

Notebook Jupyter 1a_trataCSVsSiscori

Importação e tratamento dos dados Siscori

Os dados importados no Siscore vêm em arquivos do tipo CSV no formato **CAPINNAAMM**, onde NN é o número do capítulo extraído, AA é o ano com dois dígitos e MM é o mês com dois dígitos, formando o período de referência dos dados extraídos. O arquivo CSV obtido vem configurado com o separador "@" e descrição da coluna com excesso de espaços, o que precisa de uma camada de tratamento para correta importação dos dados.

Importação das Bibliotecas

```
In [1]: import pandas as pd
import numpy as np
import os, time
```

```
In [2]: # Data e hora da execução do script
initot=time.time()
print(f'Código executado em {time.strftime("%d/%m/%Y às %H:%M", time.localtime(time.time()))}')

```

Código executado em 25/01/2022 às 10:18

Definindo dados gerais

Cria lista de arquivos CSV contidos na pasta atual

```
In [3]: arqsCSV = []
for arquivo in os.listdir("./bases/siscori/"):
    if arquivo[-3:].upper()=="CSV" and arquivo[:4]=="CAPI":
        arqsCSV.append("./bases/siscori/"+arquivo)
arqsCSV=sorted(arqsCSV)
print(arqsCSV)

['./bases/siscori/CAP1872001.CSV', './bases/siscori/CAP1872002.CSV', './bases/siscori/CAP1872003.CSV', './bases/siscori/CAP1872004.CSV', './bases/siscori/CAP1872005.CSV', './bases/siscori/CAP1872006.CSV', './bases/siscori/CAP1872007.CSV', './bases/siscori/CAP1872008.CSV', './bases/siscori/CAP1872009.CSV', './bases/siscori/CAP1872010.CSV', './bases/siscori/CAP1872011.CSV', './bases/siscori/CAP1872012.CSV', './bases/siscori/CAP1872101.CSV', './bases/siscori/CAP1872102.CSV', './bases/siscori/CAP1872103.CSV', './bases/siscori/CAP1872104.CSV', './bases/siscori/CAP1872105.CSV', './bases/siscori/CAP1872106.CSV']

```

Cria variáveis com nomes das colunas e seus tipos

```
In [4]: tipos = {'NUMERO DE ORDEM': str,
                'ANOMES': str,
                'COD.NCM': str,
                'DESCRICAO DO CODIGO NCM': object,
                'PAIS.OR': int,
                'PAIS DE ORIGEM': object,
                'PAIS.AQ': int,
                'PAIS DE AQUISICAO': object,
                'UND.ESTAT.': int,
                'UNIDADE DE MEDIDA': object,
                'UNIDADE COMERC.': object,
                'DESCRICAO DO PRODUTO': object,
                'QTDE ESTATISTICA': float,
                'PESO LIQUIDO': float,
                'VMLE DOLAR': float,
                'VL FRETE DOLAR': float,
                'VL SEGURO DOLAR': float,
                'VALOR UN.PROD.DOLAR': float,
                'QTD COMERCIAL.': float,
                'TOT.UN.PROD.DOLAR': float,
                'UNIDADE DESEMBARQUE': object,
                'UNIDADE DESEMBARACO': object,
                'INCOTERM': object,
                'NAT.INFORMACAO': object,
                'SITUACAO DO DESPACHO': object}
colunas = list(tipos.keys())
```

Inicializa a variável que conterá o tamanho total do dataset original

```
In [5]: tamanhoDataset=0
```

Inicializa um dataframe vazio que conterá os dados finais

```
In [6]: df = pd.DataFrame(columns = colunas)
df.head()
```

```
Out[6]:
```

NUMERO DE ORDEM	ANOMES	COD.NCM	DESCRICAO DO CODIGO NCM	PAIS.OR	PAIS DE ORIGEM	PAIS.AQ	PAIS DE AQUISICAO	UND.ESTAT.
0 rows × 9 columns								

Importa cada CSV, trata e concatena no DataFrame final

```

In [7]: # Executa para cada CSV na Lista
print("Iniciando importação dos arquivos...\n")
for arqCSV in arqsCSV:
    print(f"Iniciando arquivo {arqCSV}...")
    versao=pd.__version__
    if versao < '1.3.0': # argumento deprecado a partir da v1.3.0 - sem permissão d
e atualização RFB
        dftemp = pd.read_csv(arqCSV, sep='@', decimal='r',',', engine='python', encodi
ng = "ISO-8859-1", header = 0,
                                names = colunas, dtype = tipos, quotechar="'", error_bad_l
ines=False, warn_bad_lines=False)
    else:
        dftemp = pd.read_csv(arqCSV, sep='@', decimal='r',',', engine='python', encodi
ng = "ISO-8859-1", header = 0,
                                names = colunas, dtype = tipos, quotechar="'", on_bad_
lines='skip')
    print('DataFrame carregado...\nAplicando filtros...')
    # Elimina os registros sem valores ou nulos
    dftemp = dftemp.dropna()
    # Incrementa o tamanho do Dataset
    tamanhoDataset += dftemp[dftemp.columns[0]].count()
    print('Dados da importação do arquivo')
    print('Quantidade de registros válidos:' + f'{str(dftemp[dftemp.columns[0]].cou
nt()):>8}')
    # Filtra o DataFrame somente com os registros de interesse
    # Filtro 1: NCM de interesse: 87141000
    indiceNCM = dftemp['COD.NCM'] == '87141000'
    dftemp = dftemp[indiceNCM]
    # Filtro 2: Incluir registros com descrição contendo palavras da lista a inclui
r
    listafiltroincluir = ["transm", "corrente", "coroa", "pinhao|pinhão"] # A barra
vertical (|) faz o "ou".
    for termo in listafiltroincluir:
        dftemp=dftemp[dftemp['DESCRICAO DO PRODUTO'].str.contains(termo, case=False
)]
    # Filtro 3: Excluir registros com descrição contendo palavras da lista
    padraofiltroexcluir = r"semi|reposicao|reposição"
    dftemp=dftemp[dftemp['DESCRICAO DO PRODUTO'].str.contains(padraofiltroexcluir,
case=False, regex=True)==False]
    print(f'Quantidade de registros filtrados:' + f'{str(dftemp[dftemp.columns[0]].
count()):>6}')
    # Concatena com o DataFrame final
    df = pd.concat([df,dftemp])
    print(arqCSV + ' finalizado.\n')

```

Iniciando importação dos arquivos...

Iniciando arquivo ./bases/siscori/CAP1872001.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 491150
Quantidade de registros filtrados: 810
./bases/siscori/CAP1872001.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872002.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 376497
Quantidade de registros filtrados: 681
./bases/siscori/CAP1872002.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872003.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 435878
Quantidade de registros filtrados: 906
./bases/siscori/CAP1872003.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872004.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 330218
Quantidade de registros filtrados: 184
./bases/siscori/CAP1872004.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872005.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 247533
Quantidade de registros filtrados: 429
./bases/siscori/CAP1872005.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872006.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 230660
Quantidade de registros filtrados: 625
./bases/siscori/CAP1872006.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872007.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 305473
Quantidade de registros filtrados: 1236
./bases/siscori/CAP1872007.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872008.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 340985
Quantidade de registros filtrados: 1395
./bases/siscori/CAP1872008.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872009.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo

Quantidade de registros válidos: 334556
Quantidade de registros filtrados: 1421
./bases/siscori/CAP1872009.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872010.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 431598
Quantidade de registros filtrados: 1167
./bases/siscori/CAP1872010.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872011.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 435236
Quantidade de registros filtrados: 1084
./bases/siscori/CAP1872011.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872012.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 464189
Quantidade de registros filtrados: 1009
./bases/siscori/CAP1872012.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872101.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 463810
Quantidade de registros filtrados: 1278
./bases/siscori/CAP1872101.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872102.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 407665
Quantidade de registros filtrados: 1594
./bases/siscori/CAP1872102.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872103.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 549879
Quantidade de registros filtrados: 1517
./bases/siscori/CAP1872103.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872104.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 636066
Quantidade de registros filtrados: 1155
./bases/siscori/CAP1872104.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872105.CSV...
DataFrame carregado...
Aplicando filtros...
Dados da importação do arquivo
Quantidade de registros válidos: 482319
Quantidade de registros filtrados: 877
./bases/siscori/CAP1872105.CSV finalizado.

Iniciando arquivo ./bases/siscori/CAP1872106.CSV...
DataFrame carregado...
Aplicando filtros...

```
Dados da importação do arquivo
Quantidade de registros válidos: 729922
Quantidade de registros filtrados: 908
./bases/siscori/CAPI872106.CSV finalizado.
```

Resetando o índice do DataFrame importado

```
In [8]: df.reset_index(inplace=True, drop=True)
```

Excluindo colunas desnecessárias

```
In [9]: excluir=['NUMERO DE ORDEM',
                'ANOMES',
                'COD.NCM',
                'DESCRICAO DO CODIGO NCM',
                'PAIS.OR',
                'PAIS.AQ',
                'PAIS DE AQUISICAO',
                'UND.ESTAT.',
                'UNIDADE DE MEDIDA',
                'UNIDADE COMERC.',
                'QTDE ESTATISTICA',
                'PESO LIQUIDO',
                'VMLE DOLAR',
                'VL FRETE DOLAR',
                'VL SEGURO DOLAR',
                'QTD COMERCIAL.',
                'TOT.UN.PROD.DOLAR',
                'UNIDADE DESEMBARQUE',
                'UNIDADE DESEMBARACO',
                'INCOTERM',
                'NAT.INFORMACAO',
                'SITUACAO DO DESPACHO']
```

```
In [10]: df=df.drop(excluir, axis=1)
```

Remover espaços em excesso nos campos string

```
In [11]: # Remove espaços em excesso das colunas em colstr
colstr = ['PAIS DE ORIGEM',
          'DESCRICAO DO PRODUTO']
for coluna in colstr:
    df[coluna]=df[coluna].str.strip()
```

Verificando o DataFrame importado

```
In [12]: df.sample(5)
```

```
Out[12]:
```

	PAIS DE ORIGEM	DESCRICAO DO PRODUTO	VALOR UN.PROD.DOLAR
923	CHINA, REPUBLICA POP	KIT TRANSMISSÃO, COMPOSTO DE CORRENTE, COROA E...	3.8510
11208	CHINA, REPUBLICA POP	91163 - KIT CG 125 FAN (14-16) / CG 125 CARGO ...	4.5639
3789	CHINA, REPUBLICA POP	KIT DE TRANSMISSAO , MARCA RIFFEL, TITANIUM (1...	5.2300
818	CHINA, REPUBLICA POP	KIT DE TRANSMISSAO , MARCA RIFFEL, TITANIUM (1...	5.5520
13861	CHINA, REPUBLICA POP	24753 - 358034 - KIT DE TRANSMISSÃO PARA MOTOC...	4.2000

```
In [13]: df.shape
```

```
Out[13]: (18276, 3)
```

Exportando o DataFrame

Exportando para um arquivo CSV

```
In [14]: df.to_csv(r'./bases/dataframe.csv', index = False, header = True)
```

Exportando para um arquivo de planilha do Excel

```
In [15]: df.to_excel(r'./bases/dataframe.xlsx', index = False, header = True)
```

Compara o tamanho total do Dataset inicial e final

```
In [16]: print('\nQuantidade total de registros válidos importados:' + f'{str(tamanhoDataset):>8}')  
print('Tamanho do dataset após aplicação dos filtros: ' + f'{str(df.shape[0]):>8}')
```

```
Quantidade total de registros válidos importados: 7693634  
Tamanho do dataset após aplicação dos filtros:      18276
```

```
In [17]: tempotot=time.time()-initot  
if tempotot>60:  
    print(f'Tempo total de execução: {tempotot/60:.2f} minutos.')  
else:  
    print(f'Tempo total de execução: {tempotot:.2f} segundos.')
```

```
Tempo total de execução: 5.51 minutos.
```