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.

This lab looks at the basic use of trellis displays. To produce these displays you will first need to issue the R command:

```
library(lattice)
```

to make the trellis functions and data sets available.

You should also change the lattice options so that the colours you use are suitable for printing on paper. You can do this with the command

```
lset(col.whitebg())
```

although you may wany to see how things look with a gray background too.

Engine Pollution

The R data set `ethanol` contains data on tests of a single cylinder engine to investigate how the amount of nitrous oxides (NOx) produced by the engine depend on how the engine is tuned. The variables in the data set are:

1. `NOx` which gives the concentration of nitric oxide (NO) and nitrogen dioxide (NO_2) in engine exhaust, normalised by the work done by the engine.
2. `C` which gives the compression ratio of the engine.
3. `E` which gives the equivalence ratio at which the engine was run – a measure of the richness of the air/ethanol mix.

Begin by producing a plot of `NOx` against `C` and against `E`. You can produce first plot with command

```
xyplot(NOx ~ C, data = ethanol)
```

but you add better axis labelling and an overall title.

Now try some multipanel conditioning plots. First try conditioning on the equivalence ratio with the commands:

```
EE = equal.count(ethanol$E, number=9, overlap=1/4)
print(xyplot(NOx ~ C | EE, data = ethanol))
```

and then try conditioning on the compression ratio. Try producing and interpretting the graphs produced by lowess fits

```
xyplot(NOx ~ C | EE, data = ethanol,
       panel = function(x, y) {
             panel.xyplot(x, y)
             llines(lowess(x,y))
})
```

See if you can eliminate the points and just plot the smooth lines.

Death Rates

Run through the death rates examples from lectures. You can do this with the command

```
data(VADeaths)
```

Now set things up in form for trellis with the following commands.

```
rate = as.vector(VADeaths)
age = row(VADeaths, as.factor=TRUE)
group = col(VADeaths, as.factor=TRUE)
```

Try varying the number of rows and columns in the layout. (Note that the lecture notes are on line).