In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
df=pd.read_excel("Sales_Data.xlsx")
```

In [3]:

```
df.head()
```

Out[3]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	DAYS_SINCE_LASTORDE
0	**10107**	**30**	**95.70**	**2**	**2871.00**	**2018-02-24**	**82**
1	**10121**	**34**	**81.35**	**5**	**2765.90**	**2018-05-07**	**75**
2	**10134**	**41**	**94.74**	**2**	**3884.34**	**2018-07-01**	**70**
3	**10145**	**45**	**83.26**	**6**	**3746.70**	**2018-08-25**	**64**
4	**10168**	**36**	**96.66**	**1**	**3479.76**	**2018-10-28**	**58**

In [4]:

```
df.columns
```

Out[4]:

```
Index(['ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER',
       'SALES', 'ORDERDATE', 'DAYS_SINCE_LASTORDER', 'STATUS', 'PRODUCTLINE',
       'MSRP', 'PRODUCTCODE', 'CUSTOMERNAME', 'PHONE', 'ADDRESSLINE1', 'CITY',
       'POSTALCODE', 'COUNTRY', 'CONTACTLASTNAME', 'CONTACTFIRSTNAME',
       'DEALSIZE'],
      dtype='object')
```

In [5]:

```
df.shape
```

Out[5]:

```
(2747, 20)
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2747 entries, 0 to 2746
Data columns (total 20 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ORDERNUMBER          2747 non-null   int64
 1   QUANTITYORDERED      2747 non-null   int64
 2   PRICEEACH            2747 non-null   float64
```

```
 3   ORDERLINENUMBER        2747 non-null   int64
 4   SALES                  2747 non-null   float64
 5   ORDERDATE              2747 non-null   datetime64[ns]
 6   DAYS_SINCE_LASTORDER   2747 non-null   int64
 7   STATUS                 2747 non-null   object
 8   PRODUCTLINE            2747 non-null   object
 9   MSRP                   2747 non-null   int64
 10  PRODUCTCODE            2747 non-null   object
 11  CUSTOMERNAME           2747 non-null   object
 12  PHONE                  2747 non-null   object
 13  ADDRESSLINE1           2747 non-null   object
 14  CITY                   2747 non-null   object
 15  POSTALCODE             2747 non-null   object
 16  COUNTRY                2747 non-null   object
 17  CONTACTLASTNAME        2747 non-null   object
 18  CONTACTFIRSTNAME       2747 non-null   object
 19  DEALSIZE               2747 non-null   object
dtypes: datetime64[ns](1), float64(2), int64(5), object(12)
memory usage: 429.3+ KB
```

In [7]:

```
df.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
ORDERNUMBER	2747.0	10259.761558	91.877521	10100.00	10181.000	10264.00	10334.500	10425.00
QUANTITYORDERED	2747.0	35.103021	9.762135	6.00	27.000	35.00	43.000	97.00
PRICEEACH	2747.0	101.098951	42.042548	26.88	68.745	95.55	127.100	252.87
ORDERLINENUMBER	2747.0	6.491081	4.230544	1.00	3.000	6.00	9.000	18.00
SALES	2747.0	3553.047583	1838.953901	482.13	2204.350	3184.80	4503.095	14082.80
DAYS_SINCE_LASTORDER	2747.0	1757.085912	819.280576	42.00	1077.000	1761.00	2436.500	3562.00
MSRP	2747.0	100.691664	40.114802	33.00	68.000	99.00	124.000	214.00

In [8]:

```
df.describe(include='all').T
```

Out[8]:

	count	unique	top	freq	first	last	mean	std	min	25%	50%	75%
ORDERNUMBER	2747	NaN	NaN	NaN	NaT	NaT	10259.8	91.8775	10100	10181	10264	10334.5
QUANTITYORDERED	2747	NaN	NaN	NaN	NaT	NaT	35.103	9.76214	6	27	35	43
PRICEEACH	2747	NaN	NaN	NaN	NaT	NaT	101.099	42.0425	26.88	68.745	95.55	127.1
ORDERLINENUMBER	2747	NaN	NaN	NaN	NaT	NaT	6.49108	4.23054	1	3	6	9
SALES	2747	NaN	NaN	NaN	NaT	NaT	3553.05	1838.95	482.13	2204.35	3184.8	4503.09
ORDERDATE	2747	246	2018-11-14 00:00:00	38	2018-01-06	2020-05-31	NaN	NaN	NaN	NaN	NaN	NaN
DAYS_SINCE_LASTORDER	2747	NaN	NaN	NaN	NaT	NaT	1757.09	819.281	42	1077	1761	2436.5
STATUS	2747	6	Shipped	2541	NaT	NaT	NaN	NaN	NaN	NaN	NaN	NaN
PRODUCTLINE	2747	7	Classic Cars	949	NaT	NaT	NaN	NaN	NaN	NaN	NaN	NaN
MSRP	2747	NaN	NaN	NaN	NaT	NaT	100.692	40.1148	33	68	99	124
PRODUCTCODE	2747	109	S18_3232	51	NaT	NaT	NaN	NaN	NaN	NaN	NaN	NaN
CUSTOMERNAME	2747	89	Euro Shopping Channel	259	NaT	NaT	NaN	NaN	NaN	NaN	NaN	NaN
			(91) 555 94									

PHONE	**2747**	**88**	**t6p**	**259**	**NaT**	**NaT**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
	count	**unique**		**freq**	**first**	**last**	**mean**	**std**	**min**	**25%**	**50%**	**75%**
ADDRESSLINE1	**2747**	**89**	**C/ Moralzarzal, 86**	**259**	**NaT**	**NaT**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
CITY	**2747**	**71**	**Madrid**	**304**	**NaT**	**NaT**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
POSTALCODE	**2747**	**73**	**28034**	**259**	**NaT**	**NaT**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
COUNTRY	**2747**	**19**	**USA**	**928**	**NaT**	**NaT**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
CONTACTLASTNAME	**2747**	**76**	**Freyre**	**259**	**NaT**	**NaT**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
CONTACTFIRSTNAME	**2747**	**72**	**Diego**	**259**	**NaT**	**NaT**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
DEALSIZE	**2747**	**3**	**Medium**	**1349**	**NaT**	**NaT**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**

In [9]:

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

Out[9]:

```
0
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
ORDERNUMBER             0
QUANTITYORDERED         0
PRICEEACH               0
ORDERLINENUMBER         0
SALES                   0
ORDERDATE               0
DAYS_SINCE_LASTORDER    0
STATUS                  0
PRODUCTLINE             0
MSRP                    0
PRODUCTCODE             0
CUSTOMERNAME            0
PHONE                   0
ADDRESSLINE1            0
CITY                    0
POSTALCODE              0
COUNTRY                 0
CONTACTLASTNAME         0
CONTACTFIRSTNAME        0
DEALSIZE                0
dtype: int64
```

In [11]:

```
cat=[]
num=[]
for i in df.columns:
    if df[i].dtype=="object":
        cat.append(i)
    else:
        num.append(i)
print(cat)
print(num)
```

```
['STATUS', 'PRODUCTLINE', 'PRODUCTCODE', 'CUSTOMERNAME', 'PHONE', 'ADDRESSLINE1', 'CITY',
'POSTALCODE', 'COUNTRY', 'CONTACTLASTNAME', 'CONTACTFIRSTNAME', 'DEALSIZE']
['ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER', 'SALES', 'ORDERDATE',
'DAYS_SINCE_LASTORDER', 'MSRP']
```

In [12]:

```
for column in df.columns:
```

```
    if df[column].dtype == 'object':
        print(column.upper(),': ',df[column].nunique())
        print(df[column].value_counts().sort_values())
        print('\n')
```

```
STATUS :  6
Disputed        14
In Process      41
On Hold         44
Resolved        47
Cancelled       60
Shipped       2541
Name: STATUS, dtype: int64


PRODUCTLINE :  7
Trains               77
Ships               230
Trucks and Buses    295
Planes              304
Motorcycles         313
Vintage Cars        579
Classic Cars        949
Name: PRODUCTLINE, dtype: int64


PRODUCTCODE :  109
S18_2248    21
S18_4409    21
S18_1749    21
S24_3969    21
S24_2887    22
            ..
S50_1392    28
S24_1444    28
S32_2509    28
S24_2840    28
S18_3232    51
Name: PRODUCTCODE, Length: 109, dtype: int64


CUSTOMERNAME :  89
Boards & Toys Co.                3
Atelier graphique                7
Royale Belge                     8
Auto-Moto Classics Inc.          8
Microscale Inc.                 10
                               ...
AV Stores, Co.                  51
La Rochelle Gifts               53
Australian Collectors, Co.      55
Mini Gifts Distributors Ltd.   180
Euro Shopping Channel          259
Name: CUSTOMERNAME, Length: 89, dtype: int64


PHONE :  88
3105552373           3
40.32.2555           7
6175558428           8
(071) 23 67 2555     8
2125551957          10
                   ...
6175558555          51
40.67.8555          53
03 9520 4555        55
4155551450         180
(91) 555 94 44     259
Name: PHONE, Length: 88, dtype: int64


ADDRESSLINE1 :  89
```

```
4097 Douglas Av.                  3
54, rue Royale                    7
Boulevard Tirou, 255              8
16780 Pompton St.                 8
5290 North Pendale Street        10
                                ...
Fauntleroy Circus                51
67, rue des Cinquante Otages     53
636 St Kilda Road                55
5677 Strong St.                 180
C/ Moralzarzal, 86              259
Name: ADDRESSLINE1, Length: 89, dtype: int64


CITY :  71
Charleroi           8
Burbank            13
Munich             14
Sevilla            15
South Brisbane     15
                  ...
Paris              70
Singapore          79
NYC               152
San Rafael        180
Madrid            304
Name: CITY, Length: 71, dtype: int64


POSTALCODE :  73
92561        3
B-6000       8
WA1 1DP     12
80686       14
4101        15
           ...
50553       61
94217       89
10022      152
97562      205
28034      259
Name: POSTALCODE, Length: 73, dtype: int64


COUNTRY :  19
Ireland         16
Philippines     26
Switzerland     31
Belgium         33
Japan           52
Austria         55
Sweden          57
Germany         62
Denmark         63
Canada          70
Singapore       79
Norway          85
Finland         92
Italy          113
UK             144
Australia      185
France         314
Spain          342
USA            928
Name: COUNTRY, dtype: int64


CONTACTLASTNAME :  76
Schmitt       7
Cartrain      8
Kuo          10
Tseng        11
```

```
Hardy        12
           ...
Yu           80
Frick        91
Young       115
Nelson      204
Freyre      259
Name: CONTACTLASTNAME, Length: 76, dtype: int64


CONTACTFIRSTNAME :  72
Carine        7
Pascale       8
Kee          10
Thomas       12
Jesus        13
           ...
Juri         60
Michael      70
Sue          84
Valarie     257
Diego       259
Name: CONTACTFIRSTNAME, Length: 72, dtype: int64


DEALSIZE :  3
Large      152
Small     1246
Medium    1349
Name: DEALSIZE, dtype: int64
```

In [13]:

```
'ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER', 'SALES', 'ORDERDATE',
'DAYS_SINCE_LASTORDER', 'MSRP'
```

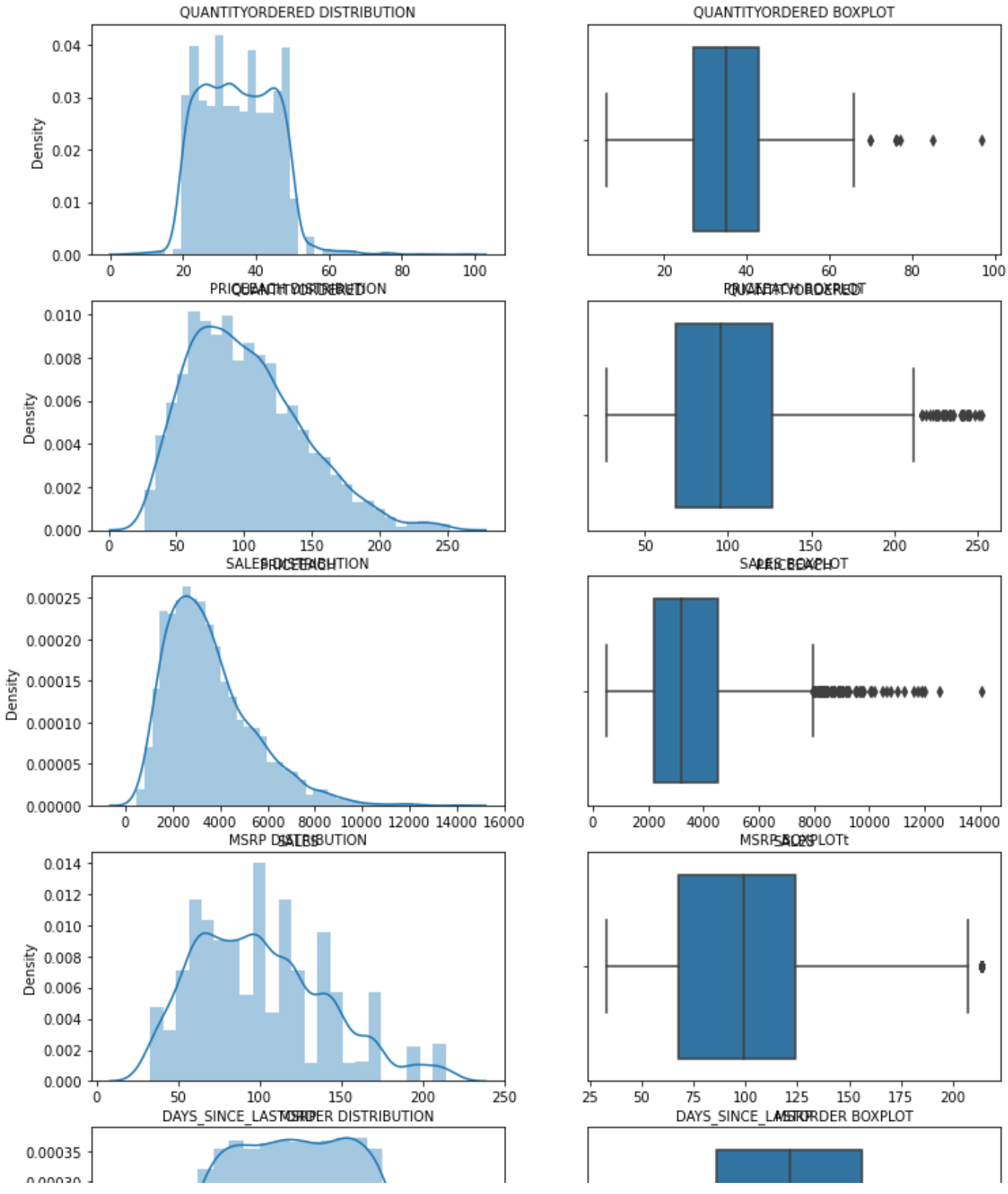
Out[13]:

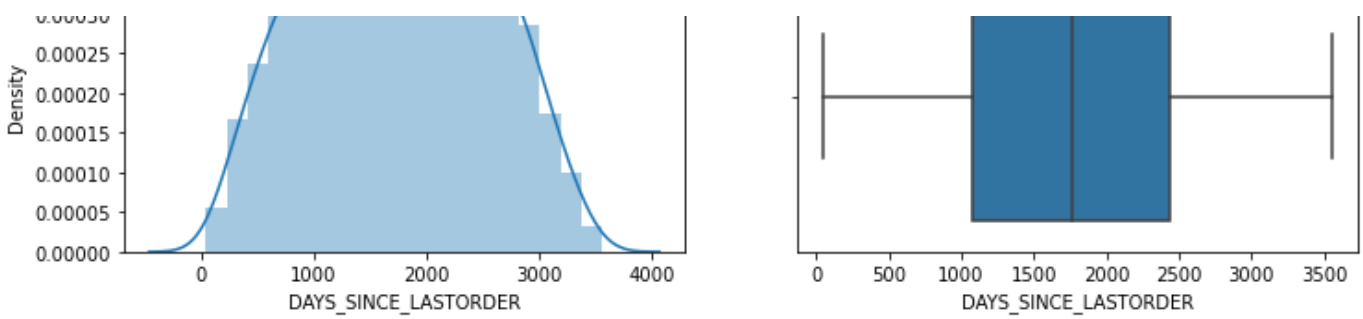
```
('ORDERNUMBER',
 'QUANTITYORDERED',
 'PRICEEACH',
 'ORDERLINENUMBER',
 'SALES',
 'ORDERDATE',
 'DAYS_SINCE_LASTORDER',
 'MSRP')
```

In [14]:

```
fig, axes = plt.subplots(nrows=5,ncols=2)
fig.set_size_inches(12,18)

a = sns.distplot(df['QUANTITYORDERED'] , ax=axes[0][0])
a.set_title("QUANTITYORDERED DISTRIBUTION",fontsize=10)

a = sns.boxplot(df['QUANTITYORDERED'] , orient = "v" , ax=axes[0][1])
a.set_title("QUANTITYORDERED BOXPLOT",fontsize=10)


a = sns.distplot(df['PRICEEACH'] , ax=axes[1][0])
a.set_title("PRICEEACH DISTRIBUTION",fontsize=10)

a = sns.boxplot(df['PRICEEACH'] , orient = "v" , ax=axes[1][1])
a.set_title("PRICEEACH BOXPLOT",fontsize=10)



a = sns.distplot(df['SALES'] , ax=axes[2][0])
a.set_title("SALES DISTRIBUTION",fontsize=10)
```

```
a = sns.boxplot(df['SALES'] , orient = "v" , ax=axes[2][1])
a.set_title("SALES BOXPLOT",fontsize=10)


a = sns.distplot(df['MSRP'] , ax=axes[3][0])
a.set_title("MSRP DISTRIBUTION",fontsize=10)

a = sns.boxplot(df['MSRP'] , orient = "v" , ax=axes[3][1])
a.set_title("MSRP BOXPLOTt",fontsize=10)


a = sns.distplot(df['DAYS_SINCE_LASTORDER'] , ax=axes[4][0])
a.set_title("DAYS_SINCE_LASTORDER DISTRIBUTION",fontsize=10)

a = sns.boxplot(df['DAYS_SINCE_LASTORDER'] , orient = "v" , ax=axes[4][1])
a.set_title("DAYS_SINCE_LASTORDER BOXPLOT",fontsize=10)


plt.show()
```

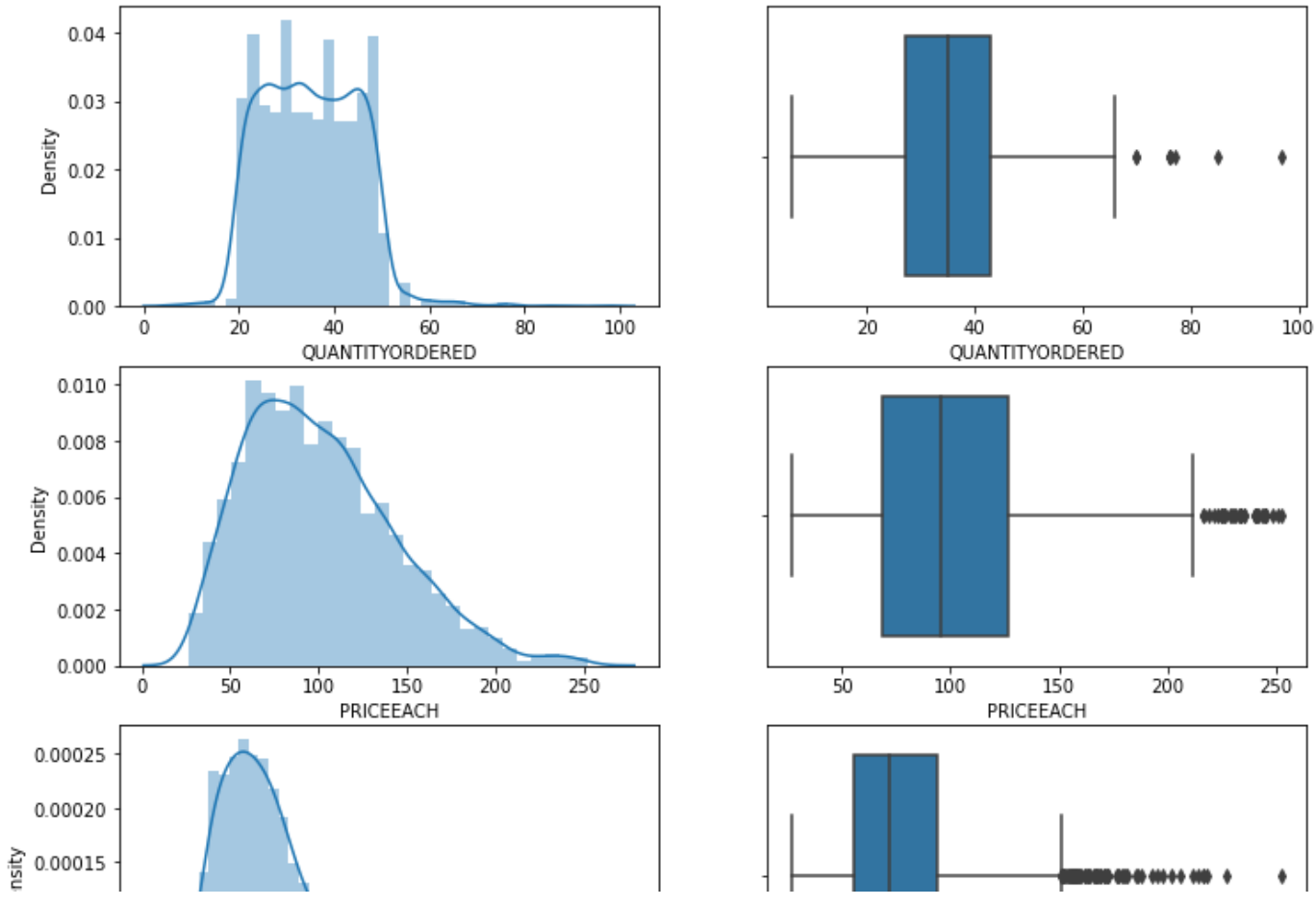


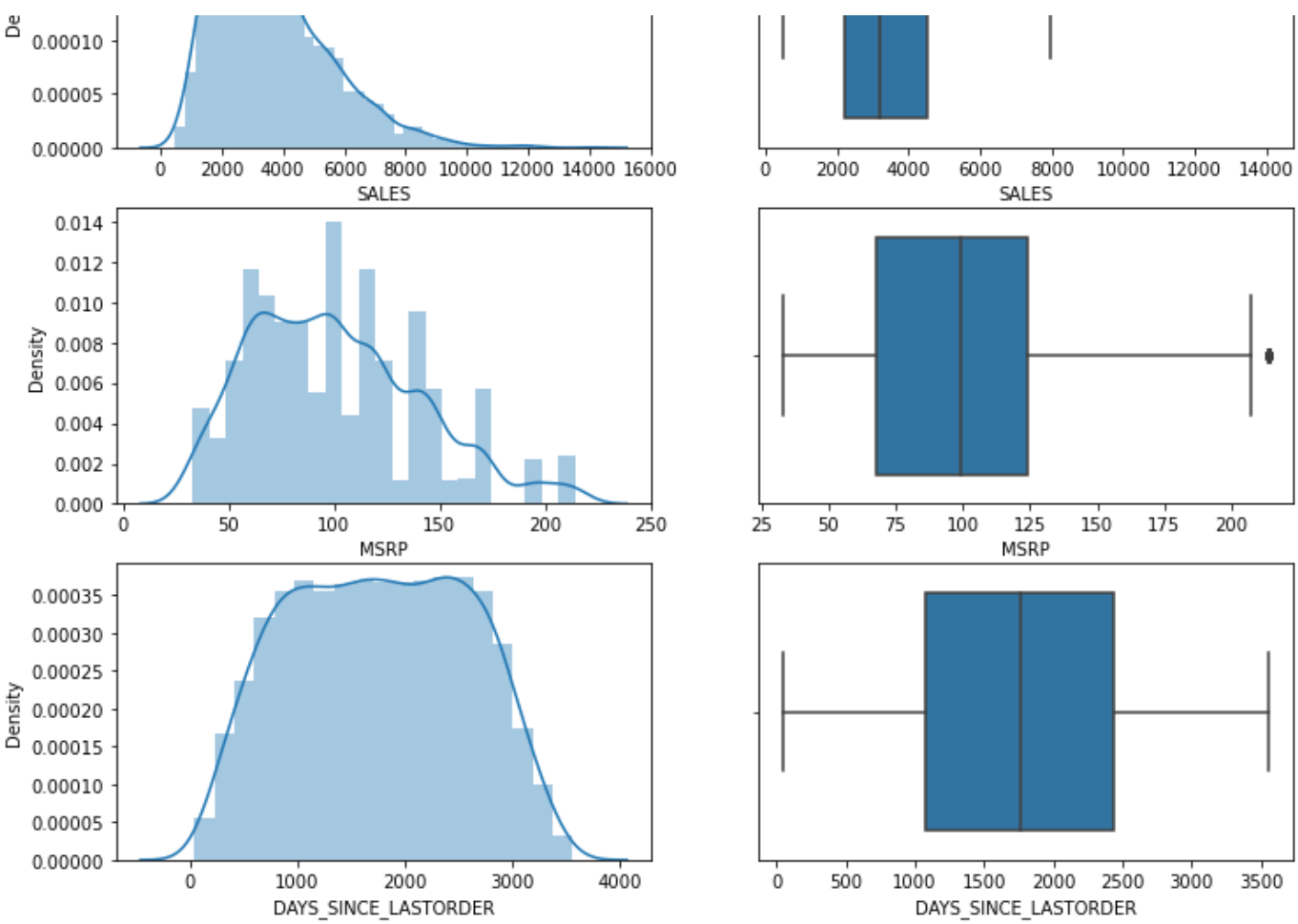


'STATUS', 'PRODUCTLINE', 'PRODUCTCODE', 'CUSTOMERNAME', 'PHONE', 'ADDRESSLINE1', 'CITY', 'POSTALCODE', 'COUNTRY', 'CONTACTLASTNAME', 'CONTACTFIRSTNAME', 'DEALSIZE'

In [15]:

```
fig, axes = plt.subplots(nrows=5,ncols=2)
fig.set_size_inches(12,18)

a = sns.distplot(df['QUANTITYORDERED'] , ax=axes[0][0])

a = sns.boxplot(df['QUANTITYORDERED'] , orient = "v" , ax=axes[0][1])

a = sns.distplot(df['PRICEEACH'] , ax=axes[1][0])

a = sns.boxplot(df['PRICEEACH'] , orient = "v" , ax=axes[1][1])

a = sns.distplot(df['SALES'] , ax=axes[2][0])

a = sns.boxplot(df['SALES'] , orient = "v" , ax=axes[2][1])

a = sns.distplot(df['MSRP'] , ax=axes[3][0])

a = sns.boxplot(df['MSRP'] , orient = "v" , ax=axes[3][1])

a = sns.distplot(df['DAYS_SINCE_LASTORDER'] , ax=axes[4][0])

a = sns.boxplot(df['DAYS_SINCE_LASTORDER'] , orient = "v" , ax=axes[4][1])

plt.show()
```



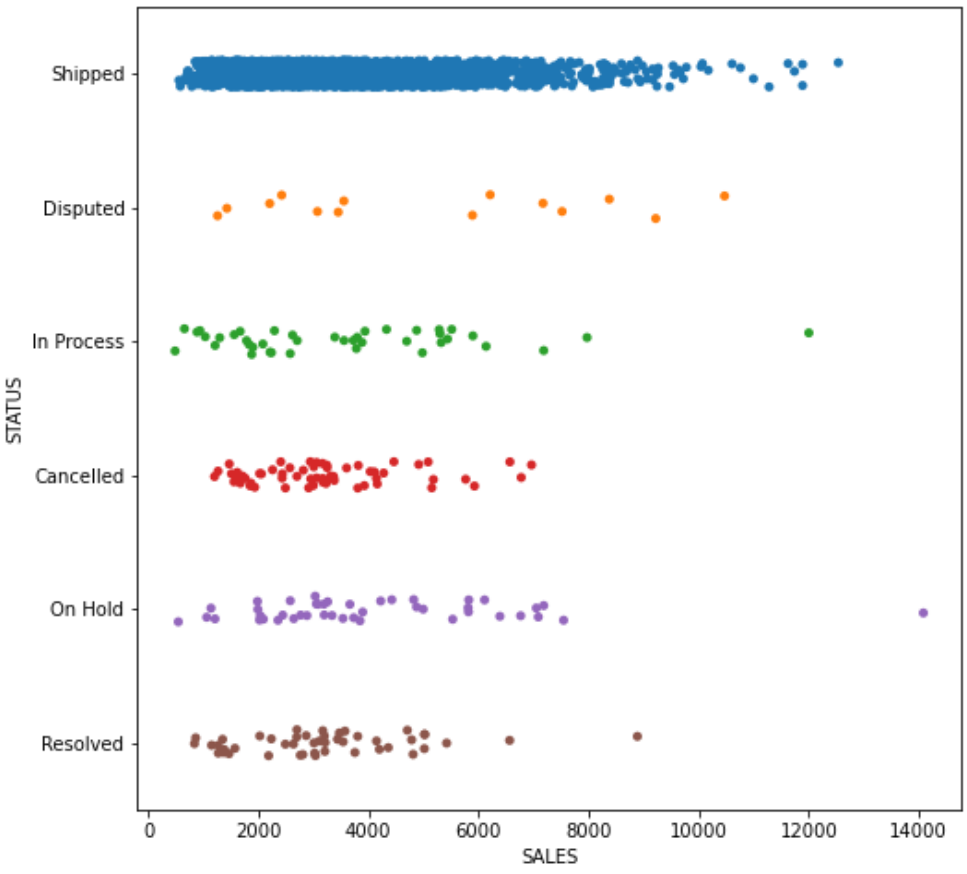


In [ ]:

```
```

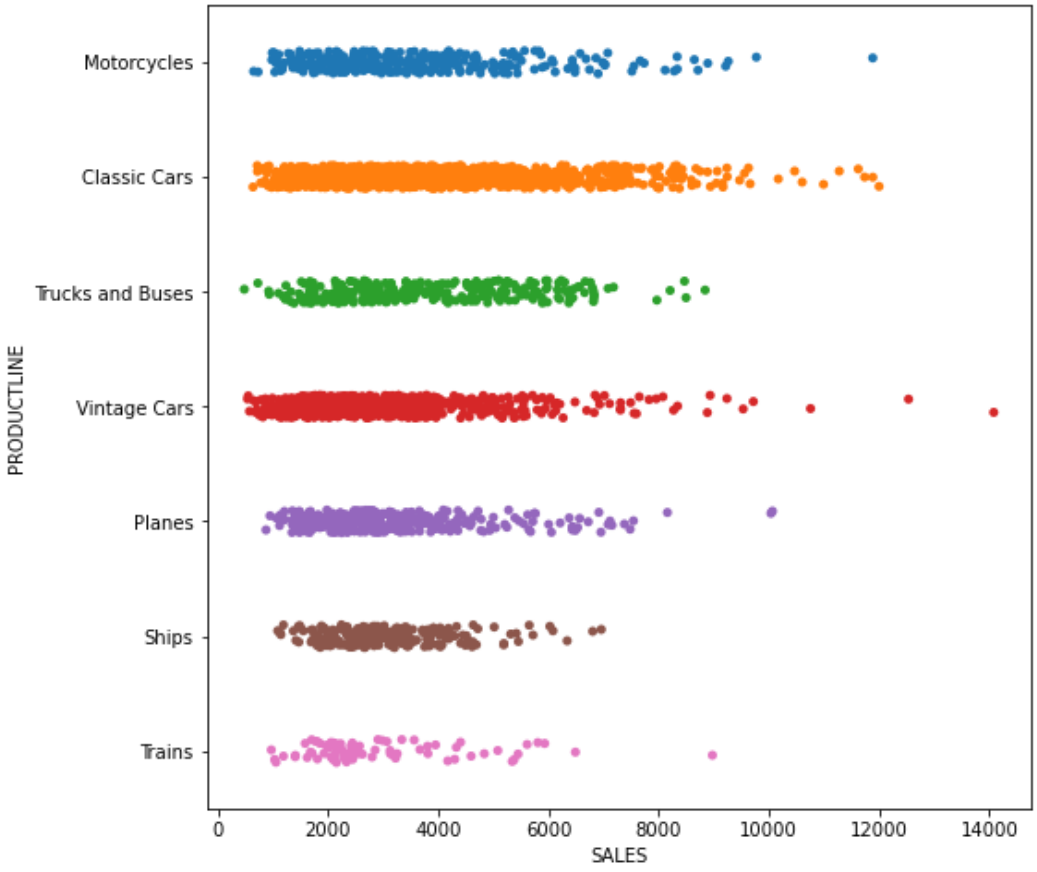
In [16]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['STATUS'], jitter=True)
plt.show()
```



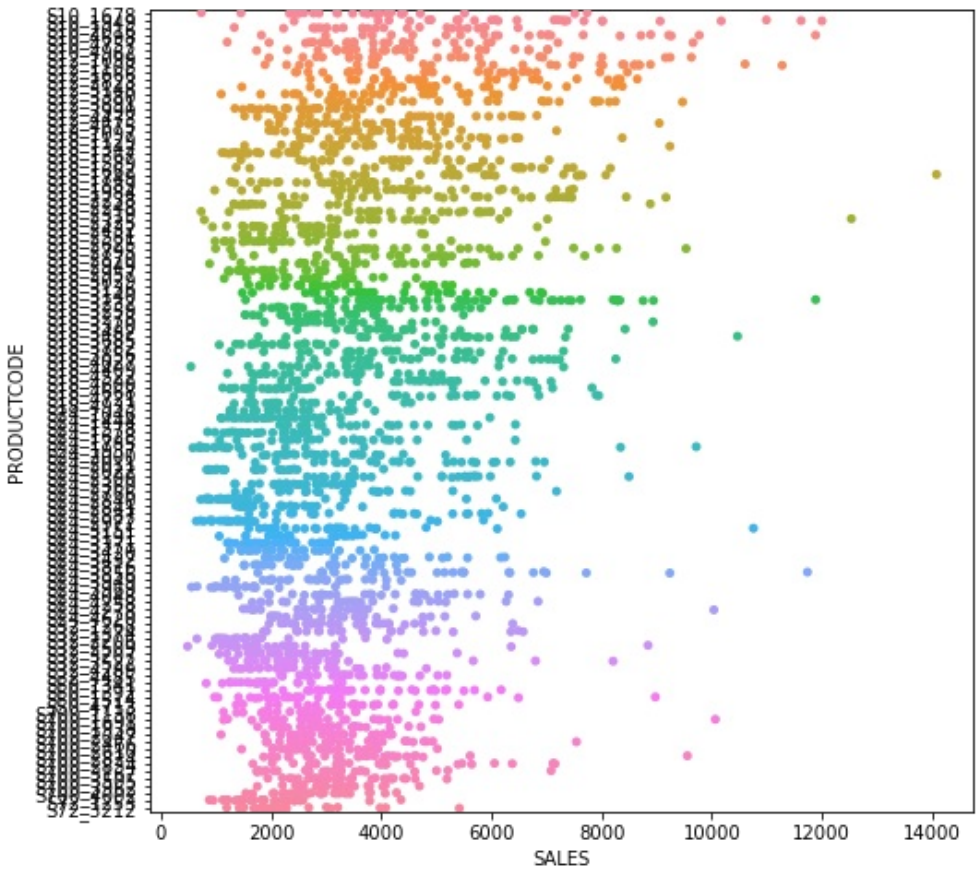
In [17]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['PRODUCTLINE'], jitter=True)
plt.show()
```



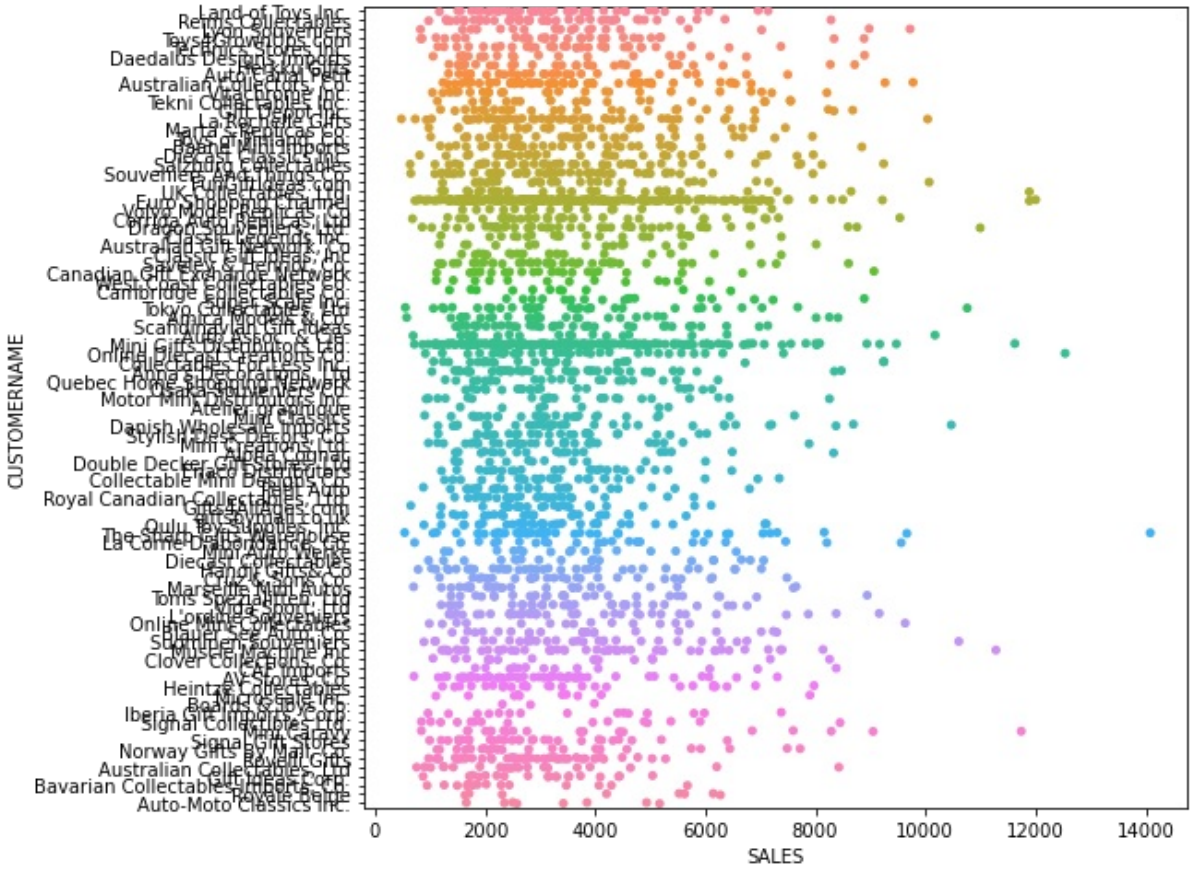
In [18]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['PRODUCTCODE'], jitter=True)
plt.show()
```



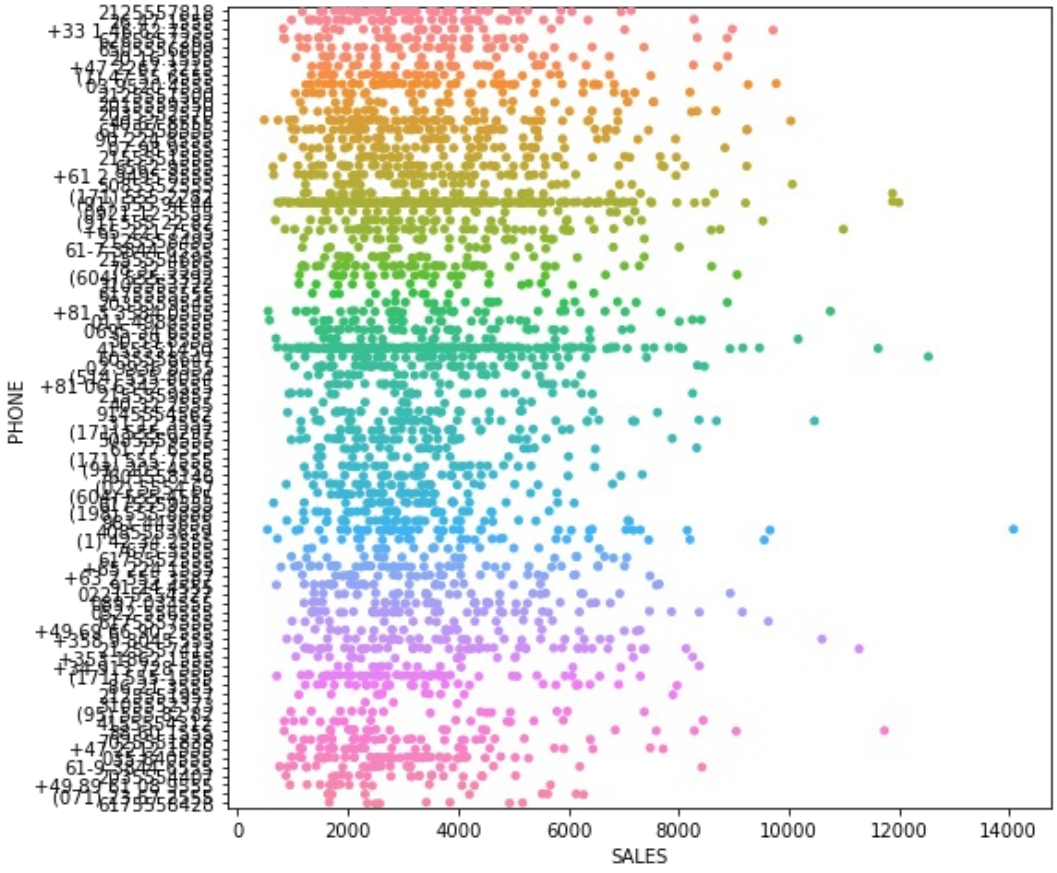
In [19]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['CUSTOMERNAME'], jitter=True)
plt.show()
```



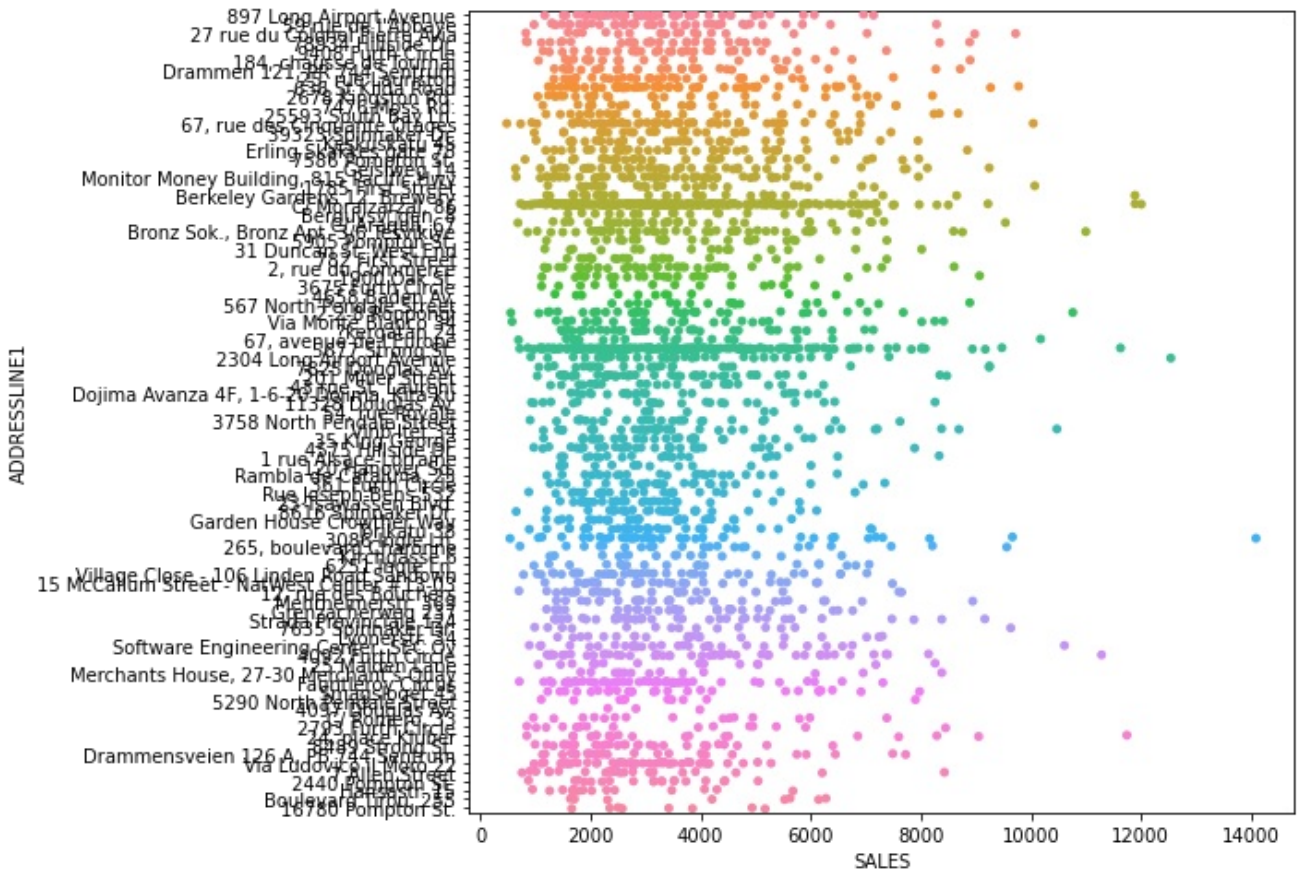
In [20]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['PHONE'], jitter=True)
plt.show()
```



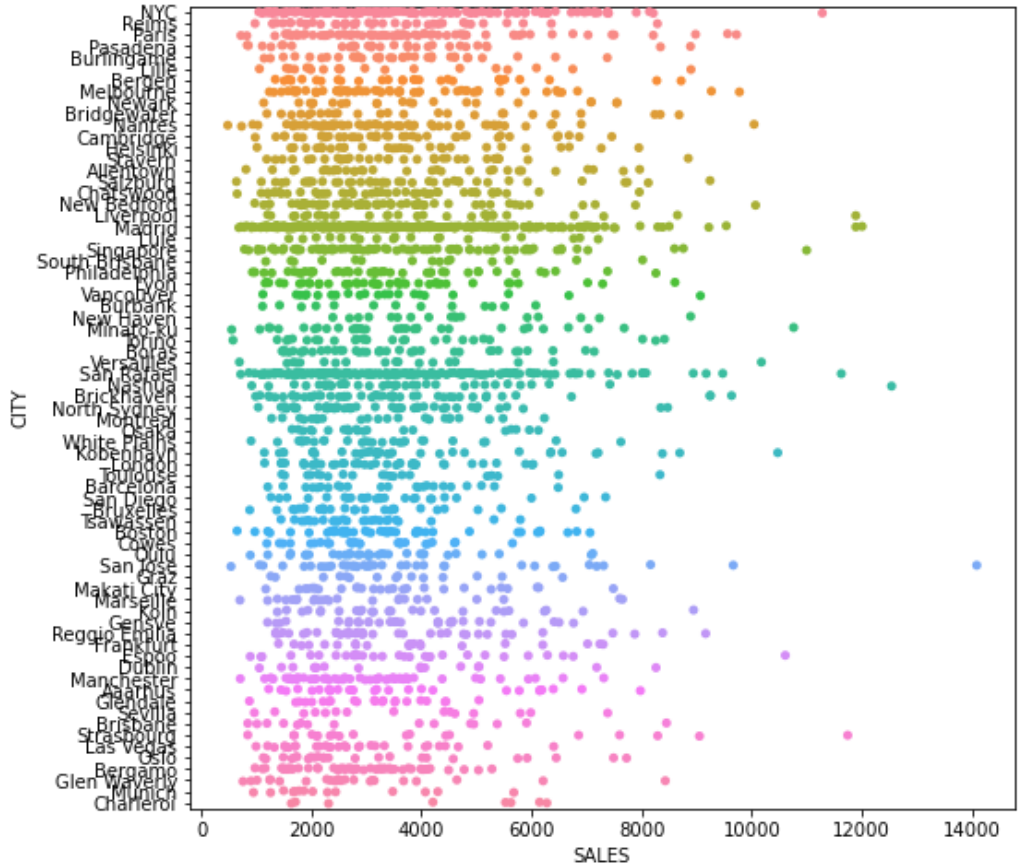
In [21]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['ADDRESSLINE1'], jitter=True)
plt.show()
```



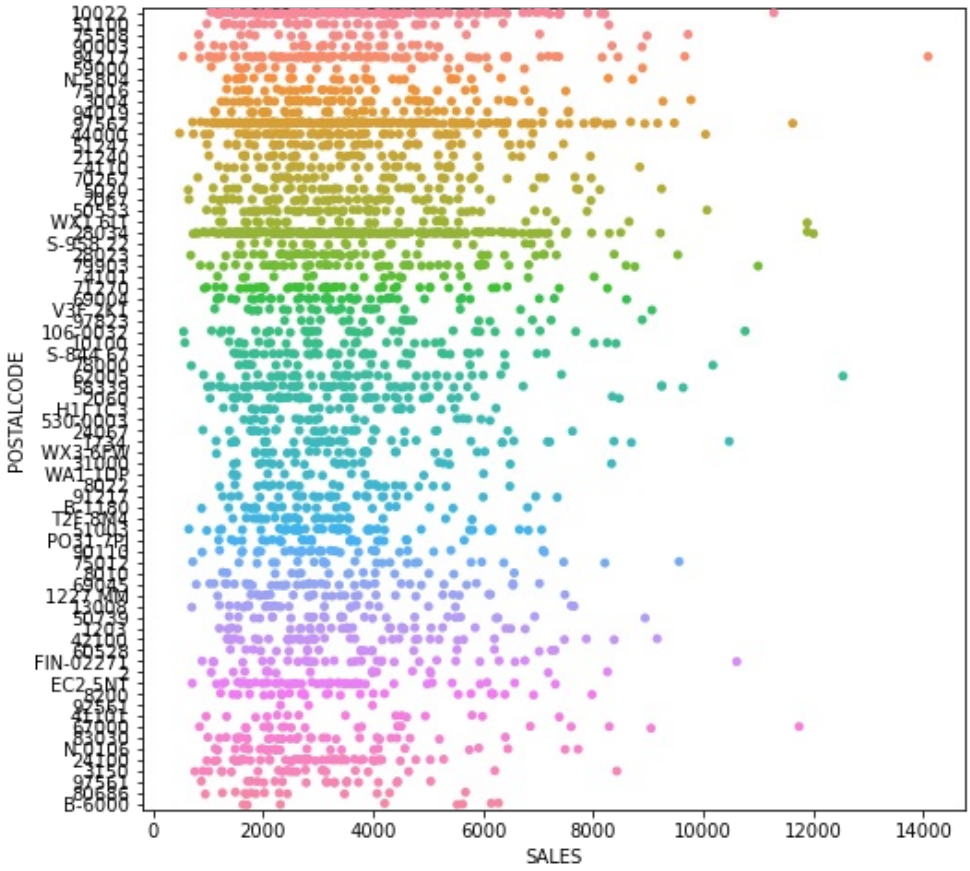
In [22]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['CITY'], jitter=True)
plt.show()
```



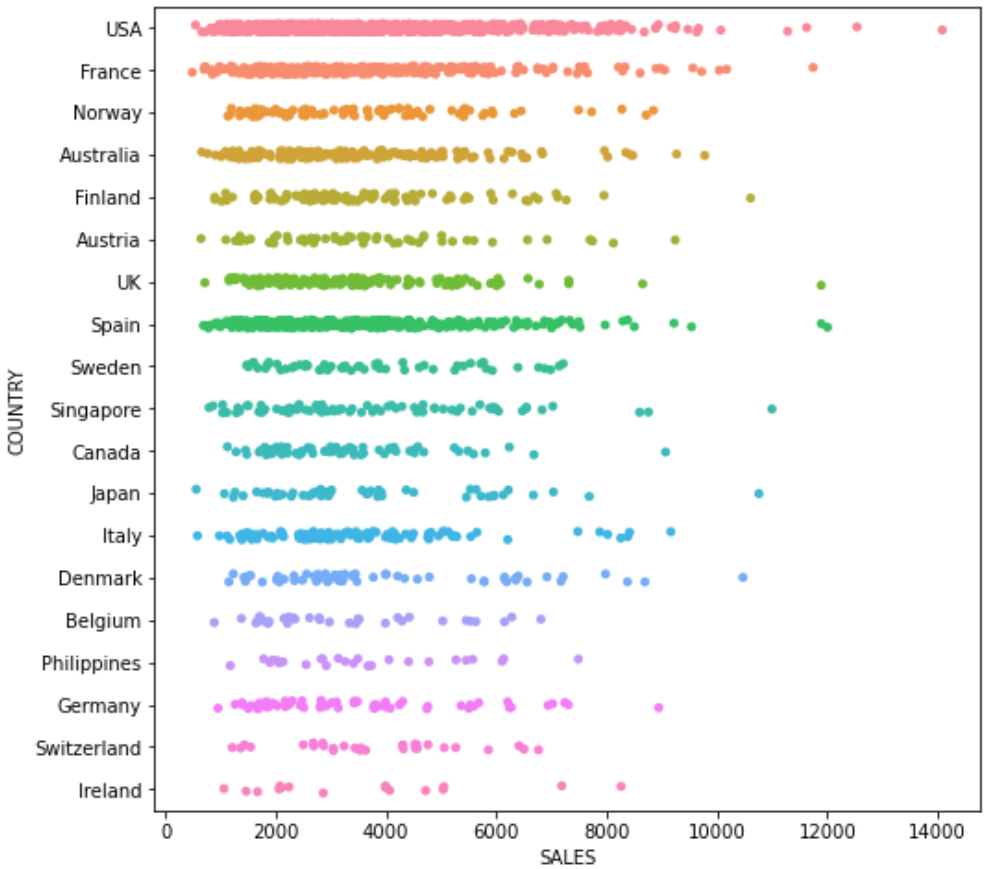
In [23]:

```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['POSTALCODE'], jitter=True)
plt.show()
```



In [24]:

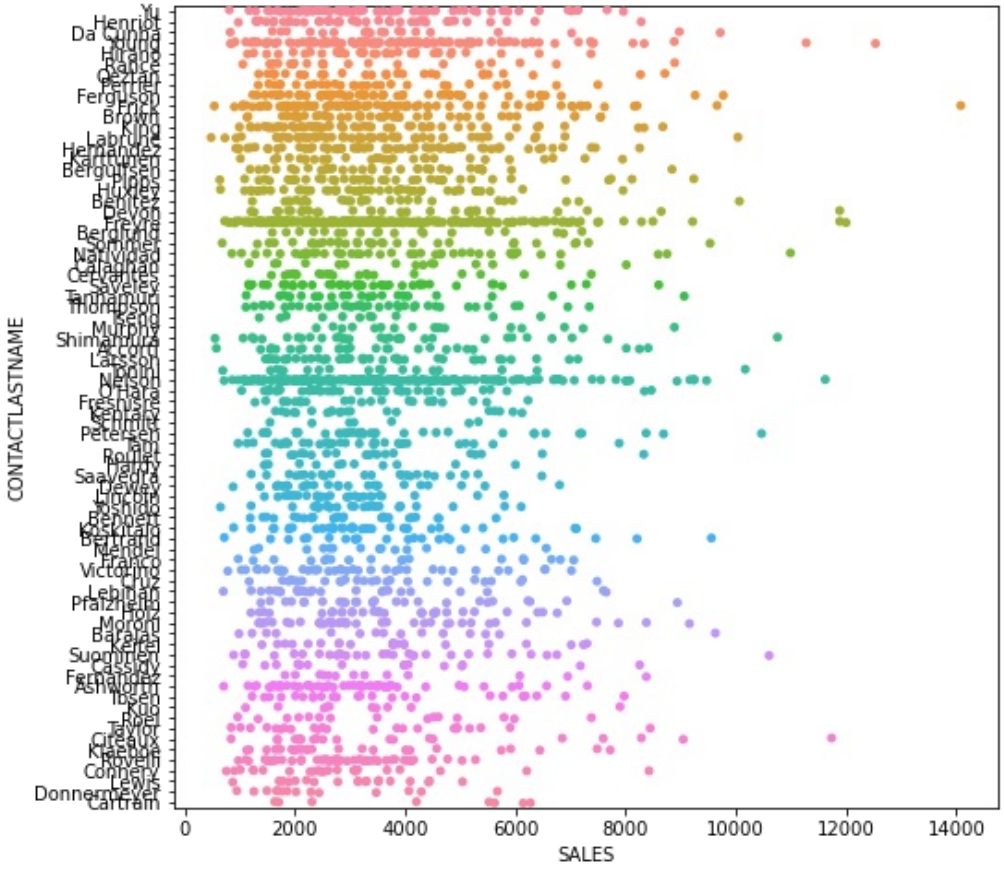
```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['COUNTRY'], jitter=True)
plt.show()
```



In [25]:

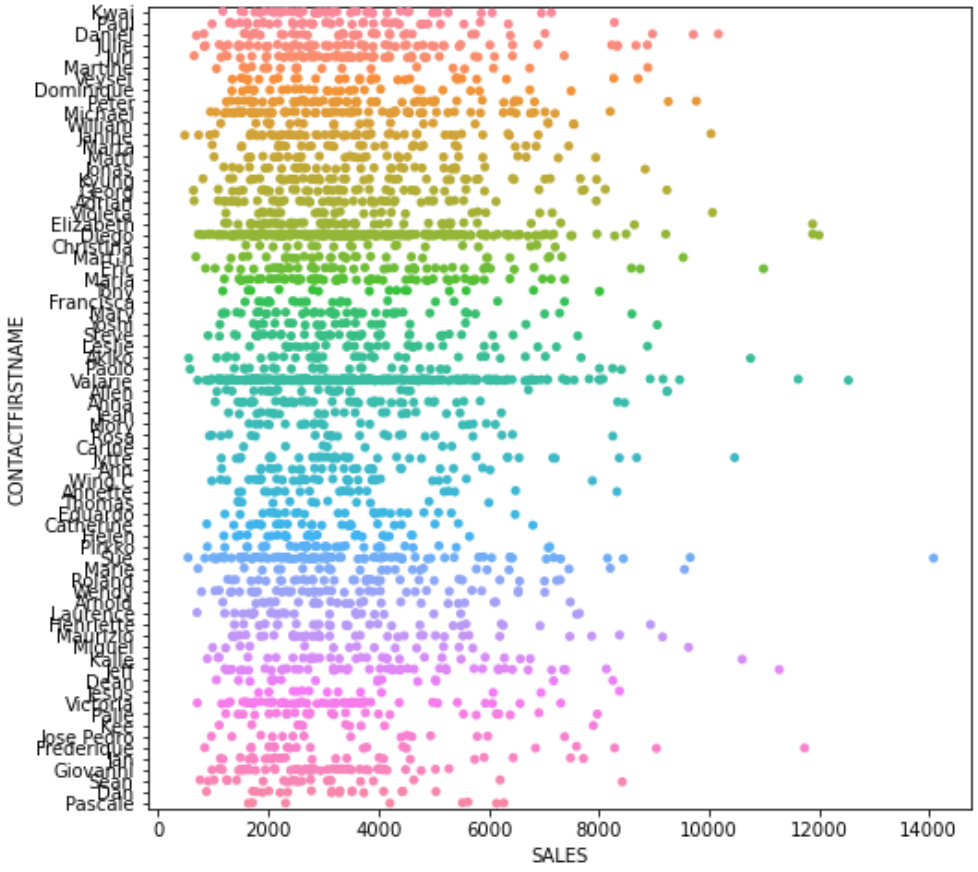
```
plt.figure(figsize=(8,8))
```

```
sns.stripplot(df["SALES"], df['CONTACTLASTNAME'], jitter=True)
plt.show()
```



In [26]:

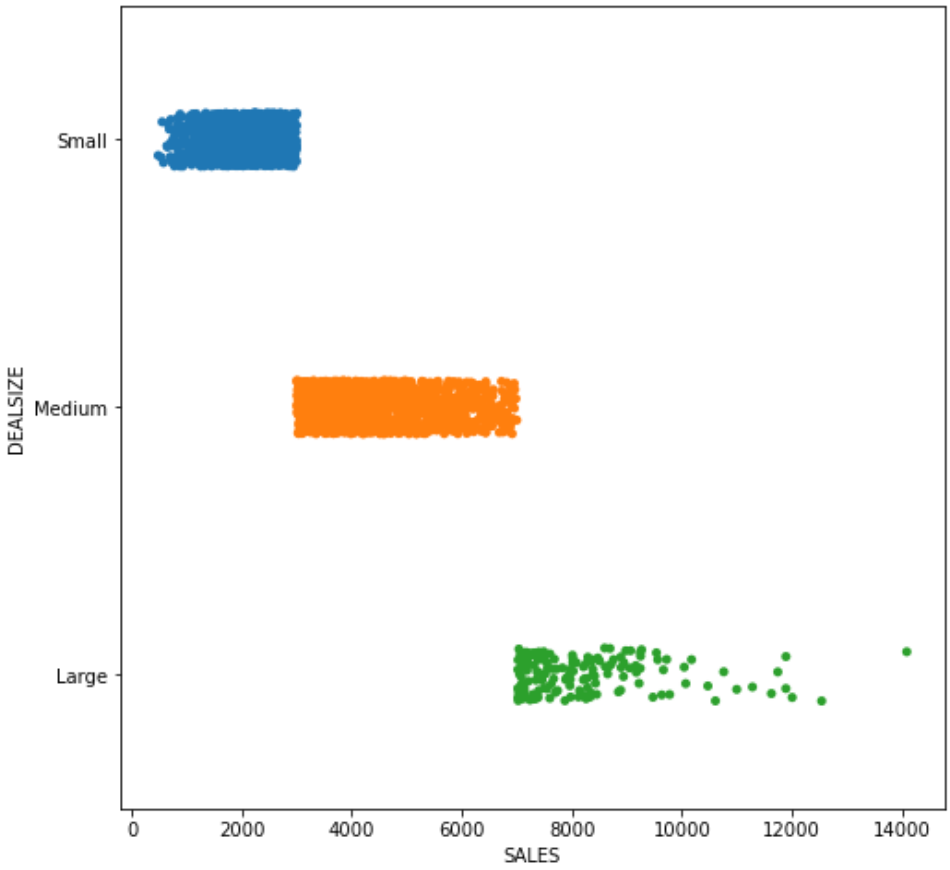
```
plt.figure(figsize=(8,8))
sns.stripplot(df["SALES"], df['CONTACTFIRSTNAME'], jitter=True)
plt.show()
```



In [27]:

```
plt.figure(figsize=(8,8))
```

```
sns.stripplot(df["SALES"], df['DEALSIZE'], jitter=True)
plt.show()
```

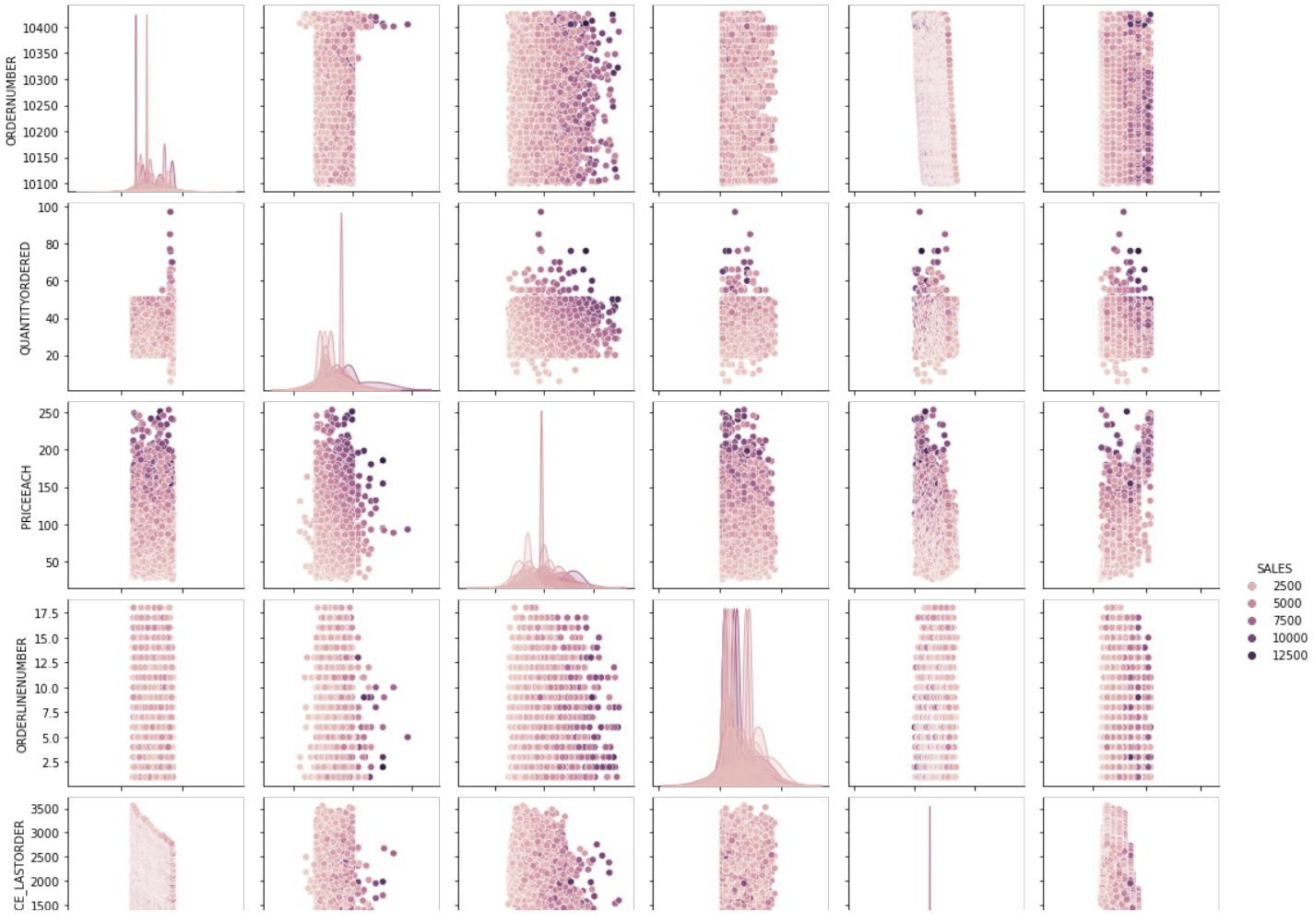


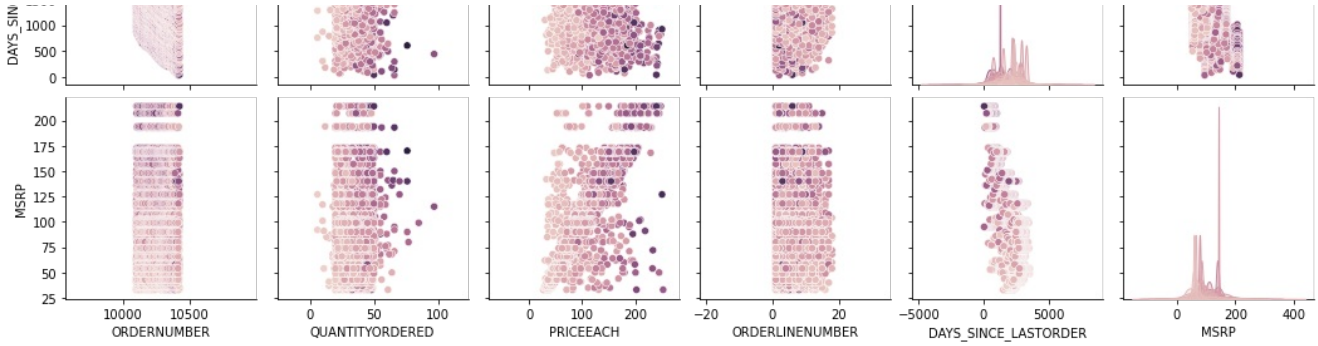
In [28]:

```
sns.pairplot(df,hue="SALES")
```

Out[28]:

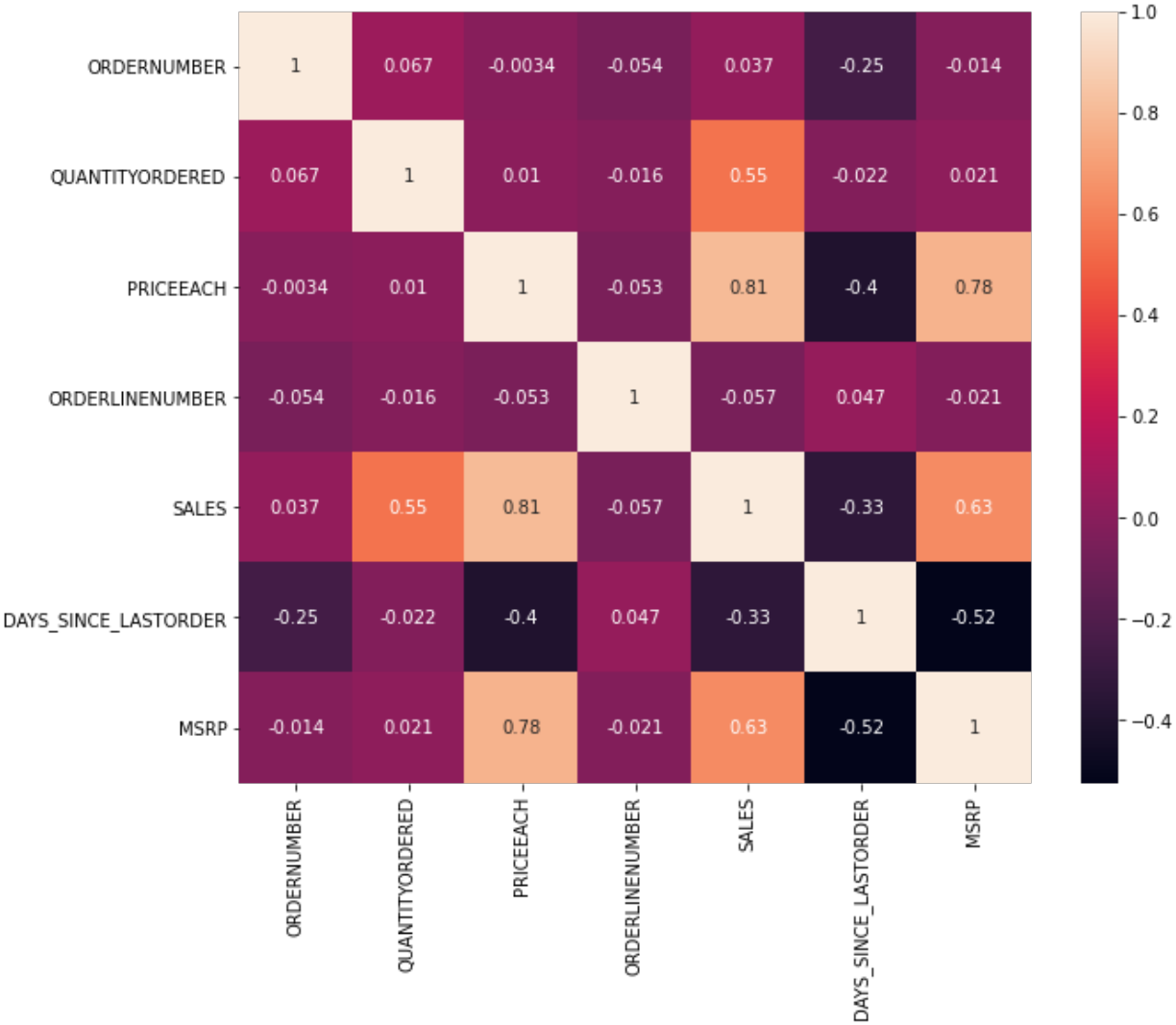
<seaborn.axisgrid.PairGrid at 0x2289f31dc10>





In [29]:

```
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(),annot=True)
plt.show()
```

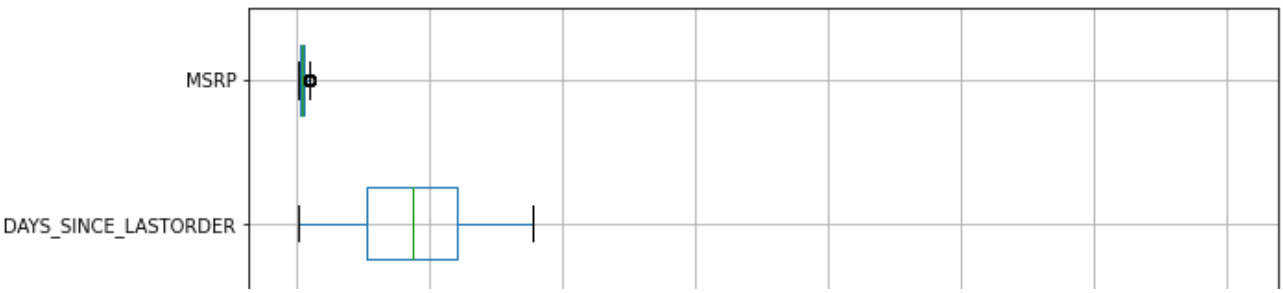


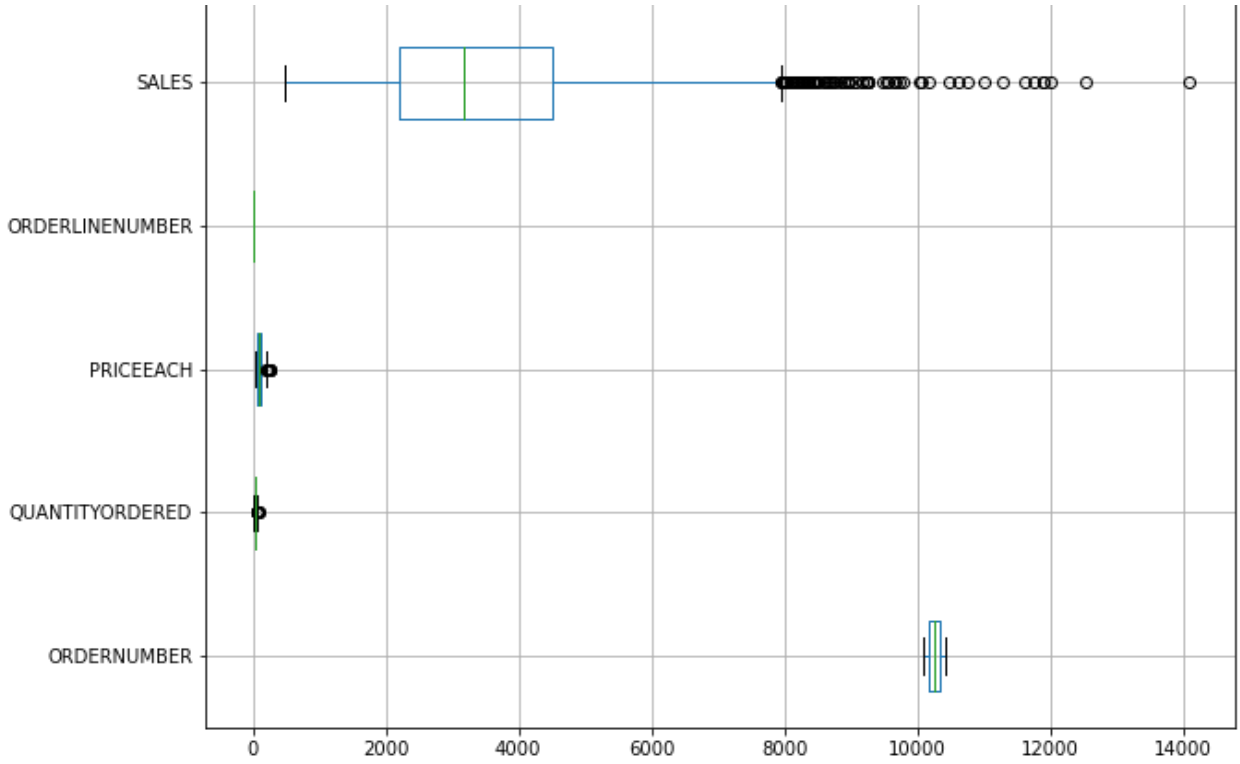
In [30]:

```
plt.figure(figsize=(10,10))
df[num].boxplot(vert=0)
```

Out[30]:

<AxesSubplot:>





In [ ]:

```
```

In [31]:

```
df_t = df.filter(['ORDERDATE','SALES'],axis=1)
```

In [32]:

```
df_t.describe()
```

Out[32]:

	SALES
count	2747.000000
mean	3553.047583
std	1838.953901
min	482.130000
25%	2204.350000
50%	3184.800000
75%	4503.095000
max	14082.800000

In [33]:

```
df_t.to_csv('SALES_TSF.csv', index=False)
```

In [34]:

```
df_t.dtypes.value_counts()
```

Out[34]:

```
float64           1
datetime64[ns]    1
dtype: int64
```

In [35]:

```
df_t.head()
```

Out[35]:

	ORDERDATE	SALES
0	2018-02-24	2871.00
1	2018-05-07	2765.90
2	2018-07-01	3884.34
3	2018-08-25	3746.70
4	2018-10-28	3479.76

In [36]:

```
dft = pd.read_csv("SALES_TSF.csv",parse_dates=True,squeeze=True,index_col=0)
```

In [37]:

```
dft.head()
```

Out[37]:

```
ORDERDATE
2018-02-24    2871.00
2018-05-07    2765.90
2018-07-01    3884.34
2018-08-25    3746.70
2018-10-28    3479.76
Name: SALES, dtype: float64
```

In [38]:

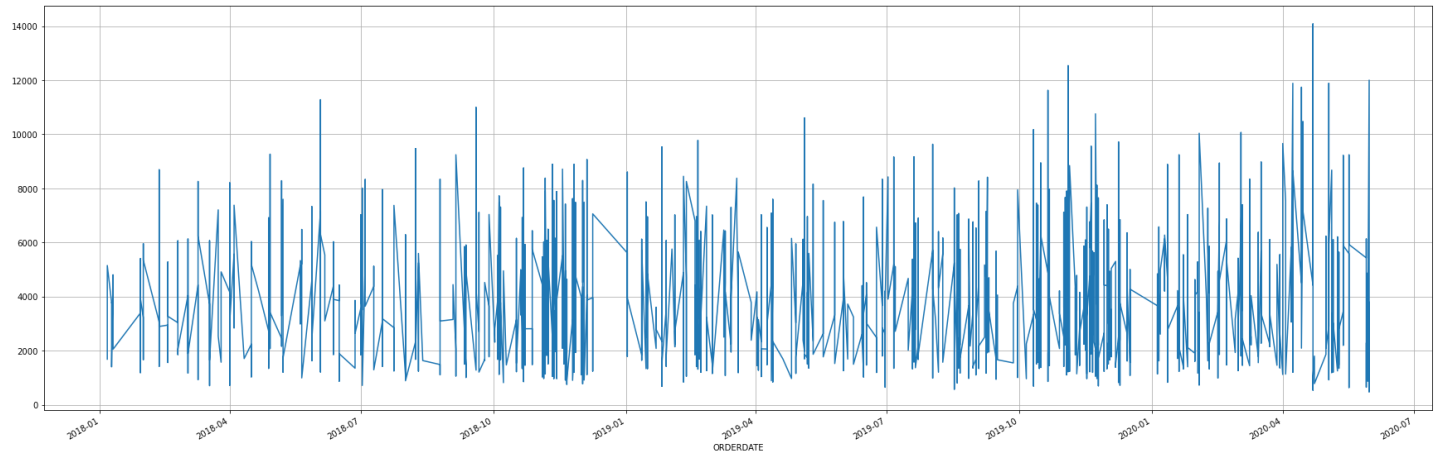
```
dft.tail()
```

Out[38]:

```
ORDERDATE
2019-12-02    2244.40
2020-01-31    3978.51
2020-03-01    5417.57
2020-03-28    2116.16
2020-05-06    3079.44
Name: SALES, dtype: float64
```

In [39]:

```
plt.figure(figsize=(30,10))
dft.plot();
plt.grid()
```



Weekly Plot

In [40]:

In [40]:

```
df_weekly_sum = dft.resample('W').sum()
df_weekly_sum
```

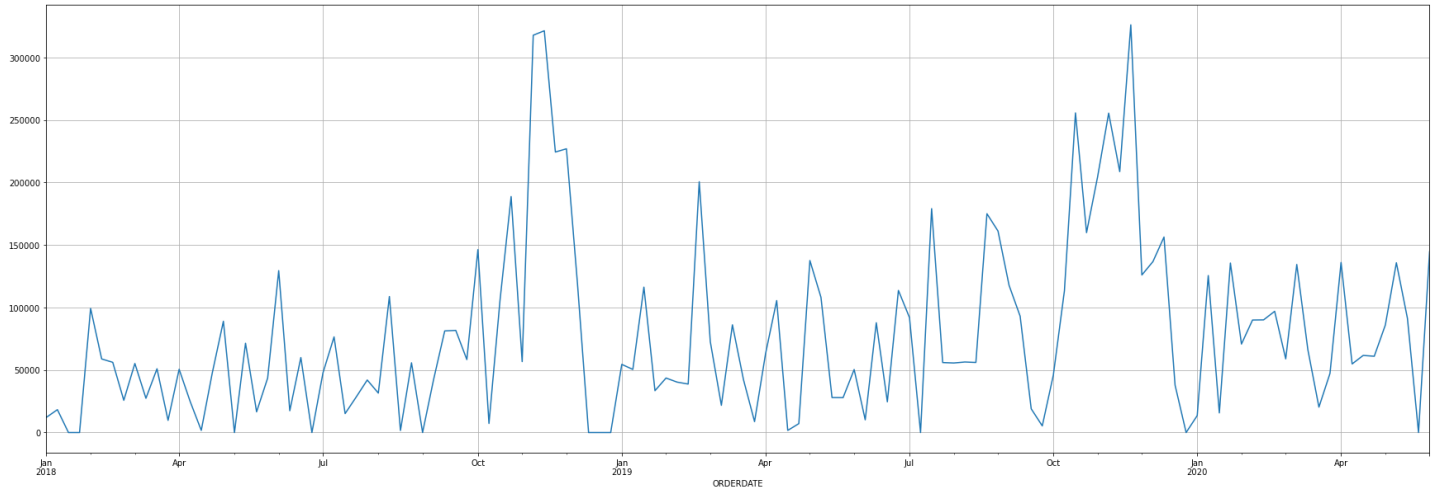
Out[40]:

```
ORDERDATE
2018-01-07     12133.25
2018-01-14     18296.39
2018-01-21         0.00
2018-01-28         0.00
2018-02-04     99323.96
                 ...
2020-05-03     85980.71
2020-05-10    135853.39
2020-05-17     91297.00
2020-05-24         0.00
2020-05-31    144729.96
Freq: W-SUN, Name: SALES, Length: 126, dtype: float64
```

The values which the original series cannot provide is taken as 0 by python if we try to resample the data on a daily basis.

In [41]:

```
plt.figure(figsize=(30,10))
df_weekly_sum.plot()
plt.grid();
```



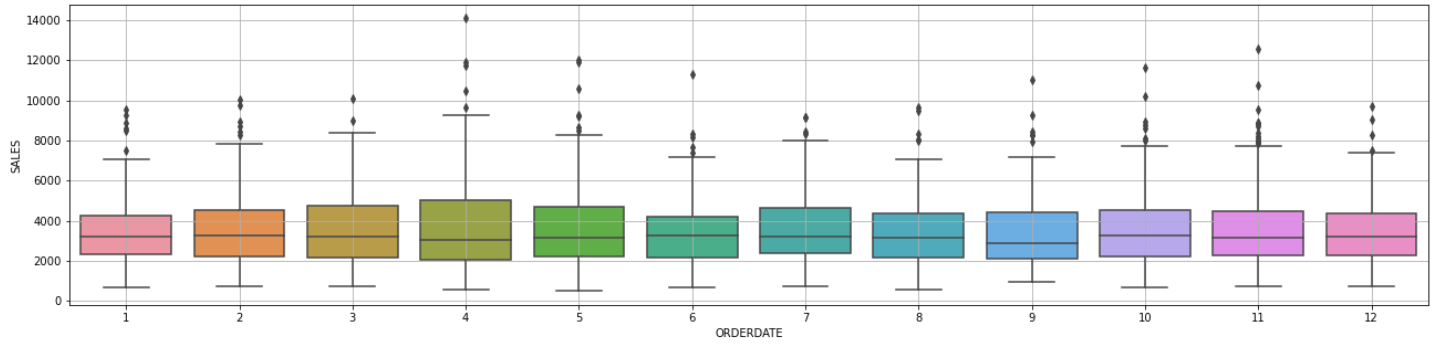
In [ ]:

```

```

Monthly Plot

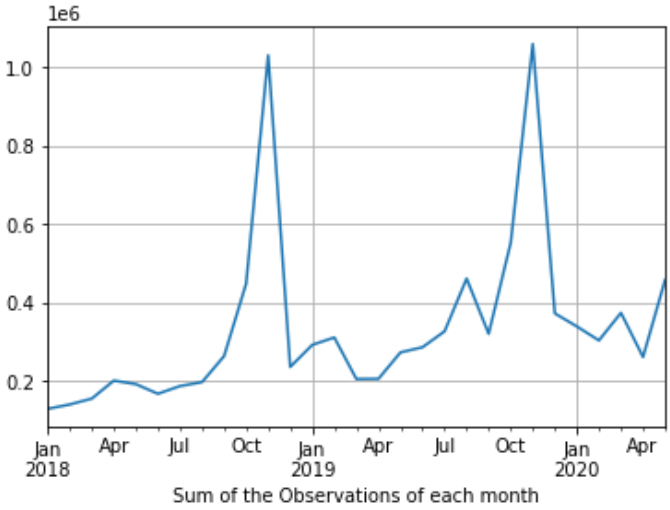
In [42]:

```
fig, ax = plt.subplots(figsize=(22,5))
sns.boxplot(dft.index.month, dft, ax=ax,whis=1.5)
plt.grid();
```



In [43]:

```
df_monthly_sum = dft.resample('M').sum()
df_monthly_sum.head()

df_monthly_sum.plot();
plt.grid()
plt.xlabel('Sum of the Observations of each month');
```



In [ ]:

```

```

In [44]:

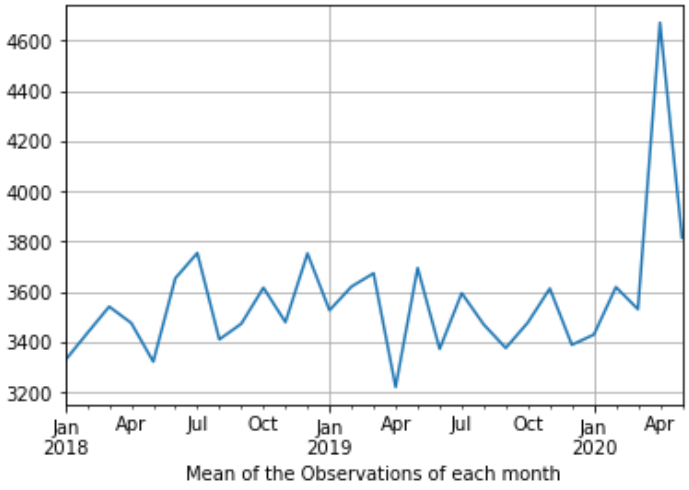
```
df_monthly_mean = dft.resample('M').mean()
df_monthly_mean.head()
```

Out[44]:

```
ORDERDATE
2018-01-31    3327.015385
2018-02-28    3435.029024
2018-03-31    3541.120909
2018-04-30    3476.026724
2018-05-31    3321.950172
Freq: M, Name: SALES, dtype: float64
```

In [45]:

```
df_monthly_mean.plot();
plt.grid()
plt.xlabel('Mean of the Observations of each month');
```



Quarterly Plot

In [46]:

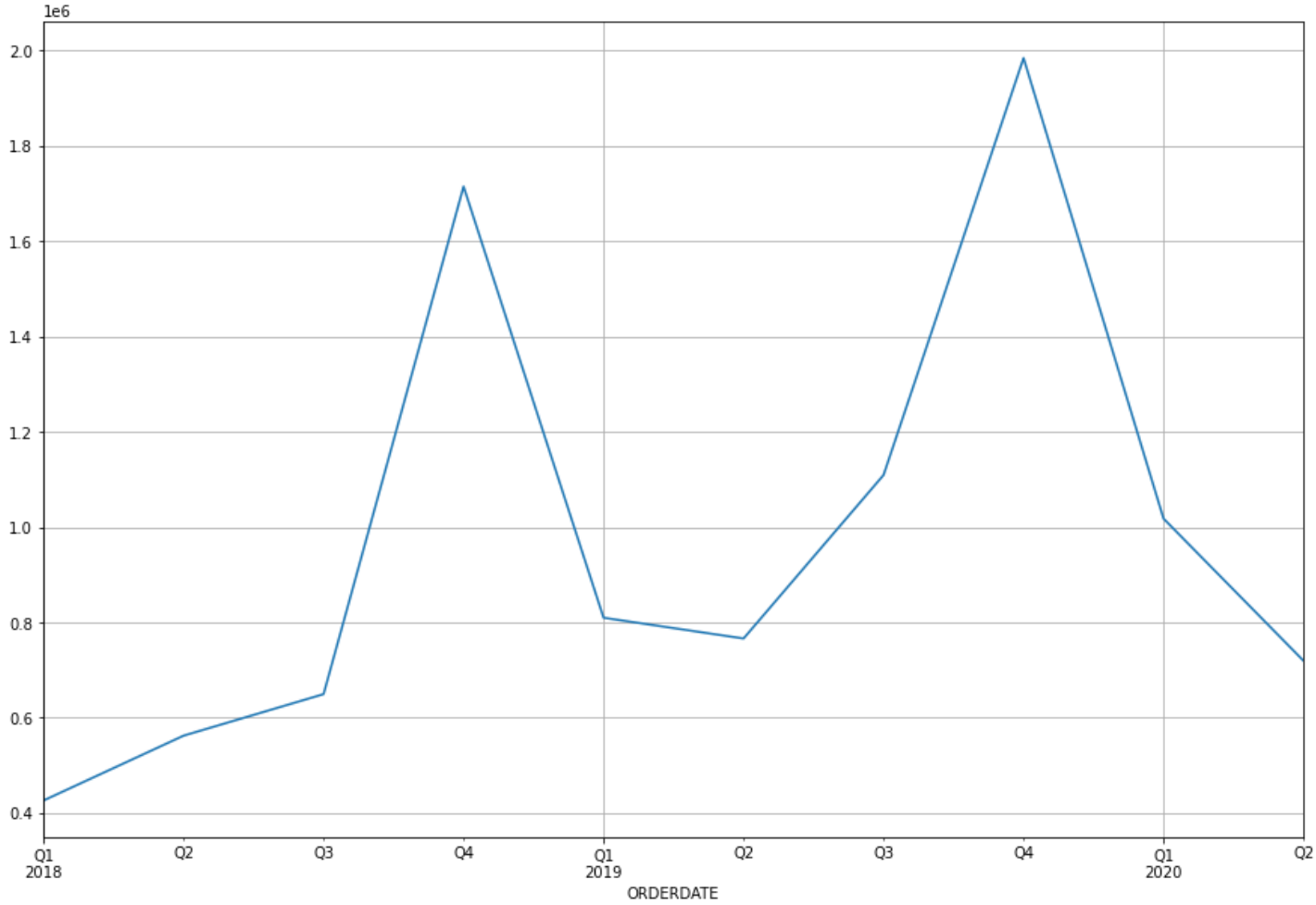
```
df_quarterly_sum = dft.resample('Q').sum()
df_quarterly_sum.head()
```

Out[46]:

```
ORDERDATE
2018-03-31     426399.11
2018-06-30     562365.22
2018-09-30     649514.54
2018-12-31    1714735.19
2019-03-31     809841.36
Freq: Q-DEC, Name: SALES, dtype: float64
```

In [47]:

```
plt.figure(figsize=(15,10))
df_quarterly_sum.plot();
plt.grid()
```



In [48]:

```
df_quarterly_mean = dft.resample('Q').mean()
df_quarterly_mean.head()
```

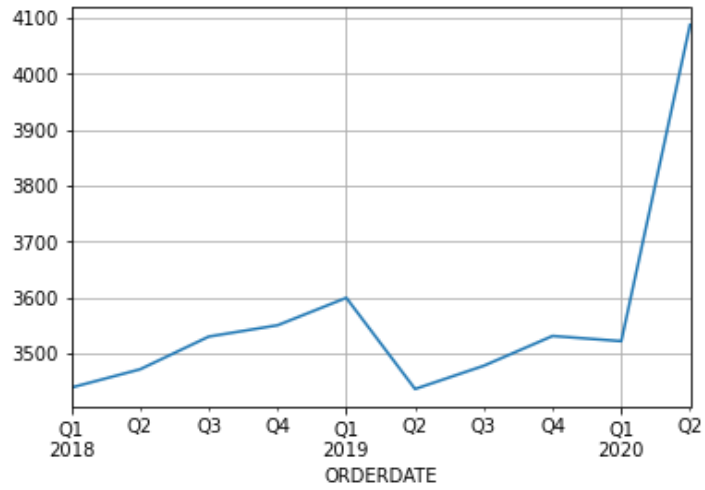
Out[48]:

```
ORDERDATE
2018-03-31    3438.702500
2018-06-30    3471.390247
2018-09-30    3529.970326
2018-12-31    3550.176377
2019-03-31    3599.294933
Freq: Q-DEC, Name: SALES, dtype: float64
```

In [49]:

```
df_quarterly_mean.plot();
```

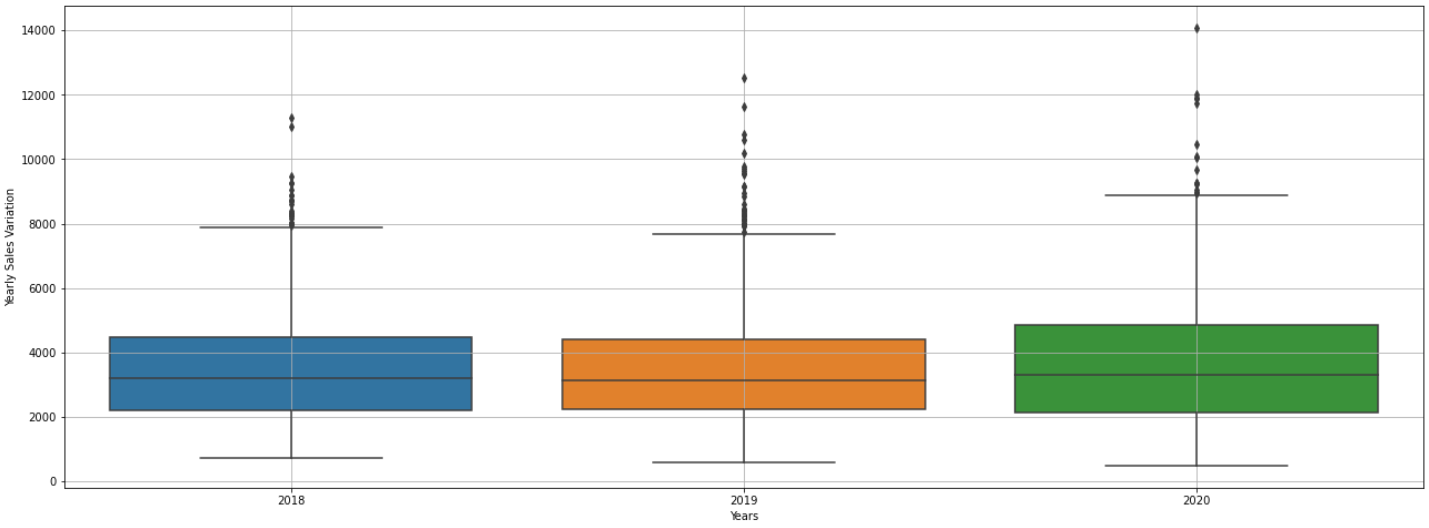
```
plt.grid()
```



YEARLY PLOT

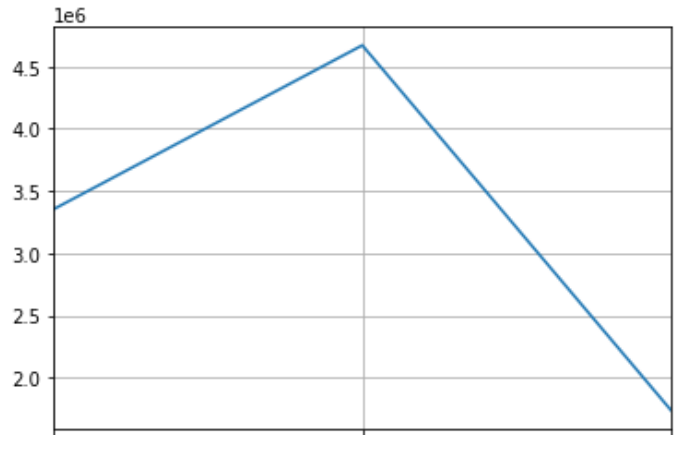
In [50]:

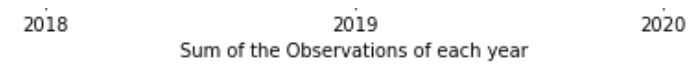
```
fig, ax = plt.subplots(figsize=(22,8))
sns.boxplot(dft.index.year, dft, ax=ax,whis=1.5)
plt.grid();
plt.xlabel('Years');
plt.ylabel('Yearly Sales Variation');
```



In [51]:

```
df_yearly_sum = dft.resample('A').sum()
df_yearly_sum.head()

df_yearly_sum.plot();
plt.grid()
plt.xlabel('Sum of the Observations of each year');
```





In [52]:

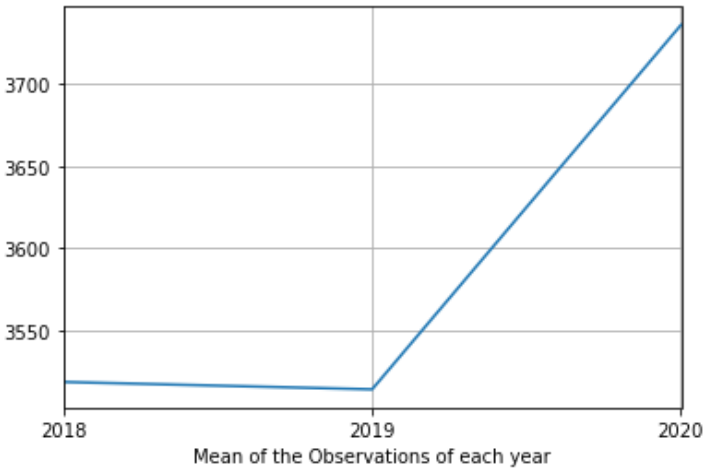
```
df_yearly_mean = dft.resample('Y').mean()
df_yearly_mean.head()
```

Out[52]:

```
ORDERDATE
2018-12-31    3518.377817
2019-12-31    3513.863476
2020-12-31    3736.092667
Freq: A-DEC, Name: SALES, dtype: float64
```

In [53]:

```
df_yearly_mean.plot();
plt.grid()
plt.xlabel('Mean of the Observations of each year');
```

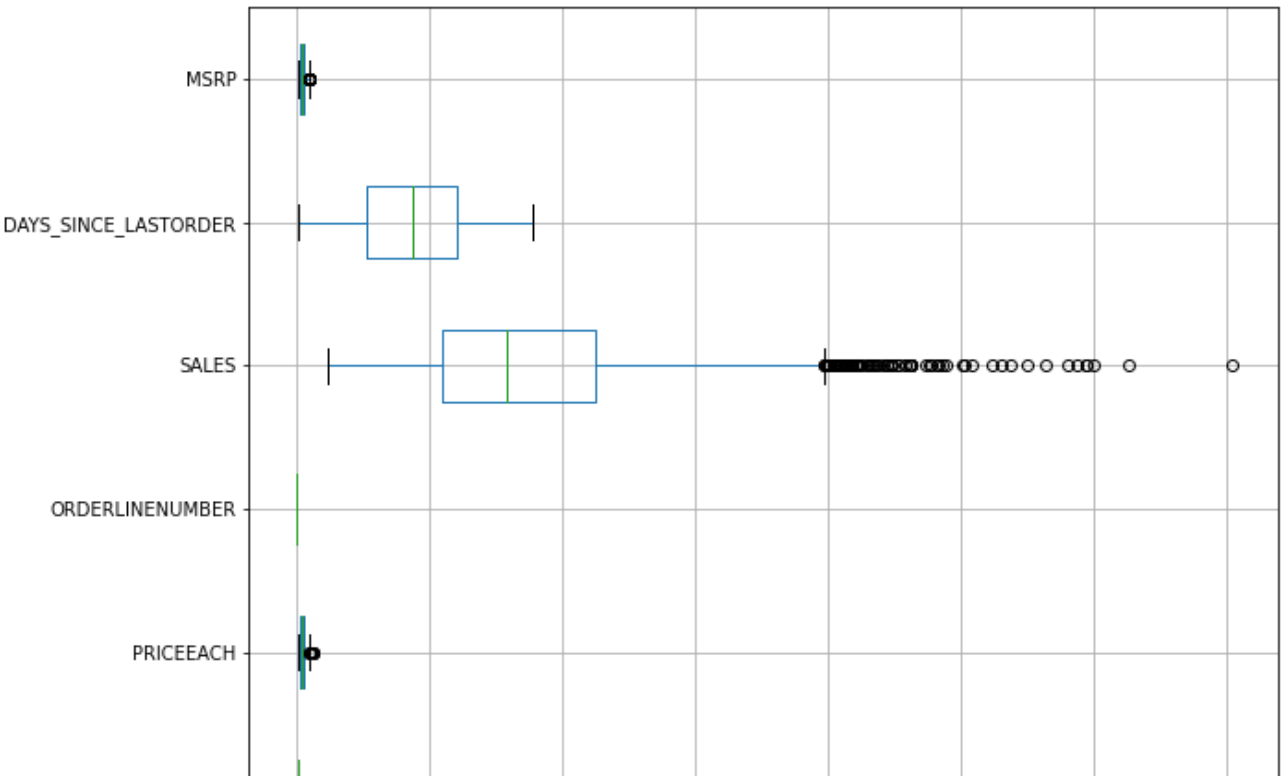


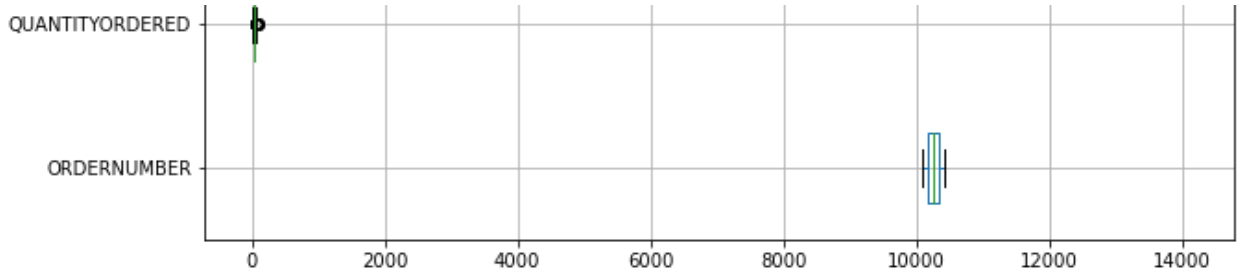
In [54]:

```
plt.figure(figsize=(10,10))
df[num].boxplot(vert=0)
```

Out[54]:

```
<AxesSubplot:>
```



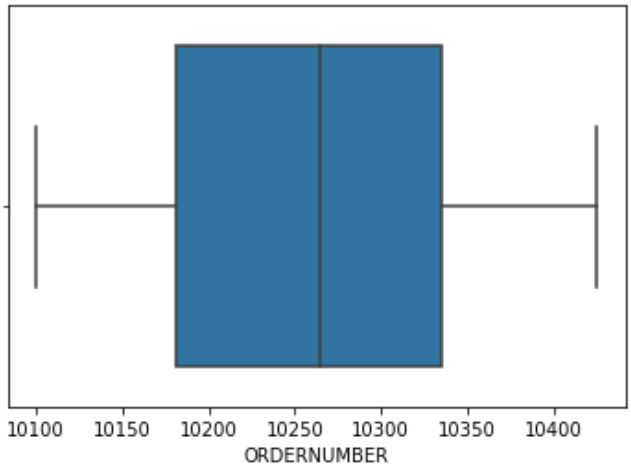


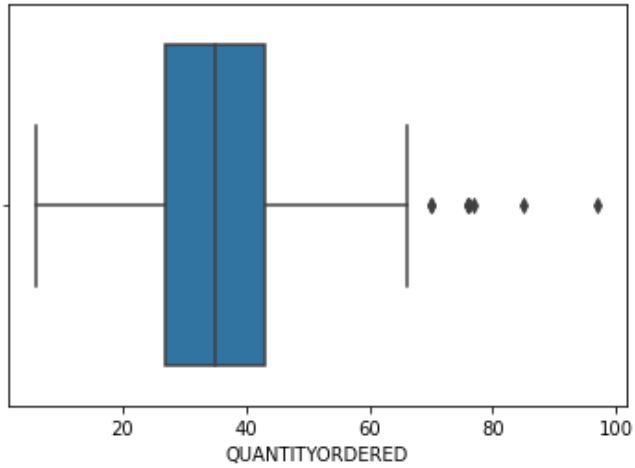
In [55]:

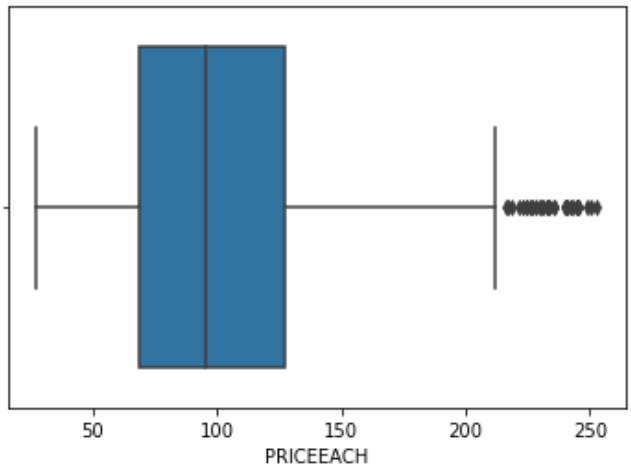
```
## Outlier Removal
```

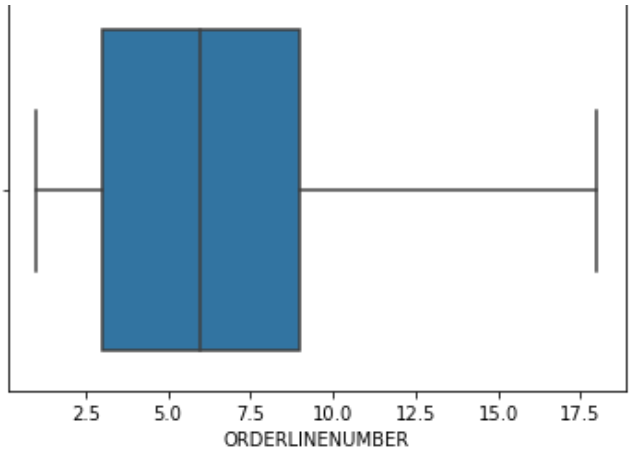
In [56]:

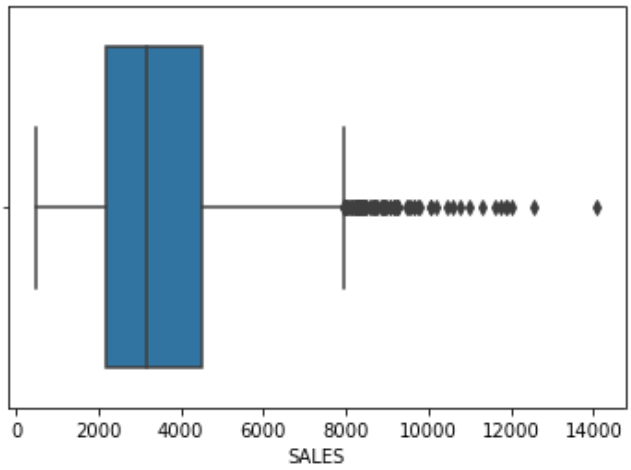
```
cols = ['ORDERNUMBER','QUANTITYORDERED','PRICEEACH','ORDERLINENUMBER','SALES','DAYS_SINC
E_LASTORDER','MSRP']
for i in cols:
    sns.boxplot(df[i],whis=1.5)
    plt.show();
```

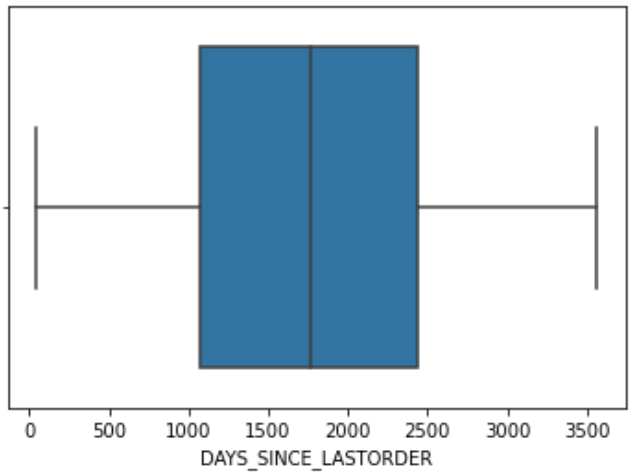


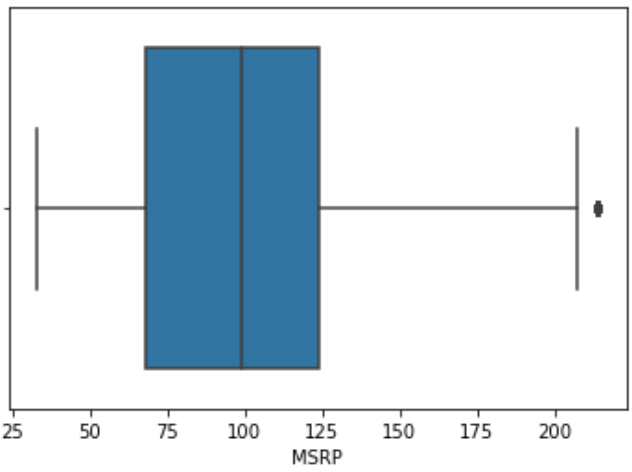












In [57]:

```
def remove_outlier(col):
    sorted(col)
    Q1,Q3=np.percentile(col,[25,75])
    IQR=Q3-Q1
    lower_range= Q1-(1.5 * IQR)
```

```
    upper_range= Q3+(1.5 * IQR)
    return lower_range, upper_range
```

In [58]:

```
lr,ur=remove_outlier(df['QUANTITYORDERED'])
print('Lower Range :',lr,'\nUpper Range :',ur)
df['QUANTITYORDERED']=np.where(df['QUANTITYORDERED']>ur,ur,df['QUANTITYORDERED'])
df['QUANTITYORDERED']=np.where(df['QUANTITYORDERED']<lr,lr,df['QUANTITYORDERED'])
```

```
Lower Range : 3.0
Upper Range : 67.0
```

In [ ]:

```

```

In [59]:

```
lr,ur=remove_outlier(df['PRICEEACH'])
print('Lower Range :',lr,'\nUpper Range :',ur)
df['PRICEEACH']=np.where(df['PRICEEACH']>ur,ur,df['PRICEEACH'])
df['PRICEEACH']=np.where(df['PRICEEACH']<lr,lr,df['PRICEEACH'])
```

```
Lower Range : -18.78749999999998
Upper Range : 214.6325
```

In [ ]:

```

```

In [60]:

```
lr,ur=remove_outlier(df['SALES'])
print('Lower Range :',lr,'\nUpper Range :',ur)
df['SALES']=np.where(df['SALES']>ur,ur,df['SALES'])
df['SALES']=np.where(df['SALES']<lr,lr,df['SALES'])
```

```
Lower Range : -1243.7674999999995
Upper Range : 7951.212499999999
```

In [ ]:

```

```

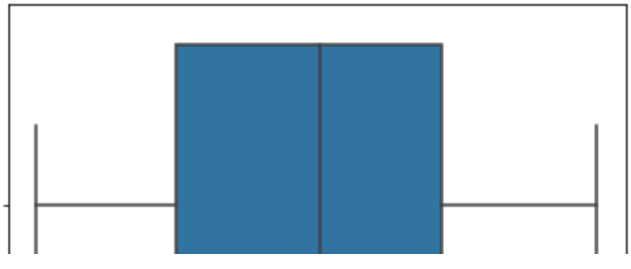
In [61]:

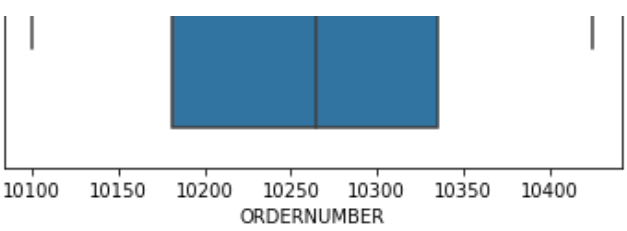
```
lr,ur=remove_outlier(df['MSRP'])
print('Lower Range :',lr,'\nUpper Range :',ur)
df['MSRP']=np.where(df['MSRP']>ur,ur,df['MSRP'])
df['MSRP']=np.where(df['MSRP']<lr,lr,df['MSRP'])
```

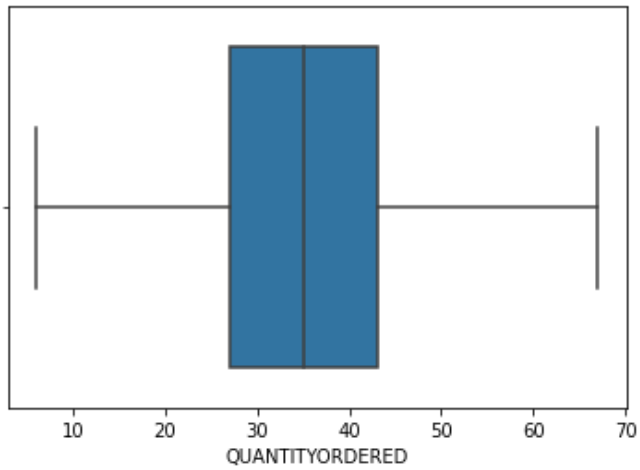
```
Lower Range : -16.0
Upper Range : 208.0
```

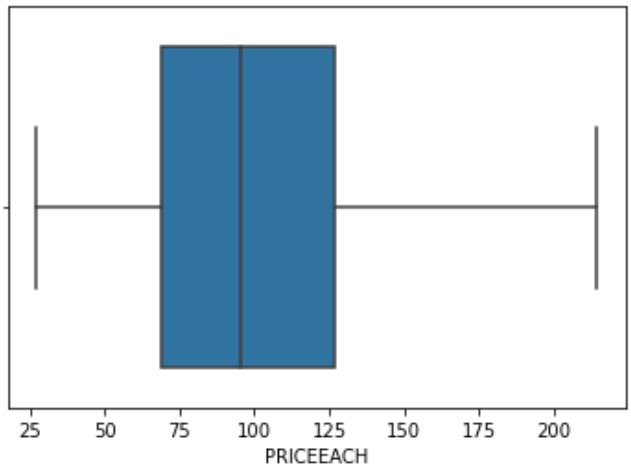
In [62]:

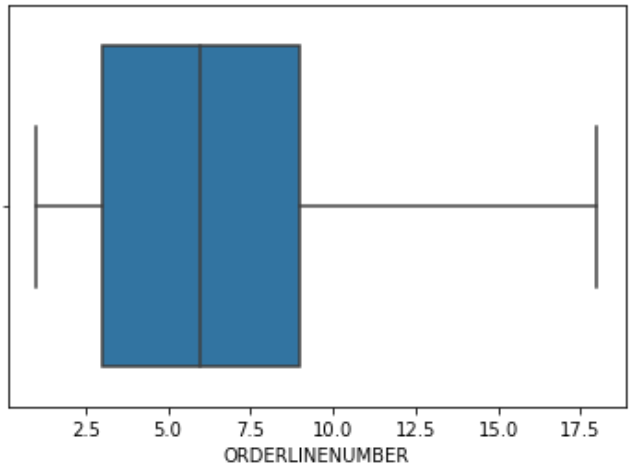
```
cols = ['ORDERNUMBER','QUANTITYORDERED','PRICEEACH','ORDERLINENUMBER','SALES','DAYS_SINC
E_LASTORDER','MSRP']
for i in cols:
    sns.boxplot(df[i])
    plt.show();
```

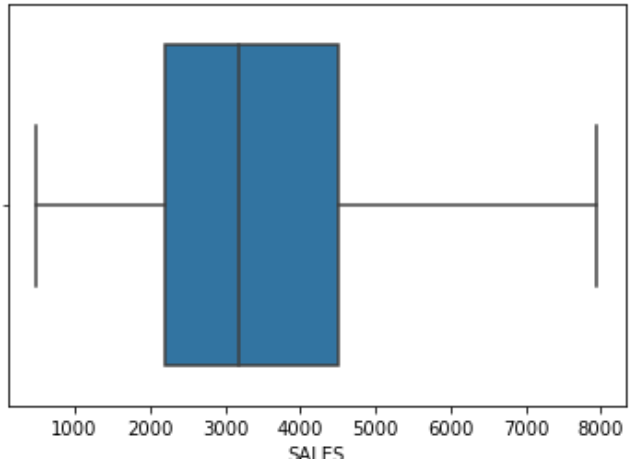


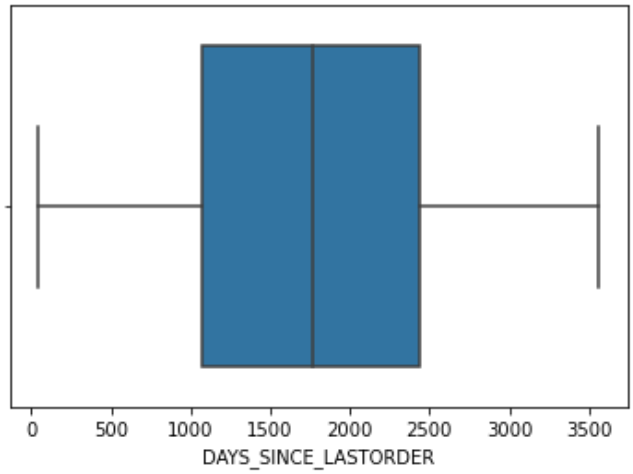

10100
10150
10200
10250
10300
10350
10400
ORDERNUMBER

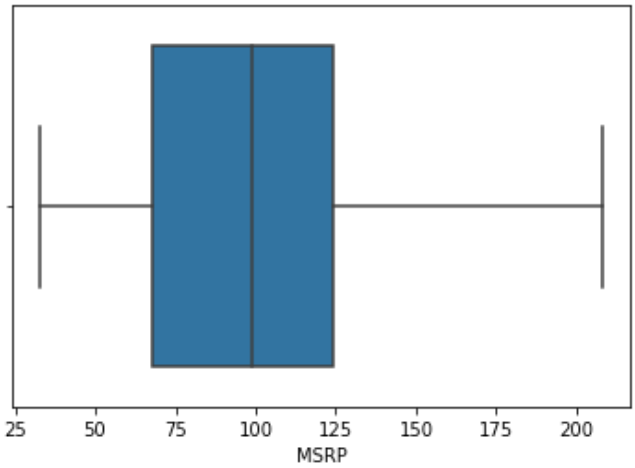

10
20
30
40
50
60
70
QUANTITYORDERED


25
50
75
100
125
150
175
200
PRICEEACH


2.5
5.0
7.5
10.0
12.5
15.0
17.5
ORDERLINENUMBER


1000
2000
3000
4000
5000
6000
7000
8000
SALES



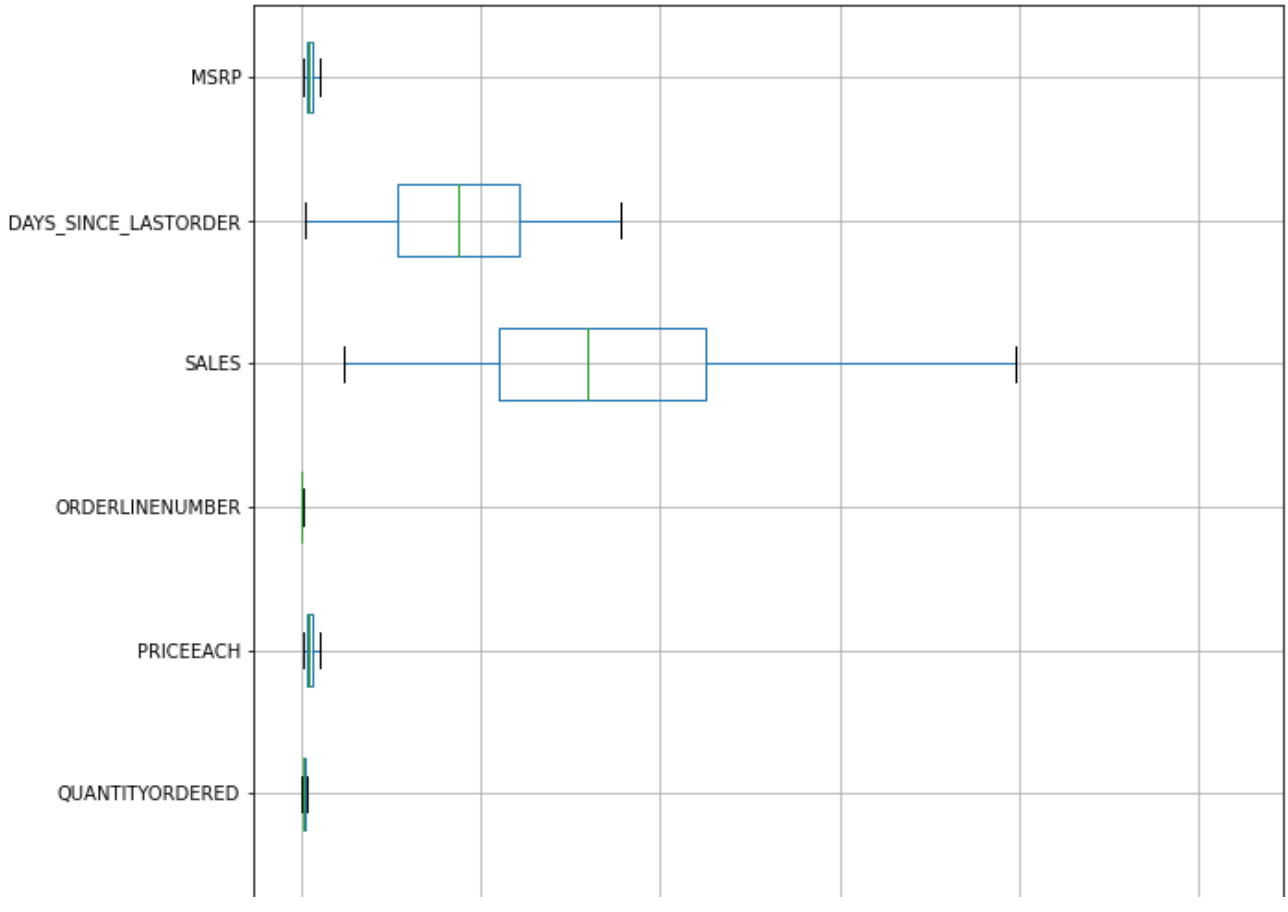


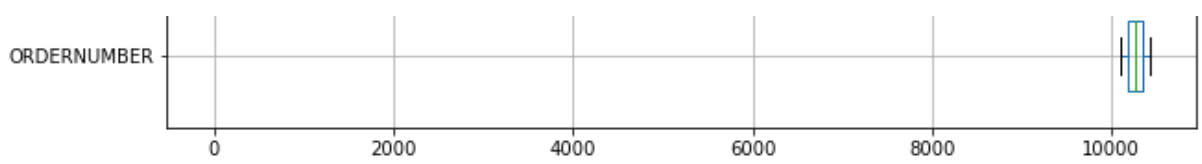
In [63]:

```
plt.figure(figsize=(10,10))
df[num].boxplot(vert=0)
```

Out[63]:

<AxesSubplot:>





In [ ]:

In [ ]: