In [1]:

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

In [2]:

```
data = pd.read_csv('online_retail.csv')
```

In [3]:

```
data.head()
```

Out[3]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	01-12-2009 07:45	6.95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	01-12-2009 07:45	6.75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	01-12-2009 07:45	6.75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	01-12-2009 07:45	2.10	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	01-12-2009 07:45	1.25	13085.0	United Kingdom

In [4]:

```
1 data.tail()
```

Out[4]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
1048570	580501	23284	DOORMAT KEEP CALM AND COME IN	2	04-12-2011 13:00	8.25	14546.0	United Kingdom
1048571	580501	22507	MEMO BOARD RETROSPOT DESIGN	3	04-12-2011 13:00	4.95	14546.0	United Kingdom
1048572	580502	22469	HEART OF WICKER SMALL	3	04-12-2011 13:15	1.65	16931.0	United Kingdom
1048573	580502	23489	VINTAGE BELLS GARLAND	2	04-12-2011 13:15	2.89	16931.0	United Kingdom
1048574	580502	23046	PAPER LANTERN 9 POINT DELUXE STAR	1	04-12-2011 13:15	6.65	16931.0	United Kingdom

In [5]:

```
1 data.shape
```

Out[5]:

```
(1048575, 8)
```

In [6]:

```
1 data.columns
```

Out[6]:

```
Index(['Invoice', 'StockCode', 'Description', 'Quantity', 'InvoiceDate',
       'Price', 'Customer ID', 'Country'],
      dtype='object')
```

In [7]:

```
1 data.duplicated().sum()
```

Out[7]:

```
34150
```

In [8]:

```
1 data = data.drop_duplicates()
```

In [9]:

```
1 data.shape
```

Out[9]:

```
(1014425, 8)
```

In [10]:

```
1 data.isnull().sum()
```

Out[10]:

```
Invoice             0
StockCode           0
Description      4265
Quantity            0
InvoiceDate         0
Price               0
Customer ID    228826
Country             0
dtype: int64
```

In [11]:

```
1 data = data.drop('Description', axis = 1)
```

In [12]:

```
1 data.shape
```

Out[12]:

```
(1014425, 7)
```

In [13]:

```
1 data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate'])
```

In [14]:

```
1 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1014425 entries, 0 to 1048574
Data columns (total 7 columns):
 #   Column       Non-Null Count    Dtype
---  ------       --------------    -----
 0   Invoice      1014425 non-null  object
 1   StockCode    1014425 non-null  object
 2   Quantity     1014425 non-null  int64
 3   InvoiceDate  1014425 non-null  datetime64[ns]
 4   Price        1014425 non-null  float64
 5   Customer ID  785599 non-null   float64
 6   Country      1014425 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 61.9+ MB
```

In [15]:

```
1 data.describe()
```

Out[15]:

	Quantity	Price	Customer ID
count	1.014425e+06	1.014425e+06	785599.000000
mean	1.009725e+01	4.590115e+00	15313.078667
std	1.352799e+02	1.215813e+02	1695.992802
min	-7.421500e+04	-5.359436e+04	12346.000000
25%	1.000000e+00	1.250000e+00	13963.000000
50%	3.000000e+00	2.100000e+00	15235.000000
75%	1.000000e+01	4.150000e+00	16788.000000
max	7.421500e+04	3.897000e+04	18287.000000

In [20]:

```
1 data.nunique()
```

Out[20]:

```
Invoice        52961
StockCode       5304
Quantity        1048
InvoiceDate    47046
Price           2784
Customer ID     5924
Country           43
Year               3
dtype: int64
```

In [16]:

```
data['Year'] = pd.DatetimeIndex(data['InvoiceDate']).year
```

In [17]:

```
data.head()
```

Out[17]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	Customer ID	Country	Year
0	489434	85048	12	2009-01-12 07:45:00	6.95	13085.0	United Kingdom	2009
1	489434	79323P	12	2009-01-12 07:45:00	6.75	13085.0	United Kingdom	2009
2	489434	79323W	12	2009-01-12 07:45:00	6.75	13085.0	United Kingdom	2009
3	489434	22041	48	2009-01-12 07:45:00	2.10	13085.0	United Kingdom	2009
4	489434	21232	24	2009-01-12 07:45:00	1.25	13085.0	United Kingdom	2009

In [18]:

```
Sales = data.loc[data['Quantity'] > 0 & ~(data['Invoice'].str.contains('C'))]
```

In [22]:

```
Sales.shape
```

Out[22]:

```
(992181, 8)
```

In [23]:

```
1 Sales.head()
```

Out[23]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	Customer ID	Country	Year
0	489434	85048	12	2009-01-12 07:45:00	6.95	13085.0	United Kingdom	2009
1	489434	79323P	12	2009-01-12 07:45:00	6.75	13085.0	United Kingdom	2009
2	489434	79323W	12	2009-01-12 07:45:00	6.75	13085.0	United Kingdom	2009
3	489434	22041	48	2009-01-12 07:45:00	2.10	13085.0	United Kingdom	2009
4	489434	21232	24	2009-01-12 07:45:00	1.25	13085.0	United Kingdom	2009

In [19]:

```
1 Sales.sample(10)
```

Out[19]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	Customer ID	Country	Year
372720	525360	84029G	2	2010-05-10 11:33:00	3.75	15039.0	United Kingdom	2010
621097	544462	22894	2	2011-02-20 14:21:00	9.95	17050.0	United Kingdom	2011
1016087	578065	21929	1	2011-11-22 15:41:00	4.13	NaN	United Kingdom	2011
217439	510497	22179	2	2010-01-06 12:39:00	6.75	18223.0	United Kingdom	2010
212773	510001	72741	9	2010-05-26 15:06:00	1.45	14649.0	United Kingdom	2010
912437	570257	22730	6	2011-10-10 09:56:00	3.75	13767.0	United Kingdom	2011
872159	567197	22507	4	2011-09-19 10:10:00	4.95	14934.0	Channel Islands	2011
350343	523461	21508	12	2010-09-22 11:32:00	0.36	17850.0	United Kingdom	2010
396508	527393	22630	12	2010-10-17 13:30:00	1.95	NaN	EIRE	2010
395327	527363	84029G	1	2010-10-17 11:11:00	3.75	14810.0	United Kingdom	2010

In [24]:

```
Sales_New = Sales.copy()
```

In [25]:

```
Sales_New['Revenue'] = Sales_New['Quantity'] * Sales_New['Price']
```

In [26]:

```
Sales_New.head()
```

Out[26]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	Customer ID	Country	Year	Revenue
0	489434	85048	12	2009-01-12 07:45:00	6.95	13085.0	United Kingdom	2009	83.4
1	489434	79323P	12	2009-01-12 07:45:00	6.75	13085.0	United Kingdom	2009	81.0
2	489434	79323W	12	2009-01-12 07:45:00	6.75	13085.0	United Kingdom	2009	81.0
3	489434	22041	48	2009-01-12 07:45:00	2.10	13085.0	United Kingdom	2009	100.8
4	489434	21232	24	2009-01-12 07:45:00	1.25	13085.0	United Kingdom	2009	30.0

In [29]:

```
Sales_Mean = Sales_New.groupby('Year').mean()['Revenue']
```

In [30]:

```
Sales_Mean.head()
```

Out[30]:

```
Year
2009    18.856094
2010    20.115675
2011    19.894068
Name: Revenue, dtype: float64
```

In [31]:

```
Sales_Mean = Sales_Mean.reset_index()
```

In [32]:

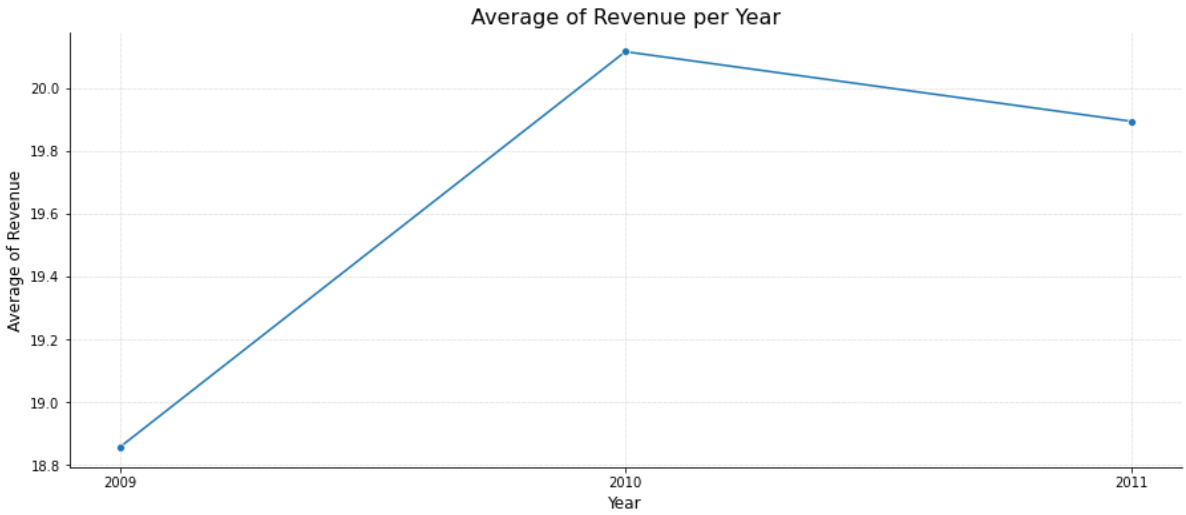
```
1 Sales_Mean.head()
```

Out[32]:

	Year	Revenue
0	2009	18.856094
1	2010	20.115675
2	2011	19.894068

In [34]:

```
1 plt.figure(figsize=(15,6))
2 sns.lineplot(Sales_Mean['Year'], Sales_Mean['Revenue'], marker='o')
3 plt.title('Average of Revenue per Year', fontsize = 16)
4 plt.xlabel('Year', fontsize = 12)
5 plt.ylabel('Average of Revenue', fontsize = 12)
6 plt.grid(color='darkgrey', linestyle=':', linewidth=0.5)
7 plt.gca().set_xticks([2009, 2010, 2011])
8 plt.gca().spines['top'].set_visible(False)
9 plt.gca().spines['right'].set_visible(False)
```



In [35]:

```
1 Sales_Finish = Sales_New[Sales_New['Customer ID'].notna()]
```

In [36]:

```
1 Sales_Finish.count()
```

Out[36]:

```
Invoice        767439
StockCode      767439
Quantity       767439
InvoiceDate    767439
Price          767439
Customer ID    767439
Country        767439
Year           767439
Revenue        767439
dtype: int64
```

In [37]:

```
1 Purchase_Canceled = data[data['Invoice'].str.contains('C')]
```

In [38]:

```
1 Purchase_Canceled.count()
```

Out[38]:

```
Invoice        18872
StockCode      18872
Quantity       18872
InvoiceDate    18872
Price          18872
Customer ID    18160
Country        18872
Year           18872
dtype: int64
```

In [39]:

```
1 Count_Finished = Sales_Finish.groupby('Year').count()['Invoice'].reset_index()
```

In [41]:

```
1 Count_Finished
```

Out[41]:

	Year	Invoice
0	2009	30279
1	2010	382156
2	2011	355004

In [42]:

```
Count_Canceled = Purchase_Canceled.groupby('Year').count()['Invoice'].reset_index()
```

In [43]:

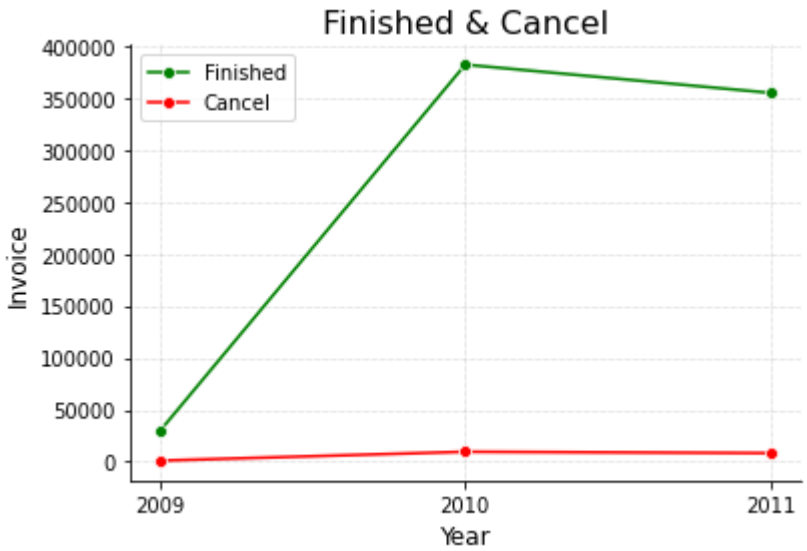
```
Count_Canceled
```

Out[43]:

	Year	Invoice
0	2009	1013
1	2010	9559
2	2011	8300

In [44]:

```
sns.lineplot(Count_Finished['Year'], Count_Finished['Invoice'],
             marker = 'o', color = 'green', label = 'Finished')
sns.lineplot(Count_Canceled['Year'], Count_Canceled['Invoice'],
             marker = 'o', color = 'red', label = 'Cancel')
plt.title('Finished & Cancel', fontsize = 16)
plt.xlabel('Year', fontsize = 12)
plt.ylabel('Invoice', fontsize = 12)
plt.grid(color='darkgrey', linestyle=':', linewidth=0.5)
plt.legend()
plt.gca().set_xticks([2009, 2010, 2011])
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```



In [45]:

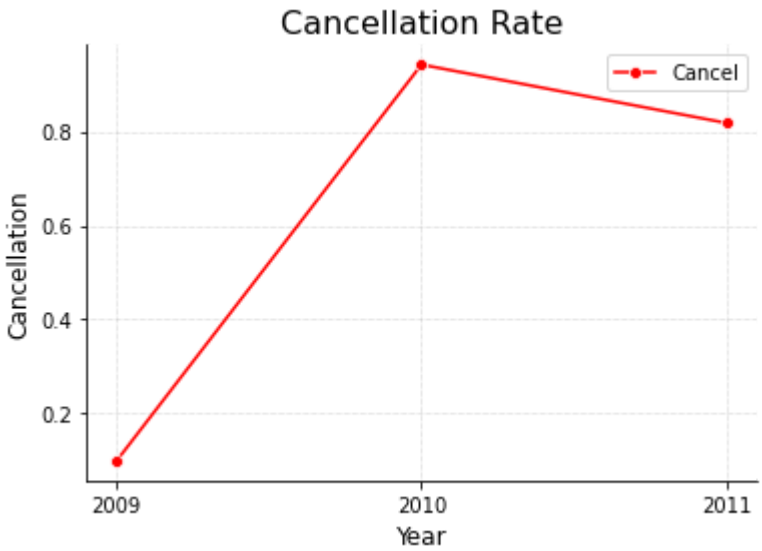
```
Cancellation_Rate = Count_Canceled['Invoice'] / data['Invoice'].count() * 100
Cancellation = Count_Canceled.assign(Cancellation_Rate=Count_Canceled['Invoice'] /
Cancellation[['Year', 'Cancellation_Rate']]
```

Out[45]:

	Year	Cancellation_Rate
0	2009	0.099860
1	2010	0.942307
2	2011	0.818198

In [47]:

```
sns.lineplot(Cancellation['Year'], Cancellation['Cancellation_Rate'],
             marker = 'o', color = 'red', label = 'Cancel')
plt.title('Cancellation Rate', fontsize = 16)
plt.xlabel('Year', fontsize = 12)
plt.ylabel('Cancellation', fontsize = 12)
plt.grid(color='darkgrey', linestyle=':', linewidth=0.5)
plt.legend()
plt.gca().set_xticks([2009, 2010, 2011])
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```



In [48]:

```
Comparison = Cancellation
Comparison['Total_Finished'] = Count_Finished['Invoice']
Comparison['Total_Canceled'] = Comparison['Invoice']
Comparison[['Year', 'Total_Finished', 'Total_Canceled', 'Cancellation_Rate']]
```

Out[48]:

	Year	Total_Finished	Total_Canceled	Cancellation_Rate
0	2009	30279	1013	0.099860
1	2010	382156	9559	0.942307
2	2011	355004	8300	0.818198