MATH3012: Statistical Methods –Work Sheet 2

Our task in this lab session will be to go through the commands one by one. The class will wait for everyone to finish the command before going to the next one. Please ask me questions if you get stuck. You can always come back and read and do the materials at your own pace later!

1. **Import the nitric acid data.**

   (a) To do this you just install the MATH3012 files and click on the `nitric` icon with the `S` symbol on my computer window.

   (b) Alternatively, you can follow **File** → **Import Data** → **From File** and then navigate through folders to the directory where you see the file `nitric.txt`. In this case name the columns (by double clicking) `x1`, `x2`, `x3` and `y`.

   (c) The data are 21 successive days of operation of a plant oxidising ammonia to nitric acid. The variables `x1`, `x2` and `x3` are respectively, flow of air to the plant, temperature of the cooling water entering the absorption tower, and concentration of nitric acid in the absorbing liquid. The response `y` is ten times the percentage of ingoing ammonia that is lost as unabsorbed nitric acid (an indirect measure of the yield).

2. **Fit linear models and obtain summary.**

   (a) We fit the linear model: $Y_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_i \qquad i = 1, \ldots, 21.$

   (b) We just need to issue the following two commands:
   `nitric.lm1 <- lm(y~x1+x2+x3, data=nitric)`
   and `summary(nitric.lm1)`

   (c) **Interpretation.** Looks like `x3` is not required in the model because its P-value is 0.344. The other parameters are all significant. The model explained about 91.36% of total variation in the data.

   (d) Can we drop `x3`?

3. **Drop one term.**

   (a) We issue the command `drop1(nitric.lm1)`

   (b) We can drop `x3` since we get a model with lower C_p than the current model.

   (c) This is also confirmed by issuing the command `anova(nitric.lm1)`. The last value 0.344 is the P-value of the likelihood ratio test for testing $H_0 : \beta_3 = 0$.

4. **Update the model by removing x3**

   (a) We can refit by omitting `x3`. S-Plus provides a simpler command.

(b) Issue the command `nitric.lm2 <- update(nitric.lm1, . ~ . -x3)`. This updates the linear model object `nitric.lm1` by removing `x3` from the formula. Now issue `summary(nitric.lm2)`.

5. **Find 95% confidence intervals (CI) for the coefficients β.**

   (a) Dumb answer is to look at the Value and standard error columns of the summary output. From those two obtain `Value` ± `Std. Error` × `critical value`. For 95% confidence interval obtain `qt(0.975, 18)` which is 2.1. Therefore $0.671 \pm 0.127 \times 2.1$ or equivalently (0.404, 0.937) provides the 95% CI for for β_2 in model 2 given by $Y_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$.

   (b) The same can be obtained using a function, see later on.

6. **Diagnostic plots and influential observations.**

   (a) For the selected model we should check the assumptions. We do this by issuing the command `plot(nitric.lm2)`.

   (b) It produces 6 pages of plots. If the model is good, the plot in the first page should not show any systematic pattern, the residuals should be random.

   (c) The plot in page 4 is the normal qqplot. This is used to check normality. If the residuals are normally distributed then all the points should lie in a straight line. In our case observation 21 shows some outlying behavior. Otherwise the plot seems to be ok.

   (d) The Cook's distance plot in page 6 is used to detect outliers. It flags up observation 21 to be the most influential with the highest residual. We can perhaps try fitting by omitting observation 21.

   (e) To remove observation 21 and then refit we just issue `nitric.lm3 <- lm(y~x1+x2, data=nitric, subset=1:20)`. This cleverly uses the subset of the data rows 1 to 20, i.e. omits observation 21. The unbiased estimate for σ^2 is now $2.549^2 = 6.50$ and R^2 increases to 94.6%. The estimates for β_1 and β_2 have changed significantly. In practice such outliers should be investigated and their possible causes found.

7. **How can we predict using the selected model?**

   (a) We can predict using the linear model output very easily using the predict function. Look at the `?predict.lm` help file.

   (b) If we just wanted to predict the means at the x-values which we have used to fit, we just use, `predict(nitric.lm2, se.fit=T, ci.fit=T)`. This gives the predicted means, standard errors and the default 95% CI.

   (c) If we want to predict the actual observations (NOT the means) at the x-values which we have used to fit, we just use, `predict(nitric.lm2, pi.fit=T)`. This gives the predicted observations and the default 95% CI.

(d) If we want to predict at new x-values then we first create a data set which has the column names exactly as the original data set. The response `y` column is not needed. Then put the x-values for which we want to predict as rows. Then use the above commands, but the second argument should be the name of the new data set.

(e) Suppose that we want to predict the `y` value at `x1=60, x2=20` using the model with two covariates `x1` and `x2`. We follow the path **Data** → **Select Data** then click `New Data` and give a name, `pred.new` for example. This brings up a new spreadsheet named `pred.new`. Type 60 and 20 in the first two columns and name those `x1` and `x2`. Then issue the commands `predict(nitric.lm2, pred.new, se.fit=T, ci.fit=T)`. This gives the predicted mean, standard errors and the default 95% CI. The command `predict(nitric.lm2, pi.fit=T)`. This gives the predicted observation and the default 95% CI.

8. **Functions:** Often it is better to put a bunch of S-Plus commands together in a function and execute the resulting object using a single command. There are very many benefits of doing this, we shall find out. For example, a strong motivation for writing functions is that after writing the basic one, we can modify parts of it without altering the other parts.

   Typically, a function takes some input and produces some output. After writing the function it should be first compiled and then executed.

   Because it is easy to make errors when writing functions, and because we often want to modify functions, it is best to write the function in a script file, and then execute the script to create the function. Then the function remains available for modification.

9. To write functions open a script file:

   File → **New** → **Script File**

   At the end of your S-Plus session you should save any Script files for future use. They can then be reopened in future sessions using **File**→**Open**

   We can also write our own functions in S-Plus. For example, there is no intrinsic function for calculating the standard deviation of a set of observations, but we can set up our own using:

10. **Our first function**

```
sd <- function(x)

{

# This calculates the variance of the argument

# first and then takes square root of it.
# The square root is returned
s <- sqrt(var(x))
return(s)
}
```

Then `sd(a)` will give the standard deviation of the data contained in S-Plus object `a`. The function performs the commands between {} and returns the object written on the last line.

11. A clever function to calculate the CI of β's

```
ci <- function(mod,a=0.95)
{
se <- sqrt(diag(summary(mod)$cov.unscaled))
se <- se * summary(mod)$sigma
aa <- (1+a)/2
upper <- coef(mod) + (qt(aa, mod$df.residual) * se)
lower <- coef(mod) - (qt(aa, mod$df.residual) * se)
return(upper, lower)
}
```

Can you work out what this function is returning?

The function takes two arguments, a linear model object and a constant. Note that the constant is given the default value of 0.95, so need not be specified.

12. The butterfly function:

```
butterfly <- function(color = 8) {
theta <- seq(from=0.0, to=24 * pi, len = 2000)
radius <- exp(cos(theta)) - 2 * cos(4 * theta)
radius <- radius + sin(theta/12)^5
x <- radius * sin(theta)
y <- - radius * cos(theta)
plot(x, y, type = "l", axes = F, xlab = "", ylab = "", col = color)
}
```