Portfolio 2

Sigurd Sørensen

2022-08-31

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 0.3.5   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

## 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) Does OpenRefine alter the raw data during sorting and filtering? - You never change the raw data but it saves a log of each new itteration and all its changes. (But I like to work in R xD xD xD )

kings <- c("Margrethe 2.  
Siden 1972  
  
Frederik 9.  
1947 - 1972  
  
Christian 10.  
1912-1947  
  
Frederik 8.  
1906-1912  
  
Christian 9.  
1863-1906  
  
Frederik 7.  
1848-1863  
  
Christian 8.  
1839-1848  
  
Frederik 6.  
1808-1839  
  
Christian 7.  
1766-1808  
  
Frederik 5.  
1746-1766  
  
Christian 6.  
1730-1746  
  
Frederik 4.  
1699-1730  
  
Christian 5.  
1670-1699  
  
Frederik 3.  
1648-1670  
  
Christian 4.  
1588-1648  
  
Frederik 2.  
1559-1588  
  
Christian 3.  
1536-1559  
  
Interregnum  
1533-1536  
  
Frederik 1.  
1523-1533  
  
Christian 2.  
1513-1523  
  
Hans  
1482-1513  
  
Christian 1.  
1448-1481  
  
Christoffer 3. af Bayern  
1440-1448  
  
Erik 7. af Pommern  
1396-1439  
  
Margrete 1.  
1387-1396  
  
Oluf 2.  
1375-1387  
  
Valdemar 4. Atterdag  
1340-1375  
  
Interregnum  
1332-1340  
  
Christoffer 2.  
1329-1332  
  
Valdemar 3.  
1326-1329  
  
Christoffer 2.  
1319-1326  
  
Erik 6. Menved  
1286-1319  
  
Erik 5. Klipping  
1259-1286  
  
Christoffer 1.  
1252-1259  
  
Abel  
1250-1252  
  
Erik 4. Plovpenning  
1241-1250  
  
Valdemar 2. Sejr  
1202-1241  
  
Knud 4.  
1182-1202  
  
Valdemar 1. den Store  
1157-1182  
  
Svend 3., Knud 3., Valdemar 1.  
1146-1157  
  
Erik 3. Lam  
1137-1146  
  
Erik 2. Emune  
1134-1137  
  
Niels  
1104-1134  
  
Erik 1. Ejegod  
1095-1103  
  
Oluf 1. Hunger  
1086-1095  
  
Knud 2. den Hellige  
1080-1086  
  
Harald 3. Hen  
1074-1080  
  
Svend 2. Estridsen  
1047-1074  
  
Magnus den Gode  
1042-1047  
  
Hardeknud  
1035-1042  
  
Knud 1. den Store  
1018-1035  
  
Harald 2.  
1014-1018  
  
Svend 1. Tveskæg  
D. 1014  
  
Harald 1. Blåtand  
D. senest 987  
  
Gorm den Gamle.  
936, d. ca. 958")  
  
  
kings <- str\_split(kings,pattern = "\n")

We do a bit of cleaning magic.

df\_kings <- data.frame(Names = kings[[1]][seq(from = 1, to = length(kings[[1]]), by = 3 )], Year = kings[[1]][seq(from = 2, to = length(kings[[1]]), by = 3 )])  
df\_kings

## Names Year  
## 1 Margrethe 2. Siden 1972  
## 2 Frederik 9. 1947 - 1972  
## 3 Christian 10. 1912-1947  
## 4 Frederik 8. 1906-1912  
## 5 Christian 9. 1863-1906  
## 6 Frederik 7. 1848-1863  
## 7 Christian 8. 1839-1848  
## 8 Frederik 6. 1808-1839  
## 9 Christian 7. 1766-1808  
## 10 Frederik 5. 1746-1766  
## 11 Christian 6. 1730-1746  
## 12 Frederik 4. 1699-1730  
## 13 Christian 5. 1670-1699  
## 14 Frederik 3. 1648-1670  
## 15 Christian 4. 1588-1648  
## 16 Frederik 2. 1559-1588  
## 17 Christian 3. 1536-1559  
## 18 Interregnum 1533-1536  
## 19 Frederik 1. 1523-1533  
## 20 Christian 2. 1513-1523  
## 21 Hans 1482-1513  
## 22 Christian 1. 1448-1481  
## 23 Christoffer 3. af Bayern 1440-1448  
## 24 Erik 7. af Pommern 1396-1439  
## 25 Margrete 1. 1387-1396  
## 26 Oluf 2. 1375-1387  
## 27 Valdemar 4. Atterdag 1340-1375  
## 28 Interregnum 1332-1340  
## 29 Christoffer 2. 1329-1332  
## 30 Valdemar 3. 1326-1329  
## 31 Christoffer 2. 1319-1326  
## 32 Erik 6. Menved 1286-1319  
## 33 Erik 5. Klipping 1259-1286  
## 34 Christoffer 1. 1252-1259  
## 35 Abel 1250-1252  
## 36 Erik 4. Plovpenning 1241-1250  
## 37 Valdemar 2. Sejr 1202-1241  
## 38 Knud 4. 1182-1202  
## 39 Valdemar 1. den Store 1157-1182  
## 40 Svend 3., Knud 3., Valdemar 1. 1146-1157  
## 41 Erik 3. Lam 1137-1146  
## 42 Erik 2. Emune 1134-1137  
## 43 Niels 1104-1134  
## 44 Erik 1. Ejegod 1095-1103  
## 45 Oluf 1. Hunger 1086-1095  
## 46 Knud 2. den Hellige 1080-1086  
## 47 Harald 3. Hen 1074-1080  
## 48 Svend 2. Estridsen 1047-1074  
## 49 Magnus den Gode 1042-1047  
## 50 Hardeknud 1035-1042  
## 51 Knud 1. den Store 1018-1035  
## 52 Harald 2. 1014-1018  
## 53 Svend 1. Tveskæg D. 1014  
## 54 Harald 1. Blåtand D. senest 987  
## 55 Gorm den Gamle. 936, d. ca. 958

## 2

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?”

df\_random <- read\_csv("SAFI\_openrefine.csv")

## Rows: 131 Columns: 62  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (45): interview\_date, province, district, ward, village, agr\_assoc, rem...  
## dbl (15): quest\_no, years\_farm, no\_membrs, \_members\_count, years\_liv, build...  
## dttm (2): start, end  
##   
## 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.

df\_random\_long <- df\_random %>%   
 separate\_rows(months\_no\_water, sep = ";") %>%   
 filter(months\_no\_water != "NULL")   
  
#Delete non-alphanumeric.  
df\_random\_long <- mutate(df\_random\_long, months\_no\_water = str\_replace\_all(df\_random\_long$months\_no\_water, "[^[:alnum:]]", ""))

df\_random\_long$months\_no\_water

## [1] "Aug" "Sept" "Aug" "Sept" "Oct" "Sept" "Oct" "Oct" "Nov" "Sept"  
## [11] "Oct" "Nov" "Aug" "Sept" "Oct" "Nov" "Oct" "Sept" "Oct" "Nov"   
## [21] "Oct" "Sept" "Oct" "Aug" "Sept" "Oct" "Nov" "Aug" "Sept" "Oct"   
## [31] "Nov" "Sept" "Oct" "Nov" "Sept" "Oct" "Aug" "Sept" "Oct" "Oct"   
## [41] "Nov" "Sept" "Oct" "Nov" "Sept" "Oct" "Sept" "Oct" "Aug" "Sept"  
## [51] "Sept" "Oct" "Aug" "Sept" "Oct" "Sept" "Oct" "Nov" "Sept" "Oct"   
## [61] "Sept" "Oct" "Nov" "Aug" "Sept" "Oct" "Aug" "Sept" "Oct" "Nov"   
## [71] "Dec" "Sept" "Oct" "Aug" "Sept" "Oct" "Aug" "Sept" "Oct" "Sept"  
## [81] "Oct" "Nov" "Aug" "Sept" "Oct" "Oct" "Nov" "Nov" "Dec" "Nov"   
## [91] "Dec" "Aug" "Sept" "Sept" "Oct" "Nov" "Nov" "Sept" "Oct" "Nov"   
## [101] "Oct" "Nov" "Oct" "Nov" "Aug" "Sept" "Oct" "Nov" "Sept" "Oct"   
## [111] "Sept" "Oct" "Nov" "Sept" "Oct" "Sept" "Oct" "Nov" "Sept" "Oct"   
## [121] "Nov" "Oct" "Nov" "Aug" "Sept" "Oct" "Nov" "Jan" "Dec" "Aug"   
## [131] "Sept" "Oct" "Nov" "Dec" "Aug" "Sept" "Oct" "Nov" "Sept" "Oct"   
## [141] "Nov" "Sept" "Oct" "Aug" "Sept" "Oct" "Sept" "Oct" "Sept" "Oct"   
## [151] "Sept" "Oct" "Nov" "Aug" "Sept" "Oct" "Nov" "Aug" "Sept" "Oct"   
## [161] "Aug" "Sept" "Sept" "Oct" "Nov" "Aug" "Sept" "Aug" "Sept" "Oct"   
## [171] "Nov" "Aug" "Sept" "Oct" "Nov" "Aug" "Sept" "Oct" "Nov" "Sept"  
## [181] "Oct" "Aug" "Sept" "Oct" "Nov" "Sept" "Oct" "Nov" "Sept" "Oct"   
## [191] "Nov" "Sept" "Oct" "Nov" "Apr" "May" "June" "July" "Aug" "Sept"  
## [201] "Oct" "Nov" "Jan" "Dec" "Aug" "Sept" "Oct" "Nov" "Dec" "Aug"   
## [211] "Sept" "Oct" "July" "Aug" "Sept" "Oct" "Nov" "Dec" "Sept" "Oct"   
## [221] "Nov" "Dec" "Oct" "Nov" "Aug" "Sept" "Oct" "Nov" "Dec" "Sept"  
## [231] "Oct" "Nov" "Sept" "Nov" "Sept" "Oct" "Nov" "Oct" "Nov" "Dec"   
## [241] "Aug" "Sept" "Oct" "Sept" "Oct" "Nov"

unique(df\_random\_long$months\_no\_water) #Sanity check to see if all is tidy.

## [1] "Aug" "Sept" "Oct" "Nov" "Dec" "Jan" "Apr" "May" "June" "July"

df\_random\_long %>%   
 filter(months\_no\_water != "NULL") %>%   
 count(months\_no\_water) %>%   
 arrange(desc(n))

## # A tibble: 10 × 2  
## months\_no\_water n  
## <chr> <int>  
## 1 Oct 74  
## 2 Sept 70  
## 3 Nov 51  
## 4 Aug 33  
## 5 Dec 11  
## 6 Jan 2  
## 7 July 2  
## 8 Apr 1  
## 9 June 1  
## 10 May 1

October and September is the two months reported the most times as being without water.

## 3

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)

df\_aarhus <- read\_csv("https://raw.githubusercontent.com/aarhusstadsarkiv/datasets/master/censuses/1801/census-1801-normalized.csv")

## Warning: One or more parsing issues, call `problems()` on your data frame for details,  
## e.g.:  
## dat <- vroom(...)  
## problems(dat)

## Rows: 44559 Columns: 17  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (11): sogn, amt, lokalitet, bygning, fnavn, enavn, koen, famstand, civil...  
## dbl (6): ft, id, loknr, famnr, alder, giftnr  
##   
## 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.

df\_aarhus %>%   
 filter(!is.na(erhverv)) %>%   
 count(erhverv) %>%   
 head()

## # A tibble: 6 × 2  
## erhverv n  
## <chr> <int>  
## 1 ?? 1  
## 2 [Snedker] 1  
## 3 [Vanfør, nyder Almisse] 1  
## 4 2. lectie hører 1  
## 5 3. lectie hører 1  
## 6 4. lectie hører 1

The data is still super messy so let us try and clean it up a bit.

df\_aarhus <- df\_aarhus %>%   
 filter(!is.na(erhverv))  
  
df\_aarhus %>%   
 mutate(erhverv = str\_to\_lower(gsub('[[:punct:]]', '', df\_aarhus$erhverv))) %>%  
 count(erhverv) %>%   
 filter(n > 3) %>%   
 arrange(desc(n)) %>%   
 head()

## # A tibble: 6 × 2  
## erhverv n  
## <chr> <int>  
## 1 bonde og gaardbeboer 2012  
## 2 huusmand med jord 749  
## 3 bonde og gårdbeboer 223  
## 4 soldat ved 1 jyske inf reg 136  
## 5 nyder ophold af gaarden 113  
## 6 nationalsoldat 110

We could now stem all words and use a danish stopword list to clean it even further. That requires a library and a stopword list that works with the danish languages.