R commands 2

This is a continuation of the earlier *R* document with commands to help answer questions from weeks 11-20. Some questions can ba answered partially or wholly using the types of commands given before, and so are not included.

To use a Markov chain model enter the library *msm* with the command

```
library(msm)
```

To generate a matrix of transition rates (in this case a 2×2 matrix with elements -0.1,0.1 in the first row and 0.3,0.-3 in the second, type

```
qmatrix <- rbind(c(-0.1, 0.1), c(0.3, -0.3))
```

To simulate this process up to time 15 type

```
sim.msm(qmatrix,15)
```

It is possible to estimate a $q - matrix$, in this case a 3×3 one where all transitions are possible by typing

```
qmatrix <- rbind(c(0, 1, 1), c(1, 0, 1), c(1, 1, 0))
```

which gives you the matrix. You will then need to enter a vector if initial values for the transition rates, and a series of observations of the process which gives both time of observation and state of the process.
Use the command

```
inits<-c(...)
```

to give the initial values (enter a vector with 6 elements).

```
states<-c(...)
```

gives the vector of states, and
`times<-c(0, ...)` the vector of times. The command

```
msm(formula=states times, qmatrix=qmatrix, inits=inits)
```

then estimates the transition rates.

To enter the aneurism data, type

```
data(aneur)
```

You will need a 4×4 transition matrix, where only transitions to the next most severe category are possible. To then perform the analysis, you will need to enter the commands (or similar)

```
msm(data=aneur, fromto=TRUE, fromstate=from, tostate=to, qmatrix=qmat, timelag=dt,
inits=c(0.001, 0.03, 0.3), method=”BFGS”, control=list(trace=2))
```

To investigate the Woodmouse data go into the ”ape” library, and type

```
data(woodmouse)
str(woodmouse)
base.freq(woodmouse)
```

To manipulate matrices, use the command ”mat.or.vec”.
To generate a matrix with 6 rows and 6 columns (all values 0), enter

```
x<-mat.or.vec(6,6).
y<-mat.or.vec(6,1)
```

gives a vector with 6 elements.

```
y1=y+1
```

adds 1 to all the elements of y,
y2<-diag(y1) creates a matrix 6 × 6 matrix, with the elements of $y1$ down the leading diagonal.

$x3 = x1\% * \%x2$

multiplies the matrix $x2$ by the matrix $x1$.

To perform bootstrap routines, enter the library ”simpleboot”.
You already know how to simulate data. To use a bootstrap routine to obtain a 100 samples and give the means of each from the data x , use the command

```
b.mean <- one.boot(x, mean, 100)
```

To obtain a confidence interval for the mean, using these samples, type

```
boot.ci(b.mean)
```

Similarly for two data sets x and y, 100 pairs of data sets (one from each) can be found, as well as the difference between their means, using

```
b <- two.boot(x, y, mean, R =1000)
```

To perform genetic analysis, enter the library ”genetics”. A vector of genotypes can be entered as follows. If there are 2 alleles (A and B), there are three genotypes; if the numbers of each are 13, 27, 24 then the vector is created by

```
g<-c(rep(”A/A”, 13 ),rep(”A/B”,27 ),rep(”B/B”, 24))
```

and then

```
g1<-genotype(g)
```

We can test for Hardy-Weinberg equilibrium using

```
HWE.chisq(g1,simulate.p.value=FALSE)
```

for the chi-square test, and

```
HWE.exact(g1)
```

for the exact test. We can estimate the degree of disequilibrium with

```
v<-diseq.ci(g1,conf=0.95)
```

To analyse phenotypic data, use the library ”ape”. To look at the bird.order data type

```
data(bird.orders)
```

You can check if the tree is ultrametric or binary using

```
is.ultrametric(bird.orders)
```

and

```
is.binary.tree(bird.orders)
```

and obtain a plot of the tree using

```
plot.phylo(bird.orders)
```

A random tree with 10 species can be generated by

```
t1<-rtree(10)
```

and converted to an ultrametric tree with

```
t2<-chronogram(t1)
```

The distances between the species can be found using

```
dist.phylo(t2)
```

To obtain estimates of the distance matrix of the Woodmouse data, using a Jukes-Cantor model, type

```
dist.dna(woodmouse,method=”JukesCantor”)
```