

20

Visualizations in Pandas

Pandas 可视化

线图、子图、散点图、柱状图、箱型图...



善良一点，因为你遇到的每个人都在打一场更艰苦的战斗。

Be kind, for everyone you meet is fighting a harder battle.

—— 柏拉图 (Plato) | 古希腊哲学家 | 424/423 ~ 348/347 BC



```
◀ pandas.DataFrame.plot() 绘制线图
◀ pandas.DataFrame.plot.area() 绘制面积图
◀ pandas.DataFrame.plot.bar() 绘制柱状图
◀ pandas.DataFrame.plot.barh() 绘制水平柱状图
◀ pandas.DataFrame.plot.box() 绘制箱型图
◀ pandas.DataFrame.plot.density() 绘制 KDE 线图
◀ pandas.DataFrame.plot.hexbin() 绘制六边形图
◀ pandas.DataFrame.plot.hist() 绘制直方图
◀ pandas.DataFrame.plot.kde() 绘制 KDE 线图
◀ pandas.DataFrame.plot.line() 绘制线图
◀ pandas.DataFrame.plot.pie() 绘制饼图
◀ pandas.DataFrame.plot.scatter() 绘制散点图
◀ pandas.plotting.scatter_matrix() 成对散点图矩阵
```



20.1 Pandas 的可视化功能

Pandas 库本身虽然主要用于数据处理和分析，但也提供了一些基本的可视化功能。这一章介绍如何用 Pandas 绘制折线图、散点图、面积图、柱状图、箱型图等等。

拿到一组样本数据，如果数据量很大，我们不可能一个个观察样本值；这时，我们就需要各种统计量来描述数据集的不同方面，包括中心趋势、离散度和分布形状。

以下是常见的单一特征统计量化描述。

- ▶ 均值 (average, mean) 是数据集中所有值的总和除以数据点的数量。
- ▶ 众数 (mode) 是数据集中出现频率最高的值。一个数据集可以有一个或多个众数。
- ▶ 中位 (median) 数是将数据集中的所有值按大小排序后位于中间位置的值。它不受异常值的影响，用于度量数据的中心趋势。当数据点数量为奇数时，中位数就是中间的值；当数据点数量为偶数时，中位数是中间两个值的平均值。
- ▶ 最大值 (maximum) 是数据集中的最大数值，而最小值 (minimum) 是数据集中的最小数值，用于表示数据的范围。
- ▶ 方差 (variance) 度量了数据点与均值之间的离散程度。较高的方差表示数据点更分散，较低的方差表示数据点更接近均值。
- ▶ 标准差 (standard deviation) 是方差的平方根，用于衡量数据的离散程度。与方差不同，标准差的单位与数据集的单位相同，因此更容易理解。
- ▶ 分位点 (percentile) 是将数据集划分成若干部分的价值，通常以百分比形式表示。例如，第 25 百分位数是将数据集划分成四分之一的值，第 50 百分位数就是中位数。
- ▶ 偏度 (skewness) 度量了数据分布的偏斜程度。如果数据分布偏向左侧 (负偏)，偏度为负数；如果数据分布偏向右侧 (正偏)，偏度为正数。偏度为零表示数据分布大致对称。
- ▶ 峰度 (kurtosis) 度量了数据分布的尖锐程度。峰度值通常与正态分布的峰度值相比较。正峰度表示数据分布具有比正态分布更尖锐的峰值，负峰度表示数据分布的峰值较平缓。

当涉及到多个特征时，我们还需要两个或多个特征的常见统计描述，比如。

- ▶ 质心 (centroid) 是多个特征的平均值，通常用于表示多维数据的中心点。对更高维数据，对每个特征分别求均值的结果就是质心。
- ▶ 协方差 (covariance) 度量了两个特征之间的线性关系，它可以为正数、负数或零。正协方差表示两个特征具有正相关关系，负协方差表示它们具有负相关关系，而零协方差表示它们之间没有线性关系。
- ▶ 皮尔逊相关系数 (Pearson Correlation Coefficient, PCC)，简称相关性系数，是协方差的标准版本，用于度量两个特征之间的线性关系的强度和方向。相关性系数在衡量线性关系时更常用，取值范围为 -1 到 1，其中 1 表示完全正相关，-1 表示完全负相关，0 表示无线性关系。
- ▶ 协方差矩阵 (covariance matrix) 是一个对称方阵，其中对角线元素为方差，其余元素表示不同特征之间的协方差。

相关系数矩阵 (correlation matrix) 相当于是协方差矩阵的标准化版本。它的对角线元素为 1，其余元素为成对特征之间的相关性系数。

从样本数据到某个统计量的过程，从数据角度来看，可以视作一种降维，也可以看成是折叠、压缩。

这些统计量可以帮助我们更好地了解和描述数据集的特征，从而支持数据分析和决策制定过程。在实际应用中，这些描述统计量通常与可视化工具结合使用，以更全面地理解数据的性质。

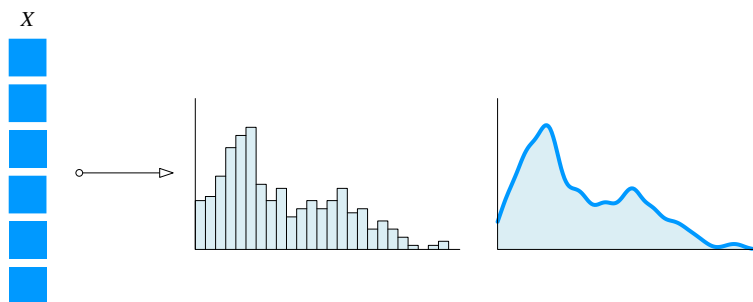


图 1. 单一特征可视化，直方图和 KDE

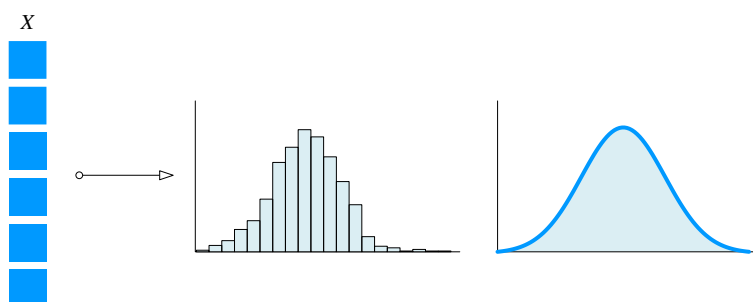


图 2. 单一特征可视化，近似一元高斯分布

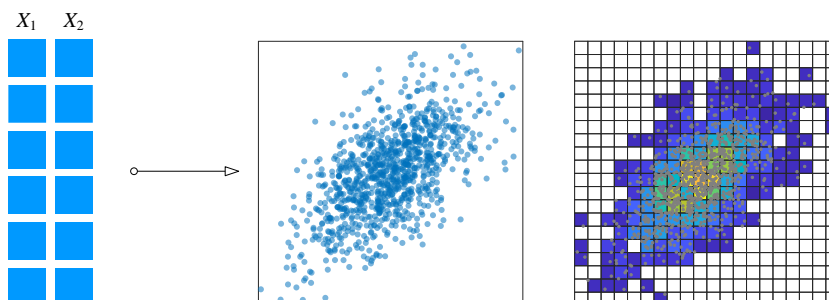


图 3. 两个特征可视化，散点图和直方热图

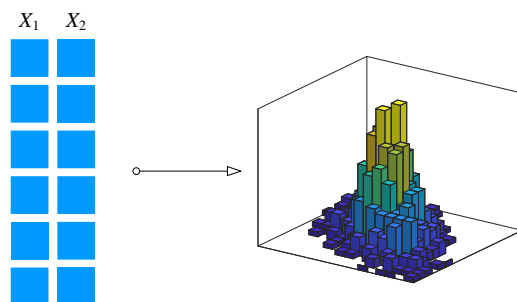


图 4. 两个特征可视化，散点图和三维直方图

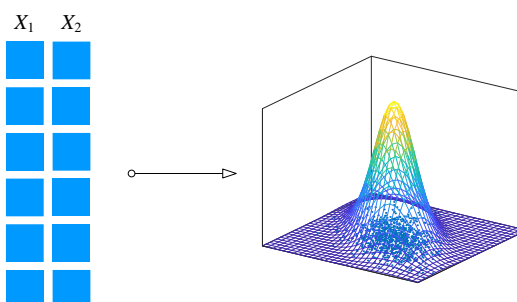


图 5. 两个特征可视化，近似二元高斯分布

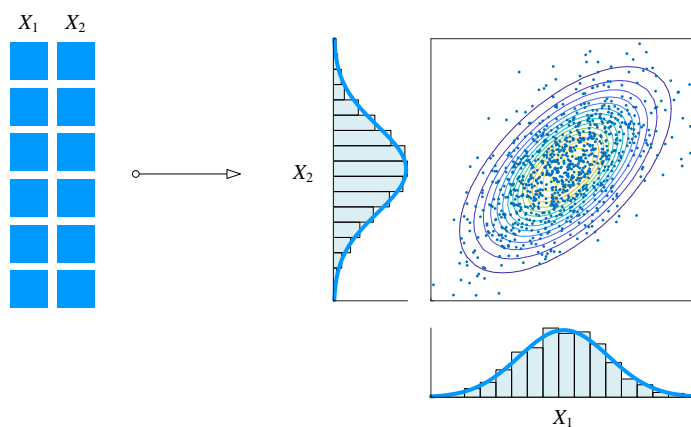


图 6. 两个特征可视化，近似二元高斯分布，边缘分布

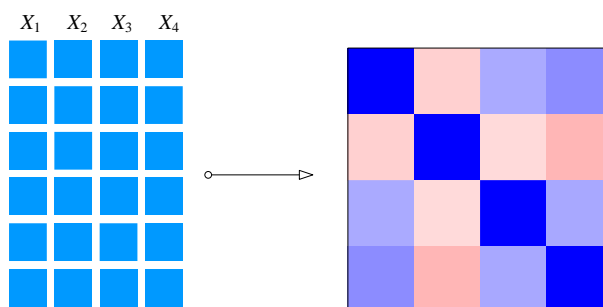


图 7. 多个特征可视化，方差协方差矩阵，相关性矩阵

本书第 12 章介绍过如何用 Seaborn 完成各种统计描述可视化，请大家回顾。

本章用的是利率数据。^a 导入 pandas_datareader。pandas_datareader 从多种数据源获取金融和经济数据，并将这些数据转换为 Pandas DataFrame 的形式。要想使用这个库，先需要安装。如^b所示，在 Anaconda prompt 使用 pip install pandas_datareader 安装这个库。

^c 从 statsmodels.multivariate.pca 导入主成分分析函数 pca。

^d 利用 pandas_datareader 从 FRED (Federal Reserve Economic Data) 下载利率数据，数据格式为 Pandas 数据帧。

^e 用 dropna() 删除数据帧中 NaN。^f 用 rename() 修改数据帧列标签。

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a import pandas_datareader as pdr
b # pip install pandas_datareader
import seaborn as sns
c import statsmodels.multivariate.pca as pca
# 下载数据
d df = pdr.data.DataReader(['DGS6M0', 'DGS1',
                           'DGS2', 'DGS5',
                           'DGS7', 'DGS10',
                           'DGS20', 'DGS30'],
                           data_source='fred',
                           start='01-01-2022',
                           end='12-31-2022')

e df = df.dropna()
# 修改数据帧列标签
f df = df.rename(columns={'DGS6M0': '0.5 yr',
                           'DGS1': '1 yr',
                           'DGS2': '2 yr',
                           'DGS5': '5 yr',
                           'DGS7': '7 yr',
                           'DGS10': '10 yr',
                           'DGS20': '20 yr',
                           'DGS30': '30 yr'})
```

图 8. 下载分析利率数据，代码

20.2 线图：pandas.DataFrame.plot()

图 9 所示为利率数据的线图。

图 11 代码中^a，对于 Pandas 数据帧，我们可以直接用 plot() 方法绘制线图。用 xlabel、ylabel 可以修改线图横纵坐标标签。参数 legend = True (默认值)，在图表显示图例；而 legend = False 则不显示图例。^b 将图片以 SVG 格式保存。

③ 绘制线图时创建一个复杂的子图布局。其中，参数 `subplots=True`，表示我们要创建多个子图，每个子图将显示 `DataFrame` 中的不同列。

参数 `layout = (2,4)` 指定子图的布局，表示总共有 2 行和 4 列子图，共计 8 个子图。这意味着 `DataFrame` 中的 8 个不同数据列将分别在这 8 个子图中显示，具体如图 10 所示。

参数 `sharex=True` 和 `sharey=True`，表示所有子图将共享相同的 x 轴和 y 轴。这意味着所有子图的 x 轴范围和 y 轴范围将相同，以便更容易比较子图之间的数据。

参数 `xticks=[]` 和 `yticks=[]` 设置 x 轴和 y 轴的刻度标签为空列表，意味着不显示。

参数 `xlim = (df.index.min(), df.index.max())` 指定了 x 轴的显示范围，即 x 轴的最小值和最大值。`df.index.min()` 取出时间序列数据帧横轴标签的最小值，`df.index.max()` 时间序列数据帧横轴标签的最大值。

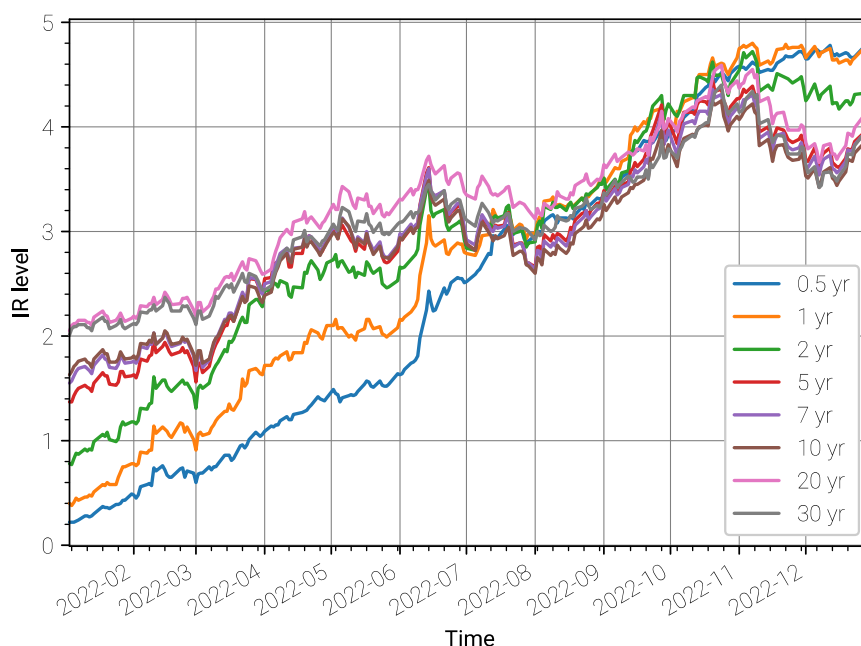


图 9. 利率时间数据线图

为了更好地美化线图，我们还可以使用图 12 代码。④ 代码 `plt.subplots(figsize=(5, 5))` 创建一个图形对象和一个轴对象。

其中，`fig` 是一个图形对象，它代表整个图形窗口。我们可以在这个图形对象上添加一个或多个轴对象，以在图形窗口上绘制图表。

而 `ax` 是一个轴对象，它代表图形窗口中的一个子图或一个坐标系。我们可以在轴对象上绘制数据，并设置轴的属性，如图表大小、标题等。

⑤ 在对数据帧使用 `.plot()` 方法时，用 `ax = ax` 指定了具体轴对象。然后，可以用各种方法美化轴对象 `ax`。

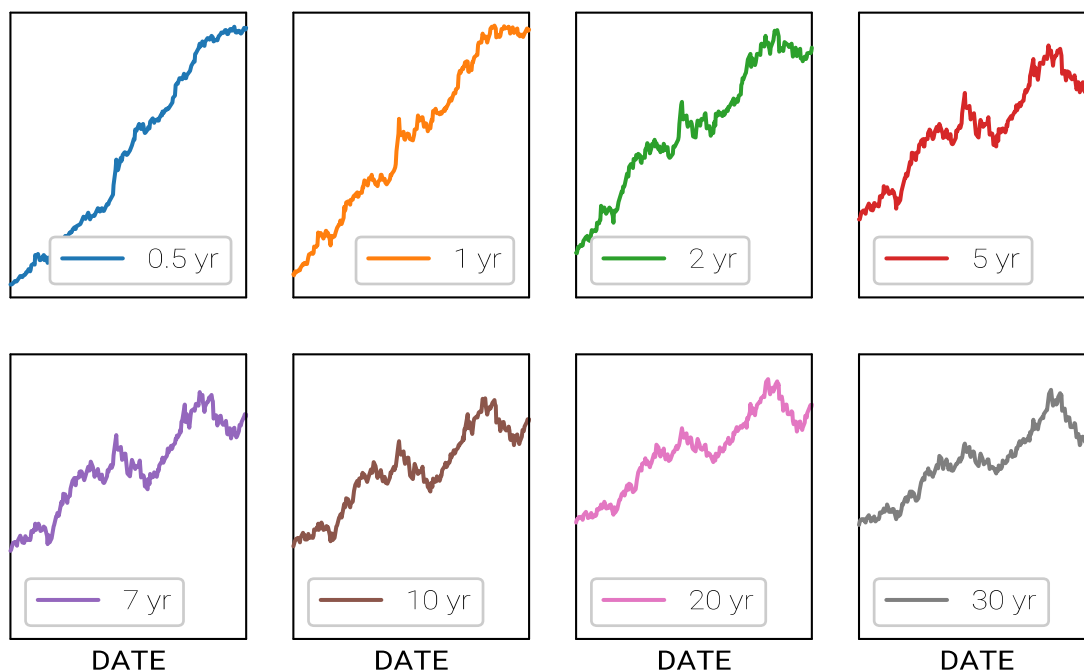


图 10. 利率时间数据线图，子图

```
# 绘制利率走势线图
a df.plot(xlabel="Time", ylabel="IR level",
          legend = True,
          xlim = (df.index.min(), df.index.max()))

b plt.savefig("利率走势线图.svg")

# 绘制利率走势线图，子图布置
c df.plot(subplots=True, layout=(2,4),
          sharex = True, sharey = True,
          xticks = [], yticks = [],
          xlim = (df.index.min(), df.index.max()))

plt.savefig("利率走势线图，子图.svg")
```

图 11. 绘制线图，使用时配合前文代码

```
# 美化线图

a fig, ax = plt.subplots(figsize = (5,5))
b df.plot(ax = ax, xlabel="Time", legend = True)
ax.set_xlim((df.index.min(), df.index.max()))
ax.set_ylim((0,5))
ax.set_xticks([])
ax.set_xlabel('Time')
ax.set_ylabel('IR level')
```

图 12. 绘制并美化线图，使用时配合前文代码

如图 13 所示，我们还可以用 `df.plot.area()` 绘制面积图，对应代码为图 14。

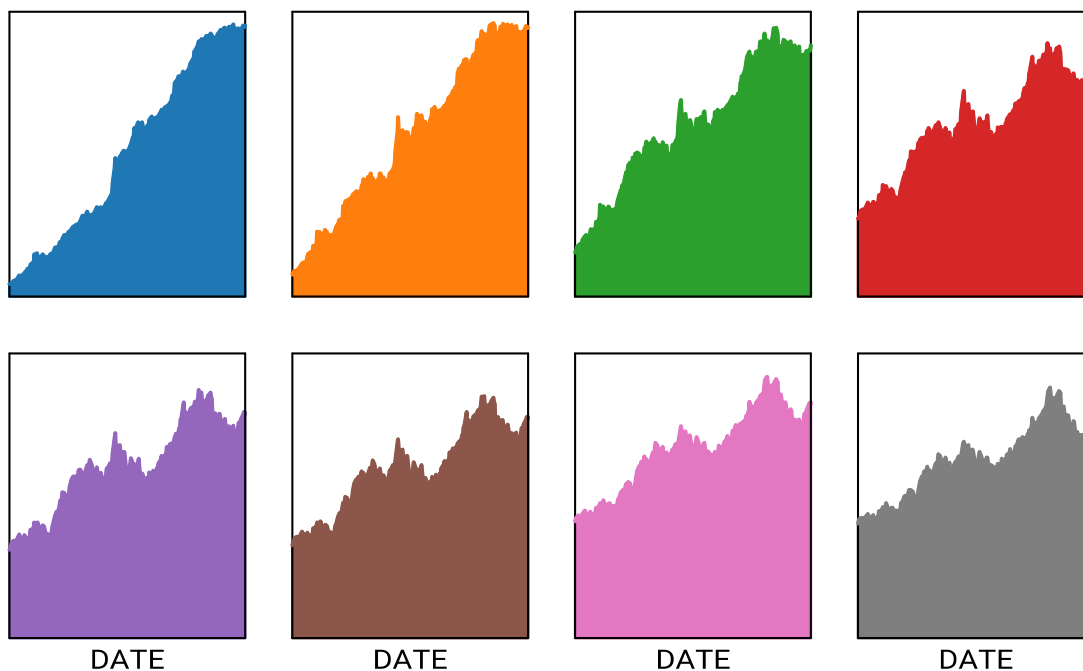


图 13. 利率时间数据面积图，子图

```
# 绘制利率走势线面积图，子图布置
a df.plot.area(subplots=True, layout=(2,4),
               sharex = True, sharey = True,
               xticks = [], yticks = [],
               xlim = (df.index.min(), df.index.max()),
               ylim = (0,5), legend = False)

plt.savefig("利率走势面积图，子图.svg")
```

图 14. 绘制面积图，使用时配合前文代码

图 16 中 ^a 用 `pct_change()` 计算日收益率。日收益率是用来衡量股票、利率在一天内的价格变动幅度的指标。日收益率通常以百分比形式表示，计算方法为：日收益率 = (当日收盘价 - 前一日收盘价) / 前一日收盘价 × 100%。

图 15 所示为利率日收益率折线图，采用 2 × 4 子图布局。

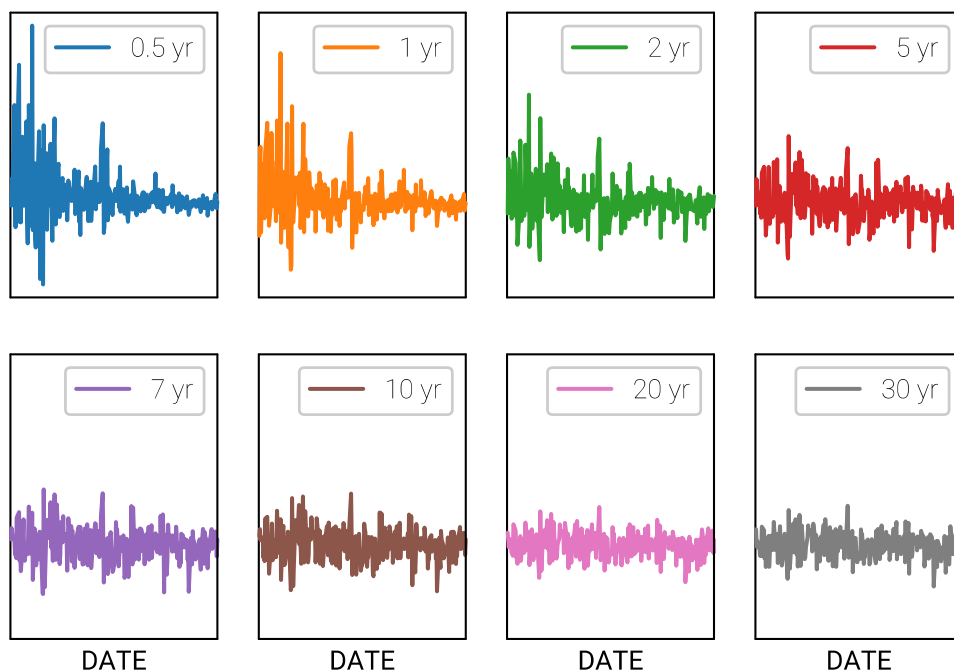


图 15. 利率日收益率折线图，子图

```

# 计算日收益率
a r_df = df.pct_change()

# 绘制利率日收益率，子图布置
b r_df.plot(subplots=True, layout=(2,4),
            sharex = True, sharey = True,
            xticks = [], yticks = [],
            xlim = (df.index.min(), df.index.max()))

plt.savefig("利率日收益率走势图，子图.svg")

```

图 16. 绘制日收益率，使用时配合前文代码

20.3 散点图

图 17 所示为利用 `pandas.DataFrame.plot.scatter()` 绘制散点图。如图 18 代码所示，我们可以用 `x="1 yr"` 和 `y="2 yr"` 指定散点图中要使用的数据列。参数 `gridsize=15` 指定六边形数量。

图 19 所示为用 `pandas.DataFrame.plot.hexbin()` 绘制的六边形图，用来可视化两个变量之间的分布情况。参数 `cmap="RdYlBu_r"` 指定了用于着色六边形的色谱/颜色映射。`"RdYlBu_r"` 是一种颜色映射的名称，它表示一种颜色渐变，从红色到黄色到蓝色，`"_r"` 表示颜色映射的反转。

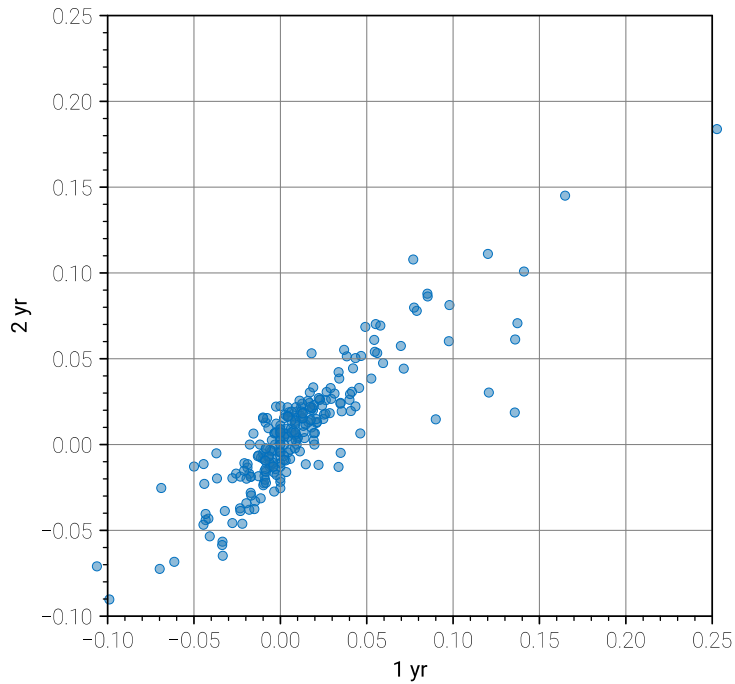


图 17. 日收益率散点图

```
# 绘制散点图
fig, ax = plt.subplots(figsize = (5,5))
a r_df.plot.scatter(x="1 yr", y="2 yr",
                   ax = ax)

a ax.set_xlim(-0.1, 0.25)
  ax.set_ylim(-0.1, 0.25)
  plt.savefig("散点图.svg")
```

图 18. 绘制散点图，使用时配合前文代码

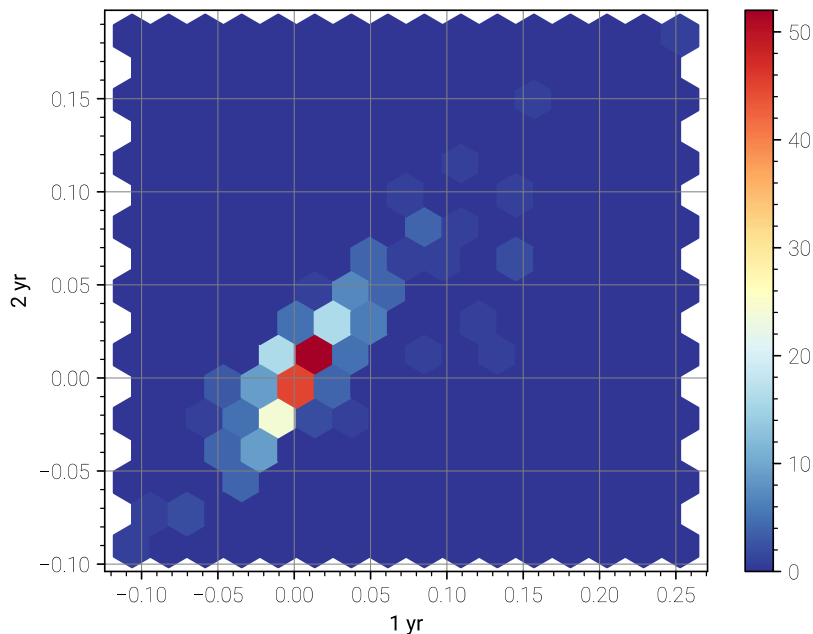


图 19. 六边形图

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

```
# 六边形图
a r_df.plot.hexbin(x="1 yr", y="2 yr",
                  gridsize = 15,
                  cmap="RdYlBu_r")
plt.savefig("六边形图.svg")
```

图 20. 绘制六边形图，使用时配合前文代码

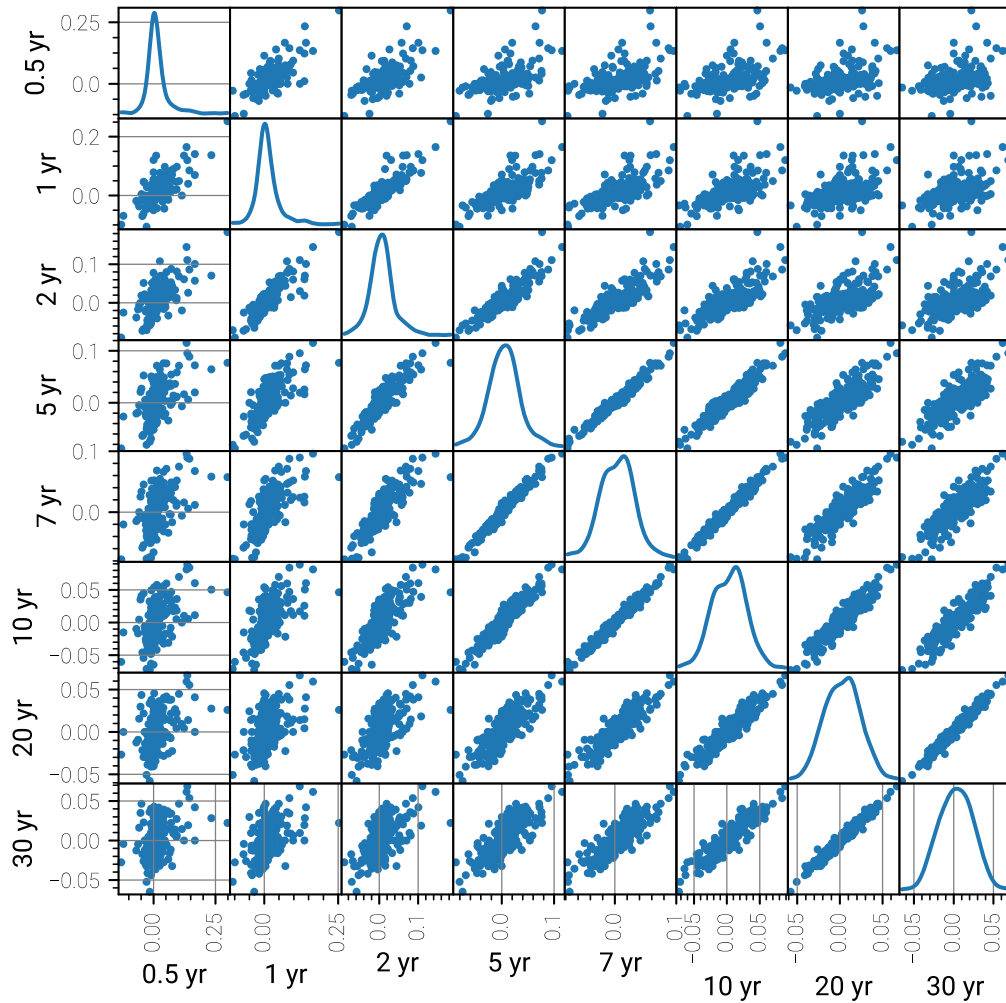


图 21. 成对特征散点图矩阵

```
# 绘制成对特征散点图
a from pandas.plotting import scatter_matrix
b scatter_matrix(r_df, alpha=0.2,
                figsize=(6, 6),
                diagonal="kde")
plt.savefig("成对特征散点图.svg")
```

图 22. 绘制成对特征散点图，使用时配合前文代码

20.4 柱状图

图 23 (a) 所示为用 `pandas.DataFrame.plot.bar()` 绘制的竖直柱状图。图 23 (b) 所示为用 `pandas.DataFrame.plot.barh()` 绘制的水平柱状图。

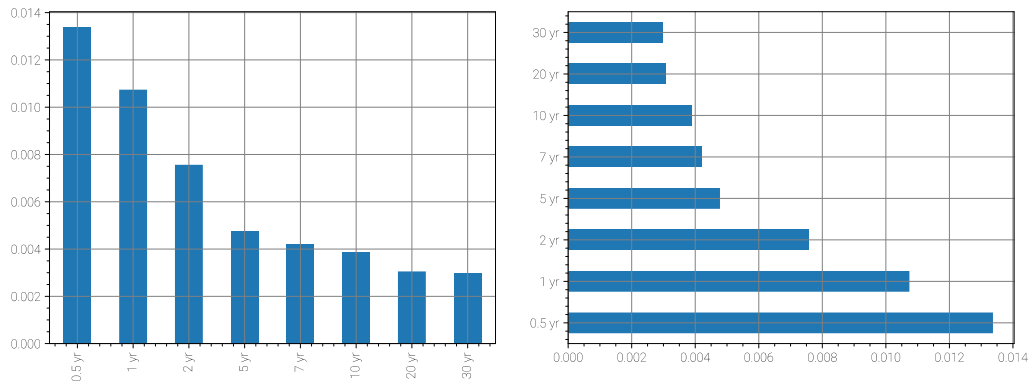


图 23. 柱状图

```
## 柱状图，竖直
a r_df.mean().plot.bar()
  plt.savefig("柱状图.svg")

## 柱状图，水平
b r_df.mean().plot.barh()
  plt.savefig("水平柱状图.svg")
```

图 24. 绘制柱状图，使用时配合前文代码

20.5 箱型图

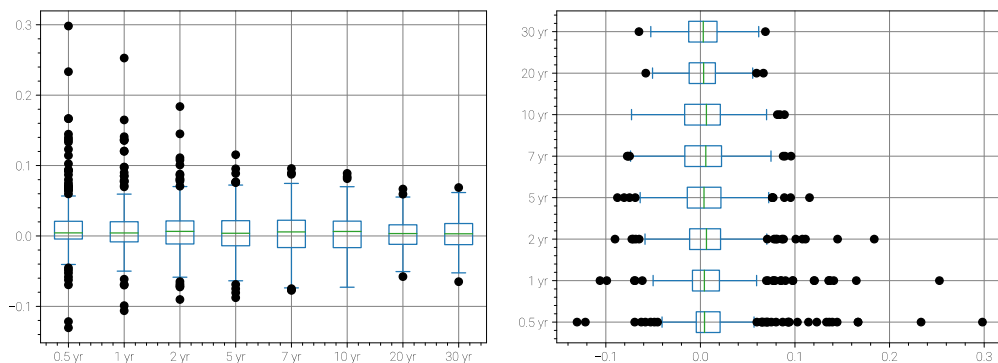


图 25. 柱状图

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课程视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

```

# 绘制箱型图
a r_df.plot.box()
plt.savefig("利率日收益率箱型图.svg")

# 绘制箱型图，水平
b r_df.plot.box(vert=False)
plt.savefig("利率日收益率箱型图，水平.svg")

```



图 26. 绘制箱型图，使用时配合前文代码

20.6 直方图

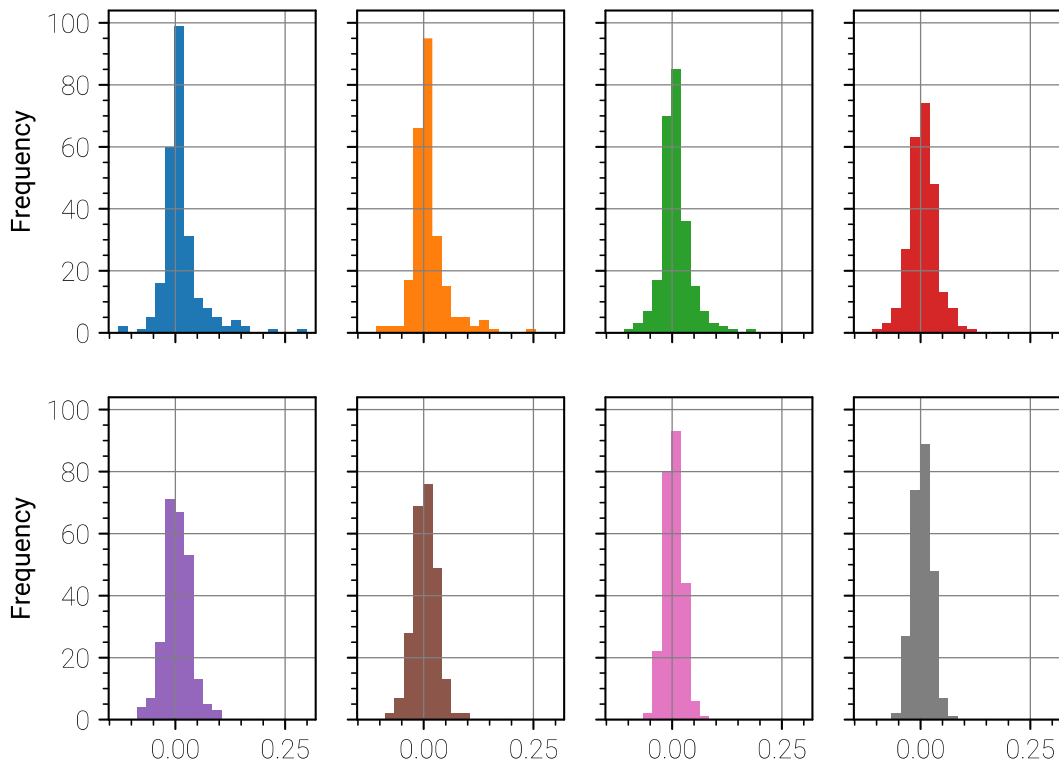


图 27. 直方图

```

# 直方图，子图布置
a r_df.plot.hist(subplots=True, layout=(2,4),
                 sharex = True, sharey = True,
                 bins = 20,
                 legend = False)

plt.savefig("利率日收益率直方图，子图.svg")

```



图 28. 绘制直方图，使用时配合前文代码

20.7 高斯核密度估计

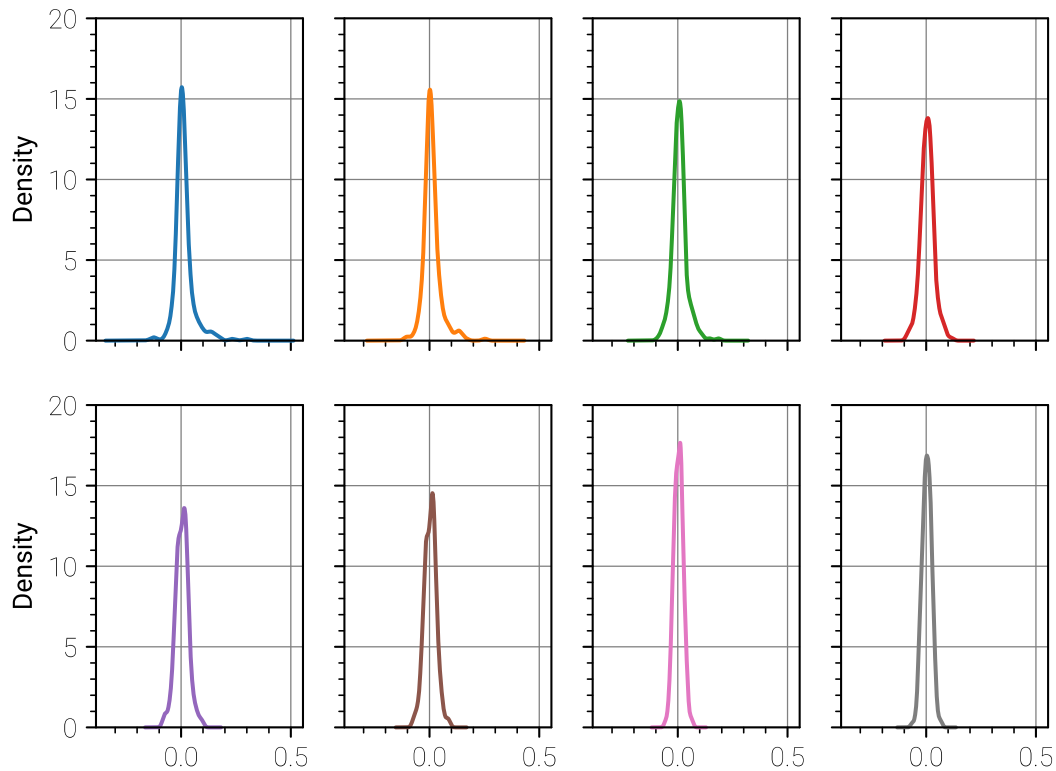


图 29. 高斯核密度估计

```
# KDE, 子图布置
a r_df.plot.kde(subplots=True, layout=(2,4),
                sharex = True, sharey = True,
                ylim = (0,20),
                legend = False)

plt.savefig("利率日收益率KDE, 子图.svg")
```

图 30. 绘制高斯核密度估计，使用时配合前文代码

本章介绍的是一些 Pandas 库中常用的可视化函数，通过这些函数，我们可以在数据分析过程中快速生成各种类型的图表以更好地理解数据。如果需要更复杂的可视化，通常还需要使用 Matplotlib、Seaborn、Plotly 或其他专门的可视化库。