Sofie Schmidt Madsen
Emilie Andersen
Kristiane Morbech Clausen

Portfolio – Delopgave 1
Regex-Openrefine

Digitale Arkiver og Metode: Hold 2, Forår 2025

Deklaration

Vi gør desuden opmærksom på, at opgavens besvarelse ikke er præget af individuelle indsatser, men er en gruppebesvarelse af følgende gruppemedlemmer; Sofie Schmidt Madsen, Emilie Andersen og Kristiane Morbech Clausen, hvor alle har bidraget i fællesskab til opgaven.

URL til GitHub Respository

https://github.com/Digital-Methods-HASS/Sofie_Kristiane_Emilie_Eksamen/tree/main/regex_openrefine_assignment

Besvarelse

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 ?

Datoer til formatet: YYYY-MM-DD

Vi har ved brug af librarycarpentry fået en oversigt og forklaring på de forskellige regex-metategn for at kunne løse opgaven, samt fået forklaringer på hjemmesiden regex101.com.[1] Til opgaven har vi også fået hjælp til at fejlrette af ChatGPT. [2]

Trin 1:
Vi har brugt Regular Expression: **(\d{1,2}).(\d{1,2}).\s?(\d+)**
til at 'fange' dato, måned og år i separate grupper.

Forklaringer:
(\d{1,2}): Matcher 1-2 cifre (dag).

.: Matcher et punktum, skråstreg eller bindestreg som separator.

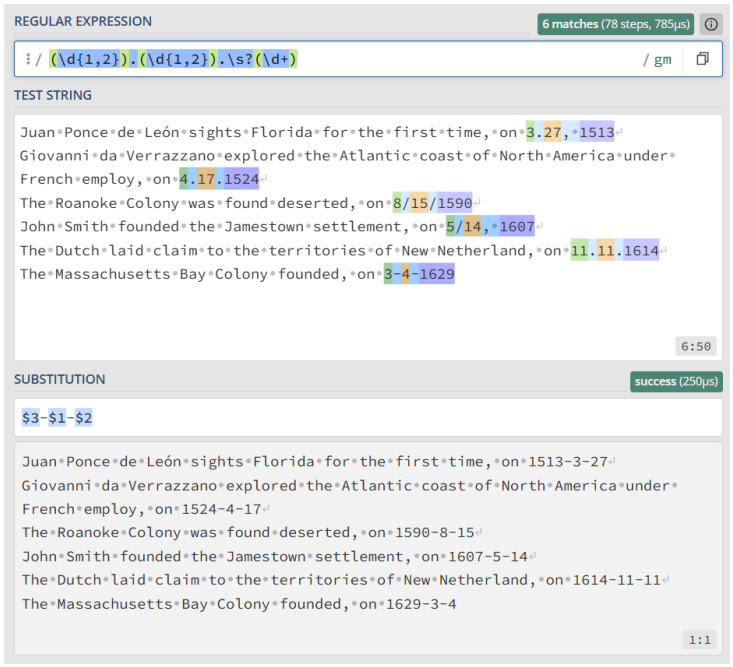
(\d{1,2}): Matcher 1-2 cifre (måned).

\s?: Matcher evt. et mellemrum.

(\d+): Matcher et eller flere cifre (år).

Trin 2:
Derefter bruges substitutionen **$3-$1-$2** til at omrangere datoerne til formatet YYYY-MM-DD.

[1] Se forklaringen på linket: https://regex101.com/r/06vj25/1 – klik på fanen **"Explanation"** under regex-feltet for at se, hvad udtrykket betyder.
[2] Library Carpentry, "Regular Expressions," *Library Carpentry: Introduction to Working with Data*, opdateret 30. september 2024, https://librarycarpentry.github.io/lc-data-intro/01-regular-expressions.html#regular-expressions (tilgået 24. maj 2025)



Link: https://regex101.com/r/06vj25/1

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)

Voyant to R

Vi har ligeledes i denne opgave gjort brug af librarycarpentry, hvor vi har fået en oversigt og forklaring på de forskellige regex-metategn for at kunne løse opgaven, samt fået forklaringer på hjemmesiden regex101.com.[3] Til opgaven har vi også fået hjælp til at fejlrette af ChatGPT.[4]

Trin 1:

Vi har brugt det Regular Expression: **([\wæøåé'"".]+)(\n|$)**

Samlet forklaring:

[3] Se forklaringerne på disse links: https://regex101.com/r/dgCAAo/2 og https://regex101.com/r/LETDkl/1 – klik på fanen **"Explanation"** under regex-feltet for at se, hvad udtrykket betyder.
[4] Library Carpentry, "Regular Expressions,"

([\wæøåé'’".]+): Matcher et ord (som kan indeholde givende danske tegn og specielle tegn).
(\n|$): Matcher enten et linjeskift eller slutningen af strengen, som betyder, at ordet er det sidste i teksten.

Trin 2:
Herefter har vi skrevet i substitution **“$1”,**
For at konvertere stopwordlisten i Voyant til en liste i Regex101, som består af “ord” adskilt af kommaer.

$1: Er alle ordene i stopwordlisten, der blev ‘fanget i en gruppe’ ved at bruge den første del i vores regular expression: ([\wæøåé'’".]+)
“”: Er skrevet omkring $1, for at sætte ordene i citationstegn.
„: Er skrevet efter “$1” for at adskille ordene med komma.

Trin 3:
For at fjerne det sidste komma efter det sidste ord, kopiere vi nu resultatet af Trin 2 ind i en ny Regexfil i feltet ‘Test string’.

Herefter skriver vi i feltet regular expression: **„$**

„$: fanger det sidste komma efter det sidste ord.

Trin 4:
For at fjerne det sidste komma, skal vi nu lade feltet substitution stå tomt.

Nu har vi slutresultatet, hvor vi har konverteret stopwordlisten i Voyant til en Regex stopwordlist.
Det helt færdige slutresultat kan ses i ‘Link - step 3 og 4’.

Link - step 1 og 2: https://regex101.com/r/dgCAAo/2
Link - step 3 og 4: https://regex101.com/r/LETDkl/1

R to Voyant
Vi har ligeledes i denne opgave gjort brug af librarycarpentry, hvor vi har fået en oversigt og forklaring på de forskellige regex-metategn for at kunne løse opgaven, samt fået forklaringer på hjemmesiden regex101.com.[5] Til opgaven har vi også fået hjælp til at fejlrette af ChatGPT.[6] **Trin 1:** Vi har brugt det Regular Expression: **,\s*** ,: Matcher et komma. \s*: Matcher eventuelle mellemrum (eller ingen mellemrum). Da der kan være mellemrum efter kommaet. **Trin 2:** Herefter har vi skrevet i substitution: **\n** For at konvertere stopwordlisten i Regex101 til en stopwordliste til Voyant. \n: erstatter komma og evt. mellemrum med et linjeskift. Så man får en 'kolonneliste'. Nu har vi slutresultatet, hvor vi har konverteret stopwordlisten i Regex til en stopwordlist i Voyant (Se link)
Link: https://regex101.com/r/pZzh2C/1

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

Nej, OpenRefine ændrer ikke på selve den rå data, men gør det muligt at sortere og filtrere den.

[5] Se forklaringen på linket: https://regex101.com/r/pZzh2C/1 – klik på fanen **"Explanation"** under regex-feltet for at se, hvad udtrykket betyder.
[6] Library Carpentry, "Regular Expressions,"

4. Fix the interviews dataset in OpenRefine enough to answer this question: "Which two months are reported as the most water-deprived/dryest by the interviewed farmer households?"

De to tørreste måneder

Trin 1
Vi indlæste filen til OpenRefine.

Trin 2
Dernæst valgte vi at trykke på kategorien 'months_no_water'

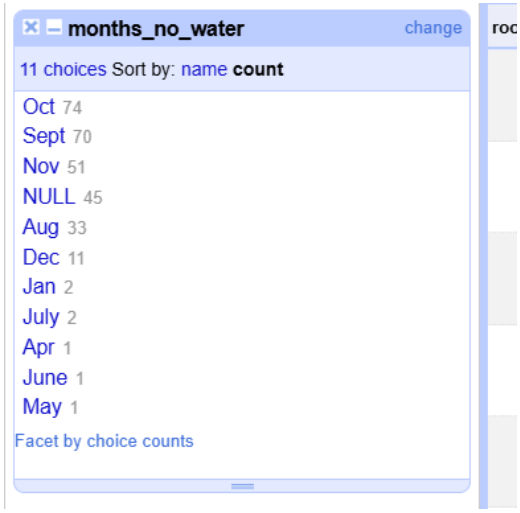
Og trykkede på 'Edit Cells' og derefter 'Transform'

Trin 3:
Derefter bruger vi GREL-expression:
value.replace("[", "").replace("]","").replace("'","").replace(" ","")

Trin 4:
Herefter trykker man i samme kategori på 'Facet' efterfulgt af 'Custome text facet'.

Trin 5:
Derefter bruger vi Grel-expression:
value.split(";")

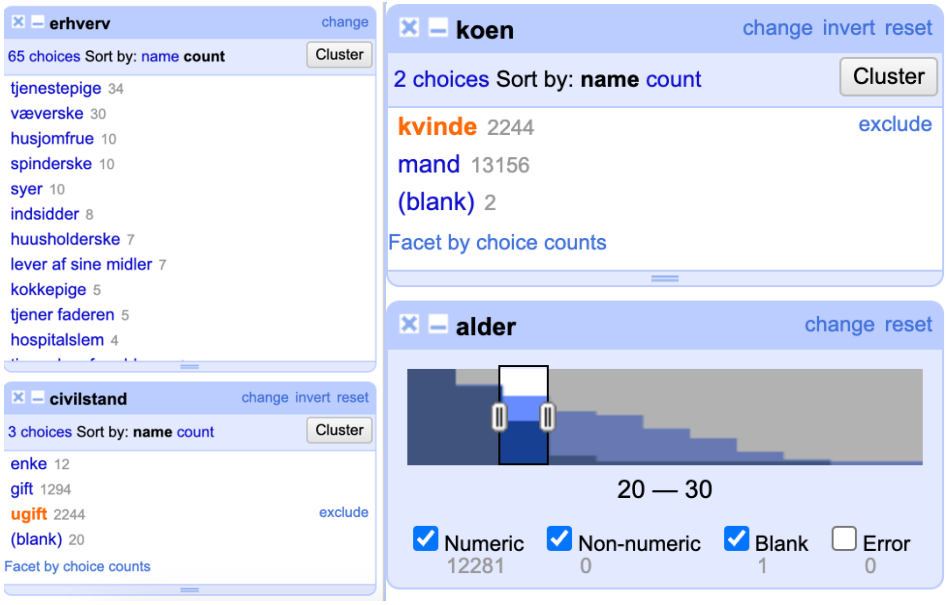
Trin 6:
Herefter har vi sorteret listen ud fra 'count'. Og derfor kan vi på baggrund af de interviewede landmænd se, at september og oktober er de to tørreste måneder, som vist nedenfor.



Bilag:
CSV-fil: Assignment_Week_8_Delopgave_4
Jason-fil: Assigment_Week_8_Delopgave_4.json

5. Real-Data Challenge: What are the 10 most frequent occupations "erhverv" among unmarried men or women of 20-30 years in 1801 Aarhus census dataset? (hint: first select either men or women to shrink the dataset to a manageable size, then filter by age, and then use merging to cut the erhvervvariation ruthlessly.)

Kvinder



Trin 1
Vi indlæste filen til OpenRefine.

Dernæst valgte vi at trykke på kategorien "koen", i sidepanelet valgte vi kønnet "kvinde", derved blev alle "mænd" og "(blank)" frasorteret.

Trin 2

Vi valgte at trykke på kategorien alder, og trykkede dernæst på "Edit cells" for videre at trykke på "Common Transforms", og valgte til sidst "To number".

Dernæst justeres aldersgrænsen til 20-30 år i sidepanelet alder.

Trin 3

Vi valgte at trykke på kategorien civilstand, i sidepanelet valgte vi civilstanden "ugift", derved blev alle "enke", "gift" og "(blank)" frasorteret.

Trin 4

Vi valgte at trykke på kategorien erhverv, trykkede på" Facet" til "Text facet".

I sidepanelet trykkes på "Cluster", hvilket åbner et nyt vindue.

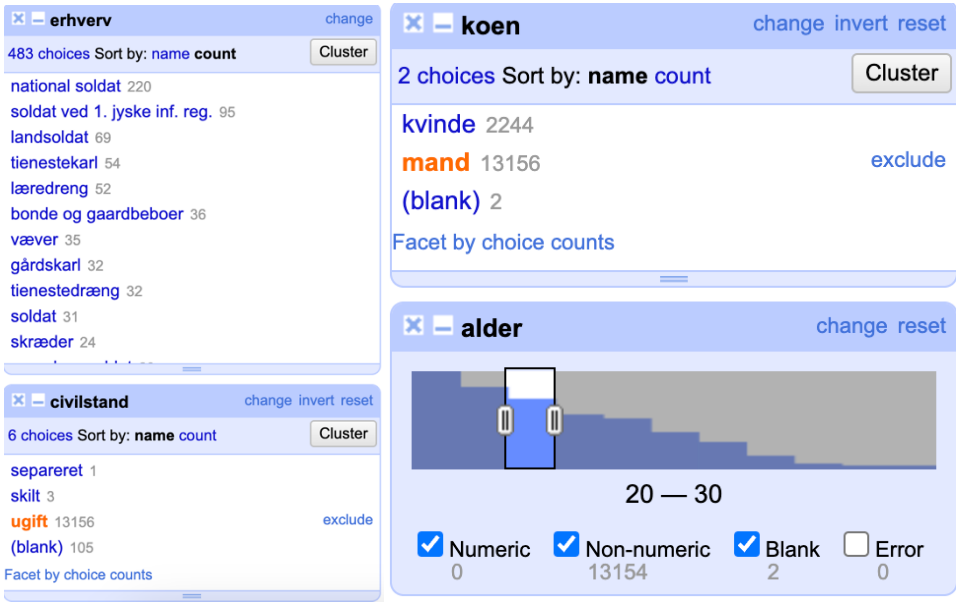
I dette vindue trykkes på "Keying function", og valgte "Metaphone3", trykkede dernæst "Cluster".

Dernæst foretog vi en grov frasortering af sideerhverv, og stod tilbage med hovederhvervet.

Ulempen ved dette er, at sideerhverv ikke bliver nævnt og dermed giver det nogle mørketal, hvor overlap forekommer.

CSV-fil: Assignment_Week_8_Delopgave_5.csv
Jason-fil: Assignment_Week_8_Delopgave_5.json

Mænd



Trin 1
Vi indlæste filen til OpenRefine.

Dernæst valgte vi at trykke på kategorien "koen", i sidepanelet valgte vi kønnet "Mænd", derved blev alle "kvinde" og "(blank)" frasorteret.

Trin 2
Vi valgte at trykke på kategorien alder, og trykkede dernæst på "Edit cells" for videre at trykke på "Common Transforms", og valgte til sidst "To number".

Dernæst justeres aldersgrænsen til 20-30 år i sidepanelet alder.

Trin 3
Vi valgte at trykke på kategorien civilstand, i sidepanelet valgte vi civilstanden "ugift", derved blev alle "enke", "gift" og "(blank)" frasorteret.

Trin 4
Vi valgte at trykke på kategorien erhverv, trykkede på" Facet" til "Text facet".

I sidepanelet trykkes på "Cluster", hvilket åbner et nyt vindue.

I dette vindue trykkes på "Keying function", og valgte "Metaphone3", trykkede dernæst "Cluster". Dernæst foretog vi en grov frasortering af sideerhverv, og stod tilbage med hovederhvervet. Ulempen ved dette er, at sideerhverv ikke bliver nævnt og dermed giver det nogle mørketal, hvor overlap forekommer.
CSV-fil: Assignment_Week_8_Delopgave_5.csv **Jason-fil:** Assignment_Week_8_Delopgave_5.json

Litteraturliste

Library Carpentry, "Regular Expressions," *Library Carpentry: Introduction to Working with Data*, opdateret 30. september 2024, https://librarycarpentry.github.io/lc-data-intro/01-regular-expressions.html#regular-expressions (tilgået 24. maj 2025)