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

Packages for Visualizing Spatiotemporal Data

- Spatial visualization is a key step on spatio-temporal data analysis
- R has several packages that can help with these tasks
- Some of the most useful are:
  - **ggmap**, a package that combines spatial information from providers of maps (e.g. Google Maps) with the beautiful graphics of the **ggplot2** package (`https://github.com/dkahle/ggmap`)
  - **leaflet**, a package that links R with one of the most popular open-source JavaScript libraries for interactive maps (`https://rstudio.github.io/leaflet/`)
  - **mapview** a package that provides functions to very quickly and conveniently create interactive visualisations of spatial data (`https://r-spatial.github.io/mapview/`)

The Package **ggmap**

- As **ggmap** uses **ggplot2** graphics, the graphs have the typical properties/components found in **ggplot2** plots
- However, some of these are fixed to map-related information:
  - The *x* aesthetic is fixed to longitude
  - The *y* aesthetic is fixed to latitude
  - The coordinate system is fixed to the Mercator projection

The process of using **ggmap**

- Download a map image
- Plot it as basic layer using **ggplot2**
- Plot additional layers of data, statistics or models on top
- In **ggmap** this is done using:
  - `get_map()` to obtain the map image
  - `ggmap()` to make the actual plot

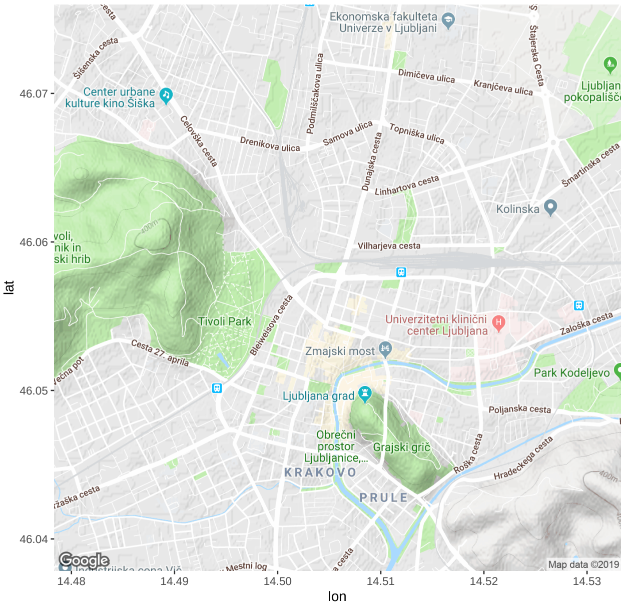
The function get_map()

- **ggmap** can use several map providers
  Google Maps, OpenStreet Maps, Stamen Maps and CloudMade Maps
- Recently (mid 2018), Google changed the policies to access map data. Now you need an API Key to obtain maps from Google Maps. Check the help page of the function `register_google` for further details/instructions

The function get_map()

A simple example

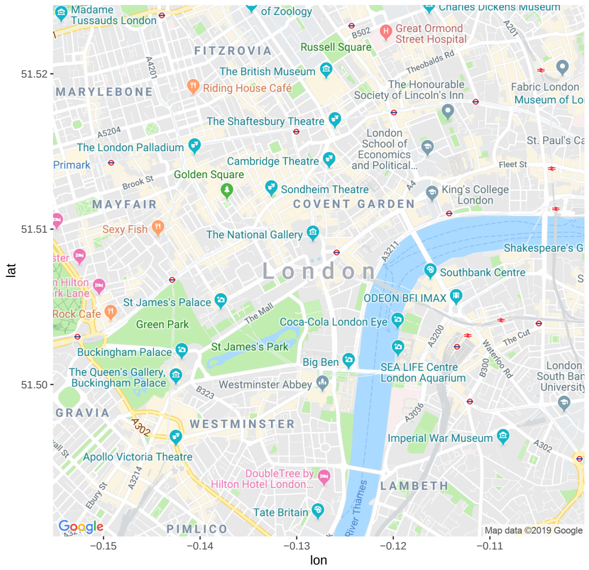
```
library(ggmap)
map <- get_map("Ljubljana",
               zoom=14)
ggmap(map)
```

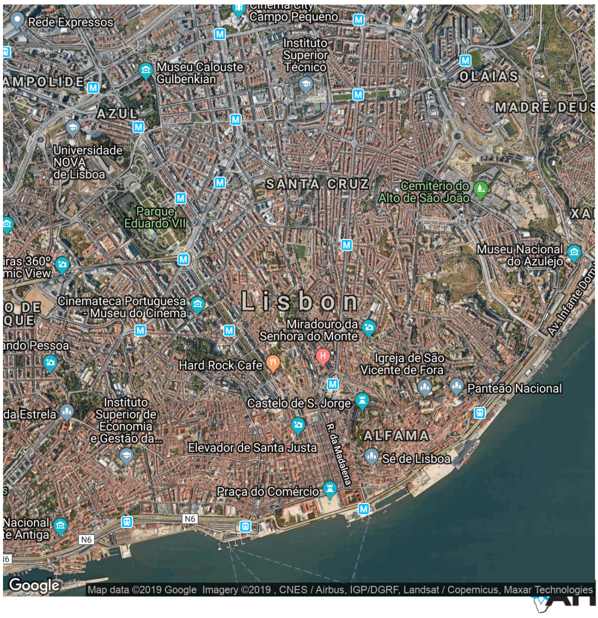


The function get_map() - a few examples

```
library(ggmap)
map <- get_map("London",zoom=14,
               maptype="roadmap")
ggmap(map)
```

```
map <- get_map("Lisbon",zoom=14,
               maptype="hybrid")
ggmap(map,extent="device")
```




Rede Expressos
Campo Pequeno
Instituto Superior Técnico
Museu Calouste Gulbenkian
AMPOLIDE
AZUL
OLAIAS
MADRE DEUS
Universidade NOVA de Lisboa
SANTA CRUZ
Cemitério do Alto de São João
Parque Eduardo VII
Museu Nacional do Azulejo
Cinemateca Portuguesa - Museu do Cinema
Lisbon
Miradouro da Senhora do Monte
Hard Rock Cafe
Igreja de São Vicente de Fora
Panteão Nacional
Instituto Superior de Economia e Gestão da...
Castelo de S. Jorge
ALFAMA
Elevador de Santa Justa
Sé de Lisboa
Praça do Comércio
Google
Map data ©2019 Google Imagery ©2019 , CNES / Airbus, IGP/DGRF, Landsat / Copernicus, Maxar Technologies

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,]


## # A tibble: 3 x 14
##   FID_    CID ano1991 ano1992 ano1993 ano1994 ano1995 ano1996 ano1997
##   <lgl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 NA        1       0       0       0       0       0       0       0
## 2 NA        2       0       0       0       0       0       0       0
## 3 NA        3       0       0       0       0       0       0       0
## # ... with 5 more variables: ano1998 <dbl>, ano1999 <dbl>, ano2000 <dbl>,
## #   x <dbl>, y <dbl>
```

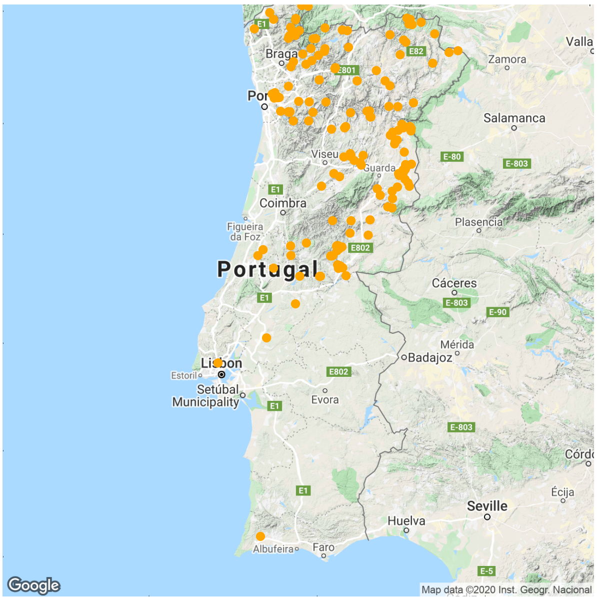
Putting the data in long format

```
library(tidyr)
dat <- gather(df[,3:14],year, burnt, ano1991:ano2000)
dat$year <- substr(dat$year,4,7)
head(dat,4)

## # A tibble: 4 x 4
##         x      y year  burnt
##     <dbl>  <dbl> <chr> <dbl>
## 1 -7.32   38.5 1991      0
## 2 -7.64   40.5 1991      0
## 3 -7.90   40.3 1991      0
## 4 -7.26   39.3 1991      0
```

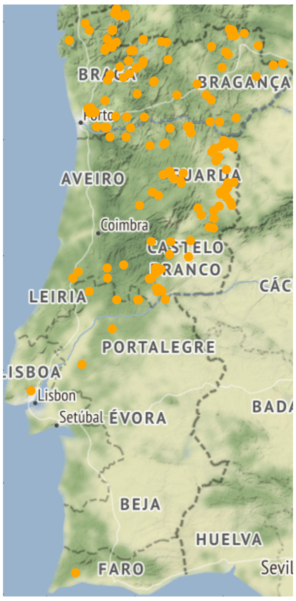
Plotting the Fires in 1999

```
library(ggmap)
pt <- get_map("Portugal",zoom=7)
data2plot <- dat[dat$year==1999 & dat$burnt == 1,]
ggmap(pt,extent="device") +
    geom_point(data=data2plot,
               aes(x=x,y=y),
               color="orange",size=3)
```



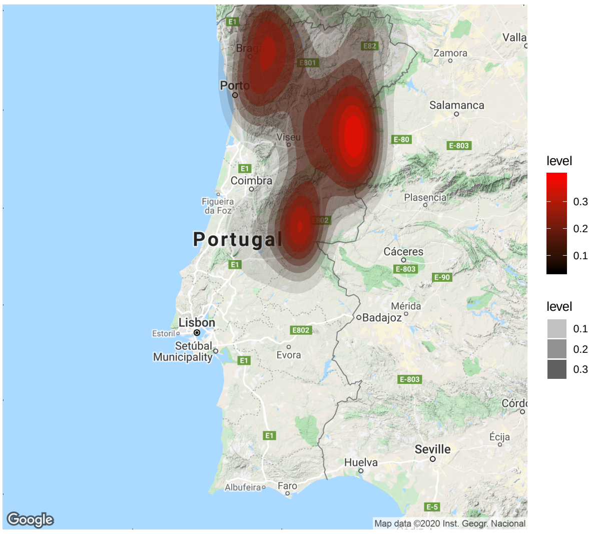
If you don't have the Google Maps API Key

```
library(ggmap)
rlat <- range(df$x)
rlon <- range(df$y)
bb <- c(left=rlat[1],bottom=rlon[1],right=rlat[2],top=rlon[2])
pt2 <- get_map(bb,source="stamen",zoom=7)
data2plot <- dat[dat$year==1999 & dat$burnt == 1,]
ggmap(pt2,extent="device") +
    geom_point(data=data2plot,
               aes(x=x,y=y),
               color="orange",size=3)
```



Adding some spatial interpolation

```
ggmap(pt,extent="device",
      base_layer=ggplot(data2plot,aes(x=x,y=y))
      ) +
    stat_density2d(aes(fill=..level..,
                       alpha=..level..),
                   bins=10,
                   geom="polygon") +
    scale_fill_gradient(low="black",high="red")
```

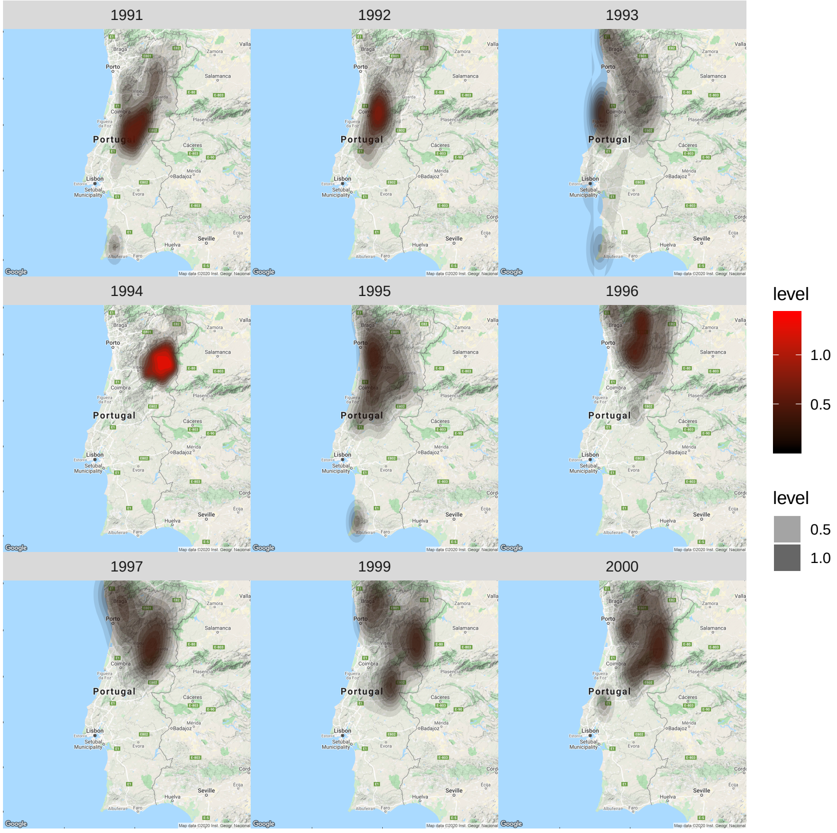


Spatio-temporal Visualization

- We need to add faceting along the year an everything else stays the same

```
data2plot <- dat[dat$burnt == 1,]
ggmap(pt,extent="device",
      base_layer=ggplot(data2plot,aes(x=x,y=y))
      ) +
    stat_density2d(aes(fill=..level.., alpha=..level..),
                   bins=10,
                   geom="polygon") +
    scale_fill_gradient(low="black",high="red") +
    facet_wrap( ~  year)
```

Spatio-temporal Visualization (cont.)



Getting geographic information of a location

- Taking advantage of the Google Maps API we can obtain information on a location

```
geocode("Ljubljana")

## # A tibble: 1 x 2
##     lon   lat
##   <dbl> <dbl>
## 1  14.5  46.1

geocode("Ljubljana",output="more")

## # A tibble: 1 x 9
##     lon   lat type     loctype    address            north south  east  west
##   <dbl> <dbl> <chr>    <chr>      <chr>              <dbl> <dbl> <dbl> <dbl>
## 1  14.5  46.1 locality approxima~ ljubljana, slove~   46.1  46.0  14.6  14.4
```

Note: Please note that the Google Maps API has several request limitations. Namely, it has an unspecified short-term rate limit as well as a 24-hour limit of 2500 requests.

Getting physical addresses from a long/lat pair

- Checking the address of one of the fire locations

```
df[1000,c("x","y")]

## # A tibble: 1 x 2
##         x     y
##     <dbl> <dbl>
## 1 -7.02   39.0

revgeocode(as.numeric(df[1000,c("x","y")]))

## [1] "Unnamed Road, 7370, Portugal"
```

Calculating distances between locations

- The distance from New York to three other cities
- Note that the default mode is driving
- Again several request limitations are imposed by Google

```
from <- rep("New York",3)
to <- c("Los Angeles","San Francisco","Toronto")
mapdist(from,to)
```

```
## # A tibble: 3 x 9
##   from     to                  m    km miles seconds minutes hours mode
##   <chr>    <chr>           <int> <dbl> <dbl>   <int>   <dbl> <dbl> <chr>
## 1 New York Los Angeles   4489747 4490. 2790.  148474   2475. 41.2  driving
## 2 New York San Francisco 4671018 4671. 2903.  155544   2592. 43.2  driving
## 3 New York Toronto        790320  790.  491.   28580    476.  7.94 driving
```

```
mapdist("Chinatown","Times Square",mode="walking")
```

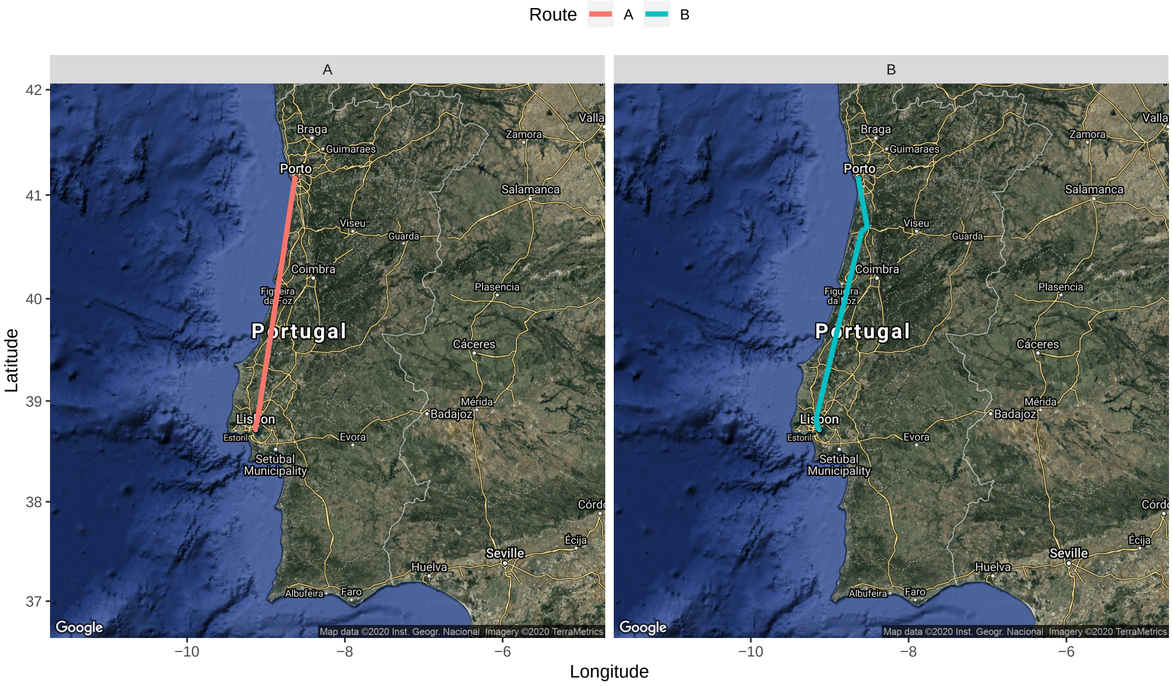
```
## # A tibble: 1 x 9
##   from      to               m    km miles seconds minutes hours mode
##   <chr>     <chr>        <int> <dbl> <dbl>   <int>   <dbl> <dbl> <chr>
## 1 Chinatown Times Square  4872  4.87  3.03    3751    62.5  1.04 walking
```

Routes between locations

- Note that the default mode is driving
- Again several request limitations are imposed by Google

```
rtDat <- route("Porto","Lisbon",mode="driving",structure="route",alternatives=TRUE)
mp <- get_map("Portugal",zoom=7,maptype="hybrid")
ggmap(mp,
      base_layer=ggplot(rtDat,aes(x=lon,y=lat,colour=route))) +
      geom_path(size=1.5,lineend="round") +
      facet_wrap(~ route) +
      labs(x = "Longitude", y = "Latitude", colour = "Route") +
      theme(legend.position = "top")
```

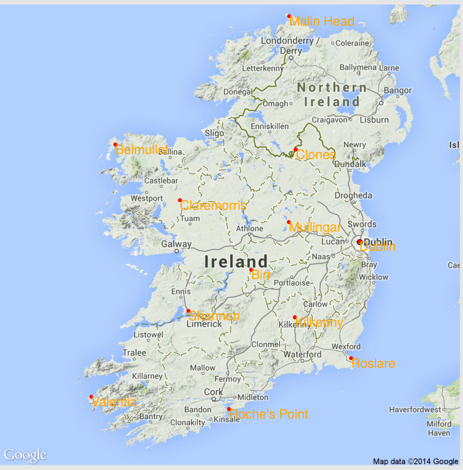
Routes between locations (cont.)



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
2. Reproduce the graph to the right



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.
4. Produce a spatio-temporal showing theses yearly averages on the stations.

Interactive spatial visualization using package `leaflet`

- Leaflet is a very popular open-source JavaScript library for interactive maps
- The R package **leaflet** allows you to create this type of graphs in R
- After installing the package we can try it using the forest fires data

Interactive spatial visualization of the forest fires

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

- The following shows the fires in 1999 in an interactive map

```
library(sp)
spatialCoords <- cbind(long=df$x, lat=df$y)
coordRefSys <- CRS("+proj=longlat +ellps=WGS84")
fires1999 <- SpatialPointsDataFrame(spatialCoords,
                                    df[, "ano1999", drop=FALSE],
                                    proj4string=coordRefSys)
library(leaflet)
leaflet() %>%
    addTiles() %>%
    addCircleMarkers(data=fires1999[fires1999$ano1999==1,])
```

Improving a bit the visualization

- Getting the addresses of the places where there were fires

```
library(ggmap)
placesWithFires <- which(fires1999$ano1999 == 1)
coord <- coordinates(fires1999)[placesWithFires,]
## Note that the Google API imposes limits on the following...
adds <- apply(coord,1,function(cs) revgeocode(cs))
## Now the interactive map (try clicking on a dot)
leaflet() %>%
    addTiles() %>%
    addCircleMarkers(data=fires1999[fires1999$ano1999==1,],
                     popup=adds,
                     clusterOptions = markerClusterOptions())
```

Learning more about leaflet

You may learn more at `https://rstudio.github.io/leaflet/`