Clustering Analysis

Nadia Ahmad

2023-04-02

Cluster Analysis on Travel Discrimination

Dataset explanation:
Variables:
Continuos
- Q1_ = Travel frequency
- Q6_15 : checkin experience rate
- Q6_18 : fly experience rate
Categorical
- Q15 = Gender
- Q17 = Race
- Q18 = Religion

Library

```
library(readr)
library(readxl)
library(tidyverse)
library(corrplot)
library(MASS)
library(ggfortify)
library(ggpubr)
library(pvclust)
library(cluster)
library(fpc)
library(factoextra)
library(gridExtra)
```

Read the dataset

```
travel <- read_excel("data_clustering.xlsx")
head(travel)
```

```
## # A tibble: 6 x 14
##   Respon~1 UserL~2 Text ~3 Q1    Q1_   Q6_15 Q6_16 Q6_17 Q6_18 Q6_19 Q14   Q15
##   <chr>    <chr>   <chr>   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 "Respon~ "User ~ "Text ~ "How~ "How~ "How~ "How~ "How~ "How~ "How~ "How~ "To ~
## 2 "{\"Imp~ "{\"Im~ "{\"Im~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~ "{\"~
## 3 "1"      "EN"    "Yes"   "<3 ~ "3"   "54"  "50"  "71"  "50"  "51"  "18-~ "Fem~
## 4 "2"      "EN"    "Yes"   "<3 ~ "3"   "52"  "52"  "51"  "53"  "52"  "35-~ "Mal~
```

```
## 5 "3"      "EN"    "Yes"   "4-6~ "5"    <NA>  <NA>  <NA>  <NA>  <NA> "65+~ "Mal~
## 6 "4"      "EN"    "Yes"   "<3 ~ "3"   "51"  "53"  "54"  "52"  "57"  "25-~ "Fem~
## # ... with 2 more variables: Q17 <chr>, Q18 <chr>, and abbreviated variable
## #   names 1: ResponseId, 2: UserLanguage, 3: `Text / Graphic`
```

Dataset contains 231 rows and 14 columns which is still messy. Thus, we'll conduct some data preprocessing steps.

DATA PREPROCESSING

```
# First, drop two first rows. Next, filter only data that has 100 in progress
travel <- travel %>%
  slice(-c(1,2))

# Select used columns
travel_df <- travel[c(1,5,6,9,12,13,14)]
```

```
# CHECK MISSING VALUE----
# Count the missing values by column wise
print("Count of missing values by column wise")
```

```
## [1] "Count of missing values by column wise"
```

```
sapply(travel_df, function(x) sum(is.na(x)))
```

```
## ResponseId        Q1_      Q6_15      Q6_18        Q15        Q17        Q18
##          0         30         72         75         74         72         78
```

```
# Missing value imputation
# Since our data contains 46 missing value, let's impute with mode
# Function to see mode
calc_mode <- function(x){

  # List the distinct / unique values
  distinct_values <- unique(na.omit(x))

  # Count the occurrence of each distinct value
  distinct_tabulate <- tabulate(match(x, distinct_values))

  # Return the value with the highest occurrence
  distinct_values[which.max(distinct_tabulate)]
}

# Impute missing value----
travel_df <- travel_df %>%
  mutate(across(everything(), ~replace_na(.x, calc_mode(.x))))
```

```
# Rename column name
travel_df_clean <- travel_df %>%
       rename(respondent_id =1, travel_frequency = 2, checkin_exp = 3,
```

```
              fly_exp = 4, gender = 5, race=6,
              religion = 7)
head(travel_df_clean)
```

```
## # A tibble: 6 x 7
##   respondent_id travel_frequency checkin_exp fly_exp gender race         relig~1
##   <chr>         <chr>            <chr>       <chr>   <chr>  <chr>        <chr>
## 1 1             3                54          50      Female Asian        Islam
## 2 2             3                52          53      Male   Black of Af~ Islam
## 3 3             5                50          50      Male   Asian        Islam
## 4 4             3                51          52      Female Asian        Islam
## 5 5             3                48          100     Male   Asian        Islam
## 6 6             3                50          50      Male   White        Atheis~
## # ... with abbreviated variable name 1: religion
```

```
# CONVERT DATA TYPE----
# Convert all variables into integer
# Convert column 2 to 6 to numeric
travel_df_clean[,2:4] <- lapply(travel_df_clean[,2:4], as.numeric)
travel_df_clean[,5:7] <- lapply(travel_df_clean[,5:7], as.factor)
head(travel_df_clean)
```

```
## # A tibble: 6 x 7
##   respondent_id travel_frequency checkin_exp fly_exp gender race         relig~1
##   <chr>                    <dbl>       <dbl>   <dbl> <fct>  <fct>        <fct>
## 1 1                            3          54      50 Female Asian        Islam
## 2 2                            3          52      53 Male   Black of Af~ Islam
## 3 3                            5          50      50 Male   Asian        Islam
## 4 4                            3          51      52 Female Asian        Islam
## 5 5                            3          48     100 Male   Asian        Islam
## 6 6                            3          50      50 Male   White        Atheis~
## # ... with abbreviated variable name 1: religion
```

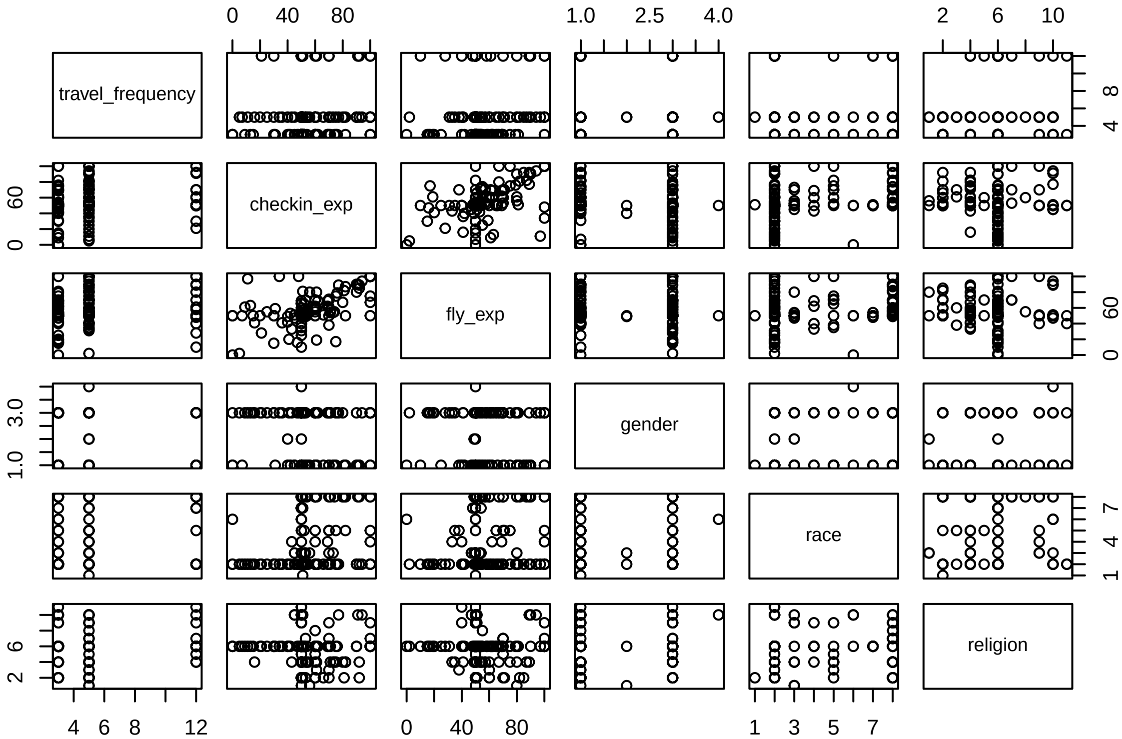
Reproducibility

Define seed number thus everytime we run the script, it yields the same result.

```
set.seed(123)
```

1. Summary Statistics

```
pairs(travel_df_clean[,-1])
```




```r
summary(travel_df_clean)
```

```
##  respondent_id      travel_frequency  checkin_exp       fly_exp
##  Length:229         Min.   : 3.000   Min.   :  0.00   Min.   :  0.00
##  Class :character   1st Qu.: 3.000   1st Qu.: 50.00   1st Qu.: 50.00
##  Mode  :character   Median : 5.000   Median : 50.00   Median : 50.00
##                     Mean   : 4.895   Mean   : 52.99   Mean   : 54.06
##                     3rd Qu.: 5.000   3rd Qu.: 52.00   3rd Qu.: 54.00
##                     Max.   :12.000   Max.   :100.00   Max.   :100.00
##
##                               gender                           race
##  Female                          : 72   Asian                     :162
##  Gender Variant/Non-Conforming:  2   White                     : 35
##  Male                            :154   Black of African American:  9
##  Prefer Not to Answer            :  1   Multiracial               :  8
##                                         Some Other Race           :  6
##                                         Hispanic or Latino        :  5
##                                         (Other)                   :  4
##                     religion
##  Islam                  :156
##  Christianity           : 28
##  Atheism/Agnosticism    : 17
##  Prefer Not to Disclose:  9
##  Other                  :  7
##  Judaism                :  3
##  (Other)                :  9
```

Standard Deviation of travel_frequency, checkin_experience, and fly_experience

```r
round(sqrt(apply(travel_df_clean[,2:4],2,var)),2)
```

```
## travel_frequency      checkin_exp          fly_exp
##             2.39            17.16            15.78
```

Since our clustering method is distanced-based, we'll scale the numerical features. ### Scale Data

```
travel_df_clean[,2:4] <- scale(travel_df_clean[,2:4], center = T, scale = T)
head(travel_df_clean)
```

```
## # A tibble: 6 x 7
##   respondent_id travel_frequency checkin_exp fly_exp gender race         relig~1
##   <chr>                    <dbl>       <dbl>   <dbl> <fct>  <fct>        <fct>
## 1 1                       -0.792      0.0590 -0.257  Female Asian        Islam
## 2 2                       -0.792     -0.0575 -0.0670 Male   Black of Af~ Islam
## 3 3                        0.0438    -0.174  -0.257  Male   Asian        Islam
## 4 4                       -0.792     -0.116  -0.130  Female Asian        Islam
## 5 5                       -0.792     -0.291   2.91   Male   Asian        Islam
## 6 6                       -0.792     -0.174  -0.257  Male   White        Atheis~
## # ... with abbreviated variable name 1: religion
```

Before that, we notice that our dataframe contains variables which is mixed data type, we'll try use Gower distance for Hierarchical Clustering.

Gower Distance

```
gower_dist <- daisy(travel_df_clean[,-1], metric = c("gower"))
```

But, since we also want to conduct k-means clustering, we will calculate distance between numerical variables only. ### Euclidian Distance

```
euc_dist <- dist(travel_df_clean[,2:4], method = "euclidian")
```

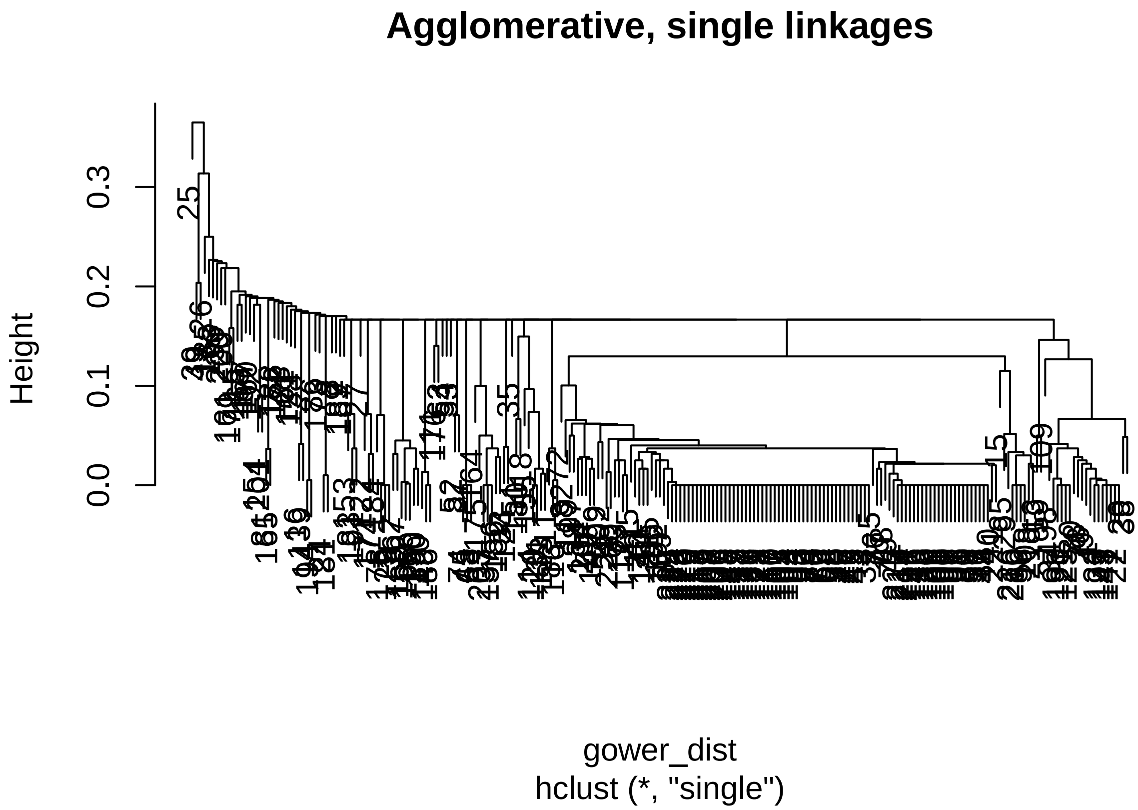
Manhattan Distance

```
manhat_dist <- dist(travel_df_clean[,2:4], method = "manhattan")
```

2. HIREARCHICAL CLUSTERING

Distance : Gower Distance for Mixed Data

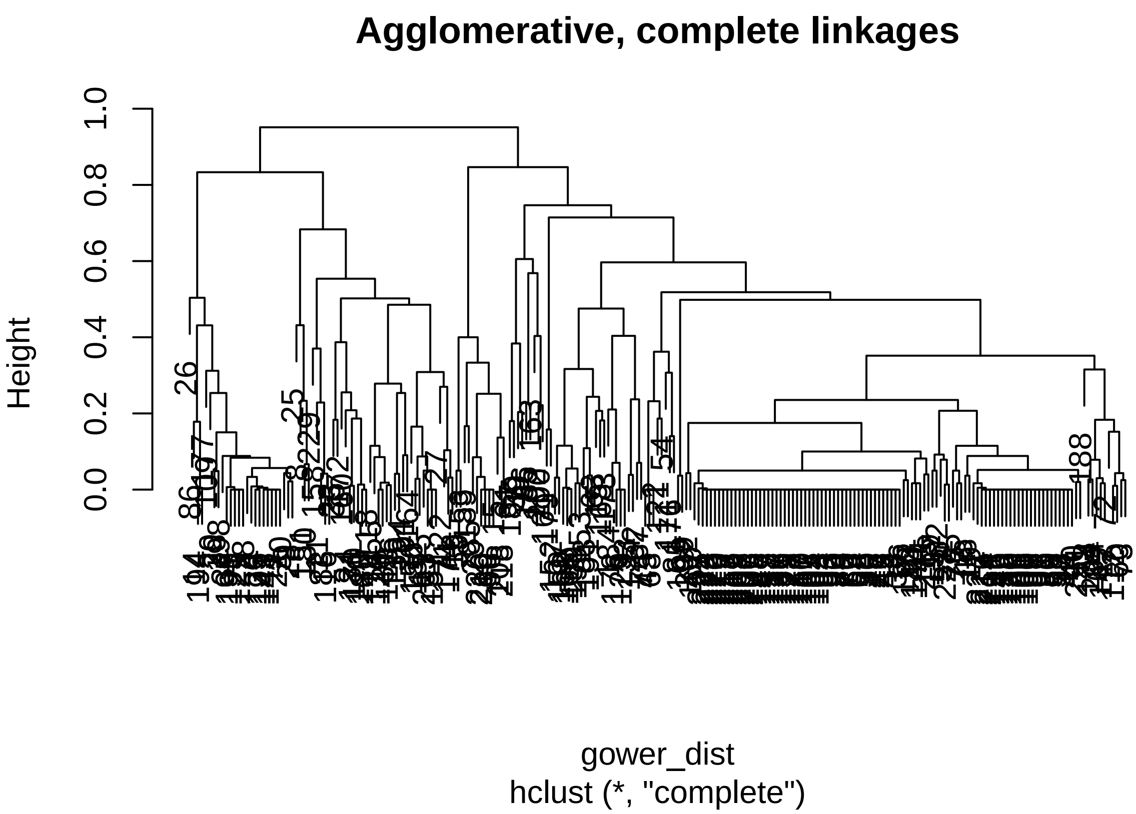
```
aggl_clust_s <- hclust(gower_dist, method = "single")
plot(aggl_clust_s,
     main = "Agglomerative, single linkages")
```



Single

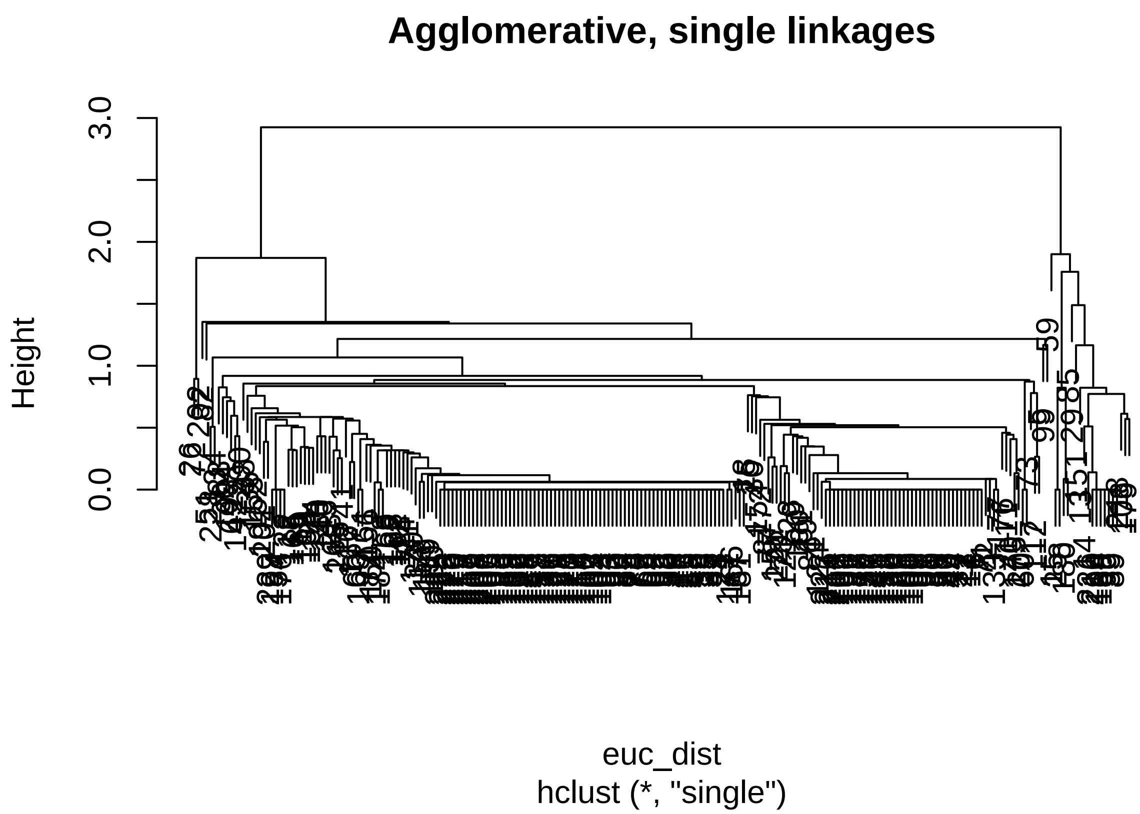
Complete

```
aggl_clust_c <- hclust(gower_dist, method = "complete")
plot(aggl_clust_c,
     main = "Agglomerative, complete linkages")
```

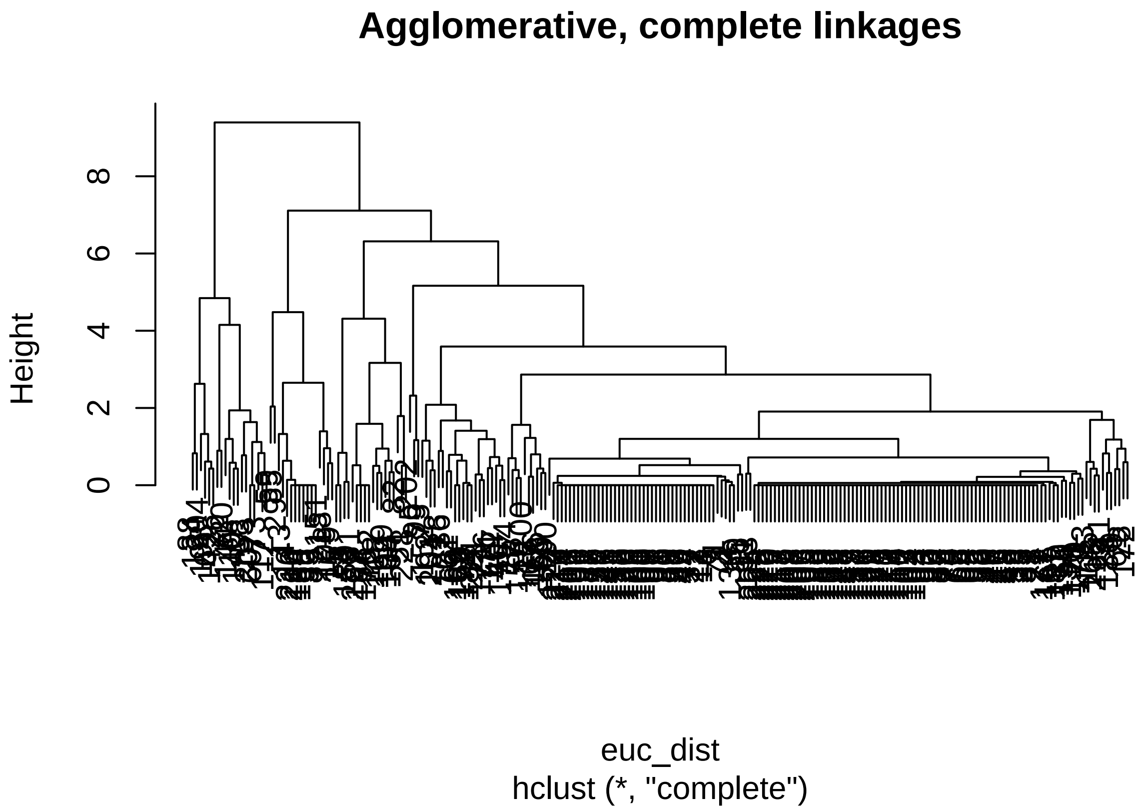


Distance : Euclidian Distance for Numerical Data Only #### Single

```
aggl_clust_s_e <- hclust(euc_dist, method = "single")
plot(aggl_clust_s_e,
     main = "Agglomerative, single linkages")
```



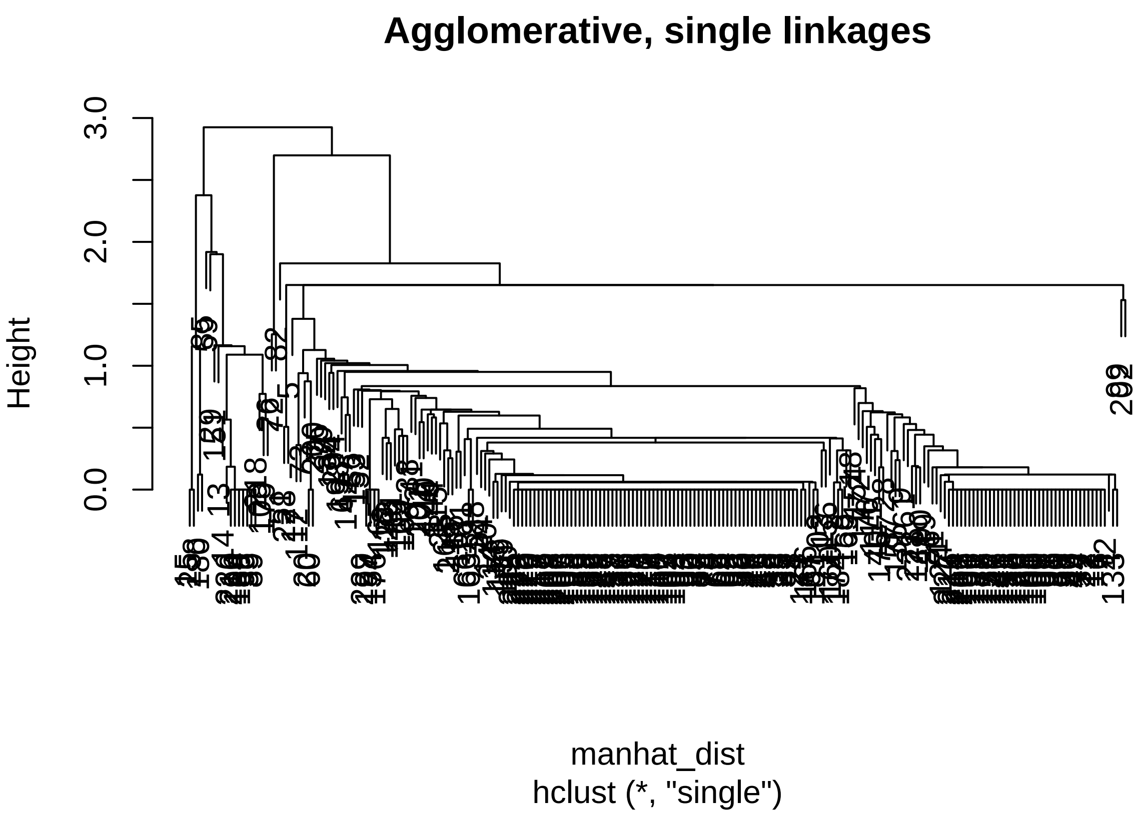

```
aggl_clust_c_e <- hclust(euc_dist, method = "complete")
plot(aggl_clust_c_e,
     main = "Agglomerative, complete linkages")
```



Complete

Distance : Manhattan Distance for Numerical Data Only #### Single

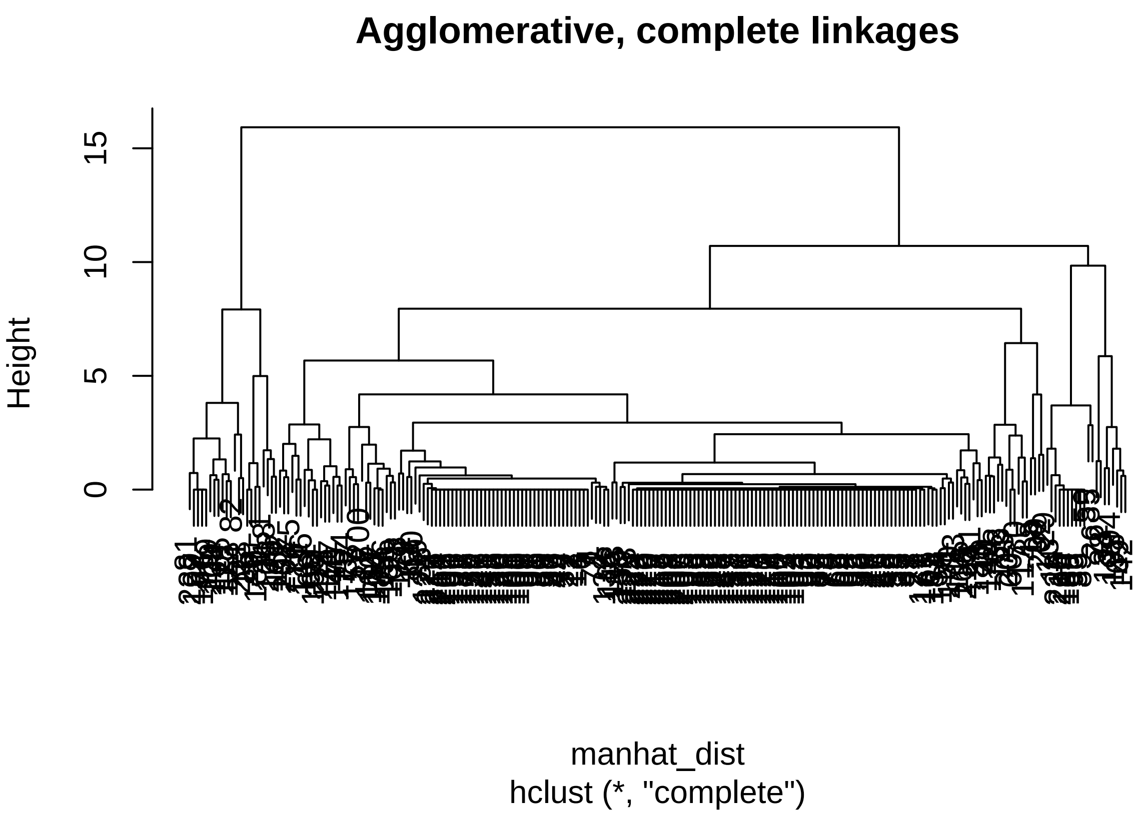
```
aggl_clust_s_m <- hclust(manhat_dist, method = "single")
plot(aggl_clust_s_m,
     main = "Agglomerative, single linkages")
```



####

Complete

```
aggl_clust_c_m <- hclust(manhat_dist, method = "complete")
plot(aggl_clust_c_m,
     main = "Agglomerative, complete linkages")
```



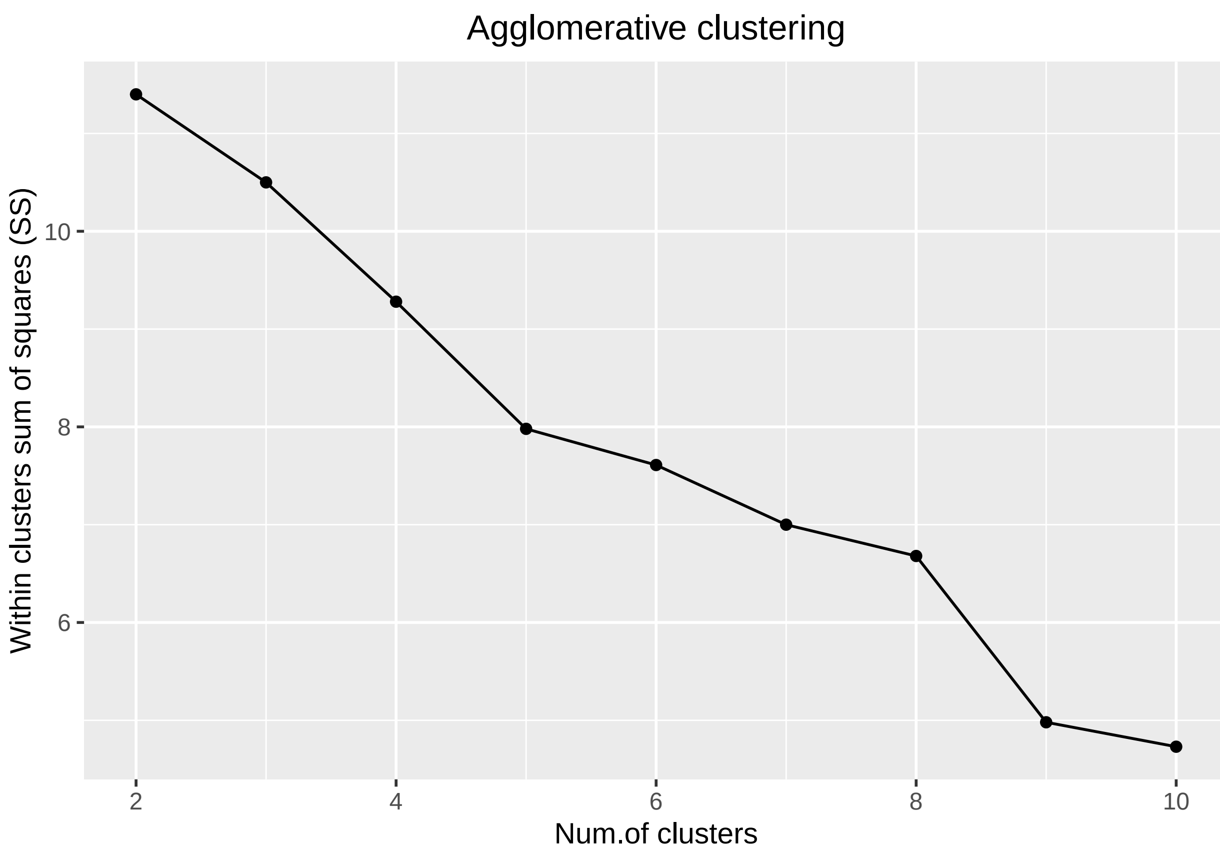
3. NUMBER OF CLUSTER TO RETAIN

```
cstats.table <- function(dist, tree, k) {
clust.assess <- c("cluster.number","n","within.cluster.ss","average.within","average.between",
                  "wb.ratio","dunn2","avg.silwidth")
clust.size <- c("cluster.size")
stats.names <- c()
row.clust <- c()
output.stats <- matrix(ncol = k, nrow = length(clust.assess))
cluster.sizes <- matrix(ncol = k, nrow = k)
for(i in c(1:k)){
  row.clust[i] <- paste("Cluster-", i, " size")
}
for(i in c(2:k)){
  stats.names[i] <- paste("Test", i-1)

  for(j in seq_along(clust.assess)){
    output.stats[j, i] <- unlist(cluster.stats(d = dist, clustering = cutree(tree, k = i))[clust.assess

  }

  for(d in 1:k) {
    cluster.sizes[d, i] <- unlist(cluster.stats(d = dist, clustering = cutree(tree, k = i))[clust.size]
    dim(cluster.sizes[d, i]) <- c(length(cluster.sizes[i]), 1)
    cluster.sizes[d, i]

  }
}
output.stats.df <- data.frame(output.stats)
```

```
cluster.sizes <- data.frame(cluster.sizes)
cluster.sizes[is.na(cluster.sizes)] <- 0
rows.all <- c(clust.assess, row.clust)
# rownames(output.stats.df) <- clust.assess
output <- rbind(output.stats.df, cluster.sizes)[ ,-1]
colnames(output) <- stats.names[2:k]
rownames(output) <- rows.all
is.num <- sapply(output, is.numeric)
output[is.num] <- lapply(output[is.num], round, 2)
output
}
```

A. Gower Distance Complete Linkage

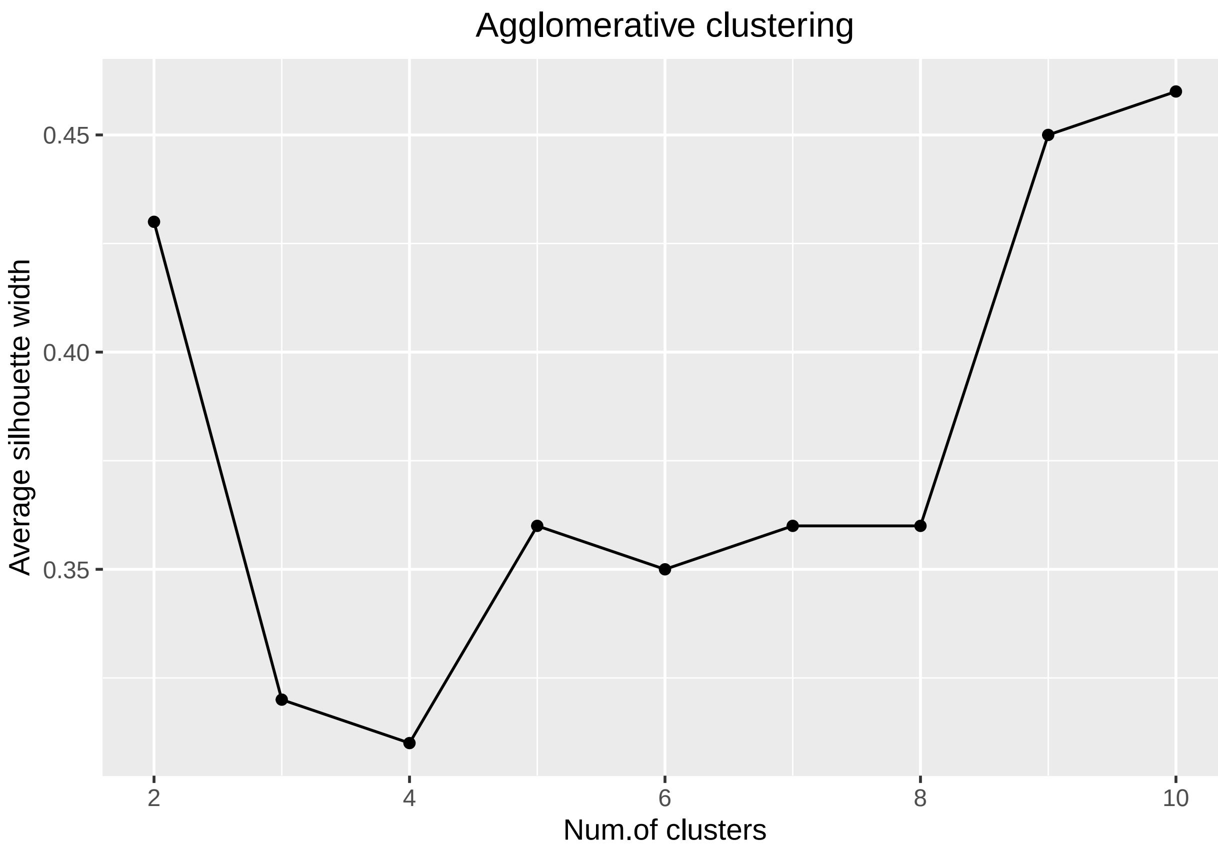
```
ggplot(data = data.frame(t(cstats.table(gower_dist, aggl_clust_c, 10))),
  aes(x=cluster.number, y=within.cluster.ss)) +
  geom_point()+
  geom_line()+
  ggtitle("Agglomerative clustering") +
  labs(x = "Num.of clusters", y = "Within clusters sum of squares (SS)") +
  theme(plot.title = element_text(hjust = 0.5))
```



Elbow Method

```
ggplot(data = data.frame(t(cstats.table(gower_dist, aggl_clust_c, 10))),
  aes(x=cluster.number, y=avg.silwidth)) +
```

```
  geom_point()+
  geom_line()+
  ggtitle("Agglomerative clustering") +
  labs(x = "Num.of clusters", y = "Average silhouette width") +
  theme(plot.title = element_text(hjust = 0.5))
```



Silhouette
Print Member with k = 6

```
member1 = cutree(aggl_clust_c,6)
table(member1)
```

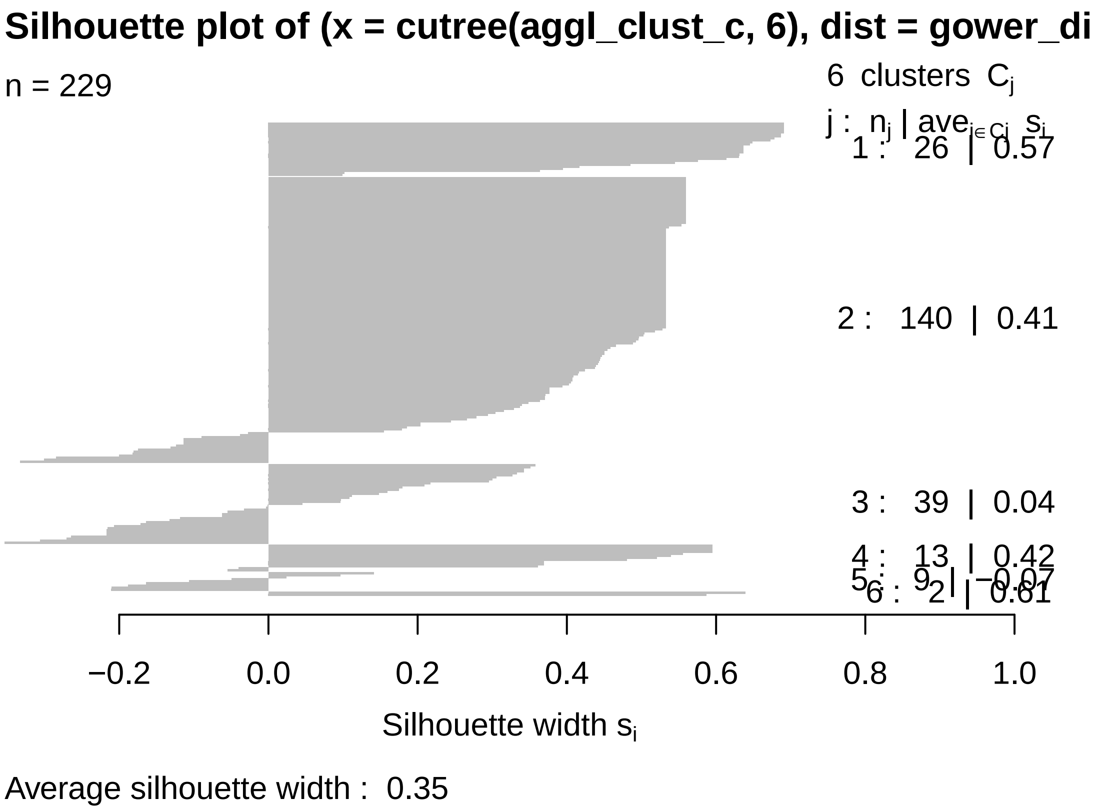
```
## member1
##   1   2   3   4   5   6
##  26 140  39  13   9   2
```

```
aggregate(travel_df_clean[,2:4],list(member1),mean)
```

Characteristic Cluster

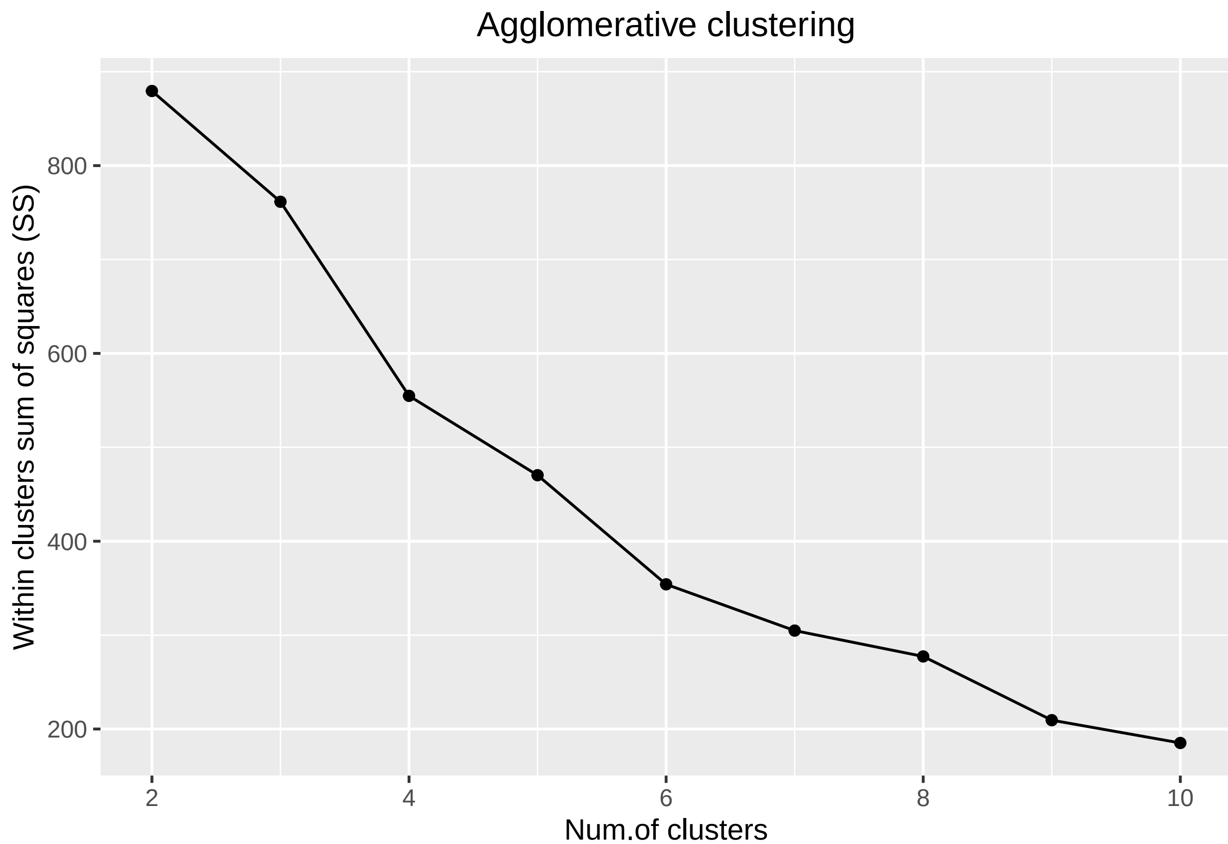
```
##   Group.1 travel_frequency checkin_exp     fly_exp
## 1       1       -0.2776700 -0.44972481 -0.23996996
## 2       2       -0.2785885 -0.07748772 -0.06152384
## 3       3        0.1723856  0.52820065  0.38792181
## 4       4        2.9691600 -0.03509637 -0.27164884
## 5       5       -0.2348076  0.15615567  0.13719832
## 6       6        1.5064791  0.49607629  1.01012763
```

```
plot(silhouette(cutree(aggl_clust_c,6), gower_dist))
```

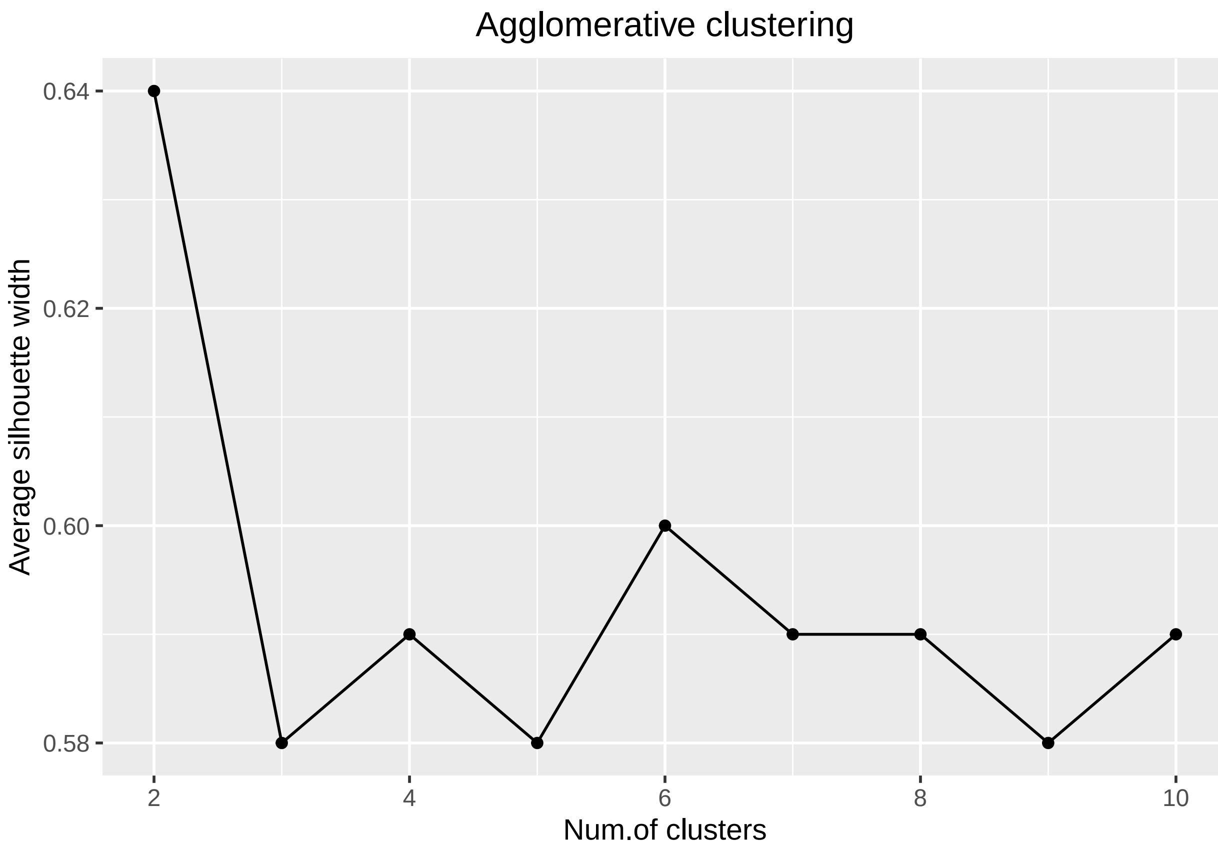


B. Manhattan Distance Complete Linkage #### Elbow Method

```
ggplot(data = data.frame(t(cstats.table(manhat_dist, aggl_clust_c_m, 10))),
  aes(x=cluster.number, y=within.cluster.ss)) +
  geom_point()+
  geom_line()+
  ggtitle("Agglomerative clustering") +
  labs(x = "Num.of clusters", y = "Within clusters sum of squares (SS)") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(data = data.frame(t(cstats.table(manhat_dist, aggl_clust_c_m, 10))),
  aes(x=cluster.number, y=avg.silwidth)) +
  geom_point()+
  geom_line()+
  ggtitle("Agglomerative clustering") +
  labs(x = "Num.of clusters", y = "Average silhouette width") +
  theme(plot.title = element_text(hjust = 0.5))
```



Silhouette

Print Member with k = 5

```
member2 = cutree(aggl_clust_c_m,5)
table(member2)
```

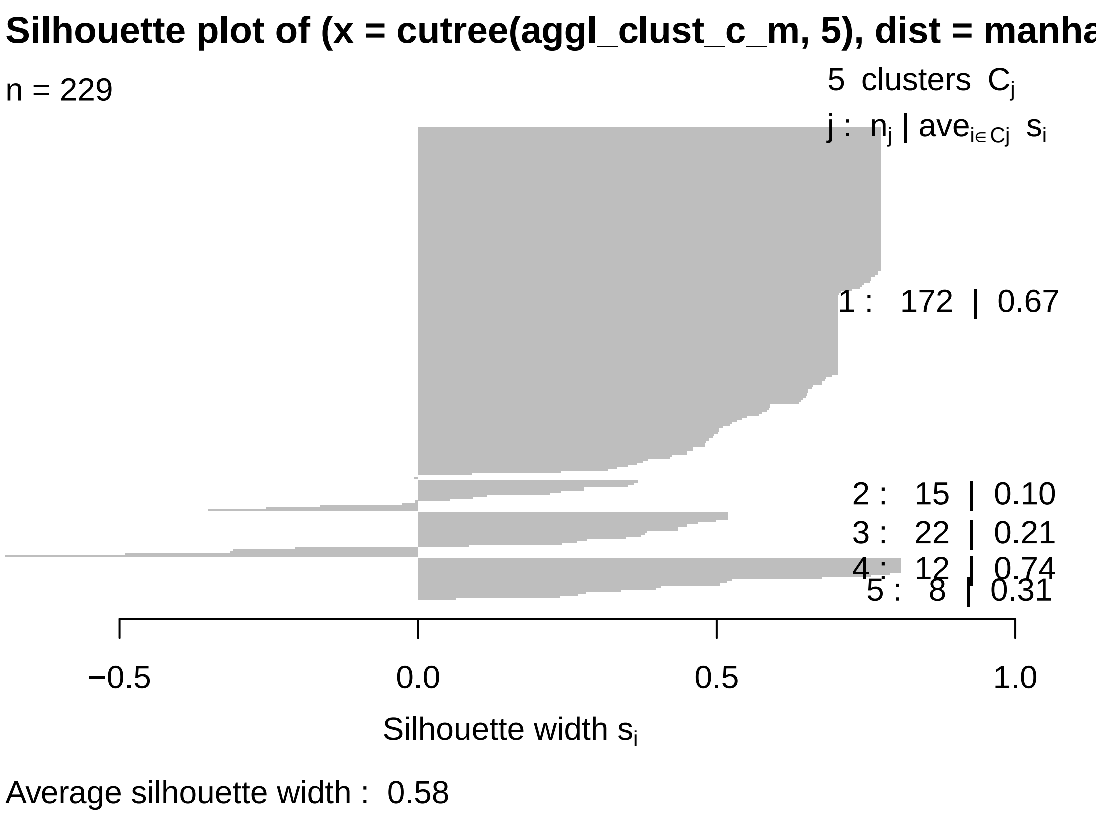
```
## member2
##   1   2   3   4   5
## 172  15  22  12   8
```

```
aggregate(travel_df_clean[,2:4],list(member2),mean)
```

Characteristic Cluster

```
##   Group.1 travel_frequency checkin_exp    fly_exp
## 1       1       -0.2720630 -0.03413212 -0.1336275
## 2       2       -0.4019711 -1.88336803  0.6173094
## 3       3        1.0315828  2.07736934  1.8942565
## 4       4        2.9691600 -0.40714135 -0.6477341
## 5       5       -0.6875421 -0.83689813 -2.5220682
```

```
plot(silhouette(cutree(aggl_clust_c_m,5), manhat_dist))
```



4. k-MEANS CLUSTERING

```r
k2 <- kmeans(travel_df_clean[,2:4], centers = 2)
str(k2)
```

```
## List of 9
##  $ cluster     : int [1:229] 1 1 2 1 2 1 1 2 1 1 ...
##  $ centers     : num [1:2, 1:3] -0.653 0.337 -0.557 0.287 -0.52 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:3] "travel_frequency" "checkin_exp" "fly_exp"
##  $ totss       : num 684
##  $ withinss    : num [1:2] 113 452
##  $ tot.withinss: num 565
##  $ betweenss   : num 119
##  $ size        : int [1:2] 78 151
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

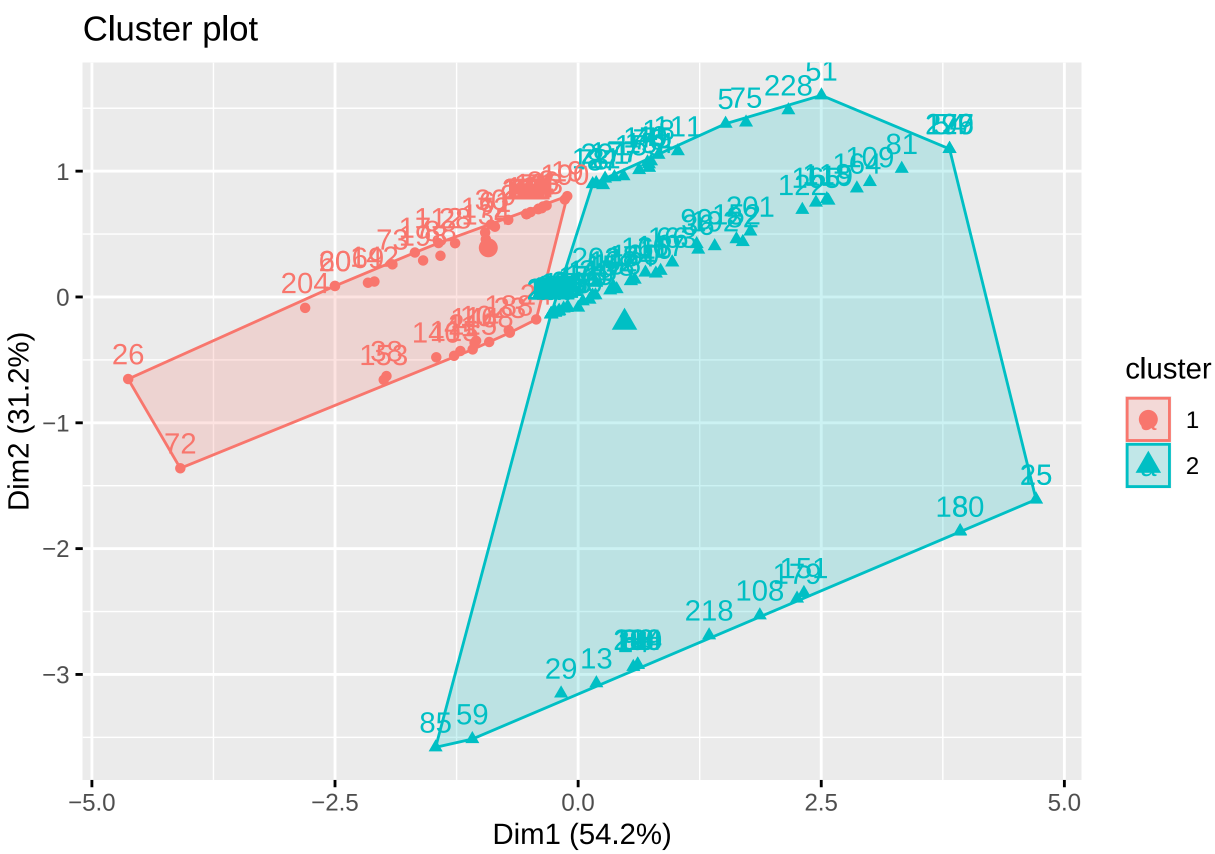
```r
k2
```

```
## K-means clustering with 2 clusters of sizes 78, 151
##
## Cluster means:
##   travel_frequency checkin_exp    fly_exp
## 1       -0.6527164   -0.556557 -0.5202063
## 2        0.3371648    0.287493  0.2687158
##
## Clustering vector:
```

```
##   [1] 1 1 2 1 2 1 1 2 1 1 1 1 2 2 2 2 1 2 1 1 2 1 1 1 2 1 2 1 2 1 2 2 2 2 2 2 1
##  [38] 1 2 2 1 2 2 1 2 2 2 2 1 2 2 2 2 2 2 1 1 1 2 1 1 2 2 2 2 2 2 2 1 2 1 1 1 2
##  [75] 2 2 2 2 2 1 2 2 2 2 2 1 2 1 2 2 1 2 2 2 1 2 1 1 2 1 1 2 2 2 2 1 2 2 2 2 2
## [112] 1 2 2 1 2 2 2 2 2 2 2 1 1 2 1 2 2 2 1 2 1 1 1 1 2 2 2 2 1 1 1 1 1 2 1 2 2
## [149] 2 2 2 2 1 2 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2
## [186] 2 1 1 2 1 1 2 2 2 2 1 2 1 1 2 2 2 1 1 2 2 2 2 2 2 1 2 2 1 1 1 2 2 2 2 2 2
## [223] 2 2 2 2 1 2 2
##
## Within cluster sum of squares by cluster:
## [1] 112.5988 452.3516
##  (between_SS / total_SS =  17.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

CLUSTER PLOT

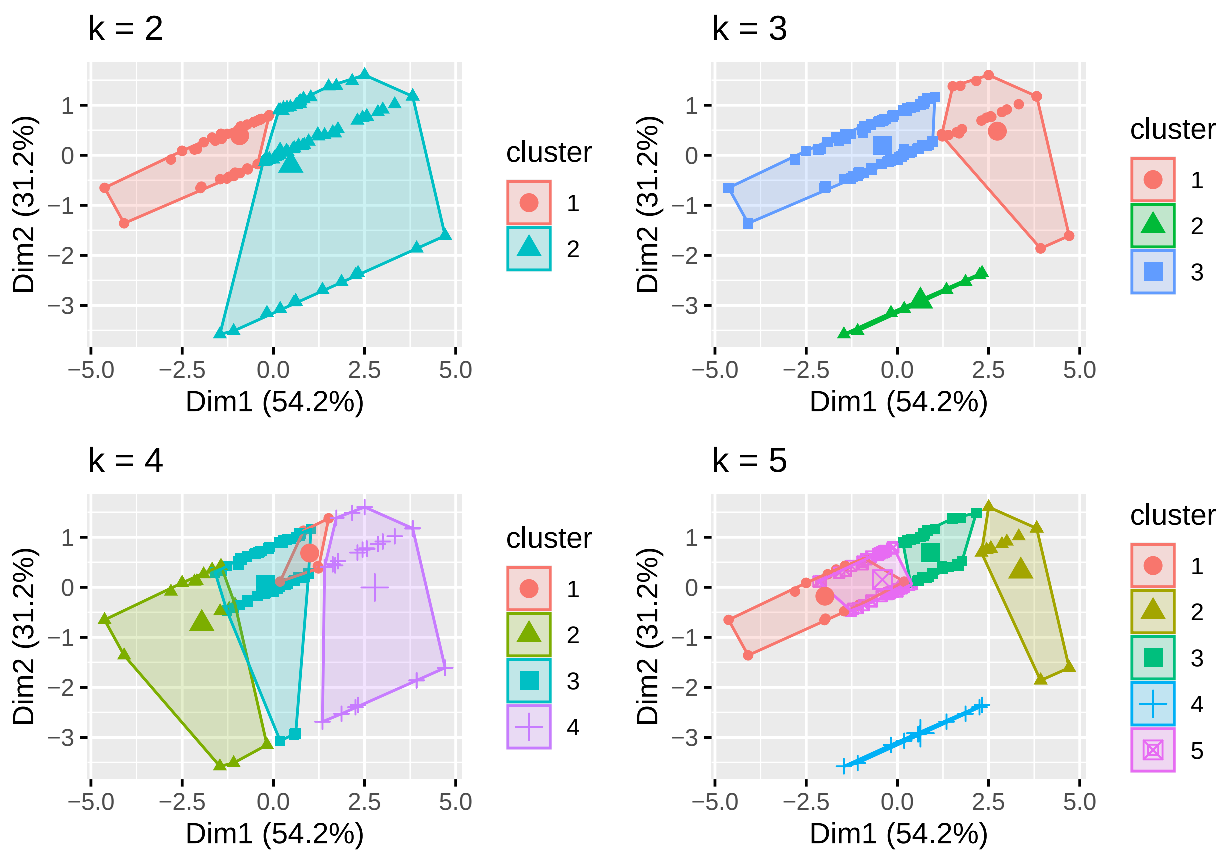
```r
fviz_cluster(k2, data = travel_df_clean[,2:4])
```



OTHER POSSIBLE k for k-MEANS

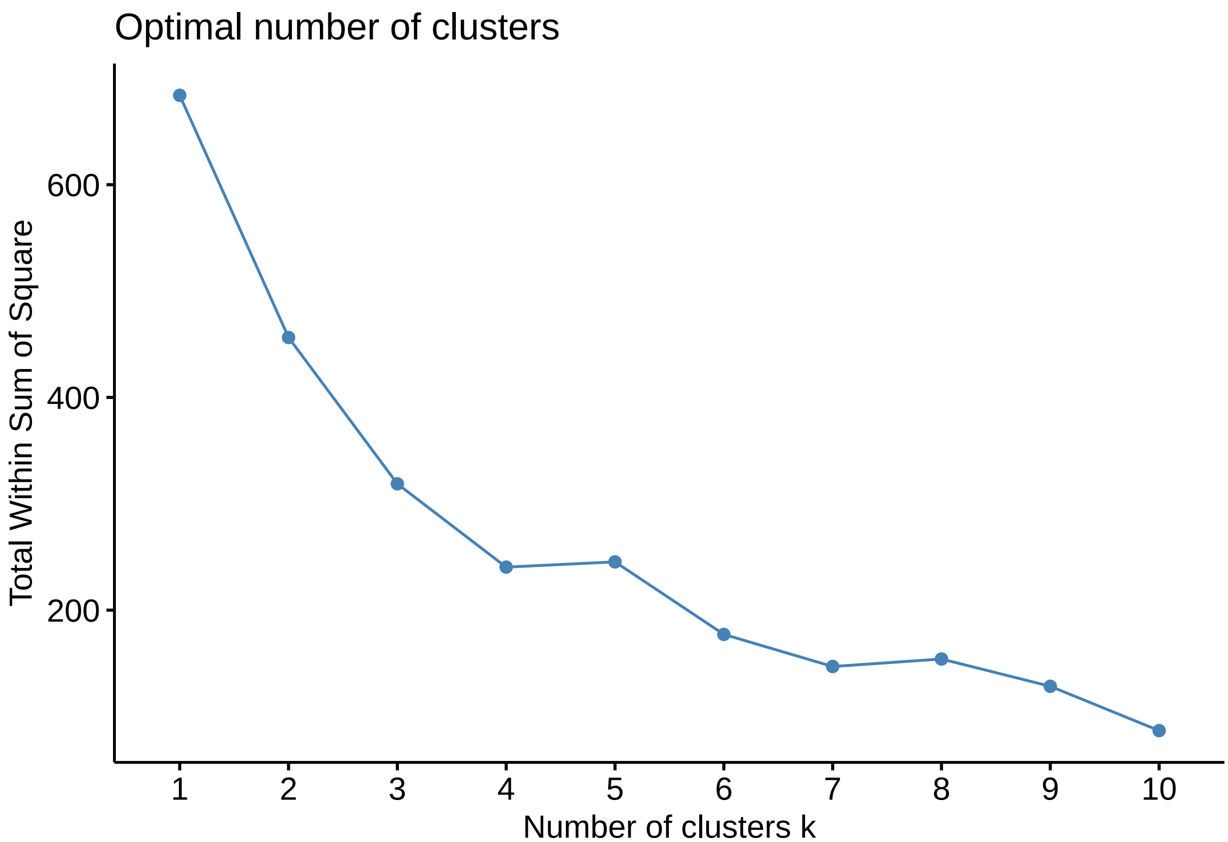
```r
k3 <- kmeans(travel_df_clean[,2:4], centers = 3)
k4 <- kmeans(travel_df_clean[,2:4], centers = 4)
k5 <- kmeans(travel_df_clean[,2:4], centers = 5)

# plots to compare
```

```
p1 <- fviz_cluster(k2, geom = "point", data =travel_df_clean[,2:4]) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = travel_df_clean[,2:4]) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = travel_df_clean[,2:4]) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = travel_df_clean[,2:4]) + ggtitle("k = 5")

grid.arrange(p1, p2, p3, p4, nrow = 2)
```



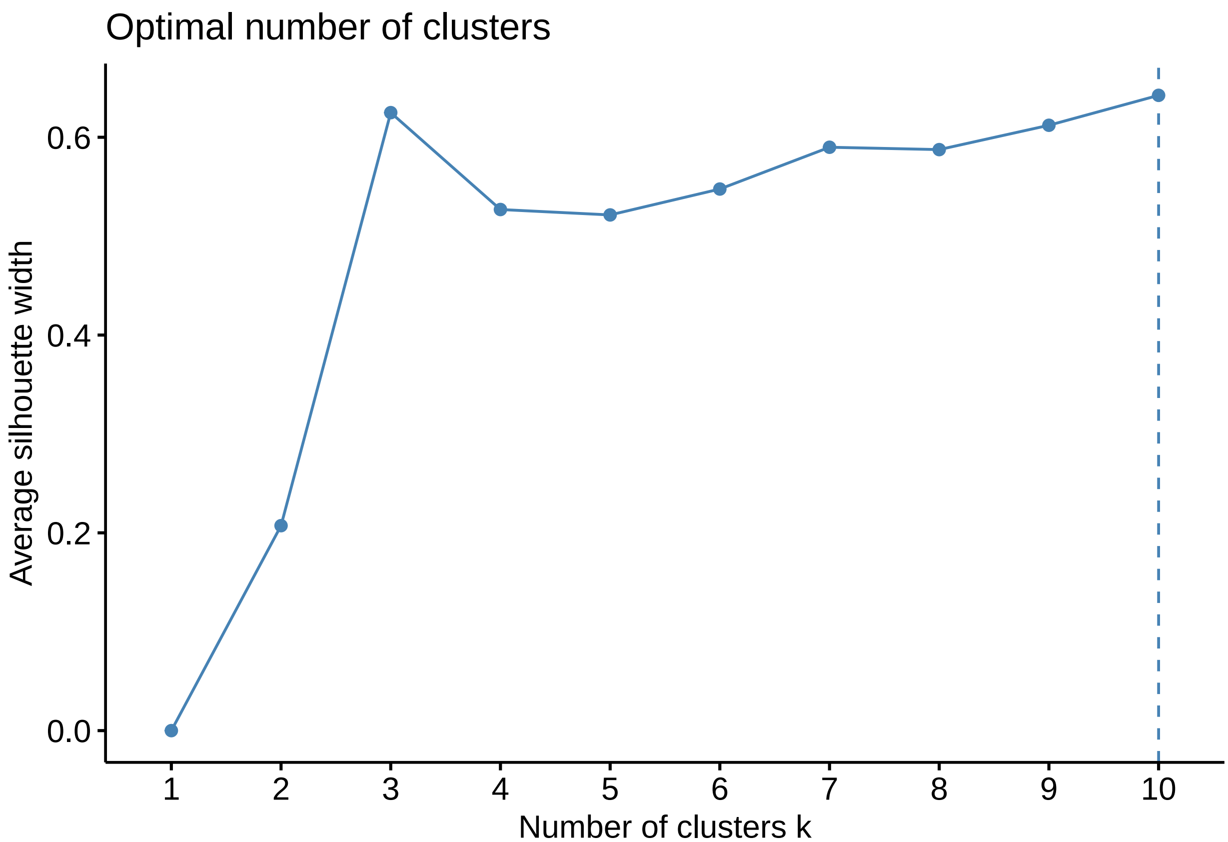
ELBOW METHOD

```
fviz_nbclust(travel_df_clean[,2:4], kmeans, method = "wss")
```



SILHOUETTE

```
fviz_nbclust(travel_df_clean[,2:4], kmeans, method = "silhouette")
```



OPTIMAL k = 4

```
final <- kmeans(travel_df_clean[,2:4], 4)
print(final)
```

```
## K-means clustering with 4 clusters of sizes 75, 116, 22, 16
##
## Cluster means:
##   travel_frequency checkin_exp      fly_exp
## 1      -0.65828850  -0.5772959 -0.538336329
## 2      -0.06428156  -0.0122976 -0.001411979
## 3       0.42371540   2.1250465  2.044011282
## 4       2.96915998  -0.1267068 -0.276827123
##
## Clustering vector:
##   [1] 1 1 2 1 3 1 1 3 1 1 1 1 4 4 3 2 1 2 2 1 2 1 1 2 3 1 2 1 4 1 2 2 2 2 2 2 1
##  [38] 1 2 2 1 2 2 1 2 2 2 2 1 2 3 2 4 3 2 1 1 1 4 1 1 2 2 2 2 2 2 2 1 2 1 1 1 2
##  [75] 3 2 2 2 2 1 3 3 2 2 4 1 2 1 4 2 1 2 2 2 1 4 1 1 2 1 1 2 2 2 2 1 2 4 3 2 2
## [112] 1 2 2 1 2 2 3 3 2 2 3 1 1 2 1 2 2 2 1 2 1 1 1 1 2 2 2 2 1 1 1 1 1 2 1 2 2
## [149] 2 2 4 3 1 2 2 2 1 2 1 2 2 2 2 3 3 2 2 1 2 2 2 1 2 2 2 3 2 1 4 3 2 2 2 2 2
## [186] 2 1 1 4 2 1 2 2 4 2 1 3 1 1 2 3 2 1 1 2 2 2 2 2 4 1 2 2 1 1 1 2 4 2 2 2 2
## [223] 2 2 2 4 1 3 3
##
## Within cluster sum of squares by cluster:
## [1] 110.05779  72.84050  61.23566  22.38637
##  (between_SS / total_SS =  61.0 %)
##
## Available components:
```

```
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
fviz_cluster(final, data = travel_df_clean[,2:4])
```