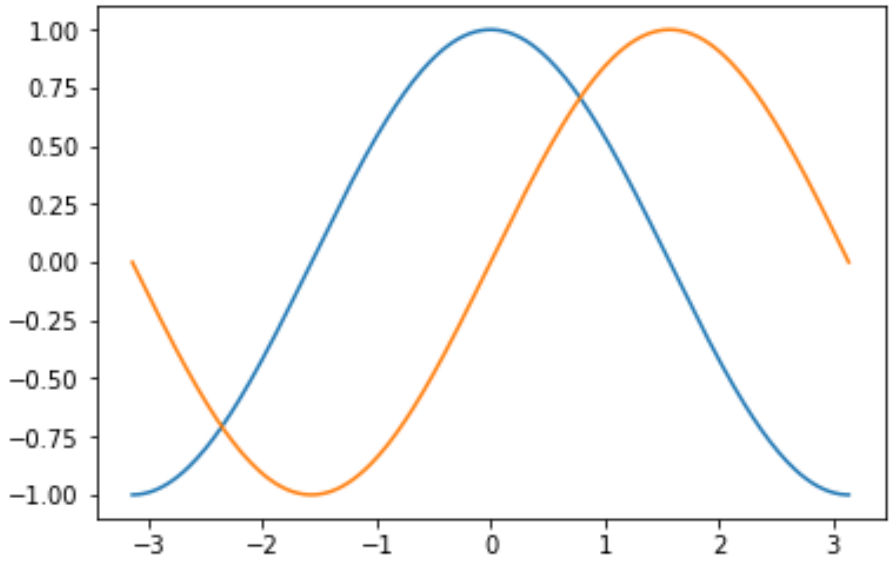
Programming for Data Science *(with Python)*

Lab 5

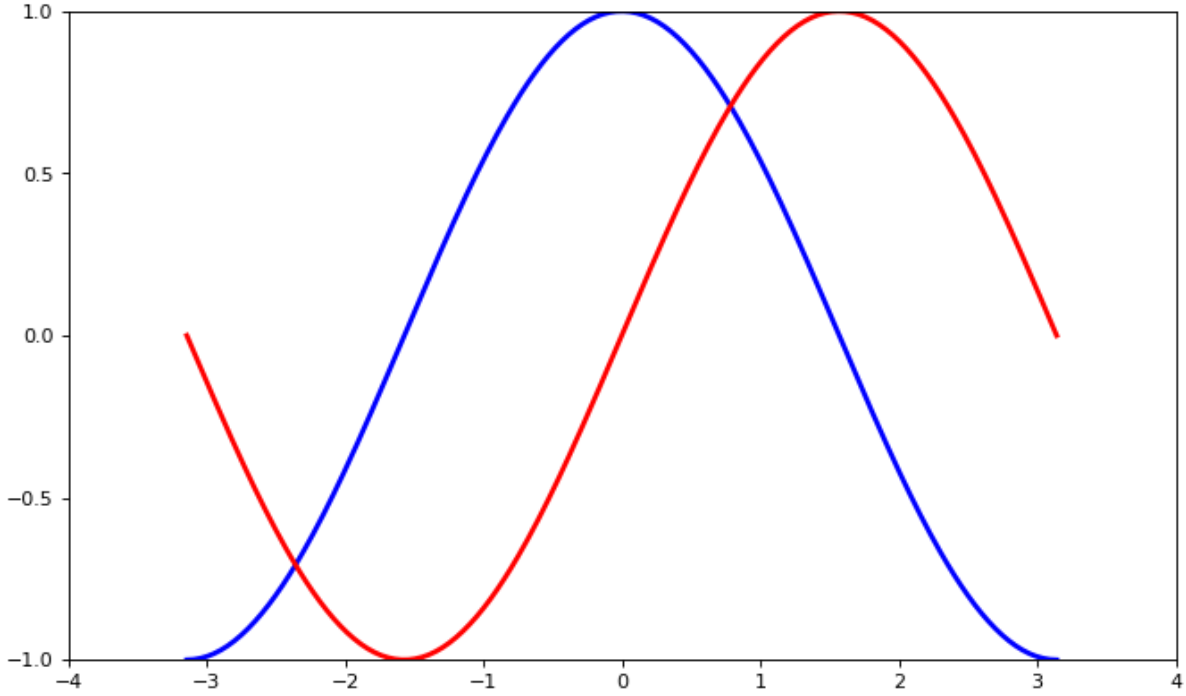
Bài 1. Viết chương trình theo mẫu

```
import numpy as np
import matplotlib.pyplot as plt
# Create a figure of size 8x6 inches, 80 dots per inch
plt.figure(figsize=(8, 6), dpi=80)
# Create a new subplot from a grid of 1x1
plt.subplot(1, 1, 1)
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C, S = np.cos(X), np.sin(X)
# Plot cosine with a blue continuous line of width 1 (pixels)
plt.plot(X, C, color="blue", linewidth=1.0, linestyle="-")
# Plot sine with a green continuous line of width 1 (pixels)
plt.plot(X, S, color="green", linewidth=1.0, linestyle="-")
# Set x limits
plt.xlim(-4.0, 4.0)
# Set x ticks
plt.xticks(np.linspace(-4, 4, 9, endpoint=True))
# Set y limits
plt.ylim(-1.0, 1.0)
# Set y ticks
plt.yticks(np.linspace(-1, 1, 5, endpoint=True))
# Save figure using 72 dots per inch
# plt.savefig("exercise_2.png", dpi=72)
# Show result on screen
plt.show()
```

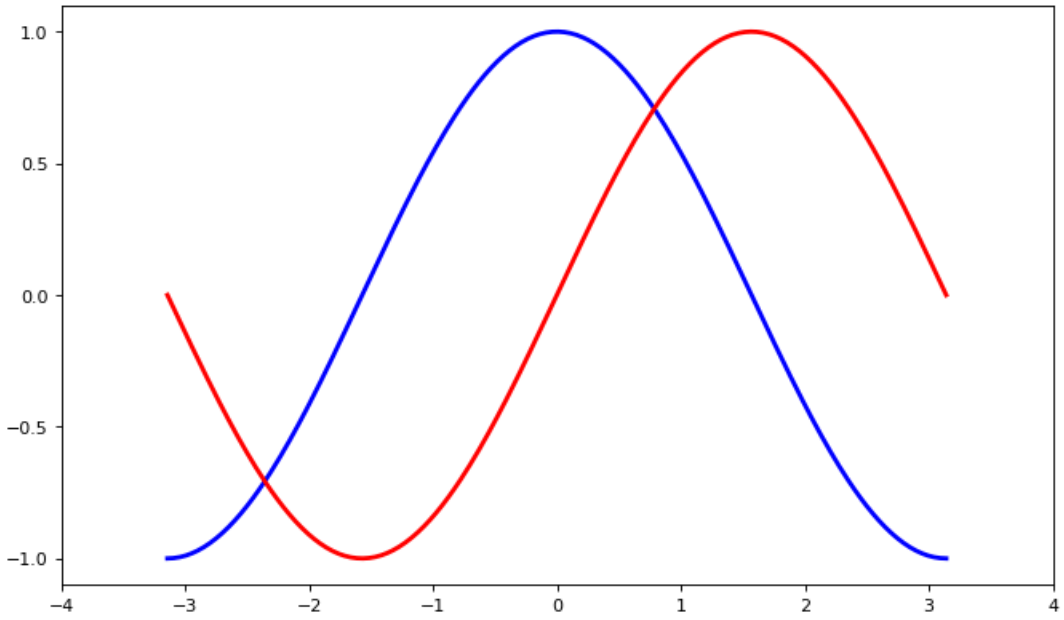
Kết quả



Bài 2. Điều chỉnh kích thước cửa sổ, độ rộng cả đường và màu sắc để có kết quả như sau



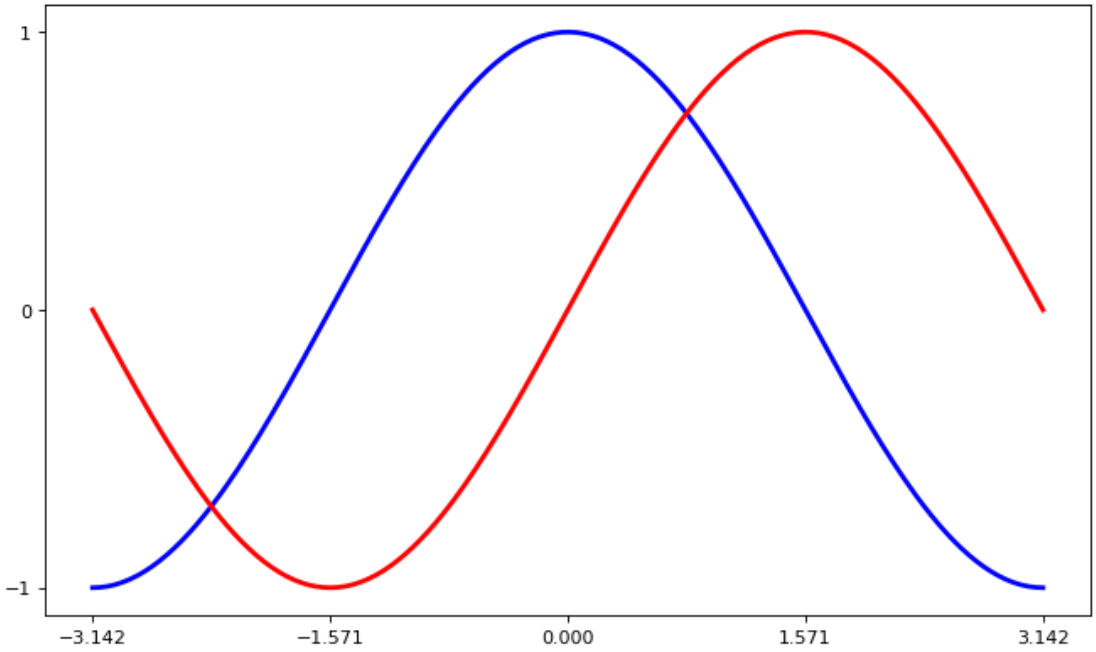
Bài 3. Điều chỉnh chương trình để có kết quả như sau



Gợi ý

```
# Set x limits
plt.xlim(X.min() * 1.1, X.max() * 1.1)
# Set y limits
plt.ylim(C.min() * 1.1, C.max() * 1.1)
```

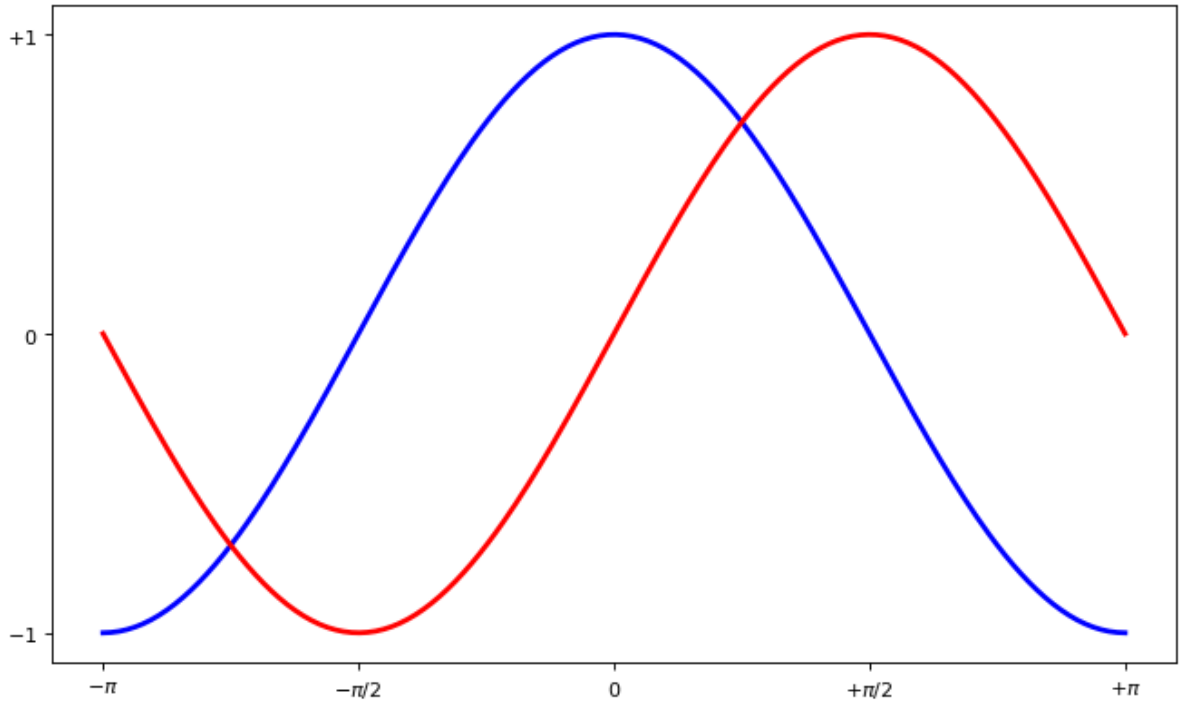
Bài 4. Điều chỉnh chương trình để có kết quả như sau



Gợi ý

```
# Set x ticks
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi])
# Set y ticks
plt.yticks([-1, 0, +1])
```

Bài 5 Điều chỉnh chương trình để có kết quả sau

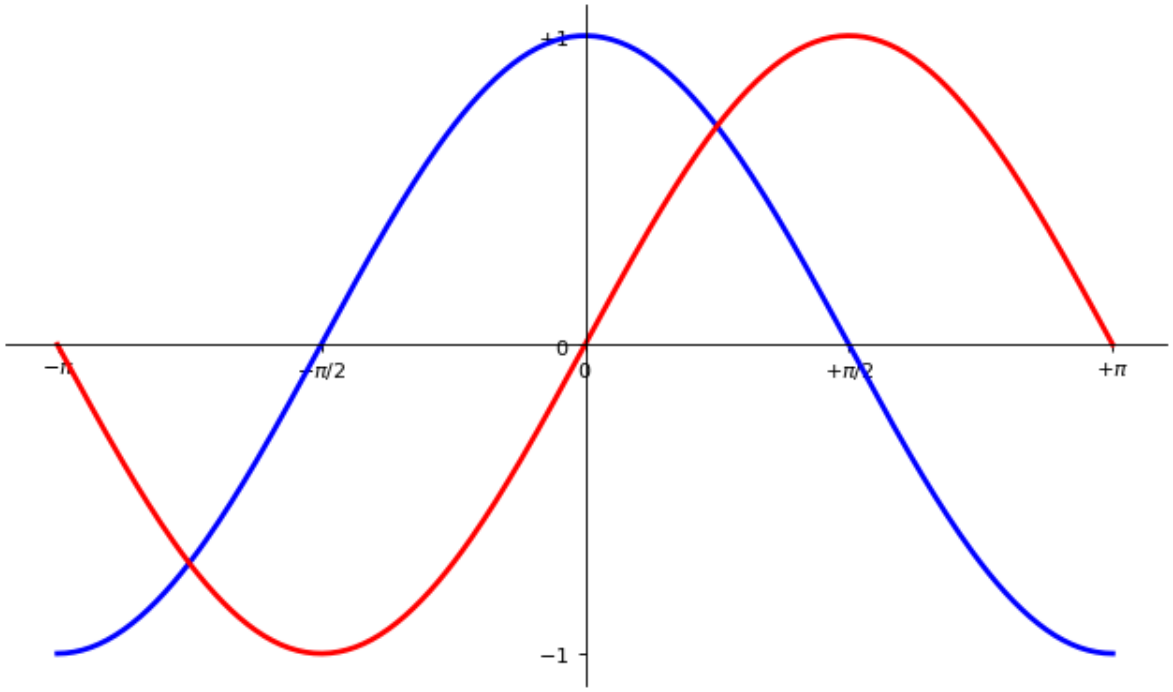


Gợi ý

```
# Set x ticks
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi],
        [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+\pi/2$', r'$+\pi$'])
# Set y ticks
plt.yticks([-1, 0, +1],
        [r'$-1$', r'$0$', r'$+1$'])
```

Giải thích

Bài 6. Điều chỉnh chương trình để có kết quả sau

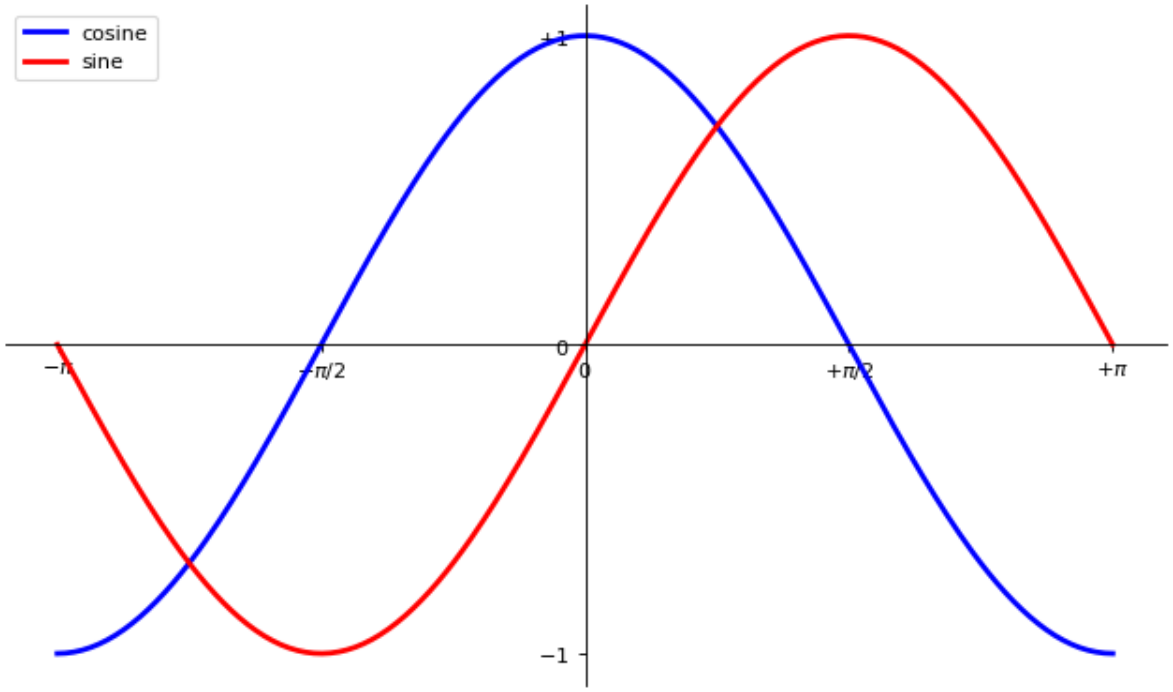


Gợi ý

```
ax = plt.gca()  # gca stands for 'get current axis'
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))
```

Giải thích

Bài 7. Điều chỉnh chương trình để có kết quả sau

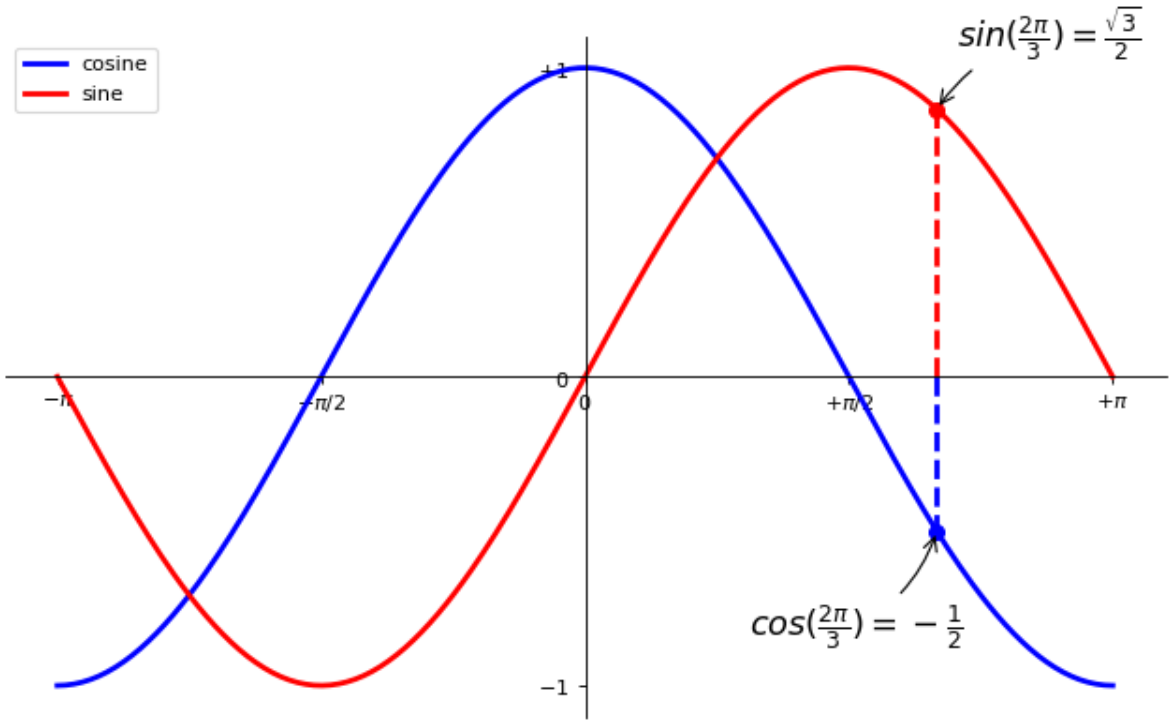


Gợi ý

```
# Plot cosine with a blue continuous line of width 1 (pixels)
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-", label="cosine")
# Plot sine with a green continuous line of width 1 (pixels)
plt.plot(X, S, color="red",  linewidth=2.5, linestyle="-", label="sine")
plt.legend(loc='upper left')
```

Giải thích?

Bài 8 . Điều chỉnh chương trình để có kết quả sau

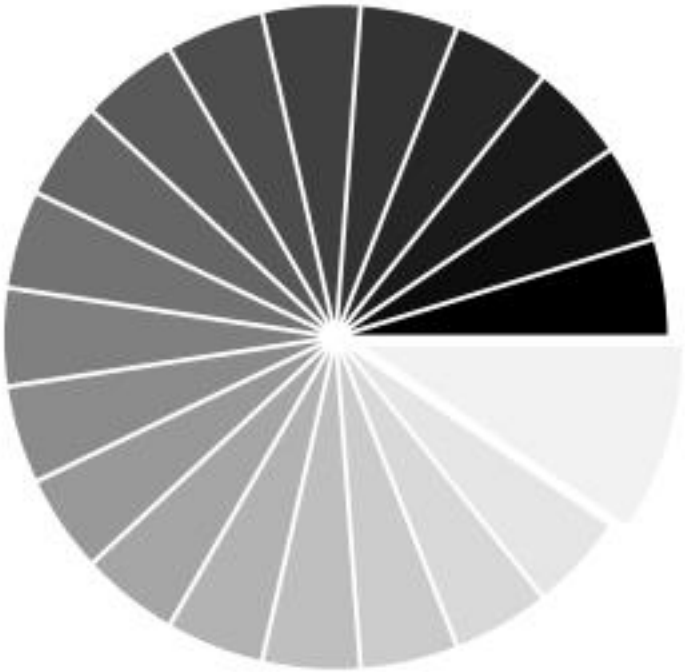


Gợi ý

```
t = 2 * np.pi / 3
plt.plot([t, t], [0, np.cos(t)], color='blue', linewidth=2.5, linestyle="--")
plt.scatter([t, ], [np.cos(t), ], 50, color='blue')
plt.annotate(r'$cos(\frac{2\pi}{3})=-\frac{1}{2}$',
        xy=(t, np.cos(t)), xycoords='data',
        xytext=(-90, -50), textcoords='offset points', fontsize=16,
        arrowprops=dict(arrowstyle="->", connectionstyle="arc3,rad=.2"))
plt.plot([t, t],[0, np.sin(t)], color='red', linewidth=2.5, linestyle="--")
plt.scatter([t, ],[np.sin(t), ], 50, color='red')
plt.annotate(r'$sin(\frac{2\pi}{3})=\frac{\sqrt{3}}{2}$',
        xy=(t, np.sin(t)), xycoords='data',
        xytext=(+10, +30), textcoords='offset points', fontsize=16,
        arrowprops=dict(arrowstyle="->", connectionstyle="arc3,rad=.2"))
```

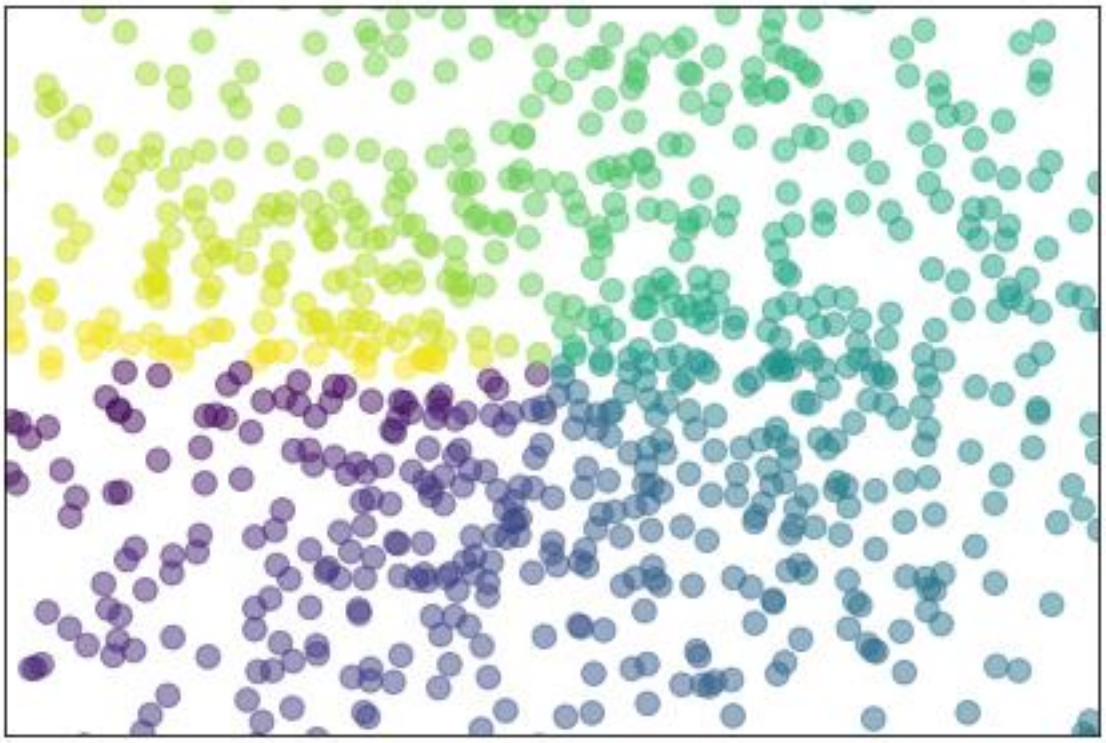
Giải thích

Bài 9.



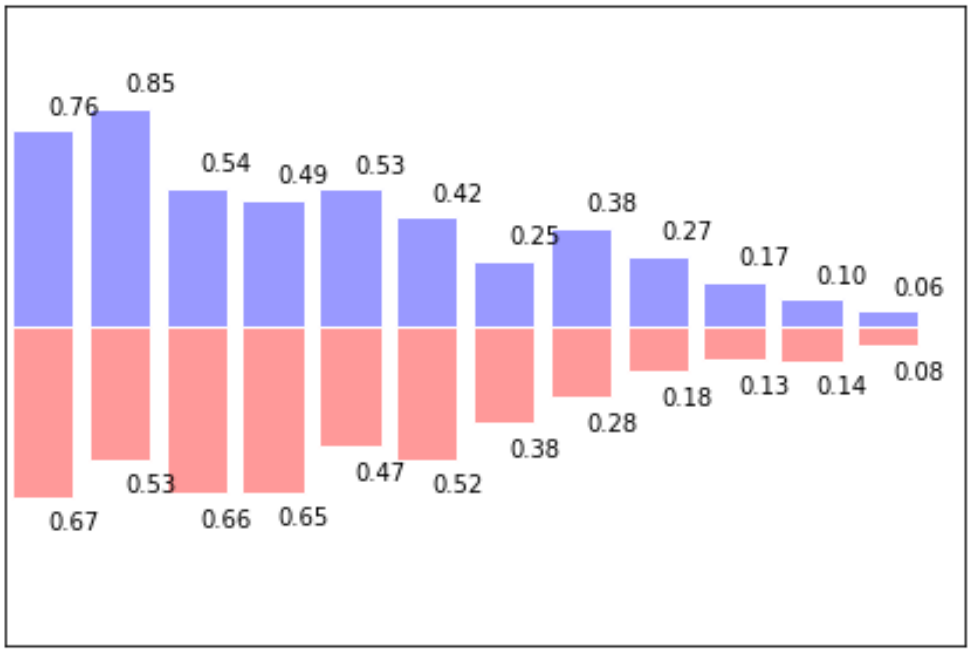
```
import numpy as np
import matplotlib.pyplot as plt
n = 20
Z = np.ones(n)
Z[-1] *= 2
plt.axes([0.025, 0.025, 0.95, 0.95])
plt.pie(Z, explode=Z*.05, colors = ['%f' % (i/float(n)) for i in range(n)])
plt.axis('equal')
plt.xticks(())
plt.yticks()
plt.show()
```

Bài 10



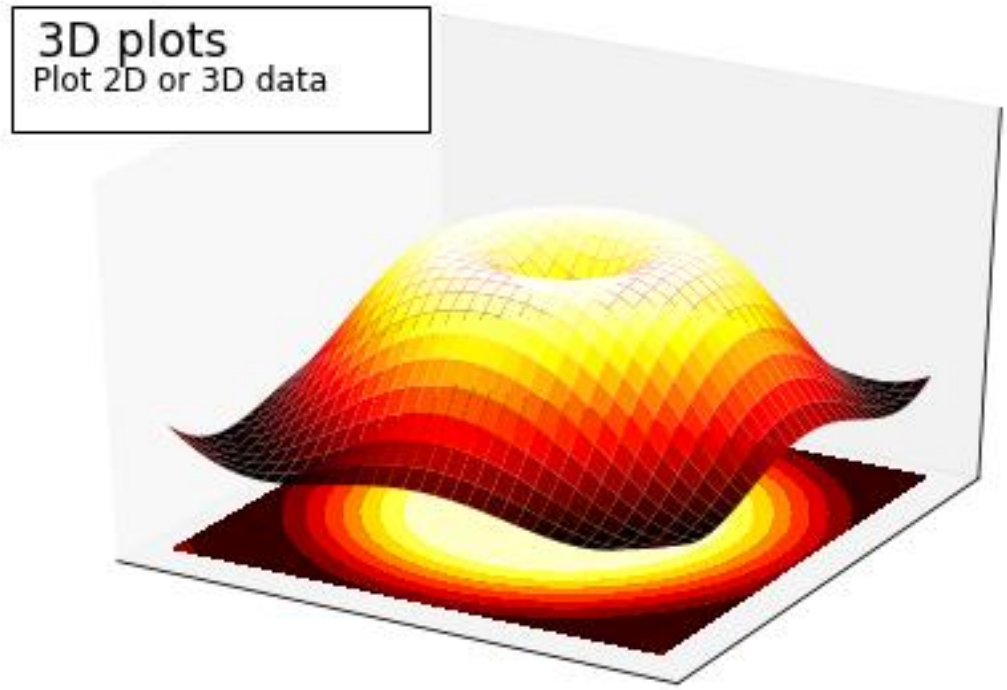
```
import numpy as np
import matplotlib.pyplot as plt
n = 1024
X = np.random.normal(0, 1, n)
Y = np.random.normal(0, 1, n)
T = np.arctan2(Y, X)
plt.axes([0.025, 0.025, 0.95, 0.95])
plt.scatter(X, Y, s=75, c=T, alpha=.5)
plt.xlim(-1.5, 1.5)
plt.xticks(())
plt.ylim(-1.5, 1.5)
plt.yticks(())
plt.show()
```

Bài 11




```
import numpy as np
import matplotlib.pyplot as plt
n = 12
X = np.arange(n)
Y1 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
Y2 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
plt.axes([0.025, 0.025, 0.95, 0.95])
plt.bar(X, +Y1, facecolor='#9999ff', edgecolor='white')
plt.bar(X, -Y2, facecolor='#ff9999', edgecolor='white')
for x, y in zip(X, Y1):
    plt.text(x + 0.4, y + 0.05, '%.2f' % y, ha='center', va= 'bottom')
for x, y in zip(X, Y2):
    plt.text(x + 0.4, -y - 0.05, '%.2f' % y, ha='center', va= 'top')
plt.xlim(-.5, n)
plt.xticks(())
plt.ylim(-1.25, 1.25)
plt.yticks(())
plt.show()
```

Bài 12




```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = Axes3D(fig)
X = np.arange(-4, 4, 0.25)
Y = np.arange(-4, 4, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X ** 2 + Y ** 2)
Z = np.sin(R)

ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=plt.cm.hot)
ax.contourf(X, Y, Z, zdir='z', offset=-2, cmap=plt.cm.hot)
ax.set_zlim(-2, 2)
```

```
plt.xticks(())
plt.yticks(())
ax.set_zticks(())

ax.text2D(0.05, .93, " 3D plots            \n",
        horizontalalignment='left',
        verticalalignment='top',
        size='xx-large',
        bbox=dict(facecolor='white', alpha=1.0),
        transform=plt.gca().transAxes)

ax.text2D(0.05, .87, " Plot 2D or 3D data",
        horizontalalignment='left',
        verticalalignment='top',
        size='large',
        transform=plt.gca().transAxes)

plt.show()
```