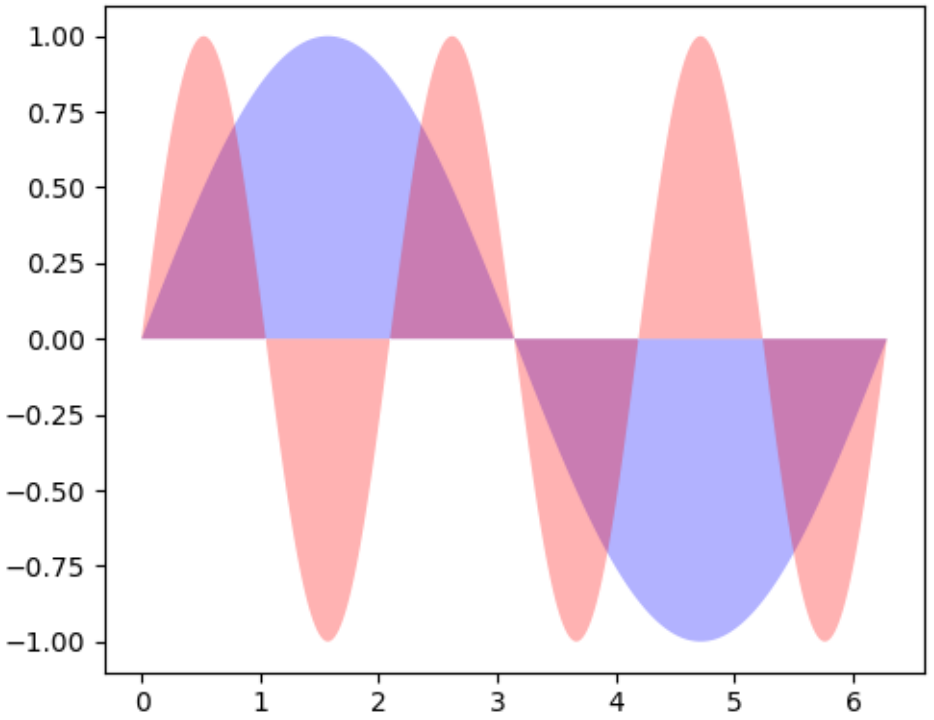
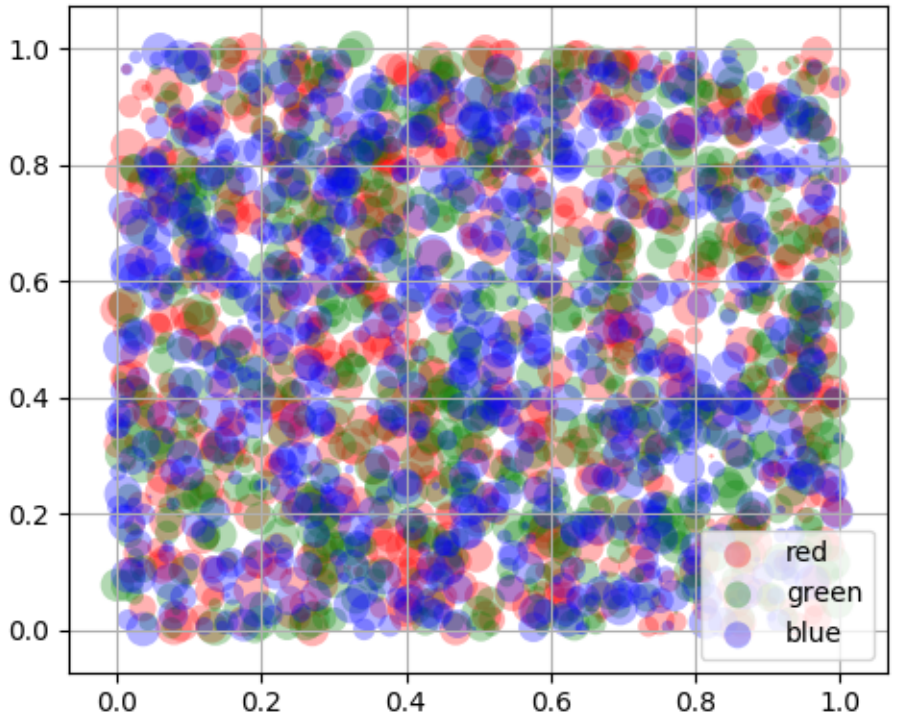
https://wiki.python.org/moin/NumericAndScientific/Plotting

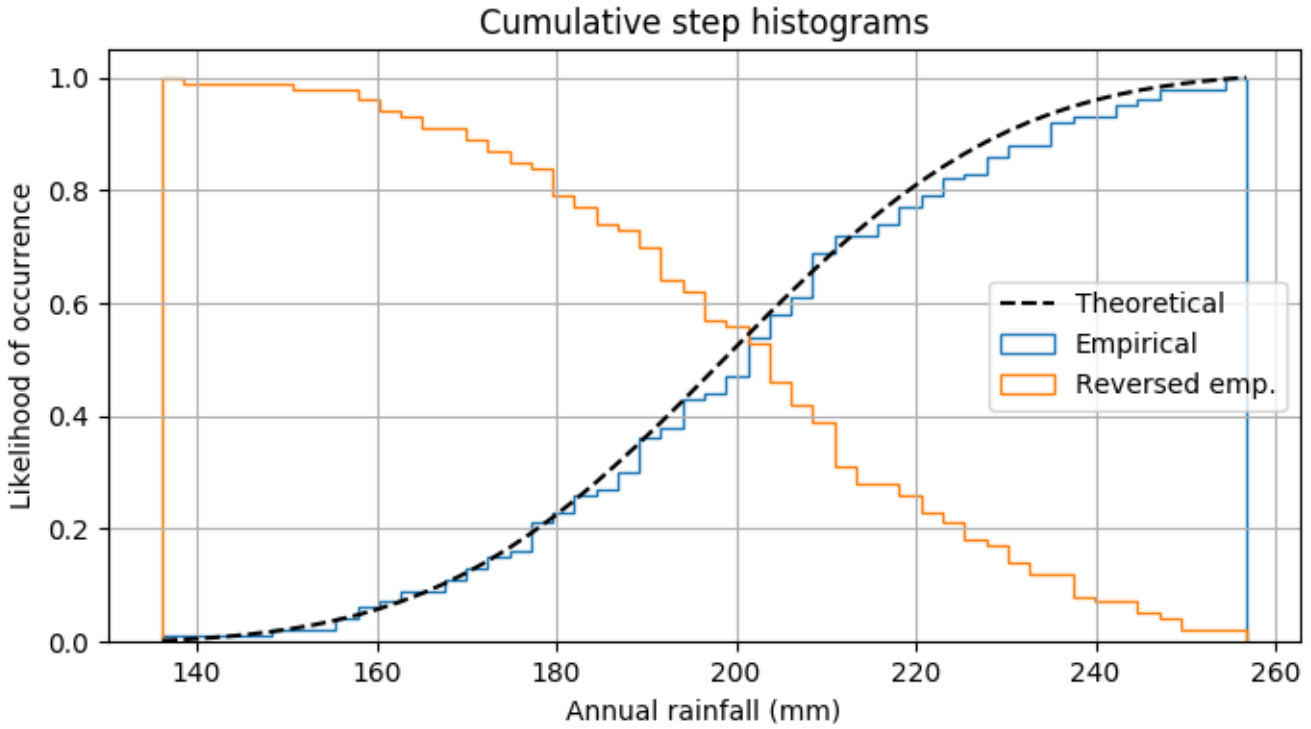
Bokeh

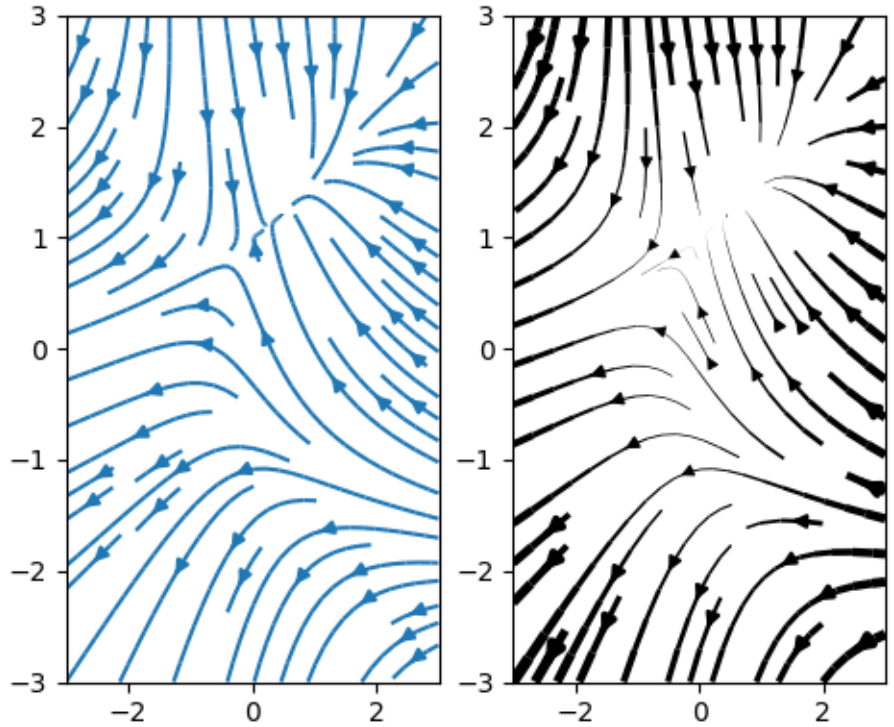


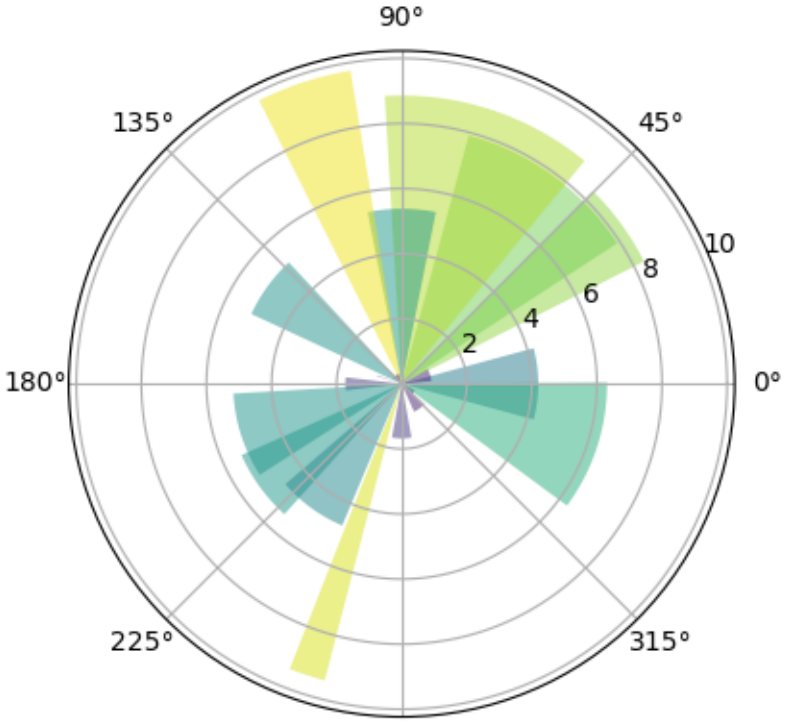
matplotlib.org

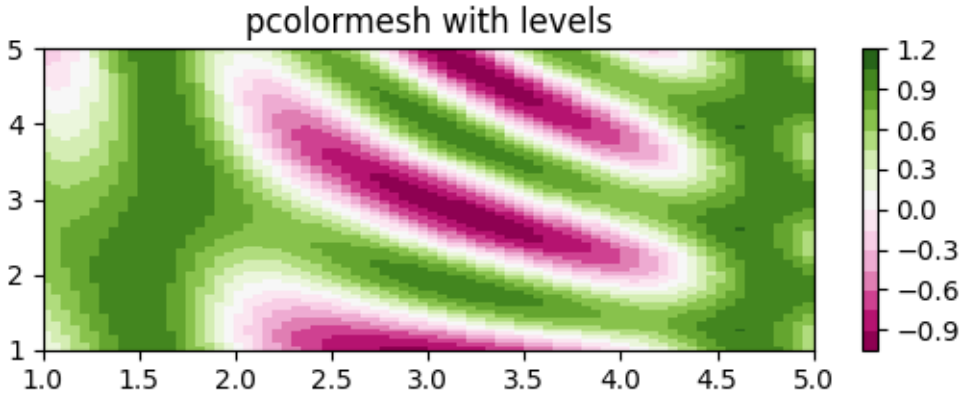

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
1
2
3
4
5
6

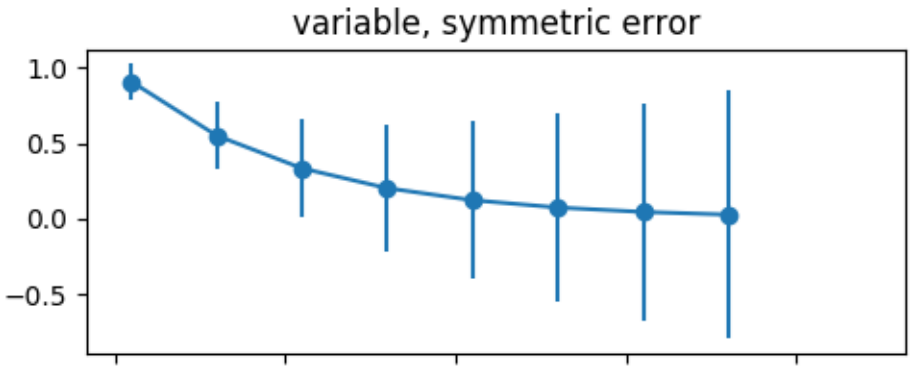

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
red
green
blue

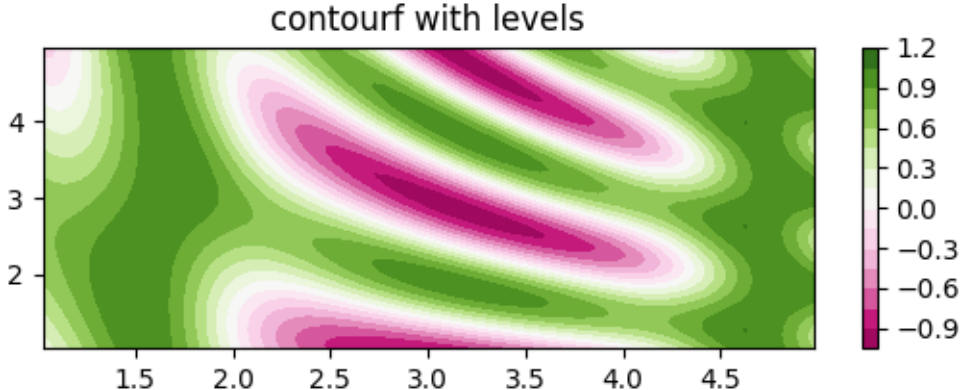

Cumulative step histograms
Likelihood of occurrence
1.0
0.8
0.6
0.4
0.2
0.0
140
160
180
200
220
240
260
Annual rainfall (mm)
Theoretical
Empirical
Reversed emp.

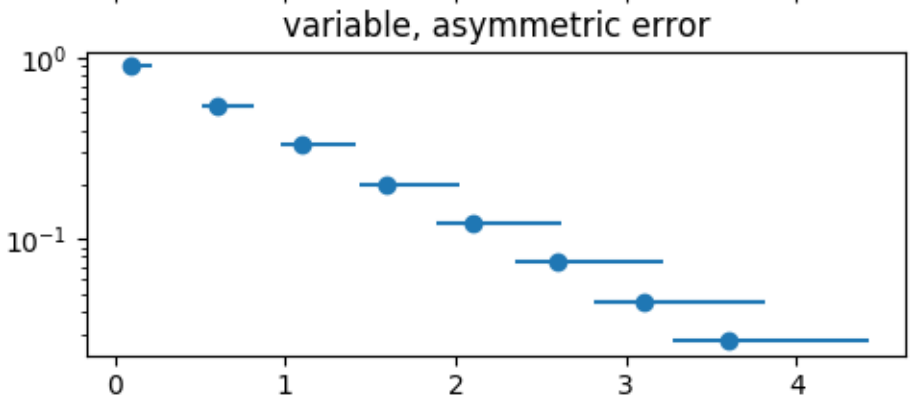



90°
135°
45°
180°
0°
225°
315°
2
4
6
8
10

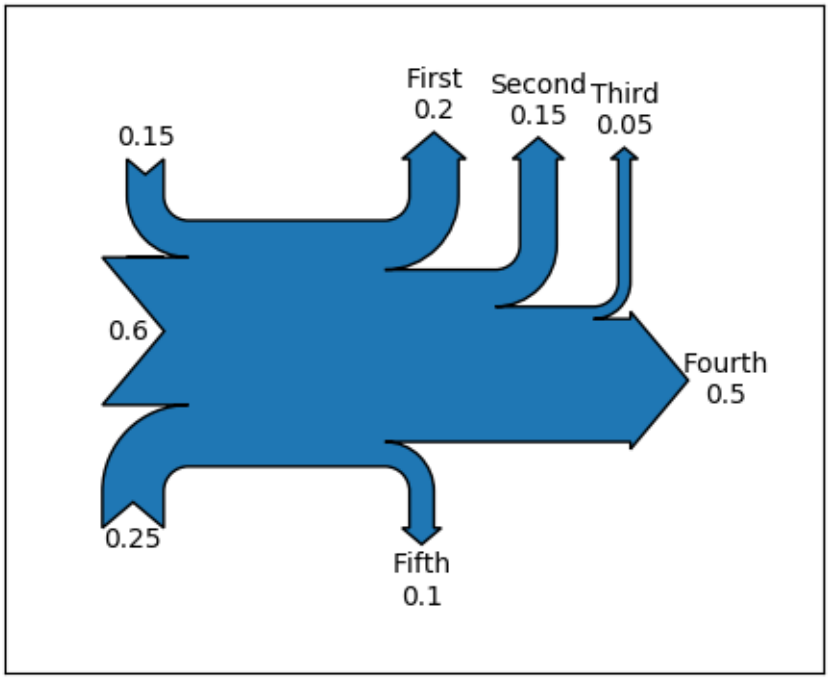

pcolormesh with levels

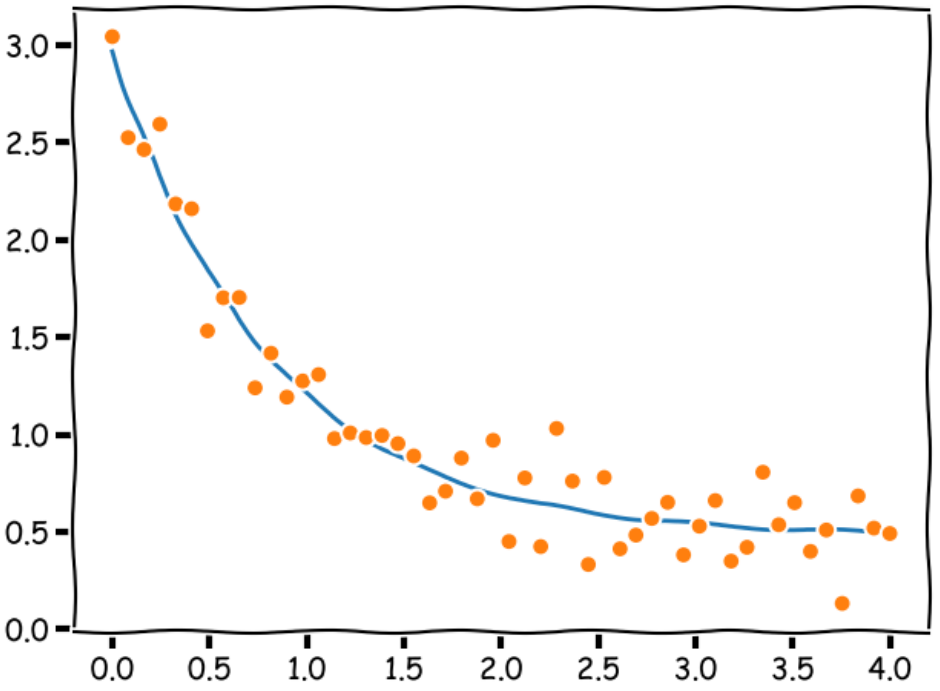

variable, symmetric error


contourf with levels

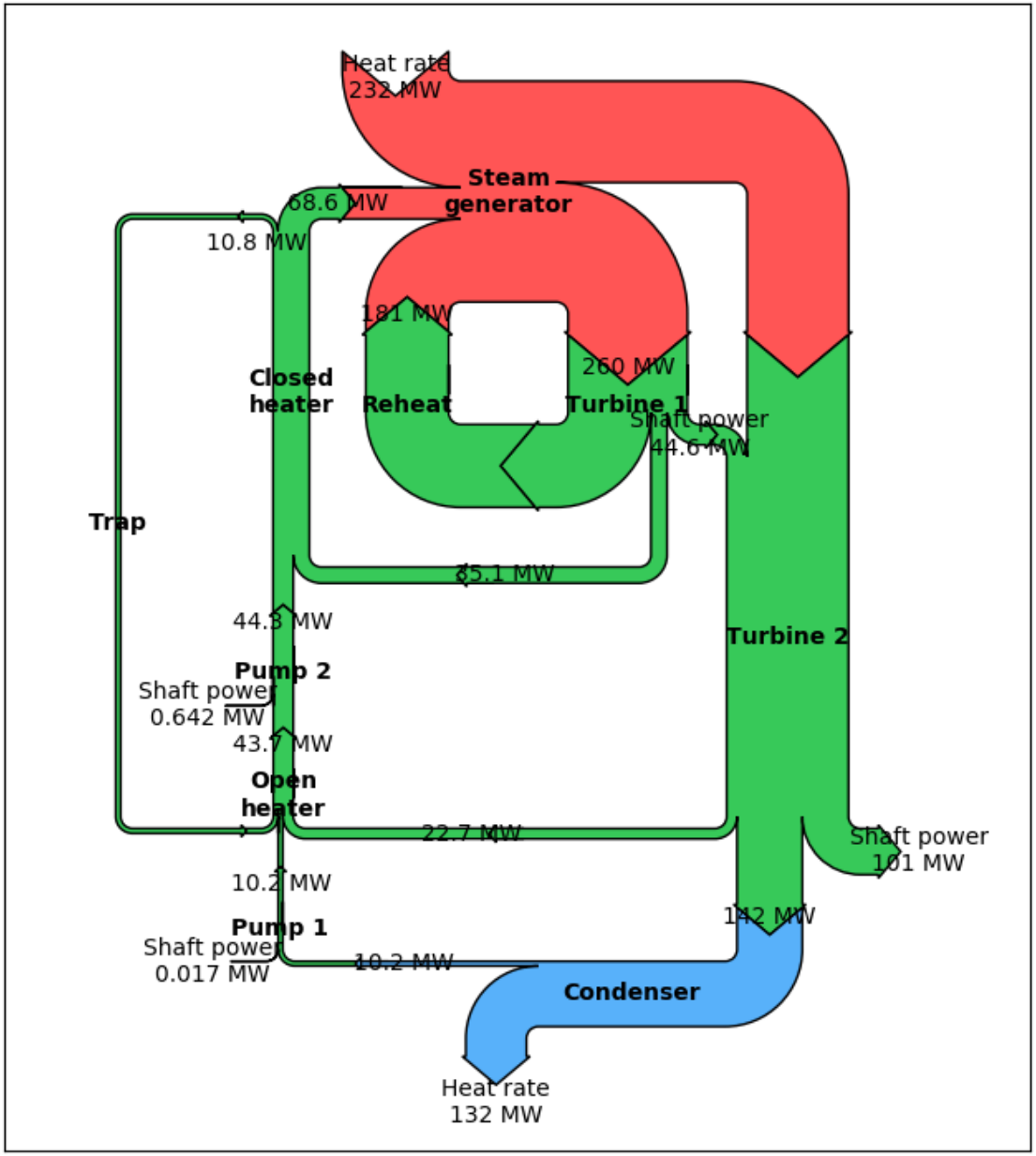

variable, asymmetric error

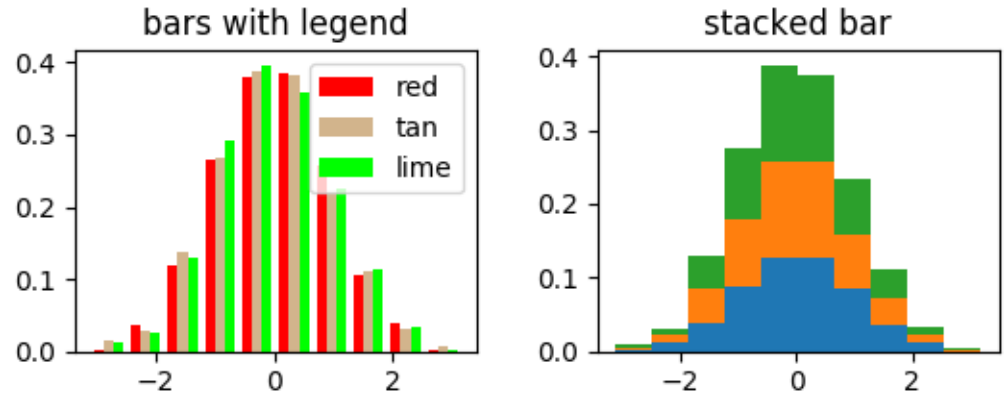
The default settings produce a diagram like this.



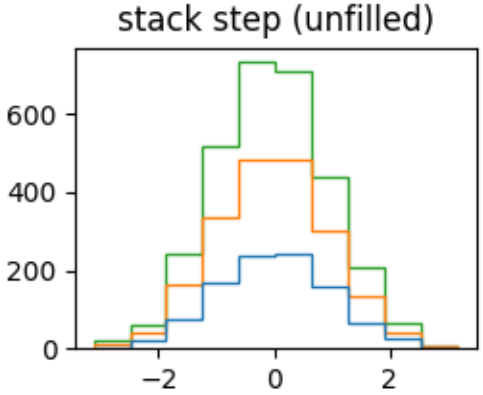



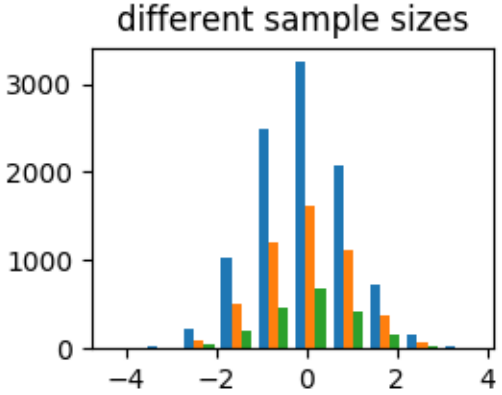

Rankine Power Cycle: Example 8.6 from Moran and Shapiro
"Fundamentals of Engineering Thermodynamics ", 6th ed., 2008

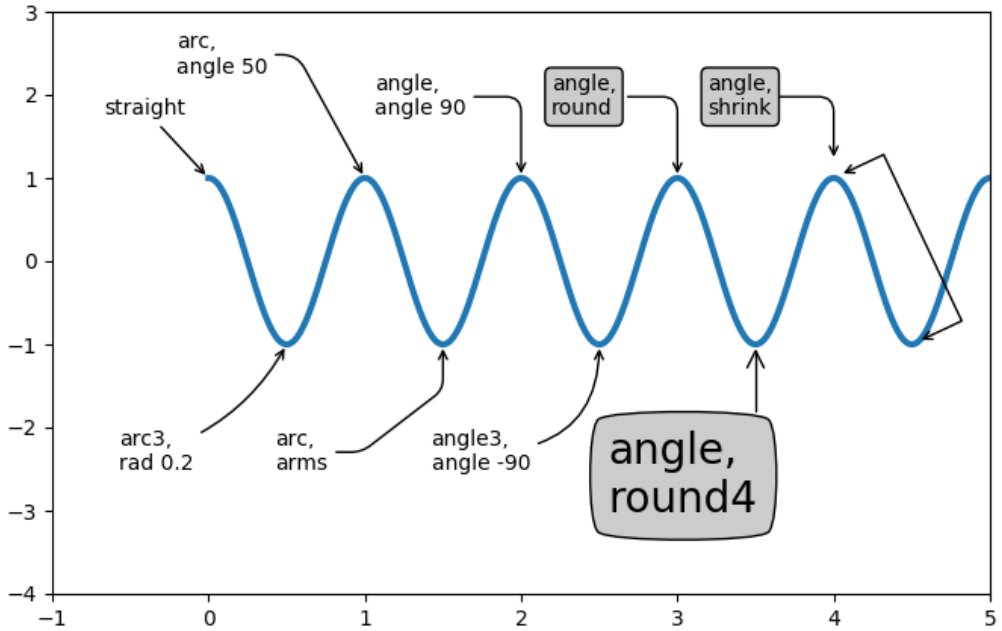


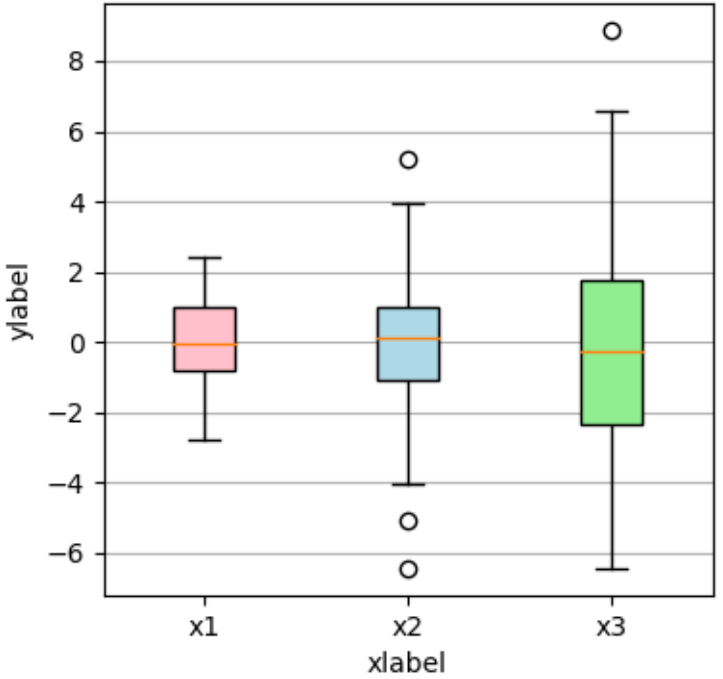

bars with legend
red
tan
lime

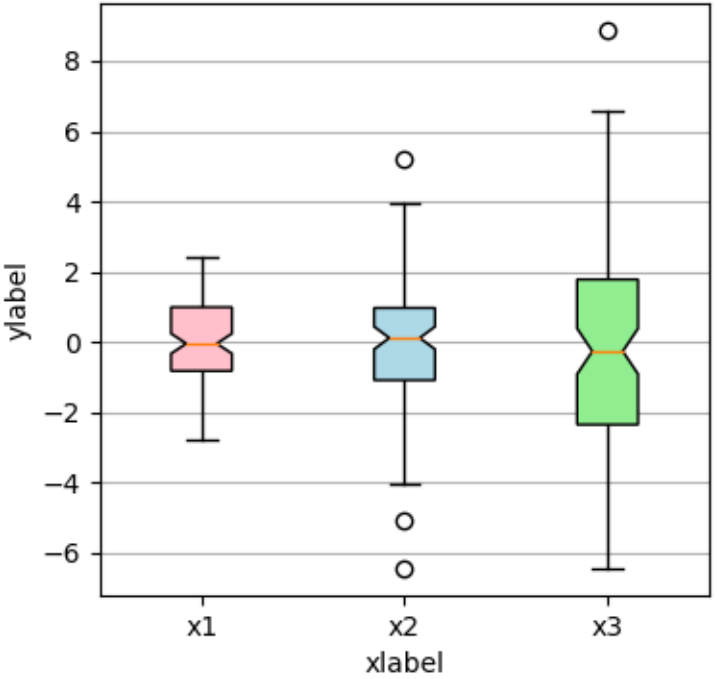

stacked bar

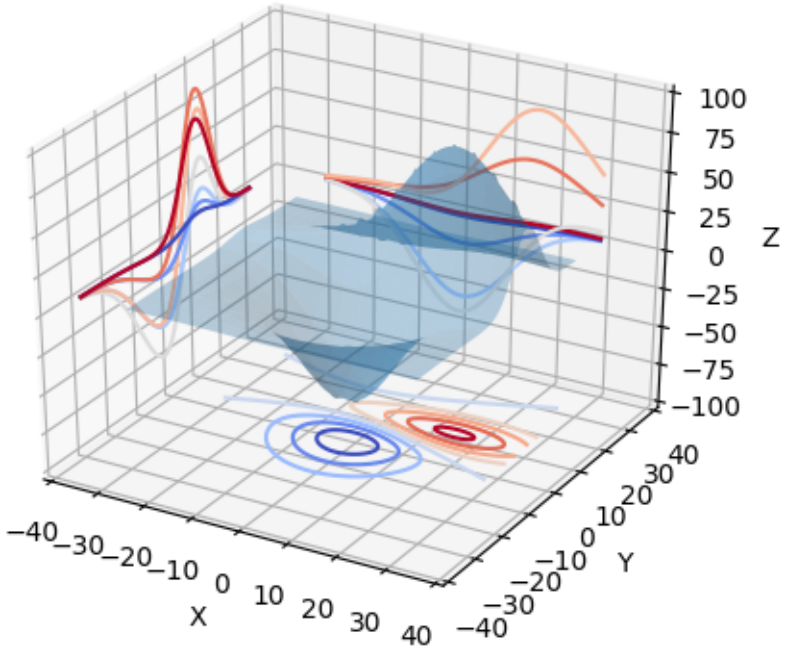

stack step (unfilled)

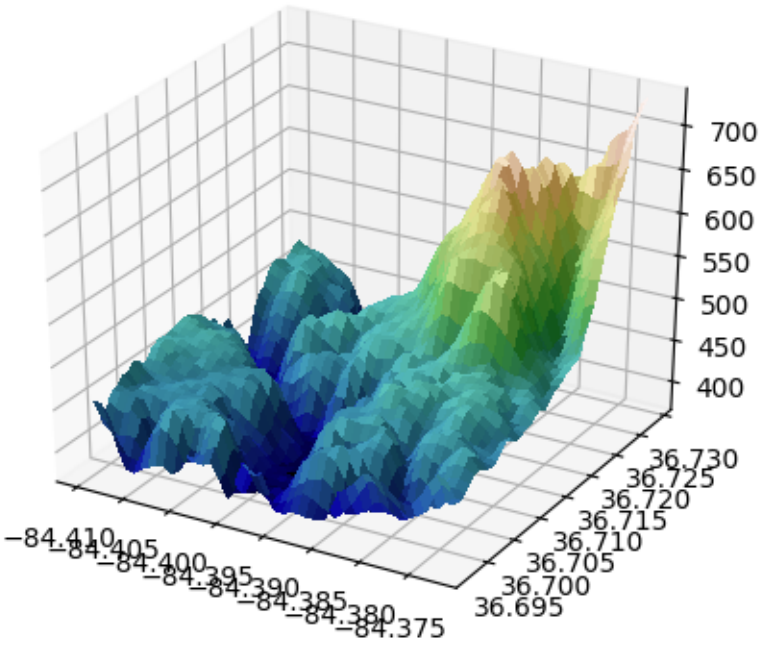

different sample sizes


straight
arc,
angle 50
angle,
angle 90
angle,
round
angle,
shrink
arc3,
rad 0.2
arc,
arms
angle3,
angle -90
angle,
round4

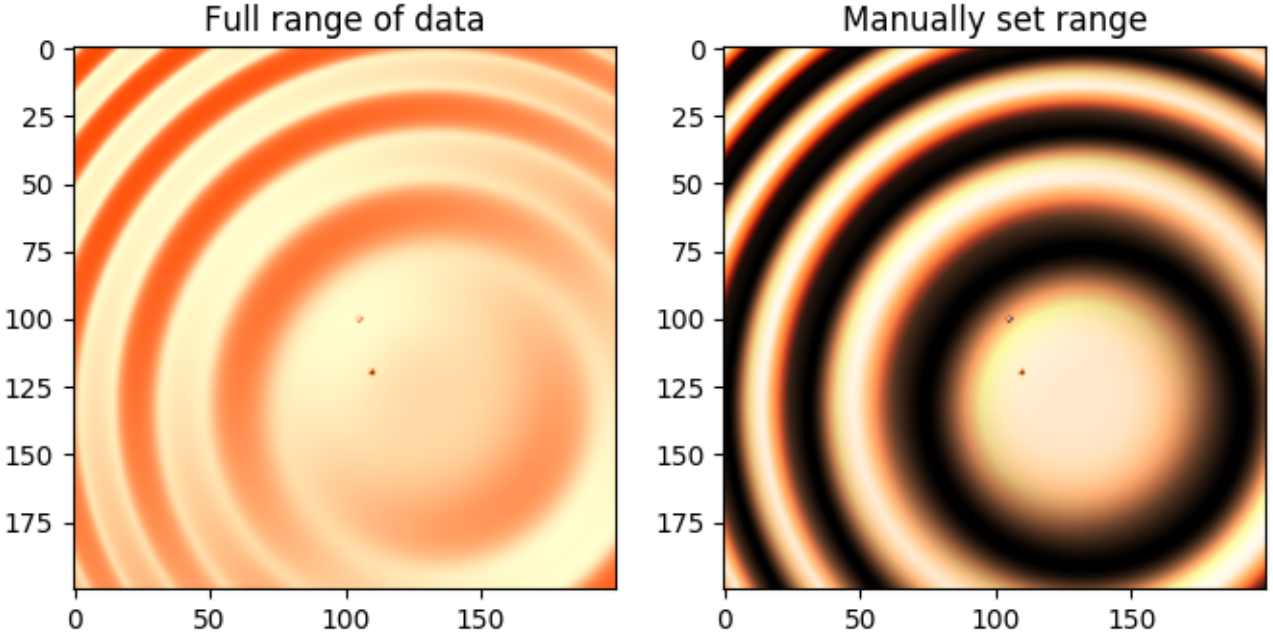

ylabel
x1
x2
x3
xlabel


ylabel
x1
x2
x3
xlabel

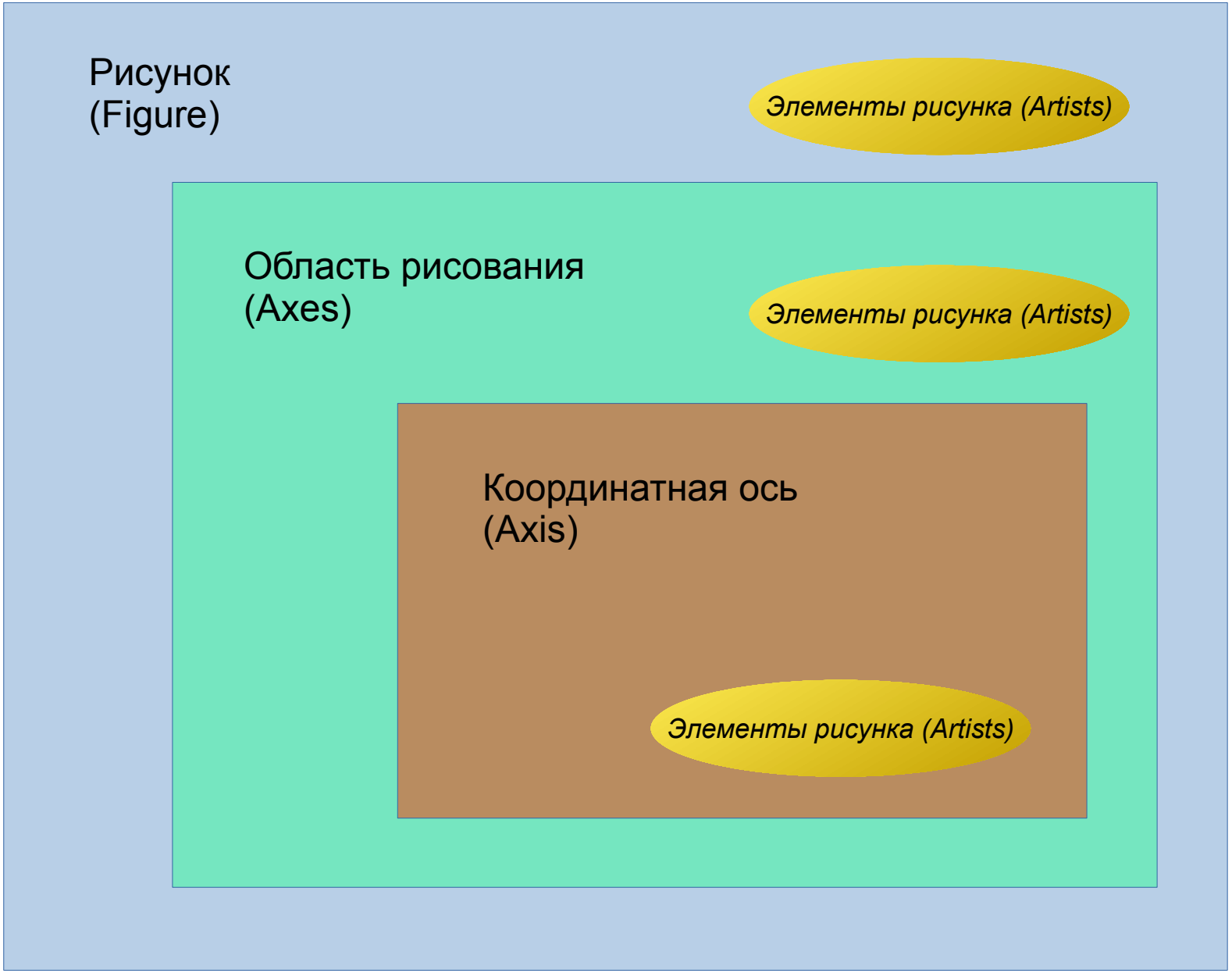



## Avoiding Outliers in Shaded Plots



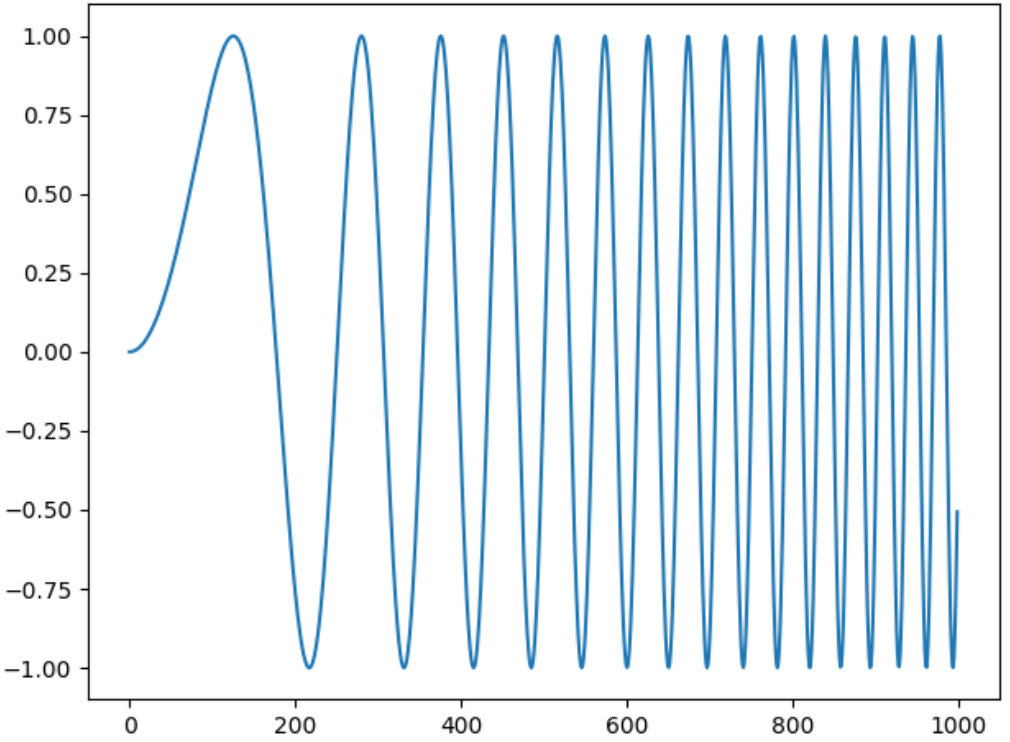
# *Иерархия рисунка*



**pyplot** – интерактивный интерфейс для построения графиков («машина состояний»)

# *pyplot.plot*

```
import matplotlib as mpl
import matplotlib.pyplot as plt

y=np.sin(np.linspace(0,10,1000)**2)
plt.plot(y)
plt.show()
```

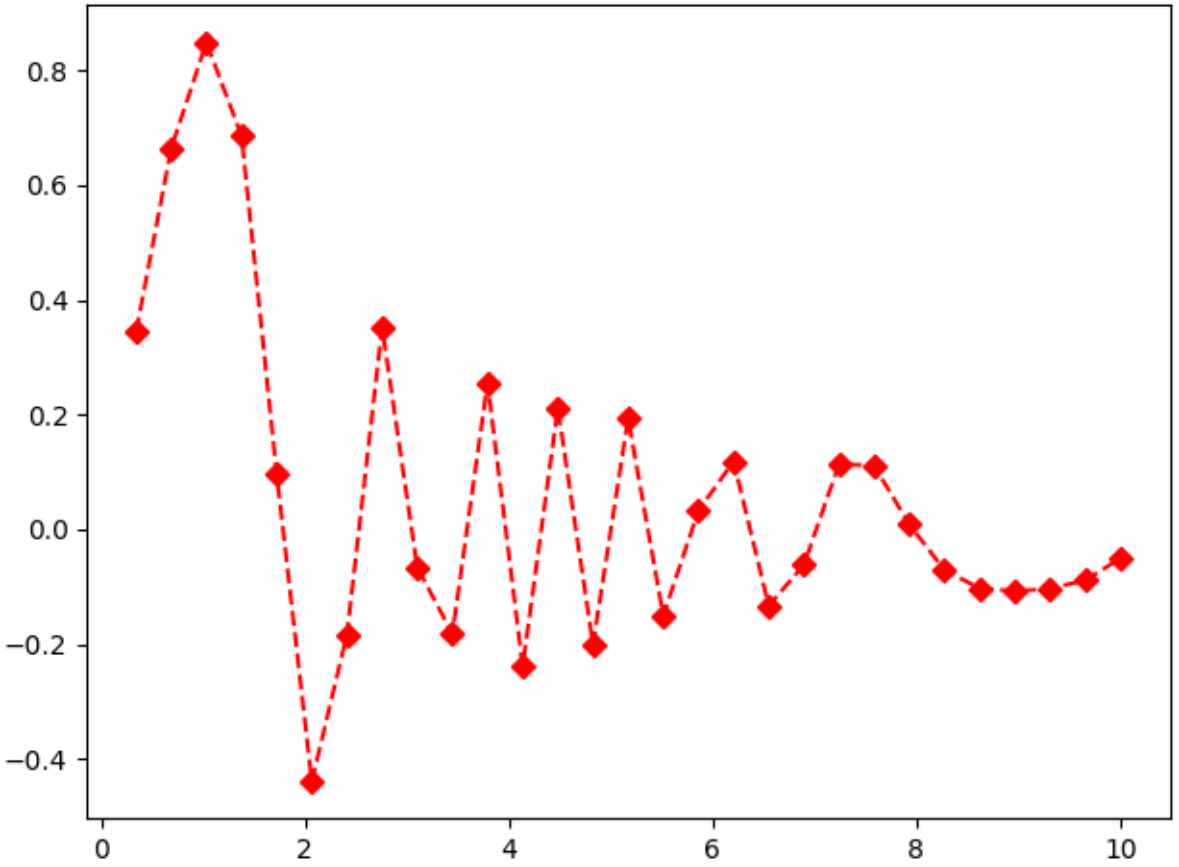



savefig(fname, dpi=None, facecolor='w', edgecolor='w', orientation='portrait', papertype=None, format=None, transparent=False, bbox_inches=None, pad_inches=0.1, frameon=None)

Форматы: eps, jpeg, pdf, png, ps, svg

# *pyplot.plot*

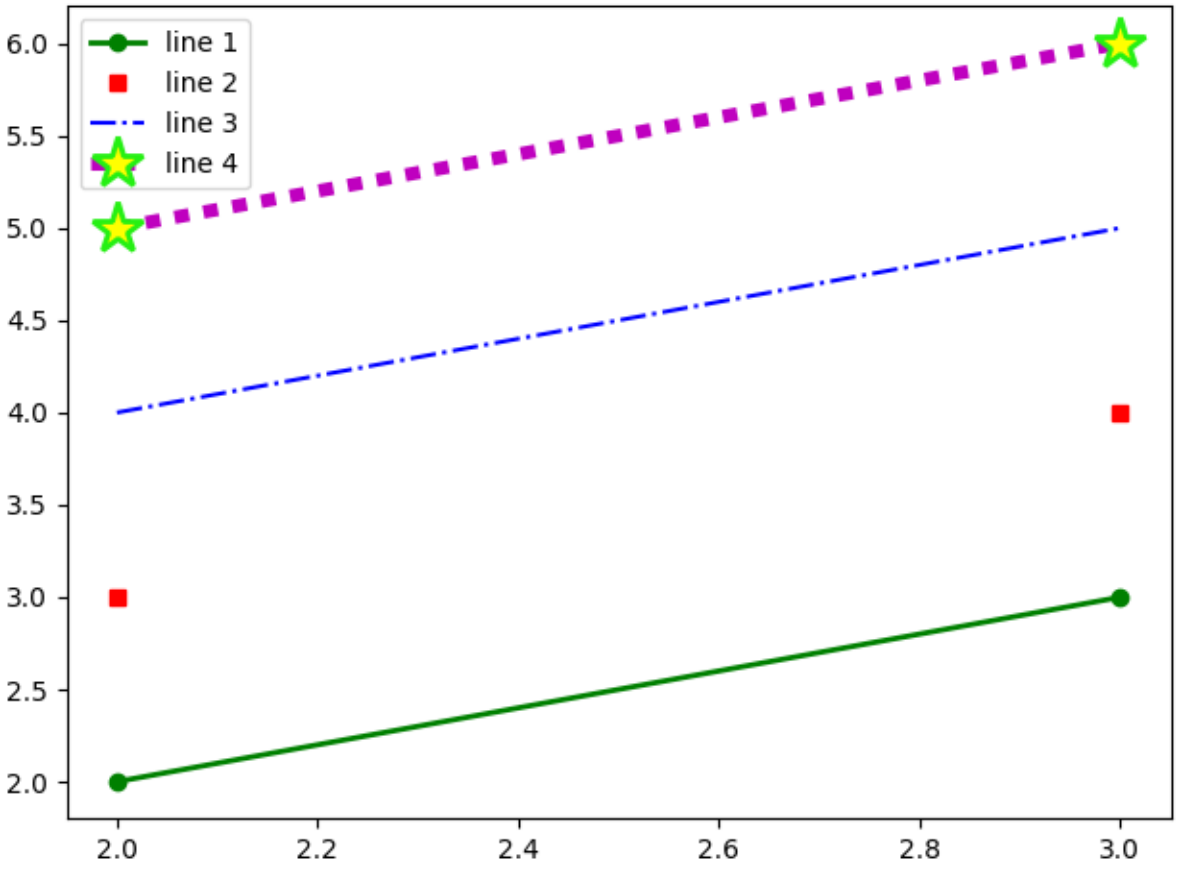
```
x=np.linspace(0,10,30)
y=np.sin(x**2)/x

plt.plot(x,y,'rD–')
```




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| agg_filter | alpha | animated | antialiased или aa | axes | clip_box | clip_on | clip_path | color или c | contains |
| dash_capstyle | dash_joinstyle | dashes | drawstyle | figure | fillstyle | gid | label | linestyle или ls | linewidth или lw |
| marker | markeredgecolor или mec | markeredgewidth или mew | markerfacecolor или mfc | markerfacecoloralt или mfcalt | markersize или ms | markevery | path_effects | picker | pickradius |
| rasterized | sketch_params | snap | solid_capstyle | solid_joinstyle | | transform | url | visible | xdata |
| ydata | zorder | | | | | | | | |

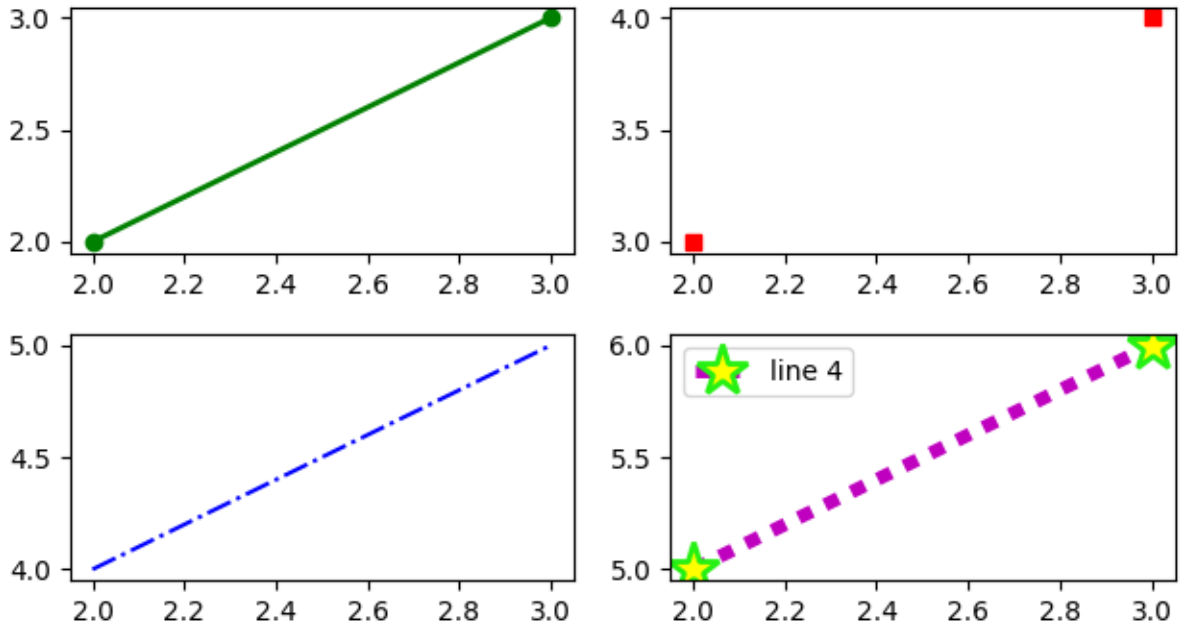
# *pyplot.plot*

```
plt.plot([2,3], [2,3], 'go-', label='line 1', linewidth=2)
plt.plot([2,3], [3,4], 'rs',  label='line 2')
plt.plot([2,3], [4,5], 'b-.', label='line 3')
plt.plot([2,3], [5,6], 'm*:', label='line 4', lw=5,mec='#25F013',
    mfc='#FFFF00',mew=2,ms=20)
plt.legend()
```



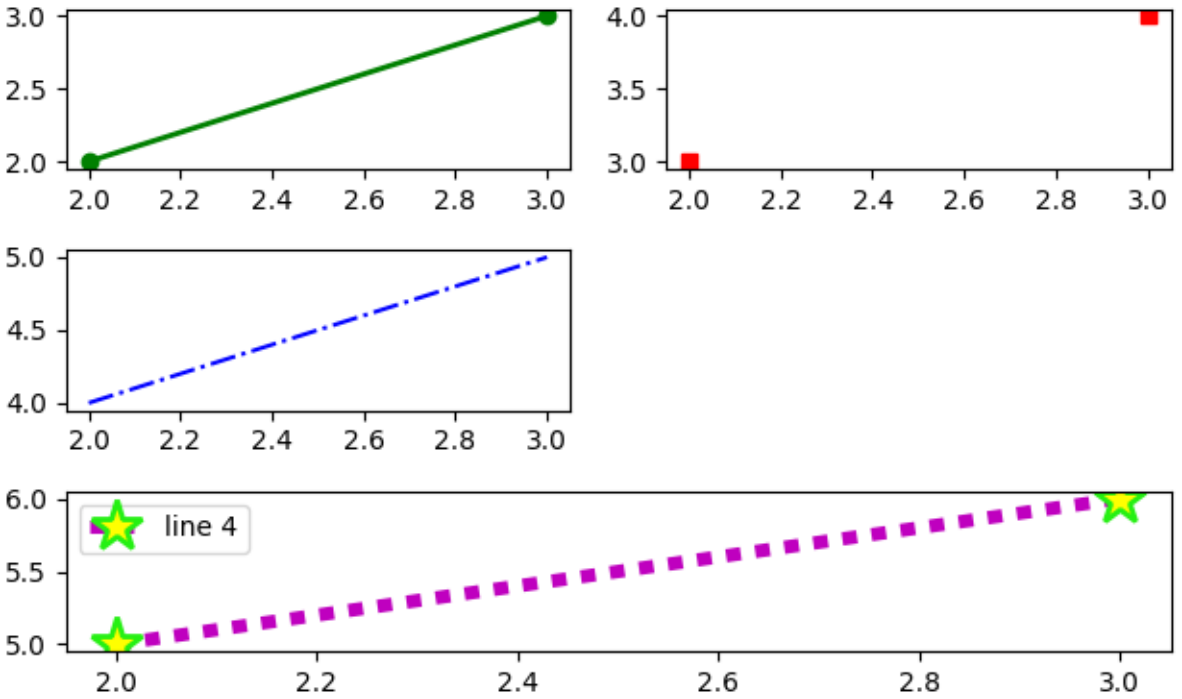
# pyplot.plot: позиционирование

```
plt.subplot(221)
plt.plot([2,3], [2,3], 'go-', label='line 1', linewidth=2)

plt.subplot(222)
plt.plot([2,3], [3,4], 'rs',  label='line 2')

plt.subplot(223)
plt.plot([2,3], [4,5], 'b-.', label='line 3')

plt.subplot(224)
plt.plot([2,3], [5,6], 'm*:', label='line 4', lw=5,mec='#25F013',mfc='#FFFF00',mew=2,ms=20)

plt.legend()
```



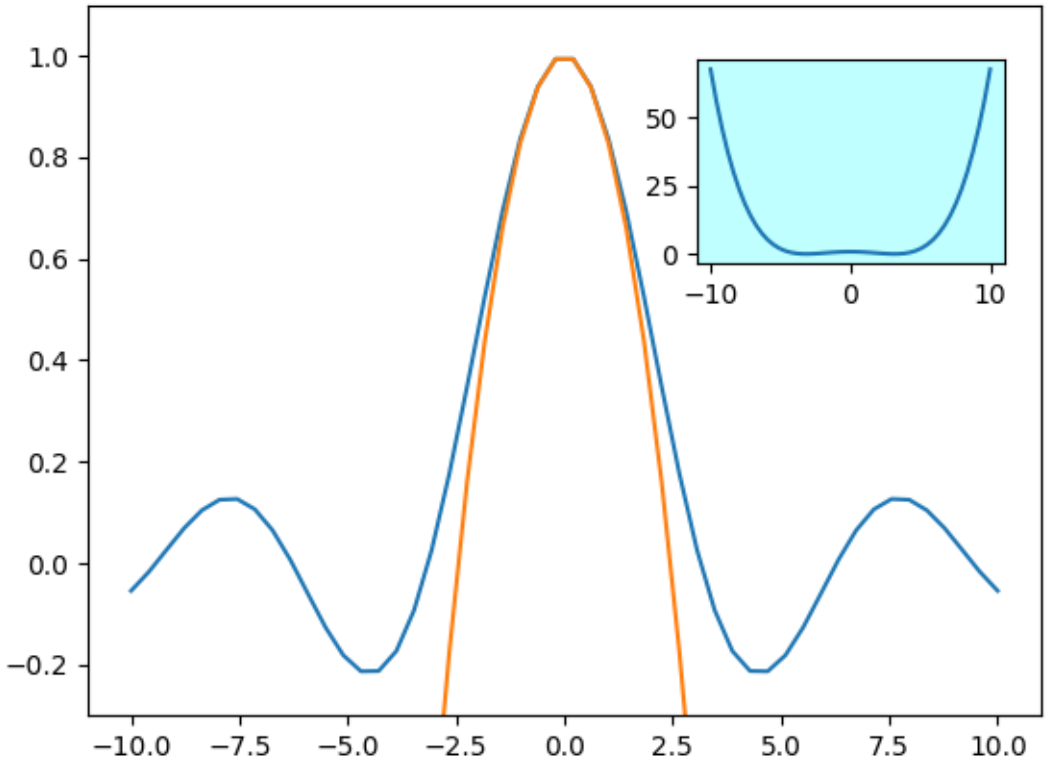
# pyplot.plot: позиционирование

```
plt.subplot2grid((3,2),(0,0))
plt.plot([2,3], [2,3], 'go-', label='line 1', linewidth=2)

plt.subplot2grid((3,2),(0,1))
plt.plot([2,3], [3,4], 'rs',  label='line 2')

plt.subplot2grid((3,2),(1,0))
plt.plot([2,3], [4,5], 'b-.', label='line 3')

plt.subplot2grid((3,2),(2,0),colspan=2)
plt.plot([2,3], [5,6], 'm*:', label='line 4', lw=5,mec='#25F013',mfc='#FFFF00',mew=2,ms=20)

plt.legend()
```

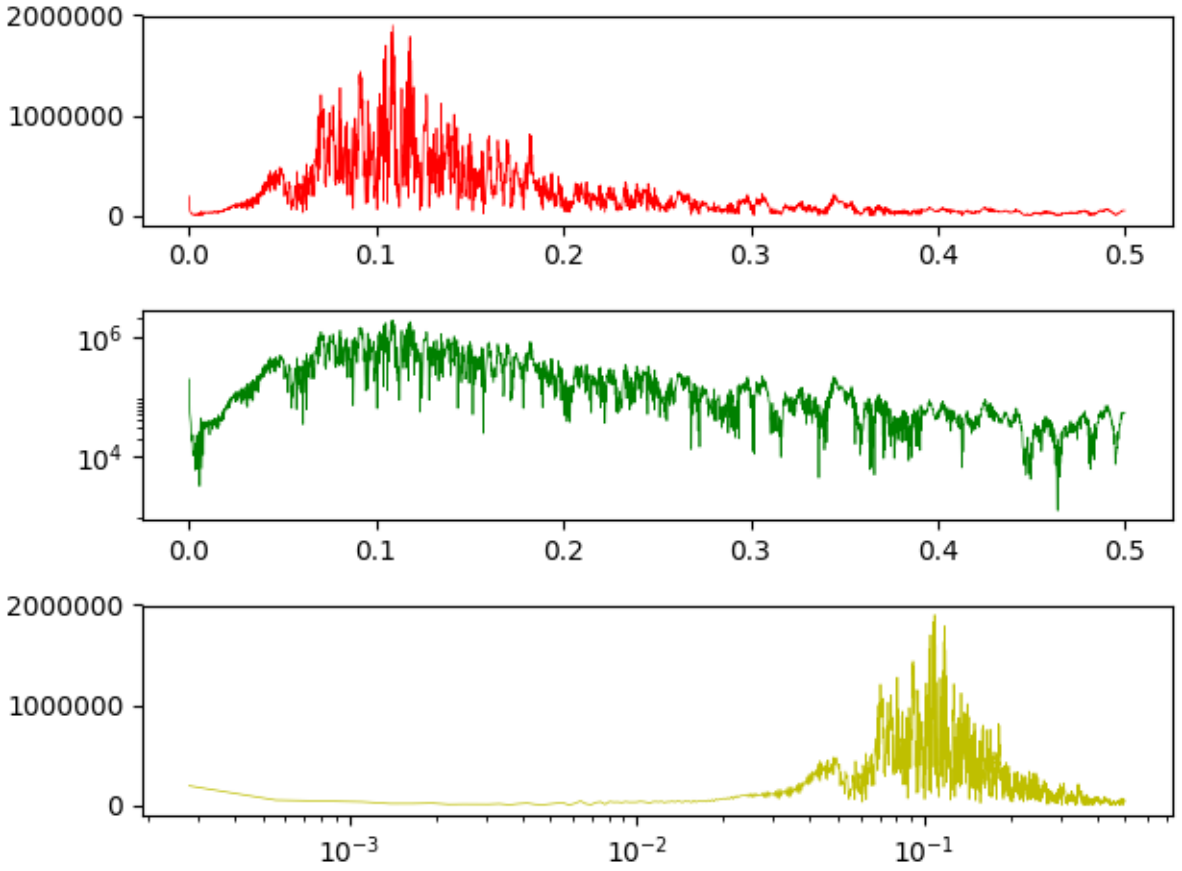


# *pyplot.plot: позиционирование*

```
x=np.linspace(-10,10,50)
plt.plot(x,np.sin(x)/x)
plt.plot(x,1-x**2/6)
plt.ylim(-0.3,1.1)
sp=plt.axes([0.62,0.6,0.25,0.22],axisbg='#BFFFFF')
plt.plot(x,1-x**2/6+x**4/120)
```



# *Логарифмические шкалы*

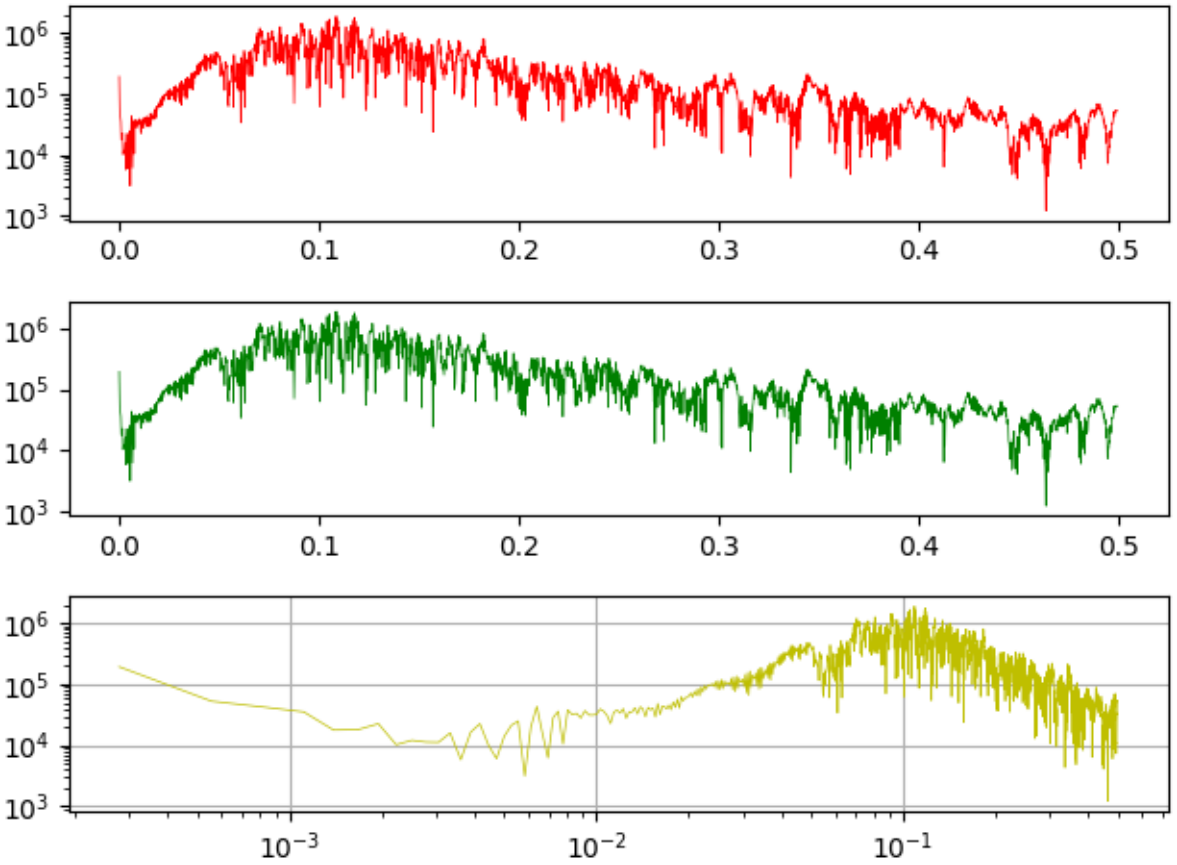


# *Логарифмические шкалы*

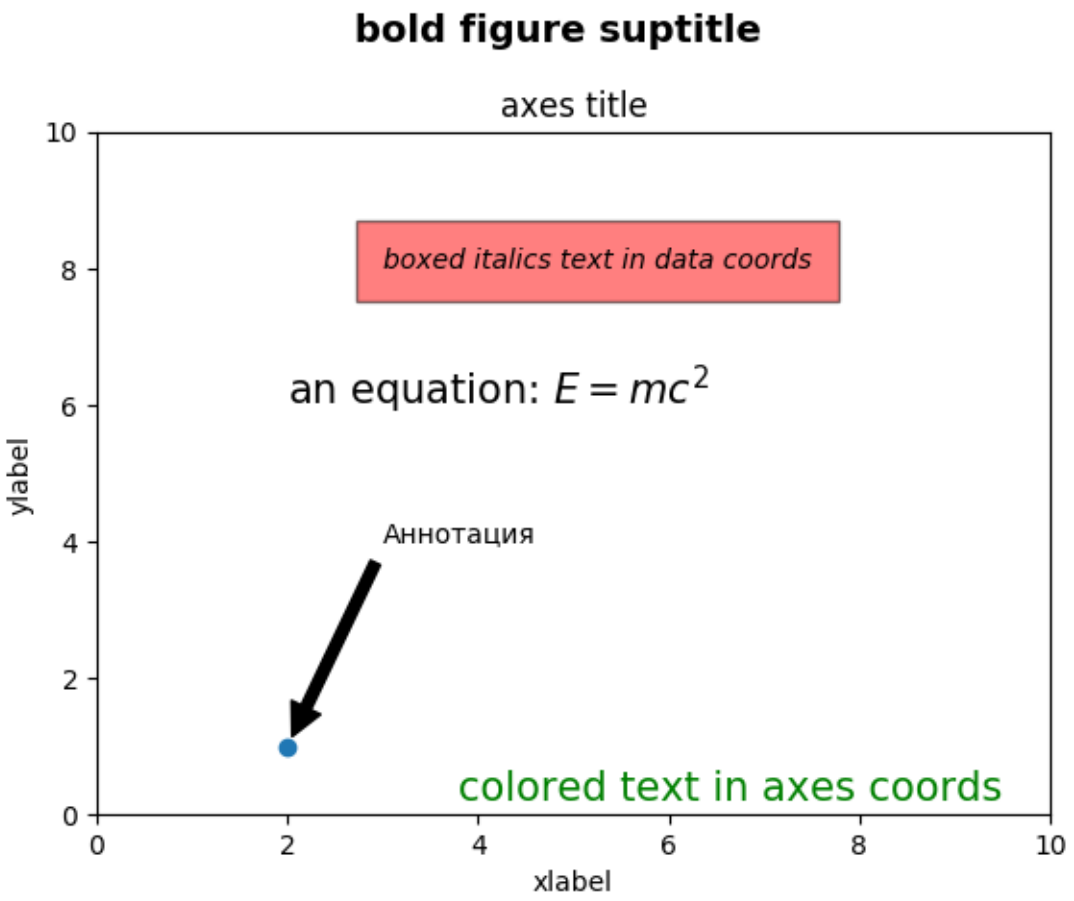
```
import scipy.fftpack as fp
sig = scipy.loadtxt('OYP_KMA03_2011-03-11T053500_1.0Hz.dat')
sample_freq = fp.fftfreq(sig.size)
sig_fft = fp.fft(sig)
pidxs = np.where(sample_freq > 0)
freqs = sample_freq[pidxs]
power = np.abs(sig_fft)[pidxs]

plt.subplot(311)
plt.plot(freqs,power,'r',lw=0.5)

plt.subplot(312)
plt.semilogy(freqs,power, 'g',lw=0.5)

plt.subplot(313)
plt.semilogx(freqs,power,'y',lw=0.5)
```

# *Логарифмические шкалы*

```
plt.subplot(311)
plt.plot(freqs,power, 'r',lw=0.5)
plt.yscale('log')

plt.subplot(312)
plt.plot(freqs,power, 'g',lw=0.5)
plt.yscale('symlog')

plt.subplot(313)
plt.loglog(freqs,power,'y',lw=0.5)
plt.grid(True)
```



# Текст, аннотации

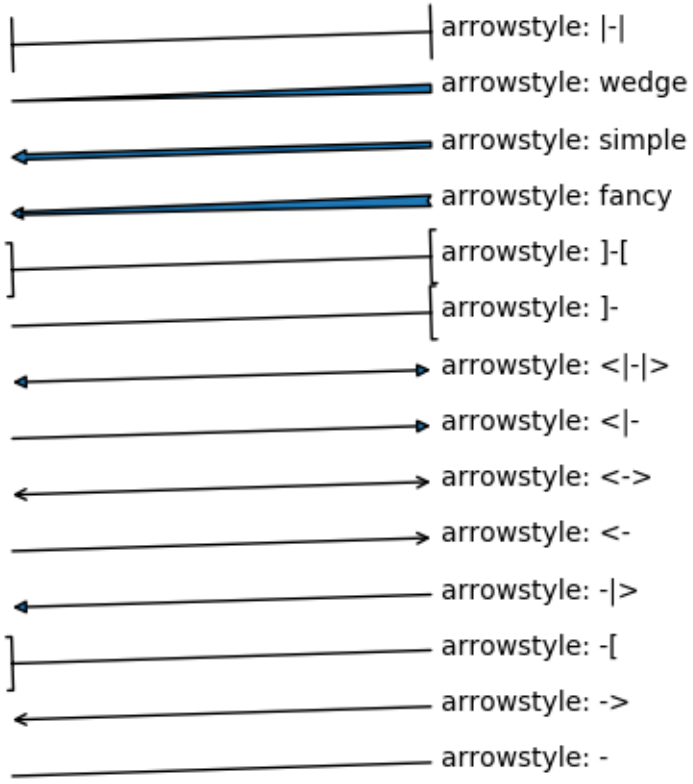


# Текст, аннотации

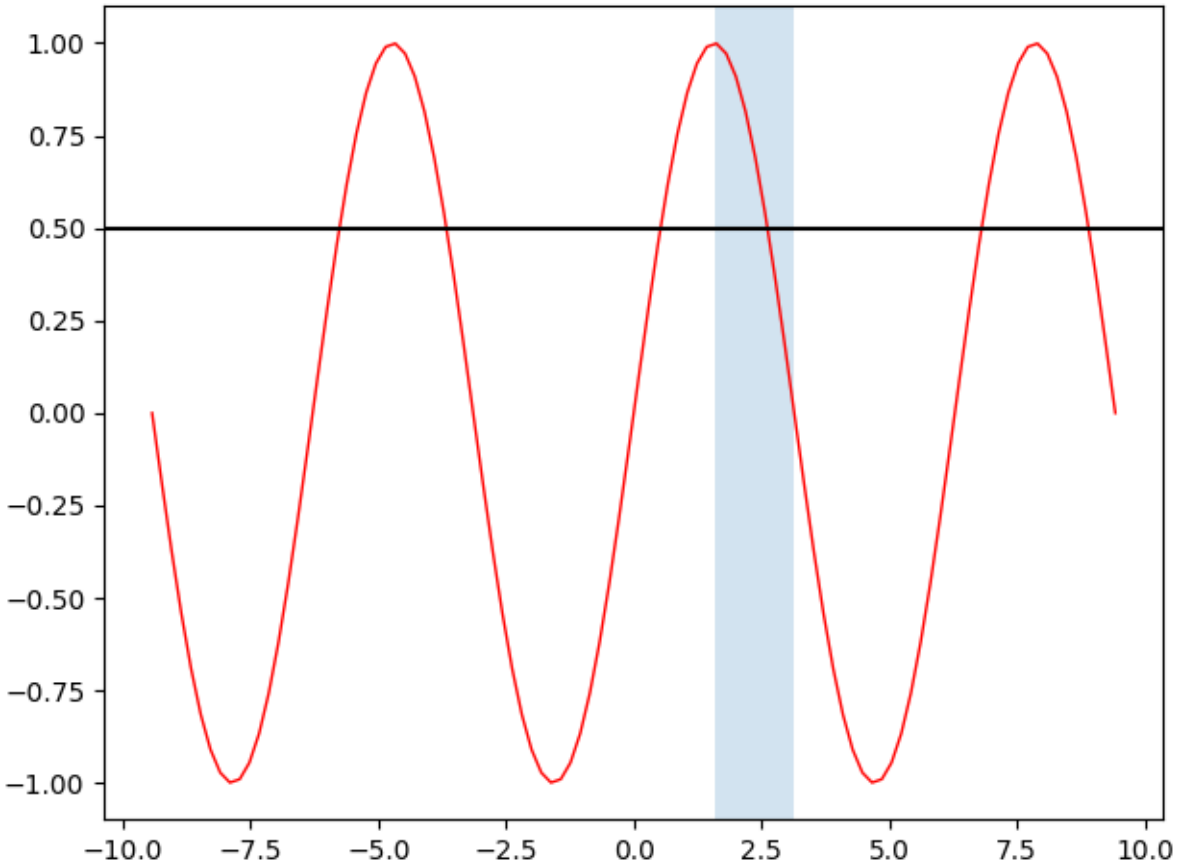
```
fig = plt.figure()
fig.suptitle('bold figure suptitle', fontsize=14, fontweight='bold')

ax = fig.add_subplot(111)
fig.subplots_adjust(top=0.85)
ax.set_title('axes title')

ax.set_xlabel('xlabel')
ax.set_ylabel('ylabel')

ax.text(3, 8, 'boxed italics text in data coords', style='italic',
        bbox={'facecolor':'red', 'alpha':0.5, 'pad':10})

ax.text(2, 6, r'an equation: $E=mc^2$', fontsize=15)

ax.text(0.95, 0.01, 'colored text in axes coords',
        verticalalignment='bottom', horizontalalignment='right',
        transform=ax.transAxes,
        color='green', fontsize=15)

ax.plot([2], [1], 'o')
ax.annotate('Аннотация', xy=(2, 1), xytext=(3, 4),
            arrowprops=dict(facecolor='black', shrink=0.05))

ax.axis([0, 10, 0, 10])
```

# Стили стрелок

```
arrowstyles = ['-', '->', '-[', '-|>',
               '<-', '<->', '<|-', '<|-|>',
               ']-', ']-[', 'fancy', 'simple',
               'wedge', '|-|']


x = 2
y = 0

dx = 10
dy = 5

for arrowstyle in arrowstyles:

    arrowprops = {'arrowstyle': arrowstyle,}
    y += 31
    plt.annotate (u'arrowstyle: ' + arrowstyle,
                  xy=(x, y),
                  xytext = (x + dx, y + dy),
                arrowprops = arrowprops)

plt.axis('off')
plt.xlim([0,25])
plt.ylim ([0, len(arrowstyles)*32])
```

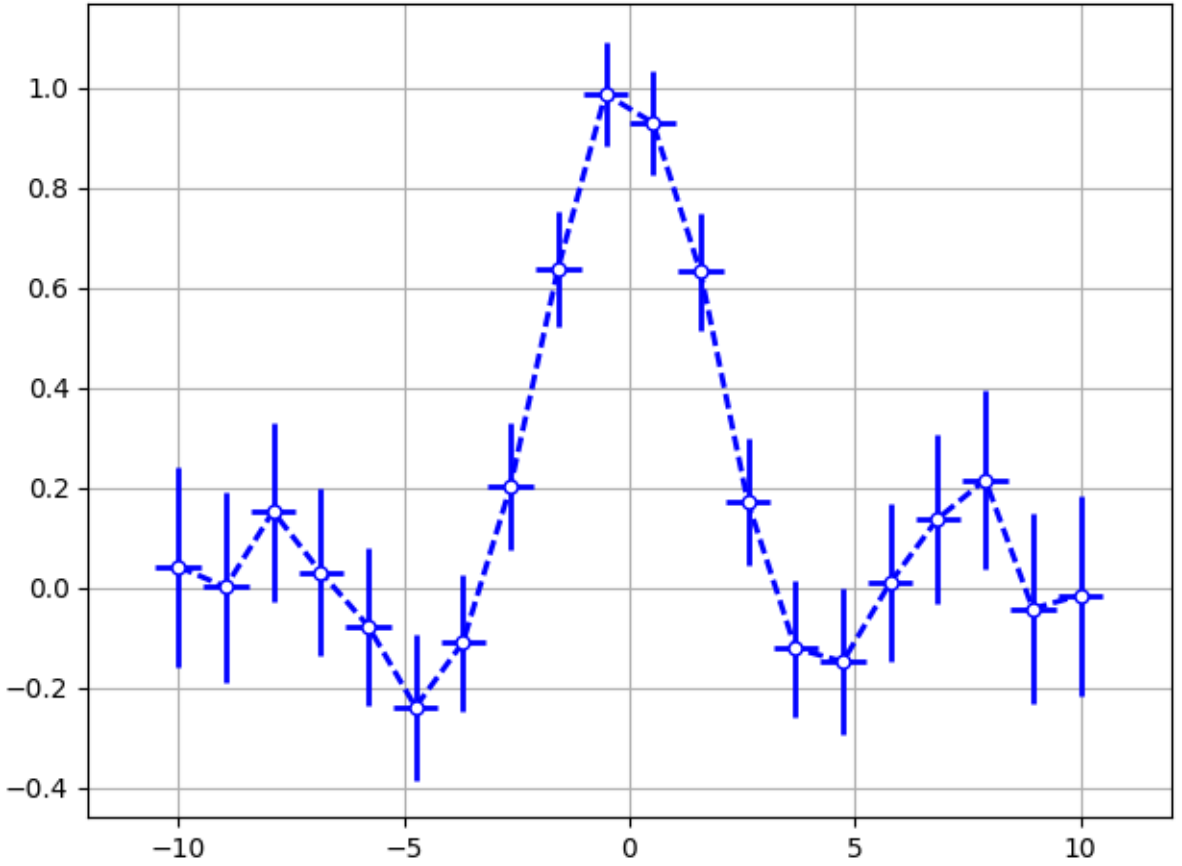


# *Диапазоны и линии*



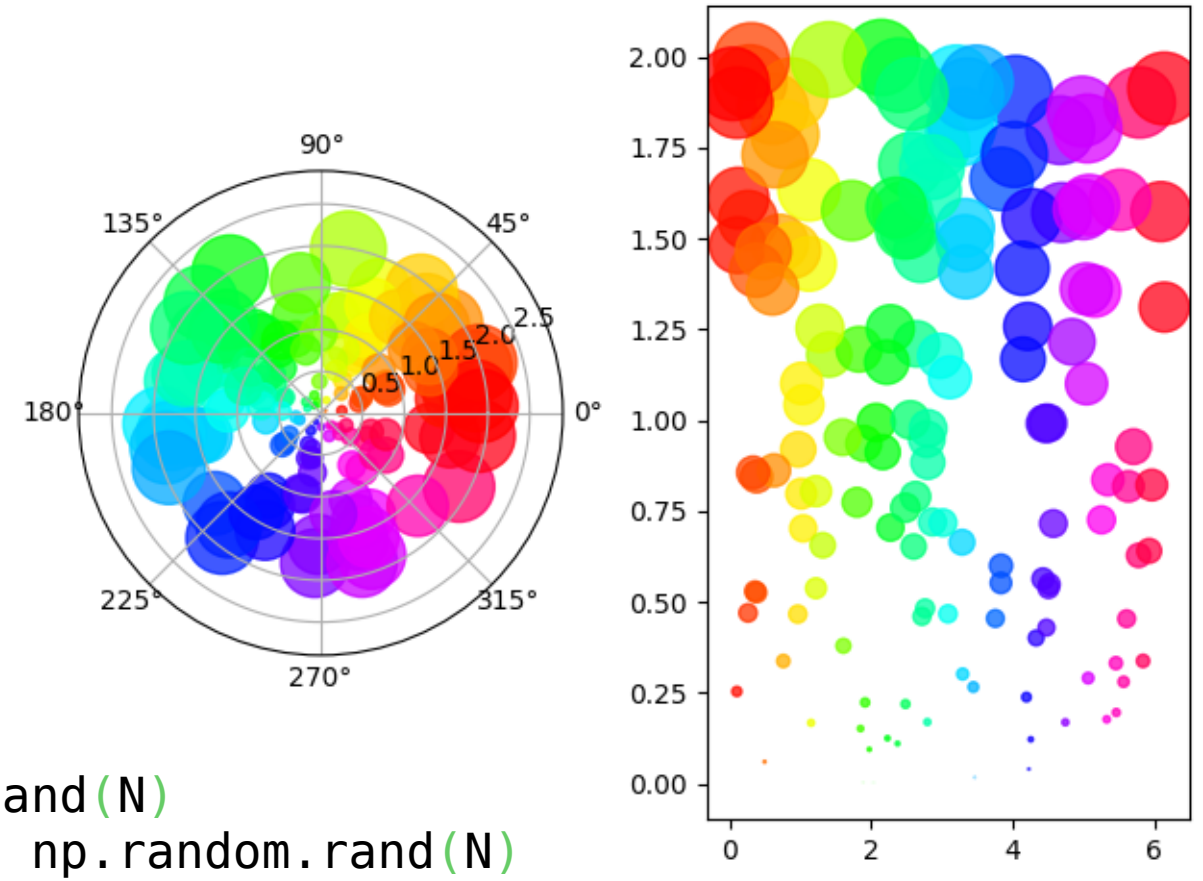

```
x = np.linspace(-3*np.pi,3*np.pi,100,endpoint= True)
y = np.sin(x)

plt.plot(x,y,'r',lw=1)
plt.axhline(0.5, color='black')
plt.axvspan(np.pi,np.pi/2,alpha=0.2)
```

# Ошибки, погрешности



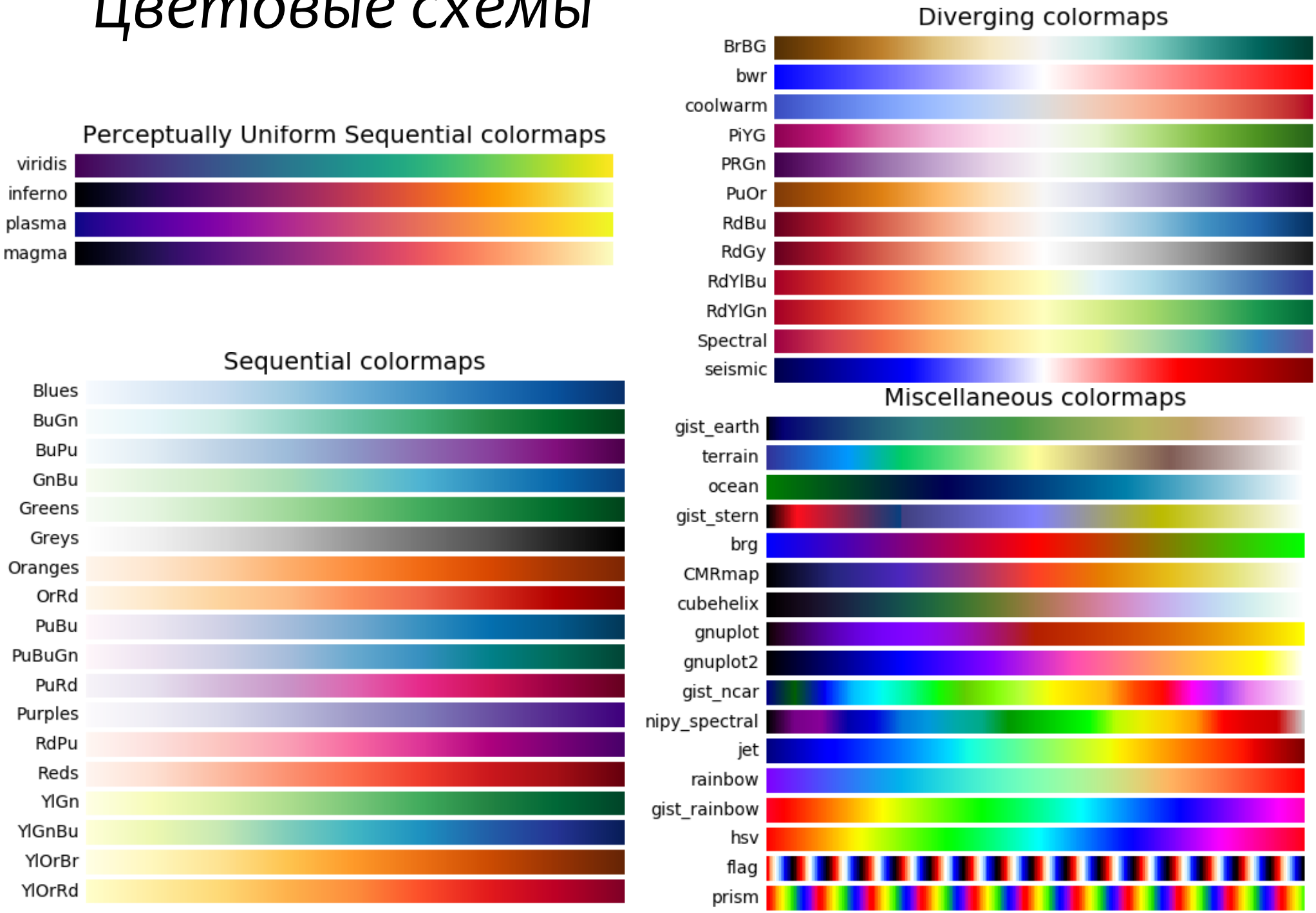

```
x=np.linspace(-10,10,20)
y=np.sin(x)/x + 0.2*np.random.rand(20)-0.1
dx = 0.5
dy = 0.1+np.abs(x)/100
plt.errorbar(x, y, xerr=dx, yerr=dy, c='b', ls='--',\
    lw=2, marker='o', mfc='w', ms=5)
plt.xlim(-12,12)
plt.grid()
```

# *Полярная диаграмма / “scatter”*



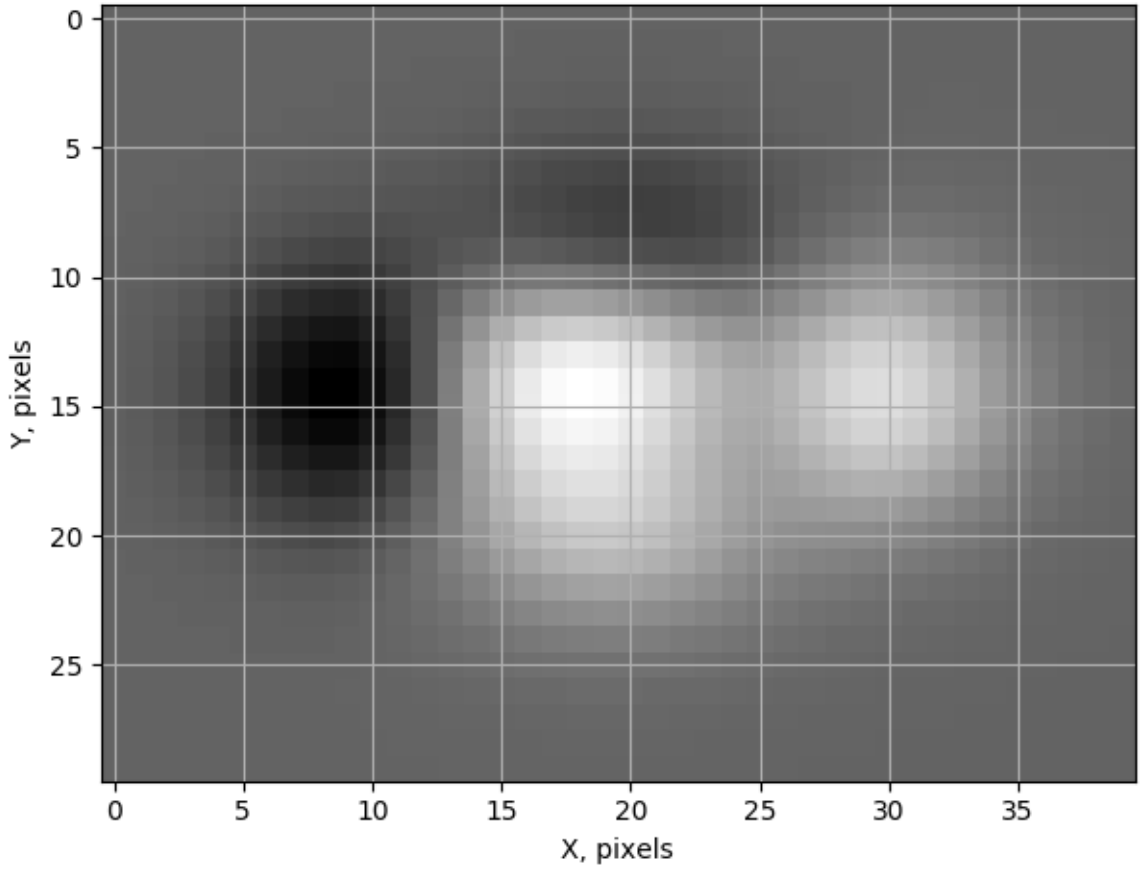

```
N = 150
r = 2 * np.random.rand(N)
theta = 2 * np.pi * np.random.rand(N)
area = 200 * r**2
colors = theta

plt.subplot(121, projection='polar')
plt.scatter(theta, r, c=colors, s=area, cmap='hsv', alpha=0.75)

plt.subplot(122)
plt.scatter(theta, r, c=colors, s=area, cmap='hsv', alpha=0.75)
```

# Цветовые схемы



# *pyplot.imshow*

```
def f(x, y):
    return (1-x/2+x**5+y**3) * np.exp(-x**2-y**2)

x = np.linspace(-3,3,40)
y = np.linspace(-3,3,30)
x, y = np.meshgrid(x,y)
array = f(x,y)
plt.figure(1)
plt.imshow(array, cmap='gray')
plt.xlabel("X, pixels")
plt.ylabel("Y, pixels")
plt.grid()
```

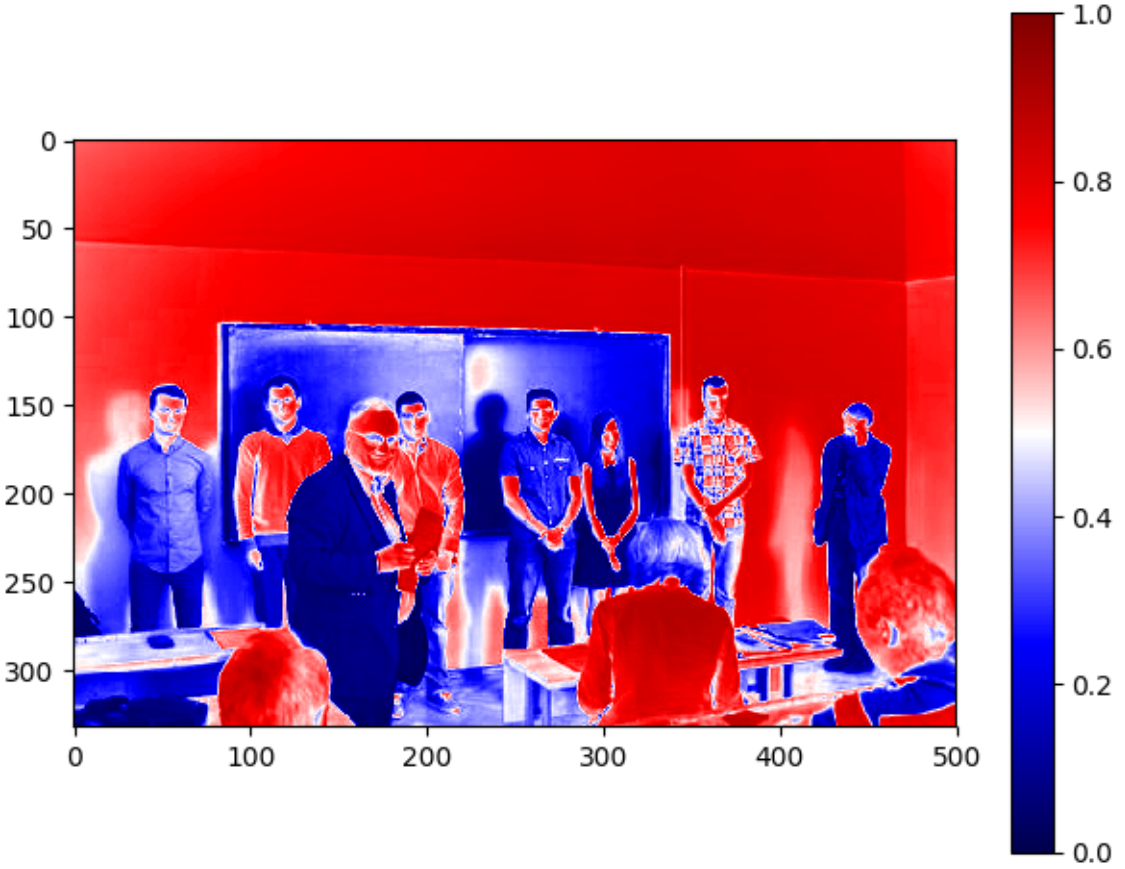


# *matplotlib.image*

```
>>> import matplotlib.image as mpimg

>>> mpimg.imread('DSC02766.JPG')
array([[[166, 168, 163],
        [166, 168, 163],
        [166, 167, 161],
        ...,
        [180, 185, 179],
        [179, 184, 178],
        [179, 184, 178]],
         ...,
        [193, 198, 202],
        [191, 198, 204],
        [190, 197, 203]]], dtype=uint8)

>>> mpimg.imread('DSC02766.JPG')/255
array([[[ 0.65098039,  0.65882353,  0.63921569],
        [ 0.65098039,  0.65882353,  0.63921569],
        [ 0.65098039,  0.65490196,  0.63137255],
        ...,
        [ 0.70588235,  0.7254902 ,  0.70196078],
        [ 0.70196078,  0.72156863,  0.69803922],
        [ 0.70196078,  0.72156863,  0.69803922]],
        ...,
        [ 0.75686275,  0.77647059,  0.79215686],
        [ 0.74901961,  0.77647059,  0.8       ],
        [ 0.74509804,  0.77254902,  0.79607843]]])
```

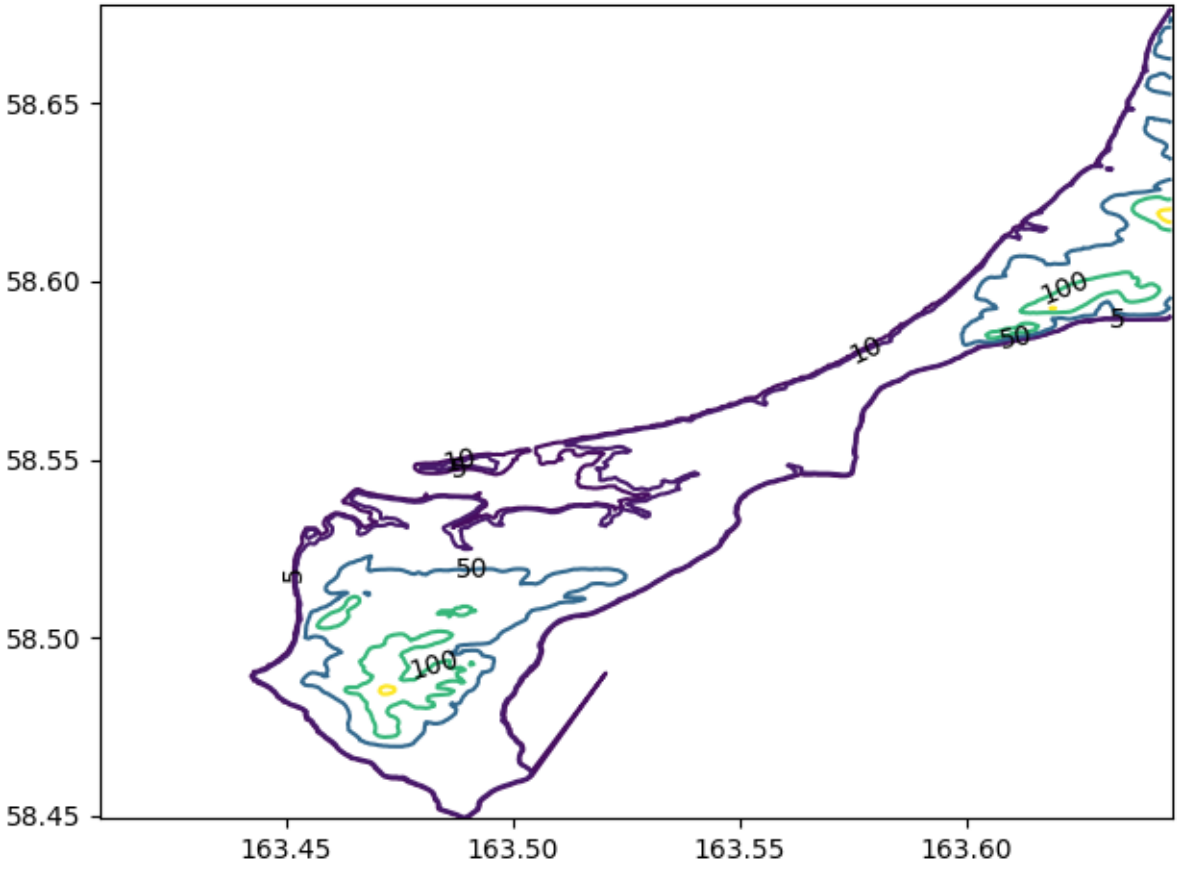
# *matplotlib.image*

```
img = mpimg.imread('DSC02766.jpg')/255
imgplot = plt.imshow(img[:,:,0],cmap='seismic')
plt.colorbar()
```



# Контуры, изолинии

```
x,y,z = np.loadtxt('/mnt/data/topo/N58E163.hgt.dat.cropped', unpack=True)
levels = [0, 5, 10, 50, 100, 150]
cnt = plt.tricontour(x,y,z,levels)
plt.clabel(cnt, fmt='%d', colors='k', inline=False)
```



## I. Самые простые графические команды:

- pyplot.scatter() - маркер или точечное рисование
- pyplot.plot() - ломаная линия
- pyplot.text() - нанесение текста

## II. Диаграммы:

- pyplot.bar(), pyplot.barh(), pyplot.barbs(), broken_barh() - столбчатая диаграмма
- pyplot.hist(), pyplot.hist2d(), pyplot.hlines - гистограмма
- pyplot.pie() - круговая диаграмма
- pyplot.boxplot() - "ящик с усами" (boxwhisker)
- pyplot.errorbar() - оценка погрешности, "усы"

## III. Изображения в изолиниях:

- pyplot.contour() - изолинии
- pyplot.contourf() - изолинии с послойной окраской

## IV. Отображения:

- pyplot.pcolor(), pyplot.pcolormesh() - псевдоцветное изображение матрицы (2D массива)
- pyplot.imshow() - вставка графики (пиксели + сглаживание)
- pyplot.matshow() - отображение данных в виде квадратов

## V. Заливка:

- pyplot.fill() - заливка многоугольника
- pyplot.fill_between(), pyplot.fill_betweenx() - заливка между двумя линиями

## VI. Векторные диаграммы:

- pyplot.streamplot() - линии тока
- pyplot.quiver() - векторное поле