

Restaurant Sales Data Analysis

Here, We have the sales data of a restaurant company from different cities (countries). This data is available in CSV file format. We are going to analyze and visualize this data.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
ds = pd.read_csv('/content/drive/MyDrive/Datasets/Sales-Data-Analysis.csv')
```

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City
0	10452	07-11-2022	Fries	3.49	573.07	Online	Gift Card	Tom Jackson	London
1	10453	07-11-2022	Beverages	2.95	745.76	Online	Gift Card	Pablo Perez	Madrid
2	10454	07-11-2022	Sides & Other	4.99	200.40	In-store	Gift Card	Joao Silva	Lisbon
3	10455	08-11-2022	Burgers	12.99	569.67	In-store	Credit Card	Walter Muller	Berlin
4	10456	08-11-2022	Chicken Sandwiches	9.95	201.01	In-store	Credit Card	Walter Muller	Berlin

Next steps: [Generate code with ds](#) [New interactive sheet](#)

```
ds.info() # basic information about dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 254 entries, 0 to 253
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Order ID        254 non-null   int64
1   Date            254 non-null   object
2   Product         254 non-null   object
3   Price           254 non-null   float64
4   Quantity        254 non-null   float64
5   Purchase Type   254 non-null   object
6   Payment Method  254 non-null   object
7   Manager         254 non-null   object
8   City            254 non-null   object
dtypes: float64(2), int64(1), object(6)
memory usage: 18.0+ KB
```

```
ds.describe() # numerical stats of dataset
```

	Order ID	Price	Quantity
count	254.000000	254.000000	254.000000
mean	10584.133858	7.102323	460.611457
std	75.889181	4.341855	214.888699
min	10452.000000	2.950000	200.400000
25%	10520.250000	3.490000	201.010000
50%	10583.500000	4.990000	538.880000
75%	10649.750000	9.950000	677.440000
max	10713.000000	29.050000	754.430000

```
ds.Manager.unique() # found out that there are additional spaces in names
```

```
array(['Tom Jackson', 'Pablo Perez', 'Joao Silva',
      'Walter Muller', 'Remy Monet', 'Remy Monet',
      'Remy Monet', 'Remy Monet', 'Pablo Perez',
      'Pablo Perez', 'Pablo Perez', 'Joao Silva',
      'Tom Jackson'], dtype=object)
```

```
ds['Manager'] = ds['Manager'].str.strip().str.replace(r'\s+', ' ', regex=True) # to remove extra spaces in names
```

```
ds.Manager.unique()
```



```
array(['Tom Jackson', 'Pablo Perez', 'Joao Silva', 'Walter Muller',
      'Remy Monet'], dtype=object)
```

```
ds.Manager.nunique()
```

```
5
```

```
ds.head(10)
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City
0	10452	07-11-2022	Fries	3.49	573.07	Online	Gift Card	Tom Jackson	London
1	10453	07-11-2022	Beverages	2.95	745.76	Online	Gift Card	Pablo Perez	Madrid
2	10454	07-11-2022	Sides & Other	4.99	200.40	In-store	Gift Card	Joao Silva	Lisbon
3	10455	08-11-2022	Burgers	12.99	569.67	In-store	Credit Card	Walter Muller	Berlin
4	10456	08-11-2022	Chicken Sandwiches	9.95	201.01	In-store	Credit Card	Walter Muller	Berlin
5	10457	08-11-2022	Fries	3.49	573.07	In-store	Credit Card	Remy Monet	Paris
6	10459	08-11-2022	Sides & Other	4.99	200.40	In-store	Credit Card	Walter Muller	Berlin
7	10460	09-11-2022	Burgers	12.99	554.27	In-store	Credit Card	Remy Monet	Paris
8	10461	09-11-2022	Chicken Sandwiches	9.95	201.01	In-store	Credit Card	Remy Monet	Paris
9	10462	09-11-2022	Fries	3.49	573.07	In-store	Credit Card	Remy Monet	Paris

Next steps: [Generate code with ds](#) [New interactive sheet](#)

Converting the datatype of columns wherever needed

```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 254 entries, 0 to 253
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Order ID        254 non-null   int64
1   Date            254 non-null   object
2   Product         254 non-null   object
3   Price           254 non-null   float64
4   Quantity        254 non-null   float64
5   Purchase Type   254 non-null   object
6   Payment Method  254 non-null   object
7   Manager         254 non-null   object
8   City            254 non-null   object
dtypes: float64(2), int64(1), object(6)
memory usage: 18.0+ KB
```

```
ds.Quantity = ds.Quantity.round()
```

```
ds.Quantity = ds.Quantity.astype(int)
```

```
ds.Date = pd.to_datetime(ds.Date, format = 'mixed')
```

```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 254 entries, 0 to 253
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Order ID        254 non-null   int64
1   Date            254 non-null   datetime64[ns]
2   Product         254 non-null   object
3   Price           254 non-null   float64
4   Quantity        254 non-null   int64
5   Purchase Type   254 non-null   object
6   Payment Method  254 non-null   object
7   Manager         254 non-null   object
8   City            254 non-null   object
dtypes: datetime64[ns](1), float64(1), int64(2), object(5)
memory usage: 18.0+ KB
```

Now, We will Analyze the Data, as per the questions asked;

Q.1) Most Preferred Payment Method ?

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon
3	10455	2022-08-11	Burgers	12.99	570	In-store	Credit Card	Walter Muller	Berlin
4	10456	2022-08-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Walter Muller	Berlin

Next steps:

[Generate code with ds](#)[New interactive sheet](#)

```
ds['Payment Method'].unique()
```

```
array([' Gift Card', ' Credit Card', ' Cash'], dtype=object)
```

```
ds['Payment Method'].value_counts()
```

	count
Payment Method	
Credit Card	120
Cash	76
Gift Card	58

```
dtype: int64
```

```
ds['Payment Method'].value_counts(normalize=True)*100 # To get the result in percentage we using normalize here
```

	proportion
Payment Method	
Credit Card	47.244094
Cash	29.921260
Gift Card	22.834646

```
dtype: float64
```

```
# Plotting the graph for same
```

```
ds['Payment Method'].value_counts().plot(kind='bar')
```

<Axes: xlabel='Payment Method'>

We can say that, Credit Card payment method is the most preferred payment method used.

Q.2) Most Selling Product ?

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon
3	10455	2022-08-11	Burgers	12.99	570	In-store	Credit Card	Walter Muller	Berlin
4	10456	2022-08-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Walter Muller	Berlin

Next steps: [Generate code with ds](#) [New interactive sheet](#)

```
ds['Product'].value_counts()
```

Product	count
Chicken Sandwiches	52
Burgers	52
Fries	51
Beverages	50
Sides & Other	49

dtype: int64

By Quantity

```
ds.groupby('Product')['Quantity'].sum()
```

Product	Quantity
Beverages	34988
Burgers	29018
Chicken Sandwiches	11133
Fries	32023
Sides & Other	9800

dtype: int64

```
ds.groupby('Product')['Quantity'].sum().sort_values(ascending=False)
```

Product	Quantity
Beverages	34988
Fries	32023
Burgers	29018
Chicken Sandwiches	11133
Sides & Other	9800

dtype: int64

```
most_quantity = ds.groupby('Product')['Quantity'].sum().sort_values(ascending=False)
most_quantity
```

Quantity	
Product	
Beverages	34988
Fries	32023
Burgers	29018
Chicken Sandwiches	11133
Sides & Other	9800

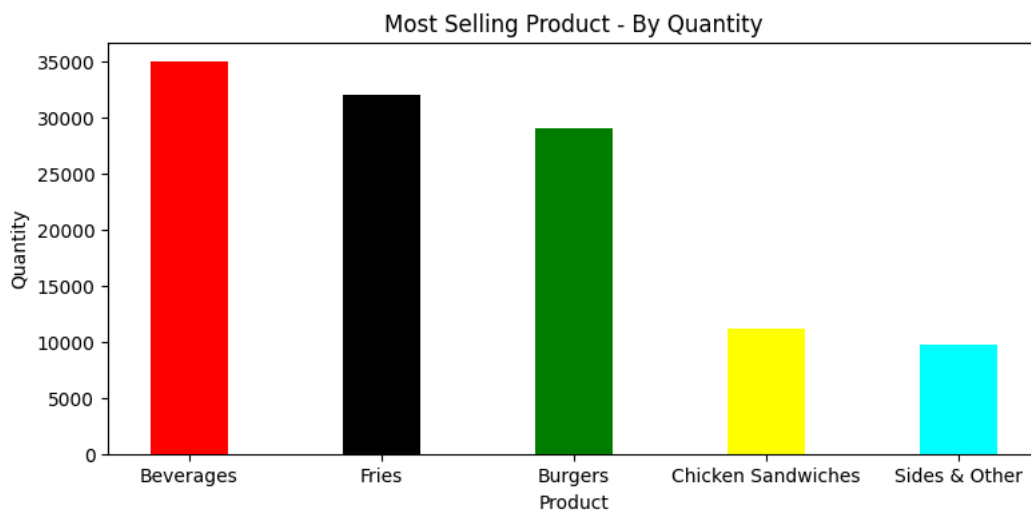
dtype: int64

```
most_quantity = most_quantity.reset_index()    # To convert the index of a Series into a column to form a DataFrame
most_quantity
```

	Product	Quantity
0	Beverages	34988
1	Fries	32023
2	Burgers	29018
3	Chicken Sandwiches	11133
4	Sides & Other	9800

Next steps: [Generate code with most_quantity](#) [New interactive sheet](#)

```
plt.figure(figsize = (9,4))
plt.bar(most_quantity['Product'], most_quantity['Quantity'], color = ['red', 'black', 'green', 'yellow', 'cyan'], width=0.4)
plt.title("Most Selling Product - By Quantity")
plt.xlabel("Product")
plt.ylabel("Quantity");
```



By Revenue

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon
3	10455	2022-08-11	Burgers	12.99	570	In-store	Credit Card	Walter Muller	Berlin
4	10456	2022-08-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Walter Muller	Berlin

Next steps: [Generate code with ds](#) [New interactive sheet](#)

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

```
ds.groupby('Product')['Revenue'].sum().sort_values(ascending = False)
```

Revenue	
Product	
Burgers	376943.82
Fries	125626.57
Chicken Sandwiches	114612.45
Beverages	103214.60
Sides & Other	48902.00

dtype: float64

```
most_revenue = ds.groupby('Product')['Revenue'].sum().sort_values(ascending = False)
most_revenue
```

Revenue	
Product	
Burgers	376943.82
Fries	125626.57
Chicken Sandwiches	114612.45
Beverages	103214.60
Sides & Other	48902.00

dtype: float64

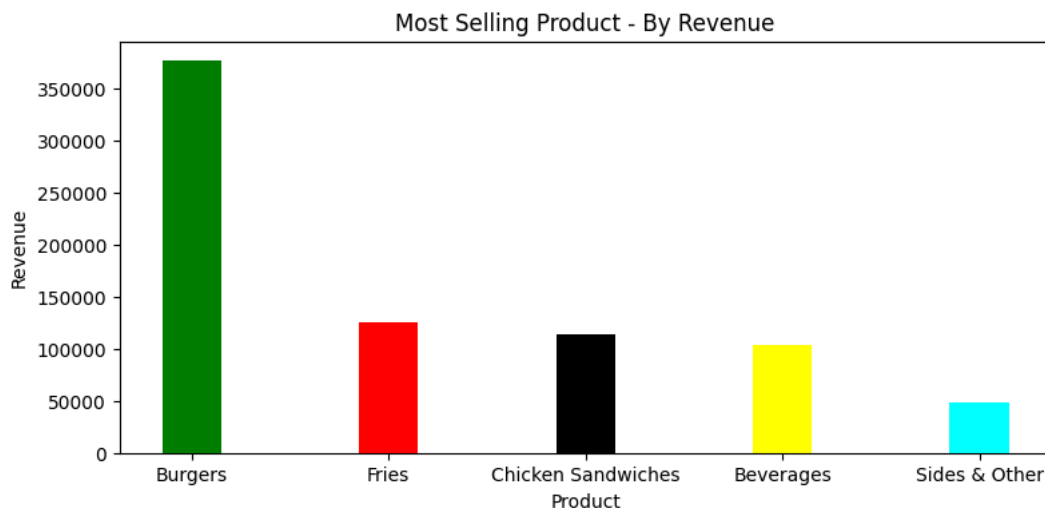
```
most_revenue = most_revenue.reset_index() # To convert the index of a Series into a column to form a DataFrame
most_revenue
```

	Product	Revenue
0	Burgers	376943.82
1	Fries	125626.57
2	Chicken Sandwiches	114612.45
3	Beverages	103214.60
4	Sides & Other	48902.00



Next steps: [Generate code with most_revenue](#) [New interactive sheet](#)

```
plt.figure(figsize=(9, 4))
plt.bar(most_revenue['Product'], most_revenue['Revenue'], color = ['green', 'red', 'black', 'yellow', 'cyan'], width = 0.3);
plt.title("Most Selling Product - By Revenue") # setting the title
plt.xlabel("Product") # setting the x-axis label
plt.ylabel("Revenue");
```



Q.3) Which city had maximum revenue

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London	1999.77
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid	2200.70
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon	998.00
3	10455	2022-08-11	Burgers	12.99	570	In-store	Credit Card	Walter Muller	Berlin	7404.30
4	10456	2022-08-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Walter Muller	Berlin	1999.95

Next steps:

[Generate code with ds](#)
[New interactive sheet](#)

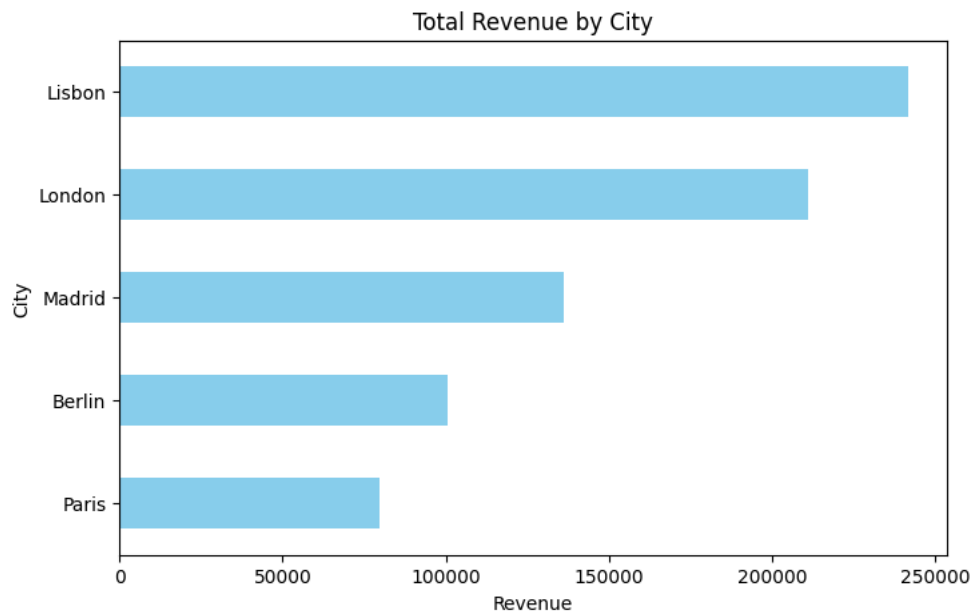
```
ds['City'].unique()
```

```
array(['London', 'Madrid', 'Lisbon', 'Berlin', 'Paris'], dtype=object)
```

```
city_revenue = ds.groupby('City')['Revenue'].sum().sort_values(ascending = True)
```

```
city_revenue.plot(kind='barh', figsize=(8,5), color='skyblue')
```

```
plt.title("Total Revenue by City")
plt.xlabel("Revenue")
plt.ylabel("City")
plt.show()
```



Q.4) Date wise revenue

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London	1999.77
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid	2200.70
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon	998.00
3	10455	2022-08-11	Burgers	12.99	570	In-store	Credit Card	Walter Muller	Berlin	7404.30
4	10456	2022-08-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Walter Muller	Berlin	1999.95

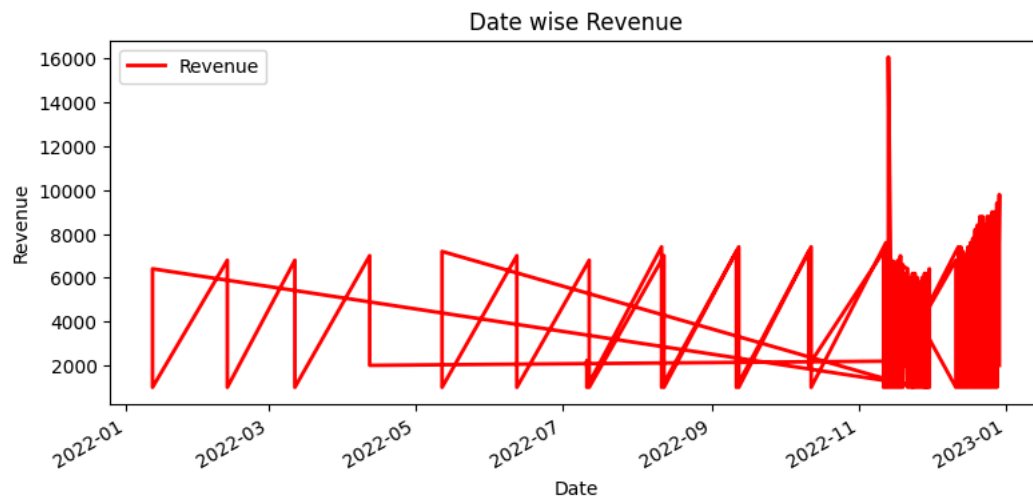
Next steps:

[Generate code with ds](#)
[New interactive sheet](#)

```
# plotting graph to visualize the date wise revenue
```

```
ds.plot('Date', 'Revenue', color='red', linewidth=2, figsize=(9,4))
plt.title("Date wise Revenue")
plt.xlabel("Date")
```

```
plt.ylabel("Revenue")
plt.show()
```



Q.5) Average Revenue

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London	1999.77
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid	2200.70
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon	998.00
3	10455	2022-08-11	Burgers	12.99	570	In-store	Credit Card	Walter Muller	Berlin	7404.30
4	10456	2022-08-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Walter Muller	Berlin	1999.95

Next steps: [Generate code with ds](#) [New interactive sheet](#)

```
ds['Revenue'].mean()
```

```
np.float64(3028.737952755905)
```

Q.6) Average Revenue of November & December month

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London	1999.77
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid	2200.70
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon	998.00
3	10455	2022-08-11	Burgers	12.99	570	In-store	Credit Card	Walter Muller	Berlin	7404.30
4	10456	2022-08-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Walter Muller	Berlin	1999.95

Next steps: [Generate code with ds](#) [New interactive sheet](#)

```
ds['Month'] = ds['Date'].dt.month
```

```
m11 = ds[ds['Month'] == 11]
```

```
m11.Revenue.mean()
```

```
np.float64(2969.3819587628864)
```

```
m12 = ds[ds['Month'] == 12]
```

```
m12.Revenue.mean()
```



```
np.float64(3194.7782978723403)
```

Q.7) Standard Deviation of Revenue and Quantity ?

ds.head()

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue	Month
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London	1999.77	7
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid	2200.70	7
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon	998.00	7

Next steps:

Generate code with ds

New interactive sheet

```
ds['Revenue'].std()
```

```
2419.9325462715055
```

```
ds['Quantity'].std()
```

```
214.97943096363838
```

Q.8) Variance of Revenue and Quantity ?

ds.head()

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue	Month
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London	1999.77	7
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid	2200.70	7
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon	998.00	7

Next steps:

Generate code with ds

New interactive sheet

```
ds['Quantity'].var()
```

```
46216.155737449764
```

```
ds['Revenue'].var()
```

```
5856073.528504092
```

Q.9) Is revenue increasing or decreasing over time?

ds.head()

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue	Month
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London	1999.77	7
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid	2200.70	7
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon	998.00	7

Next steps:

Generate code with ds

New interactive sheet

```
m11 = ds[ds['Month'] == 11]
m11
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue	Month	
16	10470	2022-11-11	Burgers	12.99	554	In-store	Credit Card	Pablo Perez	Madrid	7196.46	11	
17	10471	2022-11-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Pablo Perez	Madrid	1999.95	11	
18	10472	2022-11-11	Fries	3.49	630	In-store	Credit Card	Pablo Perez	Madrid	2198.70	11	
19	10473	2022-11-11	Beverages	2.95	678	In-store	Credit Card	Pablo Perez	Madrid	2000.10	11	
20	10474	2022-11-11	Sides & Other	4.99	200	In-store	Credit Card	Pablo Perez	Madrid	998.00	11	
...	
160	10620	2022-11-12	Burgers	12.99	585	Online	Gift Card	Tom Jackson	London	7599.15	11	
161	10621	2022-11-12	Chicken Sandwiches	9.95	201	Online	Gift Card	Tom Jackson	London	1999.95	11	

Next steps: [Generate code with m11](#) [New interactive sheet](#)

```
m11['Revenue'].sum()

np.float64(288030.05)
```

```
m12 = ds[ds.Month == 12]
m12
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue	Month	
21	10475	2022-12-11	Burgers	12.99	523	In-store	Credit Card	Pablo Perez	Madrid	6793.77	12	
22	10476	2022-12-11	Chicken Sandwiches	9.95	201	In-store	Credit Card	Pablo Perez	Madrid	1999.95	12	
23	10477	2022-12-11	Fries	3.49	630	In-store	Credit Card	Pablo Perez	Madrid	2198.70	12	
24	10478	2022-12-11	Beverages	2.95	678	In-store	Credit Card	Pablo Perez	Madrid	2000.10	12	
25	10479	2022-12-11	Sides & Other	4.99	200	In-store	Credit Card	Pablo Perez	Madrid	998.00	12	
...	
249	10709	2022-12-28	Sides & Other	4.99	200	Drive-thru	Gift Card	Walter Muller	Berlin	998.00	12	
250	10710	2022-12-29	Burgers	12.99	754	Drive-thru	Gift Card	Walter Muller	Berlin	9794.46	12	

Next steps: [Generate code with m12](#) [New interactive sheet](#)

```
m12['Revenue'].sum()

np.float64(300309.16)
```

Yes, The revenue is increasing



Q.10) Average 'Quantity Sold' & 'Average Revenue' for each product ?

```
ds.head()
```

	Order ID	Date	Product	Price	Quantity	Purchase Type	Payment Method	Manager	City	Revenue	Month	
0	10452	2022-07-11	Fries	3.49	573	Online	Gift Card	Tom Jackson	London	1999.77	7	
1	10453	2022-07-11	Beverages	2.95	746	Online	Gift Card	Pablo Perez	Madrid	2200.70	7	
2	10454	2022-07-11	Sides & Other	4.99	200	In-store	Gift Card	Joao Silva	Lisbon	998.00	7	

Next steps: [Generate code with ds](#) [New interactive sheet](#)

```
ds.groupby('Product')[['Quantity', 'Revenue']].agg({'Quantity':'mean', 'Revenue':'mean'})
```

	Quantity	Revenue	
Product			
Beverages	699.760000	2064.292000	
Burgers	558.038462	7248.919615	
Chicken Sandwiches	214.096154	2204.085577	
Fries	627.901961	2463.266078	
Sides & Other	200.000000	998.000000	