Department of Statistics

COURSE STATS 330

Model answers for Assignment 1, 2004

The data set `cpu.csv` contains data on 209 models of computer in use in the 60's and 70's. This file (in the form of a csv file) can be downloaded from the course web page. The data are also reproduced at the end of this assignment.

The data set has four variables:

`name`: manufacturer and model

`mem`: main memory in kilobytes

`cach`: cache size in kilobytes

`perf`: published performance on a benchmark mix relative to an IBM 370/158-3

Load the data into R.

Answer: The following R code will read in the csv file and create a data frame cpu.df to contain the data:

```
cpu.df<-read.table(file.choose(), header=T, sep=",")
```

[2 marks]

Then answer the following:

1. Do you think the IBM computers have better performance than the other brands? Support your conclusion with suitable graphs.

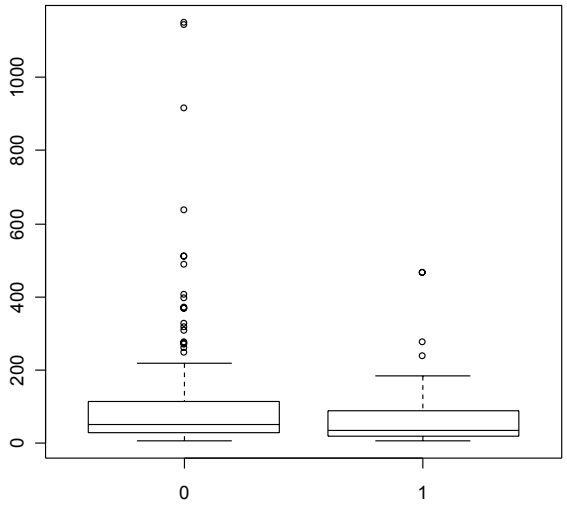
Answer: We first need to make a variable that is 0 for non-IBM computers and 1 for IBM computers. There are many ways to do this: a simple way is to note that in the data, there are 93 non-IBM machines, followed by 32 IBM, followed by 209-125=84 non-IBM. The following code does the trick:

```
Is.IBM<-rep(c(0,1,0),c(93,32,84))
```

We can draw side-by side boxplots to compare the IBM and non-IBM computers:

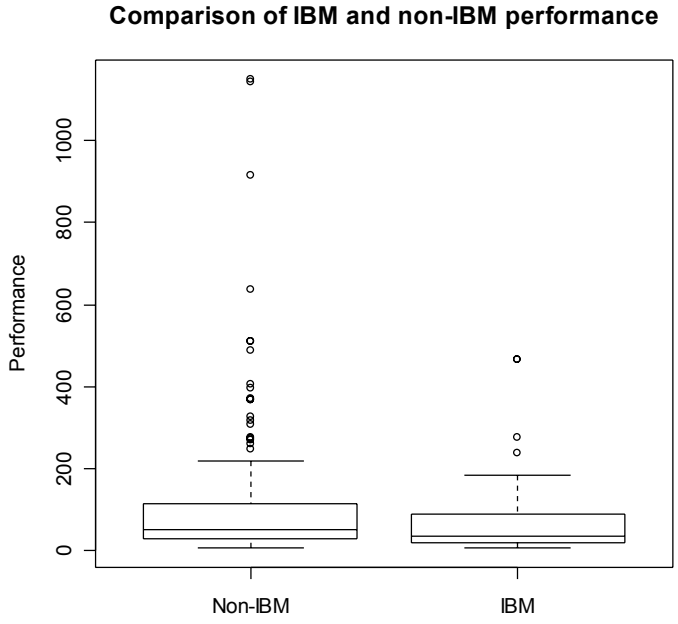
```
boxplot(cpu.df$perf~Is.IBM)
```

[3 marks for the boxplots (dotplots OK too). Should have proper labels]



This can be improved with better labels and a title:

```
boxplot(cpu.df$perf~Is.IBM,
names=c("Non-IBM","IBM"), ylab="Performance",
main="Comparison of IBM and non-IBM performance")
```

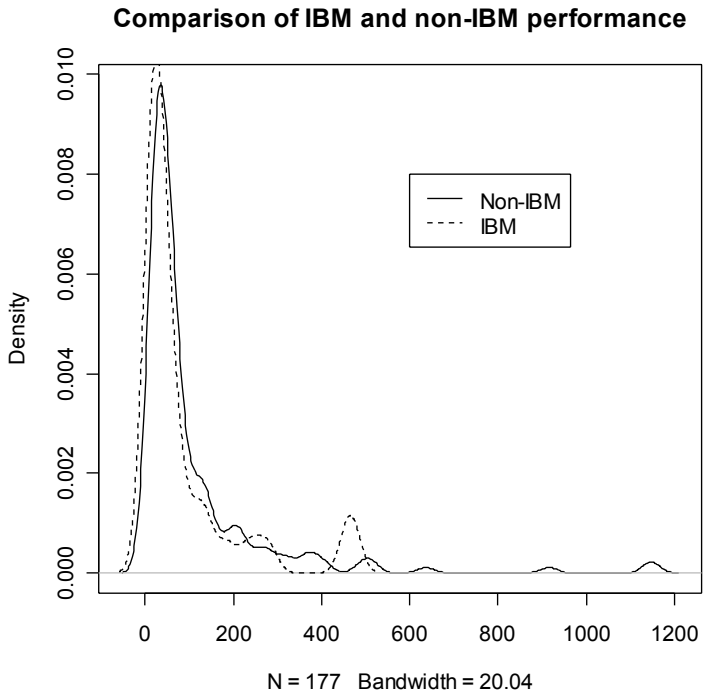


It would appear that the IBM machines have slightly inferior performance. Another possibility is to draw two density curves, and see if the IBM curve lies to the left of the non-IBM curve:

```
# plot the density for the non-IBM machines
plot(density(cpu.df$perf[Is.IBM==0]),
type="l",
main="Comparison of IBM and non-IBM performance")

# add the density curve for the IBM machines
lines(density(cpu.df$perf[Is.IBM==1]),lty=2)

# add a legend to the plot
legend(600,0.008,c("Non-IBM","IBM"),lty=1:2)
```



The IBM curve lies slightly to the left of the non-IBM curve, indicating slightly lower performance for the IBM computers. We could try other graphics, such as qq-plots, but we do not show these here.

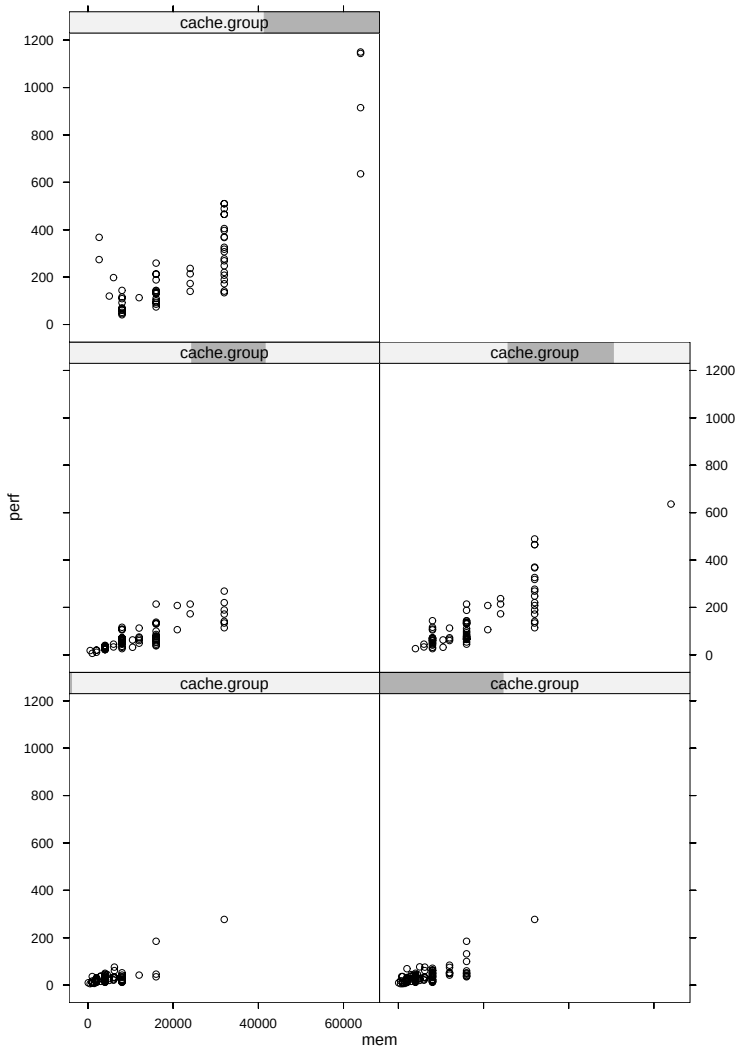
[3 marks for another suitable graphic that gives a good comparison]

[2 marks for a sensible conclusion]

2. Is there a relationship between performance and memory size? If so, is the relationship different for different cache sizes?

Answer: a trellis scatterplot of performance versus memeory size, conditioning on cache size, gives us insight into this question:

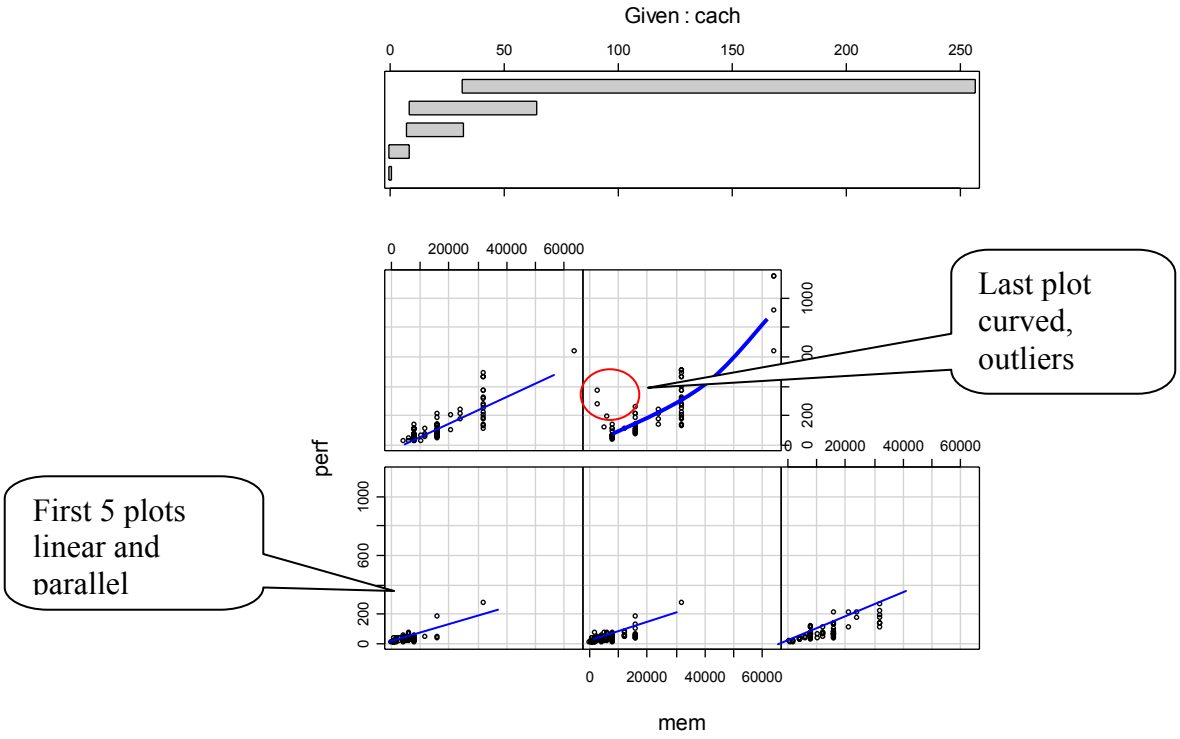
```
cache.group<-equal.count(cpu.df$cach)
xyplot(perf~mem|cache.group,data=cpu.df)
```



An alterative is to use the coplot function:

```
coplot(perf~mem|cach, data=cpu.df)
```

There is a strong relationship between performance and memory size: the bigger the memory size the bigger the performance. The variability of the performance increases as memory increases. The relationship between performance and memory is similar for small values of cache size, but seems slightly different for higher values, mainly due to the 4 points that seem to have high performance for relatively small memory, and the curve in the plot..



[4 marks for a coplot and 6 marks for sensible conclusions]

3. Fit a regression model to the data, using memory size and cache size to explain the performance. Do the variables mem and cach in fact help explain the performance of the different computers? Do we need both variables, or will one be sufficient?

Solution: The model is fitted using the code

```
cpu.lm<-lm(perf~mem+cach, data=cpu.df)
```

We can examine the fit using the R function `summary`:

```
summary(cpu.lm)

Call:
lm(formula = perf ~ mem + cach, data = cpu.df)

Residuals:
    Min      1Q  Median      3Q     Max
-233.86  -31.44    6.22   30.45  420.43

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.756e+01  7.113e+00  -5.280 3.27e-07 ***
mem          9.777e-03  5.068e-04  19.290  < 2e-16 ***
cach         1.105e+00  1.463e-01   7.553 1.36e-12 ***
---
```

```
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 72.25 on 206 degrees of freedom
Multiple R-Squared: 0.8001,     Adjusted R-squared: 0.7982
F-statistic: 412.3 on 2 and 206 DF,  p-value: < 2.2e-16
```

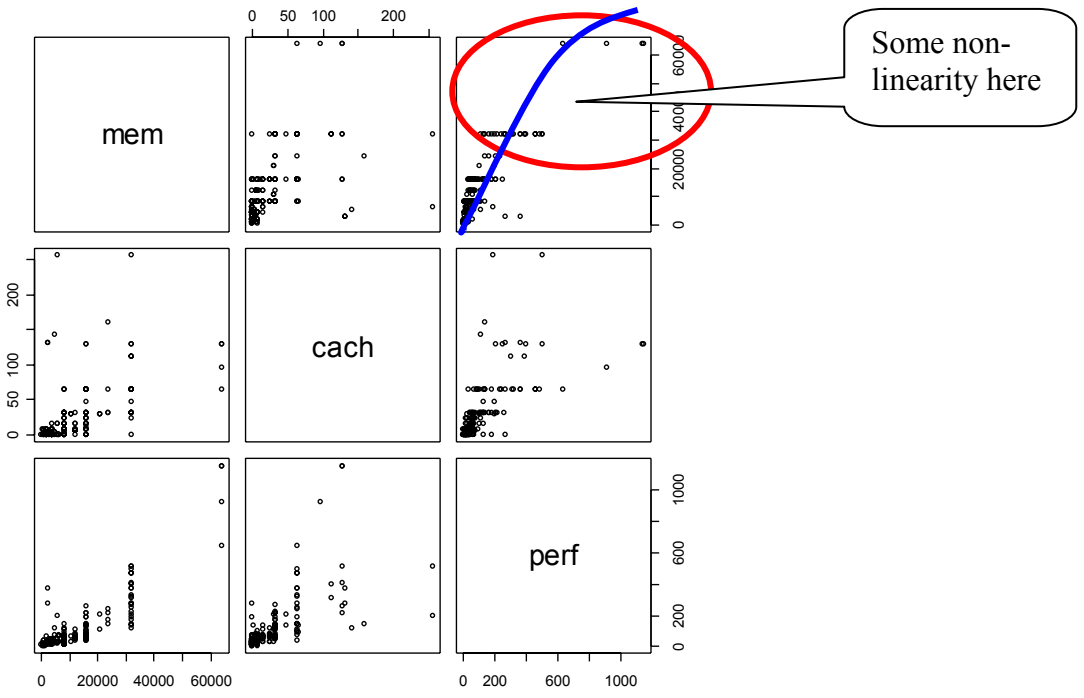
The t-values for both mem and cach are large (and the p-values are small). Both variables contribute strongly to the regression; we cannot dispense with either.

[4 marks for the correct summary, 6 marks for interpreting it properly]

4. Do you think that the relationship between these variables (if one exists) can be adequately represented by a linear (planar) surface? Is the scatter about the regression surface constant?

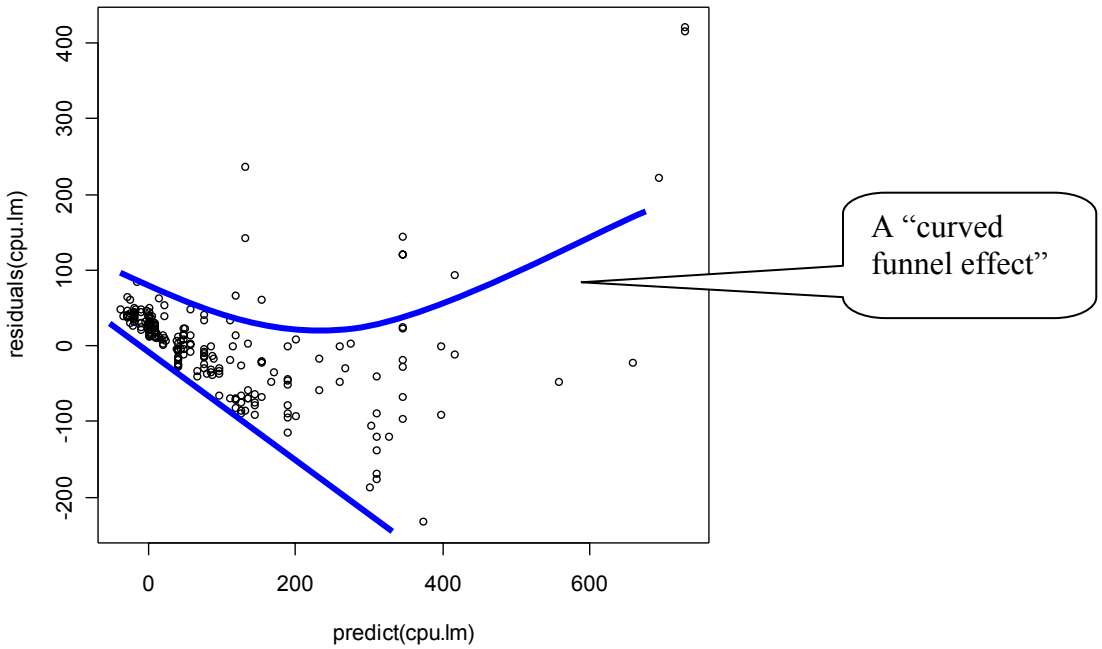
Solution: The coplot suggests that the surface is reasonably planar, except possibly for large cache sizes. This is because the linear relationships in the first 4 coplots are reasonably parallel. A pairs plot indicates some non-linearity for large values of performance:

```
pairs(cpu.df[,-1])  # don't include first column (names)
```



The plots also have a strong indication of non-uniform scatter, with variability of performance beiong much greater for higher levels of performance. This is also very obvious in a plot of residuals versus fitted values:

```
plot(predict(cpu.lm),residuals(cpu.lm))
```



[4 marks for correct interpretation of the coplot, 3 marks for drawing the residuals/fitted value plot, 3 marks for interpreting it.]

There are 10 marks per question for a total of 40. Record mark out of 10 in Cecil.