Spatial, Temporal and Spatio-Temporal Data in R

L. Torgo

ltorgo@dal.ca
Faculty of Computer Science / Institute for Big Data Analytics
Dalhousie University

Jan, 2020

Motivation

- We leave in a world where data collection devices abound
- Frequently these devices are location- and time-aware (e.g. smartphones)
- Data sets where observations are tagged by time and/or location are increasing at a fast rate

AFIRM

Motivation - 2

- If we suspect our observations my have some form of time and or space correlation we should not neglect these effects in our analysis
  - Do observations of X at time t depend on the values at recent past times ?
  - Do observations of X at location z depend on values at neighboring locations ?
  - Do observations of X at location z on time t depend on the values at neighboring locations currently or in recent past times?

AFIRM

Some Tasks/Problems Addressed in Spatio-Temporal Data Mining

- Exploratory analysis of the data with the goal of understanding the eventual spatial and temporal patterns
- Prediction
  - Spatial interpolation
  - Time series forecasting
  - Spatio-temporal forecasting

AFIRM

Why do we need something specific?

- Spatial coordinates are just numbers...
- Temporal tags can be seen as strings following some rules...
- Special purpose classes facilitate the manipulation of this type of data
  - e.g. temporal or spatial queries on a data set

Some Relevant R Packages

- Temporal data
  - Package **xts**
- Spatial data
  - Package **sp**
- Spatio-temporal data
  - Package **spacetime**

Further information on the following R CRAN Task Views:

- Handling and Analyzing Spatio/temporal Data (`https://cran.r-project.org/view=SpatioTemporal`)
- Analysis of Spatial Data (`https://cran.r-project.org/view=Spatial`)
- Time Seriesw Analysis (`https://cran.r-project.org/view=TimeSeries`)

Time Dependent Data in R

Package **xts**

```
library(xts)
```

- Loading the prices of gold from the Internet using extra package **quantmod**

```
library(quantmod)
gold <- getMetals("XAU",from=Sys.Date()-90,auto.assign=FALSE)
head(gold)

##             XAU.USD
## 2019-10-23 1491.816
## 2019-10-24 1496.285
## 2019-10-25 1505.479
## 2019-10-26 1504.521
## 2019-10-27 1504.432
## 2019-10-28 1499.380

gold["2019-12-03"]

##             XAU.USD
## 2019-12-03 1470.208

gold["2020-01-02/2020-01-05"]

##             XAU.USD
## 2020-01-02 1523.589
## 2020-01-03 1544.784
## 2020-01-04 1552.394
## 2020-01-05 1553.252
```


Time Dependent Data in R - 2

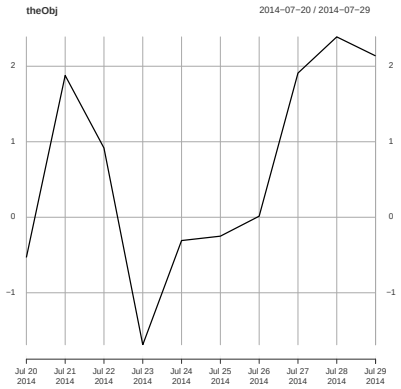
- Creating an **xts** object

```
someDates <- seq.Date(as.Date("2014-07-20"),by="day",length=10)
someDates

##  [1] "2014-07-20" "2014-07-21" "2014-07-22" "2014-07-23" "2014-07-24"
##  [6] "2014-07-25" "2014-07-26" "2014-07-27" "2014-07-28" "2014-07-29"

someValues <- rnorm(10)
someValues

##  [1] -0.53206168  1.87804940  0.91714625 -1.68886925 -0.30838397
##  [6] -0.25033867  0.01435655  1.90943693  2.38876181  2.13714832

theObj <- xts(someValues,someDates)
theObj[1:3]

##                 [,1]
## 2014-07-20 -0.5320617
## 2014-07-21  1.8780494
## 2014-07-22  0.9171463
```

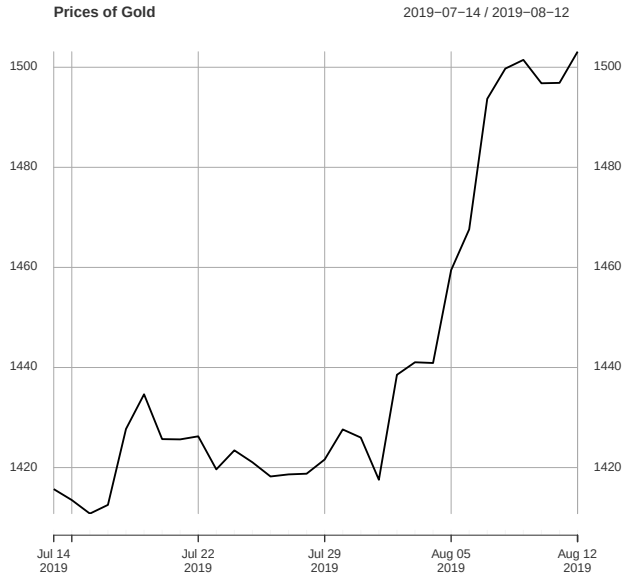
Time Dependent Data with **xts** - 3

- Visualizing an **xts** object

```
plot(theObj)
```




```
plot(gold,main="Prices of Gold")
```



Spatial Data with **sp**

- Package **sp** defines a series of classes of objects that can be used to store geo-referenced data sets
- Many other R packages build upon the classes defined in this package
- A broad overview of the (many) packages existing in R for this type of data can be found in
  Analysis of Spatial Data Task View

Some of the Classes Defined in **sp**

- `SpatialPointsDataFrame`
  A data frame like structure that can be used to store data (values of a set of variables) about geographic locations (a spatial point)
- `SpatialLinesDataFrame` and `SpatialPolygonsDataFrame`
  A data frame like structure that can be used to store data (values of a set of variables) about lines and polygons (closed sequences of lines)
- `SpatialGridDataFrame`
  A data frame like structure that can be used to store data (values of a set of variables) about a regular grid of spatial points

A Simple Illustration

Fires data of 500m^2 regions of Portugal

- Official data on fires in different regions of Portugal
- The data set we will use contains information on 25000 locations

```
library(readr)
df <- read_csv("firesnew_25000_500m.txt")
```

```
df[1:3,]
```

```
##   FID_ CID ano1991 ano1992 ano1993 ano1994 ano1995 ano1996 ano1997 ano1998
## 1   NA   1       0       0       0       0       0       0       0       0
## 2   NA   2       0       0       0       0       0       0       0       0
## 3   NA   3       0       0       0       0       0       0       0       0
##   ano1999 ano2000        x       y
## 1       0       0 -7.31924 38.5406
## 2       0       0 -7.63557 40.5022
## 3       0       0 -7.90273 40.3418
```

A Simple Illustration

Creating a `SpatialPointsDataFrame` object with 2000 data

```
library(sp)
spatialCoords <- cbind(long=df$x, lat=df$y)
coordRefSys <- CRS("+proj=longlat +ellps=WGS84")
fires2000 <- SpatialPointsDataFrame(spatialCoords,
                                    df[, "ano2000", drop=FALSE],
                                    proj4string=coordRefSys)
fires2000[1:3,]
```

```
##           coordinates ano2000
## 1 (-7.31924, 38.5406)       0
## 2 (-7.63557, 40.5022)       0
## 3 (-7.90273, 40.3418)       0
```


A Simple Illustration

Some utility functions

```
bbox(fires2000)
```

```
##           min      max
## long -9.49174 -6.20743
## lat  36.98050 42.14360
```

```
coordinates(fires2000)[1:3,]
```

```
##          long     lat
## [1,] -7.31924 38.5406
## [2,] -7.63557 40.5022
## [3,] -7.90273 40.3418
```

```
summary(fires2000)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##           min      max
## long -9.49174 -6.20743
## lat  36.98050 42.14360
## Is projected: FALSE
## proj4string : [+proj=longlat +ellps=WGS84]
## Number of points: 25000
## Data attributes:
##     ano2000
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
```

Spatio-temporal Data with **spacetime**

- Spatio-temporal data frequent formats
  - **Time-wide** - different columns have different measurements across time
  - **Space-wide** - different columns have measurements across different locations
  - **Long formats** - each data record contains measurements for a space-time combination
- Our previous fires data example used a time-wide format

Spatio-temporal Data with **spacetime** (cont.)

- Some types of spatio-temporal data sets
  - **Full grids** - we have values for all combinations of location and time stamps
    number of values = number of locations $\times$ number of time stamps
  - **Sparse grids** - similar but only non-missing values are stored
  - **Irregular data** - Each value is measured on a certain location and time, without regularity
  - **Time intervals, moving objects and trajectories** - e.g. spatial features constant but values collected on a time interval; trajectories where irregular space-time points form a sequence; etc.

Spatio-temporal Data with **spacetime** (cont.)

- The package **spacetime** defines different classes for these situations
  - **Full grids** - class `STFDF`
  - **Sparse grids** - class `STSDF`
  - **Irregular data** - class `STIDF`
  - **Time intervals, moving objects and trajectories** - class `STTDF`

A Simple Illustration with the Fires Data

The fires data again...

```
library(readr)
df <- read_csv("firesnew_25000_500m.txt")
```

```
df[1:3,]
```

```
##   FID_ CID ano1991 ano1992 ano1993 ano1994 ano1995 ano1996 ano1997 ano1998
## 1   NA   1       0       0       0       0       0       0       0       0
## 2   NA   2       0       0       0       0       0       0       0       0
## 3   NA   3       0       0       0       0       0       0       0       0
##   ano1999 ano2000        x       y
## 1       0       0 -7.31924 38.5406
## 2       0       0 -7.63557 40.5022
## 3       0       0 -7.90273 40.3418
```

Creating a STFDF object

- First let us put the data in Long Format

```
library(tidyr)
x <- gather(df[,3:14], year, burnt, ano1991:ano2000)
x[1:2,]


## # A tibble: 2 x 4
##       x     y year    burnt
##   <dbl> <dbl> <chr>   <dbl>
## 1 -7.32  38.5 ano1991     0
## 2 -7.64  40.5 ano1991     0


x$year <- substr(x$year,4,7)
x[1:2,]


## # A tibble: 2 x 4
##       x     y year  burnt
##   <dbl> <dbl> <chr> <dbl>
## 1 -7.32  38.5 1991      0
## 2 -7.64  40.5 1991      0
```


Creating a STFDF object (cont.)

- We are now ready to create the object

```
library(sp)
library(spacetime)
spatialCoords <- cbind(long=df$x,lat=df$y)
coordRefSys <- CRS("+proj=longlat +ellps=WGS84")
## The spatial points for which we have information (burnt or not)
sp <- SpatialPoints(spatialCoords,coordRefSys)
## The time stamps for which we have information (for all points - full grid)
timeStamps <- as.POSIXct(paste0(unique(x$year),"-01-01"), tz="GMT")
std <- STFDF(sp,timeStamps,data=x[,"burnt",drop=FALSE])
```

Querying the spatio-temporal data set

- Data for a certain location

```
## The first location (spatial point)
std[1, ]

##            burnt timeIndex
## 1991-08-13     0         1
## 1992-08-13     0         2
## 1993-08-13     0         3
## 1994-08-13     0         4
## 1995-08-13     0         5
## 1996-08-13     0         6
## 1997-08-13     0         7
## 1998-08-13     0         8
## 1999-08-13     0         9
## 2000-08-13     0        10
```


Querying the spatio-temporal data set

- Data for all locations on a certain time stamp

```
## Second position is a temporal query on all locations
dataSince1995 <- std[,"1995/"]
data1991 <- std[,"1991"]
## And these are the values for the first 4 locations for 1991
data1991[1:4,]

##           coordinates burnt
## 1 (-7.31924, 38.5406)     0
## 2 (-7.63557, 40.5022)     0
## 3 (-7.90273, 40.3418)     0
## 4 (-7.25657, 39.2572)     0
```