In [1]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
import datetime as dt
```

In [2]:
```
data = pd.read_csv('superstore.csv')
data.head()
```

Out[2]:

	Unnamed: 0	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	State	...	Product Name	Sales	Quantity	Discount	Profit	Shipping Cost	Order Priority	Delivery Days	order year	order month
0	0	CA-2014-AB10015140-41954	11-11-2014	13-11-2014	First Class	AB-100151402	Aaron Bergman	Consumer	Oklahoma City	Oklahoma	...	Samsung Convoy 3	221	2	0.0	62	40	High	2	2014	11
1	1	IN-2014-JR162107-41675	05-02-2014	07-02-2014	Second Class	JR-162107	Justin Ritter	Corporate	Wollongong	New South Wales	...	Novimex Executive Leather Armchair, Black	3709	9	0.1	-288	923	Critical	2	2014	2
2	2	IN-2014-CR127307-41929	17-10-2014	18-10-2014	First Class	CR-127307	Craig Reiter	Consumer	Brisbane	Queensland	...	Nokia Smart Phone, with Caller ID	5175	9	0.1	919	915	Medium	1	2014	10
3	3	ES-2014-KM1637548-41667	28-01-2014	30-01-2014	First Class	KM-1637548	Katherine Murray	Home Office	Berlin	Berlin	...	Motorola Smart Phone, Cordless	2892	5	0.1	-96	910	Medium	2	2014	1
4	4	SG-2014-RH9495111-41948	05-11-2014	06-11-2014	Same Day	RH-9495111	Rick Hansen	Consumer	Dakar	Dakar	...	Sharp Wireless Fax, High-Speed	2832	8	0.0	311	903	Critical	1	2014	11

In [3]:
```
data[['order_day','order_month','order_year']] = data['Order Date'].str.split('-', expand=True)
data['Order Date'] = data['order_year'] + '/' + data['order_month'] + '/' + data['order_day']
data['Order Date'] = pd.to_datetime(data['Order Date'])
```

In [4]:
```
data[['ship_day','ship_month','ship_year']] = data['Ship Date'].str.split('-', expand=True)
data['Ship Date'] = data['ship_year'] + '/' + data['ship_month'] + '/' + data['ship_day']
data['Ship Date'] = pd.to_datetime(data['Ship Date'])
```

In [5]:
```
data.drop(columns=['order_day','order_month','order_year',
                   'ship_day','ship_month','ship_year','Unnamed: 0'], inplace=True)
```

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Order ID        51290 non-null  object
 1   Order Date      51290 non-null  datetime64[ns]
 2   Ship Date       51290 non-null  datetime64[ns]
 3   Ship Mode       51290 non-null  object
 4   Customer ID     51290 non-null  object
 5   Customer Name   51290 non-null  object
 6   Segment         51290 non-null  object
 7   City            51290 non-null  object
 8   State           51290 non-null  object
 9   Country         51290 non-null  object
 10  Region          51290 non-null  object
 11  Market          51290 non-null  object
 12  Product ID      51290 non-null  object
 13  Category        51290 non-null  object
 14  Sub-Category    51290 non-null  object
 15  Product Name    51290 non-null  object
 16  Sales           51290 non-null  int64
 17  Quantity        51290 non-null  int64
 18  Discount        51290 non-null  float64
 19  Profit          51290 non-null  int64
 20  Shipping Cost   51290 non-null  int64
 21  Order Priority  51290 non-null  object
 22  Delivery Days   51290 non-null  int64
 23  order year      51290 non-null  int64
 24  order month     51290 non-null  int64
dtypes: datetime64[ns](2), float64(1), int64(7), object(15)
memory usage: 9.8+ MB
```

In [7]:
```
data.nunique()
```

Out[7]:
```
Order ID          25728
Order Date         1430
Ship Date          1464
Ship Mode             4
Customer ID       17415
Customer Name       796
Segment               3
City               3650
State              1102
Country             165
Region               23
Market                5
Product ID         3788
Category              3
Sub-Category         17
Product Name       3788
Sales              2259
Quantity             14
Discount             27
Profit             1604
Shipping Cost       544
Order Priority        4
Delivery Days         8
order year            4
order month          12
dtype: int64
```

In [8]:
```
data['Ship Mode'] = data['Ship Mode'].astype('category')
data['Segment'] = data['Segment'].astype('category')
data['Country'] = data['Country'].astype('category')
data['Market'] = data['Market'].astype('category')
data['Region'] = data['Region'].astype('category')
data['Category'] = data['Category'].astype('category')
data['Sub-Category'] = data['Sub-Category'].astype('category')
data['Order Priority'] = data['Order Priority'].astype('category')
```

In [9]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Order ID        51290 non-null  object
 1   Order Date      51290 non-null  datetime64[ns]
 2   Ship Date       51290 non-null  datetime64[ns]
 3   Ship Mode       51290 non-null  category
 4   Customer ID     51290 non-null  object
 5   Customer Name   51290 non-null  object
 6   Segment         51290 non-null  category
 7   City            51290 non-null  object
 8   State           51290 non-null  object
 9   Country         51290 non-null  category
 10  Region          51290 non-null  category
 11  Market          51290 non-null  category
 12  Product ID      51290 non-null  object
 13  Category        51290 non-null  category
 14  Sub-Category    51290 non-null  category
 15  Product Name    51290 non-null  object
 16  Sales           51290 non-null  int64
 17  Quantity        51290 non-null  int64
 18  Discount        51290 non-null  float64
 19  Profit          51290 non-null  int64
 20  Shipping Cost   51290 non-null  int64
 21  Order Priority  51290 non-null  category
 22  Delivery Days   51290 non-null  int64
 23  order year      51290 non-null  int64
 24  order month     51290 non-null  int64
dtypes: category(8), datetime64[ns](2), float64(1), int64(7), object(7)
memory usage: 7.1+ MB
```

In [10]:
```
def removespaces(df):
    for cols in df.columns:
        if df[cols].dtypes in ['object','category']:
            df[cols] = df[cols].str.strip()
        return df
```

In [11]:
```
data = removespaces(data)
```

In [12]:
```
data.head()
```

Out[12]:

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	State	Country	...	Product Name	Sales	Quantity	Discount	Profit	Shipping Cost	Order Priority	Delivery Days	order year	order month
0	CA-2014-AB10015140-41954	2014-11-11	2014-11-13	First Class	AB-100151402	Aaron Bergman	Consumer	Oklahoma City	Oklahoma	United States	...	Samsung Convoy 3	221	2	0.0	62	40	High	2	2014	11
1	IN-2014-JR162107-41675	2014-02-05	2014-02-07	Second Class	JR-162107	Justin Ritter	Corporate	Wollongong	New South Wales	Australia	...	Novimex Executive Leather Armchair, Black	3709	9	0.1	-288	923	Critical	2	2014	2
2	IN-2014-CR127307-41929	2014-10-17	2014-10-18	First Class	CR-127307	Craig Reiter	Consumer	Brisbane	Queensland	Australia	...	Nokia Smart Phone, with Caller ID	5175	9	0.1	919	915	Medium	1	2014	10
3	ES-2014-KM1637548-41667	2014-01-28	2014-01-30	First Class	KM-1637548	Katherine Murray	Home Office	Berlin	Berlin	Germany	...	Motorola Smart Phone, Cordless	2892	5	0.1	-96	910	Medium	2	2014	1
4	SG-2014-RH9495111-41948	2014-11-05	2014-11-06	Same Day	RH-9495111	Rick Hansen	Consumer	Dakar	Dakar	Senegal	...	Sharp Wireless Fax, High-Speed	2832	8	0.0	311	903	Critical	1	2014	11

5 rows × 25 columns

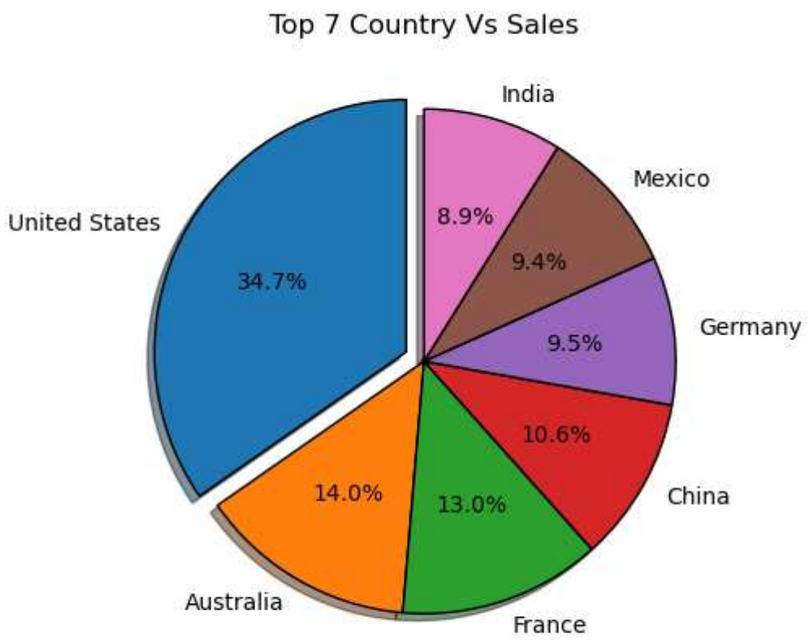
Top 7 Countires And There Total Sales

In [13]:
```
country_group = data.groupby('Country')
country_sales = country_group.agg({'Sales':'sum'})
country_sales.sort_values(by='Sales', ascending=False)
top_7 = country_sales.nlargest(7, 'Sales')
top_7.reset_index()
```

Out[13]:

	Country	Sales
0	United States	2291304
1	Australia	923807
2	France	857526
3	China	699613
4	Germany	627112
5	Mexico	620277
6	India	588711

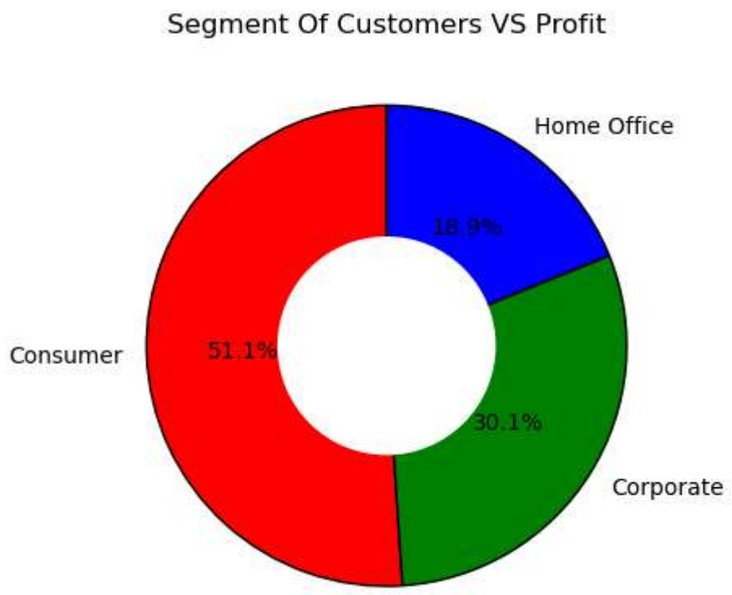
In [14]:
```
plt.figure(figsize=(5,8))
explode = [0.08, 0, 0, 0, 0, 0, 0]
plt.pie(top_7['Sales'],labels=list(top_7.index), explode=explode, shadow=True,
        startangle=90, autopct='%1.1f%%', wedgeprops={'edgecolor': 'black'})
plt.title("Top 7 Country Vs Sales");
```



Segment Of Customers VS Profit

In [15]:
```
cust_Seg = data.groupby('Segment')
df = cust_Seg.aggregate({'Profit':'sum'})
df.reset_index(inplace=True)
```

In [16]:
```
names = df['Segment'].values
marks = df['Profit'].values
my_circle = plt.Circle((0, 0), 0.45, color='white')
plt.pie(marks, labels=names, autopct='%1.1f%%',colors=['red', 'green', 'blue']
       , wedgeprops={'edgecolor': 'black'},startangle=90)
p = plt.gcf()
p.gca().add_artist(my_circle)
plt.title('Segment Of Customers VS Profit');
```

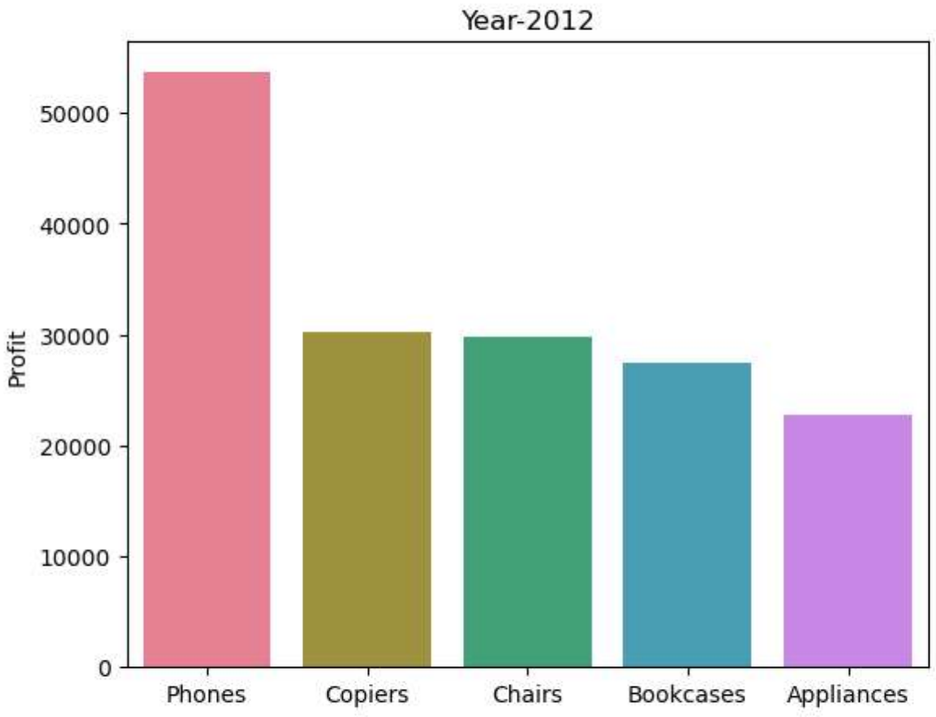


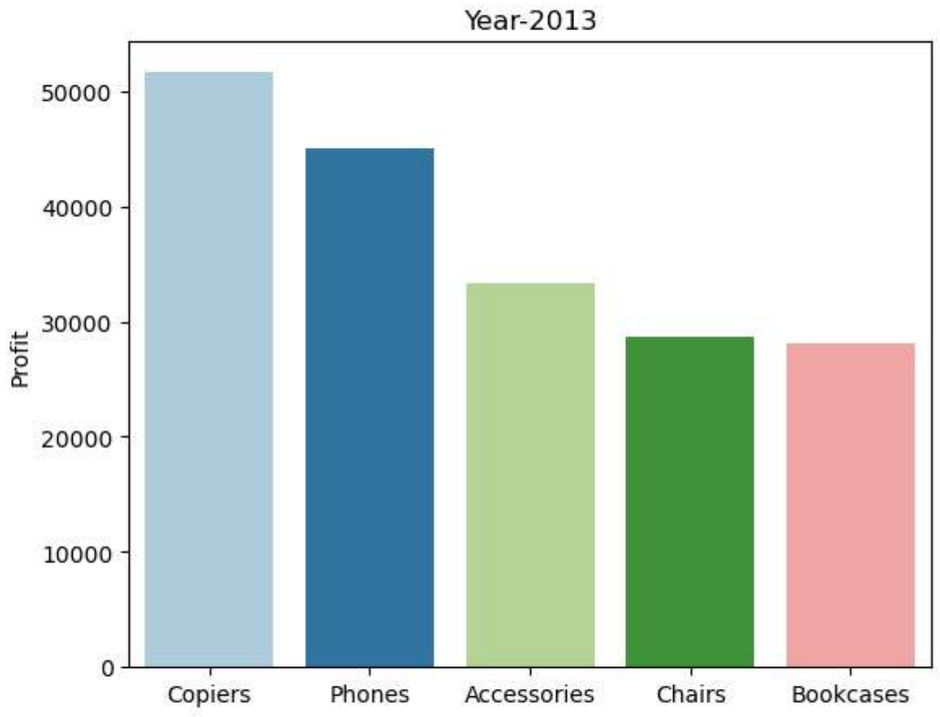
Top 5 Profit Making Product Types On Yearly Basis

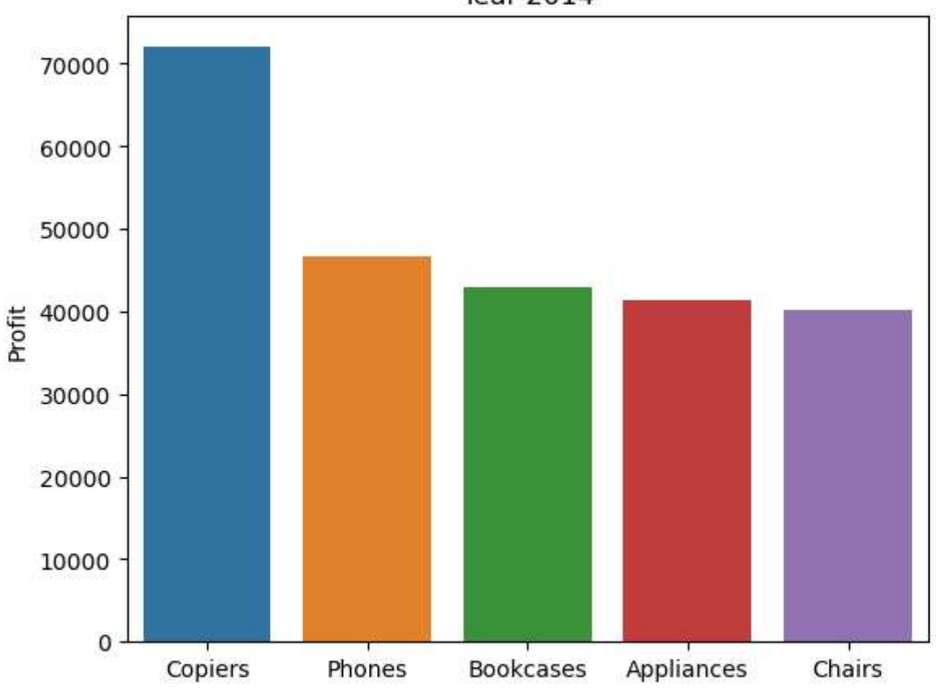
In [17]:
```
year_category_group = data.groupby(['order year','Sub-Category'])
year_category_proft_df = year_category_group.agg({'Profit':'sum'})
year_category_proft_df.reset_index(inplace=True)
category_yearly_profit = year_category_proft_df.groupby('order year')
top5_profit_category = pd.DataFrame(columns=year_category_proft_df.columns)

for g, d in category_yearly_profit:
    high_profit_categories = d.nlargest(5, 'Profit')
    top5_profit_category = pd.concat([top5_profit_category,high_profit_categories])
```

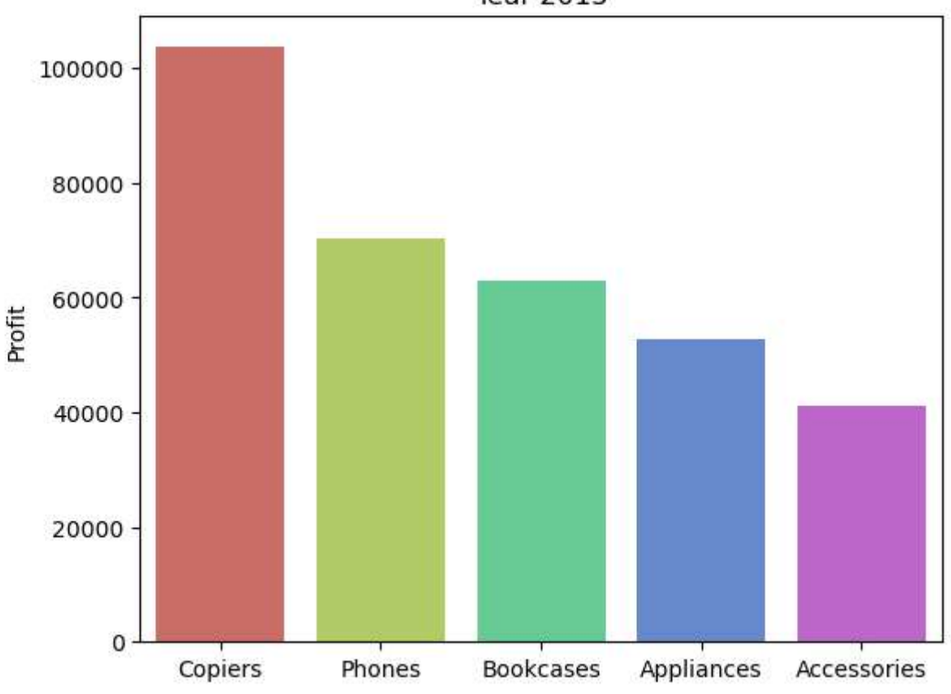
In [18]:
```
plt.figure(figsize=(14,11))

plt.subplot(2, 2, 1)
x=list(top5_profit_category[top5_profit_category['order year'] == 2012]['Sub-Category'])
y=top5_profit_category[top5_profit_category['order year'] == 2012]['Profit']
sns.barplot(x=x,y=y,palette='husl')
plt.title("Year-2012")

plt.subplot(2, 2, 2)
x=list(top5_profit_category[top5_profit_category['order year'] == 2013]['Sub-Category'])
y=top5_profit_category[top5_profit_category['order year'] == 2013]['Profit']
sns.barplot(x=x,y=y,palette='Paired')
plt.title("Year-2013")

plt.subplot(2, 2, 3)
x=list(top5_profit_category[top5_profit_category['order year'] == 2014]['Sub-Category'])
y=top5_profit_category[top5_profit_category['order year'] == 2014]['Profit']
sns.barplot(x=x,y=y)
plt.title("Year-2014")

plt.subplot(2, 2, 4)
x=list(top5_profit_category[top5_profit_category['order year'] == 2015]['Sub-Category'])
y=top5_profit_category[top5_profit_category['order year'] == 2015]['Profit']
sns.barplot(x=x,y=y, palette='hls')
plt.title("Year-2015")

plt.suptitle("Top 5 Profit Making Products For Each Year")
plt.show()
```

Top 5 Profit Making Products For Each Year









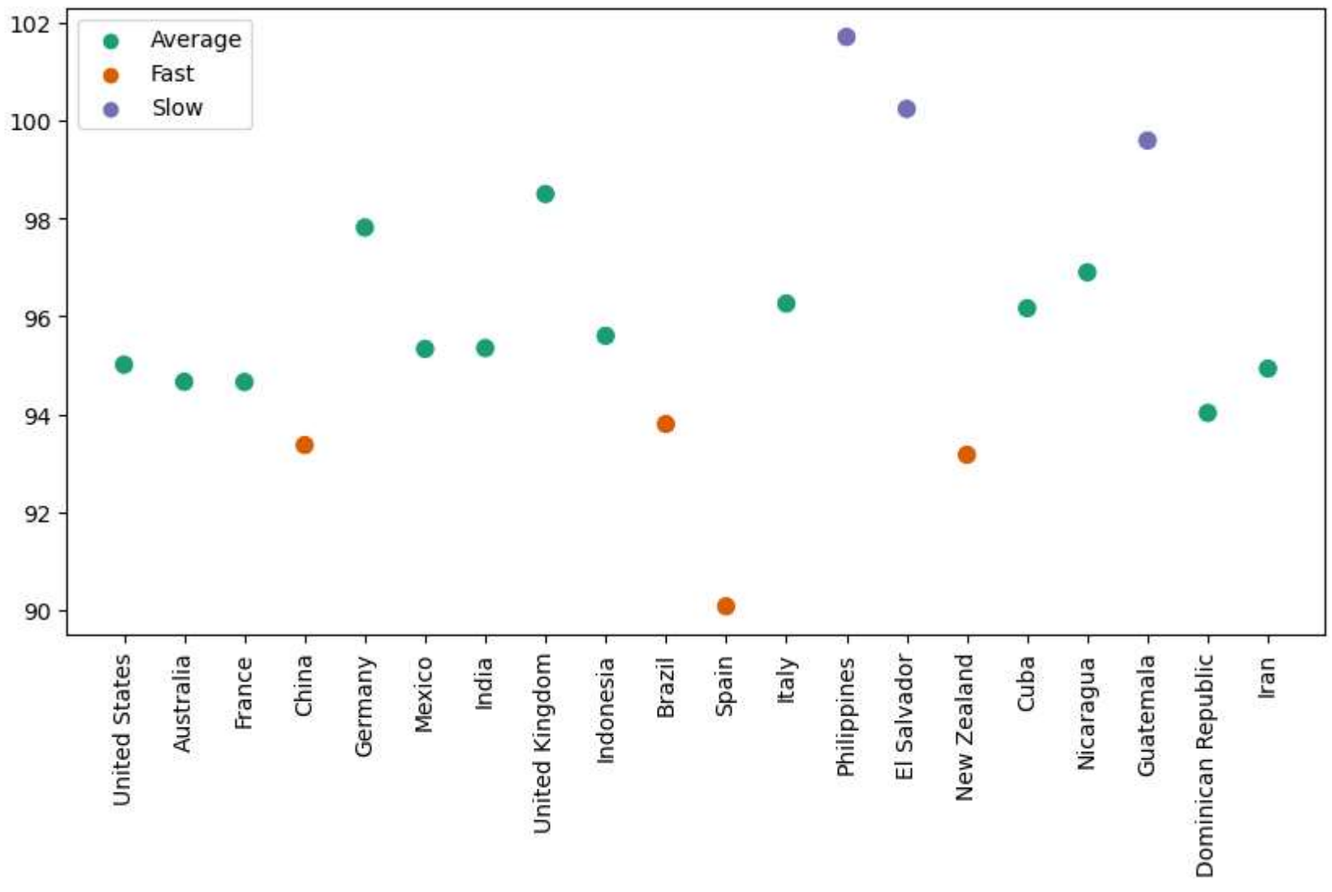
Delivery Speeds Of Top 20 Countries

In [19]:
```
top_20_sales = country_sales.nlargest(20, 'Sales')
data['Delivery Duration'] = data['Ship Date']-data['Order Date']
country_group = data.groupby('Country')
delivery_duration_df = country_group.agg({'Delivery Duration':'mean'})
delivery_duration_df['Duration In Hours'] = delivery_duration_df['Delivery Duration'] / dt.timedelta(hours=1)
```

In [20]:
```
top20_sales_country_DD =top_20_sales.merge(delivery_duration_df, how='left', left_index=True, right_index=True)
top20_sales_country_DD.reset_index(inplace=True)
top20_sales_country_DD.sort_values(by='Duration In Hours');
```

In [21]:
```
labels = []
for time in list(top20_sales_country_DD['Duration In Hours']):
    if 83 <= time <= 94: labels.append('Fast')
    elif 94 <= time <= 99: labels.append('Average')
    else: labels.append('Slow')
```

In [22]:
```
plt.figure(figsize=(10,5))
x = list(top20_sales_country_DD['Country'])
y = list(top20_sales_country_DD['Duration In Hours'])
sns.scatterplot(x=x,y=y,s=80,hue=labels,palette='Dark2')
plt. legend(loc='upper left')
plt.xticks(rotation=90);
```

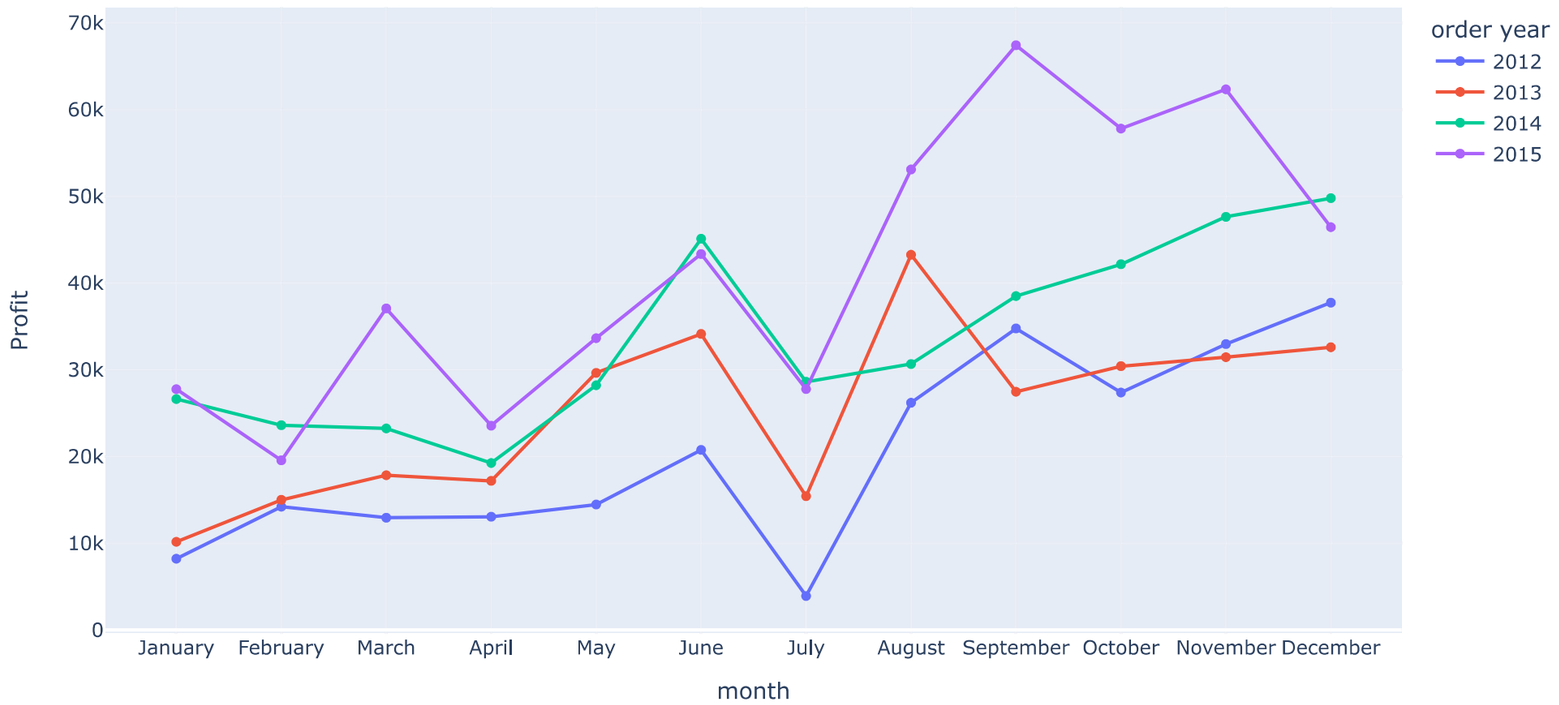


Profit Vs Month

In [23]:
```
df_month = data.groupby(['order month','order year'])
df_pro = df_month.aggregate({'Profit':'sum'})
df_sal = df_month.aggregate({'Sales':'sum'})
df_pro.reset_index(inplace=True)
df_sal.reset_index(inplace=True)
```

In [24]:
```
df =df_pro.merge(df_sal)
df['month'] = df['order month'].map({1:'January',2:'February',3:'March',4:'April',5:'May',6:'June',7:'July',
                                     8:'August',9:'September',10:'October',11:'November',12:'December'})
```

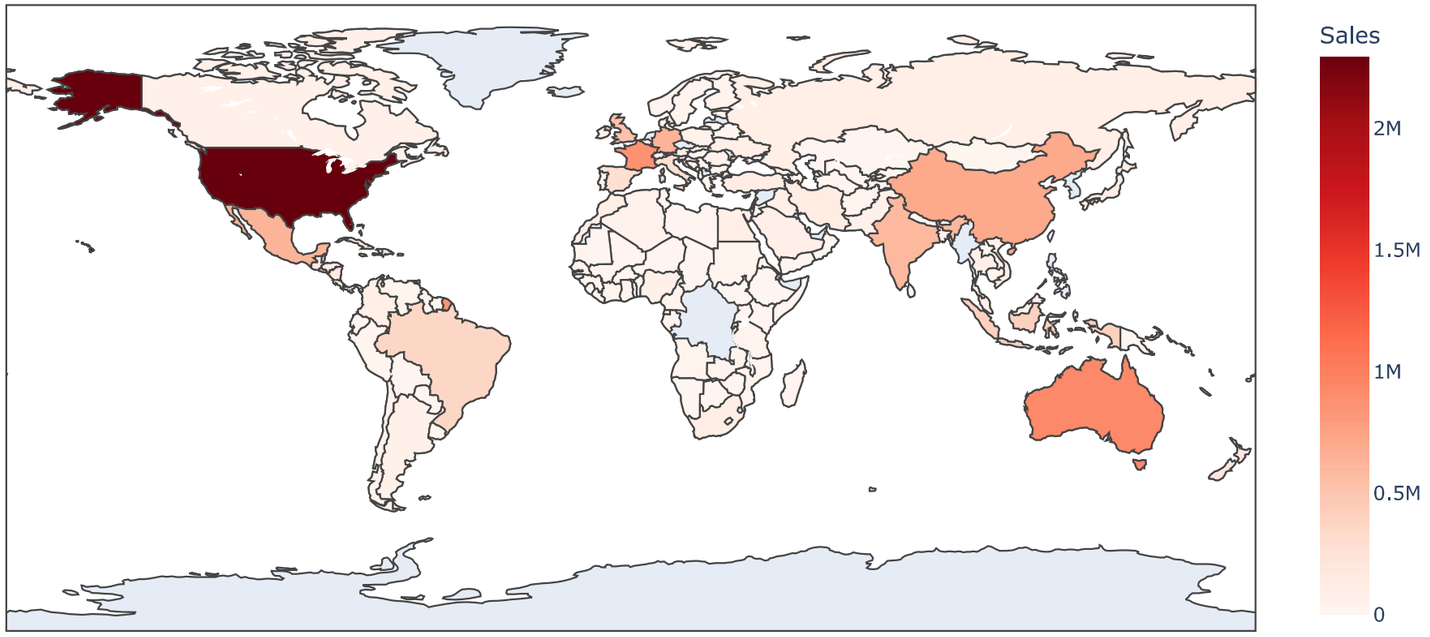
In [25]:
```
fig = px.line(df, x="month", y="Profit", color='order year',markers=True)
fig.show()
```



Countries VS Sales

In [26]:
```
iso_mapping = {'Afghanistan': 'AFG', 'Akrotiri and Dhekelia – See United Kingdom, The': 'Akrotiri and Dhekelia – See United Kingdom, The', 'Åland Islands': 'ALA', 'Albania': 'ALB', 'Algeri
```

In [27]:
```
import plotly.express as px
df = country_sales.reset_index()
df['ISO Code'] = df['Country'].map(iso_mapping)
fig = px.choropleth(df, locations="ISO Code",
                    color="Sales",
                    hover_name="Country",
                    color_continuous_scale='Reds')
fig.show()
```



In [28]:
```
India_Data = data[data['Country']=='India']
India_Data.head()
```

Out[28]:

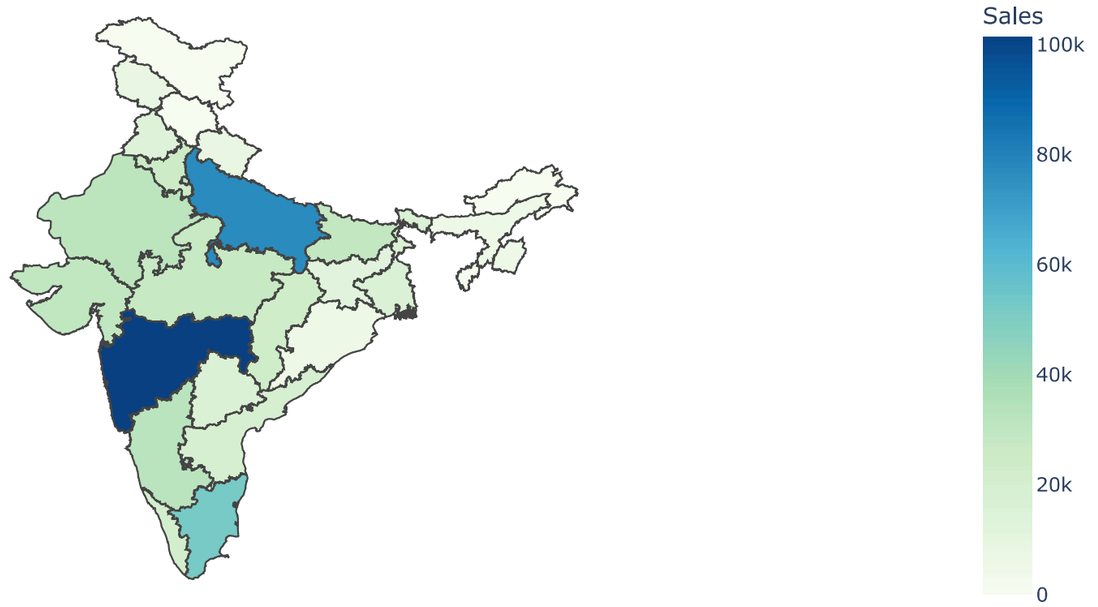
	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	State	Country	...	Sales	Quantity	Discount	Profit	Shipping Cost	Order Priority	Delivery Days	order year	order month	Delivery Duration
29	IN-2013-BP1123058-41329	2013-02-24	2013-02-24	Same Day	BP-1123058	Benjamin Patterson	Consumer	Surat	Gujarat	India	...	1878	4	0.0	582	704	Critical	0	2013	2	0 days
41	IN-2015-BF1100558-42319	2015-11-11	2015-11-15	Standard Class	BF-1100558	Barry Franz	Home Office	Gorakhpur	Haryana	India	...	4518	7	0.0	632	658	High	4	2015	11	4 days
42	IN-2015-VG2180558-42273	2015-09-26	2015-09-28	Second Class	VG-2180558	Vivek Grady	Corporate	Thiruvananthapuram	Kerala	India	...	5667	13	0.0	2097	658	Medium	2	2015	9	2 days
48	IN-2015-SW2027558-42125	2015-05-01	2015-05-01	Same Day	SW-2027558	Scott Williamson	Consumer	Jamshedpur	Jharkhand	India	...	2174	7	0.0	500	637	Critical	0	2015	5	0 days
55	IN-2013-SG2047058-41424	2013-05-30	2013-05-31	First Class	SG-2047058	Sheri Gordon	Consumer	Bhopal	Madhya Pradesh	India	...	1526	4	0.0	732	625	Critical	1	2013	5	1 days

5 rows × 26 columns

Indian States And Sales

In [29]:
```
India_by_states = India_Data.groupby('State')
df = India_by_states.agg({'Sales':'sum'})
df.reset_index(inplace=True)
df.loc[len(df.index)] = ['Arunachal Pradesh', 0]
df.loc[len(df.index)] = ['Ladakh', 0]
df.loc[len(df.index)] = ['Himachal Pradesh', 0]
```

In [30]:
```
fig = px.choropleth(
    df,
    geojson='indian_states.json',
    featureidkey='properties.ST_NM',
    locations='State',
    color='Sales',
    color_continuous_scale='gnbu'
)

fig.update_geos(fitbounds="locations", visible=False)
fig.show()
```



Indian States And Delivery Speeds

In [31]:
```
India = India_Data.copy()
India['Delivery Duration'] = India['Ship Date']-India['Order Date']
country_grp = India.groupby('State')
delivery_df = country_grp.agg({'Delivery Duration':'mean'})
delivery_df['Duration In Hours'] = delivery_df['Delivery Duration'] / dt.timedelta(hours=1)
delivery_df.reset_index(inplace=True)
delivery_df.loc[len(delivery_df.index)] = ['Arunachal Pradesh', 0,0]
delivery_df.loc[len(delivery_df.index)] = ['Ladakh', 0,0]
delivery_df.loc[len(delivery_df.index)] = ['Himachal Pradesh', 0,0]
delivery_df['Duration In Days'] = delivery_df['Duration In Hours'] // 24
```

In [32]:
```
fig = px.choropleth(
    delivery_df,
    geojson='indian_states.json',
    featureidkey='properties.ST_NM',
    locations='State',
    color='Duration In Days',
    color_continuous_scale='blackbody_r'
)

fig.update_geos(fitbounds="locations", visible=False)
fig.show()
```