UNIVERSITÀ CATTOLICA SACRO CUORE

STATISTICAL AND ACTUARIAL SCIENCES

MJ: DATA BUSINESS ANALYTICS

REST API for Real Estate rental data, a spatial bayesian modeling approach with INLA

Author:
Niccolò SALVINI

Supervisor:
Dr. Marco DELLAVEDOVA

Assistant Supervisor:
Dr. Vincenzo NARDELLI

AY 2019 / 2020

REST API for Real Estate rental data, a spatial bayesian modeling approach with INLA.

Niccolò Salvini[1]

date: Last compiled on 28 ottobre, 2020

[1]https://niccolosalvini.netlify.app/

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Main themes:

- General introduction to the problem to solve
- Open Data discussion
- Research Question
- Milan Real Estate Controversies
-
- Why a Bayesian approach

Chapter 2

Scraping

2.1 What is Web Scraping

Web Scraping is a techniques that involves informatics and statistics aimed at extracting data from static or dynamic internet web pages. It can be done automatically and simultaneously as well as by a scheduler that palns its execution at a given time. Due to the absence of API's to call regarding Milan Real Estate rental markets, this practice has to be forcly applied. The reason behind that and personal hope is to open source the scraping API so that in the future further analysis will be available without putting effort in gathering data fresh data. *Content distribution is managed by HTTP protocols, which are a common standard to connect web clients. Web browsers (i.e. one among the others web clients) download the content and parses them making possible to read them_The athomical unit of measurement in scraping is the url, which are the locations where this data exchanges abstractly happens.* [questo in API infra] Data/content in webpages is the most of the times well organized and accessible by urls. This is made possible by the effort put into building both the *website structure* and the *content architecture*. For website structure it is meant the way urls, pointing to webpages, are arranged throughout the website. Website structure constitutes a *first dimension* of hierarchy. Some popular structures examples might regard social-networks where posts can be scrolled down within a single page named the wall. Scrolling down might end due to fewer posts, but the perception is a never-ending webpage associated to a single url. Instead personal profiles are dedicated to a specified unique url and even in profiles posts are allocated into a sub domain and might be scrolled down arranged by time since the day of social subscription. Online newspapers display their articles in the front wall and by accessing to one of them all the related following articles sometimes can be reached by an arrow, pointing right or left. Articles can also be suggested and the more the website is explored the more articles are likely to be seen twice within the same session. It will soon end up into a suggestion loop that it will recursively shows the same contents; recursive structures are popular in newspaper-type websites. Online Retailers as Amazon, based on filters, groups inside a single webpage (i.e. page n° 1) a fixed set of items, having their dedicated personal url attached to them. Furthermore Amazon offers the opportunity to skip to the following page (i.e. page n° 2), searching for another different and fixed set of items and so on until the last page/url. Generally website structures try to reflect both the user expectations with respect to the product on the website and the design expression of the web developer. For these reasons for each websites usage category exists a multitude of content architectures. For each content architectures there are multiple front end languages which are destinated to multiple end users. In the future expectations are tailor made webpages on users with respect to its personal preferences. Moreover web design in scraping plays an important role since the more sophisticated graphical technologies are implied, the harder will be to scrape information.

A *second dimension* of hierarchy is brought by content architecture in the name of the language used for content creation and organization i.e. HTML. HTML stands for Hyper Text Markup Language and … HTML drives the hierarchy structure that is then generalized to the website structure. According to this point of view the hierarchical website structure is a consequence of the content architecture by means of HTML language (*arborescence*: direction from root to leaves). CSS language stands for Cascading Style Sheets and takes care of

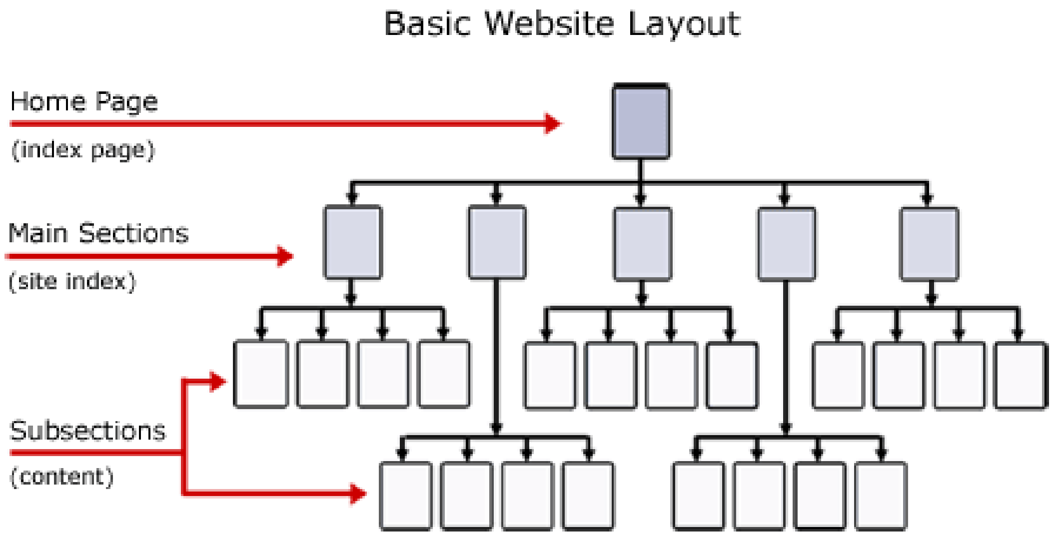


Figure 2.1: general website structure

the style of the webpage. The combination of HTML and CSS offers a wide flexibility in building web sites, once again expressed by the vast amount of different designs on the web. Some websites' components also might be tuned by Javascript language, which in the context of scraping adds a further layer of difficulty. As a matter of fact since Javascript components are dynamic within the webpage, scraping requires specialized libraries to enable different parser to get the content. CSS allows the scraper to target a class of objects in the web page that shares same style (e.g. same css query) so that each element that belongs to the class (i.e. share same style) can be gathered. This practice provides enormous advantages since by CSS a set of objects can be obtained within a single function call. First and Second dimension of the scraping problem imply hierarchy, a simple way to approach the problem is to represent it through already known data structures. One way to imagine hierarchy in both of the two dimensions are graph based data structures named as **Rooted Trees**. By analyzing the first dimension through the lenses of Rooted trees it is possible to compress the whole problem into the general graph based jargon (Diestel, 2006). Rooted trees must start with a root node which is the domain of the web page. Each *Node* is a url destination and each *Edge* is the connection between nodes. Connections have been made possible in the website by nesting urls inside webpages so that within a single webpage the user can access to a number of other related links. Furthermore, as an extension to the rooted tree framework a general graph theory component is introduced, i.e. the *Weight*. Each edge is associated to a weight whose interpretation is the run time cost to walk from a node to its conncted other nodes (e.g. from a url to the other). In addition the content inside each node takes the name of payload, which ultimately is the scope of the scraping processes. The walk from node 17 to node 8 in figure below (even though that is the case of a binary rooted tree) is called path and it represented as an ordered list of nodes connected by edges. In this context each node can have both a fixed and variable outgoing sub-nodes that are called *Children* . When root trees have a fixed set of children are called *k-ary* rooted trees. A node is said to be *Parent* to other nodes when it is connected to them by outgoing edge, in right figure below "head" is the parent of nodes "title" and "meta". Nodes in the tree that shares the same parent node are said *Siblings*, "head" and "body" are siblings in figure @ref(tree_html). Moreover *Subtrees* are a set of nodes and edges comprised of a parent and its descendants e.g. node "body" with all of its descendants might constitute a subtree. The concept of subtree in both of the dimensions plays crucial role in cutting run time scraping processes and fake headers provision (see section 2.3.1). If the website strucuture is locally reproducible and the content architecture within webpages tends to be equal, then functions for a single subtree might be extended to the rest of others subtrees that are siblings to the same parent node. Local reproducibility is a property according to which starting from a single url all the related urls can be inferred from a pattern. Equal content architecture throughout different single links means to have sort of standard shared-within advs criteria that each single rental advertisement has to refer. In addition two more metrics have might better describe the tree: *level* and *height*. The level of a node **L** counts the number of edges on the path from the root node to **L**. The height is the maximum level for any node in the tree, from now on **H**. What is worth to be anticipating is that functions are not going to be applied directly to siblings in the more general rooted tree. Instead it would be better segmenting the highest level rooted tree into a sequence of single subtrees whose roots are the siblings for reasons explained in section 2.3.1.

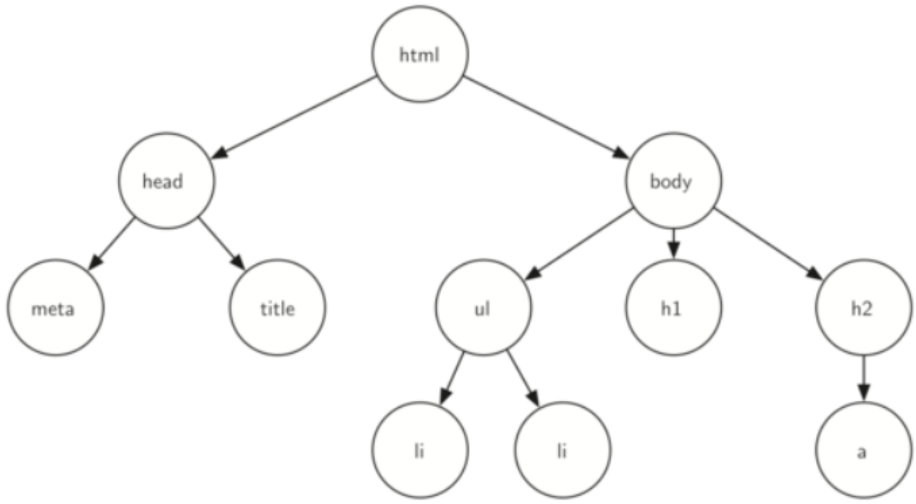


Figure 2.2: html_tree

2.1.1 Immobiliare.it Webscraping website structure

The structure of the website resembles the one encountered for the popular online retailer Amazon. According to the query filters (e.g. number of rooms 5, price less than 600 p.m. etc), the url is shaped so that each further filter added is appened at the end to the domain url `https://www.immobiliare.it/` root node (see figure below). Once filters are all applied to the root domain this constitutes the new url root domain node that might have this appereance: `https://www.immobiliare.it/affitto-case/milano/?criterio=rilevanza&superficieMi`
Sinced this is true only for page n°1 and contains the first 25 advs all the remaining siblings nodes corresponding to next pages have to be generated. Here comes handy the Local reproducibility property introduced in the previous section. The rest of the siblings, i.e. the ones belonging to page 2 (with the attached 25 advs) and to page 3 can be generated by appending `&pag=n` at the end of the url, where n is the page number reference (from now on referred as *pagination*). For page number 4 the

exact url that points to the web page, being equal the filters from above, is `https://www.immobiliare.it/affitto-case/milano/?criterio=rilevanza&superficieMi` Author customary choice is to stop pagination up to 300 pages since spatial data can not be to large due to computational burdens. Moreover thanks to the reverse engineer applied to url building tree height **H** is pruned, so the parsing part is cut short. At this point pagination has generated a list of siblings urls whose children number is fixed (i.e. 25 per page). That makes those trees k-ary, where k is 25 indicating the number of children. K-ary trees with equal content structure shared across siblings allow to design a single function to call that could be mapped to all the other siblings. In addition in order to further disassemble the website and making available the whole set of single rental advertisement links for each page (ranging from 1 to 300) a specific function has been made. As a consequence a single function call `scrape_href()` can grab all the links inside page 1. The function is then iterated along all the previously generated siblings (i.e. pages) obtaining a collection of all the single ads links belonging to the set of pages generated.

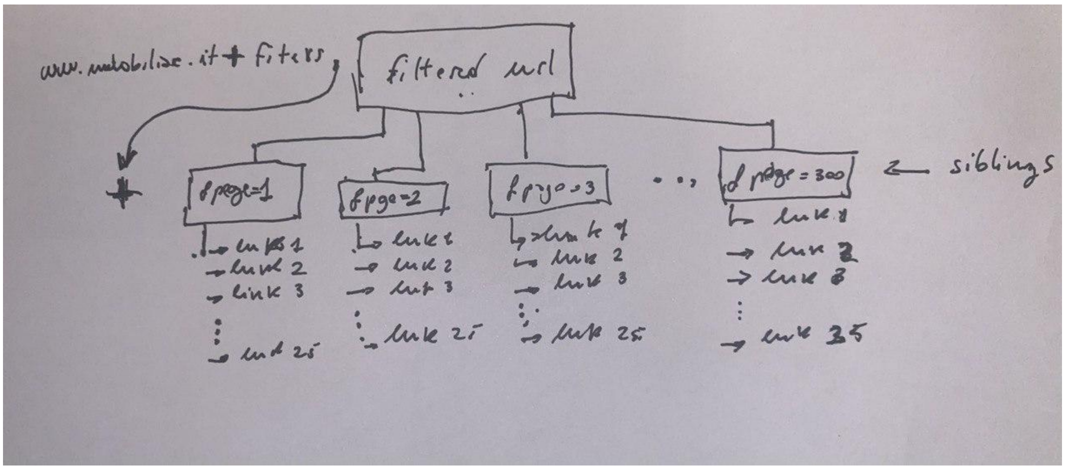


Figure 2.3: website_tree

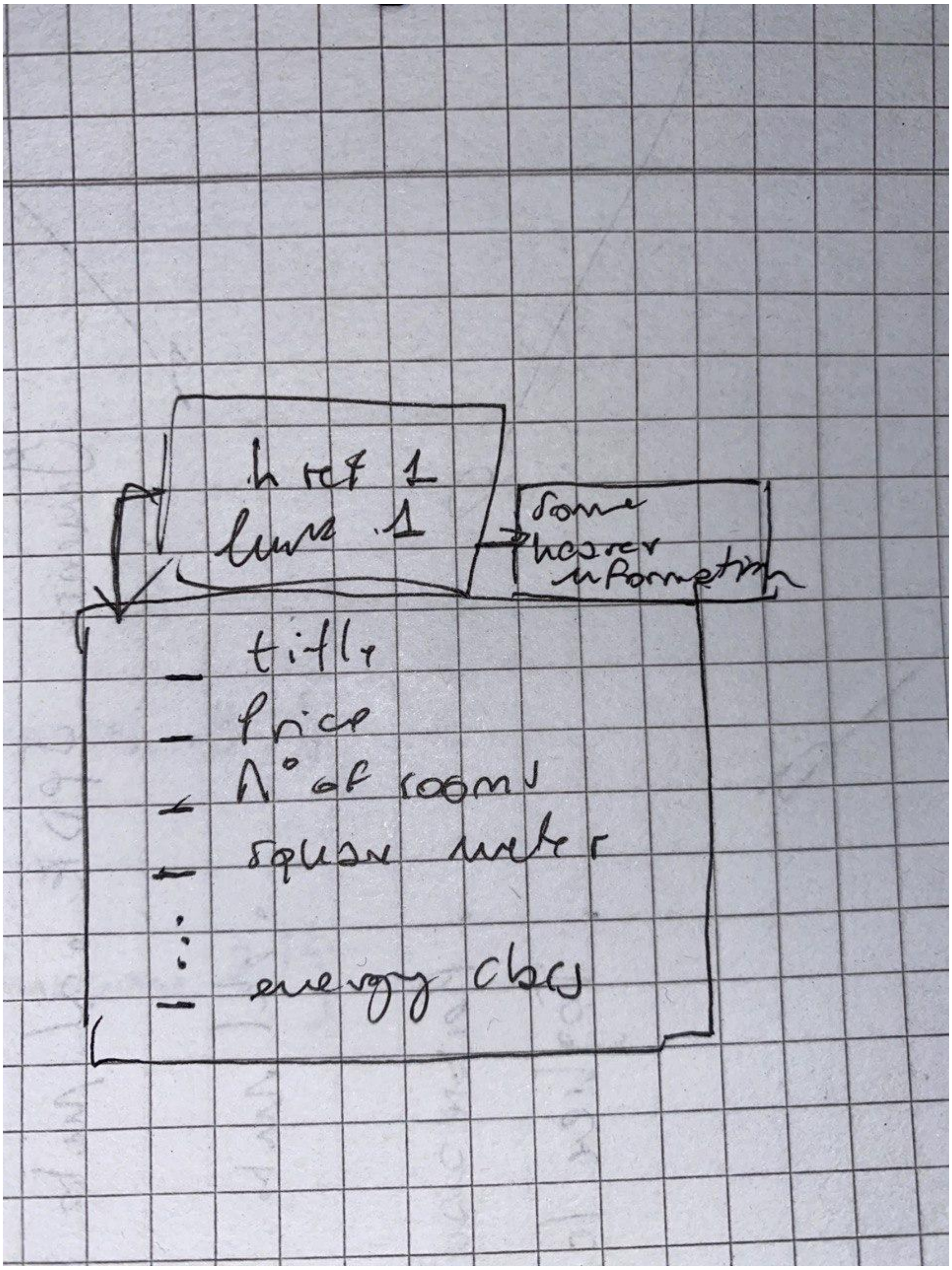


Figure 2.4: children_str

2.1.2 Immobiliare.it Webscraping content architecture with `rvest`

To start a general scraping function it is required a target url (i.e. the filtered root node url). Then a `html_session` nested list object is opened by specifying the url and the request data that the user need to send to the web server (see left part to dashed line image **??**). Information to be attached to the web server request will be further explored later, tough they are mainly three: User Agents, emails references and proxy servers. `html_session` objects contains a number of useful information such as: the url, the response, coockies, session times etc. Once the connection is established (request response 200) all the following operations rely on the opened session, in other words for the time being in the session the user will be authorized with the before-provided characteristics through the request. The list object contains mostly the html content of the webpage and that is where data needs to be parsed and collected. The list as said can disclose other interesting metadata that might be interesting but is beyond the scope of the analysis. The workflow below schematize what scraping is doing:

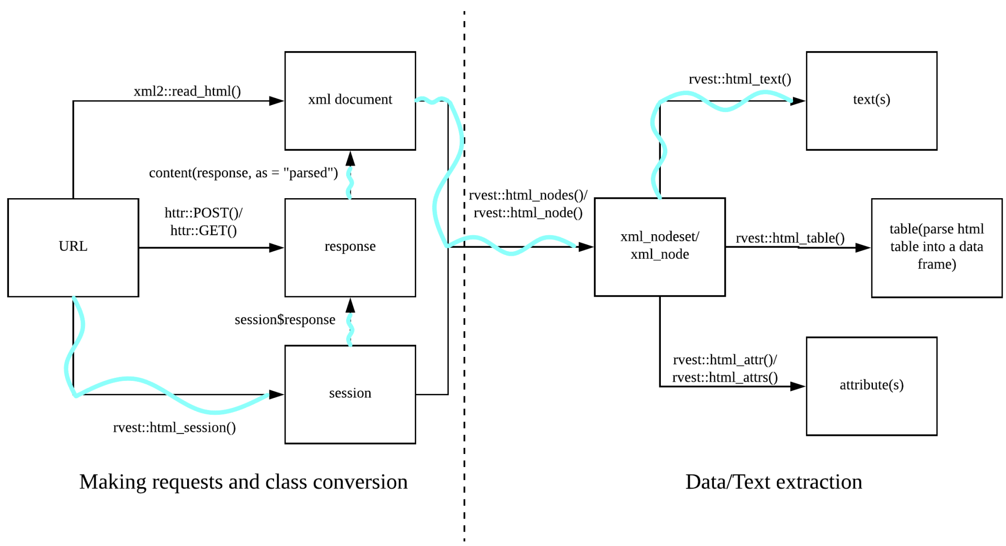


Figure 2.5: workflow

In the right part of the dashed vertical line is represented a sequence of `rvest`(Wickham, 2019) functions that follow a general step by step text comprehension rules. `rvest` first handles parsing the html content of the web page within the session object `read_html()`, secondly looking for a single node `html_nodes()` through CSS. CSS is the way to tell `rvest` to point to a precise node in the webpage and this is brought by query. Thirdly it converts the content into readable text with `html_text()`. The entire process can be also thought as an autoencoder, where adding decoding layer after layer on top of the starting url, it can be reached the final requested payload data wanted. The reason why the process appear straightforward and simple relies in the blending with `rvest` by the pipe %>% operator. Below it is shown a function that exemplifies scrapping the price.

```
scrapeprice.imm = function(session) {

opensess = read_html(session) price = opensess %>% html_nodes(css
    =".im-mainFeatures__title") %>% html_text() %>% str_trim()

if(is.null(price) || identical(price, character(0))) { price2 =
    opensess %>% html_nodes(css ='.im-features__value ,
    .im-features__title') %>% html_text() %>% str_trim()

if ("prezzo" %in% price2) { pos = match("prezzo",price2)
    return(price2[pos+1]) %>% str_replace_all(c("€"="","\\."="")) %>%
    str_extract( "\\-*\\d+\\.*\\d*") %>% str_replace_na() %>%
    str_replace("NA", "Prezzo Su Richiesta") } else {
    return(NA_character_) } } else { return(price) %>%
    str_replace_all(c("€"="","\\."="")) %>% str_extract(
    "\\-*\\d+\\.*\\d*") %>% str_replace_na() %>% str_replace("NA",
    "Prezzo Su Richiesta")

}
```

```
}
```

The function takes as a single argument a session object which is initialized in one other function. Then It reads the inner html content in the session storing the information into an obj called the `opensess`. Another obj is created, namely price, right after the pipe operator a css query into the html is called. The css query `.im-mainFeatures__title` points to a precise data stored in immobiliare web page header, right below the main title. Expectation are that price is a one-element chr vector, containing the price and some other unnecessary characters. Then the algorithm enters into the first `if` statement. The handler checks if the object `price` is empty. If it doesn't the algorithm jumps to the end of the algorithm and returns the cleaned quantity. But If it does it takes again the `opensess` and redirect to a second css query `.im-features__value , .im-features__title` where price, once again in the webpage, could be found. Please note that This is all done within the same session, so no more request additional information has to be sent. Since the latter css query points to data stored inside a list, for the time being the newly created obj price2 is a list containing various information. Then the algorithm flow enters into the second `if` statement that checks whether the `"prezzo"` is matched the list or not, if it does it returns the +1 position index element with respect to the "prezzo" positioning. This happens because data in price2 list are stored by couples sequentially, e.g. [title, "Appartamento Sempione", energy class, "G", "prezzo", 1200/al mese]. When it returns the element corresponding to +1 position index it applies also some data wrangling with `stringr` package to keep out overabundant characters. The function then escapes in the else statement by setting `price2 = NA_Character_` once no css query could be finding the price information. the *character-string* type has to be imposed due to fact that later they can not be bind. In other words if the function is evaluated for a url and returns the price quantity, but then is evaluated for url2 and outputs NA (no character) then results can not be combined into dataframe due to different types.

Once all the functions have been created they need to be called together and then data coming after them need to be combined. This is done by `get,data.catsing()` which at first checks the validity of the url, then takes the same url as input and filters it as a session object. Then simultaneously all the functions are called and then combined. All this happens inside a `foreach` parallel loop called by `scrape.all.info()`

```
scrape.all.info = function(url =
    "https://www.immobiliare.it/affitto-case...",
vedi = FALSE,
scrivi = FALSE,
silent = FALSE){
if (silent) {
start = as_hms(Sys.time()); cat('Starting the process...\n\n')
message('\nThe process has started in',format(start,usetz = TRUE))
}
# open parallel multisession
cl = makeCluster(detectCores()-1) #using max cores - 1 for parallel
    processing
registerDoParallel(cl)
start = as_hms(Sys.time())

if (silent) {
message('\n\nStart all the requests at time:', format(start,usetz =
    T))
}
ALL = foreach(i = seq_along(links),
.packages = lista.pacchetti,
.combine = "bind_rows",
.multicombine = FALSE,
.export ="links" ,
.verbose = TRUE,
.errorhandling='pass') %dopar% {
source("utils.R")
sourceEntireFolder("functions_singolourl")
get.data.catsing = function(singolourl){

# dormi()
#
if(!is_url(singolourl)){
stop(paste0("The following url does not seem either to exist or it is
    invalid", singolourl))
}

session = html_session(singolourl, user_agent(agents[sample(1)]))
if (class(session) == "session") {
session = session$response
}

id = tryCatch({scrapehouse.ID(session)},
error = function(e){ message("some problem occured in
    scrapehouse.ID") })
lat = tryCatch({scrapelat.imm(session)},
error = function(e){ message("some problem occured in scrapelat.imm")
    })
long = tryCatch({scrapelong.imm(session)},
```

```
error = function(e){ message("some problem occured in
    scrapelong.imm") })
location = tryCatch({take.address(session)},
error = function(e){ message("some problem occured in take.address")
    })
condom = tryCatch({scrapecondom.imm(session)},
error = function(e){ message("some problem occured in
    scrapecondom.imm") })
buildage = tryCatch({scrapeagebuild.imm(session)},
error = function(e){ message("some problem occured in
    scrapeagebuild.imm") })

...

combine = tibble(ID = id,
LAT = lat,
LONG = long,
LOCATION = location,
CONDOM = condom,
BUILDAGE = buildage,

...


return(combine)
}
stopCluster(cl)
return(ALL)
}
```

The skeleton constitutes a standard format adopted for many other scraping function used in the analysis. Being equal the css query what it changes is the matching term, i.e. "numero camere" instead of "prezzo" to look for how many rooms there are in the house. This is true for all the information contained in the list accessed by the fixed css query. Those that are not they are a few and they do not need to be scraped. In addition some other functions outputs need to undergo to further heavy cleaning steps in order to be usable As a consequence oh that functions need also to be broken down by pieces into many single .R files whose names correspond to each important information. Below it is printed the tree structure folder that composes the main elements of the scraping procedure. It can be notices that the two folders, namely functions_singolourl and functions_url enclose all the single functions that allows to grab single information from session. Folders with a customized function are then sourced within the two main functions, scrape.all and scrape.all.info so data can be extracted.

```
 levelName
1  immobiliare.it-WebScraping
2   ¦--functions_singolourl
3   ¦   ¦--0scrapesqfeetINS.R
4   ¦   ¦--0scrapenroomINS.R
5   ¦   ¦--0scrapepriceINS.R
6   ¦   ¦--0scrapetitleINS.R
7   ¦   ¦--ScrapeAdDate.R
8   ¦   ¦--ScrapeAge.R
9   ¦   ¦--ScrapeAgeBuilding.R
10  ¦   ¦--ScrapeAirConditioning.R
11  ¦   ¦--ScrapeAptChar.R
12  ¦   ¦--ScrapeCatastInfo.R
13  ¦   ¦--ScrapeCompart.R
14  ¦   ¦--ScrapeCondom.R
15  ¦   ¦--ScrapeContr.R
16  ¦   ¦--ScrapeDisp.R
17  ¦   ¦--ScrapeEnClass.R
18  ¦   ¦--ScrapeFloor.R
19  ¦   ¦--ScrapeHasMulti.R
20  ¦   ¦--ScrapeHeating.R
21  ¦   ¦--ScrapeHouseID.R
22  ¦   ¦--ScrapeHouseTxtDes.R
23  ¦   ¦--ScrapeLAT.R
24  ¦   ¦--ScrapeLONG.R
25  ¦   ¦--ScrapeLoweredPrice.R
26  ¦   ¦--ScrapeMetrature.R
27  ¦   ¦--ScrapePhotosNum.R
28  ¦   ¦--ScrapePostAuto.R
29  ¦   ¦--ScrapePropType.R
30  ¦   ¦--ScrapeReaReview.R
```

```
¦    ¦--ScrapeStatus.R
¦    ¦--ScrapeTotPiani.R
¦    ¦--ScrapeType.R
¦    °--take_location.R
¦--scrapeALL.R
¦--scrapeALLINFO.R
¦--functions_url
¦    ¦--ScrapeHREF.R
¦    ¦--ScrapePrice.R
¦    ¦--ScrapePrimaryKey.R
¦    ¦--ScrapeRooms.R
¦    ¦--ScrapeSpace.R
¦    °--ScrapeTitle.R
¦--libs.R
¦--utils.R
¦--README.Rmd
¦--README.md
°--simulations
     ¦--rt_match_vs_forloop.R
     °--runtime_simul.R
```

2.2 Scraping Best Practices and Robottxt

Robots.txt files are (rivedi citation) a way to kindly ask webbots, spiders, crawlers, wanderers and the like to access or not access certain parts of a webpage. The de facto 'standard' never made it beyond a informal "Network Working Group INTERNET DRAFT". Nonetheless, the use of robots.txt files is widespread (e.g. https://en.wikipedia.org/robots.txt, https://www.google.com/robots.txt) and bots from Google, Yahoo and the like will adhere to the rules defined in robots.txt files, although their *interpretation* of those rules might differ.

Robots.txt files are plain text and always found at the root of a website's domain. The syntax of the files in essence follows a fieldname: value scheme with optional preceding user-agent: … lines to indicate the scope of the following rule block. Blocks are separated by blank lines and the omission of a user-agent field (which directly corresponds to the HTTP user-agent field) is seen as referring to all bots. # serves to comment lines and parts of lines. Everything after # until the end of line is regarded a comment. Possible field names are: user-agent, disallow, allow, crawl-delay, sitemap, and host. For further notions (Meissner and Ren, 2020, Google (2020))

Some interpretation problems:

- finding no robots.txt file at the server (e.g. HTTP status code 404) implies that everything is permitted
- subdomains should have there own robots.txt file if not it is assumed that everything is allowed
- redirects involving protocol changes - e.g. upgrading from http to https - are followed and considered no domain or subdomain change - so whatever is found at the end of the redirect is considered to be the - robots.txt file for the original domain
- redirects from subdomain www to the doamin is considered no domain change - so whatever is found at the end of the redirect is considered to be the robots.txt file for the subdomain originally requested

For the thesis purposes it has been designed a dedicated function to inspect whether the domain requires specific actions or prevents some activity on thw target website. The following `checkpermission()` function has been integrated inside the scraping architecture and it is called once at the very beginning.

```r
library(robotstxt)
dominio = "immobiliare.it"
```

```r
checkpermission = function(dom) {


    robot = robotstxt(domain = dom)
    vd = robot$check()[1]
    if (vd) {
        cat("\nrobot.txt for", dom, "is okay with scraping!")
    } else {
        cat("\nrobot.txt does not like what you're doing")
        ## stop()
    }
}
checkpermission(dominio)
```

```
##
## robot.txt for immobiliare.it is okay with scraping!
```

Further improvements in this direction came from the `polite` package (Perepolkin, 2019) which combines the power of the `robotstxt`, the `ratelimitr` to rate-limiting requests and the `memoise` for response caching. This package is wrapped up around 3 simple but effective ideas:

> The three pillars of a polite session are seeking permission, taking slowly and never asking twice.

The three pillars constitute the Ethical web scraping manifesto (Densmore, 2019) which are common shared practises that are aimed to self regularize scrapers. This has not nothing to do with law but since many scrapers themselves, as website administrators or analyst, have fought with bots. Bots might fake out real client logs and might stain analytics, so here it is born the choice to fine common ground and politely ask for permission.

2.3 User agents, Proxies, Handlers

HTTP requests to the website server by web clients come with some mandatory information packed in it. The process according to which HTTP protocols allow to exchange information can be easily thought with an everyday real world analogy. As a generic person A rings the door's bell of person B's house. A comes to B door with its personal information, its name, surname, where he lives etc. At this point B may either answer to A requests by opening the door and let him enter given the set of information he has, or it may not since B is not sure of the real intentions of A. This typical everyday situation in nothing more what happens billions of times on the internet everyday, the user (in the example above A) is interacting with a server website (part B) sending packets of information. If a server does not trust the information provided by the user, if the requests are too many, if the requests seems to be scheduled due to fixed sleeping time, a server can block requests. In certain cases it can even forbid the user log to the website. The language the two parties exchanges are coded in numbers that ranges from 100 to 511, each of which has its own specific significance. A popular case of this type of interaction occurs when users are not connected to internet so the server responds 404, page not found. Servers are built with a immune-system like software that raises barriers and block users to prevent dossing or other illegal practices.

This procedure is a day to day issue to people that are trying to collect information from websites. Google performs it everyday with its spider crawlers, which are very sophisticated bots that scrapes over a enormous range of websites. This challenge can be addressed in multiple ways, there are some specific Python packages that overcome this issue. The are also certain types of scraping as the Selenium web driver automation that simulates browser automation. Selenium allows the user not to be easily detected by the server immune system and peaceful. Precautions have not been taken lightly, and a simple but effective approach is proposed.

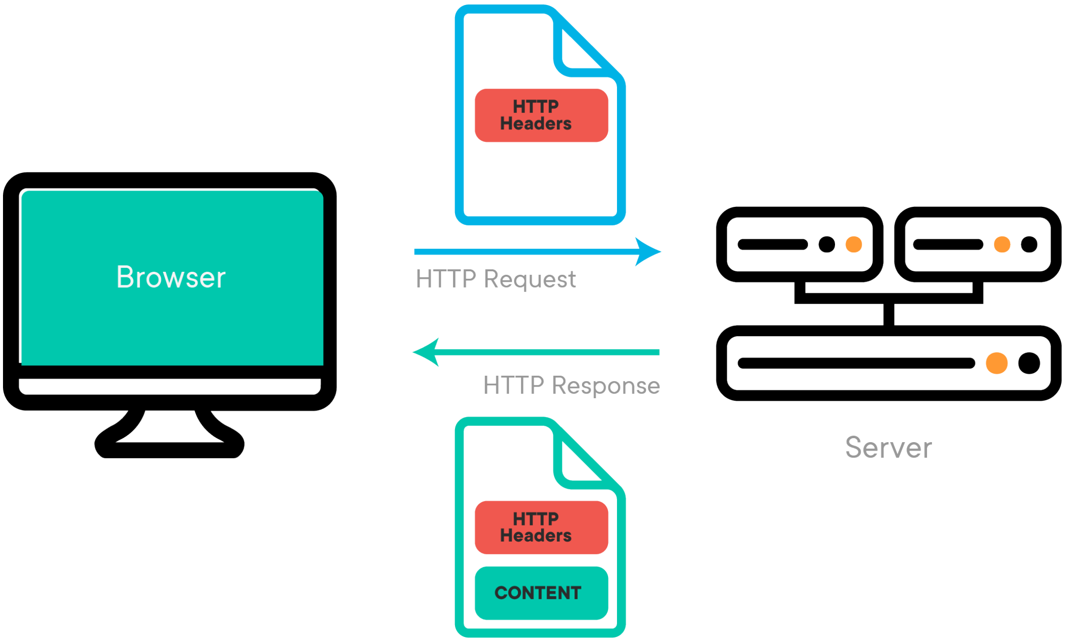


Figure 2.6: How Web Works

2.3.1 User agents Spoofing

A user agent (WhoIsHostingThis.com, 2020) is a string of characters in each web browser that serves as an identification agent. The user agent permits the web server to be able to identify the user operating system and the browser. Then, the web server uses the exchanged information to determine what content should be presented to particular operating systems and web browsers on a series of devices. The user agent string contains the user application or software, the operating system (and their versions), the web client, the web client's version, and the web engine responsible for the content display (such as AppleWebKit). The user agent string is sent in form of a HTTP request header. Since the User Agents acts as middle man between the client request and the server response what it would be better doing is to actively faking it so that each time a web browser presents himself to a web server it has a different specifications, different web client, different operating system and so on.

The simple approach followed was building a vector of samples of different existing and updated User Agents (UA). Then whenever a request from a web browser is served to a web server 1 random sample string is drawn from the user agents deck. So each time the user is sending the requests it appears to have a different "identity". Below the user agents rotating pool:

```
set.seed(27)
agents = c("Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36",
    "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36",
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/602.2.14 (KHTML, like Gecko) Version/10.0.1 Safari/602.2.14",
    "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.71 Safari/537.36",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.98 Safari/537.36",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.98 Safari/537.36",
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.71 Safari/537.36",
    "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0")
agents[sample(1)]
```

```
## [1] "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36"
```

An improvement to this could be further using rotating proxies. A proxy server acts as a gateway between the web user and the web server. It's an intermediary server himself, separating end clients from the websites they are browsing. Proxy servers provide varying layers of functionality, security, and privacy are some of the examples. While the user is exploiting a proxy server, internet traffic flows through the proxy server on its way to the server you requested. The request then comes back through that same proxy server and then the proxy server forwards the data received from the website to you. Many proxy servers are offered in a paid version so in this case since security barriers of the target website are not high they will not be implemented. It has to be mentioned that many online services are providing free proxies but the turnaround of this solutions are many, two of them are: - Proxies to be free are widely shared among people, so as long as someone has used them for illegal purposes the user is inheriting their mistakes when caught. - Some of those proxies, pretty all the ones coming from low ranked websites, are tracked so there might be a user privacy violation issue.

2.3.2 Handlers

During scraping many difficulties might be met. Starting from the things that have been previously explained at the chapter start. Some of them are: a

varying url structure so that the html tree hierarchies are changed, payloads are not correctly parsed with respect to previous scraping sessions. ending with the ones that have just been said a few lines ago. Handlers and trycatch error workarounds are explicitly built in this sense. The continuous testing of the scraping functioning while developing has required the maintainer to track where the error occurs. A few numbers: the "agglomerative" function `get.data.catsing()` triggers more than 36 scrapping functions that are going to catch 36 different data pieces. If one of them went missing then the other one would be missing too. Then when row-data is binded together one entry column might not exists making the process fail.
Then the solution to that is to call inside the aggolmerative function as much as trycatch as many scrapping functions are involved. The trycatch can leverage the gap by introducing a specified quantity and alerting that something went wrong. On top of that many other handlers are called throughout the procedure:

- `get_ua()` verifies that the user agent coming from the session request is not the default one

```
get_ua = function(sess) {
    stopifnot(is.session(sess))
    stopifnot(is_url(sess$url))
    ua = sess$response$request$options$useragent
    return(ua)
}
```

- `is_url()` verifies that the url input needed has the canonic form. This is done by a REGEX query.

```
is_url = function(url) { re =
    "^(?:(?:http(?:s)?|ftp)://)(?:\\S+(?::(?:\\S)*)?@)?(?:(?:[a-z0-9¡-<ef><U+00BF><U+00BF>](?:-)*)*(?:[a-z0-9¡-<ef><U+00BF><U+00BF>])+)(?:\\.(?:[a-z0-9¡
    grepl(re, url) }
```

- `.get_delay()` checks through the robotxt file if a delay between each request is kindly welcomed.

```
.get_delay = function(domain) {

message(sprintf("Refreshing robots.txt data for %s...", domain))

cd_tmp = robotstxt::robotstxt(domain)$crawl_delay

if (length(cd_tmp) > 0) {
star = dplyr::filter(cd_tmp, useragent=="*")
if (nrow(star) == 0) star = cd_tmp[1,]
as.numeric(star$value[1])
} else {
10L
}

}

get_delay = memoise::memoise(.get_delay)
```

2.3.3 Parallel Computing

Since are opened as many sessions as single links and since for each link are going to be called many functions computations can take a while. Reporting it into numbers in order to get a usable dataset 300 pages are considered, which at their own time contains 25 links per page, for which almost 34 different functions are called. For the sake of the analysis and the app this should not bother the end user because scraping tasks are performed daily by a scheduler and a single day is sufficient amount of runtime. In any case functions are optimized following optimization criteria. Run time is crucial when dealing with active web pages and time to market in real estate is very important, here comes the need to have always fresh data. A way to secure fresh new data is to have lightweight computation on a single machine o heavy computation spread among a pool of different machines, in this case multi-threaded sessions. All the following runtime examinations are performed on the `scrape` functions a lightweight version of the final scraping API function. A first attempt was using `furrr` package (Vaughan and Dancho, 2018) which enables mapping through a list with the `purrr`, along with a `future` parallel backend. The approach has shows decent results, but its run time drastically increases when more requests are sent. This leads to a preventive conclusion about the computational complexity: it has to be at least linear. Empirical demonstrations have been made:

```
library(furrr)

vecelaps = c()
start = c()
end = c()
for (i in 1:len(list.of.pages.imm[1:20])) { start[i] = Sys.time()
    cat("\n\n run iteration", i, "over 20 total\n")
    list.of.pages.imm[1:i] %>% furrr::future_map(get.data.caturl,
    .progress = T) %>% bind_rows()

end[i] = Sys.time() vecelaps[i] = end[i] - start[i] }
```

```
furrrmethod = tibble(start,
end,
vettoelaps)

# ggplot (themed) run time meausurament method 1
p = ggplot(furrrmethod,aes(x=1:20, y=vettoelaps)) +
geom_line( color="steelblue") +
geom_point() +
xlab("Num URLS evaluated") +
ylab("run time (in seconds)") +
```

```
ggtitle("Run-Time for First method (furrr multisession)") +
stat_smooth(method=lm) +
theme_nicco()
p
```

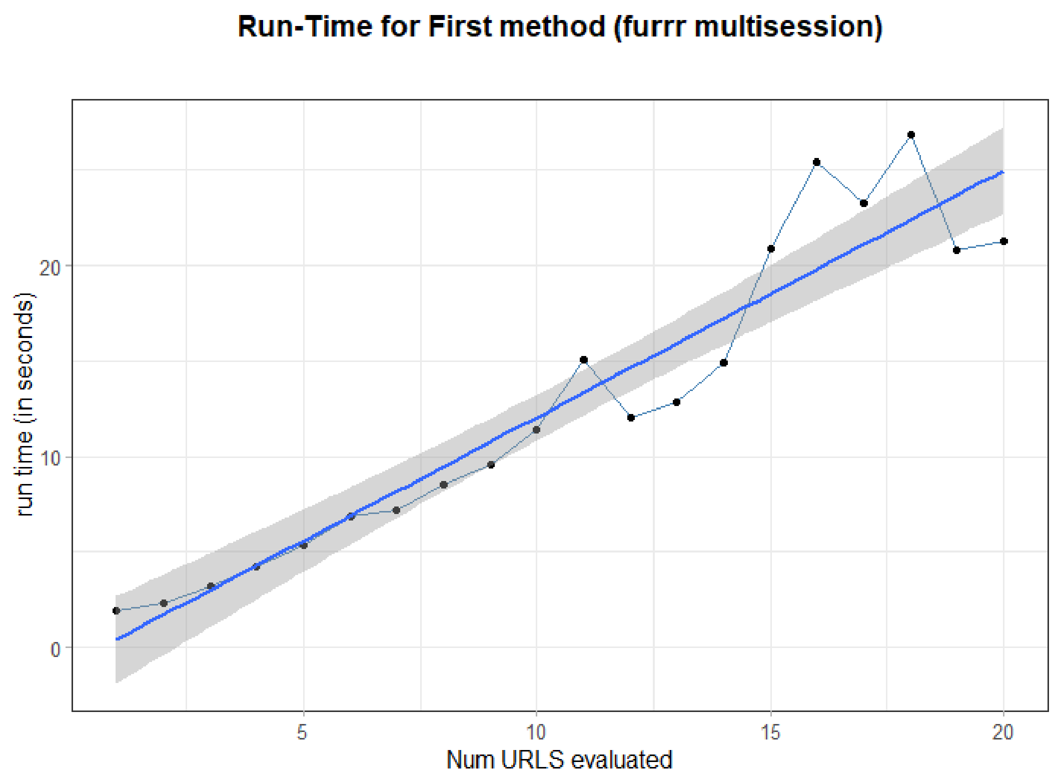


Figure 2.7: computational complexity analysis with Furrr

On the x-axis the number of urls which are evaluated together, on y axis the run time taken measured in seconds. Iteration after iteration the urls considered are increased by a unity. Looking at the smooth curve in between confidence lines a first guess might be linear time $\mathbf{O}(n)$, where n are the links considered.

A second attempt tried to explore the `foreach` package (Microsoft and Weston, 2020). This interesting package enables a new looping construct for executing R code in an iterative way. The main reason for using the `foreach` package is that it supports *parallel execution*, that is, it can execute those repeated operations on multiple processors/cores on the computer, or on multiple nodes of a cluster. The construction is straightforward:

- start clusters on processors cores
- define the iterator, in this case i = to the elements that are going to be looped through
- `.packages`: Inherits the packages that are used in the tasks define below
- `.combine`: Define the combining function that bind results at the end (say cbind, rbind or tidyverse::bind_rows). It has to be a string.
- `.errorhandling`: specifies how a task evaluation error should be handle.
- `%dopar%`: the dopar keyword suggests foreach with parallelization method
- then the function within the elements are iterated
- close clusters

One major important thing concerns the fact that th function within iterators repeats itself should be standalone. For standalone it is meant that the body function should be defined inside, as it would be a an empty environment. As a matter of fact packages has to be taken inside each time, and if the function is not defined inside body (or is not source from some other locations) the clusters can not operate and an error is printed.

```
cl = makeCluster(detectCores()-1)
registerDoParallel(cl)

vettoelaps1 = c()
start1 = c()
end1 = c()
for (j in 1:len(list.of.pages.imm[1:20])) {
start1[j] = Sys.time()
cat("\n\n run iteration",j,"over 20 total\n")
foreach(i = seq_along(list.of.pages.imm[1:j]),
.packages = lista.pacchetti,
.combine = "bind_rows",
.errorhandling='pass') %dopar% {
```

```
source("main.R")
x = get.data.caturl(list.of.pages.imm[i])
}
end1[j] = Sys.time()
vettoelaps1[j] = end1[j] -start1[j]
}
stopCluster(cl)
```

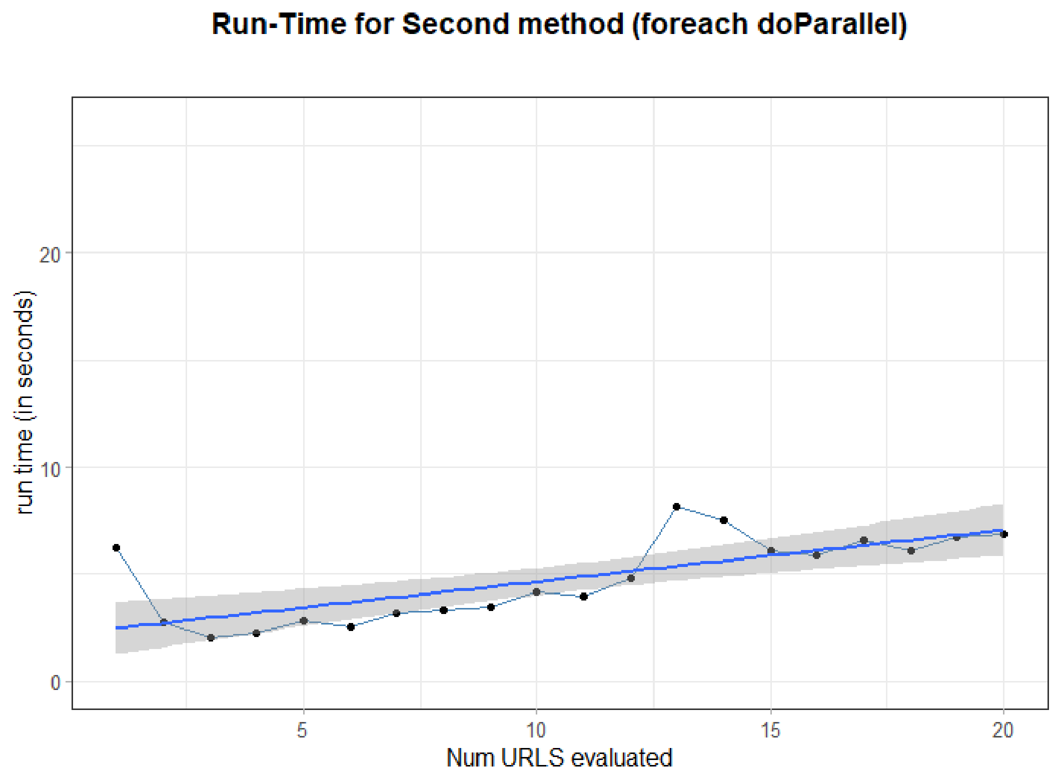


Figure 2.8: computational complexity analysis with Furrr

It can be seen quite easily that the curve now is flattened and first guess is logarithmic time $\mathbf{O}(log(n))$.

A further improvement could be obtained using a new package called `doAzureParallel` which is built on top of foreach. doAzureParallel enables different Virtual Machines operates parallel computing throughout Microsoft Azure cloud, but this comes at a substantial monetary cost. This would be a perfect match given that parallel methods seen before accelerates the number of requests sent among different processors or cluster, even though actually what it is really needed it is something that separates session. Unleashing Virtual Machines permits from one hand to further increase computational power and the number of potential requests, from the other it can splits requests among different user agents (a pool for each VM) masquerading even better the scraping automation.

2.4 Further Improvements

The main challenge remains unsolved since each single elements has been finely optimized but API continues to be unstable. What it can not be optimized are the system ad choices to change the layout of the page or to change the url structure (allocation of datain the web page). The way the API is designed really facilitates responsive fast debugging but this can not be by any means automatized. The API necessitates frequently to resort to continuous integration (i.e. CI) review to verify the working status. Moreover Error messages can not really be undestood sometimes, this is due to functions that are called within a parallel beckend that does not allow [refererenza su stack overflow di errore di print] to print error on console. So each time an error occur the "main" functions needs to be taken out of from the foreach cluster and evaluated in isolation where the loop stops. This is trivial and time consuming but for the time being no solutions have been provided.

2.5 Legal Challenges (ancora non validato)

"Data that are online and public are always available for all" is never a good answer to the question "Can I use those web data to my scope". Immobiliare.it[1] is not providing any open source data from its own database neither it is planning to do so in the future. It has not even provided a paid API through

[1]https://www.immobiliare.it/

which might be possible to perform analysis. A careful reading of their terms has been done to get this service running without any legal consequence, as a reference further information can be collected in their specialized section[2]. Nevertheless the golden standard for scraping was respected since the robottxt file talks clearly permitting any action as demonstrated above. So if it might be the case of misuse of their intellectual their property, it will be also the case of misunderstanding between servers response and immobiliare.it intent to preserve their own intellectual property. What it has been really surprising are the low barriers to request information to their severs with respect to other online players. Best practices are applied and delayed requests have been sent to normalized traffic congestion in compliance to anti-dossing rules. But scraping criteria followed are fully based on common shared best practises (see section 2.2), and not any sort of agreements between parties (i.e. general terms and agreements). As a result a possible approach could be applying scraping procedures without any prevention. It would not surely cause any sort of disservice for the website but in the long run when scrapers are few, but in the long run it will cause delays as soon as subjects will increase. Totally different was the approach proposed by Idealista.com, which is a comparable to immobiliare.it. Idealista does block requests if they are not in compliance with their servers predefined ones. User agents in this case must be rotated quite frequently and as soon as a request does not fall within the pool of user agents (i.e. is labled as web bot) it is immediately blocked and 404 response is sent back.

- Idealista content is composed by Javascript so and html parser can no get that.
- Idealista blocks also certain web browser that have a demonstrated “career” in scraping procedures.

All of this leads to accept that entry barriers to scrape are for sure higher than the one faced for Immobiliare. The reticence to share data could be a reflex

[2] https://www.immobiliare.it/terms/

on how big idealista is; as a matter of fact it has a heavy market presence in some of the Europe real estates country as Spain and France. So what they thought was to raise awareness on scraping procedure that in a certain way can hurt their business. This has been validated by the fact that prior filtering houses on their website a checkbox has to be signed. The checkbox make the user sign an agreement on their platform according to which data can not be misused and it belongs their intellectual property.

Chapter 3

Infrastructure

In order to provide a fast and easy to use service to the end user many technologies have been involved. Challenges in scraping as pointed out in section 2.4 are many and still some remains unsolved. Challenges regards not only scraping but also the way the service has to respond to the users. Service has to be fast otherwise data become obsolete and so happen to analysis that relied on those data. Service has to be deployed so that in does not only run locally. Service needs to be scaled when needed since when the number of users increases the run time performance should not decrease. Service has to be maintained up to date so that each function can be reshaped with respect to immobiliare.it layout changes. On the other hand code behind the service has to be kept freezed to certain version, so that when packages are updated service still runs. Furthemore service has to be also secured granting access only to the ones authorized. In the end Service has to be run at a certain given times and storing data on cloud, so that it can be tracked back the evolution of the phenomenon. Open source solutions are available for back-end and front-end to meet the requirements before. Documentations related to technologies served are up to date and offer flexible solutions to embed the R environment. As a general discussion technologies used can be thought as the distance between something running locally on the laptop and something that is actually put into production, seen by company stakeholders, solving business problem. When such technologies are applied data scientist and counterparts gradually close the gap. Insights are better communicated (they can be interactive or automated) and services can be shared over a wider range of subjects. Nonetheless when the infrastructure is made with vision then integrating or substituting existing technologies is not trivial. Anyway technologies can not be always embedded because they might be exclusively designed to work only on certain back ends, therefore some choices are not into discussion. With foresight RStudio by setting future-oriented guidelines has spent a lot of effort giving its users an easy, integrated and interconnected environment. By that it is meant that the RStudio community has tried to either integrate or open the possibility to a number of technologies that fill the blanks in their weaker parts. On top of many, an entire package has been dedicated to democratize REST APIs (Plumber (Trestle Technology, LLC, 2018)). As a further example developers in RStudio have created an entire new paradigm i.e. Shiny (Chang et al., 2020), a popular web app development package, that enforces the developer to have front-end and back-end technologies tied up in the same IDE. They also added performance monitoring and optimization packages that are fitted into shiny such as shinytest [metti tag] and shinyloadtest [metti tag] to simulate sessions and verify network traffic congestion. The actual idea is to provide a REST API with 4 endpoints which calls parallelized scraping functions. On the other hand (chapter 2) a daily scheduler, exposing one API endpoint, produces and later stores a .csv file in a NOSQL mongoDB Atlas could database. It is all meant to be containerized in a Linux env (Ubuntu distr) docker container hosted by a AWS EC2 server. API endpoints are going to be secured with https protocols and protected with authentication by nginx reverse proxy. A Shiny app calls an endpoint with specified parameters which returns up to date data from the former infrastructure. It then models data with bayesian spatial methods 6.

Technologies involved are:

- GitHub version control

- Scheduler cron job, section 3.1
- Docker containers, section 3.2
- Plumber REST API, section 3.3.1
- NGINX reverse proxy, section 3.4
- AWS (Amazon Web Services) EC2 3.5
- MongoDB Atlas
- Shiny, see chapter 8

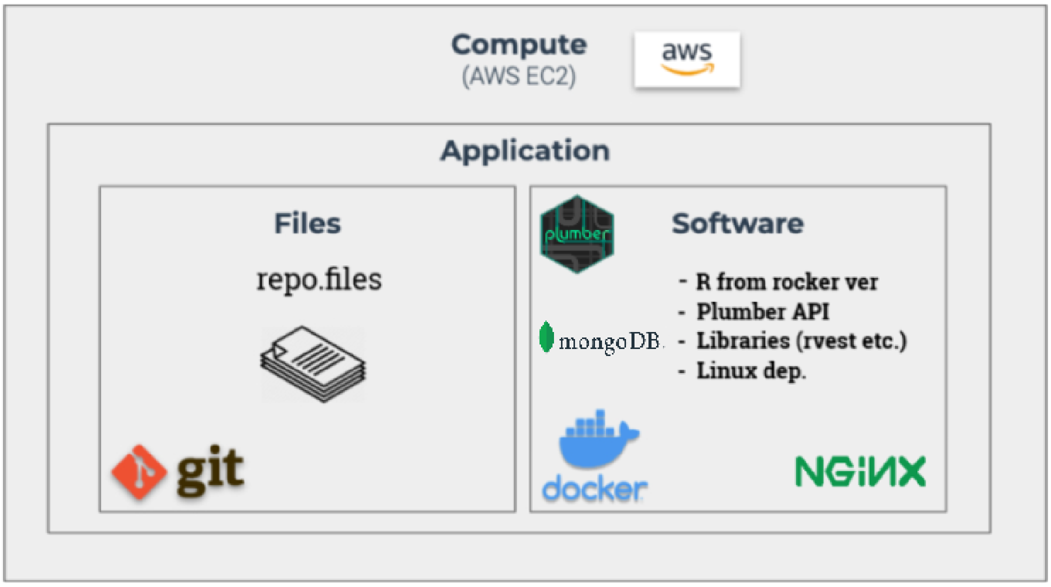


Figure 3.1: complete infrastructure (Matt Dancho source)

As a side note even each single part of this thesis has been made stand alone and can be easily accessed and modified through a gitbook deployed at @link[1]. RMarkdown knits the .rmd files extension and coverts them into .html files (the book's chapters). All the documents are then pushed to a Github repository with git. By a simple trick, since all the files are static html, they can be displayed through GH pages as it is a website whose link is a github subdomain. The pdf output for the thesis can be obtained by clicking the download button, then choosing output pdf in the upper banner. A Latex engine (Xelatex) wrapped into the website compiles the sequence of RMarkdown documents according to a predefined .tex template. Formatting can be tuned by modifying

[1]https://niccolosalvini.github.io/Thesis/

the .yml file, that sets general instructions for the pdf document. All of this has been possible thanks to Bookdown (Xie, 2020) once again a R well documented package (Xie, 2016) to build interactive books along with RMarkdown (Allaire et al., 2020).

Some of the main technologies implied will be viewed singularly, nonetheless for brevity some of them will be skipped.

3.1 Scheduler

A Scheduler in a process is a component on a OS that allows the computer to decide which activity is going to be executed. In the context of multi-programming it is thought as a tool to keep CPU occupied as much as possible. As an example it can trigger a process while some other is still waiting to finish. There are many type of scheduler and they are based on the frequency of times they are executed considering a certain closed time neighbor.

- Short term scheduler: it can trigger and queue the "ready to go" tasks
    - with pre-emption
    - without pre-emption

The ST scheduler selects the process and It gains control of the CPU by the dispatcher. In this context we can define latency as the time needed to stop a process and to start a new one.

- Medium term scheduler
- Long term scheduler

for some other useful but beyond the scope refereces, such as the scheduling algorithm the reader can refer to (Wikiversità, 2020).

3.1.1 Cron Jobs

Cron job is a software utility which acts as a time-based job scheduler in Unix-like OS. Linux users that set up and maintain software environments exploit cron to schedule their day-to-day routines to run periodically at fixed times, dates, or intervals. It typically automates system maintenance but its usage is very flexible to whichever needed. It is lightweight and it is widely used since it is a common option for Linux users. The tasks by cron are driven by a crontab file, which is a configuration file that specifies a set of commands to run periodically on a given schedule. The crontab files are stored where the lists of jobs and other instructions to the cron daemon are kept.

Each line of a crontab file represents a job, and has this structure

```
# ┌───────────── minute (0 - 59)
# │ ┌───────────── hour (0 - 23)
# │ │ ┌───────────── day of the month (1 - 31)
# │ │ │ ┌───────────── month (1 - 12)
# │ │ │ │ ┌───────────── day of the week (0 - 6) (Sunday to Saturday;
# │ │ │ │ │                                   7 is also Sunday on some systems)
# │ │ │ │ │
# │ │ │ │ │
# * * * * * <command to execute>
```

Figure 3.2: crontab

Each line of a crontab file represents a job. This example runs a shell named scheduler.sh at 23:45 (11:45 PM) every Saturday. .sh commands can update mails and other minor routines.

45 23 * * 6 /home/oracle/scripts/scheduler.sh

Some rather unusual scheduling definitions for crontabs can be found in this reference (Wikipedia contributors, 2020). Crontab's syntax completion can be made easier through this[2] GUI.

The cron job needs to be ran on scraping fucntions at 11:30 PM every single day. The get_data.R script first sources an endpoint function, then it applies the function with fixed parameters. Parameters describe the url specification,

[2] https://crontab.guru/

so that each time the scheduler runs the get_data.R collects data from the same source. Day after day .json files are generated and then stored into a NOSQL *mongoDB* database whose credentials are public. Data are collected on a daily basis with the explicit aim to track day-by-day changes both in the new entries an goners in rental market, and to investigate the evolution of price differentials over time. Spatio-Temporal modeling is still quite unexplored, data is saved for future used. Crontab configuration for daily 11:30 PM schedules has this appearance:

30 11 * * * /home/oracle/scripts/get_data.R

Since now the computational power comes from the machine on which the system is installed. A smarter solution takes care of it by considering run time limits and the substantial inability to share data. To a certain extent what it has been already done since now might fit for personal use: a scheduler can daily execute the scraping scripts and generate a .csv file. Furthermore an application can rely on those data, but evident reasons suggest that it does not suite any need. What it will do the trick would be an open source dedicated software environment or *container* that will contains scraping functions and a scheduler on cloud solving a pair of the problems arisen. This problem can be addressed with a technology that has seen a huge growth in its usage in the last few years.

3.2 Docker

3.2.1 What is Docker

Docker is a software tool to create and deploy applications using containers. *Docker containers* are a standard unit of software (i.e. software boxes) where everything needed for applications, such as libraries or dependencies can be run reliably and quickly. Furthermore they are also portable, in the sense that they can be taken from one computing environment to the following. Docker containers by default run on kernel Linux OS. Containers can be thought as an abstraction at the app layers that groups code and dependencies together. One major advantage of containers is that multiple containers can run on the same machine with the same OS. Each container can run its own isolated process in the user space, so that each task is complementary to the other. Containers are lightweight and take up less space than Virtual Machines (container images are files which can take up typically tens of MBs in size), can handle more applications and require fewer Virtual Machines and OSs.

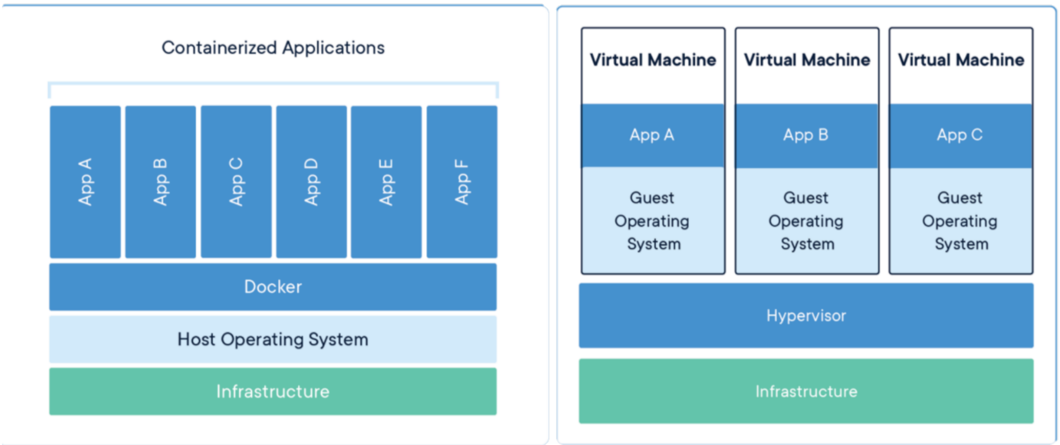


Figure 3.3: docker container vs VM

When containers are built *Docker container Images* are created and can be open sourced with Docker Hub *Docker Hub* is a web service provided by Docker for searching and sharing container images with other teams or developers in the community. Docker Hub behind authentication allows to integrate GitHub in the Docker project repository. Once the connection is authorized on local machine changes are made and then pushed by version control to the remote GH repository. The push command triggers the automatic building (pre-set by the user, branch should be given) of the image in the docker hub repository. The just created docker image can be tagged so that firstly it is recognizable and secondly can be reused in the future. Once the building stage is completed the DH repository can be pulled and then run locally on machine or cloud, see section 3.5. Docker building and testing images can be very time consumin. R packages can take a long time to install because code has to be compiled, especially if using R on a Linux server or in a Docker container. Rstudio package manager[3] includes beta support for pre-compiled R packages that can be installed faster. This dramatically reduces packages time installation (Nolis, 2020). In addition an open source project named rocker[4] have already narrowed the path building custom R images for R and docker users. What can be read from their own website about them: “The rocker project provides a collection of containers suited for different needs. find a base image to extend or images with popular software and optimized libraries pre-installed. Get the latest version or a reproducible fixed environment.” Enabling caching when container images are built heavily shorten the total duration for containerization.

3.2.2 Why Docker

Indeed, an employment-related search engine, released an article on 2019 displaying changing trends from 2015 to 2019 in Technology Job market. Many changes are relevant in key technologies. Two among the others technologies (i.e. docker and Azure) have experienced a huge growth and both refer to the same demand input: *containers* . The landscape of Data Science is changing (was previously an Economist at the Indeed Hiring Lab, 2020) from reporting to application building: In 2015 - Businesses reports drive better decisions In 2020 - Businesses need apps to empower better decision making at all levels

For all the things said what docker is bringing to business (Inc., 2020b):

- *Speed application deployment* : containers include the minimal run time requirements of the application, reducing their size and allowing them to be deployed quickly.
- *Portability across machines* : an application and all its dependencies can be bundled into a single container that is independent from the host version of Linux kernel, platform distribution, or deployment model.

[3]https://packagemanager.rstudio.com/client/#/
[4]https://www.rocker-project.org/images/

Top 20 tech skills in 2019

Percent of all tech jobs, change September 2014 to September 2019

Rank	Skill	2014 share	2019 share	% change
1	sql	23.6%	21.9%	-7%
2	java	19.7%	20.8%	6%
3	python	8.1%	18.0%	123%
4	linux	14.9%	14.9%	0%
5	javascript	12.4%	14.5%	17%
6	aws	2.7%	14.2%	418%
7	c++	10.6%	10.7%	1%
8	c	9.3%	10.3%	11%
9	c#	8.3%	9.3%	11%
10	.net	9.9%	8.4%	-15%
11	oracle	13.5%	8.4%	-38%
12	html	9.8%	8.1%	-17%
13	scrum	4.8%	8.0%	64%
14	git	3.1%	7.8%	148%
15	css	7.8%	7.3%	-5%
16	machine learning	1.3%	7.0%	439%
17	azure	0.6%	6.9%	1107%
18	unix	10.0%	6.7%	-33%
19	sql server	7.8%	6.5%	-17%
20	docker	0.1%	5.1%	4162%

Source: Indeed

indeed

Figure 3.4: docker-stats

This container can be transfered to another machine that runs Docker, and executed there without compatibility issues.

- *Version control and component reuse* : you can track successive versions of a container, inspect differences, or roll-back to previous versions. Containers reuse components from the preceding layers, which makes them noticeably lightweight. In addition due to Docker Hub it is possible to establish a connection between Git and DockerHub. Vesion
- *Sharing* : you can use a remote repository to share your container with others. It is also possible to configure a private repository hosted on Docker Hub.
- *Lightweight footprint and minimal overhead* : Docker images are typically very small, which facilitates rapid delivery and reduces the time to deploy new application containers.
- *Fault isolation* :Docker reduces effort and risk of problems with application dependencies. Docker also freezes the environment to the preferred packages version so that it guarantees continuity in deployment and isolate the container from system fails coming from package version updates.

The way to tell docker which system requirements are needed in the newly born software is a *Dockerfile*.

3.2.3 Dockerfile

Docker can build images automatically by reading instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands/rules a generic user could call on the CLI to assemble an image. Executing the command `docker build` from shell the user can trigger the image building. That executes sequentially several command-line instructions. For thesis purposes a dockerfile is written with the specific instructions and then the file is pushed to GitHub repository. Once pushed DockerHub automatically parses the repository looking for a plain text file whose name is “Dockerfile”. When It is matched then it trriggers the building of the image.

The Dockerfile used to trigger the building of the service docker container has the following set of instructions:

```
FROM rocker/tidyverse:latest

MAINTAINER Niccolo Salvini "niccolo.salvini27@gmail.com"

RUN apt-get update && apt-get install -y \
    libxml2-dev \
    libudunits2-dev

# install R packages
RUN R -e "install.packages(c('magrittr','lubridate', 'plumber', 'rvest', 'stringi', 'jsonlite', '

# install 'iterators' dep for DoParallel
RUN R -e "install.packages('https://cran.r-project.org/src/contrib/Archive/iterators/iterators_1.

# install 'foreach' dep for DoParallel
RUN R -e "install.packages('https://cran.r-project.org/src/contrib/Archive/foreach/foreach_1.4.8.

# install DoParallel from source since not avail in 4.0.2
RUN R -e "install.packages('https://cran.r-project.org/src/contrib/Archive/doParallel/doParallel_

COPY / /

# expose port
EXPOSE 8000

ENTRYPOINT ["Rscript", "main.R"]
```

Figure 3.5: dockerfile

- `FROM rocker/tidyverse:latest` : the command imports a pre-built image by the rocker team that contains the latest (tag latest) version of base-R along with the tidyverse packages.

- `MAINTAINER Niccolo Salvini "niccolo.salvini27@gmail.com"` : The command tags the maintainer and its e-mail contact information.

- `RUN apt-get update && apt-get install -y \ libxml2-dev \ libudunits2-dev` :The command update and install Linux dependencies needed for running R packages. `rvest` requires libxml2-dev and `magrittr` needs libudunits2-dev. If they are not installed then associated libraries can not be loaded. Linux dependencies needed have been found by trial and error while building containers. Building logs messages print errors and suggest which dependency is mandatory.

- `RUN R -e "install.packages(c('plumber','tibble','...',dependencies=TRUE)` : the command install all the packages required to execute the files (R files) containerized for the scraping. Since all the packages have their direct R dependencies the option `dependencies=TRUE` is needed.

- `RUN R -e "install.packages('https://cran.r-project.org/.../iterators, type='source')` `RUN R -e "install.packages('https://cran.r-project.org/.../` `type='source')` `RUN R -e "install.packages('https://cran.r-project.org/.../` `type='source')` DoParallel was not available in package manager for R version later than 4.0.0. For this reason the choice was to install a previous source version by the online repository, as well as its dependencies.

- `COPY \\` The command tells Docker copies all the files in the container.

- `EXPOSE 8000` : the commands instructs Docker that the container listens on the specified network ports 8000 at runtime. It is possible to specify whether the port exposed listens on UDP or TCP, the default is TCP (this part needs a previous set up previous installing, for further online documentation It is recommended (Inc., 2020a))

- `ENTRYPOINT ["Rscript", "main.R"]` : the command tells docker to execute the file main.R within the container that triggers the API start. In main.R it are pecified both the port and the host where API expects to be exposed (in this case port 8000).

In order to make the system stand-alone and make the service available to a wider range of subjects a choice has to be made. The service has to have both the characteristics to be run on demand and to specify query parameters.

3.3 REST API

(approfondisci REST) API stands for application programming interface and it is a set of definitions and protocols for building and integrating application software. APIs let a product or a service communicate with other products and services without having to know how they're implemented. This can simplify app development, saving time and impacting positively on the budget. APIs are sometimes thought of as contracts, with documentation that represents an agreement between parties. There are many types of API that exploit different medium to communicate with apps or services. HTTP API are a special type of API that accepts http requests as input and elaborate them through end points. An end point identifies the operation through traditional http methods (e.g. /GET /POST) that the API caller wants to perform. HTTP APIs have now become the predominant medium by which software exchanges information, further documentation and differences between REST and RESTful API can be found here[5].

API examples: - Google Maps API: allows developers to embed geo-location data using JavaScript. The Google Maps API is designed to work on mobile and desktop. - YouTube API: allows developers integrate YouTube videos and functionalities into websites or applications. - Google Analytics API: allows to track website performance in terms of audience, monetization and other important metrics through the Google Analytics interface. The website the thesis come from has this implementation working.

This is obtained by embedding the existing R source code into the Plumber API framework.

[5]https://docs.aws.amazon.com/it_it/apigateway/latest/developerguide/http-api-vs-rest.html

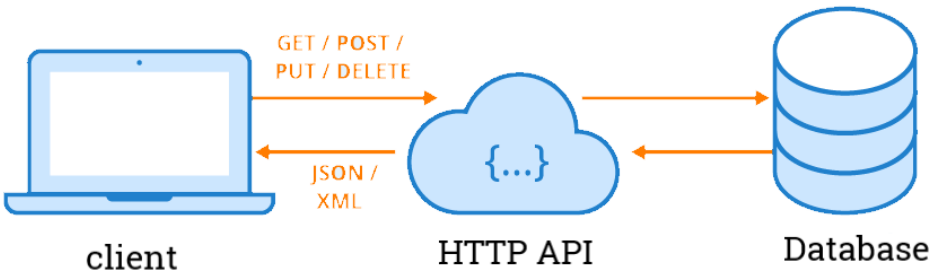


Figure 3.6: API functioning

3.3.1 Plumber API

Plumber (api, 2020) allows the user to create a web API by simply adding decoration comments to the existing R source code. Decorations are a special type of comments that suggests to plumber where and when the API parts are. Below the original API source code.

```
# plumber.R

# * Echo back the input * @param msg The message to echo * @get /echo
function(msg = "") {
    list(msg = paste0("The message is: '", msg, "'"))
}

# * Plot a histogram * @serializer png * @get /plot
function() {
```

```
    rand <- rnorm(100)
    hist(rand)
}

# * Return the sum of two numbers * @param a The first number to add * @param
# The second number to add * @post /sum
function(a, b) {
    as.numeric(a) + as.numeric(b)
}
```

three endpoints associated to 2 /GET and 1 /POST requests are made available. Functions are made clear without names so that whenever the endpoint is called functions are directly executed. Decorations are marked as this `#*` and they are followed by specific keywords denoted with `@`. - the `@params` keyword refers to parameter that specifies the corpus of the HTTP request, i.e. the inputs with respect to the expected output. If default parameters are inputted then the API response is the elaboration of the functions with default parameters. As opposite endpoint function elaborates the provided parameters and returns a response. - `#* @serializer` specifies the extension of the output file when needed. - `#* @get` specifies the method of HTTP request sent. - `/echo` is the end point name. - `@filter` decorations activates a filter layer which are used to track logs and to parse request before passing the argbody to the end points. - many more…

3.3.2 Immobiliare.it REST API

The API service is composed by three endpoints */scrape* , */links* and */complete*:

- */scrape performs a fast scraping that extracts 5 covariates directly from filtered url. url from which data extraction takes place might be composed through parameters. By default the end point scrape data from

Milan real estate rental market. Fast scraping is reached thanks to avoiding to access to single links. It is a superficial scraping and does not contain geospatial, however it might fit for regression settings.

- */links: extracts the list of single links belonging to each of the page, looking at section 2.1.1 each 25 single links for each sibling. It displays sufficient performances in terms of run time. It is propaedeutic to apply the following endpoint.
- */complete: both the function all.links and complete are sourced. The former with the aim to grab each single links and store it into an object. The latter to actually iterate scraping on each of the links.

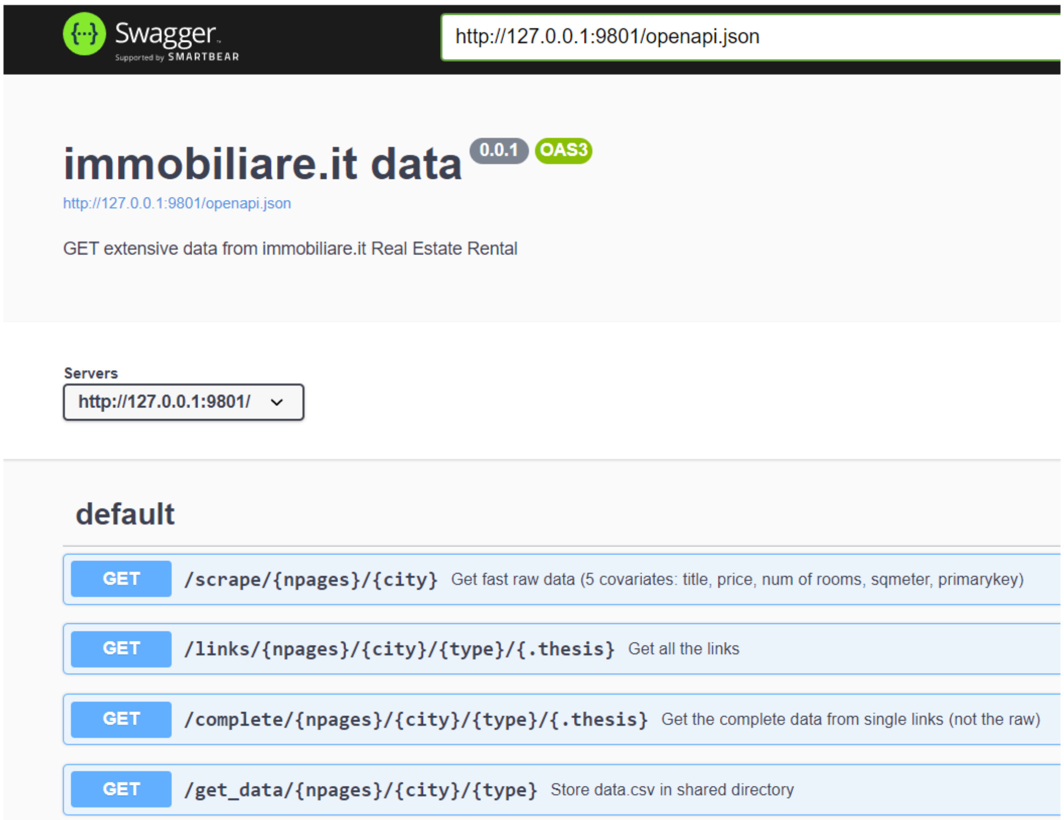


Figure 3.7: swagger

3.3.3 REST API source code

- Get FAST data, it covers 5 covariates: title, price, num of rooms, sqmeter, primarykey

```
# * Get all the links * @param city [chr string] the city you are interested
# extract data (lowercase without accent) * @param npages [positive integer]
# number of pages to scrape default = 10, min = 2, max = 300 * @param type [c
# string] affitto = rents, vendita = sell (vendita no available for now) * @g
# /links/<npages:int>/<city:chr>/<type:chr>/<.thesis:bool>
function(npages = 10, city = "milano", type = "affitto", .thesis = F, req) {
    cat("\n\n port:", req$SERVER_PORT, "\n server_name:", req$SERVER_NAME, "\n
    if (npages > 300 & npages > 0) {
        stop("npages must be between 1 and 1,000")
    }
    if (.thesis) {
        list(all.links(npages, city, type, .thesis = TRUE))
    } else {
        list(all.links(npages, city, type))
    }
}


# * Get the complete data from single links (not the raw) * @param city [chr
# string] the city you are interested to extract data (lowercase without acce
# * @param npages [positive integer] number of pages to scrape default = 10,
# = 2, max = 300 * @param type [chr string] affitto = rents, vendita = sell
# (vendita no available for now) * @get
# /complete/<npages:int>/<city:chr>/<type:chr>/<.thesis:bool>
function(npages = 10, city = "milano", type = "affitto", .thesis = F, req) {
    if (npages > 300 & npages > 0) {
        stop("npages must be between 1 and 1,000")
```

```
    }
    if (.thesis) {
        links = all.links(npages, city, type, .thesis = TRUE)
        list(complete(links))
    } else {
        links = all.links(npages, city, type)
        list(complete(links))
    }
}
```

3.3.4 REST API documentation

- Get FAST data, it covers 5 covariates: title, price, num of rooms, sqmeter, primarykey

```
      GET */scrape
      @param city [chr string] the city you are interested in (e.g. "roma", "m
      @param npages [positive integer] number of pages to scrape, default = 10
      @param type [chr string] "affitto" = rents, "vendita"  = sell
      @param macrozone [chr string] avail: Roma, Firenze, Milano, Torino; e.g.
      content-type: application/json
```

- Get all the links

```
      GET */link
      @param city [chr string] the city you are interested to extract data (lo
      @param npages [positive integer] number of pages to scrape default = 10,
      @param type [chr string] "affitto" = rents, "vendita"  = sell
      @param .thesis [logical] data used for master thesis
      content-type: application/json
```

- Get the complete set of covariates (52) from each single links, takes a while

```
GET */complete
@param city [chr string] the city you are interested to extract data (lo
@param npages [positive integer] number of pages to scrape default = 10,
@param type [chr string] "affitto" = rents, "vendita"  = sell
@param .thesis [logical] data used for master thesis
content-type: application/json
```

3.4 NGINX reverse proxy server

For analysis purposes NGINX is open source software for reverse proxying and load balancing. Proxying is typically used to distribute the load among several servers, seamlessly show content from different websites, or pass requests for processing to application servers over protocols other than HTTP. [...]

When NGINX proxies a request, it sends the request to a specified proxied server, fetches the response, and sends it back to the client. It is possible to proxy requests to an HTTP server (another NGINX server or any other server) or a non-HTTP server (which can run an application developed with a specific framework, such as PHP or Python) using a specified protocol. Supported protocols include FastCGI, uwsgi, SCGI, and memcached. [...]

.conf file and installation on Linux server. Security and Authentication.

3.5 AWS EC2 server

Executing REST API on a public server allows to share data with a various number of services thorugh multitude of subjects. Since it can not be specified a-priori how many times and users are going to enjoy the service a scalable solutio might fill the needs. Scalable infrastructure through a flexible cloud provider combined with nginx load balancing can offer a stable and reliable infrastructure for a relatively cheap price. AWS offers a wide range of services each of which for a wide range of budgets and integration. Free tier servers can be rent up to a certain amount of storage and computation that nearly 0s the total bill. The cloud provider also has a dedicated webpage to configure the service needed with respect to the usage named amazon cost manager[6]. Amazon Elastic Compute Cloud (EC2) is a web service that contributes to a secure, flexible computing capacity in the AWS cloud. EC2 allows to rent as many virtual servers as needed with customized capacity, security and storage. [few words still on EC2]

3.5.1 Launch an EC2 instance

The preliminary step is to pick up an AMI (Amazon Machine Image). AWS AMI are already-set-up machines with stadardized specification designed to speed up the process of choosing the a customed machine. Since the project is planned to be nearly 0-cost a "Free Tier Eligible" server is chosen. By checking the Free Tier box all the available free tiers are displayed. The machine selected has this specification: t2.micro with 1 CPU and 1GB RAM and runs on a Ubuntu distribution OS. First set up settings needs to be left as-is, networking and VPC can always be updated when needed. In the "add storage" step 30 GB storage are selected, moreover 30 represent the upper limit since the server can be considered free tier. Tags windows are beyond the scope. Secondly configuration needs to account security and a new entry below SSH connection (port 22) has to be set in. New security configuration has to have TCP specification and should be associated to port 8000. Port 8000, as in dockerfile section 3.2.3, has been exposed and needs to be linked to the security port opened.

At this point instance is prepared to run and in a few minutes is deployed. Key

[6] https://aws.amazon.com/en/aws-cost-management/

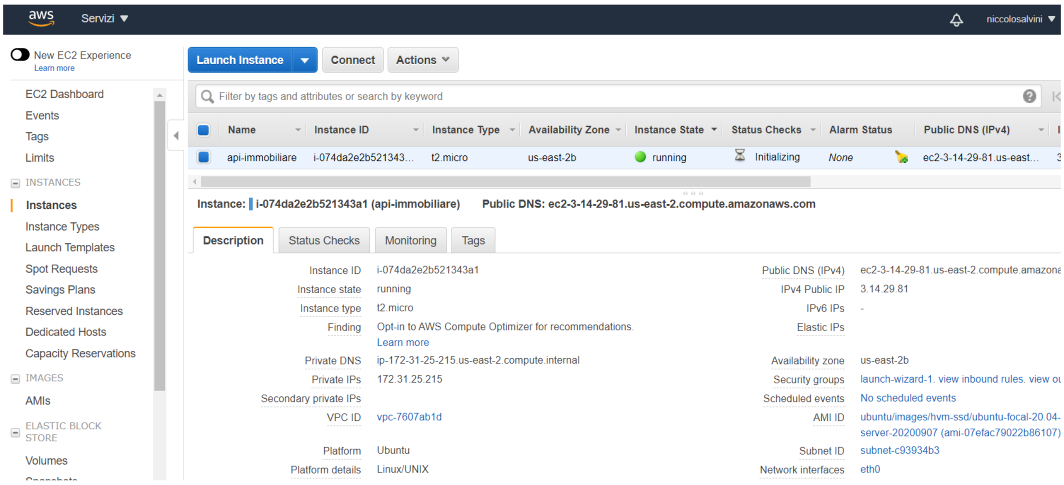


Figure 3.8: aws_dashboard

pairs, if never done before, are generated and .pem file is saved and securely stored. Key pairs are mandatory to access to the Ubuntu server via SSH. SSH connection in Windows OS can be handled with PuTTY[7], which is a SSH and telnet client designed for Windows. At first PuTTYgen has to convert the key pair .pem file into a .ppk extension (otherwise Putty can not read it). Once converted .ppk is sourced in the authorization panel. If everything works and authentication is verified then the Ubuntu server CLI appears and an interaction with the server is made available.

posso far vedere: - come si fa deplyment su AWS quindi - inizializzazione istanza - settare paramtri - mettere spazio macchina - spiegare perchè conviene tenere un server AWS

[7] https://www.putty.org/

Chapter 4

Exploratory Analysis

4.1 What data is

Data come packed from the API end point /complete in JSON format. A preliminary data assessment evidences 34 covariates and a 250 rows, even though the number could be varyign accoriding to the API query sent. Json file is then parsed by the convenient read_csv from the readr package mutating columns by their respective type. below the explanation of the columns:

ID: *numeric* a primary key number from immobiliare to identify univocally the advertisement LAT: *numeric* Exact latitude where the apartement is located (that compose the spatial covariate) LONG: *numeric* Exact longitude where the apartement is located (that compose the spatial covariate) CONDOM: *numeric* The condominium price which is an important component in the final cost structure BUILDAGE: *numeric* the year in which the building has been built FLOOR: *numeric* the floor at which is located the apartement/ house INDISVAP: *categorical* If it the rental ad regards an indipendent structure or an apartement in a building …

At a first glance it could be possible to see some missing values in some consistent measure in LOWPRCE and its derivations, Energy calss Disp Pauto.

```
## Warning: package 'readr' was built under R version 4.0.3
```

```
## Warning: package 'tibble' was built under R version 4.0.3
```

4.2 Data Glimpse

```r
library(knitr)
glimpse(data)
```

```
## Rows: 250
## Columns: 34
## $ X                        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
## $ ID                       <int> 82585523, 82710269, 80532997, 80880239, 82
## $ LAT                      <dbl> 45.4770, 45.4709, 45.4605, 45.4709, 45.470
## $ LONG                     <dbl> 9.15101, 9.20868, 9.18927, 9.20868, 9.2086
## $ LOCATION                 <chr> "piazzale arduino C.A.", "via antonio kram
## $ CONDOM                   <int> 310, 117, 417, 117, 133, 170, 333, 317, 35
## $ BUILDAGE                 <int> 1940, 1920, 1967, 1920, 1920, 1990, 1900,
## $ FLOOR                    <chr> "6°, con ascensore", "1° piano", "4°, con
## $ INDIVSAPT                <chr> "Appartamento", "Appartamento", "Appartame
## $ LOCALI                   <chr> "3 (2 camere da letto, 1 altro), 2 bagni,
## $ TPPROP                   <chr> "Residenziale", "Residenziale", "Residenzi
## $ STATUS                   <chr> "Nuovo / In costruzione", "Ottimo / Ristru
## $ HEATING                  <chr> "Centralizzato, a radiatori, alimentato a
## $ AC                       <chr> "Autonomo, freddo", NA, "Autonomo, freddo"
## $ PUB_DATE                 <chr> "2020-09-22", "2020-09-22", "2020-09-24",
## $ CATASTINFO               <chr> "Classe A/3, rendita \200 697", "Classe A/
## $ APTCHAR                  <chr> "- - fibra ottica- - - videocitofono- - -
## $ PHOTOSNUM                <int> 20, 20, 20, 20, 20, 20, 20, 20, 16, 20, 18
## $ AGE                      <chr> "Skyline RE Milano", "Skyline RE Milano",
## $ LOWRDPRICE.originalPrice <chr> NA, NA, NA, "\200 1.400", NA, NA, NA, NA,
## $ LOWRDPRICE.currentPrice  <chr> NA, NA, NA, "\200 1.300", NA, NA, NA, NA,
```

```
## $ LOWRDPRICE.passedDays    <int> NA, NA, NA, 8, NA, NA, NA, NA, NA, NA, NA,
## $ LOWRDPRICE.date          <int> NA, NA, NA, 8, NA, NA, NA, NA, NA, NA, NA,
## $ ENCLASS                  <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ CONTR                    <chr> "Affitto", "Affitto", "Affitto", "Affitto"
## $ DISP                     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ TOTPIANI                 <chr> "7 piani", "5 piani", "4 piani", "5 piani"
## $ PAUTO                    <chr> NA, NA, NA, NA, NA, NA, NA, NA, "1 in gara
## $ REVIEW                   <chr> "Rif: CITY LIFE CUBE - OK CONTRATTO SOCIET
## $ HASMULTI                 <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
## $ PRICE                    <int> 2450, 1200, 3000, 1300, 1700, 3850, 5000,
## $ SQFEET                   <int> 110, 82, 180, 82, 82, 157, 208, 160, 90, 4
## $ NROOM                    <int> 3, 2, 5, 2, 3, 5, 5, 4, 2, 2, 1, 3, 2, 3,
## $ TITLE                    <chr> "Trilocale piazzale ARDUINO, Milano", "Bil
```

We use the dataset extracted from the "PERVAL" database. This database contains all information on property transactions, property attributes as well as buyers and sellers profiles. Additionally, most observations in the database are geo-referenced and therefore spatial analysis is used. After removing the apartments with missing characteristics and eliminating the apartments corresponding to the tails11 of the price distribution, there are 7634 observations spanning from 2006 to 2017. The key variables of the dataset include the sale price, apartment structural characteristics and accessibility variables. The description of all variables is listed in Table 1. The summary statistics of all continuous variables are presented in Table

4.3 Explorative Analysis

4.3.1 Semivariogram Covariogram

Chapter 5

INLA computation

INLA (Rue et al., 2009) stands for Integrated Nested Laplace approximation and constitutes a computational alternative to traditional MCMC methods. INLA does approximate bayesian inference on special type of models called LGM (Latent Gaussian Models) due to the fact that they are *computationally* convenient. The benefits are many, some among the other are:

- Low computational costs, even for large models.
- It provides high accuracy.
- Can define very complex models within that framework.
- Most important statistical models are LGM.
- Very good support for spatial models.
- Implementation of spatio-temporal model enabled.

INLA uses a combination of analytics approximations and numerical integration to obtain an approximated posterior distribution of the parameters in a shorter time period. The chronologic steps in the methodology presentation follows the course sailed by Moraga (2019) blended with the author choice to skip details. As a matter of fact the aim of the chapter is to provide a comprehensive intuition oriented to the immediate application of the methodology, without stepping too long on mathematical details. By the way details e.g model assessment and control options are handled under the hood by the package and can be tuned within the main function, most of them are covered by Gómez Rubio (2020). Notation is imported from Marta Blangiardo (2015) and Gómez Rubio (2020), and quite differ from the one presented in the original paper by Rue, Chopin and Martino (2009). As further notation remarks: bold symbols are considered as vectors, so each time they occur they have to be considered like the *ensamble* of their values. In addition $\tilde{\pi}$ in section 5.2 are the Laplace approximation of the underlying integrals. Moreover the inner functioning of Laplace approximation and its special usage within the INLA setting is far from the scope, but an easy shortcut oriented to INLA is in Marta Blangiardo (2015).

INLA can fit only Latent Gaussian type of models and the following work tries to encapsulate its properties. Then afterwards a problem can be reshaped into the LGM framework with the explicit purpose to explore its benefits. When models are reduced to LGMs then joint posterior distribution can be rewritten and then approximated with INLA. A hierarchical bayesian structure on the model will help to integrate many parameter and hyperparameter levels and simplify interpretation. Generic Information on the project and the R-INLA package are contained in the introduction to last section 5.3. In the end a brief application on a toy spatial dataset is proposed with the aim to fasten the familiarity with the methodology and to come to grip with INLA results.

5.1 Latent Gaussian Models LGM

Given some observations $y_{i \ldots n}$ in order to define a Latent Gaussain Model within the bayesian framework it is convenient to specify at first an *exponential family* (Gaussian, Poisson, Exponential...) distribution function characterized by some parameters ϕ_i (usually expressed by the mean $E\left(y_i\right)$)) and some other hyper-parameters $\psi_k, \forall k \in \; 1 \ldots K$. The parameter ϕ_i can be defined as an additive *latent linear predictor* η_i, as pointed out by Krainski and Rubio ((2019)) through a link function $g(\cdot)$, i.e. $g\left(\phi_i\right)=\eta_i$. A comprehensive expression of the linear predictor takes into account all the possible effects on covariates

$$\eta_i = \beta_0 + \sum_{m=1}^{M} \beta_m x_{mi} + \sum_{l=1}^{L} f_l\left(z_{li}\right)$$

where β_0 is the intercept, $\beta = \{\beta_1, \ldots, \beta_M\}$ are the coefficient that quantifies the linear effects on covariates $x = (x_1, \ldots, x_M)$ and $f_l(\cdot), \forall l \in 1 \ldots L$ are a set of random effects defined in terms of a z set of covariates $z = (z_1, \ldots, z_L)$ (e.g. rw, ar1). As a consequence of the last assumption the class of LGM can receive a wide range of models e.g. GLM, GAM, GLMM, linear models and spatio-temporal models. This constitutes one of the main advantages of INLA, which can fit many different models, starting from simpler and ending with more complex. Contributors recently are extending the methodology to many areas as well as models moreover they are trying to incorporate INLA with non gaussian latent models as Rubio (2020) pointed out. All the latent components can be conveniently grouped into a variable denoted with θ such that: $\theta = \{\beta_0, \beta, f\}$ and the same can de done for hyper parameters $\psi = \{\psi_1, \ldots, \psi_K\}$. Then the probability distribution conditioned to parameters and hyper parameters is then:

$$y_i \mid \theta, \psi \sim \pi\left(y_i \mid \theta, \psi\right)$$

Since data $(y_1, \ldots, y_n)$ is drawn by the same distribution family but it is conditioned to parameters which are conditional independent (i.e. $\pi\left(\theta_i, \theta_j \mid \theta_{-i,j}\right) = \pi\left(\theta_i \mid \theta_{-i,j}\right) \pi\left(\theta_j \mid \theta_{-i,j}\right)$) (Rue and Held, 2005) then the joint distribution is given by the product of all the independent parameters i.e. the likelihood. Moreover the Product operator index i ranges from 1 to n, i.e. $\mathbf{I} = \{1 \ldots n\}$. When an observation is missing so the corresponding $i \notin \mathbf{I}$ INLA automatically will not include it in the model avoiding errors (2020). As a consequence the likelihood expression is:

$$\pi(y \mid \theta, \psi)=\prod_{i \in \mathbb{I}} \pi\left(y_{i} \mid \theta_{i}, \psi\right) \tag{5.1}$$

Each data point is connected to one combination θ_i out of all the possible linear combinations of elements in θ *latent field*. The latent aspect of the field regards the undergoing existence of many parameter combination alternatives. Furthermore hyper parameters are by definition independent, in other words ψ will be the product of many univariate priors (Gómez Rubio, 2020). A Multivariate Normal distribution is imposed on the latent field θ such that it is centered in 0 with precision matrix $Q(\psi)$ (the inverse of the covariance matrix $Q^{-1}(\psi)$) depending only on ψ hyper parameter vector i.e., $\theta \sim \operatorname{Normal}\left(\mathbf{0}, Q^{-1}(\psi)\right)$. As a notation remark some authors choose to keep the covariance matrix expression as Q and its inverse precision matrix as Q^{-1}. This is strongly not encouraged fro two reasons: the first is that the default hyperpramater option in INLA R package uses the precision matrix, the second it over complicates notation when writing down conditional expectation as Rue pointed out *miss lit*. However notation for covariance function introduced in chapter 6.2.1 i.e. Matérn has to be expressed through covariance matrix, this passage will be cleared out in the dedicated section so that confusion is avoided. The exponential family density function is then expressed through:

$$\pi(\theta \mid \psi)=(2 \pi)^{-n / 2}|Q(\psi)|^{1 / 2} \exp \left(-\frac{1}{2} \theta Q(\psi) \theta\right) \tag{5.2}$$

The conditional independence assumption on the latent field θ leads $Q(\psi)$ to be a sparse precision matrix since for a general pair of combinations θ_i and θ_j the resulting element in the precision matrix is 0 i.e. $\theta_{i} \perp \theta_{j} \mid \theta_{-i, j} \Longleftrightarrow Q_{i j}(\psi)=0$ (2015). A probability distribution function with those characteristics is said *Gaussian Markov random field* (**GMRF**). GMRF as a matter of fact are Gaussian variables with Markov properties which are encoded in the precision matrix Q (Rue et al., 2009). (puoi dire di più) From here it comes the source of run time computation saving, inherited using GMRF for inference. As a consequence of GMRF representation of the latent field, matrices are sparse so numerical methods can be exploited (Marta Blangiardo, 2015). *Moreover when Gaussian Process (see chapter 6.1), which are used to integrate spatial components, are represented as GMRF through SPDE (Stochastic Partial Differential Equations) approach, then INLA can be used as a computing choice. This last assumption will be framed in chapter* **??** *and verified in chapter 7.* Once priors distributions are specified for ψ then the joint posterior distribution for θ and ψ is

$$\pi(\theta, \psi \mid y) \propto \underbrace{\pi(\psi)}_{\text{prior}} \times \underbrace{\pi(\theta \mid \psi)}_{\text{GMRF}} \times \underbrace{\prod_{i=1}^{n} \pi\left(y_i \mid \theta_i, \psi\right)}_{\text{likelihood}}$$

Last expression is said a Latent Gaussian Models, **LGM**, if the whole set of assumptions imposed since now are met. Therefore all models that can be reduced to a LGM representation are able to host INLA methodology. Then plugging in the *likelihood* (5.1) and *GMRF* (5.2) expression the posterior distribution can be rewritten as

$$\begin{aligned}
\pi(\theta, \psi \mid y) &\propto \pi(\psi) \times \pi(\theta \mid \psi) \times \pi(y \mid \theta, \psi) \\
&\propto \pi(\psi) \times \pi(\theta \mid \psi) \times \prod_{i=1}^{n} \pi\left(y_i \mid \theta_i, \psi\right) \\
&\propto \pi(\psi) \times |Q(\psi)|^{1/2} \exp\left(-\frac{1}{2}\theta' Q(\psi)\theta\right) \times \prod_{i}^{n} \exp\left(\log\left(\pi\left(y_i \mid \theta_i, \psi\right)\right)\right)
\end{aligned}$$

And by joining exponents by their multiplicative property it is obtained

$$\pi(\theta, \psi \mid y) \propto \pi(\psi) \times |Q(\psi)|^{1/2} \exp\left(-\frac{1}{2}\theta' Q(\psi)\theta + \sum^{n} \log\left(\pi\left(y_i \mid \theta_i, \psi\right)\right)\right) \tag{5.3}$$

5.2 Approximation in INLA setting

INLA is not going to try to estimate the whole posterior distribution from expression (5.3). Instead it will try to estimate the posterior marginal distribution effects for each θ_i combination in the latent parameter θ, given the hyper parameter priors specification ψ_k. Proper estimation methods however are beyond the scope of the analysis, further excellent references are suggested in their respective part by Rubio (2020) in section 2.2.2 and Blangiardo & Cameletti (2015) in section 4.7.2. The marginal posterior distribution function for each latent parameter element θ_i is

$$\pi(\theta_i \mid y) = \int \pi(\theta, \psi \mid \mathbf{y}) \pi(\psi \mid \mathbf{y}) d\psi \tag{5.4}$$

The posterior marginal integral for each hyper parameter $ _{k}, k 1, ..., K $ is

$$\pi\left(\psi_k \mid y\right) = \int \pi(\psi \mid y) d\psi_{-k}$$

where the notation ψ_{-k} is a vector of hyper parameters ψ without considering kth element ψ_k.

The goal is to have approximated solution for latent parameter posterior distributions. To this purpose A *hierarchical procedure* is now imposed since the "lower" hyper parameter integral, whose approximation for the moment does not exist, is nested inside the "upper" parameter integral that takes hyper param as integrand. Hierarchical structures are welcomed very warmly since they are convenient later in order to fit a hierarchical bayesian model approached in the next chapter **??**). While many approximation strategies are provided and many others are emerging for both the hyper param and for the latent field, the common ground remains to unnest the structure in two steps such that:

- step 1: compute the Laplace approximation $\tilde{\pi}\left(\psi \mid y\right)$ for each hyper parameters marginal: $\tilde{\pi}\left(\psi_k \mid y\right)$

- step 2: compute Laplace approximation $\tilde{\pi}\left(\theta_i \mid \psi, y\right)$ marginals for the parameters given the hyper parameter approximation in step 1: $\tilde{\pi}\left(\theta_i \mid y\right) \approx \int \tilde{\pi}\left(\theta_i \mid \psi, y\right) \underbrace{\tilde{\pi}(\psi \mid y)}_{\text{Estim. in step 1}} \mathrm{d}\psi$

Then plugging approximation in the integral observed in (5.4) it is obtained:

$$\tilde{\pi}\left(\theta_i \mid y\right) \approx \int \tilde{\pi}\left(\theta_i \mid \psi, y\right) \tilde{\pi}(\psi \mid y) \mathrm{d}\psi$$

In the end INLA by its default approximation strategy thrpugh *simplified Laplace approximation* uses the following numerical approximation to compute marginals:

$$\tilde{\pi}\left(\theta_i \mid y\right) \approx \sum_j \tilde{\pi}\left(\theta_i \mid \psi^{(j)}, y\right) \tilde{\pi}\left(\psi^{(j)} \mid y\right) \Delta_j$$

where $\{\psi^{(j)}\}$ are a set of values of the hyper param ψ grid used for numerical integration, each of which associated to a specific weight Δ_j. The more the weight Δ_j is heavy the more the integration point is relevant. Details on how INLA finds those points is beyond the scope, but the strategy and grids seraches are offered in the appendix follwing both Rubio and Blangiardo.

5.2.1 further approximations (prolly do not note include)

INLA focus on this specific integration points by setting up a regular grid about the posterior mode of ψ with CCD (central composite design) centered in the mode (Gómez Rubio, 2020).

The approximation $\tilde{\pi}\left(\theta_i \mid y\right)$ can take different forms and be computed in different ways. Rue et al. (2009) also discuss how this approximation should be in order to reduce the numerical error (Krainski, 2019).

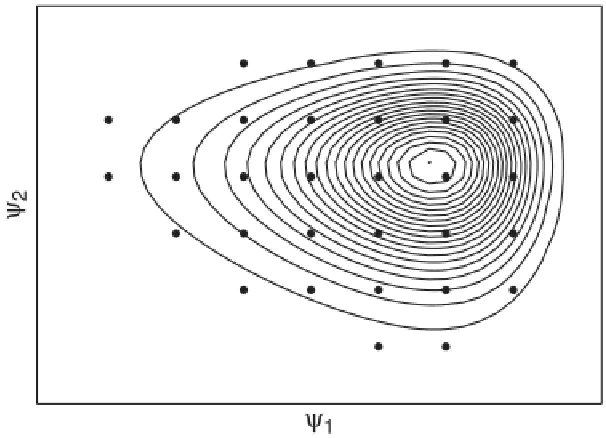


Figure 5.1: CCD to spdetoy dataset, source Marta Blangiardo (2015)

Following Gómez Rubio (2020), approximations of the joint posterior for the hyper paramer $\tilde{\pi}\left(\psi_k \mid y\right)$ is used to compute the marginals for the latent effects and hyper parameters in this way:

$$\tilde{\pi}(\psi \mid \mathbf{y}) \propto \left.\frac{\pi(\theta, \psi, y)}{\tilde{\pi}_G(\theta \mid \psi, y)}\right|_{\theta=\theta^*(\psi)}$$

In the previous equation $\tilde{\pi}_G(\theta \mid \psi, y)$ is a gaussian approximation to the full condition of the latent effect $\theta^*(\psi)$ is the mode for a given value of the hyper param vector ψ

At this point there exists three types of approximations for $\pi\left(\theta \mid \psi, y\right)$

- first with a gaussian approximation, estimating mean $\mu_i(\psi)$ and variance $\sigma_i^2(\psi)$.
- second using the *Laplace Approximation.*
- third using *simplified Laplace Approximation*

(rivedere meglio)

5.3 R-INLA package in a bayesian hierarchical regression perspective

5.3.1 Overview

INLA computations and methodology is developed by the R-INLA project whose package is available on their website[1]. Download is not on CRAN (the Comprehensive R Archive Network) so a special source repo link, which is maintained by authors and collaborators, has to be optioned. The website offers also a forum where a daily discussion group is opened and an active community is keen to answer. Moreover It also contains a number of reference books, among which some of them are fully open sourced as gitbook. Furthermore as Havaard Rue has pointed out in a web-lecture on the topic, the project is gaining importance due to its new applications and recent use cases, but by no means it is replacing the older MCMC methods, rather INLA can integrate pre existing procedures. The core function of the package is `inla()`and it works as many other regression functions like `glm()`, `lm()` or `gam()`. Inla function takes as arguments the formula (where are response and linear predictor), the data (expects a data.frame obj) on which estimation is desired together with the distribution of the data. Many other methods inside the function can be added through lists, such as `control.family` and `control.fixed` which let the analyst specifying priors distribution both for θ parameters, ψ hyper parameters and the outcome precision τ, default values are non-informative. `control.fixed` as said regulates prior specification through a plain list when there only a single fixed effect, instead it does it with nested lists when fixed effects are greater than 2, a guided example might better display the behaviour: `control.fixed = list(mean = list(a = 1, b = 2, default = 0))` In the chuck above it is assigned prior mean equal to 1 for fixed effect "a" and equal 2 for "b"; the rest of the prior means are set equal to 0. Inla objects are inla.dataframe summary-type lists containing

[1] http://www.r-inla.org

the results from model fitting. Results contained in the object are specified in the table below, even though some of them requires special method: (se riesco più elegante in kable) Following Krainski & Rubio (2019) observations $y(s_1), \ldots, y(s_n)$ are taken from a toy generated dataset and a hierarchical linear regression is fitted.

Function	Description
`summary.fixed`	Summary of fixed effects.
`marginals.fixed`	List of marginals of fixed effects.
`summary.random`	Summary of random effects.
`marginals.random`	List of marginals of random effects.
`summary.hyperpar`	Summary of hyperparameters.
`marginals.hyperpar`	List of marginals of the hyperparameters.
`mlik`	Marginal log-likelihood.
`summary.linear.predictor`	Summary of linear predictors.
`marginals.linear.predictor`	List of marginals of linear predictors.
`summary.fitted.values`	Summary of fitted values.
`marginals.fitted.values`	List of marginals of fitted values.

Figure 5.2: summary table list object, source: Krainski (2019)

5.3.2 Linear Predictor

SPDEtoy dataset, that has a spatial component, is generated from a y_i Gaussian variable; its moments are μ_i and precision τ.

The formula that describe the linear predictor has to be called directly inside the `inla()` function or it can be stored in the environment into a variable. The mean moment in the gaussian distribution μ_i is expressed as the *linear predictor* η_i (i.e. $E\left(y_i \mid \beta_0, \ldots, \beta_M, x_{i1}, \ldots, x_{iM}\right) = \eta_i$). The function that maps the linear predictor into the parameter space is identity as in section **??** i.e. $\eta_i = \beta_0 + \sum_{m=1}^{M} \beta_m x_{mi} + \sum_{l=1}^{L} f_l\left(z_{li}\right)$. After including s_1 and s_2 spatial covariates the linear predictor takes the following form: $\beta_0 + \beta_1 s_{1i} + \beta_2 s_{2i}$, where once again β_0 is the fixed effect i.e. intercept and the β_j are the linear effect on covariates. INLA allows also to include non-linear effects with the `f()` method inside the formula. `f` are foundamental since they are used to

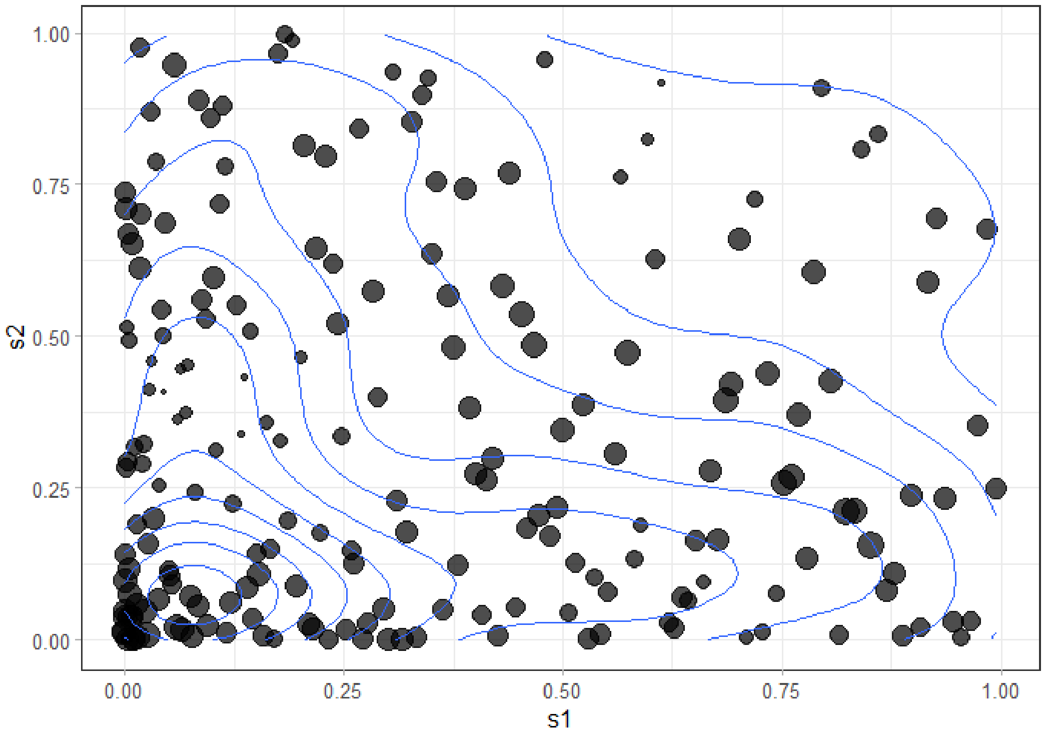


Figure 5.3: SPDEtoy plot, author's source

incorporate the spatial component in the model through the Matérn covariance function, this will be shown in section (boh). Once the formula is decided then priors has to be picked up; for the intercept a customary choice is uniform. Prior for Gaussian latent parameters are vague and they have 0 mean and 0.001 precision, then the prior for τ is a Gamma with parameters 1 and 0.00005. Prior initial choice can be later adapted.

The summary of the model parameters is:

$$y_i \sim N\left(\mu_i, \tau^{-1}\right), i = 1, \dots, 200$$

$$\mu_i = \beta_0 + \beta_1 s_{1i} + \beta_2 s_{2i}$$

$$\beta_0 \sim \text{ Uniform}$$

$$\beta_j \sim N\left(0, 0.001^{-1}\right), j = 1, 2$$

$$\tau \sim Ga(1, 0.00005)$$

```
## Warning: package 'sp' was built under R version 4.0.3
```

```
## Warning: package 'foreach' was built under R version 4.0.3

##
## Call:
##    "inla(formula = formula, data = SPDEtoy)"
## Time used:
##     Pre = 1.23, Running = 0.179, Post = 0.1, Total = 1.51
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) 10.132 0.242      9.656   10.132     10.608 10.132   0
## s1           0.762 0.429     -0.081    0.762      1.606  0.762   0
## s2          -1.584 0.429     -2.428   -1.584     -0.740 -1.584   0
##
## Model hyperparameters:
##                                          mean    sd 0.025quant 0.5quant
## Precision for the Gaussian observations 0.308 0.031      0.251    0.307
##                                         0.975quant  mode
## Precision for the Gaussian observations      0.372 0.305
##
## Expected number of effective parameters(stdev): 3.00(0.00)
## Number of equivalent replicates : 66.67
##
## Marginal log-Likelihood:  -423.18
```

The output offers among the others a summary of the posterior marginal values for intercept, coefficient and covariates, as well as precision. Below the plots for the parameters and hyperparameters. From the summary it can be seen that the mean for s2 is negative, so the more the value of the y-coordinates increases the more the output decreases, that is truer looking at the SPDEtoy cotour plot. Plots can be generated by calling the `plot` function on the inla object, however the one generated below are `ggplot2` outputs coming from the $marginals.fixed list object.

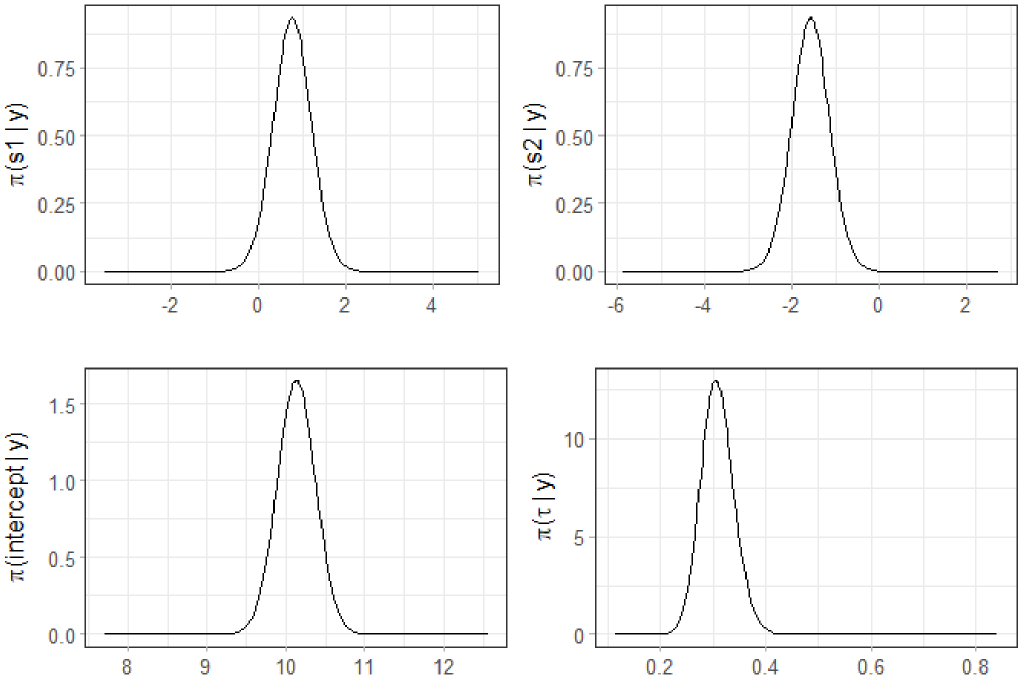


Figure 5.4: linear predictor marginals, author's creation

R-Inla also has r-base fashion function to compute statistics on marginal posterior distributions for the density, distribution as well as the quantile function respectively `inla.dmarginal`, `inla.pmarginal` and `inla.qmarginal`. One major option which is conveniently packed into a dedicated function computes the higher posterior density interval `inla.hpdmarginal` for a given covariate's coefficient, such that $\int_{q_1}^{q_2} \tilde{\pi}\left(\beta_2 \mid y\right) \mathrm{d}\beta_2 = 0.90$ with .1 Confidence Level. Recall that the interpretation is different from the frequentist: in Bayesian statistics β_j comes from probability distribution, while frequenstists considers β_j as fixed unknown quantity whose estimator (random variable conditioned to data) is used to infer the value (2015).

```r
## for s2 covariate
inla.hpdmarginal(0.9, m0$marginals.fixed$s2)
```

```
##                    low      high
## level:0.9 -2.291268 -0.879445
```

Chapter 6

Point Referenced Data Modeling

Geostatistical data are a collection of samples of geo type data indexed by coordinates (e.g. latlong, eastings and northings) that originate from a spatially continuous phenomenon (Moraga, 2019). Data as such can monitor a vast range of phenomena, as an example disease cancer detection (Bell et al., 2006) at several sites, COVID19 spread in China (Li et al., 2020), PM pollution concentration in a North-Italian region Piemonte (Cameletti et al., 2012). Moreover house prices variation, as observed in Gómez Rubio (2020), where selling prices smoothly vary between closer neighborhoods. All the Examples taken before might document a spatial nature of data according to which closer observations can display similar values, this phenomenon is named spatial autocorrelation. Spatial autocorrelation conceptually originates from geographer Waldo Tobler whose famous quote, known as first law of geography, inspires geostatisticians:

> "Everything is related to everything else, but near things are more related than distant things"
>
> — Waldo R. Tobler

Spatial models are explicitly designed to take into account this behavior and can separate spatial patterns from simply random spatial variance. Spatial data can be partitioned into three spatial data type whose modeling tools are specific with respect to their category.

- Areal Data
- **Point Referenced Data**
- Point Pattern Data

Figure 6.1: point referenced data example, Milan Rental Real Estate, Author's Source

REST API seen in chapter 3 extracts point referenced data, so modeling methodologies described in this analysis will exclusively take into account point referenced oriented techniques. In order to extend the notion from discrete measurements (i.e. point referenced) to a continuous spatial surface a stochastic process, namely Gaussian Process, has to be introduced and constrained according to convenient properties. GP are then evaluated with a specific covariance function, i.e. Matèrn. The reason why Matérn is selected as candidate for covariance function will be much more clear in the next chapter 7. Hedonic Price Models are at first introduced and then a brief literature review is offered. Hedonic Prices brings to this work the theoretical basis but they do not suggest estimation methods, which are essentially the major issue in geostatistics. For this reason Hedonic Models are exploited into a spatial bayesian regression framework with the aim to apply INLA (seen in chpater ??) methodology. At first standard Bayesian regression is presented as introduction, then the spatial component in the form of a GP is added to the model. Many parameters are considered so far, as a consequence a hierarchy structure is imposed. To this extent an entire section is dedicated to hierarchy which simplifies model building and methodology understanding as well as allowing to bring in many different parameters that come from different levels through the exchangeability property. As a matter of fact parameters originate from the Gaussian latent field, but also from Matérn covariance function tuning hyper parameters. Then INLA is applied and a GMRF representation of GP is... Spatial kriging is essential to predict the process at new locations so that the spatial surface can be plotted and analyzed. In the end models have to be checked and verified with resampling schemes which are once again specific to the data type and the scope of the analysis.

(forse mettere alla fine come further developments) As a side note Spatial data can also be measured according to a further dimension which is the Time. Latest literature suggests that spatio temporal models are the most accurate, as a consequence it might be interesting to research time correlation between subsequent spatial data time points, a valuable reference is offered in Paci et al. (2017). This will not take an enormous effort due ti the fact that on a daily basis REST API generates data which are stored as .json file on a DB. Future research on this data might consider the idea to include the time component in the model.

6.1 Gaussian Process (GP)

For simplicity lets consider y point of interest observations $y(s_1), y(s_2), \ldots, y(s_n)$ from a random spatial process Y, such that: $Y(s_1), Y(s_2), \ldots, Y(s_n)$ observed at location $s_1, \ldots, s_n$. In the context of geostatistical data each observation has to be considered as a partial realization of an unobserved random spatial process. $\{Y(s) : s \in D \subset \mathbb{R}^2\}$, where surface D is a subset of r-dimensional Euclidean space $\mathbb{R}^r$. Moreover When $r = 1$ it is the most simple stochastic process widely explored in literature i.e. time series process. However geostatistical data always have $r = 2$ (i.e. lat and long, eastings and northings) or eventually $r = 3$, when elevation data is available. The stochastic process Y is observed in a fixed set of "monitoring stations" and inference can be done regarding moments of the realized process. This information are essential to build a spatially continuous surface over the y-studied variable in order to predict the phenomenon at locations not yet observed.

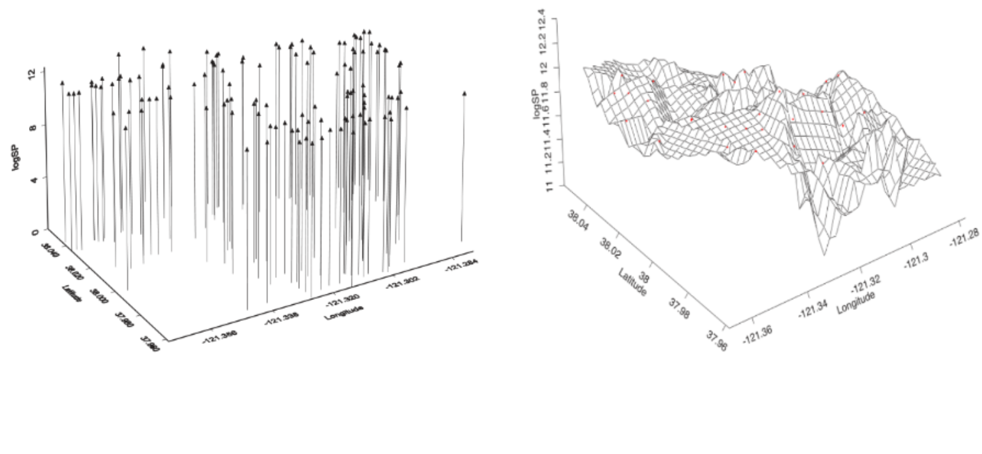


Figure 6.2: 3D scatterplot and surface, Stockton data.

A collection of n random variables, such as $Y(s_1), Y(s_2), \ldots, Y(s_n)$ that are *valid* spatial processes are said to be a **GP** if for any set of spatial index n and for each set of corresponding locations $\{y(s_1), \ldots, y(s_n)\}$ follows a multivariate *Gaussian* distribution with mean $\mu = \{\mu(s_1), \ldots, \mu(s_n)\}$ and covariance matrix $\mathbf{Q}^{-1}$, even tough sometimes it is more convenient to express it through precision matrix Q (Marta Blangiardo, 2015). The covariance matrix relates each observation to each of the others through a covariance function defined as $\mathcal{C}(\cdot)$.

GP in the spatial context must check two important properties in order to exploit INLA, even though both of these assumptions can be relaxed:

- **Stationary**.
- **Isotropy**.

Stationarity in a stochastic process can be *strong*, *weak* or *intrinsic*. The strong property forces the distribution of the process $\{y(s_1), \ldots, y(s_n)\}$ for any given spatial index n and its correspondent location sets $s_{1,\ldots,n}$ to be the same as the one in $\{y(s_1+h), \ldots, y(s_n+h)\}$, where h is a number belonging to R^2. On the other hand the weak property ensures that if the GP mean moment is constant over the study domain $\mu(\mathbf{s}) \equiv \mu$ (e.g. $E[Y(s)] = \mu, \forall s \in D$) then the covariance functions does depend only on the distance (euclidean $\|s_i - s_j\|$ distance) between each couple points. Weak stationarity consequences are the most interesting: It does not matter whether observations are placed either in a specific region, nor the direction towards they are oriented, the covariance functions $\mathcal{C}(h)$ can summarize the process through the separation vector $\mathbf{h}$ i.e. $\mathcal{C}(\mathbf{s}, \mathbf{s}+\mathbf{h}) = \mathcal{C}(\mathbf{h}), \forall \mathbf{h} \in \mathbb{R}^r$ (Banerjee et al., 2014). In other words weak stationarity in GP implies being invariant under *translation* (2019). The relationship between strong and weak is not bijective since being strong implies also being weak, but the opposite is not always true for non-Gaussian process. Furthermore through the intrinsic stationary property it is meant that $E[Y(\mathbf{s}+\mathbf{h}) - Y(\mathbf{s})] = 0$, the second moment of the latter expression can be written as $E[Y(\mathbf{s}+\mathbf{h}) - Y(\mathbf{s})]^2$ leading to $\mathrm{Var}(Y(\mathbf{s}+\mathbf{h}) - Y(\mathbf{s}))$. Last expression is called *variogram* and can be expressed with $2\gamma(\mathbf{h})$, even tough its half,i.e. $\gamma(\mathbf{h})$, is more interpretable, namely *semivariogram* (Cressie, 2015).

Semivariograms are characterized by mainly 3 tuning parameters:

- *range* σ^2: At some offset distance, the variogram values will stop changing and reach a sort of "plateau". The distance at which the effect occurs is called the range $\frac{\Delta\gamma(\mathrm{h})}{h} \approx 0$.
- *sill* τ^2: The "plateau" value at which the variogram stops changing $\frac{\Delta\gamma(\mathrm{h})}{h} = 0$.
- *nugget* $\tau^2 + \sigma^2$: The discontinuity at the origin. Although this theoretically should be zero, sampling error and short scale variability can cause it to be non-zero $\gamma(0)$.

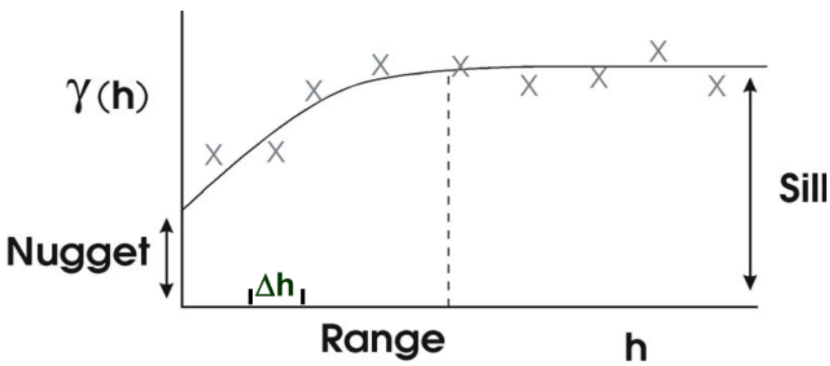


Figure 6.3: variogram example

presi i dati con le relative distanze euclidee a coppie di punti si binnano le distanze grazie ad un offset ottenendo i valori per il semivariogram. ottenuti i valori si fitta il semivargiogram a quei valori, un modo è la likelihood. A questo punto si calcolano le tre grandezze nugget sill e range per poi poter far uscire le funzioni di covarianza.

The process is said to be **Isotropic** if the covariance function depends only on the between-points distance $\|\mathbf{h}\|$ so it is invariant under *rotation* (2019). A further way of seeing the property is that Isotropy implies concentric decaying contours that resemble the vanishing of spatial dependence, and so covariance values too. then if the last assumption does not hold and direction towards point are distant from each other matters within the spatial domain D, then is said to be **Anisotropic**. Formalizing the results:

$$\mathcal{C}(\mathbf{h}) = \mathcal{C}(\|\mathbf{h}\|)$$

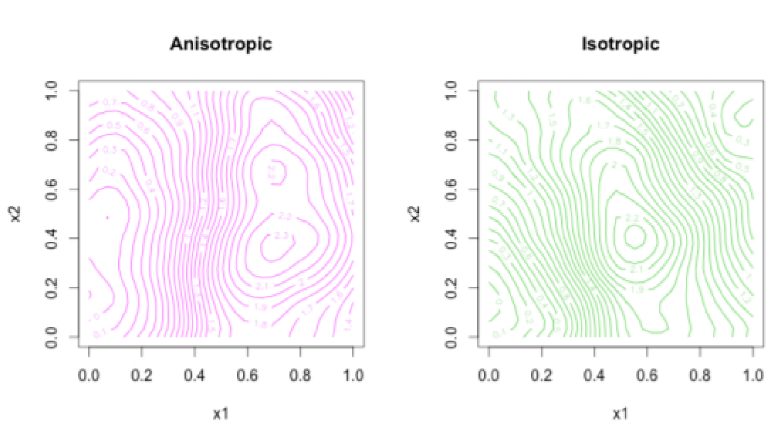


Figure 6.4: isotropy VS anisotropy, source Blanchet-Scalliet et al. (2019)

6.2 Spatial Covariance Function

The covariance function $\mathcal{C}(\cdot)$ ensures that all the values that are close together in input space will produce output values that are close together. $\mathcal{C}(\cdot)$ needs to inherits the *validity* characteristics from the random spatial process, furthermore it has to be *positive definite.* In addition covariance function must share characteristic properties of functions, such as:

(cerca di capire queste...)

- Multiply valid covariance functions (summing independent random variables)
- Mixing covariance functions (mixing distributions)
- Convolving covariance functions, this will be very important ...

Covariance functions under stationary and isotropic GPs displays two important properties: they are constant in mean within D i.e. $\mathcal{C}(\mathbf{s}, \mathbf{s}+\mathbf{h}) = \mathcal{C}(\mathbf{h}), \forall \mathbf{h} \in \mathbb{R}^r$ and they depends on distance vector $\mathbf{h}$, not direction i.e. $\mathcal{C}(\mathbf{h}) = \mathcal{C}(\|\mathbf{h}\|)$ There are many covariance functions and ways to relate distant points on a spatial domain D. Typically the choice of the Covariance can depend either on data or the scope of the analysis. Covariance functions are wrapped into special hyper parameters which are mainly three:

1. *Range*: At some offset distance, the variogram values will stop changing and reach a "plateau". The distance at which this occurs is called the range.
2. *Sill*: The "plateau" value at which the variogram stops changing.
3. *Nugget*: The discontinuity at the origin. Although this theoretically should be zero, sampling error and short scale variability can cause it to be non-zero

(espressione della covariance function insieme a alle σ^2 come: $C(\mathbf{s}+\mathbf{h}, \mathbf{s} \mid \theta) = \sigma^2 \mathbf{R}(\|h\|; \phi)$) spiega anche queste due sotto

$$\mathbf{w} = \left(w\left(\mathbf{s}_1\right), \ldots, w\left(\mathbf{s}_n\right)\right)' \sim \mathrm{N}\left(\mathbf{0}, \sigma^2 \mathbf{R}(\phi)\right) \text{ where } \mathbf{R}(\phi)_{ij} = \rho\left(\left\|\mathbf{s}_i - \mathbf{s}_j\right\|; \phi\right)$$

$\Sigma_\theta = \sigma^2 \mathbf{R}(\phi) + \tau^2 I_n$

A summary of the most used covariance functions are presented below.

$$\text{Exponential} \quad \mathcal{C}(\mathbf{h}) = \begin{cases} \tau^2 + \sigma^2 & \text{if } h = 0 \\ \sigma^2 \exp(-\phi h) & \text{if } h > 0 \end{cases}$$

$$\text{Gaussian} \quad \mathcal{C}(\mathbf{h}) = \begin{cases} \tau^2 + \sigma^2 & \text{if } h = 0 \\ \sigma^2 \exp\left(-\phi^2 h^2\right) & \text{if } h > 0 \end{cases}$$

$$\text{Matérn} \quad \mathcal{C}(\mathbf{h}) = \begin{cases} \tau^2 + \sigma^2 & \text{if } h = 0 \\ \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}(\phi h)^\nu K_\nu(\phi h) & \text{if } h > 0 \end{cases}$$

6.2.1 Matérn Covariance Function

Matérn is special since when it is used together with a stationary and isotropic GP, the SPDE approach can provide a GMRF representation of the same process, chapter 7 discloses this fundamental property. Matérn can also be accounted as the most used in geostatistics (Krainski et al., 2018) and (Gómez Rubio, 2020) and is tuned mainly by two parameters, a scaling one $\kappa > 0$, usually set equal to the range by the relation $\sigma^2 = \frac{\sqrt{8\lambda}}{\kappa}$) and a smoothing one $\nu > 0$. A *stationary* and *isotropic* Matérn covariance function has this form:

$$\mathcal{C}(\mathbf{h}) = \begin{cases} \tau^2 + \sigma^2 & \text{if } h = 0 \\ \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}(\phi t)^\nu K_\nu(\phi t) & \text{if } h > 0 \end{cases}$$

$\Gamma(\nu)$ is a Gamma function depending on ν values, $K_\nu(\cdot)$ is a modified Bessel function of second kind. The smoothness parameter ν in the figure below takes 4 different values showing the potentiality of Matérn to relates distances to covariance values. When $\nu = 1$ ... When $\nu = 1/2$ it becomes the exponential covariance function, When $\nu = 3/2$ it uncovers a convenient closed form, when $\nu \approx \infty$, in this case for representation purposes $\nu = 80$ it becomes Gaussian covariance function.

ancora di più su matern, forse di più in spde

6.3 Hedonic models Literature Review and Spatial Hedonic Price Models

The theoretical foundation of the Hedonic Price Models (from now on HPM) relies on the consumer utility theory of Lancaster (1966) together with Rosen (1974) market equilibrium. According to Lancaster the utility of a commodity does not exist by itself, instead it exists as the sum of the utilities associated to its separable characteristics. Integrating Lancater, Rosen introduces HPM

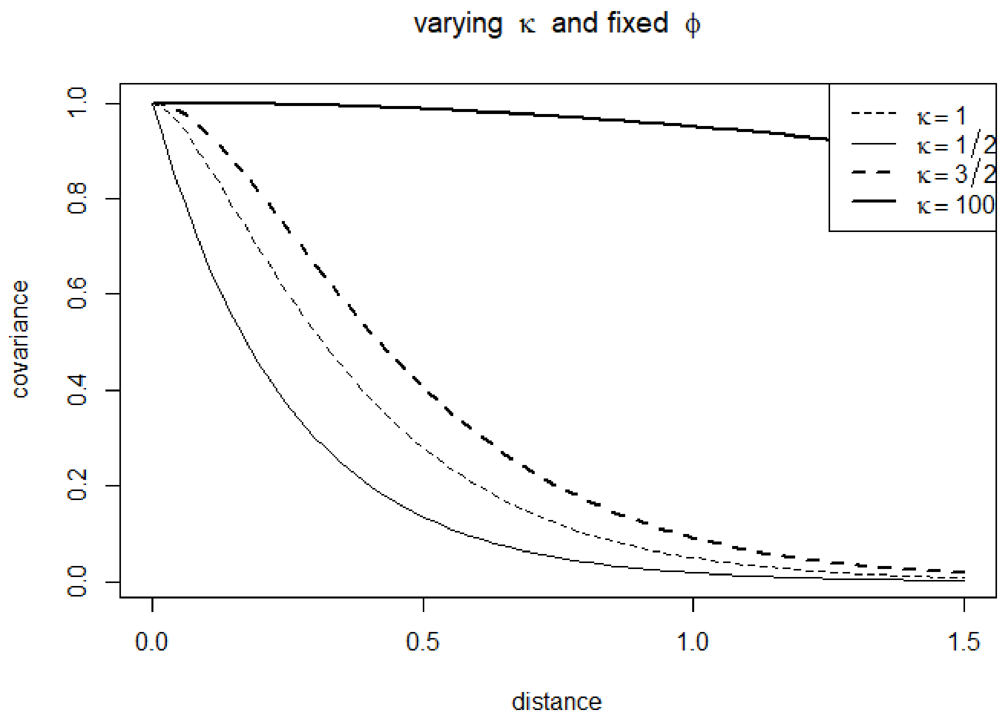


Figure 6.5: matern correlation function for 4 diff values of nu with phi fixed, author's source

and suggests that each separate commodity characteristics are priced by the markets on the basis of supply and demand equilibrium. Applying HPM to Real Estate in a market context, from the buy side house prices (but also renting) are set as the unit cost of each household attributes, conversely from the selling side the expenditures associated to build of each them. Formalizing the results, Hedonic Price P in Real Estate is expressed as a general f functional form that takes as input the house characteristics vector $\mathbf{C}$.

$$P = f\left(c_1, c_2, c_3, \ldots, c_n\right)$$

Vector $\mathbf{C}$ since now might contain a unidentified and presumably vast number of ungrouped characteristics. In this setting Malpezzi (2008) tried to organize house features by decomposing $\mathbf{C}$ into mutually exclusive and exhaustive subgroups. An overview of the vector components involved is given by Ling and Ling (2019) according to which P represents the house price, S is the structural characteristics of the house, N represents the neighborhood characteristics, L signifies the locational characteristics, C describes the contract conditions and T is time. β is the vector of the parameters to be estimated. Then

$$P = f\,(S, N, L, C, T, \beta)$$

A first attempt to include spatial effect in urban economic literature is provided by *Alonso (1964) miss ref.* Its contribution was to raise voice on house prices (also rent) mainly depending on land price and a number of purely spatial covariates like CBD, the distance from City Business District. Other covariates were transport cost per kilometer and community income, even though they were defined also as spatial parameters through distances. The model proposed by Alonso is called monocentric since the centroid from which distances are calculated is only one. Moreover a first touch to spatial data thory was done since the CBD was defined as areal unit with well-defined boundaries of regular or irregular shape. However applications of the model were not convincing since empirical studies offered a different picture. Results instead displayed a Poly-centric areal structure (universities and Malls) which might be better explaining prices. The model also assumed that covariates like CBD are only informative within city center boundaries and then displayed no significance out of the core of the city. Poly-centric theory was also more coherent with the architectural and socio-economical evolution of cities during that times, therefore mono centric theory was then criticized and abandoned. Critics regarded also neighborhood quality measure and boundary problems *Dubin (1987) miss ref.* Dubin for these reasons developed a geostatistical model including areal effects in the error term since handling these covariates was posing several hard challenges. Right here relies part of the reason why the problem in this analysis is approached as point reference and not as many other analysts do with areal data. [...] A change in focus has been made switching from theory based model to estimation methods. As a consequence to the change in focus practitioners should spend more time in variable selection and model specification with respect to their specific case. As Ling has observed the emerging trends are in the field of semi-parametric and non-parametric methods (2019). Historically semi-parametric regression considers models indexed by spatial coordinates *Pace RK (1995).* At the same time *Kammann and Wand (2003)* gave birth to geoadditive models where the spatial component is added as a covariate. [...]

A further aspect of the problem is posed by scholars that does not consider rents to be representative for the actual value of real estate. Nevertheless in empirical analysis rent value are considered a proxy for real estate pricing (Herath and Maier, 2011). A further argument to endorse this hypothesis is brought by Manganelli et al. (2013) considering housing a commodity, then the selling or the rental should be considered interchangeable economic actions with respect to same inner need to be satisfied. This is also truer to the thesis' extent since Manganelli, Morano, and Tajani have centered their analysis exactly on italian real estate data. Moreover Capozza and Seguin (1996) discussed on how much rent-price ratio predicts future changes both in rents and prices. Among all the other discussions raised they brought the decomposition of rent-price ratio into two parts: the predictable part and the unexplained residuals part. The predictable part was discovered to be negatively correlated with price changes, in other words cities in which prices are relatively high with respect to rents are associated with higher capital gains that might justify that misalignment. This is also true for the opposite, that is cities in which prices are lower with respect to the rents, and this effect can not be associated to any local condition, realize lower capital gains. A further argument is offered by Clark (Clark, 1995) which went after the Capozza and Seguin work. Rent-price ratio is negatively correlated with following future changes in rents. In other words prices are still higher when areas in which they are observed documents an increase in rent prices. All the literature review above is oriented to a long-run alignment of price and rent.

6.4 Point Referenced Regression for univariate spatial data

Since in HPM the relationships between the characteristics of the house, i.e. vector **C** and the price P is not in any case fixed by econometric literature it is possible to assume any f functional form. The open possibility to apply a wide range of relationship between covariates fit in the INLA setting, since Latent Gaussian Models are prepared to accept a any linear and non linear f functions 5.1 through the `f()` method. Hedonic price models are, as a consequence, a subset of models that can be fitted into LGM and therefore by INLA method.

Moreover what the vast majority of econometric literature *(Greene, 2018)* suggest to apply a is log-linear / square root model. This is due to the fact that log transformation / square root smooths the skewness of prices normalizing the curve, leading to more accurate estimates. Having an exponential family generating process lowers even further computational cost for reasons linked to the $\tilde{\pi}(\psi)$ hyper param INLA approximation (Marta Blangiardo, 2015). Notation is taken from the previous chapter **??**, for brevity purposes β $\mathbf{X}$ and y indicates vectors incorporating all their respective realizations and the s spatial component is left out in favor of the observation pedix i.
The simplest log linear bayesian regression model assumes linear relationship between predictors and a Normal data generating process:

$$\log\left(y_i\right) \sim \operatorname{Normal}\left(\mu_i, \sigma^2\right) y_i=\mu_i+\varepsilon_i$$

then by the following relationship $E\left(y_i \mid \beta_0, \ldots, \beta_M, x_{i1}, \ldots, x_{iM}\right) = \beta_0+\sum_{m=1}^M \beta_m x_{im}$ it is possible to specify a more general linear predictor (seen also in chapter **??**) through an identity link function i.e. $\eta_i=g\left(\mu_i\right)=\mu_i$ obtaining:

$$\eta_i = \beta_0 + \sum_{m=1}^{M} \beta_m x_{mi} + \sum_{l=1}^{L} f_l\left(z_{li}\right)$$

Where, once again, the mean structure linearly depends on some $\mathbf{X}$ covariates, β coefficients, $f_l(\cdot), \forall l \in 1 \ldots L$ are a set of random effects defined in terms of a z set of covariates $z = (z_1, \ldots, z_L)$ (e.g. rw, ar1) and ε_i white noise error. Priors have to be specified and a non informativeness for $\tau^2 = 1/\sigma^2$ and β is chosen, such that $\pi(\tau^2) \propto 1$ and $\pi(\beta) \propto 1$. As a consequence the conditional posterior for the parameters of interest β is:

$$\beta \mid \sigma^2, y, X \sim \text{MVNormal}\left(\left(X'X\right)^{-1} X'y, \sigma^2 \left(X'X\right)^{-1}\right)$$

where the mean structure corresponds to the OLS estimator: $\left(X'X\right)^{-1} X'y$.

In order to integrate the spatial component into the regression setting w_i has to be added to the equation. w_i is set as a stationary and isotropic GP with mean 0 and variance as covariance function expressed as Matérn.

$$\log(y_i) = \beta_0 + (\mathbf{X})'\beta + w_i + \varepsilon_i$$

The new regression setting integrates the *spatial error* part in the name of w_i and a *non-spatial error* part ε_i distributed normally with mean 0 and variance τ^2 ,i.e. $\mathrm{N}\left(0, \tau^2\right)$, which offers its contribution error to the nuggets through the covariance function.

6.5 Hierarchical Bayesian models

(prova a mettercix dentro direttamente la regressione appena sopra spiegata)

Spatial Models are characterized by many parameters which at their time might be tuned by some other hyper-parameters. Blangiardo e Cameletti (2015) tried to approach the problem from a different angle offering an intuitive solution on how hierarchy relates different levels parameters. This is done by reversing the problem and starting from data back to parameters, instead the other way round. So taking a few steps back the problem can be reformulated by starting from grouping observation into categories and then trying to impose a hierarchical structure on data based on the categories. As a result observations might fall into different categories, underlining their natural characteristics, such as: some of them might belong to category *levels* like males or females, married or not-married. Moreover diving into the specific problem house prices can be faceted by which floor they belong or whether they are assigned to different energy classes and many others more. As an example Blangiardo and Cameletti example consider grouping data according to just a single 9 *levels* category. Data for the reasons stated before can be organized such that each single observation (squares in figure below) belongs to its respective mutually exclusive and collectively exhaustive category (circles in figure).

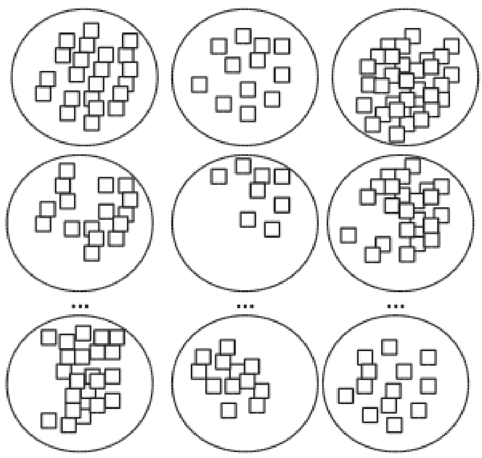
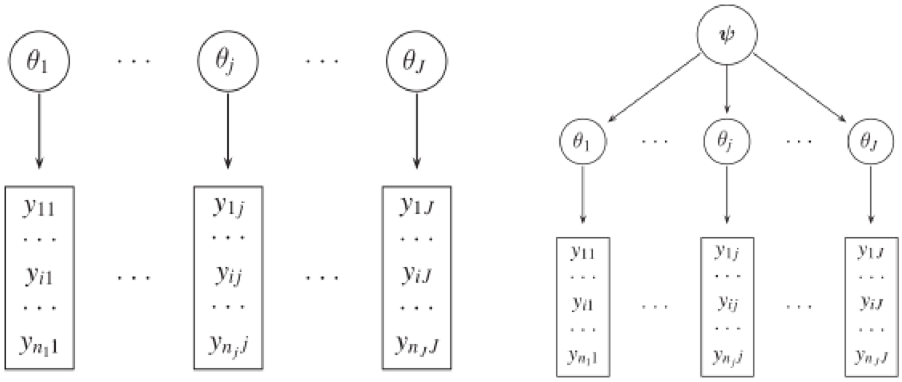


Figure 6.6: 9 levels cat vs observaitions, source Marta Blangiardo (2015)

Furthermore data can be partitioned into two meta-categories, *fist level* and *second level*, highlighting the parameter and hyper paramter chain roles. *First level* are identified by sampling observations which are drawn by the same probability distribution (squares) . *Second level* (circles) are categories and might be associated to a set of parameters $\theta = \{\theta_1, \dots, \theta_J\}$. Since the structure is hierarchical, a DAG (Directed Acyclical Graph) (2015) representation might sort out ideas. If categories are represented by different θ_j nodes and edges (arrows in the figure) are the logical belonging condition to the category then a single parameter θ model has the right figure form:



To fully take into account the hierarchical structure of the data the model should also consider further lelvels. Since $\{\theta_1, \ldots, \theta_J\}$ are assumed to come from the same distribution $\pi(\theta_j)$, then they are also assumed to be sharing information (Marta Blangiardo, 2015), (left figure). When a further parameter $\psi = \{\psi_1, \ldots, \psi_K\}$ is introduced, for which a prior distribution is specified, then the conditional distribution of θ given ψ is:

$$\pi\left(\theta_1, \ldots, \theta_J \mid \psi\right) = \int \prod_{j=1}^{J} \pi\left(\theta_j \mid \psi\right) \pi(\psi) \mathrm{d}\psi$$

This is possible thanks to the conditional independence property already encountered in chapter **??**, which means that each single θ is conditional independent given ψ This structure can extended to allow more than two levels of hierarchy since the marginal prior distributions of θ can be decomposed into the product of their conditional priors distributions given some hyper parameter ψ as well as their prior distribution $\pi(\psi)$.

$$\pi(\theta) = \int \pi\left(\theta \mid \psi_1\right) \pi\left(\psi_1 \mid \psi_2\right) \ldots \pi\left(\psi_{L-1} \mid \psi_L\right) \pi\left(\psi_L\right) \mathrm{d}\psi_1 \ldots \mathrm{d}\psi_L$$

ψ_l identifies the hyper pram for the l_{th} level of hierarchy. Each further parameter level ψ is conditioned to its previous in hierarchy level $l-1$ so that the parameter hierarchy chain is respected and all the linear combinations of parameters are carefully evaluated. The *Exchangeability* property enables to have higher H nested DAG (i.e. add further L levels) and to extend the dimensions in which the problem is evaluated, considering also time together with space. From a theoretical point of view there are no constraints to how many L levels can be included in the model, but as a drawback the more the model is nested the more it suffers in terms of interpretability and computational power. Empirical studies have suggest that three levels are the desired amount since they offer a good bias vs variance trade-off.

6.6 INLA model through hierarchical regression

INLA model seen in section 5.1 can be rearranged according to the hierarchical structure considering also the regression settings for point referenced data stated in the previous section 6.4.

As an initial step it is required to specify a probability distribution for $y = (y(s_1), \ldots, y(s_n)) = (y_1, \ldots, y_n)$, this is a mandatory step looking at the 5.3.2 methods needed to compute the `inla()` function. Normal distribution for simplicity is chosen.

As *first level* is picked up an exponential family sampling distribution (e.g. Normally distributed), which is *exchangeable* with respect to the $\theta = \{\beta_0, \beta, f\}$ *latent field* and hyper parameters ψ_1, which includes also the ones coming from the latent Matérn GP process w_i. The Spatial Guassian Process is centered in 0 and with Matérn covariance function as τ^2. w_i addresses the spatial autocorrelation between observation through a Matérn covariance function $\mathcal{C}(\cdot|\psi_1)$ which is tuned by hyper param included in ψ_1. Moreover the w_i surface has to be passed in the formula method definition 5.3.2 via the `f()` function, so that INLA takes into cosideration the spatial component. where $\mathbf{I}$ denotes the indicator function (i.e., $\mathbf{I}(i = i') = 1$ if $i = i'$, and 0 otherwise).

expressed as τ in the example in section 5.3.2 indicating the data generating process precision (i.e. $\tau = 1/\sigma^2$.

$$y \mid \theta, \psi_1 \sim \mathrm{N}\left(\beta_0 + (\mathbf{X}_i)'\beta + w_i, \tau^2 I_n\right) = \prod_{i=1}^{n} \mathrm{N}\left(y_i \mid \theta_i, \psi_1\right)$$

Then at the *second level* the latent field θ is characterized by a multivariate Normal distribution given the remaining hyper parameters ψ_2, recall that that $Q^{-1}(\psi_2)$ is the covariance matrix, depending on ψ_2 hyperparameters. INLA has to specify a Matérn covariace function in order to map the GP spatial surface into a GMRF by SPDE solutions.

$$\theta \mid \psi_2 \sim \mathrm{MVNormal}\left(0, Q^{-1}\left(\psi_2\right)\right)$$

In the end a *third level* hyper parameters $\psi = \{\psi_1, \psi_2\}$ having some specified prior distribution i.e. $\psi \sim \pi(\psi)$,

6.7 Spatial Kriging

6.8 Model Checking and Comparison

Chapter 7

SPDE approach

Observations in the spatial problem setting are considered as realizations of a stationary, isotropic unobserved GP $w(s)$ that we aim to estimate (6.1). Before approaching the problem with SPDE, GPs were treated as multivariate Gaussian densities and Cholesky factorizations were applied on the covariance matrices and then fitted with likelihood. Matrices in this settings were very dense and they were scaling with the order of $O\left(n^3\right)$, leading to obvious big-n problem. The breakthrough, came with Lindgren et al. (2011) that proves that a stationary, isotropic GP with Matérn covariance can be represented as a GMRF using SPDE solutions by finite element method (Krainski, 2019). In other words given a GP whose covariance matrix is Q, SPDE can provide a method to approximate Q without computational constraints. As a matter of fact SPDE are equations whose solutions are GPs with a chosen covariance function focused on satisfying the relationship SPDE specifies. Benefits are many but the most important is that the representation of the GP through a GMRF provides a sparse representation of the spatial effect through a sparse precision matrix Q^-1 . Sparse matrices enable convenient inner computation properties of GMRF that can be exploited with INLA. Bayesian inference on GMRF can take advantage of lower computational cost because of these properties stated before leading to a more feasible big-O $O\left(n^{3/2}\right)$. The following chapter will provide a intuition on SPDE oriented to practitioners.

The chapeter once again will follow the track of Krainski & Rubio (2019) and Blangiardo and Cameletti (2015) works, together with the street-opener paper from Miller et al. (2019) as compendium. SPDE might be complex for those who are not used to applied mathematics and physics making it difficult not only to grab the concept, but also to find its applications. One more obstacle regards SPDE software implementation, since without deep technical expertise it might be difficult to customize code with the aim to extend the methodology to different models. For a gentle introduction on what a SPDE is from a mathemathical perspective a valuable reference is Miller et al. (2019) in secction 2.1, then also its application to Matérn in 2.3.

7.1 Set SPDE Problem

Given the statistical model already encountered in chapter 6.4:

$$y\left(\mathbf{s}_{i}\right)=\mathbf{x}\left(\mathbf{s}_{i}\right)^{\prime} \beta_{j}+w(\mathbf{s})+\varepsilon\left(\mathbf{s}_{i}\right)$$

where $\eta(\mathbf{s}_i) = g(\mathbf{x}\left(\mathbf{s}_{i}\right)^{\prime} \beta_{j})$ is the linear predictor, whose link function $g(\cdot)$ is identity (can be also extended to GLM), where $w(\mathbf{s})$ is a Gaussian Process with mean structure 0 and $C(\cdot)$ covariance structure (where Q is the covariance matrix and Q^{-1} precision matrix). Then $w(s) \sim MV\mathcal{N}(0, Q_{i,j}^{-1})$ and where $\varepsilon(\mathbf{s}_i)$ is white noise error such that $\varepsilon(\mathbf{s}_i) \sim \mathcal{N}\left(0, \tau^{2}\right)$. Comprehending w in the model brings two major issues, specify a covariance function for observations as well as how to fit the model. Among all the possible reachable solutions including the SPDE, the common goal is to define covariance function between locations by approximating the precision matrix Q^{-1}, since they are an effective tool to represent covariance function as in section 5.1. For those reasons SPDE approach implies finding an SPDE whose solution have the precision matrix, that is desired for w. Lindgren et al. (2011) proves that an approximate solution to SPDE equations is to represent w as a sum of basis function multiplied by coefficients (2019). Moreover the basis function coefficients are in reality a GMRF (for which fast method comptations already exists).

7.2 SPDE within R-INLA

First point addresses the assumption that a GP with Matérn covariance function and $\nu > 0$ is a solution to *SPDE* equations. Second point addressed the issues of solving SPDE when grids are irregular, as opposite with the one seen in first point (regular grid for irregular distribution. In here comes FEM used in mathematics and engineering application with the purpose to solve differential equations. Notation is kept coherent with the one for the previous chapter.

7.3 First Point Krainsky Rubio TOO TECHNICAL

A regular 2D grid lattice is considered with infinite number of location points as vertices.

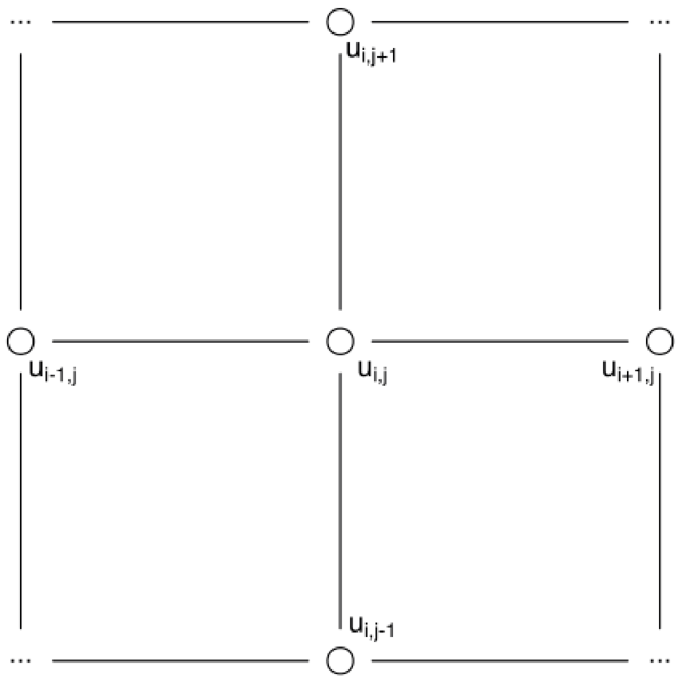


Figure 7.1: lattice 2D regular grid

Chapter 8

Shiny Web App

Some *significant* applications are demonstrated in this chapter.

8.1 Example one

8.2 Example two

Chapter 9

Final Words

We have finished a nice book.

Bibliography

(2020). An api generator for r.

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., and Iannone, R. (2020). *rmarkdown: Dynamic Documents for R*. R package version 2.3.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014). *Hierarchical Modeling and Analysis for Spatial Data*. Chapman and Hall/CRC.

Bell, B. S., Hoskins, R. E., Pickle, L., and Wartenberg, D. (2006). *International Journal of Health Geographics*, 5(1):49.

Blanchet-Scalliet, C., Helbert, C., Ribaud, M., and Vial, C. (2019). Four algorithms to construct a sparse kriging kernel for dimensionality reduction. *Computational Statistics*, pages 1–21.

Cameletti, M., Lindgren, F., Simpson, D., and Rue, H. (2012). Spatio-temporal modeling of particulate matter concentration through the SPDE approach. *AStA Advances in Statistical Analysis*, 97(2):109–131.

Capozza, D. and Seguin, P. (1996). Expectations, efficiency, and euphoria in the housing market. *Regional Science and Urban Economics*, 26:369–386.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2020). *shiny: Web Application Framework for R*. R package version 1.4.0.2.

Clark, T. E. (1995). Rents and prices of housing across areas of the United States. A cross-section examination of the present value model. *Regional Science and Urban Economics*, 25(2):237–247.

Cressie, N. (2015). *Statistics for Spatial Data*. Probability and Mathematical Statistics. Wiley-Interscience, revised edition edition.

Densmore, J. (2019). Ethics in web scraping.

Diestel, R. (2006). *Graph theory*. Graduate Texts in Mathematics. Springer, 3rd edition.

Google (2020). Presentazione dei file robots.txt - guida di search console.

Gómez Rubio, V. (2020). *Bayesian Inference with INLA*. Chapman and Hall/CRC.

Herath, S. and Maier, G. (2011). Hedonic house prices in the presence of spatial and temporal dynamics. *Territorio Italia - Land Administration Cadastre, Real Estate*, 1:39–49.

Inc., D. (2020a). Get docker.

Inc., R. H. (2020b). 7.2. advantages of using docker red hat enterprise linux 7.

Krainski, E., Gómez-Rubio, V., Bakka, H., Lenzi, A., Castro-Camilo, D., Simpson, D., Lindgren, F., and Rue, H. (2018). *Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA*. Chapman and Hall/CRC.

Krainski, E. T. (2019). *Advanced spatial modeling with stochastic partial differential equations using R and INLA*. Chapman and Hall/CRC.

Lancaster, K. J. (1966). A new approach to consumer theory. *Journal of Political Economy*, 74(2):132–157.

Li, H., Li, H., Ding, Z., Hu, Z., Chen, F., Wang, K., Peng, Z., and Shen, H. (2020). Spatial statistical analysis of coronavirus disease 2019 (covid-19) in china. *Geospatial Health*, 15(1).

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498.

Ling, Y. and Ling, Y. (2019). Time, space and hedonic prediction accuracy evidence from the corsican apartment market. *The Annals of Regional Science*, page 28.

Malpezzi, S. (2008). *Hedonic Pricing Models: A Selective and Applied Review*, pages 67–89.

Manganelli, B., Morano, P., and Tajani, F. (2013). Economic relationships between selling and rental prices in the italian housing market.

Marta Blangiardo, M. C. (2015). *Spatial and Spatio-temporal Bayesian Models with R-INLA*. Wiley.

Meissner, P. and Ren, K. (2020). *robotstxt: A 'robots.txt' Parser and 'Webbot'/'Spider'/'Crawler' Permissions Checker*. R package version 0.7.7.

Microsoft and Weston, S. (2020). *foreach: Provides Foreach Looping Construct*. R package version 1.5.0.

Miller, D. L., Glennie, R., and Seaton, A. E. (2019). Understanding the stochastic partial differential equation approach to smoothing. *Journal of Agricultural, Biological and Environmental Statistics*, 25(1):1–16.

Moraga, P. (2019). *Geospatial Health Data*. Chapman and Hall/CRC.

Nolis, J. (2020). R docker faster.

Paci, L., Beamonte, M. A., Gelfand, A. E., Gargallo, P., and Salvador, M. (2017). Analysis of residential property sales using space–time point patterns. *Spatial Statistics*, 21:149 – 165.

Perepolkin, D. (2019). *polite: Be Nice on the Web*. R package version 0.1.1.

Rosen, S. (1974). Hedonic prices and implicit markets: Product differentiation in pure competition. *Journal of Political Economy*, 82(1):34–55.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields*. Chapman and Hall/CRC.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392.

Trestle Technology, LLC (2018). *plumber: An API Generator for R*. R package version 0.4.6.

Vaughan, D. and Dancho, M. (2018). *furrr: Apply Mapping Functions in Parallel using Futures*. R package version 0.1.0.

was previously an Economist at the Indeed Hiring Lab, A. F. F. (2020). Indeed tech skills explorer: Today's top tech skills.

WhoIsHostingThis.com (2020). User agent: Learn your web browsers user agent now.

Wickham, H. (2019). *rvest: Easily Harvest (Scrape) Web Pages*. R package version 0.3.5.

Wikipedia contributors (2020). Cron — Wikipedia, the free encyclopedia. [Online; accessed 15-September-2020].

Wikiversità (2020). Scheduling — wikiversità,. [Online; accesso il 15-settembre-2020].

Xie, Y. (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-1138700109.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.20.