Session #3 - Exercises

Adrian C Lo

2/5/2020

INSTRUCTIONS

There are 3 segments with around 10 questions each that increase in difficulty. Fill in the answer within the code chunk. When you wish to test the code chunk, press the *green play button* on the right side of the code chunk to see your output.

The hints will guide you what functions are required. You can access further information with *?x* where x is the function name, e.g. ?print().

Also check this page, specifically sections 8.3 and 10.2 for additional help.

basic operations {base}

- bla
- bla

data import {readr}

There are several datasets available that we will import.

```
suppressPackageStartupMessages( library(readr) )
suppressPackageStartupMessages( library(dplyr) )
```

1. Import the happiness dataset

```
happy <- read_csv("happiness_2019.csv", col_types = cols())
happy
```

```
## # A tibble: 156 x 9
##    `Overall rank` `Country or reg~ Score `GDP per capita` `Social support`
##             <dbl> <chr>            <dbl>            <dbl>            <dbl>
##  1              1 Finland           7.77             1.34             1.59
##  2              2 Denmark           7.6              1.38             1.57
##  3              3 Norway            7.55             1.49             1.58
##  4              4 Iceland           7.49             1.38             1.62
##  5              5 Netherlands       7.49             1.40             1.52
##  6              6 Switzerland       7.48             1.45             1.53
##  7              7 Sweden            7.34             1.39             1.49
##  8              8 New Zealand       7.31             1.30             1.56
##  9              9 Canada            7.28             1.36             1.50
## 10             10 Austria           7.25             1.38             1.48
## # ... with 146 more rows, and 4 more variables: `Healthy life
## #   expectancy` <dbl>, `Freedom to make life choices` <dbl>,
## #   Generosity <dbl>, `Perceptions of corruption` <dbl>
```

data manipulations {dplyr}

```
suppressPackageStartupMessages( library(dplyr) )
suppressPackageStartupMessages( library(tidyr) )
```

In the following questions, you will perform exploratory analysis on the coronavirus. This dataset is contained in the like-named {coronavirus} library.

```
# devtools::install_github("RamiKrispin/coronavirus")
suppressPackageStartupMessages( library(coronavirus) )
```

1. Worldwide, how many confirmed cases of coronavirus have been found?

```
coronavirus %>%
    filter(type == "confirmed") %>%
    summarise(worldwide_confirmed = sum(cases))
```

```
## # A tibble: 1 x 1
##   worldwide_confirmed
##                 <int>
## 1               88371
```

2. Worldwide, how many people died from coronavirus?

```
coronavirus %>%
    filter(type == "death") %>%
    summarise(worldwide_confirmed = sum(cases))
```

```
## # A tibble: 1 x 1
##   worldwide_confirmed
##                 <int>
## 1                2996
```

3. Which are the top 5 countries with the most cases of confirmed coronavirus?

```
coronavirus %>%
    filter(type == "confirmed") %>%
    group_by(Country.Region) %>%
    summarise(confirmed = sum(cases)) %>%
    arrange(desc(confirmed)) %>%
    head(5)
```

```
## # A tibble: 5 x 2
##   Country.Region confirmed
##   <chr>              <int>
## 1 Mainland China     79826
## 2 South Korea         3736
## 3 Italy               1694
## 4 Iran                 978
## 5 Others               705
```

4. From which country is the last confirmed case?

```
coronavirus %>%
    filter(type == "confirmed") %>%
    arrange(desc(date)) %>%
    head(1)
```

```
## # A tibble: 1 x 7
##   Province.State Country.Region   Lat  Long date       cases type
##   <chr>          <chr>          <dbl> <dbl> <date>     <int> <chr>
## 1 ""             Armenia         40.1  45.0 2020-03-01     1 confirmed
```

5. From which country were the latest recovered cases?

```
coronavirus %>%
    filter(type == "recovered") %>%
    arrange(desc(date)) %>%
    head(1)
```

```
## # A tibble: 1 x 7
##   Province.State Country.Region   Lat  Long date       cases type
##   <chr>          <chr>          <dbl> <dbl> <date>     <int> <chr>
## 1 ""             Iran              32    53 2020-03-01    52 recovered
```

6. When and where were the most confirmed cases detected on a single day?

```
coronavirus %>%
    filter(type == "confirmed") %>%
    arrange(desc(cases)) %>%
    head(1)
```

```
## # A tibble: 1 x 7
##   Province.State Country.Region   Lat  Long date       cases type
##   <chr>          <chr>          <dbl> <dbl> <date>     <int> <chr>
## 1 Hubei          Mainland China  31.0  112. 2020-02-13 14840 confirmed
```

7. Were there any false positive confirmed cases?

```
coronavirus %>%
    filter(type == "confirmed") %>%
    filter(cases < 0)
```

```
## # A tibble: 8 x 7
##   Province.State        Country.Region   Lat   Long date       cases type
##   <chr>                 <chr>          <dbl>  <dbl> <date>     <int> <chr>
## 1 ""                    Japan           36    138   2020-01-23    -1 confi~
## 2 Queensland            Australia      -28.0  153.  2020-01-31    -1 confi~
## 3 Queensland            Australia      -28.0  153.  2020-02-02    -1 confi~
## 4 ""                    Japan           36    138   2020-02-07   -20 confi~
## 5 Lackland, TX (From D~ US              29.4  -98.6 2020-02-24    -2 confi~
## 6 Omaha, NE (From Diam~ US              41.3  -96.0 2020-02-24   -11 confi~
## 7 Travis, CA (From Dia~ US              38.3 -122.  2020-02-24    -5 confi~
## 8 From Diamond Princess Australia       35.4  140.  2020-02-29    -8 confi~
```

8. Which are the top 3 countries that have more than 20 deaths?

```
coronavirus %>%
    filter(type == "death") %>%
    group_by(Country.Region) %>%
    summarise(death = sum(cases)) %>%
    filter(death > 20) %>%
    arrange(desc(death))
```

```
## # A tibble: 3 x 2
##   Country.Region death
##   <chr>          <int>
```

```
## 1 Mainland China  2870
## 2 Iran              54
## 3 Italy             34
```

9. How many countries have a recovered-confirmed ratio of more than 0.60?

```
coronavirus %>%
    filter(type %in% c("confirmed", "recovered")) %>%
    group_by(Country.Region, type) %>%
    summarise(cases = sum(cases)) %>%

    # from {tidyr}: to have values put in separate columns
    pivot_wider(names_from = "type", values_from = "cases") %>%
    mutate(recovered = ifelse(is.na(recovered), 0, recovered)) %>%
    mutate(proportion = recovered / confirmed) %>%
    filter(proportion > 0.60) %>%
    arrange(desc(proportion))
```

```
## # A tibble: 10 x 4
## # Groups:   Country.Region [10]
##    Country.Region confirmed recovered proportion
##    <chr>              <int>     <dbl>      <dbl>
##  1 Cambodia               1         1      1
##  2 India                  3         3      1
##  3 Nepal                  1         1      1
##  4 Russia                 2         2      1
##  5 Sri Lanka              1         1      1
##  6 Vietnam               16        16      1
##  7 Macau                 10         8      0.8
##  8 Singapore            106        72      0.679
##  9 Thailand              42        28      0.667
## 10 Malaysia              29        18      0.621
```

10. What is the recovery-confirmed ratio for Italy?

```
coronavirus %>%
    filter(Country.Region == "Italy") %>%
    filter(type %in% c("confirmed", "recovered")) %>%
    group_by(type) %>%
    summarise(cases = sum(cases)) %>%

    # from {tidyr}: to have values put in separate columns
    pivot_wider(names_from = "type", values_from = "cases") %>%
    mutate(proportion = recovered / confirmed)
```

```
## # A tibble: 1 x 3
##   confirmed recovered proportion
##       <int>     <int>      <dbl>
## 1      1694        83     0.0490
```

In the following questions, you will explore the popularity of certain babynames. This dataset can be found in the like-named {babynames} library.

```
suppressPackageStartupMessages( library(babynames) )
```

12. What is the proportion of female babies that are called "Anna" in 1880 and 2017?

```
babynames %>%
    filter(sex == "F" & name == "Anna") %>%
    filter(year %in% c(1880,2017))
```

```
## # A tibble: 2 x 5
##    year sex   name      n    prop
##   <dbl> <chr> <chr> <int>   <dbl>
## 1  1880 F     Anna   2604 0.0267
## 2  2017 F     Anna   4520 0.00241
```

13. From 1880-1900, which was the most popular name for boys and girls?

```
babynames %>%
    filter(between(year, 1880, 1900)) %>%
    group_by(name, sex) %>%
    summarise(n = sum(n)) %>%
    arrange(desc(n)) %>%
    group_by(sex) %>%
    slice(1)
```

```
## # A tibble: 2 x 3
## # Groups:   sex [2]
##   name  sex        n
##   <chr> <chr>  <int>
## 1 Mary  F     239510
## 2 John  M     180444
```

14. From 1880-1900, which was the least popular name for boys and girls?

```
babynames %>%
    filter(between(year, 1880, 1900)) %>%
    group_by(name, sex) %>%
    summarise(n = sum(n)) %>%
    arrange(n) %>%
    group_by(sex) %>%
    slice(1)
```

```
## # A tibble: 2 x 3
## # Groups:   sex [2]
##   name    sex       n
##   <chr>   <chr> <int>
## 1 Abelina F         5
## 2 Abron   M         5
```

15. From 2000-2017, which was the most popular name for boys and girls?

```
babynames %>%
    filter(between(year, 2000, 2017)) %>%
    group_by(name, sex) %>%
    summarise(n = sum(n)) %>%
    arrange(desc(n)) %>%
    group_by(sex) %>%
    slice(1)
```

```
## # A tibble: 2 x 3
## # Groups:   sex [2]
##   name  sex        n
##   <chr> <chr>  <int>
```

```
## 1 Emma  F     339802
## 2 Jacob M     413884
```

16. From 2000-2017, which was the least popular name for boys and girls?

```
babynames %>%
    filter(between(year, 2000, 2017)) %>%
    group_by(name, sex) %>%
    summarise(n = sum(n)) %>%
    arrange(n) %>%
    group_by(sex) %>%
    slice(1)
```

```
## # A tibble: 2 x 3
## # Groups:   sex [2]
##   name  sex       n
##   <chr> <chr> <int>
## 1 Aada  F         5
## 2 Aabir M         5
```

15. For girls, what were the most popular name in 1880, 1917, 1943 and 2017?

```
babynames %>%
    filter(sex == "F") %>%
    filter(year %in% c(1880,1917,1943,2017)) %>%
    group_by(year,name) %>%
    summarise(n = sum(n)) %>%
    ungroup() %>% arrange(year, desc(n)) %>%
    group_by(year) %>%
    slice(1)
```

```
## # A tibble: 4 x 3
## # Groups:   year [4]
##    year name      n
##   <dbl> <chr> <int>
## 1  1880 Mary   7065
## 2  1917 Mary  64281
## 3  1943 Mary  66169
## 4  2017 Emma  19738
```

16. How many different boy names were there between 1880-1900?

```
babynames %>%
    filter(sex == "M") %>%
    filter(between(year,1880,1900)) %>%
    pull(name) %>%
    unique() %>%
    length()
```

```
## [1] 2411
```

17. How many different boy names were there between 2000-2017? Did we diversify compared to the previous era?

```
babynames %>%
    filter(sex == "M") %>%
    filter(between(year,2000,2017)) %>%
    pull(name) %>%
```

```
    unique() %>%
    length()
```

```
## [1] 30118
```

18. What is the popularity of your own name in 2017?

```
babynames %>%
    filter(name == "Adrian" & year == 2017)
```

```
## # A tibble: 2 x 5
##    year sex   name        n      prop
##   <dbl> <chr> <chr>   <int>     <dbl>
## 1  2017 F     Adrian    114 0.0000608
## 2  2017 M     Adrian   6203 0.00316
```

data visualizations {ggplot2}

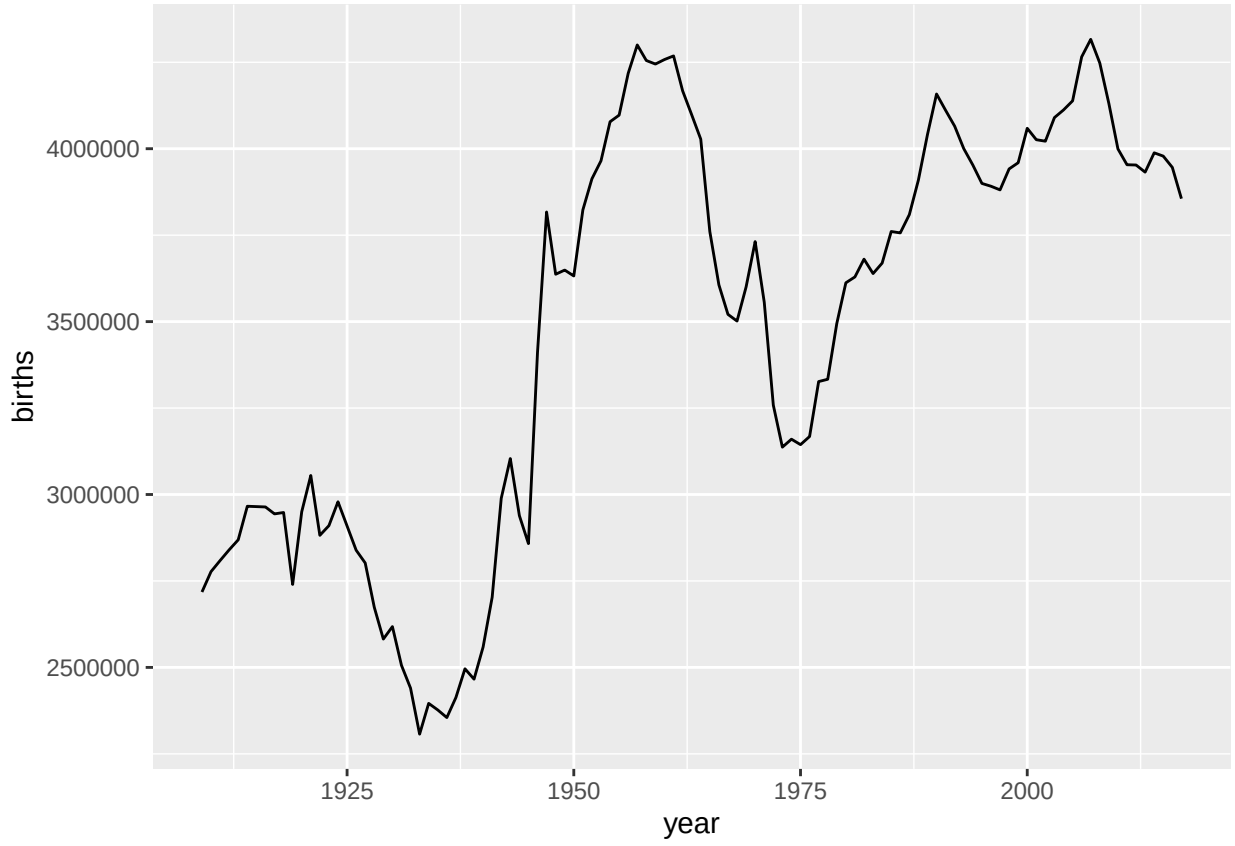
```
suppressPackageStartupMessages( library(ggplot2) )
```

In the following questions, you will visualize the popularity of certain babynames as well as the number of births. This dataset can be found in the like-named {babynames} library.

```
suppressPackageStartupMessages( library(babynames) )
```

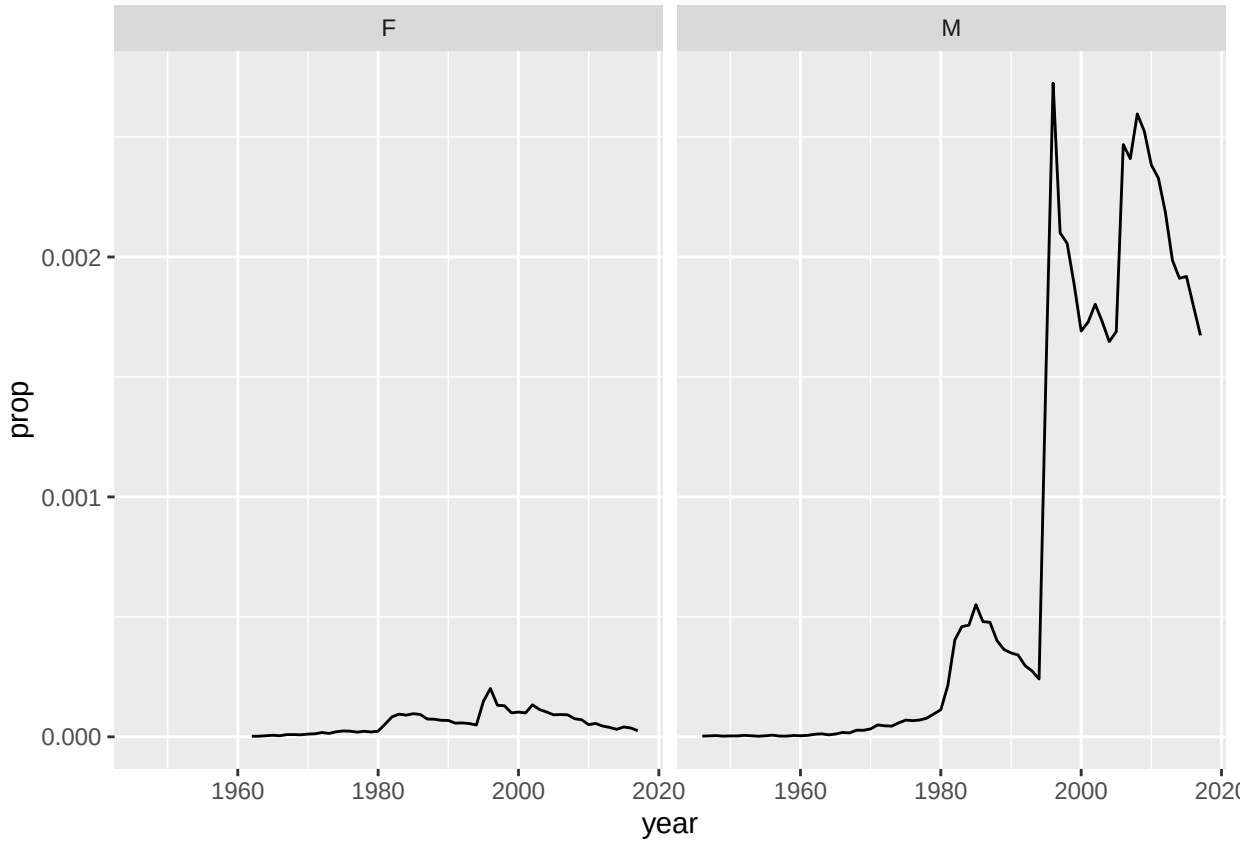
1. Plot the number of birth over time from the births dataset.

```
births %>%
    ggplot(., aes(year, births)) + geom_line()
```



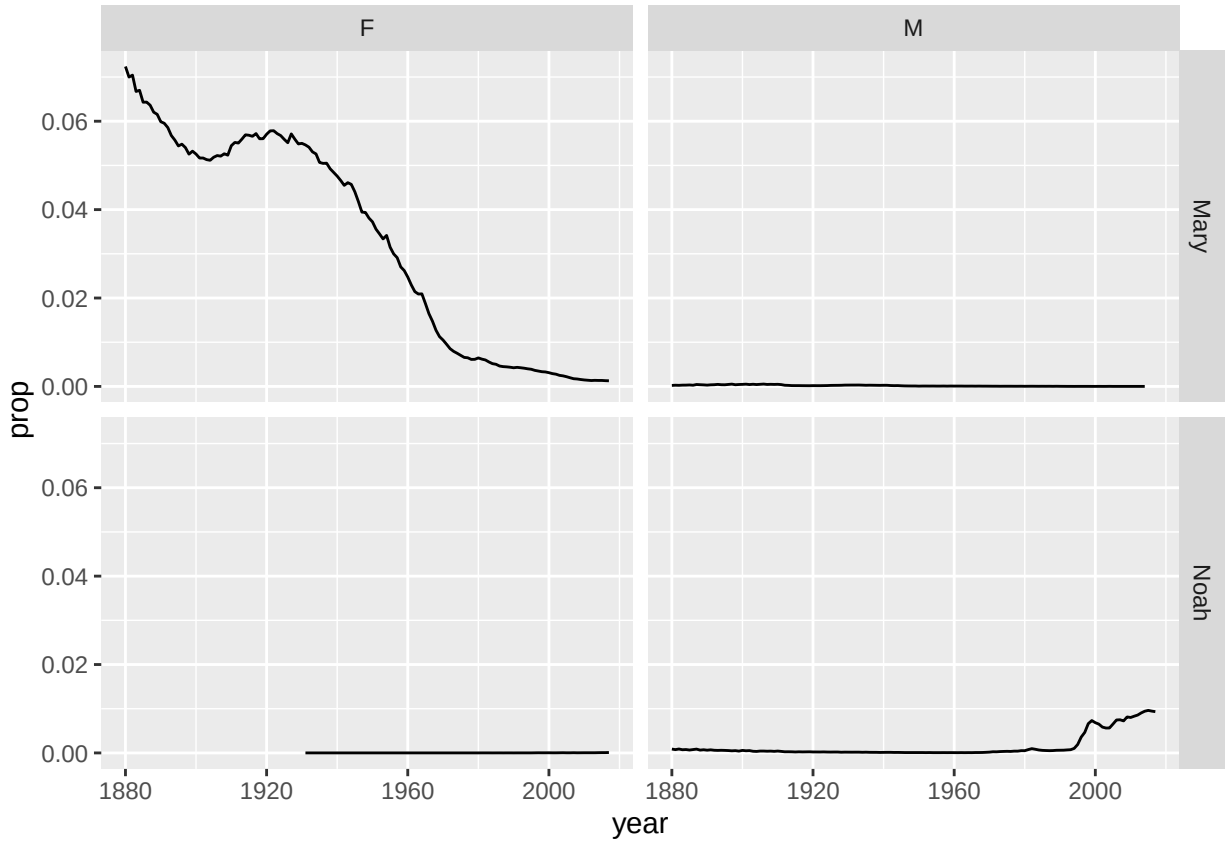
2. Plot the development of the name "Tristan" over time for each sex from the babynames dataset.

```
babynames %>%
    filter(name == "Tristan") %>%
    ggplot(., aes(year, prop)) +
    geom_line() +
    facet_wrap(~ sex)
```



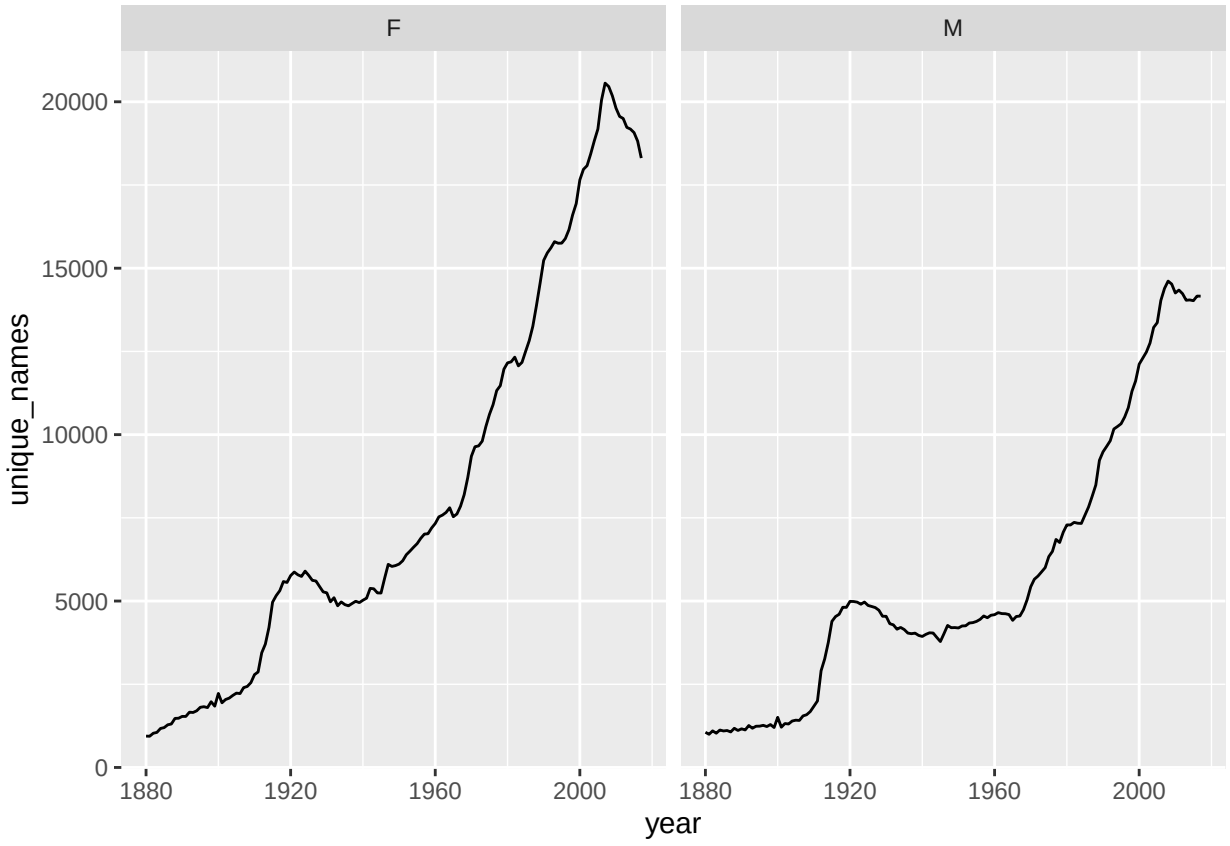
3. Plot the development of the name "Mary" and "Noah" over time for each sex. What can you say about the development of these names?

```
babynames %>%
    filter(name %in% c("Mary","Noah")) %>%
    ggplot(., aes(year, prop)) +
    geom_line() +
    facet_grid(name ~ sex)
```



4. Extract from the babynames dataset for each year, the number of unique names for boys and girls. Plot these over time side-by-side. What is the trend? Which sex has more diversified names? Any peculiar trends?

```
babynames %>%
    group_by(year, sex) %>%
    summarise(unique_names = n()) %>%
    ggplot(., aes(year, unique_names)) +
    geom_line() +
    facet_grid(. ~ sex)
```

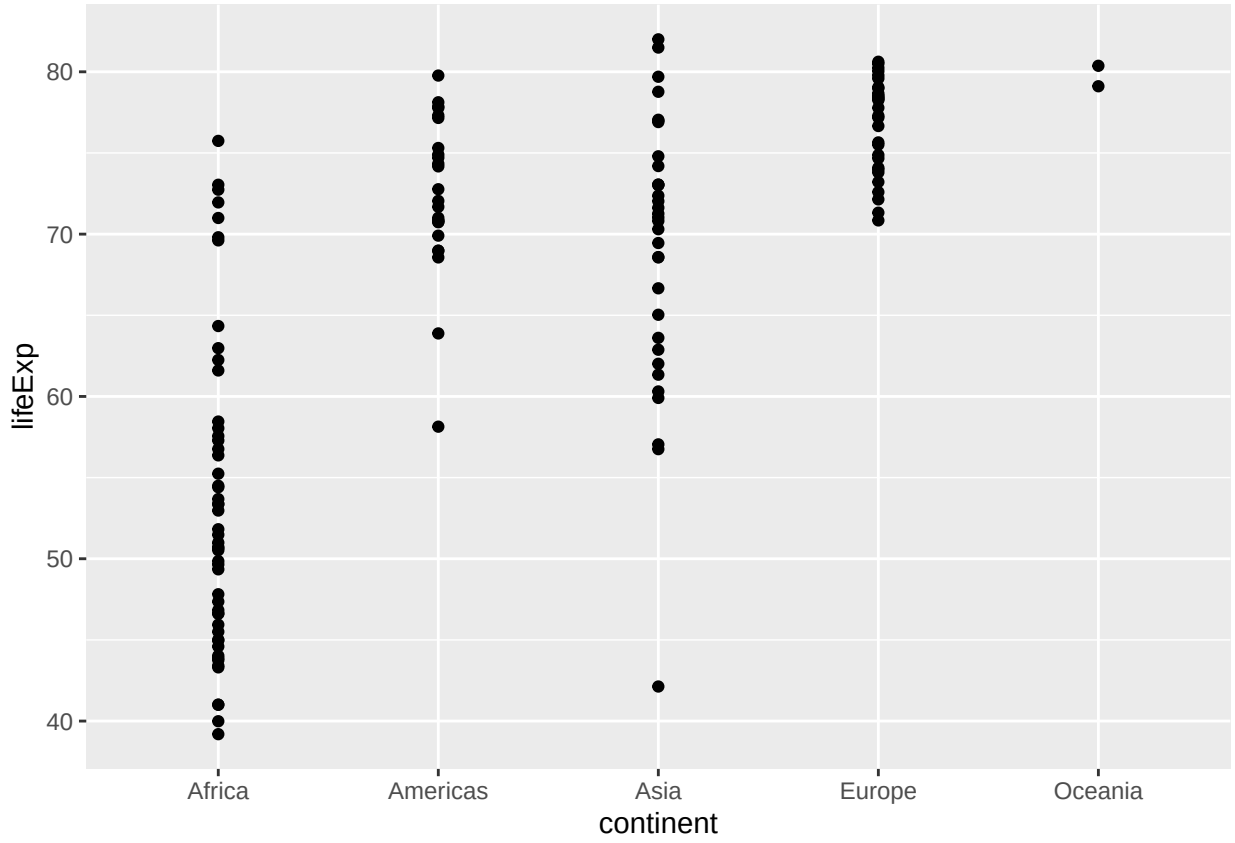


In the following questions, you will visualize the economic parameteres from the gapminder dataset. This dataset can be found in the like-named {gapminder} library.

```
suppressPackageStartupMessages( library(gapminder) )
```

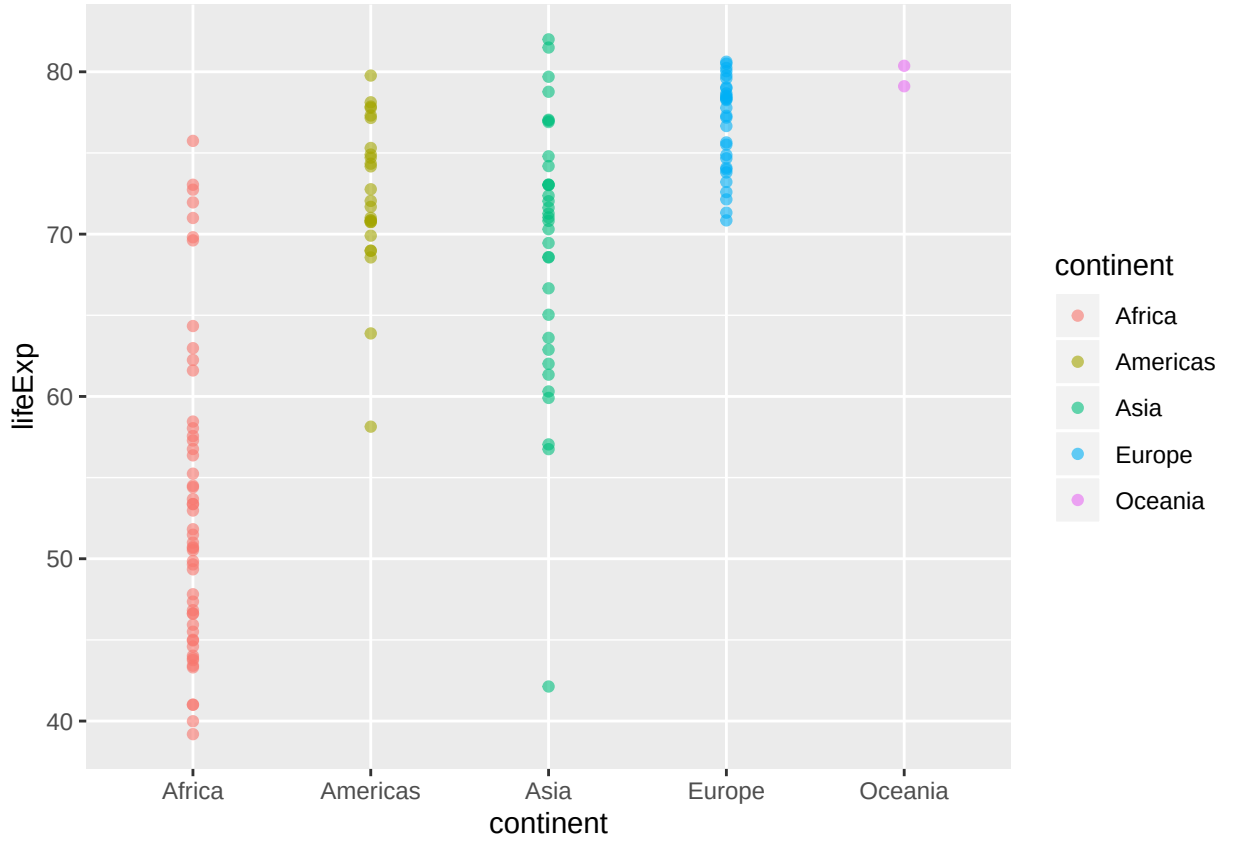
5. Plot the life expectancy for each continent from the year 2002 as individual points

```
gapminder %>%
    filter(year == 2002) %>%
    ggplot(., aes(continent, lifeExp)) +
    geom_point()
```



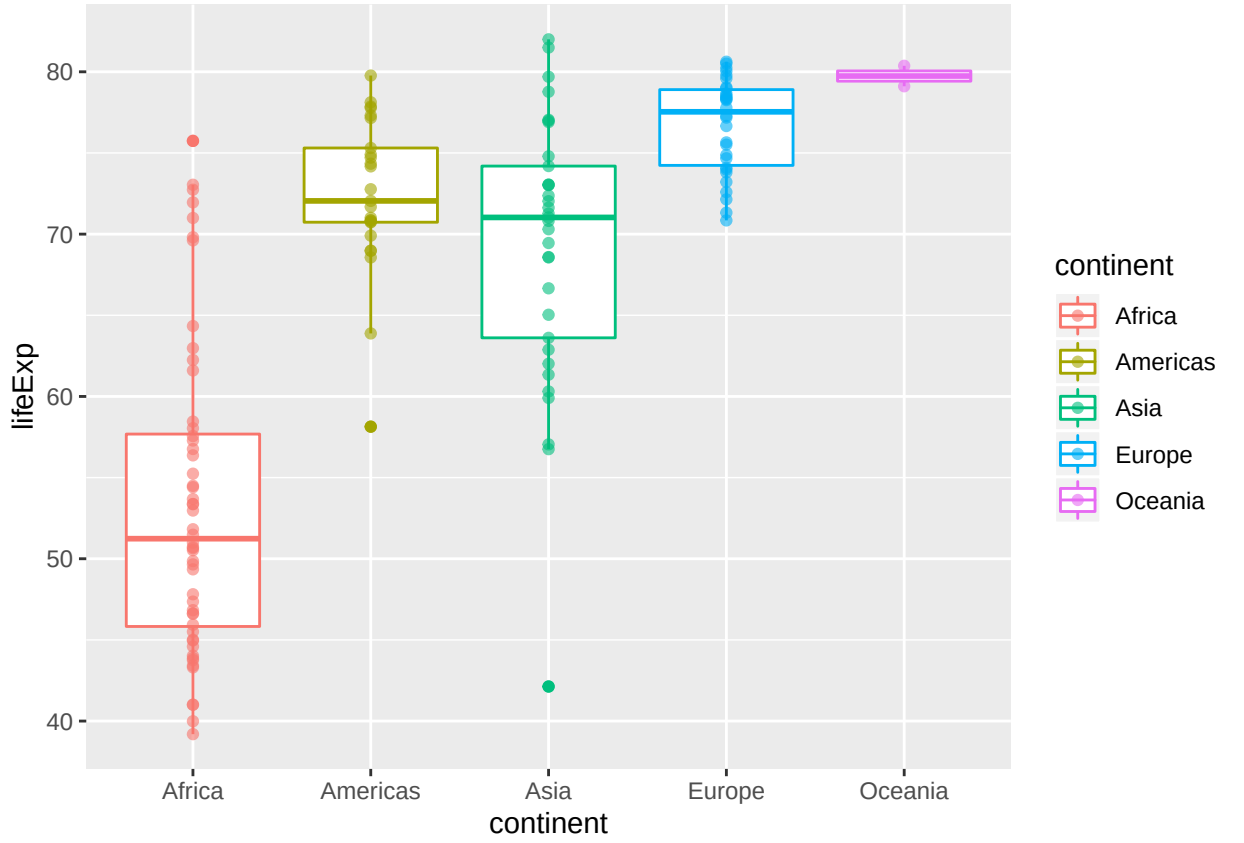
6. Continue from the previous question: add some transparency in the dots and give each continent a different color.

```
gapminder %>%
    filter(year == 2002) %>%
    ggplot(., aes(continent, lifeExp, color = continent)) +
    geom_point(alpha = 0.60)
```



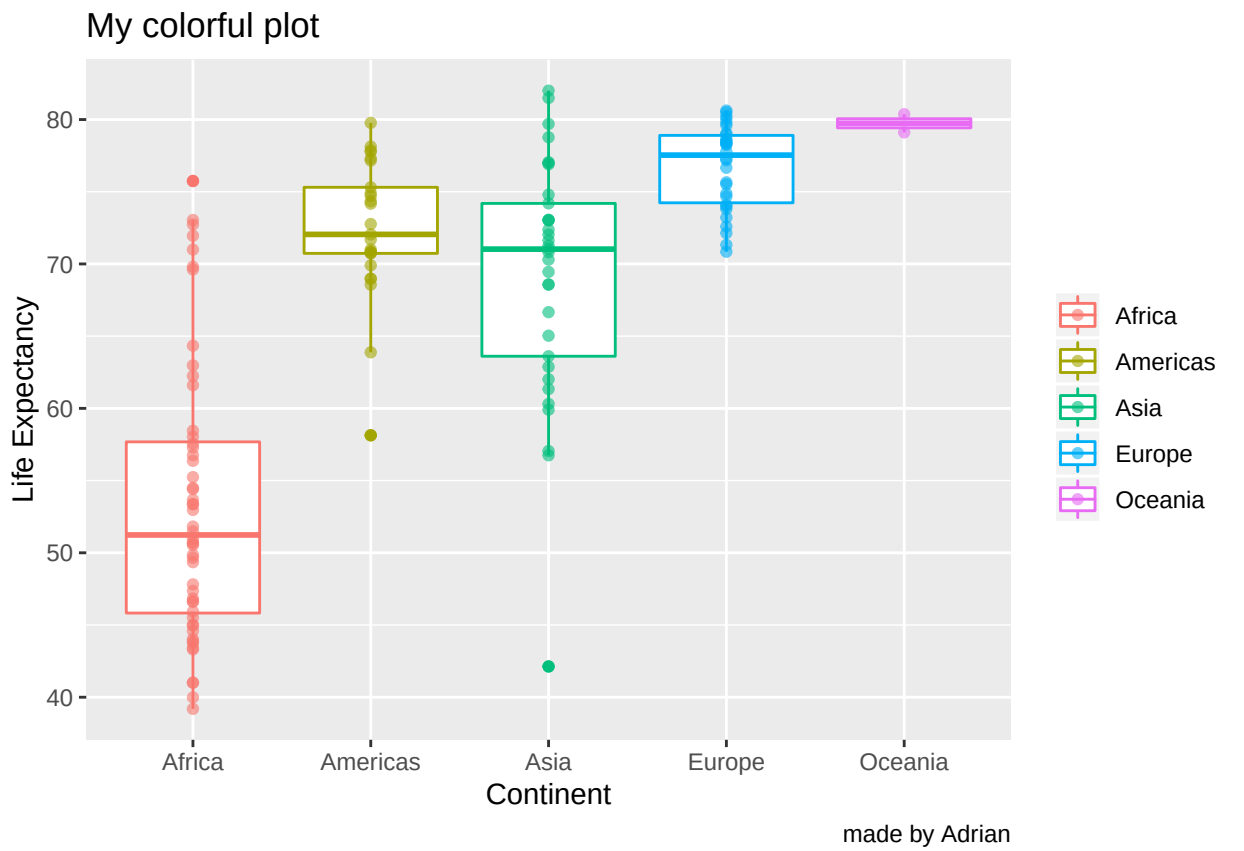
7. Continue from the previous question: add a boxplot to the graph. Add it in such a way that the individual points will still be visible.

```
gapminder %>%
    filter(year == 2002) %>%
    ggplot(., aes(continent, lifeExp, color = continent)) +
    geom_boxplot() +
    geom_point(alpha = 0.60)
```



8. Continue from the previous question: Adjust the labels so they are more presentable. Change the title on the y-axis to "Life Expectancy", and the title on the x-axis to "Continent". Add a title "My colorful plot". Also add a caption with "made by {your name}". Finally, remove the legend title.

```
gapminder %>%
    filter(year == 2002) %>%
    ggplot(., aes(continent, lifeExp, color = continent)) +
    geom_boxplot() +
    geom_point(alpha = 0.60) +
    labs(x = "Continent", y = "Life Expectancy",
         title = "My colorful plot", caption = "made by Adrian") +
    theme(legend.title = element_blank())
```



data mining {rvest}

```
suppressPackageStartupMessages( library(rvest) )
```