

# Automatização de uma planilha do Excel

O objetivo desse post é automatizar uma planilha do Excel utilizando Python como linguagem. Vamos utilizar uma planilha com as vendas de três supermercados (o dataset completo e o código pode ser baixado no link do meu github no final do post) para gerar um gráfico de barras com a soma de cada tipo de produto dividido entre homem e mulher. Temos uma visão geral da planilha na figura 1.

Figura 1 - Parte da planilha de vendas

| Invoice ID  | Branch | City      | Customer typ | Gender | Product line   | Unit price | Quantity | Tax 5%  | Total    | Date       | Time  | Payment     | cogs   | gross margin | gross Income | Rating |
|-------------|--------|-----------|--------------|--------|----------------|------------|----------|---------|----------|------------|-------|-------------|--------|--------------|--------------|--------|
| 750-67-8428 | A      | Yangon    | Member       | Female | Health and bi  | 74,69      | 7        | 26,1415 | 548,9715 | 05/01/2019 | 13:08 | Ewallet     | 522,83 | 4,76190476   | 26,1415      | 9,1    |
| 226-31-3081 | C      | Naypyitaw | Normal       | Female | Electronic aci | 15,28      | 5        | 3,82    | 80,22    | 08/03/2019 | 10:29 | Cash        | 76,4   | 4,76190476   | 3,82         | 9,6    |
| 631-41-3108 | A      | Yangon    | Normal       | Male   | Home and lif   | 46,33      | 7        | 16,2155 | 340,5255 | 03/03/2019 | 13:23 | Credit card | 324,31 | 4,76190476   | 16,2155      | 7,4    |
| 123-19-1176 | A      | Yangon    | Member       | Male   | Health and bi  | 58,22      | 8        | 23,288  | 489,048  | 27/01/2019 | 20:33 | Ewallet     | 465,76 | 4,76190476   | 23,288       | 8,4    |
| 373-73-7910 | A      | Yangon    | Normal       | Male   | Sports and tri | 86,31      | 7        | 30,2085 | 634,3785 | 08/02/2019 | 10:37 | Ewallet     | 604,17 | 4,76190476   | 30,2085      | 5,3    |
| 699-14-3026 | C      | Naypyitaw | Normal       | Male   | Electronic aci | 85,39      | 7        | 29,8865 | 627,6165 | 25/03/2019 | 18:30 | Ewallet     | 597,73 | 4,76190476   | 29,8865      | 4,1    |
| 355-53-5943 | A      | Yangon    | Member       | Female | Electronic aci | 68,84      | 6        | 20,652  | 433,692  | 25/02/2019 | 14:36 | Ewallet     | 413,04 | 4,76190476   | 20,652       | 5,8    |
| 315-22-5665 | C      | Naypyitaw | Normal       | Female | Home and lif   | 73,56      | 10       | 36,78   | 772,38   | 24/02/2019 | 11:38 | Ewallet     | 735,6  | 4,76190476   | 36,78        | 8      |
| 665-32-9167 | A      | Yangon    | Member       | Female | Health and bi  | 36,26      | 2        | 3,626   | 76,146   | 10/01/2019 | 17:15 | Credit card | 72,52  | 4,76190476   | 3,626        | 7,2    |
| 692-92-5582 | B      | Mandalay  | Member       | Female | Food and bev   | 54,84      | 3        | 8,226   | 172,746  | 20/02/2019 | 13:27 | Credit card | 164,52 | 4,76190476   | 8,226        | 5,9    |
| 351-62-0822 | B      | Mandalay  | Member       | Female | Fashion acce   | 14,48      | 4        | 2,896   | 60,816   | 06/02/2019 | 18:07 | Ewallet     | 57,92  | 4,76190476   | 2,896        | 4,5    |

Fonte: O autor

O primeiro passo é importar os pacotes necessários para poder automatizar os processos no Excel. Os principais são: Pandas e openpyxl. Com eles, podemos, primeiramente, ler a planilha e selecionar quais colunas vamos trabalhar. Aqui, vamos só utilizar as colunas 'Gender', 'Product line' e 'Total', vistas na tabela 1, a fim de calcular os gastos total de homens e mulheres de cada linha de produto.

Tabela 1 - Colunas selecionadas

|     | Gender | Product line           | Total     |
|-----|--------|------------------------|-----------|
| 0   | Female | Health and beauty      | 548.9715  |
| 1   | Female | Electronic accessories | 80.2200   |
| 2   | Male   | Home and lifestyle     | 340.5255  |
| 3   | Male   | Health and beauty      | 489.0480  |
| 4   | Male   | Sports and travel      | 634.3785  |
| ... | ...    | ...                    | ...       |
| 995 | Male   | Health and beauty      | 42.3675   |
| 996 | Female | Home and lifestyle     | 1022.4900 |
| 997 | Male   | Food and beverages     | 33.4320   |
| 998 | Male   | Home and lifestyle     | 69.1110   |
| 999 | Female | Fashion accessories    | 649.2990  |

Fonte: O autor

Após esse passo, vamos fazer uma transposição da tabela e utilizar uma função de soma para calcular o total gasto. Vemos o resultado na tabela 2.

Tabela 2 - Tabela transposta

| Product line | Electronic accessories | Fashion accessories | Food and beverages | Health and beauty | Home and lifestyle | Sports and travel |
|--------------|------------------------|---------------------|--------------------|-------------------|--------------------|-------------------|
| Gender       |                        |                     |                    |                   |                    |                   |
| Female       | 27102.0                | 30437.0             | 33171.0            | 18561.0           | 30037.0            | 28575.0           |
| Male         | 27236.0                | 23868.0             | 22974.0            | 30633.0           | 23825.0            | 26548.0           |

Fonte: O autor

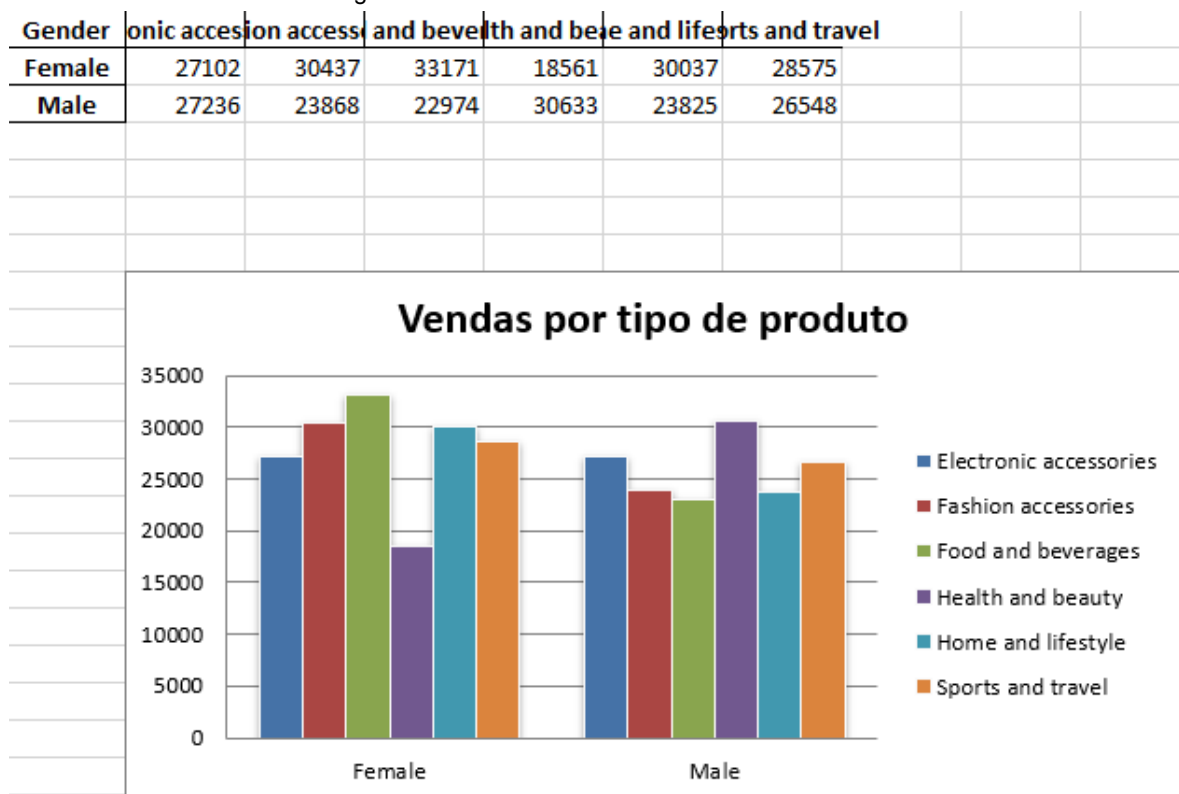
O próximo passo é atribuir dinamicamente o valor mínimo e máximo de colunas e linhas da planilha. Depois, vamos adicionar o gráfico de barras e atribuir os valores obtidos anteriormente para preencher o gráfico. O código é o seguinte:

```
barchart = BarChart()

data = Reference(sheet,
                  min_col=min_column+1,
                  max_col=max_column,
                  min_row=min_row,
                  max_row=max_row)
categories = Reference(sheet,
                       min_col=min_column,
                       max_col=min_column,
                       min_row=min_row+1,
                       max_row=max_row)
```

Depois desse bloco de código, temos o gráfico na planilha, visto na figura 2.

Figura 2 - Gráfico de barras adicionado no Excel



Fonte: O autor

Para fazermos a linha com o total somado entre homens e mulheres, temos o seguinte bloco de código:

Primeiro gerar um 'alfabeto' com as letras A-G para podermos manipular as células no Excel.

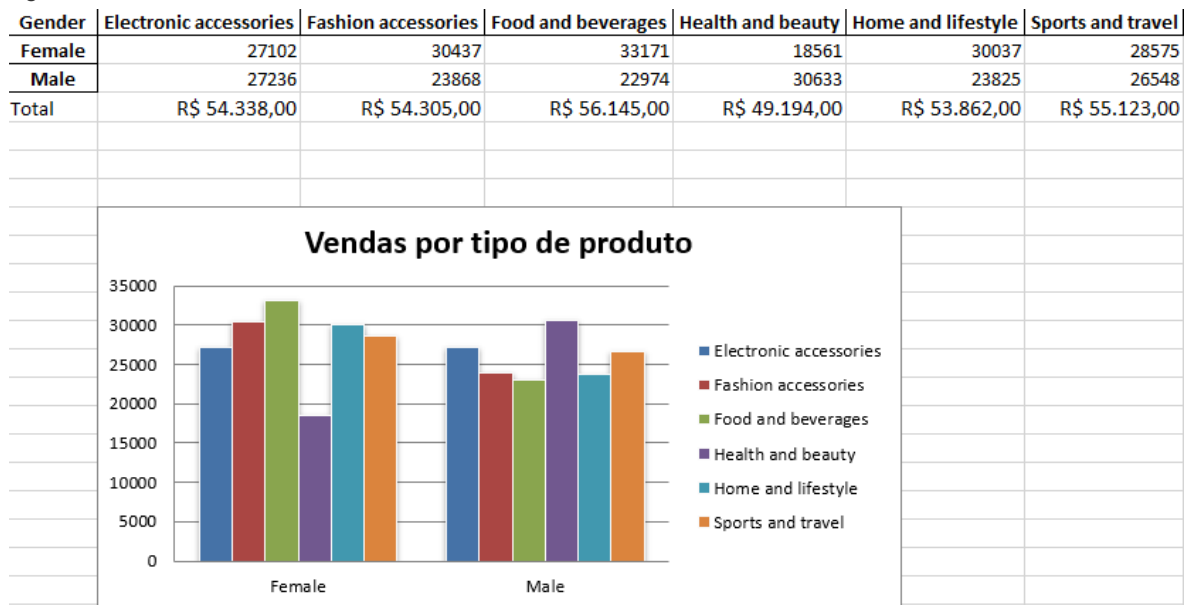
```
alphabet = list(string.ascii_uppercase)
excel_alphabet = alphabet[0:max_column]
print(excel_alphabet)
```

E, em seguida, um for loop para somar as células do total gasto por homens e mulheres.

```
for i in excel_alphabet:
    if i!='A':
        sheet2[f'{i}{max_row+1}'] = f'=SUM({i}{min_row+1}:{i}{max_row})'
        sheet2[f'{i}{max_row+1}'].number_format = 'R$ #,##0.00'
sheet[f'{excel_alphabet[0]}{max_row+1}'] = 'Total'
```

Na figura 3, temos o resultado do código.

Figura 3 - Planilha com a coluna 'total' adicionada



Fonte: O autor

Agora, para automatizarmos e poder fazer isso para todas as planilhas, temos que colocar todo o código numa função e aplicá-la para a planilha que deseja. O código é o seguinte:

```
def excel_auto(file_name):

    # ler o arquivo excel
    excel_f = pd.read_excel(file_name)

    # pivot table
    tabela1 = excel_f.pivot_table(index='Gender', columns='Product
line', values='Total', aggfunc='sum').round(0)

    # separando o nome do arquivo da sua extensao
    mes_extensao = file_name.split('_')[1]

    # salvando o arquivo como excel
    tabela1.to_excel(f'relatorio_{mes_extensao}',
sheet_name='vendas', startrow=4)

    # load no workbook e selecionando a aba
    wb = load_workbook(f'relatorio_{mes_extensao}')
    sheet = wb['vendas']
```

```

# referencia das celulas
min_column = wb.active.min_column
max_column = wb.active.max_column
min_row = wb.active.min_row
max_row = wb.active.max_row

# adicionando o grafico de barras
barchart = BarChart()
data = Reference(sheet,
                  min_col=min_column+1,
                  max_col=max_column,
                  min_row=min_row,
                  max_row=max_row)
categories = Reference(sheet,
                       min_col=min_column,
                       max_col=min_column,
                       min_row=min_row+1,
                       max_row=max_row)

# adicionando os dados e categorias
barchart.add_data(data, titles_from_data=True)
barchart.set_categories(categories)

sheet.add_chart(barchart, "B12") #location chart
barchart.title = 'Vendas por tipo de produto'
barchart.style = 10 #choose the chart style

# aplicando a formula (soma)
# criando o alfabeto para usar como referencia

alphabet = list(string.ascii_uppercase)
excel_alphabet = alphabet[0:max_column]
# somatoria nas colunas B-G
for i in excel_alphabet:
    if i!='A':
        sheet[f'{i}{max_row+1}'] =
f'=SUM({i}{min_row+1}:{i}{max_row})'
        sheet[f'{i}{max_row+1}'].number_format = 'R$ #,##0.00'
        sheet[f'{excel_alphabet[0]}{max_row+1}'] = 'Total'

# pegando o nome do mes/ano
mes = mes_extensao.split('.')[0]

# formatação do relatorio
sheet['A1'] = 'Relatório de Vendas'
sheet['A2'] = '2021'

```

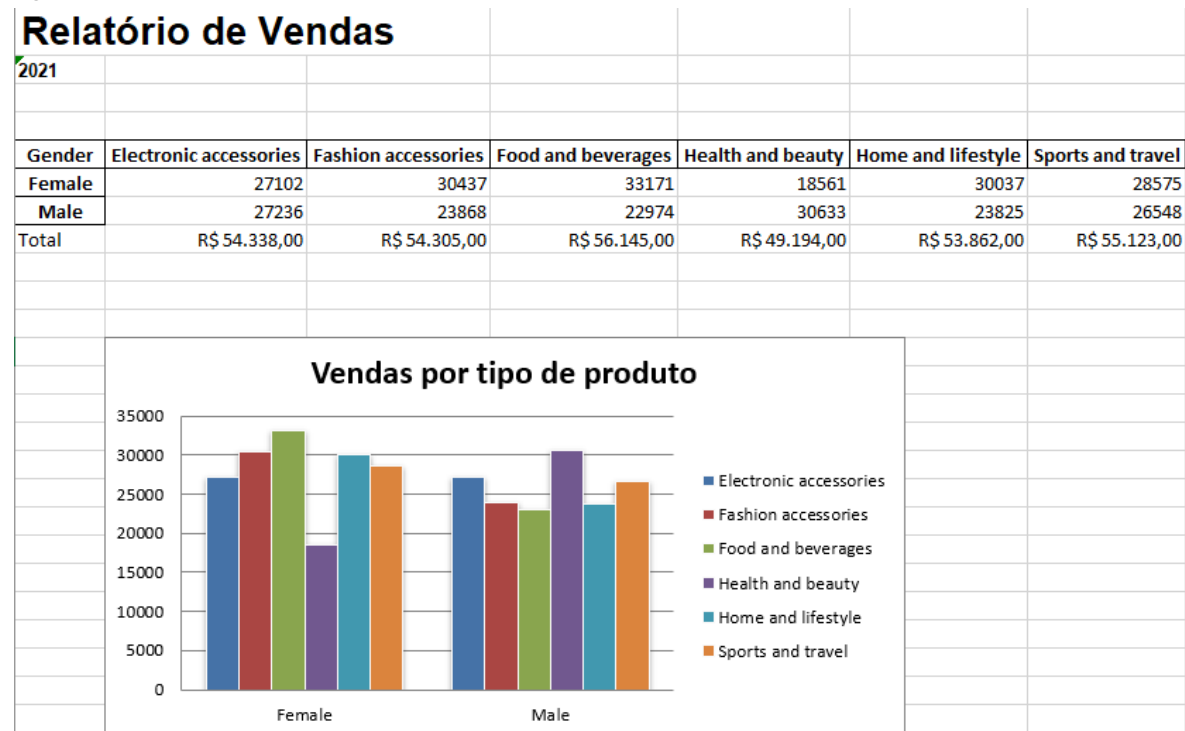
```

sheet['A1'].font = Font('Arial', bold=True, size=20)
sheet['A2'].font = Font('Arial', bold=True, size=10)
wb.save(f'relatorio_{mes_extensao}')
return

```

Acima, temos o código para gerar o gráfico de barras, coluna 'Total' e, também, para formatar o nome do arquivo a ser gerado. O resultado final é visto na figura 4.

Figura 4 - Resultado final



Fonte: O autor

Para termos o mesmo resultado da figura acima em qualquer planilha, apenas precisamos executar a função **excel\_auto**.

Vale lembrar que para utilizar o script que foi feito, os dados de entrada tem que estar dispostos da mesma maneira do arquivo original, o qual foi utilizado como base para o script.

Vou colocar 3 arquivos diferentes do original (com a mesma disposição) do arquivo original para vocês testarem, se quiserem.

Todo o código, planilhas geradas e esse pdf pode ser baixado no link do meu github.