```
# HW 8 - Due Tuesday Nov 8, 2016 in moodle and hardcopy in class.
# Upload R file to Moodle with filename: HW8_490IDS_YOURID.R
# Do not remove any of the comments. These are marked by #

### This assignment will use Google Earth for data display.
### The .rda file is uploaded to Moodle.

### Load HW8.rda and attach the XML library

load(file = "C:/Documents/GSLIS/490 Introduction to Data Science/HW8/hw8.rda")
library("XML", lib.loc="C:/Program Files for operation/R-3.3.1/library")

### Part 1.  Create the data frame
### Look at the instructions in HW8.pdf.
### Functions you'll want to use: xmlParse(), xmlRoot(), xpathSApply(), xmlGetAttr().
### It also might make it easier to use: xmlToList(), merge().

### Load the data frame called LatLon from HW8.rda.

### Download the gzipped XML factbook document from
### http://jmatchparser.sourceforge.net/factbook/
### and create an XML "tree" in R

doc = xmlParse("C:/Documents/GSLIS/490 Introduction to Data Science/HW8/factbook.xml")
factbook = xmlRoot(doc)

### Use XPath to extract the infant mortality and the CIA country codes from the XML tree
###
infant.mortality.number = xpathSApply(factbook, "//field[@name='Infant mortality rate']/rank", function(x) xmlGetAttr(node = x, name = "number"))
infant.mortality.country = xpathSApply(factbook, "//field[@name='Infant mortality rate']/rank", function(x) xmlGetAttr(node = x, name = "country"))
```

```
### 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.mortality.number, CIA.Codes = infant.mortality.country)

### 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.
population.number = xpathSApply(factbook, "//field[@name='Population']/rank", function(x) xmlGetAttr(node = x, name = "number"))
population.country = xpathSApply(factbook, "//field[@name='Population']/rank", function(x) xmlGetAttr(node = x, name = "country"))
Pop = data.frame(Population = population.number, CIA.Codes = population.country)

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

### 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)
IMPop$CIA.Codes = toupper(IMPop$CIA.Codes)
AllData = merge(IMPop, LatLon)

### Part 2.  Create a KML document
### Make the KML document described in HW8.pdf.  It should have the basic
### structure shown in that document.  You can use the addPlacemark function below to make
### the Placemark nodes, you just need to complete the line for the Point node and
### figure out how to use the function.

makeBaseDocument = function(){
### This code creates the template KML document
```

```
  doc = newXMLDoc()
  root = newXMLNode("kml", doc = doc, namespaceDefinitions = "http://www.opengis.net/kml/2.2")
  document = newXMLNode("Document", parent = root)
  child1.1 = newXMLNode("Name", "Country Facts", parent = document)
  child1.2 = newXMLNode("Description", "Infant Mortality", parent = document)
  child1.3 = newXMLNode("LookAt", parent = document)
  child1.4 = newXMLNode("Folder", parent = document)
  child2.1 = newXMLNode("longitude", "-121", parent = child1.3)
  child2.2 = newXMLNode("latitude", "43", parent = child1.3)
  child2.3 = newXMLNode("altitude", "4100000", parent = child1.3)
  child2.4 = newXMLNode("title", "0", parent = child1.3)
  child2.5 = newXMLNode("heading", "0", parent = child1.3)
  child2.6 = newXMLNode("altitudeMode", "absolute", parent = child1.3)
  child2.7 = newXMLNode("Name", "CIA Fact Book", parent = child1.4)

  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, 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)

}


doc1 = makeBaseDocument()

root.doc1 = xmlRoot(doc1)

for (i in 1:203) {

 addPlacemark(lat = AllData$Latitude[i], lon = AllData$Longitude[i], ctryCode = AllData$CIA.Codes[i], ctryName = AllData$Country.Name[i], pop = AllData$Pop[i], infM = AllData$IM[i], parent = root.doc1[[1]][['Folder']])


}


### Save your KML document here, call it Part2.kml, and open it in Google Earth.

### (You will need to install Google Earth.)

### It should have pushpins for all the countries.


saveXML(doc1, file = "Part2.kml")


### Part 3.  Add Style to your KML

### Now you are going to make the visualizatiion a bit fancier.  Pretty much all the code is given to you

### below to create style elements that are to be placed near the top of the document.

### These , 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.

# infCut = cut(x[,5], breaks = c(0, 10, 25, 50, 75, 200))
# I modify the interval boundaries to make the classification more average based on infant mortality.

infCut = cut(as.numeric(as.character(AllData$IM)), breaks = c(0, 5, 10, 20, 50, 200))
infCut = as.numeric(infCut)
table(infCut)
# infCut
# 1  2  3  4  5
# 38 37 46 47 35

# popCut = cut(log(x[,6]), breaks = 5)
popCut = cut(log(as.numeric(as.character(AllData$Pop))), breaks = 5)
popCut = as.numeric(popCut)
table(popCut)
# popCut
# 1  2  3  4  5
# 20 32 65 75 11

### 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 you for the sizes of your circles
scales1 = c(0.5, 1, 3, 5, 10)
color = c("blue","green","orange","red","yellow")

addStyle = function(inf1, pop1, parent, urlBase, scales = scales1)
{
```

```
  st = newXMLNode("Style", attrs = c("id" = paste("YOR", inf1, "-", pop1, sep="")), parent = parent)
  newXMLNode("IconStyle",
                                newXMLNode("scale", scales[pop1]),
                          newXMLNode("Icon",  paste(urlBase,  "color_label_circle_",  color[inf1],  ".png",
sep =")), parent = st)
}

root.doc2 = xmlRoot(doc2)



for (k in 1:5)
{
  for (j in 1:5)
  {
    addStyle(j, k, parent = root.doc2[[1]][['Folder']], 'http://www.stanford.edu/~vcs/StatData/circles/')
  }
}

for (i in 1:203) {
  addPlacemark(lat = AllData$Latitude[i], lon = AllData$Longitude[i], ctryCode = AllData$CIA.Codes[i], ctryName = AllData$Country.Name[i], pop = AllData$Pop[i], infM = AllData$IM[i], parent = root.doc2[[1]][['Folder']], inf1 = infCut[i], pop1 = popCut[i], style = TRUE)

}

### 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

### 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 HW8.rda.

saveXML(doc2, file = "Part3.kml")
```