STAT2008/6038 Regression Modelling Worksheet 3

This worksheet, which is the last in the series of introductory worksheets on *R*, is an assignment for you to do yourself. Before attempting this worksheet you should have: completed Worksheets 1, 2 and 3. If you are having problems with any of the earlier concepts, you should re-read the appropriate documents and worksheets and practice the concepts which are giving you trouble before attempting this exercise.

Exercise 1.

Produce a plot of the function

$$f(x) = e^{-|x|},$$

for values of x ranging from -3 to 3. This function is sometimes called the *double exponential* function. Make sure to appropriately label your plot and axes.

Exercise 2.

A study of the age and growth characteristics of selected mussel (shellfish) species in two distinct locations in Southwestern Virginia, USA. The data for this study is contained in a file in the group area called `PROB3.8`. This file contains three columns: the first indicates the location (region 1 or region 2) where the data was collected; the second contains the ages of the selected mussels; and the third column gives the weight (in grams) of the selected mussels. Name the columns of the object "location", "age" and "weight" using the `names` command.

On the same set of axes, plot the weight versus age relationship for each of the two locations, connecting the datapoints within each location with lines. Make sure to use distinct symbols and line-types for the data from the two different locations and that you label the plot appropriately. [Hint: look at the help file on lines() for instructions on how to overlay plots.]

Use `lsfit()` to fit a least-squares linear regressions to the data from each of the two locations separately. On the same set of axes, plot the weight versus age relationship for each of the two locations, and superimpose the two regression lines. Again, make sure to use different symbols and

line-types for the two different sets of points and regression lines and also to appropriately label your plot.

Exercise 3.

Recall the difference between a sample median and a sample mean - the median is less sensitive to outliers, but the mean is better for normally-distributed data. To check this out, create a sample of size 50 from a standard normal distribution, and find the mean and median for the data. Now add 100 to the first observation in your sample, to make it an outlier. How are the median and the mean affected by the outlier?

Now, do some simulations to compare the performance of the median and the mean for the standard normal and the Cauchy distributions. The latter typically produces samples with apparent outliers. First, create space to store the results:

```
means.from.normal <- rep(0, 100)
medians.from.normal <- rep(0, 100)
means.from.cauchy <- rep(0, 100)
medians.from.cauchy <- rep(0, 100)
```

Next, create 100 means and medians from samples of size 50 from the standard normal and Cauchy distributions and put them in the vectors created above:

```
   for(i in 1:100) {    x <- rnorm(50)    y <- rcauchy(50)    means.from.normal[i]
<- mean(x)    medians.from.normal[i] <- median(x)    means.from.cauchy[i] <- mean(y)
medians.from.cauchy[i] <- median(y) }
```

Now, investigate the results, using histograms, means, medians and standard deviations.

Exercise 4.

This question asks you to write a new R function to implement a method called the *jackknife*, which can be used to examine the bias of estimation procedures in many situations. We don't want to go into the theory behind the jackknife here, but we would like to write an R function to implement the jackknife bias calculation for the slope estimate from a least-squares linear regression.

We start with a dataset of size n, say, with data points $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$, which are contained in two column vectors `X` and `Y`. The basic idea of the jackknife is to form n new datasets, each one the same as the original dataset except that the i^{th} new dataset is missing the i^{th} point from the original dataset. The new datasets are therefore each of size $n-1$: for example, the first is $(X_2, Y_2), (X_3, Y_3), \ldots, (X_n, Y_n)$ (i.e. the original dataset missing the first data point), the second is $(X_1, Y_1), (X_3, Y_3), \ldots, (X_n, Y_n)$ (i.e. the original data missing the second data point), and so on.

Now, for each of these reduced datasets, calculate the slope of the least-squares linear regression and construct a vector containing the n differences of each of these slopes from the overall slope of the least-squares linear regression on the entire original dataset. In other words, create a vector of length n, whose i^{th} component is the difference between the original slope and the slope of the least-squares linear regression on the reduced dataset [Hint: Recall the function `betachng()` from Worksheet 3]. The jackknife estimate of the bias of the regression slope is then just $n-1$ times the average of this vector of differences.

Write an R function that takes two columns of data, a predictor and a response variable, and returns the jackknife estimate of the bias for a regression slope.

Suppose you wanted to generalize your function to calculate the jackknife bias estimate for a general user-chosen estimator based on a dataset $\mathcal{X}$. If the user-chosen estimator was defined to be a function of the data, say $\hat{\theta}(\mathcal{X})$ which had been coded as the R function `theta()`, briefly describe how you might modify your function to accommodate this generalization.