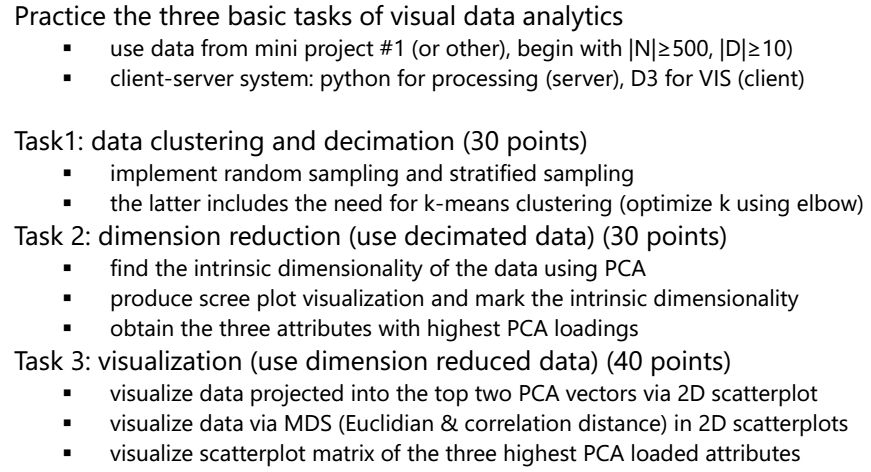
**Visualization Mini Project 2**

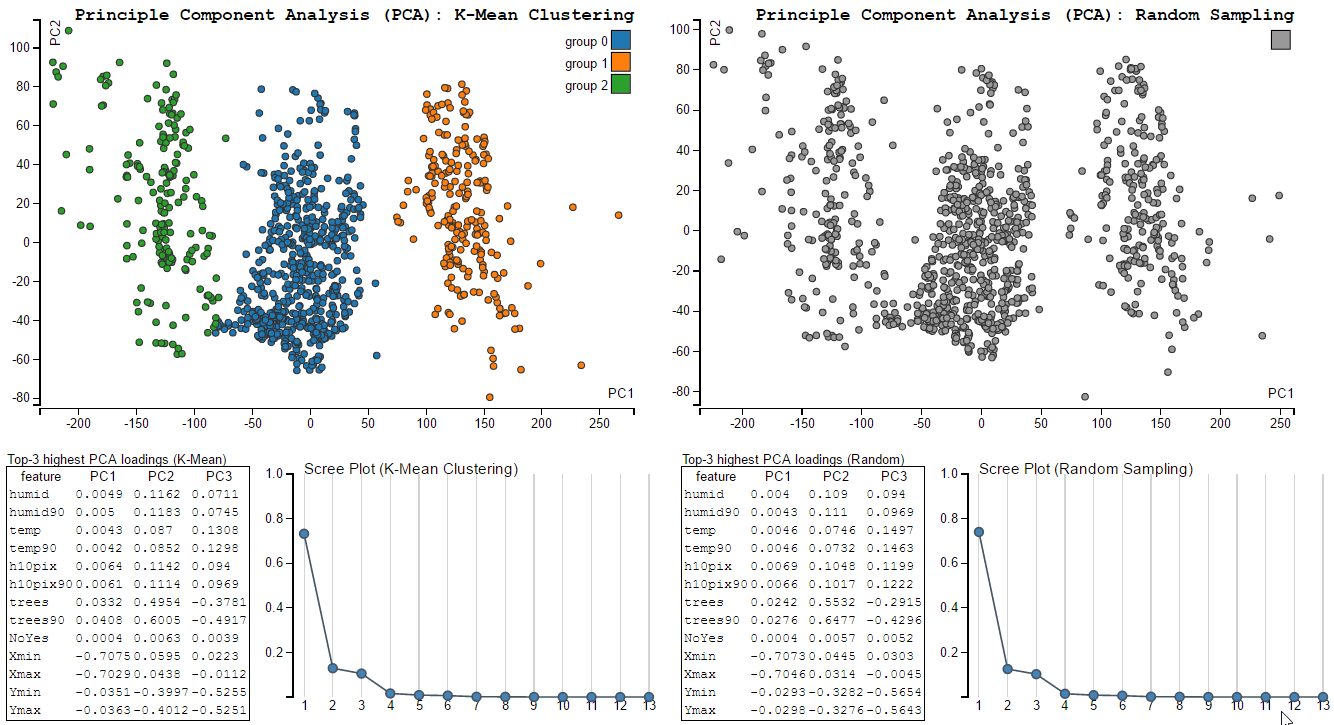
1. Introduction:

In this assignment, we learn to

1. Use Python to import CSV file and pre-process a large amount of data
2. Use Python to reduce data by random sampling and adaptive sampling with K-Means clustering
3. Use Python to perform dimension reduction by using Principal component analysis (PCA) and Multidimensional scaling (MDS)
4. Use the d3 package for elegant visual effect and animation for data visualization.
5. Build a user friendly interface via basic HTML and CSS techniques.
6. Requirement:



1. Layout
   1. Principle Component Analysis (PCA):



**(6)**

**(5)**

**(4)**

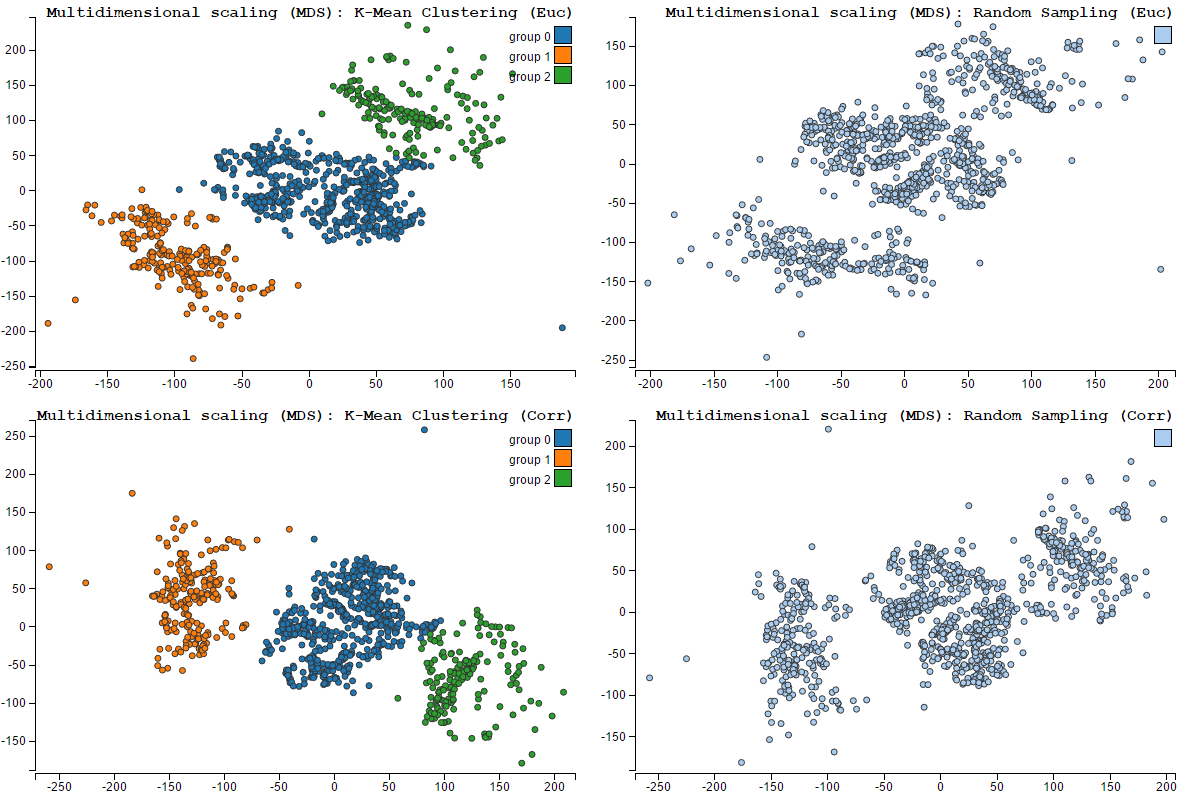
**(3)**

**(2)**

**(1)**

Components:

1. Scatter plot of data projected into the top two PCA vectors (Adaptive sampling + K-Mean clustering)
2. Scatter plot of data projected into the top two PCA vectors (Random sampling)
3. Matrix of the three highest PCA loaded attributes (Adaptive sampling + K-Mean clustering)
4. Scree plot to show all eigenvalues in PCA (Adaptive sampling + K-Mean clustering)
5. Matrix of the three highest PCA loaded attributes (Random sampling)
6. Scree plot to show all eigenvalues in PCA (Random sampling)
   1. Multi-Dimensional Scaling (MDS):



**(7)**

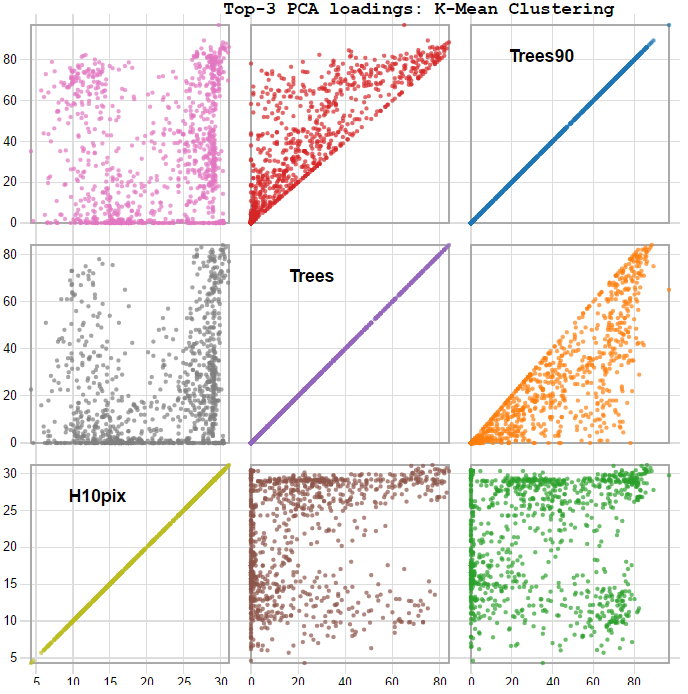
**(8)**

**(10)**

**(9)**

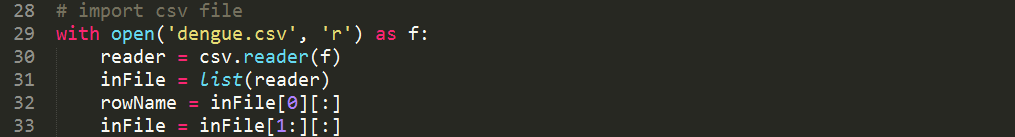
Components:

1. Scatter plot of data after MDS with Euclidian distance (Adaptive sampling + K-Mean clustering)
2. Scatter plot of data after MDS with Euclidian distance (Random sampling)
3. Scatter plot of data after MDS with Correlation distance (Adaptive sampling + K-Mean clustering)
4. Scatter plot of data after MDS with Correlation distance (Random sampling)
   1. Scatter plot matrix of top-3 PCA loadings:

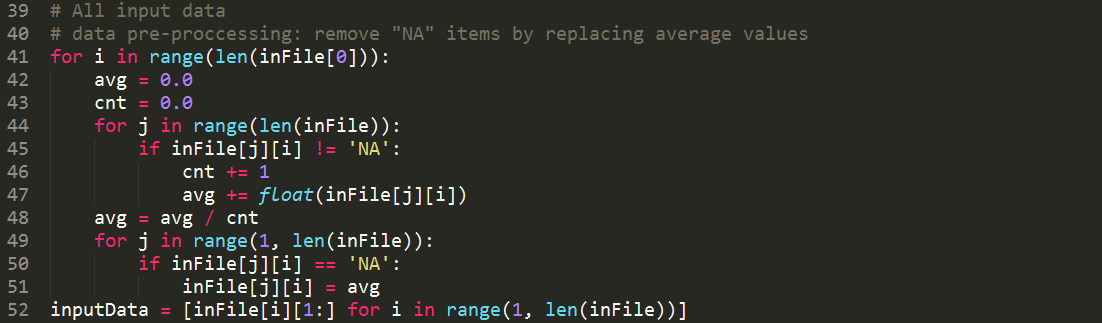


Components:

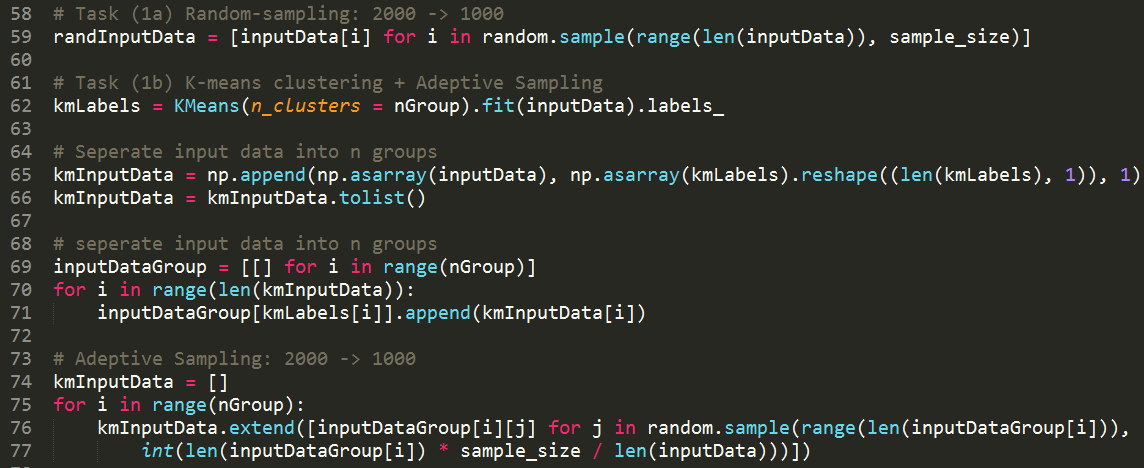
1. Scatter plot matrix of Top-3 PCA loadings (K-Mean clustering)
2. Code Implementation in Python
   1. Data Source: <https://vincentarelbundock.github.io/Rdatasets/doc/DAAG/dengue.html> -> Dengue fever
   2. Pre-process data
      1. Import csv file



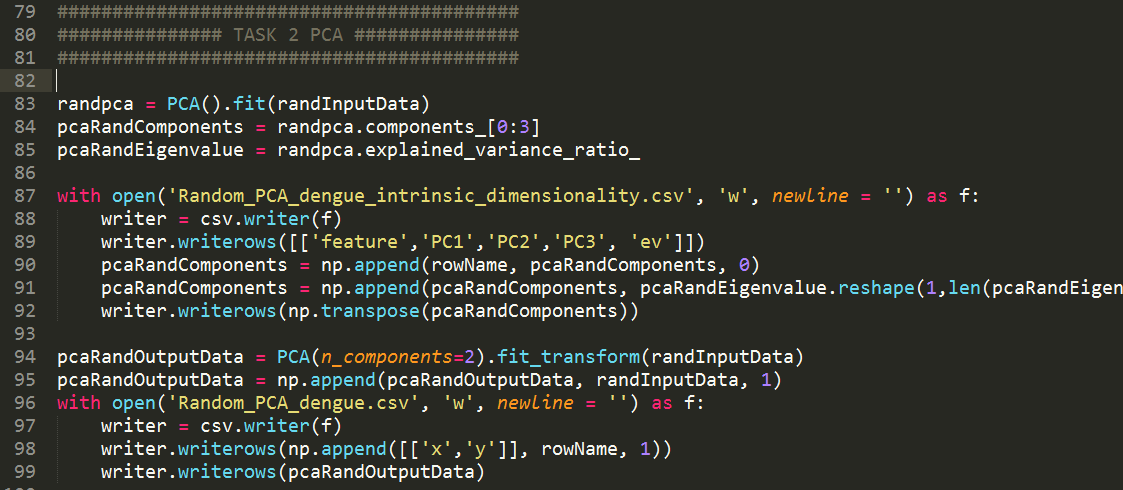
* + 1. Remove invalid values in the original data



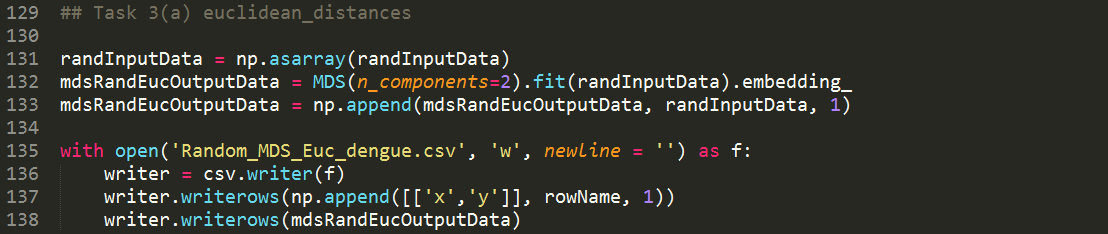
* 1. Data reduction
     1. Random sampling and Adaptive sampling with K-Mean clustering
     2. Write the output data after sampling into a csv file



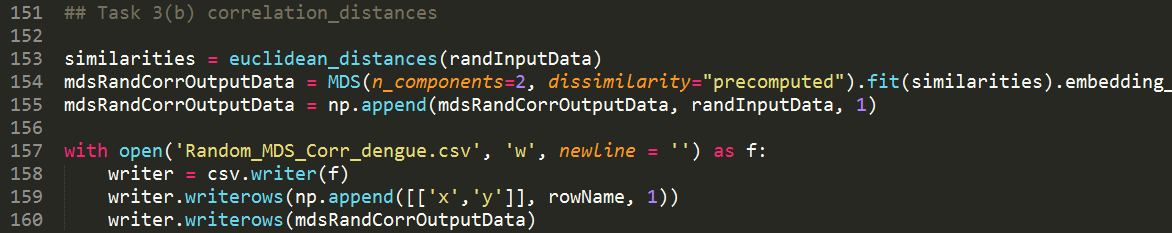
* 1. Dimension reduction - PCA
     1. Perform PCA
     2. Obtain the three attributes with highest PCA loadings (for Top-3 PCA loadings)
     3. Obtain all eigenvalues (for scree plot)
     4. Obtain the values projected into the top two PCA vectors (for scatter plot)
     5. Write the output data of PCA into a csv file



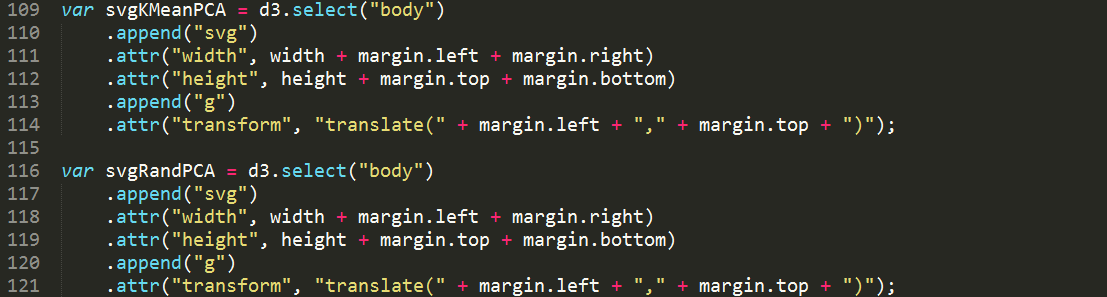
* 1. Dimension reduction - MDS
     1. Perform MDS with Euclidean distance



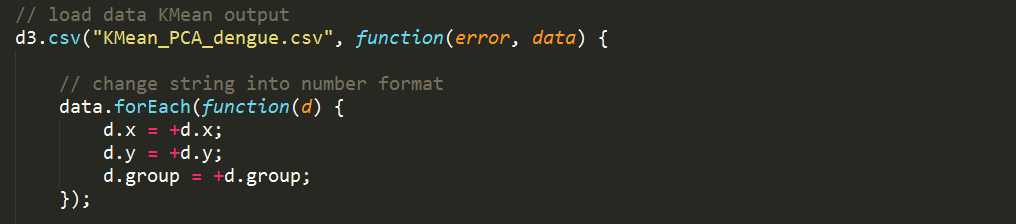
* + 1. Perform MDS with correlation distance



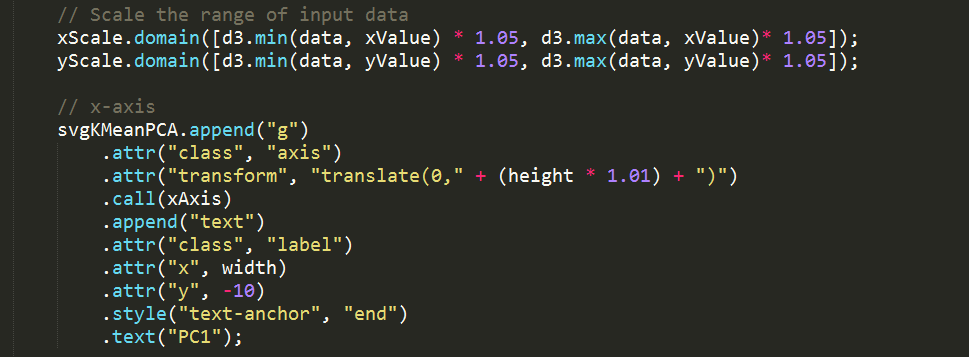
1. Code Implementation in D3
2. Create all SVG elements



1. Load csv output data



1. Scale the range of output data



1. Draw scatter plots and add d3 mouse events: MouseOver, MouseOut



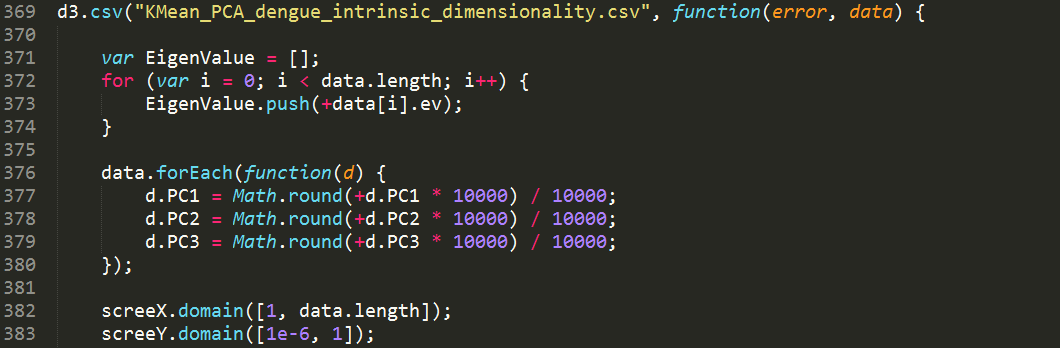
MouseOver event:

Show detail information of the selected point on the graph

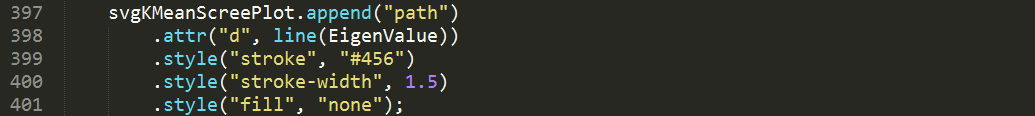
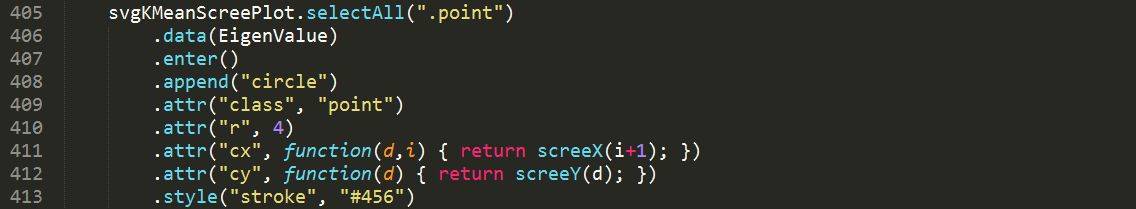
MouseOut event:

Remove the information

1. Load csv output data or scree plots



1. Draw points and lines on the scree plot



1. Build a table to show the top-3 PCA loadings



1. Scatter plots for all the results in MDS

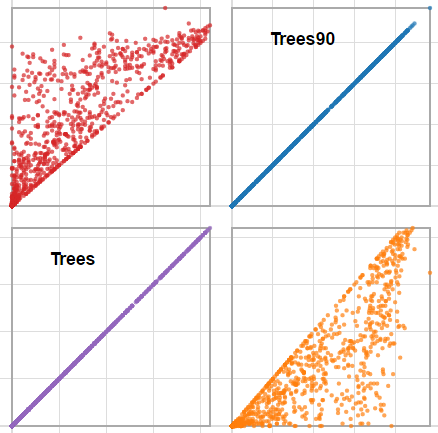


1. How to setup Python server:

Command Line: “ python –m http.server 8000 “

Open your browser with <http://your_ip_address:8000/>

1. Discussion:
   1. The grouping results via K-Means clustering are good. From the scatter plots of PCA, we can found points of different color are almost separated into three groups. (K = 3)
   2. In the scatter plot matrix, the attributes top-3 PCA loadings are “Trees”, “Trees90”, “H10pix”. It is obvious that the value in “Trees” and “Trees90” are correlated, so the figure shows a triangle instead of random plot on the figure.



* 1. MDS takes much longer processing time than PCA. In one round, PCA takes about 3-5 seconds while MDS take about 1 minute.