Rejection method for random number generation

Step by Step Explanation using R

Simulate Standard Normal Distribution (1/4)

```
fx <- function(x){
  return((2/pi)^0.5*exp(-(x^2)/2))
}
```

- Build the truncated normal function (divide Normal Distribution density function by 0.5)

```
b <- function(){
  k <- rbinom(1,1,0.5)
  if(k==0) return(-1)
  else return(1)
}
```

- Build function randomly generating 1 & -1

(relocate half of the accepted data to negative side)

Simulate Standard Normal Distribution (2/4)

```
set.seed(10)
n <- 10000
g <- rep(0,n)
count <- 0
for(i in 1:n){
  x <- rexp(1)
  y <- runif(1,0,(2*exp(1)/pi)^0.5*exp(-x))
  if (y<=fx(x)){
        g[i]<- x*b()
        count <- count+1
  }else{
        g[i] <- NA
  }
}
```

- Set simulation times
- Build empty vetor
- Build variable record the acceptance times
- Generate random variable in positive real number field (from 0 to infinity)
- Generate the envelope
- If the value generate from the envelope smaller than the value generated from truncated normal distribution, then store into vector g and add one to count. Otherwise, store NA into vector g.

Simulate Standard Normal Distribution (3/4)

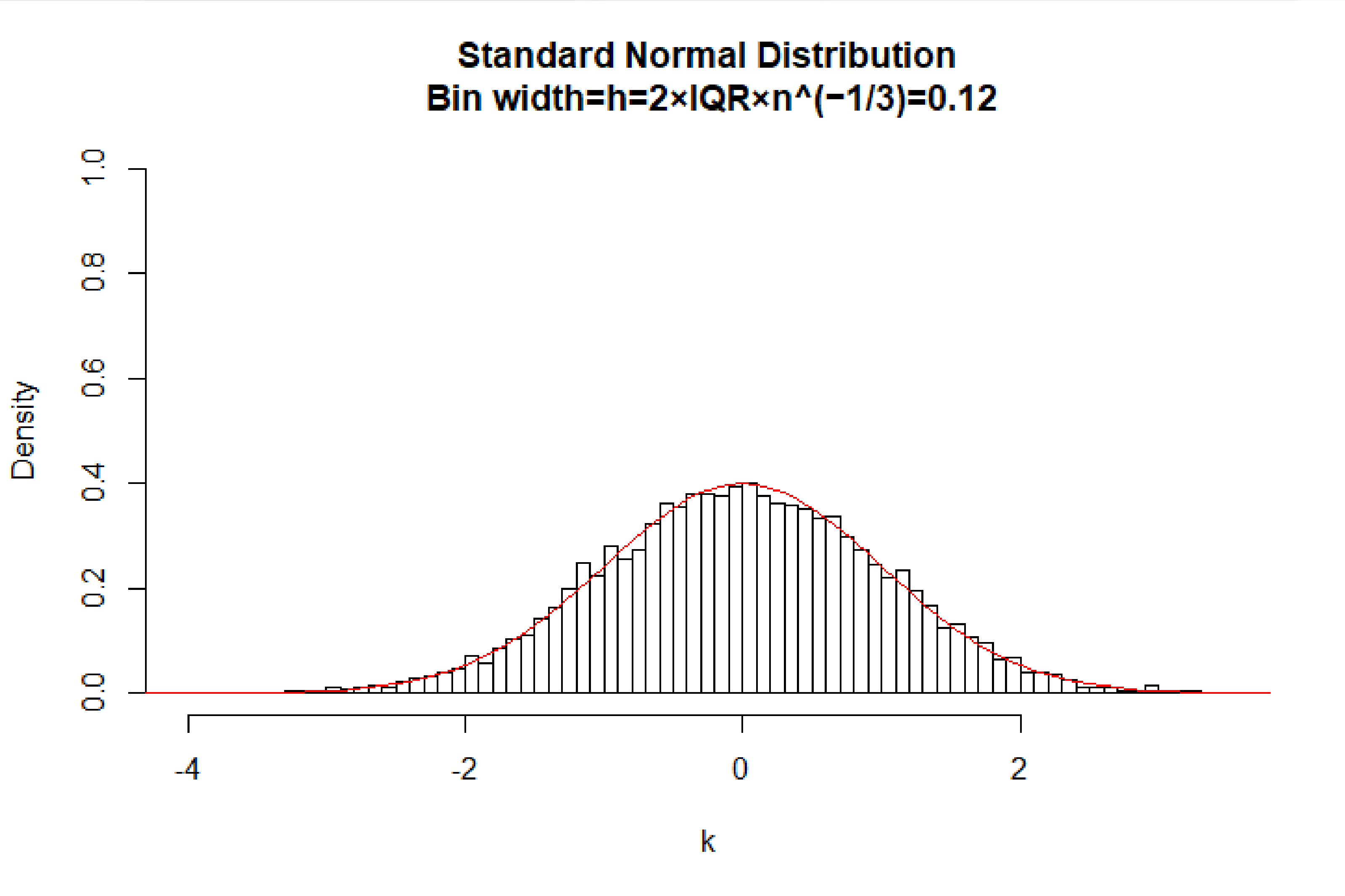
```
k<-g[!is.na(g)]
```

- Rule out the NA in the matrix

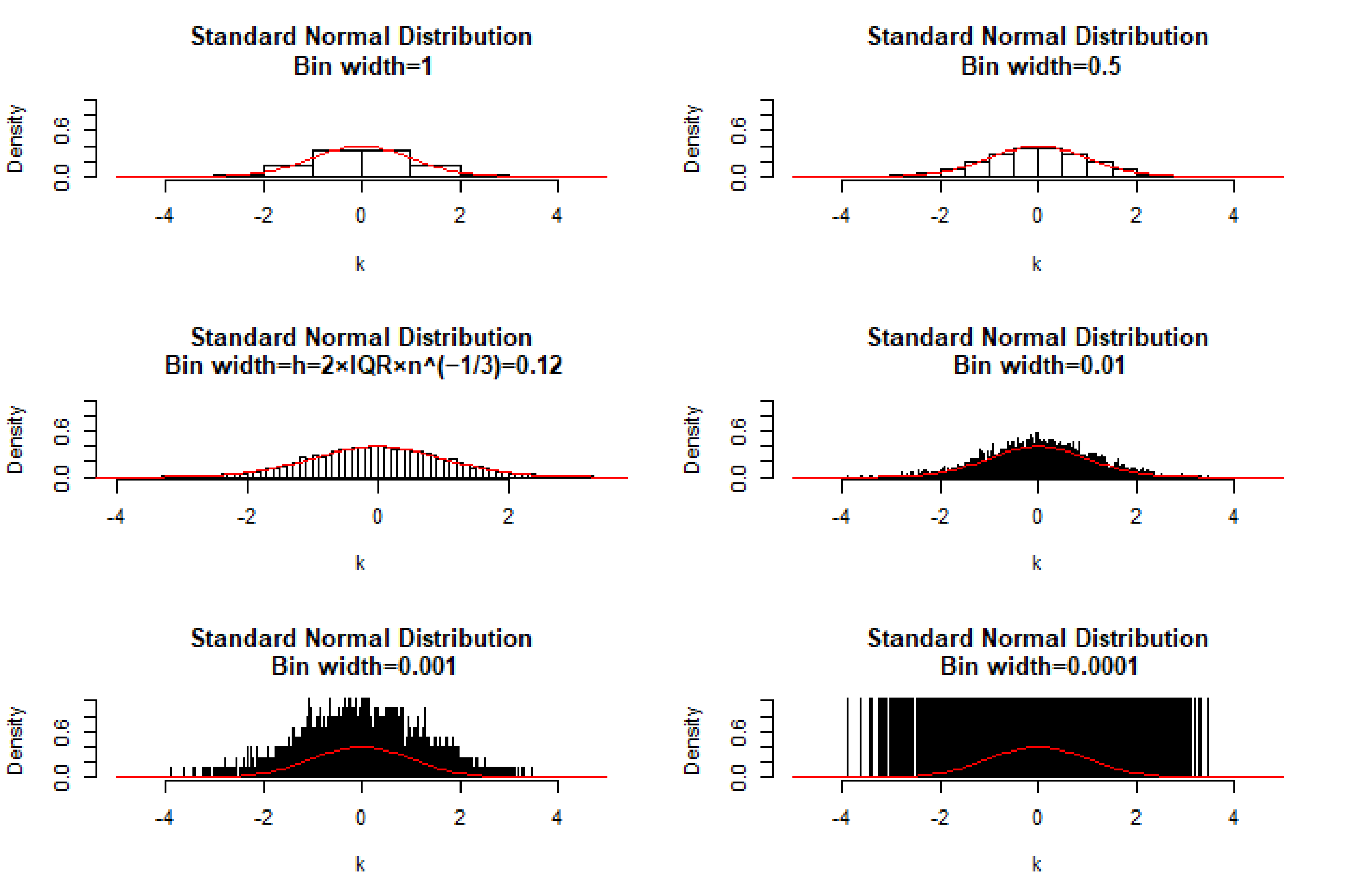
```
x=seq(-5,5,by=0.1)
hist(k,breaks="FD",freq=F,ylim=c(0,1),main="Standard Normal Distribution")
lines(x,dnorm(x),col="yellow")
ap_rate= count/n
```

- Determine the bin width with Freedman-Diaconis rule
- Plot real standard normal density
- Count acceptance rate

Simulate Standard Normal Distribution (4/4)



*Discussion: How to determine bin width?



*Discussion: How to determine bin width?

1. Sturges' formula → $k = \lceil \log_2 n \rceil + 1$

1. Doane's Rule → $\boxed{\log_2(n) + 1 + \log_2\left(1 + \frac{\sqrt{b}}{\sigma\sqrt{b}}\right)}$

Where $\sqrt{b} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^3}{\left[\sum_{i=1}^{n}(X_i - \bar{X})^2\right]^{(3/2)}}$

and $\sigma\sqrt{b} = \sqrt{\frac{6(n-2)}{(n+1)(n+3)}}$.

1. Scott's Rule → $h = \frac{3.5\hat{\sigma}}{\sqrt[3]{n}}$

1. Rice's Rule → $k = \lceil 2\sqrt[3]{n} \rceil$

1. Freedman-Diaconis rule → $\boxed{h = 2 \times \mathrm{IQR} \times n^{-1/3}}$

Application: Simulate Beta Distribution (1/3)

```
set.seed(10)
n <- 10000
g <- rep(0,n)
count <- 0
```

- Set simulation times
- Build empty vetor
- Build variable record the acceptance times

```
for(i in 1:n){
  y1<-runif(1,0,1)
  y2<-runif(1,0,1)
  if (y2<=4*(y1*(1-y1))){
    g[i]<-y1
    count <- count+1
  }else{
    g[i] <- NA
  }
}
```

- Generate random variable from uniform[0,1]
- Beta distribution with n=m
- If the value generate randomly from the niform[0,1] smaller than the value generated from Beta distribution, then store into vector g and add one to count. Otherwise, store NA into vector g.

Simulate Beta Distribution (2/3)

```
k<-g[!is.na(g)]
```

- Rule out the NA in the matrix

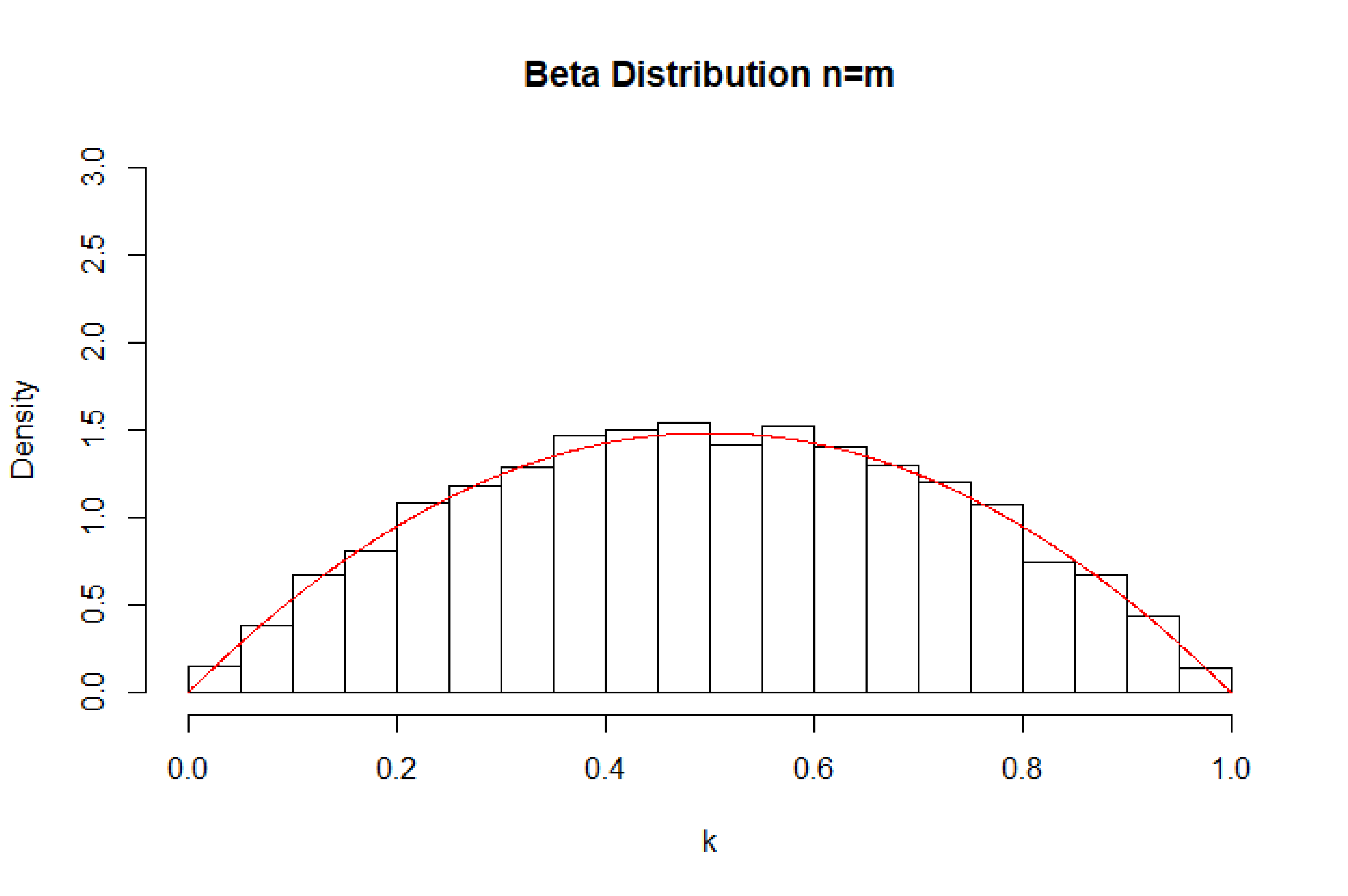
```
x=seq(0,1,by=0.001)
hist(k,breaks="FD",freq=F,ylim=
c(0,3),main="beta")
```

- Determine the bin width with Freedman-Diaconis rule

```
lines(x,5.93*x*(1-x),col="red")
ap_rate= count/n
```

- Plot real Beta distribution (n=m)
- Count acceptance rate

Simulate Beta Distribution (3/3)



THANKS!