

# 《智能思维与计算社会科学方法论》

## 第五讲 数据可视化——PYTHON

授课教师：郑大庆（博士、副教授）



## 本讲主要内容

- 数据可视化基础与Python简介
- Matplotlib库介绍
- seaborn库介绍
- 金融数据可视化实例：K线图绘制



# 01

## 数据可视化基础与Python简介



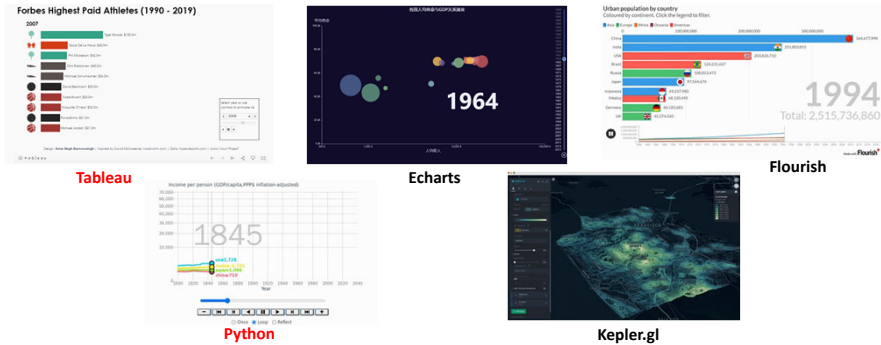
## 数据可视化



- 01 定义与理解**  
数据可视化是将数据以图形、图像或动画等形式展示出来的过程，有助于更直观地理解和分析数据。
- 02 提升决策效率**  
通过数据可视化，能够迅速洞察数据中的规律和趋势，为决策提供有力支持。
- 03 增强沟通效果**  
图形化的数据展示方式更加直观易懂，有助于跨部门、跨领域的沟通与协作。



## 数据可视化工具



## Python环境搭建与基础语法

### Python环境搭建

- 单独安装Python何必要的库：Python官网(<https://www.python.org/>)下载安装包，按照提示进行安装；配置环境变量：将Python的安装路径添加到环境变量中，方便在命令行中运行Python；安装可视化工具：通过pip等工具安装所需的可视化库，如Matplotlib、Seaborn等。
- 安装预先集成Python的发行软件，例如Anaconda。从官网 (<https://www.anaconda.com/download/>) 下载对应的发行版，直接安装，可以免去环境变量的配置这一步骤。

## Python环境搭建与基础语法

### Python基础语法

变量与数据类型：了解Python中的变量、数据类型及基本运算。

控制结构：掌握条件语句、循环语句等控制结构，能够编写简单的程序。

函数与模块：了解函数的定义与调用，掌握常用模块的导入与使用。

## 数据可视化工具概述

### 要点一

#### 主流可视化工具：Tableau

- Tableau：一种商业智能工具，能够快速创建交互式数据可视化图表。
- Power BI：由微软开发的商业智能工具，能够将数据转化为可视化图表，并支持实时更新。
- ECharts：由百度开源的数据可视化工具，支持多种图表类型，适用于各种场景。

### 要点二

#### Python可视化工具

- Matplotlib：Python中最基础的数据可视化库，支持多种图表类型，适合初学者学习。
- Seaborn：基于Matplotlib开发的数据可视化库，更加注重统计图表的美观和易用性。
- Plotly：一种交互式数据可视化工具，支持多种图表类型，并能够轻松创建动态图表。



# 02

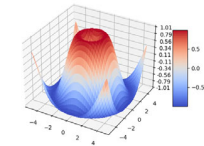
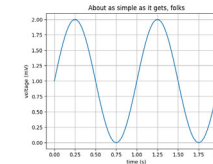
## matplotlib库介绍及应用



## Matplotlib简介

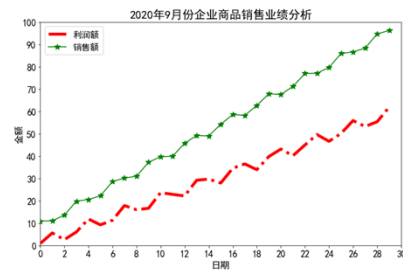
➤ Matplotlib是比较基础的Python绘图库，它是基于NumPy的数组运算，可视化功能非常强大，已经成为Python中最基础的可视化工具。

➤ `import matplotlib.pyplot as plt`

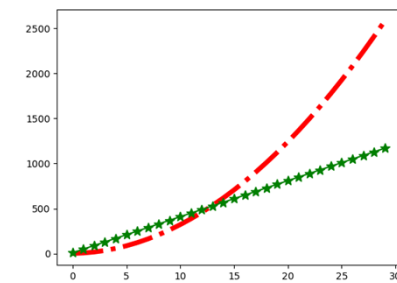
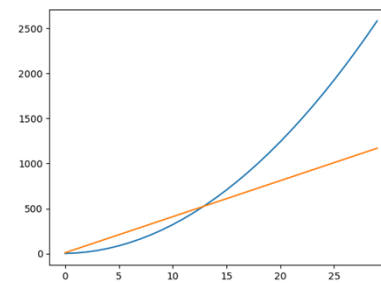


## Matplotlib参数配置

- 1. 线条的设置
- 2. 坐标轴的设置
- 3. 图例的设置
- 4. 其他绘图参数



## 线条的设置



## 线条的设置

# 线条的设置

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

```
x = np.arange(0,30,1) #生成数据
```

```
y1 = 3*x**2 + 2*x + 1
```

```
y2 = 40*x + 9
```

```
plt.plot(x,y1,linestyle='-',color='red',linewidth=5.0) #设置线的颜色，线宽，样式
```

```
plt.plot(x,y2,marker='*',color='green',markersize=10) #添加点，设置点的样式，颜色，大小
```

```
plt.show() #输出图形
```

## 线条的设置

# 线条的设置

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

```
x = np.arange(0,30,1) #生成数据
```

```
y1 = 3*x**2 + 2*x + 1
```

```
y2 = 40*x + 9
```

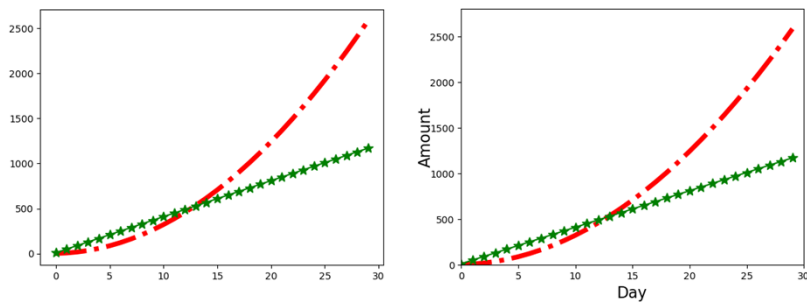
```
plt.plot(x,y1,linestyle='-',color='red',linewidth=5.0) #设置线的颜色，线宽，样式
```

```
plt.plot(x,y2,marker='*',color='green',markersize=10) #添加点，设置点的样式，颜色，大小
```

```
plt.show() #输出图形
```

rc参数名称	含义	说明
lines.linewidth	线条宽度	取0~10之间的数值，默认为1.5
lines.linestyle	线条样式	取值为“-”、“-.”、“.”和“.”4种，默认为“-”
lines.marker	线条上点的形状	取值为“o”、“O”等20种，默认为None
lines.markersize	点的大小	取0~10之间的数值，默认为1
lines.color	线条颜色	取各种颜色参数

## 坐标轴的设置



## 坐标轴的设置

# 坐标轴的设置

```
x = np.arange(0,30,1) #生成数据
```

```
y1 = 3*x**2 + 2*x + 1
```

```
y2 = 40*x + 9
```

```
plt.plot(x,y1,linestyle='-',color='red',linewidth=5.0) #设置线的颜色，线宽，样式
```

```
plt.plot(x,y2,marker='*',color='green',markersize=10) #添加点，设置点的样式，颜色，大小
```

```
plt.xlabel('Day',size=16) #给x轴加上标签
```

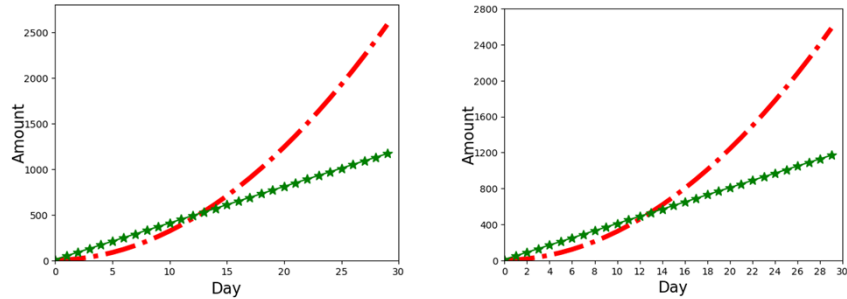
```
plt.ylabel('Amount',size=16,rotation=90,verticalalignment='center') #给y轴加上标签
```

```
plt.xlim(0,30) #设置x轴的刻度
```

```
plt.ylim(0,2800) #设置y轴的刻度
```

```
plt.show() #输出图形
```

## 坐标轴的设置



《智能思维与计算社会科学方法论》

17

## 坐标轴的设置

```
# 坐标轴的设置
x = np.arange(0,30,1) #生成数据
y1 = 3*x**2 + 2*x + 1
y2 = 40*x + 9
plt.plot(x,y1,linestyle='-',color='red',linewidth=5.0) #设置线的颜色，线宽，样式
plt.plot(x,y2,marker='*',color='green',markersize=10) #添加点，设置点的样式，颜色，大小
plt.xlabel('Day',size=16) #给x轴加上标签
plt.ylabel('Amount',size=16,rotation=90,verticalalignment='center') #给y轴加上标签
plt.xlim(0,30) #设置x轴的刻度
plt.ylim(0,2800) #设置y轴的刻度
plt.show() #输出图形
```

《智能思维与计算社会科学方法论》

18

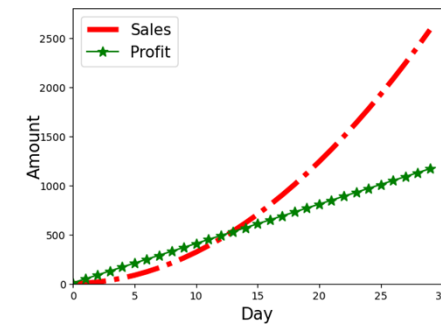
## 坐标轴的设置

```
import numpy as np
import matplotlib.pyplot as plt
#从pyplot导入MultipleLocator类，用于设置刻度间隔
from matplotlib.pyplot import MultipleLocator
.....
x_major_locator=MultipleLocator(2) #把x轴的刻度间隔设置为2，并存在变量里
y_major_locator=MultipleLocator(400) #把y轴的刻度间隔设置为200，并存在变量里
ax=plt.gca() #ax为两条坐标轴的实例
ax.xaxis.set_major_locator(x_major_locator) #把x轴的主刻度设置为2的倍数
ax.yaxis.set_major_locator(y_major_locator) #把y轴的主刻度设置为400的倍数
.....
```

《智能思维与计算社会科学方法论》

19

## 图例设置



《智能思维与计算社会科学方法论》

20

## 图例设置

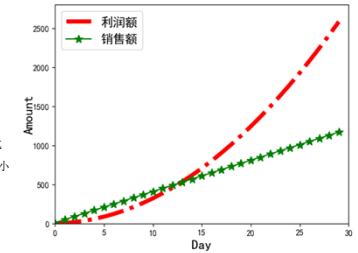
```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.arange(0,30,1) #生成数据
y1 = 3*x**2 + 2*x + 1
y2 = 40*x + 9
```

```
plt.plot(x,y1,linestyle='-',color='red',linewidth=5.0,label='convert A') #设置线的颜色，线宽，样式
plt.plot(x,y2,marker='*',color='green',markersize=10,label='convert B') #设置点的样式，颜色，大小
plt.xlabel('Day',size=16) #给x轴加上标签
plt.ylabel('Amount',size=16,rotation=90,verticalalignment='center') #给y轴加上标签
plt.xlim(0,30) #设置x轴的刻度
plt.ylim(0,2800) #设置y轴的刻度
plt.legend(labels=['Sales', 'Profit'],loc='upper left',fontsize=15) #设置图例
plt.show() #输出图形
```

## 其他参数设置

```
# 其他参数设置
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei'] # 设置中文字体
x = np.arange(0,30,1) #生成数据
y1 = 3*x**2 + 2*x + 1
y2 = 40*x + 9
plt.plot(x,y1,linestyle='-',color='red',linewidth=5.0,label='convert A') #设置线的颜色，线宽，样式
plt.plot(x,y2,marker='*',color='green',markersize=10,label='convert B') #设置点的样式，颜色，大小
plt.xlabel('Day',size=16) #给x轴加上标签
plt.ylabel('Amount',size=16,rotation=90,verticalalignment='center') #给y轴加上标签
plt.xlim(0,30) #设置x轴的刻度
plt.ylim(0,2800) #设置y轴的刻度
plt.legend(labels=['利润额', '销售额'],loc='upper left',fontsize=15) #设置中文图例
plt.show() #输出图形
```



## 03

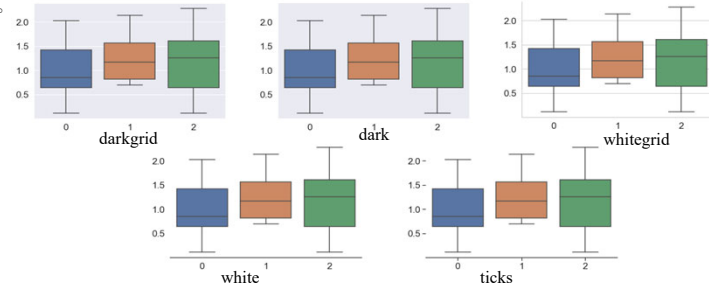
### seaborn库介绍及应用

## Seaborn简介

➤ Seaborn同Matplotlib一样，也是Python进行数据可视化分析的重要第三方包。但Seaborn是在Matplotlib的基础上进行了更高级的API封装，使得作图更加容易，图形更加漂亮。

## Seaborn风格设置

Seaborn有五种风格，分别是darkgrid、dark、whitegrid、white和ticks，默认的是darkgrid。



《智能思维与计算社会科学方法论》

25

## Seaborn风格设置: darkgrid

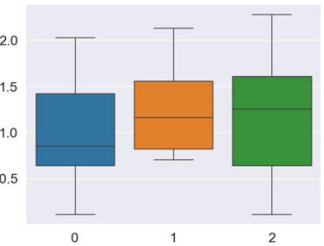
```
import seaborn as sns
```

```
sns.set_style("darkgrid")
```

```
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth": 2.5})
```

```
data = [[1.023312, 0.111484, 0.624475, 0.682342, 1.551981, 2.029264],
        [0.701567, 0.807321, 0.866991, 1.592059, 1.461618, 2.131652],
        [0.110403, 0.523769, 0.985059, 1.524016, 1.635007, 2.279868]]
```

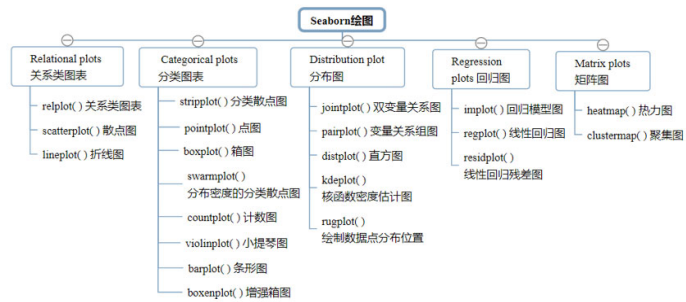
```
sns.boxplot(data=data)
```



《智能思维与计算社会科学方法论》

26

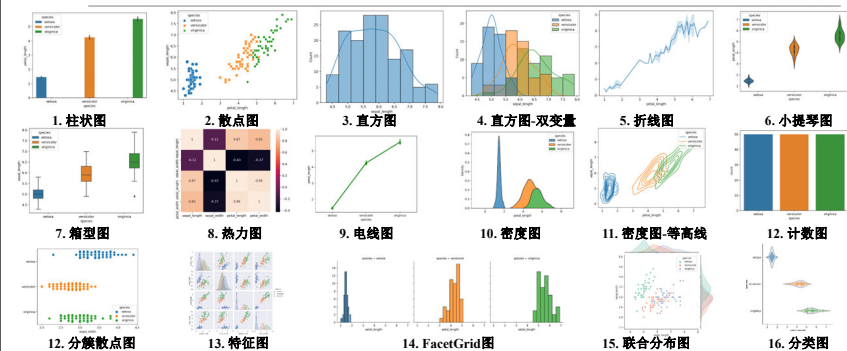
## Seaborn可以绘制的图形



《智能思维与计算社会科学方法论》

27

## Seaborn可以绘制的图形



《智能思维与计算社会科学方法论》

28

## Seaborn绘制的图形——散点图

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.datasets import load_iris

#加载sklearn的鸢尾花数据，共150条数据，
#每个数据有'sepal length', 'sepal width', 'petal length', 'petal width' 4个特征，
#共'setosa', 'versicolor', 'virginica'三种类别。
iris_data = load_iris()
print(iris_data)
```

## Seaborn绘制的图形——散点图

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.datasets import load_iris

#加载sklearn的鸢尾花数据，共150条数据，
#每个数据有'sepal length', 'sepal width', 'petal length', 'petal width' 4个特征，
#共'setosa', 'versicolor', 'virginica'三种类别。
iris_data = load_iris()
print(iris_data)
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1                3.5                1.4                0.2
1                4.9                3.0                1.4                0.2
2                4.7                3.2                1.3                0.2
3                4.6                3.1                1.5                0.2
4                5.0                3.6                1.4                0.2
...                ...                ...                ...                ...
145               6.7                3.0                5.2                2.3
146               6.3                2.5                5.0                1.9
147               6.5                3.0                5.2                2.0
148               6.2                3.4                5.4                2.3
149               5.9                3.0                5.1                1.8

target
0      0
1      0
2      0
3      0
4      0
...    ...
145   2
146   2
147   2
148   2
149   2

[150 rows x 5 columns]
```

## Seaborn绘制的图形——散点图

```
df_iris = pd.DataFrame(iris_data.data, columns = iris_data.feature_names)
df_iris['target'] = iris_data.target
print(df_iris)
```

## Seaborn绘制的图形——散点图

```
class_mapping = { 0: 'setosa', 1: 'versicolor', 2: 'virginica' }
df_iris['target'] = df_iris['target'].map(class_mapping)
df_iris
```

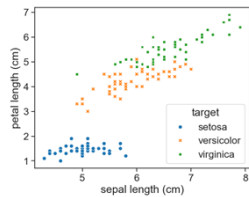
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows x 5 columns

## Seaborn绘制的图形——散点图

#x轴设置为花萼长度，y轴设置为花瓣长度，按照不同的target赋予不同的点型和颜色

```
sns.scatterplot(x='sepal length (cm)', y='petal length (cm)', hue='target', style='target', data=df_iris)
plt.show()
```



## Seaborn绘制的图形——折线图 (课堂练习)

Seaborn中的关系类图表关注的是统计量之间的关系，包括散点图 (scatterplot) 和折线图 (lineplot)，请用折线图绘制鸮尾花petal width和petal length之间的关系。

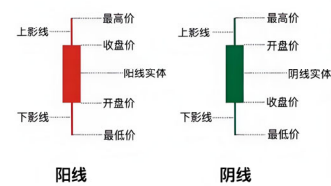
```
1 # 绘制折线图
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import pandas as pd
5 from sklearn.datasets import load_iris
6
7 iris_data = load_iris()
8 df_iris = pd.DataFrame(iris_data.data, columns = iris_data.feature_names)
9 df_iris['target'] = iris_data.target
10 class_mapping = { 0: 'setosa', 1: 'versicolor', 2: 'virginica' }
11 df_iris['target'] = df_iris['target'].map(class_mapping)
12
13 sns.set(style = 'white', color_codes = True)
14 sns.lineplot(x= , y = )
```

## 04

### 金融数据可视化实例：K线图

## K线图介绍

K线图 (Candlestick Chart) 别名阴阳线，是将各种股票某一时间单位内的开盘价、收盘价、最高价、最低价，通过绘图方式表现出来的一种图形。



## 绘制K线图-本地数据

```
# 例题1: 本地数据
import mplfinance as mpf
import pandas as pd
import numpy as np
# 创建示例数据
data = pd.DataFrame({
    'Open': [100, 102, 104, 110],
    'High': [105, 106, 107, 120],
    'Low': [98, 101, 103, 104],
    'Close': [98, 105, 106, 115],
    'Volume': [30, 12, 100, 30]
}, index=pd.date_range("2023-10-01", periods=4))
# 绘制K线图
mpf.plot(data, type='candle', mav = (3, 6), volume = True, style='charles')
```



## 绘制K线图-在线数据

```
# 例题2: 在线数据
import akshare as ak
# 读取中国平安「000001」2024-01-01到2024-12-31日收盘价数据
df = ak.stock_zh_a_hist(symbol="000001", period="daily", start_date="20240101", end_date="20241231", adjust="")
```

## 绘制K线图-在线数据

```
import pandas as pd
# 处理数据
data = df[['日期', '开盘', '最高', '最低', '收盘', '成交量']] # 筛选日期、开盘、最高、最低、收盘、成交量这几列
data = data.rename(columns={'日期': 'Date', '开盘': 'Open', '最高': 'High', '最低': 'Low', '收盘': 'Close', '成交量': 'Volume'})
data['Date'] = pd.to_datetime(data['Date'], format='%Y-%m-%d') # 把 "Date" 列的字符类型数据转化成 pandas.DatetimeIndex 数据类型。
data = data.set_index('Date') # 将日期列设置为行索引
data = data.sort_index(ascending = True) # 将日期按升序排序
```

## 绘制K线图-在线数据

```
print(data)
```

```

      Open  High  Low  Close  Volume
Date
2024-01-02  9.39  9.42  9.21  9.21  1158366
2024-01-03  9.19  9.22  9.15  9.20  733610
2024-01-04  9.19  9.19  9.08  9.11  864194
2024-01-05  9.10  9.44  9.07  9.27  1991622
2024-01-08  9.23  9.30  9.11  9.15  1121156
...
2024-12-25  11.86  12.02  11.84  11.92  1475283
2024-12-26  11.92  11.93  11.78  11.86  1000075
2024-12-27  11.87  11.90  11.66  11.83  1290012
2024-12-30  11.78  11.97  11.78  11.95  1351846
2024-12-31  11.93  11.99  11.70  11.70  1475367
[242 rows x 5 columns]
```

## 绘制K线图-在线数据

```
import mplfinance as mpf
mpf.plot(data, type = 'candle', mav = (3, 6, 9), volume = True)
```



谢谢!  
Questions & Comments

## 推荐资料

- 书籍
  - ✓ 王国平. Python数据可视化. 人民邮电出版社. 2022年
  - ✓ 魏伟一 李晓红. Python数据分析与可视化. 清华大学出版社. 2021年
- 案例
  - ✓ K线图绘制

## 课程分析报告

- 基于Python可视化软件包，绘制茅台的股票走势图，并提交最终的报告，要求如下：
  - ✓ 在线获取数据
  - ✓ 加入基本的K线图
  - ✓ 尝试加入其他的股票分析技术线



谢谢!

Questions & Comments