Homework assignment 1

Matin Borhani

```
#Libraries I used

library(babynames)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```
#install.packages("quanteda")
library(quanteda)
```

```
Package version: 4.1.0
Unicode version: 14.0
ICU version: 71.1
Parallel computing: disabled
See https://quanteda.io for tutorials and examples.
```

```
library(stringr)
```

```
#Q2.

glimpse(babynames)
```

```
Rows: 1,924,665
Columns: 5
$ year <dbl> 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880,~
$ sex  <chr> "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", ~
$ name <chr> "Mary", "Anna", "Emma", "Elizabeth", "Minnie", "Margaret", "Ida",~
$ n    <int> 7065, 2604, 2003, 1939, 1746, 1578, 1472, 1414, 1320, 1288, 1258,~
$ prop <dbl> 0.07238359, 0.02667896, 0.02052149, 0.01986579, 0.01788843, 0.016~
```

```
#There are 5 variables (year, sex, name, n,  prop) and 1,924,665 observations
```

```
#Q3.

babynames <- babynames
babynames_dict <- list(
  year = list(
    data = babynames$year,
    type = "double",
    description = "Birth year of baby"
  ),
  sex = list(
    data = babynames$sex,
    type = "character",
    description = "Sex of baby"
  ),
  name = list(
    data = babynames$name,
    type = "character",
    description = "Name of baby"
  ),
  n = list(
    data = babynames$n,
    type = "integer",
    description = "Number of babies born with this name"
  ),
  prop = list(
    data = babynames$prop,
    type = "double",
    description = "proportion of babies born with this name"
  )
)

str(babynames_dict)
```

```
List of 5
 $ year:List of 3
  ..$ data       : num [1:1924665] 1880 1880 1880 1880 1880 1880 1880 1880 1880 1880 ...
  ..$ type       : chr "double"
  ..$ description: chr "Birth year of baby"
 $ sex :List of 3
  ..$ data       : chr [1:1924665] "F" "F" "F" "F" ...
  ..$ type       : chr "character"
  ..$ description: chr "Sex of baby"
 $ name:List of 3
  ..$ data       : chr [1:1924665] "Mary" "Anna" "Emma" "Elizabeth" ...
  ..$ type       : chr "character"
  ..$ description: chr "Name of baby"
 $ n   :List of 3
  ..$ data       : int [1:1924665] 7065 2604 2003 1939 1746 1578 1472 1414 1320 1288 ...
  ..$ type       : chr "integer"
  ..$ description: chr "Number of babies born with this name"
 $ prop:List of 3
  ..$ data       : num [1:1924665] 0.0724 0.0267 0.0205 0.0199 0.0179 ...
  ..$ type       : chr "double"
  ..$ description: chr "proportion of babies born with this name"
```

```r
#Q4

range(babynames$year, na.rm = TRUE)
```

```
[1] 1880 2017
```

```r
#Range of years covered in babynames is from 1880-2017```
```

```r
#Q5.
babynames_no_n <- dplyr::select(babynames, year, sex, name, prop)
babynames_no_n
```

```
# A tibble: 1,924,665 x 4
    year sex   name        prop
   <dbl> <chr> <chr>      <dbl>
 1  1880 F     Mary      0.0724
 2  1880 F     Anna      0.0267
 3  1880 F     Emma      0.0205
 4  1880 F     Elizabeth 0.0199
```

```
 5  1880 F     Minnie    0.0179
 6  1880 F     Margaret  0.0162
 7  1880 F     Ida       0.0151
 8  1880 F     Alice     0.0145
 9  1880 F     Bertha    0.0135
10  1880 F     Sarah     0.0132
# i 1,924,655 more rows
```

```
#Q6.
#Because some values of "n" can be confused with "year". For example the value '2003' has bee
```

```
#Q7.
#Assuming 2 millennial is until Dec 31st, 2000 and 3 millennial begins from jan 1st, 2001...

babynames_no_n <- dplyr::select(babynames, year, sex, name, prop)
babynames_no_n
```

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

```
baby_female_two_mil <-dplyr::filter(babynames_no_n, year < 2001, sex == "F")
baby_female_two_mil
```

```
# A tibble: 811,077 x 4
    year sex   name        prop
   <dbl> <chr> <chr>      <dbl>
 1  1880 F     Mary      0.0724
 2  1880 F     Anna      0.0267
 3  1880 F     Emma      0.0205
 4  1880 F     Elizabeth 0.0199
 5  1880 F     Minnie    0.0179
 6  1880 F     Margaret  0.0162
 7  1880 F     Ida       0.0151
 8  1880 F     Alice     0.0145
 9  1880 F     Bertha    0.0135
10  1880 F     Sarah     0.0132
# i 811,067 more rows
```

```
baby_male_two_mil <-dplyr::filter(babynames_no_n, year < 2001, sex == "M")
baby_male_two_mil
```

```
# A tibble: 551,432 x 4
    year sex   name      prop
   <dbl> <chr> <chr>    <dbl>
 1  1880 M     John    0.0815
 2  1880 M     William 0.0805
 3  1880 M     James   0.0501
 4  1880 M     Charles 0.0452
 5  1880 M     George  0.0433
 6  1880 M     Frank   0.0274
 7  1880 M     Joseph  0.0222
 8  1880 M     Thomas  0.0214
 9  1880 M     Henry   0.0206
10  1880 M     Robert  0.0204
# i 551,422 more rows
```

```
baby_female_three_mil <- dplyr::filter(babynames_no_n, year >= 2001, sex == "F")
baby_female_three_mil
```

```
# A tibble: 327,216 x 4
    year sex   name         prop
   <dbl> <chr> <chr>       <dbl>
 1  2001 F     Emily     0.0127
 2  2001 F     Madison   0.0112
 3  2001 F     Hannah    0.0105
 4  2001 F     Ashley    0.00835
 5  2001 F     Alexis    0.00828
 6  2001 F     Sarah     0.00803
 7  2001 F     Samantha  0.00801
 8  2001 F     Abigail   0.00748
 9  2001 F     Elizabeth 0.00747
10  2001 F     Olivia    0.00706
# i 327,206 more rows
```

```
baby_male_three_mil <-dplyr::filter(babynames_no_n, year >= 2001, sex == "M")
baby_male_three_mil
```

```
# A tibble: 234,940 x 4
    year sex   name           prop
   <dbl> <chr> <chr>         <dbl>
 1  2001 M     Jacob       0.0157
 2  2001 M     Michael     0.0144
 3  2001 M     Matthew     0.0130
 4  2001 M     Joshua      0.0126
 5  2001 M     Christopher 0.0112
 6  2001 M     Nicholas    0.0110
 7  2001 M     Andrew      0.0108
 8  2001 M     Joseph      0.0106
 9  2001 M     Daniel      0.0101
10  2001 M     William     0.00972
# i 234,930 more rows
```

```
#Second millennial female = "Mary"
#Second millennial male = "John"
#Third millennial female = "Emily"
#Third millennial male = "Jacob"
```

```
#Q8.

baby_qvx_name <- babynames %>%
   filter(str_starts(name, "Q")| str_starts(name, "V")| str_starts(name, "X"))
   baby_qvx_name <- dplyr::filter(baby_qvx_name, year >= 2000 & year <= 2012)
   baby_qvx_name <- dplyr::arrange(baby_qvx_name, desc(n))

baby_qvx_name
```

```
# A tibble: 7,019 x 5
    year sex   name         n    prop
   <dbl> <chr> <chr>    <int>   <dbl>
 1  2000 F     Victoria 10923 0.00548
 2  2001 F     Victoria 10179 0.00514
 3  2002 F     Victoria  9782 0.00496
 4  2003 F     Victoria  9243 0.00461
 5  2004 F     Victoria  8274 0.00410
 6  2005 F     Victoria  7955 0.00392
 7  2006 F     Victoria  7647 0.00366
 8  2007 F     Victoria  7431 0.00351
 9  2008 F     Victoria  7118 0.00342
10  2011 F     Victoria  6888 0.00356
# i 7,009 more rows
```

```
#2000   F   Victoria    10923   0.00547551
#2007   M   Xavier  6556    0.00296193
#2012   F   Quinn   2108    0.00108871
```

```
#Q9. Note- I accidentally misspelled decade by "dacade" throughout my program
#decade_func <- function(year - year%%10)

dacade_func <- function(year) {
  return(year - year %% 10)
}
babyname_newcol <- dplyr::mutate(babynames, dacade = (year-year%%10))
babyname_newcol
```

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

```
#Q10.
 by_dacade <- dplyr::group_by(babyname_newcol, dacade, sex)
     dplyr::summarize(by_dacade,
                      mean_observation = mean(n, na.rm = TRUE),
                      sd_observation = sd(n, na.rm = TRUE),
                      n = n())
```

```
`summarise()` has grouped output by 'dacade'. You can override using the
`.groups` argument.

# A tibble: 28 x 5
# Groups:   dacade [14]
   dacade sex   mean_observation sd_observation     n
    <dbl> <chr>            <dbl>          <dbl> <int>
 1   1880 F                111.            405. 11872
 2   1880 M                101.            514. 10871
 3   1890 F                128.            508. 17331
 4   1890 M                 93.6           443. 12191
 5   1900 F                131.            573. 22292
 6   1900 M                 94.4           441. 14383
 7   1910 F                187.           1285. 43602
 8   1910 M                181.           1406. 36913
 9   1920 F                211.           1557. 56769
10   1920 M                227.           1945. 48591
# i 18 more rows
```

```
#Q11.

babyname_matin <- dplyr::filter(babyname_newcol, name == "Matin")
dplyr::arrange(babyname_matin, desc(n))
```

```
# A tibble: 26 x 6
    year sex   name      n       prop dacade
   <dbl> <chr> <chr> <int>      <dbl>  <dbl>
 1  2014 M     Matin    13 0.00000636   2010
 2  2017 M     Matin    13 0.00000662   2010
 3  2005 M     Matin    11 0.00000517   2000
 4  2001 M     Matin    10 0.00000484   2000
 5  2007 M     Matin    10 0.00000452   2000
 6  1994 M     Matin     9 0.00000442   1990
 7  2000 M     Matin     9 0.00000431   2000
 8  2004 M     Matin     9 0.00000426   2000
 9  2008 M     Matin     9 0.00000413   2000
10  2011 M     Matin     9 0.00000444   2010
# i 16 more rows
```

```
by_dacade_matin <- dplyr::group_by(babyname_matin, dacade)
dplyr::summarize(by_dacade_matin,
     mean_observation = mean(n, na.rm = TRUE),
     sd_observation = sd(n, na.rm = TRUE),
     n = n())
```

```
# A tibble: 4 x 4
  dacade mean_observation sd_observation     n
   <dbl>            <dbl>          <dbl> <int>
1   1980             8             NA        1
2   1990             7.57           1.27     7
3   2000             8.4            1.90    10
4   2010             9.25           2.55     8
```

```
#by_dacade_matin


babyname_baraa <- dplyr::filter(babyname_newcol, name == "Baraa")
dplyr::arrange(babyname_baraa, desc(n))
```

```
# A tibble: 20 x 6
    year sex   name      n       prop dacade
   <dbl> <chr> <chr> <int>      <dbl>  <dbl>
 1  2017 M     Baraa    14 0.00000713   2010
 2  2012 M     Baraa    12 0.00000592   2010
 3  2004 M     Baraa    11 0.00000521   2000
 4  2013 M     Baraa    10 0.00000496   2010
 5  2014 M     Baraa    10 0.00000489   2010
 6  2001 M     Baraa     9 0.00000435   2000
 7  2016 M     Baraa     9 0.00000446   2010
 8  2003 M     Baraa     7 0.00000333   2000
 9  2008 M     Baraa     7 0.00000321   2000
10  2009 M     Baraa     7 0.0000033    2000
11  2015 M     Baraa     7 0.00000343   2010
12  2016 F     Baraa     7 0.00000363   2010
13  2007 M     Baraa     6 0.00000271   2000
14  2008 F     Baraa     6 0.00000288   2000
15  2011 F     Baraa     6 0.0000031    2010
16  2011 M     Baraa     6 0.00000296   2010
17  1997 F     Baraa     5 0.00000262   1990
18  1997 M     Baraa     5 0.0000025    1990
```

```
19  1998 M     Baraa     5 0.00000247   1990
20  2006 M     Baraa     5 0.00000228   2000
```

```
by_dacade_baraa <- dplyr::group_by(babyname_baraa, dacade)
dplyr::summarize(by_dacade_baraa,
     mean_observation = mean(n, na.rm = TRUE),
     sd_observation = sd(n, na.rm = TRUE),
     n = n())
```

```
# A tibble: 3 x 4
  dacade mean_observation sd_observation     n
   <dbl>            <dbl>          <dbl> <int>
1   1990             5              0        3
2   2000             7.25           1.91     8
3   2010             9              2.78     9
```

```
#by_dacade_baraa


babyname_jack <- dplyr::filter(babyname_newcol, name == "Jack")
dplyr::arrange(babyname_jack, desc(n))
```

```
# A tibble: 256 x 6
    year sex   name      n    prop dacade
   <dbl> <chr> <chr> <int>   <dbl>  <dbl>
 1  1927 M     Jack  12795 0.0110    1920
 2  1928 M     Jack  12494 0.0109    1920
 3  1930 M     Jack  12431 0.0110    1930
 4  1926 M     Jack  12201 0.0107    1920
 5  1929 M     Jack  12167 0.0110    1920
 6  1925 M     Jack  12010 0.0104    1920
 7  1924 M     Jack  11924 0.0102    1920
 8  1931 M     Jack  11477 0.0107    1930
 9  1923 M     Jack  11191 0.00988   1920
10  2005 M     Jack  10903 0.00513   2000
# i 246 more rows
```

```
by_dacade_jack <- dplyr::group_by(babyname_jack, dacade)
dplyr::summarize(by_dacade_jack,
     mean_observation = mean(n, na.rm = TRUE),
     sd_observation = sd(n, na.rm = TRUE),
     n = n())
```

```
# A tibble: 14 x 4
   dacade mean_observation sd_observation     n
    <dbl>            <dbl>          <dbl> <int>
 1   1880             244.           90.3    11
 2   1890             333.          231.     14
 3   1900             545.          479.     17
 4   1910            2492.         3062.     20
 5   1920            5837          5949.     20
 6   1930            4843.         5041.     20
 7   1940            3706.         3796.     20
 8   1950            3126.         3226.     20
 9   1960            1788.         1907.     20
10   1970             937.          976.     20
11   1980             802.          812.     20
12   1990            2085.         2383.     18
13   2000            4782.         4941.     20
14   2010            4208.         4341.     16
```

```
#by_dacade_jack


babyname_scott <- dplyr::filter(babyname_newcol, name == "Scott")
dplyr::arrange(babyname_scott, desc(n))
```

```
# A tibble: 196 x 6
    year sex   name      n   prop dacade
   <dbl> <chr> <chr> <int>  <dbl>  <dbl>
 1  1971 M     Scott 30918 0.0170   1970
 2  1962 M     Scott 30707 0.0146   1960
 3  1963 M     Scott 30415 0.0147   1960
 4  1969 M     Scott 28687 0.0157   1960
 5  1970 M     Scott 28591 0.0150   1970
 6  1964 M     Scott 28507 0.0141   1960
 7  1966 M     Scott 26033 0.0143   1960
 8  1968 M     Scott 26031 0.0147   1960
 9  1967 M     Scott 25543 0.0144   1960
10  1965 M     Scott 25441 0.0134   1960
# i 186 more rows
```

```
by_dacade_scott <- dplyr::group_by(babyname_scott, dacade)
dplyr::summarize(by_dacade_scott,
     mean_observation = mean(n, na.rm = TRUE),
     sd_observation = sd(n, na.rm = TRUE),
     n = n())
```

```
# A tibble: 14 x 4
   dacade mean_observation sd_observation     n
    <dbl>            <dbl>          <dbl> <int>
 1   1880             48.4           6.57    10
 2   1890             38             7.90    10
 3   1900             34.8           5.98    10
 4   1910            108.           64.1     12
 5   1920            174.           57.8     11
 6   1930            198.           46.3     10
 7   1940            813.          966.      15
 8   1950           5600.         6898.      20
 9   1960          13381.        13796.      20
10   1970           9778.        10790.      20
11   1980           5439.         5610.      20
12   1990           2586.         2827.      19
13   2000           1476.          692.      11
14   2010            722.           89.2      8
```

```
#by_dacade_scott

#A) Matin: year of 2017 and 2010 decade
#B) Baraa: Year of 2017 and 2010 decade
#C) Jack: Year of 1927 and 1920 decade
#D) Scott: Year of 1971 and 1960 decade
```