Практическая работа №4
Визуализация данных средствами MatplotLib. Основы

Цель занятия: получить навыки использования библиотеки визуализации данных Matplotlib с использованием языка программирования Python.

Пояснения к работе

matplotlib – это основная библиотека для построения научных графиков в Python. Она включает функции для создания высококачественных визуализаций типа линейных диаграмм, гистограмм, диаграмм разброса и т.д. Визуализация данных и различных аспектов вашего анализа может дать важную информацию.

В данной работе взаимодействие с matplotlib будет проходить в Jupyter Notebook (см. Методические указания к Практическому занятию №3) на базе Google Colab (см. https://colab.research.google.com/notebooks/intro.ipynb, https://github.com/deepmipt/dlschl/wiki/Инструкция-по-работе-с-Google-Colab).

В среде Jupyter Notebook возможно вывести рисунок прямо в браузере с помощью встроенных команд %matplotlib notebook и %matplotlib inline. Рекомендуется использовать %matplotlib inline.

Использование Google Colab позволяет не устанавливать на свой компьютер Jupyter Notebook.

1. Подготовительная часть.

1.1. Зарегистрировать электронную почту google (либо использовать существующий аккаунт).

1.2. Перейти по ссылке https://colab.research.google.com/notebooks/intro.ipynb

1.3. В правом верхнем углу нажать кнопку «Войти» и затем ввести свои учетные данные google.

1.4. В верхнем левом углу найдите подменю «Файл», далее «Создать блокнот».

2. Опробовать программу для построения 2D графиков со следующим текстом.

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
# Генерируем последовательность чисел от -10 до 10 с 100 шагами
x = np.linspace(-10, 10, 100)
# Генерируем случайную амплитуду для синусоиды
a = np.random.random()
# Создаем второй массив с помощью синуса
y = a*np.sin(x)
# Функция создает линейный график на основе двух массивов
plt.plot(x, y, marker="x")
```

Пример результата показан на рисунке 4.1.

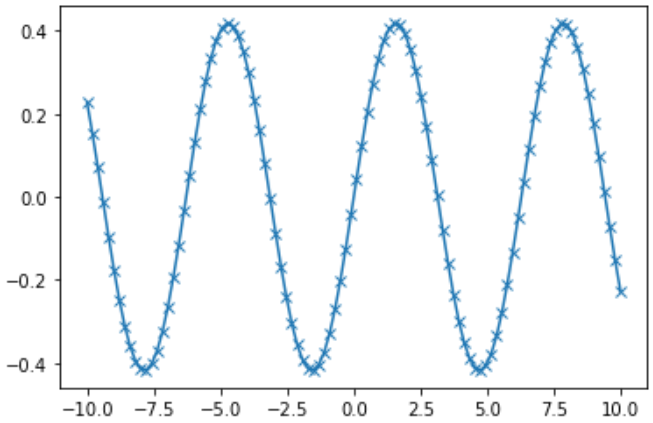

Рис. 4.1. Результат работы программы

3. Работа с данными, загруженными из открытых источников сети интернет.

3.0. В рамках данного пункта лабораторной работы будут использованы библиотеки Python pandas (https://pandas.pydata.org/), Numpy (https://numpy.org/). Стоит отметить, что библиотека pandas имеет

встроенный построитель графиков plot, который и будет использоваться в данном пункте. Будет использован набор данных (dataset) об Иммиграции в Канаду с 1980 по 2013 год - Международная миграция в отдельные страны и из них - Редакция 2015 года с веб-сайта Организации Объединенных Наций (https://www.un.org/en/development/desa/population/migration/data/empirical2/migrationflows.shtml). Набор данных содержит годовые данные о потоках международных мигрантов, регистрируемых различными странами. Данные показывают как приток, так и отток в зависимости от места рождения, гражданства или места предыдущего / следующего проживания как для иностранцев, так и для граждан. В рамках данного пункта мы сосредоточимся на данных иммиграционной службы Канады.

3.1. Загрузка и подготовка данных.

3.1.1. Импорт первичных библиотек - pandas, Numpy.

```
import numpy as np
import pandas as pd
```

3.1.2. Загрузка данных из сети интернет в *pandas dataframe*.

```
df_can = pd.read_excel('https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DV0101EN/labs/Data_Files/Canada.xlsx',
                       sheet_name='Canada by Citizenship',
                       skiprows=range(20),
                       skipfooter=2
                      )
print('Данные загружены и записаны в dataframe!')
```

3.1.3. Обзор данных – первые 5 элементов:

```
df_can.head()
```

3.1.4. Обзор данных – размер (строки и столбы) dataset’а:

```
print(df_can.shape)
```

3.1.5. Очистка данных – удаление неинформативных для нас столбцов, повторный вывод первых 5 строк:

```
df_can.drop(['AREA', 'REG', 'DEV', 'Type', 'Coverage'], axis=1, inplace=True)
df_can.head()
```

3.1.6. Приведение данных к более удобному виду – переименование нескольких столбцов, повторный вывод первых 5 строк:

```
df_can.rename(columns={'OdName':'Country', 'AreaName':'Continent','RegName':'Region'}, inplace=True)
df_can.head()
```

3.1.7. Проверка структуры данных – уточняем, являются ли наименования всех столбцов типами «строка» («string»):

```
all(isinstance(column, str) for column in df_can.columns)
```

Результатом будет скорее всего False. Поэтому выполняем преобразование.

3.1.8. Изменяем наименование всех столбцов так, чтобы они были типа string и проверяем заново:

```
df_can.columns = list(map(str, df_can.columns))
all(isinstance(column, str) for column in df_can.columns)
```

3.1.9. Приведение данных к более удобному виду – задаем в качестве строчного индекса наименование страны, повторный вывод первых 5 строк:

```
df_can.set_index('Country', inplace=True)
df_can.head()
```

3.1.10. Расширяем данные – создаем новый столбец Total, который будет являться суммой всех столбцов (фактически – количеством иммигрантов за все года с 1980 по 2013), повторный вывод первых 5 строк:

```
df_can['Total'] = df_can.sum(axis=1)
df_can.head()
```

3.1.11. Создаем новый набор данных на базе предыдущего – выделяем в него 5 стран, иммиграция из которых больше всех остальных:

```
years = list(map(str, range(1980, 2014)))
df_can.sort_values(['Total'], ascending=False, axis=0, inplace=True)
df_top5 = df_can.head()
# Транспонирование таблицы
df_top5 = df_top5[years].transpose()
df_top5.head()
```

3.2. Вывод данных в виде графика типа «Диаграмма с областями»:

```
%matplotlib inline

import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
mpl.style.use('ggplot') # опционально: задаем стиль ggplot

# Проверяем версию Matplotlib
print ('Matplotlib version: ', mpl.__version__) # >= 2.0.0

# Для построения графика изменяем тип индексов строк (года)
# на integer
df_top5.index = df_top5.index.map(int)
# Построение графика типа ‘area’ встроенной
# в pandas суб-библиотекой matplotlib
df_top5.plot(kind='area',
        stacked=False,
        figsize=(20, 10), # размер области построения графика
        )
#Задаем наименование графика
plt.title('Immigration Trend of Top 5 Countries')
#Задаем наименование оси Y
plt.ylabel('Number of Immigrants')
#Задаем наименование оси X
plt.xlabel('Years')
# Выводим график со всеми параметрами на экран
plt.show()
```

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Чем отличается построение графиков с помощью matplotlib и pandas?
2. Какое значение параметра kind нужно задать функции plot для вывода графика типа «Диаграмма с областями»?