ScrapingBooks

January 25, 2023

0.1 1) Objetivo

- Fazer raspagem de dados do site que feito exatamente para praticar webscraping (link: https://books.toscrape.com/) e obter informações relacionadas ao nome do livro, preço, avaliação, quantidade disponível em estoque, categoria, etc, navegando entre várias páginas.
- São informações fictícias, apenas para praticar extração dos dados a partir de tags e ETL.
- Projeto utilizado para servir como benchmarking para outros (pessoais ou profissionais), tais como: coletar dados de algum produto em diversos marketplaces e escolher o melhor para comprar com base nos filtros que melhor me atenderem.

0.2 2) Requirements

- Segue um arquivo com nome de requirements.txt na pasta principal com as bibliotecas e versões que utilizei para rodar os códigos abaixo (Python utilizado foi 3.9+).
- Para instalar basta executar o mesmo padrão "pip install -r requirements.txt" no terminal (sem aspas).

Segue a lista de bibliotecas (caso precise para utilizar outro ambiente):

```
- beautifulsoup4==4.11.1
- certifi==2022.12.7
- charset-normalizer==3.0.1
- cramjam==2.6.2
- fastparquet==2023.1.0
- fsspec==2023.1.0
- idna==3.4
- lxml==4.9.2
- numpy==1.24.1
- packaging==23.0
- pandas==1.5.3
- pyarrow==10.0.1
- python-dateutil==2.8.2
- pytz==2022.7.1
- requests==2.28.2
- six==1.16.0
- soupsieve==2.3.2.post1
- urllib3==1.26.14
```

0.3 3) Obs

- Testando a execução num ambiente virtual (dentro do VScode) cheguei a ter problema com Windows relacionado à permissão e para resolver abri o Power Shell como admin, digitei o conteúdo entre as aspas "Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Remote-Signed" e dei S.
- Em outros momentos já tive problema com bs4 também do BeautifulSoup na IDE VSCode, mesmo instalando corretamente (no Jupyter não precisa se preocupar), mas caso tenha o mesmo problema basta pressionar ctrl+shift+p e digitar "Select Interpreter" para escolher o interpretador de acordo com as suas configurações.

0.4 4) Dicionário de Dados

- UPC: Universal Product Code (identificador único do livro)
- Titulo: Nome do livro
- Avaliacao: Avaliação do livro (de 1 a 5)
- Disponibilidade: Se está em estoque ou não (nesse site só tem em estoque mesmo)
- Qtde_Disponivel: Quantidade de livros disponíveis
- Preco: Preço do livro (em Libras, porém gerados de forma aleatória então a moeda não faz diferença para este caso)
- Categoria: Categoria ao qual o livro pertence de acordo com o conteúdo

0.4.1 Bibliotecas Utilizadas

[1]:
```
import pandas as pd
import numpy as np
import requests
from bs4 import BeautifulSoup
import re
from time import sleep
import sqlite3
```

0.4.2 Criação das Funções

[2]:
```
titulo = []
avaliacao = []
disponibilidade = []

headers = {"User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0)␣
 ↪Gecko/20100101 Firefox/66.0",
               "Accept-Encoding":"gzip, deflate",
               "Accept":"text/html,application/xhtml+xml,application/xml;q=0.
 ↪9,*/*;q=0.8",
               "DNT":"1","Connection":"close", "Upgrade-Insecure-Requests":"1"}

def fazRequest(num_page):

    # Request para cada página do resultado de pesquisa
```

```
    requisicao = requests.get(f'https://books.toscrape.com/catalogue/
↪page-{str(num_page)}.html', headers = headers)

    if requisicao.status_code != 200:
        print('Please, wait!')
        sleep(60)
    else:
        return requisicao


def extraiConteudo(requisicao):
    # extrai o conteúdo do site
    return requisicao.content

def formataConteudo(conteudo):
    # formata o conteúdo do site
    return BeautifulSoup(conteudo, 'lxml')

def extraiNome(div):
    # busca a classe product_pod pela tag article
    for i in div.find_all('article',attrs={'class':'product_pod'}):
        titulo.append(i.h3.a['title'])

    # se o valor não for vazio extrai o nome
    if titulo is not None:
        return titulo
    else:
        return 'sem-nome'

def extraAvaliacaoBook(div):
    for i in soup.find_all('article',attrs={'class':'product_pod'}):
        avaliacao.append(i.p['class'][1])

    if avaliacao is not None:
        return avaliacao
    else:
        return '0'


def extraiDisponibilidade(div):

    for i in soup.find_all('div',attrs={'class':'product_price'}):
        if 'ok' in str(i.i['class']):
            disponibilidade.append('Em Estoque')
        else:
            disponibilidade.append('Não Disponível')
```

```
def retornaNumero(num):
    num = re.findall(r'\d+',num)
    return num

def limpaPreco(preco):
    preco = re.findall(r'\d+[.]\d+',preco)
    return preco

def limpaCaracteres(char):
    char = char.replace(']','').replace('[','').replace("'","")
    return char
```

[3]:
```
%%time
# remova esse time acima se quiser executar em outra IDE

url = 'https://books.toscrape.com/catalogue/'
preco = []
upc = []
qtde_disponivel = []
categoria = []

for n_pag in range(1,51):
    # Faz scraping do site e depois de cada link dentro dele

    # Faz a request
    requisicao = fazRequest(n_pag)

    # Exrai o conteúdo
    conteudo = extraiConteudo(requisicao)

    # Formata o conteúdo
    soup = formataConteudo(conteudo)

    extraiNome(soup)
    extraAvaliacaoBook(soup)
    extraiDisponibilidade(soup)

    for i in soup.find_all('article',attrs={'class':'product_pod'}):
            url = f"https://books.toscrape.com/catalogue/{i.h3.a['href']}"
            dados2 = requests.get(url, headers=headers).text
            soup2 = BeautifulSoup(dados2,'html.parser')
#             print(url)
            for j in soup2.find_all('table',attrs={'class':'table-striped'}):
                qtde_disponivel.append(j.find_all('td')[5].text)
                preco.append(j.find_all('td')[2].text)
                upc.append(j.find_all('td')[0].text)
            for k in soup2.find_all('ul', attrs={'class':'breadcrumb'}):
```

```
                categoria.append(k.find_all('li')[2].text.strip()) # estava␣
↪vindo com espaço
    print(f'Extração concluída da página: {n_pag}')
```

```
Extração concluída da página: 1
Extração concluída da página: 2
Extração concluída da página: 3
Extração concluída da página: 4
Extração concluída da página: 5
Extração concluída da página: 6
Extração concluída da página: 7
Extração concluída da página: 8
Extração concluída da página: 9
Extração concluída da página: 10
Extração concluída da página: 11
Extração concluída da página: 12
Extração concluída da página: 13
Extração concluída da página: 14
Extração concluída da página: 15
Extração concluída da página: 16
Extração concluída da página: 17
Extração concluída da página: 18
Extração concluída da página: 19
Extração concluída da página: 20
Extração concluída da página: 21
Extração concluída da página: 22
Extração concluída da página: 23
Extração concluída da página: 24
Extração concluída da página: 25
Extração concluída da página: 26
Extração concluída da página: 27
Extração concluída da página: 28
Extração concluída da página: 29
Extração concluída da página: 30
Extração concluída da página: 31
Extração concluída da página: 32
Extração concluída da página: 33
Extração concluída da página: 34
Extração concluída da página: 35
Extração concluída da página: 36
Extração concluída da página: 37
Extração concluída da página: 38
Extração concluída da página: 39
Extração concluída da página: 40
Extração concluída da página: 41
Extração concluída da página: 42
Extração concluída da página: 43
Extração concluída da página: 44
```

```
Extração concluída da página: 45
Extração concluída da página: 46
Extração concluída da página: 47
Extração concluída da página: 48
Extração concluída da página: 49
Extração concluída da página: 50
Wall time: 11min 9s
```

[4]:
```
df = pd.DataFrame({'UPC': upc,
                   'Titulo':titulo,
                   'Avaliacao':avaliacao,
                   'Disponibilidade':disponibilidade,
                   'Qtde_Disponivel':qtde_disponivel,
                   'Preco':preco,
                   'Categoria':categoria})
```

[5]:
```
df.head()
```

[5]:

	UPC	Titulo	Avaliacao	Disponibilidade	Qtde_Disponivel	Preco	Categoria
0	a897fe39b1053632	A Light in the Attic	Three	Em Estoque	In stock (22 available)	Â£51.77	Poetry
1	90fa61229261140a	Tipping the Velvet	One	Em Estoque	In stock (20 available)	Â£53.74	Historical Fiction
2	6957f44c3847a760	Soumission	One	Em Estoque	In stock (20 available)	Â£50.10	Fiction
3	e00eb4fd7b871a48	Sharp Objects	Four	Em Estoque	In stock (20 available)	Â£47.82	Mystery
4	4165285e1663650f	Sapiens: A Brief History of Humankind	Five	Em Estoque	In stock (20 available)	Â£54.23	History

[6]:
```
df['Preco'] = df['Preco'].astype(str).apply(limpaPreco)
df['Preco'] = df['Preco'].astype(str).apply(limpaCaracteres)
df['Qtde_Disponivel'] = df['Qtde_Disponivel'].astype(str).apply(retornaNumero)
df['Qtde_Disponivel'] = df['Qtde_Disponivel'].astype(str).apply(limpaCaracteres)

for index, i in enumerate(df['Avaliacao']):
    if i == 'One':
        df.loc[index,'Avaliacao'] = 1
    elif i == 'Two':
        df.loc[index,'Avaliacao'] = 2
    elif i == 'Three':
        df.loc[index,'Avaliacao'] = 3
    elif i == 'Four':
        df.loc[index,'Avaliacao'] = 4
    elif i == 'Five':
        df.loc[index,'Avaliacao'] = 5
```

```
    else:
        df.loc[index,'Avaliacao'] = 0

df['Preco'] = df['Preco'].astype(np.float64, copy=False)
df['Qtde_Disponivel'] = df['Qtde_Disponivel'].astype(np.int64, copy=False)
df['Avaliacao'] = df['Avaliacao'].astype(np.int64, copy=False)
```

[7]:
```
df.head()
```

[7]:

	UPC	Titulo	Avaliacao	Disponibilidade	Qtde_Disponivel	Preco	Categoria
0	a897fe39b1053632	A Light in the Attic	3	Em Estoque	22	51.77	Poetry
1	90fa61229261140a	Tipping the Velvet	1	Em Estoque	20	53.74	Historical Fiction
2	6957f44c3847a760	Soumission	1	Em Estoque	20	50.10	Fiction
3	e00eb4fd7b871a48	Sharp Objects	4	Em Estoque	20	47.82	Mystery
4	4165285e1663650f	Sapiens: A Brief History of Humankind	5	Em Estoque	20	54.23	History

0.4.3 Exportação do dataframe para diferentes formatos

[8]:
```
df.to_csv(r'datasets/books.csv',index=False)
df.to_parquet(r'datasets/books.parquet.gzip', compression='gzip')
```

[9]:
```
conn = sqlite3.connect(r'datasets/books.db')
```

[10]:
```
df.to_sql(name='fato', con=conn, index=False, if_exists='replace')
```

[10]:
```
1000
```

[11]:
```
df_sql = pd.read_sql('SELECT * FROM fato', con=conn)
```

[12]:
```
df_sql
```

[12]:

	UPC	Titulo
0	a897fe39b1053632	A Light in the Attic
1	90fa61229261140a	Tipping the Velvet
2	6957f44c3847a760	Soumission
3	e00eb4fd7b871a48	Sharp Objects
4	4165285e1663650f	Sapiens: A Brief History of Humankind
..	…	…
995	cd2a2a70dd5d176d	Alice in Wonderland (Alice's Adventures in Won…
996	bfd5e1701c862ac3	Ajin: Demi-Human, Volume 1 (Ajin: Demi-Human #1)

```
997  19fec36a1dfb4c16  A Spy's Devotion (The Regency Spies of London #1)
998  f684a82adc49f011                1st to Die (Women's Murder Club #1)
999  228ba5e7577e1d49                 1,000 Places to See Before You Die

     Avaliacao Disponibilidade  Qtde_Disponivel  Preco           Categoria
0            3      Em Estoque                22  51.77              Poetry
1            1      Em Estoque                20  53.74  Historical Fiction
2            1      Em Estoque                20  50.10             Fiction
3            4      Em Estoque                20  47.82             Mystery
4            5      Em Estoque                20  54.23             History
..         …            …               …    …                 …
995          1      Em Estoque                 1  55.53            Classics
996          4      Em Estoque                 1  57.06      Sequential Art
997          5      Em Estoque                 1  16.97  Historical Fiction
998          1      Em Estoque                 1  53.98             Mystery
999          5      Em Estoque                 1  26.08              Travel

[1000 rows x 7 columns]
```

[13]:
```
query = """
            SELECT "Titulo", "Preco","Avaliacao"
            FROM fato
            WHERE "Preco" > 50 AND "Avaliacao" >= 3;
"""
```

[14]:
```
df_sql_filtrado = pd.read_sql(query, con=conn)
```

[15]:
```
df_sql_filtrado
```

[15]:
```
                                                Titulo  Preco  Avaliacao
0                                A Light in the Attic  51.77          3
1                  Sapiens: A Brief History of Humankind  54.23          5
2    Scott Pilgrim's Precious Little Life (Scott Pi…  52.29          5
3    Our Band Could Be Your Life: Scenes from the A…  57.25          3
4                          Birdsong: A Story in Pictures  54.64          3
..                                                   …    …        …
119                          Listen to Me (Fusion #1)  58.99          3
120                        Kitchens of the Great Midwest  57.20          5
121                Giant Days, Vol. 1 (Giant Days #1-4)  56.76          4
122                                     Eat, Pray, Love  51.32          3
123   Ajin: Demi-Human, Volume 1 (Ajin: Demi-Human #1)  57.06          4

[124 rows x 3 columns]
```

[16]:
```
conn.close()
```

0.4.4 Exploração básica e rápida

[17]:
```
df.describe()
```

[17]:
```
          Avaliacao  Qtde_Disponivel       Preco
count  1000.000000      1000.000000  1000.00000
mean      2.923000         8.585000    35.07035
std       1.434967         5.654622    14.44669
min       1.000000         1.000000    10.00000
25%       2.000000         3.000000    22.10750
50%       3.000000         7.000000    35.98000
75%       4.000000        14.000000    47.45750
max       5.000000        22.000000    59.99000
```

[18]:
```
df.describe(percentiles=[0.5,0.75,0.995])
```

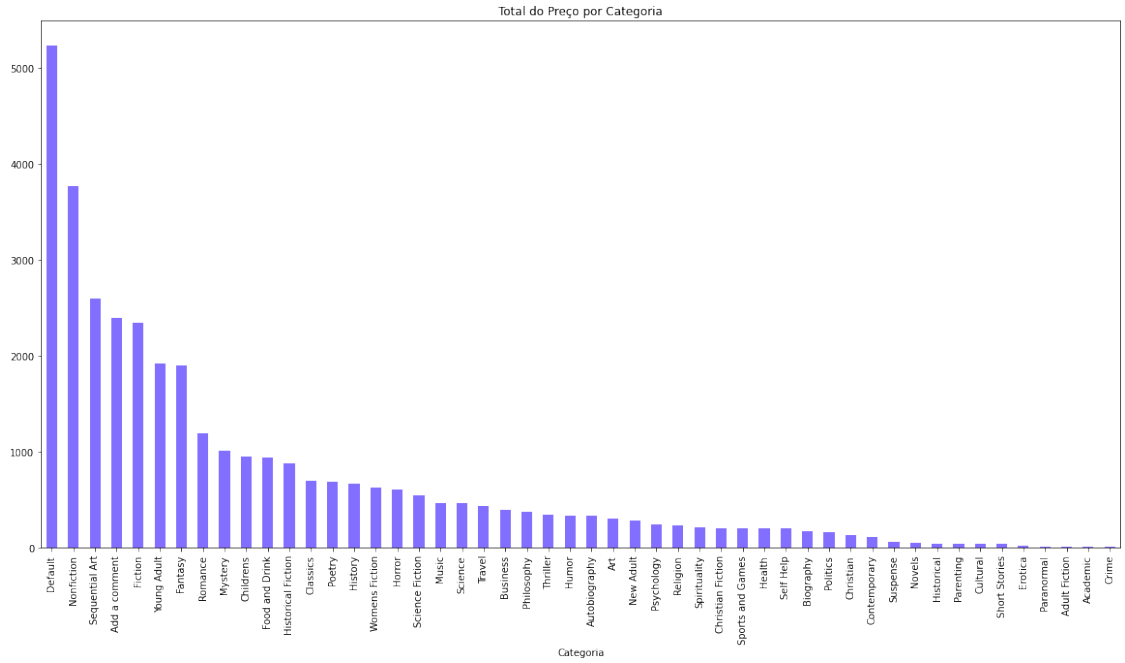
[18]:
```
          Avaliacao  Qtde_Disponivel       Preco
count  1000.000000      1000.000000  1000.00000
mean      2.923000         8.585000    35.07035
std       1.434967         5.654622    14.44669
min       1.000000         1.000000    10.00000
50%       3.000000         7.000000    35.98000
75%       4.000000        14.000000    47.45750
99.5%     5.000000        19.005000    59.71095
max       5.000000        22.000000    59.99000
```

[19]:
```
df_categoria = df.groupby('Categoria')
```

[20]:
```
import matplotlib.pyplot as plt
```

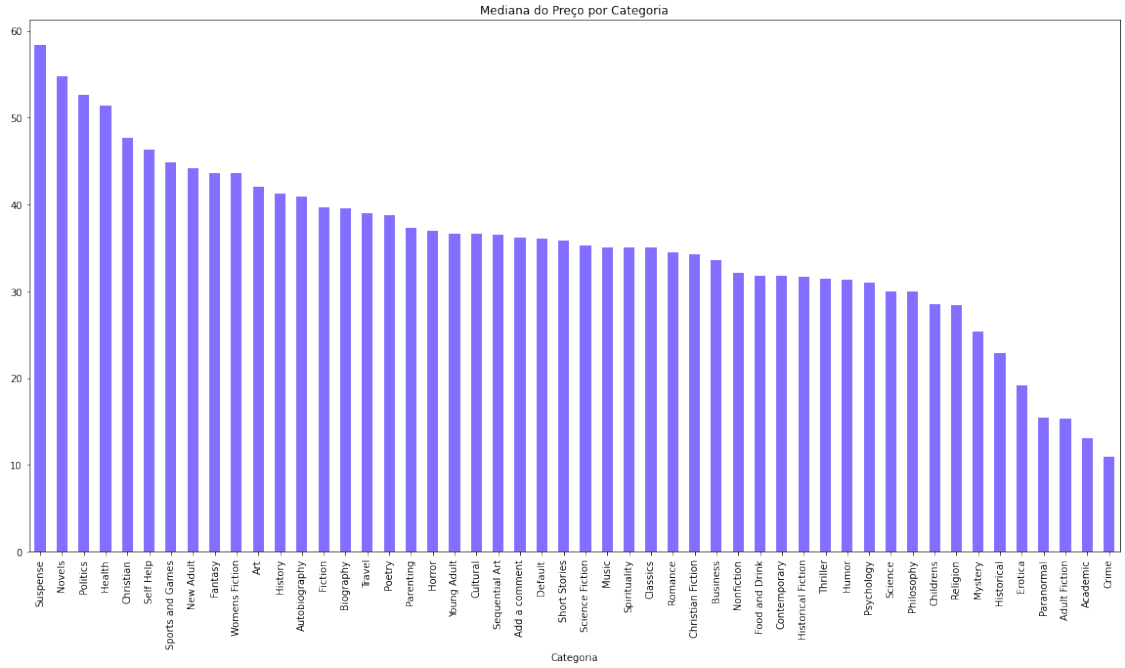
[21]:
```
plt.figure()
df_categoria['Preco'].sum().sort_values(ascending=False).plot.
↪bar(figsize=(20,10),
                                                                  ␣
↪color=['#836FFF'],
                                                                   title='Total␣
↪do Preço por Categoria')
```

[21]:
```
<AxesSubplot:title={'center':'Total do Preço por Categoria'},
xlabel='Categoria'>
```



[22]:
```
plt.figure()
df_categoria['Preco'].median().sort_values(ascending=False).plot.
↪bar(figsize=(20,10),
                                                                  ␣
↪color=['#836FFF'],
                                                                   title='Mediana␣
↪do Preço por Categoria')
```

[22]:
```
<AxesSubplot:title={'center':'Mediana do Preço por Categoria'},
xlabel='Categoria'>
```


Mediana do Preço por Categoria
60
50
40
30
20
10
0
Suspense
Novels
Politics
Health
Christian
Self Help
Sports and Games
New Adult
Fantasy
Womens Fiction
Art
History
Autobiography
Fiction
Biography
Travel
Poetry
Parenting
Horror
Young Adult
Cultural
Sequential Art
Add a comment
Default
Short Stories
Science Fiction
Music
Spirituality
Classics
Romance
Christian Fiction
Business
Nonfiction
Food and Drink
Contemporary
Historical Fiction
Thriller
Humor
Psychology
Science
Philosophy
Childrens
Religion
Mystery
Historical
Erotica
Paranormal
Adult Fiction
Academic
Crime
Categoria