STAT 406 Lab 10 : Working with XML files

1 XML Basic Overview

- XML is widely used for exchanging information on the World Wide Web. It stands for eXtensible Markup Language.
- Markup Language is a system of how the document is to be described or logically structured, e.g., HTML and XML. HTML is used for displaying data on the web browser, XML is used for carrying data on the web.
- The underpining data structure for XML files is of tree structure, called "document tree".
- Structure of XML.

  Except for the file head declaration like

  ```
  <?xml version= 1 .0    ?>
  ```

  and comments in XML file like

  ```
  <!-- This is a comment -->
  ```

  The most concerned structure of an XML file in our class is in the main body of the file:

- Elements(Nodes of the document tree) and Attributes: Elements contain the actual data of XML document. Elements must have a start-tag and an end-tag which contain the elements name. The content sits between these two tags.

  Elements have a tree-based data structure.
  The general tree structure of a XML file looks like below:

  ```
  <root>
    <child>
      <subchild>.....</subchild>
    </child>
  </root>
  ```

  Here is a graphical illustration of the XML tree data structure. (on next page)

  The following is a simple XML file, scan through it and answer the questions:

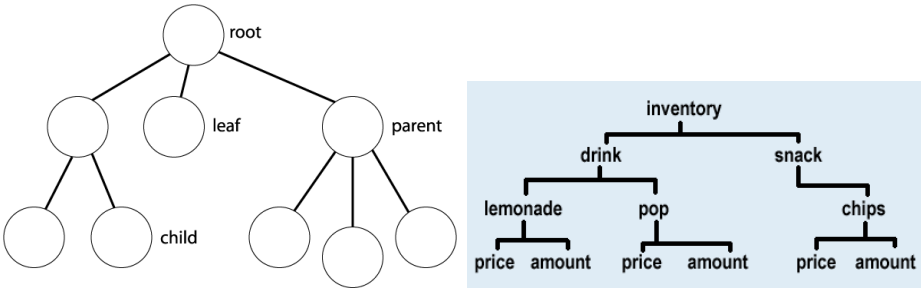

Figure 1: XML document tree structure illustration

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

1. How many root nodes does this XML file have? What is the name of the root node?
2. What are the the children nodes of the the root node? How many children nodes does the root have?
3. Please draw the document tree for this XML file.
4. What are the attribute and its value of the first "book" node? How about attribute and its value of the subchild node "title"?
5. What is the value of the node "author" for the second book?

Notice that **unlike the value of the nodes**, **the value of attributes must be quoted**, using either single quotes or double quotes, like the following.

```
<node_name attribute_name = "attribute value"> Node_value </node_name>
```

- Parse XML data: A parser is a component of the complier which checks for correct syntax and builds a data structure. The R package XML provides the required parser and extends Rs capability in processing XML files.

  Download and save the XML file "filename.xml" in your R working directory first or nothing can be done even with correct codes.

```
## Install and quote the R package XML first ##
if (!( require(XML))){
  install .packages("XML", dep=T);
  require(XML);
}
doc <- xmlTreeParse('filename.xml')
# doc contains the underpining tree structure of the file
```

In the following two examples, we will parse and process the NSF award XML file and the US congress people XML file respectively.

2 Examples of Processing XML files

2.1 Example I. NSF Award files

First we want to parse a .xml file to reveal the document tree structure. Download the **award1.xml** file in ctools and save to your working directory in R.

- Parse the file:

```
doctree = xmlParse('award1.xml');
```

- Extracting the root:

```
root <- xmlRoot(doctree);
xmlName(root);
# [1] "rootTag"
```

- List the children nodes for the root, how many are there and what are the names?

```
rtCh = xmlChildren(root);
length(rtCh);
# [1] 1
names(rtCh);
# [1] "Award"
```

- Extract all the children of the first child of the root:

```
ch <- xmlChildren(root[[1]]);
head(ch, 3);
# $AwardTitle ...
  $AwardEffectiveDate ...
  $AwardExpirationDate ...
length(ch);
```

```
# [1] 16
names(ch);
# [1] "AwardTitle"
  [2] "AwardEffectiveDate"
  [3] "AwardExpirationDate"
  [4] "AwardAmount"
  ...
```

Once we understand the document structure, we will pull out the information we need.

Question 1 : Download and save the 2016 folder in ctools to your working directory, extract with R the **title, start/end dates, amount, investigator's name and the institution name** for each award in the 2016 award folder and output these information in a data frame. Each XML file in the NSF award folder 2016 records one award.

Solution 1 : See lab10.r code.

2.2 Example II. US congress people.xml file

- Parse and explore the document tree structure as usual.

```
doc = xmlTreeParse("people.xml");
root = xmlRoot(doc);
ls_people = xmlChildren(root); length(ls_people);

person1 = root[[1]];
print(person1);
# use xmlAttrs() to list the attributes,
# and xmlGetAttr() to get each attribute's value
xmlAttrs(person1);
xmlGetAttr(person1, 'id');
# list all children nodes for person1;
ls_person1 = xmlChildren(person1);
names(ls_person); # only 1 'role' child node

person2 = root[[2]];
print(person2);
ls_person2 = xmlChildren(person2);
names(ls_person2); # 3 children nodes
# get the first committee assignment subchild for person2
ls_person2$"committee-assignment";
```

After the exploration, we begin processing the file.

Question 2 : Get the following attributes of nodes in people.xml file, they are 'id','lastname','firstname','gender','party','state','committee'.

Solution 2 : See lab10.r code.

3 Conclusion

We have used the following functions in R package XML to parse and process a XML file.

- doc = xmlTreeParse('a_file.xml') applied to a xml file as a begining for processing the file;
- root = xmlRoot('doc') to get the root node of this xml document tree;
- ch_list = xmlChildren(a_node) applied to a node, say the root, to get a list of children of this node;
- ch_list[[i]], ch_list$'subchild name', or node[[i]] to get a child node;
- xmlValue(a_leaf_node) to get the value of a leaf node (a terminal node with no child);
- xmlGetAttr(node, 'attribute') to fetch the value of 'attribute' of the node;