MARKETING ANALYSIS- AJAY KUMAR

Analysis of Power BI Graphs with SQL Queries and Python Code

Key Metrics (Number of Campaigns, Total Amount, Total Revenue)

SQL Query:

```
sql
SELECT
  COUNT (DISTINCT CampaignID) AS No_of_Campaign,
  SUM(MntTotal) AS Sum_of_MntTotal,
  SUM(TotalRevenue) AS Total_Revenue
FROM
  MarketingData;
```

Python Code:

```
python
import pandas as pd

# Assuming marketing_data is a DataFrame containing the relevant data
no_of_campaign = marketing_data['CampaignID'].nunique()
sum_of_mnttotal = marketing_data['MntTotal'].sum()
total_revenue = marketing_data['TotalRevenue'].sum()

print("No. of Campaign:", no_of_campaign)
print("Sum of MntTotal:", sum_of_mnttotal)
print("Total Revenue:", total_revenue)
```

Explanation: The SQL query and Python code both calculate the number of distinct campaigns, the total amount spent, and the total revenue generated from the marketing data.

Campaign Performance by Product Categories

SQL Query:

```
sql
SELECT
  CampaignID,
  SUM(FishProducts) AS FishProducts,
  SUM(Fruits) AS Fruits,
  SUM(MeatProducts) AS MeatProducts,
  SUM(Wines) AS Wines,
  SUM(SweetProducts) AS SweetProducts
FROM
  MarketingData
GROUP BY
  CampaignID;
```

Python Code:

python

Copy code

```
campaign_performance = marketing_data.groupby('CampaignID').sum()[['FishProducts', 'Fruits', 'MeatProducts', 'Wines', 'SweetProducts']]
print(campaign_performance)
```

Explanation: The SQL query and Python code group the marketing data by campaign ID and sum the sales of various product categories (Fish Products, Fruits, Meat Products, Wines, and Sweet Products) for each campaign.

Purchase Distribution (Regular Products vs. Gold Products)

SQL Query:

```
sql
SELECT
  SUM(CASE WHEN ProductType = 'Regular' THEN MntTotal ELSE 0 END) AS RegularProds,
  SUM(CASE WHEN ProductType = 'Gold' THEN MntTotal ELSE 0 END) AS GoldProds
FROM
  MarketingData;
```

Python Code:

```
python
regular_prods = marketing_data[marketing_data['ProductType'] == 'Regular']['MntTotal'].sum()
gold_prods = marketing_data[marketing_data['ProductType'] == 'Gold']['MntTotal'].sum()

print("Regular Products:", regular_prods)
print("Gold Products:", gold_prods)
```

Explanation: The SQL query and Python code calculate the total amount spent on regular and gold products separately by summing the respective amounts.

Modes of Purchases

SQL Query:

```
sql
ELECT
  SUM(CatalogPurchases) AS CatalogPurchases,
  SUM(DealsPurchases) AS DealsPurchases,
  SUM(StorePurchases) AS StorePurchases,
  SUM(WebPurchases) AS WebPurchases
FROM
  MarketingData;
```

Python Code:

```
python
modes_of_purchases = marketing_data[['CatalogPurchases', 'DealsPurchases', 'StorePurchases', 'WebPurchases']].sum()
print(modes_of_purchases)
```

Explanation: The SQL query and Python code calculate the total number of purchases made through different modes: Catalog, Deals, Store, and Web.

Total Purchases by Age Group

SQL Query:

```
sql
SELECT
  AgeGroup,
  SUM(MntTotal) AS MntTotal,
  SUM(Wines) AS Wines,
  SUM(Fruits) AS Fruits,
  SUM(FishProducts) AS FishProducts,
  SUM(SweetProducts) AS SweetProducts,
  SUM(MeatProducts) AS MeatProducts
FROM
  MarketingData
GROUP BY
  AgeGroup;
```

Python Code:

python

```
age_group_purchases = marketing_data.groupby('AgeGroup').sum()[['MntTotal', 'Wines', 'Fruits', 'FishProducts', 'SweetProducts', 'MeatProducts']]

print(age_group_purchases)
```

Explanation: The SQL query and Python code group the marketing data by age group and sum the total purchases for various product categories (Wines, Fruits, Fish Products, Sweet Products, Meat Products).

Other Purchases by Age Group and Marital Status

SQL Query:

sql

```
SELECT
  AgeGroup,
  MaritalStatus,
  SUM(OtherPurchases) AS OtherPurchases
FROM
  MarketingData
GROUP BY
  AgeGroup, MaritalStatus;
```

Python Code:

python

```
other_purchases_by_age_marital = marketing_data.groupby(['AgeGroup', 'MaritalStatus'])['OtherPurchases'].sum().unstack()

print(other_purchases_by_age_marital)
```

Explanation: The SQL query and Python code group the marketing data by age group and marital status, summing the "Other Purchases" for each combination.

Count of Campaigns by Marital Status

SQL Query:

sql

```
SELECT
  MaritalStatus,
  COUNT(DISTINCT CampaignID) AS Count_of_Campaign
FROM
  MarketingData
```

```
GROUP BY
  MaritalStatus;
```

Python Code:

```python
count_campaigns_marital_status = marketing_data.groupby('MaritalStatus')['CampaignID'].nunique()
print(count_campaigns_marital_status)
```

Explanation: The SQL query and Python code count the number of distinct campaigns grouped by marital status.

Count of Campaigns by Education Level

SQL Query:

```sql
SELECT
  Education,
  COUNT(DISTINCT CampaignID) AS Count_of_Campaign
FROM
  MarketingData
GROUP BY
  Education;
```

Python Code:

```python
count_campaigns_education = marketing_data.groupby('Education')['CampaignID'].nunique()
print(count_campaigns_education)
```

Explanation: The SQL query and Python code count the number of distinct campaigns grouped by education level.

Recency Analysis by Marital Status, Age Groups, and Income Bins

SQL Query:

```sql
SELECT
  MaritalStatus,
  COUNT(Recency) AS Count_of_Recency
FROM
  MarketingData
GROUP BY
```

```
  MaritalStatus;


SELECT
  AgeGroup,
  COUNT(Recency) AS Count_of_Recency
FROM
  MarketingData
GROUP BY
  AgeGroup;


SELECT
  IncomeBins,
  COUNT(Recency) AS Count_of_Recency
FROM
  MarketingData
GROUP BY
  IncomeBins;
```

Python Code:

python

```
recency_by_marital_status = marketing_data.groupby('MaritalStatus')['Recency'].count()
recency_by_age_group = marketing_data.groupby('AgeGroup')['Recency'].count()
recency_by_income_bins = marketing_data.groupby('IncomeBins')['Recency'].count()


print(recency_by_marital_status)
print(recency_by_age_group)
print(recency_by_income_bins)
```

Explanation: The SQL queries and Python code count the number of recency records grouped by marital status, age group, and income bins.

Key Performance Indicators (KPIs) for the Project

1. **Number of Campaigns:** Total distinct campaigns run within the data set.
    - **Calculation:** COUNT(DISTINCT CampaignID)
2. **Total Amount (MntTotal):** Sum of the total amounts spent across all campaigns.

- **Calculation:** SUM(MntTotal)

3. **Total Revenue:** Sum of the total revenue generated from all campaigns.
   - **Calculation:** SUM(TotalRevenue)
4. **Campaign Performance by Product Categories:** Total sales of different product categories per campaign.
   - **Calculation:** SUM(FishProducts), SUM(Fruits), SUM(MeatProducts), SUM(Wines), SUM(SweetProducts)
5. **Purchase Distribution:** Distribution of purchases between regular and gold products.
   - **Calculation:** SUM(CASE WHEN ProductType = 'Regular' THEN MntTotal ELSE 0 END) AS RegularProds, SUM(CASE WHEN ProductType = 'Gold' THEN MntTotal ELSE 0 END) AS GoldProds
6. **Modes of Purchases:** Total number of purchases made through different modes (Catalog, Deals, Store, Web).
   - **Calculation:** SUM(CatalogPurchases), SUM(DealsPurchases), SUM(StorePurchases), SUM(WebPurchases)
7. **Total Purchases by Age Group:** Total purchases made by different age groups for various product categories.
   - **Calculation:** SUM(MntTotal), SUM(Wines), SUM(Fruits), SUM(FishProducts), SUM(SweetProducts), SUM(MeatProducts) GROUP BY AgeGroup
8. **Other Purchases by Age Group and Marital Status:** Sum of "Other Purchases" grouped by age group and marital status.
   - **Calculation:** SUM(OtherPurchases) GROUP BY AgeGroup, MaritalStatus
9. **Count of Campaigns by Marital Status:** Number of distinct campaigns grouped by marital status.
   - **Calculation:** COUNT(DISTINCT CampaignID) GROUP BY MaritalStatus
10. **Count of Campaigns by Education Level:** Number of distinct campaigns grouped by education level.
    - **Calculation:** COUNT(DISTINCT CampaignID) GROUP BY Education
11. **Recency Analysis:** Count of recency records grouped by marital status, age groups, and income bins.
    - **Calculation:** ` COUNT(Recency) GROUP BY MaritalStatus, AgeGroup