# Python数据分析概述

# 目录

# 认识数据分析

狭义数据分析

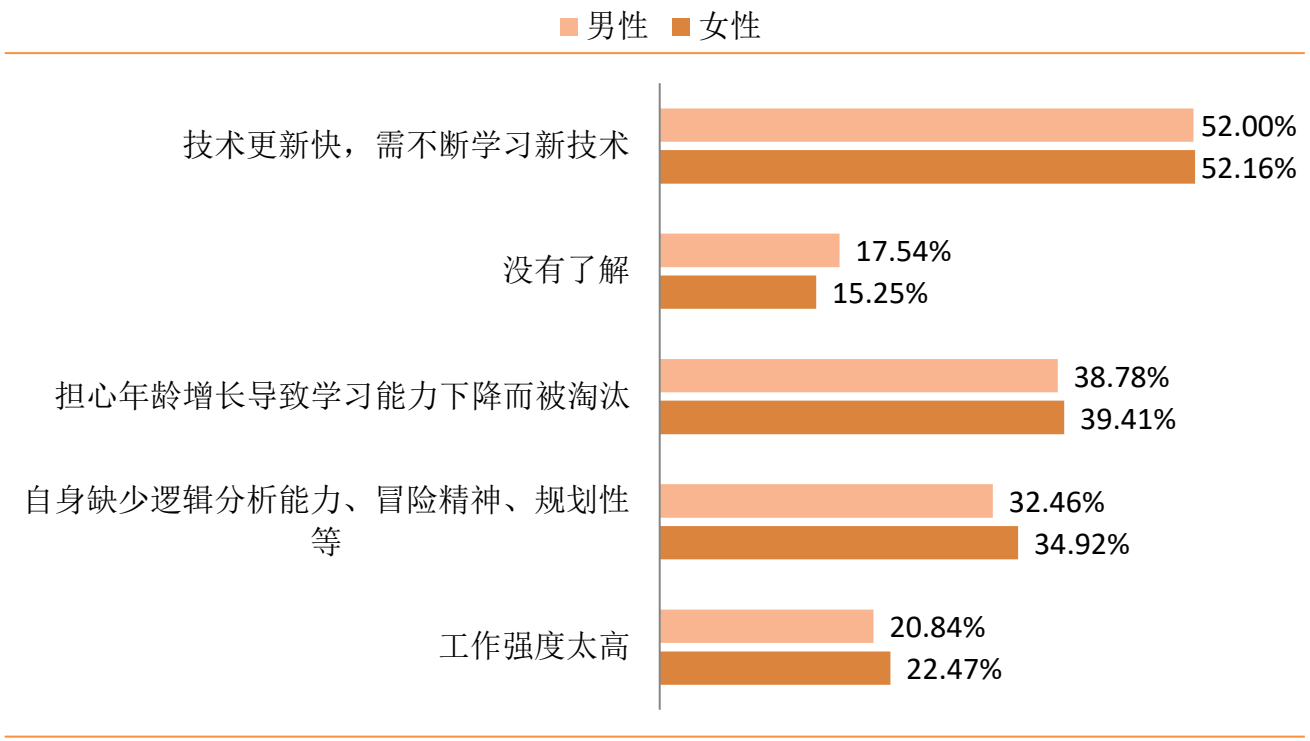
广州市地方财政收入分析

| | 2018 | 2019 | 2019年比2018年增长（%） |
|---|---|---|---|
| 地方财政收入（万元） | 31737500 | 33678876 | 6.1 |

# 认识数据分析

## 狭义数据分析

### 湾区青年女性对数字化岗位的认识【你认为从事数字化职业的困难有什么】



# 认识数据分析

数据挖掘

下列鸢尾花分别属于哪一类：setosa、versicolor、virginica

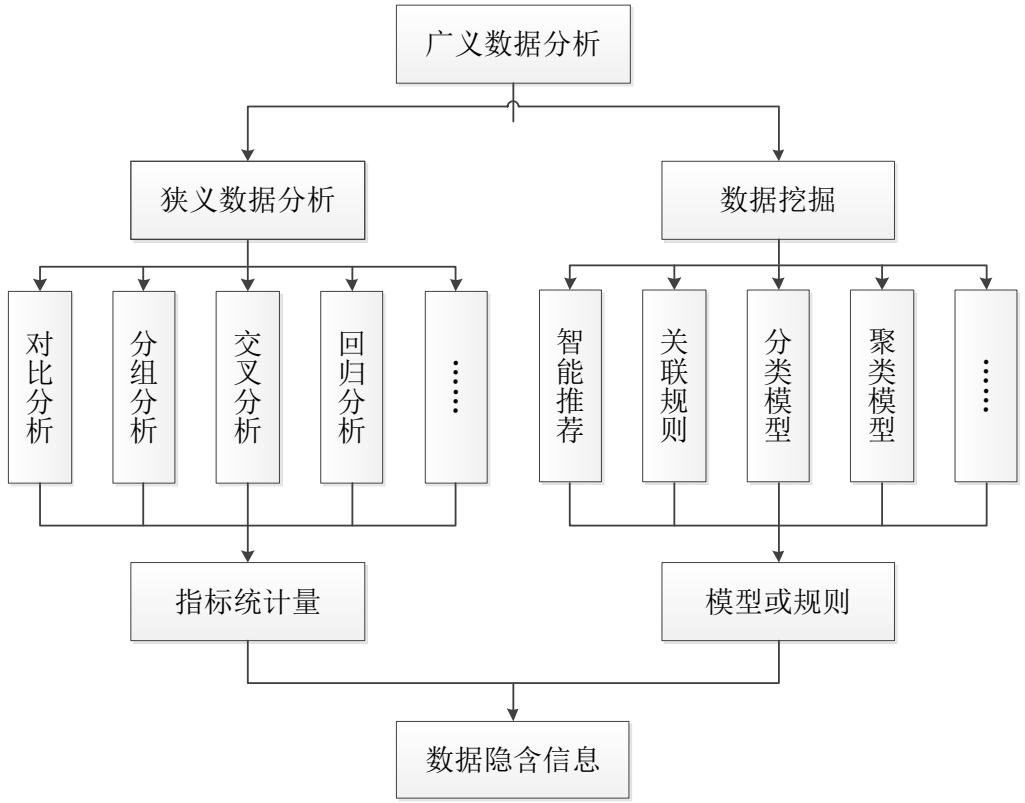


# 认识数据分析

## 数据分析的概念

**广义的数据分析包括狭义数据分析和数据挖掘。**

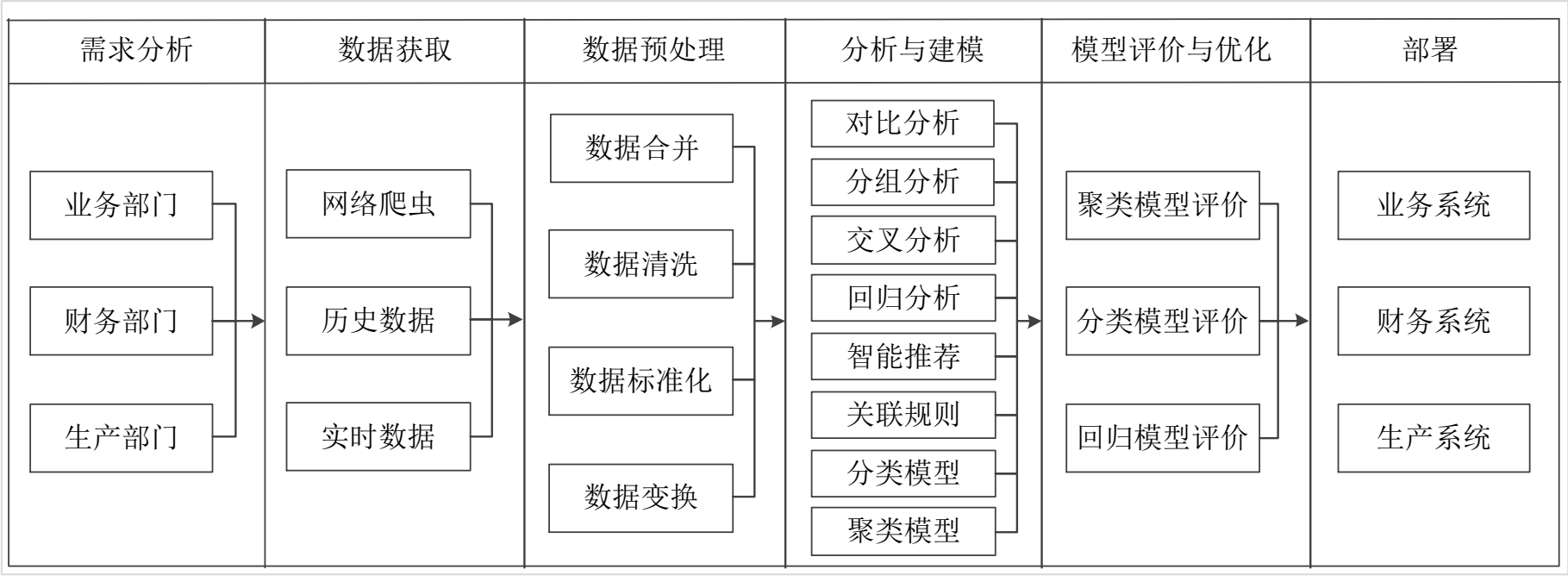
- **狭义的数据分析**是指根据分析目的，采用对比分析、分组分析、交叉分析和回归分析等分析方法，对收集来的数据进行处理与分析，提取有价值的信息，发挥数据的作用，得到一个特征统计量结果的过程。
- **数据挖掘**则是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，通过应用聚类、分类、回归和关联规则等技术，挖掘潜在价值的过程。



# 认识数据分析

## 数据分析的流程

### 典型的数据分析流程



# 认识数据分析

## 数据分析的流程

- 需求分析：数据分析中的需求分析也是数据分析环节的第一步和最重要的步骤之一，决定了后续的分析的方向、方法。
- 数据获取：数据是数据分析工作的基础，是指根据需求分析的结果提取，收集数据。
- 数据预处理：数据预处理是指对数据进行数据合并，数据清洗，数据变换和数据标准化，数据变换后使得整体数据变为干净整齐，可以直接用于分析建模这一过程的总称。
- 分析与建模：分析与建模是指通过对比分析、分组分析、交叉分析、回归分析等分析方法和聚类、分类、关联规则、智能推荐等模型与算法发现数据中的有价值信息，并得出结论的过程。
- 模型评价与优化：模型评价是指对已经建立的一个或多个模型，根据其模型的类别，使用不同的指标评价其性能优劣的过程。
- 部署：部署是指将通过了正式应用数据分析结果与结论应用至实际生产系统的过程。

# 认识数据分析

了解数据分析应用场景

## 1.客户分析

- 对基本数据进行商业行为分析，界定目标客户，并根据客户特点，使用一系列方法对其进行分析，从而提高销售效率。
- 客户采购类型、采购性质进行分类分析制定不同的营销策略。
- 还可以进行客户特征分析、客户忠诚分析、客户注意力分析、客户营销分析和客户收益分析。

# 认识数据分析

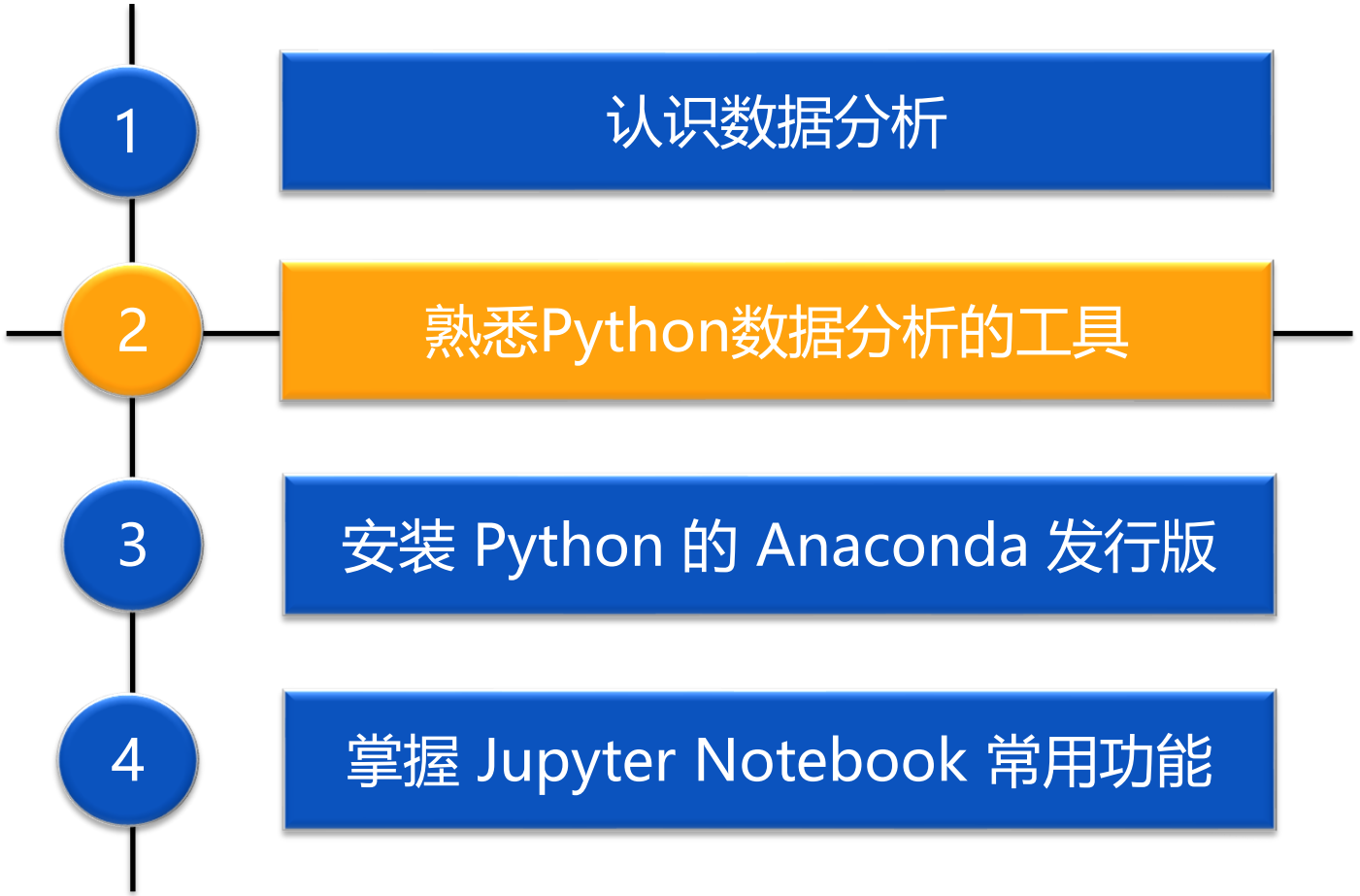
了解数据分析应用场景

**2. 社交媒体分析**

以不同社交媒体渠道生成的内容为基础，实现不同社交媒体的用户分析，访问分析，互动分析等。同时，还能为情感和舆情监督提供丰富的资料。

- **用户分析**主要根据用户注册信息，登录平台的时间点和平时发表的内容等用户数据，分析用户个人画像和行为特征。
- **访问分析**则是通过用户平时访问的内容，分析用户的兴趣爱好，进而分析潜在的商业价值。
- **互动分析**根据互相关注对象的行为预测该对象未来的某些行为特征。

# 目录


1
认识数据分析
2
熟悉Python数据分析的工具
3
安装 Python 的 Anaconda 发行版
4
掌握 Jupyter Notebook 常用功能

# 熟悉Python数据分析的工具

## Python数据分析的优势

- 简单、易学
- 免费、开源
- 高级语言
- 强大的第三方类库
- 可扩展性、可嵌入性、跨平台性

# 熟悉Python数据分析的工具

Python数据分析常用类库

1．IPython——科学计算标准工具集的组成部分

- 是一个增强的Python shell，目的是提高编写、测试、调试Python代码的速度。
- 主要用于交互式数据并行处理，是分布式计算的基础架构。
- 提供了一个类似于Mathematica的HTML笔记本，一个基于Qt框架的GUI控制台，具有绘图、多行编辑以及语法高亮显示等功能。
- Jupyter notebook的运行基础。

# 熟悉Python数据分析的工具

Python数据分析常用类库

2．NumPy(Numerical Python)—— Python 科学计算的基础包

- 快速高效的多维数组对象 ndarray。
- 对数组执行元素级的计算以及直接对数组执行数学运算的函数。
- 读写硬盘上基于数组的数据集的工具。
- 线性代数运算、傅里叶变换，以及随机数生成的功能。
- 将 C、C++、Fortran 代码集成到 Python 的工具。

# 熟悉Python数据分析的工具

## Python数据分析常用类库

3．SciPy——专门解决科学计算中各种标准问题域的模块的集合

- scipy.integrate　数值积分例程和微分方程求解器
- scipy.linalg　扩展了由 numpy.linalg 提供的线性代数例程和矩阵分解功能
- scipy.optimize　函数优化器（最小化器）以及根查找算法
- scipy.signal　信号处理工具
- scipy.sparse　稀疏矩阵和稀疏线性系统求解器
- scipy.special　SPECFUN（这是一个实现了许多常用数学函数的 Fortran 库）的包装器
- scipy.stats　检验连续和离散概率分布、各种统计检验方法，以及更好的描述统计法
- scipy.weave　利用内联 C++代码加速数组计算的工具

# 熟悉Python数据分析的工具

## Python数据分析常用类库

4．Pandas——数据分析核心库

- 提供了一系列能够快速、便捷地处理结构化数据的数据结构和函数。
- 高性能的数组计算功能以及电子表格和关系型数据库（如 SQL）灵活的数据处理功能。
- 复杂精细的索引功能，以便便捷地完成重塑、切片和切块、聚合及选取数据子集等操作。

# 熟悉Python数据分析的工具

## Python数据分析常用类库

5．Matplotlib——绘制数据图表的 Python 库

- Python的2D绘图库，非常适合创建出版物上用的图表。
- 操作比较容易，只需几行代码即可生成直方图、功率谱图、条形图、错误图和散点图等图形。
- 提供了pylab的模块，其中包括了NumPy和pyplot中许多常用的函数，方便用户快速进行计算和绘图。
- 交互式的数据绘图环境，绘制的图表也是交互式的。

# 熟悉Python数据分析的工具

Python数据分析常用类库

6．scikit-learn——数据挖掘和数据分析工具

- 简单有效，可以供用户在各种环境下重复使用。
- 封装了一些常用的算法方法。
- 基本模块主要有数据预处理、模型选择、分类、聚类、数据降维和回归 6 个，在数据量不大的情况下，scikit-learn可以解决大部分问题。

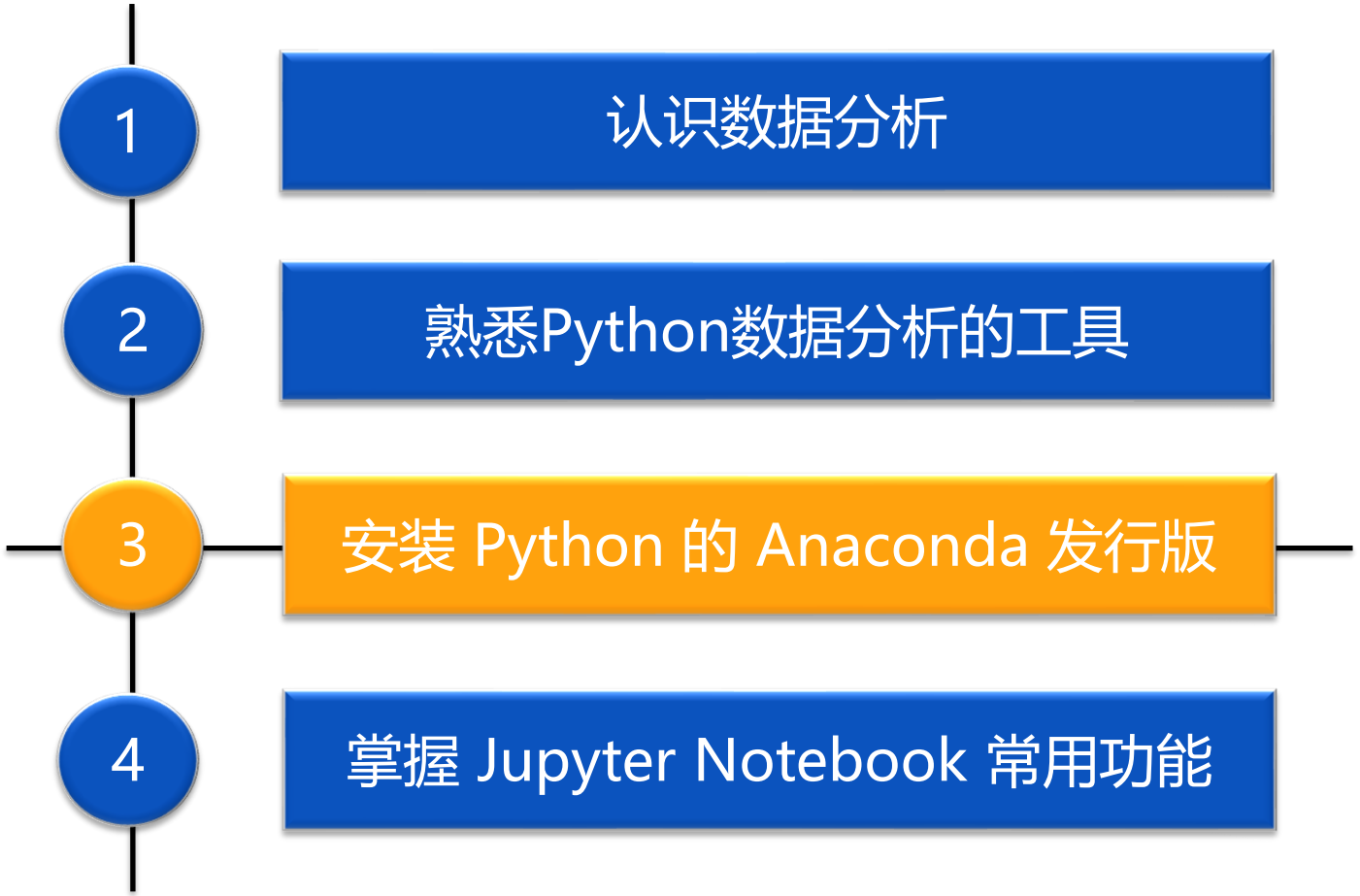
# 熟悉Python数据分析的工具

## Python数据分析常用类库

### 7. Spyder——交互式 Python 语言开发环境

- 提供高级的代码编辑、交互测试和调试等特性。
- 包含数值计算环境。
- 可用于将调试控制台直接集成到图形用户界面的布局中。
- 模仿MATLAB的“工作空间”，可以很方便地观察和修改数组的值。

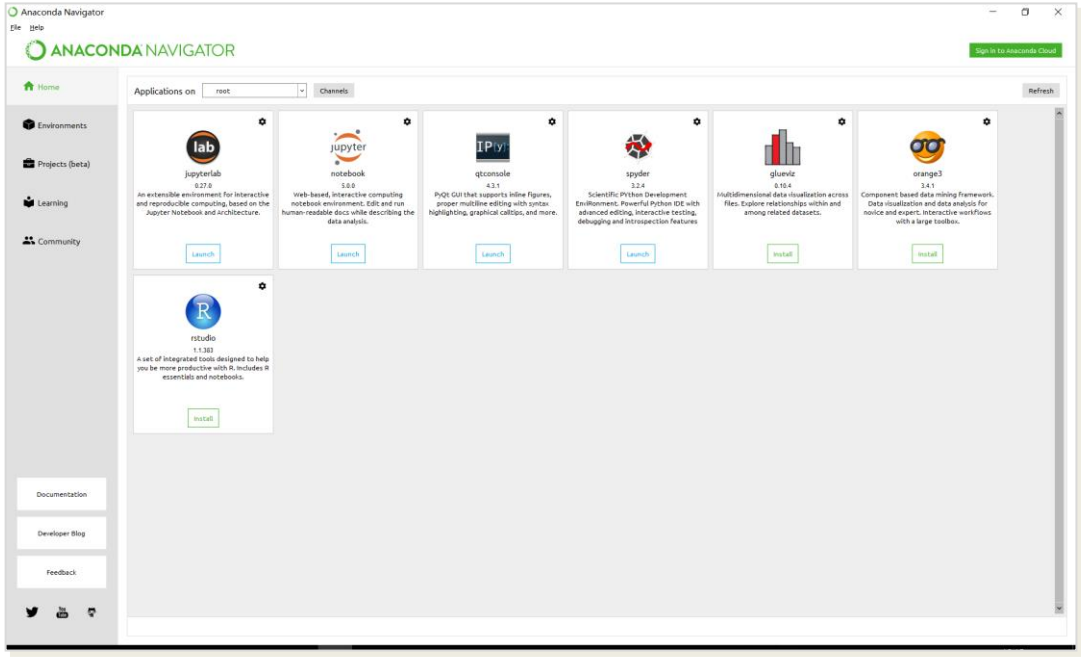
# 目录


1
认识数据分析
2
熟悉Python数据分析的工具
3
安装 Python 的 Anaconda 发行版
4
掌握 Jupyter Notebook 常用功能

# 安装 Python 的 Anaconda 发行版

## 了解 Python 的 Anaconda 发行版

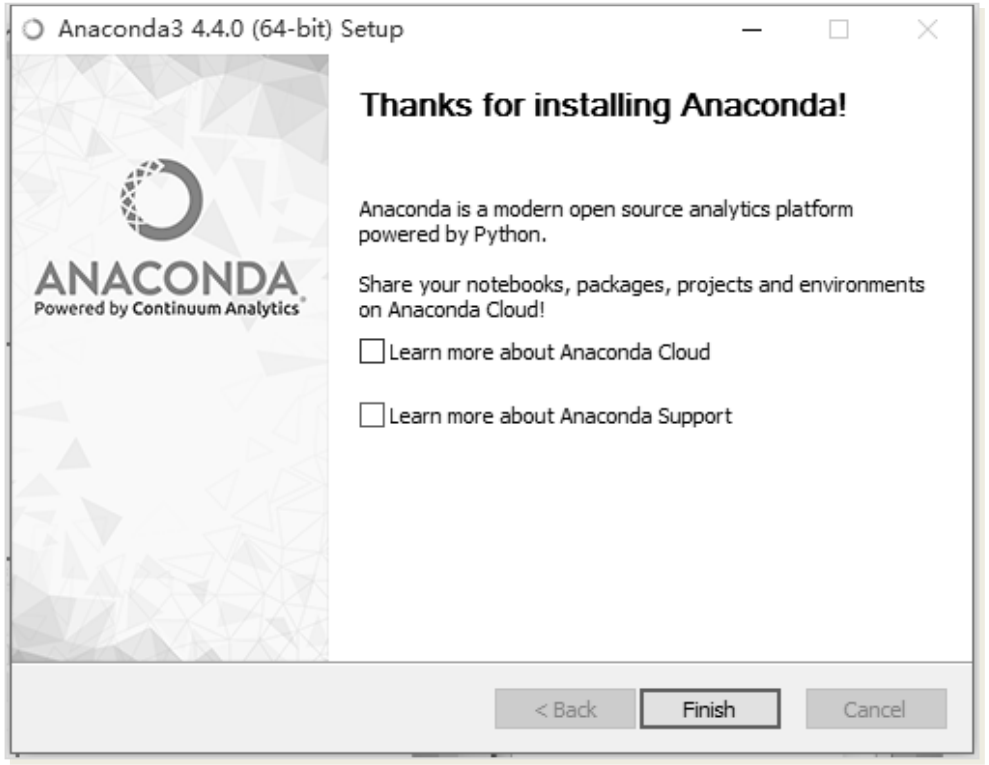
- 预装了大量常用 Packages。
- 完全开源和免费。
- 额外的加速和优化是收费的，但对于学术用途，可以申请免费的 License。
- 对全平台和几乎所有Python版本支持。



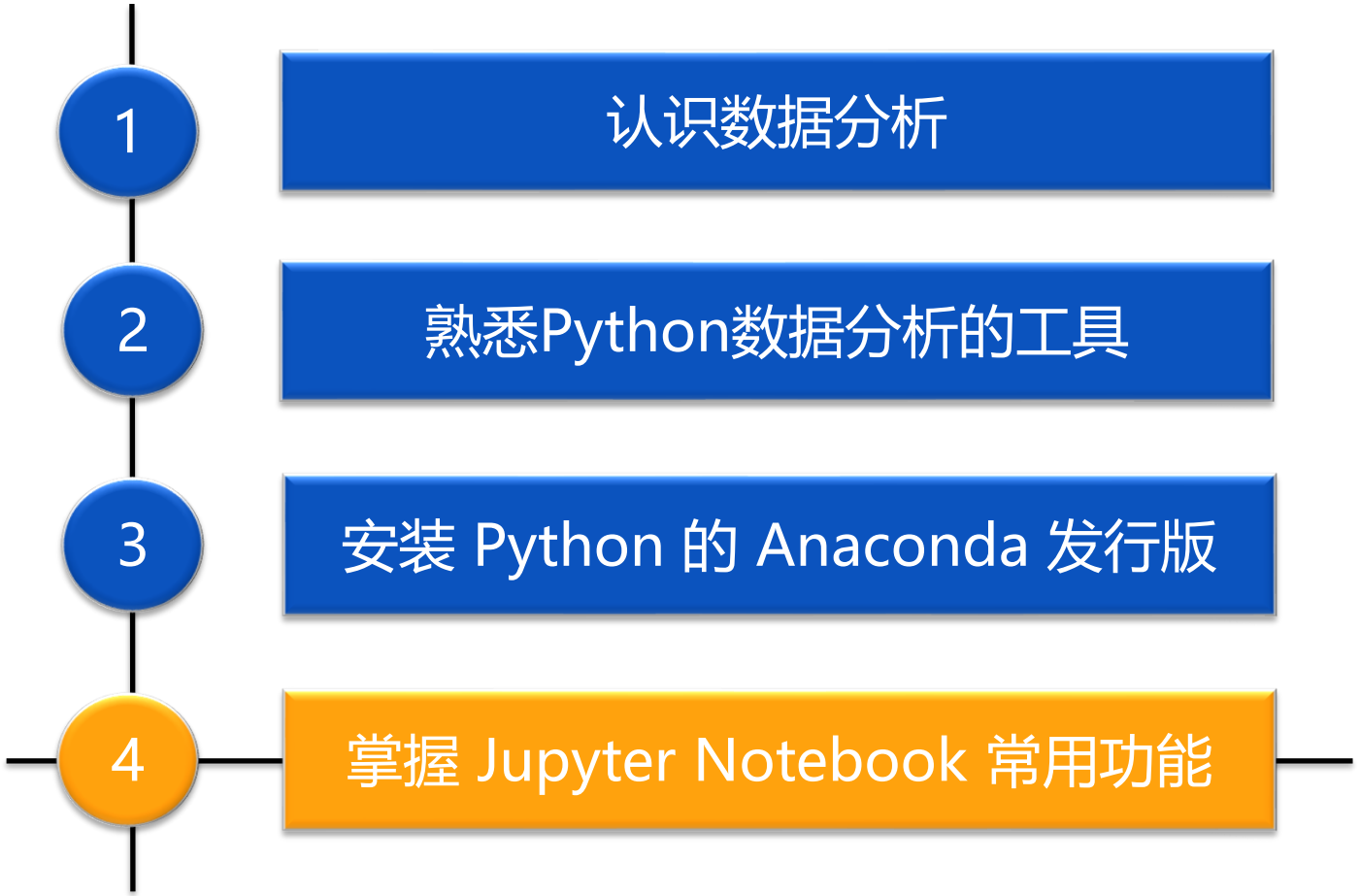
# 安装 Python 的 Anaconda 发行版

在 Windows 系统上安装 Anaconda

安装包——“next”——“I agree”——“All Users(requires admin privileges)”——选择安装路径——“Install”——“finish”。



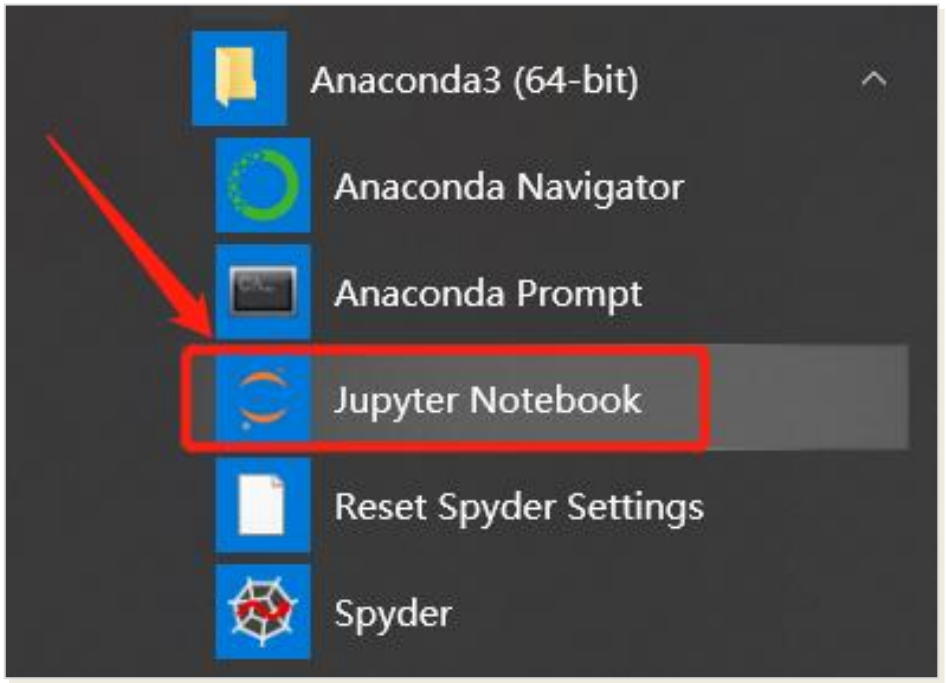
# 目录

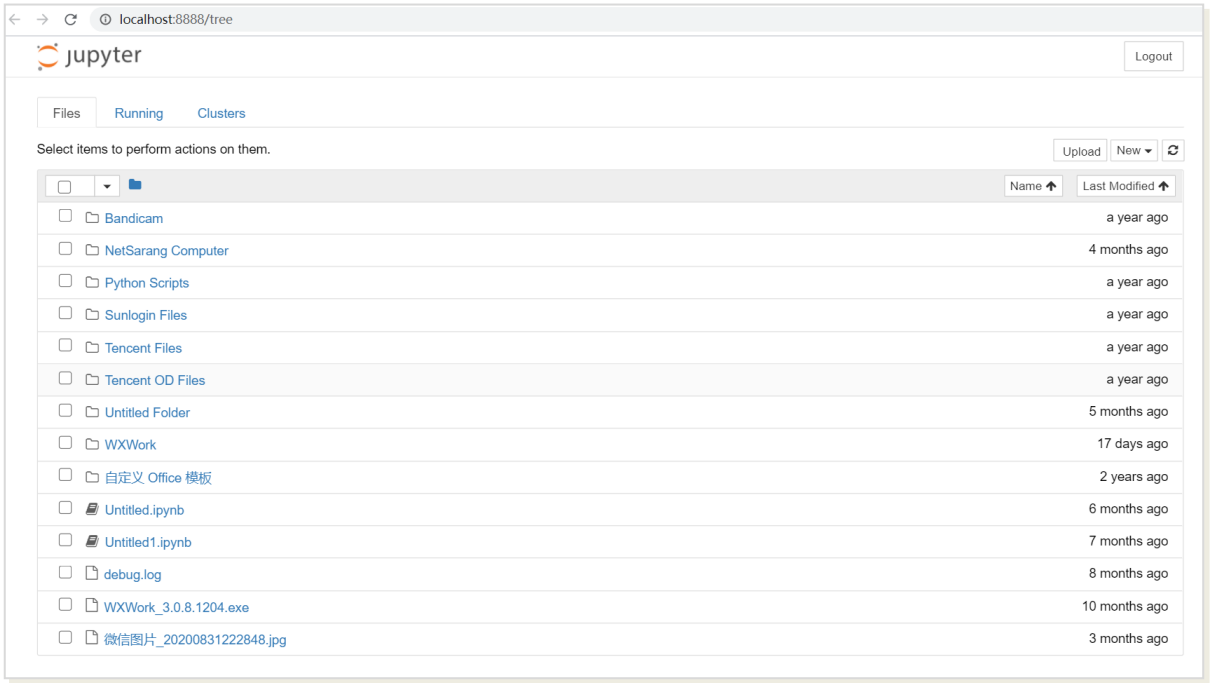

1
认识数据分析
2
熟悉Python数据分析的工具
3
安装 Python 的 Anaconda 发行版
4
掌握 Jupyter Notebook 常用功能

# 掌握 Jupyter Notebook 常用功能

打开并新建一个Notebook

开始菜单 — Anaconda3 — Jupyter Notebook

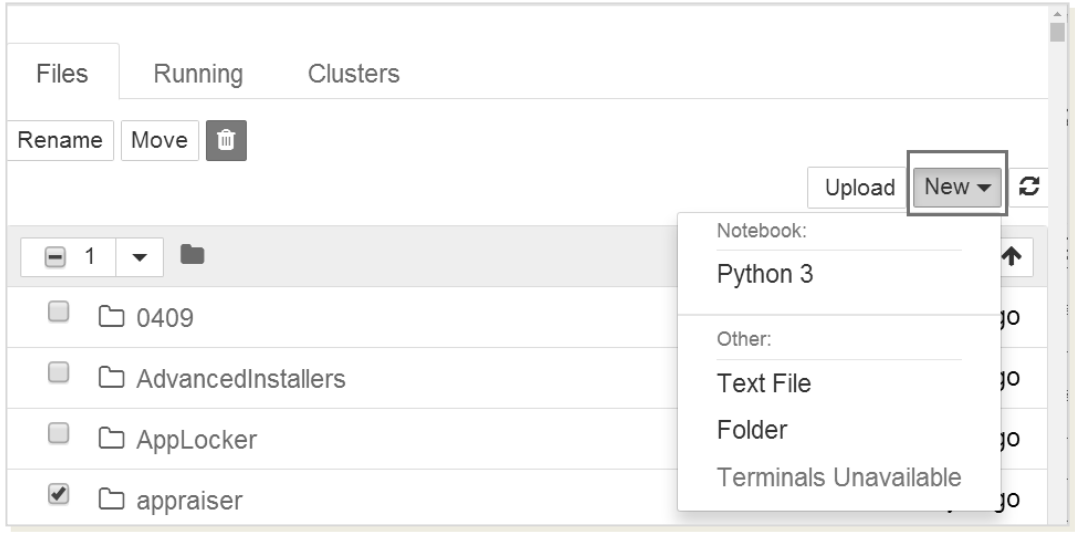


# 掌握 Jupyter Notebook 常用功能

## 打开并新建一个Notebook

- 打开 Jupyter Notebook
- “Text File” 为纯文本型
- “Folder” 为文件夹
- “Python 3” 表示 Python 运行脚本

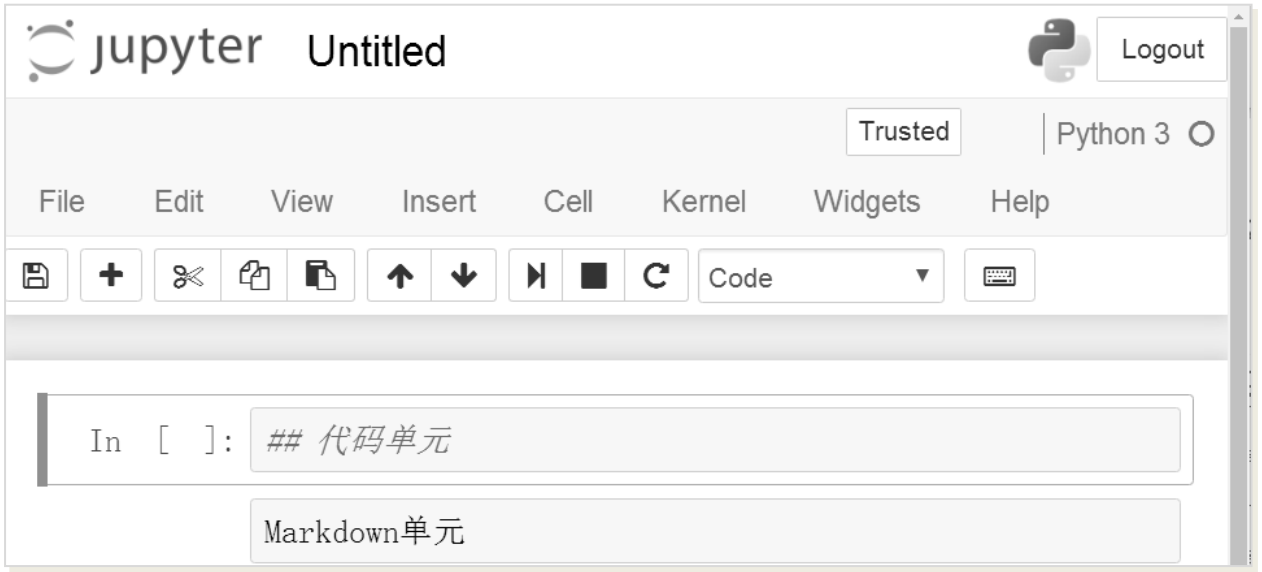


# 掌握 Jupyter Notebook 常用功能

## Jupyter Notebook 的界面及其构成

选择”Python 3”选项，进入 Python 脚本编辑界面，Notebook 文档由一系列单元（Cell）构成，主要有两种形式的单元。

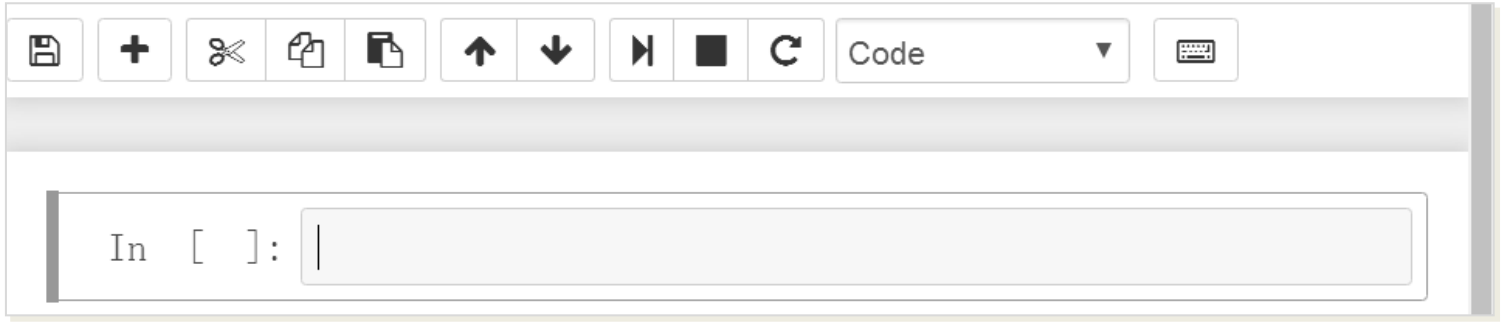
- 代码单元。这里是读者编写代码的地方。
- Markdown 单元。在这里对文本进行编辑。



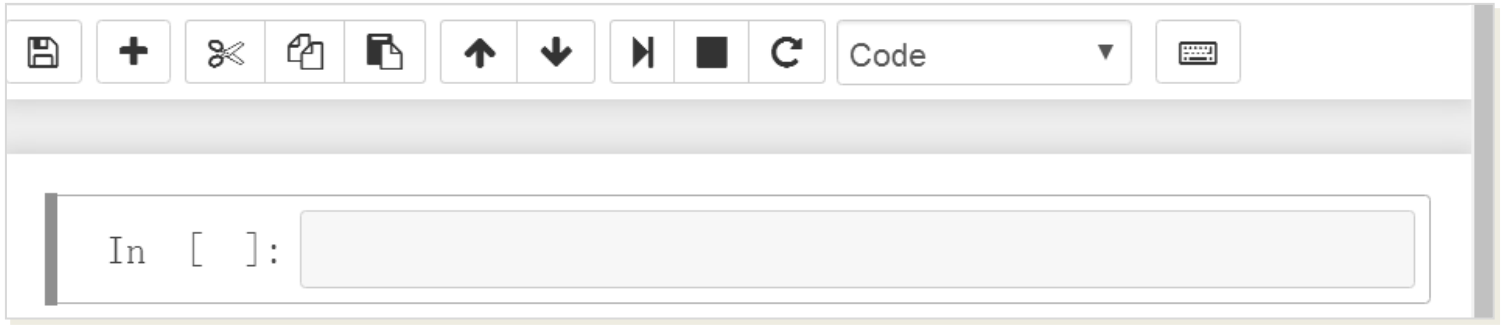
# 掌握 Jupyter Notebook 常用功能

## 编辑界面

编辑模式：用于编辑文本和代码。




命令模式：用于执行键盘输入的快捷命令。



# 掌握 Jupyter Notebook 常用功能

快捷键

“Esc”键：进入命令模式

“Y” 键：切换到代码单元

“M”键：切换到 Markdown 单元

“B”键：在本单元的下方增加一单元

“H”键：查看所有快捷命令

“Shift + Enter”组合键：运行代码

# 掌握 Jupyter Notebook 常用功能

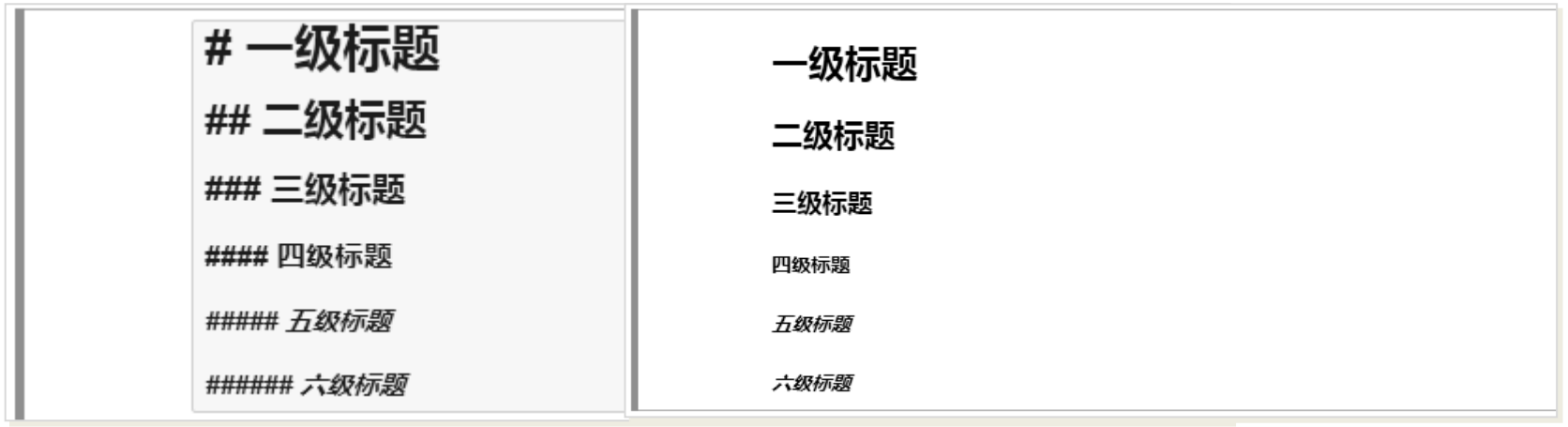
## Markdown

Markdown 是一种可以使用普通文本编辑器编写的标记语言，通过简单的标记语法，它可以使普通文本内容具有一定的格式。

# 掌握 Jupyter Notebook 常用功能

## Markdown

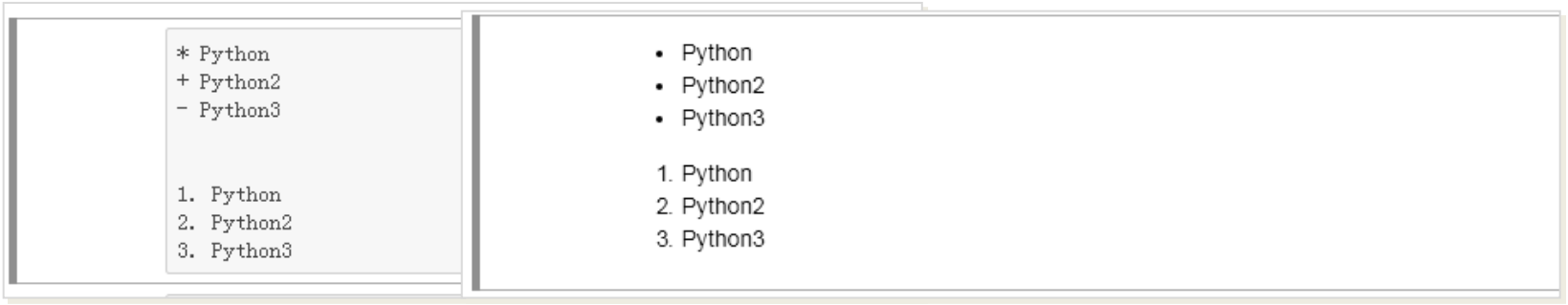
标题：标题是标明文章和作品等内容的简短语句。一个“#”字符代表一级标题，以此类推。



# 掌握 Jupyter Notebook 常用功能

## Markdown

- 列表：列表是一种由数据项构成的有限序列，即按照一定的线性顺序排列而成的数据项的集合。
- 对于无序列表，使用星号、加号或者减号作为列表标记
- 对于有序列表，则是使用数字“，”  “（一个空格）”。



# 掌握 Jupyter Notebook 常用功能

## Markdown

加粗 / 斜体：前后有两个星号或下划线表示加粗，前后有 3 个星号或下划线表示斜体。

| | |
|---|---|
| Python数据分析 | Python数据分析 |
| **Python数据分析** | **Python数据分析** |
| ***Python数据分析*** | ***Python数据分析*** |
| __Python数据分析__ | **Python数据分析** |
| ___python数据分析___ | ***python数据分析*** |

# 掌握 Jupyter Notebook 常用功能

## Markdown

表格：代码的第一行表示表头，第二行分隔表头和主体部分，从第三行开始，每一行代表一个表格行；列与列之间用符号“|”隔开，表格每一行的两边也要有符号“|”。

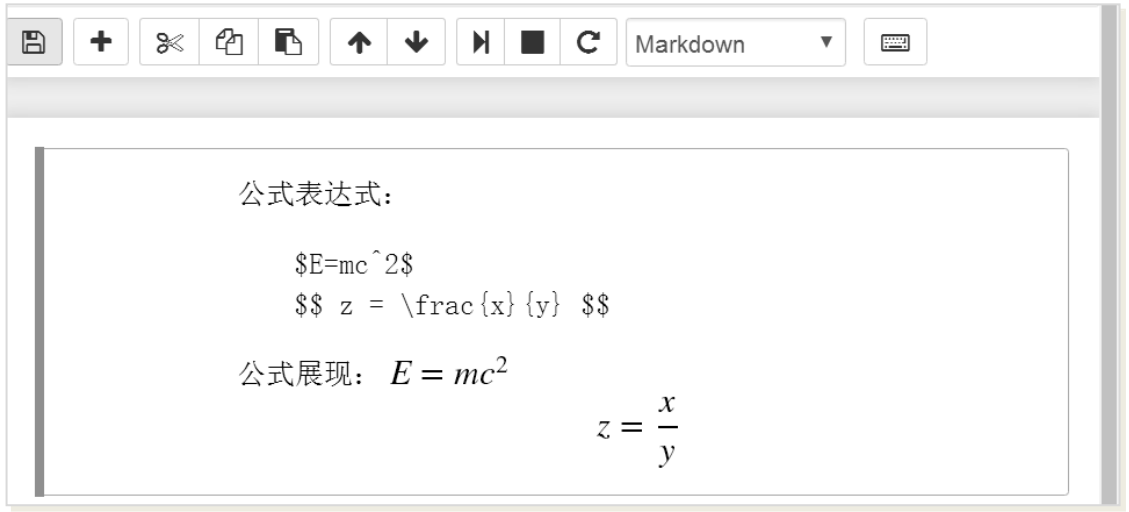
```
Python  | R| MATLAB|
------------- | -------------|----|
接口统一，学习曲线平缓 | 接口众多，学习曲线陡峭 |自由度大，学习曲线较为平缓 |
开源免费  | 开源免费 | 商业收费 |
```

| Python | R | MATLAB |
| --- | --- | --- |
| 接口统一，学习曲线平缓 | 接口众多，学习曲线陡峭 | 自由度大，学习曲线较为平缓 |
| 开源免费 | 开源免费 | 商业收费 |

# 掌握 Jupyter Notebook 常用功能

## Markdown

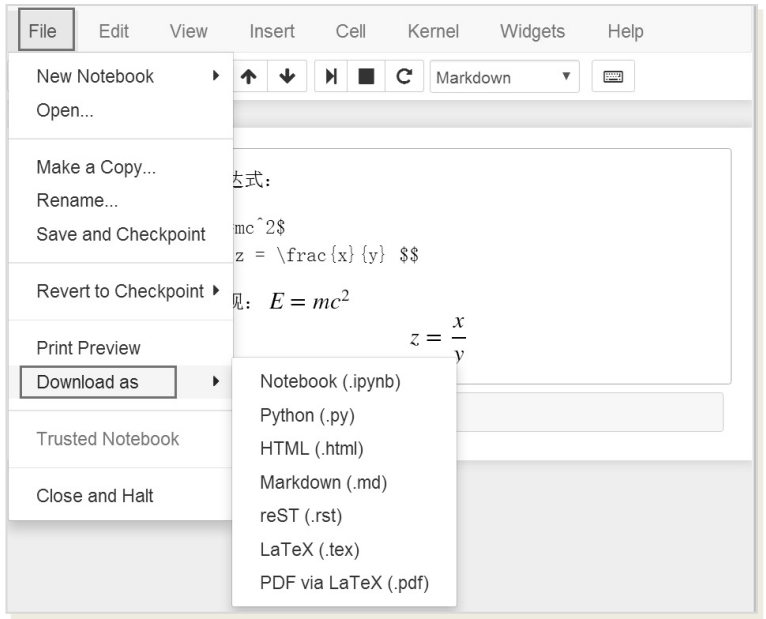
- 数学公式编辑：LaTeX 是写科研论文的必备工具， Markdown 单元中也可以使用 LaTeX 来插入数学公式。
- 在文本行中插入数学公式，应在公式前后分别加上一个“$” 符号
- 如果要插入一个数学区块，则在公式前后分别加上两个“$$”符号。



# 掌握 Jupyter Notebook 常用功能

## 导出功能

- Notebook 还有一个强大的特性，就是导出功能。可以将 Notebook 导出为多种格式，如HTML、Markdown、reST、PDF（通过 LaTeX）等格式。
- 导出功能可通过选择“File”→“Download as”级联菜单中的命令实现。



# Thank you!