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]

Last compiled on 10 novembre, 2020

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

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Trento:

- Argomento
- Problema
- Obiettivi
- Metodo
- Struttura della tesi

Main themes:

- Research Question
- Milan Real Estate Controversies in relation to research question
- why the API (perchè mi mancano i dati e perchè è il futuro)
- Open Data discussion personal hope of data sharing and benefits from oper source
- Why a Bayesian approach
- Why INLA

As a general discussion technologies implied can be thought as the distance between a service running locally on a laptop and something that it can actually be put into production, shared among company stakeholders, solving

business related problems. When such technologies are applied data scientist and interlocutors gradually close the gap. Insights are better communicated, data is up-to-date and automation can save time. Nonetheless when the infrastructure is structured 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.

Chapter 2

Scraping

The following chapter covers a gentle introduction and concepts of web scraping centered on immobiliare.it. It starts in general terms by ideally partitioning websites in two entities: website structure and content architecture, then the segmentation is applied to the specific immobiliare.it case. The abstract workflow will ease the identification of the first "high level" challenges to front, so that the following "lower levels" can be then accessed and solved. Website structure mainly accounts how urls are arranged throughout the website and how to reverse engineer the website url composition. Once the structure is unrolled in a way that each part can be singularly and directly accessed, then the flow passes to "lower levels" i.e. scraping content architecture. At this point a rooted-tree graph representation is used to map scraping functions design into immobiliare content architecture. By means of `rvest` scraping is possible and the main function involved are presented. Then a specific function to scrape price (the response var) is shown and then also a second function, that takes care of grouping all the scraping functions into a single one. Scraping best practices are applied both on the web server side, kindly requesting permission and delayed sending rate; and from the web client side by granting continuous scraping, avoiding server blocks by User Agent rotation and trycatches / handlers for easy debugging. Then run time scraping performances are fronted presenting two different options for looping construction,

`furrr` and `foreach`. The latter has displayed interesting results and further improvements are taken into consideration. Then an overview of the open challenges is offered so that this work might be extended and integrated with newer technologies or paradigms. In the end legal profiles are addressed comparing scraping results and difficulties with a counterpart case study.

2.1 What is Web Scraping

Definition 2.1 (Scraping)**.** Web Scraping is a technique aimed at extracting data from static or dynamic internet web pages. It can be applied simultaneously or automatically by a scheduler that plans execution at a given time.

Content in web pages is the most of the times well organized and accessible. 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 different web pages, are arranged throughout the website. Website structure constitutes a *first dimension* of hierarchy. A comparative example might regard the way files are arranged into folders, where in the example files are urls and folders are web pages. Some popular website structure examples are social-networks where posts can be scrolled down within a single url web page named the wall. The scrolling option might end due to the end of the feed, but the perception is a never-ending web page associated to a single url. Indeed personal profiles have dedicated unique url and personal photos or friends are allocated into specific personal profile sub-domains. Online newspapers display their articles in the front page of their websites and by accessing to one of them all the pertinent articles sometimes can be reached in the low bottom or in the side part. Suggested articles during website exploration can be seen twice, that is the more the website is covered the more is the likelihood of seeing the same article twice. Recursive websited structures are popular in newspaper-type websites since it is part of the expectation on website's user experience. Online Retailers as Amazon, based on search filters, groups inside a single web

page (i.e. page n° 1) a fixed set of items, having their dedicated single urls attached to them. Furthermore many of the retailers, Amazon's too, allows the user to search into many pages i.e. page n° 2,3, looking for another fixed set of items. The experience ends when the last page associated to the last url is met. These are few examples of how websites can be built and infinitely many others are available. Generally speaking website structures try to reflect both the user expectations on the product and the creative design expression of the web developer. This is also constrained to the building languages needed and the specific requirements of the that the website should met. For all the reason said, for each product, whether it is physical product, or service, or information there exists a multitude of website structure. For each website structure there exists multiple content architecture. For each content architecture there exists many front end languages which are ultimately designated to satisfy multiple end users. In the future chances are that websites will display tailor made website structure and contents based on specific user personal preferences. As a further addition web design in scraping plays an important role since the more are implied sophisticated graphical technologies, the harder will be scraping information.

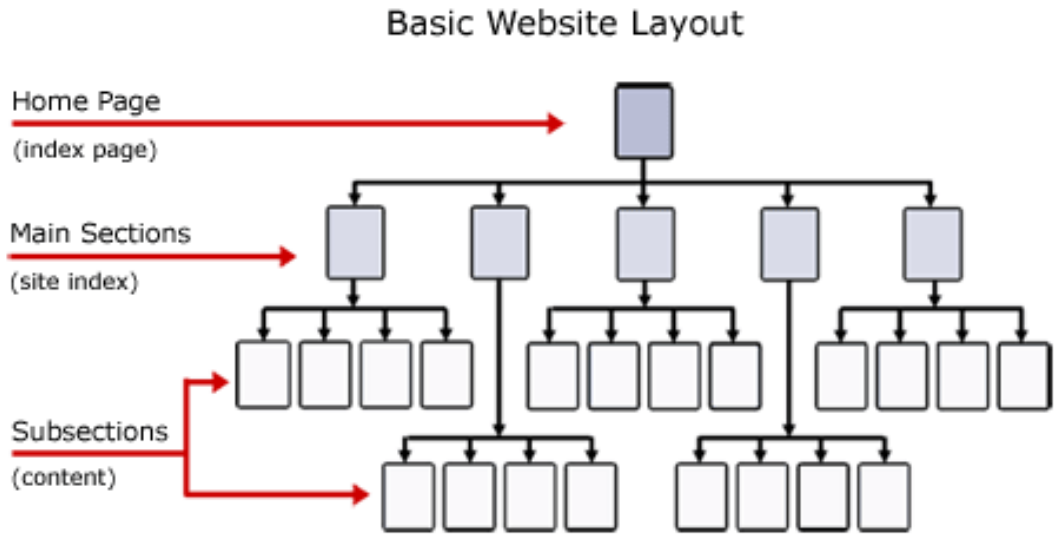


Figure 2.1: general website structure

A *second dimension* of hierarchy is brought by content architecture by means

of the language used for content creation and organization i.e. HTML. HTML stands for Hyper Text Markup Language and is the standard *markup* language for documents designed to be showed into a web browser. It can be supported by technologies such as Cascading Style Sheets (CSS) and other scripting languages, as an example JavaScript (htm, 2020). HTML inner language properties brings along the hierarchy that is then inherited from the website structure. According to this point of view the hierarchical website structure is a consequence of the language chosen for building content architecture. Since a hierarchy structure is present a direction must be chosen, this direction is from root to leaves i.e. *arborescence*. CSS language stands for Cascading Style Sheets and is a style sheet language used for modifying the appearance of a document written in a *markup* language(css, 2020). The combination of HTML and CSS offers a wide flexibility in building web sites, once again expressed by the vast amount of different websites designs on the web. Some websites' components also might be tuned by a scripting language as Javascript. JavaScript enables interactive web pages and the vast majority of websites use it for all the operations that are performed by the client in a client-server relationship (Jav, 2020). In the context of scraping Javascript adds a further layer of difficulty. As a matter of fact Javascript components are dynamic and scraping requires specialized libraries or remote web browser automation ((Harrison, 2020) R Bindings for Selenium 2.0 Remote WebDriver) to catch the website content. CSS instead 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 the same style) can be gathered. This practice provides tremendous advantages since by a single CSS query a precise set of objects can be obtained within a unique function call. First and Second dimension of the scraping problem imply hierarchy. 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 setting into tree graph jargon, as a further reference on notation and wordings can be found in

Diestel (2006). Rooted trees must start with a root node which in this context is the domain of the web page. Each *Node* is a url destination and *Edges* are the connections to web pages. Jumps from one page to the others (i.e. connections) are possible in the website by nesting urls inside webpages so that within a single webpage the user can access to a limited number of other links. Each edge is associated to a *Weight* whose interpretation is the run time cost to walk from one node to its connected others (i.e. from a url to the other). In addition the content inside each node takes the name of payload, which is ultimately the goal of the scraping processes. The walk from node "body" to node "h2" in figure below 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 edges, in the figure below "headre" is the parent of nodes "h1" and "p". Nodes in the tree that shares the same parent node are said *Siblings*, "h1" and "p" are siblings in figure @ref(fig:html_tree). Moreover *Subtrees* are a set of nodes and edges comprised of a parent and its descendants e.g. node "main" with all of its descendants might constitute a subtree. The concept of subtree in both of the problem dimensions plays crucial role in cutting run time scraping processes as well as 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 siblings subtrees. 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 a standard shared-within-webpages criteria according to which each single rental advertisement has to refer (e.g. each new advertisement replicates the structure of the existing ones). In addition two more metrics 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** , e.g. "head" and "body", are at the same

level. The height is the maximum level for any node in the tree, from now on **H**, in figure @ref(fig:html_tree). What is worth to be anticipating is that functions are not going to be applied directly to siblings in the "upper" general rooted tree (i.e. from the domain). Instead the approach follwed is segmenting the highest tree into a sequence of single children unit that shares the same level ("nav", "main", "header", "title" and "footer") for reasons explained in section 2.3.1.

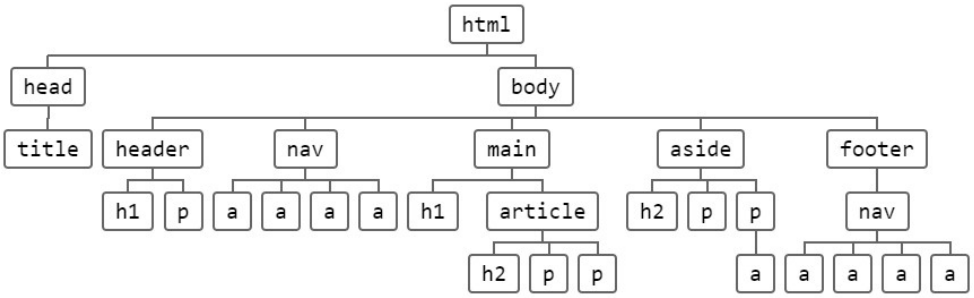


Figure 2.2: (#fig:html_tree) html tree structure of a general website, randomly generated online

2.1.1 Immobiliare.it Webscraping website structure

The structure of immobiliare website can be assumed to be similar to the one of the largest online retailer Amazon. For that reason they both fall in the same website structure category. Sharing the same website structure category might imply that the transition from customized website structure scraping functions (i.e. immobiliare) do not take too many sophistications to be extended to other comparables (i.e. Amazon). Assuming that the scraper knows where are critical information (i.e. payloads), what it takes to uncover the website structure is a way to decompose it. As a matter of fact each time the scraper script visits the website it should not to start back then from root node and down through the path straight to the bottom of the content. Indeed it should minimize the number of nodes encountered constrained to the respective weights. This is reached by engineering url composition. According to some filters selected in a

section (e.g. city, number of rooms 5, square footage less than 60 m^2 etc), the url is shaped so that each further filter is appended at the end of the domain url `https://www.immobiliare.it/` root node. Filters are appended with a proper syntax, not all the composition syntax are equal, that is why scraping script needs sophostication to be transalted from immobilaire to other websites in the same category. Once filters are all applied to the root domain this constitutes the new url root domain node that might have this appearance by filtering for rents in Milan city when square footage is less than 60 m^2: domain `+affitto-case/milano/?superficieMinima=60`. Since this is true only for page n°1 containing the first 25 advs (see figure 2.3) all the remaining siblings nodes corresponding to the subsequent pages have to be generated. Here the utility of Local reproducibility property introduced in the previous section. The remaining siblings, i.e. the ones belonging to page 2 (with the attached 25 links),to page 3 etc. can be generated by adding a further filter `&pag=n`, where n is the page number reference (from now on referred as *pagination*). Author customary choice is to stop pagination up to 300 pages since spatial data can not be to too large due to computational constraints imposed by inla methodology 4. Up to this point pagination has generated a vector of siblings nodes whose children elements number is fixed (i.e. 25 links per page 2.3 lower part). That makes those trees *k-ary*, where k is 25 indicating the number of children leaves. K-ary trees are rooted trees in which each node has no more than k children, in this particular case final leaves. The well known binary rooted tree is actually a special case of k-ary when $k = 2$. Filters reverse engineering process and 25-ary trees with equal content structure across siblings allow to design a single function to call that could be mapped for all the other siblings. In addition in order to further unroll the website a specific scraping function grabs the whole set of 25 links per page. As a result a single function call of `scrape_href()` can grab the links corresponding to page 1. Then the function is mapped for all the generated siblings nodes (i.e. up to 300) obtaining a collection of all links belonging to the set of pages. Ultimately the complete set of links corresponds to every single advertisement posted on immobiliare.it

at a given time.

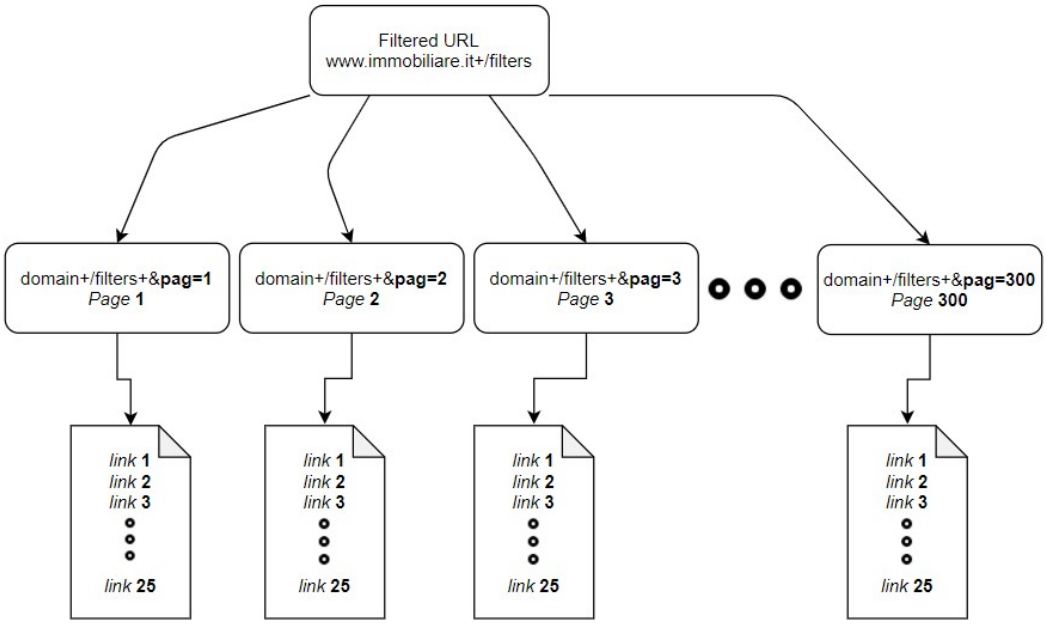


Figure 2.3: immobiliare.it website structure, author's source

2.1.2 Immobiliare.it Webscraping content architecture with `rvest`

To start a general scraping function the only requirements are a target url (i.e. the filtered root node url) and a way to compose url (i.e. pagination). Then a session class object `html_session` is opened by specifying the url and the request data that the user needs to send to the web server, see left part to dashed line in figure 2.4. 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` class objects contains a list number of useful data such as: the url, the response, cookies, session times etc. Once the connection is established (response request 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 already provided request data. The list object contains the xml/html content response of the webpage

and that is where data needs to be parsed and class converted. The list can disclose as well other interesting meta information related to the session but in this context are not collected. The light blue wavy line follows the steps required to get the content parsed from the beginning to the end.

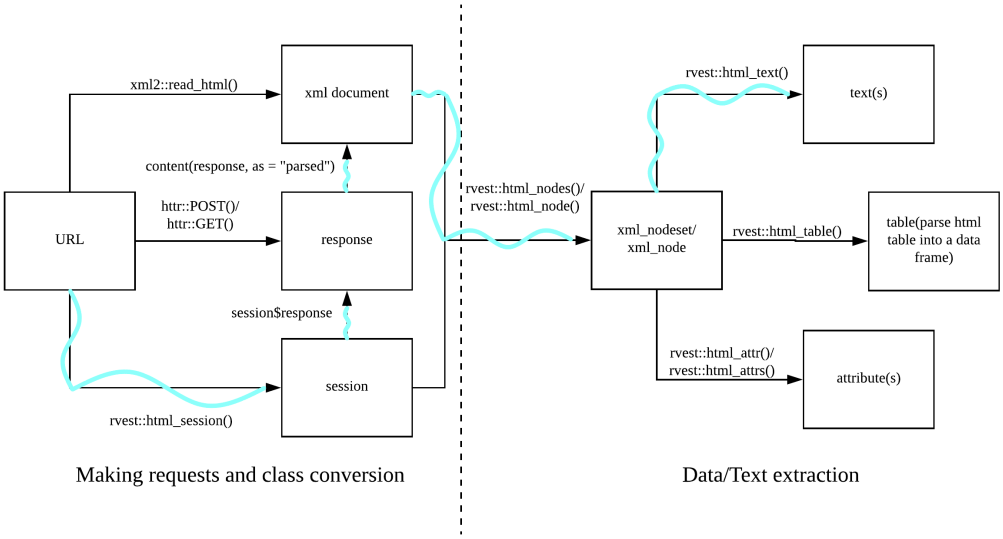


Figure 2.4: rvest general flow chart, author's source

To the right of dashed line in the flow chart 2.4 are painted a sequence of `rvest`(Wickham, 2019) functions that follow a general step by step text comprehension rules. `rvest` first handles parsing the html respose content of the web page within the session through `read_html()`. Secondly, as in figure 2.5, it looks for a single node `html_nodes()` through a specified CSS query. CSS is a way to route `rvest` to consider a precise node or set of nodes in the web page. For each information contained in each of the web page a different CSS query has to be called. Thirdly it converts the content (i.e. payload) into a human readable text with `html_text()`. A simplified version of the important contents to be scraped in each single link is sketched in figure 2.5

The code chunk below exemplifies a function that can scrape the price. The function explicitly covers only the right part to the dashed line (figure 2.4) of the whole scraping process. The initial part (left dashed in same figure), where session is opened and response is converted is handles inside the second code

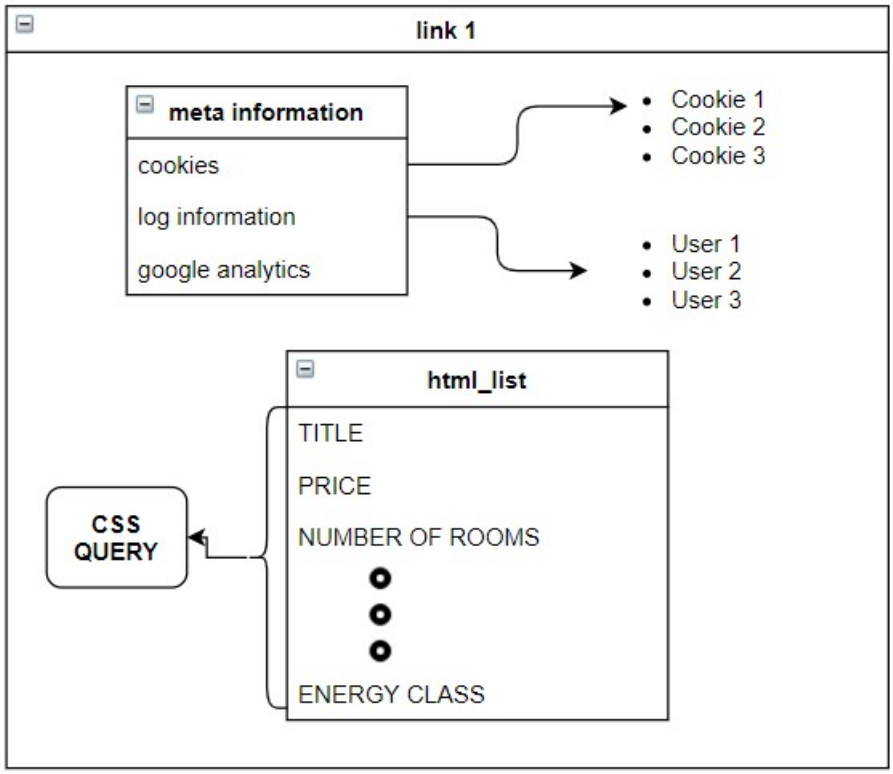


Figure 2.5: immobiliare.it important content structure, author's source

chunk where `get.data.catsing()` is.

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

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

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

        if ("prezzo" %in% price2) {
            pos = match("prezzo", price2)
            return(price2[pos + 1]) %>% str_replace_all(c(`\200` = "", `\\.` =
```

```
                str_extract("\\-*\\d+\\.*\\d*") %>% str_replace_na() %>% str_r
                "Prezzo Su Richiesta")
        } else {
            return(NA_character_)
        }
    } else {
        return(price) %>% str_replace_all(c(`\200` = "", `\\.` = "")) %>% str_
            str_replace_na() %>% str_replace("NA", "Prezzo Su Richiesta")

    }
}
```

The function takes as a single argument a session object, 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 group of data inside the web page, right below the main title. Expectation are that price is a one-element chr vector, containing the price and other unnecessary non-UTF 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 function and returns the quantity cleaned. But If it does it takes again the `opensess` and tries with a second css query `.im-features__value , .im-features__title` in a second data location where price might be also found. Please note that this is all done within the same session, so no more additional request information has to be sent. Since the latter CSS query points to data stored inside a list object, for the time being the newly created `price2` is a list containing various information. Then the algorithm flow enters into the second `if` statement that checks whether `"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. [ti-

tle, “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 *NA-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 object 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 process
  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 e
                  }

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

                  id          = tryCatch({scrapehouse.ID(session)},
                                         error = function(e){ message("some p
                  lat         = tryCatch({scrapelat.imm(session)},
                                         error = function(e){ message("some p
                  long        = tryCatch({scrapelong.imm(session)},
                                         error = function(e){ message("some p
                  location    = tryCatch({take.address(session)},
                                         error = function(e){ message("some p
```

```
                    condom     = tryCatch({scrapecondom.imm(session)},
                                          error = function(e){ message("some p
                    buildage   = tryCatch({scrapeagebuild.imm(session)},
                                          error = function(e){ message("some p

                  ...

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

                  ...


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

The skeleton used for price constitutes a standard format adopted for many other scraping function 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 of that functions need also to be broken

down by pieces by 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 noticed that the two folders, namely functions_singolourl and functions_url enclose all the single functions that allows to grab each information from the 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
31 ¦   ¦--ScrapeStatus.R
32 ¦   ¦--ScrapeTotPiani.R
33 ¦   ¦--ScrapeType.R
34 ¦   °--take_location.R
35 ¦--scrapeALL.R
36 ¦--scrapeALLINFO.R
37 ¦--functions_url
38 ¦   ¦--ScrapeHREF.R
39 ¦   ¦--ScrapePrice.R
40 ¦   ¦--ScrapePrimaryKey.R
41 ¦   ¦--ScrapeRooms.R
42 ¦   ¦--ScrapeSpace.R
43 ¦   °--ScrapeTitle.R
44 ¦--libs.R
45 ¦--utils.R
46 ¦--README.Rmd
47 ¦--README.md
48 °--simulations
49     ¦--rt_match_vs_forloop.R
50     °--runtime_simul.R
```

2.2 Scraping Best Practices and Security provisions

Robots.txt files are (rivedi citation) a way to kindly ask webbots, spiders, crawlers, wanderers 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 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 field-name value scheme with optional preceding user-agent. 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 references on robotstxt are recommended (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 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 assess

whether the domain or the related paths require specific actions or they prevent some activity on the target. The following `checkpermission()` function has been integrated inside the scraping system and it is called once at the starting point.

```
dominio = "immobiliare.it"

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()
    }
}
## metti path allowed check
checkpermission(dominio)
```

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

Further improvements in this direction might come with the `polite` package (Perepolkin, 2019) which combines the power of the `robotstxt`, the `ratelimitr` (Shah, 2018) to limit sequential requests together with the `memoise` (Wickham et al., 2017) 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. These have to be intended as best practices, not in any case as law enforcements, however many scrapers themselves, as website administrators or analyst, have fought in their daily working tasks with bots. Crawling bots in intensive scraping processes might fake real client navigation logs and as a consequence might induce distorted analytics. Due to this fact comes the need to find a common operating ground and therefore politely asking for permission.

2.3 Security provisions: User Agents, Proxies and 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 browser (in the example above A) is interacting with a server website (part B) sending packets of information, figure 2.6. 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.

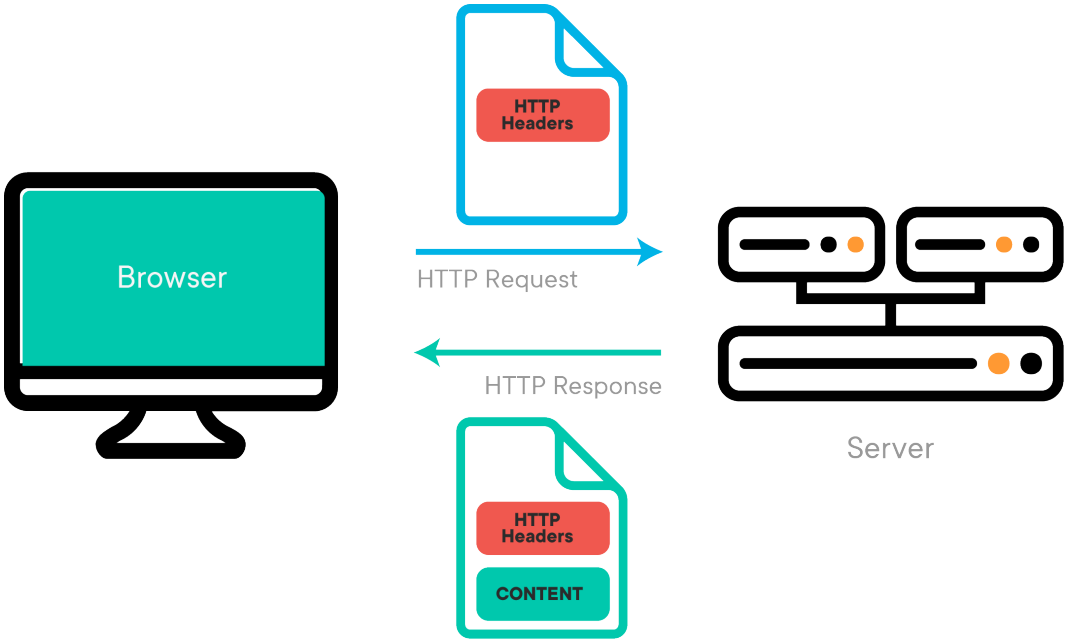


Figure 2.6: How the web interacts between broswer and server

This procedure is a daily issue to scraper 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. Precautions have not been taken lightly, and a simple but effective approach is proposed.

2.3.1 User Agents Spoofing

Definition 2.2 (User Agents)**.** A user agent (WhoIsHostingThis.com, 2020) is a string of characters in each web browser that serves as identification card. 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 includes the user application or software, the operating system (and their versions), the web client, the web client's version, as well as the web engine responsible for the content display (such as AppleWebKit). The user agent string is sent in the form of a HTTP request header. Since User Agents acts as middle man between the client request and the server response, then from a continuous scraping point of view it would be better rotating them, so that each time the middle man looks different. The solution adopted builds a vector of user agent strings identified by different specifications, different web client, different operating system and so on, then samples 1 of them Then whenever a request from a web browser is sent to a web server, 1 random sample string is drawn from the user agents pool. So each time the user is sending the request it appears to be a different User Agent. Below the user agents rotation pool:

```
set.seed(27)
agents = c("Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
    "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/602.2.14 (KHTM
    "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
    "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like G
    "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 Ge
```

A more secure approach might be a further rotation of proxies between the back and forth sending-receving process. A proxy server acts as a gateway

between the web user and the web server. While the user is exploiting a proxy server, internet traffic flows through the proxy server on its way to the server requested. The request then comes back through that same proxy server and then the proxy server forwards the data received from the website back to the client. The final result will be linear combination of User Agents ID and Proxy server for each sending requests, grating a high security level. Many proxy servers are offered in a paid version, so in this case since security barriers are not that high they will not be implemented. As a further disclaimer many online services are providing free proxies server access, but this comes at a personal security cost due to a couple of reasons: - Free plan Proxies are shared among a number of different clients, so as long as someone has used them in the past for illegal purposes the client is indirectly inheriting their legal infringements. - Very cheap proxies, for sure all of the ones free, have the activity redirected on their servers monitored, profiling in some cases a user privacy violation issue.

2.3.2 Handlers and Trycatches

During scraping many difficulties are met. Some of them might come from website structure issues, so that rooted-tree hierarchies are changed as a consequence of a restructuring. Some others might interest content architecture where data is reallocated to some other places in the webpage, as a consequence CSS query are no more able to catch data. Handlers in the form of trycatch error workarounds are explicitly built in this sense. The continuous building and testing of the scraping functioning has required the maintainer to have a precise and fast debugging experience. The following consideration might give a sense of the time consumed when debugging handlers are not implied. `get.data.catsing()` triggers 34 different scrapping functions that are supposed to point to 34 different data pieces. Within a single function call by default pagination generates 10 pages each of which contains at least 25 different single urls to be scrapped. That leads to a number of 8500 single data

information. The probability given 8500 associated to something going lost or unparsed is undoubtedly high. The solution proposed tries to handle fails by implementing as many trycatches as scrapping functions inside the single `get.data.catsing()` call. Then also inside each single scrapping function are put the handlers. This set up allows to catch (and in some cases prevent) fails starting from the very end of the scraping process. As a consequence of this setting when a scrapping function is not able to gather data an error inside the function is thrown, then the error call is intercepted by the corresponding trycatch, which at the end flags where the error occurs. Below some of the main handlers implied:

- `.get_ua()` verifies that the User Agent in the session 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
    grepl(re, url)
}
```

- `.get_delay()` checks through the robotxt file if a delay between each request is kindly welcomed. When response is NA delay is not required.

```
.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) ## so that .get_delay results are c
.get_delay(domain = dominio)
```

```
## [1] NA
```

2.4 Parallel Computing

Since are opened as many sessions as single links and since for each link are supposed to be called 34 functions run time computation can take a while. Run time is crucial when dealing with very active web pages and time to market in real estate is a major issue. From there originates the need to have always up-to-date data and consequently an API infrastructure. Run time optimization involves each level of the scraping process from the "lowest" i.e. inside each single function to the "highest" i.e. the agglomerative function. Inside single scraping functions as a general criteria for loops are avoided due to Rcpp reasons, vectorization is preferred. Within agglomerative function instead the

approach was to test two different results. All the following runtime examinations are performed on the `scrape()` functions which is a lightweight version of the function included in the API. The first attempt was using `furrr` package (Vaughan and Dancho, 2018) which enables mapping through a list with the `purrr`, along with a `future` parallel back end. The approach has shown decent performance, 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 with steep slope. Empirical demonstrations have been made:

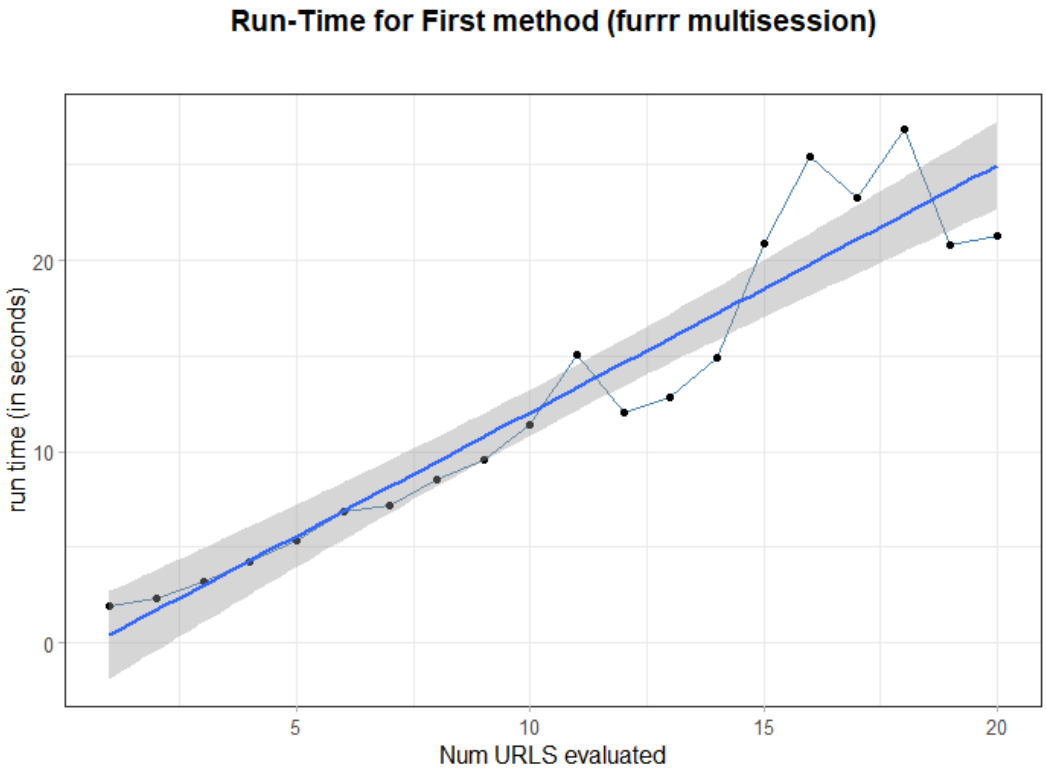


Figure 2.7: computational complexity analysis with Furrr

On the x-axis in the figure 2.7 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 cumulated one at at a time. Looking at the blue smoothing curve in between confidence lines the big-O guess might be linear time $\mathcal{O}(n)$, where n are the number of links considered.

A second attempt tried to explore the `foreach` package (Microsoft and Weston, 2020). This quite recent package enables a new looping construct for executing R code in an iterative way. The core reason for using the `foreach` package is that it supports *parallel execution*, that is, it can execute repeated operations on multiple processors/cores on the computer, or on multiple nodes of a cluster. The construction follows the r-base looping idea, below steps are summarized:

- start clusters on processors cores
- define the iterator, i.e. "i" equal to the number of elements that are going to be looped
- `.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).
- `.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 concern regards that functions inside the %dopar% should be standalone in order to be executed in parallel. For standalone it is meant that everything that is needed to be executed and to output results should be defined inside the %dopar%, as it would be opened a new empty environment for each iteration. Moreover as a further consequence packages imported into each clusters, the .packages methods takes care of that.

It can be grasped quite easily by figure 2.8 that the curve now is flattened and a confident guess might be logarithmic time $\mathcal{O}(log(n))$.

A further performance improvement could be obtained using a new package called `doAzureParallel` which is built on top of the foreach. doAzureParallel enables different Virtual Machines operating parallel computing throughout Microsoft Azure cloud, but this comes at a substantial monetary cost. This

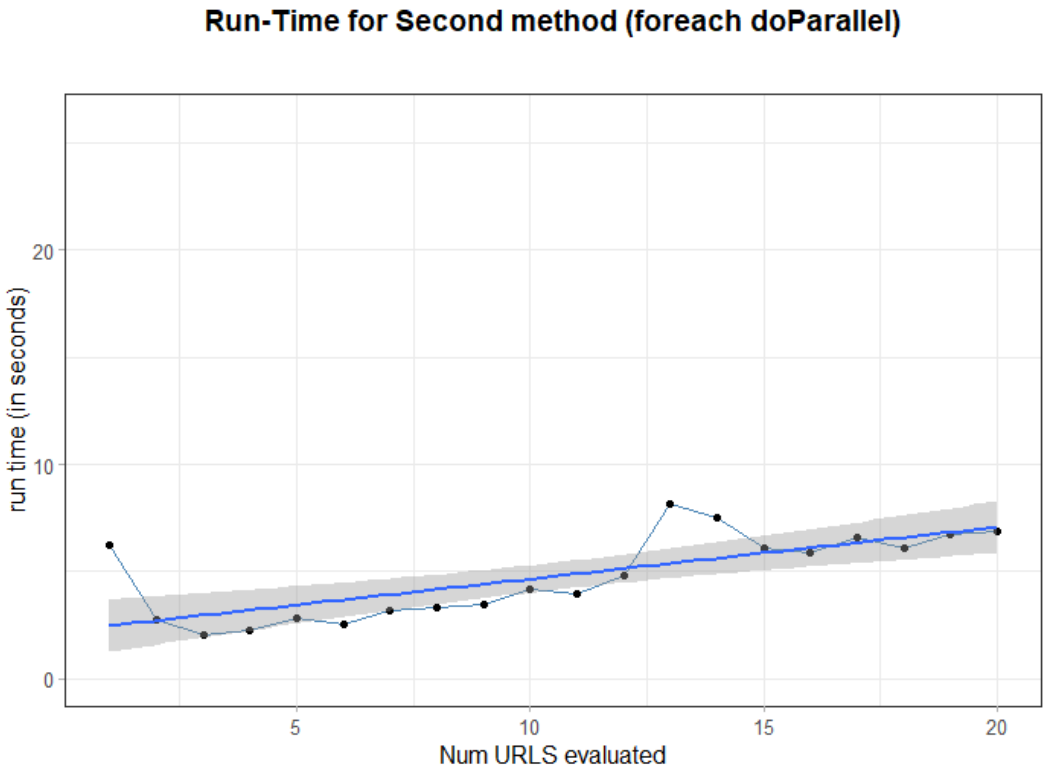


Figure 2.8: computational complexity analysis with Furrr

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 the goal is to have something that separates different sessions. Unleashing Virtual Machines allows from one hand to further increase computational capabilties, so the number of potential requests, from the other it can partition requests among different proper machines (a pool of agents for each VM) extending even more the combination of IDs and as a consequence masquerading even better the scraping automation.

2.5 Open Challenges and Further Improvements

The main challenge remains unsolved since each single element has been optimized but scraping function and, as a consequence, REST API must be contin-

uously maintained. As a matter of fact what unfortunately can not be a-priori optimized are the future changes that involves the first part of the scraping process i.e. decomposing the website structure. Indeed Content architecture, with some further improvements can take care of finding exact information within the web page even if the design is changed. This idea is developed in the package Khalil (2018), which crawls the entire website and searches for targeted keywords, even though end results are not always acceptable. The major enhancement of this approach is that it neatly separates the website structure from the content architecture by locally saving all the html/xml files that compose the website. html are known to be very lightweight so computation of this kind is not weighing down the run time. Once all files are stored the algorithm searched for the keywords within the html, then results are grabbed. For these reasons performances with algorithm of this species are very smooth but results, as anticipated, are under the expectations. Moreover the bigger the website the slower the algorithm. As a partial solution a more robust scraping code can be obtained integrating the existing code with accurate content text mining techniques together with unit testing integration tools like `testthat` Wickham (2011) and `usethis` Wickham and Bryan (2020). The latter packages also bring improvements during software development and ex-post a *TDD* (i.e.Test-Driven Development TDD (2004)) approach would have cut API software development time. Furthermore a very popular tool to develop and automate test API is Postman[1], which is very convenient when POST endpoints are available, since for the moment are not in existence it is not used and the default Swagger[2] interface takes its place. It must be said also that Khalil (2018) is designed to scrape a vast number of websites, as opposite the scraping functions here presented are exclusively built on top of immobiliare.it, even though they can be extended to other related website with no effort, for reasons pointed out in the initial part of section 2.1.1. The way this scraping functions handle errors really facilitates responsive and fast

[1]https://www.postman.com/
[2]https://editor.swagger.io/

debugging but this can not be by any means neither automatized nor notified to the maintainer. The API frequently needs to be tested and to resort to CI. A better way to move toward would be integrating a CI/TD approach as said few lines ago. Moreover error messages can not sometimes be printed out in console and be undesrtood while in parallel beckend. as it is shown the stackoverflow reproducible example[3]. As a consequence to that each time an error occurs the "main" functions needs to be taken out of from the parallel back end and separately evaluated. This is time consuming but for the time being no solutions have been found.

2.6 Legal Profiles (ancora non validato)

"Data that is online and public is always available for all" is never a good answer to the question "Can I use those web data to my scope". Immobiliare.it is not providing any open source data from its own database neither the perception is that it is planning to do so in the future. Immobiliare has not even provided a paid API through which data might be accessed. A careful reading of their terms, reviewed with a intellectual property expert, has been done to get this service running without any legal consequence, as a reference the full policy can be seen in their specialized section[4]. Nevertheless the golden standard for scraping was respected since the robottxt is neat allowing any actions as demonstrated above. So if it might be the case of misinterpretation of their policy, it will be also the case of lack of communication between servers response and immobiliare.it intent to preserve their own intellectual property. What it was shockingly surprising are the low barriers to obtain information with respect to other counterpart online players. Best practices are applied and delayed requests (even though not asked) have been sent to normalize traffic congestion. But scraping criteria followed are once again fully based on common shared best practises (see section 2.2), and *not* any sort of

[3]https://stackoverflow.com/questions/10903787/how-can-i-print-when-using-dopar
[4]https://www.immobiliare.it/terms/

general agreements between parties. As a result a plausible approach could be applying scraping procedures without any prevention. It would not surely cause any sort of disservice for the website since budjet constraints are set low, but in the long run it will cause lagging as soon as budjet or 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 inner rules. 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. Delay is kindly asked and it must be specified, consequnetly this slows down scraping function per se.

- 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 on how big idealista is; as a matter of fact it has a heavy market presence in some of the Europe real estate country as Spain and France. So the hidden intention was to raise awareness on scraping procedure that in a certain remote 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

REST API Infrastructure

In order to provide a fast and easy to use API service to the end user many technologies have been involved. Challenges in scraping as pointed out in section 2.5 are many and still some remains unsolved. Challenges regard not only scraping per se, but also the way the service has to interact with users. Interactions and subjects are many and so are obstabcles to tackle. Some of the main are: API Service has to be executed n given times during the day at $x - y - z$ hours and should be storing data on a cloud database, so that the evolution of the phenomenon can be monitored in time. API service has to be fast otherwise data become obsolete and so happen to the analysis that has relied on the data. API service has to be deployed so that it can be shared over a wider range of different background stakeholders, without having them to know what it takes. From one hand service has to be responsive to immobiliare.it unpredictable changes, thus it needs to be continuously integrated and reviewed. On the other code behind the service has to be version controlled and freezed, so that the service can guarantee continuity and prevent failures. Moreover service has to be scalable at need since, due to deployment, when the number of users increases the run time performances should not decrease. In addition API inbound traffic has to be managed both in terms traffic and security by granting access only to the ones authorized. Open source solutions for each of the requirements are available for back-end and front-end integration.

Moreover documentations related to those technologies are able to offer flexible solutions to be embedded into the R ecosystem. Given the requirements recipe the idea is to provide a REST Plumber API with 4 endpoints each of which calls scraping functions in Parallel built in section 2. On top of that a daily Cron Job scheduler exposes one precise API endpoint, which produces and later stores a .csv file in a NOSQL mongoDB Atlas could database. Containerization happens through a Linux OS (Ubuntu distr) Docker container hosted by a free tier AWS EC2 server machine equipped with 30GB max capacity. API endpoints are secured with https protocols, load balanced and protected with authentication by NGINX reverse proxy server. On a second server a Shiny App calls one endpoint gievn specified parameters that returns data from the former infrastructure. A sketch of the infrastructure is represented in figure 3.1.

Technologies involved are:

- GitHub version control
- Scheduler cron job, section 3.1
- Plumber REST API, section 3.2.1
- Docker containers, section 3.3
- NGINX reverse proxy, section 3.4
- AWS (Amazon Web Services) EC2 3.5
- MongoDB Atlas
- Shiny

As a side note each single part of this thesis has been made according to some of the API inspiring criteria of sharing and self containerization. RMarkdown (Allaire et al., 2020) documents (book's chapters) are compiled and then converted into .html files. Through Bookdown (Xie, 2016) the resulting documents are put together according to general .yml instruction file and are readble as gitbook. Files are then pushed to a Github repository[1]. By a simple trick

[1]https://github.com/NiccoloSalvini/thesis

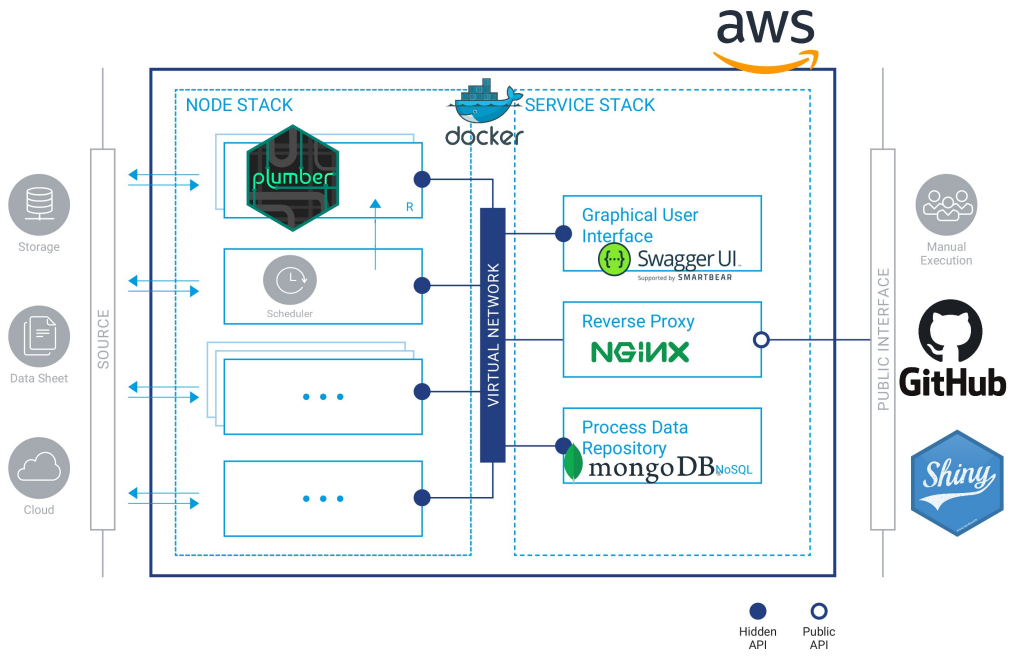


Figure 3.1: complete infrastructure, author's source

with GH pages, .html files are dispalyed into a Github subdomain hosted at link[2]. The resulting deployed gitbook can also produce a .pdf version output through a Xelatex engine. Xelatex compiles .Rmd documents according to a .tex template which formatting rules are contained in a further .yml file. The pdf version of the thesis can be obtained by clicking the download button, then choosing pdf output version in the upper banner. For further references on the topic Xie (2016)

Some of the main technologies implied will be viewed singularly, nonetheless for brevity reasons the rest needs to be skipped.

3.1 Scheduler

Definition 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

[2]https://niccolosalvini.github.io/thesis/

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 references, such as the scheduling algorithm the reader can refer to (Wikiversità, 2020).

3.1.1 Cron Jobs

Definition 3.2 (Cronjob)**.** 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

```
# ┌──────────── 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 Scheduling Syntax

cron daemon are kept. Each line of a crontab file represents a job, and the composition follows the syntax in figure 3.2

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[3] 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, 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

[3]https://crontab.guru/

To a certain extent what it has been already presented since now might fit for personal use. A scheduler can daily execute the scraping script and can generate a .csv file. Later the same .csv file can be parsed into an application and analysis can be locally reported. The solution proposed is totally *not feasible* in a production environment, since in order to be executed a vast number files has to be sourced and a number of functions should be routinely called. For these reasons the present architecture can not be shared. The solution adopted tries to minimize the analyst/scientist involvement into scraping procedures by offering a compact solution that manages all the processes without having to know how scraping under the hood is working.

3.2 REST API

Definition 3.3 (API)**.** 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 due to resource savings. APIs are thought of as contracts, with documentation that represents an general agreement between parties. There are many types of API that exploit different media and architectures to communicate with apps or services.

Definition 3.4 (REST)**.** The specification REST stands for REpresentational State Transfer and is a set of architectural principles.

When a request is made through a REST API it transfers a representation of the state to the requester. This representation, is submitted in one out of the many available formats via HTTP: JSON (Javascript Object Notation), HTML, XLT, TXT. JSON is the most popular because it is language agnostic (wha, 2018), as well as more comfortable to be read and parsed. In order for an API to be considered RESTful, it has to conform to these criteria:

- A client-server architecture made up of clients, servers, and resources, with requests managed through HTTP.
- Stateless client-server communication, meaning no client information is stored between requests and each request is separate and unconnected.
- Cacheable data that streamlines client-server interactions.
- A uniform interface between components so that information is transferred in a standard form. This requires that:
  - resources requested are identifiable and separate from the representations sent to the client.
  - resources can be manipulated by the client via the representation they receive because the representation contains enough information to do so.
  - self-descriptive messages returned to the client have enough information to describe how the client should process it.
  - hypermedia, meaning that after accessing a resource the client should be able to use hyperlinks to find all other currently available actions they can take.
- A layered system that organizes each type of server (those responsible for security, load-balancing, etc.) involved the retrieval of requested information into hierarchies, invisible to the client.

REST API accepts http requests as input and elaborates 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. Further documentation and differences between HTTP and REST API can be found to this reference[4].

open REST API examples: - BigQuery API: A data platform for customers to create, manage, share and query data. - YouTube Data API v3: The YouTube Data API v3 is an API that provides access to YouTube data, such as

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

videos, playlists, and channels. - Cloud Natural Language API: Provides natural language understanding technologies, such as sentiment analysis, entity recognition, entity sentiment analysis, and other text annotations, to developers. - Skyscanner Flight Search API: The Skyscanner API lets you search for flights & get flight prices from Skyscanner's database of prices, as well as get live quotes directly from ticketing agencies. - Openweathermap API: current weather data for any location on Earth including over 200,000 cities.

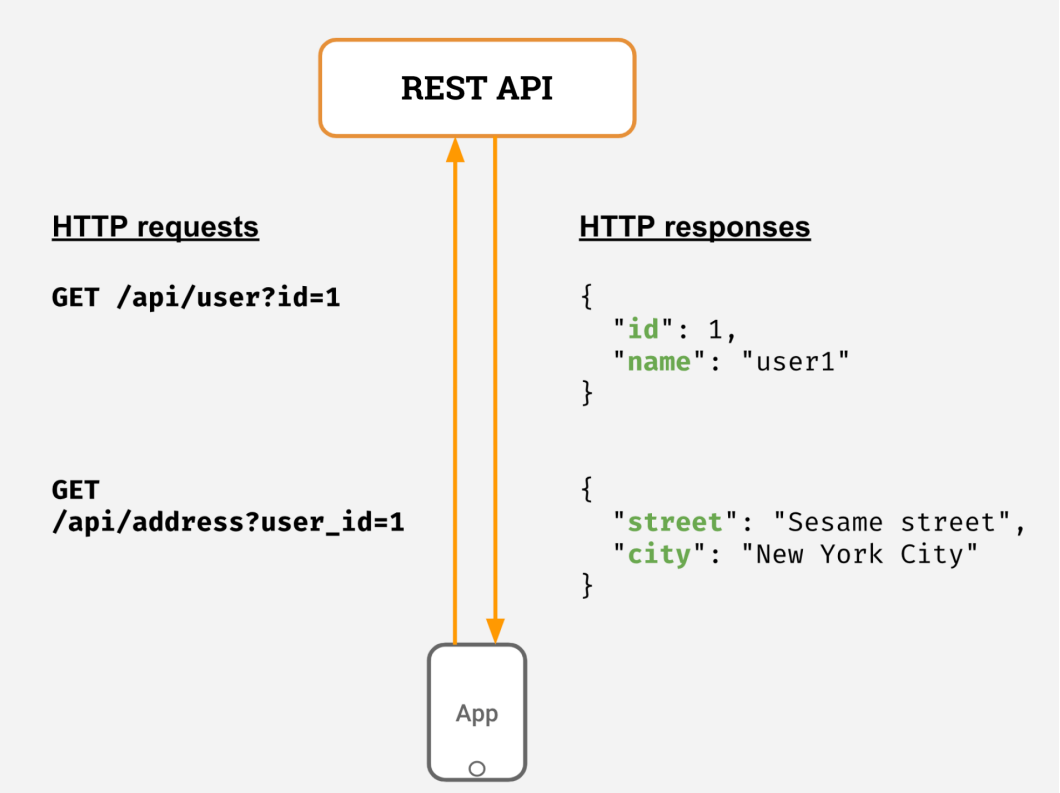


Figure 3.3: API general functioning functioning

3.2.1 Plumber REST API

Plumber allows the user to create a REST API by adding decoration comments to the existing R code, in this case to scraping code. Decorations are a special type of comments that suggests to Plumber where and when the API specifications parts are. Below a simple example extracted by the documentation:

```r
# 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 b 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 options are available to customize plumber API but are beyond the scope, a valuable resource for further insights can be found in the dedicated package website (**?**)

3.2.2 Immobiliare.it REST API

The API service is composed by 4 endpoints */scrape* , */links*, */complete* and */get_data*:

- */scrape performs a fast scraping of the website that leverages a rooted tree shortest path to get to data. This comes at the cost of the number of available covariates to scrape which are 5: title, price, number of rooms, sqmeter, primarykey. By default the end point scrape data from Milan real estate rents. It is a superficial and does not contain geospatial, however it might fit for some basic regression settings. The macrozone parameter allows to specify the NIL (Nucleo Identità Locale), targeting very detailed zones in some of the cities for which is available (Roma, Firenze, Milano, Torino).

- */links: extracts the list of each single advertisement link belonging to each of the npages parameter specified, reacall section 2.1.1. It displays sufficient performances in terms of run time. It is strictly needed 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.

- */get_data: it triggers the data extraction by sourcing the /complete endpoint and then storing .json file into the NOSQL mongoDB ATLAS

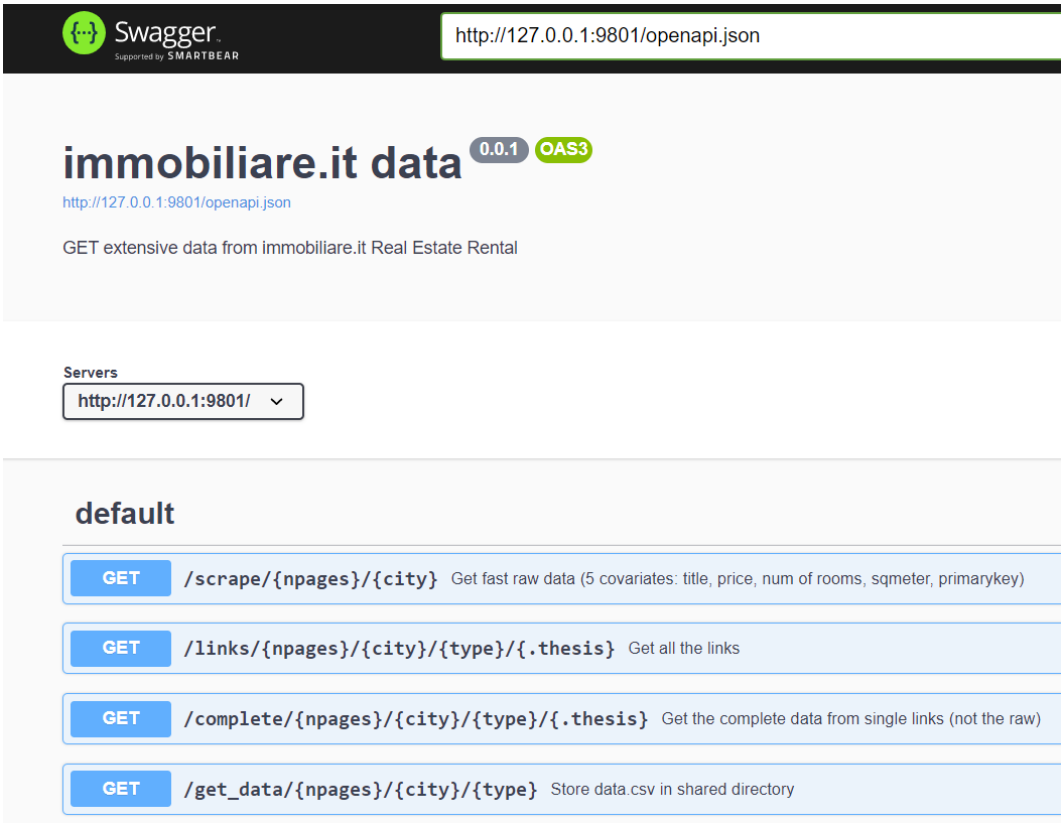


Figure 3.4: Swagger UI screenshot, author's source

3.2.3 REST API documentation

- Get FAST data, it covers 5 covariates:

```
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
```

Up to this point the API can smoothly run in local and potentially can be deployed to share results without requesting scraping process knowledge. However the API software for now is not portable and is very heavy. In addition it can also run into failures for many reasons, one among the others is package versione incompatibility due to updates. In the end it also fully relies on the laptop computational power that can be heavily stressed when a number of API calls are executed, especially for single threaded programming languages as R. The approach followed proposes a dedicated lightweight software environment that minimizes dependencies both improving performances and enabling the *cloud computing* coverage. A fast growing techonlogy is what fits the need.

3.3 Docker

Definition 3.5 (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 OS. The structure differences are figrued in @ref(fig:).

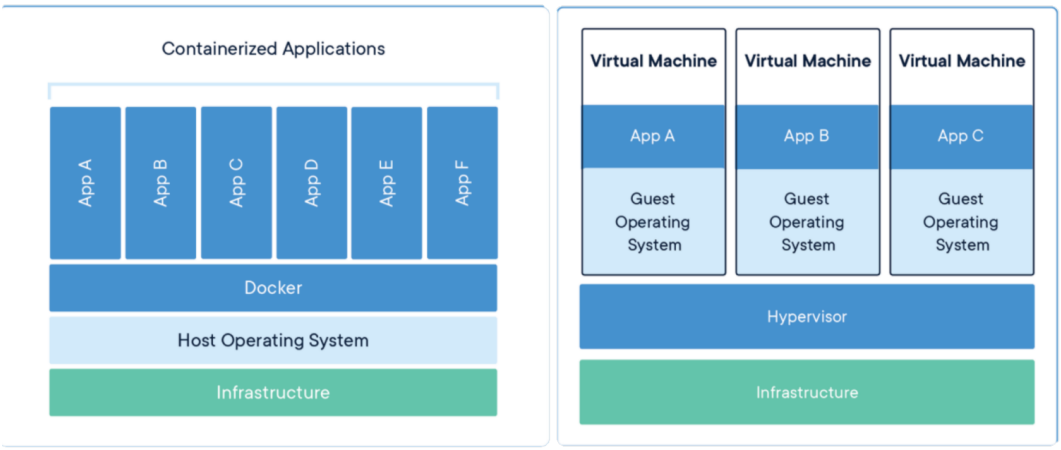


Figure 3.5: Docker containers versus Virtual Machines, *miss source*

When containers are built *Docker container Images* are created and can be open sourced through 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 can connect with GitHub behind

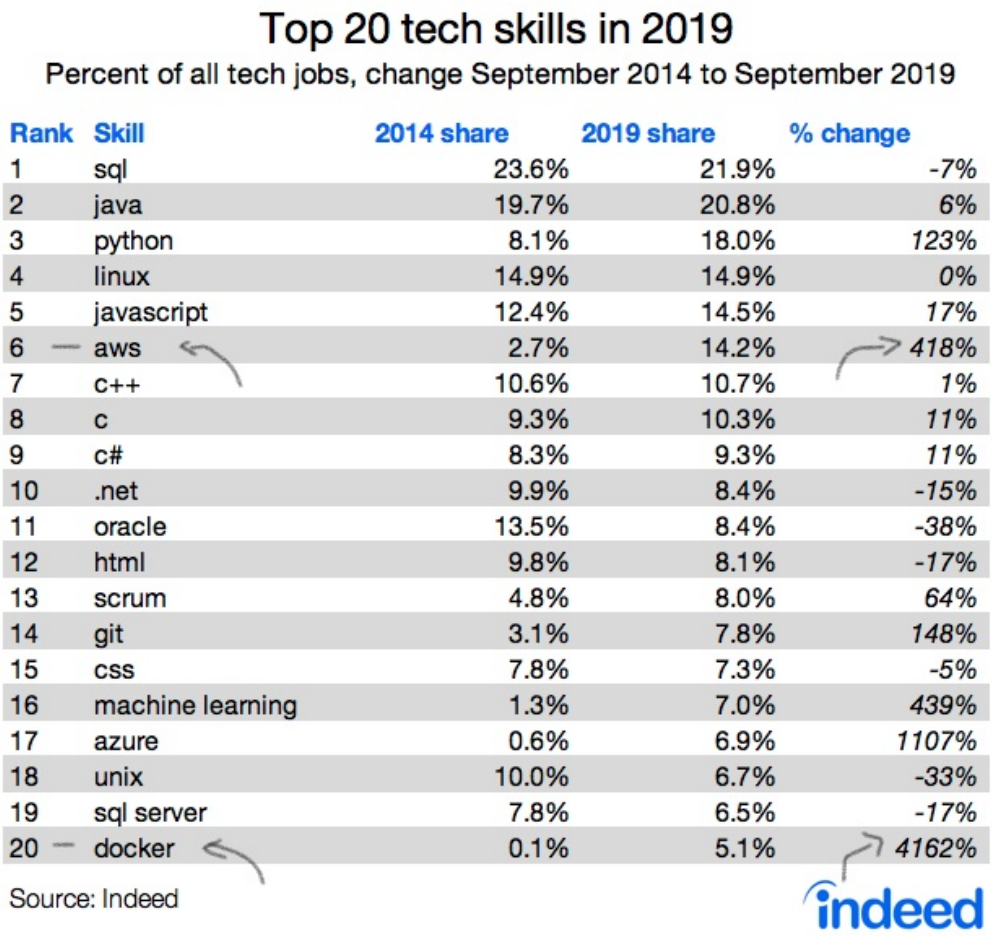
authorization entailing an image version control tool. Once the connection is established changes that are pushed with git to the GitHub repository are passed to Docker Hub. The push command automatically triggers the image building. Then docker image can be tagged (salvini/api-immobiliare:latest)so that on one hand it is recognizable and on the other 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 consuming. 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[5] includes beta support for pre-compiled R packages that can be installed faster. This dramatically reduces packages time installation (Nolis, 2020). In addition to that an open source project rocker[6] has narrowed the path for developers by building custom R docker images for a wide range of usages. What can be read from their own website about the project is: "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".

3.3.1 Why Docker

Indeed[7], an employment-related search engine, released an article on 2019 displaying changing trends from 2015 to 2019 in Technology Job market, a summary of those changes is in figure 3.6. Many changes are relevant in key technologies. Two among the others technologies (i.e. docker and Azure, arrow pointed) have experienced a huge growth and both refer to the certain demand input: *containers* and *cloud computing*. The landscape of Data Science is changing from reporting to application building: In 2015 - Businesses reports drive better decisions. In 2020 - Businesses need apps to empower better

[5]https://packagemanager.rstudio.com/client/#/
[6]https://www.rocker-project.org/images/
[7]https://it.indeed.com/

decision making at all levels.



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 3.6: Indeed top skills for 2019 in percent changes, Flowers (2020) source

For all the things said what docker is bringing to business are (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. This container can be transferred 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*.

3.3.2 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 triggers the building of the image.

The Dockerfile used to trigger the building of the docker container has the following sequential set of instructions in figure 3.7) :

```
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', 'h

# install 'iterators' dep for DoParallel
RUN R -e "install.packages('https://cran.r-project.org/src/contrib/Archive/iterators/iterators_1.0

# install 'foreach' dep for DoParallel
RUN R -e "install.packages('https://cran.r-project.org/src/contrib/Archive/foreach/foreach_1.4.8.t

# 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_1

COPY / /

# expose port
EXPOSE 8000

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

Figure 3.7: Example of a Dockerfile from Docker Hub, author's source

where the instructions are:

- `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.

3.4 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[8].

Definition 3.6 (AWS EC2)**.** Amazon Elastic Compute Cloud (EC2) is a web service that contributes to a secure, flexible computing capacity in the AWS

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

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 3.3.2, 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 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[9], 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.

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

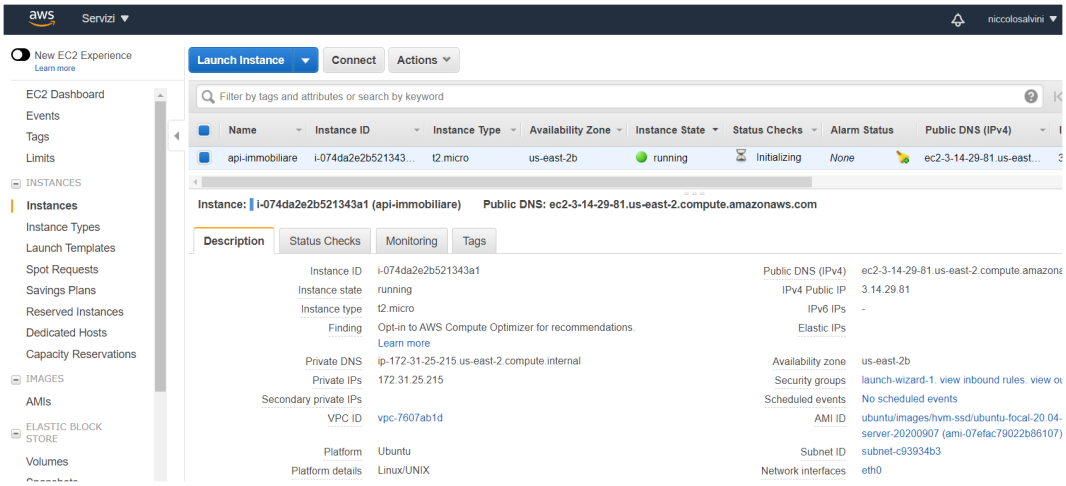


Figure 3.8: aws_dashboard

3.6 Further Integrations

Pins is an r packages this link[10] software development framework and tools for testing this work[11]

[10]https://rstudio.com/resources/rstudioconf-2020/deploying-end-to-end-data-science-with-shiny-plumber-and-pins/?mkt_tok=eyJpIjoiTmprNU1USXhPVEprWXpNMSIsInQiOiJtTUhKVzlvSjVIV2hKc0NRNVU1NTRQYSsrRGd5MWMy

[11]https://github.com/isteves/plumbplumb

Chapter 4

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 4.2 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 4. 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.

4.1 Latent Gaussian Models LGM

Given some observations $y_{i...n}$ 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 $E(y_i)$)) and some other hyper-parameters $\psi_k, \forall k \in 1 ... K$. 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. $g(\phi_i) = \eta_i$. 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, $\beta = \{\beta_1, ..., \beta_M\}$ are the coefficient that quantifies the linear effects on covariates $x = (x_1, ..., x_M)$ and $f_l(\cdot), \forall l \in 1 ... L$ are a set of random effects defined in terms of a z set of covariates $z = (z_1, ..., z_L)$ (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 $\psi = \{\psi_1, ..., \psi_K\}$. 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 $(y_1, ..., y_n)$ 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. $\mathbf{I} = \{1 ... n\}$. 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:

$$\pi(y \mid \theta, \psi) = \prod_{i \in \mathbb{I}} \pi\left(y_i \mid \theta_i, \psi\right) \tag{4.1}$$

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., $\theta \sim \text{Normal}\left(\mathbf{0}, Q^{-1}(\psi)\right)$. 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 5.2.1 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:

$$\pi(\theta \mid \psi) = (2\pi)^{-n/2} |Q(\psi)|^{1/2} \exp\left(-\frac{1}{2}\theta Q(\psi)\theta\right) \tag{4.2}$$

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. $\theta_i \perp \theta_j \mid \theta_{-i,j} \iff Q_{ij}(\psi) = 0$ (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 5.1), 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 4 and verified in chapter 6.* 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* (4.1) and *GMRF* (4.2) 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

$$\pi(\theta, \psi \mid y) \propto \pi(\psi) \times |Q(\psi)|^{1/2} \exp\left(-\frac{1}{2}\theta' Q(\psi)\theta + \sum^{n} \log\left(\pi\left(y_i \mid \theta_i, \psi\right)\right)\right) \tag{4.3}$$

4.2 Approximation in INLA setting

INLA is not going to try to estimate the whole posterior distribution from expression (4.3). 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

$$\pi(\theta_i \mid y) = \int \pi(\theta, \psi \mid \mathbf{y})\pi(\psi \mid \mathbf{y})d\psi \tag{4.4}$$

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 5.5. 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: $\tilde{\pi}(\psi_k \mid y)$

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

Then plugging approximation in the integral observed in (4.4) it is obtained:

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

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

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

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.

4.2.1 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 $\tilde{\pi}(\theta_i \mid y)$ 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).

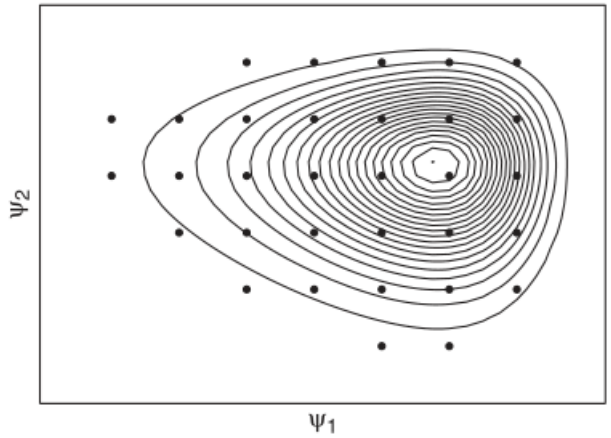


Figure 4.1: CCD to spdetoy dataset, source Marta Blangiardo (2015)

Following Gómez Rubio (2020), approximations of the joint posterior for the hyper paramer $\tilde{\pi}(\psi_k \mid y)$ 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 $\pi(\theta \mid \psi, y)$

- 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)

4.3 R-INLA package in a bayesian hierarchical regression perspective

4.3.1 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 $y(s_1), \dots, y(s_n)$ 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 4.2: summary table list object, source: Krainski (2019)

4.3.2 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. $E(y_i \mid \beta_0, \dots, \beta_M, x_{i1}, \dots, x_{iM}) = \eta_i$). The function that maps the linear predictor into the parameter space is identity as in the initial part of section 4.1 i.e. $\eta_i = \beta_0 + \sum_{m=1}^{M} \beta_m x_{mi} + \sum_{l=1}^{L} f_l(z_{li})$. 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

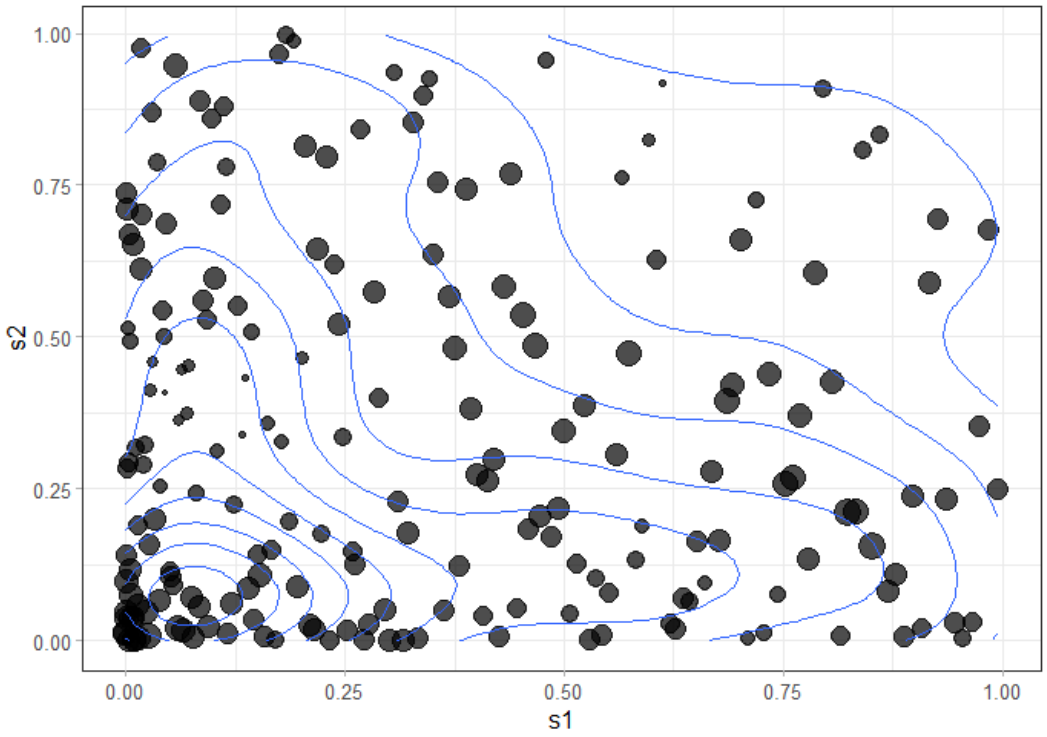


Figure 4.3: SPDEtoy plot, author's source

with the `f()` method inside the formula. `f` are foundamental since they are used to 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:

$$
\begin{aligned}
y_i &\sim N\left(\mu_i, \tau^{-1}\right), i = 1, \ldots, 200 \\
\mu_i &= \beta_0 + \beta_1 s_{1i} + \beta_2 s_{2i} \\
\beta_0 &\sim \text{ Uniform} \\
\beta_j &\sim N\left(0, 0.001^{-1}\right), j = 1, 2 \\
\tau &\sim Ga(1, 0.00005)
\end{aligned}
$$

```
data("SPDEtoy")
formula = y ~ s1 + s2
m0 = inla(formula, data = SPDEtoy)
```

	mean	sd	0.025quant	0.5quant	0.975quant	mode	
(Intercept)	10.1321487	0.2422118	9.6561033	10.1321422	10.6077866	10.1321497	7
s1	0.7624296	0.4293757	-0.0814701	0.7624179	1.6056053	0.7624315	7
s2	-1.5836768	0.4293757	-2.4275704	-1.5836906	-0.7404955	-1.5836811	7

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.

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 credibility interval `inla.hpdmarginal` for a given covariate's coefficient, such that $\int_{q_1}^{q_2} \tilde{\pi}\left(\beta_2 \mid y\right) \mathrm{d}\beta_2 = 0.90$ zwith .1 Confidence Level, in table @ref(tab:higer_posterior_density_interval).

	low	high
level:0.9	-2.291268	-0.879445

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).

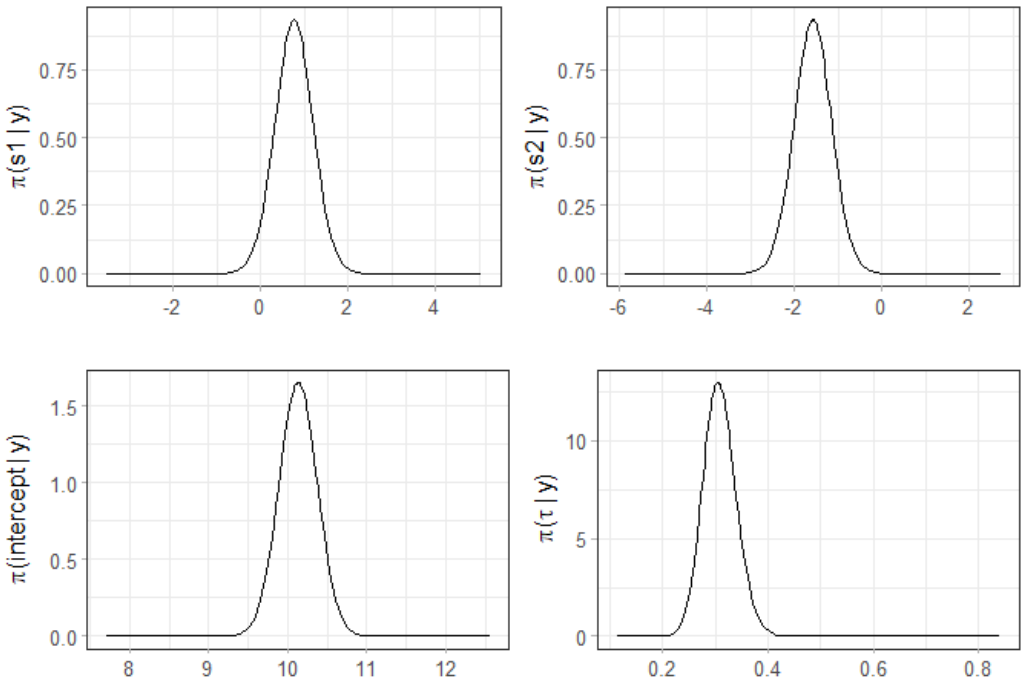


Figure 4.4: linear predictor marginals, author's creation

Chapter 5

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

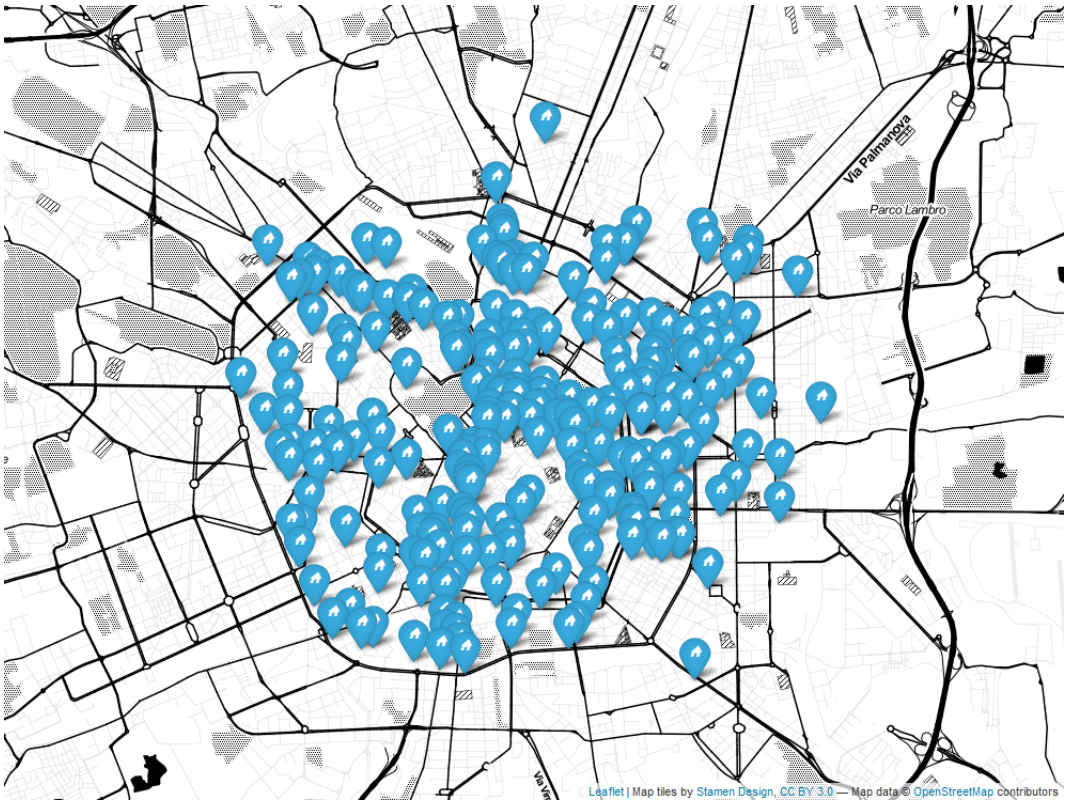


Figure 5.1: point referenced data example, Milan Rental Real Estate, Author's Source

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 6. 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 4) 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.

5.1 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.

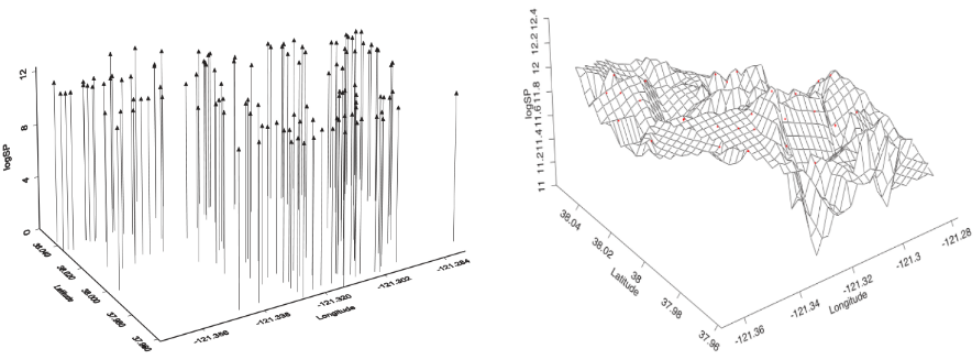


Figure 5.2: 3D scatterplot and surface, Stockton data.

Definition 5.1 (GP definition)**.** 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 $\{y(s_1), ..., y(s_n)\}$ follows a multivariate *Gaussian* distribution with mean $\mu = \{\mu(s_1), ..., \mu(s_n)\}$ and covariance matrix $\mathbf{Q}^{-1}_{i,j}, \forall i \neq j$

Even tough sometimes it is more convenient to express the covariance matrix as its inverse i.e. precision matrix $Q_{i,j}$ (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 $\{y(s_1), ..., y(s_n)\}$ for any given spatial index n and its correspondent location sets $s_{1,...,n}$ to be the same as the one in $\{y(s_1 + h), ..., y(s_n + h)\}$, 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 $\text{Var}(Y(\mathbf{s}+\mathbf{h}) - Y(\mathbf{s}))$. 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 $\gamma(0)$.

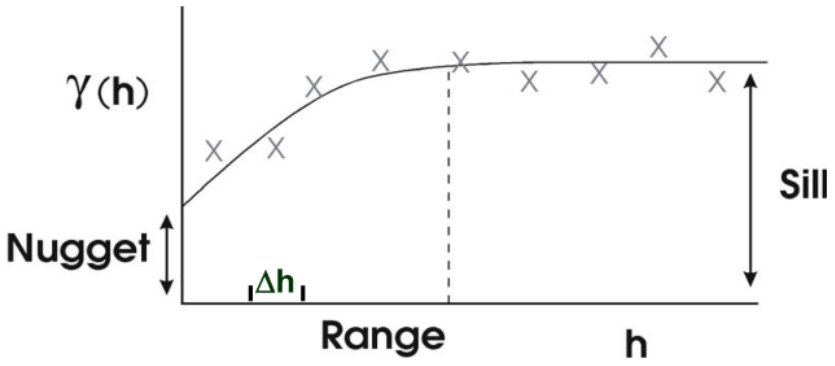


Figure 5.3: variogram example

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:

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

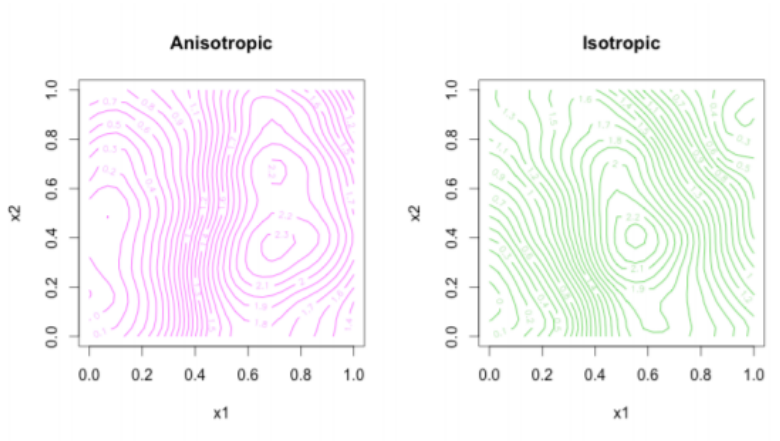


Figure 5.4: isotropy VS anisotropy, source Blanchet-Scalliet et al. (2019)

5.2 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

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

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

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

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

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

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

5.2.1 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 6 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

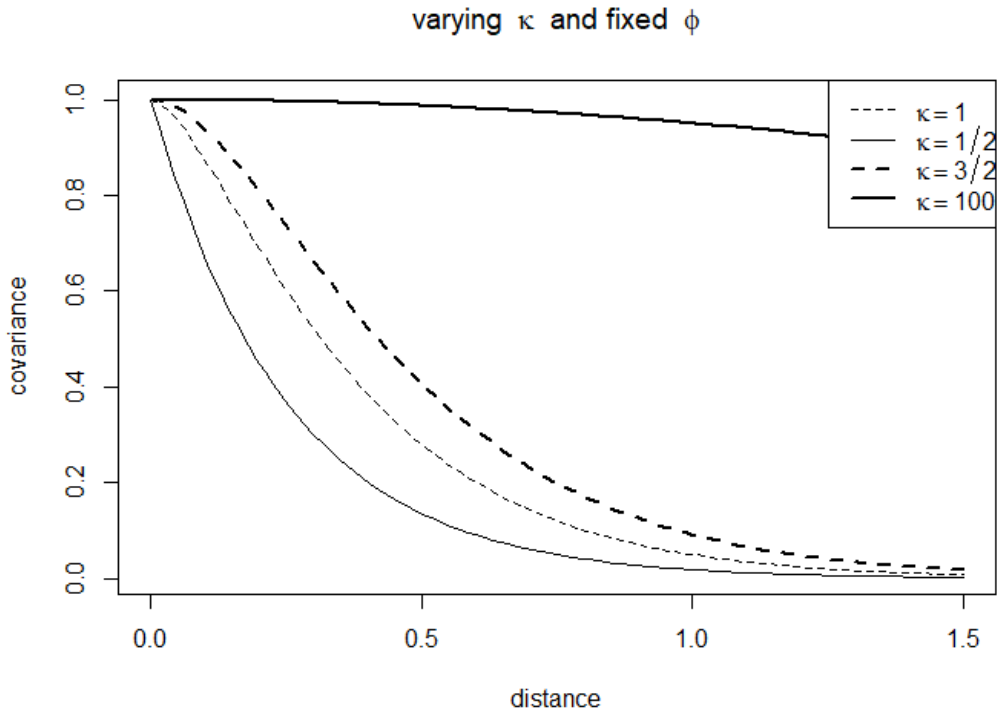


Figure 5.5: matern correlation function for 4 diff values of nu with phi fixed, author's source

5.3 Hedonic models Literature Review and Spatial Hedonic Price Models

The theoretical foundation of the Hedonic Price Models (from now on HPM) resides in 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 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 rents) 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 $\mathbf{C}$.

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

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

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

Historically 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 model including areal effects in the error term since handling these covariates was posing several hard challenges. Areal data choice for Dubin was forced since he was interested in land values, geostatics interest was not a focus also due to the difficulties in gathering accurate data. Coming to recent literature a change in focus has been made by switching from theory based model to estimation methods. As a consequence to the change in focus Ling and Ling (2019) said that practitioners should spend more time in variable selection and model specification with respect to their specific need. 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 do 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.

5.4 Point Referenced Regression for univariate spatial data

Since in HPM the relationships between the characteristics of the house, i.e. vector $\mathbf{C}$ 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 4.1 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 4, 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 has been taken out for simplicity, bu it will be then considered in the regression setting) (valuta l'idea che per interpretabilità di modellarla come Gamma exponential family anzichè tenerla normale)

$$\log(y_i) \sim \text{Normal}(\mu_i, \sigma^2)$$

$$y_i = \mu_i + \varepsilon_i$$

then by the following relationship $E(y_i \mid \beta_0, \dots, \beta_M, x_{i1}, \dots, x_{iM}) = \beta_0 + \sum_{m=1}^{M} \beta_m x_{im}$ it is possible to specify a more general linear predictor (seen also in chapter 4) through an identity link function i.e. $\eta_i = g(\mu_i) = \mu_i$ obtaining:

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

Where, once again, the mean structure linearly depends on some $\mathbf{X}$ covariates,

β coefficients, $f_l(\cdot), \forall l \in 1 \dots L$ are a set of random effects defined in terms of a z set of covariates $z = (z_1, \dots, z_L)$ (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: $\left(X'X\right)^{-1} X'y$ for β and then to obtain the marginal posterior for β it is needed to integrate with respect to σ^2.

In order to engage the spatial coordinate components 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. Recall that GP 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. $\text{N}\left(0, \tau^2\right)$, which offers its contribution error to the nuggets via the covariance function. Consequently there is one more parameter to estimate. It is worth mentioning that the distribution of w_i at a finite number of points is considered a realization of a multivariate Gaussian distribution. In this case, the likelihood estimation is possible and it is the multivariate Gaussian distribution with covariance Σ.

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

The covariance of the marginal distribution of y_i at a finite number of locations is $\Sigma_y = \Sigma + \tau^2\mathbf{I}$, where $\mathbf{I}$ denotes the indicator function (i.e., $\mathbf{I}(i = i') = 1$ if $i = i'$, and 0 otherwise). This is a short extension of the basic GF model, and gives one additional parameter to estimate

5.5 Hierarchical Bayesian models

Spatial Models are characterized by many parameters which in turn are tuned by other hyper-parameters. Traditionally Bayesian hierarchical models are not widely adopted since they have high computational burdens, indeed they can handle very complex interactions via random components, especially when dealing with spatio temporal data Ling and Ling (2019). 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).

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

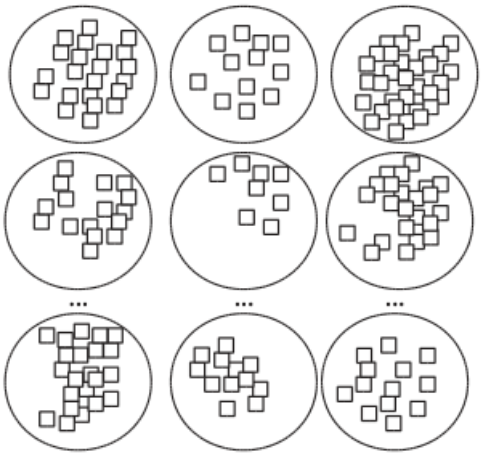
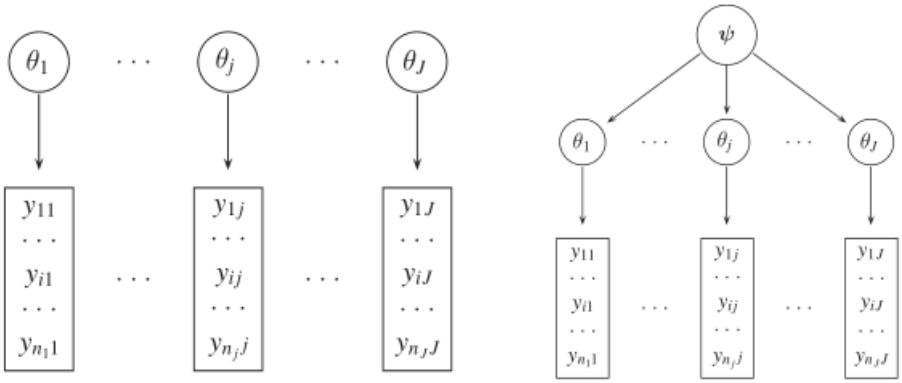


Figure 5.6: 9 levels cat vs observaitions, source Marta Blangiardo (2015)

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 4, 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 $\pi(\psi)$.

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

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.

5.6 INLA model through spatial hierarchical regression

INLA model seen in section 4.1 can be rearranged according to the hierarchical structure considering also the regression settings for point referenced data stated in the previous section 5.4.

As an initial step it is required to specify a probability distribution for $y = \left(y\left(s_1\right), ..., y\left(s_n\right)\right) = \left(y_1, ..., y_n\right)$, this is a mandatory step looking at the 4.3.2 methods needed to compute the `inla()` function. A Normal distribution for simplicity is chosen.

As *first level* is picked up an **exponential family** sampling distribution (i.e. Normally distributed, Gamma one other choice), 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 in turn is tuned by hyper param included in ψ_1. Moreover the w_i surface has to be passed in the formula method definition 4.3.2 via the `f()` function, so that INLA takes into cosideration the spatial component.

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

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

$$\theta \mid \psi_2 \sim \mathrm{N}\left(0, \mathcal{C}(\cdot, \cdot \mid \psi_2)\right)$$

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

5.7 Spatial Kriging

In Geostatistics the main interest resides in the spatial prediction of the spatial latent field pr the response variable at location not yet observed. Assumed the model in the previous section, suppose that $y^\star$ is not a observed occurrence of the response variable at location s_0 (not in the data) of the GP w_i spatial surface estimated through observed refereced points in y. As a consequence of exchangeability (first step previous section 5.6) then $y^\otimes = \{y, y^\star\}$. Then considering INLA notation it is obtained:

$$
\begin{aligned}
\pi\left(y^{\star} \mid y\right) &= \frac{\pi\left(y, y^{\star}\right)}{\pi(y)} \text{ from the conditional probability} \\
&= \frac{\int \pi\left(y^{\star} \mid \theta\right) \pi(y \mid \theta) \pi(\theta) \mathrm{d}\theta}{\pi(y)} \text{ by exchangeability} \\
&= \frac{\int \pi\left(y^{\star} \mid \theta\right) \pi(\theta \mid y) \pi(y) \mathrm{d}\theta}{\pi(y)} \text{ applying Bayes' theorem} \\
&= \int \pi\left(y^{\star} \mid \theta\right) \pi(\theta \mid y) \mathrm{d}\theta
\end{aligned}
$$

A DAG representation might offr the intuition behind Prediction in spatial models:

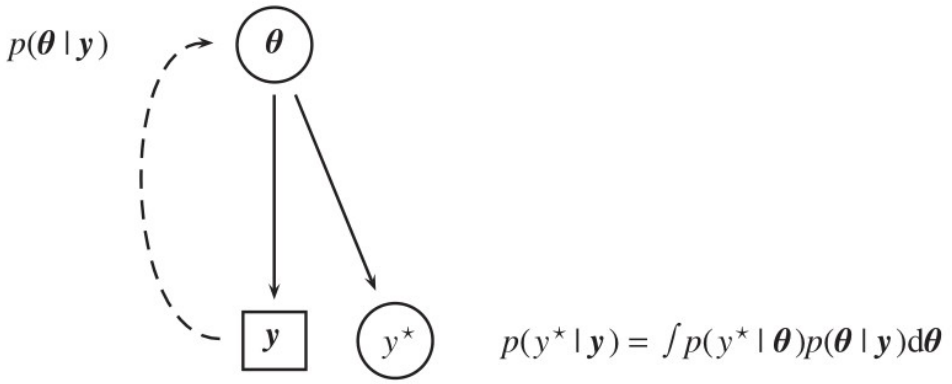


Figure 5.7: Spatial prediction representation through DAG, source Marta Blangiardo (2015)

where $\pi\left(y^{\star} \mid y\right)$ is said predictive distribution and it is meaningful only in the Bayesian framework since the posterior distribution is treated as a random variable, which is totally not true in frequentist statistics.

5.8 Model Checking

(Incrociarlo con altri tesi)

Once the model is set up and fitted a resampling scheme has to be chosen in order to evaluate the model performance. One of the most used method to assess beyasian model quality is LOOCV cross validation and defualt choice

fo R-INLA package. From data is left out one single observation and so that the Validation set is $y_v = y_{-i}$ and the Assessement set is a $y_a = y_i$ the rest of the observations. Two KPI are assumed to be representative:

- CPO conditional predictive ordinate (pettit, 1990): $CPO_i = \pi(y^\star \mid y_v)$
- PIT probability integral tranform (dawid, 1984): $PIT_i = \pi(y^\star < y_i \mid y_v)$

These quantities are used by default by setting control options in the `inla(control.compute = list())` list object by setting them equal o TRUE. Inla also provides an inner method to authomatically handlee failing in computing those two quantities, leadind to values of 1 when predictions are not reliable and the ipposite for 0.Moreover the empirical distribution of the PIT can be used to asses predictive performance: if it is Uniform, so there are not values that strongly differ from the others then the model is correctly checked. Otherwise if the dostribtuon almost approxiamtes any of the other possibles then the Cross validation assessement prediction has led incorrectly predict the "out of the bag" validation sample.

Posteerior checking method exploits a full cross validation where $y_a = y_v$ and it is called predictive checks. Th assessement set now is equal to the validation set,a s a consequence all the observation are evaluated twice. 4 quantities are driver to model estimate quality:

- the *posterior predictive distribution*: $\pi(y^\star \mid y) = \int \pi(y^\star \mid \theta_i)\pi(\theta_i \mid y)\mathrm{d}\theta_i$ which is the likelihood of a replicate observation. When values are small that indicates that are those values are coming from tails, since the area under the curve (i.e. probability) is less. If this happens for many observation then outliers are driving the model leading to poor estimates
- the *posterior predictive p-value* whose math expression is:$\pi(y^\star \leq y_i \mid y)$ for which values near to 0 and 1 indicates poor perfomances.
- *Root Mean Square Predictive Error RMSE*: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^\star)^2}$
- R^2

R-INLA has already antiticipated in chapter 4 section4.3.2 have designed function to compute statistics on posterior distribution as `inla.pmarginal()` returning the cumulative density distribution.

5.9 Prior Specification

Chapter 6

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 (5.1). 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.

6.1 Set SPDE Problem

Given the statistical model already encountered in chapter 5.4:

$$y\left(\mathbf{s}_i\right) = \mathbf{x}\left(\mathbf{s}_i\right)^{'} \beta_j + w(\mathbf{s}) + \varepsilon\left(\mathbf{s}_i\right)$$

where $\eta(\mathbf{s}_i) = g(\mathbf{x}\left(\mathbf{s}_i\right)^{'} \beta_j)$ 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 $w(s) \sim MV\mathcal{N}(0, Q^{-1}_{i,j})$ 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 4.1. 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).

6.2 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.

6.3 First Point Krainsky Rubio TOO TECHNICAL

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

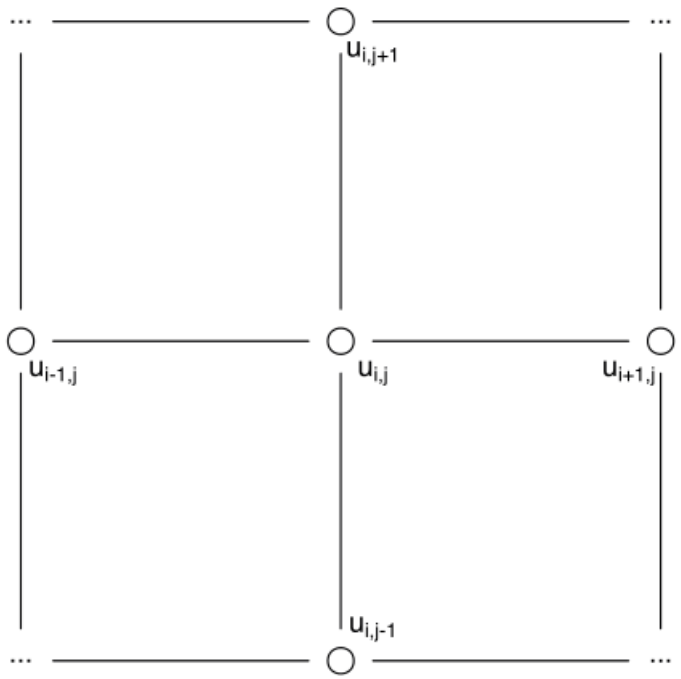


Figure 6.1: lattice 2D regular grid

Chapter 7

Exploratory Analysis

Data comes packed into the REST API end point `*/complete` in .json format. Data can be filtered out On the basis of the options set in the API endpoint argument body. Some of the options might regard the `city` in which it is aevaluated the real estate market, `npages` as the number of pages to be scraped, `type` as the choice between rental of selling. For further documentation on how to structure the API endpoint query refer to section 3.2.3. Since to the analysis purposes data should come from the same source (e.g. Milan rental real estate within "circonvallazione") a dedicated endpoint boolean option `.thesis` is passed in the argument body. What the API option under the hood is doing is specifying a structured and already filtered URL to be passed to the scraping endppoint. By securing the same URL to the scraping functions data is forced to come from the same URL source. The idea behind this concept can be thought as refreshing everyday the same immobiliare.it URL. API endpoint by default also specifies 10 pages to be scraped, in this case 120 is provided leading to to 3000 data points. The `*` refers to the EC2 public DNS that is `ec2-18-224-40-67.us-east-2.compute.amazonaws.com`

`http://*/complete/120/milano/affitto/.thesis=true`

As a further source data can also be accessed through the mondgoDB credentials with the cloud ATLAS database by picking up the latest .csv file generated. For run time reasons also related to the bookdown files continuous

building the API endpoint is called the day before the presentation so that the latest .csv file is available. As a consequence code chunks outputs are all cached due to heavy computation. Interactive maps are done with Leaflet, the result of which is a leaflet map object which can be piped to other leaflet functions. This permits multiple map layers and many control options to be added interactively (LaTex output is statically generated)

A preliminary API data exploratory analysis evidences 34 covariates and 250 rows, which are once again conditioned to the query sent to the API. Immobiliare.it furnishes many information regarding property attributes and estate agency circumstances. Data displays many NA in some of the columns but georeference coordinates, due to the design of scraping functions, are in any case present.

name	ref
ID	ID of the apartements
LAT	latitude coordinate
LONG	longitude coordinate
LOCATION	the complete address: street name and number
CONDOM	the condominium monthly expenses
BUILDAGE	the age in which the building was contructed
FLOOR	the property floor
INDIVSAPT	indipendent property type versus apartement type
LOCALI	specification of the type and number of rooms
TPPROP	property type residential or not
STATUS	the actual status of the house, ristrutturato, nuovo, abitabile
HEATING	the heating system Cen_Rad_Gas (centralizzato a radiatori,
AC	air conditioning hot and cold, Autonomo, freddo/caldo, Cent
PUB_DATE	the date of publication of the advertisement
CATASTINFO	land registry information
APTCHAR	apartement main characteristics

PHOTOSNUM	number of photos displayed in the advertisement
AGE	real estate agency name
LOWRDPRICE.originalPrice	If the price is lowered it flags the starting price
LOWRDPRICE.currentPrice	If the price is lowered it flags the current price
LOWRDPRICE.passedDays	If the price is lowered indicates the days passed since the pric
LOWRDPRICE.date	If the price is lowered indicates the date the price has change
ENCLASS	the energy class according to the land registers
CONTR	the type of contract
DISP	if it is still avaiable or already rented
TOTPIANI	the total number of the building floors
PAUTO	number of parking box or garages avaibable in the property
REVIEW	estate agency review, long chr string
HASMULTI	it if has multimedia option, such as 3D house vitualization ho
PRICE	the monthly price <- response
SQFEET	square meters footage
NROOM	the number of rooms in the house, and their types
TITLE	title of published advertisement

7.1 Data preparation

Data needs to undergo to many previous cleaning preprocess steps, this is a forced stage since API data comes in human readable format, which is not prepared to be modeled. Cleaning steps mainly regards:

- encoding from UTF-8 to Latin due to Italian characters incorrectly parsed.
- *FLOORS* covariate needs to be separated by its *ASCENSORE* and *ACCDISABILI* components, adding 2 more bivariate covariates.

- *LOCALI* needs to be separated too. 5 category levels drain out: *TOTLOCALI*, *CAMERELETTO*, *ALTRO*, *BAGNO*, *CUCINA*. *NROOM* is a duplicate for *TOTLOCALI*, so it is discarded.

- *APTCHAR* is a list column so that each observation has different categories inside. The preprocess step includes unnesting the list by creating as many bivariate columns as elements in the list. Then new columns flag the existence of the characteristics in the apartment. A slice for the fist APTCHAR observation displays:

- – fibra ottica- - - videocitofono- - - impianto di allarme- - - porta blindata- - - reception- - - balcone- - - portiere intera giornata- - - impianto tv centralizzato- - - parzialmente arredato- - - esposizione doppia- -

7.1.1 Maps and Geo-Visualisations

Geographic coordinates can be represented on a map in order to reveal first symptoms of spatial autocorrelation. Observations are spread almost equally throughout the surface even though the response var *PRICE* indicates unsurprisingly that higher prices are nearer to the city center. The map in figure @ref(fig:leaflet_visuals) is a leaflet object, which can needs to be overlapped with layers indicating different maps projections. This is interactive in the .html version, and static is proposed in the .pdf output version. The map object takes a input the latitute and longitude coordinates coming from THE API, and they do not need any CRS (Coordinate Reference System) projection since leaflet can accept the data type.

(other more tmap and ggplot)

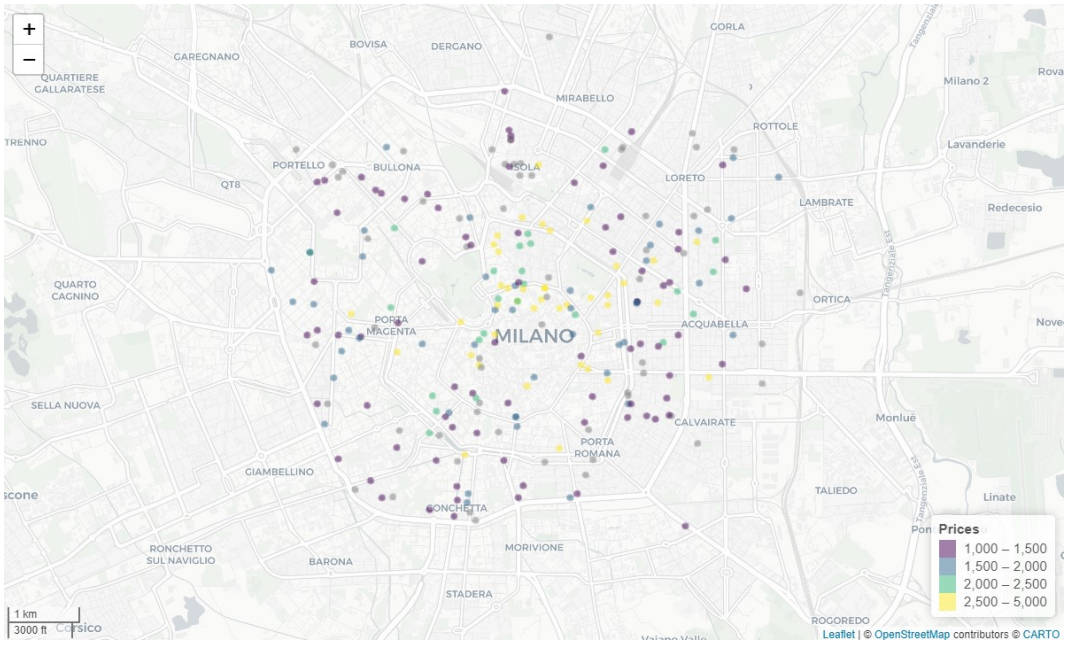


Figure 7.1: Leaflet Map

7.2 Counts and First Orientations

Arranged Counts for categorical columns can give a sense of the distribution of categories across the dataset suggesting also which predictor to include in the model. The visualization in figure 7.2 offers the rearranged factor *TOT-LOCALI*. Bilocali are the most common option for rent, then trilocali comes after. The intuition behind suggests that Milan rental market is oriented to "lighter" accommodations in terms of space and squarefootage. This should comes natural since Milan is both a vivid study and working area, so short stayings are warmly welcomed.

Two of the most requested features for comfort and livability in rents are the heating/cooling systems installed. Moreover rental market demand, regardless of the rent duration, strives for a ready-to-accomodate offer to meet clients needs. In this sense accomodation coming with the newest and most techonological systems are naturally preferred with respect the contrary. x-axis in figure 7.3 represents log_10 price for both of the two plots. Logarithmic scale is needed to smooth distributions and the resulting price interpretation have

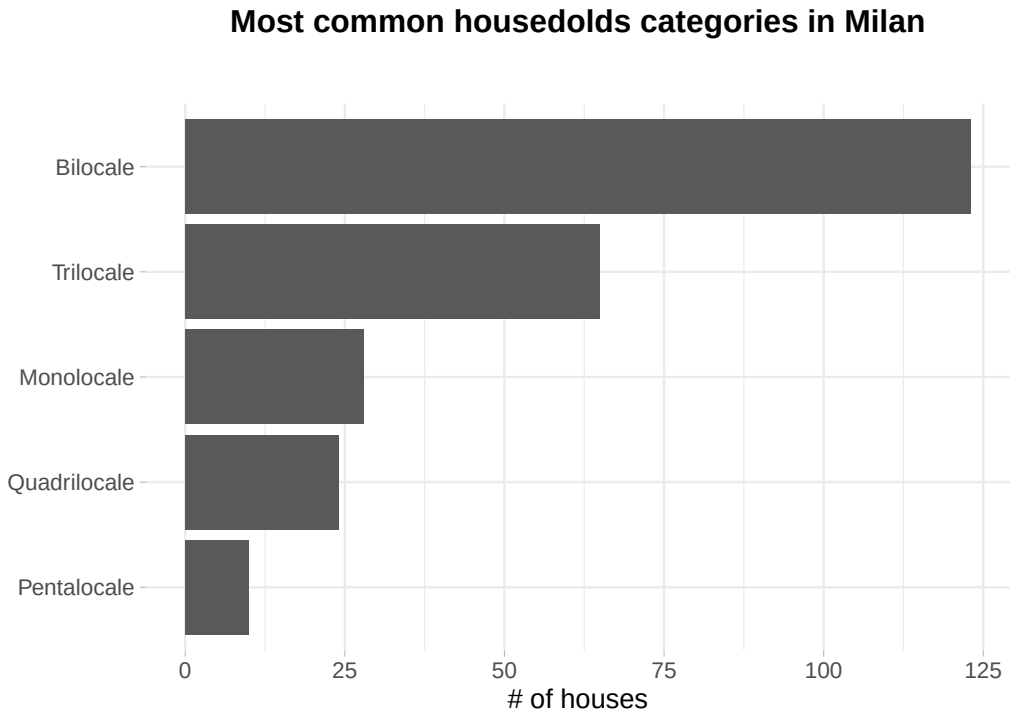


Figure 7.2: Most common housedolds categories

to considered into relative percent changes. Furthermore factors are reordered with respect to decreasing price.

y-axis are the different level for the categorical variables recoded from the original data due to simplify lables and to hold plot dimension. Moreover counts per level are expressed between brackets close to their respective factor. The top plot displays the most prevalent heating systems categories, among which the most prevalent is "Cen_Rad_Met" by far. This fact is extremely important since metano is a green energy source and if the adoption is wide spread and pipelines are well organized than it brings enormous benefit to the city. As a consequence one major concern regards that for many years policies have been oriented to reduce vehicles emission (euro1 euro2…) instead of focusing on house emissions. This was also a consequence of the lack of house data especially in rural areas. According to data there are still a 15% portion of houses powered by oil fired. Then in bottom plot Jittering is then applied to point out the number of outliers outside the IQR (Inter Quantile Range) .25

and their impact on the distribution. A first conclusion is that outliers are mainly located in autonomous systems, which leads of course to believe that the most expensive houses are heated by autonomoius heating systems. Indedd in any case this fact that does not affect monthly price. The overlapping IQR signifies that the covariates levels do not impact the response variable.

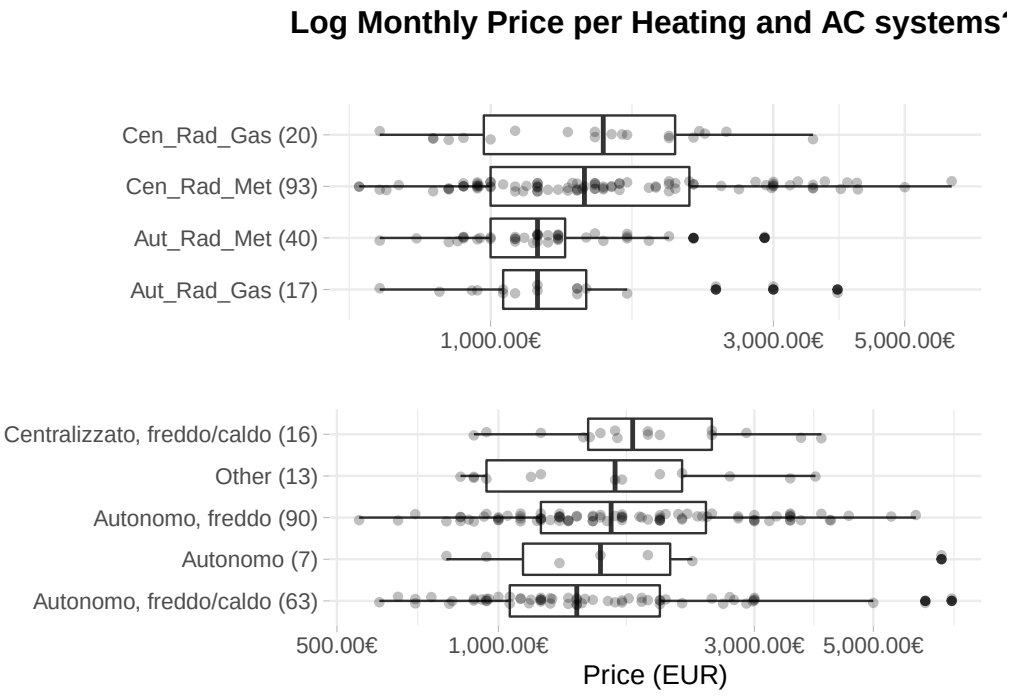
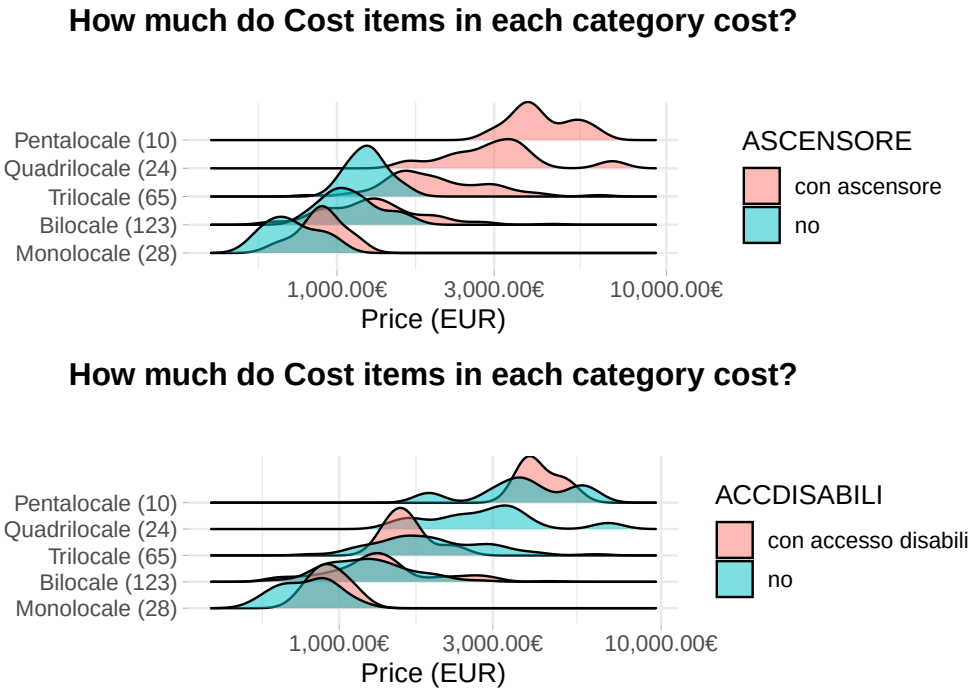


Figure 7.3: Log Monthly Price per Heating/Cooling system?

this visualization intersects allows to discover bimodality in the response variable. Log scales was needed since they are all veru skewd and log scale then is needed also in the model.

(qui ci puoi mettere a confronto per variabile bianria, così vedi cosa includere nel modello esempio sotto dove commentato,)



What it might be really relevant to research is how monthly prices change with respect to house square footage for each house configuration. The idea is to asses how much adding a further square meter affetcs the monthly price for each n-roomed flat. One implication is how the property should be developed in order to request a greater amount of money per month. As an example in a situation in which the household has to lot its property into different sub units he can be helped to decide the most proficient choice in economic terms by setting ex ante the square footage extensions for each of the sub-properties. A further implication can regard economic convenience to enlarge new property acquisitions under the expectation to broadened the square footage (construction firms). Some of the potential enlargements are economically justified, some of the other are not. The plot 7.4 has two continuous variables for x (price) and y (sqfeet) axis, the latter is log 10 scaled due to smoothness reasons. Coloration discretizes points for the each j household rooms totlocali. A sort of overlapping piece-wise linear regression (log-linear due to transformation) is fitted on each totlocali group, whose response variable is price and whose only predictor is the square footage surface (i.e. $\log_{10}(\mathbf{price_j}) \sim +\beta_{0,j} + \beta_{1,j}\mathbf{sqfeet_j}$). Five different regression models are proposed in the top left. The interesting part

regards the models slopes $\hat{\beta}_{1,j}$. The highest corresponds to "Monolocale" for which the enlargement of a 10 square meters in surface enriches the apartment of a 0.1819524% monthly price addition. Almost the same is witnessed in "Bilocale" for which a 10 square meters extension gains a 0.1194379% value. One more major thing to notice is the "ensamble" regression line obtained as the interpolation of the 5 plotted ones. The line suggests a clear slope descending pattern (logarithmic trend) from Pentalocale and beyond whose assumption is strengthened by looking at the decreasing trend in the $\hat{\beta}_1$ predictor slopes coefficients. Furthermore investing into an extension for "Quadrilocale" and "Trilocale" is *coeteris paribus* an interchangeable economic choice.

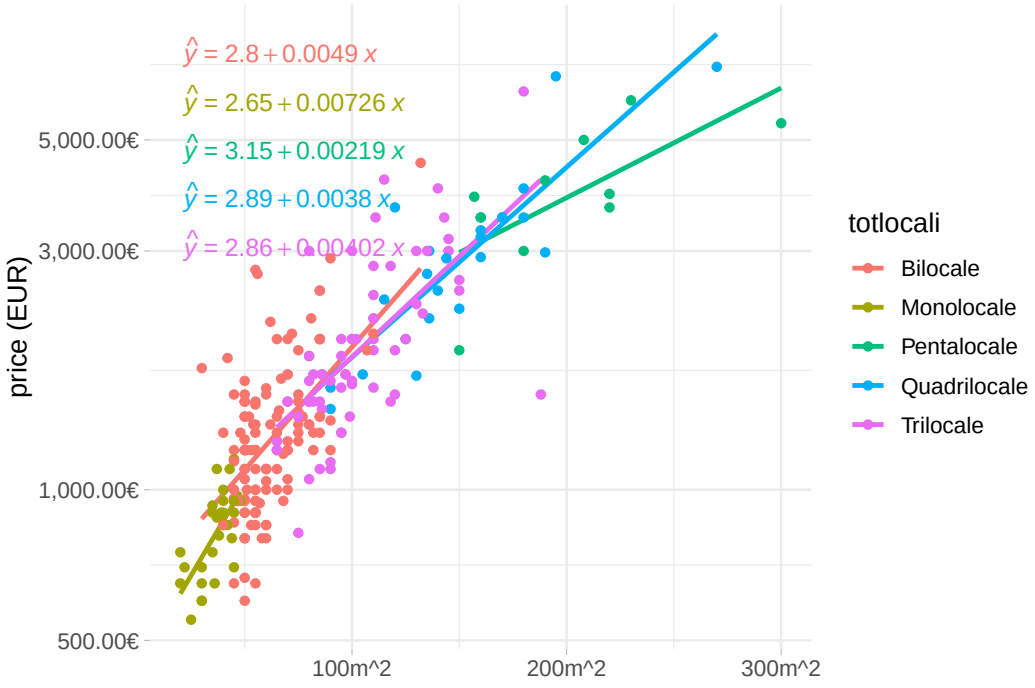


Figure 7.4: Monthly Prices change wrt square meters footage in different n-roomed apt

In table (…) resides the answer to the question "which are the most profitable properties per month in terms of the price per square meter footage ratio". The covariate floor together with the totpiani are not part of the model, indeed they can explain the importance and the height of the building justifying extraordinary prices. The first 4 observations are unsurprisingly "Bilocale",

the spatial column location, not a regressor, can lend a hand to acknowledge that the street addresses point to very central and popular zones. The zones are, first City Life, second Brera and third Moscova, proving that in modeling real estate rents the spatial component is fundamental , even more in Milan.

location	totlocali	price	sqfeet	floor	totpiani	abs_pri
viale cassiodoro 28	Bilocale	1750	30	9 piano	10 piani	58.333
via della spiga 23	Bilocale	2750	55	2 piano	4 piani	50.000
corso giuseppe garibaldi 95	Bilocale	2700	56	2 piano	5 piani	48.214
piazza san babila C.A.	Bilocale	1833	42	4 piano	4 piani	43.642
ottimo stato piano terra, C.A.	Trilocale	3000	80	Piano terra	3 piani	37.500
via federico confalonieri 5	Monolocale	750	20	1 piano	3 piani	37.500

Then as a further point it might be important to investigate a linear model whose response is price and whose covariates are the newly created abs_price and some other presumably important ones i.e. floor. The model fitted is `log2(price) ~ log2(abs_price) + floor`. The plot in figure 7.5 tries to give the intuition on unusual effects across different floors. The interpretation starts by considering the house surface effect (i.e. House Surface (doubling)) for each doubling of the square meter footage. Then what it can be seen is that both apartments and ultimo piano are unsually expensive with respect to their square meter footage. On the other hand the piano rialzato and piano terra are unusually inexpensive for their surface.

In other words the plot has the analytical purpose to demonstrate how monthly price is affected by floor conditioned to their respective square meter footage. The term log2(abs_price) has been converted to more familiar House Surface (doubling) to help with the interpretation.

(se vuoi dire qualcosa sul condominio)

Figure 7.5: Coefficient Tie fighter plot for the linear model: log2(price) ~ log2(abs_price) + condom + other_colors

7.3 Missing Assessement and Imputation

As already pointed out some data might be lost since immobiliare provides the information that in turn are pre filled by estate agencies or privates through standard document formats. Some of the missing can be reverse engineered by other information in the web pages e.g. given the street address it is possible to trace back the lat and long coordinates. Some other information can be encountered in .json files hidden inside each of the single webpages. The approach followed in this part is to prune redundant data and rare covariates trying to limit the dimensionality of the dataset.

7.3.1 Missing assessement

The first problem to assess is why information are missing. As already pointed out in the preliminary part as well as in section **??** many of the presumably

important covariates (i.e. price lat, long, title ,id …) undergo to a sequence of forced step inside scraping functions with the aim to avoid to be missed. If in the end of the sequence the covariates are still missing, the correspondent observation is not considered and it is left out of the scraped dataset. The choice originates from empirical missing patterns suggesting that when important information are missing then the rest of the covariates are more likely to be missing to, as a consequence the observation should be discarded. The missing profile is crucial since it can also raise suspicion on the scraping failures. By Taking advantage of the missing pattern in observations the maintainer can directly identify the problem and derivatives and immediately debug the error. In order to identify if the nature of the pattern a revision of missing and randomness is introduced by Little and Rubin (2014). Missing can be devided into 3 categories:

- *MCAR* (missing completely at random) likelihood of missing is equal for all the infromation, in other words missing data are one idependetn for the other.
- *MAR* (missing at random) likelihood of missing is not equal.
- *NMAR* (not missing at random) data that is missing due to a specific cause, scarping can be the cause.

MNAR is often the case of daily monitoring clinical studies (Kuhn and Johnson, 2019), where patient might drop out the experiment because of death and so all the relating data starting from the death time +1 are lost. To identify the pattern a *heat map* plot 7.6 clarifies the idea:

Looking at the top of the heat map plot, right under the “Predictor” label, the first tree split divides predictors into two sub-groups. The left branch considers from *TOTPIANI* to *CATASTINFO* and there are no evident patterns. Then missingness can be traced back to MAR. Imputation needs to be applied up to *CONDOM* included, the others are discarded due to rarity: i.e. *BUILDAGE*: 14% missing, *CATASTINFO*: 21% and *AC*: 24%. Moreover *CUCINA* and

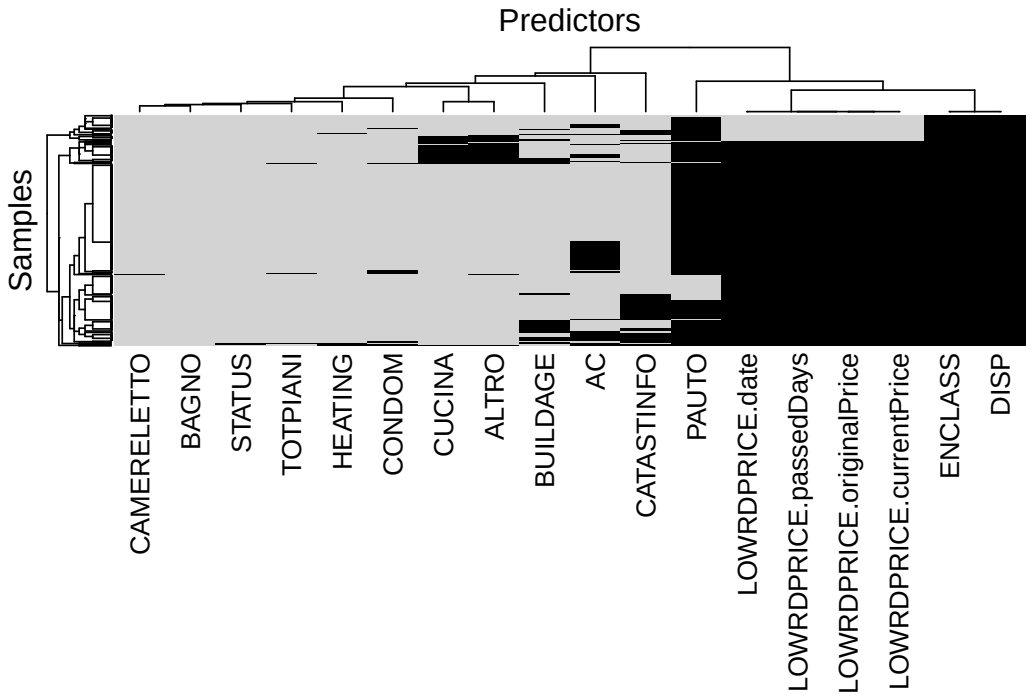


Figure 7.6: Missingness Heatmap plot

ALTRO are generated as "childred" of the original *LOCALI* variable, so it should not surprise that their missing behavior is similar ,whose prevalence is respectively 13% and 14%, for that reason are discarded. In the far right hand side *ENCLASS* and *DISP* data are completely missing and a pattern seems to be found. The most obvious reason is a scraping fail in capturing data. Further inspection of the API scraping functions focused on the two covariates is strongly advised. From *LOWRDPRICE.* covariates gorup class it seems to be witnessing a missing underlining pattern NMAR which is clearer by looking at the co_occurrence plot in figure 7.7. Co-occurrence analysis might suggest frequency of missing predictor in combination and *LOWRDPRICE.* class covariates are displaying this type of behavior. *PAUTO* is missing in the place where *LOWRDPRICE.* class covariates are missing, but this is not happening for the opposite, leading to the conclusion that *PAUTO* should be treated as a rare covariate MAR, therefore *PAUTO* is dropped. After some further investigation on *LOWRDPRICE.*, the group class flags when

the *PRICE* covariate is effectively decreased and this is unusual. That is solved by grouping the covariate's information and to encode it as a two levels categorical covariate if lowered or not. Further methods to feature engineer the *LOWRDPRICE.* class covariates can be with techniques typical of profile data, further references are on Kuhn and Johnson (2019).

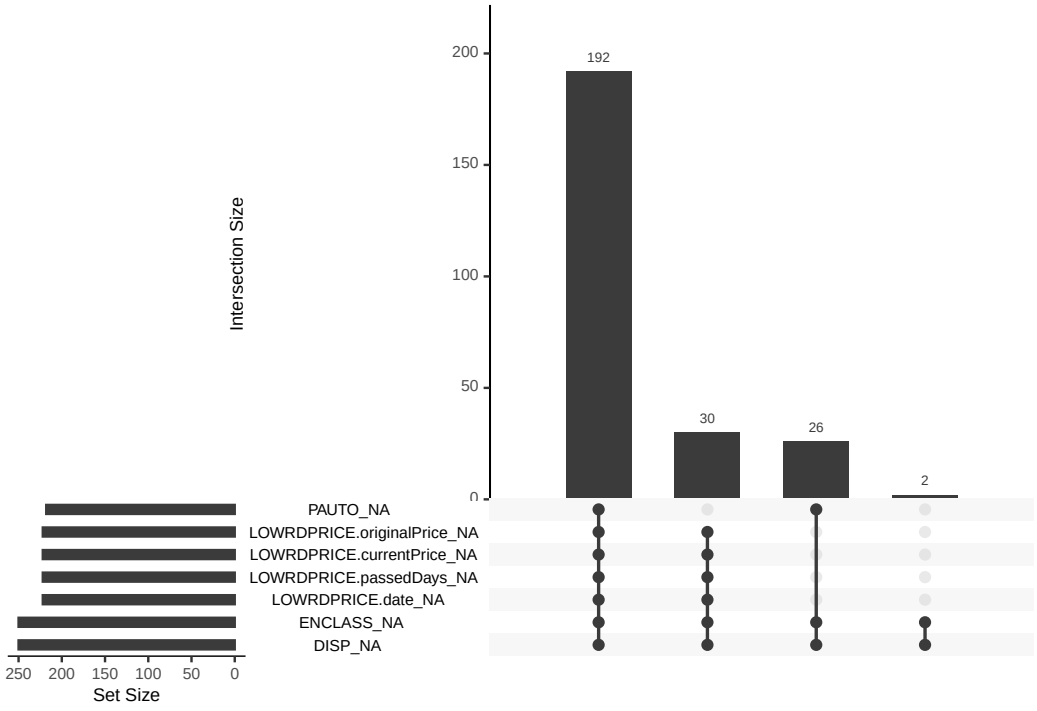


Figure 7.7: Missingness co-occurrence plot

7.3.2 Covariates Imputation

A relatively simple approach to front missingness is to build a regression model to explain the covariates that have some missing and plug-back-in the respective estimates (e.g. posterior means) from their predictive distributions Little and Rubin (2014). This approach is fast and easy to implement in most of the cases, but it ignores the uncertainty behind the imputed values (Gómez Rubio, 2020). However it has the benefit to be a more than a reasonable choice with respect to the number of computation required, especially with INLA and in a spatial setting. That makes it the first choice method to follow since

imputation regards also a small portion of data and predictors. At first it is considered the predictor *condominium* for which some observation are missing. Indices are:

```
##  [1]  19  74  77  90  99 113 116 120 179 249
```

A model is fitted based on missing data for which the response var is condominium and predictors are other important explanatory ones, i.e.`condom ~ 1 + sqfeet + totlocali + floor + heating + ascensore`. In addition to the formula in the inla function a further specification has to be provided with the command `compute = TRUE` in the argument control.predictor. The command `compute` estimates the posterior means of the predictive distribution in the response variable for the missing points. The estimated posetior mean quantities are then imputeda are in table @red(tab:CondomImputation)

	mean	sd
fitted.Predictor.019	198.11095	19.67085
fitted.Predictor.074	162.96544	13.29456
fitted.Predictor.077	99.38197	32.34108
fitted.Predictor.090	331.73519	33.05035
fitted.Predictor.099	170.54068	12.30267
fitted.Predictor.113	196.61593	15.86545
fitted.Predictor.116	108.40482	20.79689
fitted.Predictor.120	162.86977	25.61622
fitted.Predictor.179	165.03632	20.53485
fitted.Predictor.249	117.24234	30.80290

A further method for imputation has been designed by *Gómez-Rubio, Cameletti, and Blangiardo 2019) miss lit* by adding a sub-model for the imputations to the final model through the inla function. This is directly handled inside the predictor formula adding a parameter in the latent field. However the approach makes the model more complex with a further layer of uncertainty to handle. At first the additive regression model with all the

covariates is called including the covariates with missing values. The response variable *PRICE* displays no missing values and the model fitted is:

7.4 Model Specification

7.5 Mesh building

PARAFRASARE The SPDE approach approximates the continuous Gaussian field w_i as a discrete Gaussian Markov random field by means of a finite basis function defined on a triangulated mesh of the region of study. The spatial surface can be interpolated performing this approximation with the inla.mesh.2d() function of the R-INLA package. This function creates a Constrained Refined Delaunay Triangulation (CRDT) over the study region, that will be simply referred to as the mesh. Mesh should be intended as a trade off between the accuracy of the GMRF surface representation and the computational cost, in other words the more are the vertices, the finer is the GF approximation, leading to a computational funnel.

Arguments can tune triangularization through inla.mesh.2d() :

- `loc`:location coordinates that are used as initial mesh vertices
- `boundary`:object describing the boundary of the domain,
- `offset`: argument is a numeric value (or a length two vector) and it is used to set the automatic extension distance. If positive, it is the extension distance in the same scale units. If negative, it is interpreted as a factor relative to the approximate data diameter; i.e., a value of -0.10 (the default) will add a 10% of the data diameter as outer extension.
- `cutoff`: points at a closer distance than the supplied value are replaced by a single vertex. Hence, it avoids small triangles
- `max.edge`: A good mesh needs to have triangles as regular as possible in size and shape.

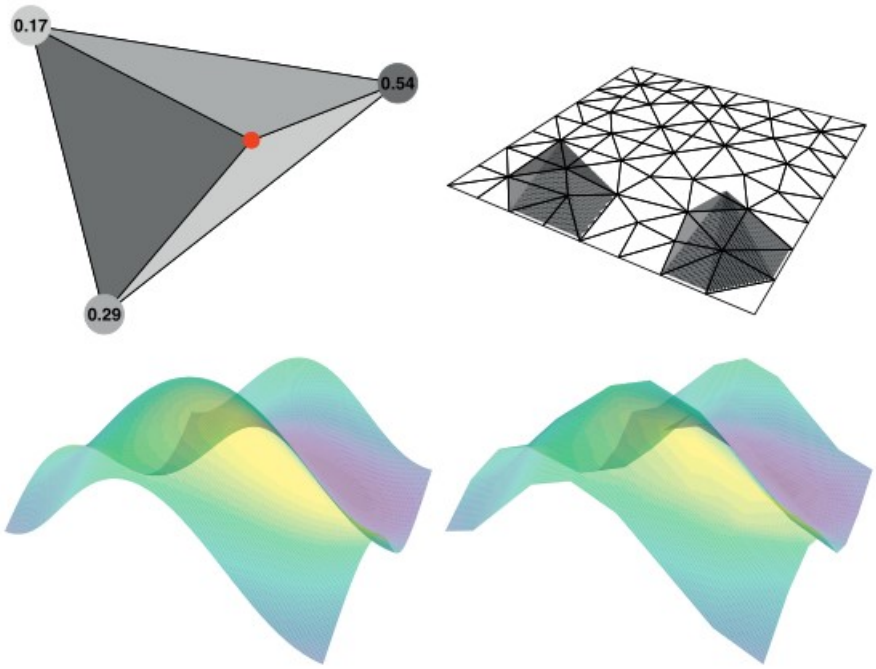


Figure 7.8: Traingularization intuition, Krainski (2019) source

- `min.angle`argument (which can be scalar or length two vector) can be used to specify the minimum internal angles of the triangles in the inner domain and the outer extension

A convex hull is a polygon of triangles out of the domain area, in other words the extension made to avoid the boundary effect. All meshes in Figure 2.12 have been made to have a convex hull boundary. If borders are available are generally preferred, so non convex hull meshes are avoided.

7.5.1 Shinyapp for mesh assessment

INLA includes a Shiny (Chang et al., 2018) application that can be used to tune the mesh params interactively

The mesh builder has a number of options to define the mesh on the left side. These include options to be passed to functions inla.nonconvex.hull() and inla.mesh.2d() and the resulting mesh displayed on the right part.

7.5.2 BUilding SPDE model on mesh

7.6 Spatial Kriging (Prediction)

QUI INCERTEZZE

Chapter 8

Model Selection & Fitting

8.1 Model Criticism

evaluation of the variables to include in the mode„ assumptions of the model i.e. exchangeablity and independence prior distribution to assign to parameters and hyper parameters.

8.2 Spatial Kriging

8.3 Model Checking

if (models > 2){

```
## Model Selection
```

```
  IDEA: proporre due modelli uno più interepretabile con distribuzione normale
```

}

Chapter 9

Shiny Application

with UI build with free tool for front end design ion shiny fomantic-ui[1]. prendi shiny app e rifai interface. in questo blog vedi Hacaton tirato e vincitori blog[2].

Senno app paula moraga che ha già simil modello dentro,

senno flexdashboard paula moraga.

this inspiration[3]

[1]https://fomantic-ui.com/
[2]https://blog.rstudio.com/2020/11/10/the-appsilon-shiny-semantic-pocontest/
[3]https://demo.appsilon.ai/apps/polluter/

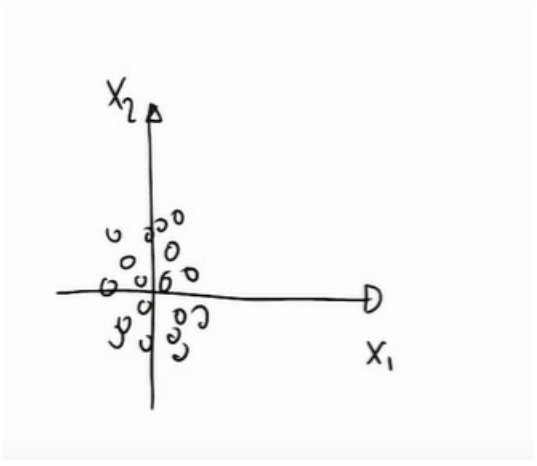
Appendix

9.1 GP basics

Nando de Freitas 1[4]

Nando de Freitas 2[5]

lets say there are a cloud of points represented by two variables x1 and x2. the cloud of points describes a realization of this two variable i.e. height and weight and then you just plot it , you might get measurement like that,



or:

each circle is a mesuduraments. now when we use multivariate gaussian we fit gaussian to data, the process of learning is to fit a gaussian to data, the ultimate goal is to describe the data, the smartest gaussian in the first image is to center the mean in the 0 and the draw a cricle containin all the other observation. Instaed for the second image it is still centering the mean in 0

[4]https://www.youtube.com/watch?v=4vGiHC35j9s&t=164s
[5]https://www.youtube.com/watch?v=MfHKW5z-OOA

but now it is an ellipse describing the variability, the size of the elipse descrube the variability of the data. the center is a vector μ_i that it is beacuse we have two components x_1 and x_2 whhose mean is 0 for each of the other. This is true for all the observation which have two coordinates too x_1 and x_2. in vector notations we have for the mean:

$$\mu = \begin{bmatrix} \mu_{x_1} \\ \mu_{x_2} \end{bmatrix}$$

for each of the points, e.g. for point 1:

$$\mathbf{x_1} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

the can be neagtive positive, the Real numbers, usually we have $\mathbb{R}^2$ extending from - infinity to + infinity, to the power of two because we have 2 dimensions, a Real plane.

any point is gaussian distributed when with mean .. an variance. how we explain covariance, thorugh *correlation.* we do it by correlation with its noraml forms. the covariance is the term that goes insisde the matrices in the upper right of the matrxi we have the expectation of x_1 times x_2, like $\mathbb{E}(x_1 \cdot x_2)$, where the extactation in the gaussian case is the mean which is 0, so the corresponding values is 0. the covariance essentially is the dot product ref dot product[6] of x_1 and x_2 variable, so what happens when you take the dot product of vectors, if for example you take a vector that looks like 1 and 0 and you take the dot product of one other vecto 1 and 0, so that:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = 1$$

You will end up with 1, recall dot productm first element first vcetor times first element second vectro and second element first vector times second element

[6]https://mathinsight.org/dot_product_matrix_notation

secon vector. So identical vector will get a high dot product value leading to a high similarity measure. Dot product can be indeded as a similarity measure. … But if you take two different vector as 1 0 and 0 1 then:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = 0$$

This time the multiplication leads to 0 value, as a matter of fact they are different. They are no similar. IF two points are closed the dot product will be high in 2D. What the covariance should be? if variances are assumed to be 1 then in this case i qould expect to be 0, i.e. covariance matrix is:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{cov_{plot1}}(x_1, x_2)$$

because I can picka poin tin two pointa in this cloud. Suppose i increase x1 then my chance of getting a x2 point that is positive or negative is the samee, knowing somthin about x1 give nothign about x2. no information is proivede. On the other hand i the second plot knowing a positive value of 1 can suggest with a certain probability that x2 will be positive (great proabibility. So some information is provided), e.g.

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} = \mathbf{cov_{plot2}}(x_1, x_2)$$

Some positive number idicates that i expect a positive inc rease iwhen boht of the two are increasing singularly. thsi is what the correlation, the basis to do linear regresssion and non linear- thei is a bivariate gaussian. If the entri3es are means that they are uncoorellated, if they are non-zero then they are correlated, theby can be both positive or negative (correlatiob)

now lets generate a gassiian distrivution so x_1 and x_2 in 2D and then a third dimension hwere we express probability, this is said joint distribution. So i am going to cu this gaussian at certain point for x_1 and cut a plain

rigght thgouth this gauissan imagine to ahava cake and then taka kkniw and cut it.(see the image)

form the man perspective you are goin to see a gaussian distribution, you will be lookong at x_ and you will be seeing a gaussain plot in green. this is the probability of x_1 gievn x_1. also said "conditioned" probabolity. This gaussian has a mean like the one alreasdy seen and this is the center of the gaussian, we can rewrite the mena and variance of the multivariate gaussian describing the cloud of points. sigma are the covaraince martix sigma.

… sigma 1 and signa 2 if you have 1 d varibale the widjth has to be postive, for mulitvariate gaussian equl so here positive definitness: covariance mateix symetric.
… any artibitray variable transposed x time the covarince matrix nedds to be positive. what is the mean of this gaussian i might want to know what is the widht of this gaussian would it be great if there is a formula that guven the cloud of point and likelihood estimation. we coilf obtain the red bell in figure. Compute the green curve how it is done? this requires some work and it is said matrix *invesrion lemma*, this is foudamental for machine learnign. let's assume it. The theorem says that the the mean fof the gaussian is the mean of x_1 and then some other operation with sigma, see below from paci (miss ref)

- $\mathrm{E}[\mathbf{X}_1 \mid \mathbf{X}_2] = \mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}(X_2 - \mu_2)$ is the conditional mean
- $\mathrm{Var}[\mathbf{X}_1 \mid \mathbf{X}_2] = \Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ is the conditional variance

the theorem says toi ocnsider a multivariate gaussian a vector 1 and a vector 2 each vecto compinent has a mena and a covarianc matrix, this by lemma gives us the expression and the math behind is no tremendous, but it is long. What it is important is to undestand fto go from a joont to a conditional distribution in our case. thats i the value od the theorem.

One background further thing: assume that we have a gaussian variable distribution that we want to sample fromm, we had now ewe are going to do

the opposite, before we had poitns and we tried to figure out the curve, now we have the curve and we are gointg to try to rpoduce data. I need to be able to draw sample froma gaussian distribution. i will assume that i have a meachnism that produces a uniform samples, so you have a random numebr generatior with equal probabolity from 0 to 1, I assume a also the cumulative aof a gaussian.

the cumulative of a gaussian is what you get if you syrta summing the area under the curve of the gaussian as you move from the left. value after valure you can plot the cumulative ahead (see figure) the point where there is a flex point is the mean beacuse tha gassias is symmetric. The asymptot is 1 becuase the are under the curve sumes to 1. If i can draw a random number form Uniform and the project it to thre cumulative and then finally projct it back to the gaussian distribution. Inverse cumulative mapping. If oyu do this multiple times you are going to have many sample palced next to the mean and as sparse as the variance. in this process of sampling try to sample a point i froma gaussian that has mean 0 a variance 1, now letes try to draw a point from a gaussian with mean mu and variance sigma. …

In the multivariate case suppose that we have evctor with two variables how do i draw a vector from a multivariate gasussian with 0 means and plot 1 covarianc ematrix. the theormeem also says that the marginal distribution can be seen by civariance matrix , fist take the men_1 and take upper left element from the covariance matrix obtaining the marginal rpobabiloty for x_1, i.e.

$$\pi(x_1) = \mathbf{N}(\mu_1, \Sigma_{11})\pi(x_2) = \mathbf{N}(\mu_2, \Sigma_{22})$$

then in our problem:

$$\pi(x_1) = \mathbf{N}(0, 1)\pi(x_2) = \mathbf{N}(0, 1)$$

Then for simplicity we can simplyfy by groupign vector into: (vectore exoression multivariate)

I need a wau to take square trotto of matrices, if x come sfroma MVG

35:01–

Bibliography

(2004). Test driven development.

(2018).

(2020). Css.

(2020). Html.

(2020). Javascript.

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.

Flowers, A. (2020). Indeed tech skills explorer: Today's top tech skills.

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

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

Harrison, J. (2020). *RSelenium: R Bindings for 'Selenium WebDriver'*. R package version 1.7.7.

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.

Khalil, S. (2018). *Rcrawler: Web Crawler and Scraper*. R package version 0.1.9-1.

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.

Kuhn, M. and Johnson, K. (2019). *Feature Engineering and Selection*. 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.

Little, R. and Rubin, D. (2014). *Statistical Analysis with Missing Data, Second Edition*, pages 200–220.

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.

Shah, T. (2018). *ratelimitr: Rate Limiting for R*. R package version 0.4.1.

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.

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

Wickham, H. (2011). testthat: Get started with testing. *The R Journal*, 3:5–10.

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

Wickham, H. and Bryan, J. (2020). *usethis: Automate Package and Project Setup*. R package version 1.6.3.

Wickham, H., Hester, J., Müller, K., and Cook, D. (2017). *memoise: Memoisation of Functions*. R package version 1.1.0.

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.