

## 引言



# Table of Contents

0.1	内容是关于什么的? . . . . .	1
0.2	分析工具: Python . . . . .	2
0.3	安装 Python . . . . .	3
0.4	脚本模式和交互模式 . . . . .	5
0.5	文档与帮助 . . . . .	7



《基于 Python 的经济分析与应用》课程专为国际商务专业硕士研究生设计，注重理论与实践相结合。课程内容涵盖 Python 编程基础、数据采集与处理、经济数据分析、可视化展示及实际案例应用。通过本课程，学生能够掌握利用 Python 进行经济数据分析的方法，提高数据处理与决策支持能力，增强在国际商务环境中的实际应用能力，为未来从事数据驱动的经济分析和国际商务决策打下坚实基础。

## 0.1 内容是关于什么的？

### 0.1.1 涉及课程

- 理论模型？
- 实证分析？
- 数据分析？
- 经济学？
- 产业经济学
- 国际贸易？

### 0.1.2 经济数据分析

对经济数据进行分析长期以来都是政策制定、投资者、企业和消费者关注的焦点：

- 宏观经济形势分析。如毕马威的[中国经济观察](#)季度报告等。
- [中国宏观经济论坛](#)
- 美国经济分析局（Bureau of Economic Analysis, BEA）：The Bureau of Economic Analysis (BEA) promotes a better understanding of the U.S. economy by providing the most timely, relevant, and accurate economic accounts data in an objective and cost-effective manner.
- 数据科学
- 数据服务商

### 0.1.3 本课程的主要内容

围绕经济学、国际贸易中有关主题展开，主要包括：

- 经济数据分析：如增长、不平等、通货膨胀等宏观数据；
- 统计分析方法：t 检验、方差分析等；
- 线性回归方法
- 蒙特卡洛模拟分析
- 机器学习基础方法
- 投入产出模型；
- 网络分析方法；

### 0.1.4 主要参考书

会用到部分 Python 有关的内容，如：

- McKinney [1], [在线阅读](#)
- VanderPlas [2], [在线阅读](#)
- [Python Programming for Economics and Finance](#)

## 0.2 分析工具：Python

我们使用 Python 作为主要的分析工具。根据 [TIOBE Index](#)，Python 是目前最流行的编程语言。

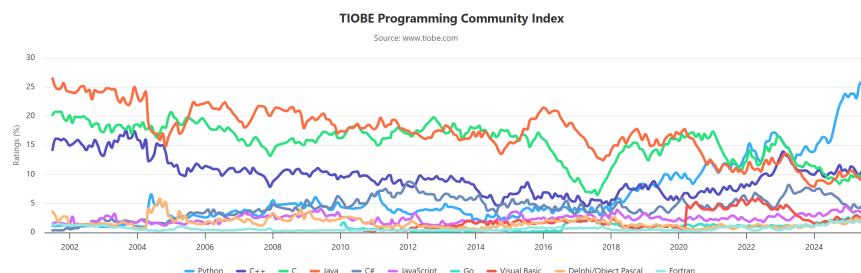


Figure 1: TIOBE Programming Community Index

### 0.2.1 功能强大，应用广泛

Python 广泛应用于机器学习、科学计算等各个领域：

1. 机器学习
2. 数据科学
3. 通讯
4. 网页开发
5. CGI and GUI
6. 自然语言处理
7. 游戏开发
8. 等等

### 0.2.2 Python 的特点

Python 具有许多优点：

1. 易读、易写和易调试；
2. 核心内容易学；
3. 众多库的支持；
4. 初学者友好
5. 支持多平台
6. 网络资源众多

## 0.3 安装 Python

### 0.3.1 下载安装 Python

1. 自[官方网站](#)下载 Python，当前版本 3.13.5
2. 在点击 “Install Now” 安装程序之前，Windows 系统注意勾选：Add Python to PATH，将 Python 的安装路径添加到操作系统的环境变量 Path 中；

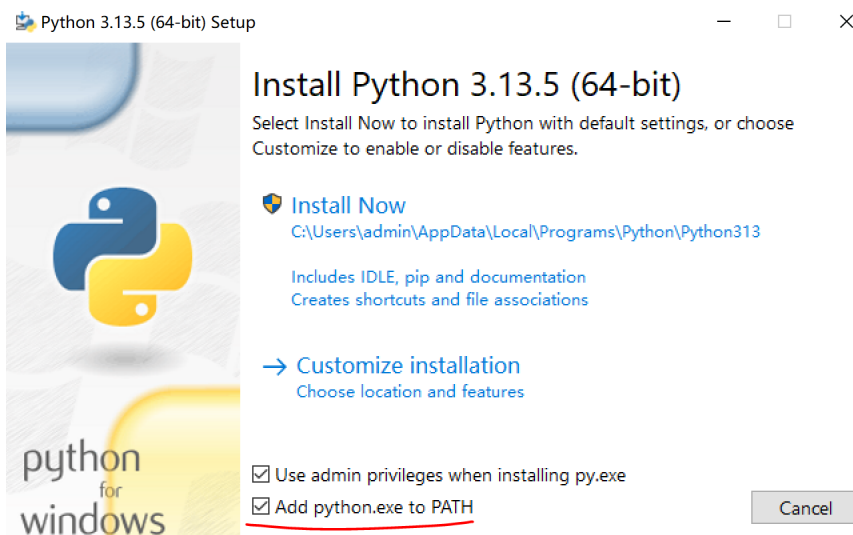


Figure 2: 将 Python 添加至环境

3. 在“命令提示符”输入 `python --version` 查看安装版本。
4. 安装 [JupyterLab](#)。通过命令提示符（或 Mac OS 的终端）安装：`pip install jupyterlab`。另外，需要经常使用 `pip` 安装，建议配置镜像源为 [清华大学开源软件镜像站](#)。
5. 推荐安装 [Visual Code Studio](#)，然后在扩展（Extensions Marketplace）搜索安装：
  - Python
  - Jupyter
  - Quarto
1. 如果只使用 JupyterLab，在命令提示符输入 `jupyter lab` 即可启动；
2. 另外，安装 [Anaconda](#) 也是非常常见的方式，会同时方便的安装常用 Python 程序包。
3. 需要安装第三程序包时，在命令提示符使用 `pip` 安装，例如：
  - Numpy: `pip install numpy`
  - Pandas: `pip install pandas`
  - Matplotlib: `pip install matplotlib`



## 0.4 脚本模式和交互模式

### 0.4.1 交互式

用户输入代码，回车运行。在如 IDLE, Ipython 都可以方便地进行交互式操作。例如：

- 在命令提示符输入 `ipython`，将打开 Ipython 的界面，输入 `3 + 3`，回车，将显示计算结果；
- 在应用程序中打开 IDLE Shell，也可以方便的进行交互式操作；
- 在 Jupyter Notebook 的代码单元格内，输入代码，点击运行显示结果；

### 0.4.2 脚本模式

将代码保存在.py 格式文件中，使用命令提示符运行：

例如，在文件夹 `pyfiles` 中保存有一个文件，存储了下面绘制圆形柱状图的代码。在命令提示符或终端中运行：

```
python pyfiles/polar_plot.py
```

当然，也可以先改变当前文件夹至 `pyfiles` 文件夹，就可以省略路径。### Jupyter notebook 兼有

在 Jupyter Notebook 中试验是否成功安装，打开链接中的例子，将代码复制到一个单元格中，绘制一个[圆形柱状图](#)：

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

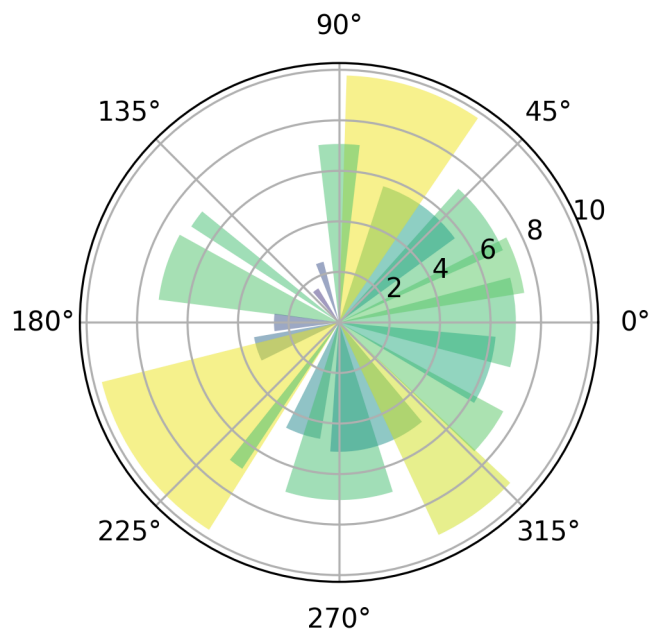
# Fixing random state for reproducibility
np.random.seed(19680801)

# Compute pie slices
N = 20
```

```
theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
radii = 10 * np.random.rand(N)
width = np.pi / 4 * np.random.rand(N)
colors = plt.cm.viridis(radii / 10.)

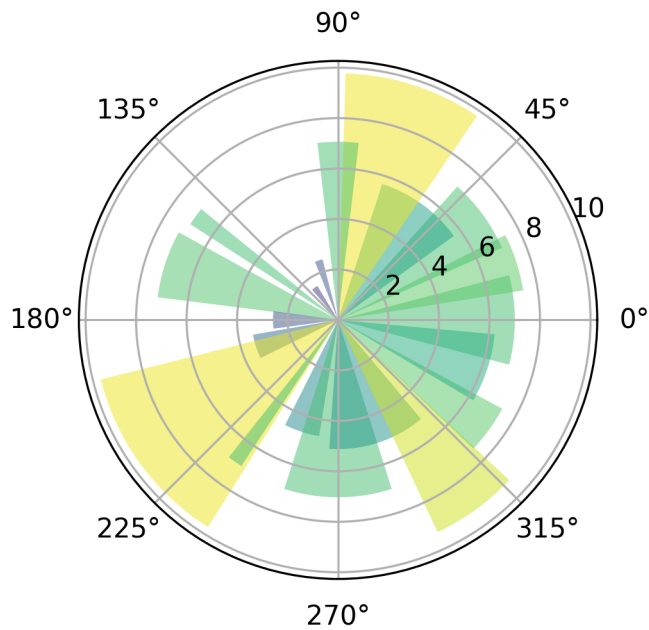
ax = plt.subplot(111, projection='polar')
ax.bar(theta, radii, width=width, bottom=0.0, color=colors, alpha=0.5)

plt.show()
```



Jupyter Notebook 延续了 ipython 中的 `%run` 命令，可以脚本模式运行：

```
%run pyfiles/polar_plot.py
```



## 0.5 文档与帮助

### 0.5.1 `help()` 和?

Python 有非常详细的官方[帮助文档](#)，帮助新用户快速的熟悉其用法。

例如，Python 有一个内置函数 `help()`，可以查看定义的文档，例如对函数 `len()`：

```
help(len)
```

```
Help on built-in function len in module builtins:
```

```
len(obj, /)
```

```
    Return the number of items in a container.
```

由于其重要性，Ipython 和 Jupyter 中可以使用? 作为缩写：

```
len?
```

当然，对自定义的对象也是适用的。下面定义的函数，有一段函数的说明文字（docstring）：

```
def square(x):
    """
    Calculates the square of a given number.

    Args:
        x (int or float): The number to be squared.

    Returns:
        int or float: The square of the input number.
    """
    return x**2
```

如果输入 `help()` 函数：

```
help(square)
```

Help on function square in module `__main__`:

```
square(x)
    Calculates the square of a given number.

    Args:
        x (int or float): The number to be squared.

    Returns:
        int or float: The square of the input number.
```

# Bibliography

- [1] Wes McKinney. *Python for data analysis: Data wrangling with pandas, numpy, and jupyter*. " O'Reilly Media, Inc.", 2022.
- [2] Jake VanderPlas. *Python data science handbook: Essential tools for working with data*. " O'Reilly Media, Inc.", 2016.