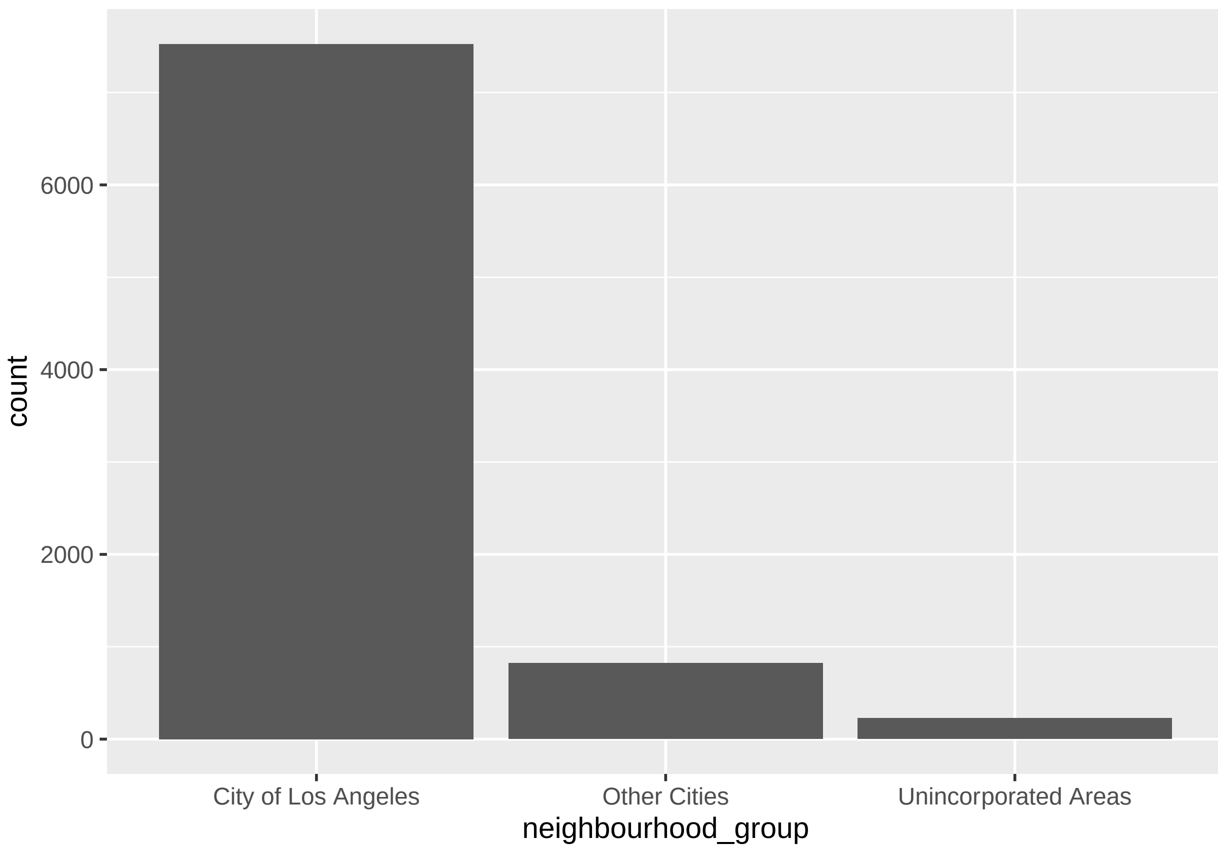
Airbnb Location Recommender: (1) K-means Clustering (2) Decision Trees

Import all library

Read final jointed table

```
# final_data = as_tibble(read.csv("./data/all_jointed_data.csv"))
final_data = as_tibble(read.csv("./input/all_data_v3.csv"))

ggplot(final_data) + geom_bar(aes(x=neighbourhood_group))
```




```
# data_dict(final_data, print_table = 'Yes')
```

convert price into unit price (price/room_type)

```
# 4, 2, 2, 1
final_data = final_data %>%
```

```r
  mutate(unit_price = ifelse(room_type=='Entire home/apt',price/4,ifelse(room_type=='Shared room',price
# drop outlier
final_data = final_data %>%
  filter(unit_price<1100)
```

one-hot-encode room_type and neighbourhood (https://community.rstudio.com/t/factor-to-one-hot-encoding-aka-dummy-variables-using-logicals/69236/2)

```r
final_data_OHE = final_data %>%
  mutate(n = 1) %>%
  pivot_wider(names_from = neighbourhood_group, values_from=n, values_fill=0) %>%
  mutate(n = 1) %>%
  pivot_wider(names_from = room_type, values_from=n, values_fill=0)
```

Group by neighborhood and average on the other attributes

```r
final_data_OHE %>% filter(neighbourhood=='Hollywood Hills')
```

```
## # A tibble: 344 x 18
##         id name          neighbourhood latitude longitude price availability_365
##      <int> <chr>         <chr>            <dbl>     <dbl> <int>            <int>
##  1  206662 "Hollywood &~ Hollywood Hi~     34.1     -118.   115               83
##  2   20786 "Mondrian-In~ Hollywood Hi~     34.1     -118.   199               53
##  3   22355 "Experience ~ Hollywood Hi~     34.1     -118.   171               63
##  4   23979 "Sleek New Y~ Hollywood Hi~     34.1     -118.   178               68
##  5   51546 "Cool Pad Un~ Hollywood Hi~     34.1     -118.   100              365
##  6   63416 "VIEWS! HOLL~ Hollywood Hi~     34.1     -118.   131              363
##  7  611841 "In the hill~ Hollywood Hi~     34.1     -118.   170              343
##  8 1535628 "The Little ~ Hollywood Hi~     34.1     -118.   118               81
##  9 1660593 "Hollywood H~ Hollywood Hi~     34.1     -118.    85              282
## 10 1700085 "Soothing Ze~ Hollywood Hi~     34.1     -118.   304               83
## # i 334 more rows
## # i 11 more variables: avg_rating <dbl>, crime_level <dbl>,
## #   crime_incidents_count <int>, unit_price <dbl>, ‘City of Los Angeles‘ <dbl>,
## #   ‘Other Cities‘ <dbl>, ‘Unincorporated Areas‘ <dbl>, ‘Private room‘ <dbl>,
## #   ‘Entire home/apt‘ <dbl>, ‘Shared room‘ <dbl>, ‘Hotel room‘ <dbl>
```

```r
grouped_neighbourhood_data = final_data_OHE %>%
                                group_by(neighbourhood) %>%
                                summarise(shared_room_count = sum(`Shared room`),
                                          private_room_count = sum(`Private room`),
                                          Entire_home_apt_count = sum(`Entire home/apt`),
                                          Hotel_room_count = sum(`Hotel room`),
                                          avg_unit_price = mean(unit_price),
                                          avg_crime_level = mean(crime_level),
                                          avg_crime_incidents_count = mean(crime_incidents_count),
                                          avg_rating = mean(avg_rating),
                                          latitude = mean(latitude),
                                          longitude = mean(longitude),
                                          housing_count = n()
                                          )
```

summarize the new grouping df

```r
# data_dict
data_dict(grouped_neighbourhood_data, print_table = 'Yes')
```

	VariableType	MissingValues	n	mean	sd	median	se	min
_count	numeric	0	141	187.65	177.53	133.15	14.95	3
el	numeric	0	141	2.26	0.17	2.27	0.01	1.76
	numeric	0	141	3.52	0.16	3.53	0.01	2.82
e	numeric	0	141	44.19	26.65	37	2.24	10.25
count	numeric	0	141	45.97	87.3	18	7.35	0
unt	numeric	0	141	0.41	1.81	0	0.15	0
t	integer	0	141	60.82	102.65	28	8.64	1
	numeric	0	141	34.06	0.11	34.06	0.01	33.73
	numeric	0	141	−118.35	0.1	−118.35	0.01	−118.63
d	character	0	141					
unt	numeric	0	141	12.76	16.27	7	1.37	0
unt	numeric	0	141	1.67	4.29	0	0.36	0

```r
# summarize_numeric
numeric_data = grouped_neighbourhood_data %>% select(avg_unit_price, avg_rating, avg_crime_level, avg_c
summarize_numeric(numeric_data)
```

```
##                   Attribute Missing Values Unique Values        Mean       Min
## 1            avg_unit_price              0           140  44.1892292 10.250000
## 2                avg_rating              0           139   3.5191650  2.820500
## 3           avg_crime_level              0           141   2.2554408  1.757919
## 4 avg_crime_incidents_count              0           140 187.6457626  3.000000
## 5         shared_room_count              0            17   1.6737589  0.000000
## 6        private_room_count              0            38  12.7588652  0.000000
## 7      Entire_home_apt_count              0            67  45.9716312  0.000000
## 8          Hotel_room_count              0             8   0.4113475  0.000000
##             Max          SD
## 1    218.166667  26.6507907
## 2      4.000000   0.1571248
## 3      2.933333   0.1746000
## 4   1118.888361 177.5324483
## 5     30.000000   4.2903487
## 6    100.000000  16.2691209
```

```
## 7  701.000000  87.3016890
## 8   15.000000   1.8129334
```

```
summarize_numeric(grouped_neighbourhood_data)
```

```
##                    Attribute Missing Values Unique Values          Mean
## 1           shared_room_count              0            17     1.6737589
## 2          private_room_count              0            38    12.7588652
## 3      Entire_home_apt_count              0            67    45.9716312
## 4            Hotel_room_count              0             8     0.4113475
## 5              avg_unit_price              0           140    44.1892292
## 6             avg_crime_level              0           141     2.2554408
## 7  avg_crime_incidents_count              0           140   187.6457626
## 8                  avg_rating              0           139     3.5191650
## 9                    latitude              0           137    34.0639525
## 10                  longitude              0           130  -118.3545219
## 11              housing_count              0            77    60.8156028
##            Min         Max          SD
## 1     0.000000   30.000000   4.2903487
## 2     0.000000  100.000000  16.2691209
## 3     0.000000  701.000000  87.3016890
## 4     0.000000   15.000000   1.8129334
## 5    10.250000  218.166667  26.6507907
## 6     1.757919    2.933333   0.1746000
## 7     3.000000 1118.888361 177.5324483
## 8     2.820500    4.000000   0.1571248
## 9    33.728605   34.305000   0.1122166
## 10 -118.634062 -118.160000   0.1017037
## 11    1.000000  783.000000 102.6484015
```

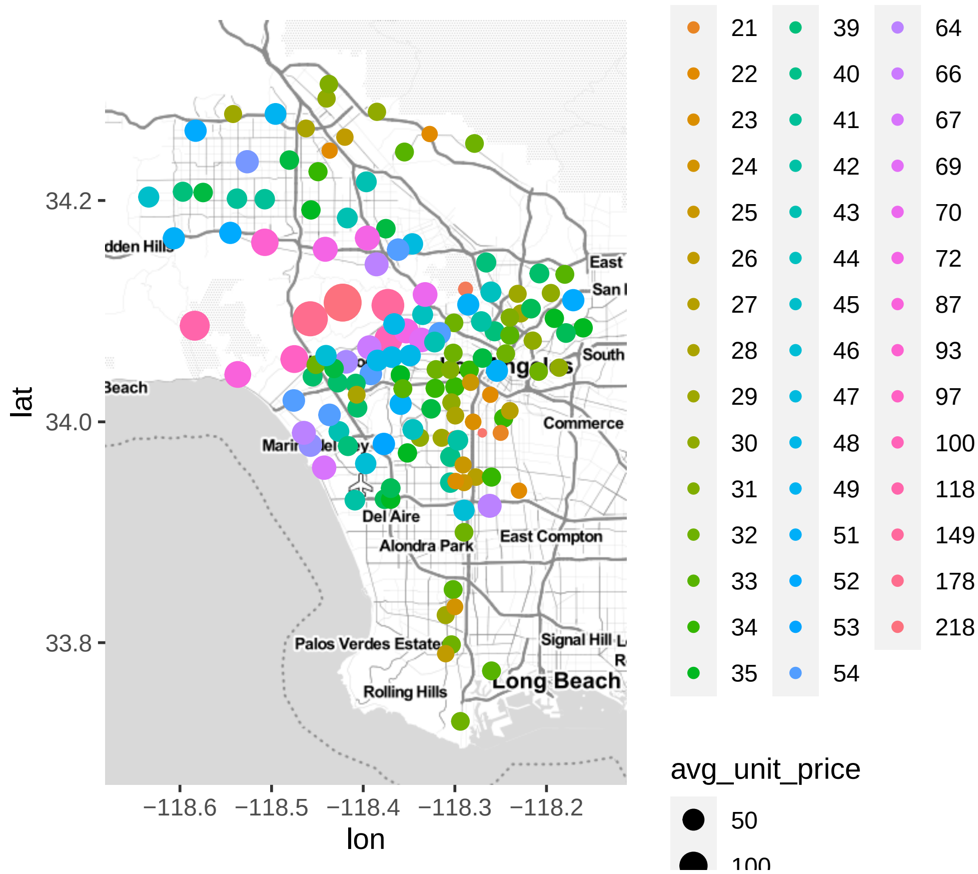
Plot avg_unit_price neighbourhood in LA map

```
grouped_neighbourhood_data_roundprice = cbind(grouped_neighbourhood_data)
grouped_neighbourhood_data_roundprice$avg_unit_price =
  round(grouped_neighbourhood_data_roundprice$avg_unit_price, 0)
height <- max(grouped_neighbourhood_data$latitude) - min(grouped_neighbourhood_data$latitude)
width <- max(grouped_neighbourhood_data$longitude) - min(grouped_neighbourhood_data$longitude)
LA_borders <- c(bottom  = min(grouped_neighbourhood_data$latitude)  - 0.1 * height,
                 top     = max(grouped_neighbourhood_data$latitude)  + 0.1 * height,
                 left    = min(grouped_neighbourhood_data$longitude) - 0.1 * width,
                 right   = max(grouped_neighbourhood_data$longitude) + 0.1 * width)

map <- get_stamenmap(LA_borders, zoom = 10, maptype = "toner-lite")
```

```
## i Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.
```

```
ggmap(map) +
    geom_point(data = grouped_neighbourhood_data_roundprice, mapping = aes(x = longitude, y = latitude,
```



exclude unimportant attributes

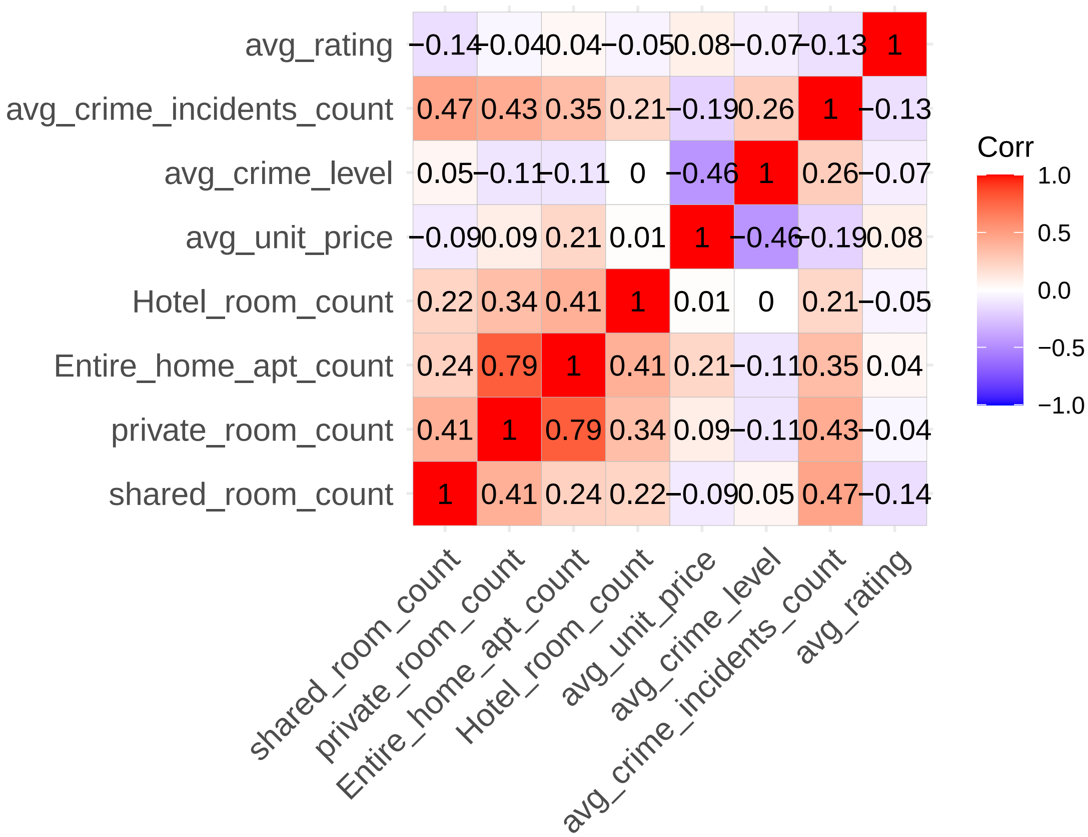
```
train_data = select(grouped_neighbourhood_data, -c(neighbourhood, housing_count, latitude, longitude))
```

Correlation matrix

```
# install.packages("ggcorrplot")
library(ggcorrplot)
```

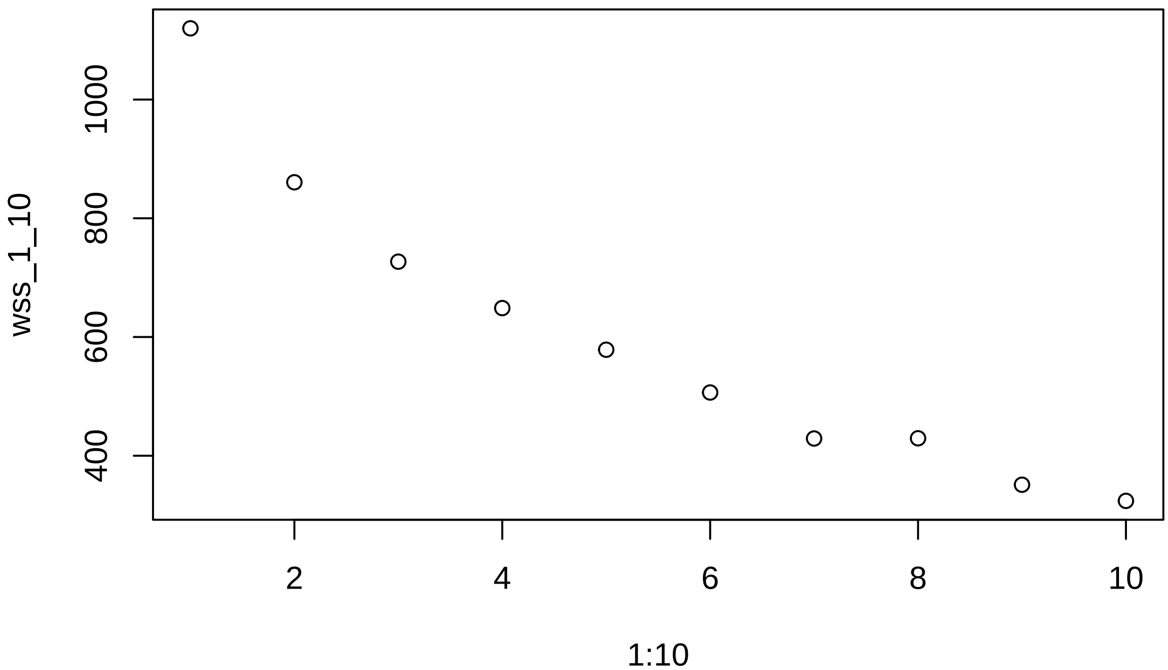
```
## Warning: package 'ggcorrplot' was built under R version 4.1.2
```

```
# cor_data = select(final_data, c(unit_price, availability_365, avg_rating, crime_level, crime_incident
fullCorrMatrix = round(cor(train_data %>% select_if(is.numeric)), 2)
ggcorrplot(fullCorrMatrix, lab = TRUE)
```



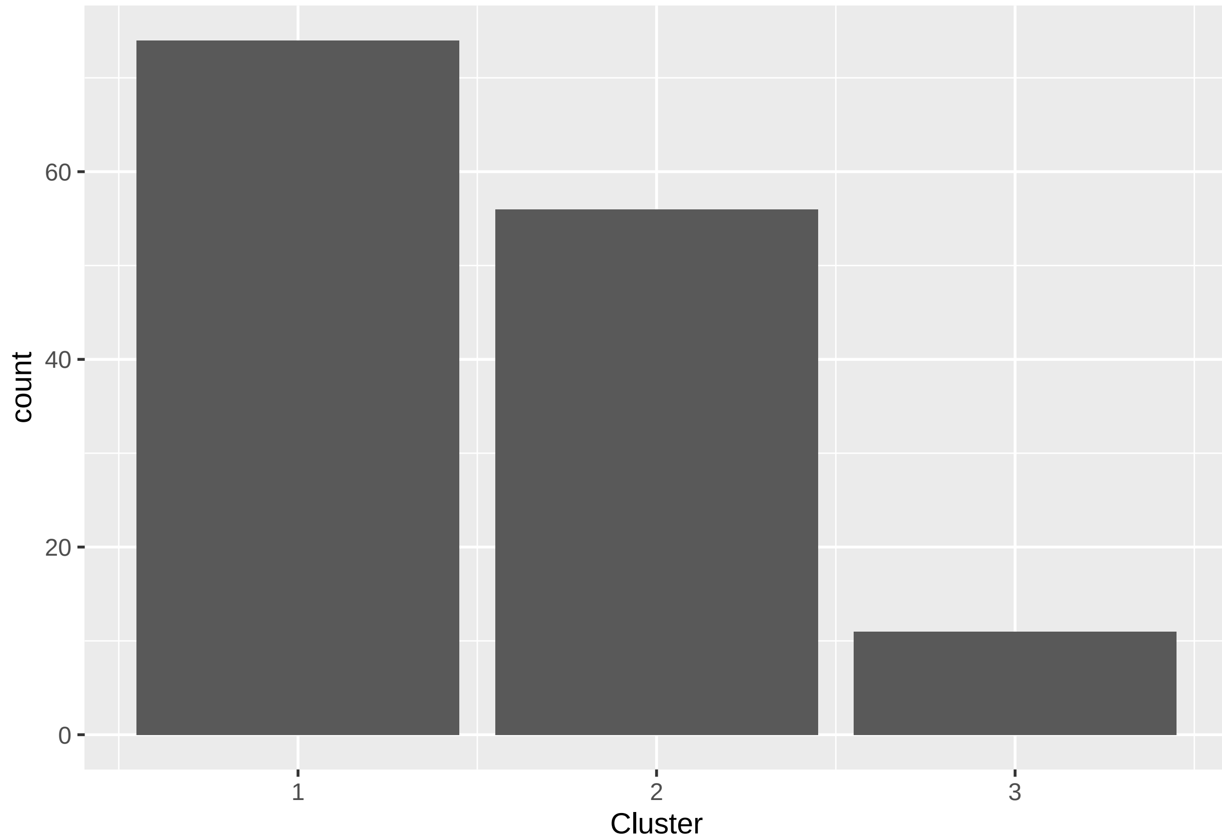
Elbow diagram

```
wss = function (k) {kmeans(scale(train_data), k, nstart=10)$tot.withinss}
wss_1_10 = map_dbl(1:10, wss) # with k from 1 to 10, means call wss function 10 times to do kmeans
plot(1:10, wss_1_10)
```



Look at a clustering with k = 3

```
res = kmeans(scale(train_data), 3, nstart=25)
grouped_neighbourhood_data_res = grouped_neighbourhood_data %>% mutate(Cluster = res$cluster)
ggplot(grouped_neighbourhood_data_res) + geom_bar(aes(x=Cluster))
```



summarise for each cluster

```
# summarize_numeric = function(dataset) {
#
#   dataset = select_if(dataset, is.numeric)
#   summary.table = data.frame(Attribute = names(dataset))
#
#   summary.table = summary.table %>%
#     mutate('Missing Values' = apply(dataset, 2, function (x) sum(is.na(x))),
#            'Unique Values' = apply(dataset, 2, function (x) length(unique(x))),
#            'Mean' = colMeans(dataset, na.rm = TRUE),
#            'Min' = apply(dataset, 2, function (x) min(x, na.rm = TRUE)),
#            'Max' = apply(dataset, 2, function (x) max(x, na.rm = TRUE)),
#            'SD' = apply(dataset, 2, function (x) sd(x, na.rm = TRUE))
#     )
#   summary.table
# }
########## cluster1 #############
grouped_neighbourhood_data_res %>%
  filter(Cluster==1) %>%
  summarize_numeric()
```

```
##                  Attribute Missing Values Unique Values       Mean
## 1         shared_room_count              0             9  0.6216216
## 2        private_room_count              0            20  6.7837838
## 3      Entire_home_apt_count              0            31 14.0945946
```

```
## 4           Hotel_room_count              0              1     0.0000000
## 5              avg_unit_price              0             74    32.7485664
## 6             avg_crime_level              0             74     2.3722163
## 7  avg_crime_incidents_count              0             73   189.5236720
## 8                  avg_rating              0             73     3.5108247
## 9                    latitude              0             71    34.0522670
## 10                  longitude              0             66 -118.3197023
## 11              housing_count              0             39    21.5000000
## 12                    Cluster              0              1     1.0000000
##            Min         Max           SD
## 1     0.000000    9.000000   1.78043171
## 2     0.000000   29.000000   7.05413562
## 3     0.000000   60.000000  15.83144860
## 4     0.000000    0.000000   0.00000000
## 5    10.250000   64.150000   8.15358222
## 6     2.122482    2.933333   0.12202629
## 7     3.000000  547.000000 140.28194074
## 8     2.820500    4.000000   0.18107084
## 9    33.728605   34.305000   0.12992155
## 10 -118.596842 -118.160000   0.09219431
## 11    1.000000   86.000000  21.09096411
## 12    1.000000    1.000000   0.00000000
```

```r
########## cluster2 #############
grouped_neighbourhood_data_res %>%
  filter(Cluster==2) %>%
  summarize_numeric()
```

```
##                    Attribute Missing Values Unique Values          Mean
## 1           shared_room_count              0              7     0.9642857
## 2          private_room_count              0             28    14.7321429
## 3       Entire_home_apt_count              0             44    58.6071429
## 4            Hotel_room_count              0              4     0.1607143
## 5              avg_unit_price              0             55    60.3219077
## 6             avg_crime_level              0             56     2.0962760
## 7  avg_crime_incidents_count              0             56   118.8611297
## 8                  avg_rating              0             56     3.5454023
## 9                    latitude              0             56    34.0867822
## 10                  longitude              0             55 -118.4047390
## 11              housing_count              0             47    74.4642857
## 12                    Cluster              0              1     2.0000000
##            Min         Max          SD
## 1     0.000000   12.000000  2.03571998
## 2     0.000000   54.000000 13.65615555
## 3     0.000000  304.000000 61.63460615
## 4     0.000000    3.000000  0.56493880
## 5    17.854167  218.166667 35.34840017
## 6     1.757919    2.289968  0.11367910
## 7     3.769231  292.686047 76.51909508
## 8     2.996143    4.000000  0.12641473
## 9    33.929091   34.278095  0.08503142
## 10 -118.634062 -118.170769  0.09769215
## 11    1.000000  344.000000 72.87372997
## 12    2.000000    2.000000  0.00000000
```

```r
########## cluster3 #############
grouped_neighbourhood_data_res %>%
  filter(Cluster==3) %>%
  summarize_numeric()
```

```
##                    Attribute Missing Values Unique Values        Mean
## 1          shared_room_count              0            11   12.363636
## 2         private_room_count              0            11   42.909091
## 3      Entire_home_apt_count              0            11  196.090909
## 4           Hotel_room_count              0             6    4.454545
## 5             avg_unit_price              0            11   39.023689
## 6            avg_crime_level              0            11    2.280154
## 7  avg_crime_incidents_count              0            11  525.188866
## 8                 avg_rating              0            11    3.441701
## 9                   latitude              0            11   34.026340
## 10                 longitude              0            11 -118.333111
## 11             housing_count              0            11  255.818182
## 12                   Cluster              0             1    3.000000
##             Min         Max           SD
## 1      0.000000   30.000000   8.66340265
## 2      4.000000  100.000000  32.10437523
## 3      4.000000  701.000000 225.33550743
## 4      0.000000   15.000000   4.96716491
## 5     21.723684   63.780013  11.43198371
## 6      2.156374    2.371429   0.05972505
## 7    103.360000 1118.888361 336.95410358
## 8      3.277816    3.570078   0.08618687
## 9     33.774400   34.096989   0.08844506
## 10 -118.464789 -118.254038   0.08169448
## 11    25.000000  783.000000 246.47142560
## 12     3.000000    3.000000   0.00000000
```

o

```r
library(ggplot2)  # The grammar of graphics package
library(maps)     # Provides latitude and longitude data for various maps
```

```
## Warning: package 'maps' was built under R version 4.1.2
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

o

```r
# ggplot(test1) + geom_point(aes(x=avg_severity_level, y = price, color=factor(Cluster), shape=factor(C
# ggplot(train_data_res) + geom_count(aes(x=longitude, y = latitude, color=factor(Cluster))) + ylim(33.
# original
```

```r
# ggplot(train_data_res) + geom_count(aes(x=longitude, y = latitude)) + ylim(33.6, 34.4) + xlim(-118.1,
# ggplot(test1) + geom_histogram(aes(x=price)) + xlim(0,1000)
# test1 = test1 %>%
#   mutate(price_level = ifelse(price<=100,0,ifelse(price>200,2,1)))
# ggplot(test1) + geom_bar(aes(x=price_level))
# ggplot(test1) + geom_count(aes(x=longitude, y = latitude, color=factor(price_level))) + ylim(33.6, 34
```

o

```r
# MainStates <- map_data("county", region = 'california')
# ggplot() +
#   geom_polygon( data=MainStates, aes(x=long, y=lat, group=group),
#                 color="black", fill="lightyellow" )
```

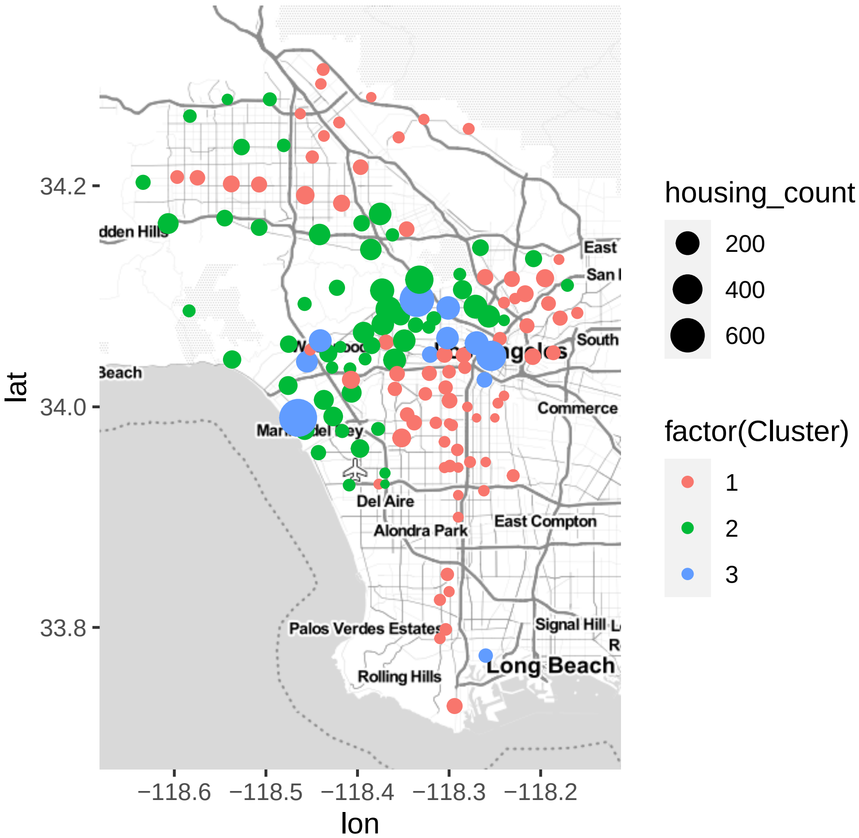
Plot on LA map https://rstudio-pubs-static.s3.amazonaws.com/434852_bb92f97f268148a9b963754d8dc5d95a.html

```r
# install.packages('ggmap')
library(ggmap)
height <- max(grouped_neighbourhood_data$latitude) - min(grouped_neighbourhood_data$latitude)
width <- max(grouped_neighbourhood_data$longitude) - min(grouped_neighbourhood_data$longitude)
LA_borders <- c(bottom  = min(grouped_neighbourhood_data$latitude)  - 0.1 * height,
                 top     = max(grouped_neighbourhood_data$latitude)  + 0.1 * height,
                 left    = min(grouped_neighbourhood_data$longitude) - 0.1 * width,
                 right   = max(grouped_neighbourhood_data$longitude) + 0.1 * width)

map <- get_stamenmap(LA_borders, zoom = 10, maptype = "toner-lite")
```

```
## i Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.
```

```r
ggmap(map) +
    geom_point(data = grouped_neighbourhood_data_res, mapping = aes(x = longitude, y = latitude, color=
```



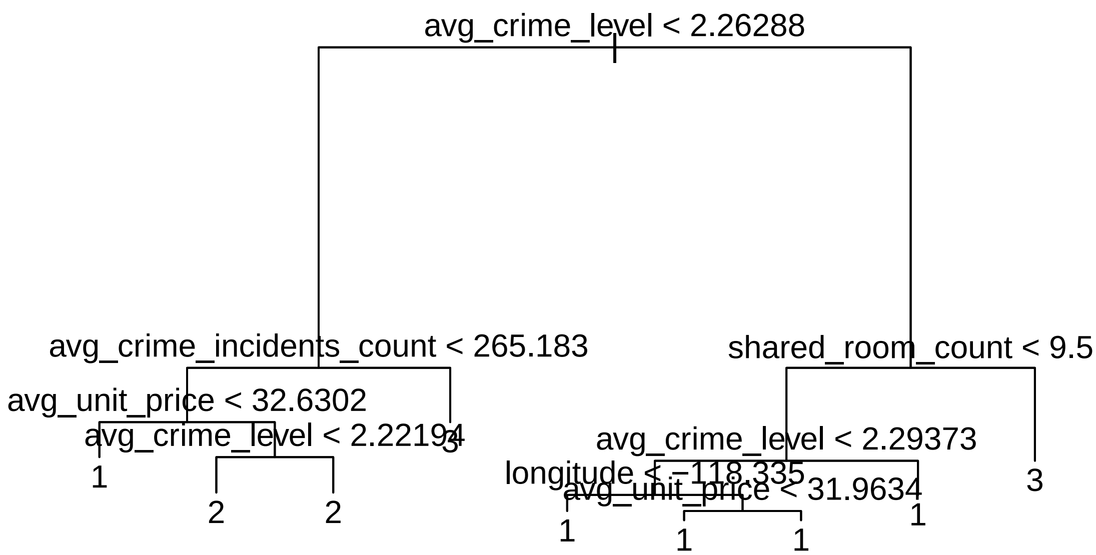
##

Whole

```
train_DT = select(grouped_neighbourhood_data_res, -c(neighbourhood))
regr_tree= tree(factor(Cluster) ~ .-Cluster, data=train_DT)
```

```
## Warning in terms.formula(formula, data = data): 'varlist' has changed (from
## nvar=12) to new 13 after EncodeVars() -- should no longer happen!
```

```
plot(regr_tree); text(regr_tree, pretty=0)
```

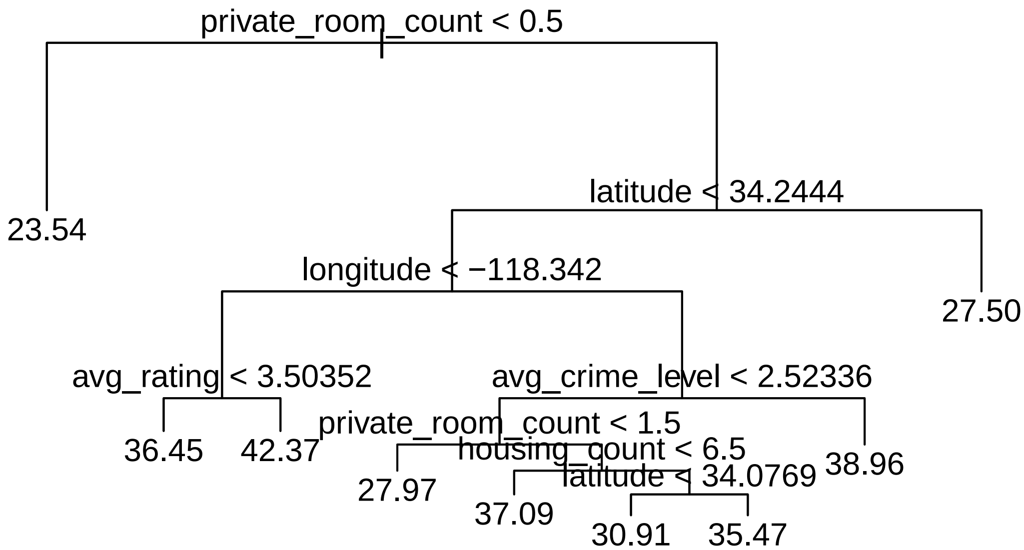


cluster 1 DT

```
data_c1 = grouped_neighbourhood_data_res %>%
  filter(Cluster==1)
train_DT = select(data_c1, -c(neighbourhood))
regr_tree= tree(avg_unit_price ~ .-avg_unit_price, data=data_c1)
```

```
## Warning in tree(avg_unit_price ~ . - avg_unit_price, data = data_c1): NAs
## introduced by coercion
```

```
plot(regr_tree); text(regr_tree, pretty=0)
```

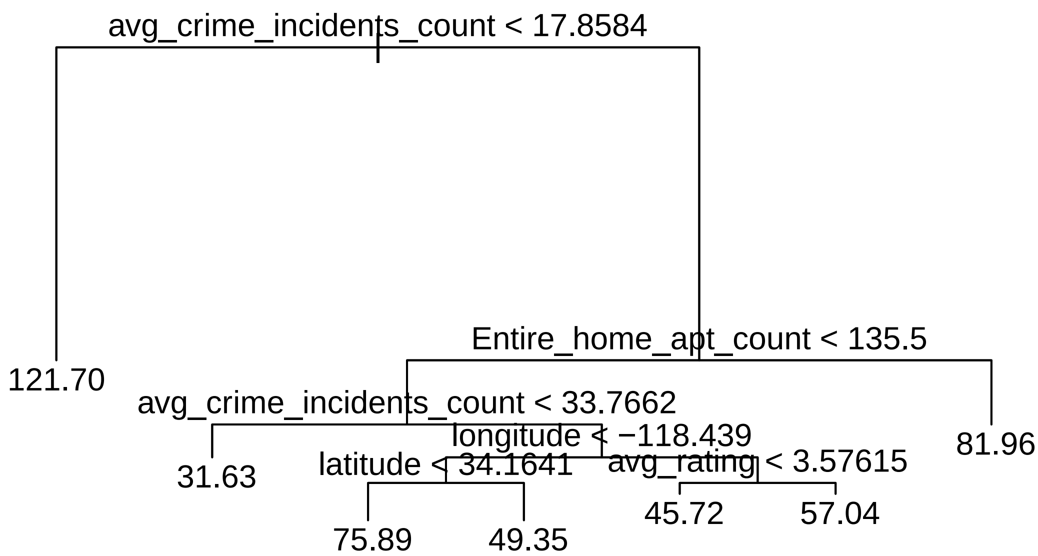


cluster 2 DT

```
data_c2 = grouped_neighbourhood_data_res %>%
  filter(Cluster==2)
train_DT = select(data_c2, -c(neighbourhood))
regr_tree= tree(avg_unit_price ~ .-avg_unit_price, data=data_c2)
```

```
## Warning in tree(avg_unit_price ~ . - avg_unit_price, data = data_c2): NAs
## introduced by coercion
```

```
plot(regr_tree); text(regr_tree, pretty=0)
```



cluster 3 DT

```
data_c3 = grouped_neighbourhood_data_res %>%
  filter(Cluster==3)
train_DT = select(data_c3, -c(neighbourhood))
regr_tree= tree(avg_unit_price ~ .-avg_unit_price, data=data_c3)
```

```
## Warning in tree(avg_unit_price ~ . - avg_unit_price, data = data_c3): NAs
## introduced by coercion
```

```
plot(regr_tree); text(regr_tree, pretty=0)
```

##
##
(Only include cities of LA) Group by neighborhood and average on the other attributes

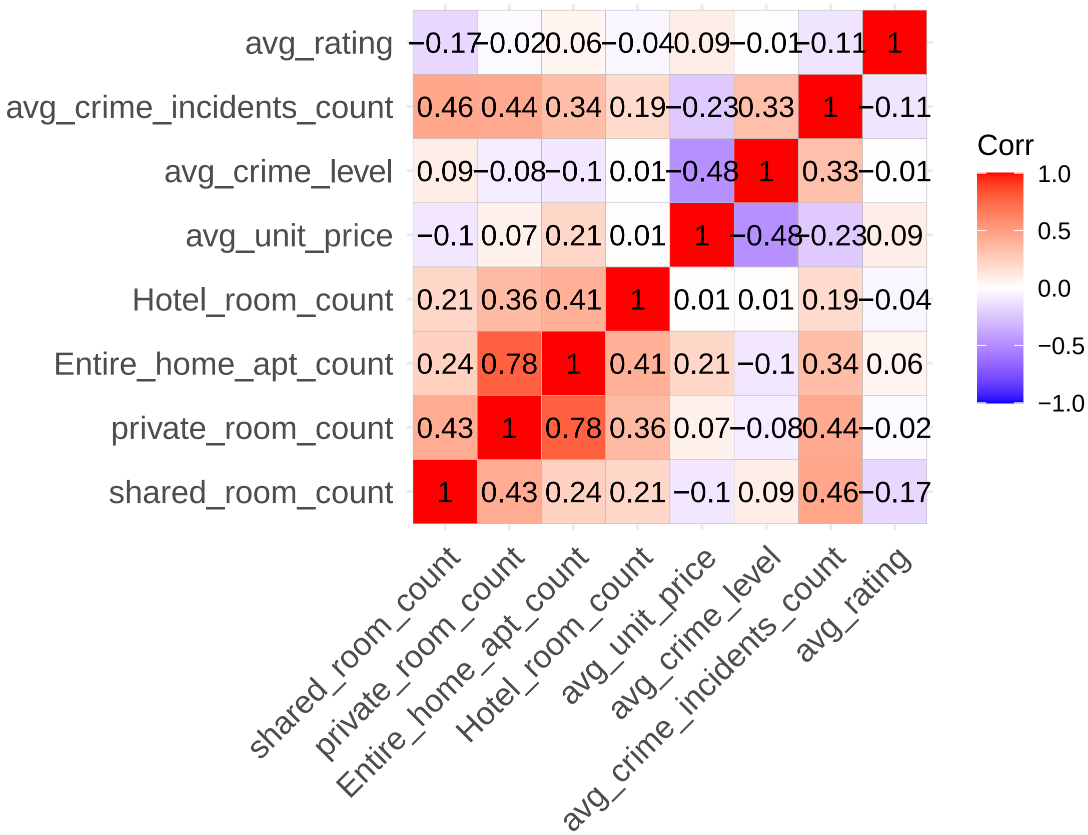
```
final_data_OHE = final_data %>%
  mutate(n = 1) %>%
  pivot_wider(names_from = room_type, values_from=n, values_fill=0)

final_data_OHE_LA = final_data_OHE %>%
  filter(neighbourhood_group=='City of Los Angeles')

# final_data_OHE_LA %>% filter(neighbourhood=='Hollywood Hills')
grouped_neighbourhood_LA = final_data_OHE_LA %>%
                                group_by(neighbourhood) %>%
                                summarise(shared_room_count = sum(`Shared room`),
                                          private_room_count = sum(`Private room`),
                                          Entire_home_apt_count = sum(`Entire home/apt`),
                                          Hotel_room_count = sum(`Hotel room`),
                                          avg_unit_price = mean(unit_price),
                                          avg_crime_level = mean(crime_level),
                                          avg_crime_incidents_count = mean(crime_incidents_count),
                                          avg_rating = mean(avg_rating),
                                          latitude = mean(latitude),
                                          longitude = mean(longitude),
                                          housing_count = n())
train_data = select(grouped_neighbourhood_LA, -c(neighbourhood, housing_count, latitude, longitude))
```

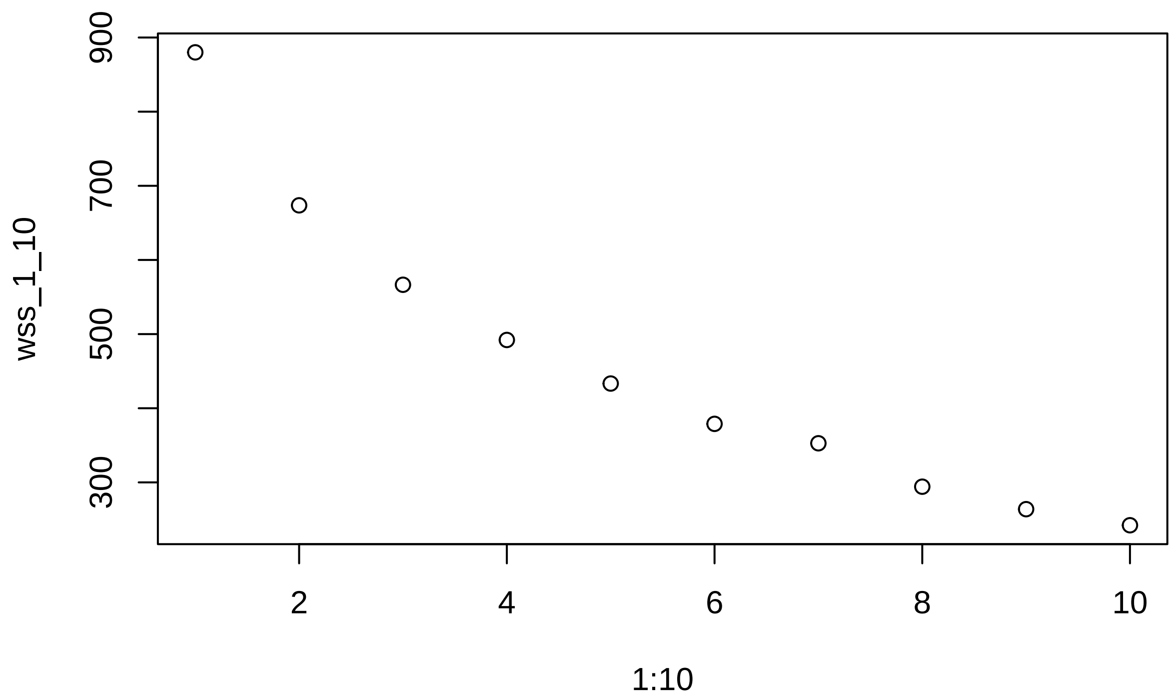
o

```
fullCorrMatrix = round(cor(train_data %>% select_if(is.numeric)), 2)
ggcorrplot(fullCorrMatrix, lab = TRUE)
```



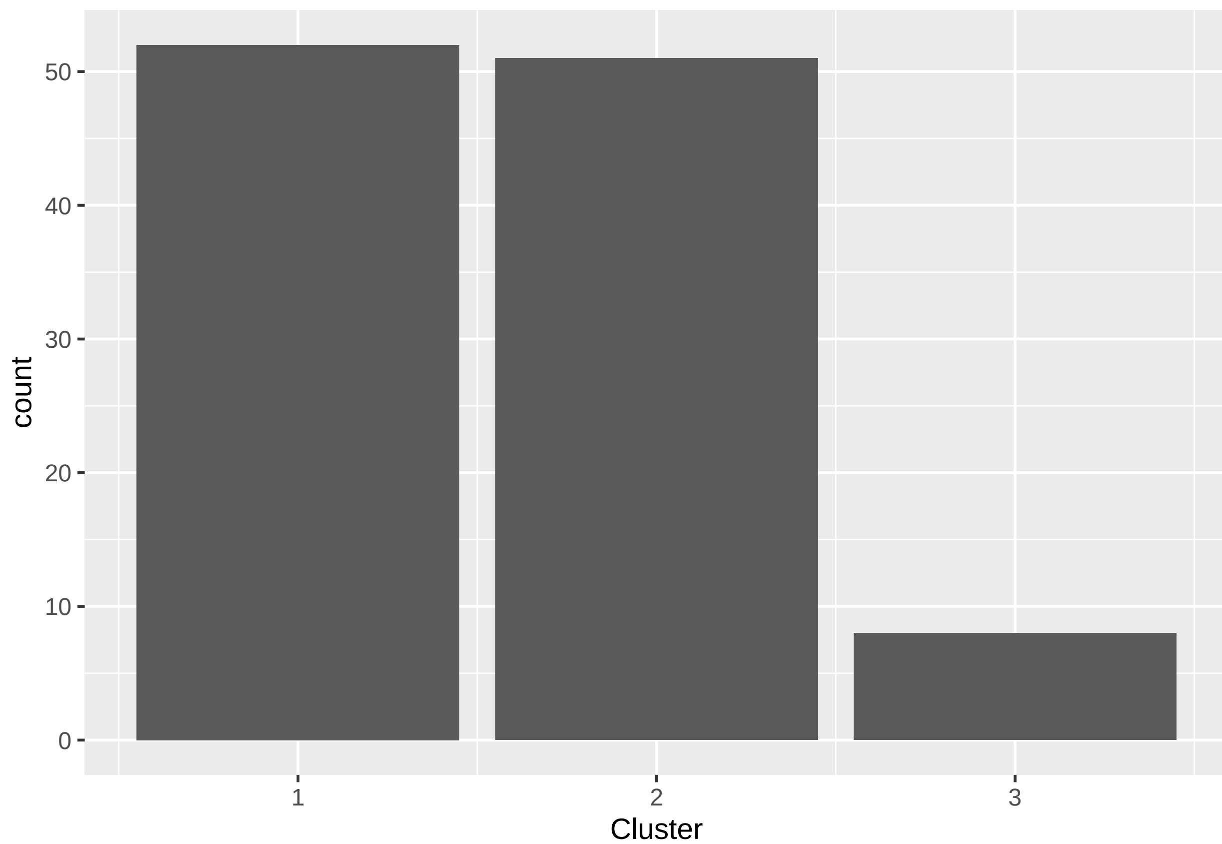
o

```
wss = function (k) {kmeans(scale(train_data), k, nstart=10)$tot.withinss}
wss_1_10 = map_dbl(1:10, wss) # with k from 1 to 10, means call wss function 10 times to do kmeans
plot(1:10, wss_1_10)
```



o

```
res = kmeans(scale(train_data), 3, nstart=25)
grouped_neighbourhood_data_res = grouped_neighbourhood_LA %>% mutate(Cluster = res$cluster)
ggplot(grouped_neighbourhood_data_res) + geom_bar(aes(x=Cluster))
```



o

```
########## cluster1 #############
grouped_neighbourhood_data_res %>%
  filter(Cluster==1) %>%
  summarize_numeric()
```

```
##                   Attribute Missing Values Unique Values        Mean
## 1          shared_room_count              0             9    1.711538
## 2         private_room_count              0            19    7.596154
## 3      Entire_home_apt_count              0            26   14.192308
## 4           Hotel_room_count              0             1    0.000000
## 5             avg_unit_price              0            52   31.585183
## 6            avg_crime_level              0            52    2.374208
## 7  avg_crime_incidents_count              0            51  239.599071
## 8                 avg_rating              0            51    3.501377
## 9                   latitude              0            50   34.060091
## 10                 longitude              0            49 -118.313897
## 11             housing_count              0            34   23.500000
## 12                   Cluster              0             1    1.000000
##             Min         Max          SD
## 1      0.000000   30.000000  5.35535846
## 2      0.000000   29.000000  7.32511023
## 3      0.000000   51.000000 14.61001330
## 4      0.000000    0.000000  0.00000000
## 5     10.250000   51.114583  7.00302540
```

```
## 6     2.187431    2.550312   0.08588084
## 7    57.500000  547.000000 146.85236370
## 8     2.820500    4.000000   0.18415947
## 9    33.728605   34.305000   0.12152020
## 10 -118.596842 -118.178750   0.09067161
## 11    1.000000   81.000000  19.85881540
## 12    1.000000    1.000000   0.00000000
```

```
########## cluster2 #############
grouped_neighbourhood_data_res %>%
  filter(Cluster==2) %>%
  summarize_numeric()
```

```
##                     Attribute Missing Values Unique Values         Mean
## 1           shared_room_count              0             8    1.1960784
## 2          private_room_count              0            27   14.3725490
## 3       Entire_home_apt_count              0            39   57.2941176
## 4            Hotel_room_count              0             4    0.1568627
## 5              avg_unit_price              0            50   58.7163092
## 6             avg_crime_level              0            51    2.1230346
## 7   avg_crime_incidents_count              0            51  137.4961355
## 8                  avg_rating              0            51    3.5275187
## 9                    latitude              0            51   34.1102786
## 10                  longitude              0            51 -118.4118037
## 11              housing_count              0            44   73.0196078
## 12                    Cluster              0             1    2.0000000
##            Min         Max          SD
## 1     0.000000   13.000000  2.67596418
## 2     1.000000   54.000000 12.83115082
## 3     2.000000  304.000000 59.50640104
## 4     0.000000    3.000000  0.57870715
## 5    17.854167  218.166667 36.48519115
## 6     1.813052    2.289968  0.10188168
## 7    11.488889  292.686047 75.47158862
## 8     2.996143    3.745143  0.12613823
## 9    33.958529   34.278095  0.08420819
## 10 -118.634062 -118.207759  0.09284585
## 11    4.000000  344.000000 70.16822363
## 12    2.000000    2.000000  0.00000000
```

```
########## cluster3 #############
grouped_neighbourhood_data_res %>%
  filter(Cluster==3) %>%
  summarize_numeric()
```

```
##                     Attribute Missing Values Unique Values        Mean
## 1           shared_room_count              0             8    8.875000
## 2          private_room_count              0             8   51.875000
## 3       Entire_home_apt_count              0             8  255.500000
## 4            Hotel_room_count              0             6    6.125000
## 5              avg_unit_price              0             8   42.115301
## 6             avg_crime_level              0             8    2.285622
## 7   avg_crime_incidents_count              0             8  588.211118
```

```
## 8                avg_rating             0            8    3.471936
## 9                  latitude             0            8   34.022122
## 10                longitude             0            8 -118.328387
## 11            housing_count             0            8  322.375000
## 12                  Cluster             0            1    3.000000
##             Min         Max           SD
## 1      0.000000   17.000000   5.96268156
## 2      5.000000  100.000000  32.49368070
## 3      5.000000  701.000000 238.67072356
## 4      0.000000   15.000000   4.85320218
## 5     31.714286   63.780013  11.25754470
## 6      2.229041    2.352458   0.04080038
## 7    103.360000 1118.888361 377.55242022
## 8      3.394380    3.570078   0.05457851
## 9     33.774400   34.096989   0.10516982
## 10 -118.464789 -118.254038   0.08136706
## 11    25.000000  783.000000 259.17447929
## 12     3.000000    3.000000   0.00000000
```

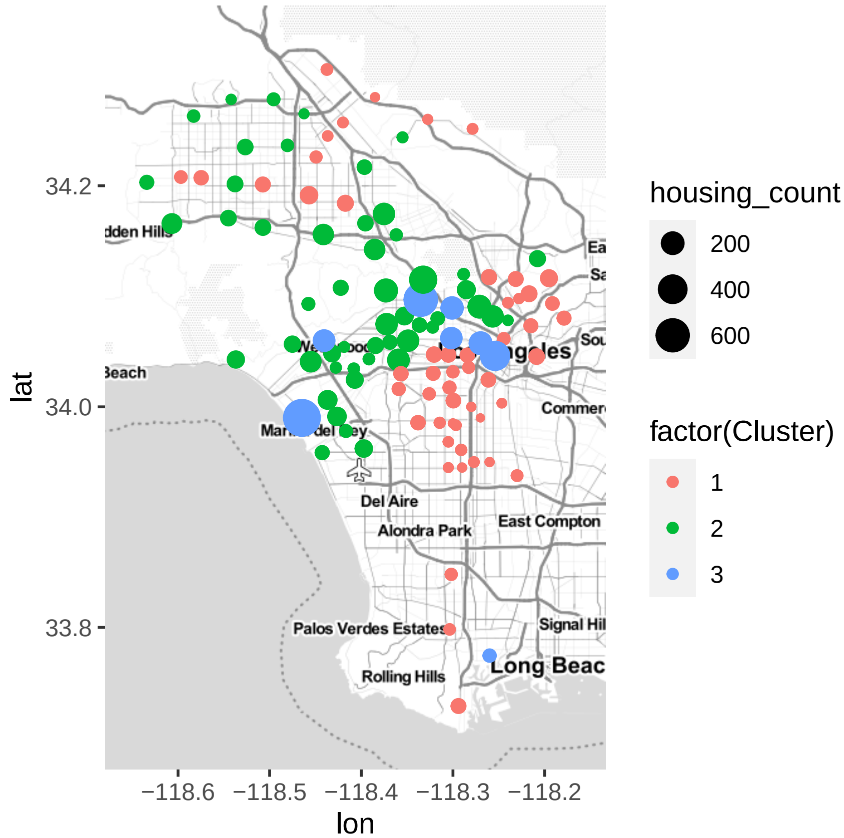
o

```
height <- max(grouped_neighbourhood_LA$latitude) - min(grouped_neighbourhood_LA$latitude)
width <- max(grouped_neighbourhood_LA$longitude) - min(grouped_neighbourhood_LA$longitude)
LA_borders <- c(bottom  = min(grouped_neighbourhood_LA$latitude)  - 0.1 * height,
                 top     = max(grouped_neighbourhood_LA$latitude)  + 0.1 * height,
                 left    = min(grouped_neighbourhood_LA$longitude) - 0.1 * width,
                 right   = max(grouped_neighbourhood_LA$longitude) + 0.1 * width)

map <- get_stamenmap(LA_borders, zoom = 10, maptype = "toner-lite")
```

```
## i Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.
```

```
ggmap(map) +
    geom_point(data = grouped_neighbourhood_data_res, mapping = aes(x = longitude, y = latitude, color=
```


housing_count
200
400
600
factor(Cluster)
1
2
3
lat
lon
34.2
34.0
33.8
−118.6
−118.5
−118.4
−118.3
−118.2
Del Aire
Alondra Park
East Compton
Palos Verdes Estates
Signal Hil
Rolling Hills
Long Beac
Commer

o