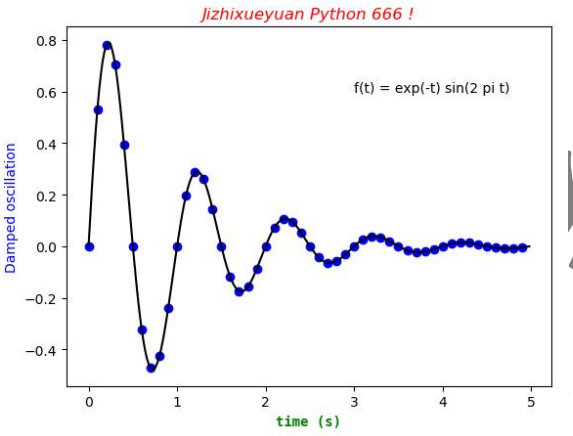
Matplotlib 快速绘图　课后测验

（命题人：极值学院讲师 张阳阳）

作业：使用 Python 编程，补全下列程序，然后在 Pycharm 中运行程序，如果运行得到的图形和题目所给图片一模一样，说明程序完全正确，否则错误。

1. 正确结果为：



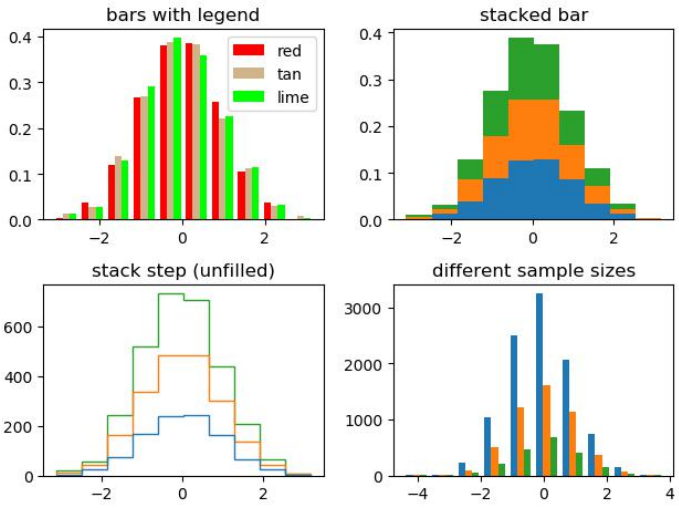
部分程序如下：

```
import matplotlib.pyplot as plt                    # 提示：导入必需的 python 库
import numpy as np                                 # 提示：导入必需的 python 库

def f(t):
    s1 = np.sin(2*np.pi*t)                         # 提示：s1 = sin(2πt)
    e1 = np.exp(-t)                                # 提示：e1 = e^(-t)
    return np.multiply(s1, e1)

t1 = np.arange(0.0, 5.0, 0.1)                      # 提示：t1 从 0.0 到 5.0，每隔 0.1 取一个数
t2 = np.arange(0.0, 5.0, 0.02)                     # 提示：t2 从 0.0 到 5.0，每隔 0.02 取一个数
fig, ax = plt.subplots()
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')          # 提示：作图，t1, f(t1)，蓝色线，以 o 表示点
plt.text(3.0, 0.6, 'f(t) = exp(-t) sin(2 pi t)')
ttext = plt.title('Jizhixueyuan Python 666 !')
ytext = plt.ylabel('Damped oscillation')           # 提示：纵坐标
xtext = plt.xlabel('time (s)')                     # 提示：横坐标
plt.setp(ttext, size='large', color='r', style='italic')
plt.setp(xtext, size='medium', name=['Courier', 'DejaVu Sans Mono'],
weight='bold', color='g')                          # 提示：绿色
plt.setp(ytext, size='medium', name=['Helvetica', 'DejaVu Sans'],
weight='light', color='b')                         # 提示：蓝色
plt.show()                                         # 提示：显示图片
```

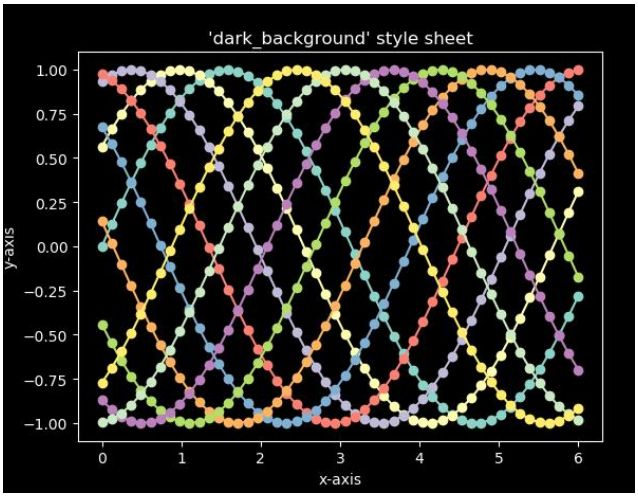
2. 正确结果为：



部分程序如下：

```
import numpy as np                                  # 提示：导入必需的 python 库
import matplotlib.pyplot as plt                     # 提示：导入必需的 python 库
np.random.seed(0)
n_bins = 10
x = np.random.randn(1000, 3)
fig, axes = plt.subplots(nrows=2, ncols=2)
ax0, ax1, ax2, ax3 = axes.flatten()
colors = ['red', 'tan', 'lime']                     # 提示：图例
ax0.hist(x, n_bins, normed=1, histtype='bar', color=colors, label=colors)
ax0.legend(prop={'size': 10})
ax0.set_title('bars with legend')                   # 提示：标题
ax1.hist(x, n_bins, normed=1, histtype='bar', stacked=True)
ax1.set_title('stacked bar')                        # 提示：标题
ax2.hist(x, n_bins, histtype='step', stacked=True, fill=False)
ax2.set_title('stack step (unfilled)')              # 提示：标题
# Make a multiple-histogram of data-sets with different length.
x_multi = [np.random.randn(n) for n in [10000, 5000, 2000]]
# 提示：包含 10000, 5000, 2000
ax3.hist(x_multi, n_bins, histtype='bar')           # 提示：直方图的类型为 bar
ax3.set_title('different sample sizes')             # 提示：标题
fig.tight_layout()
plt.show()                                          # 提示：显示图片
```

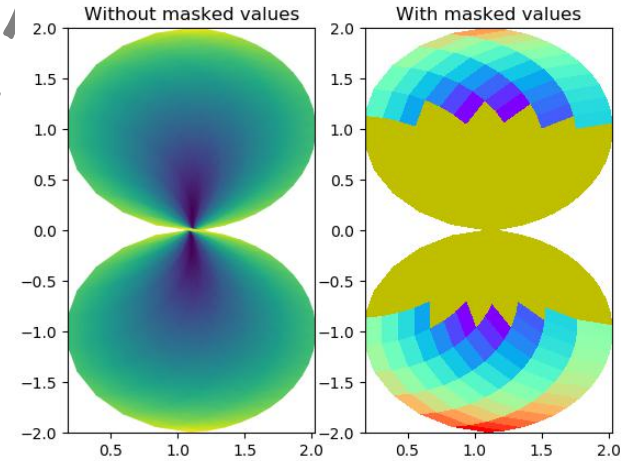
3. 正确结果为：



部分程序如下：

```
import numpy as np                          # 提示：导入必需的 python 库
import matplotlib.pyplot as plt             # 提示：导入必需的 python 库
plt.style.use('dark_background')
fig, ax = plt.subplots()
L = 6
x = np.linspace(0, L)                       # 提示：x 从 0 到 L 等间距取点
ncolors = len(plt.rcParams['axes.prop_cycle'])
shift = np.linspace(0, L, ncolors, endpoint=False)
for s in shift:                             # 提示：循环
    ax.plot(x, np.sin(x + s), 'o-')         # 提示：作图，x, sin(x + s), “o-”
ax.set_xlabel('x-axis')                     # 提示：横坐标
ax.set_ylabel('y-axis')                     # 提示：纵坐标
ax.set_title("'dark_background' style sheet")  # 提示：标题
plt.show()                                  # 提示：显示图片
```

4. 正确结果为：

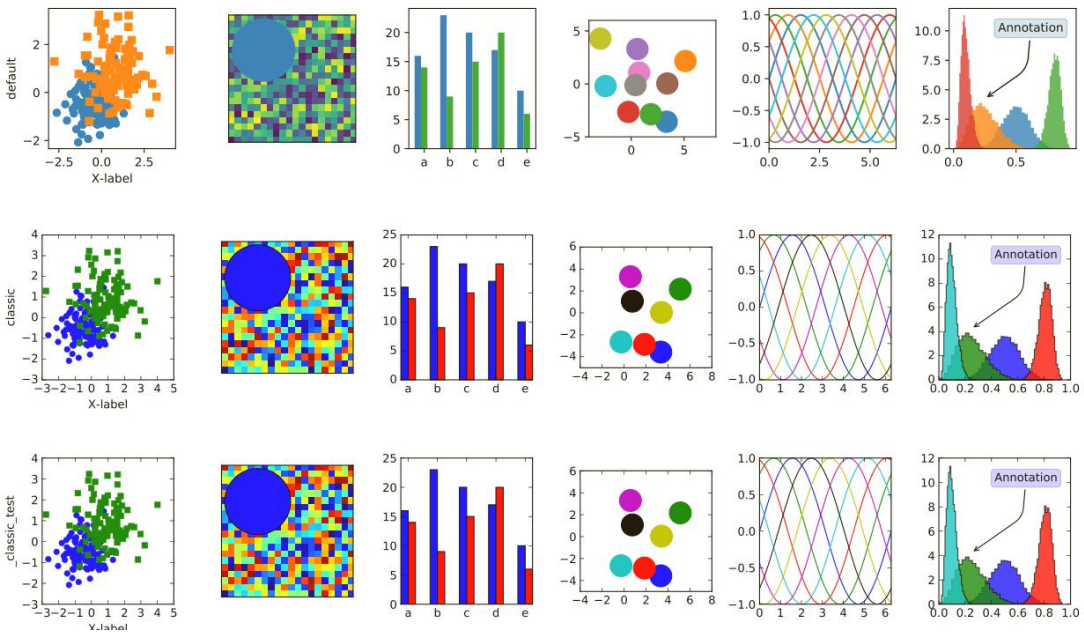


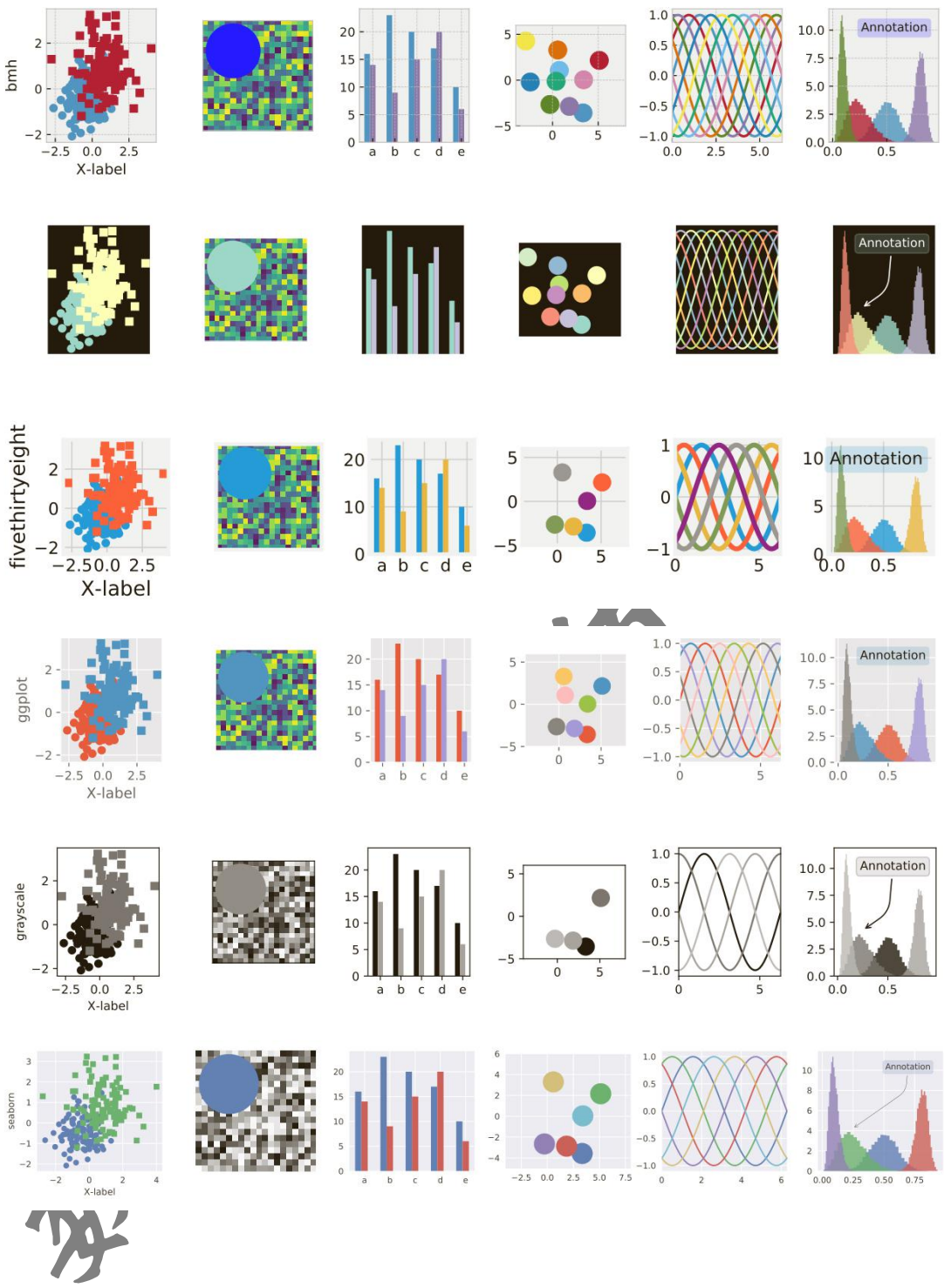
部分程序如下：

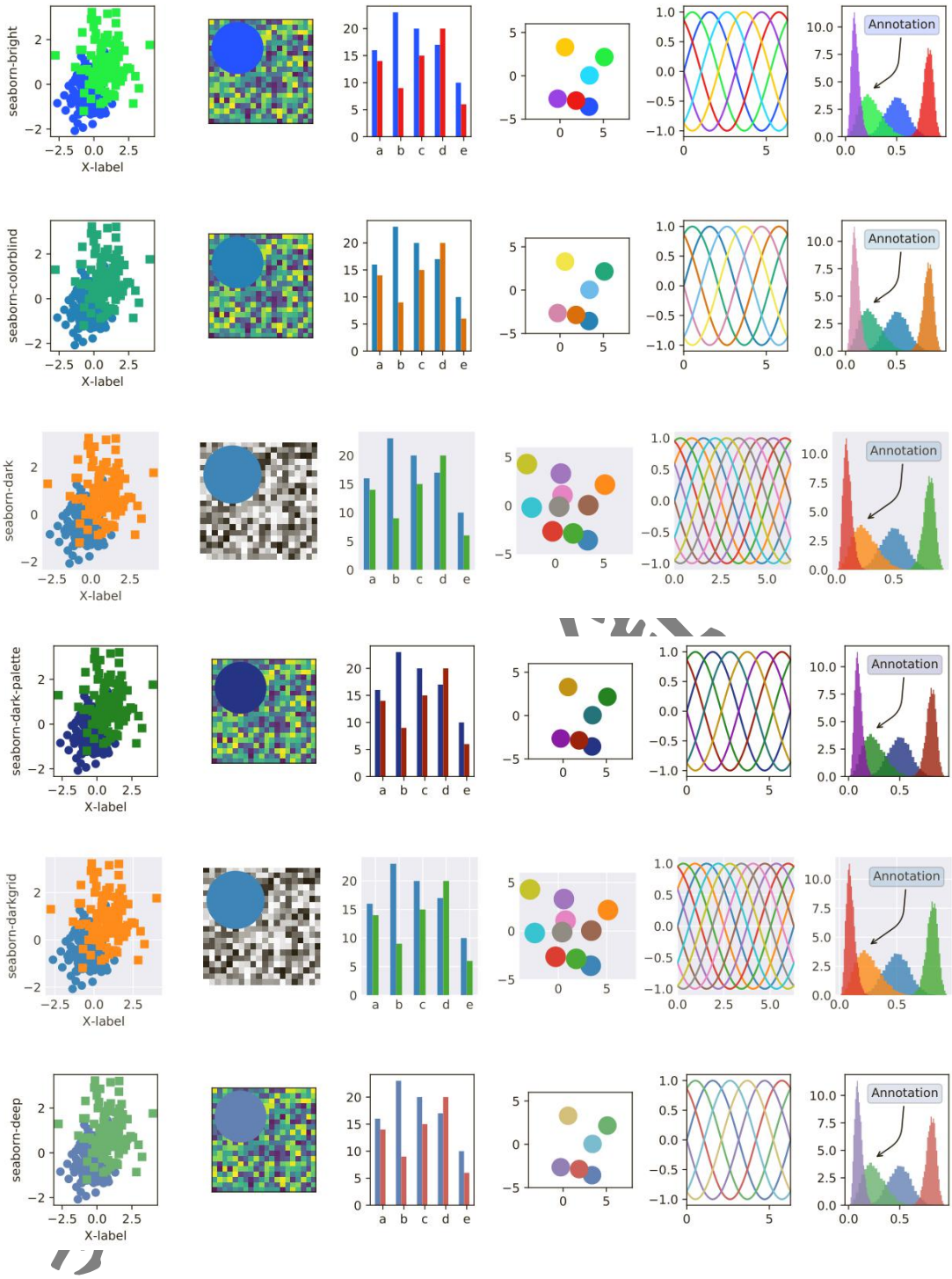
```
import numpy as np                          # 提示：导入必需的 python 库
```

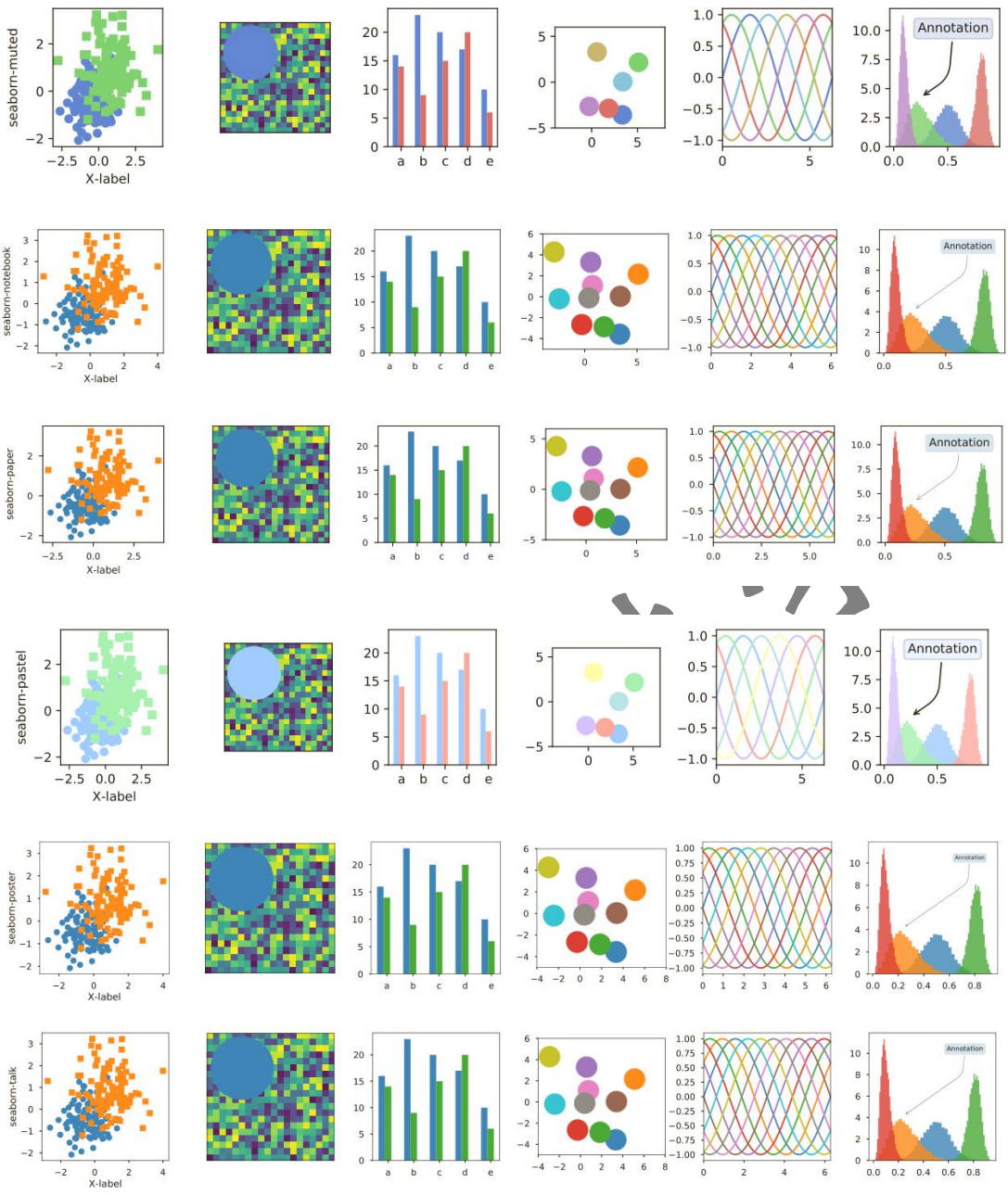
```
from matplotlib.pyplot import figure, show, savefig
from matplotlib import cm, colors
from numpy import ma
n = 12
x = np.linspace(-1.5, 1.5, n)
y = np.linspace(-1.5, 1.5, n*2)
X, Y = np.meshgrid(x, y)                    # 提示：网格化
Qx = np.cos(Y) - np.cos(X)                  # 提示：Qx = cosY- cosX
Qz = np.sin(Y) + np.sin(X)                  # 提示：Qx = sinY + sinX
Qx = (Qx + 1.1)
Z = np.sqrt(X**2 + Y**2)/5                  # 提示：Z=sqrt(X^2 + Y^2)/5
Z = (Z - Z.min()) / (Z.max() - Z.min())     # 提示：对 Z 标准化，（Z-最小值）/（极差）
# The color array can include masked values:
Zm = ma.masked_where(np.fabs(Qz) < 0.5*np.amax(Qz), Z)
fig = figure()                              # 提示：创建一幅图片
ax = fig.add_subplot(121)
ax.pcolormesh(Qx, Qz, Z, shading='gouraud')
ax.set_title('Without masked values')
ax = fig.add_subplot(122)
# You can control the color of the masked region:
cmap = cm.rainbow                           # 提示：创建彩虹图
cmap.set_bad('y', 1.0)
ax.pcolormesh(Qx, Qz, Zm, cmap=cmap)
ax.set_title('With masked values')          # 提示：标题
show()                                      # 提示：显示图片
```

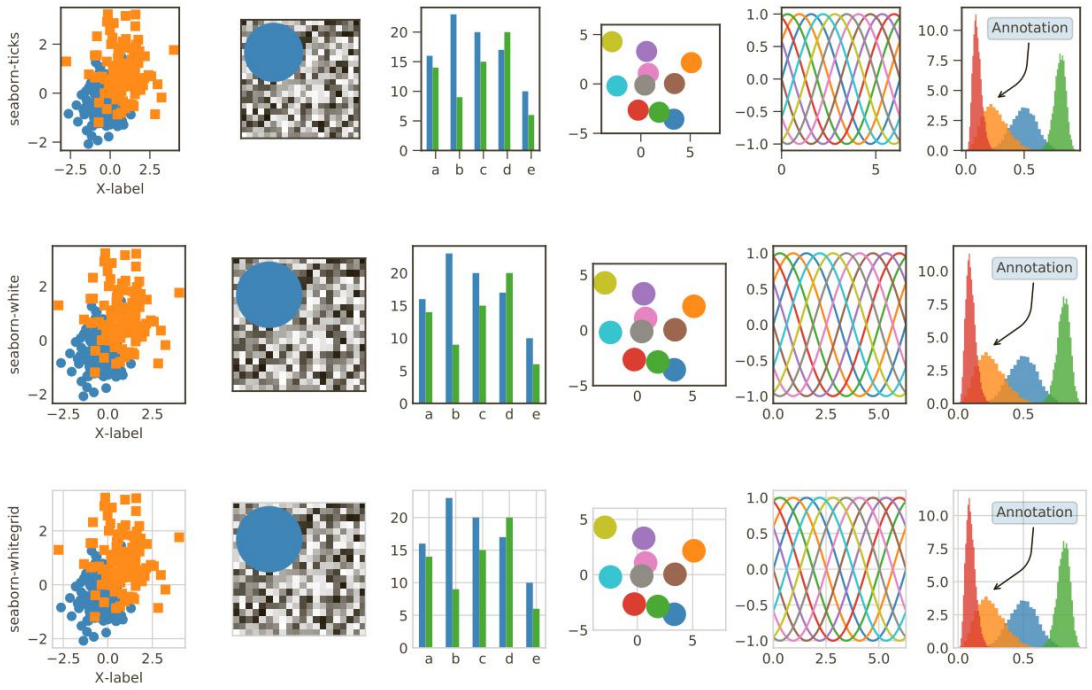
5. 正确结果为：




bmh
X-label
Annotation
fivethirtyeight
X-label
Annotation
ggplot
X-label
Annotation
grayscale
X-label
Annotation
seaborn
X-label
Annotation


seaborn-bright
seaborn-colorblind
seaborn-dark
seaborn-dark-palette
seaborn-darkgrid
seaborn-deep
X-label
Annotation
a b c d e


seaborn-muted
seaborn-notebook
seaborn-paper
seaborn-pastel
seaborn-poster
seaborn-talk
X-label
Annotation
a b c d e



部分程序如下：

```
import numpy as np                                  # 提示：导入必需的 python 库
import matplotlib.pyplot as plt                     # 提示：导入必需的 python 库


def plot_scatter(ax, prng, nb_samples=100):
    """Scatter plot.
    """
    for mu, sigma, marker in [(-.5, 0.75, 'o'), (0.75, 1., 's')]:
        x, y = prng.normal(loc=mu, scale=sigma, size=(2, nb_samples))
        ax.plot(x, y, ls='none', marker=marker)
    ax.set_xlabel('X-label')                        # 提示：横坐标
    return ax


def plot_colored_sinusoidal_lines(ax):
    """Plot sinusoidal lines with colors following the style color cycle.
    """
    L = 2 * np.pi                                   # 提示：L=2π
    x = np.linspace(0, L)                           # 提示：x 从 0 到 L 等间距取点
    nb_colors = len(plt.rcParams['axes.prop_cycle'])
    #提示：plt.rcParams['axes.prop_cycle']的长度
    shift = np.linspace(0, L, nb_colors, endpoint=False)
    for s in shift:
        ax.plot(x, np.sin(x + s), '-')              # 提示：作图，x, sin(x + s)
    ax.set_xlim([x[0], x[-1]])                      # 提示：限定 x 的取值范围为 x[0]到 x[-1]
    return ax
```

```
def plot_bar_graphs(ax, prng, min_value=5, max_value=25, nb_samples=5):
    """Plot two bar graphs side by side, with letters as x-tick labels.
    """
    x = np.arange(nb_samples)
  # 提示：x为从0开始，步长为1，取nb_samples个数组成的numpy.ndarray类型
    ya, yb = prng.randint(min_value, max_value, size=(2, nb_samples))
    width = 0.25
    ax.bar(x, ya, width)
    ax.bar(x + width, yb, width, color='C2')
    ax.set_xticks(x + width)
    ax.set_xticklabels(['a', 'b', 'c', 'd', 'e'])
    return ax


def plot_colored_circles(ax, prng, nb_samples=15):
    """Plot circle patches.
    NB: draws a fixed amount of samples, rather than using the length of
    the color cycle, because different styles may have different numbers
    of colors.
    """
    for sty_dict, j in zip(plt.rcParams['axes.prop_cycle'], range(nb_samples)):
        ax.add_patch(plt.Circle(prng.normal(scale=3, size=2),
        radius=1.0, color=sty_dict['color']))
    # Force the limits to be the same across the styles (because different
    # styles may have different numbers of available colors).
    ax.set_xlim([-4, 8])                    # 提示：设置x的显示范围为-4到8
    ax.set_ylim([-5, 6])                    # 提示：设置x的显示范围为-5到6
    ax.set_aspect('equal', adjustable='box') # to plot circles as circles
    return ax


def plot_image_and_patch(ax, prng, size=(20, 20)):
    """Plot an image with random values and superimpose a circular patch.
    """
    values = prng.random_sample(size=size)
    ax.imshow(values, interpolation='none')
    c = plt.Circle((5, 5), radius=5, label='patch')
    ax.add_patch(c)
    # Remove ticks
    ax.set_xticks([])
    ax.set_yticks([])


def plot_histograms(ax, prng, nb_samples=10000):
    """Plot 4 histograms and a text annotation.
    """
    params = ((10, 10), (4, 12), (50, 12), (6, 55))
```

```
    for a, b in params:
        values = prng.beta(a, b, size=nb_samples)
        ax.hist(values, histtype="stepfilled", bins=30, alpha=0.8, normed=True)
    # Add a small annotation.
    ax.annotate('Annotation', xy=(0.25, 4.25), xycoords='data',
    xytext=(0.9, 0.9), textcoords='axes fraction',
    va="top", ha="right",
    bbox=dict(boxstyle="round", alpha=0.2),
    arrowprops=dict(
    arrowstyle="->",
    connectionstyle="angle,angleA=-95,angleB=35,rad=10"),
    )
    return ax

def plot_figure(style_label=""):
    """Setup and plot the demonstration figure with a given style.
    """
    # Use a dedicated RandomState instance to draw the same "random" values
    # across the different figures.
    prng = np.random.RandomState(96917002)
    # Tweak the figure size to be better suited for a row of numerous plots:
    # double the width and halve the height. NB: use relative changes because
    # some styles may have a figure size different from the default one.
    (fig_width, fig_height) = plt.rcParams['figure.figsize']
    fig_size = [fig_width * 2, fig_height / 2]
    fig, axes = plt.subplots(ncols=6, nrows=1, num=style_label,
    figsize=fig_size, squeeze=True)
    axes[0].set_ylabel(style_label)
    plot_scatter(axes[0], prng)
    plot_image_and_patch(axes[1], prng)
    plot_bar_graphs(axes[2], prng)
    plot_colored_circles(axes[3], prng)
    plot_colored_sinusoidal_lines(axes[4])
    plot_histograms(axes[5], prng)
    fig.tight_layout()
    return fig

if __name__ == "__main__":
    # Setup a list of all available styles, in alphabetical order but
    # the `default` and `classic` ones, which will be forced resp. in
    # first and second position.
    style_list = list(plt.style.available)  # *new* list: avoids side effects.
    style_list.remove('classic')  # `classic` is in the list: first remove it.
    style_list.sort()
```

```
style_list.insert(0, u'default')
style_list.insert(1, u'classic')
# Plot a demonstration figure for every available style sheet.
for style_label in style_list:
    with plt.style.context(style_label):
        fig = plot_figure(style_label=style_label)
plt.show()                                    # 提示：显示图片
```