06 Data Transformation

Fabian Blasch

05/31/2022

1 Load Data

```
# source AUX
source("./../Misc/Auxilliary.R")

# packages
get.package(c("bizdays", "lubridate", "tidytext", "stopwords", "stringi",
              "SnowballC", "wordcloud"))

# load data
auctions <- readRDS("./../../Data/Bid Tab RDS/Bid_Tab_Stem.RDS")
```

2 Tabularizing the Data

Firstly, we may construct the data that is easily obtainable via the table and text data.

```
# over years
lapply(auctions, \(y){

  lapply(y, \(a){

    # apply function
    d_transform(a) |> try() # apparently there are three auctions with empty tables
                            # that slipped through
  })

}) -> res
```

```
## Error in data.frame(..., check.names = FALSE) :
##   Argumente implizieren unterschiedliche Anzahl Zeilen: 1, 0
## Error in data.frame(..., check.names = FALSE) :
##   Argumente implizieren unterschiedliche Anzahl Zeilen: 1, 0
## Error in data.frame(..., check.names = FALSE) :
##   Argumente implizieren unterschiedliche Anzahl Zeilen: 1, 0
```

```
# remove auctions that slipped
# over years
lapply(res, \(y){

  sapply(y, \(a){

    # apply function
    class(a) != "try-error" # apparently there are three auctions with empty tables
                            # that slipped through
```

```
  })

}) -> ind_slip

# remove from auctions list
Map(\(au, ind) au[ind], auctions, ind_slip) -> auctions

# remove from table list
Map(\(au, ind) au[ind], res, ind_slip) -> res

# rbind to dataframe
do.call(rbind, lapply(res, \(x) do.call(rbind, x))) -> dat_bids
```

3 Adding the Description via Stemwords

To represent the description, each stem word will be added as a factor.

```
# fetch vector of stemmed words from list
lapply(auctions, \(y){

  # over auctions
  lapply(y, \(a) a[["Stem"]])

}) -> stems

# generate all unique words
do.call(c, lapply(stems, \(x) do.call(c, x))) |> table() |>
  sort(decreasing = TRUE) -> stem_tab

# fill temporary data frame
do.call(rbind, lapply(auctions, \(y){

  do.call(rbind, lapply(y, \(a){

    # match
    match <- (names(stem_tab) %in% a[["Stem"]]) |> matrix(ncol = length(stem_tab))

    # repetitions
    reps <- nrow(a[["Table"]]) - 1

    # new matrix
    match_n <- match[rep(1, times = reps), ]

    # return
    return(match_n)

  })) |> as.data.frame()

}))|> as.data.frame() |> setNames(names(stem_tab)) -> tmp

# dimensions dont add up
lapply(auctions, \(y){
```

```
  do.call(c, lapply(y, \(a){

    # firms
    nrow(a[["Table"]]) - 1


  }))

}) -> ind_firms
```