Libraries

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

Link to data source: https://www.kaggle.com/aungpyaeap/supermarket-sales

Context

Supermarkets are experiencing significant growth in the most populated cities, and as a result, market competition is also increasing. The dataset in question represents historical sales data of a supermarket company, recorded across three different branches over the course of three months.

Data Dictionary

1. ***Invoice id:*** Computer generated sales slip invoice identification number
2. ***Branch:*** Branch of supercenter (3 branches are available identified by A, B and C).
3. ***City:*** Location of supercenters
4. ***Customer type:*** Type of customers, recorded by Members for customers using member card and Normal for without member card.
5. ***Gender:*** Gender type of customer
6. ***Product line:*** General item categorization groups - Electronic accessories, Fashion accessories, Food and beverages, Health and beauty, Home and lifestyle, Sports and travel
7. ***Unit price:*** Price of each product in USD
8. ***Quantity:*** Number of products purchased by customer
9. ***Tax:*** 5% tax fee for customer buying
10. ***Total:*** Total price including tax
11. ***Date:*** Date of purchase (Record available from January 2019 to March 2019)
12. ***Time:*** Purchase time (10am to 9pm)
13. ***Payment:*** Payment used by customer for purchase (3 methods are available – Cash, Credit card and Ewallet)

14. ***COGS:*** Cost of goods sold
15. ***Gross margin percentage:*** Gross margin percentage
16. ***Gross income:*** Gross income
17. ***Rating:*** Customer stratification rating on their overall shopping experience (On a scale of 1 to 10)

Task 1: Initial Data Exploration

```
df = pd.read_csv('supermarket_sales.csv')

df.head()
```

```
    Invoice ID Branch       City Customer type  Gender  \
0  750-67-8428      A     Yangon        Member  Female
1  226-31-3081      C  Naypyitaw        Normal  Female
2  631-41-3108      A     Yangon        Normal    Male
3  123-19-1176      A     Yangon        Member    Male
4  373-73-7910      A     Yangon        Normal    Male

             Product line  Unit price  Quantity   Tax 5%     Total
Date  \
0       Health and beauty       74.69       7.0  26.1415  548.9715
1/5/19
1  Electronic accessories       15.28       5.0   3.8200   80.2200
3/8/19
2      Home and lifestyle       46.33       7.0  16.2155  340.5255
3/3/19
3       Health and beauty       58.22       8.0  23.2880  489.0480
1/27/19
4       Sports and travel       86.31       7.0  30.2085  634.3785
2/8/19

    Time      Payment    cogs  gross margin percentage  gross income
Rating
0  13:08      Ewallet  522.83                 4.761905       26.1415
9.1
1  10:29         Cash   76.40                 4.761905        3.8200
9.6
2  13:23  Credit card  324.31                 4.761905       16.2155
7.4
3  20:33      Ewallet  465.76                 4.761905       23.2880
8.4
4  10:37      Ewallet  604.17                 4.761905       30.2085
5.3
```

```
df.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')
```

```
df.dtypes
```

```
Invoice ID                         object
Branch                             object
City                               object
Customer type                      object
Gender                             object
Product line                       object
Unit price                        float64
Quantity                          float64
Tax 5%                            float64
Total                             float64
Date                       datetime64[ns]
Time                               object
Payment                            object
cogs                              float64
gross margin percentage           float64
gross income                      float64
Rating                            float64
dtype: object
```

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

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

```
df.describe()
```

```
       Unit price    Quantity       Tax 5%        Total
cogs  \
count  996.000000  983.000000  1003.000000  1003.000000  1003.000000

mean    55.764568    5.501526    15.400368   323.407726   308.007358

std     26.510165    2.924673    11.715192   246.019028   234.303836

min     10.080000    1.000000     0.508500    10.678500    10.170000

25%     33.125000    3.000000     5.894750   123.789750   117.895000

50%     55.420000    5.000000    12.096000   254.016000   241.920000

75%     78.085000    8.000000    22.539500   473.329500   450.790000
```

```
max     99.960000   10.000000    49.650000  1042.650000   993.000000
```

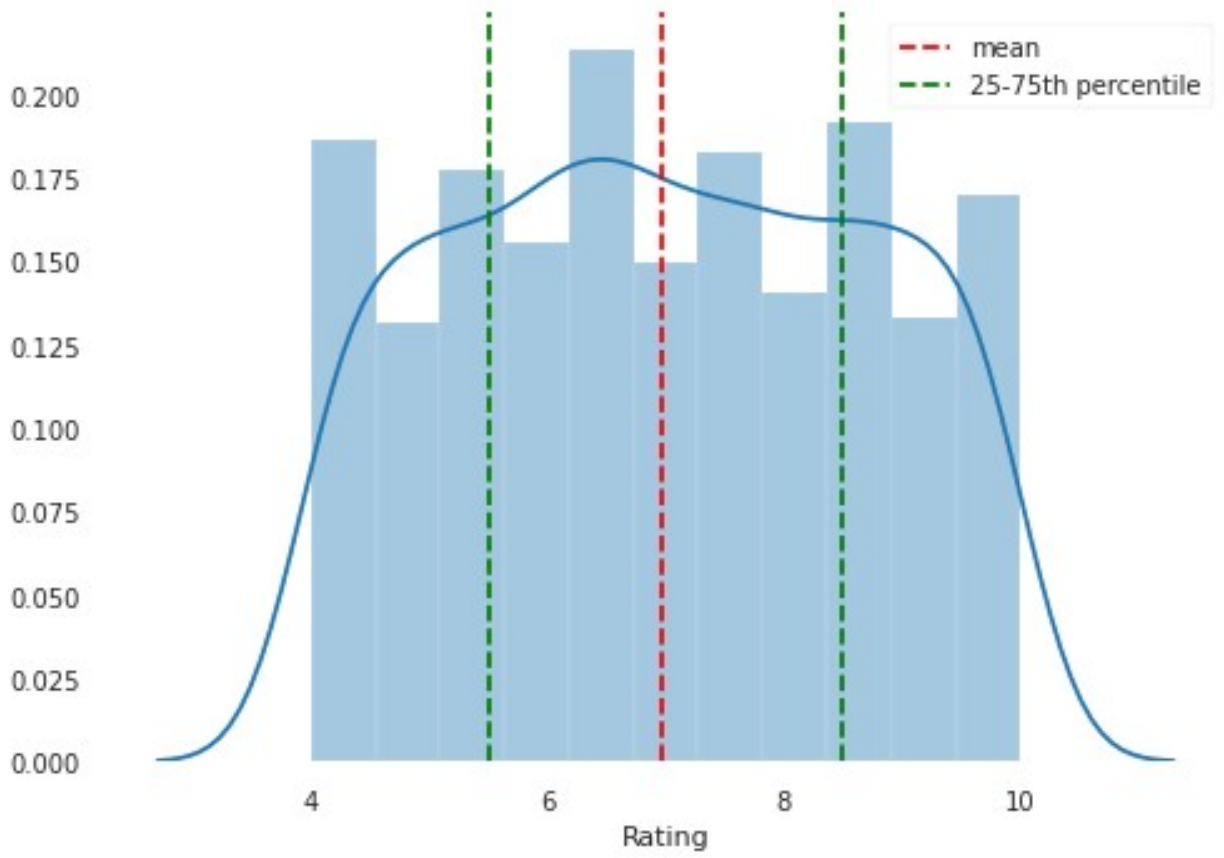
```
       gross margin percentage  gross income       Rating
count              1003.000000   1003.000000  1003.000000
mean                  4.761905     15.400368     6.972682
std                   0.000000     11.715192     1.717647
min                   4.761905      0.508500     4.000000
25%                   4.761905      5.894750     5.500000
50%                   4.761905     12.096000     7.000000
75%                   4.761905     22.539500     8.500000
max                   4.761905     49.650000    10.000000
```

Task 2: Univariate Analysis

Question 1: What does the distribution of customer ratings looks like? Is it skewed?

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

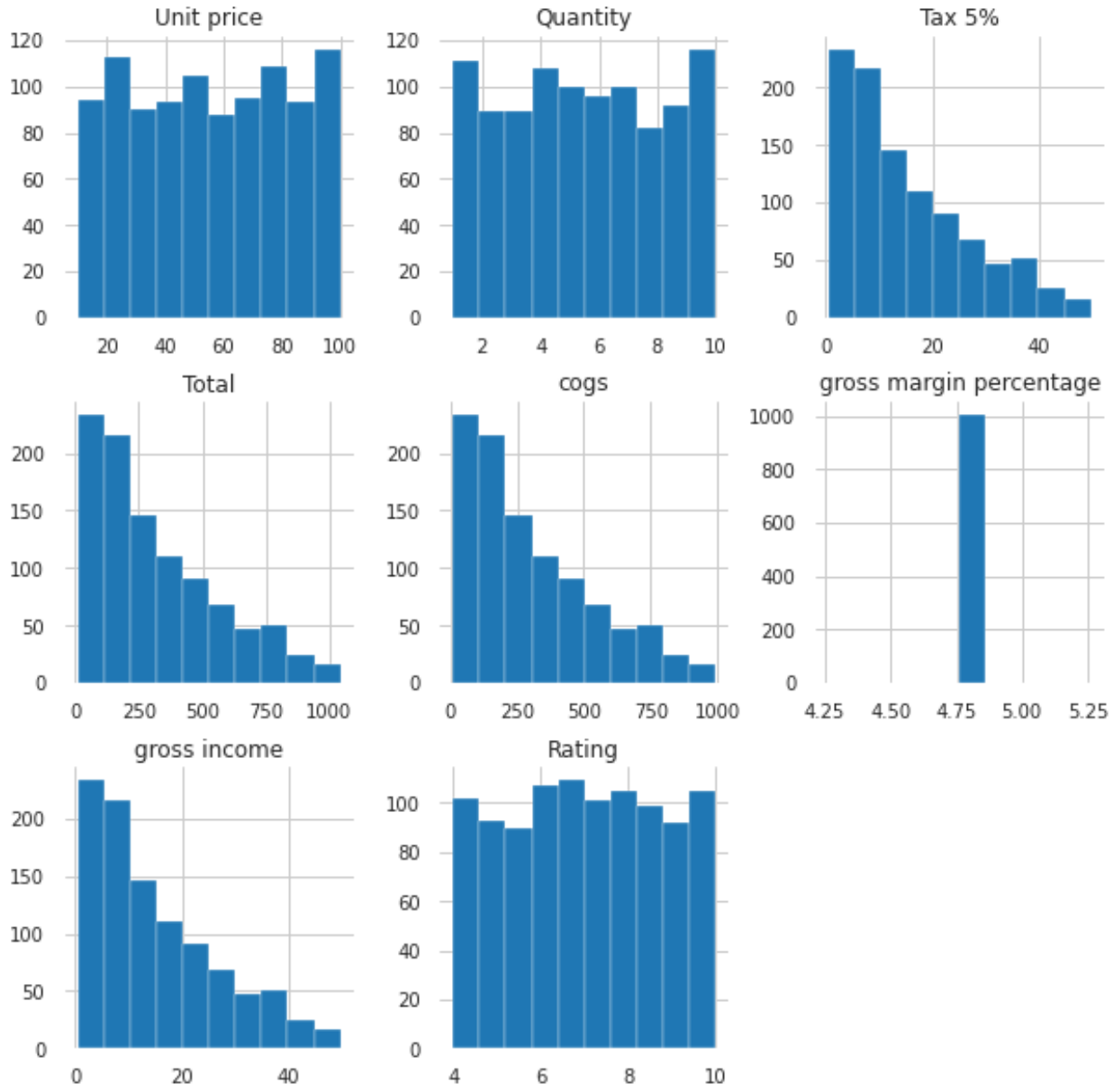
```
<matplotlib.legend.Legend at 0x7fa27b8d2240>
```



Answer: The distribution of user rating looks uniform, with no noticeable skew in either direction.

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

```
array([[<AxesSubplot:title={'center':'Unit price'}>,
        <AxesSubplot:title={'center':'Quantity'}>,
        <AxesSubplot:title={'center':'Tax 5%'}>],
       [<AxesSubplot:title={'center':'Total'}>,
        <AxesSubplot:title={'center':'cogs'}>,
        <AxesSubplot:title={'center':'gross margin percentage'}>],
       [<AxesSubplot:title={'center':'gross income'}>,
        <AxesSubplot:title={'center':'Rating'}>, <AxesSubplot:>]],
      dtype=object)
```

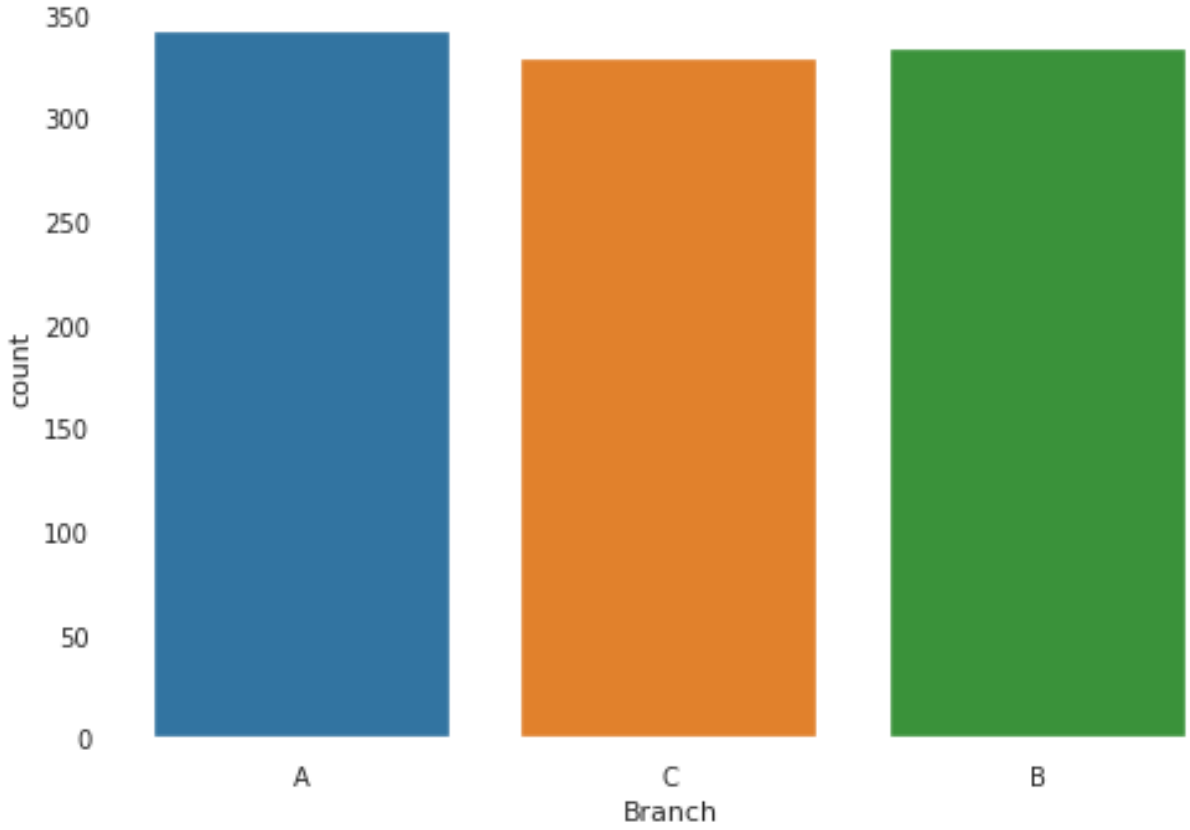


After plotting the other numerical variables we can see that the Unit price and Quantity are uniformly distributed. The tax is skewed which means that most of the tax falls between 0 and 20 but there are cases where it's over 40. Gross margin percentage is a constant value which means there is no distribution. Cost of goods sold, Total and Gross income, are highly correlated variables so they have almost identical distribution. The Rating also has uniform distribution.

Question 2: Do aggregate sales numbers differ by much between branches?

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

```
<AxesSubplot:xlabel='Branch', ylabel='count'>
```




```
df['Branch'].value_counts()
```

```
A    342
B    333
C    328
Name: Branch, dtype: int64
```

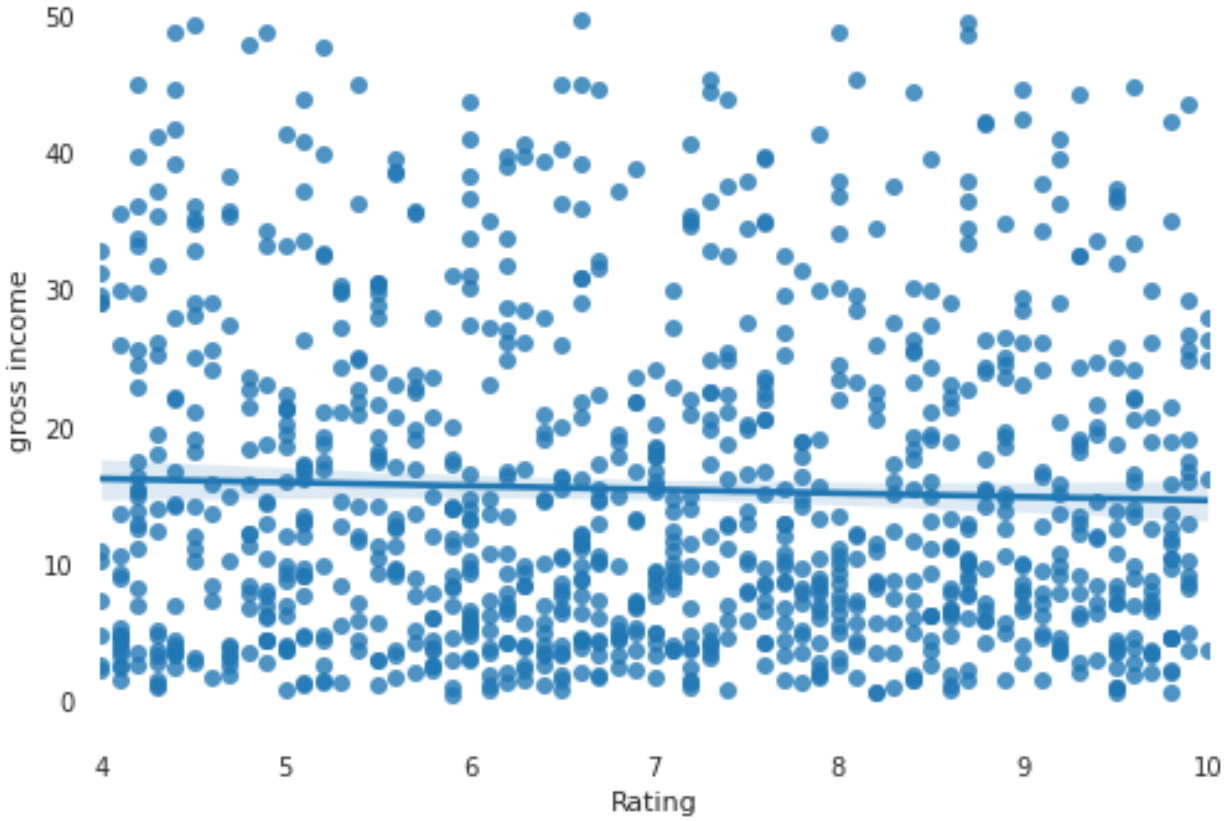
Answer: The aggregate sales numbers do not differ by much between branches.

Task 3: Bivariate Analysis

Question 3a): Is there a relationship between gross income and customer ratings?

```
sns.regplot(df['Rating'], df['gross income'])
```

```
<AxesSubplot:xlabel='Rating', ylabel='gross income'>
```

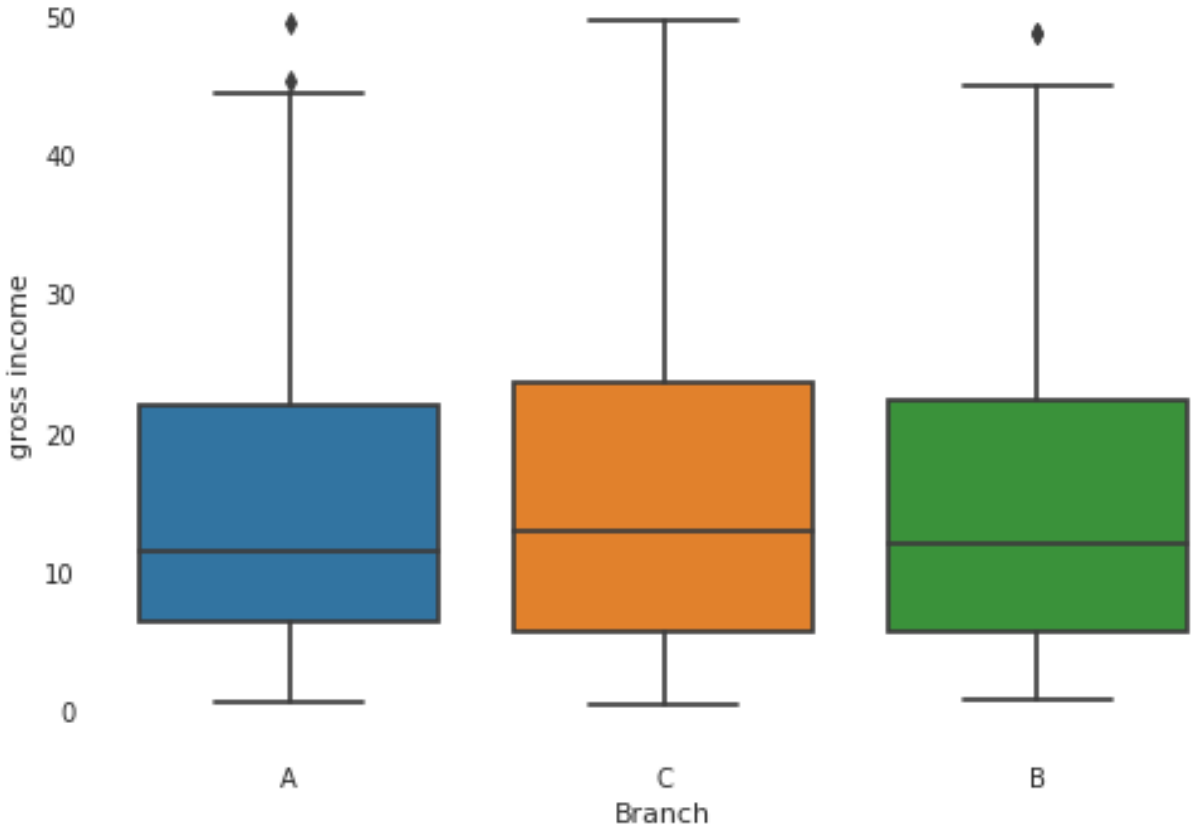


Answer: The trend line seems to be pretty flat, which means there doesn't seem to be any relationship.

Question 3b: Is there a relationship between Branch and gross income?

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

```
<AxesSubplot:xlabel='Branch', ylabel='gross income'>
```

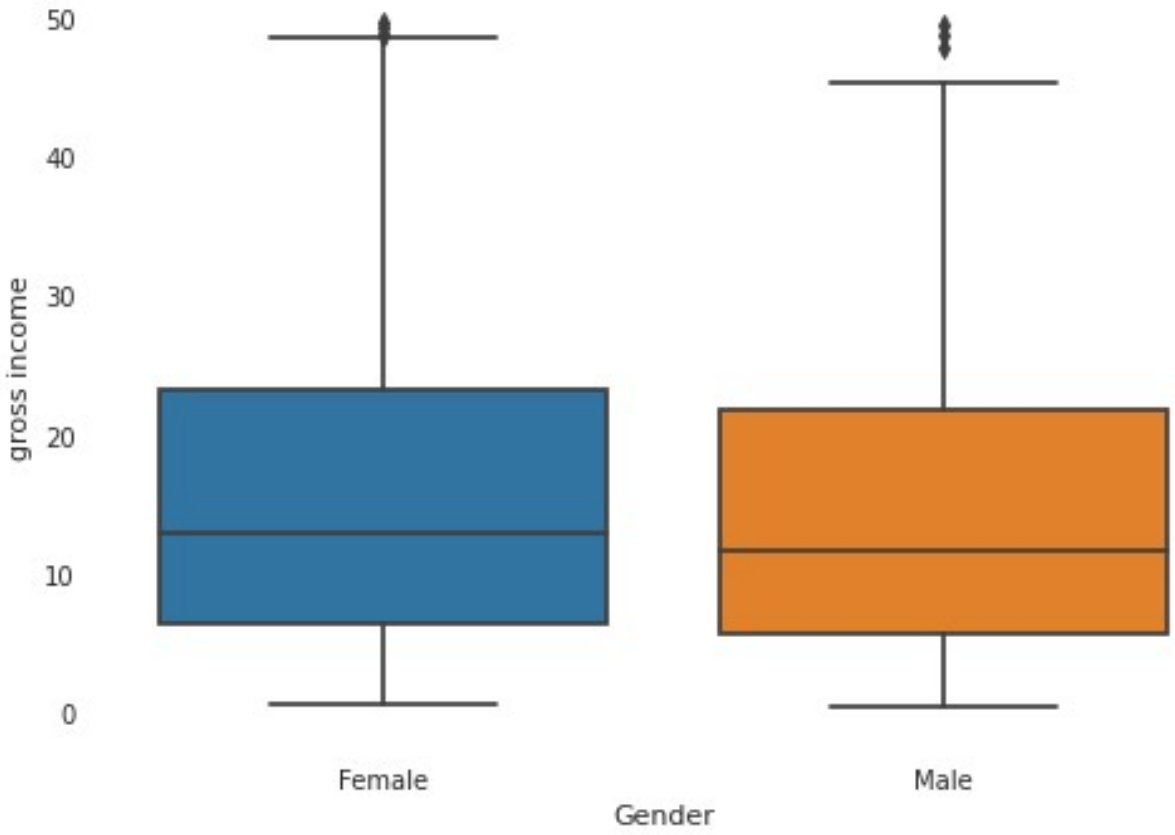


Answer: There doesn't seem to be much variation in gross income between the different branches.

Question 3c): Is there a relationship in Gender and gross income?

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

```
<AxesSubplot:xlabel='Gender', ylabel='gross income'>
```



Answer It looks like the men and women in this data set spend about the same. At 75th percentile, women spend higher than men, but on average they seem to be similar.

Question 4: Is there a noticeable time trend in gross income?

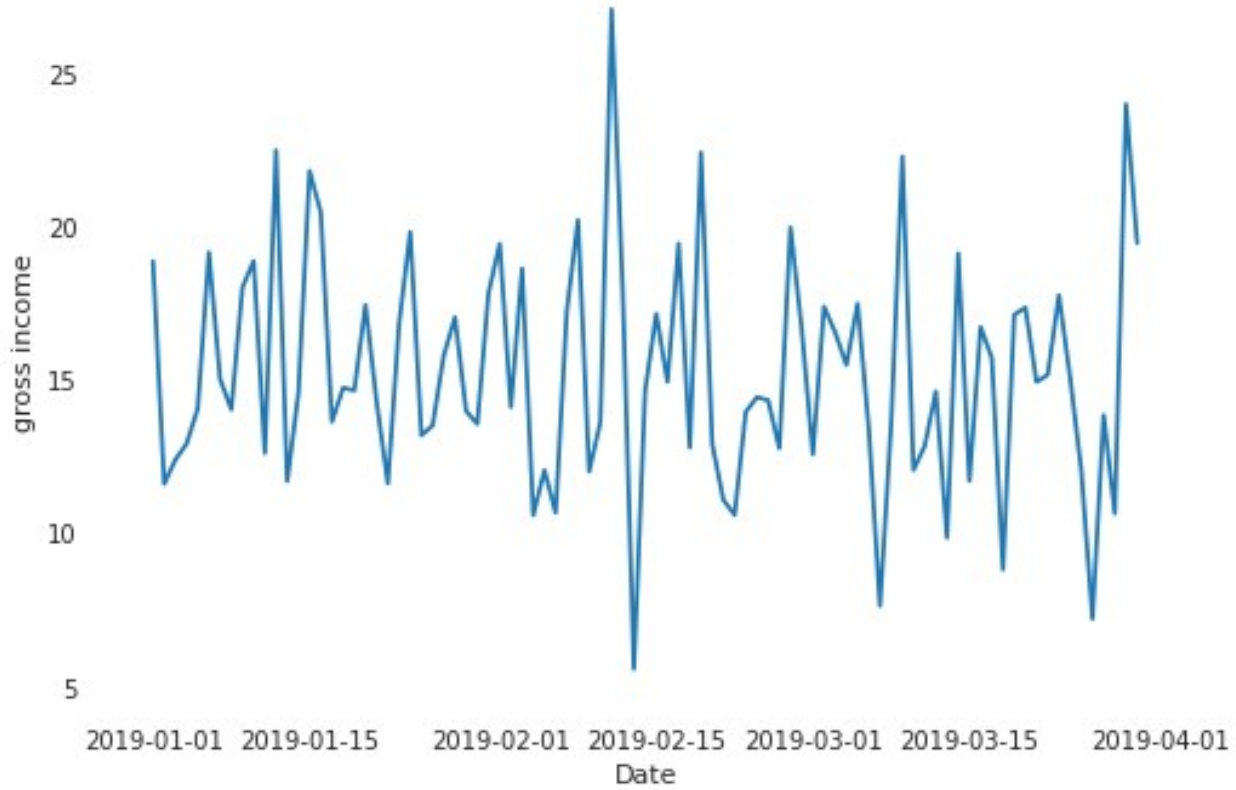
```
df.groupby(df.index).mean().index
```

```
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)
```

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

## x variable is the index which is the dates and these are the
individual unique days and the y variable is the gross income or the
mean gross income associated with those days
```

```
<AxesSubplot:xlabel='Date', ylabel='gross income'>
```



Answer: We don't notice any time trend in this graph. It looks like it varies around the same mean, there are some days where there are high numbers in gross income and some days with very low gross income. There doesn't seem to be any particular trend, and this may be because we are looking at only three months worth of data.

Task 4: Dealing With Duplicate Rows and Missing Values

```
df.duplicated().sum()
```

```
3
```

```
df[df.duplicated()==True]
##looking at the rows that are duplicated
```

```
             Invoice ID Branch      City Customer type  Gender  \
Date
2019-02-18  849-09-3807      A    Yangon        Member  Female
2019-03-10  745-74-0715      A    Yangon        Normal    Male
2019-01-26  452-04-8808      B  Mandalay        Normal    Male

                      Product line  Unit price  Quantity  Tax 5%
Total  \
Date

2019-02-18     Fashion accessories       88.34       7.0  30.919
649.299
2019-03-10  Electronic accessories         NaN       2.0   5.803
121.863
2019-01-26  Electronic accessories       87.08       NaN  30.478
640.038

             Time  Payment    cogs  gross margin percentage  gross
income  \
Date

2019-02-18  13:28     Cash  618.38                 4.761905
30.919
2019-03-10  20:46  Ewallet  116.06                 4.761905
5.803
2019-01-26  15:17     Cash  609.56                 4.761905
30.478

            Rating
Date
2019-02-18     6.6
2019-03-10     8.8
2019-01-26     5.5
```

```
df.drop_duplicates(inplace=True)
```

```
df.duplicated().sum()
##verifying that there are no duplicate rows left
```

```
0
```

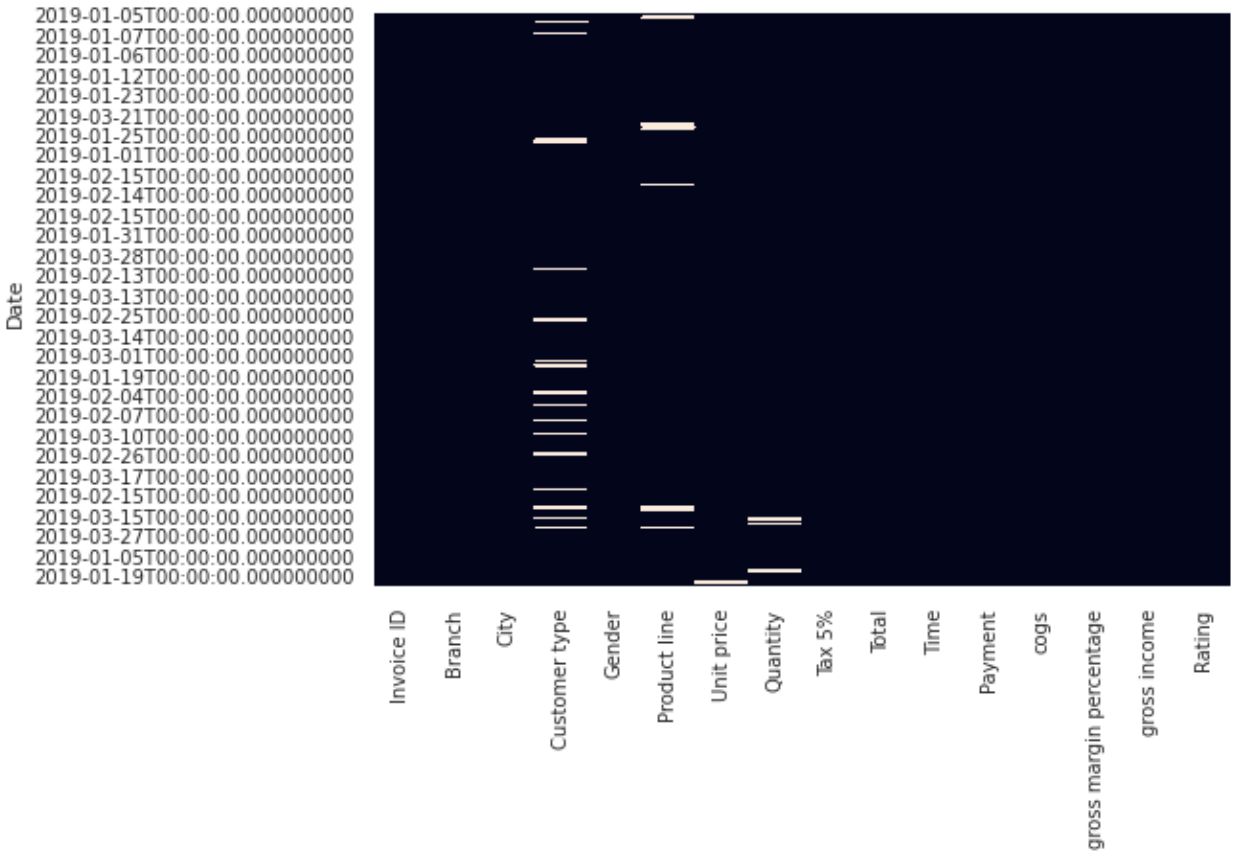
```
df.isna().sum()
##checking for missing values
```

```
Invoice ID                 0
Branch                     0
```

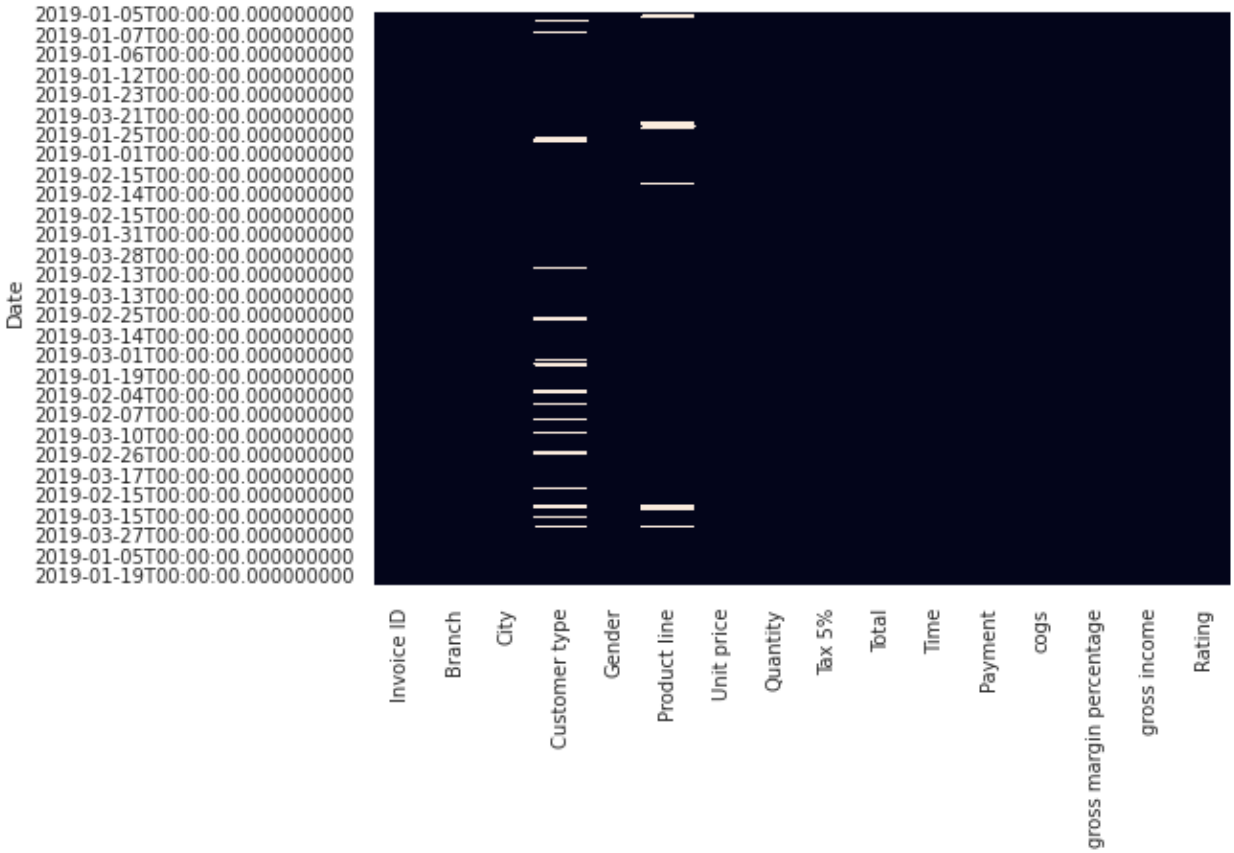
```
City                            0
Customer type                  79
Gender                          0
Product line                   43
Unit price                      6
Quantity                       19
Tax 5%                          0
Total                           0
Time                            0
Payment                         0
cogs                            0
gross margin percentage         0
gross income                    0
Rating                          0
dtype: int64
```

```
sns.heatmap(df.isnull(),cbar = False)
##visualizing the missing values
```

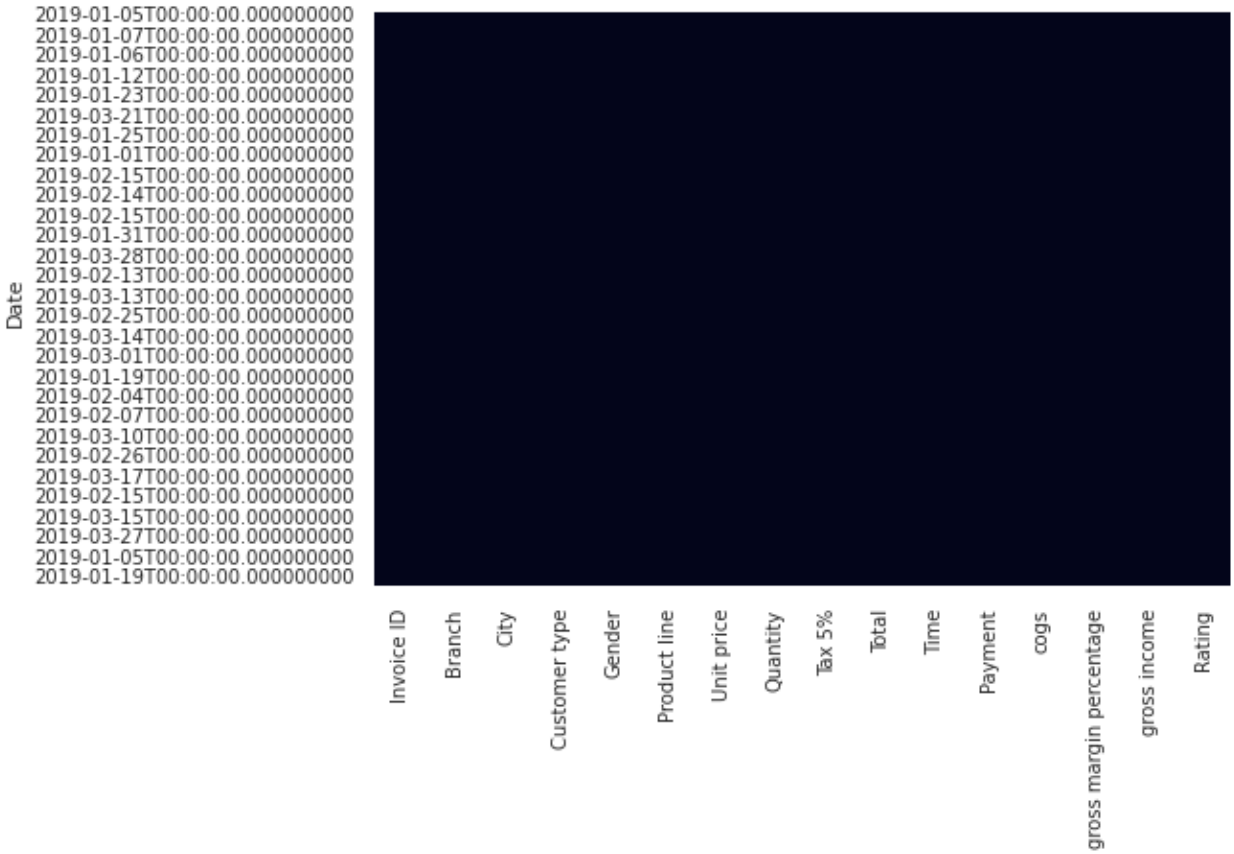
```
<AxesSubplot:ylabel='Date'>
```



```
df.fillna(df.mean(),inplace = True)
##filling the missing data with the mean which only fills the numeric
columns

sns.heatmap(df.isnull(),cbar=False)
##checking if the numerical columns are filled

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




```
df.fillna(df.mode().iloc[0], inplace = True)
##For each categorical column we are replacing the missing value with
the mode

sns.heatmap(df.isnull(), cbar = False)
##There are no missing values now

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



Task 5: Correlation Analysis

```
np.round(df.corr(),2)
##calculating every correlation between numeric columns
```

	Unit price	Quantity	Tax 5%	Total	cogs	\
Unit price	1.00	0.01	0.63	0.63	0.63	
Quantity	0.01	1.00	0.70	0.70	0.70	
Tax 5%	0.63	0.70	1.00	1.00	1.00	
Total	0.63	0.70	1.00	1.00	1.00	
cogs	0.63	0.70	1.00	1.00	1.00	
gross margin percentage	-0.00	-0.00	0.00	0.00	0.00	
gross income	0.63	0.70	1.00	1.00	1.00	
Rating	-0.01	-0.02	-0.04	-0.04	-0.04	

	gross margin percentage	gross income	Rating
Unit price	-0.0	0.63	-0.01
Quantity	-0.0	0.70	-0.02
Tax 5%	0.0	1.00	-0.04
Total	0.0	1.00	-0.04

cogs	0.0	1.00	-0.04
gross margin percentage	1.0	0.00	0.00
gross income	0.0	1.00	-0.04
Rating	0.0	-0.04	1.00

```
sns.heatmap(np.round(df.corr(),2), annot = True)
##visualizing the correlation
##Example:we can see that Rating has a low correlation with every
other variable, it does not seem like the amount which customer spends
on items is correlated to their rating
```

```
<AxesSubplot:>
```