In [48]:

```
%matplotlib inline
```

In [49]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [50]:

```
print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', 'bmh', 'classic', 'dark_backgro
und', 'fast', 'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seabor
n-bright', 'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-palette',
'seaborn-darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook',
'seaborn-paper', 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk', 'seab
orn-ticks', 'seaborn-white', 'seaborn-whitegrid', 'tableau-colorblind10']
```

In [51]:

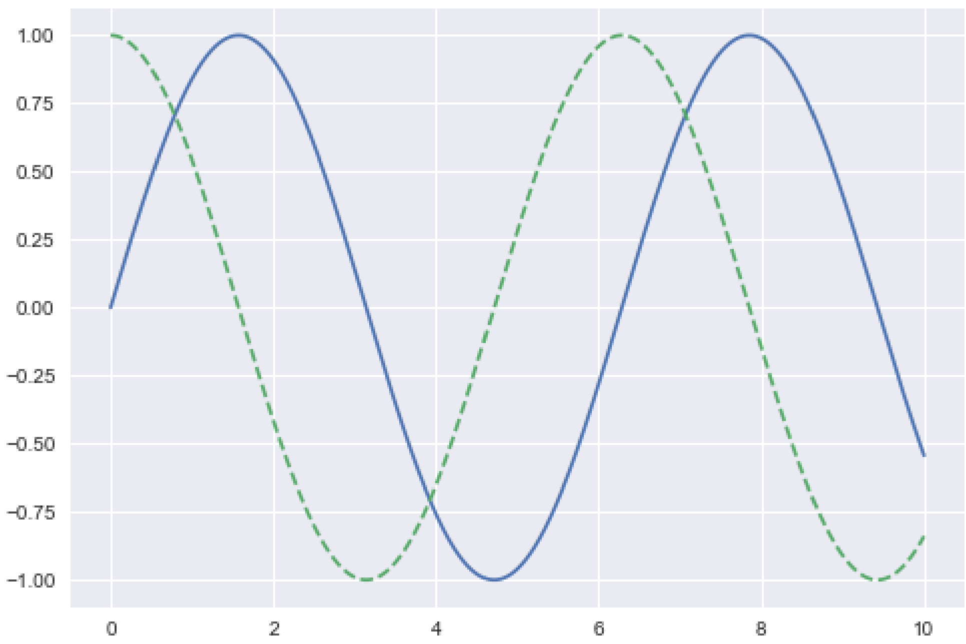
```
plt.style.use('seaborn')
#plt.style.use('classic')
```

In [52]:

```
x = np.linspace(0, 10, 10000)
fig = plt.figure()
plt.plot(x,np.sin(x))
plt.plot(x,np.cos(x),'--')
```

Out[52]:

```
[<matplotlib.lines.Line2D at 0x169447c6670>]
```



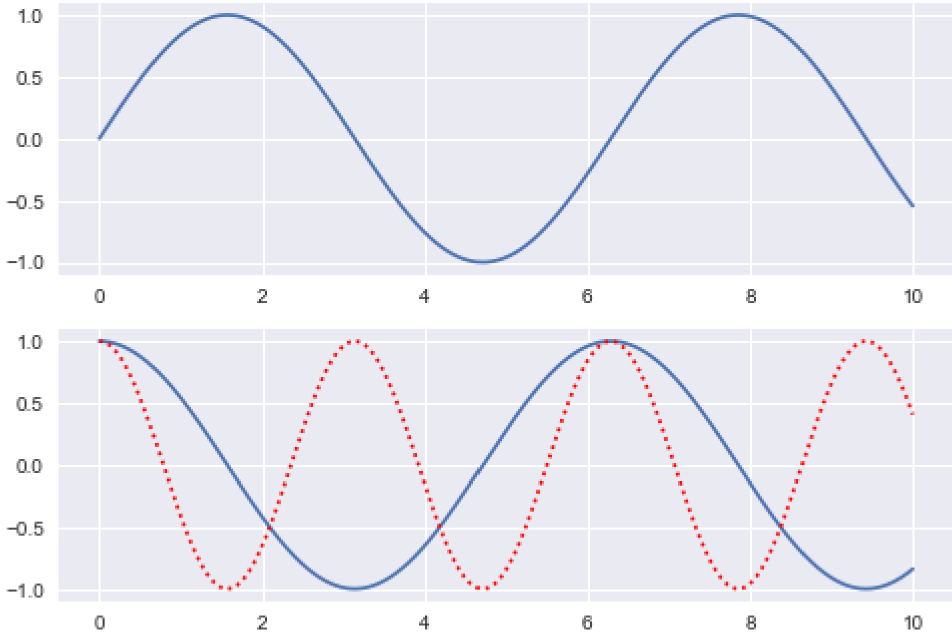
Объектно-ориентированный интерфейс

In [53]:

```
# сетка графиков
# ax - массив из двух объектов Axes (систем координат)
# fig - рисунок
fig, ax = plt.subplots(2)
ax[0].plot(x, np.sin(x))
ax[1].plot(x, np.cos(x))
ax[1].plot(x, np.cos(x*2), linestyle=':', color='red')
```

Out[53]:

```
[<matplotlib.lines.Line2D at 0x1694484aeb0>]
```

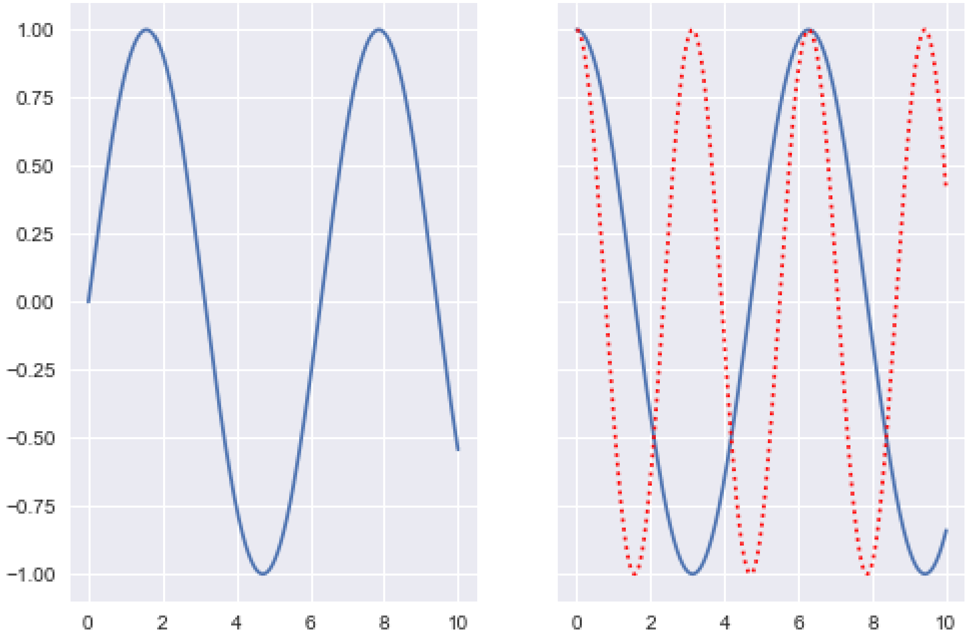


In [54]:

```
# одна строка, две колонки с общими осями
fig, ax = plt.subplots(1,2, sharex='col', sharey='row')
ax[0].plot(x, np.sin(x))
ax[1].plot(x, np.cos(x))
ax[1].plot(x, np.cos(x*2), linestyle=':', color='red')
```

Out[54]:

```
[<matplotlib.lines.Line2D at 0x169448ecc40>]
```



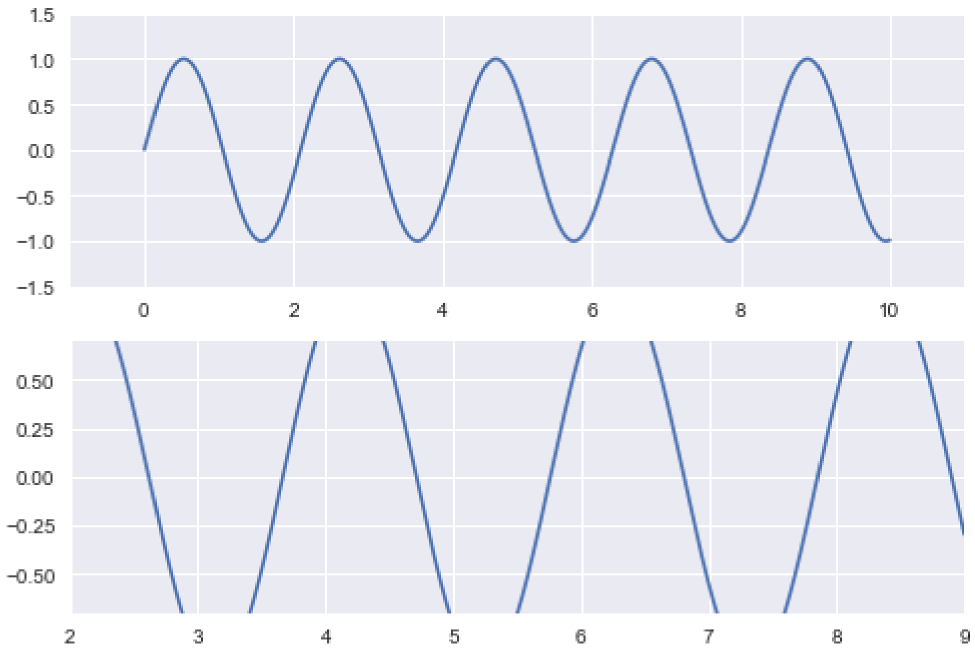
Пределы осей координат

In [55]:

```
fig, ax = plt.subplots(2)
ax[0].plot(x, np.sin(x*3))
ax[0].axis([-1,11,-1.5,1.5])
ax[1].plot(x, np.cos(x*3))
ax[1].axis([2,9,-0.7,0.7])
```

Out[55]:

```
(2.0, 9.0, -0.7, 0.7)
```



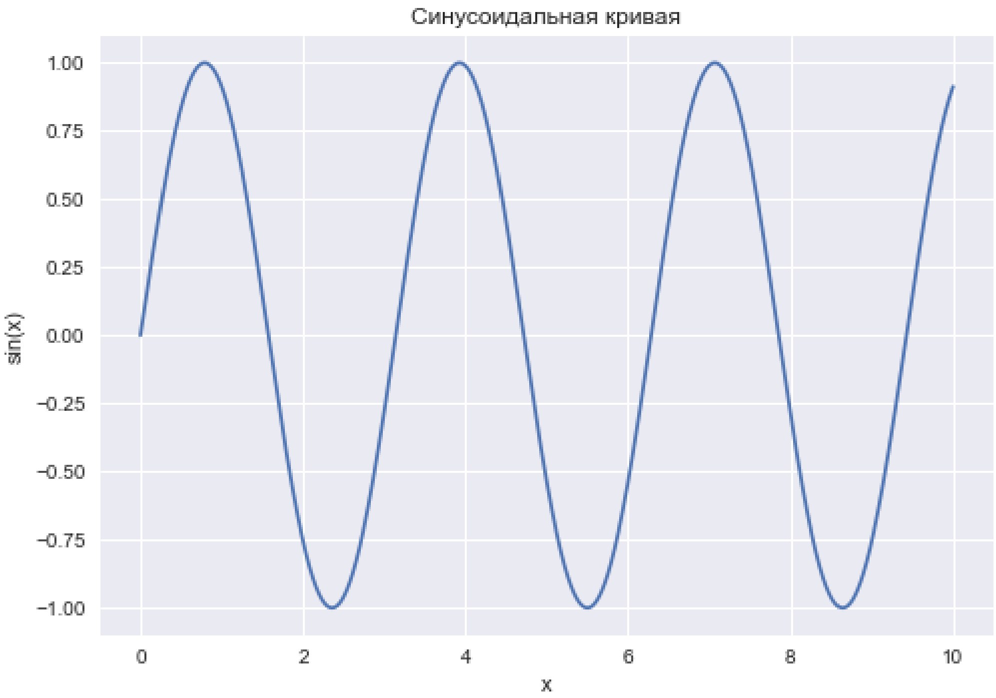
Метки на графике

In [56]:

```
fig, ax = plt.subplots(1)
ax.plot(x, np.sin(x*2))
ax.set_title('Синусоидальная кривая')
ax.set_xlabel("x")
ax.set_ylabel("sin(x)")
```

Out[56]:

```
Text(0, 0.5, 'sin(x)')
```



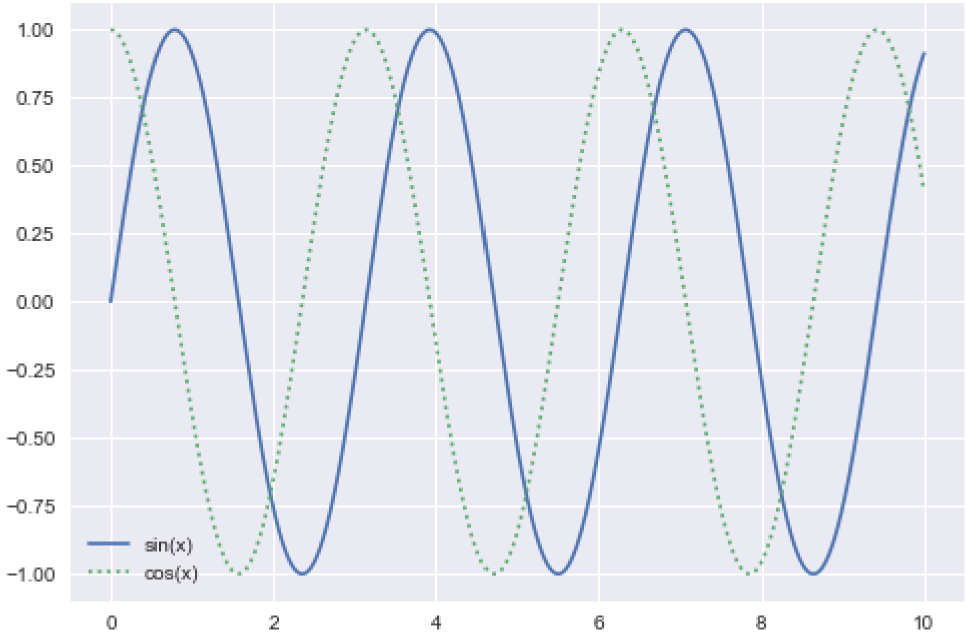
Легенда на графике

In [57]:

```
fig, ax = plt.subplots(1)
ax.plot(x, np.sin(x*2), label='sin(x)')
ax.plot(x, np.cos(x*2), ':', label='cos(x)')
ax.legend()
```

Out[57]:

```
<matplotlib.legend.Legend at 0x16945a0d1c0>
```



In [58]:

```
fig, ax = plt.subplots()
ax.plot(x, np.sin(x*2), label='sin(x)')
ax.plot(x, np.cos(x*2), ':', label='cos(x)')
ax.axis('equal')
# верхний левый угол с рамкой в две колонки
leg = ax.legend(loc='upper left', frameon=True, ncol=2)
```

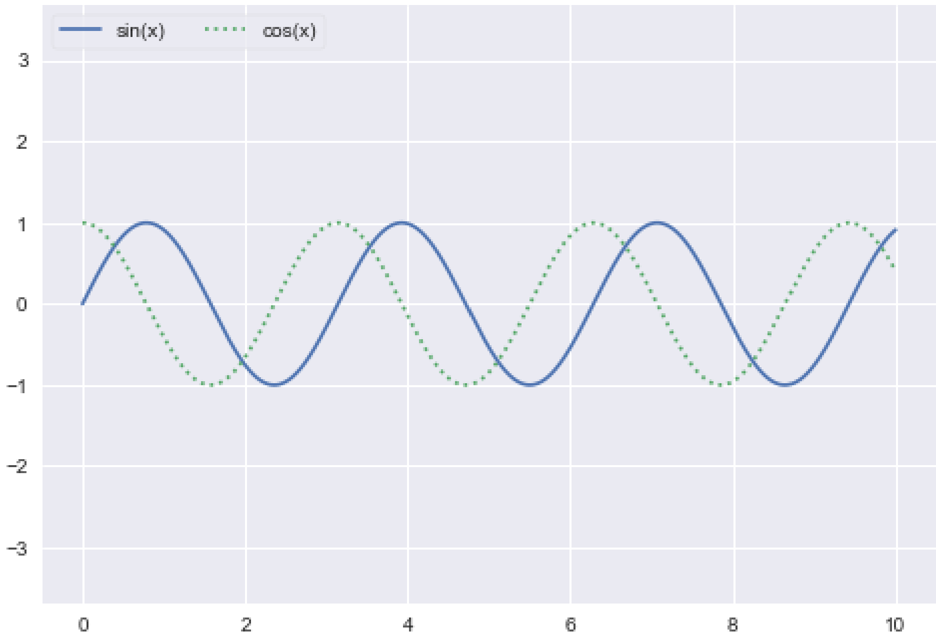


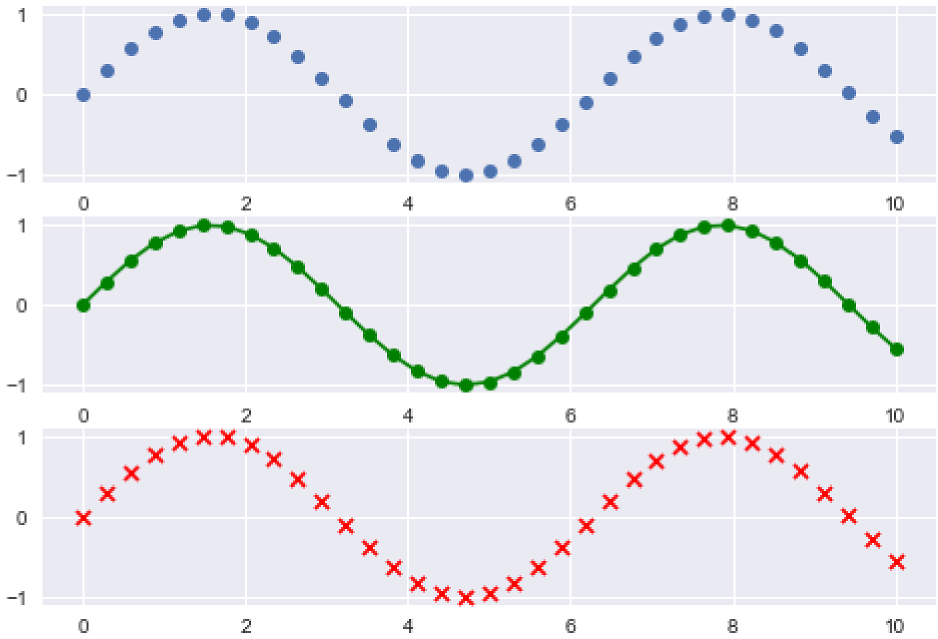
Диаграмма рассеяния

In [59]:

```
x = np.linspace(0, 10, 35)
y = np.sin(x)
fig, ax = plt.subplots(3)
ax[0].plot(x, y, 'o')
ax[1].plot(x, y, '-og')
ax[2].scatter(x, y, marker='x', color='red')
```

Out[59]:

```
<matplotlib.collections.PathCollection at 0x16945b67040>
```



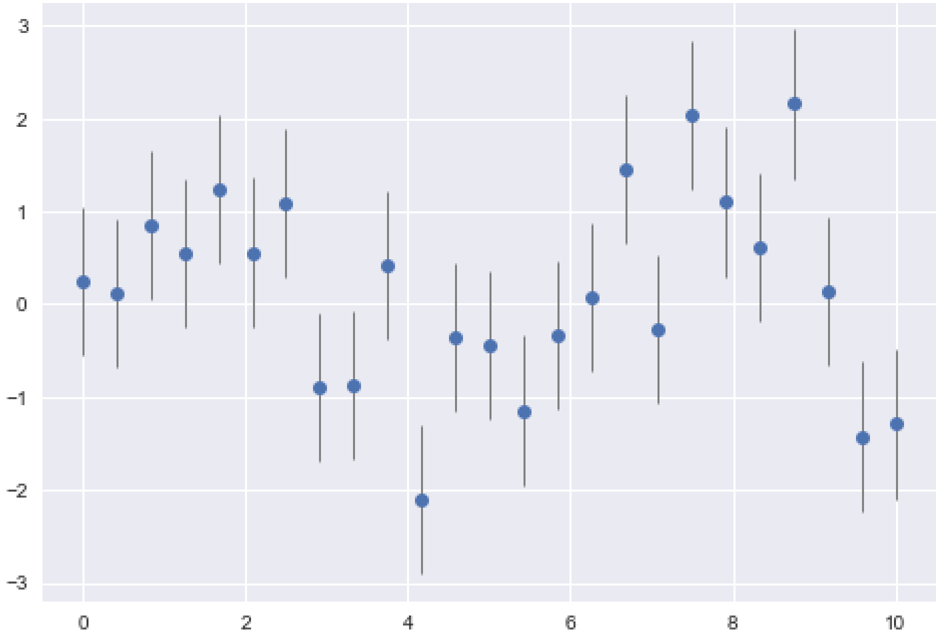
Погрешности

In [60]:

```
x = np.linspace(0, 10, 25)
dy = 0.8
y = np.sin(x)+dy*np.random.randn(25)
fig, ax = plt.subplots(1)
ax.errorbar(x, y, yerr=dy, fmt='o',
            ecolor='gray', elinewidth=1, capsize=0)
```

Out[60]:

<ErrorbarContainer object of 3 artists>



Контурные графики

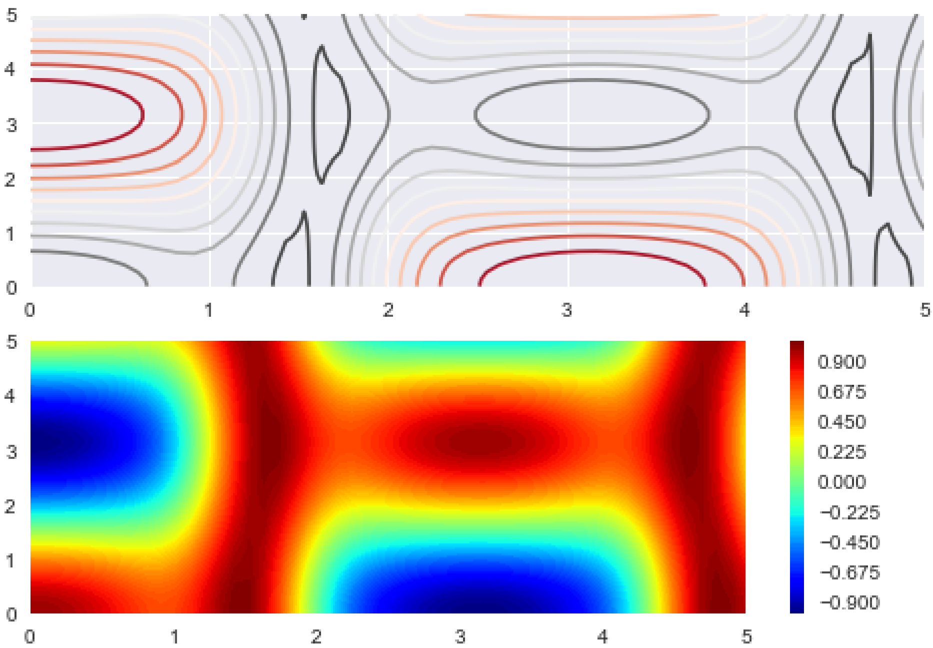
(отображение трехмерных данных в двухмерной плоскости)

In [61]:

```
def f(x, y):
    return np.sin(x) ** 10 + np.cos(y) * np.cos(x)
x = np.linspace(0, 5, 50)
y = np.linspace(0, 5, 40)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)
fig, ax = plt.subplots(2)
ax[0].contour(X, Y, Z, 10, cmap='RdGy')
cs1 = ax[1].contourf(X, Y, Z, 100, cmap='jet')
fig.colorbar(cs1, ax=ax[1])
```

Out[61]:

```
<matplotlib.colorbar.Colorbar at 0x16945d90070>
```

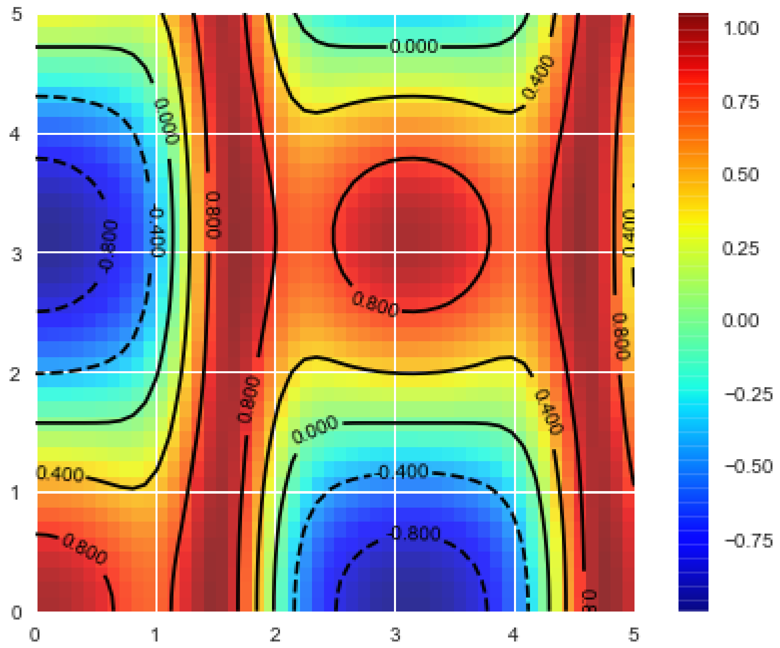


In [62]:

```
def f(x, y):
    return np.sin(x) ** 10 + np.cos(y) * np.cos(x)
x = np.linspace(0, 5, 50)
y = np.linspace(0, 5, 40)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)
fig, ax = plt.subplots(1)
contours = ax.contour(X, Y, Z, 5, colors='black')
ax.clabel(contours, inline=True, fontsize=10)
cs2 = ax.imshow(Z, extent=[0,5,0,5], origin='lower',
                   cmap='jet', alpha=0.8)
fig.colorbar(cs2, ax=ax)
```

Out[62]:

```
<matplotlib.colorbar.Colorbar at 0x16945e63eb0>
```



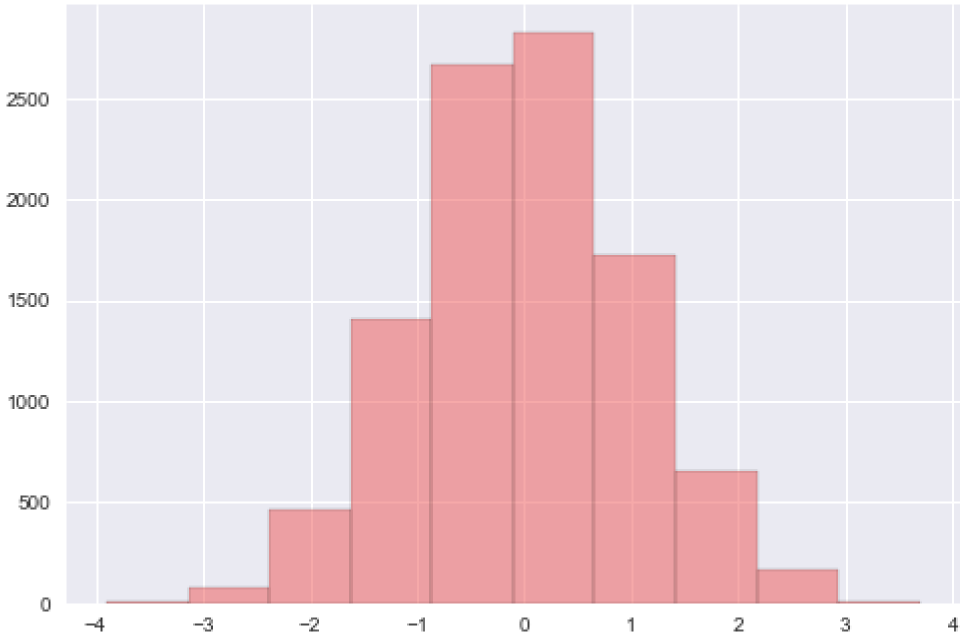
Гистограммы

In [63]:

```
data = np.random.randn(10000)
fig, ax = plt.subplots(1)
ax.hist(data, edgecolor='k', color="#ee5555", alpha=0.5)
```

Out[63]:

```
(array([   7.,   75.,  467., 1410., 2668., 2824., 1724.,  651.,  162.,
          12.]),
 array([-3.90056663, -3.14124055, -2.38191447, -1.62258838, -0.8632623 ,
        -0.10393622,  0.65538986,  1.41471595,  2.17404203,  2.93336811,
         3.69269419]),
 <a list of 10 Patch objects>)
```

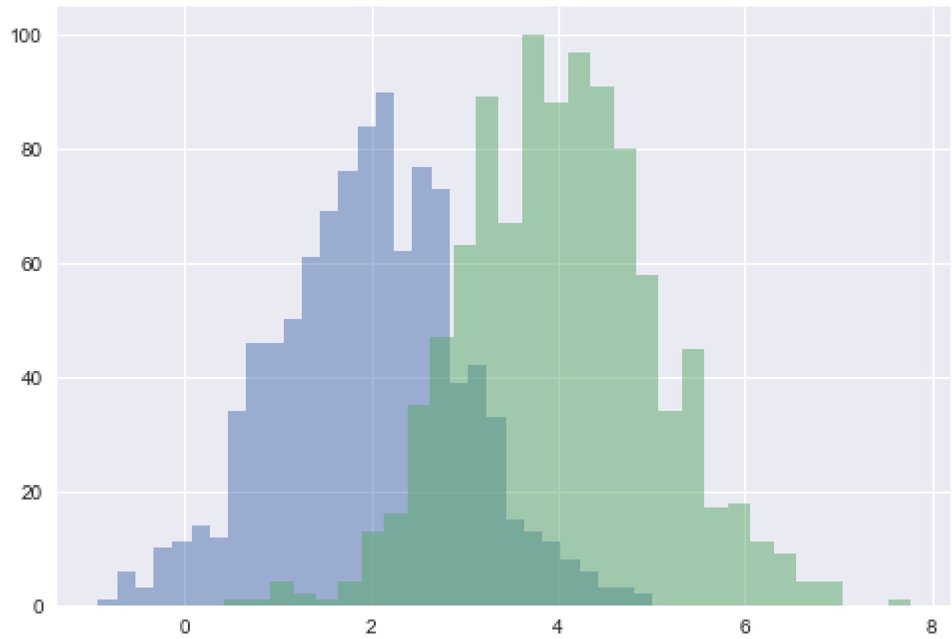


In [64]:

```
x1 = np.random.normal(2, 1, 1000)
x2 = np.random.normal(4, 1, 1000)
kwargs = dict(histtype='stepfilled', alpha=0.5, bins=30)
fig, ax = plt.subplots(1)
ax.hist(x1, **kwargs)
ax.hist(x2, **kwargs)
```

Out[64]:

```
(array([  1.,   1.,   4.,   2.,   1.,   4.,  13.,  16.,  35.,  47.,  63.,
         89.,  67., 100.,  88.,  97.,  91.,  80.,  58.,  34.,  45.,  17.,
         18.,  11.,   9.,   4.,   4.,   0.,   0.,   1.]),
 array([0.42555514, 0.6704574 , 0.91535966, 1.16026193, 1.40516419,
        1.65006646, 1.89496872, 2.13987098, 2.38477325, 2.62967551,
        2.87457778, 3.11948004, 3.3643823 , 3.60928457, 3.85418683,
        4.09908909, 4.34399136, 4.58889362, 4.83379589, 5.07869815,
        5.32360041, 5.56850268, 5.81340494, 6.05830721, 6.30320947,
        6.54811173, 6.793014  , 7.03791626, 7.28281853, 7.52772079,
        7.77262305]),
 <a list of 1 Patch objects>)
```

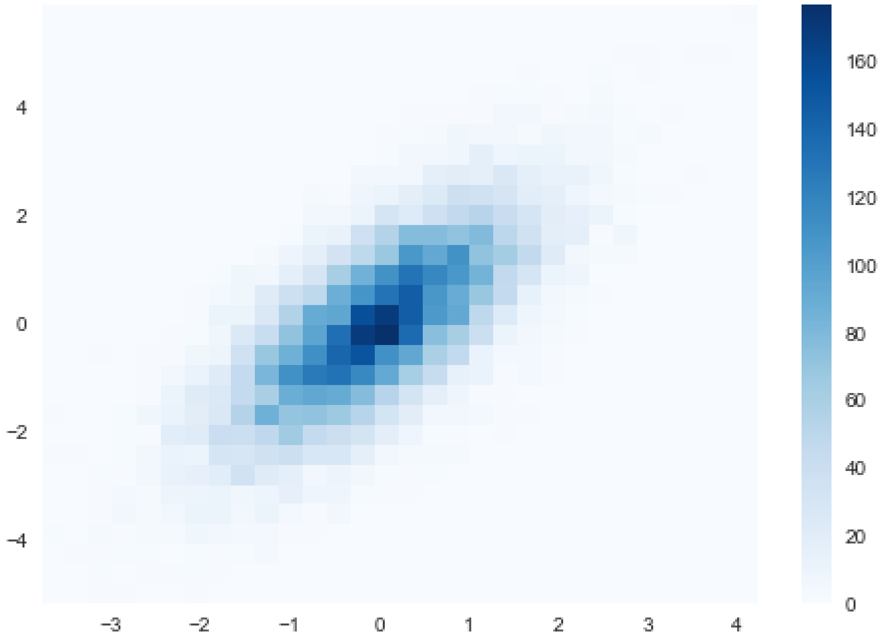


In [65]:

```
mean = [0, 0]
cov = [[1, 1], [1, 2]]
x, y = np.random.multivariate_normal(mean, cov, 10000).T
fig, ax = plt.subplots(1)
h, xedges, yedges, img = ax.hist2d(x, y, bins=30, cmap='Blues')
img
fig.colorbar(img, ax=ax)
```

Out[65]:

<matplotlib.colorbar.Colorbar at 0x16946f7cd30>

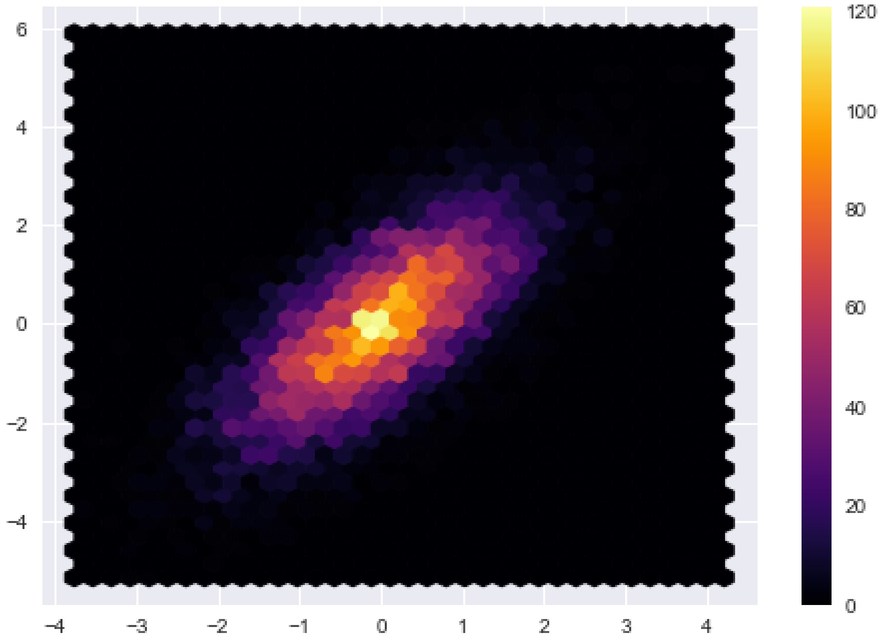


In [66]:

```
fig, ax = plt.subplots(1)
hb = ax.hexbin(x, y, gridsize=35, cmap='inferno')
fig.colorbar(hb, ax=ax)
```

Out[66]:

```
<matplotlib.colorbar.Colorbar at 0x16943abe190>
```

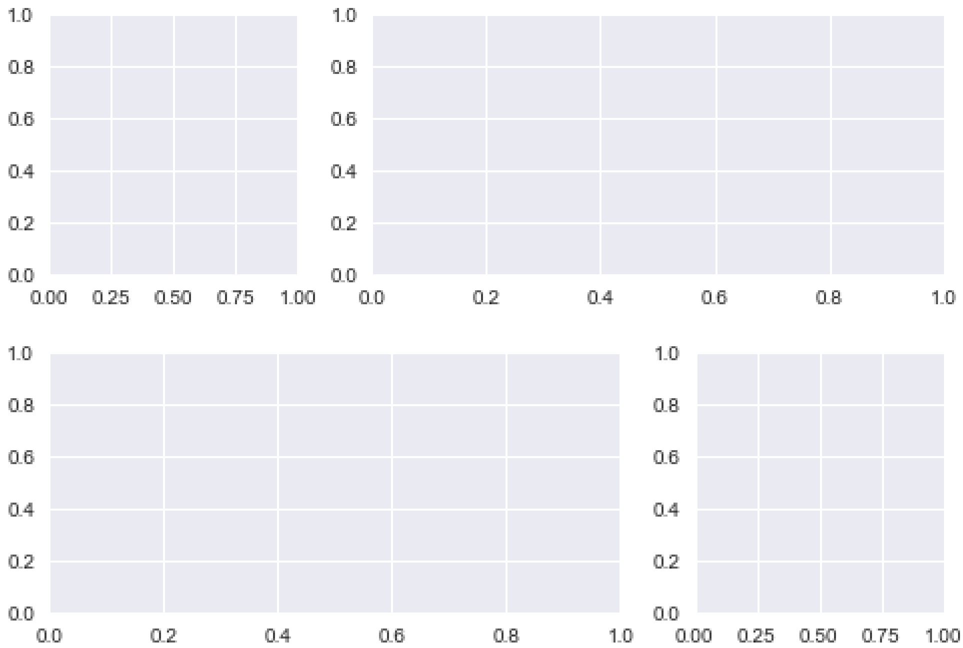


In [67]:

```
grid = plt.GridSpec(2, 3, wspace=0.3, hspace=0.3)
plt.subplot(grid[0, 0])
plt.subplot(grid[0, 1:])
plt.subplot(grid[1, :2])
plt.subplot(grid[1, 2])
```

Out[67]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16945913e20>
```

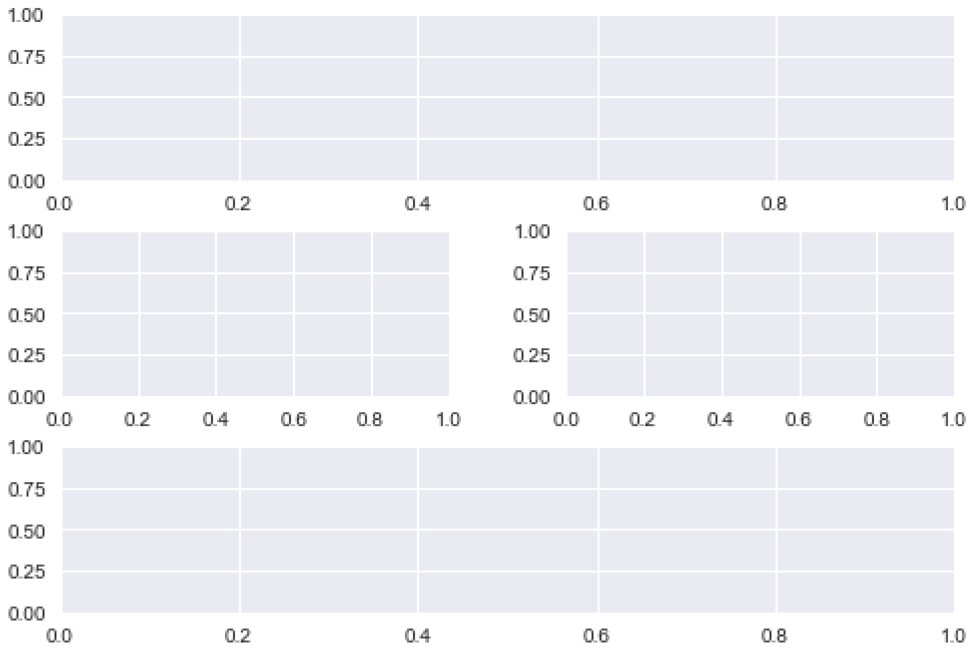


In [68]:

```
grid = plt.GridSpec(3, 2, wspace=0.3, hspace=0.3)
plt.subplot(grid[0, :2])
plt.subplot(grid[1, 0])
plt.subplot(grid[1, 1])
plt.subplot(grid[2, :2])
```

Out[68]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x169470bc850>
```



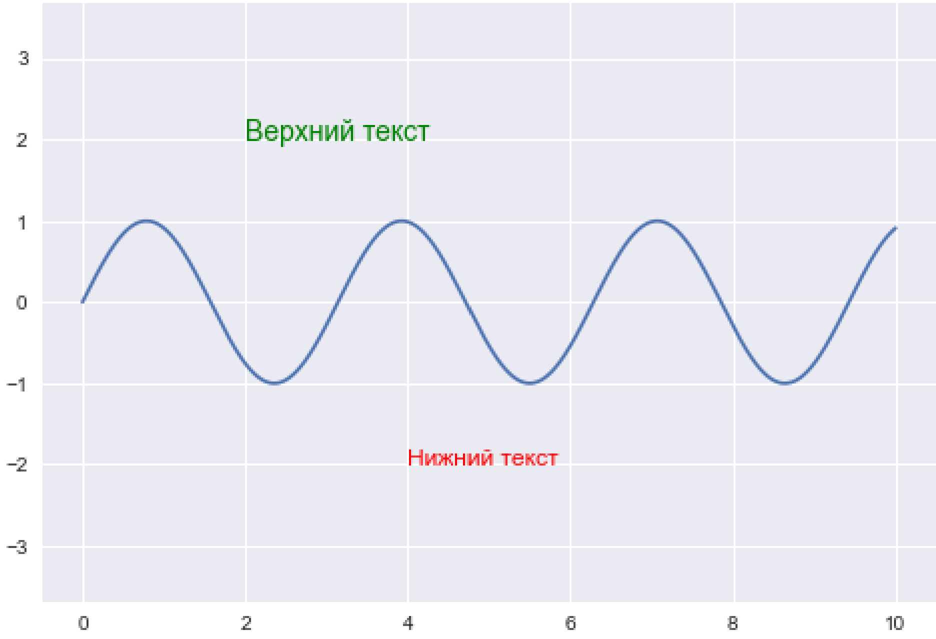
Text

In [69]:

```
x = np.linspace(0, 10, 10000)
fig, ax = plt.subplots()
ax.plot(x, np.sin(x*2))
ax.axis('equal')
ax.text(4, -2, "Нижний текст", size=12, color='r')
ax.text(2, 2, "Верхний текст", size=14, color='g')
```

Out[69]:

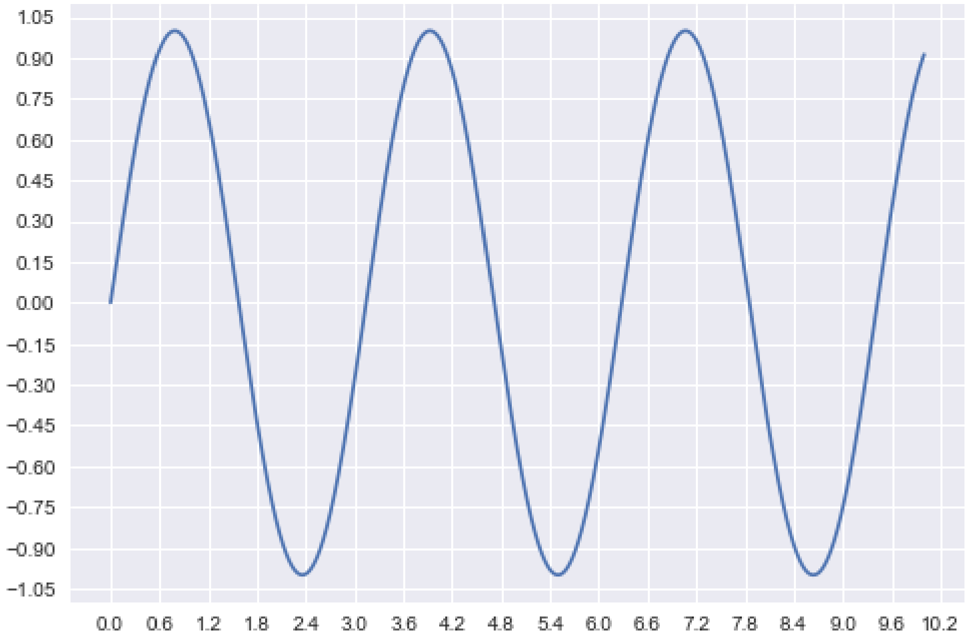
```
Text(2, 2, 'Верхний текст')
```

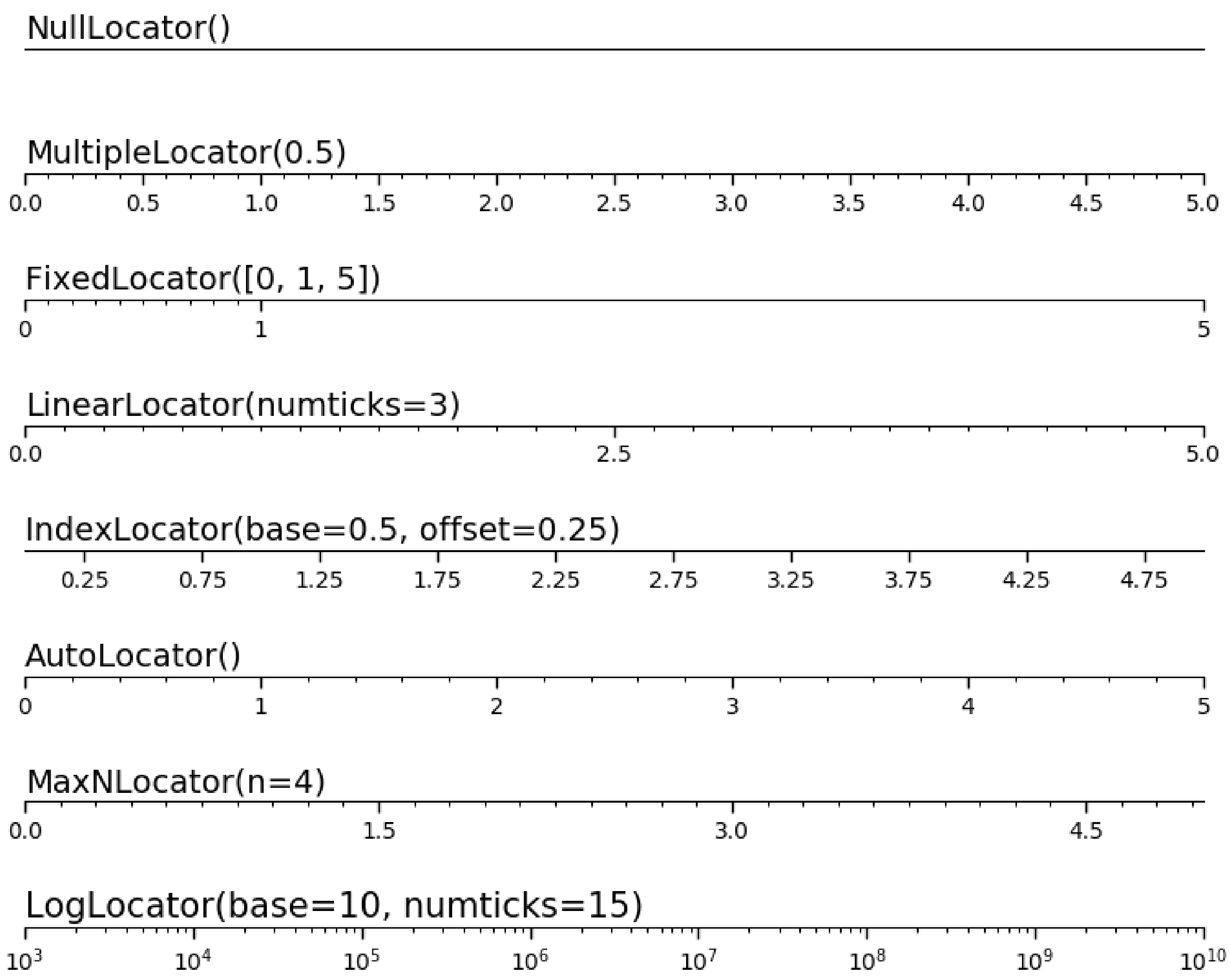


Locators (деления на графике)

In [70]:

```
import matplotlib.ticker as ticker
fig, ax = plt.subplots()
ax.plot(x, np.sin(x*2))
ax.xaxis.set_major_locator(ticker.MaxNLocator(20))
ax.yaxis.set_major_locator(ticker.MaxNLocator(20))
```

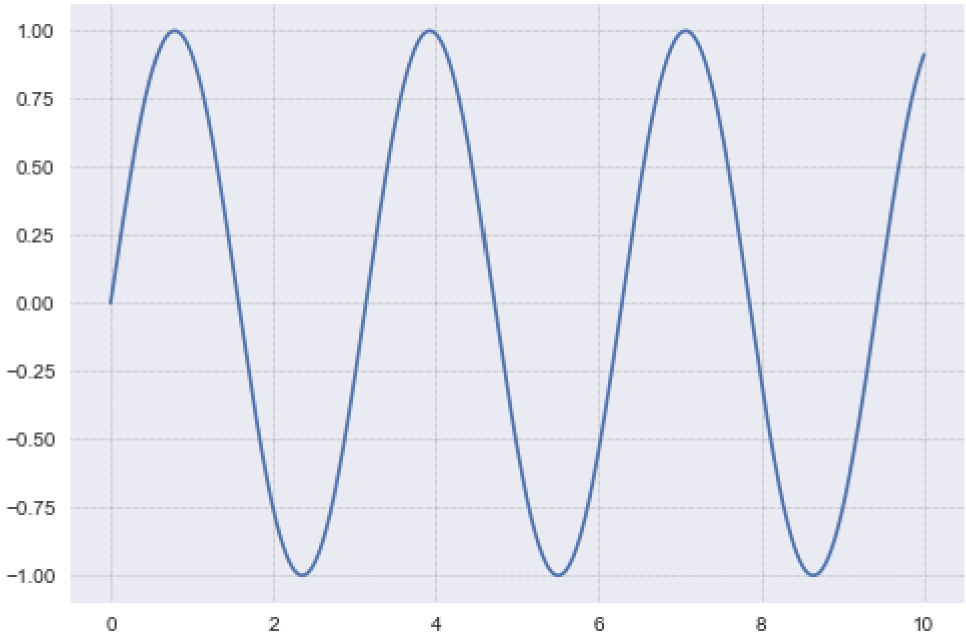



NullLocator()
MultipleLocator(0.5)
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
FixedLocator([0, 1, 5])
0 1 5
LinearLocator(numticks=3)
0.0 2.5 5.0
IndexLocator(base=0.5, offset=0.25)
0.25 0.75 1.25 1.75 2.25 2.75 3.25 3.75 4.25 4.75
AutoLocator()
0 1 2 3 4 5
MaxNLocator(n=4)
0.0 1.5 3.0 4.5
LogLocator(base=10, numticks=15)
10^3 10^4 10^5 10^6 10^7 10^8 10^9 10^{10}

Grid (Сетка)

In [71]:

```
fig, ax = plt.subplots()
ax.plot(x, np.sin(x*2))
ax.grid(True, c='gray', linewidth=0.5, linestyle=':')
```



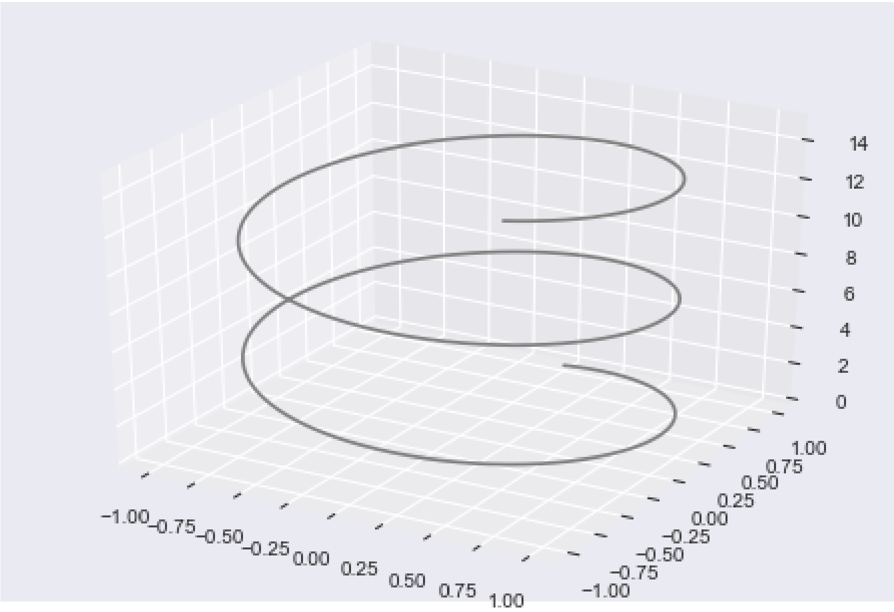
3D

In [72]:

```
from mpl_toolkits import mplot3d
fig = plt.figure()
ax = plt.axes(projection='3d')
zline = np.linspace(0,15,1000)
xline = np.sin(zline)
yline = np.cos(zline)
ax.plot3D(xline, yline, zline, c='gray')
```

Out[72]:

```
[<mpl_toolkits.mplot3d.art3d.Line3D at 0x169472aa130>]
```

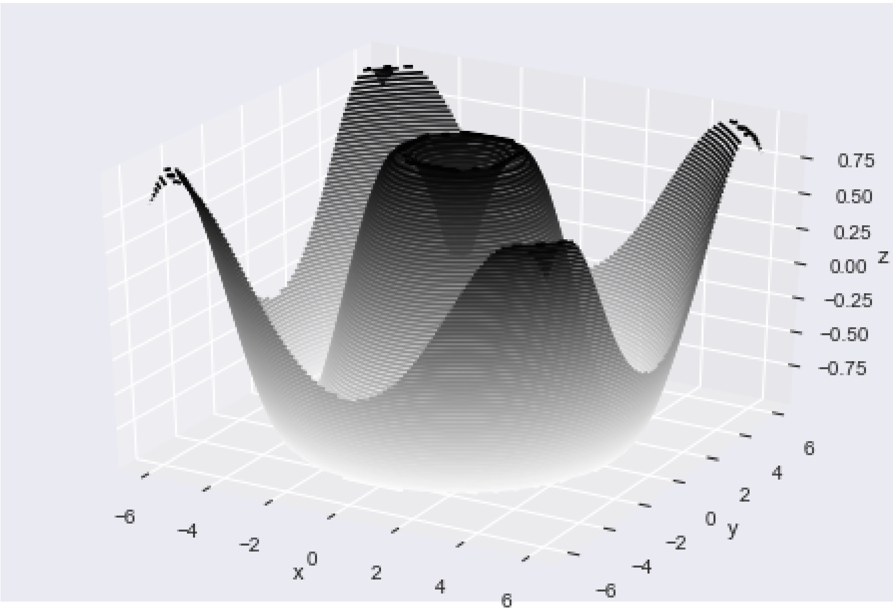


In [73]:

```
x = np.linspace(-6, 6, 30)
y = np.linspace(-6, 6, 30)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X ** 2 + Y ** 2))
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.contour3D(X, Y, Z, 70, cmap='binary')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
```

Out[73]:

```
Text(0.5, 0, 'z')
```

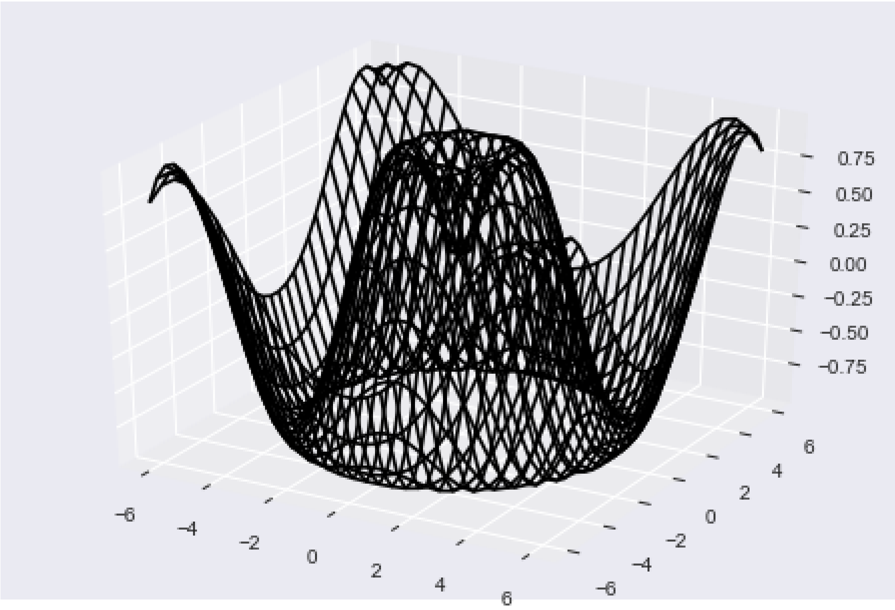


In [74]:

```
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.plot_wireframe(X, Y, Z, color='black')
```

Out[74]:

```
<mpl_toolkits.mplot3d.art3d.Line3DCollection at 0x169475f7f10>
```

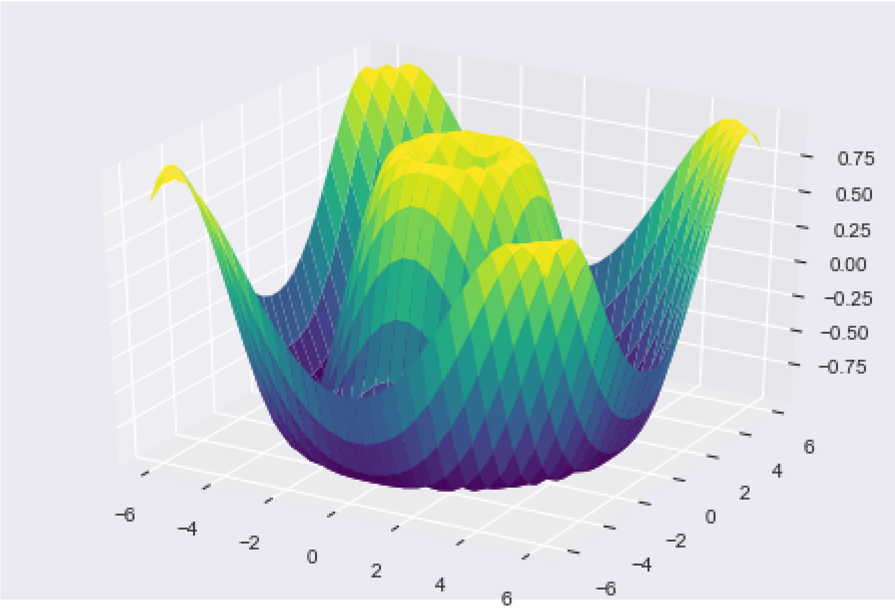


In [75]:

```
ax = plt.axes(projection='3d')
ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                cmap='viridis', edgecolor='none')
```

Out[75]:

```
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x1694768fd00>
```



Seaborn

In [76]:

```
import seaborn as sns
x1 = np.random.normal(2, 1, 1000)
x2 = np.random.normal(4, 1, 1000)
kwargs = dict(histtype='stepfilled', alpha=0.5, bins=30)
fig, ax = plt.subplots(2,2)
sns.distplot(x1, kde=False, ax=ax[0,0])
sns.distplot(x2, kde=False, ax=ax[0,0])
sns.distplot(x1, kde=True, ax=ax[1,0])
sns.distplot(x2, kde=True, ax=ax[1,0])
sns.distplot(x1, hist=False, rug=True, ax=ax[0, 1])
sns.distplot(x2, hist=False, rug=True, ax=ax[0, 1])
sns.distplot(x1, hist=False, kde_kws={"shade": True}, ax=ax[1, 1])
sns.distplot(x2, hist=False, kde_kws={"shade": True}, ax=ax[1, 1])
```

Out[76]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x169478112b0>
```