Department of Statistics

COURSE STATS 330/762

Model Answers for Assignment 1, 2010

Question 1.

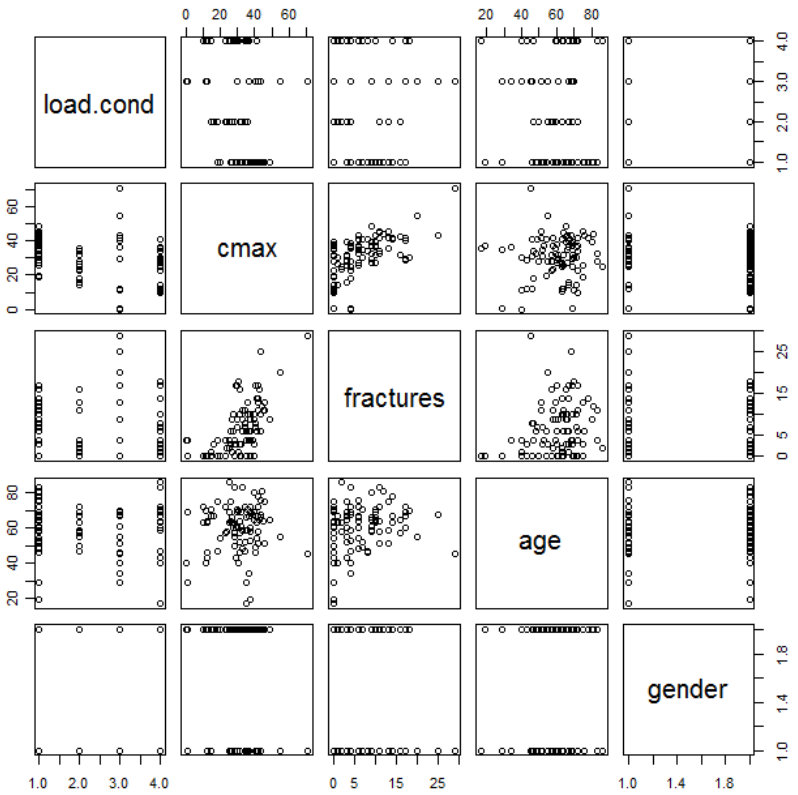
1. *Load the data into R, and make a data frame **kent.df** to contain the data. Check for any typographical errors (the data below may be taken to be the correct data, but the data on the web may have been corrupted). Print out the last 10 lines of the data file. [5 marks]*

The following code will load in the data:

```
kent.df = read.table("kent.df=read.table(
"http://www.stat.auckland.ac.nz/~lee/330/datasets.dir/kent.txt",
 header=TRUE)
```

A good way to identify data errors is a pairs plot of the data frame. We can probably ignore testid (the sixth variable):

```
pairs(kent.df[,-6])  # note how we delete the 6th column
```



There are no obvious outliers in this plot. The large value of cmax is a legitimate value. In large data sets it is not practical to proofread the input data. Instead, we usually rely on range checks (are vales too big or small to be real) or plots like the one above. We will assume that there are no bad values here.

To print out the last 10 lines, note that there are 93 lines in the file, so we want lines 84 through 93:

```
   load.cond  cmax fractures age gender testid
84      dist 36.60         4  34      f    388
85      dist 43.70        25  68      f    421
86      dist 40.60        17  67      f    422
87      dist 42.20        13  51      f    423
88      dist 55.20        20  55      f    424
89      dist  0.35         4  69      m    116
90      dist  0.35         0  29      f    143
91      dist  0.00         4  40      m    650
92      dist 11.00         0  70      m    651
93      dist 12.00         0  46      m    652
```

2. *What is the relationship between the number of fractures and cmax? Does the relationship depend on the test condition, age and gender? If so, how? Draw suitable plots to answer this question. Don't try and fit any models. [8 marks, 4 for the plots and 4 for the discussion]*

When we want to see how the relationship between two variables changes as the values of other variables change, we use coplots. In this case, there are 3 variables to condition on. Unfortunately, this leads to some panels having very few points in them. Thus, we will condition on the remaining variables one at a time. Conditioning on test.condition, we get

```
xyplot(fractures~cmax|load.cond, data=kent.df)
```

(Picture overleaf). Note that we have used xyplot for each panel since both the number of fractures and cmax are numeric variables.

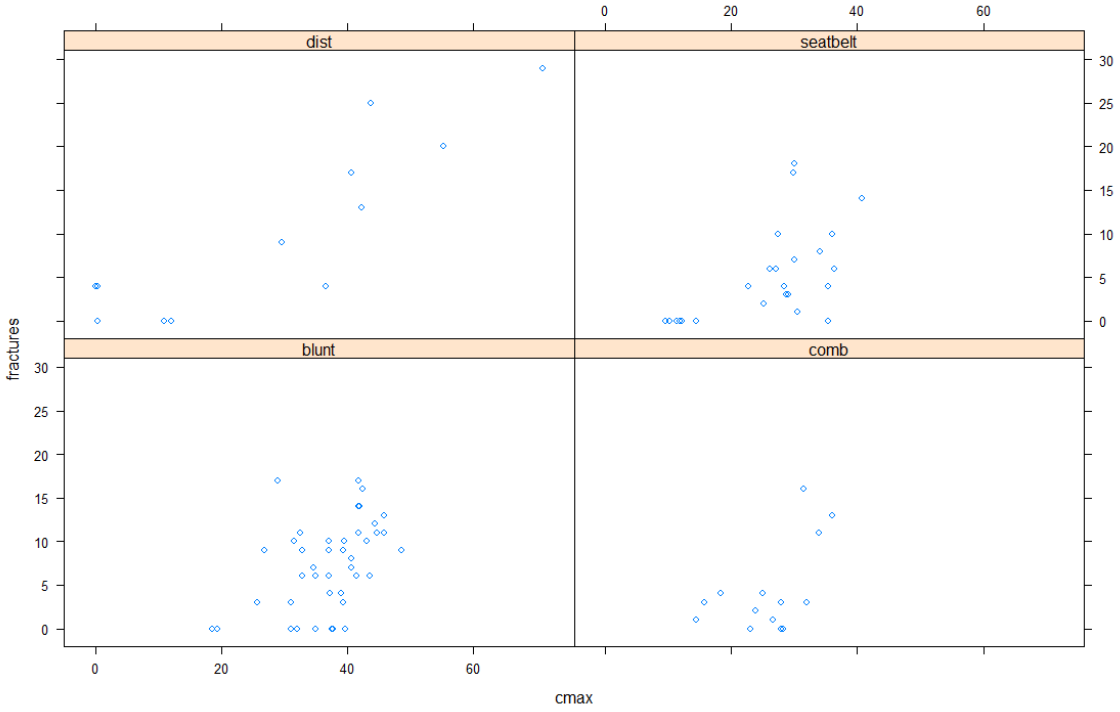
The picture shows that for the distributed loading condition, the relationship is a bit stronger than for the other 3 conditions, with possibly a hint of a curved relationship. The distributed loading condition seems to have higher values for cmax and fractures than the other conditions.

To condition on age (say 4 age groups) we type
```
xyplot(fractures~cmax|equal.count(age, number=4), data=kent.df)
```

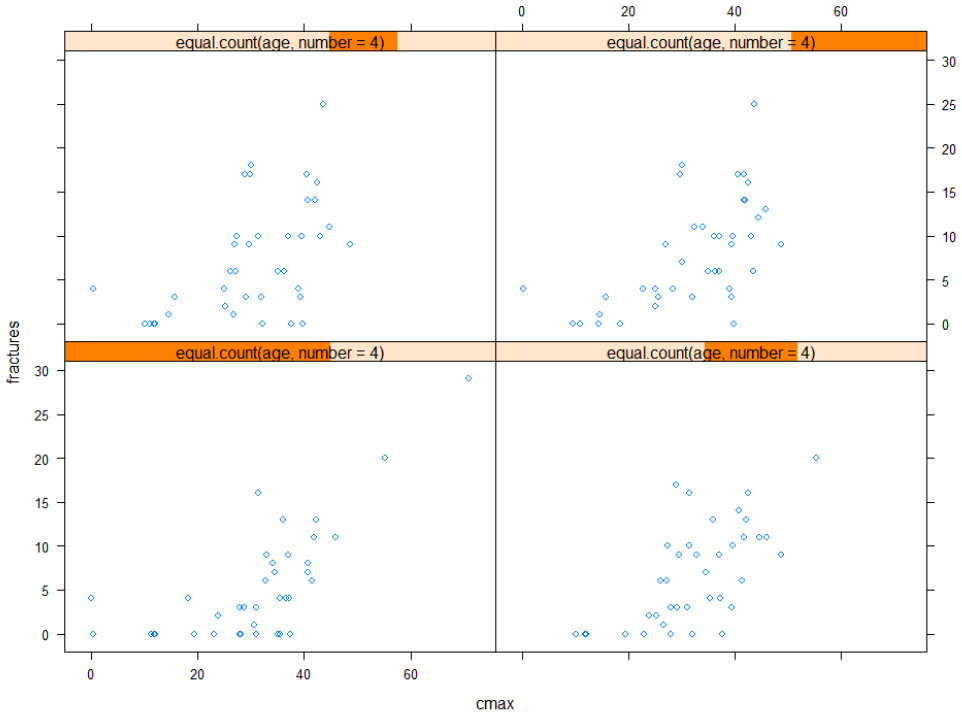
This gives the second plot overleaf. We see that the relationships are similar for the different age groups, although there are fewer points in the top right corner of the plots for the younger age groups, suggesting that both the number of fractures and cmax tend to increase with age.

```
xyplot(fractures~cmax|load.cond, data=kent.df)
```

```
xyplot(fractures~cmax|equal.count(age, number=4), data=kent.df)
```

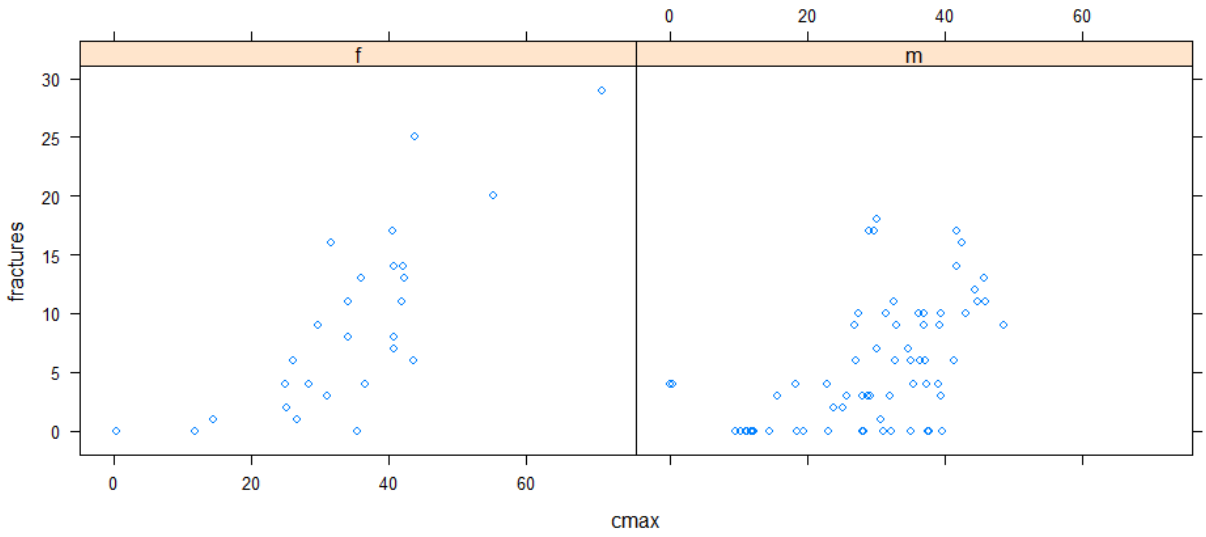



```
xyplot(fractures~cmax|equal.count(age, number=4), data=kent.df)
```



Finally, we condition on gender:

```
xyplot(fractures~cmax|gender, data=kent.df)
```

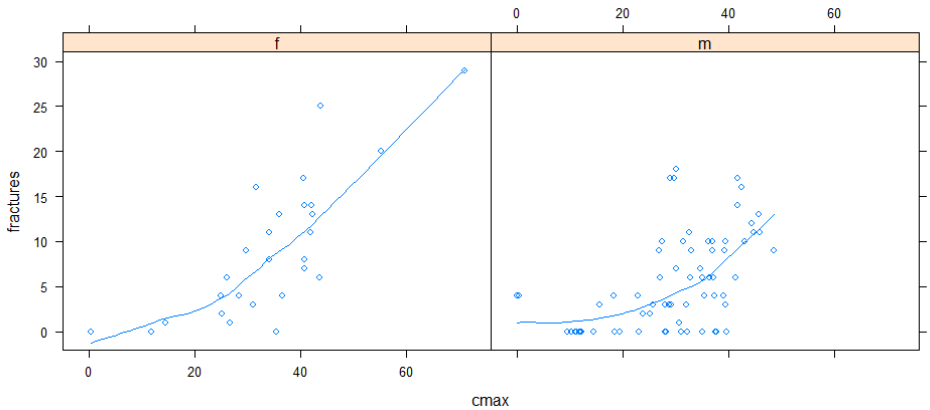


The relationship seems a bit stronger for females, and seems to have a steeper slope. Clearly the females in this sample tended to have higher values for both cmax and fractures (see the next part).

We can do plots conditioning on two factors at a time but again these are hard to interpret, with too few points in some panels. Not worth reporting.

Adding loess lines to the plots can aid interpretation. We can do this using "panel functions", which are described in more detail in the Trellis manual on the website:

```
xyplot(fractures~cmax|equal.count(age,4)*gender, data=kent.df,
panel = function(x,y){
panel.xyplot(x,y)
panel.loess(x,y)
})
```



This helps us see the difference in relationships alluded to above.

To summarise, the main findings are:

- The relationship between fractures and cmax is different for the distributed loading condition, the relationship is a bit stronger than for the other 3 conditions, with possibly a hint of a curved relationship. The values for cmax and fractures seem a bit higher for the distributed loading condition.

- The relationships are similar for the different age groups, although there is a suggestion that both the number of fractures and cmax tend to increase with age.

- The relationship seems a bit stronger for females, and seems to have a steeper slope. The females in this sample tended to have higher values for both cmax and fractures.

3. *The researchers were interested in whether the injury was "severe", defined as more than six rib fractures. Does the proportion of severe injuries depend on cmax and gender? How? Again, do not fit any models to answer this, but draw a suitable plot or plots. Hint: divide the range of cmax into suitable intervals, and plot the proportion of severe injuries falling in each cell of the resulting cmax-gender two-way table. The function* ***barplot*** *will be useful. [5 marks]*

To answer this question, we need to create a variable corresponding to different cmax ranges, then calculate the proportion of severe fractures for each cmax-group x gender combination. We will do this using non-trellis graphics. To make the age groups, we can't use the function equal.count, as this is a trellis function. Instead we use the function cut, which turns a numeric variable into a factor if we give it a set of "cut points" representing the class boundaries.

The cmax values range from 0 to 71. Lets take the cut points to be the minimum, the lower quartile, the median , the upper quartile and the maximum:

```
> cut.points = quantile(kent.df$cmax, c(0, .25,.5,.75,1))
> cut.points
  0%  25%  50%  75% 100%
 0.0   25.7 32.1 39.4 71.0
```

To make a factor cmax.factor containing the cmax groups:

```
cmax.factor = cut(cmax,cut.points)
```

To calculate the proportion of severe fractures, we define a logical variable severe having value TRUE if the fracture is severe and FALSE otherwise.

```
severe = kent.df$fractures>6
```

Calculating proportions is the same as taking the mean of a logical variable:

```
> mean(severe)
[1] 0.4193548
```

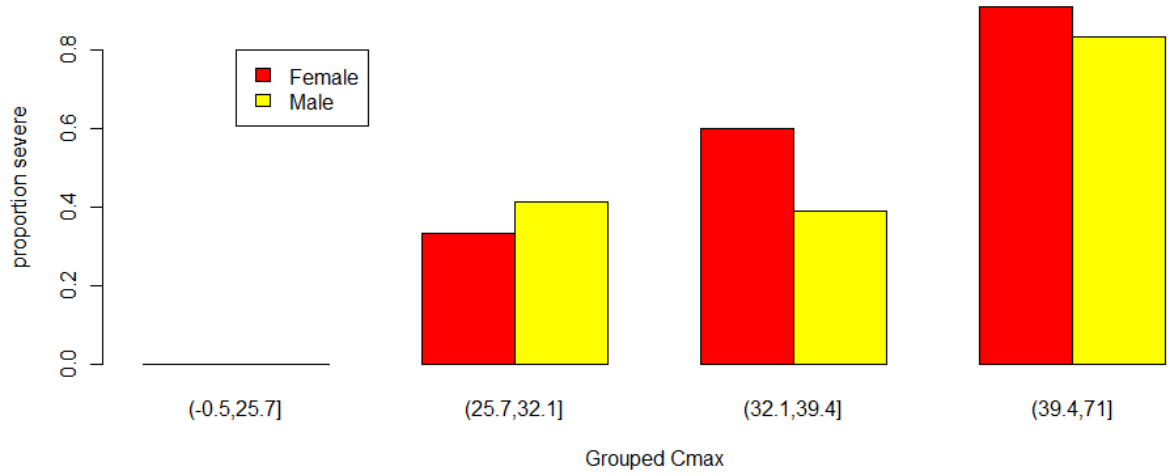
To calculate the proportion severe for cmax-groupx gender combination, we need to calculate the mean of the variable severe for each combination. The function tapply is useful for doing this:

```
> prop.severe = tapply(severe, list(cmax.factor,kent.df$gender), mean)
> prop.severe
                    f         m
(-0.5,25.7] 0.0000000 0.0000000
(25.7,32.1] 0.3333333 0.4117647
(32.1,39.4] 0.6000000 0.3888889
(39.4,71]   0.9090909 0.8333333
```

To graph these numbers with a barchart, we feed the transpose of this matrix to the barplot function:

```
> barplot(t(prop.severe), beside = TRUE, xlab = "Grouped Cmax",
+ ylab = "proportion severe", col=c("red", "yellow"))
> legend(2,0.8, legend = c("Female","Male"), col=c("red", "yellow"))
```

This produces the plot



Thus, the proportion of severe fractures goes up as cmax goes up, and is higher for females than males for high values of cmax.

4. *Are there any features of the data that might make fitting a regression model difficult? [2 marks]*

The fact that the trends in the coplot with age are not very straight suggests there might be some difficulties.

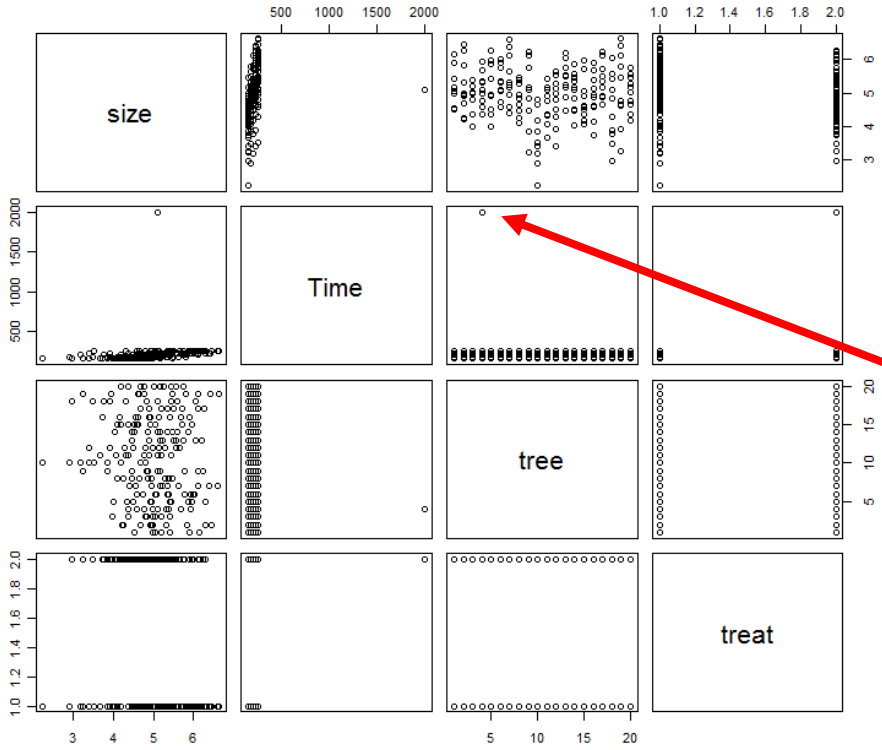
More information about this question may be found in the article by Kent and Patrie, a copy of which is available on the web.

Question 2

1. *Load the data into R, and make a data frame **sitka.df** to contain the data. Check for any typographical errors (the data below may be taken to be the correct data, but the data on the web may have been corrupted). Print out the last 5 lines of the data file, for each treatment. [5 marks]*

   The following code reads in the data and draws a pairs plot:

```
sitka.df = read.table(file.choose(), header=TRUE, sep=",")
pairs(sitka.df)
```



   Looks like a big time outlier of about 2000: inspection of the data file identifies this as the 18th line in the file: from the assignment sheet it should be 201. To correct it, type

   sitka.df$Time[18]=201

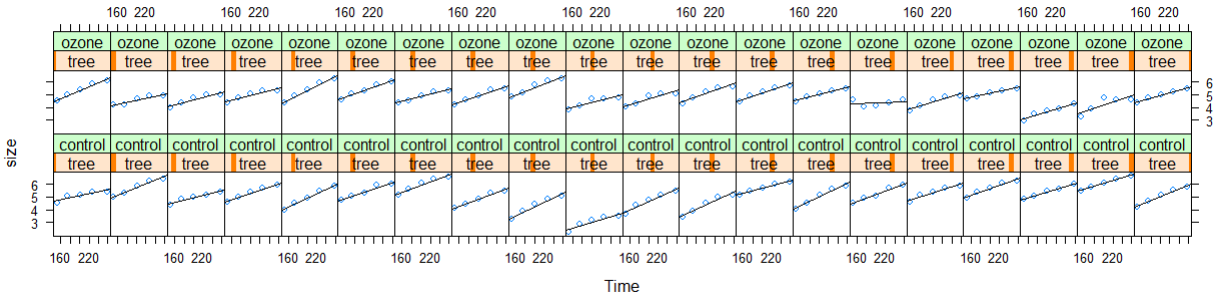
   Print out the last 5 lines of the file for each treatment

```
sitka.df[96:100,]   # ozone
sitka.df[196:200,] # control
```

2. *Draw a Trellis plot showing the growth of each tree in the two groups. Does ozone have an effect on growth? Are there any trees whose growth is different from the others in the group? [7 marks]*

To draw the trellis plot, we can type (note this includes a least squares line in each panel)

```
> xyplot(size~Time|tree*treat, data=sitka.df, panel = function(x,y){
+ panel.xyplot(x,y)
+ panel.lmline(x,y)
+ })
```



This shows that the growth curves are reasonably similar, but there are some differences in slopes: for example the 14th tree in the ozone group has a rather flat growth curve, and trees 1, 5 have quite steep curves. The trees are not all the same size at the beginning of the period, with the 10th tree in the control group being quite small. It is not easy to compare the two groups in terms of size, the next plot is better for this.

3. *Draw a plot similar to the one shown in class for the rat data, that shows the two groups on the same plot. Add lines to the plot representing the average size for the group at each time point. What do you conclude? [8 marks]*

One problem is that the tree labels repeat, so we can't distinguish the groups by this variable alone. Lets make a new tree label:

```
newtree = rep(1:40, rep(5,40))  # see the help file for how rep works
```

Then we can draw like the rat example

```
plot(sitka.df$Time,sitka.df$size, xlab="Time", ylab = "Size",main = "Growth
of Sitka pines, with mean lines addded", type="n")
for(j in 1:20)lines(sitka.df$Time[newtree==j], sitka.df$size[newtree==j],
col = "red", lty=2)
for(j in 21:40)lines(sitka.df$Time[newtree==j], sitka.df$size[newtree==j],
col = "blue", lty=2)
```

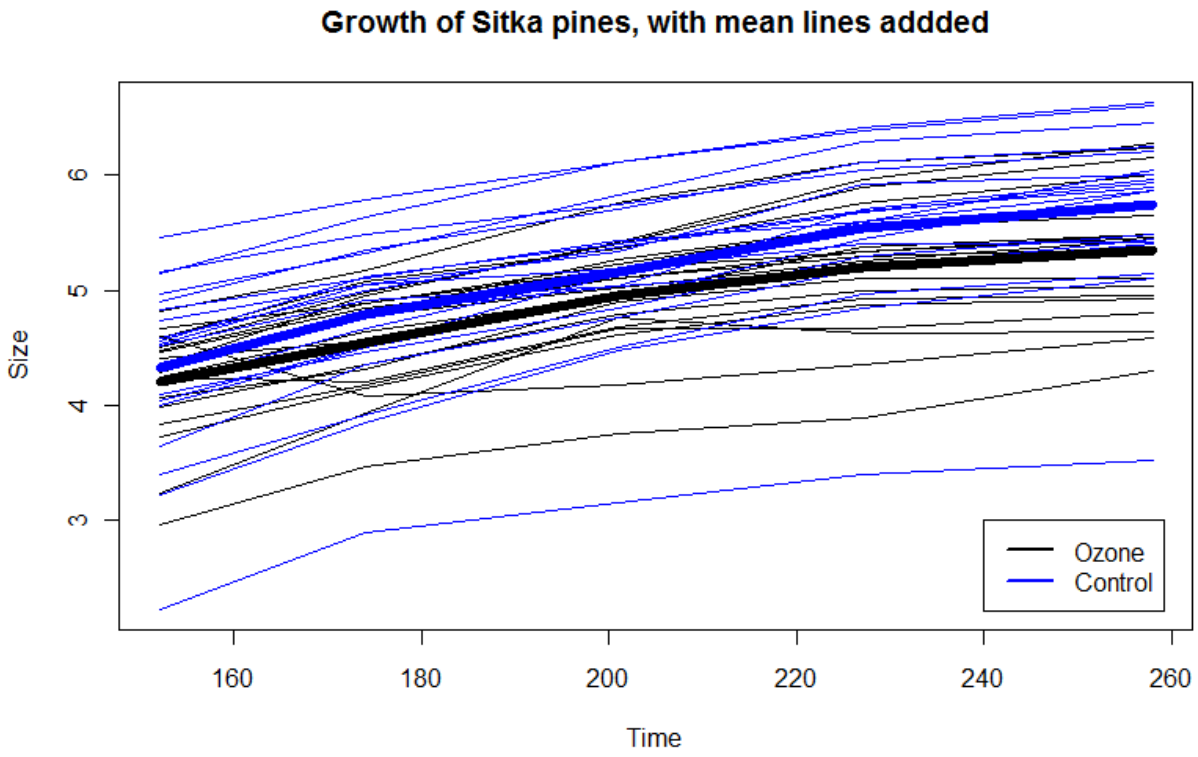
To draw the mean curve, we need to calculate the mean for each treatment/time combination: again we use tapply:

```
> means = tapply(sitka.df$size, list(sitka.df$treat,sitka.df$Time), mean)
```

```
> means
           152   174    201    227    258
control 4.3285 4.789 5.1625 5.5365 5.7495
ozone   4.2085 4.550 4.9550 5.2055 5.3490
```

Then draw these mean curves using a thick line (lwd=6)

```
lines(sitka.df$Time[1:5], means[2,], col = "black", lwd=6)
lines(sitka.df$Time[1:5], means[1,], col = "blue", lwd=6)
legend(240,3, legend = c("Ozone", "Control"), col=c("black","blue"), lwd=2)
```



The control trees on average are bigger and have a slightly higher growth rate. Note however one very small control tree.

Total for assignment: 40 marks