```
# HW 7 - Due Monday Nov 13, 2017 in moodle and hardcopy in class.
# Upload R file to Moodle with filename: HW7_490IDS_52.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 country 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
library(XML)
### Functions you'll want to use: xmlParse(), xmlRoot(), xpathSApply(), xmlGetAttr().
### It also might make it easier to use: xmlToList(), merge().

### (a) Load the data frame called LatLon from hw7.rda.
## Setting my own working directory # commented it out so if others run it there won't be an error
#setwd("./Desktop/Fall 2017/IS 490 Introduction to Data Science/Assignments/HW7")
load("hw7.rda")
### (b) Download the gzipped XML factbook document from
### http://jmatchparser.sourceforge.net/factbook/
### and create an XML "tree" in R
factbook = xmlParse("factbook.xml")
root = xmlRoot(factbook)
factbook_list = xmlToList(factbook)


### (c) Use XPath to extract the infant mortality and the CIA country codes from the XML tree
getNodeSet(root, '//field[@name = "Infant mortality rate"]')
infant_MR = as.numeric(xpathSApply(factbook, '//field[@name = "Infant mortality rate"]/rank',
xmlGetAttr, "number"))
CIA_Country_code = xpathSApply(factbook, '//field[@name="Infant mortality rate"]/rank',
xmlGetAttr, "country")

### (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" = infant_MR, "CIA.Codes" = CIA_Country_code)

### (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.
getNodeSet(root, '//field[@name = "Population"]/rank')
country_pop = as.numeric(xpathSApply(factbook, '//field[@name = "Population"]/rank',
xmlGetAttr, "number"))
country_code = xpathSApply(factbook, '//field[@name = "Population"]/rank', xmlGetAttr,
"country")
Pop = data.frame("Population" = country_pop, "CIA.Codes" = country_code)

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

### (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)
names(LatLon)
LatLon
IMPop$CIA.Codes = toupper(IMPop$CIA.Codes)
```

```
AllData = merge(IMPop, LatLon, by="CIA.Codes")
names(AllData)

### 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 = newXMLDoc()
  root = newXMLNode("kml", namespaceDefinitions = c("http://www.opengis.net/kml/2.2 "), doc =
doc)
  Document = newXMLNode("Document", parent = root)
  newXMLNode("Name", "Country Facts", parent = Document)
  newXMLNode("Description", "Infant Mortality", parent = Document)
  LookAt = newXMLNode("LookAt", parent = Document)
  Folder = newXMLNode("Folder", parent = Document)
  newXMLNode("Name", "CIA Fact Book", parent = Folder)
  newXMLNode("longtitude", "-121", parent = LookAt)
  newXMLNode("latitude", "-43", parent = LookAt)
  newXMLNode("altitude", "-4100000", parent = LookAt)
  newXMLNode("title", "0", parent = LookAt)
  newXMLNode("heading", "0", parent = LookAt)
  newXMLNode("altitudeMode", "absolute", parent = LookAt)


  return(doc)
}


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.
KML = makeBaseDocument()
KML_nodes = xmlChildren(KML)
Document_tag = KML_nodes[[1]][[1]]

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_tag)
}

saveXML(KML, "Part2.kml")

### Part 3.  Add Style to your KML
### Now you are going to make the visualizatiion a bit fancier. To be more specific, instead of
pushpins, we
### want different circle labels for countris with size representing population and the color
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()

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

pop_Coun = as.numeric(AllData$Population)
popCut = cut(pop_Coun, breaks = 5 )
popCut = as.numeric(popCut)

### 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 different
### setting of scale here.

# scale = c(XX,XX,XX,XX,XX) Try your scale here for better visualization

scale = c(3,4,5.5,7,9.5)


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="")), parent =
parent)
  newXMLNode("IconStyle",
        newXMLNode("scale", scales[pop1]),
        newXMLNode("Icon", paste(DirBase, "color_label_circle_", colors[col1], ".png", sep ="")),
parent = st)
}


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


for (k in 1:5)
{
 for (j in 1:5)
 {
  addStyle(j, k, DocNode, 'color_label_circle/')
 }
}

### 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 files
### 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 = DocNode, inf1 = infCut[i], pop1 = popCut[i], style = T)
}

### 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(DocNode, "Part3.kml")
```