Example Final Seminar

Sergio Uribe

Contents

Aim

This is a sample seminar about the final work that each one should present.

The objective of the seminar is to demonstrate your skills in performing an exploratory analysis.

The minimum elements that the seminar should have include

- pose a question that can be answered with data
- load data into R, ideally from an online source
- load r packages
- comment in detail all the code
- explore the data
- identify tabulated data in tidy format
- identify the location and proportion of null data in the dataset
- create one or more summary tables
- Create one or more exploratory graphics
- Data wrangling: Apply one or more of these commands: filter, select, mutate, pivot

- answer the question posed with a correctly formatted graphic
- generate a codebook using the reporteR package (or dataMaid if reporteR is not available yet)
- use the rmarkdown format to integrate text and code
- export the document together with the code to a pdf or doc file

Minimum code must include

- Packages
  - pacman::p_load()
- Data import
  - read_csv()
- Data exploration
  - head()
  - summary()
  - dim()
- Data wrangling
  - %>%
  - filter()
  - select()
  - mutate()
  - group_by()
  - summarize()
- Tables
  - gtsummary::tbl_summary()
- Graphs
  - ggplot()
- Codebook
  - dataMaid::codebook()

Extra points:

- use packages that we haven't seen in classes
- use join_
- use log10 transformations for axes

Below is a sample seminar. Your code may be more or less than the example, there is no maximum or minimum. The important thing is that:

- you must use the minimum commands listed before,
- you must comment most of your code to document the steps of your analysis and
- you must export your code script to a pdf or docx document (go to Preview Notebook tab and select the format to export, detailed instructions here)

IMPORTANT: Your code should be executable

Some recommendations

Write clear code (select code, CTRL+SHIFT+A): it will help you when something doesn't work

In case something doesn't work, don't despair, that happens to everyone, beginners and advanced. The important thing is to be able to detect the error. In case some error message appears, I suggest you first verify that your code doesn't have some obvious error (like some orphan parenthesis, a period instead of a comma, etc) and if the error persists, copy and paste the error message in google to find out the solution.

Remember: there is no problem that cannot be solved without the proper use of google or a hammer!

SEMINAR EXAMPLE

Question

What's the birth rate for european countries and for continents and What is the birth rate for the Baltic countries?

Packages

```
# install the pacman package if is not installed previously, uncomment next line

# install.packages("pacman")

pacman::p_load(tidyverse, # several packages for data science
               visdat,  # to visualize NAs
               gtsummary, # for nice tables
               dataMaid, # for the codebook
               janitor) # for data cleaning
```

Dataset

Found in the World Bank data

https://data.worldbank.org/indicator/SP.DYN.CBRT.IN Found the Birth rate, crude (per 1,000 people) Is in zip format, I created a copy online in google drive, published as a csv file and imported into R I will call my dataset as df for Data Frame

```
df <- read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vStv7Pr69DtRKv6Nw6gVBep8hbT3pEeO6B1vNwx
```

```
## Warning: Missing column names filled in: 'X3' [3], 'X4' [4], 'X5' [5], 'X6' [6],
## 'X7' [7], 'X8' [8], 'X9' [9], 'X10' [10], 'X11' [11], 'X12' [12], 'X13' [13],
## 'X14' [14], 'X15' [15], 'X16' [16], 'X17' [17], 'X18' [18], 'X19' [19],
## 'X20' [20], 'X21' [21], 'X22' [22], 'X23' [23], 'X24' [24], 'X25' [25],
## 'X26' [26], 'X27' [27], 'X28' [28], 'X29' [29], 'X30' [30], 'X31' [31],
## 'X32' [32], 'X33' [33], 'X34' [34], 'X35' [35], 'X36' [36], 'X37' [37],
## 'X38' [38], 'X39' [39], 'X40' [40], 'X41' [41], 'X42' [42], 'X43' [43],
## 'X44' [44], 'X45' [45], 'X46' [46], 'X47' [47], 'X48' [48], 'X49' [49],
## 'X50' [50], 'X51' [51], 'X52' [52], 'X53' [53], 'X54' [54], 'X55' [55],
## 'X56' [56], 'X57' [57], 'X58' [58], 'X59' [59], 'X60' [60], 'X61' [61],
## 'X62' [62], 'X63' [63], 'X64' [64], 'X65' [65]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   ‘Data Source‘ = col_character(),
##   ‘World Development Indicators‘ = col_character(),
##   X3 = col_character(),
##   X4 = col_character()
## )
## i Use ‘spec()‘ for the full column specifications.
```

Data cleaning

```r
head(df)
```

```
## # A tibble: 6 x 65
##   ‘Data Source‘ ‘World Developm~ X3    X4        X5     X6     X7     X8     X9
##   <chr>         <chr>            <chr> <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 <NA>          <NA>             <NA>  <NA>    NA     NA     NA     NA     NA
## 2 Last Updated~ 2020-10-15       <NA>  <NA>    NA     NA     NA     NA     NA
## 3 <NA>          <NA>             <NA>  <NA>    NA     NA     NA     NA     NA
## 4 Country Name  Country Code     Indi~ Indi~ 1960   1961   1962   1963   1964
## 5 Aruba         ABW              Birt~ SP.D~   35.7   34.5   33.3   32.0   30.7
## 6 Afghanistan   AFG              Birt~ SP.D~   51.3   51.4   51.5   51.5   51.6
## # ... with 56 more variables: X10 <dbl>, X11 <dbl>, X12 <dbl>, X13 <dbl>,
## #   X14 <dbl>, X15 <dbl>, X16 <dbl>, X17 <dbl>, X18 <dbl>, X19 <dbl>,
## #   X20 <dbl>, X21 <dbl>, X22 <dbl>, X23 <dbl>, X24 <dbl>, X25 <dbl>,
## #   X26 <dbl>, X27 <dbl>, X28 <dbl>, X29 <dbl>, X30 <dbl>, X31 <dbl>,
## #   X32 <dbl>, X33 <dbl>, X34 <dbl>, X35 <dbl>, X36 <dbl>, X37 <dbl>,
## #   X38 <dbl>, X39 <dbl>, X40 <dbl>, X41 <dbl>, X42 <dbl>, X43 <dbl>,
## #   X44 <dbl>, X45 <dbl>, X46 <dbl>, X47 <dbl>, X48 <dbl>, X49 <dbl>,
## #   X50 <dbl>, X51 <dbl>, X52 <dbl>, X53 <dbl>, X54 <dbl>, X55 <dbl>,
## #   X56 <dbl>, X57 <dbl>, X58 <dbl>, X59 <dbl>, X60 <dbl>, X61 <dbl>,
## #   X62 <dbl>, X63 <dbl>, X64 <dbl>, X65 <dbl>
```

Seems that there are some extra rows at the top of the dataset. I will re-read, adding the option to skip those rows

```r
df <- read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vStv7Pr69DtRKv6Nw6gVBep8hbT3pEe06B1vNwx
               skip = 4) # add the option to skip 4 rows when importing the file
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   ‘Country Name‘ = col_character(),
##   ‘Country Code‘ = col_character(),
##   ‘Indicator Name‘ = col_character(),
##   ‘Indicator Code‘ = col_character(),
##   ‘2019‘ = col_logical(),
##   ‘2020‘ = col_logical()
## )
## i Use ‘spec()‘ for the full column specifications.
```

Now it’s ok

I will standardize the names to facilitate handling

```
df <- df %>%                # create a new dataset with the former, dataset
  janitor::clean_names()  # and clean all the names
```

the last two columns are empty, so I will delete it

```
df <- df %>%
  select(-x2019,  # here I unselect these columns
         -x2020)
```

Data wrangling

Check the dimensions

```
dim(df)
```

```
## [1] 264  63
```

Check the variables included

```
glimpse(df)
```

```
## Rows: 264
## Columns: 63
## $ country_name   <chr> "Aruba", "Afghanistan", "Angola", "Albania", "Andorr...
## $ country_code   <chr> "ABW", "AFG", "AGO", "ALB", "AND", "ARB", "ARE", "AR...
## $ indicator_name <chr> "Birth rate, crude (per 1,000 people)", "Birth rate,...
## $ indicator_code <chr> "SP.DYN.CBRT.IN", "SP.DYN.CBRT.IN", "SP.DYN.CBRT.IN"...
## $ x1960          <dbl> 35.67900, 51.27900, 49.08000, 40.92400, NA, 47.79008...
## $ x1961          <dbl> 34.52900, 51.37300, 48.77900, 40.36800, NA, 47.55839...
## $ x1962          <dbl> 33.3200, 51.4570, 48.5470, 39.6270, NA, 47.3276, 46....
## $ x1963          <dbl> 32.05000, 51.53000, 48.43000, 38.72300, NA, 47.09162...
## $ x1964          <dbl> 30.73700, 51.58900, 48.45000, 37.69500, NA, 46.84421...
## $ x1965          <dbl> 29.4130, 51.6310, 48.6220, 36.5990, NA, 46.5771, 43....
## $ x1966          <dbl> 28.12100, 51.65200, 48.93600, 35.49600, NA, 46.28291...
## $ x1967          <dbl> 26.90800, 51.65000, 49.34300, 34.43500, NA, 45.96055...
## $ x1968          <dbl> 25.81700, 51.62300, 49.78700, 33.45800, NA, 45.61137...
## $ x1969          <dbl> 24.87200, 51.57400, 50.23100, 32.59000, NA, 45.23716...
## $ x1970          <dbl> 24.09900, 51.50200, 50.61900, 31.83700, NA, 44.84362...
## $ x1971          <dbl> 23.50500, 51.41100, 50.90300, 31.18300, NA, 44.44035...
## $ x1972          <dbl> 23.06800, 51.30300, 51.06200, 30.58700, NA, 44.03865...
## $ x1973          <dbl> 22.76000, 51.18400, 51.09400, 30.01900, NA, 43.64783...
## $ x1974          <dbl> 22.56100, 51.05800, 51.00500, 29.47300, NA, 43.27485...
## $ x1975          <dbl> 22.45200, 50.93000, 50.82500, 28.94900, NA, 42.92493...
## $ x1976          <dbl> 22.41400, 50.80300, 50.60000, 28.45500, NA, 42.60063...
## $ x1977          <dbl> 22.4240, 50.6780, 50.3860, 28.0040, NA, 42.2929, 29....
## $ x1978          <dbl> 22.45400, 50.55500, 50.22600, 27.60600, NA, 41.98993...
## $ x1979          <dbl> 22.47800, 50.43600, 50.13900, 27.26200, NA, 41.68051...
## $ x1980          <dbl> 22.47200, 50.32100, 50.13400, 26.98100, NA, 41.34983...
## $ x1981          <dbl> 22.42400, 50.21000, 50.20700, 26.77200, NA, 40.98353...
## $ x1982          <dbl> 22.32900, 50.09800, 50.32200, 26.62700, NA, 40.57024...
## $ x1983          <dbl> 22.18700, 49.98400, 50.44900, 26.52800, NA, 40.10007...
## $ x1984          <dbl> 21.98900, 49.86500, 50.56900, 26.45200, NA, 39.56784...
## $ x1985          <dbl> 21.72600, 49.73500, 50.66300, 26.36700, NA, 38.96898...
## $ x1986          <dbl> 21.39700, 49.58600, 50.71200, 26.24100, 11.90000, 38...
## $ x1987          <dbl> 21.00800, 49.41800, 50.71100, 26.04700, 11.00000, 37...
## $ x1988          <dbl> 20.5700, 49.2360, 50.6570, 25.7620, 11.6000, 36.8177...
## $ x1989          <dbl> 20.08900, 49.04800, 50.54700, 25.37200, 12.50000, 36...
## $ x1990          <dbl> 19.57100, 48.88000, 50.38300, 24.86700, 11.90000, 35...
## $ x1991          <dbl> 19.02100, 48.76300, 50.16800, 24.24500, 11.90000, 34...
## $ x1992          <dbl> 18.44600, 48.70900, 49.91900, 23.52900, 12.10000, 33...
## $ x1993          <dbl> 17.85900, 48.71700, 49.65200, 22.74200, 11.40000, 33...
## $ x1994          <dbl> 17.27000, 48.77000, 49.37800, 21.90200, 10.90000, 32...
## $ x1995          <dbl> 16.69100, 48.83500, 49.11300, 21.02000, 11.00000, 31...
## $ x1996          <dbl> 16.13200, 48.87000, 48.87000, 20.10600, 10.90000, 30...
## $ x1997          <dbl> 15.59800, 48.83300, 48.65200, 19.17300, 11.20000, 30...
## $ x1998          <dbl> 15.09000, 48.68800, 48.46000, 18.23800, 11.90000, 29...
## $ x1999          <dbl> 14.61500, 48.41900, 48.29300, 17.32100, 12.60000, 29...
## $ x2000          <dbl> 14.17300, 48.02100, 48.15000, 16.43600, 11.30000, 28...
## $ x2001          <dbl> 13.76200, 47.50500, 48.02700, 15.59000, 11.80000, 28...
## $ x2002          <dbl> 13.37500, 46.90100, 47.91100, 14.79000, 11.20000, 28...
## $ x2003          <dbl> 13.01000, 46.23100, 47.78600, 14.04800, 10.30000, 27...
## $ x2004          <dbl> 12.66700, 45.50700, 47.63900, 13.38100, 10.90000, 27...
## $ x2005          <dbl> 12.34800, 44.72300, 47.45300, 12.82100, 10.70000, 27...
## $ x2006          <dbl> 12.05300, 43.87000, 47.21500, 12.39800, 10.60000, 27...
## $ x2007          <dbl> 11.78800, 42.94400, 46.92000, 12.11800, 10.10000, 27...
## $ x2008          <dbl> 11.556, 41.949, 46.563, 11.973, 10.400, 27.463, 12.5...
## $ x2009          <dbl> 11.361, 40.903, 46.143, 11.945, 9.900, 27.496, 12.20...
## $ x2010          <dbl> 11.21400, 39.82900, 45.65600, 12.00100, 9.80000, 27....
## $ x2011          <dbl> 11.12300, 38.75000, 45.10200, 12.10000, NA, 27.48487...
## $ x2012          <dbl> 11.0900, 37.6900, 44.4930, 12.1970, 9.5000, 27.3893,...
## $ x2013          <dbl> 11.11100, 36.67000, 43.84700, 12.25700, NA, 27.21144...
## $ x2014          <dbl> 11.17900, 35.70600, 43.18200, 12.25900, NA, 26.94078...
## $ x2015          <dbl> 11.28100, 34.80900, 42.52000, 12.19700, NA, 26.57699...
## $ x2016          <dbl> 11.4040, 33.9810, 41.8820, 12.0800, 8.8000, 26.1348,...
## $ x2017          <dbl> 11.53200, 33.21100, 41.28100, 11.93400, NA, 25.64801...
## $ x2018          <dbl> 11.65200, 32.48700, 40.72900, 11.78000, 7.20000, 25....
```

Check the content of some variables with simple tables

```
table(df$indicator_name)
```

```
##
## Birth rate, crude (per 1,000 people)
##                                  264
```

```
table(df$indicator_code)
```

```
##
## SP.DYN.CBRT.IN
##            264
```

So, both columns are keys, that is some constant and not variables, so I will delete it

```
df <- df %>%
  select(-indicator_name,
         -indicator_code)
```

Check

```
summary(df)
```

```
##  country_name       country_code           x1960           x1961
##  Length:264         Length:264         Min.   :13.40   Min.   :13.70
##  Class :character   Class :character   1st Qu.:27.81   1st Qu.:26.90
##  Mode  :character   Mode  :character   Median :42.69   Median :42.58
##                                        Mean   :38.13   Mean   :37.81
##                                        3rd Qu.:47.29   3rd Qu.:47.21
##                                        Max.   :58.12   Max.   :58.19
##                                        NA’s   :25      NA’s   :26
##      x1962           x1963           x1964           x1965
##  Min.   :12.90   Min.   :13.10   Min.   :13.10   Min.   :13.10
##  1st Qu.:28.41   1st Qu.:28.33   1st Qu.:27.56   1st Qu.:26.67
##  Median :42.18   Median :42.16   Median :41.83   Median :41.19
##  Mean   :37.89   Mean   :37.80   Mean   :37.35   Mean   :36.85
##  3rd Qu.:47.01   3rd Qu.:46.83   3rd Qu.:46.45   3rd Qu.:46.13
##  Max.   :58.23   Max.   :58.21   Max.   :58.15   Max.   :58.04
##  NA’s   :25      NA’s   :26      NA’s   :26      NA’s   :26
##      x1966           x1967           x1968           x1969
##  Min.   :12.70   Min.   :14.00   Min.   :13.70   Min.   :12.52
##  1st Qu.:25.28   1st Qu.:24.57   1st Qu.:23.72   1st Qu.:23.02
##  Median :40.38   Median :39.76   Median :39.09   Median :38.16
##  Mean   :36.25   Mean   :35.96   Mean   :35.62   Mean   :35.22
##  3rd Qu.:45.95   3rd Qu.:45.74   3rd Qu.:45.56   3rd Qu.:45.32
##  Max.   :57.87   Max.   :57.66   Max.   :57.43   Max.   :57.19
##  NA’s   :25      NA’s   :26      NA’s   :26      NA’s   :26
##      x1970           x1971           x1972           x1973
##  Min.   :11.57   Min.   :10.87   Min.   :10.34   Min.   : 9.943
##  1st Qu.:22.10   1st Qu.:22.54   1st Qu.:22.52   1st Qu.:21.971
##  Median :37.16   Median :36.64   Median :35.94   Median :35.267
##  Mean   :34.71   Mean   :34.50   Mean   :34.16   Mean   :33.722
##  3rd Qu.:45.27   3rd Qu.:45.08   3rd Qu.:44.92   3rd Qu.:44.911
##  Max.   :56.95   Max.   :56.73   Max.   :56.55   Max.   :56.409
##  NA’s   :22      NA’s   :24      NA’s   :23      NA’s   :22
##      x1974            x1975            x1976           x1977
##  Min.   : 9.701   Min.   : 9.715   Min.   :10.13   Min.   :10.30
##  1st Qu.:21.115   1st Qu.:20.888   1st Qu.:20.20   1st Qu.:20.11
##  Median :34.877   Median :34.493   Median :34.09   Median :33.76
##  Mean   :33.426   Mean   :33.058   Mean   :32.65   Mean   :32.41  
##  3rd Qu.:44.528   3rd Qu.:44.619   3rd Qu.:44.51   3rd Qu.:43.78  
##  Max.   :56.315   Max.   :56.274   Max.   :56.29   Max.   :56.35  
##  NA's   :22       NA's   :22       NA's   :20      NA's   :20     
##      x1978           x1979           x1980           x1981      
##  Min.   :10.40   Min.   :10.50   Min.   :11.10   Min.   :10.40  
##  1st Qu.:19.78   1st Qu.:19.75   1st Qu.:19.86   1st Qu.:20.66  
##  Median :33.52   Median :33.36   Median :33.18   Median :32.94  
##  Mean   :32.20   Mean   :32.08   Mean   :31.95   Mean   :31.95  
##  3rd Qu.:43.20   3rd Qu.:43.56   3rd Qu.:43.24   3rd Qu.:42.96  
##  Max.   :56.44   Max.   :56.54   Max.   :56.63   Max.   :56.68  
##  NA's   :20      NA's   :20      NA's   :20      NA's   :21     
##      x1982           x1983           x1984           x1985      
##  Min.   :10.30   Min.   : 9.90   Min.   :10.10   Min.   :10.20  
##  1st Qu.:20.48   1st Qu.:20.17   1st Qu.:19.90   1st Qu.:19.00  
##  Median :32.99   Median :32.22   Median :31.99   Median :31.32  
##  Mean   :31.74   Mean   :31.44   Mean   :31.12   Mean   :30.80  
##  3rd Qu.:42.81   3rd Qu.:42.33   3rd Qu.:41.89   3rd Qu.:41.11  
##  Max.   :56.69   Max.   :56.63   Max.   :56.52   Max.   :56.37  
##  NA's   :20      NA's   :20      NA's   :19      NA's   :19     
##      x1986           x1987           x1988           x1989      
##  Min.   : 9.80   Min.   : 9.70   Min.   :10.10   Min.   : 9.90  
##  1st Qu.:18.60   1st Qu.:19.14   1st Qu.:18.85   1st Qu.:18.35  
##  Median :30.58   Median :30.20   Median :29.72   Median :29.14  
##  Mean   :30.51   Mean   :30.18   Mean   :29.73   Mean   :29.19  
##  3rd Qu.:40.66   3rd Qu.:40.07   3rd Qu.:39.85   3rd Qu.:39.16  
##  Max.   :56.18   Max.   :55.98   Max.   :55.80   Max.   :55.63  
##  NA's   :19      NA's   :17      NA's   :18      NA's   :17     
##      x1990           x1991           x1992           x1993      
##  Min.   :10.00   Min.   : 9.90   Min.   : 9.80   Min.   : 9.40  
##  1st Qu.:18.56   1st Qu.:17.53   1st Qu.:17.22   1st Qu.:16.70  
##  Median :28.45   Median :27.71   Median :27.04   Median :26.10  
##  Mean   :28.89   Mean   :28.28   Mean   :27.78   Mean   :27.32  
##  3rd Qu.:38.60   3rd Qu.:38.09   3rd Qu.:37.50   3rd Qu.:37.16  
##  Max.   :55.48   Max.   :55.35   Max.   :55.22   Max.   :55.07  
##  NA's   :16      NA's   :15      NA's   :14      NA's   :17     
##      x1994           x1995           x1996           x1997      
##  Min.   : 9.40   Min.   : 8.60   Min.   : 8.10   Min.   : 7.70  
##  1st Qu.:16.00   1st Qu.:15.60   1st Qu.:15.15   1st Qu.:15.02  
##  Median :25.35   Median :24.88   Median :24.24   Median :23.68  
##  Mean   :26.72   Mean   :26.21   Mean   :25.70   Mean   :25.37  
##  3rd Qu.:35.89   3rd Qu.:34.92   3rd Qu.:34.33   3rd Qu.:34.01  
##  Max.   :54.91   Max.   :54.73   Max.   :54.53   Max.   :54.31  
##  NA's   :15      NA's   :15      NA's   :12      NA's   :15     
##      x1998           x1999           x2000           x2001      
##  Min.   : 7.60   Min.   : 7.80   Min.   : 7.80   Min.   : 7.20  
##  1st Qu.:14.34   1st Qu.:14.05   1st Qu.:14.04   1st Qu.:13.60  
##  Median :23.21   Median :22.71   Median :22.17   Median :21.70  
##  Mean   :24.92   Mean   :24.65   Mean   :24.25   Mean   :23.83  
##  3rd Qu.:33.26   3rd Qu.:32.77   3rd Qu.:32.07   3rd Qu.:31.96  
##  Max.   :54.08   Max.   :53.82   Max.   :53.54   Max.   :53.24  
##  NA's   :16      NA's   :16      NA's   :16      NA's   :15     
##      x2002           x2003           x2004           x2005       
##  Min.   : 7.10   Min.   : 6.90   Min.   : 7.20   Min.   : 7.812  
##  1st Qu.:13.44   1st Qu.:13.46   1st Qu.:13.22   1st Qu.:13.045
##  Median :21.20   Median :20.99   Median :20.79   Median :20.654
##  Mean   :23.47   Mean   :23.34   Mean   :23.12   Mean   :22.891
##  3rd Qu.:31.61   3rd Qu.:31.28   3rd Qu.:30.81   3rd Qu.:30.826
##  Max.   :52.91   Max.   :52.57   Max.   :52.20   Max.   :51.820
##  NA’s   :13      NA’s   :16      NA’s   :14      NA’s   :12
##      x2006            x2007           x2008           x2009
##  Min.   : 8.122   Min.   : 8.30   Min.   : 8.30   Min.   : 8.10
##  1st Qu.:13.123   1st Qu.:13.10   1st Qu.:12.96   1st Qu.:12.71
##  Median :20.694   Median :20.52   Median :20.22   Median :20.00
##  Mean   :22.692   Mean   :22.57   Mean   :22.47   Mean   :22.27
##  3rd Qu.:30.308   3rd Qu.:30.02   3rd Qu.:29.83   3rd Qu.:29.93
##  Max.   :51.428   Max.   :51.03   Max.   :50.62   Max.   :50.22
##  NA’s   :10       NA’s   :11      NA’s   :13      NA’s   :13
##      x2010           x2011           x2012           x2013
##  Min.   : 8.30   Min.   : 8.30   Min.   : 8.20   Min.   : 7.90
##  1st Qu.:12.71   1st Qu.:12.58   1st Qu.:12.60   1st Qu.:12.41
##  Median :19.61   Median :19.42   Median :18.89   Median :18.96
##  Mean   :22.01   Mean   :21.88   Mean   :21.59   Mean   :21.34
##  3rd Qu.:29.61   3rd Qu.:29.57   3rd Qu.:29.12   3rd Qu.:28.67
##  Max.   :49.80   Max.   :49.37   Max.   :48.93   Max.   :48.47
##  NA’s   :11      NA’s   :13      NA’s   :13      NA’s   :14
##      x2014           x2015           x2016           x2017
##  Min.   : 7.90   Min.   : 8.00   Min.   : 7.80   Min.   : 6.70
##  1st Qu.:12.48   1st Qu.:12.10   1st Qu.:11.91   1st Qu.:11.59
##  Median :18.83   Median :18.60   Median :18.32   Median :18.01
##  Mean   :21.06   Mean   :20.83   Mean   :20.53   Mean   :20.19
##  3rd Qu.:28.34   3rd Qu.:28.13   3rd Qu.:27.67   3rd Qu.:27.27
##  Max.   :47.99   Max.   :47.50   Max.   :47.02   Max.   :46.54
##  NA’s   :11      NA’s   :14      NA’s   :13      NA’s   :13
##      x2018
##  Min.   : 5.90
##  1st Qu.:11.42
##  Median :17.60
##  Mean   :19.83
##  3rd Qu.:27.09
##  Max.   :46.08
##  NA’s   :13
```

Reshaping

Ok, dataset is in wide format, hence I will reshape it into long format

```
df %>%
  pivot_longer(x1960:x2018,
               names_to = "year",
               values_to = "value")
```

```
## # A tibble: 15,576 x 4
##    country_name country_code year  value
##    <chr>        <chr>        <chr> <dbl>
##  1 Aruba        ABW          x1960  35.7
##  2 Aruba        ABW          x1961  34.5
##   3 Aruba        ABW          x1962  33.3
##   4 Aruba        ABW          x1963  32.0
##   5 Aruba        ABW          x1964  30.7
##   6 Aruba        ABW          x1965  29.4
##   7 Aruba        ABW          x1966  28.1
##   8 Aruba        ABW          x1967  26.9
##   9 Aruba        ABW          x1968  25.8
## 10 Aruba        ABW          x1969  24.9
## # ... with 15,566 more rows
```

Correct, hence I will store as a new dataframe Since I will not use the wide, I will rewrite it

```
df <- df %>%
  pivot_longer(x1960:x2018,
               names_to = "year",
               values_to = "value")
```

Now I will fix the year column from this:

```
head(df$year)
```

```
## [1] "x1960" "x1961" "x1962" "x1963" "x1964" "x1965"
```

```
df <- df %>%
  mutate(year = str_sub(year, 2)) # remove the x
```

Join

Now we need a row for continent. SInce I have a tree code country column, googled “three code country continent csv” and found a csv with the three code to match and the continent.

Found a file here: https://datahub.io/JohnSnowLabs/country-and-continent-codes-list

```
continents <- read_csv("https://datahub.io/JohnSnowLabs/country-and-continent-codes-list/r/country-and-
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Continent_Name = col_character(),
##   Continent_Code = col_character(),
##   Country_Name = col_character(),
##   Two_Letter_Country_Code = col_character(),
##   Three_Letter_Country_Code = col_character(),
##   Country_Number = col_double()
## )
```

So I will select only the relevant columns, Three_Letter_Country_Code and Continent_Name

```
continents <- continents %>%
  select(Three_Letter_Country_Code, Continent_Name)
```

Check

```r
head(continents)
```

```
## # A tibble: 6 x 2
##   Three_Letter_Country_Code Continent_Name
##   <chr>                     <chr>
## 1 AFG                       Asia
## 2 ALB                       Europe
## 3 ATA                       Antarctica
## 4 DZA                       Africa
## 5 ASM                       Oceania
## 6 AND                       Europe
```

Now try to join

```r
left_join(df, continents,
          by = c("country_code" = "Three_Letter_Country_Code"))
```

```
## # A tibble: 15,989 x 5
##    country_name country_code year  value Continent_Name
##    <chr>        <chr>        <chr> <dbl> <chr>
##  1 Aruba        ABW          1960   35.7 North America
##  2 Aruba        ABW          1961   34.5 North America
##  3 Aruba        ABW          1962   33.3 North America
##  4 Aruba        ABW          1963   32.0 North America
##  5 Aruba        ABW          1964   30.7 North America
##  6 Aruba        ABW          1965   29.4 North America
##  7 Aruba        ABW          1966   28.1 North America
##  8 Aruba        ABW          1967   26.9 North America
##  9 Aruba        ABW          1968   25.8 North America
## 10 Aruba        ABW          1969   24.9 North America
## # ... with 15,979 more rows
```

Works!, so let's join

```r
df  <- left_join(df, continents,
          by = c("country_code" = "Three_Letter_Country_Code"))
```

and delete the continents dataframe

```r
rm(continents)
```

Finally, change the year for date format

This was *tricky*, finally found the answer here: https://stackoverflow.com/questions/30255833/convert-four-digit-year-values-to-a-date-type

```r
df <- df %>%
  mutate(year = as.Date(as.character(year), format = "%Y"))
```

So, dataset ready for analysis!

Exploratory data analysis

```r
head(df)
```

```
## # A tibble: 6 x 5
##   country_name country_code year       value Continent_Name
##   <chr>        <chr>        <date>     <dbl> <chr>
## 1 Aruba        ABW          1960-12-09  35.7 North America
## 2 Aruba        ABW          1961-12-09  34.5 North America
## 3 Aruba        ABW          1962-12-09  33.3 North America
## 4 Aruba        ABW          1963-12-09  32.0 North America
## 5 Aruba        ABW          1964-12-09  30.7 North America
## 6 Aruba        ABW          1965-12-09  29.4 North America
```

How many countries?

```r
df %>%
  distinct(country_name) # check unique values in one specified column
```

```
## # A tibble: 264 x 1
##    country_name
##    <chr>
##  1 Aruba
##  2 Afghanistan
##  3 Angola
##  4 Albania
##  5 Andorra
##  6 Arab World
##  7 United Arab Emirates
##  8 Argentina
##  9 Armenia
## 10 American Samoa
## # ... with 254 more rows
```

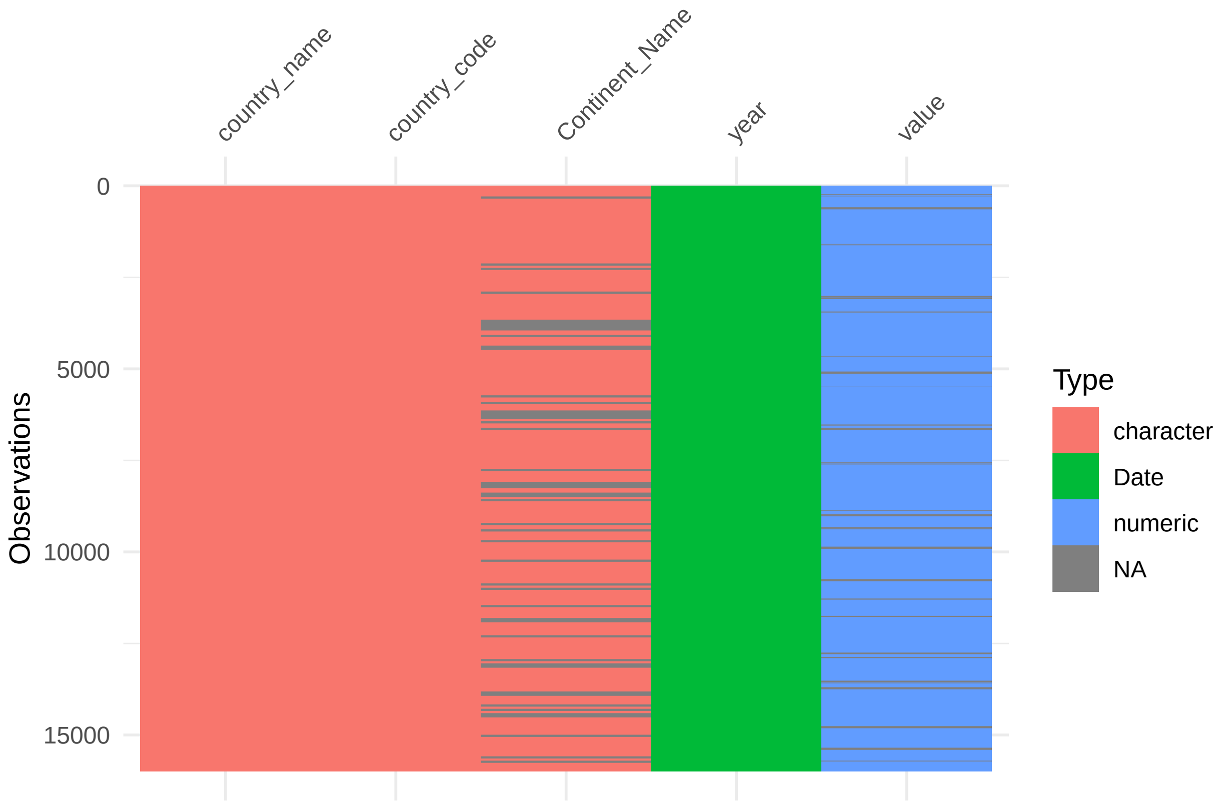
ok, we have 264 countries, from

```r
df %>%
  distinct(Continent_Name)
```

```
## # A tibble: 7 x 1
##   Continent_Name
##   <chr>
## 1 North America
## 2 Asia
## 3 Africa
## 4 Europe
## 5 <NA>
## 6 South America
## 7 Oceania
```

Check the NAs values

```
df %>%
  visdat::vis_dat() # visualize the variables and NAs from a dataset
```



Check in more detail the NAs from continents:

```
df %>%
  filter(is.na(Continent_Name)) %>%  # filter the NAs values from the Continent_name column
  distinct(country_name)
```

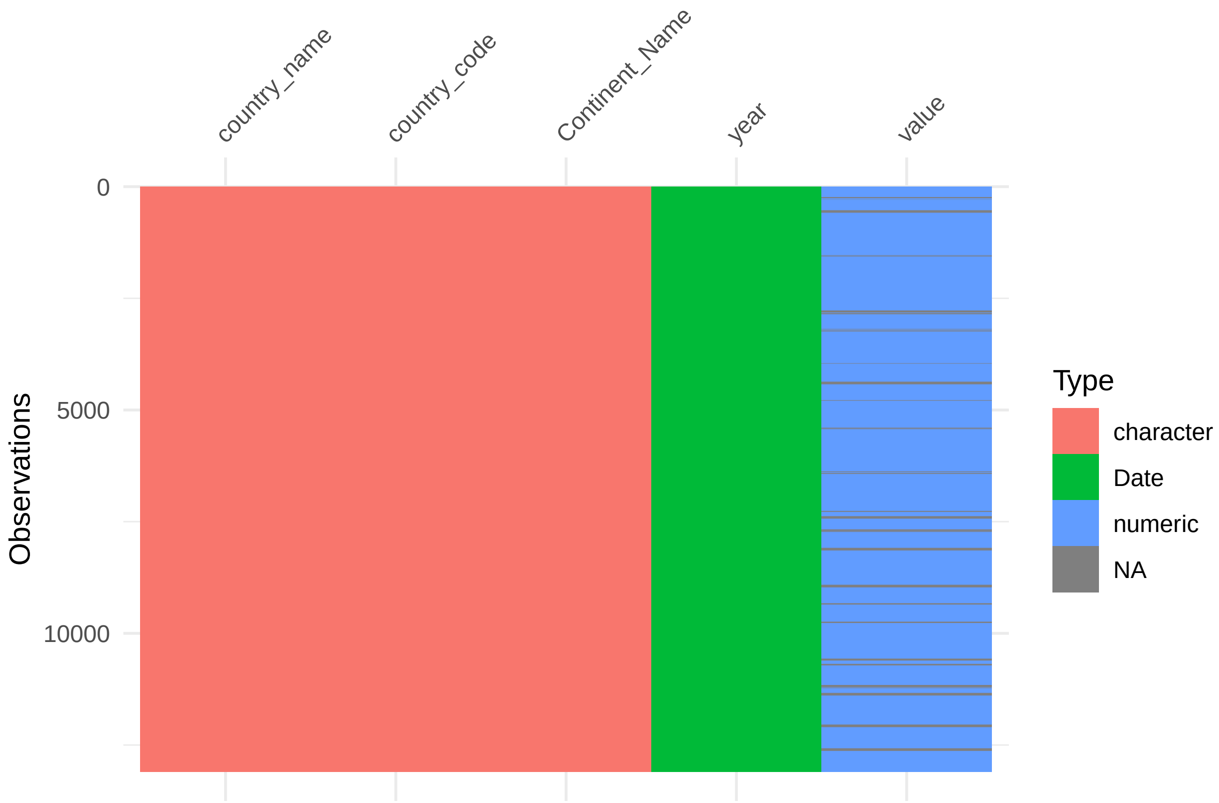
```
## # A tibble: 49 x 1
##    country_name
##    <chr>
##  1 Arab World
##  2 Central Europe and the Baltics
##  3 Channel Islands
##  4 Caribbean small states
##  5 East Asia & Pacific (excluding high income)
##  6 Early-demographic dividend
##  7 East Asia & Pacific
##  8 Europe & Central Asia (excluding high income)
##  9 Europe & Central Asia
## 10 Euro area
## # ... with 39 more rows
```

ok, there are some values, I will remove all of them and leave only the countries

```
df <- df %>%
  filter(!is.na(Continent_Name)) # here the ! makes the trick, means "Is not NA"
```

check again

```
df %>%
  visdat::vis_dat()
```



there are some NAs values, let's find them

```
df %>%
  filter(is.na(value))
```

```
## # A tibble: 981 x 5
##    country_name country_code year       value Continent_Name
##    <chr>        <chr>        <date>     <dbl> <chr>
##  1 Andorra      AND          1960-12-09    NA Europe
##  2 Andorra      AND          1961-12-09    NA Europe
##  3 Andorra      AND          1962-12-09    NA Europe
##  4 Andorra      AND          1963-12-09    NA Europe
##  5 Andorra      AND          1964-12-09    NA Europe
##  6 Andorra      AND          1965-12-09    NA Europe
##  7 Andorra      AND          1966-12-09    NA Europe
##  8 Andorra      AND          1967-12-09    NA Europe
##  9 Andorra      AND          1968-12-09    NA Europe
## 10 Andorra      AND          1969-12-09    NA Europe
## # ... with 971 more rows
```

ok, again, remove, now I will use drop_na

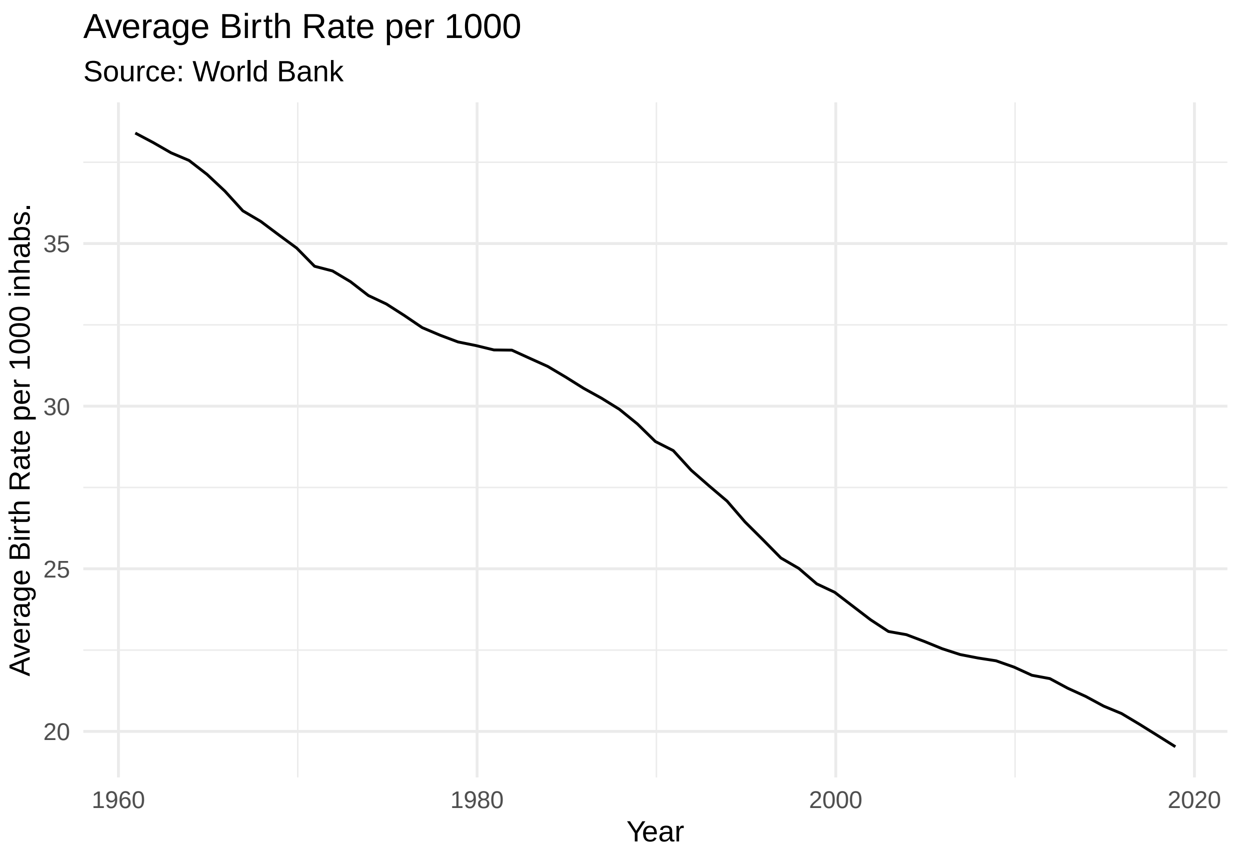
```
df <- df %>%
  drop_na(value)
```

What is the average birth rate per year?

Check the average change in the birth rate

```
df %>%
  group_by(year) %>%  # group by year
  summarise(average_birth_rate = mean(value)) %>% # now calculate the mean for each year
  ggplot(aes(x = year,
             y = average_birth_rate)) +
  geom_line(group = 1) +  # since there is only one point per year, I say here "use the points and merg
  labs(title = "Average Birth Rate per 1000",
       subtitle = "Source: World Bank",
       x = "Year",
       y = "Average Birth Rate per 1000 inhabs.") +
  theme_minimal()   # this use the theme minimal
```

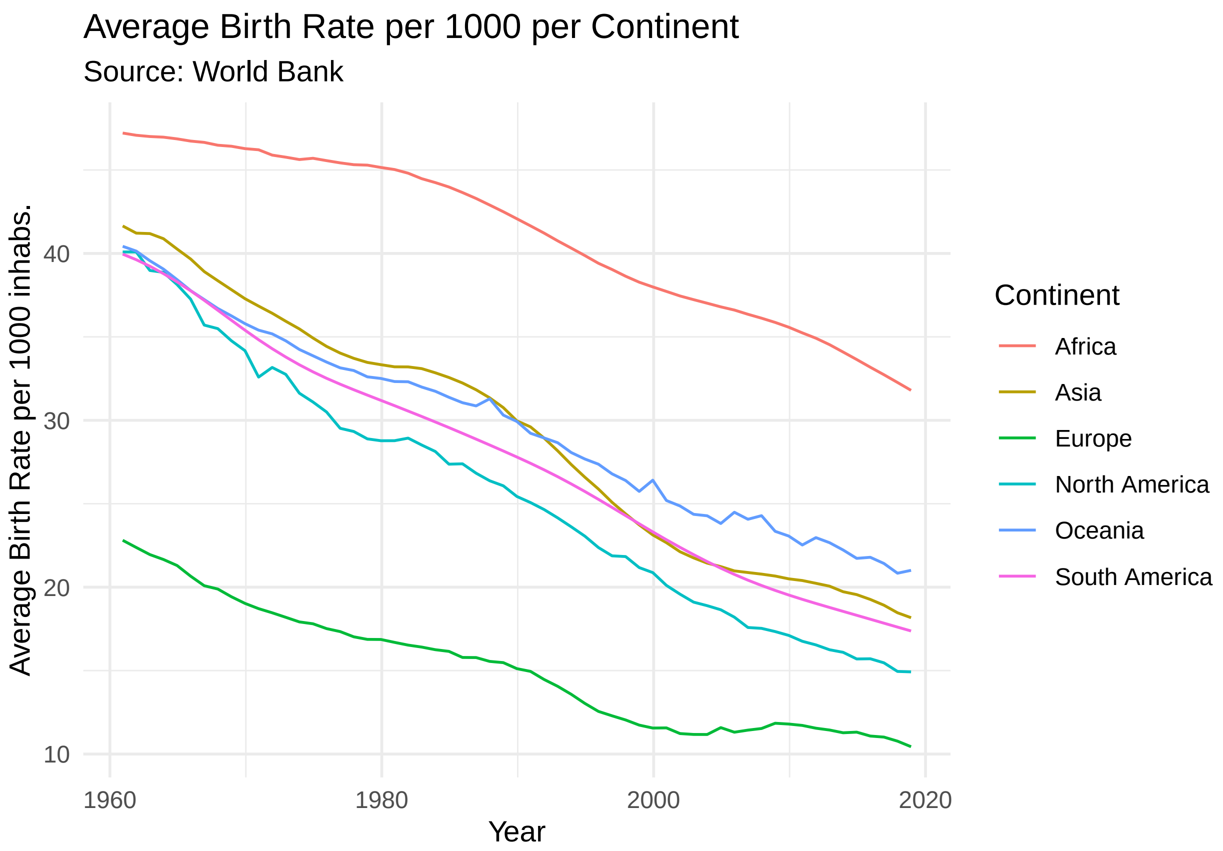
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



What is the average birth rate per year and continent?

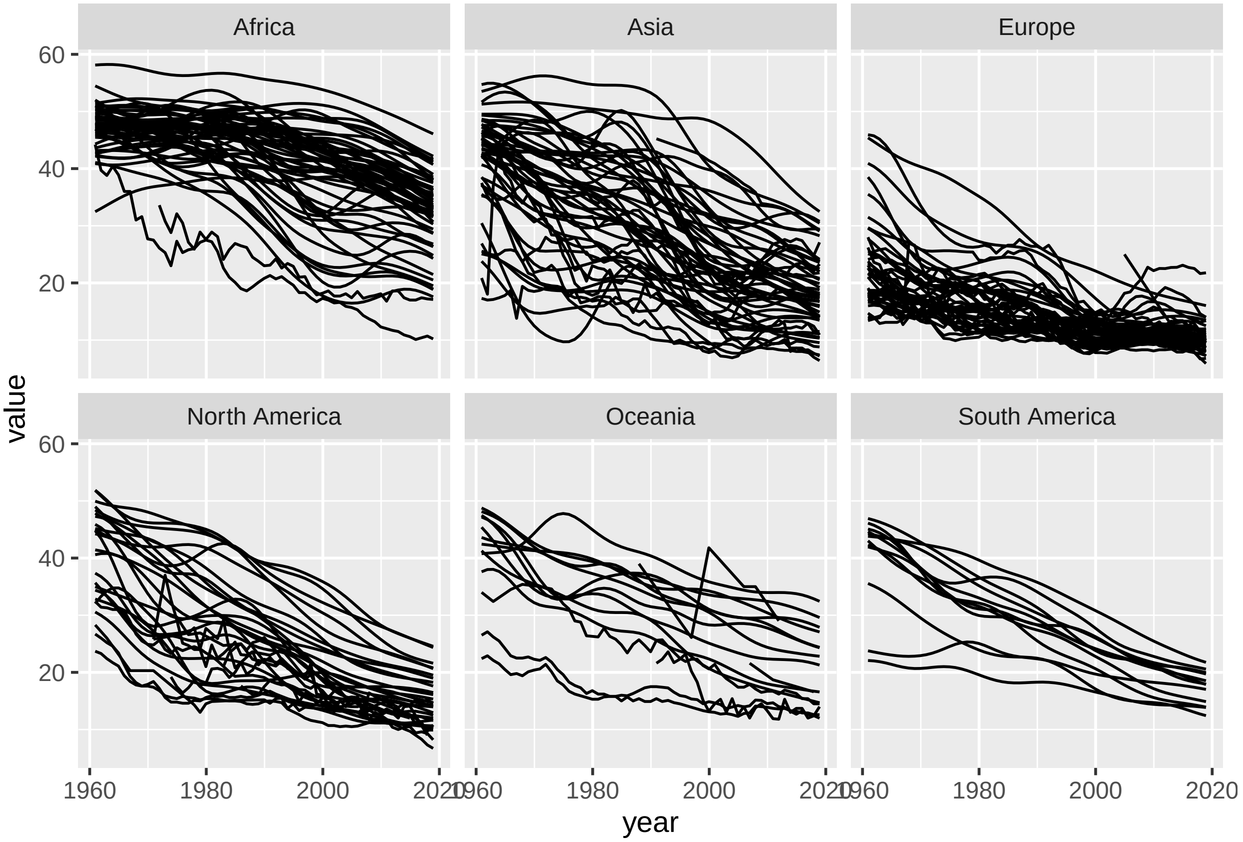
```
df %>%
  group_by(Continent_Name, year) %>%
  summarise(average_birth_rate = mean(value)) %>%
  ggplot(aes(x = year,
             y = average_birth_rate,
             color = Continent_Name)) +
  geom_line() +
  labs(title = "Average Birth Rate per 1000 per Continent",
       subtitle = "Source: World Bank",
       x = "Year",
       y = "Average Birth Rate per 1000 inhabs.",
       color = "Continent") +
  theme_minimal()    # this use the theme minimal
```

```
## `summarise()` regrouping output by 'Continent_Name' (override with `.groups` argument)
```



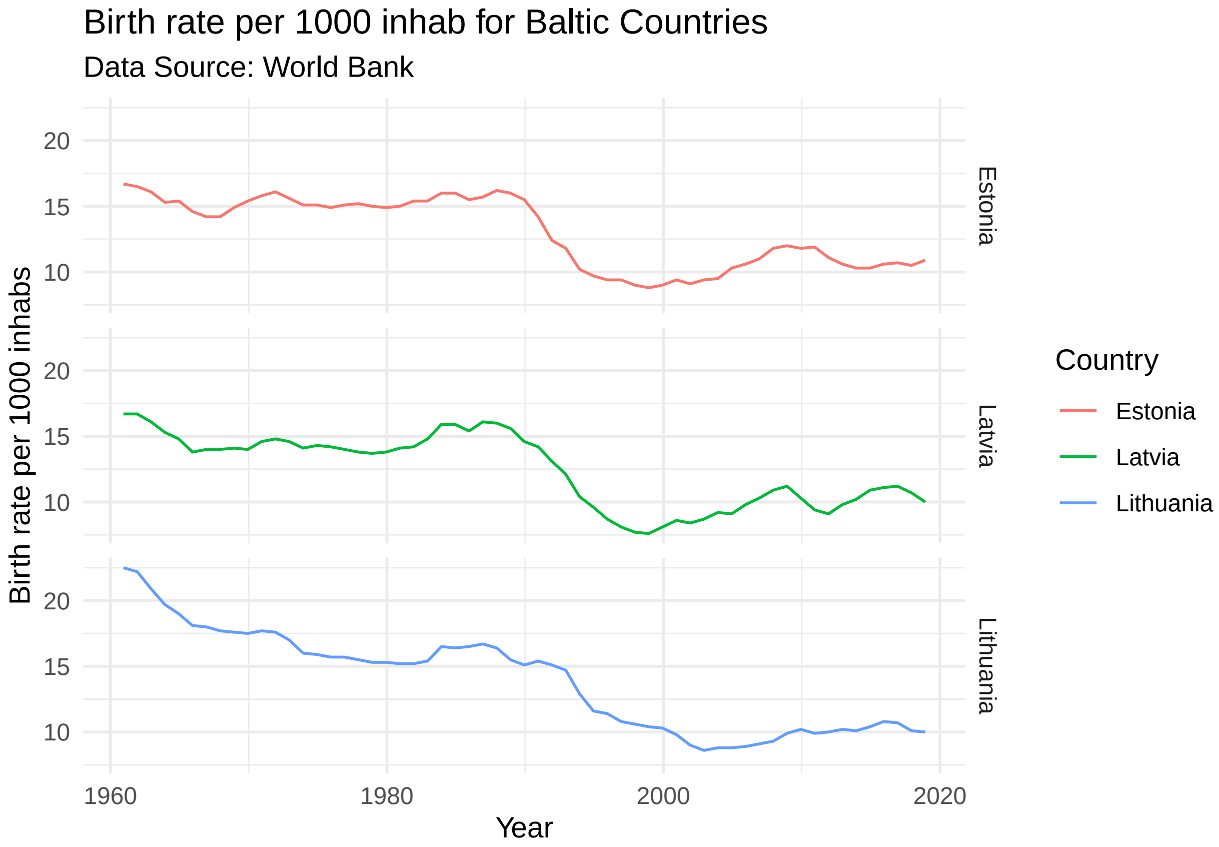
Check each country individually

```
df %>%
  ggplot(aes(x = year,
             y = value,
             group = country_name)) +
  geom_line() +
  facet_wrap(~Continent_Name)
```



### What is the average birth rate per year for the Baltic countries? We will focus on the Baltic countries

```
df %>%
  filter(country_name %in% c("Latvia", "Estonia", "Lithuania")) %>%  # Select only the three baltic cou
   ggplot(aes(x = year,
             y = value,
             group = country_name,
             color = country_name)) +
  geom_line() +
  facet_grid(country_name~.) + # order the facet with the countries in three rows. Try changing to (. ~
  theme_minimal() +
  labs(
    title = "Birth rate per 1000 inhab for Baltic Countries",
    subtitle = "Data Source: World Bank",
    x = "Year",
    y = "Birth rate per 1000 inhabs",
    color = "Country"
  )
```



Tables

I will calculate the change in birth rate from 1988 to 2018 for european countries and create a table.

I will use a new package, "DT". The documentation is here: https://rstudio.github.io/DT/

```r
pacman::p_load(DT)
```

```r
df %>%
  select(-country_code) %>%  #unselect this column, since is useless
  filter(Continent_Name == "Europe") %>%    # filter only european countries
  select(-Continent_Name) %>%  # and now unselect this useless column
  # convert the date from YYYYMMDD format to YYYY
  mutate(year = lubridate::year(year)) %>%
  filter(year == "1988" |
           year == "2018")  %>%
  # now convert to wide format to calculate the difference
  pivot_wider(names_from = year,
              values_from = value)  %>%
  # change the name of the columns
  rename( "x1988" = "1988",
          "x2018" = "2018") %>%
  # now calculate the difference
  mutate("Difference in birth rate per 1000 inhabs. 1988-2018" = x2018 - x1988) %>% # create a new vari
  # filter only rows with values
  drop_na() %>%
  # round numbers
  mutate_if(is.numeric, round, 1) %>%
  # create a nice table
  select(country_name, "Difference in birth rate per 1000 inhabs. 1988-2018") %>%
  # and now the table, copy and paste from the documentation
  datatable()
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, pleas
```

Create a codebook

Use the dataMaid package

uncomment the next line to generate a codebook

```
# dataMaid::makeCodebook(df)
```

This command create a codebook in PDF format.