Varianza totale e generalizzata

Si consideri la seguente matrice di varianze/covarianze $\underset{2\times 2}{S} = \left[\begin{array}{cc} 2.2 & 0.4 \\ 0.4 & 2.8 \end{array}\right]$.

```
S <- matrix(c(2.2, 0.4, 0.4, 2.8),nrow=2,ncol=2)
S
```

```
##      [,1] [,2]
## [1,]  2.2  0.4
## [2,]  0.4  2.8
```

1. Si calcoli la varianza totale, la varianza generalizzata e l'indice realtivo di variabilità.

```
( S <- matrix(c(2.2, 0.4, 0.4, 2.8),nrow=2,ncol=2) )
```

```
##      [,1] [,2]
## [1,]  2.2  0.4
## [2,]  0.4  2.8
```

```
# varianza totale
( vartot = sum( diag(S) ) )
```

```
## [1] 5
```

```
# varianza generalizzata
( vargen = det( S ) )
```

```
## [1] 6
```

```
# indice relativo di variabilità
 ( ir = vargen / prod( diag(S) ) )
```

```
## [1] 0.974026
```

```
R = diag(diag(S)^(-1/2)) %*% S %*% diag(diag(S)^(-1/2))
det( R )
```

```
## [1] 0.974026
```

2. Si calcolino gli autovalori λ_1 e λ_2 e gli autovettori normalizzati $\underset{2\times 1}{v_1}$ e $\underset{2\times 1}{v_2}$ di $\underset{2\times 2}{S}$, verificando che gli autovettori normalizzati hanno lunghezza unitaria e sono ortogonali

```
eigen <- eigen(S)

# autovalori
( lambda1 <- eigen$values[1] )
```

```
## [1] 3
```

```
( lambda2 <- eigen$values[2] )
```

```
## [1] 2
```

```
# autovettori (normalizzati)
( v1 <- eigen$vectors[,1, drop=F] )
```

```
##           [,1]
## [1,] 0.4472136
## [2,] 0.8944272
```

```r
( v2 <- eigen$vectors[,2, drop=F] )
```

```
##            [,1]
## [1,] -0.8944272
## [2,]  0.4472136
```

```r
# verifico che gli autovettori sono di lunghezza unitaria
t(v1) %*% v1
```

```
##      [,1]
## [1,]    1
```

```r
t(v2) %*% v2
```

```
##      [,1]
## [1,]    1
```

```r
# verifico che gli autovettori sono ortogonali
t(v1) %*% v2
```

```
##      [,1]
## [1,]    0
```

```r
t(v2) %*% v1
```

```
##      [,1]
## [1,]    0
```

Dati Animals

1. Caricare il data set `Animals`, presente nella libreria `MASS`. Calcolare la matrice di varianze/covarianze S considerandi le variabili trasformate al logaritmo `log(brain)` e `log(body)`, la varianza totale di S, la varianza generalizzata di S e l'indice relativo di variabilità.

```r
rm(list=ls())
library(MASS)
data(Animals)
X = log(Animals)
n = nrow(X)
( S = var(X) * (n-1)/n )
```

```
##            body    brain
## body  13.710061 6.800118
## brain  6.800118 5.550965
```

```r
# varianza totale
( vartot = sum( diag(S) ) )
```

```
## [1] 19.26103
```

```r
# varianza generalizzata
( vargen = det( S ) )
```

```
## [1] 29.86247
```

```r
# indice relativo di variabilità
 ( ir = vargen / prod( diag(S) ) )
```

```
## [1] 0.3923899
```

2. Calcolare gli autovalori λ_1 e λ_2 e gli autovettori v_1 e v_2 di S.

```
eigen <- eigen(S)

# autovalori
( lambda1 <- eigen$values[1] )
```

```
## [1] 17.56048
```

```
( lambda2 <- eigen$values[2] )
```

```
## [1] 1.70055
```

```
# autovettori (normalizzati)
( v1 <- eigen$vectors[,1, drop=F] )
```

```
##            [,1]
## [1,] -0.8701860
## [2,] -0.4927234
```

```
( v2 <- eigen$vectors[,2, drop=F] )
```

```
##            [,1]
## [1,]  0.4927234
## [2,] -0.8701860
```

3. Costruire il diagramma di dispersione per `log(brain)` e `log(body)`. Aggiungere al grafico il baricentro $\bar{x}'$ con il comando `points` specificando `pch=19`.

Utilizzando la libreria `ellipse`, aggiungere al diagramma di dispersione l'ellisse (comando `ellipse()`) specificando l'argomento `x` pari a S, `centre` pari al baricentro e l'argomento `t` pari a 2.

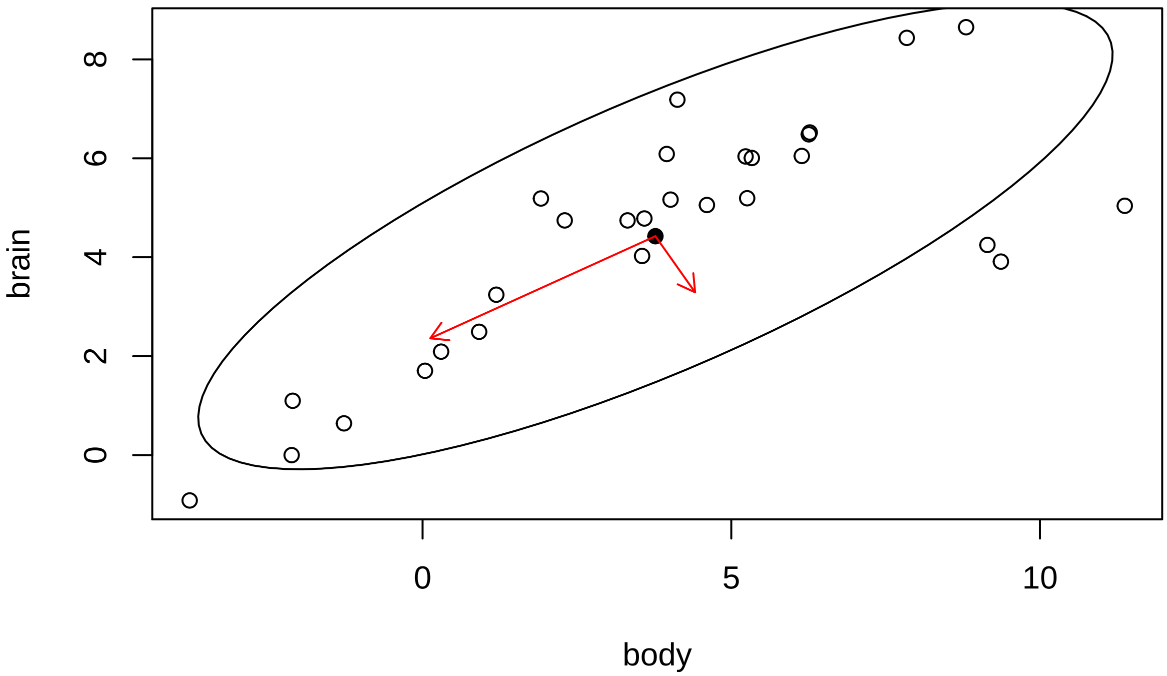
Aggiungere gli assi dell'ellisse centrata sul baricentro con il comando `arrows()`, la cui direzione è determinata dai due autovettori e la cui lunghezza è proporzionale alla radice quadrata dei due autovalori.

```
# diagramma dispersione
plot(X)
# baricentro
( bc = colMeans(X) )
```

```
##     body    brain
## 3.771306 4.425446
```

```
points(bc[1],bc[2], pch=19)

# elisse
library(ellipse)
lines(ellipse(x=S,centre = bc, t = 2))
arrows(x0 = bc[1], y0 = bc[2], x1 = sqrt(lambda1)*v1[1] + bc[1], y1=sqrt(lambda1)*v1[2] + bc[2], length
arrows(x0 = bc[1], y0 = bc[2], x1 = sqrt(lambda2)*v2[1] + bc[1], y1=sqrt(lambda2)*v2[2] + bc[2], length
```


brain
8
6
4
2
0
0
5
10
body