Creating 3D Surface plots in R

Ralph Mansson

Introduction

Surface plots are three dimensional representations of data that has specific useful applications, including displaying terrains for geographical data or the predictions from a model fitted in two or more dimensions.

To illustrate this type of graph we will consider surface elevation data that is available in the `geoR` package via the **R** software. The data set is called `elevation` and is a record of elevation height in feet (stored as multiples of ten) for a grid of x and y coordinates (recorded as multiples of 50 feet). To access this data we load the `geoR` pacakage and then use the `data` function:

```
require(geoR)
data(elevation)
```

We first make a copy of this data and save it in a data frame for use in creating the surface plot so that we can run some additional calculations on the heights for the fitted surface:

```
elevation.df = data.frame(x = 50 *
elevation$coords[,"x"], y = 50 *
elevation$coords[,"y"], z = 10 *
elevation$data)
```

3d Surface plots

To create a surface plot we can fit a local trend surface to the elevation data via the `loess` function - we make use of a quadratic surface is estimated using weighted least squares:

```
elevation.loess = loess(z ~ x*y, data =
elevation.df, degree = 2, span = 0.25)
```

The next step is to create an array of x and y co-ordinates covering the region of interest and then to calculate the height of the fitted local trend surface at these points. These values will then be used by the plotting functions to create the level plots.

```
elevation.fit = expand.grid(list(x = seq(10,
300, 1), y = seq(10, 300, 1)))
z = predict(elevation.loess, newdata =
elevation.fit)
elevation.fit$Height = as.numeric(z)
```

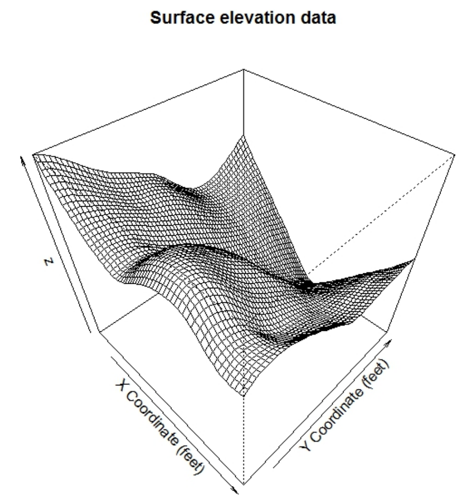
The function `expand.grid` creates a data frame based on the ranges of x and y specified above. The `predict` function then uses the fitted model object to estimate the height of the surface and this is saved in an object `z` as the different graph functions need the data in different formats. The fitted surface heights are converted to a numeric vector and attached as an additional column to the object that was used to create the predictions.

Base Graphics

The function `persp` is the **base** graphics function for creating a wireframe surface plot. The `persp` function requires a list of x and y values covering the grid of vertical values. The heights for the display are specified as a table of values which we saved previously as the object `z` during the calculations when the local trend surface model was fitted to the data. The text on the axis labels are specified by the `xlab` and `ylab` function arguments and the `main` argument determines the overall title for the graph.

```
persp(seq(10, 300, 5), seq(10, 300, 5), z,
phi = 45, theta = 45, xlab = "X Coordinate
(feet)", ylab = "Y Coordinate (feet)", main
= "Surface elevation data")
```

The function arguments `phi` and `theta` are used to rotate the angle which the surface is viewed from and it is best to experiment with these until arriving on a suitable viewing angle.



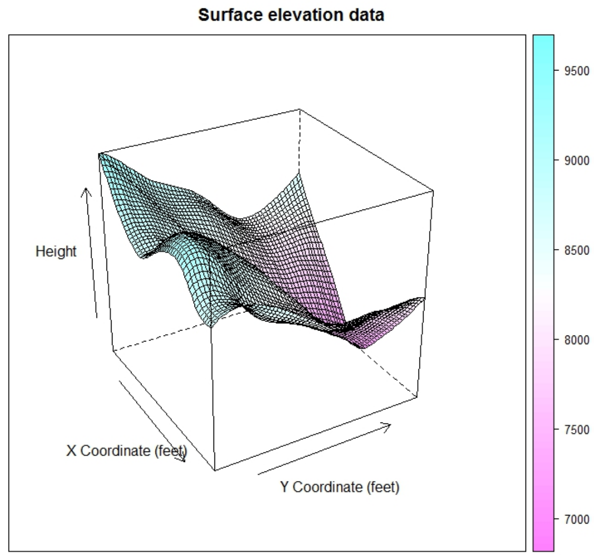
The graph is reasonable simple with a black and white surface mesh which allows us to see the variation in height across the region of interest in the data set.

Lattice Graphics

The **lattice** graphics package has a function `wireframe` and we use the data in the object `elevation.fit` to create the graph. We use the formula interface to specify first the z axis data (the heights) followed by the two variables specifying the x and y axis coordinates for the data.

```
wireframe(Height ~ x*y, data =
elevation.fit, xlab = "X Coordinate (feet)",
ylab = "Y Coordinate (feet)", main = "Surface
elevation data", drape = TRUE, colorkey =
TRUE, screen = list(z = -60, x = -60))
```

The axes labels and title are specified in the same way as the **base** graphics and a colour key is added using the `colorkey` function argument.



The surface produced by the `wireframe` function is similar to the `persp` function with the main difference between the colours used on the surface.

ggplot2 Graphics

There are currently no facilities in the **ggplot2** package for creating three dimensional surface plots.