Homework assignment 1

Anil

Quarto

library(babynames)

```
library(babynames)
```

```
Warning: package 'babynames' was built under R version 4.2.3
```

```
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.2.3
```

```
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
```

```
# 2. How many variables and observations does this package contain?
dim(babynames)
```

```
[1] 1924665       5
```

```r
# 3. Create a data dictionary for each of the variables that includes the variable name,da
#type, and a description
dictionary <- data.frame(
  Variable = c("year", "sex", "name", "n", "prop"),
  DataType = c("double", "character", "character", "integer", "double"),
  Description = c(
    "The year of birth of the babies",
    "The sex of the babies (M for male, F for female)",
    "The name given to the babies",
    "The count of babies with the given name and sex in that year",
    "The proportion of babies given this name out of all babies born in that year and sex"
  )
)

print(dictionary)
```

```
  Variable  DataType
1     year    double
2      sex character
3     name character
4        n   integer
5     prop    double
                                                                        Description
1                                                  The year of birth of the babies
2                                The sex of the babies (M for male, F for female)
3                                                   The name given to the babies
4                       The count of babies with the given name and sex in that year
5 The proportion of babies given this name out of all babies born in that year and sex
```

```r
# 4. What is the range of years covered in babynames?

year_range <- range(babynames$year)
year_range
```

```
[1] 1880 2017
```

```r
# 5. Create an object from the babynames package that does not include the variable n.
babynames_no_n <- babynames %>%
  select(-n)
```

```r
head(babynames_no_n)
```

```
# A tibble: 6 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
```

```r
# 7. Using the object created in Question 5, what was the most popular name for both sexes

# a) Most popular name in the 2nd millennium (years <= 2000)
most_popular_2nd_millennium <- babynames_no_n %>%
  filter(year <= 2000) %>%
  group_by(sex, name) %>%
  summarize(total_prop = sum(prop)) %>%
  arrange(desc(total_prop)) %>%
  slice(1)
```

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

```r
# b) Most popular name in the 3rd millennium (years > 2000)
most_popular_3rd_millennium <- babynames_no_n %>%
  filter(year > 2000) %>%
  group_by(sex, name) %>%
  summarize(total_prop = sum(prop)) %>%
  arrange(desc(total_prop)) %>%
  slice(1)
```

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

```r
# Display results
most_popular_2nd_millennium
```

```
# A tibble: 2 x 3
# Groups:   sex [2]
  sex   name  total_prop
  <chr> <chr>      <dbl>
1 F     Mary        4.49
2 M     John        5.24
```

```r
most_popular_3rd_millennium
```

```
# A tibble: 2 x 3
# Groups:   sex [2]
  sex   name  total_prop
  <chr> <chr>      <dbl>
1 F     Emma       0.165
2 M     Jacob      0.182
```

```r
# 8. What were the most popular names beginning with the letters Q, V, and X between 2000

# Filter for years between 2000 and 2012
names_2000_2012 <- babynames_no_n %>%
  filter(year >= 2000 & year <= 2012)

# Filter names starting with Q, V, and X
popular_Q <- names_2000_2012 %>%
  filter(startsWith(name, "Q")) %>%
  group_by(sex, name) %>%
  summarize(total_prop = sum(prop)) %>%
  arrange(desc(total_prop)) %>%
  slice(1)
```

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

```r
popular_V <- names_2000_2012 %>%
  filter(startsWith(name, "V")) %>%
```

```
    group_by(sex, name) %>%
    summarize(total_prop = sum(prop)) %>%
    arrange(desc(total_prop)) %>%
    slice(1)
```

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

```
  popular_X <- names_2000_2012 %>%
    filter(startsWith(name, "X")) %>%
    group_by(sex, name) %>%
    summarize(total_prop = sum(prop)) %>%
    arrange(desc(total_prop)) %>%
    slice(1)
```

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

```
  # Display results
  popular_Q
```

```
# A tibble: 2 x 3
# Groups:   sex [2]
  sex   name  total_prop
  <chr> <chr>      <dbl>
1 F     Quinn    0.00479
2 M     Quinn    0.00685
```

```
  popular_V
```

```
# A tibble: 2 x 3
# Groups:   sex [2]
  sex   name     total_prop
  <chr> <chr>         <dbl>
1 F     Victoria     0.0522
2 M     Victor       0.0235
```

```
popular_X
```

```
# A tibble: 2 x 3
# Groups:   sex [2]
  sex   name   total_prop
  <chr> <chr>       <dbl>
1 F     Ximena    0.00545
2 M     Xavier    0.0327
```

```
# 9. Create a new object that retains all the variables of the babynames package, but crea


# Create a new object with a decade column
babynames_with_decade <- babynames %>%
  mutate(decade = floor(year / 10) * 10)

# 10. What is the mean and median number of female and male babies in each decade?

mean_median_by_decade <- babynames_with_decade %>%
  group_by(decade, sex) %>%
  summarize(
    mean_n = mean(n),
    median_n = median(n),
    .groups = 'drop' # ensures the result is ungrouped after summarizing
  )


# 11.In which decade(s) and year(s), was:

# a) Find the most popular decade(s) and year(s) for "Anil"
anil_popularity <- babynames_with_decade %>%
  filter(name == "Anil") %>%
  arrange(desc(n)) %>%
  slice(1)

# b) Find the most popular decade(s) and year(s) for your supervisor's name (replace "Supe
supervisor_popularity <- babynames_with_decade %>%
  filter(name == "Dylan") %>%
  arrange(desc(n)) %>%
  slice(1)
```

```r
# c) Find the most popular decade(s) and year(s) for "Jack"
jack_popularity <- babynames_with_decade %>%
  filter(name == "Jack") %>%
  arrange(desc(n)) %>%
  slice(1)

# Find the most popular decade(s) and year(s) for "Scott"
scott_popularity <- babynames_with_decade %>%
  filter(name == "Scott") %>%
  arrange(desc(n)) %>%
  slice(1)

# View results
anil_popularity
```

```
# A tibble: 1 x 6
   year sex   name      n      prop decade
  <dbl> <chr> <chr> <int>     <dbl>  <dbl>
1  1989 M     Anil     45 0.0000215   1980
```

```r
supervisor_popularity
```

```
# A tibble: 1 x 6
   year sex   name      n    prop decade
  <dbl> <chr> <chr> <int>   <dbl>  <dbl>
1  2001 M     Dylan 16496 0.00798   2000
```

```r
jack_popularity
```

```
# A tibble: 1 x 6
   year sex   name      n   prop decade
  <dbl> <chr> <chr> <int>  <dbl>  <dbl>
1  1927 M     Jack  12795 0.0110   1920
```

```r
scott_popularity
```

```
# A tibble: 1 x 6
   year sex   name      n   prop decade
  <dbl> <chr> <chr> <int>  <dbl>  <dbl>
1  1971 M     Scott 30918 0.0170   1970
```

6. What is one reason for not including n, but keeping the variable prop?

One reason to to include prop (proportion) of babies of N, is that it will give you standardized measure of the amount of baby names in a given year relative to the amount of babies born that year, providing a better ability to compare names across time periods. The total amount of babies born each year is variable and the standard n (count) would not be a true representation of data.