

Отчет по практической работе №5

Тема: Визуализация данных средствами Matplotlib. Диаграммы

Цель работы: Получить навыки использования библиотеки визуализации данных Matplotlib с использованием языка программирования Python, а также развернуть Telegram-бота с визуализацией данных в Docker.

1. Подготовительная часть

1. Зарегистрировать учетную запись Google (если еще нет).
2. Перейти по ссылке [Google Colab](#).
3. Войти в учетную запись Google.
4. Открыть блокнот Jupyter, созданный в рамках предыдущей практической работы.
5. Выполнить все ячейки блокнота.

2. Построение гистограмм



Рис. 1 - Построение гистограммы

3. Построение Bar Charts

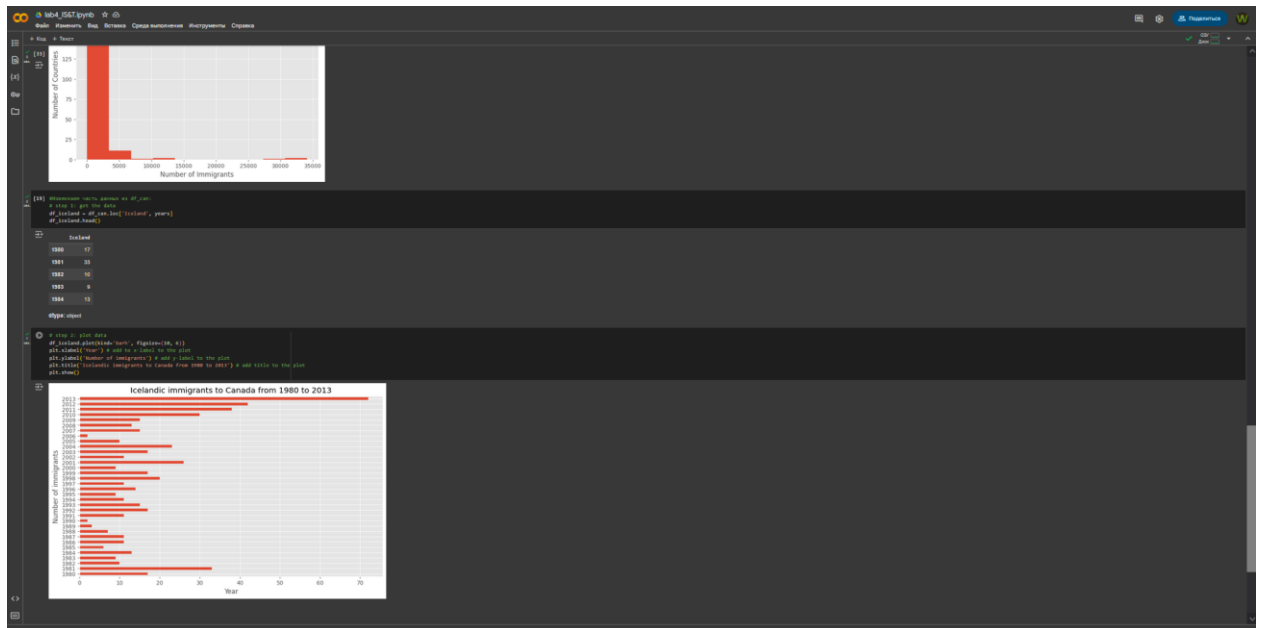


Рис. 2 - Построение Bar Charts

4. Разработка и запуск Telegram-бота

4.1. Код Telegram-бота

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import asyncio
from telegram import Update
from telegram.ext import Application, CommandHandler, CallbackContext
import logging

TOKEN = "7654177667:AAH3A4B1PI9DXK6N7Xno9Y6kIVx49Lt****"

async def send_graph1(update: Update, context: CallbackContext) -> None:
    x = np.linspace(0, 10, 100)
    y = np.sin(x)
    plt.figure()
    plt.plot(x, y, label="sin(x)", color="blue")
    plt.title("График sin(x)")
    plt.xlabel("X")
    plt.ylabel("Y")
    plt.legend()
    plt.savefig("graph1.png")
    plt.close()
    await update.message.reply_photo(photo=open("graph1.png", "rb"))

async def send_graph2(update: Update, context: CallbackContext) -> None:
    x = np.linspace(0, 10, 100)
    y = np.cos(x)
    plt.figure()
    plt.plot(x, y, label="cos(x)", color="red")
    plt.title("График cos(x)")
    plt.xlabel("X")
    plt.ylabel("Y")
    plt.legend()
```

```

async def send_graph3(update: Update, context: CallbackContext) -> None:
    df_iceland.plot(kind='barh', figsize=(10, 6))
    plt.xlabel('Year')
    plt.ylabel('Number of immigrants')
    plt.title('Icelandic immigrants to Canada from 1980 to 2013')
    plt.savefig("graph3.png")
    plt.close()
    await update.message.reply_photo(photo=open("graph3.png", "rb"))

```

```
def main():
    app = Application.builder().token(TOKEN).build()
    app.add_handler(CommandHandler("start", start))
    app.add_handler(CommandHandler("graph1", send_graph1))
    app.add_handler(CommandHandler("graph2", send_graph2))
    app.add_handler(CommandHandler("graph3", send_graph3))
    app.run_polling()
```

5. Разворачивание бота в Docker



6. Проверка работоспособности бота

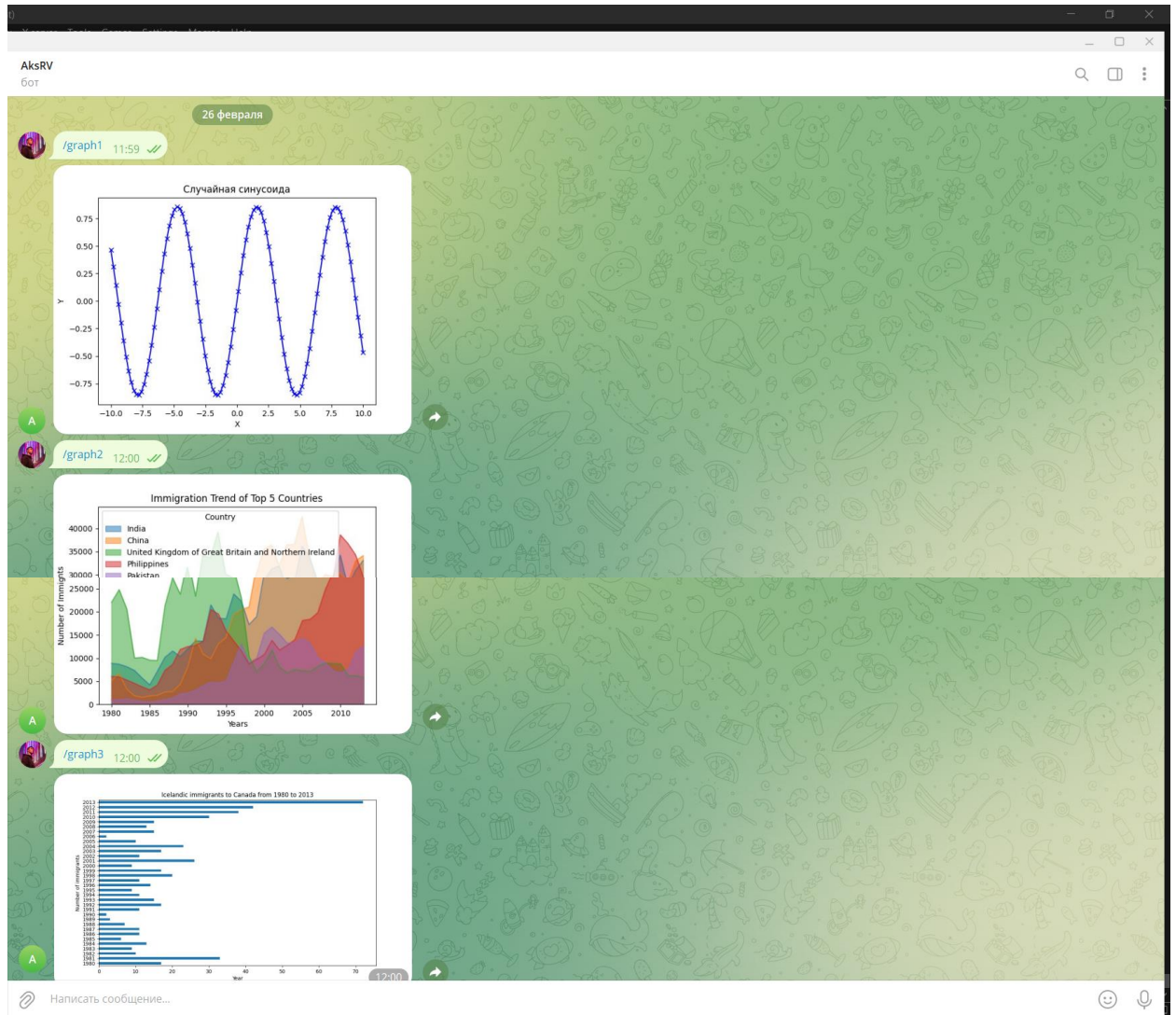


Рис. 4 - Проверка работоспособности бота

Вывод

В ходе работы были освоены методы визуализации данных с использованием библиотеки Matplotlib, разработан Telegram-бот, который генерирует и отправляет графики пользователям, а также развернут данный бот в контейнере Docker для постоянной работы. В работе была использована библиотека `python-telegram-bot` вместо `telorot`, так как `telorot` не поддерживает некоторые новые функции, и давно не обновлялась.