wikipedia_stats_enonce

May 17, 2019

1 Statistiques Wikipedia - énoncé

On s'instéresse aux statistiques de consultations de Wikipédia : pageviews. Ce TD commence par récupération des données avant de s'intéresser aux séries temporelles.

```
In [1]: from jyquickhelper import add_notebook_menu
        add_notebook_menu()
```

```
Out[1]: <IPython.core.display.HTML object>
```

1.1 Récupération des données

Les statistiques sont disponibles pour chaque heure et chaque jour. Compressés, cela représente environ 60Mo. On regarde un fichier.

```
In [2]: import os
        folder = "wikipv"
        if not os.path.exists(folder):
            os.mkdir(folder)
```

```
In [3]: from mlstatpy.data.wikipedia import download_pageviews
        import os
        from datetime import datetime

        %timeit -n1 -r1 download_pageviews(datetime(2016,9,1), folder=folder)
```

```
1 loop, best of 1: 42.6 s per loop
```

```
In [4]: %load_ext pyensae
```

```
In [5]: %head wikipv/pageviews-20160901-000000
```

```
Out[5]: <IPython.core.display.HTML object>
```

```
In [6]: os.stat("wikipv/pageviews-20160901-000000").st_size / 2**20, "Mo"
```

```
Out[6]: (150.17064571380615, 'Mo')
```

Ca va prend un petit peu de temps et d'espace de télécharger ces données.

1.2 Exercice 1 : parallélisation du téléchargement

Regarde le module multiprocessing et implémenter une version parallélisée du programme suivant. multiprocessing est la librairie standard mais il en existe beaucoup d'autres : ParallelProcessing, joblib.

```
In [7]: from mlstatpy.data.wikipedia import download_pageviews
        from datetime import datetime
        folder = "wikipv"

        for h in range(0, 24):  # boucle sur les 24 heures de la journée
            dt = datetime(2016,9,1,h)
            print("téléchargement", dt, "début", datetime.now())
            download_pageviews(dt, folder=folder)
```

```
téléchargement 2016-09-01 00:00:00 début 2016-09-11 21:32:18.373115
téléchargement 2016-09-01 01:00:00 début 2016-09-11 21:32:18.374114
téléchargement 2016-09-01 02:00:00 début 2016-09-11 21:32:18.374114
téléchargement 2016-09-01 03:00:00 début 2016-09-11 21:32:18.374114
téléchargement 2016-09-01 04:00:00 début 2016-09-11 21:32:18.375114
téléchargement 2016-09-01 05:00:00 début 2016-09-11 21:32:18.375114
téléchargement 2016-09-01 06:00:00 début 2016-09-11 21:32:18.375114
téléchargement 2016-09-01 07:00:00 début 2016-09-11 21:32:18.375114
téléchargement 2016-09-01 08:00:00 début 2016-09-11 21:32:18.375114
téléchargement 2016-09-01 09:00:00 début 2016-09-11 21:32:18.376113
téléchargement 2016-09-01 10:00:00 début 2016-09-11 21:32:18.376113
téléchargement 2016-09-01 11:00:00 début 2016-09-11 21:32:58.745096
téléchargement 2016-09-01 12:00:00 début 2016-09-11 21:34:08.073304
téléchargement 2016-09-01 13:00:00 début 2016-09-11 21:35:04.923348
téléchargement 2016-09-01 14:00:00 début 2016-09-11 21:36:10.377303
téléchargement 2016-09-01 15:00:00 début 2016-09-11 21:37:20.523141
téléchargement 2016-09-01 16:00:00 début 2016-09-11 21:38:21.088853
téléchargement 2016-09-01 17:00:00 début 2016-09-11 21:39:24.186874
téléchargement 2016-09-01 18:00:00 début 2016-09-11 21:40:11.545482
téléchargement 2016-09-01 19:00:00 début 2016-09-11 21:41:05.327336
téléchargement 2016-09-01 20:00:00 début 2016-09-11 21:41:56.814023
téléchargement 2016-09-01 21:00:00 début 2016-09-11 21:42:50.729708
téléchargement 2016-09-01 22:00:00 début 2016-09-11 21:43:49.187079
téléchargement 2016-09-01 23:00:00 début 2016-09-11 21:44:48.095661
```

1.3 Exercice 2 : statistiques

On veut comparer les habitudes de lectures des utilisateurs pour différents types de pages, politique, musique, cinéma, science, littérature… On prendra une semaine quelconque comme période d'étude. Que proposez-vous ?

```
In [8]:
```