Decomposizione Spettrale

1. Si consideri la seguente matrice di varianze/covarianze $\underset{2\times 2}{S} = \begin{bmatrix} 2.2 & 0.4 \\ 0.4 & 2.8 \end{bmatrix}$. Decomporre la matrice S in $\underset{2\times 2}{V}\underset{2\times 2}{\Lambda}\underset{2\times 2}{V'}$, verificando che $S = V\Lambda V'$ e che V è una matrice ortogonale $VV' = V'V = \underset{2\times 2}{I}$.

```
( 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
```

```
# calcolo autovalori e autovettori di S
eigen <- eigen(S)
# Lambda
( Lambda = diag(eigen$values) )
```

```
##      [,1] [,2]
## [1,]    3    0
## [2,]    0    2
```

```
# V
V = eigen$vectors
colnames(V) = c("v1","v2")
V
```

```
##             v1         v2
## [1,] 0.4472136 -0.8944272
## [2,] 0.8944272  0.4472136
```

```
# verifico la decomposizione spettrale
round(
S - V %*% Lambda %*% t(V)
, 8)
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0    0
```

```
# verifico che V è ortogonale
V %*% t(V)  - diag(c(1,1))
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0    0
```

```
# t(V) %*% V - diag(c(1,1))
```

2. Calcolare $\underset{2\times 2}{S^{1/2}} = \underset{2\times 2}{V}\underset{2\times 2}{\Lambda^{1/2}}\underset{2\times 2}{V'}$ e $\underset{2\times 2}{S^{-1}} = \underset{2\times 2}{V}\underset{2\times 2}{\Lambda^{-1}}\underset{2\times 2}{V'}$ e verificare che $\underset{2\times 2}{S^{-1}}\underset{2\times 2}{S} = \underset{2\times 2}{I}$

```
# S^(1/2)
( SqrtS = V %*% Lambda^(1/2) %*% t(V) )
```

```
##           [,1]      [,2]
## [1,] 1.4777810 0.1271349
## [2,] 0.1271349 1.6684834
```

```r
# S^(-1)
( InvS = V %*% diag( 1/diag(Lambda) ) %*% t(V) )
```

```
##             [,1]        [,2]
## [1,]  0.46666667 -0.06666667
## [2,] -0.06666667  0.36666667
```

```r
# verifico che è l'inversa di S
round(
InvS %*% S
, 8)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

3. Verificare che la varianza totale di S è la somma degli autovalori di S e che la varianza generalizzata di S è il prodotto degli autovalori di S

```r
# varianza totale di S
sum(diag(Lambda)) # coincide con sum(diag(S))
```

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

```r
# varianza generalizzata di S
prod(diag(Lambda)) # coincide con det(S)
```

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

Matrice dei dati ortogonalizzati

1. Si consideri la seguente matrice `X` di dimensioni 10×2

```r
rm(list=ls())
( X <- matrix(c(2,3,3,4,4,5,6,6,7,8,7,8,10,6,8,10,12,13,11,12),nrow=10,ncol=2) )
```

```
##       [,1] [,2]
##  [1,]    2    7
##  [2,]    3    8
##  [3,]    3   10
##  [4,]    4    6
##  [5,]    4    8
##  [6,]    5   10
##  [7,]    6   12
##  [8,]    6   13
##  [9,]    7   11
## [10,]    8   12
```

```r
n <- nrow(X)
p <- ncol(X)
```

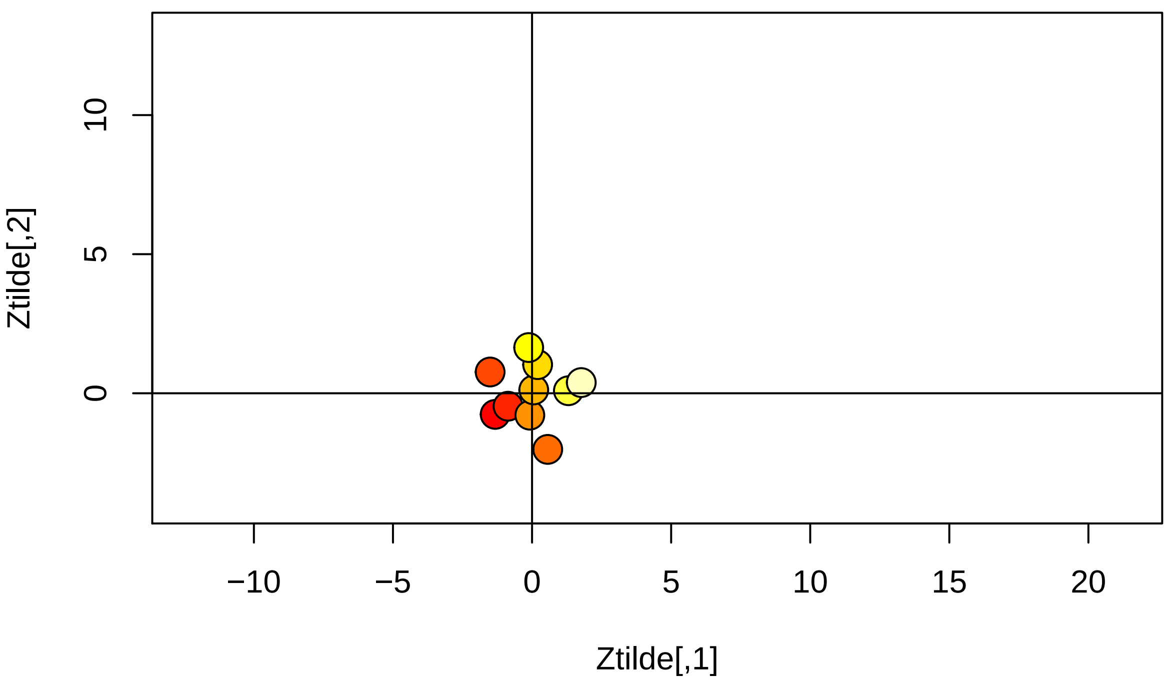
Ottenere la matrice di dati ortogonalizzati $\tilde{Z} = \tilde{X}S^{-1/2}$, e visualizzarla con il diagramma di dispersione. Verificare che per i dati ortogonalizzati il vettore delle medie è nullo e che la matrice di varianze/covarianze è la matrice identità

```r
one.n <- matrix(rep(1,n),ncol=1)
I.n <- diag(rep(1,n))
H <- I.n - (1/n) * one.n %*% t(one.n)
```

```r
Xtilde <- H %*% X

# S
S = (1/n)* t(Xtilde) %*% Xtilde

# Decomposizione Spettrale
eigen = eigen(S)
Lambda = diag(eigen$values)
V = eigen$vectors


# S^(1/2)
( InvSqrtS = V %*% diag(diag(Lambda)^(-.5)) %*% t(V) )
```

```
##            [,1]       [,2]
## [1,]  0.7841063 -0.3218073
## [2,] -0.3218073  0.6150037
```

```r
# Dati ortogonalizzati
Ztilde = Xtilde %*% InvSqrtS

plot(Ztilde, xlim=c(-4,13),ylim=c(-4,13),
     bg=heat.colors(n),pch=21,cex=2,asp=1)
abline(h=0)
abline(v=0)
```




```r
# vettore delle medie
round(
(1/n) * t(Ztilde) %*% one.n
,8)
```

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

```
## [2,]    0
```

```r
# matrice di varianze e covarianze
round(
(1/n)* t(H %*% Ztilde)%*% (H %*% Ztilde)
, 8)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

Decomposizione in Valori Singolari

1. Decomporre la matrice $\underset{10\times 2}{\tilde{X}}$ ottenuta nell'esercizio precedente in valori singolari $\underset{10\times 2}{U_r}\,\underset{2\times 2}{\Delta_r}\,\underset{2\times 2}{V_r'}$, verificando che $\tilde{X} = U_r \Delta_r V_r'$ dove $r = \text{rango}(\tilde{X})$.

```r
# rango di Xtilde
( r = qr(Xtilde)$rank )
```

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

```r
# SVD di Xtilde
SVD=svd(Xtilde)

( Ur = SVD$u )
```

```
##              [,1]         [,2]
##  [1,] -0.44636750  0.185536968
##  [2,] -0.28366992  0.126393478
##  [3,] -0.09995551  0.525097399
##  [4,] -0.39654396 -0.530805894
##  [5,] -0.21282955 -0.132101973
##  [6,]  0.04172524  0.008106498
##  [7,]  0.29628002  0.148314969
##  [8,]  0.38813722  0.347666929
##  [9,]  0.27526319 -0.309532443
## [10,]  0.43796076 -0.368675933
```

```r
( Vr = SVD$v )
```

```
##           [,1]       [,2]
## [1,] 0.6106905 -0.7918694
## [2,] 0.7918694  0.6106905
```

```r
( Deltar = diag(SVD$d) )
```

```
##          [,1]     [,2]
## [1,] 8.620656 0.000000
## [2,] 0.000000 3.063378
```

```r
# verifico SVD
round( Xtilde - Ur %*% Deltar %*% t(Vr) , 8)
```

```
##       [,1] [,2]
##  [1,]    0    0
##  [2,]    0    0
##  [3,]    0    0
```

```
##   [4,]    0    0
##   [5,]    0    0
##   [6,]    0    0
##   [7,]    0    0
##   [8,]    0    0
##   [9,]    0    0
##  [10,]    0    0
```

2. Verificare che V_r e V relativo a $S = V\Lambda V'$ e V_r coincidono.

```
# verifico che Vr = V
round(
Vr - V,
8)
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0    0
```

3. Verificare che U_r sono i primi r autovettori di $\tilde{X}\tilde{X}'$, V_r sono i primi r autovettori di $\tilde{X}'\tilde{X}$ e che Δ_r contiene la radice quadrata dei primi r autovalori di $\tilde{X}'\tilde{X}$ (o di $\tilde{X}\tilde{X}'$)

```
eigen( Xtilde %*% t(Xtilde) )$vectors[,1:r]
```

```
##              [,1]         [,2]
##  [1,] -0.44636750 -0.185536968
##  [2,] -0.28366992 -0.126393478
##  [3,] -0.09995551 -0.525097399
##  [4,] -0.39654396  0.530805894
##  [5,] -0.21282955  0.132101973
##  [6,]  0.04172524 -0.008106498
##  [7,]  0.29628002 -0.148314969
##  [8,]  0.38813722 -0.347666929
##  [9,]  0.27526319  0.309532443
## [10,]  0.43796076  0.368675933
```

```
eigen( t(Xtilde) %*% Xtilde )$vectors[,1:r]
```

```
##           [,1]       [,2]
## [1,] 0.6106905 -0.7918694
## [2,] 0.7918694  0.6106905
```

```
sqrt( eigen(Xtilde %*% t(Xtilde))$values[1:r] )
```

```
## [1] 8.620656 3.063378
```