R for Urbanism

Introduction and demo of R for applications in Urbanism

Clémentine Cottineau
Claudiu Forgaci

19.11.2021

TT Peer-Teaching Session #1

Intro. Why we switched to R

Claudiu:

- **Due to challenges encountered during my PhD thesis:** having to combine different kinds of software, spending time of repetitive tasks, making mistakes due to repetitive tasks done manually, not being able to customise.

- **Seeing how overcoming these challenges allows me to scale up:** answer questions that I could not answer before, automate tasks that would take a lot of time.

- **Seeing how extensible GIS can be:** with spatial R packages, also easy to link to QGIS, PostGIS, Python or web-based map services.

- **Long-standing interest in programming.**

Intro. Why we switched to R

Clementine:

Because of my postdoc project:

“To analyse the sensitivity of agglomeration economies to city definitions, I had to compute the same regression coefficient and make the same map for 4929 different ways of defining cities.
Doing it by hand was not just a waste of time… it was not possible!”

- **Open source / free**

When you change institutions, you do not have to check that they can provide you with licenses, etc.

- **Big investment but cumulative returns**

You recycle lines of codes all the time.

Intro. Why R for Urbanism?

- Programming language to **automate** repetitive tasks (tests, updates, generation of similar outputs, etc.)
- **Integrated** software for Statistics, GIS, Survey processing (APIs), Network analysis, Mapping, etc.
- Can produce interactive notebooks, slides and applications as **Reproducible archive** for sharing and tracing analyses for others and your future self

Part 1. An ecosystem in development

“R is an integrated suite of software facilities for data manipulation, calculation and graphical display.”

https://www.r-project.org/about.html

It is also sometimes called an **environment** or an **ecosystem** because it works with **packages** which play a different role along the data analysis **workflow**.

A package comprises **functions** to perform specific **tasks** (e.g. mapping, computing segregation indices, transforming table formats, reading files, etc.).

They are contributed by programmers and amateurs, and rely on similar **conventions** (formats, documentation etc.) to be **compatible** and **complementary**.

Part 1. An ecosystem in development

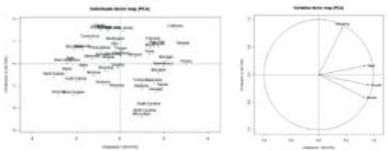
Statistics

R's original *raison d'être* is to process data and produce **statistical analysis** efficiently and soundly, with a **common language** and an extensive **documentation**.
All statistical operation you are used to perform with other dedicated softwares (Excel, Stata, SPSS, etc.) can be done in R.

Frequently used packages for statistics:

- **'stats'** for chi-square test, analysis of variance etc.
- **'lm'** for linear regression
- **'tidymodels'** for modelling and machine learning within the 'tidy' framework
- **'FactoMineR'** for factorial analysis (pca etc.)

ex. `result <- PCA(mydata)`



Part 1. An ecosystem in development

GIS

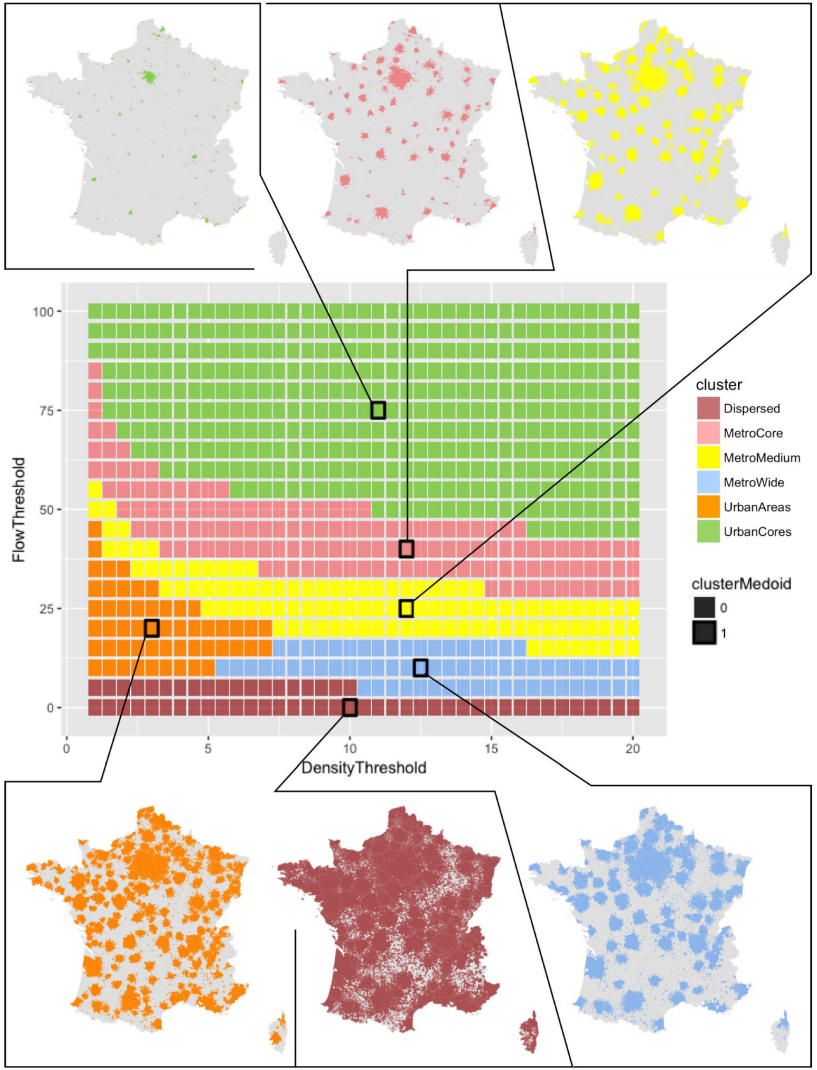
For Urbanists, R is used more and more often as a GIS to:

- **represent spatial data**
- **modify geometries** (fuse polygons, remove lines, add points)
- **conduct spatial analysis** (accessibility, visibility, etc)

Frequently used packages for GIS:

- **‘sf’** to handle geometries as a variable
- **‘rgeos’** for topology operations on geometries
- **‘rgdal’** to handle projections and make transformations
- **‘stars’** for spatial-temporal analyses

Part 1. An ecosystem in development



Part 1. An ecosystem in development

Visualisation and Maps (final and interactive)

Once your data is processed and analysed, you can use the same environment to visualise it.

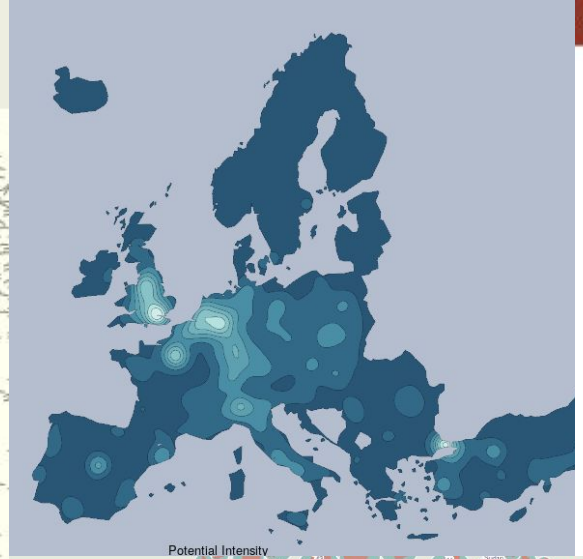
This is true for **graphs** in multiple dimensions, as well as **maps**, including beautiful and **interactive** maps!

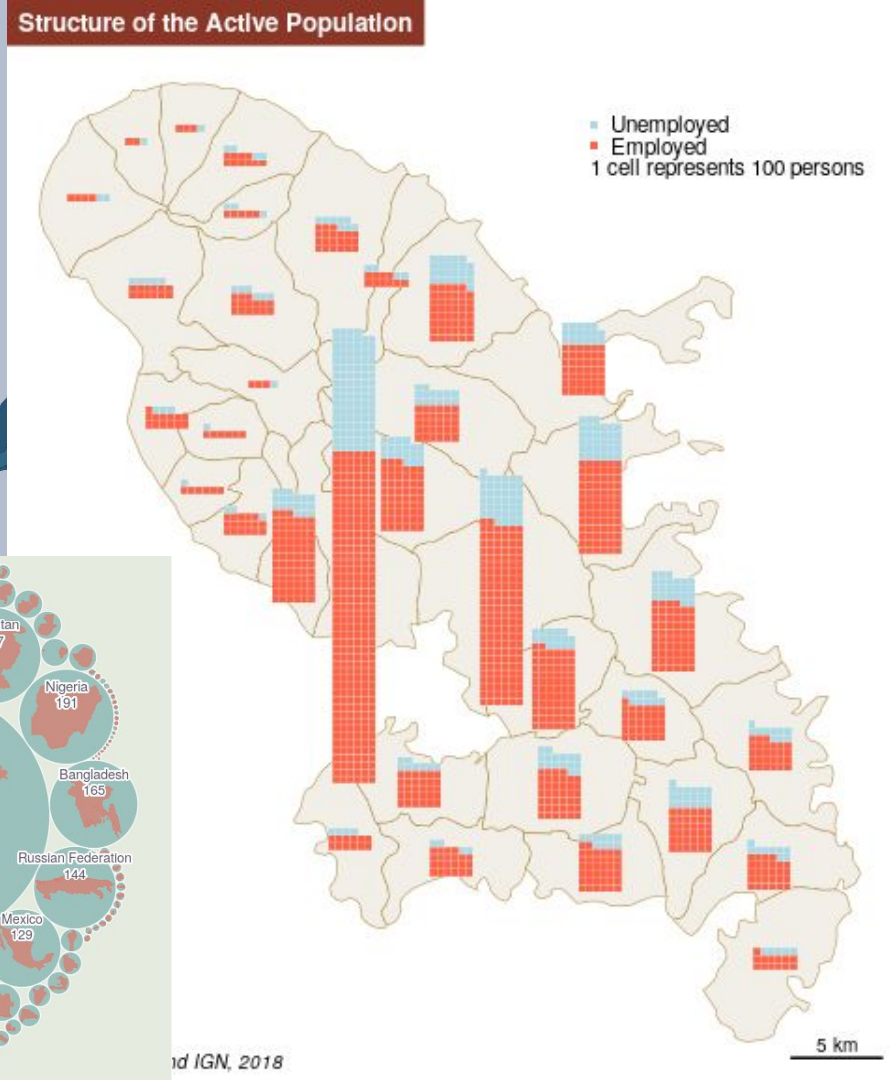
You can also extract maps from OpenStreetMap databases to generate your base maps.

Frequently used packages for Visualisation and mapping:

- **‘ggplot2’** for data visualisation
- **‘cartography’** for final mapping
- **‘mapsf’** for mapping of sf objects
- **‘linemap’** for line maps
- **‘leaflet’** for interactive maps
- **‘osmextract’** and **‘osmdata’** for osm data

https://rgeomatic.hypotheses.org/category/r-en





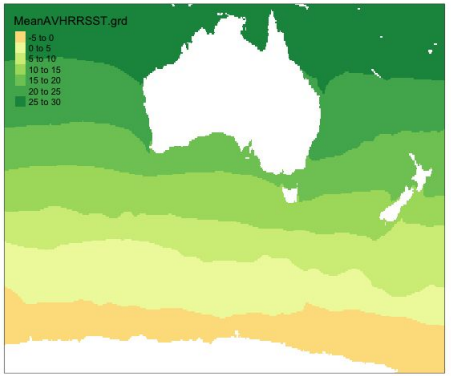
Part 1. An ecosystem in development

Raster data

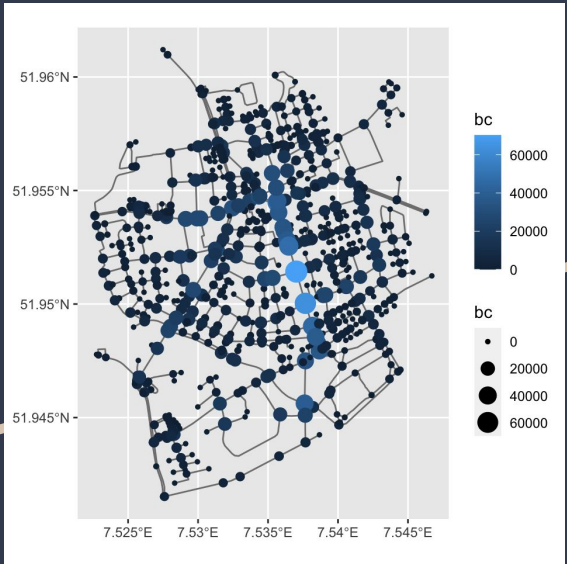
Maps are not always vectorial, you can be working with **raster** images (i.e. pixels of information), for instance land cover databases and satellite imagery, simulation model outputs, etc.

Frequently used packages for raster analysis:

- **‘raster’** for the creation, manipulation, intersection and modelling of spatial data.
- **‘terra’** for data manipulation of vector and raster spatial data, including interpolation and regression methods of very large data files.



Part 1. An ecosystem in development



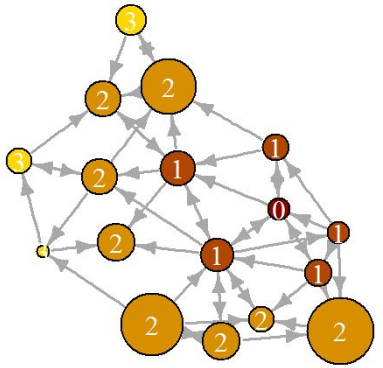
https://luukvdmeer.github.io/sfnetworks/articles/intro.html

Network analysis

Your data may be in the form of matrices and include network relations. You can use R to analyse **graphs** and **networks**, compute network metrics (centrality, connectivity etc.) as well as visualise them (network of actors and stakeholders, of streets, of citations, of material flows, etc.).

Frequently used packages for network analysis:

- **'igraph'** for network analysis and visualisation
- **'tidygraph'** to match igraph functions with the tidyverse framework
- **'sfnetworks'** to match igraph functions with the sf framework



https://kateto.net/netscix2016.html

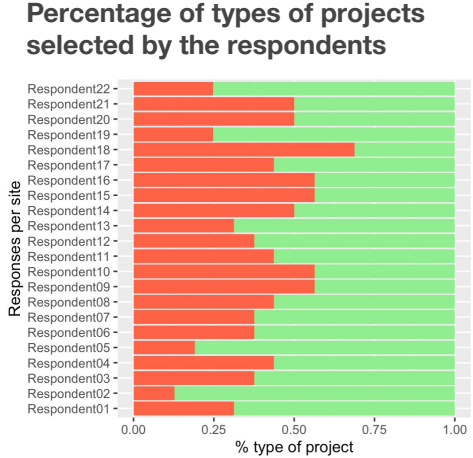
Part 1. An ecosystem in development

Survey processing

While qualitative analysis is limited in R, survey processing of quantitative survey data (e.g. Likert-scale questions) is possible and quite effective. You can extract questions with quantitative variables (either continuous or categorical) and analyse it in R.

Packages for survey processing:

- for instance, the **‘qualtRics’** package can be used to extract quantitative data from Qualtrics surveys automatically



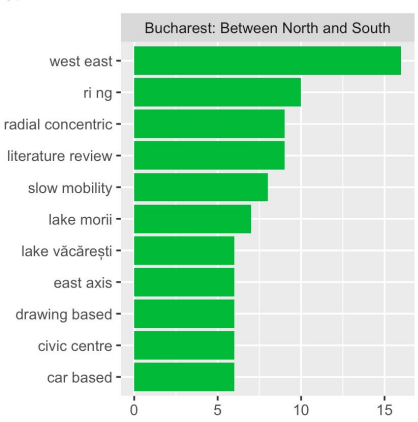
Part 1. An ecosystem in development

Text analysis

R is also increasingly used for text analysis (mining) applications, including topic modeling. One can analyse a large corpus of texts (e.g., hundreds of books) to extract patterns, analyse sentiments, etc. In urbanism, one can use it to extract and model main topics from a body of literature, or a series of reports, or theses.

Frequently used packages for text analysis:

- **‘tidytext’** for text analysis compatible with tidy data
- **‘tm’** package for text mining
- **‘quanteda’** for managing and analyzing textual data



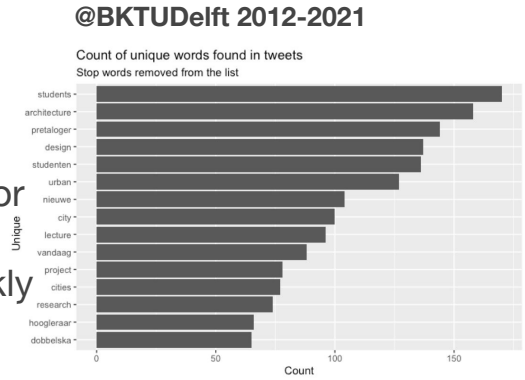
Part 1. An ecosystem in development

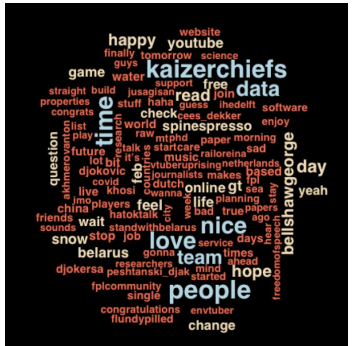
Processing Twitter Data

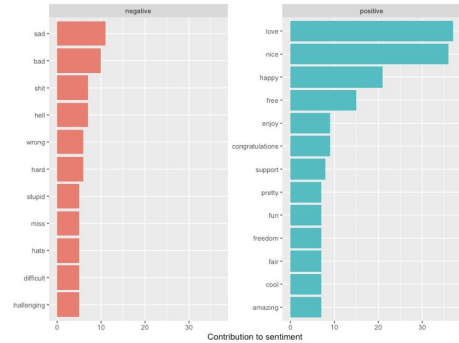
Collecting data from the Twitter API (for free) can provide insight about an event or a user.

Frequently used packages for Twitter data collection:

- **'rtweet'** or **'twittR'** for getting twitter data
- **'wordcloud'** to quickly visualise patterns in text data







Part 1. An ecosystem in development

Interacting with the Web

Working with data often involves interacting with Application Programming Interfaces, or **APIs,** (e.g., the QuickOSM QGIS plugin uses the Overpass API to download OSM data to QGIS).

Sometimes, when data is not made available via APIs or direct download, **web scraping** (that is getting data directly from the HTML of web pages) can be an effective alternative.

Frequently used R packages:

- **‘osmdata’** is an example of a package working with an API to access the OSM Overpass API
- **‘rvest’** most commonly used for web scraping
- Note that web scraping should be done with care and as a last resort

Part 1. An ecosystem in development

“R is an integrated suite of software facilities for data manipulation, calculation and graphical display.”

https://www.r-project.org/about.html

But it can also be used for one job and be linked to other software in a workflow.

Ex.
> import data of different formats
> use R scripts in QGIS
> use R with Python (it's not an either/or)
> use R for the quantitative components of a research and other software for qualitative data analysis (e.g., Atlas.ti or NVivo for QDA)
>...

Part 2. Archiving for yourself and others

By writing **code** rather than **clicking** buttons to perform data operations, you leave a trace.

Before you know it, you ensure that your operations are **archived, repeatable,** and **shareable** for **your future self** and **others**.

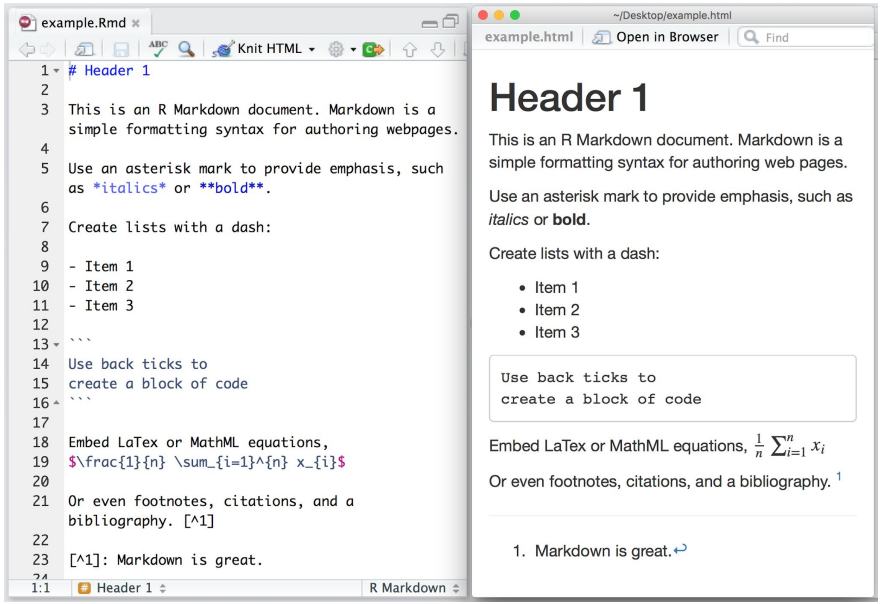
Possible usage of this archive:

- **R script** stored locally for your future use
- **R Markdown** to generate documentation as an html page, a pdf document or text file. (ex. to accompany a research article)
- **R Shiny application** to let the user explore your data and some parameters of your analysis
- **R package** to bundle and share your work

R Markdown

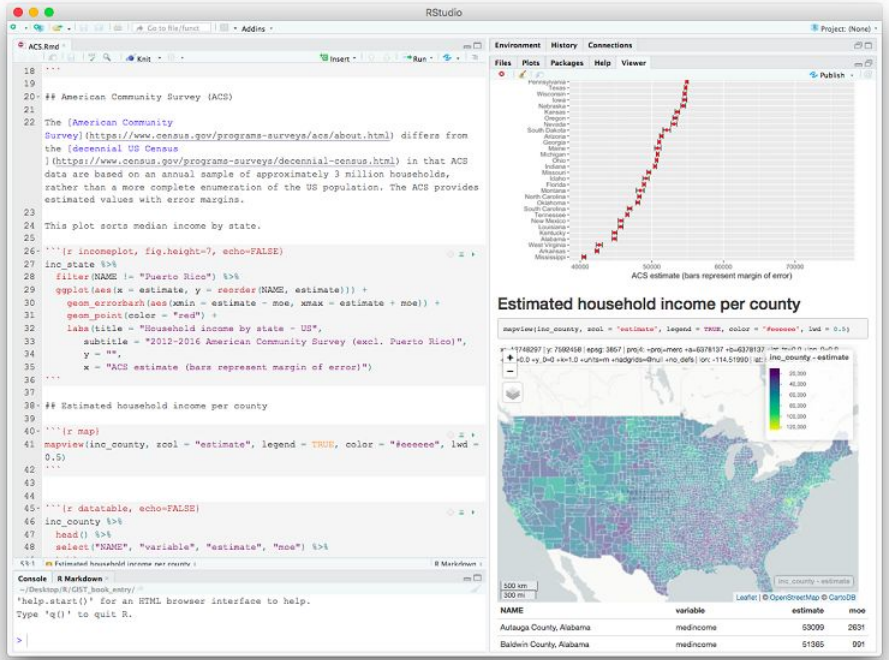
What is markdown and R Markdown?

- Markdown is a simple, readable plain text format



Source: https://rmarkdown.rstudio.com/authoring_quick_tour.html

R Markdown

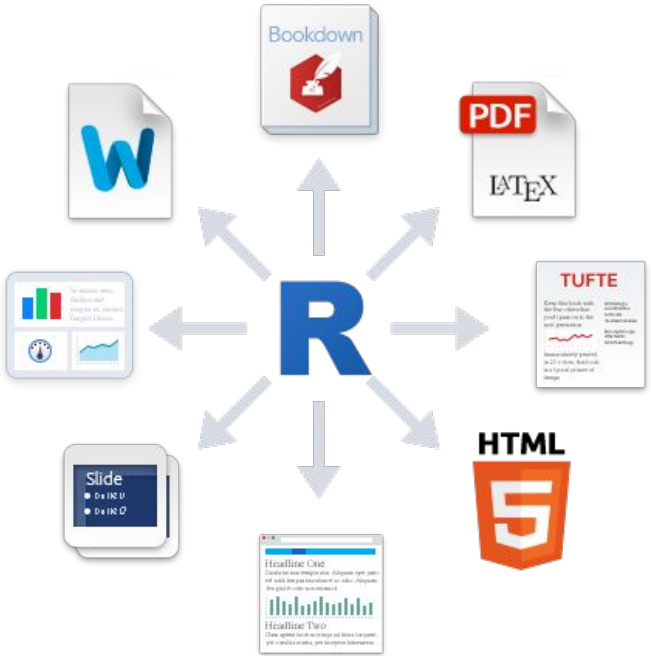


Source: https://gistbok.ucgis.org/bok-topics/r-geospatial-analysis-and-mapping

What is markdown and R Markdown?

- Markdown is a simple, readable plain text format
- R Markdown combines narrative written in Markdown with code

R Markdown

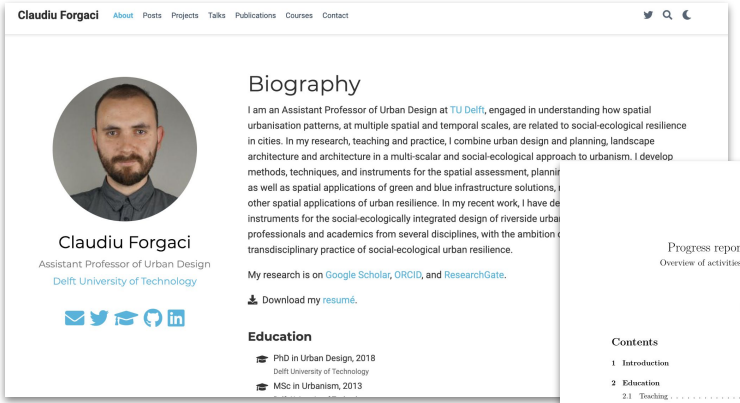


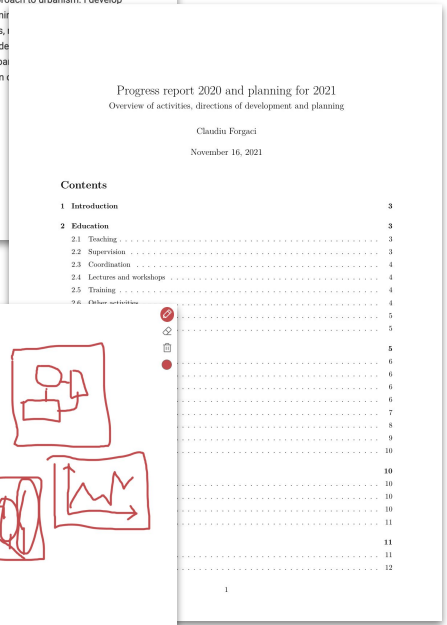
Source: https://rmarkdown.rstudio.com/lesson-13.html

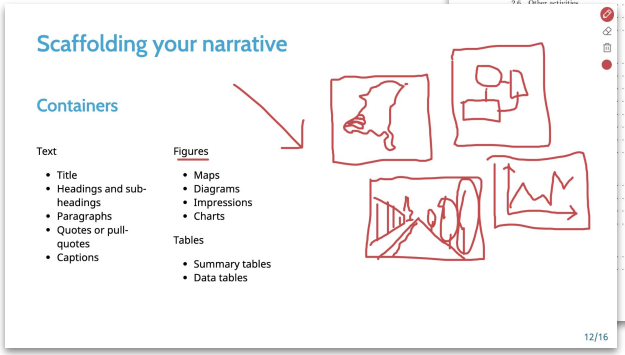
What is markdown and R Markdown?

- Markdown is a simple, readable plain text format
- R Markdown combines narrative written in Markdown with code
- Powered by Pandoc, it can be exported (or 'knitted') in various formats (PDF, HTML, DOCX, etc.)

R Markdown







What is markdown and R Markdown?

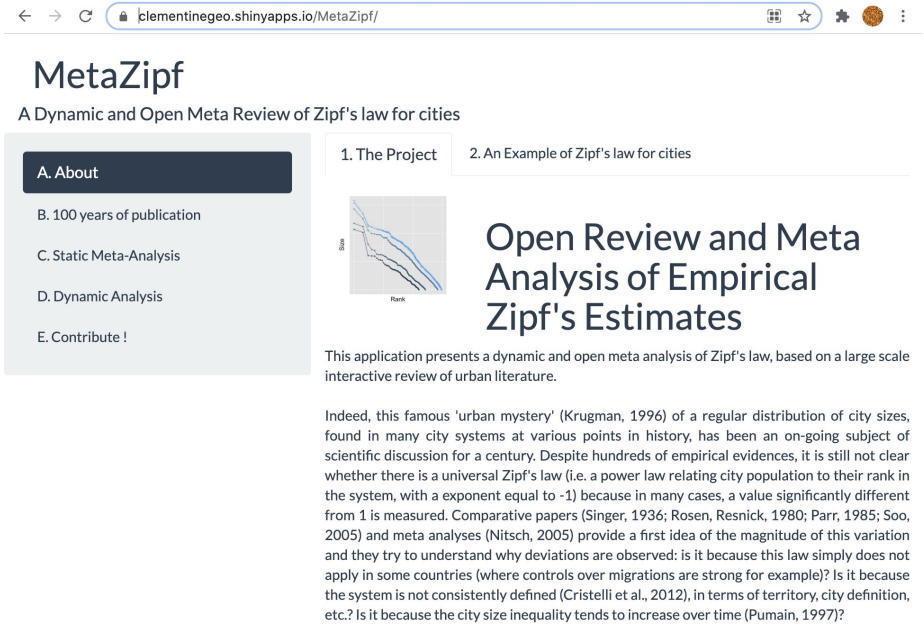
- Markdown is a simple, readable plain text format
- R Markdown combines narrative written in Markdown with code
- Powered by Pandoc, it can be exported (or 'knitted') in various formats (PDF, HTML, DOCX, etc.)
- Other outputs include: slides, websites, course material in HTML format, and data-driven reports

R Markdown

Advantages for research and education:

- Transparency and open access to code
- Archive of data and methods
- Companion document to articles
- It enhances reproducibility
- For Python users, Jupyter Notebooks are the alternative (but R can also be run in Jupyter notebooks, and vice versa)

Shiny apps



https://clementinegeo.shinyapps.io/MetaZipf/

What is Shiny and R Shiny apps?

- **‘shiny’** is a R package which allows to create interactive apps, interactive documents or interactive presentation slides.
- **Shiny apps** work like most applications.
- They are composed of:
  - a front end (‘user interface’) which you program to display information and graphical elements which are allowed to vary
  - a back end (‘server’) which contains the code to run when buttons are clicked or sliders slided.

Shiny apps

https://clementinegeo.shinyapps.io/MetaZipf/

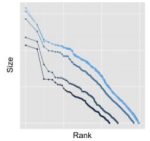
clementinegeo.shinyapps.io/MetaZipf/

MetaZipf

A Dynamic and Open Meta Review of Zipf's law for cities

- A. About
- B. 100 years of publication
- C. Static Meta-Analysis
- D. Dynamic Analysis
- E. Contribute !

1. The Project | 2. An Example of Zipf's law for cities



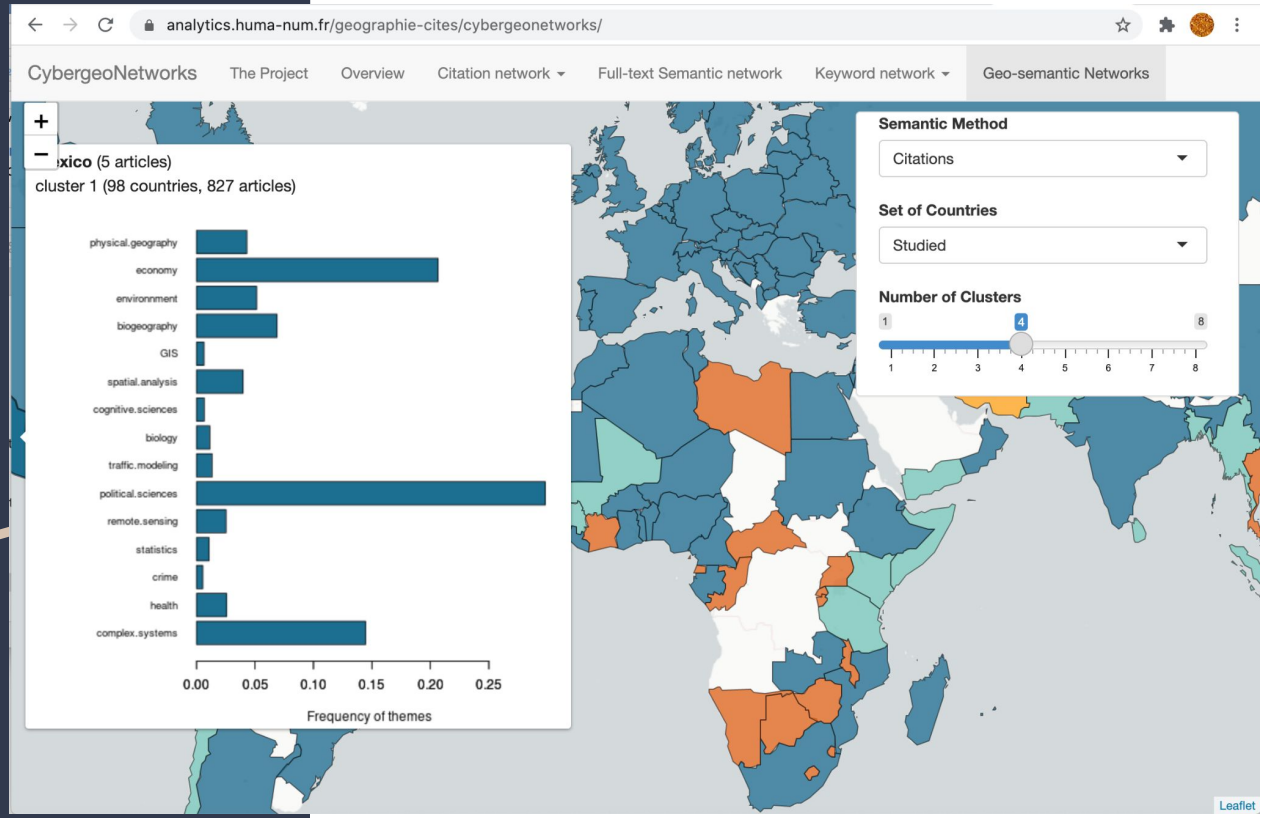
Open Review and Meta Analysis of Empirical Zipf's Estimates

This application presents a dynamic and open meta analysis of Zipf's law, based on a large scale interactive review of urban literature.

Indeed, this famous 'urban mystery' (Krugman, 1996) of a regular distribution of city sizes, found in many city systems at various points in history, has been an on-going subject of scientific discussion for a century. Despite hundreds of empirical evidences, it is still not clear whether there is a universal Zipf's law (i.e. a power law relating city population to their rank in the system, with a exponent equal to -1) because in many cases, a value significantly different from 1 is measured. Comparative papers (Singer, 1936; Rosen, Resnick, 1980; Parr, 1985; Soo, 2005) and meta analyses (Nitsch, 2005) provide a first idea of the magnitude of this variation and they try to understand why deviations are observed: is it because this law simply does not apply in some countries (where controls over migrations are strong for example)? Is it because the system is not consistently defined (Cristelli et al., 2012), in terms of territory, city definition, etc.? Is it because the city size inequality tends to increase over time (Pumain, 1997)?

Shiny apps

https://analytics.huma-num.fr/geographie-cites/cybergeonetworks/



Shiny apps

Advantages for research and education:

- Reproducibility of the analysis
- Sensitivity analysis
- User engagement
- Easy access to data (filtering, search engine etc.)

Demonstration

Fill in the survey

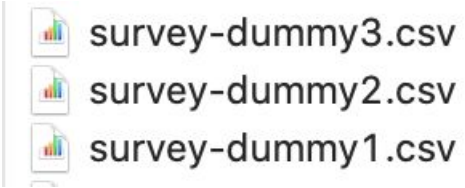
https://docs.google.com/forms/d/1BS62tO0e6DtsaoETiiJkpHGqd5DtN0AtgFzftfGWgPE/edit

And let’s have a 15min break!

Demonstration: input files

Let's say we have a **survey** about people's longevity at TU Delft, interest in R and last place of residence.

We collected **3 answers** to this survey, one in each file, which look like this:



We now want to analyse our results, but we know that **there is always one person who will send their file late**, so we might have to redo it with another file.

Let's use R!

Demonstration: input files

Let's say we have 3 answers to this survey, which look like this:

	A	B
1	Name	Sjoerd
2	Year of arrival at TU Delft	2014
3	City/Place you were living in before TT	Delft NL
4	Level of interest for R (1=low/5=high)	2
5	Have you used a programming language before (Y/N)	N
6		

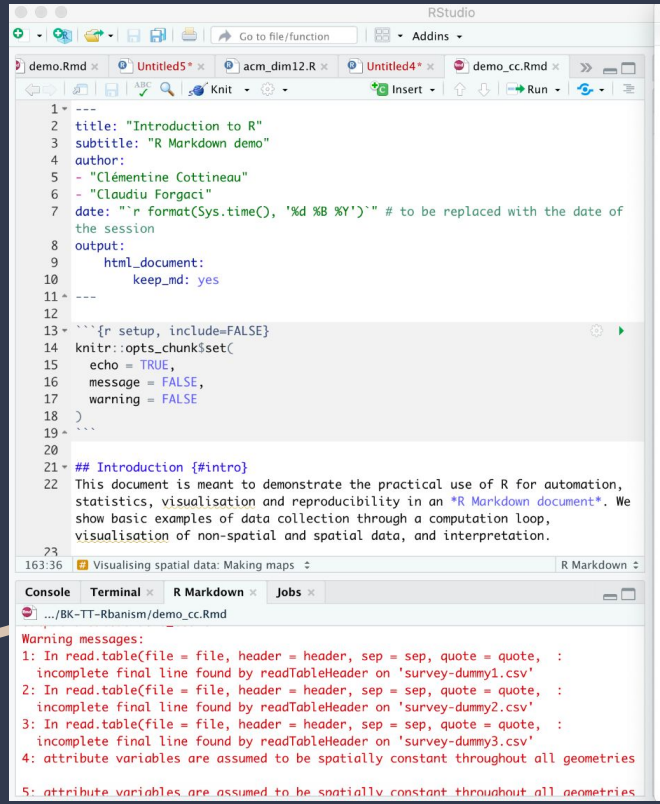
	A	B
1	Name	Marina
2	Year of arrival at TU Delft	2019
3	City/Place you were living in before TT	Bucharest
4	Level of interest for R (1=low/5=high)	5
5	Have you used a programming language before (Y/N)	Y
6		

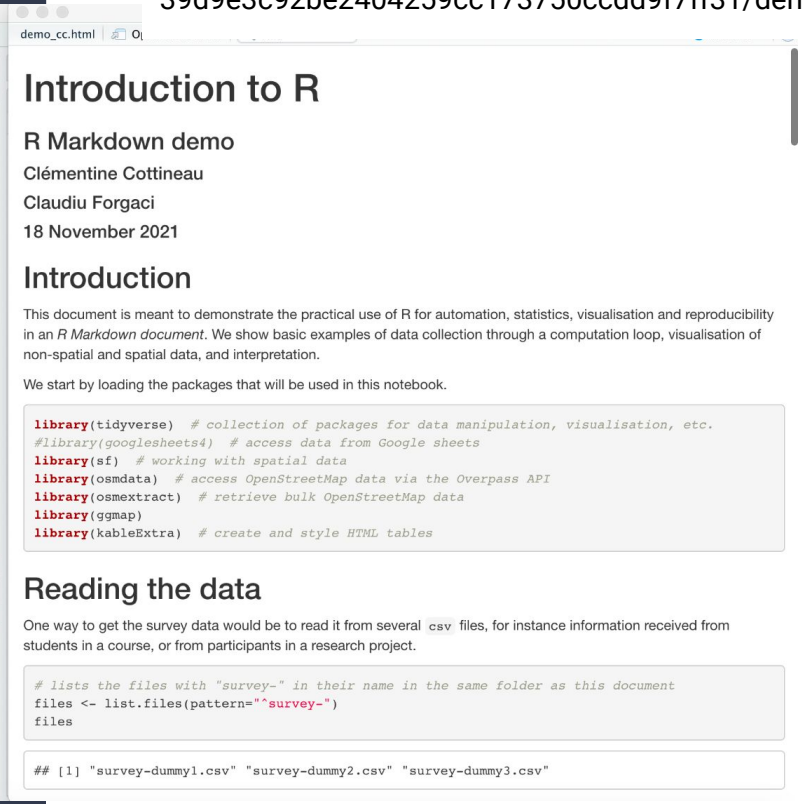
	A	B
1	Name	Damien
2	Year of arrival at TU Delft	2020
3	City/Place you were living in before TT	Paris, France
4	Level of interest for R (1=low/5=high)	4
5	Have you used a programming language before (Y/N)	N

Demonstration

We analyse & archive them at the same time, using a R Markdown document.

https://github.com/ClementineCttn/BK-TT-Rbanism/blob/a9d39d9e3c92be2404259cc173750ccdd9f7ff31/demo_cc.html





Demonstration: libraries

A script tends to always start like this: a list of 'library' command.

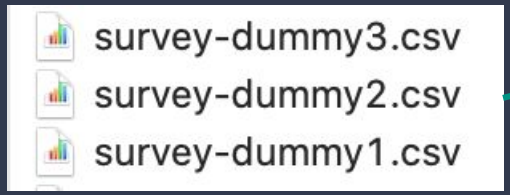
Each library corresponds to a package, or a collection of pre-defined functions.

ex. 'osmdata' contains Open Street Map data, 'sf' has functions to work with spatial data, etc.

We start by loading the packages that will be used in this notebook.

```
library(tidyverse)  # collection of packages for data manipulation, visualisati
on, etc.
#library(googlesheets4)  # access data from Google sheets
library(sf)  # working with spatial data
library(osmdata)  # access OpenStreetMap data via the Overpass API
library(osmextract)  # retrieve bulk OpenStreetMap data
library(ggmap)
library(kableExtra)  # create and style HTML tables
```

Demonstration: reading inputs



Reading the data

One way to get the survey data would be to read it from several `csv` files, for instance information received from students in a course, or from participants in a research project.

```
# lists the files with "survey-" in their name in the same folder as this docum
ent
files <- list.files(pattern="^survey-")
files
```

So that late files are read too!

```
## [1] "survey-dummy1.csv" "survey-dummy2.csv" "survey-dummy3.csv"
```

For this, we will use a 'for-loop', i.e. a simple recursive function very useful to automate operations.

```
# creates an empty table
survey <- tibble()

n <- length(files)
# the 'for-loop':
for (i in 1:n){
  f <- read.csv2(files[i], header=F)
  if(i == 1){
    survey <- f
  } else {
    survey <- cbind(survey, f[,2])
  }
}
```

Demonstration: a for-loop

Reading the data

One way to get the survey data would be to read it from several `csv` files, for instance information received from students in a course, or from participants in a research project.

```
# lists the files with "survey-" in their name in the same folder as this docum
ent
files <- list.files(pattern="^survey-")
files
```

```
## [1] "survey-dummy1.csv" "survey-dummy2.csv" "survey-dummy3.csv"
```

For this, we will use a 'for-loop', i.e. a simple recursive function very useful to automate operations.

```
# creates an empty table
survey <- tibble()

n <- length(files)
# the 'for-loop':
for (i in 1:n){
  f <- read.csv2(files[i], header=F)
  if (i == 1){
    survey <- f
  } else {
    survey <- cbind(survey, f[,2])
  }
}
```

Demonstration: Combined table

Reading the data

One way to get the survey data would be to read it from several `csv` files, for instance information received from students in a course, or from participants in a research project.

```
# lists the files with "survey-" in their name in the same folder as this docum
ent
files <- list.files(pattern="^survey-")
files
```

```
## [1] "survey-dummy1.csv" "survey-dummy2.csv" "survey-dummy3.csv"
```

For this, we will use a 'for-loop', i.e. a simple recursive function very useful to automate operations.

```
# creates an empty table
survey <- tibble()

n <- length(files)
# the 'for-loop':
for (i in 1:n){
  f <- read.csv2(files[i], header=F)
  if(i == 1){
    survey <- f
  } else {
    survey <- cbind(survey, f[,2])
  }
}
```

This chunk of code reads all the files.
Its result: a combined table

Demonstration: Combined table

This chunk of code contains basic functions to reorganise data and variables

Its result: a combined table in a usable format.

```
transpose <- as.data.frame(t(survey))
colnames(transpose) <- c("Name", "ArrivalYearTU", "CityBefore", "InterestR", "E
xperience")
survey_together <- transpose[-1,]
rownames(survey_together) <- 1:n
 survey_together$ArrivalYearTU <- as.numeric(survey_together$ArrivalYearTU)
 survey_together$InterestR <- as.numeric(survey_together$InterestR)

 # as with spreadsheets, we can create new variables
 survey_together$SeniorityAtTUD <- 2021 - survey_together$ArrivalYearTU
survey_together
```

```
##      Name ArrivalYearTU    CityBefore InterestR Experience SeniorityAtTUD
## 1 Sjoerd          2014      Delft NL         2          N             7
## 2 Marina          2019     Bucharest         5          Y             2
## 3 Damien          2020 Paris, France         4          N             1
```

It still goes quicker to do it by hand for 3 files, but when the number of files increases (or when files change), the code is the same, and this is when it makes sense to automate the process.

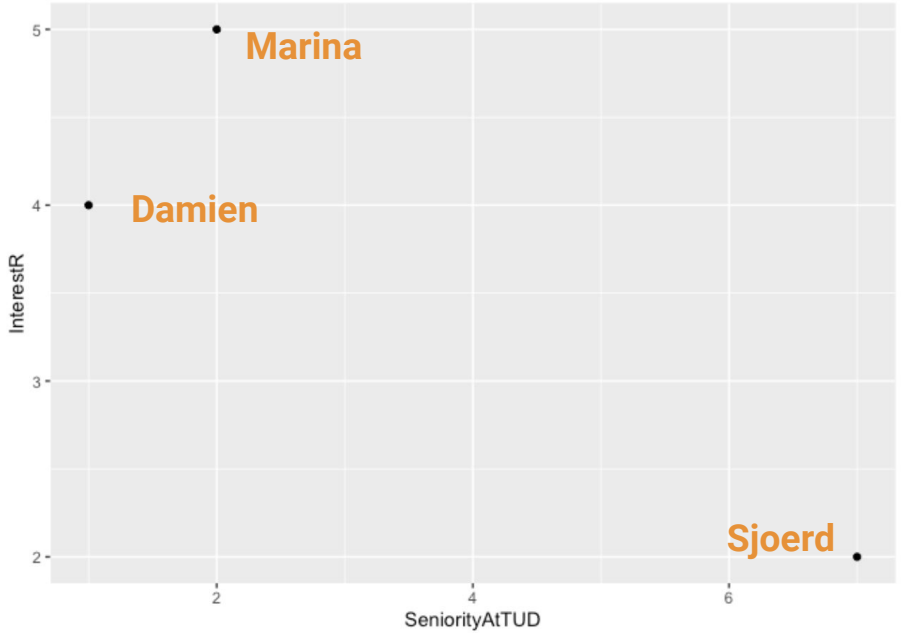
Demonstration: graphs

```
##      Name ArrivalYearTU    CityBefore InterestR Experience SeniorityAtTUD
## 1 Sjoerd          2014      Delft NL         2          N              7
## 2 Marina          2019     Bucharest         5          Y              2
## 3 Damien          2020 Paris, France         4          N              1
```

Visualising non-spatial data: Making graphs

Let's visualise some non spatial data, for instance the relation between the semiority at TU Delft and the reported interest in R.

```
library(ggplot2)

q <- ggplot(survey_together, aes(x=SeniorityAtTUD, y=InterestR))
q+ geom_point()
```

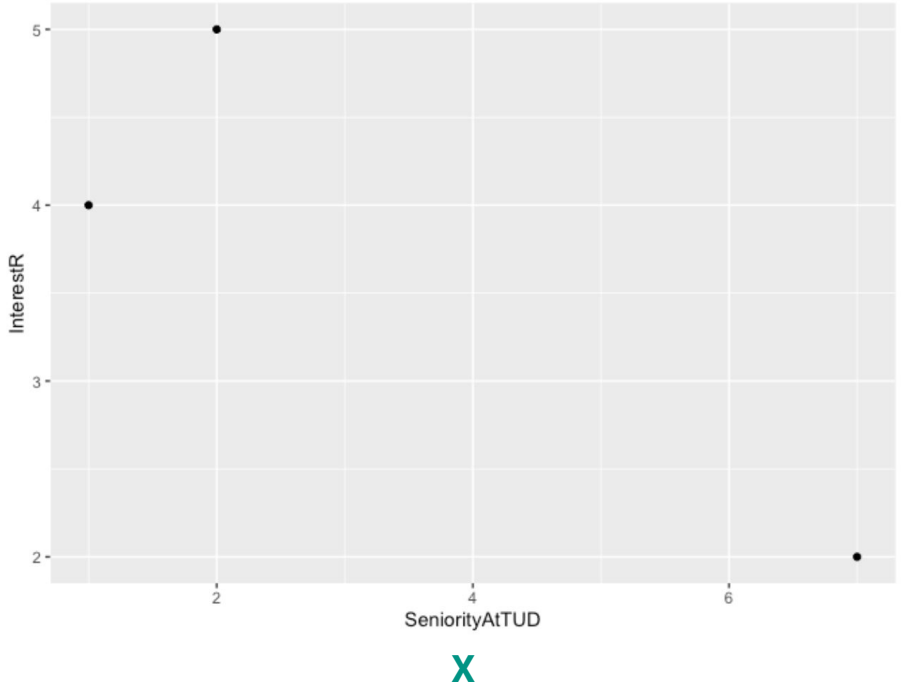


Demonstration: graphs

Visualising non-spatial data: Making graphs

Let's visualise some non spatial data, for instance the relation between the semiority at TU Delft and the reported interest in R.

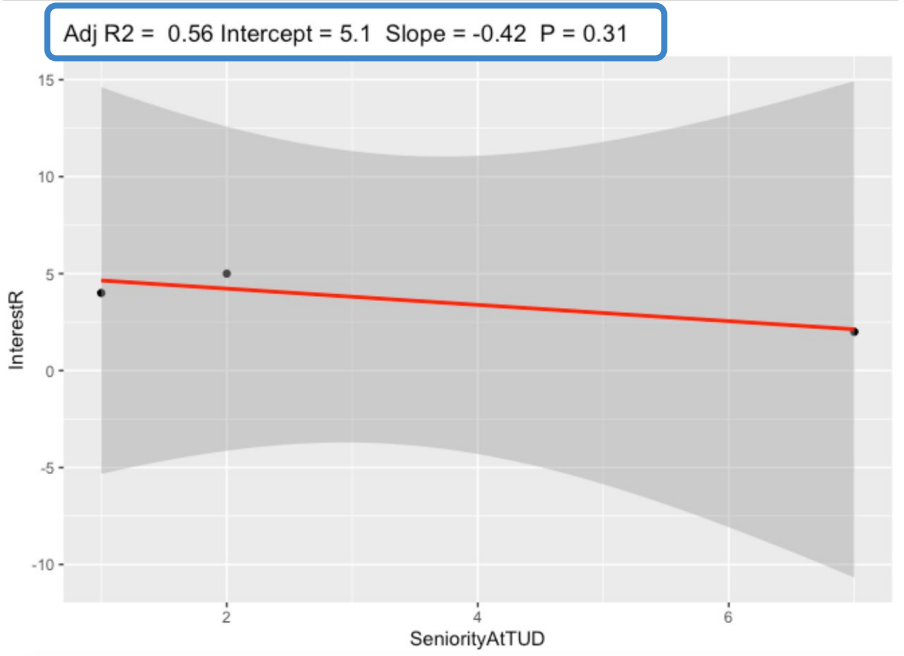
```
library(ggplot2)

q <- ggplot(survey_together, aes(x=SeniorityAtTUD, y=InterestR))
q+ geom_point()
```



Demonstration: regression model

We can even estimate a linear regression between the two and visualise the result.

```
fit <- lm(InterestR ~ SeniorityAtTUD, data = survey_together)

ggplot(fit$model, aes_string(x = names(fit$model)[2], y = names(fit$model)[1]))
 +
  geom_point() +
  stat_smooth(method = "lm", col = "red") +
  labs(title = paste("Adj R2 = ",round(signif(summary(fit)$adj.r.squared, 2)  d
igits = 3),

                     "Intercept =",signif(fit$coef[[1]],2 ),
                     " Slope =",signif(fit$coef[[2]], 2),
                     " P =",signif(summary(fit)$coef[2,4], 2)), cex=2)
```



Demonstration: mapping tools

‘osm’ packages
‘leaflet’ package

another for-loop to map each of the cities mentioned in the survey answers

It is actually looking through osm.org!

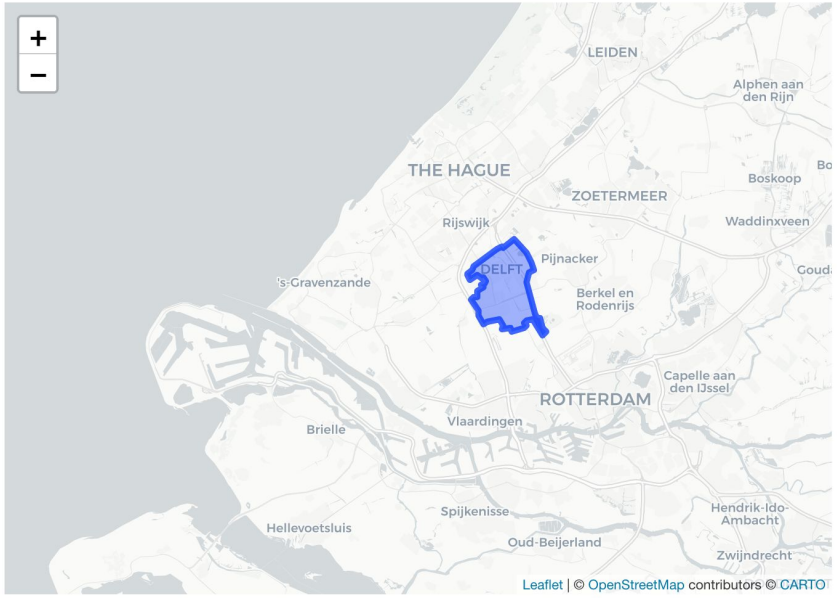
Visualising spatial data: Making maps

```
library(sf)
library(osmextract)
library(osmdata)
library(leaflet)

# get places from csv survey
places <- survey_together[, "CityBefore"]

for (i in 1:n){
## get municipal boundary from osmdata
bb <- osmdata::getbb(
            place_name = places[i],
            format_out = 'sf_polygon',
            silent = FALSE
        )
  assign(paste0("bb", i), bb)
  if(i == 1){
    bbx <- bb
  } else{
    bbx <- rbind(bbx, bb)
  }
}
```

```
## [1] "https://nominatim.openstreetmap.org/?q=Delft%20NL&featuretype=settlemen
t&polygon_text=1&format=json&limit=10"
## [1] "https://nominatim.openstreetmap.org/?q=Bucharest&featuretype=settlement
&polygon_text=1&format=json&limit=10"
## [1] "https://nominatim.openstreetmap.org/?q=Paris%2C%20France&featuretype=se
ttlement&polygon_text=1&format=json&limit=10"
```

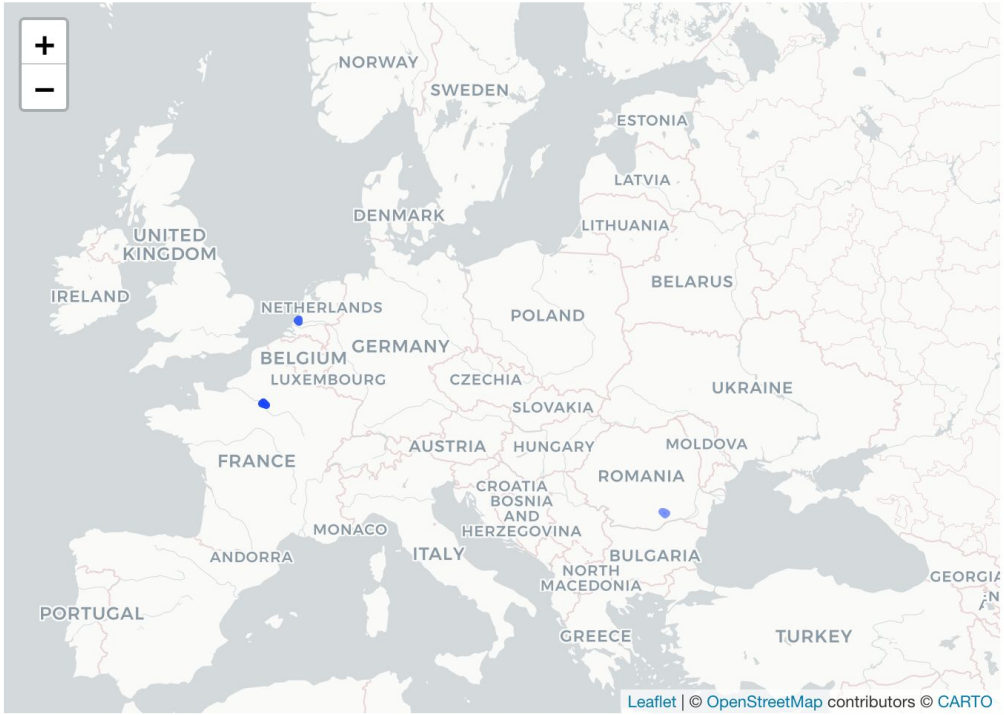
Demonstration: Interactive maps

```
#map places on interactive map
    leaflet(bbx) %>% addProviderTiles("CartoDB.Positron") %>%
      addPolygons()
```



An interactive map with only 2 lines of code :)

```
#map places on interactive map
    leaflet(bbx) %>% addProviderTiles("CartoDB.Positron") %>%
      addPolygons()
```



Demonstration: Automated maps

```
## create a function to extract the streets of a city x
map_city_streets <- function(x){
osm_lines = oe_get(x, stringsAsFactors = FALSE, quiet = TRUE)
nrow(osm_lines)
ht <- c("primary", "secondary", "tertiary", "unclassified") # highway types of interest
bb <- osmdata::getbb(
            place_name = x,
            format_out = 'sf_polygon',
            silent = FALSE
        )
osm_major_roads_place <- osm_lines[osm_lines$highway %in% ht, ] %>%
    sf::st_intersection(bb)

plot(osm_major_roads_place)

    }
```

The recipe

```
## let's map it for Rotterdam
map_city_streets("Rotterdam")
```

The application to Rotterdam

```
## [1] "https://nominatim.openstreetmap.org/?q=Rotterdam&featuretype=settlement&polygon_text=1&format=json&limit=10"
```

Demonstration: Automated maps

```
## create a function to extract the streets of a city x
map_city_streets <- function(x){
osm_lines = oe_get(x, stringsAsFactors = FALSE, quiet = TRUE)
nrow(osm_lines)
ht <- c("primary", "secondary", "tertiary", "unclassified") # highway types of interest
bb <- osmdata::getbb(
            place_name = x,
            format_out = 'sf_polygon',
            silent = FALSE
        )
osm_major_roads_place <- osm_lines[osm_lines$highway %in% ht, ] %>%
    sf::st_intersection(bb)

plot(osm_major_roads_place)

    }


## let's map it for Rotterdam
map_city_streets("Rotterdam")
```

The application to Rotterdam

```
## [1] "https://nominatim.openstreetmap.org/?q=Rotterdam&featuretype=settlement&polygon_text=1&format=json&limit=10"
```


```
map_city_streets(places[n])
```

The application to the last answer...

```
## [1] "https://nominatim.openstreetmap.org/?q=Paris%2C%20France&featuretype=settlement&polygon_text=1&format=json&limit=10"
```

Demonstration (google survey)

Conclusion

R for Urbanism. Why?

- Programming language to **automate** repetitive tasks (tests, updates, generation of similar outputs, etc.)
- **Integrated** software for Statistics, GIS, Survey processing (APIs), Network analysis, Mapping
- Can produce **interactive** notebooks, slides and applications as **reproducible archive** for sharing and tracing analyses for others and your future self

Going further

/!\ Courses at the TU:

- Data carpentry
- Software carpentry

We want to build an uRbanism community

- Interested PhD candidates, researchers, students, or any other colleagues from the field of urbanism who could benefit from R? Let them know! Feel free to share this presentation, send them to colleagues, or take on the challenge of learning R together with them.

- Let us or each other know if you need any guidance. Main highlights of the worldwide R community are active peer support, inclusion, and open science support. #rstats

- There is a recurring R Cafe at TU Delft with an growing and helpful community of R users. Let us know if you want to join.

TT Peer-Teaching Resources

https://docs.google.com/document/d/1kNRe3xIo5GxM_7dZxd_HqYV5w4oeiCeLFkK3d6fcKzw/edit?usp=sharing

Resources for teaching

- Textbook: Geocomputation with R

https://geocompr.robinlovelace.net/

This book by Robin Lovelace, Jakub Nowosad and Jannes Muenchow is fully accessible for non-commercial use. It provides an articulate presentation of R and detailed examples of its use in transportation, ecology and geomarketing. It goes through a range of analyses of geographical data using the sf package. It is well illustrated with maps and graphs and contains exercises.

- Granolarrr: R for Data Science

https://sdesabbata.github.io/granolarr/

This resource has been put together by Stefano de Sabbata from the Uni of Leicester UK. It is an introduction course to R, its use in data cleaning, analysis and machine learning. What is good about it is that it contains videos of the lectures as well as a practical coursebook with exercises, and that it is applied to spatial data.

- Tutorial on data visualisation with R

https://wilkelab.org/SDS375/syllabus.html

This resource has been put together by Claus Wilke from the Uni of Austin Texas. It is a data

TT Peer-Teaching Resources

Next session:

16/12/2021
Rodrigo Cardoso.
Setting up an elective course.