МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

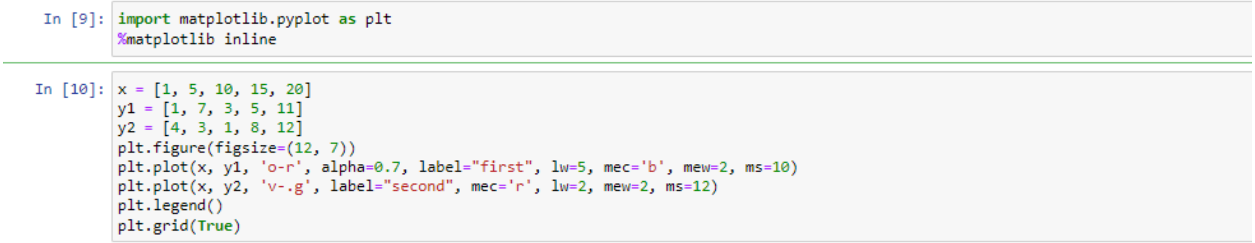
Отчет по лабораторной работе №5
«Визуализация данных с помощью matplotlib»

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-20-1
Симоненко А.С. « » ___________ 2022г.
Подпись студента ___________
Работа защищена « »____________2022г.
Проверил Воронкин Р.А. ________
(подпись)

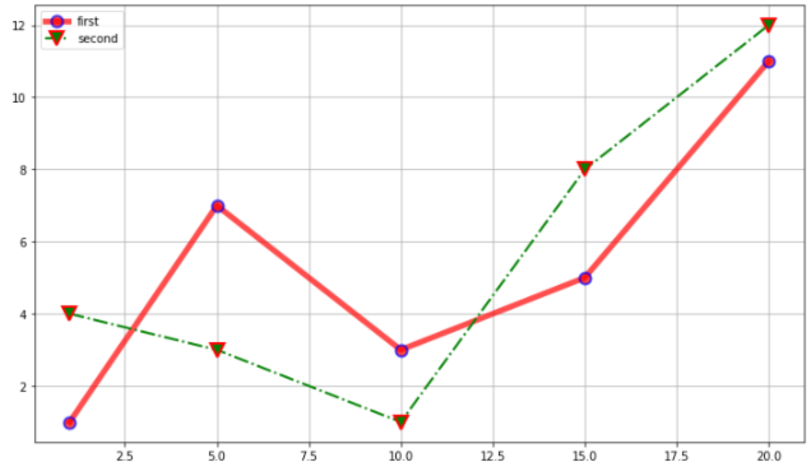
Ставрополь, 2022 г.

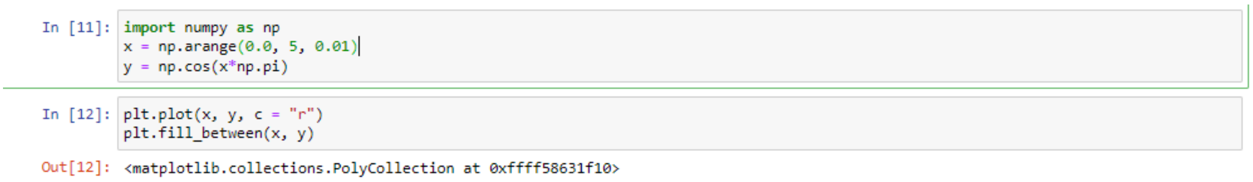
Примеры



```
In [9]:  import matplotlib.pyplot as plt
         %matplotlib inline
```

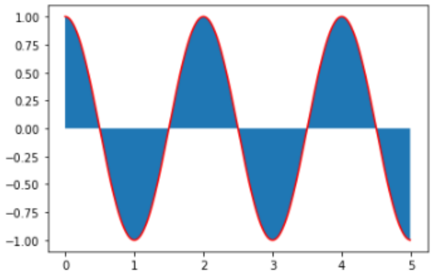
```
In [10]: x = [1, 5, 10, 15, 20]
         y1 = [1, 7, 3, 5, 11]
         y2 = [4, 3, 1, 8, 12]
         plt.figure(figsize=(12, 7))
         plt.plot(x, y1, 'o-r', alpha=0.7, label="first", lw=5, mec='b', mew=2, ms=10)
         plt.plot(x, y2, 'v-.g', label="second", mec='r', lw=2, mew=2, ms=12)
         plt.legend()
         plt.grid(True)
```

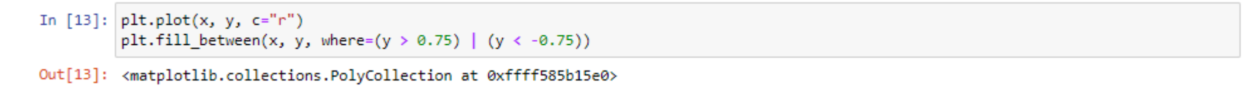




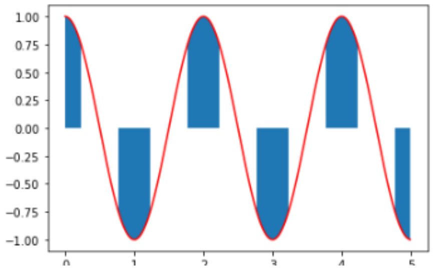
```
In [11]: import numpy as np
         x = np.arange(0.0, 5, 0.01)
         y = np.cos(x*np.pi)
```

```
In [12]: plt.plot(x, y, c = "r")
         plt.fill_between(x, y)

Out[12]: <matplotlib.collections.PolyCollection at 0xffff58631f10>
```

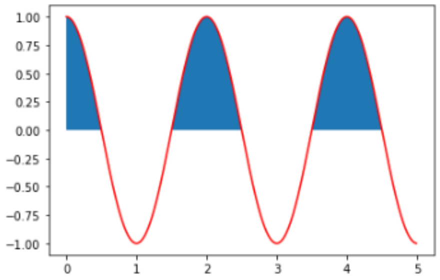




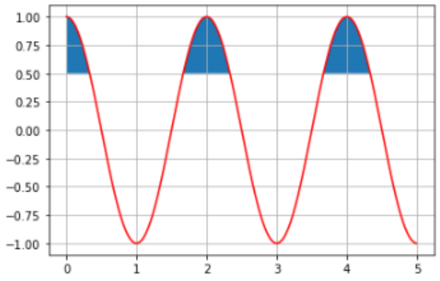
```
In [13]: plt.plot(x, y, c="r")
         plt.fill_between(x, y, where=(y > 0.75) | (y < -0.75))

Out[13]: <matplotlib.collections.PolyCollection at 0xffff585b15e0>
```



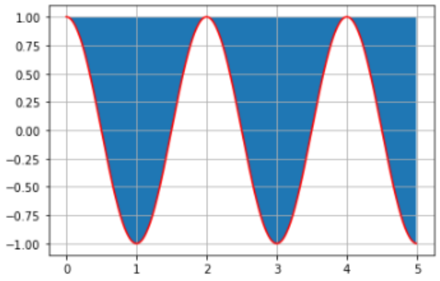
```
In [14]: plt.plot(x, y, c="r")
         plt.fill_between(x, y, where=(y > 0))

Out[14]: <matplotlib.collections.PolyCollection at 0xffff584719a0>
```



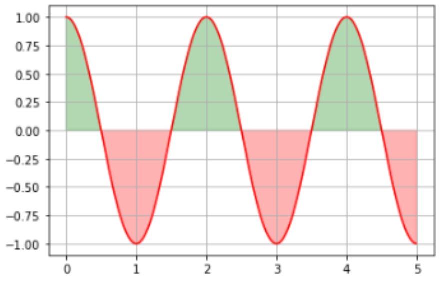
```
In [15]: plt.plot(x, y, c="r")
         plt.grid()
         plt.fill_between(x, 0.5, y, where=(y>=0.5))

Out[15]: <matplotlib.collections.PolyCollection at 0xffff583e6190>
```



```
In [16]: plt.plot(x, y, c="r")
         plt.grid()
         plt.fill_between(x, y, 1)

Out[16]: <matplotlib.collections.PolyCollection at 0xffff583ceeb0>
```

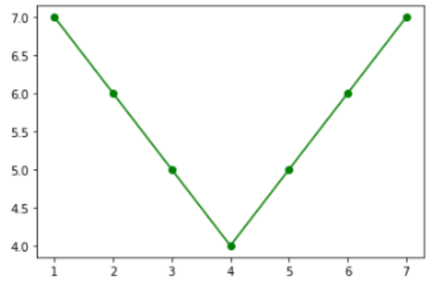


```
In [17]: plt.plot(x, y, c="r")
         plt.grid()
         plt.fill_between(x, y, where=y>=0, color="g", alpha=0.3)
         plt.fill_between(x, y, where=y<=0, color="r", alpha=0.3)

Out[17]: <matplotlib.collections.PolyCollection at 0xffff583508b0>
```



```
In [18]: x = [1, 2, 3, 4, 5, 6, 7]
         y = [7, 6, 5, 4, 5, 6, 7]
         plt.plot(x, y, marker="o", c="g")
```

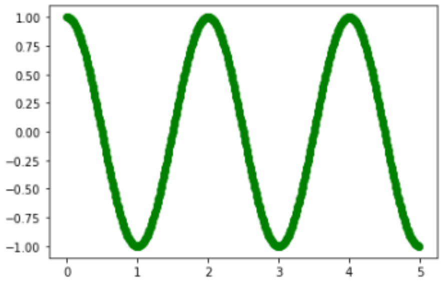
```
Out[18]: [<matplotlib.lines.Line2D at 0xffff582c9370>]
```



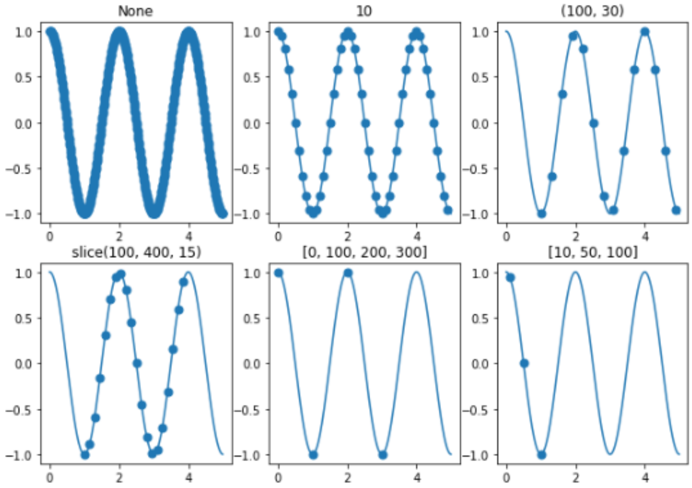
```
In [19]: import numpy as np
         x = np.arange(0.0, 5, 0.01)
         y = np.cos(x*np.pi)
```

```
In [20]: plt.plot(x, y, marker="o", c="g")
```

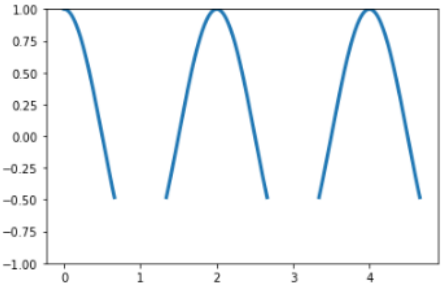
```
Out[20]: [<matplotlib.lines.Line2D at 0xffff582289d0>]
```



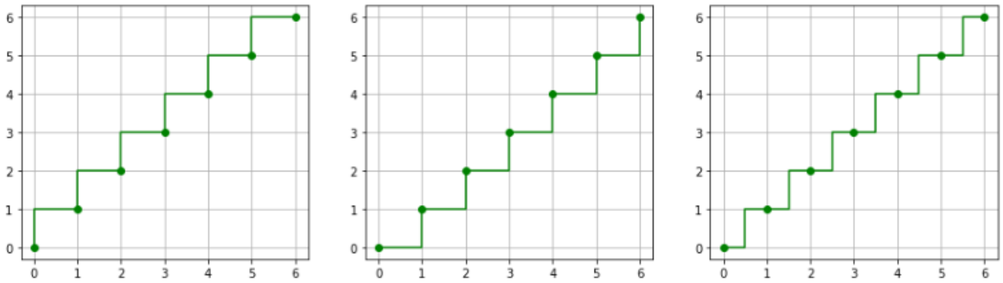
```
In [23]: x = np.arange(0.0, 5, 0.01)
         y = np.cos(x * np.pi)
         m_ev_case = [None, 10, (100, 30), slice(100,400,15), [0, 100, 200, 300], [10,
         50, 100]]
         fig, ax = plt.subplots(2, 3, figsize=(10, 7))
         axs = [ax[i, j] for i in range(2) for j in range(3)]
         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)
```



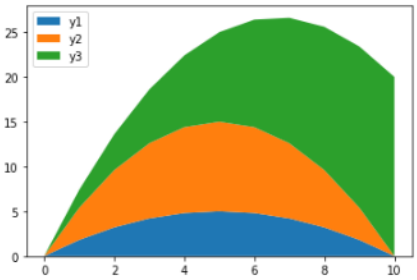
```
In [24]: x = np.arange(0.0, 5, 0.01)
         y = np.cos(x * np.pi)
         y_masked = np.ma.masked_where(y < -0.5, y)
         plt.ylim(-1, 1)
         plt.plot(x, y_masked, linewidth=3)

Out[24]: [<matplotlib.lines.Line2D at 0xffff53ef2940>]
```



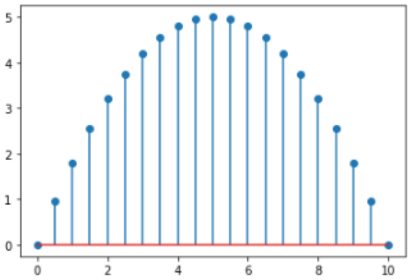
```
In [26]: x = np.arange(0, 7)
         y = x
         where_set = ['pre', 'post', 'mid']
         fig, axs = plt.subplots(1, 3, figsize=(15, 4))
         for i, ax in enumerate(axs):
             ax.step(x, y, "g-o", where=where_set[i])
             ax.grid()
```



```
In [27]: 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')

Out[27]: <matplotlib.legend.Legend at 0xffff53d6e550>
```

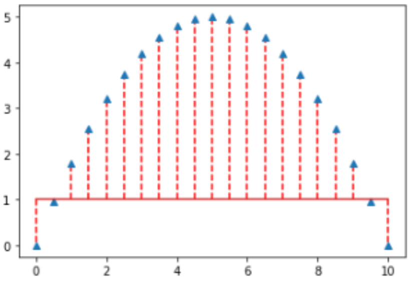


```
In [28]: x = np.arange(0, 10.5, 0.5)
         y = np.array([(-0.2)*i**2+2*i for i in x])
         plt.stem(x, y)

Out[28]: <StemContainer object of 3 artists>
```



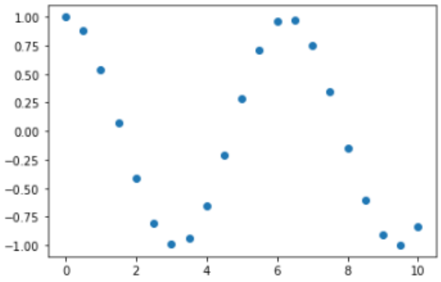
```
In [29]: plt.stem(x, y, linefmt="r--", markerfmt="^", bottom=1)
```

```
Out[29]: <StemContainer object of 3 artists>
```



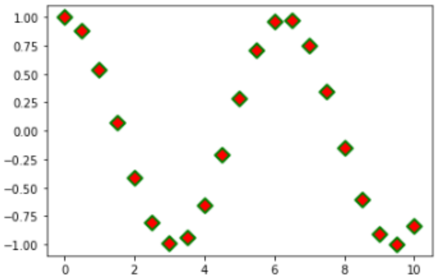
```
In [30]: x = np.arange(0, 10.5, 0.5)
         y = np.cos(x)
         plt.scatter(x, y)
```

```
Out[30]: <matplotlib.collections.PathCollection at 0xffff586a0c10>
```



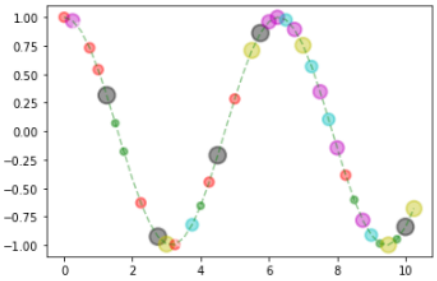
```
In [31]: x = np.arange(0, 10.5, 0.5)
         y = np.cos(x)
         plt.scatter(x, y, s=80, c="r", marker="D", linewidths=2, edgecolors="g")
```

```
Out[31]: <matplotlib.collections.PathCollection at 0xffff58140d90>
```



```
In [32]: import matplotlib.colors as mcolors
         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)
```

```
Out[32]: [<matplotlib.lines.Line2D at 0xffff58664520>]
```

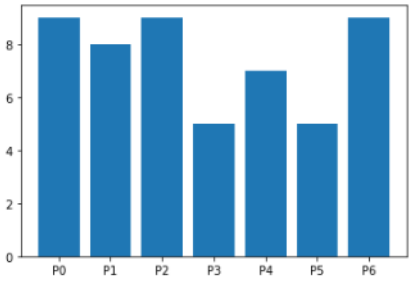


In [33]:

```
np.random.seed(123)
groups = [f"P{i}" for i in range(7)]
counts = np.random.randint(3, 10, len(groups))
plt.bar(groups, counts)
```

Out[33]:

```
<BarContainer object of 7 artists>
```

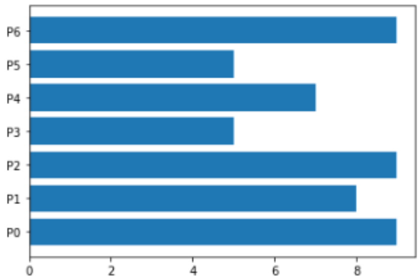


In [34]:

```
plt.barh(groups, counts)
```

Out[34]:

```
<BarContainer object of 7 artists>
```

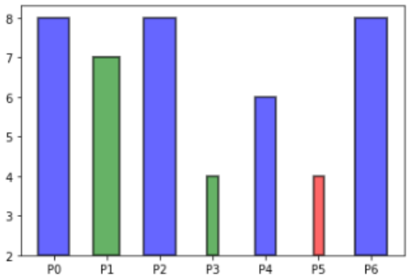


In [35]:

```
import matplotlib.colors as mcolors
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)
```

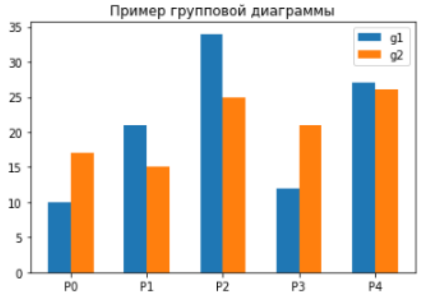
Out[35]:

```
<BarContainer object of 7 artists>
```



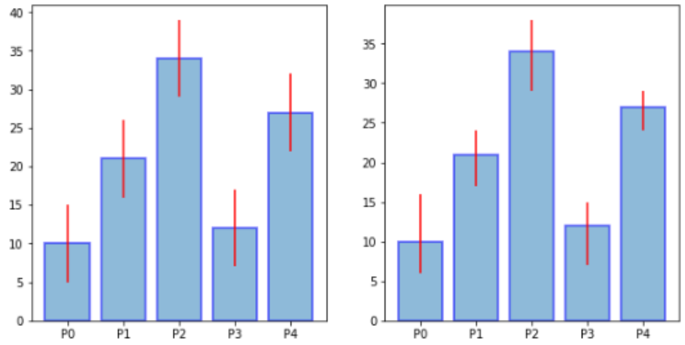
```
In [36]: cat_par = [f"P{i}" for i in range(5)]
         g1 = [10, 21, 34, 12, 27]
         g2 = [17, 15, 25, 21, 26]
         width = 0.3
         x = np.arange(len(cat_par))
         fig, ax = plt.subplots()
         rects1 = ax.bar(x - width/2, g1, width, label='g1')
         rects2 = ax.bar(x + width/2, g2, width, label='g2')
         ax.set_title('Пример групповой диаграммы')
         ax.set_xticks(x)
         ax.set_xticklabels(cat_par)
         ax.legend()
```

```
Out[36]: <matplotlib.legend.Legend at 0xffff581a77f0>
```



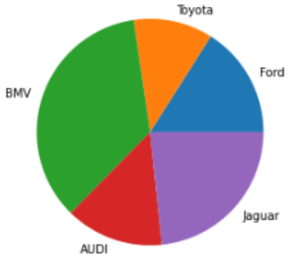
```
In [37]: np.random.seed(123)
         rnd = np.random.randint
         cat_par = [f"P{i}" for i in range(5)]
         g1 = [10, 21, 34, 12, 27]
         error = np.array([[rnd(2,7),rnd(2,7)] for _ in range(len(cat_par))]).T
         fig, axs = plt.subplots(1, 2, figsize=(10, 5))
         axs[0].bar(cat_par, 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)
```

```
Out[37]: <BarContainer object of 5 artists>
```



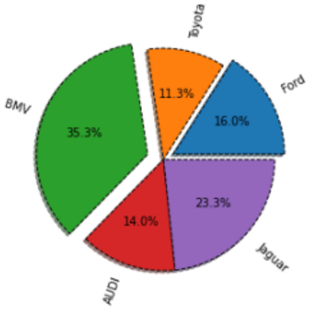
```
In [38]: vals = [24, 17, 53, 21, 35]
         labels = ["Ford", "Toyota", "BMV", "AUDI", "Jaguar"]
         fig, ax = plt.subplots()
         ax.pie(vals, labels=labels)
         ax.axis("equal")
```

```
Out[38]: (-1.1163226287452406,
          1.1007772680354877,
          -1.1107362350259515,
          1.1074836529113834)
```



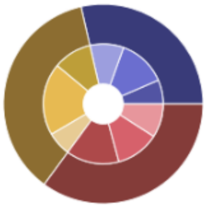
```
In [39]: vals = [24, 17, 53, 21, 35]
         labels = ["Ford", "Toyota", "BMV", "AUDI", "Jaguar"]
         explode = (0.1, 0, 0.15, 0, 0)
         fig, ax = plt.subplots()
         ax.pie(vals, labels=labels, autopct='%1.1f%%', shadow=True, explode=explode,
         wedgeprops={'lw':1, 'ls':'--','edgecolor':"k"}, rotatelabels=True)
         ax.axis("equal")
```

```
Out[39]: (-1.2704955621219602,
          1.1999223938155328,
          -1.1121847055183558,
          1.1379015332518725)
```



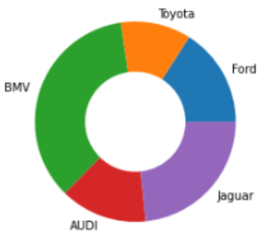
```
In [40]: fig, ax = plt.subplots()
         offset=0.4
         data = np.array([[5, 10, 7], [8, 15, 5], [11, 9, 7]])
         cmap = plt.get_cmap("tab20b")
         b_colors = cmap(np.array([0, 8, 12]))
         sm_colors = cmap(np.array([1, 2, 3, 9, 10, 11, 13, 14, 15]))
         ax.pie(data.sum(axis=1), 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'))
```

```
Out[40]: ([<matplotlib.patches.Wedge at 0xffff53dbed30>,
           <matplotlib.patches.Wedge at 0xffff53e437c0>,
           <matplotlib.patches.Wedge at 0xffff53dde280>,
           <matplotlib.patches.Wedge at 0xffff53dde4c0>,
           <matplotlib.patches.Wedge at 0xffff53dde250>,
           <matplotlib.patches.Wedge at 0xffff53db58b0>,
           <matplotlib.patches.Wedge at 0xffff53db5f70>,
           <matplotlib.patches.Wedge at 0xffff53db5460>,
           <matplotlib.patches.Wedge at 0xffff581e1760>],
          [Text(0.646314344414094, 0.13370777166859046, ''),
           Text(0.4521935266177387, 0.4807504700829865, ''),
           Text(0.040366679721656945, 0.6587643973138266, ''),
           Text(-0.34542288787409087, 0.5623904591409097, ''),
           Text(-0.6578039053946477, 0.05379611554331286, ''),
           Text(-0.48987451889717687, -0.44229283934431896, ''),
           Text(-0.12049606360635531, -0.6489073112975174, ''),
           Text(0.39011356818311405, -0.53236397691752 1, ''),
           Text(0.6332653697075483, -0.1859434632601054, '')])
```



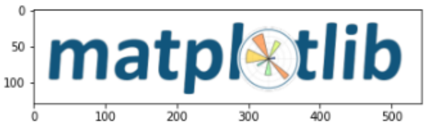
```
In [41]: vals = [24, 17, 53, 21, 35]
         labels = ["Ford", "Toyota", "BMV", "AUDI", "Jaguar"]
         fig, ax = plt.subplots()
         ax.pie(vals, labels=labels, wedgeprops=dict(width=0.5))
```

```
Out[41]: ([<matplotlib.patches.Wedge at 0xffff53ba7ee0>,
           <matplotlib.patches.Wedge at 0xffff53e07220>,
           <matplotlib.patches.Wedge at 0xffff53e07700>,
           <matplotlib.patches.Wedge at 0xffff53e07be0>,
           <matplotlib.patches.Wedge at 0xffff53e12100>],
          [Text(0.9639373540021144, 0.5299290306818474, 'Ford'),
           Text(0.22870287165240302, 1.075962358309037, 'Toyota'),
           Text(-1.046162158377023, 0.3399187231970734, 'BMV'),
           Text(-0.3617533684721028, -1.0388139873909512, 'AUDI'),
           Text(0.8174592712713289, -0.7360437078139777, 'Jaguar')])
```



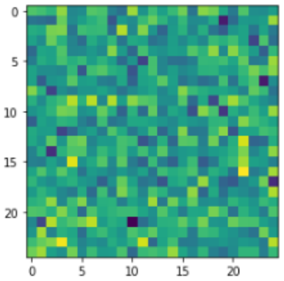
```
In [45]: from PIL import Image
         import requests
         from io import BytesIO
         response = requests.get('https://matplotlib.org/_static/logo2.png')
         img = Image.open(BytesIO(response.content))
         plt.imshow(img)
```

```
Out[45]: <matplotlib.image.AxesImage at 0xffff5325d1c0>
```



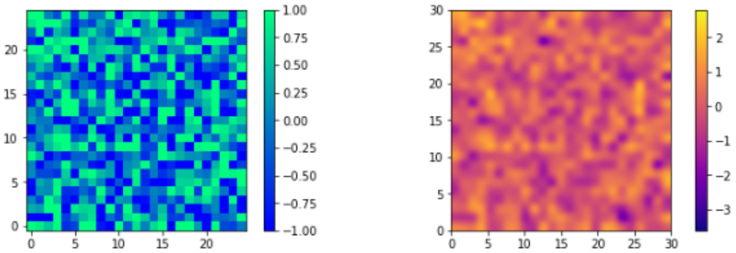
```
In [46]: np.random.seed(19680801)
         data = np.random.randn(25, 25)
         plt.imshow(data)
```

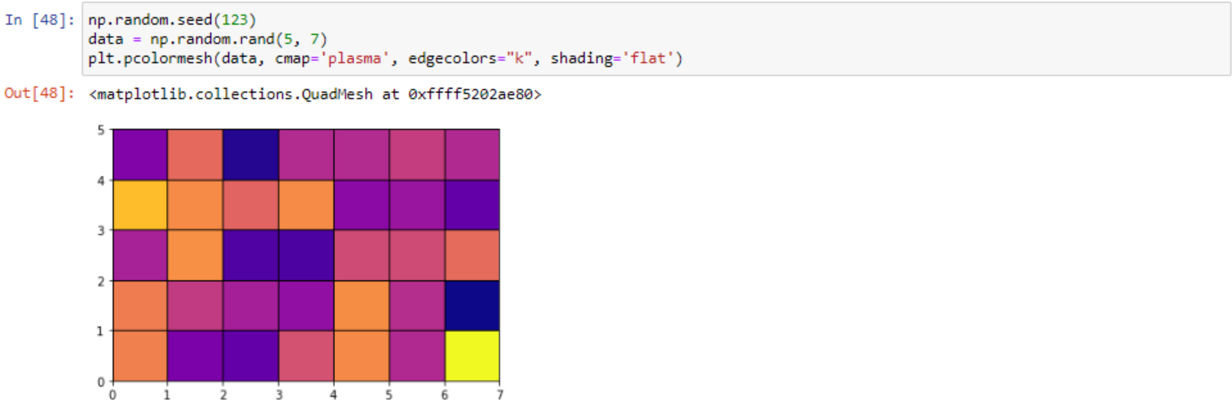
```
Out[46]: <matplotlib.image.AxesImage at 0xffff53233ee0>
```



```
In [47]: fig, axs = plt.subplots(1, 2, figsize=(10,3), constrained_layout=True)
         p1 = axs[0].imshow(data, cmap='winter', aspect='equal', vmin=-1, vmax=1,
         origin="lower")
         fig.colorbar(p1, ax=axs[0])
         p2 = axs[1].imshow(data, cmap='plasma', aspect='equal',
         interpolation='gaussian', origin="lower", extent=(0, 30, 0, 30))
         fig.colorbar(p2, ax=axs[1])
```

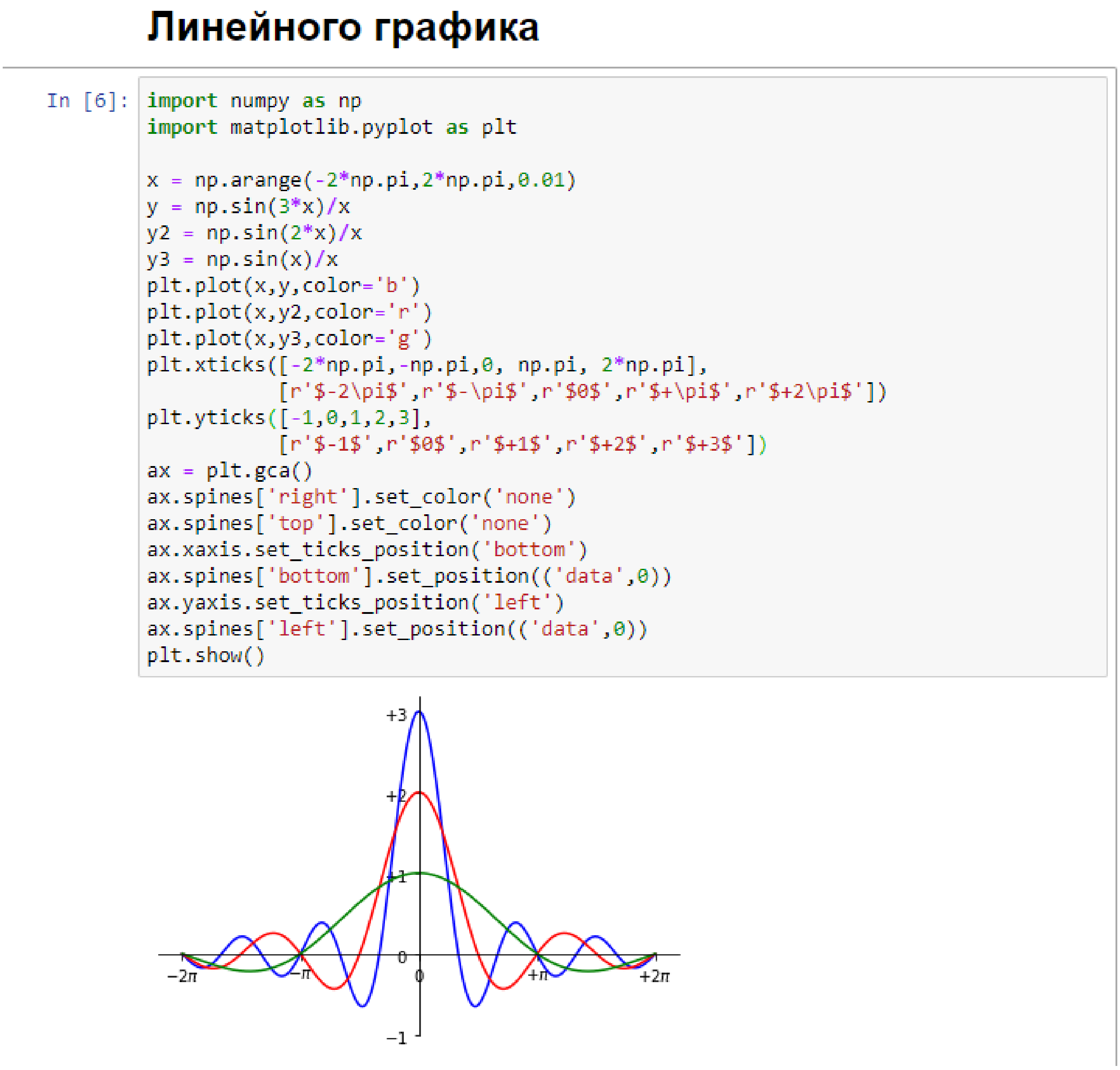
```
Out[47]: <matplotlib.colorbar.Colorbar at 0xffff531f5d00>
```





```
In [48]: np.random.seed(123)
         data = np.random.rand(5, 7)
         plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')

Out[48]: <matplotlib.collections.QuadMesh at 0xffff5202ae80>
```

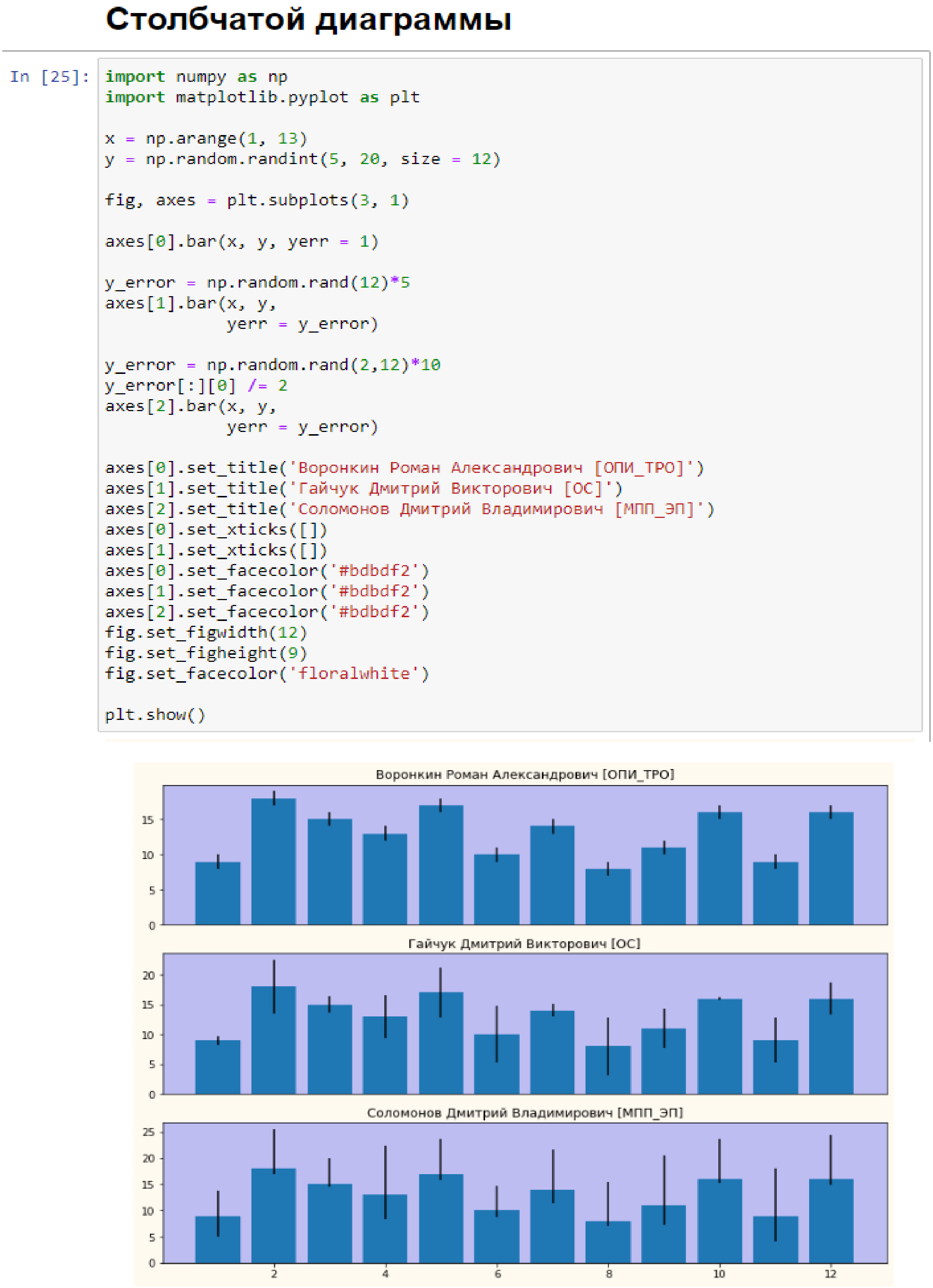
Линейного графика



Линейного графика

```
In [6]: import numpy as np
        import matplotlib.pyplot as plt

        x = np.arange(-2*np.pi,2*np.pi,0.01)
        y = np.sin(3*x)/x
        y2 = np.sin(2*x)/x
        y3 = np.sin(x)/x
        plt.plot(x,y,color='b')
        plt.plot(x,y2,color='r')
        plt.plot(x,y3,color='g')
        plt.xticks([-2*np.pi,-np.pi,0, np.pi, 2*np.pi],
                   [r'$-2\pi$',r'$-\pi$',r'$0$',r'$+\pi$',r'$+2\pi$'])
        plt.yticks([-1,0,1,2,3],
                   [r'$-1$',r'$0$',r'$+1$',r'$+2$',r'$+3$'])
        ax = plt.gca()
        ax.spines['right'].set_color('none')
        ax.spines['top'].set_color('none')
        ax.xaxis.set_ticks_position('bottom')
        ax.spines['bottom'].set_position(('data',0))
        ax.yaxis.set_ticks_position('left')
        ax.spines['left'].set_position(('data',0))
        plt.show()
```

Столбчатой диаграммы

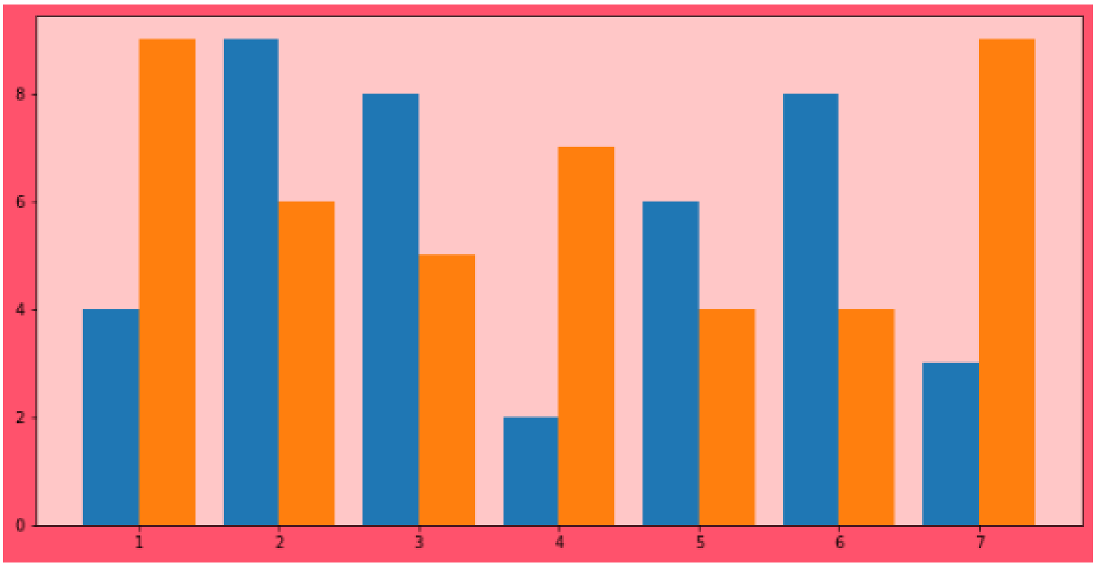


Столбчатой диаграммы

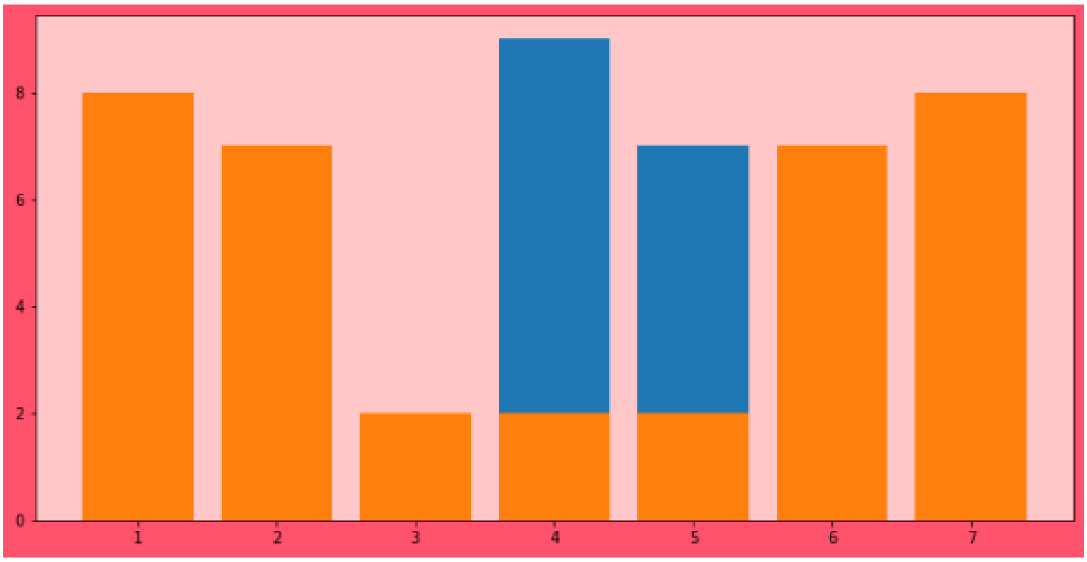
```
In [25]: import numpy as np
         import matplotlib.pyplot as plt

         x = np.arange(1, 13)
         y = np.random.randint(5, 20, size = 12)

         fig, axes = plt.subplots(3, 1)

         axes[0].bar(x, y, yerr = 1)

         y_error = np.random.rand(12)*5
         axes[1].bar(x, y,
                     yerr = y_error)

         y_error = np.random.rand(2,12)*10
         y_error[:][0] /= 2
         axes[2].bar(x, y,
                     yerr = y_error)

         axes[0].set_title('Воронкин Роман Александрович [ОПИ_ТРО]')
         axes[1].set_title('Гайчук Дмитрий Викторович [ОС]')
         axes[2].set_title('Соломонов Дмитрий Владимирович [МПП_ЭП]')
         axes[0].set_xticks([])
         axes[1].set_xticks([])
         axes[0].set_facecolor('#bdbdf2')
         axes[1].set_facecolor('#bdbdf2')
         axes[2].set_facecolor('#bdbdf2')
         fig.set_figwidth(12)
         fig.set_figheight(9)
         fig.set_facecolor('floralwhite')

         plt.show()
```

Воронкин Роман Александрович [ОПИ_ТРО]
0
5
10
15
Гайчук Дмитрий Викторович [ОС]
0
5
10
15
20
Соломонов Дмитрий Владимирович [МПП_ЭП]
0
5
10
15
20
25
2
4
6
8
10
12

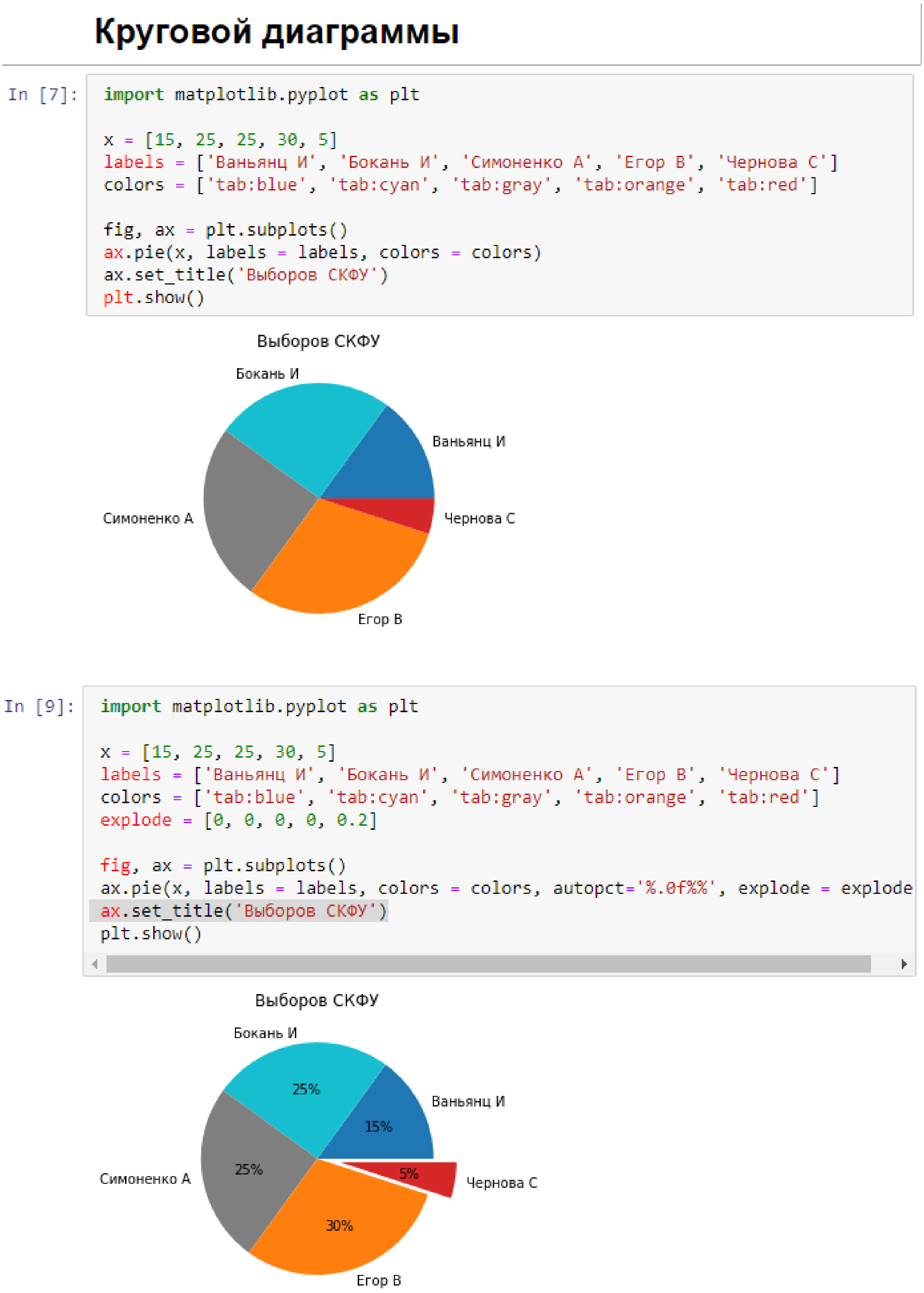
```
In [22]: import numpy as np
         import matplotlib.pyplot as plt

         x1 = np.arange(1, 8) - 0.2
         x2 = np.arange(1, 8) + 0.2
         y1 = np.random.randint(1, 10, size = 7)
         y2 = np.random.randint(1, 10, size = 7)

         fig, ax = plt.subplots()

         ax.bar(x1, y1, width = 0.4)
         ax.bar(x2, y2, width = 0.4)

         ax.set_facecolor('#ffc7c7')
         fig.set_figwidth(12)
         fig.set_figheight(6)
         fig.set_facecolor('#ff526c')

         plt.show()
```



```
In [24]: import numpy as np
         import matplotlib.pyplot as plt

         x = np.arange(1, 8)
         y1 = np.random.randint(1, 10, size = 7)
         y2 = np.random.randint(1, 10, size = 7)

         fig, ax = plt.subplots()

         ax.bar(x, y1)
         ax.bar(x, y2)

         ax.set_facecolor('#ffc7c7')
         fig.set_figwidth(12)
         fig.set_figheight(6)
         fig.set_facecolor('#ff526c')

         plt.show()
```



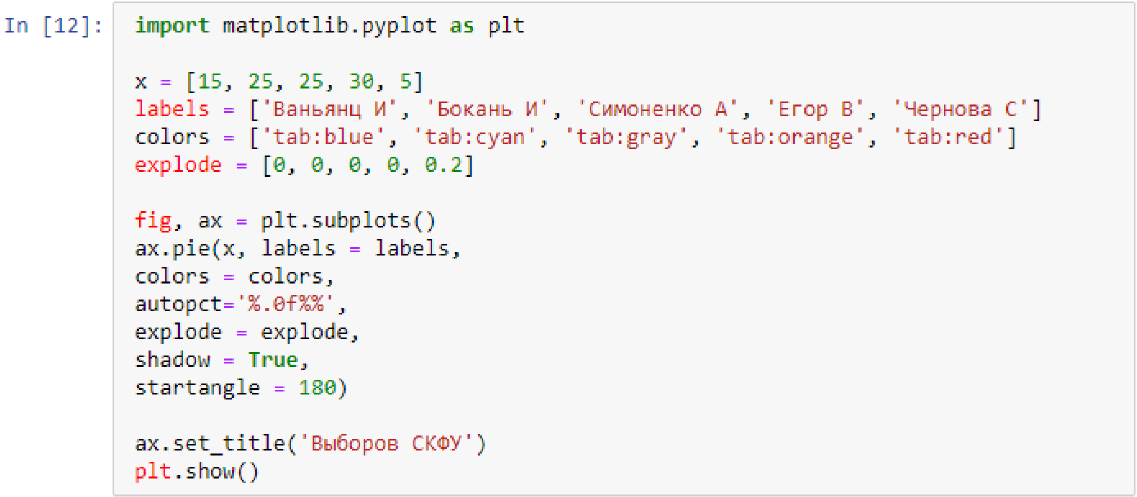
Круговой диаграммы



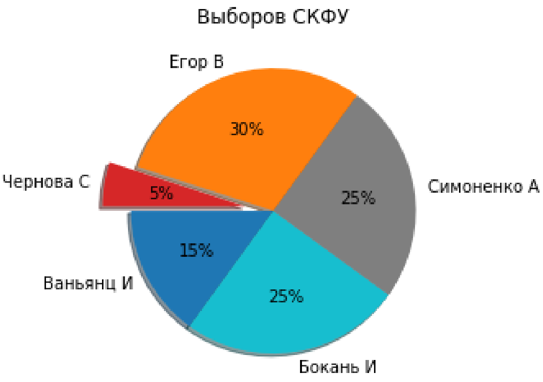
Круговой диаграммы

In [7]:
```
import matplotlib.pyplot as plt

x = [15, 25, 25, 30, 5]
labels = ['Ваньянц И', 'Бокань И', 'Симоненко А', 'Егор В', 'Чернова С']
colors = ['tab:blue', 'tab:cyan', 'tab:gray', 'tab:orange', 'tab:red']

fig, ax = plt.subplots()
ax.pie(x, labels = labels, colors = colors)
ax.set_title('Выборов СКФУ')
plt.show()
```

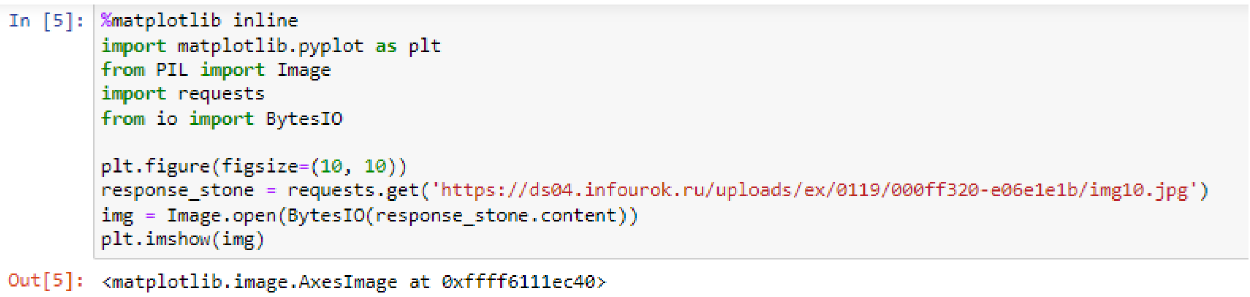
In [9]:
```
import matplotlib.pyplot as plt

x = [15, 25, 25, 30, 5]
labels = ['Ваньянц И', 'Бокань И', 'Симоненко А', 'Егор В', 'Чернова С']
colors = ['tab:blue', 'tab:cyan', 'tab:gray', 'tab:orange', 'tab:red']
explode = [0, 0, 0, 0, 0.2]

fig, ax = plt.subplots()
ax.pie(x, labels = labels, colors = colors, autopct='%.0f%%', explode = explode
ax.set_title('Выборов СКФУ')
plt.show()
```



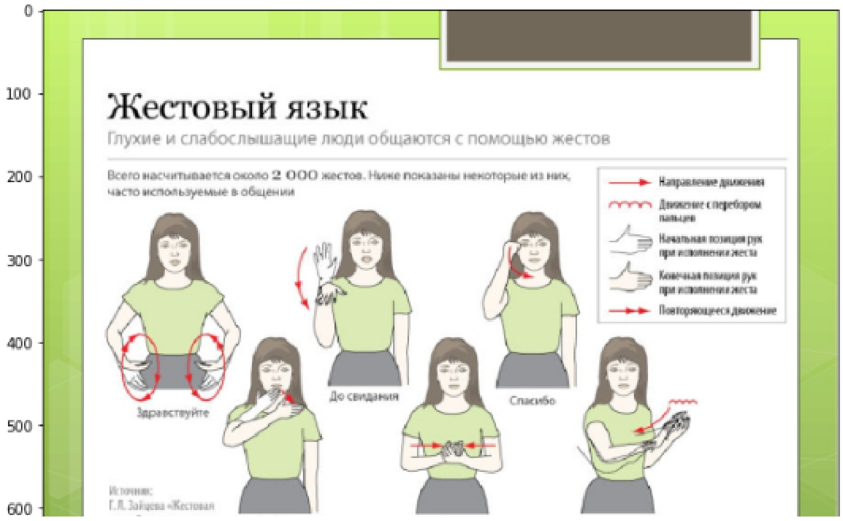
```
In [12]: import matplotlib.pyplot as plt

         x = [15, 25, 25, 30, 5]
         labels = ['Ваньянц И', 'Бокань И', 'Симоненко А', 'Егор В', 'Чернова С']
         colors = ['tab:blue', 'tab:cyan', 'tab:gray', 'tab:orange', 'tab:red']
         explode = [0, 0, 0, 0, 0.2]

         fig, ax = plt.subplots()
         ax.pie(x, labels = labels,
         colors = colors,
         autopct='%.0f%%',
         explode = explode,
         shadow = True,
         startangle = 180)

         ax.set_title('Выборов СКФУ')
         plt.show()
```



Изображение



```
In [5]: %matplotlib inline
        import matplotlib.pyplot as plt
        from PIL import Image
        import requests
        from io import BytesIO

        plt.figure(figsize=(10, 10))
        response_stone = requests.get('https://ds04.infourok.ru/uploads/ex/0119/000ff320-e06e1e1b/img10.jpg')
        img = Image.open(BytesIO(response_stone.content))
        plt.imshow(img)

Out[5]: <matplotlib.image.AxesImage at 0xffff6111ec40>
```


0
100
200
300
400
500
600
Жестовый язык
Глухие и слабослышащие люди общаются с помощью жестов
Всего насчитывается около 2 000 жестов. Ниже показаны некоторые из них, часто используемые в общении
Направление движения
Движение с перебором пальцев
Начальная позиция рук при исполнении жеста
Конечная позиция рук при исполнении жеста
Повторяющееся движение
Здравствуйте
До свидания
Спасибо
Источник:
Г.Л. Зайцева «Жестовая