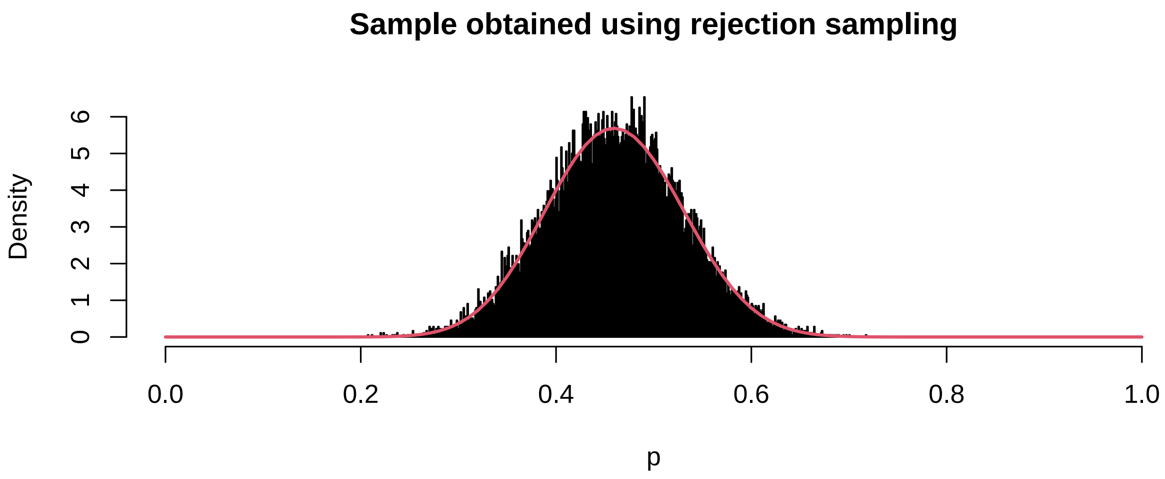
Lecture 10 R scripts

Rejection sampling

To demonstrate rejection sampling, consider the case where we wish to draw a sample from a beta distribution. In this example, assume $\alpha = \beta = 0.5$, n can be either 5, 20, or 50, and $y \sim \text{Bin}(n, 0.4)$.

```
set.seed(123456)
a=b=0.5 #Hyper-parameters
n<-50
y<-rbinom(1,n,0.4) #simulate a trial.

pmax <- (y+a-1)/(n+a+b-2) #Posterior mode
M=pmax^(y+a-1)*(1-pmax)^(n-y+b-1)/beta(y+a,n-y+b) #Construct upper bound.
x<-runif(100000,0,1) #uniform number for accept-reject rule.
sam.cand<-runif(100000,0,1) #candidate sample
sam.keep<-sam.cand[x < dbeta(sam.cand,y+a,n-y+b)/M] #retained sample
#Construct histogram of retained sample and compare to analytic posterior
hist(sam.keep,breaks=500,xlab='p',xlim=c(0,1),
     main='Sample obtained using rejection sampling', freq=FALSE)
curve(dbeta(x,y+a,n-y+b),xlab='p',ylab='p(p|y)',add=TRUE,col=2,lwd=2)
```



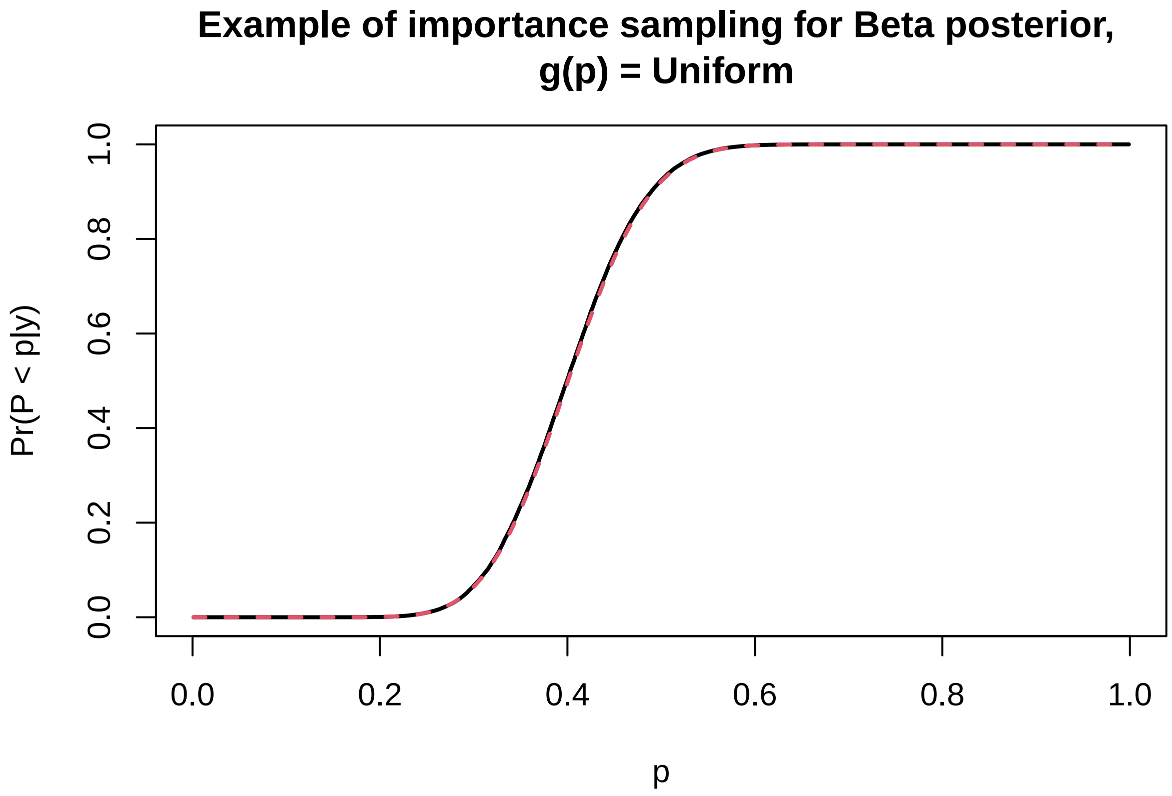
```
length(sam.keep) #Determine the number of candidate samples that were kept.
```

```
## [1] 17585
```

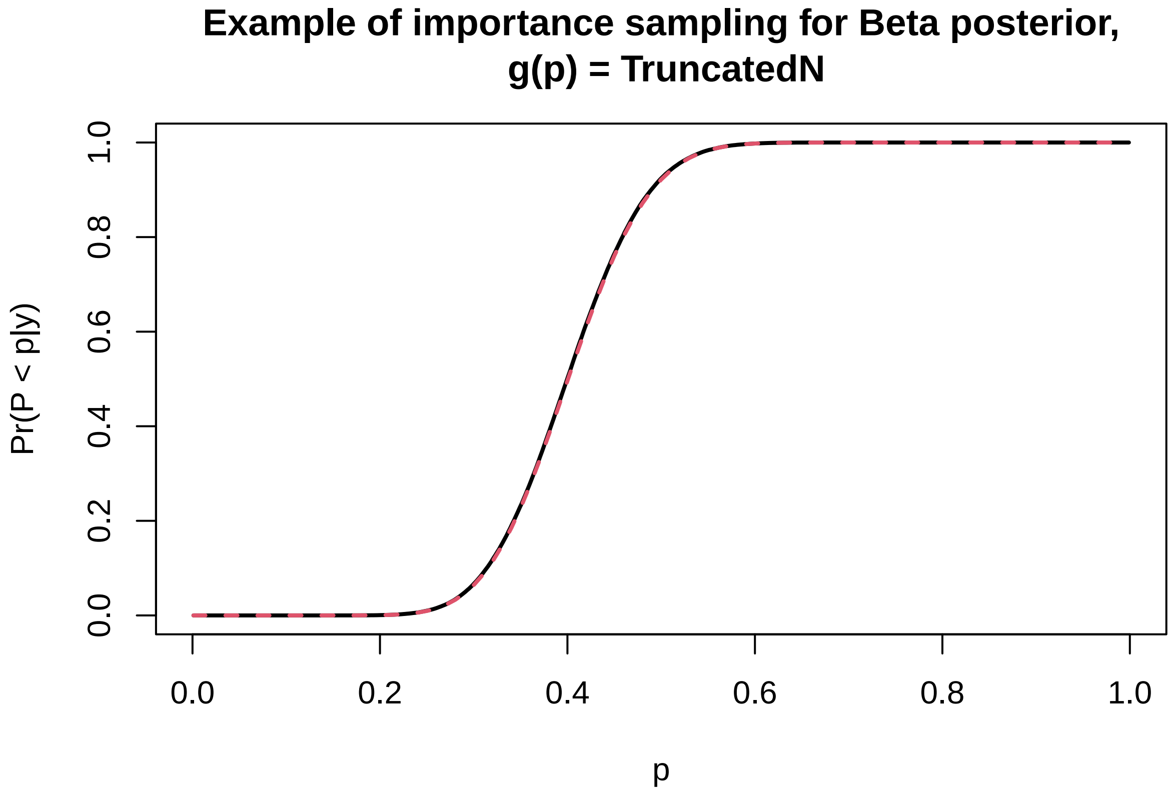
Importance sampling

To demonstrate Importance sampling, we will use the same example as we did in Rejection sampling. We will consider two possibilities for $g(p)$, the standard uniform and a truncated normal with mean and variance determined by finding the posterior mode.

```
#Uniform distribution for g(p)
#hyper-parameters and data.
a=b=0.5
n<-50
y<-rbinom(1,n,0.4)
x1<-runif(100000,0,1) #candidates, theta^s.

Qp <- function(p){p^(y+a-1)*(1-p)^(n-y+b-1)} #Un-normalised density
wp<-sapply(x1,FUN=Qp,simplify=TRUE) #Note as g() is standard uniform g(p) = 1 for all p.
N<-500
#Estimate cdf.
beta.cdf<-0
for(i in 1:500){
ind<-which(x1 < i/N)
beta.cdf[i] <- sum(wp[ind])/sum(wp)
}
#Look at the retained sample.
plot((1:500-0.5)/N,beta.cdf,xlab='p',ylab='Pr(P < p|y)',type='l',lwd=2,
     main='Example of importance sampling for Beta posterior, \n g(p) = Uniform')
curve(pbeta(x,y+a,n-y+b),xlab='p',ylab='Pr(P< p|y)', add=TRUE, col=2, lty=2, lwd=2)
```

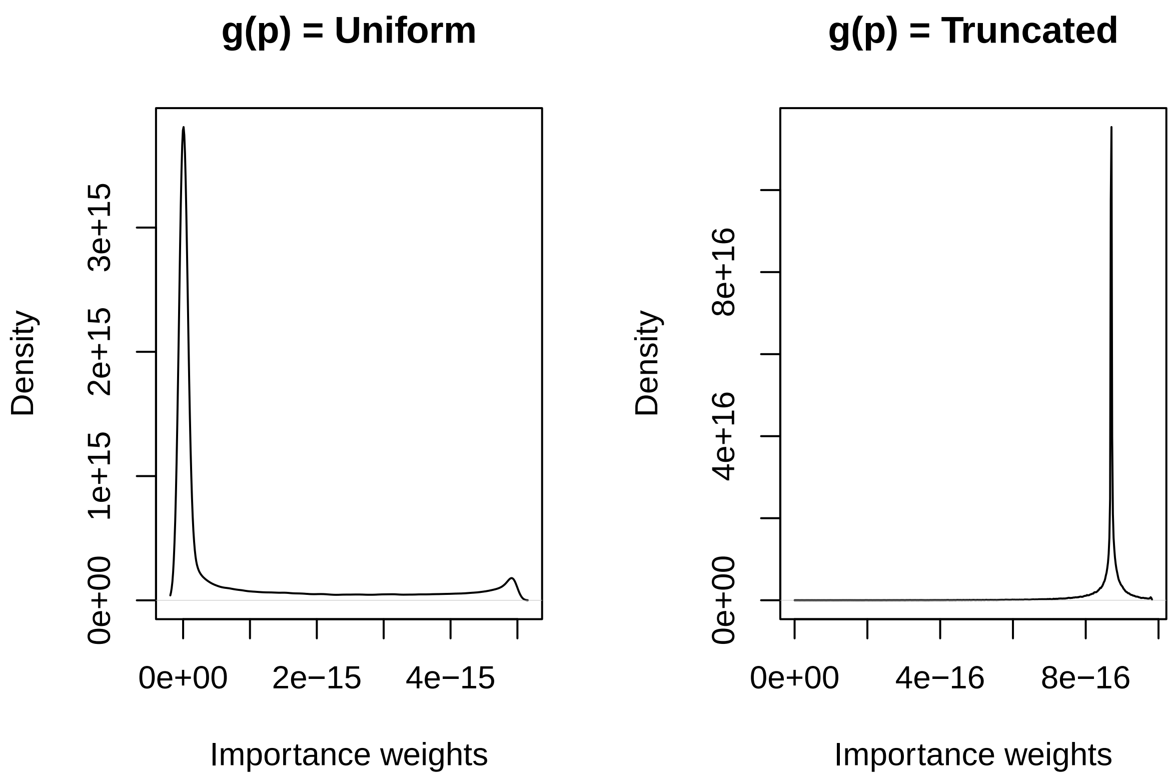


```
#Note that by choosing a standard uniform for g(p), the numerical integration
#ends up looking very like direct approximation of the posterior.
#Version two: Truncated normal for g(p),
pmax <- (y+a-1)/(n+a+b-2) #Posterior mode estimates of mean and variance.
vpmax<- pmax*(1-pmax)/(n+a+b-2)
x2<-rnorm(130000,mean=pmax,sd=sqrt(vpmax)) #Candidates from truncated normal.
x2<-x2[x2<=1];x2<-x2[x2>=0]
#Note as trucated normal is proportional to normal,
#so an un-normalised density for g is sufficient for finding draws.
wp2<-sapply(x2,FUN=Qp,simplify=TRUE)/dnorm(x2,mean=pmax,sd=sqrt(vpmax)) #Calculate weights.
N<-500
#Estimate cdf.
beta.cdf2<-0
for(i in 1:500){
ind<-which(x2 < i/N)
beta.cdf2[i] <- sum(wp2[ind])/sum(wp2)
}
#Compare
plot((1:500-0.5)/N,beta.cdf2,xlab='p',ylab='Pr(P < p|y)',type='l',lwd=2,
main='Example of importance sampling for Beta posterior,\n g(p) = TruncatedN')
curve(pbeta(x,y+a,n-y+b),xlab='p',ylab='Pr(P< p|y)',add=TRUE,col=2,lty=2,lwd=2)
```



```
#Compare distribution of weights
par(mfrow=c(1,2))
plot(density(wp),xlab='Importance weights',main='g(p) = Uniform');
```

```r
plot(density(wp2),xlab='Importance weights', main='g(p) = Truncated')
```



```r
summary(wp);summary(wp2)
```

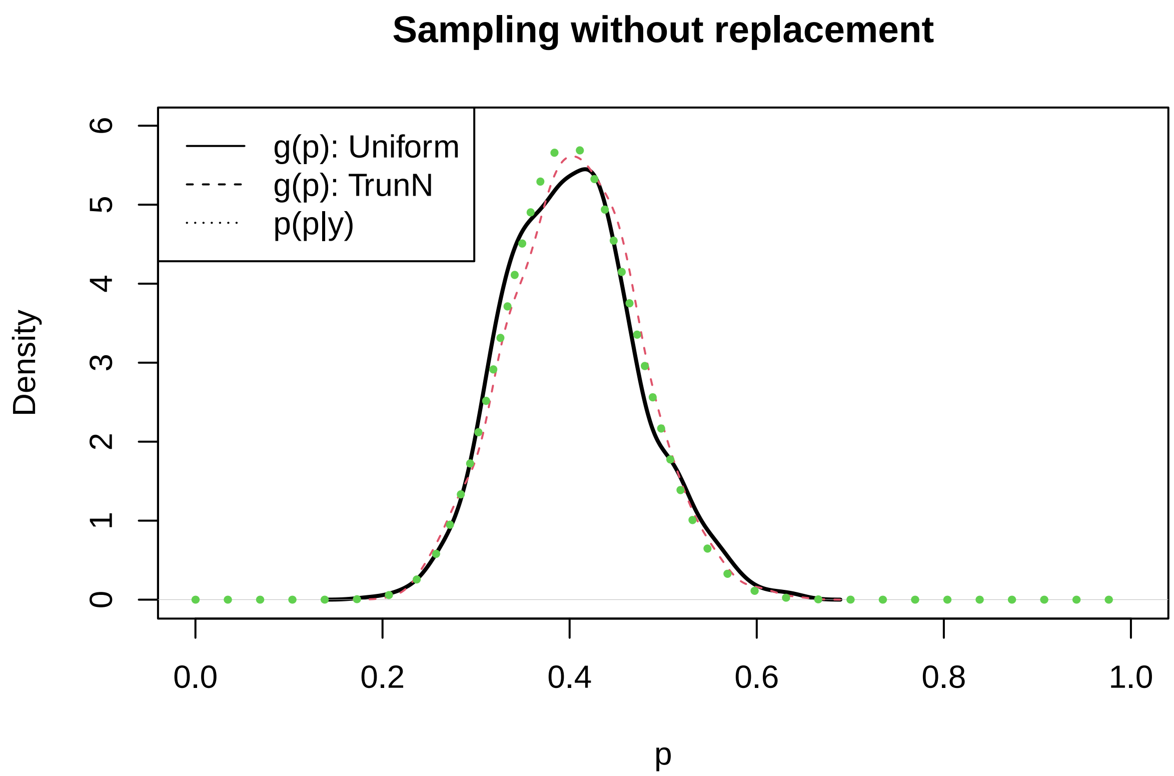
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.000e+00 0.000e+00 2.942e-18 8.566e-16 9.498e-16 4.964e-15

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.937e-18 8.637e-16 8.700e-16 8.573e-16 8.743e-16 9.802e-16
```

```r
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.000e+00 0.000e+00 2.727e-18 5.993e-16 6.901e-16 3.423e-15
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 2.015e-18 6.009e-16 6.047e-16 5.957e-16 6.068e-16 6.344e-16
```

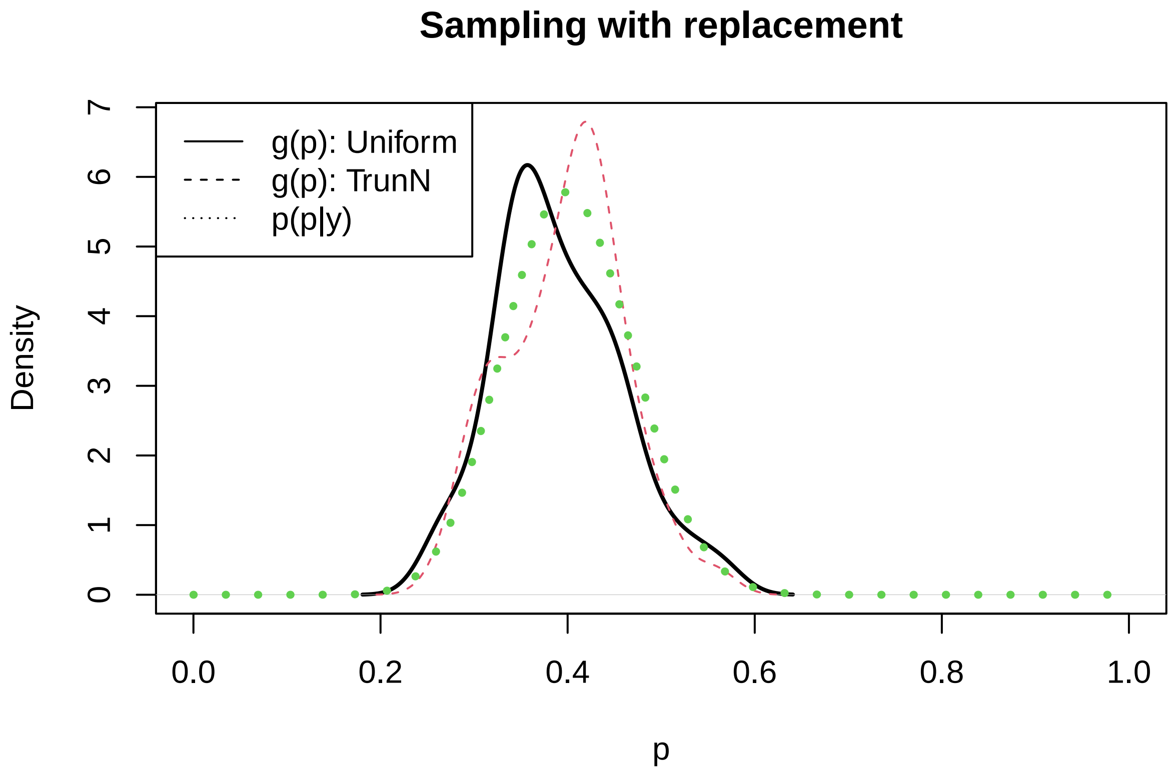
The second component of importance sampling is using the importance weights to re-sample, so that we can obtain a random sample from the posterior.

```r
#Importance re-sampling.
sam.1<-sample(x=x1,size=1000,prob=wp,replace=FALSE) #re-sample when g(p) = Uniform.
sam.2<-sample(x=x2,size=1000,prob=wp2,replace=FALSE) #re-sample when g(p) = Truncated Normal.
#Sampling without replacement.
d1<- dbeta((y+a-1)/(n+a+b-2),y+a,n-y+b)
plot(density(sam.1),ylim=c(0,d1+0.2),xlim=c(0,1),xlab='p',main='Sampling without replacement',lwd=2)
```

```
lines(density(sam.2),col=2,lty=2)
curve(dbeta(x,y+a,n-y+b),add=TRUE,col=3,lty=3,lwd=4)
legend(if(y> 2){'topleft'} else {'topright'},legend=c('g(p): Uniform','g(p): TrunN', 'p(p|y)'),lty=1:3)
```



```
#Sampling with replacement.
sam.1r<-sample(x=x1,size=100,prob=wp,replace=TRUE)
sam.2r<-sample(x=x2,size=100,prob=wp2,replace=TRUE)
plot(density(sam.1r),ylim=c(0,d1+1.0),xlim=c(0,1),xlab='p',main='Sampling with replacement',lwd=2)
lines(density(sam.2r),col=2,lty=2)
curve(dbeta(x,y+a,n-y+b),add=TRUE,col=3,lty=3,lwd=4)
legend(if(y> 2){'topleft'} else {'topright'},legend=c('g(p): Uniform', 'g(p): TrunN', 'p(p|y)'),lty=1:3
```


Sampling with replacement
g(p): Uniform
g(p): TrunN
p(p|y)
Density
0
1
2
3
4
5
6
7
0.0
0.2
0.4
0.6
0.8
1.0
p