0_BemerkungenZumAnfang

September 20, 2022

Hier findest du in kurzer Form:

- Was ist ein (Jupyter-)Notebook?
- Hinweise zur Installation
- Bemerkungen zu den Zelltypen
- Anmerkungen zu Besonderheiten im Umgang mit Jupyter-Notebooks

1 Was ist ein Notebook?

Ein **Notebook** ist (nach Wikipedia) ein Notizbuch, das für viele Zwecke verwendet wird, z.B. zum Aufzeichnen von Notizen, zum Schreiben von Texten oder zum Darstellen von Graphiken.

Ein Notizbuch wird verwendet, um eine Idee oder eine wichtige Information mithilfe von Texten, Diagrammen, Zeichnungen, Bildern, Gleichungen, Tabellen auszudrücken. Und das ist auch - in digitaler Form - der Zweck von Jupyter-Notebooks

2 Kurze Historie

Auszug aus:

© Springer Nature Limited Nature 589, S. 344–348, 2021

Jeffrey M. Perkel (Technologie-Redakteur bei »Nature«.)

Fernando Pérez war 2001 ein Doktorand »auf der Suche nach Ablenkung«, als er beschloss, sich mit einer Kernkomponente von Python, einer weit verbreiteten Programmiersprache, zu beschäftigen. Häufig griff man dabei auf ein System namens REPL (Read-Evaluate-Print-Loop) zurück: Man gibt seinen Code ein, das Programm führt ihn Zeile für Zeile aus und liefert ein Ergebnis. Das funktionierte zwar zuverlässig und schnell, aber Pérez fiel auf, dass Python nicht für das wissenschaftliche Arbeiten entwickelt worden war. Zum Beispiel war es nicht möglich, bestimmte Code-Module auf einfache Weise vorzuladen oder Datenvisualisierungen offen zu halten. Deshalb entschied Perez, selbst eine solche Version zu schreiben.

Das Ergebnis war IPython, ein interaktives Programmpaket zum Entwickeln und Ausführen von Pythonprogrammen, das Pérez im Dezember 2001 vorstellte – und das aus lediglich 259 Zeilen bestand. Zusammen mit dem Physiker Brian Granger und dem Mathematiker Evan Patterson zog er das System zehn Jahre später auf den Webbrowser um, brachte das IPython Notebook auf den Markt und löste damit eine Revolution in der Datenwissenschaft aus.

Computergestützte Notebooks bestehen nicht nur aus Programmcodes, sondern vereinen auch Texte und Grafiken in einem Dokument. Im Gegensatz zu anderen Systemen dieser Art (wie die Software

Mathematica oder Maple) war IPython quelloffen und erlaubte Beiträge anderer Entwickler. Außerdem unterstützte es die Programmiersprache Python, die unter Wissenschaftlern sehr beliebt ist.

2014 entwickelte sich IPython zu Jupyter, das inzwischen mehr als 100 Programmiersprachen zulässt und es den Benutzern ermöglicht, Daten auf entfernten Supercomputern ebenso einfach zu verarbeiten wie auf ihren eigenen Laptops. »Für Datenwissenschaftler hat sich Jupyter zu einem Standardwerkzeug entwickelt«, urteilte Nature 2018. Damals gab es 2,5 Millionen Jupyter-Notebooks auf der Code-Sharing-Plattform GitHub; inzwischen hat sich die Anzahl vervierfacht. Darunter befinden sich auch jene Notebooks, die 2016 die Entdeckung von Gravitationswellen und 2019 die Abbildung eines Schwarzen Lochs dokumentieren. »Dass wir einen kleinen Beitrag zu diesen bahnbrechenden Arbeiten geleistet haben, macht uns sehr stolz«, sagt Pérez.

3 Was ist ein *Jupyter-Notebook?*

Ein **Jupyter-Notebook** ist in diesem Sinne tatsächlich ein Notizbuch; denn es wird benutzt, um Texte, Programm-Codes, Bilder, Gleichungen, Diagramme, Tabellen, Visualisierungen zu erzeugen und darzustellen.

Jupyter Notebook ist eine Open-Source-Webanwendung, mit der man Dokumente erstellen und freigeben kann. Es enthält Programm-Code, Visualisierungen Bilder, Text...

Um Jupyter-Notebooks benutzen zu können, ist ein Jupyter-System notwendig.

4 Das *Jupyter-System*

Project Jupyter ist eine Non-Profit-Organisation, die gegründet wurde, um *Open-Source-Software, offene Standards und Services für interaktives Arbeiten mit Dutzenden Programmiersprachen zu entwickeln.*

Das Jupyter-System unterstützt über 100 Programmiersprachen (im Jupyter-System als ***Kernel*** bezeichnet), darunter *Python*, *Java*, *R*, *Julia*, *Matlab* und viele mehr.

Am häufigsten wird das Jupyter-Notizbuch mit einem Python-Kernel verwendet.

5 Externes *Jupyter-System* nutzen

Um Jupyter-Notebooks zu benutzen, können Sie den folgenden Link nutzen:

Jupyter-Server

Sie können dabei temporäre Kopien von Notebooks sehen und bearbeiten.

6 *Anaconda* als lokales Jupyter-System

Um Jupyter-Notebooks benutzen zu können, kann man z.B. ***Anaconda*** installieren.

Auf der Seite *Anaconda* findet man die zu Ihrem Betriebssystem passende Version und eine Installationsanweisung.

Nach erfolgreicher Installation kann man den Jupyter-Notebook-Server starten, so dass eigene oder fremde Notebooks geladen und neue Notebooks erzeugt werden können.

7 Zellen in einem Jupyter-Notebook

Jedes Jupyter-Notebook besteht aus Zellen. Es gibt zwei (für diesen Zusammenhang interessierende) Zelltypen:

- Code-Zellen. Man erkennt sie an `In [nummer]:` am linken Rand neben der Zelle.
  - In einer solchen Zelle kann man Python-Code eingeben und ausführen lassen. Mehr dazu später.
- Text-Zellen.
  - In diese Zellen kann man Texte, Bilder, Tabellen, Links, Aufzählungen, ... aufnehmen. Dazu benutzt man die Auszeichnungssprache ***MarkDown***, eine Art *Simple-HTML*. Bitte informieren Sie sich bei Bedarf im Internet, z.B. auf den Seiten MarkDown.

8 Python-Programme in Notebooks

Ein Python-Programm beinhaltet in der Regel alle Definitionen, die notwendig sind, sowie den Code des *Hauptprogramms*. Externe Funktionen können über Importe eingebunden werden.

In einem Jupyter-Notebook sucht man (zunächst) vergebens nach einem Hauptprogramm. Man findet stattdessen (neben Zellen, in denen Texte oder auch Bilder zu finden sind) viele **Eingabezellen**, in denen (einzeiliger oder mehrzeiliger) Python-Code steht.

Jede Eingabezelle ist (sofern syntaktisch korrekt) eine ausführbare Python-Anweisung. Also kann man jede Eingabezelle manuell ausführen (Cursor in die Zelle bewegen und per abschließen), und die Reaktion des Systems beobachten. Falls keine Fehlermeldung erfolgt, wurde die Python-Anweisung ausgeführt:

- Falls die Zelle eine Import-Anweisung enthalten hat, wurde diese Bibliotheke importiert. Dann kann man in anderen Zellen die Funktionen aus dieser Bibliothek nutzen. Der Kernel dieses Notebooks hat diese Bibliothek integriert.
- Falls in der Zelle eine Variablenzuweisung erfolgte, wird diese Variablenbelegung dauerhaft in dem Kernel aufbewahrt. Diese Variable existiert dann weiterhin ab jetzt in allen anderen Zellen.

Dieses Verhalten ist zunächst ungewohnt, da die Sichtbarkeit einer Variablen oder einer importierten Bibliotheksfunktion dauerhaft in dem aktiven Kernel dieses Notebooks ist, also auch in Zellen *oberhalb* derjenigen, in der die Aktivierung erfolgte.

Falls man also die folgende Zelle aktiviert, erfolgt zunächst eine Fehlermeldung (entfernen Sie zuvor die Kommentar-Marke):

[ ]:
```
# print (x)
```

Jetzt aktivieren Sie die folgende Zelle:

[ ]:
```
x = 42
print (x)
```

Wenn Sie jetzt die erstgenannte Zelle erneut aktivieren, erfolgt **keine(!)** Fehlermeldung mehr.

Starten Sie den Kernel erneut, ist die Variablenbindung von `x` aufgehoben.

Tipp:

- Beim Erstellen eines Notebooks sollte man die Reihenfolge der Zellen so vornehmen, dass ein sequenzieller Durchlauf zum erwünschten Ergebnis führt.
- Als Benutzer eins (fremden) Notebooks ist es ratsam, die Zellen sequenziell zu aktivieren. Falls man Änderungen vorgenommen hat (z.B. Variablen neu belegt), kann man den Kernel erneut initialisieren und starten (auf Wunsch mit oder ohne automatische Zellenaktivierungen).

Jetzt wünsche ich Ihnen viel Spaß beim Arbeiten mit Python und Jupyter-Notebooks.

[ ]: