

pandas

August 22, 2019

O **Pandas** é um pacote que possibilita o manejo dos dados.

{: .alert .alert-danger} **Aviso:** Esse *post* tem a finalidade de mostrar os comandos básicos e me deixar com uma “cola” rápida para meu uso cotidiano. Todas os códigos são exemplificativos e podem/devem ser alterados, indicando o nome dos arquivos e diretórios corretamente.

{: .box-note} **Nota:** É possível acessar esse *post* em **formato .pdf** e, ainda, no **repositório do GitHub**.

1 Importando Bibliotecas

As bibliotecas básicas, ou *packages*, necessárias para criação do mapa são: - O **Pandas**, que tem a missão de trabalhar com dados, criar *subsets*, selecionar e filtros dados e; - O **Geopandas**, que é a biblioteca lê geojson!

```
[3]: import pandas as pd
```

```
[4]: # Lendo e filtrando dados
empresas = pd.read_csv('data/empresas.xz')
empresas = empresas[empresas['state'] == 'SP']
empresas = empresas[empresas['city'] == 'SANTOS']

empresas.dtypes
```

```
[4]: name                object
      situation          object
      neighborhood      object
      address           object
      number            object
      zip_code          object
      city              object
      state             object
      cnpj              object
      status            object
      additional_address_details  object
      main_activity      object
      latitude          float64
      longitude         float64
```

dtype: object

2 GeoPandas

```
[14]: #
```

3 Exportando o *Jupyter Notebook* para outros formatos

O arquivo *.ipynb* pode ser exportado em formatos diversos. Abaixo carrego uma função que escrevi para facilitar o processo de exportação do arquivo em diferentes locais do PC para, posteriormente, atualizar os repositórios contidos no *GitHub*.

<https://github.com/michelmetran>

```
[5]: # %load '~/Documents/SourceCode/codes/files/export_jupyter.py'
def export_jupyter(path, extensions=['html', 'markdown', 'latex', 'pdf',
    ↪ 'python'], today=True):
    """
    Export .ipynb file to others formats
    :return: File in other formats
    """

    # Import Packages
    import os
    import datetime

    # Data
    timestamp = datetime.datetime.now()
    srt_today = (str(timestamp.year) + '-' +
                  str(f"{timestamp.month:02d}") + '-' +
                  str(f"{timestamp.day:02d}"))

    # Extensions
    for extension in extensions:
        if today==True:
            os.system('jupyter nbconvert --to {} {} --output {}'.
                        format(extension, get_jupyternotebook_name(),
                               os.path.join(path,
    ↪ srt_today+'-'+get_jupyternotebook_name().split('.')[0]))
                        print('Arquivo {} exportado corretamente para o formato {} usando
    ↪ prefixo da data.'.
                              format(get_jupyternotebook_name(), extension))
```

```

else:
    os.system('jupyter nbconvert --to {} {} --output {}'.
              format(extension, get_jupyternotebook_name(),
                      os.path.join(path, get_jupyternotebook_name()).
↳split('.')[0]))
    print('Arquivo {} exportado corretamente para o formato {} sem usar_
↳prefixo da data.'.
          format(get_jupyternotebook_name(), extension))

```

```

[11]: # %load '~/Documents/SourceCode/codes/files/get_jupyternotebook_name.py'
def get_jupyternotebook_name():
    """
    Returns the name of the current notebook as a string
    From https://mail.scipy.org/pipermail/ipython-dev/2014-June/014096.html
    :return: Returns the name of the current notebook as a string
    """
    # Import Packages
    from IPython.core.display import Javascript
    from IPython.display import display

    display(Javascript('IPython.notebook.kernel.execute("theNotebook = " + \
    "\""+IPython.notebook.notebook_name+"\"");'))

    # Result
    return theNotebook

```

Com as funções para exportar o *Jupyter Notebook* e para obter o nome do arquivo *.ipynb* carregadas, basta exportar o arquivo, inicialmente para a pasta *docs* dentro do projeto e também, visando atualizar os *posts* do site, para a respectiva pasta.

```

[12]: export_jupyter('docs', ['pdf'], False)
export_jupyter('/home/michel/Documents/SourceCode/michelmetran.github.io/
↳_posts', ['markdown'], True)

```

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

Arquivo *pandas.ipynb* exportado corretamente para o formato *pdf* sem usar prefixo da data.

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

Arquivo pandas.ipynb exportado corretamente para o formato markdown usando prefixo da data.

4 Atualizando Repositório do Projeto e do *site*

Após as exportações dos arquivos nos formatos necessários, basta atualizar o repositório diretamente pelo *Jupyter Notebook*. Abaixo é atualizado o repositório desse projeto específico, bem como a derivação desse projeto no *site*.

```
[13]: %run '~/Documents/SourceCode/codes/git/update_github.py'
```

```
[25]: git_full('/home/michel/Documents/SourceCode/package_pandas', '.', 'Atualizando')
      git_full('/home/michel/Documents/SourceCode/michelmetran.github.io', '.', '
      ↪Atualizando')
```

```
b'' b''
b'[master 2f8e4e6] Atualizando\n 1 file changed, 1 insertion(+), 1
deletion(-)\n' b''
b'' b'To github.com:michelmetran/package_pandas.git\n    477c79e..2f8e4e6  master
-> master\n'
Done!!
b'' b''
b'"On branch master\nYour branch is up to date with 'origin/master'.\n\nnothing
to commit, working tree clean\n" b''
b'' b'Everything up-to-date\n'
Done!!
```