

# Homework assignment 1

Anil

## Quarto

```
suppressPackageStartupMessages({  
  library(dplyr)  
  library(babynames)  
})
```

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

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

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

```
[1] 1924665      5
```

```
str(babynames)
```

```
tibble [1,924,665 x 5] (S3: tbl_df/tbl/data.frame)  
$ year: num [1:1924665] 1880 1880 1880 1880 1880 1880 1880 1880 1880 1880 ...  
$ sex : chr [1:1924665] "F" "F" "F" "F" ...  
$ name: chr [1:1924665] "Mary" "Anna" "Emma" "Elizabeth" ...  
$ n : int [1:1924665] 7065 2604 2003 1939 1746 1578 1472 1414 1320 1288 ...  
$ prop: num [1:1924665] 0.0724 0.0267 0.0205 0.0199 0.0179 ...
```

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

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

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

```
[1] 1880 2017
```

```
# 5. Create an object from the baby names package that does not include the variable n.
```

```
babynames_no_n <- babynames %>%
  select(-n)
```

```
print(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
```

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

```
# 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.

```
# 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.

```
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
```

```
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
```

```
# 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.

```
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.

```
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
```

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

print(babynames_with_decade)
```

```
# A tibble: 1,924,665 x 6
  year sex  name      n  prop decade
<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
```

```
# 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
  )

print(mean_median_by_decade)

```

```

# A tibble: 28 x 4
  decade sex    mean_n median_n
  <dbl> <chr>   <dbl>    <dbl>
1  1880 F      111.      13
2  1880 M      101.      12
3  1890 F      128.      13
4  1890 M       93.6     12
5  1900 F      131.      12
6  1900 M       94.4     12
7  1910 F      187.      12
8  1910 M      181.      12
9  1920 F      211.      12
10 1920 M      227.      13
# i 18 more rows

```

```

# 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 "Super"
supervisor_popularity <- babynames_with_decade %>%
  filter(name == "Dylan") %>%
  arrange(desc(n)) %>%
  slice(1)

```

```

# 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

```

```

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

```

```

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

```

```

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 include prop (proportion) of babies of N, is that it will give you standardized measure (percentage) of the amount of baby names in a given year relative to the total 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 the data.