Lab 8 XML Extraction

Ryan Chui

April 6, 2015

revised version!-christine

In this lab, we will use the XML package to read exchange rates (against the euro) from the European Central Bank and create a time series plot showing how the rates for four different currencies–the British pound (GBP), the US dollar (USD), the Canadian dollar (CAD), and the Japanese yen (JPY)–have changed over time.

Before jumping to the code portions, open a browser and visit this URL:

http://www.ecb.europa.eu/stats/eurofxref/eurofxref-hist.xml

Examine the structure of the XML. What are the exchange rates for the British pound (GBP) for 4/2 and 4/1? *Type your answer here*

```
library(XML)

# Read the data into R
er = xmlParse("http://www.ecb.europa.eu/stats/eurofxref/eurofxref-hist.xml")
```

```
# Get the root node of the XML tree
erRoot = xmlRoot(er)

xmlSize(erRoot)
```

```
## [1] 3
```

```
names(erRoot)
```

```
##   subject    Sender      Cube
## "subject"  "Sender"    "Cube"
```

We'll extract the exchange rates for the British pound (GBP) as an example. First, we need to understand more about the structure of this XML tree. It looks like all of the data are descendents of a "Cube" node.

Here, we examine the first child of the Cube node. If you were to draw a diagram for this tree, with the following expression, we would be looking at the following node:

Envelope > Cube > Cube

```
erRoot[['Cube']][1]
```

```
## $Cube
## <Cube time="2017-04-13">
##   <Cube currency="USD" rate="1.063"/>
##   <Cube currency="JPY" rate="116.01"/>
##   <Cube currency="BGN" rate="1.9558"/>
##   <Cube currency="CZK" rate="26.704"/>
##   <Cube currency="DKK" rate="7.4376"/>
```

```
##   <Cube currency="GBP" rate="0.84763"/>
##   <Cube currency="HUF" rate="312.55"/>
##   <Cube currency="PLN" rate="4.245"/>
##   <Cube currency="RON" rate="4.5189"/>
##   <Cube currency="SEK" rate="9.582"/>
##   <Cube currency="CHF" rate="1.0686"/>
##   <Cube currency="NOK" rate="9.1033"/>
##   <Cube currency="HRK" rate="7.4255"/>
##   <Cube currency="RUB" rate="59.935"/>
##   <Cube currency="TRY" rate="3.8991"/>
##   <Cube currency="AUD" rate="1.4027"/>
##   <Cube currency="BRL" rate="3.3277"/>
##   <Cube currency="CAD" rate="1.4069"/>
##   <Cube currency="CNY" rate="7.3227"/>
##   <Cube currency="HKD" rate="8.2652"/>
##   <Cube currency="IDR" rate="14091.66"/>
##   <Cube currency="ILS" rate="3.8841"/>
##   <Cube currency="INR" rate="68.4915"/>
##   <Cube currency="KRW" rate="1202.93"/>
##   <Cube currency="MXN" rate="19.7661"/>
##   <Cube currency="MYR" rate="4.6905"/>
##   <Cube currency="NZD" rate="1.5178"/>
##   <Cube currency="PHP" rate="52.584"/>
##   <Cube currency="SGD" rate="1.4845"/>
##   <Cube currency="THB" rate="36.503"/>
##   <Cube currency="ZAR" rate="14.3805"/>
## </Cube>
##
## attr(,"class")
## [1] "XMLInternalNodeList" "XMLNodeList"
```

As we can see, this node is the parent of another node named Cube that has a "time" attribute. In turn, the Cube node with the "time" attribute node is the parent of several other Cube nodes with "currency" and "rate" attributes.

We'll show you two ways to pull out the GBP exchange rate.

First way: Split into two steps. 1. The first is to obtain all of the nodes that have the name "Cube" and an attribute "currency" set to GBP.

We provide getNodeSet() the root node and the XPath to the nodes we're interested in. Provide the predicate in the XPath expression to locate those Cube nodes with a currency attribute value of "GBP"

```
gbp_nodes = getNodeSet(erRoot, '//x:Cube[@currency="GBP"]',
                       namespaces = "x")
```

Note we haven't covered namespaces, so don't worry about the "x:" part in the specification of the path above.

Now for the second step. 2. Use xmlSApply() to get the value of the "rate" attribute on those nodes.

```
gbp1 = xmlSApply(gbp_nodes, xmlGetAttr, "rate")
```

Let's check it out. Do the values for 4/2 and 4/1 match the values you found by inspection of the XML file?

```
head(gbp1)
```

```
## [1] "0.84763" "0.8484"  "0.8533"  "0.85335" "0.85573" "0.8559"
```

```
length(gbp1)
```

```
## [1] 4684
```

```
class(gbp1)
```

```
## [1] "character"
```

ANSWER THE QUESTIONS: *Do the values match what you found earlier?* Yes *How many dates are there?* 4162 *What's the correct data type for this vector?* numeric

The second approach does this all in one step with the xPathSApply() function. Note that this function is similarly spelled, but xmlSApply() and xpathSApply() are not the same. The latter provides an XPath expression to specify the nodes to which the function is applied.

```
gbp2 = xpathSApply(erRoot, '//x:Cube[@currency="GBP"]',
                   xmlGetAttr, "rate", namespaces = "x")
```

It is easy to check that these two methods produce the same result!

```
identical(gbp1, gbp2)
```

```
## [1] TRUE
```

Now that we've extracted the GBP exchange rate, let's get the dates associated with each of those values. YOUR TURN: Fill in the XPAth expression to locate all Cube nodes with a time attribute. Remeber to use x: in front of the Cube tag name.

```
days = xpathSApply(erRoot, '//x:Cube[@time]',
                    xmlGetAttr, "time", namespaces = "x")
```

Let's take a look at our work and reformat the return object so that it's a Date object:

```
head(days)
```

```
## [1] "2017-04-13" "2017-04-12" "2017-04-11" "2017-04-10" "2017-04-07"
## [6] "2017-04-06"
```

```
dayze = as.Date(days)
```

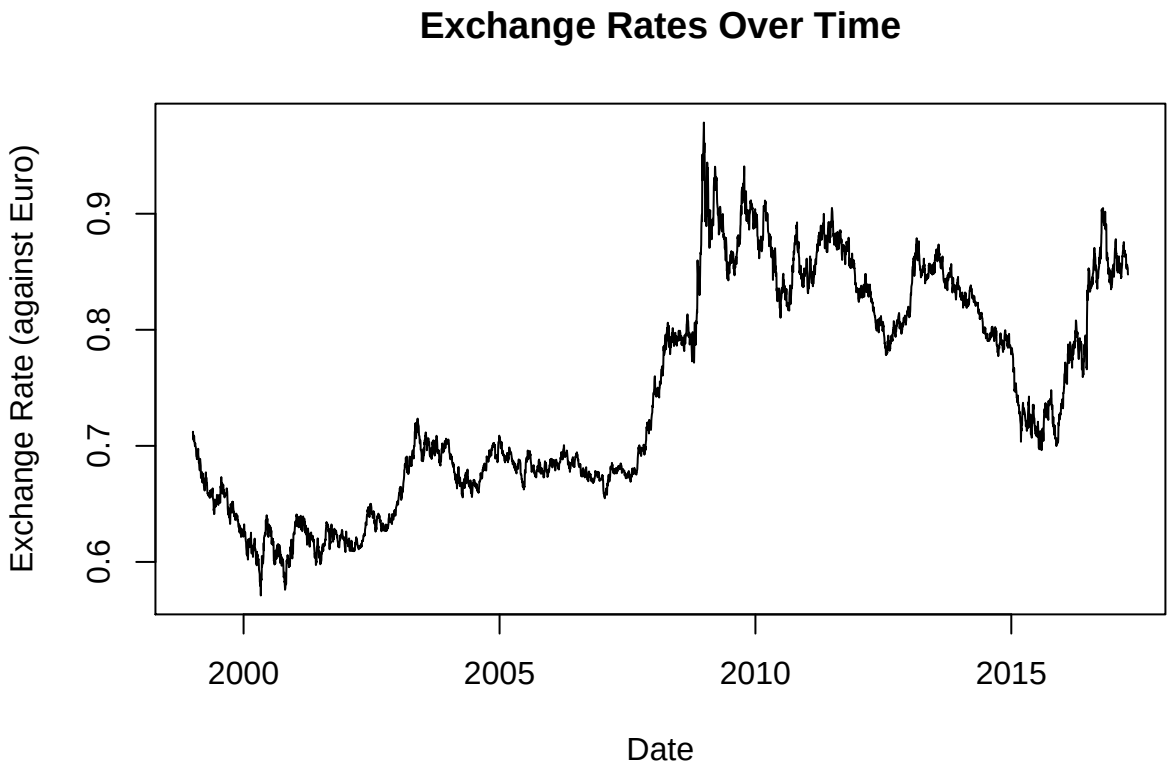
For a quick sanity check: did we pull out many days as there are exchange rates?

```
length(dayze) == length(gbp1)
```

```
## [1] TRUE
```

This should be TRUE.

Great! Now we've successfully extracted the exchange rates and dates for the British pound. We can create our first simple plot.



Extract rates for three other currencies

We gave you an example with the British pound. Now, we extract the exchange rates for other currencies and add it to the line plot we just created.

Rather than repeating code, we write a helper function to do the extraction. This function takes the root node of the XML document, the three letter currency abbreviation as a character string, and an argument that specifies the number of days that should be present in the document. It returns a numeric vector with the daily exchange rates for the associated currency.

```
getExchangeRates = function(abbrev,root, numDays = NULL ){
  # get rates
  currency_path = sprintf('//x:Cube[@currency="%s"]',
                          abbrev)
  rates = xpathSApply(root, currency_path,
                      xmlGetAttr, "rate", namespaces = "x")

  # format as numeric vector:
  rates = as.numeric(rates)
```

```
  # Check to see if there are any missing rates.
  # print a warning
  if (!is.null(numDays)) {
    if(length(rates) != numDays) {
      warning("Incorrect number of dates!")
      }
  }
  return(rates)
}
```

Apply your function to get the exchange rates for the GBP, USD, CAD, and JPY. Fill in the follwing call to lapply:

```
currencies = lapply(X=c("GBP","USD","CAD","JPY"),
                    FUN=function(x){
                      getExchangeRates(x, root=erRoot,
                                       numDays=length(dayze))})
```

Now we are ready to make our plot!

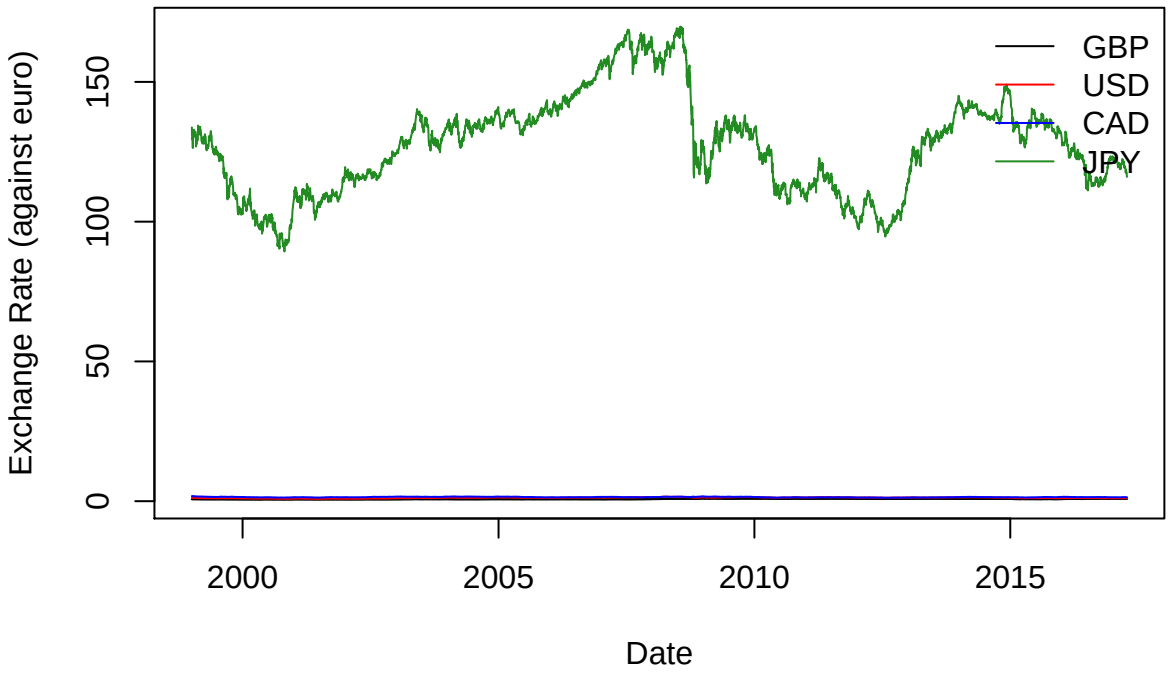
```
# Find the range of these exchange rates
rateMin = min(sapply(currencies, min))
rateMax = max(sapply(currencies, max))
rateRange = c(rateMin, rateMax)

plot(x = dayze, y = currencies[[1]], typ = "l",
     ylim=rateRange,
     ylab="Exchange Rate (against euro)",
     xlab="Date")

# Add the lines for the countries
mapply(function(rateVec, rateCol) {
  lines(rateVec ~ dayze, col= rateCol)
  }, rateVec = currencies,
  rateCol = c("black", "red", "blue", "forestgreen"))
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
```

```
# Add a legend
legend("topright", legend=c("GBP", "USD", "CAD", "JPY"),
       col=c("black", "red", "blue", "forestgreen"),
       lwd=1, bty="n")
```



OPTIONAL FOR FUN:

The currency for China (CNY) isn't available for all the dates in the file. Use the getExchangeRates function to extract the CNY rates. How many rates are in this vector?

2563

Write another extraction to get the dates of those Cube nodes where the CNY rate is present. One way to do this is to locate the CNY Cube nodes and then add one more step to the path to travel back up the tree one step to the parent Cube. To do this, remember that the step consists of an axis followed by :: followed by a node name and then an optional predicate. The axis "parent" is a valid axis.

```
cny = getExchangeRates(root = erRoot, abbrev = "CNY")

cnyDays = xpathSApply(erRoot, '//x:Cube[@currency="CNY"]/parent::x:Cube',
                      xmlGetAttr, "time",
                      namespaces = "x")
```