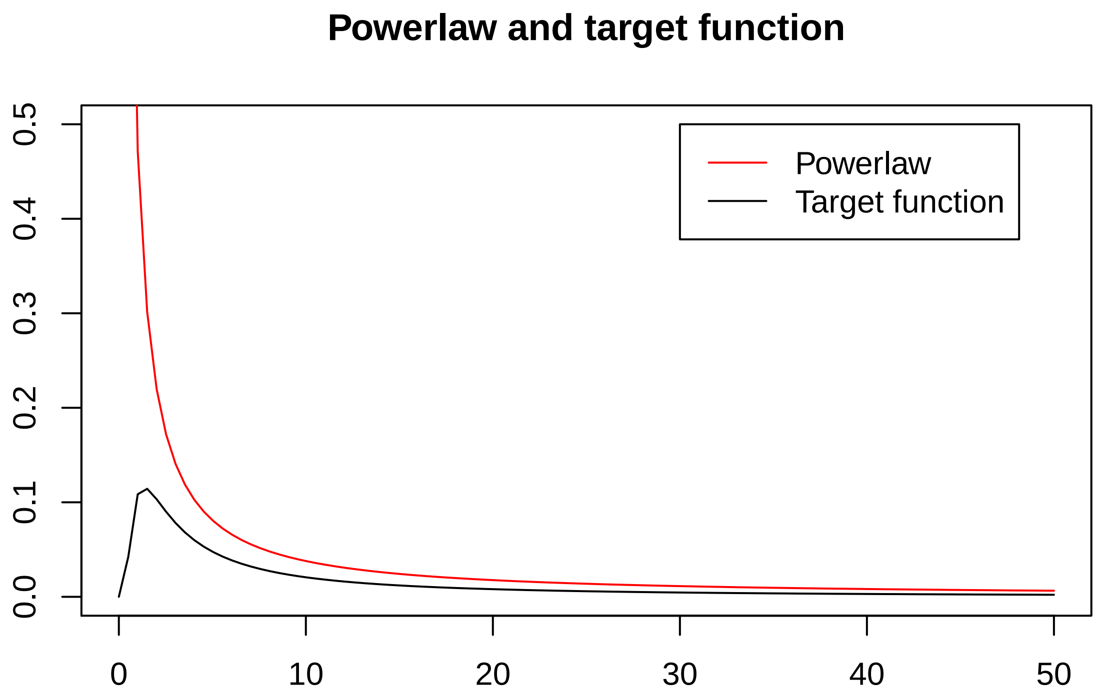
Lab 3

Shwetha, Suhani , Hoda

11/21/2020

1

```
f = function(x,c){
  res = 0
  res = c * ((sqrt(2*pi))^-1) * exp(-(c^2)/(2*x)) * x^(-3/2)
  res[x<=0]<-0
  return(res)
}
fp = function(x, a, tmin) {
  return((a - 1 / tmin) * (x / tmin)^(-a))
  }
c = 2
tmin =  1.33
a = 1.1
x = seq(0,50,length.out = 100)
plot(x, f(x,c), type = "l", ylim = c(0,0.5),xlab = "", ylab = "")
lines(x,fp(x,a,tmin),col="red")
title("Powerlaw and target function")
legend(30,0.5,legend = c("Powerlaw","Target function"), lty = 1, col = c("red","black"))
```



The support for power law is from tmin to infinity , but our target function is on support from 0 to infinity. Hence we cannot use power law by itself. However we can combine it with uniform distribution for 0 to tmin support. Since the max(f(x)) is achieved when the value of x is c^2/3, we have choosen tmin value to be the same, because powerlaw is strictly decreasing funtion and maximum of it will be at tmin. Alpha has been choosen by visually deciding which value makes powerlaw function similar to our target function.

Majorizing density function is combination of uniform density and powerlaw density function.

$$Majoritydensity = (probability * uniformdensity * 1[0, tmin]) + (1 - probability * powerlawdensity * 1[tmin, \inf]) whereprobabli$$

Then we can find majorizing constant as follows

$$MajoringConstant = max\left(\frac{f(x)}{majoritydensity(x)}\right)$$

```
p = 0.0832 #for c = 2
#x = seq(0,100,length.out = 1000)
max = max(f(x,c))
nd = function(x,tmin,a,p){
  res = c()
  for(i in 1:length(x)){
    if(x[i]>=0 && x[i]<= tmin){
      res[i] = p * dunif(x[i],0,tmin)
    }
    if(x[i]>tmin){
      res[i] = (1-p) * fp(x[i],a,tmin)
    }
```

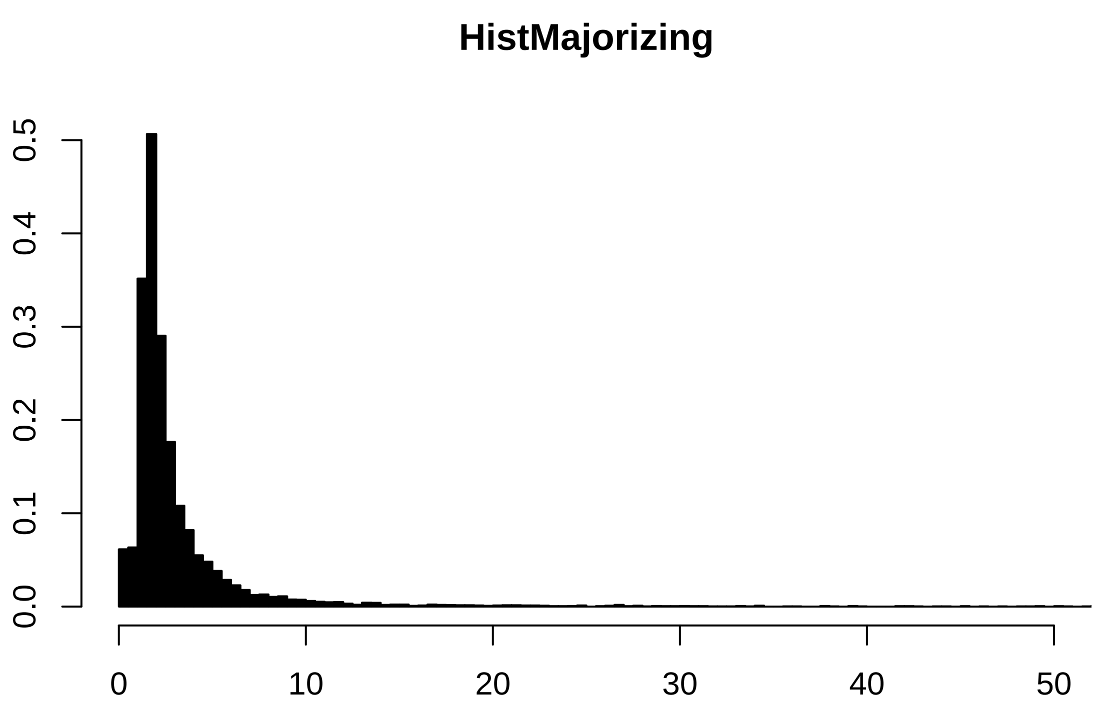
```
  }
  return(res)
}

c_maj = max(f(x,c)/as.vector(nd(x,tmin,a,p)))
```

2

Acceptance-rejection algorithm :

```
a =2.5
rmajorizing<-function(n){
  sapply(1:n,function(i){
    res<-NA
    component<-sample(1:2,1,prob = c(0.0833,0.9167))
    if(component==1){res<-runif(1,0,tmin)}
    if(component==2){res<-rplcon(1,tmin,a)}
    res
  })
}
Nsample<-10000
num_histbreaks<-100
hist(rmajorizing(Nsample),breaks=2000,col="black",xlab="",ylab="",main="HistMajorizing",freq=FALSE, xli
```



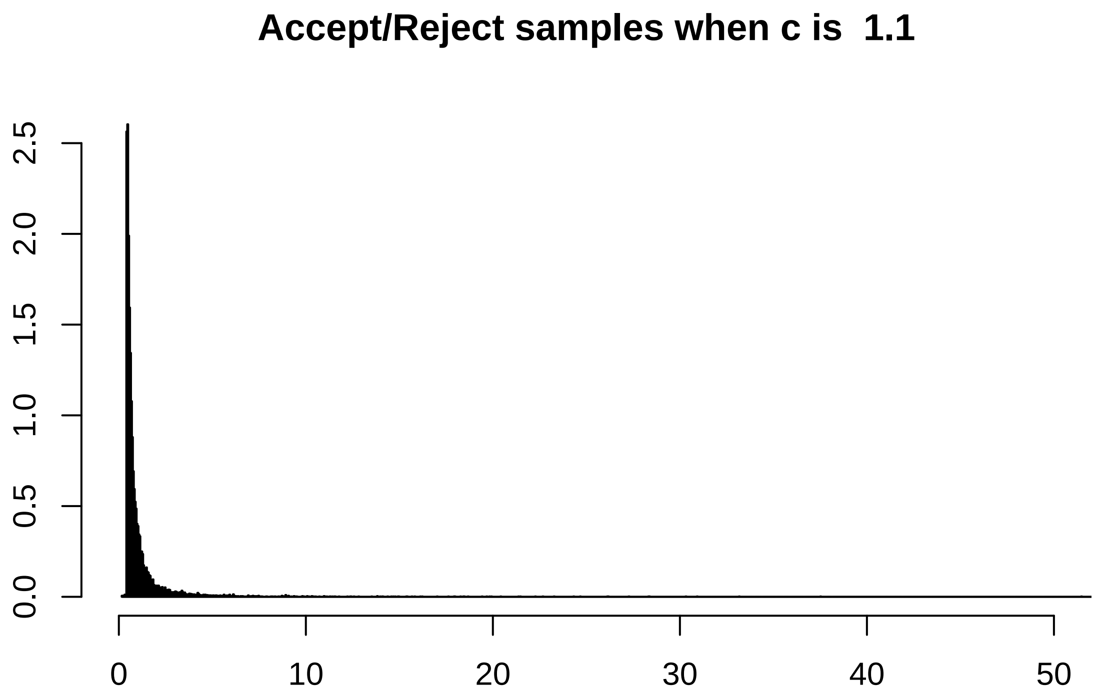
```
accept_reject<-function(c,c_maj,t){
  x<-NA
  num_reject<-0
  while (is.na(x)){
    y<-rmajorizing(1)
    u<-runif(1)
    if (u<=f(y,c)/(c_maj*nd(y,a,tmin = t,p))){
      x<-y
    } else{
      num_reject<-num_reject+1
    }
  }
 res = c(x,num_reject)
 return(res)
}
```

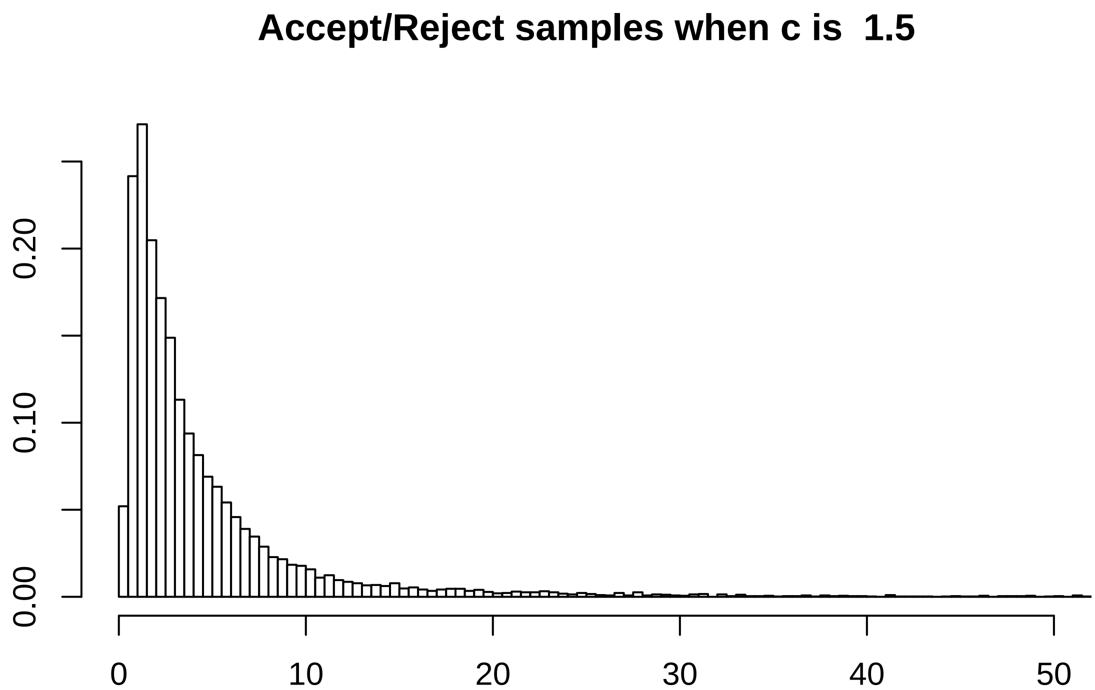
3

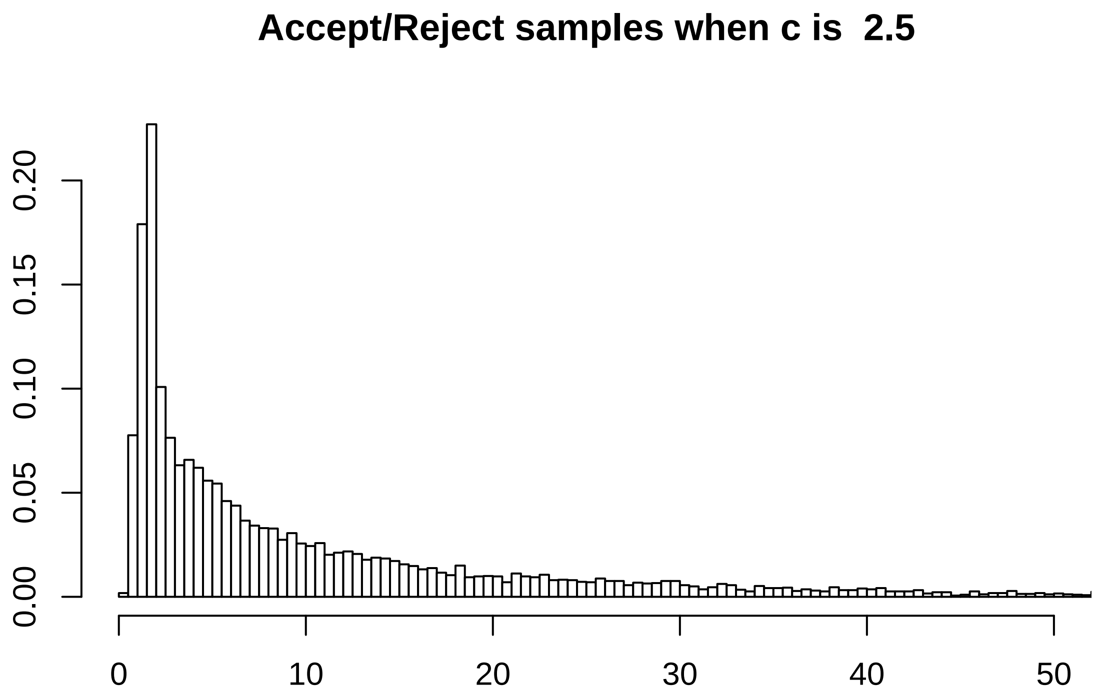
Generating large samples from above sampler to different values of c. When we compare the results of f(x) for different

```
set.seed(12345)
c = c(1.1,1.5,2.5)
p = 0.0833
t = c(0.4033333,0.75,2.0833) # tmin chosen as c^2 / 3
cm = c(max(f(x,1.1)/nd(x,a,t[1],p)),max(f(x,1.5)/nd(x,a,t[2],p)),max(f(x,2.5)/nd(x,a,t[3],p)))
mean=var=rejection_rate=c()
for(i in 1:length(c)){
tmin = t[i]
fx_acceptreject<-sapply(rep(c[i],Nsample),accept_reject,c_maj = cm[i], t = tmin)

hist(fx_acceptreject[1,],col="white",border = "black",breaks=10000,xlab="",ylab="",freq=FALSE,main="",xl
title(capture.output(cat("Accept/Reject samples when c is ", c[i])))
mean =cbind(mean, mean(fx_acceptreject[,1]))
var = cbind(var,var(fx_acceptreject[,1]))
rejection_rate = cbind(rejection_rate,sum(fx_acceptreject[,2])/Nsample)
}
```


Accept/Reject samples when c is 1.1
0.0
0.5
1.0
1.5
2.0
2.5
0
10
20
30
40
50


Accept/Reject samples when c is 1.5
0.00
0.10
0.20
0
10
20
30
40
50



Mean for c = 1.1 is 0.8100103

Mean for c = 1.5 is 5.6142188

Mean for c = 2.5 is 6.0692443

Variance for c = 1.1 is 1.3122333

Variance for c = 1.5 is 11.3839036

Variance for c = 2.5 is 48.6247028

Rejection rate for c = 1.1 is 6.0560238×10^{-5}

Rejection rate for c = 1.5 is 3.4814113×10^{-4}

Rejection rate for c = 2.5 is 0.0018567

We can see that as value of c increases, the mean and variance also increases. This is because as c increases , target function peak moves further towards right on x as max of target function is achieved when x is c^2 / 3. As c increases , the max of target function decreases, hence reducing the sharp peaks and making it smoother. This inturn increases the values being sampled by accept reject algorithm, thus increasing mean and variance. Rejection rate also increases as the value of c increases.The proposal density goes way higher than our target density. This results in increase in number of rejections.

2