MathBook

Simon Kamerling

Nov 27, 2020

CONTENTS

This is a small sample book to give you a feel for how book content is structured.

Check out the content pages bundled with this sample book to get started.

CHAPTER

ONE

PLOTTER

Nous allons ici plotter via plotly et cufflinks, qui permettent une visualisation rapide et efficace d'un certain nombre de données.

Il est nécessaire de se renseigner sur pandas et les DataFrame pour obtenir les fonctions de manipulation les plus pratiques.

La méthode implémentée ici est de créer un DataFrame , puis d'appeler dessus la méthode .iplot() (implémentée dans cufflinks). Une autre méthode est d'utiliser directement plotly en rajoutant différents layout sur une figure.

Il y a un certain nombre d'options dans iplot(); par exemple:

- y = ['mdot(h)'] -> choisit la fonction en y, possibilité de mettre plusieurs clés
- secondary_y -> pareil mais pour le deuxième axe
- kind = 'box' -> type de graphiques. Une courbe simple: 'scatter'; beaucoup de styles sont implémentés

De nombreux paramètres sont accessibles via help(cf.iplot)

```
import pandas as pd
from ipywidgets import interact, interactive, fixed, interact_manual, IntSlider
# Standard plotly imports
import chart_studio.plotly as py
import plotly.graph_objs as go
from plotly.offline import iplot, init_notebook_mode
# Using plotly + cufflinks in offline mode
import cufflinks as cf
cf.go_offline(connected=False)
init_notebook_mode(connected=False)
```

Exo 1 : Pour n entier naturel non nul, on pose $H_n = \sum_{k=1}^{n} \frac{1}{k}$ (série harmonique).

1. Montrer que : $\forall n \in \mathbb{N}, \ln(n+1) < H_n < 1 + \ln(n)$ et en déduire la limite en $+\infty$ de H_n.
2. Pour n entier naturel non nul, on pose $u_n = H_n ln(n)$ et $v_n = H_n ln(n+1)$. Montrer que les suites (u_n) et (v_n) convergent vers un réel .

```
from math import log
nMax=100
logN=[1+log(n) for n in range(1,nMax+1)]
logNPlus1=[log(n+1) for n in range(1,nMax+1)]

Hn=[1./1.]

for k in range(2,nMax+1):
    Hn.append(Hn[-1]+1./k)
#Pour créer le DataFrame, il faut lui donner une matrice de la bonne taille
```

(continues on next page)

(continued from previous page)

```
data=[Hn,logN,logNPlus1]
#il faut transposer le dataFrame (pas dans le bon sens)
df=pd.DataFrame(data=data).T
# df.describe() : affiche une description du DataFrame
df.columns=["$H_n$","1+ln(n)","ln(n+1)"]
df.iplot(kind='scatter',title='Différentes suites')
```

```
import plotly.io as pio
import plotly.express as px
import plotly.offline as py

df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length", color="species", size="sepal_
↪length")
fig
```

CHAPTER

TWO

MARKDOWN FILES

Whether you write your book's content in Jupyter Notebooks (`.ipynb`) or in regular markdown files (`.md`), you'll write in the same flavor of markdown called **MyST Markdown**.

2.1 What is MyST?

MyST stands for "Markedly Structured Text". It is a slight variation on a flavor of markdown called "CommonMark" markdown, with small syntax extensions to allow you to write **roles** and **directives** in the Sphinx ecosystem.

2.2 What are roles and directives?

Roles and directives are two of the most powerful tools in Jupyter Book. They are kind of like functions, but written in a markup language. They both serve a similar purpose, but **roles are written in one line**, whereas **directives span many lines**. They both accept different kinds of inputs, and what they do with those inputs depends on the specific role or directive that is being called.

2.2.1 Using a directive

At its simplest, you can insert a directive into your book's content like so:

```
```{mydirectivename}
My directive content
```
```

This will only work if a directive with name `mydirectivename` already exists (which it doesn't). There are many pre-defined directives associated with Jupyter Book. For example, to insert a note box into your content, you can use the following directive:

```
```{note}
Here is a note
```
```

This results in:

Note: Here is a note

In your built book.

For more information on writing directives, see the MyST documentation.

2.2.2 Using a role

Roles are very similar to directives, but they are less-complex and written entirely on one line. You can insert a role into your book's content with this pattern:

```
Some content {rolename}`and here is my role's content!`
```

Again, roles will only work if `rolename` is a valid role's name. For example, the `doc` role can be used to refer to another page in your book. You can refer directly to another page by its relative path. For example, the role syntax `{doc}`intro`` will result in: intro.

For more information on writing roles, see the MyST documentation.

2.2.3 Adding a citation

You can also cite references that are stored in a `bibtex` file. For example, the following syntax: `{cite}`holdgraf_evidence_2014`` will render like this: [holdgraf_evidence_2014].

Moreoever, you can insert a bibliography into your page with this syntax: The `{bibliography}` directive must be used for all the `{cite}` roles to render properly. For example, if the references for your book are stored in `references.bib`, then the bibliography is inserted with:

````
```{bibliography} references.bib
```
````

Resulting in a rendered bibliography that looks like:

2.2.4 Executing code in your markdown files

If you'd like to include computational content inside these markdown files, you can use MyST Markdown to define cells that will be executed when your book is built. Jupyter Book uses *jupytext* to do this.

First, add Jupytext metadata to the file. For example, to add Jupytext metadata to this markdown page, run this command:

```
jupyter-book myst init markdown.md
```

Once a markdown file has Jupytext metadata in it, you can add the following directive to run the code at build time:

````
```{code-cell}
print("Here is some code to execute")
```
````

When your book is built, the contents of any `{code-cell}` blocks will be executed with your default Jupyter kernel, and their outputs will be displayed in-line with the rest of your content.

For more information about executing computational content with Jupyter Book, see The MyST-NB documentation.