Documentação *Sprint* 3

A tabela utilizada foi a STG_MVT_CRD que contém quatro colunas com valores inválidos. O objetivo desta entrega foi exibir os campos válidos e inválidos em forma de tabela dessas respectivas colunas. Os nomes das colunas com valores inválidos são: VLR_SDO_UTZ_CRD_RTO, VLR_TOT_FAT, VLR_MIM_FAT e VLR_PCL_FAT.

A biblioteca utilizada foi o pandas, que foi abreviado como "pd" para ser chamada nas outras linhas de código para a utilização de suas ferramentas (Fig. 1).

Figura 1: Biblioteca pandas importada como "pd".

```
In [1]: import pandas as pd
```

Os parâmetros que não apareceram nas anteriores foram *usecols=['']* e *squeeze*, pertencentes ao método *read_csv()*. Aquele é utilizado para selecionar as colunas que se quer exibir, o nome das colunas são escritos em aspas simples ou duplas; este recebe argumentos booleanos e vêm como padrão o argumento *false*, dessa forma a formatação da planilha é mantida, o que restringe a manipulação de dados. O argumenta dado para este parâmetro foi *true* para que o retorno de uma coluna somente com valores inválidos ou com valores válidos fosse possível (Fig. 2).

Dentro dos parênteses do método *read_csv()*[1], digita-se os parâmetros em ordem estipulada pela biblioteca pandas. Caso não seja explicitado os parâmetros e só digitado o argumento, deve-se respeitar as entradas corretas para não causar erro de interpretação. Por exemplo, se fosse digitado o nome da coluna no segundo parâmetro sem explicitar o *usecols=['']*, seria mostrado um erro por argumento indevido por este ser o sétimo parâmetro (Fig. 3). Para que seja alterada a ordem dos parâmetros, deve-se explicitá-los conforme figura 2.

[1] Para acessar a ordem dos parâmetros, clique dentro dos parênteses e pressione as teclas SHIFT+TAB+TAB. O segundo TAB é para expandir a lista.

Figura 2: Método *read_csv()* com os parâmetros explicitados *usecols=['']* e *squeeze*, com seus respectivos argumentos.

```
#coluna índice 1: valores inválidos.
a1 = pd.read_csv('STG_MVT_CRD_csv.csv', usecols=['VLR_SDO_UTZ_CRD_RTO'], squeeze=True)
a1[a1.isnull()]
```

Figura 3: Alguns parâmetros do método *read_csv()* na sua ordem padrão. Nota-se que os parâmetros *usecols* e *squeeze* estão nas posições 7 e 8 respectivamente.

```
Signature:
pd.read_csv(
    filepath_or_buffer: Union[str, pathlib.Path, IO[~AnyStr]],
    sep=',',
    delimiter=None,
    header='infer',
    names=None,
    index_col=None,
    usecols=None,
    squeeze=False,
```

Foi exibida uma sub-tabela da tabela principal dos valores inválidos e válidos, tendo como padrão todas as colunas que tinham pelo menos um valor nulo. Com o parâmetro *squeeze=False* isso não seria possível. Lembrando que o índice começa do 0 (zero), os valores nulos da coluna VLR_PCL_FAT correspondem às linhas 10 (9+1) e 15 (14+1) (Fig. 4).

Figura 4: Sub-tabelas da tabela STG_MVT_CRD_csv. In: refere-se ao código da busca; Out: refere-se ao resultado da busca com uma tabela não formatada, o nome da coluna usada (*Name*) e o tipo[2] de dado armazenado (*dtype*). Reparar que em *Out* a primeira coluna é o índice e a segunda coluna é o valor do campo. A: retorna os campos nulos. B: retorna os campos não nulos.

A

```
In [39]: #coluna índice 4: valores inválidos.
         a4 = pd.read_csv('STG_MVT_CRD_csv.csv', usecols=['VLR_PCL_FAT'], squeeze=True)
         a4[a4.isnull()]

Out[39]: 9    NaN
         14   NaN
         Name: VLR_PCL_FAT, dtype: float64
```

B

```
In [43]: #coluna índice 4: valores válidos.
         a8 = pd.read_csv('STG_MVT_CRD_csv.csv', usecols=['VLR_PCL_FAT'], squeeze=True)
         a8[a8.notnull()]

Out[43]: 0     4.595517e+10
         1     7.387237e+10
         2     2.868181e+09
         3     6.111251e+10
         4     2.152609e+06
         5     4.607337e+08
         6     2.638331e+11
         7     1.838029e+11
         8     9.738713e+08
         10    9.496722e+09
         11    3.556795e+11
         12    1.834447e+09
         13    1.733950e+08
         Name: VLR_PCL_FAT, dtype: float64
```

[2] Notar que, por haver dados nulos, a biblioteca pandas considera todos os valores como ponto flutuante.