

3: W35: Start with R

Task 1

At first, I create a vector called “rooms” containing the elements from the assignment description. For this I use the <- function. The function c() is used to add new elements to a vector.

```
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)
```

Now I want to remove the NA elements from the “rooms” vector. I create a new vector called “rooms-noNA” containing the elements from the “rooms” vector but without the NAs:

```
rooms_noNA <- rooms[!is.na(rooms)]
```

Then I create a new vector which only contain the values from the rooms_noNA vector which is greater than 2:

```
#Then I create a new vector only containing the elements which is greater than 2  
list(rooms_noNA[rooms_noNA>2])  
rooms_above_2 <- rooms_noNA[rooms_noNA>2]  
print(rooms_above_2)
```

Then I use the sum() function on the “rooms_above_2” vector to find the sum of the elements greater than two. The sum is 55

```
> sum(rooms_above_2)  
[1] 55
```

Task 2

The data in the “rooms” vector is I use the function is.numeric() to check, if the data is numeric data. The printed answer is TRUE, which means that the data is numeric.

```
> #TASK 2: what type of data is the "rooms" vvector?  
> is.numeric(rooms)  
[1] TRUE  
>
```

Task 3

To begin this task, I install the tidyverse package in RStudio with the library() function. Then I download the file SAFI_clean.csv with the download.file() and read_csv() functions:

```
#Downloading the SAFI_clean.csv file into RStudio

library(tidyverse)

download.file("https://ndownloader.figshare.com/files/11492171",
             "SAFI_clean.csv", mode = "wb")
read_csv("SAFI_clean.csv")
```

Now a part of the data from SAFI_clean.csv is shown in the window in the lower left corner of the screen.

```
R 4.2.1 · ~/
> read_csv("SAFI_clean.csv")
Rows: 131 Columns: 14
— Column specification —
Delimiter: ","
chr (7): village, respondent_wall_type, memb_assoc, affect_conflicts, items_owned, months_lack_food, instanceID
dbl (6): key_ID, no_membrs, years_liv, rooms, liv_count, no_meals
dtm (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.
# A tibble: 131 × 14
  key_ID village interview_date no_membrs years_liv respond...1 rooms memb...2 affec...3 liv_c...4 items...5 no_me...6 month...7
  <dbl> <chr> <dtm> <dbl> <dbl> <chr> <dbl> <chr> <chr> <dbl> <chr> <dbl> <chr>
1 1 God 2016-11-17 00:00:00 3 4 muddaub 1 NULL NULL 1 bicycl... 2 Jan
2 1 God 2016-11-17 00:00:00 7 9 muddaub 1 yes once 3 cow_ca... 2 Jan;Se...
3 3 God 2016-11-17 00:00:00 10 15 burntbri... 1 NULL NULL 1 solar_... 2 Jan;Fe...
4 4 God 2016-11-17 00:00:00 7 6 burntbri... 1 NULL NULL 2 bicycl... 2 Sept;0...
5 5 God 2016-11-17 00:00:00 7 40 burntbri... 1 NULL NULL 4 motorc... 2 Aug;Se...
6 6 God 2016-11-17 00:00:00 3 3 muddaub 1 NULL NULL 1 NULL 2 Aug;Se...
7 7 God 2016-11-17 00:00:00 6 38 muddaub 1 no never 1 motorc... 3 Nov
8 8 Chirodzo 2016-11-16 00:00:00 12 70 burntbri... 3 yes never 2 motorc... 2 Jan
9 9 Chirodzo 2016-11-16 00:00:00 8 6 burntbri... 1 no never 3 televi... 3 Jan;Dec
10 10 Chirodzo 2016-12-16 00:00:00 12 23 burntbri... 5 no never 2 cow_ca... 3 Jan;Oc...
```

I have taken a screenshot of my RStudio interface, which shows the line of code I used to create the object, the 'interviews' object in the Environment, and the file structure of my R project in the bottom right "Files" pane. The screenshot can be accessed on Github here:

https://github.com/Digital-Methods-HASS/au672638_Jorgensen_Emma-Marie/blob/main/hw3_screenshot.png

Task 4

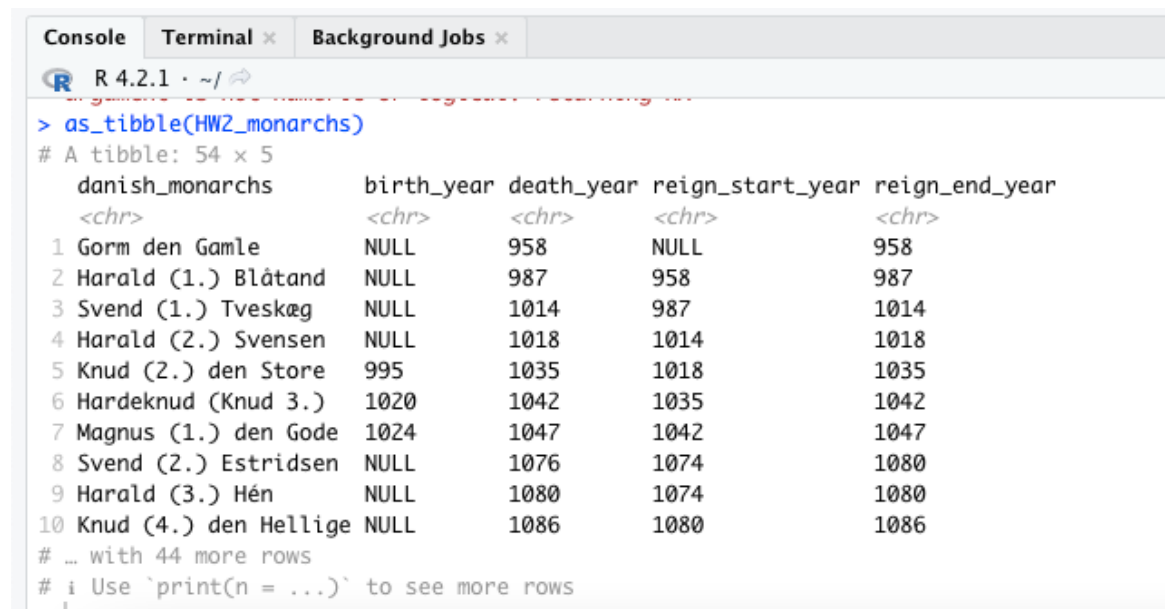
I started out by installing the tidyverse packages to be able to create tibble from my data set:

```
# Installing tidyverse (to be able to make tibbles)
install.packages("tidyverse")
library(tidyverse)
```

For the homework assignment 2 I created a tidy spreadsheet with data about the Danish monarchs called “HW2_monarchs”, which I will use in this task. To load the data as a tibble, I used the function `as_tibble(HW2_monarchs)`

In the console I now see the first ten rows from the “HW2_monarchs” dataset:

```
# Creating a tibble out of the data set
as_tibble(HW2_monarchs)
print(as_tibble(HW2_monarchs))
```



```
R 4.2.1 · ~/
> as_tibble(HW2_monarchs)
# A tibble: 54 × 5
  danish_monarchs birth_year death_year reign_start_year reign_end_year
  <chr>           <chr>      <chr>      <chr>      <chr>
1 Gorm den Gamle  NULL      958        NULL      958
2 Harald (1.) Blåtand NULL      987        958      987
3 Svend (1.) Tveskæg NULL      1014       987     1014
4 Harald (2.) Svensen NULL      1018       1014    1018
5 Knud (2.) den Store 995       1035       1018    1035
6 Hardeknud (Knud 3.) 1020      1042       1035    1042
7 Magnus (1.) den Gode 1024      1047       1042    1047
8 Svend (2.) Estridsen NULL      1076       1074    1080
9 Harald (3.) Hén      NULL      1080       1074    1080
10 Knud (4.) den Hellige NULL      1086       1080    1086
# ... with 44 more rows
# i Use `print(n = ...)` to see more rows
```

The missing data is called NULL in my data set, but R doesn’t interpret NULL as missing data. Before I can calculate anything from my data set, I therefore must transform the NULL values into NA:

```
# Changing the NULL values into NA, which R interprets as missing data
```

```
HW2_monarchs[HW2_monarchs == "NULL"] <- NA
print(HW2_monarchs)
|
```

When I print the tibble again, the missing data is now named NA:

```
> print(as_tibble(HW2_monarchs))
# A tibble: 55 x 6
  danish_monarchs birth_year death_year reign_start_year reign_end_year years_ruled
  <chr>           <chr>      <chr>      <chr>           <chr>      <chr>
1 Gorm den Gamle  NA        958        NA             958        NA
2 Harald (1.) Blåtand NA        987        958            987        29
3 Svend (1.) Tveskæg NA        1014       987            1014       27
4 Harald (2.) Svensen NA        1018       1014           1018       4
5 Knud (2.) den Store 995       1035       1018           1035       17
6 Hardeknud (Knud 3.) 1020      1042       1035           1042       7
7 Magnus (1.) den Gode 1024      1047       1042           1047       5
8 Svend (2.) Estridsen NA        1076       1074           1080       6
9 Harald (3.) Hén      NA        1080       1074           1080       6
10 Knud (4.) den Hellige NA        1086       1080           1086       6
# ... with 45 more rows
```

The data type isn't numeric and that is a problem when I will calculate the mean and median. But before changing the data type, I remove the NAs from the data set:

```
#Filtereing the NA columns away
HW2_monarchs <- HW2_monarchs %>% filter(years_ruled != "NA")
```

Now I can change the data type into numeric with the as.numeric() function:

```
#Changing the data into numeric data
HW2_monarchs$years_ruled <- as.numeric(HW2_monarchs$years_ruled)
class(HW2_monarchs$years_ruled)

# Calculating the mean

> class(HW2_monarchs$years_ruled)
[1] "numeric"
```

Now I will calculate the mean and median duration of rule over time with the mean() and the median() functions:

```
> mean(HW2_monarchs$years_ruled)
[1] 19.71698
> # Calculating the median
> median(HW2_monarchs$years_ruled)
[1] 17
```

The mean duration of rule over time is 19.72 years, and the median durations of rule over time is 17 years.

To find the three monarchs, who have been ruling the longest, I sorted the “years_ruled” column by size with the sort() function to be able to see the three greatest values, which was 60, 50 and 43 years:

```
> sort(HW2_monarchs$years_ruled)
[1] 2 2 3 4 4 5 6 6 6 6 7 7 8 8 9 9 9 9 10 10 10 11 11 11 15 16 17 20 20 22 25 25 25 25 27 27
[37] 27 29 29 29 30 31 31 32 33 33 35 35 39 42 43 50 60
>
```

Then I found out to which rows the three values belonged using the which() function:

```
> which(HW2_monarchs == 60, arr.ind=TRUE)
      row col
[1,]  39   6
> which(HW2_monarchs == 50, arr.ind=TRUE)
      row col
[1,]  53   6
> which(HW2_monarchs == 43, arr.ind=TRUE)
      row col
[1,]  49   6
```

Then I print row 39, 53 and 49 to find the names of the three monarchs I’m looking for.

```
> print(HW2_monarchs[39,])
# A tibble: 1 × 6
  danish_monarchs birth_year death_year reign_start_year reign_end_year years_ruled
<chr>           <chr>      <chr>      <chr>           <chr>           <dbl>
1 Christian 4.    1577      1648      1588           1648             60
> print(HW2_monarchs[53,])
# A tibble: 1 × 6
  danish_monarchs birth_year death_year reign_start_year reign_end_year years_ruled
<chr>           <chr>      <chr>      <chr>           <chr>           <dbl>
1 Margrethe 2.    1940      NA        1972           NA              50
> print(HW2_monarchs[49,])
# A tibble: 1 × 6
  danish_monarchs birth_year death_year reign_start_year reign_end_year years_ruled
<chr>           <chr>      <chr>      <chr>           <chr>           <dbl>
1 Christian 9.    1818      1906      1863           1906             43
```

The three monarchs with the longest duration of rule over time is Christian 4th, Margrethe 2nd and Christian 9th.

To find the number of days the three monarchs have ruled, I multiply the number of years by 365, because a year consists of 365 days in average. It must be noted that I haven’t taken leap years into account.

```
> #Christian 4th (row 39)
> 60*365
[1] 21900
> # Margrethe 2nd (row 53)
> 50*365
[1] 18250
> # Christian 9th (row 49)
> 43*365
[1] 15695
```

Christian 9th has ruled for 21900 days, Margrethe 2nd for 18250 days and Christian 9th for 15695 days.