Statistical Learning Project - Unsupervised Learning

```r
#https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```r
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.6.3

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```r
library(tidyr)
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 3.6.3

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

## Note: As of version 1.0.0, cowplot does not change the

##   default ggplot2 theme anymore. To recover the previous

##   behavior, execute:
##   theme_set(theme_cowplot())

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

##
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggmap':
##
##     theme_nothing
```

```r
library(magick)
```

```
## Warning: package 'magick' was built under R version 3.6.3

## Linking to ImageMagick 6.9.9.14
## Enabled features: cairo, freetype, fftw, ghostscript, lcms, pango, rsvg, webp
## Disabled features: fontconfig, x11
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#world_map <- map_data("newyork")
```

Read Dataset

```
ds = read.csv("AB_NYC_2019.csv")
head(ds)
```

```
##     id                                             name host_id   host_name
## 1 2539               Clean & quiet apt home by the park    2787        John
## 2 2595                            Skylit Midtown Castle    2845    Jennifer
## 3 3647              THE VILLAGE OF HARLEM....NEW YORK !    4632   Elisabeth
## 4 3831                  Cozy Entire Floor of Brownstone    4869 LisaRoxanne
## 5 5022 Entire Apt: Spacious Studio/Loft by central park    7192       Laura
## 6 5099        Large Cozy 1 BR Apartment In Midtown East    7322       Chris
##   neighbourhood_group neighbourhood latitude longitude       room_type price
## 1            Brooklyn    Kensington 40.64749 -73.97237    Private room   149
## 2           Manhattan       Midtown 40.75362 -73.98377 Entire home/apt   225
## 3           Manhattan        Harlem 40.80902 -73.94190    Private room   150
## 4            Brooklyn  Clinton Hill 40.68514 -73.95976 Entire home/apt    89
## 5           Manhattan   East Harlem 40.79851 -73.94399 Entire home/apt    80
## 6           Manhattan   Murray Hill 40.74767 -73.97500 Entire home/apt   200
##   minimum_nights number_of_reviews last_review reviews_per_month
## 1              1                 9  2018-10-19              0.21
## 2              1                45  2019-05-21              0.38
## 3              3                 0                            NA
## 4              1               270  2019-07-05              4.64
## 5             10                 9  2018-11-19              0.10
## 6              3                74  2019-06-22              0.59
##   calculated_host_listings_count availability_365
## 1                              6              365
## 2                              2              355
## 3                              1              365
## 4                              1              194
## 5                              1                0
## 6                              1              129
```

Data cleaning

Check for NA and NULL values

```
#Check for NA
apply(ds,2,function(x) sum(is.na(x)))
```

```
##                             id                           name
##                              0                              0
##                        host_id                      host_name
##                              0                              0
##            neighbourhood_group                  neighbourhood
##                              0                              0
##                       latitude                      longitude
##                              0                              0
##                      room_type                          price
##                              0                              0
##                 minimum_nights              number_of_reviews
##                              0                              0
##                    last_review              reviews_per_month
##                              0                          10052
## calculated_host_listings_count               availability_365
##                              0                              0
```

```
# NOTES
# Remove NA, empty
#
#
#
#
```

Normalisation and selection of the variables

```
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}


clean_data = function(ds)
{
  ds = select (ds,-c(host_id, id, host_name, name,minimum_nights,number_of_reviews,
                     neighbourhood,last_review,availability_365,

                     reviews_per_month,calculated_host_listings_count))


  numerical = c("price","longitude", "latitude")
  categorical = c("neighbourhood_group")

  ds[numerical] = scale(ds[numerical])
  ds$neighbourhood_group = factor(ds$neighbourhood_group,
                                  level= c("Brooklyn","Manhattan",
                                           "Queens","Staten Island", "Bronx"),
                                  labels=c(1,2,3,4,5))
  ds$room_type = factor(ds$room_type,
                        level= c("Private room","Entire home/apt","Shared room"),
                        labels=c(1,2,3))

  return(ds)
}
#ggdraw() +
#  draw_image("New_York_City_.png") +
#  draw_plot(myplot)

dataset = clean_data(ds)

head(dataset)
```

```
##   neighbourhood_group   latitude  longitude room_type       price
## 1                   1 -1.4938339 -0.4376476         1 -0.01549291
## 2                   2  0.4524314 -0.6846321         2  0.30097047
## 3                   2  1.4683845  0.2224944         1 -0.01132892
## 4                   1 -0.8033893 -0.1644481         2 -0.26533242
## 5                   2  1.2756468  0.1772139         2 -0.30280835
## 6                   2  0.3433173 -0.4946274         2  0.19687067
```

========================= K-MEANS ===============

#x: numeric matrix, numeric data frame or a numeric vector #centers: Possible values are the number of clusters (k) or a set of initial (distinct) cluster centers. If a number, a random set of (distinct) rows in x is chosen as the initial centers. #iter.max: The maximum number of iterations allowed. Default value is 10. #nstart: The number of random starting partitions when centers is a number. Trying nstart > 1 is often recommended.

```r
km.res = kmeans(dataset, 4, nstart = 25)

cat("First 10 Clusters association",km.res$cluster[1:10])
```

```
## First 10 Clusters association 3 4 4 3 4 4 3 4 4 4
```

```r
cat("\nCenters")
```

```
## 
## Centers
```

```r
print(km.res$centers)
```

```
##   neighbourhood_group   latitude   longitude room_type      price
## 1            1.868421  0.1287874 -0.50832977  1.824561 14.7620392
## 2            3.346637  0.3881026  1.78314575  1.427095 -0.2611466
## 3            1.002839 -0.8078700  0.01786529  1.515912 -0.1419676
## 4            2.051265  0.6141866 -0.51376305  1.650961  0.1253757
```

```r
cat("\ntotss",km.res$totss)
```

```
## 
## totss 195866.3
```

```r
cat("\nwithinss",km.res$withinss)
```

```
## 
## withinss 9312.047 22019.68 21164.81 39332.74
```

```r
cat("\ntot_withinss",km.res$tot.withinss)
```

```
## 
## tot_withinss 91829.28
```

```r
cat("\nbetweenss",km.res$betweenss)
```

```
## 
## betweenss 104037.1
```

```r
cat("\nSize",km.res$size)
```

```
## 
## Size 114 6289 20079 22413
```

```
cat("\niter",km.res$iter)
```

```
##
## iter 4
```

```
cat("\nifault",km.res$ifault)
```

```
##
## ifault 0
```

To create a beautiful graph of the clusters generated with the kmeans() function, will use the factoextra package.

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.6.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

Cluster number for each of the observations

```
head(km.res$cluster)
```

```
## [1] 3 4 4 3 4 4
```

Cluster size

```
km.res$size
```

```
## [1]   114  6289 20079 22413
```

Cluster means

```
km.res$centers
```

```
##   neighbourhood_group   latitude   longitude room_type      price
## 1            1.868421  0.1287874 -0.50832977  1.824561 14.7620392
## 2            3.346637  0.3881026  1.78314575  1.427095 -0.2611466
## 3            1.002839 -0.8078700  0.01786529  1.515912 -0.1419676
## 4            2.051265  0.6141866 -0.51376305  1.650961  0.1253757
```

```
#dataset$neighbourhood_group = as.numeric( dataset$neighbourhood_group)
#dataset$room_type = as.numeric(  dataset$room_type)
#fviz_cluster(km.res, data = dataset,
#             palette = c("#00AFBB","#2E9FDF", "#E7B800", "#FC4E07"),
#             ggtheme = theme_minimal(),
#             main = "Partitioning Clustering Plot"
#)

#res <- hcut(dataset, k = 4, stand = FALSE)
#fviz_dend(km.res, rect = TRUE, cex = 0.5,
#          k_colors = c("#00AFBB","#2E9FDF", "#E7B800", "#FC4E07"))
```

PAM ALGORITHM

https://towardsdatascience.com/clustering-on-mixed-type-data-8bbd0a2569c3

```
library(cluster)
library(readr)
library(Rtsne)
```

```
## Warning: package 'Rtsne' was built under R version 3.6.3
```

Compute Gower distance

```
dim(dataset)
```

```
## [1] 48895     5
```

```
smp_size <- floor(0.9 * nrow(dataset))
set.seed(123)

train_ind <- sample(seq_len(nrow(dataset)), size = smp_size)

prova = dataset[-train_ind,]
pam.res <- pam(prova, 4)

gower_dist <- daisy(prova, metric = "gower")
```

```
start.time <- Sys.time()
sil_width <- c(NA)
for(i in 2:8){
  pam_fit <- pam(gower_dist, diss = TRUE, k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
}


end.time <- Sys.time()
time.taken <- end.time - start.time

print("-- Time: -- ")
```

```
## [1] "-- Time: -- "
```

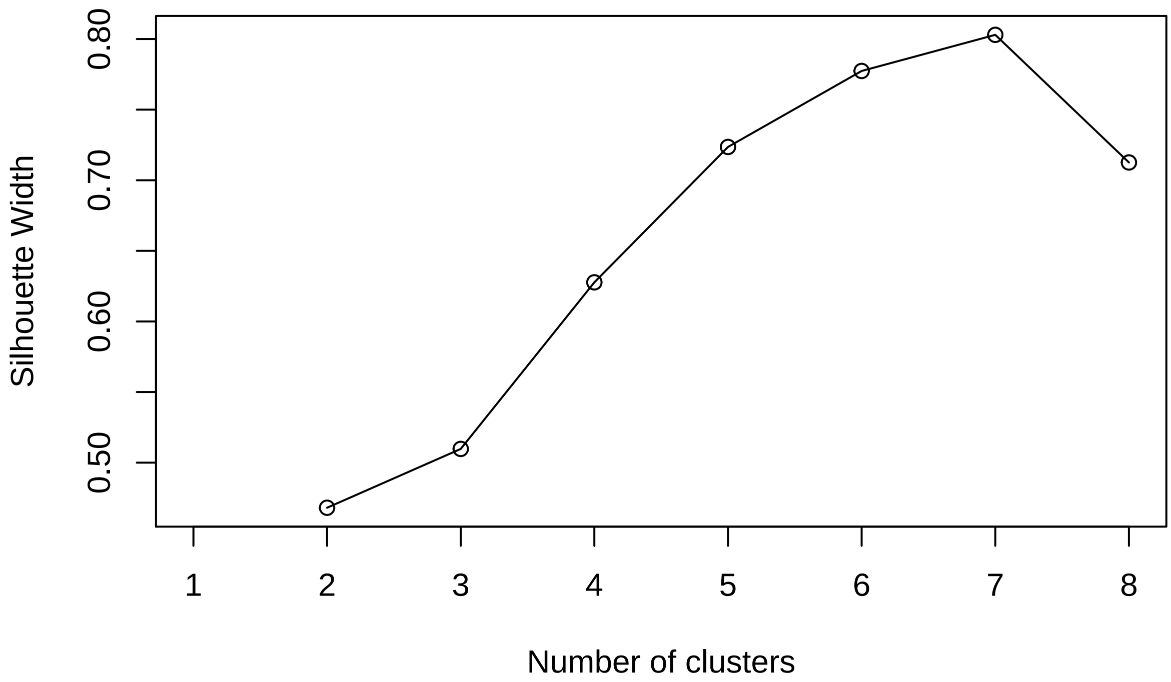
```
time.taken
```

```
## Time difference of 1.86202 mins
```

```
print("")
```

```
## [1] ""
```

```
plot(1:8, sil_width,
      xlab = "Number of clusters",
      ylab = "Silhouette Width")
lines(1:8, sil_width)
```



======================== FAMD ================

http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/115-famd-factor-analysis-of-mixed-data-in-r-essentials/

#https://nextjournal.com/pc-methods/calculate-pc-mixed-data

#https://cran.r-project.org/web/packages/FactoMineR/index.html #https://stats.stackexchange.com/questions/5774/can-principal-component-analysis-be-applied-to-datasets-containing-a-mix-of-cont