Homeworkassignment1

Taha Abbasi-Hashemi

Q1

```
library(tidyverse)
library(babynames)
library(datadictionary)
```

Q2

5 variables 1924665 observations

```
names <- babynames
names_og <- babynames
babynames
```

```
## # A tibble: 1,924,665 x 5
##     year sex   name          n   prop
##    <dbl> <chr> <chr>     <int>  <dbl>
##  1  1880 F     Mary       7065 0.0724
##  2  1880 F     Anna       2604 0.0267
##  3  1880 F     Emma       2003 0.0205
##  4  1880 F     Elizabeth  1939 0.0199
##  5  1880 F     Minnie     1746 0.0179
##  6  1880 F     Margaret   1578 0.0162
##  7  1880 F     Ida        1472 0.0151
##  8  1880 F     Alice      1414 0.0145
##  9  1880 F     Bertha     1320 0.0135
## 10  1880 F     Sarah      1288 0.0132
## # i 1,924,655 more rows
```

Q3

```
names.labels <- c(year="Year of Birth",
                  sex="sex of name",
                  name="name of child",
                  n="number of children born with that name in that year",
                  prop="proportion of the total births with this name")
names_dict <- create_dictionary(names, var_labels = names.labels)
```

```
## Warning in character_summary(dataset, column): sex has fewer than 10 unique values, did you want a
## factor?
```

Q4

range

```
min(names["year"])
```

```
## [1] 1880
```

```
max(names["year"])
```

```
## [1] 2017
```

Q5

```
names <- names[, -which(names(names) == "n")]
```

Q6

Allows for potentially normalized viewage of names. Because what will happen is as time moves on, there will be different number of births. n=1000 might mean alot in the past, but not so much now.

Q7

Using the object created in Question 5, what was the most popular name for both sexes in: a) the 2nd millennium? and b) the 3rd millennium?

John then Jacob

```
names_2nd <- names[names$year >= 1900 & names$year < 2000, ]
names_3rd <- names[names$year > 2000, ]

max_prop_2nd <- names_2nd[which.max(names_2nd$prop), ]
max_prop_3rd <- names_3rd[which.max(names_3rd$prop), ]

# View the result
print(max_prop_2nd)
```

```
## # A tibble: 1 x 4
##    year sex   name    prop
##   <dbl> <chr> <chr>  <dbl>
## 1  1900 M     John  0.0606
```

```r
print(max_prop_3rd)
```

```
## # A tibble: 1 x 4
##    year sex   name    prop
##   <dbl> <chr> <chr>  <dbl>
## 1  2001 M     Jacob 0.0157
```

Q8

I can use grep interestingly enough I originally wanted to try using regex. quinn, victoria, xavier

```r
names_q <- names_3rd[names_3rd$year >= 2000 & names_3rd$year <= 2012 & grepl("^[Q]",
                          names_3rd$name, ignore.case = TRUE), ]

names_v <- names_3rd[names_3rd$year >= 2000 & names_3rd$year <= 2012 & grepl("^[V]",
                          names_3rd$name, ignore.case = TRUE), ]

names_x <- names_3rd[names_3rd$year >= 2000 & names_3rd$year <= 2012 & grepl("^[X]",
                          names_3rd$name, ignore.case = TRUE), ]
max_prop_3rd <- names_q[which.max(names_q$prop), ]
print(max_prop_3rd)
```

```
## # A tibble: 1 x 4
##    year sex   name     prop
##   <dbl> <chr> <chr>   <dbl>
## 1  2012 F     Quinn 0.00109
```

```r
max_prop_3rd <- names_v[which.max(names_v$prop), ]
print(max_prop_3rd)
```

```
## # A tibble: 1 x 4
##    year sex   name        prop
##   <dbl> <chr> <chr>      <dbl>
## 1  2001 F     Victoria 0.00514
```

```r
max_prop_3rd <- names_x[which.max(names_x$prop), ]
print(max_prop_3rd)
```

```
## # A tibble: 1 x 4
##    year sex   name      prop
##   <dbl> <chr> <chr>    <dbl>
## 1  2007 M     Xavier 0.00296
```

Q9

I dont know if I'm doing this right. So I wanted to merge all names together within the decade. SO if there was a mary in 1800 and 1801, the N would be added together. I didnt know what do with prop so I just took the mean.

```r
names_og$decade <- floor(names_og$year / 10) * 10
names_by_decade <- aggregate(cbind(n, prop) ~ name + sex + decade, data = names_og,
                             FUN = function(x) c(sum = sum(x), avg = mean(x)))
```

Q10

This should get the mean and median value for n across sex and decade.

```r
mean_median_by_decade_sex <- aggregate(n ~ sex + decade, data = names_og,
                                       FUN = function(x) c(mean = mean(x), median = median(x)))

mean_median_by_decade_sex <- do.call(data.frame, mean_median_by_decade_sex)
names(mean_median_by_decade_sex)[3:4] <- c("mean_n", "median_n")
print(mean_median_by_decade_sex)
```

```
##    sex decade    mean_n median_n
## 1    F   1880 110.57017       13
## 2    M   1880 100.76497       12
## 3    F   1890 128.18406       13
## 4    M   1890  93.59019       12
## 5    F   1900 131.32904       12
## 6    M   1900  94.38963       12
## 7    F   1910 187.06284       12
## 8    M   1910 180.83854       12
## 9    F   1920 210.54574       12
## 10   M   1920 226.78161       13
## 11   F   1930 214.19867       12
## 12   M   1930 253.28957       13
## 13   F   1940 262.20824       12
## 14   M   1940 368.40859       14
## 15   F   1950 288.47692       13
## 16   M   1950 460.86555       14
## 17   F   1960 234.71960       12
## 18   M   1960 415.51792       13
## 19   F   1970 147.20851       11
## 20   M   1970 265.55153       12
## 21   F   1980 134.25355       11
## 22   M   1980 236.98189       11
## 23   F   1990 113.07160       11
## 24   M   1990 187.35187       11
## 25   F   2000  96.45799       11
## 26   M   2000 149.06677       11
## 27   F   2010  91.69925       11
## 28   M   2010 133.67495       11
```

Q11

I wrote a function do this. I doubt Taha and Baraa have been the most popular name of a baby in the USA from 1800-2017... If they are never the most popular it returns "never","never"

```r
find_most_popular <- function(name_input) {

  # MAke decade
  names_og$decade <- floor(names_og$year / 10) * 10
    most_popular_each_year <- names_og %>%
    group_by(year) %>%
    filter(n == max(n)) %>%
    ungroup()


  most_popular_for_name <- most_popular_each_year[most_popular_each_year$name == name_input, ]

  # I doubt Taha and Baraa have been the most popular name of a baby in the USA from 1800-2017...
  if (nrow(most_popular_for_name) == 0) {
    return(c("never", "never"))
  }
  return(most_popular_for_name[, c("decade", "year")])
}

# Results
find_most_popular("Taha")
```

```
## [1] "never" "never"
```

```r
find_most_popular("Baraa")
```

```
## [1] "never" "never"
```

```r
find_most_popular("Mike")
```

```
## [1] "never" "never"
```

```r
find_most_popular("Jack")
```

```
## [1] "never" "never"
```

```r
find_most_popular("Scott")
```

```
## [1] "never" "never"
```

```r
# Testing....
find_most_popular("Mary")
```

```
## # A tibble: 49 x 2
##    decade  year
##     <dbl> <dbl>
##  1   1880  1885
##  2   1880  1886
##  3   1880  1887
```

```
##  4   1880  1888
##  5   1880  1889
##  6   1890  1890
##  7   1890  1891
##  8   1890  1892
##  9   1890  1893
## 10   1890  1894
## # i 39 more rows
```

```r
find_most_popular("John")
```

```
## # A tibble: 5 x 2
##   decade  year
##    <dbl> <dbl>
## 1   1880  1880
## 2   1880  1881
## 3   1880  1882
## 4   1880  1883
## 5   1880  1884
```