Lattice Package

Linda Li

10/26/2017

Introduction

When it comes to data visualizations, R is always a good choice as it has various practical tools, like ggplot2 and the base R graphics, to accomplish the goal. Today, in this post, I will talk about another package called lattice that is designed to deal with multivariable data. I will give a brief overview for how to utilize different functions in the package.

Background

Lattice package is written by Deepayan Sarkar. It is an upgraded version of the base R graphics, as it provides better defaults and is capable of easily displaying multivariate relationships. The package is able to create trellis graphs, which are graphs that display a variable or the realationship between variables, conditioned on one or more other variables.

Graph Descriptions and Corresponding Codes:

Univariate:

- Bar plots: barchart()
- Box and Whisker plots: bwplot()
- Kernel density plots: densityplot()
- Dot plots: dotplot()
- Histograms: histogram()
- Quantile plots against mathematical distributions: qqmath()
- 1-dimensional scatterplot: stripplot()

Bivariate:

- q-q plot for comparing two distributions: qq()
- Scatter plot: xyplot()

Trivariate:

- Level plots: levelplot()
- Contour plots: contourplot()
- 3-D scatter plots: cloud()
- 3-D surfaces: wireframe()

Hypervariate:

- Scatterplot matrix: splom()
- Parallel coordinate plots: parallel()

Miscellaneious:

- Residual and fitted value plot: rfs()
- Tukey Mean-Difference plot: tmd()

Load Lattice Package

```
library(lattice)
```

Data Set

The following data is collected and calculated with the source www.basketball-reference.com. Efficiency is calculated by efficiency = (points + rebounds + assists + steals + blocks - missed_field_goals - missed_free_throws - turnovers)/games_played. Salary is the total amount of salaries in dollars for each team. Points is the sum of all points that the team has gained in games, and expereience is the total amount of years of playing for each team.

```
dat <- read.csv(file = "../data/nba2017-teams.csv")
dat1 <- read.csv(file = "../data/nba2017-roster.csv")
dat
```

```
##     X team salary points efficiency experience
## 1   1  ATL  90.89   7759   140.3269         93
## 2   2  BOS  91.92   8857   148.2525         63
## 3   3  BRK  65.45   7495   147.7823         52
## 4   4  CHI  92.08   7349   139.1025         58
## 5   5  CHO 100.25   8127   145.2994         66
## 6   6  CLE 125.79   8605   177.8585        128
## 7   7  DAL  92.10   6910   148.2243         62
## 8   8  DEN  78.38   8769   167.3595         74
## 9   9  DET 103.07   8309   136.3762         55
## 10 10  GSW  98.69   9473   172.3916        101
## 11 11  HOU  84.66   8469   155.1031         56
## 12 12  IND  84.57   7918   135.0697         84
## 13 13  LAC 114.78   8911   147.1242        124
## 14 14  LAL  86.27   7354   143.9768         66
## 15 15  MEM 108.34   7995   140.9707         83
## 16 16  MIA  72.78   8312   151.9902         63
## 17 17  MIL  90.27   8390   153.2588         64
## 18 18  MIN  59.38   8634   144.8383         48
## 19 19  NOP  90.63   6563   164.2521         55
## 20 20  NYK  97.01   8060   143.9033         59
## 21 21  OKC  86.98   8104   146.8680         55
## 22 22  ORL 102.41   7408   125.1406         57
## 23 23  PHI  55.78   7116   164.0916         34
## 24 24  PHO  72.53   8399   144.3065         68
## 25 25  POR 103.03   8254   143.7321         43
## 26 26  SAC  88.19   6348   148.3954         68
## 27 27  SAS 104.69   8578   146.6236         99
## 28 28  TOR 108.46   8166   158.7658         57
## 29 29  UTA  80.32   8258   145.8193         71
## 30 30  WAS  98.78   8163   143.0117         56
```

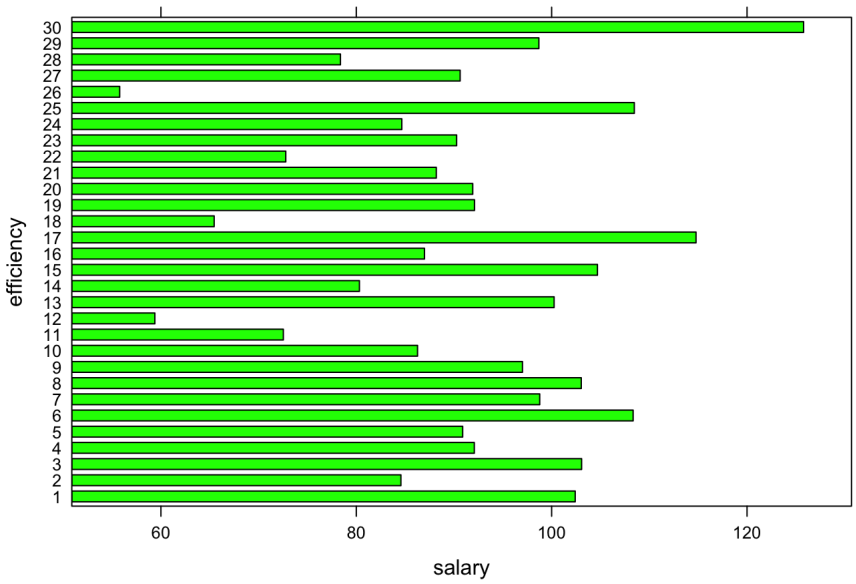
dat is a data set that contains information about efficiency, salary, points, and experience for each tem. dat1 is a data set that contains position, height, weight, age, experience and salary for each player in each of the teams.

Use of Graphs

In the interest of keeping this post concise, I will not show all the graphs, but just some of the most common ones.

Bar Plots

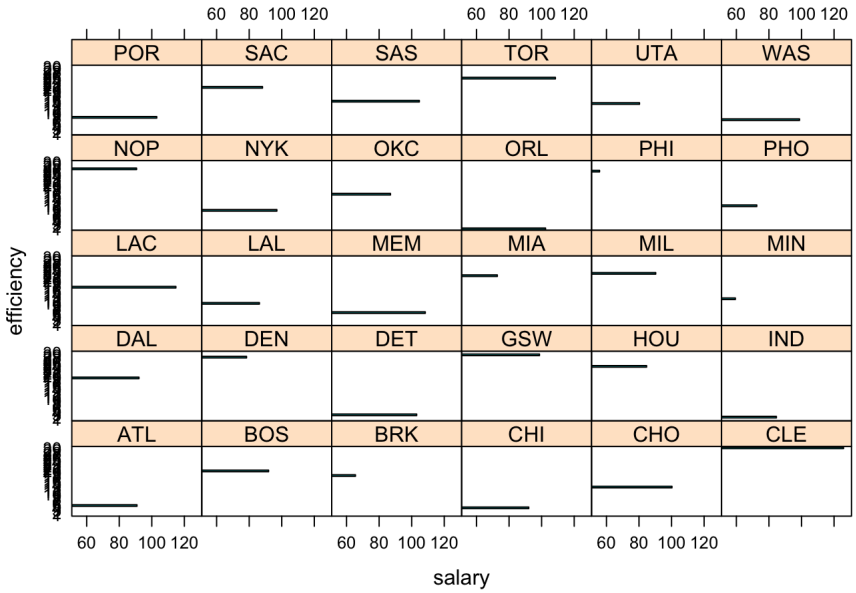
```
barchart(efficiency~salary, data = dat, col = "green")
```



The barchart gives information about salary with regard to efficiency.

Bar Chart Grouping

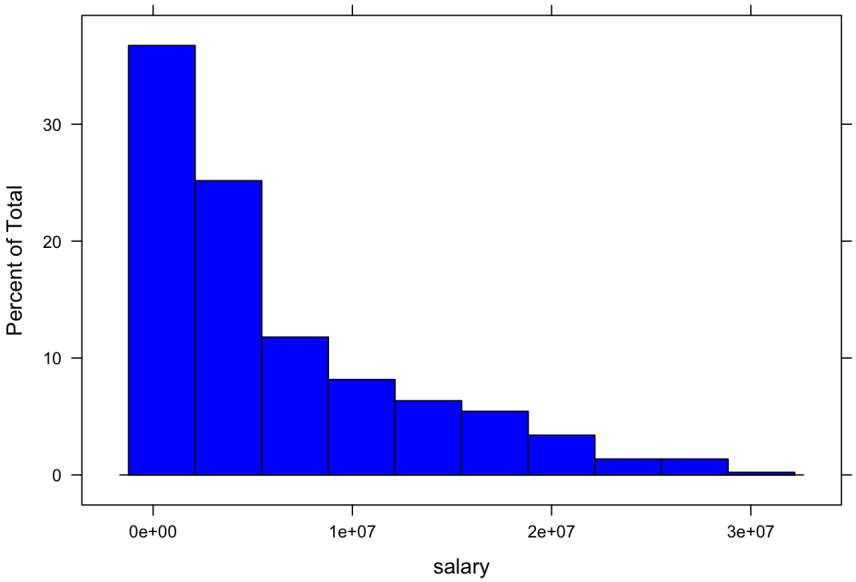
```
barchart(efficiency~salary|factor(team), data = dat)
```



This graph gives the same information as the previous graph, but it is divided by teams.

Histogram

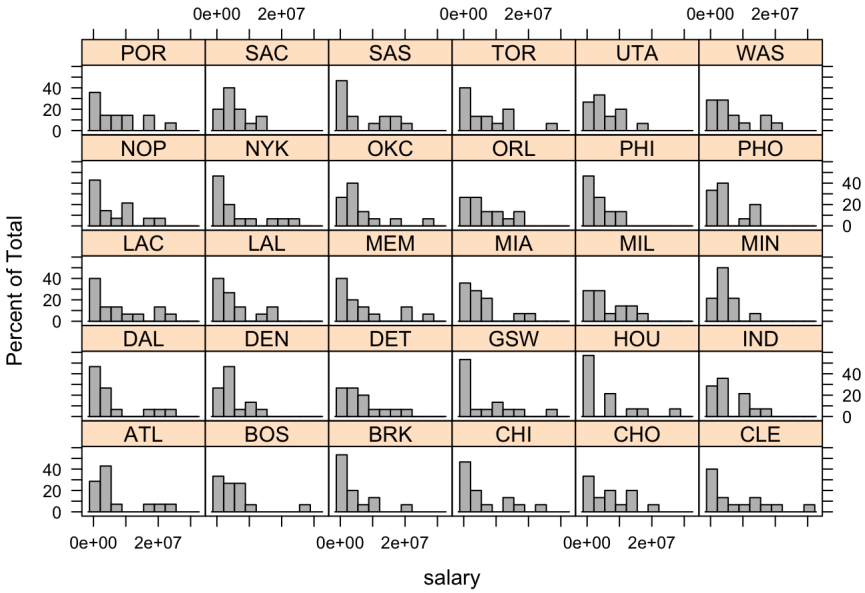
```
# Distribution of the salary for all
histogram(~ salary, data = dat1, col = "blue")
```



The lattice histogram comes in handy if you are trying to find how values are distributed for a specific date set. It is achievable with ggplot, but would be more complicated, as you need to redefine y component.

Historgram (grouping)

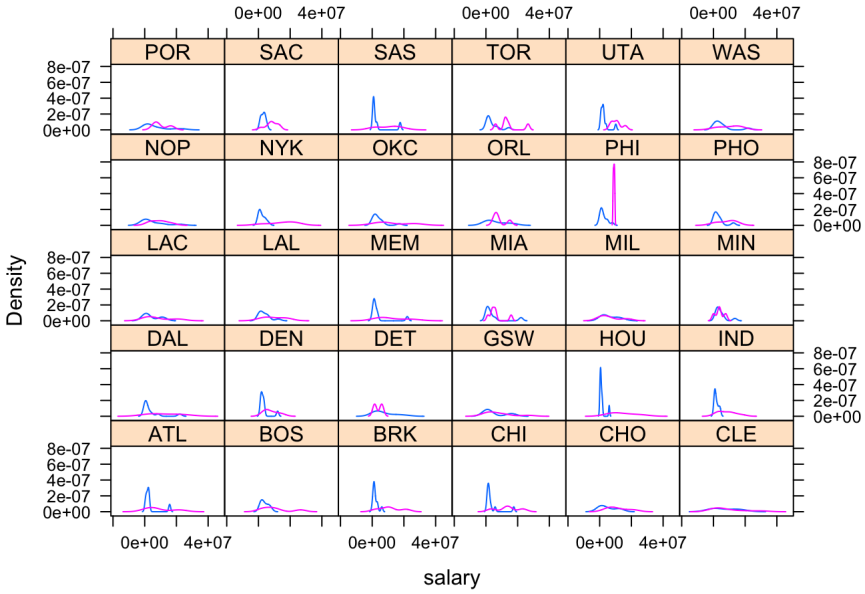
```
# General formula for conditioning: (conditioning variable | primary variable)
histogram(~salary | factor(team), groups = experience, data = dat1, col = "grey")
```



The graphs show the distribution salaries for each player for each team. You can also achieve this goal with ggplot. However, it is a lot more complicated than it is for lattice, because for ggplot, you would need to use facet() to group different graphs.

Kernel Density Plot

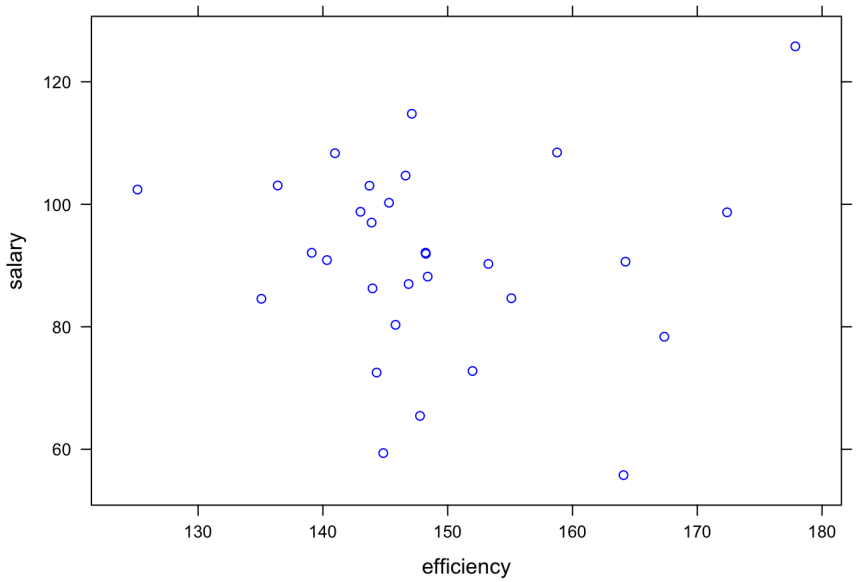
```
densityplot(~salary | factor(team), groups = experience>5, data = dat1, plot.points = FALSE)
```



The density function gives different density functions for salary for different players in each team. There are two densities in each plot. The pink one is for those who have more than five years of experiences, and the blue one is for those who have experiences less than 5 years. Similar to the previous graph, this graph can also be acheived with ggplot() function + facet(), but it would be a lot more compliated and thus error-prone compare to lattice.

Scatter Plot

```
xyplot(salary~efficiency, data = dat, col= "blue")
```

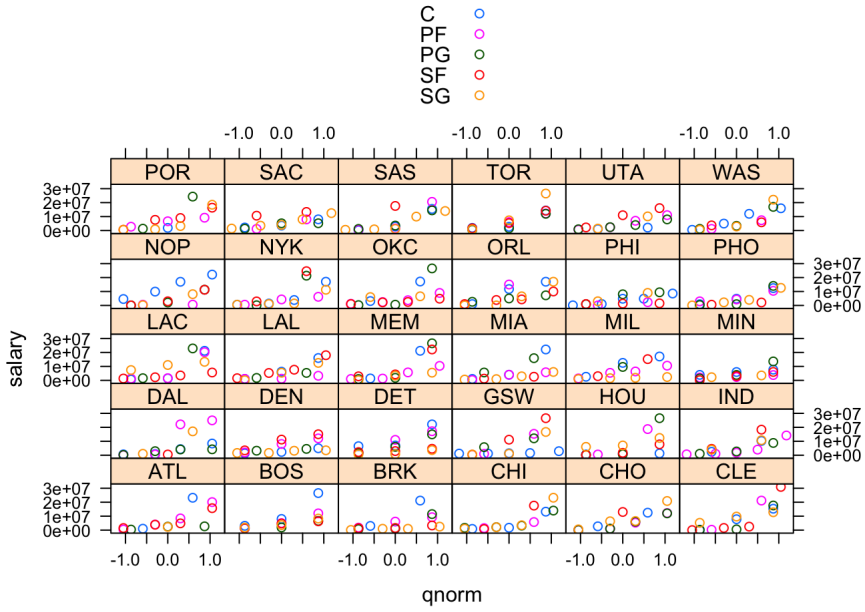


This graph gives the information about the salary with regard to efficianecy. It does not have a general pattern, all points are very scattered.

Q-Q plot

This is one of the unique but also useful tools that lattice has. It is used to determine if two data sets com from populations with a common distribution. It is a plot of quantiles of the first data set against the quantiles of the second data set. However, with my data set I can not give a good desmonstration. I will provide the graph with my code:

```
qqmath(~salary | factor(team), data = dat1, groups = position, auto.key = TRUE)
```



This graph shows the amount of salary for each position in each team. There is no general pattern in this graph due to data selection reasons.

1-D scatter plot

stripplot produces one dimensional scatter plots (or dot plots) of the given data. These plots are a good alternative to boxplots when sample sizes are small.

```
# Create a smaller sample size
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

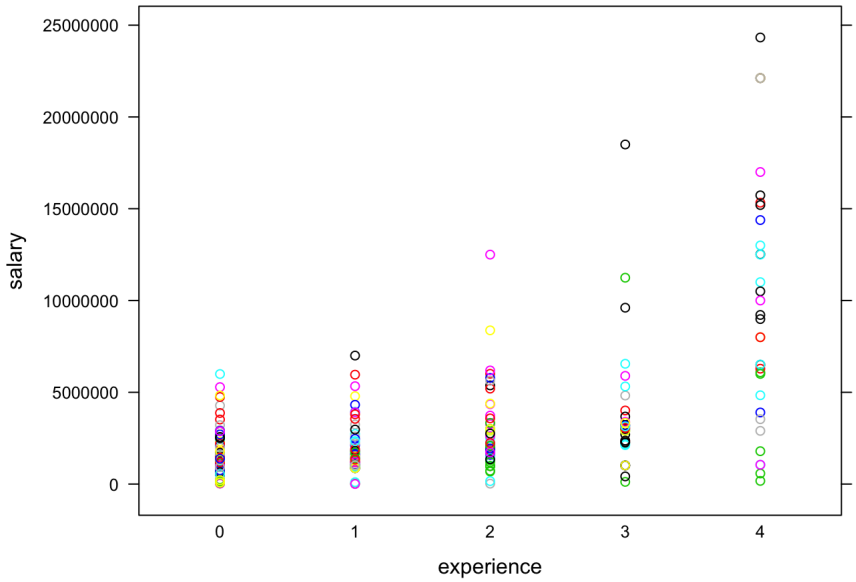
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
dat2 <- filter(dat1, experience <5)
```

dat2 is a sub data set of dat1 with all players that have experiences less than 5 years.

```
# Creating a strip plot
stripplot(salary ~ factor(experience), data = dat2, xlab = "experience", col = dat2$team)
```

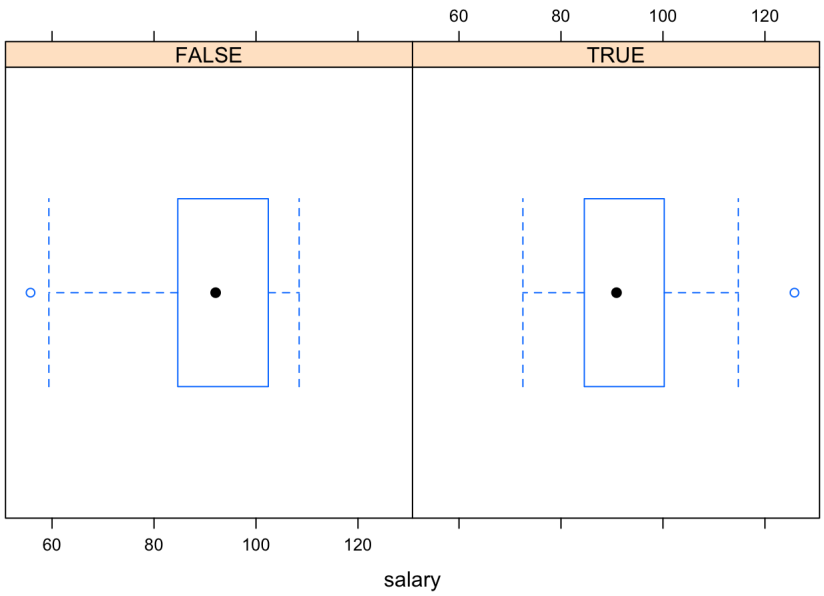


This graph gives me information about the salaries for each player that has less than 5 years of experience in each team (distinguished by colors).

Box and whisker plot

This plot is used to give a visualization for describing numerical data through their quantiles.

```
bwplot(~ salary | experience>60, data = dat, groups = team)
```



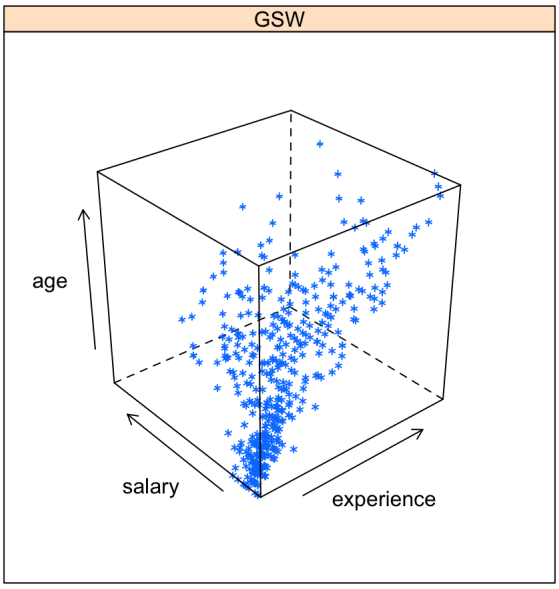
The graph decently describes the distribution of the relationship between salary and experience, as there is barely any outlier in the graph. When experience < 60, more salaries are within the lower than the average. When experience > 60, slightly more salaries are above average than below.

Trivariate

Trivariate graphs are used to dipict the relationship between three different variables. Trivariate functions generate three dimentional graphs as it is more dicrect for us to observe the trends.

3-D scatter plot

```
# The axis are defined as follwing: z~x*y
cloud(age~experience*salary | factor("GSW"), data = dat1)
```

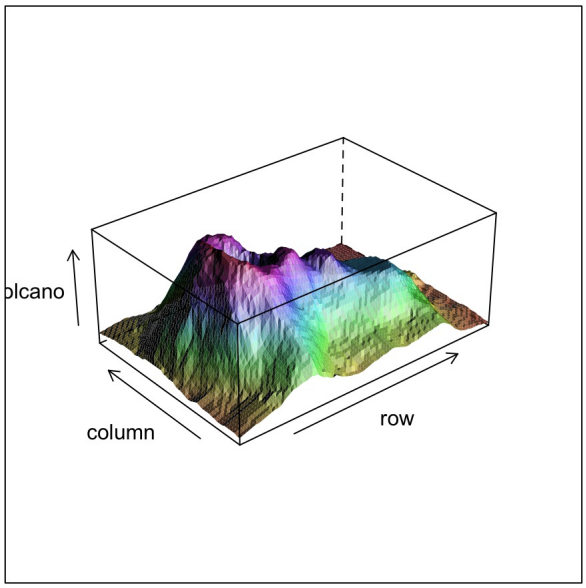


With cloud(), there are a lot of functions you can use to adjust the plot according to your preference. For example, changing limits for axises using xlim(), ylim(), and zlim(). You can also change the label of each axis with xlab(), ylab() and zlab(). Moreover, you can use scales() to change the style of the axis, ususally from arrows to tick marks. Screen() can also be used to change the angel of displaying the 3-D plot.

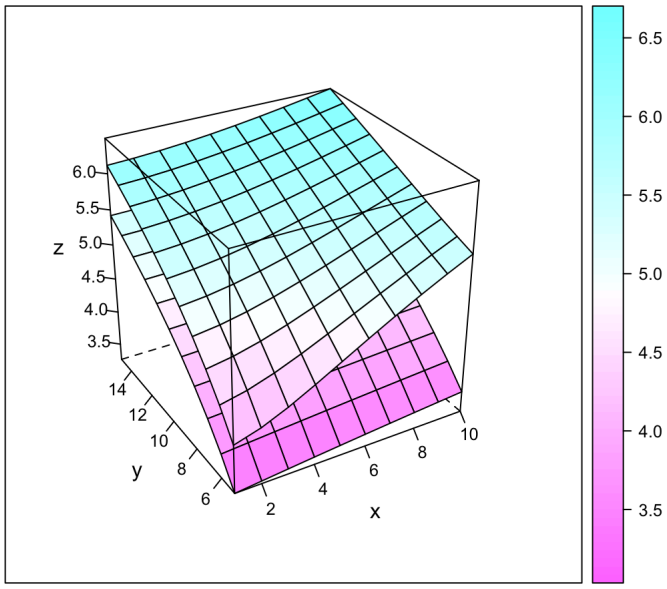
3-D surfaces

The wireframe() is a type of graph that is used to display a surface. For example, when analysising geogreaphic data, this graph could be very useful. It could also be used to display a fitted model with more than one explanatory variable. Depending on what data set you are using, you will get different shapes.

```
#Using the data set "volcano" that came with R
wireframe(volcano, shade = TRUE,
aspect = c(61/87, 0.4),
light.source = c(10,0,10))
```



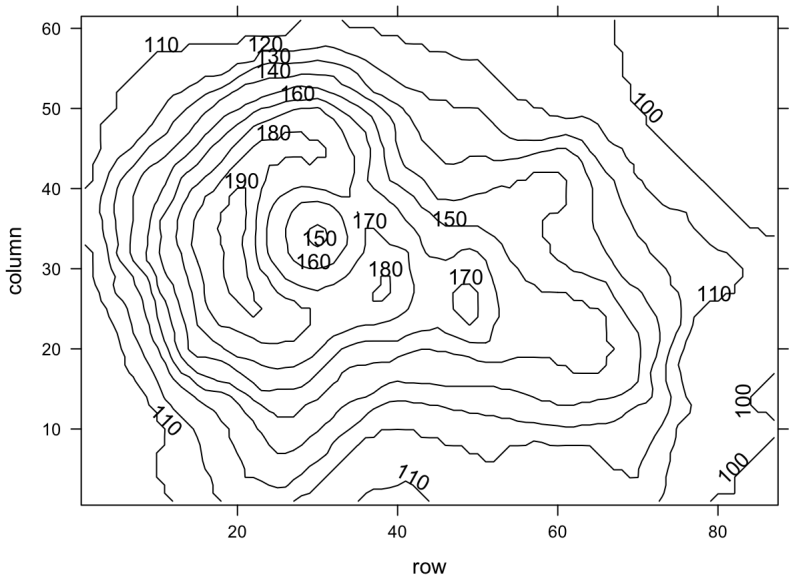

```
# Another Example
g <- expand.grid(x = 1:10, y = 5:15, gr = 1:2)
g$z <- log((g$x^g$gr + g$y^2) * g$gr)
wireframe(z ~ x * y, data = g, groups = gr,
scales = list(arrows = FALSE),
drape = TRUE, colorkey = TRUE,screen = list(z = 30, x = -60))
```



With the same volcano data set, we can also plot contour and level plots.

Contour Plot

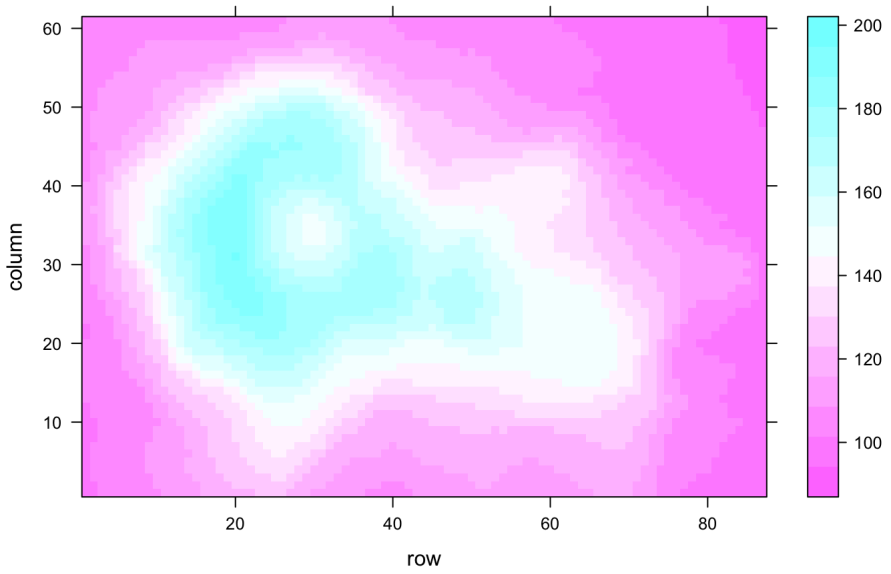
```
contourplot(volcano, cuts = 16)
```



This graph shows the altitudes for the volcano at different levels. The closer the rings, the steeper the slope.

Level Plot

```
levelplot(volcano)
```



This graph gave the same information as the contour graph but in a different form. It uses color to indicate the altitude at different points.

Pros and Cons for Lattice

Pros

- Most useful for conditing types of plots: Looking at how y changes with x across levels of z
- Thinks like spacing set automatically because the entire plot is specified at once
- Good for putting multiple plots on a screen

Cons

- Sometimes hard to specify an entire plot in a single function call
- Annotations in plots are not intuitive
- Can not “add” to the plot once it is created

Summary

Not all data analysing/visualiztion tools are flawless, you need to choose the one that fits your data analysis the most in order to get the best result from different tools. If you need to analyze a set of data wit regard to two factors, then lattice would be a good choice, since it is fairly easy to display multiple graphs for different groups of data compare to ggplot.

Sources:

- https://stackoverflow.com/questions/2759556/r-what-are-the-pros-and-cons-of-using-lattice-versus-ggplot2
- https://www.dezyre.com/data-science-in-r-programming-tutorial/data-visualizations-tools-r
- https://www.statmethods.net/advgraphs/trellis.html
- https://cran.r-project.org/web/packages/lattice/lattice.pdf
- http://astrostatistics.psu.edu/su07/R/html/lattice/html/Lattice.html
- https://stat.ethz.ch/R-manual/R-devel/library/lattice/html/Lattice.html
- https://github.com/rdpeng/CourseraLectures/blob/master/ggplot2_part1.pptx