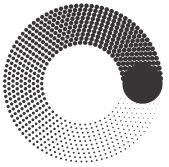
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

****

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий***

***Кафедра Информатики и информационных технологий***

**Направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 2**

**Дисциплина:** Инструмента визуализации данных

**Тема:** Визуализация данных с помощью библиотек языка Python

**Выполнил: студент группы 231-338**Шаура Илья Максимович

**Дата, подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата) (Подпись)

**Проверил:** Солонец В.И., к.т.н., доцент **\_\_\_\_\_\_\_\_\_\_**

(Фамилия И.О., степень, звание) (Оценка)

**Дата, подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата) (Подпись)

**Замечания:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2024**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import pycountry

from matplotlib import cm

from mpl\_toolkits.mplot3d import Axes3D

PLOT\_LABEL\_FONT\_SIZE = 14

# Генерация цветовой схемы

# Возвращает список цветов

def getColors(n):

    COLORS = []

    cm = plt.cm.get\_cmap('hsv', n)

    for i in np.arange(n):

        COLORS.append(cm(i))

    return COLORS

# Установка размера 2D графика

def set\_plot\_size(w,h,figure=plt):

    fig\_size = plt.rcParams['figure.figsize']

    fig\_size[0] = 12

    fig\_size[1] = 4.5

    figure.rcParams['figure.figsize'] = fig\_size

set\_plot\_size(12, 4.5)

def dict\_sort(my\_dict):

    keys = []

    values = []

    my\_dict = sorted(my\_dict.items(), key=lambda x:x[1], reverse=True)

    for k, v in my\_dict:

        keys.append(k)

        values.append(v)

    return (keys,values)

df = pd.read\_csv('./scrubbed.csv', escapechar='`', low\_memory=False)

df.fillna(value='unknown')

shapes\_label\_count = pd.value\_counts(df['shape'].values, sort=True)

country\_count = pd.value\_counts(df['country'].values, sort=True)

country\_count\_keys, country\_count\_values = dict\_sort(dict(country\_count))

TOP\_COUNTRY = len(country\_count\_keys)

plt.title('Страны, где больше всего наблюдений', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.bar(np.arange(TOP\_COUNTRY), country\_count\_values, color=getColors(TOP\_COUNTRY))

plt.xticks(np.arange(TOP\_COUNTRY), country\_count\_keys, rotation=0, fontsize=12)

plt.yticks(fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.ylabel('Количество наблюдений', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.show()

shapes\_type\_count = pd.value\_counts(df['shape'].values, sort=True)

shapes\_type\_count\_keys, shapes\_type\_count\_values = dict\_sort(dict(shapes\_type\_count))

OBJECT\_COUNT = len(shapes\_type\_count\_keys)

plt.title('Типы объектов', fontsize=PLOT\_LABEL\_FONT\_SIZE)

bar = plt.bar(np.arange(OBJECT\_COUNT), shapes\_type\_count\_values, color=getColors(OBJECT\_COUNT))

plt.xticks(np.arange(OBJECT\_COUNT), shapes\_type\_count\_keys, rotation=90, fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.yticks(fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.ylabel('Сколько раз видели', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.show()

MONTH\_COUNT = [0,0,0,0,0,0,0,0,0,0,0,0]

MONTH\_LABEL = ['Январь', 'Февраль', 'Март', 'Апрель', 'Май', 'Июнь',

    'Июль', 'Август', 'Сентябрь' ,'Октябрь' ,'Ноябрь' ,'Декабрь']

for i in df['datetime']:

    m,d,y\_t =  i.split('/')

    MONTH\_COUNT[int(m)-1] = MONTH\_COUNT[int(m)-1] + 1

plt.bar(np.arange(12), MONTH\_COUNT, color=getColors(12))

plt.xticks(np.arange(12), MONTH\_LABEL, rotation=90, fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.ylabel('Частота появления', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.yticks(fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.title('Частота появления объектов по месяцам', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.show()

shapes\_durations\_dict = {}

for i in shapes\_type\_count\_keys:

     dfs = df[['duration (seconds)', 'shape']].loc[df['shape'] == i]

     shapes\_durations\_dict[i] = dfs['duration (seconds)'].mean(axis=0)/60.0/60.0

#shapes\_durations\_dict\_keys, shapes\_durations\_dict\_values = dict\_sort(shapes\_durations\_dict)

shapes\_durations\_dict\_keys = []

shapes\_durations\_dict\_values = []

for k in shapes\_type\_count\_keys:

    shapes\_durations\_dict\_keys.append(k)

    shapes\_durations\_dict\_values.append(shapes\_durations\_dict[k])

plt.title('Среднее время появление каждого объекта', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.bar(np.arange(OBJECT\_COUNT), shapes\_durations\_dict\_values, color=getColors(OBJECT\_COUNT))

plt.xticks(np.arange(OBJECT\_COUNT), shapes\_durations\_dict\_keys, rotation=90, fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.yticks(fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.ylabel('Среднее время появления в часах', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.show()

shapes\_durations\_dict = {}

for i in shapes\_type\_count\_keys:

     dfs = df[['duration (seconds)', 'shape']].loc[df['shape'] == i]

     shapes\_durations\_dict[i] = dfs['duration (seconds)'].median(axis=0)/60.0/60.0

#shapes\_durations\_dict\_keys, shapes\_durations\_dict\_values = dict\_sort(shapes\_durations\_dict)

shapes\_durations\_dict\_keys = []

shapes\_durations\_dict\_values = []

for k in shapes\_type\_count\_keys:

    shapes\_durations\_dict\_keys.append(k)

    shapes\_durations\_dict\_values.append(shapes\_durations\_dict[k])

plt.title('Медианное время появление каждого объекта', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.bar(np.arange(OBJECT\_COUNT), shapes\_durations\_dict\_values, color=getColors(OBJECT\_COUNT))

plt.xticks(np.arange(OBJECT\_COUNT), shapes\_durations\_dict\_keys, rotation=90, fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.yticks(fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.ylabel('Медианное время появления в часах', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.show()

df['datetime'] = pd.to\_datetime(df['datetime'], errors='coerce')

df['decade'] = (df['datetime'].dt.year // 10) \* 10

decadal\_count = df['decade'].value\_counts().sort\_index()

plt.figure(figsize=(12, 6))

decadal\_count.plot(kind='bar', color=getColors(OBJECT\_COUNT), width=0.8)

plt.title('Количество наблюдений НЛО по десятилетиям', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.xlabel('Десятилетие', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.ylabel('Количество наблюдений', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.xticks(rotation=45, fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.show()

top\_cities = df['city'].value\_counts().head(10)

plt.figure(figsize=(12, 6))

top\_cities.plot(kind='bar', color=getColors(OBJECT\_COUNT))

plt.title('Топ-10 городов с наибольшим количеством наблюдений НЛО', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.xlabel('Город', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.ylabel('Количество наблюдений', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.xticks(rotation=45, fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.show()

us\_df = df[df['country'] == 'us']

shapes\_us\_count = pd.value\_counts(us\_df['shape'].values, sort=True)[:10]

shapes\_us\_count\_keys, shapes\_us\_count\_values = dict\_sort(dict(shapes\_us\_count))

plt.figure(figsize=(12, 6))

plt.bar(shapes\_us\_count\_keys, shapes\_us\_count\_values, color=getColors(12))

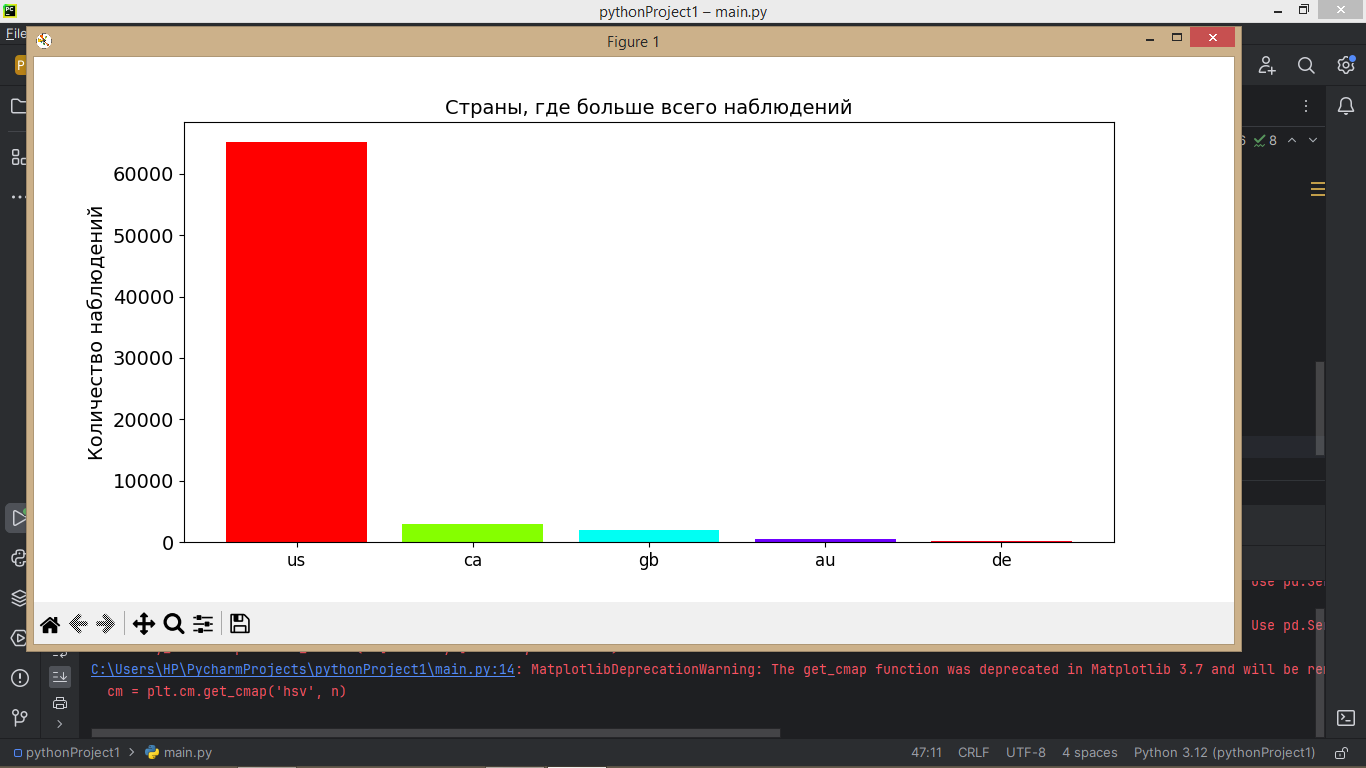
plt.title('Топ-10 форм наблюдаемых объектов в США', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.xlabel('Форма объекта', fontsize=PLOT\_LABEL\_FONT\_SIZE)

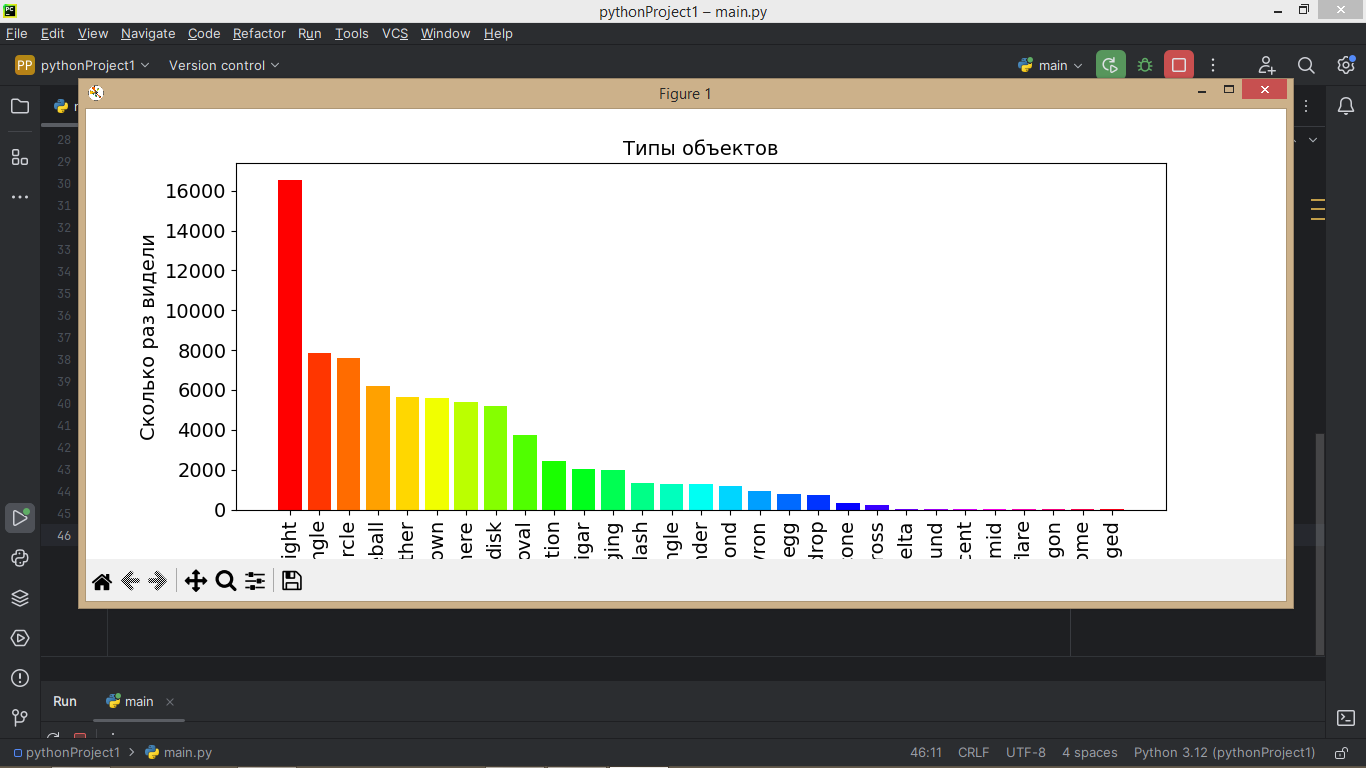
plt.ylabel('Количество наблюдений', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.xticks(rotation=45, fontsize=PLOT\_LABEL\_FONT\_SIZE)

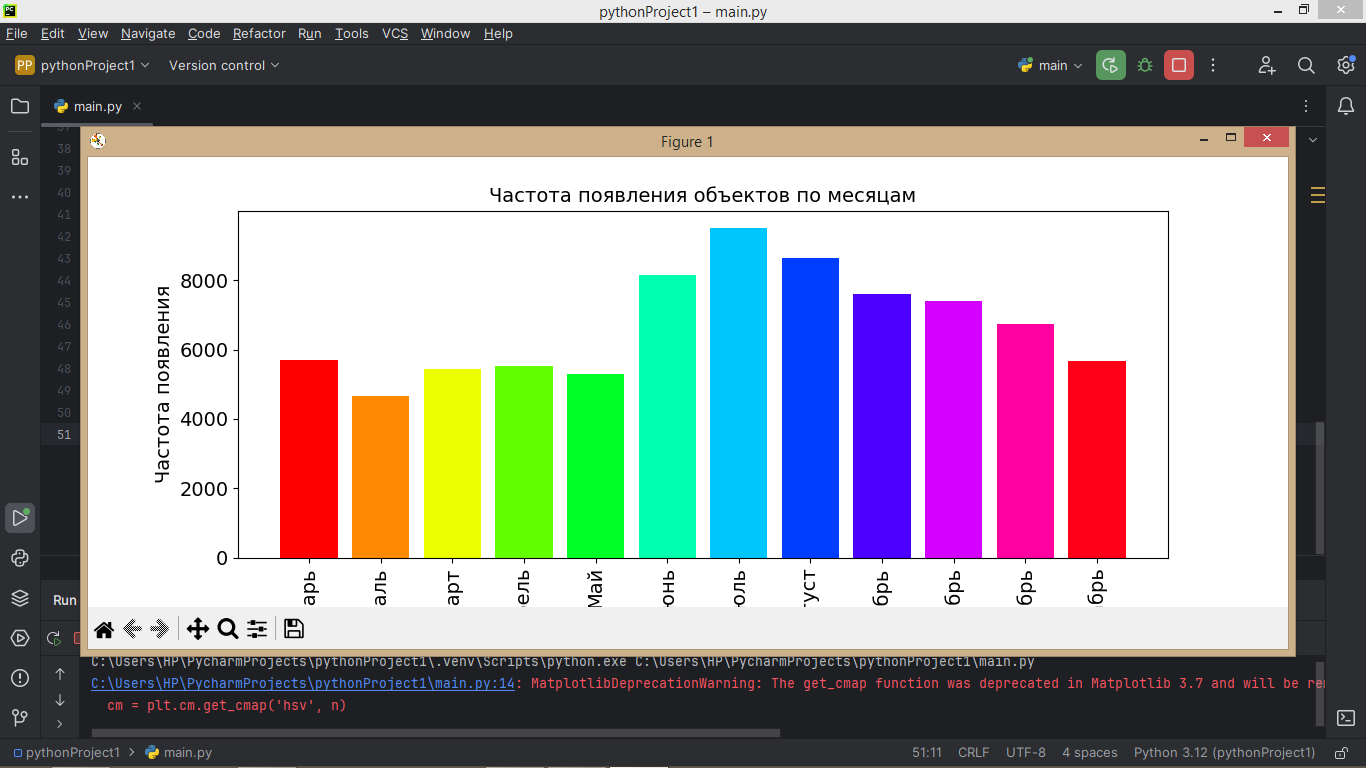
plt.show()



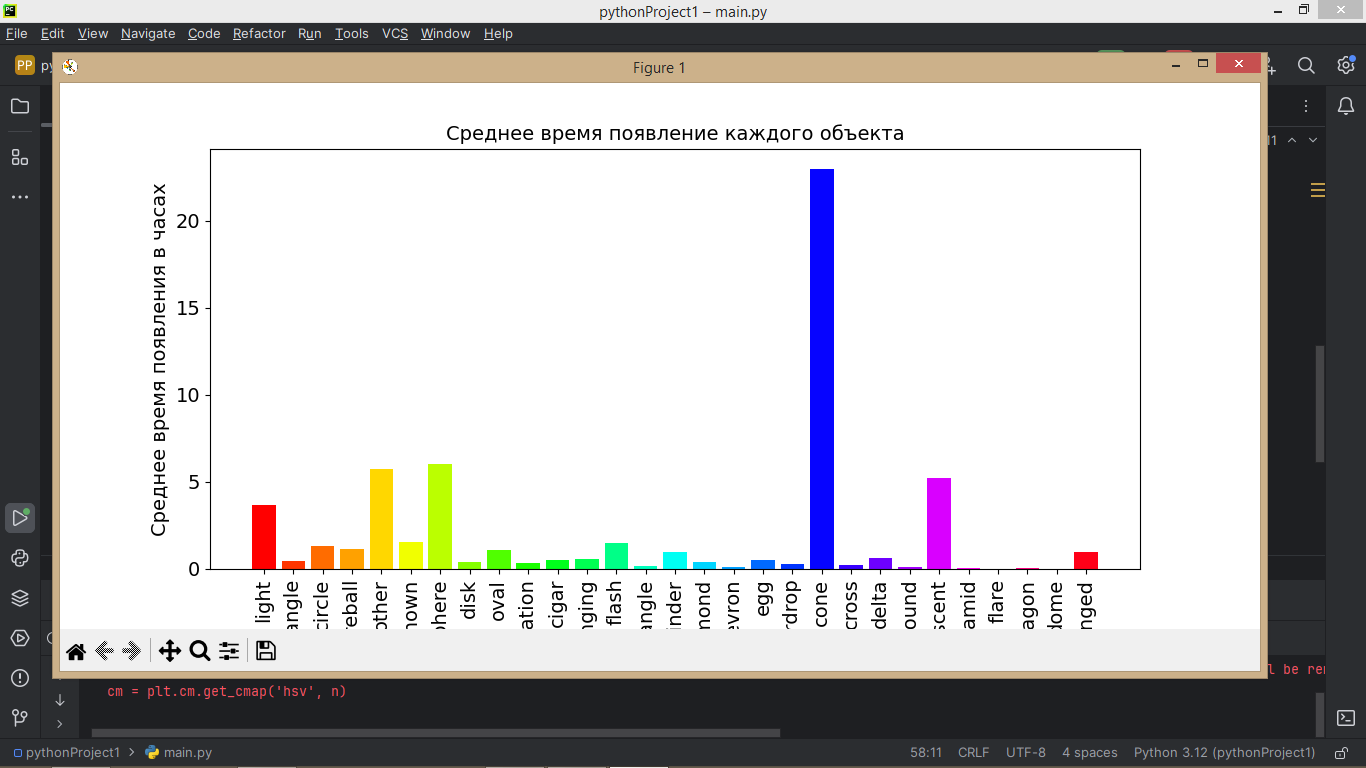
На данном графике продемонстрировано количество наблюдений НЛО в пяти странах, в качестве названий стран использованы их сокращения. По графику можем заметить, что наибольшее количество раз НЛО было замечено в США.



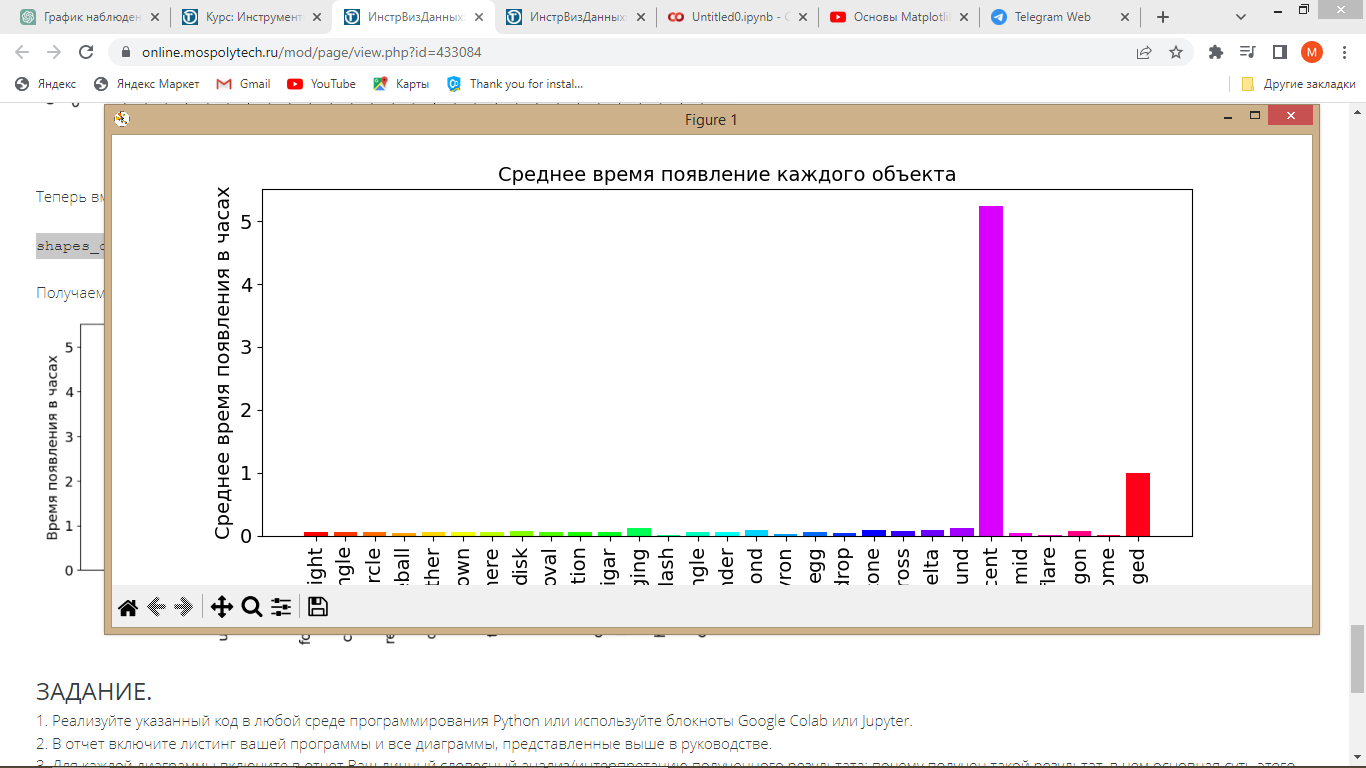
Данная гистограмма показывает, сколько раз объекты той или иной формы были замечен



Данная программа выводит столбчатую диаграмму, где можно узнать, количество появлений НЛО в разные месяцы.



Данная диаграмма показывает, сколько времени в среднем объект той или иной формы был активен.



Данная диаграмма показывает медианное время объектов.

