Untitled

Daniel Frederico Lins Leite

5 March 2017

R Markdown

```
library(scales)
library(MASS)

generateData <- function (u, sdx, sdy){
  size <- 1000
  x0 <- rnorm(mean = u, sd = sdx, n = size)
  x1 <- rnorm(mean = u, sd = sdy, n = size)
  X <- matrix(x0, ncol = 1)
  cbind(X, matrix(x1, ncol = 1))
}

plotMatrix <- function(X){
  xdf <- data.frame(X)
  plot(xdf, ylim=c(-10, 10), asp=1, col = alpha("black",0.5))
}

plotEigenvectors <- function(X){
  C <- covarianceMatrix(X)
  eigenC <- eigen(C, symmetric = TRUE)
  dimensions <- eigenC$vectors %*% diag(sqrt(eigenC$values))
  arrows(0,0,dimensions[1,1],dimensions[2,1], lwd = 4, col = "red")
  arrows(0,0,dimensions[1,2],dimensions[2,2], lwd = 4, col = "blue")
}

rotationMatrix <- function(angle){
  matrix(c(cos(angle),sin(angle),-sin(angle),cos(angle)), nrow = 2, ncol= 2)
}

covarianceMatrix <- function(X){
  mu <- matrix(colMeans(X), nrow = 1)
  t(X)%*%X * (1/(nrow(X)-1)) - (t(mu)%*%mu)
}
```

Theory

The eigenvectors of the covariance matrix should form a new basis perpendicular to variation of the data.

```
C <- diag(1,2,2)
eigenC <- eigen(C)
dimensions <- eigenC$vectors %*% diag(eigenC$values)
C
```

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

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

```r
eigenC
```

```
## $values
## [1] 1 1
##
## $vectors
##      [,1] [,2]
## [1,]    0   -1
## [2,]    1    0
```

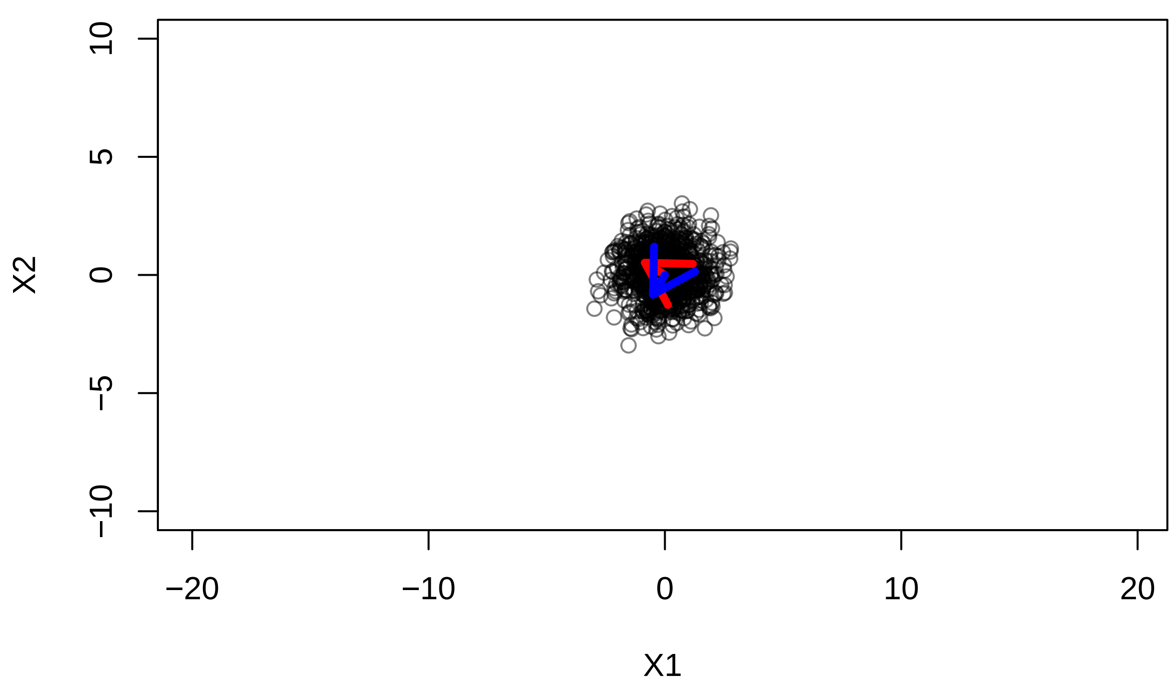
```r
dimensions
```

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

White Data

A white data (Gaussian with mean = 0 and Standard Deviation = 1) should have its eigenvector as near as possivle to the default basis.

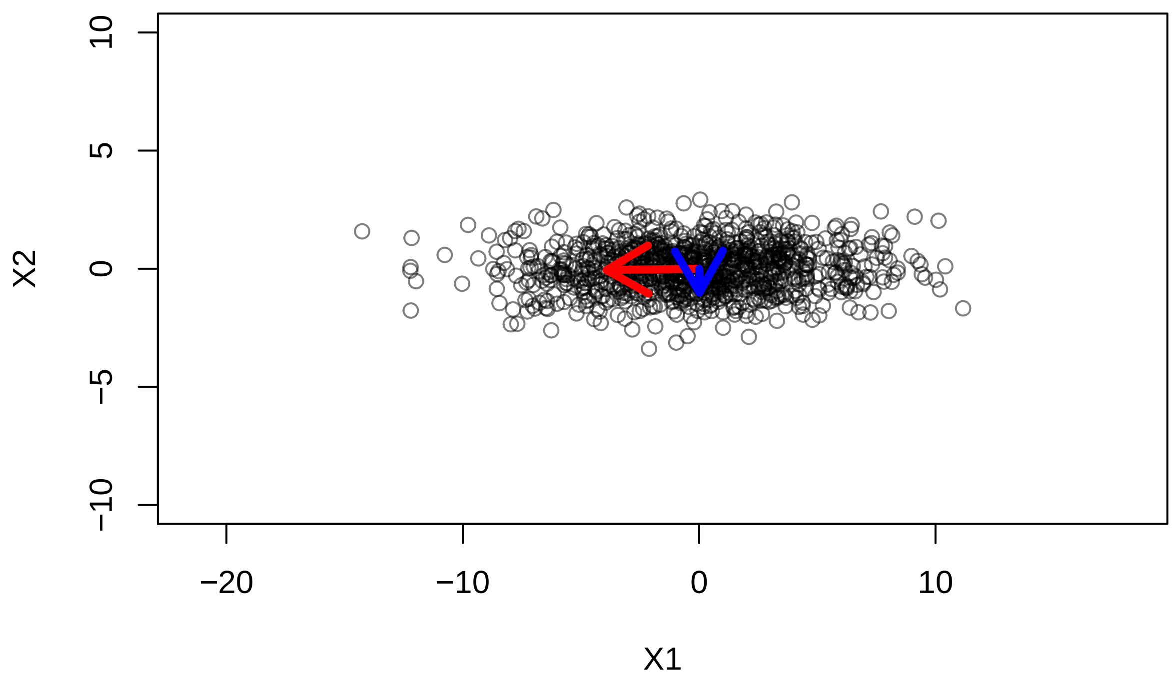
```r
X <- generateData(0,1,1)
plotMatrix(X)
plotEigenvectors(X)
```



Scaled Gaussian

A Scaled Gaussian should have one bigger eigenvector pointing to the variation of the data.

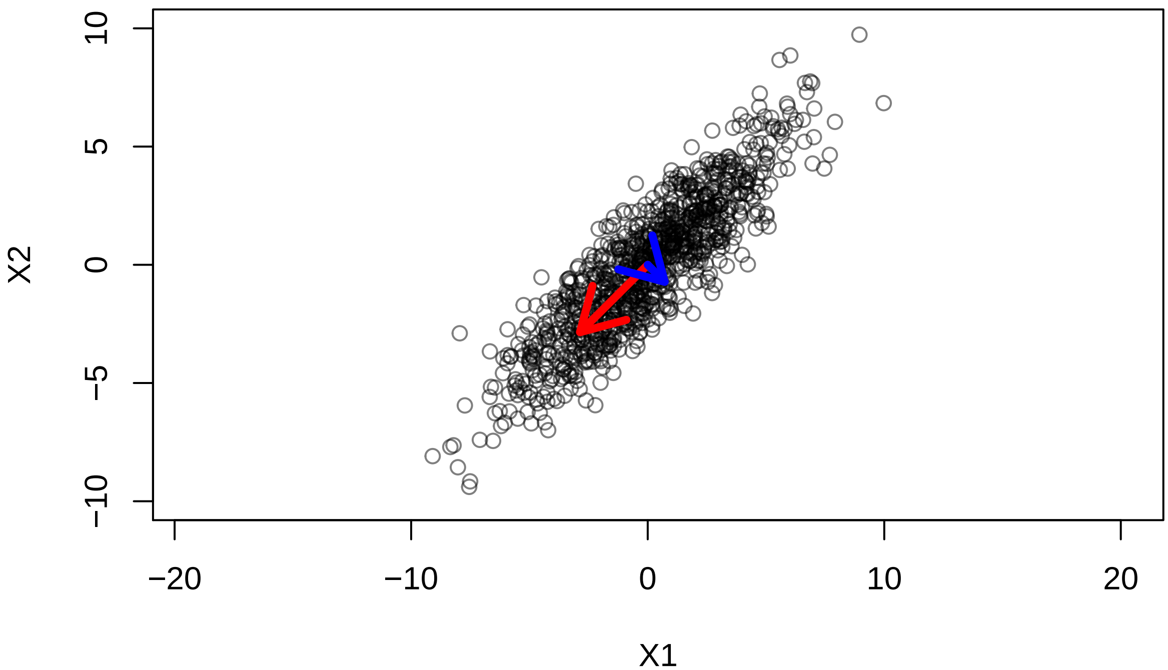
```
X <- generateData(0, 4, 1)
plotMatrix(X)
plotEigenvectors(X)
```



Scaled and Rotated Gaussian

The Scaled and Rotated Gaussian should have its eigenvectors rotated adn poiting to new variation direction.

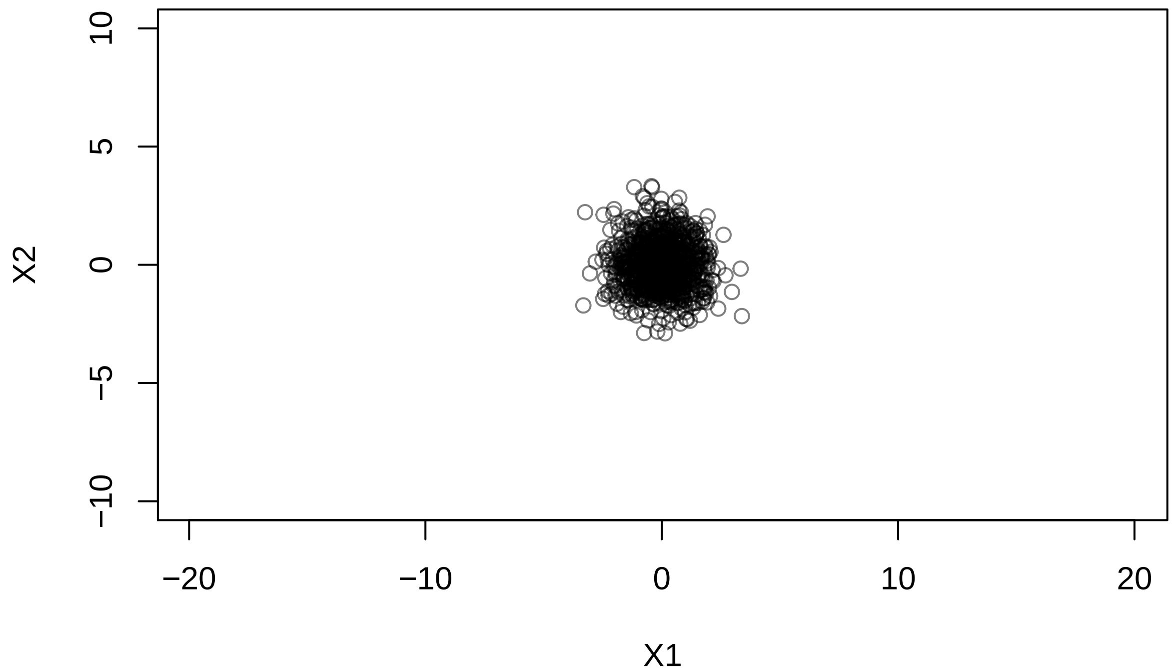
```
X <- generateData(0, 4, 1)
rot45 <- rotationMatrix(-pi/4)
X <- X %*% rot45
C <- covarianceMatrix(X)
eigenC <- eigen(C)
plotMatrix(X)
plotEigenvectors(X)
```



Covariance Matrix is the Linear Transformation Squared

The Scaled and Rotated Gaussian can be generated from a White Data Gaussian using two Linear Transformations: 1 - Scale Transformation 2 - Rotation Transformation

```
X <- generateData(0,1,1)
R <- rotationMatrix(-pi/4)
S <- matrix(c(4,0,0,1), nrow=2, ncol=2)
plotMatrix(X)
```



The complete transformation can be aggregated in the Matrix:

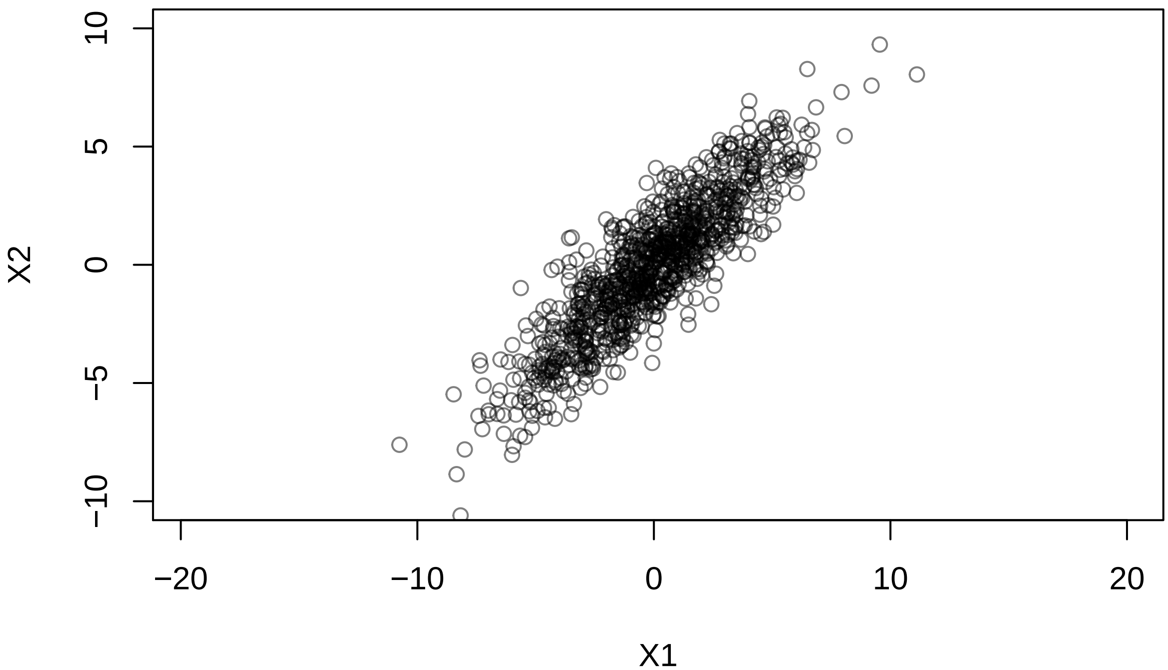
```
T <- R%*%S
```

In thos case the covariance matrix is:

```
#R is a orthonormal matrix so t(R) = ginv(R)
C <- T%*%t(T)
```

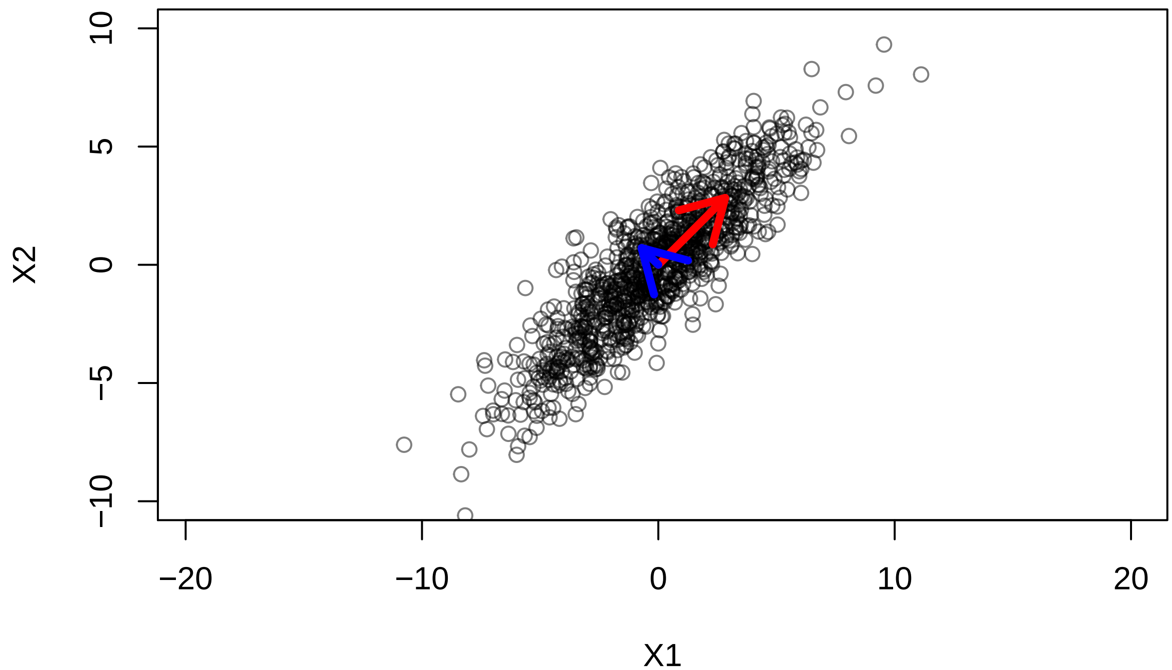
The final data is identical

```
plotMatrix(X%*%S%*%R)
```



In this case its eigenvectors can be calculated as:

```
eigenVectors <- t(R)%*%S
plotMatrix(X%*%S%*%R)
arrows(0,0,eigenVectors[1,1],eigenVectors[2,1], lwd = 4, col = "red")
arrows(0,0,eigenVectors[1,2],eigenVectors[2,2], lwd = 4, col = "blue")
```




```r
# These matrices should be as near as zero as possible
eigenC$vectors - t(R)
```

```
##           [,1]      [,2]
## [1,] -1.415677  1.412747
## [2,] -1.412747 -1.415677
```

```r
diag(sqrt(eigenC$values), 2, 2) - S
```

```
##            [,1]       [,2]
## [1,] 0.03828931 0.00000000
## [2,] 0.00000000 0.01577302
```