August 8, 2022

The results below are generated from an R script.

```
#### settings ####
source("./scripts/functions_REN.R")

#### Extraction des relations ####
df_links = data.frame()
df_debiteur_rente = data.frame()
unknow_node <- 0
rentes <- df_main

for(i in 1:nrow(rentes)){
  A <- NULL
  B <- NULL
  if(!is.na(rentes[i,7])){
    text_rente <- rentes[i,7]

    #suppression des expression "ki fu + anthroponyme"
    regex_remove <- "ki fu(rent)? ((femm?e )|((le )?maistre )|((le )?vallés ))?"
    regex <- str_c(regex_remove,regex_anthroponyme_caps)
    text_rente <- str_remove(text_rente,regex)

    #suppression des expression "Si fu + valeur + anthroponyme"
    regex_sifu <- "Si fu.*$"
    text_rente <- str_remove(text_rente,regex)

    #### division en souschaines ####
    regex_sep <- "(S|s|K|k)i s?ien?t"
    if(str_detect(text_rente,regex_sep)){
      loc <- str_locate(text_rente,regex_sep)
      substring1 <- str_sub(text_rente,0,loc[1,1]-1)
      substring2 <-  str_sub(text_rente,loc[1,2]+1,str_length(text_rente))
      A <- ren_extract_caps(substring1, first = FALSE)
      #cas ou aucune EN est detectee dans la premiere substring
      if(length(A)==0){
        A[1] <- str_c("ukn",unknow_node)
        unknow_node <-  unknow_node +1
      }
      df_debiteur_rente <- rbind(df_debiteur_rente,c(A[1],rentes$numRente[i]))

      B <- ren_extract_caps(substring2)
      #### dataframe des aretes ####
      if (length(B)!=0) {
        for(k in 1:length(B)){
          df_links <- rbind(df_links, c(A[1],B[k],rentes$numEscroete[i],
```

```
                                        rentes$numConnetablie[i],
                                        rentes$rdv[i],rentes$numRente[i]))
        }
      }
    }
  }
}
colnames(df_links) <- c("From", "To","NumEscroete",
                        "NumConnetablie", "RdV", "NumRente")

#### Correction des unk_nodes ####
df_uknCorr <- data.frame(
  ukn_node=c("Ukn0","Ukn1","Ukn2","Ukn3","Ukn4","Ukn5","Ukn6",
             "Ukn7","Ukn8","Ukn9","Ukn10","Ukn11","Ukn12","Ukn13","Ukn14"),
  correction=c("LOTIN", "ROBIERT",NA    ,"LES 2 SEREURS DES LICES",
               "LE PRESTRE DES CHARTERIERS"     ,"GODIN"        ,NA     ,NA     ,
               "DES CARTERIERS DES MALADES",
               "HAMIEL" ,"DANIEL"       ,"MARIIEN DE L'EVE","MARIEN",
               "L'OSPITAL DES WES",NA )
)

for (i in 1:nrow(df_uknCorr)) {
  if (!is.na(df_uknCorr$correction[i])) {
    df_links$To[df_links$To == str_c('ukn',i-1)] <- df_uknCorr$correction[i]
  }
}

#### Corrections supplementaires ####

df_add_rel <- data_frame(From= c("ROBIERT","TENEMENT DES MALADES"       ,
                                 "DANIEL"       ,"MARIIEN DE L'EVE",
                                 "MARIIEN DE L'EVE","MARIEN"),
                         To= c("ROBIERT DE FIERIN",     "BAUDE L'ARTISIEN",
                               "JEHAN LE GIERMAIN",     "PIERON DE HASNON",
                               "MARIEN",        "MARGOT DE MAGNI"),
                         NumEscroete = c("I1","I1","II1","II1","II1","II1"),
                         NumConnetablie = c("2°1","2°1","28°","32°","32°","32°"),
                         RdV = c ("A","B","A",NA,NA,NA),
                         NumRente = c("12.6","14.8","139.4","149.2","151.4","152.5")
                         )
names(df_debiteur_rente) <- c("To","NumRente")
df_debiteur_rente <- rbind(df_debiteur_rente,df_add_rel[,c(2,6)])
df_links <- rbind(df_links,df_add_rel)

#### suppression des rentes qui point vers elle mêmes ####
ret <- NULL
for(i in 1:nrow(df_links)){
  if(!is.na(df_links$From[i]) && !is.na(df_links$To[i])){
    if(df_links$From[i] == df_links$To[i]){
      ret <-  c(ret,i)
    }
  }
}
```

```
df_links <- df_links[-ret,]

#### suppression des doublons 'simple' ####
df_links <-  distinct(df_links)

#### suppression des liens reciproques ####
ret <- NULL
for(i in 1:nrow(df_links)){
  df_links[i,] <- c(df_links$To[i],df_links$From[i],
                    df_links$NumEscroete[i],
                    df_links$NumConnetablie[i],
                    df_links$RdV[i],
                    df_links$NumRente[i])
  if(filter(df_links, From == df_links$From[i] & To == df_links$To[i]) %>%
     nrow() >1){
    ret <-  c(ret, i)
  }
}
df_links <- df_links[-ret,]
df_links <- df_links[!is.na(df_links$From),]
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.0.3 (2020-10-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS  12.3.1
##
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] fr_BE.UTF-8/fr_BE.UTF-8/fr_BE.UTF-8/C/fr_BE.UTF-8/fr_BE.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices datasets  utils     methods   base
##
## other attached packages:
##  [1] RColorBrewer_1.1-3 concaveman_1.1.0   ggforce_0.3.3      scales_1.2.0
##  [5] ggrepel_0.9.1      readxl_1.3.1       tidygeocoder_1.0.5 ggraph_2.0.5.9000
##  [9] ggmap_3.0.0        igraph_1.3.0       comparator_0.1.2   forcats_0.5.1
## [13] dplyr_1.0.9        purrr_0.3.4        readr_2.1.2        tidyr_1.2.0
## [17] tibble_3.1.8       ggplot2_3.3.6      tidyverse_1.3.1    stringr_1.4.0.9000
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-7        fs_1.5.2            lubridate_1.8.0     httr_1.4.2
##  [5] tools_4.0.3         backports_1.4.1     utf8_1.2.2          R6_2.5.1
##  [9] DBI_1.1.2           colorspace_2.0-3    withr_2.5.0         sp_1.5-0
## [13] tidyselect_1.1.2    gridExtra_2.3       curl_4.3.2          compiler_4.0.3
## [17] cli_3.3.0           rvest_1.0.2         xml2_1.3.3          proxy_0.4-26
## [21] digest_0.6.29       jpeg_0.1-9          pkgconfig_2.0.3     highr_0.9
## [25] dbplyr_2.1.1        rlang_1.0.4         rstudioapi_0.13     farver_2.1.1
## [29] generics_0.1.3      jsonlite_1.8.0      magrittr_2.0.3      Rcpp_1.0.9
```

```
## [33] munsell_0.5.0       fansi_1.0.3         viridis_0.6.2       lifecycle_1.0.1
## [37] stringi_1.7.6       MASS_7.3-53         plyr_1.8.7          grid_4.0.3
## [41] crayon_1.5.0        lattice_0.20-41     graphlayouts_0.8.0  haven_2.4.3
## [45] hms_1.1.1           knitr_1.37          pillar_1.8.0        rjson_0.2.21
## [49] reprex_2.0.1        glue_1.6.2          evaluate_0.15       renv_0.15.4
## [53] modelr_0.1.8        png_0.1-7           vctrs_0.4.1         tzdb_0.2.0
## [57] tweenr_1.0.2        RgoogleMaps_1.4.5.3 cellranger_1.1.0    gtable_0.3.0
## [61] polyclip_1.10-0     clue_0.3-60         assertthat_0.2.1    xfun_0.30
## [65] broom_0.7.12        tidygraph_1.2.1     viridisLite_0.4.0   tinytex_0.37
## [69] cluster_2.1.0       ellipsis_0.3.2

Sys.time()

## [1] "2022-08-08 07:41:58 CEST"
```