

Отчет по практической работе №4

Тема: Визуализация данных средствами Matplotlib. Основы.

Цель работы: Получить навыки использования библиотеки визуализации данных Matplotlib в языке программирования Python.

Ход работы

1. Подготовительный этап

1. Зарегистрирован Google-аккаунт.
2. Вход выполнен через Google Colab.
3. Создан новый блокнот в Google Colab для выполнения лабораторной работы.

2. Построение 2D-графиков с Matplotlib

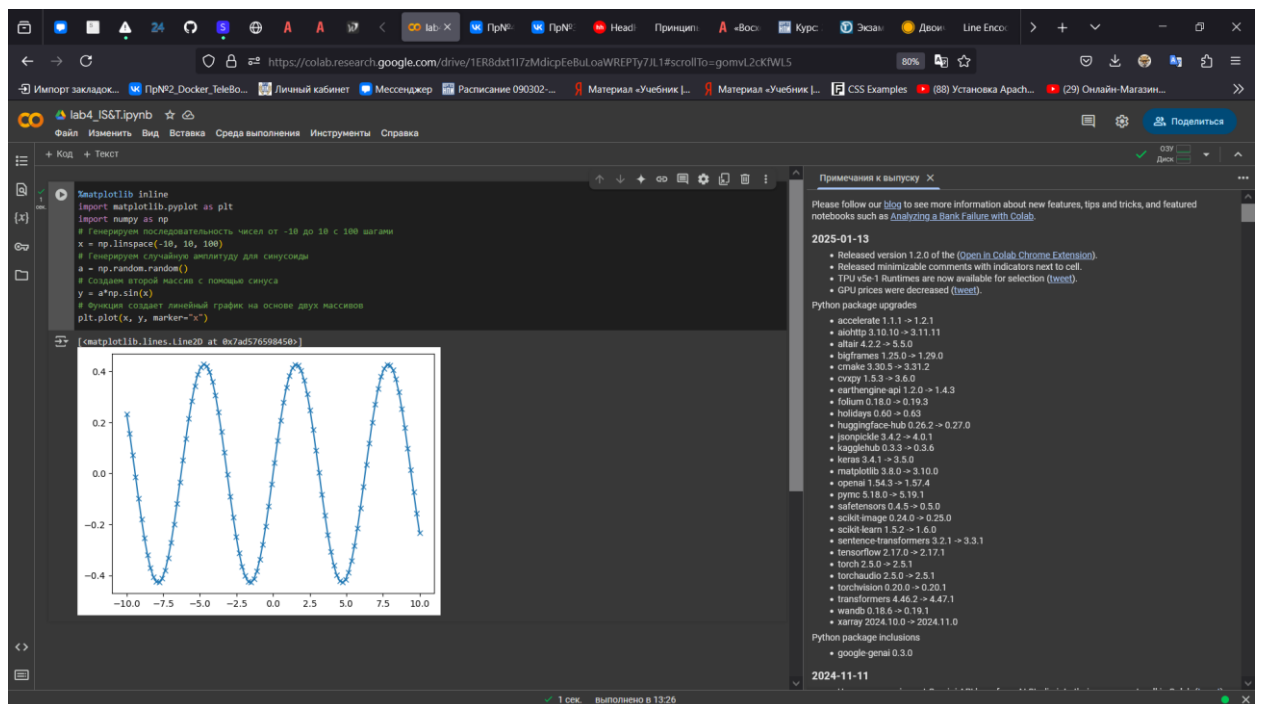


Рис. 1- Построение 2D-графика синусоиды

3. Работа с данными об иммиграции в Канаду

```
lab4_IS&Tjpyrb
Файл Изменить Вид Вставка Среда выполнения Инструменты Справка
+ Код + Текст
[2] Import numpy as np
Import pandas as pd

[3] #Загрузка данных из сети интернет в pandas dataframe.
df_can = pd.read_excel('https://s3-ap-sou-1.amazonaws.com/soflayer-net/cf-courses-data/CognitiveClass/04010EN/labs/Data_Files/Canada.xlsx',
                      sheet_name='Canada by Citizenship',
                      skiprows=range(20),
                      skipfooter=2)
print('Данные загружены и записаны в dataframe!')

Данные загружены и записаны в dataframe!

[4] df_can.head() #Обзор данных - первые 5 элементов

  Type  Coverage  OName  ARCA  ARCAname  RES  RegName  DEV  DevName  1980  ...  2004  2005  2006  2007  2008  2009  2010  2011  2012  2013
0 Immigrants  Foreigners  Afghanistan  935  Asia  5501  Southern Asia  902  Developing regions  16  ...  2978  3436  3009  2652  2111  1746  1758  2203  2635  2004
1 Immigrants  Foreigners  Albania  908  Europe  925  Southern Europe  901  Developed regions  1  ...  1450  1223  856  702  560  716  561  539  620  603
2 Immigrants  Foreigners  Algeria  903  Africa  912  Northern Africa  902  Developing regions  80  ...  3616  3626  4807  3623  4005  5393  4752  4325  3774  4331
3 Immigrants  Foreigners  American Samoa  909  Oceania  957  Polynesia  902  Developing regions  0  ...  0  0  1  0  0  0  0  0  0  0
4 Immigrants  Foreigners  Andorra  908  Europe  925  Southern Europe  901  Developed regions  0  ...  0  0  1  1  0  0  0  0  1  1
5 rows x 43 columns

[5] #Обзор данных - размер (строки и столбцы) dataset'a:
print(df_can.shape)

(195, 43)

[6] #Исчистка 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()

  OName  ARCAname  RegName  DevName  1980  1981  1982  1983  1984  1985  ...  2004  2005  2006  2007  2008  2009  2010  2011  2012  2013
0 Afghanistan  Asia  Southern Asia  Developing regions  16  39  39  47  71  340  ...  2978  3436  3009  2652  2111  1746  1758  2203  2635  2004
1 Albania  Europe  Southern Europe  Developed regions  1  0  0  0  0  0  ...  1450  1223  856  702  560  716  561  539  620  603
2 Algeria  Africa  Northern Africa  Developing regions  80  67  71  69  63  44  ...  3616  3626  4807  3623  4005  5393  4752  4325  3774  4331
3 American Samoa  Oceania  Polynesia  Developing regions  0  1  0  0  0  0  ...  0  0  1  0  0  0  0  0  0  0
4 Andorra  Europe  Southern Europe  Developed regions  0  0  0  0  0  0  ...  0  0  1  1  0  0  0  0  1  1
5 rows x 36 columns

[7] #Приведение данных к более удобному виду - переименование нескольких столбцов, повторный вывод первых 5 строк:
df_can.rename(columns={'OName':'Country', 'ARCAname':'Continent', 'RegName':'Region'}, inplace=True)
df_can.head()

  Country  Continent  Region  DevName  1980  1981  1982  1983  1984  1985  ...  2004  2005  2006  2007  2008  2009  2010  2011  2012  2013
0 Afghanistan  Asia  Southern Asia  Developing regions  16  39  39  47  71  340  ...  2978  3436  3009  2652  2111  1746  1758  2203  2635  2004
```

Рис. 2- Загрузка и подготовка данных

```
lab4_IS&Tjpyrb
Файл Изменить Вид Вставка Среда выполнения Инструменты Справка
+ Код + Текст
5 rows x 36 columns

[7] #Приведение данных к более удобному виду - переименование нескольких столбцов, повторный вывод первых 5 строк:
df_can.rename(columns={'OName':'Country', 'ARCAname':'Continent', 'RegName':'Region'}, inplace=True)
df_can.head()

  Country  Continent  Region  DevName  1980  1981  1982  1983  1984  1985  ...  2004  2005  2006  2007  2008  2009  2010  2011  2012  2013
0 Afghanistan  Asia  Southern Asia  Developing regions  16  39  39  47  71  340  ...  2978  3436  3009  2652  2111  1746  1758  2203  2635  2004
1 Albania  Europe  Southern Europe  Developed regions  1  0  0  0  0  0  ...  1450  1223  856  702  560  716  561  539  620  603
2 Algeria  Africa  Northern Africa  Developing regions  80  67  71  69  63  44  ...  3616  3626  4807  3623  4005  5393  4752  4325  3774  4331
3 American Samoa  Oceania  Polynesia  Developing regions  0  1  0  0  0  0  ...  0  0  1  0  0  0  0  0  0  0
4 Andorra  Europe  Southern Europe  Developed regions  0  0  0  0  0  0  ...  0  0  1  1  0  0  0  0  1  1
5 rows x 36 columns

[8] #Проверка структуры данных - проверка, являются ли наименования всех столбцов типами «строки» («string»):
all(isinstance(column, str) for column in df_can.columns)

False

[9] #Изменение наименования всех столбцов так, чтобы они были типа string и проверка заново:
df_can.columns = list(map(str, df_can.columns))
all(isinstance(column, str) for column in df_can.columns)

True

[10] #Приведение данных к более удобному виду - задать в качестве строкового индекса наименование страны, повторный вывод первых 5 строк:
df_can.set_index('Country', inplace=True)
df_can.head()

  Continent  Region  DevName  1980  1981  1982  1983  1984  1985  ...  2004  2005  2006  2007  2008  2009  2010  2011  2012  2013
Country
Afghanistan  Asia  Southern Asia  Developing regions  16  39  39  47  71  340  496  ...  2978  3436  3009  2652  2111  1746  1758  2203  2635  2004
Albania  Europe  Southern Europe  Developed regions  1  0  0  0  0  0  1  ...  1450  1223  856  702  560  716  561  539  620  603
Algeria  Africa  Northern Africa  Developing regions  80  67  71  69  63  44  69  ...  3616  3626  4807  3623  4005  5393  4752  4325  3774  4331
American Samoa  Oceania  Polynesia  Developing regions  0  1  0  0  0  0  0  ...  0  0  1  0  0  0  0  0  0  0
Andorra  Europe  Southern Europe  Developed regions  0  0  0  0  0  0  2  ...  0  0  1  1  0  0  0  0  1  1
5 rows x 37 columns

[11] #Расширение данных - создаем новый столбец Total, который будет являться суммой всех
#столбцов (всего 36 столбцов - количество иммигрантов за все годы с 1980 по 2013), повторный вывод первых 5 строк:
#и преобразуем все числовые столбцы в тип float (can.int)
df_can.iloc[:, 1:] = df_can.iloc[:, 1:].apply(pd.to_numeric, errors='coerce')
#и теперь можно безопасно выполнить суммирование
df_can['Total'] = df_can.sum(axis=1, numeric_only=True)
```

Рис. 3 – Приведение данных к более удобному виду

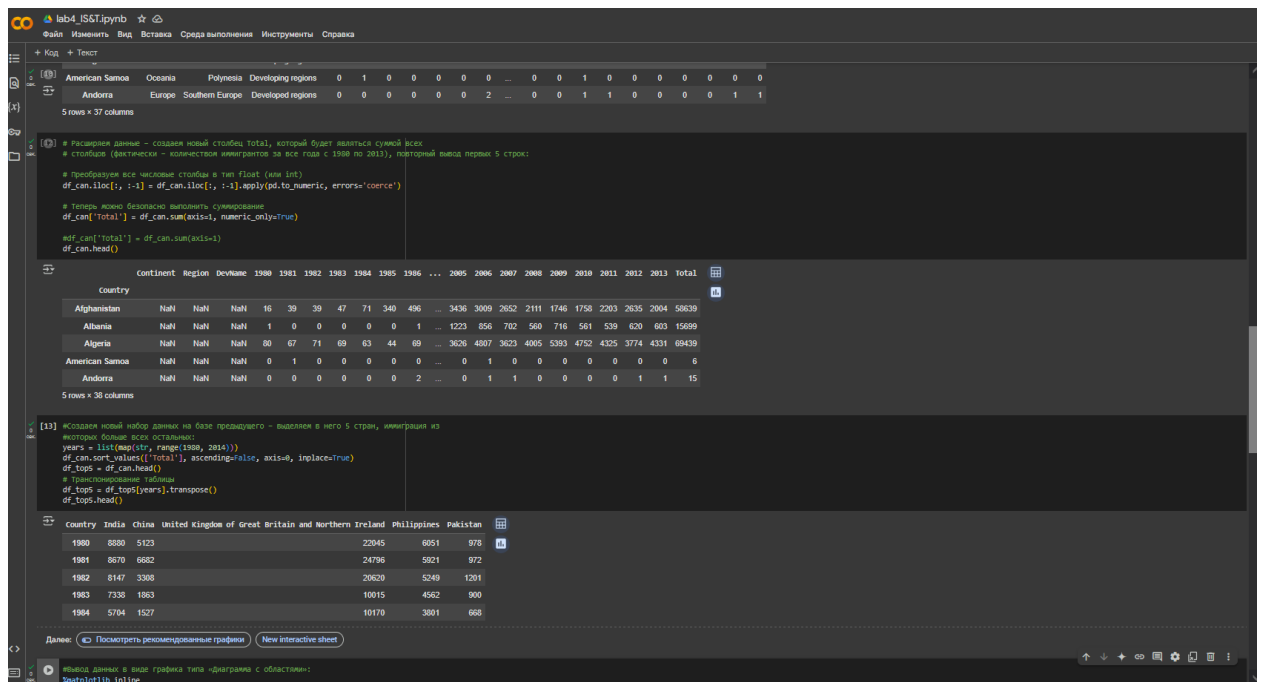


Рис. 4 - Создание нового набора данных на базе предыдущего

4. Визуализация данных (диаграмма с областями)

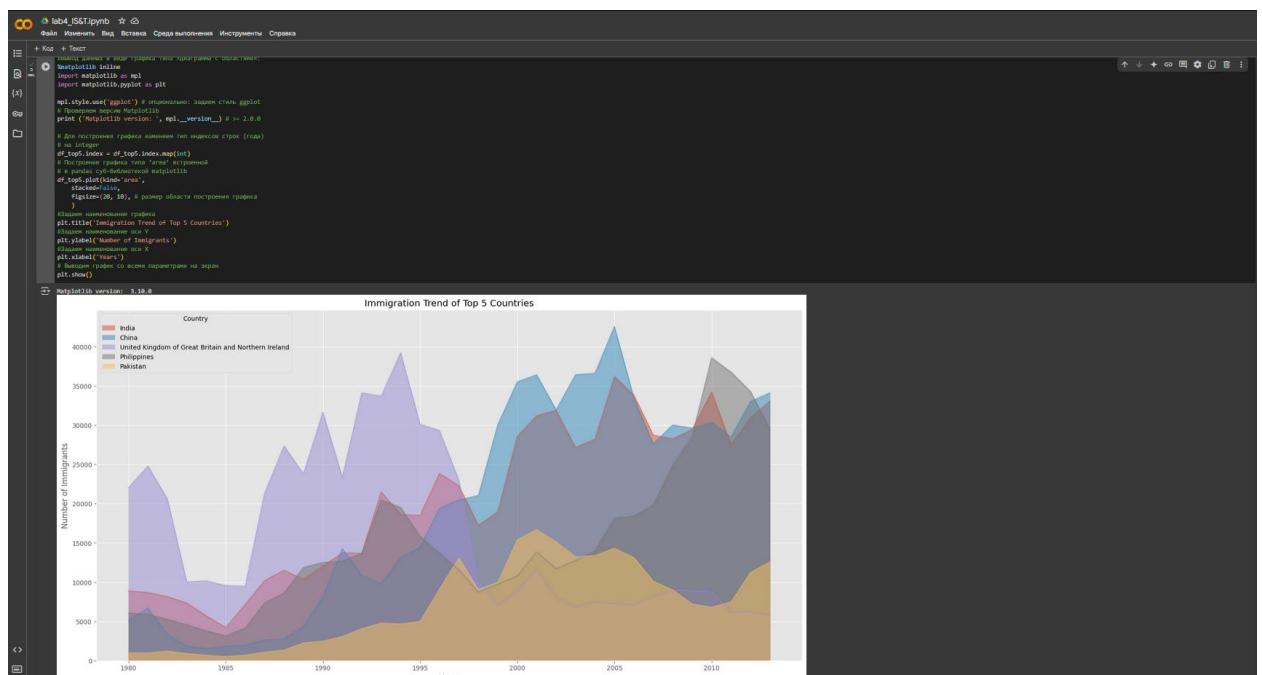


Рис. 5 - Визуализация данных (диаграмма с областями)

Вывод

1. Изучены основные принципы работы с библиотекой Matplotlib для построения графиков.
2. Получены навыки загрузки и обработки данных с помощью Pandas.
3. Построены и проанализированы графики трендов иммиграции.
4. Освоены базовые стилизации и настройка графиков в Matplotlib.