In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import calmap
from pandas_profiling import ProfileReport
```

In [2]:

```
df=pd.read_csv(r'C:\Users\sonim\OneDrive\Desktop\Projects\Supermarket EDA\archive (3)/super
```

In [74]:

```
dataset=pd.read_csv(r'C:\Users\sonim\OneDrive\Desktop\Projects\Supermarket EDA\archive (3)/
prof=ProfileReport(dataset)
prof
```

100% 2.51s/it]

In [3]:

```
df.head()
```

Out[3]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	To
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.97
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.22
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.52
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.04
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.37

In [4]:

```
df.tail()
```

Out[4]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	
995	233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1	2.0175	4
996	303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10	48.6900	102
997	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	1.5920	3
998	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1	3.2910	6
999	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7	30.9190	64

In [5]:

```
df.shape
```

Out[5]:

```
(1000, 17)
```

In [6]:

```
df.columns
```

Out[6]:

```
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')
```

In [7]:

```
df.dtypes
```

Out[7]:

```
Invoice ID                 object
Branch                     object
City                       object
Customer type              object
Gender                     object
Product line               object
Unit price                float64
Quantity                    int64
Tax 5%                    float64
Total                     float64
Date                       object
Time                       object
Payment                    object
cogs                      float64
gross margin percentage   float64
gross income              float64
Rating                    float64
dtype: object
```

In [8]:

```
df['Date']=pd.to_datetime(df['Date'])
```

In [9]:

```
df.set_index('Date', inplace=True)
```

In [10]:

```
df.head()
```

Out[10]:

Date	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	
2019-01-05	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	54
2019-03-08	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	8
2019-03-03	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	34
2019-01-27	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	48
2019-02-08	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	63

In [11]:

```
df.describe()
```

Out[11]:

	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gr inc
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1.000000e+03	1000.000
mean	55.672130	5.510000	15.379369	322.966749	307.58738	4.761905e+00	15.379
std	26.494628	2.923431	11.708825	245.885335	234.17651	6.220360e-14	11.708
min	10.080000	1.000000	0.508500	10.678500	10.17000	4.761905e+00	0.508
25%	32.875000	3.000000	5.924875	124.422375	118.49750	4.761905e+00	5.924
50%	55.230000	5.000000	12.088000	253.848000	241.76000	4.761905e+00	12.088
75%	77.935000	8.000000	22.445250	471.350250	448.90500	4.761905e+00	22.445
max	99.960000	10.000000	49.650000	1042.650000	993.00000	4.761905e+00	49.650

Question 1: What does the distribution of customer rating look like? And is it skwed?

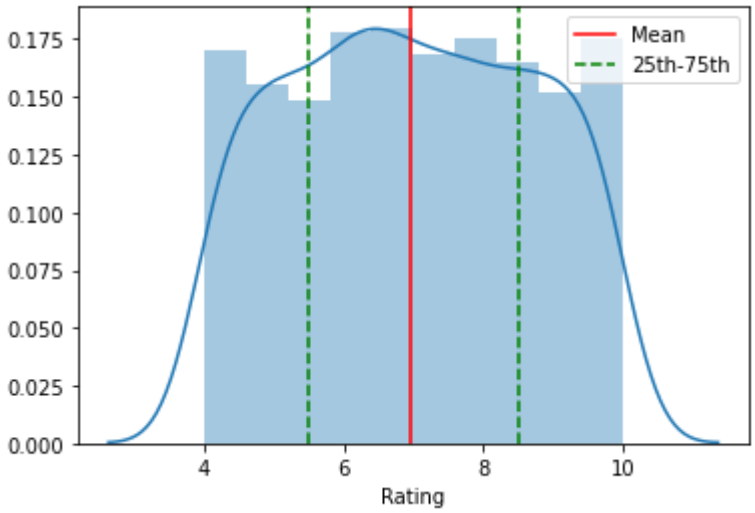
Rating: Customer stratification rating on their overall shopping experience (On a scale of 1 to 10)

In [12]:

```
sns.distplot(df['Rating'])
plt.axvline(x=np.mean(df['Rating']), color="red", label='Mean')
plt.axvline(x=np.percentile(df['Rating'], 25), color='Green', ls='--', label="25th-75th")
plt.axvline(x=np.percentile(df['Rating'], 75), color='Green', ls='--')
plt.legend()
```

Out[12]:

```
<matplotlib.legend.Legend at 0x121b0b80>
```



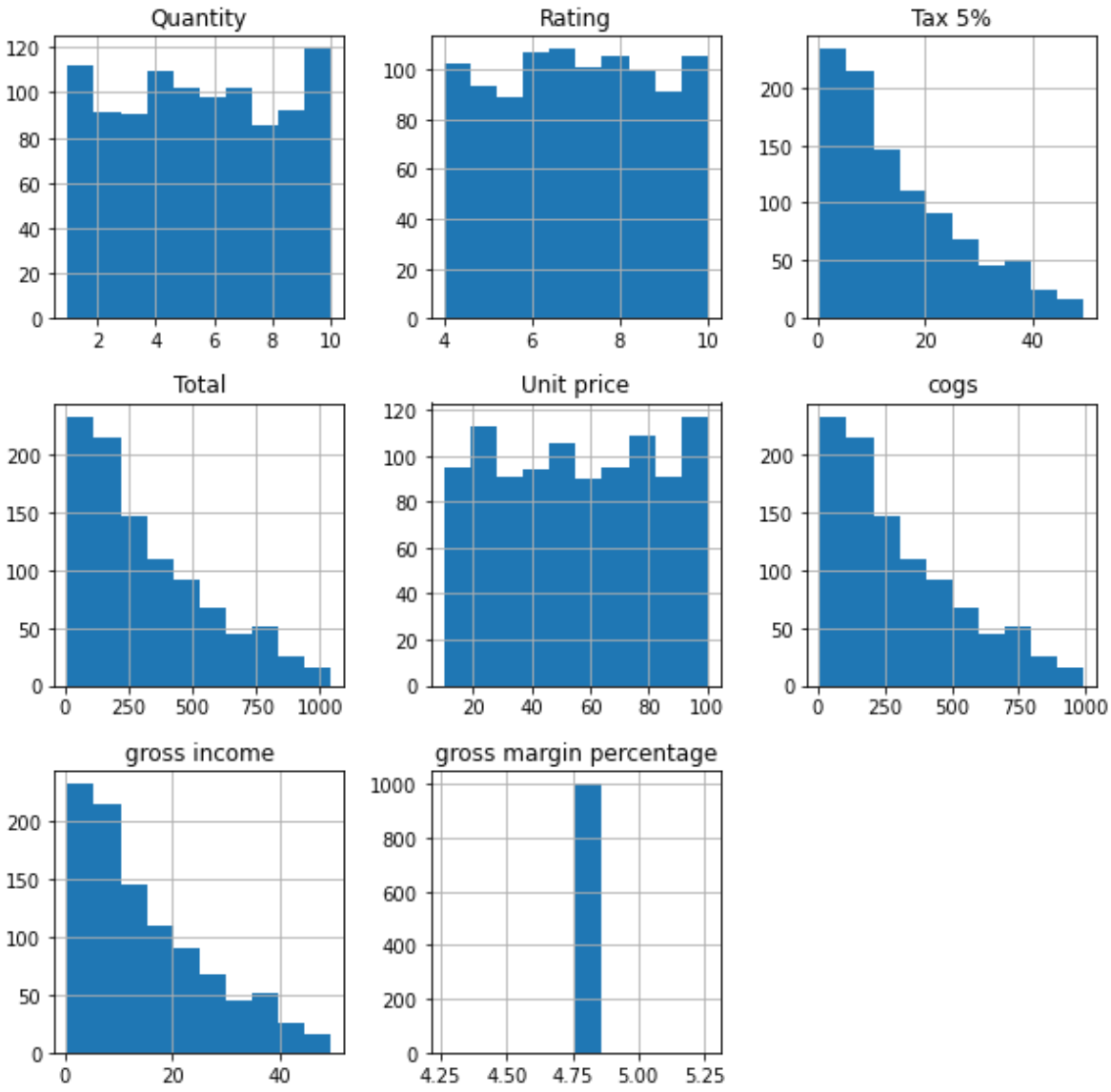
The figure above shows no skewness and hence customers are equally likely to give any ratings.

In [13]:

```
df.hist(figsize=(10,10))
```

Out[13]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x11888460>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x122753A0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12287D90>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x122A3790>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x122C2190>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x122D33B8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x122D3B80>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x122F65B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1232B1C0>]],
      dtype=object)
```

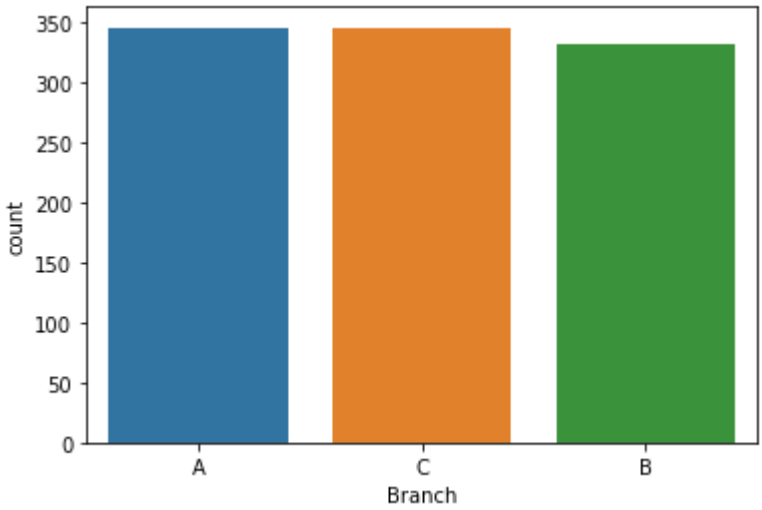


In [15]:

```
sns.countplot(df['Payment'])
sns.countplot(df['Branch'])
```

Out[15]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1267aa30>
```

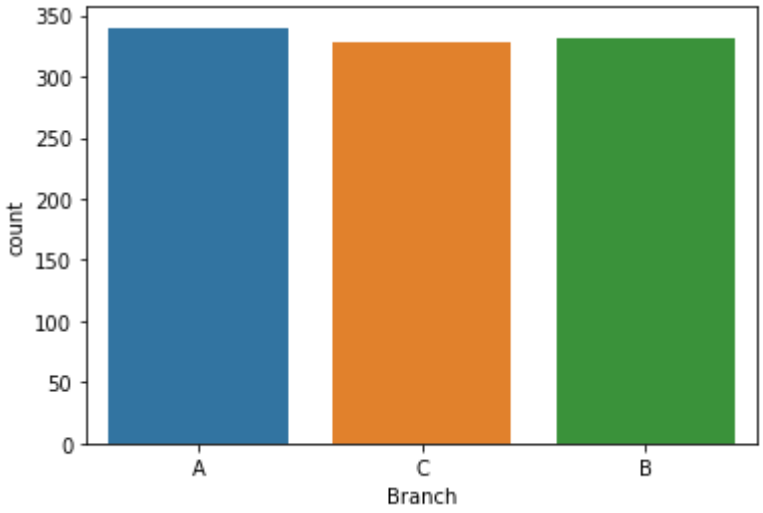


In [16]:

```
sns.countplot(df['Branch'])
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x12827340>
```



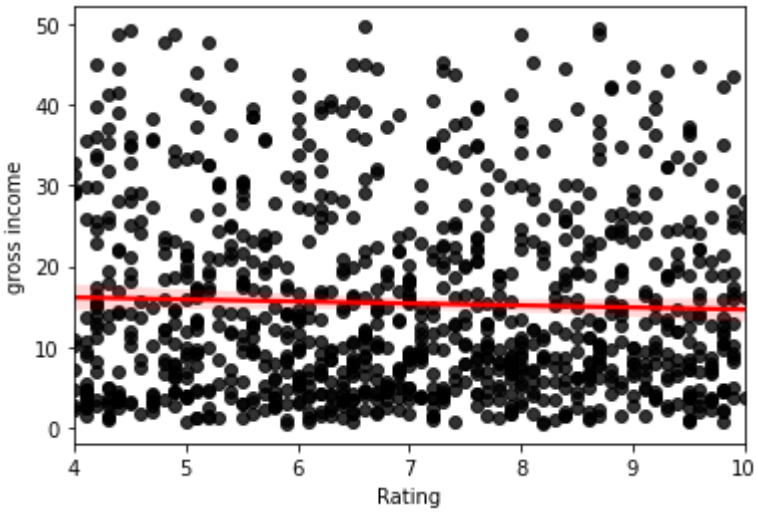
Question 2 is there a relationship between customer ratings and gross income?

In [28]:

```
#sns.scatterplot(df['Rating'], df['gross income'])
sns.regplot(df['Rating'], df['gross income'],scatter_kws={"color": "black"}, line_kws={"col
sns.regplot
```

Out[28]:

```
<function seaborn.regression.regplot(x, y, data=None, x_estimator=None, x_bi
ns=None, x_ci='ci', scatter=True, fit_reg=True, ci=95, n_boot=1000, units=No
ne, seed=None, order=1, logistic=False, lowess=False, robust=False, logx=Fal
se, x_partial=None, y_partial=None, truncate=True, dropna=True, x_jitter=Non
e, y_jitter=None, label=None, color=None, marker='o', scatter_kws=None, line
_kws=None, ax=None)>
```



clearly there is no relationship whatsoever, between the two variables.

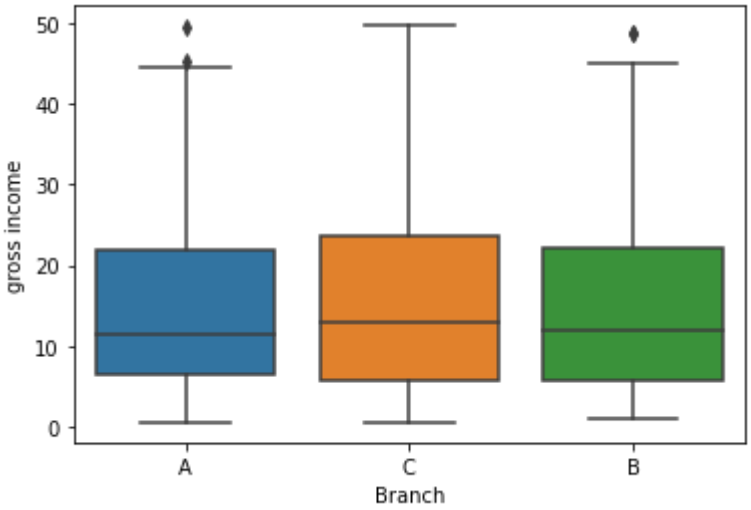
Relationship between Branch and gross income

In [33]:

```
sns.boxplot(df['Branch'], df['gross income'])
```

Out[33]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1429fef8>
```



Median lines of A and B are similar just above 10, and for C its above A and B. Thus there is not much variation between the three branches and gross incomes.

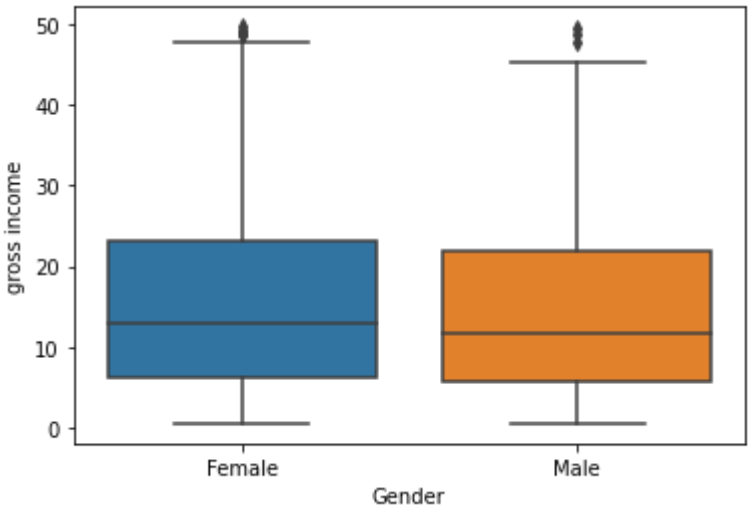
Gender and gross income

In [34]:

```
sns.boxplot(df['Gender'], df['gross income'])
```

Out[34]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1425cf10>
```



while on 75 percentile female spend more than men, but on an overage level, the two genders seem to spend at pretty similar scale.

In [36]:

```
df.groupby(df.index).mean()
#With this we are able to extract unique date values and every variable now is of the avera
```

Out[36]:

Date	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
2019-01-01	54.995833	6.750000	18.830083	395.431750	376.601667	4.761905	18.830083	6.583333
2019-01-02	44.635000	6.000000	11.580375	243.187875	231.607500	4.761905	11.580375	6.050000
2019-01-03	59.457500	4.625000	12.369813	259.766062	247.396250	4.761905	12.369813	8.112500
2019-01-04	51.743333	5.333333	12.886417	270.614750	257.728333	4.761905	12.886417	6.516667
2019-01-05	61.636667	4.583333	14.034458	294.723625	280.689167	4.761905	14.034458	7.433333
...	...	...	...	...	...	...	...	...
2019-03-26	42.972308	4.000000	7.188692	150.962538	143.773846	4.761905	7.188692	6.623077
2019-03-27	56.841000	4.500000	13.822950	290.281950	276.459000	4.761905	13.822950	6.760000
2019-03-28	45.525000	4.800000	10.616200	222.940200	212.324000	4.761905	10.616200	7.050000
2019-03-29	66.346250	6.750000	23.947875	502.905375	478.957500	4.761905	23.947875	6.925000
2019-03-30	67.408182	6.090909	19.424500	407.914500	388.490000	4.761905	19.424500	6.800000

89 rows × 8 columns

In [47]:

```
df.groupby(df.index).mean().index
```

Out[47]:

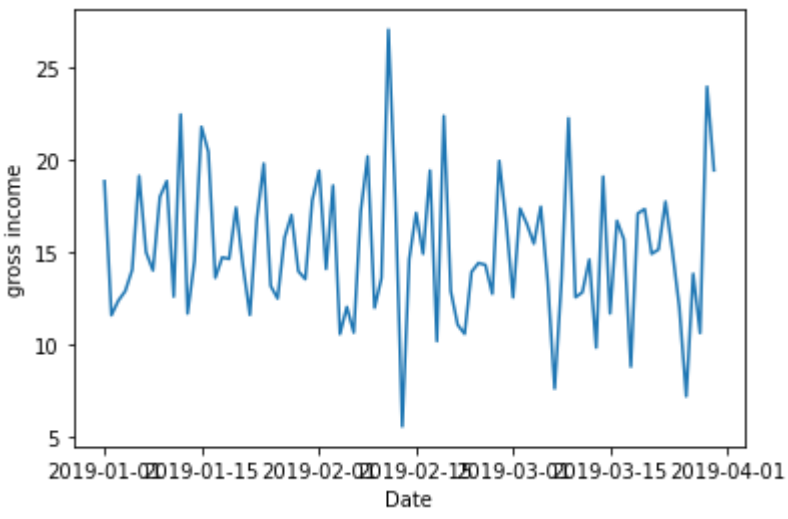
```
DatetimeIndex(['2019-01-01', '2019-01-02', '2019-01-03', '2019-01-04',
               '2019-01-05', '2019-01-06', '2019-01-07', '2019-01-08',
               '2019-01-09', '2019-01-10', '2019-01-11', '2019-01-12',
               '2019-01-13', '2019-01-14', '2019-01-15', '2019-01-16',
               '2019-01-17', '2019-01-18', '2019-01-19', '2019-01-20',
               '2019-01-21', '2019-01-22', '2019-01-23', '2019-01-24',
               '2019-01-25', '2019-01-26', '2019-01-27', '2019-01-28',
               '2019-01-29', '2019-01-30', '2019-01-31', '2019-02-01',
               '2019-02-02', '2019-02-03', '2019-02-04', '2019-02-05',
               '2019-02-06', '2019-02-07', '2019-02-08', '2019-02-09',
               '2019-02-10', '2019-02-11', '2019-02-12', '2019-02-13',
               '2019-02-14', '2019-02-15', '2019-02-16', '2019-02-17',
               '2019-02-18', '2019-02-19', '2019-02-20', '2019-02-21',
               '2019-02-22', '2019-02-23', '2019-02-24', '2019-02-25',
               '2019-02-26', '2019-02-27', '2019-02-28', '2019-03-01',
               '2019-03-02', '2019-03-03', '2019-03-04', '2019-03-05',
               '2019-03-06', '2019-03-07', '2019-03-08', '2019-03-09',
               '2019-03-10', '2019-03-11', '2019-03-12', '2019-03-13',
               '2019-03-14', '2019-03-15', '2019-03-16', '2019-03-17',
               '2019-03-18', '2019-03-19', '2019-03-20', '2019-03-21',
               '2019-03-22', '2019-03-23', '2019-03-24', '2019-03-25',
               '2019-03-26', '2019-03-27', '2019-03-28', '2019-03-29',
               '2019-03-30'],
              dtype='datetime64[ns]', name='Date', freq=None)
```

In [48]:

```
x=df.groupby(df.index).mean().index
y=df.groupby(df.index).mean()['gross income']
sns.lineplot(x,y)
```

Out[48]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1454a268>
```

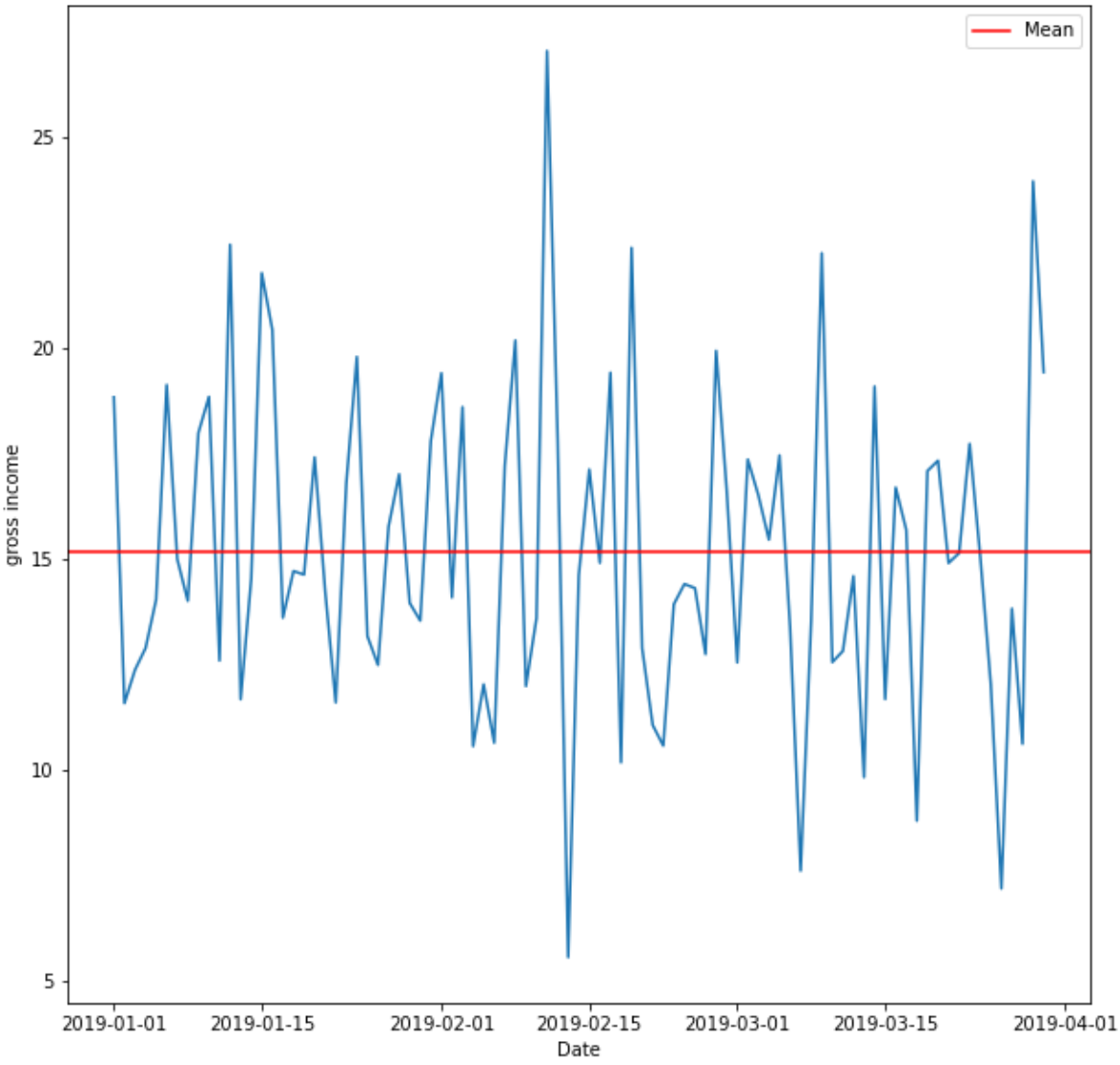


In [55]:

```
plt.figure(figsize=(10,10))
sns.lineplot(df.groupby(df.index).mean().index, df.groupby(df.index).mean()['gross income']
plt.axhline(y=(df.groupby(df.index).mean()['gross income']).mean(), color='Red', label='Mea
plt.legend()
```

Out[55]:

```
<matplotlib.legend.Legend at 0x125a0da8>
```

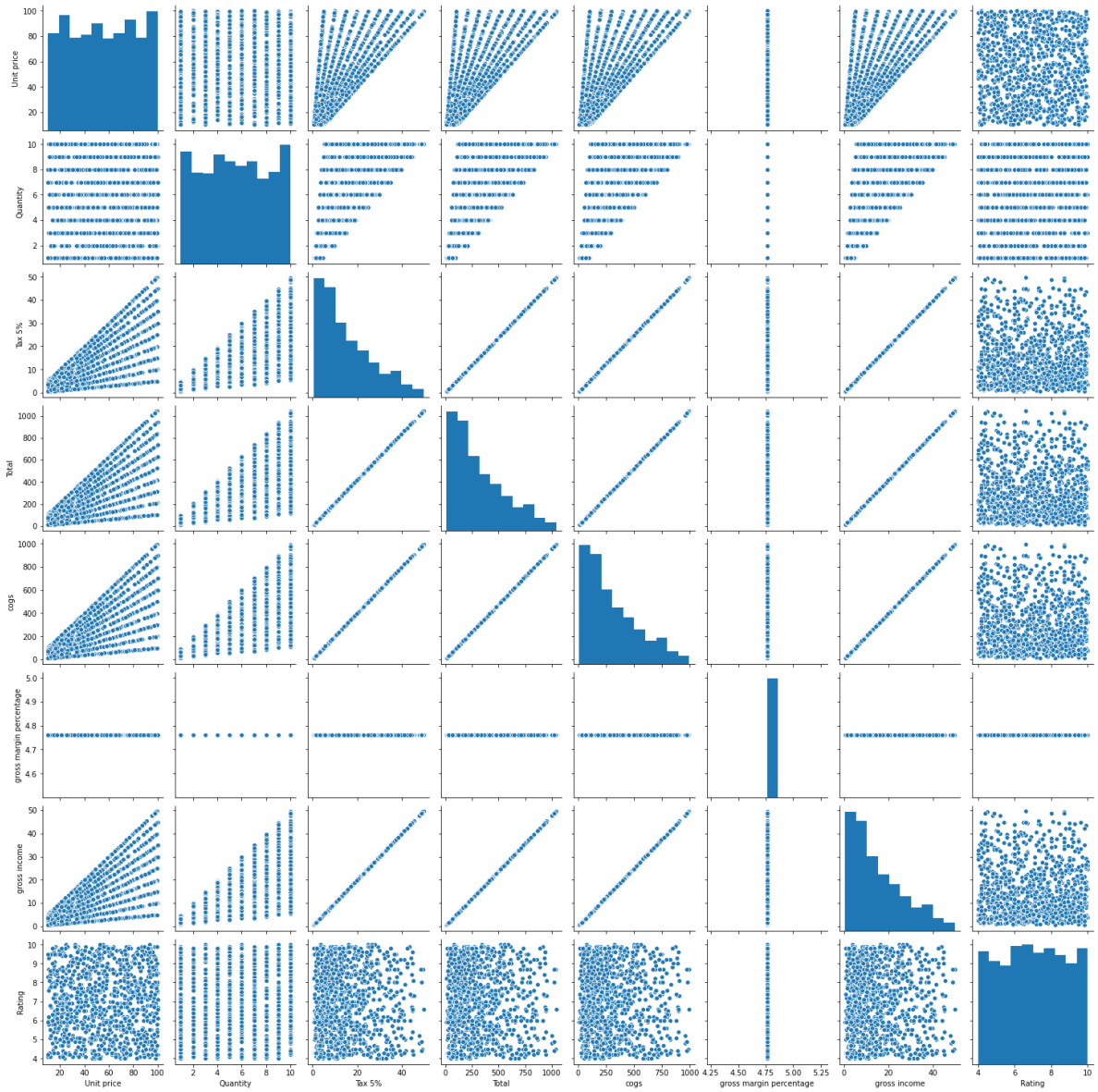


In [56]:

```
sns.pairplot(df)
```

Out[56]:

```
<seaborn.axisgrid.PairGrid at 0x14b7c0a0>
```



In [63]:

```
df.duplicated().sum()
#df.drop_duplicated(inplace=True)
```

Out[63]:

```
0
```

In [64]:

```
df.isna().sum()
```

Out[64]:

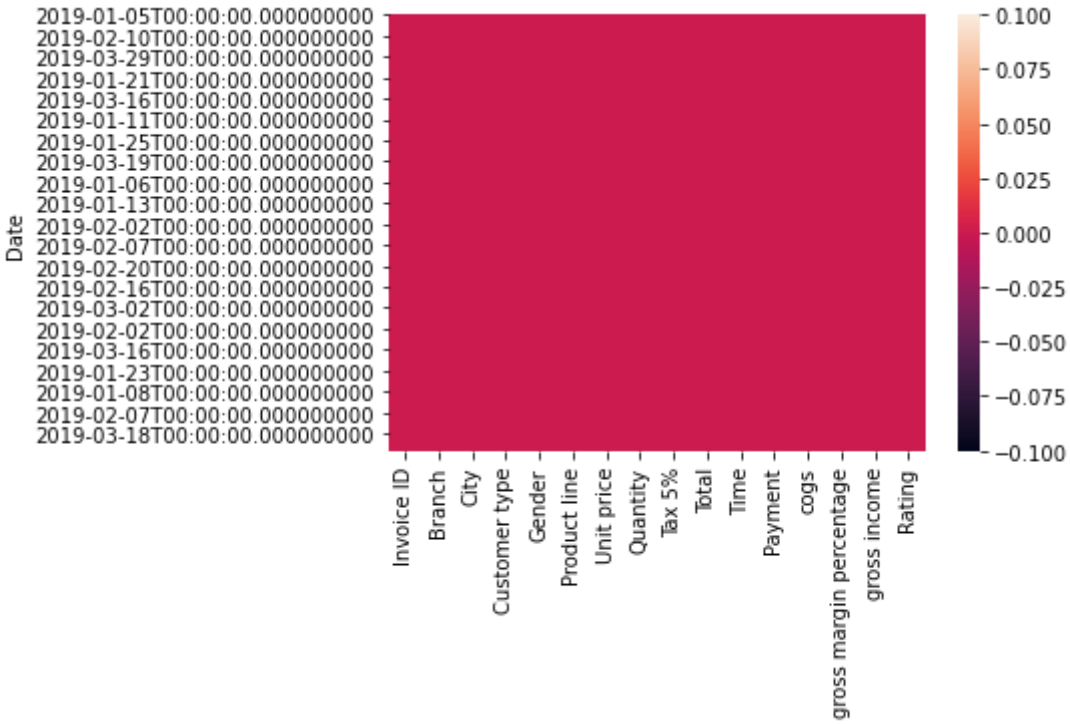
```
Invoice ID                 0
Branch                     0
City                       0
Customer type              0
Gender                     0
Product line               0
Unit price                 0
Quantity                   0
Tax 5%                     0
Total                      0
Time                       0
Payment                    0
cogs                       0
gross margin percentage    0
gross income               0
Rating                     0
dtype: int64
```

In [66]:

```
sns.heatmap(df.isnull())
```

Out[66]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1692f388>
```



In [67]:

```
df.fillna(df.mean(), inplace=True)
```

In [72]:

```
df.fillna(df.mode().iloc[0], inplace=True)
```

In [77]:

```
round(np.corrcoef(df['gross income'], df['Rating'])[1][0], 2)
```

Out[77]:

```
-0.04
```

In [78]:

```
np.round(df.corr(), 2)
```

Out[78]:

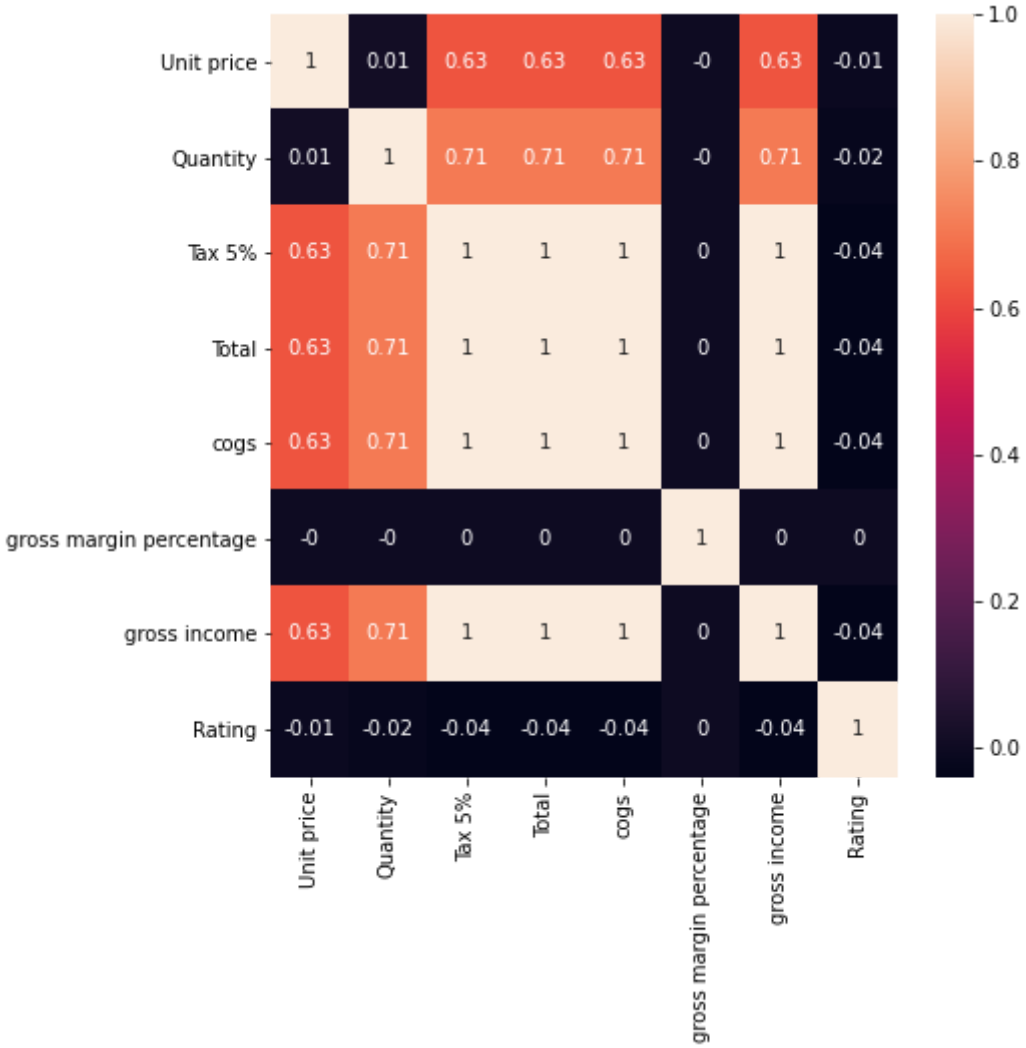
	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
Unit price	1.00	0.01	0.63	0.63	0.63	-0.0	0.63	-0.01
Quantity	0.01	1.00	0.71	0.71	0.71	-0.0	0.71	-0.02
Tax 5%	0.63	0.71	1.00	1.00	1.00	0.0	1.00	-0.04
Total	0.63	0.71	1.00	1.00	1.00	0.0	1.00	-0.04
cogs	0.63	0.71	1.00	1.00	1.00	0.0	1.00	-0.04
gross margin percentage	-0.00	-0.00	0.00	0.00	0.00	1.0	0.00	0.00
gross income	0.63	0.71	1.00	1.00	1.00	0.0	1.00	-0.04
Rating	-0.01	-0.02	-0.04	-0.04	-0.04	0.0	-0.04	1.00

In [82]:

```
plt.figure(figsize=(7,7))
sns.heatmap(np.round(df.corr(), 2), annot=True)
```

Out[82]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x14e7f538>
```



In [ ]: