

Little_Rookie

[博客园](#) [首页](#) [新随笔](#) [管理](#)

<		2019年5月						>	
日	一	二	三	四	五	六			
28	29	30	1	2	3	4			
5	6	7	8	9	10	11			
12	13	14	15	16	17	18			
19	20	21	22	23	24	25			
26	27	28	29	30	31	1			
2	3	4	5	6	7	8			

我的标签

[r\(50\)](#)
[excel\(16\)](#)
[mysql\(13\)](#)
[python\(13\)](#)
[信用知识\(12\)](#)
[机器学习\(8\)](#)
[tableau\(3\)](#)
[pandas\(2\)](#)
[matplotlib\(1\)](#)
[分类\(1\)](#)
[更多](#)

随笔档案

[2018年10月 \(1\)](#)
[2017年8月 \(2\)](#)
[2017年6月 \(1\)](#)
[2017年4月 \(2\)](#)
[2017年3月 \(5\)](#)
[2017年2月 \(28\)](#)
[2017年1月 \(2\)](#)
[2016年12月 \(25\)](#)
[2016年11月 \(48\)](#)

积分与排名

积分 - 276530
排名 - 1014

最新评论

- Re: Mysql中的定时任务
感谢感谢，学到了
--身轻如阿宝
- Re: Mysql中的定时任务
感谢分享,赞
--古音移魂
- Re: 机器学习算法中的过拟合与欠拟合
博主解说的很好，解决了我的疑问，

Jupyter Notebook 快速入门

Jupyter Notebook (此前被称为 IPython notebook) 是一个交互式笔记本，支持运行 40 多种编程语言。在本文中，我们将介绍 Jupyter notebook 的主要特性，以及为什么对于希望编写漂亮的交互式文档的人来说是一个强大工具。

在开始使用 notebook 之前，我们先需要安装该库。你可以在 [Jupyter 官网](#) 上找到完整的步骤。

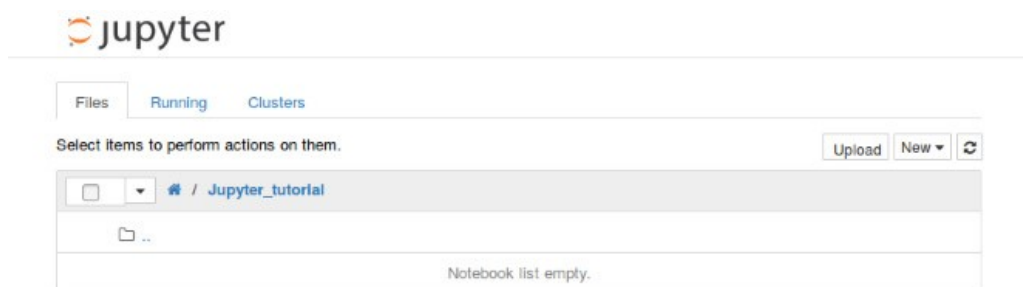
译者注：其实只要 `pip install jupyter` 就可以了

jupyter notebook

运行上面的命令之后，你将看到类似下面这样的输出：

```
[I 20:06:36.367 NotebookApp] Writing notebook server cookie secret to /run/user
/1000/jupyter/notebook_cookie_secret
[I 20:06:36.813 NotebookApp] Serving notebooks from local directory:
/home/your_username
[I 20:06:36.813 NotebookApp] 0 active kernels
[I 20:06:36.813 NotebookApp] The IPython Notebook is running at:
http://localhost:8888/
[I 20:06:36.813 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
```

同时，会在你开启 notebook 的文件夹中启动 Jupyter 主界面，如下所示：



如果想新建一个 notebook，只需要点击 New，选择你希望启动的 notebook 类型即可。

谢谢

--李远李远啊

4. Re:ABC卡

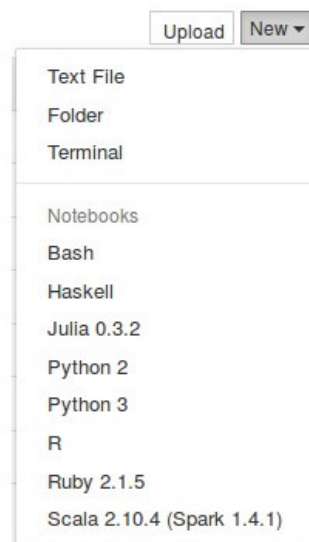
hello, 请问可以交个朋友吗? 我也是做风控的信用评分卡模型的, 可以多交流一下。

--big_smile

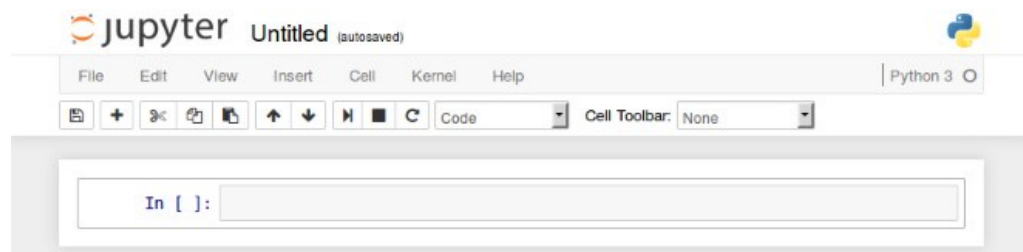
5. Re:Python数据分析之numpy学习

写的很好, 谢谢

--Loser_King



这里, 因为我只有一个 Python 内核, 所以我们运行一个 Python notebook。在新打开的标签页中, 我们会看到 notebook 界面, 目前里面什么也没有。



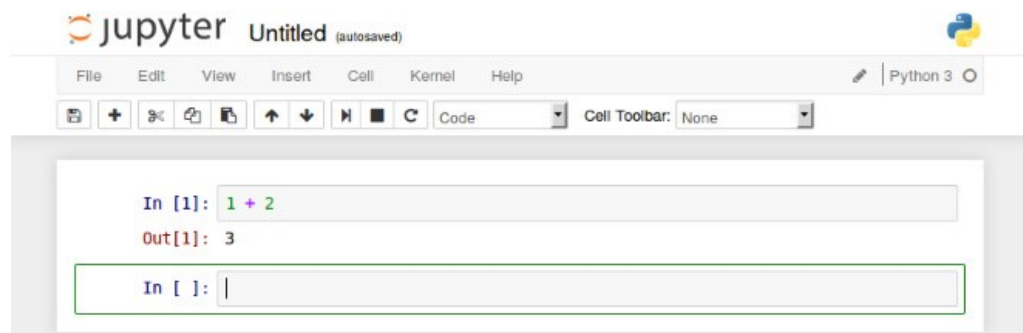
notebook 界面由以下部分组成:

1. notebook 的名称
2. 主工具栏, 提供了保存、导出、重载 notebook, 以及重启内核等选项
3. 快捷键
4. notebook 主要区域, 包含了 notebook 的内容编辑区

慢慢熟悉这些菜单和选项。如果想要详细了解有关 notebook 或一些库的具体话题, 可以使用菜单栏右侧的帮助菜单。

下方的主要区域, 由被称为单元格的部分组成。每个 notebook 由多个单元格构成, 而每个单元格又可以有不同的用途。

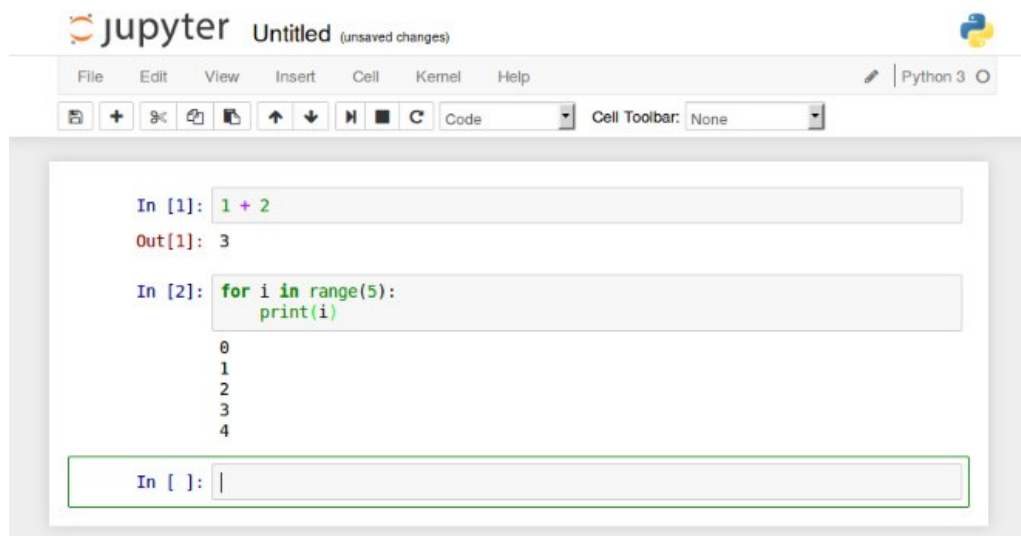
下方截图中看到的是一个代码单元格 (code cell), 以 `[]` 开头。在这种类型的单元格中, 可以输入任意代码并执行。例如, 输入 `1 + 2` 并按下 `Shift + Enter`。之后, 单元格中的代码就会被计算, 光标也会被移动到新的单元格中。你会得到如下结果:



根据绿色边框线，我们可以轻松地识别出当前工作的单元格。接下来，我们在第二个单元格中输入些其他代码，例如：

```
for i in range(5):  
    print(i)
```

对上面的代码求值时，你会得到：



和前一个示例一样，代码被计算之后，马上就会显示结果。你应该注意到了，这次没有出现类似Out[2]这样的文字。这是因为我们将结果打印出来了，没有返回任何的值。

notebook 有一个非常有趣的特性，就是可以修改之前的单元格，对其重新计算，这样就可以更新整个文档了。试着把光标移回第一个单元格，并将1 + 2修改成2 + 3，然后按下Shift + Enter重新计算该单元格。你会发现结果马上就更新成了 5。如果你不想重新运行整个脚本，只想用不同的参数测试某个程式的话，这个特性显得尤其强大。不过，你也可以重新计算整个 notebook，只要点击Cell -> Run all即可。

现在我们已经知道了如何输入代码，为什么不尝试着让这个 notebook 更加漂亮、内容更丰富？为此，我们需要使用其他类型的单元格，即 Header单元格和 Markdown单元格。

首先，我们在顶部添加一个 notebook 的标题。选中第一个单元格，然后点击 Insert -> Insert单元格above（在上方插入单元格）。你会发现，文档的顶部马上就出现了一个新的单元格。点击在快捷键栏中的单元格类型，将其变成一个标题单元格（heading cell）：



选中下拉选项中的 Heading。然后会出现一个弹出消息，告诉你如何创建不同层级的标题，这样你就有了一个不同类型的 cell：



这个单元格以#标记开头，意味着这是一个一级标题。如果需要子标题，可以使用以下标记表

示（改变单元格类型时弹出消息中有解释）：

#：一级标题

##：二级标题

###：三级标题

...

在#之后写下文档的标题，然后计算该单元格。你会发现一个样式非常好看的标题。作为示例和练习，我还添加了其他几个标题单元格：

My first title in Jupyter

A very simple operation

```
In [1]: 1 + 2
```

```
Out[1]: 3
```

Counter

```
In [2]: for i in range(5):  
        print(i)
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
In [ ]:
```

添加好标题之后，我们在编写一些解释，介绍每个代码单元格中的情况。为此，我们要在相应的地方插入单元格，然后将其类型变成 Markdown。然后，计算新的单元格。就这样，你的解释文本就漂亮地渲染出来了！

My first title in Jupyter

A very simple operation

Let's **add** two numbers:

```
In [1]: 1 + 2
```

```
Out[1]: 3
```

Counter

Let's *count* from 0 to 4:

```
In [2]: for i in range(5):  
        print(i)
```

```
0  
1  
2  
3  
4
```

```
In [ ]:
```

最后，你可以重命名该 notebook，点击 **File** -> **Rename**，然后输入新的名称。这样，新的名称将会出现在窗口的左上角，在 Jupyter 的标志旁边。

单元格操作

高级单元格操作，将让编写 notebook 变得更加方便。举例如下：

- 如果想删除某个单元格，可以选择该单元格，然后依次点击 **Edit** -> **Delete Cell**；
- 如果想移动某个单元格，只需要依次点击 **Edit** -> **Move cell** [up | down]；
- 如果想剪贴某个单元测，可以先点击 **Edit** -> **Cut Cell**，然后在点击 **Edit** -> **Paste Cell** [Above | Below]；
- 如果你的 notebook 中有很多单元格只需要执行一次，或者想一次性执行大段代码，那么可以选择合并这些单元格。点击 **Edit** -> **Merge Cell** [Above | below]。

记住这些操作，它们可以帮助你节省许多时间。

Markdown 单元格高级用法

我们再来看看 Markdown 单元格。虽然它的类型是 markdown，但是这类单元格也接受 HTML 代码。这样，你就可以在单元格类实现更加丰富的样式，添加图片，等等。例如，如果想在 notebook 中添加 Jupyter 的 logo，将其大小设置为 100px x 100px，并且放置在单元格左侧，可以这样编写：

```

```

计算该单元格之后，会出现这样的结果：



In []:

另外，markdown 单元格还支持 LaTeX 语法。例如：

$$\int_0^{+\infty} x^2 dx$$

计算上述单元格，将获得下面的 LaTeX 方程式：

$$\int_0^{+\infty} x^2 dx$$

In []:

导出功能

notebook 还有一个强大的特性，就是其导出功能。可以将 notebook 导出为多种格式：

- HTML
- Markdown
- ReST
- PDF (通过 LaTeX)
- Raw Python

导出 PDF 功能，可以让你不用写 LaTeX 即可创建漂亮的 PDF 文档。你还可以将 notebook 作为网页发布在你的网站上。甚至，你可以导出为 ReST 格式，作为软件库的文档。

Matplotlib 集成

如果你用 Python 绘制过图形，那你肯定知道 matplotlib。Matplotlib 是一个用于创建漂亮图形的 Python 库，结合 Jupyter notebook 使用时体验更佳。

要想在 Jupyter notebook 中使用 matplotlib，需要告诉 Jupyter 获取 matplotlib 生成的所有图形，并将其嵌入 notebook 中。为此，需要计算：

```
%matplotlib inline
```

译注：要想执行成功，需要先 `pip install matplotlib`。

运行这个指令可能要花个几秒钟，但是在 notebook 中需要执行一次即可。接下来，我们来绘制一个图形，看看具体的集成效果：

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

```
x = np.arange(20)
y = x**2
```

```
plt.plot(x, y)
```

上面的代码将绘制方程式 $y=x^2$ 。计算单元格后，会得到如下图形：

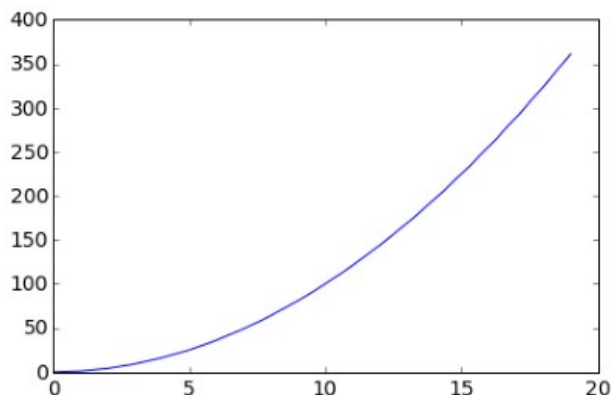
```
In [1]: %matplotlib inline
```

```
In [2]: import matplotlib.pyplot as plt
import numpy as np

x = np.arange(20)
y = x**2

plt.plot(x, y)
```

```
Out[2]: [<matplotlib.lines.Line2D at 0x7f50cd73a588>]
```



我们看到，绘制出的图形直接添加在了 notebook 中，就在代码的下面。我们可以之后修改代码，重新计算，这时图形也会动态更新。这是每个数据科学家都想要的一个特性：将代码和图片放在同一个文件中，清楚地看出每段代码的效果。

非本地内核

我们可以非常容易地在一台电脑上启动 Jupyter，而且支持多人通过网络连接同一个 Jupyter 实例。在上一篇文章中，你有没有注意启动 Jupyter 时出现过这样一段话：

The IPython Notebook is running at: <http://localhost:8888/>

这意味着，你的 notebook 是本地运行的，可以在浏览器上打开 <http://localhost:8888/>，从而访问 notebook。你也可以修改下配置，让该 notebook 可以被公开访问。这样，任何知道 notebook 地址的人都可以连接到 notebook 进行远程修改。

结语

从这两篇快速入门介绍中，我们可以看到：Jupyter notebook 是一个非常强大的工具，可以创建漂亮的交互式文档，制作教学材料，等等。建议你马上开始使用 Jupyter notebook，探索更多 notebook 的强大功能。

链接来自：<https://www.packtpub.com/books/content/getting-started-jupyter-notebook-part-1>

即使只是凡世中一颗小小的尘埃，命运也要由自己主宰，像向日葵般，迎向阳光、勇敢盛开

分类: [python](#)

标签: [python](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

[Little_Rookie](#)

关注 - 1

粉丝 - 276

[+加关注](#)

« [上一篇：R语言 | 数据特征分析](#)

» [下一篇：P2P贷款全攻略,贷前、贷中、贷后工作事项解析](#)

posted @ 2017-03-17 15:18 Little_Rookie 阅读(181931) 评论(7) 编辑 收藏

评论列表

#1楼 2017-07-26 15:43 cloud&ken

博主写的非常不错！~学到了~！

[支持\(0\)](#) [反对\(0\)](#)

#2楼 2017-11-15 09:24 mai_sui

感谢楼主分享~

[支持\(0\)](#) [反对\(0\)](#)

#3楼 2018-01-30 19:48 缪阿布

numpy 第一帖，谢谢！

[支持\(0\)](#) [反对\(0\)](#)

#4楼 2018-03-29 21:00 青青园中葵

小白表示很容易理解上手，感谢！

[支持\(0\)](#) [反对\(0\)](#)

#5楼 2018-04-28 09:39 张金棒

写的真好

[支持\(0\)](#) [反对\(0\)](#)

#6楼 2018-06-26 19:37 064曾志彬

怎样将标题去掉#这个符号，将蓝色变为黑色，调整字体大小

[支持\(0\)](#) [反对\(0\)](#)

#7楼 2018-11-16 09:48 姜小豆

@ 064曾志彬

运行即可

[支持\(0\)](#) [反对\(0\)](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，访问网站首页。

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

Copyright ©2019 Little_Rookie