Bhanuja Nagore

ADS | 001994132

Week 5 Assignment

Report

# Hierarchial clustering:

In [data mining](https://en.wikipedia.org/wiki/Data_mining) and [statistics](https://en.wikipedia.org/wiki/Statistics), hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) which seeks to build a [hierarchy](https://en.wikipedia.org/wiki/Hierarchy) of clusters.

Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

* Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.
* Compute distances (similarities) between the new cluster and each of the old clusters.
* Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Distance between clusters:

* Single-linkage (connectedness or minimum):

Distance between two clusters to be equal to the shortest distance from any member of one cluster to any member of the other cluster.

* Complete-linkage clustering (diameter or maximum):

Distance between two clusters to be equal to the greatest distance from any member of one cluster to any member of the other cluster.

* Average-linkage clustering:

Distance between two clusters to be equal to the average distance from any member of one cluster to any member of the other cluster.

## Animal Hierarchial Clustering:

This data set considers 6 binary attributes for 20 animals.

**Define variables:**

animals <- cluster::animals

colnames(animals) <- c("warm-blooded",

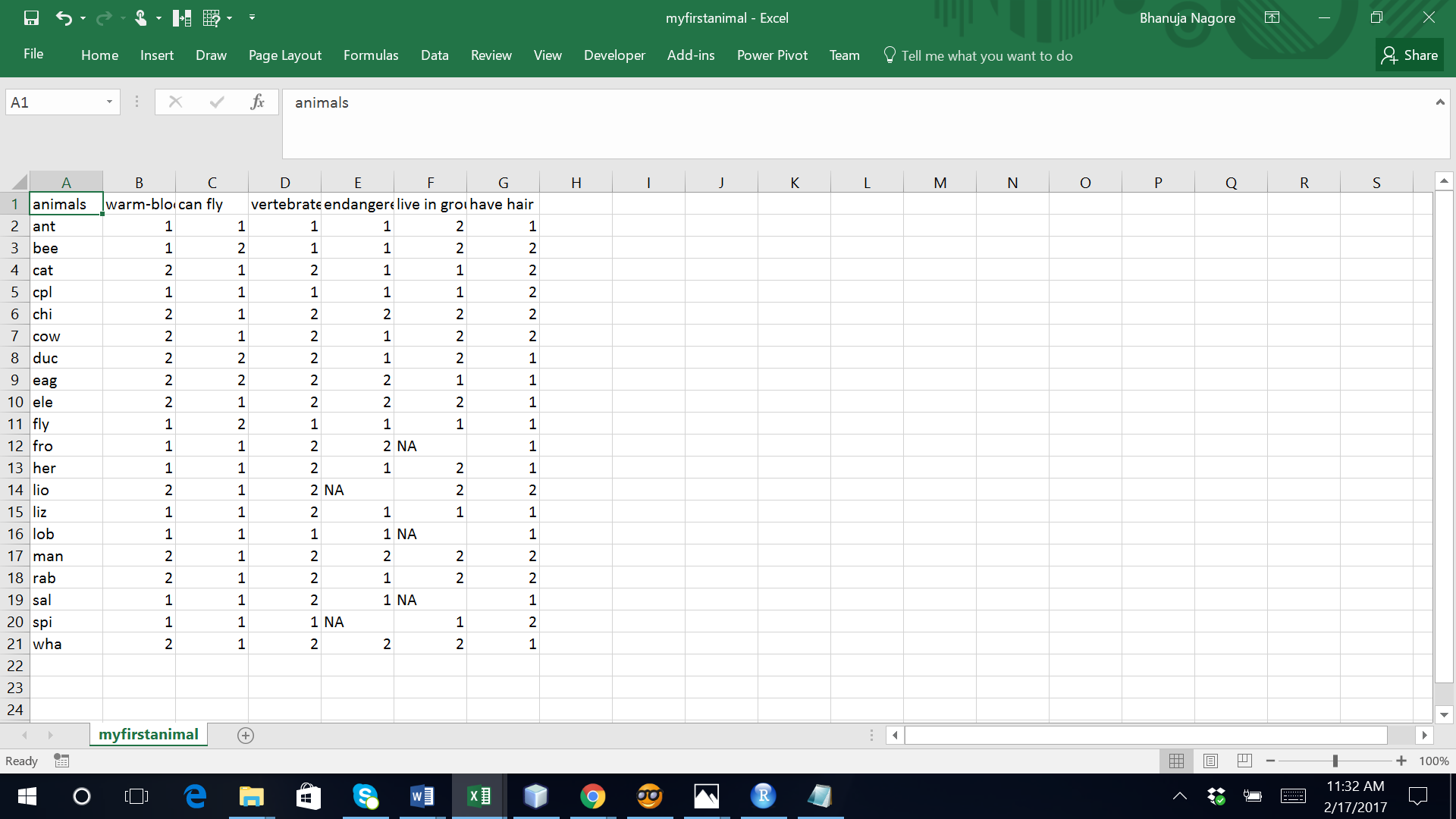
"can fly",

"vertebrate",

"endangered",

"live in groups",

"have hair")



library(dendextend)

library(corrplot)

animals <- cluster::animals

colnames(animals) <- c("warm-blooded",

"can fly",

"vertebrate",

"endangered",

"live in groups",

"have hair")

dend\_r <- animals %>% dist(method = "man") %>% hclust(method = "ward.D") %>% as.dendrogram %>% ladderize %>%

color\_branches(k=4)

dend\_r

plot(dend\_r)

dend\_c <- t(animals) %>% dist(method = "man") %>% hclust(method = "com") %>% as.dendrogram %>% ladderize%>%

color\_branches(k=3)

plot(dend\_c)

# some\_col\_func <- function(n) rev(colorspace::heat\_hcl(n, c = c(80, 30), l = c(30, 90), power = c(1/5, 1.5)))

# some\_col\_func <- colorspace::diverge\_hcl

# some\_col\_func <- colorspace::sequential\_hcl

some\_col\_func <- function(n) (colorspace::diverge\_hcl(n, h = c(246, 40), c = 96, l = c(65, 90)))

# par(mar = c(3,3,3,3))

# library(gplots)

gplots::heatmap.2(as.matrix(animals-1),

main = "Attributes of Animals",

srtCol = 35,

Rowv = animals,

Colv = t(animals),

trace="row", hline = NA, tracecol = "darkgrey",

margins =c(6,3),

key.xlab = "no / yes",

denscol = "grey",

density.info = "density",

col = some\_col\_func

)

hclust\_methods <- c("ward.D", "single", "complete", "average", "mcquitty",

"median", "centroid", "ward.D2")

animals\_dendlist <- dendlist()

for(i in seq\_along(hclust\_methods)) {

tmp\_dend <- animals %>% dist(method = "man") %>%

hclust(method = hclust\_methods[i]) %>% as.dendrogram

animals\_dendlist <- dendlist(animals\_dendlist, tmp\_dend)

}

names(animals\_dendlist) <- hclust\_methods

# votes.repub\_dendlist

cophenetic\_cors <- cor.dendlist(animals\_dendlist)

corrplot::corrplot(cophenetic\_cors, "pie", "lower")

remove\_median <- dendlist(animals\_dendlist, which = c(1:8)[-6] )

FM\_cors <- cor.dendlist(remove\_median, method = "FM\_index", k = 4)

corrplot::corrplot(FM\_cors, "pie", "lower")

data(animals)

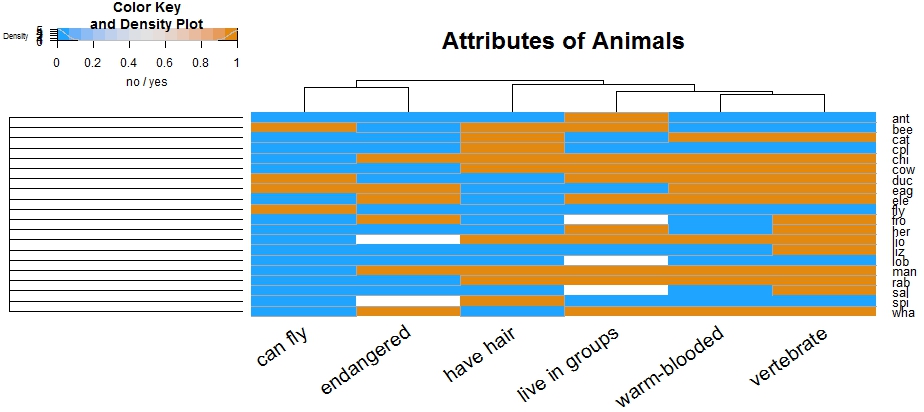
write.csv(animals, file = "myfirstani.csv", row.names = FALSE)

str(animals)

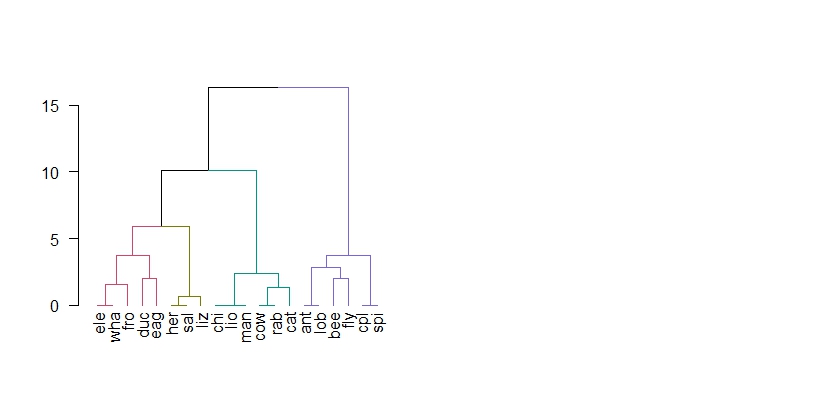
Several groups of variables: the “can fly” and “endangered” (which usually are both “no”), the “have hair”, and the “warm-blooded”, “vertebrate”, and “live in groups”.

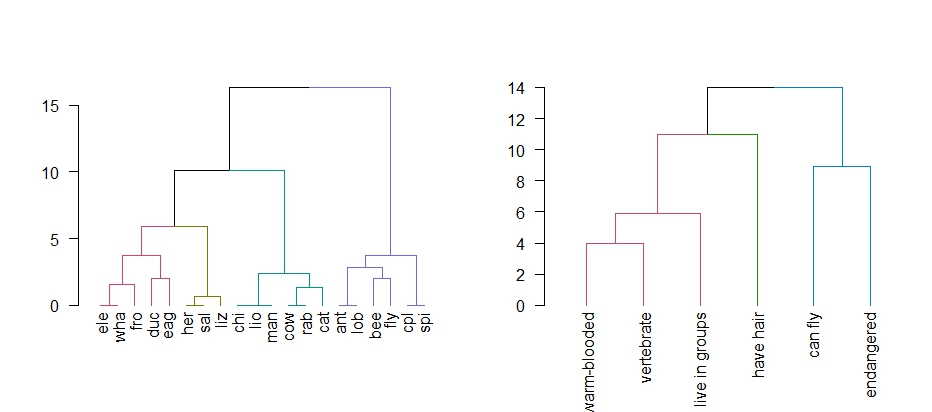
We see that within the animals there are (roughly!) the following 4 groups:

* The cold-blooded non-vertebrates, which are mostly not endangered.
* The warm-blooded vertebrates, which live in groups, have hair, cannot fly, and mostly are not endangered.
* The cold-blooded vertebrates, without hair, cannot fly, and are not endangered.
* The (mostly) warm-blooded vertebrates, without hair, some can fly, and some are endangered.

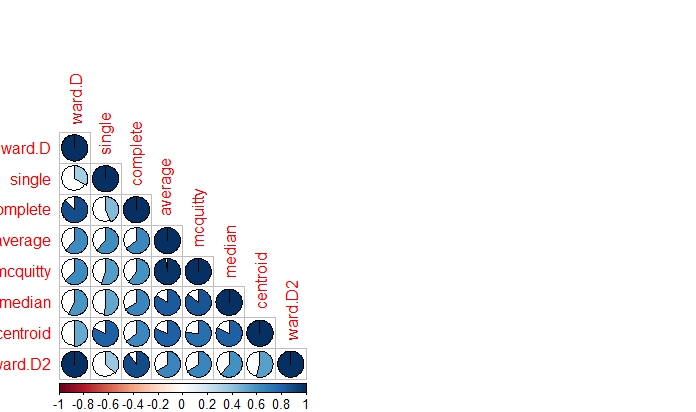


We first calculate the clustering using 8 different methods:

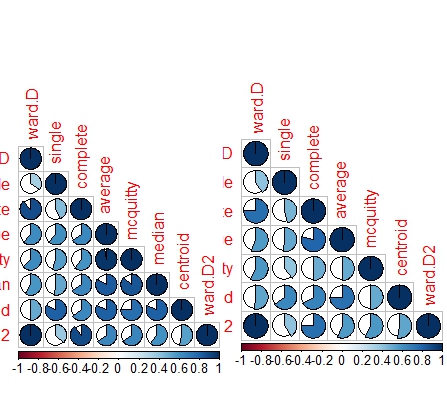




We see that the different methods (other than ward.D and ward.D2), all give quite different results.



For this purpose, we compare the clustering solution of each algorithm with one another, when cut to k=4 clusters, using the Fowlkes-Mallows Index. This measure is similar to rand (or rand adjusted) index, and gives a value of 1 when the two clusters conform, and 0 when they do not:



We removed the “median” method since it did not have k=4 possible. In general, the results seems sensitive to the algorithm used, and the different algorithm methods do not seem to agree with one another (with regards to k=4), so further analyses may be in place in order to decide on which algorithm and interpretation are most appropriate for these data.

(Other possible data sets for the future: chorSub, flower, plantTraits, pluton, ruspini, agriculture)

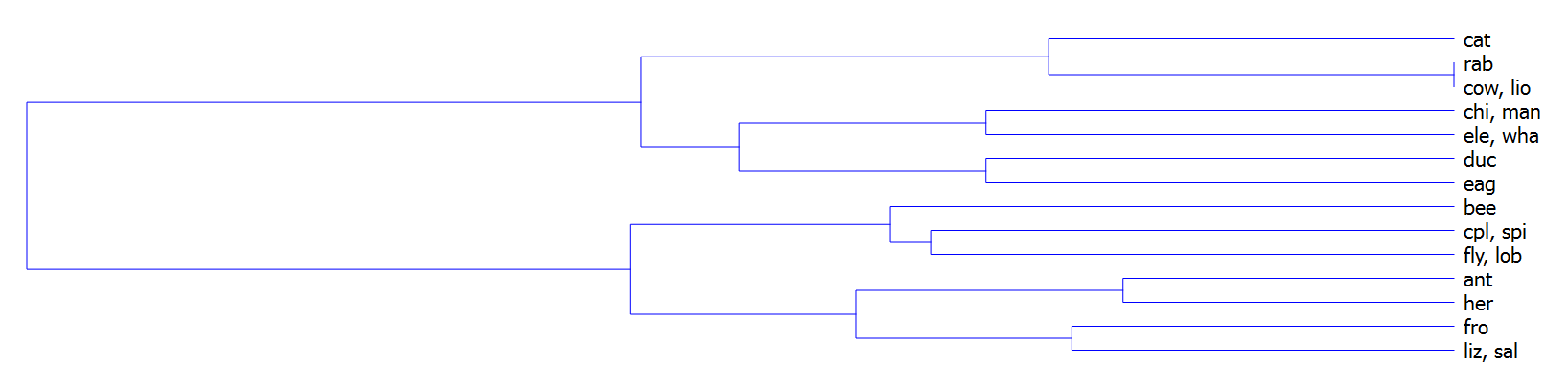
Orange:

Data from input file is imported using “file” component.

The data is continuized using “Continuize” component i.e. categorical values are turned continuous.

“Distances” widget is used to calculate the distance matrix using Euclidean metric.

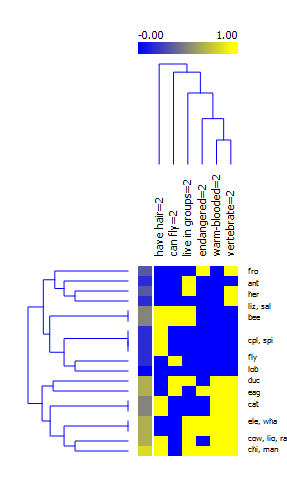
“Hierarchial Clustering” widget is used to cluster the data hierarchically where we can specify the linkage type and the number of clusters.

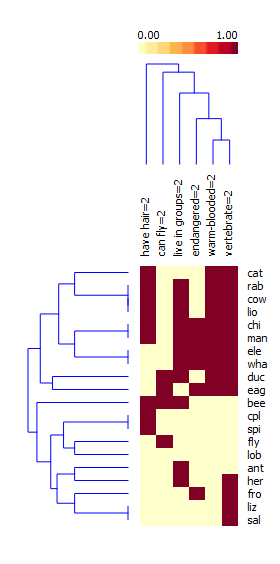


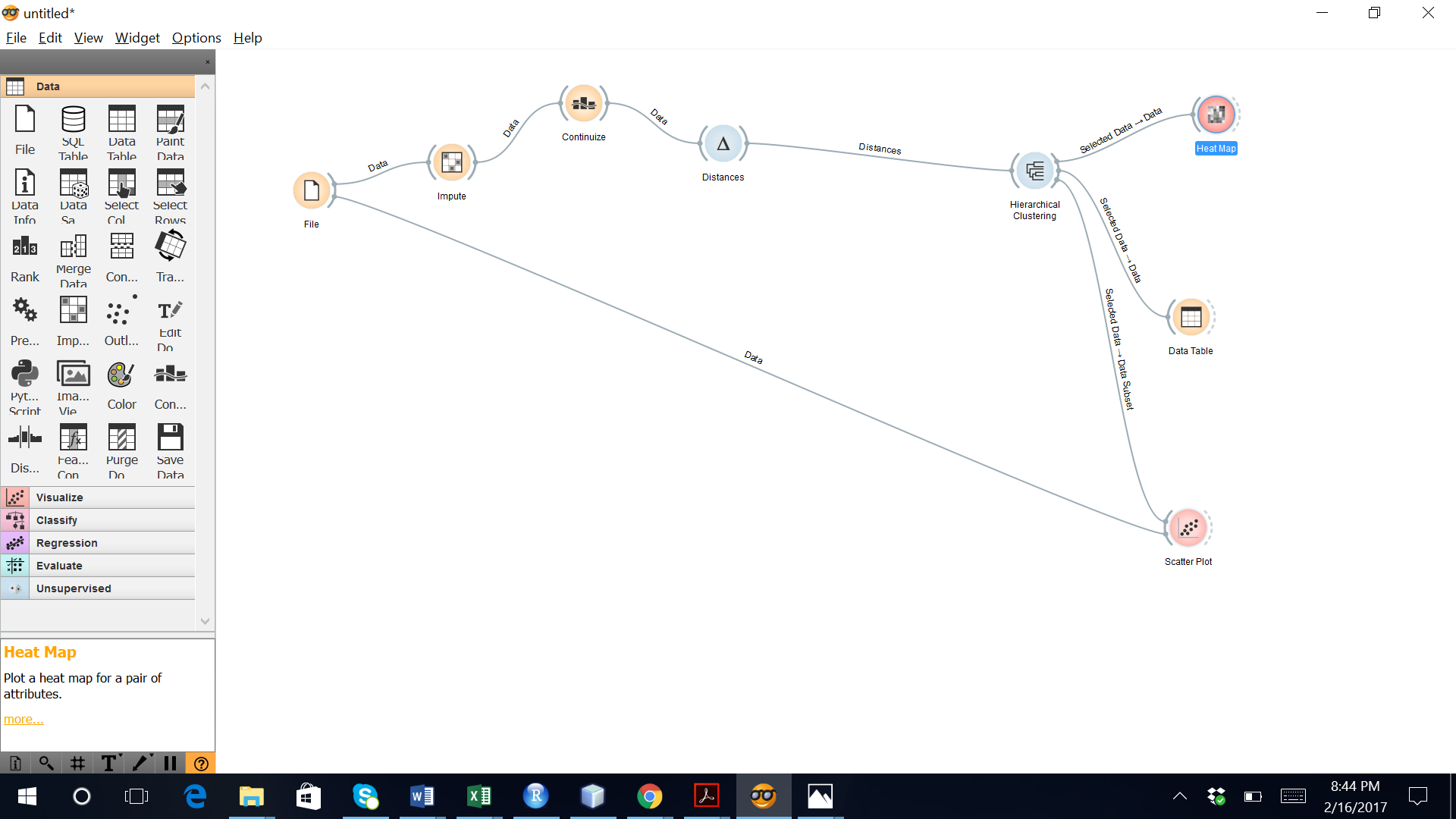


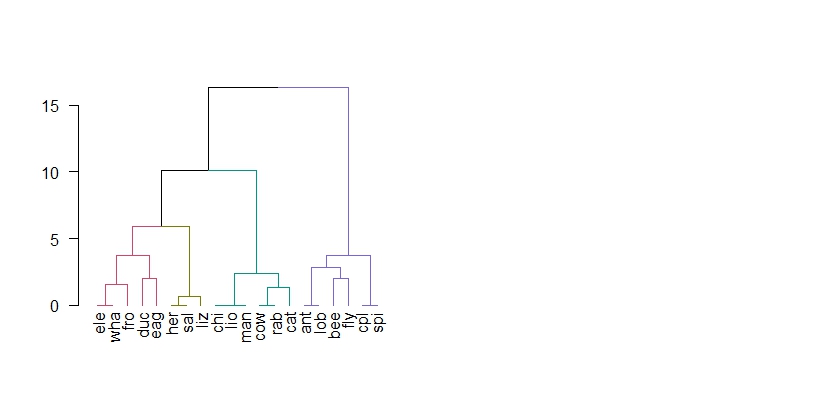
“Hierarchial Clustering” widget can be then connected to a “DataTable” component to view the cluster compnents in tabular format.

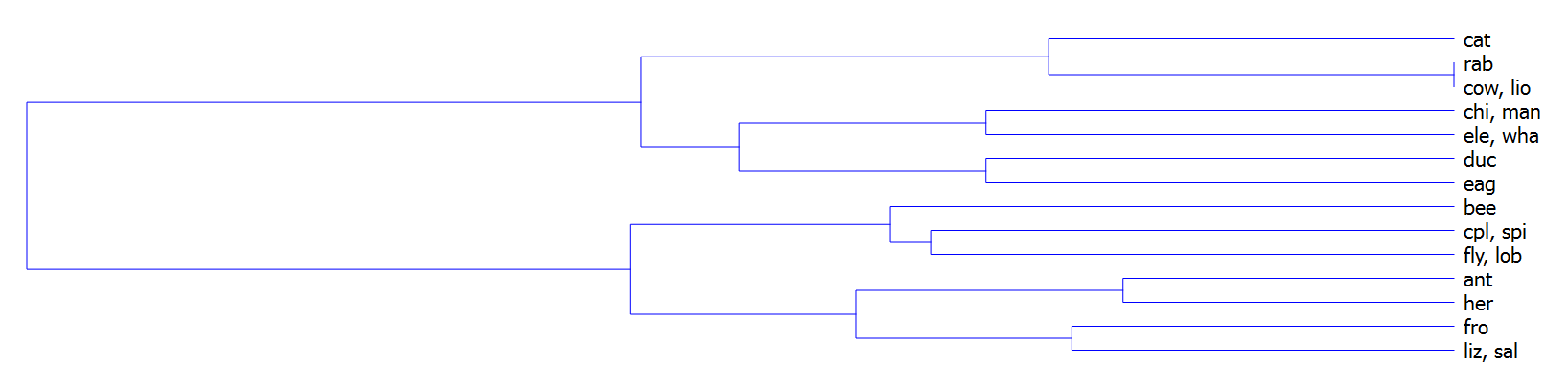
“Scatter Plot” is used to visualise the clusters where we can choose the axes and the color of the plot.











**Inference:** Hierarchial clustering on Play data using both R and Orange resulted the same clusters.

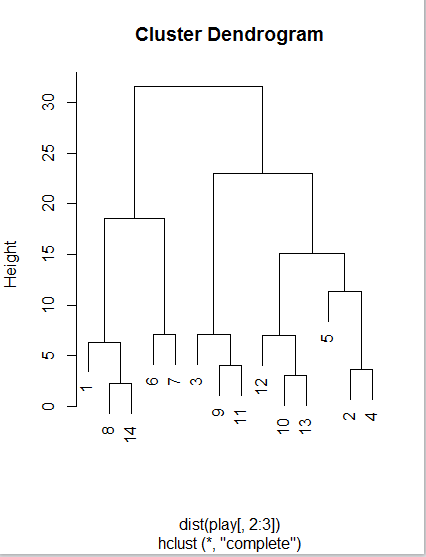
# Play Hierarchial Clustering:

R:

The play data is read from the csv file.

Hierarchial cluster analysis by complete linkage is done using “hcluster” function of R with distance matrix calculated on temperature and humidity of data.

Cluster is plotted using plot.



Hierarchial cluster is cut by specifying he number of clusters=2

Cluster is tabulated with the decision variable of data

clusterCut dont\_play play

1 3 2

2 2 7

Hierarchial cluster analysis by average/mean linkage is done using “hcluster” function of R with distance matrix calculated on temperature and humidity of data and further steps are repeated.

*clusters <- hclust(dist(play[, 2:3]), method = 'average')*

*plot(clusters)*



The cluster is cut by number of groups=2 using “ctree” and then the clusters are tabulated with the decision variable of the data.

*clusterCut <- cutree(clusters, 2)*

*table(clusterCut, play$Decision)*

clusterCut dont\_play play

1 3 4

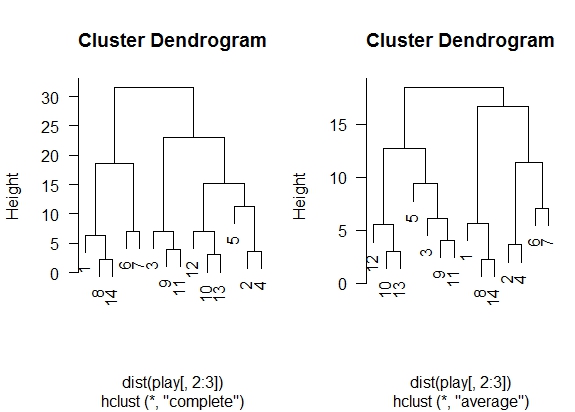
2 2 5

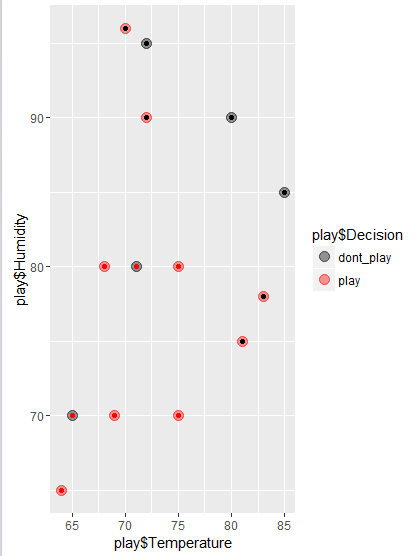
A plot on play data with aesthetics as Temperature and Humidity and color as Decision of data is created using “ggplot” and is converted to scatter plot using “geom\_point” and manually colors are selected for plot using “scale\_color\_manual”.

*ggplot(play, aes(play$Temperature, play$Humidity, color = play$Decision)) +*

*geom\_point(alpha = 0.4, size = 3.5) + geom\_point(col = clusterCut) +*

*scale\_color\_manual(values = c('black', 'red', 'green'))*

**



> play <- readWorksheetFromFile("D:/Subjects/ADS/assignment week5/play.xlsx", sheet = 1)

> clusters <- hclust(dist(play[, 2:3]))

> plot(clusters)

> clusterCut <- cutree(clusters, 2)

> table(clusterCut, play$Decision)

clusterCut dont\_play play

1 3 2

2 2 7

> clusters <- hclust(dist(play[, 2:3]), method = 'average')

> plot(clusters)

> clusterCut <- cutree(clusters, 2)

> table(clusterCut, play$Decision)

clusterCut dont\_play play

1 3 4

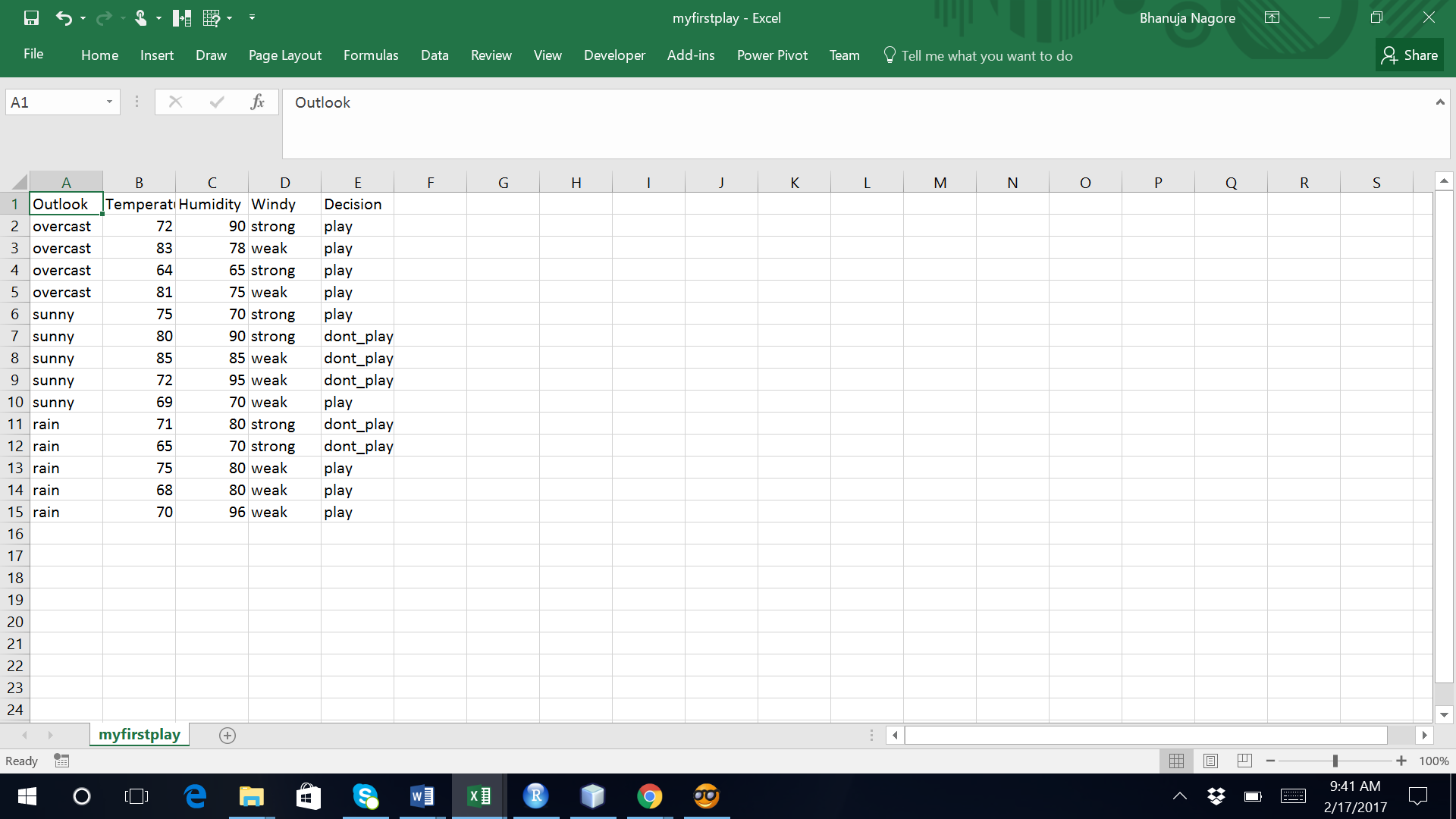
2 2 5

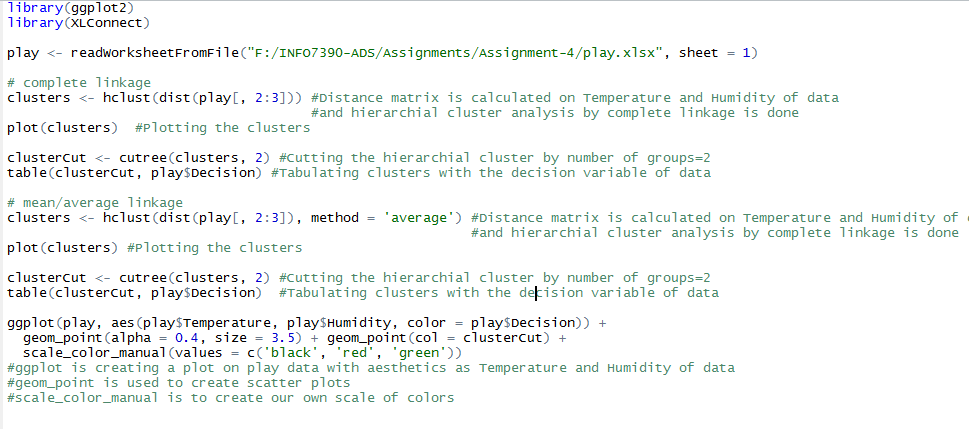
> ggplot(play, aes(play$Temperature, play$Humidity, color = play$Decision)) +

+ geom\_point(alpha = 0.4, size = 3.5) + geom\_point(col = clusterCut) +

+ scale\_color\_manual(values = c('black', 'red', 'green'))

> write.csv(play, file = "myfirstplay.csv", row.names = FALSE)





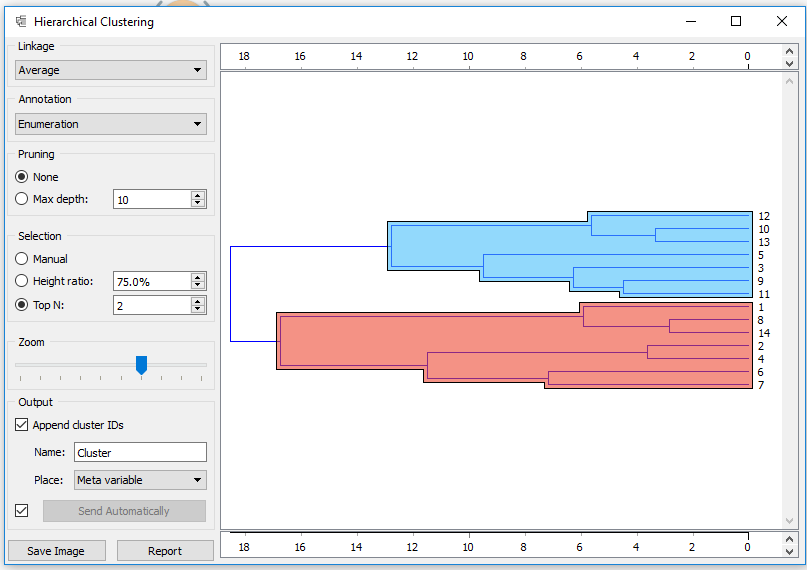
Orange:

Data from input file is imported using “file” component.

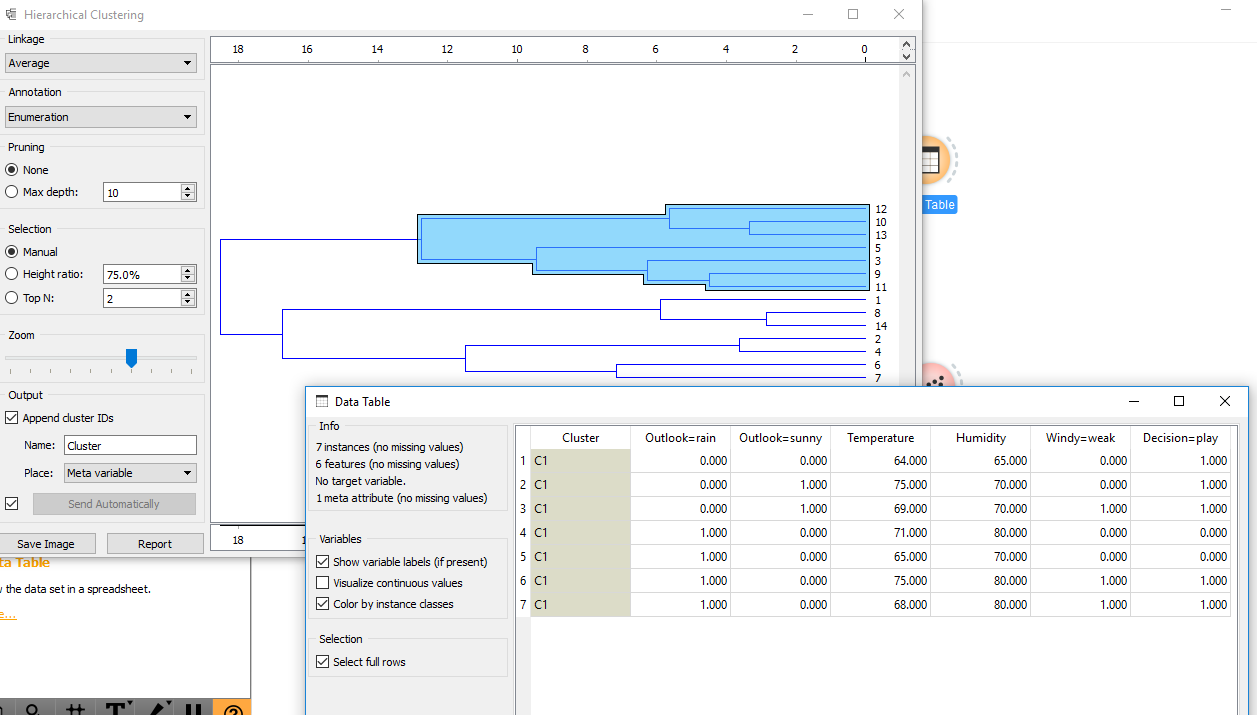
The data is continuized using “Continuize” component i.e. categorical values are turned continuous.

“Distances” widget is used to calculate the distance matrix using Euclidean metric.

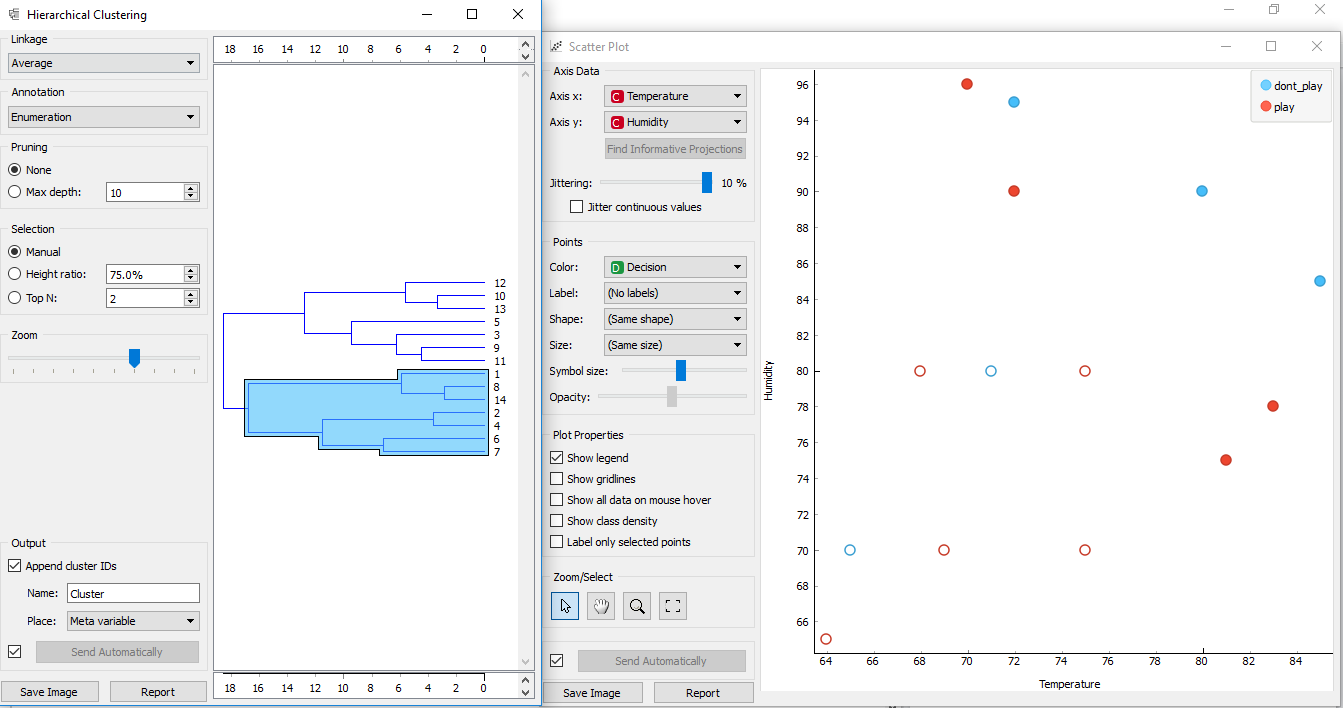
“Hierarchial Clustering” widget is used to cluster the data hierarchically where we can specify the linkage type and the number of clusters.

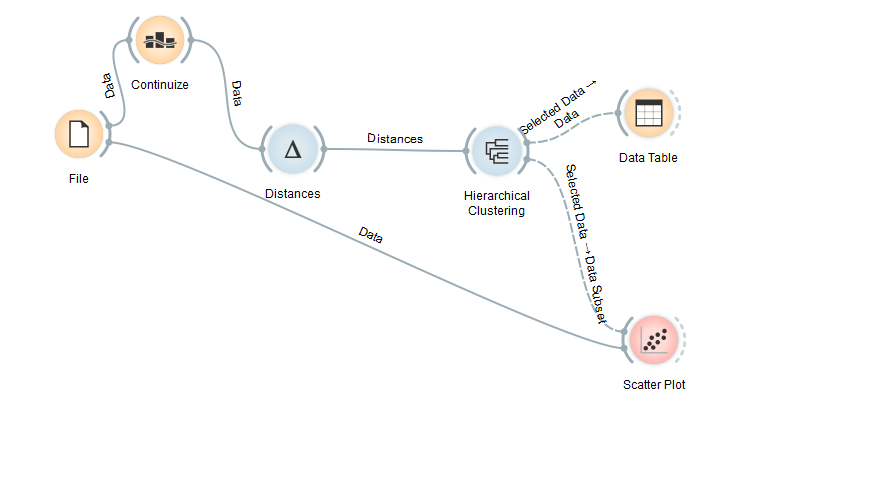


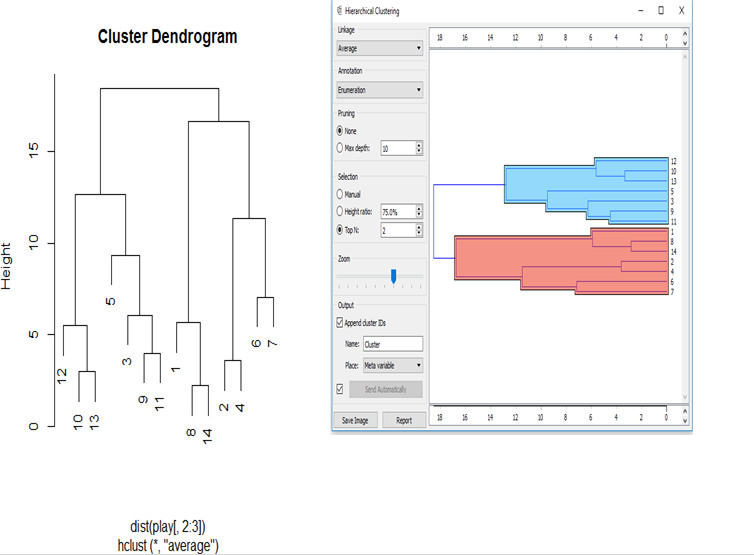
“Hierarchial Clustering” widget can be then connected to a “DataTable” component to view the cluster compnents in tabular format.



“Scatter Plot” is used to visualise the clusters where we can choose the axes and the color of the plot.







**Inference:** Hierarchial clustering on Play data using both R and Orange resulted the same clusters.

# Iris Hierarchial Clustering:

R:

A user defined function by name “clusterPlot” is defined which takes an input argument ‘type’ which is used to create the hierarchial cluster and plot the same.

The function calculates the distance matrix using Petal.Length and Petal.Width.

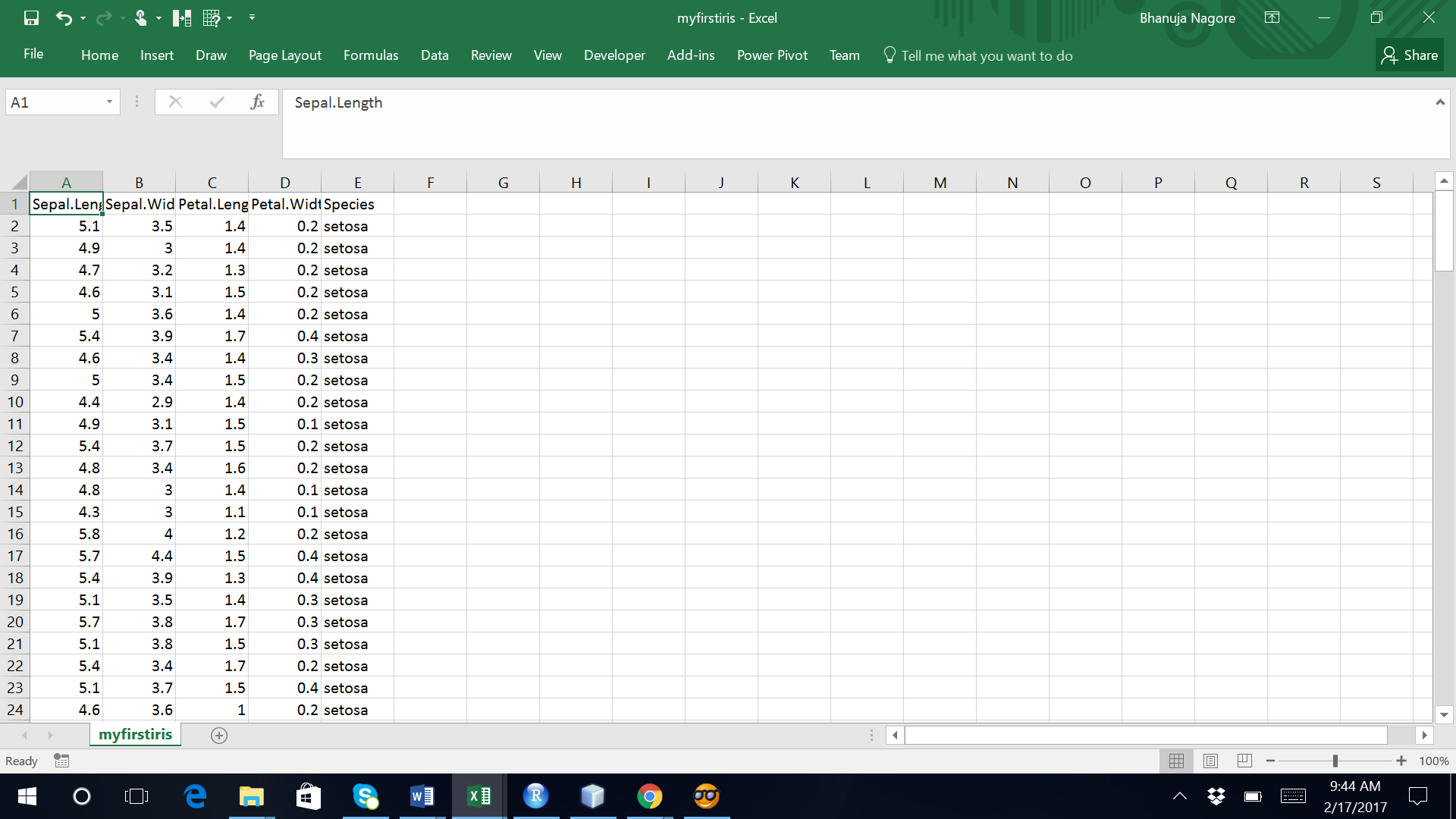
And then Hierarchial cluster has been created using the linkage type specified by the user input for ‘type’ argument.

The cluster will be plotted using plot.

The hierarchial cluster is cut for 3 clusters and the clusters are tabulated with Species variable of data.

A plot on Iris data with aesthetics as Petal.Length and Petal.Width and color as Species of data is created using “ggplot” and is converted to scatter plot using “geom\_point” and manually colors are selected for plot using “scale\_color\_manual”.

Then the function “clusterPlot” can be called by giving the input argument for linkage type.



Hierarchial cluster is cut by specifying he number of clusters=3

Cluster is tabulated with the decision variable of data

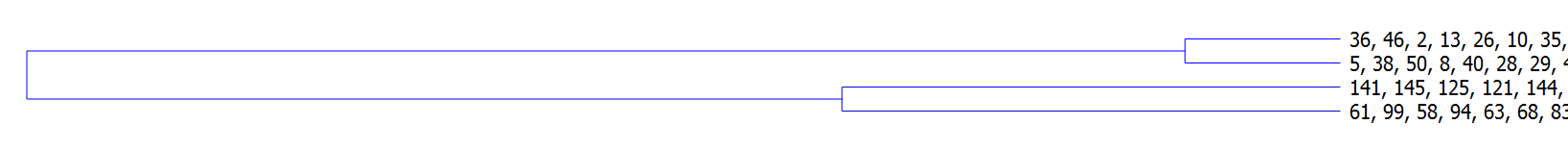
Hierarchial cluster analysis by average/mean linkage/ward/complete is done using “hcluster” function of R with distance matrix calculated on widht and height of data and further steps are repeated.

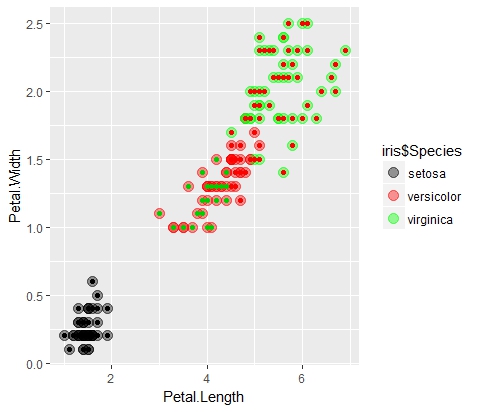
clusterPlot('complete')

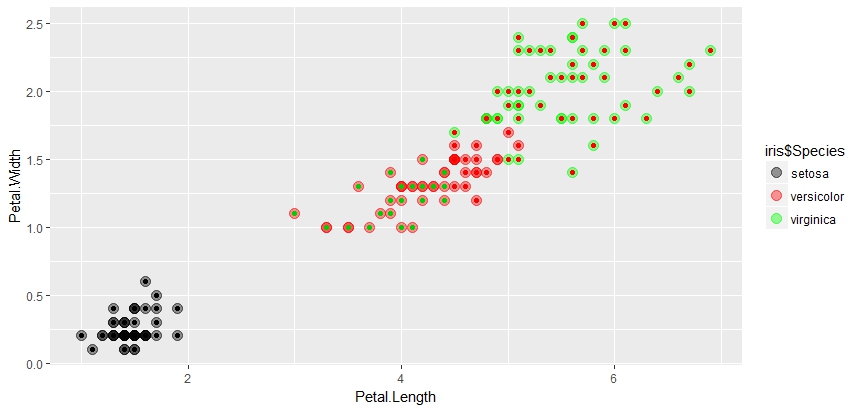
clusterPlot('average')

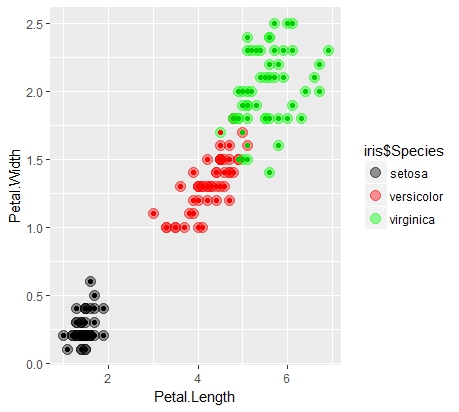
clusterPlot('single')

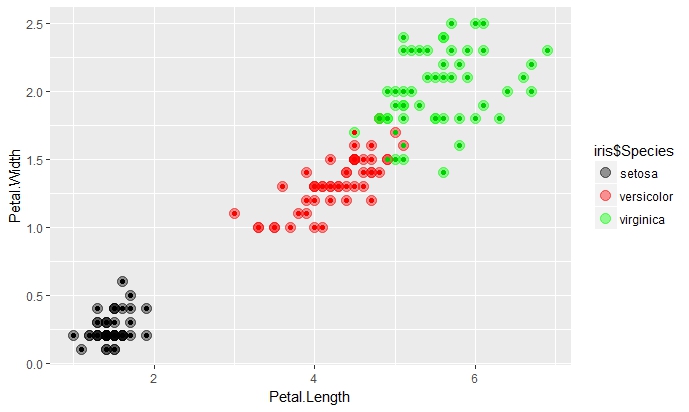
clusterPlot('ward.D')

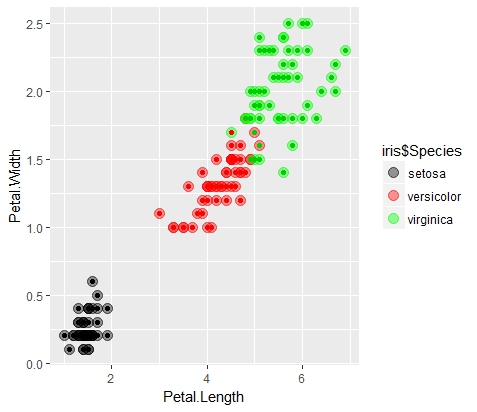


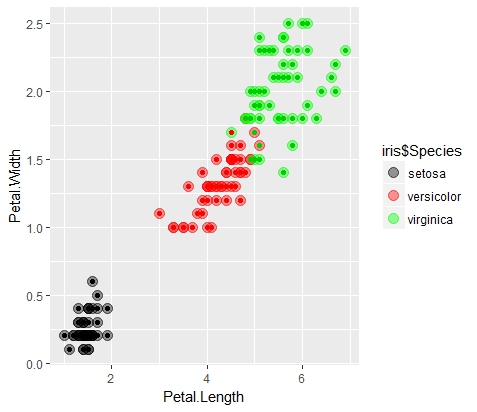


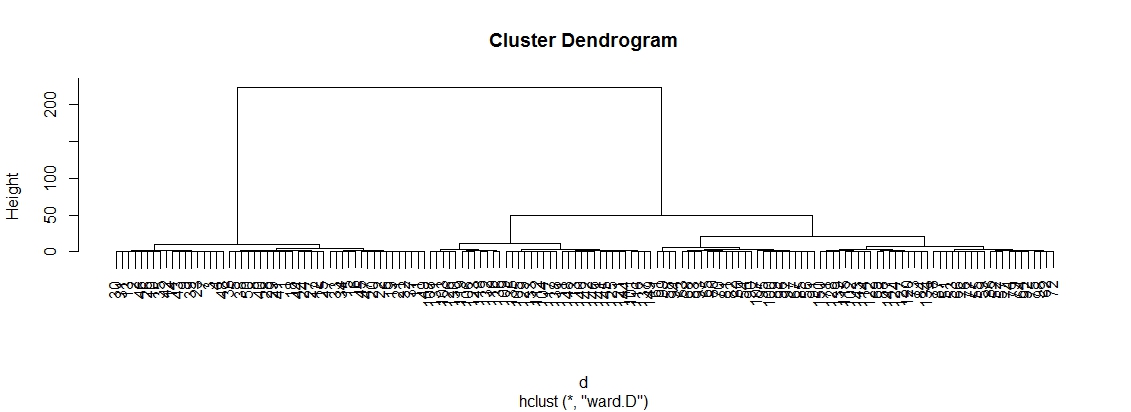








**

**

The cluster is cut by number of groups=3 using “ctree” and then the clusters are tabulated with the decision variable of the data.

*clusterCut <- cutree(clusters, 3)*

A plot on play data with aesthetics as width and height and species as Decision of data is created using “ggplot” and is converted to scatter plot using “geom\_point” and manually colors are selected for plot using “scale\_color\_manual”.

*library(ggplot2)*

*clusterPlot <- function(type) {*

*clusters <- hclust(dist(iris[, 3:4]), method = type)*

*plot(clusters)*

*clusterCut <- cutree(clusters, 3)*

*show(table(clusterCut, iris$Species)) # show required, else will not print*

*ggplot(iris, aes(Petal.Length, Petal.Width, color = iris$Species)) +*

*geom\_point(alpha = 0.4, size = 3.5) + geom\_point(col = clusterCut) +*

*scale\_color\_manual(values = c('black', 'red', 'green'))*

*}*

*clusterPlot('complete')*

*clusterPlot('average')*

*clusterPlot('single')*

*clusterPlot('ward.D')*

*write.csv(iris, file = "myfirstiris.csv", row.names = FALSE)*

> library(ggplot2)

> clusterPlot <- function(type) {

+ clusters <- hclust(dist(iris[, 3:4]), method = type)

+ plot(clusters)

+ clusterCut <- cutree(clusters, 3)

+ show(table(clusterCut, iris$Species)) # show required, else will not print

+ ggplot(iris, aes(Petal.Length, Petal.Width, color = iris$Species)) +

+ geom\_point(alpha = 0.4, size = 3.5) + geom\_point(col = clusterCut) +

+ scale\_color\_manual(values = c('black', 'red', 'green'))

+ }

> clusterPlot('complete')

clusterCut setosa versicolor virginica

1 50 0 0

2 0 21 50

3 0 29 0

> clusterPlot('average')

clusterCut setosa versicolor virginica

1 50 0 0

2 0 45 1

3 0 5 49

> clusterPlot('single')

clusterCut setosa versicolor virginica

1 50 0 0

2 0 49 50

3 0 1 0

> clusterPlot('ward.D')

clusterCut setosa versicolor virginica

1 50 0 0

2 0 45 1

3 0 5 49

> clusterPlot('ward.D')

clusterCut setosa versicolor virginica

1 50 0 0

2 0 45 1

3 0 5 49

> clusterPlot('average')

clusterCut setosa versicolor virginica

1 50 0 0

2 0 45 1

3 0 5 49

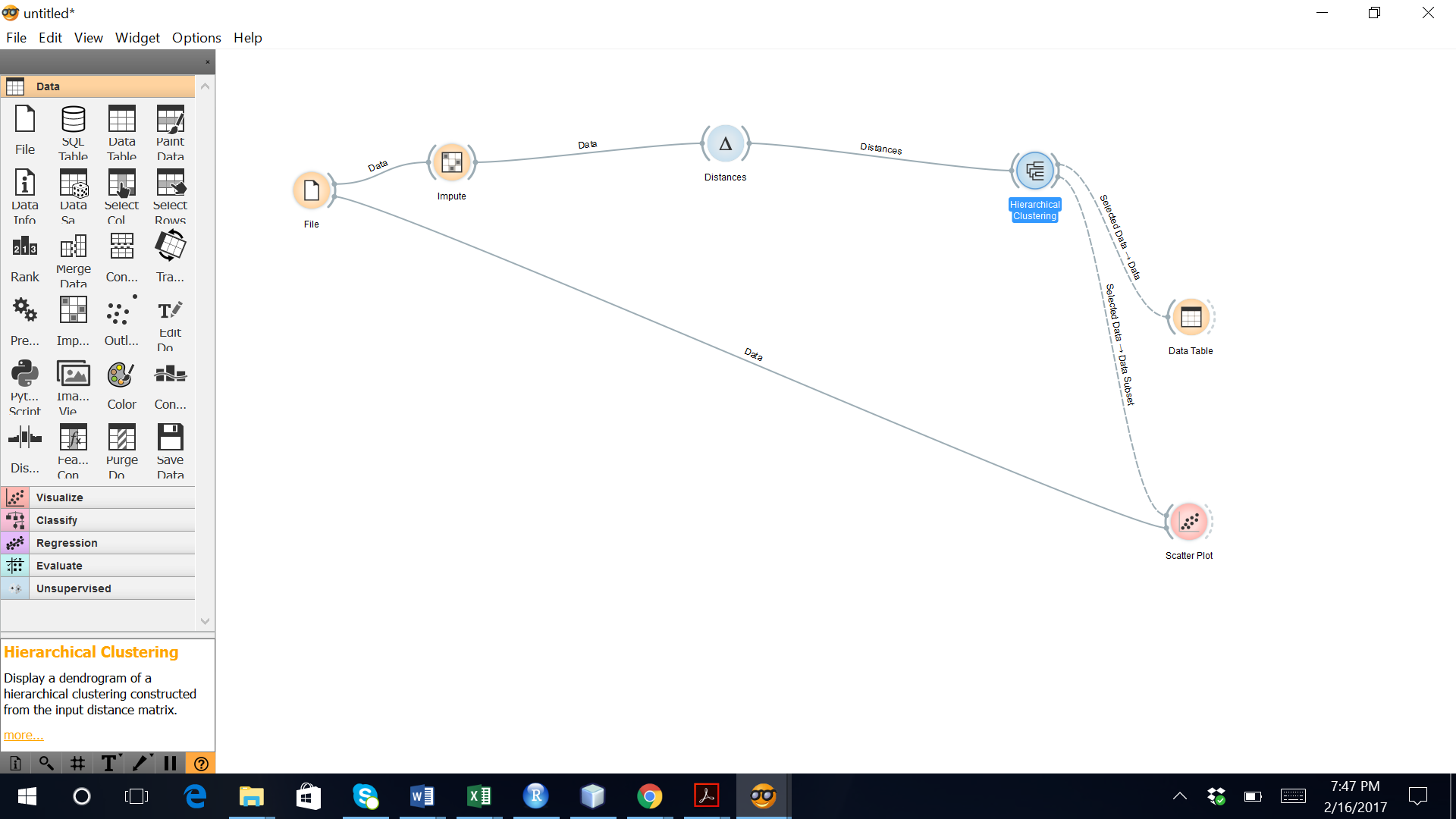
Orange:

Data from input file is imported using “file” component.

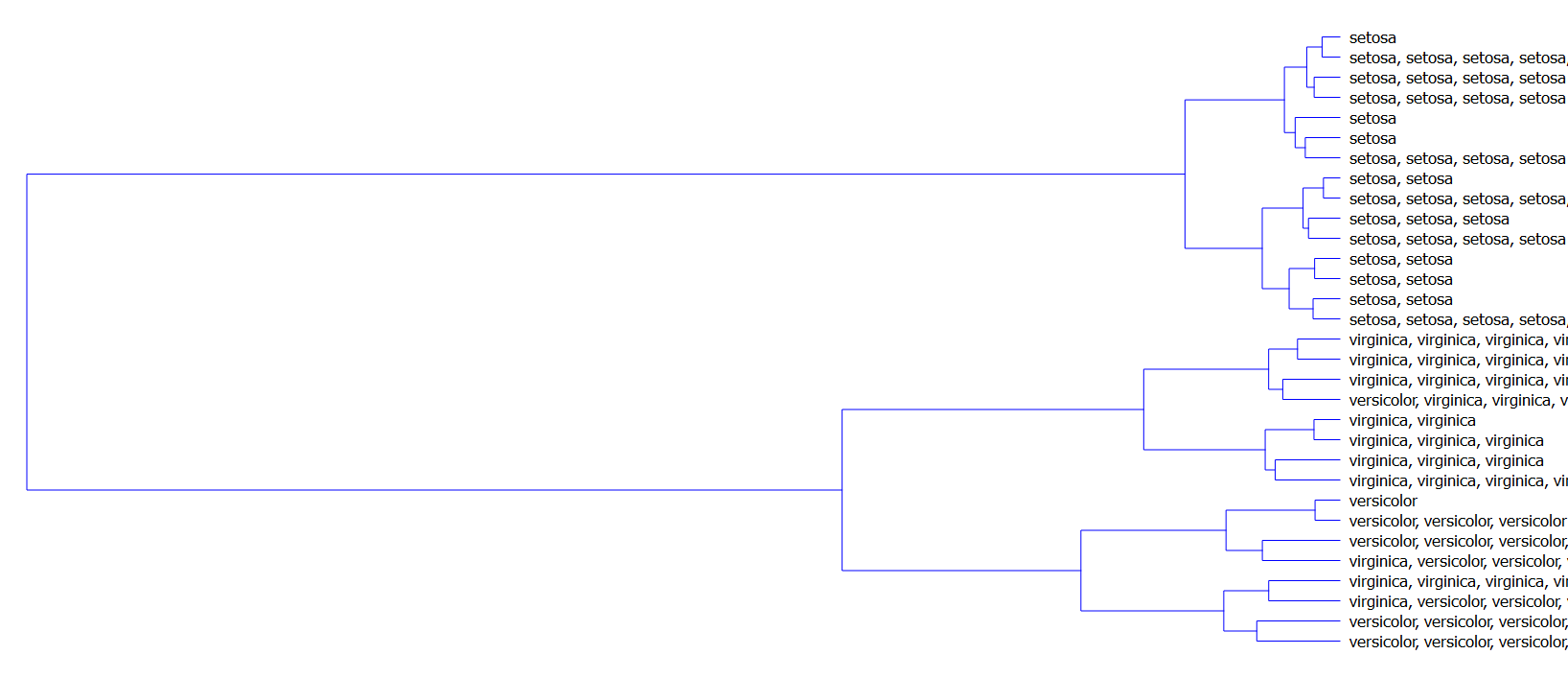
The data is continuized using “Continuize” component i.e. categorical values are turned continuous.

“Distances” widget is used to calculate the distance matrix using Euclidean metric.

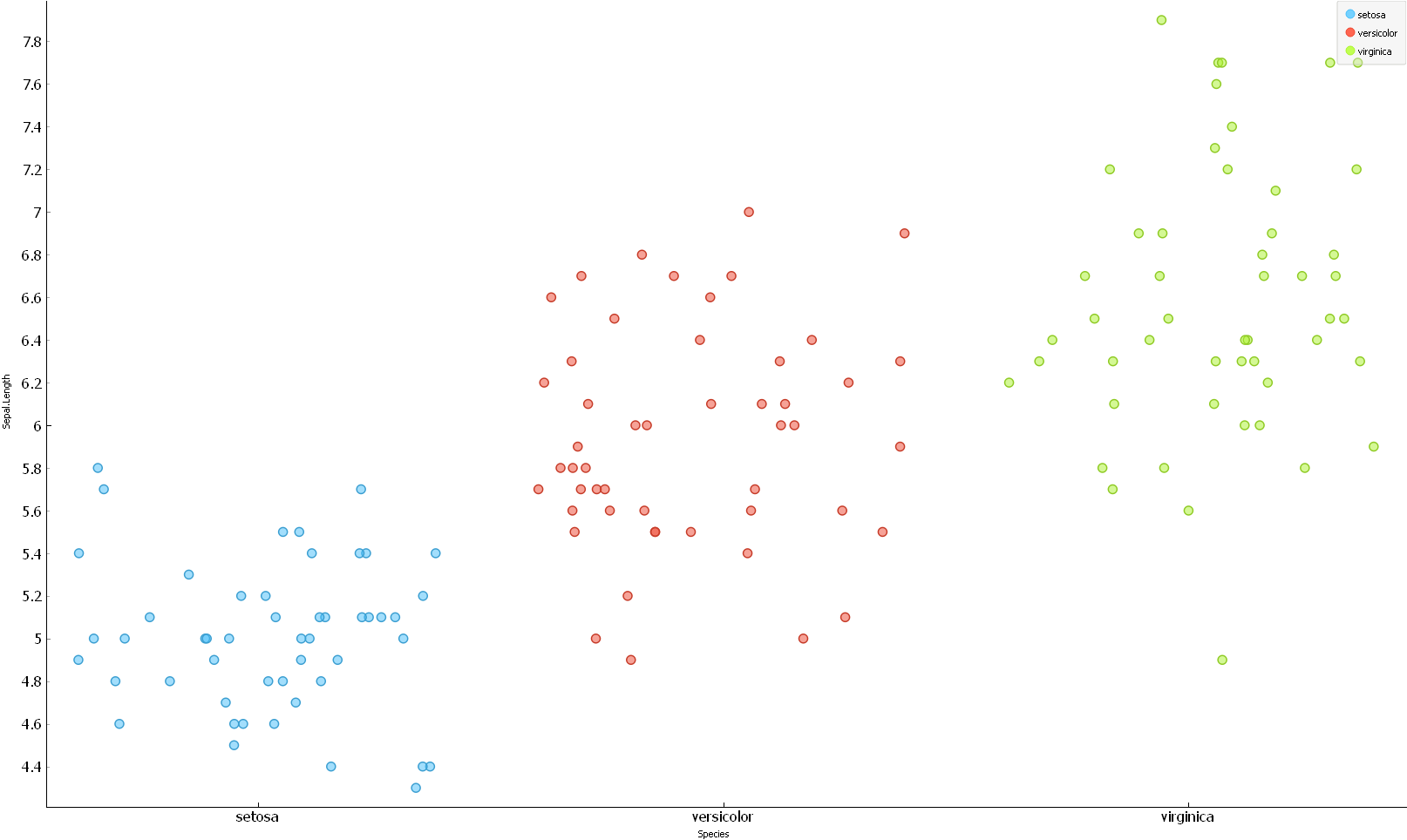
“Hierarchial Clustering” widget is used to cluster the data hierarchically where we can specify the linkage type and the number of clusters.

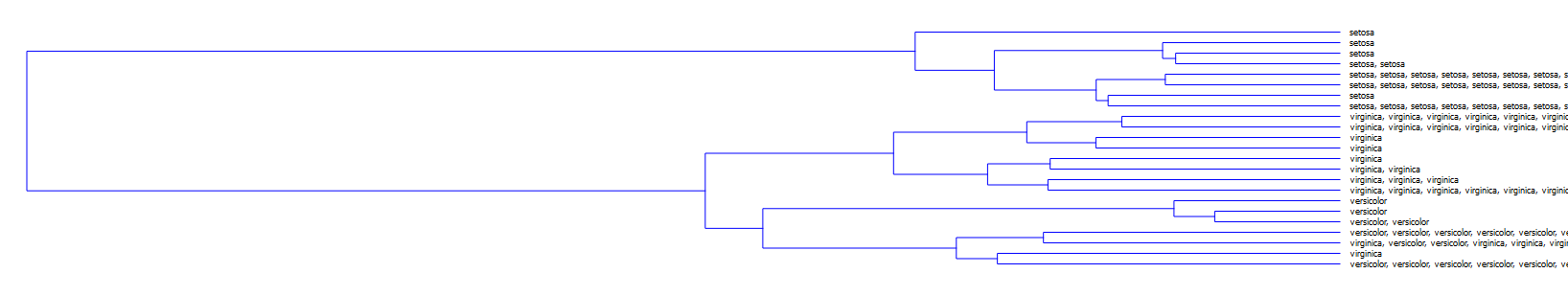


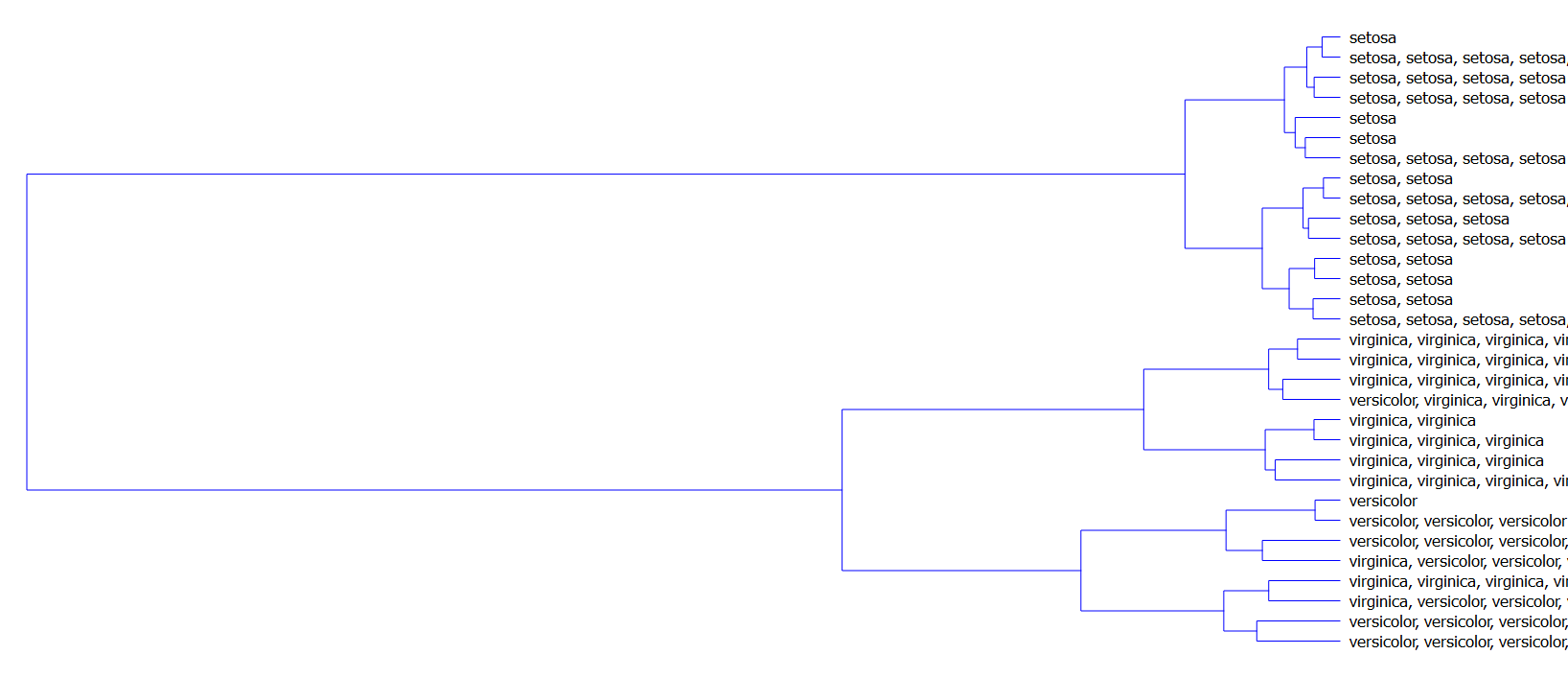
“Hierarchial Clustering” widget can be then connected to a “DataTable” component to view the cluster compnents in tabular format.



“Scatter Plot” is used to visualise the clusters where we can choose the axes and the color of the plot.







**Inference:** Hierarchial clustering on Play data using both R and Orange resulted the same clusters.

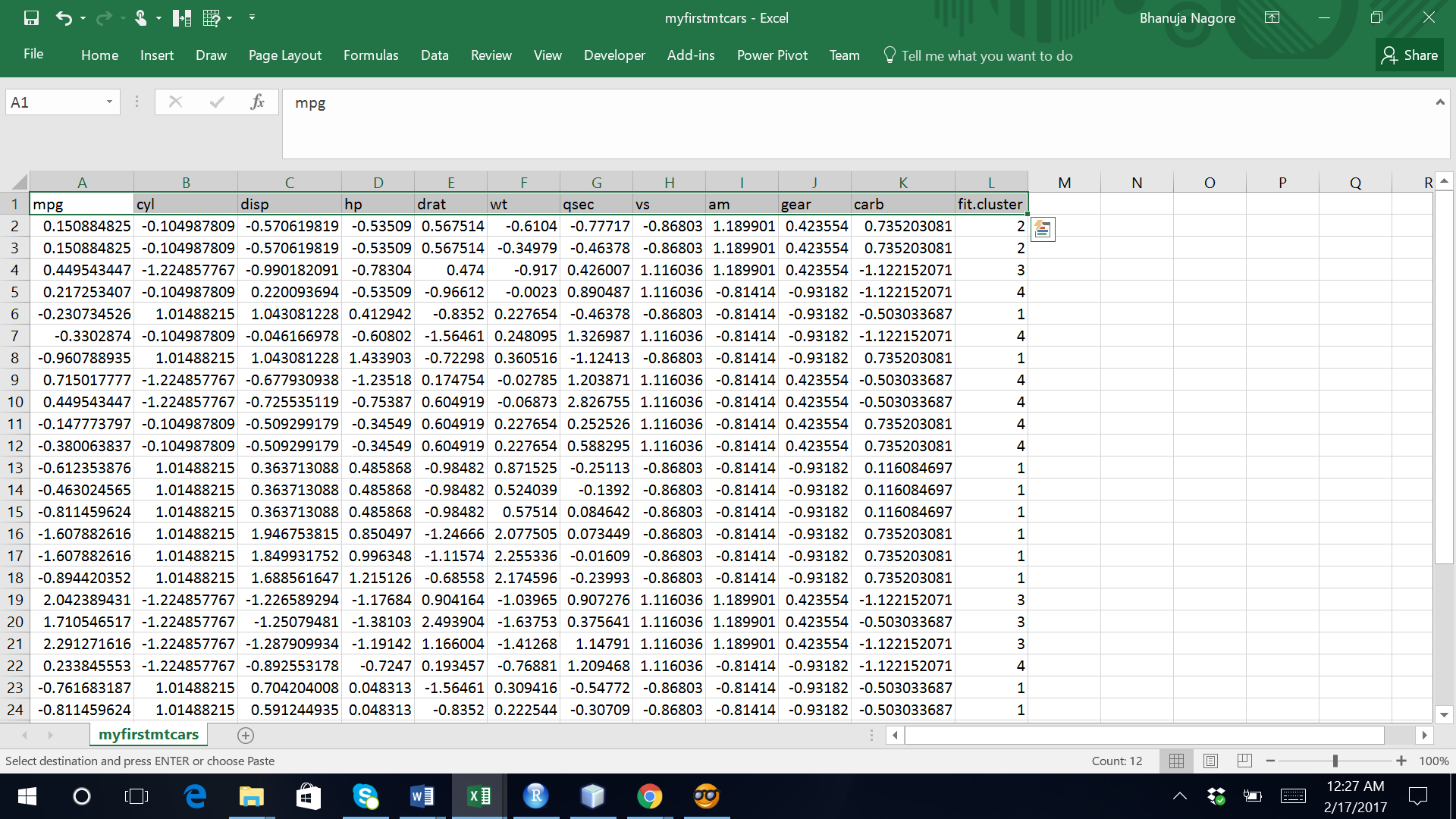
# Kmeans + Hierarchial clustering:

R:

The play data is read

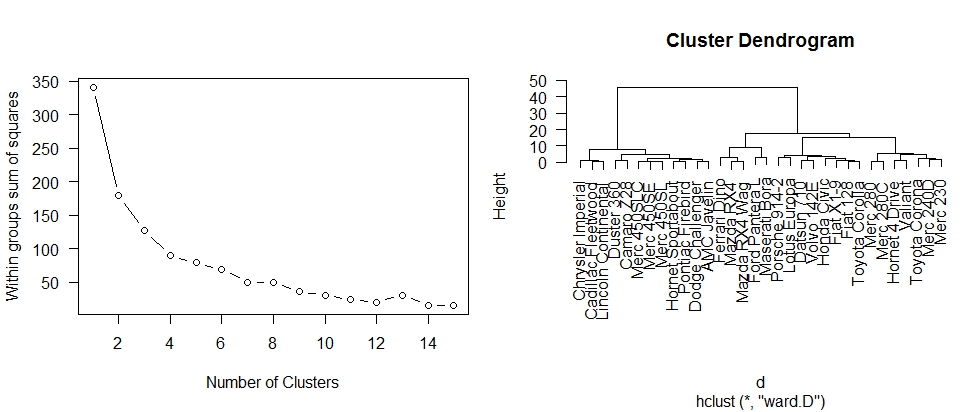
*data(mtcars)*

*mydata <- mtcars*



Cluster is plotted using plot.

*plot(clusters)*

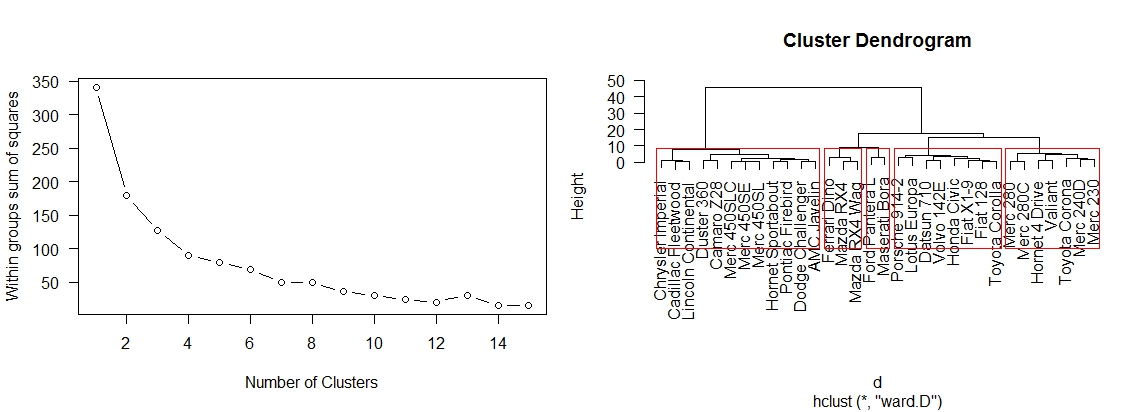
**

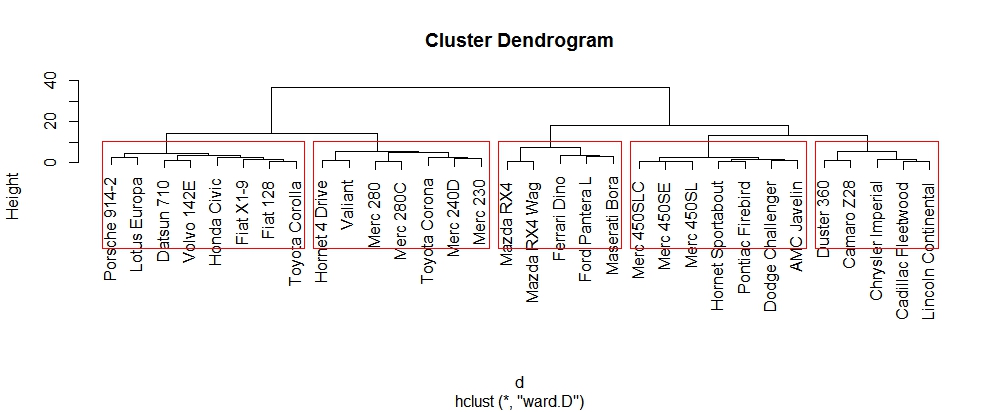
Hierarchial cluster is cut by specifying he number of groups/clusters=5

Cluster is tabulated with the decision variable of data

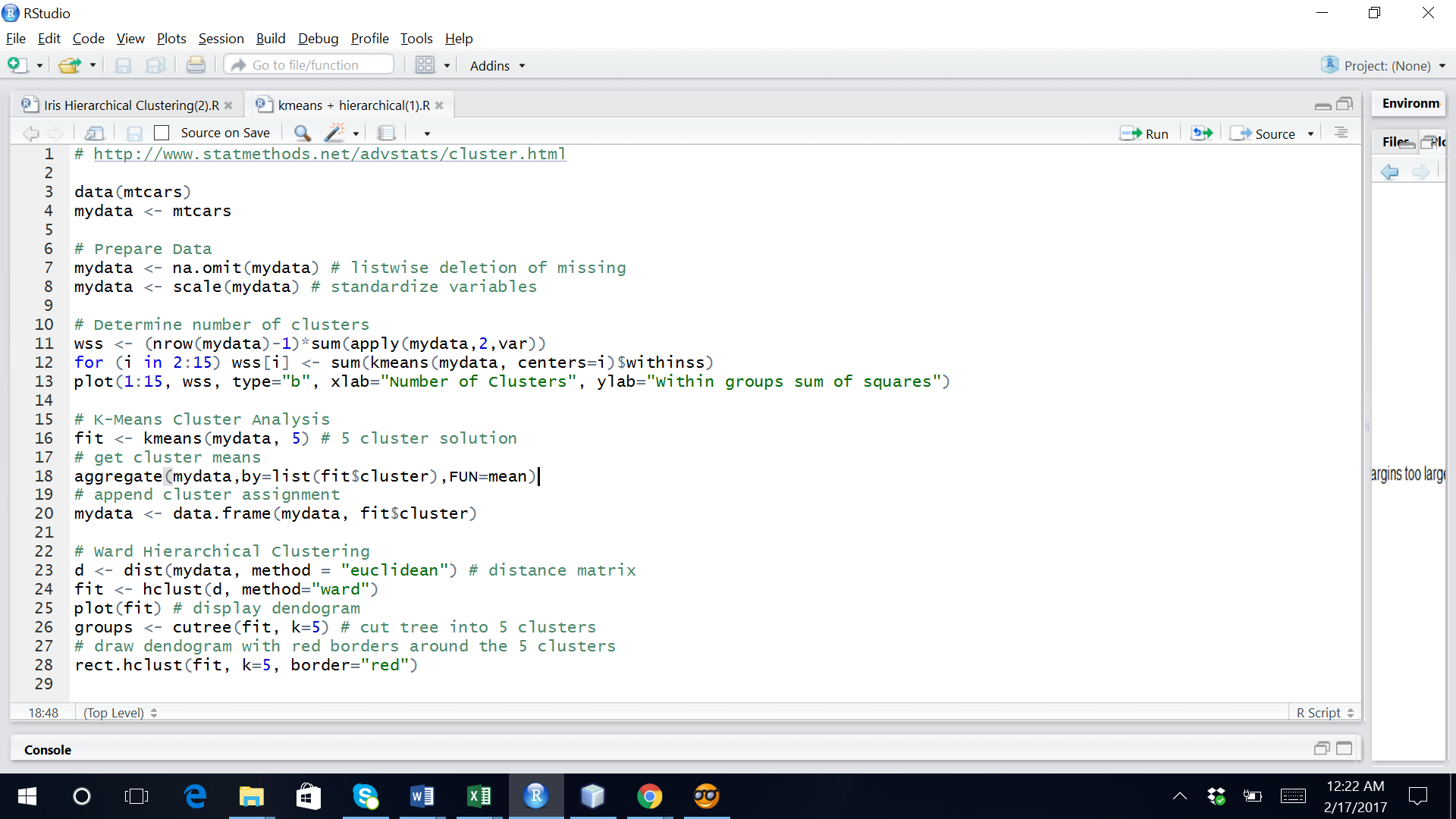
*fit$cluster*

Hierarchial cluster analysis by ward method is done using “hcluster” function of R with distance matrix

**

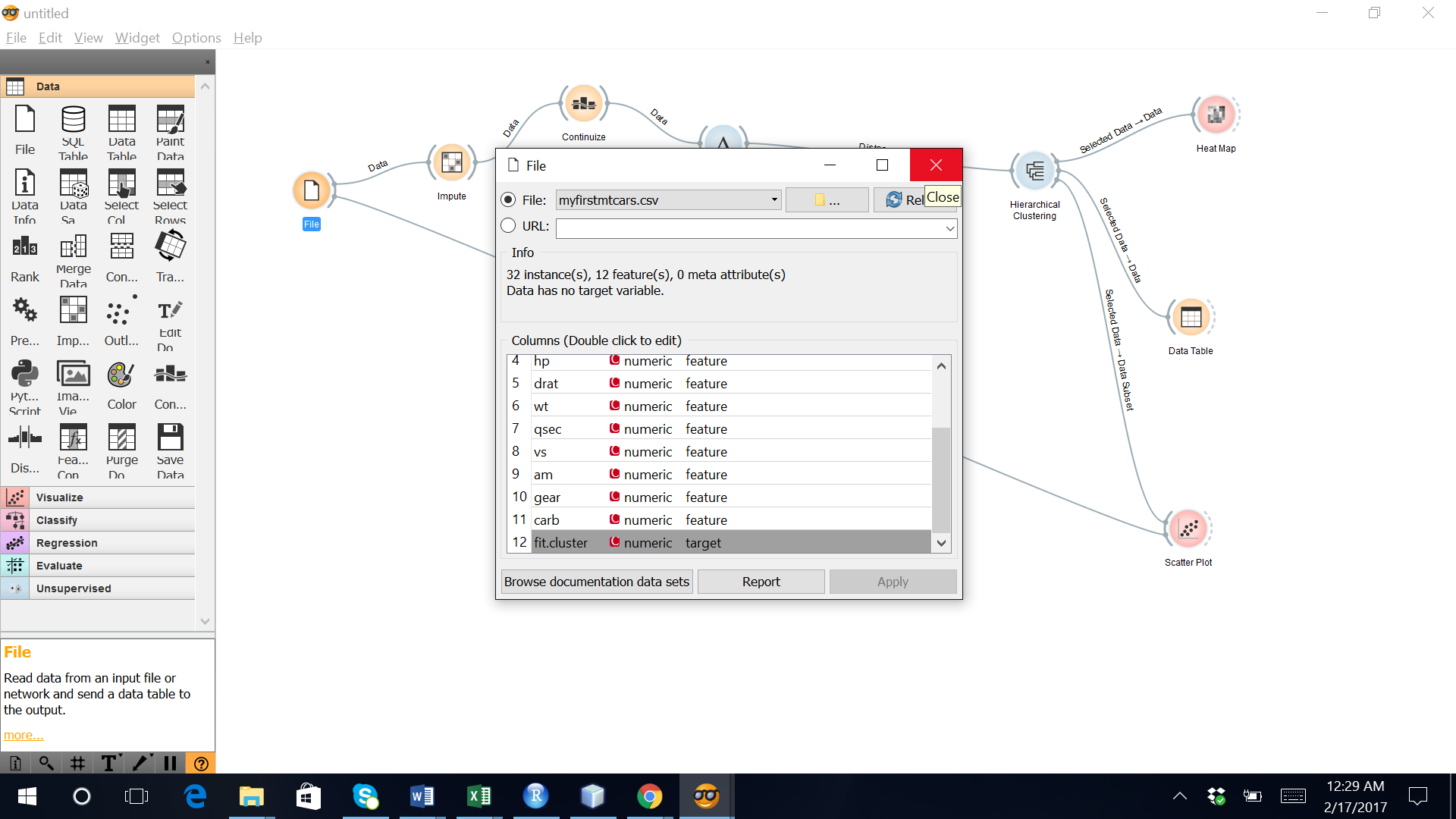
**

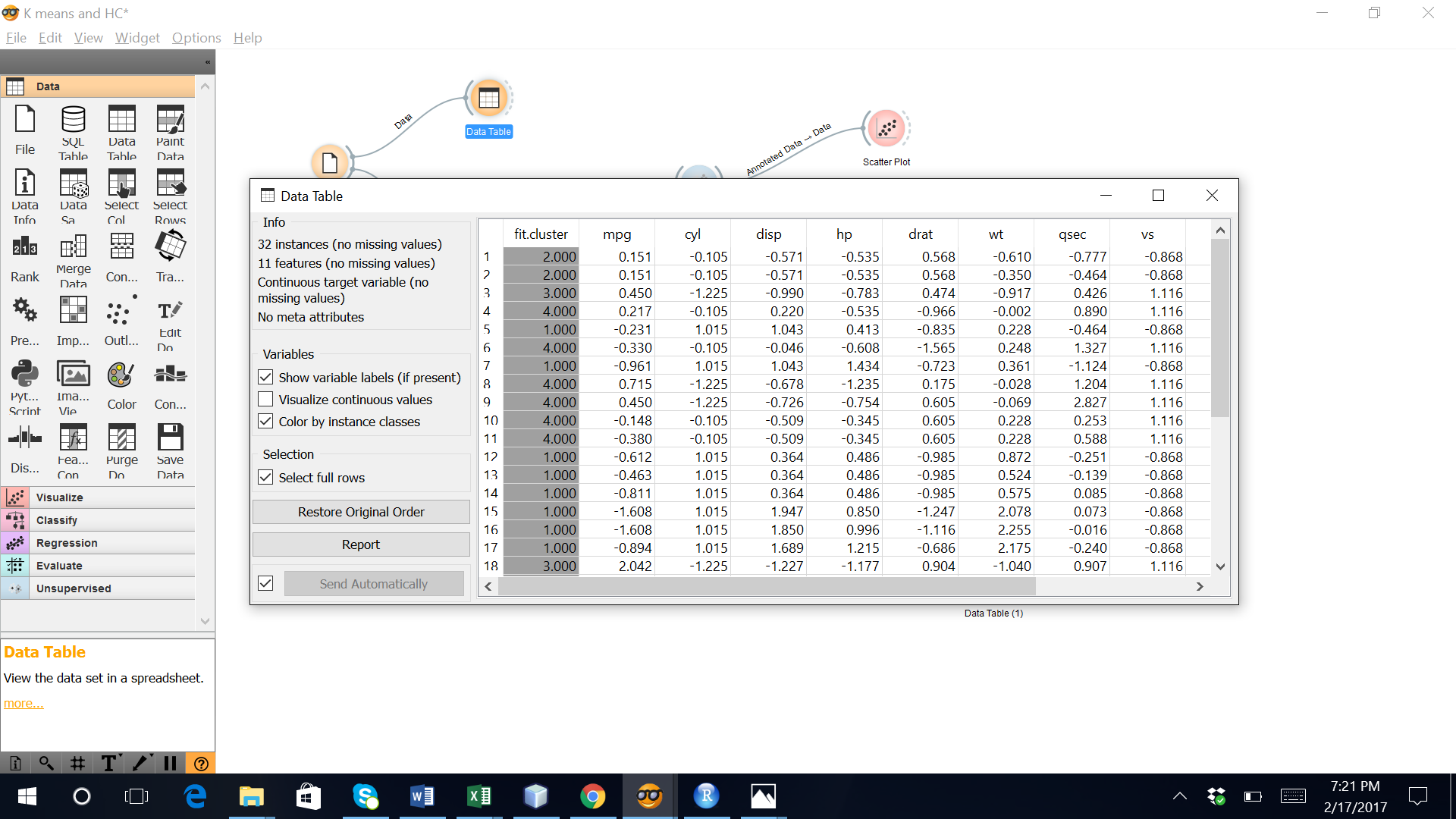
|  |
| --- |
| > data(mtcars)  > mydata <- mtcars  > # Prepare Data  > mydata <- na.omit(mydata) # listwise deletion of missing  > mydata <- scale(mydata) # standardize variables  > wss <- (nrow(mydata)-1)\*sum(apply(mydata,2,var))  > for (i in 2:15) wss[i] <- sum(kmeans(mydata, centers=i)$withinss)  > plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")  > fit <- kmeans(mydata, 5)  > aggregate(mydata,by=list(fit$cluster),FUN=mean)  Group.1 mpg cyl disp hp drat wt  1 1 -0.8363478 1.0148821 1.0238513 0.6924910 -0.88974768 0.90635862  2 2 0.3043622 -0.3849553 -0.6809566 -0.3673636 0.68440631 -0.62956238  3 3 1.3739630 -1.2248578 -1.1370289 -0.9643131 1.03241235 -1.21514154  4 4 0.1082193 -0.5849321 -0.4486701 -0.6496905 -0.04967936 -0.02346989  5 5 -0.7782753 1.0148821 0.7687521 2.2287939 0.53010805 0.15611307  qsec vs am gear carb  1 -0.3952280 -0.8680278 -0.8141431 -0.9318192 0.1676779  2 -0.7995497 -0.8680278 1.1899014 1.1012409 0.7352031  3 0.4763722 1.1160357 1.1899014 0.6171790 -0.8568156  4 1.1854841 1.1160357 -0.8141431 -0.1573201 -0.4145882  5 -1.8460295 -0.8680278 1.1899014 1.7789276 1.9734398  > mydata <- data.frame(mydata, fit$cluster)  > d <- dist(mydata, method = "euclidean")  > fit <- hclust(d, method="ward")  > plot(fit)  > groups <- cutree(fit, k=5) # cut tree into 5 clusters  > rect.hclust(fit, k=5, border="red") |
|  |
| |  | | --- | |  | |



Orange:

Data from input file is imported using “file” component.

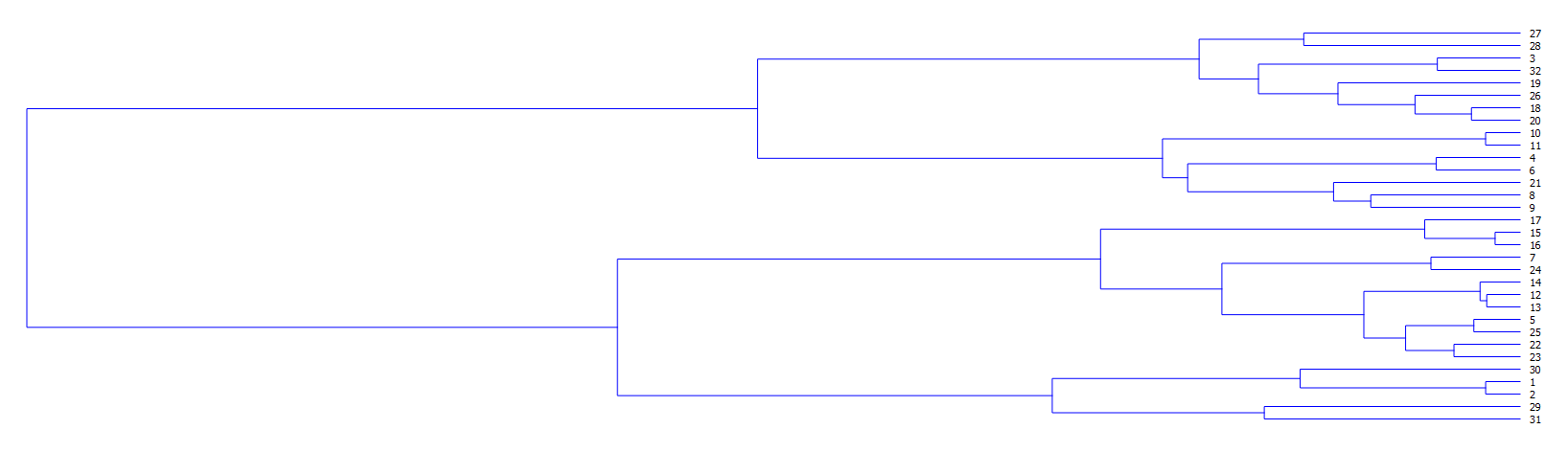




The data is continuized using “Continuize” component i.e. continuous values are used.

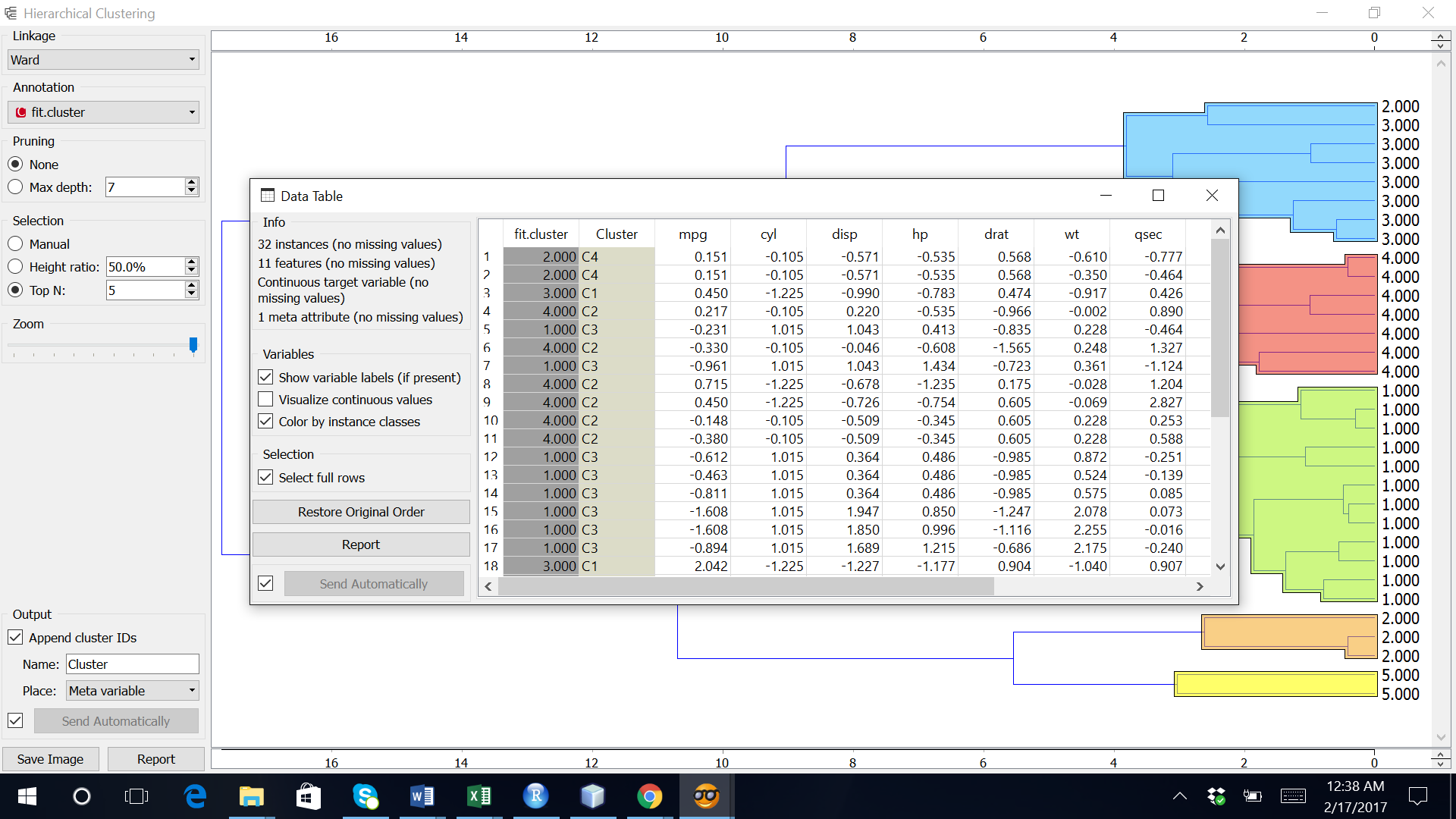
“Distances” widget is used to calculate the distance matrix using Euclidean metric.

“Hierarchial Clustering” widget is used to cluster the data hierarchically where we can specify the linkage type and the number of clusters is 5.



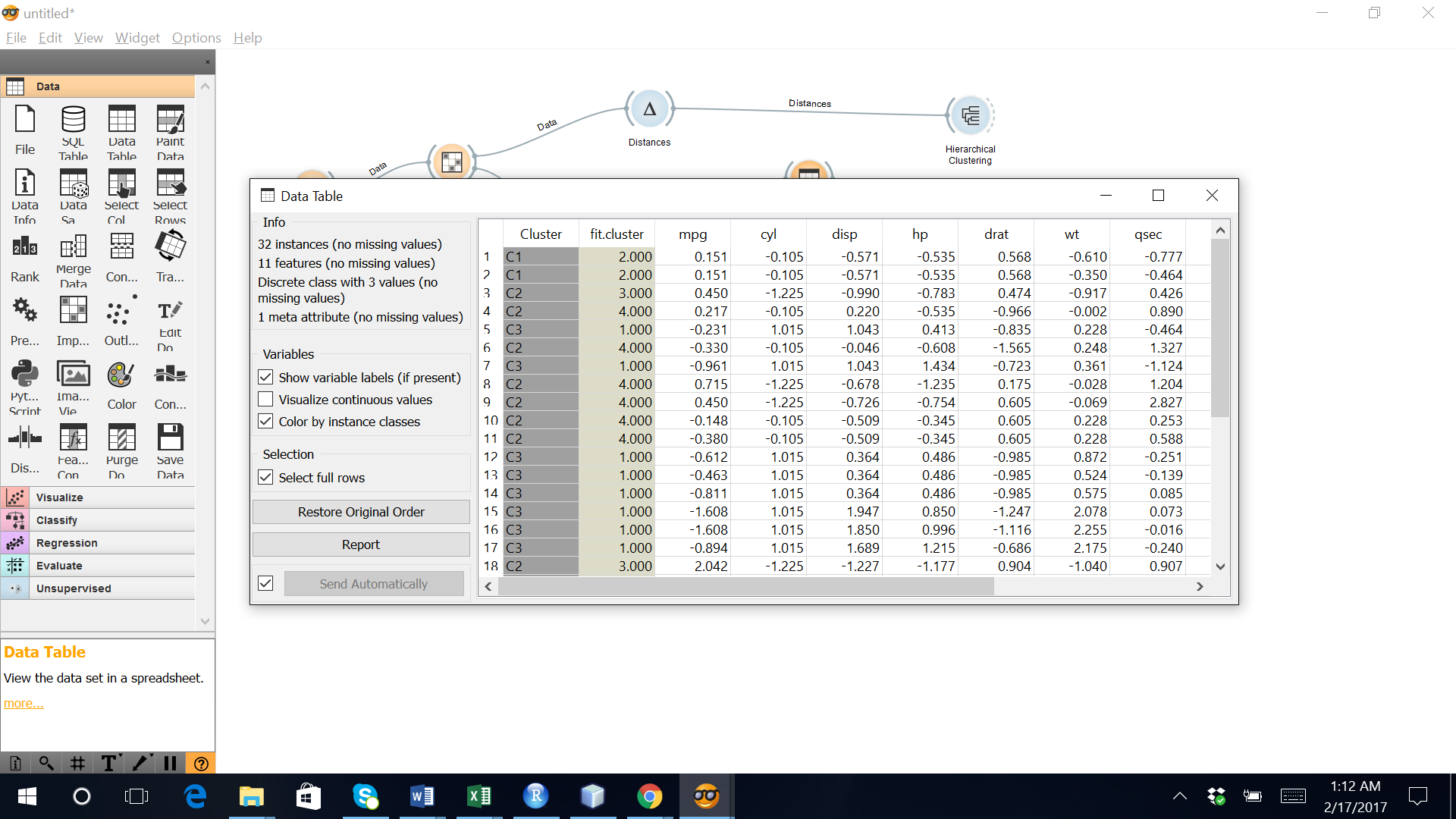


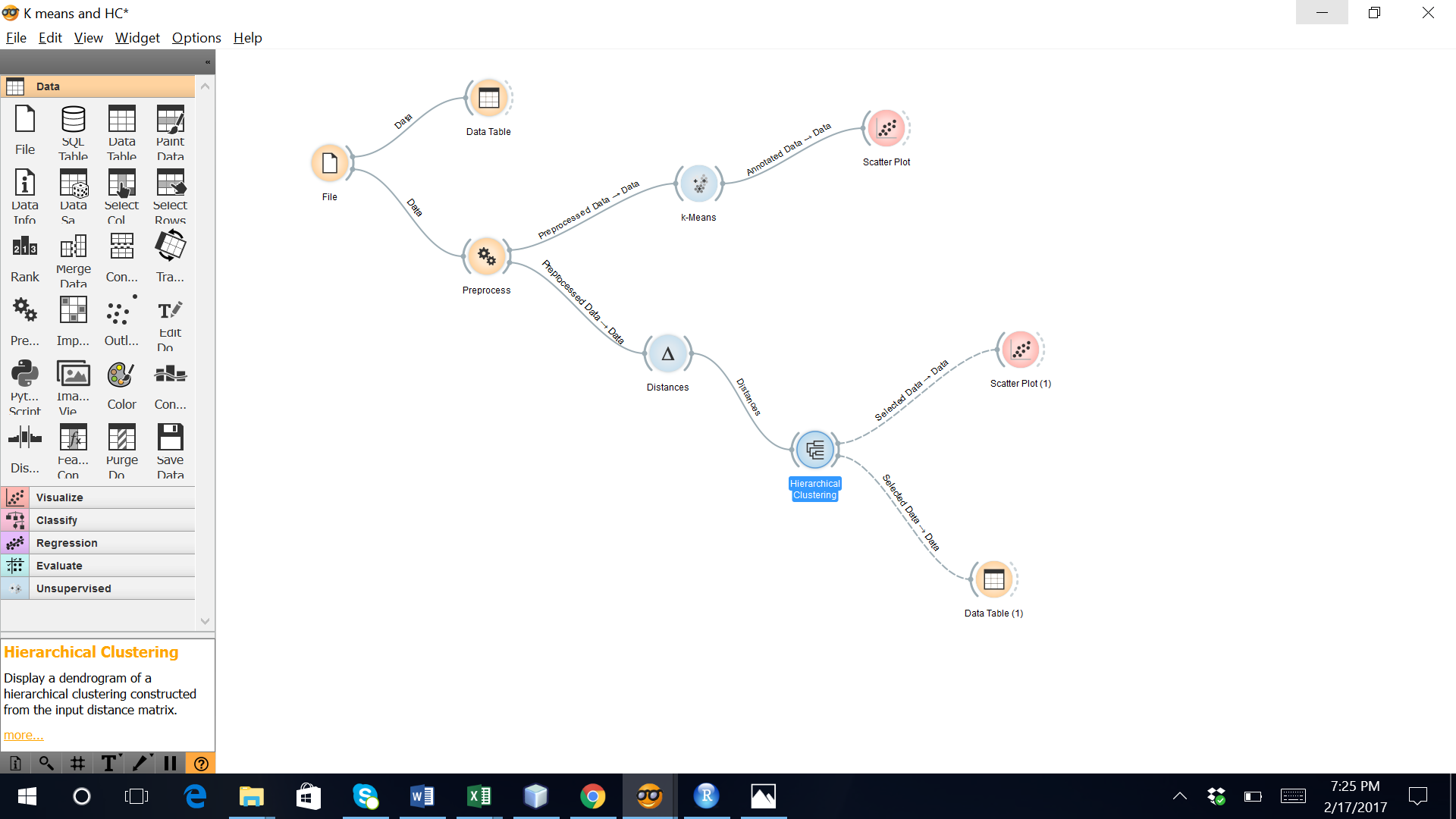
“Hierarchial Clustering” widget can be then connected to a “DataTable” component to view the cluster compnents in tabular format.



“Scatter Plot” is used to visualise the clusters where we can choose the axes and the color of the plot.







**Inference:** Hierarchical clustering on data using both R and Orange resulted the same clusters.

# Kmeans clustering:

K Means Clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity. Unsupervised learning means that there is no outcome to be predicted, and the algorithm just tries to find patterns in the data. In k means clustering, we have the specify the number of clusters we want the data to be grouped into. The algorithm randomly assigns each observation to a cluster, and finds the centroid of each cluster. Then, the algorithm iterates through two steps:

Reassign data points to the cluster whose centroid is closest.

Calculate new centroid of each cluster.

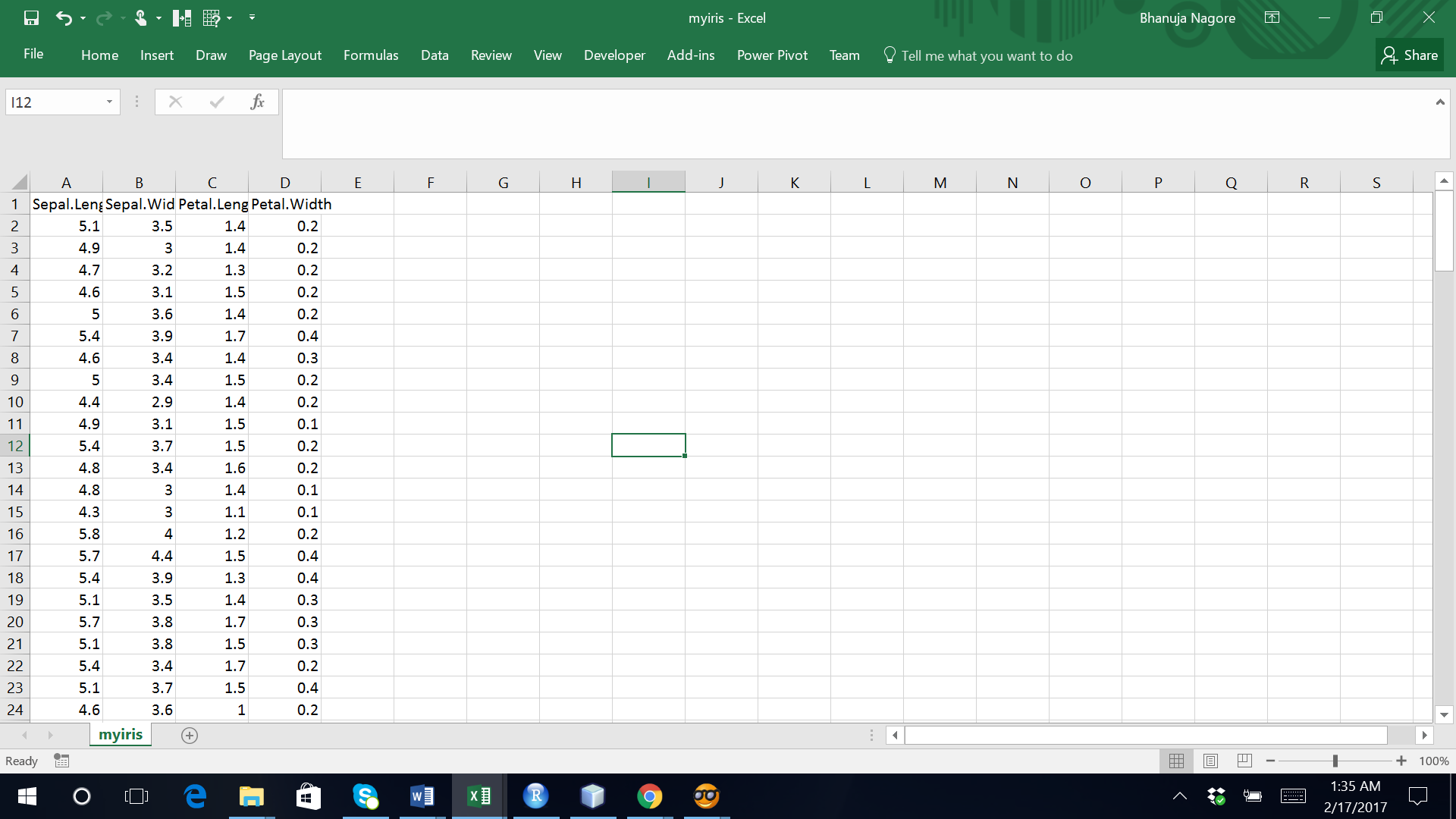
These two steps are repeated till the within cluster variation cannot be reduced any further. The within cluster variation is calculated as the sum of the euclidean distance between the data points and their respective cluster centroids.

The iris dataset contains data about sepal length, sepal width, petal length, and petal width of flowers of different species.

R:

The data is read

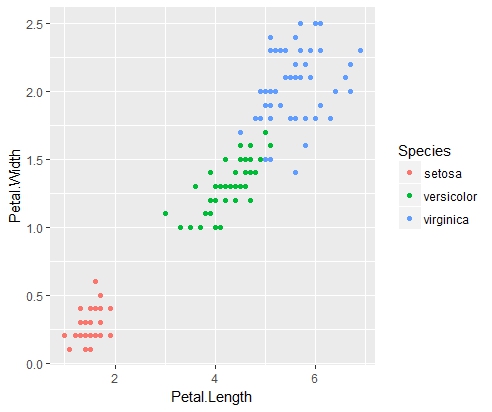
iris



Cluster is plotted using plot.

Petal.Length and Petal.Width were similar among the same species but varied considerably between different species

*plot(clusters)*

**

hierarchical cluster is cut by specifying he number of groups/clusters

Cluster is tabulated with the decision variable of data

> library(datasets)

> head(iris)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

1 5.1 3.5 1.4 0.2 setosa

2 4.9 3.0 1.4 0.2 setosa

3 4.7 3.2 1.3 0.2 setosa

4 4.6 3.1 1.5 0.2 setosa

5 5.0 3.6 1.4 0.2 setosa

6 5.4 3.9 1.7 0.4 setosa

> library(ggplot2)

> ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom\_point()

> set.seed(20)

> irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20)

> irisCluster

K-means clustering with 3 clusters of sizes 50, 52, 48

Cluster means:

Petal.Length Petal.Width

1 1.462000 0.246000

2 4.269231 1.342308

3 5.595833 2.037500

Clustering vector:

[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[38] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

[75] 2 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 2 3 3 3 3

[112] 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3

[149] 3 3

Within cluster sum of squares by cluster:

[1] 2.02200 13.05769 16.29167

(between\_SS / total\_SS = 94.3 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss"

[6] "betweenss" "size" "iter" "ifault"

> table(irisCluster$cluster, iris$Species)

setosa versicolor virginica

1 50 0 0

2 0 48 4

3 0 2 46

> irisCluster$cluster <- as.factor(irisCluster$cluster)

> ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom\_point()

> # Determine number of clusters

> mydata <- iris

> mydata$Species <- NULL

> wss <- (nrow(mydata)-1)\*sum(apply(mydata,2,var))

> for (i in 2:10) wss[i] <- sum(kmeans(mydata, centers=i)$withinss)

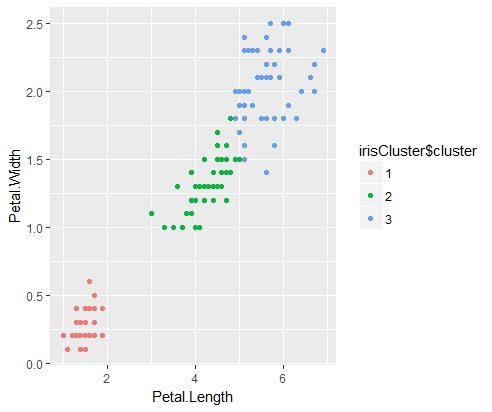
> plot(1:10, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")

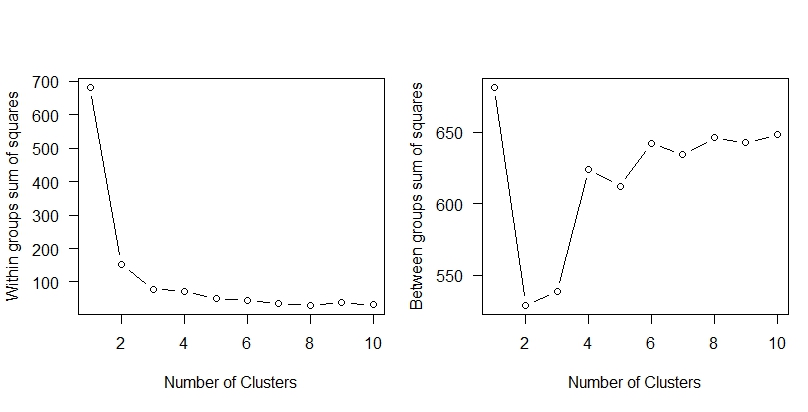
> for (i in 2:10) wss[i] <- sum(kmeans(mydata, centers=i)$betweenss)

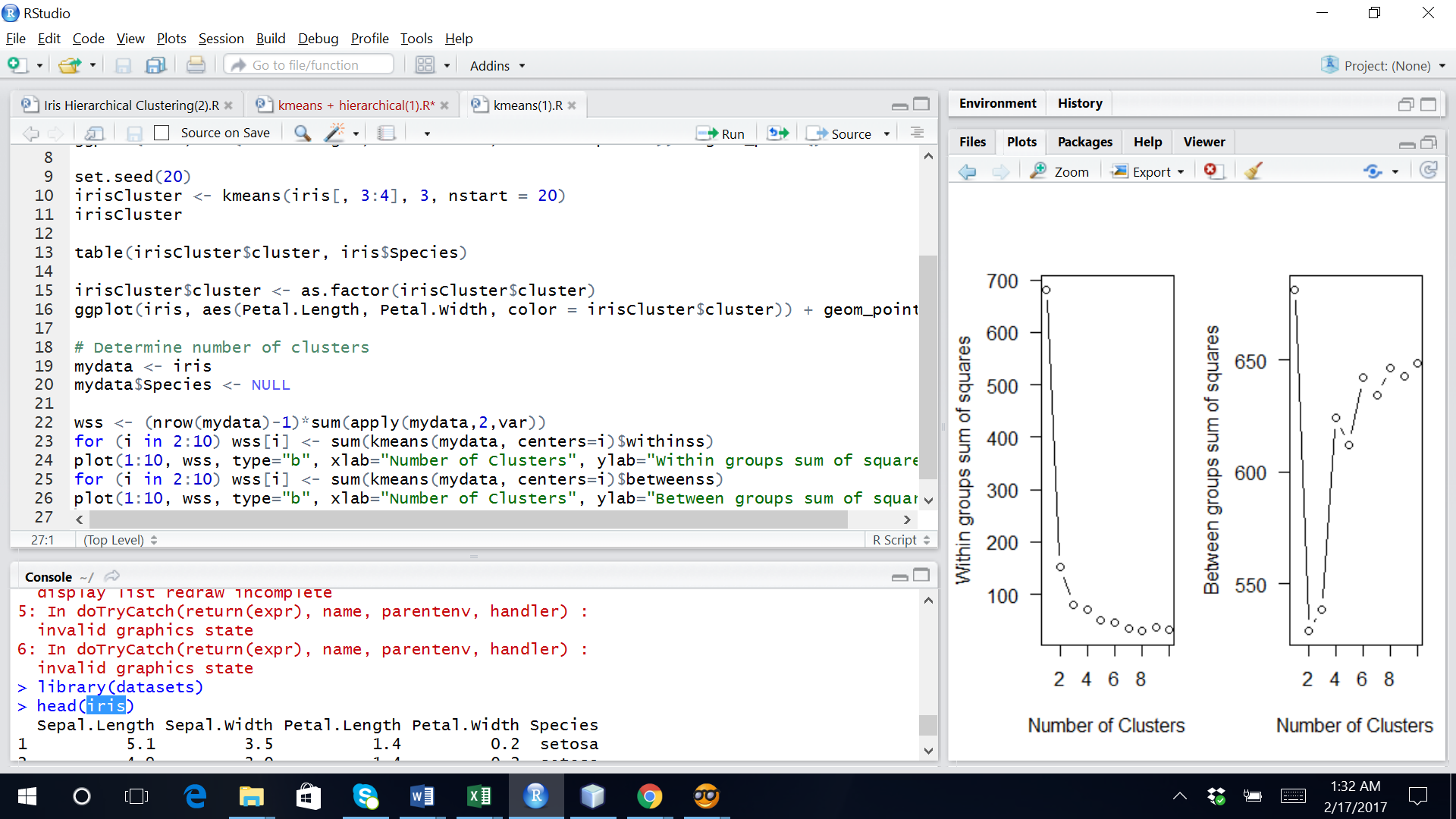
> plot(1:10, wss, type="b", xlab="Number of Clusters", ylab="Between groups sum of squares")

the data belonging to the setosa species got grouped into cluster 3, versicolor into cluster 2, and virginica into cluster 1. The algorithm wrongly classified two data points belonging to versicolor and six data points belonging to virginica.

plot the data to see the clusters:

**

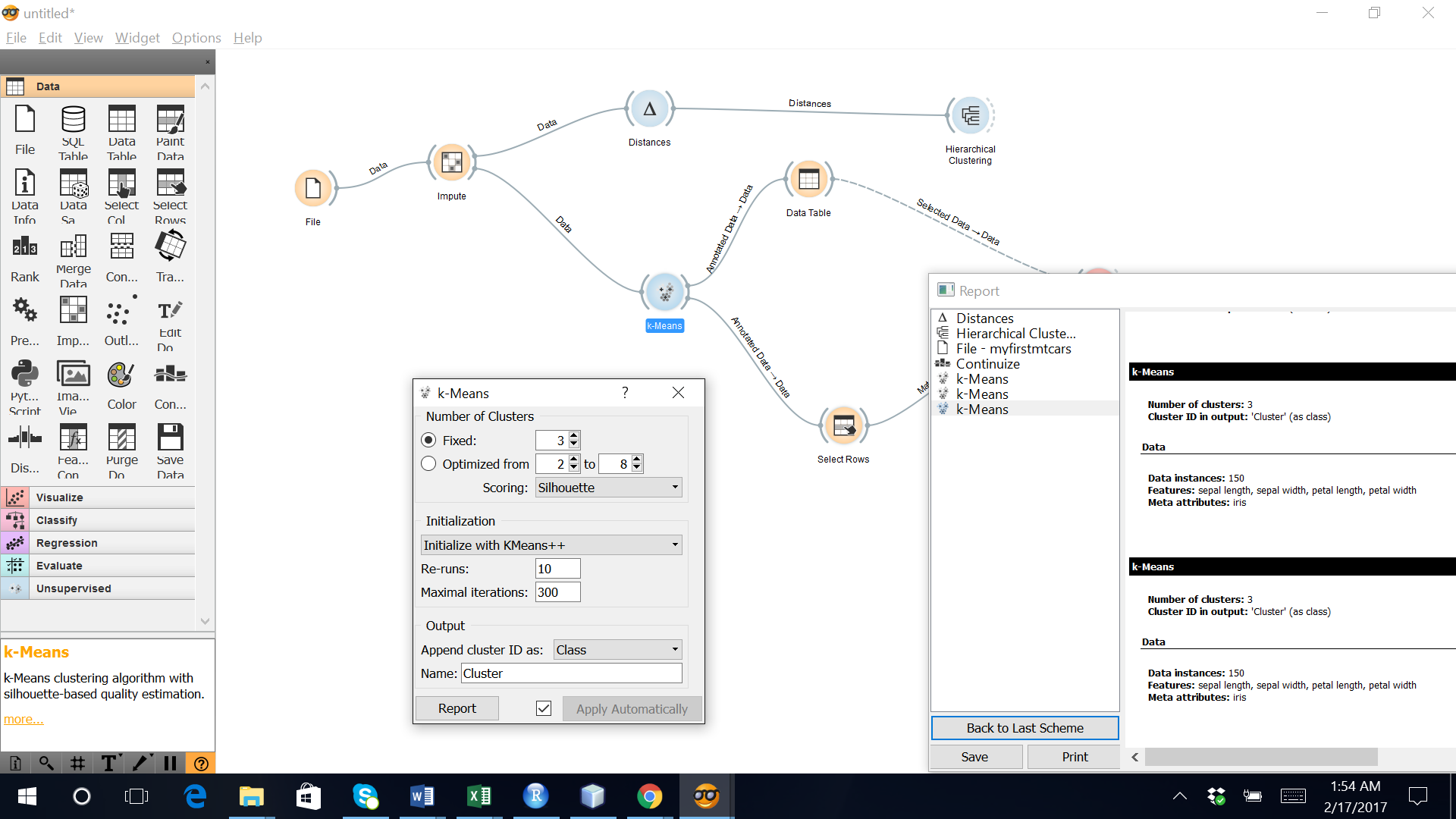




Orange:

Data from input file is imported using “file” component.

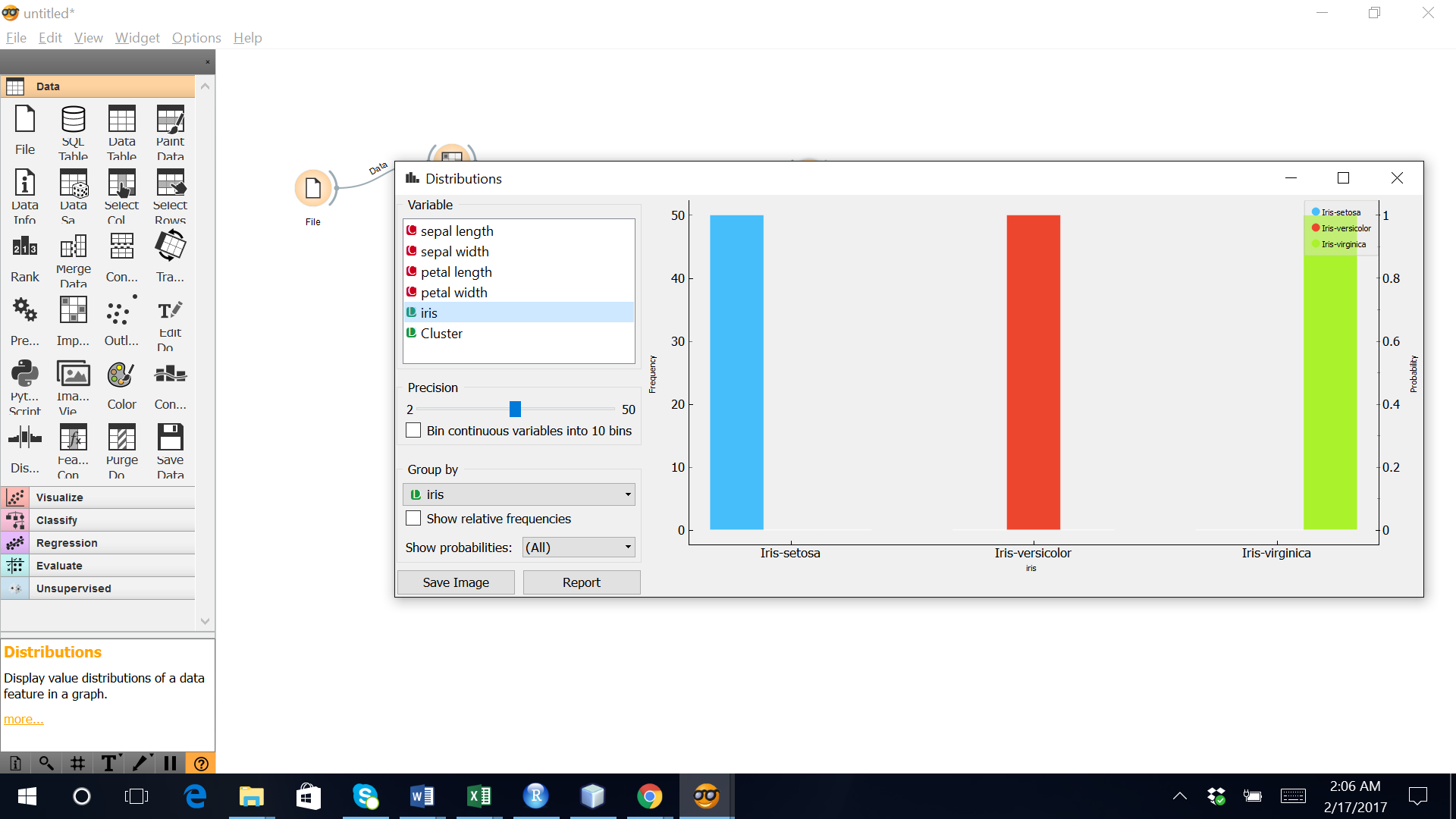
[k-Means clustering](https://en.wikipedia.org/wiki/K-means_clustering) algorithm to the data and outputs a new data set in which the cluster index is used as a class attribute. The original class attribute, if it exists, is moved to meta attributes. Scores of clustering results for various k are also shown in the widget.

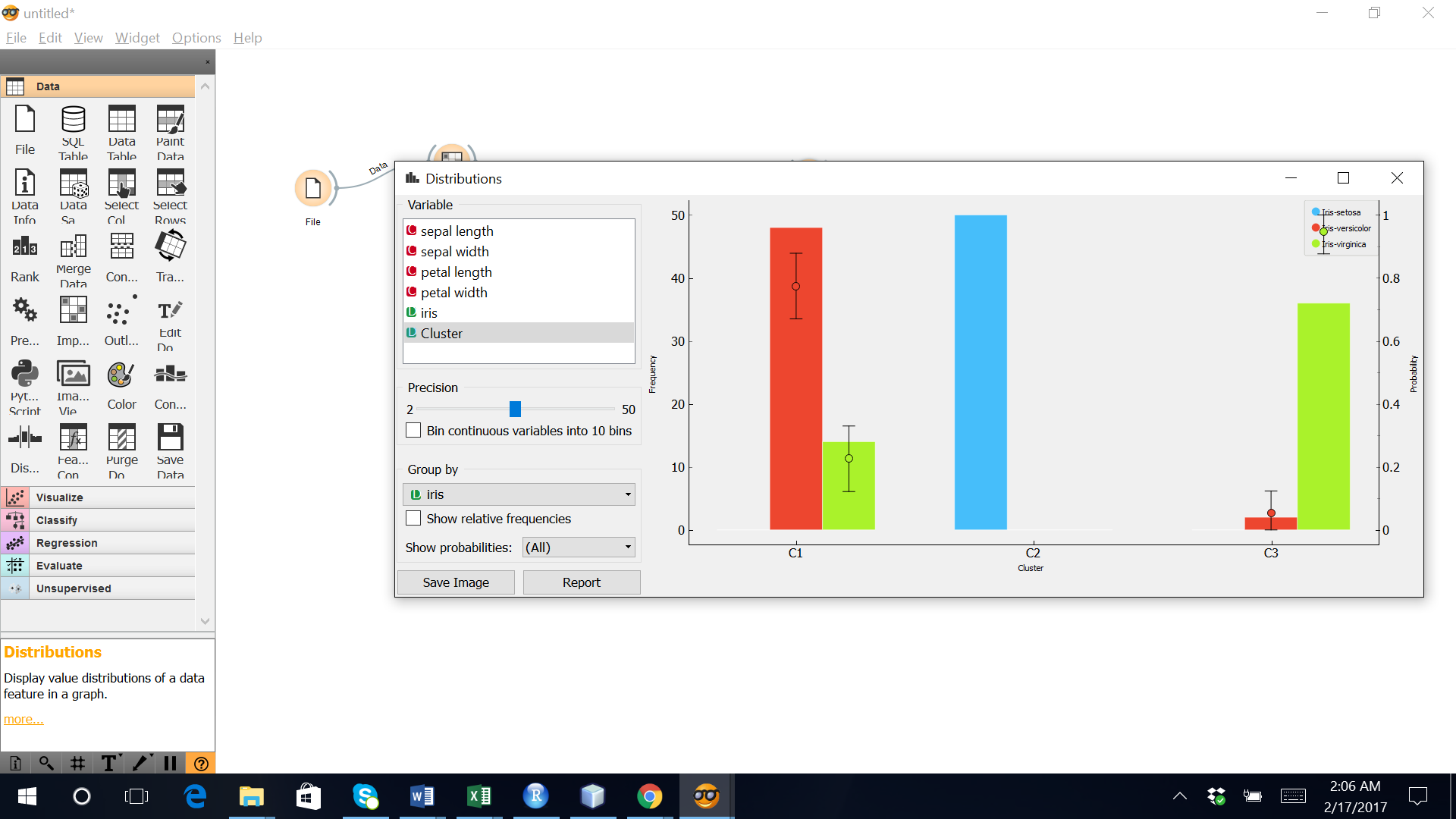


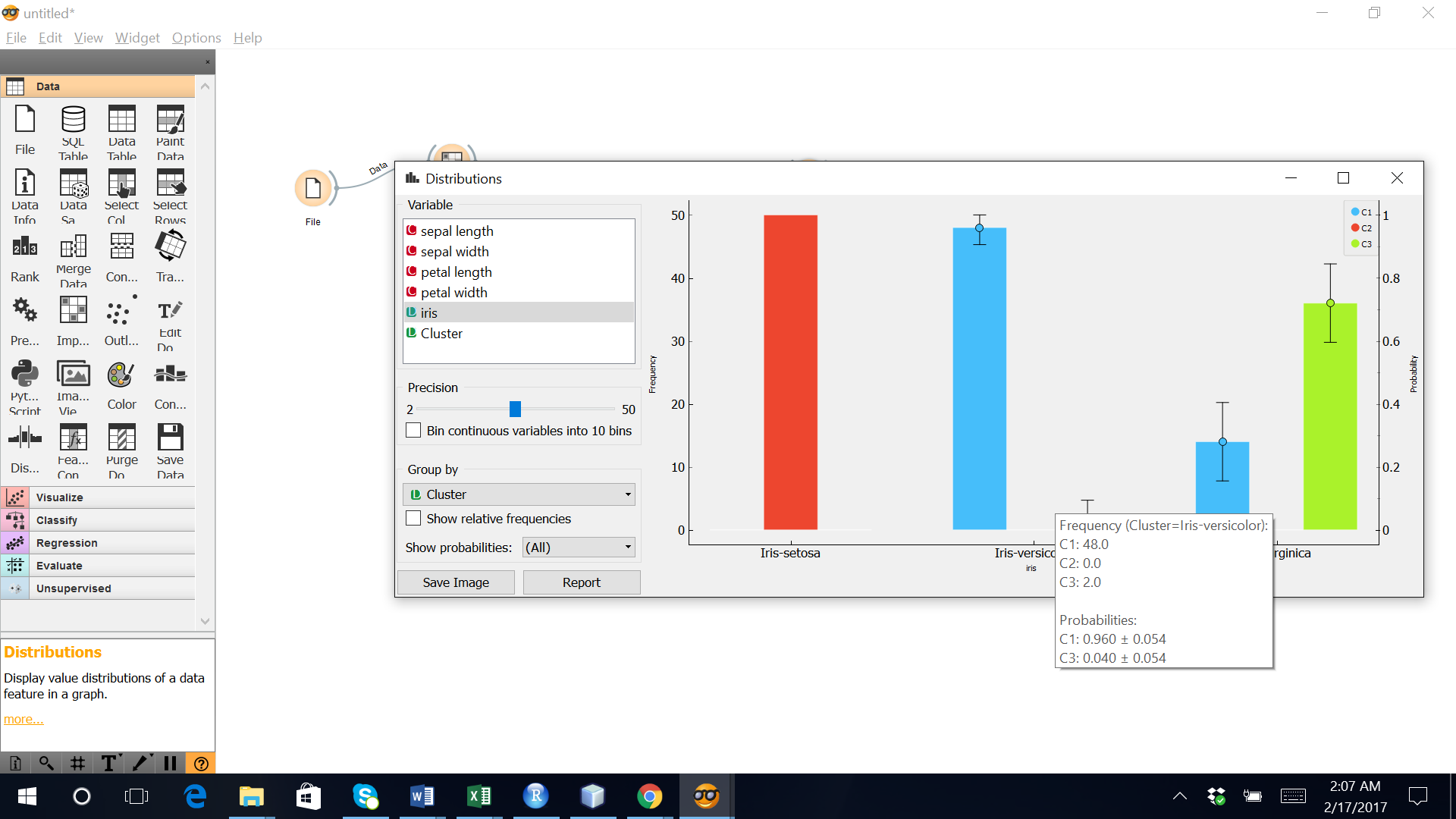
“Distances” widget is used to calculate the distance matrix using Euclidean metric.

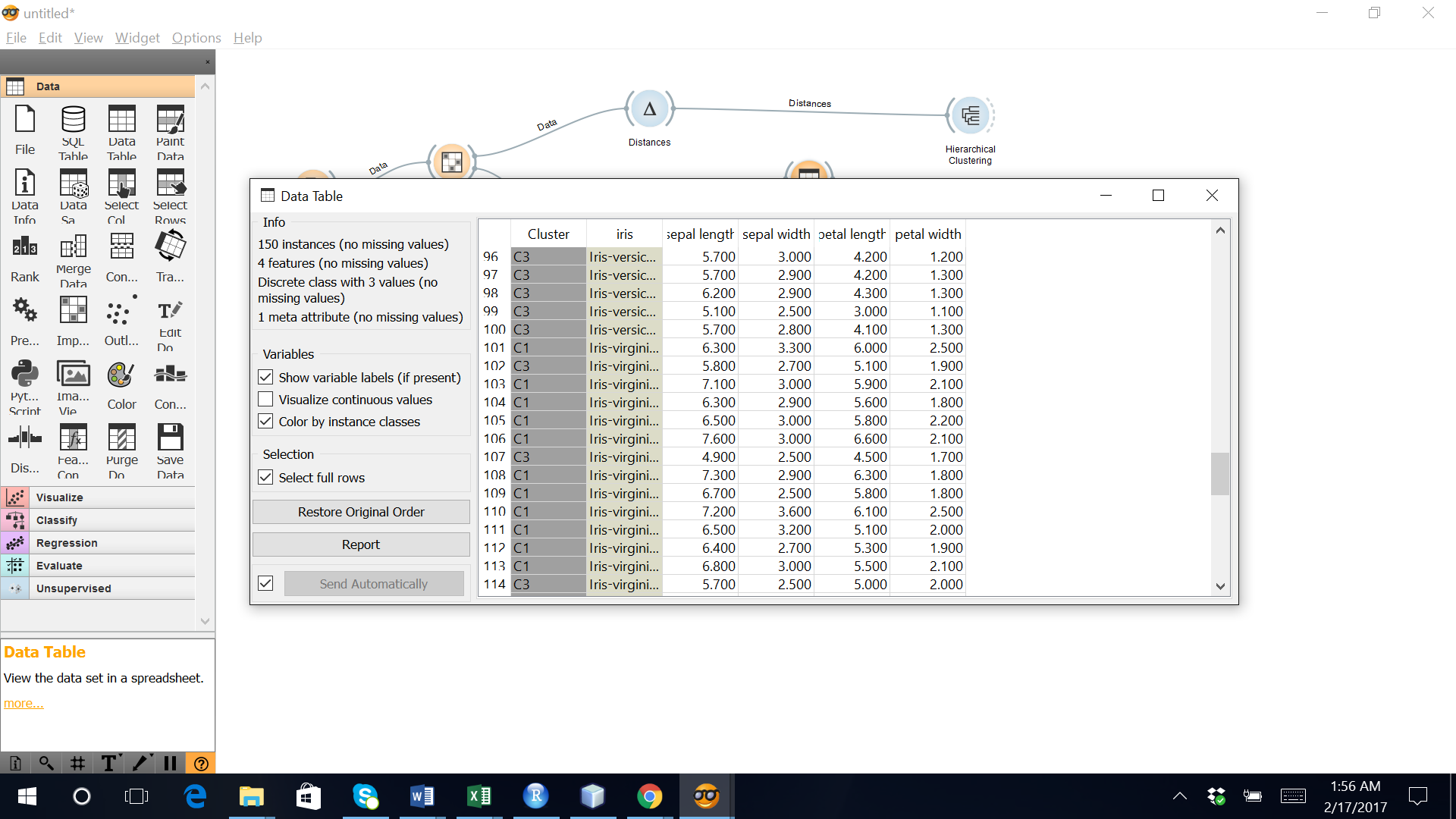
“K mean Clustering” widget is used to cluster the data hierarchically where we can specify the linkage type and the number of clusters is 3.

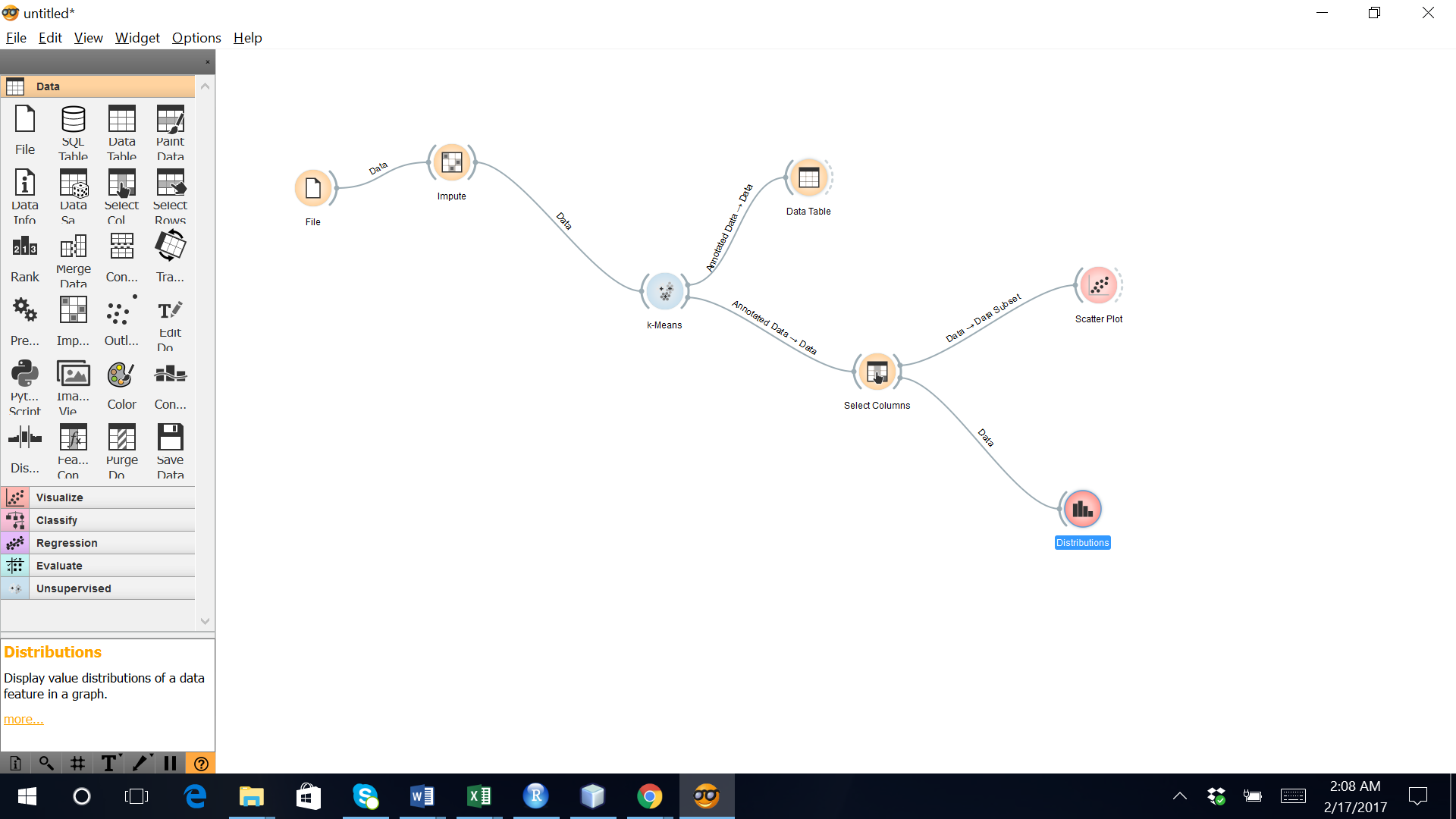
“Scatter Plot” is used to visualise the clusters where we can choose the axes and the color of the plot.











**Inference:** Hierarchical clustering on data using both R and Orange resulted the same clusters.