wee9-Task

November 22, 2021

Name: Mohammad Aziz, Student ID: 20192233

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import datetime
```

[2]:
```
df= pd.read_csv("online_retail2.csv")
```

[3]:
```
df.head()
```

[3]:
```
   Invoice StockCode                          Description  Quantity  \
0  489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS        12
1  489434    79323P                   PINK CHERRY LIGHTS        12
2  489434    79323W                  WHITE CHERRY LIGHTS        12
3  489434     22041         RECORD FRAME 7" SINGLE SIZE         48
4  489434     21232       STRAWBERRY CERAMIC TRINKET BOX        24

           InvoiceDate  Price  Customer ID         Country
0  2009-12-01 07:45:00   6.95      13085.0  United Kingdom
1  2009-12-01 07:45:00   6.75      13085.0  United Kingdom
2  2009-12-01 07:45:00   6.75      13085.0  United Kingdom
3  2009-12-01 07:45:00   2.10      13085.0  United Kingdom
4  2009-12-01 07:45:00   1.25      13085.0  United Kingdom
```

[ ]:
```

```

1 1. Do we have missing data in this dataset?

if yes which columns and how many?

[ ]:
```

```

[4]:
```
count=2
for i in df.columns:
```

```
    if df.isnull().sum()[i]!=0:
        print("Yes! we have missing values in the column name '{}'\n where␣
↪there are missing values are {}.\n".format(df.columns[count], df.isnull().
↪sum()[i]))
        count=count+4
```

```
Yes! we have missing values in the column name 'Description'
 where there are missing values are 4382.

Yes! we have missing values in the column name 'Customer ID'
 where there are missing values are 243007.
```

2 2. Using value count, what are the top 5 countries

[6]:
```
C=df["Country"].value_counts().sort_values(ascending=False)
print("The Top 5 countries are\n")
C.head(5)
```

```
The Top 5 countries are
```

[6]:
```
United Kingdom    981330
EIRE               17866
Germany            17624
France             14330
Netherlands         5140
Name: Country, dtype: int64
```

3 3. Convert `InvoiceDate` to a datetime object

[7]:
```
df.dtypes
```

[7]:
```
Invoice         object
StockCode       object
Description     object
Quantity         int64
InvoiceDate     object
Price          float64
Customer ID    float64
Country         object
dtype: object
```

[8]:
```
# code here
```

[9]:
```
df["InvoiceDate"]=pd.to_datetime(df["InvoiceDate"])
```

[10]:
```
df.dtypes
```

[10]:
```
Invoice                 object
StockCode               object
Description             object
Quantity                 int64
InvoiceDate    datetime64[ns]
Price                  float64
Customer ID            float64
Country                 object
dtype: object
```

4 4. Create a new dataframe called `df2` which has `InvoiceDate` and `Price`

[11]:
```
df2=df[["InvoiceDate", "Price"]]
```

[12]:
```
df2.head()
```

[12]:

	InvoiceDate	Price
0	2009-12-01 07:45:00	6.95
1	2009-12-01 07:45:00	6.75
2	2009-12-01 07:45:00	6.75
3	2009-12-01 07:45:00	2.10
4	2009-12-01 07:45:00	1.25

5 5. Make `InvoiceDate` the index

[14]:
```
df2=df2.set_index("InvoiceDate")
```

[16]:
```
df2
```

[16]:

InvoiceDate	Price
2009-12-01 07:45:00	6.95
2009-12-01 07:45:00	6.75
2009-12-01 07:45:00	6.75
2009-12-01 07:45:00	2.10
2009-12-01 07:45:00	1.25
…	…
2011-12-09 12:50:00	2.10
2011-12-09 12:50:00	4.15
2011-12-09 12:50:00	4.15

```
2011-12-09 12:50:00   4.95
2011-12-09 12:50:00  18.00

[1067371 rows x 1 columns]
```

6 6 What is the start date and end date in this dataset

[65]:
```
## Start Date in the Date column
```

[63]:
```
df2.index.min()
```

[63]:
```
Timestamp('2009-12-01 07:45:00')
```

[66]:
```
## End Date in the Date column
```

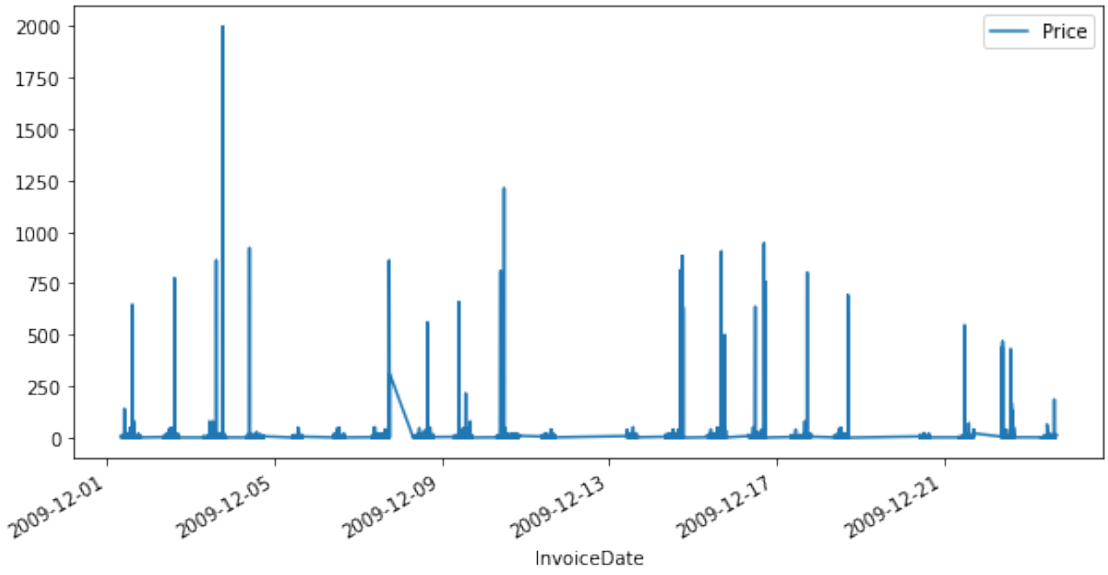
[64]:
```
df2.index.max()
```

[64]:
```
Timestamp('2011-12-09 12:50:00')
```

7 7. plot the timeseries for 2009, 2010, 2011

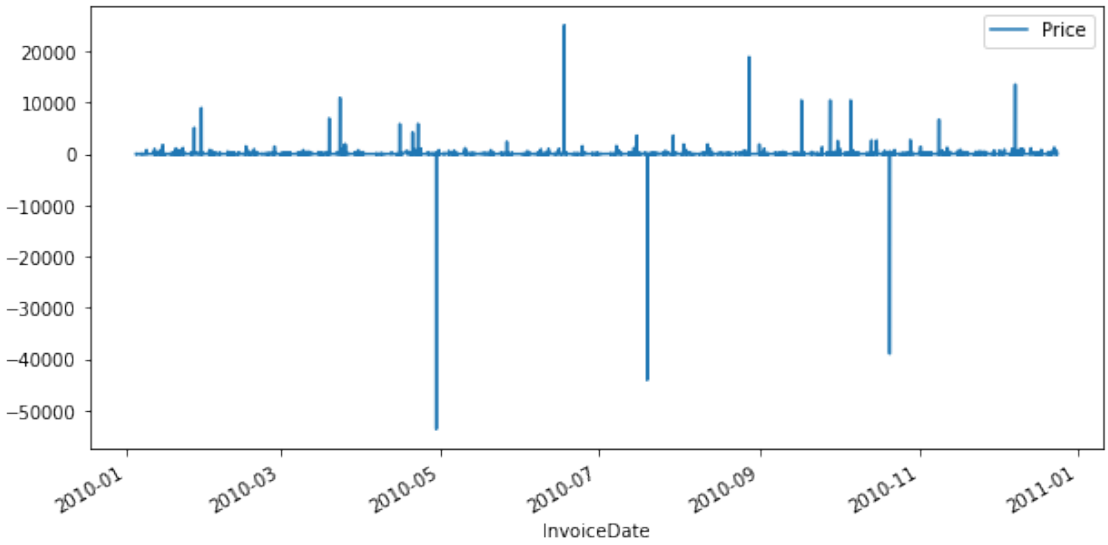
[49]:
```
##Plot timeseries for 2009
```

[43]:
```
df2[df2.index.year == 2009].plot(figsize=(10,5));
```



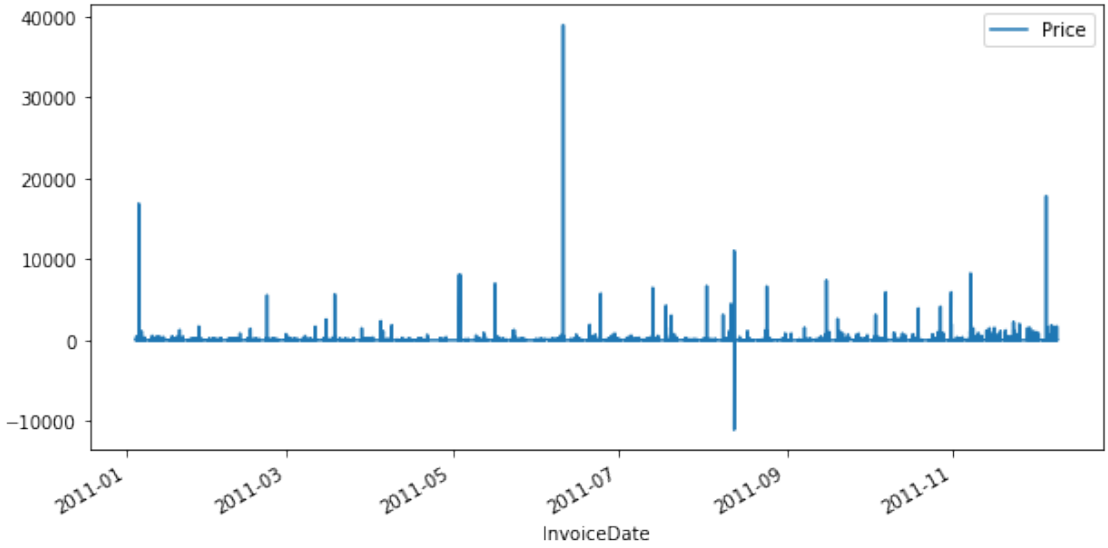
[50]:
```
##Plot timeseries for 2010
```

[51]:
```
df2[df2.index.year == 2010].plot(figsize=(10,5));
```



[52]:
```
##Plot timeseries for 2011
```

[53]:
```
df2[df2.index.year == 2011].plot(figsize=(10,5));
```



8 8. Plot 2010 - May

[60]:
```
## Plot timeseries for 2010-May
```

[59]:
```
df2.loc['2010-05'].plot(figsize=(10,5));
```