Danish Kings Week 10

Mathias Dyhr Pedersen

2025-03-05

The task here is to load your Danish Monarchs csv into R using the `tidyverse` toolkit, calculate and explore the kings' duration of reign with pipes `%>%` in `dplyr` and plot it over time.

Load the kings

Make sure to first create an `.Rproj` workspace with a `data/` folder where you place either your own dataset or the provided `kings.csv` dataset.

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:komma

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()`

```
# FILL IN THE CODE BELOW and review the outputs

library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.4
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
kings1 <- read.csv("data/danish_kings_rigtig.csv")

kings2 <- read_csv("data/danish_kings_rigtig.csv")
```

```
## Rows: 56 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): regent
## dbl (4): fødselsår, dødsår, første_regeringsår, sidste_regeringsår
##
## 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.
```

```
kings3 <- read.csv2("data/danish_kings_rigtig.csv")

kings4 <- read_csv2( "data/danish_kings_rigtig.csv")
```

```
## ℹ Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
## Rows: 56 Columns: 1── Column specification ────────────────────────────────────────────────
──────────
## Delimiter: ";"
## chr (1): regent ,fødselsår,dødsår,første_regeringsår,sidste_regeringsår
## 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.
```

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

the “read_csv” functions are tivyverse functions as they reveal a “tibble”

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

When I ran the functions with “.” it reveals a data frame, whereas the functions with “_” reveals a tibble. And further, the dataset recognizes the data as characters when using the “read.csv” functions, whereas the “read_csv” functions recognizes the data as digits.

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

The objects which I created using “read.csv” and “read_csv” contains 5 columns because my data is seperated by commas. However the objects which I created using “read.csv2” and “read_csv2) contains 1 column because my data is not seperated by semicolons.

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

See following functions below

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

kings <- kings2

class(kings)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

```
glimpse(kings)
```

```
## Rows: 56
## Columns: 5
## $ regent            <chr> "gorm_den_gamle", "harald_1_blåtand", "svend_1_tves…
## $ fødselsår         <dbl> 908, NA, NA, NA, 995, 1020, 1024, 1047, NA, NA, NA,…
## $ dødsår            <dbl> 958, 987, 1014, 1018, 1035, 1042, 1047, 1076, 1080,…
## $ første_regeringsår <dbl> 936, 958, 987, 1014, 1018, 1035, 1042, 1047, 1074, …
## $ sidste_regeringsår <dbl> 958, 987, 1014, 1018, 1035, 1042, 1047, 1074, 1080,…
```

```
head(kings)
```

```
## # A tibble: 6 × 5
##   regent           fødselsår dødsår første_regeringsår sidste_regeringsår
##   <chr>                <dbl>  <dbl>              <dbl>              <dbl>
## 1 gorm_den_gamle         908    958                936                958
## 2 harald_1_blåtand        NA    987                958                987
## 3 svend_1_tveskæg         NA   1014                987               1014
## 4 harald_2                NA   1018               1014               1018
## 5 knud_1_den_store       995   1035               1018               1035
## 6 hardeknud             1020   1042               1035               1042
```

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()`

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

```
kings_no_na <- kings2 %>%
  filter(!is.na(sidste_regeringsår))

kings_duration <- kings_no_na %>%
  mutate(duration = sidste_regeringsår - første_regeringsår)
```

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)`

```
kings_no_na <- kings2 %>%
  filter(!is.na(sidste_regeringsår))

kings_duration <- kings_no_na %>%
  mutate(duration = sidste_regeringsår - første_regeringsår)

kings_average_duration <- kings_duration$duration %>%
  mean( ,kings_duration$duration)
```

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()`

```
kings_long_regin <- kings_duration %>%
  filter(duration>kings_average_duration)

count(kings_long_regin)
```

```
## # A tibble: 1 × 1
##       n
##   <int>
## 1    26
```

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!

```
kings_longest_reign <- kings_duration %>%
  select(duration, regent) %>%
  arrange(desc(duration)) %>%
  slice_max(order_by = duration, n = 3)

kings_longest_reign %>%
  mutate(days = duration * 365 + floor(duration / 4))
```

```
## # A tibble: 3 × 3
##   duration regent             days
##      <dbl> <chr>             <dbl>
## 1       52 margrethe_2       18993
## 2       43 erik_7_af_pommern 15705
## 3       43 christian_9       15705
```

Challenge: Plot the kings’ duration of reign through time

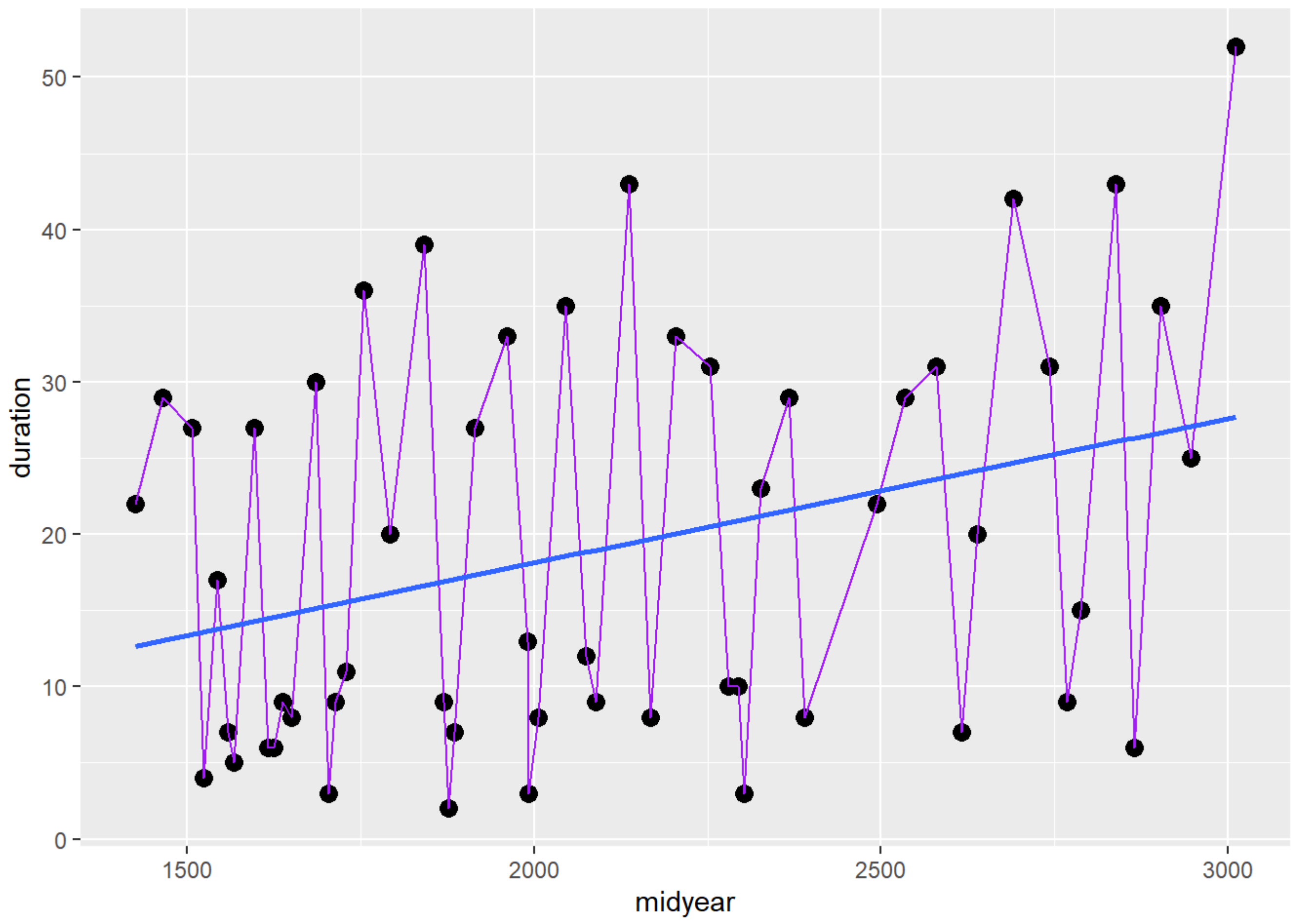
What is the long-term trend in the duration of reign among Danish monarchs? How does it relate to the historical violence trends ?

- Try to plot the duration of reign column in `ggplot` with `geom_point()` and `geom_smooth()`
- In order to peg the duration (which is between 1-99) somewhere to the x axis with individual centuries, I recommend creating a new column `midyear` by adding to `startYear` the product of `endYear` minus the `startYear` divided by two (`startYear + (endYear-startYear)/2`).
- Now you can plot the kings dataset, plotting `midyear` along the x axis and `duration` along y axis
- BONUS: add a title, nice axis labels to the plot and make the theme B&W and font bigger to make it nice and legible!

```
kings_graph <- kings_duration %>%
  mutate(midyear = første_regeringsår + (sidste_regeringsår - første_regeringsår / 2))

ggplot(kings_graph, mapping = aes(x = midyear, y = duration))+
  geom_point(size = 3)+
  geom_line(colour = "purple")+
  geom_smooth(method = lm, se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



And to submit this rmarkdown, knit it into html. But first, clean up the code chunks, adjust the date, rename the author and change the `eval=FALSE` flag to `eval=TRUE` so your script actually generates an output. Well done!