Task 1: Documentation

Diego Kolzowski

The following document will present a brief summary of some key-points from the first-task's workflow.

First objective

absolute number of articles for the given year

Counting criteria

- *first author counting* considers only the first author of each publication. This means that each publication is considered only once.
- *whole counting* give one credit to every author of each publication. This means that each publication is consider as many times as authors has.
- *whole-normalized counting* consider all authors but distributes one credit between them equally distributed. This means that all the authors are consider but each publication is considered as one credit.
- *complete-normalized counting* consider all authors and distributes one credit per publication, but in an unequally distributed way.

Given that the goal of the task is to count how many articles where published for Germany in 2010 for STEM, I consider that the *first author counting* is the best criteria. The final count of the *whole-normalized counting* and the *complete-normalized counting* would give the same result. *whole counting* is not useful as it would inflate the result we are interested in.

To achieve this result I remove the duplicated rows by `ut`.

Result

```
>> 74286
```

Second objective

clean and re-code the variable "organization"

First, to identify all articles which have at least one author who is affiliated to a university I need to remove those rows where no organization is defined.

Then, I remove all numbers, punctuation marks, accents, etc. For this, I define the following cleaning function:

```
text_cleaner <- function(x){
  #replace numbers
  x <-  stringr::str_replace_all(x, stringr::regex("[0-9]*"),"")
  #replace replace punctuation
  x <- stringr::str_replace_all(x,stringr::regex("(\\+|\\-|\\=|\\:|;|\\.|,|_|\\?|¿|\\!|¡|\\\\|\\(|\\)|\
  #replace repeted line breaks and carriage returns
  x <- stringr::str_replace_all(x,'(\r\n)|(\n\r)','\n') %>%
    stringr::str_replace_all('\n+','\n') %>%
    stringr::str_replace_all('\r+','\r') %>%
```

```
    stringr::str_replace_all('(\r\n)|(\n\r)','\n') %>%
    stringr::str_replace_all('\n+','\n')
  #to lowercase
  x <- str_to_lower(x)
  #remove accents
  x <- stringi::stri_trans_general(x,"Latin-ASCII")
  return(x)
}
```

After this, I need to find if their organization is a university or a private institution. A brief inspection of the dataset shows that the keyword for universities is *'univ'*. I filter the results that contain 'univ' as part of the organization name (after cleaning)

In order to address the problem of multiple names, I summarize the information by suborganization and city:

```
##    suborganizations   city     len_org orgs
##    <chr>              <chr>      <int> <chr>
##  1 Inst Med Virol     frankfu~      12 clin goethe univ  | goethe univ fra~
##  2 Dept Surg          munich         9 klinikum univ munchen  | lm univ mu~
##  3 Dept Urol          munich         8 klinikum univ muenchen  | klinikum ~
##  4 Dept Internal Med~ regensb~       7 hosp univ regensburg  | regensburg ~
##  5 Inst Klin Radiol   munich         7 klinikum ludwig maximilians univ mu~
##  6 Inst Pathol        freiburg       7 univ freiburg  | univ freiburg klin~
##  7 Inst Pathol        mainz          7 johannes gutenberg univ hosp  | joh~
##  8 Dept Cardiovasc S~ freiburg       6 med univ clin  | univ clin freiburg~
##  9 Dept Chem          munich         6 lmu univ munich  | ludwig maximilia~
## 10 Dept Diagnost Rad~ freiburg       6 med phys univ hosp  | univ freiburg~
## # ... with 23,415 more rows
```

The majority of the dataset (91% of the suborganization & city pairs) have one Organization per suborganization & city.

For the rest of the dataset where the suborganization is defined, the organization's names only repeat few times and by a brief inspection, they all refer to the same institution. This means, the couple suborganization & city can be used for unifying the organization name field [1]

When we analyse the data with no suborganizations, we found much more repetition by city and it is not clear that they all belong to the same organization:

```
##    suborganizations city     len_org orgs
##    <chr>            <chr>      <int> <chr>
##  1 <NA>             munich        41 bw univ munich  | chirurg univ klin  ~
##  2 <NA>             hamburg       35 dvgw forsch stelle tech univ hamburg ~
##  3 <NA>             heidelb~      35 chirurg univ klin  | heidelberg univ ~
##  4 <NA>             berlin        25 alice salomon univ appl sci  | benjam~
##  5 <NA>             freiburg      25 anasthesiol univ klin freiburg  | chi~
##  6 <NA>             tubingen      23 childrens univ hosp  | orthopad univ ~
##  7 <NA>             frankfu~      20 clin johann wolfgang goethe univ  | g~
##  8 <NA>             leipzig       20 hno univ klin leipzig  | inst univ le~
##  9 <NA>             ulm           20 orthopad univ klinikum rku  | univ ap~
## 10 <NA>             essen         19 duisburg essen univ  | folkwang univ ~
## # ... with 144 more rows
```

[1] note: for a real workflow situation a much more careful analysis should be made in order to avoid unifying different organizations

Unification of the organization-label

The proposed workflow is the following:

1. For the data with suborganization & city: count the number of times each organization name is used. When there is a tie, I will choose the shortest one.
2. Define a codebook which associates the most used name with the others.
3. recode the names based on the codebook.

The *codebook* looks like the following table:

```
## # A tibble: 1,328 x 2
##    organization                          organization_new
##    <chr>                                 <chr>
##  1 free univ berlin                      free univ berlin
##  2 med univ klin freiburg                med univ klin freiburg
##  3 johannes gutenberg univ mainz         johannes gutenberg univ mainz
##  4 chirurg univ klin freiburg            univ freiburg
##  5 univ freiburg                         univ freiburg
##  6 univ klinikum freiburg i br           univ freiburg
##  7 univ klinikum heidelberg              univ klinikum heidelberg
##  8 univ marburg                          univ marburg
##  9 univ klin kinder & jugendmed tubing~ univ klin kinder  jugendmed tubing~
## 10 univ klin kinder & jugendmed          univ klin kinder  jugendmed
## # ... with 1,318 more rows
```

I need to add those organizations that don't appear in the codebook because they don't have suborganization. For this, I join the codebook with the original dataset, and filter those cases where *organization_new* is empty. I append those cases and assign the cleaned organization name as the value for *organization_new*

The final cleaning is to unify the use of words:

- remove *univ* and derivatives from the text and re-add it at the end (normalized way) in order to recongnize they are from universities
- normalize the derivatives of *klinikum*
- final adjustments of other variations

```
## Original organization names: 1721

## before extra cleaning:1339 new names

## after extra cleaning: 1284 new names
```

The number of organizations names is reduced in 437

The final step is to recode the original data and save the results.

Final notes

Program choice

I decided to use R as it allows to use powerfull libraries for Text Mining, and hence for the normalization process, and also embed the documentation in the code.