Created By:
Anjali Keshri

SALES REPORT

(Python,Excel And Power BI)

Introduction

Welcome to our Sales Report Presentation. This sales report provides a comprehensive analysis of regional sales performance across different states and customer segments. The report highlights key metrics such as total sales, order quantity, average sales per order, and sales trends over time. Visualizations have been used to identify top-performing regions, customer behavior, and the relationship between income and purchase patterns. This analysis aims to support data-driven decisions and uncover opportunities for market growth and optimization.

Executive Summary

This sales report analyzes regional performance from 2014 to 2018. Total sales reached ₹1.24 billion from over 64,000 orders and 541,000+ units sold. The West region led in sales, while the Northeast lagged behind. January saw the highest sales, indicating a strong seasonal trend. Some high-population states had lower sales, revealing untapped market potential. Key customers and sales channels were identified for strategic focus. The report highlights areas to optimize growth and improve regional performance.

Data Cleaning & Preprocessing

```
[49]: import pandas as pd
      import numpy as np
      import seaborn as sns
      import matplotlib.pyplot as plt
      import warnings
      warnings.filterwarnings("ignore")
```

```
[50]: data=pd.read_excel("D:/Data Analyst/End-To-End Projects/Complete Project/dataset/Regional Sales Dataset.xlsx",sheet_name=None)
```

```
[3]: data
```

```
[3]: {'Sales Orders':        OrderNumber  OrderDate  Customer Name Index      Channel  \
      0       SO - 000225 2014-01-01                  126    Wholesale
      1      SO - 0003378 2014-01-01                   96  Distributor
      2      SO - 0005126 2014-01-01                    8    Wholesale
      3      SO - 0005614 2014-01-01                   42       Export
      4      SO - 0005781 2014-01-01                   73    Wholesale
      ...             ...        ...                  ...          ...
      64099  SO - 0007573 2018-02-28                   74    Wholesale
      64100  SO - 0007706 2018-02-28                   51       Export
      64101  SO - 0007718 2018-02-28                  136  Distributor
      64102  SO - 0008084 2018-02-28                  158  Distributor
      64103  SO - 0008654 2018-02-28                   22  Distributor

            Currency Code Warehouse Code  Delivery Region Index  \
      0               USD          AXW291                    364
      1               USD          AXW291                    488
      2               USD          AXW291                    155
      3               USD          AXW291                    473
```

```
[4]: sales_order=data["Sales Orders"]
     customers=data["Customers"]
     products=data["Products"]
     regions=data["Regions"]
     state_reg=data["State Regions"]
     budgets=data["2017 Budgets"]
```

```
[5]: sales_order.head(2)
```

[5]:

	OrderNumber	OrderDate	Customer Name Index	Channel	Currency Code	Warehouse Code	Delivery Region Index	Product Description Index	Order Quantity	Unit Price	Line Total	Total Unit Cost
0	SO - 000225	2014-01-01	126	Wholesale	USD	AXW291	364	27	6	2499.1	14994.6	1824.343
1	SO - 0003378	2014-01-01	96	Distributor	USD	AXW291	488	20	11	2351.7	25868.7	1269.918

```
[6]: customers.head(2)
```

[6]:

	Customer Index	Customer Names
0	1	Geiss Company
1	2	Jaxbean Group

```
[7]: products.head(2)
```

[7]:

	Index	Product Name
0	1	Product 1
1	2	Product 2

```
[8]: regions.head(2)
```

[8]:

	id	name	county	state_code	state	type	latitude	longitude	area_code	population	households	median_income	land_area	water_area	
0	1	Auburn	Lee County	AL	Alabama	City	32.60986	-85.48078	334	62059	21767	38342	152375113	2646161	Ameri
1	2	Birmingham	Shelby County/Jefferson County	AL	Alabama	City	33.52744	-86.79905	205	212461	89972	31061	378353942	6591013	Ameri

```
[9]: state_reg.head()
```

[9]:

	Column1	Column2	Column3
0	State Code	State	Region
1	AL	Alabama	South
2	AR	Arkansas	South
3	AZ	Arizona	West
4	CA	California	West

```
[10]: new_header=state_reg.iloc[0]
      state_reg.columns=new_header
      state_reg=state_reg[1:].reset_index(drop=True)
      state_reg.head()
```

[10]:

	State Code	State	Region
0	AL	Alabama	South
1	AR	Arkansas	South
2	AZ	Arizona	West
3	CA	California	West
4	CO	Colorado	West

```
[11]: budgets.head(2)
```

[11]:

	Product Name	2017 Budgets
0	Product 1	3016489.209
1	Product 2	3050087.565

```
[12]: print("Sales Shape:",sales_order.shape)
      print("Customers Shape:",customers.shape)
      print("Products Shape:",products.shape)
      print("Regions Shape:",regions.shape)
      print("state Regions Shape:",state_reg.shape)
      print("budgets shape:",budgets.shape)
```

```
Sales Shape: (64104, 12)
Customers Shape: (175, 2)
Products Shape: (30, 2)
Regions Shape: (994, 15)
state Regions Shape: (48, 3)
budgets shape: (30, 2)
```

Check null values

```
[13]: print("Sales :\n",sales_order.isnull().sum())
      print("\nCustomers :\n",customers.isnull().sum())
      print("\nProducts :\n",products.isnull().sum())
      print("\nRegions :\n",regions.isnull().sum())
      print("\nstate Regions :\n",state_reg.isnull().sum())
      print("\nbudgets:\n",budgets.isnull().sum())
```

```
Sales :
 OrderNumber                0
OrderDate                   0
Customer Name Index         0
Channel                     0
Currency Code               0
Warehouse Code              0
Delivery Region Index       0
Product Description Index   0
Order Quantity              0
Unit Price                  0
Line Total                  0
Total Unit Cost             0
dtype: int64
```

```
Customers :
 Customer Index    0
Customer Names     0
dtype: int64

Products :
 Index             0
Product Name       0
dtype: int64

Regions :
 id                 0
name                0
county              0
state_code          0
state               0
type                0
latitude            0
longitude           0
area_code           0
population          0
households          0
median_income       0
land_area           0
water_area          0
time_zone           0
dtype: int64

state Regions :
 0
State Code     0
State          0
Region         0
dtype: int64

budgets:
 Product Name     0
2017 Budgets      0
dtype: int64
```

Data Cleaning and Wrangling

[14]:
```
#Merge with customers

Df=sales_order.merge(customers,how="left",
                     left_on="Customer Name Index",
                     right_on="Customer Index")
Df.head()
```

[14]:

	OrderNumber	OrderDate	Customer Name Index	Channel	Currency Code	Warehouse Code	Delivery Region Index	Product Description Index	Order Quantity	Unit Price	Line Total	Total Unit Cost	Customer Index	Customer Names
0	SO - 000225	2014-01-01	126	Wholesale	USD	AXW291	364	27	6	2499.1	14994.6	1824.343	126	Rhynoodle Ltd
1	SO - 0003378	2014-01-01	96	Distributor	USD	AXW291	488	20	11	2351.7	25868.7	1269.918	96	Thoughtmix Ltd
2	SO - 0005126	2014-01-01	8	Wholesale	USD	AXW291	155	26	6	978.2	5869.2	684.740	8	Amerisourc Corp
3	SO - 0005614	2014-01-01	42	Export	USD	AXW291	473	7	7	2338.3	16368.1	1028.852	42	Colgate-Pa Group
4	SO - 0005781	2014-01-01	73	Wholesale	USD	AXW291	256	8	8	2291.4	18331.2	1260.270	73	Deseret Group

[15]:
```
#Merge with product

Df=Df.merge(products,how="left",
                     left_on="Product Description Index",
                     right_on="Index")
Df.head()
```

[15]:

	OrderNumber	OrderDate	Customer Name Index	Channel	Currency Code	Warehouse Code	Delivery Region Index	Product Description Index	Order Quantity	Unit Price	Line Total	Total Unit Cost	Customer Index	Customer Names	Index	Pr
0	SO - 000225	2014-01-01	126	Wholesale	USD	AXW291	364	27	6	2499.1	14994.6	1824.343	126	Rhynoodle Ltd	27	Pr
1	SO - 0003378	2014-01-01	96	Distributor	USD	AXW291	488	20	11	2351.7	25868.7	1269.918	96	Thoughtmix Ltd	20	Pr
2	SO - 0005126	2014-01-01	8	Wholesale	USD	AXW291	155	26	6	978.2	5869.2	684.740	8	Amerisourc Corp	26	Pr
3	SO - 0005614	2014-01-01	42	Export	USD	AXW291	473	7	7	2338.3	16368.1	1028.852	42	Colgate-Pa Group	7	Pr
4	SO - 0005781	2014-01-01	73	Wholesale	USD	AXW291	256	8	8	2291.4	18331.2	1260.270	73	Deseret Group	8	Pr

[16]:
```
#Merge with regions

Df=Df.merge(regions,how="left",
                    left_on="Delivery Region Index",
                    right_on="id")
Df.head()
```

[16]:

	OrderNumber	OrderDate	Customer Name Index	Channel	Currency Code	Warehouse Code	Delivery Region Index	Product Description Index	Order Quantity	Unit Price	...	type	latitude	longitude	area_code	populatic
0	SO - 000225	2014-01-01	126	Wholesale	USD	AXW291	364	27	6	2499.1	...	City	32.08354	-81.09983	912	1456
1	SO - 0003378	2014-01-01	96	Distributor	USD	AXW291	488	20	11	2351.7	...	City	39.61366	-86.10665	317	555
2	SO - 0005126	2014-01-01	8	Wholesale	USD	AXW291	155	26	6	978.2	...	City	37.66243	-121.87468	925	795
3	SO - 0005614	2014-01-01	42	Export	USD	AXW291	473	7	7	2338.3	...	City	39.16533	-86.52639	812	840
4	SO - 0005781	2014-01-01	73	Wholesale	USD	AXW291	256	8	8	2291.4	...	Town	41.77524	-72.52443	959	580

5 rows × 31 columns

[17]:
```
#Merge with state regions

Df=Df.merge(state_reg[["State Code","Region"]],how="left",
                    left_on="state_code",
                    right_on="State Code")
Df.head()
```

[17]:

	OrderNumber	OrderDate	Customer Name Index	Channel	Currency Code	Warehouse Code	Delivery Region Index	Product Description Index	Order Quantity	Unit Price	...	longitude	area_code	population	households
0	SO - 000225	2014-01-01	126	Wholesale	USD	AXW291	364	27	6	2499.1	...	-81.09983	912	145674	52798
1	SO - 0003378	2014-01-01	96	Distributor	USD	AXW291	488	20	11	2351.7	...	-86.10665	317	55586	20975
2	SO - 0005126	2014-01-01	8	Wholesale	USD	AXW291	155	26	6	978.2	...	-121.87468	925	79510	26020
3	SO - 0005614	2014-01-01	42	Export	USD	AXW291	473	7	7	2338.3	...	-86.52639	812	84067	30232
4	SO - 0005781	2014-01-01	73	Wholesale	USD	AXW291	256	8	8	2291.4	...	-72.52443	959	58007	24141

5 rows × 33 columns

[18]:
```
#Merge with Bugets

Df=Df.merge(budgets,how="left",
                    on="Product Name")
Df.head()
```

[18]:

	OrderNumber	OrderDate	Customer Name Index	Channel	Currency Code	Warehouse Code	Delivery Region Index	Product Description Index	Order Quantity	Unit Price	...	area_code	population	households	median_inco
0	SO - 000225	2014-01-01	126	Wholesale	USD	AXW291	364	27	6	2499.1	...	912	145674	52798	36
1	SO - 0003378	2014-01-01	96	Distributor	USD	AXW291	488	20	11	2351.7	...	317	55586	20975	54
2	SO - 0005126	2014-01-01	8	Wholesale	USD	AXW291	155	26	6	978.2	...	925	79510	26020	124
3	SO - 0005614	2014-01-01	42	Export	USD	AXW291	473	7	7	2338.3	...	812	84067	30232	30
4	SO - 0005781	2014-01-01	73	Wholesale	USD	AXW291	256	8	8	2291.4	...	959	58007	24141	63

5 rows × 34 columns

[19]:
```
#clean  up redundant columns
cols_to_drop=["Customer Index","Index","id","State Code"]
Df=Df.drop(columns=cols_to_drop,errors="ignore")

Df.head()
```

[19]:

	OrderNumber	OrderDate	Customer Name Index	Channel	Currency Code	Warehouse Code	Delivery Region Index	Product Description Index	Order Quantity	Unit Price	...	longitude	area_code	population	households
0	SO - 000225	2014-01-01	126	Wholesale	USD	AXW291	364	27	6	2499.1	...	-81.09983	912	145674	52798
1	SO - 0003378	2014-01-01	96	Distributor	USD	AXW291	488	20	11	2351.7	...	-86.10665	317	55586	20975
2	SO - 0005126	2014-01-01	8	Wholesale	USD	AXW291	155	26	6	978.2	...	-121.87468	925	79510	26020
3	SO - 0005614	2014-01-01	42	Export	USD	AXW291	473	7	7	2338.3	...	-86.52639	812	84067	30232
4	SO - 0005781	2014-01-01	73	Wholesale	USD	AXW291	256	8	8	2291.4	...	-72.52443	959	58007	24141

5 rows × 30 columns

[20]:
```
#convert all columns to lower case for consistency and easier access

Df.columns=Df.columns.str.lower()
Df.columns
```

[20]:
```
Index(['ordernumber', 'orderdate', 'customer name index', 'channel',
       'currency code', 'warehouse code', 'delivery region index',
       'product description index', 'order quantity', 'unit price',
       'line total', 'total unit cost', 'customer names', 'product name',
       'name', 'county', 'state_code', 'state', 'type', 'latitude',
       'longitude', 'area_code', 'population', 'households', 'median_income',
       'land_area', 'water_area', 'time_zone', 'region', '2017 budgets'],
      dtype='object')
```

[21]:
```
#Keep the important columns and delete the columns that we dont need

cols_to_keep=['ordernumber',
              'orderdate','customer names', 'channel','product name',
              'order quantity', 'unit price',
              'line total', 'total unit cost', 'state_code', 'state','county',
              'latitude',
               'longitude',  'region','2017 budgets']
```

[22]:
```
Df=Df[cols_to_keep]
Df.head()
```

[22]:

	ordernumber	orderdate	customer names	channel	product name	order quantity	unit price	line total	total unit cost	state_code	state	county	latitude	longitude	region	
0	SO - 000225	2014-01-01	Rhynoodle Ltd	Wholesale	Product 27	6	2499.1	14994.6	1824.343	GA	Georgia	Chatham County	32.08354	-81.09983	South	9
1	SO - 0003378	2014-01-01	Thoughtmix Ltd	Distributor	Product 20	11	2351.7	25868.7	1269.918	IN	Indiana	Johnson County	39.61366	-86.10665	Midwest	20
2	SO - 0005126	2014-01-01	Amerisourc Corp	Wholesale	Product 26	6	978.2	5869.2	684.740	CA	California	Alameda County	37.66243	-121.87468	West	56
3	SO - 0005614	2014-01-01	Colgate-Pa Group	Export	Product 7	7	2338.3	16368.1	1028.852	IN	Indiana	Monroe County	39.16533	-86.52639	Midwest	8
4	SO - 0005781	2014-01-01	Deseret Group	Wholesale	Product 8	8	2291.4	18331.2	1260.270	CT	Connecticut	Hartford County	41.77524	-72.52443	Northeast	10

Rename columns:

[23]:
```
Df=Df.rename(columns={"line total":"revenue","total unit cost":"cost","2017 budgets":"budgets"})

Df.head()
```

[23]:

	ordernumber	orderdate	customer names	channel	product name	order quantity	unit price	revenue	cost	state_code	state	county	latitude	longitude	region
0	SO - 000225	2014-01-01	Rhynoodle Ltd	Wholesale	Product 27	6	2499.1	14994.6	1824.343	GA	Georgia	Chatham County	32.08354	-81.09983	South
1	SO - 0003378	2014-01-01	Thoughtmix Ltd	Distributor	Product 20	11	2351.7	25868.7	1269.918	IN	Indiana	Johnson County	39.61366	-86.10665	Midwest
2	SO - 0005126	2014-01-01	Amerisourc Corp	Wholesale	Product 26	6	978.2	5869.2	684.740	CA	California	Alameda County	37.66243	-121.87468	West
3	SO - 0005614	2014-01-01	Colgate-Pa Group	Export	Product 7	7	2338.3	16368.1	1028.852	IN	Indiana	Monroe County	39.16533	-86.52639	Midwest
4	SO - 0005781	2014-01-01	Deseret Group	Wholesale	Product 8	8	2291.4	18331.2	1260.270	CT	Connecticut	Hartford County	41.77524	-72.52443	Northeast

Blank out bugets for non-2017 order

[24]:
```
Df.loc[Df["orderdate"].dt.year != 2017,"budgets"] = pd.NA
```

Line total is revenue

[25]:
```
Df[["orderdate","product name","revenue","budgets"]]
```

Information about dataset

[27]:
```
Df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64104 entries, 0 to 64103
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   ordernumber     64104 non-null  object
 1   orderdate       64104 non-null  datetime64[ns]
 2   customer names  64104 non-null  object
 3   channel         64104 non-null  object
 4   product name    64104 non-null  object
 5   order quantity  64104 non-null  int64
 6   unit price      64104 non-null  float64
 7   revenue         64104 non-null  float64
 8   cost            64104 non-null  float64
 9   state_code      64104 non-null  object
 10  state           64104 non-null  object
 11  county          64104 non-null  object
 12  latitude        64104 non-null  float64
 13  longitude       64104 non-null  float64
 14  region          64104 non-null  object
 15  budgets         15263 non-null  float64
dtypes: datetime64[ns](1), float64(6), int64(1), object(8)
memory usage: 7.8+ MB
```

Line total is revenue

[25]:
```
Df[["orderdate","product name","revenue","budgets"]]
```

[25]:

	orderdate	product name	revenue	budgets
0	2014-01-01	Product 27	14994.6	NaN
1	2014-01-01	Product 20	25868.7	NaN
2	2014-01-01	Product 26	5869.2	NaN
3	2014-01-01	Product 7	16368.1	NaN
4	2014-01-01	Product 8	18331.2	NaN
...	...	...	...	...
64099	2018-02-28	Product 26	21788.4	NaN
64100	2018-02-28	Product 21	5185.8	NaN
64101	2018-02-28	Product 13	43483.0	NaN
64102	2018-02-28	Product 20	27717.9	NaN
64103	2018-02-28	Product 15	7986.4	NaN

64104 rows × 4 columns

Filter the dataset to include only records from year 2017

[28]:
```
Df_2017=Df[Df["orderdate"].dt.year == 2017]
```

[29]:
```
Df_2017
```

[29]:

	ordernumber	orderdate	customer names	channel	product name	order quantity	unit price	revenue	cost	state_code	state	county	latitude	longitude	reg
46363	SO - 0002544	2017-01-01	NCS Group	Wholesale	Product 30	6	1239.5	7437.0	1028.785	NJ	New Jersey	Hudson County	40.77955	-74.02375	North
46364	SO - 0006431	2017-01-01	Epic Group	Wholesale	Product 13	5	1829.1	9145.5	1207.206	CO	Colorado	Mesa County	39.06387	-108.55065	W
46365	SO - 0007491	2017-01-01	State Ltd	Wholesale	Product 15	9	2412.0	21708.0	1664.280	CA	California	Los Angeles County	33.96168	-118.35313	W
46366	SO - 0008741	2017-01-01	Fivebridge Ltd	Wholesale	Product 8	8	904.5	7236.0	750.735	IA	Iowa	Dubuque County	42.50056	-90.66457	Mid
46367	SO - 0009295	2017-01-01	Tagfeed Ltd	Wholesale	Product 2	12	1112.2	13346.4	811.906	FL	Florida	Hernando County	28.47689	-82.52546	So
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
61621	SO - 0003524	2017-12-31	Zooveo Company	Wholesale	Product 1	9	984.9	8864.1	512.148	MO	Missouri	Jackson County/Clay County	39.08547	-94.35210	Mid
61622	SO - 0004785	2017-12-31	Wordware Company	Wholesale	Product 3	12	201.0	2412.0	90.450	PA	Pennsylvania	Philadelphia County	39.95234	-75.16379	North
61623	SO - 0004950	2017-12-31	Dynazzy Company	Distributor	Product 6	9	3825.7	34431.3	3098.817	FL	Florida	Volusia County	29.13832	-80.99561	So
61624	SO - 0006829	2017-12-31	Pixoboo Corp	Distributor	Product 6	5	1835.8	9179.0	972.974	CT	Connecticut	New Haven County	41.22509	-73.06111	North
61625	SO - 0009850	2017-12-31	Rooxo Company	Distributor	Product 26	8	207.7	1661.6	93.465	CA	California	Madera County	36.96134	-120.06072	W

15263 rows × 16 columns

Calculate total cost and profit ¶

[30]:
```
Df["total_cost"]=Df["order quantity"]* Df["cost"]
```

[31]:
```
Df["profit"]=Df["revenue"]-Df["total_cost"]

Df["profit_margin_%"]=Df["profit"]/Df["revenue"]*100
```

[32]:
```
Df.head()
```

[32]:

	ordernumber	orderdate	customer names	channel	product name	order quantity	unit price	revenue	cost	state_code	state	county	latitude	longitude	region	b
0	SO - 000225	2014-01-01	Rhynoodle Ltd	Wholesale	Product 27	6	2499.1	14994.6	1824.343	GA	Georgia	Chatham County	32.08354	-81.09983	South	
1	SO - 0003378	2014-01-01	Thoughtmix Ltd	Distributor	Product 20	11	2351.7	25868.7	1269.918	IN	Indiana	Johnson County	39.61366	-86.10665	Midwest	
2	SO - 0005126	2014-01-01	Amerisourc Corp	Wholesale	Product 26	6	978.2	5869.2	684.740	CA	California	Alameda County	37.66243	-121.87468	West	
3	SO - 0005614	2014-01-01	Colgate-Pa Group	Export	Product 7	7	2338.3	16368.1	1028.852	IN	Indiana	Monroe County	39.16533	-86.52639	Midwest	
4	SO - 0005781	2014-01-01	Deseret Group	Wholesale	Product 8	8	2291.4	18331.2	1260.270	CT	Connecticut	Hartford County	41.77524	-72.52443	Northeast	

[51]:
```
Df.to_csv("Regional_dataset.csv")
```

EXPLORATORY DATA ANALYSIS

Top 10 Products

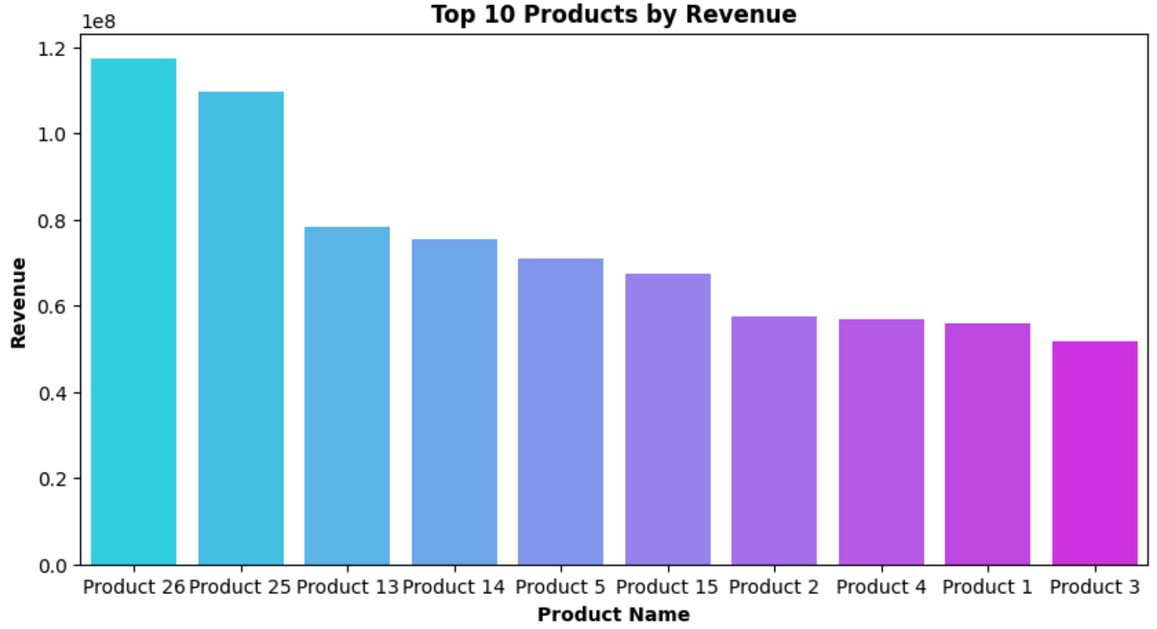
[33]:
```
product_revenue=Df.groupby("product name")["revenue"].sum().reset_index()
```

[34]:
```
top_10=product_revenue.sort_values(by="revenue",ascending=False).head(10)
top_10
```

[34]:

	product name	revenue
18	Product 26	117291821.4
17	Product 25	109473966.6
4	Product 13	78281379.6
5	Product 14	75390396.6
25	Product 5	70804380.6
6	Product 15	67331623.2
11	Product 2	57401097.6
24	Product 4	56701537.2
0	Product 1	55952289.6
22	Product 3	51764816.4

[35]:
```
plt.figure(figsize=(10,5))
sns.barplot(x="product name",y="revenue",data=top_10,palette="cool")
plt.xlabel("Product Name",fontweight="bold")
plt.ylabel("Revenue",fontweight="bold")
plt.title("Top 10 Products by Revenue",fontweight="bold")
plt.show()
```

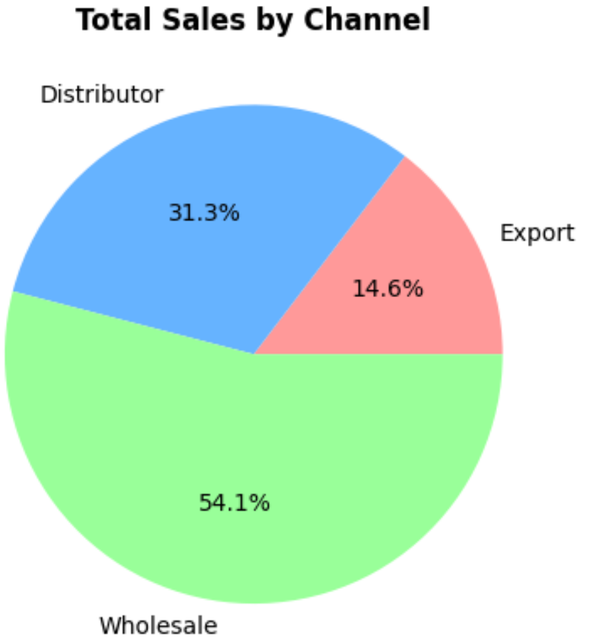


Sales by channel

[36]:
```
chan_sales=Df.groupby("channel")["revenue"].sum().sort_values(ascending=True)
chan_sales
```

[36]:
```
channel
Export         180631866.0
Distributor    387139788.6
Wholesale      668197244.4
Name: revenue, dtype: float64
```

[37]:
```
colors=["#ff9999","#66b3ff","#99ff99"]
plt.pie(chan_sales.values,labels=chan_sales.index,autopct="%1.1f%%",colors=colors)
plt.title("Total Sales by Channel",fontweight="bold")
plt.show()
```

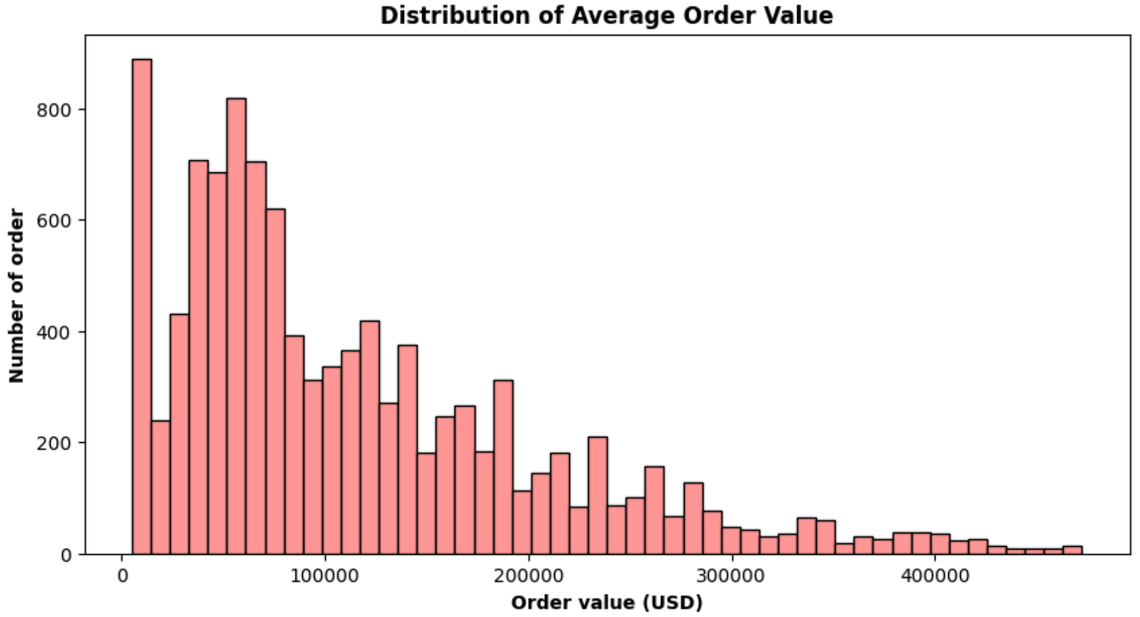


Average order value distribution

[38]:
```
Avg_od=Df.groupby("ordernumber")["revenue"].sum()
Avg_od
```

[38]:
```
ordernumber
SO - 0001000     38592.0
SO - 0001001      8361.6
SO - 0001002    347127.0
SO - 0001003    195372.0
SO - 0001004     81686.4
                   ...
SO - 0010780    163212.0
SO - 0010781     47275.2
SO - 0010782     79113.6
SO - 0010783     40762.8
SO - 0010784    345800.4
Name: revenue, Length: 10684, dtype: float64
```

[39]:
```
plt.figure(figsize=(10,5))
plt.hist(Avg_od,bins=50,color="#ff9999",edgecolor="black")
plt.title("Distribution of Average Order Value",fontweight="bold")
plt.xlabel("Order value (USD)",fontweight="bold")
plt.ylabel("Number of order",fontweight="bold")
plt.show()
```



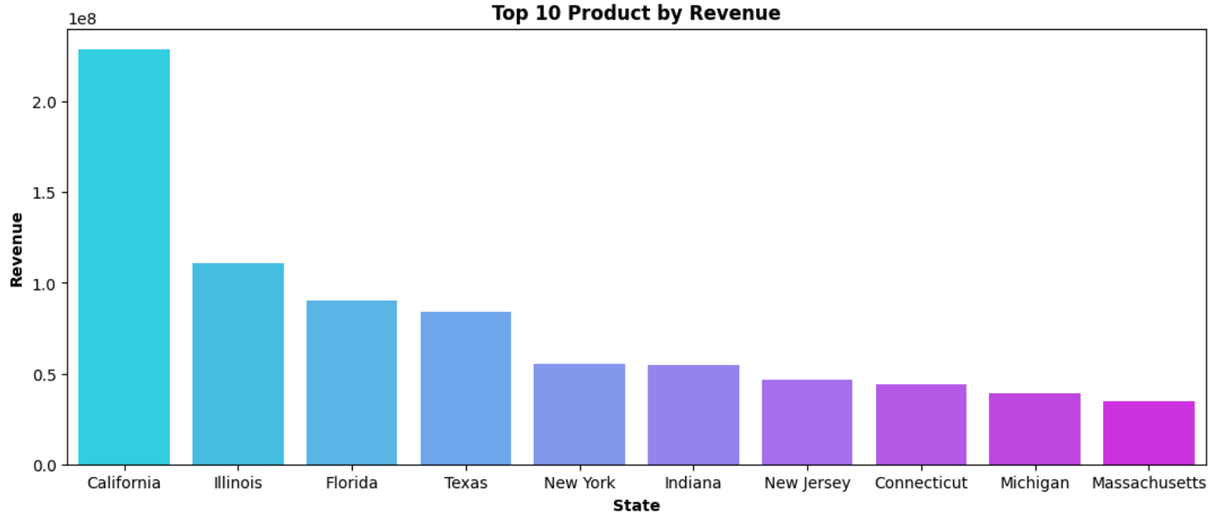
[40]:
```
State_Rev=Df.groupby("state")["revenue"].sum().sort_values(ascending=False).reset_index()
State_Rev
```

[40]:

	state	revenue
0	California	228785436.0
1	Illinois	111050965.7
2	Florida	90204679.5
3	Texas	84011903.0
4	New York	55534960.0
5	Indiana	54601690.2
6	New Jersey	46830956.5
7	Connecticut	44251228.7
8	Michigan	39025315.8

[41]:
```
Top_10_rev=State_Rev.head(10)
```

[42]:
```
plt.figure(figsize=(13,5))
sns.barplot(x="state",y="revenue",data=Top_10_rev,palette="cool")
plt.xlabel("State",fontweight="bold")
plt.ylabel("Revenue",fontweight="bold")
plt.title("Top 10 Product by Revenue",fontweight="bold")
plt.show()
```



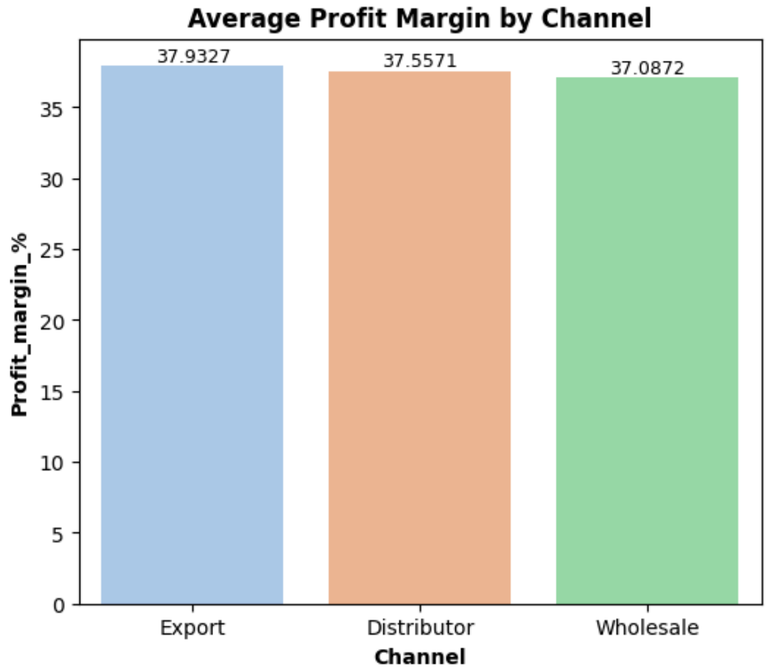
Average Profit MArgin by channel

[43]:
```
channel_margin=Df.groupby("channel")["profit_margin_%"].mean().sort_values(ascending=False).reset_index()
channel_margin
```

[43]:

	channel	profit_margin_%
0	Export	37.932704
1	Distributor	37.557091
2	Wholesale	37.087236

[44]:
```
plt.figure(figsize=(6,5))
ax=sns.barplot(x="channel",y="profit_margin_%",data=channel_margin,palette="pastel")
plt.xlabel("Channel",fontweight="bold")
plt.ylabel("Profit_margin_%",fontweight="bold")
plt.title("Average Profit Margin by Channel",fontweight="bold")
for container in ax.containers:
    ax.bar_label(container,label_type="edge",fontsize=9,color="black")
plt.show()
```



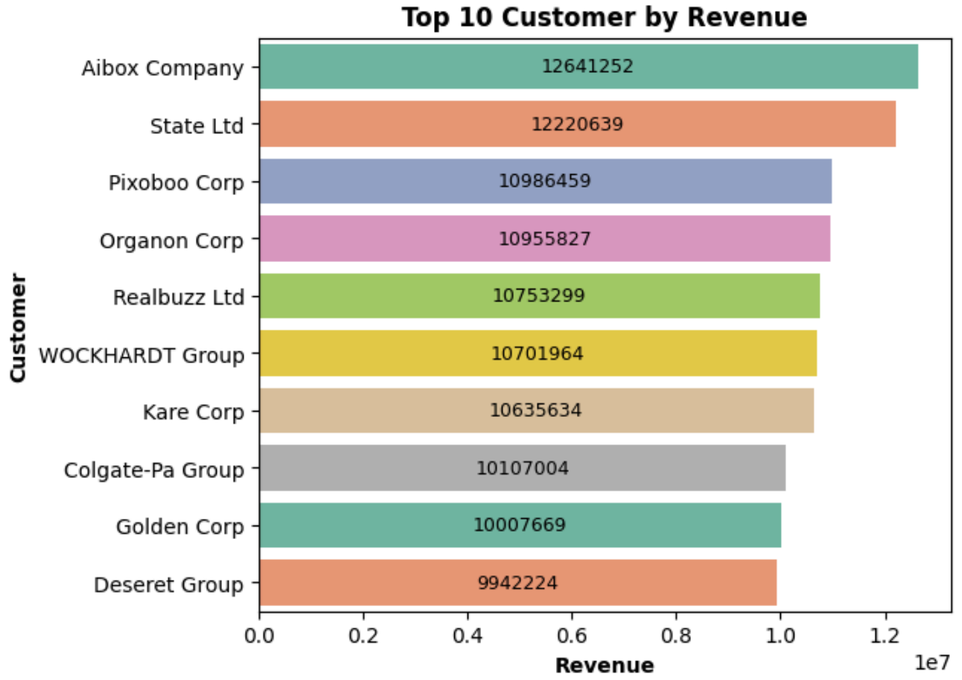
Top 10 customers by revenue

[45]:
```
Top_cust=Df.groupby("customer names")["revenue"].sum().sort_values(ascending=False).reset_index().head(10)
Top_cust
```

[45]:

	customer names	revenue
0	Aibox Company	12641251.8
1	State Ltd	12220639.2
2	Pixoboo Corp	10986459.0
3	Organon Corp	10955826.6
4	Realbuzz Ltd	10753299.0
5	WOCKHARDT Group	10701963.6
6	Kare Corp	10635633.6
7	Colgate-Pa Group	10107003.6
8	Golden Corp	10007669.4
9	Deseret Group	9942223.8

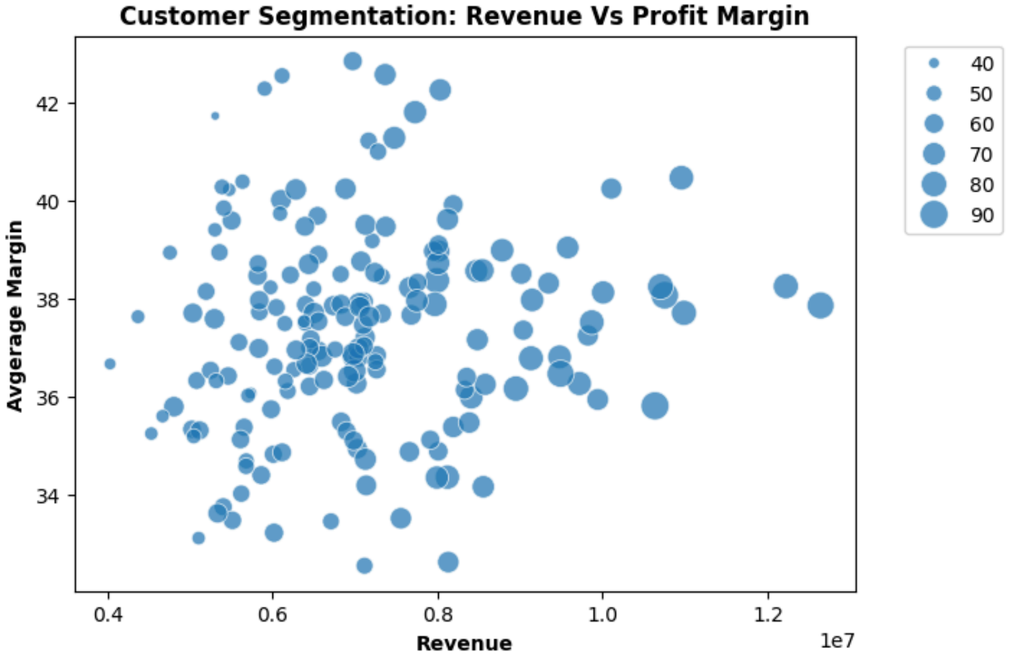
[46]:
```
plt.figure(figsize=(6,5))
ax=sns.barplot(y="customer names",x="revenue",data=Top_cust,palette="Set2")
plt.ylabel("Customer",fontweight="bold")
plt.xlabel("Revenue",fontweight="bold")
plt.title("Top 10 Customer by Revenue",fontweight="bold")
for container in ax.containers:
    ax.bar_label(container,fmt="%.0f",label_type="center",fontsize=9,color="black")
plt.show()
```



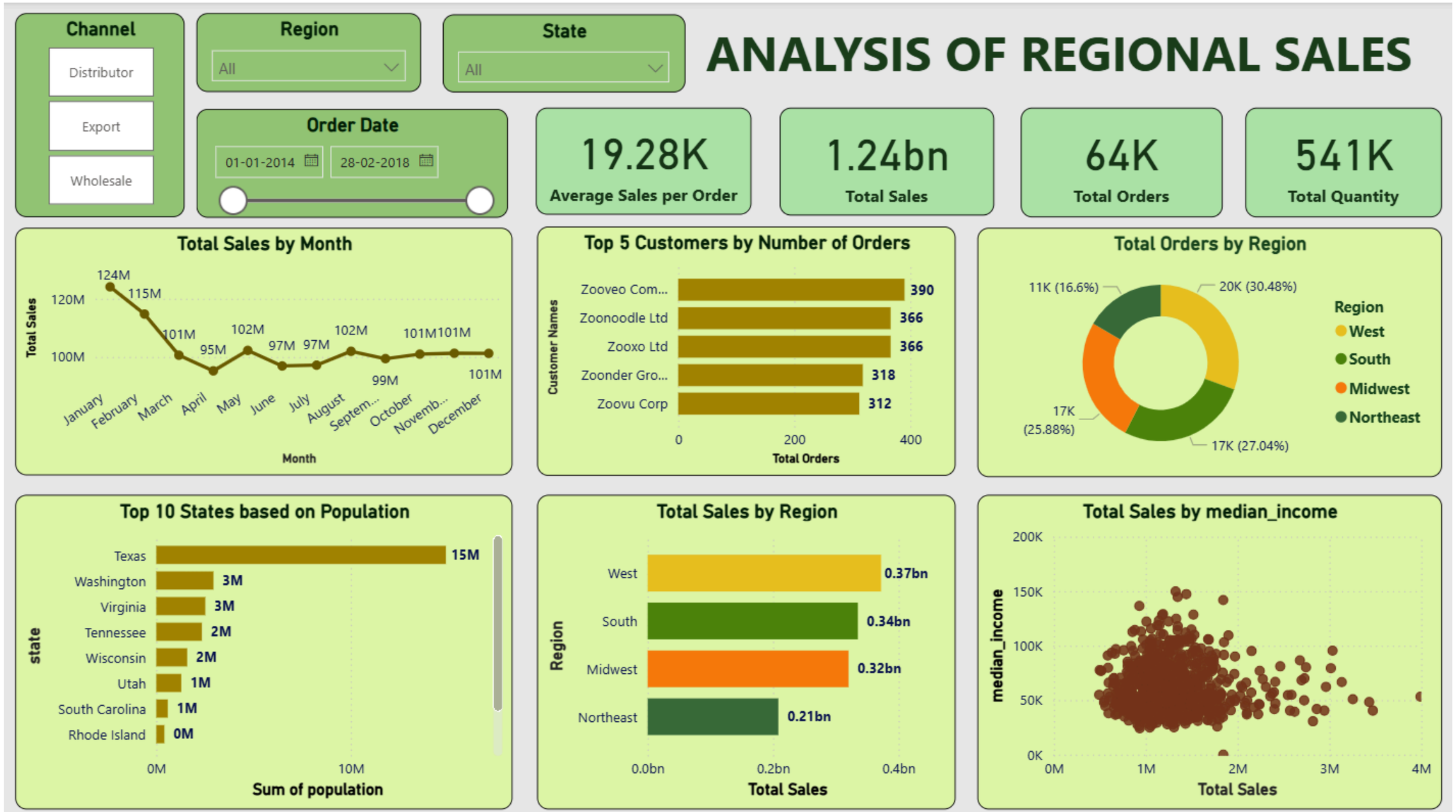
Customer Segmentation :Revenue vs Profit Margin

[47]:
```
cust_summary=Df.groupby("customer names").agg(total_revenue=("revenue","sum"),
                                              total_profit=("profit","sum"),
                                              avg_margin=("profit_margin_%","mean"),
                                              orders=("ordernumber","nunique"))
```

[48]:
```
plt.figure(figsize=(7,5))
sns.scatterplot(x="total_revenue",y="avg_margin",size="orders",data=cust_summary,sizes=(20,200),alpha=0.7)
plt.title("Customer Segmentation: Revenue Vs Profit Margin",fontweight="bold")
plt.xlabel("Revenue",fontweight="bold")
plt.ylabel("Avgerage Margin",fontweight="bold")
plt.legend(bbox_to_anchor=(1.05,1),loc=2)
plt.show()
```



DASHBOARD (POWER BI)



Insights

1.West region generated the highest revenue, contributing approximately ₹370M, making it the top-performing region overall.
2. Northeast region had the lowest sales and order volume, indicating a strong opportunity for market expansion.
3. January recorded the highest monthly sales, suggesting a seasonal peak early in the year.
4. June had the lowest sales across all months, indicating a potential off-season period for targeted promotions.
5. Over 64,000 orders were placed, with a total revenue exceeding ₹1.24 billion, reflecting a healthy order volume.
6. The average sales per order stood at ₹19.28K, showing consistent order value across regions.
7. Top 5 customers alone contributed over 1,800 orders, highlighting strong B2B relationships.
8. High-population states like Texas showed strong sales, while others like Washington and Virginia underperformed, revealing untapped potential.
9. There is no strong correlation between median income and total sales, suggesting product affordability across all income levels.
10. Sales distribution by channel showed a balanced contribution, with room to optimize underperforming channels further.

Thank You