DEPARTMENT OF STATISTICS

Course STATS 330: Advanced Statistical Modelling

Tutorial Sheet 6: September 18, 2008

This tutorial is designed to give you practice in fitting models having a mix of categorical and continuous explanatory variables, and interpreting the result.

In this tutorial we will be using the **calf data** which you can download from the web site. You are invited to explore data from an experiment investigating the rate at which calves gain weight. The average daily weight gain (ADG) during 3 to 9 months of age for 3 breed of calves was studied.

It was thought that ADG is partly an inherited trait. Thus the ADG of each calf's sire (father) and ADG of each calf's dam (mother) when the calves were growing during 3 to 9 months of age, were considered as possible covariates, as well as the breed.

The variables in the data set are:

`breed`:	breed of calf, a factor with levels 1, 2 or 3.
`adg`:	average daily weight gain of calf.
`sadg`:	average daily weight gain of calf's sire (as a calf).
`dadg`:	average daily weight gain of calf's dam (as a calf).

For this tutorial, you are to compare the average daily weight gains for the three breeds of calves. You are to fit a model that utilises the covariates breed and dadg. You should run suitable diagnostic checks for your fitted model.

Task 1: Read in the data and crate a suitable data frame for the analysis

Warning: the variable `breed` is categorical but the levels are coded as numbers, so you will need to turn it into a factor. You can use R-code

```
temp.df<-read.table(
"http://www.stat.auckland.ac.nz/~lee/330/datasets.dir/calf.txt", header=T)
attach(temp.df)
calf.df<-data.frame(breed=factor(breed), adg=adg,  dadg=dadg)
detach(temp.df)
```

Now everything will work properly.

Task 2: Fit an initial model

The response is `adg`, and the explanatories are `breed` (categorical) and `dadg`, which is continuous.

The natural model to try is one that fits a separate regression line to each of the 3 breeds. This will have 6 parameters (constant term and `dadg` coefficient for each of 3 regressions).

```
> calf.lm<-lm(adg ~  breed + dadg + breed:dadg, data=calf.df)
```
(NB: can also write model as `adg ~ breed*dadg`)
```
> summary(calf.lm)
```

Do you think all the variables are required? Try an anova

```
> anova(calf.lm)
```

Seems we can dispense with the `dadg` x `breed` interaction – why – what does this mean? (coefficient of `dadg` the same in all 3 regressions, parallel lines)

All this suggests the model `adg ~ breed + dadg`

Task 3: Diagnostics

Run a quick diagnostic check on this model. Is there any indication of a bad model fit?

Task 4: Interpretation

How do we interpret this model? Which breed has the greatest weight gain? First, let's identify the fitted lines:

```
Call:
lm(formula = adg ~ breed + dadg, data = calf.df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.35183 -0.08102  0.01518  0.09045  0.29735

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.08165    0.16683  12.477 1.34e-10 ***
breed2      -0.44675    0.09522  -4.692 0.000159 ***
breed3       0.88217    0.10509   8.394 8.15e-08 ***
dadg         0.44591    0.07405   6.022 8.57e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.176 on 19 degrees of freedom
Multiple R-squared: 0.9581,     Adjusted R-squared: 0.9515
F-statistic: 144.9 on 3 and 19 DF,  p-value: 2.874e-13
```

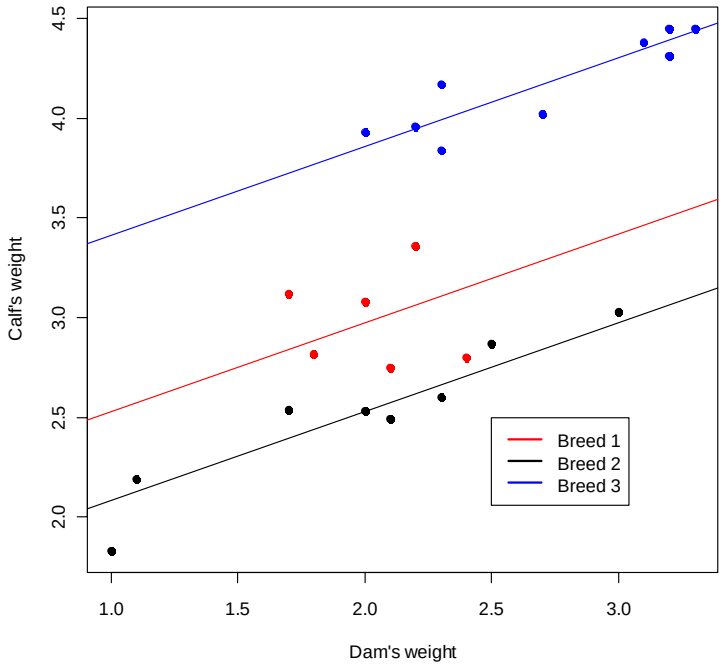
For Breed 1, the line is
adg ~ `2.08165+ 0.44591 dadg`

For Breed 2, the plane is
`adg ~ (2.08165 – 0.44675) + 0.44591 dadg` i.e.
`adg ~ 1.63495 + 0.44591 dadg`

For Breed 3, the plane is
`adg ~ (2.08165 + 0.88217) + 0.44591 dadg` i.e.
`adg ~ 2.96382 + 0.44591 dadg`

These lines are illustrated in the following plot:

```
plot(adg ~dadg, data=calf.df, type="n", xlab="Dam's weight",ylab = "Calf's
weight")
mycol = c("red", "black", "blue")
points(calf.df$dadg, calf.df$adg, pch=19, col = mycol[calf.df$breed])
abline(2.08165, 0.44591 , col=mycol[1])
abline(1.63495, 0.44591 , col=mycol[2])
abline(2.96382, 0.44591 , col=mycol[3])
legend(2.5,2.5, legend=paste("Breed", 1:3), col=mycol, lwd=2)
```



We can also calculate means for each breed:

```
> tapply(calf.df$adg,calf.df$breed, mean)

       1        2        3
2.988333 2.510000 4.167778
```

Note that the means are determined by the regression line and the values of dadg.