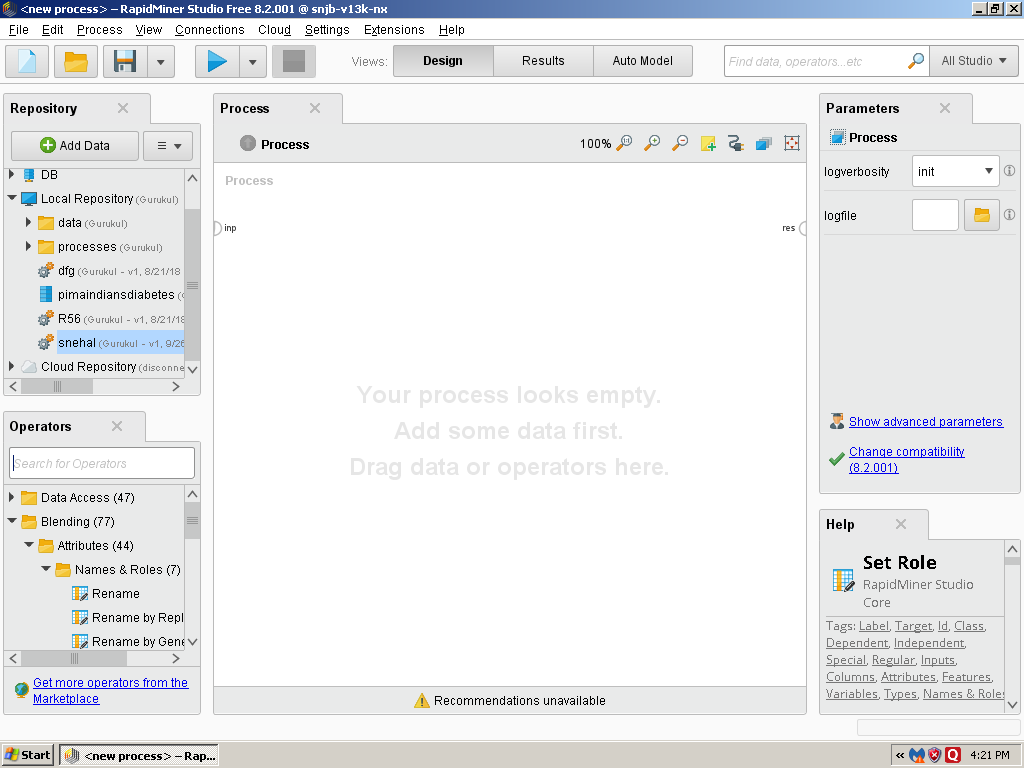
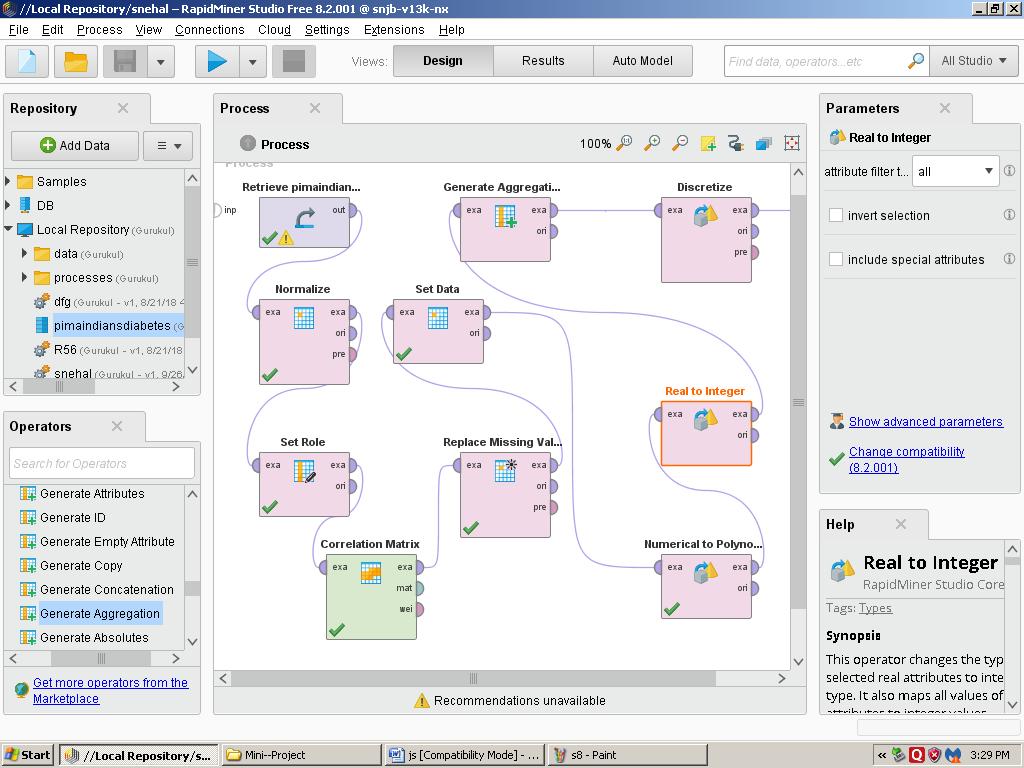
Mini Project(DMW)

**Dataset:-**Pimaindiansdiabets.

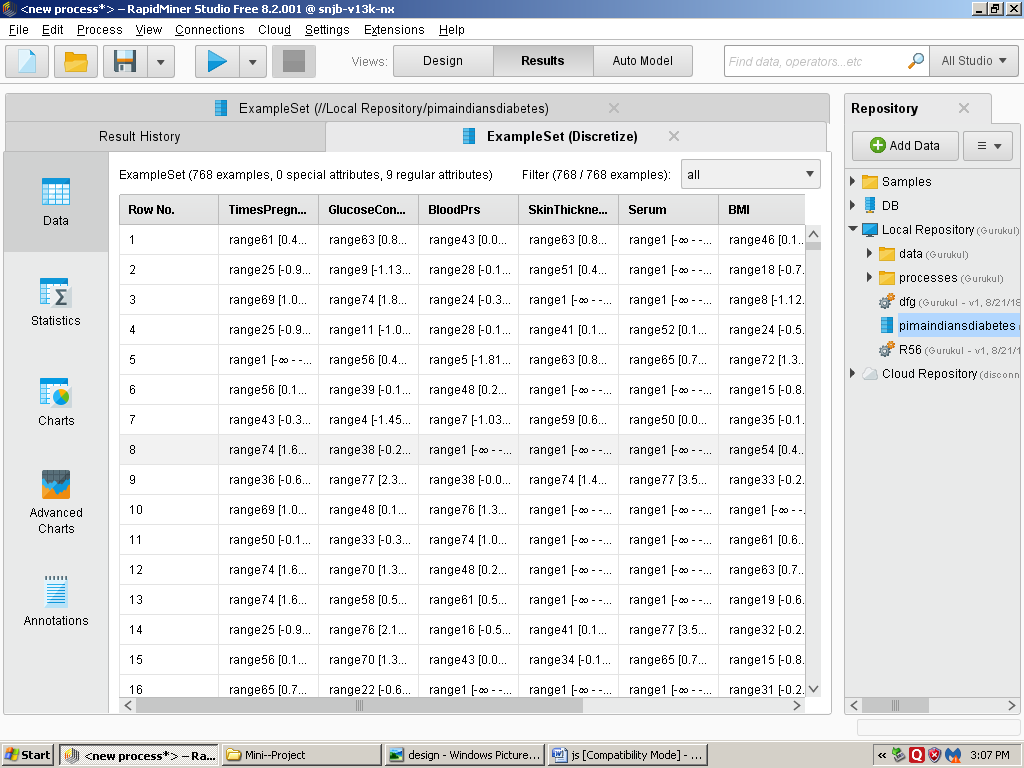
1. Open rapidminer select file=>new process=>blank=>start, following screen is appear.



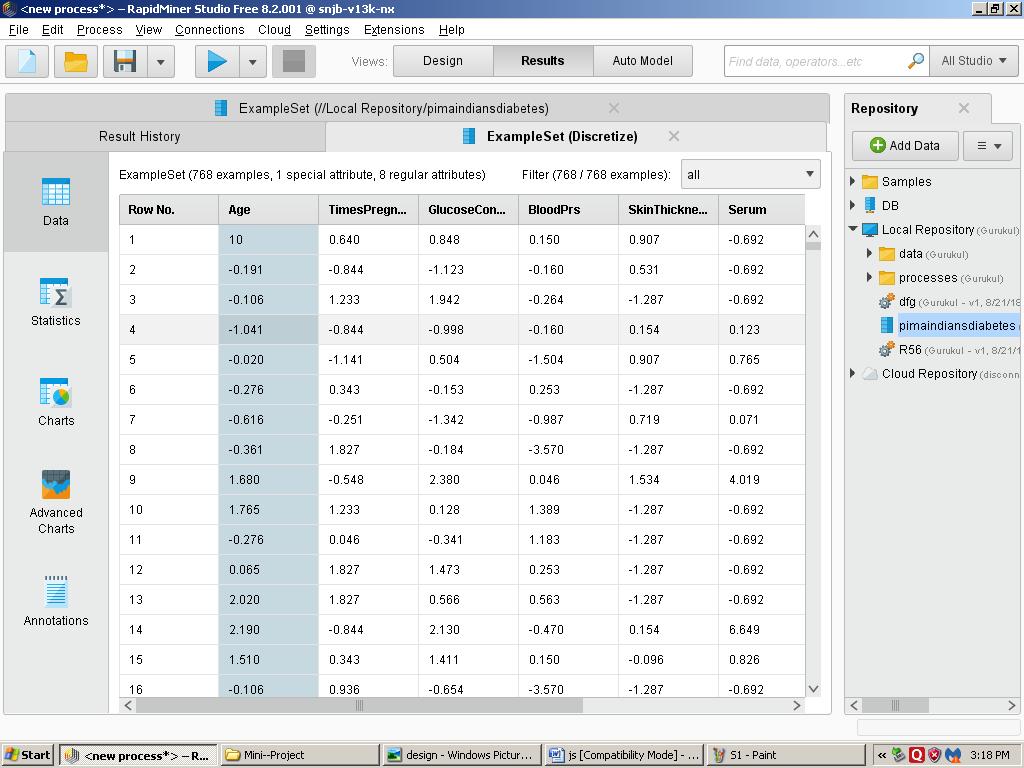
2)Add dataset on Retrieveoperation then perform the different operations with using different operators.



1. Press F11 i.e. run process locally then result will be display.

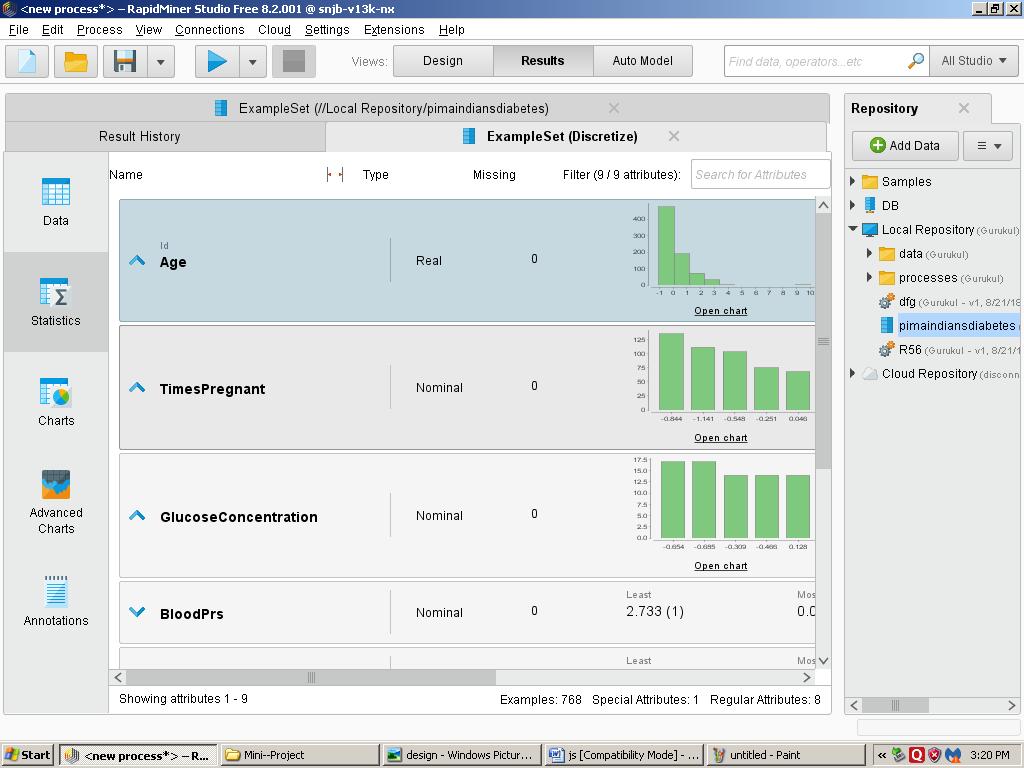


1)

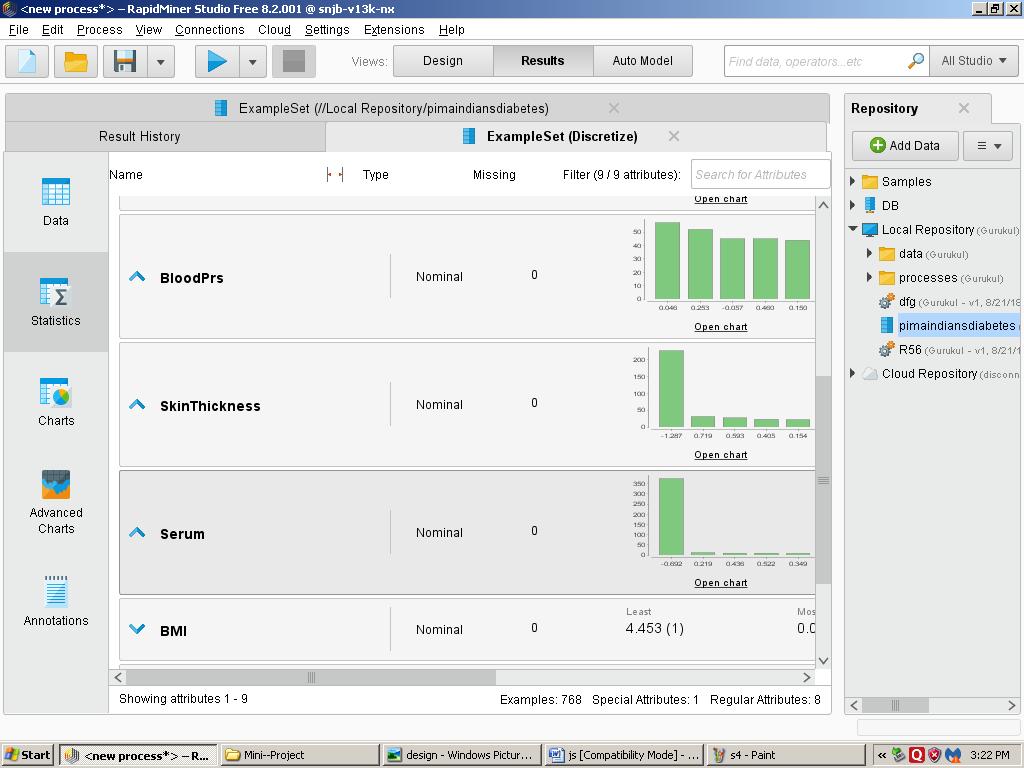


2)

1. If you want to view dataset satistically then press statistics.



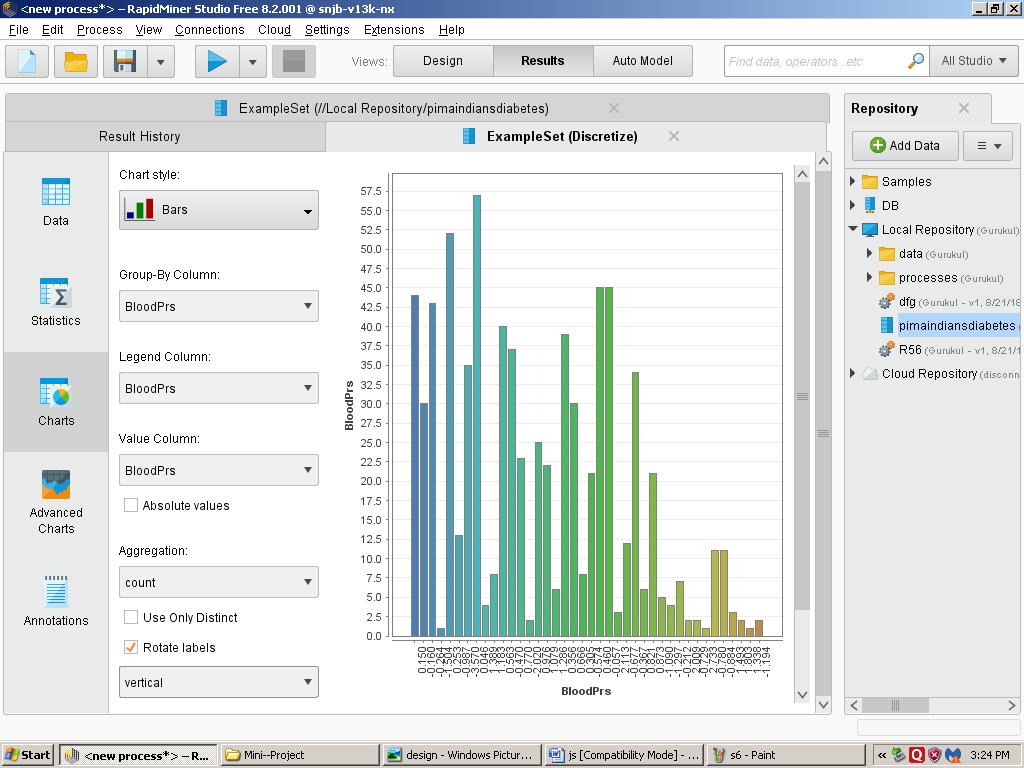
1)



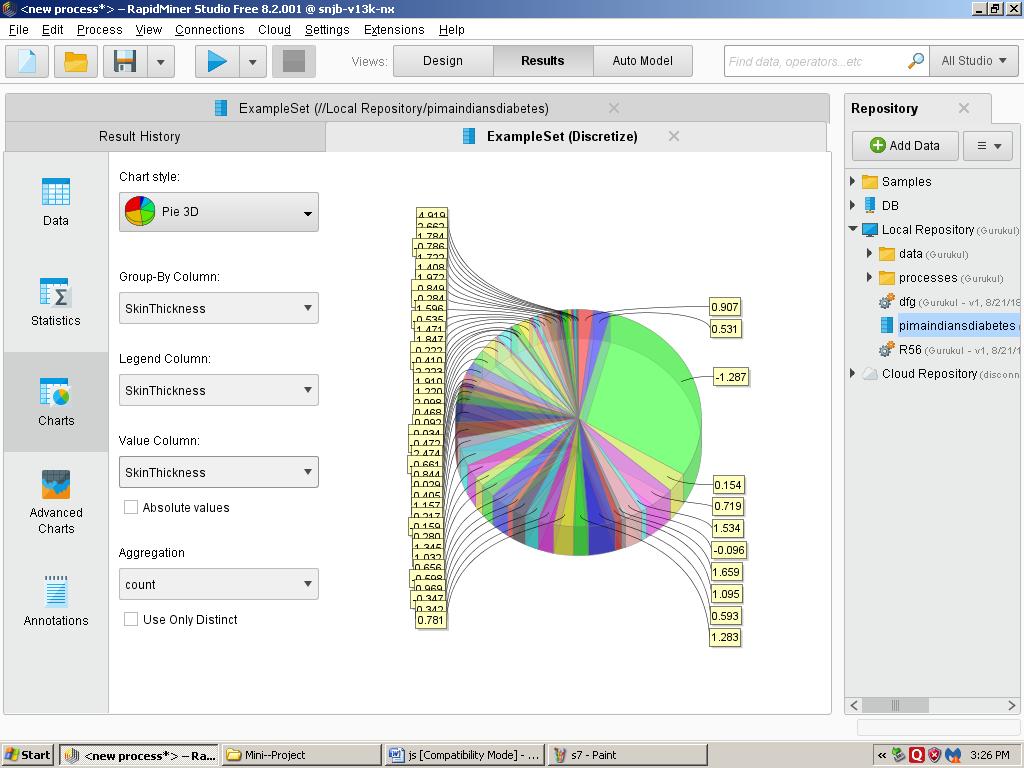
2)

1. If you want to view dataset in chart style then press chart.

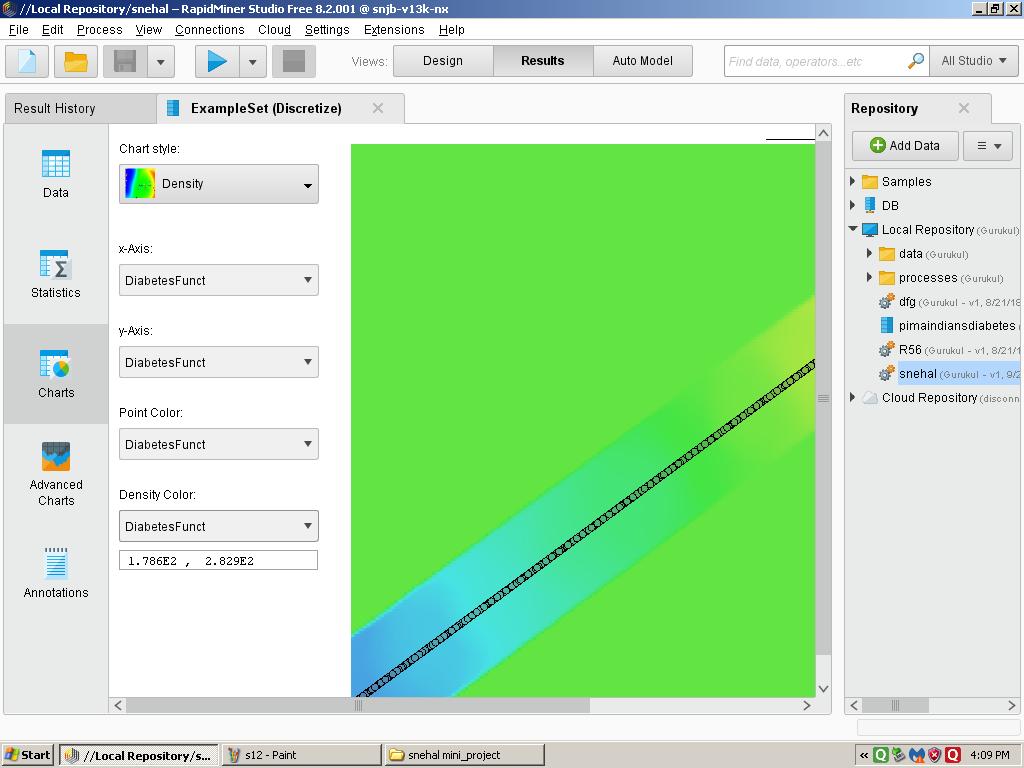
i)Bars.



ii)Pie 3D



iii)Density



Clustering using R-Programming

install.packages("factoextra")

install.packages("cluster")

install.packages("magrittr")

library("cluster")

library("factoextra")

library("magrittr")

**Data preparation**

* Demo data set: the built-in R dataset named USArrest
* Remove missing data
* Scale variables to make them comparable

# Load and prepare the data

data("USArrests")

my\_data <- USArrests %>%

na.omit() %>% # Remove missing values (NA)

scale() # Scale variables

# View the firt 3 rows

head(my\_data, n = 3)

Murder Assault UrbanPop Rape

Alabama 1.24256408 0.7828393 -0.5209066 -0.003416473

Alaska 0.50786248 1.1068225 -1.2117642 2.484202941

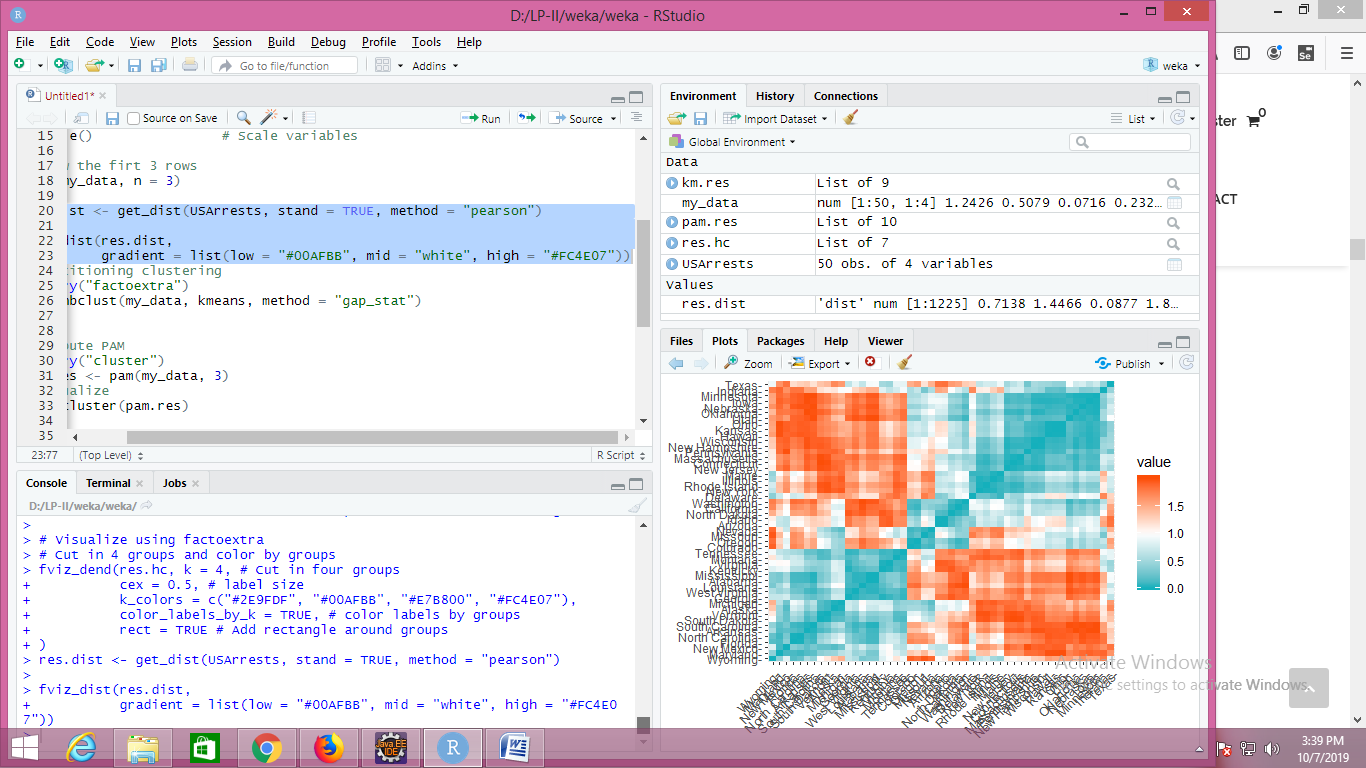
Arizona 0.07163341 1.4788032 0.9989801 1.042878388

**Distance measures**

The classification of objects, into clusters, requires some methods for measuring the distance or the (dis)similarity between the objects. Chapter [Clustering Distance Measures Essentials](https://www.datanovia.com/en/lessons/clustering-distance-measures/) covers the common distance measures used for assessing similarity between observations.

It’s simple to compute and visualize distance matrix using the functions [get\_dist() and fviz\_dist()](https://rpkgs.datanovia.com/factoextra/reference/dist.html) [factoextra R package]:

* get\_dist(): for computing a distance matrix between the rows of a data matrix. Compared to the standard dist() function, it supports correlation-based distance measures including “pearson”, “kendall” and “spearman” methods.
* fviz\_dist(): for visualizing a distance matrix
* res.dist <- get\_dist(USArrests, stand = TRUE, method = "pearson")
* fviz\_dist(res.dist,
* gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))



**Partitioning clustering**

Partitioning algorithms are clustering techniques that subdivide the data sets into a set of k groups, where k is the number of groups pre-specified by the analyst.

There are different types of partitioning clustering methods. The most popular is the [K-means clustering](https://www.datanovia.com/en/lessons/k-means-clustering-in-r-algorith-and-practical-examples/) (MacQueen 1967), in which, each cluster is represented by the center or means of the data points belonging to the cluster. The K-means method is sensitive to outliers.

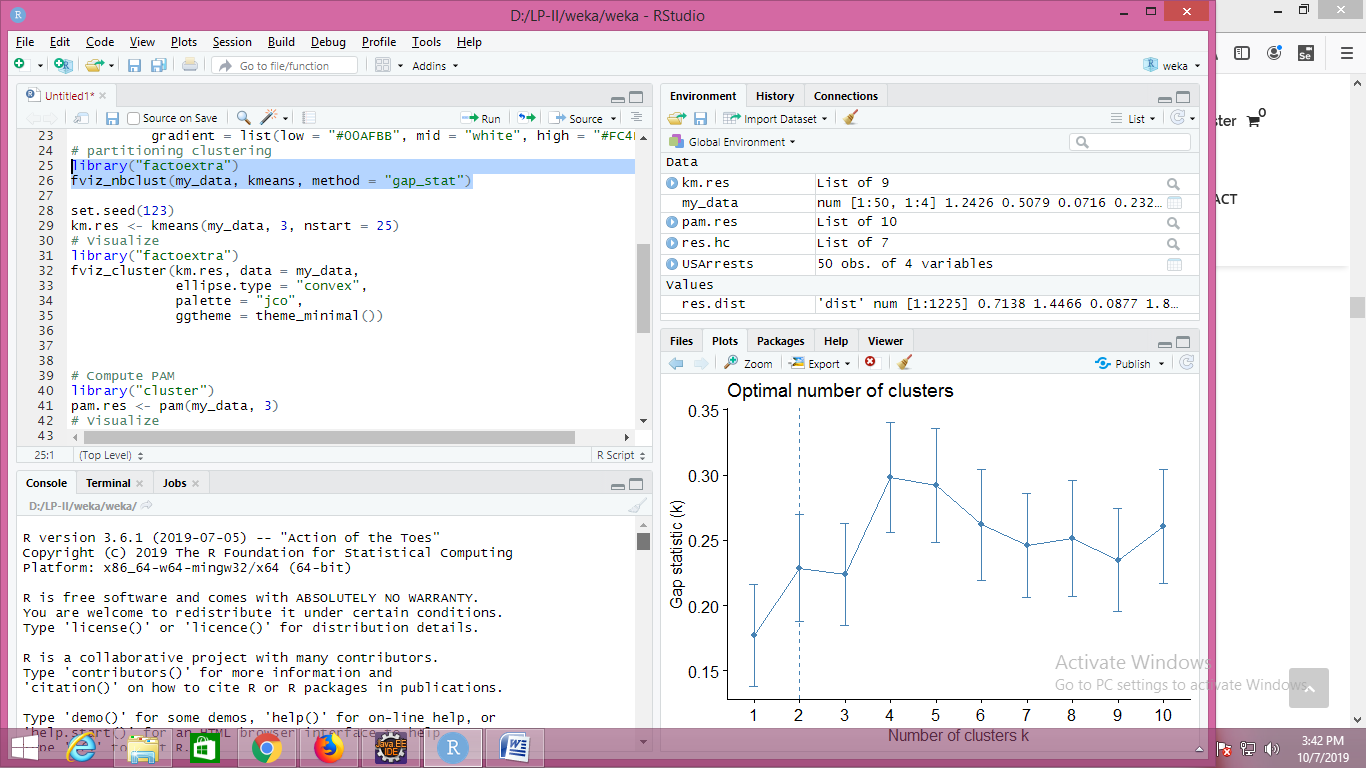
An alternative to k-means clustering is the [K-medoids clustering](https://www.datanovia.com/en/lessons/k-medoids-in-r-algorithm-and-practical-examples/) or PAM (Partitioning Around Medoids, Kaufman & Rousseeuw, 1990), which is less sensitive to outliers compared to k-means.

Read more: [Partitioning Clustering methods](https://www.datanovia.com/en/courses/partitional-clustering-in-r-the-essentials/).

The following R codes show how to determine the optimal number of clusters and how to compute k-means and PAM clustering in R.

[Determining the optimal number of clusters](https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/): use factoextra::fviz\_nbclust()

1. library("factoextra")
2. fviz\_nbclust(my\_data, kmeans, method = "gap\_stat")



set.seed(123)

km.res <- kmeans(my\_data, 3, nstart = 25)

# Visualize

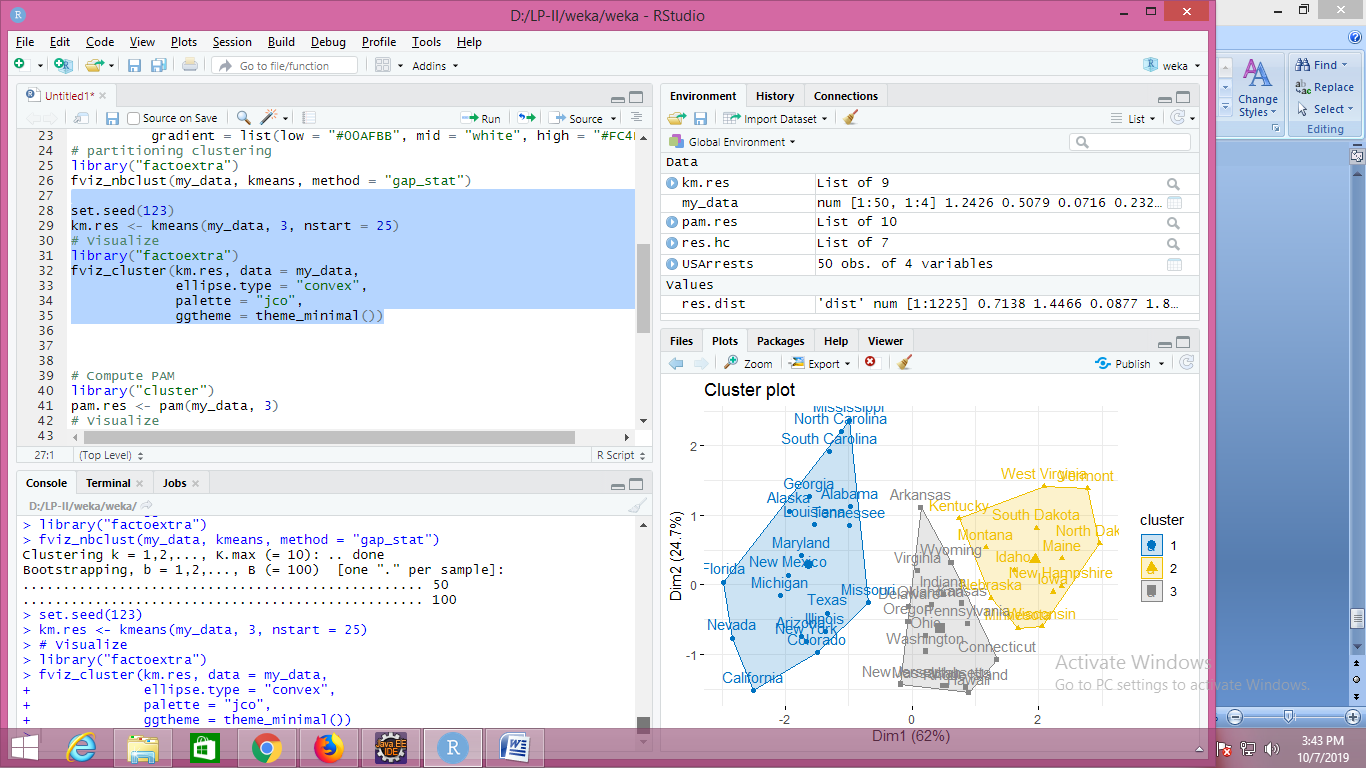
library("factoextra")

fviz\_cluster(km.res, data = my\_data,

ellipse.type = "convex",

palette = "jco",

ggtheme = theme\_minimal())



# Compute PAM

library("cluster")

pam.res <- pam(my\_data, 3)

# Visualize

fviz\_cluster(pam.res)

## Hierarchical clustering

Hierarchical clustering is an alternative approach to partitioning clustering for identifying groups in the dataset. It does not require to pre-specify the number of clusters to be generated.

The result of hierarchical clustering is a tree-based representation of the objects, which is also known as dendrogram. Observations can be subdivided into groups by cutting the dendrogram at a desired similarity level.

R code to compute and visualize hierarchical clustering:

# Compute hierarchical clustering

res.hc <- USArrests %>%

scale() %>% # Scale the data

dist(method = "euclidean") %>% # Compute dissimilarity matrix

hclust(method = "ward.D2") # Compute hierachical clustering

# Visualize using factoextra

# Cut in 4 groups and color by groups

fviz\_dend(res.hc, k = 4, # Cut in four groups

cex = 0.5, # label size

k\_colors = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),

color\_labels\_by\_k = TRUE, # color labels by groups

rect = TRUE # Add rectangle around groups

)

