R Programming: Worksheet 4

By the end of today you should be able to plot one and two-dimensional data; generate plots from scratch; add points, lines, blocks and legends;

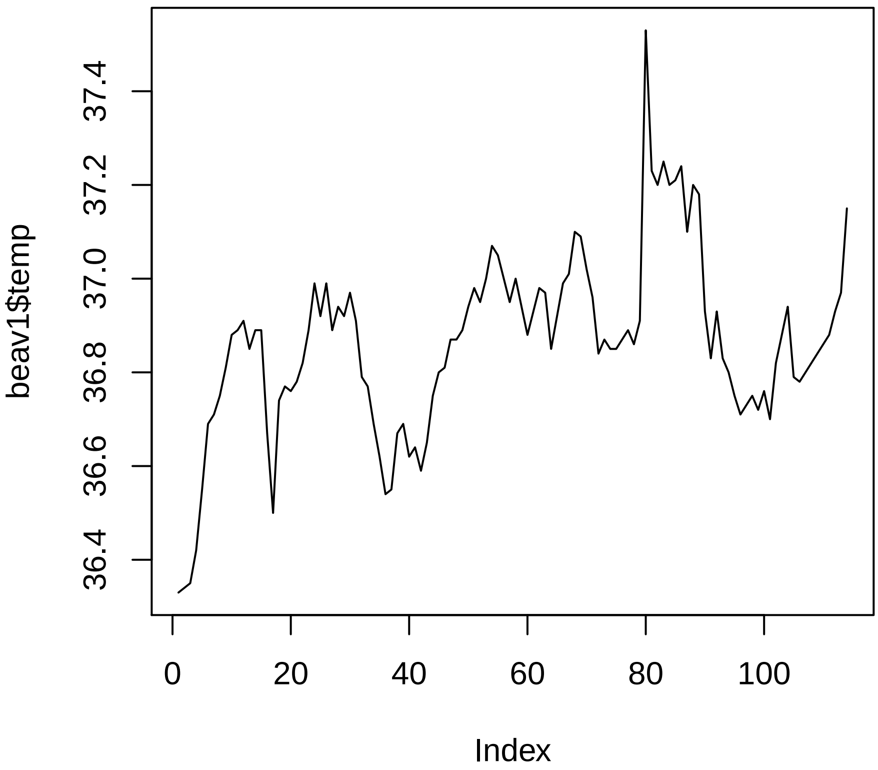
1. **Time Series**

   Load the data `beav1` from the `MASS` package, and look at the field `temp` (ignore the other fields for now).

   (a) Using the function `ma3()` which you constructed on Sheet 3, calculate the moving average of length 3 for the `temp` data.

   (b) Plot the original temperature data [use `plot()` with the option `type="l"`, as we did for the random walk on sheet 1.]

   ```
   > library(MASS)
   > data(beav1)
   > plot(beav1$temp, type = "l")
   ```



   (c) Try plotting the moving average of length 3 on the same graph. First plot the original data as usual. Then, use the `points()` function to add on the second vector:

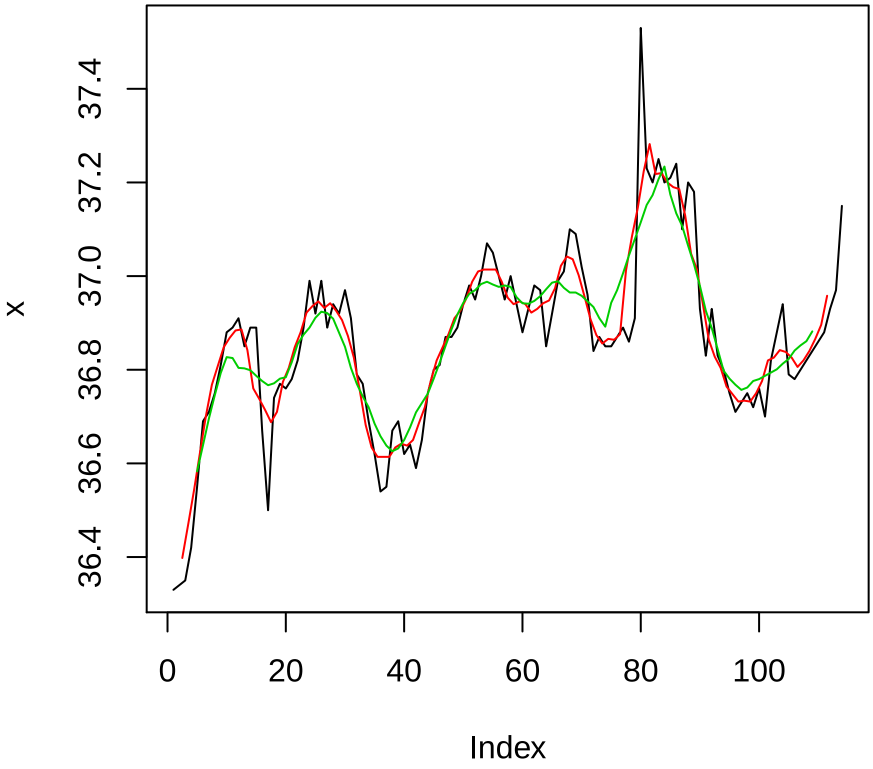
```
> n <- length(beav1$temp)
> points(2:(n - 1), ma3(beav1$temp), type = "l", col = 2)
```

(d) Write a second function `plotMA` which takes two vector arguments, `x` and `k`, and (i) calculates the moving averages of length `k` for each entry in `k`; (ii) plots them on top of each other as above.

Ideally, each line should be a different colour.

Using `ma()` the function from Worksheet 3:

```
> library(MASS)
> plotMA = function(x, k) {
+     plot(x, type = "l")
+     n = length(x)
+     k = k[k != 1]
+     for (i in seq_along(k)) {
+         tmp = ma(x, k[i])
+         points(seq(from = k[i]/2, length = n - k[i] + 1,
+             by = 1), tmp, type = "l", col = i + 1)
+     }
+ }
> plotMA(beav1$temp, c(5, 10))
```



2. **ChickWeight Data**

Take a look at the ChickWeight data.

```
> ?ChickWeight
```

Do some exploration of the different fields and their qualities. For parts (a)-(e), use the base R function only (i.e. without loading the lattice package). If you want to unattach a package, use the following command:

```
> detach(package:lattice)
```

(a) Do a basic scatter plot of the logarithm of chick-weight against age. What are the limitations of this? *Either of:*

```
> with(ChickWeight, plot(Time, log(weight)))
> plot(ChickWeight$Time, log(ChickWeight$weight))
```

You can't see which measurements are on the same chicks, nor which treatment they received.

(b) Use colour to distinguish between the different diets.

```
> with(ChickWeight, plot(Time, log(weight), col = Diet))
```

(c) What is the range of log-weights and ages? [I'll let you guess the command for this one.]

```
> range(ChickWeight$Time)
> range(log(ChickWeight$weight))
```

(d) Using this, construct a new (blank) plot window with these ranges, and add x- and y-axes.

```
> with(ChickWeight, {
+     plot.new()
+     plot.window(xlim = range(Time), ylim = range(log(weight)))
+     axis(side = 1)
+     axis(side = 2)
+ })
```

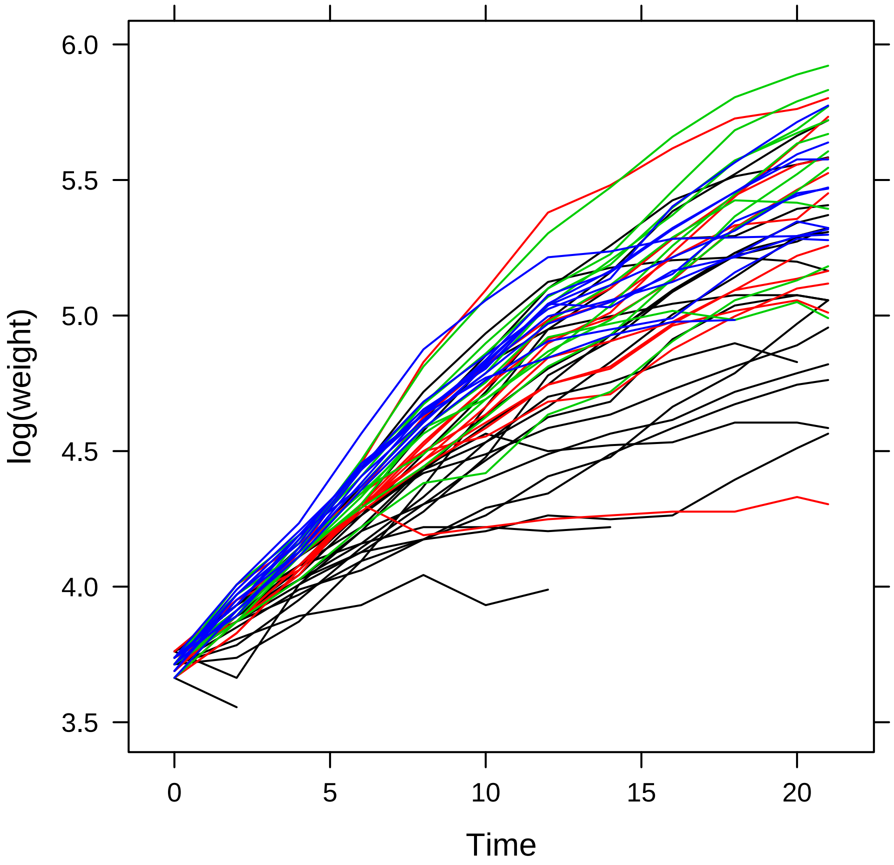
(e) Now plot an individual line for each chick's measurements, with colour corresponding to the diet they were given.

This plot should look much like the lattice one below.

Do you see any further shortcomings in this plot? *The chicks are ordered by diet, so in the plot all the chicks on diet 4 are shown on top of the rest; this draws the eye unfairly. A solution would be to use* `sample()` *and add them in a random order.*

(f) Can you achieve the same plot using lattice? *Getting the colours corresponding to the diet proved a bit of a problem; this was the best I could manage:*

```
> library(lattice)
> xyplot(log(weight) ~ Time, groups = Chick, col = rep(c(1,
+     1, 2:4), each = 10), data = ChickWeight, type = "l")
```



(g) Export your final plot to a PDF file.

3. **Lynxes and Hares**

   I have sent you a file called `hares.dat`; it contains data on the annual number of lynx and hare pelts recovered by the Hudson's Bay Company in Canada. Save the file somewhere appropriate and read it into `R` as a data frame.

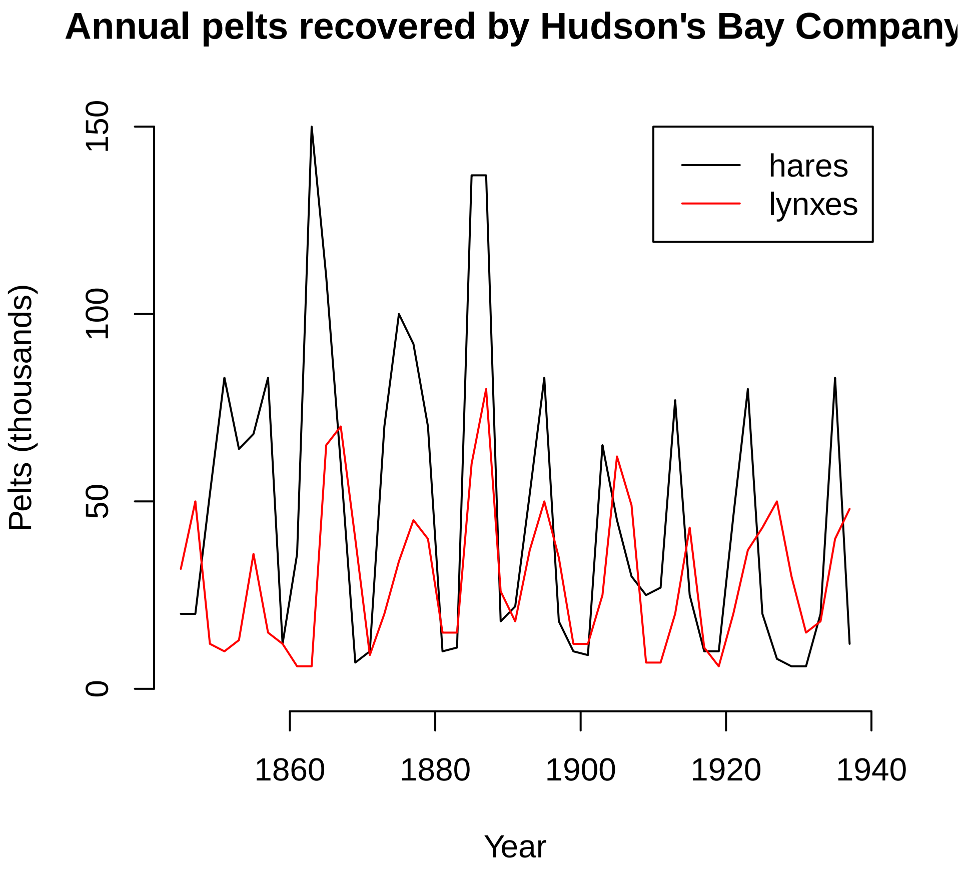
   Produce a single plot of these data, consisting of a line for each species (which should be different colours). Make sure that:

   (a) the y-axis starts at 0.

   (b) both axes are correctly labelled and your plot has a title;

   (c) you provide a legend to label your series.

```
> dat = read.table("hares.dat", header=TRUE)
>
> plot.new()
> plot.window(xlim=range(dat$year), ylim=c(0,max(dat[,2:3])))
> axis(side=1)
> axis(side=2)
> title(xlab="Year", ylab="Pelts (thousands)",
+          main="Annual pelts recovered by Hudson's Bay Company")
> points(dat$year, dat$hare, type="l")
> points(dat$year, dat$lynx, type="l", col=2)
> legend(1910,150,legend=c("hares", "lynxes"),col=1:2,lwd=c(1,1))
```



4. **Binning Data***

   Write (from scratch) your own function to produce histograms, using `cut()` and `table()`.