

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
mercado = pd.read_csv('C:/Users/Rodrigo_df/Desktop/trabalhofaculdade/trabalho16.csv')
```

```
mercado.shape
```

```
(1000, 17)
```

```
mercado.head()
```

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.14
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.82
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.21
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23288.00
4	373-73-7010	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.20

```
mercado.columns
```

```
Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
      'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Date',
      'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
      'Rating'],
      dtype='object')
```

```
# Traduzindo o nome das colunas
```

```
mercado.columns = ['ID', 'Filial', 'Cidade', 'Tipo de cliente', 'Genero',
                  'Linha de produtos', 'Preçounitário', 'Quantidade', 'Imposto 5%', 'Total', 'Data',
                  'Tempo', 'Pagamento', 'engrenagens', 'porcentagem da margem bruta', 'receita bruta',
                  'Avaliação']
```

```
mercado.head()
```

	ID	Filial	Cidade	Tipo de cliente	Genero	Linha de produtos	Preçounitário	Quantidade
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8

Valor da coluna de Avaliação
mercado['Avaliação']

```
0      9.1
1      9.6
2      7.4
3      8.4
4      5.3
...
995    6.2
996    4.4
997    7.7
998    4.1
999    6.6
Name: Avaliação, Length: 1000, dtype: float64
```

mercado['Avaliação'].unique()

```
array([ 9.1,  9.6,  7.4,  8.4,  5.3,  4.1,  5.8,  8. ,  7.2,  5.9,  4.5,
        6.8,  7.1,  8.2,  5.7,  4.6,  6.9,  8.6,  4.4,  4.8,  5.1,  9.9,
        6. ,  8.5,  6.7,  7.7,  7.5,  7. ,  4.7,  7.6,  7.9,  6.3,  5.6,
        9.5,  8.1,  6.5,  6.1,  6.6,  5.4,  9.3, 10. ,  6.4,  4.3,  4. ,
        8.7,  9.4,  5.5,  8.3,  7.3,  4.9,  4.2,  9.2,  7.8,  5.2,  9. ,
        8.8,  6.2,  9.8,  9.7,  5. ,  8.9])
```

Media da Avaliação
mercado['Avaliação'].mean()

```
6.9727000000000003
```

mercado.describe()

	Preçounitário	Quantidade	Imposto 5%	Total	engrenagens	porcentagem da margem bruta
count	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000	1.000000e+03
mean	55.672130	5.510000	8938.880935	1.877165e+05	307.58738	4.761905e+01
std	26.494628	2.923431	11287.611805	2.370398e+05	234.17651	6.220360e+01
min	10.080000	1.000000	0.508500	1.067850e+01	10.17000	4.761905e+01

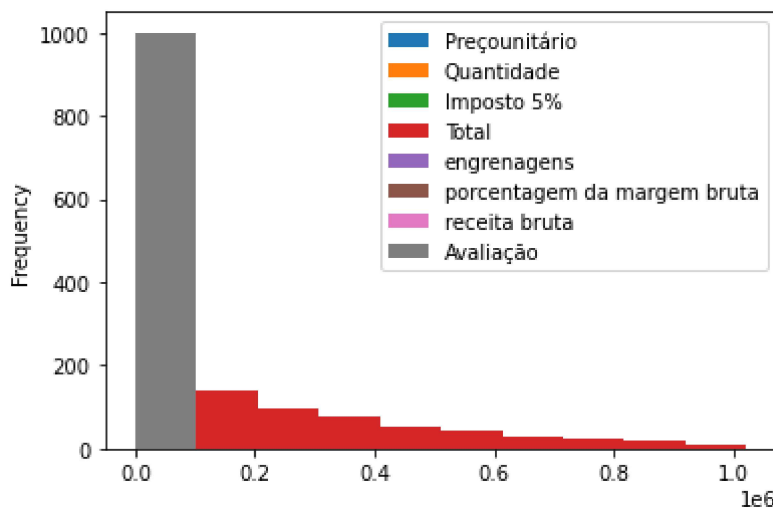
```
# importando uma nova biblioteca
from matplotlib import pyplot as plt
```

```
mercado.sum()
```

```
ID          750-67-8428226-31-3081631-41-3108123-19-117637...
Filial      ACAAACACABBBAAABAAABCBBAABABABBBACCAACBBCBCCB...
Cidade      YangonNaypyitawYangonYangonYangonNaypyitawYang...
Tipo de cliente  MemberNormalNormalMemberNormalNormalMemberNorm...
Genero      FemaleFemaleMaleMaleMaleMaleFemaleFemaleFemale...
Linha de produtos  Health and beautyElectronic accessoriesHome an...
Preçounitário          55672.1
Quantidade            5510
Imposto 5%           8.93888e+06
Total               1.87716e+08
Data          1/5/20193/8/20193/3/20191/27/20192/8/20193/25/...
Tempo         13:0810:2913:2320:3310:3718:3014:3611:3817:151...
Pagamento    EwalletCashCredit cardEwalletEwalletEwalletEwa...
engrenagens          307587
porcentagem da margem bruta  4761.9
receita bruta       8.93888e+06
Avaliação         6972.7
dtype: object
```

```
# Verificando um quadro geral dos dados na planilha
mercado.plot(kind='hist')
```

```
<AxesSubplot:ylabel='Frequency'>
```



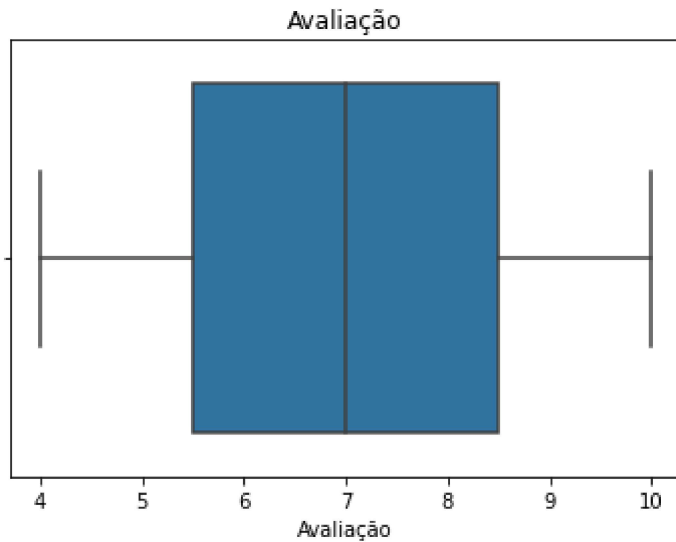
```
import seaborn as sns
import statistics as stc
```

```
import statistics as st
```

```
# Boxplot da coluna Avaliação
srn.boxplot(mercado['Avaliação']).set_title('Avaliação')
```

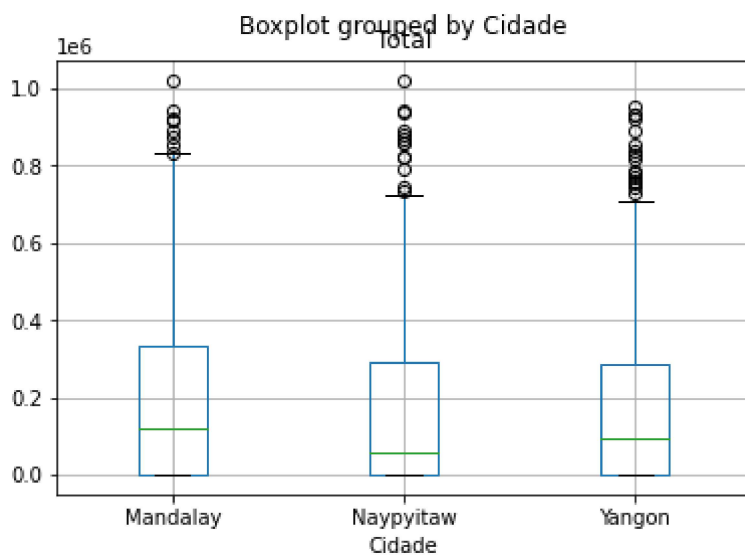


```
C:\Users\Rodrigo_df\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
warnings.warn(
Text(0.5, 1.0, 'Avaliação')
```



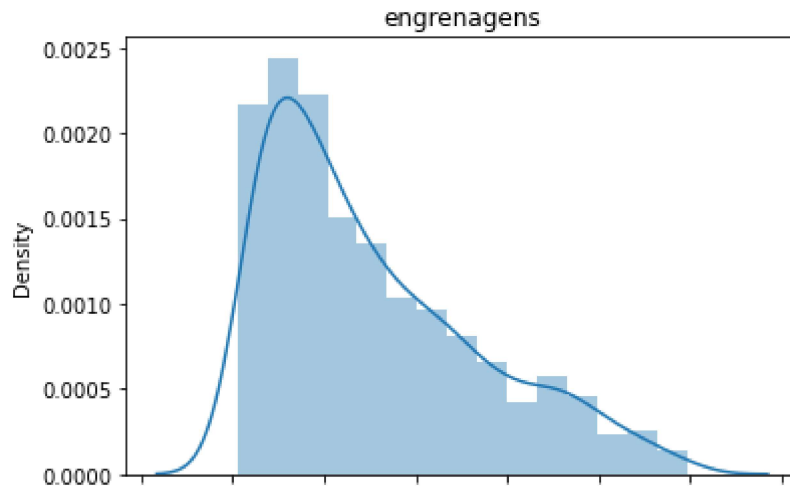
```
#Visualizando o gráfico de boxplot dos dados
mercado.boxplot(by='Cidade',column='Total')
```

```
<AxesSubplot:title={'center':'Total'}, xlabel='Cidade'>
```



```
# Gerando um Gráfico
srn.distplot(mercado['engrenagens']).set_title('engrenagens')
```

```
C:\Users\Rodrigo_df\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning.warn(msg, FutureWarning)
Text(0.5, 1.0, 'engrenagens')
```



```
mercado['Imposto 5%'].isnull().sum()
```

```
0
```

```
# Verificando se existem dados duplicados
mercado[mercado.duplicated(['ID'],keep = False)]
```

ID	Filial	Cidade	Tipo de cliente	Genero	Linha de produtos	Preçounitário	Quantidade	Imposto 5%
----	--------	--------	-----------------	--------	-------------------	---------------	------------	------------

```
# # Quantidade de Homens e Mulheres
agrupado = mercado.groupby(['Genero']).size()
agrupado
```

```
Genero
Female    501
Male      499
dtype: int64
```

```
# A diferença é mínima como pode ser visualizada no gráfico
agrupado.plot.bar(color = 'gray')
```

```
<AxesSubplot:xlabel='Tipo de cliente'>
```



```
agrupado = mercado.groupby(['Tipo de cliente']).size()
agrupado
```

```
Tipo de cliente
```

```
Member      501
```

```
Normal      499
```

```
dtype: int64
```



```
agrupado = mercado.groupby(['Linha de produtos']).size()
agrupado
```

```
Linha de produtos
```

```
Electronic accessories      170
```

```
Fashion accessories        178
```

```
Food and beverages         174
```

```
Health and beauty          152
```

```
Home and lifestyle          160
```

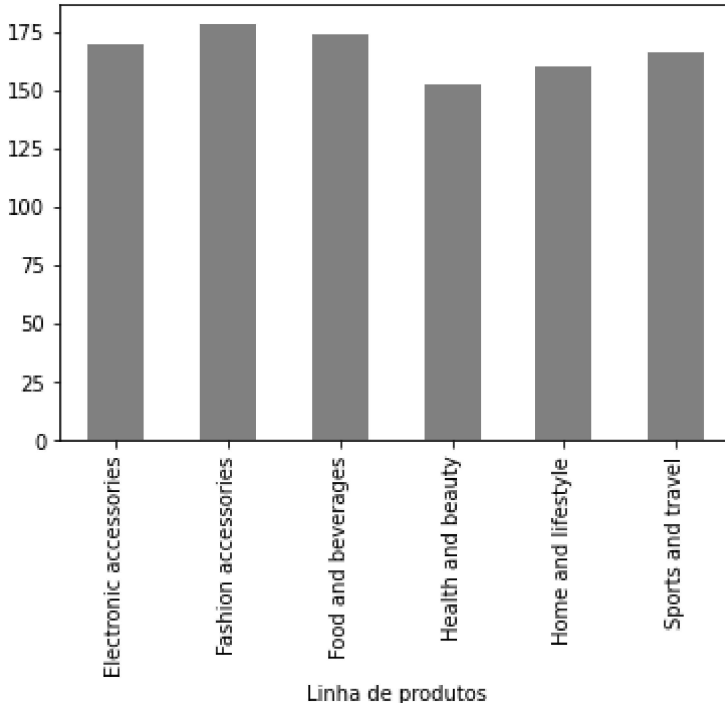
```
Sports and travel           166
```

```
dtype: int64
```

```
# Gráfico por quantidade de cada tipo de produto
```

```
agrupado.plot.bar(color = 'gray')
```

```
<AxesSubplot:xlabel='Linha de produtos'>
```



```
# Quantas compras foram registradas na base
```

```
agrupado = mercado.groupby(['ID']).size()
```

```
agrupado
```

```
ID
```

```
101-17-6199      1
```

```
101-81-4070      1
```

```

102-06-2002    1
102-77-2261    1
105-10-6182    1
..
894-41-5205    1
895-03-6665    1
895-66-0685    1
896-34-0956    1
898-04-2717    1
Length: 1000, dtype: int64

```

```

# Foi verificado que não temos números repetidos na Base
mercado['ID'].nunique()

```

```

1000

```

Tipo de célula incompatível Clique duas vezes para inspecionar/editar o conteúdo

```

# Criando uma atribuição para quantidade de compras feita por homens

```

```

mercado[(mercado['Genero'] == 'Male')]

```

```

# Contando quantas linhas possui o filtro realizado
mercado[(mercado['Genero'] == 'Male']]['ID'].count()

```

```

499

```

```

# Criando uma atribuição para quantidade de compras feita por mulheres

```

```

mercado[(mercado['Genero'] == 'Female')]

```

```

# Contando quantas linhas possui o filtro realizado
mercado[(mercado['Genero'] == 'Female']]['ID'].count()

```

```

501

```

```

# Filial que teve mais vendas efetuadas

```

```

mercado['Filial'].value_counts()

```

```

A    340
B    332
C    328
Name: Filial, dtype: int64

```

```

# Testando outra forma de fazer a mesma contagem
agrupado = mercado.groupby(['Filial']).size()
agrupado

```

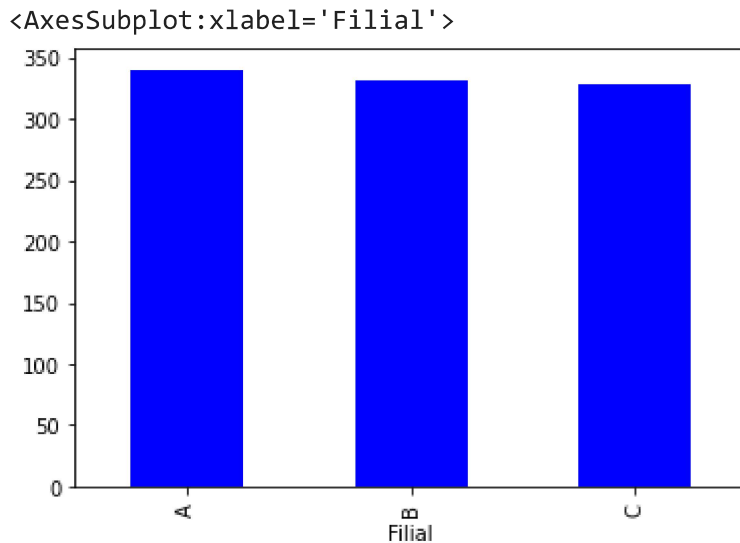
```

Filial
A    340
B    332

```

```
C      328
dtype: int64
```

```
# Gráfico para análise
agrupado.plot.bar(color = 'blue')
```



```
# Qual filial as mulheres e homens fizeram mais compras
```

```
mercado.groupby(['Filial','Genero'])['ID'].count()
```

```
Filial  Genero
A      Female    161
      Male      179
B      Female    162
      Male      170
C      Female    178
      Male      150
Name: ID, dtype: int64
```

```
# Valor medio de compras na cidade Mandalay
```

```
ds_temp = mercado[(mercado['Cidade'] == 'Mandalay')]
```

```
ds_temp[(mercado['Tipo de cliente'] == 'Member')]['Total'].describe()
```

```
<ipython-input-79-68873ba6e6d1>:5: UserWarning: Boolean Series key will be reindexed
ds_temp[(mercado['Tipo de cliente'] == 'Member')]['Total'].describe()
count      165.000000
mean      189671.383873
std       238655.073134
min        20.107500
25%       411.379500
50%      91056.000000
75%     317835.000000
max     944622.000000
Name: Total, dtype: float64
```

```
import requests
from io import StringIO
```



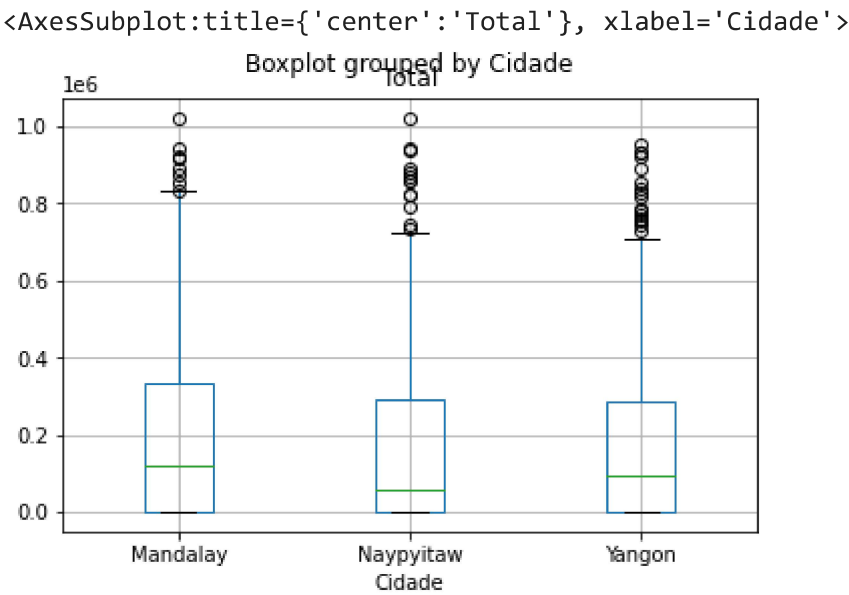
```
mercado.groupby(['Cidade'])['Total'].describe()
```

	count	mean	std	min	25%	50%	75%
Cidade							
Mandalay	332.0	202826.115361	243138.062218	18.6375	422.428125	117484.5	333490.5
Naypyitaw	328.0	173518.808675	233878.059940	10.6785	343.573125	55461.0	293564.2
Yangon	340.0	186659.000263	233830.604161	12.6945	363.720000	92778.0	287295.7

Tipo de célula incompatível Clique duas vezes para inspecionar/editar o conteúdo

Tipo de célula incompatível Clique duas vezes para inspecionar/editar o conteúdo

```
mercado.boxplot(by='Cidade',column='Total')
```



```
# Verificando os tipos de pagamentos
mercado.groupby(['Pagamento'])['ID'].count()
```

```
Pagamento
Cash          344
Credit card   311
Ewallet       345
Name: ID, dtype: int64
```

```
NameError                                Traceback (most recent call last)
<ipython-input-86-7be470d7d60e> in <module>
----> 1 mercado_consults[10:3]
```

