Week 1 - Quiz 1

Raphael Carvalho

07/07/2019

1.The American Community Survey distributes downloadable data about United States communities. Download the 2006 microdata survey about housing for the state of Idaho using download.file() from here: https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06hid.csv and load the data into R. The code book, describing the variable names is here: https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FPUMSDataDict06.pdf How many properties are worth $1,000,000 or more?

```
fileURL <- "https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06hid.csv"
download.file(fileURL, destfile = "./quiz1_q1.csv", method="curl")

dt <- read.csv("quiz1_q1.csv")

dt %>% filter(VAL == 24) %>% summarize(n())
```

```
##   n()
## 1  53
```

[ ] 47

[x] 53

[ ] 164

[ ] 31

2. Use the data you loaded from Question 1. Consider the variables FES in the code book. Which of the "tidy data" principles does this variable violate?

[x] Tidy data has one variable per column.

[ ] Each tidy data table contains information about only one type of observation.

[ ] Each variable in a tidy data set has been transformed to be interpretable.

[ ] Tidy data has one observation per row.

3. Download the Excel spreadsheet on Natural Gas Aquisition Program here: https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FDATA.gov_NGAP.xlsx Read rows 18-23 and columns 7-15 into R and assign the result to a variable called:

```
fileURL <- "https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FDATA.gov_NGAP.xlsx"
download.file(fileURL, destfile = "./quiz1_q3.xlsx", mode = "wb")
dat <- xlsx::read.xlsx("quiz1_q3.xlsx", sheetIndex = 1, rowIndex = 18:23, colIndex = 7:15)
sum(dat$Zip*dat$Ext,na.rm=T)
```

```
## [1] 36534720
```

[ ] 33544718

[x] 36534720

[ ] 154339

[ ] NA

4. Read the XML data on Baltimore restaurants from here: https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Frestaurants.xml How many restaurants have zipcode 21231?

```
fileURL <- "https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Frestaurants.xml"
doc <- XML::xmlTreeParse(sub("s", "", fileURL), useInternal = TRUE)
rootNode <- XML::xmlRoot(doc)
zipcodes <- XML::xpathSApply(rootNode, "//zipcode", XML::xmlValue)
zipcodes <- data.table::data.table(zipcode = zipcodes)
summarize(filter(zipcodes, zipcode == "21231"), n())
```

```
##   n()
## 1 127
```

[x] 127

[ ] 100

[ ] 17

[ ] 181

5. The American Community Survey distributes downloadable data about United States communities. Download the 2006 microdata survey about housing for the state of Idaho using download.file from here: https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06pid.csv using the fread() command load the data into an R object "DT". The following are ways to calculate the average of the value pwgtp15 broken down by sex. Using the data.table package, which will theliver the fastest user time?

```
DT <- data.table::fread("https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06pid.csv")

system.time(sapply(split(DT$pwgtp15,DT$SEX),mean))
```

```
##    user  system elapsed
##   0.003   0.000   0.001
```

```
system.time(DT[,mean(pwgtp15),by=SEX])
```

```
##    user  system elapsed
##   0.027   0.001   0.008
```

```
system.time(rowMeans(DT)[DT$SEX==1])
```

```
## Error in rowMeans(DT): 'x' deve ser numérico
```

```
## Timing stopped at: 1.059 0.053 0.658
```

```
system.time(rowMeans(DT)[DT$SEX==2])
```

```
## Error in rowMeans(DT): 'x' deve ser numérico
```

```
## Timing stopped at: 0.458 0.044 0.505
```

```
system.time(mean(DT$pwgtp15,by=DT$SEX))
```

```
##    user  system elapsed
##   0.000   0.000   0.001
```

```
system.time(tapply(DT$pwgtp15,DT$SEX,mean))
```

```
##    user  system elapsed
##   0.001   0.001   0.001
```

```
system.time(mean(DT[DT$SEX==1,]$pwgtp15))
```

```
##    user  system elapsed
##   0.008   0.000   0.002
```

```
system.time(mean(DT[DT$SEX==2,]$pwgtp15))
```

```
##    user  system elapsed
##   0.009   0.001   0.002
```

[ ] sapply(split(DT$pwgtp15, DT$SEX),mean)

[x] DT[,mean(pwgtp15),by=SEX]

[ ] rowMeans(DT)[DT$SEX==1]; rowMeans(DT)[DT$SEX==2]

[ ] mean(DT$pwgtp15, by = DT$SEX)

[ ] tapply(DT$pwgtp15, DT$SEX,mean)

[ ] mean(DT[DT$SEX==1,]$pwgtp15); mean(DT[DT$SEX==2,]$pwgtp15)