setwd('~/Dropbox/chapman/book/')

Oscar Perpiñán Lamigueiro

June 28, 2012

Contents

SIAR

1 Different meteorological variables from the same station

There are a variety of scientific researches interested in the relationship between several meteorological variables. A suitable approach is to display the time evolution of all of them either using a panel for each of the variables or plotting them superposed. The superposition of variables with different ranges is not very useful (unless their values were previously rescaled), so this option is postponed for later use.

For this example we will use the daily data from the SIAR meteorological station located at Aranjuez (Madrid). The code of the province (28) and station (3) is extracted from http://solar.r-forge.r-project.org/data/SIAR.csv. Let's download a multivariate time series of eight years of daily data from January 2004 to December 2011.

```
prov=28 ##madrid
est=3 ##aranjuez
start='01/01/2004'
end='31/12/2011'
```

The URL of this data is defined with the information of the province, station, and start and end dates.

```
URL = paste("http://www.marm.es/siar/exportador.asp?T=DD&P=",
  prov, "&E=", est, "&I=", start, "&F=", end, sep = "")
```

The `read.zoo` from the `zoo` package accepts this string and downloads the data to construct a `zoo` object[1]. Several arguments are directly passed to `read.table` (`header`, `skip`, etc.) and are conveniently detailed in the help page of this function. The `index.column` is the number of the column with the time index, and `format` defines the date format of this index.

```
library(zoo)

aranjuez <- read.zoo(URL, index.column = 1,
                     format = "%d/%m/%Y",
                     header = TRUE, skip = 1, fill = TRUE,
                     dec = ",", as.is = TRUE)
```

We will retain only a subset of the meteorological variables: average, maximum and minimum ambient temperature; average and maximum humidity; average and maximum wind speed; rainfall; solar radiation on the horizontal plane; and evotranspiration.

```
aranjuezClean <- aranjuez[, c(1, 2, 4, 6, 7, 11, 13, 16, 17, 18)]

names(aranjuezClean) <- c('TempAvg', 'TempMax', 'TempMin',
                          'HumidAvg', 'HumidMax',
                          'WindAvg', 'WindMax',
                          'Rain', 'Radiation', 'ET')

summary(aranjuezClean)
```

```
     Index               TempAvg          TempMax          TempMin
 Min.   :2004-01-01   Min.   :-5.309   Min.   :-2.362   Min.   : -12.980
 1st Qu.:2005-12-29   1st Qu.: 7.692   1st Qu.:14.530   1st Qu.:   1.517
 Median :2008-01-09   Median :13.810   Median :21.670   Median :   7.170
 Mean   :2008-01-03   Mean   :14.405   Mean   :22.531   Mean   :   8.241
 3rd Qu.:2010-01-02   3rd Qu.:21.615   3rd Qu.:30.875   3rd Qu.:  12.590
 Max.   :2011-12-31   Max.   :30.680   Max.   :41.910   Max.   :1133.000
```

[1] The `solaR` package provides the function `readSIAR` which is designed around this code.

```
   HumidAvg          HumidMax            WindAvg              WindMax
Min.   : 19.89   Min.   :   35.88   Min.   :  0.2510   Min.   :   0.000
1st Qu.: 47.04   1st Qu.:   81.60   1st Qu.:  0.6672   1st Qu.:   3.844
Median : 62.58   Median :   91.00   Median :  0.9210   Median :   5.155
Mean   : 62.16   Mean   :   91.43   Mean   :  1.6378   Mean   :   8.177
3rd Qu.: 77.38   3rd Qu.:   94.90   3rd Qu.:  1.4418   3rd Qu.:   6.791
Max.   :100.00   Max.   :2310.00   Max.   :359.6000   Max.   :2142.000

      Rain          Radiation            ET
Min.   : 0.000   Min.   : 0.277   Min.   :0.000
1st Qu.: 0.000   1st Qu.: 9.370   1st Qu.:1.168
Median : 0.000   Median :16.660   Median :2.758
Mean   : 1.094   Mean   :16.742   Mean   :3.091
3rd Qu.: 0.200   3rd Qu.:24.650   3rd Qu.:4.926
Max.   :49.730   Max.   :32.740   Max.   :8.564
NA's   :4        NA's   :13       NA's   :18
```

From the summary it is clear that parts of these time series include erroneous outliers which can be safely removed:

```
aranjuezClean <- within(as.data.frame(aranjuezClean),{
  TempMin[TempMin>40] <- NA
  HumidMax[HumidMax>100] <- NA
  WindAvg[WindAvg>10] <- NA
  WindMax[WindMax>10] <- NA
})

aranjuez <- zoo(aranjuezClean, index(aranjuez))
```

1.1 xyplot

This multivariate time series can be directly displayed with the `xyplot` method of `lattice` using the `layout` argument to display the variables with parallel panels arranged in rows.

```
xyplot(aranjuez, layout=c(1, ncol(aranjuez)))
```

This first attempt can be improved with a custom panel function. This function will generate the content of each panel using the information processed by `xyplot`. Since these functions are executed consecutively, the order of the functions determines the superposition of information:

`aranjuez.pdf`

Figure 1: Time plot of the collection of meteorological time series of the Aranjuez station.

- The label of each time series is displayed with text inside each panel instead of using the strips mechanism. The `panel.text` prints the name of each variable with the aid of `panel.number`.
- The alternation of years is displayed with blocks of gray and white color using the `panel.xblocks` function from `latticeExtra`. The year is extracted (as character) from the time index of the `zoo` object with `format.POSIXlt`.
- Those values below the mean of each variable are highlighted with short red color blocks at the bottom of each panel, again with the `panel.xblocks` function.

```r
Year <- function(x)format(x, "%Y")

xyplot(aranjuez, layout=c(1, ncol(aranjuez)), strip=FALSE,
       scales=list(y=list(cex=0.6, rot=0)),
       panel=function(x, y, ...){
         panel.xblocks(x, Year, col = c("lightgray", "white"),
                       border = "darkgray")
         panel.xblocks(x, y<mean(y, na.rm=TRUE), col = "indianred1",
                       height=unit(0.05, 'npc'))
         panel.xyplot(x, y, ...)
         panel.text(x[1], min(y, na.rm=TRUE),
                    names(aranjuez)[panel.number()],
                    cex=0.7, pos=2,...)
       })
```

1.2 Splom

But, what if instead of displaying the time evolution we want to confront the variables between them? Then a matrix of scatter plots is the answer. This graphical tool is implemented in the `splom` function.

```r
splom(as.data.frame(aranjuez))
```

However, for large datasets, the display of a large number of points in a scatter plot matrix produces hidden point density, long computation times and slow displays. These problems can be circumvented with the estimation and representation of points densities. A common encoding uses gray scales, pseudo colors or partial transparency. An improved scheme encodes density as the size of hexagon symbols inscribed within hexagonal binning region [**?**].

The `hexbin` package includes several functions for hexagonal binning. The `panel.hexbinplot` is a good substitute for the default panel function. Besides, our first attempt with `splom` can be improved with several modifications:

- The scales ticks and labels are suppressed with `pscale=0`.
- The panels of the lower part of the matrix (`lower.panel`) will include a locally weighted scatterplot smoothing (loess) with `panel.loess`.
- The diagonal panels (`diag.panel`) will display the kernel density estimate of each variable. The `density` function computes this estimate. The result is adjusted to the panel limits (calculated

Figure 2: Enhanced time plot of the collection of meteorological time series of the Aranjuez station.

with `current.panel.limits`). The kernel density is plotted with `panel.lines` and the `diag.panel.splom` function completes the content of each diagonal panel (since `pscale=0` this function only generates the label of each variable).

- The point density is encoded with the palette `BTC` (ligther colors for high density values and darker colors for almost empty regions, with a gradient of blue hues for intermediate values).

```
library(hexbin)

splom(as.data.frame(aranjuez),
```

aranjuezSplom.pdf

Figure 3: Scatter plot matrix of the collection of meteorological time series of the Aranjuez station.

```
panel=panel.hexbinplot, xlab='',
colramp=BTC,##function(n)BTC(n, beg=250, end=5),
diag.panel = function(x, ...){
  yrng <- current.panel.limits()$ylim
  d <- density(x, na.rm=TRUE)
  d$y <- with(d, yrng[1] + 0.95 * diff(yrng) * y / max(y))
  panel.lines(d)
  diag.panel.splom(x, ...)
},
lower.panel = function(x, y, ...){
  panel.hexbinplot(x, y, ...)
```

```
  panel.loess(x, y, ..., col = 'red')
},
pscale=0, varname.cex=0.7
)
```

aranjuezSplom.pdf

Figure 4: Scatter plot matrix of the collection of meteorological time series of the Aranjuez station using hexagonal binning.

Let's add a bit of interactivity to this plot with the identification of some points. This task is easy with `panel.link.splom`. The points are selected via mouse clicks (and highlighted in green). Clicks other than left-clicks terminate the procedure. The output of this function is the set of chosen points.

```
trellis.focus('panel', 1, 1)
idx <- panel.link.splom(pch=13, cex=0.6, col='green')
```

```
aranjuez[idx,]
```

```
Error en grid.Call.graphics(L_downviewport, name$name, strict) :
  Viewport 'plot_01.panel.1.1.vp' was not found
Error en grid.Call.graphics(L_downviewport, name$name, strict) :
  Viewport 'subpanel.6.7' was not found
Error en `[.zoo`(aranjuez, idx, ) : objeto 'idx' no encontrado
```

```
Error: inesperado símbolo en "Error en"
Error: inesperado string constante en "  Viewport 'plot_01.panel.1.1.vp'"
Error: inesperado símbolo en "Error en"
Error: inesperado string constante en "  Viewport 'subpanel.6.7'"
Error: inesperado símbolo en "Error en"
```

1.3 Reshape

A drawback of the matrix of scatter plots is that each panel is drawn independently so it is impossible to compute a common color key for all of them. In other words, two cells with exactly the same color in different panels encode different points densities.

It is possible to display a reduced set of variables against another one and generate a common color key using the `hexbinplot` function. First, the dataset must be reshaped from the wide format (one colum for each variable) to the long format (only one column for the values with one row for each observation).

The `reshape` function needs several arguments to perform the conversion. The most important is the `data.frame` to be transformed. Then there is the names of variables to be mapped to a single variable in the long dataset (the three ambient temperatures). The name of this variable can be set with `v.names`. Finally, `timevar` is the name of the column in long format that differentiates multiple observations from the same variable. The values of this column are defined with the `times` argument.

```
aranjuezRshp <- reshape(as.data.frame(aranjuez), direction='long',
                        varying=list(names(aranjuez)[1:3]),
                        v.names='Temperature',
                        times=names(aranjuez)[1:3],
                        timevar='Statistic')

head(aranjuezRshp)
```

```
          HumidAvg HumidMax WindAvg WindMax Rain Radiation        ET Statistic
1.TempAvg     88.3     95.9   0.746   3.528    0     5.490 0.5352688   TempAvg
2.TempAvg     83.3     98.5   1.078   6.880    0     6.537 0.7710499   TempAvg
3.TempAvg     75.0     94.4   0.979   6.576    0     8.810 0.8361229   TempAvg
4.TempAvg     82.0     97.0   0.633   3.704    0     9.790 0.6861381   TempAvg
5.TempAvg     83.2     97.0   0.389   2.244    0    10.300 0.5152422   TempAvg
6.TempAvg     84.5     96.5   0.436   2.136    0     9.940 0.4886631   TempAvg
          Temperature id
1.TempAvg       4.044  1
2.TempAvg       5.777  2
3.TempAvg       5.850  3
4.TempAvg       4.408  4
5.TempAvg       3.081  5
6.TempAvg       2.304  6
```

The `hexbinplot` displays this dataset with a different panel for each type of temperature (average, maximum and minimum) but with a common color key encoding the point density. Now, two cells with the same color in different panels encode the same value.

```
hexbinplot(Radiation~Temperature|Statistic, data=aranjuezRshp,
           layout=c(1, 3), colramp=BTC,
           panel=function(x, y, xlim,...){
             panel.hexbinplot(x, y, ...)
             panel.loess(x, y, ..., col = 'red')
           }
           )
```

aranjuezHexbinplot.pdf

Figure 5: Scatter plot with hexagonal binning of temperature versus solar radiation using data of the Aranjuez station.