

R Intro GIS

Marcio Nicolau

2016-07-06

Contents

1	Prerequisites	5
2	Introdução	7
2.1	Estatística Básica	7
2.2	Gráficos	7
2.3	GIS	7
3	Modelos	11
3.1	Lineares	11
3.2	Generalizações	11
4	Métodos de Análise Multivariada	13
4.1	Agrupamentos	13
4.2	Análise Componentes Principais	13
4.3	Escalonamento Multidimensional	13
5	Interatividade	15
5.1	Histórico	15
5.2	Shiny	15

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

For now, you have to install the development versions of **bookdown** from Github:

```
devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading #.

To compile this example to PDF, you need to install XeLaTeX.

Chapter 2

Introdução

2.1 Estatística Básica

2.1.1 Medidas de Posição

2.1.2 Medidas de Dispersão

2.1.3 Tipos de Distribuição

2.1.4 Análise Exploratória

2.2 Gráficos

2.2.1 Base

2.2.2 GGplot2

2.3 GIS

O conjunto mínimo de pacotes para trabalhar com GIS (Geographical Information System) no R é exibido a seguir

```
libs <- c("ggmap", "rgdal", "rgeos",  
         "maptools", "dplyr", "tidyr",  
         "raster", "rasterVis",  
         "tmap") # declare the list of packages to load on environment  
supply(libs, require, character.only = TRUE) # load the required packages
```

```
## Loading required package: tmap
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,  
## logical.return = TRUE, : there is no package called 'tmap'
```

```
##      ggmap      rgdal      rgeos  maptools      dplyr      tidyr      raster  
##      TRUE       TRUE       TRUE      TRUE      TRUE      TRUE      TRUE  
## rasterVis      tmap  
##      TRUE      FALSE
```

Após instalar e carregar os pacotes que serão usados, é hora de realizar o download do arquivo que será trabalhado durante os exemplos.

Os comandos para realizar o download e descompactar o arquivo no diretório atual são executados dentro da sessão do R.

```
download.file('https://github.com/Robinlovelace/Creating-maps-in-R/zipball/master/',
             destfile = 'file.zip',
             mode = 'wb')

unzip(zipfile = 'file.zip', exdir = 'GIS-Data')
file.copy('GIS-Data/Robinlovelace-Creating-maps-in-R-930376b/data/', to = 'GIS-Data', recursive = TRUE)

## [1] TRUE
```

2.3.1 Bibliotecas para GIS

- ggmap
- rgeos
- rgdal
- maptools
- tmap

2.3.2 Bibliotecas para Raster

Ao se trabalhar com arquivos raster no R será utilizado principalmente duas bibliotecas

- raster
- rasterVis

O R armazena o arquivo raster de tres formas a saber:

- RasterLayer: menor porção de informação de um arquivo do tipo raster
- RasterStack: conjunto de layers (RasterLayer) que são armazenados e processados a partir dos arquivos originias (em disco)
- RasterBrick: conjunto de layers (RasterBrick) combinado para otimizar o trabalho em memória, todas as informações são armazenadas internamente no R.

Para iniciar os trabalhos com este tipo de arquivo e realizar algumas operações básicas, iremos trabalhar com dois arquivos raster sobre Evapotranspiração (2013 e 2014) gentilmente cedidos (?) para os trabalhos durante o curso.

2.3.2.1 Evapotranspiração

Para carregar os arquivos que estão na pasta 'dataClass' utilizados o seguinte código

```
listaArquivos <- list.files('dataClass/dados_curso_R_Evapot',
                           pattern = '*.tif$',
                           full.names = TRUE)

crsW84 = CRS("+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0")

evapo = stack(listaArquivos)
projection(evapo) = crsW84
```


Para saber mais sobre os códigos de projeções cartográficas acesse <http://spatialreference.org/>.

Podemos verificar o resumo do arquivo (sumário) com o comando

```
evapo13
```

```
## class      : RasterLayer
## dimensions : 5000, 5000, 2.5e+07 (nrow, ncol, ncell)
## resolution : 1, 1 (x, y)
## extent     : -0.5, 4999.5, -4999.5, 0.5 (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
## data source : /Users/marcio/Analysis/Cursos/r-intro-gis/dataClass/dados_curso_R_Evapot/ET_26062013_jaiba
## names      : ET_26062013_jaiba
## values     : 0, 6.1148 (min, max)
```

Podemos realizar algumas operações básicas como os dados das células (pixel) do arquivo raster, veja alguns exemplos

```
m1314=mean(evapo13, evapo14)
cellStats(evapo13, mean)
cellStats(evapo14, cv)
cellStats(evapo14, 'rms') # Root Mean Square
cellStats(evapo13, 'skew') # Skewness
# v <- sampleRegular(evapo13, size=ncell(evapo13)*.25, asRaster=TRUE)
# sampleRandom(r, size=10)
freq(evapo13)
crosstab(evapo, long = TRUE)
zonal(evapo13, evapo14, 'mean') # character function is better for large files
# In above example the mean was calculated for each code zone from evapo14
## see too extract function
# Example below look for data around a pixel
# rook <- matrix(c(NA, 1, NA,
#                  1, 0, 1,
#                  NA, 1, NA), ncol=3, byrow=TRUE)
# a0 = adjacent(evapo13, 1:ncell(evapo13), direction=rook, pairs = TRUE)
# tb = table(evapo13[a0[,1]], evapo13[a0[,2]])
# tb
# tb = unclass(tb)
# plot(raster(tb))
```

Também é possível fazer outras operação no raster, como por exemplo, recortar uma região de interesse com o método *drawExtent* ou com o uso de um arquivo shapefile.

```
plot(evapo13)
e <- drawExtent()
plot(evapo14)
mean(values(crop(evapo14, drawExtent())))
```

Para agrupar Layers podemos usar as funções *merge* e *mosaic*.

```
meEvapo = merge(evapo13, evapo14)
moEvapo = mosaic(evapo13, evapo14, fun=min)
```

Para calcular a autocorrelação espacial para uma layer raster, podemos usar *Geary*, *Moran*, *MoranLocal*, *GearyLocal*.

```
Moran(evapo13)
Moran(evapo14)
```

Também é possível realizar outras operações com cálculos, a partir de funções definidas pelo usuário. Pode-se calcular o índice NDVI de um raster Landsat 7 pela combinação das bandas 4 e 3.

$$NDVI = \frac{NIR - R}{NIR + R}$$

```
# Define the function to calculate NDVI from
ndvCalc <- function(x) {
  ndvi <- (x[[4]] - x[[3]]) / (x[[4]] + x[[3]])
  return(ndvi)
}
# ndvi2 <- calc(x=gewata, fun=ndvCalc)
ndvOver <- function(x, y) {
  ndvi <- (y - x) / (x + y)
  return(ndvi)
}
# ndvi3 <- overlay(x=gewata[[3]], y=gewata[[4]], fun=ndvOver)
```

Conforme o interesse é possível reprojeter o raster para outras projeções cartográficas.

```
ndviLL <- projectRaster(evapo13, crs='+proj=longlat')
```

Também é possível utilizar modelos para fazer previsões em dados de raster. Para tanto, utilizamos a função *predict*. Se o modelo faz uso da posição (x e y) como variável é necessário utilizar a função *interpolate*.

2.3.3 Shapefile

Chapter 3

Modelos

Here is a review of existing methods.

3.1 Lineares

3.2 Geralizações

3.2.1 GLM

3.2.2 SEM

Chapter 4

Métodos de Análise Multivariada

We describe our methods in this chapter.

4.1 Agrupamentos

4.1.1 Métodos supervisionados

4.1.2 Métodos não-supervisionados

4.2 Análise Componentes Principais

4.3 Escalonamento Multidimensional

Chapter 5

Interatividade

Some *significant* applications are demonstrated in this chapter.

5.1 Histórico

5.2 Shiny

5.2.1 Exemplos para GIS

Bibliography