DEPARTMENT OF STATISTICS
Course STATS 330: Advanced Statistical Modelling
Tutorial Sheet 3: Week 4, 2012

This tutorial is designed to give you practice in the following:

- Calculating VIF's
- Checking a model for linearity (planar data) and selecting transformations
- Checking for outliers

In this tutorial we will be using the **car data** on the website. These data were used in Assignment 2, 2002. After completing the tutorial, you should have the skills you need to attempt Assignment 2.

In this tutorial you will explore a data set consisting of various measurements on 138 cars that were taken from Road and Track's "*The Complete '99 Car Buyer's Guide"*.
The variables in this data set are:

CITY: mileage (miles per gallon) in city driving, (response)
PRICE: price in dollars (US),
WEIGHT: weight in pounds,
DISP: displacement in cubic centimetres,
COMP: compression ratio as value to 1,
HP: horsepower at 6300 rpm,
TORQ: torque at 5200 rpm,
TRANS: transmission (1 = automatic, 0 = manual),
CYL: number of cylinders.

Task 1: Read in the data

Download the cars data from the web. Make a data frame **`cars.df`**. The response variable of interest is **`CITY`**, and we want to model this in terms of the other variables. The variable names in this example are UPPER CASE.

Note that you can make the data frame directly from the web page by typing

```
cars.df=read.table(
"http://www.stat.auckland.ac.nz/~lee/330/datasets.dir/cars99.data",
header=T)
```

You can get the URL for the data by copying and pasting from the address bar of the web browser. In this method, you don't have to download and store a copy of the text file. However, if you want to use Notepad to edit the data (e.g. to correct errors), you will need to download the text file.

Task 2: make a pairs plot

Make a pairs plot of the data. Are any of the explanatory variables closely related to each other? Calculate the correlations between the variables to confirm what you see in the plot. Do the relationships seem linear?

```
pairs(cars.df)
X<-cars.df[,-1]  # why do we do this?
round(cor(X),3)
```

Task 3: Fit the model and compute the VIF's

```
cars.lm = lm(CITY~ PRICE + WEIGHT + DISP + COMP + HP + TORQ
+ TRANS + CYL, data=cars.df)
summary(cars.lm)
```

Compute VIF's:

```
X.explan=cars.df[,-c(1,4)]    # delete columns 1 and 4 from
                              # the data frame
VIFs = diag(solve(cor(X.explan)))
```

There are a few of high VIF's. Which variables are affected? Can you explain why some variables are quite highly correlated with the response but are not significant in the regression?

Task 4: examine the residuals

```
par(mfrow=c(2,2)) # 2x2 layout of plots
plot(cars.lm)
```

Main points:

- Residuals/fitted values plot confirms the non-linearity
- Outliers (pts 30, 47, 125? 69?)

Sometimes we get a lot of outliers when the data are non-planar, as is the case here.

Task 5: transform

The relationship between the response CITY and the other variables doesn't seem linear. Have a look at the GAM fits to see which, if any of the explanatory variables or the response need transforming.

```
# do the GAM plots, don't forget to load the mgcv library
library(mgcv)
```

```
plot(gam(CITY~+s(PRICE) + s(WEIGHT) + s(DISP) + s(COMP)+
s(HP)+ s(TORQ)+ TRANS+ CYL,data=cars.df))
```

These plots show that almost all the variables need transforming. In this case, it is often a good idea to try transforming the response with a power transformation.

Task 6: transforming the response

Try transforming the response with a variety of powers (including log and reciprocal), and refitting. Note how the R^2 changes. The reciprocal seems good – the R^2 goes up from 78% to 88%. (The reciprocal is interpreted as gallons per mile rather than miles per gallon.) The Box-Cox plot could be useful here – indicates reciprocal or reciprocal square root.

Task 7: re-examine residuals

```
recip.lm<-lm(I(1/CITY)~ PRICE + WEIGHT + DISP + COMP + HP +
                    TORQ + TRANS + CYL, data = cars.df)
summary(recip.lm)
```

Residual plots look OK, except for the outlier 47 (Ferrari F355 Berlinetta).

Remove this:

```
recip.no47.lm<-lm(I(1/CITY)~ PRICE + WEIGHT + DISP + COMP +
HP + TORQ + TRANS + CYL, subset=(1:138)[-47],data =
cars.df)
```

At this point we have a reasonable model: the R^2 is now 92% and the residual plots look OK. Not all variables are significant however – we could drop some. See Lectures 14 and 15 for techniques for choosing which variables to retain.

Task 8: Add a variable to the data frame

As an alternative to using the **`I(1/CITY)`** construction above, you could make a new variable**,** **`recip.city`** say, and add it to the data frame, using the code

```
recip.city = 1/cars.df$CITY

newcars.df = data.frame(cars.df, recip.city = recip.city)
```

or, more compactly, as

```
newcars.df = data.frame(cars.df, recip.city =
1/cars.df$CITY)
```

Task 9: Assess the size of the largest studentised residual

We noted in class that the studentised residuals are approximately normally distributed, so lie between +/- 2 with 95% probability.

However, the *largest* residual is typically bigger than 2 in magnitude. The bigger the sample size, the bigger the largest residual will be.

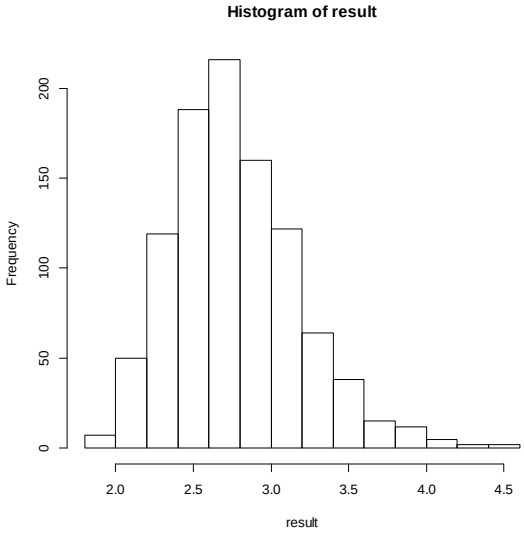
To quantify this, we can do a small simulation: repeatedly draw normal samples, and calculate and record the largest value. We can then draw a histogram of these largest values to get an idea what is typical.

For example, for the cars data, there are n=137 observations, excluding the Ferrari F355 Berlinetta.

The following R code draws N=1000 random normal samples of size n=137, and records the 1000 maximum values in a vector result:

```
N=1000
n=100
result = numeric(N)
for(i in 1:N)result[i] = max(abs(rnorm(n)))
```

Now draw a histogram of the results:



The largest studentised residual from the regression fit had magnitude 2.61326, so this is not atypical.

```
> max(abs(rstudent(recip.no47.lm)))
[1] 2.61326
```