#Logistic distribution

test1 <- function(){

set.seed(1234)

Nsim <- 10^4

U <- runif(Nsim)

U.mean <- mean(U)

beta <- 1

X <- U.mean - (log((1/U) - 1) \* beta)

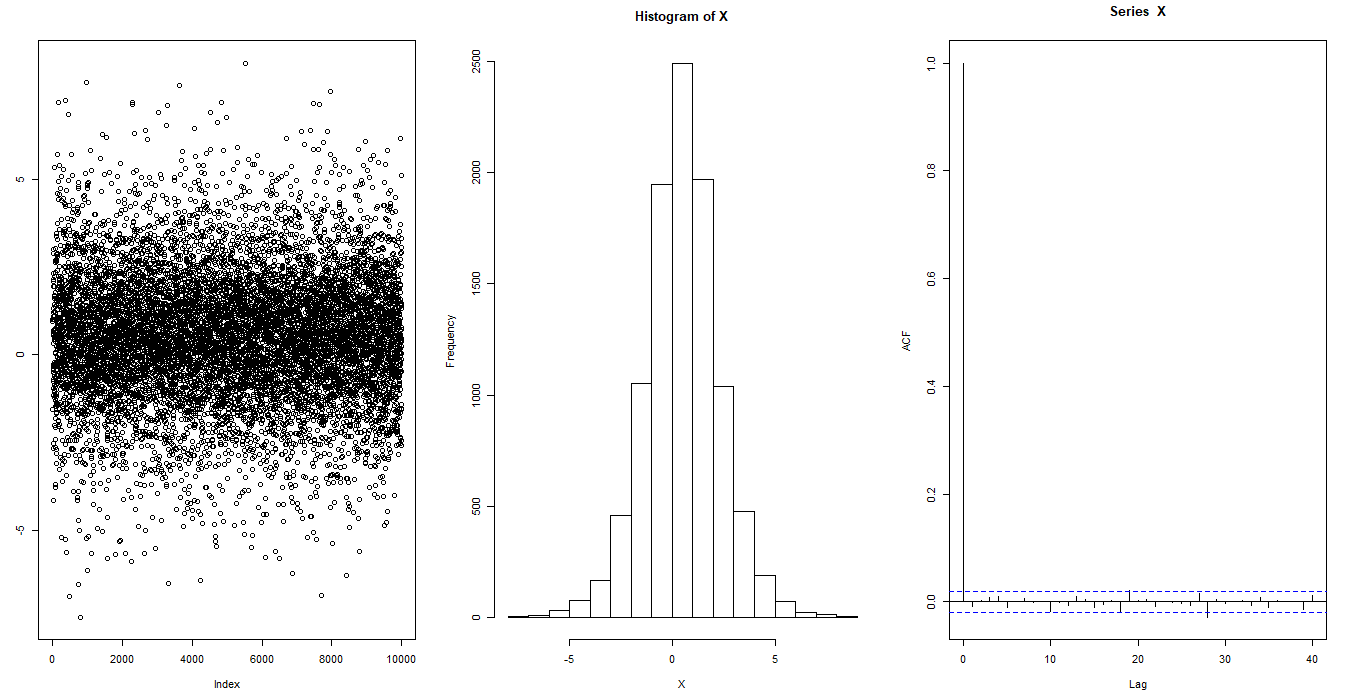
par(mfrow=c(1,3))

plot(X)

hist(X)

acf(X)

}



system.time(test1());

# > system.time(test1());

# user system elapsed

# 0.15 0.53 0.75

test2 <- function(){

set.seed(1234)

Nsim <- 10^4

X <- rlogis(Nsim)

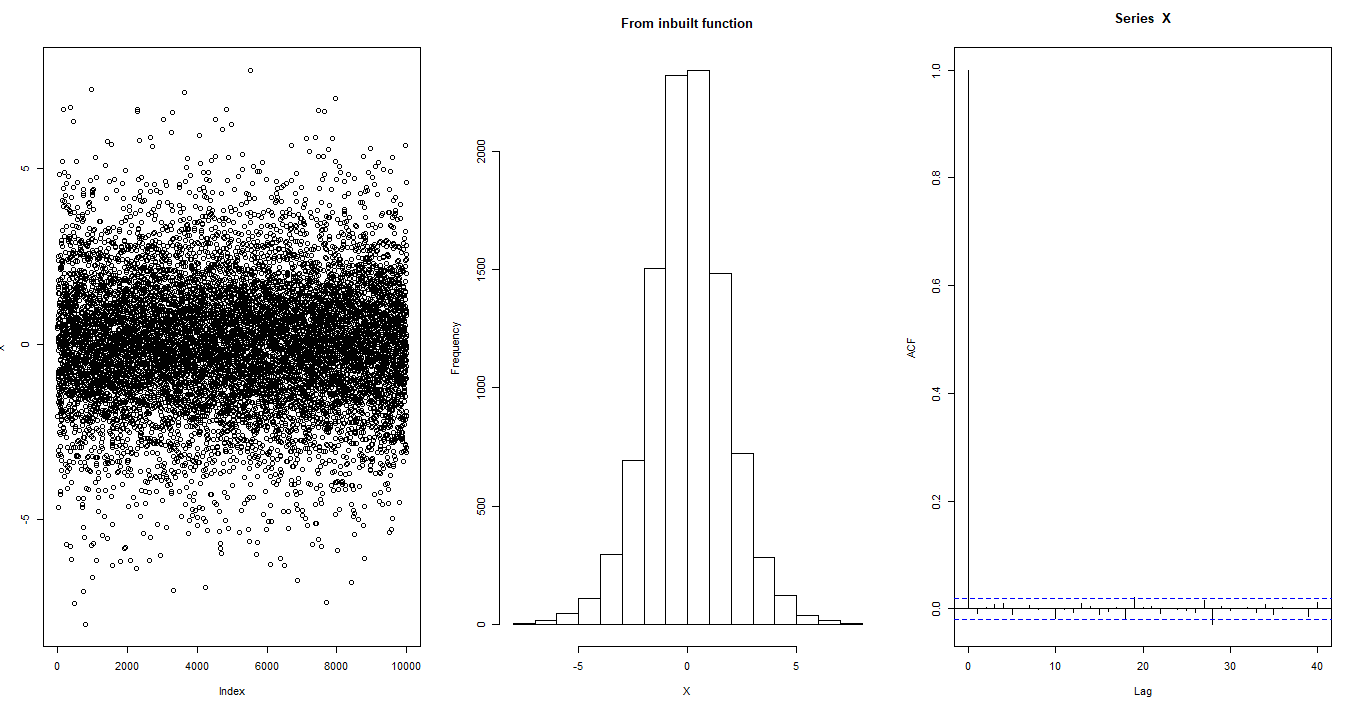
par(mfrow=c(1,3))

plot(X)

hist(X, main = "From inbuilt function")

acf(X)

}



system.time(test2());

# > system.time(test2());

# user system elapsed

# 0.14 0.55 0.73

#cauchy distribution

test3 <- function(){

set.seed(1234)

Nsim <- 10^4

U <- runif(Nsim)

U.mean <- mean(U)

X <- (tan((U - 0.5) \* pi) ) \* sd(U)

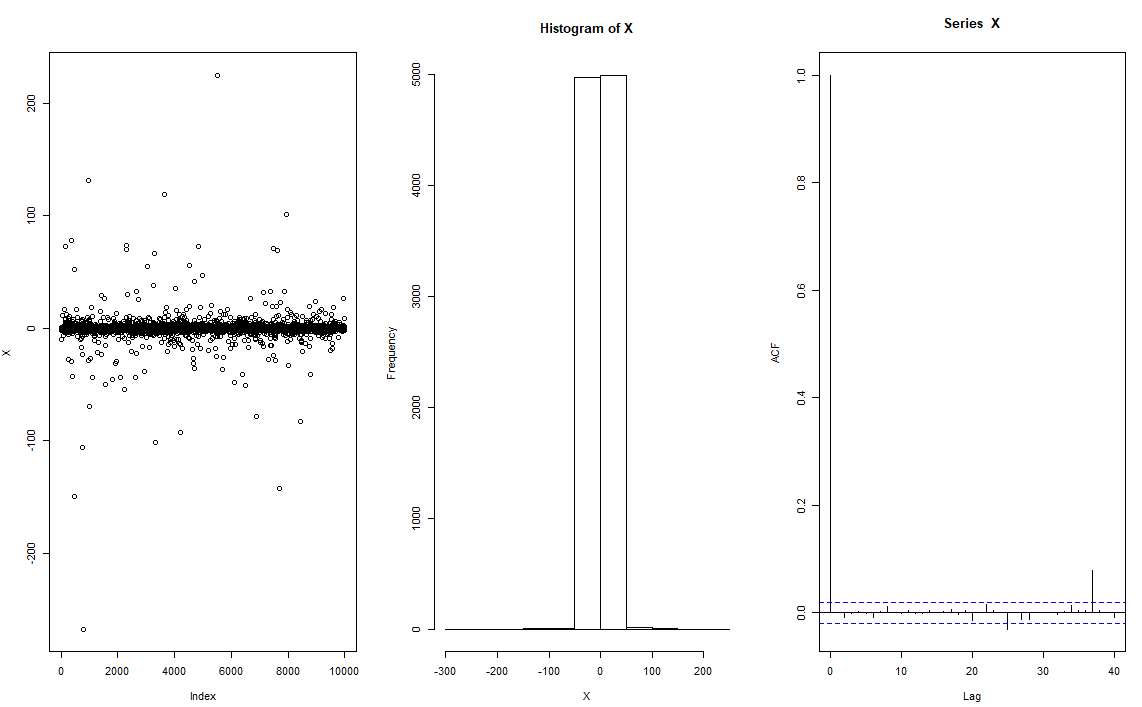
par(mfrow=c(1,3))

plot(X)

hist(X)

acf(X)

}



system.time(test3());

# user system elapsed

# 0.19 0.70 1.06

test4 <- function(){

set.seed(1234)

Nsim <- 10^4

X <- rcauchy(Nsim)

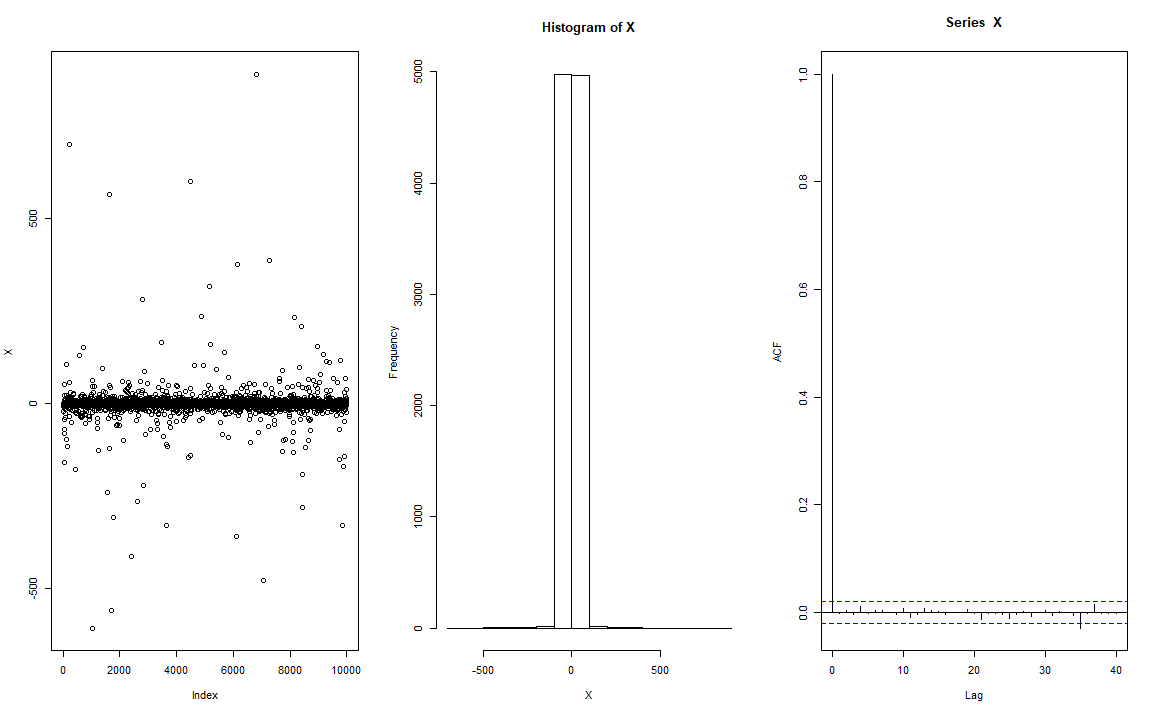
par(mfrow=c(1,3))

plot(X)

hist(X)

acf(X)

}



# > system.time(test4());

# user system elapsed

# 0.14 0.66 0.91

#2.7)

k <- function(x,alpha,beta,a,b){

return (dbeta(x,alpha,beta)/dbeta(x,a,b))

}

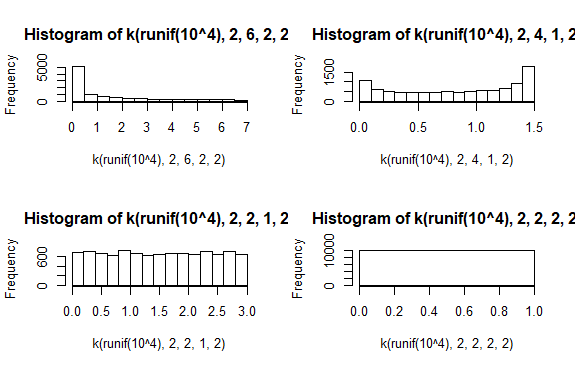
hist(k(runif(10^4),2,6,2,2))

hist(k(runif(10^4),2,4,1,2))

hist(k(runif(10^4),2,2,1,2))

#when a = alpha and b - beta

hist(k(runif(10^4),2,2,2,2))



#2.13)

x <- seq(0,1,0.01)

N <- 1000

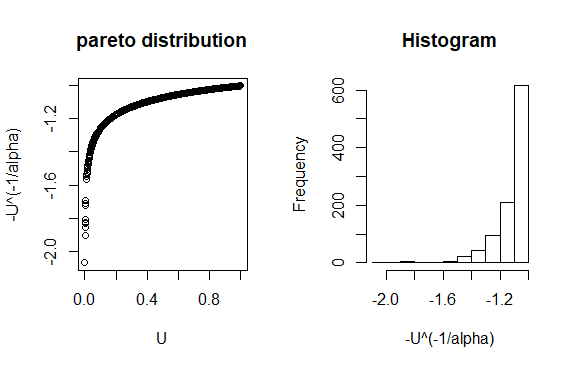
U <- runif(N)

par(mfrow=c(1,2))

alpha <- 10

plot(U,-U^(-1/alpha), main = "pareto distribution")

hist(-U^(-1/alpha), main = "Histogram")



#2.15)

Nsim=10^4;

lambda=1

spread=3\*sqrt(lambda)

t=round(seq(max(0,lambda-spread),lambda+spread,1))

prob=ppois(t, lambda)

X=rep(0,Nsim)

for (i in 1:Nsim){

u=runif(1)

X[i]=t[1]+sum(prob<u) }

x <- seq(0,1,0.0001)

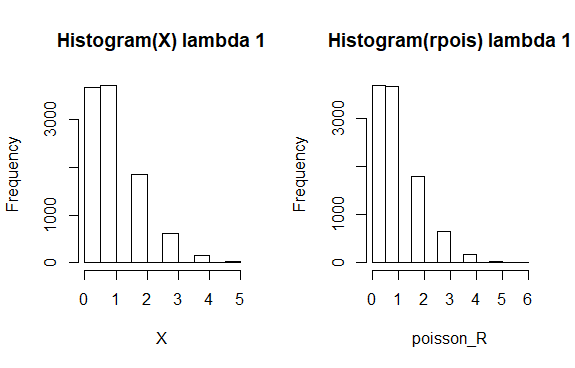
poisson\_R <- rpois(x,lambda)

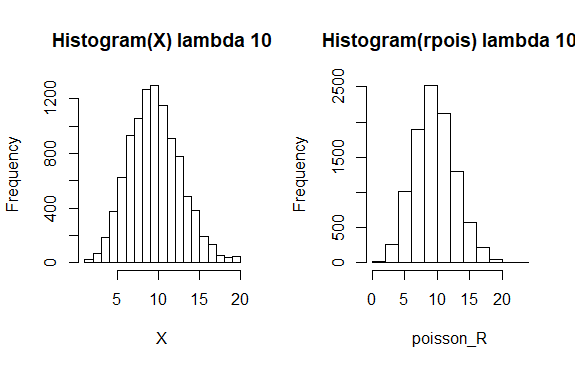
par(mfrow=c(1,2))

hist(X)

hist(poisson\_R)

**For lambda equals 1**





**#2.18)**

**#a)**

**library(ggplot2)**

**f <- function(x){**

**value1 <- exp(-(x^2)/2)**

**value2 <- (sin(6 \* x))^2 + 3 \* (cos(x))^2 \* (sin(4\*x))^2 + 1**

**return( value1 \* value2)**

**}**

**g <- function(x){**

**return ( exp((-(x)^2)/sqrt(2\*pi)) )**

**}**

**X <- seq(-2,2,by=0.0016)**

**X <- X[0:2500]**

**N <- 2500**

**#X <- runif(N)**

**sim <- optimize(f, interval = c(0,1) , maximum = TRUE)**

**M <- sim$objective**

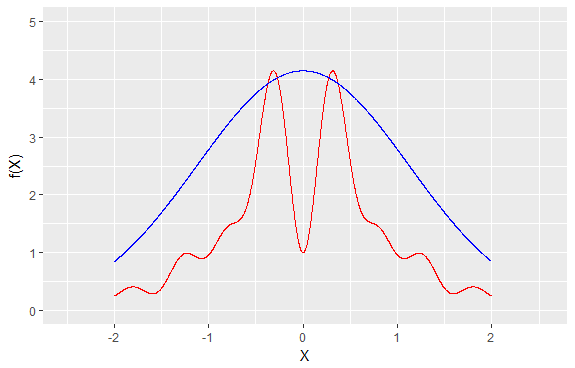
**#M <- 1/sim.obj**

**#when M is 4.14**

**ggplot() +**

**geom\_line(aes(x = X, y = f(X)), color = "red") + xlim(c(-2.5,2.55)) + ylim(c(0,5)) +**

**geom\_line(aes(x = X, y = M \* g(X)), color = "blue")**

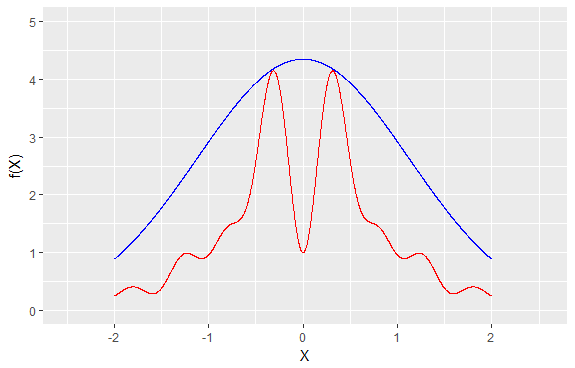


**#When M is 4.35**

**ggplot() +**

**geom\_line(aes(x = X, y = f(X)), color = "red") + xlim(c(-2.5,2.55)) + ylim(c(0,5)) +**

**geom\_line(aes(x = X, y = 4.35 \* g(X)), color = "blue")**



**#b)**

**M <- 4.35**

**#Optimal M value is 4.35**

**h <- function(x)**

**{**

**return ( f(x) / g(x) )**

**}**

**U <- runif(2500, min = 0, max = M)**

**Fn <- f(U)**

**Gn <- g(U)**

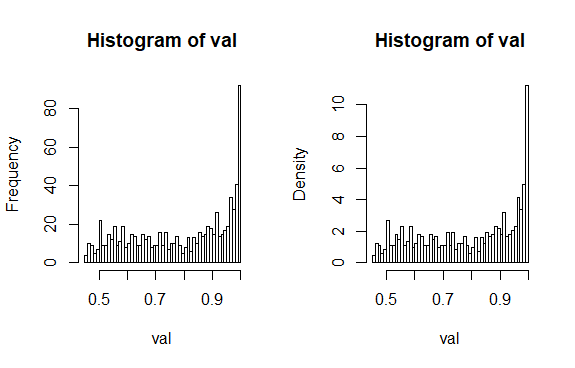
**val <- Gn[U < h(U)]**

**length(val)/N**

**0.32**

**hist(val,breaks = 40)**

**hist(val,breaks = 40, prob = T)**



**#c**

**NMval <- max(h(X))**

**Nval <- NMval / M**

**#Nval or normalizing constant of f is 0.992**

**par(mfrow= c(1,2))**

**plot(X,f(X)/Nval , main = "normalized f(X)")**

**hist(f(X))**

