.....exploratory data analysis.....

TASK - 4:

first importing it...

In [12]:
```
# importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings as wr
wr.filterwarnings('ignore')
df = pd.read_csv("C:\\Users\\ACER\\Desktop\\financial.csv")
print(df.head())
```

```
      segment  country      product  Discount Band   Units Sold  \
0  Government   Canada   Carretera            None       1618.5
1  Government  Germany   Carretera            None       1321.0
2   Midmarket   France   Carretera            None       2178.0
3   Midmarket  Germany   Carretera            None        888.0
4   Midmarket   Mexico   Carretera            None       2470.0

   Manufacturing Price   Sale Price   Gross Sales   Discounts         Sales
\
0                $3.00       $20.00    $32,370.00        $-      $32,370.00
1                $3.00       $20.00    $26,420.00        $-      $26,420.00
2                $3.00       $15.00    $32,670.00        $-      $32,670.00
3                $3.00       $15.00    $13,320.00        $-      $13,320.00
4                $3.00       $15.00    $37,050.00        $-      $37,050.00

          COGS        Profit         Date  Month Number  Month Name   Year
0   $16,185.00    $16,185.00   01-01-2014             1     January   2014
1   $13,210.00    $13,210.00   01-01-2014             1     January   2014
2   $21,780.00    $10,890.00   01-06-2014             6        June   2014
3    $8,880.00     $4,440.00   01-06-2014             6        June   2014
4   $24,700.00    $12,350.00   01-06-2014             6        June   2014
```

In [13]:
```
# shape of the data
df.shape
```

Out[13]: (700, 16)

In [47]:
```
df.tail()
```

Out[47]:

	segment	country	product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Dis
695	Small Business	France	Amarilla	High	2475.0	$260.00	$300.00	$7,42,500.00	$1,11,
696	Small Business	Mexico	Amarilla	High	546.0	$260.00	$300.00	$1,63,800.00	$24,
697	Government	Mexico	Montana	High	1368.0	$5.00	$7.00	$9,576.00	$1,
698	Government	Canada	Paseo	High	723.0	$10.00	$7.00	$5,061.00	$
699	Channel Partners	United States of America	VTT	High	1806.0	$250.00	$12.00	$21,672.00	$3,

In [14]:
```
#data information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   segment                700 non-null    object 
 1   country                700 non-null    object 
 2   product                700 non-null    object 
 3    Discount Band         700 non-null    object 
 4   Units Sold             700 non-null    float64
 5    Manufacturing Price   700 non-null    object 
 6    Sale Price            700 non-null    object 
 7    Gross Sales           700 non-null    object 
 8    Discounts             700 non-null    object 
 9     Sales                700 non-null    object 
 10   COGS                  700 non-null    object 
 11   Profit                700 non-null    object 
 12  Date                   700 non-null    object 
 13  Month Number           700 non-null    int64  
 14   Month Name            700 non-null    object 
 15  Year                   700 non-null    int64  
dtypes: float64(1), int64(2), object(13)
memory usage: 52.0+ KB
```

In [15]:
```
# describing the data
df.describe()
```

Out[15]:

	Units Sold	Month Number	Year
count	700.000000	700.000000	700.000000
mean	1608.294286	7.900000	2013.750000
std	867.427859	3.377321	0.433322
min	200.000000	1.000000	2013.000000
25%	905.000000	5.750000	2013.750000
50%	1542.500000	9.000000	2014.000000
75%	2229.125000	10.250000	2014.000000
max	4492.500000	12.000000	2014.000000

In [16]:
```
#column to list
df.columns.tolist()
```

Out[16]:
```
['segment',
 'country',
 'product',
 ' Discount Band ',
 'Units Sold',
 ' Manufacturing Price ',
 ' Sale Price ',
 ' Gross Sales ',
 ' Discounts ',
 '  Sales ',
 ' COGS ',
 ' Profit ',
 'Date',
 'Month Number',
 ' Month Name ',
 'Year']
```

In [17]:
```
# check for missing values:
df.isnull().sum()
```

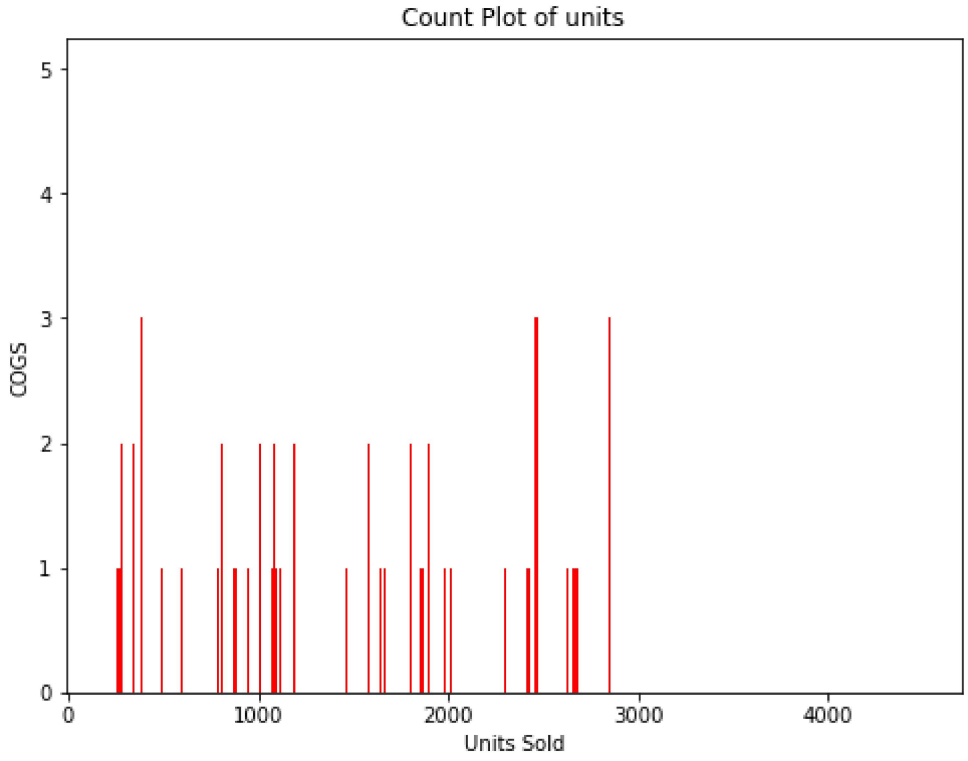
Out[17]:
```
segment                 0
country                 0
product                 0
 Discount Band          0
Units Sold              0
 Manufacturing Price    0
 Sale Price             0
 Gross Sales            0
 Discounts              0
  Sales                 0
 COGS                   0
 Profit                 0
Date                    0
Month Number            0
 Month Name             0
Year                    0
dtype: int64
```

In [18]:
```
#checking duplicate values
df.nunique()
```

Out[18]:
```
segment                  5
country                  5
product                  6
 Discount Band           4
Units Sold              510
 Manufacturing Price      6
 Sale Price               7
 Gross Sales            550
 Discounts              515
  Sales                 559
 COGS                   545
 Profit                 557
Date                     16
Month Number             12
 Month Name              12
Year                      2
dtype: int64
```

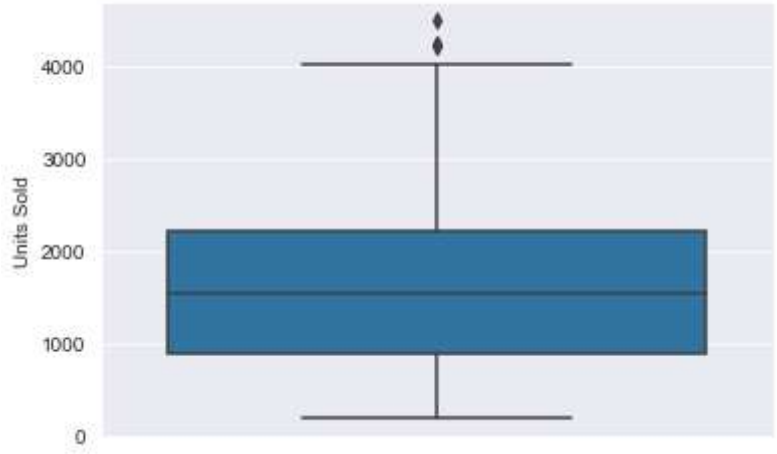
univariate analysis

In [22]:
```
# Assuming 'df' is your DataFrame
quality_counts = df['Units Sold'].value_counts()

# Using Matplotlib to create a count plot
plt.figure(figsize=(8, 6))
plt.bar(quality_counts.index, quality_counts, color='red')
plt.title('Count Plot of units')
plt.xlabel('Units Sold')
plt.ylabel('COGS')
plt.show()
```

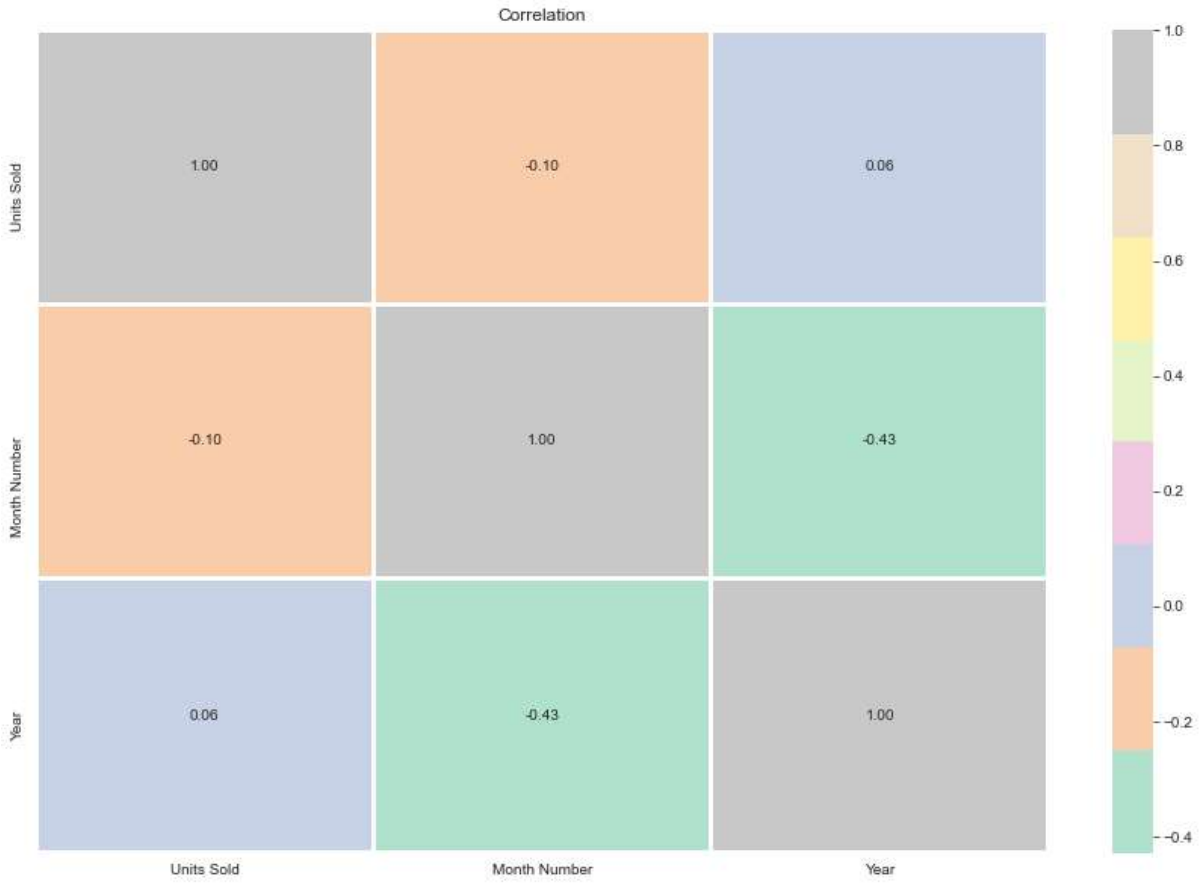


In [35]:
```
#plotting box plot
sns.boxplot( y='Units Sold', data=df)
```

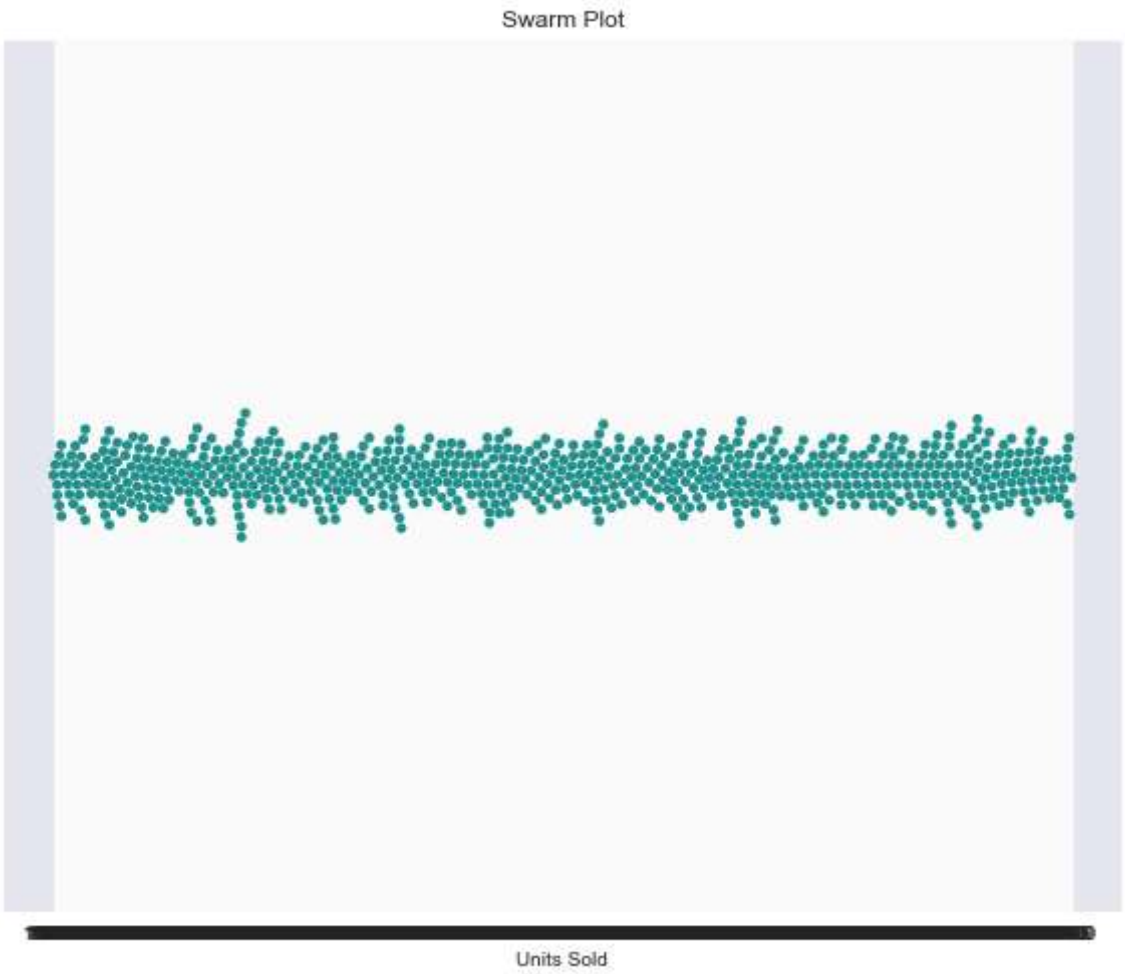
Out[35]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x9851868>
```



In [36]:
```
# Assuming 'df' is your DataFrame
plt.figure(figsize=(15, 10))

# Using Seaborn to create a heatmap
sns.heatmap(df.corr(), annot=True, fmt='.2f', cmap='Pastel2', linewidths=2)

plt.title('Correlation')
plt.show()
```

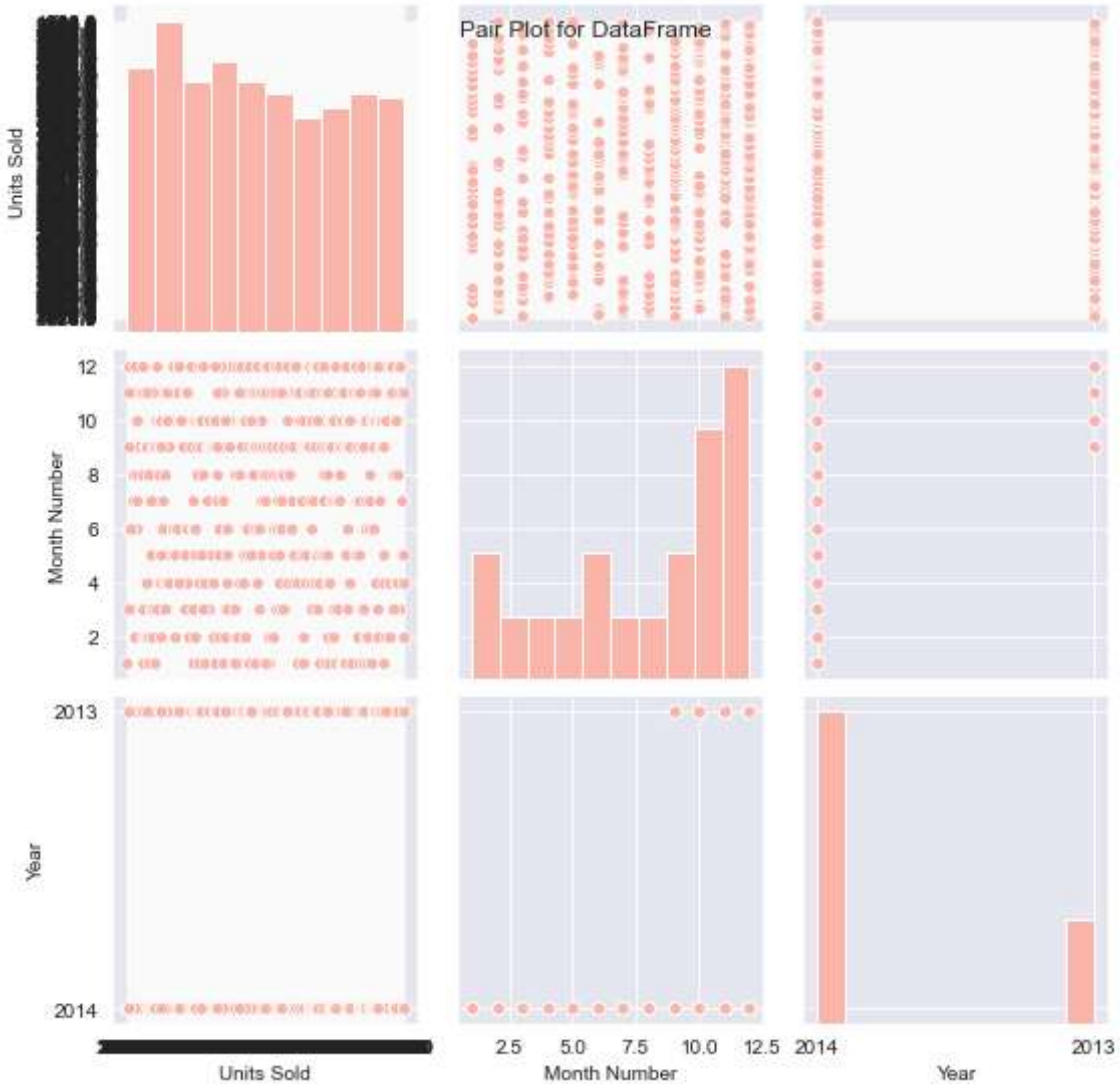


In [42]:
```
# Assuming 'df' is your DataFrame
plt.figure(figsize=(10, 8))

# Using Seaborn to create a swarm plot
sns.swarmplot(x="Units Sold", data=df, palette='viridis')

plt.title('Swarm Plot')
plt.xlabel('Units Sold')

plt.show()
```

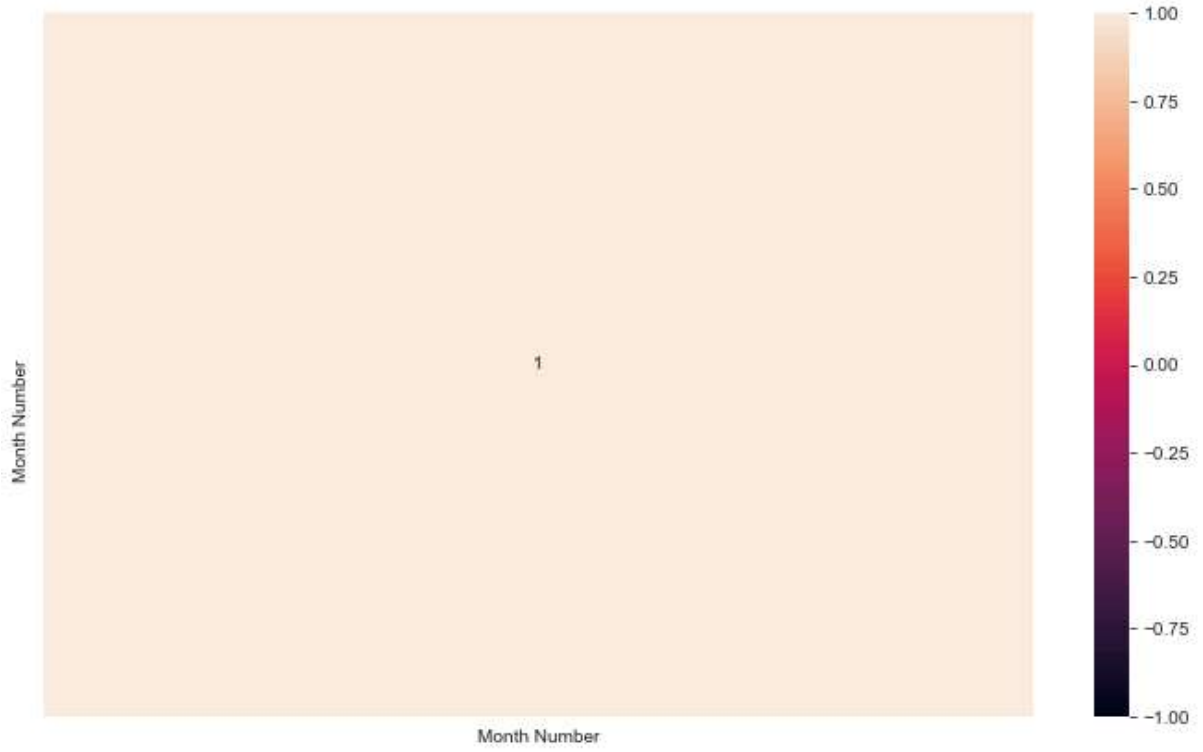


In [43]:
```
# Set the color palette
sns.set_palette("Pastel1")

# Assuming 'df' is your DataFrame
plt.figure(figsize=(10, 6))

# Using Seaborn to create a pair plot with the specified color palette
sns.pairplot(df)

plt.suptitle('Pair Plot for DataFrame')
plt.show()
```

```
<Figure size 720x432 with 0 Axes>
```



In [53]:
```
plt.figure(figsize=(12, 7))
sns.heatmap(df.drop(['Units Sold','Year'],axis=1).corr(), annot = True, vmin =
plt.show()
```



DONE BY:-

K.K Sreevalli

Data analysis for python