## Exercise 1: Distance Iris-Dataset

*Dataset:*

| Flower 1 | 5,1 | 3,5 | 1,4 | 0,2 | Iris-setosa |
|----------|-----|-----|-----|-----|-------------|
| Flower 2 | 4,9 | 3,0 | 1,4 | 0,2 | Iris-setosa |
| Flower 3 | 6,3 | 3,3 | 4,7 | 1,6 | Iris-versicolor |

All distances between the individual attributes are weighted, added up and their average then forms the total distance between the two data objects.
The distance between the numeric attributes is calculated using their normalized, euclidean distance (meaning their absolute difference is divided by the difference between the highest and lowest value of the respective attribute), whereas the distance of the nominal feature is either 1 (when they're different) or 0 (when they're the same).
The resulting distance is already normalized since all five individual distances fall in the intervall [0, 1] and their sum is divided by 5. The feature weights could be adjusted, but since there is no indication to support this, we decided to keep them at 1 for all attributes.

*Please refer to the Jupyter Notebook "211110_ml_g1_u02_e01.ipynb" as well.*

$$d(1,\ 2)\ =\ \frac{1(|5.1-4.9|\div(6.3-4.9)) + 1(|3.5-3.0|\div(3.5-3.0)) + 1(|1.4-1.4|\div(4.7-1.4)) + 1(|0.2-0.2|\div(1.6-0.2)) + 1(0)}{5}$$

$$\approx \frac{0.14 + 1 + 0 + 0 + 0}{5} \approx 0.23$$

$$d(1,\ 3)\ =\ \frac{1(|5.1-6.3|\div(6.3-4.9)) + 1(|3.5-3.3|\div(3.5-3.0)) + 1(|1.4-4.7|\div(4.7-1.4)) + 1(|0.2-1.6|\div(1.6-0.2)) + 1(1)}{5}$$

$$\approx \frac{0.86 + 0.40 + 1 + 1 + 1}{5} \approx 0.85$$

## Exercise 2: Getting K-means and K-medoids work

*Dataset:*

|   | x | y |
|---|---|---|
| A | 2 | 10 |
| B | 2 | 5 |
| C | 8 | 4 |
| D | 5 | 8 |
| E | 7 | 5 |
| F | 6 | 4 |
| G | 1 | 2 |
| H | 4 | 9 |

*Please refer to the Jupyter Notebooks "211110_ml_g1_u02_e02_kmeans.ipynb" and "211110_ml_g1_u02_e02_kmedoids.ipynb" for the code that was used for this exercise.*
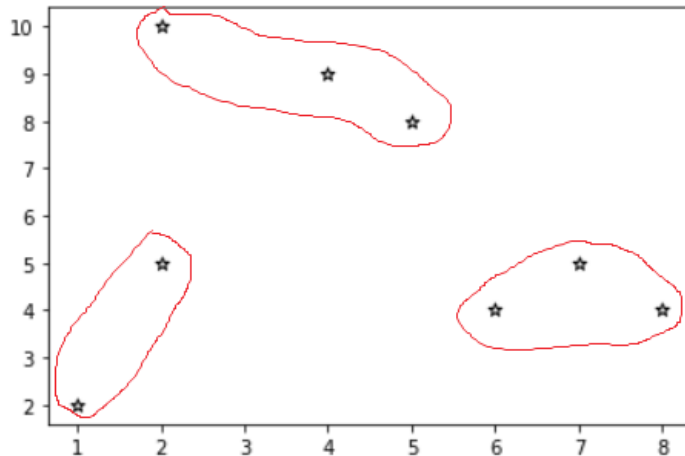
1. There are 2 variations of 3 noticeable groupings that could belong in clusters. Their distance to one another is lesser and as a group, they have a distinct distance and lesser similarity to other groups.

   Cluster proposal 1:
   $C_1 = (1,2), (2,5)$
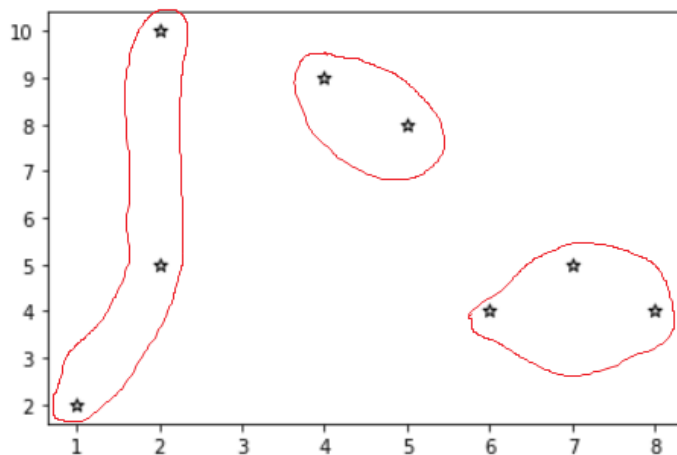   $C_2 = (2,10), (4,9), (5,8)$
   $C_3 = (6,4), (7,5), (8,4)$

   

   Within the clusters, the distance from one point to another seems to be the smallest if the clusters are split in this constellation. Their distance to other clusters is further than their distances to the points in their own clusters.

   Cluster proposal 2:
   $C_1 = (2,10), (2,5), (1,2)$
   $C_2 = (4,9), (5,8)$
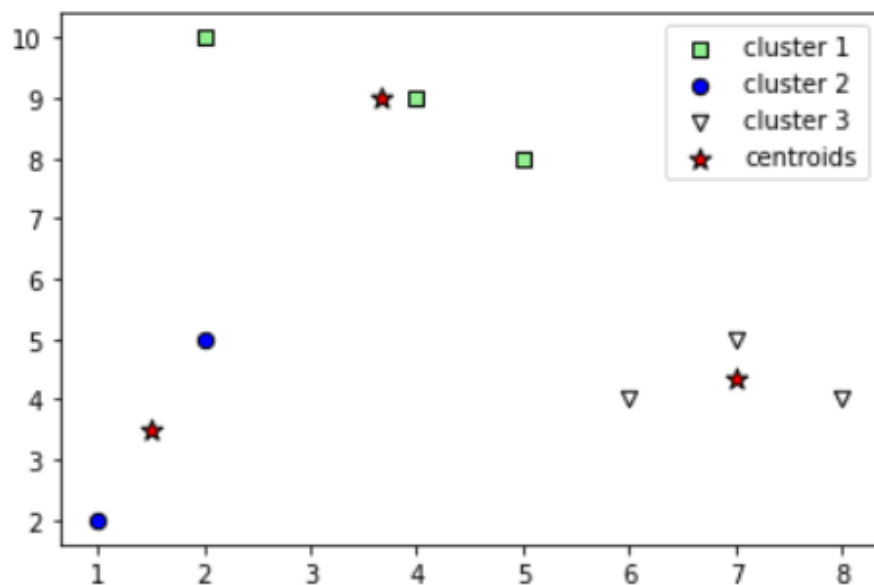   $C_3 = (6,4), (7,5), (8,4)$

   

   In this variation, in $C_1$ there is a dissimilarity in distances, yet the other two clusters seem to contain points that are very similar to one another, which excludes the objects of $C_1$, thus it creates its own category. It's a less likely cluster partition than the one above.

2.  The scatter plot shows the 3 clusters for the point dataset. All objects in a cluster have the same color and shape. They were identified in this constellation as natural clusters in the exercise above.
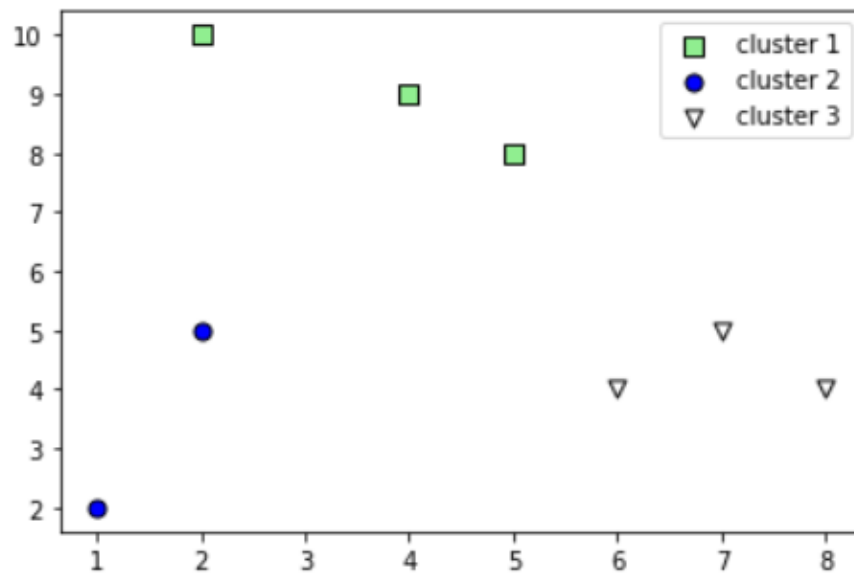


3.  Scatter Plot including the centroids



$$C_1 = (2, 10), (4, 9), (5, 8)$$
$$C_2 = (1, 2), (2, 5)$$
$$C_3 = (6, 4), (7, 5), (8, 4)$$

All of the centroids are artificial points which are positioned in order to satisfy minimal objects' distances. The two data objects in $C_2$ have a seemingly symmetric distance to their centroid. Both $C_1$ and $C_3$ have one point that is closer to the centroid as well as two other points that are further away, both with similar distance to the centroid, however.
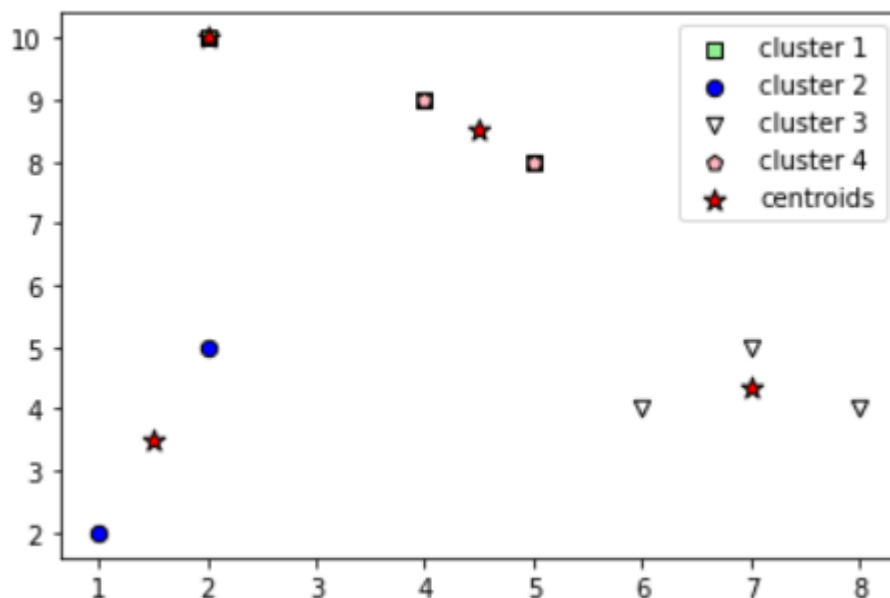
4.  Varying the numbers of clusters

**k = 2**
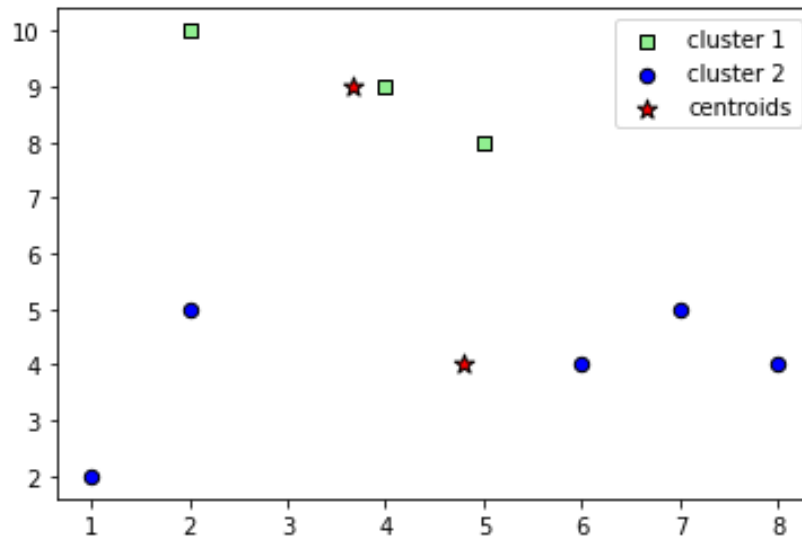


Two clusters are formed where $C_2$ is larger in spread than $C_1$. The centroid is placed artificially in the middle, but it doesn't fulfil the condition that the data points are near enough to their centroids thus the number of clusters needs to be increased.

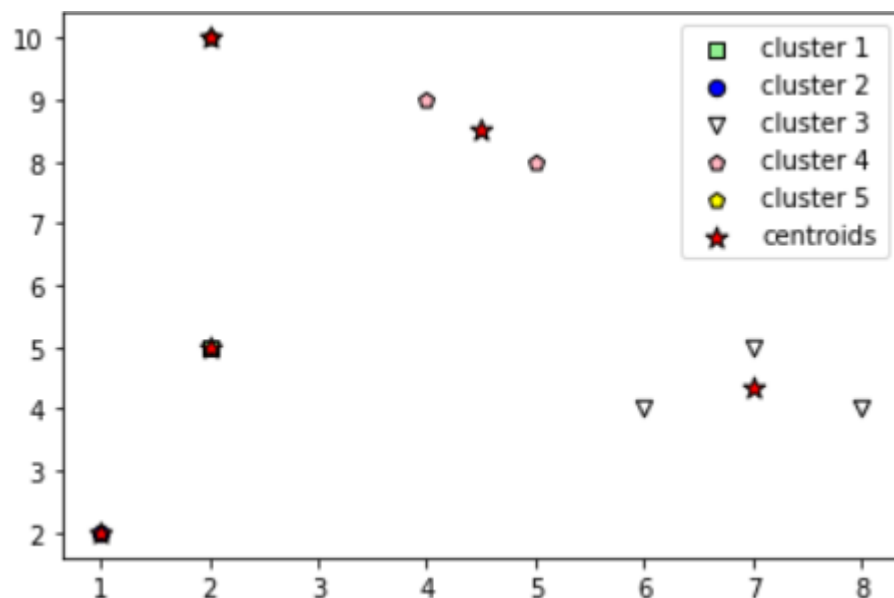Clusters with k > 3

**k = 4**



$$C_1 = (2, 10)$$
$$C_2 = (1, 2), (2, 5)$$
$$C_3 = (6, 4), (7, 5), (8, 4)$$
$$C_4 = (4, 9), (5, 8)$$

The point (2, 10) is now its own cluster $C_1$ (and centroid at the same time). It used to be one with cluster $C_4$, where the centroid was centered to attain proximity to the point. Now, the centroid of $C_4$ has moved to be between the points (4, 9) and (5, 8). $C_2$ as well as $C_3$ have not changed.

**k = 5**



$$C_1 = (2, 5)$$
$$C_2 = (2, 10)$$
$$C_3 = (6, 4), (7, 5), (8, 4)$$
$$C_4 = (4, 9), (5, 8)$$
$$C_5 = (1, 2)$$

With k = 5, another cluster was split to become two new clusters with a single object. The points (1, 2) and (2, 5) have now become their own cluster's centroids.

**Finding k by SSE:**

**Sum of squared errors**
SSE[k]:
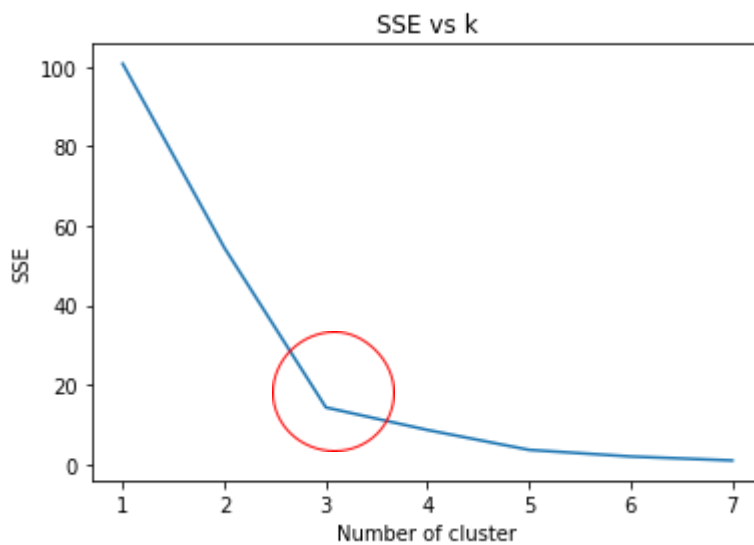
SSE[1]: 100.75
SSE[2]: 51.46666666666667
SSE[3]: 14.333333333333332
SSE[4]: 8.666666666666666
SSE[5]: 3.666666666666667
SSE[6]: 2.0
SSE[7]: 1.0

The optimal count of clusters is determined by plotting SSE vs. k and choosing the point where drop in the curve is the sharpest. According to SSE for the point1.csv, the optimal count of clusters is indeed 3 which corresponds with the estimation above.

5. K-Medoids

As opposed to k-Means, k-Medoids uses actual data objects as cluster centers. The medoids are the most central points in their cluster. With k = 3 the cluster composition remains the same, but the centers have changed as described.

If there are two data points, such as with $C_3$ with its values $(1, 2),\ (2, 5)$, either one or the other data point can randomly be the medoid.
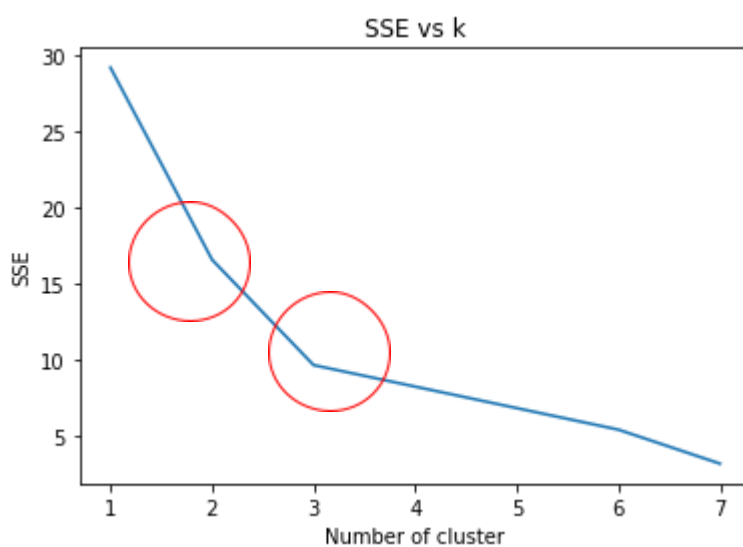
**k = 3**
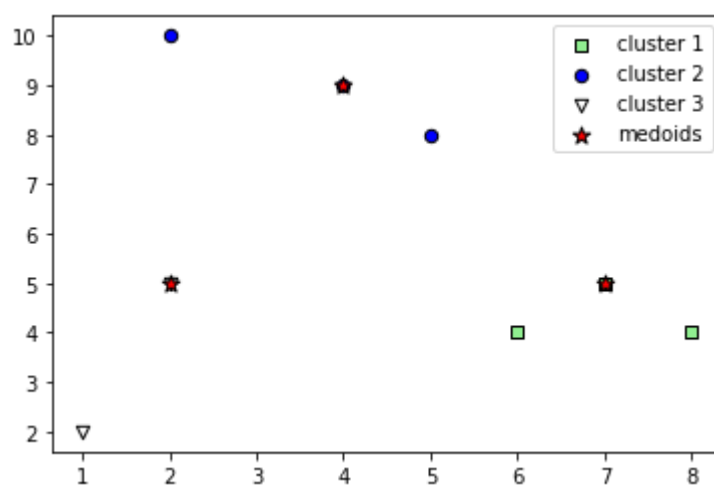
**Sum of squared errors with k-medoids**

SSE[1]: 29.202164976965996
SSE[2]: 16.572765534998144
SSE[3]: 9.640986324787455
SSE[4]: 8.22677276241436
SSE[5]: 6.812559200041264
SSE[6]: 5.39834563766817
SSE[7]: 3.1622776601683795

As expected, the SSE for the same values of k are higher with the k-Medoids as with the k-Means algorithm.



There are two sharp drops at n = 2 and n = 3.

At n = 2, the clusters look as following:
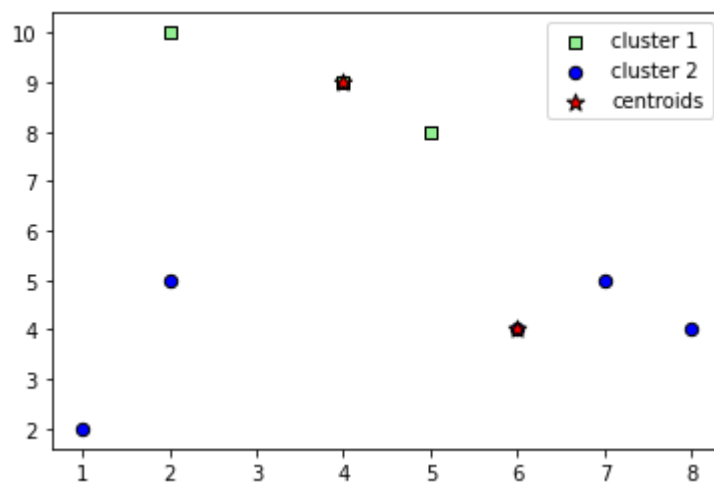


$C_1 = (2, 10), (4, 9), (5, 8)$

$C_2 = (1, 2), (2, 5), (6, 4), (7, 5), (8, 4)$

Even though there is a sharper drop at n= 2, it isn't as obvious on the plot that the points should form a cluster, since the data points on the left are quite remote from the ones on the right.

It depends on the medoid initialization algorithm how the clusters are set. Using the heuristic or pam (partitioning around medoids) approach, it gives the result from above. The heuristic approach forms clusters by the smallest sum distance to every other point. The k-medoids++ approach gives us a different combination of data points in a cluster.
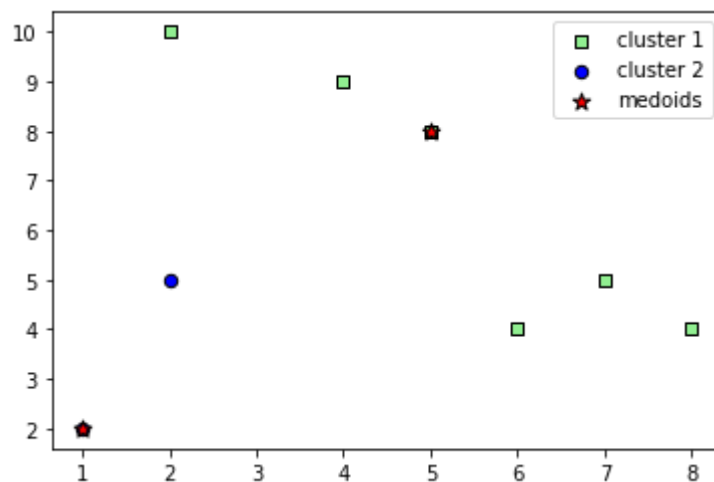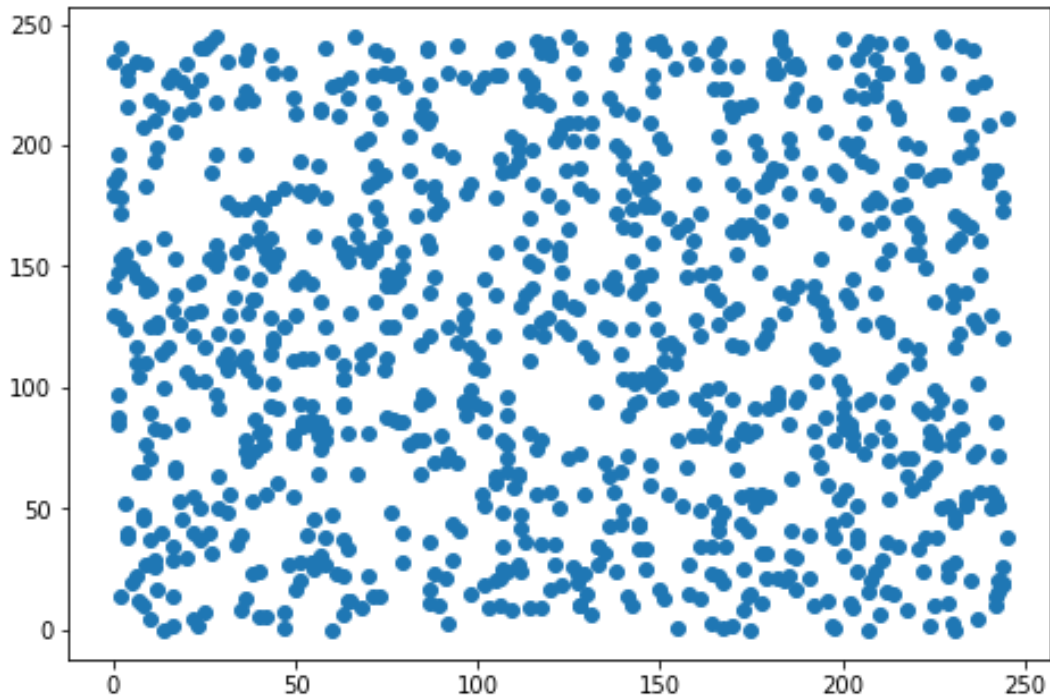


$$C_1 = (2, 10), (4, 9), (5, 8), (6, 4), (7, 5), (8, 4)$$
$$C_2 = (1, 2), (2, 5)$$

**Exercise 3: Random repartition of the data**

*Please refer to the Jupyter Notebook "211110_ml_g1_u02_e03.ipynb", as well.*

Scatter plot of 1000 points with random values. There are no natural clusters visible.



SSE with the k-means algorithm and k values ranging from 2 to 100. It has a rectangular hyperbola shape with diminishing gains in lower SSE after k = 10. Therefore we think 10 is an optimal number of clusters with a balance of good SSE and computing time.
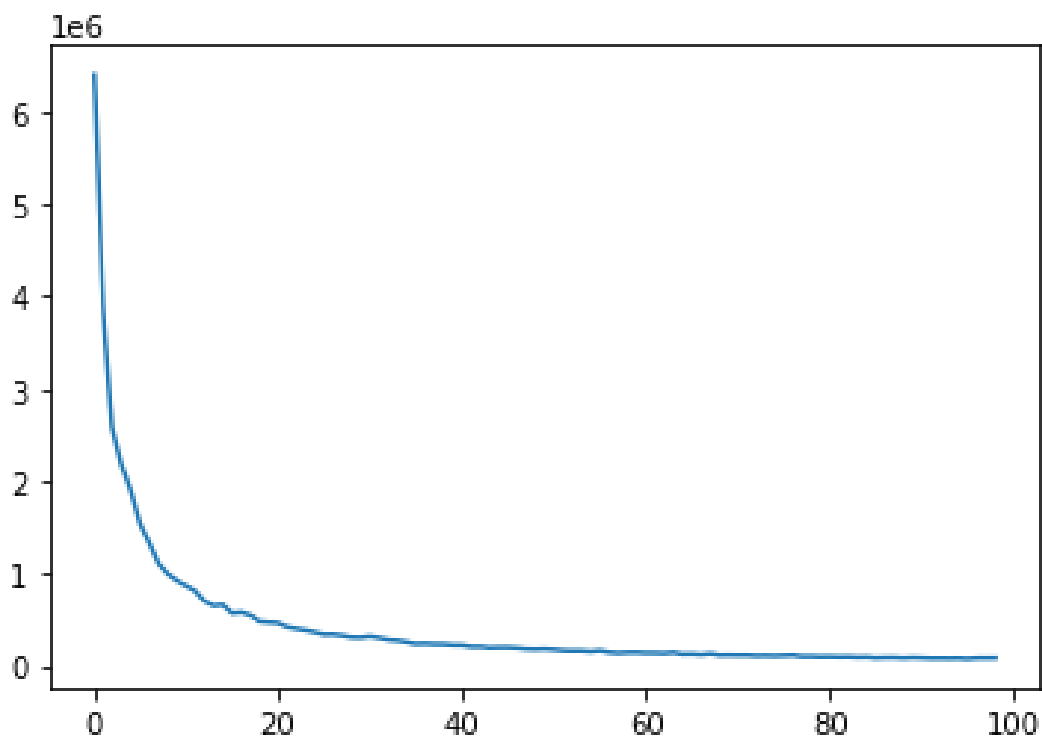
Clusters with k = 10 (Centroids marked with cross)
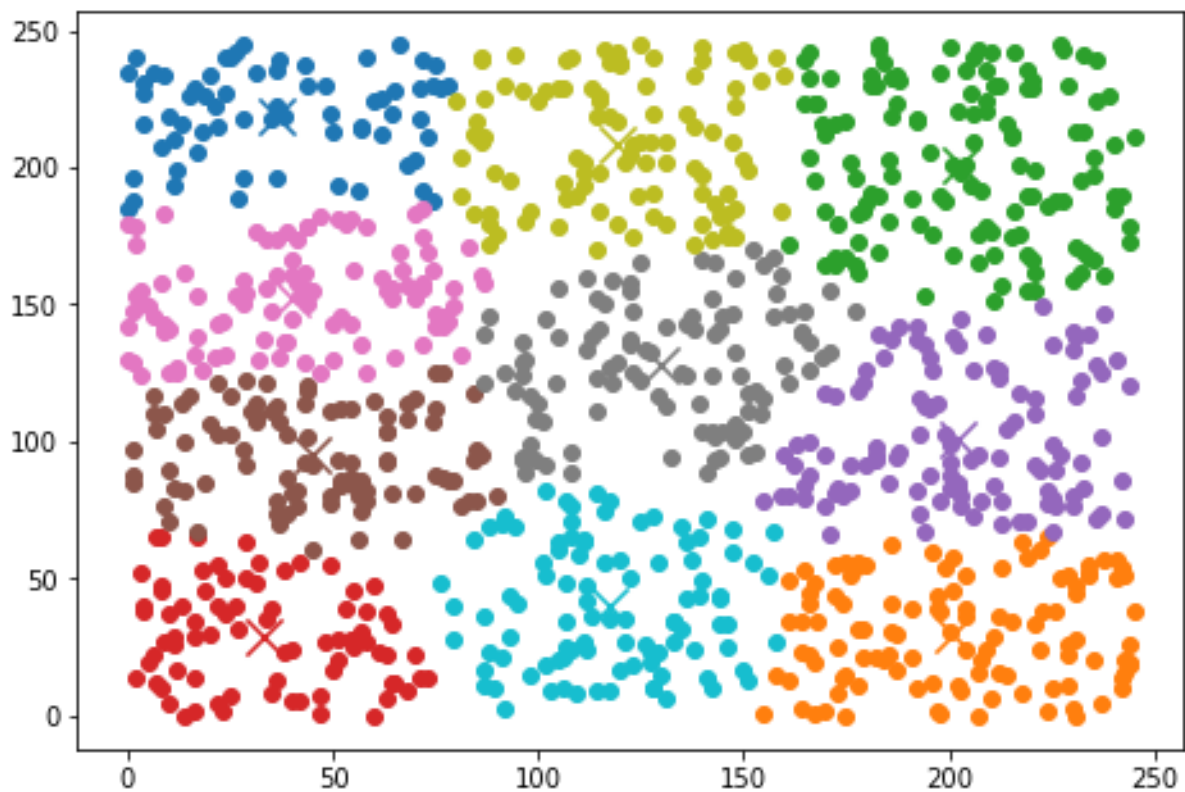


Calculating the Hopkins Statistic with 100 random and uniform samples we got a factor of approximately 0.489, which is very close to 0.5, indicating the data set "is uniformly distributed and thus contains no meaningful clusters" (cp. book p. 486). This test should be iterated to test the homogenous hypothesis.

**Exercice 4: Clustering the Iris dataset**

1. ***Read the data.*** *Note RapidMiner: the attribute* `class` *should have the role* `Label` *so that it is ignored by a clustering algorithm.*

| Label | | | Least | Most | Values |
|---|---|---|---|---|---|
| ⌄ **class** | Nominal | 0 | Iris-virginica (50) | Iris-setosa (50) | Iris-setosa (50), Iris-versicolor (50), ...[1 more] |

2. ***Report the summary statistics.***

| | | | | Min | Max | Average | Deviation |
|---|---|---|---|---|---|---|---|
| ⌃ sepallength | Real | 0 | | 4.300 | 7.900 | 5.843 | 0.828 |
| ⌃ sepalwidth | Real | 0 | | 2 | 4.400 | 3.054 | 0.434 |
| ⌃ petallength | Real | 0 | | 1 | 6.900 | 3.759 | 1.764 |
| ⌃ petalwidth | Real | 0 | | 0.100 | 2.500 | 1.199 | 0.763 |

| | Petal length | Petal width | Sepal length | Sepal width |
|---|---|---|---|---|
| iris-setosa | Min: 1<br>Median: 1.5<br>Max: 1.9 | Min: 0.1<br>Median: 0.2<br>Max: 0.6 | Min: 4.3<br>Median: 5<br>Max: 5.8 | Min: 2.3<br>Median: 3.4<br>Max: 4.4 |
| iris-versicolor | Min: 3<br>Median: 4.35<br>Max: 5.1 | Min: 1<br>Median: 1.3<br>Max: 1.8 | Min: 4.9<br>Median: 5.9<br>Max: 7 | Min: 2<br>Median: 2.8<br>Max: 3.4 |
| iris-virginica | Min: 4.5<br>Median: 5.55<br>Max: 6.9 | Min: 1.4<br>Median: 2<br>Max: 2.5 | Min: 4.9<br>Median: 6.5<br>Max: 7.9 | Min: 2.2<br>Median: 3<br>Max: 3.8 |

3. ***Explore your data with different visualizations.*** *From this exploration, which assumption can you make concerning the four attributes to predict the class of an iris flower? Explain your answer.*

We can already see from the import, that the data set is partitioned into 3 equal parts, each containing 50 elements.
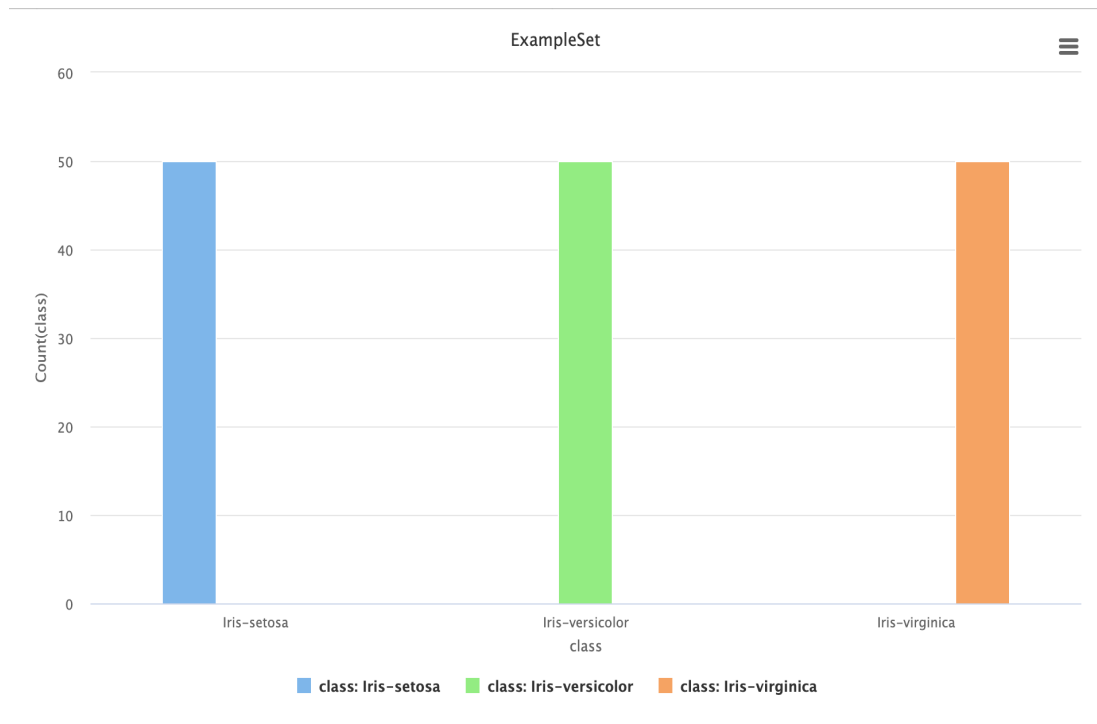
Through the visualization of Boxplots and Scatterplots, we can assume that both petallength and petalwidth attributes are more relevant for the prediction of an Iris flower.

Indeed we can see through the visualizations, and by selecting these two attributes in particular, that the distribution of the flowers is more conclusive.

In the visualisation below it shows that the iris-setosa is unique when looking at its petal length and width, and can't be classified as another type of iris flower.

We can also assume that an iris flower with a petal length less than 4.5 and petal width less than 1.4 can be an iris-versicolor, but higher than these two values it can be an iris-versicolor as it can also be an iris-virginica.

This means that we can predict the type of the iris flower.
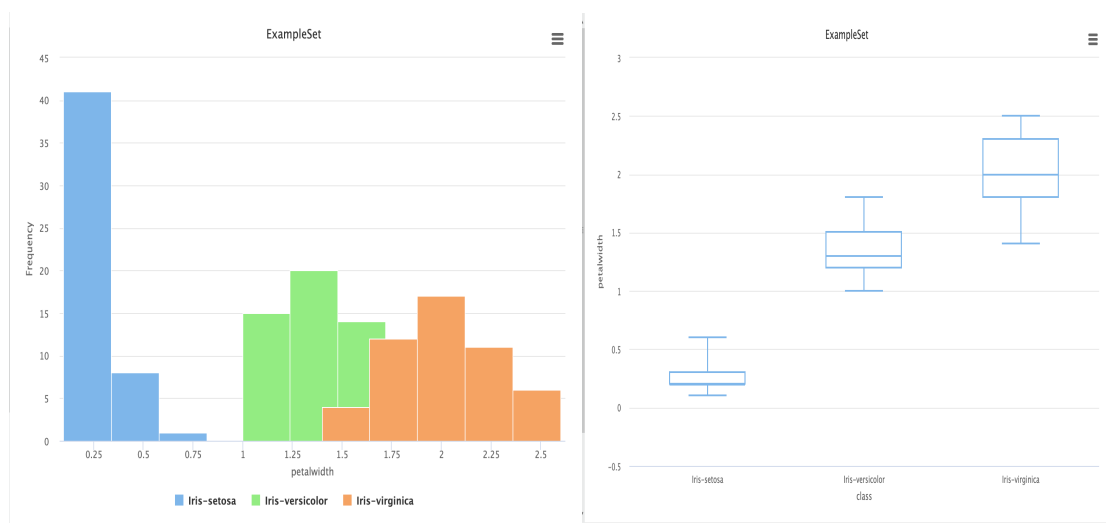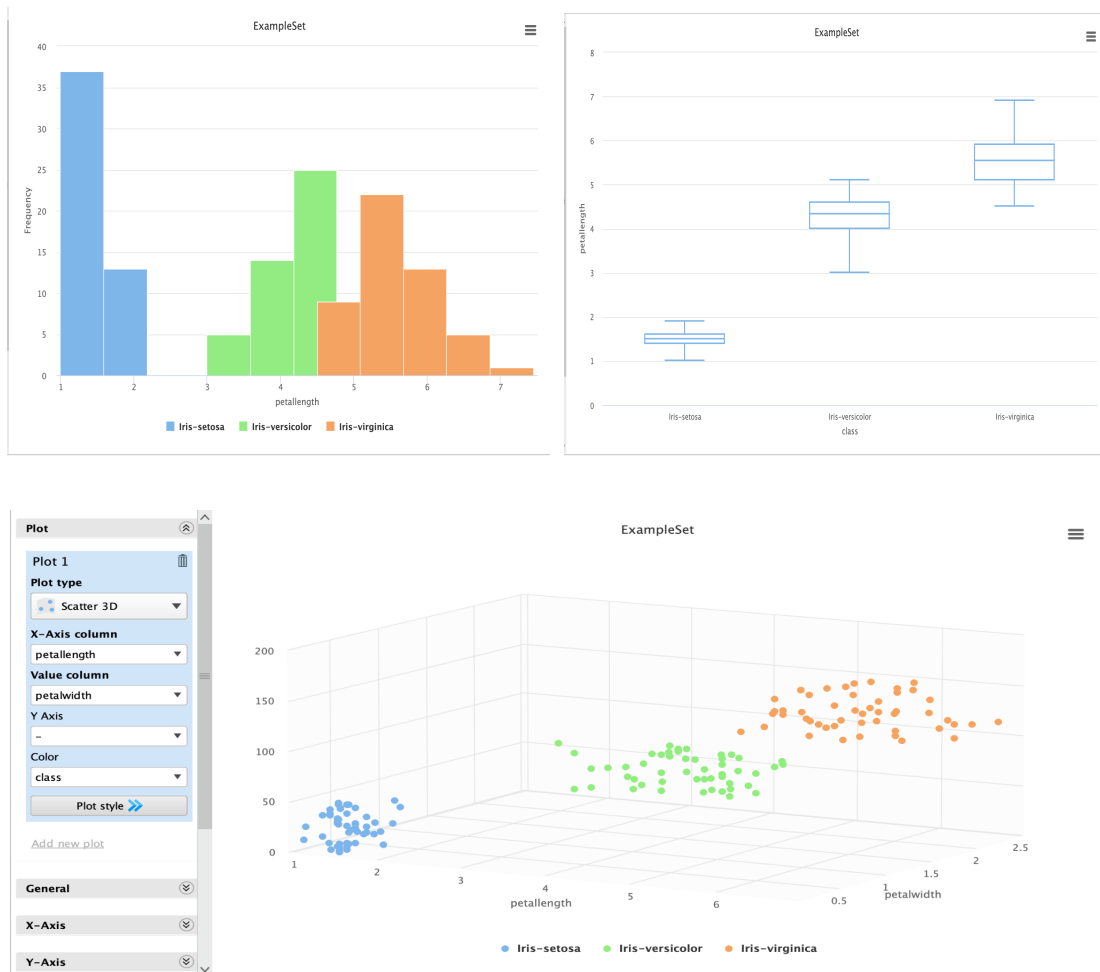
However, when looking at the two other attributes, sepal length and sepal width, we can clearly see and assume that those two attributes are not really helpful for the prediction of the type of iris flower.

Indeed, when we have a close look at the boxplot or at the scatterplot, we see that compared with those two attributes, an iris flower could be from any type.

Thus, we cannot differentiate between the different types of the flower.
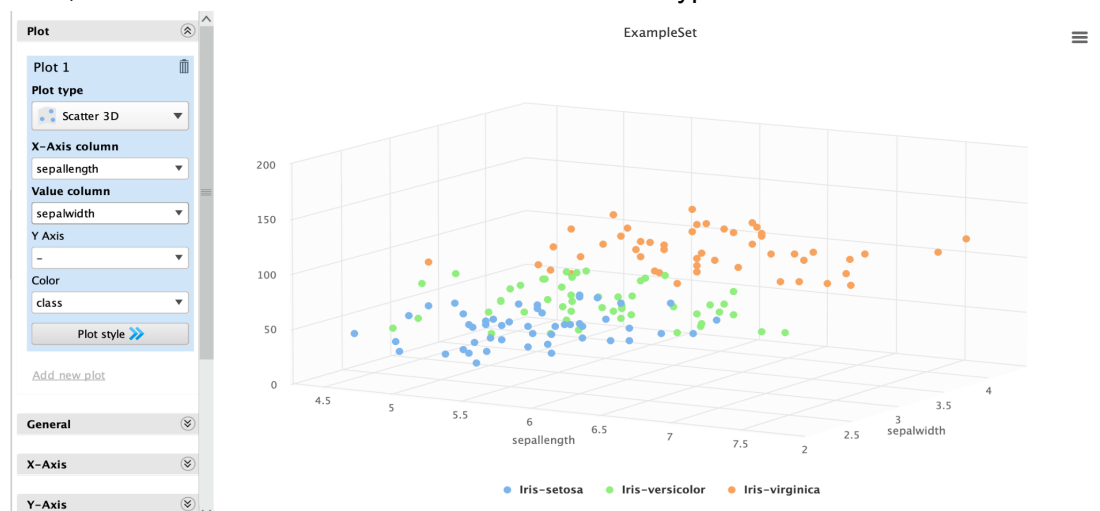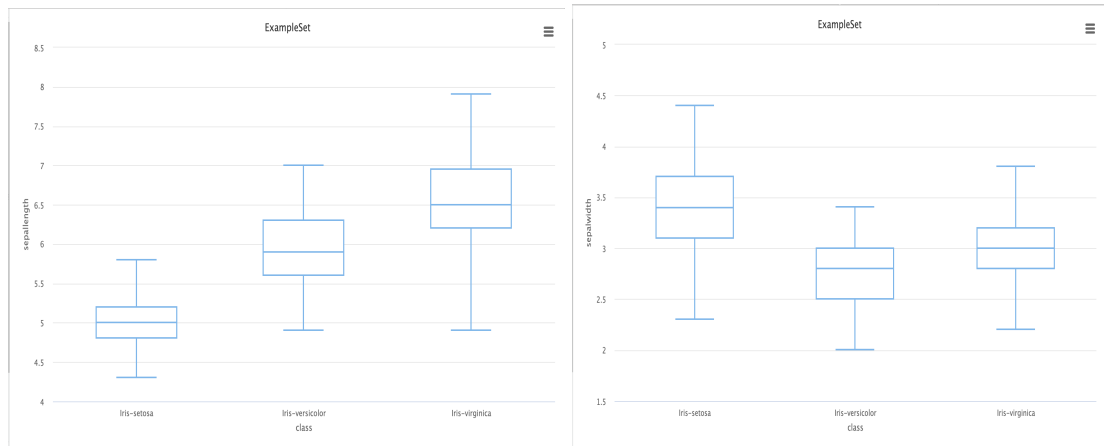
4. **Clustering as classification: Cluster the data into three clusters (you should try out different clustering algorithms and different attribute selection / transformation) and report the accuracy (proportions of objects in the right clusters) for each algorithm and each selection / transformation of attributes that you have used. This kind of evaluation is called extrinsic: we have got the class as "ground truth" to compare the clusters with.**
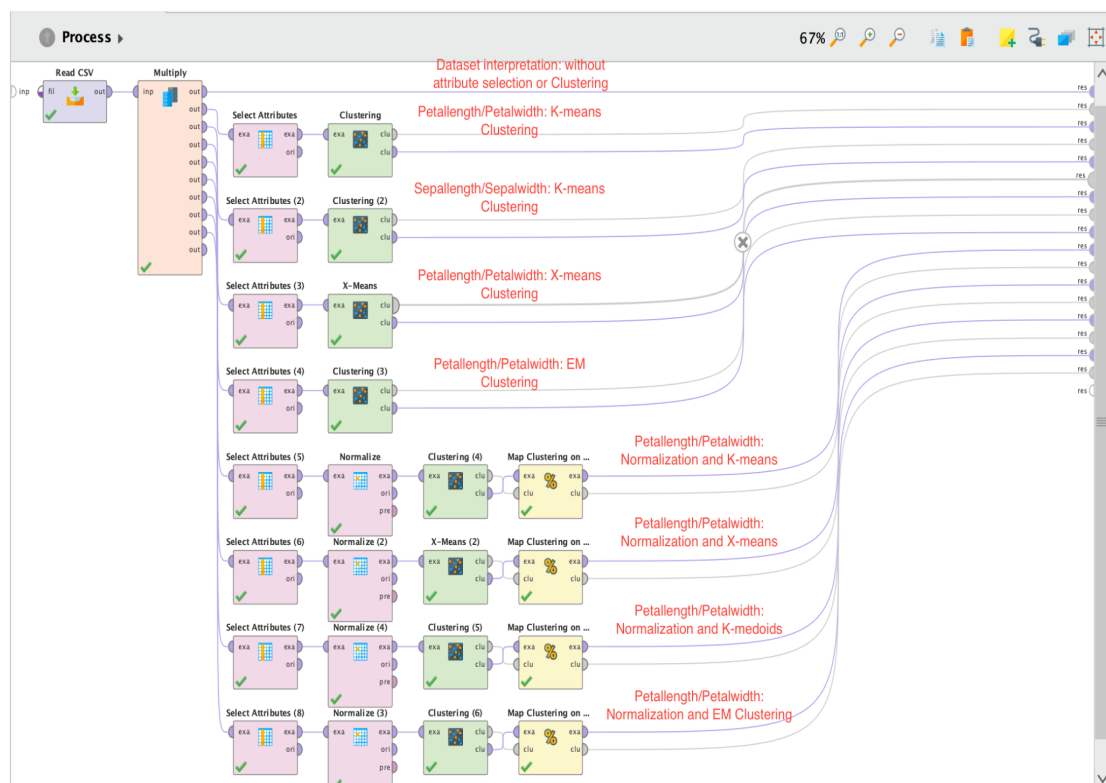
- **Case 1 & 2: K-means clustering with attribute selection**

   When looking at the result, we can already confirm our assumption from the previous question, that selecting (petal length / petal width) is more accurate than selecting (sepal length / sepal width). This also applies to any type of clustering.

Petal length and petal width
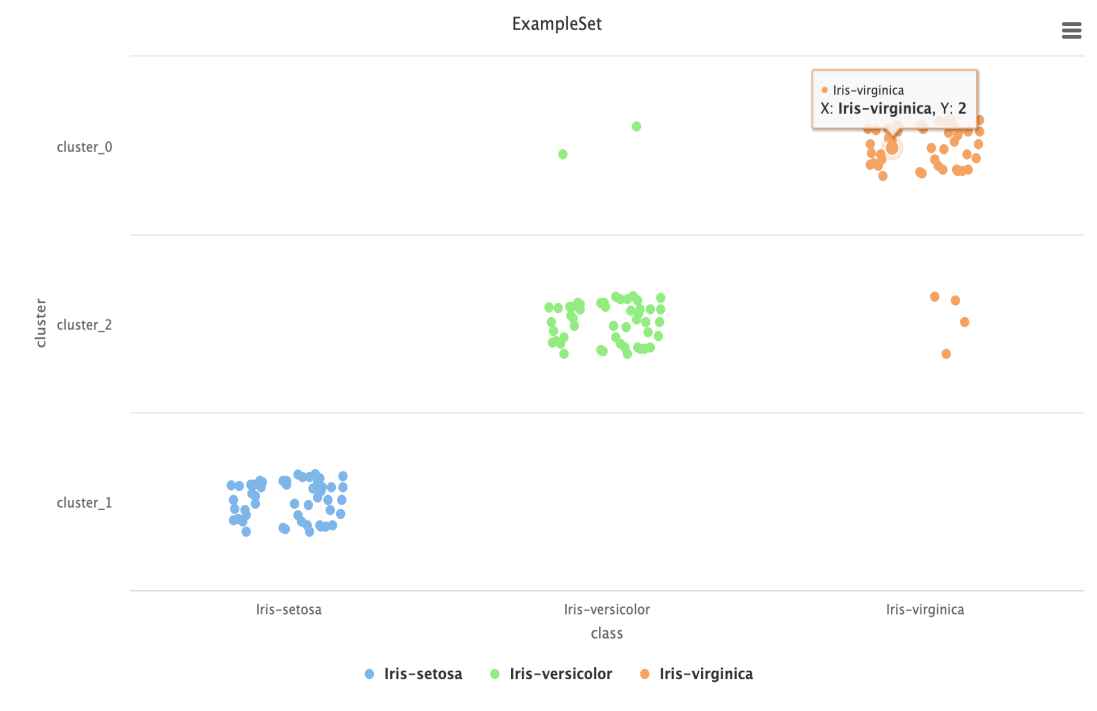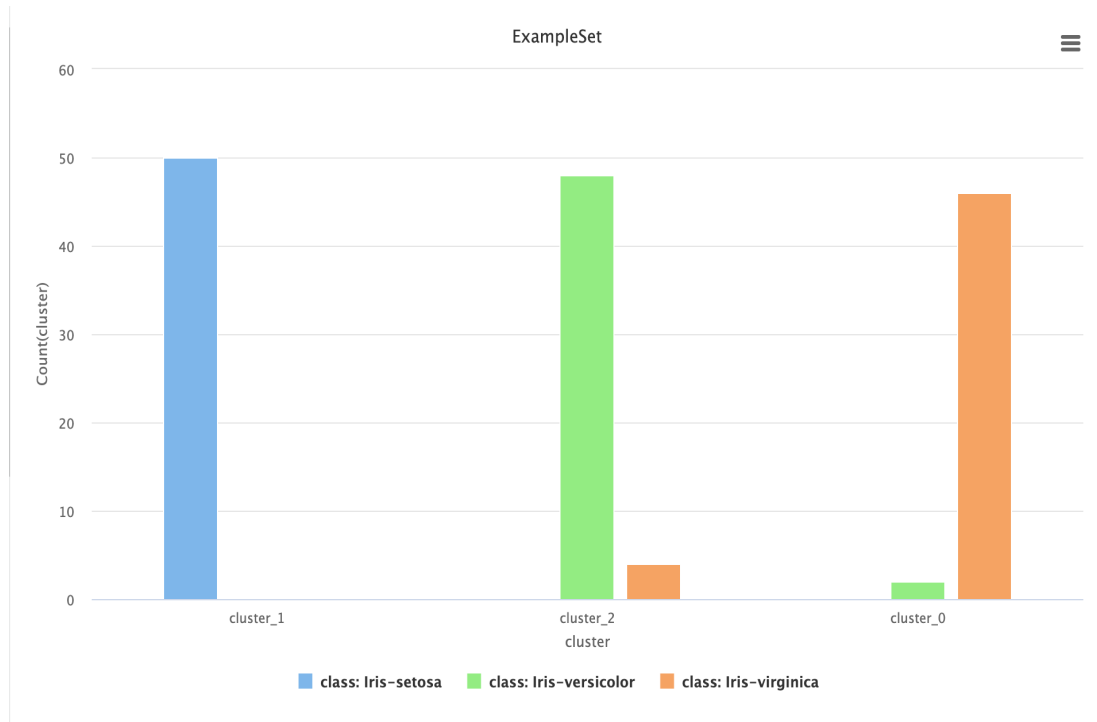
## Sepal length and sepal width
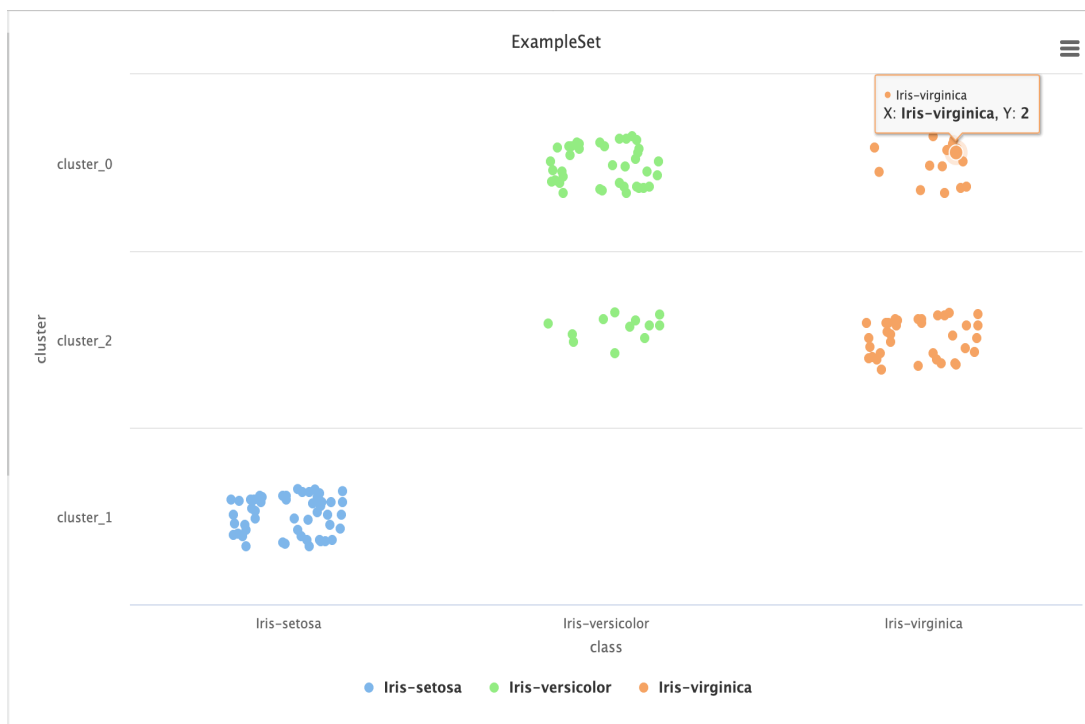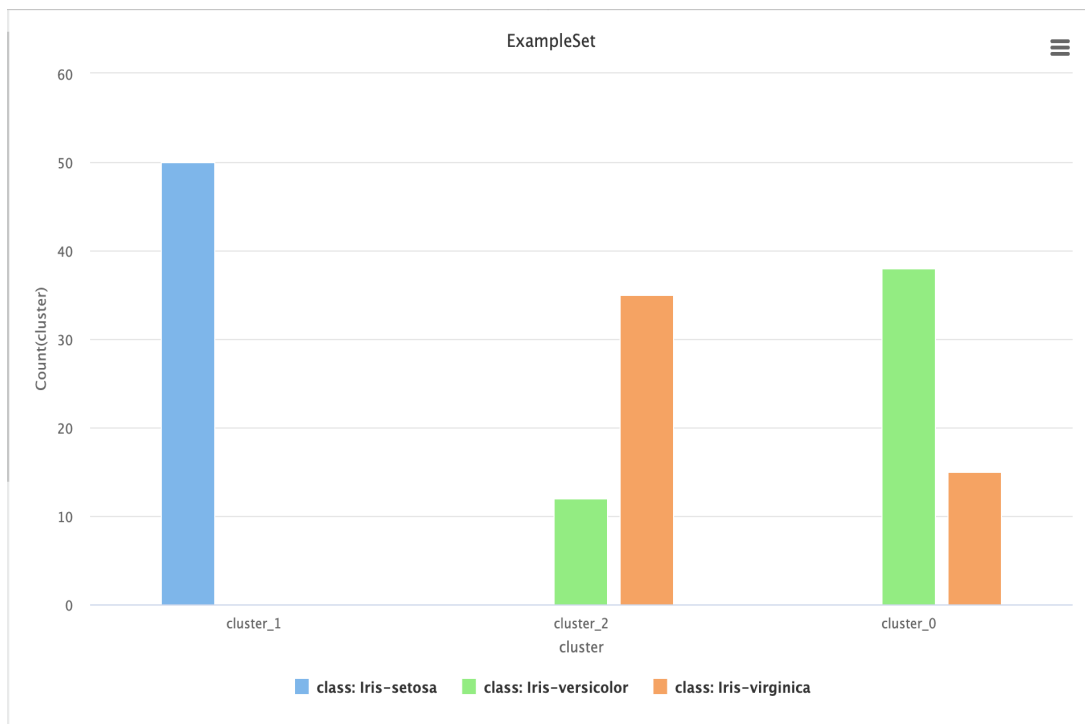
- **Case 3 & 4: X-means / EM clustering with attribute selection (petal length:petal width)**
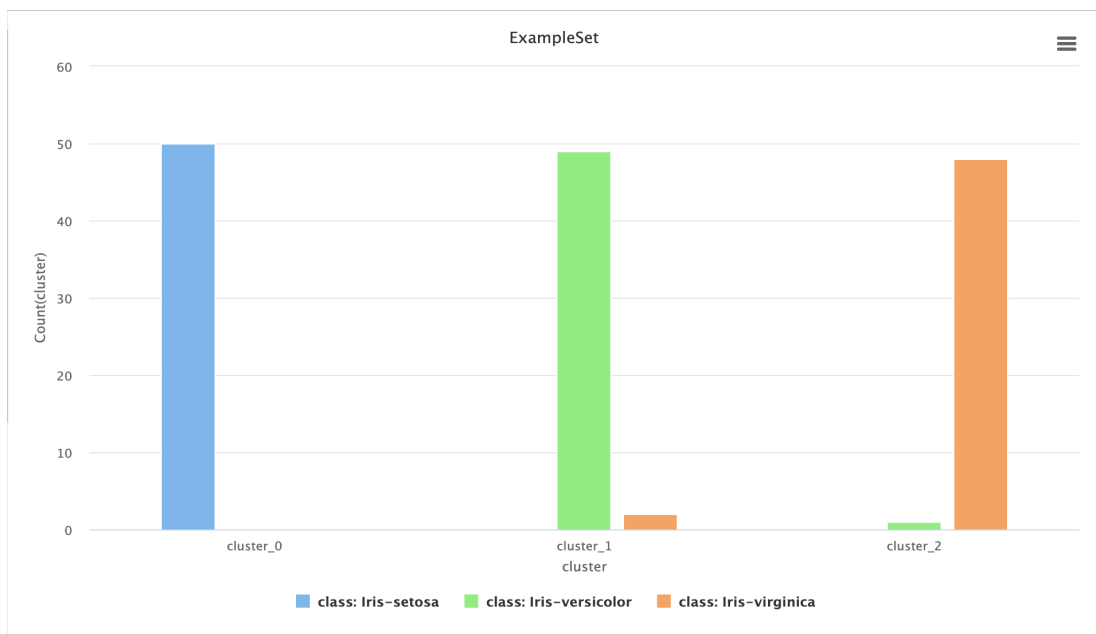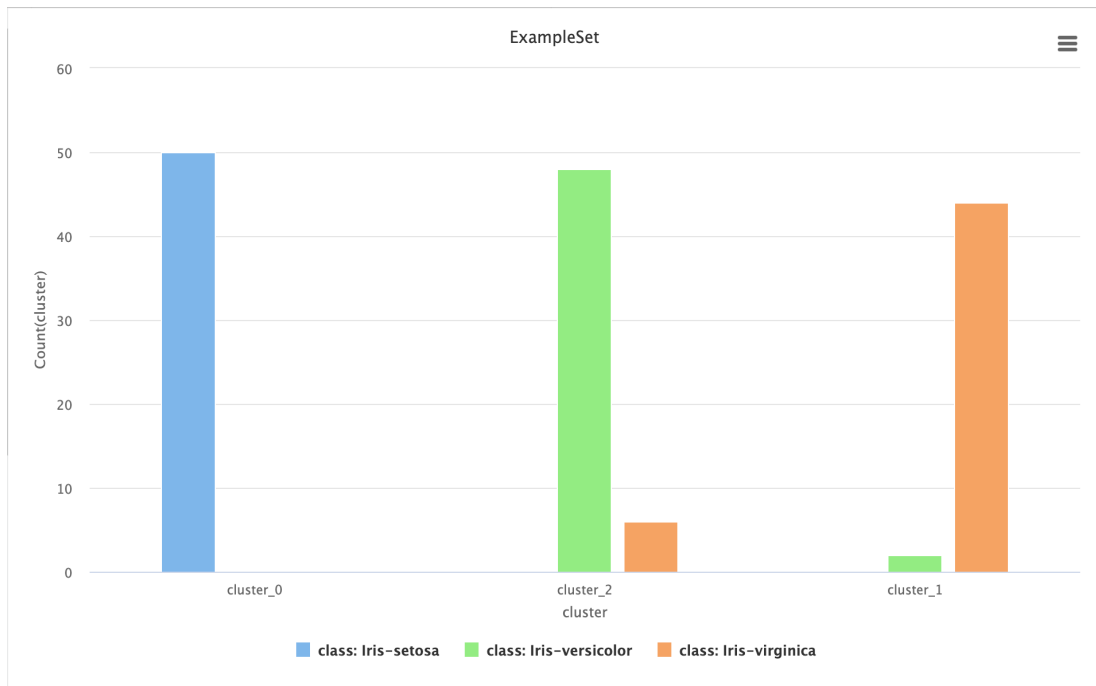
Looking at the same pair of attributes (petal length:petal width) with different clustering algorithms, we can see for our cases that the EM clustering is more performant than K-means or X-means.

Now that we selected our more accurate attributes, we can try different pairs of normalization and clustering algorithms.

⇒ Normalization is used to scale values so they fit in a specific range. Adjusting the value range is very important when dealing with Attributes of different units and scales.

- **Case 5: Normalization / K-means clustering**
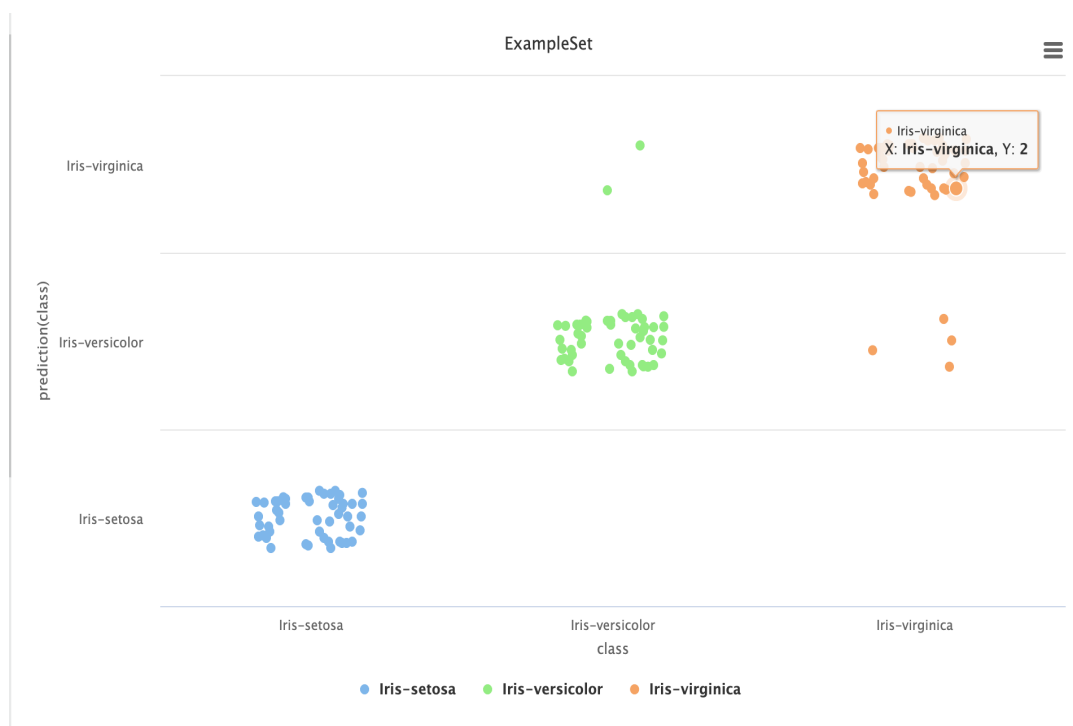
- **Case 6: Normalization / X-means clustering**



| | | | | Least | Most | Values |
|---|---|---|---|---|---|---|
| ⌄ Label **class** | | Polynominal | 0 | Iris-virginica (50) | Iris-setosa (50) | Iris-setosa (50), Iris-versicolor (50), Iris-virginica (50) Details... |
| | | | | Open visualizations | | |
| ⌄ Prediction **prediction(class)** | | Polynominal | 0 | Iris-virginica (48) | Iris-versicolor (52) | Iris-versicolor (52), Iris-setosa (50), Iris-virginica (48) Details... |
| | | | | Open visualizations | | |
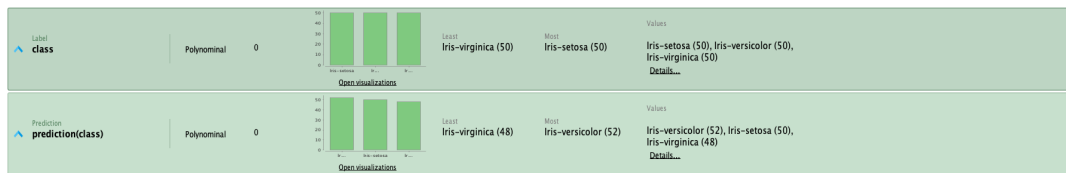


⇒ K-means and X-means Clustering give the same result when using normalization.

## Cluster Model

```
Cluster 0: 52 items
Cluster 1: 50 items
Cluster 2: 48 items
Total number of items: 150
```

## Cluster Model

```
Cluster 0: 50 items
Cluster 1: 48 items
Cluster 2: 52 items
Total number of items: 150
```

- **Case 7: Normalization / K-medoids clustering**



⇒ K-medoids are not performant compared to k-means and K-means algorithms.



**Cluster Model**

```
Cluster 0: 43 items
Cluster 1: 57 items
Cluster 2: 50 items
Total number of items: 150
```

- **Case 8: Normalization / EM clustering**

⇒ EM Clustering algorithm is the most performant compared to the other Clustering's algorithms.

## Cluster Model

```
Cluster 0: 50 items
Cluster 1: 51 items
Cluster 2: 49 items
Total number of items: 150

--------------------------------------------------------

cluster probabilities:
Cluster 0: 0.33333294405516917
Cluster 1: 0.34099593970199354
Cluster 2: 0.32567111624283723

cluster means:
Cluster 0: -1.3005217092420223; -1.2509381718144315
Cluster 1: 0.29991658535146803; 0.17892975708648792
Cluster 2: 1.0170886387363205; 1.0930185860207295

cluster covariance matrices:
Cluster 0:
0.009477089825815221      0.0041468906007055325
0.0041468906007055325     0.019340104960167605
Cluster 1:
0.07762789599283748       0.059045179203929966
0.059045179203929966      0.07122781436667235
Cluster 2:
0.09933105090680239       0.037416270093988964
0.037416270093988964      0.1258638900536604
```

**5.** *What is your answer:* **are the four attributes of this set, namely sepallength, sepalwidth, petallength and petalwidth are appropriate to determine the class of an iris?**

For the case of using all four attributes, the clustering's algorithms will not be performant and we will get a gap between what we are expecting and what was predicted, so in this case, using all attributes is not appropriate to determine the class of an iris.

However, in the case of selecting the right attributes, in our case (petal length:petal width), it is possible to determine the class of an iris.

⇒ The accuracy of the prediction will eventually depend on the clustering algorithm used.

**Exercise 5: Wrap-up**

*You are given a dataset and are asked whether clusters can be identified in the data. List the methodological steps that you should undertake to solve this task.*

We assume the set has already been cleaned from missing, erroneous, duplicate or obsolete data.

1. **Confirm data is metric (quantitative)**
   While one can cluster data even if features are not metric, many of the statistical methods available for clustering require that the data are so. This means not only that all data are numbers, but also that the numbers have an actual numerical meaning, that is, 1 is less than 2, which is less than 3 etc. The main reason for this is that one needs to define distances between observations, and often distances (e.g. the "Euclidean distance") are defined only with metric data.
2. **Scale the Data (Normalization)**
   Normalizing the values of a given attribute allows for easier application of clustering algorithms and makes the features (respectively their distances) comparable, thus allows for clustering based on more than one attribute
3. **Select attributes**
   The decision which attributes should be used for clustering is critically important, that will have a big impact on the clustering solution. So we need to think carefully about the attributes we will choose for clustering.
   Moreover, we often use only a few of the data attributes for segmentation (the segmentation attributes) and use some of the remaining ones (the profiling attributes) only to profile the clusters.
4. **Define similarity measure**
   The goal of clustering and segmentation is to group observations based on how similar they are. It is therefore crucial that we have a good understanding of what makes two observations similar.
   Most statistical methods for clustering and segmentation use common mathematical measures of distance. Typical measures are, for example, the Euclidean distance or the Manhattan distance.
5. **Visualize pairwise distances**
   Having defined what we mean with "two observations are similar", next is to get a first understanding of the data through visualizations like scatter plots and box plots
6. **Method and number of Segments**
   There are many statistical methods for clustering and segmentation. In practice one may use various approaches and then eventually select the solution that is statistically robust, interpretable, and actionable — among other criteria.
   (e.g. K-means, EM, Hierarchical Clustering Method, etc.)
7. **Interpret the segments**
   Having decided (for now) how many clusters to use, we would like to get a better understanding of those clusters and interpret the segments.
   To this purpose, one needs to spend time visualizing and understanding the data within each of the selected segments.