Introduction to Cultural Data Science Exam Portfolio

Anja Meerwald (202008898@post.au.dk)

School of Communication and Cognition, University of Aarhus, Jens Chr. Skous Vej 2, 8000, Aarhus Denmark

Lecturer: Adéla Sobotkova
January 12, 2023

Anja Meerwald
202008898

Contents of the Portfolio

This portfolio collates the following documents:
• Assignments 1-6
• Final project report that includes a link to the digital product (i.e. GitHub)

For each assignment, the instructions are included. For assignment 2, a pdf of the created spreadsheet is also included. For assignment 3, the screenshot and html are included. For assignments 4 and 6, the html documents are included. All of these can also be found on Github.

To open these assignments, please use the link before for my repository where you will find each of the assignments by number:

https://github.com/Digital-Methods-HASS/au665920_Meerwald_Anja

Anja Meerwald
202008898

1:W35: Regular expressions and spreadsheets

DESCRIPTION

Upload your answers/solutions to the problems below. Beware of making the submission legible and understandable to another reader:

1. What regular expressions do you use to extract all the dates in this blurb: http://bit.ly/regexexercise2 and to put them into the following format YYYY-MM-DD ?

   (\d+).(\d+).\s?(\d+) ← to make each part of the date it's own variable

   $3 - $1 - $2 ← to put the variables in the desired order

2. Write a regular expression to convert the stopwordlist (list of most frequent Danish words) from Voyant in http://bit.ly/regexexercise3 into a neat stopword list for R (which comprises "words" separated by commas, such as http://bit.ly/regexexercise4). Then take the stopwordlist from R http://bit.ly/regexexercise4 and convert it into a Voyant list (words on separate line without interpunction)

   Converting stopwordlist to a neat stopword list for R

   Regular expression: (\S+)\n

   Substitution: "$1",

   Converting a stopword list for R into a Voyant list

   Regular expression: (", ")

   Substitution: \n

3. In 250 words, answer the following question: "What are the basic principles for using spreadsheets for good data organization?"

   Data validation, in order to control how and what data is entered. This includes drop down menus or only allowing certain numbers/range of numbers to be entered to minimize erroneous entries. Only having one value in each box, one entry per row and one variable per column. Remember that changes made cannot be tracked or undone easily so leave the raw data alone and work off another version of the data. Colors won't be transferred when exporting elsewhere so they can be used but without a legend or explanation it can be confusing later. Be consistent in labeling, if collecting a measurement, be sure to specify what is being measured, cm or mm for example.

4. Challenge (OPTIONAL)!Can you find all the instances of 'Dis Manibus' invocation in the EDH inscriptions in https://bit.ly/regexexercise5? Beware of the six possible canonical versions of the Dis Manibus formula (see day 1 slides)!

Anja Meerwald
202008898

Work in progress…

\b[Dd]i. [Mm]anibus

\b(di*s? manibus) (sacrum)?|dms?

(\bd m s?)|(\bdi*s? manibus)|(\bdi*s? manibus sacrum)

\bdi*s? ?m(anibus)? ?s?(acrum)?

Anja Meerwald
202008898

2:W35: Open Refine

DESCRIPTION

Upload your answers to these questions:

1. Create a spreadsheet listing the names of Danish monarchs with their birth- and death-date and start and end year of reign. Make it *tidy*! They should be sortable by year of birth. Suitable source websites are here and here, but you can also use another source, provided you reference it. (Group collaboration is expected and welcome. Remember to attach this spreadsheet to Brightspace submission)

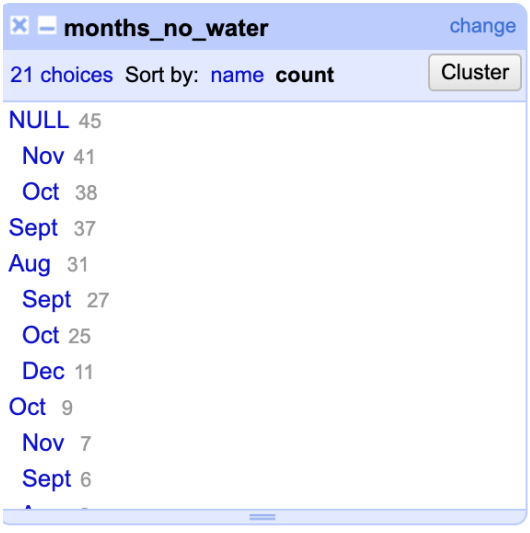
   kings_data

2. Does OpenRefine alter the raw data during sorting and filtering?

   No it does not, making it a good tool to use that will not cause issues with the original data.

3. Fix the interviews dataset in OpenRefine enough to answer this question: "Which two months are reported as the most water-deprived/driest by the interviewed farmer households?"
   a. Started with text transform for the column months_no_water
      i. Text transform on 86 cells in column months_no_water: grel:value.replace("[", "").replace("]","").replace("""","")
   b. Then saw the months are separated with a ";" so used that to split the cells with just one month in each cell
      i. Split multi-valued cells in column months_no_water
   c. Then facet by text and use the count function, Nov and Oct show as the two months most water deprived

      i.



Anja Meerwald
202008898

4. Real-Data-Challenge: What are the 10 most frequent occupations (erhverv) among unmarried men and women in 1801 Aarhus? (hint: some expert judgement interpretation is necessary, look at the HISCO classification "Historical International Standard of Classification of Occupations" on Dataverse if ambitious)

- Cluster editing the erhverv column to get rid of multiple spellings
- Use text facet the civilstand column and pick ugift

- Use text facet again for erhverv, sort my count and then can see the top 10

-

Anja Meerwald
20208898

Kings	birth_year	start_year	end_year	king_years	death_year
Jellingekonger					
Gorm den Gamle	NA	NA	958		
Harald (1.) Blåtand	NA	958	986	28	
Svend (1.) Tveskæg	NA	986	1014	28	
Harald (2.) Svensen	NA	1014	1018	4	
Knud (2.) den Store	995	1018	1035	17	
Hardeknud (Knud 3.)	1020	1035	1042	7	
Magnus (1.) den Gode	1024	1042	1047	5	
Svend (2.) Estridsen	NA	1047	1074	27	1076
Harald (3.) Hén	NA	1074	1080	6	
Knud (4.) den Hellige	NA	1080	1086	6	
Oluf (1.) Hunger	NA	1086	1095	9	
Erik (1.) Ejegod	1056	1095	1103	8	
Niels	NA	1104	1134	30	
Erik (2.) Emune	NA	1134	1137	3	
Erik (3.) Lam	NA	1137	1146	9	
Svend (3.) Grathe	NA	1146	1157	11	
Knud (5.) Magnussen	NA	1146	1157	11	
Valdemar (1.) den Store	1131	1157	1182	25	
Knud 6.	1163	1182	1202	20	
Valdemar (2.) Sejr	1170	1202	1241	39	
Erik (4.) Plovpenning	1216	1241	1250	9	
Abel	1218	1250	1252	2	
Christoffer 1.	1219	1252	1259	7	
Erik (5.) Klipping	1249	1259	1286	27	
Erik (6.) Menved	1274	1286	1319	33	
Christoffer 2.	1276	1320	1326	6	
Valdemar (3.) Eriksen	1314	1326	1330	4	
Christoffer 2.		1330	1332	3	
Valdemar (4.) Atterdag	1320	1340	1375	35	
Oluf 2.	1370	1376	1387	11	
Margrete 1.	1353	1387	1412	25	
Erik (7.) af Pommern	1382	1412	1439	27	1459
Christoffer 3. af Bayern	1416	1440	1448	8	
Christian 1.	1426	1448	1481	33	
Hans	1455	1481	1513	32	
Christian 2.	1481	1513	1523	10	1559
Frederik 1.	1471	1523	1533	10	

Anja Meerwald

Christian 3.	1503		1534		1559		25
Frederik 2.	1534		1559		1588		29
Christian 4.	1577		1588		1648		60
Frederik 3.	1609		1648		1670		22
Christian 5.	1646		1670		1699		29
Frederik 4.	1671		1699		1730		31
Christian 6.	1699		1730		1746		16
Frederik 5.	1723		1746		1766		20
Christian 7.	1749		1766		1808		42
Frederik 6.	1768		1808		1839		31
Christian 8.	1786		1839		1848		9
Frederik 7.	1808		1848		1863		15
Christian 9.	1818		1863		1906		43
Frederik 8.	1843		1906		1912		6
Christian 10.	1870		1912		1947		35
Frederik 9.	1899		1947		1972		25
Margrethe 2.	1940		1972	NA		NA	
Kronprins Frederik	1968	NA		NA		NA	

Anja Meerwald
202008898

3:W35: Start with R

DESCRIPTION

Instructions: For this assignment, you need to answer a couple questions with code and then take a screenshot of your working environment.
Submit the solutions including the URL to the screenshot in a doc/pdf to Brightspace.

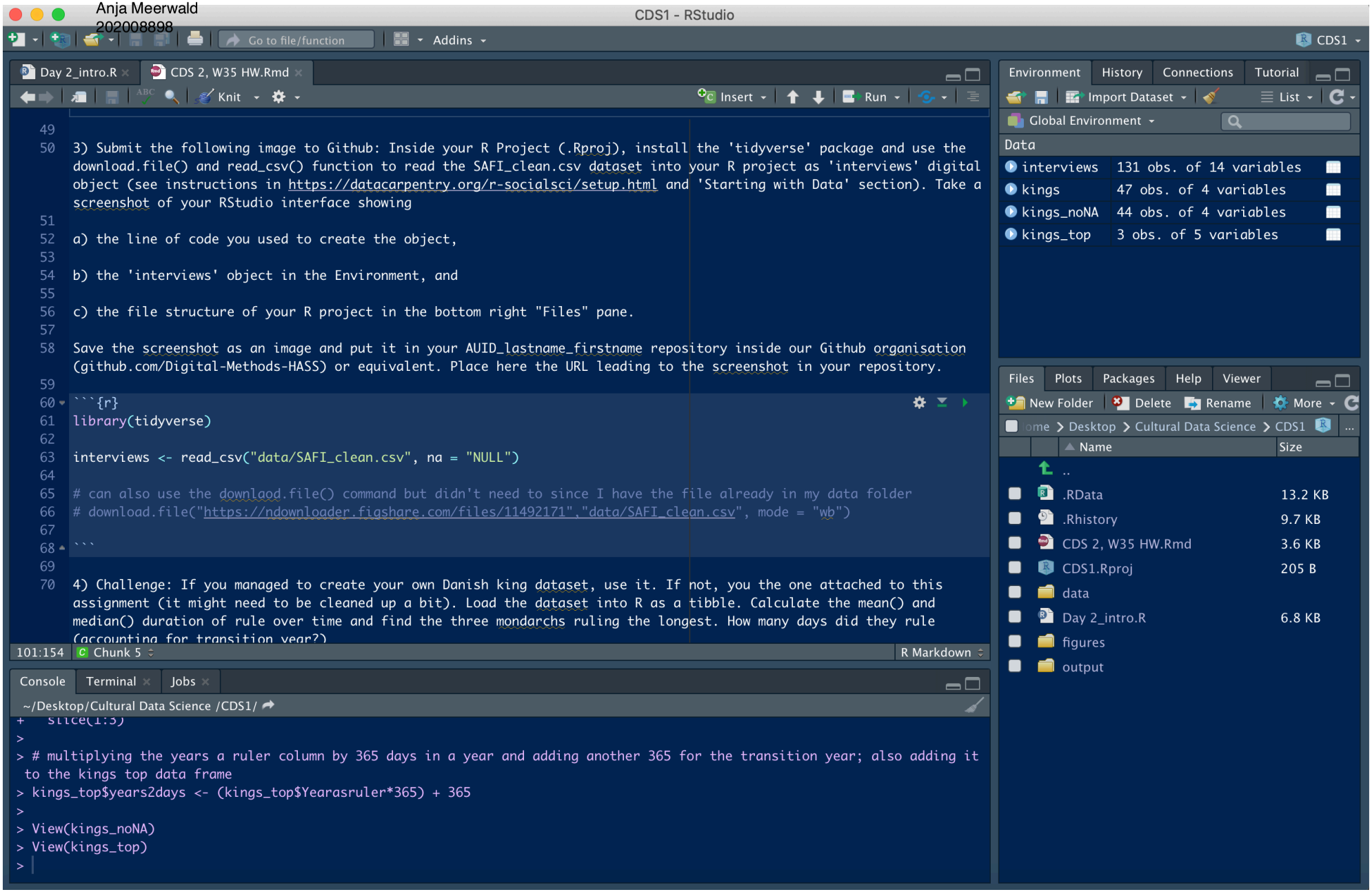
1) Use R to figure out how many elements in the vector below are **greater than 2** and then tell me what their **sum** (of the larger than 2 elements) is.
rooms <- c(1, 2, 4, 5, 1, 3, 1, NA, 3, 1, 3, 2, 1, NA, 1, 8, 3, 1, 4, NA, 1, 3, 1, 2, 1, 7, 1, 9, 3, NA)

2) What **type** of data is in the 'rooms' vector?

3) Submit the following image to Github: Inside your R Project (.Rproj), install the 'tidyverse' package and use the download.file() and read_csv() function to read the SAFI_clean.csv dataset into your R project as 'interviews' digital object (see instructions in https://datacarpentry.org/r-socialsci/setup.html and 'Starting with Data' section). Take a screenshot of your RStudio interface showing
a) the line of code you used to create the object,
b) the 'interviews' object in the Environment, and
c) the file structure of your **R project** in the bottom right "Files" pane.
Save the screenshot as an image and put it in your **AUID_lastname_firstname** repository inside our Github organisation (github.com/Digital-Methods-HASS) or equivalent. Place **here** the URL leading to the screenshot in your repository.

https://raw.githubusercontent.com/Digital-Methods-HASS/au665920_Meerwald_Anja/main/Screen%20Shot%203_W35_Start%20with%20R.png

4) Challenge: If you managed to create your own Danish king dataset, use it. If not, you the one attached to this assignment (it might need to be cleaned up a bit). Load the dataset into R as a tibble. Calculate the mean() and median() duration of rule over time and find the three mondarchs ruling the longest. How many days did they rule (accounting for transition year?)


Anja Meerwald
202008898
CDS1 - RStudio
Go to file/function
Addins
CDS1
Day 2_intro.R
CDS 2, W35 HW.Rmd
Knit
Insert
Run
49
50 3) Submit the following image to Github: Inside your R Project (.Rproj), install the 'tidyverse' package and use the download.file() and read_csv() function to read the SAFI_clean.csv dataset into your R project as 'interviews' digital object (see instructions in https://datacarpentry.org/r-socialsci/setup.html and 'Starting with Data' section). Take a screenshot of your RStudio interface showing
51
52 a) the line of code you used to create the object,
53
54 b) the 'interviews' object in the Environment, and
55
56 c) the file structure of your R project in the bottom right "Files" pane.
57
58 Save the screenshot as an image and put it in your AUID_lastname_firstname repository inside our Github organisation (github.com/Digital-Methods-HASS) or equivalent. Place here the URL leading to the screenshot in your repository.
59
60 ```{r}
61 library(tidyverse)
62
63 interviews <- read_csv("data/SAFI_clean.csv", na = "NULL")
64
65 # can also use the downlaod.file() command but didn't need to since I have the file already in my data folder
66 # download.file("https://ndownloader.figshare.com/files/11492171","data/SAFI_clean.csv", mode = "wb")
67
68 ```
69
70 4) Challenge: If you managed to create your own Danish king dataset, use it. If not, you the one attached to this assignment (it might need to be cleaned up a bit). Load the dataset into R as a tibble. Calculate the mean() and median() duration of rule over time and find the three mondarchs ruling the longest. How many days did they rule (accounting for transition year?)
101:154
Chunk 5
R Markdown
Console
Terminal
Jobs
~/Desktop/Cultural Data Science /CDS1/
+ slice(1:3)
>
> # multiplying the years a ruler column by 365 days in a year and adding another 365 for the transition year; also adding it to the kings top data frame
> kings_top$years2days <- (kings_top$Yearasruler*365) + 365
>
> View(kings_noNA)
> View(kings_top)
>
Environment
History
Connections
Tutorial
Import Dataset
List
Global Environment
Data
interviews 131 obs. of 14 variables
kings 47 obs. of 4 variables
kings_noNA 44 obs. of 4 variables
kings_top 3 obs. of 5 variables
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Desktop > Cultural Data Science > CDS1
Name
Size
..
.RData 13.2 KB
.Rhistory 9.7 KB
CDS 2, W35 HW.Rmd 3.6 KB
CDS1.Rproj 205 B
data
Day 2_intro.R 6.8 KB
figures
output

3: W35: Start with R

Anja Meerwald
2020-08-08

Anja Meerwald

2:W35: Start with R DESCRIPTION

Instructions: For this assignment, you need to answer a couple questions with code and then take a screenshot of your working environment.

Submit the solutions including the URL to the screenshot in a doc/pdf to Brightspace.

1. Use R to figure out how many elements in the vector below are greater than 2 and then tell me what their sum (of the larger than 2 elements) is.

```
rooms <- c(1, 2, 4, 5, 1, 3, 1, NA, 3, 1, 3, 2, 1, NA, 1, 8, 3, 1, 4, NA, 1, 3, 1, 2, 1, 7, 1, 9, 3, NA)
```

```
# showing the amount of elements greater than 2 in the vector, including NAs
rooms[rooms>2]
```

```
##  [1]  4  5  3 NA  3  3 NA  8  3  4 NA  3  7  9  3 NA
```

```
# calculating the total number of rooms greater than 2 (=12)
sum(rooms>2, na.rm = TRUE)
```

```
## [1] 12
```

```
# removing the NAs
rooms_omitNA <- na.omit(rooms)

# creating a new vector with only rooms 3 or more
big_rooms <- rooms_omitNA[!(rooms_omitNA < 3)]

# the total of those amount of rooms, 55
sum(big_rooms)
```

```
## [1] 55
```

2. What type of data is in the ‘rooms’ vector?

```
class(rooms)
```

```
## [1] "numeric"
```

3. Submit the following image to Github: Inside your R Project (.Rproj), install the ‘tidyverse’ package and use the download.file() and read_csv() function to read the SAFI_clean.csv dataset into your R project as

'interviews' digital object (see instructions in https://datacarpentry.org/r-socialsci/setup.html (https://datacarpentry.org/r-socialsci/setup.html) and 'Starting with Data' section). Take a screenshot of your RStudio interface showing

Anja Meerwald 202008898

a. the line of code you used to create the object,

b. the 'interviews' object in the Environment, and

c. the file structure of your R project in the bottom right "Files" pane.

Save the screenshot as an image and put it in your AUID_lastname_firstname repository inside our Github organisation (github.com/Digital-Methods-HASS) or equivalent. Place here the URL leading to the screenshot in your repository.

file:///Users/anjameerwald/Desktop/Screen%20Shot%202022-09-04%20at%204.08.24%20PM.png (file:///Users/anjameerwald/Desktop/Screen%20Shot%202022-09-04%20at%204.08.24%20PM.png)

```
library(tidyverse)

interviews <- read_csv("data/SAFI_clean.csv", na = "NULL")
```

```
## Rows: 131 Columns: 14
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): village, respondent_wall_type, memb_assoc, affect_conflicts, items...
## dbl  (6): key_ID, no_membrs, years_liv, rooms, liv_count, no_meals
## dttm (1): interview_date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# can also use the downlaod.file() command but didn't need to since I have the file alre
ady in my data folder
# download.file("https://ndownloader.figshare.com/files/11492171","data/SAFI_clean.csv",
mode = "wb")
```

4. Challenge: If you managed to create your own Danish king dataset, use it. If not, you the one attached to this assignment (it might need to be cleaned up a bit). Load the dataset into R as a tibble. Calculate the mean() and median() duration of rule over time and find the three mondarchs ruling the longest. How many days did they rule (accounting for transition year?)

```
kings <- read_csv2("data/kings.csv")
```

```
## ℹ Using "',' as decimal and "'.'" as grouping mark. Use `read_delim()` for more cont
rol.
```

```
## Rows: 47 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ";"
## chr (2): Kings, Yearasruler
## dbl (2): Start_date, End_date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# removing NAs and putting it into a new df
kings_noNA <- na.omit(kings)

# getting a overview of what the df looks like
str(kings_noNA)
```

```
## tibble [44 × 4] (S3: tbl_df/tbl/data.frame)
##  $ Kings      : chr [1:44] "Harald 2." "Knud 1. den Store" "Hardeknud" "Magnus den Go
de" ...
##  $ Start_date : num [1:44] 1014 1018 1035 1042 1047 ...
##  $ End_date   : num [1:44] 1018 1035 1042 1047 1074 ...
##  $ Yearasruler: chr [1:44] "4" "17" "7" "5" ...
##  - attr(*, "na.action")= 'omit' Named int [1:3] 1 2 3
##   ..- attr(*, "names")= chr [1:3] "1" "2" "3"
```

```
# making the yearsaruler variable numeric
kings_noNA$Yearasruler <- as.numeric(kings_noNA$Yearasruler)

str(kings_noNA)
```

```
## tibble [44 × 4] (S3: tbl_df/tbl/data.frame)
##  $ Kings      : chr [1:44] "Harald 2." "Knud 1. den Store" "Hardeknud" "Magnus den Go
de" ...
##  $ Start_date : num [1:44] 1014 1018 1035 1042 1047 ...
##  $ End_date   : num [1:44] 1018 1035 1042 1047 1074 ...
##  $ Yearasruler: num [1:44] 4 17 7 5 27 6 6 9 8 30 ...
##  - attr(*, "na.action")= 'omit' Named int [1:3] 1 2 3
##   ..- attr(*, "names")= chr [1:3] "1" "2" "3"
```

```
# mean of the years a ruler variable
mean(kings_noNA$Yearasruler)
```

```
## [1] 18.68182
```

```
# median of the years a ruler variable
median(kings_noNA$Yearasruler)
```

```
## [1] 14
```

Anja Meerwald
202008898

```
# can see the kings in order from most years to least years
kings_noNA %>%
  arrange(desc(Yearasruler))
```

```
## # A tibble: 44 × 4
##    Kings                Start_date End_date Yearasruler
##    <chr>                     <dbl>    <dbl>       <dbl>
##  1 Christian 4.               1588     1648          60
##  2 Erik 7. af Pommern         1396     1439          43
##  3 Christian 7.               1766     1808          42
##  4 Valdemar 2. Sejr           1202     1241          39
##  5 Erik 6. Menved             1286     1319          35
##  6 Valdemar 4. Atterdag       1340     1375          35
##  7 Chrstian 1.                1448     1481          33
##  8 Hans                       1482     1513          31
##  9 Frederik 4.                1699     1730          31
## 10 Frederik 6.                1808     1839          31
## # … with 34 more rows
```

```
# working with just the top three rulers in terms of time and putting them into their ow
n data frame
kings_top <- kings_noNA %>%
  arrange(desc(Yearasruler)) %>%
  slice(1:3)
kings_top
```

```
## # A tibble: 3 × 4
##   Kings              Start_date End_date Yearasruler
##   <chr>                   <dbl>    <dbl>       <dbl>
## 1 Christian 4.             1588     1648          60
## 2 Erik 7. af Pommern       1396     1439          43
## 3 Christian 7.             1766     1808          42
```

```
# multiplying the years a ruler column by 365 days in a year and adding another 365 for
the transition year; also adding it to the kings top data frame
kings_top$years2days <- (kings_top$Yearasruler*365) + 365
kings_top
```

```
## # A tibble: 3 × 5
##   Kings              Start_date End_date Yearasruler years2days
##   <chr>                   <dbl>    <dbl>       <dbl>      <dbl>
## 1 Christian 4.             1588     1648          60      22265
## 2 Erik 7. af Pommern       1396     1439          43      16060
## 3 Christian 7.             1766     1808          42      15695
```

Anja Meerwald
202008898

4:W35: Visualize data (not only) with ggplot

DESCRIPTION

For this Visualization assignment, you engage in **one of the two** tasks and submit a rmarkdown and html document (ie, a knitted result) that collects the results of both tasks:

Choose **one** of the two options below and follow the instructions in rmarkdowns. These have slightly different content depending on what you wish to practice, whether facets in ggplot or animation. For the latter, pay attention to the prerequisites R specifies for your system:

1) Historical homicide trends across Western Europe (ggplot practice)
https://github.com/Digital-Methods-HASS/HomicideHistory

OR

2) Global development since 1957 (learn how to create animations with *gganimate* package! :)
https://github.com/Digital-Methods-HASS/GlobalDevelopment

How to complete this assignment: I am asking you to work with assignments from Github and submit answers to Github. You can do so manually following step in Manual guide, or on the command-line(CLI)

Manual guide to task B:

1. Download the homework repository (click green Code button > download ZIP)
2. Look at the .html file to see the tasks
3. Edit the .Rmd file to complete the tasks (ie. create additional code chunks and comments in the text sections) and then press the Knit button to render the results as a new .html file the same way you did in Day 01 UN Votes exercise
4. Upload these via 'Upload File' button to your au##### folder in Digital-Methods-HASS organization.

CLI guide (only if you use command-line):

1. clone **your** au##### repository from Github Digital Methods HASS so you have it available locally
2. **clone** one of the homework repositories next to your your **local** au##### folder
3. complete the tasks and knit the result
4. once happy with the result, **copy** the finished .Rmd and .html files from the HomicideHistory/GlobalDevelopment folder into your **local** au##### folder
5. **add** and **commit** the changes in your au##### folder
6. **push** the knitted result to **your au#####** repository in Github

Anja Meerwald
202008898

Submit here a *link* to **your au#####** repository in https://github.com/Digital-Methods-HASS ... which leads directly to the place where you have posted your solution as **both .Rmd** and **.html** files.

Note: if you are unsure that you uploaded the .html to Github correctly, use https://htmlpreview.github.io/ to preview it!

HW4 - Make Data Move

Anja Meerwald
10005804

Anja Meerwald

05/10/2020

Explore global development with R

Today, you will load a filtered gapminder dataset - with a subset of data on global development from 1952 - 2007 in increments of 5 years - to capture the period between the Second World War and the Global Financial Crisis.

Your task: Explore the data and visualise it in both static and animated ways, providing answers and solutions to 7 questions/tasks below.

Get the necessary packages

First, start with installing the relevant packages 'tidyverse', 'gganimate', and 'gapminder'.

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✔ ggplot2 3.3.5     ✔ purrr   0.3.4
## ✔ tibble  3.1.6     ✔ dplyr   1.0.8
## ✔ tidyr   1.2.0     ✔ stringr 1.4.1
## ✔ readr   2.1.2     ✔ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Look at the data and tackle the tasks

First, see which specific years are actually represented in the dataset and what variables are being recorded for each country. Note that when you run the cell below, Rmarkdown will give you two results - one for each line - that you can flip between.

```
str(gapminder)
```

```
## tibble [1,704 × 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##  $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 128818
16 13867957 16317921 22227415 ...
##  $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

```
unique(gapminder$year)
```

Anja Meerwald
202008898

```
##  [1] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
```

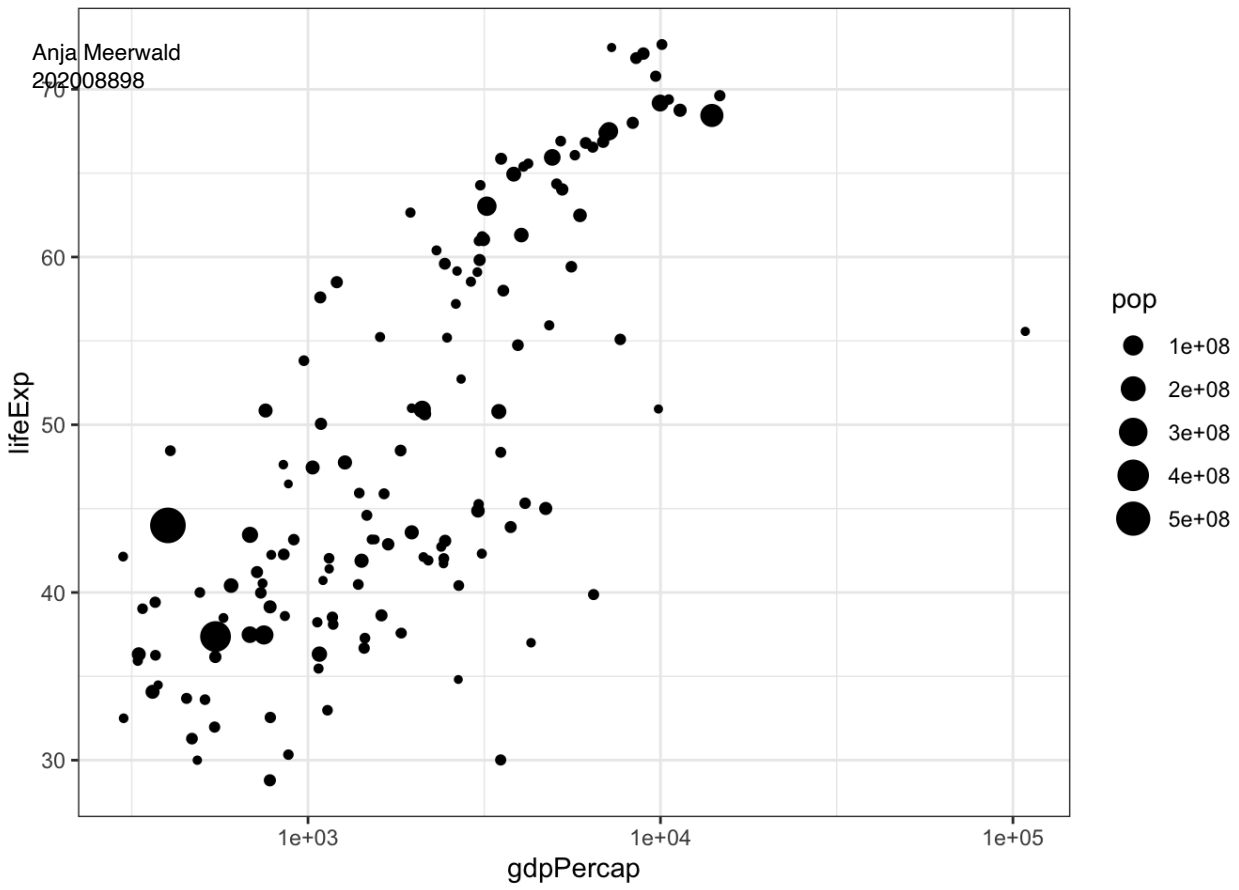
```
head(gapminder)
```

```
## # A tibble: 6 × 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

The dataset contains information on each country in the sampled year, its continent, life expectancy, population, and GDP per capita.

Let's plot all the countries in 1952.

```
theme_set(theme_bw())  # set theme to white background for better visibility

ggplot(subset(gapminder, year == 1952), aes(gdpPercap, lifeExp, size = pop)) +
  geom_point() +
  scale_x_log10()
```



...

We see an interesting spread with an outlier to the right. Answer the following questions, please:

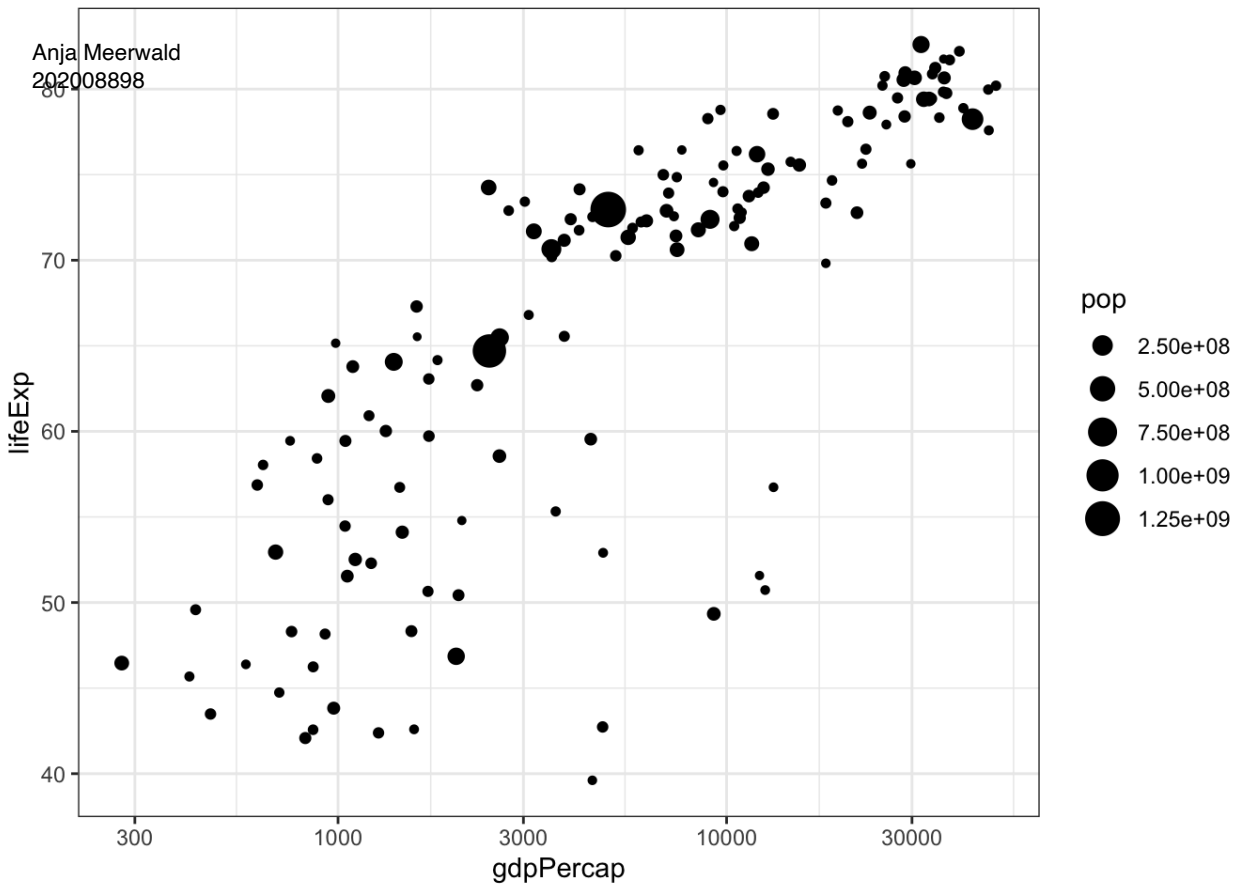
1. *Why does it make sense to have a log10 scale on x axis?* To make the data less skewed, which makes it easier to read.
2. *Who is the outlier (the richest country in 1952 - far right on x axis)?* Kuwait

```
gapminder %>%
  subset(year == "1952") %>% # gives you the year
  slice_max(gdpPercap) # gets the highest gdp
```

```
## # A tibble: 1 × 6
##   country continent  year lifeExp    pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>  <int>     <dbl>
## 1 Kuwait  Asia       1952    55.6 160000   108382.
```

Next, you can generate a similar plot for 2007 and compare the differences

```
ggplot(subset(gapminder, year == 2007), aes(gdpPercap, lifeExp, size = pop)) +
  geom_point() +
  scale_x_log10()
```



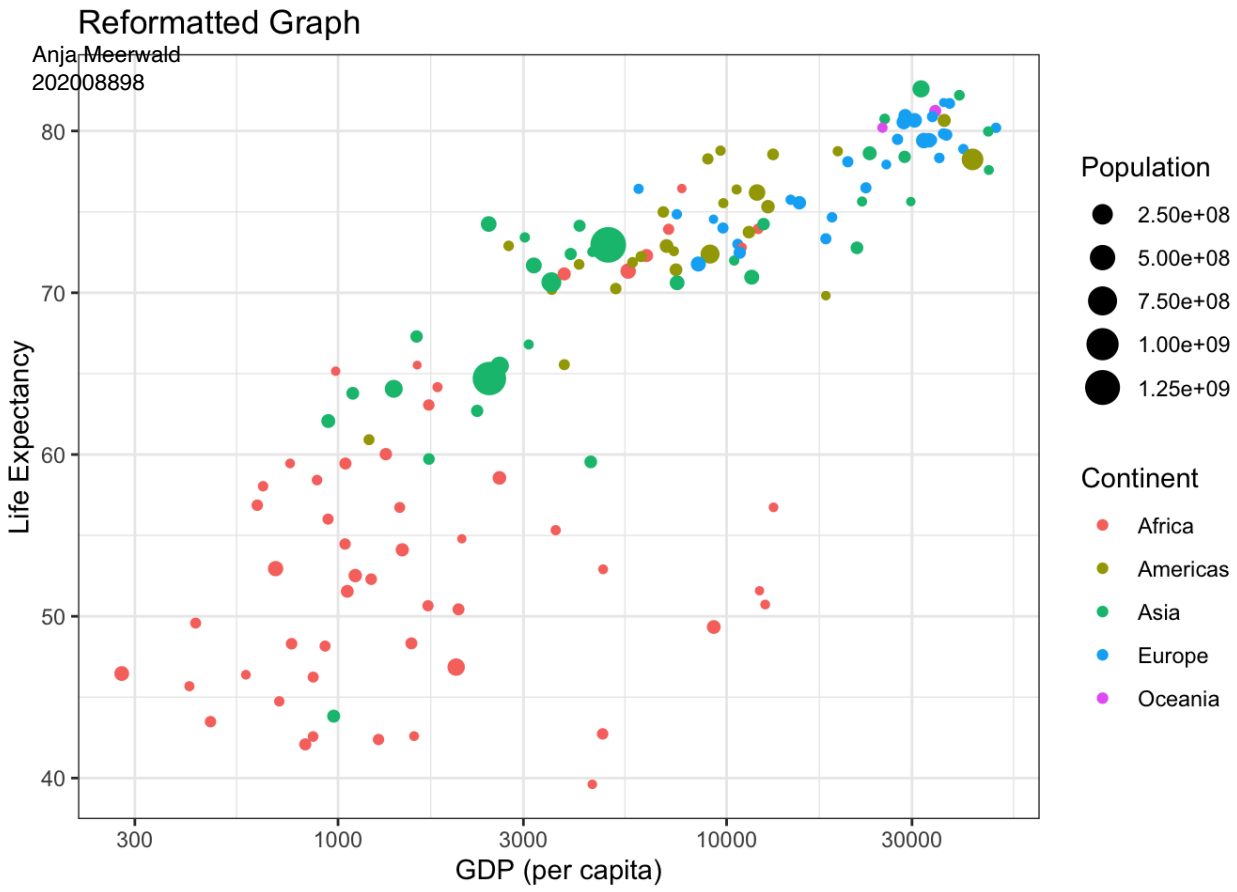
…

The black bubbles are a bit hard to read, the comparison would be easier with a bit more visual differentiation.

Tasks:

3. *Differentiate the **continents** by color, and fix the axis labels and units to be more legible (**Hint**: the 2.50e+08 is so called "scientific notation", which you might want to eliminate)*

```
ggplot(subset(gapminder, year == 2007), aes(gdpPercap, lifeExp, size = pop, color = cont
inent)) +
  geom_point() +
  labs(x = "GDP (per capita)", y = "Life Expectancy", title = "Reformatted Graph") + # t
his gives labels to the axes and a title
  guides(size = guide_legend(title = "Population"), # changes the titles of the legends
         color = guide_legend(title = "Continent")) +
  scale_x_log10()         # Modify formatting of axis
```




```
options(scipen=999) # This is used to get rid of the scientific notification
```

4. *What are the five richest countries in the world in 2007?*

```
gapminder %>%
  subset(year == "2007") %>% # gets only 2007 data
  arrange(desc(gdpPercap)) %>% # gives you the gdp from highest to lowest
  slice(1:5) # gets the top 5
```

```
## # A tibble: 5 × 6
##   country       continent  year lifeExp       pop gdpPercap
##   <fct>         <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Norway        Europe     2007    80.2   4627926    49357.
## 2 Kuwait        Asia       2007    77.6   2505559    47307.
## 3 Singapore     Asia       2007    80.0   4553009    47143.
## 4 United States Americas   2007    78.2 301139947    42952.
## 5 Ireland       Europe     2007    78.9   4109086    40676.
```

Make it move!

The comparison would be easier if we had the two graphs together, animated. We have a lovely tool in R to do this: the `gganimate` package. Beware that there may be other packages your operating system needs in order to glue interim images into an animation or video. Read the messages when installing the package.

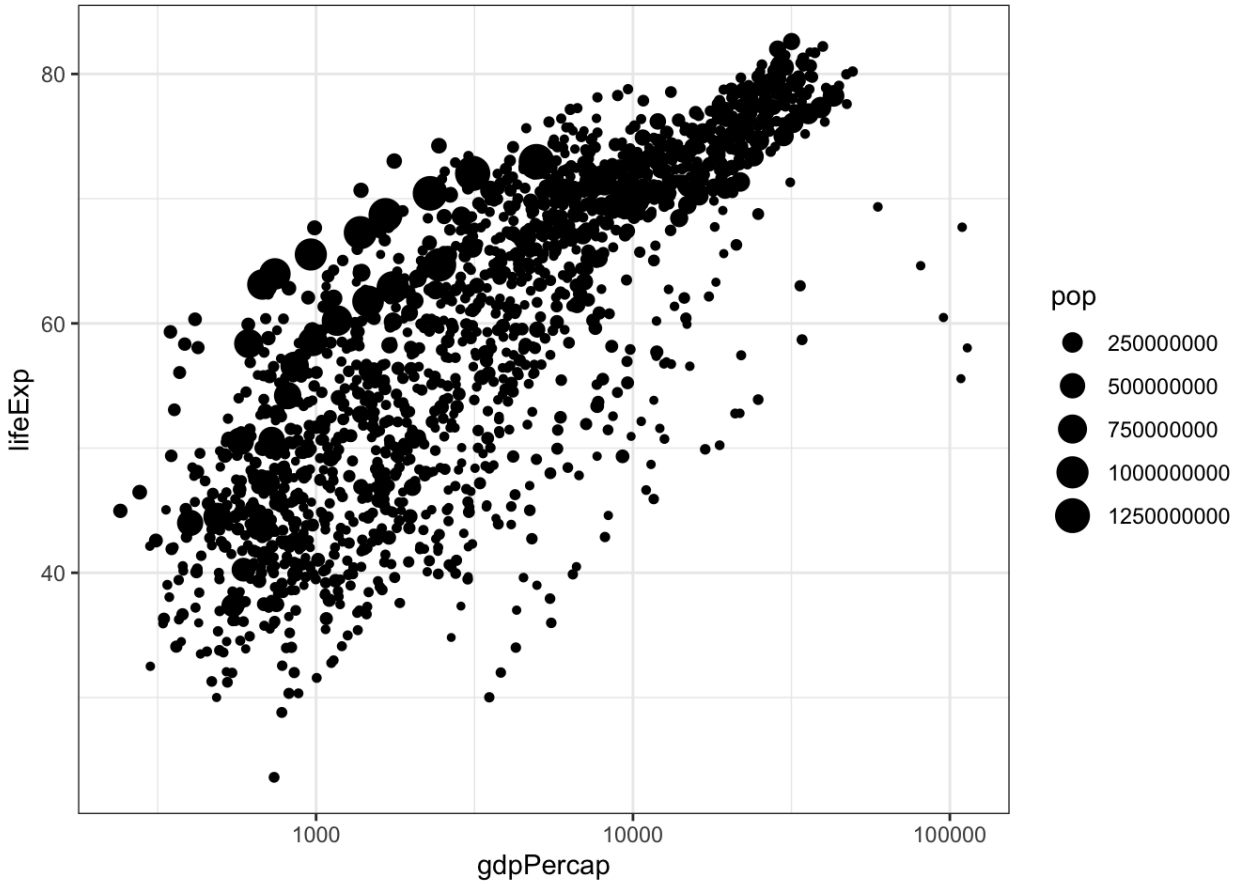
Also, there are *two* ways of animating the gapminder ggplot.

Anja Meerwald
202008898

Option 1: Animate using transition_states()

The first step is to create the object-to-be-animated

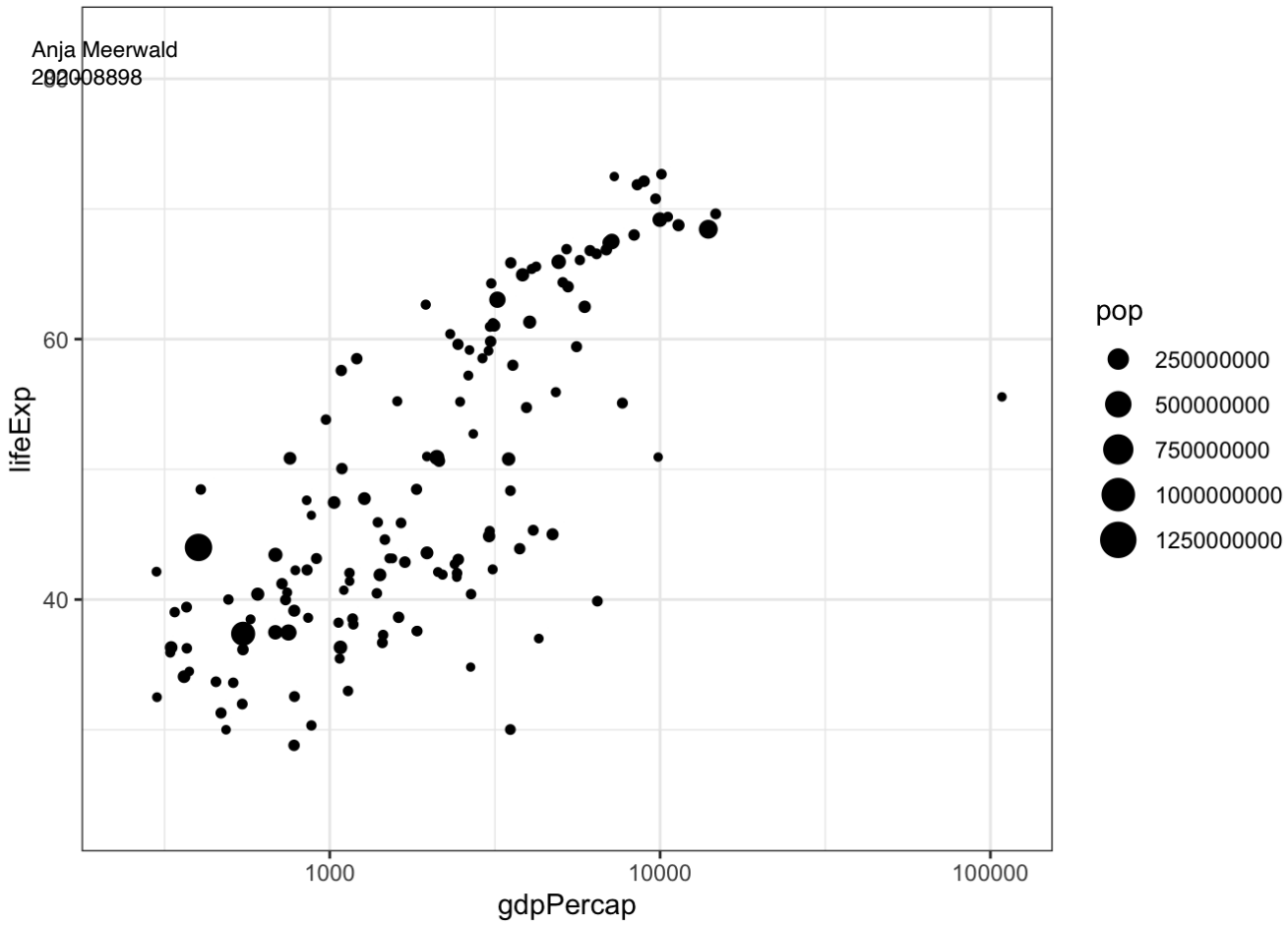
```
anim <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +
  geom_point() +
  scale_x_log10()  # convert x to log scale
anim
```



...

This plot collates all the points across time. The next step is to split it into years and animate it. This may take some time, depending on the processing power of your computer (and other things you are asking it to do). Beware that the animation might appear in the bottom right 'Viewer' pane, not in this rmd preview. You need to `knit` the document to get the visual inside an html file.

```
anim + transition_states(year,
                        transition_length = 1,
                        state_length = 1)
```



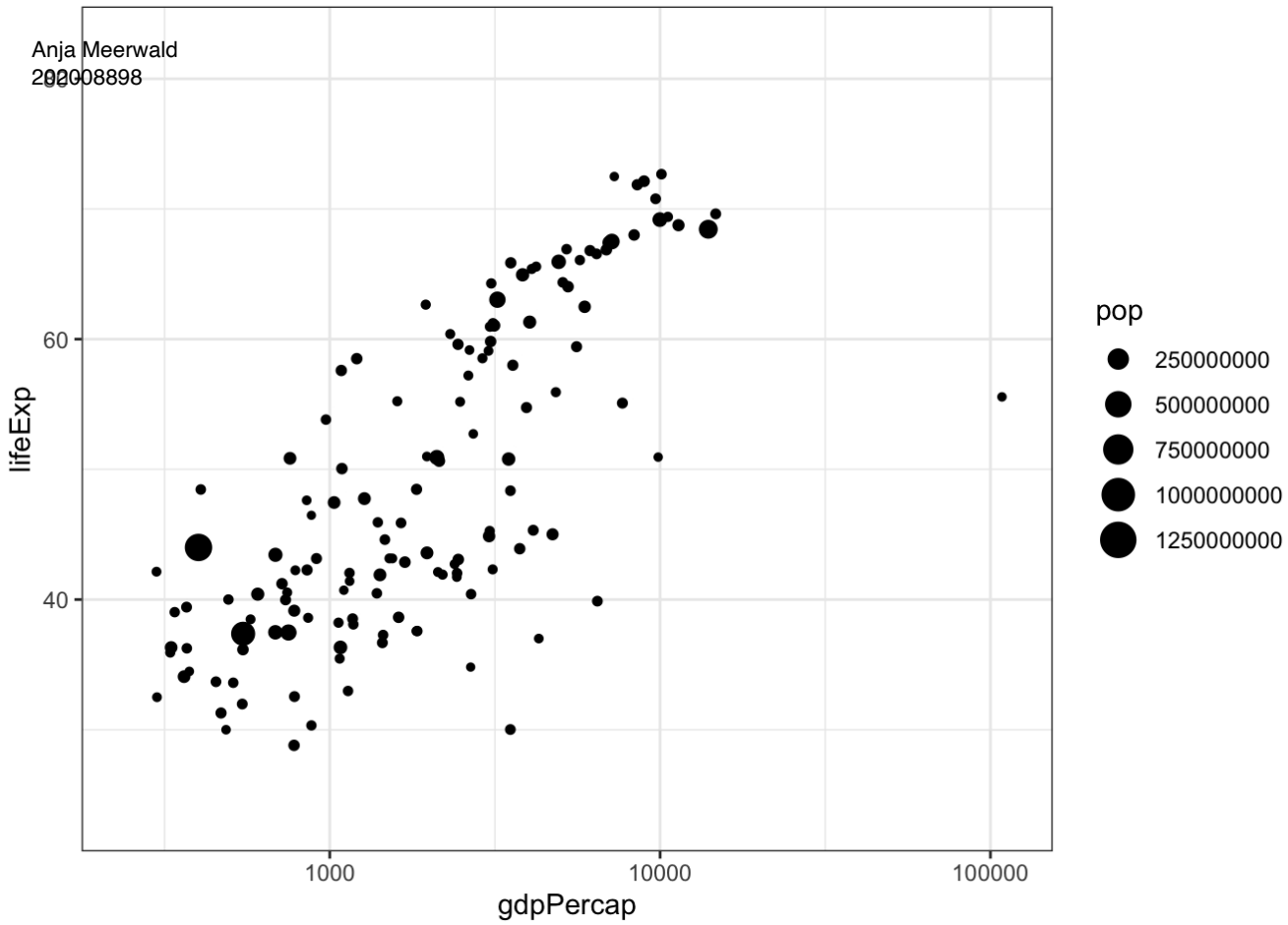
...

Notice how the animation moves jerkily, 'jumping' from one year to the next 12 times in total. This is a bit clunky, which is why it's good we have another option.

Option 2 Animate using transition_time()

This option smoothes the transition between different 'frames', because it interpolates and adds transitional years where there are gaps in the timeseries data.

```
anim2 <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +
  geom_point() +
  scale_x_log10() + # convert x to log scale
  transition_time(year)
anim2
```

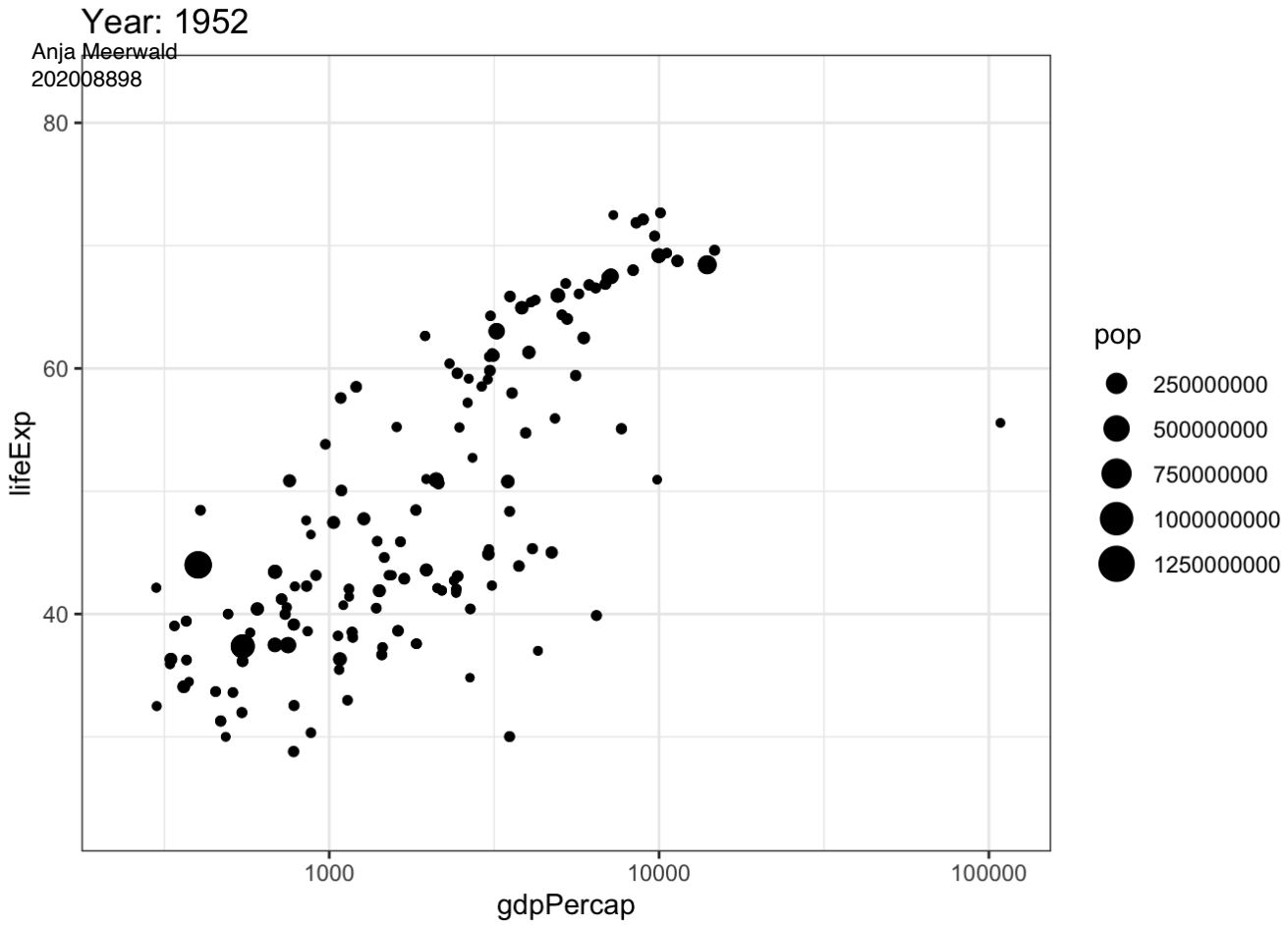


The much smoother movement in Option 2 will be much more noticeable if you add a title to the chart, that will page through the years corresponding to each frame.

Now, choose one of the animation options and get it to work. You may need to troubleshoot your installation of `gganimate` and other packages

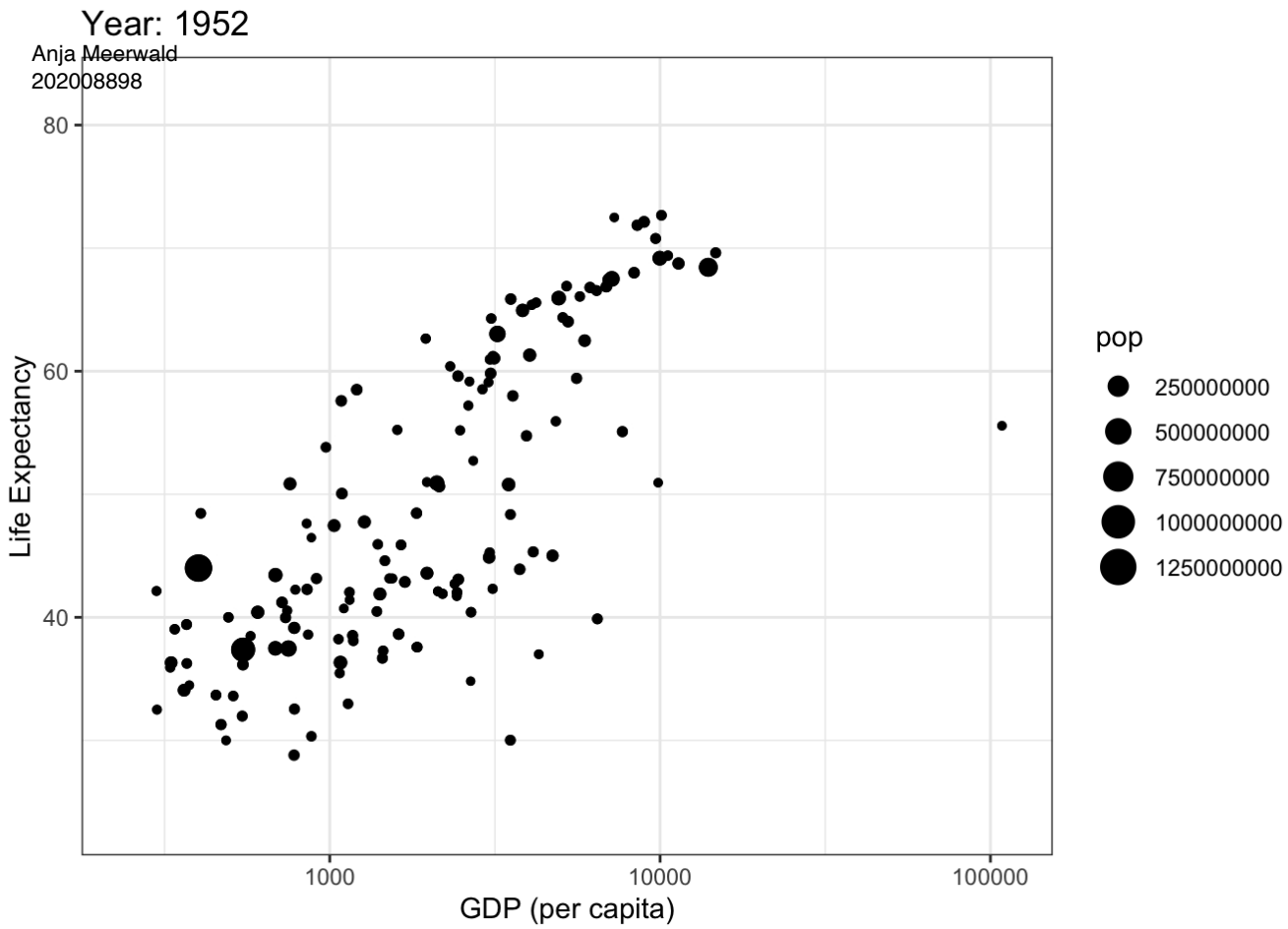
5. *Can you add a title to one or both of the animations above that will change in sync with the animation? (**Hint**: search labeling for `transition_states()` and `transition_time()` functions respectively)*

```
anim3 <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +
  geom_point() +
  scale_x_log10() +
  transition_reveal(year) +
  labs(title = 'Year: {frame_along}')
anim3
```



6. *Can you make the axes' labels and units more readable? Consider expanding the abbreviated labels as well as the scientific notation in the legend and x axis to whole numbers.*

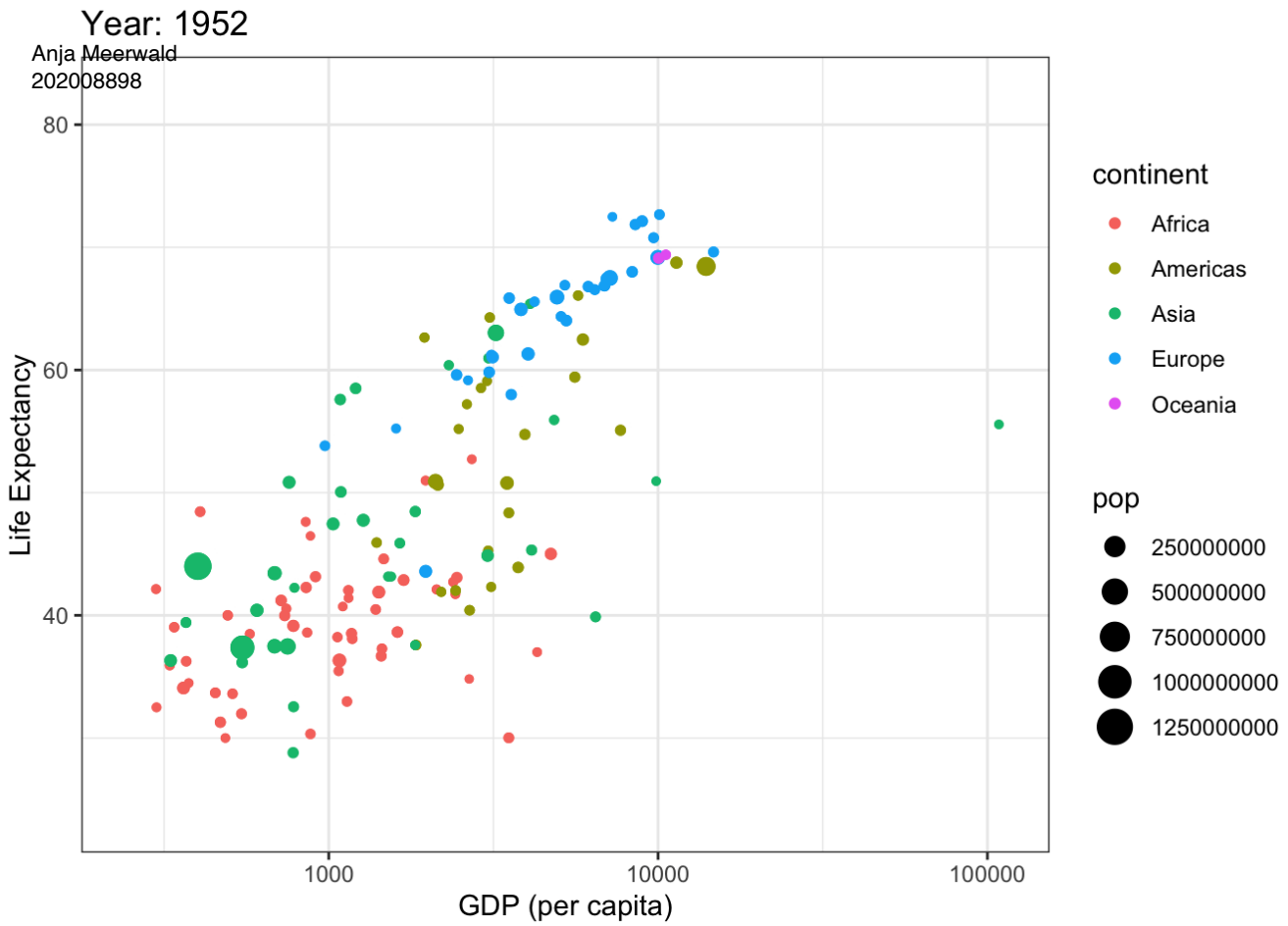
```
anim4 <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +
  geom_point() +
  scale_x_log10() +
  transition_reveal(year) +
  labs(title = 'Year: {frame_along}',x = "GDP (per capita)", y = "Life Expectancy")
anim4
```



7. *Come up with a question you want to answer using the gapminder data and write it down. Then, create a data visualization that answers the question and explain how your visualization answers the question. (Example: you wish to see what was mean life expectancy across the continents in the year you were born versus your parents' birth years). [Hint: if you wish to have more data than is in the filtered gapminder, you can load either the* `gapminder_unfiltered` *dataset and download more at https://www.gapminder.org/data/ (https://www.gapminder.org/data/)]*

How has life expectancy and GDP has changed over the years and across the various continents?

```
anim5 <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, color = continent)) +
  geom_point() +
  scale_x_log10() +
  transition_reveal(year) +
  labs(title = 'Year: {frame_along}', x = "GDP (per capita)", y = "Life Expectancy")
anim5
```



By making this visual animated, it is easy to see the increase in both life expectancy and GDP over the years. Coloring by continent shows where contienents lie in terms of those two variables and where they are in relation to the other continents. With the continued use of size indication population it is also easy to compare population sizes of the continents over the years.

Anja Meerwald
202008898

5:W35: Managing Files on Steroids with Shell

DESCRIPTION

Your supervisor has shared a folder of photos on Sciencedata.dk with you (password is 2020CDS, folder is 500Mb and contains 189 images) and needs your help with a couple diagnostics:

1) Identify the names and format of the 3 biggest files. Can you come up with a command to generate a numerically ordered list of 3 biggest files? (hint: consider using **wc** to gauge image size)
[lists all the files with sizes starting with the largest, and then taking the first three of that list]
→ ls -S | head -n 3
Lists the files from largest to smallest | gets the top three results which are the biggest files

9240_Overview_S.RW2
9247_Overview_SW.RW2
9237_Overview_W.RW2

→ wc -c * | sort -n | tail -n 4
Find the byte size | sorts from smallest to largest | gets the last three 3 which are the biggest files, need 4 because it also gets the total line

2) Some of the image files are empty, a sign of corruption. Can you **find** the empty photo files (0 kb size) , count them, and generate a list of their filenames to make their later replacement easier?

→ find -empty > empty_files.txt
→ wc -l empty_files.txt

Tells us there are 74 empty files

3) **Optional/Advanced:** Imagine you have a directory goodphotos/ (same password as above) with original non-zero-length files sitting at the same level as the current directory. How would you write a loop to replace the zero length files?

Anja Meerwald
202008898

6:W43: Practicing functions with Gapminder

DESCRIPTION

Use the gapminder dataset from Week 43 to produce solutions to the three tasks below. Post the .R script or .Rmd and .html in your au##### github repository and link it here:

https://github.com/Digital-Methods-HASS/au665920_Meerwald_Anja

1. Define a defensive function that calculates the Gross Domestic Product of a nation from the data available in the gapminder dataset. You can use the population and GDPpercapita columns for it. Using that function, calculate the GDP of Denmark in the following years: 1967, 1977, 1987, 1997, 2007, and 2017.
2. Write a script that loops over each country in the gapminder dataset, tests whether the country starts with a 'B' , and prints out whether the life expectancy is smaller than 50, between 50 and 70, or greater than 70. (**Hint:** remember the grepl function, and review the Control Flow tutorial)
3. **Challenge/Optional**: Write a script that loops over each country in the gapminder dataset, tests whether the country starts with a 'M' and graphs life expectancy against time (using plot() function) as a line graph if the mean life expectancy is under 50 years.

Hint: If you are struggling with the gapminder tibble format, consider converting it into a dataframe, either by downloading it from the internet and loading it via read.csv (not read_csv), and/or using as.data.frame() conversion function and then appropriately subsetting.

HW6 - Practicing functions with Gapminder

Anja Meerwald
10/30/2022

Anja Meerwald

10/30/2022

1. Define a defensive function that calculates the Gross Domestic Product of a nation from the data available in the gapminder dataset. You can use the population and GDPpercapita columns for it. Using that function, calculate the GDP of Denmark in the following years: 1967, 1977, 1987, 1997, 2007, and 2017.

```
# creating the function, used from https://swcarpentry.github.io/r-novice-gapminder/10-functions/index.html

calcGDP <- function(dat, year=NULL, country=NULL) { # defining the name of the function and which variables are included. They will be null if not specified
  if(!is.null(year)) {   # if it's not null, then...
    dat <- dat[dat$year %in% year, ]   # creates a temporary variable dat which takes the subsetted data by year if it's specified
  }
  if (!is.null(country)) {
    dat <- dat[dat$country %in% country,]  # same as above, subsetting by country if provided
  }
  gdp <- dat$pop * dat$gdpPercap  # calculating the gdp with the population and gdpPercap variables

  new <- cbind(dat, gdp=gdp) # pputs that subsetted data from above with a new gdp column and treturns the result
  return(new)
}

# calculating Denmark's GDP in the specified years except for 2017 because it's not included in the dataset
calcGDP(gapminder, country = "Denmark", year = c(1967, 1977, 1987, 1997, 2007, 2017))
```

```
##   country continent year lifeExp     pop gdpPercap          gdp
## 1 Denmark    Europe 1967  72.960 4838800  15937.21  77116977700
## 2 Denmark    Europe 1977  74.690 5088419  20422.90 103920280028
## 3 Denmark    Europe 1987  74.800 5127024  25116.18 128771236166
## 4 Denmark    Europe 1997  76.110 5283663  29804.35 157476118456
## 5 Denmark    Europe 2007  78.332 5468120  35278.42 192906627081
```

2. Write a script that loops over each country in the gapminder dataset, tests whether the country starts with a ‘B’ , and prints out whether the life expectancy is smaller than 50, between 50 and 70, or greater than 70. (Hint: remember the grepl function, and review the Control Flow tutorial)

```
lowerThreshold <- 50
upperThreshold <- 70
```

Anja Meerwald
202008898

```
B_countries <- grep("^B", unique(df$country), value = TRUE) # using grep to find countri
es that start with 'B' and assigning them to the new variable, B_countries

for (iCountry in B_countries) {    # looping through the countries within the B_countrie
s variable
    tmp <- mean(df[df$country == iCountry, "lifeExp"])   # getting the mean life expecta
ncy
# using the lower and upper thresholds to determine the printed output
    if (tmp < lowerThreshold) {
        cat("Average Life Expectancy in", iCountry, "is less than", lowerThreshold,
"\n") # if the life expectancy is less than 50, print it's less than
    } else if(tmp > lowerThreshold && tmp < upperThreshold) {   # if the life expectancy
is between 50-70, print it is between
        cat("Average Life Expectancy in", iCountry, "is between", lowerThreshold, "and",
upperThreshold, "\n")
    } else {
        cat("Average Life Expectancy in", iCountry, "is greater than", upperThreshold,
"\n") # and if it's greater than 70, print that
    } # end if
    rm(tmp)
} # end for loop
```

```
## Average Life Expectancy in Bahrain is between 50 and 70
## Average Life Expectancy in Bangladesh is less than 50
## Average Life Expectancy in Belgium is greater than 70
## Average Life Expectancy in Benin is less than 50
## Average Life Expectancy in Bolivia is between 50 and 70
## Average Life Expectancy in Bosnia and Herzegovina is between 50 and 70
## Average Life Expectancy in Botswana is between 50 and 70
## Average Life Expectancy in Brazil is between 50 and 70
## Average Life Expectancy in Bulgaria is between 50 and 70
## Average Life Expectancy in Burkina Faso is less than 50
## Average Life Expectancy in Burundi is less than 50
```

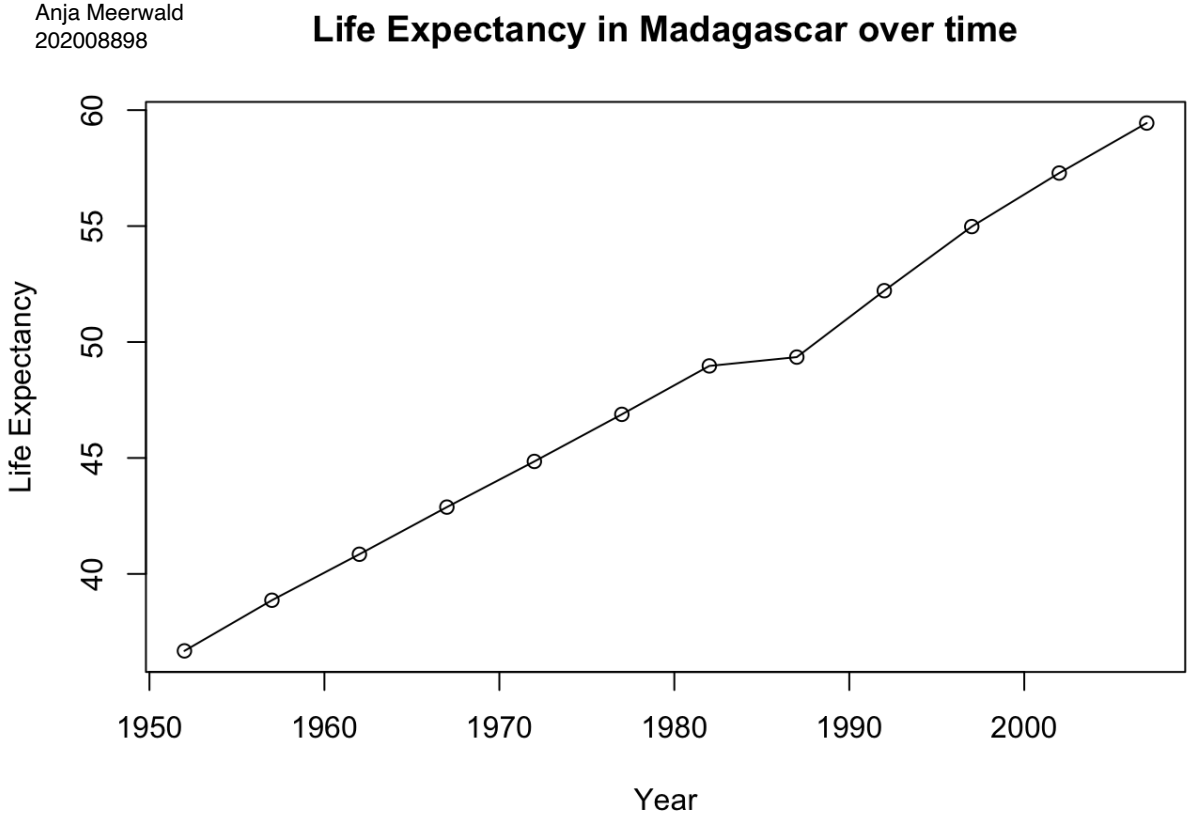
3. Challenge/Optional: Write a script that loops over each country in the gapminder dataset, tests whether the country starts with a ‘M’ and graphs life expectancy against time (using plot() function) as a line graph if the mean life expectancy is under 50 years.

```
thresholdValue <- 50 # setting the threshold to 50 years
candidateCountries <- grep("^M", unique(gapminder$country), value = TRUE) # using grep t
o find countries that start with 'M' and assigning them to the new variable, M_countries

for (iCountry in candidateCountries) {   # looping through the countries within the M_co
untries variable
    tmp <- mean(df[df$country == iCountry, "lifeExp"]) # calculating mean life expectanc
y

    if (tmp < thresholdValue) { # if the mean life expectancy is less than the threshold
(50), print that and the plot it
        cat("Average Life Expectancy in", iCountry, "is less than", thresholdValue, "plo
tting life expectancy graph... \n")

        with(subset(gapminder, country == iCountry),  #
                plot(year, lifeExp,
                     type = "o",
                     main = paste("Life Expectancy in", iCountry, "over time"),
                     ylab = "Life Expectancy",
                     xlab = "Year"
                     ) # end plot
             ) # end with
    } # end if
    rm(tmp)
}
```

Anja Meerwald
202008898

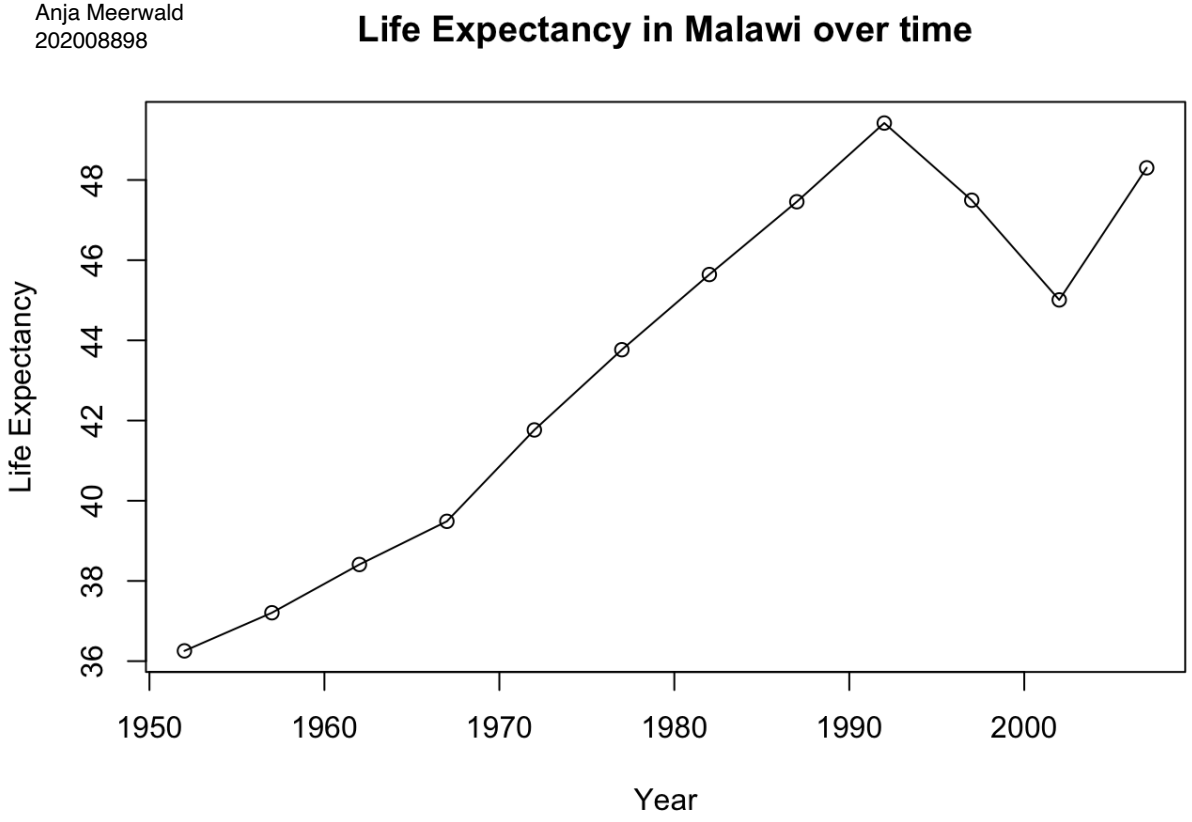
```
## Average Life Expectancy in Madagascar is less than 50 plotting life expectancy grap
h...
```

Anja Meerwald
202008898



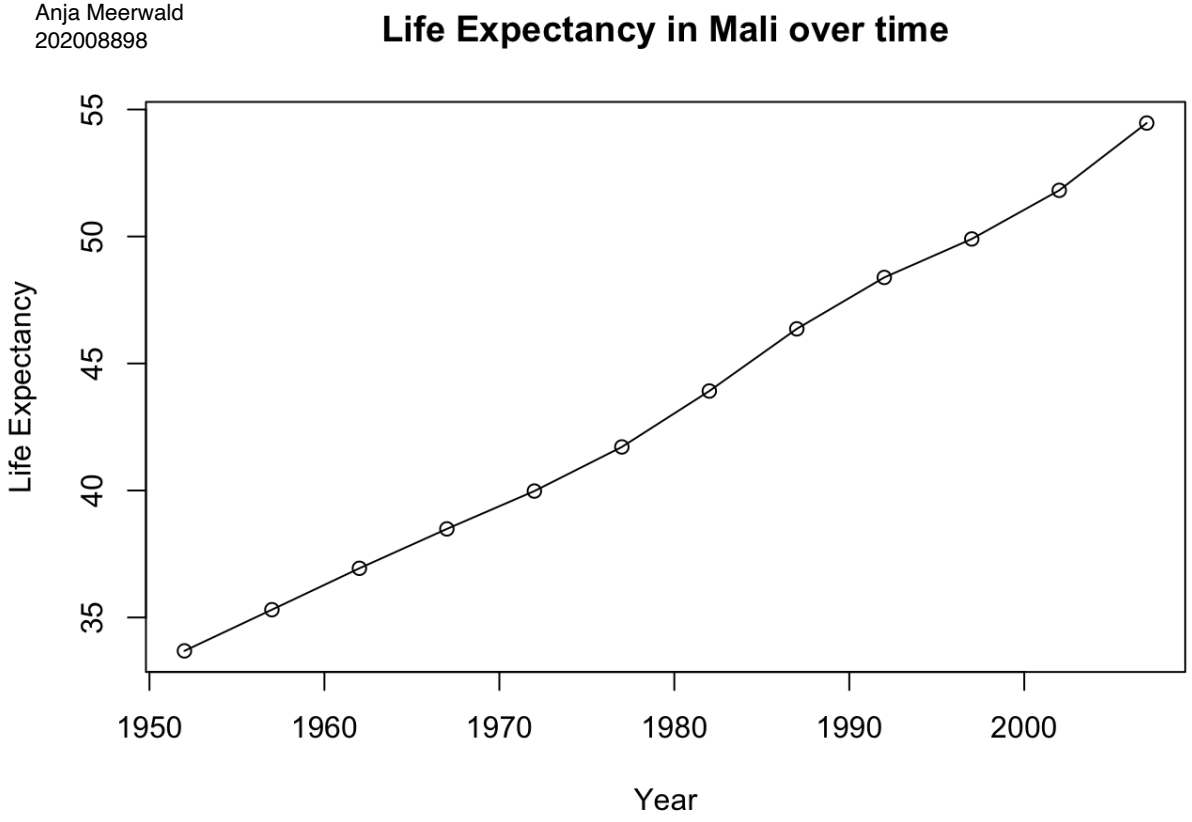

```
## Average Life Expectancy in Malawi is less than 50 plotting life expectancy graph...
```

Anja Meerwald
202008898




```
## Average Life Expectancy in Mali is less than 50 plotting life expectancy graph...
```

Anja Meerwald
202008898




```
## Average Life Expectancy in Mozambique is less than 50 plotting life expectancy grap
h...
```

Anja Meerwald
202008898

The effect of swinger clubs

What type of effect do swinger clubs have on the sexual interests of their visitors?

Anja Meerwald
202008898

Why study sex and swinger clubs?

- Inspiration: How do taboo ideas/behaviors spread?
- Bachelors project - where do sexual kinks come from
  - Influence of swinger clubs

Anja Meerwald
202008898

Hypothesis

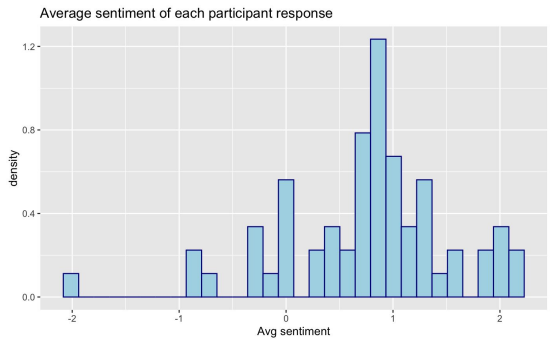
Swinger clubs have a positive effect and therefore positive sentiment will be found in participant's answers.

Method

- Sentiment analysis

Results

Overall sentiment was positive in both surveys



The effect of swinger clubs on sexual interests

Anja Meerwald (202008898@post.au.dk)

School of Communication and Cognition, University of Aarhus, Jens Chr. Skous Vej 2, 8000, Aarhus Denmark

Lecturer: Adéla Sobotkova

Abstract:
Studying taboo behavior, where we adopt these behaviors from and what effects they have on us is an important but understudied field. The aim of this paper is to investigate the effects of swinger clubs on their members' sexual interests by using sentiment analysis on free text responses. Results show that sentiment is overall positive but highlight concerns with the available tools, Sentida, used to evaluate the text.

Keywords: Taboo behavior, sexual interests, swinger club, NLP, sentiment analysis, SENTIDA

Anja Meerwald
202008898

Table of Contents

Anja Meerwald
202008898

Introduction

How do we adopt new behaviors or interests when we rarely see or hear about them? There are many studies and theories behind how ideas and behaviors spread, from twitter hashtags to fistbumps. These often rely on repeated exposure of the behavior or idea so how do behaviors or interests evolve if the subject is taboo and therefore occasionally discussed and even more rarely observed? These questions have led to my interest in the taboo subject of sexual kinks and how we acquire them. It is the subject of my bachelor's thesis as well as a previous study. This project, which supplements my thesis, will look at the effect of swinger clubs on its visitors by using sentiment analysis on participant responses from a survey.

Cultural context

Due to the often taboo nature of sex, it is an under researched field and therefore I find it is an important field to study. As a result of this gap in research, this field lacks strong frameworks, theories, and so on to guide future research. It makes this field both incredibly exciting and challenging. The comparison between swingers and non swingers (what the kink community calls “normal” sexual behaviors or individuals) was inspired by a personal connection. I was introduced to a swinger club through my højskole in a psychology class. We went on a field trip to tour the club during their off hours and speak to the couple who owns the club. It was a fascinating and educational experience that stuck with me. Now years later, after considering the question of where we acquire kinks, the visitors of a swinger club become a unique community to learn from. They encounter and engage with sex in a more open and direct way than most people. Unfortunately, studies have found that kink practitioners can feel stigmatized for their interests (Colosi & Lister, 2019; Lin, 2017) therefore, minimizing stigma around their community is a motivator in studying and shedding light on this community.

Cultural differences and attitudes towards sexuality have an immense impact (Heinemann et al, 2016) therefore it is important to understand that Danes have a reputation for being relaxed and open about sex. This dates back to the 1960s when they became the first country to legalize porn (Berdychevsky, & Nimrod, 2017). It is possible not all respondents were native Danes, however all answers were provided in Danish. That is important to keep in mind when discussing this project, the results and the willingness of participants to share details about their sexual behavior.

Anja Meerwald
202008898

Aim of the study

Specifically, I will be looking at a subset of the data where respondents wrote a free text response to the question: “Har klubben påvirket dine seksuelle interesser?” (Has the club affected your sexual interests?). Sentiment analysis will be applied to participant’s responses to investigate this further. This is a form of natural language processing (NLP) using computers to identify the attitudes and emotions of text (Gobinda, 2003; Medhat et al., 2014). Sentida, a Danish sentiment analysis tool provides a unique way to do this. First, the responses do not need to be translated which avoids the possibility that meaning could be lost in the translation process. Second, it does not simply access sentiment word by word but has the capability to assess a whole phrase. It was created by three students at Aarhus University using the same framework as the previous tool AFINN, a sentiment score for each word “from -5 (very negative) to +5 meaning (very positive)” (Nielsen, 2011, p.2). A key difference however is that three people rated each word in Sentida with an acceptable intercoder reliability score whereas AFINN was only rated by a single person (Lauridsen et al., 2019).

Hypothesis

In order to investigate the effects of swinger clubs on their members, free text answers will be reviewed. Providing more robust data, the responses come from two surveys, conducted months apart, and from members of multiple clubs. Sentiment analysis will be applied to determine overall sentiment of each participant's response.

Based on strong cultural influences and openness towards sexuality, Danes who are members of swinger clubs will express positive sentiment when discussing the clubs’ influence on their sexual interests.

Software Framework

The code for this project was written on a 2020 Macbook Air, 8 Gb RAM, which runs Mac OS Catalina, version 10.15.5. The programming language R (4.1.2) was used to write the code (R Core Team, 2020) and this was done in the desktop version of RStudio (1.3.1073; Rstudio Team, 2020) as an integrated development environment. A R-markdown file with the accompanying code can be found on the author’s Github.

Anja Meerwald
202008898

The following packages in R were used: tidyverse (v1.3.0; Wickham et al., 2019), Sentida (v1.1; Lauridsen et al.,2019), ggplot2 (v3.3.3; Wickham, 2016), dplyr (v1.0.8; Wickham et al, 2022), rstatix (V0.7.0; Kassambara, 2021), stopwords (v2.3; Benoit et al, 2021), ggwordcloud (v0.5.0; Le Pennec & Slowikowski, 2019), tidytext (Silge & Robinson, 2016), SnowballC (v0.7.0; Bouchet-Valat, 2020), wordcloud (v2.6; Fellows, 2018), RColorBrewer (v1.1-3; Neuwirth, 2022), and tm (v0.7-9; Feinerer & Hornik, 2022).

Data Acquisition and Processing

The data used in this project come from two surveys, collected in May 2022 and November 2022. They were circulated by four different swinger clubs in their private member groups. The second survey was intended as a replication of the first so questions were the same. Surveys were done on google forms and the data came in the form of a .csv file for each survey. These files can be found on the author's Github. After loading in the packages mentioned in the previous section, the .csv files were imported. The columns were renamed and shortened for easier viewing and understanding. In the first survey, certain numeric responses were not limited so basic cleaning of those columns were needed. In the second survey this was adjusted and therefore less cleaning was needed. Converting yes/no responses to 1/0 for future analysis was also done. Then the data for this project was isolated, called 'club_affect' and NA responses were removed. Which left for the original study (n = 66) and the reproduction (n = 62). All data manipulations from this point on were done twice, once to each data frame. To easily distinguish, 'repo_' was added to the same variable names so it is clear which data frame it is referring to. The data for this project is respondents free text phrases therefore they were split by phrase and also by word. A Danish stop word list was applied to assess the most popular words used as well as applying SENTIDA for sentiment analysis of the phrases. Lastly, two t-tests were used to determine if there was a difference in sentiment between the two surveys but also to evaluate SENTIDA as a tool, determining if there was a significant difference in the results between total and average sentiment.

Empirical Results

The results consist of two word clouds, showing the top 100 words used in participants free text responses. Visually in figure 1, it is apparent that they share many of the most frequently used words.

Anja Meerwald
202008898

Figure 1. Word clouds, the original study and then the reproduced study.

Once applying SENTIDA to the responses, the results are visualized, see figure 2, and a statistical summary is applied. When assessing total sentiment for the original study (N = 66), *(M = 2.93, SD = 2.87)* and the reproduced study (N = 62), *(M = 1.65, SD = 2.32)* showed a significant difference $t(123.3) = -2.87$, $p < 0.05$. However, the original study's average sentiment *(M = 0.86, SD = 2.87)* did not show a significant difference compared to the reproduced study ($M = 0.75$, $SD = 0.79$), $t(115.16) = -0.88$, $p > 0.05$.

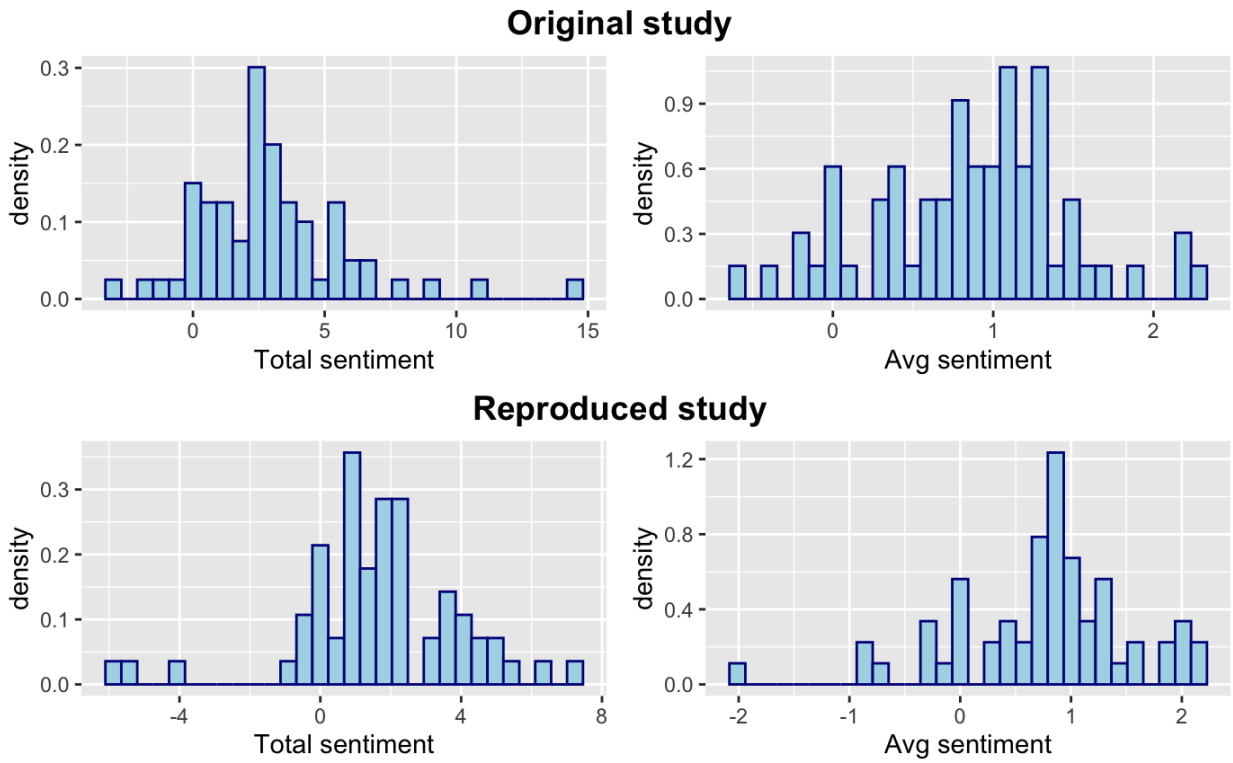


Figure 2. Histograms showing both total and average sentiment for responses in both studies.

Lastly, figure 3 shows the amount of positive and negatively valenced phrases according to SENTIDA. Their y-axis are slightly offset making a complete direction comparison difficult but overall it appears to be mostly positive and similar between the studies.

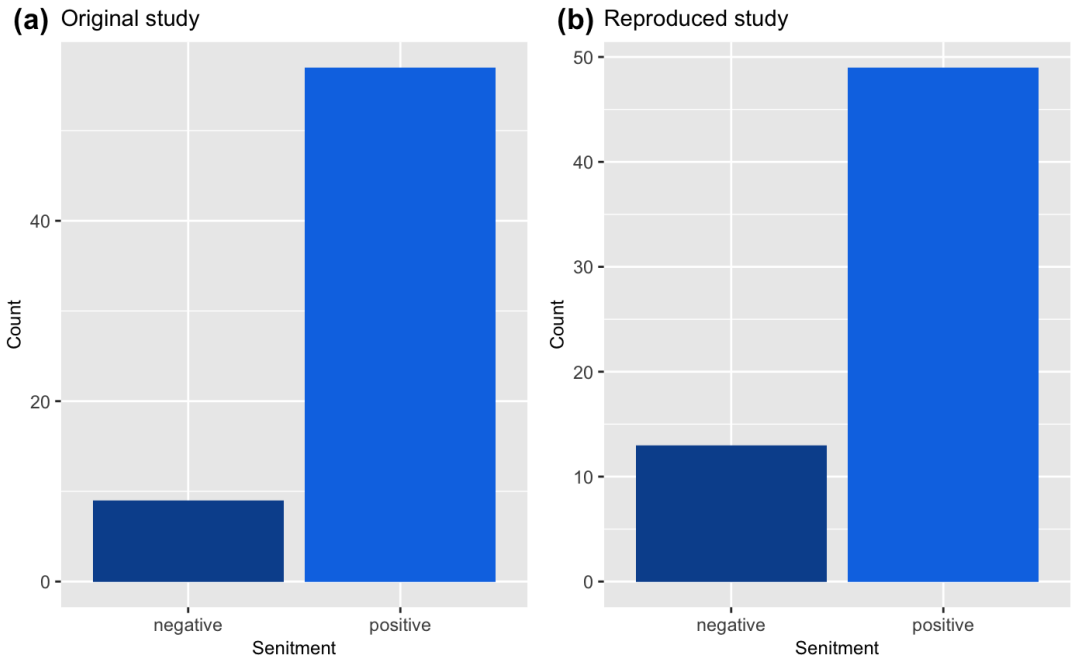


Figure 2: Shows the overall sentiment compared between both studies.

Anja Meerwald
202008898

Critical evaluation

Results were found to support the hypothesis. The sentiments in both surveys were overall positive. Although there was a significant difference between the total sentiment ratings. Despite both being mostly positive, the original study was significantly more so. The sample sizes are similar so it is clear why this is the case. Perhaps diving deeper into the provided data could explain this, perhaps the group is older, or has been swinging for longer. Overall, hopefully this study achieves the goal set out for this project, of highlighting swinger clubs, removing some stigma, and showing their club members find positive effects on their exploration of sexual interests.

Given that respondents are swinger club members this does not come as a surprise. One is hardly a club member of something they do not enjoy or receive some personal benefit from. What becomes more interesting in this project is the use of Sentida. While it is very nice to have the Danish sentiment tool, Sentida (2020) and not just one that looks at individual words but can take into account the sentiment of a phrase, issues still arise.

When looking through the data and sentiment scores, phrases where participants expressed happiness or positive experiences were still sometimes given a negative sentiment rating. Here is an example using the answer with the most negative average sentiment from the original study, “Blev introduceret for bdsm og jeg fandt det jeg manglede” which translates to, “I was introduced to bdsm and found what I was missing”. It does not seem like a negatively valenced statement however, when looking at Sentida’s rating word by word, there is a surprising result. Many of the words are neutral, however “fandt” (found) is very negative which skews this phrase to have an overall highly negative sentiment. This is why when relying on a tool such as Sentida, it becomes crucial to understand how it works, not just simply accept the results.

Sentida as a tool was overall easy to use and again better than nothing. When working with an uncommon language it is helpful to have a tool to use at all. It would be interesting in a future project to translate the participants' responses into English, apply an English sentiment tool and evaluate which is more accurate.

Lastly, when a topic such sex, a social desirability bias should be mentioned. This bias is when participants provide answers that they believe are more culturally acceptable (Nikolopoulou, 2022). It is possible participants were even more positive because they feel they should be as a swinger club member or that they downplayed their thoughts because it is less culturally acceptable.

Anja Meerwald
202008898

Conclusions

The project used sentiment analysis to review participants' responses regarding how swinger clubs have affected their sexual interests. The results supported the hypothesis that the sentiment would overall be positive. The most interesting result however, was that Sentida incorrectly rated some phrases highlighting the need to be critical of the tools we rely on and mention their shortcomings. It would be interesting and perhaps helpful to investigate this further by comparing Sentida with translated text and English sentiment tools available.

Anja Meerwald
202008898

References

Benoit, K., Muhr, D., & Watanabe, K. (2021). stopwords: Multilingual Stopword Lists. R package version 2.3, URL: https://CRAN.R-project.org/package=stopwords.

Berdychevsky, L., & Nimrod, G. (2017). Sex as Leisure in Later life: A Netnographic Approach. *Leisure Sciences*, *39*(3), 224–243. https://doi.org/10.1080/01490400.2016.1189368

Bouchet-Valat, M. (2020). SnowballC: Snowball Stemmers Based on the C 'libstemmer' UTF-8 Library. R package version 0.7.0, URL: https://CRAN.R-project.org/package=SnowballC.

Colosi, R., & Lister, B. (2019). Kinking it up: An exploration of the role of online social networking site FetLife in the stigma management of kink practices. In *British Criminology Conference* (Vol. 19, pp. 5-24).

Feinerer I, Hornik K (2022). tm: Text Mining Package. R package version 0.7-9, URL: https://CRAN.R-project.org/package=tm.

Fellows, I. (2018). wordcloud: Word Clouds. R package version 2.6, URL: https://CRAN.R-project.org/package=wordcloud.

Gobinda, G. C. (2003). Natural language processing. *Annual Review of Information Science and Technology*, *37*, 51-89.

Heinemann, J., Atallah, S., & Rosenbaum, T. (2016). The impact of culture and ethnicity on sexuality and sexual function. *Current Sexual Health Reports*, *8*(3), 144-150.

Kassambara, A. (2021). *Rstatix:Pipe-friendly framework for basic statistical tests.* https://rpkgs.datanovia.com/rstatix/

Lauridsen, G. A., Dalsgaard, J. A., & Svendsen, L. K. (2019). SENTIDA: A new tool for sentiment analysis in Danish. *Journal of Language Works,* 4(1), 38-53. https://tidsskrift.dk/lwo/article/view/115711

Lauridsen, G., Svendsen, L., & Dalsgaard, J. (2020). Sentida: Sentiment scoring of words and sentences. *R package,* version 1.1. https://github.com/Guscode/Sentida

Le Pennec, E. & Slowikowski, K. (2019). ggwordcloud: A Word Cloud Geom for 'ggplot2'. R package version 0.5.0, URL: https://CRAN.R-project.org/package=ggwordcloud.

Lin, K. (2017). The medicalization and demedicalization of kink: Shifting contexts of sexual politics. *Sexualities*, *20*(3), 302-323.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: a survey. Ain Shams Eng. J. 5 (4), 1093–1113.

Neuwirth E (2022). RColorBrewer: ColorBrewer Palettes. R package version 1.1-3, URL: https://CRAN.R-project.org/package=RColorBrewer.

Nikolopoulou, K. (2022, November 18). *What is social desirability bias? Definition and examples.* Scribbr. https://www.scribbr.com/research-bias/social-desirability-bias/?fbclid=IwAR0Ef9Dse7ltohweYjrInQnz8sw2WOSkMb7K69qAaqpBYHcuVcReNIvTsz8

R Core Team (2020). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rstudio Team (2020). Rstudio: Integrated Development for R. Rstudio, PBC, Boston, MA URL http://www.rstudio.com/.

Silge, J., & Robinson, D. (2016). *tidytext: Text Mining and Analysis Using Tidy Data Principles in R.* JOSS, 1(3). doi: 10.21105/joss.00037 (URL: https://doi.org/10.21105/joss.00037), URL: http://dx.doi.org/10.21105/joss.00037.

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis* (2nd ed.) [PDF]. Springer International Publishing.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K.,

Ooms, J., Robinson, D., Seidel, D., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686.

Wickham, H., François, R., Henry, L., Müller, K (2022). *dplyr: A grammar of data manipulation*. https://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr

Anja Meerwald
202008898

Required Metadata

Table 1 – Software metadata

Nr	Software metadata description	***Please fill in this column***
S1	Current software version	*R 4.1.2, Rstudio 1.3.1073*
S2	Permanent link to Github repository where you put your script or R project	*https://github.com/Digital-Methods-HASS/au665920_Meerwald_Anja*
S3	Legal Software License	*Not applicable*
S4	Computing platform / Operating System	*OS Catalina, version 10.15.5*
S5	Installation requirements & dependencies for software not used in class	*Not applicable*
S6	If available Link to software documentation for special software	*Not applicable*
S6	Support email for questions	*202008898@post.au.dk*

Table 2 – Data metadata

Nr	Metadata description	***Please fill in this column***
D1	Sex Survey.csv	*Survey responses to questionnaire collected in May 2022 collected by Anja Meerwald, in connection with the Social and Cultural Communication exam. There are 85 columns including demographic data, information about kink enjoyment, swinger club attendance, and for this project specifically a column for how swinger clubs have affected participants' sexual interests.*
D2	Reproduction Sex Survey CDS.csv	*Survey responses to questionnaire collected in November 2022 collected by Anja Meerwald, in connection with bachelor's thesis. There are 86 columns including demogrphic data, information about kink enjoyment, swinger club attendance, and for this project specifically a column for how swinger clubs have affected participants' sexual interests.*