Creating 3-Dimensional Surface and Contour Plots in R

In the following handout words and symbols in **bold** are R functions and words and symbols in *italics* are entries supplied by the user; underlined words and symbols are optional entries (all current as of version R-2.4.1). Sample texts from an R session are highlighted with gray shading.

To create a three-dimensional plot in R, three objects are necessary: a vector containing values for one independent variable (the \(x\)-axis), a vector containing values for the second independent variable (the \(y\)-axis), and a matrix containing values for the dependent variable (the \(z\)-axis) for all combinations of the two independent variables. For example, let's assume that our model is

\[
Z = 23.86 + 5.525 Y - 2.5725 X - 6.6413 Y^2 - 5.1862 X^2
\]

with values of \(X\) and \(Y\) ranging from -1 to +1. To create a sequence of values for each independent variable we use the command

\[
\text{seq}(\text{from}, \text{to}, \text{by} = \text{value}, \text{length} = \text{value})
\]

where \(\text{from}\) is the first value in the sequence, \(\text{to}\) is the last value in the sequence and \(\text{by}\) or \(\text{length}\) (only one can be used) determines the form of the sequence. The command \(\text{by}\) gives the difference between successive values (the step-size) and the command \(\text{length}\) gives the number of values in the sequence. Two examples are shown here:

\[
\text{seq}(0, 6, \text{by} = 2)
\]

[1] 0 2 4 6

\[
\text{seq}(0, 6, \text{length} = 3)
\]

[1] 0 3 6

Omitting the commands \(\text{by}\) and \(\text{length}\) sets the step-size to 1 by default.

\[
\text{seq}(0, 6)
\]

[1] 0 1 2 3 4 5 6

Typing the command “\(\text{by} = \)” is optional. If you provide three values to \(\text{seq}\(\), the third value is taken to be the step-size.

\[
\text{seq}(0, 6, 2)
\]

[1] 0 2 4 6
A useful plot should contain sufficient levels for the independent variables to show the structure of the three-dimensional surface. The right balance often requires some trial-and-error experimentation; this example uses increments of 0.1, giving 21 levels for each independent variable.

```R
> x = seq(-1, 1, 0.1)
> y = seq(-1, 1, 0.1)
```

To create the matrix of values for the dependent variable, we use the command for the outer product of two vectors, which evaluates all possible combinations of each vector’s values; the command takes the form

```
outer(dependent variable 1, dependent variable 2, function)
```

where `function` is a user-defined function containing the model. For our example, we create the following function

```R
> model = function (a, b){
+ 23.86+5.525*b-2.5725*a-6.6413*b^2-5.1862*a^2
+}
```

and then create the matrix for the dependent variable’s values

```R
> z=outer(x, y ,model); z

[3,] 10.432532 12.246879 13.928400 15.477095 16.89296
[4,] 10.953212 12.767559 14.449080 15.997775 17.41364
... omitted to save space...

[,]21
[1,] 20.13000
```

which contains 21 columns (corresponding to `x`) and 21 rows (corresponding to `y`).

The command to create a three-dimensional surface plots is

```
persp(dep. var. 1,dep. var. 2, ind. var., theta= 0, phi=15, ticktype=“simple“)
```
The optional commands *theta* and *phi* allow the plot to be rotated about the z-axis and tilted relative to the x-y plane, respectively, allowing the user to find the best view of the response surface. The optional command *ticktype = “detailed”* adds a numerical scale to the three axes in place of an arrow showing the direction in which the scale increases, which is the default for *ticktype = “simple”*.

```
> persp(x,y,z,theta=30,phi=30,ticktype="detailed")
```

To create a contour plot, the command is

```
contour(dep. var. 1,dep. var. 2, ind. var., nlevels = 10)
```

where the optional command *nlevels* indicates the number of contour lines to be drawn (which defaults to 10).
> contour(a,b,data,nlevels=12)