

STAT 428 Homework 0: Solution

Vishesh Gupta

1/31/2021

Exercise

Below are a few toy problems. You can type your answers under each problem. If you like, you can submit your .Rmd and .pdf files on Gradescope. I'll read them.

Problem 1

```
#install.packages("LaplacesDemon")  
#install.packages("VGAM")
```

```
qlaplace <- function (u, mu, std){  
  x <- mu - (std * (sign(u - 0.5)) * (log(1 - (2*(abs(u - 0.5))))))  
  return (x)  
}
```

```
u = runif(n = 1000)  
head(u)
```

```
## [1] 0.06447939 0.62356897 0.19449879 0.82609551 0.40273922 0.34632097
```

```
mu = 1  
std = 2  
X <- qlaplace(u, mu, std)  
head(X)
```

```
## [1] -3.0965250 1.5677465 -0.8883643 3.1122037 0.5673623 0.2655158
```

```
library(LaplacesDemon)
```

```
##  
## Attaching package: 'LaplacesDemon'
```

```
## The following object is masked _by_ '.GlobalEnv':  
##  
## qlaplace
```

```
library(VGAM)
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
##
```

```
## Attaching package: 'VGAM'
```

```
## The following object is masked _by_ '.GlobalEnv':
```

```
##
```

```
##      qlaplace
```

```
## The following objects are masked from 'package:LaplacesDemon':
```

```
##
```

```
##      cloglog, dgpdp, dlaplace, dpareto, logit, loglog, plaplace, ppareto,
```

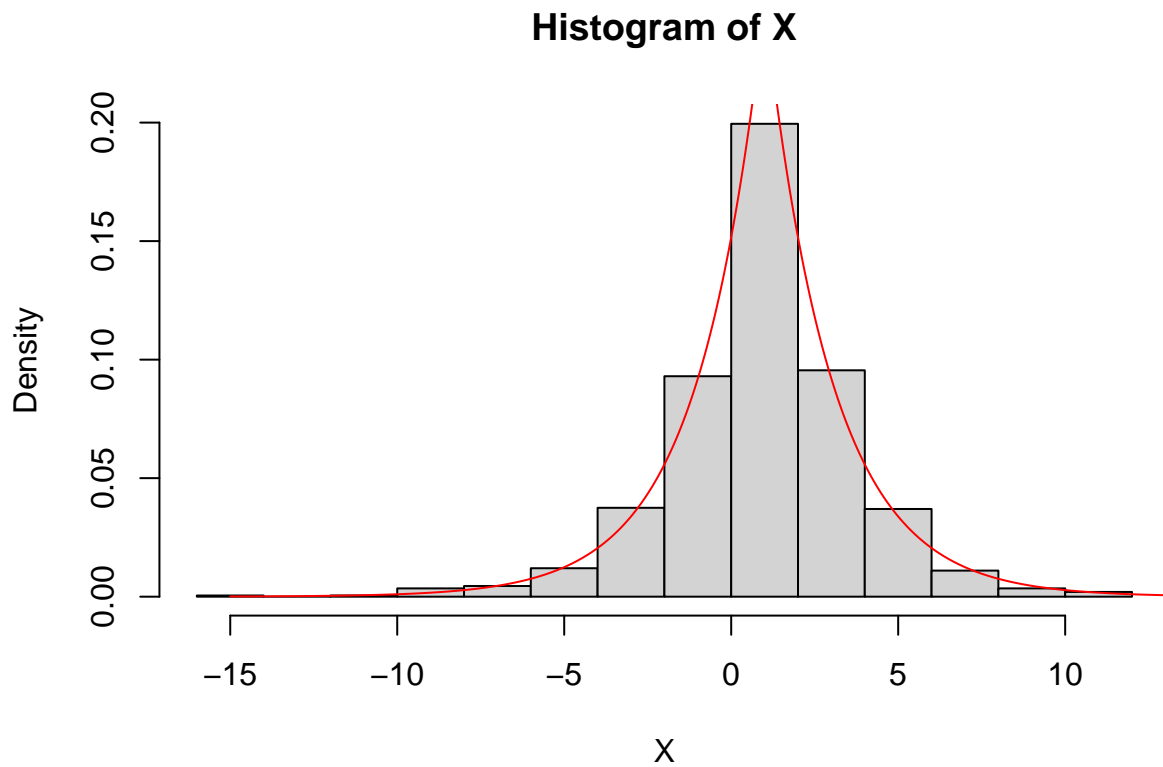
```
##      qlaplace, qpareto, rgpd, rlaplace, rpareto
```

```
hist(X, freq = F)
```

```
xx <- seq(-15,15,.01)
```

```
fx <- dlaplace(xx, 1, 2)
```

```
lines(xx,fx, type = 'l', col = 'red')
```



Problem 2

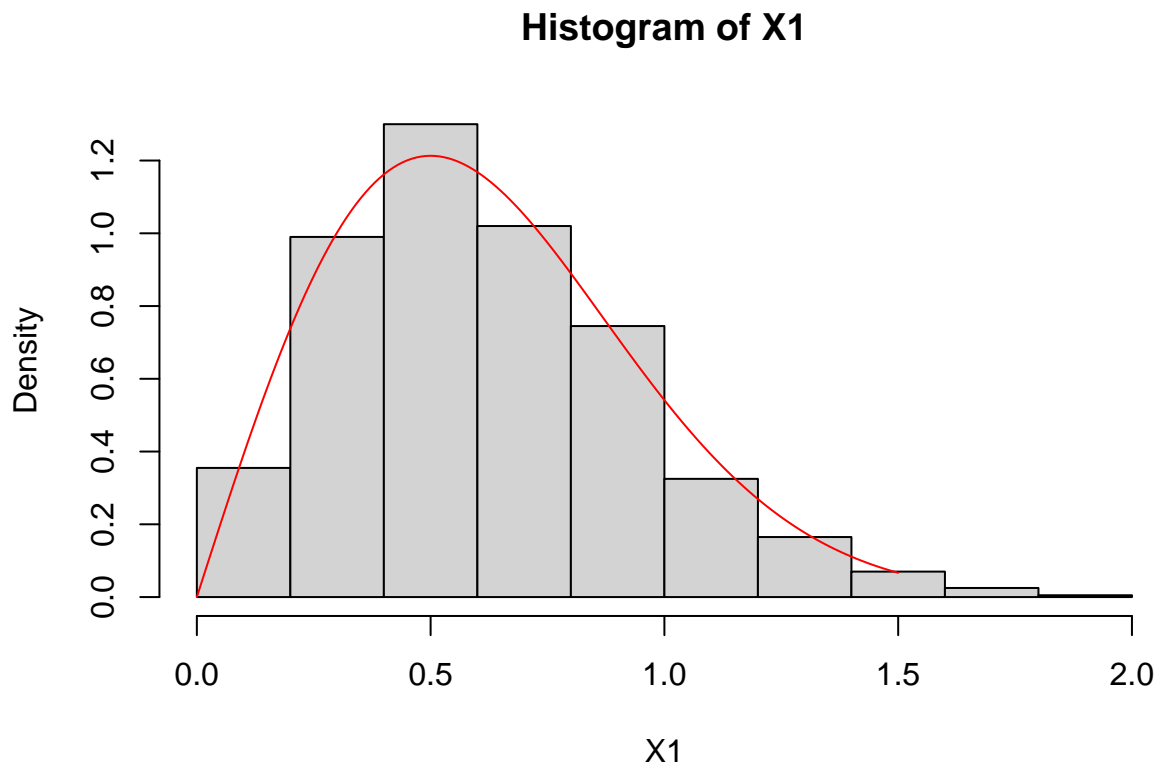
```
grayl <- function (u1, std1){  
  x <- sqrt(-2 * (std1^2) * log(u1))  
  return (x)  
}
```

```
u1 = runif(n = 1000)  
head(u1)
```

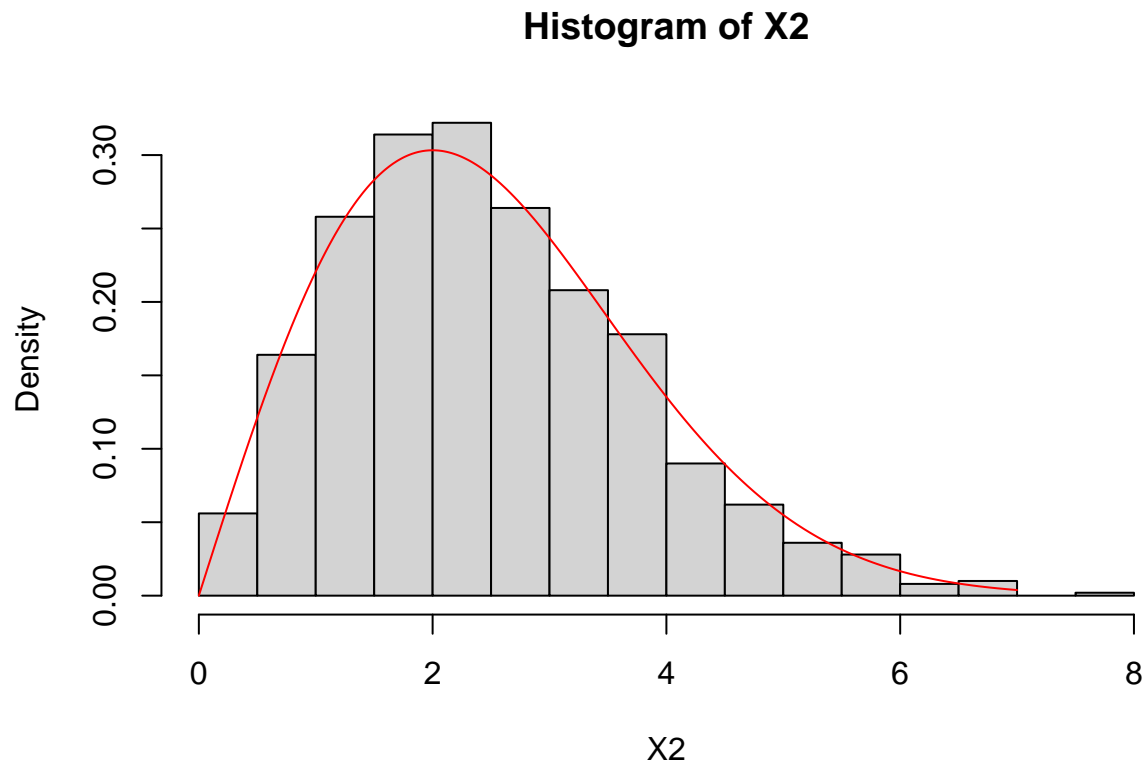
```
## [1] 0.1203801 0.2475700 0.9446075 0.5887718 0.3829895 0.6279158
```

```
std1 = 0.5  
std2 = 2  
std3 = 100  
X1 <- grayl(u1, std1)  
X2 <- grayl(u1, std2)  
X3 <- grayl(u1, std3)
```

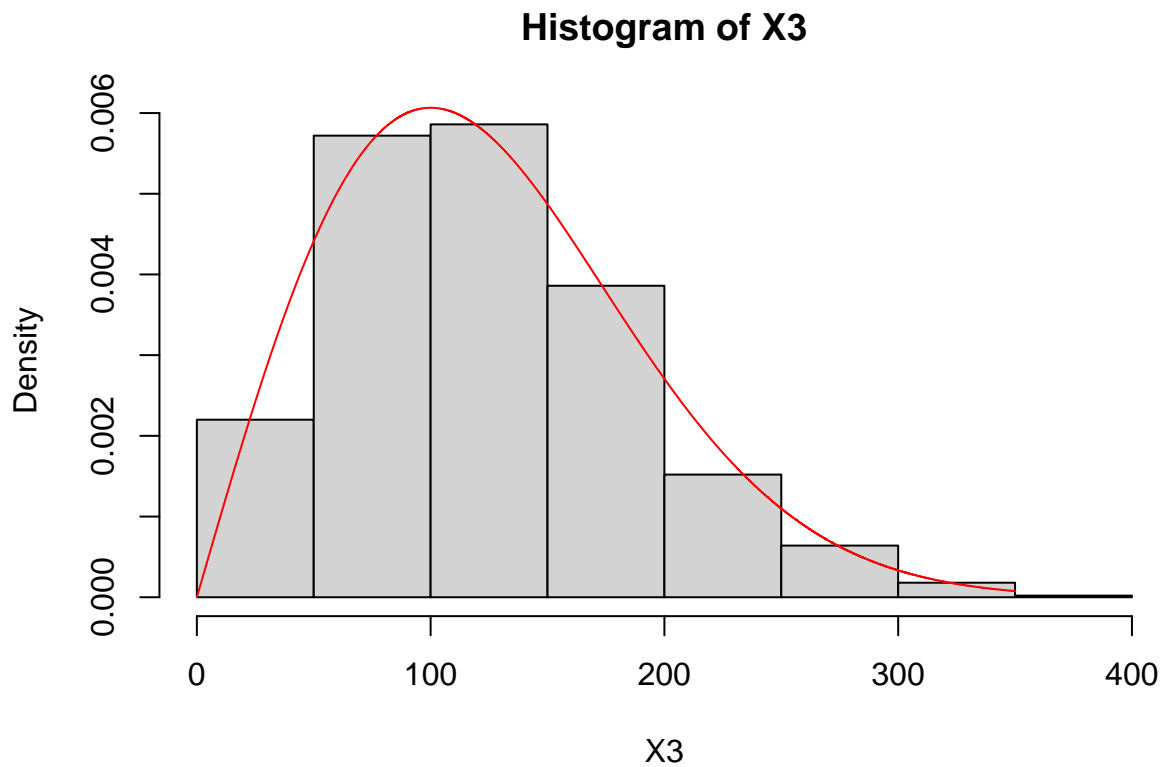
```
hist (X1, freq = F)  
xx <- seq(0,1.5,.01)  
fx <- dgamma(xx, 0.5)  
lines(xx,fx, type = 'l', col = 'red')
```



```
hist (X2, freq = F)
xx <- seq(0,7,.01)
fx <- drayleigh(xx, 2)
lines(xx,fx, type = 'l', col = 'red')
```



```
hist (X3, freq = F)
xx <- seq(0,350,.01)
fx <- drayleigh(xx, 100)
lines(xx,fx, type = 'l', col = 'red')
```



Problem 3

```
dic <- function(u, p){
  Fx <- cumsum (p)
  m <- length (p)
  x <- 1
  for ( i in 1:m){
    x <- x + (u > Fx[i])
  }
  return (x)
}
```

```
U <- runif(1000)
head(U)
```

```
## [1] 0.75689325 0.76232135 0.00351624 0.57852035 0.27766171 0.17621593
```

```
probs <- c(0.1, 0.2, 0.2, 0.2, 0.3)
X_dic <- dic (U, p = probs)
head(X_dic)
```

```
## [1] 5 5 1 4 2 2
```

```
table(X_dic)/1000
```

```
## X_dic
##      1      2      3      4      5
## 0.101 0.190 0.215 0.204 0.290
```

using the R function

```
X_sample <- sample (x = 1:5, size = 1000, prob = probs, replace = T)
table(X_sample)/1000
```

```
## X_sample
##      1      2      3      4      5
## 0.092 0.198 0.210 0.188 0.312
```

Problem 4

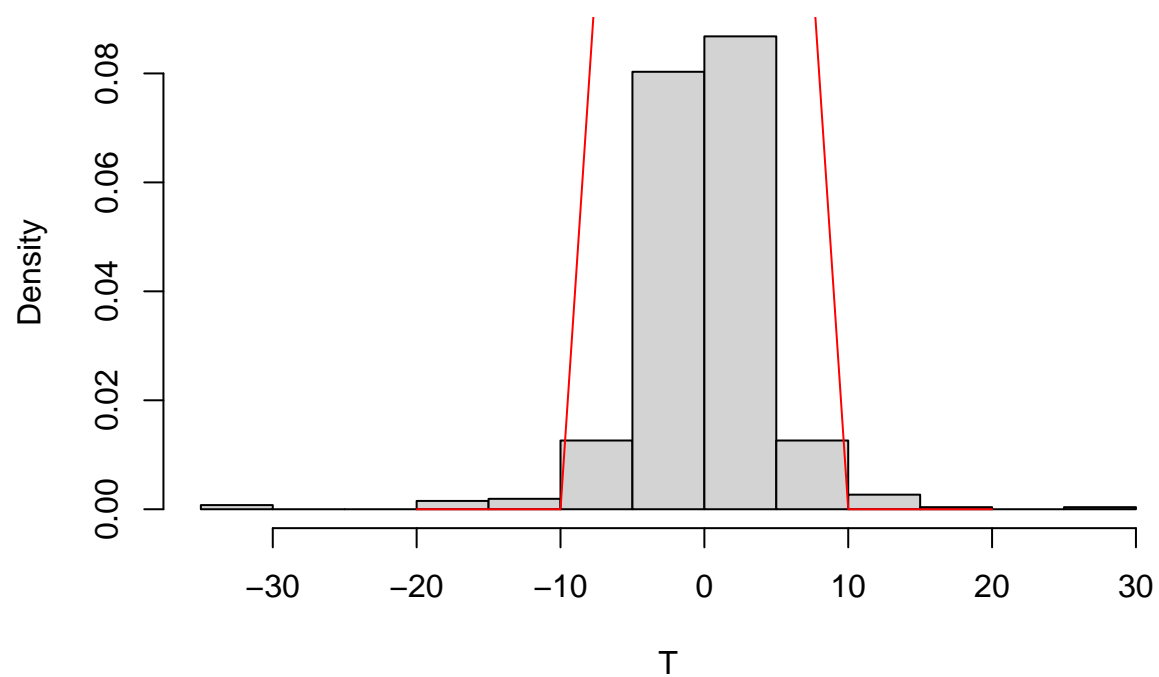
```
Z <- rnorm(1000)
Z1 <- matrix ( rnorm (1000*30), nrow = 1000, ncol = 30)
X <- rowSums(Z1)
v <- 30
T <- Z / (sqrt (X/v))
```

```
## Warning in sqrt(X/v): NaNs produced
```

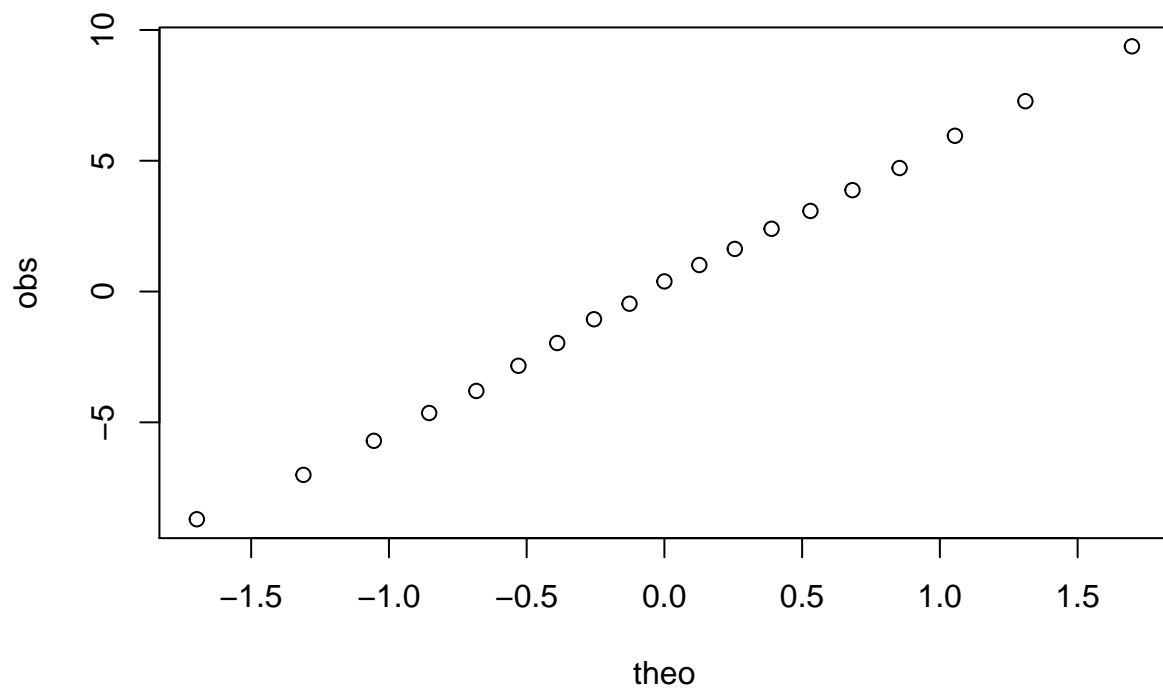
```
hist(T, freq = F)
xx <- seq(-20,20,10)
lines(xx, dt(xx,df = 30), type = "line", col = "red")
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): plot type 'line' will be
## truncated to first character
```

Histogram of T



```
ps <- seq(0.05, 0.95, by = 0.05)
theo <- qt(ps, df = 30)
obs <- quantile(X, probs=ps)
plot(theo, obs)
```

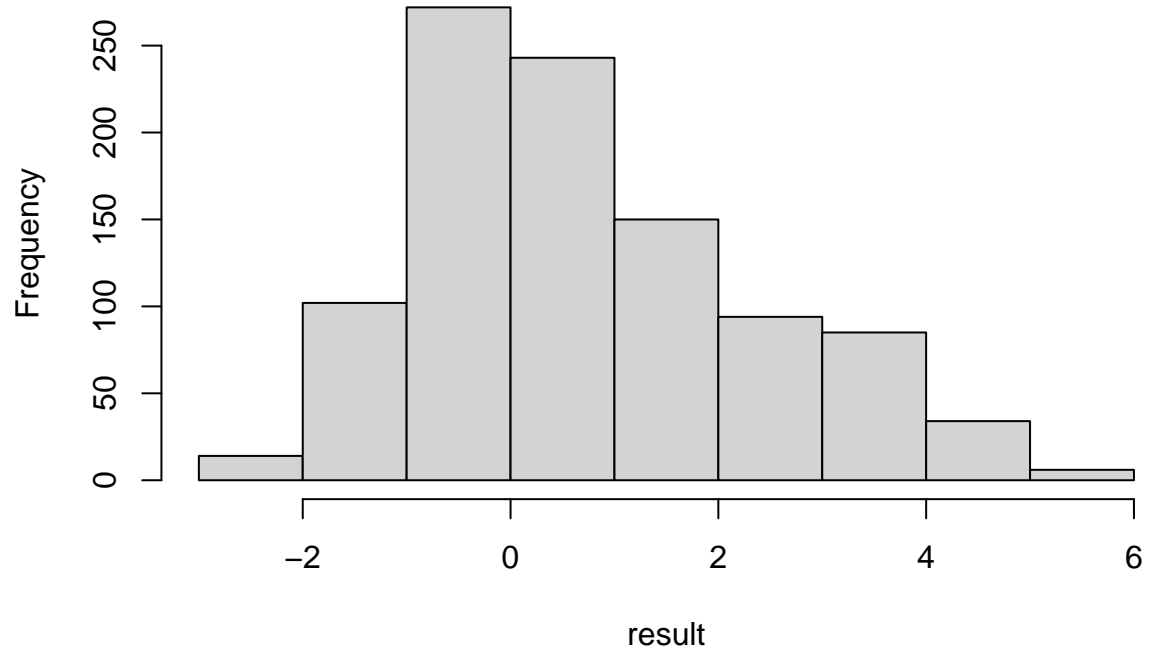


Problem 5

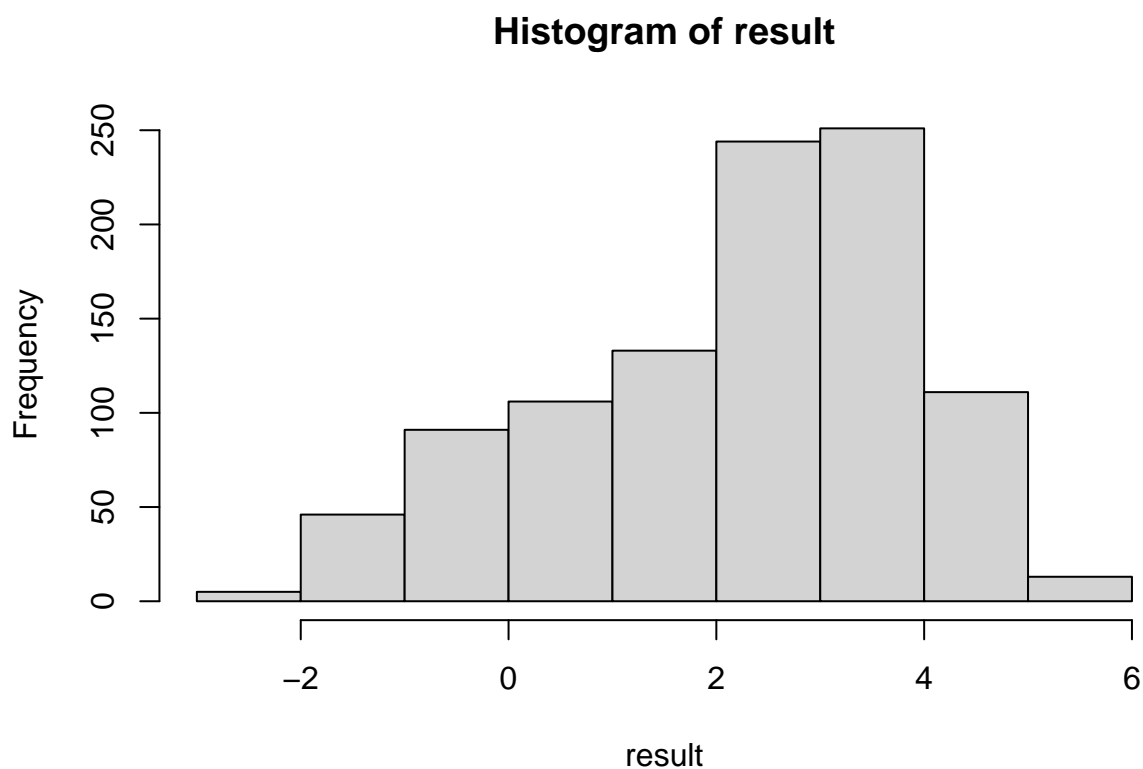
```
dmd = function(p){  
  X <- 1 * ((1-p)<runif(1000))  
  means <- c(3,0)  
  stds <- c(1,1)  
  result <- rnorm (1000, means[X+1], stds[X+1] )  
  hist(result)  
}
```

```
dmd(0.75)
```


Histogram of result



```
dmd(0.25)
```



it is not bi-modal since we see no twin peaks

Problem 6

```
z <- matrix (rnorm(1000*3), nrow = 1000, ncol = 3)
mu <- matrix (c(0,1,2), nrow = 1000, ncol = 3, byrow=T)
sig <- matrix (c(1,-0.5,0.5,-0.5,1,-0.5,0.5,-0.5,1), nrow = 3, ncol = 3)
A <- chol(sig)
x <- z%*%A + mu
```

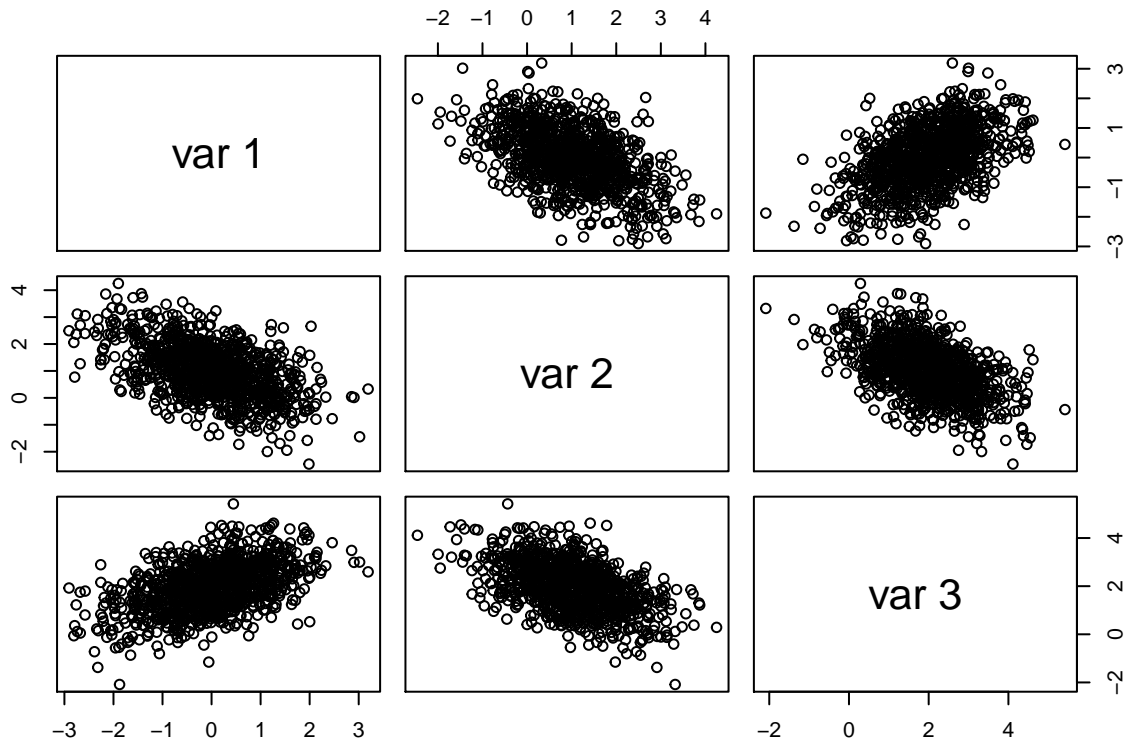
```
round(cov(x),3)
```

```
##      [,1] [,2] [,3]
## [1,] 0.995 -0.503 0.491
## [2,] -0.503 0.984 -0.514
## [3,] 0.491 -0.514 1.035
```

```
round(colMeans(x),3)
```

```
## [1] -0.004 1.003 1.979
```

```
pairs(x)
```



We can thereby notice that the visual plots agree with the covariance matrix