

Ajuster le format des notebooks Jupyter

Pierre MASSÉ

13 janvier 2020

1 Objectif

L'objectif de ce mode opératoire est de permettre de documenter la configuration à amener pour customizer l'export des notebooks en pdf. Cela permet, entre autres, de réduire la taille de la police utilisée, en entrée et en sortie, tout en conservant la mise en forme propre aux notebooks.

2 Création du template

L'essentiel de l'ajustement se situe au niveau du template utilisé par jupyter nbconvert. Il est nécessaire de construire un nouveau template permettant de modifier la taille de la police utilisée, retirer la numérotation des pages (pour inclusion dans un autre pdf, qui a sa propre numérotation).

```
((- extends 'article.tplx' -*))           % extension du modèle de base,
                                           % qui rend bien

((* set charlim = 108 *))                 % la limite de caractères pour
                                           % les lignes de texte
                                           % dans les cellules de sortie

((* block definitions *))
  ((( super() )))
  \author{Pierre \textsc{Massé}}          % définition de l'auteur
((* endblock definitions *))

((* block docclass *))
  \documentclass{article}                 % utilisation du type de document
                                           % standard Latex
((* endblock docclass *))

((* block predoc *))
  ((* block maketitle *))
    ((( super() )))                      % On garde le block maketitle tel
                                           % quel.
  ((* endblock maketitle *))
```

```

\pagestyle{empty}                % retrait de la numérotation des
                                  % pages
\thispagestyle{empty}            % bizarrement, nécessaire pour la
                                  % première page, qui reste numérotée
                                  % sinon.
\footnotesize                    % ajustement de la taille de la police
((* endblock predoc *))

```

Attention, bien penser à retirer les commentaires (après #), qui ne sont pas autorisés dans un template Jinja. Enregistrer ce fichier dans le répertoire :

```
~/jupyter/templates/
```

Sur mon PC Windows, il s'agit de :

```
C:\Users\pmasse\jupyter\templates\
```

Utiliser l'extension « .tplx ». J'ai nommé mon fichier « article1.tplx ». Ce template est utilisable de manière simple en ligne de commande, en utilisant la ligne de commande suivante :

```
jupyter nbconvert --to <pdf ou latex> --template article1 <nom du notebook à convertir>
```

3 Configuration des apps jupyter

Cette étape permet d'utiliser le template défini ci-dessus par défaut sans avoir à le spécifier à l'appel de nbconvert en ligne de commande, et de pouvoir l'utiliser directement depuis le notebook Jupyter (File > Download as > PDF via Latex). Créer (ou amender s'ils existent déjà) deux fichiers de configuration dans le répertoire :

```
~/jupyter/
```

nommés :

- jupyter_nbconvert_config.py : pour l'utilisation en ligne de commande de nbconvert
- jupyter_notebook_config.py : pour l'export directement depuis le notebook

Y ajouter le contenu suivant :

```

import os
c.LatexExporter.template_path.append(os.path.expanduser('~/.jupyter/templates'))
c.LatexExporter.template_file = 'article1.tplx'

```

4 Ajustement à amener dans le notebook

Dans le notebook Jupyter, pour que la mise en forme des dataframes pandas soit correctement prise en compte, ajouter la ligne suivante (après avoir importé pandas en tant que pd) :

```
pd.options.display.width=108
```

Cela pourra être modifié si jamais je trouve un meilleur moyen d'afficher les dataframes dans le pdf.