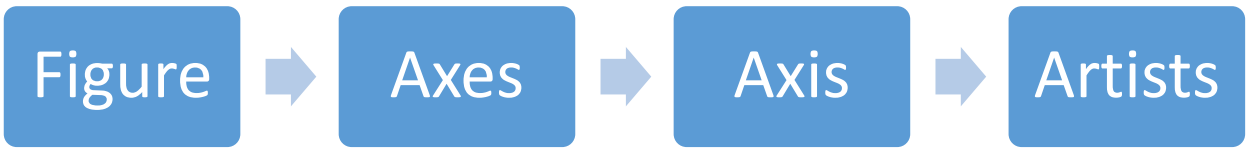
Лабораторная работа № 1

Тема: «Визуализация данных на языке Python с помощью библиотеки matplotlib»

Цель работы: изучить основы работы с matplotlib, освоить способы построения базовых графиков и их настройки.

Теоретическая справка

Matplotlib - это основная библиотека для построения научных графиков в Python. Включает функции для создания высококачественных визуализаций: линейных диаграмм, гистограмм и т.д. Пользовательская работа подразумевает операции с разными уровнями:



1. Рисунок (Figure) Любой рисунок в matplotlib имеет вложенную структуру. Рисунок - это объект самого верхнего уровня, на котором располагаются: области рисования (Axes); элементы рисунка Artists (заголовки, легенда и т.д.); основа-холст (Canvas). На рисунке может быть несколько областей рисования Axes, но данная область рисования Axes может принадлежать только одному рисунку Figure.

2. Область рисования (Axes) Объект среднего уровня. Это часть изображения с пространством данных. Каждая область рисования Axes содержит две (или три в случае трёхмерных данных) координатных оси (Axis объектов), которые упорядочивают отображение данных.

3. Координатная ось (Axis) Координатная ось является объектом среднего уровня, которая определяет область изменения данных. На них наносятся: деления ticks; подписи к делениям ticklabels. Расположение делений определяется объектом Locator, а подписи делений обрабатывает объект Formatter. Конфигурация координатных осей заключается в комбинировании различных свойств объектов Locator и Formatter.

4. Элементы рисунка (Artists) Практически всё, что отображается на рисунке является элементом рисунка (Artist), даже объекты Figure, Axes и Axis. Элементы рисунка Artists включают в себя такие простые объекты как: текст (Text); плоская линия (Line2D); фигура (Patch) и другие. Когда происходит отображение рисунка (figure rendering), все элементы рисунка Artists наносятся на основу-холст (Canvas). Большая часть из них связывается с областью рисования Axes. Также элемент рисунка не может совместно использоваться несколькими областями Axes или быть перемещён с одной на другую.

Pyplot - интерфейс для построения графиков простых функций. Позволяет пользователю сосредоточиться на выборе готовых решений и настройке базовых параметров рисунка. Это его главное достоинство, поэтому изучение matplotlib лучше всего начинать именно с интерфейса pyplot. Рисунки в matplotlib создаются путём последовательного вызова команд. Графические элементы (точки, линии, фигуры и т.д.) наслаиваются одна на другую последовательно. При этом последующие перекрывают предыдущие, если они занимают общее участки на рисунке (регулируется параметром zorder).

В matplotlib работает правило "текущей области" ("current axes"), которое означает, что все графические элементы наносятся на текущую область рисования. Несмотря на то, что областей рисования может быть несколько, один из них всегда является текущей. Так как главным объектом в matplotlib является рисунок Figure, создание научной графики нужно начинать именно с создания рисунка. Создать рисунок figure позволяет метод plt.figure().

После вызова любой графической команды (функции), которая создаёт какой-либо графический объект, например, plt.scatter() или plt.plot(), всегда существует хотя бы одна область для рисования (по умолчанию прямоугольной формы). Чтобы текущее состояние рисунка отразилось на экране, можно воспользоваться командой plt.show(). Будут показаны все рисунки (figures), которые были созданы.
Графические команды - это функции, которые, принимая некоторые параметры, возвращают ка кой-то графический результат. Это может быть текст, линия, график, диаграмма и др.

1. Базовые команды:
   - plt.scatter() - маркер или точечное рисование;
   - plt.plot() - ломаная линия;
   - plt.text() - нанесение текста.
2. Диаграммы
   - plt.bar(),- столбчатая диаграмма;
   - plt.hist()- гистограмма;
   - plt.pie() - круговая диаграмма;
   - plt.boxplot() - "ящик с усами" (boxwhisker);
3. Отображения
   - plt.matshow() - отображение данных в виде квадратов.

Axis - контейнер в matplotlib, который привязан к области рисования Axes и на котором располагаются деления осей (ticks), подписи делений (tick labels) и подписи осей (axis labels). Это третья "матрёшка" после Figure и Axes. Координатные оси являются экземплярами класса matplotlib.axis.Axis.

Любая область рисования Axes содержит два особых Artist-контейнера: XAxis и YAxis. Они отвечают за отрисовку делений (ticks) и подписей (labels) координатных осей, которые хранятся как экземпляры в переменных xaxis и yaxis. Чтобы обратиться к экземпляру axis, отвечающему, например, за ось ординат, нужно обратиться к контейнеру yaxis соответствующей области рисования ax.

Каждый экземпляр axis содержит атрибут подписей (label) координатной оси и список главных (major ticks) и вспомогательных (minor ticks) делений, а также является хранилищем для экземпляров делений (ticks).

Деления - это экземпляры класса matplotlib.axis.Tick, которые визуализируют деления (размер, цвет, толщину и т.д.) и подписи к ним. Деления создаются динамически исходя из области изменения переданных данных. В результате на координатной оси появляются и хранятся экземпляры классов matplotlib.axis.XTick и matplotlib.axis.YTick. Они родственны классу matplotlib.axis.Tick.

Хотя все графические примитивы, с помощью которых создаётся облик координатной оси содержатся в делениях ticks, у экземпляров axis есть средства для управления линиями делений (tick lines), подписями делений (tick labels), а также местоположением делений (tick locations).

Одними из самых базовых графических команд являются команды, отображающие текст. Например, такой командой является команда plt.text(). Она не привязана к какому-либо объекту вроде координатной оси или делений координатной оси, а в качестве входных данных принимает координаты положения будущей строки и сам текст в виде строки.

Ниже представлен список дополнительных текстовых команд в pyplot.

- plt.xlabel() - добавляет подпись оси абсции OX;
- plt.ylabel() - добавляет подпись оси ординат OY;
- plt.title() - добавляет заголовок для области рисования Axes;
- plt.figtext() - добавляет текст на рисунок Figure;
- plt.suptitle() - добавляет заголовок для рисунка Figure;
- plt.annotate() - добавляет примечание, которое состоит из текста и необязательной стрелки в указанную область на рисунке

matplotlibrc - файл настройки в котором хранятся значения по умолчанию для разных свойств графических элементов. Он инициализируется при каждой загрузке модуля matplotlib. Изменив содержание файла matplotlibrc можно сохранить пользовательские настройки для работы при следующих загрузках модуля matplotlib. matplotlibrc представляет собой текстовый файл, каждая строка которого описывает параметры в виде: параметр : значение При этом некоторые параметры имеют вложенную структуру.

В настройках matplotlibrc или rcParams существует такой параметр как fonts, то есть шрифты. Всего существует 5 наборов шрифтов: cursive; fantasy; monospace; sans-serif; serif. Один из этих пяти наборов является текущим. За это отвечает параметр font.family. Каждый набор может состоять из одного или более шрифтов. Данная настройка определяет шрифт для всех подписей и текста на рисунке.

Помимо семейств, текст также может иметь стиль. Атрибут стиля style может быть либо 'italic', либо 'oblique', либо 'normal' (по умолчанию). Толщина или "жирность" шрифта, может быть задана через атрибут fontweight, который принимает значения 'bold', 'light' или 'normal' (по умолчанию). Стили и форматы можно комбинировать.

Некоторые виды графиков

- Диаграмма рассеяния — это классический и фундаментальный вид диаграммы, используемый для изучения взаимосвязи между двумя переменными. Если у вас есть несколько групп в ваших данных, вы можете визуализировать каждую группу в другом цвете.
- Ящик с усами (диаграмма размаха). (англ. box-and-whiskers diagram or plot, box plot). — график, использующийся в описательной статистике, компактно изображающий одномерное распределение вероятностей.
- Тепловая карта - это метод визуализации данных, который показывает величину явления в виде цвета в двух измерениях. Изменение цвета может быть вызвано оттенком или интенсивностью, что дает читателю очевидные визуальные подсказки о том, как явление группируется или изменяется в пространстве.
- Построчная гистограмма. Построчные гистограммы имеют гистограмму вдоль переменных оси X и Y. Это используется для визуализации отношений между X и Y вместе с одномерным распределением X и Y по отдельности. Этот график часто используется в анализе данных (EDA).

Самостоятельное задание

1. Сгенерировать данные в соответствии с кодом:

```
import numpy as np
import pandas as pd

n=500
df=pd.DataFrame({'код
респондента':np.random.randint(1,np.int(0.33*n),size=n),
            'дата опроса':np.random.choice(['2020-01-31','2021-01-31','2022-02-
03','2019-06-06'],size=n),
   'пол':np.random.choice(['муж','жен'],size=n,p=(0.35,0.65)),
'образование':np.random.choice(['высшее','незаконченное высшее',
                        'среднее','среднее специальное'],p=(0.1,0.1,0.4,0.4),size=n),
            'социальный
статус':np.random.choice(['холост/незамужем','женат/замужем',

'вдовец/вдова','разведен/разведена'],p=(0.45,0.05,0.05,0.45),size=n),
'возраст':np.rint(np.random.normal(45,15,size=n)),
'рост':list(np.rint(np.random.normal(160,20,size=int(n*0.9))))+
    list(np.rint(np.random.normal(180,20,size=n-int(n*0.9))))
    })
df['вес']=df['рост']-(100+(df['рост']-
100)/20)+np.rint(np.random.normal(10,2,size=n))
```

2. По полученным данным создать следующие базовые диаграммы
   2.1. Создать круговую диаграмму, описывающую частоты значений переменной «образование». Добавить название графика.
   2.2. Создать диаграмму рассеяния, описывающую зависимость между переменными «рост» и «вес». Добавить подписи осей. Добавить сетку.
   2.3. Создать диаграмму «ящик-с-усами» описывающую распределение переменной «рост». Добавить значения и названия статистик на график.
   2.4. Создать гистограмму, описывающую частоты значений переменной «социальный статус». Добавить подписи значений частот на диаграмму
3. Добавить линию тренда (линейной регрессии) на диаграмму 2.2
4. Создать единую диаграмму, с использованием вложенных графиков (subplots), с использованием графиков из пп.2
5. Создать тепловую карту, соответствующую матрице корреляции между переменными «рост» и «вес».
6. Добавить разделение по полу на график из 2.3
7. Создать тепловую карту, соответствующую изменению среднего значения переменной «вес», в зависимости от «пол» и «социальный статус».
8. Создать матрицу диаграмм «ящик-с-усами» для переменных «рост», «вес» в зависимости от переменной «пол».
9. Создать график с двумя осями – для эмпирической плотности частоты переменных «социальный статус» и «образование»
10. Создать построчную диаграмму для переменных «рост» и «вес».