Toronto's Underpriviledged Neighbourhoods are Close to Each Other

Ke-Li Chiu & Diego Mamanche Castellanos

29/02/2020

Abstract

The City of Toronto consists of 140 neighbourhoods. While each of the neighbourhood has its own strength and vitality, some neighbourhoods share similar socioeconomic traits. In this study, we intend to identify neighbourhoods that have disadvantaged socioeconomic status through a clustering analysis using hierarchical clustering. Features of low-income rate, lack of education and racial diversity are employed to cluster the neighbourhoods into two groups. There are 34 neighbourhoods in group 1, while the other 106 are assigned to group 2. The 34 neighbourhoods in group 1 have a higher percentage of the population that have low-income status, possess no educational certification, and are visible minorities; the neighbourhoods in group 1 are seemed as more disadvantaged. The clusters of neighbourhoods are plotted as a choropleth map, which allows us to examine the spatial relationship between disadvantaged neighbourhoods. What we discover is that the disadvantaged the neighbourhoods not only share similar socioeconomic characteristics, but they are also geographically close to each other.

Introduction

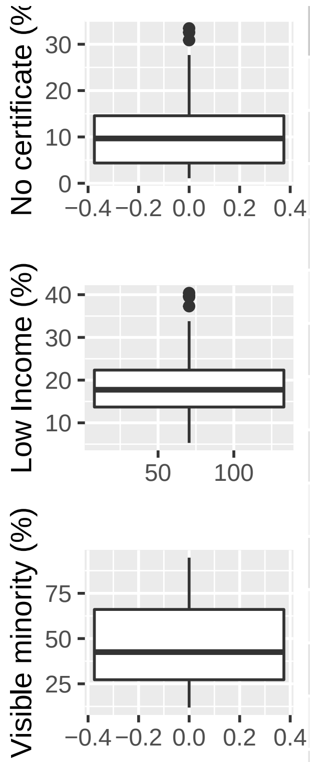
There are 140 neighbourhoods officially recognized by the City of Toronto, and the divisions are used for internal planning purposes. Many shortcomings of economic or social policy planning and making come from the over-reliance on the "one-size-fits-all" approach that overlooks their differences in socioeconomic status. However, customizing policies for 140 neighbourhoods is also improbable. Can we then create groups of neighbourhoods that share similar socioeconomic traits so policy-makers can plan strategically and efficiently? Can we identify the group of neighbourhoods that are vulnerable in their social and economic development, so they receive more attention from the Government? In this study, clustering analysis is conducted to be used to define neighbourhood types and display their characteristics and geographic proximity.

Troubling differences in poverty, education and diversity exist among neighbourhoods in the City of Toronto. According to David Hulchanski, a professor of the University of Toronto who uses the 2016 census to create demographic charts of Toronto, the city is segregated by race and income. For example, visible minorities are concentrated in low-income neighbourhoods, and white residents are dominating affluent areas in numbers far higher than their share of the population. The segregation pointed out by Hulchanski leads to the choices of traits we use to conduct the clustering analysis for the neighbourhoods — low-income rate, the population percentage of residents who have no educational certification, and visible minority population percentage. We intend to articulate the social stratification and spatial stratification in the City of Toronto.

Dataset and Method Description

Dataset

To address the research question, the City of Toronto, through the Open Toronto Data Portal, offers a comprehensive dataset called Neighbourhood Profiles. The data set contains several categories divided by topics, that in turn, are broken down into different characteristics, presented in a total of 2383 rows. As for the neighbourhoods, all of them are displayed as columns. For this analysis, the variables "no certificate, diploma, or degree 2", "18 to 64 years %", and "total visible minority population" will feed into the model to answer the research question. The variables are renamed to “no certificate,” “visible minority,” and “low income.” Descriptive statistics of the variables are shown in Figure 1.



stats	No certificate	Low Income	Visible minority
nbr.val	140.00	140.00	140.00
nbr.null	0.00	0.00	0.00
nbr.na	0.00	0.00	0.00
min	1.08	5.30	11.86
max	33.44	40.40	94.70
range	32.36	35.10	82.84
sum	1462.46	2587.30	6507.23
median	9.67	17.75	42.52
mean	10.45	18.48	46.48
SE.mean	0.61	0.60	1.86
CI.mean.0.95	1.20	1.18	3.67
var	51.30	49.92	482.36
std.dev	7.16	7.07	21.96
coef.var	0.69	0.38	0.47

Figure 1: Descriptive statistics of the data

Method

Exposing the most disadvantaged neighbourhoods in the City of Toronto is a classification case. The goal is to group all neighbourhoods by clusters based on how similar they are. To achieve this, the classification method used is hierarchical clustering, an unsupervised machine learning technique that partitions a set of objects with similar characteristics into subsets.

Different approaches were analyzed. Clustering techniques such as K-means, Density-based classification, and Hierarchical clustering were tested out with the dataset, finding the last one the most suitable for this investigation. Hierarchical clustering allows us to view at once, each possible number of clusters (k) through the dendrogram, which is a tree representation of those clusters. Moreover, the hierarchical method needs no k in advance. The undetermination of k is important because the purpose of this study is to find the most disadvantaged neighbourhoods without pre-assumption on the number of groups there should be. Lastly, by using the Elbow curve, the best number of clusters (k) can be impartially calculated.

Limitation of the Dataset and Approach

In 2016, the long-form Census became mandatory once again and was distributed to one out of every four households (25%). Income data were gathered solely by linking with administrative data (Canada Revenue Agency). Census data is subject to error, such as coverage, non-response, and sampling errors. Moreover, although the Census is mandatory, it is still likely to be affected by Response Bias because people do not always provide accurate or truthful information. Error is especially common in the collection of income data. Correct income data are often more challenging to obtain because the wealthiest households are more prompted to underreport their incomes. Therefore, income inequality based on data from household surveys are likely to be underestimated.

Another limitation lies in the lack of time-series data to be compared with the data sourced from 2016. This prevents a holistic observation in the change of socioeconomic dynamics in the neighbourhoods across time. The neighbourhoods may not be assigned to the same clusters based on data sourced from past years. Such shifts would provide insights that are not available in this study.

As the approach we employed in the clustering analysis, Hierarchical clustering has some limitations, such as sensitivity to noise and outliers. Moreover, once the grouping is performed, it can never be undone. Lastly, for large data sets, performing Hierarchical clustering has a high computational cost.

Ethical Considerations

Features such as race or income can be highly predictive for certain problems. The three variables in poverty, education, and visible minority population in this clustering analysis do not provide details of underlying contributors to the social division in the neighbourhoods. Without further investigation, the study possibly shapes the categories of perception and classification through which readers internalize social divisions. These categories would shape the way we envision class structure and social problems associated with the neighbourhoods and paved the way for implicit stigmatization. More crucially, the incorporation of these categories would influence decisions to buy properties and, for parents, to send children to the local schools.

Hierarchical Clustering of Toronto's Neighbourhoods

Calculating distance between observations (Neighbourhoods)

Firstly, Euclidean Distance Between Toronto's neighbourhoods is calculated and used to construct a dendrogram, shown in Figure 2. The height of the branch points indicates how similar or different the neighbourhoods are from each other; the greater the height, the greater the difference.

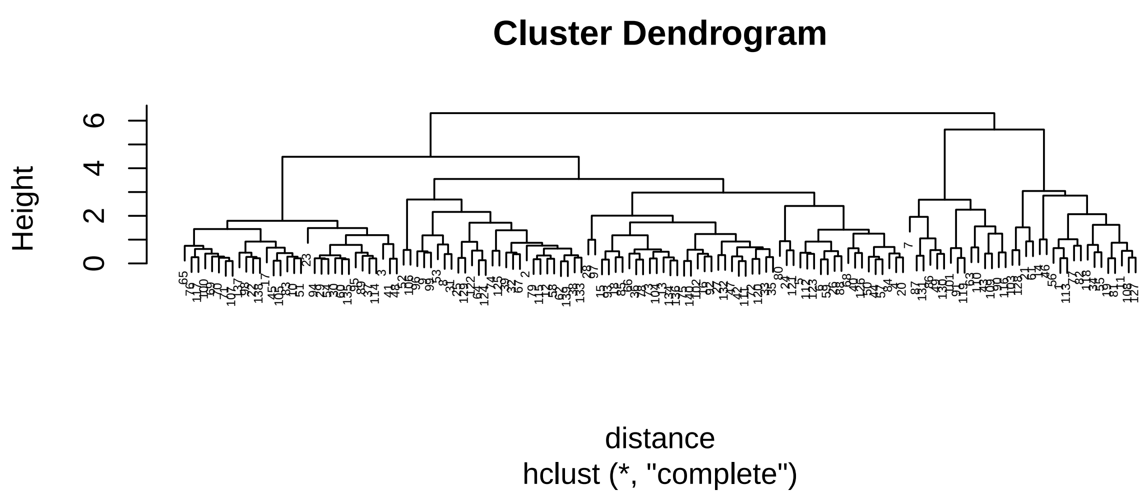


Figure 2: Cluster dendrogram that illustrats Euclidean Distances between the neighbourhoods

Determine optimal number of clusters

The Elbow Method is used to determine the optimal number of clusters. Viewing the Scree Plot shown in Figure 3, the "elbow" on the arm is "2" on the x-axis. Therefore, the neighbourhoods will be assigned to two different clusters. In Figure 4, the neighbourhoods are grouped in the cluster dendrogram by the red rectangles.

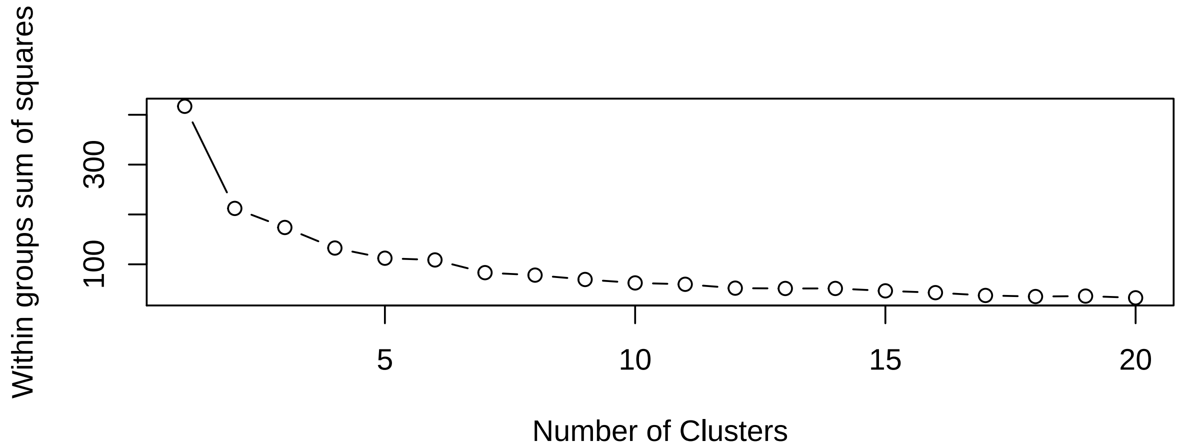


Figure 3: Scree plot that indicates the optimal nuber of clusters is 2

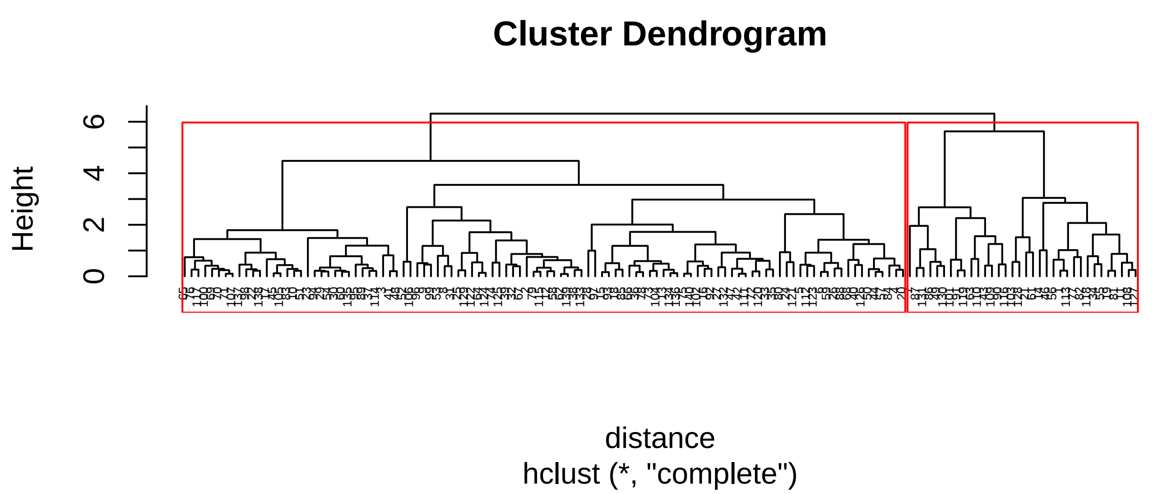


Figure 4: Grouping neghbourhoods into two clusters

Assessing Characteristics of Neighbourhood Clusters

The two clusters have different numbers of members. As shown in Table 1, in Group 1, there are 34 neighbourhoods while the other 106 are assigned to Group 2. Table 2 showcases the distances between the means of the two clusters in normalized data values. The distinction between the two clusters is clear. As shown in Table 2, all values in Group 1 are positive and all values in Group 2 are all negative. The positive values in Group 1 indicate its characteristics in higher percentage in poverty, lack of education and visibile minority population.

member	Freq
1	34
2	106

Table 1: Cluster membership

Group.1	no_certificate	low_income	visible_minority
1	0.987	1.051	0.904
2	−0.317	−0.337	−0.29

Table 2: Cluster mean distance in normalized data values

Clusters Distribution

The grouped scattor plots in Figure 5 show that there is a pattern of neighbourhoods distribution with the three selected variables. In general, the 34 neighbourhoods in Group 1 are located at the top-right area of the graphs, which the position visualizes their characteristics in higher percentage in poverty, lack of education and visibile minority population. The exact differences in the two groups can be assessed in the summary tables of the two clusters in Table 3 and Table 4.

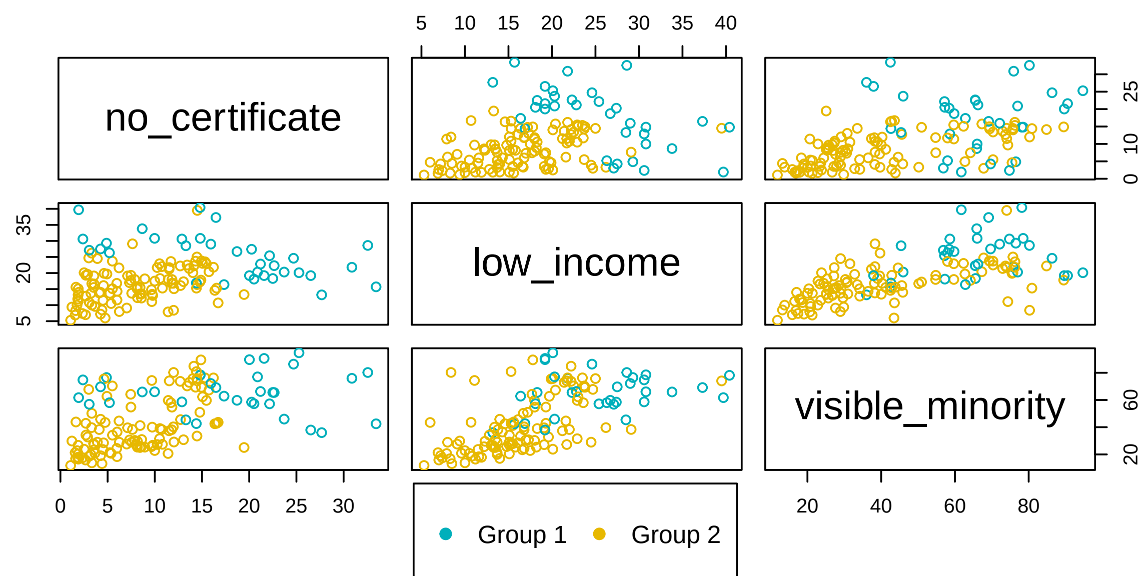


Figure 5: Grouped scatter plots

no_certificate	low_income	visible_minority
Min. : 1.932	Min. :13.20	Min. :36.01
1st Qu.:12.974	1st Qu.:20.15	1st Qu.:58.12
Median :19.364	Median :26.50	Median :65.97
Mean :17.517	Mean :25.91	Mean :66.33
3rd Qu.:22.609	3rd Qu.:30.27	3rd Qu.:76.37
Max. :33.440	Max. :40.40	Max. :94.70

Table 3: Summary table of Group 1

no_certificate	low_income	visible_minority
Min. : 1.083	Min. : 5.30	Min. :11.86
1st Qu.: 3.718	1st Qu.:12.85	1st Qu.:25.15
Median : 7.758	Median :15.95	Median :33.83
Mean : 8.178	Mean :16.10	Mean :40.11
3rd Qu.:11.925	3rd Qu.:19.85	3rd Qu.:53.80
Max. :19.456	Max. :29.10	Max. :89.50

Table 4: Summary table of Group 2

Toronto's Neighbourhoods Clusters Map

The map in Figure 6 shows the distribution of neighbourhoods in the two clusters. We can see that the neighbourhoods in Group 1 not only share similar socieconomic traits, they are also geographically close to each other and are located in the peripheral areas of the city.

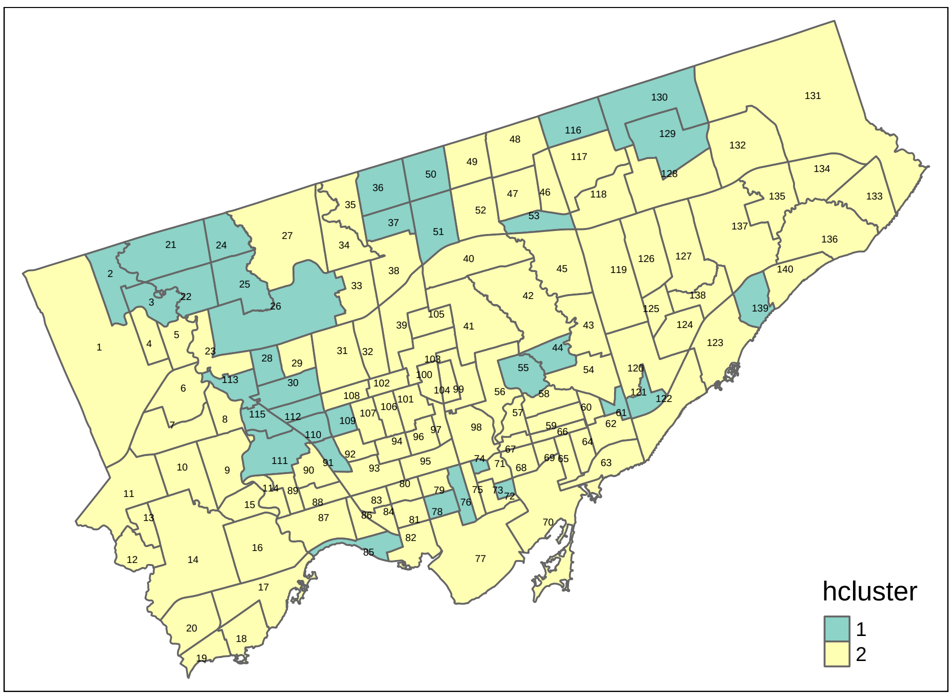


Figure 6: Map of clustered neighbourhood

Appendix A

```
knitr::opts_chunk$set(echo = FALSE, include = FALSE)
# Importing libraries
library(opendatatoronto)
library(dplyr)
library(tidyr)
library(ggplot2)
library(knitr)
library(sf)
library(tmap)
library(tmaptools)
library(leaflet)
library(corrplot) #For correlation matrix
library(car)
library(dbscan) # For density-based model
library(fpc) # For dessity-based model
library(factoextra)
library(gridExtra)
library(pastecs)
#setwd("~/Desktop/MI/INF2178 Experienment DS/Problem set 3")
# Get the resource we want from this package
neighbourhood_raw <-
  list_package_resources("6e19a90f-971c-46b3-852c-0c48c436d1fc") %>%
  filter(name == "neighbourhood-profiles-2016-csv") %>%
  get_resource()
main_df_raw <- neighbourhood_raw
main_df_raw <- as.data.frame(main_df_raw)
# Remove "X2011 prefix from all column names"
### GEO-LOCATION DATASET TORONTO BY NEIGHBOURHOODS ###
# get package
package <- show_package("4def3f65-2a65-4a4f-83c4-b2a4aed72d46")
package
# get all resources for this package
resources <- list_package_resources("4def3f65-2a65-4a4f-83c4-b2a4aed72d46")
# identify datastore resources; by default, Toronto Open Data sets datastore
# resource format to CSV for non-geospatial and GeoJSON for geospatial resources
datastore_resources <- filter(resources, tolower(format) %in% c('csv', 'geojson'))
# load the first datastore resource as a sample
geo_data <- filter(datastore_resources, row_number()==1) %>% get_resource()
### Cleaning geo-location dataset
clean_geo_data <- janitor::clean_names(geo_data)
clean_geo_data <- tidyr::extract(clean_geo_data, area_name,
                         into = "neighbourhoods" ,
                         regex = "([^(0-9)]+)")
clean_geo_data["neighbourhoods"] <-
  janitor::make_clean_names(as.matrix(clean_geo_data["neighbourhoods"]))
clean_geo_data <- janitor::clean_names(clean_geo_data)
clean_geo_data <- select(clean_geo_data, neighbourhoods, longitude, latitude, geometry)
filter(clean_geo_data, neighbourhoods %in% c("mimico","weston_pellam_park"))
clean_geo_data["neighbourhoods"][c(17,67),] <-
  c("mimico_includes_humber_bay_shores","weston_pelham_park")
clean_geo_data
```

```r
#Save neighbourhoods into a dataframe
col_names_2011 <- as.data.frame(colnames(main_df_raw))
col_names_2011 <- col_names_2011[7:nrow(col_names_2011),]
col_names_2011 <- as.data.frame(col_names_2011)
colnames(col_names_2011) <- "neighbourhoods"
#Filter education per neighbourhood
main_df <- filter(main_df_raw,
                      Category == "Neighbourhood Information" |
                      Category == "Income" |
                      Characteristic == "Population, 2016" |
                      Characteristic == "Rate of unsuitable housing"|
                      Topic == "Visible minority population" |
                      Topic == "Highest certificate, diploma or degree" |
                      Topic == "Low income in 2015")
# Reshape the dataframe (swap row and columns)
main_df_reshaped <- data.frame(t(main_df[-1]))
colnames(main_df_reshaped) <- main_df[, 1]
# Slice the reshaped dataframe
main_df_sliced <- main_df_reshaped %>%
  dplyr::slice(4:nrow(main_df_reshaped))
# Turn characteristics to column names
names(main_df_sliced) <- as.matrix(main_df_sliced[1, ])
main_df_sliced <- main_df_sliced[-1, ]
main_df_sliced[] <- lapply(main_df_sliced, function(x) type.convert(as.character(x)))
# Clean column names
library(janitor)
main_df_sliced <- main_df_sliced %>% clean_names()
main_df_sliced$total_population_aged_15_years_and_over_by_major_field_of_study_classification_of_instru
main_df_sliced$x18_to_64_years_percent
main_df_sliced = main_df_sliced[-1,]
main_df_sliced <- mutate(main_df_sliced, neighbourhoods = col_names_2011$neighbourhoods)
# Select wanted columns to make a new dataframe
df_cleaned <- main_df_sliced %>%
  select(
    "neighbourhood_number",
    "neighbourhoods",
    "population_2016",
    "total_highest_certificate_diploma_or_degree_for_the_population_aged_25_to_64_years_in_private_hous

    # education
    "no_certificate_diploma_or_degree_2",
    # low income percentage
    "x18_to_64_years_percent",

    # visible minority
    "total_visible_minority_population"
  )
# transform df to numeric
df_cleaned$population_2016 =
  as.numeric(gsub(",", "", df_cleaned$population_2016))
df_cleaned$total_visible_minority_population =
  as.numeric(gsub(",", "", df_cleaned$total_visible_minority_population))
df_cleaned
```

```
## Turn absolute values to percentage
total_population_education <-
  df_cleaned$total_highest_certificate_diploma_or_degree_for_the_population_aged_25_to_64_years_in_priva
# Education data transformation
df_cleaned <-
  mutate(df_cleaned, no_certificate_diploma_or_degree_2 =
           (df_cleaned$no_certificate_diploma_or_degree_2/total_population_education)*100)
# Visible miniority data transformation
df_cleaned <-
  mutate(df_cleaned, total_visible_minority_population =
           (df_cleaned$total_visible_minority_population/df_cleaned$population_2016)*100)
df_cleaned
# Compute descriptive statistics boxplots
ggplot(df_cleaned) +
  aes(y = no_certificate_diploma_or_degree_2) +
  geom_boxplot() +
  labs(x = "", y = "No certificate (%)") -> p1
ggplot(df_cleaned) +
  aes(x = 1:nrow(df_cleaned), y = x18_to_64_years_percent) +
  #geom_bar(stat="identity", width=1)
  geom_boxplot() +
  labs(x = "", y = "Low Income (%)") -> p2
ggplot(df_cleaned) +
  aes(y = total_visible_minority_population) +
  geom_boxplot() +
  labs(x = "", y = "Visible minority (%)") -> p3

#tt2 <- ttheme_default(core=list(fg_params=list(hjust=.5, x=0.4)),
#                      rowhead=list(fg_params=list(hjust=.5, x=0.4)))
grid1 <- grid.arrange(p1, p2, p3, ncol = 1, nrow = 3)
# Compute descriptive statistics
tt1 <- ttheme_default()
stats_table <- stat.desc(df_cleaned[,c(5:7)])
stats_table <- round(stats_table, 2)
stats_table <- mutate(stats_table, stats = row.names(stats_table))
stats_table <- select(stats_table, stats, no_certificate_diploma_or_degree_2,
                      x18_to_64_years_percent, total_visible_minority_population)
colnames(stats_table) <- c("stats","No certificate","Low Income","Visible minority")
grid2 <- grid.arrange(tableGrob(stats_table, theme = tt1, rows = NULL), ncol = 1, nrow = 1)
#kable(stats_table)
#grid.arrange(grid1, grid2, ncol = 2, nrow = 1, padding = 1, heights=c(2,3))
 grid.arrange(arrangeGrob(grid1, ncol=1, nrow=1),
         arrangeGrob(grid2, ncol=1, nrow=1), heights=c(28,1), widths=c(1,3),
         bottom = "Figure 1: Descriptive statistics of the data")
###draw correlation matrix of the numeric independent variables only
num_data <- df_cleaned[,-c(1:2)] ### only numeric independent vars
colnames(num_data) <- c("population","total_highest_cert",
                        "no_certificate","18_to_64_per","visible_minority")
correlationMatrix <- cor(num_data, method = "pearson")
correlationMatrix    # a 6x6 matrix
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(correlationMatrix, method="color", col=col(200),
         type="upper", order="hclust",
```

```r
         tl.col="black", tl.srt=45, tl.cex= 0.7, #Text label color and rotation
         # Combine with significance
         sig.level = 0.01,
         # hide correlation coefficient on the principal diagonal
         diag=FALSE
)
# Rename columns
df_cleaned <- rename(df_cleaned,
    no_certificate = no_certificate_diploma_or_degree_2,
    low_income = x18_to_64_years_percent,
    visible_minority = total_visible_minority_population
    )
# Normalize data by subtracting the mean and dividing it by the standard deviation.
df_selected_columns = df_cleaned[,c(
   "no_certificate",
   "low_income",
   "visible_minority"
  )]
means = apply(df_selected_columns,2,mean)
sds = apply(df_selected_columns,2,sd)
nor = scale(df_selected_columns,means,sds)
nor
###draw correlation matrix of the numeric variables only
num_data <- as.data.frame(nor) ### only numeric independent vars
colnames(num_data) <- c("no_certificate","18_to_64_per","visible_minority")
correlationMatrix <- cor(num_data, method = "pearson")
correlationMatrix    # a 6x6 matrix
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(correlationMatrix, method="color", col=col(200),
         type="upper", order="hclust",
         tl.col="black", tl.srt=45, tl.cex= 0.7, #Text label color and rotation
         # Combine with significance
         sig.level = 0.01,
         # hide correlation coefficient on the principal diagonal
         diag=FALSE
)
# Get the distance of the normalized data
distance = dist(nor)
plot_data <- nor
# Hierarchical agglomerative clustering using default complete linkage
plot_data_comp = hclust(distance)
plot(plot_data_comp,labels=plot_data_comp$ID,main='Cluster Dendrogram', cex=.4)
# Scree Plot
wss <- (nrow(nor)-1)*sum(apply(nor,2,var))
for (i in 2:20) wss[i] <- sum(kmeans(nor, centers=i)$withinss)
plot(1:20, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
plot(plot_data_comp,hang=-1, cex=.4)
rect.hclust(plot_data_comp, k = 2, border = "red")
# Characterizing clusters
member <- cutree(plot_data_comp,2)
freq_df <- table(member)
freq_df <- data.frame(freq_df)
group1_grob <- grid.arrange(tableGrob(freq_df, theme = tt1, rows = NULL),
```

```
                             ncol = 1, nrow = 1,
                             bottom = "Table 1: Cluster membership")
member_df <- as.data.frame(member)
variables_df <- aggregate(nor,list(member),mean)
variables_df <- round(variables_df,3)
group1_grob <- grid.arrange(tableGrob(variables_df, theme = tt1, rows = NULL),
                             ncol = 1, nrow = 1,
                             bottom = "Table 2: Cluster mean distance in normalized data values")
#get hclust label to neighbourhoods
hcluster <- cutree(plot_data_comp, k=2)
hcluster <- as.data.frame(hcluster)
df_w_vector <- mutate(df_cleaned, hcluster = hcluster$hcluster)
#merge dataset with geo dataset
df_w_vector["neighbourhoods"] <-
  janitor::make_clean_names(as.matrix(df_w_vector["neighbourhoods"]))
merged_df <- merge(df_w_vector, clean_geo_data, by = 'neighbourhoods')
merged_df$hcluster <- as.factor(merged_df$hcluster)
sf_merged_df <- st_sf(merged_df, sf_column_name = "geometry")
colors <- c("#00AFBB", "#E7B800", "#FC4E07", "#000000")
colors <- colors[merged_df$hcluster]
plot(df_selected_columns, col = colors)
par(xpd = T, mar = par()$mar + c(0,0,0,7))
legend(.36, .08, horiz = TRUE,
       legend = c("Group 1", "Group 2"),
       col =unique(colors),
       pch = 19, bty = "o",
       cex = .8)
group1 <- df_selected_columns %>% filter(hcluster == "1")
group1 <- summary(group1)
kable(group1)
group2 <- df_selected_columns %>% filter(hcluster == "2")
group2 <- summary(group2)
kable(group2)
group1_grob <- grid.arrange(tableGrob(group1, theme = tt1, rows = NULL),
                             ncol = 1, nrow = 1,
                             bottom = "Table 3: Summary table of Group 1")
group2_grob <- grid.arrange(tableGrob(group2, theme = tt1, rows = NULL),
                             ncol = 1, nrow = 1,
                             bottom = "Table 4: Summary table of Group 2")
 grid.arrange(arrangeGrob(group1_grob,  ncol=1, nrow=1, name = "title"),
              arrangeGrob(group2_grob, ncol=1, nrow=1), heights=c(4,1), widths=c(1,1))
#Plot all majors without palette
tmap_mode("plot")
tm_shape(sf_merged_df) +
tm_layout(legend.show = TRUE, legend.position =
            c("right", "bottom"), title.size = 2,
          title.position = c("center","center")) +
  tm_polygons(c("hcluster"), style = "pretty") +
  tm_text("neighbourhood_number",
          auto.placement = TRUE, xmod = 0, size = 0.3)+
  tm_facets(sync = TRUE, ncol = 1)
```

References

DBSCAN: Density-Based Clustering Essentials. (n.d.). Retrieved February 27, 2020, from https://www.datanovia.com/en/lessons/dbscan-density-based-clustering-essentials/

Department of Justice, & Research and Statistics Division. (2015, January 7). A One-Day Snapshot of Aboriginal Youth in Custody Across Canada. Retrieved February 27, 2020, from https://www.justice.gc.ca/eng/rp-pr/cj-jp/yj-jj/yj1-jj1/p1_6.html

Open Data Dataset. (n.d.). Retrieved January 30, 2020, from https://open.toronto.ca/dataset/ wellbeing-toronto-demographics-nhs-indicators/

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Sharla Gelfand (2019). opendatatoronto: Access the City of Toronto Open Data Portal. R package version 0.1.1. https://CRAN.R-project.org/package=opendatatoronto

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2019). dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr

Hadley Wickham and Lionel Henry (2019). tidyr: Tidy Messy Data. R package version 1.0.0. https://CRAN.R-project.org/package=tidyr

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Yihui Xie (2019). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.25.

Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible

Matt Dowle and Arun Srinivasan (2019). data.table: Extension of `data.frame`. R package version 1.12.6. https://CRAN.R-project.org/package=data.table Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

Pebesma, E., 2018. Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446, https://doi.org/10.32614/RJ-2018-009

Statistics Canada. (2019, May 1). Census of Population. Retrieved February 27, 2020, from https://www23.statcan.gc.ca/imdb/p2SV.pl?Function=getSurvey&SDDS=3901

Tennekes M (2018). "tmap: Thematic Maps in R." *Journal of Statistical Software*, *84*(6), 1-39. doi: 10.18637/jss.v084.i06 (URL: https://doi.org/10.18637/jss.v084.i06).

Martijn Tennekes (2019). tmaptools: Thematic Map Tools. R package version 2.0-2. https://CRAN.R-project.org/package=tmaptools

Joe Cheng, Bhaskar Karambelkar and Yihui Xie (2019). leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library. R package version 2.0.3. https://CRAN.R-project.org/package=leaflet