Geospatial Analysis using R

Introduction to data-driven decision making

Ana J. Alegre, Cristian Silva. iGISc. November 3, 2021.

Agenda

Workshop day 1

1. Introduction
2. Data-driven decision making
3. R language basics, package installing and RMarkdown usage
4. Hands-on session
5. Break
6. Plain and spatial data wrangling using R
7. Hands-on session
8. Break
9. Exploratory Data Analysis (EDA)
10. Hands-on session
11. Q&A

About us

Meet the presenters

Ana J. Alegre

jalegre@centrogeo.edu.mx

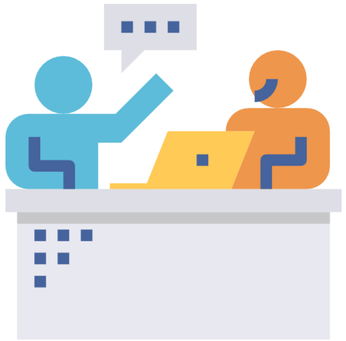
Cristian Silva

csilva@centrogeo.edu.mx

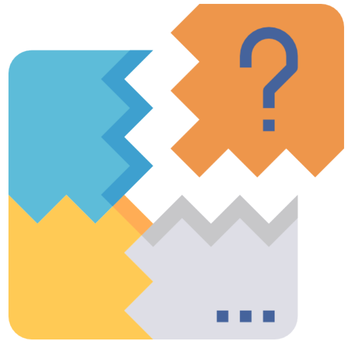
How this workshop will work?

Directions and recommendations

We will show some slides to explain the main concepts to be used during the workshop. After the keynote, we will perform the practice and comment it.



Please be attentive to the explanations given during the practice. **Try not to be distracted by taking notes during this period,** you will be able to perform the exercises by yourself at home anytime later.



Anytime you have a question, please write it in the chat window. At the end of the session we will be happy to answer all questions.

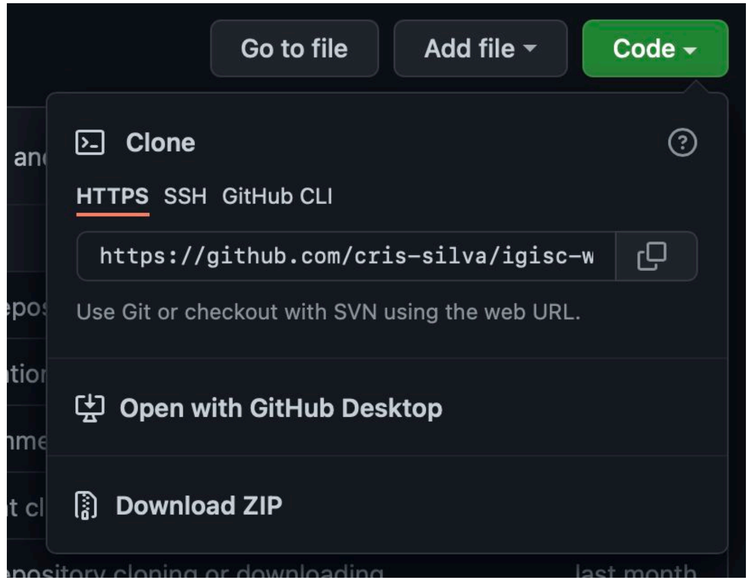
If possible, download the resources and run at home the interactive workbooks provided in the GitHub repository to practice what you have learn in the workshop. Write down any questions you have and we will answer them during tomorrow's session.

Icons made by Becris from www.flaticon.com

GitHub Repository

Resources used in this workshop

https://github.com/cris-silva/igisc-workshop_2021




```
$ git clone https://github.com/cris-silva/igisc-workshop_2021.git
```

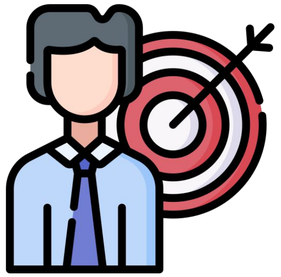
Data-driven decision making

“I never guess. It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.”

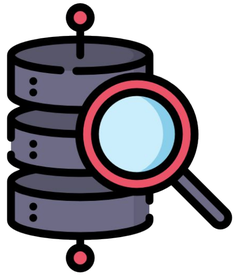
Sir Arthur Conan Doyle, Author of Sherlock Holmes stories

Data-driven decision making

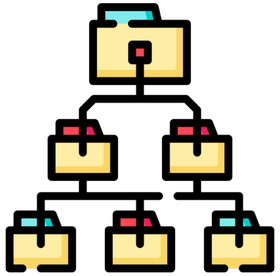
How to make data-driven decisions?



Know your mission



Identify data sources



Clean and organize data

Perform statistical analysis

Draw conclusions

References taken from https://www.northeastern.edu/graduate/blog/data-driven-decision-making/
Icons made by Pixel perfect from www.flaticon.com

What is spatial analysis?

Understanding geographic information

What is so special about spatial data, compared to just data?

Why they occur where they occur?

Context

Environment

Interaction

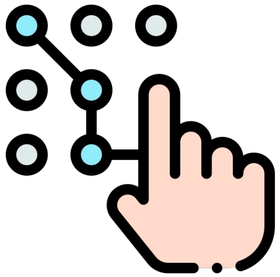
Value-added

What is spatial analysis?

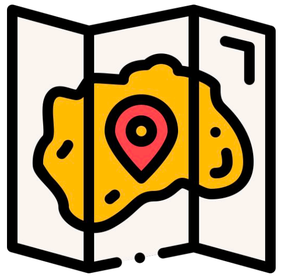
Understanding geographic information

Where things happen?

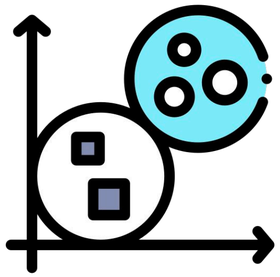
Application of analytical methods



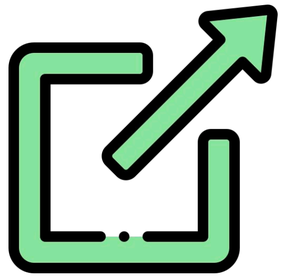
Patterns



Hotspot



Clusters



Outliers

Where they should be most optimally located?

Data-driven decision making

Decisions based on data instead of intuition/observation

Everything that happens in the world happens somewhere else

The decision-making process supported by Data Analytics and Geographic Information Systems (GIS) has proven to be much more efficient and faster

Find insights from data and transform them into actions and decisions

Territorial problems or to the optimization of resources involved in their use

- *Where should a hospital be located to provide a good service to the community?*
- *What is the optimal route for building a road?*
- *In which area should more resources be allocated after a disaster?*
- *How can I get somewhere faster?*
- *Which bank, restaurant, store is closest to me?*

R Language basics, package installing and RMarkdown usage

What is R?

The fundamentals of R programming

Comprehensive language

Open source

Wide array of packages

Handling all sorts of data

Various graphical libraries

Fast calculation

Large community of active users

Go-to language for statistics and data science

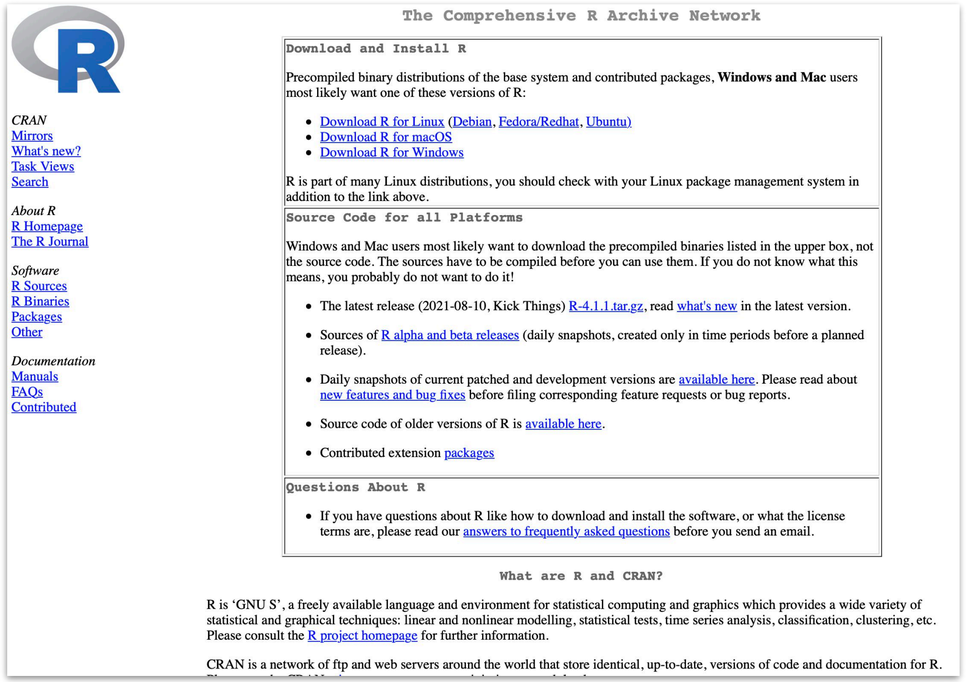
References taken from https://data-flair.training/blogs/features-of-r-programming/
Icons made by Pixel perfect from www.flaticon.com

AS SEEN BY USERS OF …

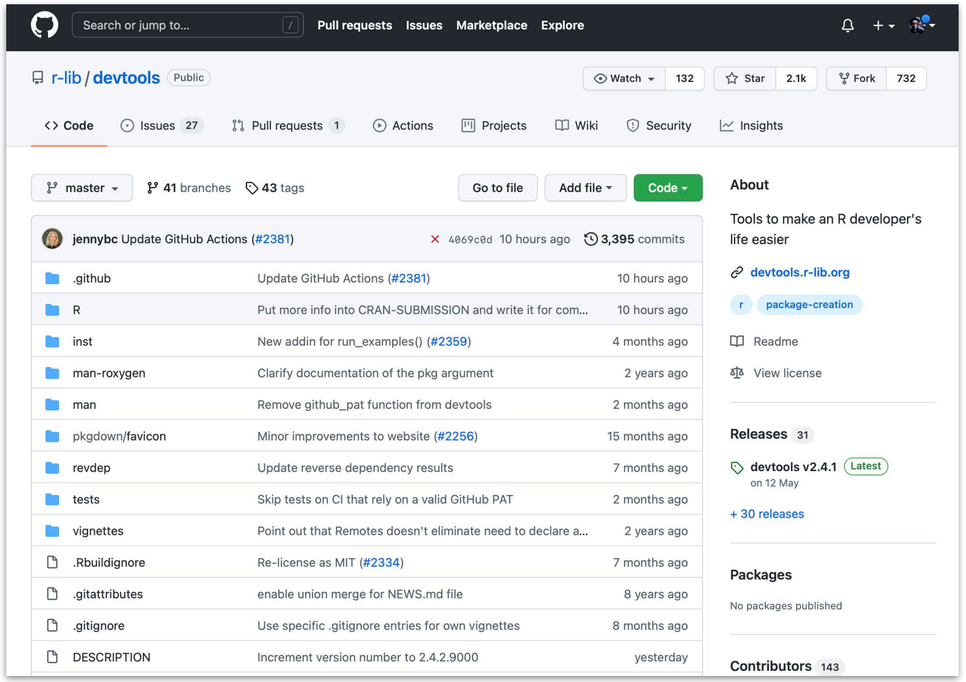
The CRAN & other sources

Where to download R and packages?

CRAN mirrors



GitHub repos




```
install.packages("package name")
```

```
devtools::install_github("package name")
```

Installing R & RStudio

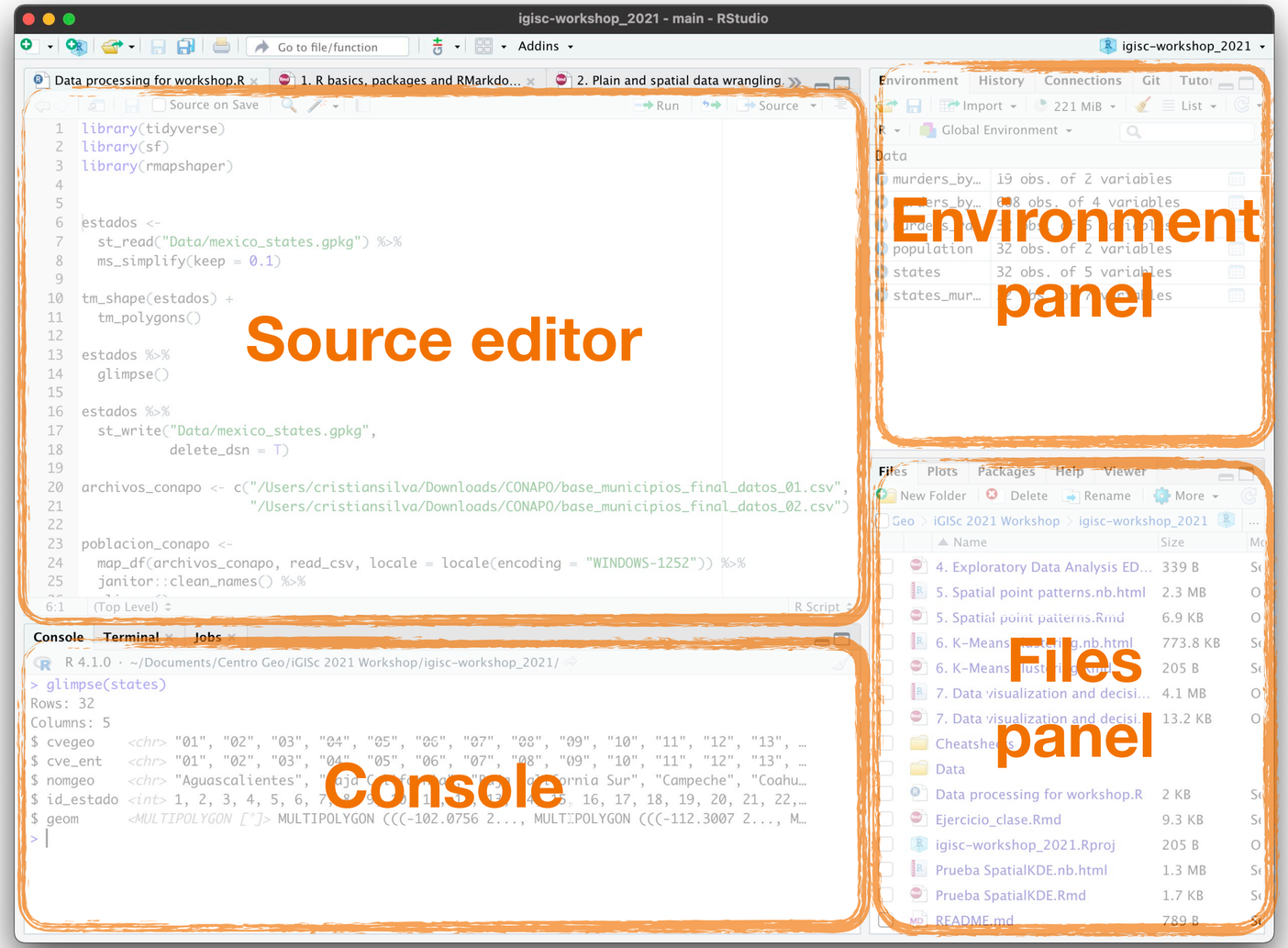
Setup environment for the workshop

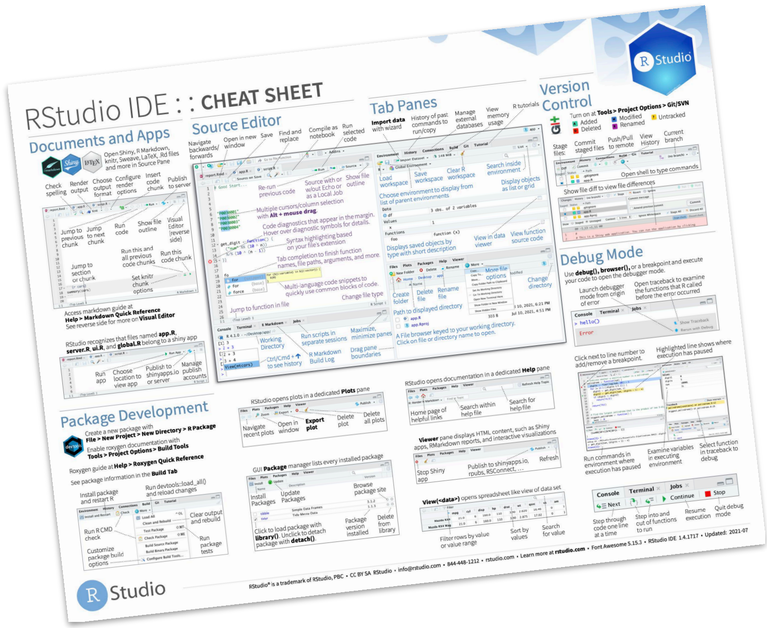
Download and install the latest version of **R** from https://cloud.r-project.org

Download and install the latest version of **RStudio Desktop** from https://www.rstudio.com/products/rstudio/

RStudio IDE

Quick review of user interface





For further reference, check out **RStudio IDE Cheatsheet** included in the GitHub repository or RStudio Help Menu.

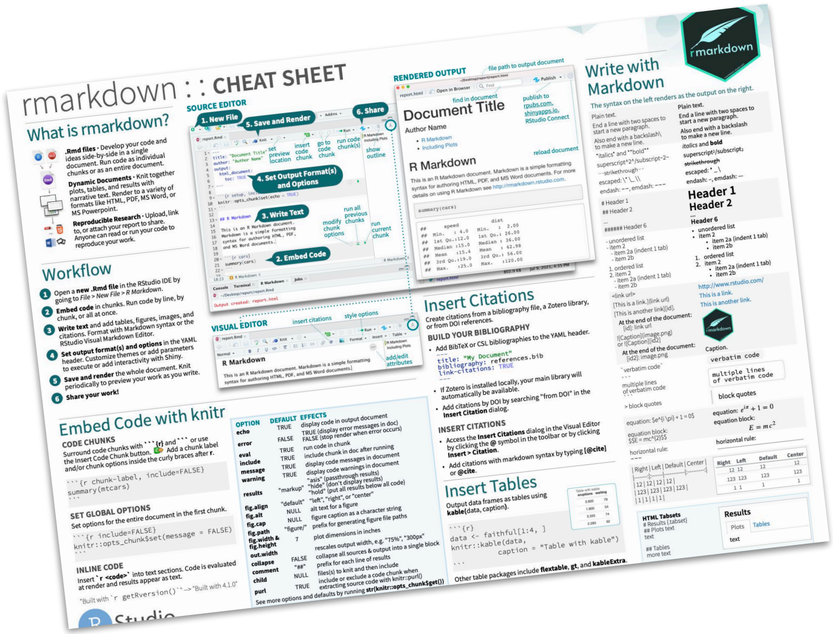
RMarkdown

Communicate and reproduce results

.Rmd files: used to develop code and ideas side by side and run code as individual chunks or as an entire document.

Dynamic Documents: Knit together plots, tables, and results with narrative text. Render to a variety of formats like HTML, PDF, MS Word, or MS Powerpoint.

Reproducible Research: Upload, link to, or attach your report to share. Anyone can read or run your code to reproduce your work.



For further reference, check out **RMarkdown cheat sheet** included in the GitHub repository or RStudio Help Menu.

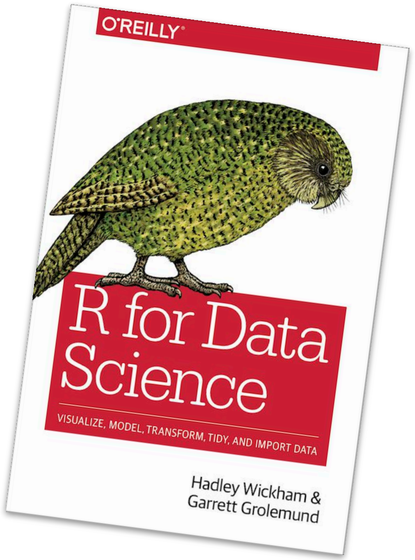
Where to get help?

Sites where you can find examples and answers about R

stackoverflow

R-bloggers

Google



Interviewer: What is your programming experience?

Me: I can use Stack Overflow.

Interviewer: You're hired.

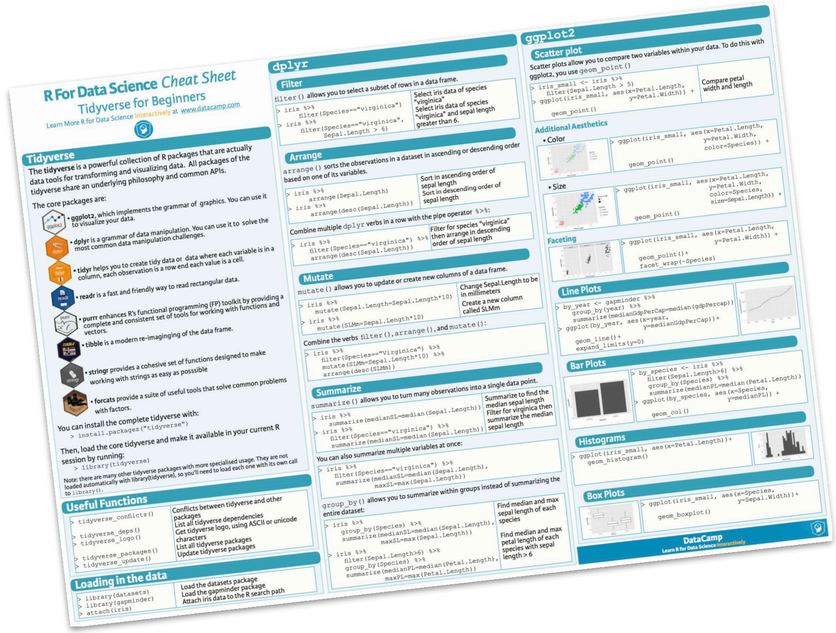
Hands-on session

Break

Plain and spatial data wrangling using R

Tidyverse

Packages for transforming and visualizing data



For further reference, check out **Tidyverse for beginners cheat sheet** included in the GitHub repository or RStudio Help Menu.

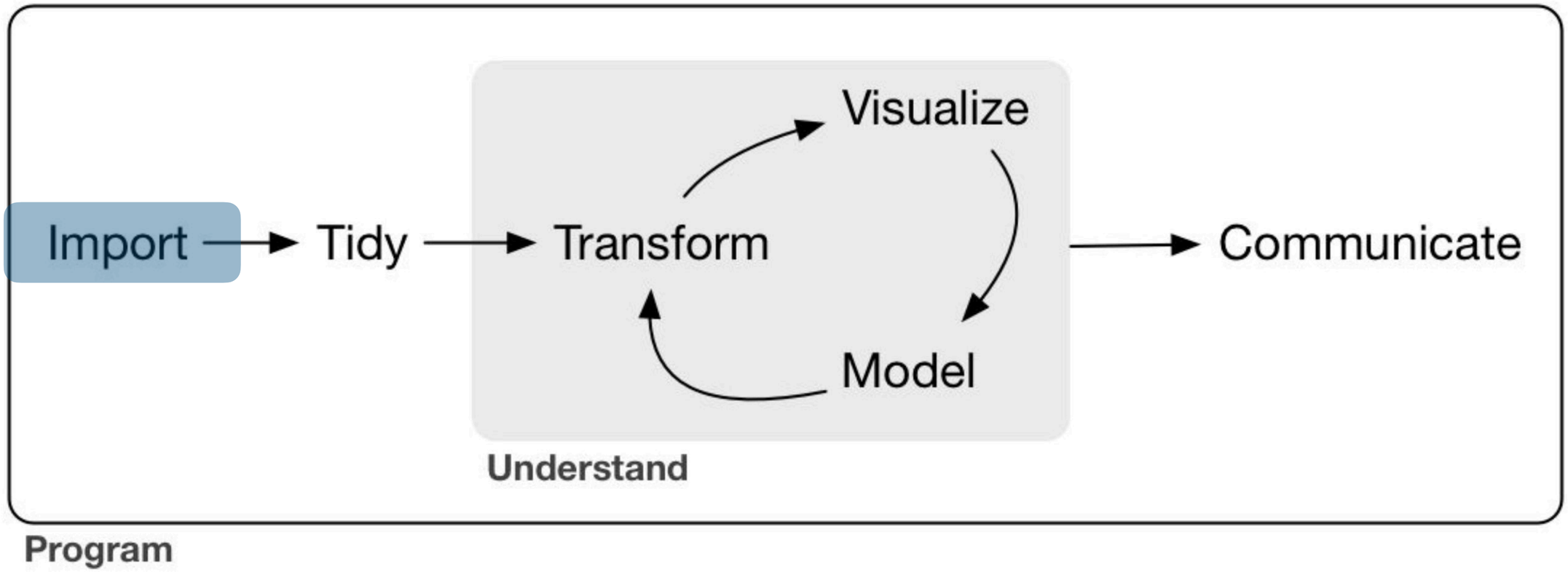
Using Base R for your analysis and copy pasting your results into tables in Word

learning how to use dplyr visualize data with ggplot2 and report your analysis in rmarkdown documents

wielding the entire might of the tidyverse

Getting data into R

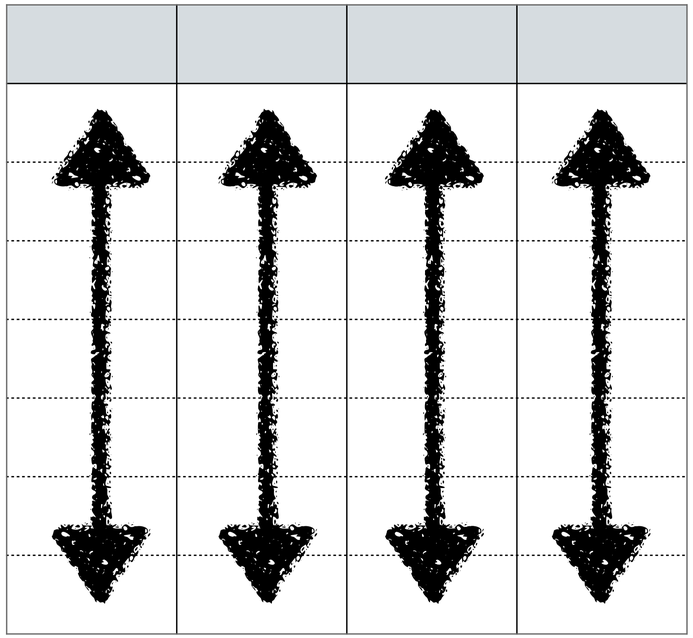
Data science journey



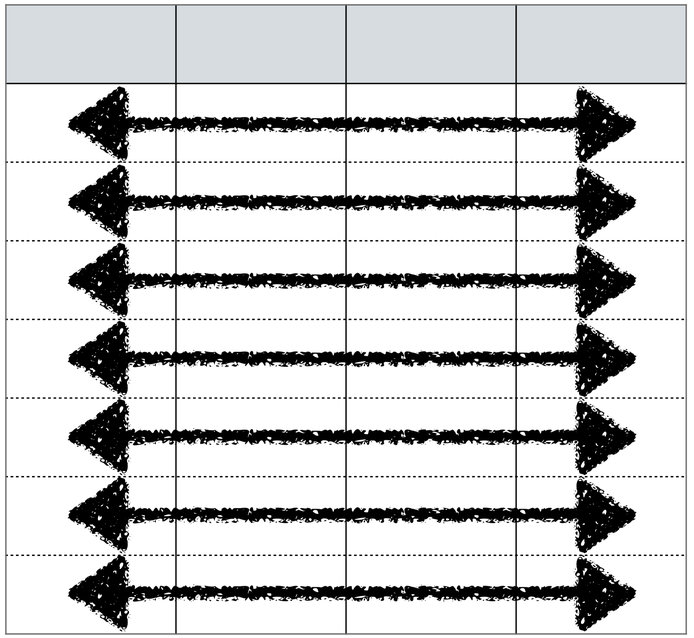
Reference from Wickham, H., & Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data.* O'Reilly Media, Inc.

Tidy data

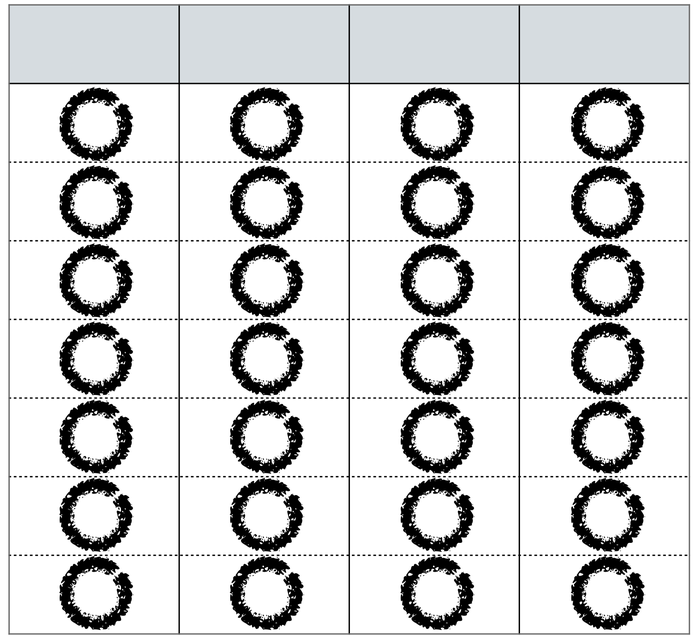
Rules to make a dataset tidy



Each variable is a column



Each observation is a row



Each value must have its own cell

Reference from Wickham, H., & Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data.* O'Reilly Media, Inc.

Tidy data

Shapes of data tables

Longer table

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

table4

Gathering data in tidy form

Wider table

country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

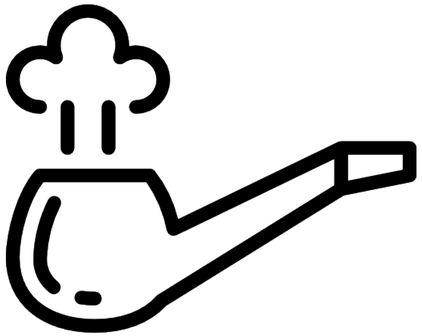
country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

Spreading data makes it tidy

Reference from Wickham, H., & Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data.* O'Reilly Media, Inc.

Pipe operator (%>%)

Concatenate functions in R



```
x %>% f(y)
```

becomes

```
f(x,y)
```

A ***pipe operator*** takes the outcome of a function and turns it to the income of another function.

```
glimpse(arrange(select(mutate(filter(iris,
Sepal.Length > 5), Petal.Ratio = Petal.Length/
Petal.Width, Sepal.Ratio = Sepal.Length/
Sepal.Width), Species, Sepal.Ratio, Petal.Ratio),
Species))
```

using %>% becomes

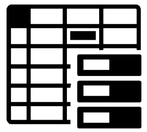
```
iris %>%
  filter(Sepal.Length > 5) %>%
  mutate(Petal.Ratio = Petal.Length/Petal.Width,
         Sepal.Ratio = Sepal.Length/Sepal.Width) %>%
  select(Species, Sepal.Ratio, Petal.Ratio) %>%
  arrange(Species) %>%
  glimpse()
```

Tibbles

Modern data frames for the Tidyverse

Comparing vs classic *data.frame*

Better printing & visualization



Easiest & fastest access to values

Coercible from other object types

Icons made by Freepik from www.flaticon.com.

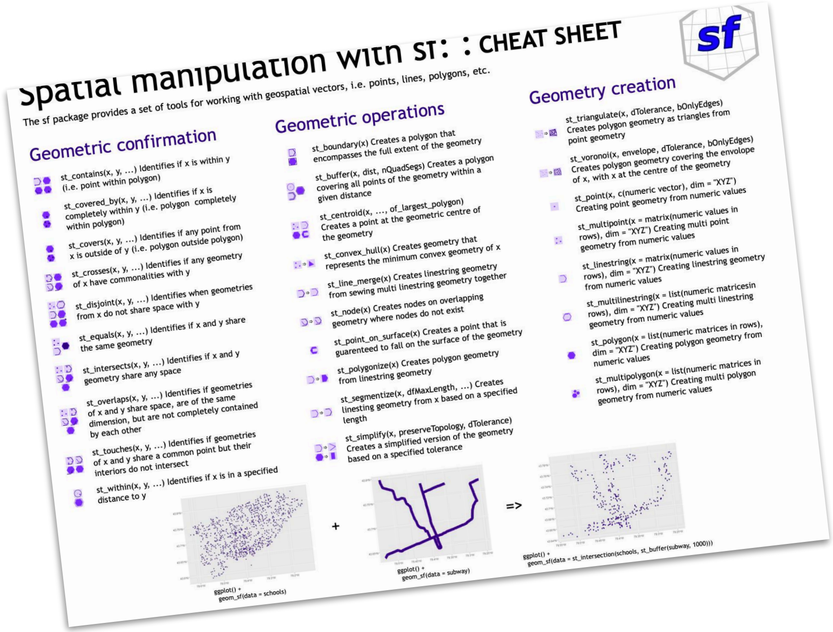
sf

Manipulating spatial data in R

R package used to store, access and manipulate spatial data using the *simple features* standard and common geographic libraries like *GEOS*, *GDAL* and *PROJ*.

Unlike the *sp* package (also for spatial data access and manipulation), *sf* is a modern take to use spatial data in R and it's works well with the *Tidyverse* package family.

Its data format can be handled just like a *tibble*, but adding a geometry variable to store spatial data and allow geometric operations.


Spatial manipulation with sf: : CHEAT SHEET

The sf package provides a set of tools for working with geospatial vectors, i.e. points, lines, polygons, etc.

Geometric confirmation

st_contains(x, y, ...) Identifies if x is within y (i.e. point within polygon)

st_covered_by(x, y, ...) Identifies if x is completely within y (i.e. polygon completely within polygon)

st_covers(x, y, ...) Identifies if any point from x is outside of y (i.e. polygon outside polygon)

st_crosses(x, y, ...) Identifies if any geometry of x have commonalities with y

st_disjoint(x, y, ...) Identifies when geometries from x do not share space with y

st_equals(x, y, ...) Identifies if x and y share the same geometry

st_intersects(x, y, ...) Identifies if x and y geometry share any space

st_overlaps(x, y, ...) Identifies if geometries of x and y share space, are of the same dimension, but are not completely contained by each other

st_touches(x, y, ...) Identifies if geometries of x and y share a common point but their interiors do not intersect

st_within(x, y, ...) Identifies if x is in a specified distance to y

Geometric operations

st_boundary(x) Creates a polygon that encompasses the full extent of the geometry

st_buffer(x, dist, nQuadSegs) Creates a polygon covering all points of the geometry within a given distance

st_centroid(x, ..., of_largest_polygon) Creates a point at the geometric centre of the geometry

st_convex_hull(x) Creates geometry that represents the minimum convex geometry of x

st_line_merge(x) Creates linestring geometry from sewing multi linestring geometry together

st_node(x) Creates nodes on overlapping geometry where nodes do not exist

st_point_on_surface(x) Creates a point that is guarenteed to fall on the surface of the geometry

st_polygonize(x) Creates polygon geometry from linestring geometry

st_segmentize(x, dfMaxLength, ...) Creates linesting geometry from x based on a specified length

st_simplify(x, preserveTopology, dTolerance) Creates a simplified version of the geometry based on a specified tolerance

Geometry creation

st_triangulate(x, dTolerance, bOnlyEdges) Creates polygon geometry as triangles from point geometry

st_voronoi(x, envelope, dTolerance, bOnlyEdges) Creates polygon geometry covering the envolope of x, with x at the centre of the geometry

st_point(x, c(numeric vector), dim = "XYZ") Creating point geometry from numeric values

st_multipoint(x = matrix(numeric values in rows), dim = "XYZ") Creating multi point geometry from numeric values

st_linestring(x = matrix(numeric values in rows), dim = "XYZ") Creating linestring geometry from numeric values

st_multilinestring(x = list(numeric matricesin rows), dim = "XYZ") Creating multi linestring geometry from numeric values

st_polygon(x = list(numeric matrices in rows), dim = "XYZ") Creating polygon geometry from numeric values

st_multipolygon(x = list(numeric matrices in rows), dim = "XYZ") Creating multi polygon geometry from numeric values

ggplot() + geom_sf(data = schools)

ggplot() + geom_sf(data = subway)

ggplot() + geom_sf(data = st_intersection(schools, st_buffer(subway, 1000)))

For further reference, check out **Spatial manipulation with sf cheat sheet** in included in the GitHub repository or RStudio Help Menu.

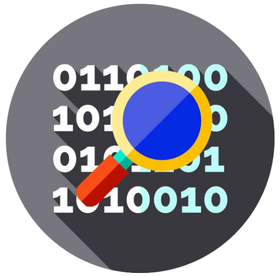
Icons made by Freepik from www.flaticon.com.

Case of use

Example of geospatial analysis

Murder in Mexico City and the Country

Import & tidy



Analyze

Visualize

Decision making

Hands-on session

Break

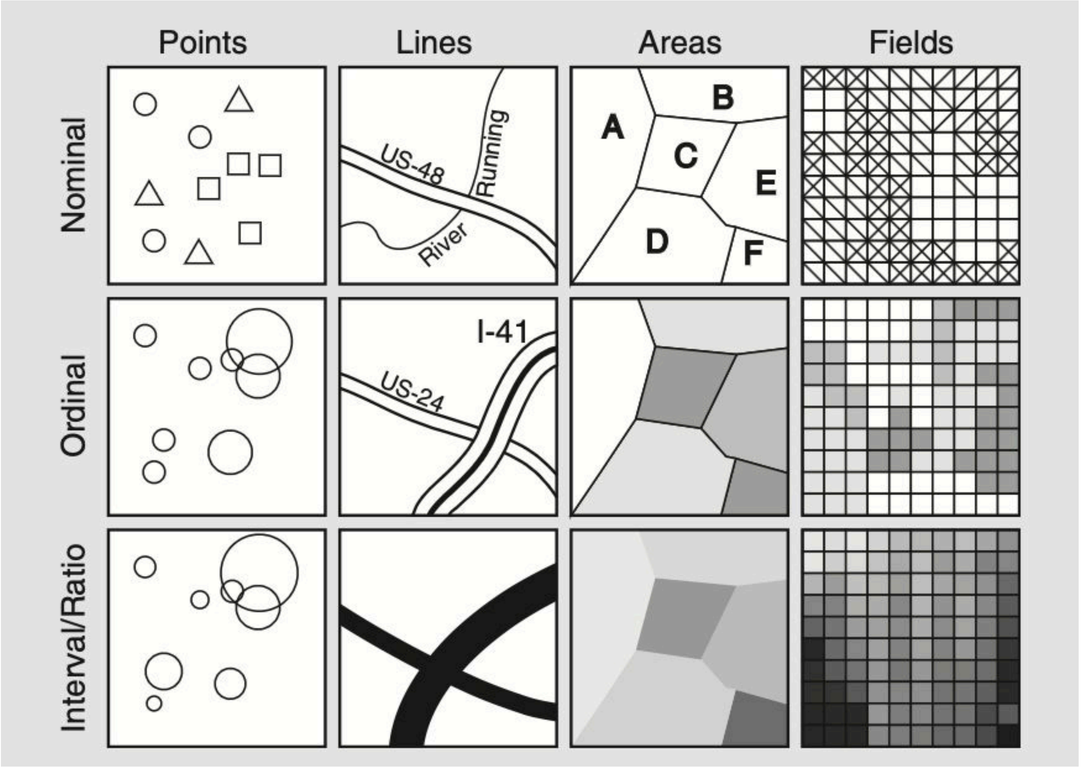
Exploratory Data Analysis (EDA)

Spatial data types

Geometries used in GIS

1. **Vector view.** Records locational (x, y) coordinates of the features that make up a map.
   - Point
   - Line
   - Area

2. **Raster systems.** Instead of starting with objects on the ground, a grid of small units, called pixels, of the Earth's surface is defined.



A schematic representation of entity-attribute spatial data types

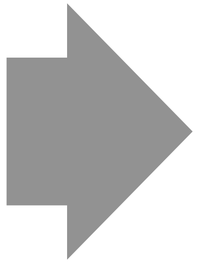
Reference taken from O'sullivan, D., & Unwin, D. (2003). *Geographic information analysis*. John Wiley & Sons.

Exploratory Data Analysis (EDA)

Using statistical measures to analyze data distributions

Descriptive statistics:

- Measures of Central Tendency
  - Mode
  - Median
  - Mean
- Measures of Dispersion:
  - Range
  - Standard deviation
  - Variance



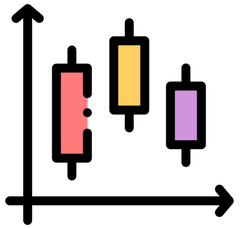
Spatial Statistics:

- Spatial Measures of Central Tendency
  - Spatial Median/Mean Center
  - Weighted Spatial Mean/Mean Center
  - Spatial Median/Median Center
- Spatial Measures of Dispersion:
  - Standard Distance
  - Standard Deviation Ellipse
  - Variance

Reference taken from Oyana, T. J. (2020). Spatial Analysis with R: Statistics, Visualization, and Computational Methods. CRC press.

Exploratory Data Analysis (EDA)

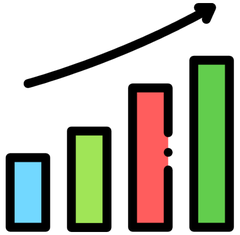
Using statistical measures to analyze data distributions



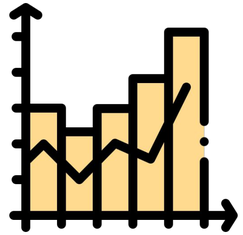
1. Identify the largest and small values

2. Derive the range

3. Determine the number of classes

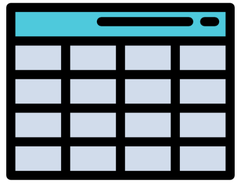


4. Define the class interval



5. Determine the frequency for each class

6. Compile this information in a table



Reference taken from Oyana, T. J. (2020). Spatial Analysis with R: Statistics, Visualization, and Computational Methods. CRC press.

Hands-on session

Questions?

See you tomorrow. Thank you!