Spatial, Temporal and Spatio-Temporal Data in R

Solutions to Handson Exercises

L. Torgo

ltorgo@fc.up.pt

Faculdade de Ciências / LIAAD-INESC TEC, LA
Universidade do Porto

Jan, 2017

Jožef Stefan Institute

Hands On Time Series

Package **quantmod** (an extra package that you need to install) contains several facilities to handle financial time series. Among them, the function `getMetals` allows you to download the prices of metals from `oanda.com`. Explore the help page of the function to try to understand how it works, and the answer the following:

1. Obtain the prices of gold of the current year solution
2. Show the prices in January solution
3. Show the prices from February 10 till March 15 solution
4. Obtain the prices of silver in the last 30 days
   Tip: explore the function `seq.Date()` solution
5. Plot the prices of silver in the last 7 days
   Tip: explore the function `last()` on package **xts** solution

Jožef Stefan Institute

Solution to Exercise 1

- Obtain the prices of gold of the current year

```r
library(quantmod)
getMetals("gold",from="2015-01-01",base.currency="EUR")

## [1] "XAUEUR"
```

Go Back

Solution to Exercise 2

- Show the prices in January

```
XAUEUR["2015-01"]

##              XAU.EUR
## 2015-01-01  977.427
## 2015-01-02  982.924
## 2015-01-03  990.466
## 2015-01-04  990.484
## 2015-01-05 1000.130
## 2015-01-06 1014.030
## 2015-01-07 1024.680
## 2015-01-08 1024.840
## 2015-01-09 1027.120
## 2015-01-10 1033.100
## 2015-01-11 1033.110
## 2015-01-12 1035.700
## 2015-01-13 1046.720
## 2015-01-14 1045.820
## 2015-01-15 1059.860
## 2015-01-16 1090.280
## 2015-01-17 1106.630
```

Solution to Exercise 3

- Show the prices from February 10 till March 15

```
XAUEUR["2015-02-10/2015-03-15"]

##            XAU.EUR
## 2015-02-10 1094.63
## 2015-02-11 1089.61
## 2015-02-12 1077.75
## 2015-02-13 1075.67
## 2015-02-14 1079.54
## 2015-02-15 1079.56
## 2015-02-16 1080.91
## 2015-02-17 1073.88
## 2015-02-18 1060.29
## 2015-02-19 1065.14
## 2015-02-20 1063.03
## 2015-02-21 1056.05
## 2015-02-22 1056.03
## 2015-02-23 1057.84
## 2015-02-24 1059.15
## 2015-02-25 1062.88
## 2015-02-26 1069.82
```

Solution to Exercise 4

- Obtain the prices of silver in the last 30 days

```
fstDate <- Sys.Date() - 30
getMetals("silver",from=fstDate,base.currency="EUR")

## [1] "XAGEUR"
```

or a more general setting

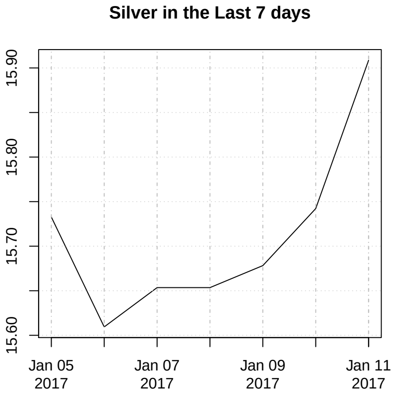
```
fstDate <- seq.Date(from=Sys.Date(),by="-30 days",length.out=2)[2]
getMetals("silver",from=fstDate,base.currency="EUR")
```

Go Back

Solution to Exercise 5

- Plot the prices of silver in the last 7 days

```
plot(last(XAGEUR,"7 days"),main="Silver in the Last 7 days")
```

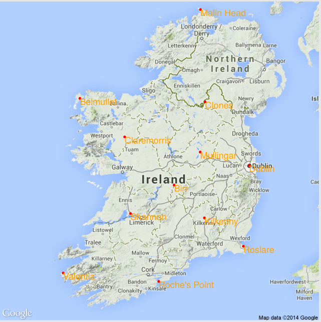


Jožef Stefan Institute

Hands On Spatio-Temporal Data with ggmap

The file `irishWind.Rdata` contains two data frames with information on wind data values collected in several meteorological stations in Ireland along several years. The data frame **wind** contains the wind values for the different stations (in wide format), while the **wind.loc** data frame contains information on the stations. Using this data set answer the following questions:

1. Obtain the geographic coordinates of the stations solution
2. Reproduce the graph to the right solution



Solutions to Exercise 1

- Obtain the geographic coordinates of the stations

```r
library(ggmap)
load("irishWind.Rdata") # The file contains two data frames: "wind" and "wind.loc"
```

```r
## alternative 1 (using google maps)
wind.loc <- cbind(wind.loc,geocode(paste(wind.loc$Station,"Ireland",sep=", ")))
## alternative 2 (translating between formats)
library(sp)
wind.loc$lat <- as.numeric(char2dms(as.character(wind.loc[["Latitude"]])))
wind.loc$lon <- as.numeric(char2dms(as.character(wind.loc[["Longitude"]])))
```

Go Back

Solutions to Exercise 2

- Reproduce the graph

```
ir <- get_map("Ireland",zoom=7)
ggmap(ir,extent="device",
      base_layer=ggplot(data=wind.loc,aes(x=lon,y=lat,label=Station))) +
    geom_point(color="red",size=2) +
    geom_text(hjust=0,vjust=1,color="orange")
```

Go Back

Jožef Stefan Institute

Hands On Spatio-Temporal Data with ggmap (cont.)

3. Using the functionalities provided by packages **tidyr** and **dplyr** obtain a data frame with the average yearly wind speed for each station. solution
4. Produce a spatio-temporal showing theses yearly averages on the stations. solution

Jožef Stefan Institute

Solutions to Exercise 3

- Obtain the geographic coordinates of the stations

```
library(tidyr)
library(dplyr)

wL <- gather(wind,
             Code, Wind,
             RPT:MAL)

wL <- tbl_df(wL) # move into dplyr data table class

## Obtain the yearly averages per station
yavg <-  group_by(wL, Code, year) %>%
         summarise(meanWind=mean(Wind)) %>%
         inner_join(wind.loc[,c(2,6,7)])


## Warning in inner_join_impl(x, y, by$x, by$y, suffix$x, suffix$y):  joining character vector
and factor, coercing into character vector
```

Go Back

Jožef Stefan Institute

Solutions to Exercise 4

- Reproduce the graph to the right

```
ggmap(ir,extent="device",
      base_layer=ggplot(data=yavg,
                        aes(x=lon, y=lat,
                        size=meanWind,
                        color=meanWind))) +
    geom_point() +
    facet_wrap( ~ year, ncol=6)
```

Go Back