Name : Rishita Reddy Punuru
SBU ID : 113274815

Assignment 2 - Report

Chosen Dataset : Pokémon Statistics
Dataset link : https://www.kaggle.com/alopez247/pokemon
Cleaned Dataset file link :
https://github.com/rishitareddy/visualization/blob/main/pokemon_filtered_data.csv

Dataset Description : The dataset contains 721 Pokémon statistics. I have cleaned and used 10 numerical attributes and 5 categorical attributes for the assignment.
The attributes I have chosen are as follows:

Attribute	Type	Brief Description
Total	Numerical	Sum of the base stats
HP	Numerical	Base Health Points
Attack	Numerical	Base Attack
Defense	Numerical	Base Defense
Sp_Atk	Numerical	Base Special Attack
Sp_Def	Numerical	Base Special Defense
Speed	Numerical	Base Speed
Height_m	Numerical	Height of the Pokémon, in meters
Weight_kg	Numerical	Weight of the Pokémon, in kilograms
Catch_Rate	Numerical	Catch Rate of the Pokémon
Type_1	Categorical	Primary type
Type_2	Categorical	Secondary type
Color	Categorical	Color of the Pokémon
Egg_Group_1	Categorical	Egg Group of the Pokémon
Body_Style	Categorical	Body Style of the Pokémon

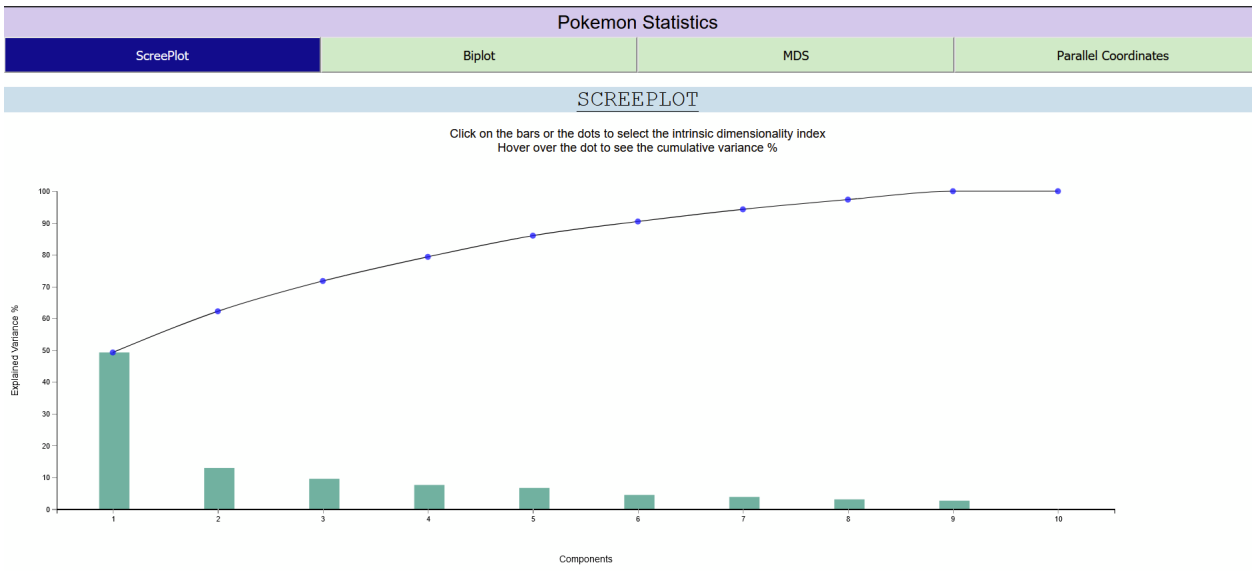
Code Implementation: The assignment was done using HTML, CSS, JavaScript, AJAX and D3.js.

The following files can be found in my project folder:

1. api.py : The python file through which the code will start running.
   You can run the code by running the command -> python api.py
   This will give you the link where the visualization can be accessed.
2. index.html : The main html file where all the component placements are defined.
3. screeplot.js :This file contains the implementation for the screeplot.
4. biplot.js : This file contains the implementation for the biplot.
5. pcaLoadingsTable.js : This file contains the implementation for the loadings table.
6. scatterplotMatrix.js : This file contains the implementation for the scatterplot matrix.
7. mdsEuclidean.js : This file contains the implementation for the euclidean mds plot.
8. mdsCorrelation.js : This file contains the implementation for the correlation mds plot.
9. parallelCoordinates.js : This file contains the implementation for the parallel coordinates plot.
10. main.css : It contains the styling components.
11. pokemon-filtered-data.csv : This is the dataset used for the visualization.

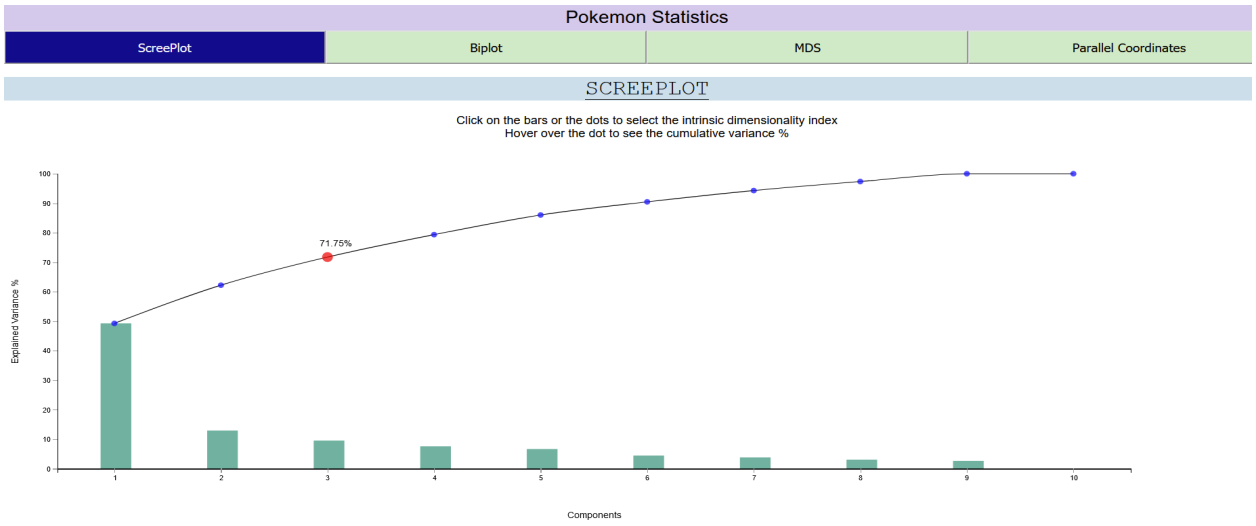
Below are the features implemented.

1. (a) Screeplot
(b) Interaction element on screeplot to select dimensionality index



On clicking on the bar or the point, the index is selected.

The user can also hover on the dot to view the cumulative variance %.



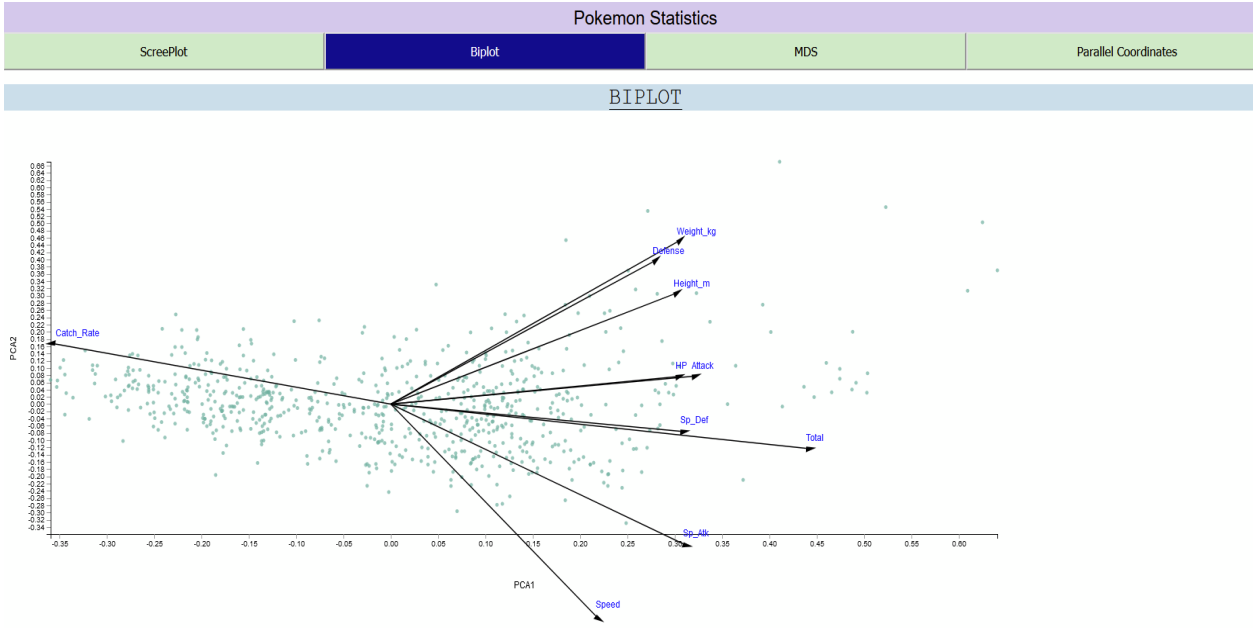
Strength of a screeplot for PCA analysis:

- We can easily find the least number of dimensions we can choose without losing too much data.
- It is a great tool for dimensionality reduction analysis.

Weakness:

- If the variables are not strongly correlated, it won't help.

(c) PCA based biplot



Strength of a PCA based biplot:

- We can easily view the attributes and data together and find their distribution and relationship.

Weakness:

- If there are more than 2 significant PCA's, variability will be lost.
- There can be inaccuracies as neighbours in 2D may not be neighbours in higher dimensions.

2. (a) PCA loadings table

The loadings of top 4 attributes are generated based on the di selected from the screeplot. If 3 was selected, the table would be as follows.

PCA LOADINGS TABLE

The intrinsic dimensionality index selected is 3

Attribute	PC1	PC2	PC3
Speed	0.21613059434285423	-0.5858782817303179	-0.31073686246542914
Sp_Def	0.3052486906547468	-0.0751553456356026	0.5890279178002893
Defense	0.2760948374358623	0.39303529174634966	0.4531801720704073
Weight_kg	0.30160812603523784	0.448101541922272	-0.23457807817391566

If 5 was selected, the table would be as follows:

PCA LOADINGS TABLE

The intrinsic dimensionality index selected is 5

Attribute	PC1	PC2	PC3	PC4	PC5
HP	0.30020221381622086	0.07563639105671462	-0.19441915657539238	0.4301509805254751	0.6868843478542932
Speed	0.21613059434285423	-0.5858782817303179	-0.31073686246542914	-0.19863232602703274	-0.35065003059258354
Defense	0.2760948374358623	0.39303529174634966	0.4531801720704073	-0.41633548599546283	-0.10878053574071281
Attack	0.3173846690758078	0.07657494705374342	-0.3177804866548836	-0.5690732227559959	0.27248784734891884

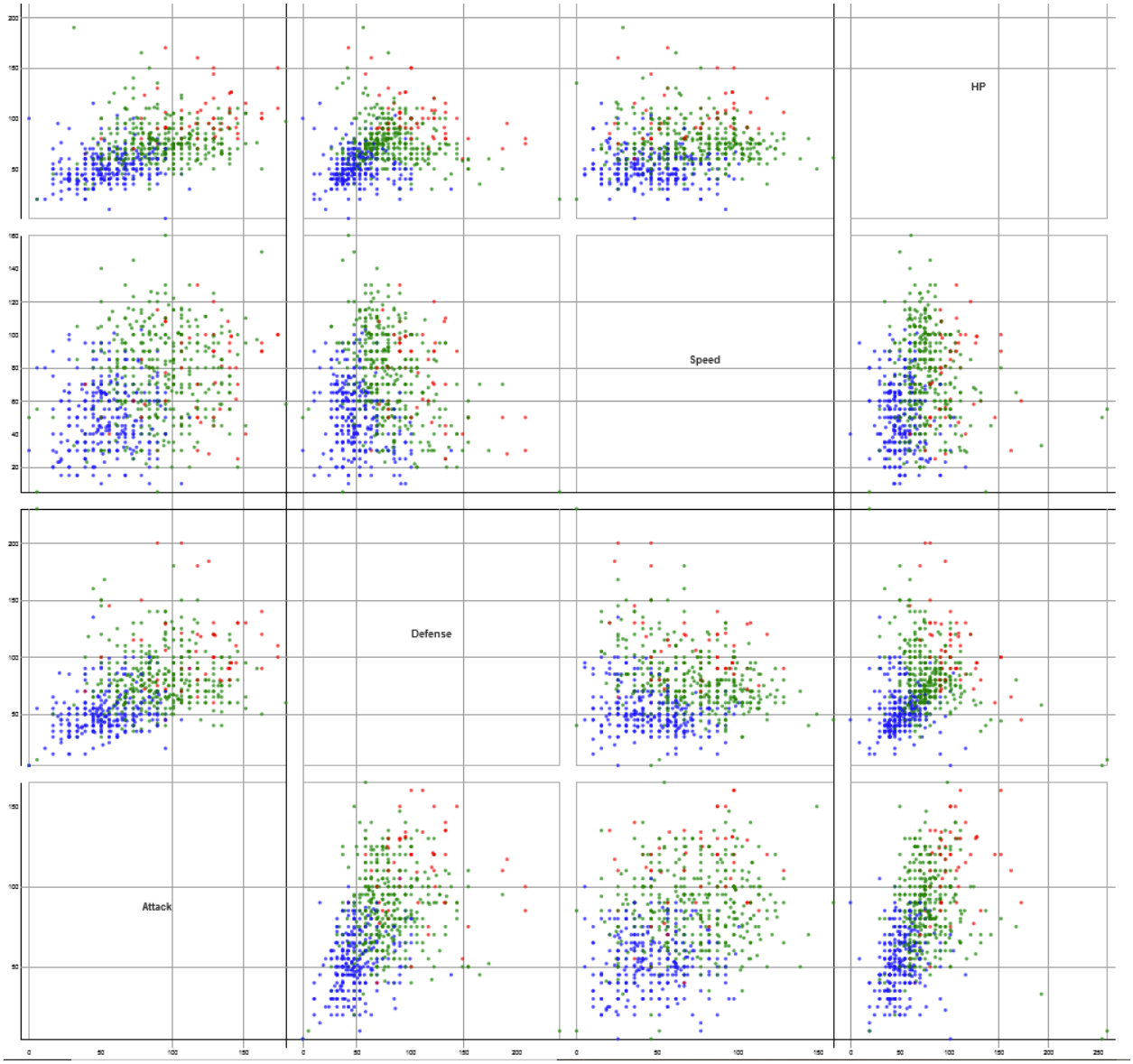
Strength of PCA loadings table:

- We can view the correlation between attributes easily; and also if they are positively or negatively related.

(b) Scatterplot Matrix

(c) K-means to cluster and color the points

Similar to the 2(a) part, matrix depends on the dimensionality index selected. If di was 5, the following matrix gets displayed with 3 clusters.



Strength of a scatterplot matrix:

- It will easily help to determine if there is a linear correlation between multiple variables. For example, in the above plot, we can see that Attack, Defense and HP are linearly correlated.

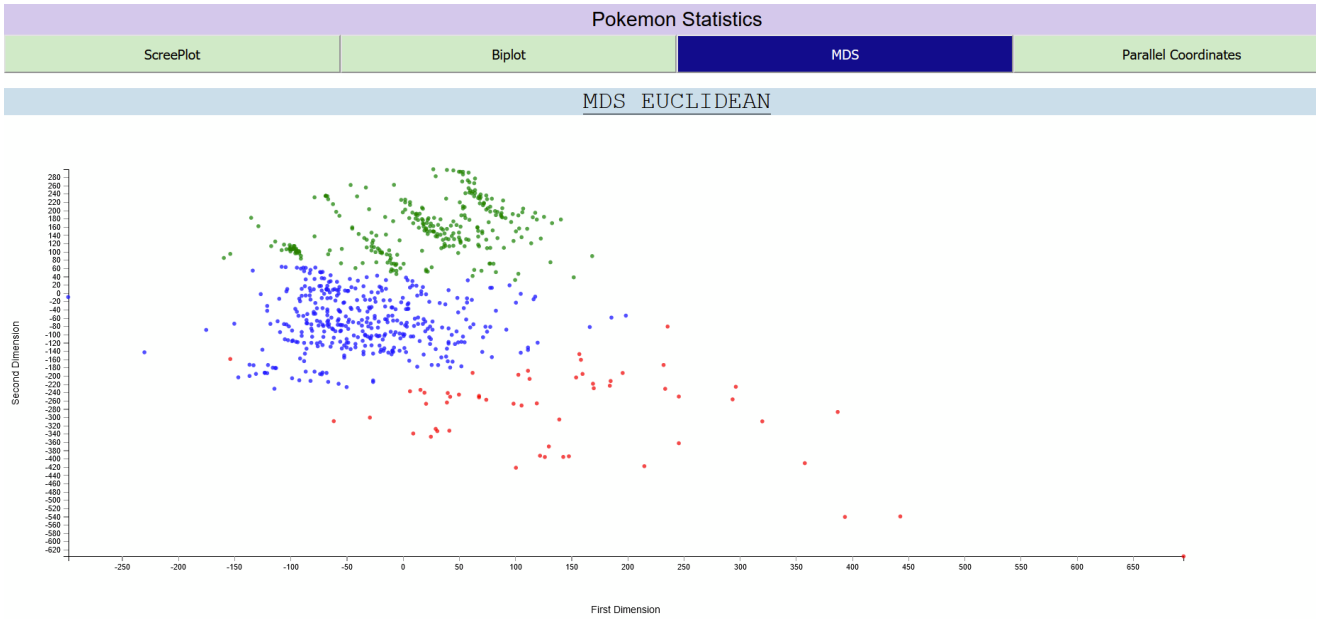
Weakness:

- As multivariate relationships are scattered across, it is hard to see all of the relationships clearly in detail.

I have also added brushing for the scatterplot matrix to view subset of datapoints and their relationship.

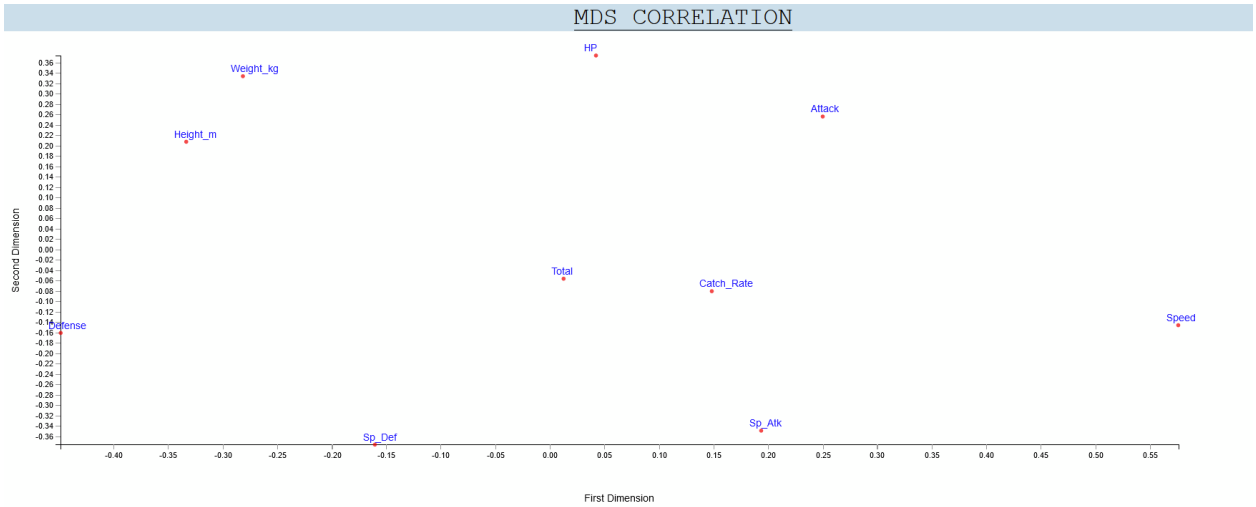
3. (a) MDS plot using Euclidean Distance
(b) K-means to cluster and color the points

Multidimensional scaling is done for datapoints using Euclidean distance, and the generated values are clustered using kmeans (same as scatterplot matrix) and colored.
Below is the plot generated. It doesn't have an X and Y axis in the normal sense as it is just a layout.



(c) MDS plot using Correlation Distance

Multidimensional scaling is done using correlation distance to find similarity metric between attributes. The plot generated for the 10 numerical attributes is as follows:



Strength of MDS plot:

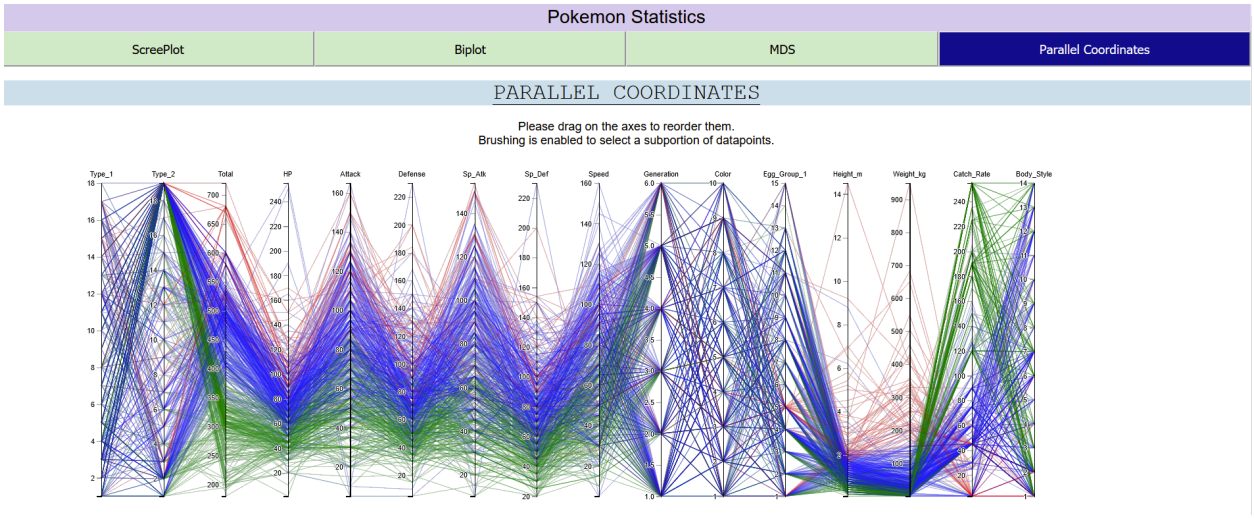
- It preserves the similarity relationships, and helps prevent ambiguity.
- It helps us assess relationships very easily for even a large amount of datapoints. For example, in the MDS using correlation plot, we can directly see the points closer together which are more strongly correlated, like height and weight.

Weakness:

- We need to apply formulas on data before visualizing it, and plot takes more time preprocessing and loading all the datapoints.

4. (a) Parallel Coordinates Plot
(b) Axes ordering with user interaction
(c) K-means to color the polylines by cluster id

PCP is generated for the datapoints (both numerical and categorical) , and the generated polylines are colored using kmeans' cluster id (same as scatterplot matrix).



The user can drag the axes in order to manually rearrange the attribute axes and view relationships between them easily.
I have also enabled brushing to view subselection of datapoints and how they are related among all dimensions.

Strength of PCP:

- It helps us see how all attributes are related to one another.
  For example, in the above plot, we can see height and weight are positively correlated.

Type 1 and Type 2 are negatively correlated.
Total, HP, Attack, Defense, Sp_Atk and Sp_Def are positively correlated.

- It is very easy to find new trends from it.

Weakness:

- Due to the overlaying of datalines, hard to visualize few relationships.

Interesting observations:

- I could easily get all the information about the entire dataset without even worrying about the actual points.
- I was especially impressed to find all the correlations between the attributes (listed as strengths for PCP, MDS and Scatterplot Matrix).