Introduction

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
If we suspect our observations may 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?

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
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 Series Analysis
  (https://cran.r-project.org/view=TimeSeries)
Time Dependent Data in R

Package \texttt{xts}

\begin{verbatim}
\textbf{library(xts)}

\textbf{\textbullet\ Loading the prices of gold from the Internet using extra package quantmod}

\textbf{library(quantmod)}
gold <- getMetals("XAU", \texttt{from-Sys.Date()-90}, \texttt{auto.assign=FALSE})
\textbf{head(gold)}

\begin{verbatim}
## 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
\end{verbatim}

\textbf{gold["2019-12-03"]}

\begin{verbatim}
## XAU.USD
## 2019-12-03 1470.208
\end{verbatim}

\textbf{gold["2020-01-02/2020-01-05"]}

\begin{verbatim}
## XAU.USD
## 2020-01-02 1523.589
## 2020-01-03 1544.784
## 2020-01-04 1552.394
## 2020-01-05 1553.252
\end{verbatim}
\end{verbatim}

\section*{Time Dependent Data in R - 2}

\textbf{\textbullet\ Creating an \texttt{xts} object}

\begin{verbatim}
someDates <- \texttt{seq.Date(as.Date("2014-07-20"), by="day", length=10)}
someDates

\begin{verbatim}
\end{verbatim}

someValues <- \texttt{rnorm(10)}
someValues

\begin{verbatim}
## [1] -0.53206168 1.87804940 0.91714625 -1.68886925 -0.30838397
## [6] -0.25033867 0.01435655 1.90943693 2.38876181 2.13714832
\end{verbatim}

theObj <- \texttt{xts(someValues, someDates)}
theObj[1:3]

\begin{verbatim}
## [,1]
## 2014-07-20 -0.5320617
## 2014-07-21 1.8780494
## 2014-07-22 0.9171463
\end{verbatim}
\end{verbatim}
Time Dependent Data with \texttt{xts} - 3

- Visualizing an \texttt{xts} object

\begin{verbatim}
\texttt{plot(theObj)}
\end{verbatim}

\begin{verbatim}
\texttt{plot(gold,main="Prices of Gold")}
\end{verbatim}

Spatial Data with \texttt{sp}

- Package \texttt{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 500m² regions of Portugal

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

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

df[1:3,]
```

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 NA 1 0 0 0 0 0 0 0 0 0</td>
<td>-7.31924</td>
<td>38.5406</td>
</tr>
<tr>
<td>2 NA 2 0 0 0 0 0 0 0 0 0</td>
<td>-7.63557</td>
<td>40.5022</td>
</tr>
<tr>
<td>3 NA 3 0 0 0 0 0 0 0 0 0</td>
<td>-7.90273</td>
<td>40.3418</td>
</tr>
</tbody>
</table>
A Simple Illustration
Creating a `SpatialPointsDataFrame` object with 2000 data

```r
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)
```

```r
bbox(fires2000)
## min max
## long -9.49174 -6.20743
## lat 36.98050 42.14360
```

```r
coordinates(fires2000)[1:3,]
## long lat
## [1,] -7.31924 38.5406
## [2,] -7.63557 40.5022
## [3,] -7.90273 40.3418
```

```r
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
##   Mean :0.01612
##   3rd Qu.:0.00000
##   Max. :1.00000
```
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

---

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.
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...

```r
library(readr)

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

```
## 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 -7.31924 38.5406
## 2 0 -7.63557 40.5022
## 3 0 -7.90273 40.3418
```
Creating a **STFDF** object

- First let us put the data in Long Format

```r
library(tidyrr)
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
```

```r
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
```

© L.Torgo (Dalhousie University)

Creating a **STFDF** object (cont.)

- We are now ready to create the object

```r
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(uniq(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**

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

<table>
<thead>
<tr>
<th></th>
<th>burnt</th>
<th>timeIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1991-08-13</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1992-08-13</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1993-08-13</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1994-08-13</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1995-08-13</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1996-08-13</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>1997-08-13</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>1998-08-13</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>1999-08-13</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>2000-08-13</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

- **Data for all locations on a certain time stamp**

```r
## 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,]
```

<table>
<thead>
<tr>
<th></th>
<th>coordinates</th>
<th>burnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-7.31924, 38.5406)</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>(-7.63557, 40.5022)</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>(-7.90273, 40.3418)</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>(-7.25657, 39.2572)</td>
<td>0</td>
</tr>
</tbody>
</table>