Exemplo de Análise de Componentes Principais (PCA)

20 de agosto de 2012

1 Exercício dos vendedores

Uma empresa avaliou o desempenho dos seus vendedores por meio de escores em quatro testes (exames) e três indicadores de vendas. A avaliação foi feita para os 50 vendedores da empresa. Os indicadores de vendas foram:

- crescimento das vendas (CV),
- rentabilidade vendas (RV),
- vendas de novas contas (VNC).

As medidas desses indicadores foram convertidas para uma escala onde 100 indica o desempenho médio. E, também, cada um dos vendedores foi submetido a quatro testes com o propósito de medir:

- a criatividade (CR);
- o raciocínio mecânico (RM);
- o raciocínio abstrato (RA);
- a habilidade em matemática (HM).

Estes são dados para as $n = 50$ observações de $p = 7$ variáveis.
Use Análise de Componentes Principais para explorar as informações dadas pelas variáveis mensionadas.

1.1 Minha resolução:

Importando dados de um arquivo .xls

```
> library(RODBC)
> dados <- odbcConnectExcel("exerc-vendedores.xls")
> X <- sqlFetch(dados, "Vendedores")
> odbcClose(dados)
```

Verificando a ordem :

```
> dim(X)
```

```
[1] 50  7
```

```
> str(X)
```

```
'data.frame':	50 obs. of  7 variables:
 $ CV : num  93 88.8 95 101.3 102 ...
 $ RV : num  96 91.8 100.3 103.8 107.8 ...
 $ VNC: num  97.8 96.8 99 106.8 103 ...
 $ CR : num  9 7 8 13 10 10 9 18 10 14 ...
 $ RM : num  12 10 12 14 15 14 12 20 17 18 ...
 $ RA : num  9 10 9 12 12 11 9 15 13 11 ...
 $ HM : num  20 15 26 29 32 21 25 51 31 39 ...
```

Calculando matriz de covariância e de correlação
É imprescindível o cálculo das matrizes S e R, pois a análise de componentes principais ocorre por meio delas. Quando utilizar S ou R, será descrito porteriormente. Observe que essas matrizes são simétricas. Cada elemento de S e R foi calculado da seguinte maneira: (Johnson;Wincher (2007), p.7-8, p.139)

$$s_{kk} = s_k^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_{jk} - \bar{x}_k)^2$$

$$s_{ik} = \frac{1}{n-1}\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)$$

$$r_{jk} = \frac{s_{jk}}{\sqrt{s_{jj}}\sqrt{s_{kk}}}$$

onde $j = 1, ..., n$, $k = 1, ..., p$ e $i = 1, ..., p$.
Matricialmente:

$$S = \frac{1}{n-1}\mathbf{X}'\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'\right)\mathbf{X}$$

$$R = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}$$

onde

$$\mathbf{D}^{-1/2} = \begin{bmatrix} \sqrt{s_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{s_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{s_{pp}} \end{bmatrix}^{-1}$$

```
> (S <- round(cov(X),2))

       CV     RV   VNC    CR    RM    RA     HM
CV  53.84  68.79 30.56 16.58 17.59 10.59  71.70
RV  68.79 102.50 40.20 21.66 25.56 10.08 100.74
VNC 30.56  40.20 22.21 13.04 10.17  6.46  42.34
CR  16.58  21.66 13.04 15.60  7.90  1.24  17.18
RM  17.59  25.56 10.17  7.90 11.46  2.80  20.49
RA  10.59  10.08  6.46  1.24  2.80  4.58  12.77
HM  71.70 100.74 42.34 17.18 20.49 12.77 111.04

> (R <- round(cor(X), 2))

      CV   RV  VNC   CR   RM   RA   HM
CV  1.00 0.93 0.88 0.57 0.71 0.67 0.93
RV  0.93 1.00 0.84 0.54 0.75 0.47 0.94
VNC 0.88 0.84 1.00 0.70 0.64 0.64 0.85
CR  0.57 0.54 0.70 1.00 0.59 0.15 0.41
```

```
RM  0.71 0.75 0.64 0.59 1.00 0.39 0.57
RA  0.67 0.47 0.64 0.15 0.39 1.00 0.57
HM  0.93 0.94 0.85 0.41 0.57 0.57 1.00
```

Teste de esfericidade de Bartlett
De acordo com Mingoti (2007), para que a análise de componentes principais tenha algum sentido, é necessário que as variáveis sejam correlacionadas. Se as matrizes de covariância e de correlação forem diagonais, a aplicaçao desta técnica simplesmente vai desenvolver, em alguma ordem, as próprias variáveis originais. Assim, testamos as seguintes hipóteses:

$$H_0 : R = I \ \times \ H_0 : R \neq I$$

$$\chi^2 = -\left((n-1) - \frac{2p+5}{6}\right) ln\,|R|$$

com $\nu = \frac{p(p-1)}{2}$graus de liberdade.

```
> # Insira matriz de correlação R e número de observações
> library(psych)
> cortest.bartlett(R, n = nrow(X))

$chisq
[1] 495.5147

$p.value
[1] 1.276926e-91

$df
[1] 21
```

Como o p-valor é praticamente 0, a matrix de correlação não é diagonal.

Análise de componentes principais utilizando S e R
De acordo com Mingoti(2007), quando a análise de componentes principais é utiliza a matriz S, as covariâncias são influenciadas pelas variáveis de maior variância, sendo, portanto, muito utilizada nos casos em que existe uma discrepância muito acentuada entre essas variâncias. A discrepância é muitas vezes causada pela diferença das unidades de medidas das variáveis. Esse problema pode ser amenizado se uma transformação for efetuada nos dados originais, de modo a equilibrar os valores de variância ou a colocar os dados na mesma escala de medida. Uma das transformações mais comuns é aquela em que cada variável é padronizada pela sua média e desvio padrão, sendo a técnica de componentes principais aplicada à matriz de covariâncias das variáveis padronizadas. Este procedimento é equivalente a obter-se as componentes principais através da matriz de correlação R das variáveis originais.

Padronização dos dados
Os dados podem ser padronizados da seguinte forma:

$$\bar{x}_k = \frac{1}{n}\sum_{n=1}^{j} x_{jk}$$

$$z_{ji} = \frac{x_{ji} - \bar{x}_k}{s_k}$$

ou matricialmente,

$$\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X}'\mathbf{1}$$

$$\mathbf{Z} = \mathbf{V}^{-1/2}\left[\mathbf{X} - \bar{\mathbf{x}}\right]$$

Padronizando elementos de X

```
> z <- scale(X)
```

Iniciando a ACP

Calculando autovalores, autovetores, porcentual de explicação sobre matriz R

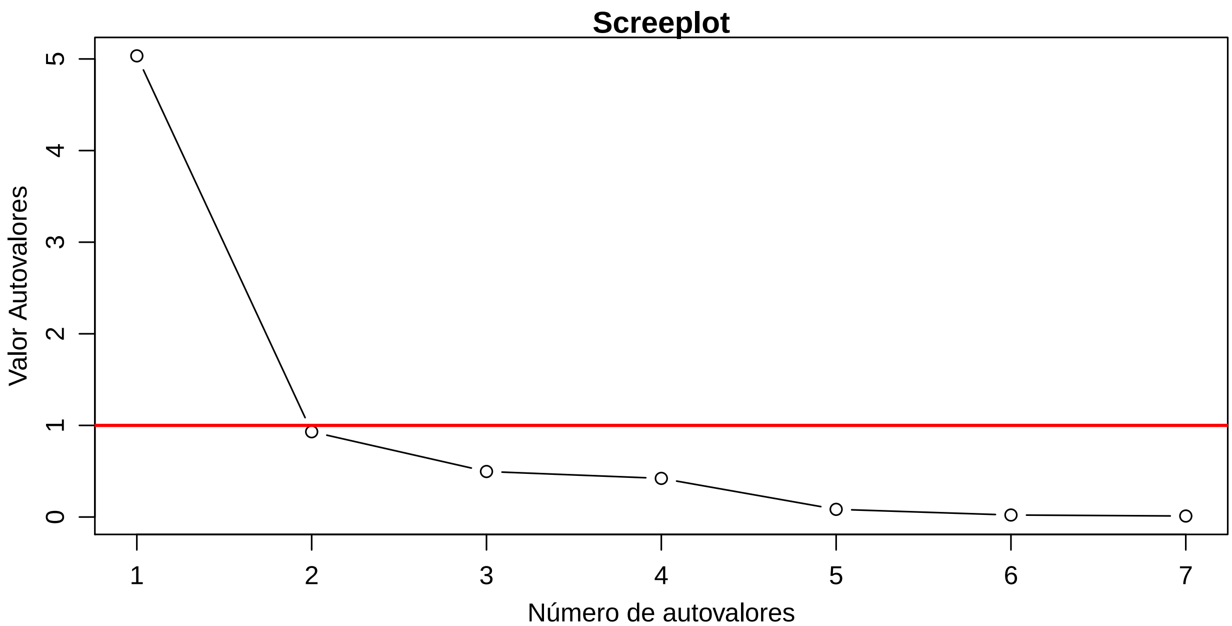
```
> # Autovalores e autovetores
> autovalor.autovetor<- as.data.frame(eigen(R))
> # Recebendo porcentual das variâncias e porcentagem acumulativa
> var.porc = autovalor.autovetor$values / sum(autovalor.autovetor$values)*100
> var.acum = cumsum(var.porc)
> (porc.explic <- round(data.frame(autovalores = autovalor.autovetor$values,
+                            var.porc = var.porc, var.acum = var.acum), 3))

  autovalores var.porc var.acum
1       5.034   71.910   71.910
2       0.931   13.298   85.208
3       0.497    7.100   92.308
4       0.422    6.027   98.335
5       0.084    1.194   99.529
6       0.022    0.308   99.837
7       0.011    0.163  100.000
```

Escolha de m componentes

A escolha de quantas componentes principais deve-se:

- ao porcentual de variância explicada;
- ao número de autovalores maiores do que 1 (critério Kaiser);
- ao gráfico screeplot;
- à experiência do pesquisador.

```
> plot(porc.explic$autovalores, main = "Screeplot", type = "b",
+      ylab = "Valor Autovalores", xlab = "Número de autovalores", axes = T)
> abline(h=1, col=2, lwd=2)
```



As componentes principais $\mathbf{Y}_k$ são dadas por

$$\mathbf{Y}_k = \mathbf{e}_k'\mathbf{z}$$

onde $\mathbf{z}$ é a matriz de dados padronizados.

```
> # Matriz e dos coeficientes das combinações lineares referentes às componentes principais Yk
>
> (e <- round(as.matrix(autovalor.autovetor[,-1]),3))

     vectors.1 vectors.2 vectors.3 vectors.4 vectors.5 vectors.6 vectors.7
[1,]    -0.434     0.110     0.091    -0.043     0.611     0.321     0.560
[2,]    -0.421    -0.025     0.438     0.021     0.004    -0.791     0.060
[3,]    -0.420    -0.014    -0.208    -0.323    -0.712     0.126     0.390
[4,]    -0.294    -0.672    -0.444    -0.304     0.276    -0.097    -0.294
[5,]    -0.350    -0.294    -0.005     0.843    -0.175     0.208    -0.076
[6,]    -0.290     0.636    -0.614     0.154     0.105    -0.219    -0.227
[7,]    -0.407     0.209     0.427    -0.256    -0.044     0.392    -0.622
```

logo,

$$Y_1 = -0,433z_1 - 0,421z_2 - 0,420z_3 - 0,294z_4 - 0,349z_5 - 0,289z_6 - 0,407z_7$$

$$Y_2 = 0,109z_1 - 0,025z_2 - 0,014z_3 - 0,672z_4 - 0,294z_5 + 0,636z_6 + 0,209z_7$$

$$Y_3 = 0,091z_1 + 0,438z_2 - 0,208z_3 - 0,444z_4 - 0,004z_5 - 0,614z_6 + 0,427z_7$$

$$Y_4 = -0,043z_1 + 0,021z_2 - 0,323z_3 - 0,304z_4 + 0,843z_5 + 0,154z_6 - 0,256z_7$$

$$Y_5 = 0,611z_1 + 0,004z_2 - 0,712z_3 + 0,276z_4 - 0,175z_5 + 0,105z_6 - 0,044z_7$$
$$Y_6 = 0,321z_1 - 0,791z_2 + 0,126z_3 - 0,097z_4 + 0,208z_5 - 0,219z_6 + 0,392z_7$$
$$Y_7 = 0,560z_1 + 0,060z_2 + 0,390z_3 - 0,293z_4 - 0,076z_5 - 0,227z_6 - 0,622z_7$$

onde $V(Y) = \begin{bmatrix} 5,034 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,931 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,497 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,422 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,084 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,022 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,011 \end{bmatrix}$

```
> (V <- round(diag(autovalor.autovetor[,1]),3))
```

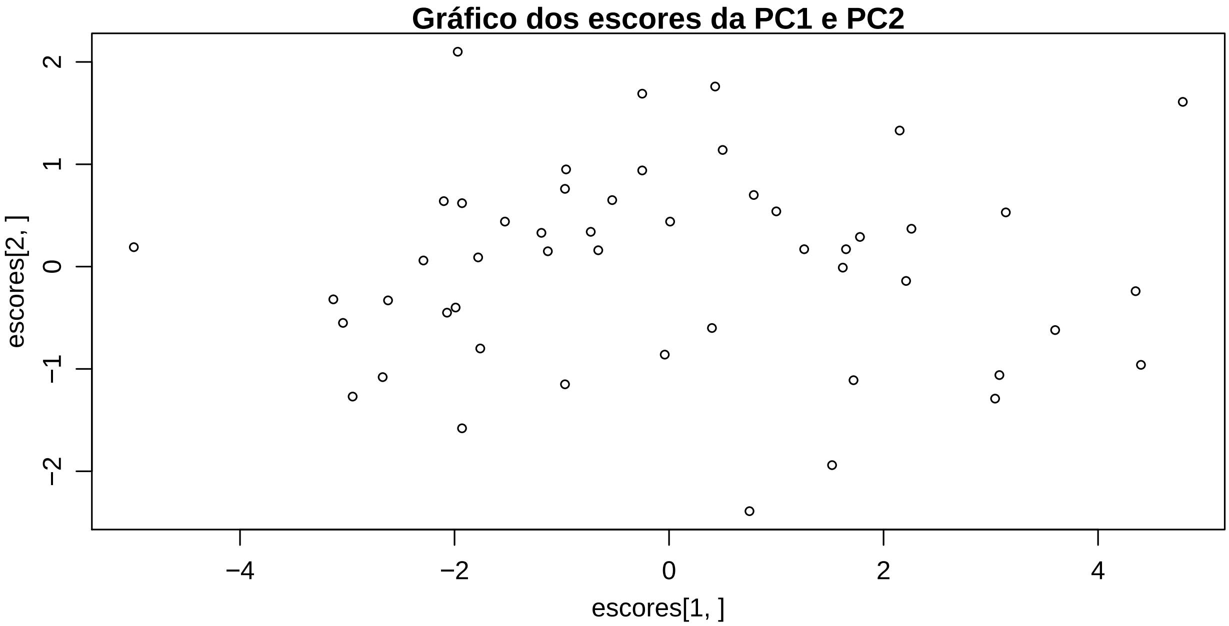
```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
[1,] 5.034 0.000 0.000 0.000 0.000 0.000 0.000
[2,] 0.000 0.931 0.000 0.000 0.000 0.000 0.000
[3,] 0.000 0.000 0.497 0.000 0.000 0.000 0.000
[4,] 0.000 0.000 0.000 0.422 0.000 0.000 0.000
[5,] 0.000 0.000 0.000 0.000 0.084 0.000 0.000
[6,] 0.000 0.000 0.000 0.000 0.000 0.022 0.000
[7,] 0.000 0.000 0.000 0.000 0.000 0.000 0.011
```

Escores das componentes 1 e 2

Os scores de cada componente principal podem ser encontrados pela substituição dos valores de **z**. Serão apresentados apenas os scores das componentes Y_1 e Y_2.

```
> escores <- round(t(e) %*% t(z),2)
```

```
> plot(escores[2,] ~ escores[1,], cex = 0.7, main = "Gráfico dos escores da PC1 e PC2")
```

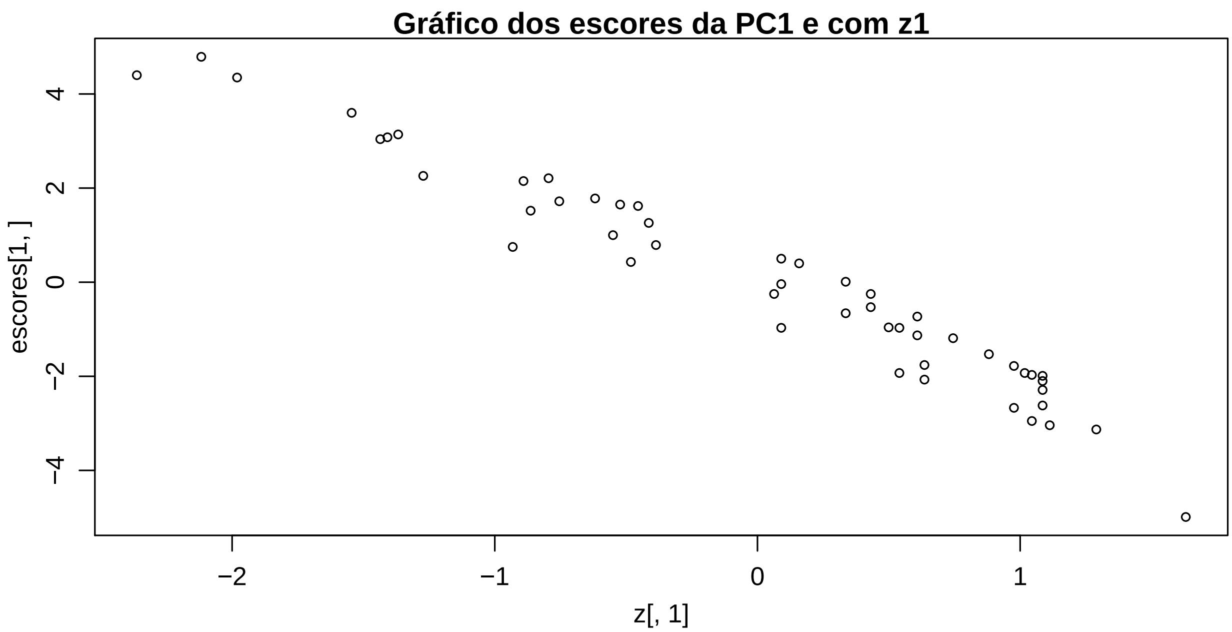


A correlação entre Y_k e X_i pode ser dada por:

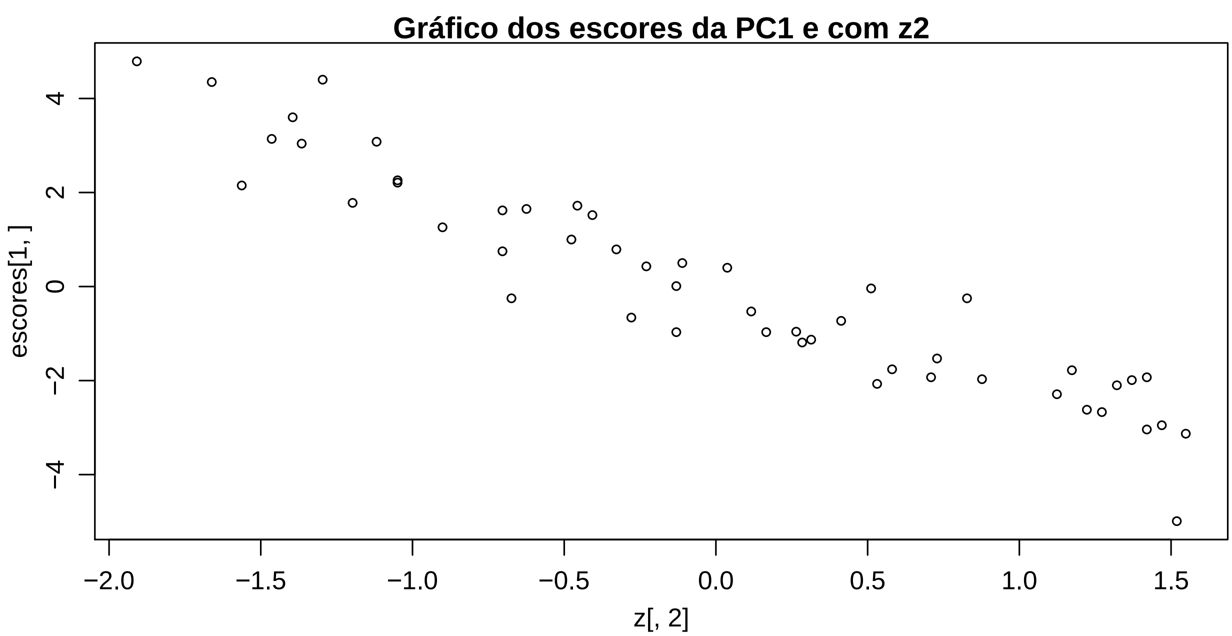
$$\rho(Y_k, X_i) = \frac{e_{ki}\sqrt{\lambda}_k}{r_{kk}} = e_{ki}\sqrt{\lambda}_k$$

```
> # Calculando a matriz de correlação entre Y[k] e as variáveis X[i]
> (corr.Y.com.X <- round(e %*%sqrt(V),2))

      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
[1,] -0.97  0.11  0.06 -0.03  0.18  0.05  0.06
[2,] -0.94 -0.02  0.31  0.01  0.00 -0.12  0.01
[3,] -0.94 -0.01 -0.15 -0.21 -0.21  0.02  0.04
[4,] -0.66 -0.65 -0.31 -0.20  0.08 -0.01 -0.03
[5,] -0.79 -0.28  0.00  0.55 -0.05  0.03 -0.01
[6,] -0.65  0.61 -0.43  0.10  0.03 -0.03 -0.02
[7,] -0.91  0.20  0.30 -0.17 -0.01  0.06 -0.07
```

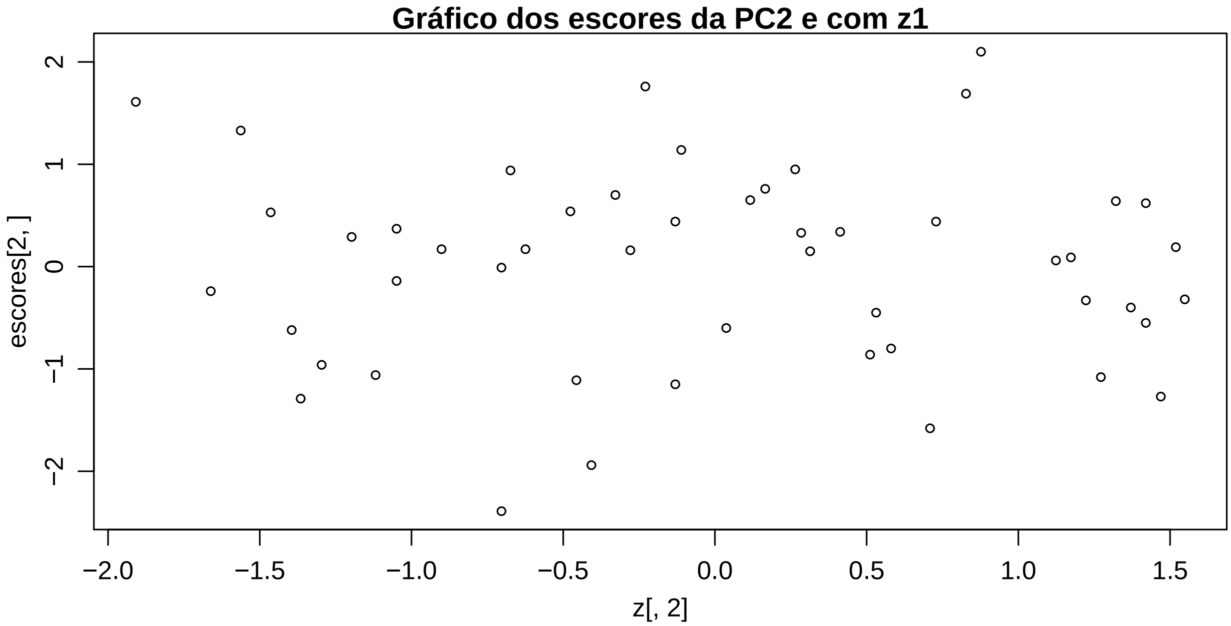
```
> plot(escores[1,] ~ z[,1], cex = 0.7, main = "Gráfico dos escores da PC1 e com z1")
```



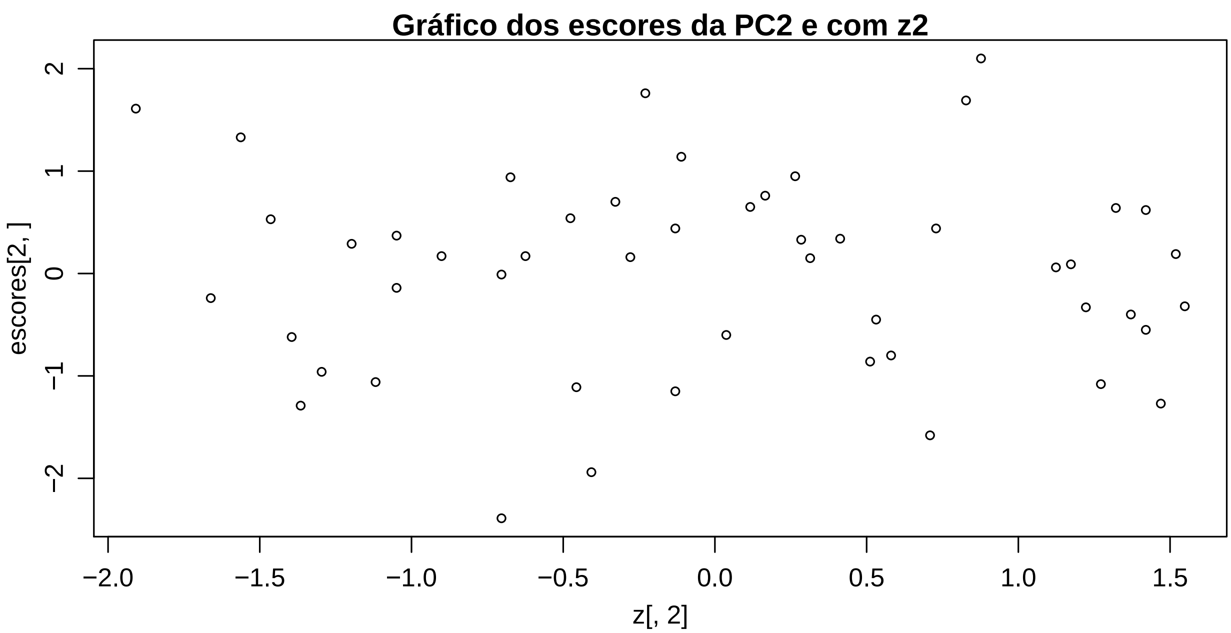

```
> plot(escores[1,] ~ z[,2], cex = 0.7, main = "Gráfico dos escores da PC1 e com z2")
```



```
> plot(escores[2,] ~ z[,2], cex = 0.7, main = "Gráfico dos escores da PC2 e com z1")
```




```
> plot(escores[2,] ~ z[,2], cex = 0.7, main = "Gráfico dos escores da PC2 e com z2")
```



1.2 Funções do R para ACP

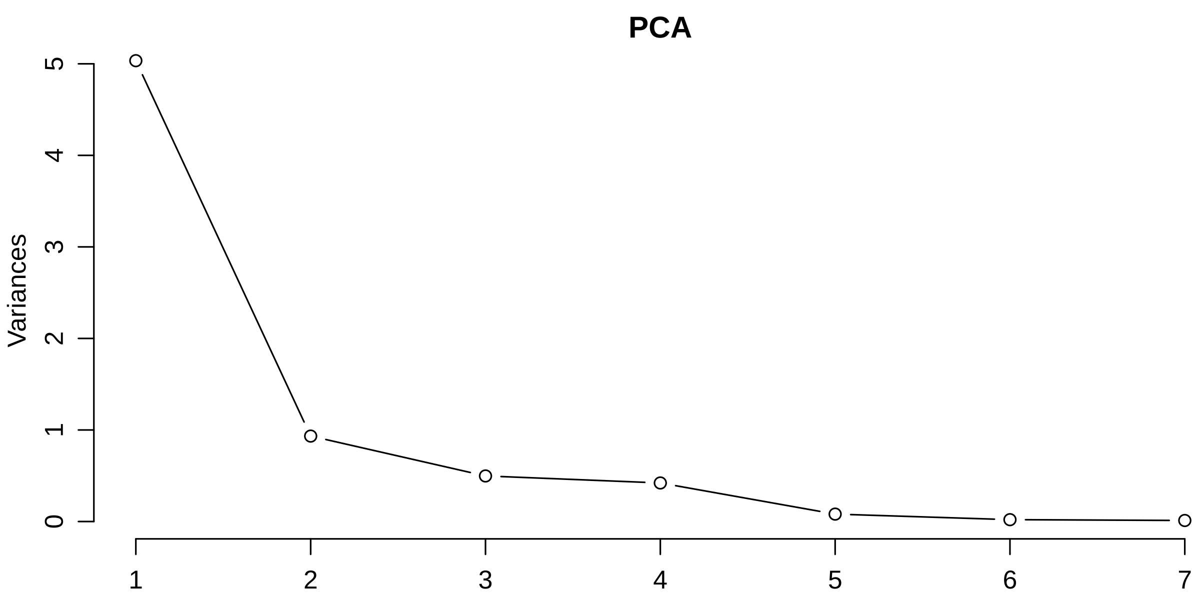
```
> args(prcomp)

function (x, ...)
NULL

> PCA <- prcomp(X, scale = TRUE)
> print(PCA)
```

```
Standard deviations:
[1] 2.2437909 0.9661864 0.7056343 0.6490343 0.2846760 0.1426206 0.1064883

Rotation:
          PC1          PC2          PC3         PC4           PC5        PC6
CV  0.4336719 -0.111754422  0.075488541  0.04237344 -0.6324942624  0.3365963
RV  0.4202136  0.029287495  0.442478953 -0.01075255  0.0001182093 -0.7853424
VNC 0.4210510  0.009201975 -0.204189315  0.32492838  0.7010262539  0.1568114
CR  0.2942863  0.668415809 -0.451492333  0.30271208 -0.2610080204 -0.1141710
RM  0.3490920  0.294944379 -0.005921773 -0.84660356  0.1742634819  0.1969091
RA  0.2891669 -0.642377957 -0.603779622 -0.15367411 -0.0869586057 -0.2362610
HM  0.4074041 -0.200367651  0.434039576  0.24601320  0.0495826418  0.3711105
            PC7
CV  -0.52782527
RV  -0.09948330
VNC -0.39916419
CR   0.29995962
RM   0.07231139
RA   0.22844351
HM   0.63622351
```

```
> summary(PCA) # dá a raiz dos autovalores
```

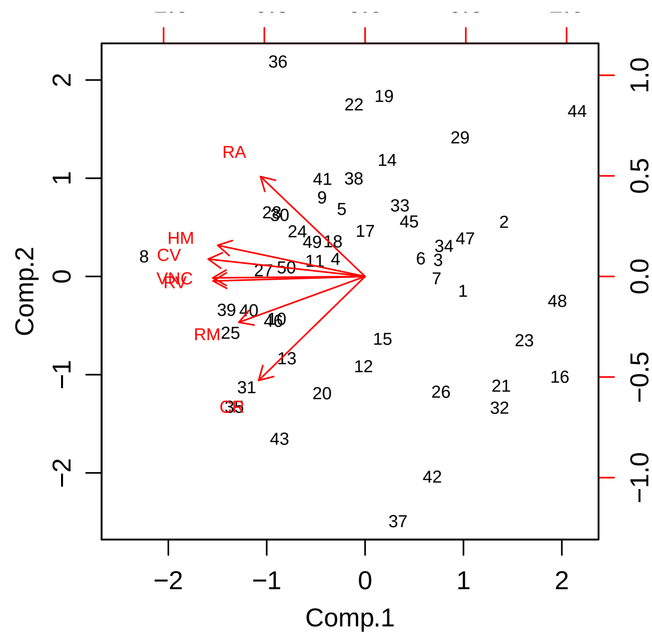
```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     2.2438 0.9662 0.70563 0.64903 0.28468 0.14262 0.10649
Proportion of Variance 0.7192 0.1334 0.07113 0.06018 0.01158 0.00291 0.00162
Cumulative Proportion  0.7192 0.8526 0.92372 0.98390 0.99547 0.99838 1.00000
```

```
> screeplot(PCA, type = c("lines"), main = deparse(substitute(PCA)))
```



Segundo Venables;Ripley(2002), o biplot é um método para representar ambos os casos e variáveis. Vamos supor que a matriz de dados tenha sido centralizada. O biplot representa a matriz de dados por meio de dois conjuntos de vetores de dimensão n e p produzindo um posto = 2 aproximando a matriz de dados. A interpretação é baseada no produto interno entre os vetores dos dois conjuntos.

```
> biplot(princomp(X, cor = T), pc.biplot = T, cex = 0.7, expand = 0.8)
```



2 Exercício: Heptatlo Olímpico de Seul (1988)

O pentatlo para mulheres foi realizado pela primeira vez na Alemanha, em 1928. Inicialmente isto consistia do arremesso de peso, salto em distância, 100m, salto em altura e eventos de lançamento de dardo realizaram durante dois dias. O pentatlo foi introduzido pela primeira vez em Jogos Olímpicos em 1964, em que ele consistia os 80 m com barreiras, tiros, salto em altura, salto em comprimento e 200 m. Em 1977, a 200 m foi substituído pelos 800 m e de 1981 a IAAF trouxe o heptatlo sete-evento no lugar do pentatlo, com um dia que contém os eventos-100 m barreiras, tiro, salto em altura, 200 m e dia dois, o salto em comprimento, lançamento de dardo e 800 m. Um sistema de pontuação é utilizado para atribuir pontos aos resultados de cada evento, e o vencedor é a mulher que acumula mais pontos durante os dois dias. O evento fez sua primeira aparição olímpica em 1984.
Nos Jogos Olímpicos de 1988, em Seul, o heptatlo foi vencido por uma das estrelas do atletismo feminino, nos EUA, Jackie Joyner-Kersee. Os resultados para todas as 25 concorrentes são dadas aqui.
O pacote ”HSAUR”contém os dados de 25 competidoras do heptatlo com 8 variáveis.

```
> # reportando dados
> (data("heptathlon", package = "HSAUR")); heptathlon
```

```
[1] "heptathlon"
```

	hurdles	highjump	shot	run200m	longjump	javelin	run800m	score
Joyner-Kersee (USA)	12.69	1.86	15.80	22.56	7.27	45.66	128.51	7291
John (GDR)	12.85	1.80	16.23	23.65	6.71	42.56	126.12	6897
Behmer (GDR)	13.20	1.83	14.20	23.10	6.68	44.54	124.20	6858
Sablovskaite (URS)	13.61	1.80	15.23	23.92	6.25	42.78	132.24	6540
Choubenkova (URS)	13.51	1.74	14.76	23.93	6.32	47.46	127.90	6540
Schulz (GDR)	13.75	1.83	13.50	24.65	6.33	42.82	125.79	6411
Fleming (AUS)	13.38	1.80	12.88	23.59	6.37	40.28	132.54	6351
Greiner (USA)	13.55	1.80	14.13	24.48	6.47	38.00	133.65	6297
Lajbnerova (CZE)	13.63	1.83	14.28	24.86	6.11	42.20	136.05	6252
Bouraga (URS)	13.25	1.77	12.62	23.59	6.28	39.06	134.74	6252
Wijnsma (HOL)	13.75	1.86	13.01	25.03	6.34	37.86	131.49	6205
Dimitrova (BUL)	13.24	1.80	12.88	23.59	6.37	40.28	132.54	6171

```
Scheider (SWI)        13.85     1.86 11.58   24.87     6.05   47.50  134.93  6137
Braun (FRG)           13.71     1.83 13.16   24.78     6.12   44.58  142.82  6109
Ruotsalainen (FIN)    13.79     1.80 12.32   24.61     6.08   45.44  137.06  6101
Yuping (CHN)          13.93     1.86 14.21   25.00     6.40   38.60  146.67  6087
Hagger (GB)           13.47     1.80 12.75   25.47     6.34   35.76  138.48  5975
Brown (USA)           14.07     1.83 12.69   24.83     6.13   44.34  146.43  5972
Mulliner (GB)         14.39     1.71 12.68   24.92     6.10   37.76  138.02  5746
Hautenauve (BEL)      14.04     1.77 11.81   25.61     5.99   35.68  133.90  5734
Kytola (FIN)          14.31     1.77 11.66   25.69     5.75   39.48  133.35  5686
Geremias (BRA)        14.23     1.71 12.95   25.50     5.50   39.64  144.02  5508
Hui-Ing (TAI)         14.85     1.68 10.00   25.23     5.47   39.14  137.30  5290
Jeong-Mi (KOR)        14.53     1.71 10.83   26.61     5.50   39.26  139.17  5289
Launa (PNG)           16.42     1.50 11.78   26.16     4.88   46.38  163.43  4566
```

Variáveis:

- hurdles: resultados de 100 m com barreiras
- highjump: resultados de salto em altura
- shot: resultados de arremesso de peso
- run200m: resultados de 200 m rasos
- longjump: resultados de salto em distância
- javelin: resultados de lançamento de dardos
- run800m: resultados de 800 m rasos
- score: pontuaçao total

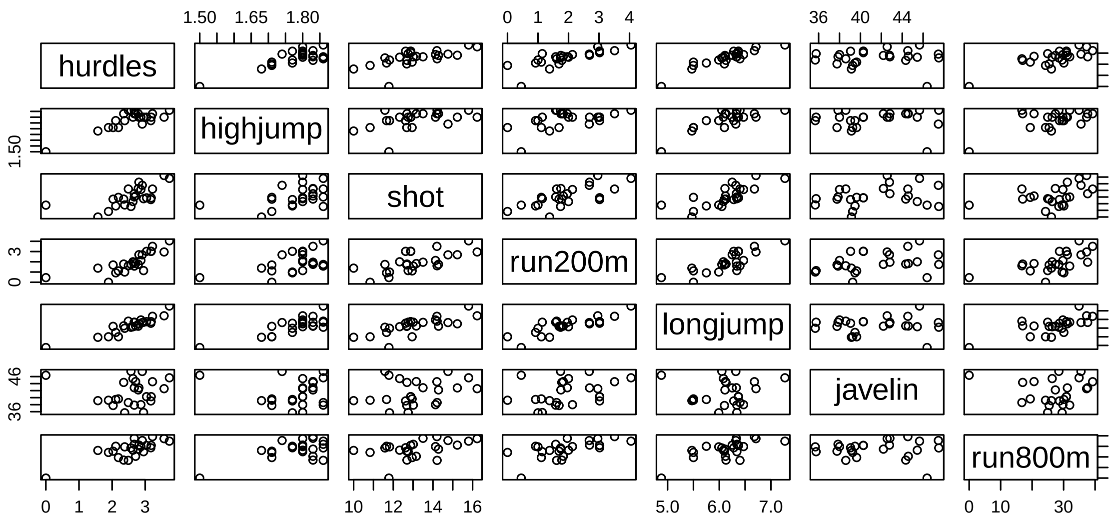
Será aplicado à esses dados a Análise de Componentes Principais visando a exploranção da estrutura dos dados e e avaliar como os escores das componentes principais se relacionam com os escores do sistema oficial de pontuação.
O objetivo básico da análise de componentes principais é descrever a variação em um conjunto de variáveis correlacionadas $x_1, x_2, \cdots, x_p$ em termos de um novo conjunto de variáveis não correlacionadas $y_1, y_2, \cdots, y_p$, as quais são combinações lineares das variáveis $x_i, \quad i = 1, \cdots, p$.
As novas variáveis são derivadas em ordem decrescente de "importância"no sentido que y_1 representa o máximo da variação nos dados originais entre todas as combinações lineares de $x_1, x_2, \cdots, x_p$.
Em seguida, y_2 é escolhido para representar tanto quanto possível da variaçao restante, sendo não correlacionado com y_1, e assim por diante, isto é, formando um sistema de coordenadas ortogonais. Nesse sentido, as novas variáveis $y_1, y_2, \cdots, y_p$ serão as componentes principais.
A esperança geral da análise de componentes principais é a de que as primeiras componentes serão responsáveis por uma proporção substancial da variação no original variáveis $x_1, x_2, \cdots, x_p$, e podem ser usados para fornecer um resumo de menor dimensão conveniente destas variáveis, que podem ser úteis para uma variedade de razões.
Em algumas aplicações, as componentes principais pode ser um fim em si e podem ser passíveis de interpretação de uma forma similar como os factores de uma análise factorial exploratória.
Observe as variáveis hurdles, run200m e run800m. Para estas variáveis, são bons os resultados menores, e, para as demais variáveis, são bons resultados maiores. Assim, será feita uma transformação na escala das variáveis hurdles, run200m e run800m (obtendo o máximo valor e subtraindo dele os resultados finais das competidoras) para ter a mesma direção das demais variáveis.

```
> heptathlon$hurdles <- max(heptathlon$hurdles)-heptathlon$hurdles
> heptathlon$run200m <- max(heptathlon$run200m)-heptathlon$run200m
> heptathlon$run800m <- max(heptathlon$run800m)-heptathlon$run800m
> head(heptathlon)
```

	hurdles	highjump	shot	run200m	longjump	javelin	run800m	score
Joyner-Kersee (USA)	3.73	1.86	15.80	4.05	7.27	45.66	34.92	7291
John (GDR)	3.57	1.80	16.23	2.96	6.71	42.56	37.31	6897
Behmer (GDR)	3.22	1.83	14.20	3.51	6.68	44.54	39.23	6858
Sablovskaite (URS)	2.81	1.80	15.23	2.69	6.25	42.78	31.19	6540
Choubenkova (URS)	2.91	1.74	14.76	2.68	6.32	47.46	35.53	6540
Schulz (GDR)	2.67	1.83	13.50	1.96	6.33	42.82	37.64	6411

O gráfico a seguir mostra uma matrix scatterplot dos resultados das 25 competidoras nos 7 eventos. Vemos que a maioria dos pares são relacionados para um grau maior ou menor. As exceções envolvem o evento dardo - este é o evento mais técnico.

```
> score <- which(colnames(heptathlon) == "score")
> plot(heptathlon[,-score])
```



Vamos agora encontrar a matriz de correlação do heptatlo:

```
> (R <- round(cor(heptathlon[, -score]), 2))
```

	hurdles	highjump	shot	run200m	longjump	javelin	run800m
hurdles	1.00	0.81	0.65	0.77	0.91	0.01	0.78
highjump	0.81	1.00	0.44	0.49	0.78	0.00	0.59
shot	0.65	0.44	1.00	0.68	0.74	0.27	0.42
run200m	0.77	0.49	0.68	1.00	0.82	0.33	0.62
longjump	0.91	0.78	0.74	0.82	1.00	0.07	0.70
javelin	0.01	0.00	0.27	0.33	0.07	1.00	-0.02
run800m	0.78	0.59	0.42	0.62	0.70	-0.02	1.00

Comparando o gráfico scatterplot e os valores da matriz de correlação, podemos observar as correlações entre as variáveis.
Vamos aplicar o teste de Bartlett para verificarmos as hipóteses $H_0 : R = I \ \times \ H_0 : R \neq I$.

```
> library(psych)
> cortest.bartlett(R, n = nrow(heptathlon))
```

```
$chisq
[1] 139.7356

$p.value
[1] 1.532748e-19

$df
[1] 21
```

Como o p-valor é praticamente 0, a matrix de correlação não é diagonal.

Iniciando a Análise de Componentes Principais.

```
> heptathlon_pca <- prcomp(heptathlon[, -score], scale = T)
> print(heptathlon_pca)

Standard deviations:
[1] 2.1119364 1.0928497 0.7218131 0.6761411 0.4952441 0.2701029 0.2213617

Rotation:
                PC1         PC2         PC3         PC4         PC5         PC6
hurdles  -0.4528710  0.15792058 -0.04514996  0.02653873 -0.09494792 -0.78334101
highjump -0.3771992  0.24807386 -0.36777902  0.67999172  0.01879888  0.09939981
shot     -0.3630725 -0.28940743  0.67618919  0.12431725  0.51165201 -0.05085983
run200m  -0.4078950 -0.26038545  0.08359211 -0.36106580 -0.64983404  0.02495639
longjump -0.4562318  0.05587394  0.13931653  0.11129249 -0.18429810  0.59020972
javelin  -0.0754090 -0.84169212 -0.47156016  0.12079924  0.13510669 -0.02724076
run800m  -0.3749594  0.22448984 -0.39585671 -0.60341130  0.50432116  0.15555520
                 PC7
hurdles   0.38024707
highjump -0.43393114
shot     -0.21762491
run200m  -0.45338483
longjump  0.61206388
javelin   0.17294667
run800m  -0.09830963

> summary(heptathlon_pca)

Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6    PC7
Standard deviation     2.1119 1.0928 0.72181 0.67614 0.49524 0.27010 0.2214
Proportion of Variance 0.6372 0.1706 0.07443 0.06531 0.03504 0.01042 0.0070
Cumulative Proportion  0.6372 0.8078 0.88223 0.94754 0.98258 0.99300 1.0000

> # Combinação linear da primeira componente principal
> (comp1 <- heptathlon_pca$rotation[,1])

   hurdles   highjump       shot    run200m   longjump    javelin    run800m
-0.4528710 -0.3771992 -0.3630725 -0.4078950 -0.4562318 -0.0754090 -0.3749594

> # Combinação linear da segunda componente principal
> (comp2 <- heptathlon_pca$rotation[,2])

    hurdles    highjump        shot     run200m    longjump     javelin     run800m
 0.15792058  0.24807386 -0.28940743 -0.26038545  0.05587394 -0.84169212  0.22448984
```

```
> # Combinação linear da terceira componente principal
> (comp3 <- heptathlon_pca$rotation[,3])

    hurdles    highjump        shot     run200m    longjump     javelin     run800m
-0.04514996 -0.36777902  0.67618919  0.08359211  0.13931653 -0.47156016 -0.39585671

> # Combinação linear da quarta componente principal
> (comp4 <- heptathlon_pca$rotation[,4])

    hurdles    highjump        shot     run200m    longjump     javelin     run800m
 0.02653873  0.67999172  0.12431725 -0.36106580  0.11129249  0.12079924 -0.60341130

> # Combinação linear da quinta componente principal
> (comp5 <- heptathlon_pca$rotation[,5])

    hurdles    highjump        shot     run200m    longjump     javelin     run800m
-0.09494792  0.01879888  0.51165201 -0.64983404 -0.18429810  0.13510669  0.50432116

> # Combinação linear da sexta componente principal
> (comp6 <- heptathlon_pca$rotation[,6])

    hurdles    highjump        shot     run200m    longjump     javelin     run800m
-0.78334101  0.09939981 -0.05085983  0.02495639  0.59020972 -0.02724076  0.15555520

> # Combinação linear da sétima componente principal
> (comp7 <- heptathlon_pca$rotation[,7])

    hurdles    highjump        shot     run200m    longjump     javelin     run800m
 0.38024707 -0.43393114 -0.21762491 -0.45338483  0.61206388  0.17294667 -0.09830963
```

Podemos observar que as competições de 200m e salto em distância receberam maior peso enquanto que os resultados de lançamento de dardo são menos importantes.
Para obter os escores das componentes principais é necessário deixar os dados numa escala apropriada, pois estamos utilizando a matriz de correlação. A centralizaçao e a padronizaçao usada por `prcomp` internamente podem ser extraídas de `heptathlon _ pca` com os comandos:

```
> center <- heptathlon_pca$center
> scale <- heptathlon_pca$scale
```

Vamos aplicar agora a função `scale` aos dados e multiplicar os carregamentos da matrix de modo a obter os escores da primeira componente pricipal para cada competidor.

```
> hm <- as.matrix(heptathlon[, -score])
> drop(scale(hm, center = center, scale = scale) %*% heptathlon_pca$rotation[,1])

Joyner-Kersee (USA)          John (GDR)        Behmer (GDR)  Sablovskaite (URS)
       -4.121447626        -2.882185935        -2.649633766        -1.343351210
  Choubenkova (URS)        Schulz (GDR)       Fleming (AUS)       Greiner (USA)
       -1.359025696        -1.043847471        -1.100385639        -0.923173639
   Lajbnerova (CZE)       Bouraga (URS)       Wijnsma (HOL)     Dimitrova (BUL)
       -0.530250689        -0.759819024        -0.556268302        -1.186453832
     Scheider (SWI)         Braun (FRG) Ruotsalainen (FIN)         Yuping (CHN)
        0.015461226         0.003774223         0.090747709        -0.137225440
        Hagger (GB)         Brown (USA)        Mulliner (GB)    Hautenauve (BEL)
        0.171128651         0.519252646         1.125481833         1.085697646
       Kytola (FIN)      Geremias (BRA)       Hui-Ing (TAI)      Jeong-Mi (KOR)
        1.447055499         2.014029620         2.880298635         2.970118607
        Launa (PNG)
        6.270021972
```

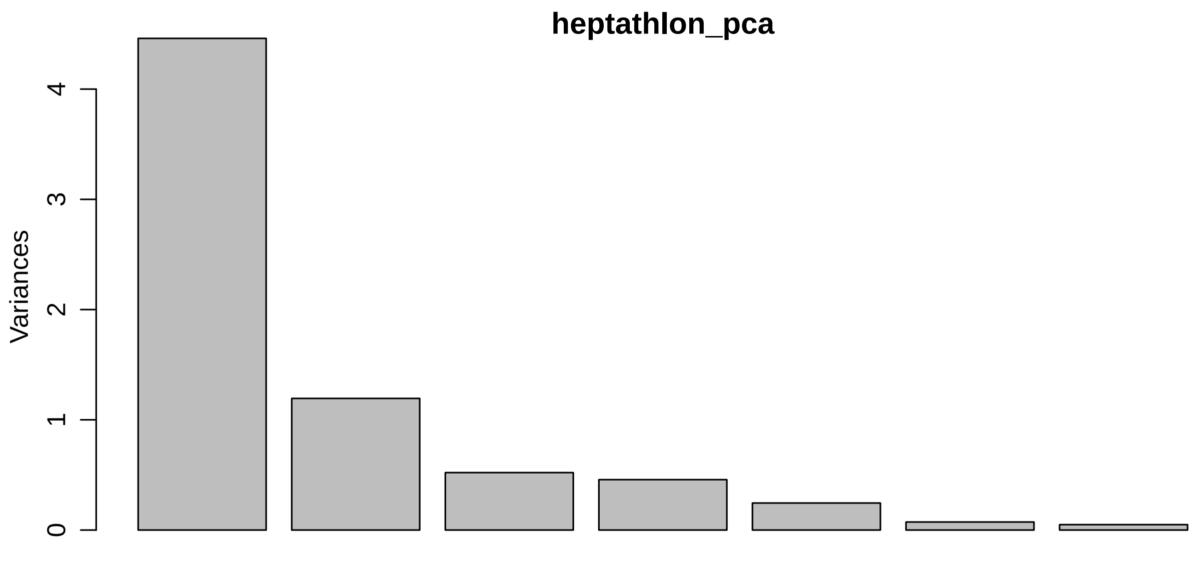
```
> predict(heptathlon_pca)[,1]

Joyner-Kersee (USA)          John (GDR)        Behmer (GDR)  Sablovskaite (URS)
       -4.121447626        -2.882185935        -2.649633766        -1.343351210
  Choubenkova (URS)        Schulz (GDR)       Fleming (AUS)        Greiner (USA)
       -1.359025696        -1.043847471        -1.100385639        -0.923173639
   Lajbnerova (CZE)       Bouraga (URS)       Wijnsma (HOL)     Dimitrova (BUL)
       -0.530250689        -0.759819024        -0.556268302        -1.186453832
     Scheider (SWI)        Braun (FRG)  Ruotsalainen (FIN)         Yuping (CHN)
        0.015461226         0.003774223         0.090747709        -0.137225440
        Hagger (GB)        Brown (USA)       Mulliner (GB)    Hautenauve (BEL)
        0.171128651         0.519252646         1.125481833         1.085697646
       Kytola (FIN)      Geremias (BRA)       Hui-Ing (TAI)      Jeong-Mi (KOR)
        1.447055499         2.014029620         2.880298635         2.970118607
        Launa (PNG)
        6.270021972
```

Isto pode ser feito ainda pelo seguinte comando:


```
> predict(heptathlon_pca)[,1]

Joyner-Kersee (USA)          John (GDR)        Behmer (GDR)  Sablovskaite (URS)
       -4.121447626        -2.882185935        -2.649633766        -1.343351210
  Choubenkova (URS)        Schulz (GDR)       Fleming (AUS)        Greiner (USA)
       -1.359025696        -1.043847471        -1.100385639        -0.923173639
   Lajbnerova (CZE)       Bouraga (URS)       Wijnsma (HOL)     Dimitrova (BUL)
       -0.530250689        -0.759819024        -0.556268302        -1.186453832
     Scheider (SWI)        Braun (FRG)  Ruotsalainen (FIN)         Yuping (CHN)
        0.015461226         0.003774223         0.090747709        -0.137225440
        Hagger (GB)        Brown (USA)       Mulliner (GB)    Hautenauve (BEL)
        0.171128651         0.519252646         1.125481833         1.085697646
       Kytola (FIN)      Geremias (BRA)       Hui-Ing (TAI)      Jeong-Mi (KOR)
        1.447055499         2.014029620         2.880298635         2.970118607
        Launa (PNG)
        6.270021972
```

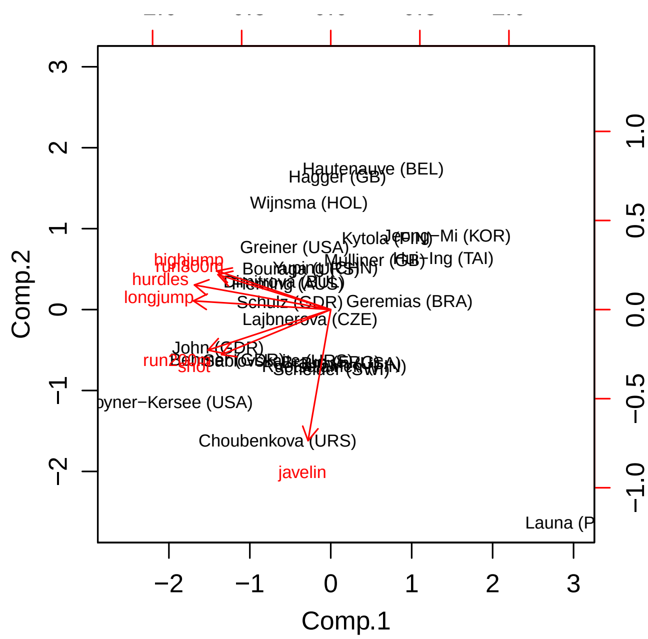

O gráfico a seguir representa os autovalores, os quais são entendidos como variâncias explicadas pelas componentes principais.

> *plot(heptathlon_pca)*



A primeira componente principal representa 81% da variância. O gráfico de barras da variância das componentes principais mostra o quanto de variância as duas primeiras componentes dominam. O gráfico dos dados no espaço das duas primeiras componentes principais, com pontos nomeados pelos nomes das competidoras, será mostrado a seguir:

> *biplot(princomp(heptathlon[,-score], cor = T), pc.biplot = T, cex = 0.7, expand = 0.8)*



Os dois primeiros carregamentos dos eventos são dados no sistema de coordenadas de segunda dimensão, e ilustram o papel principal do evento lançamento de dardos.
A correlação entre os escores dados para cada atleta pela sistema de escores usados pelo heptatlo e os escores das primeiras componentes principais pode ser encontrados com:

```
> cor(heptathlon$score, heptathlon_pca$x[,1])
```

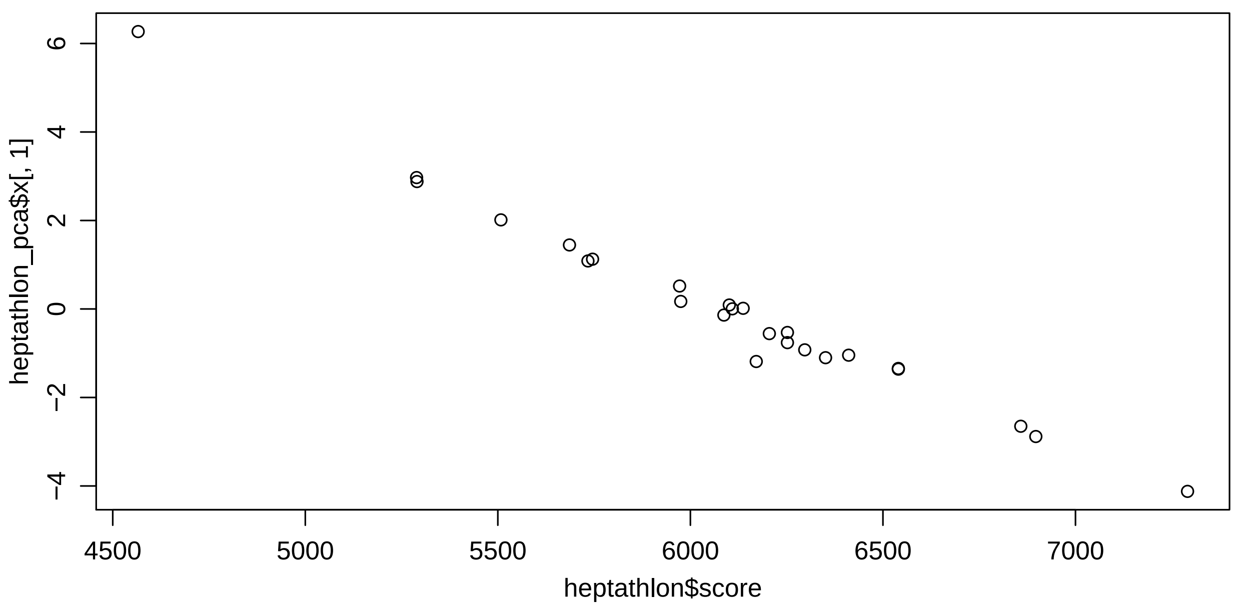
```
[1] -0.9910978
```

```
> cor(heptathlon$score, heptathlon_pca$x[,2])
```

```
[1] -0.09788578
```

Isto significa que as primeiras componentes principais são concordantes com os escores obtidos dos atletas pelas regras Olímpicas, veja os seguintes gráficos:

```
> plot(heptathlon$score, heptathlon_pca$x[,1])
```



```
> plot(heptathlon$score, heptathlon_pca$x[,2])
```