

In [ ]:

In [ ]: *# Dados de vendas de 3 filiais de um Supermercado por 3 meses, fazer uma análise*

```
In [1]: #base para o pydatasteak
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns

#escolher cores diferentes com aparencia profissional
import colorsys
```

```
In [2]: # usar o estilo seaborn - talk - fontes, bordas, legendas etc.
plt.style.use('seaborn-talk')

# import e filters, onde no futuro va mudar alguma coisa, para nao ver as mensagens
import warnings
warnings.filterwarnings('ignore')

#para executar o grafico aqui no notebook mesmo:
%matplotlib inline
```

```
In [3]: # importar o arquivo e visualizar na tela:
dataset = pd.read_csv('supermarket_sales - Sheet1.csv')
dataset
```

Out[3]:

	Invoice ID	Branch	City	Customertype	Gender	Productline	Unitprice	Quantity	Tax 5%
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085
...	...	...	...	...	...	...	...	...	...
995	233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1	2.0175
996	303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10	48.6900
997	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	1.5920
998	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1	3.2910
999	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7	30.9190

1000 rows × 17 columns



In [ ]:

```
In [6]: # Resumo estatístico do dataset, qtd para cada uma das colunas, Md, Dp, Min, Max e
# de distribuição
print(dataset.describe())
```

	Unitprice	Quantity	Tax 5%	Total	cogs \
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	55.672130	5.510000	15.379369	322.966749	307.58738
std	26.494628	2.923431	11.708825	245.885335	234.17651
min	10.080000	1.000000	0.508500	10.678500	10.17000
25%	32.875000	3.000000	5.924875	124.422375	118.49750
50%	55.230000	5.000000	12.088000	253.848000	241.76000
75%	77.935000	8.000000	22.445250	471.350250	448.90500
max	99.960000	10.000000	49.650000	1042.650000	993.00000

	gross margin percentage	gross income	Rating
count	1.000000e+03	1000.000000	1000.000000
mean	4.761905e+00	15.379369	6.97270
std	6.220360e-14	11.708825	1.71858
min	4.761905e+00	0.508500	4.00000
25%	4.761905e+00	5.924875	5.50000
50%	4.761905e+00	12.088000	7.00000
75%	4.761905e+00	22.445250	8.50000
max	4.761905e+00	49.650000	10.00000

```
In [7]: # verificar o formato do dataset:
dataset.shape
```

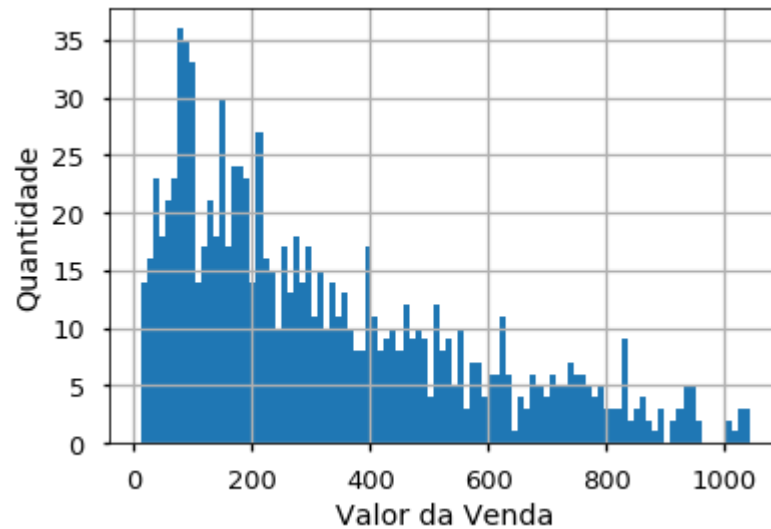
```
Out[7]: (1000, 17)
```

```
In [8]: #verificar se ha linhas vazias
dataset.isnull().sum()
```

```
Out[8]: Invoice ID      0
Branch      0
City        0
Customertype  0
Gender      0
Productline  0
Unitprice   0
Quantity    0
Tax 5%      0
Total       0
Date        0
Time        0
Payment     0
cogs        0
gross margin percentage  0
gross income  0
Rating      0
dtype: int64
```

```
In [9]: # Gerar um Histograma com a Distribuição das Vendas por quantidades e valores de  
# Possivel verificar onde está a concentração por valor.
```

```
dataset.Total.hist( bins = 100)  
plt.xlabel('Valor da Venda')  
plt.ylabel('Quantidade')  
plt.show()
```



```
In [35]: # Verificar a distribuição das vendas por sexo, onde está a maior concentração:
```

```
# para definir a quantidade:
labels = dataset.Gender.value_counts().index
num = len(dataset.Total.value_counts().index)

# Criar uma lista de cores:

# Criar o Gráfico de Pizza:
#definir as fatias e o texto
fatias, texto = plt.pie(dataset.Gender.value_counts( ))
#aspectos para os eixos.
plt.axes( ).set_aspect('equal','datalim')
# legenda ao lado do grafico
plt.legend(fatias,labels, bbox_to_anchor = (1.8,1))
plt.title('Distribuição das vendas por sexo')
plt.show()
```

Distribuição das vendas por sexo



Female  
Male

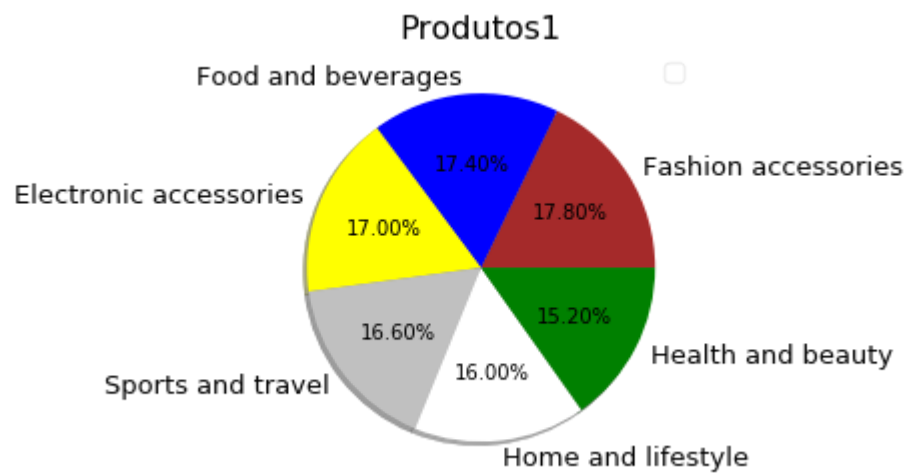
```
In [37]: dataset['Productline'].value_counts()
```

```
Out[37]: Fashion accessories      178
Food and beverages               174
Electronic accessories           170
Sports and travel                166
Home and lifestyle               160
Health and beauty               152
Name: Productline, dtype: int64
```

```
In [38]: # Participação das vendas por produtos:

fatias = [178, 174, 170, 166, 160, 152]
labels = "Fashion accessories", "Food and beverages", "Electronic accessories", "Sports and travel", "Home and lifestyle", "Health and beauty"

colors = ['brown', 'blue', 'yellow', 'silver', 'white', 'green']
explode = [0, 0, 0, 0, 0, 0]
plt.pie(fatias, labels = labels, colors = colors, explode = explode, shadow = True)
plt.title('Produtos1', fontsize = 16)
plt.axis('off')
plt.legend('')
plt.show()
```



In [41]: *# Quais os principais produtos vendidos*

*#Definir a quantidade*

```
num = len(dataset.Productline.value_counts().index)
```

```
label = dataset.Productline.value_counts( ).index
```

```
colors = ['Orange', 'red', 'blue', 'green', 'brouw', 'silver', 'black']
```

*#Gráfico de Pizza*

```
fatias, texto=plt.pie(dataset.Productline.value_counts(), startangle = 90)
```

```
plt.axes().set_aspect('equal','datalim')
```

```
plt.legend(fatias,lebel, bbox_to_anchor =(1.5,1))
```

```
plt.title('Produtos')
```

```
plt.show()
```

**NameError**

Traceback (most recent call last)

<ipython-input-41-7ba2c077fd69> in <module>

```
10 fatias, texto=plt.pie(dataset.Productline.value_counts(), startangle =  
90)
```

```
11 plt.axes().set_aspect('equal','datalim')
```

```
---> 12 plt.legend(fatias,lebel, bbox_to_anchor =(1.5,1))
```

```
13 plt.title('Produtos')
```

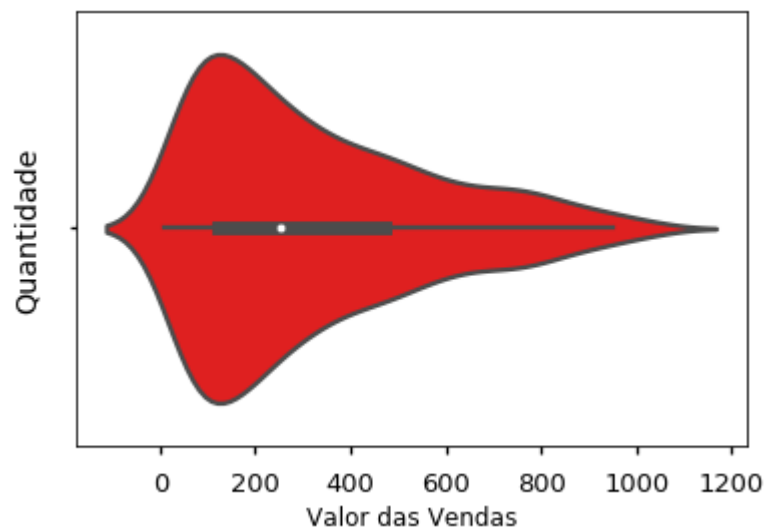
```
14 plt.show()
```

**NameError**: name 'lebel' is not defined



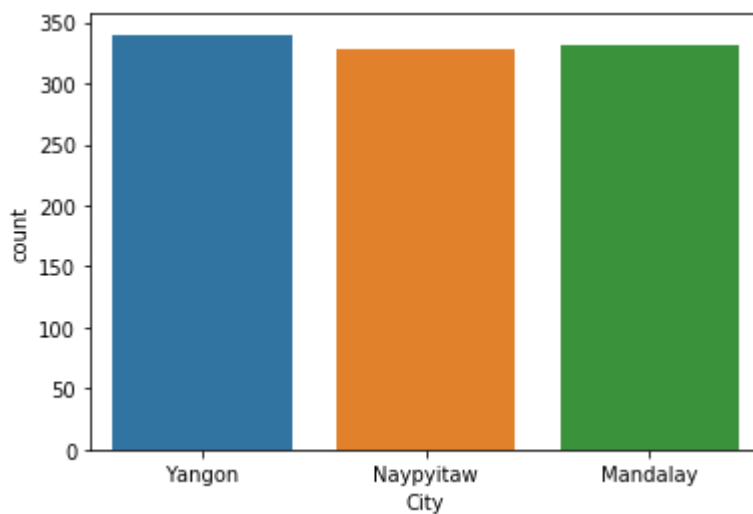
In [ ]:

```
In [52]: sns.violinplot(dataset['Total'],color = 'red')
plt.title('', fontsize = 12)
plt.xlabel('Valor das Vendas', fontsize = 12)
plt.ylabel('Quantidade')
plt.show()
```



```
In [13]: sns.countplot(dataset['City'])
```

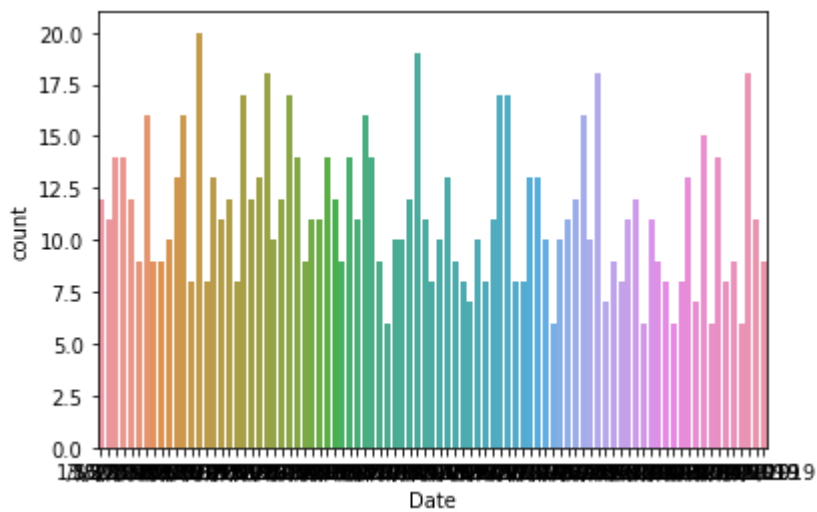
Out[13]: <matplotlib.axes.\_subplots.AxesSubplot at 0x28229e9a548>





```
In [14]: sns.countplot(dataset['Date'])
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x28229f06508>
```



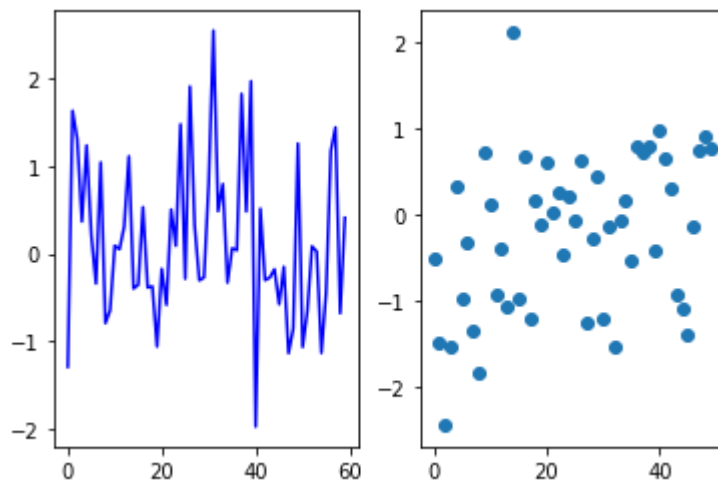
```
In [17]: %matplotlib inline
```

```
fig = plt.figure()
ax1= plt.figure()

ax1 = fig.add_subplot(1,2,1)
ax1.plot(np.random.randn(60), color = 'blue')

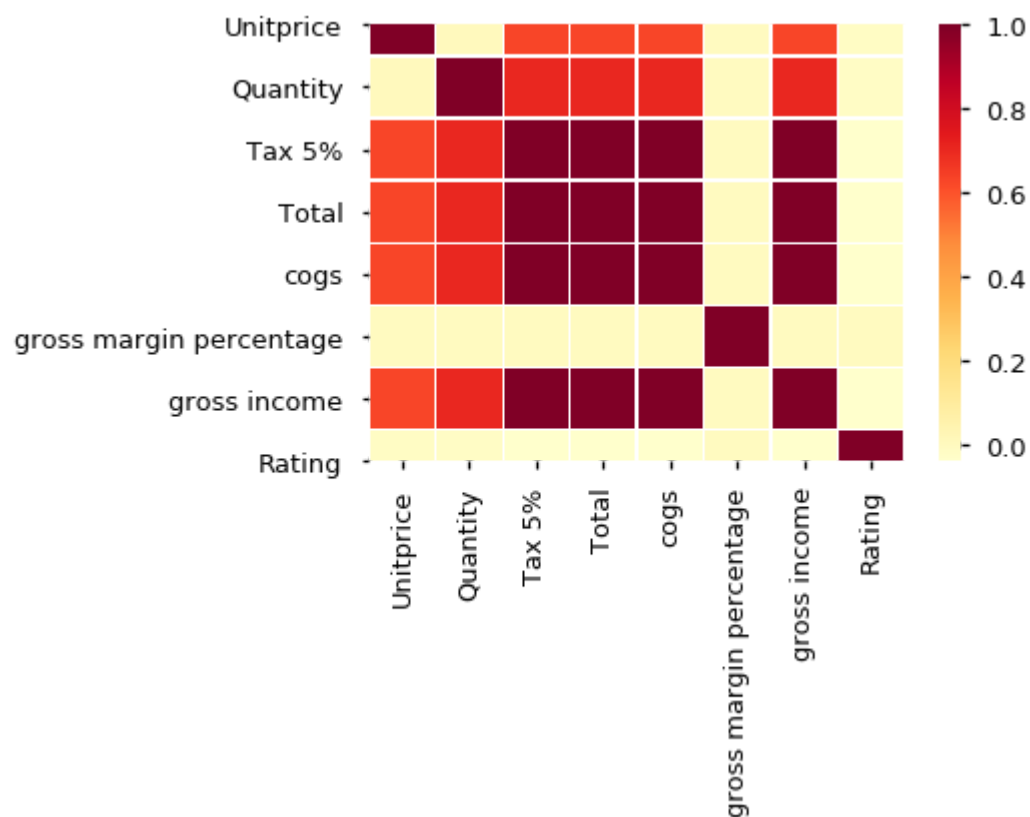
ax2 = fig.add_subplot( 1,2,2)
ax2.scatter(np.arange(50), np.random.randn(50))

plt.show()
```



```
<Figure size 432x288 with 0 Axes>
```

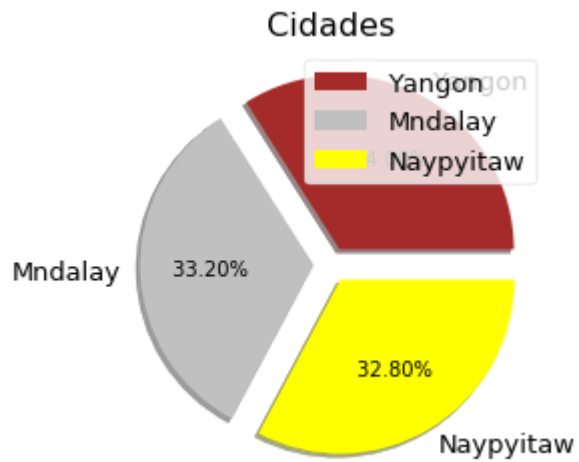
```
In [60]: corr = dataset.corr()
sns.heatmap(corr, cmap = "YlOrRd",linewidths = 0.2)
plt.show()
```



```
In [19]: dataset['City'].value_counts()
```

```
Out[19]: Yangon      340
Mandalay    332
Naypyitaw   328
Name: City, dtype: int64
```

```
In [59]: fatias = [340, 332, 328]
labels = "Yangon", "Mndalay", "Naypyitaw"
colors = ['brown', 'silver', 'yellow']
explode = [0.1, 0.1, 0.1]
plt.pie(fatias, labels = labels, colors = colors, explode = explode, shadow = True)
plt.title(' Cidades ', fontsize = 16)
plt.axis('on')
plt.legend()
plt.show()
```



In [ ]: