### **Задание1**

Отобразите зависимости косинуса и синуса на одном графике. Воспользуйтесь шаблонами формата линии:

* треугольники ^;
* квадраты s.

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

x = np.linspace(10, 20, 50)

plt.plot(x, np.sin(x), 'k^', label='y=sinx')

plt.plot(x, np.cos(x), 'bs', label='y=cosx')

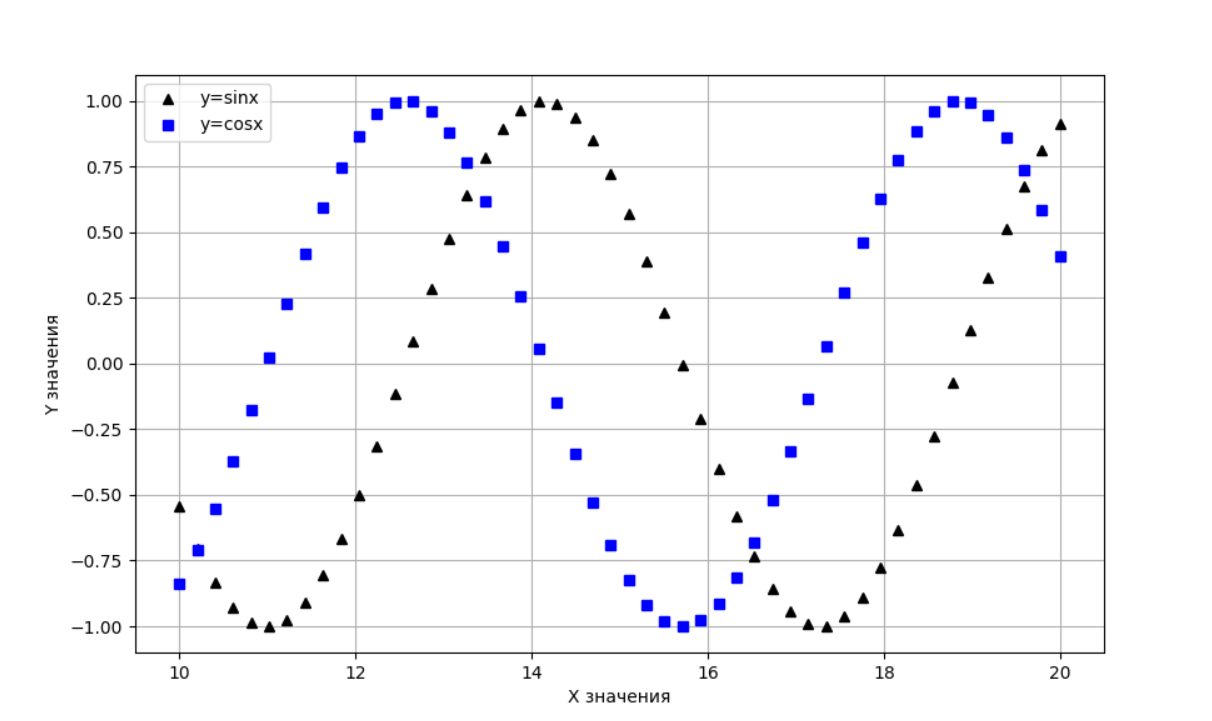
plt.xlabel('X значения')

plt.ylabel('Y значения')

plt.grid(True)

plt.legend()

plt.show()



### **Задание2**

Отобразите сгенерированные данные в виде точек с помощью функции plt.scatter():

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

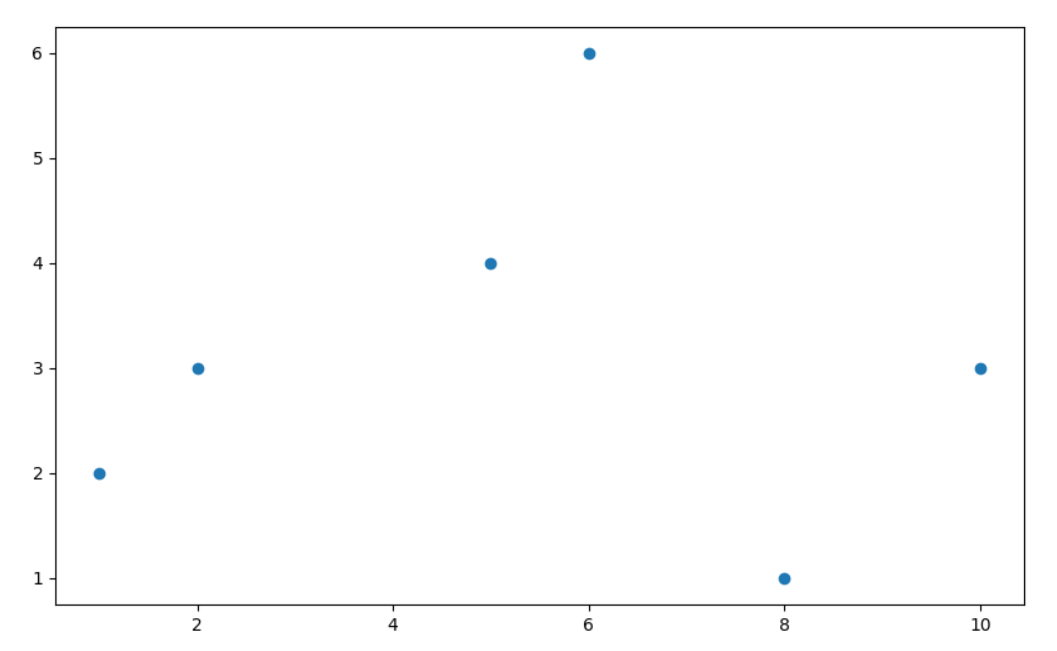
from matplotlib import pyplot as plt

x = np.array([1, 2, 5, 8, 6, 10])

y = np.array([2, 3, 4, 1, 6, 3])

plt.scatter(x,y)

plt.show()



### **Задание**

Нарисуйте сеткой IMG_256 графики прямых с коэффициентами наклона от 2 до 7 включительно.

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

x = np.linspace(0, 10, 50)

for i in range(2,8):

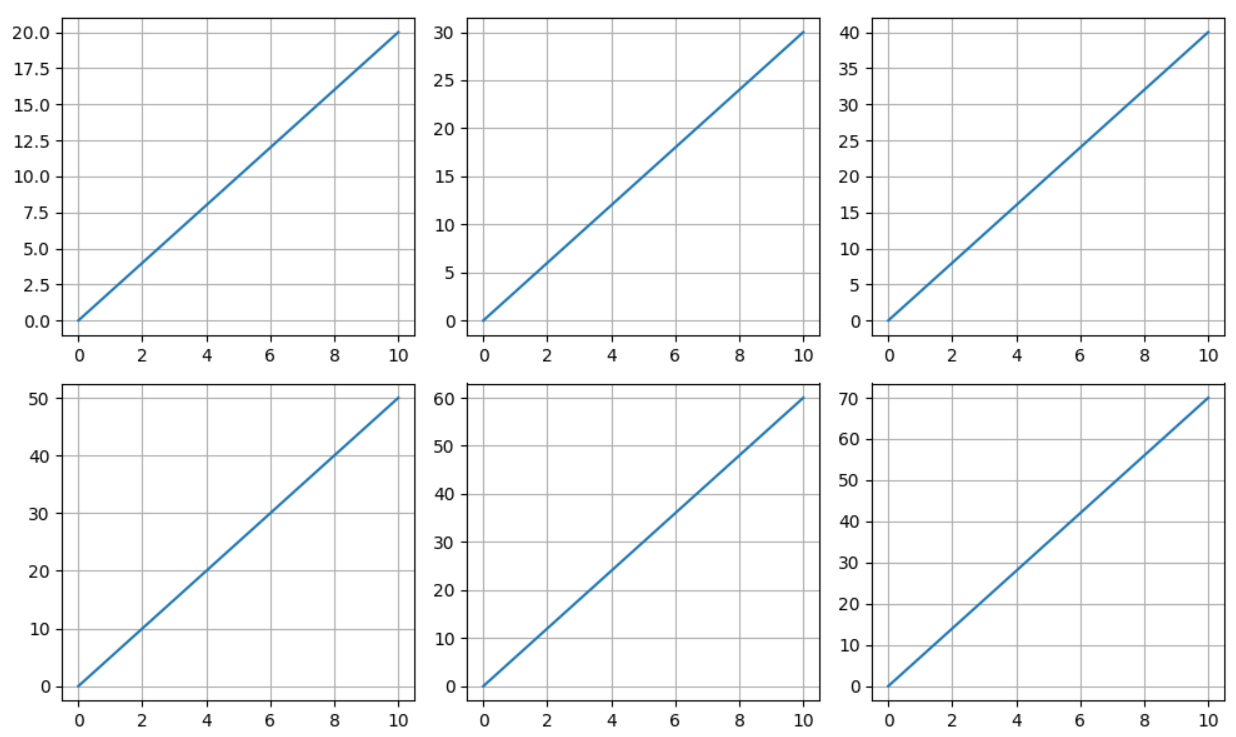
plt.subplot(2,3,i-1)

plt.plot(x, x\*i)

plt.grid(True)

plt.tight\_layout()

plt.show()



### **Задание**

Нарисуйте графики функций np.log1p() и np.log() в диапазоне [0.1,10]. Задайте размер области отображения:(15,9) .

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

x = np.linspace(0.1,10)

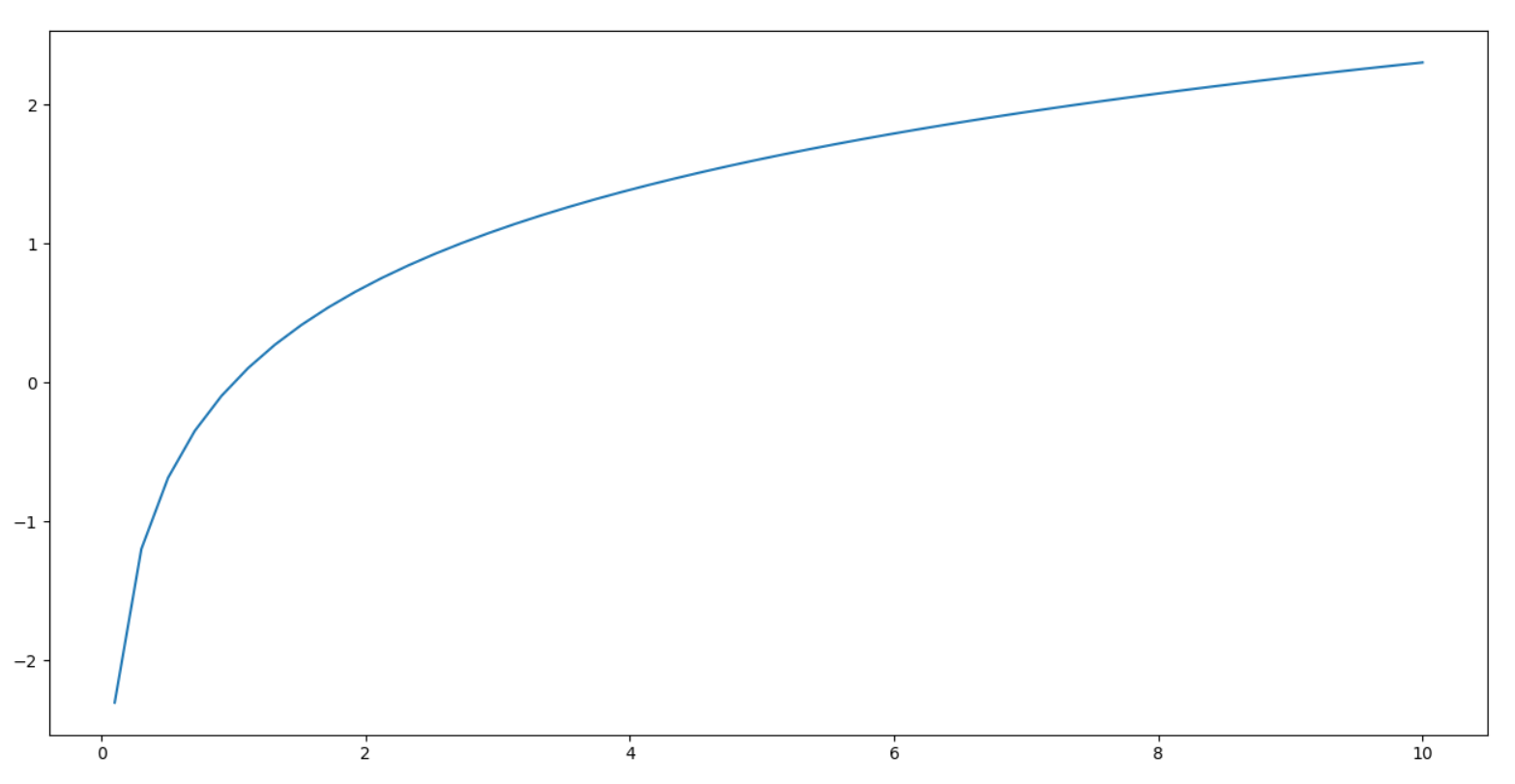
plt.figure(figsize=(15, 9))

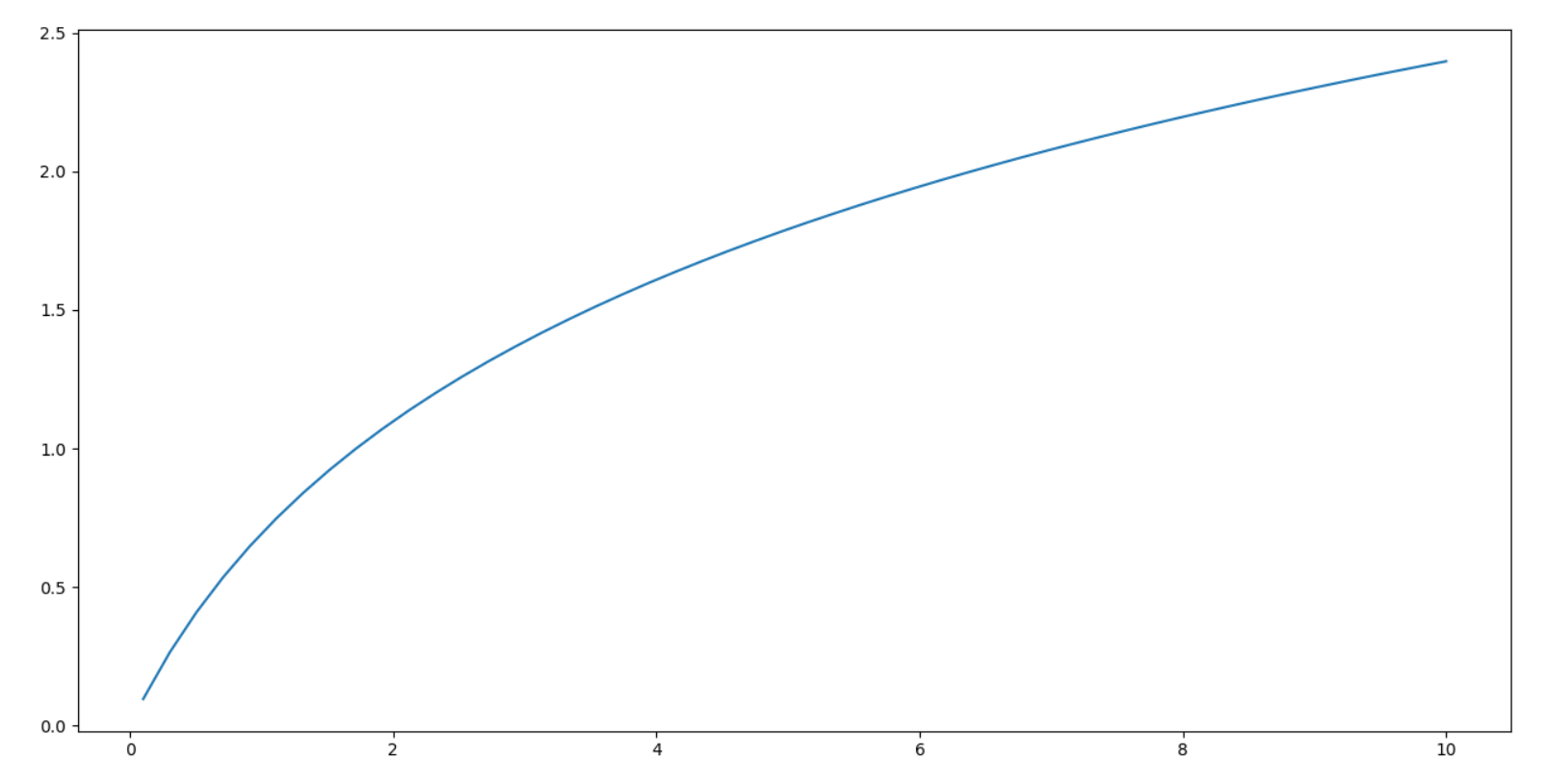
plt.plot(x,np.log1p(x))

plt.figure(figsize=(15, 9))

plt.plot(x,np.log(x))

plt.show()





### **Задание**

Отобразите функцию параболы  в диапазоне x[-3;3] и подпишите точку экстремума для k=[-1,0,1]. Необходимо нарисовать все три графика с подписями с помощью зон отображения.

import matplotlib as mpl

import numpy as np

import math

import matplotlib.colors as mcolors

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR = 'black'

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

def f(k,t):

return k-x\*\*2

x=np.linspace(-3,3,50)

k=[-1,0,1]

fig, ax = plt.subplots(nrows=1, ncols=3, squeeze=False)

ax[0,0].plot(x, f(k[0],x), 'r')

ax[0,0].annotate(

'Экстремум',

xy=(0,k[0]),

xytext=(1,k[0]+1),

arrowprops=dict(facecolor='black', shrink=0.05),

)

ax[0,0].text(x=0.6, y=k[0], s='(0,-1)')

ax[0,0].grid(True)

ax[0,1].plot(x, f(k[1],x), 'r')

ax[0,1].annotate(

'Экстремум',

xy=(0,k[1]),

xytext=(1,k[1]+1),

arrowprops=dict(facecolor='blue', shrink=0.05),

)

ax[0,1].text(x=0.6, y=k[1], s='(0,0)')

ax[0,1].grid(True)

ax[0,2].plot(x, f(k[2],x), 'r')

ax[0,2].annotate(

'Экстремум',

xy=(0,k[2]),

xytext=(1,k[2]+1),

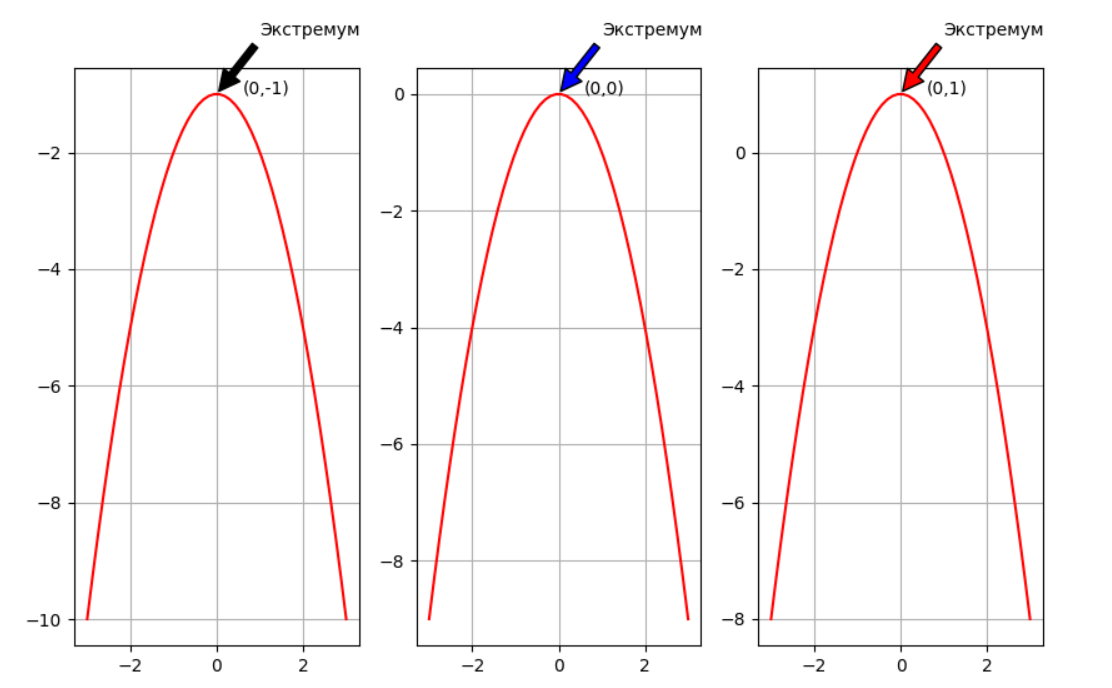
arrowprops=dict(facecolor='red', shrink=0.05),

)

ax[0,2].text(x=0.6, y=k[2], s='(0,1)')

ax[0,2].grid(True)

plt.show()



# **Задачки**

Отобразите две функции на одном графике:

y=sin(x) толщиной 2, красным цветом пунктиром;

y=cos(x)+1 толщиной 1, синим цветом крестиками;

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

x=np.linspace(0,20,500)

y1=np.sin(x)

y2=np.cos(x)+1

plt.plot(x,y1,'r--',linewidth=2)

plt.plot(x,y2,'bx',linewidth=1)

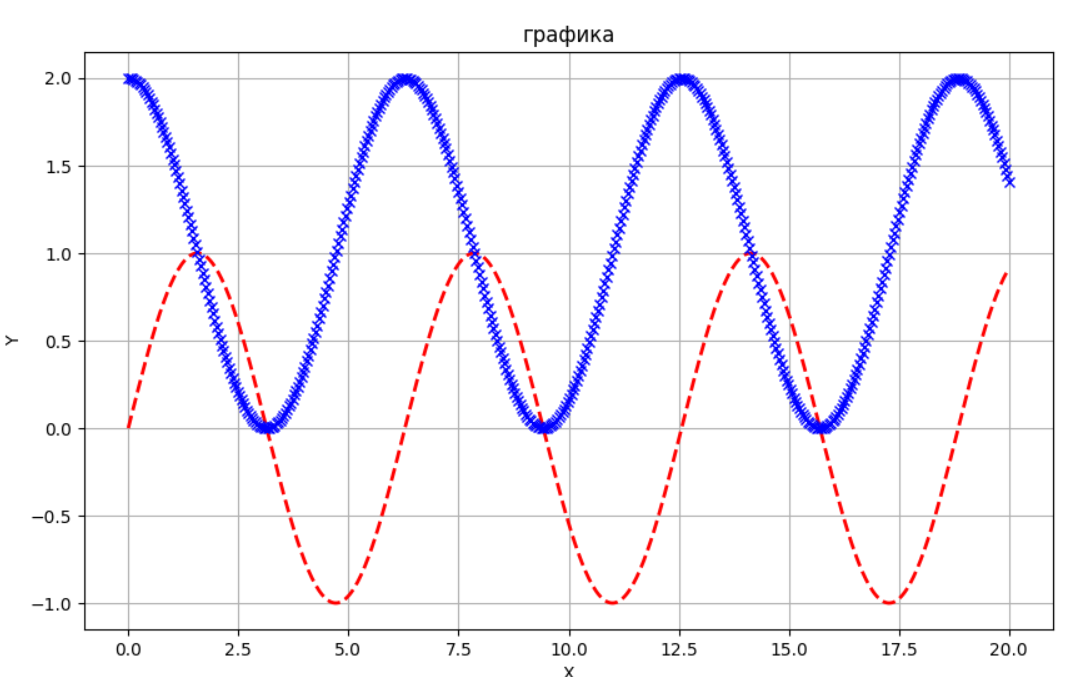
plt.grid()

plt.xlabel('X')

plt.ylabel('Y')

plt.title('графика')

plt.show()



Отобразите график, чтобы точки данных были синими с радиусом 3, при этом точки должны быть соединены зелеными пунктирными линиями:

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

x = np.array([1, 4, 5, 6, 7])

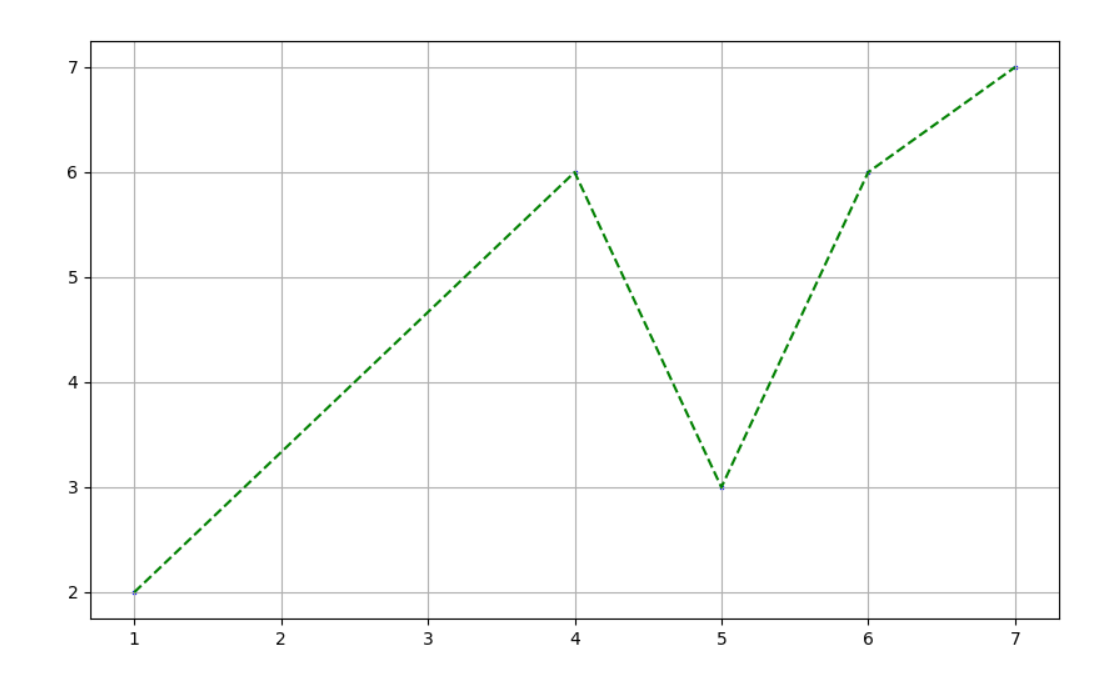
y = np.array([2, 6, 3, 6, 7])

plt.plot(x,y,'g--')

plt.scatter(x,y,c='b',s=3,marker='o')

plt.grid()

plt.show()



Нарисуйте данные графиками колонок (bar chart) на двух разных зонах отображения для каждой группы:

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

group1\_scores = (22, 30, 33, 30, 26)

group2\_scores = (25, 32, 30, 35, 29)

x\_labels = ['G1', 'G2', 'G3', 'G4', 'G5']

fig, ax = plt.subplots(nrows=1, ncols=2, squeeze=False)

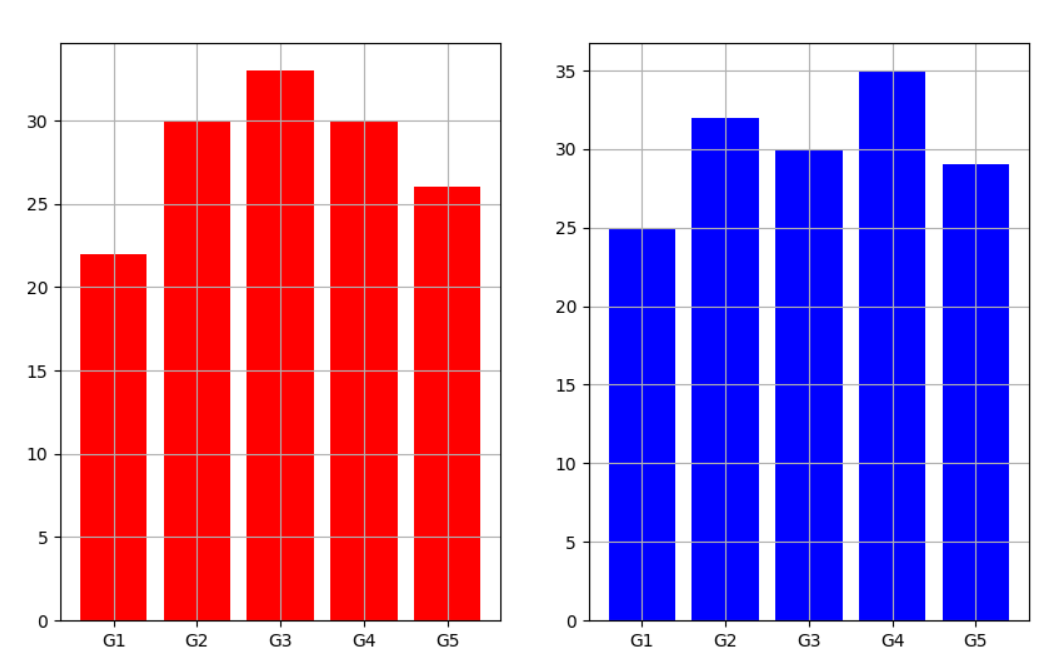
ax[0,0].bar(x\_labels,group1\_scores,color='red')

ax[0,0].grid()

ax[0,1].bar(x\_labels,group2\_scores,color='blue')

ax[0,1].grid()

plt.show()



Отобразите средние значения для каждой группы со стандартными отклонениями:

import matplotlib as mpl

import numpy as np

import math

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

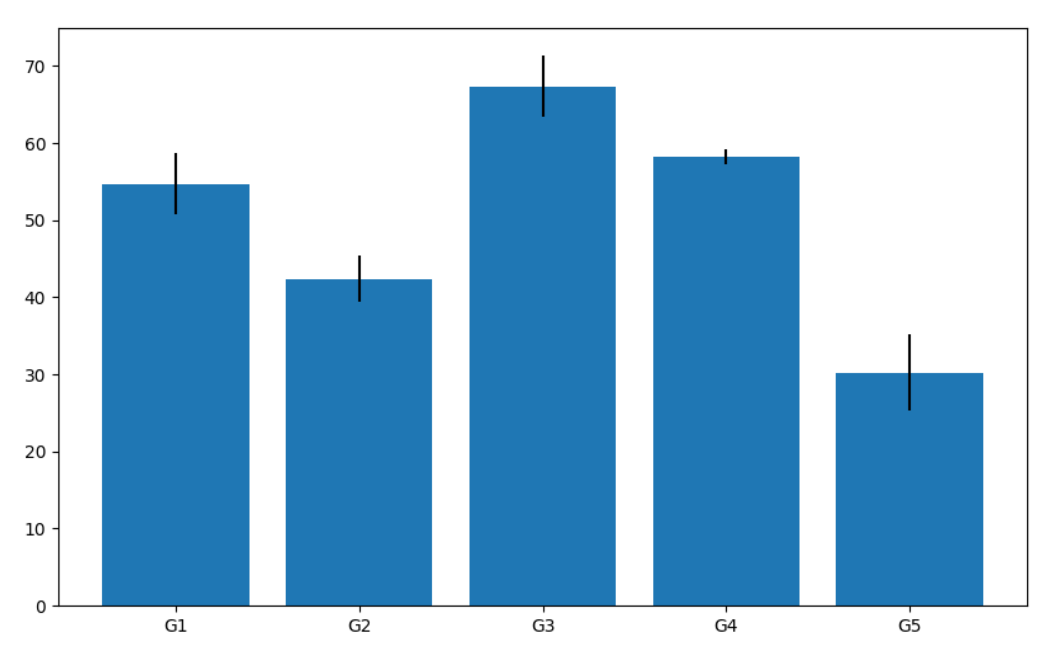
groups\_means = (54.74, 42.35, 67.37, 58.24, 30.25)

groups\_stds = (4, 3, 4, 1, 5)

x\_labels = ['G1', 'G2', 'G3', 'G4', 'G5']

plt.bar(x\_labels, groups\_means,yerr=groups\_stds)

plt.show()



Отобразите данные точечным графиком, в котором цвет зависит от номера группы groups, а размер точки зависит от показателя площади areas.

import matplotlib as mpl

import numpy as np

import math

import matplotlib.colors as mcolors

TEXT\_COLOR='black'

mpl.rcParams['figure.figsize']=(10,6)

mpl.rcParams['text.color']='black'

mpl.rcParams['axes.labelcolor']=TEXT\_COLOR

mpl.rcParams['xtick.color']=TEXT\_COLOR

mpl.rcParams['ytick.color']=TEXT\_COLOR

from matplotlib import pyplot as plt

n\_samples = 20

x = np.random.randint(0, 10, size=n\_samples)

y = np.random.randint(-10, 0, size=n\_samples)

areas = np.pi \* np.random.randint(5, 15, size=n\_samples)\*\*2

groups = np.random.randint(0, 3, size=n\_samples)

bc = mcolors.BASE\_COLORS

colors = [list(bc.keys())[i] for i in groups]

plt.scatter(x, y, s=areas, c=colors, linewidths=2)

plt.grid(True)

plt.show()

