# 607 - Week 7

*Jose Mawyin*

*10/9/2019*

## Working with XML and JSON in R

For this homework, I created a table in Numbers from MacOS and then exported the table as an .CSV file. Then I used the following online conversion tools to load the .CSV file and convert it to to the required formats.

### Convert CSV to JSON, XML

http://www.convertcsv.com/csv-to-json.htm http://www.convertcsv.com/csv-to-xml.htm http://www.convertcsv.com/csv-to-html.htm

### Loading HTML Table

```
# Load HTML file
HTML.File <- "/Users/josemawyin/Library/Mobile Documents/com~apple~CloudDocs/Data Science Masters /607/6
# Parse HTML data
HTML.Table <- readHTMLTable(HTML.File)
```

The easiest load was from HTM into a data.frame using "readHTMLTable" from the XML package. Right away, you could use the created data.frame for further analysis.

```
HTML.Table
```

```
## $`NULL`
##                  Title
## 1           Evolution
## 2 The Player of Games
## 3        Eye of Terra
##                                                                                         Authors
## 1                                                                              Stephen Baxter
## 2                                                                               Iain M. Banks
## 3 Graham McNeill, Aaron Dembski-Bowden, Chris Wraight, Gav Thorpe, Matthew Farrer, Rob Sanders
##    Published Date     Publisher       ISBN Pages
## 1 January 1, 2003       Del Rey  345457838   672
## 2  March 26, 2008         Orbit  316005401   416
## 3   June 27, 2017 Black Library B074R7ZCDF   480
```

### Loading XML File

Extracting data from a XML file is a bit more elaborate because we need to specify the XML tag that defines the individual entries. To load the .XML file we use the "xmlParse" function from the XML package.

```
# Load XML file
XML.File <- "/Users/josemawyin/Library/Mobile Documents/com~apple~CloudDocs/Data Science Masters /607/60
# Parse XML data
xmlfile <- xmlParse(XML.File)
xmlfile
```

```
## <?xml version="1.0" encoding="UTF-8"?>
## <root>
##   <row>
##     <Title>Evolution</Title>
##     <Authors>Stephen Baxter</Authors>
##     <Published_Date>January 1, 2003</Published_Date>
##     <Publisher>Del Rey</Publisher>
##     <ISBN>345457838</ISBN>
##     <Pages>672</Pages>
##   </row>
##   <row>
##     <Title>The Player of Games</Title>
##     <Authors>Iain M. Banks</Authors>
##     <Published_Date>March 26, 2008</Published_Date>
##     <Publisher>Orbit</Publisher>
##     <ISBN>316005401</ISBN>
##     <Pages>416</Pages>
##   </row>
##   <row>
##     <Title>Eye of Terra</Title>
##     <Authors>Graham McNeill, Aaron Dembski-Bowden, Chris Wraight, Gav Thorpe, Matthew Farrer, Rob San
##     <Published_Date>June 27, 2017</Published_Date>
##     <Publisher>Black Library</Publisher>
##     <ISBN>B074R7ZCDF</ISBN>
##     <Pages>480</Pages>
##   </row>
## </root>
##
```

Here we define the XML tag that defines a node or entry from the encoded table using the "xmlToDataFrame" function from the XML package.

```
# Get place nodes
XML.Table <- xmlToDataFrame(nodes=getNodeSet(xmlfile,"//row"))
XML.Table
```

```
##                  Title
## 1            Evolution
## 2 The Player of Games
## 3         Eye of Terra
##                                                                                          Authors
## 1                                                                                 Stephen Baxter
## 2                                                                                  Iain M. Banks
## 3 Graham McNeill, Aaron Dembski-Bowden, Chris Wraight, Gav Thorpe, Matthew Farrer, Rob Sanders
##    Published_Date     Publisher       ISBN Pages
## 1 January 1, 2003       Del Rey  345457838   672
## 2  March 26, 2008         Orbit  316005401   416
## 3   June 27, 2017 Black Library B074R7ZCDF   480
```

```
colnames(XML.Table)
```

```
## [1] "Title"          "Authors"        "Published_Date" "Publisher"
## [5] "ISBN"           "Pages"
```

### Loading JSON File

Working with a .JSON file was the most difficult from all the file types. I used the “fromJSON” function fromt the rJason package to load the file into a table.

```r
#Load JSON file
JSON.File <- "/Users/josemawyin/Library/Mobile Documents/com~apple~CloudDocs/Data Science Masters /607/
# You can pass directly the filename
my.JSON <- fromJSON(file=JSON.File)
str(my.JSON)
```

```
## List of 3
##  $ :List of 6
##   ..$ Title         : chr "Evolution"
##   ..$ Authors       : chr "Stephen Baxter"
##   ..$ Published Date: chr "January 1, 2003"
##   ..$ Publisher     : chr "Del Rey"
##   ..$ ISBN          : chr "345457838"
##   ..$ Pages         : num 672
##  $ :List of 6
##   ..$ Title         : chr "The Player of Games"
##   ..$ Authors       : chr "Iain M. Banks"
##   ..$ Published Date: chr "March 26, 2008"
##   ..$ Publisher     : chr "Orbit"
##   ..$ ISBN          : chr "316005401"
##   ..$ Pages         : num 416
##  $ :List of 6
##   ..$ Title         : chr "Eye of Terra"
##   ..$ Authors       : chr "Graham McNeill, Aaron Dembski-Bowden, Chris Wraight, Gav Thorpe, Matthew
##   ..$ Published Date: chr "June 27, 2017"
##   ..$ Publisher     : chr "Black Library"
##   ..$ ISBN          : chr "B074R7ZCDF"
##   ..$ Pages         : num 480
```

The “fromJSON” function creates a list of list that now we need to parse through to create a data.frame. This we do below through the function “lapply”.

```r
df <- lapply(my.JSON, function(play) # Loop through each "play"
  {
  # Convert each group to a data frame.
  # This assumes you have 6 elements each time
  data.frame(matrix(unlist(play), ncol=6, byrow=T))
  })

# Now you have a list of data frames, connect them together in
# one single dataframe
df <- do.call(rbind, df)

# Make column names nicer, remove row names
#colnames(df) <- names(my.JSON[[1]][[1]])
colnames(df) <- colnames(XML.Table)
rownames(df) <- NULL

df
```

```
##                  Title
## 1           Evolution
## 2 The Player of Games
## 3        Eye of Terra
##                                                                                          Authors
## 1                                                                                Stephen Baxter
## 2                                                                                 Iain M. Banks
## 3 Graham McNeill, Aaron Dembski-Bowden, Chris Wraight, Gav Thorpe, Matthew Farrer, Rob Sanders
##    Published_Date     Publisher       ISBN Pages
## 1 January 1, 2003       Del Rey  345457838   672
## 2  March 26, 2008         Orbit  316005401   416
## 3   June 27, 2017 Black Library B074R7ZCDF   480
```

We have shown how loaded data from 3 different formats (HTML, XML and JSON) and successfully loaded the data into a data.frame. The conversion was easier with the HML and XML formats. However, JSON encapsulation of data into a string makes it suitable as a vehicle for tabulated data as well.