

```
# HW 7 - Due Monday Nov 13, 2017 in moodle and hardcopy in class.
# Upload R file to Moodle with filename: HW7_490IDS_41.R
# Do not remove any of the comments. These are marked by #
```

```
### For this assignment will extract useful information from XML and
### use Google Earth for data visualization.
### The hw7.rda file containing the contry geographic coordinate is uploaded to Moodle.
```

```
### Look at detail instructions for the assignment in hw7_Intro.pdf.
```

```
### Part 1. Create the data frame from XML file
```

```
### Functions you'll want to use: xmlParse(), xmlRoot(), xpathApply(), xmlGetAttr().
### It also might make it easier to use: xmlToList(), merge().
```

```
### (a) Load the data frame called LatLon from hw7.rda.
```

```
load("hw7.rda")
head(LatLon)
```

	Country.Name	CIA.Codes	Latitude	Longitude
1	AFGHANISTAN	AF	33.0000	65.0
2	ALBANIA	AL	41.0000	20.0
3	ALGERIA	AG	28.0000	3.0
4	AMERICAN SAMOA	AQ	-14.3333	-170.0
5	ANDORRA	AN	42.5000	1.5
6	ANGOLA	AO	-12.5000	18.5

```
### (b) Download the gzipped XML factbook document from
### http://jmatchparser.sourceforge.net/factbook/
```

```
### and create an XML "tree" in R
library("XML")
data = xmlParse("factbook.XML")
data.tree = xmlRoot(data)
class(data.tree)
[1] "XMLInternalElementNode" "XMLInternalNode" "XMLAbstractNode"
```

```
### (c) Use XPath to extract the infant mortality and the CIA country codes from the XML
tree
```

To be able to find the path, I viewed the XML file in my browser and searched for the key word "Infant mortality rate" to find the names associated to the information. (Is there a way to do this in R?)

```
-<field dollars="false" unit="(deaths/1,000 live births)" rankorder="1" name="Infant mortality rate" id="f2091">
-<description>
  This entry gives the number of deaths of infants under one year old in a given year per 1,000 live births in the same year. This rate is often used as an indicator of the level of health in a country.
</description>
<rank number="117.23" dateEstimated="true" dateLatest="2014-12-31" dateEarliest="2014-01-01" dateText="2014 est." country="af"/>
<rank number="104.34" dateEstimated="true" dateLatest="2014-12-31" dateEarliest="2014-01-01" dateText="2014 est." country="ml"/>
<rank number="100.14" dateEstimated="true" dateLatest="2014-12-31" dateEarliest="2014-01-01" dateText="2014 est." country="so"/>
<rank number="92.86" dateEstimated="true" dateLatest="2014-12-31" dateEarliest="2014-01-01" dateText="2014 est." country="ct"/>
<rank number="90.92" dateEstimated="true" dateLatest="2014-12-31" dateEarliest="2014-01-01" dateText="2014 est." country="pu"/>
```

```
extracted.data = sapply(c("number","country"),function(variable) xpathApply(data.tree, "
//field[@name='Infant mortality rate']/rank", function(x) xmlGetAttr(x,variable)))
head(extracted.data)
```

```
      number  country
[1,] "117.23" "af"
[2,] "104.34" "ml"
[3,] "100.14" "so"
```

```
[4,] "92.86" "ct"
[5,] "90.92" "pu"
[6,] "90.30" "cd"
```

(d) Create a data frame called IM using this XML file.
 ### The data frame should have 2 columns: for Infant Mortality and CIA.Codes.

```
IM = data.frame("Infant Mortality"= as.numeric(extracted.data[,1]),"CIA.Codes" = toupper(
extracted.data[,2]))
head(IM)
```

	Infant.Mortality	CIA.Codes
1	117.23	AF
2	104.34	ML
3	100.14	SO
4	92.86	CT
5	90.92	PU
6	90.30	CD

(e) Extract the country populations from the same XML document
 ### Create a data frame called Pop using these data.
 ### This data frame should also have 2 columns, for Population and CIA.Codes.

```
extracted.pop.data = sapply(c("number","country"),function(variable) xpathSapply(data.tre
e, "//field[@name='Population']/rank", function(x) xmlGetAttr(x,variable)))
head(extracted.pop.data)
```

	number	country
[1,]	"1355692576"	"ch"
[2,]	"1236344631"	"in"
[3,]	"511434812"	"ee"
[4,]	"318892103"	"us"
[5,]	"253609643"	"id"
[6,]	"202656788"	"br"

```
Pop = data.frame("Population"= as.numeric(extracted.pop.data[,1]),"CIA.Codes" = toupper(e
xtracted.pop.data[,2]))
head(Pop)
```

	Population	CIA.Codes
1	1355692576	CH
2	1236344631	IN
3	511434812	EE
4	318892103	US
5	253609643	ID
6	202656788	BR

(f) Merge the two data frames to create a data frame called IMPop with 3 columns:
 ### IM, Pop, and CIA.Codes

```
IMPop <- merge(x=IM, y=Pop, by=intersect(names(IM), names(Pop)))
head(IMPop)
```

	CIA.Codes	Infant.Mortality	Population
1	AA	11.74	110663
2	AC	13.29	91295
3	AE	10.92	5628805
4	AF	117.23	31822848
5	AG	21.76	38813722
6	AJ	26.67	9686210

(g) Now merge IMPop with LatLon (from newLatLon.rda) to create a data frame called AllData that has 6 columns
 ### for Latitude, Longitude, CIA.Codes, Country Name, Population, and Infant Mortality
 ### (please check lat,long are not reversed in the file)

```
AllData = merge(x=IMPop, y=LatLon, by=intersect(names(IMPop), names(LatLon)))
names(AllData)
```

```
[1] "CIA.Codes"          "Infant.Mortality" "Population"          "Country.Name"        "Latitude"
[6] "Longitude"
```

```
### Part 2. Create a KML document for google earth visualization.
### Make the KML document with stucture described in hw7_Intro.pdf. You can use the addPlacemark function below to make
### the Placemark nodes, for which you need to complete the line for the Point node and
### figure out how to use the function.
```

```
makeBaseDocument = function(){
+ ### This code creates the template for KML document
+ ### Your code here
+
+ doc.XML = newXMLDoc()
+ root = newXMLNode("kml", doc = doc.XML, namespaceDefinitions = "http://www.opengis.net/kml/2.2")
+ Document_Node= newXMLNode("Document", parent = root)
+
+ LookAt_Node = newXMLNode("LookAt", parent = root)
+
+ newXMLNode(name="Name", "Country Facts", parent = Document_Node)
+ newXMLNode(name="Description", "Infant Mortality", parent = Document_Node)
+
+
+ Folder_Node = newXMLNode(name="Folder", parent = Document_Node)
+ newXMLNode(name="Name", "CIA Factbook", parent = Folder_Node )
+
+ return(doc.XML)
+ }
```

```
addPlacemark = function(lat, lon, ctryCode, ctryName, pop, infM, parent,
+                         inf1, pop1, style = FALSE)
+ {
+   pm = newXMLNode("Placemark",
+                   newXMLNode("name", ctryName), attrs = c(id = ctryCode),
+                   parent = parent)
+   newXMLNode("description", paste(ctryName, "\n Population: ", pop,
+                                   "\n Infant Mortality: ", infM, sep = ""),
+               parent = pm)
+
+   newXMLNode("Point" , newXMLNode("coordinates", paste(lon, ", ", lat, ", ", 0, sep = ""))
+ ), parent=pm)
+
+ ### You need to fill in the code for making the Point node above, including coordinates
+
+ ### The line below won't work until you've run the code for the next section to set up
+ ### the styles.
+
+   if(style) newXMLNode("styleUrl", paste("#YOR", inf1, "-", pop1, sep = ''), parent = pm)
+ }
```

```
### Use the two functions that you just implemented to created the KML document and save it
### as 'Part2.kml'. open it in Google Earth. (You will need to install Google Earth.)
### It should have pushpins for all the countries.
```

```
### Your code here
```

```
# We start by making the initial structure
```

```
Part2.kml = makeBaseDocument()
```

```
Part2.root = xmlRoot(Part2.kml)
Part2.children = xmlChildren(Part2.kml)
```

#We check the structure of the file
Part2.kml

```
<?xml version="1.0"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Name>Country Facts</Name>
    <Description>Infant Mortality</Description>
    <Folder>
      <Name>CIA Factbook</Name>
    </Folder>
  </Document>
</LookAt/>
</kml>
```

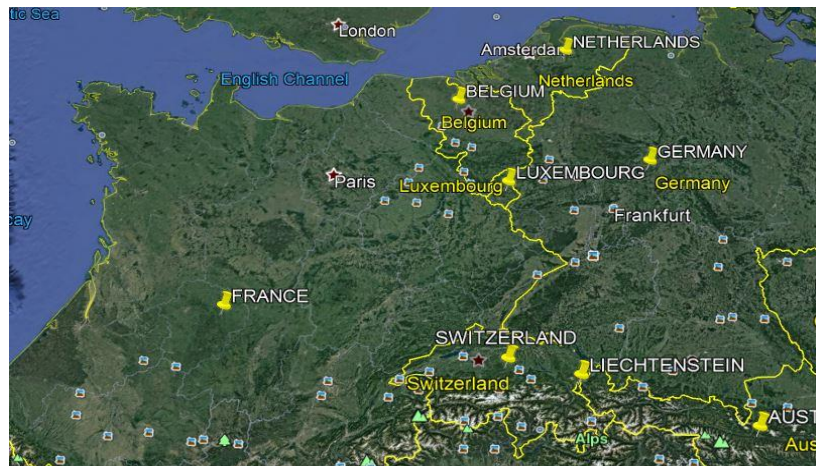
```
Document_Node2 <- Part2.children[[1]][[1]]
Document_Node2
```

```
<Document>
  <Name>Country Facts</Name>
  <Description>Infant Mortality</Description>
  <Folder>
    <Name>CIA Factbook</Name>
  </Folder>
</Document>
```

#Now actually incorporate the data.

```
for(i in 1:(dim(AllData)[1])) {
+   addPlacemark(lat=AllData$Latitude[[i]], lon=AllData$Longitude[[i]], ctryCode=AllData$
CIA.Codes[[i]], ctryName=AllData$Country.Name[[i]],
+               pop=AllData$Population[[i]], infM=AllData$Infant.Mortality[[i]], parent=
Document_Node2)
+ }
saveXML(doc=Part2.kml, file="Part2.kml")
[1] "Part2.kml"
```

Here is how the file looks like when loaded to Google Earth.



```
### Part 3. Add style to your KML
### Now you are going to make the visualization a bit fancier. To be more specific, inst
ead of pushpins, we
### want different circle labels for countris with size representing population and the c
olor representing
```

```
### the infant motality rate.
### Pretty much all the code is given to you below to create style elements.
### Here, you just need to figure out what it all does.
```

```
### Start fresh with a new KML document, by calling makeBaseDocument()
```

```
doc2 = makeBaseDocument()
Part3.root = xmlRoot(doc2)
Part3.children = xmlChildren(doc2)
doc2
<?xml version="1.0"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Name>Country Facts</Name>
    <Description>Infant Mortality</Description>
    <Folder>
      <Name>CIA Factbook</Name>
    </Folder>
  </Document>
  <LookAt/>
</kml>
```

```
Document_Node3 <- Part3.children[[1]][[1]]
```

```
### The following code is an example of how to create cut points for
### different categories of infant mortality and population size.
### Figure out what cut points you want to use and modify the code to create these
### categories.
```

```
infCut = cut(AllData$Infant.Mortality, breaks = c(0, 10, 25, 50, 75, 200))
infCut = as.numeric(infCut)
summary(infCut)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	1.000	2.000	2.167	3.000	5.000

```
popCut = cut(log(AllData$Population), breaks = 5) #I used the log in this case since othe
rwise the cuts were uneven
popCut = as.numeric(popCut)
summary(popCut)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	2.000	3.000	3.123	4.000	5.000

```
### Now figure out how to add styles and placemarks to doc2
### You'll want to use the addPlacemark function with style = TRUE
### Below is code to make style nodes.
### You should not need to do much to it.
```

```
### You do want to figure out what scales to use for the sizes of your circles. Try diffe
rent
### setting of scale here.
```

```
scale = c(0.5,1,1.5,2,2.5) #Try your scale here for better visualization
colors = c("blue","green","yellow","orange","red")
```

```
addStyle = function(col1, pop1, parent, DirBase, scales = scale)
+ {
+   st = newXMLNode("Style", attrs = c("id" = paste("YOR", col1, "-", pop1, sep="")), par
ent = parent)
+   newXMLNode("IconStyle",
+             newXMLNode("scale", scales[pop1]),
+             newXMLNode("Icon", paste(DirBase, "color_label_circle_", colors[col1], ".p
ng", sep = "")), parent = st)
+ }
```



```

root2 = xmlRoot(doc2)
DocNode = root2[["Document"]]

for (k in 1:5)
+ {
+   for (j in 1:5)
+   {
+     addStyle(j, k, Document_Node3, 'C:\\Users\\Vanessa\\Box Sync\\Classes Fall 2017\\IN
FO 490\\Homework\\HW7\\', scales = scale)
+   }
+ }

```

You will need to figure out what order to call addStyle() and addPlacemark()
 ### so that the tree is built properly. You may need to adjust the code to call the png f
 iles
 ### Your code here

```

for(i in 1:(dim(AllData)[1])) {
+
+   addPlacemark(lat=AllData$Latitude[[i]], lon=AllData$Longitude[[i]], ctryCode=AllData$
CIA.Codes[[i]], ctryName=AllData$Country.Name[[i]],
+     pop=AllData$Population[[i]], infM=AllData$Infant.Mortality[[i]], parent=
Document_Node3, inf1=infCut[[i]], pop1=popCut[[i]],
+     style=TRUE)
+ }

```

Finally, save your KML document, call it Part3.kml and open it in Google Earth to
 ### verify that it works. For this assignment, you only need to submit your code,
 ### nothing else. You can assume that the grader has already loaded hw7.rda.

```

saveXML(doc=doc2, file="Part3.kml")
[1] "Part3.kml"

```

Here is how the file looks like when loaded to Google Earth.

