

0. Preparação do ambiente

credito.csv

```
1 %%writefile credito.csv
2 id_vendedor,valor_emprestimos,quantidade_emprestimos,data
3 104271,448.0,1,20161208
4 21476,826.7,3,20161208
5 87440,313.6,3,20161208
6 15980,-8008.0,6,20161208
7 215906,2212.0,5,20161208
8 33696,2771.3,2,20161208
9 33893,2240.0,3,20161208
10 214946,-4151.0,18,20161208
11 123974,2021.95,2,20161208
12 225870,4039.0,2,20161208
```

 Overwriting credito.csv

Vamos ler o conteúdo do arquivo em uma lista onde cada elemento é um dicionário representando as linhas do arquivo.

```
1 # Usando a função readlines para converter o arquivo em uma lista sendo cada linha uma string
2 with open(file="./credito.csv", mode="r", encoding="utf8") as file:
3     linhas = file.readlines()
4 linhas
```

```
['id_vendedor,valor_emprestimos,quantidade_emprestimos,data\n',
 '104271,448.0,1,20161208\n',
 '21476,826.7,3,20161208\n',
 '87440,313.6,3,20161208\n',
 '15980,-8008.0,6,20161208\n',
 '215906,2212.0,5,20161208\n',
 '33696,2771.3,2,20161208\n',
 '33893,2240.0,3,20161208\n',
 '214946,-4151.0,18,20161208\n',
 '123974,2021.95,2,20161208\n',
 '225870,4039.0,2,20161208\n']
```

```
1 # Formatando os valores da lista, para obter uma lista de listas de variáveis
2 linhas_listadas = list(map(lambda x: x.replace("\n","").split(sep=","), linhas))
3 linhas_listadas
```

```
[['id_vendedor', 'valor_emprestimos', 'quantidade_emprestimos', 'data'],
 ['104271', '448.0', '1', '20161208'],
 ['21476', '826.7', '3', '20161208'],
 ['87440', '313.6', '3', '20161208'],
 ['15980', '-8008.0', '6', '20161208'],
 ['215906', '2212.0', '5', '20161208'],
 ['33696', '2771.3', '2', '20161208'],
 ['33893', '2240.0', '3', '20161208'],
 ['214946', '-4151.0', '18', '20161208'],
 ['123974', '2021.95', '2', '20161208'],
 ['225870', '4039.0', '2', '20161208']]
```

```
1 # Usando a função map para unir cada uma das variáveis com seu index em um dicionário, usando a função zip
2 emprestimos = list(map(lambda x: dict(zip(linhas_listadas[0], x)), linhas_listadas[1:]))
3 for emprestimo in emprestimos:
4     print(emprestimo)
```

```
{'id_vendedor': '104271', 'valor_emprestimos': '448.0', 'quantidade_emprestimos': '1', 'data': '20161208'}
{'id_vendedor': '21476', 'valor_emprestimos': '826.7', 'quantidade_emprestimos': '3', 'data': '20161208'}
{'id_vendedor': '87440', 'valor_emprestimos': '313.6', 'quantidade_emprestimos': '3', 'data': '20161208'}
{'id_vendedor': '15980', 'valor_emprestimos': '-8008.0', 'quantidade_emprestimos': '6', 'data': '20161208'}
{'id_vendedor': '215906', 'valor_emprestimos': '2212.0', 'quantidade_emprestimos': '5', 'data': '20161208'}
{'id_vendedor': '33696', 'valor_emprestimos': '2771.3', 'quantidade_emprestimos': '2', 'data': '20161208'}
{'id_vendedor': '33893', 'valor_emprestimos': '2240.0', 'quantidade_emprestimos': '3', 'data': '20161208'}
{'id_vendedor': '214946', 'valor_emprestimos': '-4151.0', 'quantidade_emprestimos': '18', 'data': '20161208'}
{'id_vendedor': '123974', 'valor_emprestimos': '2021.95', 'quantidade_emprestimos': '2', 'data': '20161208'}
{'id_vendedor': '225870', 'valor_emprestimos': '4039.0', 'quantidade_emprestimos': '2', 'data': '20161208'}
```

✓ 1. Função map

1.1 Aplique a função map na lista de `emprestimos` para extrair os valores da chave

- ✓ `valor_emprestimos` na lista `valor_emprestimos_lista`. Faça também a conversão de `str` para `float`.

```
1 # Usando a função map para obter dos empréstimos apenas a variável valor_emprestimos, convertendo esse para float
2 valor_emprestimos_lista = list(map(lambda x: float(x["valor_emprestimos"]), empréstimos))
3 valor_emprestimos_lista

[448.0,
 826.7,
 313.6,
 -8008.0,
 2212.0,
 2771.3,
 2240.0,
 -4151.0,
 2021.95,
 4039.0]
```

✓ 2. Função filter

2.1 Aplique a função filter na lista de `valor_emprestimos_lista` para filtrar apenas os valores

- ✓ maiores que zero (os valores negativos são erros na base de dados). Salve os valores na lista `valor_emprestimos_lista_filtrada`.

```
1 # Usando filter para obter os valores positivos de empréstimo
2 valor_emprestimos_lista_filtrada = list(filter(lambda x: x > 0, valor_emprestimos_lista))
3 valor_emprestimos_lista_filtrada

[448.0, 826.7, 313.6, 2212.0, 2771.3, 2240.0, 2021.95, 4039.0]
```

✓ 3. Função reduce

Com a nossa lista de valores de empréstimo pronta, vamos extrair algumas métricas.

✓ 3.1 Função reduce para extrair a **soma**

```
1 from functools import reduce

1 def float_p_real(valor: float) -> str:
2     '''
3         Converte um valor_compr float para um R$
4         :param valor (float): Float to be converted
5         :return d (str): String edited
6     '''
7     a = f'R$ {valor:,.2f}'
8     b = a.replace(',', 'v')
9     c = b.replace('.', ',')
10    d = c.replace('v', '.')
11    return d
```

```

1 # Usando a função reduce para somar os valores de emprestimo
2 soma_valor_emprestimos = reduce(lambda x, y: x + y, valor_emprestimos_lista_filtrada)
3 soma_valor_emprestimos

14872.550000000001

1 print(f"Soma dos valores de empréstimos: {float_p_real(soma_valor_emprestimos)}")

Soma dos valores de empréstimos: R$ 14.872,55

```

✓ 3.2 Função reduce para extrair a **media aritmética**

Aplique a função reduce para extrair a média aritmética (mais informações [aqui](#)) dos elementos da lista `valor_emprestimos_lista_filtrada` na variável `media_valor_emprestimos`.

Dica: Para calcular o tamanho da lista, isto é a quantidade de elementos, utilize a função `len()`, dentro do argumento da função coloque a lista `valor_emprestimos_lista_filtrada`.

Formula da média aritmética

$$\bar{X} = \frac{\sum x}{n}$$

```

1 # Usando a função reduce para somar os valores de emprestimo, dividindo esse valor pelo número de valores para obter a média aritmética
2 media_art_valor_emprestimo = reduce(lambda x, y: x + y, valor_emprestimos_lista_filtrada) / len(valor_emprestimos_lista_filtrada)
3 media_art_valor_emprestimo

1859.0687500000001

1 # Refatorando o código da célula anterior, porém usando a soma salva na variável soma_valor_emprestimos
2 media_art_valor_emprestimo = soma_valor_emprestimos / len(valor_emprestimos_lista_filtrada)
3 print(f"Valor médio de empréstimos: {float_p_real(media_art_valor_emprestimo)}")

Valor médio de empréstimos: R$ 1.859,07

```

✓ 3.3 (Desafio) Função reduce para extrair o **desvio padrão amostral**

Aplique a função reduce para extrair a média aritmética (mais informações [aqui](#)) dos elementos da lista `valor_emprestimos_lista_filtrada` na variável `desvio_padrao_valor_emprestimos`.

Formula do desvio padrão amostral

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

```

1 # Obtendo o desvio padrão, usando o reduce na soma de valores de emprestimo para obtenção da média e usando reduce para somar o quadrado
2 desvio_padrao_valor_emprestimos = (reduce(lambda i, j: i + j, list(map(lambda x: (x - reduce(lambda x, y: x + y, valor_emprestimos_lista_filtrada))**2, valor_emprestimos_lista_filtrada))) / len(valor_emprestimos_lista_filtrada))**0.5
3 desvio_padrao_valor_emprestimos

1271.9972711497849

1 # Refatorando o código anterior, para melhorar a legibilidade
2 numero_de_itens = len(valor_emprestimos_lista_filtrada) - 1
3
4 subtra_media_valor_emprestimos = list(map(lambda x: (x - media_art_valor_emprestimo)**2, valor_emprestimos_lista_filtrada))
5 media_subtra_media_valor_emprestimos = reduce(lambda i, j: i + j, subtra_media_valor_emprestimos) / numero_de_itens
6 desvio_padrao_valor_emprestimos = media_subtra_media_valor_emprestimos**0.5
7 desvio_padrao_valor_emprestimos

```

