Opgave uge 10

Gruppe: Silas Bramsen, Olivia Clara Munk-Olsen, Johann Destailleur, Freja Bille Deichmann og Laura Holm Løjborg

Load the kings

1. Look at the dataset that are you loading and check what its columns are separated by? (hint: open it in plain text editor to see)

List what is the

separator: The separator is a comma (,).

2. Create a `kings` object in R with the different functions below and inspect the different outputs.

- `read.csv()`
- `read_csv()`
- `read.csv2()`
- `read_csv2()`

Answer:

1. Which of these functions is a `tidyverse` function? Read data with it below into a `kings` object?

The one who is a tidyverse function is the one that R reads as a tibble, “tbl”
In this case with this particular dataset, the ones who are tidyverse functions are the ones that include _ rather than a full stop, so kings2 and kings4

```
# FILL IN THE CODE BELOW and review the outputs
library(tidyverse)
kings1 <- read.csv("data/kongerække.csv")
head(kings1)
glimpse(kings1)
class(kings1)

kings2 <- read_csv("data/kongerække.csv")
head(kings2)
glimpse(kings2)
class(kings2)

kings3 <- read.csv2("data/kongerække.csv")
head(kings3)
glimpse(kings3)
class(kings3)

kings4 <- read_csv2( "data/kongerække.csv", na="NA")
head(kings4)
glimpse(kings4)
class(kings4)
```

2. What is the result of running `class()` on the `kings` object created with a tidyverse function?

```
> class(kings2)
[1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
> class(kings4)
[1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

3. How many columns does the object have when created with these different functions?

By using the ncol() function the object (kings <- kings4) is revealed to have 8 columns.

4. Show the dataset so that we can see how R interprets each column.

The result of head(kings):

A tibble: 6 × 8

Navn <chr>	Fødselsår <dbl>	Dødsår <dbl>	Regering_Start <dbl>	Regering_Slut <dbl>	Usikkere_Årstal <chr>	Antal_regeringsår <dbl>	Slægt <chr>
Gorm_den_Gamle	*NA*	958	936	958	Ja	22	*NA*
Harald_1_Blåtand	*NA*	987	950	987	Ja	37	*NA*
Svend_1_Tveskæg	963	1014	987	1014	Ja	27	*NA*
Harald_2	*NA*	1018	1014	1018	Ja	4	*NA*
Knud_1_den_Store	995	1035	1018	1035	Nej	17	*NA*
Hardeknud	1020	1042	1035	1042	Nej	7	*NA*

6 rows

The result of tail(kings):

A tibble: 6 × 8

Navn <chr>	Fødselsår <dbl>	Dødsår <dbl>	Regering_Start <dbl>	Regering_Slut <dbl>	Usikkere_Årstal <chr>	Antal_regeringsår <dbl>	Slægt <chr>
Christian_9	1818	1906	1863	1906	Nej	43	Glücksborg
Frederik_8	1843	1912	1906	1912	Nej	6	Glücksborg
Christian_10	1870	1947	1912	1947	Nej	35	Glücksborg
Frederik_9	1899	1972	1947	1972	Nej	25	Glücksborg
Margrete_2	1940	*NA*	1972	2024	Nej	52	Glücksborg
Frederik_10	1968	*NA*	2024	*NA*	Nej	1	Glücksborg

6 rows

```
# COMPLETE THE BLANKS BELOW WITH YOUR CODE, then turn the 'eval' flag in this chunk to TRUE.

kings <- _____

class(kings)


____(kings)


____(kings)
```

Calculate the duration of reign for all the kings in your table

You can calculate the duration of reign in years with `mutate` function by subtracting the equivalents of your `startReign` from `endReign` columns and writing the result to a new column called `duration`. But first you need to check a few things:

- Is your data messy? Fix it before re-importing to R
- Do your start and end of reign columns contain NAs? Choose the right strategy to deal with them: `na.omit()`, `na.rm=TRUE`, `!is.na()`

> The dataset contains missing data, but by using the function na=”NA”, when loading the csv.file, it is already omitted

Create a new column called `duration` in the kings dataset, utilizing the `mutate()` function from tidyverse. Check with your group to brainstorm the options.

We start by selecting the names, start and end of reign, followed by a pipe. Then we mutate start and end of reign into a new column called duration

```
# YOUR CODE

kings_duration <- select(kings,Navn,Regering_Start,Regering_Slut) %>%
  mutate(duration=Regering_Slut-Regering_Start)
```

This is the result of the that code:

A tibble: 56 × 4

Navn <chr>	Regering_Start <dbl>	Regering_Slut <dbl>	duration <dbl>
Gorm_den_Gamle	936	958	22
Harald_1_Blåtand	950	987	37
Svend_1_Tveskæg	987	1014	27
Harald_2	1014	1018	4
Knud_1_den_Store	1018	1035	17
Hardeknud	1035	1042	7
Magnus_den_Gode	1042	1047	5
Svend_2_Estridsen	1047	1074	27
Harald_3_Hen	1074	1080	6
Knud_2_den_Hellige	1080	1086	6
Oluf_1_Hunger	1086	1095	9
Erik_1_Ejegod	1095	1103	8
Niels	1104	1134	30
Erik_2_Emune	1134	1137	3
Erik_3_Lam	1137	1146	9
Svend_3_Grathe	1146	1157	11
Knud_3	1146	1157	11
Valdemar_1_den_Store	1146	1157	11
Valdemar_1_den_Store	1157	1182	25
Knud_4	1182	1202	20
Valdemar_2_Sejr	1202	1241	39
Erik_4_Plovpenning	1241	1250	9
Abel	1250	1252	2
Christoffer_1	1252	1259	7
Erik_5_Klipping	1259	1286	27
Erik_6_Menved	1286	1319	33
Christoffer_2	1319	1326	7
Valdemar_3	1326	1329	3
Christoffer_2	1329	1332	3
Valdemar_4_Atterdag	1340	1375	35
Oulf_2	1375	1387	12
Margrete_1	1387	1396	9
Erik_7_af_Pommern	1396	1439	43
Christoffer_3_af_Bayern	1440	1448	8
Christian_1	1448	1481	33
Hans	1482	1513	31
Christian_2	1513	1523	10
Frederik_1	1523	1533	10
Christian_3	1536	1559	23
Frederik_2	1559	1588	29
Christian_4	1596	1648	52
Frederik_3	1648	1670	22
Christian_5	1670	1699	29
Frederik_4	1699	1730	31
Christian_6	1730	1746	16
Frederik_5	1746	1766	20

1-46 of 56 rows

A tibble: 56 × 4

Navn <chr>	Regering_Start <dbl>	Regering_Slut <dbl>	duration <dbl>
Christian_7	1766	1808	42
Frederik_6	1808	1839	31
Christian_8	1839	1848	9
Frederik_7	1848	1863	15
Christian_9	1863	1906	43
Frederik_8	1906	1912	6
Christian_10	1912	1947	35
Frederik_9	1947	1972	25
Margrete_2	1972	2024	52
Frederik_10	2024	*NA*	*NA*

Calculate the average duration of reign for all rulers

Do you remember how to calculate an average on a vector object? If not, review the last two lessons and remember that a column is basically a vector. So you need to subset your `kings` dataset to the `duration` column. If you subset it as a vector you can calculate average on it with `mean()` base-R function. If you subset it as a tibble, you can calculate average on it with `summarize()` tidyverse function. Try both ways!

- You first need to know how to select the relevant `duration` column. What are your options?
- Is your selected `duration` column a tibble or a vector? The `mean()` function can only be run on a vector. The `summarize()` function works on a tibble.
- Are you getting an error that there are characters in your column? Coerce your data to numbers with `as.numeric()`.
- Remember to handle NAs: `mean(X, na.rm=TRUE)`

Using the kings_duration object we created before, we select the duration and summarise it, and calling it “genm”

```
```{r}
# YOUR CODE
kings_duration %>%
  select(duration) %>%
  summarise(genm=mean(duration, na.rm = TRUE))
```

The result:

A tibble: 1 × 1

genm
<dbl>
19.92727

1 row

How many and which kings enjoyed a longer-than-average duration of reign?

You have calculated the average duration above. Use it now to `filter()` the `duration` column in `kings` dataset. Display the result and also count the resulting rows with `count()`

```
(We chose to name the object for rulers that ruled longer than average “OB” - over gennemsnittet)
27
Out method and result in R:
```
```

```r
# YOUR CODE

og <- kings_duration %>%
  filter(duration>19.92727)
count(og)

```

A tibble: 1 × 1

n
<int>
27

1 row

How many days did the three longest-ruling monarchs rule?

- Sort kings by reign `duration` in the descending order. Select the three longest-ruling monarchs with the `slice()` function
- Use `mutate()` to create `Days` column where you calculate the total number of days they ruled
- BONUS: consider the transition year (with 366 days) in your calculation!

```r
# YOUR CODE
arrange(og, duration) %>%
  slice(25:27)

days <- og %>%
  mutate(daysreign=duration*365) %>%
  arrange(daysreign)
```

The three longest-ruling rulers were:

25	Christian_9	1863	1906	43	15695
26	Christian_4	1596	1648	52	18980
27	Margrete_2	1972	2024	52	18980