

August 8, 2022

The results below are generated from an R script.

```
##### Import #####
nets <- read.csv("./sources/portes_nets.csv", header = TRUE, sep = ";")
nodes <- read.csv("./sources/portes_nodes.csv", header = TRUE, sep = ";")
corresp_conn <- read_excel("./sources/correspondance_connetable.xlsx")
df_repere <- read_excel("./sources/coord_histo.xlsx")
df_repere$lat <- as.numeric(df_repere$lat)
df_repere$lng <- as.numeric(df_repere$lng)

bbox = c(left = 3.0620, bottom = 50.3625, right = 3.0950, top = 50.3820)
p = get_stamenmap(bbox, zoom=16, source = "stamen", maptype ="watercolor")

## 49 tiles needed, this may take a while (try a smaller zoom).

##### Obtention des coordonnees #####
geo_conn <- corresp_conn %>%
  geocode(correspondance, method = 'osm', lat = lat , long = lng)

## Passing 65 addresses to the Nominatim single address geocoder
## Query completed in: 65.3 seconds

#correction
corr <- c(50.36646162324021, 3.0847360694545984,
         50.36792251302541, 3.080996196322546,
         50.36868284787462, 3.0804070338975666,
         50.372161921584464, 3.081554536863286)

j <- 1
for (i in c("En le rue Pepin",
           "Ou Pont","En le rue Saint Piere",
           "En le rue des Bouloires")) {
  geo_conn$lat[geo_conn$connetable == i] <- corr[j]
  geo_conn$lng[geo_conn$connetable == i] <- corr[j+1]
  j <- j+2
}

##### df_conn_node from df_main #####
df_conn_nodes <- unique(df_main[c(4,3,1)])

row.names(df_conn_nodes) <- NULL
nbRente <- NULL

#colonne nbRente
for (i in df_conn_nodes$numConnetable) {
  count <- df_main$numConnetable[df_main$numConnetable == i] %>% length()
  nbRente <- c(nbRente, count)
```

```

}

df_conn_nodes <- cbind(df_conn_nodes, nbRente)
#correction de l'espace blanc#
for(i in 1:length(df_conn_nodes$connetablie)){
  df_conn_nodes$connetablie[i] <- str_remove(df_conn_nodes$connetablie[i], " ")
}
#jointure
df_conn_nodes <- merge(geo_conn,df_conn_nodes)

##### Graphe des portes #####
names(nets) <- c("porte","porte2")
murs <- inner_join(nets,nodes)

## Joining, by = "porte"

names(murs) <- c("From","porte","From_lat","From_lng")
murs <- inner_join(murs,nodes)

## Joining, by = "porte"

names(murs) <- c("From","porte","From_lat","From_lng","To_lat","To_lng")

portes <- graph_from_data_frame(d=nets,vertices=nodes, directed=FALSE)
V(portes)$lat <- nodes$lat
V(portes)$lng <- nodes$lng
E(portes)$From_lat <- murs$From_lat
E(portes)$From_lng <- murs$From_lng
E(portes)$To_lat <- murs$To_lat
E(portes)$To_lng <- murs$To_lng

subset_nodes <- df_nodes %>%
  select(Anthroponyme,NumConnetablie,NumRente)
names(subset_nodes)[2] <- "numConnetablie"
subset_nodes <- inner_join(subset_nodes,df_conn_nodes)

## Joining, by = "numConnetablie"

for (i in 1:nrow(df_conn_nodes)) {
  df_conn_nodes$numEscroete[i] <- df_conn_nodes$numEscroete[i] %>%
    str_extract("[IV]+")
  df_conn_nodes$numConnetablie[i] <- df_conn_nodes$numConnetablie[i] %>%
    str_remove("1$")
}
noms_escroetes <- data.frame(
  numEscroete = c("I","II","III","IV","V","VI","VII"),
  escroete =c("Markiet","Canteleu","Més","Wés","Nuevile",
             "Deuwioel","Escroete VII"))
df_conn_nodes <- merge(df_conn_nodes,noms_escroetes)
df_conn_nodes <- na.omit(df_conn_nodes)

##### map plotting connetablie simple #####
ggmap(p, base_layer = ggraph(portes)) +
  geom_point(aes(lng,lat), color = 'red', shape = 18, size= 5)+ #portes
  geom_edge_link(aes(x = From_lng, y = From_lat, xend = To_lng, yend = To_lat ) ,
                 color = 'red', width = 1) + #murs
  geom_node_point(data=df_conn_nodes, aes(lng, lat, fill = numConnetablie),

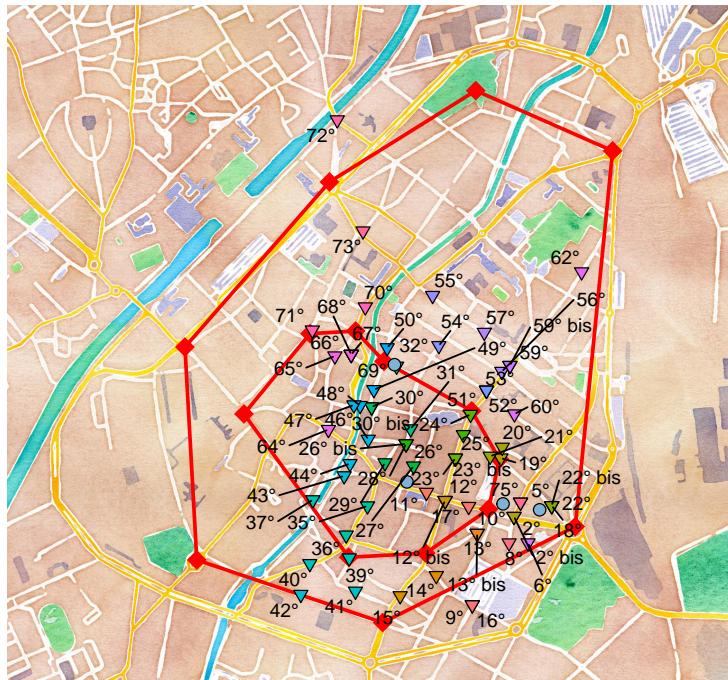
```

```

        colour= 'black', shape = 25, size = 2.5) + #connetablies
geom_node_point(data = df_repere,aes(lng,lat),
                fill = '#7FB3D5', shape = 21, size= 3, color='black')+ 
geom_text_repel(data=df_conn_nodes,
                 aes(x=lng, y=lat, label = numConnetablie)) +
theme(legend.position = "none")

## Using 'stress' as default layout
## Using 'stress' as default layout
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé

```



```

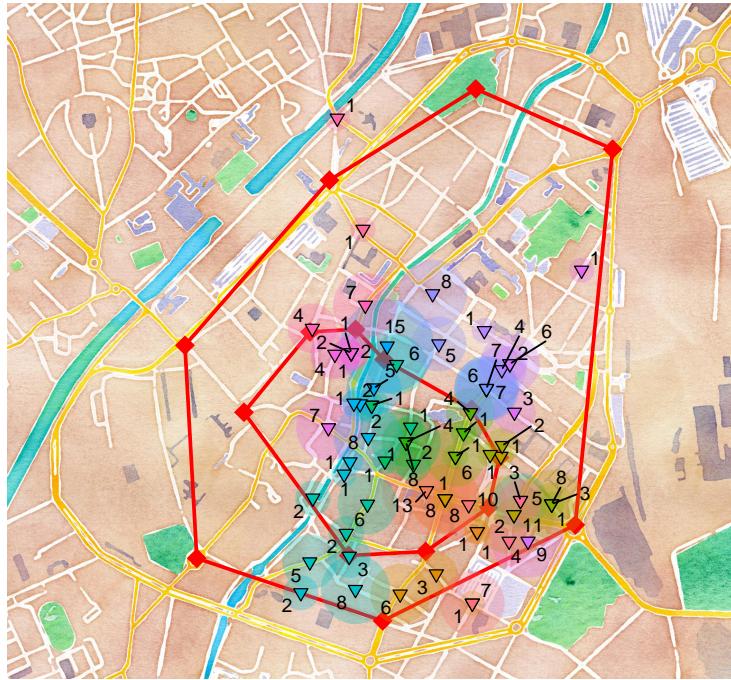
##### map nb rente #####
ggmap(p, base_layer = ggraph(portes)) +
  geom_point(aes(lng,lat), color = 'red', shape = 18, size= 5)+ 
  geom_edge_link(aes(x = From_lng, y = From_lat, xend = To_lng, yend = To_lat ),
                 color = 'red', width = 1) +
  geom_node_point(data=df_conn_nodes,
                  aes(lng, lat, colour = numConnetablie, size = nbRente ),
                  shape = 16,alpha = 0.3) +
  geom_node_point(data=df_conn_nodes,
                  aes(lng, lat, fill = numConnetablie),
                  colour= 'black', shape = 25, size = 2.5) +
  scale_size_area(max_size = 25) +
  theme(legend.position = "none") +
  geom_text_repel(data=df_conn_nodes,
                 aes(x=lng, y=lat, label = nbRente))

```

```

## Using 'stress' as default layout
## Using 'stress' as default layout
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in min(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé

```

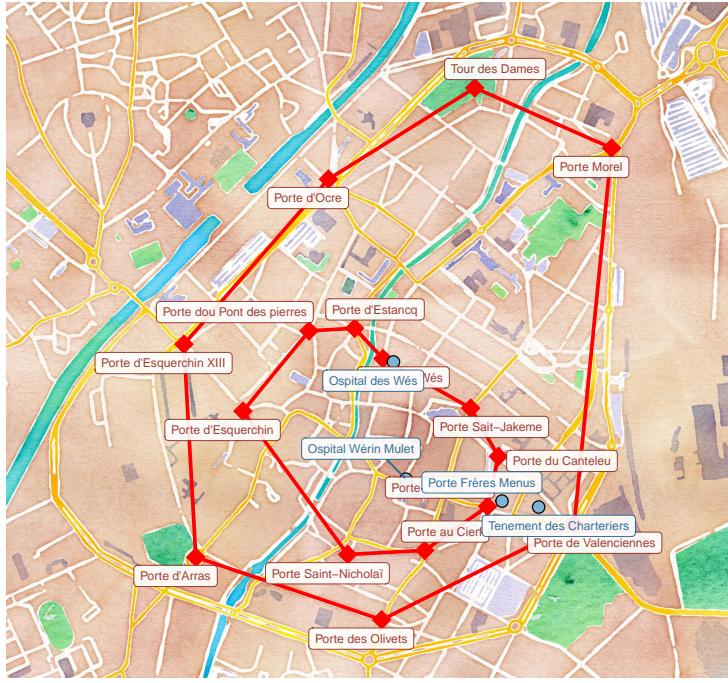


```

##### map plotting Name #####
ggmap(p, base_layer = ggraph(portes)) +
  geom_point(aes(lng,lat), color = 'red', shape = 18, size= 5)+ 
  geom_edge_link(aes(x = From_lng, y = From_lat, xend = To_lng, yend = To_lat ),
                 color = 'red', width = 1) + 
  geom_node_point(data = df_repere,aes(lng,lat),
                  fill = '#7FB3D5', shape = 21, size= 3, color='black')+ 
  geom_label_repel(aes(x=lng, y=lat, label = name),
                   color='#922B21', size = 2.5, alpha = 0.9) + 
  geom_label_repel(data =df_repere,aes(x=lng, y=lat, label = nom),
                   color='#1F618D', size = 2.5, alpha = 0.9) + 
  theme(legend.position = "none")

## Using 'stress' as default layout
## Using 'stress' as default layout
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in min(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé

```

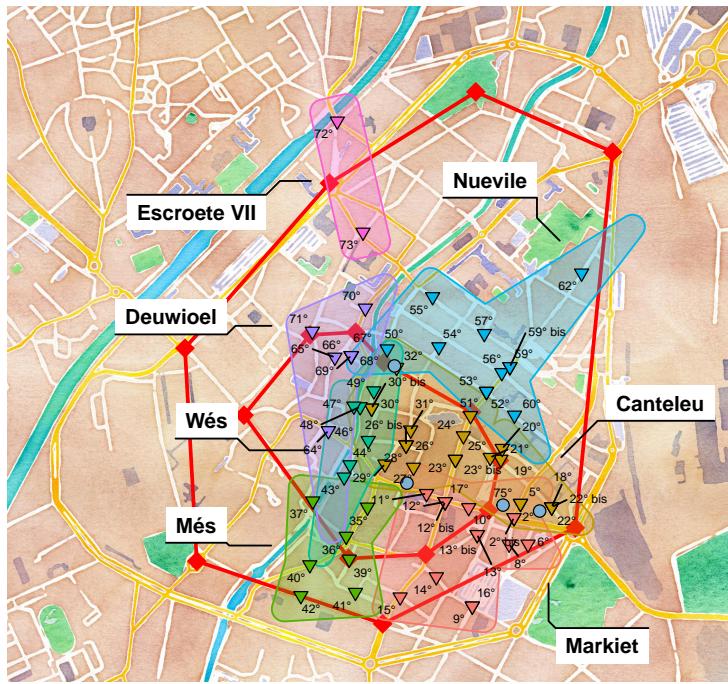


```

##### map plotting Escroete #####
ggmap(p, base_layer = ggraph(portes)) +
  geom_point(aes(lng,lat), color = 'red', shape = 18, size= 5)+ 
  geom_edge_link(aes(x = From_lng, y = From_lat, xend = To_lng, yend = To_lat ),
                 color = 'red', width = 1) + 
  geom_mark_hull(data = df_conn_nodes,
                  aes(x = lng, y = lat,
                      fill=numEscroete, color=numEscroete,
                      label= escroete),label.width = NULL)+ 
  geom_node_point(data=df_conn_nodes, aes(lng, lat, fill = numEscroete),
                  colour= 'black', shape = 25, size = 2.5) + 
  geom_node_point(data = df_repere,aes(lng,lat),
                  fill = '#7FB3D5', shape = 21, size= 3, color='black') + 
  geom_text_repel(data=df_conn_nodes,
                  aes(x=lng, y=lat, label = numConnetable),
                  size = 2.5) + 
  theme(legend.position = "none")

## Using 'stress' as default layout
## Using 'stress' as default layout
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in min(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé

```

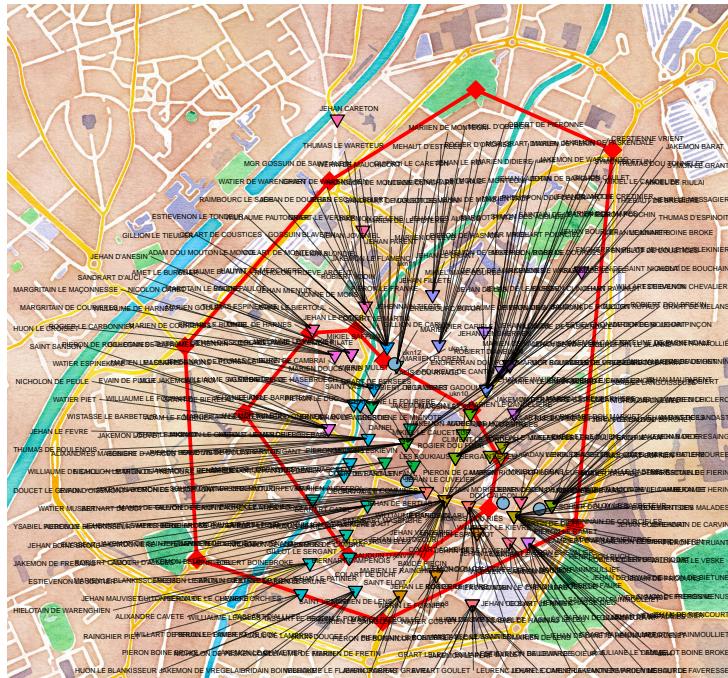


```

##### map ploting anthroponyme #####
ggmap(p, base_layer = ggraph(portes)) +
  geom_point(aes(lng,lat), color = 'red', shape = 18, size= 5)+ 
  geom_edge_link(aes(x = From_lng, y = From_lat, xend = To_lng, yend = To_lat ),
                 color = 'red', width = 1) + 
  geom_node_point(data=df_conn_nodes,
                  aes(lng, lat, fill = numConnetable),
                  colour= 'black', shape = 25, size = 2.5) + 
  geom_node_point(data = df_repere,aes(lng,lat),
                  fill = '#7FB3D5', shape = 21, size= 3, color='black')+ 
  geom_text_repel(data=subset_nodes, aes(x=lng, y=lat, label = Anthroponyme),
                  size = 1.5, segment.size = 0.1,segment.alpha = 0.7) + 
  theme(legend.position = "none")

## Using 'stress' as default layout
## Using 'stress' as default layout
## Warning: Removed 58 rows containing missing values (geom_text_repel).
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé

```



```

##### map plotting anthroponyme focus #####
`%!in%` <- Negate(`%in%`)
personne_of_interest <- "(BOINE ?BROKE) | (DE FRANCHE)"
personne_of_interest <- subset_nodes$Anthroponyme[str_detect(subset_nodes$Anthroponyme,
                                                               personne_of_interest) ]

ggmap(p, base_layer = ggraph(portes)) +
  geom_point(aes(lng,lat), color = 'red', shape = 18, size= 5, alpha = 0.5) +
  geom_edge_link(aes(x = From_lng, y = From_lat, xend = To_lng, yend = To_lat ),
                 color = 'red', width = 1, alpha = 0.5) +
  geom_node_point(data=df_conn_nodes, aes(lng, lat, fill = numConnetable),
                  colour= 'black', shape = 25, size = 2.5, alpha = 0.5) +
  geom_node_point(data = df_reperes,aes(lng,lat),
                  fill = '#7FB3D5', shape = 21, size= 3,
                  color='black', alpha = 0.5) +
  geom_text_repel(data=subset_nodes %>%
                      filter(Anthroponyme %!in% personne_of_interest),
                  aes(x=lng, y=lat, label = Anthroponyme),
                  size = 1.5, alpha =0.3,
                  segment.size = 0.1,segment.alpha =  0.3) +
  geom_text_repel(data=subset_nodes %>%
                      filter(Anthroponyme %in% personne_of_interest),
                  aes(x=lng, y=lat, label = Anthroponyme),
                  size = 3, nudge_x = -0.004,nudge_y = 0.001,
                  segment.size = 0.3,segment.alpha =  0.8, color = '#154360') +
  theme(legend.position = "none")

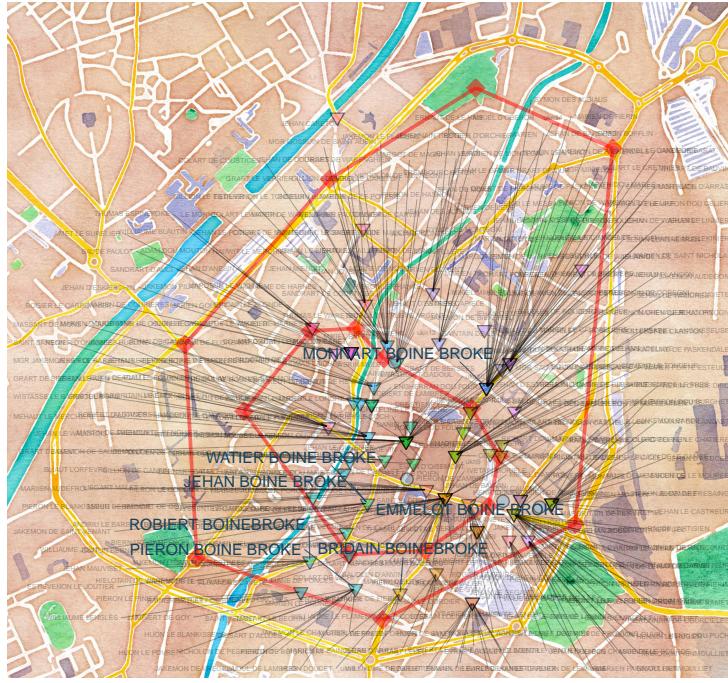
## Using 'stress' as default layout
## Using 'stress' as default layout
## Warning: Removed 58 rows containing missing values (geom_text_repel).
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé

```

```

## Warning in max(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé

```

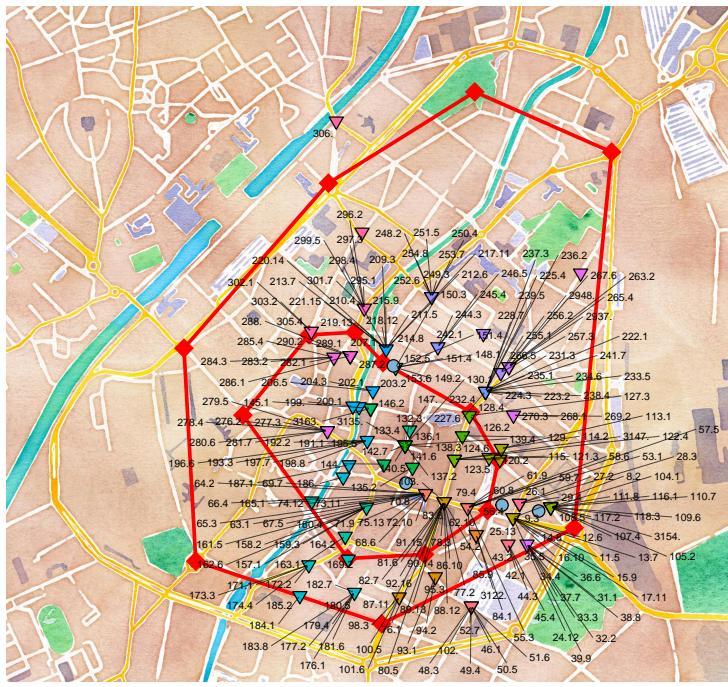


```

##### map plotting rente #####
ggmap(p, base_layer = ggraph(portes)) +
  geom_point(aes(lng,lat), color = 'red', shape = 18, size= 5) +
  geom_edge_link(aes(x = From_lng, y = From_lat, xend = To_lng, yend = To_lat ),
                 color = 'red', width = 1) +
  geom_node_point(data=df_conn_nodes, aes(lng, lat, fill = numConnetable),
                  colour= 'black', shape = 25, size = 2.5) +
  geom_node_point(data = df_repere,aes(lng,lat), fill = '#7FB3D5',
                  shape = 21, size= 3, color='black')+
  geom_text_repel(data=subset_nodes %>% filter(!is.na(NumRente)),
                  aes(x=lng, y=lat, label = NumRente),
                  size = 2, segment.size = 0.1,segment.alpha = 0.7) +
  theme(legend.position = "none")

## Using 'stress' as default layout
## Using 'stress' as default layout
## Warning: Removed 25 rows containing missing values (geom_text_repel).
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé
## Warning in min(x): aucun argument trouvé pour min ; Inf est renvoyé
## Warning in max(x): aucun argument pour max ; -Inf est renvoyé

```



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       labeling_0.4.2
## [21] proxy_0.4-26     digest_0.6.29     jpeg_0.1-9       pkgconfig_2.0.3
```

```
## [25] highr_0.9           dbplyr_2.1.1        rlang_1.0.4         rstudioapi_0.13
## [29] farver_2.1.1         generics_0.1.3      jsonlite_1.8.0     magrittr_2.0.3
## [33] Rcpp_1.0.9           munsell_0.5.0       fansi_1.0.3         viridis_0.6.2
## [37] lifecycle_1.0.1      stringi_1.7.6       MASS_7.3-53        plyr_1.8.7
## [41] grid_4.0.3           crayon_1.5.0        lattice_0.20-41    graphlayouts_0.8.0
## [45] haven_2.4.3          hms_1.1.1           knitr_1.37        pillar_1.8.0
## [49] rjson_0.2.21         reprex_2.0.1        glue_1.6.2         evaluate_0.15
## [53] V8_4.2.0             renv_0.15.4         modelr_0.1.8       png_0.1-7
## [57] vctrs_0.4.1          tzdb_0.2.0          tweenr_1.0.2       RgoogleMaps_1.4.5.3
## [61] cellranger_1.1.0     gtable_0.3.0        polyclip_1.10-0    clue_0.3-60
## [65] assertthat_0.2.1     xfun_0.30           broom_0.7.12       tidygraph_1.2.1
## [69] viridisLite_0.4.0    tinytex_0.37        cluster_2.1.0      ellipsis_0.3.2

Sys.time()

## [1] "2022-08-08 07:47:42 CEST"
```