Data Analysis in R

By
Dr. Hussein Hazimeh

Lebanese University
Faculty of Information 1

Data Science Departement

2rd year – Data Analysis in R

Spring – 2022 – Chapter 2

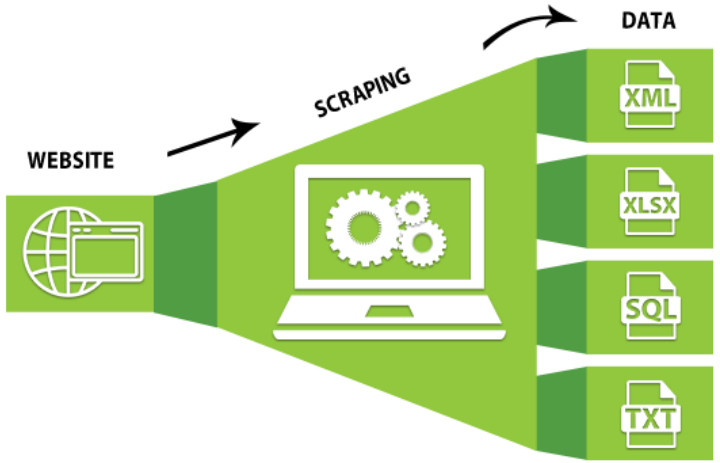
Agenda

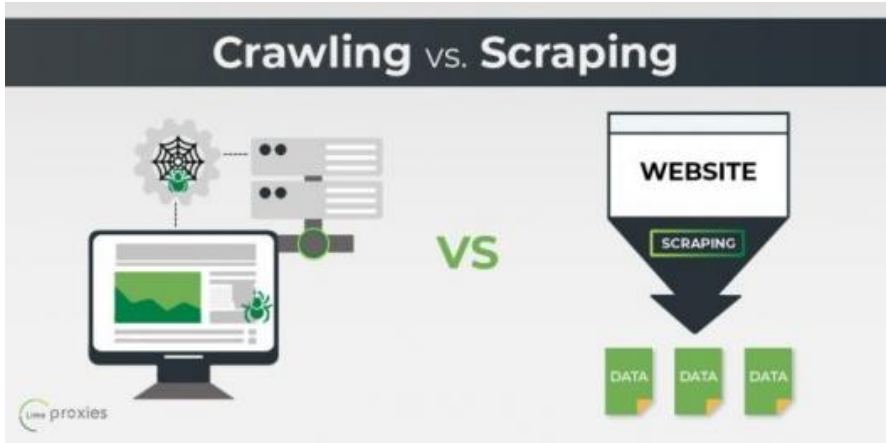
- Whats is data scraping?
- Data scraping vs data analysis
- Data scraping techniques
- Data scraping in R
- RSelenium introduction
- RSelenium configuration
- RSelenium tutorial
- Web data extraction
- Assignment

What is Data Scraping?

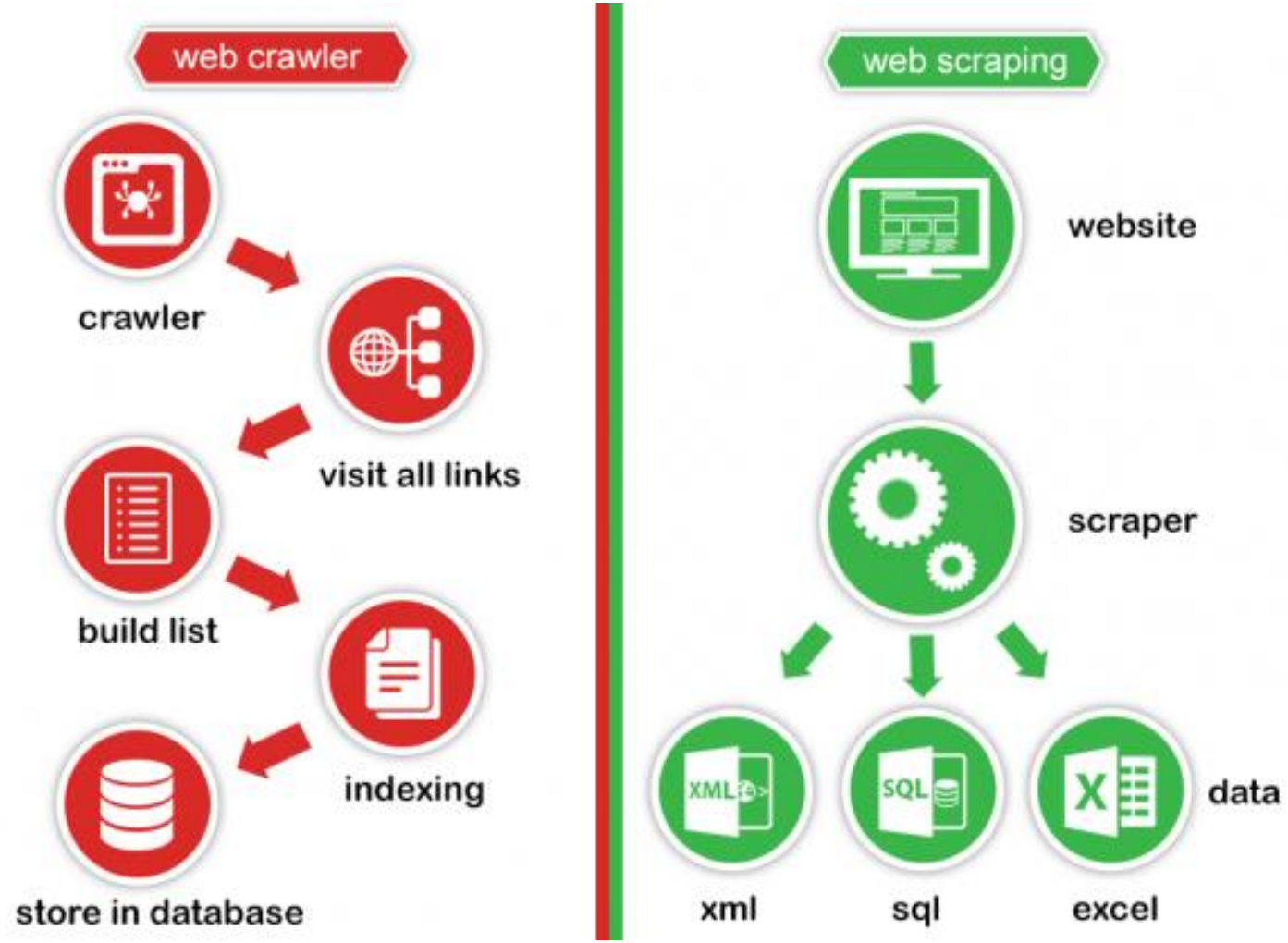
Whats is Data Scraping?

- **Data scraping** is the process of collecting pieces of data from one or multiple data sources.
- A typical example is the extraction of data from web pages.
- **Web scraping** is the part of data scraping that is oriented to web page data collection.
- **Data crawling** is different from data scraping.
- It is the process of navigating and visiting web pages, typically via links.
- Data scraping is a part of the data crawling process.
- **Data scrapping results** can be saved in different formats such as excel, json, xml, txt, etc.



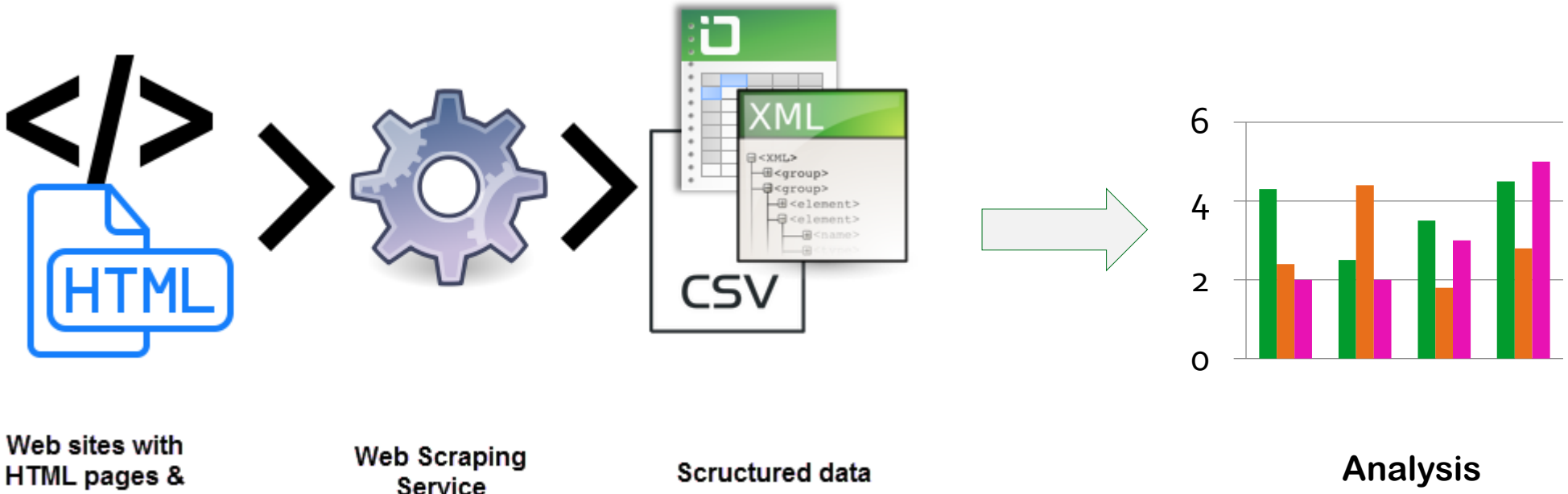


Data Scraping vs Data Crawling?



Data Scraping vs Data Analysis?

» Data scraping is a part of the data analysis process.



Data Scraping techniques?

- **Copy-pasting**
- **DOM Parsing**
- **HTTP Programming**: Using socket programming, posting HTTP requests can help one retrieve dynamic as well as static web page information.
- **Web scraping Software**: It can automatically retrieve the information off the web page, convert it into recognizable information, and store it in a local database.
  - Selenium [https://www.selenium.dev/]

Data Scraping in R?

- R provides multi-puropose libraries for extracting data from different formats.
  - Databases
  - Web pages
  - Text documents
  - Json files
  - XML files

- It provides also very useful libraries for extracting web data
  - RSelenium
- The libraries can be simply implemented.

What is Selenium?

- **Selenium** primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.
- Boring web-based administration tasks can (and should) also be automated as well.

- **Selenium IDE**: Selenium IDE is a complete integrated development environment (IDE) for Selenium tests. It is implemented as a Firefox Add-On and as a Chrome Extension. It allows for recording, editing and debugging of functional tests.

- **Selenium WebDriver:** is the successor to Selenium RC. Selenium WebDriver accepts commands (sent in Selenese, or via a Client API) and sends them to a browser. This is implemented through a browser-specific browser driver, which sends commands to a browser and retrieves results.

- **Selenium Grid**: is a server that allows tests to use web browser instances running on remote machines. With Selenium Grid, one server acts as the hub. Tests contact the hub to obtain access to browser instances.

What is RSelenium?

- The goal of **RSelenium** is to make it easy to connect to a Selenium Server/ Remote Selenium Server from within R.
- RSelenium provides R bindings for the Selenium Webdriver API.
- **RSelenium** allows you to carry out unit testing and regression testing on your webapps and webpages across a range of browser/OS combinations.
- This allows us to integrate from within R testing and manipulation of popular projects such as shiny, sauceLabs.
- Installation:
  - install.packages("RSelenium")

Rselenium tutorial

How do I connect to a running server?

`RSelenium` has a main reference class named `remoteDriver`. To connect to a server you need to instantiate a new `remoteDriver` with appropriate options.

```
# RSelenium::startServer() if required
require(RSelenium)
remDr ← remoteDriver(remoteServerAddr = "localhost"
                     , port = 4444
                     , browserName = "firefox"
                     )
```

Rselenium web data extraction

Finding web elements by name:

```
con$navigate("http://www.google.com/ncr")
webElem <- con$findElement(using = "name", value = "q")
webElem$getElementAttribute("name")
```