HW05 - DS6306-402

Dan Crouthamel

June 07, 2019

Backstory: Your client is expecting a baby soon. However, he is not sure what to name the child. Being out of the loop, he hires you to help him figure out popular names. He provides for you raw data in order to help you make a decision.

1 - Data Munging (30 points):

Utilize yob2016.txt for this question. This file is a series of popular children's names born in the year 2016 in the United States. It consists of three columns with a first name, a gender, and the amount of children given that name. However, the data is raw and will need cleaning to make it tidy and usable.

a) First, import the .txt file into R so you can process it. Keep in mind this is not a CSV file. You might have to open the file to see what you're dealing with. Assign the resulting data frame to an object, df, that consists of three columns with humanreadable column names for each.

```r
# File is delimited by ';'
# Glad to see Danile in top 15
df <- read.table("yob2016.txt", header=FALSE, sep = ";")
names(df) <- c("Name", "Sex", "Count")
```

b) Display the summary and structure of df

```r
summary(df)
```

```
##       Name          Sex            Count        
##  Aalijah:    2   F:18758   Min.   :    5.0  
##  Aaliyan:    2   M:14111   1st Qu.:    7.0  
##  Aamari :    2             Median :   12.0  
##  Aarian :    2             Mean   :  110.7  
##  Aarin  :    2             3rd Qu.:   30.0  
##  Aaris  :    2             Max.   :19414.0  
##  (Other):32857
```

```r
str(df)
```

```
## 'data.frame':    32869 obs. of  3 variables:
##  $ Name : Factor w/ 30295 levels "Aaban","Aabha",..: 9317 22546 3770 26409 12019 20596 6185 339 9298
##  $ Sex  : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Count: int  19414 19246 16237 16070 14722 14366 13030 11699 10926 10733 ...
```

c) Your client tells you that there is a problem with the raw file. One name was entered twice and misspelled. The client cannot remember which name it is; there are thousands he saw! But he did mention he accidentally put three y's at the end of the name. Write an R command to figure out which name it is and display it.

```r
# Find rows that match
indicies <- grep("yyy$", df$Name, ignore.case = TRUE)
# Turns out there is only one, so display the row
df[indicies,]
```

```
##         Name Sex Count
## 212 Fionayyy   F  1547
```

d) Upon finding the misspelled name, please remove this particular observation, as the client says it's redundant. Save the remaining dataset as an object: y2016

```
y2016 <- df[-c(indicies),]
```

2 - Data Merging (30 points):

Utilize yob2015.txt for this question. This file is similar to yob2016, but contains names, gender, and total children given that name for the year 2015.

a) Like 1a, please import the .txt file into R. Look at the file before you do. You might have to change some options to import it properly. Again, please give the dataframe human-readable column names. Assign the dataframe to y2015.

```
# Note, I could have used read.csv since it's comma separated
y2015 <- read.table("yob2015.txt", header=FALSE, sep = ",")
names(y2015) <- c("Name", "Sex", "Count")
```

b) Display the last ten rows in the dataframe. Describe something you find interesting about these 10 rows.

```
tail(y2015,10)
```

```
##          Name Sex Count
## 33054   Ziyu   M     5
## 33055   Zoel   M     5
## 33056  Zohar   M     5
## 33057 Zolton   M     5
## 33058   Zyah   M     5
## 33059 Zykell   M     5
## 33060 Zyking   M     5
## 33061  Zykir   M     5
## 33062  Zyrus   M     5
## 33063   Zyus   M     5
```

It's all males. When looking at the file, you'll see it first list females, then males. Also, there is no data for values less than 5. It's the same for females.

```
tail(y2015[y2015$Sex=="F",],10)
```

```
##            Name Sex Count
## 19045     Zulia   F     5
## 19046   Zuliana   F     5
## 19047     Zulie   F     5
## 19048   Zuriana   F     5
## 19049 Zurianna   F     5
## 19050    Zuriya   F     5
## 19051    Zyleah   F     5
## 19052    Zyllah   F     5
## 19053   Zynique   F     5
## 19054 Zyrielle   F     5
```

c) Merge y2016 and y2015 by your Name column; assign it to final. The client only cares about names that have data for both 2016 and 2015; there should be no NA values in either of your amount of children rows after merging.

```
# Note, I'm merging by both Name and Sex. It makes more sense to me ...
final <- merge(y2016,y2015,by=c("Name","Sex"))
# I'm going to rename the counts columns as well. Easier to read.
names(final)[3:4]=c("2016-Total", "2015-Total")
```

3 - Data Summary (30 points):

Utilize your data frame object final for this part.

a) Create a new column called "Total" in final that adds the amount of children in 2015 and 2016 together. In those two years combined, how many people were given popular names?

```
final$Total <- final$`2015-Total` + final$`2016-Total`
totalCount <- sum(final$Total)
```

There were 7239231 people given popular names.

b) Sort the data by Total. What are the top 10 most popular names?

```
final <- final[order(-final$Total),]
head(final,10)
```

```
##           Name Sex 2016-Total 2015-Total Total
## 8290      Emma   F      19414      20415 39829
## 19886   Olivia   F      19246      19638 38884
## 19594     Noah   M      19015      19594 38609
## 16114     Liam   M      18138      18330 36468
## 23273   Sophia   F      16070      17381 33451
## 3252       Ava   F      16237      16340 32577
## 17715    Mason   M      15192      16591 31783
## 25241  William   M      15668      15863 31531
## 10993    Jacob   M      14416      15914 30330
## 10682 Isabella   F      14722      15574 30296
```

c) The client is expecting a girl! Omit boys and give the top 10 most popular girl's names.

```
girlData <- subset(final,final$Sex=="F")
head(girlData,10)
```

```
##            Name Sex 2016-Total 2015-Total Total
## 8290       Emma   F      19414      20415 39829
## 19886    Olivia   F      19246      19638 38884
## 23273    Sophia   F      16070      17381 33451
## 3252        Ava   F      16237      16340 32577
## 10682  Isabella   F      14722      15574 30296
## 18247       Mia   F      14366      14871 29237
## 5493  Charlotte   F      13030      11381 24411
## 277     Abigail   F      11699      12371 24070
## 8273      Emily   F      10926      11766 22692
## 9980     Harper   F      10733      10283 21016
```

d) Write these top 10 girl names and their Totals to a CSV file. Leave out the other columns entirely.

```
write.csv(girlData[1:10,c("Name", "Total")], file = "Top10GirlNames.csv", row.names = FALSE)
```

4 - Upload to GitHub (10 points):

Push at minimum your RMarkdown for this homework assignment and a Codebook to one of your GitHub repositories (you might place this in a Homework repo like last week). The Codebook should contain a short definition of each object you create, and if creating multiple files, which file it is contained in. You are welcome and encouraged to add other files—just make sure you have a description and directions that are helpful for the grader.

Please see Unit 5 directory in the following repo. https://github.com/bSharpCyclist/MSDS-6306---Intro-To-Data-Science