Department of Statistics

COURSE STATS 330

Model answers for Assignment 3, 2008

Q1.

(a) Read the data into R, checking the data as usual. The data are in the file **college.txt** which is on the course web page. [5 marks]

Read the data into R and make a summary of the resulting data frame:

```
> infile = "http://www.stat.auckland.ac.nz/~lee/330/datasets.dir/presidents.txt"
> presidents.df = read.table(infile, header=T)
> summary(presidents.df)
      Comp            Gen              Rev              Endow             Top50
 Min.   :121.5   Min.   :0.000   Min.   : 15.00   Min.   :  15.36   Min.   :0.0000
 1st Qu.:193.0   1st Qu.:1.000   1st Qu.: 39.00   1st Qu.:  76.37   1st Qu.:0.0000
 Median :235.7   Median :1.000   Median : 62.00   Median : 136.52   Median :0.0000
 Mean   :244.8   Mean   :0.789   Mean   : 78.07   Mean   : 231.35   Mean   :0.3853
 3rd Qu.:276.0   3rd Qu.:1.000   3rd Qu.: 98.00   3rd Qu.: 318.22   3rd Qu.:1.0000
 Max.   :591.0   Max.   :2.000   Max.   :298.00   Max.   :1207.41   Max.   :1.0000
      Net               G.S              PExp              TExp
 Min.   :  26.03   Min.   : 3.011   Min.   :  9.182   Min.   : 15.72
 1st Qu.: 114.72   1st Qu.: 9.214   1st Qu.: 30.517   1st Qu.: 37.16
 Median : 199.14   Median :13.266   Median : 43.405   Median : 52.31
 Mean   : 308.75   Mean   :16.885   Mean   : 50.571   Mean   : 60.03
 3rd Qu.: 394.20   3rd Qu.:21.194   3rd Qu.: 66.500   3rd Qu.: 75.46
 Max.   :1461.00   Max.   :68.250   Max.   :146.101   Max.   :173.97
```

A maximum value for the variable GEN is suspect:

```
> stem(presidents.df$Gen, scale=0.1, width = 120)

  The decimal point is at the |

  0 | 00000000000000000000
  1 | 00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
  2 | 0
```

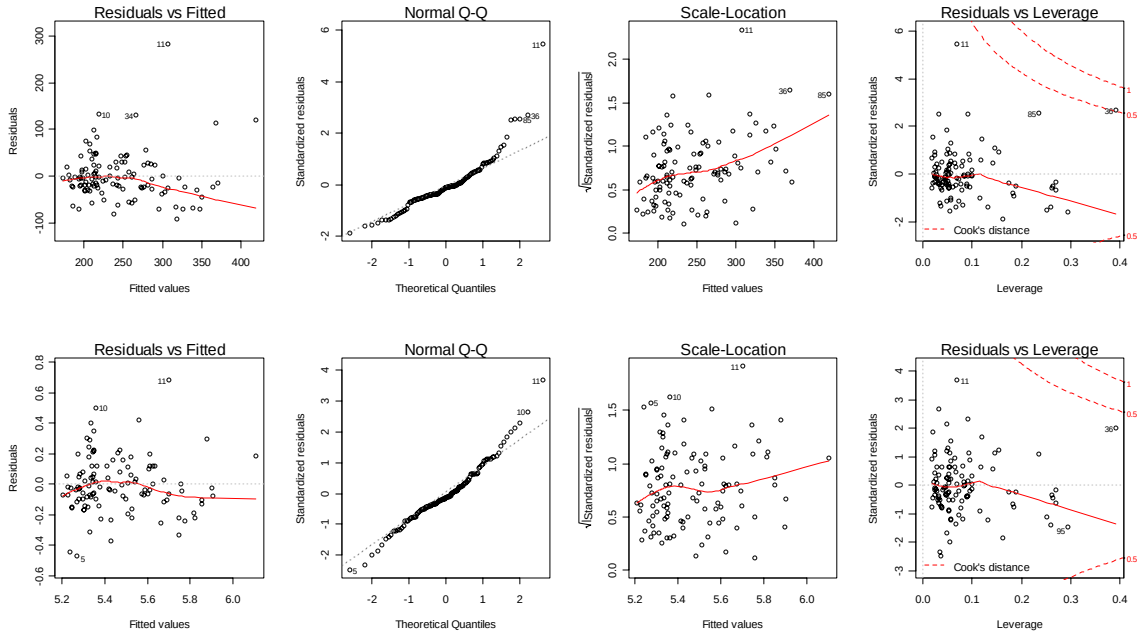
The value 2 is invalid, as this is a 0/1 variable. Going back to the assignment sheet we see that the offending observation is 19, and that the value of Gen should be 1 (male). We correct it as follows:

presidents.df[19,2] = 1

(b) Fit a model to the data using all the explanatory variables. Is it better to use comp as a response or log(comp)? [10 mark]

We fit the model using all the variables, with and without a log transformation of the variable Comp:

```
> pres.log.lm = lm(log(Comp)~., data=presidents.df)
> pres.raw.lm = lm(Comp~., data=presidents.df)
> par(mfrow=c(2,4))
> plot(pres.raw.lm)
> plot(pres.log.lm)
> summary(pres.raw.lm)$r.squared
[1] 0.4925899
> summary(pres.log.lm)$r.squared
[1] 0.5146939
```



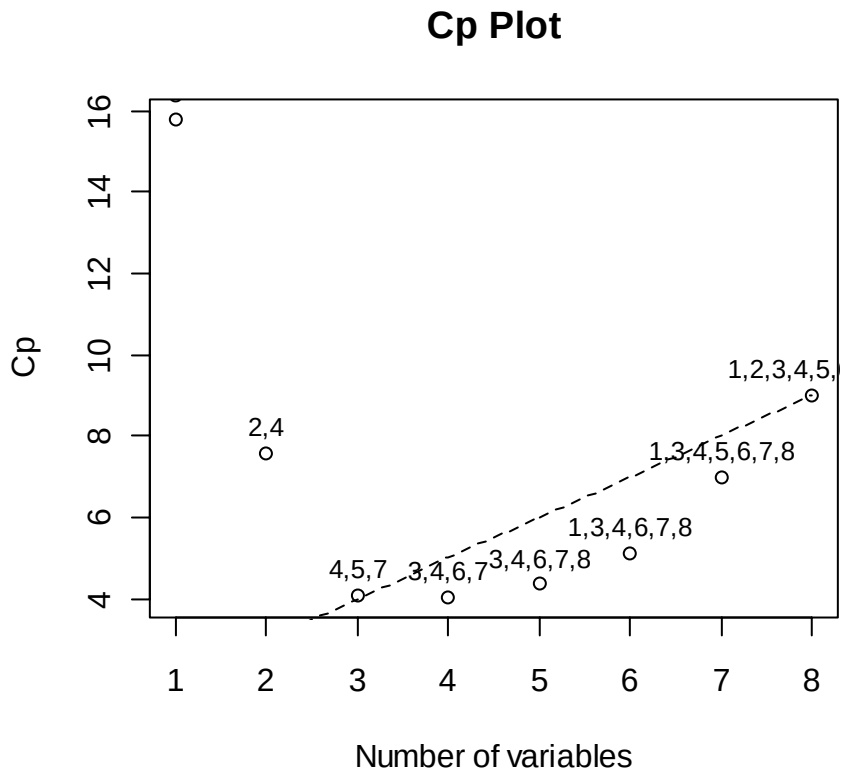
It seems that the R^2 is slightly better using the log transformation; the residual plots are also better, with no influential points, a more normal q-qplot, and no hint of variance depending on mean.

(c) Using the response you chose in (b), develop a prediction equation that will allow you to predict the salary of a college president. Are all the variables required in the regression? Use variable selection techniques to choose a suitable subset if not. If you see any problems with your chosen model take appropriate corrective action until the model is satisfactory. [20 marks]

If we use the all possible regressions approach with the log transformation, we get the following output:

```
> library(leaps)
> all.poss.regs(log(Comp)~., data=presidents.df)
   rssp sigma2 adjRsq     Cp     AIC     BIC    CV Gen Rev Endow Top50 Net G.S PExp TExp
1 4.449  0.042  0.408 15.797 124.797 130.179 0.434   0   0     0     0   1   0    0    0
2 4.073  0.038  0.453  7.569 116.569 124.643 0.399   0   1     0     1   0   0    0    0
```

```
3 3.870  0.037  0.475  4.078 113.078 123.843 0.390   0   0     0     1   1   0    1    0
4 3.795  0.036  0.481  4.031 113.031 126.488 0.389   0   0     1     1   0   1    1    0
5 3.733  0.036  0.484  4.363 113.363 129.511 0.393   0   0     1     1   0   1    1    1
6 3.688  0.036  0.485  5.129 114.129 132.968 0.393   1   0     1     1   0   1    1    1
7 3.683  0.036  0.481  7.004 116.004 137.535 0.406   1   0     1     1   1   1    1    1
8 3.683  0.037  0.476  9.000 118.000 142.222 0.421   1   1     1     1   1   1    1    1
```



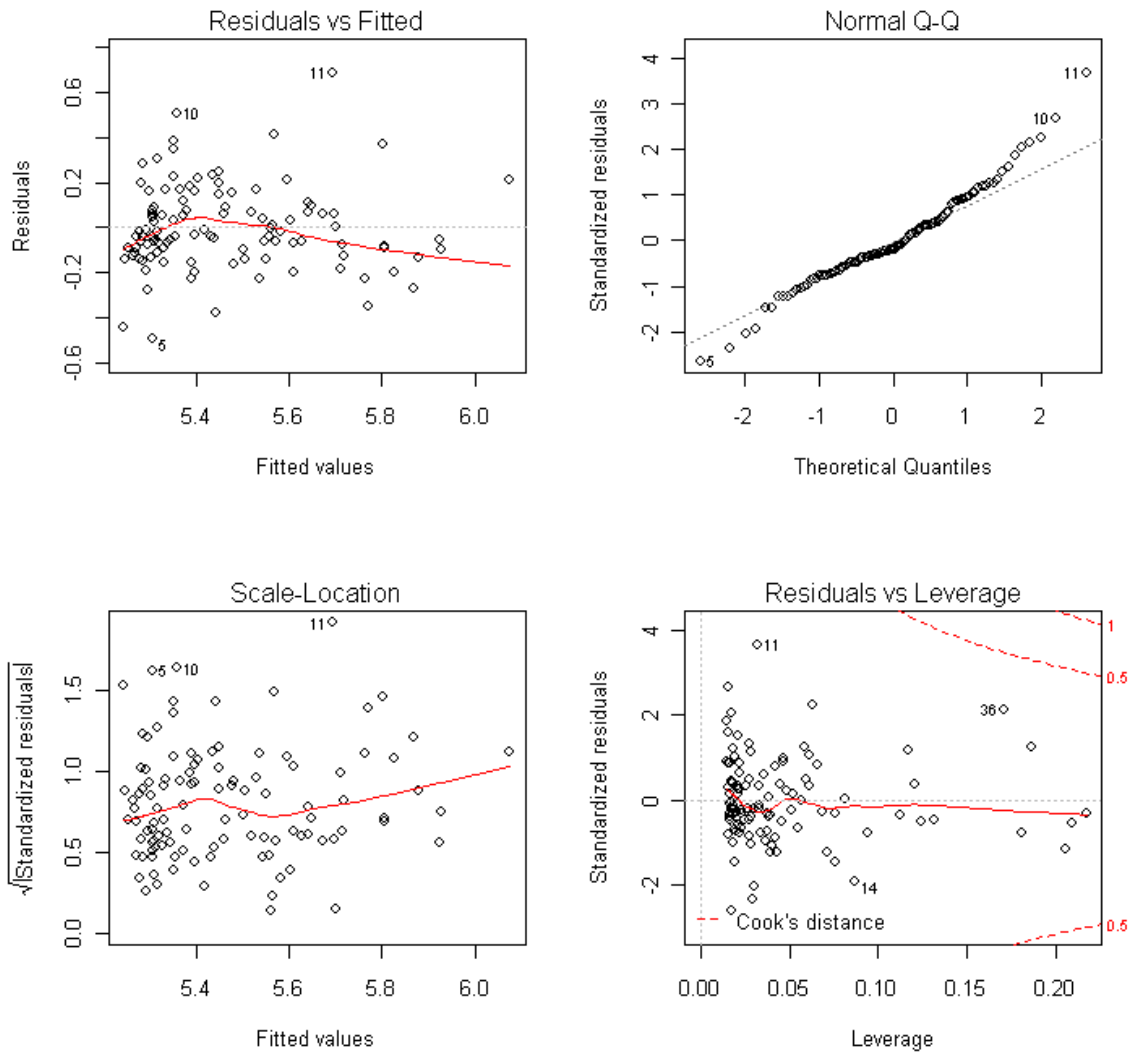
Good models are

Endow + Top50 + G.S + PExp (smallest CV, smallest AIC, good CP)
Top50 + Net + PExp (smallest BIC, best CP)
Gen + Endow + Top50 + G.S + PExp +TExp (Adjusted R^2, s^2)

We will choose the first, as CV usually gives a good model for prediction.

Refit the model and plot

```
pres.sub.lm = lm(log(Comp)~Endow+Top50+G.S+PExp, data=presidents.df)
plot(pres.sub.lm)
```



Point 11, while not high leverage, has quite a large standardized residual and has an influence on the standard errors. We delete it and refit. The resulting residual plots are not too bad. We then tried some gam plots, and after some experimentation decided to transform PExp as a cubic. The residual plots from this fit indicated that point 85 had a large cooks distance so we deleted this point as well. After a final check of residuals so we decided to go with this fit:

```
pres.final.lm = lm(log(Comp)~Endow+Top50+G.S+poly(PExp,3),
data=presidents.df, subset=-c(11,85))
```

An alternative model is the model without the polynomial terms, deleting point 11 only. This model is fitted by

```
pres.alternative.lm = lm(log(Comp)~Endow+Top50+G.S+ PExp,
data=presidents.df, subset=-11)
```

(d) Use your final model to predict the compensation of the female president of a top 50 college having revenue of $117m, an endowment of $156.256m, net assets of $235.154m, grants and supports of $20.676m, program expenses of $73.357m and total expenses of $90.712m. Express your prediction as a point prediction. [5 marks]

```
pred.data = data.frame(Endow=156.256, Top50=1, G.S=20.676, PExp=73.357)
> predict(pres.final.lm, pred.data)
       1
5.552961
>
> predict.log = predict(pres.final.lm, pred.data)
> exp(predict.log)
       1
258.0005
```

Thus, our prediction for the president's salary is about $258,000.

In actual fact the data in pred.data corresponded to an actual college that I had removed from the data set. The true salary of the president of this college was $244,858.

The prediction obtained using the alternative model is $259,249.

Q2 (STATS 762 only).

How well would stepwise regression do in the college example? To investigate this, assume that the model chosen in Q1 is correct. Then,

(a) Generate 1000 data sets by using the complete set of X variables, but generating a new set of responses each time by adding to the fitted values from the Q1 model a vector of random normal errors. Set the standard deviation of the errors to be equal to the estimate obtained in Q1.

(b) For each data set, perform a stepwise regression and record the model chosen.

(c) Make a table of the 1000 chosen models. What fraction of these coincide with the true model (i.e. the model you used to generate the 1000 data sets)? What do you conclude about stepwise regression as a technique?

Show all R code used.

First, lets examine how we can generate a data set from the model we fitted. We will use the same X-data set each time. In general, y-data is equal to a mean plus an error:

$Y = \mu + \varepsilon$

Assuming that the alternative model we fitted in Q1 is true is equivalent to assuming that the true plane is the same as the plane fitted in Q1, i.e. the means are the same as the fitted values. The estimated standard deviation for our fitted model is 0.1791:

```
> summary(pres.alternative.lm)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.2061771  0.0387690 134.287   <2e-16 ***
Endow       0.0002795  0.0001093   2.557   0.0120 *
Top50       0.1118877  0.0486155   2.301   0.0234 *
G.S         0.0035520  0.0018906   1.879   0.0631 .
PExp        0.0016423  0.0010082   1.629   0.1064
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1791 on 103 degrees of freedom
Multiple R-squared: 0.5098,     Adjusted R-squared: 0.4908
F-statistic: 26.78 on 4 and 103 DF,  p-value: 3.085e-15
```

or, more compactly

```
> summary(pres.alternative.lm)$sigma
[1] 0.1790798
```

Thus, we can generate a set of y-values by the code

```
mu = predict(pres. alternative.lm)
y = predict(pres. alternative.lm) + rnorm(length(mu), sd= summary(pres.
alternative.lm)$sigma)
```

To find the model selected by stepwise regression, we need fit the null model, and use step:

```
null.lm = lm(y~1, data=presidents.df[-11,])
full.formula =  y~ Gen +   Rev + Endow + Top50 + Net + G.S + PExp + TExp
stuff = step (null.lm, scope=full.formula, direction = "both", trace=0)
```

From the object stuff, we can extract a character string containing the actual model chosen:

```
> paste(sort(attr(stuff$terms,"term.labels")), collapse="+")
[1] "Endow+Net+PExp+Top50"
```

Now, to repeat this 1000 times, we need a loop. The result is a character string representing the model chosen. We can store all these in a character vector of length 1000.

The code is

```
Nsim=1000
mu = predict(pres. alternative.lm)
result = character(Nsim)
for(i in 1:Nsim){
y = mu + rnorm(length(mu), sd=summary(pres. alternative.lm)$sigma)
null.lm = lm(y~1, data=presidents.df[-11,])
```

```
full.formula =  y ~  Gen +   Rev + Endow + Top50 + Net + G.S + PExp + TExp
stuff = step (null.lm, scope=full.formula, direction = "both", trace=0)
result[i] = paste(sort(attr(stuff$terms,"term.labels")), collapse="+")
}
```

Then, to make a sorted table of the chosen models, we just type

```
sorted.table=sort(table(result), decreasing=TRUE)
```

The first 10 elements are

```
> sorted.table[1:10]
result
     G.S+Net+Top50     Net+TExp+Top50     Net+PExp+Top50      G.S+Rev+Top50
               117                 69                 50                 49
    G.S+TExp+Top50    Endow+G.S+Top50          Rev+Top50 G.S+Net+TExp+Top50
                44                 33                 27                 26
         Net+Top50  G.S+Gen+Net+Top50
                26                 25
```

The most commonly chosen model is `G.S+Net+Top50` which is chosen in 11.7% of the 1000 repetitions.

The true model is `Endow+G.S+PExp+Top50,` which was only chosen once!

```
> sum(result=="Endow+G.S+PExp+Top50")
[1] 1
```

Thus, stepwise is pretty bad at finding the true model.

Note that we have represented models in a particular form, with the variables listed alphabetically, and no spaces in the string.