In these labs you should work through the problems on the sheet and type your answers into a Microsoft Word document. When you have completed the tasks, print the document and hand it in.

Your answers can benefit from having small sections of R output and graphs copied and pasted into your word document. It can be useful to limit the width of output produced by R so that it fits into your document. You can change the width of output produced by R with a command like: `options(width=50)`.

The labs will count for 1 mark in the current assignment. They will be graded on a 0-1 basis. Don't forget to put your name and student ID on the document.

The basic bar charts produced by R are very plain. In this lab we'll learn some tricks which will allow us to draw more "interesting" ones. Essentially, we throw away the R barplot code and draw things from first principles. We'll use the "meat" data from class

```
y = c(41, 8, 10, 16, 25)
names(y) = c("Beef", "Lamb", "Mutton", "Pigmeat", "Poultry")
```

However, you should write your code so that it will work on any data set. E.g. use `length(y)` rather than `5` when specifying the number of data values.

We'll assume that we have a set of values giving the bar heights in a vector called `y`. We'll also assume that the bars are to be drawn centred on 1, 2, . . . , `length(y)` and have a width of

```
w = 0.7
```

Setting Up The Plot Region

To draw a basic bar plot you should will need to start a new plot and set up the plotting region. This can be done as follows:

```
x = 1:length(y)
xlim = range(x) + c(-0.5, 0.5)
ylim = range(pretty(c(y, 0)))
plot.new()
plot.window(xlim = xlim, ylim = ylim, xaxs = "i", yaxs = "i")
```

The `pretty` function computes a "pretty" set of values which cover the range its argument (try it!).

Drawing the Bars

Once the plot region has been set up, you can draw the bars with the `rect` function. The basic function call is of the form

```
rect(x - w/2, 0, x + w/2, y, col = "lightblue")
```

but you can choose a better colour.

Annotation

The basic plot is now done, but you probably want to add a y axis, labels under the bars, etc.

```
mtext(names(y), side = 1, at = x, line = 0.5)
axis(2, las = 1)
box()
```

You can also use the `title` function to add a main title and a label for the y axis.

```
title(main=list("New Zealand Meat Consumption", cex = 2),
      ylab = "Percentage of Meat Consumed")
```

Background Grids

You can add a set of background grid lines to the plot using `abline`. You could specify exacly where the lines are to go, or you could use `pretty` to compute the locations. For example,

```
abline(h = pretty(ylim, 10), lty = "dotted")
```

will produce approximately 10 grid lines, located at “pretty” y values.

Colouring the Background

It is common to see coloured backgrounds in presentation graphics. You can do this by adding some code between the setup of the plot and the drawing of the bars. To do this you need to find out the coordinates of the full *plot region* and to colour it with `rect`.

```
usr = par("usr")
rect(usr[1], usr[3], usr[2], usr[4], col = "wheat")
```

Gradient Backgrounds

You will see many plots produced with a *colour gradient* background. This can be produced in a very similar way to the colour fill above. This is done by drawing a sequence of rectangles, each a slightly different colour from the previous one.

```
usr = par("usr")
rect(usr[1],
     seq(ylim[1], ylim[2], length = ngrad),
     usr[2],
     ylim[2],
     col = hsv(1/2,
               seq(0.25, 0, length = ngrad),
               seq(0.85, 1, length = ngrad)), border = NA)
```