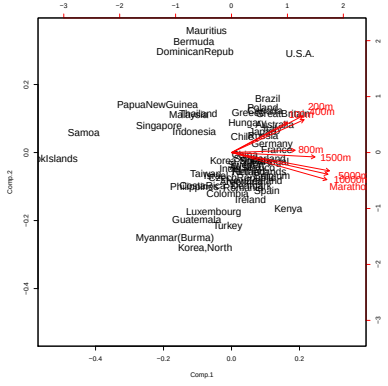
Screeplot

```
plot(modelPCArunC, main = "Screeplot")
```



Biplot

```
biplot(modelPCArunC, cex = 1.5)
```



Análise de texto

Análise de componentes principais de texto: um exemplo usando a obra do Machado de Assis.

Origem dos dados: http://www.machadodeassis.ufsc.br/obras.html (eu salvei as páginas em formato texto simples).

```
list.files("data/Machado/")
```

```
##  [1] "1872_ressurreicao.txt"
##  [2] "1874_a_mao_e_a_luva.txt"
##  [3] "1876_helena.txt"
##  [4] "1878_iaia_garcia.txt"
##  [5] "1880_memorias_postumas_de_bras_cubas.txt"
##  [6] "1885_casa_velha.txt"
##  [7] "1891_quincas_borba.txt"
##  [8] "1899_dom_casmurro.txt"
##  [9] "1904_esau_e_jaco.txt"
## [10] "1908_memorial_de_aires.txt"
```

Lendo os arquivos e removendo pontuação

```
fname <- list.files("data/Machado/")
f <- function(x){
  rawData <- readLines(con <- file(paste0("data/Machado/", x), encoding =
  close(con)
  for(i in seq_along(rawData)){
    if(substr(rawData[i],1,1) %in% c("*","/","-")) {
      rawData[i] <- ""
    } else if(rawData[i] == " "){
      rawData[i] <- ""
    }
  }
  rawData <- gsub("[[:punct:]]", "", rawData)
  rawData <- gsub("-", "", rawData) # emdash
  rawData <- paste0(rawData, collapse = " ")
  rawData <- gsub("\\s{2,}", " ", rawData)
  return(tolower(rawData))
}
```

Stopwords

```
library(tm)
stopwords("portuguese")
```

```
##   [1] "de"          "a"           "o"           "que"
##   [5] "e"           "do"          "da"          "em"
##   [9] "um"          "para"        "com"         "não"
##  [13] "uma"         "os"          "no"          "se"
##  [17] "na"          "por"         "mais"        "as"
##  [21] "dos"         "como"        "mas"         "ao"
##  [25] "ele"         "das"         "à"           "seu"
##  [29] "sua"         "ou"          "quando"      "muito"
##  [33] "nos"         "já"          "eu"          "também"
##  [37] "só"          "pelo"        "pela"        "até"
##  [41] "isso"        "ela"         "entre"       "depois"
##  [45] "sem"         "mesmo"       "aos"         "seus"
##  [49] "quem"        "nas"         "me"          "esse"
##  [53] "eles"        "você"        "essa"        "num"
```

Stopwords

```
f <- function(x) removeWords(tolower(x),
                             stopwords("portuguese"))
romances <- lapply(romances, f)
```

Corpus

```
D <- VCorpus(VectorSource(romances))
# tdm contém matriz de palavras que apareceram
# pelo menos duas vezes
tdm <- TermDocumentMatrix(D,
                          control = list(bounds =
                                           list(global = c(2, Inf))))
tdm$dimnames$Docs <- names(romances) <- gsub(".txt$", "",
                                             list.files("data/Machado/"))
```

Alguns dados

```
dados <- inspect(tdm)
```

```
## <<TermDocumentMatrix (terms: 14522, documents: 10)>>
## Non-/sparse entries: 60681/84539
## Sparsity           : 58%
## Maximal term length: 19
## Weighting          : term frequency (tf)
## Sample             :
##        Docs
## Terms   1872_ressurreicao 1874_a_mao_e_a_luva 1876_helena 1878_iaia_gar
##   ainda                82                  80          99
##   casa                 80                  55         147
##   coisa                47                  74          88
##   disse               151                  97         209
##   nada                 39                  80          80
##   olhos                96                 114         151
##   outra                39                  78          51
```

Palavras mais frequentes

```
d2 <- apply(as.matrix(tdm), 1, sum)
sort(d2, decreasing = TRUE)[1:20]
```

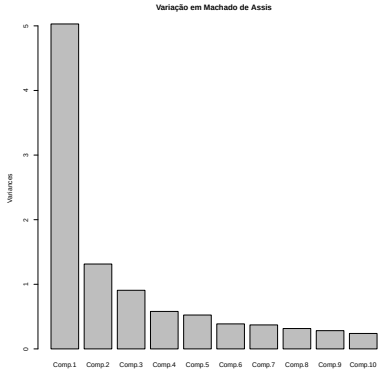
```
##  disse   casa  olhos   tudo    ser  ainda   nada  tempo  outra  coisa
##   1426   1235   1234   1144   1135   1133   1080   1063   1028    964
##  outro    vez    tão  pouco  agora  assim    dia  podia porque   dois
##    955    923    861    834    798    778    778    767    752    750
```

Bag of Words com 200 palavras

```
bow <- sort(d2, decreasing = TRUE)[1:200]
X <- as.matrix(tdm)[names(bow),] # Indexando por valor
temp <- gsub("_"," ", list.files("data/Machado/"))
temp <- gsub("\\.txt", "", temp)
anos <- substr(temp, 1, 4)
colnames(X) <- paste0(substr(temp, 6, 100), " (",
                      anos, ")")
```

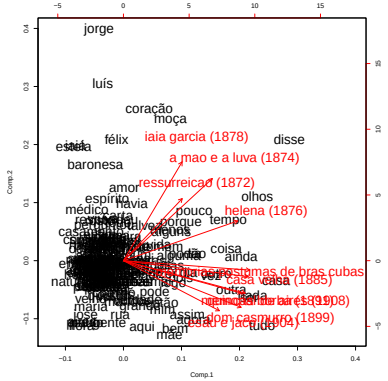
Análise de componentes principais

```
plot(model <- princomp(scale(X)), "Variação em Machado de Assis")
```



Análise de componentes principais

```
biplot(model, cex = 2)
```



Interpretação

- Primeiro loading dá pesos parecidos a todas as características (romances). Em outras palavras: provavelmente o que está explicando a variabilidade entre romances é a quantidade média de palavras por romance (romances longos contra romances breves).
- Segundo loading distingue entre romances da fase romântica (pré-Memórias Póstumas...) e romances da fase realista ("Memórias Póstumas..." em diante). Termos como "amor", "espírito", "olhos" e "coração" são mais comuns na fase romântica, enquanto "tudo", "nada", "velho", "ser" e "natural" são mais comuns na fase realista.