Spis treści

[# Starter 2](#_Toc166492639)

[# STEP 0 - reading dta data 2](#_Toc166492640)

[# STEP 1 – generating unique address for each line 2](#_Toc166492641)

[# STEP 2 – dividing the dataset into municipalities 3](#_Toc166492642)

[# STEP 3 - matching for Quebec - 23027 3](#_Toc166492643)

[# STEP 4 - making clusters by address (street & number) 3](#_Toc166492644)

[# STEP 5 – matching of clusters by distance, same side of street and dates of in/out 5](#_Toc166492645)

[# Step 6 – matching of non-address with clusters 6](#_Toc166492646)

[# STEP 7 – tracking for proper radius of knn1 of non-linked rows 7](#_Toc166492647)

[# STEP 8 – clustering of no-address rows (if they are in close neighbourhood to others) 7](#_Toc166492648)

[# STEP 9 – summaries 8](#_Toc166492649)

# # Starter

# my laptop

setwd("C:/Users/Admin/Desktop/mydata")

# # STEP 0 - reading dta data

library(tidyverse)

library(haven)

library(sp)

library(dbscan)

data<-read\_dta(file="MapingChanges.dta") # class tibble

data<-as.data.frame(data)

ID<-1:dim(data)[1] # adding unique ID

data<-cbind(ID, data)

summary(data)

# # STEP 1 – generating unique address for each line

# we check the variance of noinf – for var=0 lines we build address

vars<-apply(data[,18:33], 1, sd, na.rm=TRUE) # noinf

means<-apply(data[,18:33], 1, mean, na.rm=TRUE) # noinf

data$NOINF<-means

data$NOINF[vars>0]<-NA

b<-which(is.nan(data$NOINF)==TRUE)

data$NOINF[b]<-NA

# we check noinf and rue to find the rows with same values

# operation to generate single column with the proper address

sub1<-data[data$Out==2020 & is.na(data$Out)==FALSE,]

sub2<-data[data$Out==2019 & is.na(data$Out)==FALSE,]

sub3<-data[data$Out==2018 & is.na(data$Out)==FALSE,]

sub4<-data[data$Out==2017 & is.na(data$Out)==FALSE,]

sub5<-data[data$Out==2016 & is.na(data$Out)==FALSE,]

sub6<-data[data$Out==2015 & is.na(data$Out)==FALSE,]

sub7<-data[data$Out==2014 & is.na(data$Out)==FALSE,]

sub8<-data[data$In==2021 & is.na(data$In)==FALSE,]

sub9<-data[data$In==2020 & is.na(data$In)==FALSE,]

sub10<-data[data$In==2019 & is.na(data$In)==FALSE,]

sub11<-data[data$In==2018 & is.na(data$In)==FALSE,]

sub12<-data[data$In==2017 & is.na(data$In)==FALSE,]

sub13<-data[data$In==2016 & is.na(data$In)==FALSE,]

sub14<-data[data$In==2015 & is.na(data$In)==FALSE,]

#sub1$address<-paste(sub1$rue2020, sub1$noinf2020, sep="\_")

#sub2$address<-paste(sub2$rue2019, sub2$noinf2019, sep="\_")

#sub3$address<-paste(sub3$rue2018, sub3$noinf2018, sep="\_")

#sub4$address<-paste(sub4$rue2017, sub4$noinf2017, sep="\_")

#sub5$address<-paste(sub5$rue2016, sub5$noinf2016, sep="\_")

#sub6$address<-paste(sub6$rue2015, sub6$noinf2015, sep="\_")

#sub7$address<-paste(sub7$rue2014, sub7$noinf2014, sep="\_")

#sub8$address<-paste(sub8$rue2021, sub8$noinf2021, sep="\_")

#sub9$address<-paste(sub9$rue2020, sub9$noinf2020, sep="\_")

#sub10$address<-paste(sub10$rue2019, sub10$noinf2019, sep="\_")

#sub11$address<-paste(sub11$rue2018, sub11$noinf2018, sep="\_")

#sub12$address<-paste(sub12$rue2017, sub12$noinf2017, sep="\_")

#sub13$address<-paste(sub13$rue2016, sub13$noinf2016, sep="\_")

#sub14$address<-paste(sub14$rue2015, sub14$noinf2015, sep="\_")

sub1$address<-paste(sub1$rue2020, sub1$NOINF, sep="\_")

sub2$address<-paste(sub2$rue2019, sub2$NOINF, sep="\_")

sub3$address<-paste(sub3$rue2018, sub3$NOINF, sep="\_")

sub4$address<-paste(sub4$rue2017, sub4$NOINF, sep="\_")

sub5$address<-paste(sub5$rue2016, sub5$NOINF, sep="\_")

sub6$address<-paste(sub6$rue2015, sub6$NOINF, sep="\_")

sub7$address<-paste(sub7$rue2014, sub7$NOINF, sep="\_")

sub8$address<-paste(sub8$rue2021, sub8$NOINF, sep="\_")

sub9$address<-paste(sub9$rue2020, sub9$NOINF, sep="\_")

sub10$address<-paste(sub10$rue2019, sub10$NOINF, sep="\_")

sub11$address<-paste(sub11$rue2018, sub11$NOINF, sep="\_")

sub12$address<-paste(sub12$rue2017, sub12$NOINF, sep="\_")

sub13$address<-paste(sub13$rue2016, sub13$NOINF, sep="\_")

sub14$address<-paste(sub14$rue2015, sub14$NOINF, sep="\_")

**# creating single dataset – for all years**

data1<-rbind(sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8, sub9, sub10, sub11, sub12, sub13, sub14)

library(stringr)

a<-str\_detect(data1$address, "NA")

a1<-which(a==TRUE)

data1$address[a1]<-NA

# 🡪 here we have the correct full last/first address

**# object data1 is the one to work with**

## # STEP 2 – dividing the dataset into municipalities

**#########################################################**

# codes we keep

#code\_mun=="23027" | /\*Quebec\*/

#code\_mun=="23057" | /\*Ancienne-Lorette\*/

#code\_mun=="23072" | /\*St-Augustin-de-Desmaures\*/

#code\_mun=="25213" | /\*Levis\*/

**#########################################################**

## # STEP 3 - matching for Quebec - 23027

data2<-data1[data1$code\_mun=="25213",] # 10300 obs

data2$even\_num<-data2$NOINF %% 2 == 0 # Create even/odd logical for NOINF

## # STEP 4 - making clusters by address (street & number)

# vector of all unique cleaned addresses

uni<-unique(data2$address)

k<-which(is.na(uni)) # eliminating NA address

uni<-uni[-k]

nn<-length(uni)

data2$clust<-0 # new variable in basic dataset

b.collected<-matrix(0, nrow=nn, ncol=20) # object - IDs of clusters

# loop by unique addresses

for(i in 1:nn){ # for all correct addresses

vec<-uni[i] # analysed address

b0<-data2$address %in% vec # searched in all years

b1<-which(b0==TRUE)

name<-paste0("clustA\_",i) # cluster name

**# saving the information**

ile<-length(b1) # how many IDs

ile3<-ile+3

b.collected[i,1]<-vec # saving the address

b.collected[i,2]<-name # saving the name of cluster

b.collected[i,3]<-ile # number of rows in clusters

b.collected[i,4:ile3]<-b1 # saving IDs with address

data2$clust[b1]<-name} # cluster name added to full dataset

# 17 columns are the places to list rows ID are in given cluster

colnames(b.collected)<-c("adddress", "cluster", "nb\_rows", "row1", "row2", "row3", "row4", "row5", "row6", "row7", "row8", "row9", "row10", "row11", "row12", "row13", "row14", "row15", "row16", "row17")

table(b.collected[,3])

# 1 10 2 3 4 8

#3407 1 1346 28 4 1

# 🡪 this gives us following info:

# which rows are linked by the same address with another rows

# which address has how many rows

# which rows were never linked or even considered (no full address or no link with other rows)

**head(b.collected)**

adddress cluster nb\_rows row1 row2 row3 row4 row5 row6

[1,] "Marches-Naturelles\_26" "clustA\_1" "2" "1" "3973" "0" "0" "0" "0"

[2,] "Marches-Naturelles\_20" "clustA\_2" "2" "2" "3974" "0" "0" "0" "0"

[3,] "Marches-Naturelles\_18" "clustA\_3" "2" "3" "3975" "0" "0" "0" "0"

[4,] "Magella-Laforest\_25" "clustA\_4" "1" "6" "0" "0" "0" "0" "0"

[5,] "Saint-Raphael\_74" "clustA\_5" "2" "7" "3994" "0" "0" "0" "0"

[6,] "Saint-Raphael\_80" "clustA\_6" "2" "8" "3995" "0" "0" "0" "0"

row7 row8 row9 row10 row11 row12 row13 row14 row15 row16 row17

[1,] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"

[2,] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"

[3,] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"

[4,] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"

[5,] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"

[6,] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"

## # STEP 5 – matching of clusters by distance, same side of street and dates of in/out

# from clustered data we check if already clustered by address

# for clustered data we choose neighbours up to 100 m & on right side of street

# and if dates of IN differ by 1 year from dates OUT

# if yes, we make new cluster in cluster 2 column

N<-dim(data2)[1]

data2$done<-0

data2$clust2<-0

for(i in 1:N){

clust<-data2$clust[i] # cluster ID of analysed case

even\_odd<-data2$even\_num[i]

if (clust==0) {next}

if (data2$done[i]=="done") {next}

# neighbours in radius of 100m from address

data2$a<-spDistsN1(as.matrix(data2[,5:6]), as.matrix(data2[i,5:6]), longlat=FALSE) # dist from cluster to all other obs (clust and non-clust)

b1<-data2[data2$a<100, ] # neighbours in 100 m

b2<-b1[b1$even\_num==even\_odd,] # the same side of the street

b3<-b2[is.na(b2$ID)==FALSE,] # non-empty rows

b3$In[is.na(b3$In)==TRUE & is.nan(mean(b3$In, na.rm=TRUE))==TRUE]<-10000

b3$Out[is.na(b3$Out)==TRUE & is.nan(mean(b3$Out, na.rm=TRUE))==TRUE]<-10000

c1<-mean(b3$In, na.rm=TRUE)==mean(b3$Out, na.rm=TRUE)+1 # if consequtive years

c2<-length(unique(b3$In)) # we want just two values: NA and one year

c3<-length(unique(b3$Out)) # we want just two values: NA and one year

if (c1==TRUE & c2==2 & c3==2) {b3<-b3}

else {b3<-b3[b3$clust2!=0,]}

DONE<-data2$ID %in% b3$ID

data2$done[DONE==TRUE]<-"done"

data2$clust2[DONE==TRUE]<-paste0("clustB\_",i)

}

## # Step 6 – matching of non-address with clusters

# the same as in STEP 5 but for non-clustered data

# in single run of the loop: we take single cluster and calculate distance

# to all un-clustered points (<50m)

# then we check dates of in/out

# merging cluster labels from 1st and 2nd clustering

data2$clust2.1<-0

data2$clust2.1<-ifelse(data2$clust2!=0, data2$clust2, data2$clust)

data2$clust3<-0

data2$done2<-0

# checking how many clusters we have now

clu<-unique(data2$clust2.1[data2$clust2.1!=0]) # no cluster=0

N<-length(clu)

# loop for each cluster

for(i in 1:N){

clu.now<-clu[i]

b1<-data2[data2$clust2.1==clu.now, ] # cluster subset

crds<-apply(b1[,5:6], 2, mean) # average crds in cluster

# adding potential non-address observations which could be clustered

data2$a<-spDistsN1(as.matrix(data2[,5:6]), as.matrix(crds), longlat=FALSE)

b1<-data2[(data2$clust2.1==clu.now | data2$clust2.1==0) & data2$a**<50**, ]

b1 # can be empty

if (length(unique(b1$clust2.1))==1) {next} # we take groups with new row only

if (dim(b1)[1]==0) {next} # for empty dataset go next

# we check In and Out dates

b1$In[is.na(b1$In)==TRUE & is.nan(mean(b1$In, na.rm=TRUE))==TRUE]<-10000

b1$Out[is.na(b1$Out)==TRUE & is.nan(mean(b1$Out, na.rm=TRUE))==TRUE]<-10000

c1<-mean(b1$In, na.rm=TRUE)==mean(b1$Out, na.rm=TRUE)+1

if (c1==TRUE) {b1<-b1}

else {b1<-b1[b1$clust3!=0,]}

DONE<-data2$ID %in% b1$ID

data2$done2[DONE==TRUE]<-"done"

data2$clust3[DONE==TRUE]<-paste0("clustC\_",i)

}

data2$CLUST<-0

data2$CLUST<-ifelse(data2$clust3!=0, data2$clust3, data2$clust2.1)

## # STEP 7 – tracking for proper radius of knn1 of non-linked rows

# parcel sizes differ – smaller in the center and bigger outside

# we check the average distance to knn1 in 1km surrounding of buildings

# the average distance will be taken as radius for searching clusters

**# distance to knn1 for each observation**

a<-kNN(data2[,5:6], k=1) # knn1 for each point

a2<-a$dist

colnames(a2)<-"dist\_knn1"

data2$dist\_knn1<-a2 # distance to the knn1 neighbour

data2$av.dist.knn1<-0

a3<-frNN(data2[,5:6], eps=1000) # neighbours within a radius of 1 km

# average distance to knn1 for all points in radius of 1km

for(i in 1:dim(data2)[1]){

oo<-data2[unlist(a3$id[i]),]

data2$av.dist.knn1[i]<-mean(oo$dist\_knn1)}

#data2$status<-ifelse(data2$dist\_knn1>data2$av.dist.knn1, "alone", "close neighbourhood")

data2$status2<-"close neighbourhood"

data2$status2[data2$dist\_knn1>data2$av.dist.knn1]<-"alone"

## # STEP 8 – clustering of no-address rows (if they are in close neighbourhood to others)

N<-length(data2$CLUST[data2$CLUST==0]) # for each non-clustered row

vec<-which(data2$CLUST==0)

data2$clust4<-0

# adding potential non-address observations which could be clustered

for(i in 1:N){

crds<-data2[vec[i],5:6]

data2$a<-spDistsN1(as.matrix(data2[,5:6]), as.matrix(crds), longlat=FALSE)

b1<-data2[data2$CLUST==0 & data2$a<=data2$av.dist.knn1,]

# we check In and Out dates

b1$In[is.na(b1$In)==TRUE & is.nan(mean(b1$In, na.rm=TRUE))==TRUE]<-10000

b1$Out[is.na(b1$Out)==TRUE & is.nan(mean(b1$Out, na.rm=TRUE))==TRUE]<-10000

c1<-mean(b1$In, na.rm=TRUE)==mean(b1$Out, na.rm=TRUE)+1

if (c1==TRUE) {b1<-b1}

else {b1<-b1[b1$CLUST!=0,]}

DONE<-data2$ID %in% b1$ID

data2$done2[DONE==TRUE]<-"done"

data2$clust4[DONE==TRUE]<-paste0("clustD\_",i)

}

data2$CLUST2<-0

data2$CLUST2<-ifelse(data2$CLUST!=0, data2$CLUST, data2$clust4)

## # STEP 9 – summaries

##############

# saving the file with clusterings

write.table(data2, file="dataALL2.txt")

# summaries – size of clusters 🡪 too big to display

table(data2$CLUST2)

FINAL<-data.frame(clust=unique(data2$CLUST2), count\_Out=0, count\_In=0, date\_Out=0, date\_In=0)

for(i in 1:dim(FINAL)[1]){

sub<-data2[data2$CLUST2==FINAL$clust[i],]

FINAL$count\_In[i]<-length(sub$In[is.na(sub$In)==FALSE])

FINAL$count\_Out[i]<-length(sub$Out[is.na(sub$Out)==FALSE])

FINAL$date\_In[i]<-mean(sub$In[is.na(sub$In)==FALSE])

FINAL$date\_Out[i]<-mean(sub$Out[is.na(sub$Out)==FALSE])

}

FINAL$status<-ifelse(FINAL$count\_Out==1 & FINAL$count\_In==1,"one-to-one", ifelse(FINAL$count\_Out==1 & FINAL$count\_In>1, "one-to-many", ifelse(FINAL$count\_Out==0 | FINAL$count\_In==0, "non-clustered", ifelse(FINAL$count\_Out>1 & FINAL$count\_In==1, "many-to-one", ifelse(FINAL$count\_Out>1 & FINAL$count\_In>1, "many-to-many", "other")))))

head(FINAL)

# clust count\_Out count\_In date\_Out date\_In status

#1 clustB\_1 3 3 2020 2021 many-to-many

#2 clustD\_1262 1 1 2020 2021 one-to-one

#3 clustC\_1702 1 1 2020 2021 one-to-one

#4 clustB\_6 1 1 2020 2021 one-to-one

#5 clustA\_5 1 1 2020 2021 one-to-one

#6 clustA\_6 1 1 2020 2021 one-to-one

table(FINAL$status)

#many-to-many many-to-one not-cluster one-to-many one-to-one

# 251 156 2573 295 1527

N<-length(data2$CLUST2[data2$CLUST2==0]) # nun-clustered obs

N

# 1822

table(data2$status2)

# alone close neighbourhood

# 2684 7615

# saving the file with clusterings

write.table(FINAL, file="FINAL\_summary.txt")