Linear models, part I   Fall 2009

Lab report due Nov 10

Lab 2: Ordinary least squares, diagnostics and goodness-of-fit

For this lab you will analyze two data sets; (1) frost duration; and (2) weights of potatoes. Both data set are available on the class home page.
The frost data set consists of average duration (in days) of frost conditions each year, recorded at 20 different weather stations in Virginia. The elevation and latitude of the stations are also included in the data set.
Let's suppose that frost duration is our response variable of interest. In this lab you'll get a chance to explore the data set, and try to model the frost duration as a function of the other variables.
Start by reading the data into R;

```
frost<-read.table(‘‘frost.txt’’,header=T)
```

Make sure you use the correct file name when you upload the data. Check the names of the uploaded variables by issuing the command `names(frost)`.
Goal of the lab: In this lab, I want you to; demonstrate some basic analysis skill; show that you are aware of the basic assumptions underlying a least squares fit; check these assumptions; perform basic diagnostics of a least squares fit (residual analysis).
The tasks I will outline are not difficult. The challenge in this lab is to draw conclusions, and interpret the outcome of this basic analysis. I also expect you to make a serious attempt at presenting the results in a coherent report. Please review the class notes on structuring a report and follow them closely. Some things to remember; NEVER simply quote a book or the lab, make sure to rephrase everything in your own words; NEVER contradict yourself, the data, or the results - you have to be consistent (if some results confuse you, say so and say why); THINK about the readability/accesability of your report (space things out, full sentences, only the relevant figures with captions,...).

1 Plotting the data

Explore the frost data set graphically. Try to summarize the data in a visual and meaningful way. Include this graphical analysis in your report, make sure to motivate why you include a certain graph - be selective.
We talked about numerical and graphical summaries in class - comment on different summaries you could use for these data, pros and cons.
(Some (not all) R functions to consider: `hist()`,`boxplot()`,`qqnorm()`,`fivenum()`. Read the help files if you're unsure on how to use them.)
Use various graphical tools to get a feel for the relationship between the variables in the data (e.g. duration vs elevation). Comment on your findings. Could a linear model be used to summarize this relationship?

2 OLS; ordinary least squares

The function `lm()` in R fits a linear model to a data set using least squares and assuming normal errors for making inferences. Review the help file for the function. Now fit the data with duration as the response variable, and elevation and latitude as explanatory variables.

```
mod1<-lm(frost$dur∼frost$elev+frost$lat)
```

Note; when you call a variable in a data frame like `frost` it's enough to use as many letters as needed to distinguish the variable from other variables (e.g. above you could use `frost$d, frost$e, frost$l`).

Perform some basic diagnostics - residual plots (e.g. fitted/residuals, explanatories/residuals), checking normality and homoscedasticity assumptions etc. Comment on the fit of the model, the coefficient estimates and standard errors. Does the linear model seem appropriate? Can you interpret the model coefficients?

Useful commands: `summod1<-summary(mod1)`, `names(summod1)`, `plot(mod1)`, `plot(mod1$fit, mod1$res)`, `qqplot()`, etc. Check the help files of these commands if you're unsure how to call them.

3 Leverage

The function `lm.influence` computes the diagonal elements of the hatmatrix, and the change in slope and σ^2 estimates if one observation is dropped from the model. These outputs are all good tools for identifying observations with high leverage that may influence the fit. Read the help file on the function and apply it. Provide some graphical displays and comment on the result. The command `par(mfrow=c(m,n))` allows you to put multiple plots in the same window. The `identify` function can be helpful here. Example:

```
plot(model$fit,model$res,xlab=''fitted values'',ylab=''residuals'')
identify(model$fit,model$res,seq(1,20))
```

You can identify observations by index by clicking the left mouse button, quit by clicking the right mouse button. Do something similar for the leverage plots.

Are there any observations in this data set that you would classify as outliers, as points of high leverage, as influential on the fit? What happens if you remove one or more of these observations? Make sure you state which observation(s) you removed, and comment on the updated model coefficients and model fit.

```
mod1b<-lm(frost$dur[-i]∼frost$elev[-i]+frost$lat[-i])
```

The square brackets are used to identify a subset of observation indices. Here `[-i]` denotes *all indices not equal to i*, i.e. the command "removes" observation i from the fit.

4 Testing

Based on theory and the output above, construct approximate 95% confidence intervals for the intercept and slope coefficients in the original model fit. Discuss the outcome. Do you think we can simplify the model by dropping a coefficient, and in that case which?

Fit a simplified model using the command `lm`, be sure to name it something different than your full model above. An alternative is to use the `update` function. Let's say your first model

(the full model) is called `mod1`. If you want to remove `frost$elevation` from the fit, use the command `mod2<-update(mod1,.∼.-frost$elevation)`.
Using the function `anova` you can perform an F-test to decide between the full and simplified model.

```
anova(mod1,mod2)
```

What is the outcome of the F-test in your case? Be sure to report the sum of squares, the F-ratio and the degrees of freedom, and the p-value. What is your conclusion?

5 Collinearity and numerical instabilities

The second data set consists of measurements on 18 potatoes. The data set contains the weight, the length and the breadth of the potatoes. We will consider weight as the response variable. Consider a linear model of weight as a function of length and breadth. Fit this model to the data and discuss the outcome (diagnostics, coefficients, goodness-of-fit).
Now, the length and breadth measures are heavily correlated (compute the pairwise correlations in the data yourself `cor()`). Remember from class that $V(\hat{\beta}) = \sigma^2(X'X)^{-1}$, where the $X'X$ matrix is proportional to $COV(X)$. The effect high correlations among explanatory variables (x's) is to inflate the estimation variance of the coefficients. That results in an unstable model, meaning that small perturbations of the data can lead to radically different coefficient estimates. I want you to examine the impact of correlated variables using the potato data.

Now, let's first fit the full model with the potato data:

```
moda<-lm(potato$weight ~ potato$length+potato$breath) .
```

Fit a reduced model using only length:

```
modb<-lm(potato$weight ~ potato$length) .
```

Compare the estimates and standard errors of the length coefficient in both models. Which model has a larger standard error for the coefficient associated with it? Why?

Let us now add a small amount of noise to the potato weight data and refit both models:

```
newweight<-potato$weight+rnorm(18,sd=4)
```

(You can pick another value for the added noise sd if you want. The sd of the weight data is 23.)

```
moda2<-lm(newweight~ potato$length+potato$breath) .
modb2<-lm(newweight ~ potato$length) .
```

Check the coefficient values and standard errors in the two noise added models. Discuss.

It is probably a good idea to repeat the exercise several times and compare the coefficient values. You can also consider adding noise to the other variables and see how this affects the fit.
You can store the coefficient values from several iterations of the above:

```
Coefmata<-matrix(0,B,3)
Coefmatb<-matrix(0,B,2)
for (k in (1:B)) {
newweight<-potato$weight+rnorm(18,sd=4)
moda2<-lm(newweight ~ potato$length+potato$breadth)
modb2<-lm(newweight ~ potato$length).
Coefmata[k,]<-moda2$coef
Coefmatb[k,]<-modb2$coef
}.
```

You can now study the effect of the noise on the coefficient estimates. Plot the stored coefficient values in Coefmata (the 2nd and 3rd columns contain the estimates of the coefficients for length and breadth) against each other. Compare the variability of the coefficient estimates for length weight and the variability of the sum of those two coefficients. Compare with the estimate of the length coefficient in the reduced model (Coefmatb, column 2). Discuss. What is the impact of correlation variables on estimation? (Hint: consider using `boxplot` or scatter plots to compare the coefficient estimates.)

An extra question: if you assume that the potatoes are uniform in density - can you think of another mathematical model for relating length and breadth to the weight (not an additive linear model)? You can create your own predictor variable from the length and breadth variables and fit a new model.

```
newvariable<-???
newmodel<-lm(potato$weight newvariable)
```

How well does the new model fit the data compared with the additive model.

6 Summary

Be sure to summarize the analyses of the frost duration and potato data sets. What are your main conclusions? Did you check assumptions? Were there any clear violations? Were any particular observations identified by the diagnostics or leverage analysis? Could you interpret the models?