

Visualização de dados Lattice e ggplot2 (uma introdução)

David Henriques da Matta e Omar Abbara

Universidade Estadual de Campinas
R Campinas User Group

16 de setembro, 2015

- O objetivo desta apresentação é dar início ao estudo do pacote ggplot2 do software R.

- O objetivo desta apresentação é dar início ao estudo do pacote ggplot2 do software R.
- Dentre os diversos pacotes disponíveis no software R para análise gráfica, podemos citar o Lattice como um importante antecessor do ggplot2.

- O objetivo desta apresentação é dar início ao estudo do pacote ggplot2 do software R.
- Dentre os diversos pacotes disponíveis no software R para análise gráfica, podemos citar o Lattice como um importante antecessor do ggplot2.
- Antes de iniciarmos os estudos ggplot2 introduziremos portanto algumas formas e características da análise gráfica via Lattice ("Lattice: Multivariate Data Visualization With R").

- O objetivo desta apresentação é dar início ao estudo do pacote ggplot2 do software R.
- Dentre os diversos pacotes disponíveis no software R para análise gráfica, podemos citar o Lattice como um importante antecessor do ggplot2.
- Antes de iniciarmos os estudos ggplot2 introduziremos portanto algumas formas e características da análise gráfica via Lattice ("Lattice: Multivariate Data Visualization With R").
- Introduziremos também algumas análises gráficas pontuais de outros pacotes que julgarmos interessantes.

Lattice: Carregando Dados

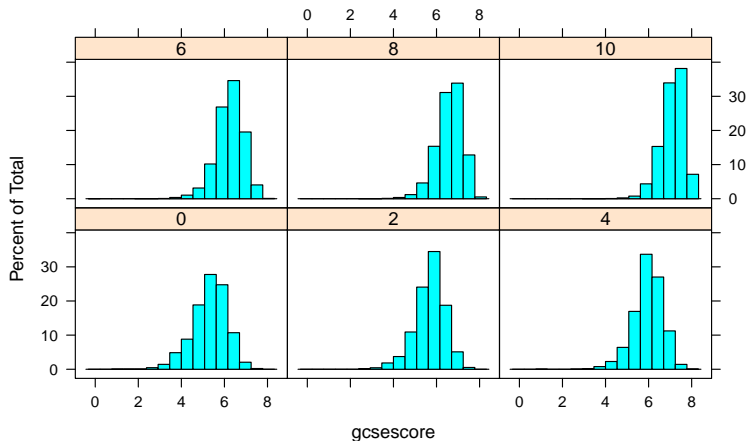
```
library("lattice")
data(Chem97, package = "mlmRev")
data(barley, package = "lattice")
data(postdoc, package = "latticeExtra")
data(Oats, package = "MEMSS")
data(Cars93, package = "MASS")

xtabs(~ score, data=Chem97)
```

```
## score
##      0      2      4      6      8     10
## 3688 3627 4619 5739 6668 6681
```

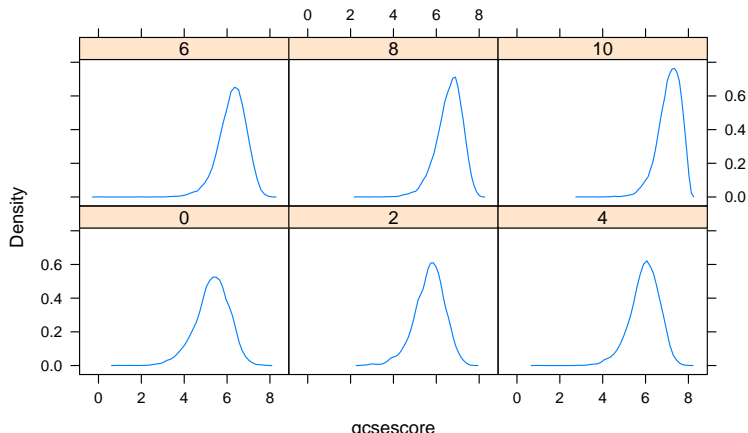
Lattice: histogram

```
histogram(~gcsescore|factor(score),data=Chem97,aspect =0.7)
```



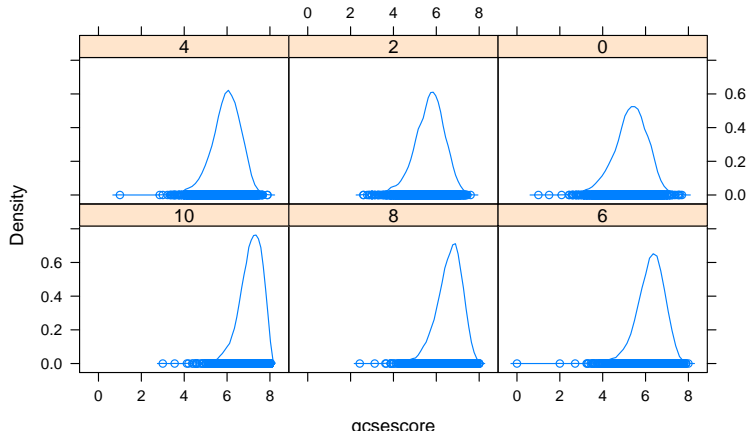
Lattice: densityplot

```
densityplot(~gcsescore|factor(score),data=Chem97,  
plot.points=FALSE,layout=c(3,2),aspect =0.7)
```



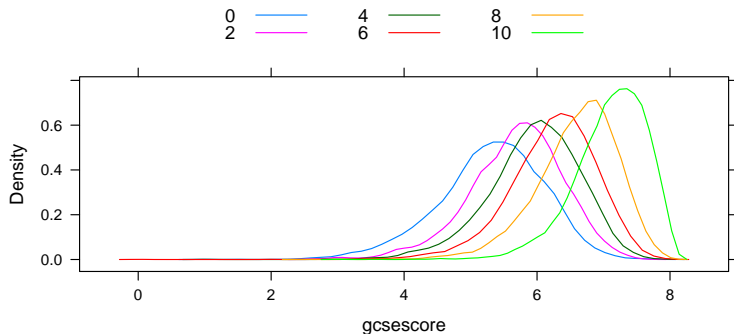
Lattice: densityplot

```
densityplot(~gcsescore|factor(score, c("10", "8", "6", "4",  
"2", "0")), data=Chem97, plot.points=TRUE, aspect =0.7)
```



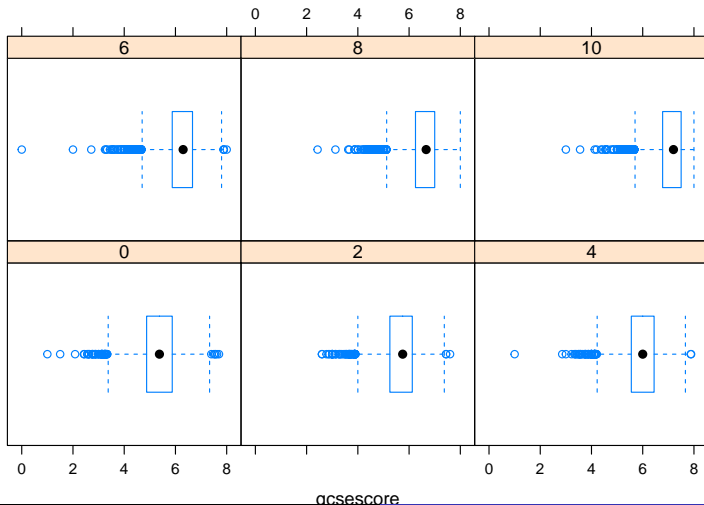
Lattice: densityplot

```
densityplot(~gcsescore,data=Chem97,groups=score,  
plot.points=FALSE,auto.key=list(columns=3))
```



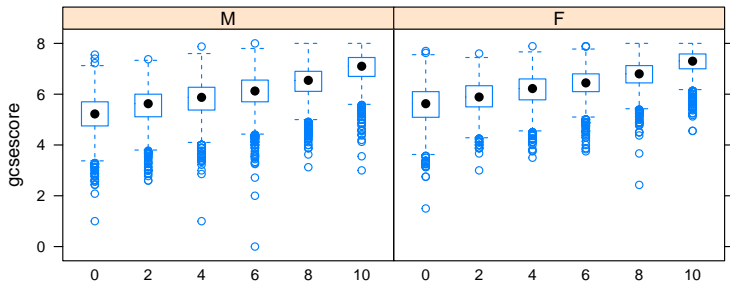
Lattice: bwplot

```
bwplot(~gcsescore | factor(score), data=Chem97)
```



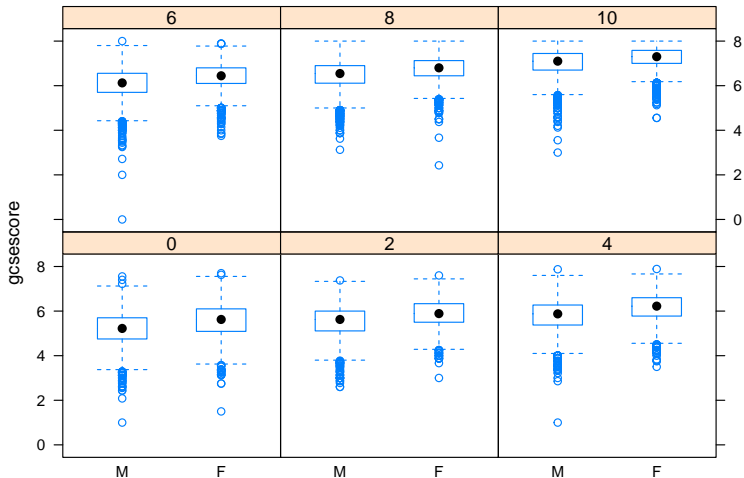
Lattice: bwplot

```
bwplot(gcsescore~factor(score) | factor(gender), data=Chem97, a
```



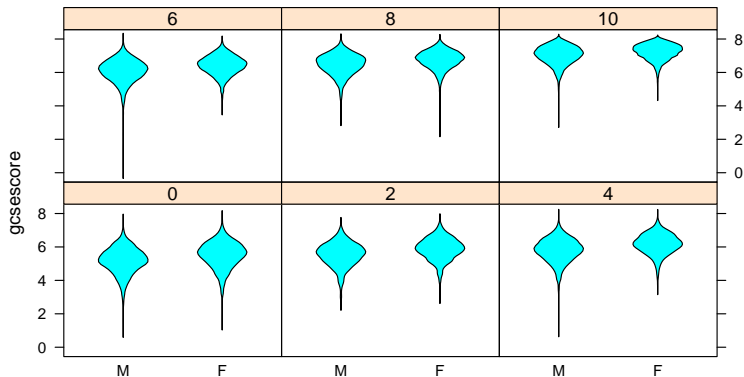
Lattice: bwplot

```
bwplot(gcsescore~factor(gender) | factor(score), data=Chem97)
```



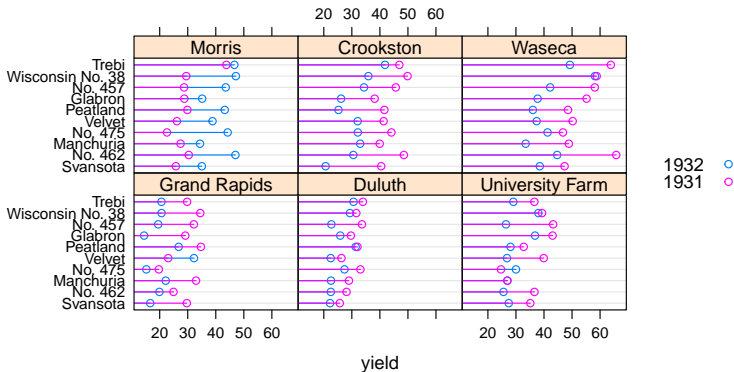
Lattice: bwplot

```
bwplot(gcsescore~factor(gender) | factor(score), data=Chem97,  
panel = panel.violin, aspect =0.7)
```



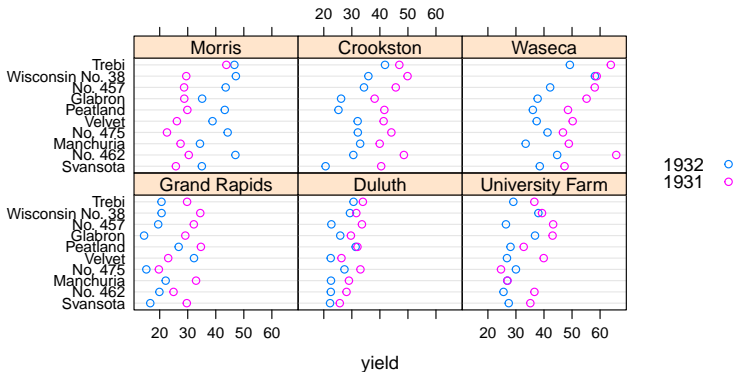
Lattice: dotplot

```
dotplot(variety~yield|site,barley,layout=c(3,2),aspect=0.7,
groups=year,type=c("p","h"),auto.key=list(space="right"))
```



Lattice: dotplot

```
dotplot(variety~yield|site,barley,layout=c(3,2),aspect=0.7,
groups=year, auto.key =list(space = "right"))
```



Lattice: dotplot

```
VADeaths
```

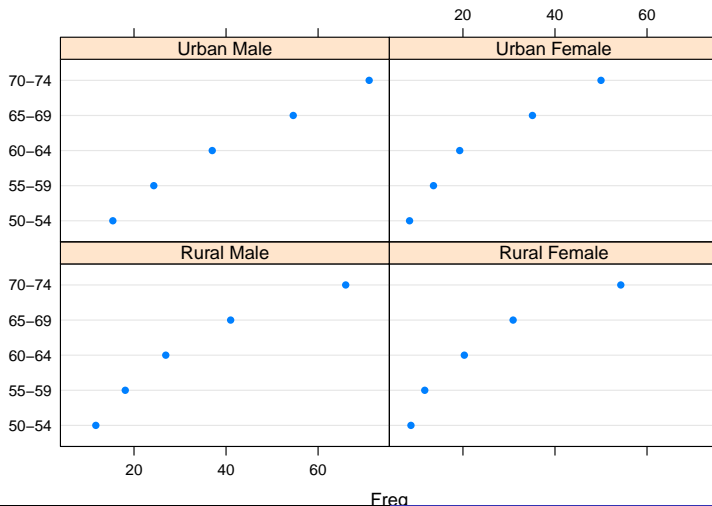
```
##           Rural Male Rural Female Urban Male Urban Female
## 50-54           11.7           8.7           15.4           8.4
## 55-59           18.1           11.7           24.3           13.6
## 60-64           26.9           20.3           37.0           19.3
## 65-69           41.0           30.9           54.6           35.1
## 70-74           66.0           54.3           71.1           50.0
```

```
class(VADeaths)
```

```
## [1] "matrix"
```

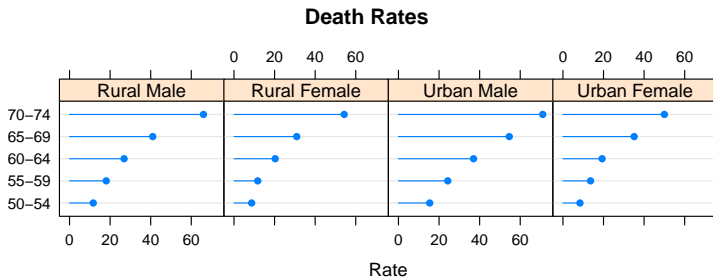
Lattice: dotplot

```
dotplot(VADeaths, groups = FALSE)
```



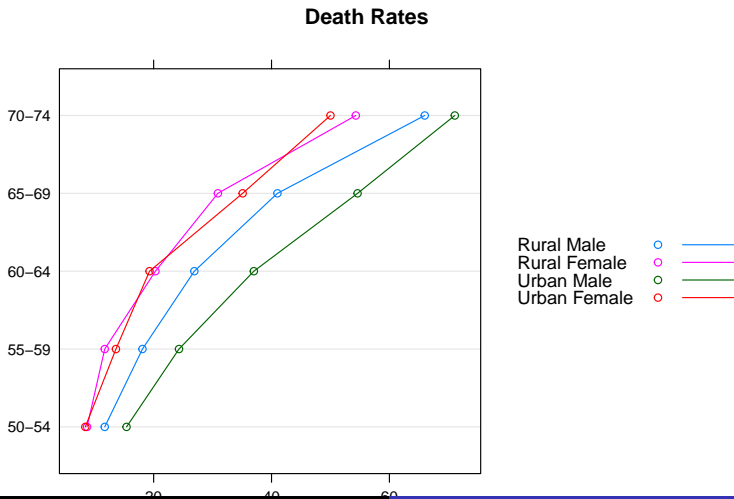
Lattice: dotplot

```
dotplot(VADeaths, groups=FALSE, layout=c(4,1), aspect=0.7,  
origin=0, type=c("p", "h"), main="Death Rates", xlab="Rate")
```



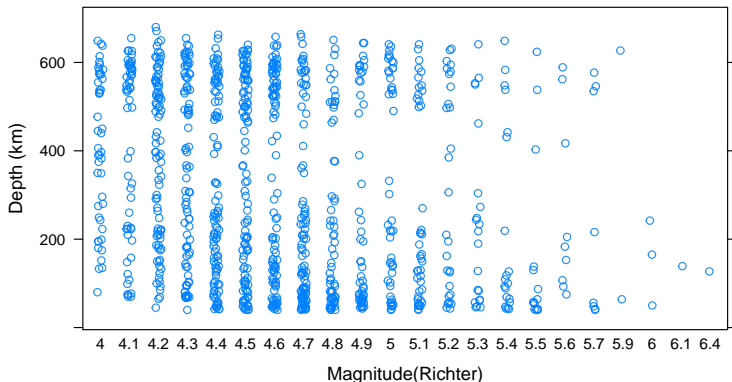
Lattice: dotplot

```
dotplot(VADeaths,type = "o",auto.key=list(lines=TRUE,aspect=0.5,space="right"),main = "Death Rates",xlab="Rate")
```



Lattice: stripplot

```
stripplot(depth~factor(mag), quakes, jitter.data=TRUE,  
alpha=0.9, xlab="Magnitude(Richter)", ylab="Depth (km)")
```



Lattice: stripplot

```
Stripplot<-stripplot(sqrt(abs(residuals(lm(yield~variety+
year+site))))~site,data=barley,groups=year,jitter.data=TRUE,
auto.key=list(points=TRUE,lines=TRUE,columns=2),
type=c("p", "a"), fun=mean,
ylab = expression(abs("Residual Barley Yield")^{1 / 2}))
```

Lattice: stripplot

Stripplot

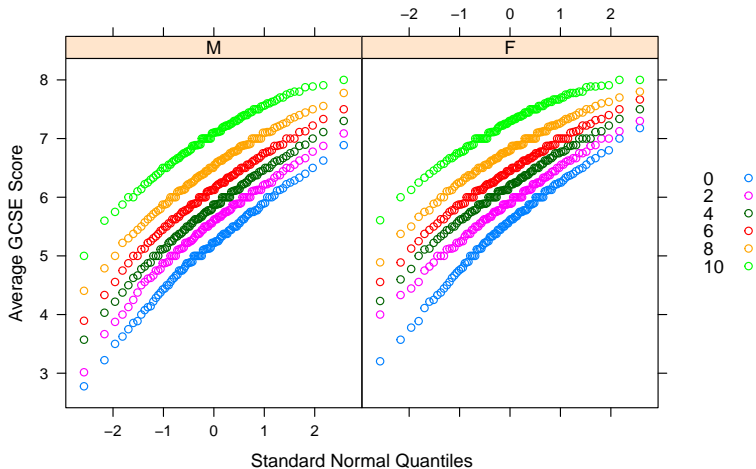


Lattice: stripplot

```
Qqplot<-qqmath(~ gcsescore | gender, Chem97, groups = score  
f.value = ppoints(100), auto.key = list(space = "right"),  
xlab = "Standard Normal Quantiles",  
ylab = "Average GCSE Score")
```

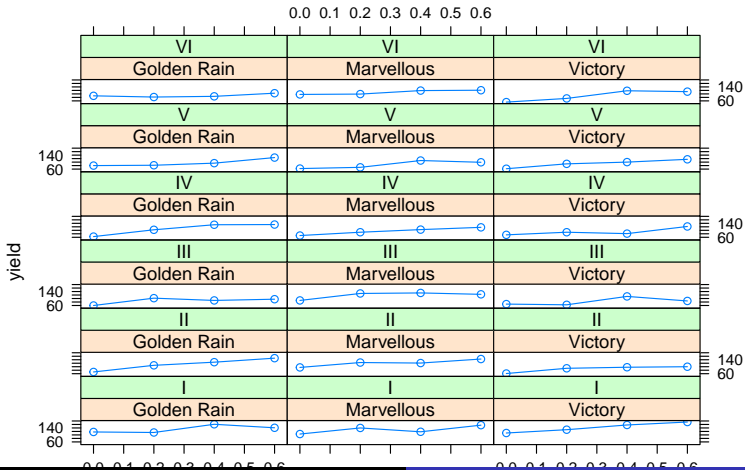

Lattice: stripplot

Qqplot



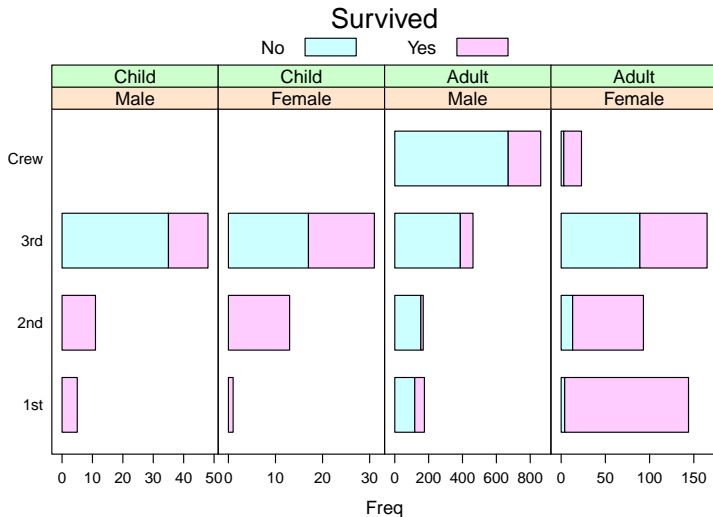
Lattice: xyplot

```
xyplot(yield ~ nitro | Variety + Block, data = Oats,  
       type = "o")
```



```
barchart(Class ~ Freq | Sex + Age,  
         data = as.data.frame(Titanic),  
         groups = Survived, stack = TRUE,  
         layout = c(4, 1),  
         auto.key = list(title = "Survived",  
                         columns = 2), scales = list(x = "free"))
```

Lattice: barchart

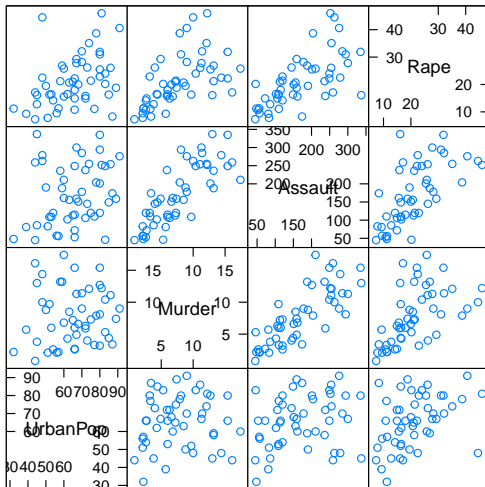


Lattice: splom

```
splom(~USArrests[c(3, 1, 2, 4)])
```

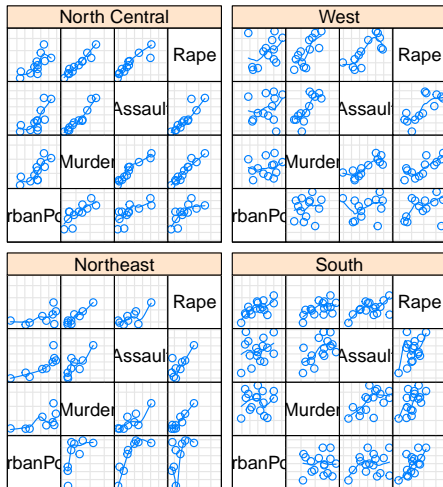
```
splom(~USArrests[c(3, 1, 2, 4)] | state.region,  
      pscales = 0, type = c("g", "p", "smooth"))
```

Lattice: splom, gráfico incondicional (1o exemplo)



Scatter Plot Matrix

Lattice: splom, gráfico condicional (2o exemplo)

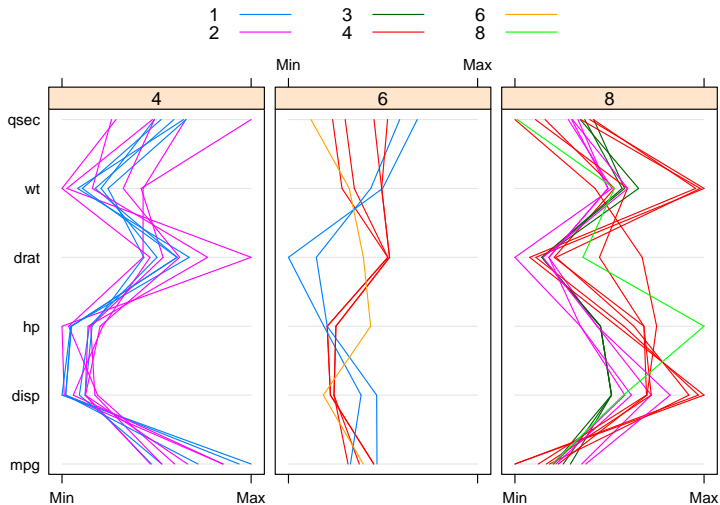


Scatter Plot Matrix

Lattice: parallel

```
parallel(~mtcars[c(1, 3, 4, 5, 6, 7)] | factor(cyl),  
mtcars, groups = carb, layout = c(3, 1),  
auto.key = list(space = "top", columns = 3))
```


Lattice: parallel



O lattice também possui funções que permitem plotar gráficos em 3D, sendo elas:

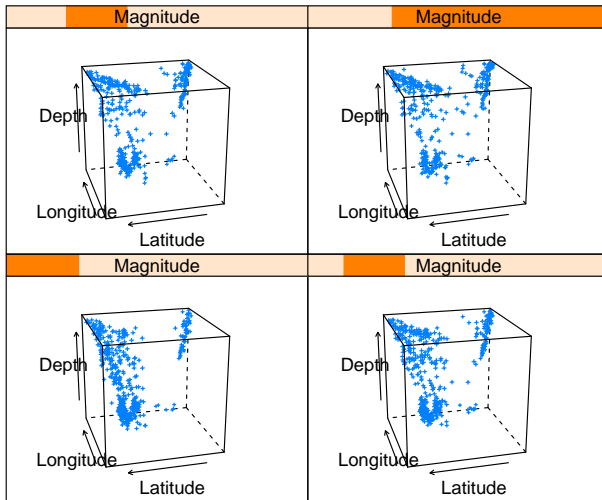
- `cloud`: plota scatterplot em 3D
- `wireframe`: plota superfícies em 3D
- `levelplot`: plota gráfico falsa-cor
- `contourplot`: plota curvas de nível

Lattice: scatterplot em 3D

- Este exemplo procura encontrar a relação entre a profundidade da ocorrência de um terremoto na ilha de Fiji com a longitude e dos mesmos.
- Aqui a análise será condicionada pela magnitude do terremoto. Como esta variável é contínua, ela será discretizada pela função `equal.count`, em que são construídas 4 *shingles* (fatores construídos a partir de intervalos com possibilidade de *overlap*).

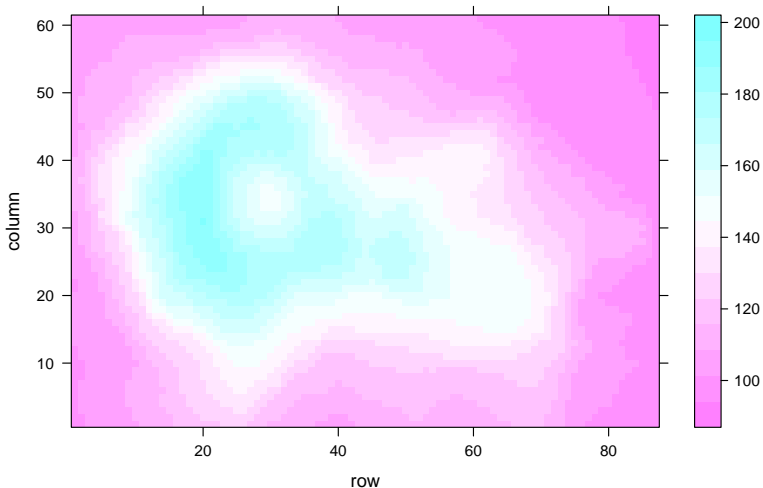
```
quakes$Magnitude <- equal.count(quakes$mag, 4)
cloud(depth ~ lat * long | Magnitude, data = quakes,
      zlim = rev(range(quakes$depth)),
      screen = list(z = 105, x = -70), panel.aspect = 0.75,
      xlab = "Longitude", ylab = "Latitude", zlab = "Depth")
```

Lattice: scatterplot em 3D



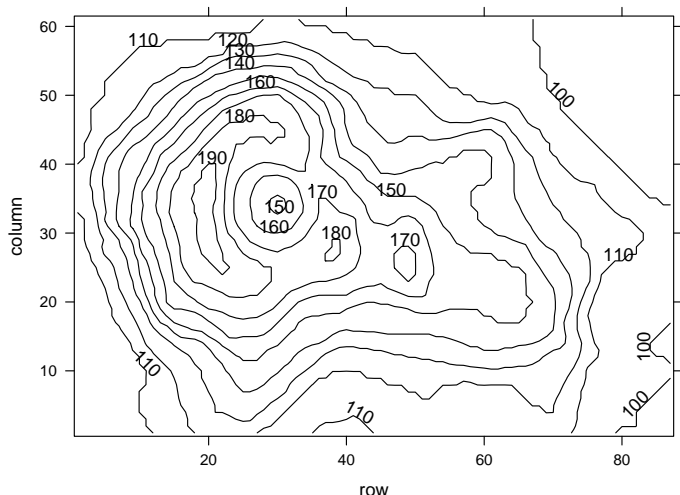
Lattice: levelplot

```
levelplot(volcano)
```



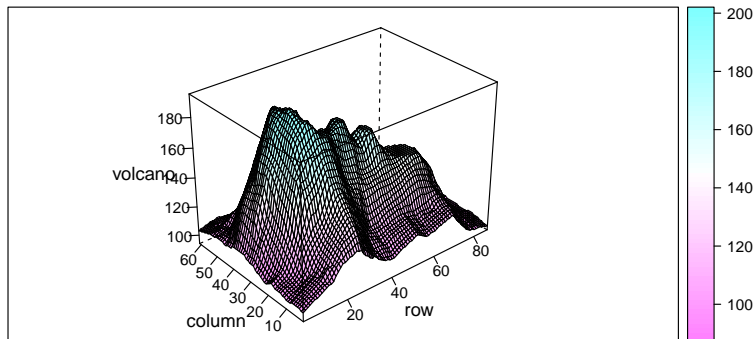
Lattice: contourplot

```
contourplot(volcano, cuts = 10, label = TRUE)
```



Lattice: wireframe

```
wireframe(volcano, panel.aspect = 0.5,  
          zoom = 1, scales=list(arrows = FALSE),  
          drape = TRUE, colorkey = TRUE)
```



Visualização de dados discretos

```
cor.Cars93 <-  
cor(Cars93[, !sapply(Cars93, is.factor)], use = "pair")  
Level.plot<-levelplot(cor.Cars93,scales=  
  list(x = list(rot = 90)))
```


Visualização de dados discretos

Level.plot

