STATS 769
Data Formats

Paul Murrell

The University of Auckland

July 30, 2021

Overview

- In this lecture we will discuss some modern data formats.
  - JSON.
  - XML (and XPath).
- The aim is to become familiar with these formats and to be able to work with them in R.

JSON

- JavaScript Object Notation.
- Text format.
- JSON Data Types: `null`, `true`, `false`, number, string.
- JSON Data Containers: ordered (and unnamed) arrays `[ ]`, named (and unordered) arrays `{ }`.
- Can mix types within arrays.

JSON in R

- The **jsonlite** package has `fromJSON()` (and `toJSON()`).
- The result is the simplest structure possible, in the order: vector, matrix, data frame, list.
- Nested JSON objects can produce nested data frames; the `flatten()` function turns these into normal 2-D data frames.
- Use `prettify()` to eyeball raw JSON and `str()` to explore nested list structures.

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <row>
    <row _uuid="00000000-0000-0000-AF9A-401551B08E58">
      <month>Jan</month>
      <pets_adopted>129</pets_adopted>
    </row>
    <row _uuid="00000000-0000-0000-F7B9-E37345BC66E7">
      <month>Mar</month>
      <pets_adopted>126</pets_adopted>
    </row>
  </row>
</response>
```

XML in R

- The **xml2** package has `read_xml()`
- The result is NOT a data frame.
- Extract elements of interest using `xml_find_all()` and XPath expressions
- Use `xml_text()` to extract text content from elements.
- Use `xml_attr()` to extract attribute values from elements.
- **NOTE** that all content and attribute values are **character** values.

XPath

Language for expressing subsets of an XML document.

- An XPath expression consists of some combination of **location paths** of the form ...

  `axisname::nodetest[predicate]`

  ... but usually just ...

  `nodetest`

  ... or ...

  `nodetest[predicate]`

XPath

An XPath expression is formed by combining several location paths, separated by a forward slash, `/`.

- If the expression **begins** with a forward slash, matching starts from the document root node.

  ```
  /a/b/c
  ```

- A double forward slash, `//`, is short for `/descendant-or-self::node()/`.

  ```
  /a//c
  ```

XPath

The nodetest is commonly just the name of an element or `@name` to match an attribute.

- It can also be a wildcard, `*`, which matches any element, or `@*`, which matches any attribute.

```
/a/b
/a/@id
/a/*/c
```

XPath

The predicate is like a subsetting expression.

- It can be a simple integer.

```
/a/b[1]
```

- It can be a comparison.

```
/a/b[@year > 2000]
/a[@lang = "en"]
```

- It can be a special function.

```
/a/b[last() - 1]
/a/b[contains(@id, "paul")]
```

XPath

The axis is relative to the current node in the XML document. The default axis is `child`, which means to search children of the current node (this is what happens if we do not specify an axis in a location path).

- The `following-sibling` axis can force the search to look at siblings of the current node.

  ```
  /a/b/following-sibling::*
  ```

- The `parent` or `ancestor` axes can force the search to look back up the hierarchy of XML nodes.

  ```
  /a/b/c/ancestor::a
  ```

- These can be useful within a predicate.

  ```
  //c[ancestor::b@lang = "en"]
  ```

Resources

- 'jsonlite' mapping between R objects and JSON http://cran.r-project.org/web/packages/jsonlite/vignettes/json-mapping.pdf
- w3schools XPath tutorial http://www.w3schools.com/xpath/