Parsing XML

SICSS Bologna 2024

Paride Carrara

University of Bologna

September 24, 2025

Outline

1. What is XML

2. Parsing XML

3. Data Cleaning

Outline

1 What is XML

2 Parsing XML

3 Data Cleaning

Scraping vs Parsing

- **Scraping**:
  - **Definition**: Extracting data from websites by downloading the HTML and processing it to retrieve the desired information.
  - **Purpose**: Used to gather data from web pages where structured data formats like XML or JSON are not available.
- **Parsing**:
  - **Definition**: Analyzing a structured document (like XML or JSON) to extract specific pieces of information.
  - **Purpose**: Used when working with data in structured formats to easily navigate and retrieve specific data.

Scraping vs Parsing: Comparison

- **Input**:
  - Scraping: HTML from web pages.
  - Parsing: Structured data formats like XML, JSON.
- **Output**:
  - Scraping: Extracted data from unstructured or semi-structured web pages.
  - Parsing: Extracted data from structured documents.
- **Complexity**:
  - Scraping: Can be complex due to varying web page structures and dynamic content.
  - Parsing: More straightforward due to predefined document structure.

What is XML?

- **Definition**:
  - Extensible Markup Language (XML) is a flexible text format designed to meet the challenges of large-scale electronic publishing and also plays an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

- **Purpose**:
  - XML is used to describe data. It focuses on what data is, unlike HTML which focuses on how data looks. XML is designed to be self-descriptive.

- **Key Features**:
  - **Human-readable**: XML documents are readable and understandable by both humans and machines.
  - **Platform-independent**: XML provides a software- and hardware-independent way of storing data.
  - **Extensible**: XML is not a fixed format like HTML. It is extensible because it allows users to create their own tags.

XML vs. HTML

- **HTML**:
  - Used to display data and focus on how data looks.
  - Has a predefined set of tags like `<h1>`, `<p>`, etc.
- **XML**:
  - Used to describe and store data.
  - Tags are not predefined. Users can define their own tags.

XML vs. HTML

- **HTML:**

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

- **XML:**

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

XML Structure

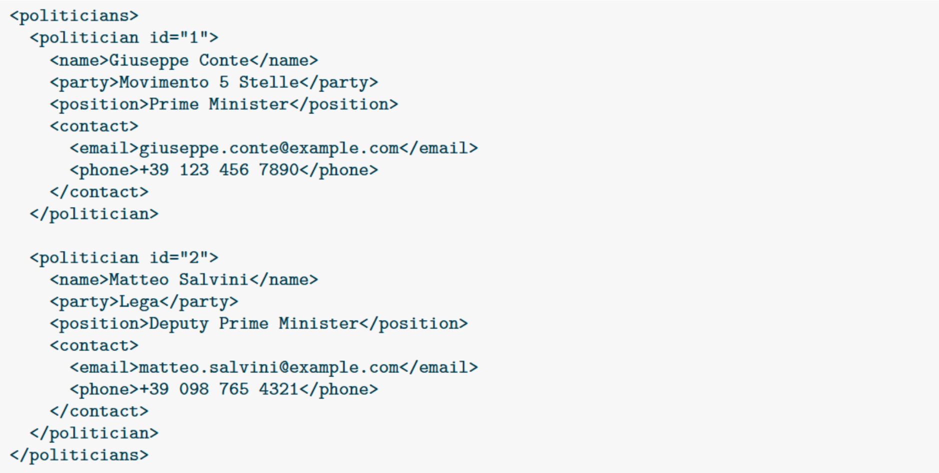
- **Elements:**
  - XML building blocks, defined by tags, e.g., `<title>Introduction to XML</title>`.
  - An element consists of a start tag, content, and an end tag. Content can be Text, Nested Elements, Attributes, or a mix.

- **Attributes**: Provide additional information (more about proprieties or metadata) about elements, e.g., `<book id="123">`

- **Prolog**: Contains the XML declaration of version and encoding, e.g., `<?xml version="1.0" encoding="UTF-8"?>`.

- **Hierarchy:** Elements can be nested, forming a tree structure.

Sample XML code

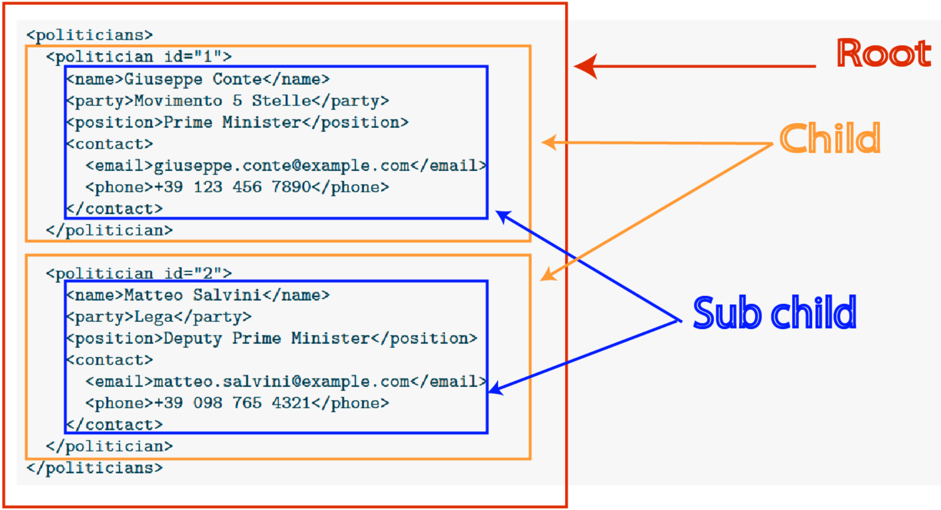


```
<politicians>
  <politician id="1">
    <name>Giuseppe Conte</name>
    <party>Movimento 5 Stelle</party>
    <position>Prime Minister</position>
    <contact>
      <email>giuseppe.conte@example.com</email>
      <phone>+39 123 456 7890</phone>
    </contact>
  </politician>

  <politician id="2">
    <name>Matteo Salvini</name>
    <party>Lega</party>
    <position>Deputy Prime Minister</position>
    <contact>
      <email>matteo.salvini@example.com</email>
      <phone>+39 098 765 4321</phone>
    </contact>
  </politician>
</politicians>
```

Figure: An example of XML code

Sample XML Document



```xml
<politicians>
  <politician id="1">
    <name>Giuseppe Conte</name>
    <party>Movimento 5 Stelle</party>
    <position>Prime Minister</position>
    <contact>
      <email>giuseppe.conte@example.com</email>
      <phone>+39 123 456 7890</phone>
    </contact>
  </politician>
  <politician id="2">
    <name>Matteo Salvini</name>
    <party>Lega</party>
    <position>Deputy Prime Minister</position>
    <contact>
      <email>matteo.salvini@example.com</email>
      <phone>+39 098 765 4321</phone>
    </contact>
  </politician>
</politicians>
```

Figure: An example of XML code

Structure of an XML code

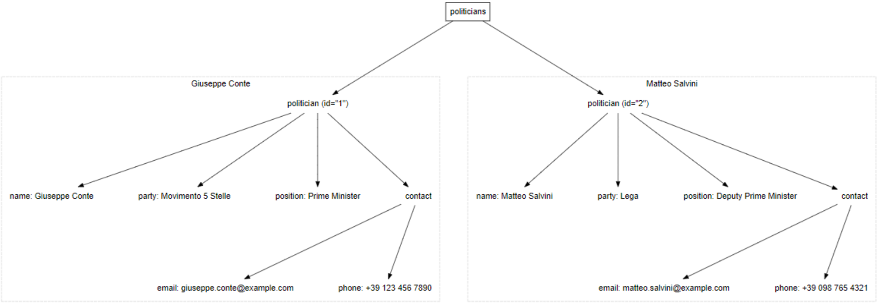


Figure: Structure XML Document

Outline

1. What is XML

2. Parsing XML

3. Data Cleaning

Parsing XML with xml2 Package

- We are going to parse the legislative debates from the Italian Senate
- Focus on the parliamentary session of the 5th June 2018
- Investiture debate of the Conte I cabinet
- We use the xml2 package

Exercise in **Rstudio**

Outline

1. What is XML

2. Parsing XML

3. Data Cleaning

Data Cleaning

Once we parsed the data, are we ready to analyze it?

- Usually no...

Data Cleaning

- Due to the structure of XML, it should be relatively easy to extract the desired information
- However, data is (almost) always messy
- After collecting the data, we need to clean the data

Some of the Tools

- dplyr
- stringr

Exercise in **Rstudio**