CSC 120 (R Section)— Lab Exercise 1

This is a non-credit excercise, which you do not hand in.
You may work own your own or together with another student, as you please.

For this exercise, I've put data on the course web page about deaths each day in Toronto in 1993–1995 and the average temperature each of those days. We can look at whether number of deaths seems to be related to the temperature, and whether it's also related to the season, or perhaps only to the season (not temperature).

You can read in the two sets of numbers (deaths and temperature) in the way we've done before:

```
deaths <- scan("http://www.cs.utoronto.ca/~radford/csc120/deaths.txt")
temp <- scan("http://www.cs.utoronto.ca/~radford/csc120/temp.txt")
```

You'll also need to create a vector with numbers indicating the day of the year (from 1 to 365, since none of these years are leap-years). You can do this using the ":" operator, which produces a vector of consecutive integer — eg, 1:10 gives numbers 1 through 10 — and the "rep" operator that repeats a vector some number of times — eg, rep(c(3,4),2) gives the same vector as c(3,4,3,4).

You can start by reading the data and looking at it (perhaps plotting it) just to be sure you know what it's like. But you should then try to write an R Script that will produce four plots:

- Deaths for each day, with the horizontal axis being just numbers from 1 up. You can get a plot like this with just plot(deaths).
- Deaths on the vertical axis versus temperature on the horizontal axis. You can get a plot like this with plot(temp,deaths).
- Temperature versus day of year.
- Deaths versus day of year.

To make all four plots visible at once, you can use the command par(mfrow=c(2,2)). Each plot will then go into the next slot in a two-by-two array of plots.

Once you've written a script that does this, you can try to modify it so that the points in the plots other than the first show what year they are for by colour — red for 1993, green for 1994, and blue for 1995. To do this, you can use the "col" option to plot, which can be a vector, giving the colour of each point plotted.

You can then write a second script, in which you try to predict the number of deaths based on the day of the year (ie, the season). You should try a prediction equation of the following form:

```
prediction <- a + b*cos((day_of_year-s)*2*pi/365)
```

You'll need to set the variables a, b, and s before doing this. Play around to see what values seem to be best. One hint: It's probably a good idea for mean(prediction) to be about the same as mean(deaths).

You can test your predictions by plotting deaths versus day of the year, and then plotting your predictions on top of this plot. You can add points to a plot using the "points" function.

You can put the commands for setting a, b, and s, making the predictions, and plotting the data and the prediction in an R Script, which you can easily run again and again with different values for a, b, and s.

Once you've found what seem to be good values for a, b, and s, you can try seeing whether the error in this prediction is related to temperature. Since the prediciton is based only on the season, not the actual temperature, this might reveal whether temperature itself is related to deaths, or whether it's just that deaths are related to the season, and temperature is related to the season.

(Note: To draw any firm conclusions about the real relationship of temperature and death, you'd want to look at more than three years of data, in more places than Toronto. This is just an exercise!)