Section 3: Random Variates.

Ensure you have completed the previous worksheet before advancing.

1 Inverse-Transform Technique

In class we discussed how, if $X$ is a continuous random variable with c.d.f. $F(X)$, then the random variable $R = F(X)$ is Uniform $(0, 1)$. For example, if $R \sim U(0, 1)$, then $X = -\log(R)/\lambda \sim \text{Exp}(\lambda)$.

Hence, to generate 1000 simulations from an exponential distribution with parameter $\lambda = 3$ we can do the following:

```r
> ru=runif(1000,0,1)
> re=-log(ru)/3
```

To check everything has worked we can do the following (have a read of the help files of any function you don’t understand):

```r
> hist(re,freq=FALSE,col="grey",breaks=50)
> curve(dexp(x,3),col=2,add=TRUE)
```

Of course we could simulate Exponential random variables in R directly using the `rexp` command, and the `dexp` command above evaluates $f(x)$ when $X \sim \text{Exp}(\lambda)$.

**Task:** Consider the triangle distribution with p.d.f. $f(x) = 2x$ for $0 \leq x \leq 1$. This has c.d.f. $F(x) = x^2$, and hence if $r = F(x)$, then $x = \sqrt{r}$. Knowing this, simulate 10,000 samples from this distribution and check the random variables do appear to follow the correct distribution (hint: the command `abline(a,b,col=2)` adds a line to a plot with intercept $a$ and slope $b$, or you can use `curve(f(x),col=2,add=TRUE)` if you replace $f(x)$ with the appropriate p.d.f.).

```r
>
>
>
>
```
2 Acceptance-Rejection Technique

N.B. We focus more on this in the next laboratory, once more details have been provided in lectures, so here we are only going to have a look at some illustrative examples.

To generate random variates $X \sim U(1/3, 1)$ we accept our random $R \sim U(0,1)$ if $R \geq 1/3$, if we want 1000 such random variables we are not sure how many such $R$ will be needed (though we can expect it to be near 1500).

In an R-script we can run the following (what does each line do?):

```r
k=0
count=0
ru2=c()

while(k<1000){
  test=runif(1,0,1)
  if(test>1/3){
    ru2=c(ru2,test)
    k=k+1
  }
  count=count+1
}
```

How many samples were required, and how many were wasted? We’ll look at efficiencies later.

**Task:** Task, generate a vector called `rtn` which contains 1,000 samples from a Truncated Normal Distribution with mean 0 and variance 1, with truncation $X > 0$, *i.e.*, $X$ would have the general p.d.f. of a Normal Distribution, but is not allowed to take values below 0 (hint: the `rnorm` function can be used to directly sample from the normal distribution).

> 

>
To check things have gone correctly do the following:

```r
> hist(rtn,freq=FALSE,col="grey",breaks=50)
> curve(2*dnorm(x),col=2,add=TRUE)
```

We discussed the method of simulating arrival times according to a Non-Stationary Poisson Process. Let's assume that arrival rates are 10, 5 and 15 for the first three hours respectively. Then to generate arrivals we can do the following using an R-script:

```r
times=c()
lambda=function(t){
  if(t<=1){ans=10}
  if(1<t){if(t<2){ans=5}}
  if(2<=t){ans=15}
  ans
}
lmax=15
t=rexp(1,lmax)
while(t<3){
  if(runif(1,0,1)<=(lambda(t)/lmax)){
    times=c(times,t)
  }
}
```
3 Direct Transformation

Note that, if $X \sim \Gamma(a, \theta)$, and $Y \sim \Gamma(b, \theta)$, then $X/(X + Y) \sim Beta(a, b)$. So to generate 10,000 samples from a $Beta(1, 2)$ we can do the following:

```r
> xlist=rgamma(10000,1,1)
> ylist=rgamma(10000,2,1)
> zlist=xlist/(xlist+ylist)
```

**Task:** Plot a histogram from the samples generated from the $Beta(1, 2)$ distribution and check against the p.d.f. for this distribution (hint: the p.d.f. for a Beta distribution can be obtained using the `dbeta` function).

```
> 
```

```
> Task: Recall that if $X_1, \ldots, X_n$ are independent standard normal random variables, then $X_1^2 + \ldots + X_n^2$ has a $\chi^2_n$ distribution. Knowing this, simulate 10,000 samples from a $\chi^2_3$ distribution, plot the histogram of your sample, and using the `dchisq` function check the accuracy of your results.

```