Amenity Weihts

08/06/2021

0) Useful Libraries

```
# wrangling/convenience
library(tidyverse)
library(stringr)
library(imputeTS)
library(corrplot)
library(tidyr)
library(hablar)
options(qwraps2_markup='markdown')

# For pretty knitting
library(lemon)
knit_print.data.frame <- lemon_print
knit_print.tbl <- lemon_print
knit_print.summary <- lemon_print
```

1) Data Wrangling

```
### source file

source('weights_function.R')

library_obj<-read.csv("../../data/raw/public_library_data.csv",fileEncoding="latin1")
review_poi<-read_csv("../../data/clean/google_reviews_poi_with_hours.csv")
van_poi<-read_csv("../../data/clean/vancouver_facilities_2.csv")
### import travel time matrix
ttm <- read_csv('../../data/clean/ttm.csv')
```

Import dataset

```
#Merge review dataset with vancouver point of interest
left_join(review_poi,van_poi,by=c("poi_name"="name"))%>%distinct()->merged_data
```

Merge data review_poi and van_poi Convert the data to numeric

```
merged_data%>%convert(num(Rating, Total_Review,open_days,Total_hours))->merged_data
```

Number of amenity in each type of arts facility

```
merged_data%>%group_by(type)%>%count()
```

```
## # A tibble: 9 x 2
## # Groups:   type [9]
##   type                                  n
```

```
##   <chr>                                <int>
## 1 art or cultural centre                   5
## 2 artist                                  48
## 3 festival site                            2
## 4 gallery                                 99
## 5 heritage or historic site               28
## 6 library or archives                     86
## 7 miscellaneous                            6
## 8 museum                                  92
## 9 theatre/performance and concert hall    75
```

Our primary interest would be gallery(n=99),library(n=86),museum(n=92) and theatre(n=75)

Wrangling objective library weight data

data cleaning

```
van_list = c('Richmond','Vancouver','Burnaby','Township of Langley','North Vancouver','New Westminster'
            'Coquitlam','Delta','Fort Langley','White Rock','Pitt Meadows','Port Coquitlam', 'West Vanc

# select those only in big vancouver area
library_obj%>%filter(CITY%in%van_list)->library_obj

# select column and rename them
colnames(library_obj)
```

```
##  [1] "LIBRARY_SYSTEM"         "LOCATION"               "BRANCH_UNIQUE_ID"
##  [4] "SCHOOL_DISTRICT_SERVED" "PHONE"                  "PHYSICAL_ADDRESS"
##  [7] "CITY"                   "PROVINCE"               "POSTAL_CODE"
## [10] "LATITUDE"               "LONGITUDE"              "MTLS_OUTLET"
## [13] "MTLS_CIRC_B"            "CIRC_CHILD_MTLS_B"      "REF_TRANS_B"
## [16] "VISITS_B"               "AD_INLIB_PGMS_B."       "AD_OUT_PGMS_B."
## [19] "ADULT_ATTEND_B."        "CH_INLIB_PGMS_B."       "CH_OUT_PGMS_B."
## [22] "CHILD_ATTEND_B."        "YA_INLIB_PGMS_B."       "YA_OUT_PGMS_B."
## [25] "YA_ATTEND_B."           "ESL_INLIB_PGMS_B."      "ESL_OUT_PGMS_B."
## [28] "ESL_ATTEND_B."          "LIBRARIAN_HRS_B."       "LIB_TECH_HRS_B."
## [31] "COMM_LIB_HRS_B."        "OTH_HRS_B."             "branch_copiers."
## [34] "LEED_CERT_B."           "SHRD_FAC."              "FLOORSPACE."
## [37] "HRS_OPEN."              "DAYS_OPEN."
```

```
library_obj%>%select('BRANCH_UNIQUE_ID', 'POSTAL_CODE', 'LATITUDE', 'LONGITUDE', 'MTLS_OUTLET', 'VISITS_

colnames(lib_df)<-c('Branch_Id', 'Postal_Code', 'lat', 'lon', 'Total_Volumes', 'Annual_Visitors', 'Total

# check nas
colMeans(is.na(lib_df))
```

```
##       Branch_Id     Postal_Code             lat             lon   Total_Volumes
##               0               0               0               0               0
## Annual_Visitors     Total_Space    Hrs_per_Year   Days_per_Year
##               0               0               0               0
```

```
#### select relevent variable
merged_data%>%select(poi_name,Name,type,open_days,Total_hours,Rating,Total_Review,lat,lon)->df
colnames(df) <- c("POI Name", "Google Place Name","Type","Open Days","Open Hours","Rating","Total Revie
df%>%filter(Type=="library or archives")->df_lib
```

```
# left join

#round to 2 decimal

df_lib%>% convert(num(lat,lon))->df_lib
```

```
## Warning in as_reliable_num(.): NAs introduced by coercion

## Warning in as_reliable_num(.): NAs introduced by coercion
```

```
df_lib$lat<-round(df_lib$lat,2)
df_lib$lon<-round(df_lib$lon,2)
lib_df$lat<-round(lib_df$lat,2)
lib_df$lon<-round(lib_df$lon,2)
df_lib<-na_mean(df_lib)
```

```
## Warning: imputeTS: No imputation performed for column 2 because of this Error in na_mean(data[, i], o
```

```
merged_lib<-left_join(df_lib,lib_df,by = c("lat", "lon"))

colnames(merged_lib)
```

```
##  [1] "POI Name"          "Google Place Name" "Type"
##  [4] "Open Days"         "Open Hours"        "Rating"
##  [7] "Total Review"      "lat"               "lon"
## [10] "Branch_Id"         "Postal_Code"       "Total_Volumes"
## [13] "Annual_Visitors"   "Total_Space"       "Hrs_per_Year"
## [16] "Days_per_Year"
```

```
merged_lib%>%select("POI Name","Open Days","Open Hours", "Rating","Total Review","Total_Volumes","Annual
merged_lib%>% convert(num(Total_Volumes,Annual_Visitors,Total_Space))->merged_lib
```

```
## Warning in as_reliable_num(.): NAs introduced by coercion

## Warning in as_reliable_num(.): NAs introduced by coercion

## Warning in as_reliable_num(.): NAs introduced by coercion
```

2) Weights computing

Compute the weights separately for library amenity Since only library amenity has extra features from outside of google place data for example, "Total_Volumes","Annual_Visitors","Total_Space" thus I decide to compute it separately

```
# exclude the Name variable
n_features<-merged_lib%>%rowwise()%>%apply(1,function(x) sum(is.na(x)==F))

#normaliz with library
normalize <- function(x) {
return ((x - min(x,na.rm = T)) / (max(x,na.rm = T) - min(x,na.rm = T)))
  }
norm_lib<-merged_lib%>%mutate_if(is.numeric, normalize)
norm_lib$n_features<-n_features-1
norm_lib%>%rowwise() %>%mutate(Total_features=sum(`Open Days`,`Open Hours`,Rating,`Total Review`,Total_V
norm_lib%>%select(`POI Name`,index)->lib_poi
colnames(lib_poi)<-c("poi_name","Index" )
head(lib_poi)
```

```
## # A tibble: 6 x 2
## # Rowwise:
##   poi_name                       Index
##   <chr>                          <dbl>
## 1 Maple Ridge Public Library     0.559
## 2 White Rock Library             0.460
## 3 Bob Prittie Metrotown Branch   0.606
## 4 Bob Prittie Metrotown Library 0.606
## 5 Cameron                        0.576
## 6 Cameron Library                0.474
```

Compute for generael amenity

```
#### Select the point of interest
poi_int<-c("museum","gallery","theatre/performance and concert hall")
df<-weights(merged_data,Amenity="museum")%>%as.data.frame()
colnames(df)
```

```
## [1] "poi_name" "Index"
```

```
df<-df[F,]

for(name in poi_int){
  tem<-weights(merged_data,Amenity=name)
  df<-rbind(invisible(tem),df)
}
```

Combine library with other amenities

```
rbind(df,lib_poi)->df_poi
poi_index<-left_join(df_poi,van_poi,by=c("poi_name"="name"))
```

3) Weight IDs comparsion with ttm

```
# clean weights
amenity_wts <- poi_index[, c('id', 'Index')]
names(amenity_wts) <- c('id', 'weight')
amenity_wts$id <-  as.factor(amenity_wts$id)
amenity_wts[!duplicated(amenity_wts$id), ]->amenity_wts
#amenity_wts %>% group_by(id) %>% summarize(n = n()) %>% arrange(desc(n))

# Check: are all the ttm amenity IDs in the weighted IDs set?
check <- all(unique(ttm$toId) %in% unique(amenity_wts$id))
# needs to be true for the join to work
paste('Are all the ttm amenity IDs in the weighted IDs set? =', check)
```

```
## [1] "Are all the ttm amenity IDs in the weighted IDs set? = FALSE"
```

Fixed unequal number of ttm amenity IDs in the weighted IDs

```
# convert Ids to  factor
ttm$fromId <- as.factor(ttm$fromId)
ttm$toId <- as.factor(ttm$toId)
# ttm ids that appear in the weights ids
ttm_id_in_wts <- unique(ttm$toId)[unique(ttm$toId) %in% unique(amenity_wts$id)]
```

```
# subset these id that not in amenity wts
ttm_id_not_in_wts <- unique(subset(ttm, !(toId %in% ttm_id_in_wts))$toId)
ttm_id_not_in_wts <- as.data.frame(list("id_not_in_wts" = ttm_id_not_in_wts))

paste('Number of id that not in amenity wts ', ttm_id_not_in_wts %>% nrow())
```

```
## [1] "Number of id that not in amenity wts  2"
```

```
# assign minimum weight on those places
ttm_id_not_in_wts$weight <- min(amenity_wts$weight)
colnames(ttm_id_not_in_wts)[1] <- "id"

# add it to amenity weights
amenity_wts <- rbind(amenity_wts, ttm_id_not_in_wts)

amenity_wts<-amenity_wts[complete.cases(amenity_wts),]
# Check: are all the ttm amenity IDs in the weighted IDs set?
check <- all(unique(ttm$toId) %in% unique(amenity_wts$id))
paste('Are all the ttm amenity IDs in the weighted IDs set? (needs to be true for the join to work) =',
```

```
## [1] "Are all the ttm amenity IDs in the weighted IDs set? (needs to be true for the join to work) = "
```

4) Export csv

amenity_wts file contains id and wieghts

```
#poi_index<-left_join(df,van_poi,by=c("poi_name"="name"))
write.csv(amenity_wts,'../../data/amenity_weights/amenity_wts.csv',row.names = FALSE)
```