Assignment 2-1  \textit{Object oriented programming I}

We deal now with object oriented programming in Python. For this purpose perform the following steps:

1. Write a \textit{Point} class. A \textit{Point} class takes an $x$ and an $y$ coordinate as an argument.

2. Further this class shall have a setter method \textit{setXY} which takes an $x$ and $y$ coordinate and sets the attributes to the new provided values.

3. The class shall also have a getter method \textit{getXY} which returns the current $x$ and $y$ coordinates of the Point.

4. Write a method \textit{distance} which takes another \textit{Point} object and returns the euclidean distance between the provided Point and the Point itself.

   \textit{Hint: Take \texttt{import math} to use \texttt{math.sqrt(somevalue)} in order to compute the square root.}

Assignment 2-2  \textit{Object oriented programming II}

In a next step the task is to create a class \textit{Shape}. For this purpose perform the following steps

1. Create a class \textit{Shape} which takes a name and a color as parameters.

2. Define a method \textit{area} which just returns 0.0.

3. Define a method \textit{perimeter} which just returns 0.0.

Now create a class \textit{Rectangle} which \texttt{inherits} from \textit{Shape} and in which you \texttt{implement} the \textit{area} and \textit{perimeter} methods.
Assignment 2-3  Pandas

For this assignment, we will use the file moviemetadata.csv, which contains entries from the IMDB movie database. The original source of the data is Kaggle: https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset/. Please also consider to consult the documentation http://pandas.pydata.org/pandas-docs/stable/ if needed. Solve the following tasks:

1. Read the csv file as a DataFrame for further processing using pandas.read_csv().
2. Inspect the read csv file using .shape, .columns, .info and .describe().
3. Display the first five records of the data set using .head(5) and the last five records using .tail(5).
4. Select from the data set the first five records. Those records shall only contain the following columns: movie.title, duration and num_voted_users.
5. Select the first five movies containing the genre 'Action'. Display only the columns movie.title and genres.
6. Sort the action movies by their 'imdb_score' and display the names and scores the top-10 scored movies.
7. Group the movies by column 'director' and display the top-10 directors with the highest mean gross of their movies.
8. Optional: Delete all rows, which contain at least one missing value. Visualize parts of the data using pandas.plotting.scatter_matrix and DataFrameGroupBy.hist.

Assignment 2-4  Numpy I - some basic functions

In this assignment you will become familiar with the numpy library and some of its basic functionality. Please also consider to consult the documentation https://docs.scipy.org/doc/numpy-dev/index.html if needed. Solve the following tasks:

1. Create an numpy array of floats containing the numbers from 0 to 4.
2. Create the following matrix as a numpy matrix:

   \[
   M = \begin{bmatrix}
   1 & 2 & 3 \\
   4 & 5 & 6 \\
   \end{bmatrix}
   \]

3. Get the shape of the matrix M.
4. Check if the value 2 is in M.
5. Given the array \( a = \text{np.array}([0,1,2,3,4,5,6,7,8,9], \text{np.float32}) \). Reshape it to an 5x2 matrix.
6. Transpose the previously introduced matrix M.
7. Flatten matrix M.
8. Given the array \( b = \text{np.array}([0,1,2,3], \text{np.float32}) \). Increase the dimensionality of b.
9. Create an 3x3 identity matrix.
Assignment 2-5  

**Numpy II - linear algebra and statistics**

This assignment has its focus on numpy function of the linear algebra and statistics domain. Solve the following tasks using numpy:

1. Given the following two numpy arrays:
   
   \[
   a = np.array([1,2,3], np.float32), \quad b = np.array([4,5,6], np.float32)
   \]

   Compute the dot product of \(a\) and \(b\).

2. Given the following matrix \(M\):

   \[
   M = \begin{bmatrix}
   1 & 2 & 3 \\
   4 & 5 & 6 \\
   7 & 8 & 9 
   \end{bmatrix}
   \]

   Compute the determinant of \(M\) by using the `linalg` package of the numpy library.

3. Compute the eigenvalues and eigenvectors of \(M\).

4. Compute the inverse of \(M\).

5. Given the numpy array \(c = np.array([1,4,3,8,3,2,3], np.float32)\), compute the mean of \(c\).

6. Using \(c\), compute the median.

7. Given the following matrix

   \[
   C = \begin{bmatrix}
   1 & 1 \\
   3 & 4 
   \end{bmatrix}
   \]

   Compute the covariance of \(C\).

Assignment 2-6  

**Matplotlib + k-Means**

In this exercise, we will implement a \(k\)-means clustering algorithm.

1. Load the dataset `blobs.csv` and visualize it using `matplotlib.pyplot.scatter`.

2. Implement a function `kmeans(data,k)`.

3. Optional: Visualize intermediate results after each iteration.

4. Apply your method to the blobs dataset using different values for \(k\) and plot the results.

5. Load the dataset `mouse.csv` and visualize it. Apply your method to the mouse dataset as well and discuss the differences.