DEPARTMENT OF STATISTICS
Course STATS 330: Advanced Statistical Modelling
Tutorial Sheet 4: August 21, 2008

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

- Checking for outliers and high-leverage points
- Checking for equal scatter
- Checking for independence

In this tutorial we will be using the **car data** that were used in tutorial 3 and the **sales data** used in Lecture 12.

Task 1: Read in the data

If you haven't retained the data from last week, make a data frame **`cars.df`**. from the cars data. Fit the model using 1/CITY as the response. Call the result **`recip.lm.`**

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

Task 2: outlier identification

Identify outliers:

```
source("http://www.stat.auckland.ac.nz/~lee/330/R330.txt")
par(mfrow=c(2,2))
plot(recip.lm)
par(mfrow=c(3,5))
influence.plots(recip.lm)
```

There are two points that seem influential – 47 and 69, with 47 being the worst. Deleting 47 gives a better set of plots:

```
recip.no47.lm<-lm(I(1/CITY)~ PRICE + WEIGHT + DISP + COMP +
HP + TORQ + TRANS + CYL, subset=-47,data = cars.df)
par(mfrow=c(2,2))
plot(recip.no47.lm)
par(mfrow=c(3,5))
influence.plots(recip.lm)
```

The Cov Ratio and the DFFITS have a few points exceeding the threshold but no real problems.

Task 3: examine the residuals for equal scatter

Evidence of unequal scatter? Funnel effect? Trend in scale-location plot?

Try estimating the regression weights:

```
vars = funnel(recip.no47.lm)
```

In the scale-location plot and the variance function plot there is a quadratic shape but this seems largely driven by edge effects. No real evidence of unequal scatter.

Task 4. Read in the advertising data.

The data set has two variables, **sales** and **advertising** . Each row refers to a month. Make a data frame ad.df with these two variables. We use the advertising variable to make two new variables **spend** (current month's advertising), and **prev.spend** (last month's advertising). We need to adjust the lengths of the vectors so that everything is the same length.

```
spend = ad.df$advertising[-1]   # first ob has no previous
prev.spend =  ad.df$advertising[-36]   # 36 months in all
sales = ad.df$sales[-1] # vectors must have the same length

advertising.lm =lm(sales~spend + prev.spend)
# don't need to specify  a data frame as these are vectors
which R can see
```

Task 5. Plot residuals in various ways:

```
res = residuals(advertising.lm)
par(mfrow=c(1,3)

#  residuals versus previous
n<-length(res)
plot.res<-res[-1]
prev.res<-res[-n]
plot(prev.res,plot.res, xlab="previous residual",
ylab="residual",main="Residual versus previous residual \n
for the advertising data")
abline(coef(lm(plot.res~prev.res)), col="red", lwd=2)

# Time series plot
plot(res, type="b", xlab="Time Sequence", ylab =
"Residual",
main = "Time series plot of residuals for the advertising
data")
abline(h=0, lty=2, lwd=2,col="blue")
```

```
# autocorrelation

acf(res, main ="Correlogram of residuals for the
advertising data")
```

Task 6: Do the DW test

```
rhohat<-cor(plot.res,prev.res)
DW<-2*(1-rhohat)
> DW
[1] 1.109853
```

Is this significant? Use the table in lecture 12. A copy is attached to this sheet.

Task 7: Refit using arima

```
arima(sales,order=c(1,0,0), xreg=cbind(spend,prev.spend))
```

You need worry only about the bits in *italics*, see a time series course for the rest.

Task 8: Simulating the distribution of a statistic

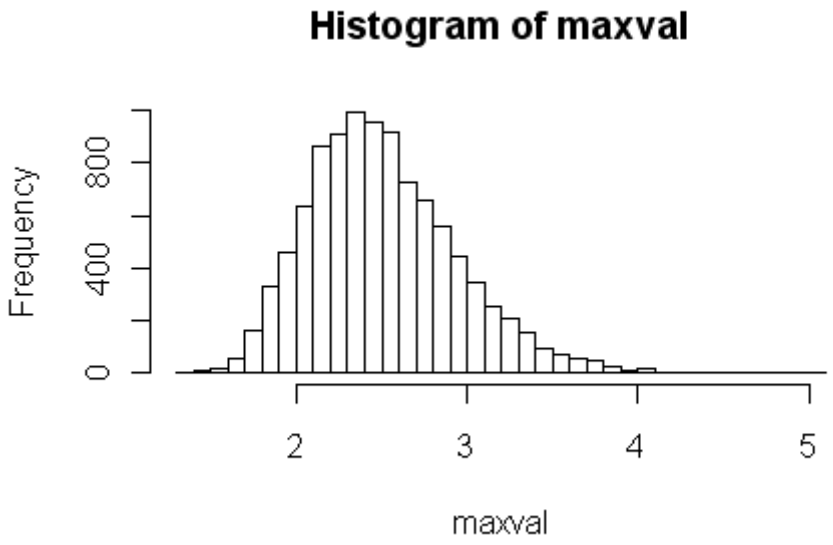
R makes it easy to simulate the distribution of any statistic. For example, suppose we want to study the distribution of biggest (in absolute value) member of a sample of size 50 taken from a normal distribution with mean 0 and sd 1. (This will approximate the distribution of the biggest standardized residual in a regression with 50 observations.).

We can build up the distribution of the maximum value by drawing repeated samples, storing the value of the maximum each time, and plotting the results. The following code does this 10000 times:

```
n=50  # set sample size
N=10000  # set number of repeats
maxval = numeric(N)  # make a place to store the results
# do the simulation
for(i in 1:N){   #  repeat 10000 times
     maxval[i] = max(abs(rnorm(n))) # draw sample, take max
     }

# draw a picture of the results

hist(maxval, nclass=50)
```



About 13% of the 10,000 are more than 3, about 2.5% more than 3.5. What does this say about standardised residuals?

```
> sum(maxval>3)/N
[1] 0.1308

> sum(maxval>3.5)/N
[1] 0.0251
```