

# Homework 04

Shiva Kumar Peddapuram

## Instructions

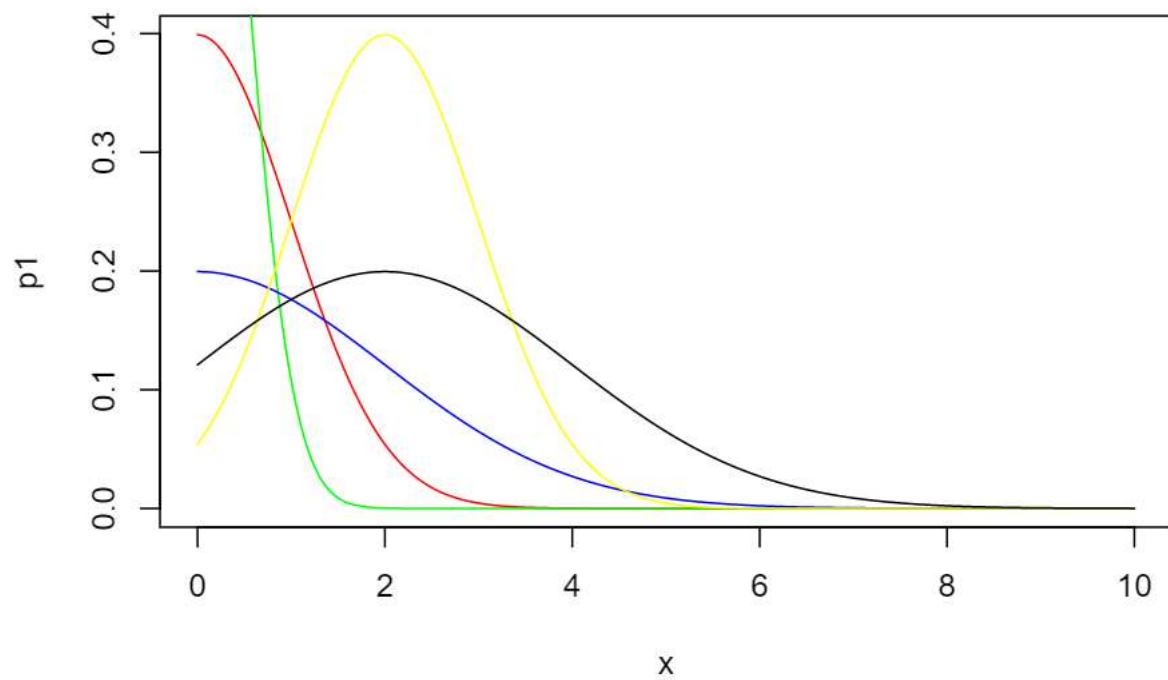
- Do not make changes to this file above this line.
- Do not change the statements of the questions
- Insert your computations in the “code chunks” provided (between three backtics)
- Add textual answers, outside the code chunks
- Click on the “Knit” button above in order to run the computations and create the pdf file (you might have to allow pop-ups in your web browser to see the pdf)
- Download the final PDF file to your personal computer, then upload it to the Canvas assignment
- **Deadline:** *Tuesday November 22nd, by 11:59 pm*

## Questions

1. We are going to study the functions `dnorm`, `pnorm`, and `qnorm`, for different values of the parameters. Consider these values of  $\mu$  and  $\sigma$ :  $(\mu_1, \sigma_1) = (0, 1)$ ,  $(\mu_2, \sigma_2) = (0, 1/2)$ ,  $(\mu_3, \sigma_3) = (0, 2)$ ,  $(\mu_4, \sigma_4) = (2, 1)$ ,  $(\mu_5, \sigma_5) = (2, 2)$ . Use the `curve` function in R to:
  - a. Plot the five density functions, on the same graph, using different colors.

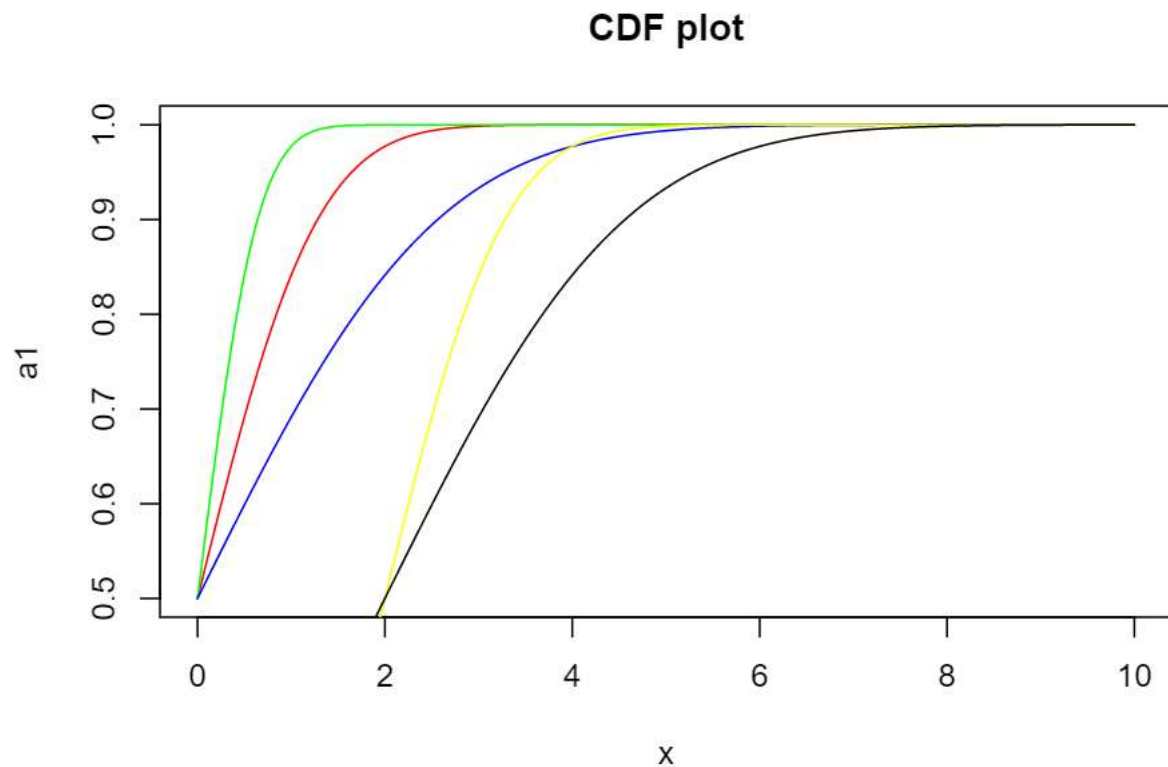
```
mean = c(0,0,0,2,2)
sd = c(1,0.5,2,1,2)
x<-seq(0,10,by=0.01)
p1<-dnorm(x,mean=0,sd=1)
p2<-dnorm(x,mean =0,sd=0.5)
p3<-dnorm(x,mean=0,sd=2)
p4<-dnorm(x,mean=2,sd=1)
p5<-dnorm(x,mean=2,sd=2)
plot(x,p1,type="l",col="red",main="PDF plot")
lines(x,p2,col="green")
lines(x,p3,col="blue")
lines(x,p4,col="yellow")
lines(x,p5,col="black")
```

## PDF plot



b. Plot the five CDFs, using the same colors as above.

```
a1<-pnorm(x,mean=0,sd=1)
a2<-pnorm(x,mean =0,sd=0.5)
a3<-pnorm(x,mean=0,sd=2)
a4<-pnorm(x,mean=2,sd=1)
a5<-pnorm(x,mean=2,sd=2)
plot(x,a1,type="l",col="red",main="CDF plot")
lines(x,a2,col="green")
lines(x,a3,col="blue")
lines(x,a4,col="yellow")
lines(x,a5,col="black")
```



c. Plot the five inverse CDFs, using the same colors as above.

```
b1<-seq(0,1,by=0.001)
c1<-qnorm(x,mean=0,sd=1)
```

```
## Warning in qnorm(x, mean = 0, sd = 1): NaNs produced
```

```
c2<-qnorm(x,mean =0,sd=0.5)
```

```
## Warning in qnorm(x, mean = 0, sd = 0.5): NaNs produced
```

```
c3<-qnorm(x,mean=0,sd=2)
```

```
## Warning in qnorm(x, mean = 0, sd = 2): NaNs produced
```

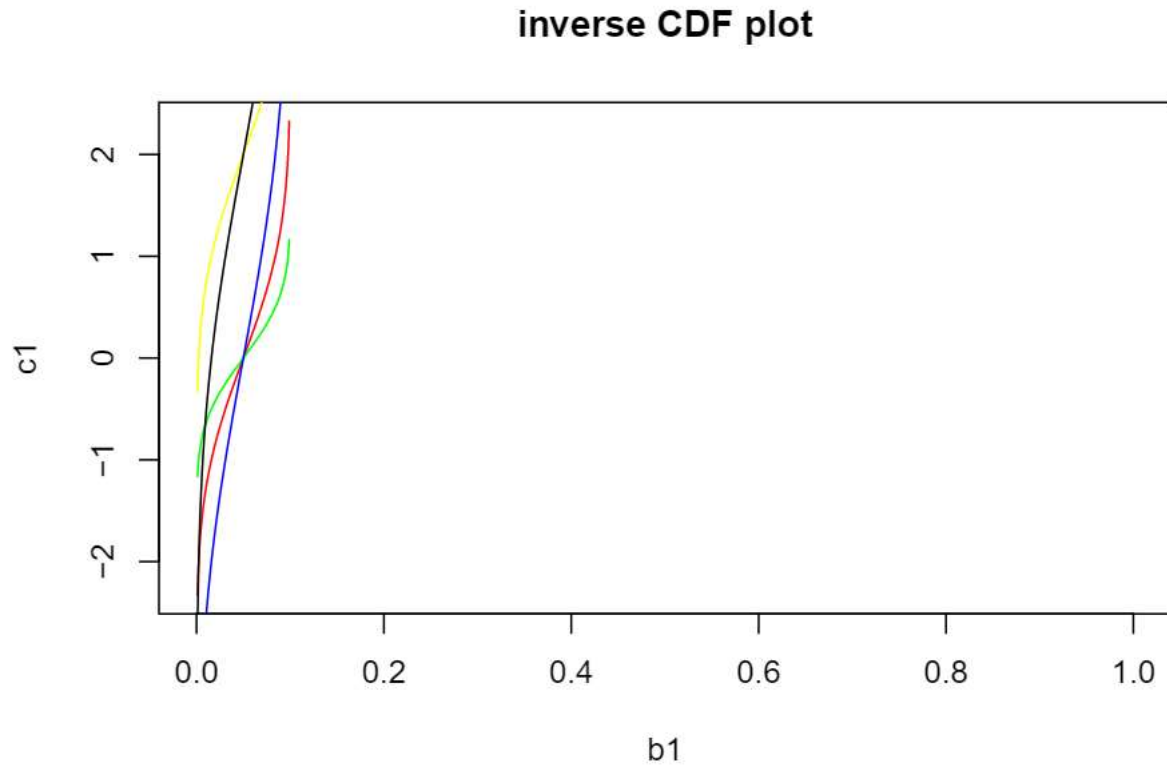
```
c4<-qnorm(x,mean=2,sd=1)
```

```
## Warning in qnorm(x, mean = 2, sd = 1): NaNs produced
```

```
c5<-qnorm(x,mean=2,sd=2)
```

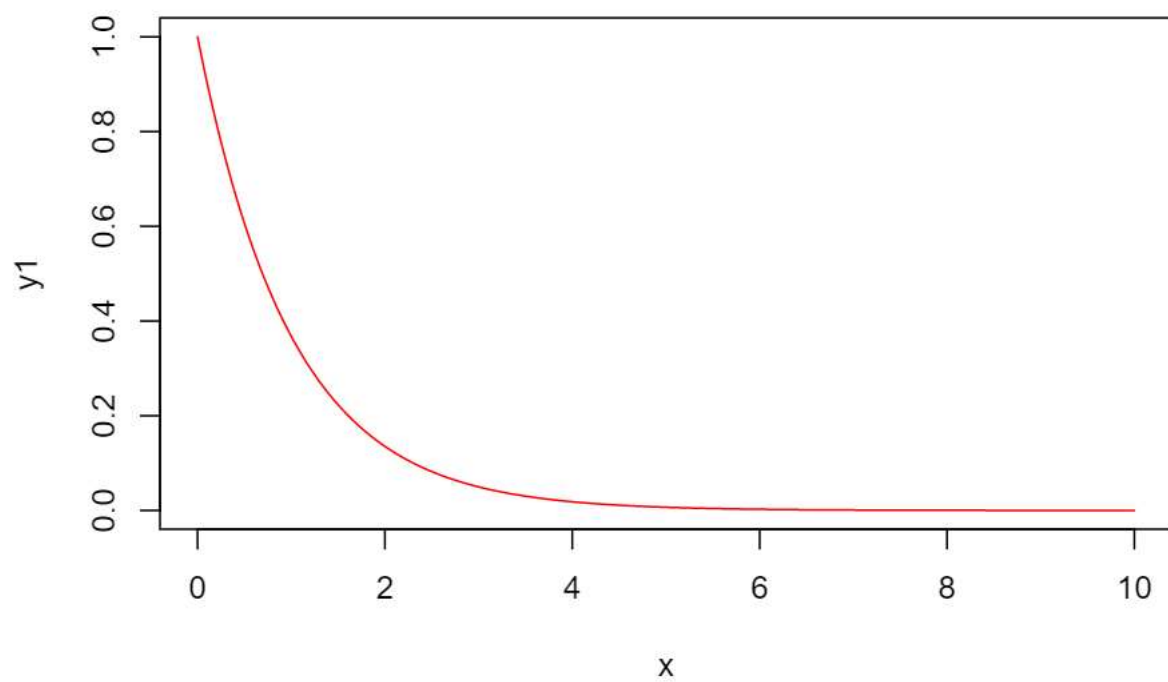
```
## Warning in qnorm(x, mean = 2, sd = 2): NaNs produced
```

```
plot(b1,c1,,type ="l" ,col="red",main="inverse CDF plot")
lines(b1,c2,col="green")
lines(b1,c3,col="blue")
lines(b1,c4,col="yellow")
lines(b1,c5,col="black")
```



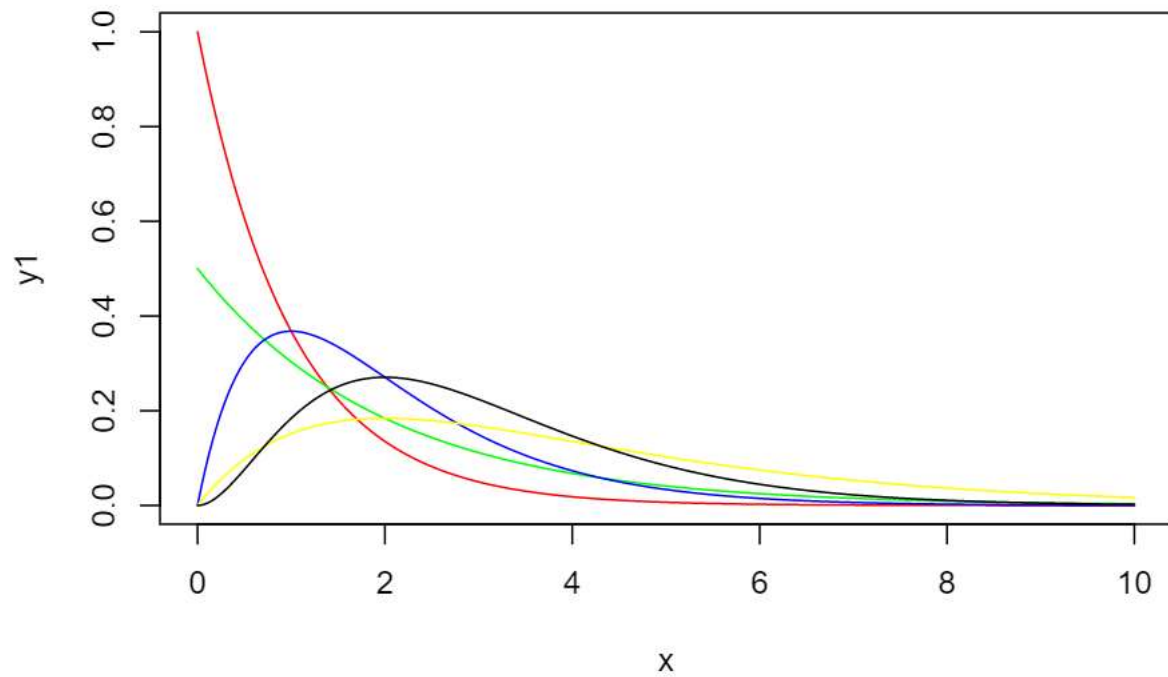
2. Repeat the steps in Question 1, but for the Gamma distribution. Here are the five pairs of **rate** and **scale** values:

```
x<-seq(0,10,by=0.01)
y1<-dgamma(x,shape=1,scale=1)
y2<-dgamma(x,shape=1,scale=2)
y3<-dgamma(x,shape=2,scale=1)
y4<-dgamma(x,shape=2,scale=2)
y5<-dgamma(x,shape=3,scale=1)
plot(x,y1,type='l',col="red")
```



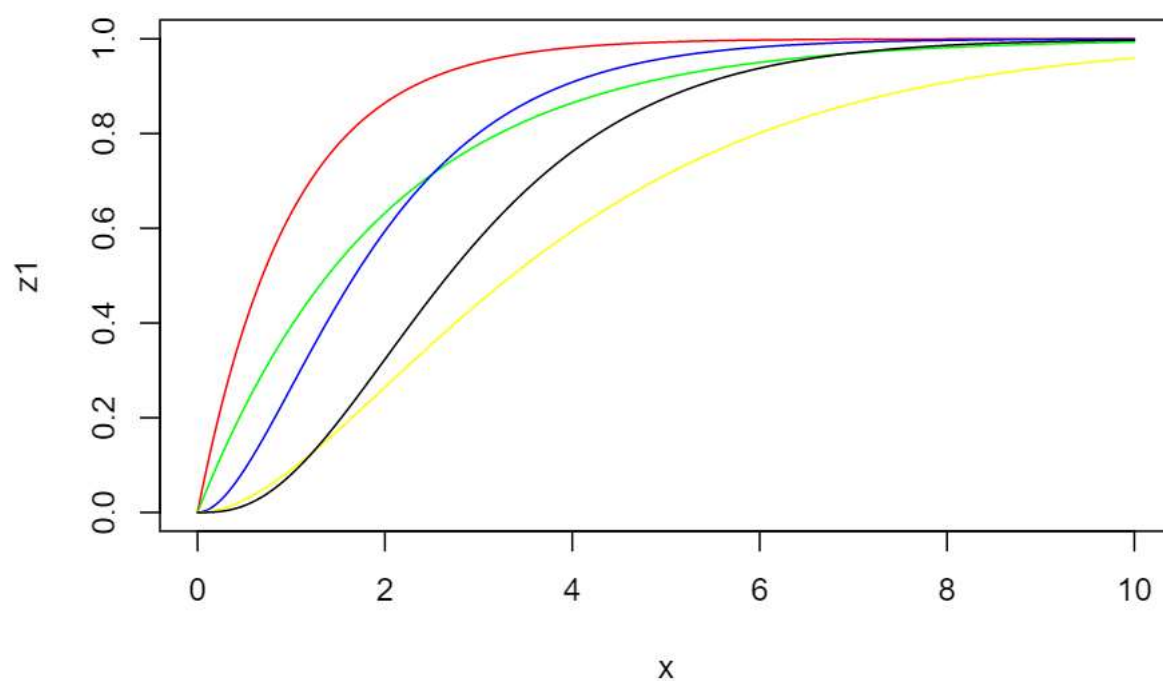
```
plot(x,y1,type='l',col="red",main="PDF plot")
lines(x,y2,col="green")
lines(x,y3,col="blue")
lines(x,y4,col="yellow")
lines(x,y5,col="black")
```

## PDF plot

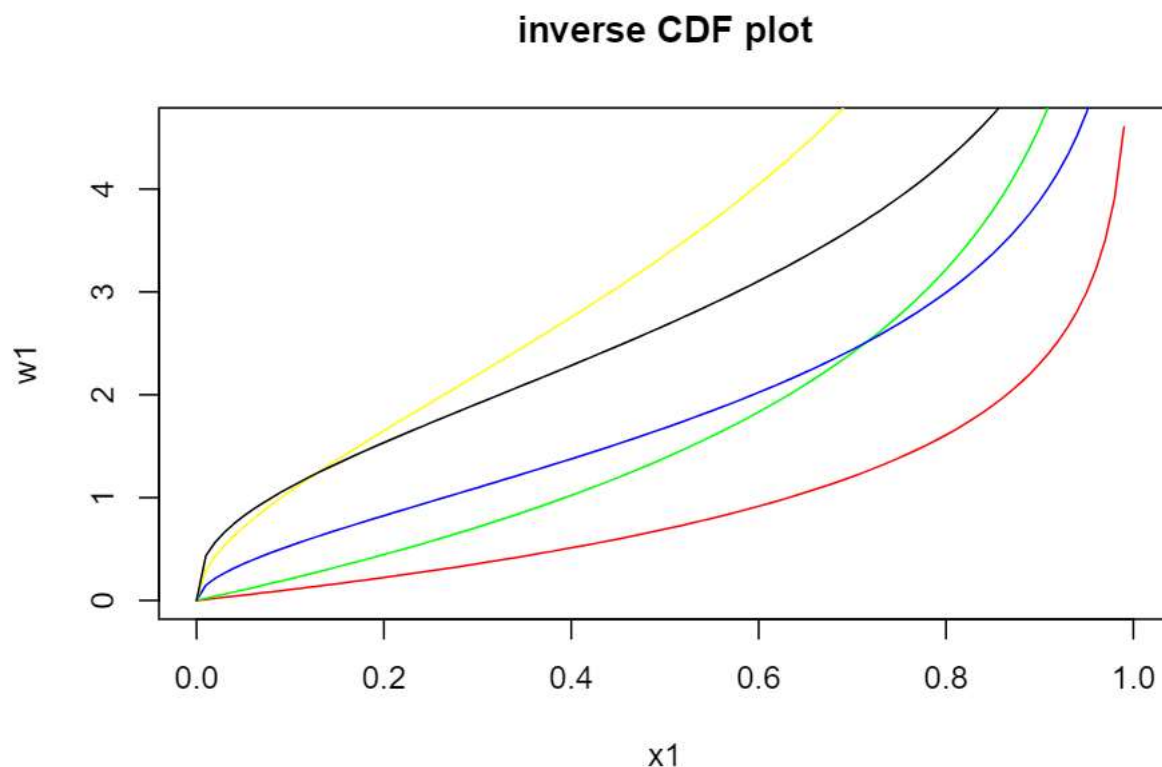


```
z1<-pgamma(x,shape=1,scale=1)
z2<-pgamma(x,shape=1,scale=2)
z3<-pgamma(x,shape=2,scale=1)
z4<-pgamma(x,shape=2,scale=2)
z5<-pgamma(x,shape=3,scale=1)
plot(x,z1,type='l',col="red",main="CDF plot")
lines(x,z2,col="green")
lines(x,z3,col="blue")
lines(x,z4,col="yellow")
lines(x,z5,col="black")
```

CDF plot



```
x1<-seq(0,1,by=0.01)
w1<-qgamma(x1,shape=1,scale=1)
w2<-qgamma(x1,shape=1,scale=2)
w3<-qgamma(x1,shape=2,scale=1)
w4<-qgamma(x1,shape=2,scale=2)
w5<-qgamma(x1,shape=3,scale=1)
plot(x1,w1,type='l',col="red",main="inverse CDF plot")
lines(x1,w2,col="green")
lines(x1,w3,col="blue")
lines(x1,w4,col="yellow")
lines(x1,w5,col="black")
```

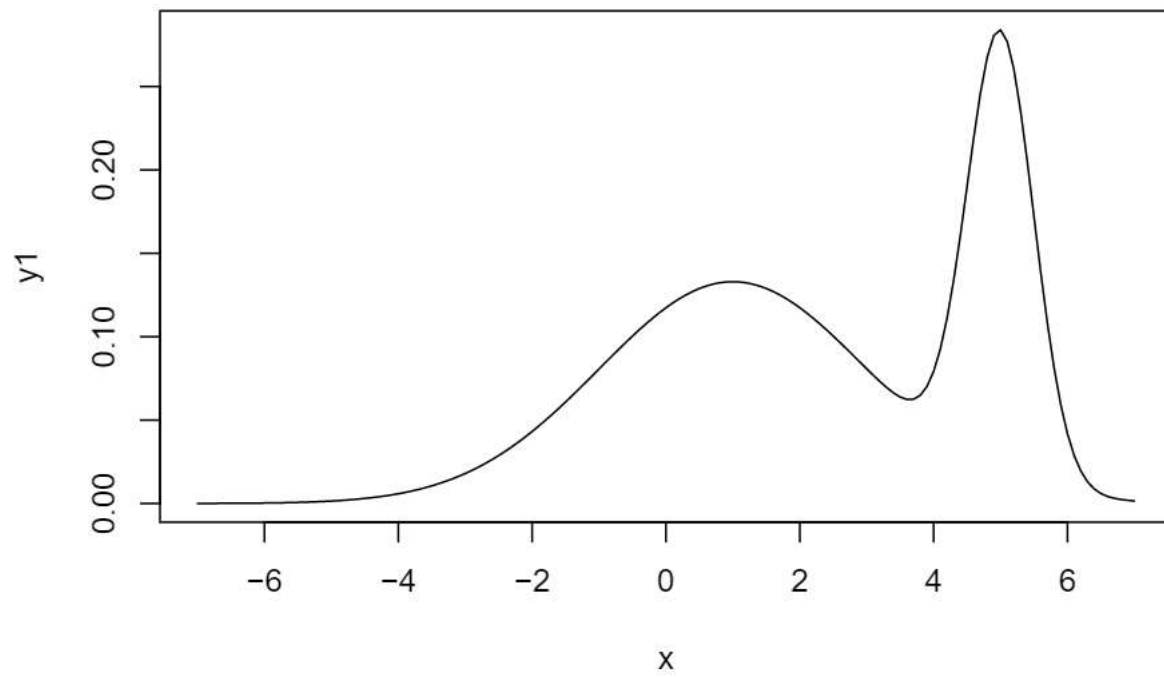


3. Consider a random variable  $X$  with value given by the following experiment: First, flip a coin, with probability of heads equal to  $2/3$  and probability of tails  $1/3$ . Then, if heads come up, let  $X$  be a random draw from  $\mathcal{N}(\mu = 1, \sigma = 2)$ ; if tails come up, let  $X$  be a random draw from  $\mathcal{N}(\mu = 5, \sigma = 1/2)$ . The distribution of  $X$  is called a *mixture distribution*. Let  $F_X$  and  $f_X$  be the CDF and pdf for  $X$ .
- Prove that  $F_X = \frac{2}{3}F_{\mathcal{N}(\mu=1, \sigma=2)} + \frac{1}{3}F_{\mathcal{N}(\mu=5, \sigma=1/2)}$
  - Prove that  $f_X = \frac{2}{3}f_{\mathcal{N}(\mu=1, \sigma=2)} + \frac{1}{3}f_{\mathcal{N}(\mu=5, \sigma=1/2)}$
  - Plot  $F_X$  and  $f_X$

```
x <- seq(-7,7,by=0.1)
y1 <- (2/3)*dnorm(x,1,2)+(1/3)*dnorm(x,5,1/2)
plot(x,y1, type="l",main = "PDF plot(part(b))")
```

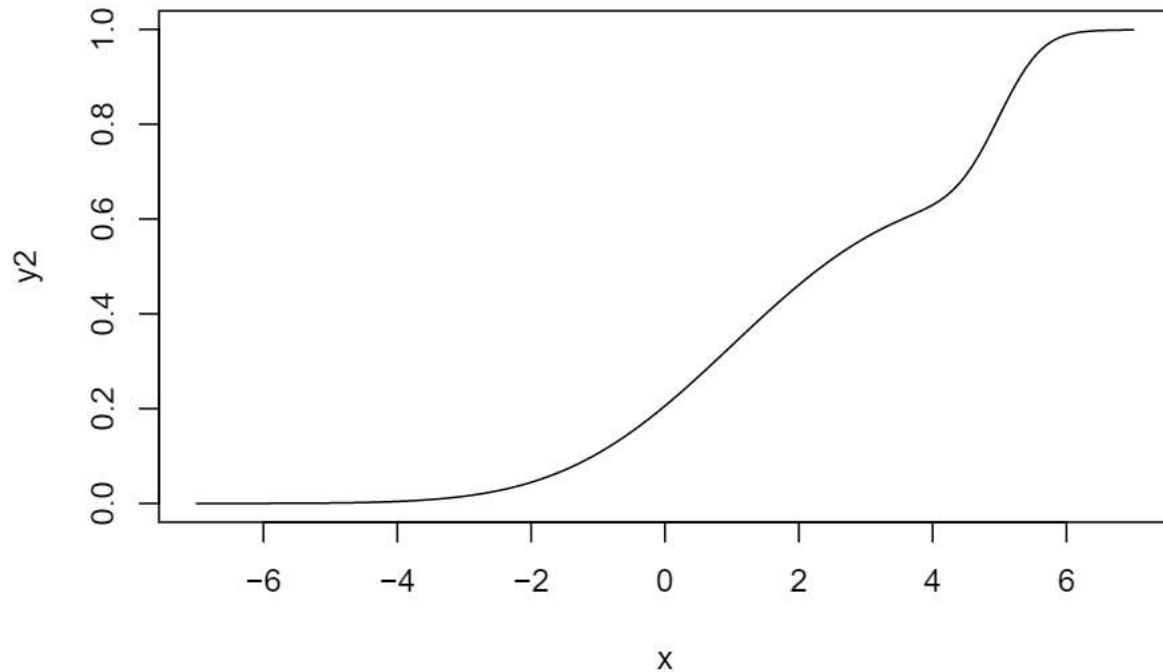


PDF plot(part(b))



```
y2 <- (2/3)*pnorm(x,1,2)+(1/3)*pnorm(x,5,1/2)
plot(x,y2, type = "l", main = "CDF plot(part(a))")
```

CDF plot(part(a))



d. Generate 1,000 random numbers, i.i.d. with the distribution of  $X$ . Plot a histogram of these data.

*Proof of (a)*  $F_X(x) = P(X \leq x) \dots F_X(x) = P(X \leq x) = P(X \leq x|head)P(head) + P(X \leq x|tail)P(tail) = \frac{2}{3}F_{\mathcal{N}(\mu=1, \sigma=2)} + \frac{1}{3}F_{\mathcal{N}(\mu=5, \sigma=1/2)}$

5. Using the same setup as in Question 3, define the following functions in R (keep the code as simple as you can; no need to optimize the code, or to validate the inputs):

a. `pmix`, with one parameter `q`, providing the values of  $F_X$

```
pmix<- function(q)
{
  (2/3)*pnorm(q,1,2)+(1/3)*pnorm(q,5,1/2)
}
q = seq(-7,7,by=0.1)
pmix(q)
```

```
## [1] 2.111416e-05 2.605040e-05 3.206423e-05 3.937261e-05 4.823203e-05
## [6] 5.894486e-05 7.186649e-05 8.741344e-05 1.060724e-04 1.284104e-04
## [11] 1.550861e-04 1.868622e-04 2.246195e-04 2.693719e-04 3.222828e-04
## [16] 3.846834e-04 4.580920e-04 5.442349e-04 6.450688e-04 7.628046e-04
## [21] 8.999320e-04 1.059246e-03 1.243876e-03 1.457308e-03 1.703420e-03
## [26] 1.986509e-03 2.311316e-03 2.683059e-03 3.107459e-03 3.590764e-03
## [31] 4.139777e-03 4.761874e-03 5.465024e-03 6.257804e-03 7.149407e-03
## [36] 8.149648e-03 9.268965e-03 1.051840e-02 1.190961e-02 1.345481e-02
## [41] 1.516675e-02 1.705871e-02 1.914437e-02 2.143785e-02 2.395355e-02
## [46] 2.670610e-02 2.971031e-02 3.298098e-02 3.653286e-02 4.038051e-02
```

```
## [51] 4.453813e-02 4.901951e-02 5.383777e-02 5.900533e-02 6.453366e-02
## [56] 7.043318e-02 7.671311e-02 8.338129e-02 9.044404e-02 9.790604e-02
## [61] 1.057702e-01 1.140374e-01 1.227068e-01 1.317750e-01 1.412369e-01
## [66] 1.510849e-01 1.613091e-01 1.718974e-01 1.828354e-01 1.941065e-01
## [71] 2.056917e-01 2.175701e-01 2.297188e-01 2.421129e-01 2.547257e-01
## [76] 2.675291e-01 2.804935e-01 2.935882e-01 3.067814e-01 3.200408e-01
## [81] 3.333333e-01 3.466259e-01 3.598852e-01 3.730785e-01 3.861731e-01
## [86] 3.991376e-01 4.119409e-01 4.245538e-01 4.369478e-01 4.490965e-01
## [91] 4.609750e-01 4.725602e-01 4.838313e-01 4.947693e-01 5.053576e-01
## [96] 5.155819e-01 5.254300e-01 5.348923e-01 5.439617e-01 5.526337e-01
## [101] 5.609071e-01 5.687847e-01 5.762757e-01 5.833977e-01 5.901826e-01
## [106] 5.966835e-01 6.029847e-01 6.092151e-01 6.155614e-01 6.222816e-01
## [111] 6.297119e-01 6.382629e-01 6.484002e-01 6.606046e-01 6.753129e-01
## [116] 6.928456e-01 7.133316e-01 7.366465e-01 7.623817e-01 7.898547e-01
## [121] 8.181666e-01 8.462984e-01 8.732310e-01 8.980639e-01 9.201126e-01
## [126] 9.389653e-01 9.544940e-01 9.668233e-01 9.762685e-01 9.832614e-01
## [131] 9.882768e-01 9.917748e-01 9.941600e-01 9.957632e-01 9.968370e-01
## [136] 9.975635e-01 9.980675e-01 9.984304e-01 9.987031e-01 9.989166e-01
## [141] 9.990895e-01
```

b. 'dmix', with one parameter 'x', providing the values of \$f\_X\$

```
dmix<- function(x)
{
  (2/3)*dnorm(x,1,2)+(1/3)*dnorm(x,5,1/2)
}
x = seq(-7,7,by=0.1)
dmix(x)
```

```
## [1] 4.461008e-05 5.441880e-05 6.621849e-05 8.037553e-05 9.731564e-05
## [6] 1.175319e-04 1.415934e-04 1.701550e-04 2.039673e-04 2.438882e-04
## [11] 2.908942e-04 3.460938e-04 4.107397e-04 4.862436e-04 5.741896e-04
## [16] 6.763494e-04 7.946961e-04 9.314195e-04 1.088940e-03 1.269921e-03
## [21] 1.477283e-03 1.714214e-03 1.984177e-03 2.290922e-03 2.638484e-03
## [26] 3.031188e-03 3.473645e-03 3.970748e-03 4.527656e-03 5.149782e-03
## [31] 5.842767e-03 6.612451e-03 7.464843e-03 8.406073e-03 9.442346e-03
## [36] 1.057988e-02 1.182486e-02 1.318335e-02 1.466120e-02 1.626401e-02
## [41] 1.799699e-02 1.986490e-02 2.187194e-02 2.402162e-02 2.631672e-02
## [46] 2.875911e-02 3.134969e-02 3.408831e-02 3.697361e-02 4.000300e-02
## [51] 4.317253e-02 4.647686e-02 4.990916e-02 5.346111e-02 5.712286e-02
## [56] 6.088303e-02 6.472868e-02 6.864542e-02 7.261739e-02 7.662738e-02
## [61] 8.065691e-02 8.468635e-02 8.869508e-02 9.266163e-02 9.656385e-02
## [66] 1.003791e-01 1.040846e-01 1.076575e-01 1.110749e-01 1.143146e-01
## [71] 1.173551e-01 1.201757e-01 1.227567e-01 1.250801e-01 1.271293e-01
## [76] 1.288894e-01 1.303476e-01 1.314931e-01 1.323175e-01 1.328146e-01
## [81] 1.329808e-01 1.328146e-01 1.323175e-01 1.314931e-01 1.303476e-01
## [86] 1.288894e-01 1.271293e-01 1.250801e-01 1.227567e-01 1.201757e-01
## [91] 1.173551e-01 1.143146e-01 1.110749e-01 1.076576e-01 1.040850e-01
## [96] 1.003801e-01 9.656649e-02 9.266839e-02 8.871171e-02 8.472565e-02
## [101] 8.074613e-02 7.682201e-02 7.302533e-02 6.946690e-02 6.631808e-02
## [106] 6.383759e-02 6.239983e-02 6.251642e-02 6.483884e-02 7.012658e-02
## [111] 7.916651e-02 9.263644e-02 1.109208e-01 1.339066e-01 1.608071e-01
## [116] 1.900729e-01 2.194444e-01 2.461714e-01 2.673854e-01 2.805600e-01
```

```
## [121] 2.839585e-01 2.769591e-01 2.601746e-01 2.353331e-01 2.049526e-01
## [126] 1.718937e-01 1.388997e-01 1.082244e-01 8.141207e-02 5.924589e-02
## [131] 4.183674e-02 2.879951e-02 1.945734e-02 1.302606e-02 8.750613e-03
## [136] 5.985753e-03 4.227876e-03 3.112402e-03 2.392112e-03 1.908845e-03
## [141] 1.566503e-03
```

c. 'rmix', with one parameter 'n', for generating  $n$  random numbers i.i.d. with the distribution of  $X$

```
rmix<- function(n)
{
  N<-1000
  n<-rep(0,N)
  for(i in 1:N)
  {
    s<-sample(c(1,0),size=1,prob=c(2/3,1/3))
    if(s==1)
    {
      n[i]<-rnorm(1,1,2)
    }
    else
    {
      n[i]<-rnorm(1,5,1/2)
    }
  }
}
```

d. 'qmix', with one parameter 'p', providing the values of  $F_X^{-1}(p)$  (the inverse CDF, i.e., the quantile function). \*Hint:\* Use the function 'uniroot', with the function 'pmix' and option 'extendInt="yes"'

```
qmix<- function(p)
{
  (2/3)*qnorm(q,1,2)+(1/3)*qnorm(q,5,1/2)
}
q = seq(-0.1,0.1,by=1)
```

6. (BONUS) Repeat Question 5, but for a general mixture of two normal distributions, with extra parameters: mean1, sd1, mean2, sd2, and mixprob. Call your functions pnormmix, dnormmix, qnormmix, and rnormmix. Make sure that qnormmix can accept p as a vector (it is OK to use a for loop to deal with that), and that values of p equal to 0 or 1 produce the appropriate infinities.

```
qmix<- function(p) { (2/3)qnorm(q,1,2)+(1/3)qnorm(q,5,1/2) } q = seq(-0.1,0.1,by=1)
```