td2a_correction_session_2C

January 1, 2023

1 2A.soft - IPython et commandes magiques - correction

Aperçu des commandes magiques pour automatiser un peu plus les tâches courantes.

[1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

```
Populating the interactive namespace from numpy and matplotlib
```

[2]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

[2]:
```
<IPython.core.display.HTML object>
```

1.0.1 Exercice 1 : votre propre magique commande

On implémente la commande `%monplot` qui prend un fichier texte, appelle la commande plot et stocke le résultat dans la variable `df` qui est ajoutée à l'espace de travail.

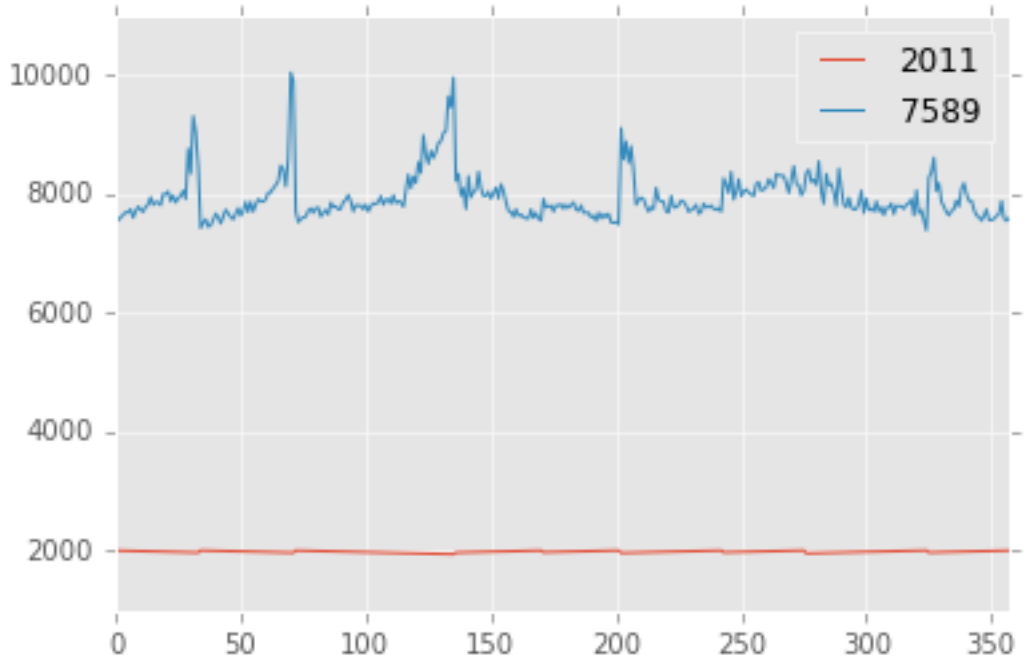
[3]:
```
from IPython.core.magic import Magics, magics_class, line_magic, cell_magic,␣
 ↪line_cell_magic
import pandas

@magics_class
class CustomMagics(Magics):

    @line_magic
    def monplot(self, line):
        df = pandas.read_csv(line, sep="\t")
        pl = df.plot()
        if self.shell is not None:
            self.shell.user_ns["df"] = df

ip = get_ipython()
ip.register_magics(CustomMagics)
```

[4]:
```
import pyensae.datasource
pyensae.datasource.download_data("marathon.txt")
```

```
    downloading of
http://www.xavierdupre.fr/enseignement/complements/marathon.txt  to
marathon.txt
```

[4]: 'marathon.txt'

[5]:
```
%monplot marathon.txt
```



Sans prétraitement, cela n'a pas toujours un sens. Mais cela devient pratique dès qu'il faut automatiser la création d'un rapport à partir de données qui se présentent toujours sous le même format. D'une manière générale, cela revient à réduire la syntaxe des tâches répétitives.

[6]: