In [1]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import plotly.express as px
import seaborn as sns
```

In [2]:
```
df=pd.read_excel('Amazon Sales data.xlsx')
```

In [3]:
```
df
```

Out[3]:

	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
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.50	951410.50
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.44	248406.36
2	Europe	Russia	Office Supplies	Offline	L	2014-02-05 00:00:00	341417157	2014-08-05 00:00:00	1779	651.21	524.96	1158502.59	933903.84	224598.75
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	2014-05-07 00:00:00	8102	9.33	6.92	75591.66	56065.84	19525.82
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-01-02 00:00:00	115456712	2013-06-02 00:00:00	5062	651.21	524.96	3296425.02	2657347.52	639077.50
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	2011-03-09 00:00:00	888	109.28	35.84	97040.64	31825.92	65214.72
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11 00:00:00	810711038	12/28/2011	6267	9.33	6.92	58471.11	43367.64	15103.47
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	2016-01-06 00:00:00	728815257	6/29/2016	1485	154.06	90.93	228779.10	135031.05	93748.05
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	2015-08-08 00:00:00	5767	81.73	56.67	471336.91	326815.89	144521.02
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-10-02 00:00:00	665095412	2/15/2012	5367	668.27	502.54	3586605.09	2697132.18	889472.91

100 rows × 14 columns

In [4]:
```
df.dtypes
```

Out[4]:
```
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
```

In [6]:
```
df.shape
```

Out[6]:
```
(100, 14)
```

In [7]:
```
# Data has 100 rows and 14 columns.
```

In [9]:
```
df.duplicated().sum()
```

Out[9]:
```
0
```

In [10]:
```
# Data has no duplicate value.
```

In [11]:
```
# Handling missing data
df.isnull().sum()
```

Out[11]:
```
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
```

In [12]:
```
# There is no missing data .
```

In [13]:
```
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
```

In [14]:
```
df.describe().transpose()
```

Out[14]:

	count	mean	std	min	25%	50%	75%	max
Order ID	100.0	5.550204e+08	2.606153e+08	1.146066e+08	3.389225e+08	5.577086e+08	7.907551e+08	9.940222e+08
Units Sold	100.0	5.128710e+03	2.794485e+03	1.240000e+02	2.836250e+03	5.382500e+03	7.369000e+03	9.925000e+03
Unit Price	100.0	2.767613e+02	2.355922e+02	9.330000e+00	8.173000e+01	1.798800e+02	4.372000e+02	6.682700e+02
Unit Cost	100.0	1.910480e+02	1.882082e+02	6.920000e+00	3.584000e+01	1.072750e+02	2.633300e+02	5.249600e+02
Total Revenue	100.0	1.373488e+06	1.460029e+06	4.870260e+03	2.687212e+05	7.523144e+05	2.212045e+06	5.997055e+06
Total Cost	100.0	9.318057e+05	1.083938e+06	3.612240e+03	1.688680e+05	3.635664e+05	1.613870e+06	4.509794e+06
Total Profit	100.0	4.416820e+05	4.385379e+05	1.258020e+03	1.214436e+05	2.907680e+05	6.358288e+05	1.719922e+06

In [19]:
```
df['Order Date']=pd.to_datetime(df['Order Date']).replace('-','/')
```

In [20]:
```
df['Ship Date']=pd.to_datetime(df['Ship Date']).replace('-','/')
```

In [21]:
```
df.dtypes
```

Out[21]:
```
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
```

In [22]:
```
df
```

Out[22]:

	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
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248406.36
2	Europe	Russia	Office Supplies	Offline	L	2014-02-05	341417157	2014-08-05	1779	651.21	524.96	1158502.59	933903.84	224598.75
3	Sub/Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-05-07	8102	9.33	6.92	75591.66	56065.84	19525.82
4	Sub/Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-01-02	115456712	2013-06-02	5062	651.21	524.96	3296425.02	2657347.52	639077.50
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	Sub/Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-03-09	888	109.28	35.84	97040.64	31825.92	65214.72
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	58471.11	43367.64	15103.47
97	Sub/Saharan Africa	Sierra Leone	Vegetables	Offline	C	2016-01-06	728815257	2016-06-29	1485	154.06	90.93	228779.10	135031.05	93748.05
98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	471336.91	326815.89	144521.02
99	Sub/Saharan Africa	Mozambique	Household	Offline	L	2012-10-02	665095412	2012-02-15	5367	668.27	502.54	3586605.09	2697132.18	889472.91

100 rows × 14 columns

In [25]:

```
# Extracting Year from OrderDate
df['sale_year'] = df['Order Date'].dt.year

# Extracting Month from OrderDate
df['sale_month'] = df['Order Date'].dt.month
# Extracting Month Year from OrderDate
df['year_month'] = df['Order Date'].apply(lambda x:x.strftime('%Y-%m'))
```

In [26]:
```
df
```

Out[26]:

	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	sale_yea
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50	201(
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248406.36	2012
2	Europe	Russia	Office Supplies	Offline	L	2014-02-05	341417157	2014-08-05	1779	651.21	524.96	1158502.59	933903.84	224598.75	2014
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-05-07	8102	9.33	6.92	75591.66	56065.84	19525.82	2014
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-01-02	115456712	2013-06-02	5062	651.21	524.96	3296425.02	2657347.52	639077.50	2013
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
95	Sub-Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-03-09	888	109.28	35.84	97040.64	31825.92	65214.72	2011
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	58471.11	43367.64	15103.47	2011
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	2016-01-06	728815257	2016-06-29	1485	154.06	90.93	228779.10	135031.05	93748.05	2016
98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	471336.91	326815.89	144521.02	2015
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-10-02	665095412	2012-02-15	5367	668.27	502.54	3586605.09	2697132.18	889472.91	2012

100 rows × 17 columns

Exploring data

In [31]:
```
#Sales-trend -> month-wise, year-wise, yearly_month-wise
```

In [32]:
```
df[['Total Revenue','sale_month','sale_year','year_month']]
```

Out[32]:

	Total Revenue	sale_month	sale_year	year_month
0	2533654.00	5	2010	2010-05
1	576782.80	8	2012	2012-08
2	1158502.59	2	2014	2014-02
3	75591.66	6	2014	2014-06
4	3296425.02	1	2013	2013-01
...	...	...	...	...
95	97040.64	7	2011	2011-07
96	58471.11	11	2011	2011-11
97	228779.10	1	2016	2016-01
98	471336.91	7	2015	2015-07
99	3586605.09	10	2012	2012-10

100 rows × 4 columns

Plot Yearly Sales Trend

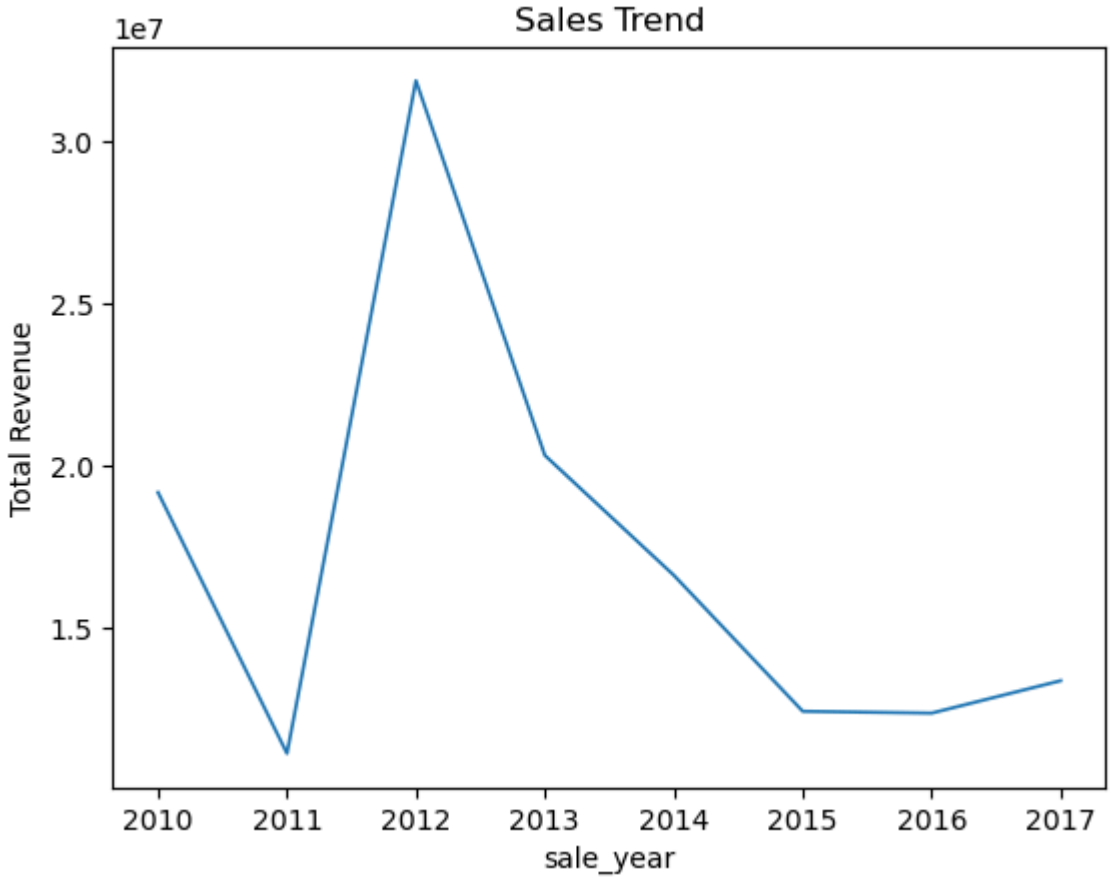
In [33]:
```
df.sale_year.unique()
```

Out[33]:
```
array([2010, 2012, 2014, 2013, 2015, 2011, 2017, 2016])
```

In [35]:
```
years = [unique for unique in df.sale_year.unique()]
years
```

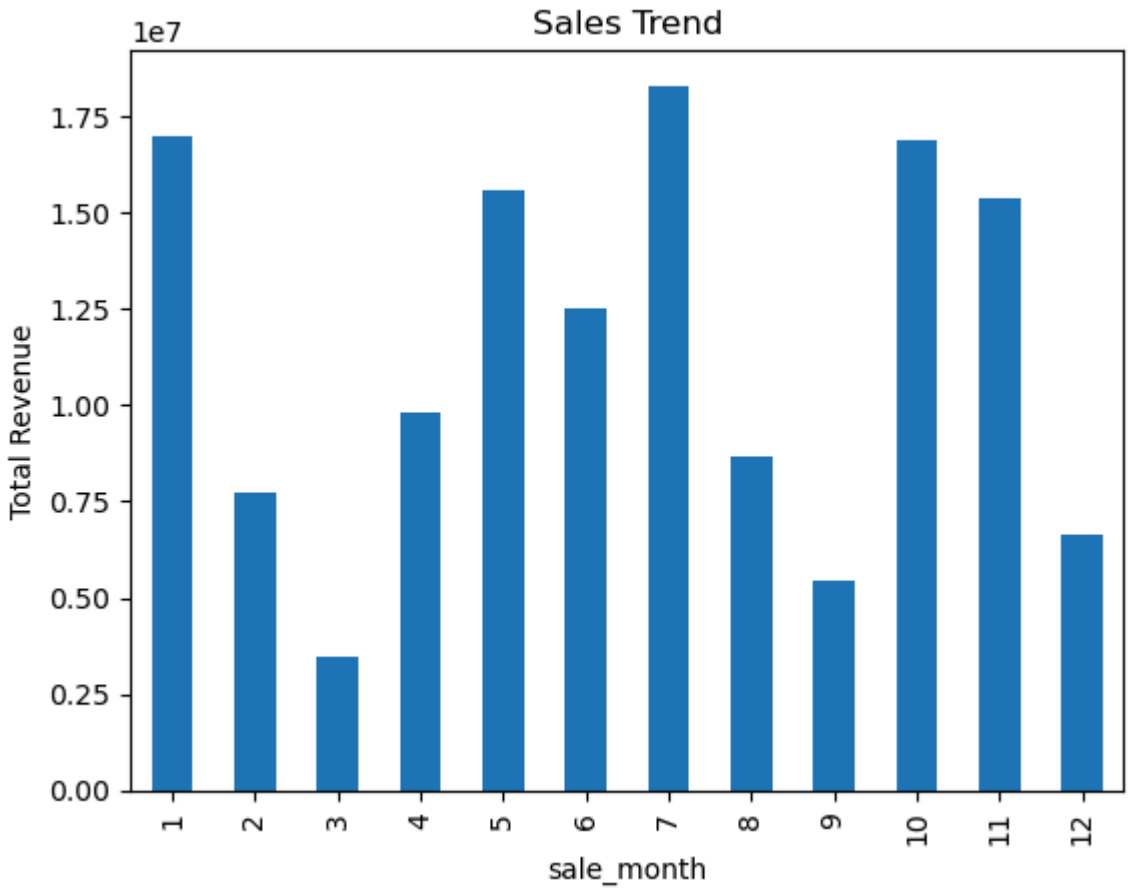
Out[35]:
```
[2010, 2012, 2014, 2013, 2015, 2011, 2017, 2016]
```

In [37]:
```
df.groupby('sale_year')['Total Revenue'].sum().plot(linewidth=1.2,
                                                    ylabel='Total Revenue',
                                                    xlabel='sale_year',
                                                    title='Sales Trend');
```



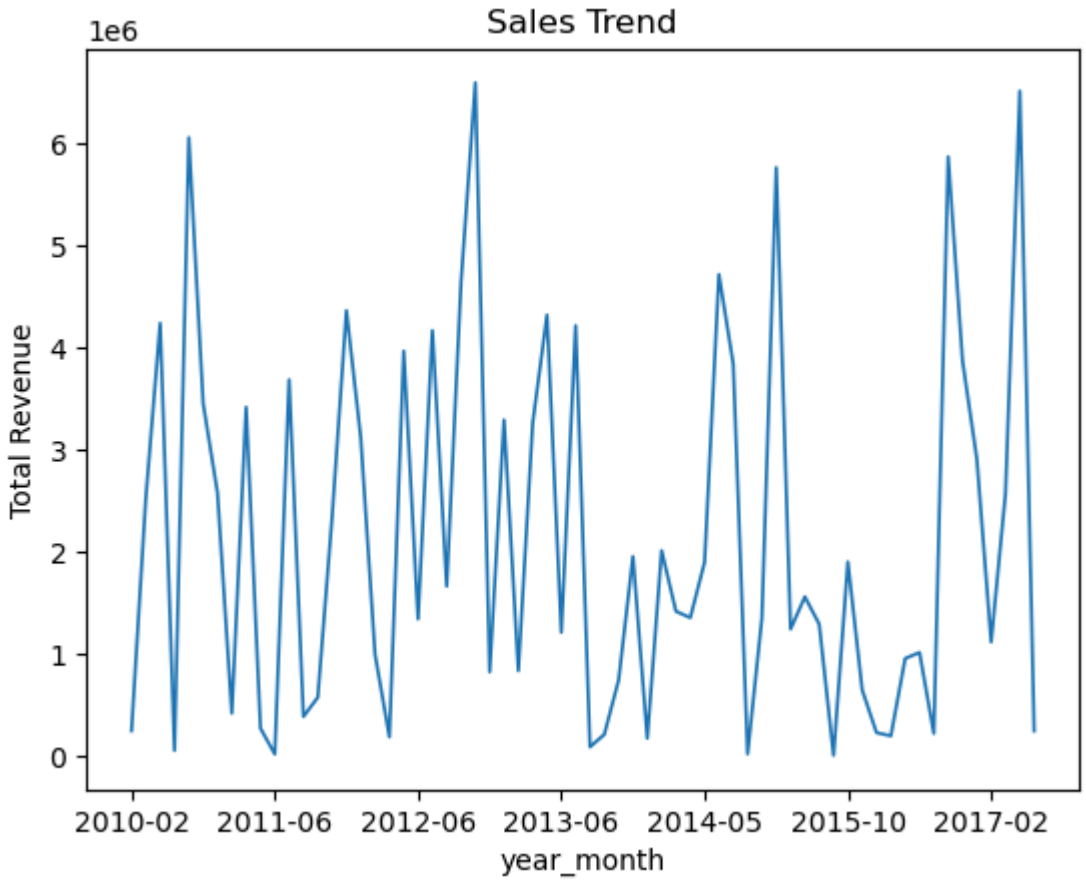
In [ ]:
```
# Sales trend plot shows most of the sale are made in the year 2012.then it starts decreasing and after 2016 there is slight incr
```

In [44]:
```
df.groupby('sale_month')['Total Revenue'].sum().plot(linewidth=1.2, kind='bar',
                                                     ylabel='Total Revenue',
                                                     xlabel='sale_month',
                                                     title='Sales Trend');
```



In [ ]:
```
# Sales trend plot shows most of the sale are made in the month of july i.e 7th month.
```

In [41]:
```
df.groupby('year_month')['Total Revenue'].sum().plot(linewidth=1.2,
                                                     ylabel='Total Revenue',
                                                     xlabel='year_month',
                                                     title='Sales Trend');
```



In [42]:
```
# Product Analysis
```

In [45]:
```
products = pd.DataFrame(df[['Units Sold','Total Revenue','Item Type','sale_month', 'Region']].groupby('sale_month')['Item Type'].
products
```

Out[45]:

		count
sale_month	**Item Type**	
1	**Beverages**	2
	Office Supplies	2
	Vegetables	1
	Meat	1
	Household	1
...	**...**	...
11	**Clothes**	1
	Personal Care	1
12	**Personal Care**	2
	Cosmetics	1
	Household	1

69 rows × 1 columns

In [73]:
```
df[['sale_month','Item Type']]['Item Type'].value_counts()
```

Out[73]:
```
Item Type
Clothes           13
Cosmetics         13
Office Supplies   12
Fruits            10
Personal Care     10
Household          9
Beverages          8
Baby Food          7
Cereal             7
Vegetables         6
Snacks             3
Meat               2
Name: count, dtype: int64
```

In [74]:
```
# It means the item types which are sold maximum are Clothes and Cosmetics.
```

In [60]:
```
df
```

Out[60]:

	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	sale_yea
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50	201(
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248406.36	201
2	Europe	Russia	Office Supplies	Offline	L	2014-02-05	341417157	2014-08-05	1779	651.21	524.96	1158502.59	933903.84	224598.75	201
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-05-07	8102	9.33	6.92	75591.66	56065.84	19525.82	201
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-01-02	115456712	2013-06-02	5062	651.21	524.96	3296425.02	2657347.52	639077.50	201
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	Sub-Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-03-09	888	109.28	35.84	97040.64	31825.92	65214.72	201
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	58471.11	43367.64	15103.47	201
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	2016-01-06	728815257	2016-06-29	1485	154.06	90.93	228779.10	135031.05	93748.05	201(
98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	471336.91	326815.89	144521.02	201
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-10-02	665095412	2012-02-15	5367	668.27	502.54	3586605.09	2697132.18	889472.91	201

100 rows × 19 columns

In [109…
```
df.drop(['Sales'],axis=1,inplace=True)
```

In [110…
```
df
```

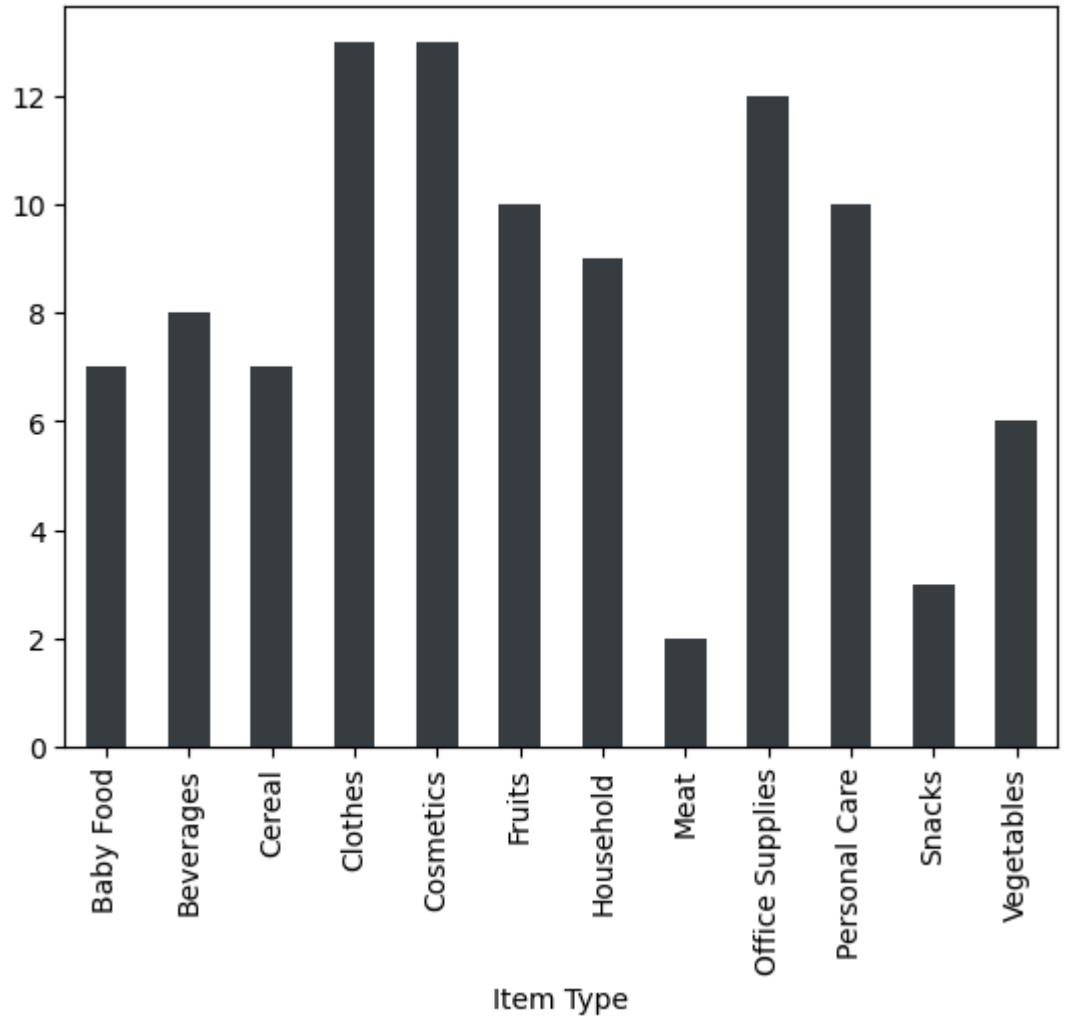
Out[110]:

	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	sale_yea
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50	201(
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248406.36	201
2	Europe	Russia	Office Supplies	Offline	L	2014-02-05	341417157	2014-08-05	1779	651.21	524.96	1158502.59	933903.84	224598.75	201
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-05-07	8102	9.33	6.92	75591.66	56065.84	19525.82	201
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-01-02	115456712	2013-06-02	5062	651.21	524.96	3296425.02	2657347.52	639077.50	201
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	Sub-Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-03-09	888	109.28	35.84	97040.64	31825.92	65214.72	201
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	58471.11	43367.64	15103.47	201
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	2016-01-06	728815257	2016-06-29	1485	154.06	90.93	228779.10	135031.05	93748.05	201(
98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	471336.91	326815.89	144521.02	201
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-10-02	665095412	2012-02-15	5367	668.27	502.54	3586605.09	2697132.18	889472.91	201

100 rows × 17 columns

In [82]:
```
df.groupby('Item Type')['Item Type'].value_counts().plot(kind='bar', color='#374045')
```

Out[82]:
```
<Axes: xlabel='Item Type'>
```



In [83]:
```
#Region Monthly Revenue
```

In [84]:
```
region_sales = pd.DataFrame(df[['Units Sold','Total Revenue','Item Type','sale_month', 'Region']]).groupby(['sale_month','Region'
region_sales = pd.DataFrame(region_sales)
region_sales
```

Out[84]:

sale_month	Region	Total Revenue
1	Asia	10784524.82
	Sub/Saharan Africa	6212943.30
2	Asia	400558.73
	Europe	5247541.58
	North America	524870.06
	Sub/Saharan Africa	1573432.70
3	Middle East and North Africa	835759.10
	Sub/Saharan Africa	2611750.34
4	Asia	3262562.10
	Australia and Oceania	759202.72
	Europe	182825.44
	Middle East and North Africa	4870.26
	Sub/Saharan Africa	5579849.65
5	Australia and Oceania	2533654.00
	Europe	5082788.06
	Middle East and North Africa	4324782.40
	Sub/Saharan Africa	3636749.55
6	Asia	19103.44
	Australia and Oceania	1082418.40
	Central America and the Caribbean	339490.50
	Europe	1238924.26
	Middle East and North Africa	4478800.21
	North America	4647149.58

		Total Revenue
sale_month	Region	
	Sub/Saharan Africa	692938.74
7	Australia and Oceania	7158762.97
	Central America and the Caribbean	600821.44
	Europe	982476.34
	North America	471336.91
	Sub/Saharan Africa	9075815.60
8	Australia and Oceania	20404.71
	Central America and the Caribbean	7484647.55
	Middle East and North Africa	6279.09
	Sub/Saharan Africa	1177591.82
9	Asia	574951.92
	Australia and Oceania	140287.40
	Europe	3786589.20
	Middle East and North Africa	861563.52
	Sub/Saharan Africa	71253.21
10	Australia and Oceania	2399534.93
	Central America and the Caribbean	745426.00
	Europe	9490570.54
	Middle East and North Africa	668356.48
	Sub/Saharan Africa	3586605.09
11	Asia	3745915.91
	Europe	3458252.00
	Middle East and North Africa	2872295.52

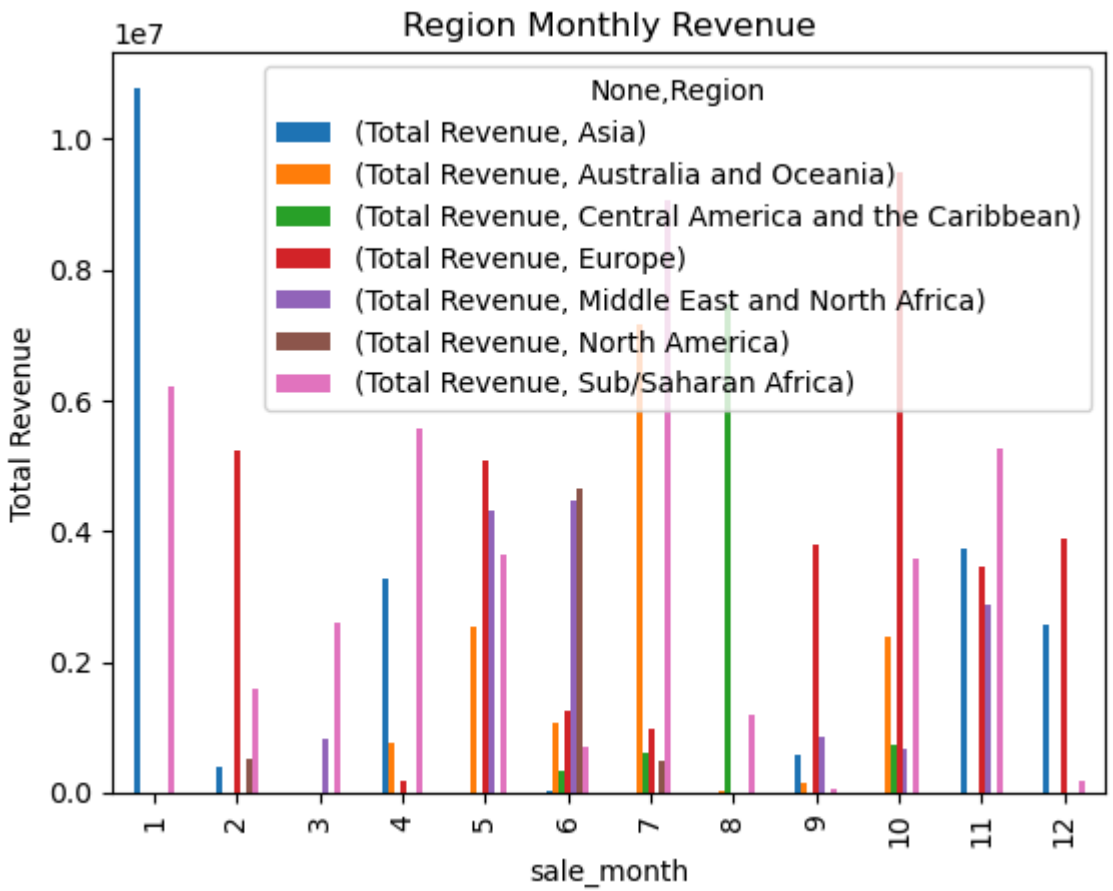
		Total Revenue
sale_month	**Region**	
	Sub/Saharan Africa	5279425.18
12	**Asia**	2559474.10
	Europe	3898964.69
	Sub/Saharan Africa	173676.25

In [85]:
```
region_sales = region_sales.reset_index()
region_sales = region_sales.pivot_table(values=['Total Revenue'], index=['sale_month'], columns=['Region'], aggfunc= np.sum)
region_sales
```

Out[85]:

	Total Revenue						
Region	**Asia**	**Australia and Oceania**	**Central America and the Caribbean**	**Europe**	**Middle East and North Africa**	**North America**	**Sub/Saharan Africa**
sale_month							
1	10784524.82	NaN	NaN	NaN	NaN	NaN	6212943.30
2	400558.73	NaN	NaN	5247541.58	NaN	524870.06	1573432.70
3	NaN	NaN	NaN	NaN	835759.10	NaN	2611750.34
4	3262562.10	759202.72	NaN	182825.44	4870.26	NaN	5579849.65
5	NaN	2533654.00	NaN	5082788.06	4324782.40	NaN	3636749.55
6	19103.44	1082418.40	339490.50	1238924.26	4478800.21	4647149.58	692938.74
7	NaN	7158762.97	600821.44	982476.34	NaN	471336.91	9075815.60
8	NaN	20404.71	7484647.55	NaN	6279.09	NaN	1177591.82
9	574951.92	140287.40	NaN	3786589.20	861563.52	NaN	71253.21
10	NaN	2399534.93	745426.00	9490570.54	668356.48	NaN	3586605.09
11	3745915.91	NaN	NaN	3458252.00	2872295.52	NaN	5279425.18
12	2559474.10	NaN	NaN	3898964.69	NaN	NaN	173676.25

In [86]:
```
region_sales.plot(kind='bar', ylabel='Total Revenue', title='Region Monthly Revenue');
```



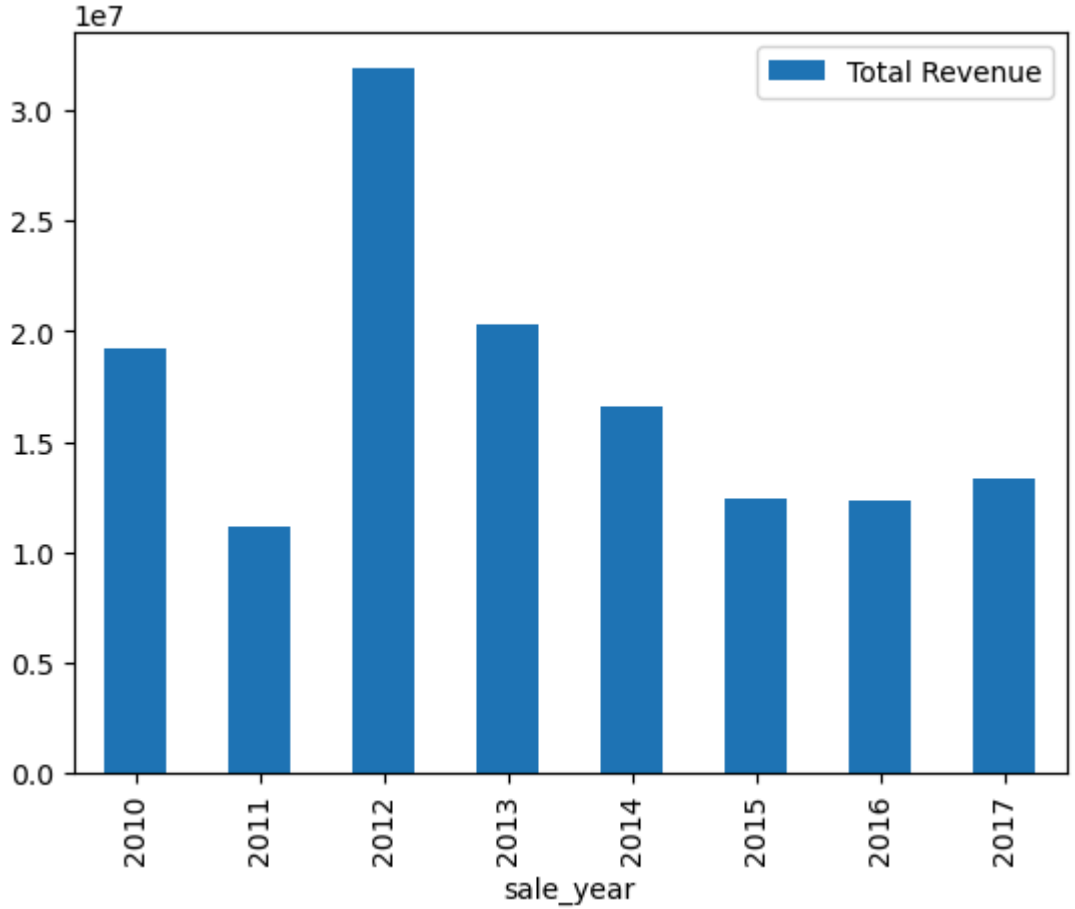
In [90]:
```
#Yearly changes in Total revenue
Yearly_changes = pd.DataFrame(df.groupby(df['sale_year'] )['Total Revenue'].sum())
Yearly_changes
```

Out[90]:

sale_year	Total Revenue
2010	19186024.92
2011	11129166.07
2012	31898644.52
2013	20330448.66
2014	16630214.43
2015	12427982.86
2016	12372867.22
2017	13373419.63

In [92]:
```
Yearly_changes.sort_values('sale_year').plot(kind='bar')
```

Out[92]:
```
<Axes: xlabel='sale_year'>
```



Top 3 Item Types

In [113…

```
df.groupby('Item Type')['Units Sold'].sum().sort_values(ascending=False)
```

Out[113]:
```
Item Type
Cosmetics          83718
Clothes            71260
Beverages          56708
Fruits             49998
Personal Care      48708
Office Supplies    46967
Household          44727
Baby Food          40545
Cereal             25877
Vegetables         20051
Snacks             13637
Meat               10675
Name: Units Sold, dtype: int64
```

In [111…
```
# Top 3 Item Types are Cosmetics,Clothes and Beverages.
```

Country wise Total Revenue

In [123…
```
Country_revenue = pd.DataFrame(df.groupby(df['Country'] )['Total Revenue'].sum().sort_values(ascending=False))
Country_revenue
```

Out[123]:

Country	Total Revenue
Honduras	6336545.48
Myanmar	6161257.90
Djibouti	6052890.86
Turkmenistan	5822036.20
Mexico	5643356.55
...	...
Syria	35304.72
Slovakia	26344.26
New Zealand	20404.71
Kyrgyzstan	19103.44
Kuwait	4870.26

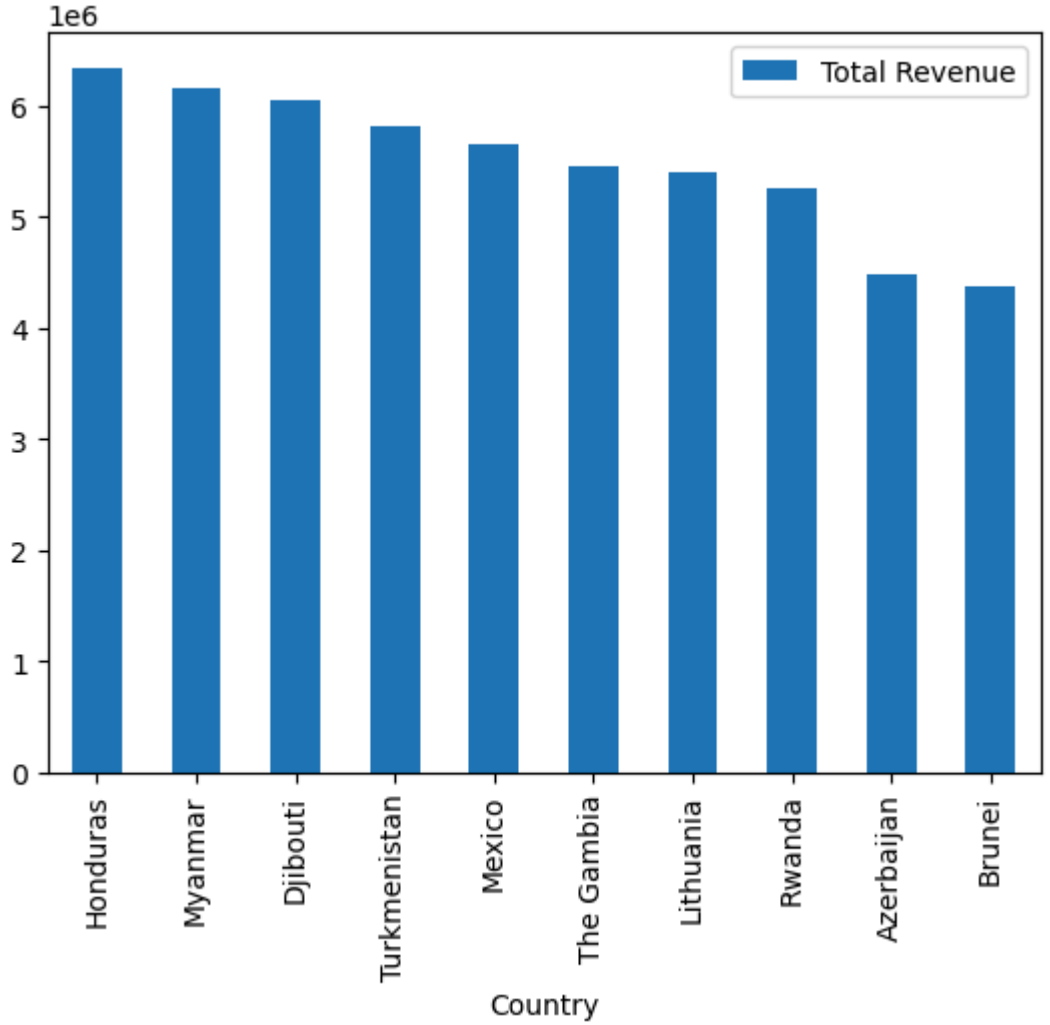
76 rows × 1 columns

In [121…

```
Country_revenue.iloc[0:10].plot(kind='bar')
```

Out[121]:

```
<Axes: xlabel='Country'>
```



In [ ]:
```
 # Top 10 Countries in terms of revenue are
# Honduras
#Myanmar
#Djibouti
#Turkmenistan
#Mexico
#The Gambia
#Lithuania
#Rwanda
```

```
#Azerbaijan
#Brunei
```

In [ ]:

```
#Conclusion/Recommendation:
-The best months for sales are January,July, October and November. The company should look into creating jingles during these per
-Focus the ad targeted audience on Asia and Europe Regions
-The top 2 revenue generating countries are Honduras and Myanmar.
-Finally Cosmetics,Clothes and Beverages sell most during these periods consider getting more of them.
```