数据可视化

Table of Contents

官方使用教程是非常重要的学习来源。

0.1 图形的构成

图形的构成可以参考 Matplotlib 官方网站上Anatomy of a figure的说明：

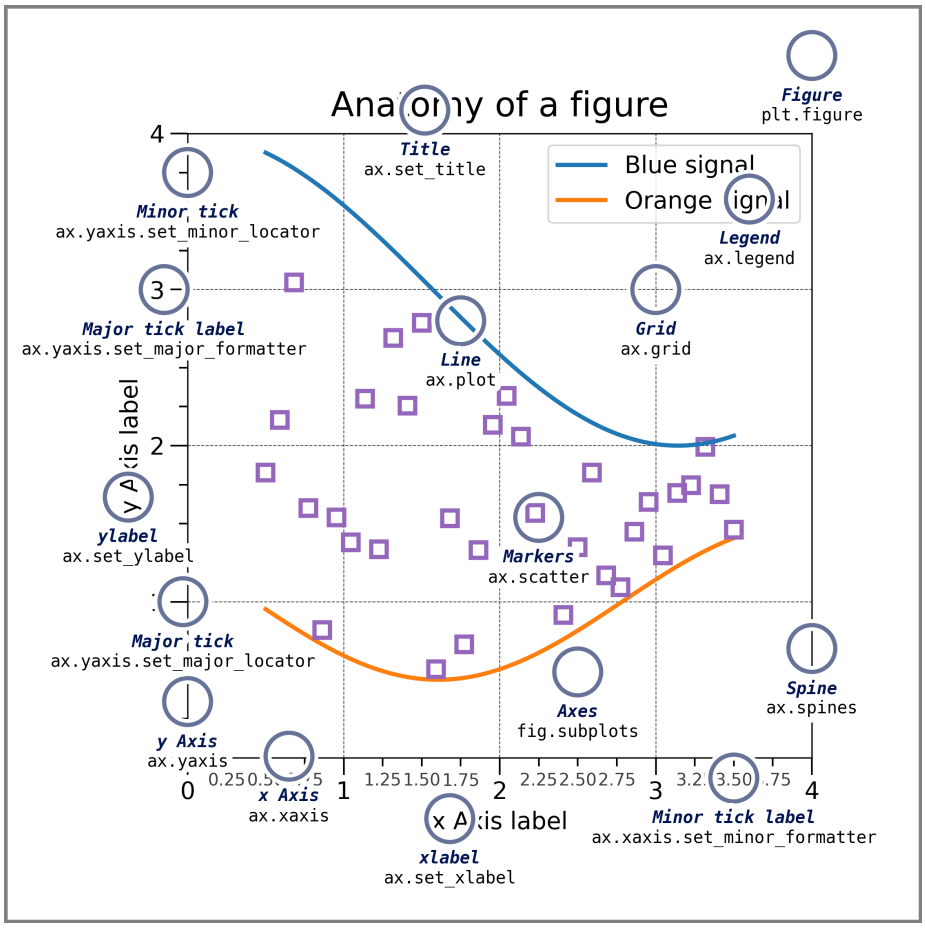


Figure 1: 图形解剖图

我们用一个例子来说明绘图的过程。下面的全球电影票房数据来自维基百科全球最高電影票房收入列表

```
import pandas as pd
import re
data = pd.read_excel("datasets/highest_gross_films.xlsx")
data['全球票房'] = data['全球票房'].apply(lambda ser: pd.to_numeric(re.sub(r'\D',''
df = data[:10]
df
```

	排名	峰值	影片名称	全球票房	年份
0	1	1	阿凡达	2923706026	2009
1	2	1	复仇者联盟：终局之战	2797501328	2019
2	3	3	阿凡达：水之道	2320250281	2022
3	4	1	泰坦尼克号	2257844554	1997
4	5	5	哪吒 2	2217080000	2025
5	6	3	星球大战：原力觉醒	2068223624	2015
6	7	4	复仇者联盟：无限战争	2048359754	2018
7	8	6	蜘蛛侠：英雄无归	1922598800	2021
8	9	8	头脑特工队 2	1698863816	2024
9	10	3	侏罗纪世界	1671537444	2015

0.2 绘图一般步骤

图形的种类非常多，应用 Python 绘图时可以大致分为几个步骤。

0.2.1 载入必要的库

除了基本的绘图工具Matplotlib外，Seaborn库也经常使用，在应用之前均应载入。

```
import matplotlib.pyplot as plt
import seaborn as sns
```

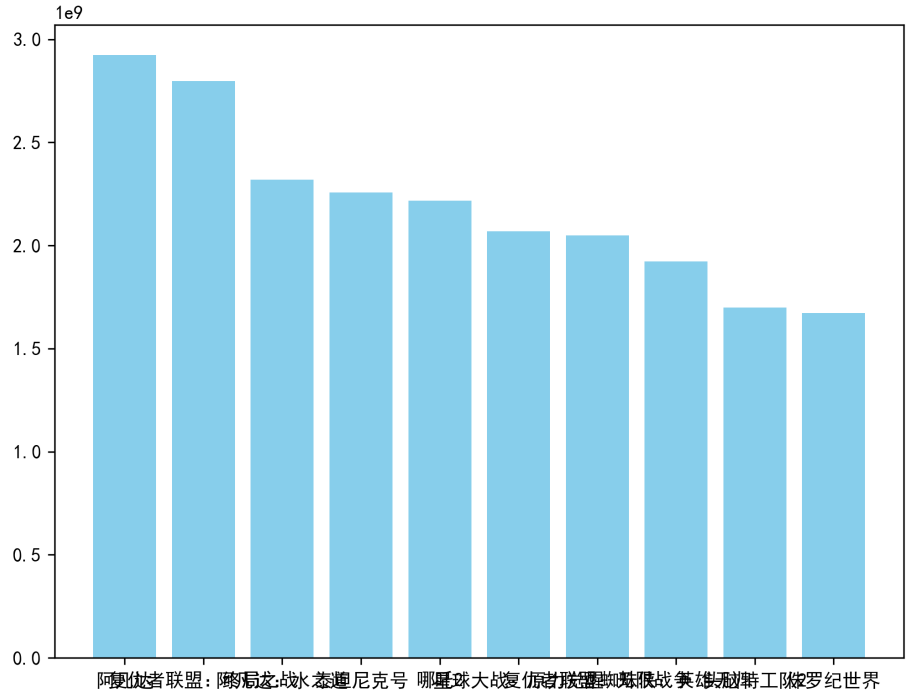
另外，Matplotlib 默认情况下不支持中文字符。如果你直接在图表标题、坐标轴标签或图例中使用中文，很可能会看到方框乱码或者问号。

```
plt.rcParams['font.sans-serif'] = ['SimHei', 'Heiti TC', 'WenQuanYi Zen Hei', 'Arial']

plt.rcParams["axes.unicode_minus"] = False
```

现在有了图形轴（Axes）的实例，就可以在上面绘制图形了。例如绘制一幅柱形图，使用.bar() 方法：

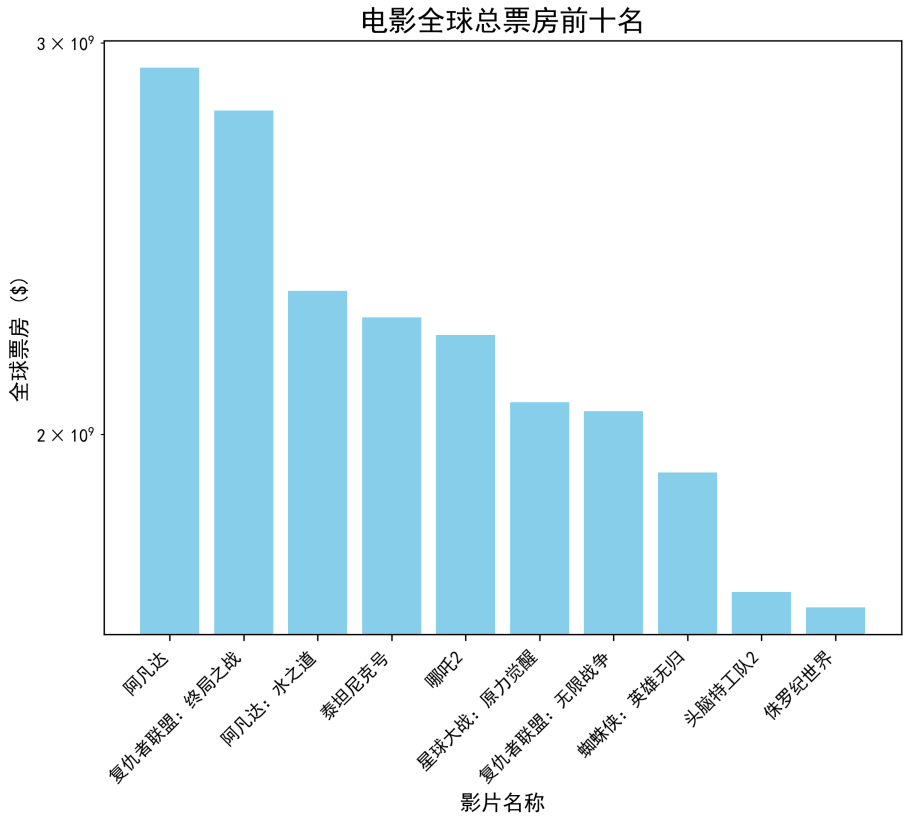
```
fig, ax = plt.subplots(figsize=(8, 6))
ax.bar(x=df['影片名称'], height=df['全球票房'], color='skyblue')
plt.show()
```



显然，图形还有改善的空间。比如横轴的标签，即电影名字挤在一起看不清楚，也可以设置纵轴标签、图形标题等：

```
fig, ax = plt.subplots(figsize=(8, 6))
ax.bar(x=df['影片名称'], height=df['全球票房'], color='skyblue')
```

```
ax.set_yscale("log")
ax.set_title('电影全球总票房前十名', fontsize=16, fontweight='bold')
ax.set_xlabel('影片名称', fontsize=12)
ax.set_ylabel('全球票房 ($)', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.show()
```

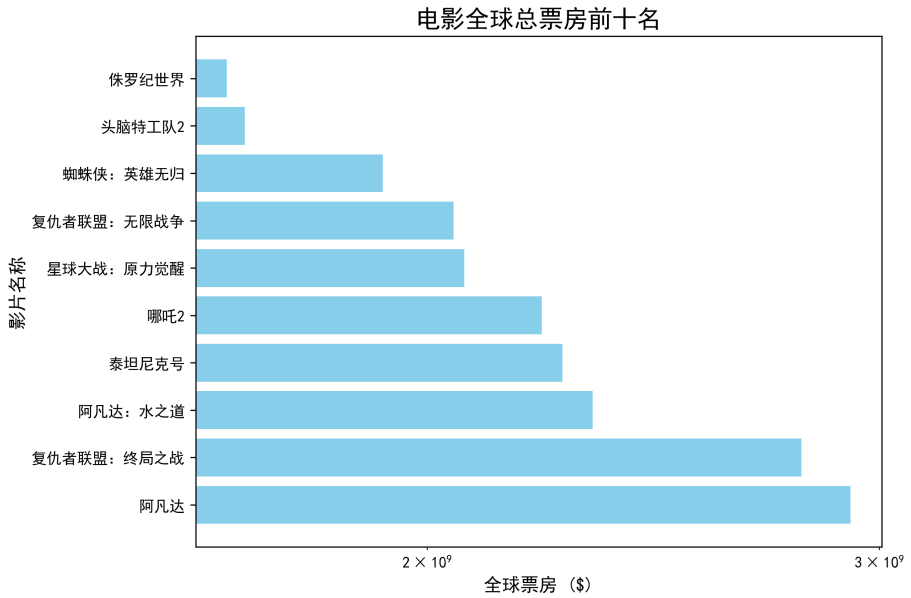


横向柱形图

```
fig, ax = plt.subplots(figsize=(8, 6))
ax.barh(y = df['影片名称'], width=df['全球票房'], color='skyblue')
ax.set_xscale("log")
ax.set_title('电影全球总票房前十名', fontsize=16, fontweight='bold')
ax.set_ylabel('影片名称', fontsize=12)
```

```
ax.set_xlabel('全球票房 ($)', fontsize=12)

plt.show()
```



0.3 常用的图形

0.3.1 直方图

直方图可能是使用频率最高的图形。下面的例子，自雅虎财经网站下载几支股票的月度（后复权调整）收盘价数据，然后使用 `df.pct_change()` 函数计算了简单收益率；

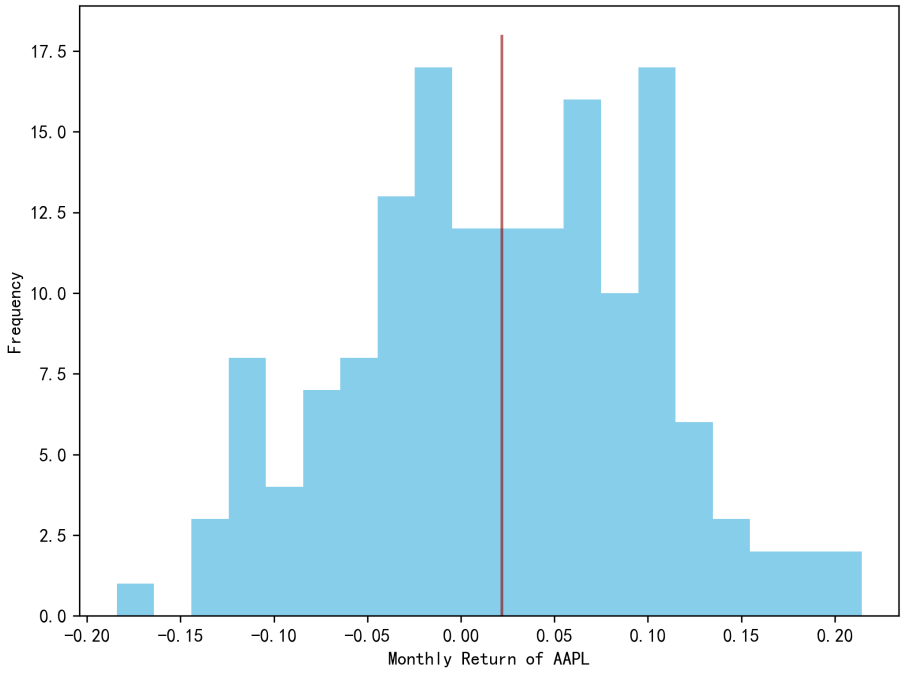
```
import yfinance as yf
import seaborn as sns

stocks_list = ['AAPL','BA','MGM','AMZN','IBM','TSLA','GOOG','^GSPC']
start_date = "2012-01-01"
end_date = "2024-12-31"
```

```
stocks_price_us = yf.download(tickers=stocks_list,
         start=start_date,
         end=end_date,
         interval="1mo",
         auto_adjust=True,
         progress=False)['Close']
returns = stocks_price_us.pct_change()
```

例如，绘制苹果公司股票收益率的直方图：

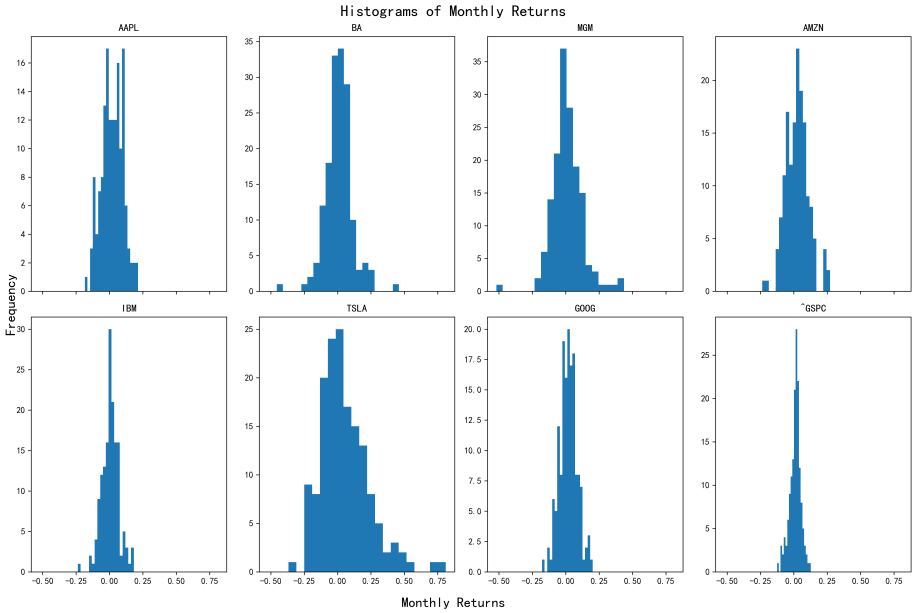
```
fig, ax = plt.subplots(figsize=(8, 6))
ax.hist(x=returns['AAPL'], bins=20,
        color="skyblue")
ax.vlines(x=returns['AAPL'].mean(),
        ymin=0, ymax=18, colors="darkred",
        alpha=0.6)
ax.set_xlabel("Monthly Return of AAPL")
ax.set_ylabel("Frequency")
plt.show()
```



如果纵轴希望表示为概率密度，加上参数 `density=True`。

下载的数据包含 8 家 7 家企业以及标准普尔 500 指数（GSPC），下面将收益率为子图绘制直方图。为了可比，横轴使用了 `sharex=True` 参数：

```
fig, axes = plt.subplots(nrows=2,
                         ncols=4,
                         figsize=(15, 10),
                         sharex=True)
axes = axes.flatten()
for i, stock in enumerate(stocks_list):
    axes[i].hist(returns[stock], bins=20)
    axes[i].set_title(f'{stock}', fontsize=12)
fig.supxlabel("Monthly Returns", fontsize=16)
fig.supylabel("Frequency", fontsize=16)
fig.suptitle("Histograms of Monthly Returns", fontsize=18)
plt.tight_layout()
plt.show()
```



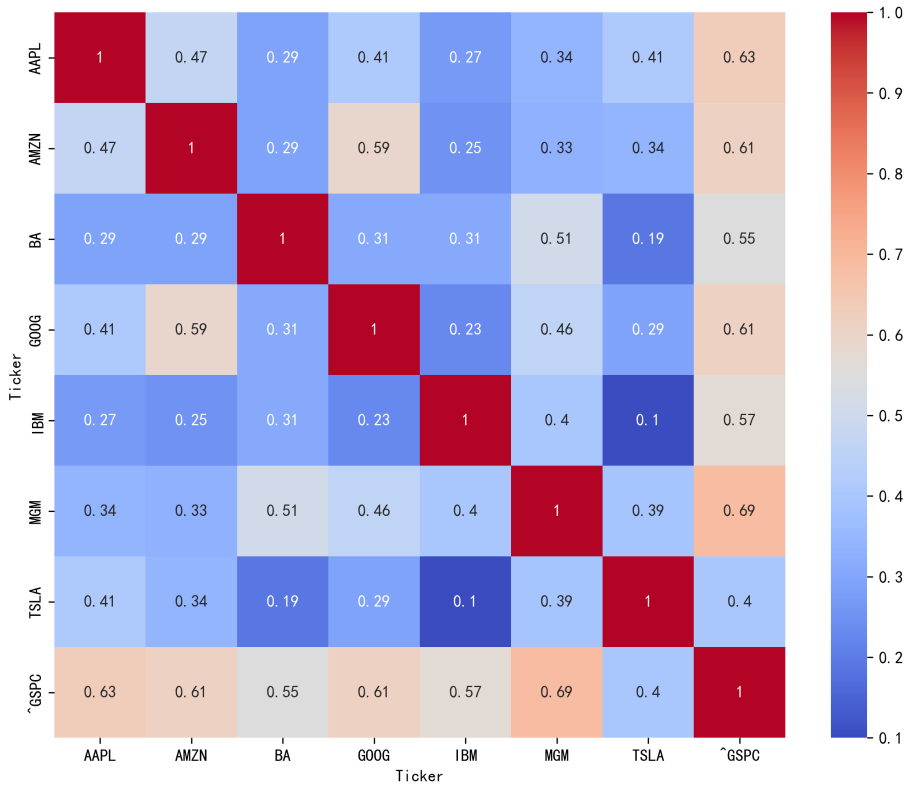
0.3.2 核密度图

设 $(x_1, x_2, \cdots, x_n)$ 为从单变量分布中抽取的独立同分布样本，给定点 x 有未知的概率密度 f ，函数 f 核密度估计量是：

0.3.3 热图

```
corr = returns.corr().round(2)

fig, ax = plt.subplots(figsize=(10, 8))
sns.heatmap(corr, annot=True,
            ax=ax,
            cmap='coolwarm')
plt.show()
```



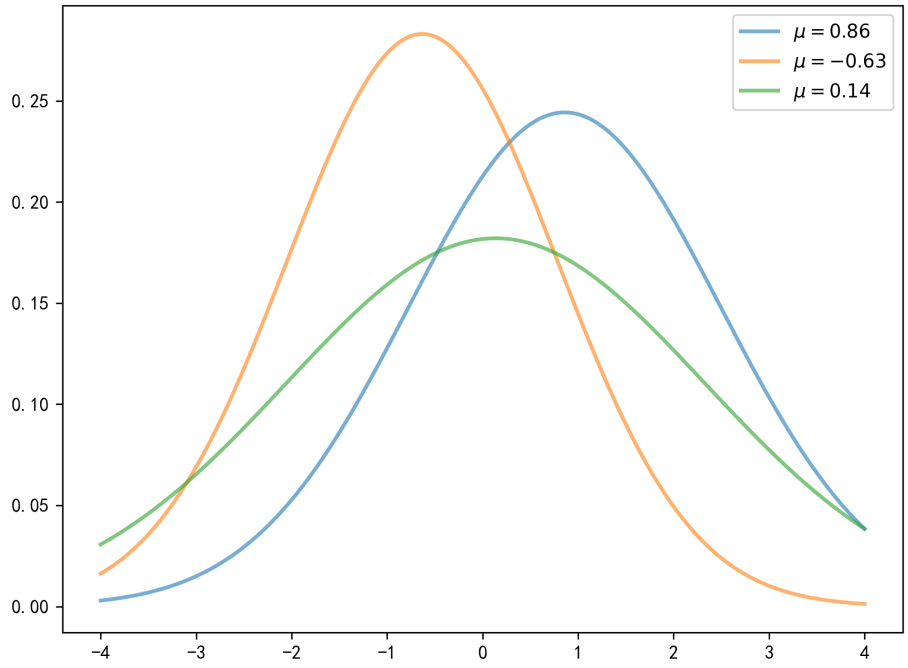
0.4 多图和子图

0.5 多图

```
# 正态分布
import numpy as np
from scipy.stats import norm
np.random.seed(12345)

fig, ax = plt.subplots(figsize=(8, 6))
x = np.linspace(-4, 4, 500)
for i in range(3):
```

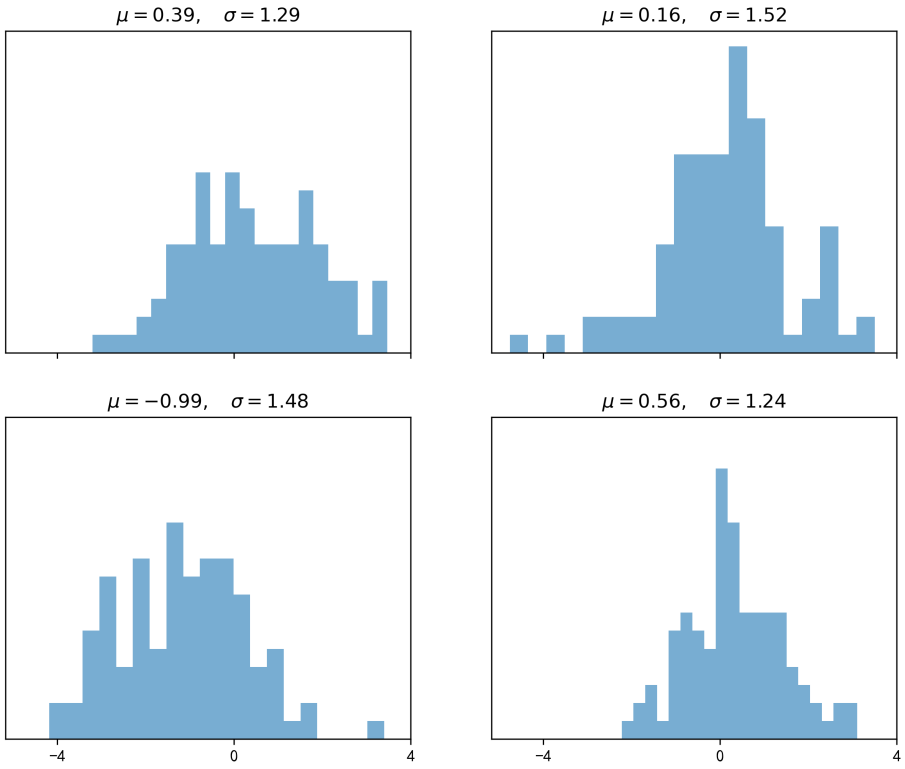
```
    mu, std = np.random.uniform(-1,1), np.random.uniform(1, 3)
    y = norm.pdf(x, loc = mu, scale = std)
    current_label = rf"$\mu = {mu:.2f}$"
    ax.plot(x, y, linewidth = 2, alpha = 0.6, label = current_label)
ax.legend()
plt.show()
```



0.5.1 子图

```
np.random.seed(123)
fig, axes = plt.subplots(nrows=2, ncols=2,figsize=(10,8),sharex=True,sharey=True)
for i in range(2):
    for j in range(2):
        m, s = np.random.uniform(-1, 1), np.random.uniform(1, 2)
        x = np.random.normal(m,s,100)
        axes[i, j].hist(x, alpha = 0.6, bins=20)
```

```
        title = rf"$\mu = {m:.2f},\quad \sigma = {s:.2f}$"
        axes[i, j].set(title = title, xticks = [-4, 0, 4],yticks = [])
plt.show()
```



0.5.2 图形风格

```
plt.style.available
```

```
['Solarize_Light2',
 '_classic_test_patch',
 '_mpl-gallery',
 '_mpl-gallery-nogrid',
 'bmh',
 'classic',
```

```
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'petroff10',
 'seaborn-v0_8',
 'seaborn-v0_8-bright',
 'seaborn-v0_8-colorblind',
 'seaborn-v0_8-dark',
 'seaborn-v0_8-dark-palette',
 'seaborn-v0_8-darkgrid',
 'seaborn-v0_8-deep',
 'seaborn-v0_8-muted',
 'seaborn-v0_8-notebook',
 'seaborn-v0_8-paper',
 'seaborn-v0_8-pastel',
 'seaborn-v0_8-poster',
 'seaborn-v0_8-talk',
 'seaborn-v0_8-ticks',
 'seaborn-v0_8-white',
 'seaborn-v0_8-whitegrid',
 'tableau-colorblind10']
```

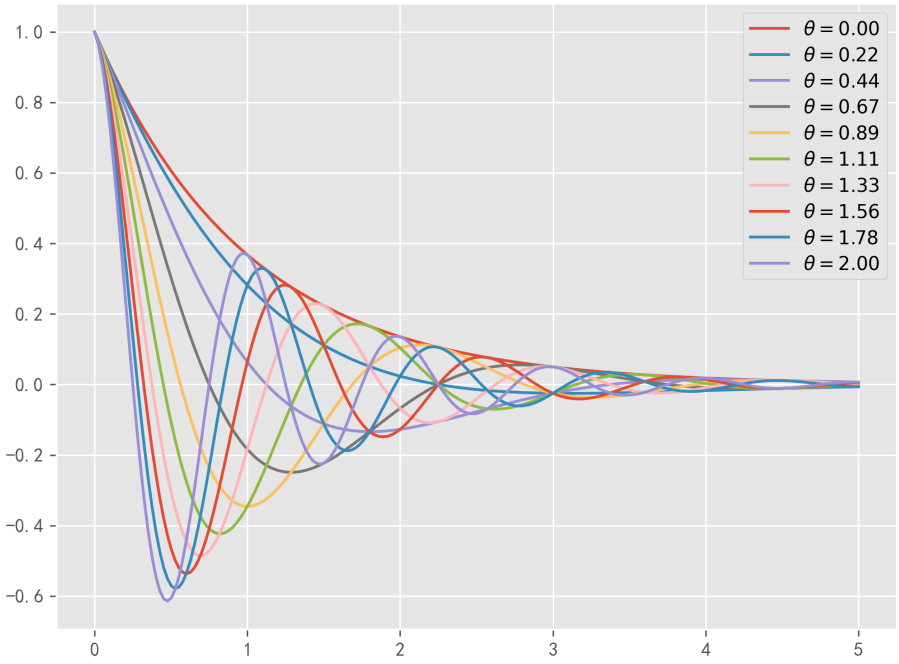
```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use("ggplot")

def f(x, theta):
    return np.cos(np.pi * theta * x ) * np.exp(- x)

 _vals = np.linspace(0, 2, 10)
x = np.linspace(0, 5, 200)
fig, ax = plt.subplots(figsize=(8, 6))
```

```
for theta in  _vals:
    ax.plot(x, f(x, theta), label = rf"$\theta = {theta:.2f}$")

ax.legend()
plt.show()
```



0.6 应用：收益率的几个典型事实

这部分内容，

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import yfinance as yf
import scipy.stats as stats
```

```
import statsmodels.api as sm
```

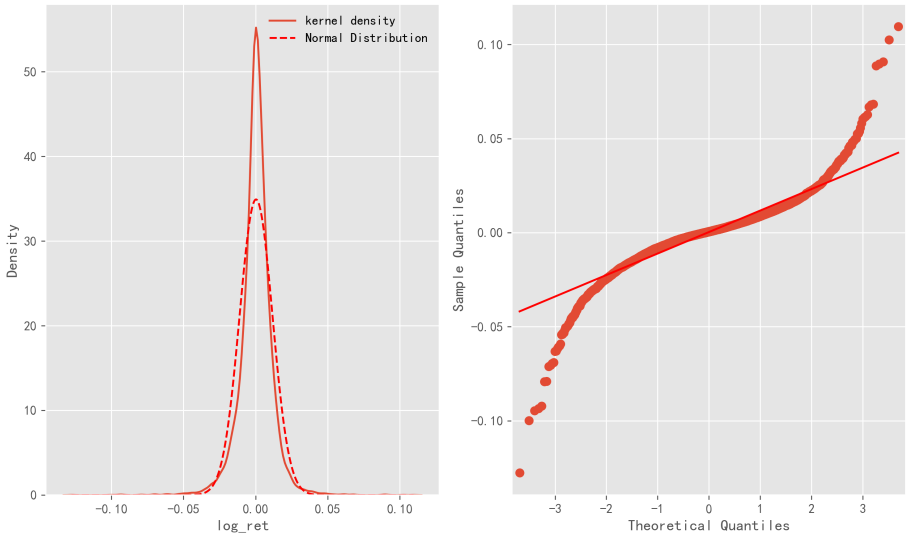
然后下载或读取数据：

```
# df = yf.download(['^GSPC', '^VIX'],
#                  start='1990-01-01',
#                  auto_adjust=True,
#                  progress=False)['Close']
# df.to_csv('datasets/sp500_vix.csv')

df = pd.read_csv('datasets/sp500_vix.csv',
                 header=0,
                 index_col=0,
                 parse_dates=True)
df.columns = ["SP500", "VIX"]
df['log_ret'] = np.log(df['SP500']/df['SP500'].shift(1))
df.dropna(inplace=True)
```

0.6.1 厚尾

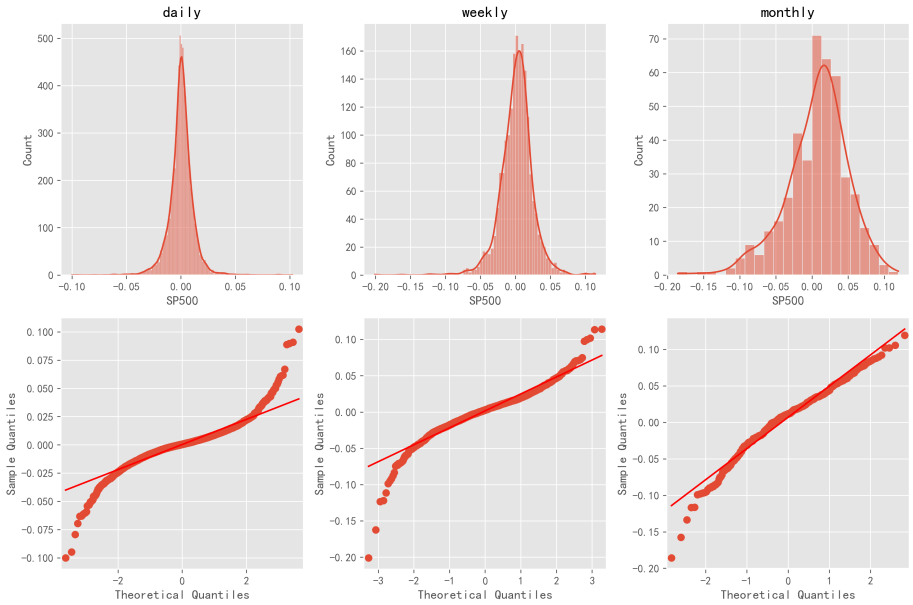
```
fig,ax =plt.subplots(1,2,figsize=(10,6))
sns.kdeplot(df['log_ret'], fill=False,label='kernel density',ax=ax[0])
mu, sigma = stats.norm.fit(df['log_ret'])
x = np.linspace(df['log_ret'].min(), df['log_ret'].max(), 1000)
y = stats.norm.pdf(x, mu, sigma)
ax[0].plot(x, y, color='r', label='Normal Distribution',linestyle='--')
ax[0].legend(frameon=False)
sm.qqplot(df['log_ret'], line='s', ax=ax[1])
plt.tight_layout()
plt.show()
```



0.6.2 高斯性质

```
periods = ['D','W','ME']
frequency = ['daily','weekly','monthly']

fig, ax = plt.subplots(2,3,figsize=(12,8))
for i, p in enumerate(periods):
    df_resample = df.resample(p).last()
    log_return = np.log(df_resample['SP500']/df_resample['SP500'].shift(1)).dropna()
    sns.histplot(log_return, kde=True,label='Histogram',ax=ax[0][i])
    ax[0][i].set_title(frequency[i])
    sm.qqplot(log_return, line='s', ax=ax[1][i])
plt.tight_layout()
plt.show()
```



0.6.3 波动集聚性

```
fig, ax = plt.subplots(dpi=300,figsize=(10,6))
ax.plot(df['log_ret']*100,
        label='SP500 Log Return')
ax.set_ylabel('Log Returns(%)')
ax.legend()
plt.tight_layout()
plt.show()
```

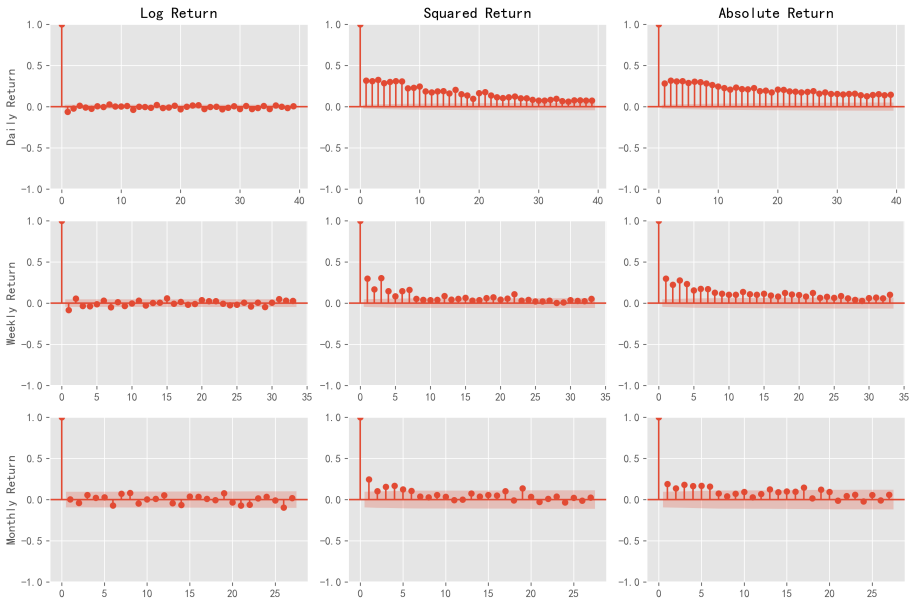
0.6.4 自相关

```
from statsmodels.graphics.tsaplots import plot_acf

periods = ['D','W','ME']
frequency = ['daily','weekly','monthly']

fig, ax = plt.subplots(3,3,figsize=(12, 8))
for i, p in enumerate(periods):
    df_resample = df.resample(p).last()
    log_return = np.log(df_resample['SP500']/df_resample['SP500'].shift(1)).dropna()
    plot_acf(log_return,ax=ax[i][0],title='')
    plot_acf(log_return**2,ax=ax[i][1],title='')
    plot_acf(np.abs(log_return),ax=ax[i][2],title='')
ax[0][0].set_ylabel('Daily Return')
ax[1][0].set_ylabel('Weekly Return')
ax[2][0].set_ylabel('Monthly Return')
ax[0][0].set_title('Log Return')
ax[0][1].set_title('Squared Return')
```

```
ax[0][2].set_title('Absolute Return')
plt.tight_layout()
plt.show()
```



0.6.5 杠杆效应

```
df = pd.read_csv('datasets/sp500_vix.csv',
                 header=0,
                 index_col=0,
                 parse_dates=True)
df.columns = ["SP500", "VIX"]
ret_sp = np.log(df/df.shift(1)).dropna()

fig, ax = plt.subplots(2,1,figsize=(10, 8))
ax[0].plot(df['SP500'].loc["2018":],
             color='blue',lw=2)
ax[0].set_xlabel('Time')
```

```
ax[0].set_ylabel('Close Price')
ax2 = ax[0].twinx()
ax2.plot(ret_sp['SP500'].loc["2018":],
             color='red',alpha=0.7,lw=0.5)
ax2.set_ylabel('Log Return')
sns.regplot(x='SP500', y='VIX',
            data=ret_sp, ax=ax[1])
ax[1].set_xlabel('Log Returns of SP500')
ax[1].set_ylabel('Log Returns of Implied VIX')
plt.tight_layout()
plt.show()
```