Amazon sales Analysis

Problem Statement:

- Sales management has gained importance to meet increasing competition and the need for improved methods of distribution to reduce cost and to increase profits. Sales management today is the most important function in a commercial and business enterprise.

In [1]:
```
# import libraries

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

In [2]:
```
import warnings

# Set the warning filter to 'ignore'
warnings.filterwarnings('ignore')
```

In [3]:
```
# read data set

df=pd.read_csv(r"E:\Projects\Unified_Projects\Amazon sales DataAnalysis\Amazon Sales data.csv")
```

In [4]:
```
# see top 5 rows

df.head()
```

Out[4]:

	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	5/2/2014	341417157	5/8/2014	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	7/5/2014	8102	9.33	6.92	75591.66	56065.84	19525.82
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3296425.02	2657347.52	639077.50

In [5]:
```
# see Dimension of data

df.shape
```

Out[5]: (100, 14)

In [6]:
```
# see column data type and some info

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 [7]:
```
# see precentege of missing value in each column

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

Out[7]:
```
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 [8]:
```
# check if duplicated in data

df.duplicated().any()
```

Out[8]:
```
False
```

In [9]:
```
# see quick info of numeric values

df.describe()
```

Out[9]:

	Order ID	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	1.000000e+02	100.000000	100.000000	100.000000	1.000000e+02	1.000000e+02	1.000000e+02
mean	5.550204e+08	5128.710000	276.761300	191.048000	1.373488e+06	9.318057e+05	4.416820e+05
std	2.606153e+08	2794.484562	235.592241	188.208181	1.460029e+06	1.083938e+06	4.385379e+05
min	1.146066e+08	124.000000	9.330000	6.920000	4.870260e+03	3.612240e+03	1.258020e+03
25%	3.389225e+08	2836.250000	81.730000	35.840000	2.687212e+05	1.688680e+05	1.214436e+05
50%	5.577086e+08	5382.500000	179.880000	107.275000	7.523144e+05	3.635664e+05	2.907680e+05
75%	7.907551e+08	7369.000000	437.200000	263.330000	2.212045e+06	1.613870e+06	6.358288e+05
max	9.940222e+08	9925.000000	668.270000	524.960000	5.997055e+06	4.509794e+06	1.719922e+06

In [ ]:
```

```

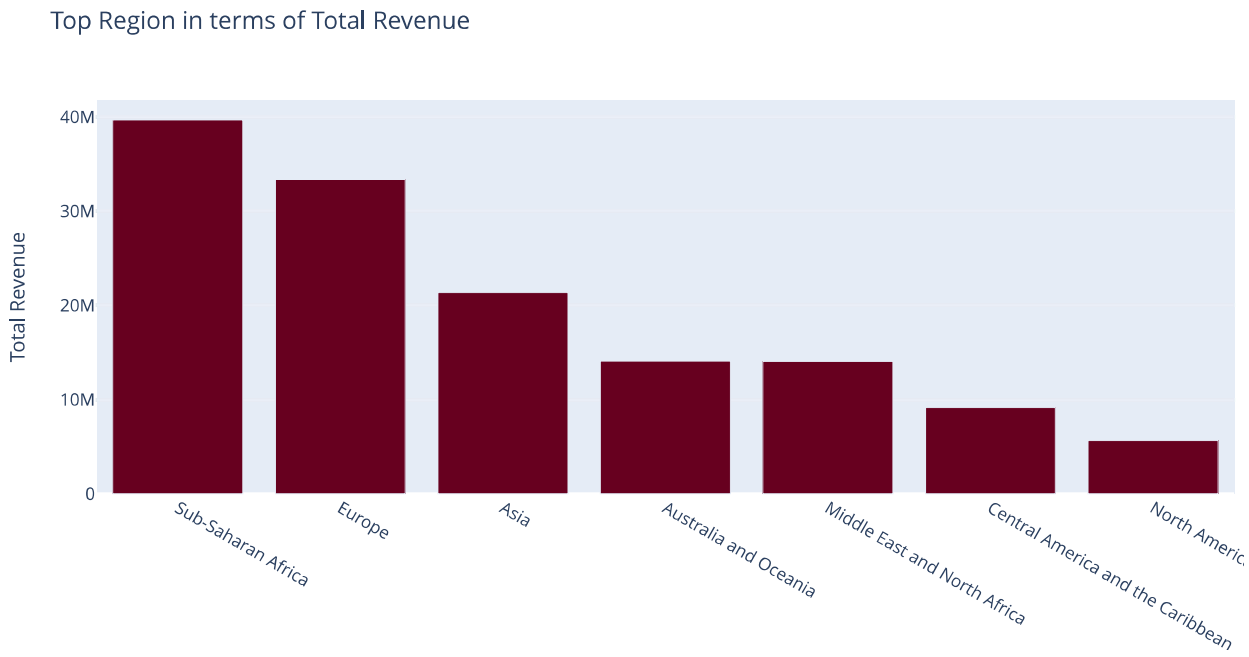
In [10]:
```
# see quick info of category values

df.describe(include = object)
```

Out[10]:

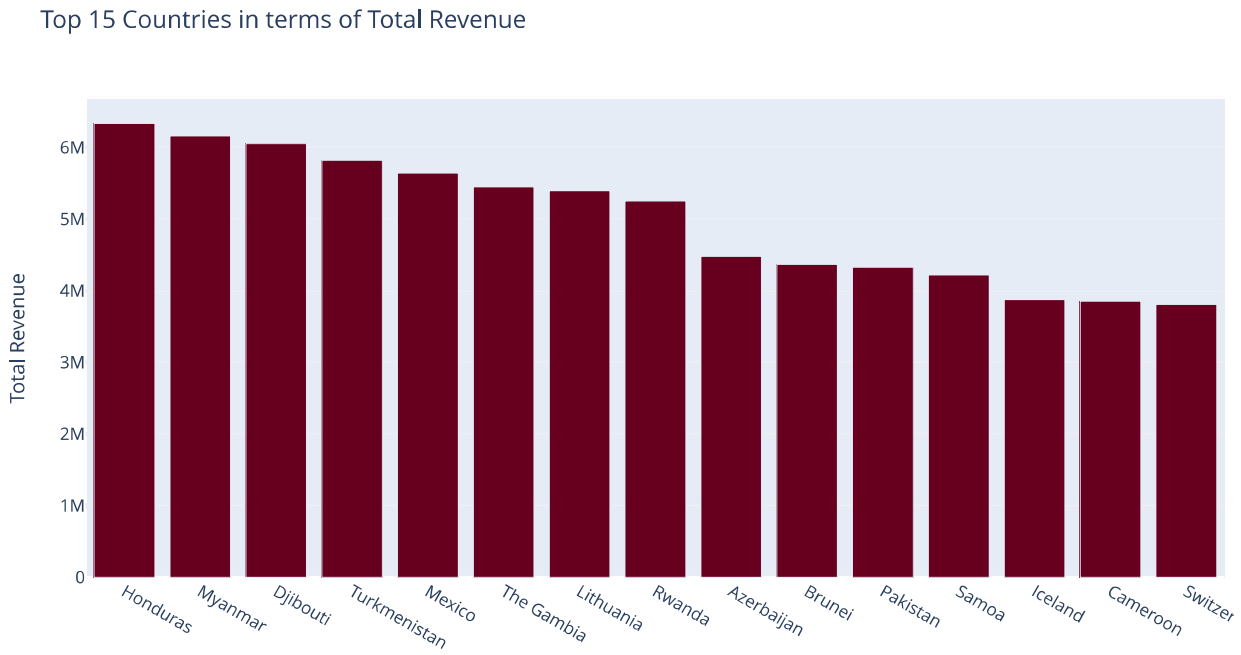
	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Ship Date
count	100	100	100	100	100	100	100
unique	7	76	12	2	4	100	99
top	Sub-Saharan Africa	The Gambia	Clothes	Offline	H	5/28/2010	11/17/2010
freq	36	4	13	50	30	1	2

In [11]:
```
# validate columns types

df['Order Date']=pd.to_datetime(df['Order Date'])
df['Ship Date']=pd.to_datetime(df['Ship Date'])
```

In [12]:
```
by("Region").sum().sort_values("Total Revenue",ascending=False).head(15)
on[['Total Revenue']].round(2)
ex(inplace=True)


,x='Region',y='Total Revenue',title="Top Region in terms of Total Revenue",color_discrete_sequence=px.colors.sequent
```



In [13]:
```
top_countries= df.groupby("Country").sum().sort_values("Total Revenue",ascending=False).head(15)
top_countries= top_countries[['Total Revenue']].round(2)
top_countries.reset_index(inplace=True)
top_countries

fig=px.bar(top_countries,x='Country',y='Total Revenue',title="Top 15 Countries in terms of Total Revenue",color_dis
fig.show()
```



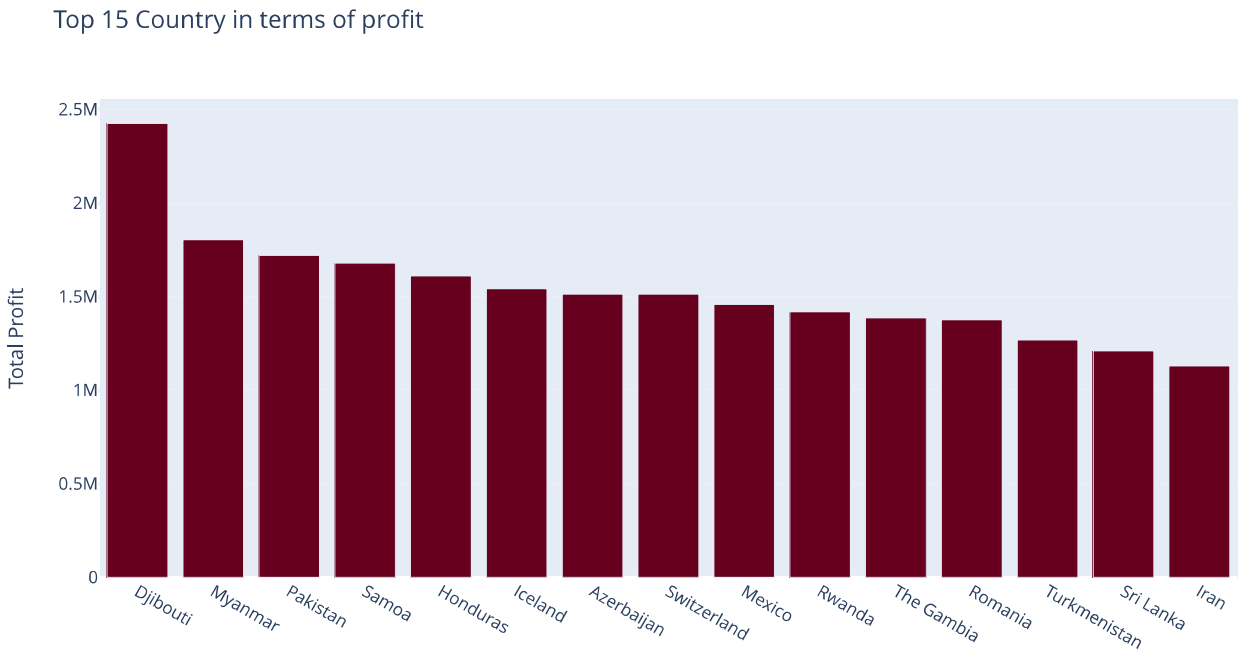
In [14]:
```
# Group by Country and sum the profits

profit_by_country = df.groupby('Country')['Total Profit'].sum()
profit_by_country
```

Out[14]:
```
Country
Albania            166635.36
Angola             693911.51
Australia          576605.12
Austria            495007.89
Azerbaijan        1512926.83
                       ...
The Gambia        1385883.27
Turkmenistan      1267258.40
Tuvalu             951410.50
United Kingdom      46735.86
Zambia             225246.90
Name: Total Profit, Length: 76, dtype: float64
```

In [15]:
```
# Sort the values in descending order

country_max_profit_sorted = profit_by_country.sort_values(ascending=False)
country_max_profit_sorted
```

Out[15]:
```
Country
Djibouti       2425317.87
Myanmar        1802771.70
Pakistan       1719922.04
Samoa          1678540.98
Honduras       1609947.52
                     ...
Slovakia         10795.23
Syria             9119.44
Kyrgyzstan        7828.12
New Zealand       5270.67
Kuwait            1258.02
Name: Total Profit, Length: 76, dtype: float64
```

In [16]:
```
top_region_profit= df.groupby("Country").sum().sort_values("Total Profit",ascending=False).head(15)
top_region_profit= top_region_profit[['Total Profit']].round(2)
top_region_profit.reset_index(inplace=True)
top_region_profit

fig=px.bar(top_region_profit,x='Country',y='Total Profit',title="Top 15 Country in terms of profit",color_discrete_
fig.show()
```



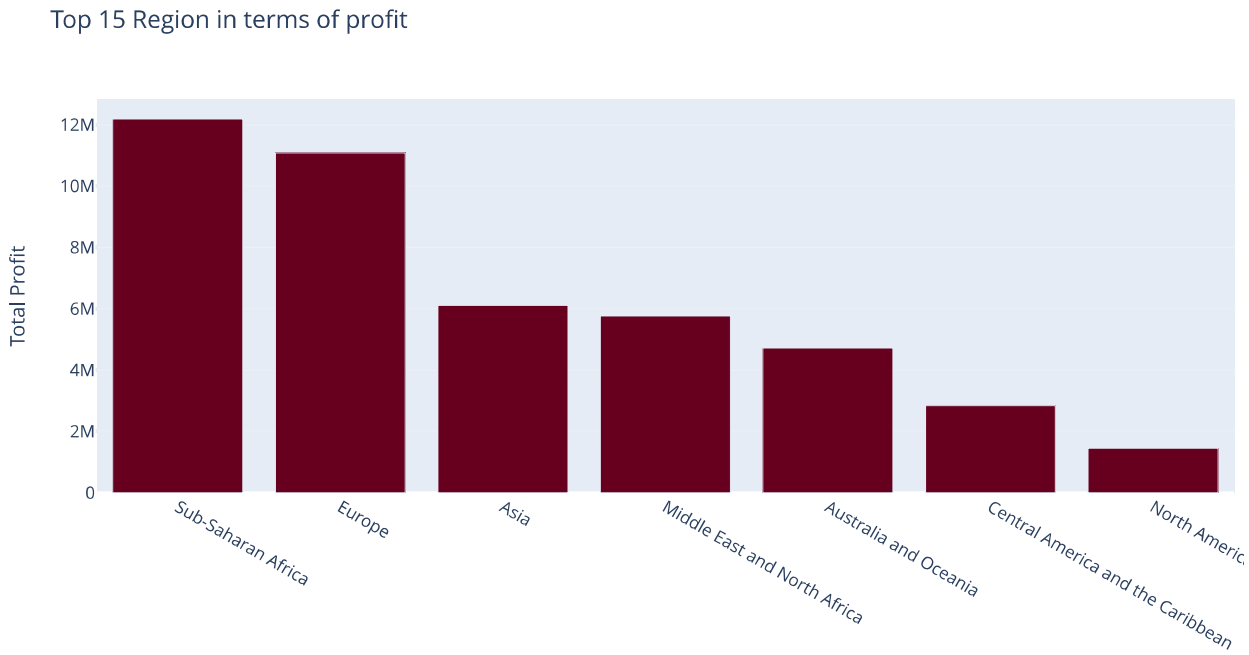
In [17]:
```
# Group by Country and sum the profits

profit_by_Region = df.groupby('Region')['Total Profit'].sum()
profit_by_Region
```

Out[17]:
```
Region
Asia                                6113845.87
Australia and Oceania               4722160.03
Central America and the Caribbean   2846907.85
Europe                             11082938.63
Middle East and North Africa        5761191.86
North America                       1457942.76
Sub-Saharan Africa                 12183211.40
Name: Total Profit, dtype: float64
```

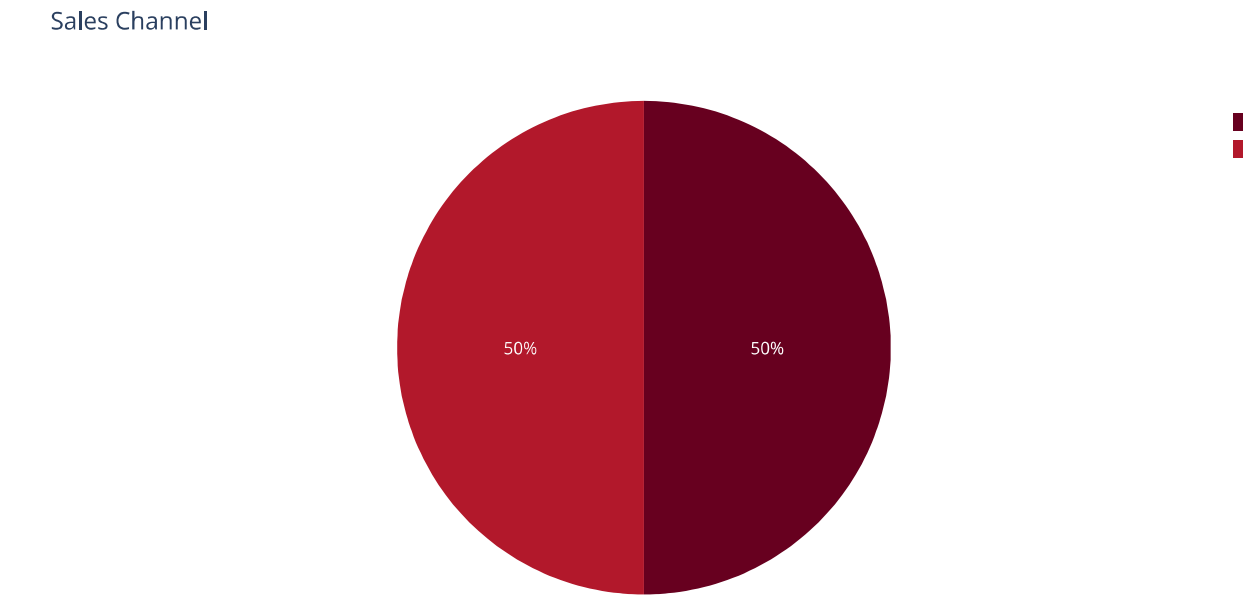
In [18]:
```
# Sort the values in descending order

Region_max_profit_sorted = profit_by_Region.sort_values(ascending=False)
Region_max_profit_sorted
```

Out[18]:
```
Region
Sub-Saharan Africa                 12183211.40
Europe                             11082938.63
Asia                                6113845.87
Middle East and North Africa        5761191.86
Australia and Oceania               4722160.03
Central America and the Caribbean   2846907.85
North America                       1457942.76
Name: Total Profit, dtype: float64
```

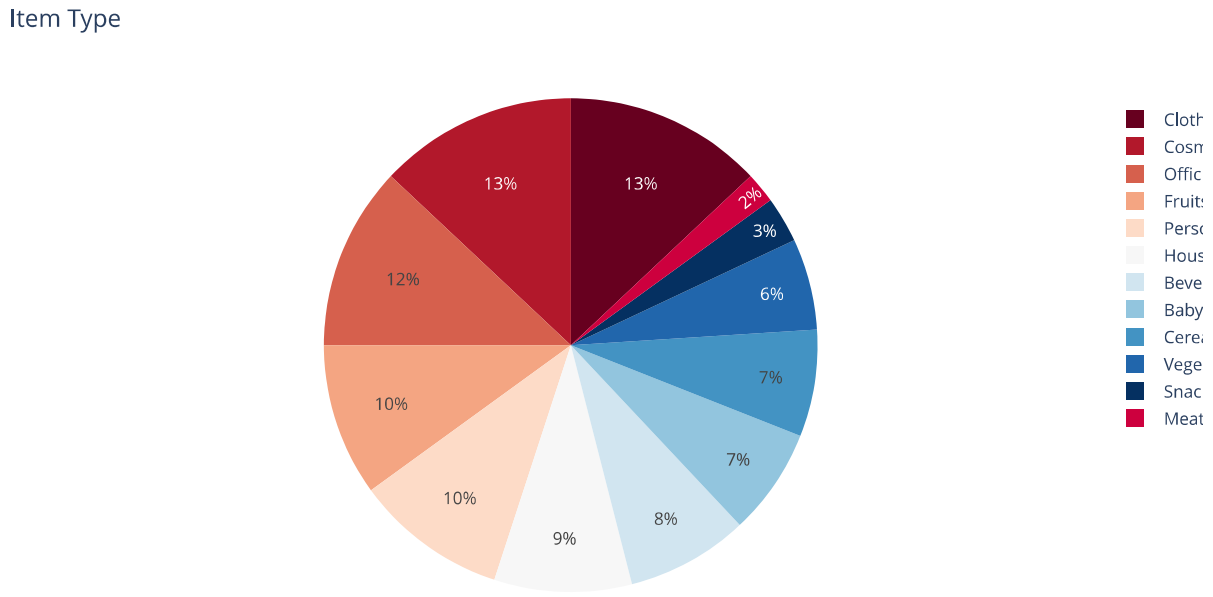
In [19]:
```
top_region_profit= df.groupby("Region").sum().sort_values("Total Profit",ascending=False).head(15)
top_region_profit= top_region_profit[['Total Profit']].round(2)
top_region_profit.reset_index(inplace=True)
top_region_profit

fig=px.bar(top_region_profit,x='Region',y='Total Profit',title="Top 15 Region in terms of profit",color_discrete_se
fig.show()
```



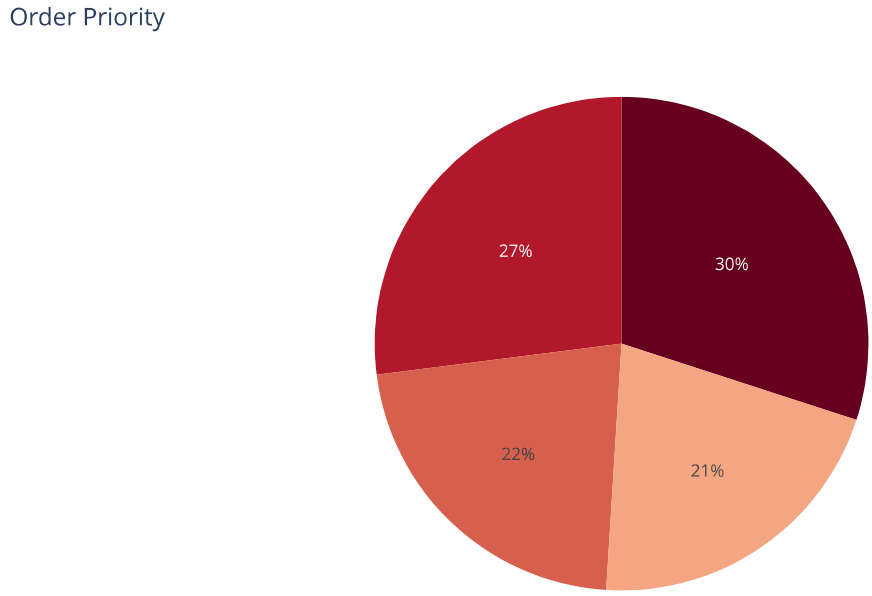
In [20]:
```
fig = px.pie(df,values=np.ones(100), names='Sales Channel', title='Sales Channel',color_discrete_sequence=px.colors
fig.show()
```



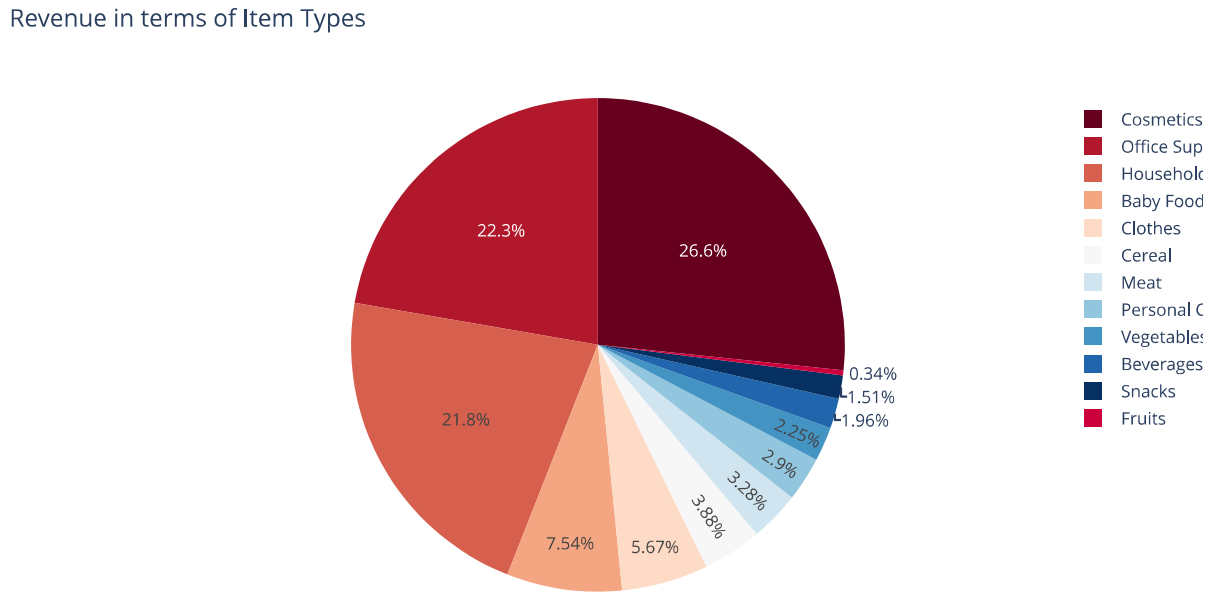
In [21]:
```
fig = px.pie(df,values=np.ones(100), names='Item Type', title='Item Type',color_discrete_sequence=px.colors.sequent
fig.show()
```



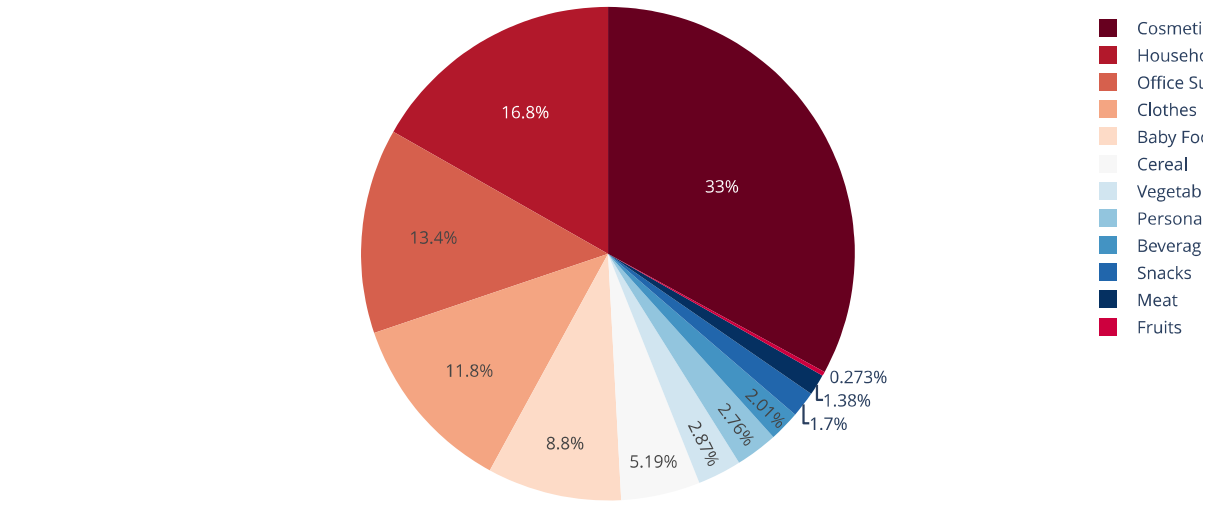
In [22]:
```
fig = px.pie(df,values=np.ones(100), names='Order Priority', title='Order Priority',color_discrete_sequence=px.colo
fig.show()
```



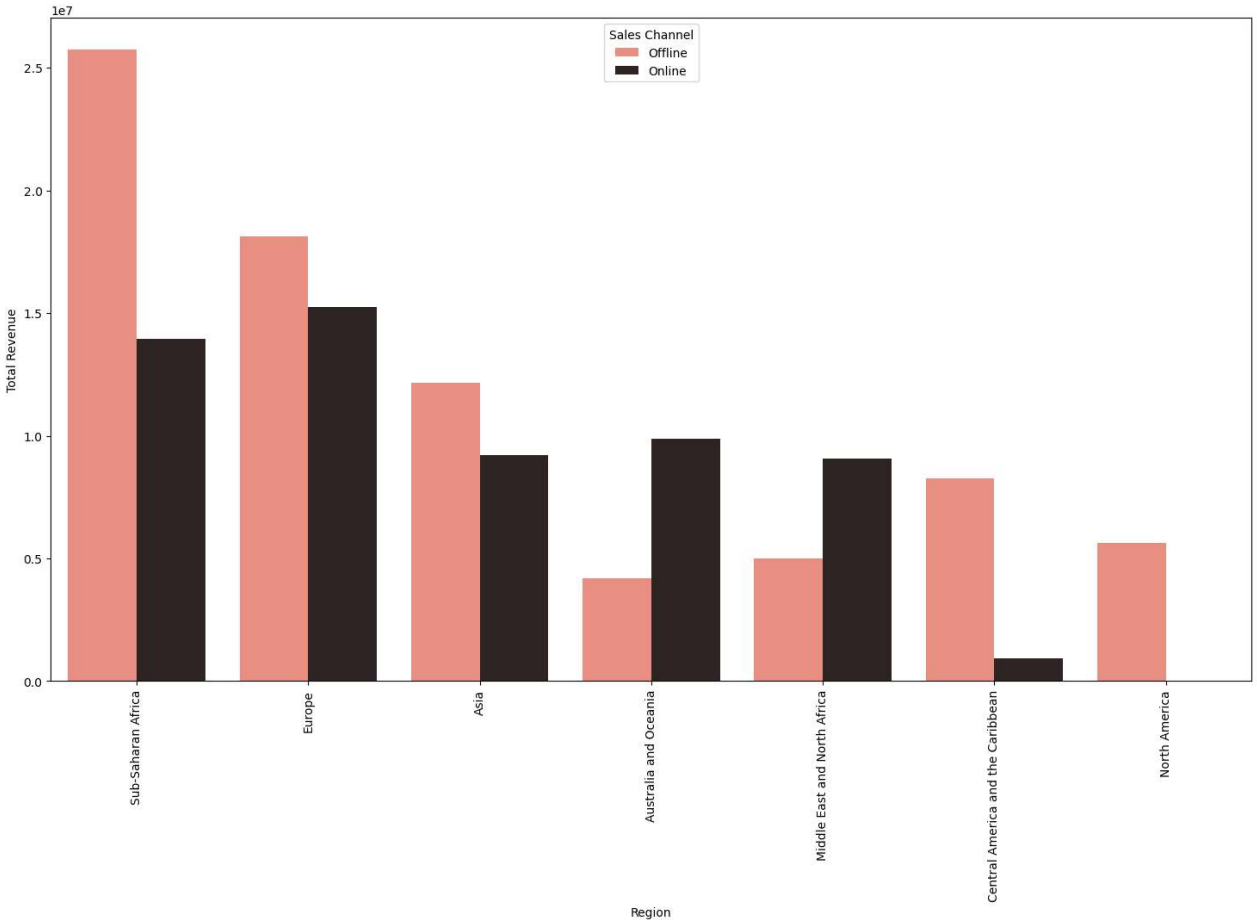
In [23]:
```
Revenue_item= df.groupby("Item Type").sum().sort_values("Item Type",ascending=False)
Revenue_item= Revenue_item[['Total Revenue']].round(2)
Revenue_item.reset_index(inplace=True)

fig=px.pie(Revenue_item,names='Item Type',values='Total Revenue',title="Revenue in terms of Item Types",color_discr
fig.show()
```



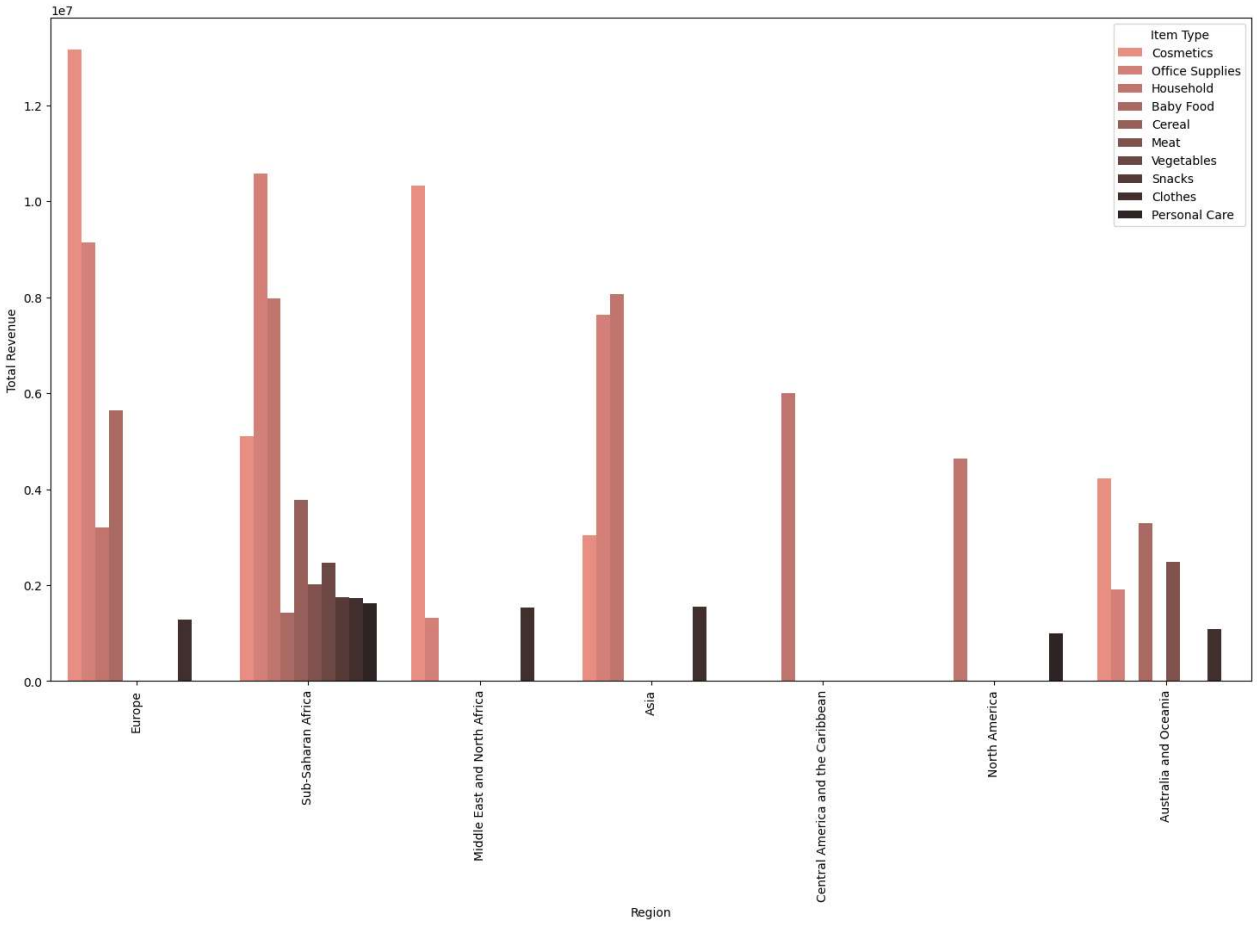
In [24]:
```
Revenue_item= df.groupby("Item Type").sum().sort_values("Item Type",ascending=False)
Revenue_item= Revenue_item[['Total Profit']].round(2)
Revenue_item.reset_index(inplace=True)

fig=px.pie(Revenue_item,names='Item Type',values='Total Profit',title="Profit in terms of Item Types",color_discrete
fig.show()
```



In [25]:
```
plt.figure(figsize=(18,10))
regions_sales_revenue= df.groupby(["Region","Sales Channel"]).sum().sort_values("Total Revenue",ascending=False).hea
regions_sales_revenue= regions_sales_revenue[['Total Revenue']].round(2)
regions_sales_revenue.reset_index(inplace=True)
sns.barplot(x='Region',y='Total Revenue',hue='Sales Channel',data=regions_sales_revenue,palette='dark:salmon_r')
plt.xticks(rotation='vertical')
plt.show()
```

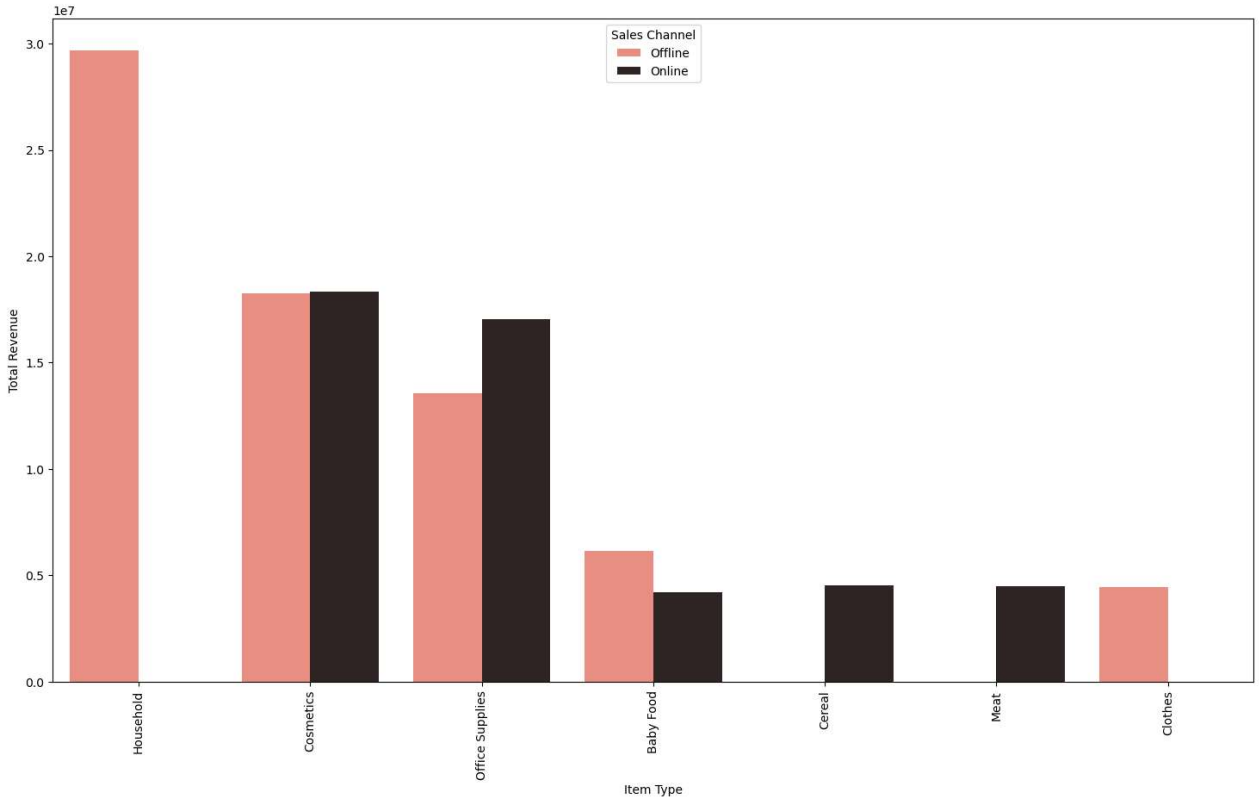


In [26]:
```
plt.figure(figsize=(18,10))
regions_sales_revenue= df.groupby(["Region","Item Type"]).sum().sort_values("Total Revenue",ascending=False).head(3
regions_sales_revenue= regions_sales_revenue[['Total Revenue']].round(2)
regions_sales_revenue.reset_index(inplace=True)
sns.barplot(x='Region',y='Total Revenue',hue='Item Type',data=regions_sales_revenue,palette='dark:salmon_r')
plt.xticks(rotation='vertical')
plt.show()
```



As we see in the above figure the most common item types in every region which help in markting in the common product for each region.

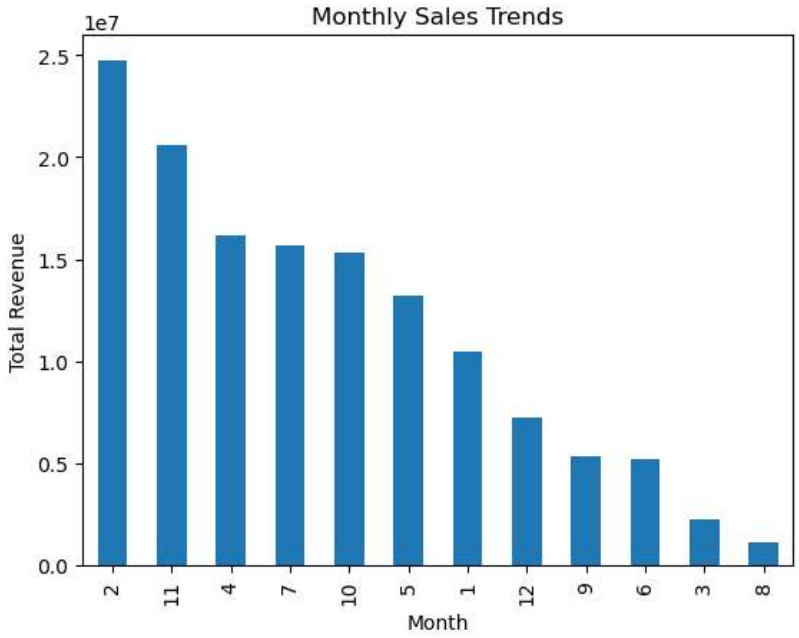
In [27]:
```
plt.figure(figsize=(18,10))
item_sales_revenue= df.groupby(["Item Type","Sales Channel"]).sum().sort_values("Total Revenue",ascending=False).hea
item_sales_revenue= item_sales_revenue[['Total Revenue']].round(2)
item_sales_revenue.reset_index(inplace=True)
sns.barplot(x='Item Type',y='Total Revenue',hue='Sales Channel',data=item_sales_revenue,palette='dark:salmon_r')
plt.xticks(rotation='vertical')
plt.show()
```



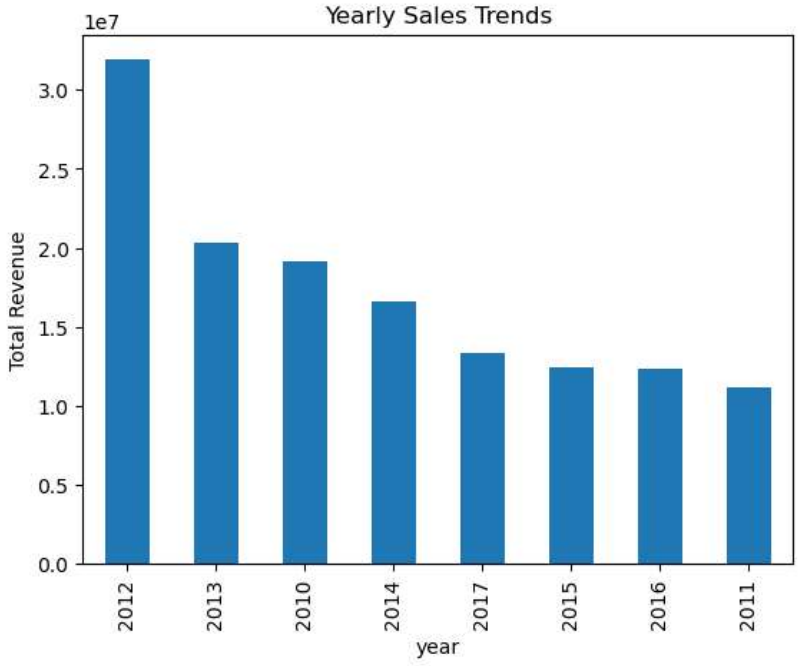
In the above figure we can see the most sales channel in every item types .

In [28]:
```
# Convert the 'date' column to datetime format
df['Order Date']=pd.to_datetime(df['Order Date'])

# Extract month and year
df['month'] = df['Order Date'].dt.month
df['year'] = df['Order Date'].dt.year
df['year_month'] = df['Order Date'].dt.to_period('M')
```

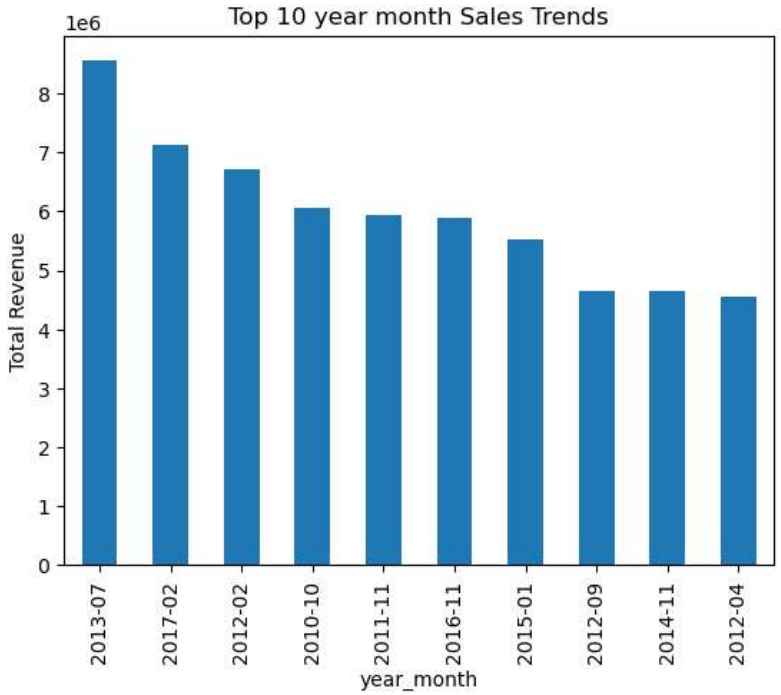
In [29]:
```
monthly_sales = df.groupby('month')['Total Revenue'].sum().sort_values(ascending=False)
monthly_sales.plot(kind='bar', xlabel='Month', ylabel='Total Revenue', title='Monthly Sales Trends')
plt.show()
```



In [30]:
```
yearly_sales = df.groupby('year')['Total Revenue'].sum().sort_values(ascending=False)
yearly_sales.plot(kind='bar', xlabel='year', ylabel='Total Revenue', title='Yearly Sales Trends')
plt.show()
```



In [31]:
```
year_month_sales = df.groupby('year_month')['Total Revenue'].sum().sort_values(ascending=False).head(10)
year_month_sales.plot(kind='bar', xlabel='year_month', ylabel='Total Revenue', title='Top 10 year month Sales Trends
plt.show()
```



Findings :-

- Region - The highest Revenue & Profit both generated are from Sub Saharan Africa.
- Country - The highest Revenue generated are from Honduras and highest profit generated are from Djibouti.
- Channels :
  - There is equal number of orders from both online & offine channels.
  - The most sale items in offline channel is household and The most sale items in online channel is Cosmetics.
  - Sub Saharan Africa has highest revenue in terms of offline sales & Europe has highest revenue in terms of online sales.
- Item Types - The highest Revenue & Profit both generated are from cosmetics.
- The most common sold items are cosmetics , households and office supplies.
- Monthly sales trend - Most numbers of sales is from febuary month.
- Yearly sales trend - 2012 has the most numbers of sales.
- year month Sales Trends - 2013 july has recorded as the most number of sales.