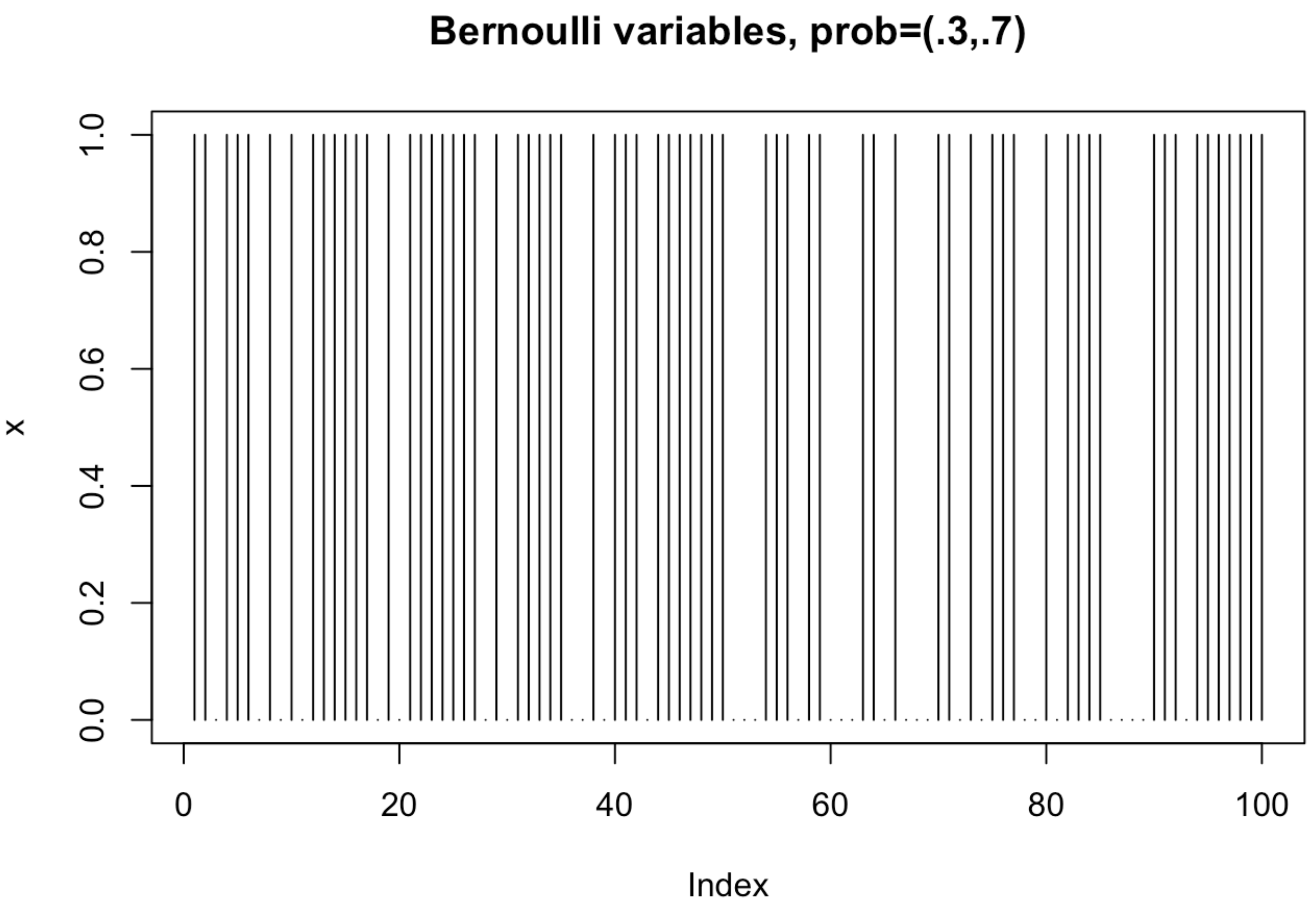
Statistics_intro.R

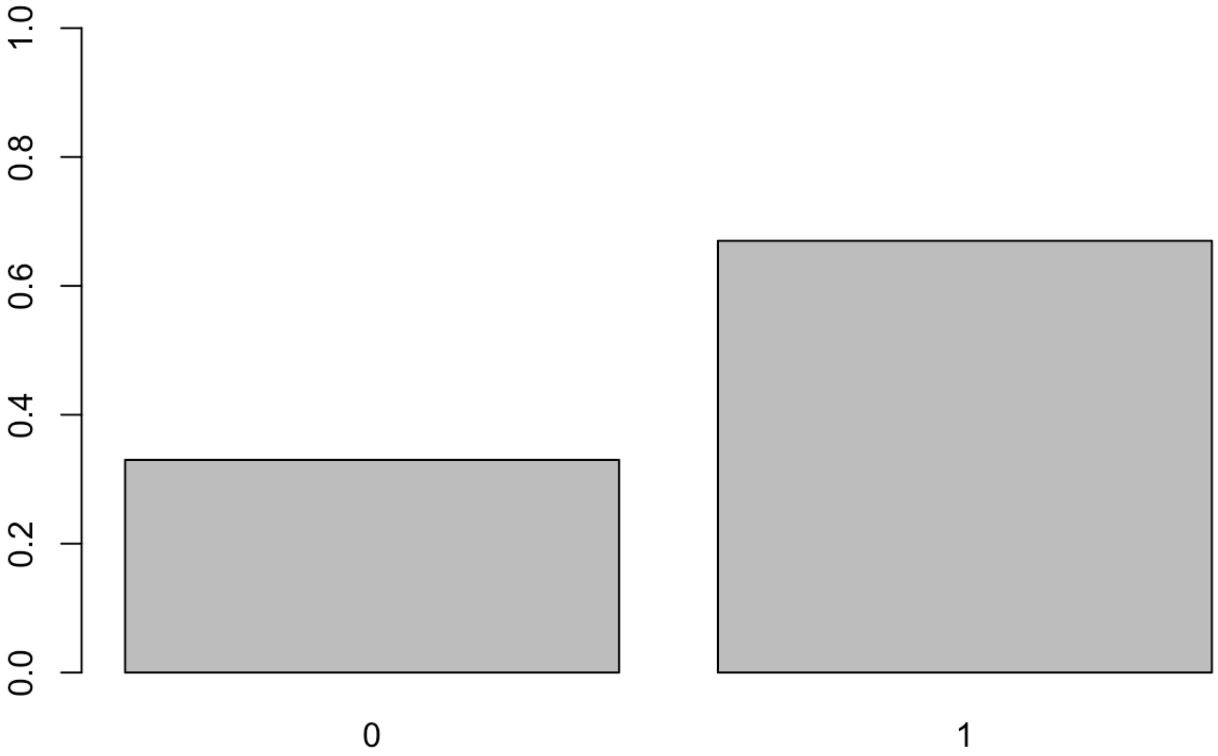
buddhananda

Wed Feb 19 21:57:47 2020

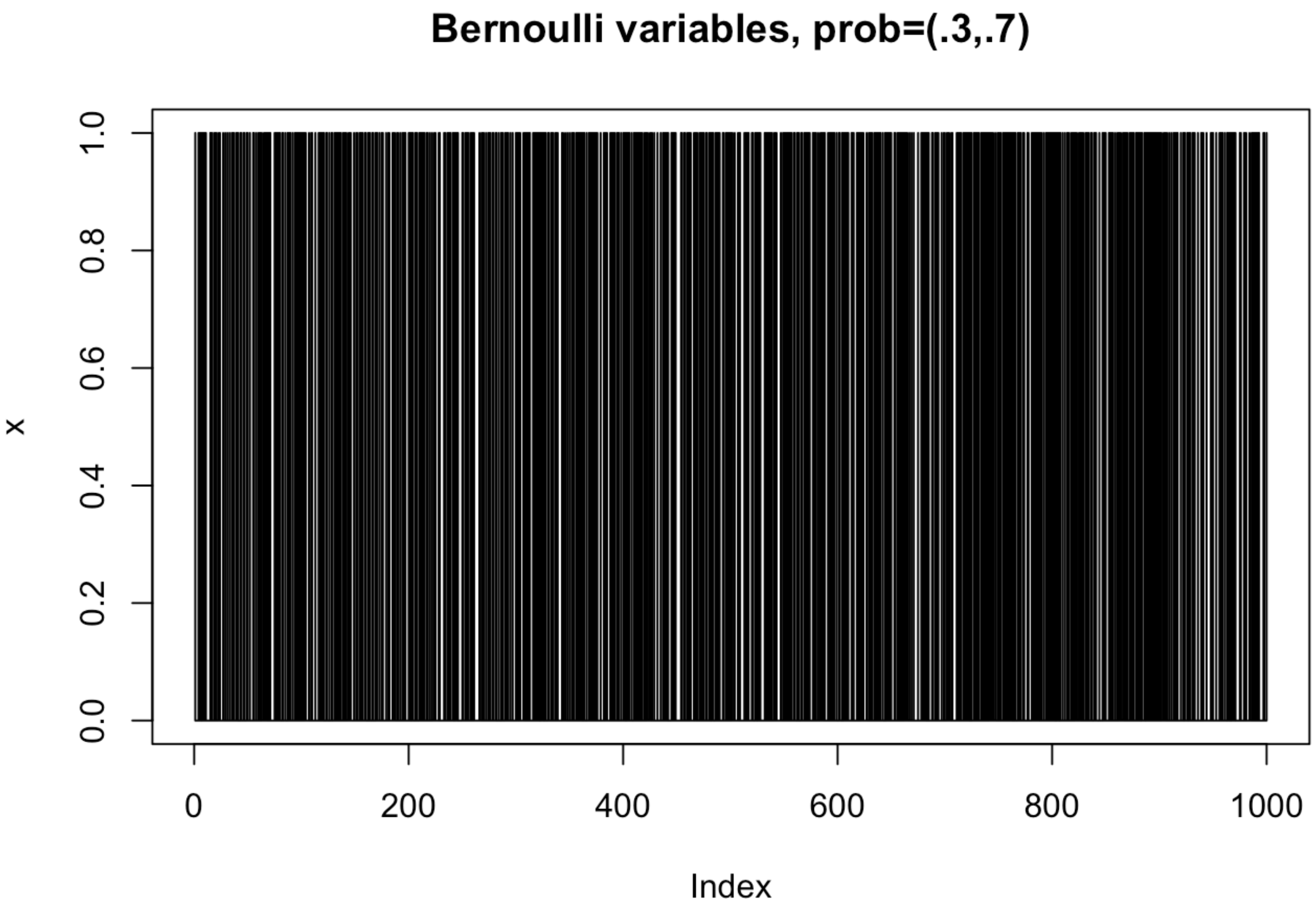
```
# Binary data of parameter P(X=1)=0.7
n <- 100
x <- sample(c(0,1), n, replace=T, prob=c(.3,.7))
par(mfrow=c(1,1))
plot(x, type='h',main="Bernoulli variables, prob=(.3,.7)")
```



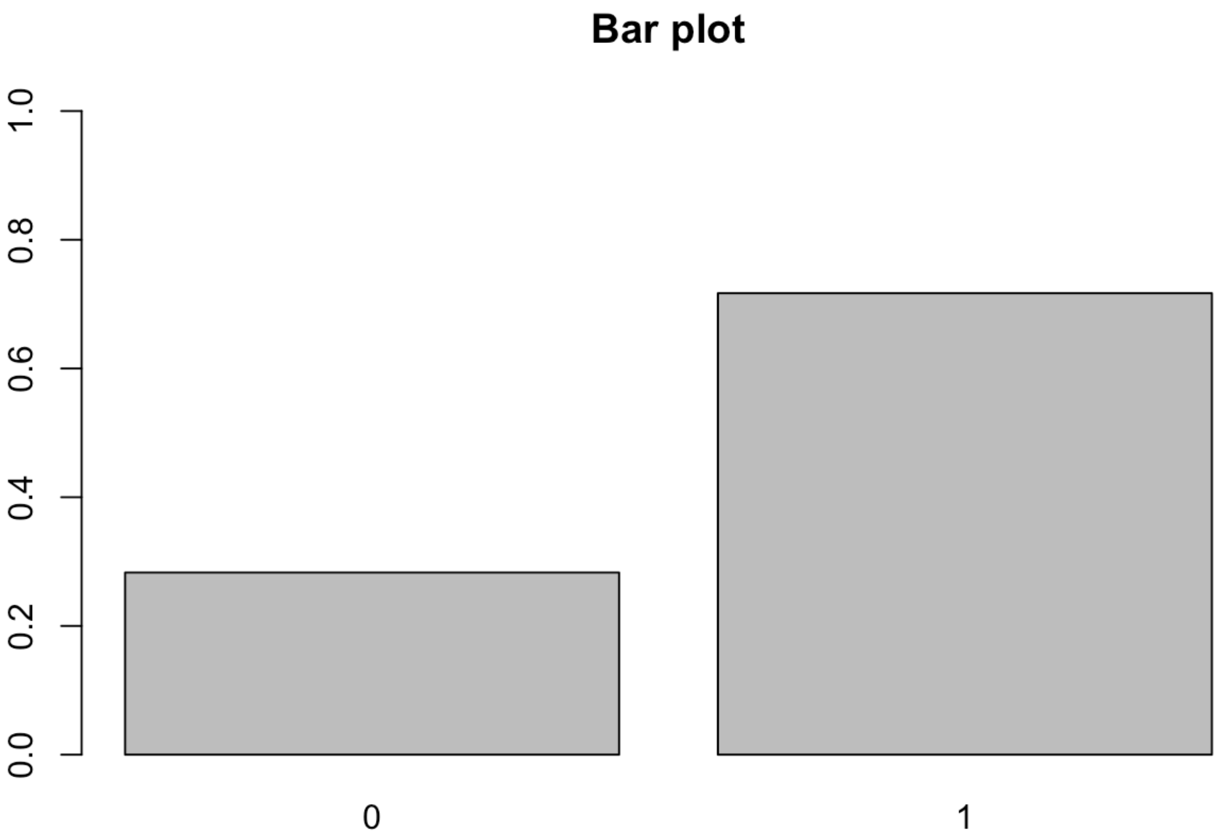

```
barplot(table(x)/n, ylim = c(0,1))
```




```
# Binary/ Bernoulli distribution of parameter p=0.7
n <- 1000
x <- sample(c(0,1), n, replace=T, prob=c(.3,.7))
par(mfrow=c(1,1))
plot(x, type='h', main="Bernoulli variables, prob=(.3,.7)")
```



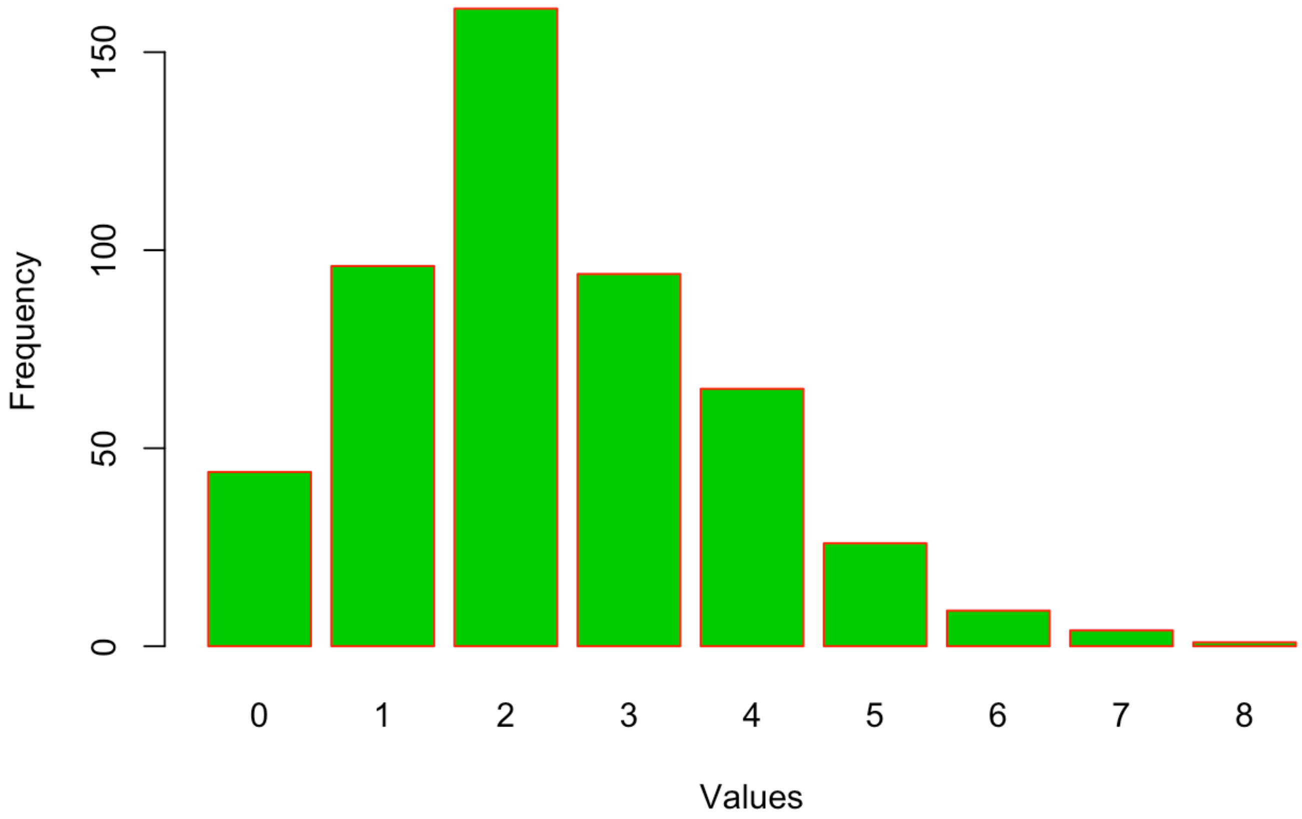

```
barplot(table(x)/n, ylim = c(0,1), main = "Bar plot")
```



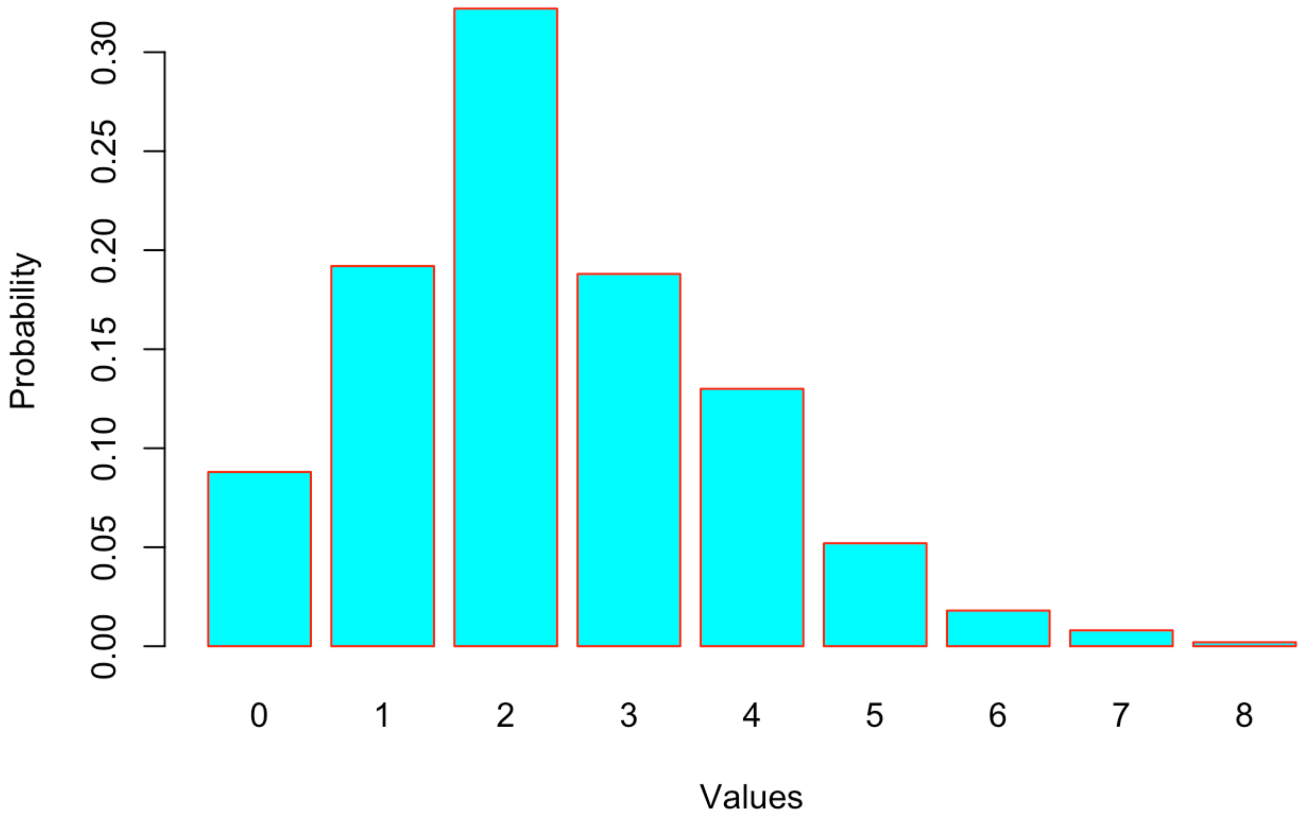

```
# Poisson data
data_pois<-rpois(n = 500,lambda = 2.54)
## TABLE of DISCRETE DATA
t<- table(data_pois)
print(t)
```

```
## data_pois
##   0   1   2   3   4   5   6   7   8
##  44  96 161  94  65  26   9   4   1
```

```
# jpeg("barplot.jpeg")
barplot(t,xlab = "Values", ylab = "Frequency", horiz = F, col = 3, border = 2)
```




```
barplot(t/sum(t),xlab = "Values", ylab = "Probability", horiz = F, col=5 ,border = 
2)
```




```
# dev.off()
##  MEAN
data_mean<-mean(data_pois)
cat("Mean of the data=", data_mean,"\n")
```

```
## Mean of the data= 2.36
```

```
## VARIANCE
data_var<-var(data_pois)
cat("Varicance of the data=", data_var,"\n")
```

```
## Varicance of the data= 2.154709
```

```
## STANDARD DEVIATION
data_sd<-sd(data_pois)
cat("Standred deviation of the data=", data_sd,"\n")
```

```
## Standred deviation of the data= 1.467893
```

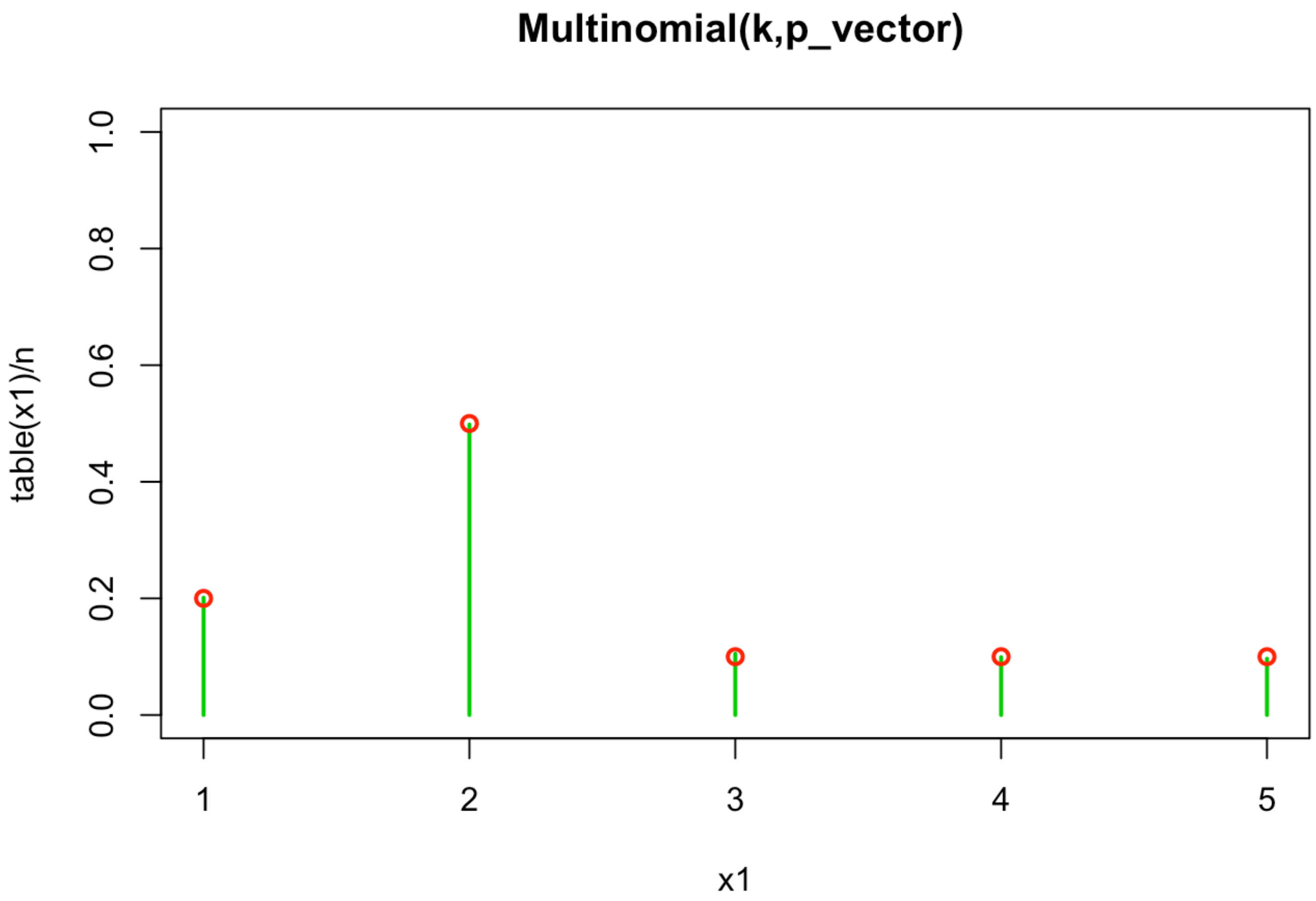
```r
#######################
print(summary(data_pois))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    1.00    2.00    2.36    3.00    8.00
```

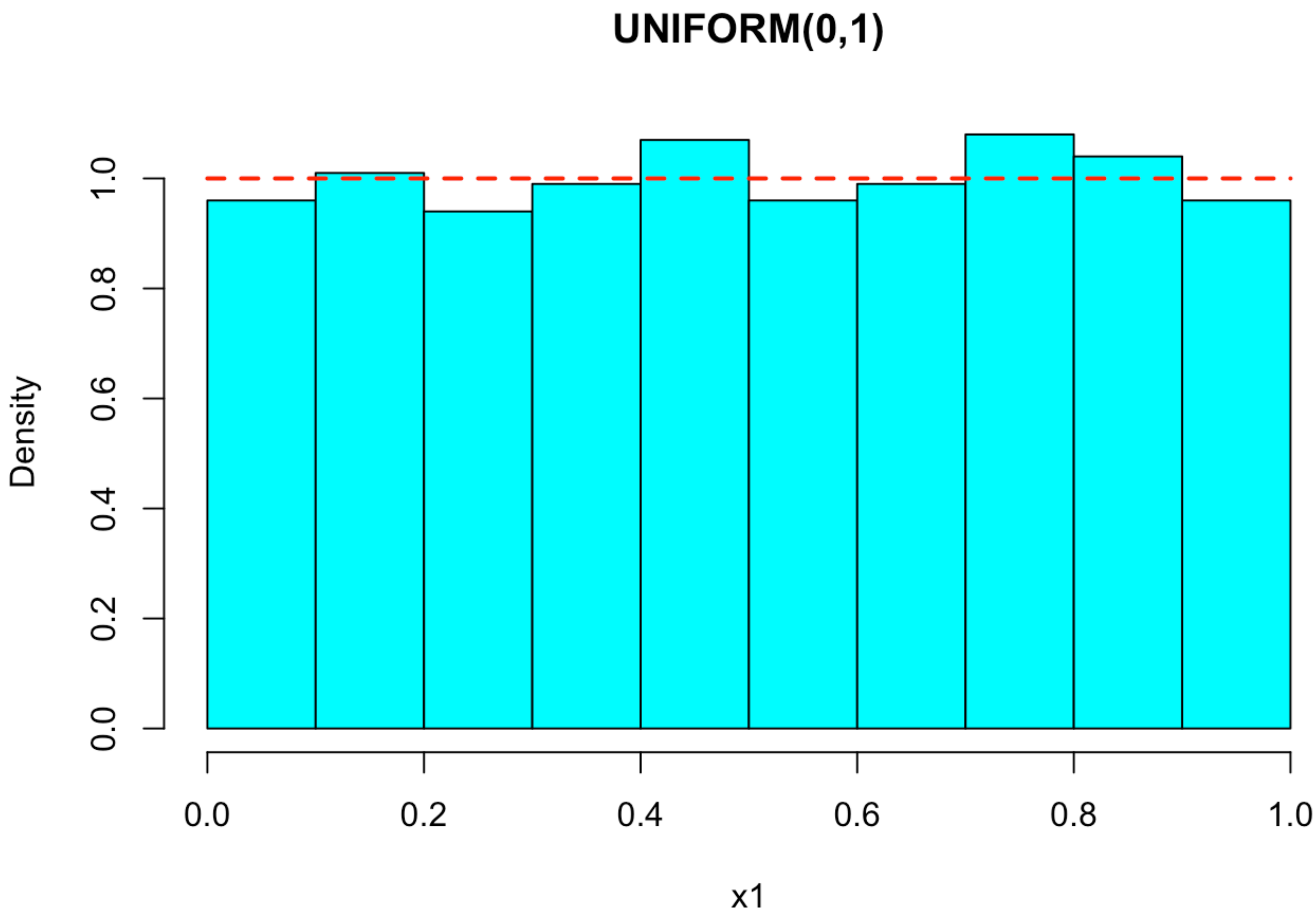
```r
#dev.off()





##############################
#Multinomial(k,p_vector)
##############################
k <- 5  # categories
n <- 10000 # sample size
p <- c(.2,.5,.1,.1,.1) # probbility vector
x1 <- sample(1:k, n, replace=T, prob=p) # data
print(table(x1)/n )
```

```
## x1
##      1      2      3      4      5 
## 0.2012 0.4981 0.1051 0.0991 0.0965
```

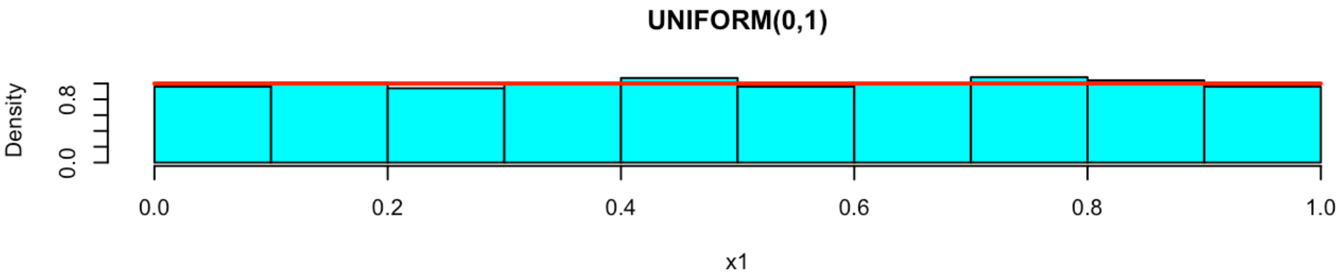
```r
par(mfrow=c(1,1))
plot(table(x1)/n,ylim=c(0,1), col=3, main = " Multinomial(k,p_vector)")
lines(p, type='p', col=2, lwd=2)
```

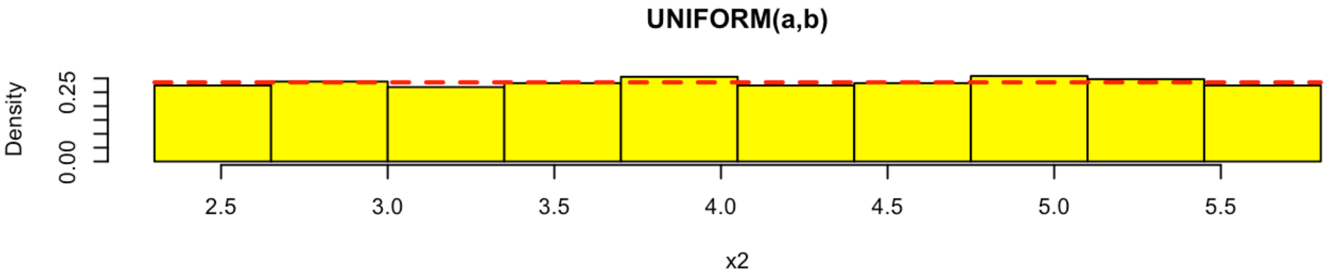


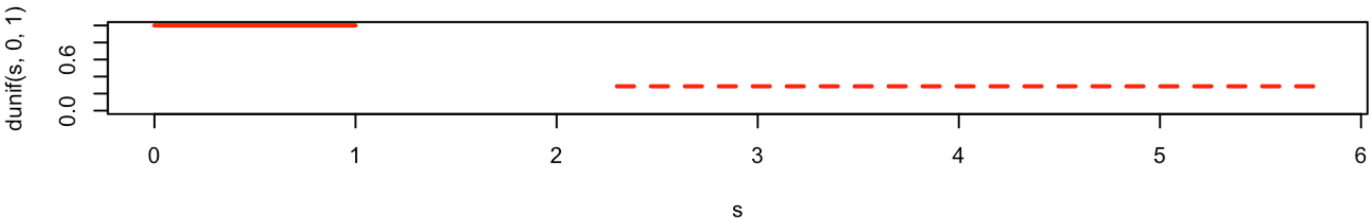

```
###############################
#UNIFORM(a,b)
###############################
a<-0
b<-1
n<- 1000 #Sample size
x1<-runif(n, a, b)
s<-seq(a,b,length=11)
par(mfrow=c(1,1))
hist(x1, col=5,probability = T, breaks =s, main='UNIFORM(0,1)')
lines(dunif(s,a,b)~s, col=2, lwd=2, lty=2)
```



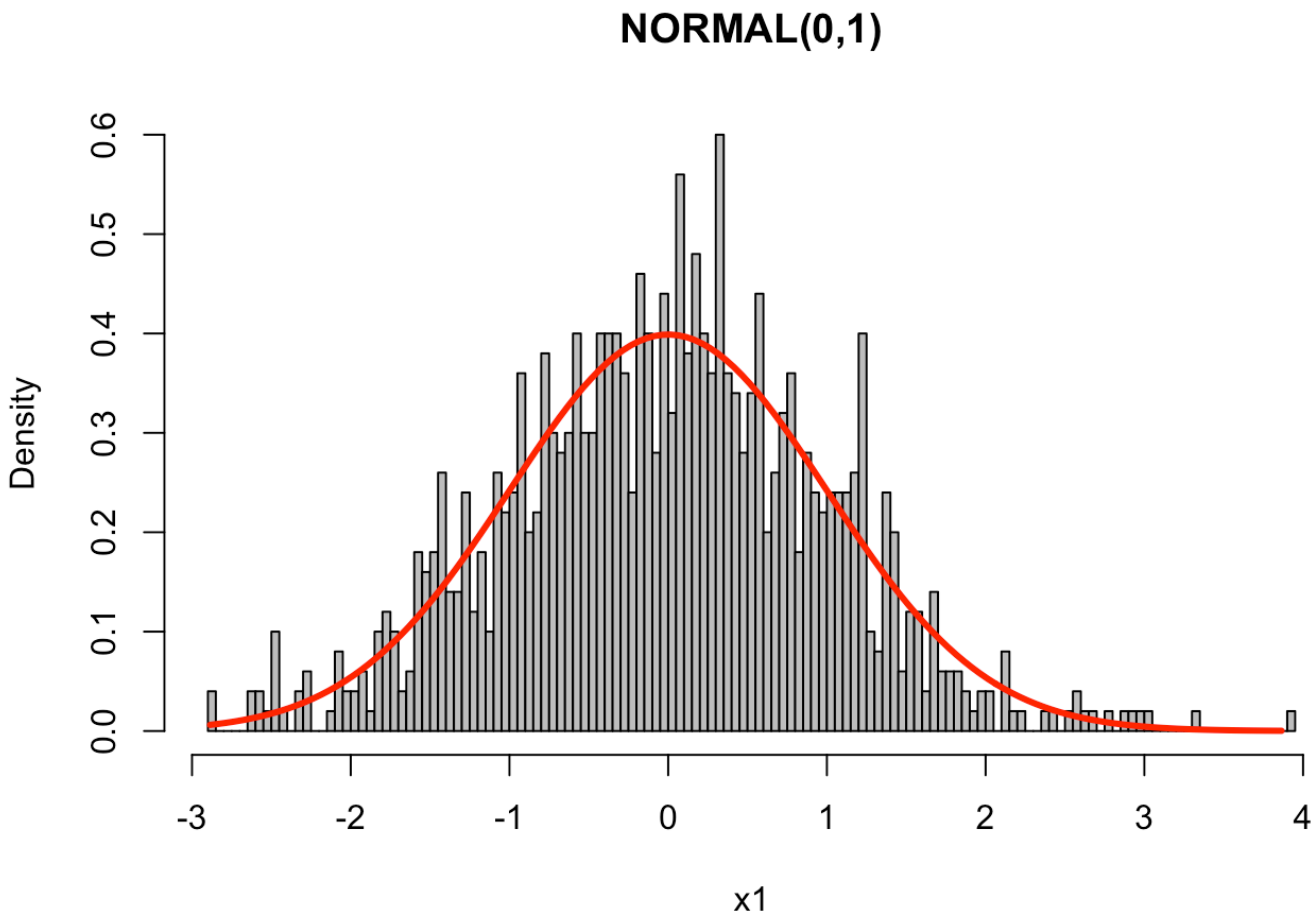

```
a<-2.3
b<-5.8

x2<-a+(b-a)*x1
ss<-a+s*(b-a)


par(mfrow=c(3,1))
hist(x1, col=5,probability = T, breaks =s, main='UNIFORM(0,1)')
lines(dunif(s,0,1)~s, col=2, lwd=2, lty=1)

hist(x2, col=7,probability = T, breaks =ss, main='UNIFORM(a,b)')
lines(dunif(ss,a,b)~ss, col=2, lwd=2, lty=2)

plot(dunif(s,0,1)~s, xlim=c(0,b),ylim=c(0,1), col=2, lwd=2, type='l')
lines(dunif(ss,a,b)~ss, col=2, lwd=2, lty=2)
```



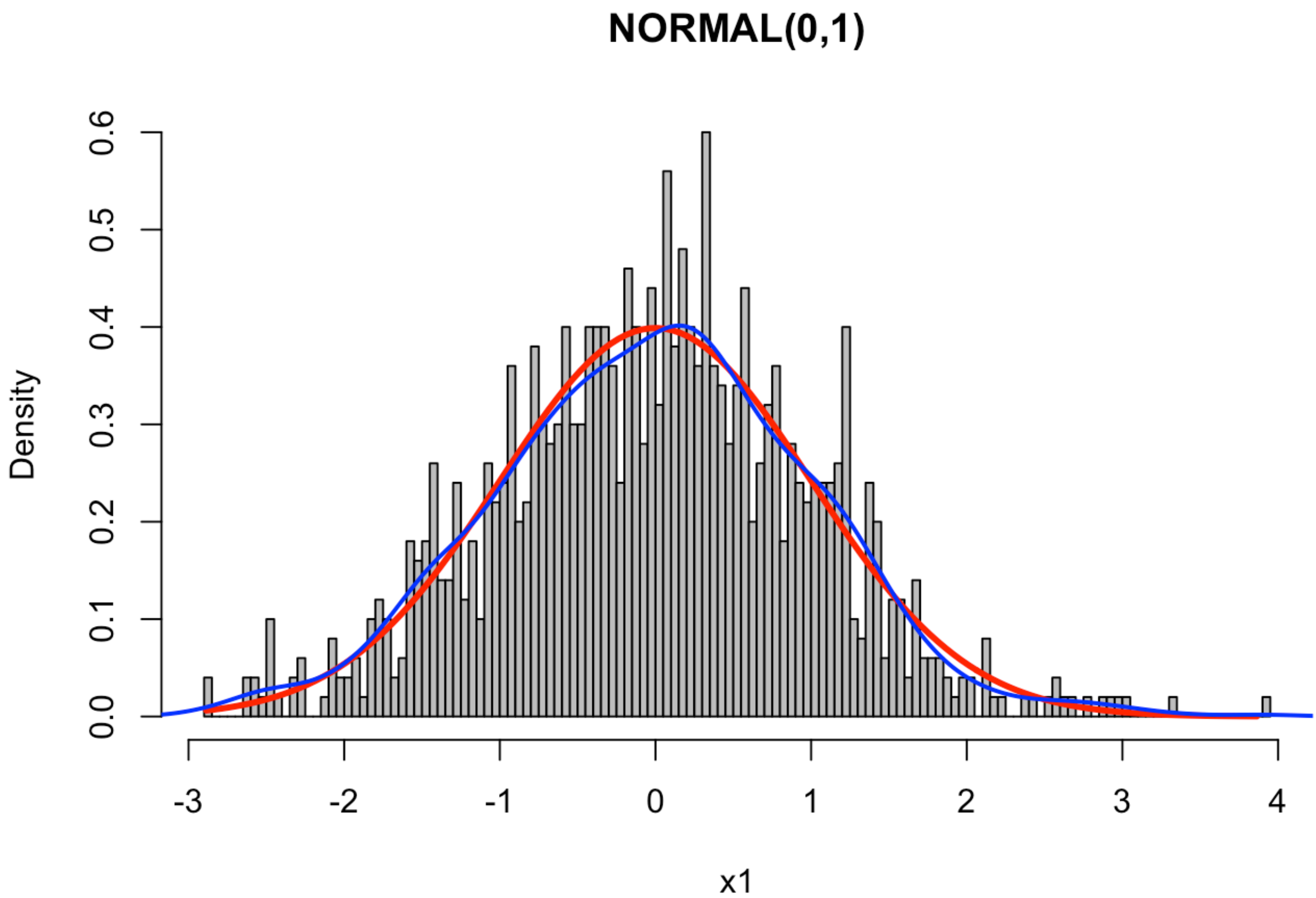




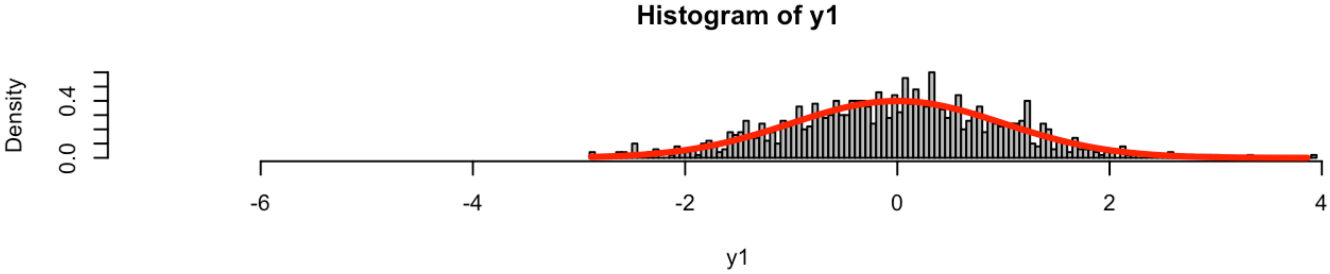

```
##################################
# NORMAL(mu, sigma)
##################################

mu=0   # location
sigma=1 # scale

n<-1000
x1<-rnorm(n, mu,sigma)
s<-seq(min(x1),max(x1),by=0.05)


par(mfrow=c(1,1))
hist(x1,probability = T,breaks = 100, col=8, main='NORMAL(0,1)')
lines(dnorm(s, mean=0, sd=1)~s, col=2, lwd=3)
```

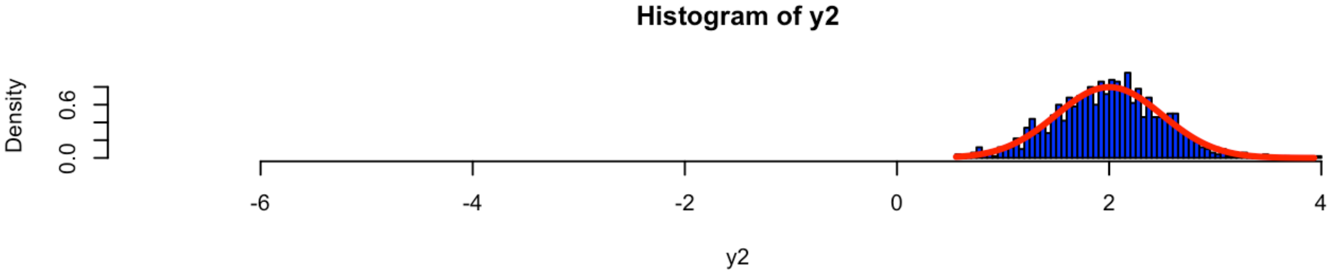


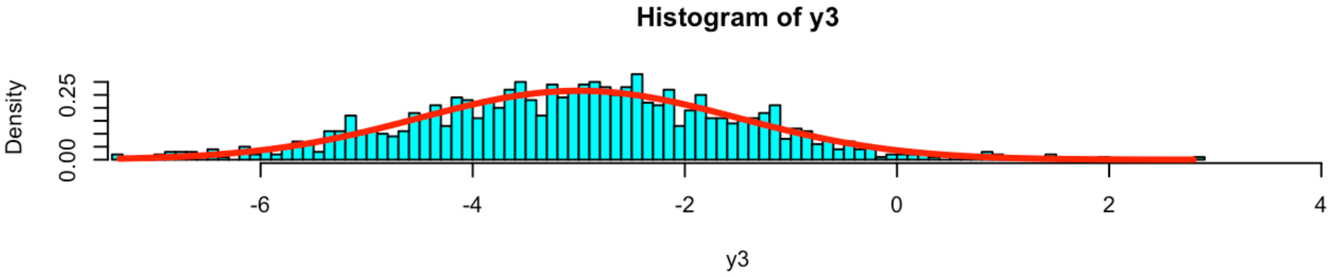

```
par(mfrow=c(1,1))
hist(x1,probability = T,breaks = 100, col=8, main='NORMAL(0,1)')
lines(dnorm(s, mean=0, sd=1)~s, col=2, lwd=3)
lines(density(x1), col=4, lwd=2)
```




```
y1<-x1
s1<-s
y2<-2+0.5*x1
s2<-2+0.5*s
y3<- -3+1.5*x1
s3<- -3+1.5*s

par(mfrow=c(3,1))
hist(y1,probability = T,breaks = 100, col=8, xlim=c(-7,4))
lines(dnorm(s1, mean=0, sd=1)~s1, col=2, lwd=3)

hist(y2,probability = T,breaks = 100, col=4, xlim=c(-7,4))
lines(dnorm(s2, mean=2, sd=0.5)~s2, col=2, lwd=3)
hist(y3,probability = T,breaks = 100, col=5, xlim=c(-7,4))
lines(dnorm(s3, mean= -3, sd=1.5)~s3, col=2, lwd=3)
```








```
cat("mean(Y1)=", mean(y1), "sd(Y1)=",sd(y1),"\n")
```

```
## mean(Y1)= -0.01053586 sd(Y1)= 1.007734
```

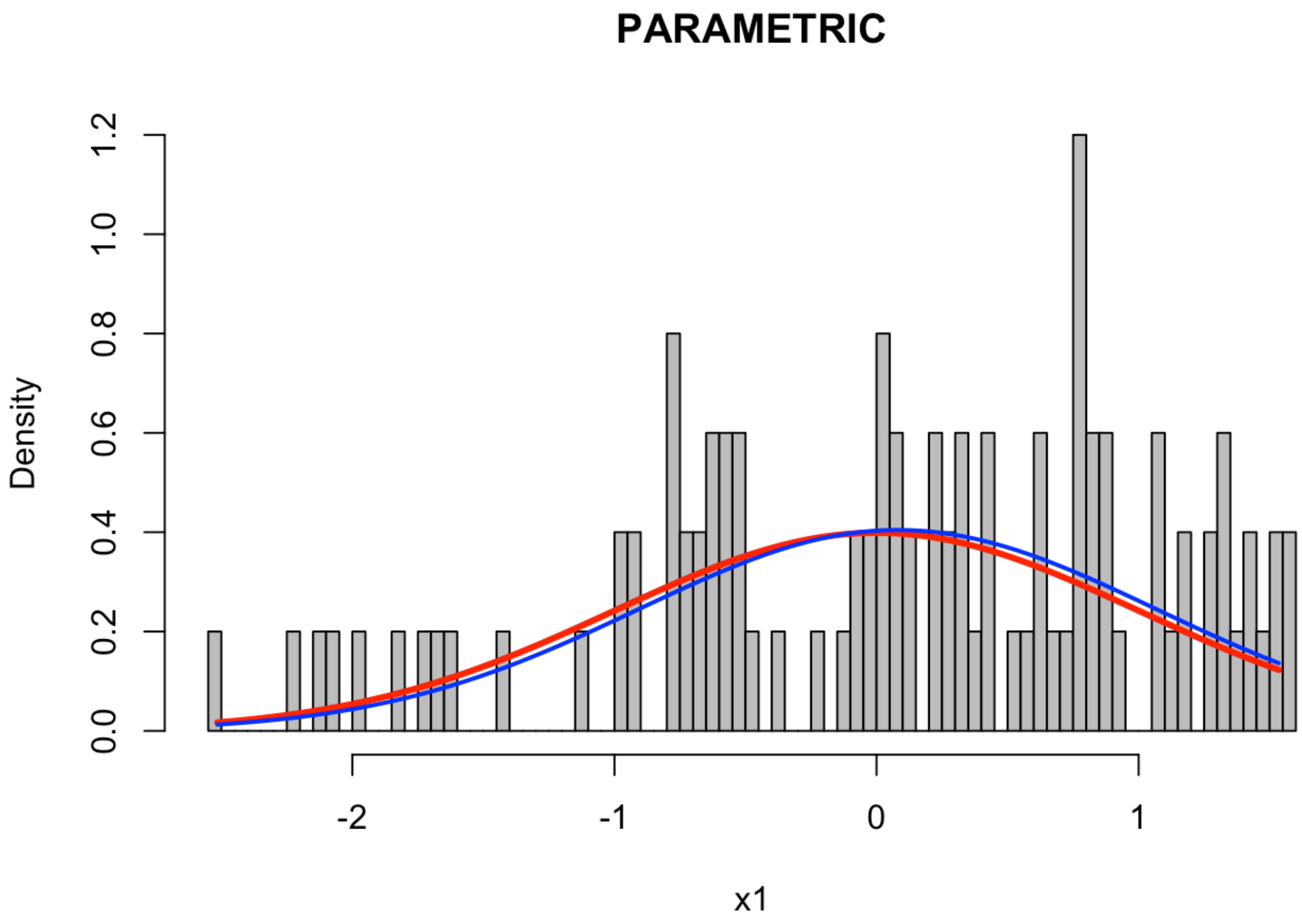
```
cat("mean(Y2)=", mean(y2), "sd(Y2)=",sd(y2),"\n")
```

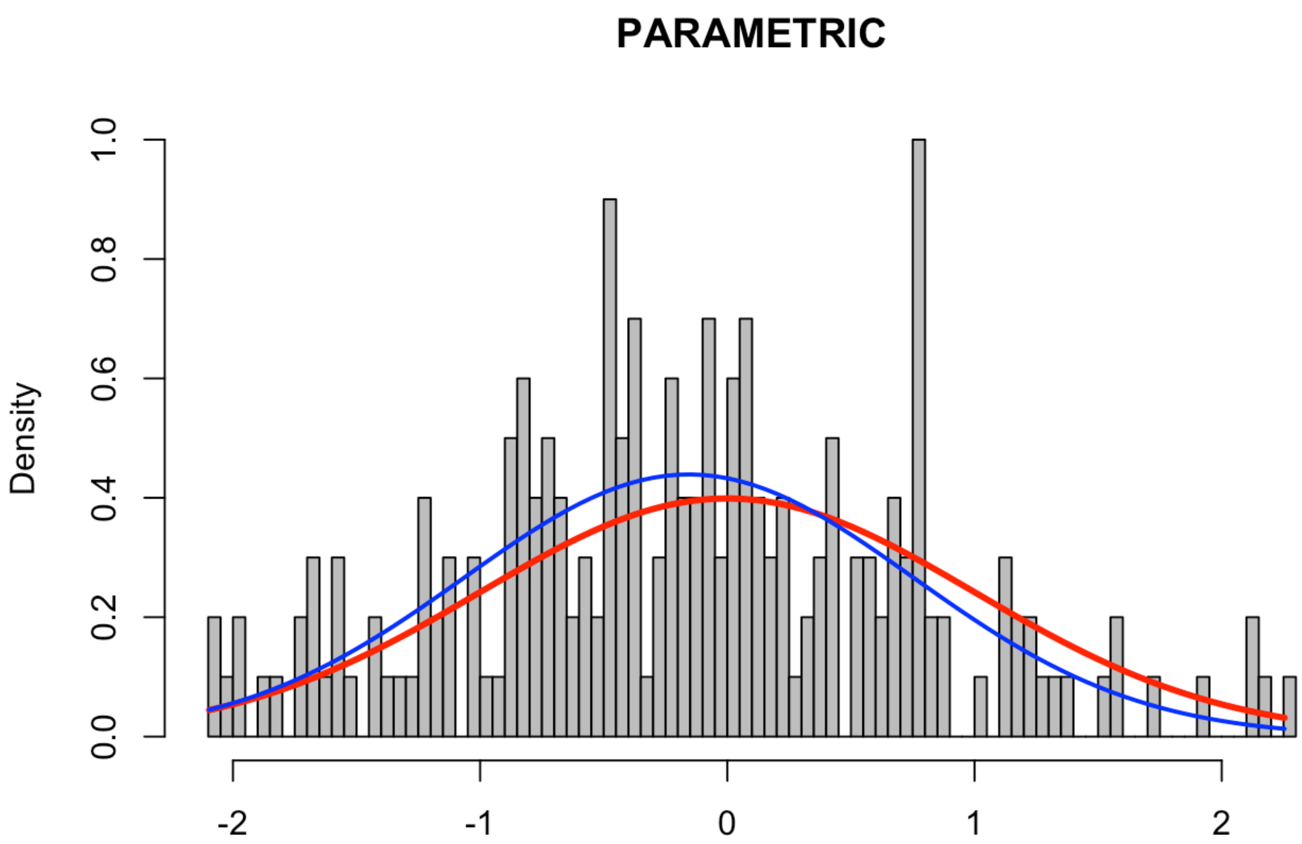
```
## mean(Y2)= 1.994732 sd(Y2)= 0.5038669
```

```
cat("mean(Y3)=", mean(y3), "sd(Y2)=",sd(y3),"\n")
```

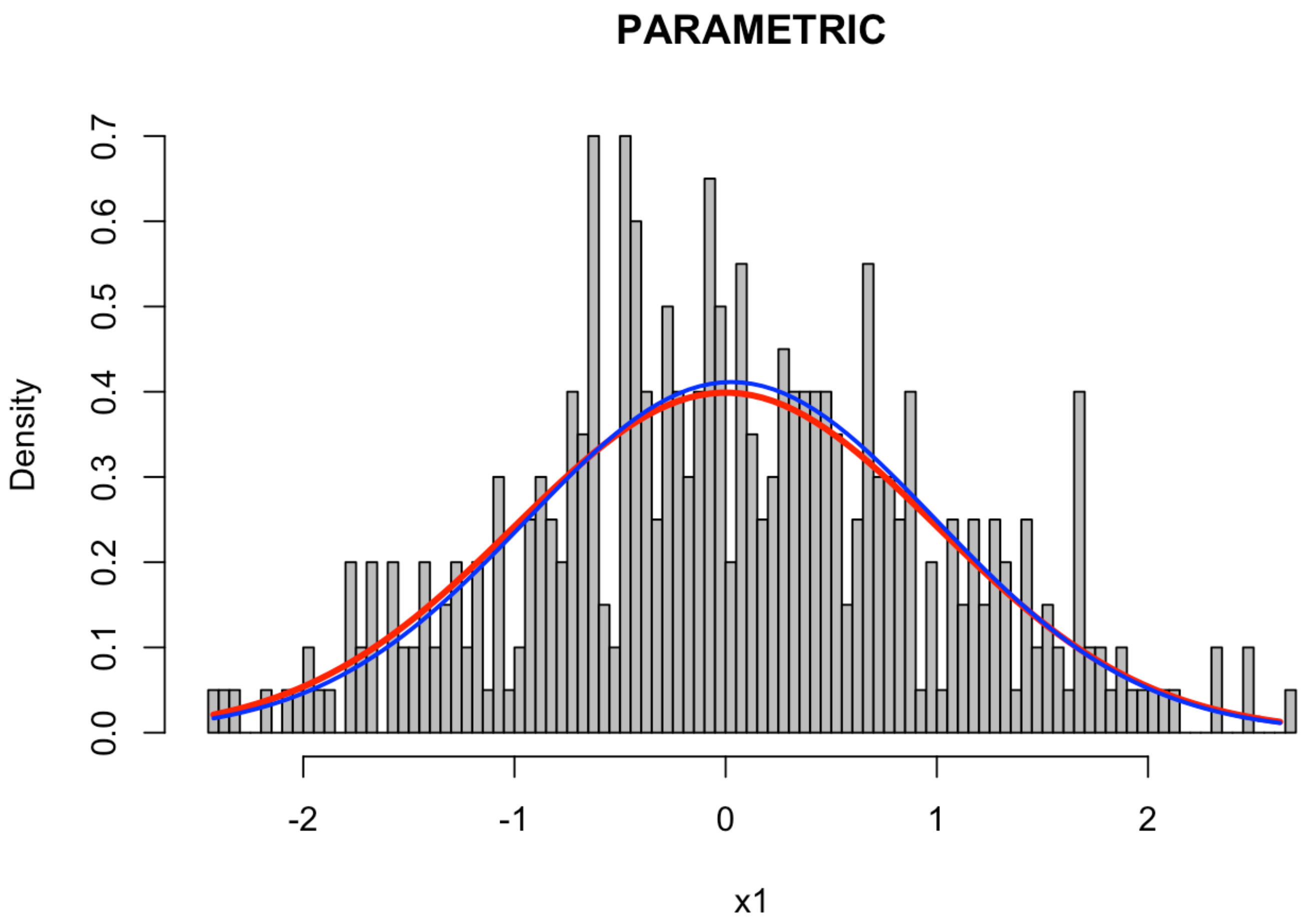
```
## mean(Y3)= -3.015804 sd(Y2)= 1.511601
```

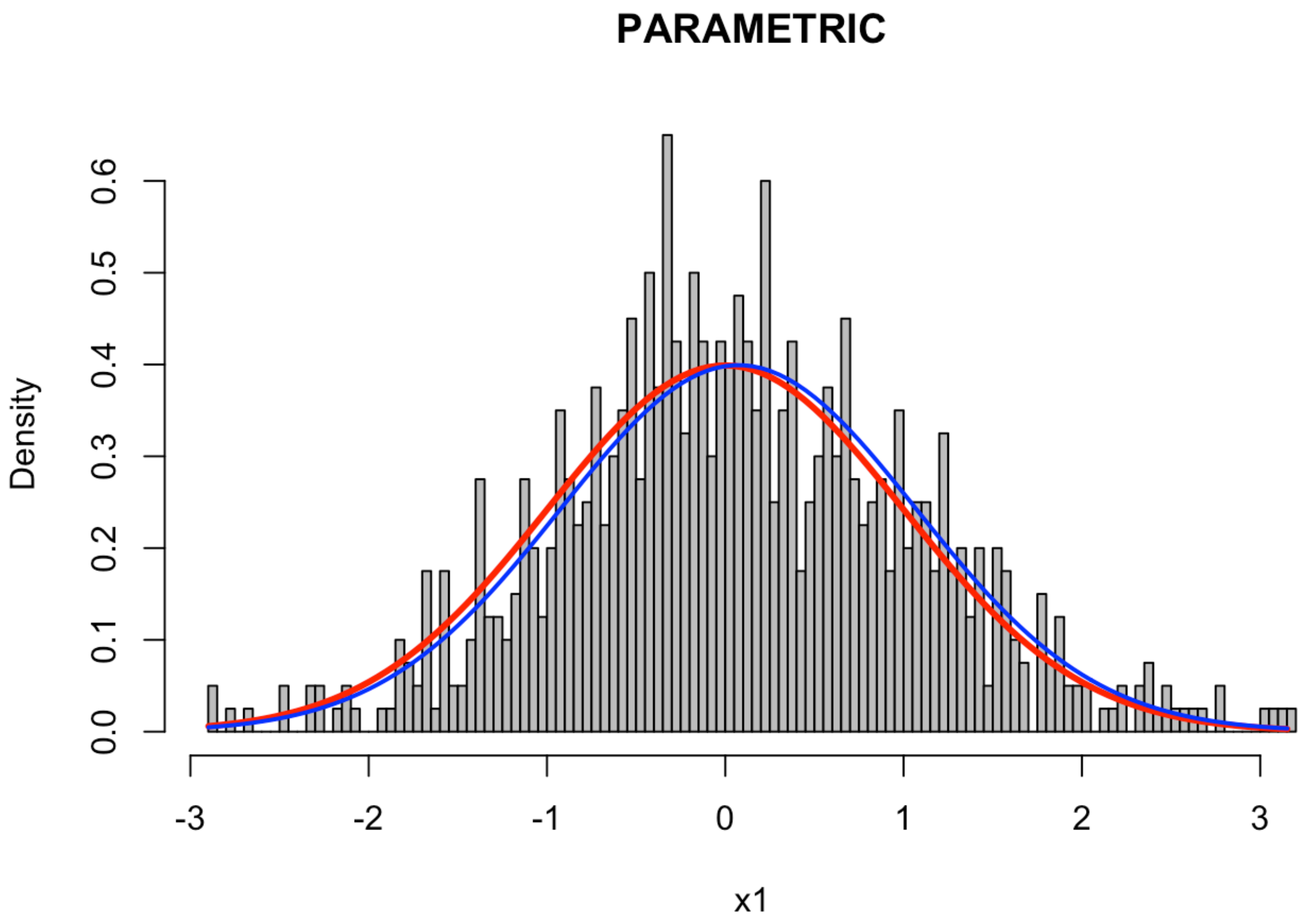
```
##### ESTIMATION ########

########## MLE ###############
for(i in 1 : 16){
  n<-50*2^i
  x1<-rnorm(n, mu,sigma)
  s<-seq(min(x1),max(x1),by=0.05)
  par(mfrow=c(1,1))
  hist(x1,probability = T,breaks = 100, col=8, main="PARAMETRIC")
  lines(dnorm(s, mean=0, sd=1)~s, col=2, lwd=3)
  lines(dnorm(s, mean=mean(x1), sd=sd(x1))~s, col=4, lwd=2)
  Sys.sleep(2)
}
```

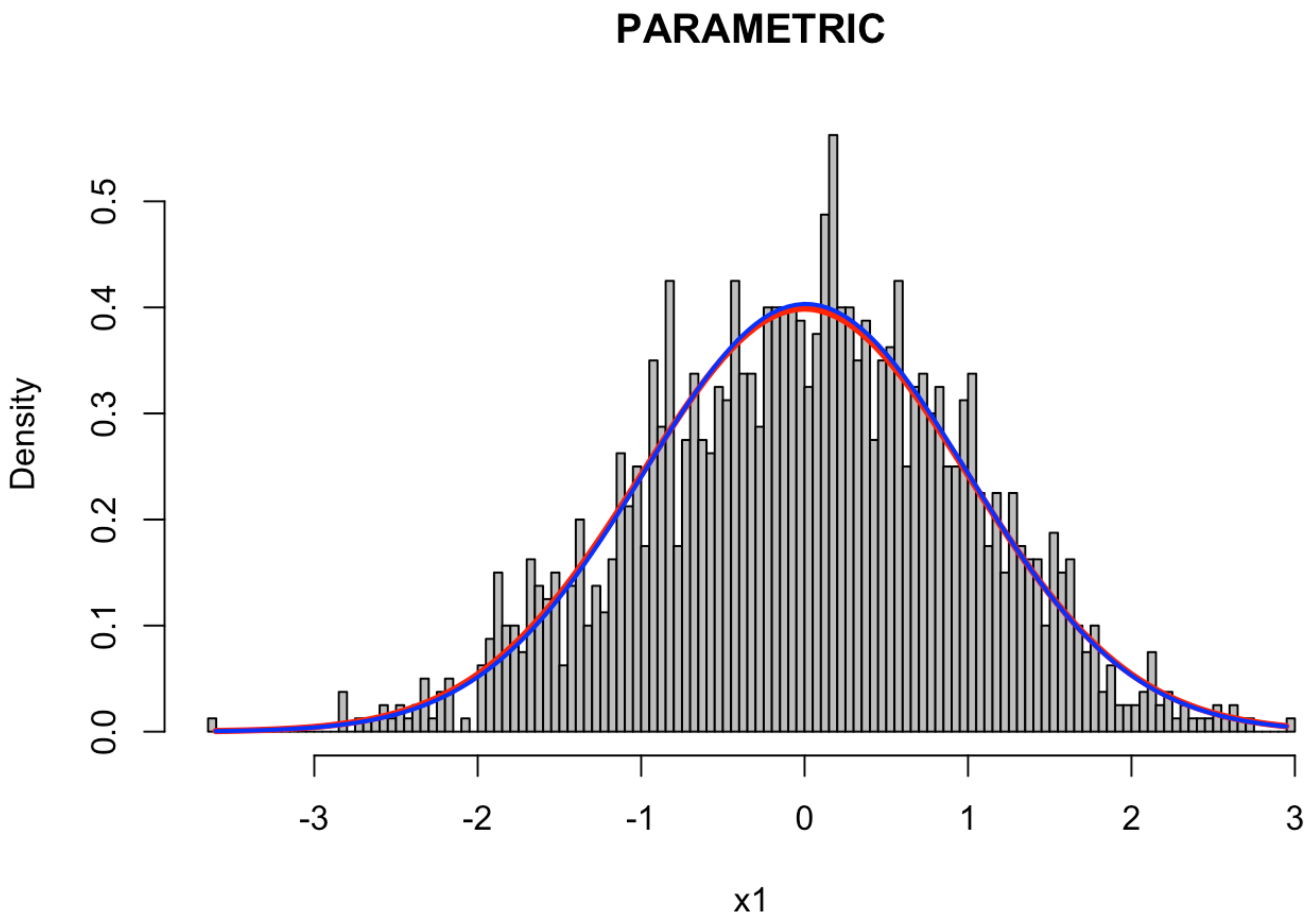

PARAMETRIC
Density
x1
0.0
0.2
0.4
0.6
0.8
1.0
1.2
-2
-1
0
1

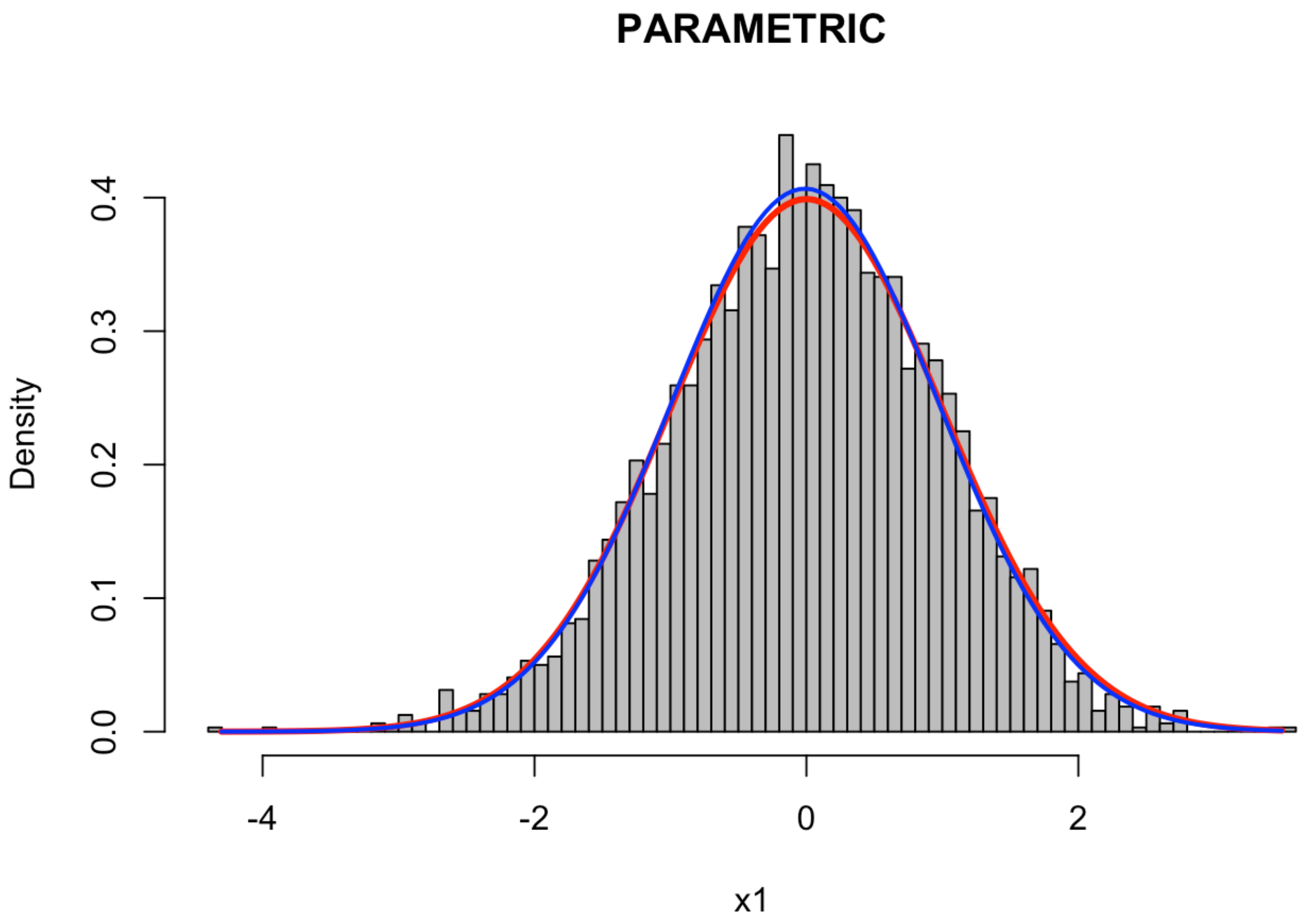

PARAMETRIC
Density
0.0
0.2
0.4
0.6
0.8
1.0
-2
-1
0
1
2

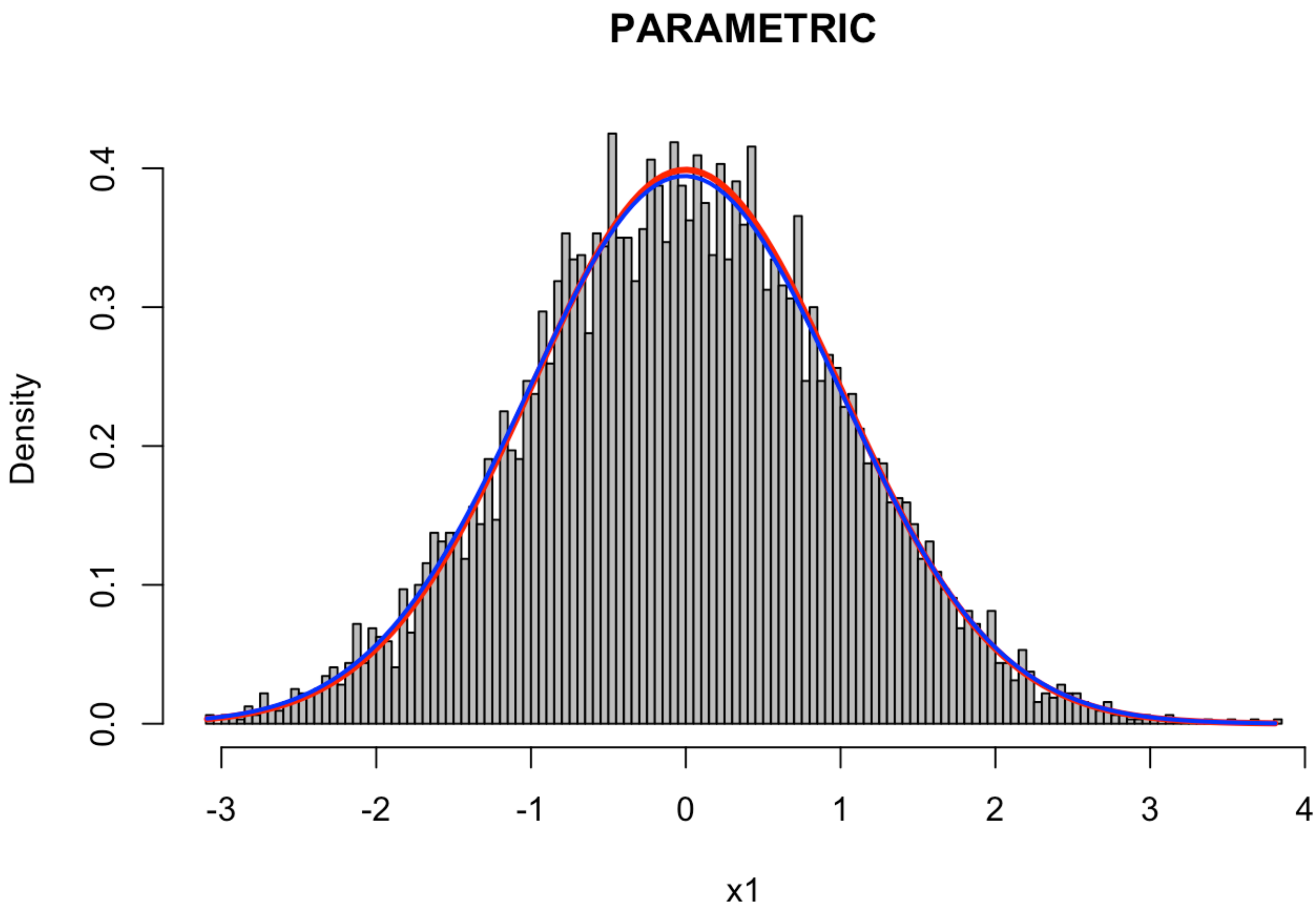
x1

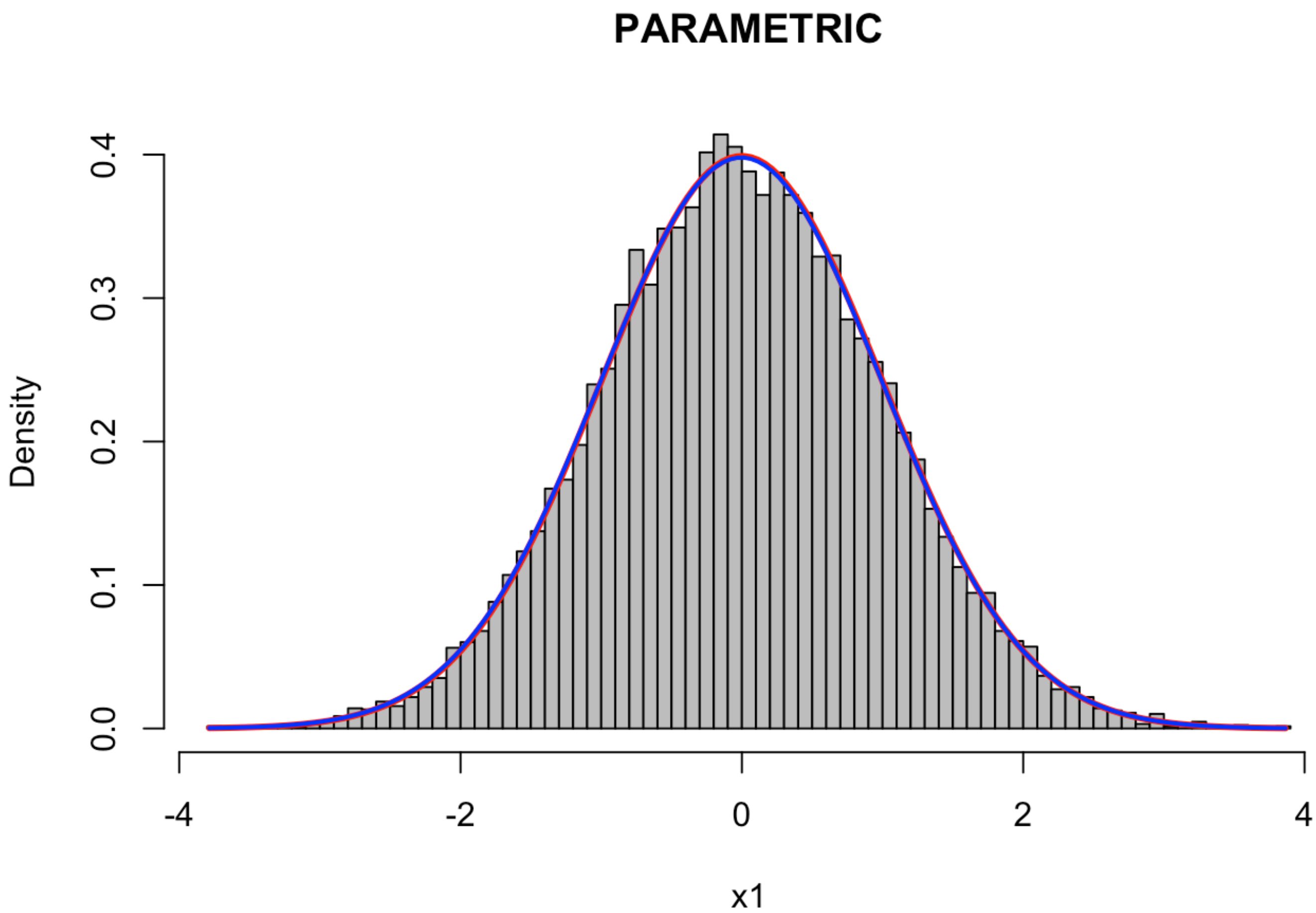


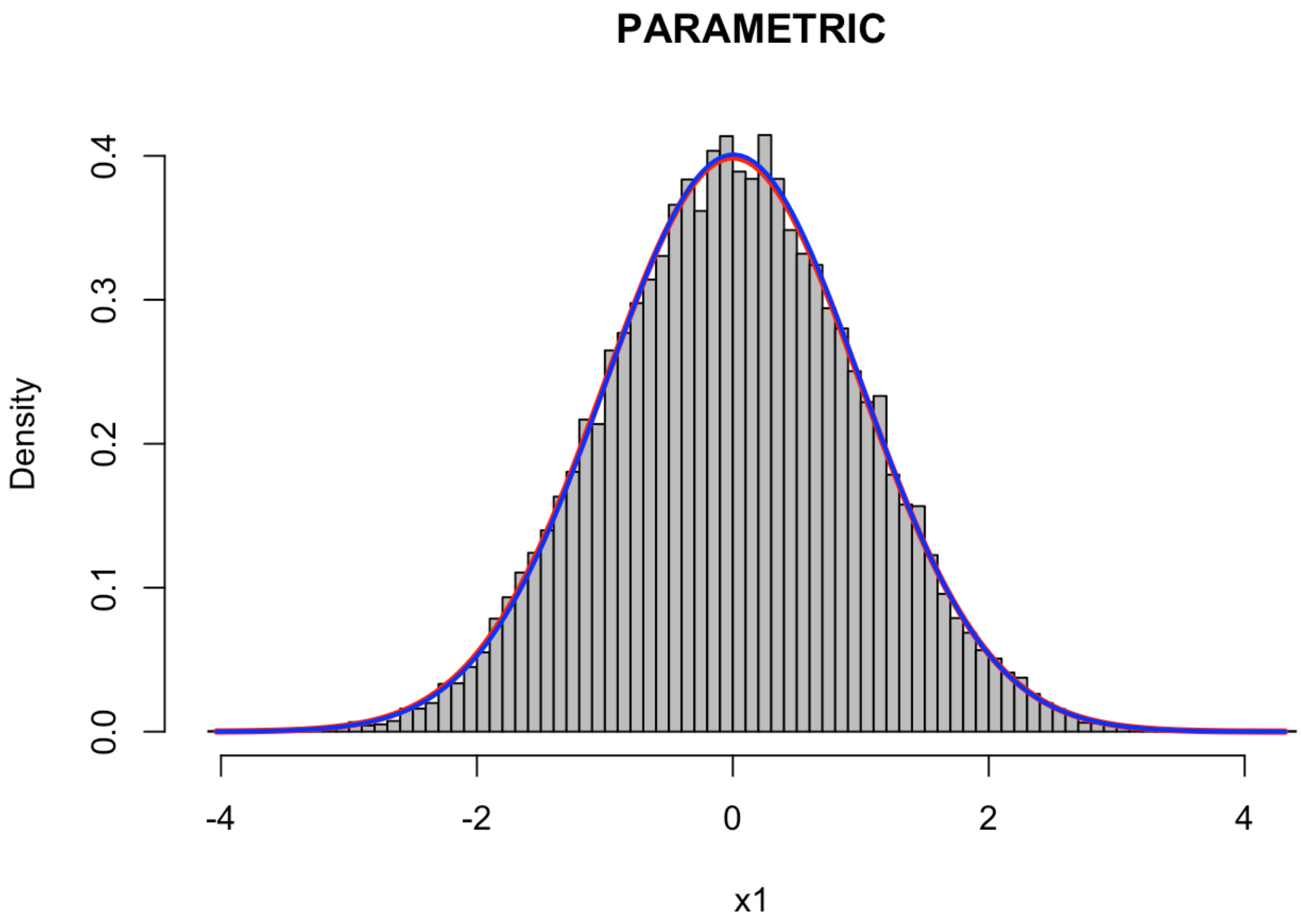

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
-3
-2
-1
0
1
2
3
x1

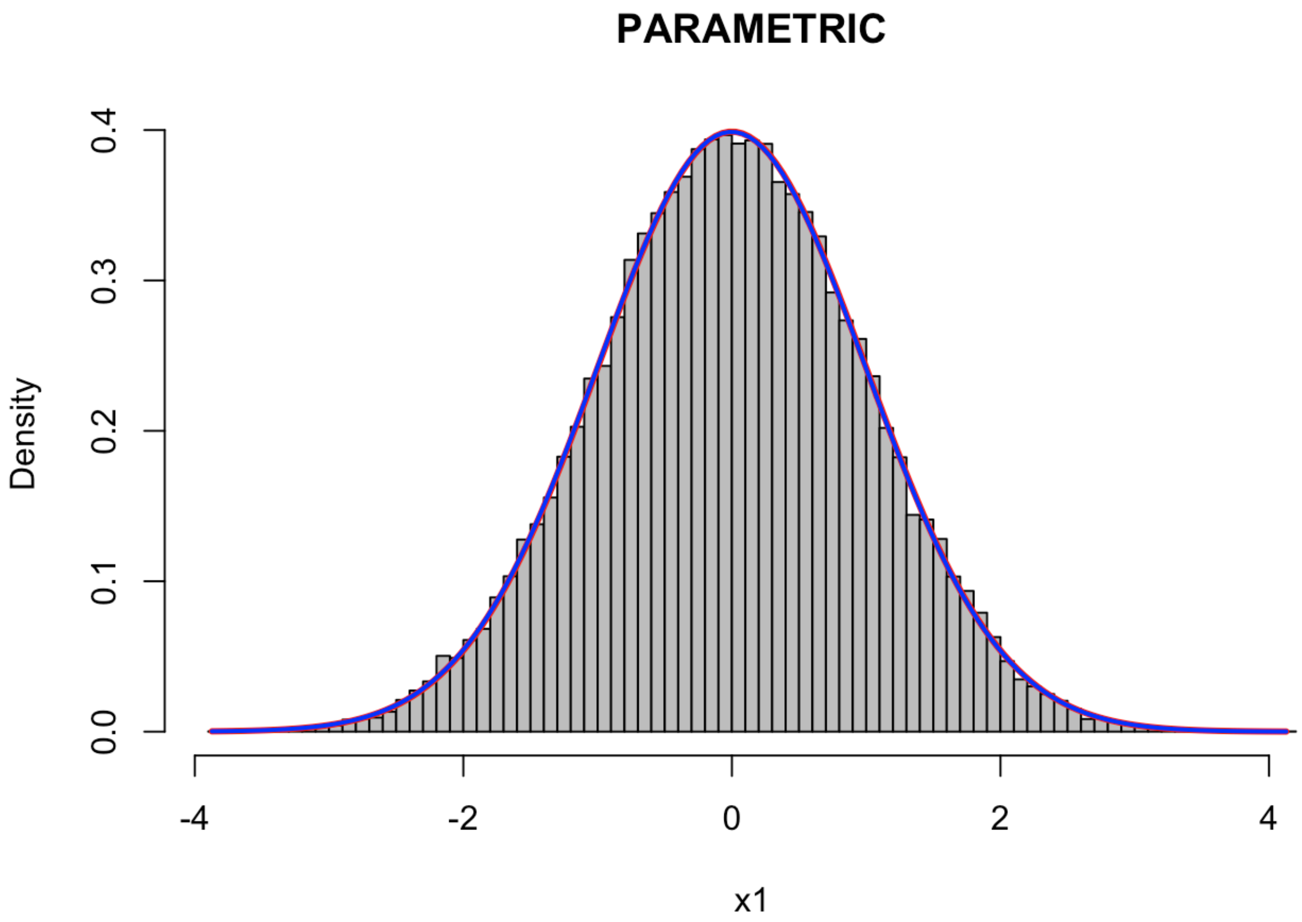

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
0.5
-3
-2
-1
0
1
2
3
x1

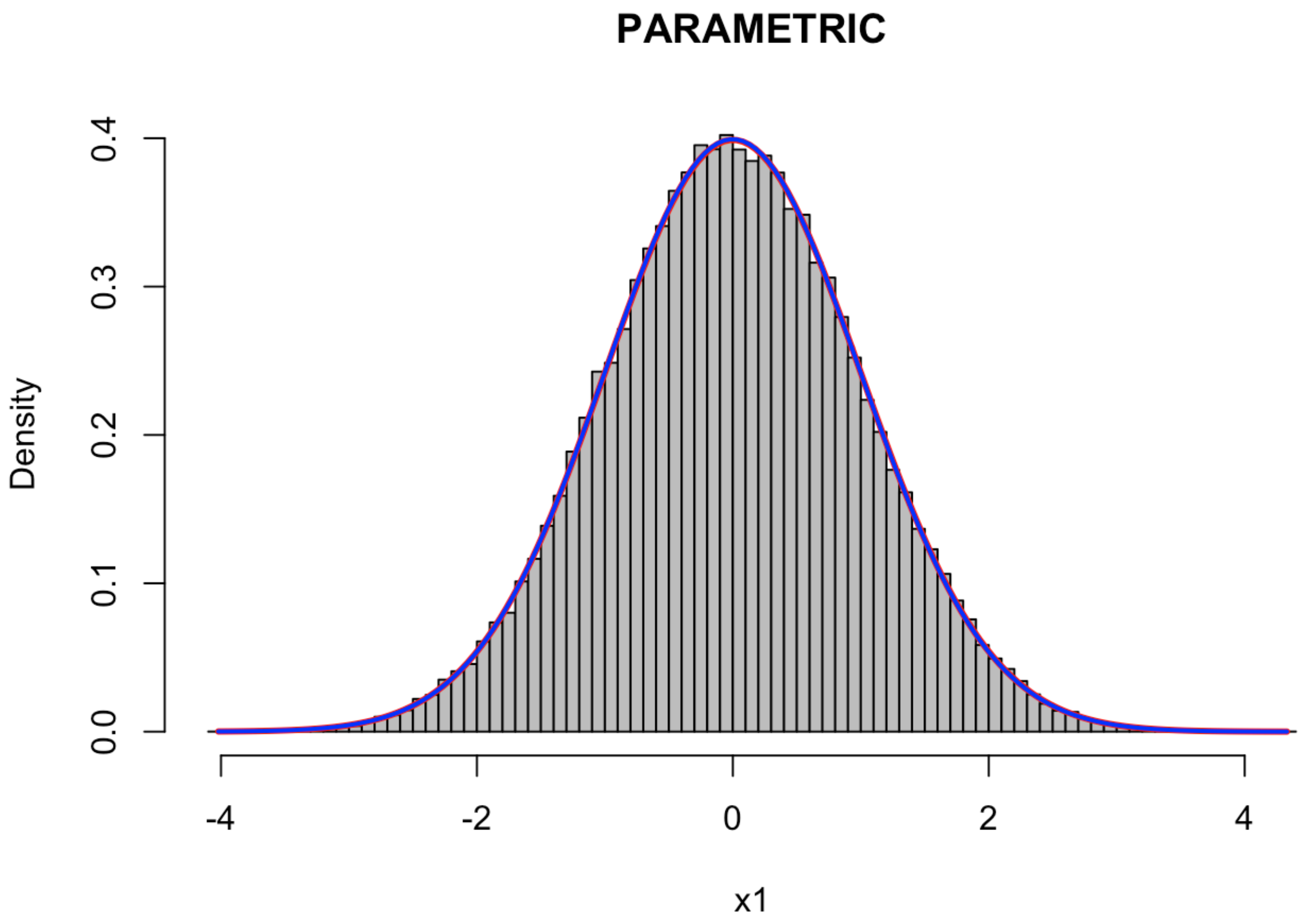

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
x1

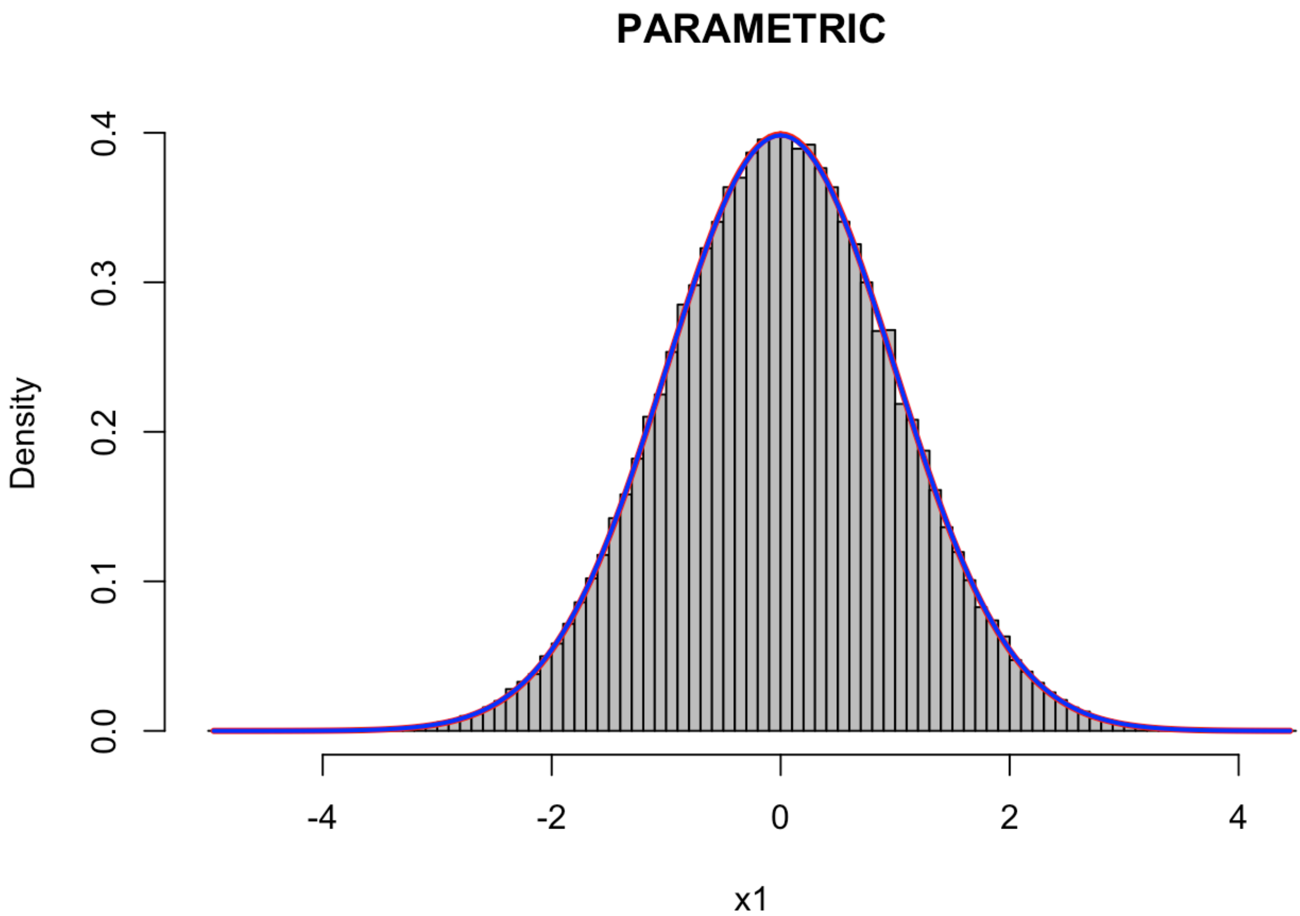

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-3
-2
-1
0
1
2
3
4
x1

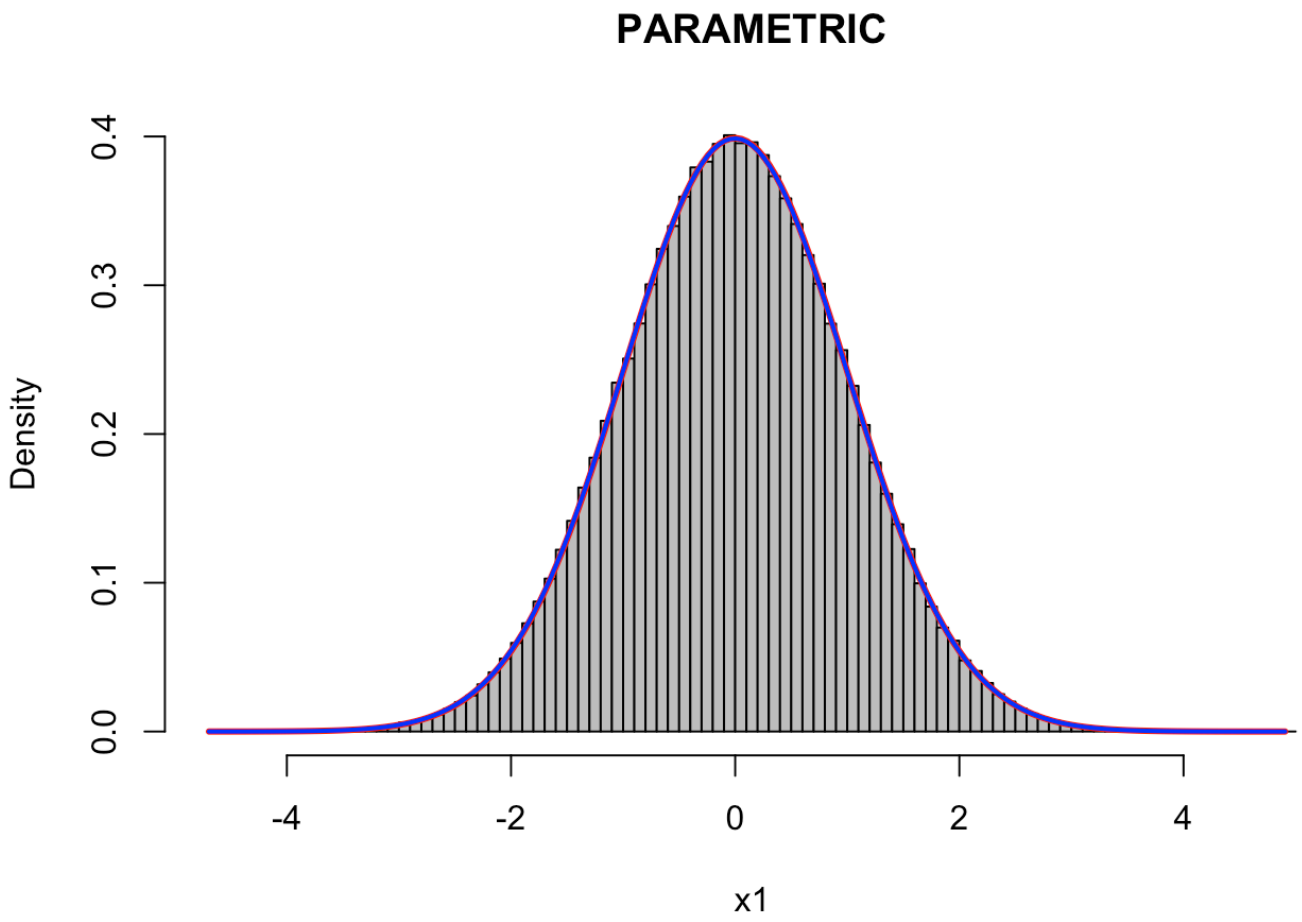

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

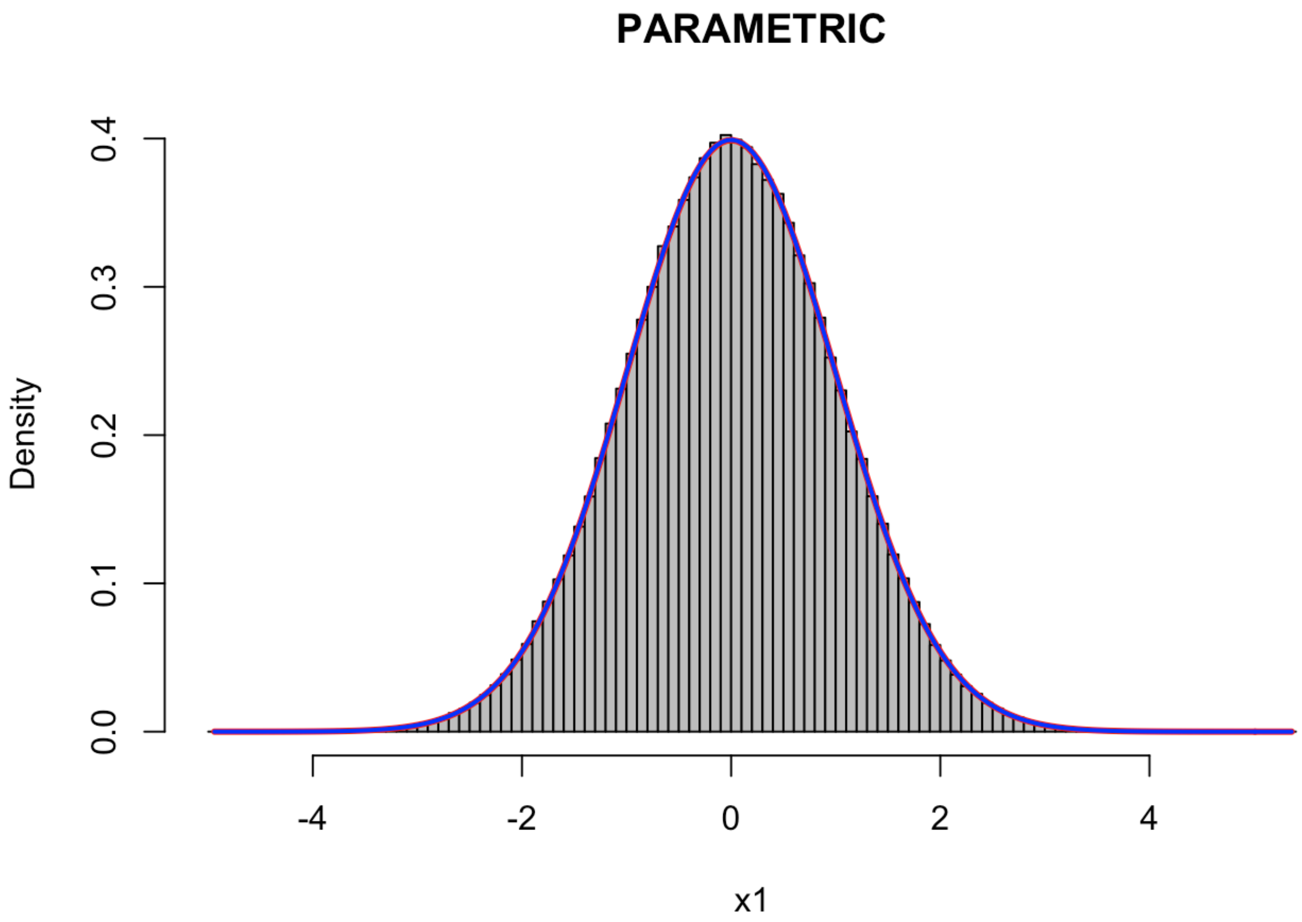

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

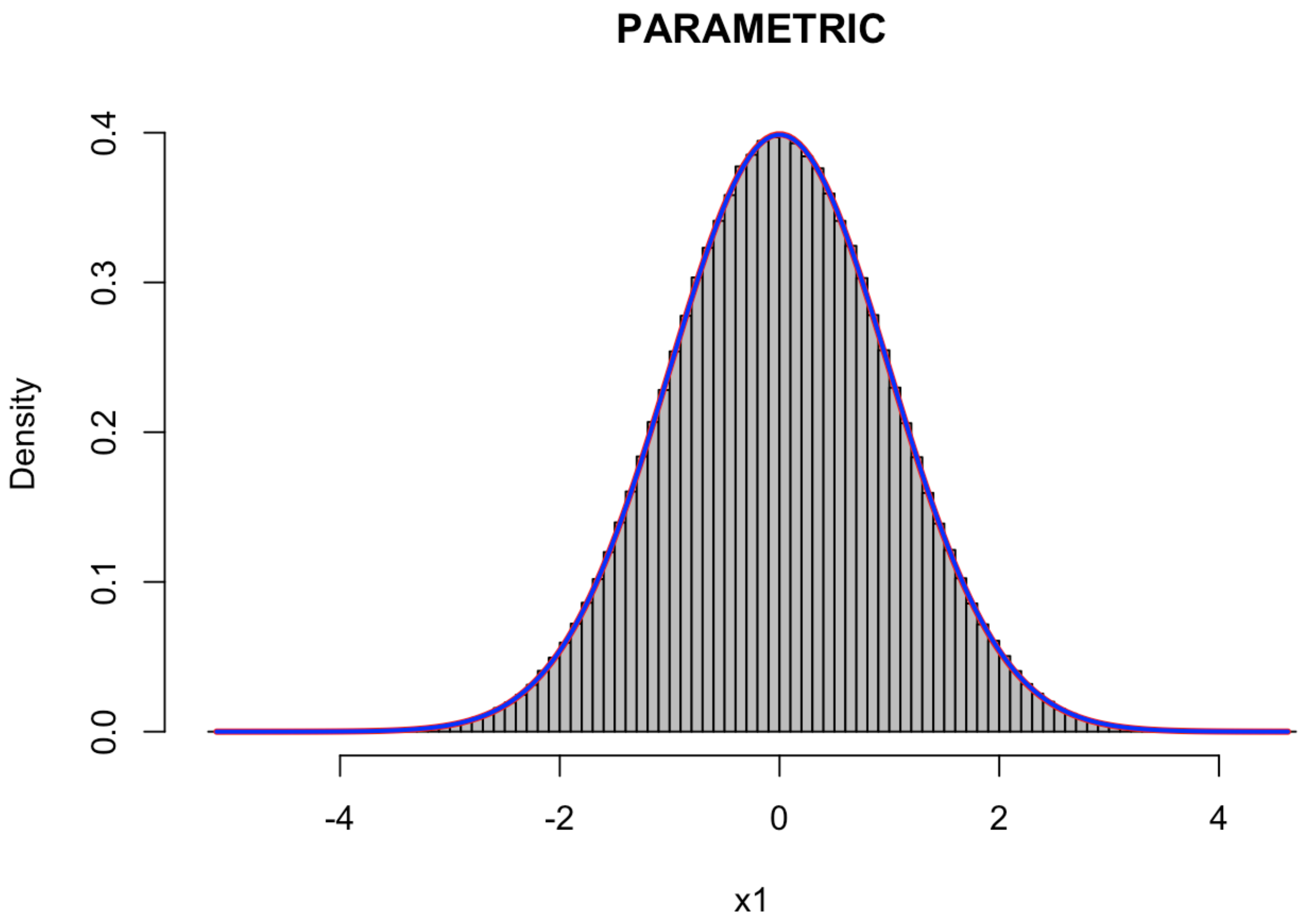

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1


PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

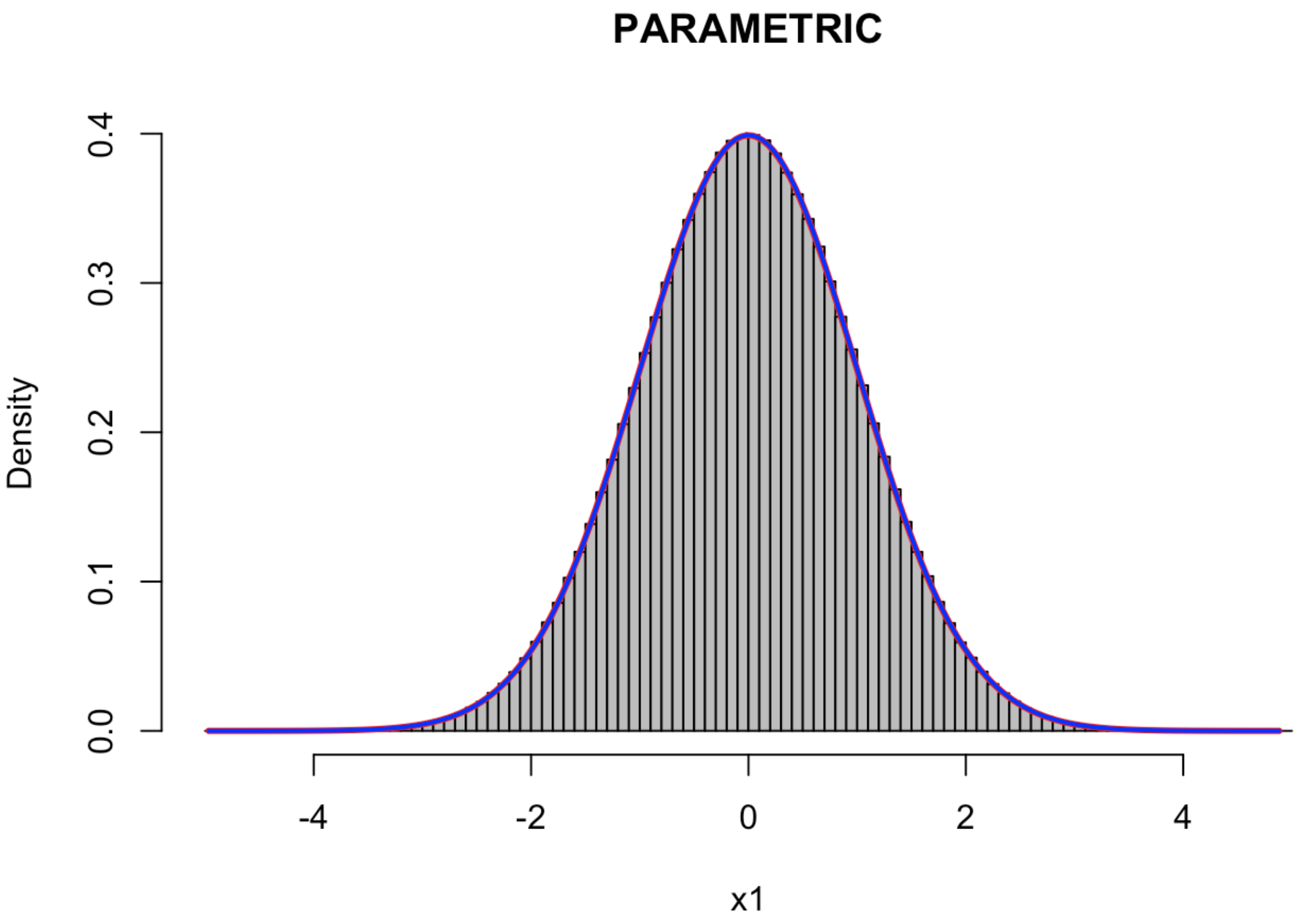

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1


PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

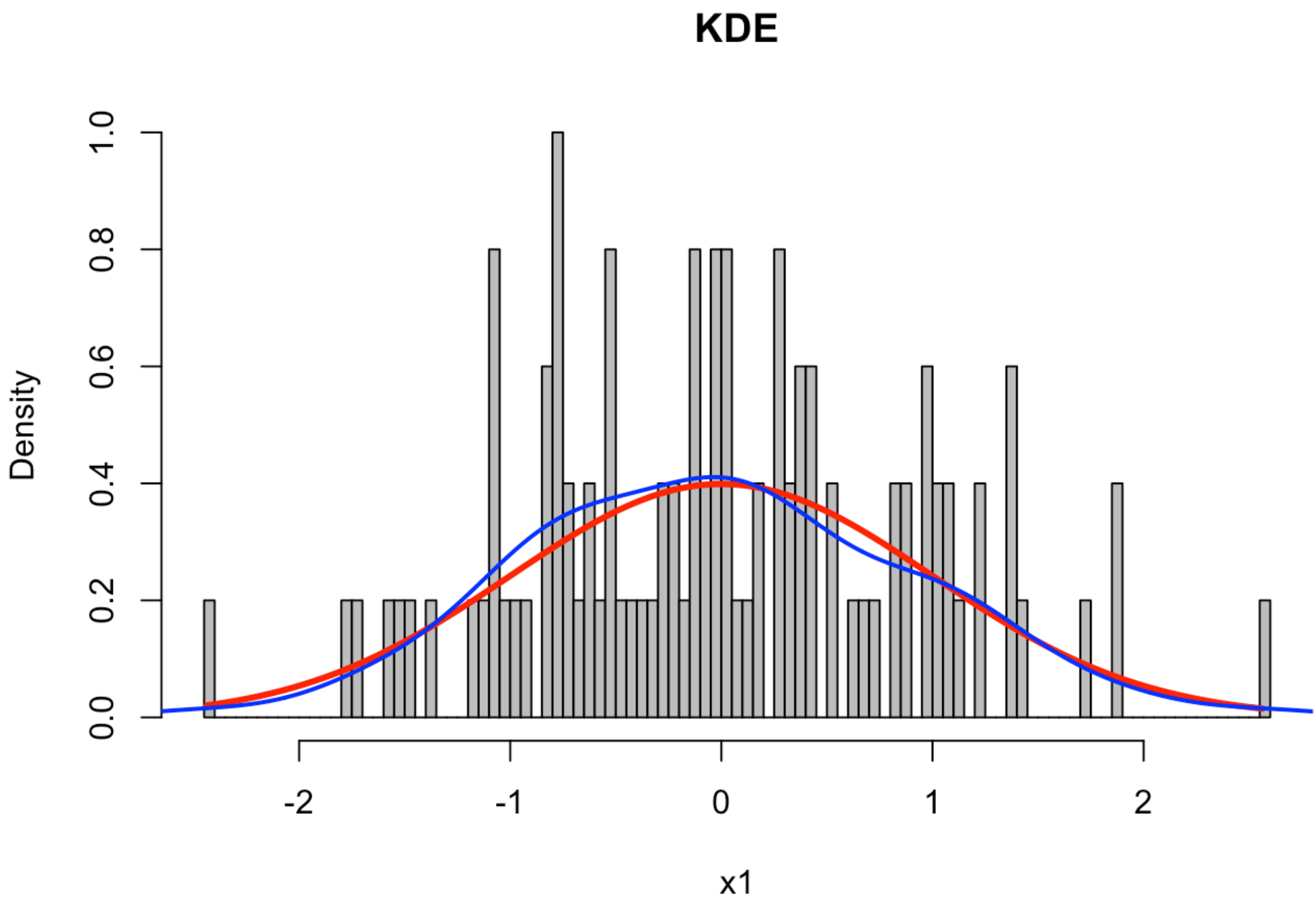

PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

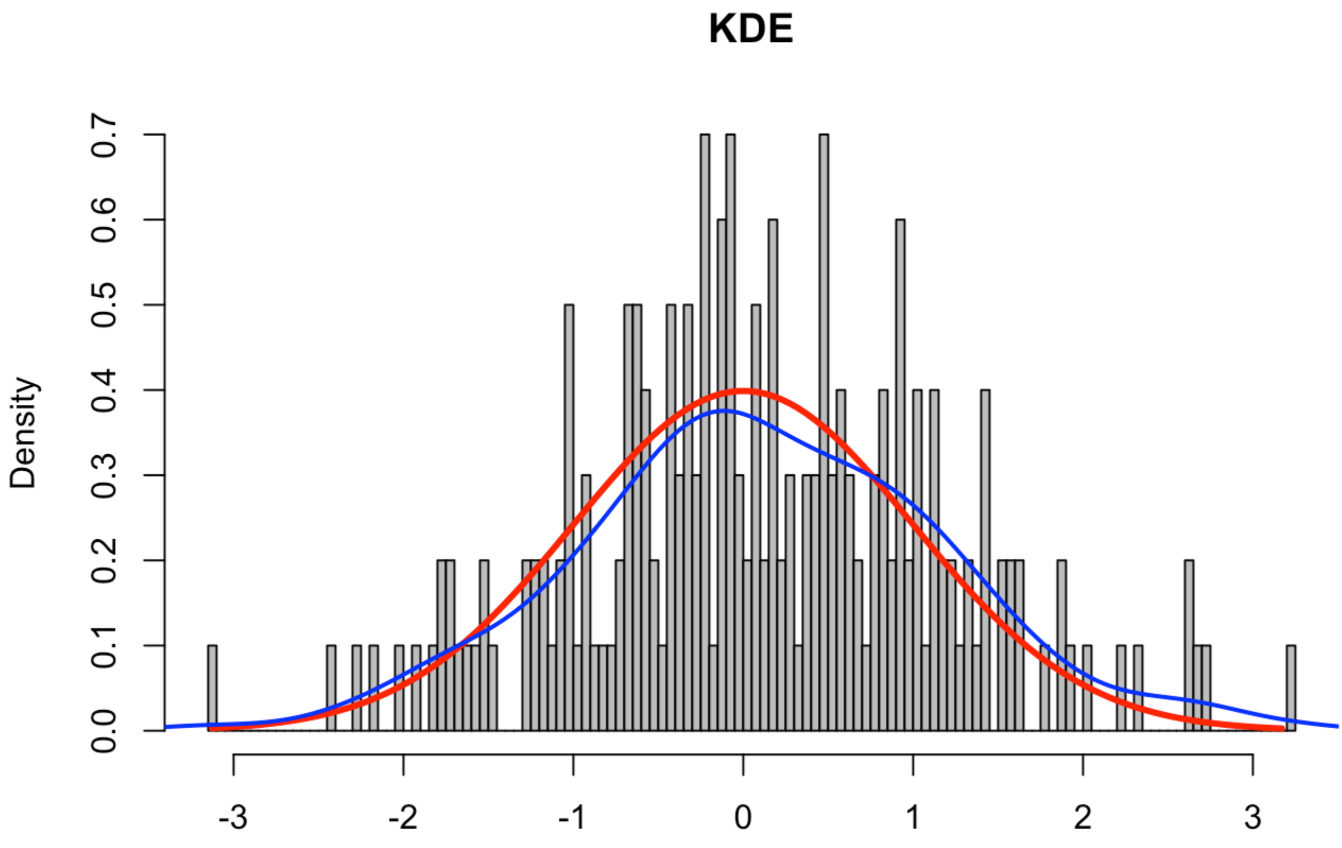


PARAMETRIC
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

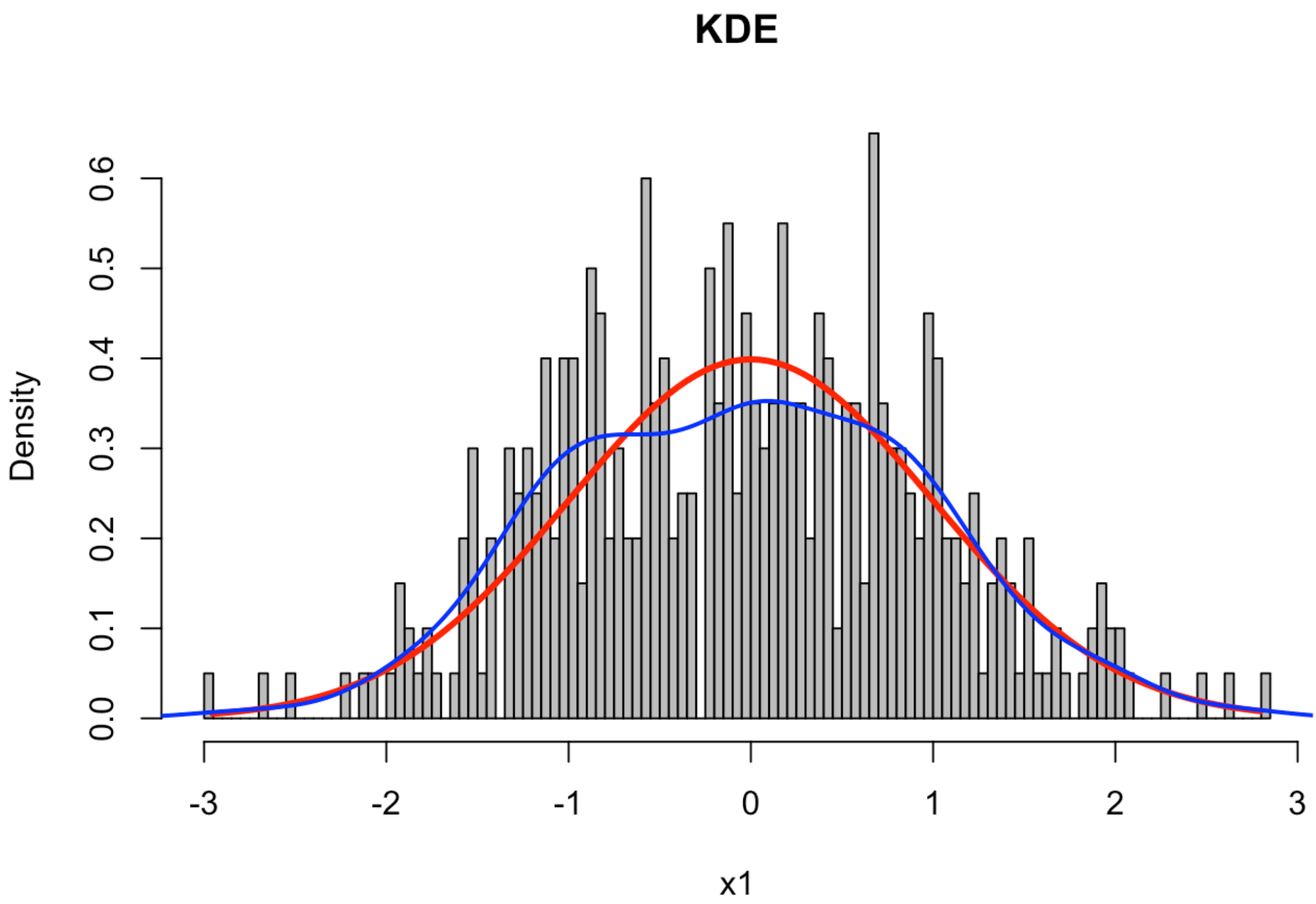


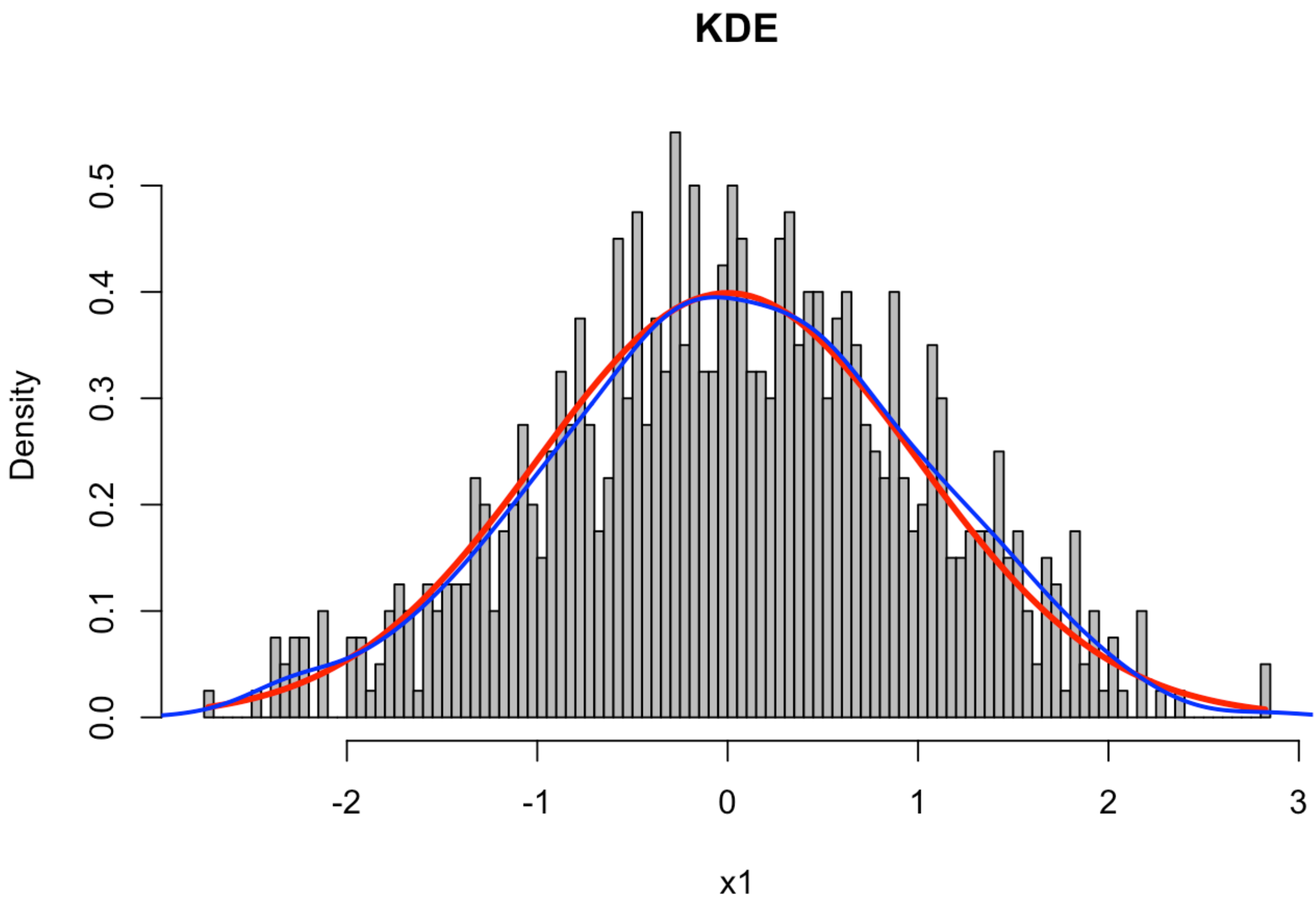

```
######### KDE##############

for(i in 1 : 16){
n<-50*2^i
x1<-rnorm(n, mu,sigma)
s<-seq(min(x1),max(x1),by=0.05)
par(mfrow=c(1,1))
hist(x1,probability = T,breaks = 100, col=8, main='KDE')
lines(dnorm(s, mean=0, sd=1)~s, col=2, lwd=3)
lines(density(x1), col=4, lwd=2)
Sys.sleep(2)
}
```

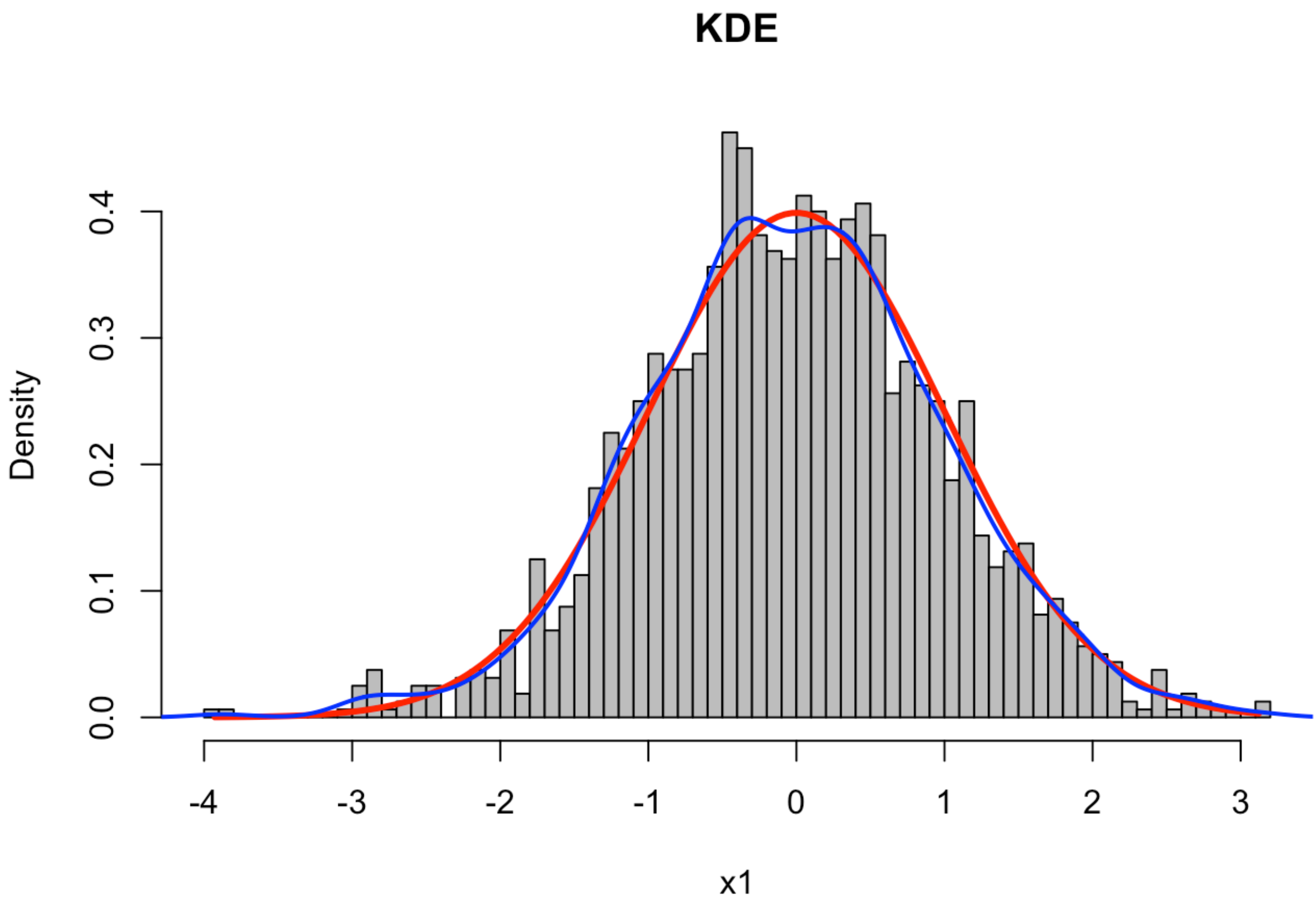

KDE
Density
1.0
0.8
0.6
0.4
0.2
0.0
-2
-1
0
1
2
x1

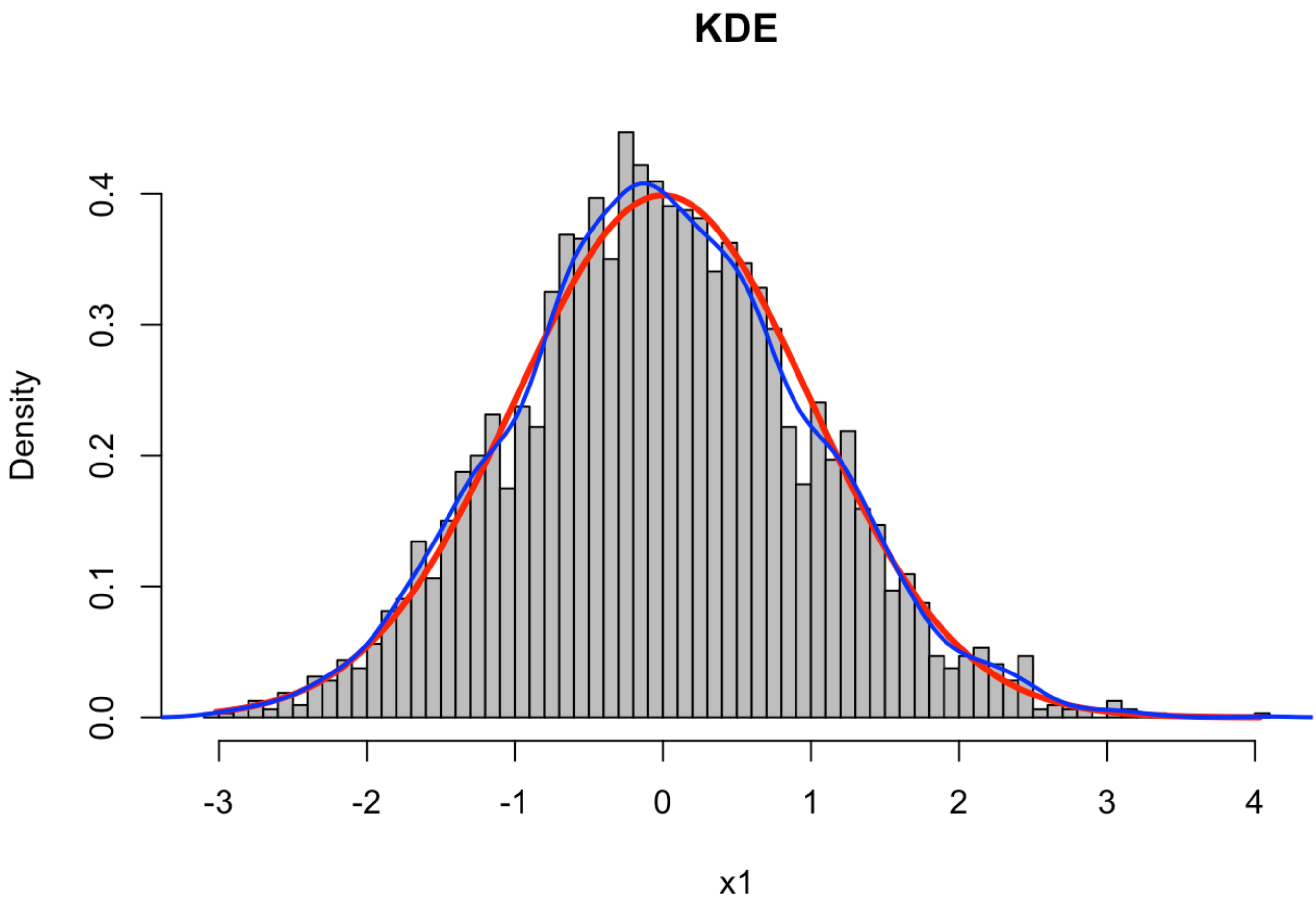

KDE
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-3
-2
-1
0
1
2
3

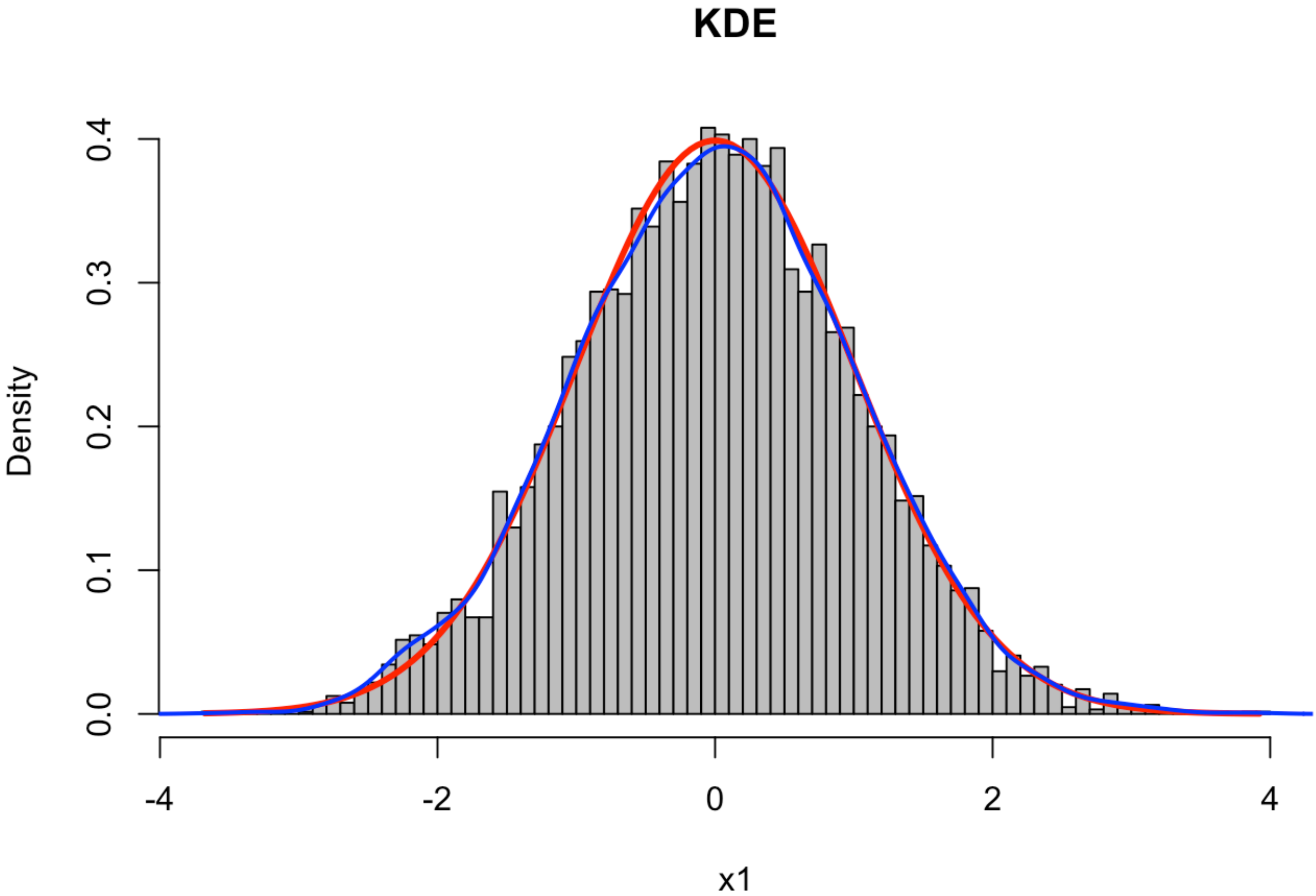
x1

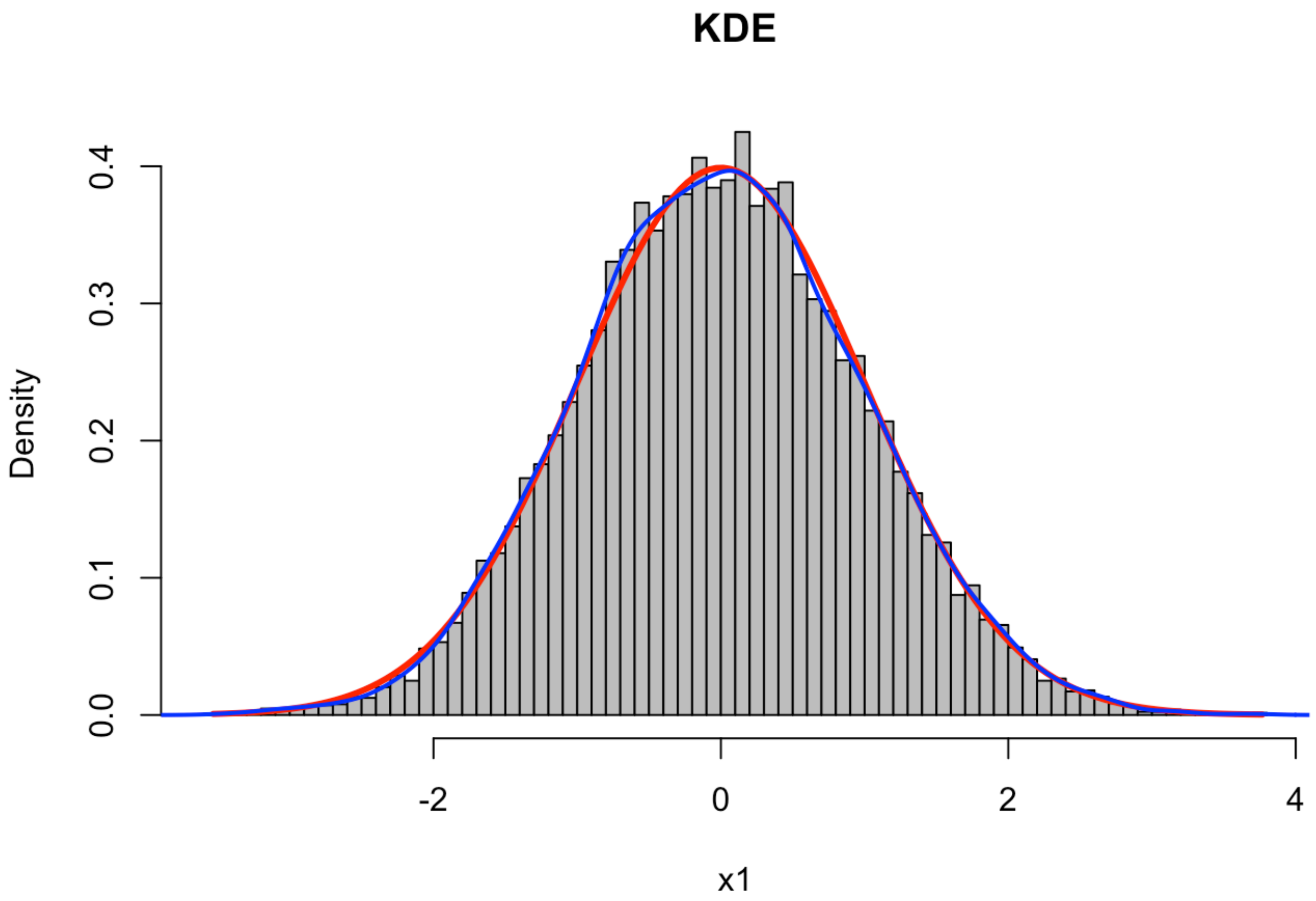


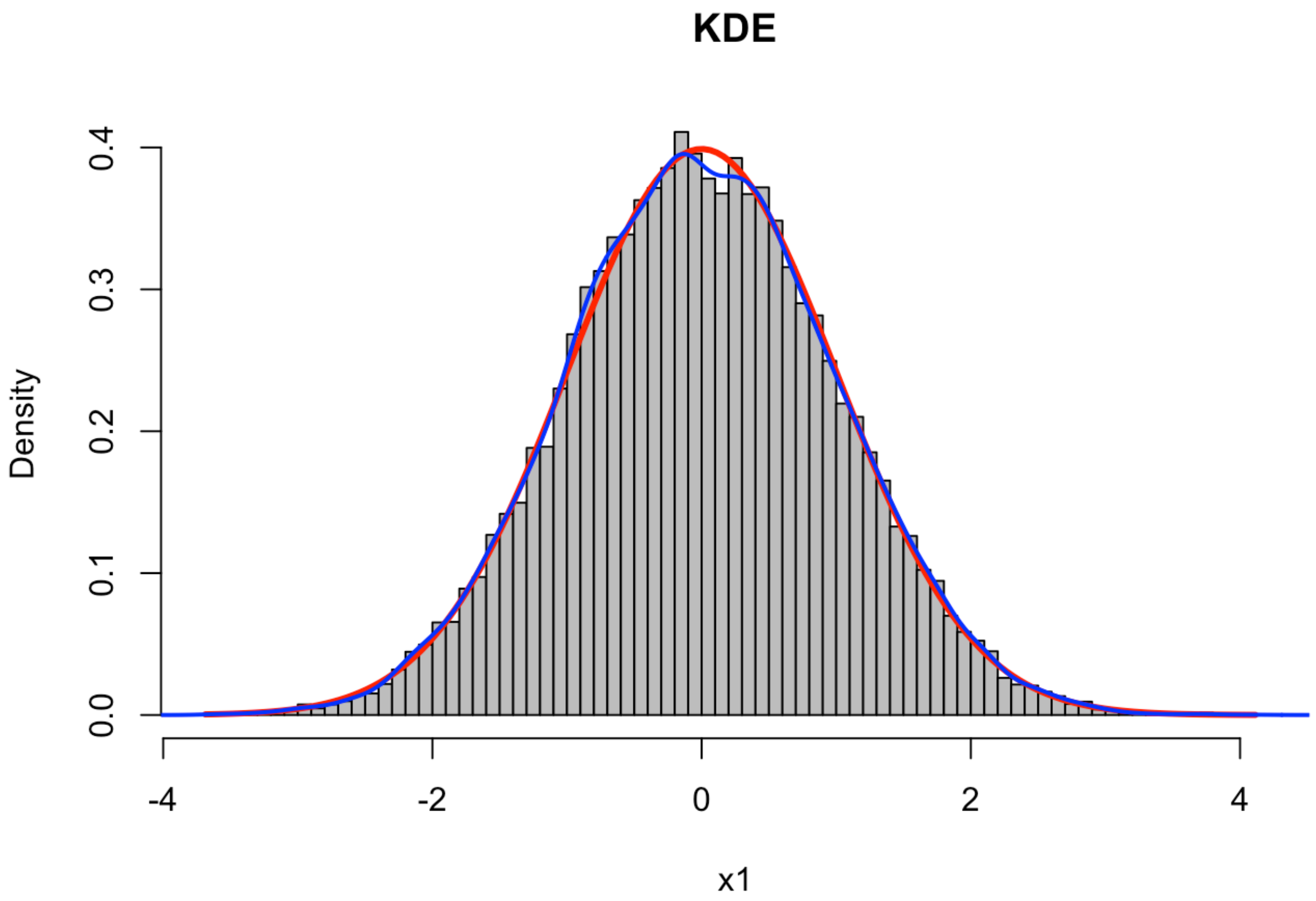

KDE
Density
0.0
0.1
0.2
0.3
0.4
0.5
-2
-1
0
1
2
3
x1

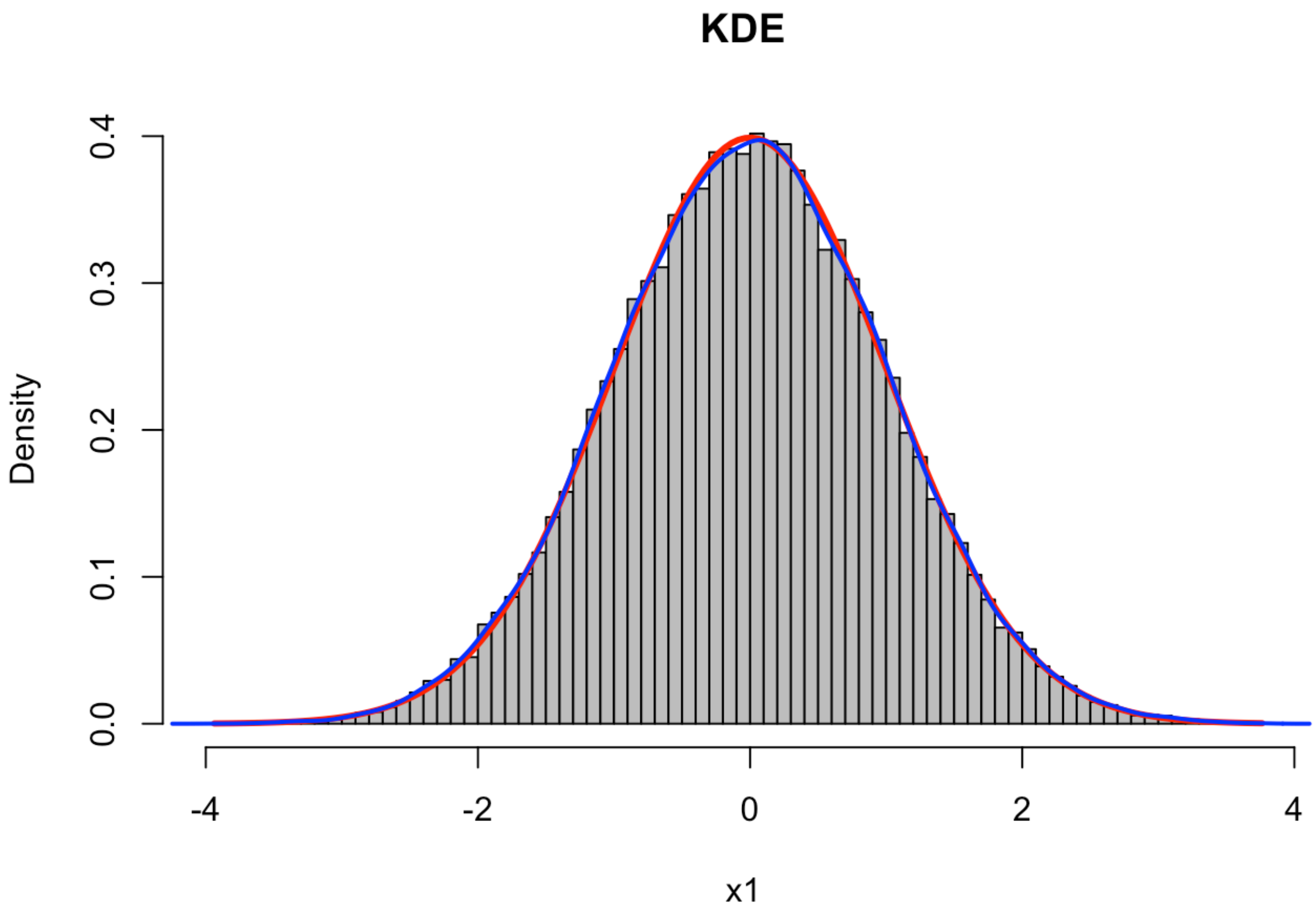

KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-3
-2
-1
0
1
2
3
x1

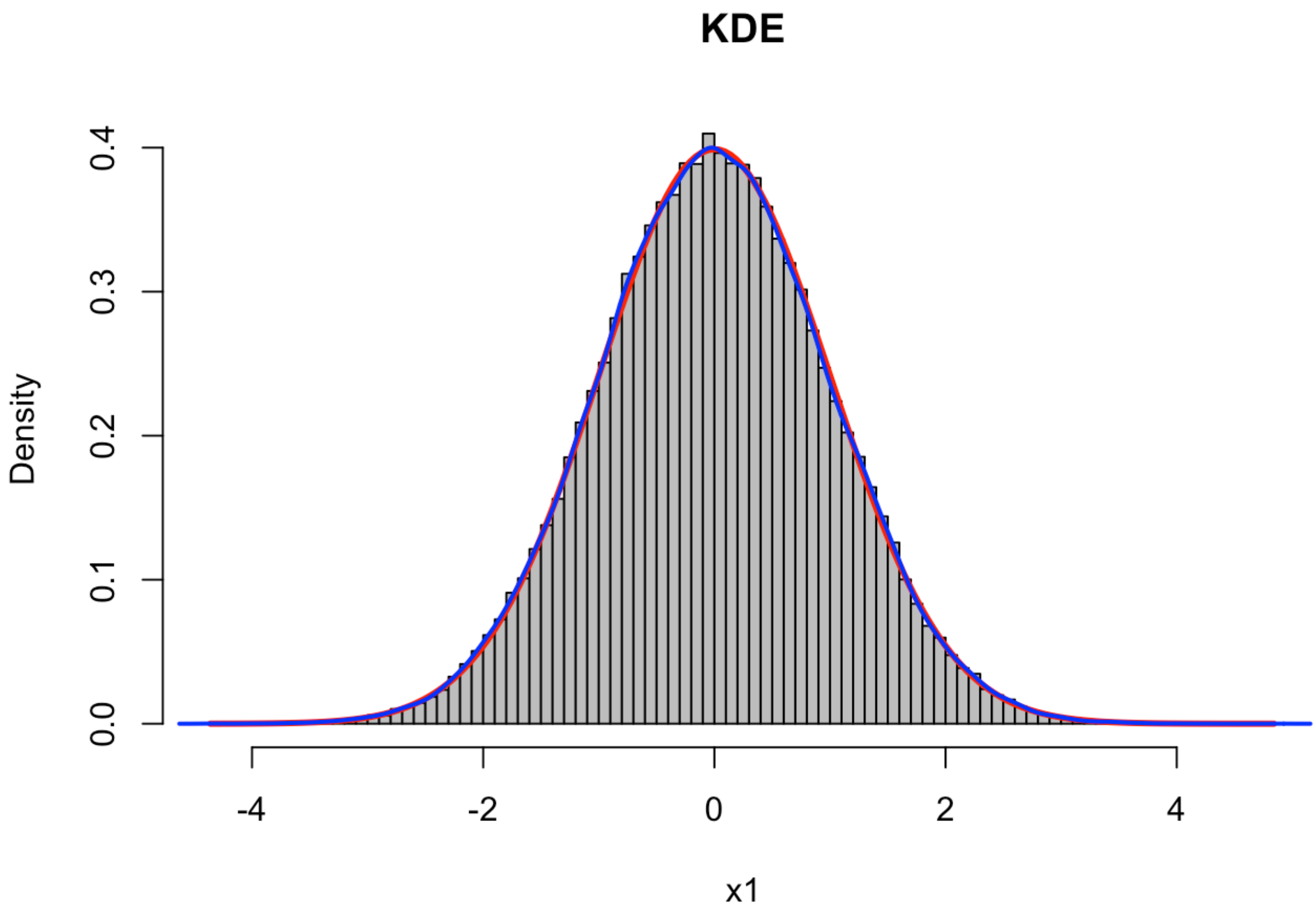

KDE
Density
0.0
0.1
0.2
0.3
0.4
-3
-2
-1
0
1
2
3
4
x1

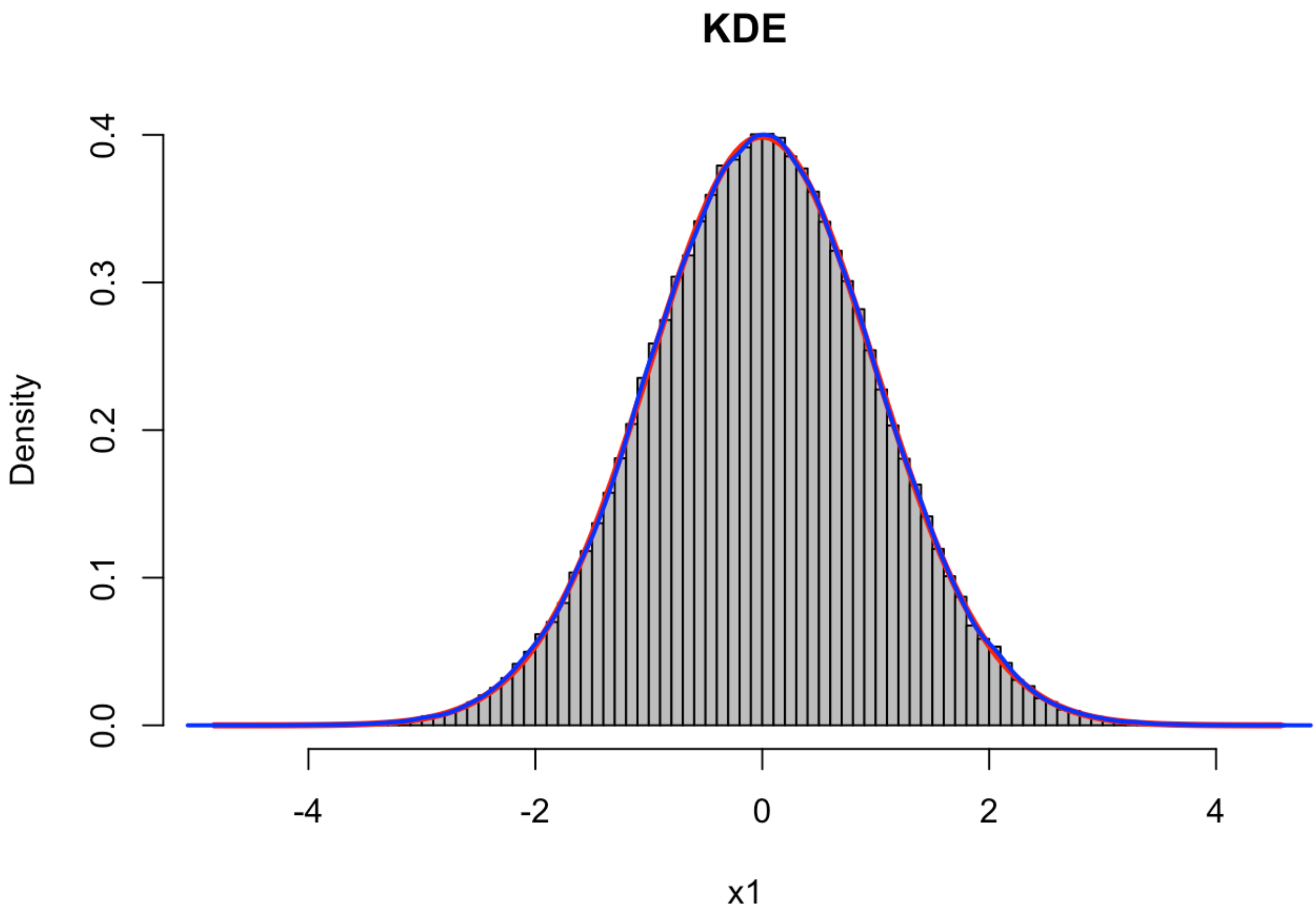

KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

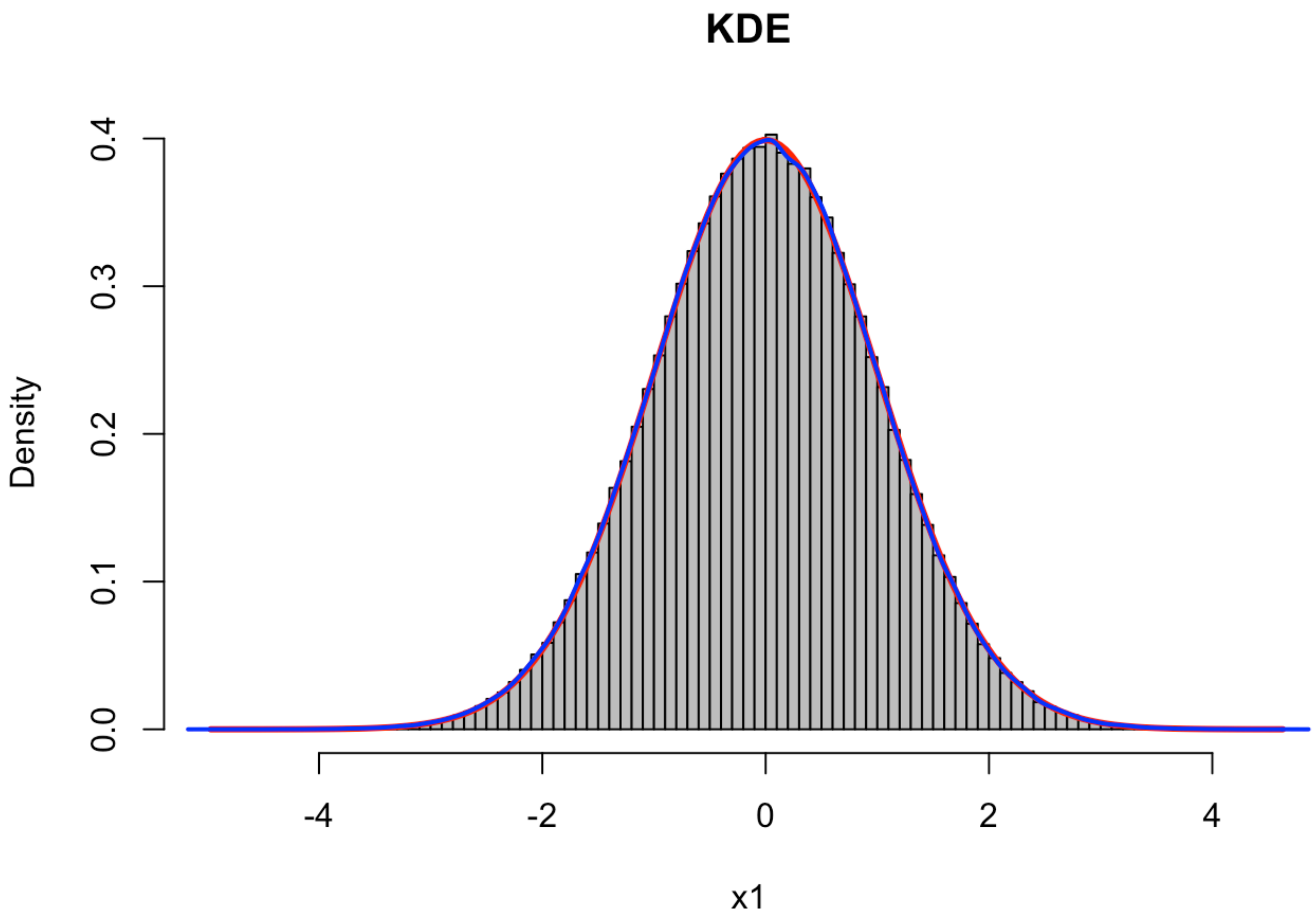

KDE
Density
0.0
0.1
0.2
0.3
0.4
-2
0
2
4
x1

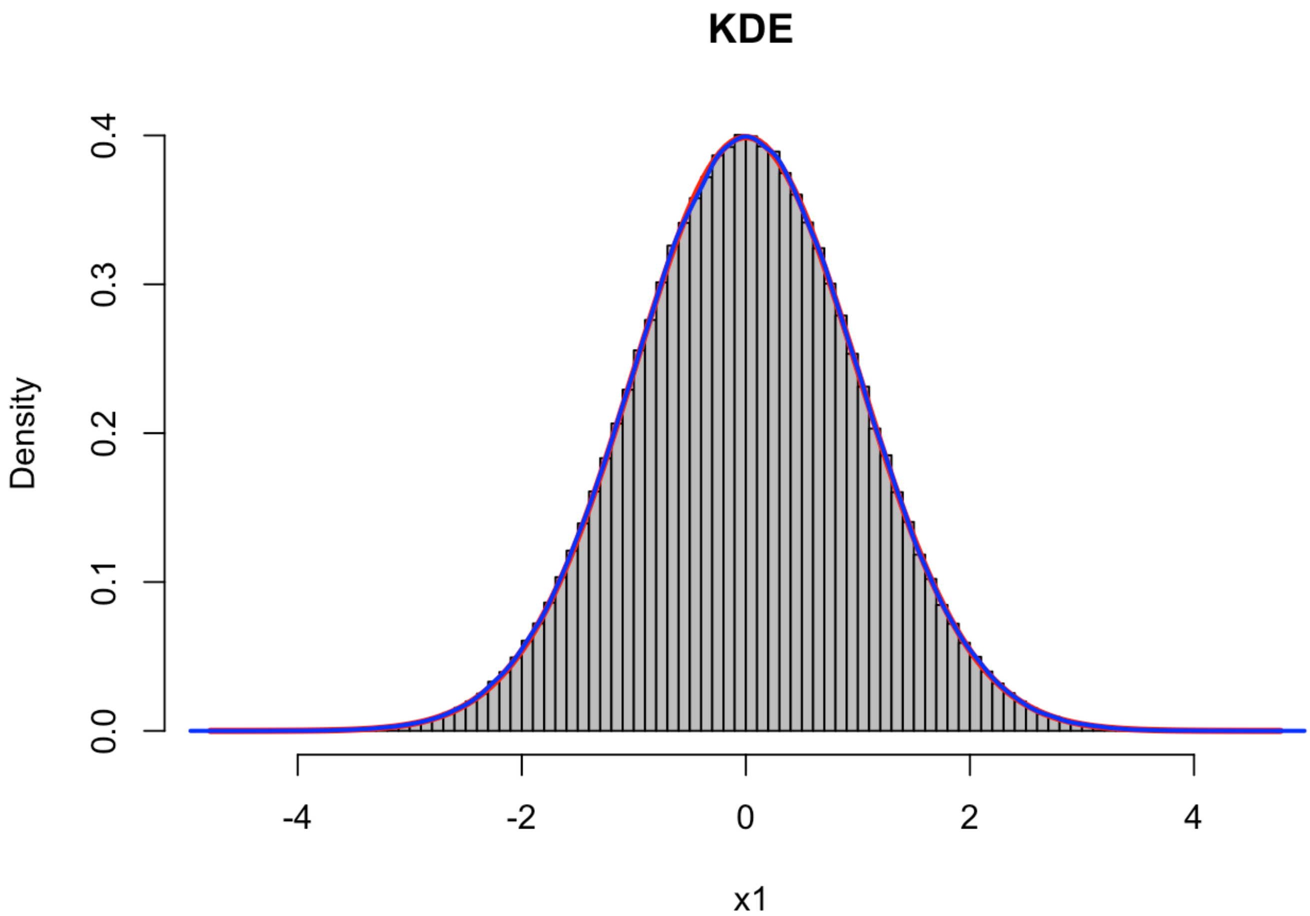

KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

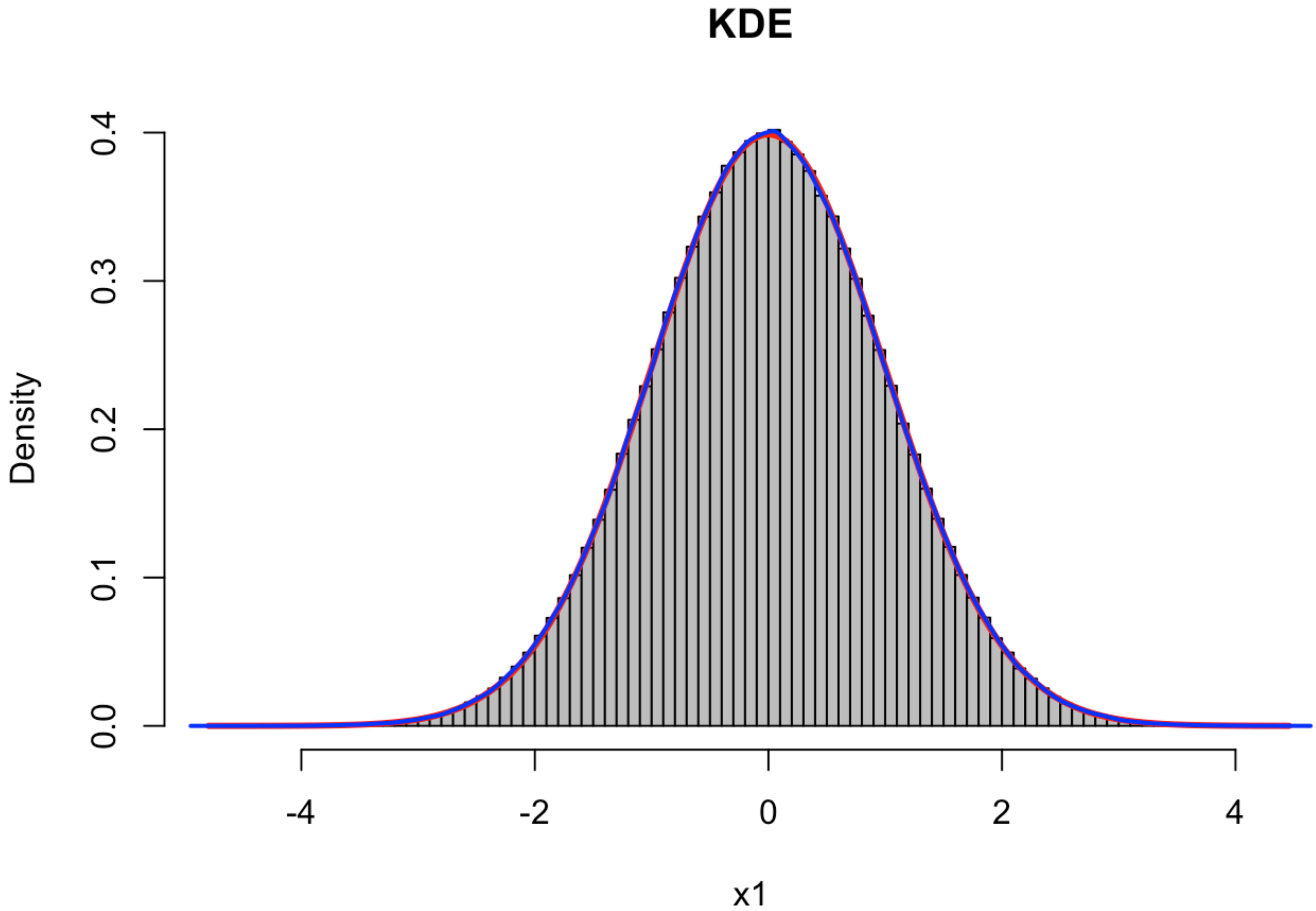

KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

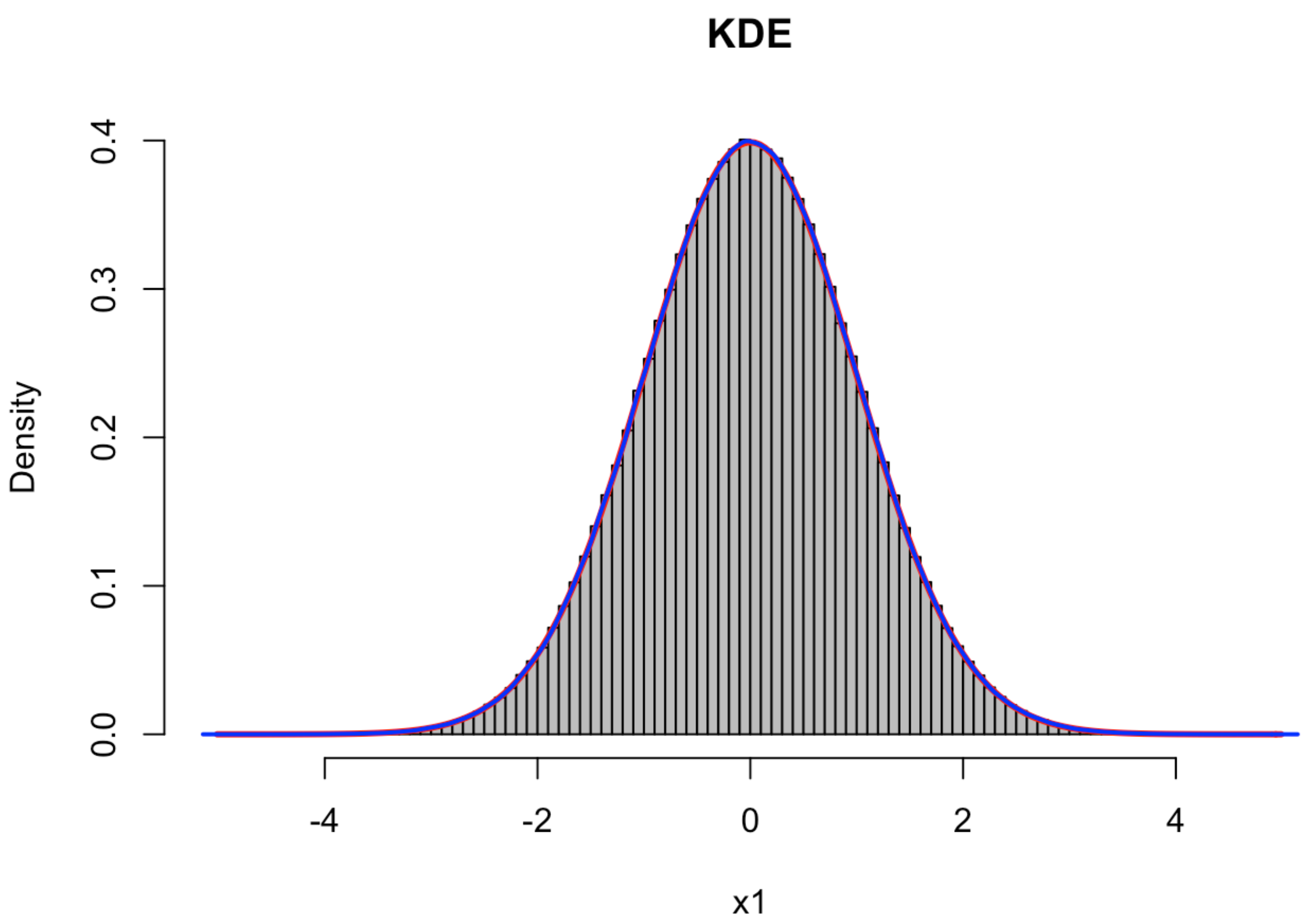

KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1


KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

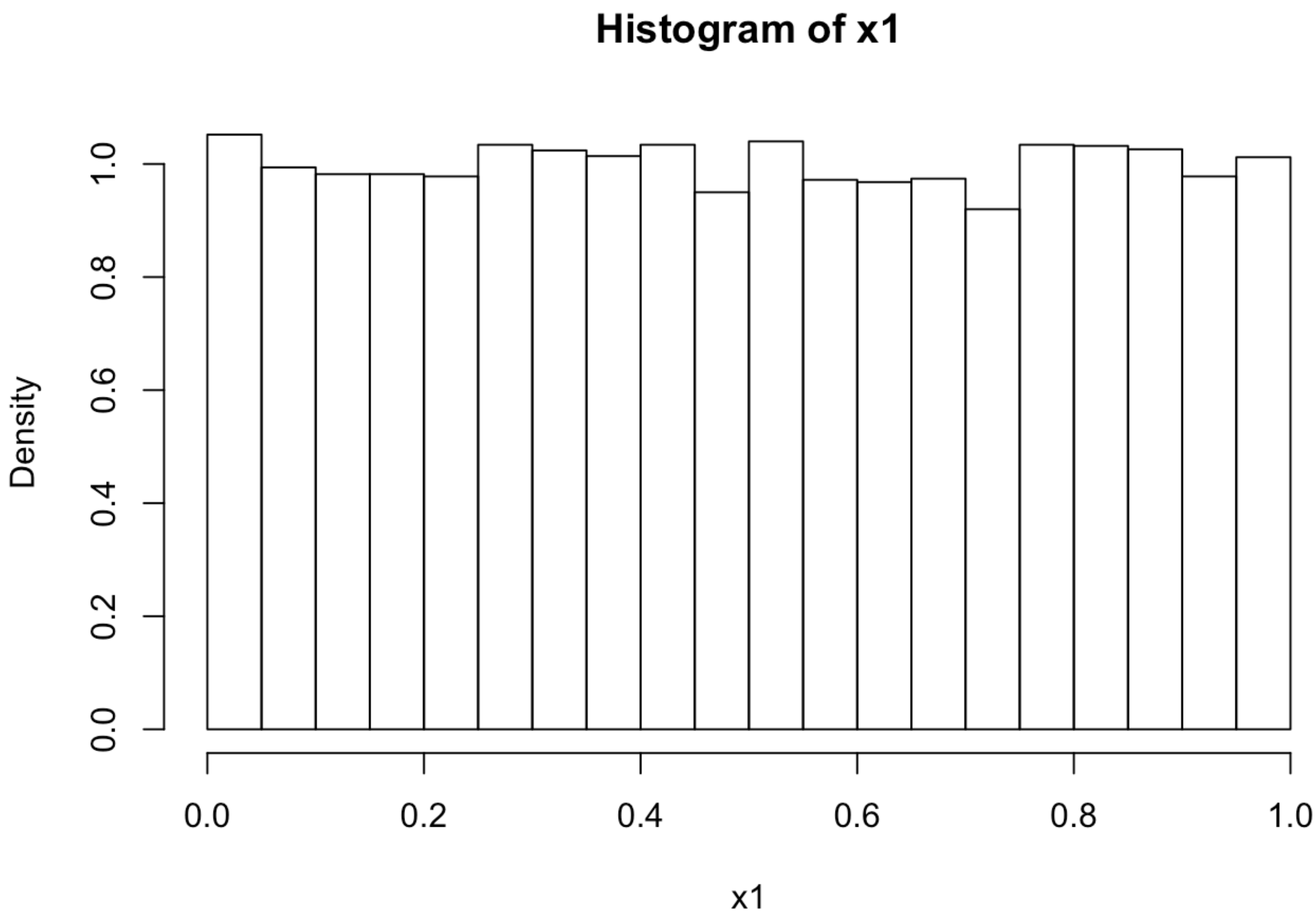

KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1


KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1

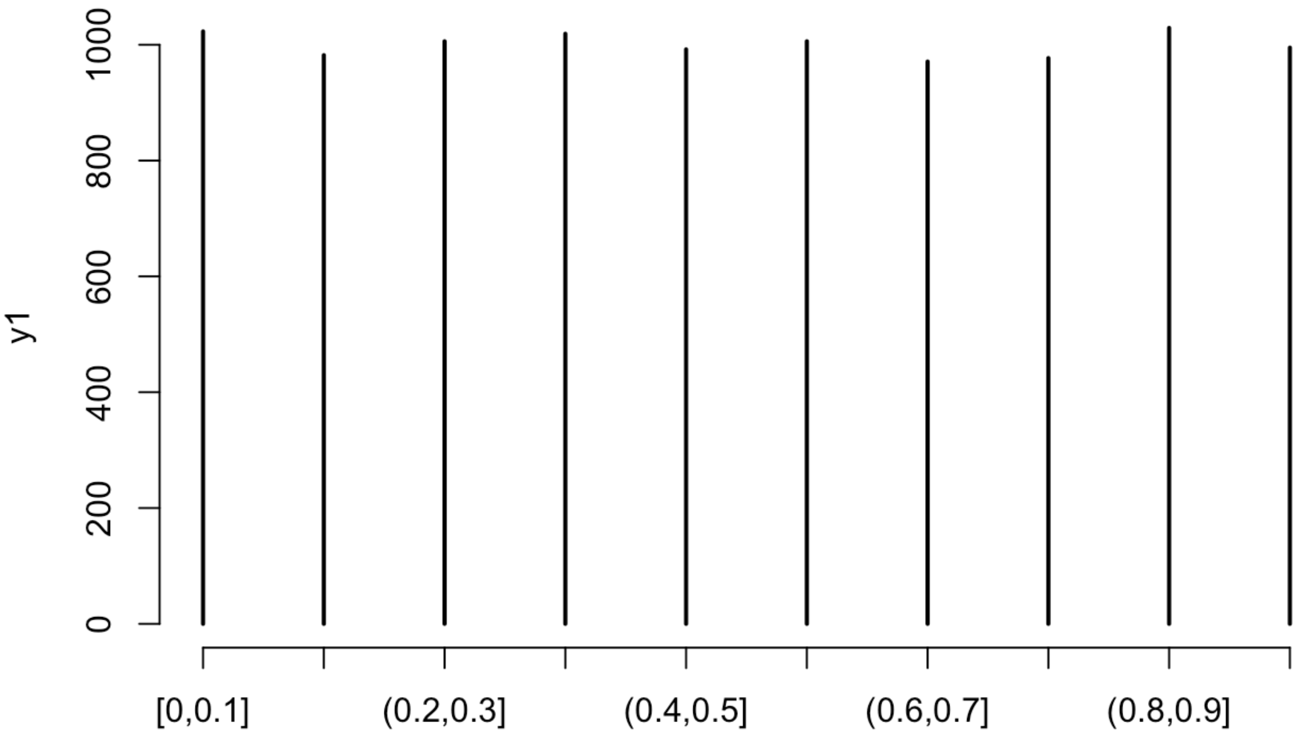

KDE
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
x1




```
########### ENTROPY #########

#install.packages("entropy")
library("entropy")


# sample from continuous uniform distribution
x1 = runif(10000)
hist(x1, xlim=c(0,1), freq=FALSE)
```




```
# 'discretize' puts observations from a continuous random variable into bins and r
eturns the corre-sponding vector of counts.
#discretize into 10 categories
y1 = discretize(x1, numBins=10, r=c(0,1))
plot(y1)
```




```
print(y1)
```

```
##
##   [0,0.1] (0.1,0.2] (0.2,0.3] (0.3,0.4] (0.4,0.5] (0.5,0.6] (0.6,0.7]
##      1023       982      1006      1019       992      1006       971
## (0.7,0.8] (0.8,0.9]   (0.9,1]
##       977      1029       995
```

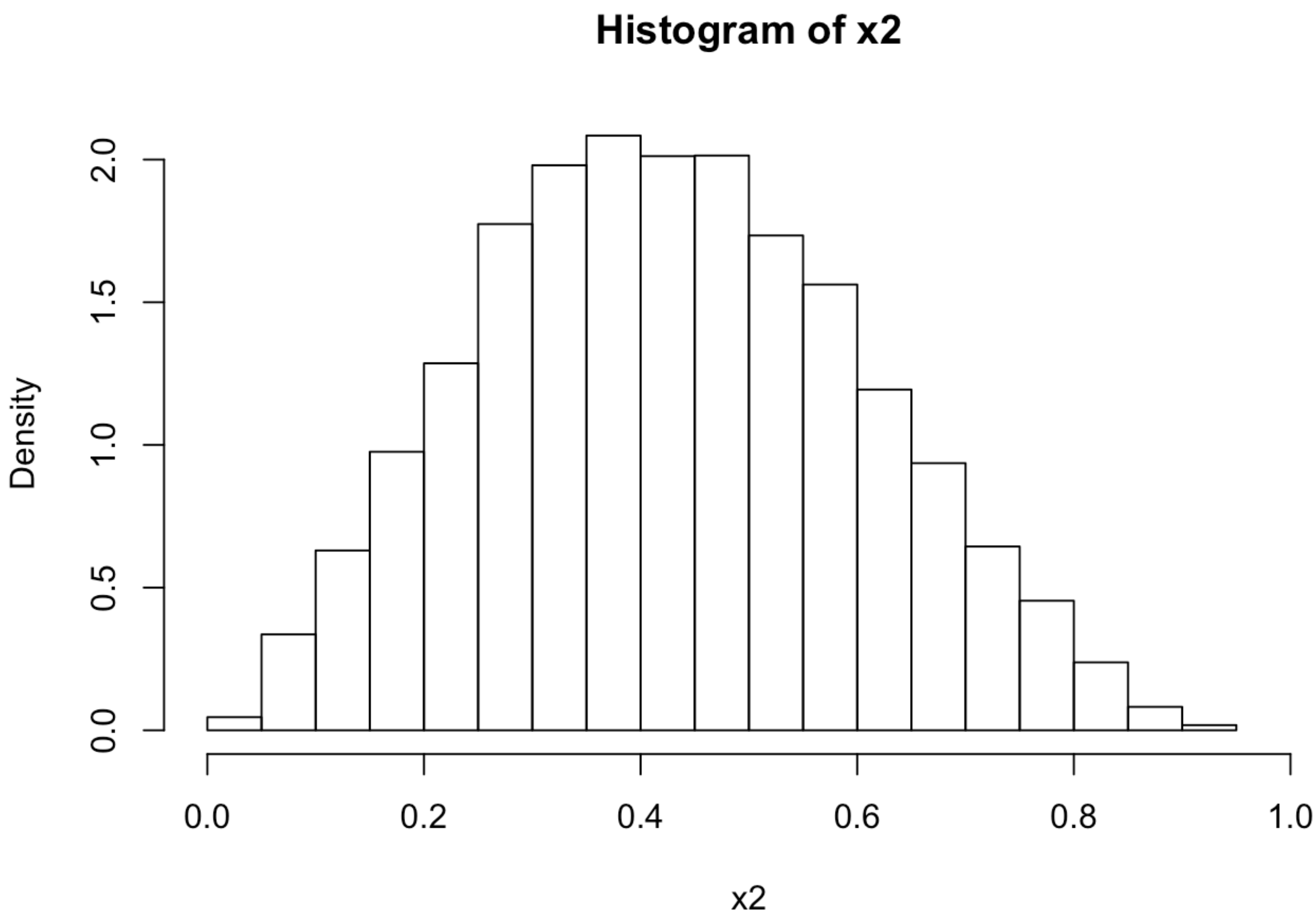
```
# compute entropy from counts
entropy(y1,unit = "log") # empirical estimate near theoretical maximum
```

```
## [1] 2.302406
```

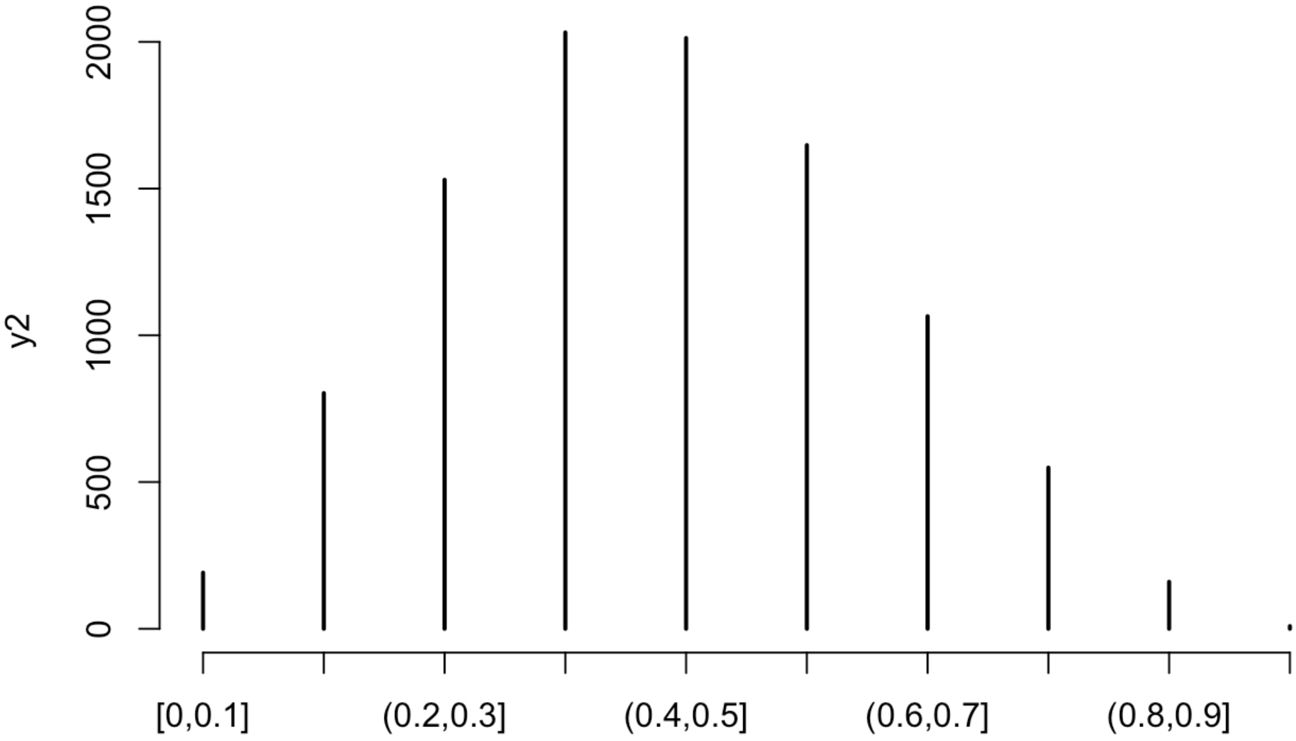
```
log(10) # theoretical value for discrete uniform distribution with 10 bins
```

```
## [1] 2.302585
```

```
# sample from a non-uniform distribution
x2 = rbeta(10000, 3, 4)
hist(x2, xlim=c(0,1), freq=FALSE)
```




```
# discretize into 10 categories and estimate entropy
y2 = discretize(x2, numBins=10, r=c(0,1))
plot(y2)
```




```
print(y2)
```

```
##
##   [0,0.1] (0.1,0.2] (0.2,0.3] (0.3,0.4] (0.4,0.5] (0.5,0.6] (0.6,0.7]
##       191       803      1530      2032      2013      1648      1065
## (0.7,0.8] (0.8,0.9]   (0.9,1]
##       549       160         9
```

```
# estimated entropy from data
est_dbeta<-entropy(y2)
s<-seq(0,1, by=0.1)
prob_beta<-pbeta(q=s,shape1 = 3,shape2 = 4)
dprob<-prob_beta[-1]-prob_beta[-11]
# theoritial enropy after discretization
edbeta<-sum(dprob*(-log(dprob)))

cat("Estimated entropy=", est_dbeta ,'\n')
```

```
## Estimated entropy= 1.979296
```

```
cat("Computed entropy=", edbeta ,'\n')
```

```
## Computed entropy= 1.975482
```

```
y2d = discretize2d(x1, x2, numBins1=10, numBins2=10)
# joint entropy
H12 = entropy(y2d )
H12
```

```
## [1] 4.348783
```

```
log(100) # theoretical maximum for 10x10 table
```

```
## [1] 4.60517
```

```
# mutual information
mi.empirical(y2d) # approximately zero
```

```
## [1] 0.004613659
```

```
# another way to compute mutual information
# compute marginal entropies
H1 = entropy(rowSums(y2d))
H2 = entropy(colSums(y2d))
H1+H2-H12 # mutual entropy
```

```
## [1] 0.004613659
```

```
# KL divergence

KL.empirical(y1,y2)
```

```
## [1] 0.6661323
```