

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

**Отчет по лабораторной работе №5
*«Визуализация данных с помощью matplotlib»***

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-20-1

Бокань И.П. « » _____ 2022г.

Подпись студента _____

Работа защищена « » _____ 2022г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

1. Вывод (примеры)

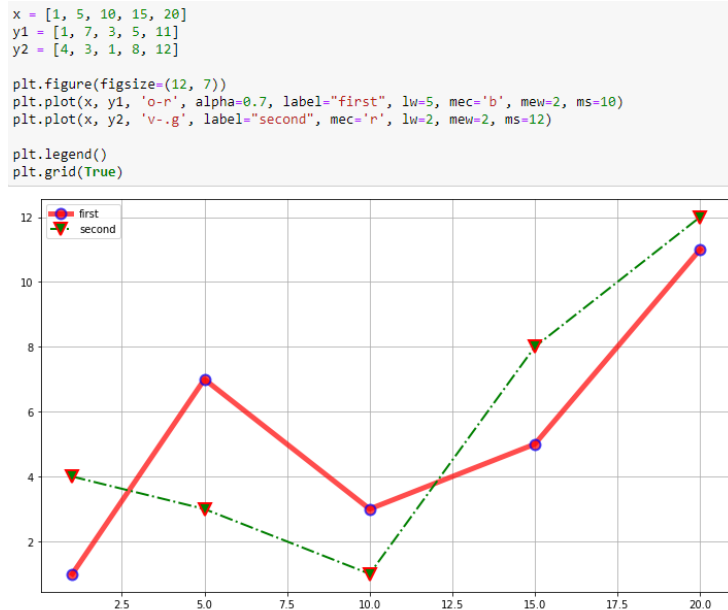


Рисунок 1.1 - Результаты примера построение графика

```
x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)

# Настройка размеров подложки
plt.figure(figsize=(16, 7))

plt.subplot(3, 3, 1)
plt.plot(x, y, c="r")
plt.fill_between(x, y)

plt.subplot(3, 3, 2)
plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))

plt.subplot(3, 3, 3)
plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0))

plt.subplot(3, 3, 4)
plt.plot(x, y, c="r")
plt.grid()
plt.fill_between(x, 0.5, y, where=(y>=0.5))

plt.subplot(3, 3, 5)
plt.plot(x, y, c="r")
plt.grid()
plt.fill_between(x, y, 1)

plt.subplot(3, 3, 6)
plt.plot(x, y, c="r")
plt.grid()
plt.fill_between(x, y, where=y>=0, color="g", alpha=0.3)
plt.fill_between(x, y, where=y<=0, color="r", alpha=0.3)

plt.show()
```

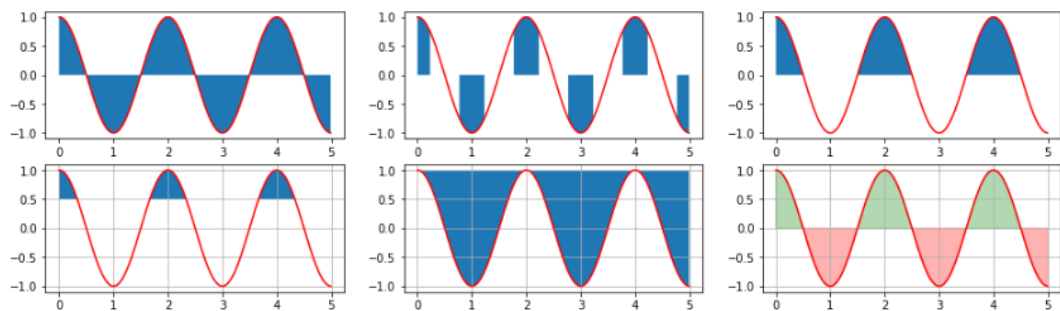


Рисунок 1.2 - Результаты примера заливка области между графиком и осью

```

x = [1, 2, 3, 4, 5, 6, 7]
y = [7, 6, 5, 4, 5, 6, 7]

# Настройка размеров подложки
plt.figure(figsize=(16, 7))

plt.plot(x, y, marker="o", c="g")
plt.subplot(2, 2, 1)

x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)
plt.subplot(2, 2, 2)

plt.plot(x, y, marker="o", c="g")
plt.show()

x = np.arange(0.0, 5, 0.01)
y = np.cos(x * np.pi)
m_ev_case = [None, 10, (100, 30), slice(100,400,15), [0, 100, 200, 300], [10, 50, 100]]
fig, ax = plt.subplots(2, 3, figsize=(10, 7))
axs = [ax[i, j] for i in range(2) for j in range(3)]

for i, case in enumerate(m_ev_case):
    axs[i].set_title(str(case))
    axs[i].plot(x, y, "o", ls='-', ms=7, markevery=case)

plt.show()

```

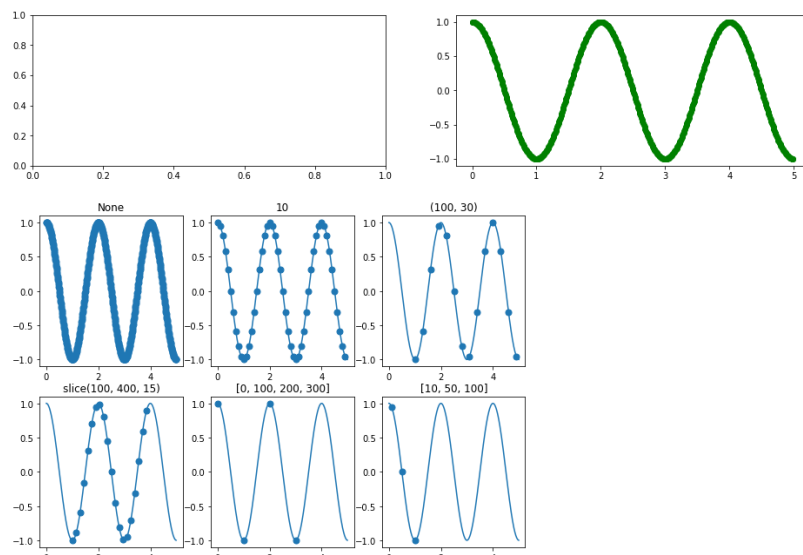


Рисунок 1.3 - Результаты примера настройка маркировки графиков

```

x = np.arange(0.0, 5, 0.01)
y = np.cos(x * np.pi)

y_masked = np.ma.masked_where(y < -0.5, y)

plt.ylim(-1, 1)
plt.plot(x, y_masked, linewidth=3)

plt.show()

```

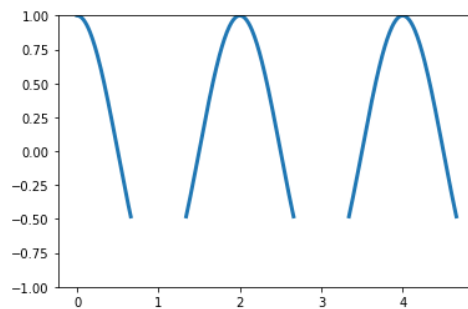


Рисунок 1.4 - Результаты примера обрезка графика

```

x = np.arange(0, 7)
y = x

where_set = ['pre', 'post', 'mid']
fig, axs = plt.subplots(1, 3, figsize=(15, 4))

for i, ax in enumerate(axs):
    ax.step(x, y, "g-o", where=where_set[i])
    ax.grid()

plt.show()

```

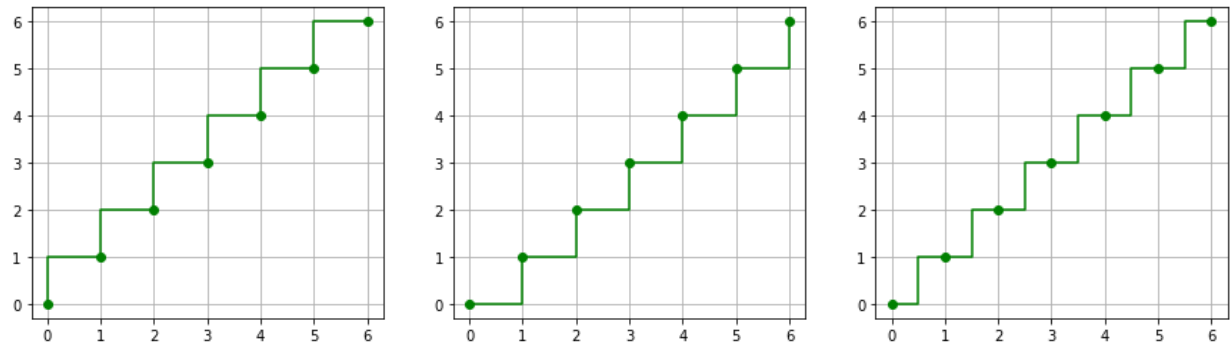


Рисунок 1.5 - Результаты примера ступенчатый график

```

x = np.arange(0, 11, 1)

y1 = np.array([(-0.2)*i**2+2*i for i in x])
y2 = np.array([(-0.4)*i**2+4*i for i in x])
y3 = np.array([2*i for i in x])

labels = ["y1", "y2", "y3"]

fig, ax = plt.subplots()

ax.stackplot(x, y1, y2, y3, labels=labels)
ax.legend(loc='upper left')

plt.show()

```

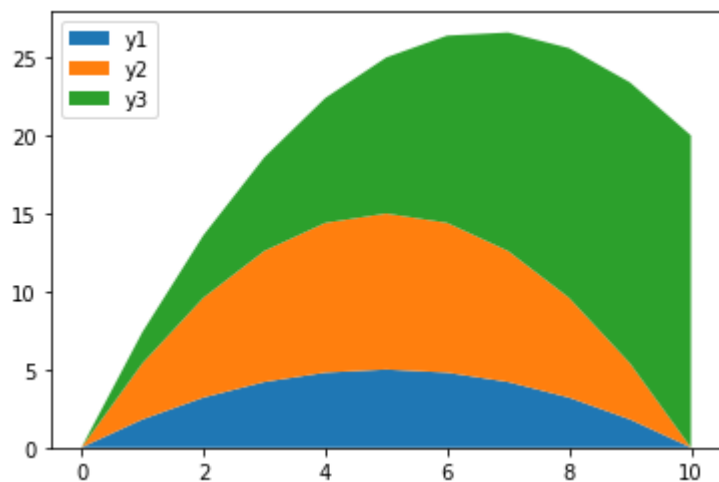


Рисунок 1.6 - Результаты примера стековый график

```
x = np.arange(0, 10.5, 0.5)
y = np.array([(-0.2)*i**2+2*i for i in x])

plt.figure(figsize=(16, 7))

plt.subplot(2, 2, 1)
plt.stem(x, y)

plt.subplot(2, 2, 2)
plt.stem(x, y, linefmt="r--", markerfmt="^", bottom=1)
plt.show()
```

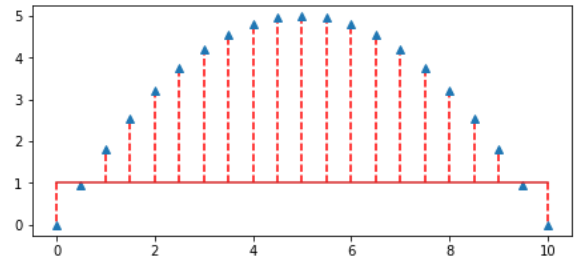
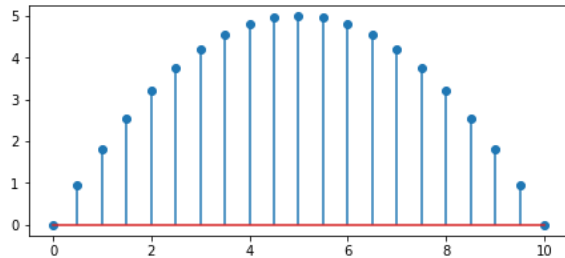


Рисунок 1.7 - Результаты примера stem-график

```
x = np.arange(0, 10.5, 0.5)
y = np.cos(x)

plt.figure(figsize=(16, 7))

plt.subplot(3, 3, 1)
plt.scatter(x, y)

plt.subplot(3, 3, 2)
x = np.arange(0, 10.5, 0.5)
y = np.cos(x)
plt.scatter(x, y, s=80, c="r", marker="D", linewidths=2, edgecolors="g")

plt.subplot(3, 3, 3)
bc = mcolors.BASE_COLORS
x = np.arange(0, 10.5, 0.25)
y = np.cos(x)
num_set = np.random.randint(1, len(mcolors.BASE_COLORS), len(x))
sizes = num_set * 35
colors = [list(bc.keys())[i] for i in num_set]
plt.scatter(x, y, s=sizes, alpha=0.4, c=colors, linewidths=2, edgecolors="face")
plt.plot(x, y, "g--", alpha=0.4)

plt.show()
```

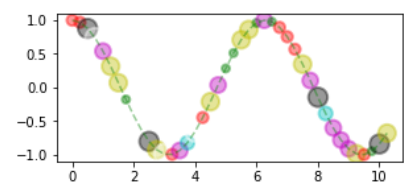
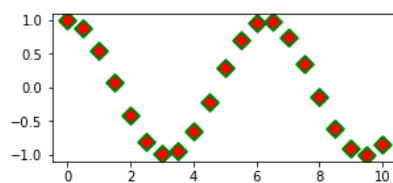
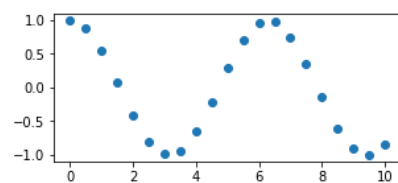


Рисунок 1.8 - Результаты примера точенный график

```

np.random.seed(123)

groups = [f"P{i}" for i in range(7)]
counts = np.random.randint(3, 10, len(groups))

plt.figure(figsize=(16, 9))

plt.subplot(3, 3, 1)
plt.bar(groups, counts)

plt.subplot(3, 3, 2)
plt.barh(groups, counts)

plt.subplot(3, 3, 3)
bc = mcolors.BASE_COLORS
np.random.seed(123)
groups = [f"P{i}" for i in range(7)]
counts = np.random.randint(0, len(bc), len(groups))
width = counts*0.1
colors = [{"r", "b", "g"}[int(np.random.randint(0, 3, 1))] for _ in counts]

plt.bar(groups, counts, width=width, alpha=0.6, bottom=2, color=colors, edgecolor="k", linewidth=2)

plt.show()

```

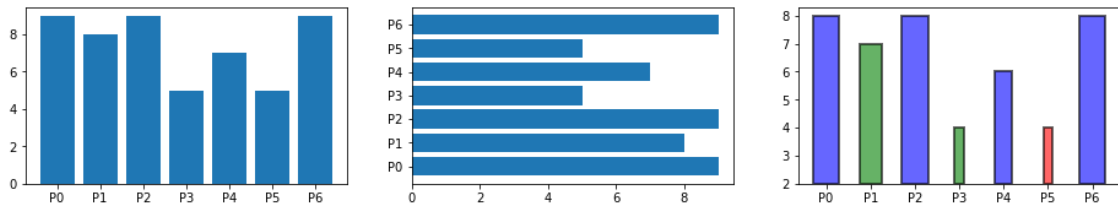


Рисунок 1.9 - Результаты примера столбчатые диаграммы

```

cat_par = [f"P{i}" for i in range(5)]
g1 = [10, 21, 34, 12, 27]
g2 = [17, 15, 25, 21, 26]
width = 0.3
x = np.arange(len(cat_par))
fig, ax = plt.subplots()

rects1 = ax.bar(x - width/2, g1, width, label='g1')
rects2 = ax.bar(x + width/2, g2, width, label='g2')

ax.set_title('Пример групповой диаграммы')
ax.set_xticks(x)
ax.set_xticklabels(cat_par)
ax.legend()

plt.show()

```

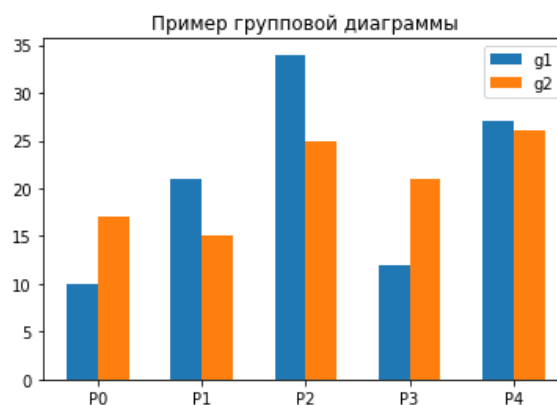


Рисунок 1.10 - Результаты примера групповые столбчатые диаграммы

```

np.random.seed(123)

rnd = np.random.randint
cat_par = [f"P{i}" for i in range(5)]
g1 = [10, 21, 34, 12, 27]

error = np.array([[rnd(2,7),rnd(2,7)] for _ in range(len(cat_par))]).T

fig, axs = plt.subplots(1, 2, figsize=(10, 5))

axs[0].bar(cat_par, g1, yerr=5, ecolor="r", alpha=0.5, edgecolor="b", linewidth=2)
axs[1].bar(cat_par, g1, yerr=error, ecolor="r", alpha=0.5, edgecolor="b", linewidth=2)

plt.show()

```

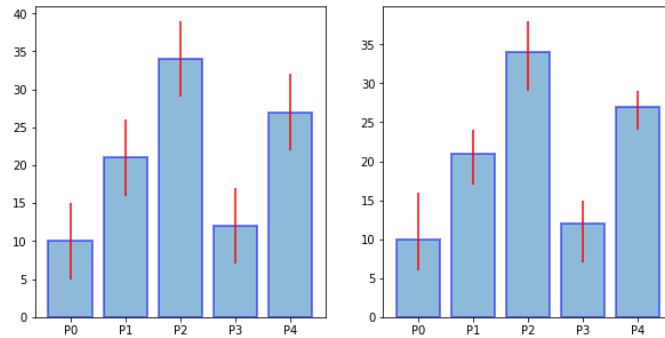


Рисунок 1.11 - Результаты примера диаграмма с errorbar элементом

```

vals = [24, 17, 53, 21, 35]
labels = ["Ford", "Toyota", "BMW", "AUDI", "Jaguar"]
fig, ax = plt.subplots()

ax.pie(vals, labels=labels)
ax.axis("equal")
plt.show()

explode = (0.1, 0, 0.15, 0, 0)
fig, ax = plt.subplots()
ax.pie(vals, labels=labels, autopct='%1.1f%%', shadow=True, explode=explode,
ax.axis("equal")
plt.show()

```

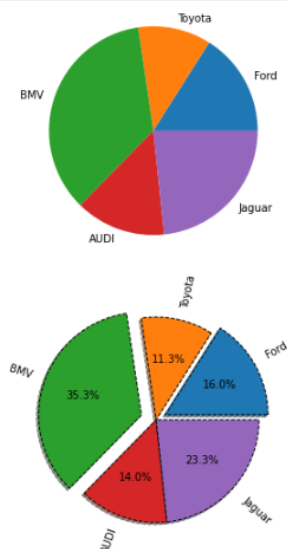


Рисунок 1.12 - Результаты примера классическая круговая диаграмма

```
fig, ax = plt.subplots()
offset=0.4
data = np.array([[5, 10, 7], [8, 15, 5], [11, 9, 7]])
cmap = plt.get_cmap("tab20b")
b_colors = cmap(np.array([0, 8, 12]))
sm_colors = cmap(np.array([1, 2, 3, 9, 10, 11, 13, 14, 15]))

ax.pie(data.sum(axis=1), radius=1, colors=b_colors, wedgeprops=dict(width=offset, edgecolor='w'))
ax.pie(data.flatten(), radius=1-offset, colors=sm_colors, wedgeprops=dict(width=offset, edgecolor='w'))

plt.show()
```



Рисунок 1.13 - Результаты примера вложенные круговые диаграммы

```
vals = [24, 17, 53, 21, 35]

labels = ["Ford", "Toyota", "BMW", "AUDI", "Jaguar"]
fig, ax = plt.subplots()
ax.pie(vals, labels=labels, wedgeprops=dict(width=0.5))

plt.show()
```

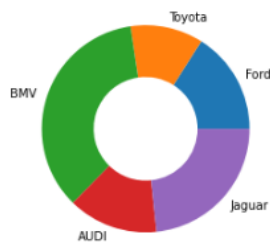


Рисунок 1.14 - Результаты примера круговая диаграмма в виде бублика

```
response = requests.get('https://dagman.ru/deyatelnost/modules/cube-dir2.png')
img = Image.open(BytesIO(response.content))

plt.imshow(img)
plt.show()

np.random.seed(32098201)
data = np.random.randn(25, 25)
plt.imshow(data)
plt.show()

fig, axes = plt.subplots(1, 2, figsize=(10,3), constrained_layout=True)
p1 = axes[0].imshow(data, cmap='winter', aspect='equal', vmin=-1, vmax=1, origin="lower")
fig.colorbar(p1, ax=axes[0])
p2 = axes[1].imshow(data, cmap='plasma', aspect='equal', interpolation='gaussian', origin="lower", extent=(0, 30, 0, 30))
fig.colorbar(p2, ax=axes[1])
plt.show()

np.random.seed(123)
data = np.random.rand(5, 7)
plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')
plt.show()
```

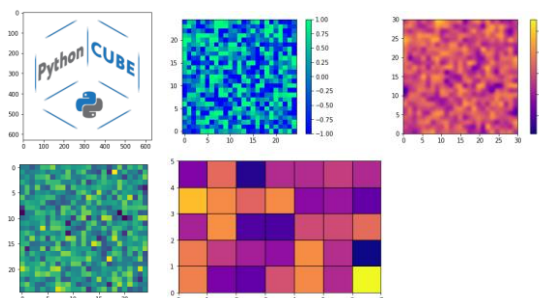


Рисунок 1.15 - Результаты примера круговая диаграмма в виде бублика

2. Решение вычислительной задачи

Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения линейного графика, условие которой предварительно необходимо согласовать с преподавателем.

```
dates = []
values = {}

ind = 0
size = ind + 5

with open("RU_newreg.csv", newline='', encoding='utf-8') as f:
    for row in csv.reader(f, delimiter = ';', quotechar = '\\'):
        if dates == []:
            try:
                dates = [ dt.datetime.strptime("{}-01".format(d), '%Y-%m-%d').date() for d in row[1:] ]
            except ValueError:
                pass

            continue
        values[ row[0] ] = row[1:]

dpi = 80
fig = plt.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )
mpl.rcParams.update({'font.size': 10})

plt.figure(figsize=(16, 7))

plt.title('Регистрация новых Доменных имен RU')
plt.xlabel('Год')
plt.ylabel('Домен')

ax = plt.axes()
ax.yaxis.grid(True)
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
ax.xaxis.set_major_locator(mdates.YearLocator())

xMax = 5
for reg in values.keys():
    if xMax <= 0:
        break

    plt.plot(dates[ind:size], values[reg][ind:size], linestyle = 'solid', label = reg)
    plt.stem(dates[ind:size], values[reg][ind:size], linefmt="r--", markerfmt="^", bottom=1)

    xMax -= 1

plt.legend(loc='upper left', frameon = False)

plt.show()
```

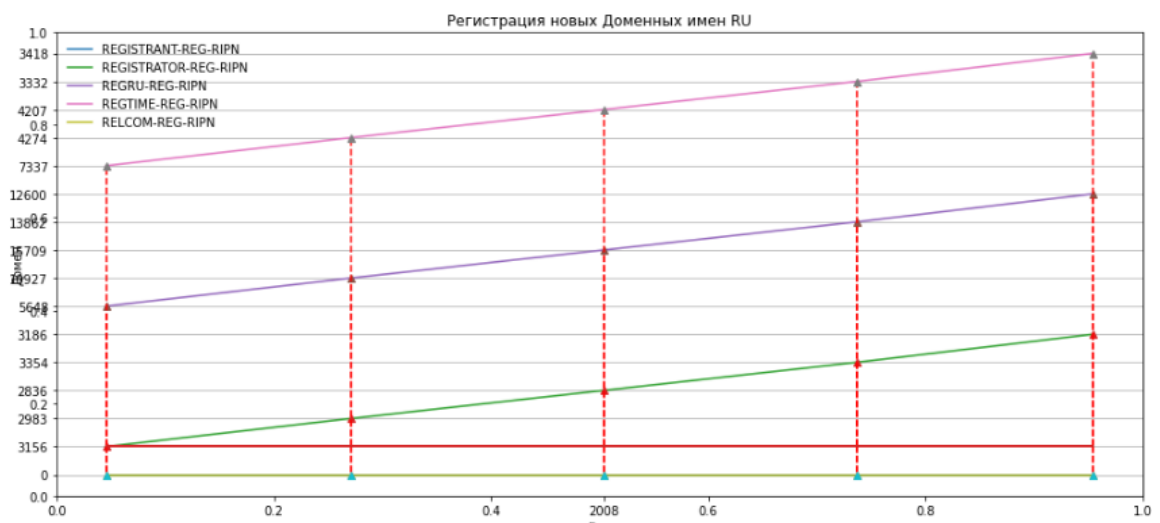


Рисунок 2.1 - Результаты построения линейного графика

Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения столбчатой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

```
data_names = ['кафе', 'аптека', 'топливо', 'банк', 'утилизация отходов', 'банкомат', 'скамейка', 'парковка', 'ресторан', 'место поклонения']
data_values = [9124, 8652, 7592, 7515, 7041, 6487, 6374, 6277, 5092, 3629]

dpi = 80
fig = plt.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )
mpl.rcParams.update({'font.size': 10})

plt.figure(figsize=(16, 7))
plt.title('Типы точек "Открытая Карта Улиц"')

ax = plt.axes()
ax.yaxis.grid(True, zorder = 1)

xs = range(len(data_names))

plt.bar([x + 0.05 for x in xs], [d * 0.9 for d in data_values], width = 0.2, color = 'Green', alpha = 0.7, label = '2022',
plt.bar([x + 0.3 for x in xs], data_values, width = 0.2, color = 'red', alpha = 0.7, label = '2019', zorder = 2)
plt.xticks(xs, data_names)

fig.autofmt_xdate(rotation = 25)

plt.legend(loc='upper right')
plt.show()
```

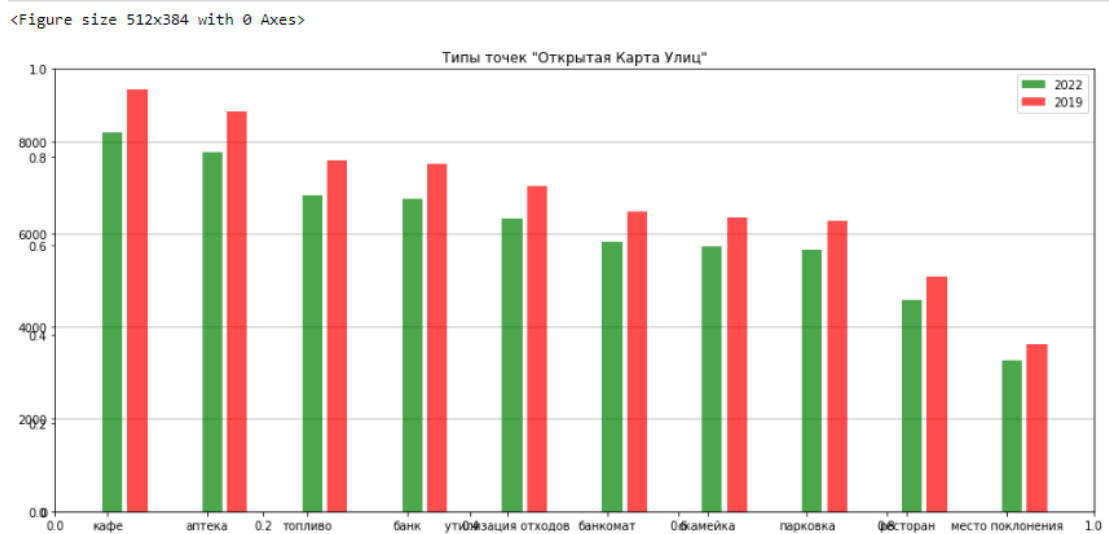


Рисунок 2.2 - Результаты построения столбчатой диаграммы

Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения круговой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

```
data_names = ['Москва', 'Санкт-Петербург', 'Сочи', 'Архангельск', 'Владимир', 'Краснодар', 'Курск', 'Воронеж', 'Ставрополь', 'Мурманск']
data_values = [1076, 979, 222, 189, 137, 134, 124, 124, 91, 79]

dpi = 80
fig = plt.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )
mpl.rcParams.update({'font.size': 9})

plt.title('Распределение кафе по городам России (%)')

xs = range(len(data_names))

plt.pie(data_values, autopct='%1f', radius = 1.1, explode = [0.15] + [0 for _ in range(len(data_names) - 1)] )
plt.legend(bbox_to_anchor = (-0.16, 0.45, 0.25, 0.25), loc = 'lower left', labels = data_names )

plt.show()
```



Рисунок 2.3 - Результаты построения круговой диаграммы

Найти какое-либо изображение в сети Интернет. Создать ноутбук, в котором будет отображено выбранное изображение средствами библиотеки matplotlib по URL из сети Интернет.

```
response = requests.get('https://i0.wp.com/img.aapks.com/imgs/3/6/d/36d063240e57333a4829601c1e26f9e1_icon.png')
plt.imshow(Image.open(BytesIO(response.content)))
plt.show()
```

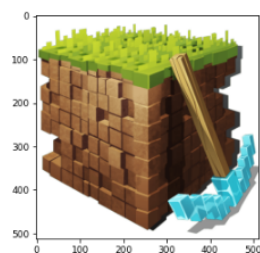


Рисунок 2.4 - Результаты построения круговой диаграммы