Example

```r
#loading in packages
library(readr)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```r
library(NbClust)
library(ggplot2)
library(cluster)
```

Exploring the Data

Data came from Stat 495 final project. (use info from project…). Needed a sample of 1000…

Importing the data:

```r
#using data from final stat 495 project
#library(readr)
data_subset <- read_csv("CopyOfdata_subset.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   geo_name = col_character(),
##   geo = col_character(),
##   zip = col_character(),
##   TRI.ID = col_character(),
##   County.x = col_character(),
##   County.y = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```r
set.seed(1)
#getting a sample of 1000 observations
mysample <- data_subset[sample(1:nrow(data_subset), 1000,
   replace=FALSE),]
```

Picking variables to focus on–> expanding conclusions from Stat 495 project

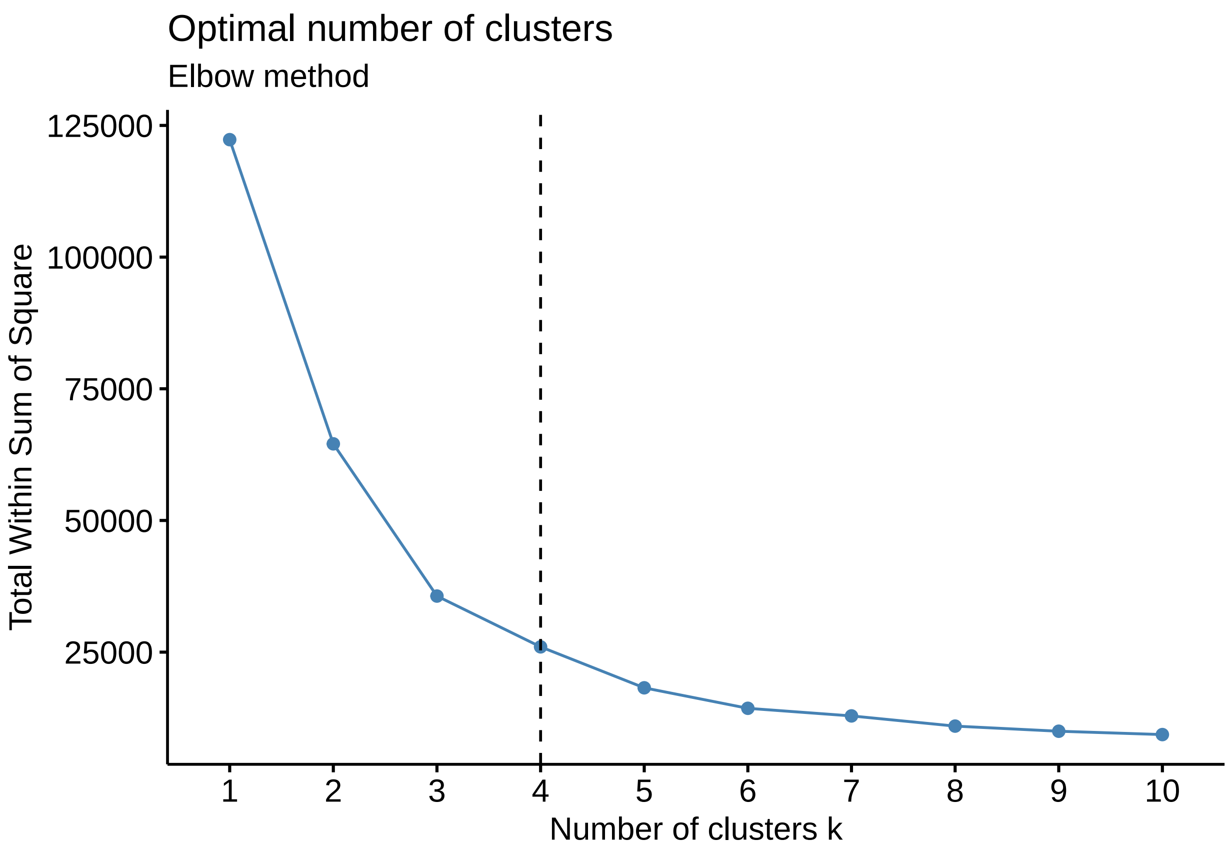
```r
#only keeping the variables I want to look at
myvars <- c("Latitude_tri", "Longitude_tri", "poor_or_fair_health", "poor_physical_health_days", "physic
smallsample <- mysample[myvars]
```

Applying CLARA

Step 1: finding k

```r
#finding k with project data, using Elbow Method
#pkgs <- c("factoextra",  "NbClust")
#install.packages(pkgs)
```

```r
#library(factoextra)
#library(NbClust)
#library(ggplot2)
new<- na.omit(smallsample)
# Elbow method
fviz_nbclust(new, kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```



Step 2: Run CLARA function

```r
#new<- na.omit(smallsample)
#library(cluster)
## run CLARA
clarasamp <- clara(new[1:6], 4)
```

```r
## print components of clarax
print(clarasamp)
```

```
## Call:     clara(x = new[1:6], k = 4)
## Medoids:
##       Latitude_tri Longitude_tri poor_or_fair_health
## [1,]       38.6364      -83.6929               0.200
## [2,]       40.3973      -75.9357               0.165
## [3,]       36.1335      -96.0532               0.196
## [4,]       45.4342     -123.0000               0.110
##       poor_physical_health_days physical_inactivity adult_obesity
## [1,]                        4.4               0.299         0.283
```

```
## [2,]                       3.7              0.245         0.308
## [3,]                       4.6              0.353         0.355
## [4,]                       3.3              0.137         0.244
## Objective function:   4.691022
## Clustering vector:    int [1:925] 1 2 1 3 1 3 1 1 1 1 1 3 1 2 1 3 3 3 ...
## Cluster sizes:            453 195 230 47
## Best sample:
##  [1]   5  11  24  86 139 149 162 175 177 192 208 224 242 285 306 311 316
## [18] 353 361 370 389 400 404 410 429 468 471 478 489 506 589 679 691 703
## [35] 719 726 736 741 800 811 815 818 877 882 883 895 902 918
##
## Available components:
##  [1] "sample"     "medoids"    "i.med"      "clustering" "objective"
##  [6] "clusinfo"   "diss"       "call"       "silinfo"    "data"
```

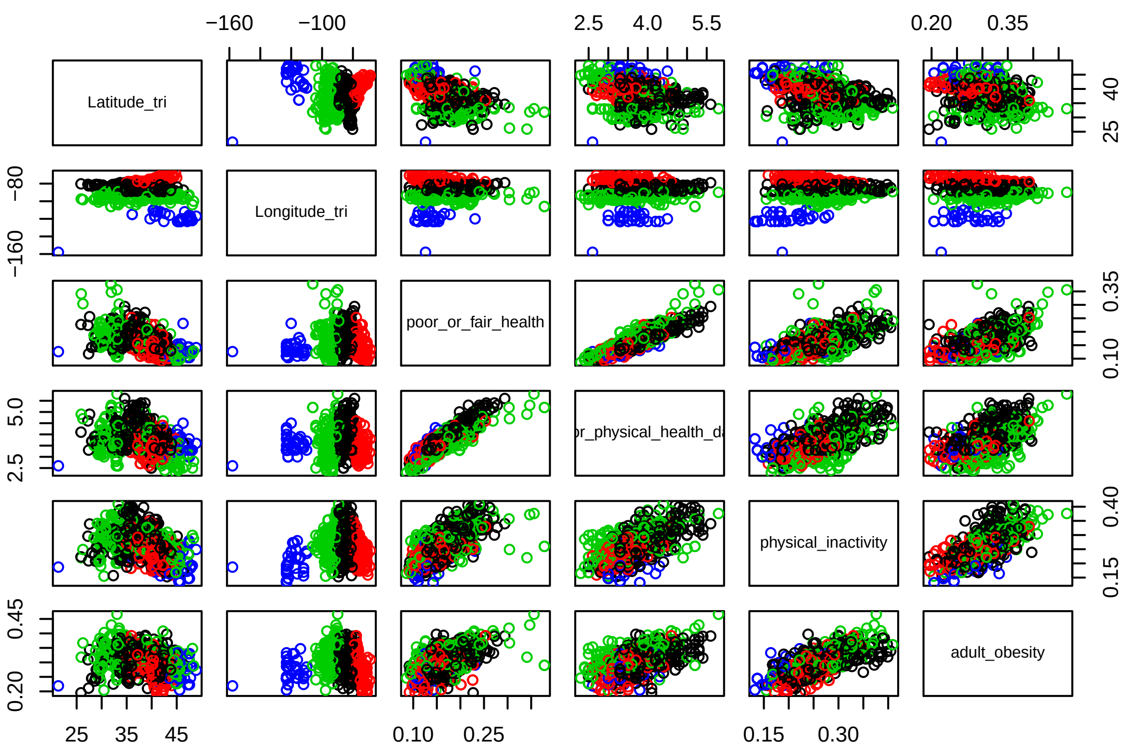
```
summary(clarasamp)
```

```
## Object of class 'clara' from call:
##  clara(x = new[1:6], k = 4)
## Medoids:
##      Latitude_tri Longitude_tri poor_or_fair_health
## [1,]      38.6364      -83.6929               0.200
## [2,]      40.3973      -75.9357               0.165
## [3,]      36.1335      -96.0532               0.196
## [4,]      45.4342     -123.0000               0.110
##      poor_physical_health_days physical_inactivity adult_obesity
## [1,]                       4.4              0.299         0.283
## [2,]                       3.7              0.245         0.308
## [3,]                       4.6              0.353         0.355
## [4,]                       3.3              0.137         0.244
## Objective function:    4.691022
## Numerical information per cluster:
##      size  max_diss  av_diss isolation
## [1,]  453 13.228882 4.578281  1.656594
## [2,]  195  8.392191 2.971767  1.050917
## [3,]  230 14.554453 6.249473  1.153918
## [4,]   47 42.497226 5.284278  1.489171
## Average silhouette width per cluster:
## [1] 0.2863797 0.6457187 0.4655863 0.9673973
## Average silhouette width of best sample: 0.4306859
##
## Best sample:
##  [1]   5  11  24  86 139 149 162 175 177 192 208 224 242 285 306 311 316
## [18] 353 361 370 389 400 404 410 429 468 471 478 489 506 589 679 691 703
## [35] 719 726 736 741 800 811 815 818 877 882 883 895 902 918
## Clustering vector:
##   [1] 1 2 1 3 1 3 1 1 1 1 1 3 1 2 1 3 3 3 3 2 1 3 2 1 1 3 3 2 2 1 1 2 3 1 3
##  [36] 1 1 1 3 3 1 1 1 1 1 1 1 1 1 2 1 1 3 1 1 1 1 4 1 1 1 1 2 2 3 3 1 2 1 3
##  [71] 1 1 3 3 1 3 1 4 1 2 1 2 4 1 1 3 1 2 4 3 3 3 1 1 3 4 4 1 2 2 1 2 3 3 1
## [106] 1 1 3 3 4 1 3 3 4 2 1 2 3 2 3 2 2 1 3 1 2 1 1 1 3 2 4 1 2 1 1 2 1 3 3
## [141] 1 2 1 1 1 1 2 1 1 2 2 1 1 3 3 1 3 1 3 1 3 1 1 2 1 3 1 4 3 1 1 3 1 3 3
## [176] 1 3 1 1 4 2 1 1 1 1 2 1 1 3 1 1 3 2 1 2 3 3 2 1 1 1 2 1 3 2 1 2 1 1 1
## [211] 1 1 1 2 1 3 4 2 4 3 1 1 3 2 1 1 3 4 3 2 2 1 1 1 3 3 1 1 3 1 1 2 1 2 3
## [246] 2 1 1 3 1 1 3 4 2 1 1 2 3 1 2 1 1 1 3 1 4 2 3 2 1 3 2 2 3 1 1 3 2 1 3
## [281] 1 1 3 3 1 1 3 1 2 2 3 1 1 3 1 2 1 1 1 1 1 3 1 1 1 1 3 3 1 3 1 1 3 1 1
```

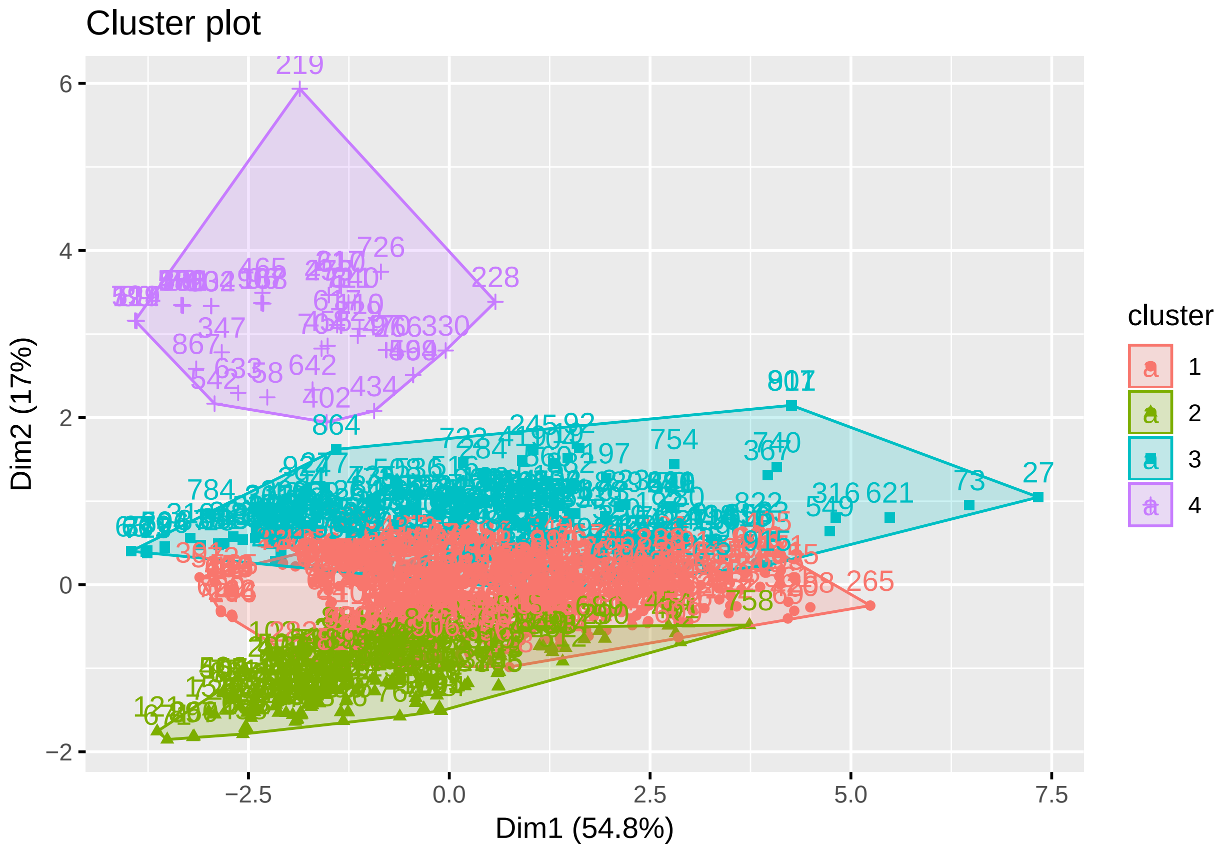
```
## [316] 3 1 2 1 1 1 3 2 3 1 1 3 1 1 4 2 1 1 1 3 1 2 1 2 2 1 3 1 1 3 3 4 3 1 1
## [351] 1 1 1 1 1 3 1 3 1 1 1 3 3 1 2 1 3 1 2 3 1 3 3 2 2 2 1 1 3 2 3 3 3 1 1
## [386] 1 1 2 1 3 1 1 2 1 2 2 1 3 3 2 1 4 2 2 1 1 1 1 1 1 3 1 1 1 1 1 2 1 3 1
## [421] 3 3 2 3 1 2 1 3 2 1 1 1 1 4 4 1 1 3 1 2 1 1 1 3 1 2 1 1 2 1 3 1 1 2 4
## [456] 1 1 2 1 1 1 2 1 1 4 3 1 1 4 4 1 1 3 1 4 1 1 1 1 3 4 2 1 1 1 1 2 2 3 1
## [491] 2 3 1 3 3 1 2 2 2 2 1 2 2 3 1 1 2 3 2 1 1 2 3 2 3 3 2 2 2 3 2 1 3 2 3
## [526] 3 3 1 4 1 1 2 2 2 1 3 2 1 1 1 3 4 1 1 3 1 1 2 3 1 2 1 3 1 1 3 2 3 3 3
## [561] 1 2 1 2 1 3 2 3 1 4 1 1 3 3 2 1 1 3 2 2 1 1 2 1 2 1 1 1 1 3 2 1 3 4 1
## [596] 3 1 1 2 1 3 2 3 4 1 3 1 1 1 4 3 1 2 1 1 3 4 2 1 2 3 2 1 2 3 1 1 2 1 2
## [631] 1 1 4 3 1 2 3 1 1 3 1 4 3 1 1 1 2 2 3 2 1 1 1 2 1 1 1 2 3 1 3 1 1 3 1
## [666] 1 1 1 3 1 2 1 3 1 1 1 1 1 2 2 3 1 2 3 2 2 1 1 2 2 3 2 1 1 1 2 1 3 3 1
## [701] 1 1 1 4 2 1 3 1 1 2 4 3 1 1 1 3 3 1 1 1 1 3 1 1 2 4 2 3 1 1 1 3 1 1 3
## [736] 1 2 1 2 3 2 2 2 1 3 3 3 1 1 1 2 3 3 3 1 3 3 2 1 2 3 3 1 2 4 2 1 1 3 1
## [771] 2 1 1 1 3 1 3 3 1 1 3 3 1 3 1 2 1 1 3 1 4 1 1 1 1 3 2 2 2 4 1 3 3 1 1
## [806] 1 3 1 1 1 2 1 1 1 2 2 2 1 1 2 1 3 3 3 3 1 3 2 4 2 1 2 3 1 1 3 2 1 1 4
## [841] 2 2 2 2 2 1 3 1 2 1 2 4 3 3 1 1 3 1 1 1 1 1 2 3 3 2 4 1 2 1 1 1 1 2 1
## [876] 1 3 1 1 1 3 3 1 1 1 3 1 1 3 1 1 3 1 3 1 3 2 2 2 3 1 2 2 2 1 1 4 1 2 1
## [911] 3 1 3 2 3 3 1 1 2 1 3 1 1 3 1
##
## Silhouette plot information for best sample:
##     cluster neighbor   sil_width
## 11        1        2  0.51160188
## 703       1        2  0.50784463
## 149       1        3  0.50590891
## 208       1        2  0.48693148
## 162       1        3  0.47468187
## 471       1        2  0.44956360
## 410       1        2  0.42362707
## 389       1        2  0.42157038
## 5         1        2  0.41457596
## 719       1        2  0.41289559
## 468       1        2  0.37020730
## 361       1        2  0.35061984
## 306       1        2  0.34569720
## 285       1        2  0.34506744
## 818       1        3  0.30770633
## 506       1        3  0.21671937
## 589       1        2  0.20444489
## 918       1        2  0.19851903
## 736       1        3  0.18549695
## 478       1        2  0.16554915
## 24        1        2  0.15565598
## 311       1        2  0.04768370
## 353       1        3  0.03173421
## 883       1        2 -0.12930919
## 895       1        2 -0.24550081
## 224       2        1  0.76643863
## 679       2        1  0.75957431
## 902       2        1  0.74802681
## 404       2        1  0.74508892
## 242       2        1  0.73417881
## 400       2        1  0.73283435
## 741       2        1  0.61737358
## 811       2        1  0.47833894
```

```
## 815        2        1  0.43844208
## 429        2        1  0.43689016
## 691        3        1  0.62256512
## 370        3        1  0.62175433
## 489        3        1  0.61988506
## 882        3        1  0.60671130
## 139        3        1  0.58178390
## 177        3        1  0.55053388
## 175        3        1  0.43464488
## 86         3        1  0.29525285
## 192        3        1  0.28445790
## 316        3        1  0.27777790
## 877        3        1  0.22608234
## 800        4        3  0.96743579
## 726        4        3  0.96735884
##
## 1128 dissimilarities, summarized :
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0507  6.3776 10.6340 12.8320 15.7130 51.8530
## Metric :  euclidean
## Number of objects : 48
##
## Available components:
##  [1] "sample"     "medoids"    "i.med"      "clustering" "objective"
##  [6] "clusinfo"   "diss"       "call"       "silinfo"    "data"
```

```r
## plot clusters
plot(new, col = clarasamp$cluster)
## plot centers
points(clarasamp$centers, col = 1:2, pch = 8)
```



```r
#plotting clara
factoextra::fviz_cluster(clarasamp)
```



Evaluation of CLARA

Model to Predict Cluster

First, had to include a cluster variable in the original data set, using the data provided by the CLARA function.

```r
#adding each data point's cluster #
cluster<- clarasamp$clustering
cluster_data<- cbind(new, cluster)
```