HW 4 (solution)

1) Finite Mixture Model

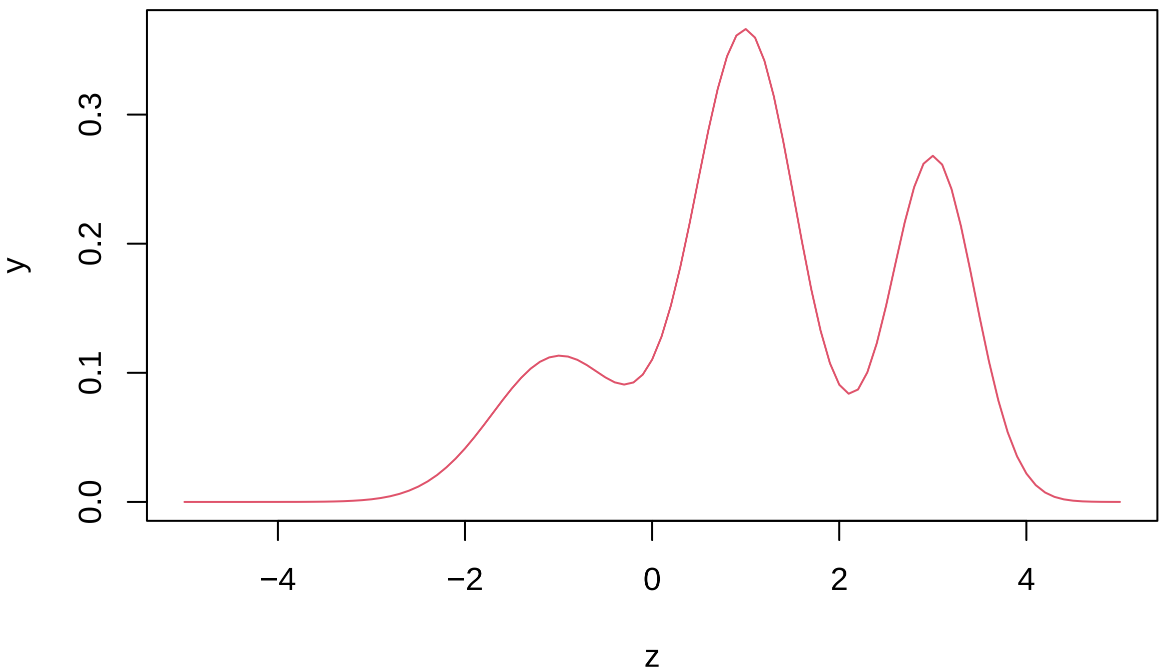
In a finite Mixture model, the marginal distribution of a random variable (RV, say x) is a weighted sum of a finite number of densities. The following is an example of a mixture of three normal densities

$$f(x) = \pi_1 N(x|\mu_1, \sigma_1^2) + \pi_2 N(x|\mu_2, \sigma_2^2) + (1 - \pi_1 - \pi_2) N(x|\mu_3, \sigma_3^2) \tag{1}$$

where $N(x|\mu_*, \sigma_*^2) = \frac{e^{-\frac{(x-\mu_*)^2}{2\sigma_*^2}}}{\sqrt{2\pi\sigma_*^2}}$

The following is a density plot from the above mixture with:

- $\mu_1 = -1, \sigma_1^2 = 0.5$,
- $\mu_2 = 1, \sigma_1^2 = 0.3$,
- $\mu_3 = 3, \sigma_3^2 = 0.2$, and
- $\pi_1 = .2$, $\pi_2 = .5$.



You can see how, by mixing parametric symmetric distributions such as the normals, we can get density functions that are very flexible.

A mixture model can also be represented as the joint distribution of two RVs $p(x, z) = p(x|z)p(z)$ where z is an indicator variable for the mixture component ($z \in \{1, 2, 3\}$, $p(z = 1) = \pi_1$, $p(z = 2) = \pi_2$, and $p(z = 3) = 1 - \pi_1 - \pi_2$) and $p(x|z) = N(x|\mu_z, \sigma_z^2)$.

The marginal distribution of x is

$p(x) = p(z = 1)N(x|\mu_1, \sigma_1^2) + p(z = 2)N(x|\mu_2, \sigma_2^2) + p(z = 3)N(x|\mu_3, \sigma_3^2$

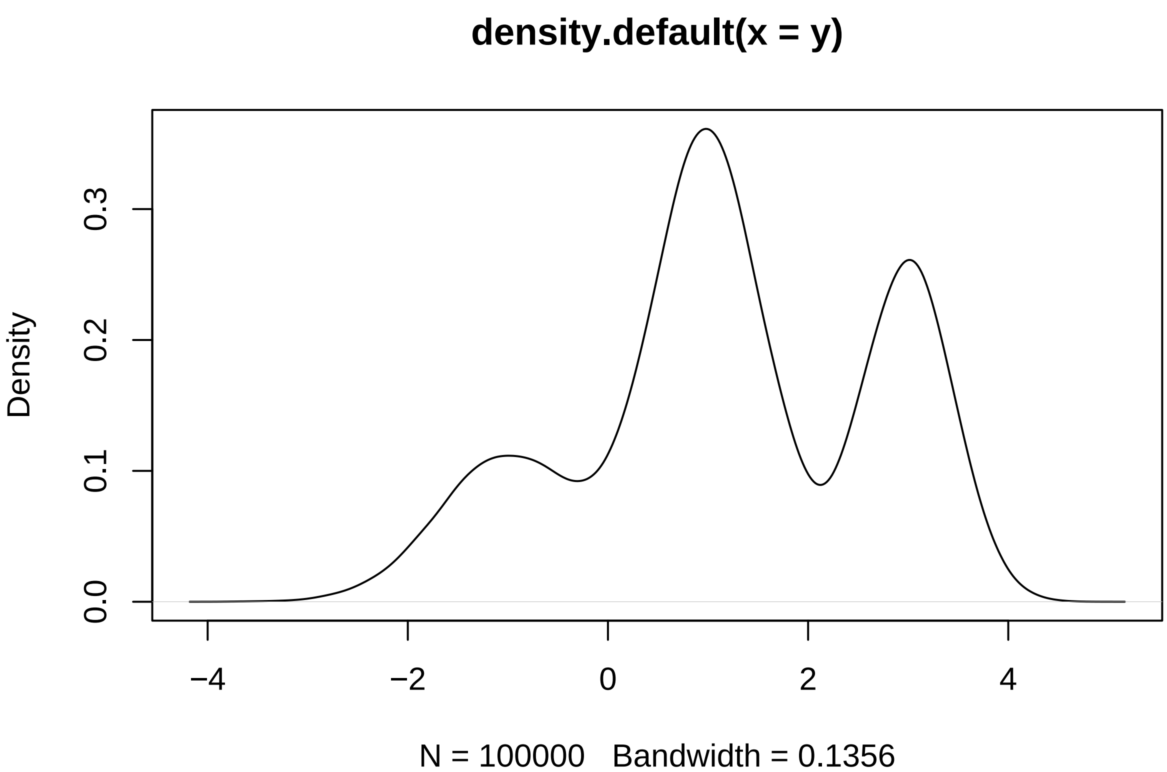
which is the same as expression (1).

Task:

- Use composition sampling to generate 100,000 samples from $p(z, x) = p(x|z)p(z)$.
- Plot the empirical density of x using `plot(density(x))` where x are the samples you generated.
- Hint: z is a multinational RV. To sample it you can use:

```
pi_1=0.2
pi_2=0.5
n=100 # set this to the desired number of samples
u=runif(n)
z=ifelse(u<pi_1,1,ifelse(u<(pi_1+pi_2),2,3))
```

```
n=100000
u=runif(n)
z=ifelse(u<pi[1],1,ifelse(u<sum(pi[1:2]),2,3))
y=rnorm(n)*sqrt(v[z])+mu[z]
plot(density(y))
```



2) The Metropolis Algorithm

Let

- $p(x)$ be the density of x (or a function proporitional to it), and
- $q(x_t|x_{t-1})$ be a symmetric distribution satisfying $q(x_t|x_{t-1}) = q(x_{t-1}|x_t)$

In Metropolis, to generate samples from $p(x)$ we:

- Initialize x_0 with some value with non-zero probability and then,
- In a loop for t in 1:n we:
  - Generate a candidate $w \ \sim q(x_t|x_{t-1})$

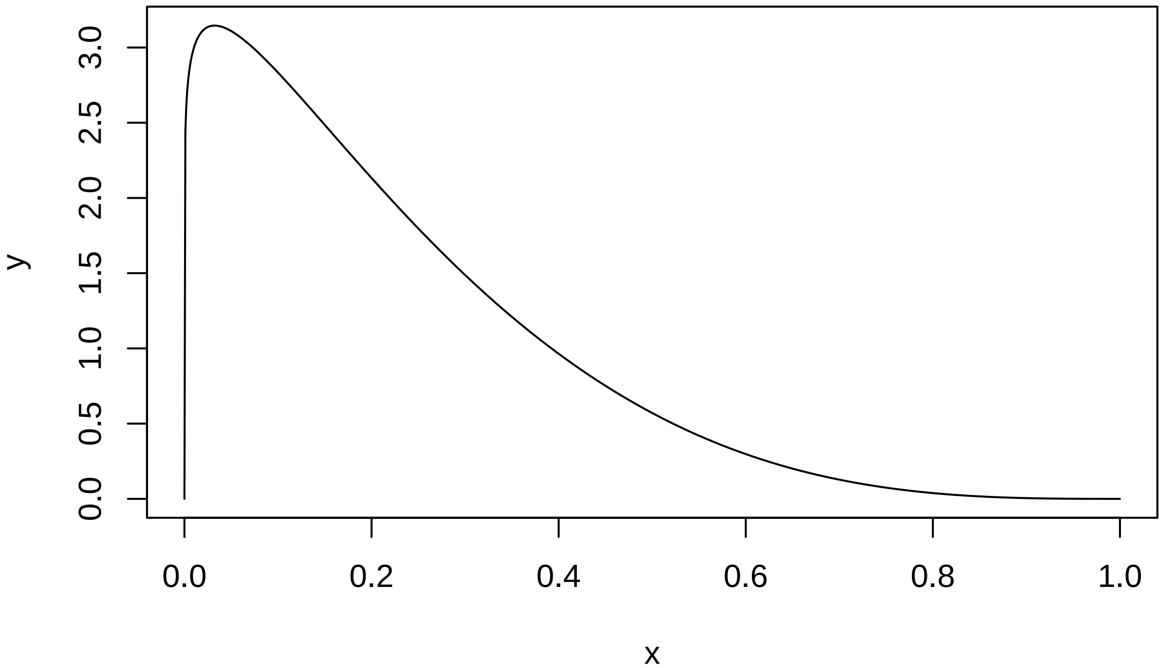
  - Compute the acceptance ratio $\alpha = min\{1, f(w)/f(x_{t-1})\}$
  - With probability α set $x_t = w$ and with probability $1-\alpha$ set $x_t = x_{t-1}$.

Example:

- Target density beta with shape1=1.1, and shape2=4 `dbeta(shape1, shape2)`
- $q(x_t|x_{t-1}) = Uniform(0,1)$

The Beta Density

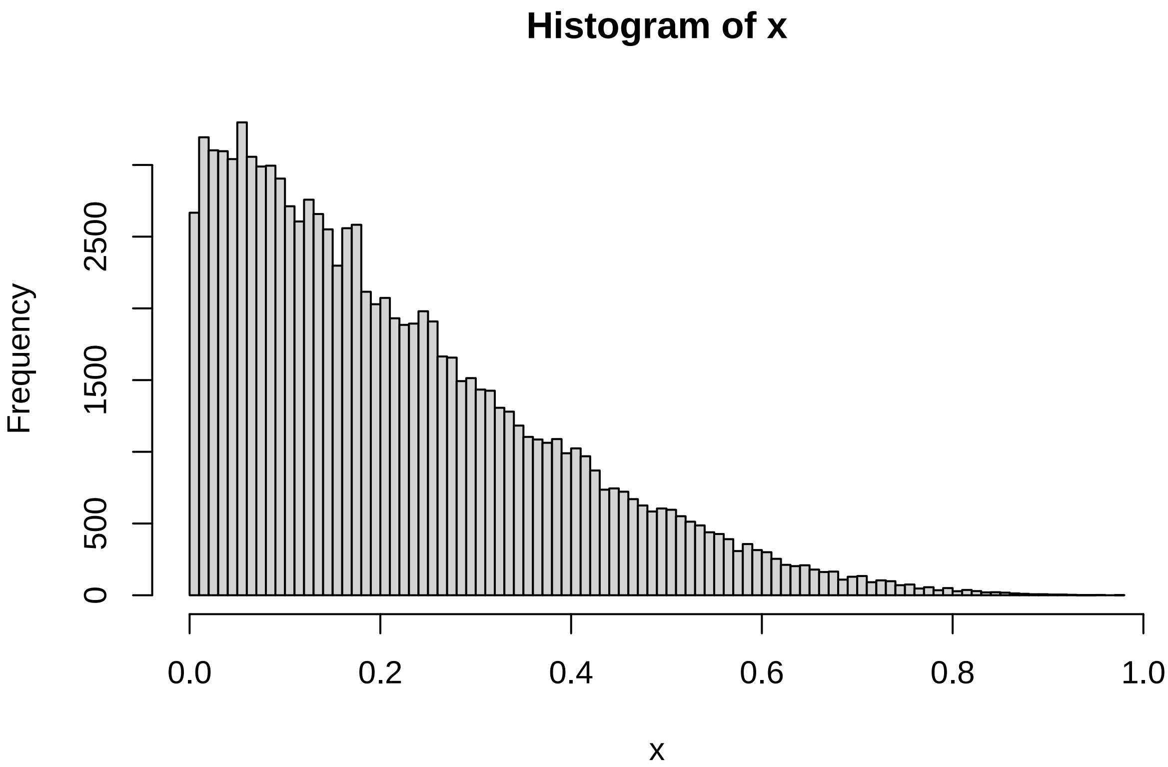
```
x=seq(from=0,to=1,length=1000)
y=dbeta(shape1=1.1,shape2=4,x=x)
plot(y~x,type='l')
```



Sampling from a Beta distribution using Metropolis

```
n=100000
x=rep(NA,n)
x[1]=.5

for(i in 2:n){
  candidate=runif(1)
  alpha=min(1,dbeta(shape1=1.1,shape2=4,x=candidate)/dbeta(shape1=1.1,shape2=4,x=x[i-1]))
  z=runif(1)<alpha
  x[i]=ifelse(z,candidate,x[i-1])
}
hist(x,100)
```



Tasks:

- Generate 100,000 samples from the mixture distribution of the first problem using $N(x[t-1], 1)$ to generate candidates.
- Hint: Create a function to evaluates the target density: `f=function(x,proportions, means, variances){ ...}`. Internally, the function should evaluate the mixture density for x, which is a weighted sum of `dnorm()` (see formula [1]).

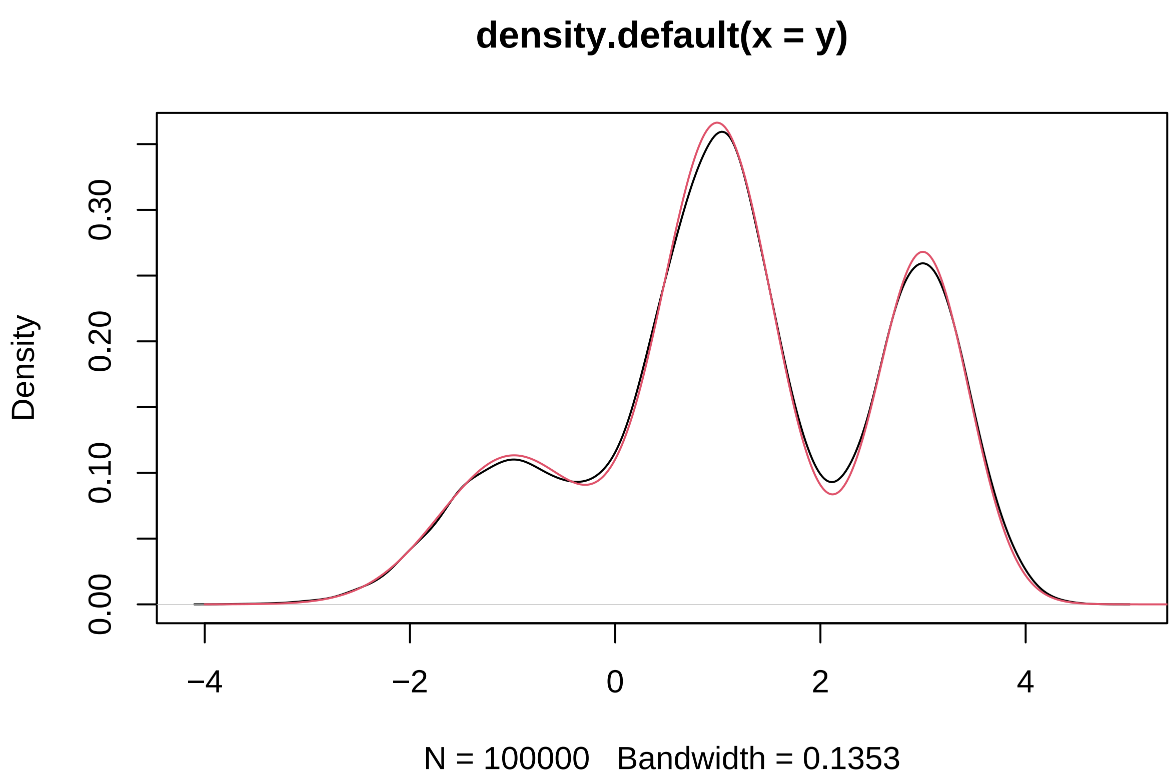
```
n=100000


y=rep(NA,n)
y[1]=0

for(i in 2:n){
  candidate=rnorm(mean=y[i-1],1) # here I use as the mean of the candidate generator to be the previou
  acceptanceProb=min(1,f(candidate,pi,mu,v)/f(y[i-1],pi,mu,v))
  accept=(runif(1)<acceptanceProb)
  if(accept){
    y[i]=candidate
  }else{
    y[i]=y[i-1]
  }
 }

plot(density(y))

z=seq(from=-4,to=8,by=.01)
y2=f(z,pi,mu,v)
lines(x=z,y=y2,col=2)
```




```
plot(y,xlim=c(1,300),type='o')
```