PYTHON
数据可视化-1

编译环境：

PyCharm 2019.3 (Community Edition)

Build #PC-193.5233.109, built on November 28, 2019

Runtime version: 11.0.4+10-b520.11 amd64

VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

Windows 10 10.0

GC: ParNew, ConcurrentMarkSweep

Memory: 1963M

Cores: 8

Registry:

Non-Bundled Plugins:

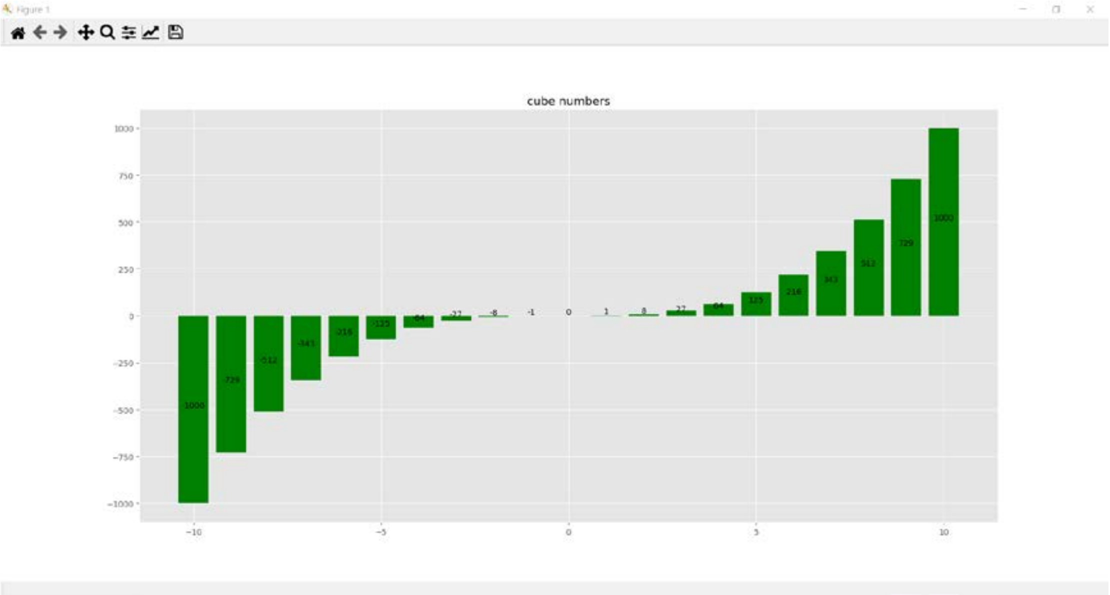
python 版本：3.7（Anaconda3）

作业 1：

q24.py:

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

x = np.array([-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
y = x * x * x
plt.style.use('ggplot')
ax = plt.subplot()
ax.set_title("cube numbers")
plt.bar(x, y, color='g')
for a, b in zip(x, y):
    plt.text(a, b / 2, '%d' % b, ha='center', va='bottom', fontsize=10)
plt.show()
```

运行结果：



作业 2：如果把 0-1000 之间的随机数改为 0-100 之间的，图比较清晰。

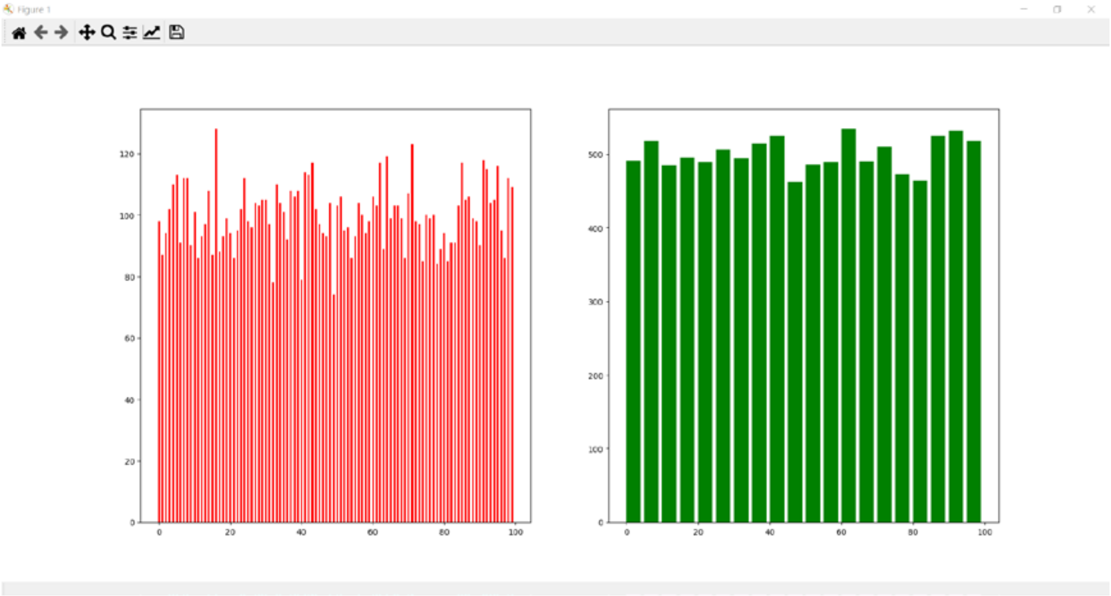
q25.py:

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


def count_elements(scores):
    scores_count = {}
    for i in scores:
        scores_count[int(i)] = scores_count.get(int(i), 0) + 1
    return scores_count


def process(data):
    result = {}
    for item in data.keys():
        key = item // 5 * 5
        result[key] = result.get(key, 0) + data[item]
    return result


#score = np.random.random(10000) * 100
```

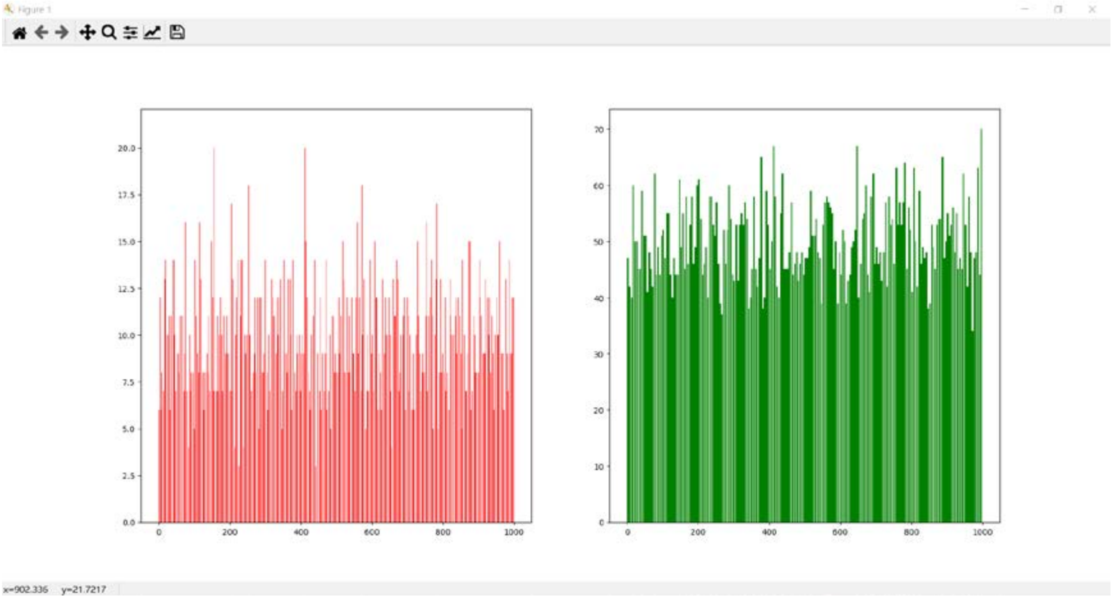
```
score = np.random.random(10000) * 1000
counted = count_elements(score)
sum_counted = process(counted)
fig, ax = plt.subplots(1, 2)
ax[0].bar(counted.keys(), counted.values(), 0.5, color='r')
ax[1].bar(sum_counted.keys(), sum_counted.values(), 4.0, color='g', align='edge')
plt.show()
```

运行结果：

0-100：



0-1000：



x=902.336 y=21.7217

作业 3：

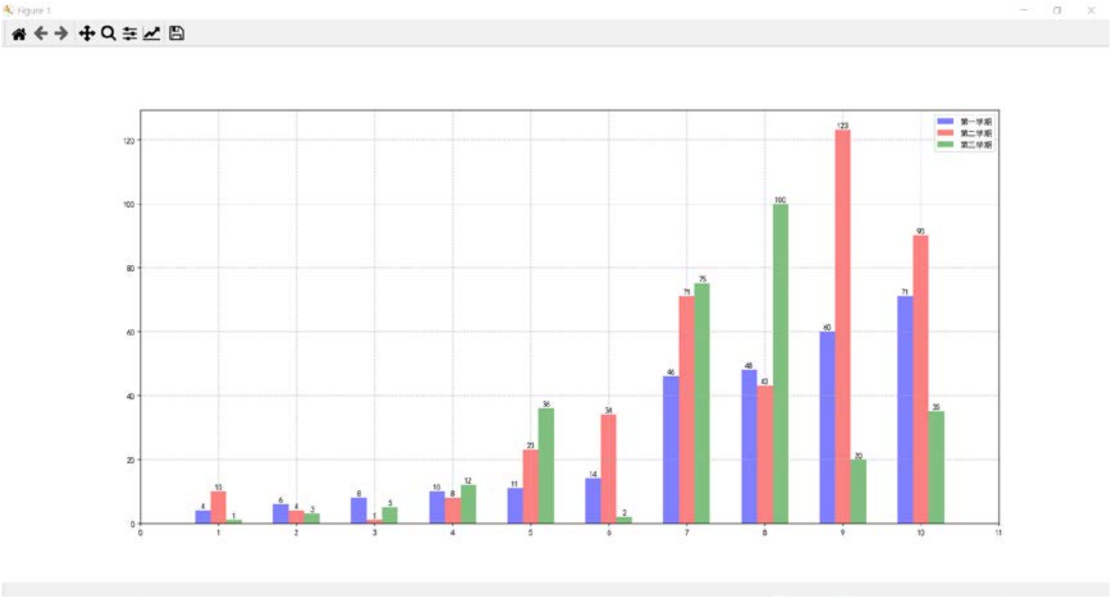
Data.csv:

```
4,10,1
6,4,3
8,1,5
10,8,12
11,23,36
14,34,2
46,71,75
48,43,100
60,123,20
71,90,35
```

q26.py:

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

fig, ax = plt.subplots()
plt.rcParams['font.sans-serif'] = ['SimHei']
semester1 = []
semester2 = []
semester3 = []
with open('data.csv', 'r') as csvfile:
    f_csv = csv.reader(csvfile)
    for row in f_csv:
        semester1.append(int(row[0]))
        semester2.append(int(row[1]))
        semester3.append(int(row[2]))
x = np.arange(1, 11)
plt.bar(x, semester1, 0.2, alpha=0.5, color='b')
plt.bar(x + 0.2, semester2, 0.2, alpha=0.5, color='r', align='center')
plt.bar(x + 0.4, semester3, 0.2, alpha=0.5, color='g')
for a, b in zip(x, semester1):
    plt.text(a, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=10)
for a, b in zip(x, semester2):
    plt.text(a + 0.2, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=10)
for a, b in zip(x, semester3):
    plt.text(a + 0.4, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=10)
plt.legend(["第一学期", "第二学期", "第三学期"], loc='upper right')
plt.grid(True, linestyle='--', alpha=0.8)
plt.xticks([0.2, 1.2, 2.2, 3.2, 4.2, 5.2, 6.2, 7.2, 8.2, 9.2, 10.2, 11.2],[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
plt.show()
```

运行结果：

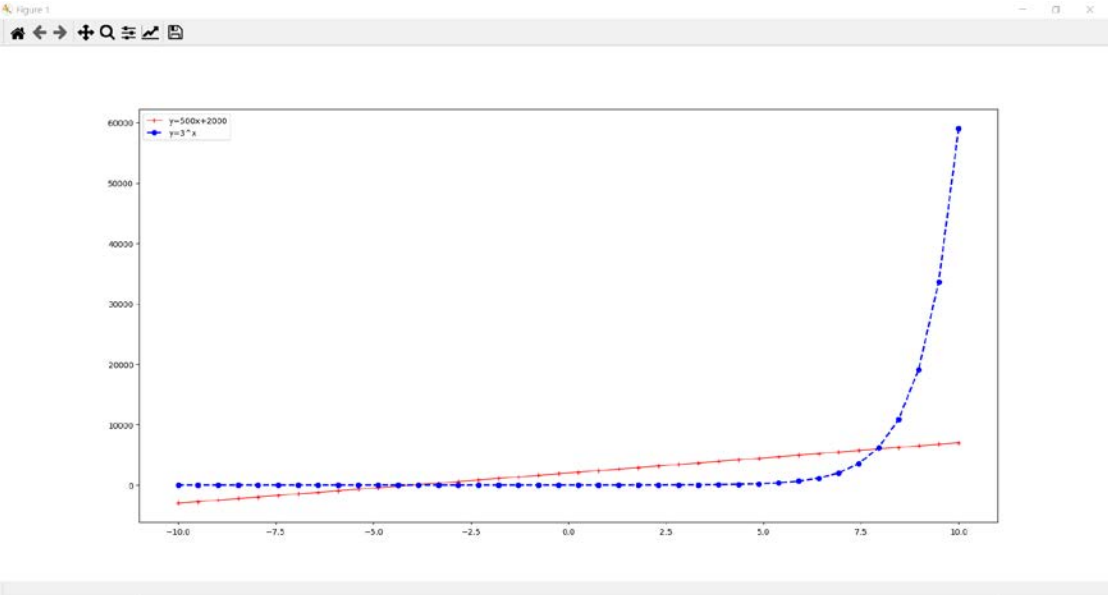


作业 4：

q27.py:

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

fig, ax = plt.subplots()
x = np.linspace(-10, 10, 40)
y1 = 500 * x + 2000
y2 = 3 ** x
ax.plot(x, y1, 'r+', color='red', linewidth=1.0, linestyle='-', label='line1')
ax.plot(x, y2, 'bo', color='blue', linewidth=2.0, linestyle='--', label='line2')
plt.xlim(-11, 11)
ax.legend(["y=500x+2000", "y=3^x"], loc='upper left')
plt.show()
```

运行结果：

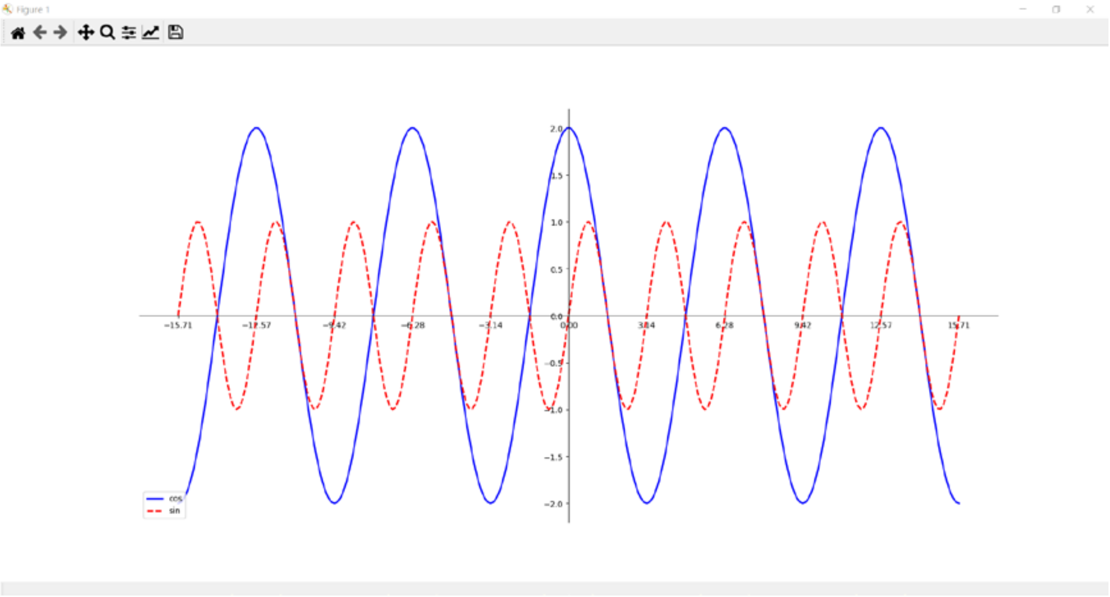


作业 5：

q28.py:

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

fig, ax = plt.subplots()
x = np.linspace(-5 * np.pi, 5 * np.pi, 256)
cos, sin = 2 * np.cos(x), np.sin(2 * x)
ax.set_xticks(
    [-5 * np.pi, -4 * np.pi, -3 * np.pi, -2 * np.pi, -1 * np.pi, 0, np.pi, 2 * np.pi, 3 * np.pi, 4 * np.pi, 5 * np.pi])
plt.plot(x, cos, color='blue', linewidth=2.0, linestyle='-', label='cos')
plt.plot(x, sin, color='red', linewidth=2.0, linestyle='--', label='sin')
ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.spines["bottom"].set_position(('data', 0))
ax.xaxis.set_ticks_position('bottom')
ax.spines["left"].set_position(('data', 0))
ax.yaxis.set_ticks_position('left')
ax.legend(loc='lower left')
plt.show()
```

运行结果：



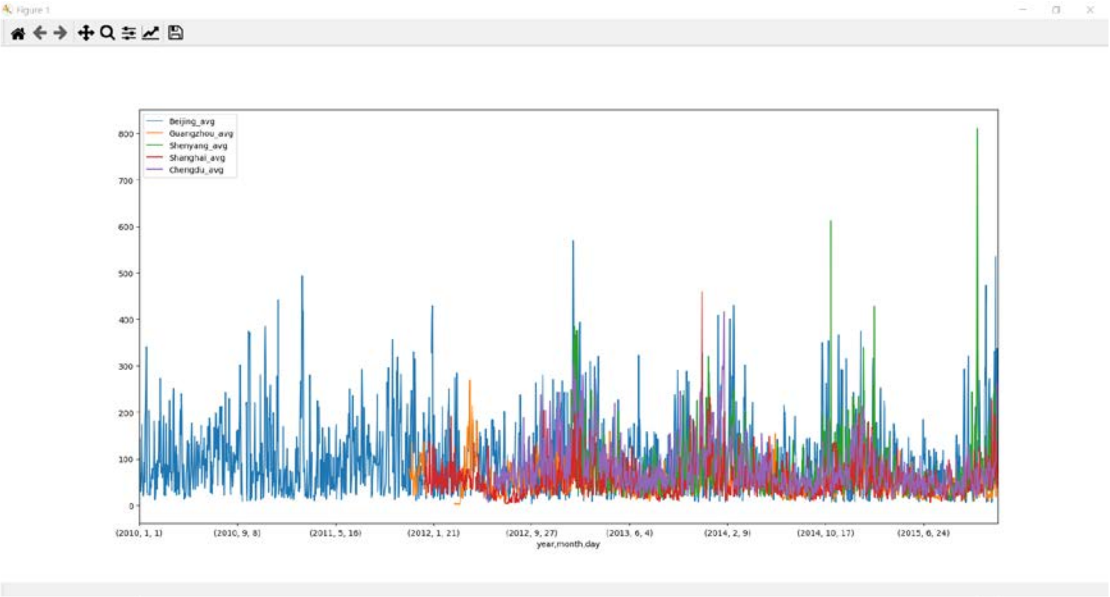
作业6：

q29.py:

```
import pandas as pd
import matplotlib.pyplot as plt

fileNameStr1 = 'BeijingPM20100101_20151231.csv'
df1 = pd.read_csv(fileNameStr1, encoding='utf-8', usecols=[1, 2, 3, 6, 7, 8, 9])
fileNameStr2 = 'GuangzhouPM20100101_20151231.csv'
df2 = pd.read_csv(fileNameStr2, encoding='utf-8', usecols=[1, 2, 3, 6, 7, 8])
fileNameStr3 = 'ShenyangPM20100101_20151231.csv'
df3 = pd.read_csv(fileNameStr3, encoding='utf-8', usecols=[1, 2, 3, 6, 7, 8])
fileNameStr4 = 'ShanghaiPM20100101_20151231.csv'
df4 = pd.read_csv(fileNameStr4, encoding='utf-8', usecols=[1, 2, 3, 6, 7, 8])
fileNameStr5 = 'ChengduPM20100101_20151231.csv'
df5 = pd.read_csv(fileNameStr5, encoding='utf-8', usecols=[1, 2, 3, 6, 7, 8])

df1['sum'] = df1[['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan', 'PM_US Post']].sum(axis=1)
df1['count'] = df1[['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan', 'PM_US Post']].count(axis=1)
df1['Beijing_avg'] = round(df1['sum'] / df1['count'], 2)
df_mean1 = df1.groupby(["year", "month", "day"])[['Beijing_avg']].mean()
df_mean1 = pd.DataFrame(df_mean1)

df2['sum'] = df2[['PM_City Station', 'PM_5th Middle School', 'PM_US Post']].sum(axis=1)
df2['count'] = df2[['PM_City Station', 'PM_5th Middle School', 'PM_US Post']].count(axis=1)
df2['Guangzhou_avg'] = round(df2['sum'] / df2['count'], 2)
```

```
df_mean2 = df2.groupby(["year", "month", "day"])[['Guangzhou_avg']].mean()
df_mean2 = pd.DataFrame(df_mean2)

df3['sum'] = df3[['PM_Taiyuanjie', 'PM_US Post', 'PM_Xiaoheyan']].sum(axis=1)
df3['count'] = df3[['PM_Taiyuanjie', 'PM_US Post', 'PM_Xiaoheyan']].count(axis=1)
df3['Shenyang_avg'] = round(df3['sum'] / df3['count'], 2)
df_mean3 = df3.groupby(["year", "month", "day"])[['Shenyang_avg']].mean()
df_mean3 = pd.DataFrame(df_mean3)

df4['sum'] = df4[['PM_Jingan', 'PM_US Post', 'PM_Xuhui']].sum(axis=1)
df4['count'] = df4[['PM_Jingan', 'PM_US Post', 'PM_Xuhui']].count(axis=1)
df4['Shanghai_avg'] = round(df4['sum'] / df4['count'], 2)
df_mean4 = df4.groupby(["year", "month", "day"])[['Shanghai_avg']].mean()
df_mean4 = pd.DataFrame(df_mean4)

df5['sum'] = df5[['PM_Caotangsi', 'PM_Shahepu', 'PM_US Post']].sum(axis=1)
df5['count'] = df5[['PM_Caotangsi', 'PM_Shahepu', 'PM_US Post']].count(axis=1)
df5['Chengdu_avg'] = round(df5['sum'] / df5['count'], 2)
df_mean5 = df5.groupby(["year", "month", "day"])[['Chengdu_avg']].mean()
df_mean5 = pd.DataFrame(df_mean5)

df = pd.concat([df_mean1, df_mean2, df_mean3, df_mean4, df_mean5], axis=1, join='inner')
df.plot()
plt.show()
```

运行结果：



作业 7：

q30.py:

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

fig, ax = plt.subplots()
iris = pd.read_csv('iris.csv')
color = ['r', 'y', 'b']
Species = iris.Species.unique()

plt.subplot(4, 4, 1)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Petal.Width'],
                iris.loc[iris.Species == Species[i], 'Petal.Width'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Petal.Width')
plt.ylabel('Petal.Width')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 2)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Petal.Length'],
                iris.loc[iris.Species == Species[i], 'Petal.Width'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Petal.Width')
plt.ylabel('Petal.Width')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 3)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Sepal.width'],
                iris.loc[iris.Species == Species[i], 'Petal.Width'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Sepal.width')
plt.ylabel('Petal.Width')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 4)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Sepal.length'],
                        iris.loc[iris.Species == Species[i], 'Petal.Width'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Sepal.length')
plt.ylabel('Petal.Width')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 5)
for i in range(len(Species)):
      plt.scatter(iris.loc[iris.Species == Species[i], 'Petal.Width'],
                        iris.loc[iris.Species == Species[i], 'Petal.Length'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Petal.Width')
plt.ylabel('Petal.Length')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 6)
for i in range(len(Species)):
      plt.scatter(iris.loc[iris.Species == Species[i], 'Petal.Length'],
                        iris.loc[iris.Species == Species[i], 'Petal.Length'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Petal.Length')
plt.ylabel('Petal.Length')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 7)
for i in range(len(Species)):
      plt.scatter(iris.loc[iris.Species == Species[i], 'Sepal.width'],
                        iris.loc[iris.Species == Species[i], 'Petal.Length'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Sepal.width')
plt.ylabel('Petal.Length')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 8)
for i in range(len(Species)):
      plt.scatter(iris.loc[iris.Species == Species[i], 'Sepal.length'],
                        iris.loc[iris.Species == Species[i], 'Petal.Length'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Sepal.length')
plt.ylabel('Petal.Length')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 9)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Petal.Width'],
                iris.loc[iris.Species == Species[i], 'Sepal.width'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Petal.Width')
plt.ylabel('Sepal.width')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 10)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Petal.Length'],
                iris.loc[iris.Species == Species[i], 'Sepal.width'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Petal.Length')
plt.ylabel('Sepal.width')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 11)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Sepal.width'],
                iris.loc[iris.Species == Species[i], 'Sepal.width'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Sepal.width')
plt.ylabel('Sepal.width')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 12)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Sepal.length'],
                iris.loc[iris.Species == Species[i], 'Sepal.width'], s=35, c=color[i],
label=Species[i])
plt.xlabel('Sepal.length')
plt.ylabel('Sepal.width')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 13)
```

```
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Petal.Width'],
                iris.loc[iris.Species == Species[i], 'Sepal.length'], s=35, c=color[i], label=Species[i])
plt.xlabel('Petal.Width')
plt.ylabel('Sepal.length')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 14)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Petal.Length'],
                iris.loc[iris.Species == Species[i], 'Sepal.length'], s=35, c=color[i], label=Species[i])
plt.xlabel('Petal.Length')
plt.ylabel('Sepal.length')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 15)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Sepal.width'],
                iris.loc[iris.Species == Species[i], 'Sepal.length'], s=35, c=color[i], label=Species[i])
plt.xlabel('Sepal.width')
plt.ylabel('Sepal.length')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.subplot(4, 4, 16)
for i in range(len(Species)):
    plt.scatter(iris.loc[iris.Species == Species[i], 'Sepal.length'],
                iris.loc[iris.Species == Species[i], 'Sepal.length'], s=35, c=color[i], label=Species[i])
plt.xlabel('Sepal.length')
plt.ylabel('Sepal.length')
plt.grid(True, linestyle='--', alpha=0.8)
plt.legend(loc='lower right')

plt.show()
```

运行结果：

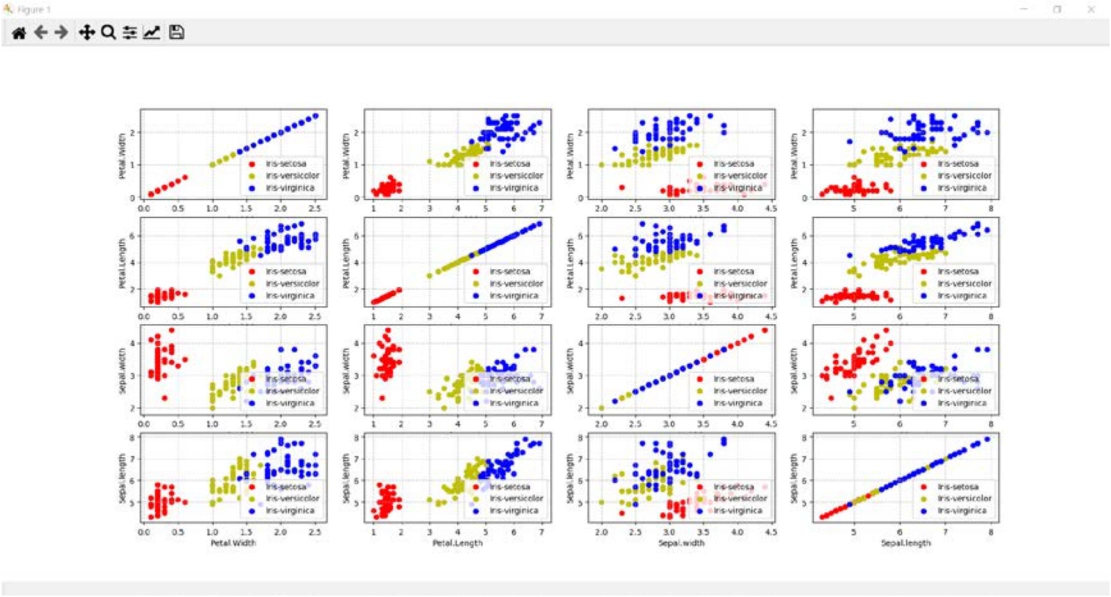


Figure 1
Petal.Width
Petal.Length
Sepal.width
Sepal.length
Iris-setosa
Iris-versicolor
Iris-virginica
0.0 0.5 1.0 1.5 2.0 2.5
1 2 3 4 5 6 7
2.0 2.5 3.0 3.5 4.0 4.5
5 6 7 8
Petal Width
Petal.Length
Sepal.width
Sepal.length