SkylerRoh_HW6_XML

Skyler Roh 23313980

April 9, 2015

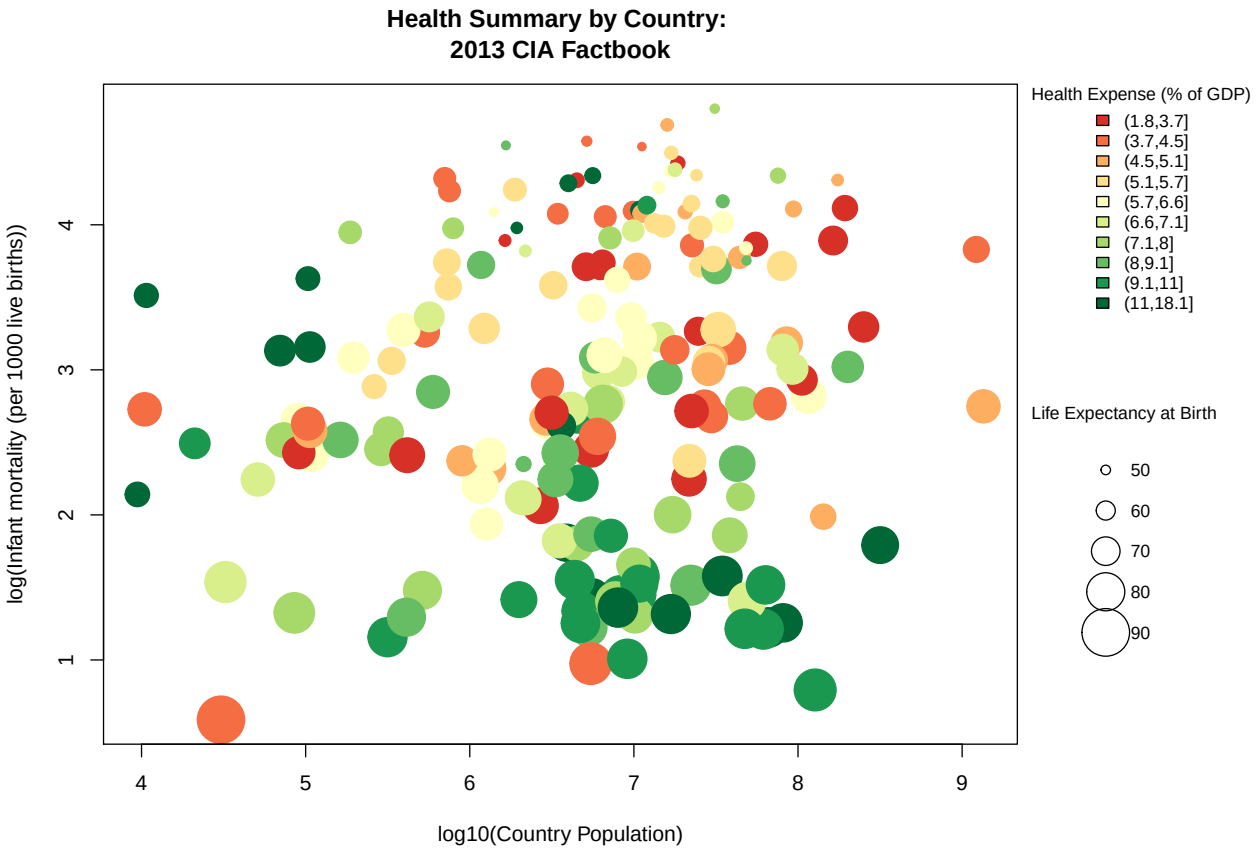
```
library(XML)
factbook = xmlParse("http://www.stat.berkeley.edu/users/nolan/data/factbook/factbook.xml")
factbookRoot = xmlRoot(factbook)
```

```
#Infant Mortality
infantMortality = as.numeric(xpathSApply(factbook,
                                         '//field[@name = "Infant mortality rate"]/rank',
                                         xmlGetAttr, 'number'))
countryCodes1 = xpathSApply(factbook, '//field[@name = "Infant mortality rate"]/rank',
                            xmlGetAttr, 'country')
IM = data.frame(infantMortality, countryCode = countryCodes1, stringsAsFactors = FALSE)

#Population
countryPop = as.numeric(xpathSApply(factbook, '//field[@name = "Population"]/rank',
                                         xmlGetAttr, 'number'))
countryCodes2 = xpathSApply(factbook, '//field[@name = "Population"]/rank',
                            xmlGetAttr, 'country')
populations = data.frame(countryPop, countryCode = countryCodes2, stringsAsFactors = FALSE)

#Life Expectancy
countryLifeExpect = as.numeric(xpathSApply(factbook,
                                           '//field[@name = "Life expectancy at birth"]/rank',
                                           xmlGetAttr, 'number'))
countryCodes3 = xpathSApply(factbook, '//field[@name = "Life expectancy at birth"]/rank',
                            xmlGetAttr, 'country')
lifeExpectancy = data.frame(countryLifeExpect, countryCode = countryCodes3, stringsAsFactors = FALSE)

#Health Expenditure
percentGDP = as.numeric(xpathSApply(factbook, '//field[@name = "Health expenditures"]/rank',
                                                 xmlGetAttr, 'number'))
countryCodes4 = xpathSApply(factbook, '//field[@name = "Health expenditures"]/rank',
                            xmlGetAttr, 'country')
healthExpense = data.frame(percentGDP, countryCode = countryCodes4, stringsAsFactors = FALSE)
```

```
#combine data frames by matching country codes
variables = list(IM, populations, lifeExpectancy, healthExpense)
allVar = Reduce(merge, variables)
```

```
#color
library(RColorBrewer)
spectrum = brewer.pal(11, "RdYlGn")[-1]
healthGDPquantiles = quantile(allVar$percentGDP, probs = seq(from = 0, to = 1, by = .1))
roundedGDP = round(healthGDPquantiles, digits = 1)
healthGDPgradient = cut(allVar$percentGDP, breaks = roundedGDP)
healthGDPColors = spectrum[healthGDPgradient]
```

```
par(mar = c(5, 4, 4, 10))
plot(x = log10(allVar$countryPop), y = log(allVar$infantMortality), pch = 19,
     cex = (allVar$countryLifeExpect - 40)/10,
     col = healthGDPColors,
     xlab = "log10(Country Population)",
     ylab = "log(Infant mortality (per 1000 live births))",
     main = "Health Summary by Country: \n2013 CIA Factbook")
legend(x = 9.4, y = 5, title = "Health Expense (% of GDP)", fill = spectrum,
       legend = levels(healthGDPgradient),
       cex = .8, bty = "n", xpd = TRUE)
legend(x = 9.4, y = 2.8, title = "Life Expectancy at Birth",
       legend = c("  50", "  60", "  70", "  80", "  90"),
       bty = "n", cex = .8, pch = 21, pt.cex = (5:9*10 - 40)/10, y.intersp = 2, xpd = TRUE)
```



The plot that I created was based on four variables for each country that had all four pieces of data available: population, infant mortality rate, life expectancy at birth, and health expenditure as percent of GDP. As a scatter plot, the log of infant mortality was plotted against the log base 10 of population to create a visual that was fairly evenly distributed across the plotting space. Following, health expenditure was cut into approximate 10% quantiles and set as a gradient from red being the lowest to green being the highest %GDP. Lastly, size of each point is based off of the number of years past 40 that one is expected to live for each country.