Advanced Geospatial Data Analysis in R: Environmental Application

Marj Tonini, Haokun Liu

2023-11-21

Contents

Chapter 1

About

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

1.1 Usage

Each **bookdown** chapter is an .Rmd file, and each .Rmd file can contain one (and only one) chapter. A chapter *must* start with a first-level heading: `# A good chapter`, and can contain one (and only one) first-level heading.

Use second-level and higher headings within chapters like: `## A short section` or `### An even shorter section`.

The `index.Rmd` file is required, and is also your first book chapter. It will be the homepage when you render the book.

1.2 Render book

You can render the HTML version of this example book without changing anything:

1. Find the **Build** pane in the RStudio IDE, and
2. Click on **Build Book**, then select your output format, or select "All formats" if you'd like to use multiple formats from the same book source files.

Or build the book from the R console:

```
bookdown::render_book()
```

To render this example to PDF as a `bookdown::pdf_book`, you'll need to install XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

1.3 Preview book

As you work, you may start a local server to live preview this HTML book. This preview will update as you edit the book when you save individual .Rmd files. You can start the server in a work session by using the RStudio add-in "Preview book", or from the R console:

```
bookdown::serve_book()
```

Chapter 2

Introduction to R

2.1 R Language

R is a complete programming language and software environment for statistical computing and graphical representation. As part of the GNU Project (free software, mass collaboration project), the source code is free available. Its functionalists can be expanded by importing packages. For more details on R see https://www.r-project.org/.

2.1.1 R Packages

A package is a file generally composed of R scripts (e.g., functions). On all operation systems the function "install.packages()" can be used to download and install a package automatically. Once a package has been installed, it can be loaded in a session by using the command library(package). To check the list of the installed libraries, the function library() can be used. When you open an **R Markdown** document (.Rmd) the program propose you automatically to install the libraries listed there.

2.1.2 Some tips

- R is case sensitive!
- Previously used command can be recalled in the console by using the *up arrow* on the keyboard.
- The working directory by default is "*C:/user/.../Documents*".
  - It can be found using the command getwd()
  - It can be changed using the command line setwd("C:/Your/own/path")

- In **R Markdown**: the working directory when evaluating R code chunks is the directory of the input document by default.
  - To access to a specific file in a sub-folder use ". /subfolder/file.ext"
  - To access to a specific file in a up-folder use ". . /upfolder/file.ext"

2.1.3 R Commands (online resources)

Many table resuming the main R commands can be found online. Here some useful links:

- A short list of the most useful R commands
- Table of Useful R commands
- Basic Commands to Get Started with R

2.2 R Markdown

This is an R Markdown document :-)

Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. It is a simple and easy to use **plain text language** used to combine R code, results from your data analysis (including plots and tables), and written commentary into a single nicely formatted and reproducible document (like a report, publication, thesis chapter or a web pages).

Code lines are organized as code block, seeking to solve e specified task, and referred to as **"code chunk"**. For more details on using R Markdown see http://rmarkdown.rstudio.com.

All what you have to do during the computing labs is to read each explanatory paragraph before running each individual R code chunk, one by one, and to interpret the results. Finally, to create a personal document (usually PDF) from rmarkdown, you need to **Knit** the document. Knitting a document simply means taking all the text and code and creating a nicely formatted document.

2.3 Data type in computational analysis

2.3.1 Variables

Variables are used to store values in a computer program. Values can be numbers (real and complex), words (string), matrices, and even tables.

The fundamental or atomic data in R Programming can be:

- **integer**: number without decimals
- **numeric**: number with decimals (float or double depending on the precision)
- **character**: string, label
- **factors**: a label with a limited number of categories
- **logical**: true/false

Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”

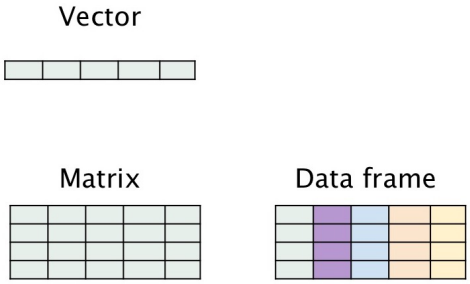


Figure 2.1: Data Types in R

2.3.2 Data structure in R

R's base data structures can be organised by their dimensionality (1d, 2d, or nd) and whether they are homogeneous (all contents must be of the same type) or heterogeneous (the contents can be of different types).

This gives rise to the four data structures most often used in data analysis:

Dimensions	Homogeneous	Heterogeneous
1d	Atomic vector	List
2d	Matrix	Data frame

Figure 2.2: Data structures in R

A **Vector** is a one-dimensional structure winch can contain object of one type only: numerical (integer and double), character, and logical.

```
# Investigate vector's types:

v1 <- c(0.5, 0.7); v1; typeof(v1)
#> [1] 0.5 0.7
#> [1] "double"

v2 <-c(1:10); v2; typeof(v2)
#>  [1]  1  2  3  4  5  6  7  8  9 10
#> [1] "integer"

v3 <- c(TRUE, FALSE); v3; typeof(v3)
#> [1]  TRUE FALSE
#> [1] "logical"

v4 <- c("Swiss", "Itay", "France", "Germany"); v4; typeof(v4)
#> [1] "Swiss"   "Itay"    "France"  "Germany"
#> [1] "character"
```

```
#Create a sequence from 0 to 5 with a step of 0.5:

v5 <- seq(1, 5, by=0.5); v5; typeof(v5)
#> [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
#> [1] "double"

length(v5)
#> [1] 9

summary(v5)
```

```
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>       1       2       3       3       4       5
```

```
#Extract the third element of the vector
v5[3]
#> [1] 2

#Exclude the third element from the vector and save as new vector
v5[-3]
#> [1] 1.0 1.5 2.5 3.0 3.5 4.0 4.5 5.0
w5<-v5[-3]; w5
#> [1] 1.0 1.5 2.5 3.0 3.5 4.0 4.5 5.0
```

A **Matrix** is a two-dimensional structure winch can contain object of one type only. The function matrix() can be used to construct matrices with specific dimensions.

```
# Matrix of elements equal to "zero" and dimension 2x5
m1<-matrix(0,2,5); m1  #(two rows by five columns)
#>      [,1] [,2] [,3] [,4] [,5]
#> [1,]    0    0    0    0    0
#> [2,]    0    0    0    0    0

# Matrix of integer elements (1 to 12, 3x4)
m2<-matrix(1:12, 3,4); m2
#>      [,1] [,2] [,3] [,4]
#> [1,]    1    4    7   10
#> [2,]    2    5    8   11
#> [3,]    3    6    9   12

# Extract the second row
m2[2, ]
#> [1]  2  5  8 11
# Extract the third column
m2[,3]
#> [1] 7 8 9
# Extract the the second element of the third column
m2[2,3]
#> [1] 8
```

2.3.3 Data Frame

A **data frame** allows to collect data of different type. All elements must have the same length.

A **list** is a more flexible structure since it can contain variables of different types and lengths. Nevertheless, the preferred structure for statistical analyses and computation is the data frame.

It is a good practice to explore the data frame before performing further computation on the data. This can be simply accomplished by using the commands str to explore the structure of the data and summary to display the summary statistics and quickly summarize the data. For numerical vectors the command hist() can be used to plot the basic histogram of the given values.

```
# Create the vectors with the variables

cities <- c("Berlin", "New York", "Paris", "Tokyo")
area <- c(892, 1214, 105, 2188)
population <- c(3.4, 8.1, 2.1, 12.9)
continent <- c("Europe", "Norh America", "Europe", "Asia")
```

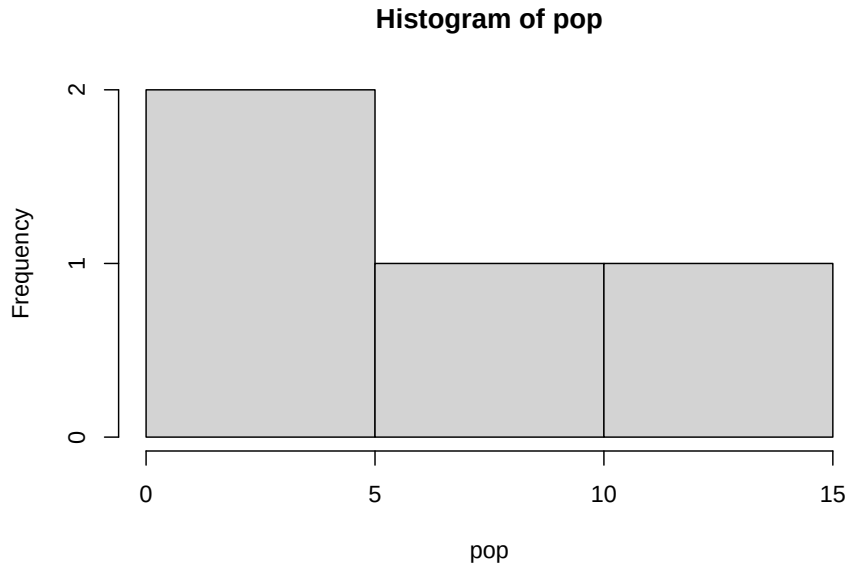
```
# Concatenate the vectors into a new data frame
df1 <- data.frame(cities, area, population, continent)
df1
#>     cities area population    continent
#> 1   Berlin  892        3.4       Europe
#> 2 New York 1214        8.1 Norh America
#> 3    Paris  105        2.1       Europe
#> 4    Tokyo 2188       12.9         Asia

#Add a column (e.g., language spoken) using the command "cbind"
df2 <- cbind (df1, "Language" = c ("German", "English", "Freanch", "Japanese"))
df2
#>     cities area population    continent Language
#> 1   Berlin  892        3.4       Europe   German
#> 2 New York 1214        8.1 Norh America  English
#> 3    Paris  105        2.1       Europe  Freanch
#> 4    Tokyo 2188       12.9         Asia Japanese
```

```
#Explore the data frame
str(df2) # see the structure
#> 'data.frame':    4 obs. of  5 variables:
#>  $ cities    : chr  "Berlin" "New York" "Paris" "Tokyo"
#>  $ area      : num  892 1214 105 2188
#>  $ population: num  3.4 8.1 2.1 12.9
#>  $ continent : chr  "Europe" "Norh America" "Europe" "Asia"
#>  $ Language  : chr  "German" "English" "Freanch" "Japanese"
summary(df2) # compute basic statistics
#>     cities              area          population
#>  Length:4           Min.   : 105.0   Min.   : 2.100
```

```
#>  Class :character   1st Qu.: 695.2   1st Qu.: 3.075
#>  Mode  :character   Median :1053.0   Median : 5.750
#>                     Mean   :1099.8   Mean   : 6.625
#>                     3rd Qu.:1457.5   3rd Qu.: 9.300
#>                     Max.   :2188.0   Max.   :12.900
#>   continent           Language
#>  Length:4           Length:4
#>  Class :character   Class :character
#>  Mode  :character   Mode  :character
#>
#>
#>

# Use the symbol "$" to address a particular column
pop<-(df2$population)
pop
#> [1]  3.4  8.1  2.1 12.9
hist(pop) # plot the histogram
```



Chapter 3

Geographically Weighed Summary Statistics

3.1 Introduction

In fire management, it is crucial to investigate where fires occurred more frequently and to distinguish between **small** and **large fires**. This is key information to understand the ignition factors and planning strategies to reduce forest fires, control and manage ignition sources, and identify areas at risk.

Despite the availability of forest fires spatio-temporal inventories, it is not evident to extract information about their pattern distribution simply by looking at the original arrangement of the mapped burnt areas. To this end, **Geographically Weighed Summary Statistics (GWSS)** can be computed, under the assumption that burned areas generally follow a geographic trend.

We compute here the GW local means, the GW local standard deviation and the GW localized skewness of burned areas in continental Portugal, registered in the period 1990-2013. This application is inspired by the work of (**?**)

3.2 The overall methodology

Summary statistics include a number of measures that can be used to summarize a set of observations, the most important of which are measures of *central tendency* (arithmetic mean, median and mode) and measures of *dispersion around the mean* (variance and standard deviation). In addition, measures of *skewness* and *kurtosis* are descriptors of the shape of the probability distribution function, the former indicating the asymmetry and the latter the peakedness/tailedness of the curve.

In the case of **spatial data**, these global statistical descriptors may vary from one region to another, as their values may be affected by local environmental and socio-economic factors. In this case, an appropriately localized calibration can provide a better description of the observed values. One way to achieve this goal is to *weight* the above statistical measures for a given quantitative variable based on their geographical location.

We introduce here the method proposed by (**?**) and implemented in the **function GWSS** presented in the R package GWmodel (**??**). The evaluation of geographically weighted summary statistics is obtained by computing a summary for a small area around each geolocalized punctual observation, by using the *kernel density estimation* technique (KDE) (**?**). KDE is estimated at each point, taking into account the influence of the points falling within an area, with increasing weight towards the center, corresponding to the point location. A surface summary statistic is thus obtained.

3.3 Forest fires dataset

Forest fires inventories indicating the location, the starting date and other related variables, such as the cause of ignition and the size of the area burned, are broadly available with a different degree of accuracy in different countries.

In the present study, we consider the **Portuguese National Mapping Burnt Areas (NMBA 2016)**, freely available from the website of the Institute for the Conservation of Nature and Forests (ICNF). This is a long spatio-temporal dataset (from 1975) resulting from the processing of satellite images acquired once a year at the end of the summer season. Row data consist of records of observed fire scars. The burned areas were estimated by using image classification techniques, then compared with ground data to resolve the discrepancies. Polygons have been converted into point shapefile, where each point represent the centroid of the burned areas, while the size of the burned areas and the starting date of the fires events are given as attributes. In this work, for consistency reasons, we consider only fires occurred between 1990 and 2013 and with a burned area above 5 hectares. Figure 3.1.

3.3.1 Load the libraries

First you have to load the following libraries:

- **splancs**: for display and analysis of spatial point pattern data
- **GWmodel**: techniques from a particular branch of spatial statistics, termed geographically-weighted (GW) models
- **sf**: support for simple features, a standardized way to encode spatial vector data

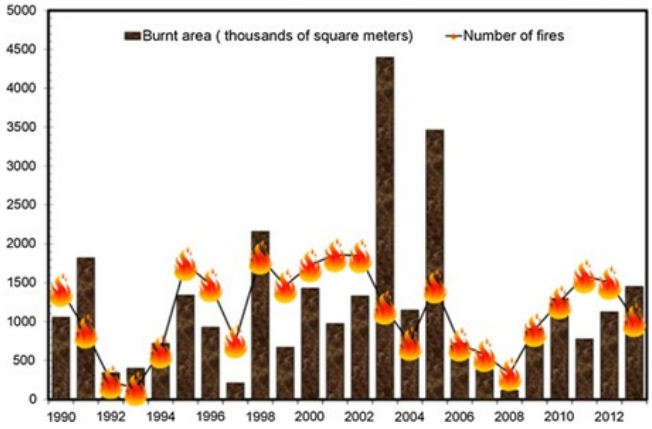


Figure 3.1: Total annual number of forest fire events, expressed in thousands of square metres

- **ggplot2**: a system for 'declaratively' creating graphics
- **sp**: classes and methods for spatial data

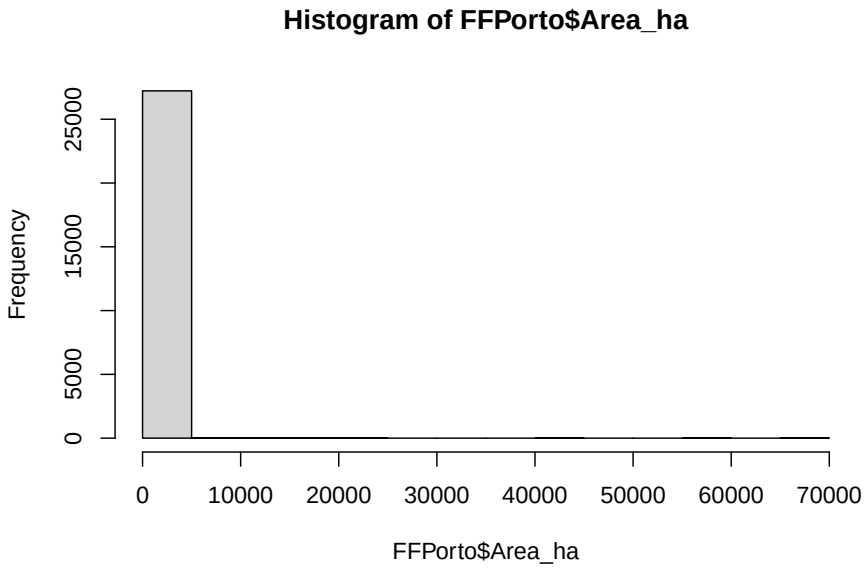
```
#>  [1] "ggplot2"    "sf"         "GWmodel"    "Rcpp"
#>  [5] "robustbase" "splancs"    "sp"         "stats"
#>  [9] "graphics"   "grDevices"  "utils"      "datasets"
#> [13] "methods"    "base"
```

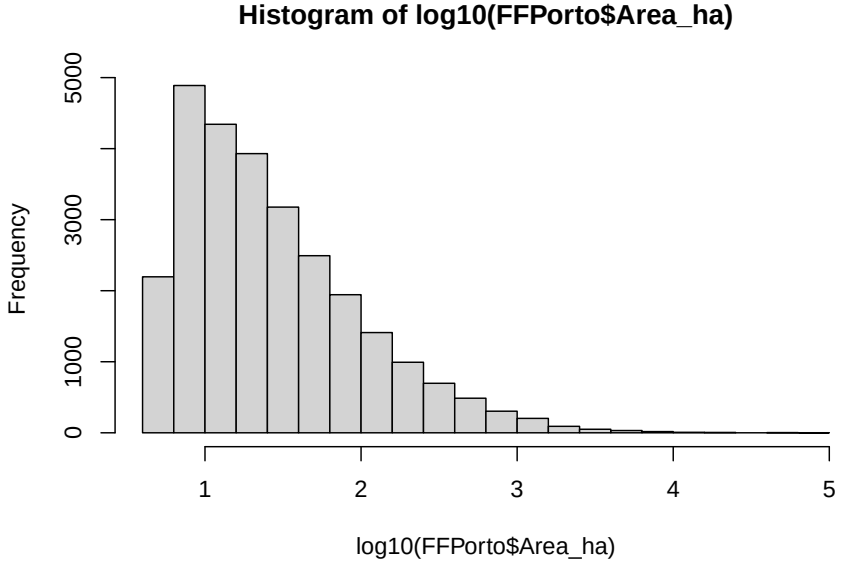
3.3.2 Import the Portuguese forest fire dataset

In this section you will load the geodata representing the dataset of forest fires occurred in the continental Portuguese area in the period 1990-2013. You will also load the boundaries of the study area. You will start by exploring the datasets using mainly visual tools (plotting and histogram).

You can explore the dataset by using different tools for **exploratory data analyses**. You will start by visualizing the databases. In the GIS environment, this correspond to the attribute table of a vector punctual feature.

Than you can **plot the histogram** of events distribution based on the variable *"Area_ha"* (the size in hectares of the burned area). Since this is a power low distribution, for a better understanding it's recommended to transform the data using a logarithmic scale. Using Log10 you can easily evaluate the frequency distribution of the burned areas.


Histogram of FFPorto$Area_ha
Frequency
0
5000
15000
25000
0
10000
20000
30000
40000
50000
60000
70000
FFPorto$Area_ha

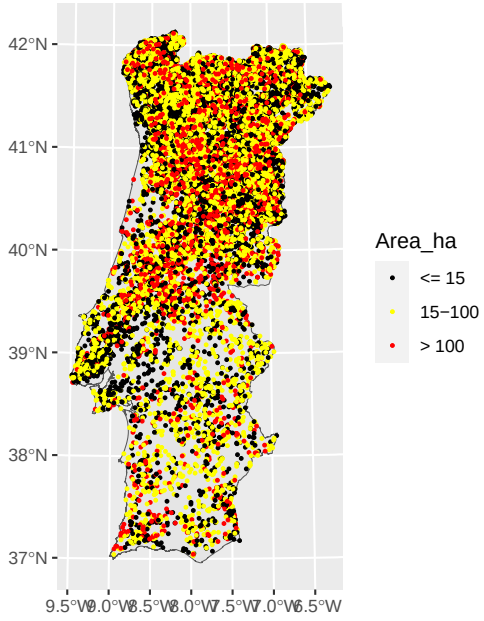

Histogram of log10(FFPorto$Area_ha)
Frequency
0
1000
3000
5000
1
2
3
4
5
log10(FFPorto$Area_ha)

3.3.3 Forest fires spatial distribution

For a better understanding of the phenomenon, you can group the events according to the size of the burned area. Based on the frequency distribution of the burned areas, the following three classes can be defined:

- **Small fires**: less than 15 ha
- **Medium fires**: between 15 ha and 100 ha
- **Large fires**: bigger than 100 ha

Plotting the forest fires events using different colors, based on the size of the burned areas, can simplify the understanding of their **pattern distribution**, knowing that fires of different size have normally different drivers.



3.4 Compute the geographically whited statistics

From the exploratory data analysis performed above, it seems that a simple plotting of the forest fires events based on their spatial distribution, even if classified based on their size, can not really help to understand their behaviors.

This is because we face to a huge number of events and the variable that we are using to characterize them (i.e., the size of the burned area) is very heterogeneous. To this aim, we can compute basic and robust **GWSS** and plot the data accordingly.

GWSS includes *geographically weighted means*, *standard deviations* and the *skweness*. As you can see from the R Documentation (command: help(gwss), same data manipulations are necessary to transform the forest fires dataset in a compatible data frame format.

GWSS parameters:

- We summarize the data based on the size of the burned area (*vars*).
- We use here an adaptive kernel where the bandwidth (*bw*) corresponds to the number (100 in this case) of nearest neighbors (i.e. adaptive distance).
- We keep the default values for the other parameters.

3.4.1 Look at the results

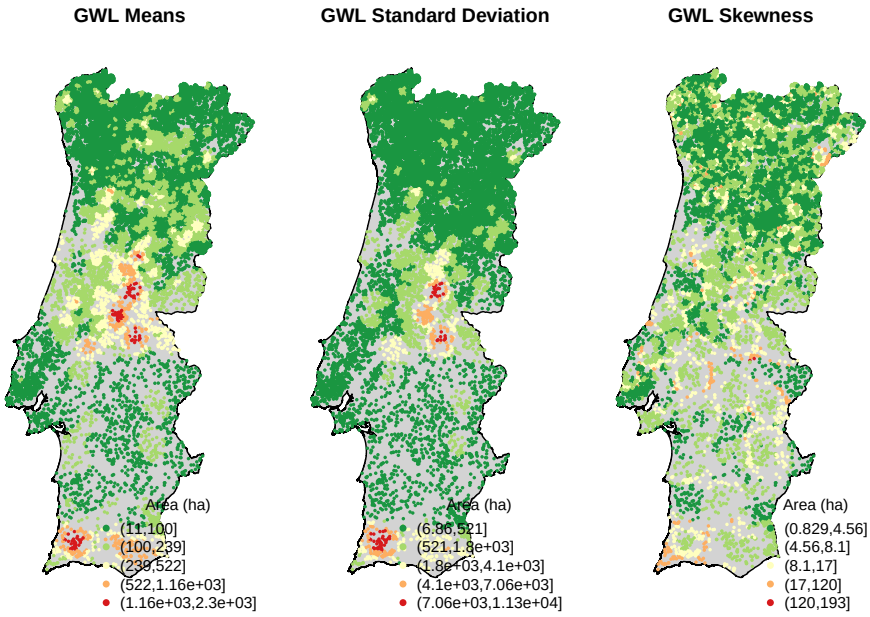
The resulting object (**FFgwss**) has a number of components. The most important one is the spatial data frame containing the results of local summary statistics for each data point location, stored in *FFgwss$SDF* (that is a spatial DataFrame).

3.4.2 GWSS maps

To produce a map of the local geographically weighted summary statistic of your choice, firstly we need to enter a small R function definition. This is just a short R program to draw a map: you can think of it as a command that tells R how to draw a map (see [Geographically Weighted Summary Statistics] (https://rpubs.com/chrisbrunsdon/99667) for more details). The advantage of defining a function is that the entire map can now be drawn using a single command for each variable, rather than having to repeat those steps each time. To define the intervals for the classification we used Jenks natural breaks classification method (textcolor{red}{style="fisher"}).

Finally the function is called by entering:

```
quick.map(gwss.object,variable.name,legend.title,main.map.title)
```



3.5 Conclusions and further analyses

This practical computer lab allowed you to familiarize with GWSS, by the proposed application about geographically weighted summary statistics. This method allowed us to explore how the average burned area vary locally through Continental Portugal in the period 1990-2013.

The global Geographically Weighted (GW) means informs you about the local average value of the burned area, based of the neighboring events occurred in a given period. Similarly, you may compute a GW standard deviation to see the extent to which the size of the burned area spread around this mean. Finally you can compute the GW skewness to measure the symmetry of distribution: a positively skewed distribution means that there is a higher number of data points having low values, with mean value lower that the median; and the contrary for a negatively skewed distribution.

To be sure that everything is perfectly clear for you, we propose you to **answer the following questions** and to discuss your answers with the other participants to the course or directly with the teacher.

1) What is the pattern distribution of the GW-means for burned area in Portugal during the investigated periods?

2) Does the GW-standard deviation follows the same pattern? How can you

interpret the two pattern in terms of burned area and their characterization?

3) GW-skewness has positive values everywhere: what does it means? What do these values suggest to be the distribution of the burned areas, in terms of their size, around the local means?

4) Which can be other applications of GWSS for geo-environmental data? In other words, can you imagine other geo-environmental dataset that can be analysed using GWSS?

5) You can finally play with the code and try to run it using a different numbers of nearest neighbors (bw=x) and comparing the results.

   NB: You have to rename the original pdf to avoid overwriting it. In addition, if a pdf with the same name saved in the same destination folder is open, you will receive an error message, so close it before Knitting.

Chapter 4

Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART\*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 5

Footnotes and citations

5.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

5.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package (**?**) (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** (**?**) (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The `bs4_book` theme makes footnotes appear inline when you click on them. In this example book, we added `csl: chicago-fullnote-bibliography.csl` to the `index.Rmd` YAML, and include the `.csl` file. To download a new style, we recommend: https://www.zotero.org/styles/

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

Chapter 6

Blocks

6.1 Equations

Here is an equation.

$$f\left(k\right) = \binom{n}{k} p^k \left(1-p\right)^{n-k} \tag{6.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (6.1).

6.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 6.1.

Theorem 6.1. *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the* ***other*** *two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

6.3 Callout blocks

The `bs4_book` theme also includes special callout blocks, like this `.rmdnote`.

You can use **markdown** inside a block.

```
head(beaver1, n = 5)
#>   day time  temp activ
#> 1 346  840 36.33     0
#> 2 346  850 36.34     0
#> 3 346  900 36.35     0
#> 4 346  910 36.42     0
#> 5 346  920 36.55     0
```

It is up to the user to define the appearance of these blocks for LaTeX output.

You may also use: `.rmdcaution`, `.rmdimportant`, `.rmdtip`, or `.rmdwarning` as the block name.

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Chapter 7

Sharing your book

7.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

7.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

7.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `bs4_book` provides enhanced metadata for social sharing, so that each chapter shared will have a unique description, auto-generated based on the content.

Specify your book's source repository on GitHub as the `repo` in the `_output.yml` file, which allows users to view each chapter's source file or suggest an edit. Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/bs4_book.html

Or use:

```
?bookdown::bs4_book
```

Bibliography

Brunsdon, C. Estimating probability surfaces for geographical point data: An adaptive kernel algorithm. 21(7):877–894.

Brunsdon, C. RPubs - GWSS - (7th channel network conference).

Brunsdon, C., Fotheringham, A., and Charlton, M. Geographically weighted summary statistics - a framework for localised exploratory data analysis. 26(6):501–524.

Lu, B., Harris, P., Charlton, M., and Brunsdon, C. The GWmodel r package: further topics for exploring spatial heterogeneity using geographically weighted models. 17(2):85–101. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10095020.2014.917453.

Tonini, M., Pereira, M. G., Parente, J., and Vega Orozco, C. Evolution of forest fires in portugal: from spatio-temporal point events to smoothed density maps. 85(3):1489–1510.

Xie, Y. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.