Clustering Municipalities by Building Typologies and SecHaz variables (For Counterfactual Testing)

Brain K. Masinde

```
# Environment:

#   Cleaning working environment
rm(list = ls())

#   Loading libraries
library(here)
library(cluster)
library(tibble)
library(purrr)
library(dplyr)
```

```
# read melor
melor15_CF_data <- read.csv(here("data", "melor15_CF_data2.csv"))

nrow(melor15_CF_data)
```

```
## [1] 1590
```

```
# # we need the renaming function for cleaning
# source(here("R", "col_rename.R"))
#
# base_data_regions <- read.csv(here("data", "base_data_regions.csv"))
#
# base_data_regions <- col_rename(base_data_regions)
#
# nrow(base_data_regions)
```

Clustering municipalities across regions

I want to find municipalities that are more or less similar to each other across the regions.

```
mun_properties  <- melor15_CF_data %>%
    distinct(Mun_Code,
             blue_ss_frac,
             blue_ls_frac,
             red_ls_frac,
             orange_ls_frac,
             yellow_ss_frac,
```

```r
             red_ss_frac,
             orange_ss_frac,
             yellow_ls_frac,
             roof_strong_wall_strong,
             roof_strong_wall_light,
             roof_strong_wall_salv,
             roof_light_wall_strong,
             roof_light_wall_light,
             roof_light_wall_salv,
             roof_salv_wall_strong,
             roof_salv_wall_light,
             roof_salv_wall_salv,
             island_groups,
             .keep_all = FALSE)

# variables I'm interested in for matching:
match_vars  <- c("blue_ss_frac",
                 "blue_ls_frac",
                 "red_ls_frac",
                 "orange_ls_frac",
                 "yellow_ss_frac",
                 "red_ss_frac",
                 "orange_ss_frac",
                 "yellow_ls_frac",
                    'roof_strong_wall_strong',
                    'roof_strong_wall_light',
                    'roof_strong_wall_salv',
                    'roof_light_wall_strong',
                    'roof_light_wall_light',
                    'roof_light_wall_salv',
                    'roof_salv_wall_strong',
                    'roof_salv_wall_light',
                    'roof_salv_wall_salv'
                   )

# Normalize the variables using z-score
mun_scaled <- mun_properties %>%
 mutate(across(all_of(match_vars), scale))


# # Split dataset by group
# group1 <- mun_properties %>% filter(island_groups == "Luzon")
# group2 <- mun_properties %>% filter(island_groups == "Visayas")
# group3 <- mun_properties %>% filter(island_groups == "Mindanao")

#Split dataset by group
group1 <- mun_scaled %>% filter(island_groups == "Luzon")
group2 <- mun_scaled %>% filter(island_groups == "Visayas")
group3 <- mun_scaled %>% filter(island_groups == "Mindanao")

# Ensure only numeric columns are used for matching
group1_data <- group1 %>% select(-Mun_Code, -island_groups)
group2_data <- group2 %>% select(-Mun_Code, -island_groups)
```

```r
group3_data <- group3 %>% select(-Mun_Code, -island_groups)


all_data <- bind_rows(
  group1 %>% mutate(island_region = "Luzon"),
  group2 %>% mutate(island_region = "Visayas"),
  group3 %>% mutate(island_region = "Mindanao")
)

# Remove non-numeric columns except for Mun_Code and region
all_numeric <- all_data %>% select(-Mun_Code, -island_groups, -island_region)

# Perform clustering
set.seed(4838)  # For reproducibility
k <- 5 # Number of clusters (adjust as needed)
clusters <- kmeans(all_numeric, centers = k, nstart = 50)

# Add cluster assignments back to the data
all_data$Cluster <- clusters$cluster

# Create a tibble summarizing cluster sizes and municipality codes
cluster_summary <- all_data %>%
  group_by(Cluster) %>%
  summarise(
    Luzon = list(Mun_Code[island_region == "Luzon"]),
    Visayas = list(Mun_Code[island_region == "Visayas"]),
    Mindanao = list(Mun_Code[island_region == "Mindanao"])
  )



# Print outputs
print(cluster_summary)  # Summarized tibble with Mun_Code
```

```
## # A tibble: 5 x 4
##   Cluster Luzon       Visayas     Mindanao
##     <int> <list>      <list>      <list>
## 1       1 <chr [172]> <chr [40]>  <chr [103]>
## 2       2 <chr [42]>  <chr [3]>   <chr [10]>
## 3       3 <chr [86]>  <chr [220]> <chr [315]>
## 4       4 <chr [11]>  <chr [41]>  <chr [8]>
## 5       5 <chr [473]> <chr [47]>  <chr [19]>
```

```r
# Clean up:
#   Removing the outlier cluster 3
#   Get the row id of the cluster 3 observations
#cluster3_id <- which(all_data$Cluster==3)
#all_data <- all_data[-cluster3_id, ]


#   change column Cluster from numerical to charactor/factor
#all_data  <- all_data  %>%
#  mutate(Cluster = as.character(Cluster)) %>%
#  mutate(Cluster = as.factor(Cluster))
```

```
# Join: inner join counterfactual dataset with cluster dataset
#   Counterfactual dataset = melor15_CF_data
#   Cluster dataset = all_data
#   Join by Mun_code

melor15_CF_data <- melor15_CF_data %>%
  inner_join(all_data %>% select(Mun_Code, Cluster), by = "Mun_Code")
```

```
# Column clean up and create new

# columns to remove:
cols_to_remove <- c("X",
                    "rain_max6h",
                    "rain_max24h",
                    "ls_risk_pct",
                    "ss_risk_pct",
                    "slope_mean",
                    "elev_mean",
                    "ruggedness_sd",
                    "ruggedness_mean",
                    "slope_sd",
                    "poverty_pct",
                    "has_coast",
                    "coast_length",
                    "housing_units",
                    "vulnerable_groups",
                    "pantawid_benef",
                    "damage_perc",
                    "Mun_Code_2",
                    "Unnamed..0",
                    "X10.Digit.Code",
                    "Correspondence.Code",
                    "Income.Class",
                    "Population.2020.Census." )

clustered_M15_CF_data <- melor15_CF_data %>%
  select(-all_of(cols_to_remove))
```

```
# Create a tibble summarizing cluster sizes and municipality codes
cluster_summary <- clustered_M15_CF_data %>%
  group_by(Cluster) %>%
  summarise(
    Luzon = list(Mun_Code[island_groups == "Luzon"]),
    Visayas = list(Mun_Code[island_groups == "Visayas"]),
    Mindanao = list(Mun_Code[island_groups == "Mindanao"])
  )



# Print outputs
print(cluster_summary)  # Summarized tibble with Mun_Code
```

```
## # A tibble: 5 x 4
##   Cluster Luzon       Visayas     Mindanao
##     <int> <list>      <list>      <list>
## 1       1 <chr [172]> <chr [40]>  <chr [103]>
## 2       2 <chr [42]>  <chr [3]>   <chr [10]>
## 3       3 <chr [86]>  <chr [220]> <chr [315]>
## 4       4 <chr [11]>  <chr [41]>  <chr [8]>
## 5       5 <chr [473]> <chr [47]>  <chr [19]>
```

Characteristics of cluster 5

```r
visayas_cluster5 <- clustered_M15_CF_data %>%
  filter(island_groups == "Visayas", Cluster == "5") %>%
  select(roof_strong_wall_strong,
                    roof_strong_wall_light,
                    roof_strong_wall_salv,
                    roof_light_wall_strong,
                    roof_light_wall_light,
                    roof_light_wall_salv,
                    roof_salv_wall_strong,
                    roof_salv_wall_light,
                    roof_salv_wall_salv)


luzon_cluster5 <- clustered_M15_CF_data %>%
  filter(island_groups == "Luzon", Cluster == "5") %>%
  select(roof_strong_wall_strong,
                    roof_strong_wall_light,
                    roof_strong_wall_salv,
                    roof_light_wall_strong,
                    roof_light_wall_light,
                    roof_light_wall_salv,
                    roof_salv_wall_strong,
                    roof_salv_wall_light,
                    roof_salv_wall_salv)

mindanao_cluster5 <- clustered_M15_CF_data %>%
  filter(island_groups == "Mindanao", Cluster == "5") %>%
  select(roof_strong_wall_strong,
                    roof_strong_wall_light,
                    roof_strong_wall_salv,
                    roof_light_wall_strong,
                    roof_light_wall_light,
                    roof_light_wall_salv,
                    roof_salv_wall_strong,
                    roof_salv_wall_light,
                    roof_salv_wall_salv)
```

Output

```
# Output:
#   Save the clustered counterfactual dataset

write.csv(clustered_M15_CF_data, file = here("data", "clustered_M15_CF_data2.csv"))
```