Example code for simulation

This piece of code determines the variance of the median value of a sample of size 11 from a $\chi^2(9)$ distribution, and makes a histogram of the values. The square bracket indicates position in a vector. For example, `x[8]` is the 8th element of the vector `x`.

```
nloop = 100000   ## this is the number of samples of size 11
mval = 1:nloop   ## the values of the sample median are stored in this vector
for(iloop in 1:nloop){   ## beginning of loop
   x=rchisq(11,9)       ### the vector x has 11 independent chi-sq (9) values
   mval[iloop] = median(x)   ## the square brackets indicate the position
}         ## end loop
### now we have a vector mval of 100000 medians
var(mval)    ## this is the variance
hist(mval, br=100)   ## lots of breaks makes a smooth histogram
```

Code for computation of conditional probability

This code is used in the imperfect disease testing example of Chapter 3. The `if` statement is used, as well as the "double equals." In `R`, the equals sign is used as an assignment, while the double equals is evaluated as true or false. The piece of code `x==y` will return `TRUE` or `FALSE` depending on whether the values of `x` and `y` are the same. In an `if` statement, the code in the curly brackets gets executed when the code within the parentheses evaluates to `TRUE`.

```
nloop=100000
numerator=0;denominator=0
for(iloop in 1:nloop){
   disease=sample(0:1,1,prob=c(.999,.001))  ## sample a person
   if(disease==0){
      test=sample(0:1,1,prob=c(.98,.02))   ## if no disease
   }else{
      test=sample(0:1,1,prob=c(.01,.99))  ## if disease
   }
   if(test==1){
      denominator=denominator+1
      if(disease==1){ numerator=numerator+1}
   }
}
numerator/denominator
```

Code for simulating from triangular density

For the answer to Exercise 19.9, we use the square brackets to indicate positions in a vector. If `u` is a vector of one million values as defined below, then `u<=1/2` is a vector of one million values of `TRUE` and `FALSE`. The vector `y[u<=1/2]` contains those values of `y` for which the values of `u` are no larger than 1/2. In this piece of code, the vector `y` is defined, then filled in two steps according to the value of `u`.

```
u=runif(1000000)
y=1:1000000
y[u<=1/2]=sqrt(2*u[u<=1/2])
y[u>1/2]=2-sqrt(2*(1-u[u>1/2]))
hist(y,freq=FALSE,main="")
lines(c(0,1),c(0,1),lwd=3,col=2)
lines(c(1,2),c(1,0),lwd=3,col=2)
```

Code for making joint uniform figure

This code makes the figure shown in the Chapter 30 example determining the distribution of a sum of independent uniform random variables.

```
par(mar=c(0,0,0,0))
par(mfrow=c(1,1))
par(bty="n")
plot(c(-.12,1.02),c(-.12,1.02),pch="",xaxt="n",yaxt="n")
lines(c(-1,2),c(11/4,-1/4))
polygon(c(0,1,1,3/4,0),c(0,0,3/4,1,1),col="wheat1")

lines(c(-1,1),c(5/4,-3/4))
text(.2,.2,expression(u[1]+u[2]==y))
text(.34,.1,"(y in (0,1))",cex=.8)
polygon(c(0,1/4,0),c(0,0,1/4),col="wheat3")

text(.8,.8,expression(u[1]+u[2]==y))
text(.84,.71,"(y in (1,2))",cex=.8)

lines(c(-1,2),c(0,0))
lines(c(0,0),c(-1,2))
lines(c(0,0),c(0,1),lwd=3)
lines(c(1,1),c(0,1),lwd=3)
lines(c(0,1),c(0,0),lwd=3)
lines(c(0,1),c(1,1),lwd=3)
lines(c(1,1),c(-.03,.03))
text(1,-.08,1)
text(.5,-.11,expression(u[1]))
```

```
text(-.11,.5,expression(u[2]))
lines(c(-.03,.03),c(1,1))
text(-.08,1,1)
```

Code for making the cover figure

This code makes the figure shown on the cover of the book, and also as the first example in Chapter 24.

```
x=0:25*.88
y=0:25*.88
z=matrix(nrow=26,ncol=26)
for(i in 1:26){for(j in 1:26){z[i,j]=exp(-x[i]/5-y[j]/5)}}
zs=z
zs[x>=2,y>=0]=0;zs[x<=0,y>=2]=0

par(mar=c(1.6,0,0,0))

x10=c(-2,22);y10=x10;f0=matrix(0,nrow=2,ncol=2)
persp(x10,y10,f0,xlim=c(-2,20),ylim=c(-2,20),zlim=c(0,1),th=45,phi=20,shade=.1,box=FALSE
par(new=TRUE)

persp(x,y,z,theta=45,phi=20,xlim=c(-2,20),ylim=c(-2,20),zlim=c(0,1),tick="detailed",xlab

par(new=TRUE)

res=persp(x,y,z,theta=45,phi=20,xlim=c(-2,20),ylim=c(-2,20),zlim=c(0,1),tick="detailed",

lines(trans3d(c(2,2),c(2,2),c(0,exp(-4/5)),res))
lines(trans3d(c(1,1),c(2,2),c(0,exp(-3/5)),res))
lines(trans3d(c(1,1),c(1,1),c(0,exp(-2/5)),res))
lines(trans3d(c(2,2),c(1,1),c(0,exp(-3/5)),res))
lines(trans3d(c(1,2),c(1,1),c(0,0),res))
lines(trans3d(c(1,1),c(1,2),c(0,0),res))
lines(trans3d(c(1,2),c(2,2),c(0,0),res))
lines(trans3d(c(2,2),c(1,2),c(0,0),res))
xp=0:10/10+1;yp=xp
lines(trans3d(xp,yp*0+1,exp(-1/5-xp/5),res),lwd=3)
lines(trans3d(xp,yp*0+2,exp(-2/5-xp/5),res),lwd=3)
lines(trans3d(xp*0+2,yp,exp(-2/5-xp/5),res),lwd=3)
lines(trans3d(xp*0+1,yp,exp(-1/5-xp/5),res),lwd=3)
text(trans3d(3,1,0,res),"A")

lines(trans3d(c(0,30),c(0,0),c(0,0),res))
lines(trans3d(c(0,0),c(0,30),c(0,0),res))
lines(trans3d(c(0,0),c(0,0),c(0,1),res))
```

```
points(trans3d(0,0,0,res),pch=20,cex=1.2)
text(trans3d(-1.2,-1.2,0,res),"(0,0)")
text(trans3d(0,30,0,res),"^",srt=-65)
text(trans3d(23.45,0,0,res),"^",srt=-125)

text(trans3d(22.3,-.9,0,res),expression(y[1]))
text(trans3d(-2,24,0,res),expression(y[2]))
```