Stat 123 HW 13

Jonathan Wilson

February 10, 2019

```
knitr::opts_knit$set(root.dir =
"C:\\Users\\jon\\Documents\\School\\R\\HW\\HW13")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(plyr)
```

```
## -------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first,
then dplyr:
## library(plyr); library(dplyr)
```

```
## -------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(readr)
library(ggplot2)
library(ggrepel)
require(scales)
```

```
## Loading required package: scales
```

```
##
## Attaching package: 'scales'
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
require(stringr)
```

```
## Loading required package: stringr
```

Complete the definition of the function. It takes one argument “year” (where “year” can range from “1910” to “2013”) and returns a data.frame giving the count by gender of names for babies born in the specified year. The variable names should be “name”, “sex”, and “count”. If the value of the “year” argument is “all”, a data.frame containing all the data should be returned. Use the data provided in the attached folder “names-full-datasets”
Hint: You will probably want to use the ‘paste’ function.

```
#All consolidates (Combines Names) all the files and combines their counts.
baby.names <- function(year="all") {
  if(year %in% 1910:2013){
    #Return a data.frame with vars name sex and count
    year<-as.character(year)
    path<-"yob"
    #yob<-"yob"
    txt<-c(".txt")
    file<-paste(path, year, txt, sep = "")
    babyNames<-NULL
    babyNames<-data.frame()
    babyNames <- read.table(file, sep=",", header=FALSE,
stringsAsFactors=FALSE)
    names(babyNames)<-c("Name", "Sex", "Count")
    return(babyNames)
  }
  else if(year=="all"){
    babyNames<-NULL
    babyNames<-data.frame()
    for(i in 1910:2013){
      year<-i
      path<-"yob"
      #yob<-"\\yob"
      txt<-".txt"
      file<-paste(path, year, txt, sep = "")
      tmp <- read.table(file, sep=",", header=FALSE, stringsAsFactors=FALSE)
      babyNames<-rbind(babyNames, tmp)
    }

    names(babyNames)<-c("Name", "Sex", "Count")
    babyNames<-ddply(babyNames,.(Name,Sex),numcolwise(sum)) #This took me so
long to figure out.
    #https://stackoverflow.com/questions/7449198/quick-elegant-way-to-
construct-mean-variance-summary-table
    return(babyNames)
  }
  else{
    return("Error: Date out of range.")
  }
}
```

Execute the code below and use the resulting data.frame for the rest of this problem.

```
yall <- baby.names()
```

Complete the function definition below that counts the number of occurences of the supplied “name” for a vector of “years”.

```
#This function calculates the total counts of the given name and years. I did
it this way becuase I assumed that we needed to use
#this function for the next problem which asks for this.
lookup <- function(name,years){
  n<-0
  for(i in years){
    df<-NULL
    df <- baby.names(i)
    ndf<-select(df, Count) %>% filter(df$Name==name)
    if(!is.null(ndf[1,]) & !is.na(ndf[1,])){
      n<-n+ndf[1,]
    }
  }
  return(n)
}

#This function calculates "single"" instances of the name
"  lookup <- function(name,years){
  n<-0
  for(i in years){
    df <- baby.names(i)
    n<-select(df, Count) %>% filter(df$Name==name)
    if(!is.null(n)){
      n<-n+n
    }
  }
  return(count)
  }
"

## [1] "  lookup <- function(name,years){\n  n<-0\n  for(i in years){\n    df
<- baby.names(i)\n    n<-select(df, Count) %>% filter(df$Name==name)\n
if(!is.null(n)){\n      n<-n+n\n    }\n  }\n  return(count)\n  }  \n"
```

Use your function to find how many babies where born with the name “Jennifer” in each of the following years: 1913, 1923, ..., 2003, 2013.

```
lookup("Jennifer", 1913:2013)#1461186
```

```
## [1] 1461186
```

```
#lookup("Mary", 1910:1910)
```

What is the overall most popular name for a boy? How about for a girl?

```
  #yall2<- group_by(yall, Name) %>% summarise(Count = sum(Count))

  boy <- group_by(yall, Sex) %>% filter(Count == max(Count) & Sex == "M")
  boy<-boy$Name
  boy
```

```
## [1] "James"
```

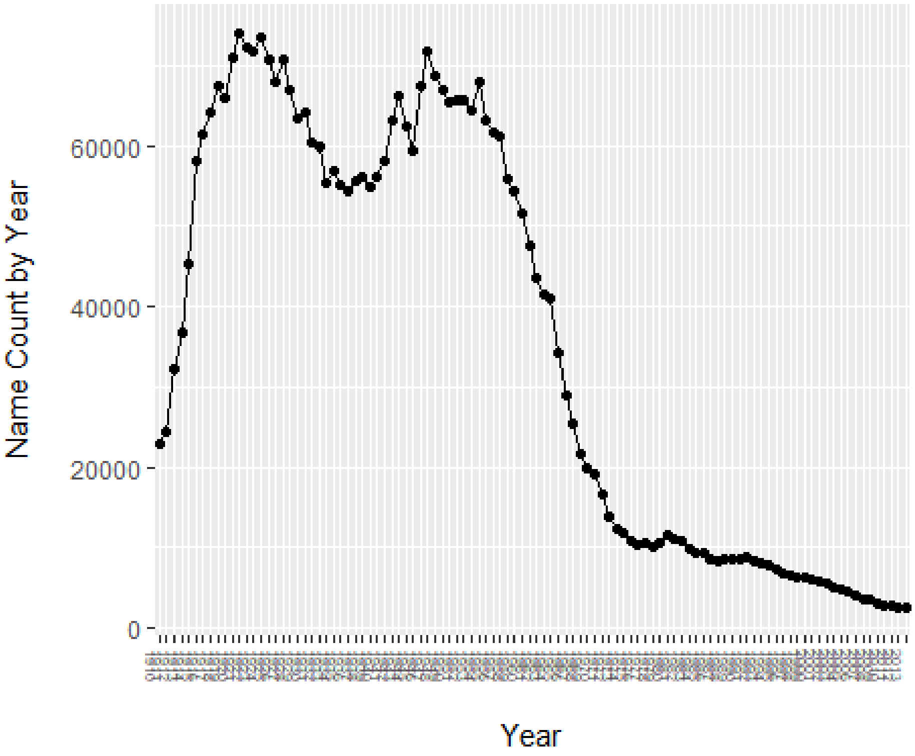
```
  girl <- group_by(yall, Sex) %>% filter(Count == max(Count) & Sex == "F")
  girl<-girl$Name
  girl
```

```
## [1] "Mary"
```

Produce a line plot showing the proportion of babies named “Mary” among all female names over time.

```
babyfiles <- function(babyfile) {
  year <- str_extract(paste0(babyfile), "\\d{4}")
  df <- read_delim(file = babyfile,
             delim = ",",
             col_names = FALSE,
             col_types = cols(
               X1 = col_character(),
               X2 = col_character(),
               X3 = col_integer()
             ))
  df$year <- year
  names(df) <- c("name", "sex", "count", "year")
  df
}

location <- "C:\\Users\\jon\\Documents\\School\\R\\HW\\HW13\\names-full-
datasets"
setwd(location)

temp = list.files(path = location, pattern="*.txt")
temp <- tibble(files = temp)

test <- temp %>%
  group_by(files) %>%
  do(babyfiles(.$files))

test %>%
  filter(name %in% c("Mary") & sex == "F") %>%
```

```r
ggplot(aes(year, count, group = name)) +
geom_line() +
geom_point() +
xlab("\nYear") +
ylab("\nName Count by Year\n") +
guides(color = FALSE) +
theme_grey() +
theme(axis.text.x = element_text(angle = -90, size = 5))
```


Name Count by Year
Year