Problem 3.1

Chen Bo Calvin Zhang

19/10/2020

We want to use an example to check that the mean and variance obtained after applying the Mahalanobis transformation is consistent with the results we have obtained in the notes. In particular we want to check that

$$E(\boldsymbol{x}) = \boldsymbol{\mu} \text{ and } Var(\boldsymbol{x}) = \boldsymbol{\Sigma} \implies E(\boldsymbol{y}) = \boldsymbol{0} \text{ and } Var(\boldsymbol{y}) = \boldsymbol{I}$$

where

$$\boldsymbol{y} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x} - \boldsymbol{\mu})$$

is the Mahalanobis transformation on a vector $\boldsymbol{x}$.

Firstly, let us create the data matrix

$$\boldsymbol{X} = \begin{pmatrix} 4 & 3 \\ -1 & 3 \\ 3 & 5 \end{pmatrix}$$

```r
X = matrix(c(4, -1, 3, 1, 3, 5), ncol=2)
print(X)
```

```
##      [,1] [,2]
## [1,]    4    1
## [2,]   -1    3
## [3,]    3    5
```

Now compute the column means and covariance matrix.

```r
mu = colMeans(X)
print(mu)
```

```
## [1] 2 3
```

```r
Sigma = cov(X)
print(Sigma)
```

```
##      [,1] [,2]
## [1,]    7   -1
## [2,]   -1    4
```

In matrix notation, we can write

$$\boldsymbol{Y} = \boldsymbol{C}_n \boldsymbol{X} \boldsymbol{\Sigma}^{-\frac{1}{2}}$$

where $\boldsymbol{C}_n$ is the centering matrix.

Now, let us find the inverse square root of $\boldsymbol{\Sigma}$ using the eigenvalue decomposition.

```
eigen = eigen(Sigma)
lambda = eigen$values
print(lambda)
```

```
## [1] 7.302776 3.697224
```

```
U = eigen$vectors
print(U)
```

```
##            [,1]       [,2]
## [1,] -0.9570920 -0.2897841
## [2,]  0.2897841 -0.9570920
```

```
inv_sqrt_Sigma = U %*% diag(lambda^(-1/2)) %*% t(U)
print(inv_sqrt_Sigma)
```

```
##           [,1]      [,2]
## [1,] 0.3826445 0.0416092
## [2,] 0.0416092 0.5074721
```

Finally, compute the Mahalanobis transformation and check that the column means and the covariance matrix are as expected.

```
# sweep() function used to centre the matrix X, 2 means that we are working column-wise
Y = sweep(X, 2, mu) %*% inv_sqrt_Sigma
print(Y)
```

```
##            [,1]       [,2]
## [1,]  0.6820706 -0.9317258
## [2,] -1.1479335 -0.1248276
## [3,]  0.4658629  1.0565534
```

```
# check that E(Y) = 0 matrix and Var(Y) = I
print(colMeans(Y))
```

```
## [1]  3.700743e-17 -3.238150e-17
```

```
print(cov(Y))
```

```
##               [,1]          [,2]
## [1,]  1.000000e+00 -9.778148e-17
## [2,] -9.778148e-17  1.000000e+00
```

As we can see, the mean is very close to zero for both columns and the covariance matrix is the identity matrix. The difference is due to floating point approximation.