

Ferramentas de Estatística Espacial no R para Iniciantes

25/10/2017

Jony Arrais Pinto Junior

1 - Conteúdo do minicurso

- Motivação;
- Tipos de dados;
- Sistema de Informação Geográfica (SIG);
- Utilizando o R como ferramenta para visualizar dados espaciais;
- Georreferenciando dados utilizando o R.

2 - Motivação

Toda observação possui referência **temporal** e **espacial**.

- Dados sobre o consumo do produto *H* nas cidades do estado do Rio de Janeiro;
- Dados sobre a temperatura dos bairros da zona sul do município do Rio de Janeiro;
- Dados sobre assaltos na cidade do Rio de Janeiro.

Muitos estudos **não fazem o uso da informação espacial**. No entanto, em alguns casos, essas referências espaciais são importantes na análise.

Estatística Espacial é o conjunto de métodos de análise de dados em que a localização geográfica é usada explicitamente na análise.

Principais etapas:

- Visualização;
- Análise exploratória;
- Modelagem;
- Teste de adequação do modelo.

Exemplos

1. Homicídios em Belo Horizonte

Crimes não acontecem totalmente ao acaso. É preciso um ofensor, uma vítima potencial e uma oportunidade. Existem grandes diferenças no risco de ser vítima de um crime dependendo da **idade**, do **sexo**, da **hora do dia**, do **dia da semana**, do **mês** no ano, etc.

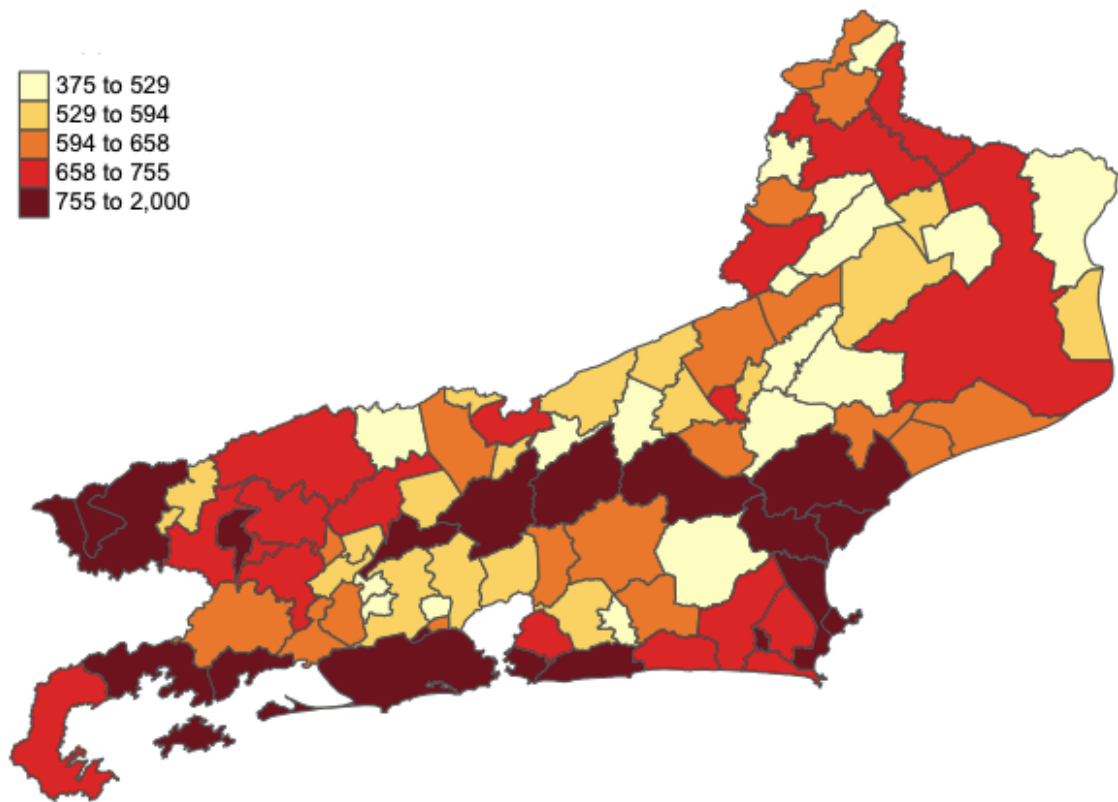
Existem também grandes **diferenças geográficas** dentro de uma cidade. Estas diferenças dependem do **tipo de crime**: crimes contra o patrimônio atingem mais as áreas ricas enquanto que crimes contra a pessoa atingem mais as áreas pobres.



Suponha, que você é o responsável por fazer a alocação do policiamento nas ruas do centro da cidade de Belo Horizonte. Você poderia estar interessado em descobrir: **quais são as áreas com maiores incidências de homicídios, existem características dessas localidades que estejam associadas com a maior incidência de homicídios.** Com isso, você conseguiria implementar, muito provavelmente, um **plano mais eficaz de combate a criminalidade** nesta região.

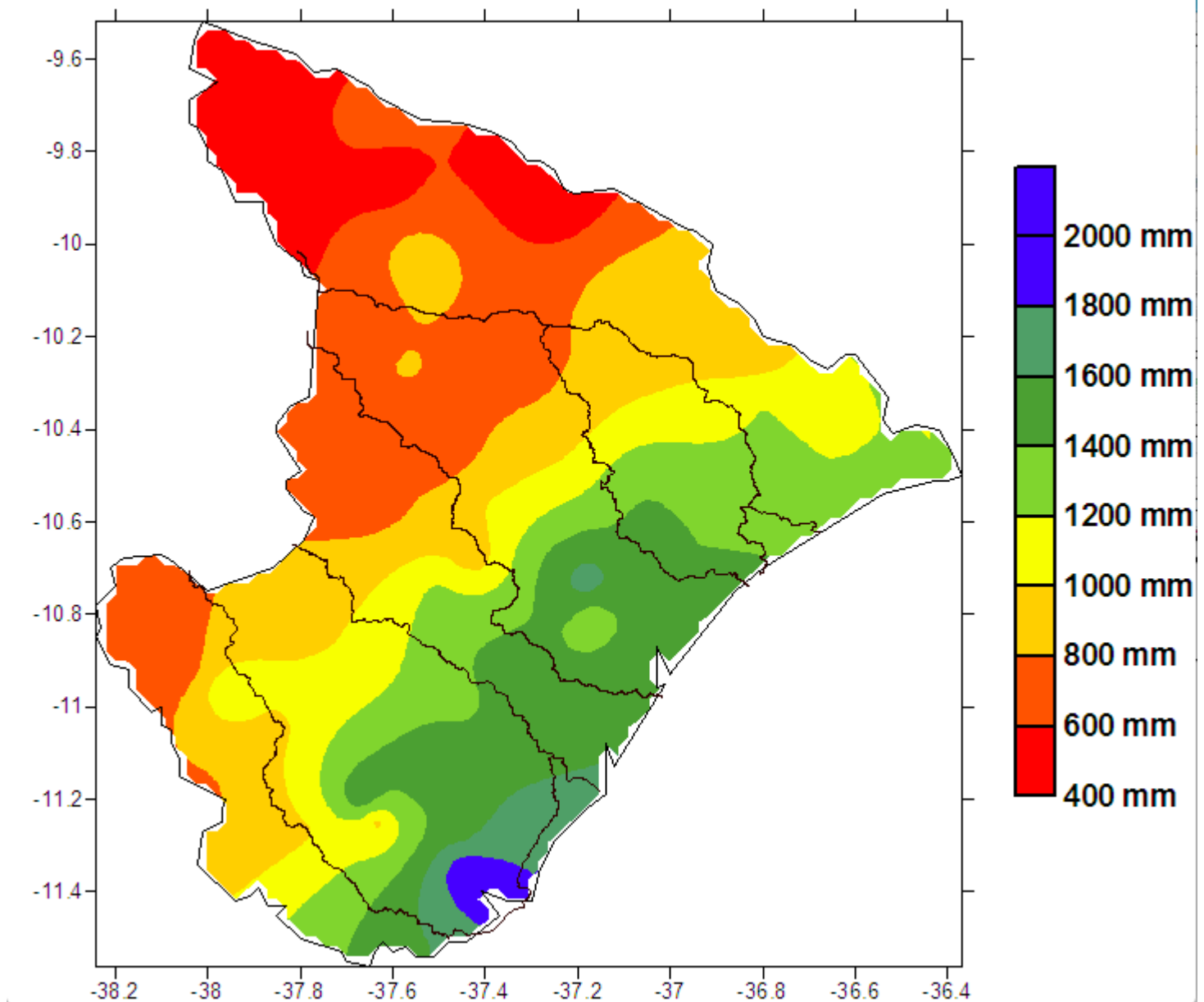
2. Vendas do produto H nas cidades do estado do Rio de Janeiro

Suponha que você esteja interessado em entender os **fatores que explicam o consumo** do produto H nas diversas cidades do estado do Rio de Janeiro. De modo, que se **existirem características dos municípios** que expliquem um maior ou menor consumo do produto, possa te ajudar a decidir em quais cidades de um outro estado você poderia iniciar a implementação de venda do produto H .



3. Precipitação anual no estado do Sergipe

Suponha que um pesquisador esteja interessado em entender os **fatores geográficos** que **influenciam a precipitação anual** no estado do Sergipe. Para fazer **previsões para regiões não observadas** e **criar um plano de criação de reservatórios inteligentes** para o abastecimento de água do estado.



Quais as diferenças entre os dados discutidos nos exemplos acima?

3 - Tipos de dados

Os dados espaciais são classificados em três grupos (Cressie, 1993):

- Padrão de pontos - homicídios em BH;
- Dados de área - vendas do produto *H* nos municípios do RJ;
- Dados de superfícies contínuas (geoestatísticos) - precipitação no estado do Sergipe.

Características destes grupos:

3.1 - Padrão de pontos

As **localizações** estão associadas à ocorrência de um **fenômeno de interesse (conhecido)** e são **aleatórias (desconhecidas)**.

Um dos objetivos é investigar se os eventos observados exibem algum **padrão sistemático** (regularidade ou

agrupamento)

Outros exemplos?

3.2 - Dados de área

Dados **agregados** em subregiões que totalizam uma dada área de interesse.

As subregiões podem ter forma regular ou irregular.

Mais de uma variável pode ser coletada nas subregiões.

Principais objetivos:

- investigar se as subregiões apresentam **padrão aleatório**;
- identificar **agrupamentos de área**;
- explorar e determinar **fatores determinantes dos padrões** identificados.

Outros exemplos?

3.3 - Dados geoestatístico

Observações de um **processo contínuo no espaço**, coletadas em um conjunto **finito de localizações**.

Principais objetivos:

- inferir sobre o **processo gerador** desses dados (media, variabilidade e estrutura de associação espacial)
- **previsão** em localidades não observadas.

Outros exemplos?

4 - Sistema de Informação Geográfica (SIG)

Um **Sistema de Informação Geográfica (SIG** ou **GIS** - Geographic Information System, do inglês) é um sistema de hardware, software, com procedimentos computacionais que permitem e facilitam a análise, gestão ou representação do espaço e dos fenômenos que nele ocorrem.

São exemplos de GIS - **ArcGIS, QGIS, GeoDa, GRASS GIS, Google Earth Pro**, etc...

O software R se apresenta como uma ótima ferramenta de modelagem, manipulação e visualização de dados.

É possível **combinar a potencialidade do GIS com o R**.

O [Task View](#) são temas ou categorias que agrupam pacotes com base em suas funcionalidades. Existe uma categoria relacionada com **Estatística Espacial** que merece ser investigada.

5 - Utilizando o R como ferramenta para visualizar dados espaciais.

Importando arquivos *shapefile* para o R.

```
#Carregando o pacote rgdal
library(rgdal)

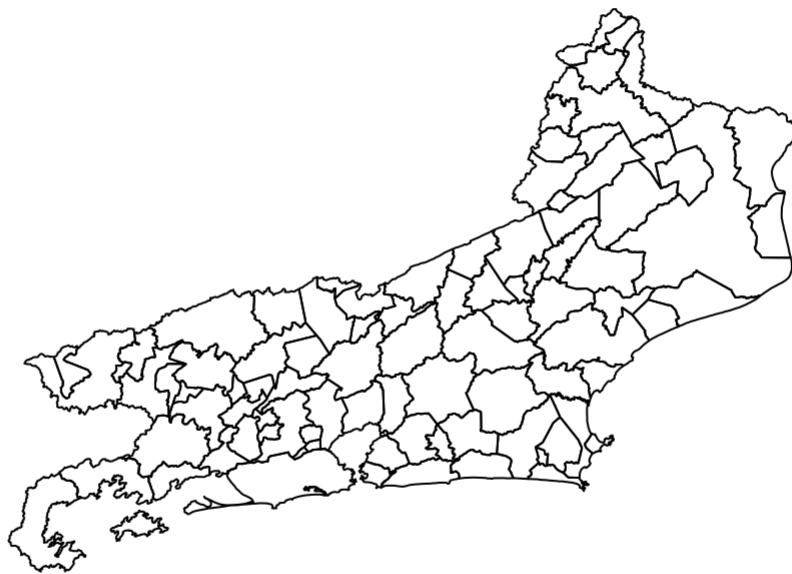
## Lendo o shapefile com os municípios do Rio de Janeiro
## O shapefile pode ser baixado em http://downloads.ibge.gov.br/downloads_geociencias.h
tm pelo caminho
## organizacao_do_territorio/malhas_territoriais/malhas_municipais/municipio_2015/UFs/R
```

```
J/rj_municipios.zip
rio = readOGR("RJ_Mun97_region.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "RJ_Mun97_region.shp", layer: "RJ_Mun97_region"
## with 91 features
## It has 19 fields
```

- Função `readOGR(...)`: faz a leitura do arquivo shapefile.

```
## Plotando o shapefile com todos os municípios do estado do Rio de Janeiro
plot(rio)
```



- Função `plot(...)`: faz o plot do arquivo shapefile.

```
## Informações gerais sobre o shapefile
class(rio)
```

```
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
names(rio)
```

```
## [1] "CODMUN6" "MUNIC_PIO" "SEM_ACENTO" "MUNIC_PIO" "UF"
## [6] "UF_IBGE" "REGI_O" "MESO_IBGE" "MESO_IBG0" "MICRO_IBGE"
## [11] "MICRO_IB0" "MESO_IBG1" "MICRO_IB1" "AREA_97" "SEDE"
```

```
## [16] "SEDE0"      "N91"        "N96"        "CODCOMP"
```

```
head(rio@data)
```

```
##      CODMUN6  MUNIC_PIO SEM_ACENTO MUNIC_PIO UF UF_IBGE REGI_O      MESO_IBGE
## 0   330380    PARATI    PARATI    330380 RJ      33      SE SUL FLUMINENSE
## 1   330420    RESENDE    RESENDE    330420 RJ      33      SE SUL FLUMINENSE
## 2   330225    ITATIAIA    ITATIAIA    330225 RJ      33      SE SUL FLUMINENSE
## 3   330411    PORTO REAL    PORTO REAL    330411 RJ      33      SE SUL FLUMINENSE
## 4   330440    RIO CLARO    RIO CLARO    330440 RJ      33      SE SUL FLUMINENSE
## 5   330412    QUATIS    QUATIS    330412 RJ      33      SE SUL FLUMINENSE
##      MESO_IBG0      MICRO_IBGE MICRO_IB0 MESO_IBG1 MICRO_IB1
## 0           05      BAIA DA ILHA GRANDE      013      3305      330513
## 1           05 VALE DO PARAIBA FLUMINENSE      011      3305      330511
## 2           05 VALE DO PARAIBA FLUMINENSE      011      3305      330511
## 3           05 VALE DO PARAIBA FLUMINENSE      011      3305      330511
## 4           05 VALE DO PARAIBA FLUMINENSE      011      3305      330511
## 5           05 VALE DO PARAIBA FLUMINENSE      011      3305      330511
##      AREA_97      SEDE      SEDE0      N91      N96      CODCOMP
## 0  928.4739 -44.713 -23.218 24525 1003699 330501303807
## 1 1113.4390 -44.447 -22.469 94307  93961 330501104201
## 2  224.9654 -44.563 -22.496 16711  21216 330501102254
## 3   50.6217 -44.290 -22.420    0   8664 330501104110
## 4  841.3873 -44.136 -22.723 13724  14449 330501104409
## 5  286.2088 -44.258 -22.407  8916   9866 330501104128
```

- Função `class(...)`: apresenta a classe do objeto.
- Função `names(...)`: apresenta os nomes das variáveis contidas no objeto.
- Função `head(...)`: apresenta as linhas iniciais de um data frame.

```
## Lendo o arquivo de dados
## Esses dados foram simulados representam a venda do produto H e a renda dos municípios do estado do RJ
dados_aux <- read.table("dados.txt", header=TRUE)

head(dados_aux)
```

```
##      Cod Satisfacao Venda_1SEM Venda_2SEM
## 1 330412      16.4      535.96   135.7124
## 2 330395      16.9      657.99   467.3231
## 3 330280      16.4      649.40   652.3771
## 4 330360      15.9      580.49   594.6061
## 5 330555      16.5      604.82   617.1833
## 6 330620      15.8      701.06   706.8891
```

```
class(dados_aux)
```

```
## [1] "data.frame"
```

- Função `read.table(...)`: importa um arquivo .txt.

```
## Associando os municípios do Rio de Janeiro a satisfação media e número de produtos H
```


vendidos

```
## As informações são relacionadas a partir do código único para cada município (coluna CODMUN6 no objeto rio e coluna Cod no objeto dados_aux)
```

```
dados_rio <- merge(rio, dados_aux, by.x = "CODMUN6", by.y = "Cod")
```

```
#Convertendo em projeções
```

```
dados_rio <- spTransform(dados_rio, CRS("+proj=longlat +datum=WGS84 +no_defs"))
```

```
head(dados_rio@data)
```

```
##      CODMUN6  MUNIC_PIO SEM_ACENTO MUNIC_PIO UF UF_IBGE REGI_O
## 52   330380    PARATI    PARATI    330380 RJ      33      SE
## 62   330420    RESENDE    RESENDE    330420 RJ      33      SE
## 34   330225    ITATIAIA    ITATIAIA    330225 RJ      33      SE
## 58   330411 PORTO REAL PORTO REAL    330411 RJ      33      SE
## 64   330440    RIO CLARO    RIO CLARO    330440 RJ      33      SE
## 59   330412    QUATIS    QUATIS    330412 RJ      33      SE
##      MESO_IBGE MESO_IBG0      MICRO_IBGE MICRO_IB0 MESO_IBG1
## 52 SUL FLUMINENSE      05      BAIA DA ILHA GRANDE      013      3305
## 62 SUL FLUMINENSE      05 VALE DO PARAIBA FLUMINENSE      011      3305
## 34 SUL FLUMINENSE      05 VALE DO PARAIBA FLUMINENSE      011      3305
## 58 SUL FLUMINENSE      05 VALE DO PARAIBA FLUMINENSE      011      3305
## 64 SUL FLUMINENSE      05 VALE DO PARAIBA FLUMINENSE      011      3305
## 59 SUL FLUMINENSE      05 VALE DO PARAIBA FLUMINENSE      011      3305
##      MICRO_IB1  AREA_97  SEDE  SEDE0  N91  N96  CODCOMP
## 52      330513  928.4739 -44.713 -23.218 24525 1003699 330501303807
## 62      330511 1113.4390 -44.447 -22.469 94307  93961 330501104201
## 34      330511  224.9654 -44.563 -22.496 16711  21216 330501102254
## 58      330511   50.6217 -44.290 -22.420    0   8664 330501104110
## 64      330511  841.3873 -44.136 -22.723 13724 14449 330501104409
## 59      330511  286.2088 -44.258 -22.407  8916   9866 330501104128
##      Satisfacao Venda_1SEM Venda_2SEM
## 52      13.40      734.36  139.9393
## 62      13.85      915.21  931.2893
## 34      13.80      773.48  776.7658
## 58      15.50      577.07  587.4925
## 64      16.90      623.62  636.2800
## 59      16.40      535.96  135.7124
```

- Função `merge(...)`: combina duas bases por meio de uma variável.
- Função `CRS(...)`: cria um objeto pertencente a classe do sistema de referências coordenadas.

```
#Carregando os pacotes RColorBrewer e sp
```

```
library(RColorBrewer)
```

```
library(sp)
```

```
# Mapas coropléticos
```

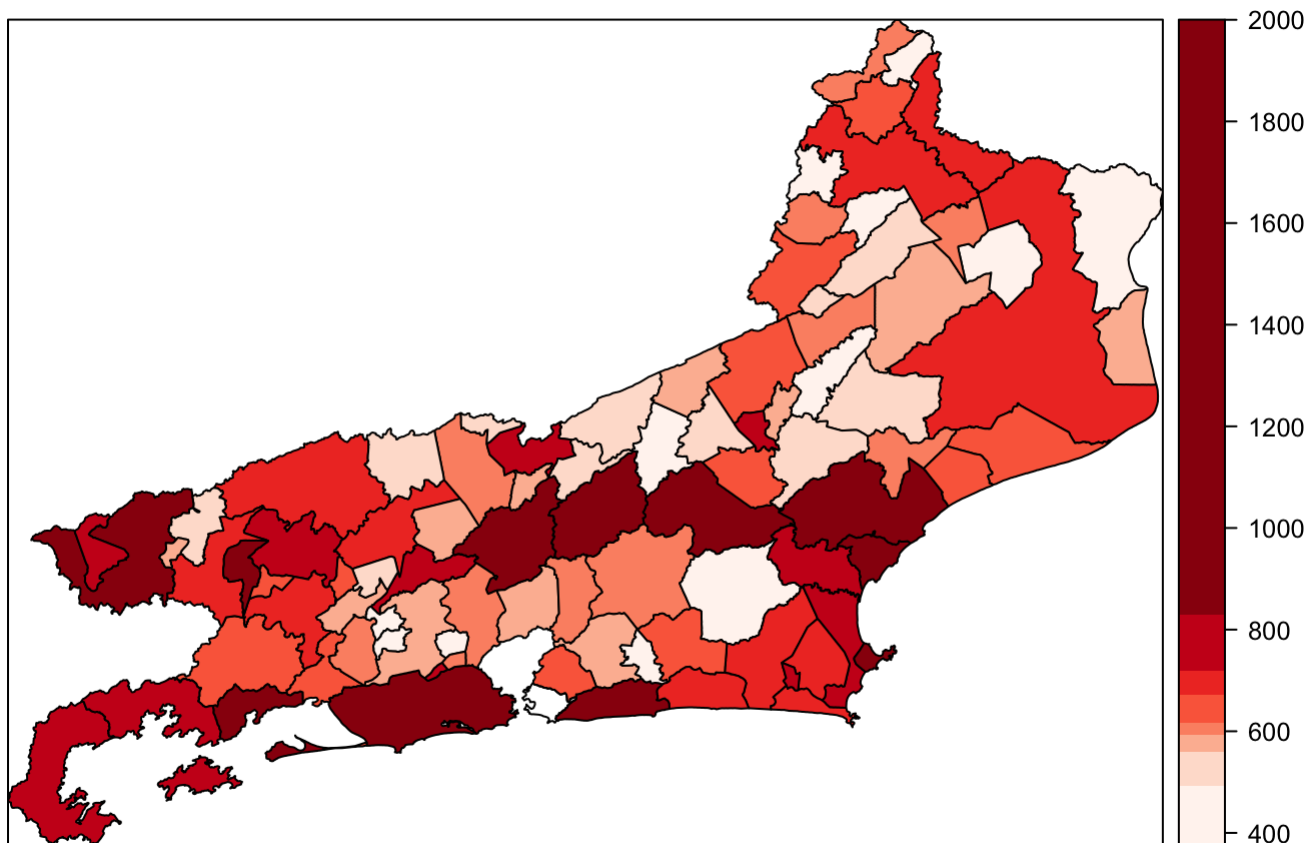
```
# Renda - mapa coroplético
```

```
#Criando os intervalos com base em quantis
```

```
intervalos=quantile(dados_rio$Venda_1SEM, probs = seq(0,1,0.125))
```

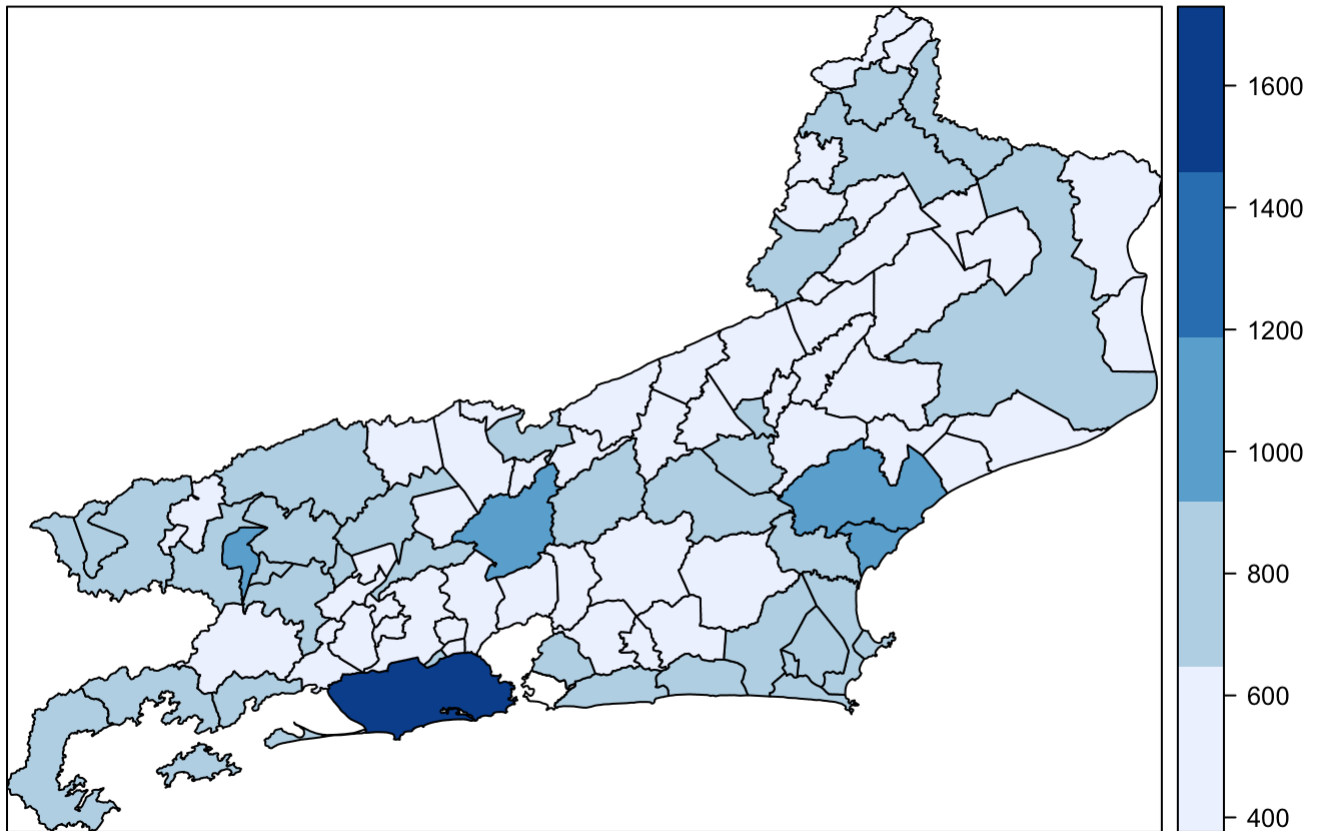
```
#Plotando o mapa com tons avermelhados
```

```
spplot(dados_rio,c("Venda_1SEM"),at=intervalos,col.regions =brewer.pal(8, "Reds")) #Out  
ras opções de cores: Greens, BrBG, Accent
```

```
#Criando os intervalos de mesmo tamanho
ampli_dados = max(dados_rio$Venda_1SEM) - min(dados_rio$Venda_1SEM)
k = 6 #número de classes + 1
ampli_classe = ampli_dados/k
intervalos2 = NULL
intervalos2[1] = min(dados_rio$Venda_1SEM)
for(i in 2:k){
  intervalos2[i] = intervalos2[i-1] + ampli_classe
}

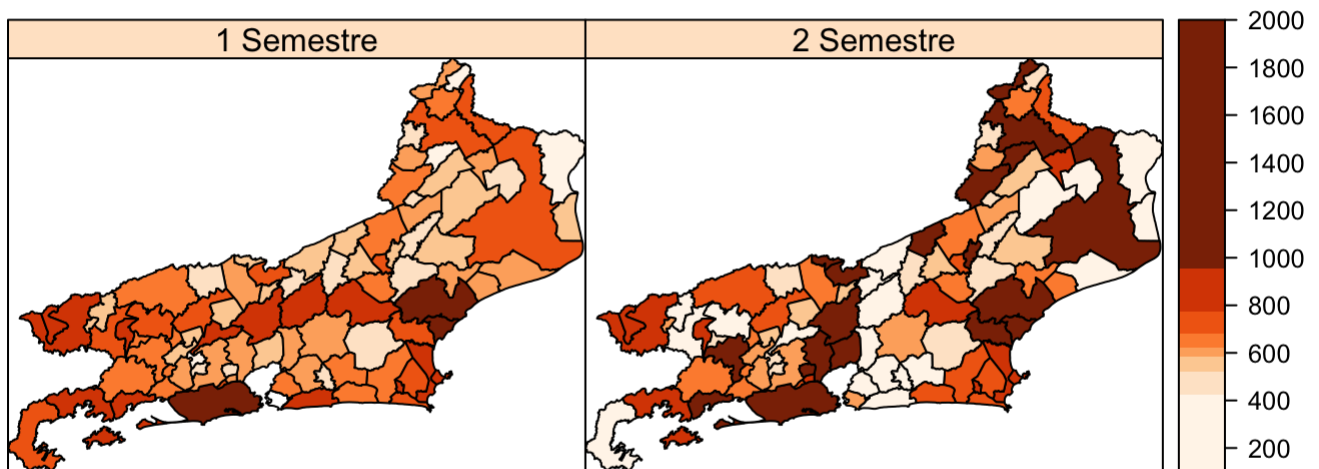
#Plotando o mapa com tons azulados
spplot(dados_rio,c("Venda_1SEM"),at=intervalos2,col.regions =brewer.pal(5, "Blues"))
```



```
# Comparando dois mapas

#Criando os intervalos com base em quantis
intervalos3=quantile(c(dados_rio$Venda_1SEM,dados_rio$Venda_2SEM), probs = seq(0,1,0.125))

#Plotando o mapa com tons alaranjados
spplot(dados_rio,c("Venda_1SEM","Venda_2SEM"),at=intervalos3,col.regions =brewer.pal(8,"Oranges"),names.attr = c("1 Semestre","2 Semestre"))
```



- Função `splot(...)`: plota um layout coroléptico.

```
#Carregabndo o pacote plotGoogleMaps
library(plotGoogleMaps)

# Pacote plotGoogleMaps
# http://www2.uaem.mx/r-mirror/web/packages/plotGoogleMaps/vignettes/plotGoogleMaps-intro.pdf

#Criando um dataframe com os dados Venda_1SEM, Satisfacao e nome do município sem acento
dados_google <- dados_rio[, c("Venda_1SEM", "Satisfacao", "SEM_ACENTO")]

#Modificando o nome das variáveis
names(dados_google) <- c("Venda (1 Semestre)", "Satisfacao com o Produto", "Municipio")

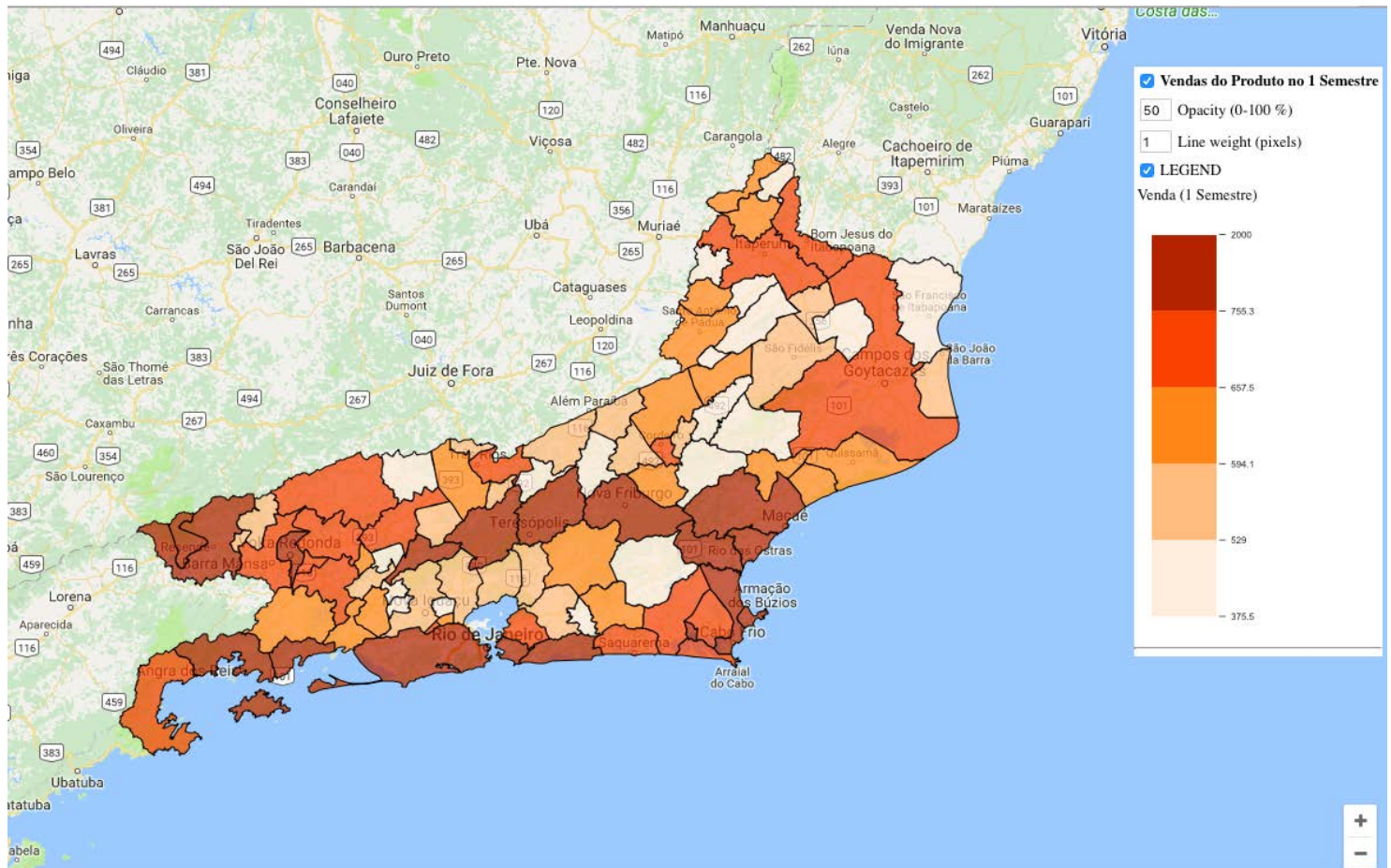
#Visualizando os dados
head(dados_google)
```

```
## class      : SpatialPolygonsDataFrame
## features   : 6
## extent     : -44.88834, -43.87912, -23.3671, -22.23994 (xmin, xmax, ymin, ymax)
## coord. ref.: +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## variables  : 3
## names      : Venda (1 Semestre), Satisfacao com o Produto, Municipio
## min values :          535.96,          13.40, ITATIAIA
## max values :          915.21,          16.90, RIO CLARO
```

```
#Plotando o mapa do número de vendas no 1 semestre
```

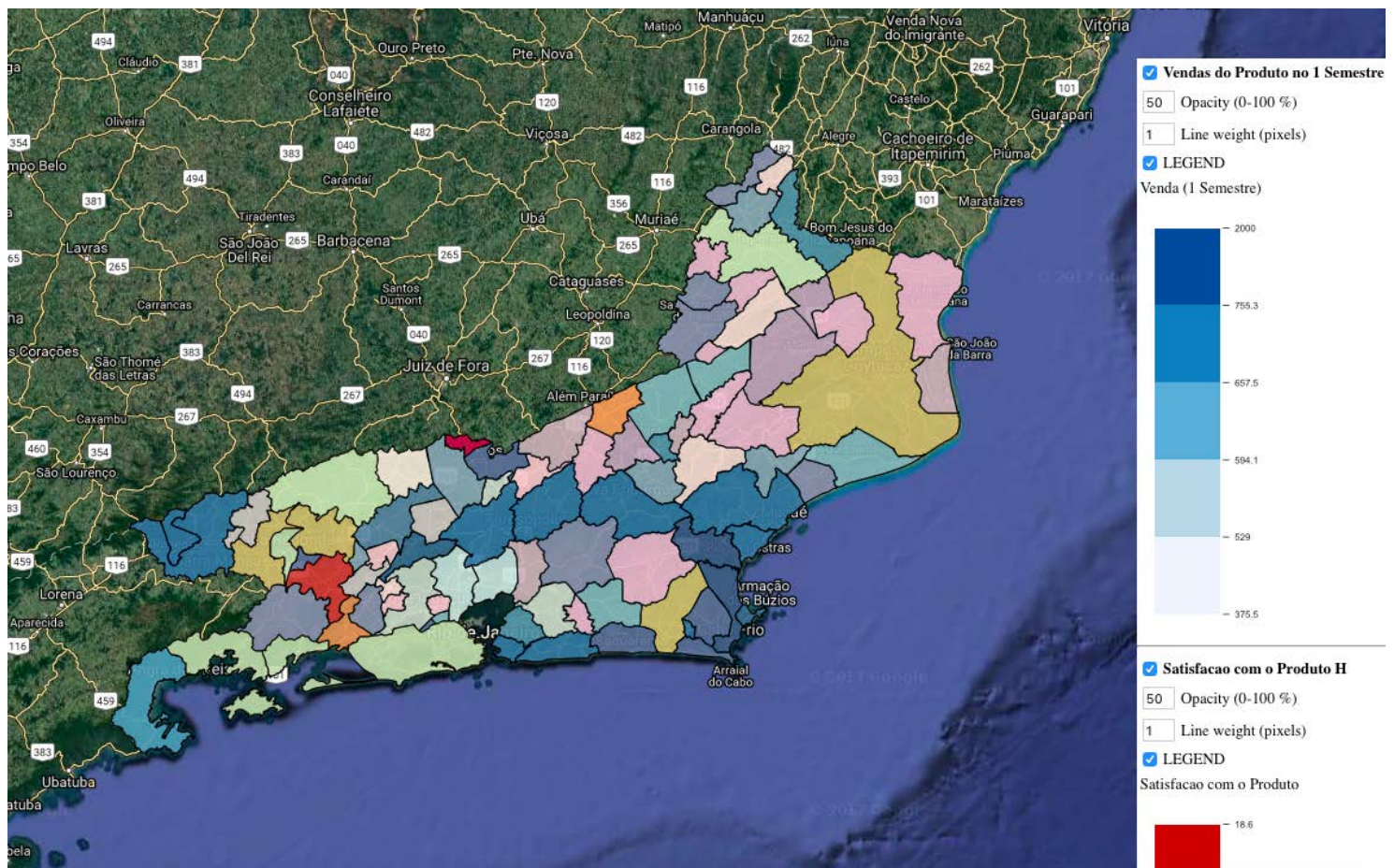
```
map <- plotGoogleMaps(dados_google, zcol = "Venda (1 Semestre)", zoom = 8, fitBounds = F, filename = "Map_GoogleMaps.html", layerName = "Vendas do Produto no 1 Semestre", colPalette = brewer.pal(5, "Oranges"), mapTypeId = "ROADMAP") #Outras opções de mapTypeId: HYBRID, SATELLITE, TERRAIN
```

- Função `plotGoogleMaps(...)`: plota o gráfico coroléptico da renda no google maps.



```
#Adicionando a Satisfacao
map2 <- plotGoogleMaps(dados_google, zcol = "Venda (1 Semestre)", zoom = 8, fitBounds = F, filename = "Map_GoogleMaps2.html", layerName = "Vendas do Produto no 1 Semestre", colPalette = brewer.pal(5, "Blues"), add=TRUE)

map3 <- plotGoogleMaps(dados_google, zcol = "Satisfacao com o Produto", zoom = 8, fitBounds = F, filename = "Map_GoogleMaps3.html", layerName = "Satisfacao com o Produto H", colPalette = brewer.pal(5, "YlOrRd"), previousMap = map2)
```

```
#Carregando pacotes
library(rworldmap)
library(RgoogleMaps)
library(googleVis)

# Leitura dos dados
laurus = read.table("laurus.txt", header=TRUE)

#Visualizando os dados
head(laurus)
```

```
##      country      lat      lon optional
## 1      France 43.76502  4.74985      TRUE
## 2      France 43.76426  4.74886      TRUE
## 3       Italy 41.45709 13.19514      TRUE
## 4      France 42.64391  2.52931      TRUE
## 5 New Zealand -41.29837 174.76669      TRUE
## 6       Italy 44.90357 11.66037      TRUE
```

```
#Verificando a classe do objeto
class(laurus)
```

```
## [1] "data.frame"
```

```
#Transformar o data frame com os dados em um objeto espacial
coordinates(laurus) <- c("lon", "lat")

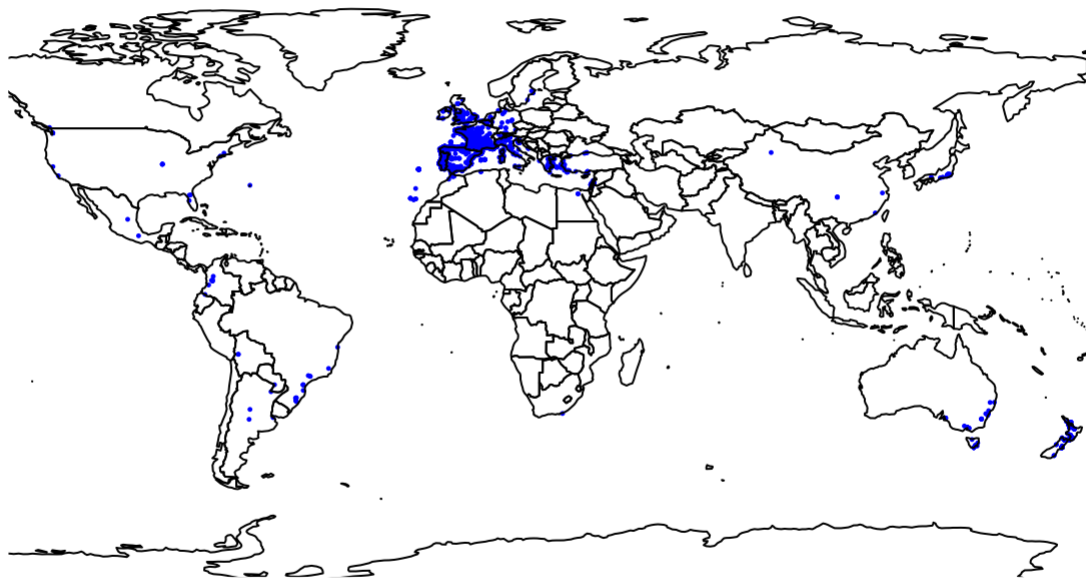
# Definir projeção espacial
crs.geo <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84") # geographical, datum WGS84
```

```
proj4string(laurus) <- crs.geo
```

```
#Verificando a classe do objeto
class(laurus)
```

```
## [1] "SpatialPointsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
# Mapa das ocorrências
mapa_mundo <- getMap()
plot(laurus, pch = 19, cex = 0.2, col = "blue")
plot(mapa_mundo, add = TRUE)
```



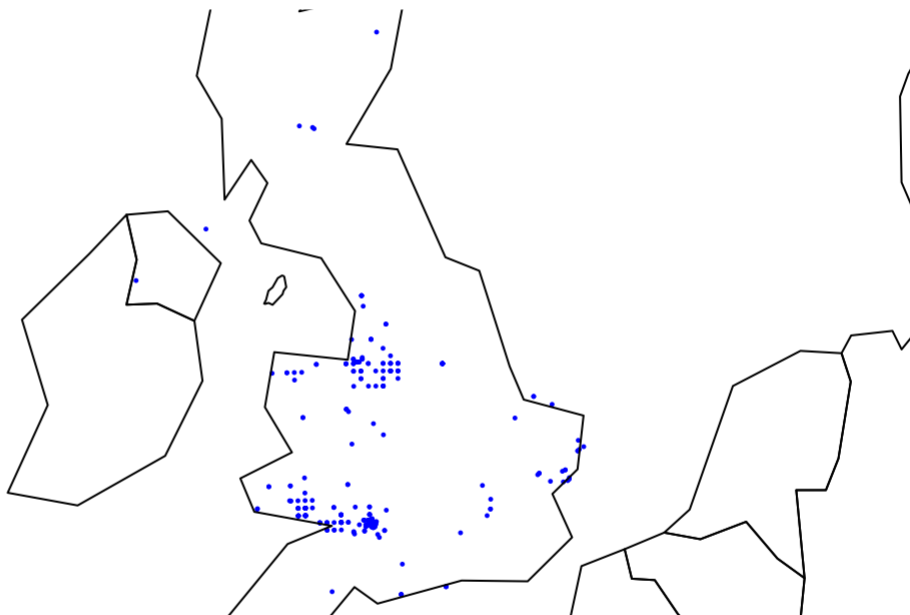
```
# Número de ocorrências por país
table(laurus$country)
```

```
##
##      Algeria      Argentina      Australia      Belgium      Bermuda
##           3           3           39           7           1
##      Bolivia      Brazil      Canada      China      Colombia
##           2          16           1           6           7
##      Croatia      Cyprus      Egypt      France      Germany
##          16           1           2      3617          17
##      Greece      Ireland      Isle of Man      Israel      Italy
##          30           7           2      1340          59
```

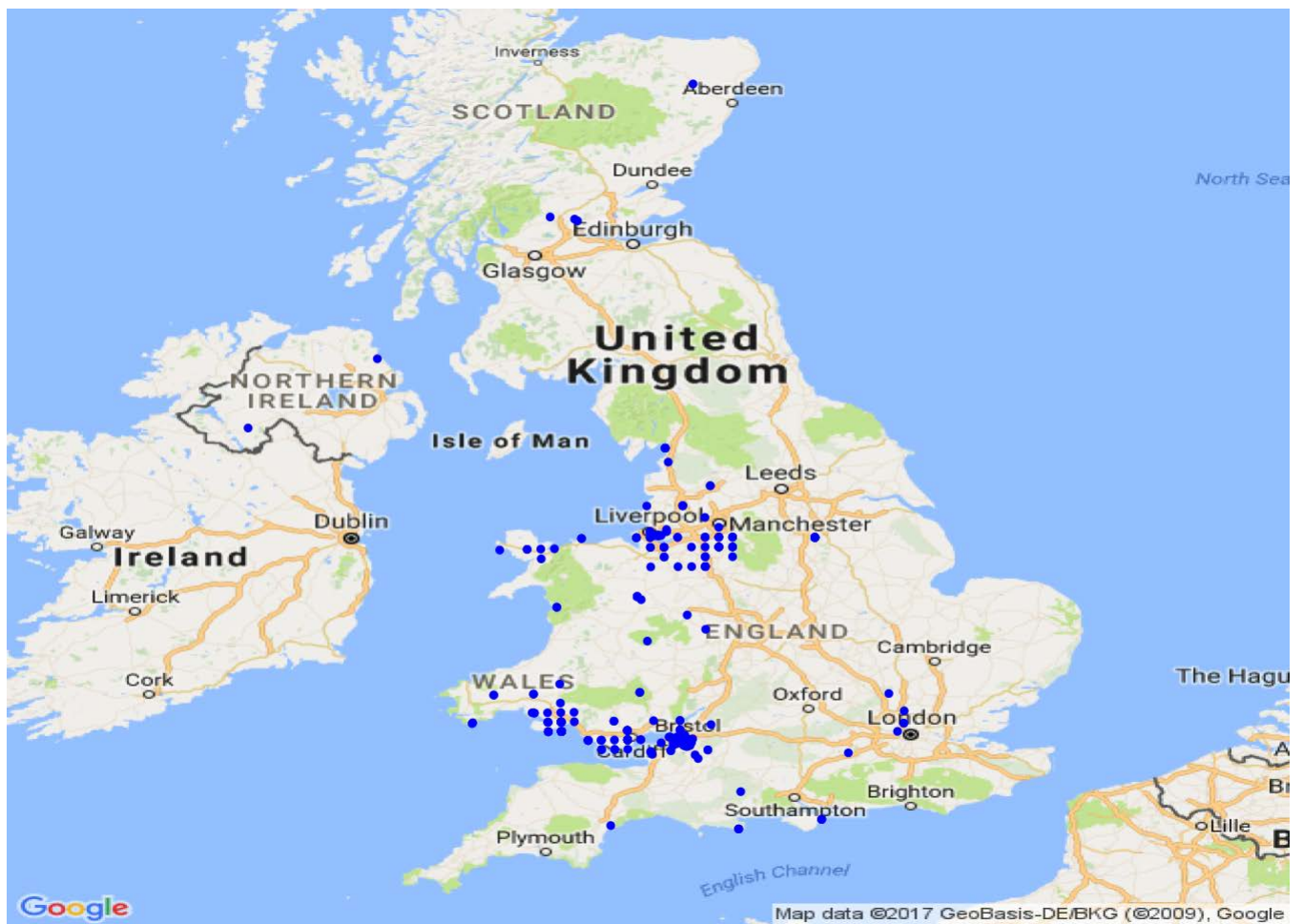
##	Japan	Lebanon	Liechtenstein	Mexico	Morocco
##	19	2	1	2	10
##	Netherlands	New Zealand	Palestina	Paraguay	Portugal
##	4	142	3	1	478
##	Slovenia	South Africa	Spain	Sweden	Switzerland
##	1	1	2203	7	6
##	Turkey	United Kingdom	United States	Vatican City	
##	19	320	17	1	

```
# Laurus nobilis no Reino Unido
laurus_uk = subset(laurus, country == "United Kingdom")
```

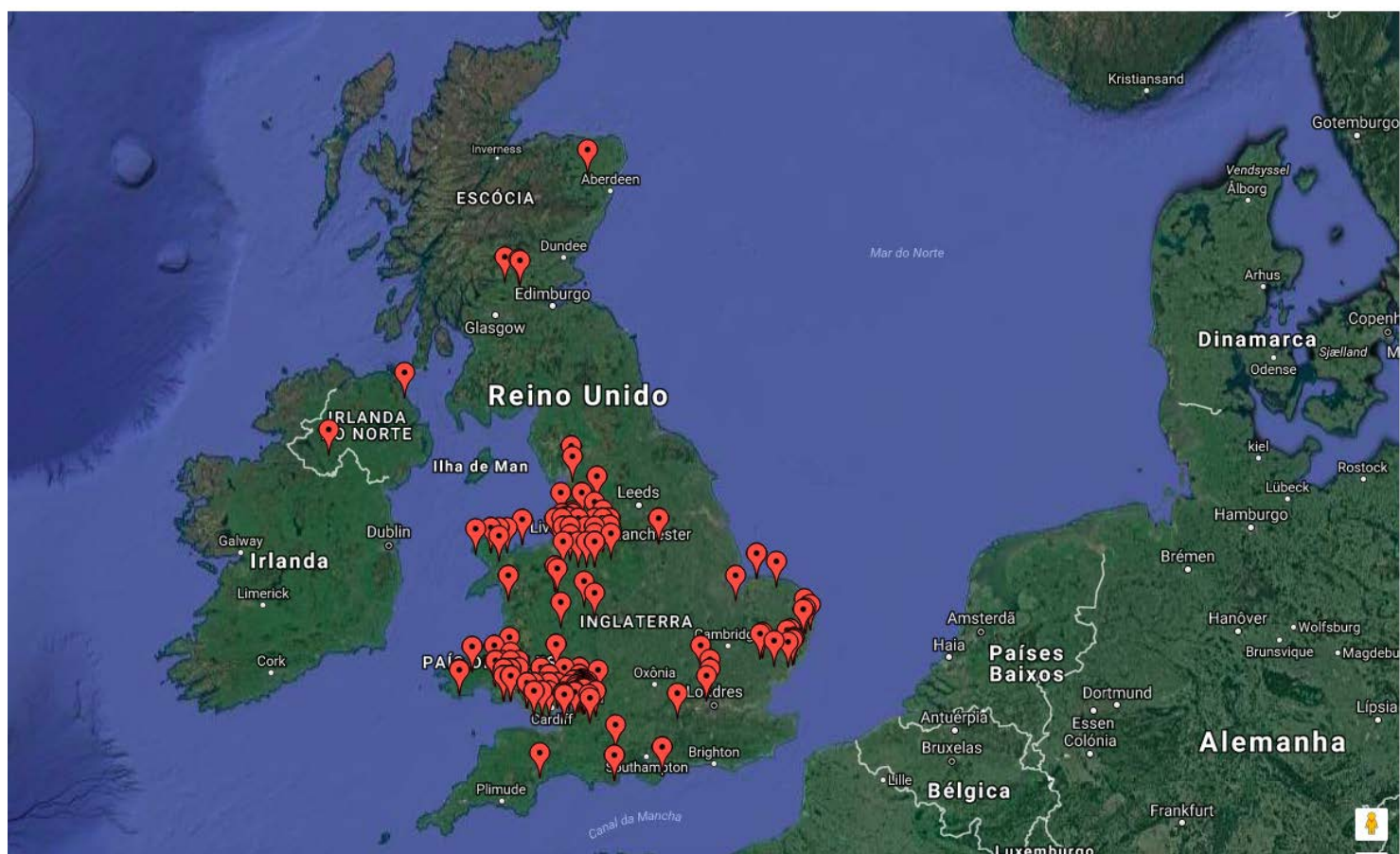
```
# Mapa da distribuição das ocorrências
plot(laurus_uk, pch = 19, cex = 0.2, col = "blue")
plot(mapa_mundo, add=TRUE)
```



```
# Mapa usando RgoogleMaps
laurus_uk_coords <- as.data.frame(coordinates(laurus_uk))
mapa_rgml <- GetMap(center = c(54, -3), zoom = 6, maptype = "mobile") # "roadmap", "satellite", "terrain", "hybrid", "mapmaker-roadmap", "mapmaker-hybrid"
PlotOnStaticMap(mapa_rgml, lat = laurus_uk_coords$lat, lon = laurus_uk_coords$lon, cex = 0.5, pch = 19, col = "blue", FUN = points)
```

```
#Mapas com googleVis (internet)
pontos_uk <- as.data.frame(laurus_uk)
pontos_uk$latlon <- paste(pontos_uk$lat, pontos_uk$lon, sep=":")
mapa_uk <- gvisMap(pontos_uk, locationvar = "latlon", tipvar = "country",
  options = list(showTip = T, showLine = F, enableScrollWheel = TRUE,
    useMapTypeControl = T, width = 1400, height = 800))
plot(mapa_uk)
```



```
#Carregando pacotes
```

```
library(spdep)
```

```
## Descobrimos os vizinhos de cada região
```

```
vizinhos <- poly2nb(rio, queen = TRUE)
```

```
vizinhos
```

```
## Neighbour list object:
```

```
## Number of regions: 91
```

```
## Number of nonzero links: 444
```

```
## Percentage nonzero weights: 5.361671
```

```
## Average number of links: 4.879121
```

```
#Matriz de vizinhanças
```

```
plot(rio)
```

```
plot(vizinhos, coordinates(rio), add=T, pch = 19)
```



```
## Matriz de vizinhanças (W)
W_rio <- nb2mat(vizinhos, style = "B") # outras opções de estilo: W
colnames(W_rio) <- rownames(W_rio)

head(W_rio)
```

```
##      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 1  0  0  1  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 3  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##      27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
## 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##      51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
## 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##      75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
## 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
```

```
## 1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
## 2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 3  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
## 4  0  0  0  1  1  0  0  0  0  0  1  1  0  0  0  1
## 5  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0
```

```
#Carregando pacote
library(leaflet)

## Carregando conjunto de dados
## Localizações e concentrações de metais pesados da camada superficial, juntamente com
uma série de variáveis de solo e
## paisagem nos locais de observação, numa planície de inundação do rio Meuse.
data("meuse")

## Carregando limites do rio Meuse
data(meuse.riv)

## Informações gerais sobre o conjunto de dados
names(meuse)
```

```
## [1] "x"      "y"      "cadmium" "copper" "lead"    "zinc"    "elev"
## [8] "dist"   "om"     "ffreq"   "soil"   "lime"    "landuse" "dist.m"
```

```
str(meuse)
```

```
## 'data.frame':    155 obs. of  14 variables:
## $ x      : num  181072 181025 181165 181298 181307 ...
## $ y      : num  333611 333558 333537 333484 333330 ...
## $ cadmium: num  11.7 8.6 6.5 2.6 2.8 3 3.2 2.8 2.4 1.6 ...
## $ copper  : num  85 81 68 81 48 61 31 29 37 24 ...
## $ lead    : num  299 277 199 116 117 137 132 150 133 80 ...
## $ zinc    : num  1022 1141 640 257 269 ...
## $ elev    : num  7.91 6.98 7.8 7.66 7.48 ...
## $ dist    : num  0.00136 0.01222 0.10303 0.19009 0.27709 ...
## $ om      : num  13.6 14 13 8 8.7 7.8 9.2 9.5 10.6 6.3 ...
## $ ffreq   : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
## $ soil    : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2 1 1 2 ...
## $ lime    : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
## $ landuse : Factor w/ 15 levels "Aa","Ab","Ag",...: 4 4 4 11 4 11 4 2 2 15 ...
## $ dist.m  : num  50 30 150 270 380 470 240 120 240 420 ...
```

```
summary(meuse)
```

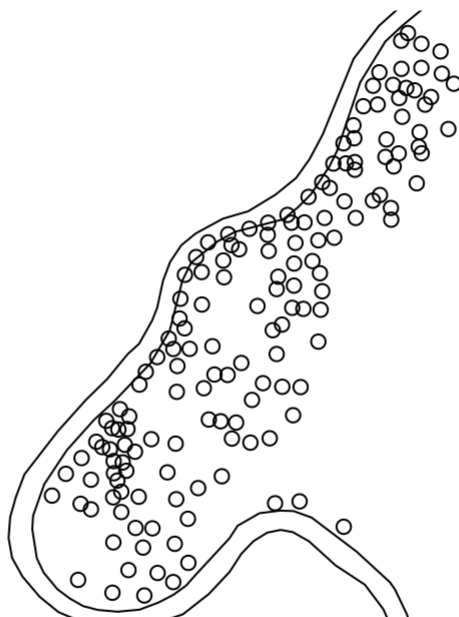
```
##           x              y          cadmium          copper
## Min.      :178605   Min.    :329714   Min.      : 0.200   Min.      : 14.00
## 1st Qu.:179371   1st Qu.:330762   1st Qu.: 0.800   1st Qu.: 23.00
## Median :179991   Median :331633   Median : 2.100   Median : 31.00
## Mean     :180005   Mean     :331635   Mean     : 3.246   Mean     : 40.32
## 3rd Qu.:180630   3rd Qu.:332463   3rd Qu.: 3.850   3rd Qu.: 49.50
## Max.     :181390   Max.      :333611   Max.     :18.100   Max.     :128.00
##
##           lead          zinc          elev          dist
## Min.      : 37.0   Min.      : 113.0   Min.      : 5.180   Min.      :0.00000
```

```
## 1st Qu.: 72.5    1st Qu.: 198.0    1st Qu.: 7.546    1st Qu.:0.07569
## Median :123.0    Median : 326.0    Median : 8.180    Median :0.21184
## Mean   :153.4    Mean   : 469.7    Mean   : 8.165    Mean   :0.24002
## 3rd Qu.:207.0    3rd Qu.: 674.5    3rd Qu.: 8.955    3rd Qu.:0.36407
## Max.   :654.0    Max.   :1839.0    Max.   :10.520    Max.   :0.88039
##
##      om      ffreq  soil   lime      landuse      dist.m
## Min.   : 1.000    1:84   1:97   0:111   W         :50   Min.   : 10.0
## 1st Qu.: 5.300    2:48   2:46   1: 44   Ah        :39   1st Qu.: 80.0
## Median : 6.900    3:23   3:12           Am        :22   Median : 270.0
## Mean   : 7.478                Fw        :10   Mean   : 290.3
## 3rd Qu.: 9.000                Ab         : 8   3rd Qu.: 450.0
## Max.   :17.000                (Other):25   Max.   :1000.0
## NA's   :2                NA's     : 1
```

```
## Transformando os dados em um objeto espacial
coordinates(meuse)<-c("x","y")
class(meuse)
```

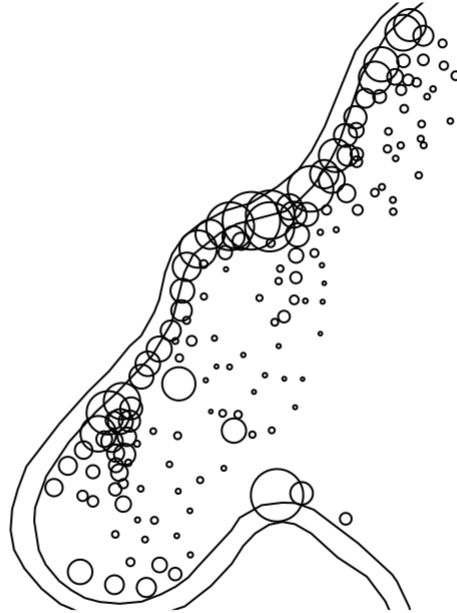
```
## [1] "SpatialPointsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
## Mapas exploratórios
plot(meuse, asp = 1, pch = 1)
lines(meuse.riv)
```

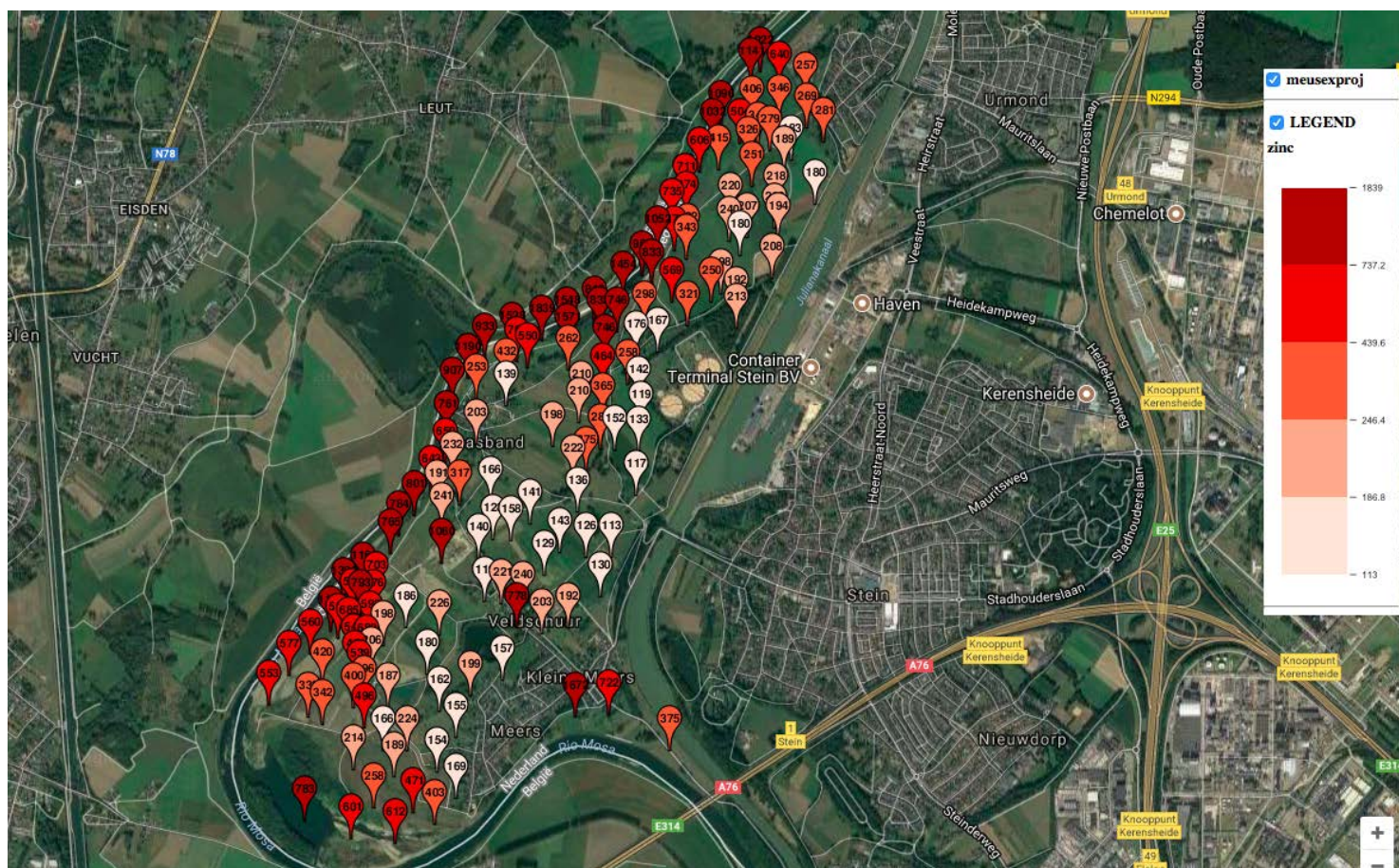


```
# Mapa 1 (dados originais para facilitar a visualização das diferenças entre localizações)
```

```
plot(meuse, asp = 1, cex = 4 * meuse$zinc/max(meuse$zinc), pch = 1)  
lines(meuse.riv)
```

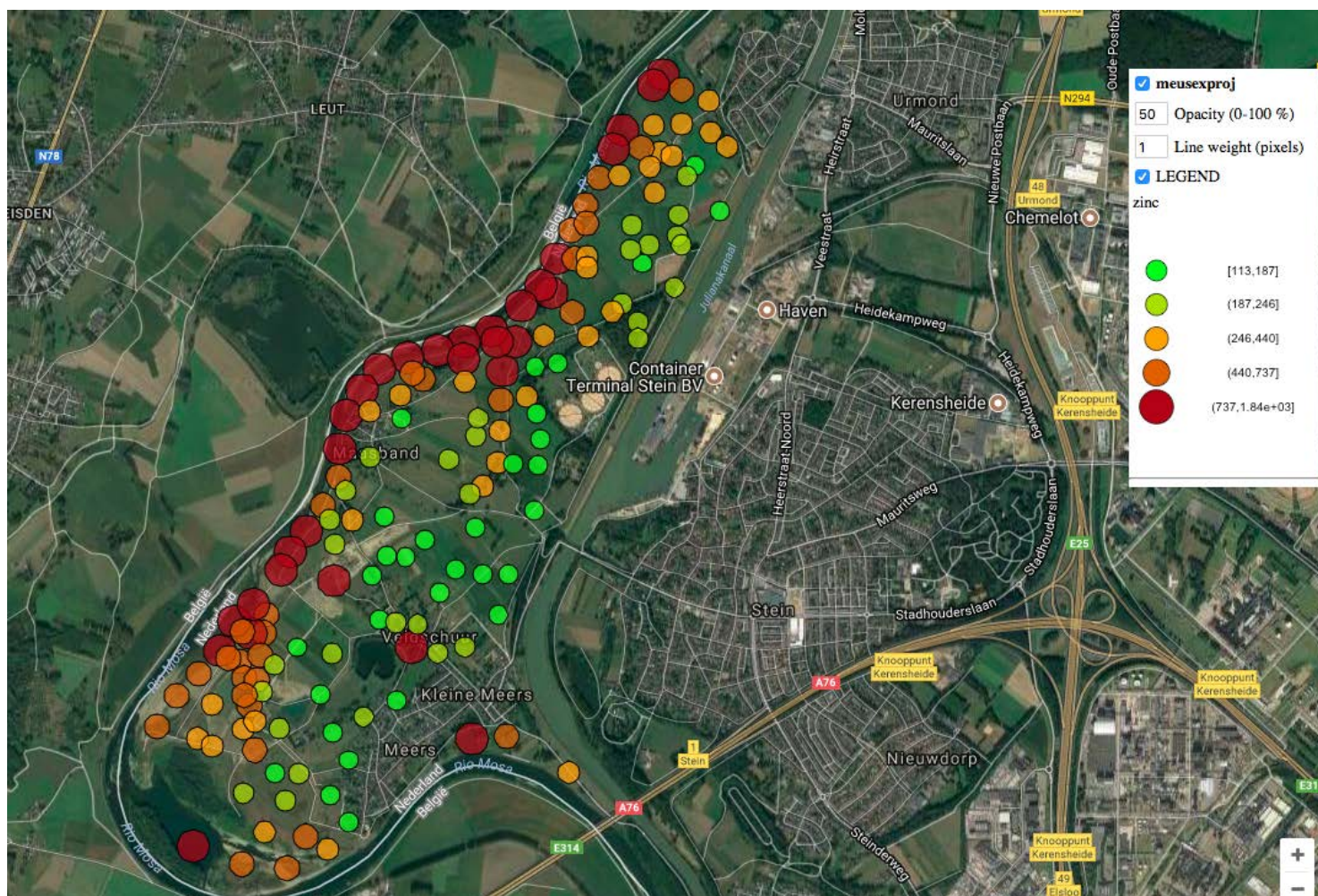


```
# Mapa 2  
meuse_proj = meuse  
proj4string(meuse_proj) <- CRS("+init=epsg:28992")  
mapa2 <- plotGoogleMaps(meuse_proj, filename='zinco_mapa2.htm', zcol = "zinc", colPalette  
= brewer.pal(5, "Reds"))
```

Mapa 3

```
mapa3 <- bubbleGoogleMaps(meuse_proj, zcol='zinc', max.radius = 80, filename='zinco_mapa3.htm')
```



6 - Georreferenciando dados utilizando o R

```
#carregando pacote
library(ggmap)

#Georreferenciando um endereço
geocode("Universidade Federal Fluminense")

geocode("Universidade Federal Fluminense", output = "latlona")

geocode("Universidade Federal Fluminense", output = "more")

mapImageData1 <- get_map(location = c(lon = -43.11672, lat = -22.90303),
  color = "color",
  source = "google",
  maptype = "satellite",
  zoom = 17)

ggmap(mapImageData1,
  extent = "device",
  ylab = "Latitude",
  xlab = "Longitude")
```



Atividade

Procure a localização da sua residência ou de um ponto turístico famoso e plot o mapa.