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

```
df = pd.read_csv ('/content/Amazon Sales data.csv')
```

```
df.sample()
```

1 entry Filter

index	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
88	Middle East and North Africa	Kuwait	Fruits	Online	M	4/30/2012	513417565	5/18/2012	522	9.33	6.92	4870.26	3612.24	1258.02

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

```
df.columns
```

```
Index(['Region', 'Country', 'Item Type', 'Sales Channel', 'Order Priority',
       'Order Date', 'Order ID', 'Ship Date', 'Units Sold', 'Unit Price',
       'Unit Cost', 'Total Revenue', 'Total Cost', 'Total Profit'],
      dtype='object')
```

```
df.shape
```

```
(100, 14)
```

```
df.size
```

```
1400
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Region          100 non-null    object 
 1   Country         100 non-null    object 
 2   Item Type       100 non-null    object 
 3   Sales Channel   100 non-null    object 
 4   Order Priority  100 non-null    object 
 5   Order Date      100 non-null    object 
 6   Order ID        100 non-null    int64  
 7   Ship Date       100 non-null    object 
 8   Units Sold      100 non-null    int64  
 9   Unit Price      100 non-null    float64
 10  Unit Cost       100 non-null    float64
 11  Total Revenue   100 non-null    float64
 12  Total Cost      100 non-null    float64
 13  Total Profit    100 non-null    float64
dtypes: float64(5), int64(2), object(7)
memory usage: 11.1+ KB
```

```
df.describe()
```

1 to 8 of 8 entries Filter

index	Order ID	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	100.0	100.0	100.0	100.0	100.0	100.0	100.0
mean	555020412.36	5128.71	276.7613	191.048	1373487.6831	931805.6991000001	441681.98399999994
std	260615257.13142592	2794.4845616956904	235.59224058433128	188.20818124855495	1460028.7068235006	1083938.2521883622	438537.90705963754
min	114606559.0	124.0	9.33	6.92	4870.26	3612.24	1258.02
25%	338922488.0	2836.25	81.73	35.84	268721.2125	168868.0275	121443.58499999999
50%	557708561.0	5382.5	179.88	107.275	752314.36	363566.385	290767.995
75%	790755080.75	7369.0	437.2	263.33	2212044.6825	1613869.7175	635828.8
max	994022214.0	9925.0	668.27	524.96	5997054.98	4509793.96	1719922.04

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

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

	0
Region	0
Country	0
Item Type	0
Sales Channel	0
Order Priority	0
Order Date	0
Order ID	0
Ship Date	0
Units Sold	0
Unit Price	0
Unit Cost	0
Total Revenue	0
Total Cost	0
Total Profit	0

dtype: int64

```
df.dtypes
```

	0
Region	object
Country	object
Item Type	object
Sales Channel	object
Order Priority	object
Order Date	object
Order ID	int64
Ship Date	object
Units Sold	int64
Unit Price	float64
Unit Cost	float64
Total Revenue	float64
Total Cost	float64
Total Profit	float64

dtype: object

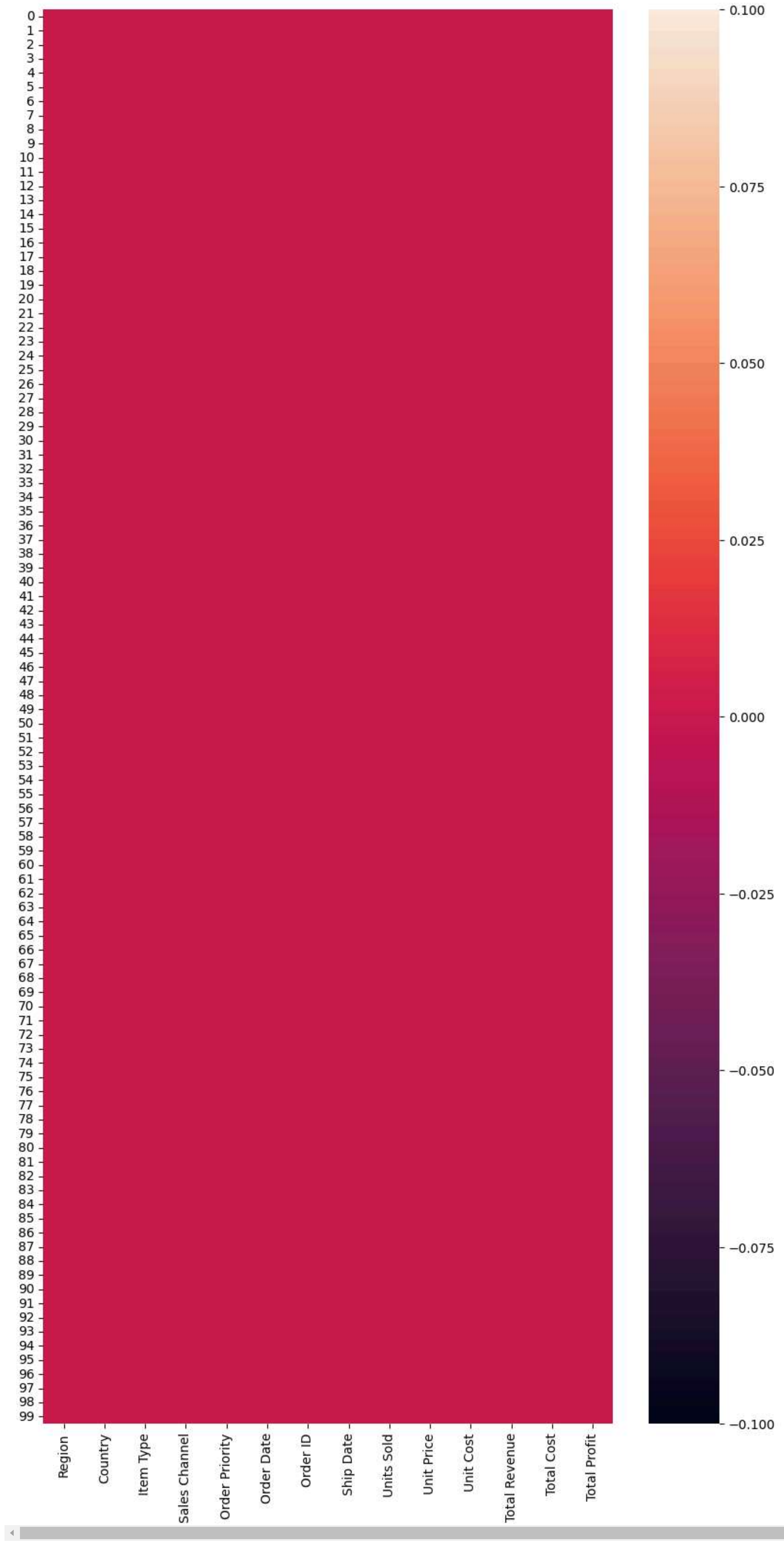
```
df = df.astype ({'Ship Date': 'datetime64[ns]', 'Order Date': 'datetime64[ns]'})
df.dtypes
```

	0
Region	object
Country	object
Item Type	object
Sales Channel	object
Order Priority	object
Order Date	datetime64[ns]
Order ID	int64
Ship Date	datetime64[ns]
Units Sold	int64
Unit Price	float64
Unit Cost	float64
Total Revenue	float64
Total Cost	float64
Total Profit	float64

dtype: object

```
plt.figure (figsize = (10, 20))
sns.heatmap (df.isnull())
```

```
<Axes: >
```

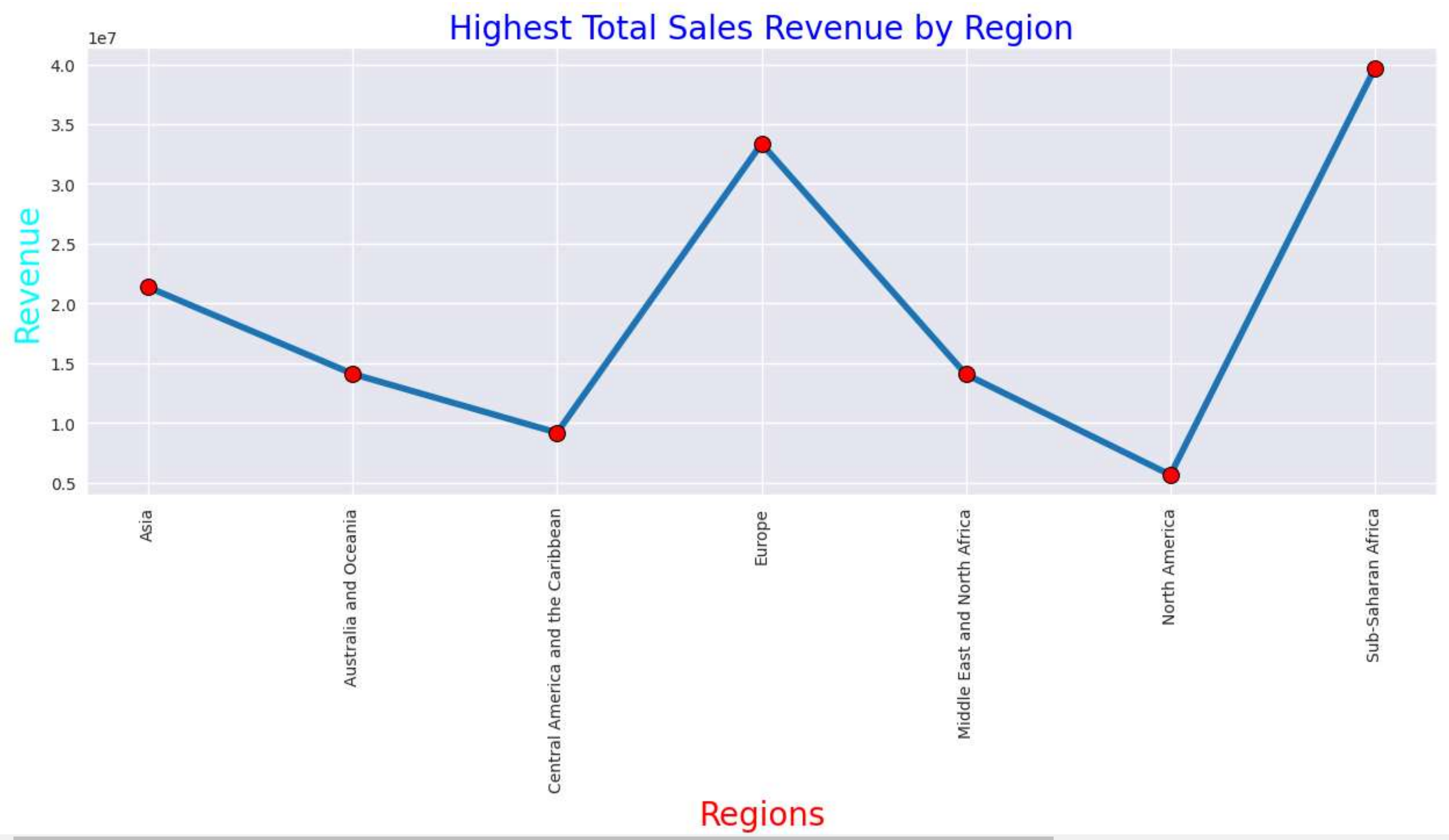


1. Which region has the highest total sales revenue?

```
HTR = df.groupby (df['Region']) ['Total Revenue'].sum()
HTR.idxmax()
```

```
'Sub-Saharan Africa'
```

```
# Grouping data and summing 'Total Revenue' by 'Region'
group_data = df.groupby('Region')['Total Revenue'].sum().reset_index()

# Setting Seaborn style
sns.set_style('darkgrid')

# Creating the plot
plt.figure(figsize=(15, 5))
sns.lineplot(
    x='Region',
    y='Total Revenue',
    data=group_data,
    linestyle='-',  # Use a valid linestyle
    linewidth=4,
    marker='o',
    markersize=10,
    markerfacecolor='red',
    markeredgecolor='black'
)

# Setting plot labels, title, and styling
plt.xticks(rotation=90)
plt.title('Highest Total Sales Revenue by Region', fontsize=20, color='blue')
plt.xlabel('Regions', fontsize=20, color='red')
plt.ylabel('Revenue', fontsize=20, color='cyan')

# Displaying the plot
plt.show()

```



2, What is the average unit price & unit cost for each item type?

```
# Calculating the average unit price and unit cost for each item type
avg_unit_price = df.groupby('Item Type')['Unit Price'].mean()
avg_unit_cost = df.groupby('Item Type')['Unit Cost'].mean()

# Combining the results into a single DataFrame
avg_price_cost = pd.DataFrame({
    'Average Unit Price': avg_unit_price,
    'Average Unit Cost': avg_unit_cost
})

# Resetting the index (optional)
avg_price_cost.reset_index(inplace=True)

# Displaying the DataFrame
avg_price_cost

```

1 to 12 of 12 entries Filter

index	Item Type	Average Unit Price	Average Unit Cost
0	Baby Food	255.28	159.42
1	Beverages	47.45	31.79
2	Cereal	205.7	117.11
3	Clothes	109.28	35.84
4	Cosmetics	437.19999999999993	263.33
5	Fruits	9.33	6.92
6	Household	668.27	502.5400000000001
7	Meat	421.89	364.69
8	Office Supplies	651.21	524.96
9	Personal Care	81.73	56.67
10	Snacks	152.58	97.44
11	Vegetables	154.06	90.93

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots New interactive sheet

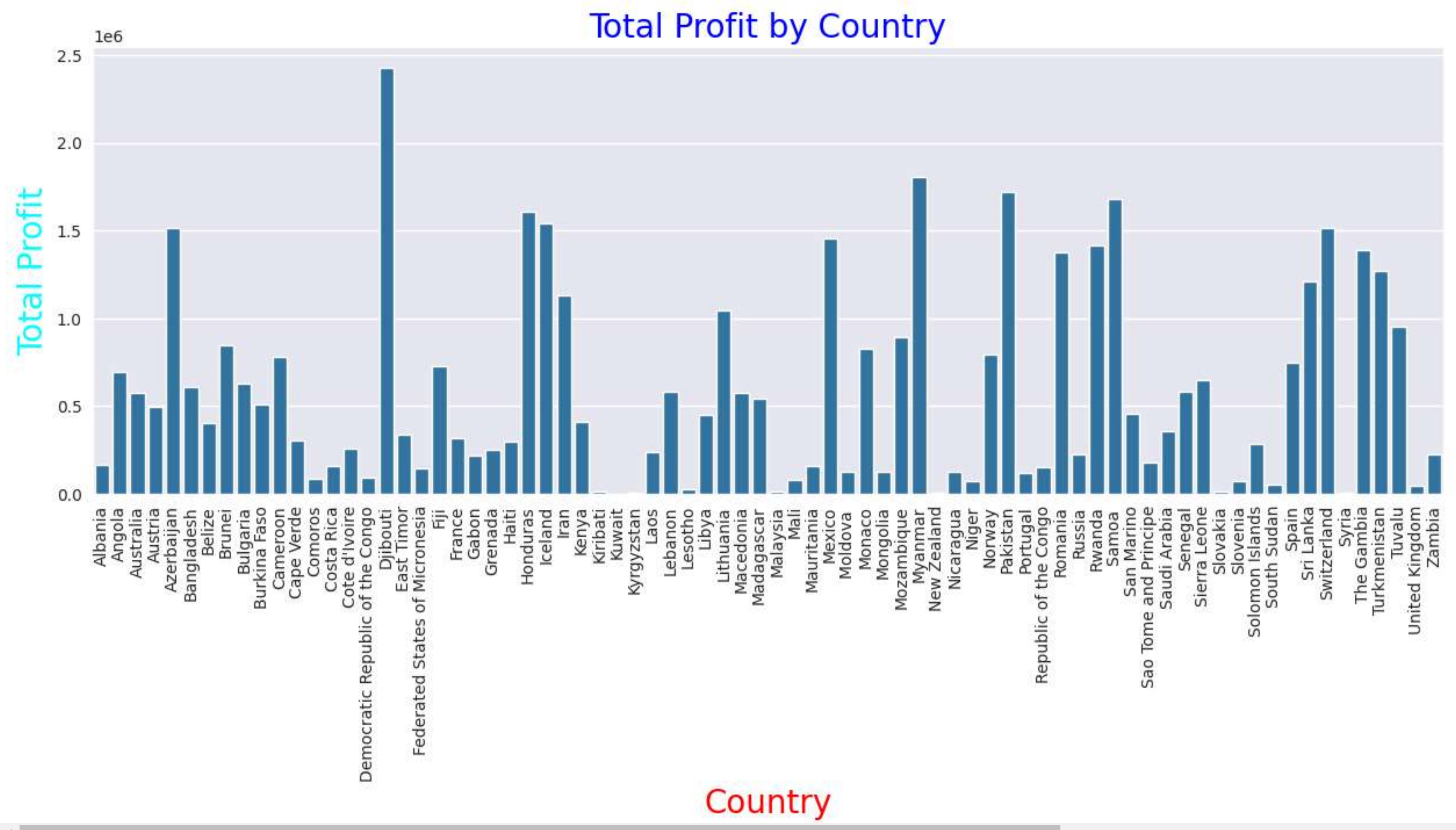
3. Which country has the highest total profit?

```
# Grouping and summing Total Profit by Country
total_profit = df.groupby(['Country'])['Total Profit'].sum()

# Finding the country with the highest total profit
highest_total_profit = total_profit.idxmax()

# Corrected print statement
print (f"The Country with the highest Total Profit is {highest_total_profit}")
```

```
The Country with the highest Total Profit is Djibouti
```

```
# Grouping data and summing 'Total Profit' by 'Country'
group_data = df.groupby('Country')['Total Profit'].sum().reset_index()

# Setting Seaborn style
sns.set_style('darkgrid')

# Creating the plot
plt.figure(figsize=(15, 5))
sns.barplot(
    x='Country',
    y='Total Profit',
    data=group_data  # Pass the grouped data to the plot
)

```

```
# Setting plot labels, title, and styling
plt.xticks(rotation=90)
plt.title('Total Profit by Country', fontsize=20, color='blue')
plt.xlabel('Country', fontsize=20, color='red')
plt.ylabel('Total Profit', fontsize=20, color='cyan')

# Displaying the plot
plt.show()
```



4. How does the sales channel affect the order priority distribution ?

```
sales_channel = df.groupby(['Sales Channel']) ['Order Priority'].value_counts().reset_index()
sales = pd.DataFrame (sales_channel)
sales
```

1 to 8 of 8 entries Filter

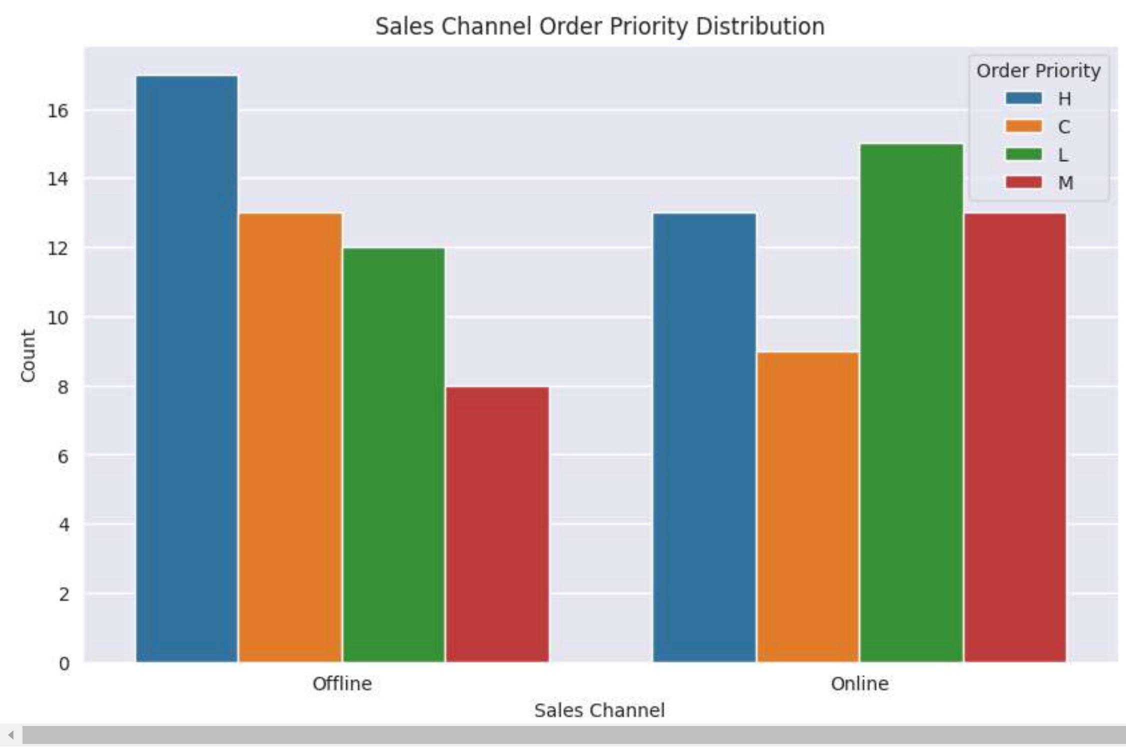
index	Sales Channel	Order Priority	count
0	Offline	H	17
1	Offline	C	13
2	Offline	L	12
3	Offline	M	8
4	Online	L	15
5	Online	H	13
6	Online	M	13
7	Online	C	9

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots New interactive sheet

```
sales.reset_index()

sns.set_style ('darkgrid')

plt.figure (figsize = (10, 6))
sns.barplot (
    x = 'Sales Channel',
    y = 'count',
    hue = 'Order Priority',
    data = sales
)

plt.xlabel ('Sales Channel')
plt.ylabel ('Count')
plt.title ('Sales Channel Order Priority Distribution')

plt.show()
```



Double-click (or enter) to edit

5. What is the average order processing time for each sales channel?

*Order Processing Time: duration b/w order & ship dates

```
df['Processing Time'] = df['Ship Date'] - df ['Order Date']

apt = df.groupby (['Sales Channel']) ['Processing Time'].mean().reset_index()
apt
```

1 to 2 of 2 entries Filter

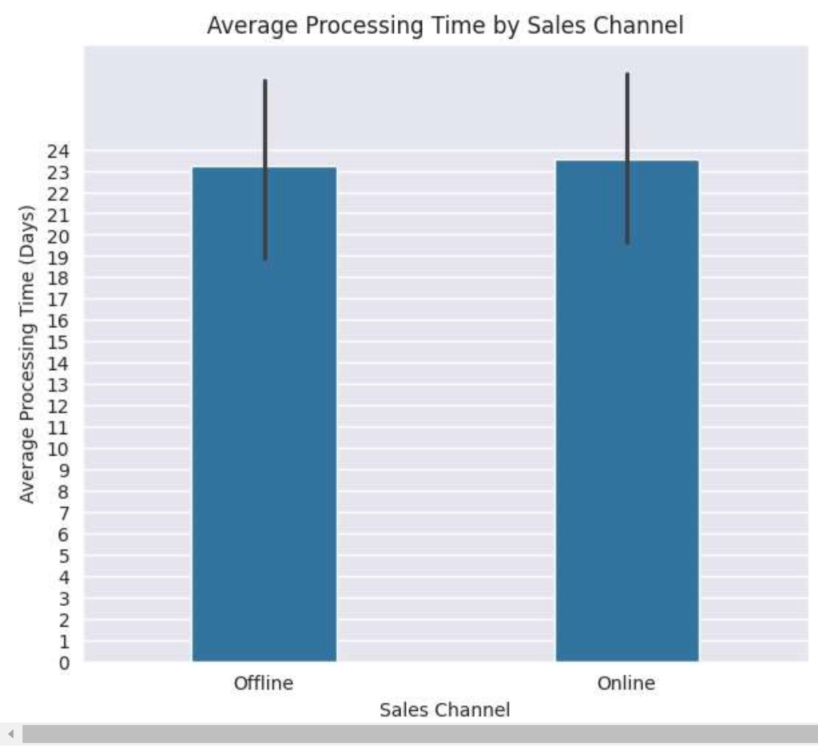
index	Sales Channel	Processing Time
0	Offline	23 days 04:48:00
1	Online	23 days 12:28:48

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots New interactive sheet

```
# Creating the bar plot
plt.figure (figsize = (7, 6))
sns.barplot (data = df,
             x = df ['Sales Channel'],
             y = df ['Processing Time'].dt.days,
             width = 0.4)

# Adding labels, title, and ticks
plt.title ('Average Processing Time by Sales Channel')
plt.xlabel ('Sales Channel')
plt.yticks (np.arange (0, 25, 1))
plt.ylabel ('Average Processing Time (Days)')

# Display the plot
plt.show()
```

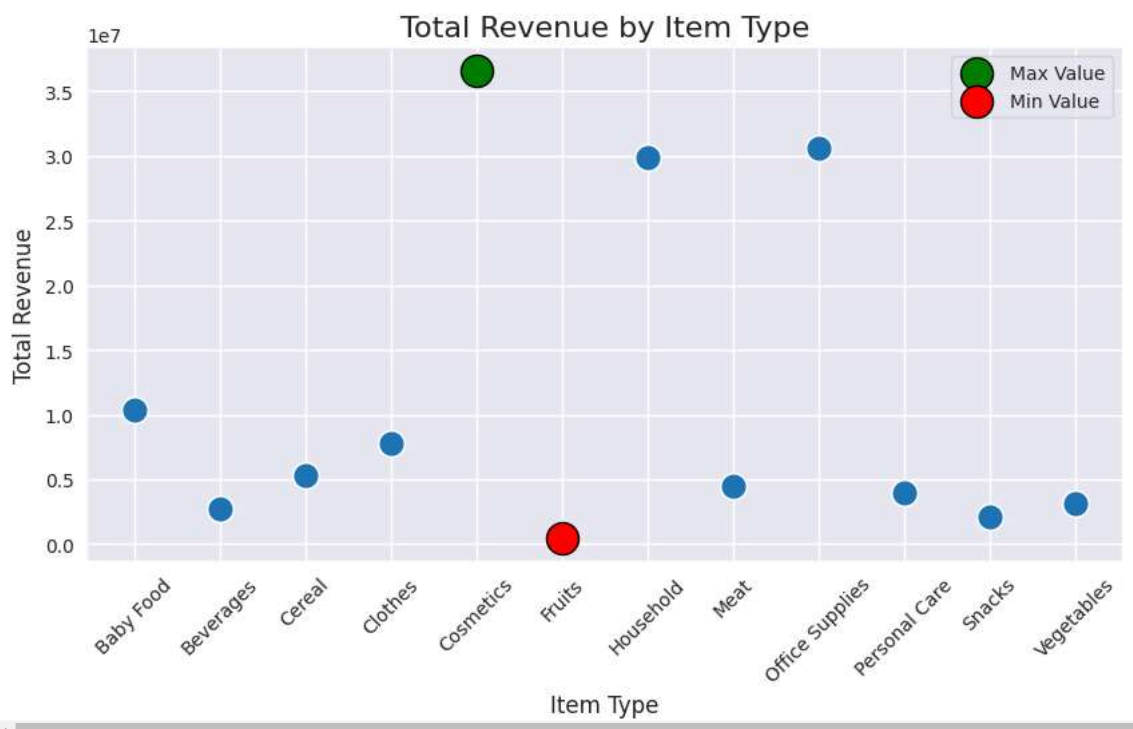


6. Which item types have the highest & lowest total sales?

```
# Grouping by 'Item Type' and summing 'Total Revenue'
item_type = df.groupby(['Item Type'])['Total Revenue'].sum().reset_index()

# Finding the highest and lowest sales revenue item types
highest_sales_revenue_item_type = item_type.loc[item_type['Total Revenue'].idxmax(), 'Item Type']
lowest_sales_revenue_item_type = item_type.loc[item_type['Total Revenue'].idxmin(), 'Item Type']

# Printing the results
print(f"Highest Sales Revenue by Item Type: {highest_sales_revenue_item_type}")
print(f"Lowest Sales Revenue by Item Type: {lowest_sales_revenue_item_type}")
```

```
Highest Sales Revenue by Item Type: Cosmetics
Lowest Sales Revenue by Item Type: Fruits
```

```python
# Plotting a scatter plot for Total Revenue by Item Type
plt.figure(figsize=(10, 5))

# Scatter plot using Seaborn
sns.scatterplot(
    data=item_type,
    x='Item Type',
    y='Total Revenue',
    s=200
)

# Highlighting the Max Value
max_index = item_type['Total Revenue'].idxmax()
plt.scatter(
    x=item_type.loc[max_index, 'Item Type'],
    y=item_type.loc[max_index, 'Total Revenue'],
    s=300,
    color='Green',
    edgecolor='Black',
    label='Max Value'
)

# Highlighting the Min Value
min_index = item_type['Total Revenue'].idxmin()
plt.scatter(
    x=item_type.loc[min_index, 'Item Type'],
    y=item_type.loc[min_index, 'Total Revenue'],
    s=300,
    color='Red',
    edgecolor='Black',
    label='Min Value'
)

# Adding labels and title
plt.title('Total Revenue by Item Type', fontsize=16)
plt.xlabel('Item Type', fontsize=12)
plt.ylabel('Total Revenue', fontsize=12)

# Rotate x-axis labels for readability
plt.xticks(rotation=45)

# Add legend
plt.legend()

# Display the plot
plt.show()
```

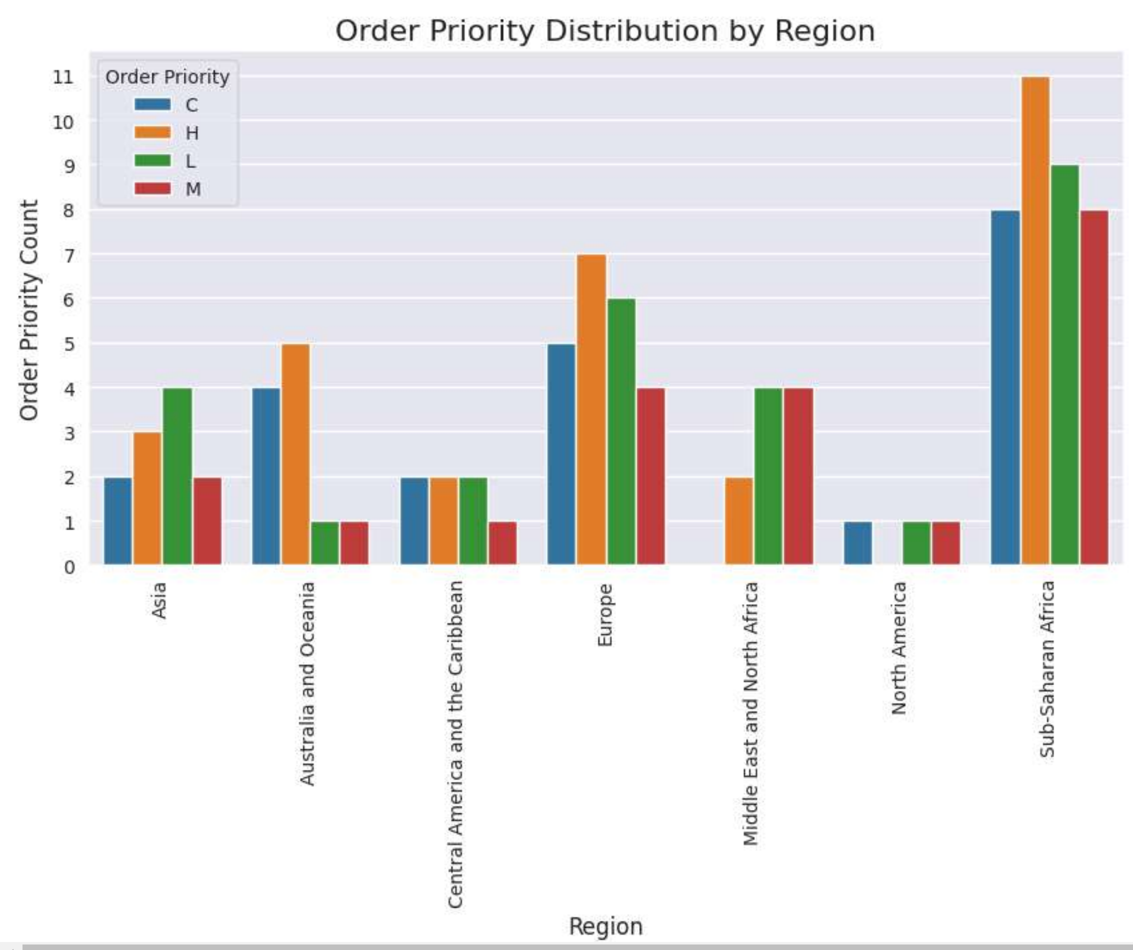


7. How does the order priority vary across different regions?

```python
diff_region_by_order_priority = df.groupby (['Region']) ['Order Priority'].value_counts()
diff_region_by_order_priority
```

```
                                                    count
                         Region  Order Priority
Asia                             L                  4
                                 H                  3
                                 C                  2
                                 M                  2
Australia and Oceania            H                  5
                                 C                  4
                                 L                  1
                                 M                  1
Central America and the Caribbean C                 2
                                 H                  2
                                 L                  2
                                 M                  1
Europe                           H                  7
                                 L                  6
                                 C                  5
                                 M                  4
Middle East and North Africa     L                  4
                                 M                  4
                                 H                  2
North America                    C                  1
                                 L                  1
                                 M                  1
Sub-Saharan Africa               H                 11
                                 L                  9
                                 C                  8
                                 M                  8

dtype: int64
```

```python
# Grouping data and counting occurrences of Order Priority by Region
Diff_regions_by_order_priority = (
    df.groupby(['Region', 'Order Priority'])
    .size()
    .reset_index(name='Order Priority Count')
)

# Setting up the plot
plt.figure(figsize=(10, 5))

sns.barplot(
    data=Diff_regions_by_order_priority,
    x='Region',
    y='Order Priority Count',
    hue='Order Priority'
)

# Enhancing the visualization
plt.xticks(rotation=90)
plt.yticks(
    np.arange(
        0,
        Diff_regions_by_order_priority['Order Priority Count'].max() + 1,
        1
    )
)
plt.title('Order Priority Distribution by Region', fontsize=16)
plt.xlabel('Region', fontsize=12)
plt.ylabel('Order Priority Count', fontsize=12)

# Display the plot
plt.show()
```

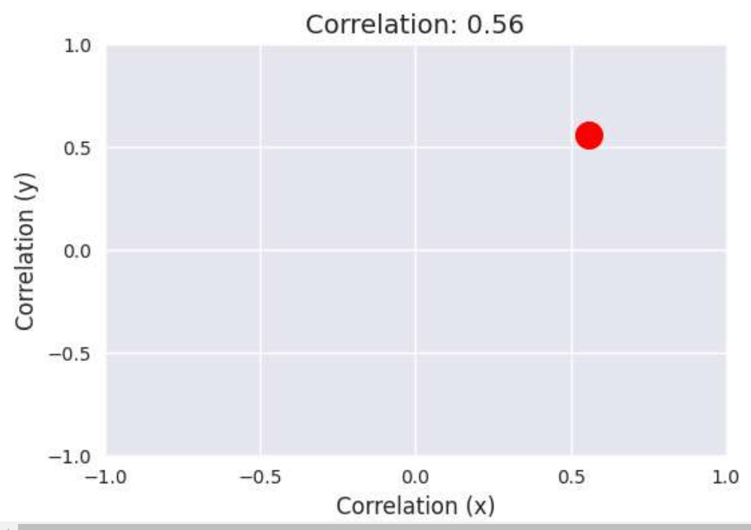


8. What is the correlation between Unit Price & Total Profit?

```
cor = df['Unit Price'].corr (df['Total Profit'])

print (f"Correlation between Unit Price & Total Profit: {cor: .2f}")
```

```
Correlation between Unit Price & Total Profit:  0.56
```

```
# Plotting
plt.figure(figsize=(6, 4))
plt.scatter(
    x=cor,
    y=cor,
    s=200,
    color='red'
)

# Adjusting axis ticks and limits
plt.xticks(np.arange(-1, 1.5, 0.5))
plt.yticks(np.arange(-1, 1.5, 0.5))
plt.xlim(-1, 1)
plt.ylim(-1, 1)

# Adding title and axis labels
plt.title(f'Correlation: {cor:.2f}', fontsize=14)
plt.xlabel('Correlation (x)', fontsize=12)
plt.ylabel('Correlation (y)', fontsize=12)

# Display the plot
plt.show()
```



9. Are there any seasonal trends or patterns in the sales data?

```
# Ensure 'Order Date' is in datetime format
df['Order Date'] = pd.to_datetime(df['Order Date'])

# Grouping data by month and summing 'Total Revenue'
monthly_sales = df.groupby(df['Order Date'].dt.month)['Total Revenue'].sum().reset_index()

# Mapping month numbers to month names
month = {
    1: 'JAN',
    2: 'FEB',
    3: 'MAR',
    4: 'APR',
    5: 'MAY',
    6: 'JUN',
    7: 'JUL',
    8: 'AUG',
    9: 'SEPT',
    10: 'OCT',
    11: 'NOV',
    12: 'DEC'
}
monthly_sales['Month'] = monthly_sales['Order Date'].map(month)

# Selecting only Month and Revenue
monthly_sales = monthly_sales[['Month', 'Total Revenue']]

# Display the result
monthly_sales
```

1 to 12 of 12 entries Filter

index	Month	Total Revenue
0	JAN	10482467.12
1	FEB	24740517.77
2	MAR	2274823.87
3	APR	16187186.33
4	MAY	13215739.99
5	JUN	5230325.77
6	JUL	15669518.5
7	AUG	1128164.91
8	SEPT	5314762.5600000005
9	OCT	15287576.61
10	NOV	20568222.759999998
11	DEC	7249462.12

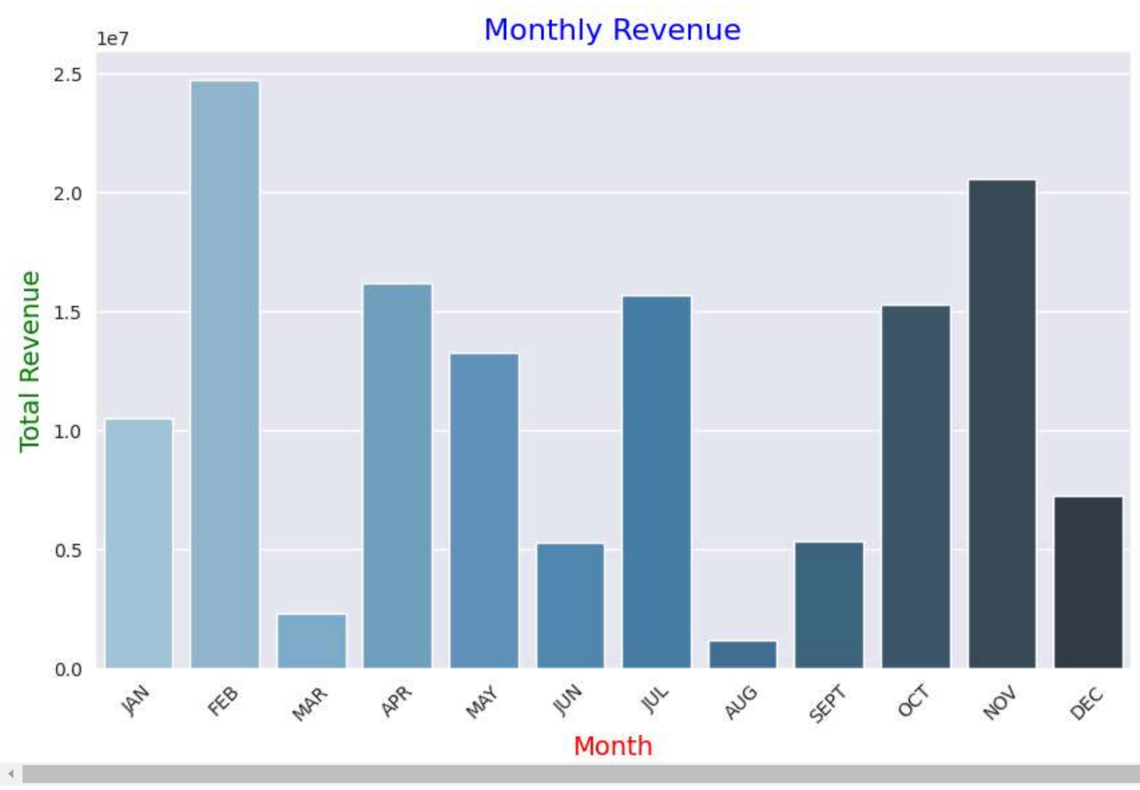
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots New interactive sheet

```
# Creating the bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=monthly_sales, x='Month', y='Total Revenue', hue='Month',
            palette='Blues_d', dodge=False, legend=False)

# Adding labels and title
plt.title('Monthly Revenue', fontsize=16, color='blue')
plt.xlabel('Month', fontsize=14, color='red')
plt.ylabel('Total Revenue', fontsize=14, color='green')
```

```
# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Display the plot
plt.show()
```



10. How does the number if units sold across different countries?

```
countries = df.groupby (df ['Country']) ['Units Sold'].sum().reset_index(name = 'Units Sold')

pd.set_option ('display.max_rows', None)

countries
```

1 to 25 of 76 entries Filter

index	Country	Units Sold
0	Albania	2269
1	Angola	4187
2	Australia	12995
3	Austria	2847
4	Azerbaijan	9255
5	Bangladesh	8263
6	Belize	5498
7	Brunei	6708
8	Bulgaria	5660
9	Burkina Faso	8082
10	Cameroon	10948
11	Cape Verde	4168
12	Comoros	962
13	Costa Rica	6409
14	Cote d'Ivoire	3482
15	Democratic Republic of the Congo	5741
16	Djibouti	23198
17	East Timor	5908
18	Federated States of Micronesia	9379
19	Fiji	9905
20	France	1815
21	Gabon	8656
22	Grenada	2804
23	Haiti	1705
24	Honduras	11199

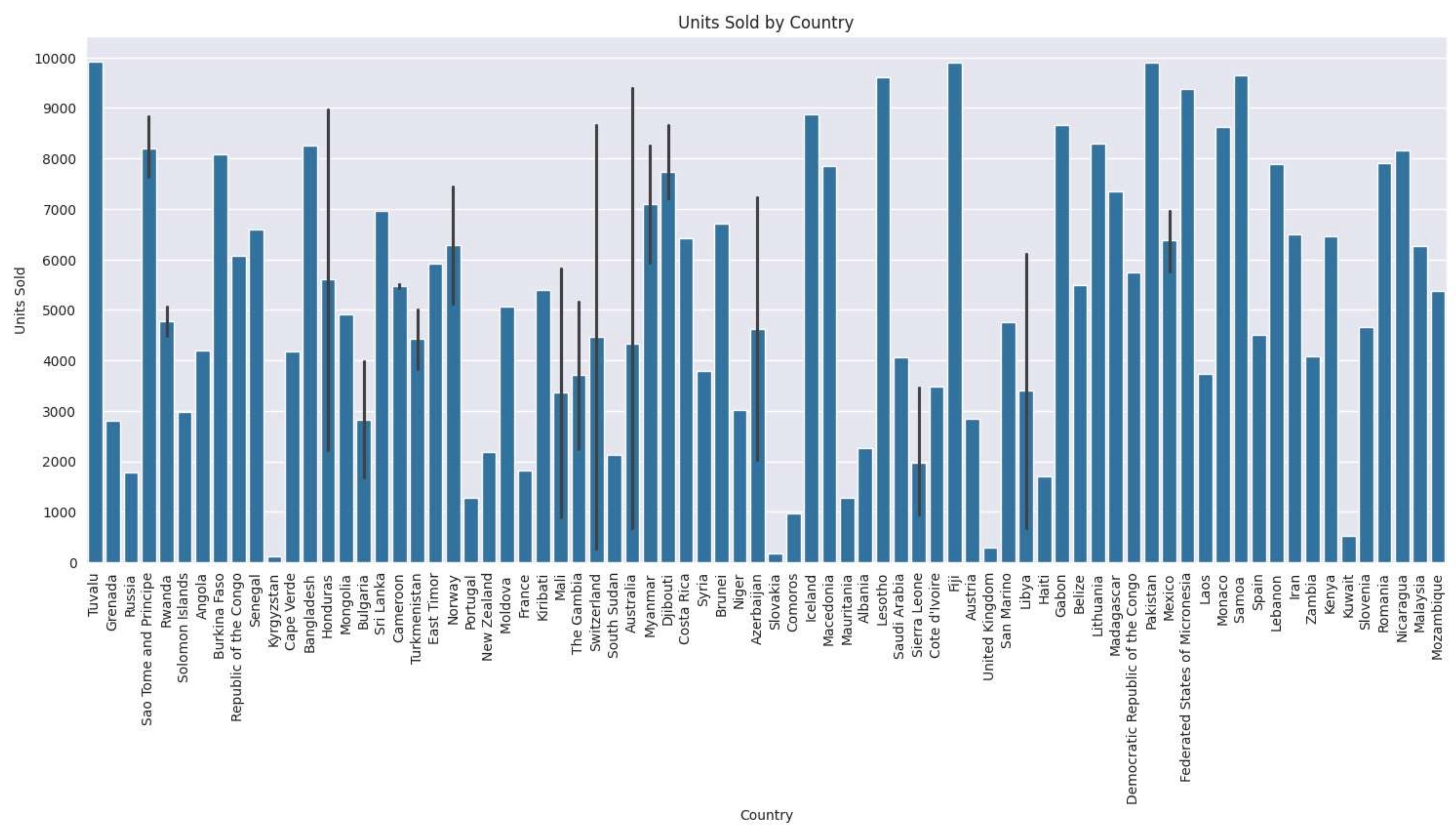
Show 25 per page 1 2 3 4

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots New interactive sheet

Units Sold

```
# @title Units Sold
plt.figure(figsize=(18, 7))

# If 'df' is the correct DataFrame name, and 'countries' contains the aggregated sales per country
sns.barplot(data=df, x='Country', y='Units Sold')

# Rotate x-axis labels for readability
plt.xticks(rotation=90)

# Set y-axis tick marks based on the range of 'Unit Sold'
plt.yticks(np.arange(0, df['Units Sold'].max() + 1000, 1000))

# Add titles and axis labels
plt.title("Units Sold by Country")
plt.xlabel("Country")
plt.ylabel("Units Sold")

plt.show()
```



11. How does the total sales revenue vary across different countries?

```
sales_revenue = df.groupby (['Country']) ['Total Revenue'].sum().reset_index(name = 'Total Revenue')
sales_revenue
```

1 to 25 of 76 entries Filter

index	Country	Total Revenue
0	Albania	247956.32
1	Angola	2798046.49
2	Australia	2489933.49
3	Austria	1244708.4
4	Azerbaijan	4478800.21
5	Bangladesh	902980.64
6	Belize	600821.44
7	Brunei	4368316.68
8	Bulgaria	2779199.71
9	Burkina Faso	1245112.92
10	Cameroon	3851030.28
11	Cape Verde	455479.04
12	Comoros	197883.4
13	Costa Rica	523807.57
14	Cote d'Ivoire	380512.96
15	Democratic Republic of the Congo	272410.45
16	Djibouti	6052890.86
17	East Timor	2492526.12
18	Federated States of Micronesia	445033.55
19	Fiji	1082418.4
20	France	793518.0
21	Gabon	707454.88
22	Grenada	576782.8
23	Haiti	745426.0
24	Honduras	6336545.48

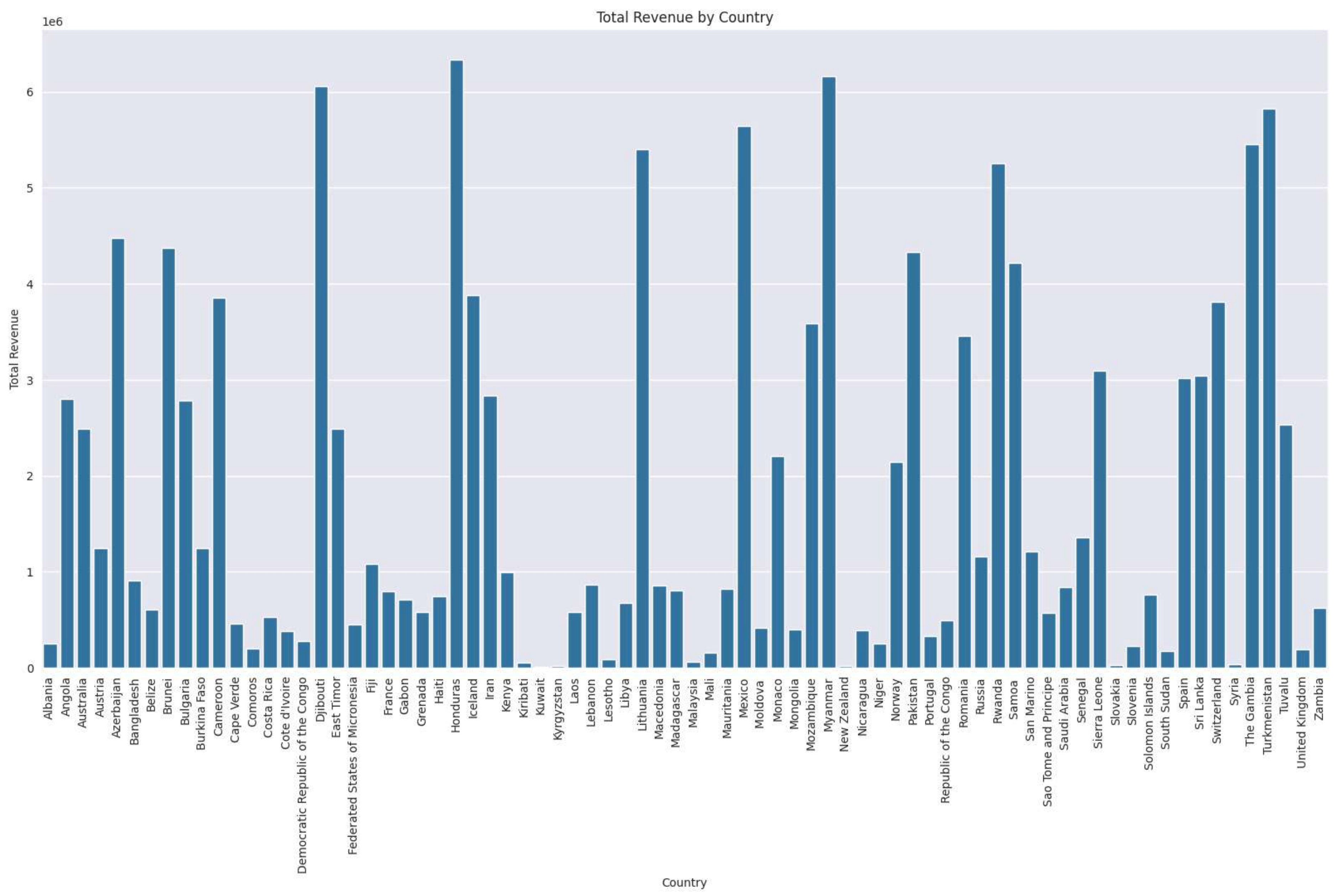
Show 25 per page 1 2 3 4

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots New interactive sheet

Total Revenue

Show code



12. What is the distribution of Unit Prices for each item type?

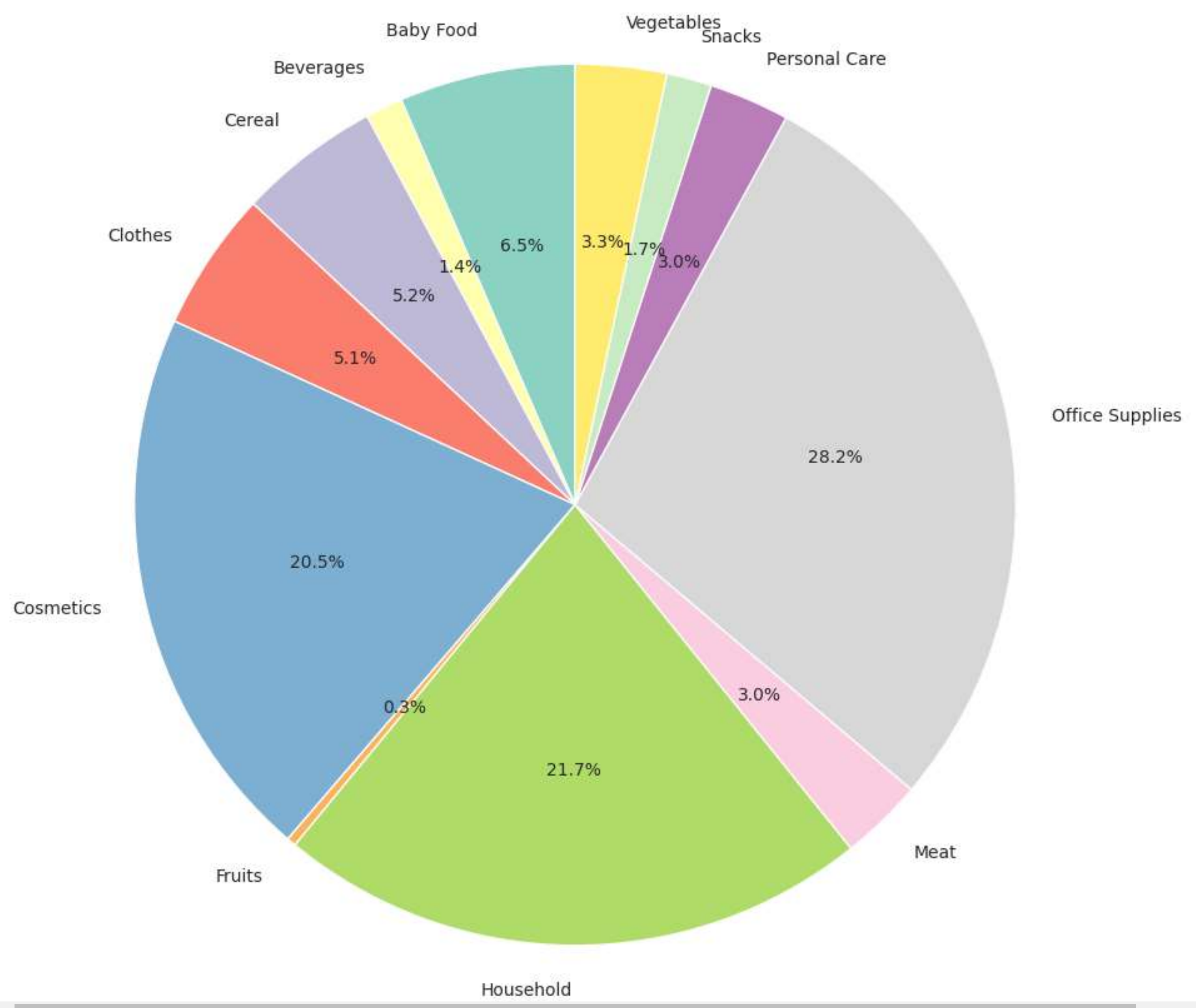
```
item_unit = df.groupby (df ['Item Type']) ['Unit Price'].sum().reset_index (name = 'Unit Price')
item_unit
```

	Item Type	Unit Price
0	Baby Food	1786.96
1	Beverages	379.60
2	Cereal	1439.90
3	Clothes	1420.64
4	Cosmetics	5683.60
5	Fruits	93.30
6	Household	6014.43
7	Meat	843.78
8	Office Supplies	7814.52
9	Personal Care	817.30
10	Snacks	457.74
11	Vegetables	924.36

Next steps: View recommended plots New interactive sheet

Unit Price

```
# @title Unit Price
plt.figure(figsize=(10, 10))  # Optional: Adjust the figure size for better readability

# Pie chart for 'Unit Price' distribution, with 'Item Type' as labels
plt.pie(x=item_unit['Unit Price'],
        labels=item_unit['Item Type'],
        autopct='%1.1f%%',  # Display percentage of each slice
        startangle=90,      # Start the pie chart at 90 degrees
        colors=sns.color_palette('Set3', len(item_unit)))  # Optional: Color palette for clarity

# Equal aspect ratio ensures that the pie is drawn as a circle
plt.axis('equal')

# Add a title to the chart for better context
# plt.title("Distribution of Unit Prices by Item Type")

# Show the plot
plt.show()
```



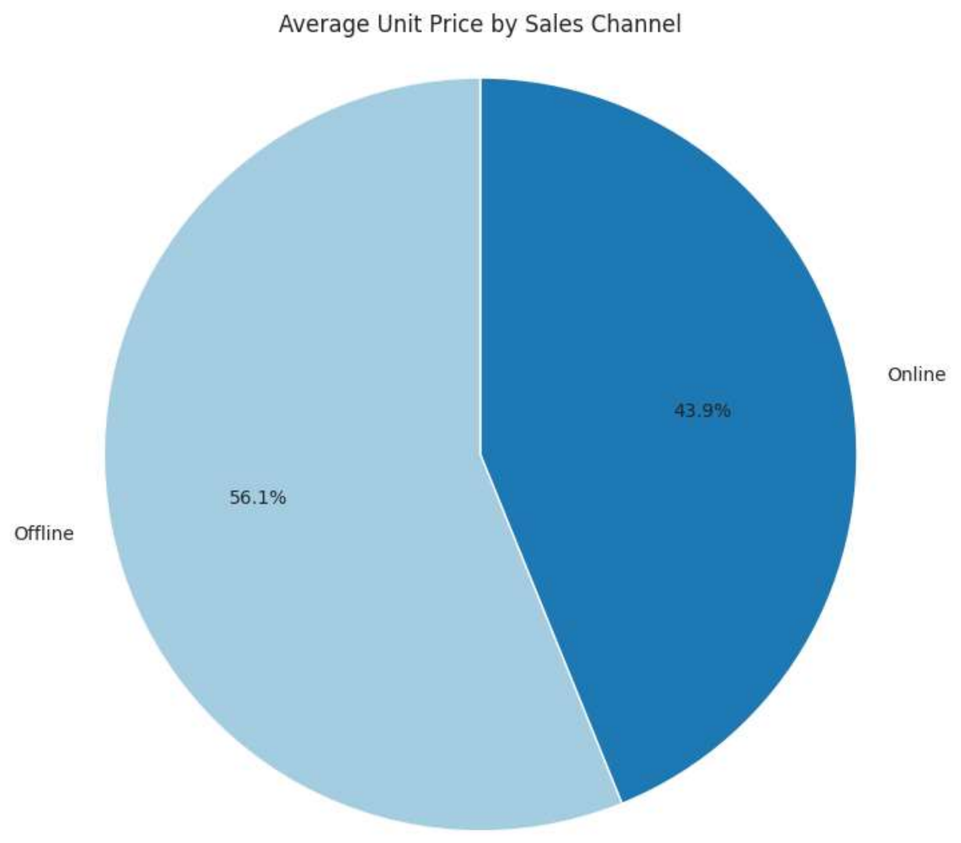
13. Which Sales channel has the highest average unit price?

```
highest_avg_price = df.groupby (['Sales Channel']) ['Unit Price'].mean().reset_index (name = 'Unit Price')
highest_avg_price
```

	Sales Channel	Unit Price
0	Offline	310.7206
1	Online	242.8020

Next steps: View recommended plots New interactive sheet

```
plt.figure(figsize=(8, 8))
plt.pie(highest_avg_price['Unit Price'], labels=highest_avg_price['Sales Channel'],
        autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)
plt.title('Average Unit Price by Sales Channel')
plt.axis('equal')  # Equal aspect ratio ensures that pie chart is drawn as a circle.
plt.show()
```



14. Are there any outliers in the Total Cost Distribution?

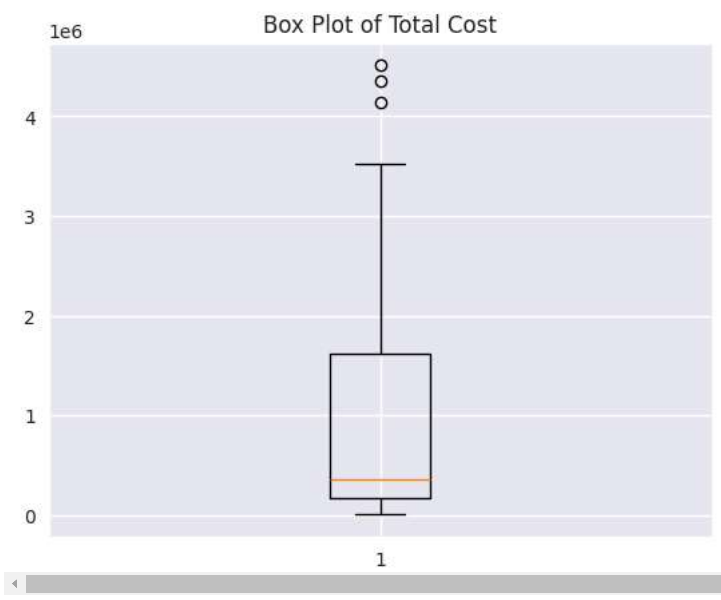
```
q1 = df['Total Cost'].quantile (0.25)
q3 = df['Total Cost'].quantile (0.75)

iqr = q3 - q1

lower_fence = q1 - 1.5 * iqr
upper_fence = q3 + 1.5 * iqr

outliers = df[(df['Total Cost'] < lower_fence) | (df['Total Cost'] > upper_fence)].reset_index(drop = True)

outliers
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit	Processing Time
0	Central America and the Caribbean	Honduras	Household	Offline	H	2017-02-08	522840487	2017-02-13	8974	668.27	502.54	5997054.98	4509793.96	1487261.02	5 days

Next steps: View recommended plots New interactive sheet

Order ID

```
# @title Order ID


plt.boxplot (df['Total Cost'])
plt.title ('Box Plot of Total Cost')
plt.show()
```



15. How does the total profit vary acroos different item types?

```
x = df.groupby (df['Item Type']) ['Total Profit'].sum().reset_index (name = 'Total Profit')
x
```

	Item Type	Total Profit
0	Baby Food	3886643.70
1	Beverages	888047.28
2	Cereal	2292443.43
3	Clothes	5233334.40
4	Cosmetics	14556048.66
5	Fruits	120495.18
6	Household	7412605.71
7	Meat	610610.00
8	Office Supplies	5929583.75
9	Personal Care	1220622.48
10	Snacks	751944.18
11	Vegetables	1265819.63

Next steps: View recommended plots New interactive sheet

16. What is the average order processing time for each country?

```
avg_time = df.groupby (df['Country']) ['Processing Time'].mean().reset_index()
avg_time
```

	Country	Processing Time
0	Albania	44 days 00:00:00
1	Angola	4 days 00:00:00
2	Australia	18 days 16:00:00
3	Austria	7 days 00:00:00
4	Azerbaijan	30 days 00:00:00
5	Bangladesh	47 days 00:00:00
6	Belize	44 days 00:00:00
7	Brunei	37 days 00:00:00
8	Bulgaria	26 days 12:00:00
9	Burkina Faso	10 days 00:00:00
10	Cameroon	12 days 12:00:00
11	Cape Verde	17 days 00:00:00
12	Comoros	31 days 00:00:00
13	Costa Rica	13 days 00:00:00
14	Cote d'Ivoire	19 days 00:00:00