```
#################################################################

#Don't forget to use the "help" command,
#for example, at the command line, type
#
#>help(rexp)
#
#to find out what the "rexp" function does
#

################################################################
n <- 500
lambda <- 3
nits <- 5000
xmin <- rep(0,nits)
xmax <- rep(0,nits)
for(i in 1:nits){
        x <- rexp(n,lambda)
        xmin[i] <- min(x)
        xmax[i] <- max(x)
}
hist(xmin)
hist(xmax)
length(xmin[xmin < 0.0001])/nits
1-exp(-n*lambda*0.0001)
length(xmin[xmin > 0.0005])/nits
exp(-n*lambda*0.0005)
length(xmax[xmax <= 2])/nits
(1-exp(-lambda*2))^n
length(xmax[xmax > 3])/nits
1-(1-exp(-lambda*3))^n

################################################################

x <- c(0:5000)/500
cdf.x <- function(x,lambda){1-exp(-lambda*x)}
cdf.xmin <- function(x,lambda,n){1-(1-cdf.x(x,lambda))^n}
cdf.xmax <- function(x,lambda,n){(cdf.x(x,lambda))^n}

################################################################

n <- 500
theta <- 0.25
nits <- 5000
xmin <- rep(0,nits)
xmax <- rep(0,nits)
for(i in 1:nits){
        x <- rgeom(n,theta)
        xmin[i] <- min(x)
        xmax[i] <- max(x)
}
hist(xmin)
hist(xmax)

################################################################
n <- 500
nits <- 5000
umin <- rep(0,nits)
umax <- rep(0,nits)
for(i in 1:nits){
        u <- runif(n)
        y <- c(0,sort(u),1)
        udiff <- diff(y)
        umin[i] <- min(udiff)
        umax[i] <- max(udiff)
}
hist(umin)
hist(umax)

################################################################
```

```
#Read - or scan - in the data into a matrix (160 rows, 4 columns)
TMP.data<-matrix(scan("c:\\TEMP\\TMPproteinlengthdata.txt"),ncol=4,byrow=T)

#Remove column 3 as it is redundant
TMP.data<-TMP.data[,c(1,2,4)]

#Count the number of proteins
#Should be 160
TMP.nproteins<-nrow(TMP.data)
TMP.nproteins

#Column 1 contains the protein name or ID
#Column 2 contains the protein amino acid sequences
#Column 3 contains the secondary structure information

#For each protein, compute the sequence length and store the lengths in
#the vector TMP.proteinlengths.

TMP.proteinlengths<-numeric(length=TMP.nproteins)
for(i in 1:TMP.nproteins){
        s1<-TMP.data[i,2]
        TMP.proteinlengths[i]<-nchar(s1)
}
print(TMP.proteinlengths)

#Draw a histogram
hist(TMP.proteinlengths)

################################################################################
#
#Now repeat for the randomly selected proteins
#The format of this file is a little stranger ... it needs some more pre-
processing
#before the analysis can be performed

#Here are some string manipulation functions
#Highlight them all, the RUN.

strsplit <- function(x, split){
    if (length(x) > 1) {
        return(lapply(x, strsplit, split))
        }
    result <- character(0)
    if (nchar(x) == 0) return(result)
    posn <- regexpr(split, x)
    if (posn <= 0) return(x)
    c(result, substring(x, 1, posn - 1),
        Recall(substring(x, posn+1, nchar(x)), split))
}


strsub <- function(pattern, replacement, x){
        np <- nchar(pattern)
        nx <- nchar(x)
        posn <- regexpr(pattern, x)
        if (posn <= 0) return(x)
        result <- if (posn == 1) paste(replacement, substring(x, np + 1,
        nx), sep="")
        else paste(substring(x, 1, posn - 1), replacement, substring(x,
        posn + np, nx), sep="")
        result
}

gsub <- function(pattern, replacement, x){
    if (regexpr(pattern, x) <= 0) return(x)
    x <- strsub(pattern, replacement, x)
    Recall(pattern, replacement, x)
```

```
}

#Now start reading in the data

protein.data<-scan("c:\\TEMP\\proteinlengthdata.txt")

n<-length(protein.data)
index<-c(1:n)

start.lines<-index[substring(protein.data,1,1) == ">"]
end.lines<-c(start.lines,n+1)
end.lines<-end.lines[-1]-1

proteins.data<-matrix(" ",ncol=2,nrow=length(start.lines))
for(i in 1:length(start.lines)){

        v<-character()
        for(j in (start.lines[i]+1):end.lines[i]){
                tmp.v<-protein.data[j]
                v<-paste(v,tmp.v)
        }
        v<-gsub(" ","",v)
        proteins.data[i,1]<-gsub(">","",protein.data[start.lines[i]])
        proteins.data[i,2]<-v
}

#Now, for the proteins.data matrix
#Column 1 contains the protein name or ID
#Column 2 contains the protein amino acid sequences

protein.nproteins<-nrow(proteins.data)
print(protein.nproteins)

#For each protein, compute the sequence length and store the lengths in
#the vector TMP.proteinlengths.

protein.proteinlengths<-numeric(length=protein.nproteins)
for(i in 1:protein.nproteins){
        s1<-proteins.data[i,2]
        protein.proteinlengths[i]<-nchar(s1)
}
print(protein.proteinlengths)

#Draw a histogram
hist(protein.proteinlengths)

TMP.proteinlengths
mean(TMP.proteinlengths)
var(TMP.proteinlengths)
mean(protein.proteinlengths)
var(protein.proteinlengths)
```