

Задание 1.

1). На языке программирования Python написать программу вывода простейшего поточечного графика.

Воспользоваться можно любой из библиотек:

- с использованием встроенных возможностей Pandas - **df.plot(kind='line')**
- посредством Matplotlib (plt - псевдоним matplotlib.pyplot) - **plt.plot(df.index, df.income)**
- посредством Seaborn (sns - псевдоним) - **sns.lineplot(x=df.index, y=df.income)**

Обосновать выбор библиотеки, сравнив возможности каждой из них.

2). Набор данных для построения графика необходимо считать из внешнего файла. Внешний файл определяется пользователем и может быть любым: текстовым, табличным. В одном файле могут находиться несколько наборов данных.

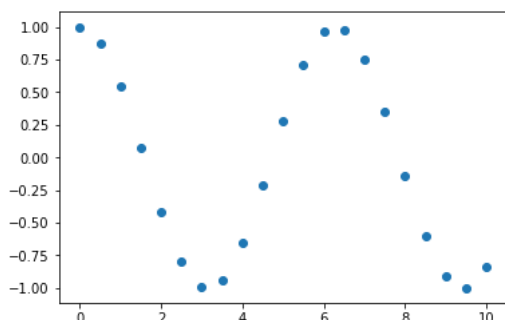
Данные в файле заполняются самостоятельно, но не менее 5-ти пар вещественных точек (x,y).

3). Для отображения точечного графика в Python предназначена функция **scatter()**. В простейшем виде в Matplotlib точечный график можно получить передав функции **scatter()** наборы точек для (x,y) координат, например, функции **y=cos(x)**:

```
x = np.arange(0, 10.5, 0.5)
```

```
y = np.cos(x)
```

```
plt.scatter(x, y)
```



Здесь **NumPy** или — **np** — это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами.

Для более детальной настройки отображения можно воспользоваться дополнительными параметрами функции *scatter()*, сигнатура ее вызова имеет следующий вид:

scatter(x, y, s=None, c=None, marker=None, cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, verts=None, edgecolors=None, *, plotnonfinite=False, data=None, **kwargs)

Некоторые из параметров функции *scatter()*:

Параметр	Тип	Описание
<i>x</i>	<i>array_like, shape (n,)</i>	Набор данных для оси абсцисс
<i>y</i>	<i>array_like, shape (n,)</i>	Набор данных для оси ординат
<i>s</i>	<i>scalar</i> или <i>array_like, shape (n,), optional</i>	Масштаб точек
<i>c</i>	<i>color, sequence, или sequence of color, optional</i>	Цвет
<i>marker</i>	<i>MarkerStyle, optional</i>	Стиль точки объекта
<i>cmap</i>	<i>Colormap, optional, default: None</i>	Цветовая схема
<i>norm</i>	<i>Normalize, optional, default: None</i>	Нормализация данных
<i>alpha</i>	<i>scalar, optional, default: None</i>	Прозрачность
<i>linewidths</i>	<i>scalar или array_like, optional, default: None</i>	Ширина границы маркера
<i>edgecolors</i>	<i>{'face', 'none', None}</i> или <i>color</i> или <i>sequence of color, optional.</i>	Цвет границы

Функция **arange()** возвращает одномерный массив с равномерно разнесенными значениями внутри заданного интервала. В общем виде:

numpy.arange([start,]stop, [step,]dtype=None)

Параметры:

start - число (необязательный)

Число, которое является началом интервала. По умолчанию равно 0.

stop - число

Определяет конец интервала, при этом сам интервал это значение не включает. Если шаг (*step*) не является целым числом, то в некоторых случаях, из-за округление вещественных чисел указанное значение *stop* может оказаться включенным в интервал.

step - число (необязательный)

Определяет интервал между значениями. По умолчанию равен 1. Если данный параметр указан, то значение *start* так же необходимо указать.

dtype - тип данных NumPy (необязательный)

Определяет тип данных выходного массива. Если этот параметр не указан, то он будет определен автоматически на основе других параметров.

Возвращает:

результат - массив NumPy, как массив равномерно разнесенных значений.

4). Необходимо реализовать возможность по запросу пользователя вывода из файла графика для одного набора по выбору или нескольких графиков для различных наборов данных (не менее двух). При этом необходимо учесть случай, когда точки могут отстоять друг от друга неравномерно.