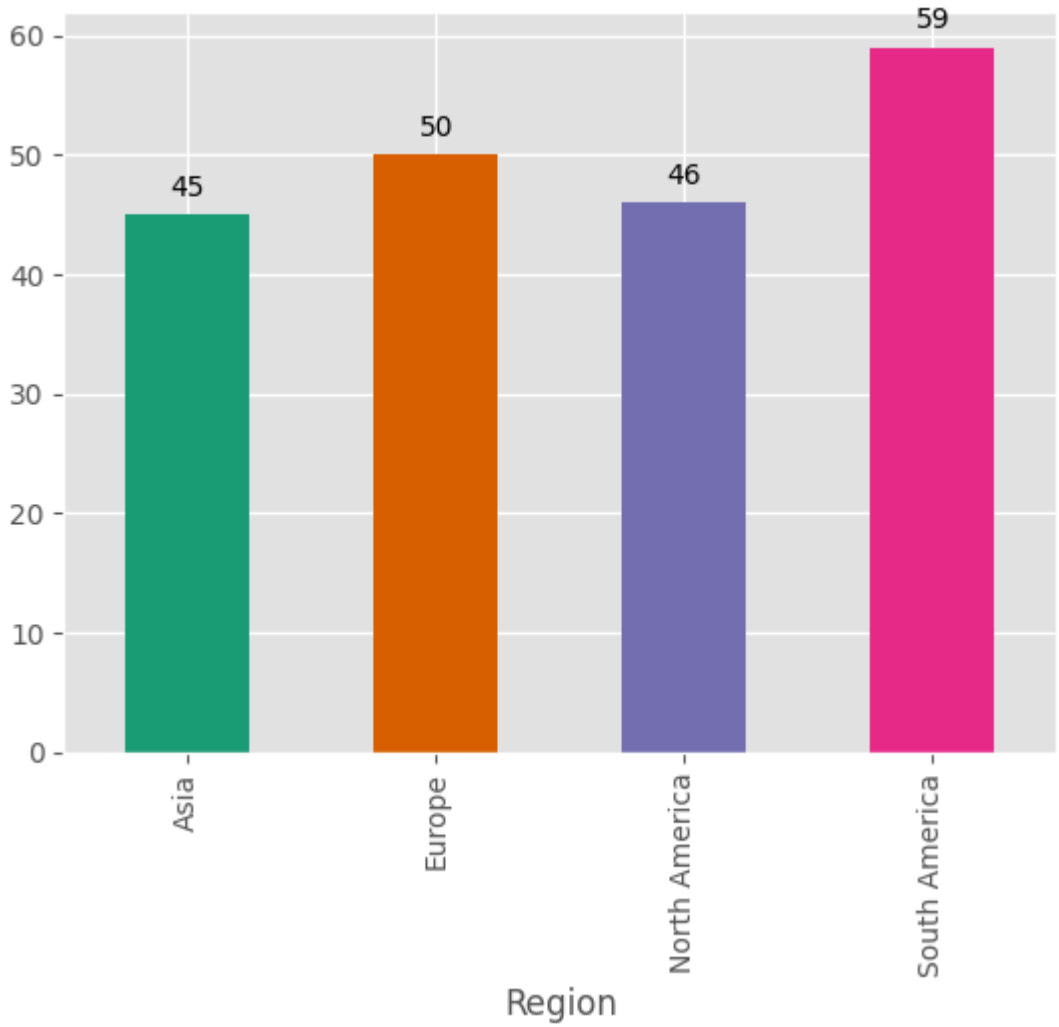
Exploratory Data Analysis

Customer Analysis :

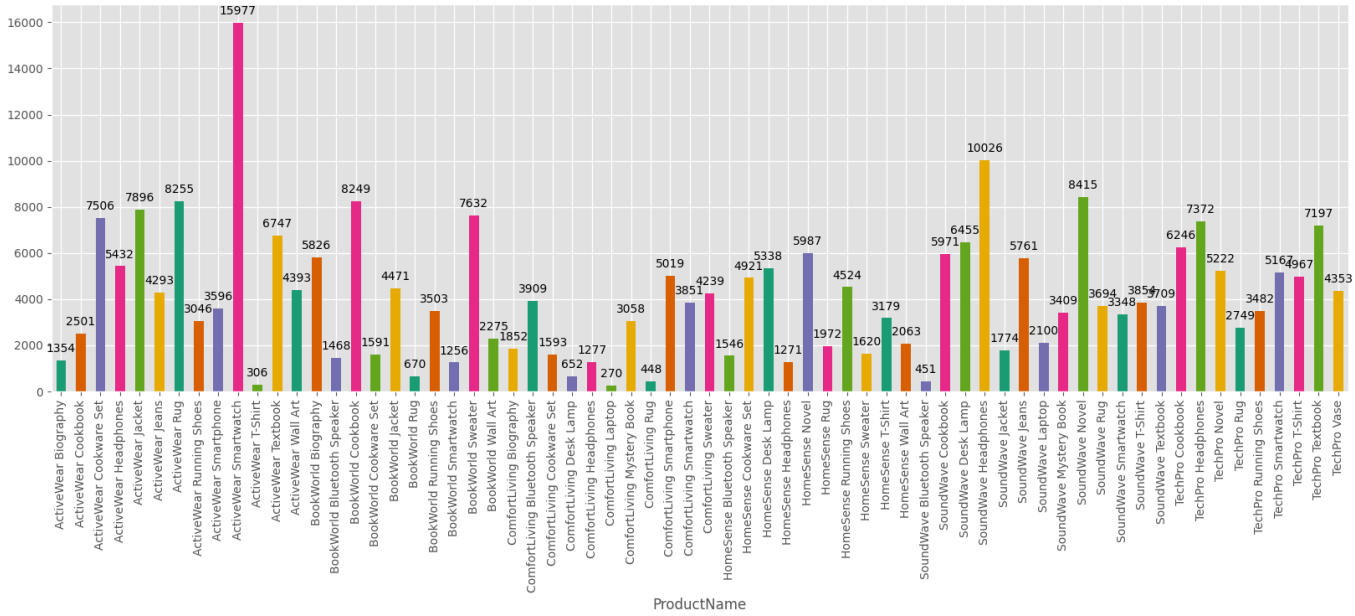
1. **South America** => **59**
2. **Europe** => **50**
3. **North America** => **46**
4. **Asia** => **45**

- **Average signup per day** : **1**
- **Max signup per day** : **3**
- **Min signup per day** : **1**



Product Sales & Profit Analysis

Number of Unique Products: 100



Top 5 Most Sold Products (Count Wise)

ProductID	Count	ProductName	Category	Price
P059	19	SoundWave Jeans	Clothing	303.20
P029	17	TechPro Headphones	Electronics	433.64
P062	16	HomeSense Novel	Books	374.16
P079	16	ActiveWear Rug	Home Decor	417.37
P054	16	SoundWave Cookbook	Books	57.30

Top 5 Least Sold Products (Count Wise)

ProductID	Count	ProductName	Category	Price
P024	5	SoundWave Cookbook	Books	338.66
P014	4	ActiveWear Jacket	Clothing	26.26
P031	4	SoundWave Headphones	Electronics	196.40
P044	4	ActiveWear Running Shoes	Clothing	18.82
P099	4	SoundWave Mystery Book	Books	354.29

Top 5 Most Profitable Products (Profit Wise)

ProductID	Count	ProductName	Category	Price	Total Profit
P029	17	TechPro Headphones	Electronics	433.64	7371.88
P079	16	ActiveWear Rug	Home Decor	417.37	6677.92
P048	15	TechPro Cookbook	Books	416.40	6246.00
P062	16	HomeSense Novel	Books	374.16	5986.56
P083	13	ActiveWear Smartwatch	Electronics	455.72	5924.36

Top 5 Least Profitable Products (Profit Wise)

ProductID	Count	ProductName	Category	Price	Total Profit
P070	6	HomeSense T-Shirt	Clothing	48.69	292.14
P073	10	ComfortLiving Laptop	Electronics	26.99	269.90
P056	8	SoundWave Smartwatch	Electronics	16.08	128.64
P014	4	ActiveWear Jacket	Clothing	26.26	105.04
P044	4	ActiveWear Running Shoes	Clothing	18.82	75.28

Transaction & Product Price Summary

Transaction Value Statistics

Metric	Value
Average Transaction Value	690
Maximum Transaction Value	1991
Minimum Transaction Value	16

Product Price Statistics

Metric	Value
Average Price of Products	273
Maximum Price of Products	498
Minimum Price of Products	16

Customer Spending Analysis

Top 5 Most Spending Customers

CustomerID	Customer Name	Region	Signup Date	Total Spending ($)
C0141	Paul Parsons	Europe	2023-02-23	10673.87
C0054	Bruce Rhodes	Asia	2024-09-29	8040.39
C0065	Gerald Hines	North America	2024-07-10	7663.70
C0156	William Adams	North America	2023-08-19	7634.45
C0082	Aimee Taylor	South America	2022-05-13	7572.91

Top 5 Least Spending Customers

CustomerID	Customer Name	Region	Signup Date	Total Spending ($)
C0014	Deborah Wilcox	Europe	2024-06-22	318.66
C0151	Amber Gonzalez	South America	2024-11-22	223.96
C0097	Tina Ford	Asia	2023-12-18	137.54
C0033	Tyler Holt	North America	2024-08-04	132.64
C0060	James Murphy	Europe	2022-04-22	82.36

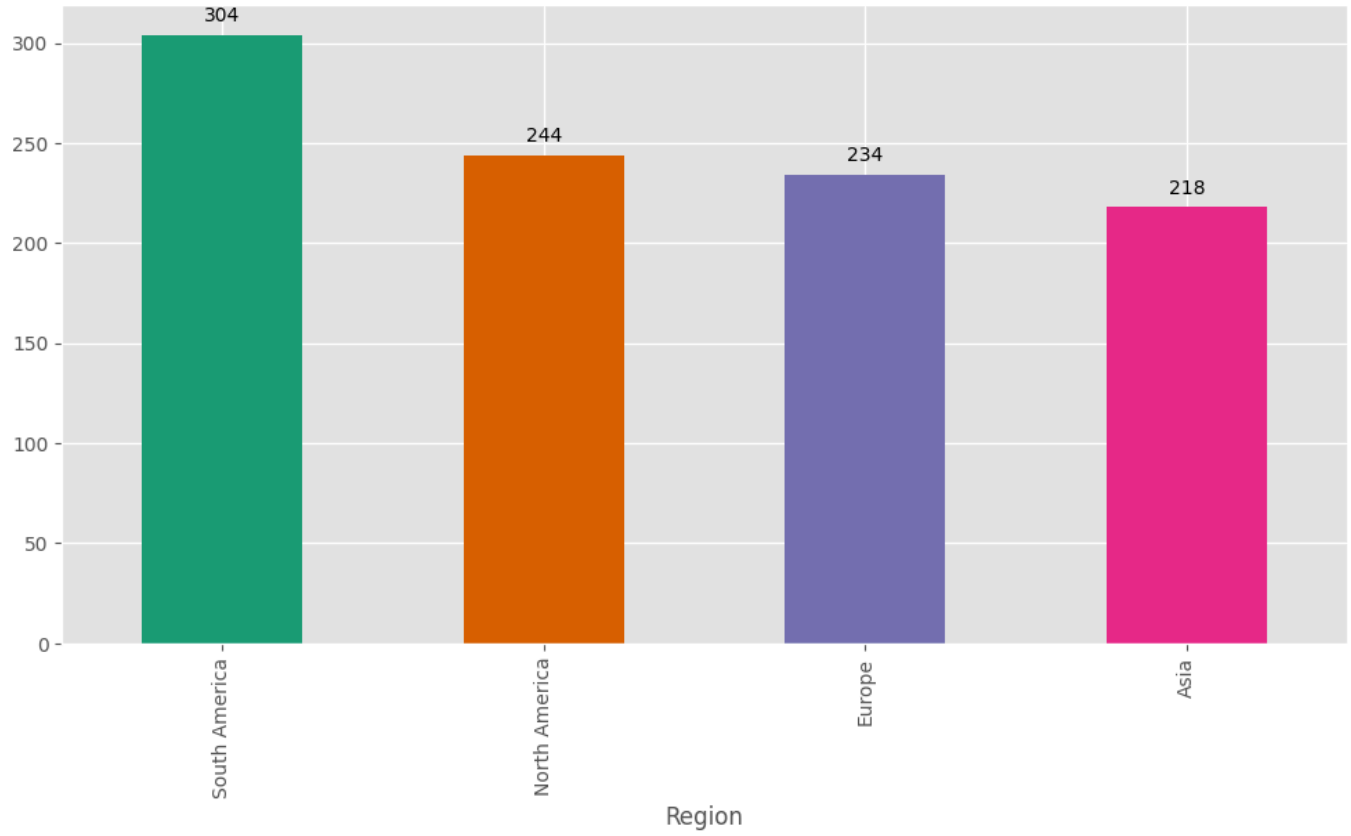
Spending by Region

Region	Total Spending ($)
South America	219,352.56
Europe	166,254.63
North America	152,313.40
Asia	152,074.97

Customer Spending Summary

Metric	Value ($)
Average Spending per Customer	3,467

Most transaction happening regions:



Regional Spending Summary

| Region | Avg Spending ()|$MaxSpending()$ | Min Spending ($) | |----------|----------|----------|----------| | Europe | 3,325 | 10,674 | 82 | | Asia | 3,456 | 8,040 | 138 | | North America | 3,311 | 7,664 | 133 | | South America | 3,718 | 7,573 | 224 | |----------|----------|----------|----------|

Assignment Tasks:

Task 1: Exploratory Data Analysis (EDA) and Business Insights

1. Perform EDA on the provided dataset.
2. Derive at least 5 business insights from the EDA.
   - Write these insights in short point-wise sentences (maximum 100 words per insight).

Deliverables:

- A Jupyter Notebook/Python script containing your EDA code.
- A PDF report with business insights (maximum 500 words).

Task 2: Lookalike Model

Build a Lookalike Model that takes a user's information as input and recommends 3 similar customers based on their profile and transaction history. The model should:

- Use both customer and product information.
- Assign a similarity score to each recommended customer.

Deliverables:

- Give the top 3 lookalikes with there similarity scores for the first 20 customers (CustomerID: C0001 - C0020) in Customers.csv. Form an “Lookalike.csv” which has just one
- A Jupyter Notebook/Python script explaining your model development.

Evaluation Criteria:

- Model accuracy and logic.
- Quality of recommendations and similarity scores

Evaluation Criteria:

- Model accuracy and logic.
- Quality of recommendations and similarity scores.

Task 3: Customer Segmentation / Clustering

Perform customer segmentation using clustering techniques. Use both profile information (from Customers.csv) and transaction information (from Transactions.csv). ● You have the flexibility to choose any clustering algorithm and any number of clusters in between(2 and 10) ● Calculate clustering metrics, including the DB Index(Evaluation will be done on this). ● Visualise your clusters using relevant plots.

Deliverables:

● A report on your clustering results, including:

- The number of clusters formed.
- DB Index value.
- Other relevant clustering metrics.

● A Jupyter Notebook/Python script containing your clustering code.

Evaluation Criteria:

- Clustering logic and metrics.
- Visual representation of clusters.

Submission Instructions:

1. GitHub Link
   - Upload all the PDF and code files in a public GitHub repository.
2. File Naming Convention:
   - Use the following naming convention for all your files:
     - FirstName_LastName_EDA.pdf
     - FirstName_LastName_EDA.ipynb
     - FirstName_LastName_Lookalike.csv
     - FirstName_LastName_Lookalike.ipynb
     - FirstName_LastName_Clustering.pdf

- FirstName_LastName_Clustering.ipynb

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
```

```
customer=pd.read_csv('Customers.csv')
transactions=pd.read_csv('Transactions.csv')
products=pd.read_csv('Products.csv')
```

```
customer.isna().sum()
```

	0
CustomerID	0
CustomerName	0
Region	0
SignupDate	0

dtype: int64

```
customer.describe()
```

	CustomerID	CustomerName	Region	SignupDate
count	200	200	200	200
unique	200	200	4	179
top	C0001	Lawrence Carroll	South America	2024-11-11
freq	1	1	59	3

```
customer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   CustomerID    200 non-null    object
 1   CustomerName  200 non-null    object
 2   Region        200 non-null    object
 3   SignupDate    200 non-null    object
```

```
dtypes: object(4)
memory usage: 6.4+ KB
```

```
transactions.isna().sum()
```

	0
TransactionID	0
CustomerID	0
ProductID	0
TransactionDate	0
Quantity	0
TotalValue	0
Price	0

dtype: int64

```
transactions.describe()
```

	Quantity	TotalValue	Price
count	1000.000000	1000.000000	1000.00000
mean	2.537000	689.995560	272.55407
std	1.117981	493.144478	140.73639
min	1.000000	16.080000	16.08000
25%	2.000000	295.295000	147.95000
50%	3.000000	588.880000	299.93000
75%	4.000000	1011.660000	404.40000
max	4.000000	1991.040000	497.76000

```
transactions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   TransactionID    1000 non-null   object 
 1   CustomerID       1000 non-null   object 
 2   ProductID        1000 non-null   object 
 3   TransactionDate  1000 non-null   object 
 4   Quantity         1000 non-null   int64  
 5   TotalValue       1000 non-null   float64
 6   Price            1000 non-null   float64
```

```
dtypes: float64(2), int64(1), object(4)
memory usage: 54.8+ KB
```

```
products.isna().sum()
```

	0
ProductID	0
ProductName	0
Category	0
Price	0

dtype: int64

```
products.describe()
```

	Price
count	100.000000
mean	267.551700
std	143.219383
min	16.080000
25%	147.767500
50%	292.875000
75%	397.090000
max	497.760000

```
products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   ProductID    100 non-null    object 
 1   ProductName  100 non-null    object 
 2   Category     100 non-null    object 
 3   Price        100 non-null    float64
dtypes: float64(1), object(3)
memory usage: 3.3+ KB
```

```
customer['SignupDate'] = pd.to_datetime(customer['SignupDate'])
transactions['TransactionDate'] = pd.to_datetime(transactions['TransactionDate'])
```

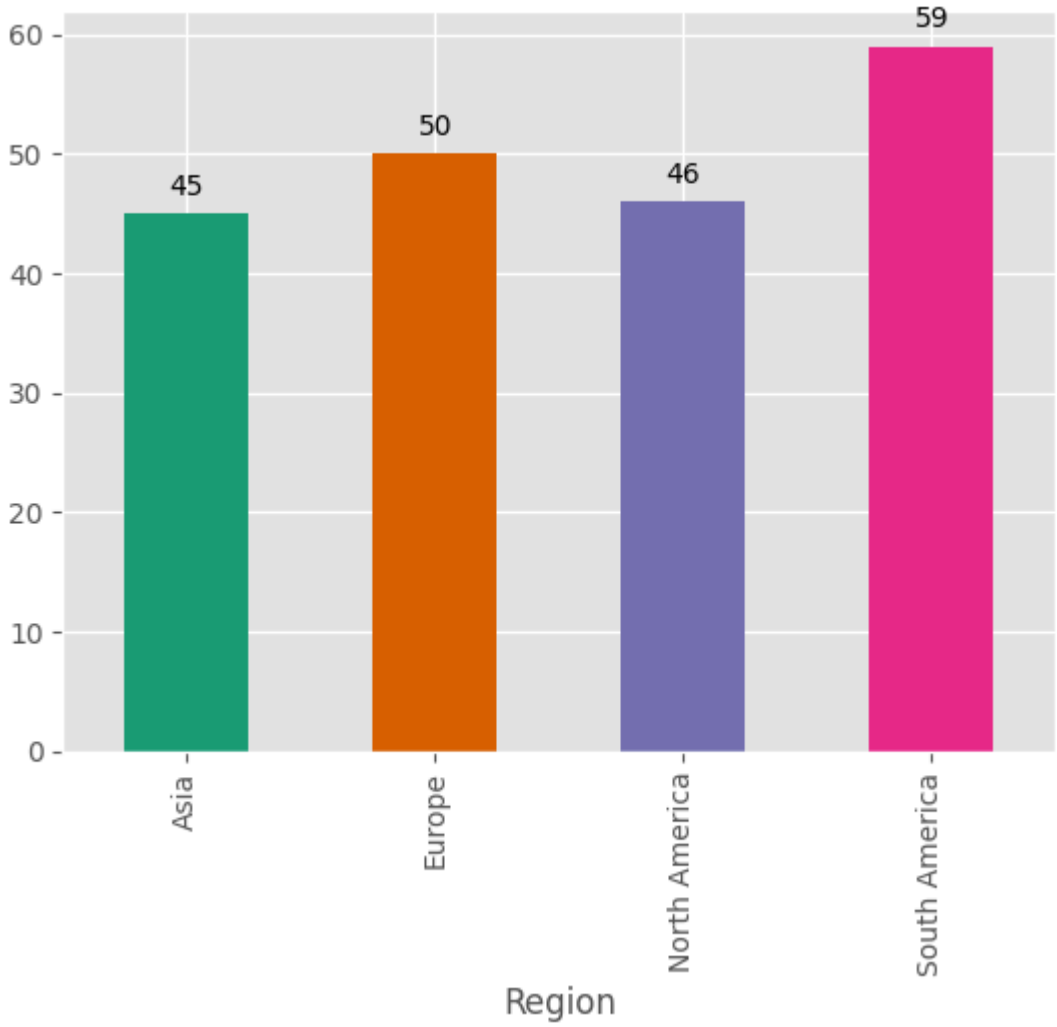
```
print(customer.describe())
print(transactions.describe())
```

```
                SignupDate
count                  200
mean   2023-07-19 08:31:12
min    2022-01-22 00:00:00
25%    2022-09-26 12:00:00
50%    2023-08-31 12:00:00
75%    2024-04-12 12:00:00
max    2024-12-28 00:00:00
                     TransactionDate     Quantity   TotalValue       Price
count                           1000  1000.000000  1000.000000  1000.00000
mean   2024-06-23 15:33:02.768999936     2.537000   689.995560   272.55407
min              2023-12-30 15:29:12     1.000000    16.080000    16.08000
25%       2024-03-25 22:05:34.500000     2.000000   295.295000   147.95000
50%       2024-06-26 17:21:52.500000     3.000000   588.880000   299.93000
75%             2024-09-19 14:19:57     4.000000  1011.660000   404.40000
max             2024-12-28 11:00:00     4.000000  1991.040000   497.76000
std                             NaN     1.117981   493.144478   140.73639
```

Region Of Customers

```
# @title Region Of Customers
print("Customers region :")
customer = pd.read_csv('Customers.csv')
customer['SignupDate'] = pd.to_datetime(customer['SignupDate'])
ax = customer.groupby('Region').size().plot(kind='bar', color=sns.palettes.mpl_pa

# Add numbers on top of bars
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, 10),
                textcoords = 'offset points')

plt.show()
```

Customers region :



customer

	CustomerID	CustomerName	Region	SignupDate
0	C0001	Lawrence Carroll	South America	2022-07-10
1	C0002	Elizabeth Lutz	Asia	2022-02-13
2	C0003	Michael Rivera	South America	2024-03-07
3	C0004	Kathleen Rodriguez	South America	2022-10-09
4	C0005	Laura Weber	Asia	2022-08-15
...	...	...	...	...
195	C0196	Laura Watts	Europe	2022-06-07
196	C0197	Christina Harvey	Europe	2023-03-21
197	C0198	Rebecca Ray	Europe	2022-02-27
198	C0199	Andrea Jenkins	Europe	2022-12-03
199	C0200	Kelly Cross	Asia	2023-06-11

200 rows × 4 columns

Next steps: Generate code with customer | View recommended plots | New interactive sheet

```
# Signup count using customers csv
signup_counts = customer.groupby('SignupDate').size()
print("Average signup per day : ", round(signup_counts.mean()))
print("Max signup per day : ", round(signup_counts.max()))
print("Min signup per day : ", round(signup_counts.min()))
```

```
Average signup per day :  1
Max signup per day :  3
Min signup per day :  1
```

```
# Most & least customer regions
customer.groupby('Region').size().sort_values(ascending=False)
```

Region	0
South America	59
Europe	50
North America	46
Asia	45

dtype: int64

Customer Insights

1. Customers are from four major regions : South and North america, asia & europe.

```
South America : 59
North America : 50
Europe : 46
Asia : 45
```

2. Average signup per day : 1.1173184357541899
3. Max signup per day : 3
4. Min signup per day : 1

```
transactions.head()
```

	TransactionID	CustomerID	ProductID	TransactionDate	Quantity	TotalValue
0	T00001	C0199	P067	2024-08-25 12:38:23	1	300.68
1	T00112	C0146	P067	2024-05-27 22:23:54	1	300.68
2	T00166	C0127	P067	2024-04-25 07:38:55	1	300.68

Next steps: Generate code with transactions | View recommended plots | New interactive sheet

```
products.head()
```

	ProductID	ProductName	Category	Price
0	P001	ActiveWear Biography	Books	169.30
1	P002	ActiveWear Smartwatch	Electronics	346.30
2	P003	ComfortLiving Biography	Books	44.12
3	P004	BookWorld Rug	Home Decor	95.69
4	P005	TechPro T-Shirt	Clothing	429.31

Next steps: Generate code with products | View recommended plots | New interactive sheet

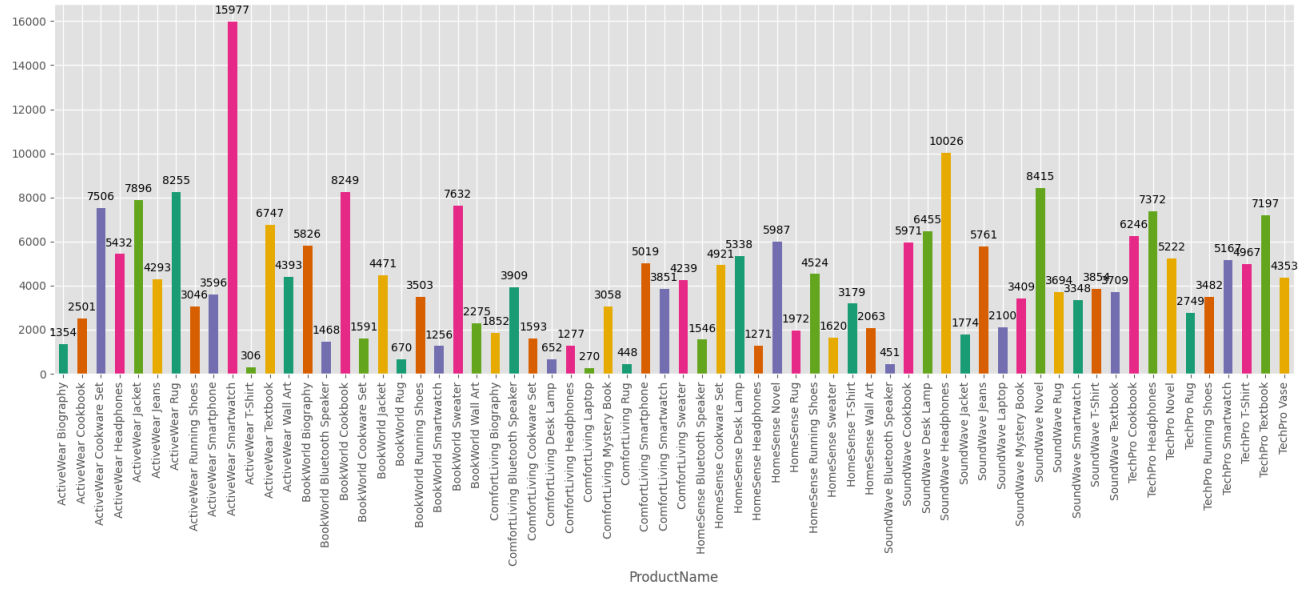
Total no. of products

```
# @title Total no. of products
print("Number of unique products",len(transactions['ProductID'].unique()))
print("------------------------------------------------ |\n\nTop 5 most sold pr
products_Count = transactions['ProductID'].value_counts().reset_index()
products_Count.columns = ['ProductID', 'Count']
product_analysis = pd.merge(products_Count, products, on='ProductID', how='inner'

print(product_analysis.head(5))


print("------------------------------------------------ |\n\n\nTop 5 least sold
print(product_analysis.tail(5))


product_analysis['TotalProfitPerProduct'] = product_analysis['Count'] * product_a


print("------------------------------------------------ |\n\n\nTop 5 most profi
print(product_analysis.sort_values(by='TotalProfitPerProduct', ascending=False).h
```

```
print("------------------------------------------------- |\n\n\nTop 5 least prof
print(product_analysis.sort_values(by='TotalProfitPerProduct', ascending=False).t



plt.figure(figsize=(20, 6))
TotalProfitPerProduct = product_analysis.groupby('ProductName')['TotalProfitPerPr

ax = TotalProfitPerProduct.plot(kind='bar', color=sns.palettes.mpl_palette('Dark2

for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, 10),
                textcoords = 'offset points')

plt.show()
product_analysis.head()
```

```
Number of unique products 100
---------------------------------------------------- |

Top 5 most sold products (Count wise):
-----------------------------------------------------|

  ProductID  Count         ProductName     Category   Price
0      P059     19     SoundWave Jeans     Clothing  303.20
1      P029     17  TechPro Headphones  Electronics  433.64
2      P062     16     HomeSense Novel        Books  374.16
3      P079     16      ActiveWear Rug   Home Decor  417.37
4      P054     16  SoundWave Cookbook        Books   57.30
---------------------------------------------------- |


Top 5 least sold products (Count wise):
---------------------------------------------------- |

   ProductID  Count               ProductName     Category   Price
95      P024      5        SoundWave Cookbook        Books  338.66
96      P014      4         ActiveWear Jacket     Clothing   26.26
97      P031      4      SoundWave Headphones  Electronics  196.40
98      P044      4  ActiveWear Running Shoes     Clothing   18.82
99      P099      4    SoundWave Mystery Book        Books  354.29
---------------------------------------------------- |


Top 5 most profitable products (Profit wise):
---------------------------------------------------- |

   ProductID  Count            ProductName     Category   Price  \
1       P029     17     TechPro Headphones  Electronics  433.64
3       P079     16         ActiveWear Rug   Home Decor  417.37
6       P048     15       TechPro Cookbook        Books  416.40
2       P062     16        HomeSense Novel        Books  374.16
14      P083     13  ActiveWear Smartwatch  Electronics  455.72

    TotalProfitPerProduct
1                 7371.88
3                 6677.92
6                 6246.00
2                 5986.56
14                5924.36
---------------------------------------------------- |


Top 5 least profitable products (Profit wise):
---------------------------------------------------- |

   ProductID  Count               ProductName     Category  Price  \
90      P070      6         HomeSense T-Shirt     Clothing  48.69
42      P073     10      ComfortLiving Laptop  Electronics  26.99
63      P056      8      SoundWave Smartwatch  Electronics  16.08
96      P014      4         ActiveWear Jacket     Clothing  26.26
98      P044      4  ActiveWear Running Shoes     Clothing  18.82

    TotalProfitPerProduct
90                292.14
42                269.90
63                128.64
```

```
96    105.04
98     75.28
```



	ProductID	Count	ProductName	Category	Price	TotalProfitPerProduct
0	P059	19	SoundWave Jeans	Clothing	303.20	5760.80
1	P029	17	TechPro Headphones	Electronics	433.64	7371.88
2	P062	16	HomeSense Novel	Books	374.16	5986.56
3	P079	16	ActiveWear Rug	Home Decor	417.37	6677.92
4	P054	16	SoundWave Cookbook	Books	57.30	916.80

Next steps: Generate code with product_analysis | View recommended plots | New interactive she

Average transaction value & Average price of all products

```
# @title Average transaction value & Average price of all products
print("Average transaction value : ", round(transactions['TotalValue'].mean()))
print("\nMaximum transaction value : ", round(transactions['TotalValue'].max()))
print("\nMinimum transaction value : ", round(transactions['TotalValue'].min()))

print("\n\n------------------------------------------------------------------------
# Average price of all products
print("\nAverage price of all products : ", round(transactions['Price'].mean()))
print("\nMaximum price of products : ", round(transactions['Price'].max()))
print("\nMinimum price of products : ", round(transactions['Price'].min()))
```

```
Average transaction value :  690

Maximum transaction value :  1991

Minimum transaction value :  16


------------------------------------------------------------------------


Average price of all products :  273

Maximum price of products :  498

Minimum price of products :  16
```

```
# Calculating total spending per customer
customer_spending = pd.merge(customer, transactions, on='CustomerID', how='inner'
customer_spending_value = customer_spending.groupby('CustomerID')['TotalValue'].s
customer_spending_value.columns = ['CustomerID', 'TotalValue']
customer_spending_value = pd.merge(customer, customer_spending_value , on='Custom


# Finding customers with the highest total spending
# top_spenders = customer_spending.sort_values(ascending=False)
print("\n ================================= \n Top 5 most spending customers : \n
print(customer_spending_value.head(5))


print("\n ================================= \n Top 5 least spending customers : \
print(customer_spending_value.tail(5))

print("\n ================================= \n Most spending region : \n ========
print(customer_spending_value.groupby('Region')['TotalValue'].sum().sort_values(a


print("\n ================================= \n Least spending region : \n =======
```

```
print(customer_spending_value.groupby('Region')['TotalValue'].sum().sort_values(a


print("\n ================================= \n Average spending of customer : ",r

for r in customer_spending_value['Region'].unique():
  data = customer_spending_value[customer_spending_value['Region'] == r ]
  print("\n ================================= \n")
  print(f"Average spending of customer in {r} : {round(data['TotalValue'].mean())
  print(f"Maximum spending of customer in {r} : {round(data['TotalValue'].max())}
  print(f"Minimum spending of customer in {r} : {round(data['TotalValue'].min())}
  print("\n ================================= \n\n\n")
```

```
Average spending of customer :  3467
=================================
```

```
==================================

Average spending of customer in South America : 3718

Maximum spending of customer in South America : 7573

Minimum spending of customer in South America : 224


==================================
```

```
customer_spending_value
```

	CustomerID	CustomerName	Region	SignupDate	TotalValue
140	C0141	Paul Parsons	Europe	2023-02-23	10673.87
53	C0054	Bruce Rhodes	Asia	2024-09-29	8040.39
64	C0065	Gerald Hines	North America	2024-07-10	7663.70
155	C0156	William Adams	North America	2023-08-19	7634.45
81	C0082	Aimee Taylor	South America	2022-05-13	7572.91
...	...	...	...	...	...
13	C0014	Deborah Wilcox	Europe	2024-06-22	318.66
150	C0151	Amber Gonzalez	South America	2024-11-22	223.96
96	C0097	Tina Ford	Asia	2023-12-18	137.54
32	C0033	Tyler Holt	North America	2024-08-04	132.64
59	C0060	James Murphy	Europe	2022-04-22	82.36

199 rows × 5 columns

Next steps: Generate code with customer_spending_value | View recommended plots | New inter

```
print("Most transaction happened region's : \n")

plt.figure(figsize=(12, 6))
ax = customer_spending['Region'].value_counts().plot(kind='bar', color=sns.palett

for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, 10),
                textcoords = 'offset points')

plt.show()
```

```
Most transaction happened region's :
```