Linear models list 1

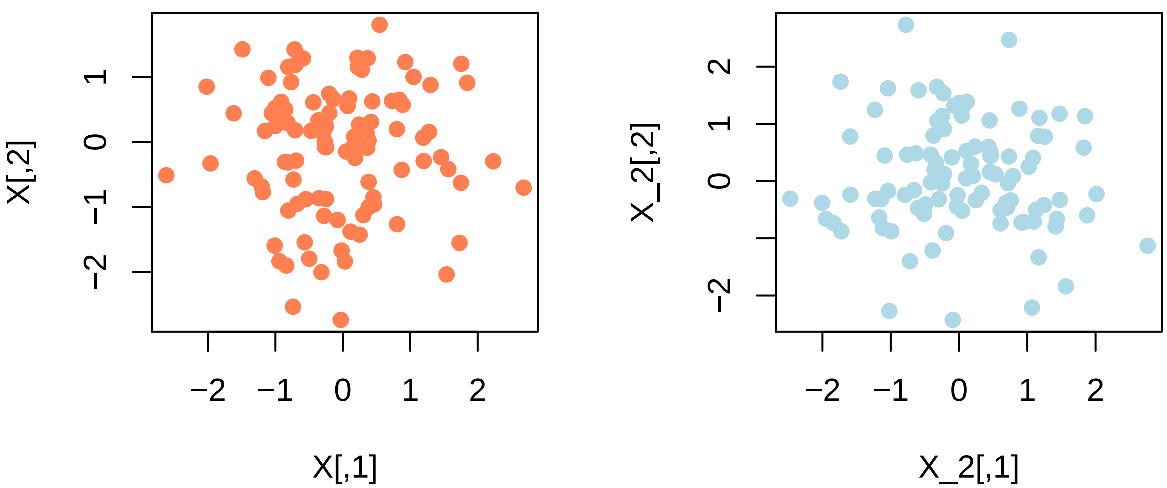
Klaudia Weigel

1 Exercise 1

Use a rnorm function (in R) to generate 100 random vectors from a two dimensional normal distribution $N(0; I)$ *and plot them.*

We will generate 200 random variables from the standard normal distribution and divide them into two columns. We can compare the results with a built in function `rmvnorm` from the *mvtnorm* library.

```
set.seed(9)
par(mfrow = c(1,2))
X = matrix(rnorm(100 * 2, 0, 1), ncol = 2)
plot(X, pch = 19, col = 'coral')
X_2 = rmvnorm(100, mean=c(0,0), sigma = diag(2))
plot(X_2, pch = 19, col = 'lightblue')
```



2 Exercise 2

Find an affine transformation, which transforms above cloud of points into a cloud of points from a normal distribution $N(\mu, \Sigma)$*.*

Affine transformation has the form $Y = AX + B$.

Theorem 1 *Suppose* $X \in \mathbb{R}^n$ *has a multivariate normal distribution* $N_n(\mu, \Sigma)$*. Let* $Y = AX + b$*, where* A *is a* $m \times n$ *matrix and* $b \in \mathbb{R}^m$*. Then* Y *has a* $N_m(A\mu + b, A\Sigma A^T)$ *distribution.*

Definition 1 *A matrix* $A \in \mathbb{R}^{n\times n}$ *is symmetric positive-definite if and only if it is symmetric* ($A = A^T$) *and for every nonzero vector* $x \in \mathbb{R}^n$ *it is true that* $x^T Ax > 0$*.*

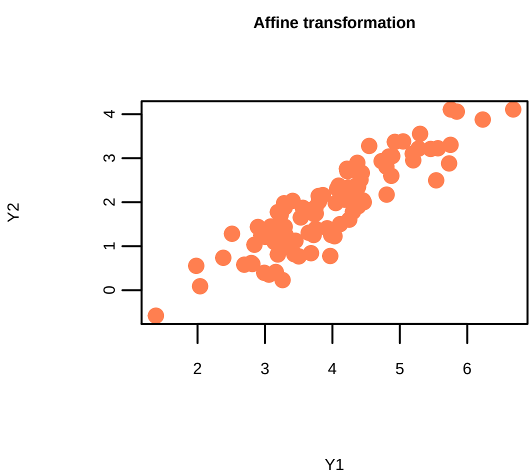
Theorem 2 (Cholesky Factorization Theorem) *Given a symmetric positive definite matrix A there exists a lower triangular matrix L such that $A = LL^T$. Such a decomposition is unique if the diagonal elements of L are restricted to be positive.*

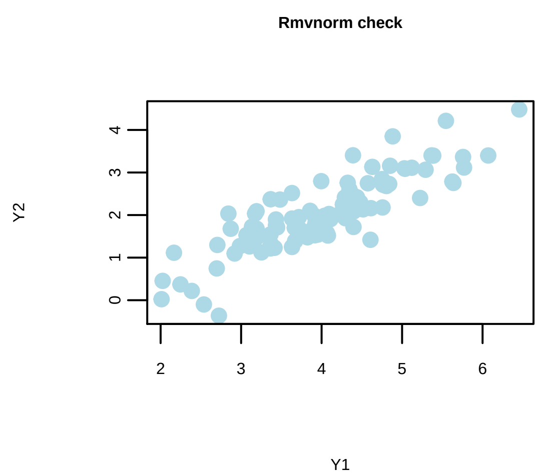
A covariance matrix is always semi positive definite ($x^T Ax \geq 0$) and is positive define if all the columns of the matrix are linearly independent, which is the case in our exercise.

In the exercise we have $\mu = A \cdot 0 + b = b$ and $\Sigma = AIA^T = AA^T$. The second equation is satisfied by the Cholesky factor, which we can find with a built in function `chol`.

a) $\mu = (4, 2)$, $\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$.

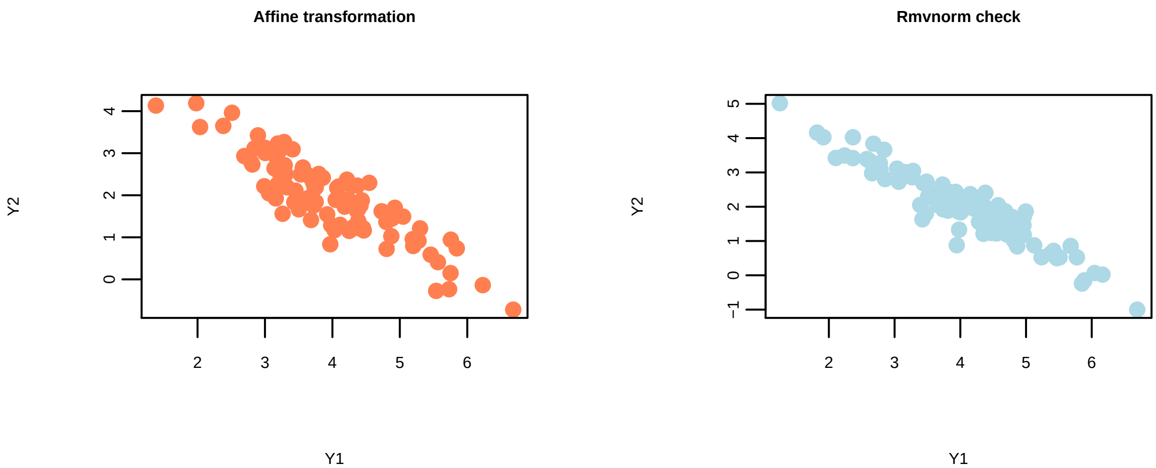
```
affine_transformation =  function(X, A, b) {
  return(A %*% X + b)
}

apply_and_plot = function(X, Sigma, b) {
  par(mfrow = c(1,2), mgp=c(3,0.5,0),mar=c(5,4,4,2)+0.1)
  # chol finds an upper traingular matrix, so we have to transpose it
  A = t(chol(Sigma)) # Sigma = AA^T

  # Apply an affine transformation to the original cloud of points
  Y = t(apply(X, MARGIN = 1, FUN = affine_transformation, A = A, b = b))
  plot(Y, xlab = "Y1", ylab = "Y2", pch=19,
       cex.main = 0.5, cex.lab = 0.5, cex.axis = 0.5,
       col="coral", main = "Affine transformation")

  # Check the results using the rmvnorm function
  Y_check = rmvnorm(n=100, mean = b, sigma = Sigma)
  plot(Y_check, xlab = "Y1", ylab = "Y2", pch=19,
       cex.main = 0.5, cex.lab = 0.5, cex.axis = 0.5,
       col="lightblue", main = "Rmvnorm check")
}

S_a = matrix(c(1, 0.9,  0.9, 1), nrow = 2, byrow = TRUE)
b = c(4, 2)
apply_and_plot(X, S_a, b)
```





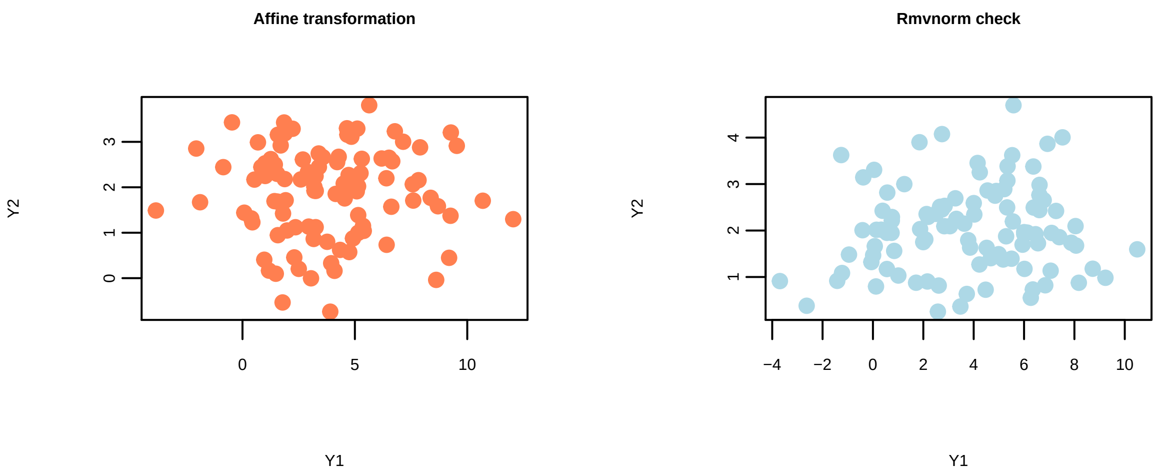
b) $\mu = (4, 2)$, $\Sigma = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}$.

```
S_b = matrix(c(1, -0.9,  -0.9, 1), nrow = 2, byrow = TRUE)
b = c(4, 2)
apply_and_plot(X, S_b, b)
```



c) $\mu = (4, 2)$, $\Sigma = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$.

```
S_c = matrix(c(9, 0,  0, 1), nrow = 2, byrow = TRUE)
b = c(4, 2)
apply_and_plot(X, S_c, b)
```

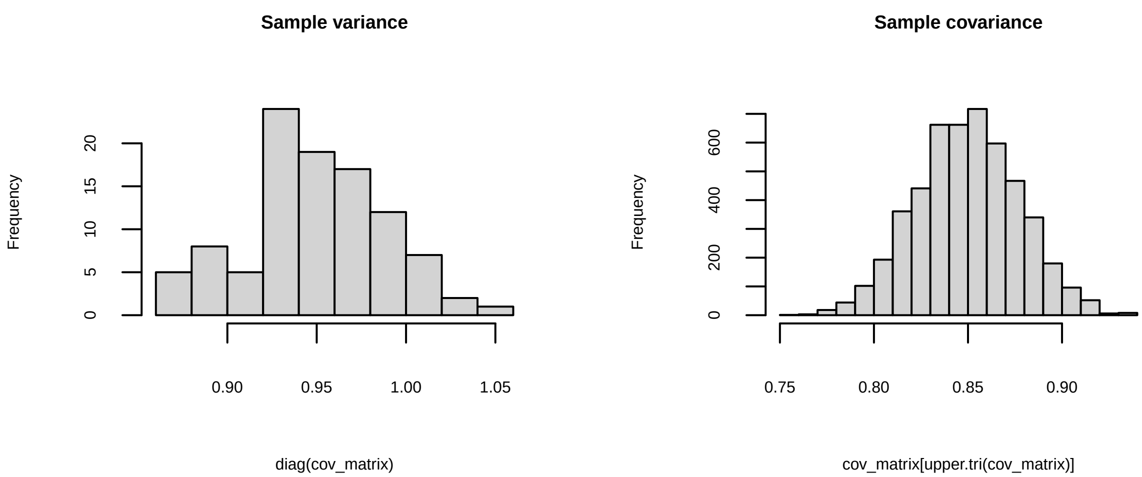


3 Exercise 3

Use a rnorm function (in R) to generate 200 random vectors from a multivariate normal distribution $N(0, I_{100\times100})$. Insert them into a matrix $X_{200\times100}$. Construct a matrix A, such that rows of a matrix $\tilde{X} = XA$ contains 200 vectors from a multivariate normal distribution $N(0, \Sigma_{100\times100})$, where $\Sigma(i,i) = 1$ and $\Sigma(i,j) = 0.9$ for $i \neq j$. In order to verify your solution, calculate a sample variance{covariance matrix. Next, on its basis, plot a histogram of obtained sample variances and calculate their mean. Do the same for sample covariances.

For a random vector $X = (X_1, ..., X_n)^T$, we define a covariance matrix as $\Sigma^X(i,j) = Cov(X_i, X_j) = E[X_iX_j] - E[X_i]E[X_j]$, where $\Sigma^X \in \mathbb{R}^{n\times n}$. Sample covariance between two vectors is calculated from the formula $Cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$. A built in function `cov` calculates the covariance between columns if given a matrix, which we will use here.

```
X = matrix(rnorm(200 * 100, 0, 1), ncol = 100)
Sigma = diag(nrow = 100, ncol = 100)
Sigma[lower.tri(Sigma)] = 0.9
Sigma[upper.tri(Sigma)] = 0.9
# We don't transpose because the transformation is XA
A = chol(Sigma)
X_tilde = X %*% A

cov_matrix = cov(X_tilde)
par(mfrow = c(1,2))
hist(diag(cov_matrix),
     cex.main = 0.6, cex.lab = 0.5, cex.axis = 0.5,
     main = "Sample variance")
hist(cov_matrix[upper.tri(cov_matrix)],
     cex.main = 0.6, cex.lab = 0.5, cex.axis = 0.5,
     main = "Sample covariance")
```




```
mean(diag(cov_matrix))
```

```
## [1] 0.9491657
```

```
mean(cov_matrix[upper.tri(cov_matrix)])
```

```
## [1] 0.8496006
```

We see that the variances they lie close to 1 and the covariances lie close 0.9.