Homework 01 R Basics

Due by 11:59pm, Friday, 1.24.25

S&DS 230/530/ENV 757

---

(1) RMarkdown Practice *(24 points)* Change the markdown code below as indicated.

Make this line bold

Make this line italics

Make this line a second level header

- Make this line a bullet point
  - Make this line an indented (or level two) bullet point

LINK (make the word LINK at left link to the New York Times home page AND make it bold)

```
Make this line look like R Code
```

Below this line, insert a new R chunk, create a vector called `xvec` that contains the integers 5 through 9, and have R display what is in `xvec`.

```
xvec <- 5:9
xvec
```

```
## [1] 5 6 7 8 9
```

(2) R Syntax Practice *(12 points)* Modify the R code below to follow good R Syntax practices

```
x <- 5

x <- c(1, 2, 3)

length(x)

for (i in 1:10) {
  x <- 1+1
}

x <- 1
y <- c(3, 4)
```

(3) Data handling *36 pts*

(3.1) Insert a new R code chunk below.

```r
wb <- read.csv("http://reuningscherer.net/S&DS230/data/WB_2024.csv")
dim(wb)
```

```
## [1] 217  17
```

```r
names(wb)
```

```
##  [1] "Country"    "Population" "Rural"      "GNI"        "Imports"
##  [6] "Exports"    "Military"   "Cell"       "Fertility"  "Measles"
## [11] "InfMort"    "LifeExp"    "PM2.5"      "CO2"        "EnergyUse"
## [16] "Renewable"  "Debt"
```

```r
head(wb, 5)
```

```
##          Country Population  Rural   GNI  Imports  Exports Military      Cell
## 1    Afghanistan   42239854 73.067   360 37.06956 14.34215       NA  56.55443
## 2        Albania    2745972 35.397  6770 44.70882 31.30916 1.584881  92.31992
## 3        Algeria   45606480 24.732  4490 23.38840 23.88251 4.779438 106.42354
## 4 American Samoa      43914 12.765    NA 92.53333 44.26667       NA        NA
## 5        Andorra      80088 12.226 50080       NA       NA       NA 118.67298
##   Fertility Measles InfMort LifeExp     PM2.5         CO2 EnergyUse Renewable
## 1     4.523      68    44.8  62.879 46.087094 0.138000720      2.94      20.0
## 2     1.376      86     8.4  76.833 15.707004 1.615083618      2.27      41.9
## 3     2.829      79    18.7  77.129 25.552656 3.943578663      5.61       0.1
## 4        NA      NA      NA      NA  6.715147 0.002258713        NA       0.4
## 5        NA      98     2.6      NA  9.080281          NA      1.89      18.4
##        Debt
## 1        NA
## 2 56.302323
## 3  3.733707
## 4        NA
## 5        NA
```

```r
sapply(wb, class)
```

```
##     Country  Population       Rural         GNI     Imports     Exports
## "character"   "numeric"   "numeric"   "integer"   "numeric"   "numeric"
##    Military        Cell   Fertility     Measles     InfMort     LifeExp
##   "numeric"   "numeric"   "numeric"   "integer"   "numeric"   "numeric"
##       PM2.5         CO2   EnergyUse   Renewable        Debt
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
```

```r
# class(wb[, "GNI"])
wb_Subset <- wb[wb$GNI > 70000 & is.na(wb$GNI)==F,
                c("Country", "GNI", "EnergyUse", "Measles")]
wb_Subset
```

```
##           Country    GNI EnergyUse Measles
## 22        Bermuda 134640      1.52      NA
## 54        Denmark  73520      1.96      95
```

```
## 66  Faroe Islands  74420        NA      NA
## 89         Iceland  73930     12.33      91
## 94         Ireland  79730      1.09      90
## 117     Luxembourg  89200      1.98      99
## 148         Norway  96770      3.43      96
## 160          Qatar  70070      7.20      99
## 189    Switzerland  95490      1.53      96
## 207 United States  76590      4.24      92
```

```
wb_Stats <- summary(wb["Debt"], digits = 2)
wb_Stats
```

```
##       Debt
##  Min.   :  2.4
##  1st Qu.: 32.8
##  Median : 47.4
##  Mean   : 56.2
##  3rd Qu.: 66.4
##  Max.   :423.6
##  NA's   :102
```

```
length(wb_Stats)
```

```
## [1] 7
```

```
wb_Stats[c(1, 2, 3, 5, 6)]
```

```
## [1] "Min.   :  2.4  " "1st Qu.: 32.8  " "Median : 47.4  " "3rd Qu.: 66.4  "
## [5] "Max.   :423.6  "
```

(3.2) Read the .csv stored HERE into a new data frame and call is "wb". This is similar to the World Bank data I discussed in class two (this is a more current version). You can get the longer description of each variable HERE

(3.3) Get the dimension of wb.

(3.4) Get the variable names of wb.

(3.5) Show the first 5 lines of wb.

(3.6) Get the data type of each variable.

(3.7) In your code, insert a comment that gives the data type of the variable `GNI`?

(3.8) Create a new object called `wb_Subset` that has only the variables Country, GNI, EnergyUse, and Measles (in that order) AND only for countries where GNI is greater than 70000. Make sure you show the value of `wb_Subset`.

(3.9) Get summary statistics for `Debt`. Store the results in a new object called `wb_Stats`. Incidentally, `wb_Stats` will be a vector!

(3.10) Get the length of `wb_Stats`.

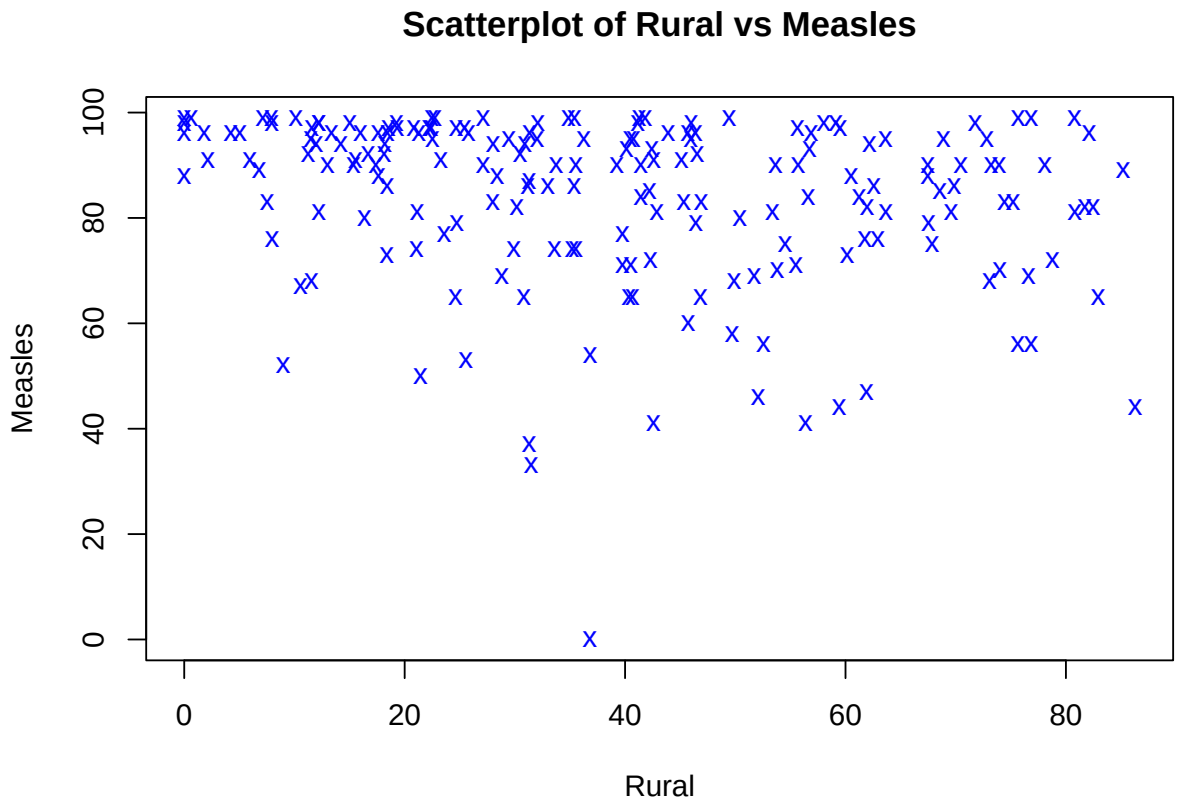
(3.11) Get r to show the following elements of `wb_Stats` : 1,2,3,5,6 AND round the result to 1 decimal place.

(4) Plots *16 pts*

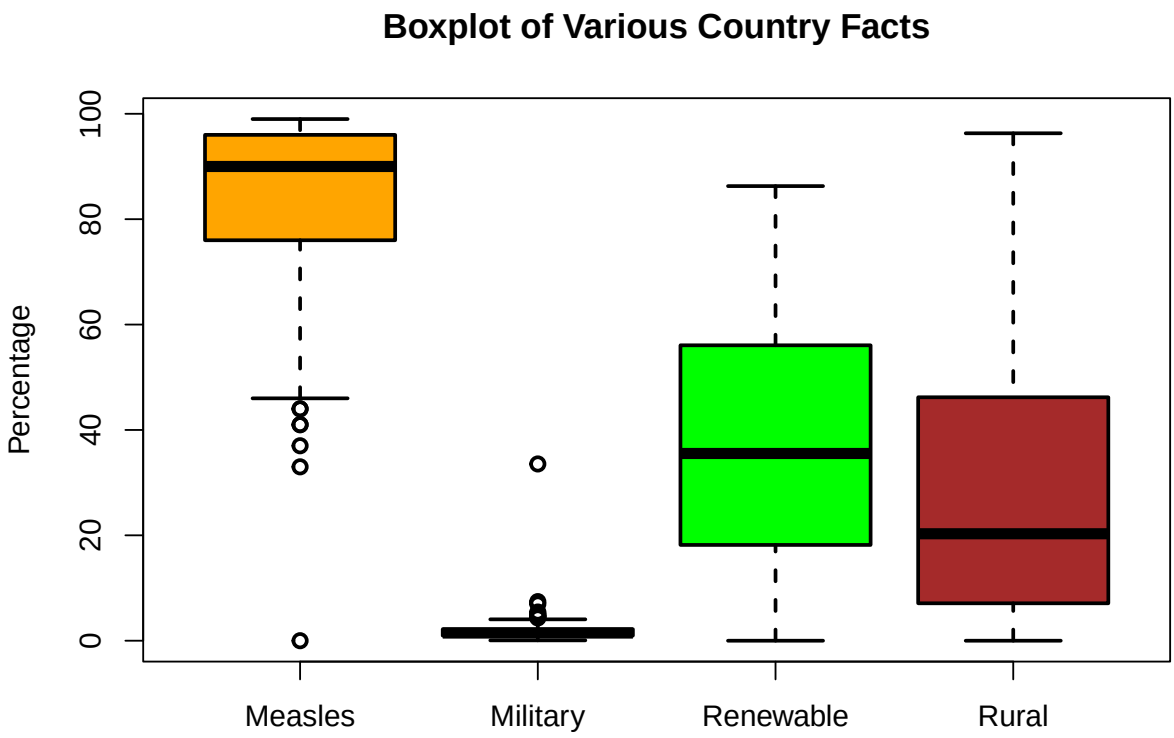
(4.1) Using the WB dataset loaded above, make a scatterplot of "Rural" on the x axis and "Measles" on the y axis. Include a main title, axis titles, and a non-default symbol color and symbol type. *Hint: check out ?par or see examples from class 1 or class 3.*

(4.2) Make a boxplot of the the following `wb` variables that are all percentages: `Measles`, `Military`, `Renewable`, `Rural`. You want all four variables on one plot. Ensure the plot has a main title, axis labels, and a unique color for each variable.

```
plot(wb$Rural, wb$Measles,
     xlab="Rural",
     ylab="Measles",
     main="Scatterplot of Rural vs Measles",
     col="Blue", pch='x')
```




```
boxplot(wb$Measles, wb$Military, wb$Rural, wb$Renewable,
        names=c("Measles", "Military", "Renewable", "Rural"),
        main="Boxplot of Various Country Facts",
        ylab="Percentage",
        col=c("Orange", "Red", "Green", "Brown"),
        lwd=2)
```



(5) Lists *12 pts* The code below creates a list called `aList`

(5.1) Compute the sum of the second element of the list's third element. Store the result into an object named `mySum`. You'll want to use the `sum()` function.

```
aList <- list(c(1, 5, 4), letters[c(1, 6, 4, 9, 22, 3)], list(c(1, 1, 1),
                        c(14, 13, 12), c(3, 2, 1)), c(runif(8)))
mySum <- sum(aList[[3]][[2]])
mySum
```

```
## [1] 39
```

(5.2) What is the difference between what is returned from the following two commands?

```
aList[[3]][2]
```

```
## [[1]]
## [1] 14 13 12
```

```
aList[[3]][[2]]
```

```
## [1] 14 13 12
```

Both commands have the same first part, aList[[3]], which extracts the third element in aList, which is list(c(1, 1, 1), c(14, 13, 12), c(3, 2, 1)). The first command, which has [2], returns the second element of this list contained in a 1-element list, or list(c(14, 13, 12)). The second command, which has [[2]], just returns the second element of the list, which is c(14, 13, 12).