In [ ]:

```
import pandas as pd
```

In [ ]:

```
mtn=pd.read_csv('/content/mtn_customer_churn.csv')
mtn
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	Subscription Plan	Unit Price
0	CUST0001	Ngozi Berry	Jan-25	27	Kwara	4G Router	Male	2	Fair	2	165GB Monthly Plan	35000
1	CUST0002	Zainab Baker	Mar-25	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	12.5GB Monthly Plan	5500
2	CUST0003	Saidu Evans	Mar-25	21	Sokoto	5G Broadband Router	Male	1	Poor	60	150GB FUP Monthly Unlimited	20000
3	CUST0003	Saidu Evans	Mar-25	21	Sokoto	Mobile SIM Card	Male	1	Poor	60	1GB+1.5mins Daily Plan	500
4	CUST0003	Saidu Evans	Mar-25	21	Sokoto	Broadband MiFi	Male	1	Poor	60	30GB Monthly Broadband Plan	9000
...	...	...	...	...	...	...	...	...	...	...	...	...
969	CUST0498	Shehu Harris	Jan-25	72	Osun	4G Router	Female	3	Good	42	25GB Monthly Plan	9000
970	CUST0499	Tega Hood	Mar-25	41	Bayelsa	5G Broadband Router	Female	2	Fair	20	30GB Monthly Broadband Plan	9000
971	CUST0500	Oghene Hamilton	Feb-25	67	Kebbi	4G Router	Male	4	Very Good	31	165GB Monthly Plan	35000
972	CUST0500	Oghene Hamilton	Feb-25	67	Kebbi	Mobile SIM Card	Male	4	Very Good	31	7GB Monthly Plan	3500
973	CUST0500	Oghene Hamilton	Feb-25	67	Kebbi	Broadband MiFi	Male	4	Very Good	31	150GB FUP Monthly Unlimited	20000

974 rows × 17 columns

```
In [ ]: mtn.columns
```

```
Out[ ]: Index(['Customer ID', 'Full Name', 'Date of Purchase', 'Age', 'State',
               'MTN Device', 'Gender', 'Satisfaction Rate', 'Customer Review',
               'Customer Tenure in months', 'Subscription Plan', 'Unit Price',
               'Number of Times Purchased', 'Total Revenue', 'Data Usage',
               'Customer Churn Status', 'Reasons for Churn'],
              dtype='object')
```

```
In [ ]: #Data cleaning
        #checking for missing values and datatypes
```

```
In [ ]: mtn.info() #checking the datatypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 974 entries, 0 to 973
Data columns (total 17 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Customer ID                974 non-null    object 
 1   Full Name                  974 non-null    object 
 2   Date of Purchase           974 non-null    object 
 3   Age                        974 non-null    int64  
 4   State                      974 non-null    object 
 5   MTN Device                 974 non-null    object 
 6   Gender                     974 non-null    object 
 7   Satisfaction Rate          974 non-null    int64  
 8   Customer Review            974 non-null    object 
 9   Customer Tenure in months  974 non-null    int64  
 10  Subscription Plan          974 non-null    object 
 11  Unit Price                 974 non-null    int64  
 12  Number of Times Purchased  974 non-null    int64  
 13  Total Revenue              974 non-null    int64  
 14  Data Usage                 974 non-null    float64
 15  Customer Churn Status      974 non-null    object 
 16  Reasons for Churn          284 non-null    object 
dtypes: float64(1), int64(6), object(10)
memory usage: 129.5+ KB
```

```
In [ ]: mtn.isna().sum()#checking for missing values
```

```
Out[ ]:
                                 0
                 Customer ID     0
                   Full Name     0
            Date of Purchase     0
                         Age     0
                       State     0
                  MTN Device     0
                      Gender     0
           Satisfaction Rate     0
             Customer Review     0
  Customer Tenure in months     0
           Subscription Plan     0
                  Unit Price     0
  Number of Times Purchased     0
               Total Revenue     0
                  Data Usage     0
       Customer Churn Status     0
          Reasons for Churn   690

dtype: int64
```

```
In [ ]: mtn.dtypes
```

Out[ ]:

	0
Customer ID	object
Full Name	object
Date of Purchase	object
Age	int64
State	object
MTN Device	object
Gender	object
Satisfaction Rate	int64
Customer Review	object
Customer Tenure in months	int64
Subscription Plan	object
Unit Price	int64
Number of Times Purchased	int64
Total Revenue	int64
Data Usage	float64
Customer Churn Status	object
Reasons for Churn	object

dtype: object

In [ ]:

```
#Data cleaning. changing date of purchase to datetimes
pd.to_datetime(mtn['Date of Purchase'],format='%b-%y')
```

Out[ ]:

	Date of Purchase
0	2025-01-01
1	2025-03-01
2	2025-03-01
3	2025-03-01
4	2025-03-01
...	...
969	2025-01-01
970	2025-03-01
971	2025-02-01
972	2025-02-01
973	2025-02-01

974 rows × 1 columns

dtype: datetime64[ns]

In [ ]:

```
mtn["Date of Purchase"]=pd.to_datetime(mtn['Date of Purchase'],format='%b-%y')#Effecting change inside the data
```

In [ ]:

```
mtn.dtypes #checking if change has been effected.
```

Out[ ]:

	0
Customer ID	object
Full Name	object
Date of Purchase	datetime64[ns]
Age	int64
State	object
MTN Device	object
Gender	object
Satisfaction Rate	int64
Customer Review	object
Customer Tenure in months	int64
Subscription Plan	object
Unit Price	int64
Number of Times Purchased	int64
Total Revenue	int64
Data Usage	float64
Customer Churn Status	object
Reasons for Churn	object

dtype: object

In [ ]:

```
#checking for duplicates
mtn.duplicated().sum()
```

Out[ ]:

```
np.int64(0)
```

ADDITION OF COLUMNS. FIRST COLUMN IS THE "HIGH VALUE CUSTOMER". USING THE TOTAL REVENUE COLUMN REVENUE GREATER THAN THE MEDIAN IS YES AND BELOW THE MEDIAN IS NO.

```
In [ ]: mtn['Total Revenue'].median()#calculating median
```

```
Out[ ]: 108000.0
```

```
In [ ]: #conditonal statement
        import numpy as np

        mtn["High Value Customer"] = np.where(
            mtn["Total Revenue"] > mtn["Total Revenue"].median(),
            "Yes",
            "No")
```

```
In [ ]: mtn.head()
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	Subscription Plan	Unit Price	Nu of Purcl
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	165GB Monthly Plan	35000	
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	12.5GB Monthly Plan	5500	
2	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	5G Broadband Router	Male	1	Poor	60	150GB FUP Monthly Unlimited	20000	
3	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	Mobile SIM Card	Male	1	Poor	60	1GB+1.5mins Daily Plan	500	
4	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	Broadband MiFi	Male	1	Poor	60	30GB Monthly Broadband Plan	9000	

ADDING ANOTHER COLUMN "DATA USAGE PER MONTH", BY DIVIDING THE DATA USAGE BY THE CUSTOMER TENURE IN MONTHS.

In [ ]:

```
mtn["Data Usage Per Month"]=mtn['Data Usage']/mtn['Customer Tenure in months']
```

In [ ]:

```
mtn.head()
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	Subscription Plan	Unit Price	Nu of Purcl
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	165GB Monthly Plan	35000	
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	12.5GB Monthly Plan	5500	
2	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	5G Broadband Router	Male	1	Poor	60	150GB FUP Monthly Unlimited	20000	
3	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	Mobile SIM Card	Male	1	Poor	60	1GB+1.5mins Daily Plan	500	
4	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	Broadband MiFi	Male	1	Poor	60	30GB Monthly Broadband Plan	9000	

In [ ]:

```
#rounding off the data usage per month to 2 decimal places
round(mtn['Data Usage Per Month'],2)
```

Out[ ]:

	Data Usage Per Month
0	22.24
1	0.90
2	0.16
3	3.28
4	1.27
...	...
969	0.19
970	7.71
971	4.67
972	5.11
973	2.78

974 rows × 1 columns

dtype: float64

In [ ]:

```
mtn["Data Usage Per Month"]=round(mtn['Data Usage Per Month'],2)
```

In [ ]:

```
mtn.head()
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	Subscription Plan	Unit Price	Nu of Purcl
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	165GB Monthly Plan	35000	
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	12.5GB Monthly Plan	5500	
2	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	5G Broadband Router	Male	1	Poor	60	150GB FUP Monthly Unlimited	20000	
3	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	Mobile SIM Card	Male	1	Poor	60	1GB+1.5mins Daily Plan	500	
4	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	Broadband MiFi	Male	1	Poor	60	30GB Monthly Broadband Plan	9000	

ADDING ANOTHER COLUMN "REVENUE PER MONTH", BY DIVIDING THE TOTAL REVENUE BY THE CUSTOMER TENURE IN MONTHS.

In [ ]:

```
mtn["Revenue Per Month"]=round(mtn['Total Revenue']/mtn['Customer Tenure in months'])
```

In [ ]:

```
mtn.head(2)
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	Subscription Plan	Unit Price	Numbe of Time Purchase
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	165GB Monthly Plan	35000	19
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	12.5GB Monthly Plan	5500	12

ADDING ANOTHER COLUMN "PLAN TYPE CATEGORY", THE GOAL IS TO CLASSIFY EACH CUSTOMER'S SUBSCRIPTION PLAN INTO EASY TO ANALYSE CATEGORIES(DAILY,MONTHLY,UNLIMITED AND OTHER

In [ ]:

```
mtn["Plan Type Category"] = mtn["Subscription Plan"].str.extract(
    r'(\bDaily\b|\bMonthly\b|\bUnlimited\b)', expand=False
).fillna("Other")
```

In [ ]:

```
mtn.head(2)
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	...	Unit Price	Number of Times Purchased	Tota Revenu
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	...	35000	19	665000
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	...	5500	12	66000

2 rows × 21 columns

ADDING ANOTHER COLUMN "AGE GROUP" TO FURTHER CLASSIFY THE AGE

In [ ]:

```
#unique values in the age column
mtn.Age.unique()
```

Out[ ]:

```
array([27, 16, 21, 36, 57, 74, 24, 53, 35, 23, 72, 78, 22, 70, 46, 54, 50,
       49, 42, 58, 55, 65, 67, 52, 19, 44, 69, 75, 64, 76, 40, 32, 25, 68,
       28, 48, 61, 51, 18, 80, 59, 31, 43, 33, 66, 71, 79, 62, 20, 63, 26,
       56, 39, 47, 30, 41, 45, 17, 34, 29, 37, 73, 77, 60, 38])
```

In [ ]:

```
mtn["Age Group"] = ""
```

In [ ]:

```
mtn.columns
```

Out[ ]:

```
Index(['Customer ID', 'Full Name', 'Date of Purchase', 'Age', 'State',
       'MTN Device', 'Gender', 'Satisfaction Rate', 'Customer Review',
       'Customer Tenure in months', 'Subscription Plan', 'Unit Price',
       'Number of Times Purchased', 'Total Revenue', 'Data Usage',
       'Customer Churn Status', 'Reasons for Churn', 'High Value Customer',
       'Data Usage Per Month', 'Revenue Per Month', 'Plan Type Category',
       'Age Group'],
      dtype='object')
```

In [ ]:

```
#conditional statement for age group
def classify_age(age):
    if age < 24:
        return "Youth"
    elif age < 40:
        return "Adult"
    elif age < 60:
        return "Mid-Age"
    else:
        return "Senior"

mtn["Age Group"] = mtn["Age"].apply(classify_age)
```

In [ ]:

```
mtn.head(2)
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	...	Number of Times Purchased	Total Revenue	Dat Usag
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	...	19	665000	44.4
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	...	12	66000	19.7

2 rows × 22 columns

ADDING ANOTHER COLUMN "DEVICE CATEGORY" TO FURTHER CLASSIFY THE DEVICES

In [ ]:

```
def categorize_device(device):
    device = device.lower()  # make text lowercase for easy matching

    if "router" in device:
        return "Router"
    elif "mifi" in device:
        return "MiFi"
    elif "sim" in device:
        return "Mobile SIM"
    else:
        return "Other"

mtn["Device Category"] = mtn["MTN Device"].apply(categorize_device)
```

In [ ]:

```
mtn.head(3)
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	...	Total Revenue	Data Usage	Cust C S
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	...	665000	44.48	
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	...	66000	19.79	
2	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	5G Broadband Router	Male	1	Poor	60	...	160000	9.64	

3 rows × 23 columns

ADDING A COLUMN "MONTH OF PURCHASE" TO HELP WITH THE MONTHLY ANALYSIS

In [ ]:

```
mtn["Month Of Purchase"] = mtn["Date of Purchase"].dt.month_name()
```

In [ ]:

```
mtn.head(2)
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	...	Data Usage	Customer Churn Status	Reas for Ch
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	...	44.48	Yes	Relocat
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	...	19.79	Yes	Be Offers fr Competit

2 rows × 24 columns

CREATING KPI'S FOR ANALYSIS

In [ ]:

```
#TOTAL NUMBER OF CUSTOMERS
mtn['Customer ID'].nunique()
```

Out[ ]:

```
496
```

In [ ]:

```
#TOTAL REVENUE
mtn['Total Revenue'].sum()
```

Out[ ]:

```
np.int64(199348200)
```

In [ ]:

```
#TOTAL NUMBER OF CHURNED CUSTOMERS
mtn[mtn["Customer Churn Status"] == "Yes"]["Customer ID"].nunique()
```

Out[ ]:

```
146
```

In [ ]:

```
#TOTAL NUMBER OF CUSTOMERS THAT HAVE NOT CHURNED
mtn[mtn["Customer Churn Status"] == "No"]["Customer ID"].nunique()
```

Out[ ]:

```
350
```

In [ ]:

```
#TOTAL NUMBER OF ORDERS
mtn['Number of Times Purchased'].sum()
```

```
Out[ ]: np.int64(10290)
```

```
In [ ]: (146/496)*100 #% CHURNED
```

```
Out[ ]: 29.435483870967744
```

```
In [ ]: (350/496)*100 # % ACTIVE CUSTOMERS
```

```
Out[ ]: 70.56451612903226
```

```
In [ ]: #MONTHS USED FOR ANALYSIS
        mtn['Month Of Purchase'].unique()
```

```
Out[ ]: array(['January', 'March', 'February'], dtype=object)
```

```
In [ ]: mtn['Plan Type Category'].unique()
```

```
Out[ ]: array(['Monthly', 'Daily', 'Other'], dtype=object)
```

```
In [ ]: mtn.head()
```

Out[ ]:

	Customer ID	Full Name	Date of Purchase	Age	State	MTN Device	Gender	Satisfaction Rate	Customer Review	Customer Tenure in months	...	Data Usage	Customer Churn Status	fc
0	CUST0001	Ngozi Berry	2025-01-01	27	Kwara	4G Router	Male	2	Fair	2	...	44.48	Yes	Re
1	CUST0002	Zainab Baker	2025-03-01	16	Abuja (FCT)	Mobile SIM Card	Female	2	Fair	22	...	19.79	Yes	Off Con
2	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	5G Broadband Router	Male	1	Poor	60	...	9.64	No	
3	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	Mobile SIM Card	Male	1	Poor	60	...	197.05	No	
4	CUST0003	Saidu Evans	2025-03-01	21	Sokoto	Broadband MiFi	Male	1	Poor	60	...	76.34	No	

5 rows × 24 columns

Top 10 Customers by Total Revenue (with Satisfaction rate & Churn Status)

In [ ]:

```
import seaborn as sns
```

In [ ]:

```
import matplotlib.pyplot as plt
```

In [ ]:

```
(mtn.groupby(["Customer ID", "Full Name"], as_index=True)
    .agg({"Total Revenue": "sum",
        "Satisfaction Rate": "mean",
        "Customer Churn Status": "first"})
    .sort_values(by="Total Revenue", ascending=False)
    .head(10))
```

Out[ ]:

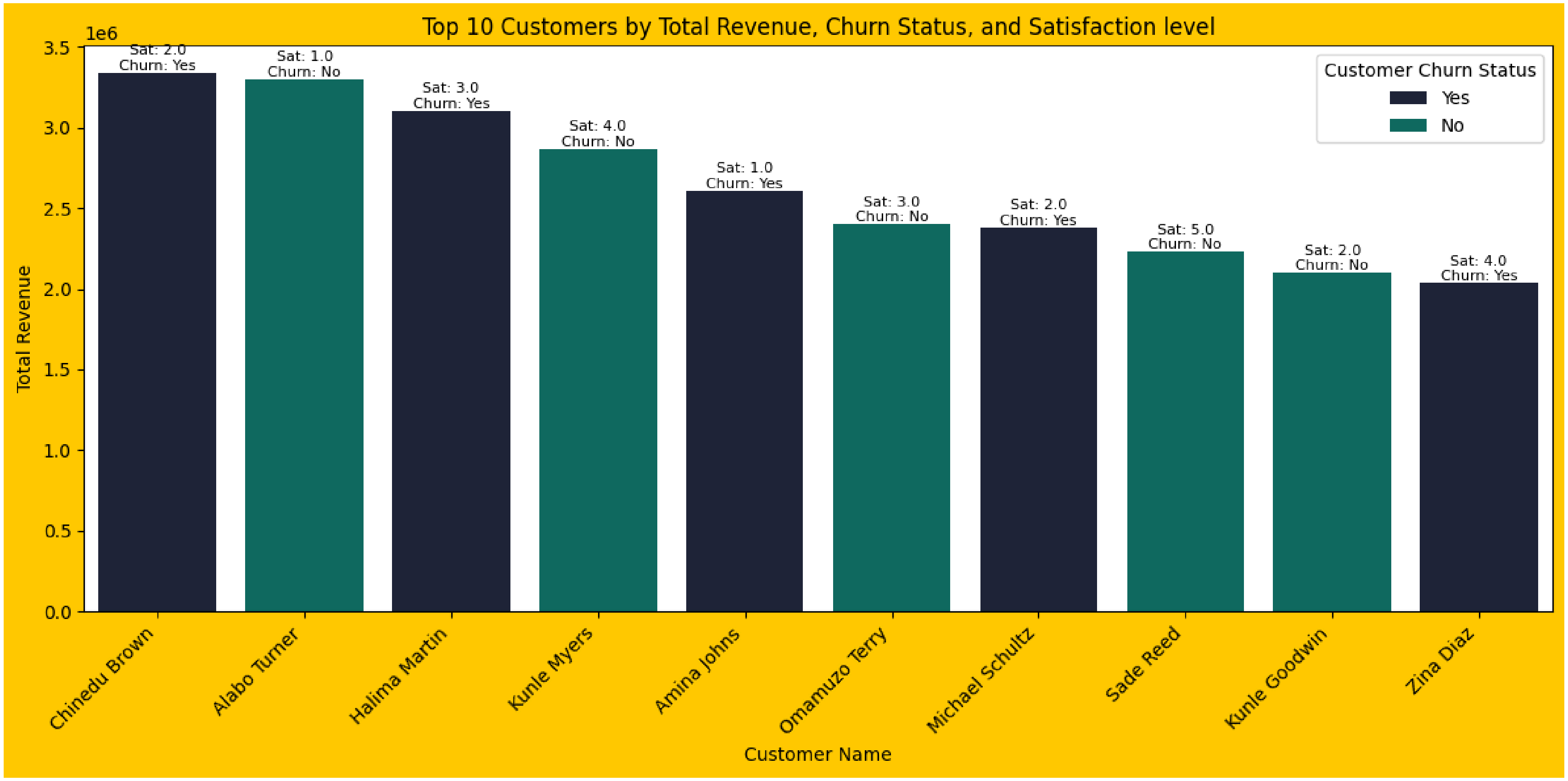
		Total Revenue	Satisfaction Rate	Customer Churn Status
Customer ID	**Full Name**			
CUST0405	**Chinedu Brown**	3340000	2.0	Yes
CUST0085	**Alabo Turner**	3300000	1.0	No
CUST0494	**Halima Martin**	3105000	3.0	Yes
CUST0365	**Kunle Myers**	2870000	4.0	No
CUST0260	**Amina Johns**	2606500	1.0	Yes
CUST0119	**Omamuzo Terry**	2400000	3.0	No
CUST0089	**Michael Schultz**	2381500	2.0	Yes
CUST0481	**Sade Reed**	2230000	5.0	No
CUST0480	**Kunle Goodwin**	2100000	2.0	No
CUST0343	**Zina Diaz**	2034500	4.0	Yes

In [ ]:

```
Top10=(mtn.groupby(["Customer ID", "Full Name"], as_index=True)
    .agg({"Total Revenue": "sum",
        "Satisfaction Rate": "mean",
        "Customer Churn Status": "first"})
    .sort_values(by="Total Revenue", ascending=False)
    .head(10))
```

In [ ]:

```
plt.figure(figsize=(12, 6), facecolor='#FFCC00')
ax = sns.barplot(x=Top10.index.get_level_values('Full Name'), y=Top10["Total Revenue"], hue=Top10["Customer Churn Sta
plt.xticks(rotation=45, ha='right')
plt.title("Top 10 Customers by Total Revenue, Churn Status, and Satisfaction level")
plt.xlabel("Customer Name")
plt.ylabel("Total Revenue")

# Add annotations for Satisfaction Rate and Customer Churn Status
for i, row in Top10.reset_index().iterrows():
    ax.text(i, row["Total Revenue"], f'Sat: {row["Satisfaction Rate"]:.1f}\nChurn: {row["Customer Churn Status"]}', 
plt.tight_layout()
plt.show()

plt.tight_layout()
plt.show()
```



```
<Figure size 640x480 with 0 Axes>
```

CUSTOMER COUNT BY STATE ANALYSIS

In [ ]:

```
customer_count_by_state = mtn.groupby("State")["Customer ID"].nunique().sort_values(ascending = False)
customer_count_by_state.name="Customer Count"
print(customer_count_by_state)
```

```
State
Osun            20
Kogi            20
Bauchi          20
Abia            18
Borno           18
Yobe            17
Abuja (FCT)     17
Enugu           17
Oyo             17
Gombe           16
Benue           16
Imo             16
Ekiti           16
Edo             16
Katsina         16
Zamfara         15
Anambra         15
Nasarawa        15
Sokoto          14
Cross River     14
Kebbi           14
Taraba          13
Plateau         13
Jigawa          13
Niger           12
Adamawa         11
Kwara           11
Akwa Ibom       10
Kano            10
Bayelsa         10
Delta           10
Rivers          10
Ondo             9
Lagos            9
Kaduna           8
Name: Customer Count, dtype: int64
```

TOTAL NUMBER OF ORDERS BASED ON PLAN TYPE CATEGORY AND DEVICE CATEGORY

In [ ]:

```
#find out sum of the number of items purchased from each category
(mtn.groupby(["Plan Type Category","Device Category"])
.agg({"Number of Times Purchased":"sum"})
.sort_values(by="Number of Times Purchased", ascending=False))
```

Out[ ]:

		Number of Times Purchased
Plan Type Category	**Device Category**	
Monthly	**Router**	4158
	MiFi	2556
	Mobile SIM	1513
Other	**Mobile SIM**	995
Daily	**Mobile SIM**	555
Other	**Router**	513

TOTAL NUMBER OF ORDERS/TOTAL REVENUE BASED ON SUBSCRIPTION PLAN

In [ ]:

```
(mtn.groupby("Subscription Plan")
.agg( Total_number_of_orders=("Number of Times Purchased", "sum"),
Total_Revenue=("Total Revenue", "sum"))
.sort_values(by="Total_number_of_orders", ascending=False)
.head(10))
```

Out[ ]:

	Total_number_of_orders	Total_Revenue
Subscription Plan		
150GB FUP Monthly Unlimited	949	18980000
60GB Monthly Broadband Plan	909	13180500
30GB Monthly Broadband Plan	880	7920000
300GB FUP Monthly Unlimited	859	25770000
165GB Monthly Plan	750	26250000
120GB Monthly Broadband Plan	665	15960000
10GB+10mins Monthly Plan	622	2799000
65GB Monthly Plan	582	9312000
25GB Monthly Plan	480	4320000
12.5GB Monthly Plan	406	2233000

TOTAL REVENUE BY PLAN TYPE CATEGORY AND DEVICE CATEGORY

In [ ]:

```
(mtn.groupby(["Plan Type Category", "Device Category"])
   .agg({"Total Revenue": "sum"})
   .sort_values(by="Total Revenue", ascending=False))
```

Out[ ]:

Plan Type Category	Device Category	Total Revenue
Monthly	**Router**	79271000
Other	**Router**	58575000
Monthly	**MiFi**	48067500
	Mobile SIM	12357500
Other	**Mobile SIM**	848000
Daily	**Mobile SIM**	229200

AVERAGE USAGE AND REVENUE BY PLAN TYPE CATEGORY AND DEVICE CATEGORY.

In [ ]:

```
(mtn.groupby(["Plan Type Category", "Device Category"])
.agg( Avg_Data_Usage=("Data Usage", "mean"),
Avg_Data_Per_Month=("Data Usage Per Month", "mean"),
Avg_Revenue_Per_Month=("Revenue Per Month", "mean")).reset_index().round(2))
```

Out[ ]:

	Plan Type Category	Device Category	Avg_Data_Usage	Avg_Data_Per_Month	Avg_Revenue_Per_Month
0	Daily	Mobile SIM	102.74	6.35	231.55
1	Monthly	MiFi	103.46	6.63	12998.30
2	Monthly	Mobile SIM	103.06	6.59	5693.60
3	Monthly	Router	95.55	6.57	15300.01
4	Other	Mobile SIM	97.93	5.69	574.66
5	Other	Router	97.38	8.98	77651.00

AGE GROUP AND THEIR TOTAL AND AVERAGE REVENUE

In [ ]:

```
 (mtn.groupby("Age Group").agg(Total_Revenue=("Total Revenue", "sum"),Avg_Revenue=("Total Revenue", "mean"))
    .round(2).sort_values(by="Total_Revenue", ascending=False))
```

Out[ ]:

	Total_Revenue	Avg_Revenue
Age Group		
Mid-Age	67514000	210981.25
Adult	58811200	219444.78
Senior	53253500	178105.35
Youth	19769500	227235.63

CUSTOMER CHURN STATUS BY AGE GROUP

In [ ]:

```
age_churn_crosstab = pd.crosstab(mtn["Age Group"], mtn["Customer Churn Status"])
age_churn_crosstab
```

Out[ ]:

Customer Churn Status	No	Yes
Age Group		
Adult	176	92
Mid-Age	233	87
Senior	219	80
Youth	62	25

In [ ]:

```
age_churn_crosstab = pd.crosstab(mtn["Age Group"], mtn["Customer Churn Status"])

plt.figure(figsize=(10, 6), facecolor='#FFCC00') # MTN yellow background

age_churn_crosstab.plot(kind='bar', stacked=True, figsize=(10, 6), color=["#00796B", "#1B263B"]) # MTN teal and dark

plt.title("Customer Churn Status by Age Group", fontsize=14, fontweight="bold", color="black", pad=20)
plt.xlabel("Age Group", fontsize=12, color="black")
plt.ylabel("Number of Customers", fontsize=12, color="black")
plt.xticks(rotation=0, color="black")
plt.yticks(color="black")
plt.legend(title="Churn Status", facecolor="#FFCC00", edgecolor="black", labelcolor="black")
plt.tight_layout()
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```

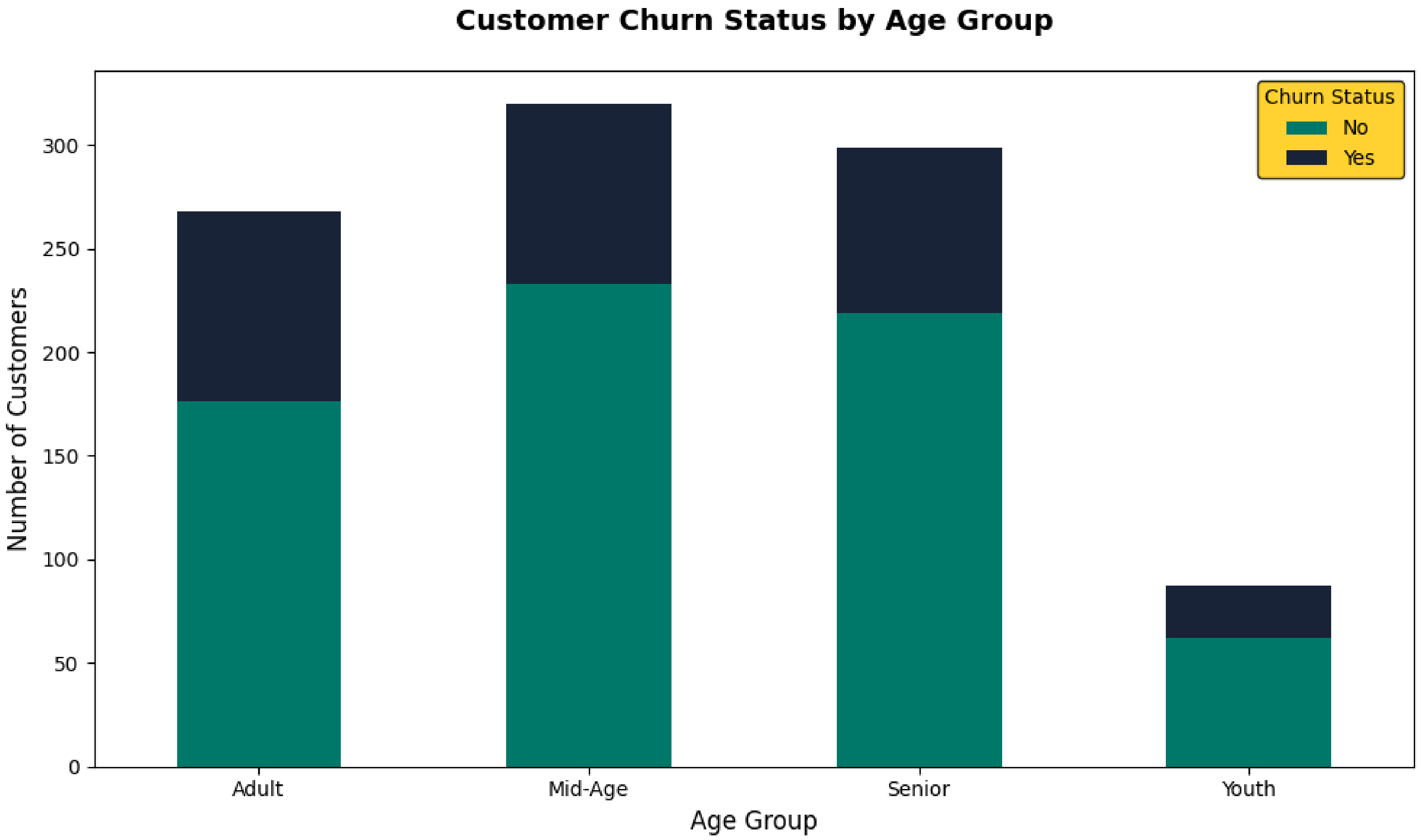
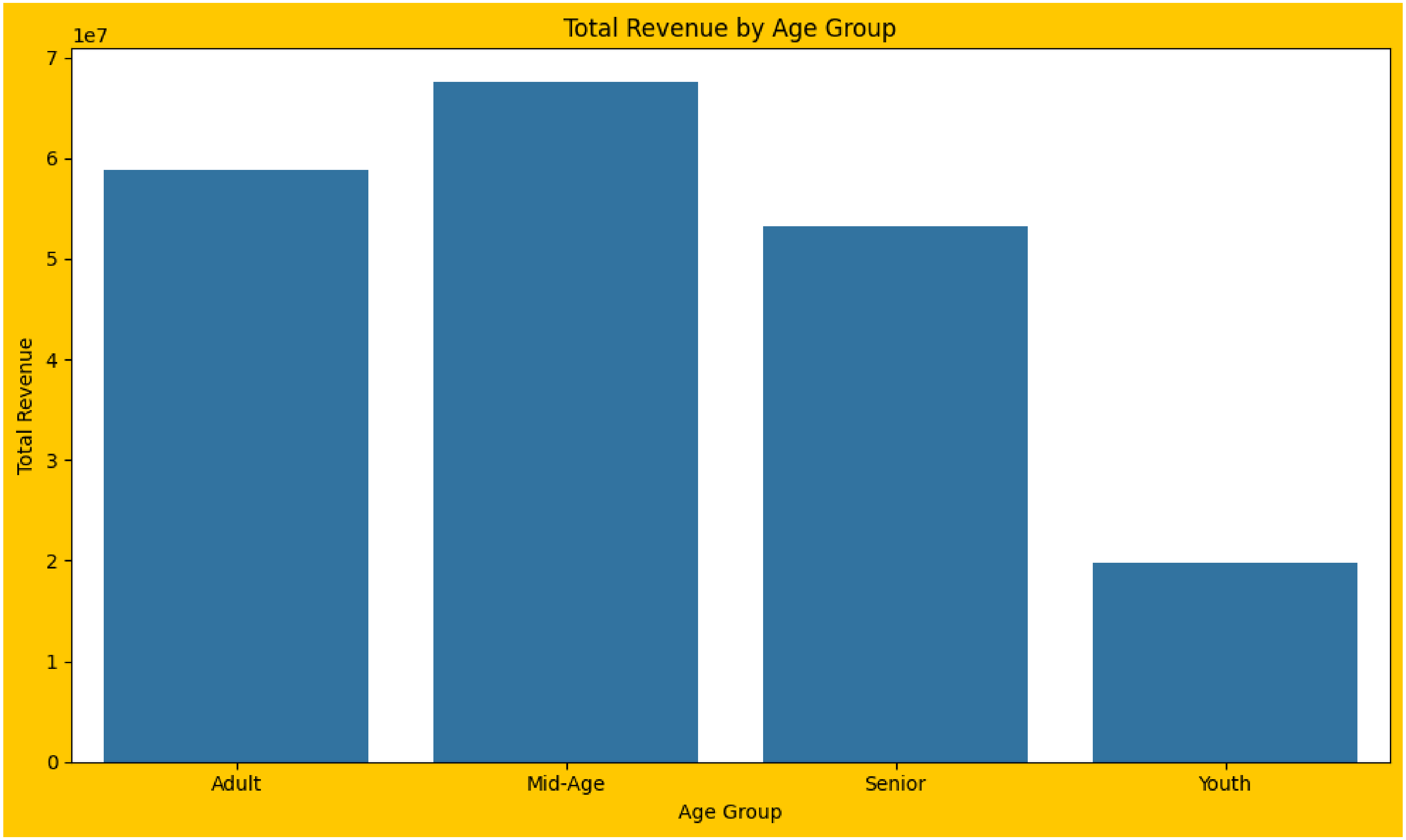


CHART SHOWING TOTAL REVENUE BY AGE GROUP

```
In [ ]: age_group_revenue = mtn.groupby("Age Group")["Total Revenue"].sum().reset_index()

        plt.figure(figsize=(10, 6), facecolor='#FFCC00')
```

```
sns.barplot(x="Age Group", y="Total Revenue", data=age_group_revenue)
plt.title("Total Revenue by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Total Revenue")
plt.tight_layout()
plt.show()
```



MONTHLY ANALYSIS SHOWING TOTAL REVENUE

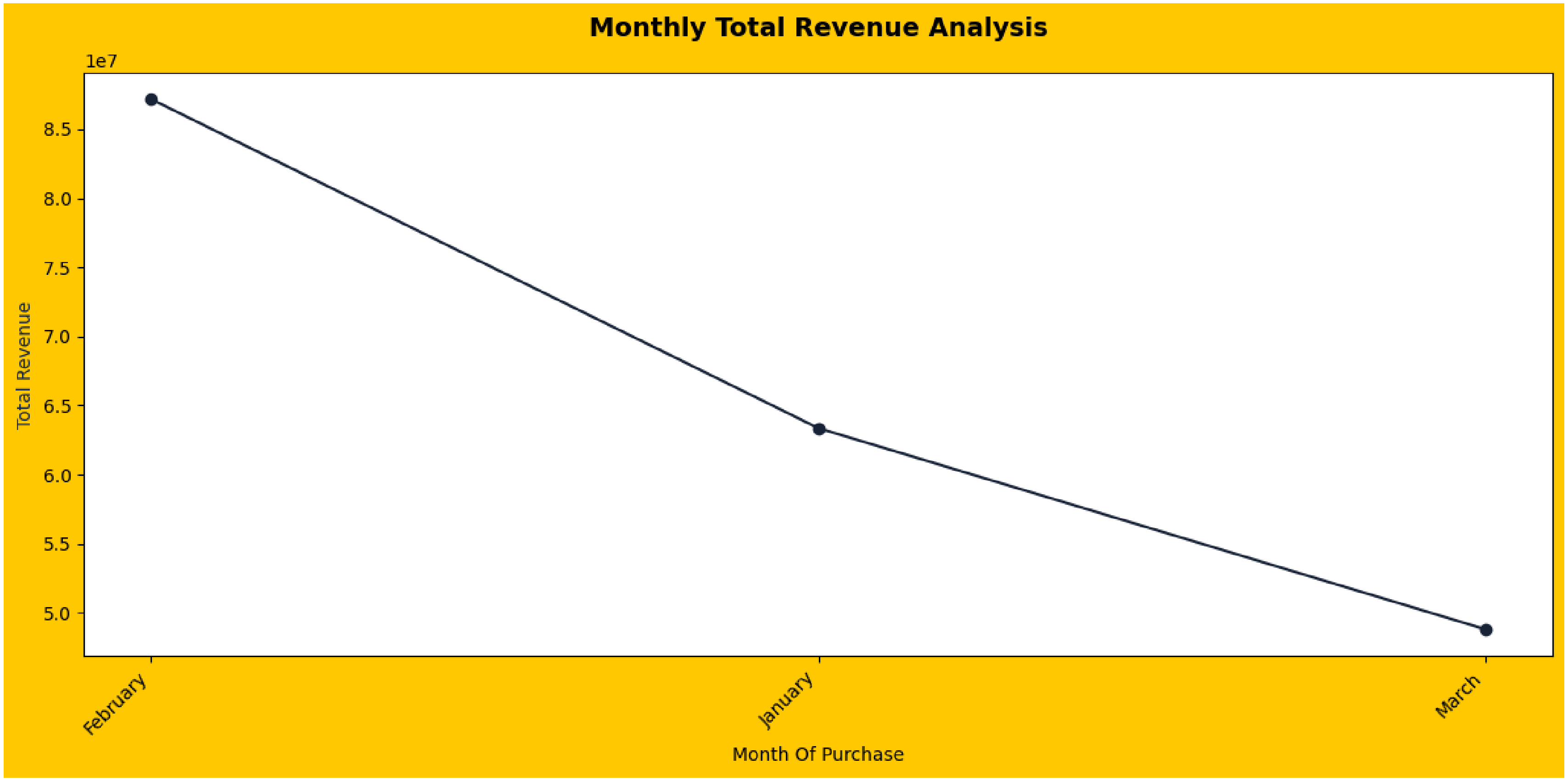
```
In [ ]: display(mtn.groupby("Month Of Purchase")["Total Revenue"].sum().sort_values(ascending=False))
```

	Total Revenue
Month Of Purchase	
February	87173550
January	63366500
March	48808150

dtype: int64

In [ ]:

```python
plt.figure(figsize=(12, 6), facecolor='#FFCC00')

# Plotting Total Revenue
color = '#1B263B' # MTN Dark Blue
monthly_revenue_analysis = mtn.groupby("Month Of Purchase")["Total Revenue"].sum().sort_values(ascending=False)
plt.plot(monthly_revenue_analysis.index, monthly_revenue_analysis.values, color=color, marker='o')
plt.xlabel("Month Of Purchase")
plt.ylabel("Total Revenue", color=color)
plt.title("Monthly Total Revenue Analysis", fontsize=14, fontweight="bold", color="black", pad=20)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



In [ ]:

```
monthly_data_usage = mtn.groupby("Month Of Purchase")["Data Usage"].sum().sort_values(ascending=False)#MONTHLY ANALYS
display(monthly_data_usage)
```

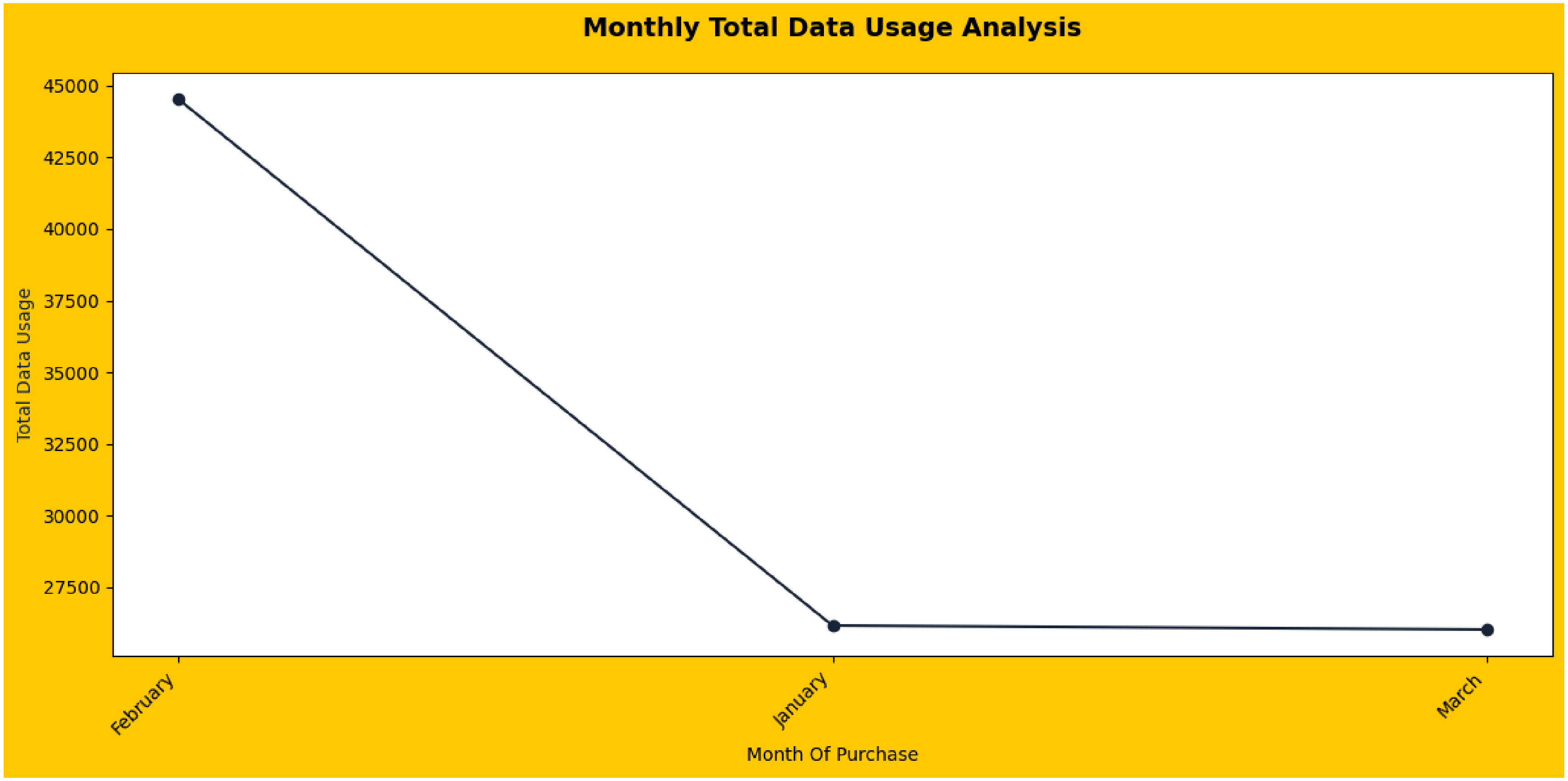
	Data Usage
Month Of Purchase	
February	44534.82
January	26160.81
March	26027.21

dtype: float64

MONTHLY ANALYSIS SHOWING TOTAL DATA USAGE

In [ ]:

```
plt.figure(figsize=(12, 6), facecolor='#FFCC00')

# Plotting Total Data Usage
color = '#1B263B' # MTN Dark Blue
plt.plot(monthly_data_usage.index, monthly_data_usage.values, color=color, marker='o')
plt.xlabel("Month Of Purchase")
plt.ylabel("Total Data Usage", color=color)
plt.title("Monthly Total Data Usage Analysis", fontsize=14, fontweight="bold", color="black", pad=20)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

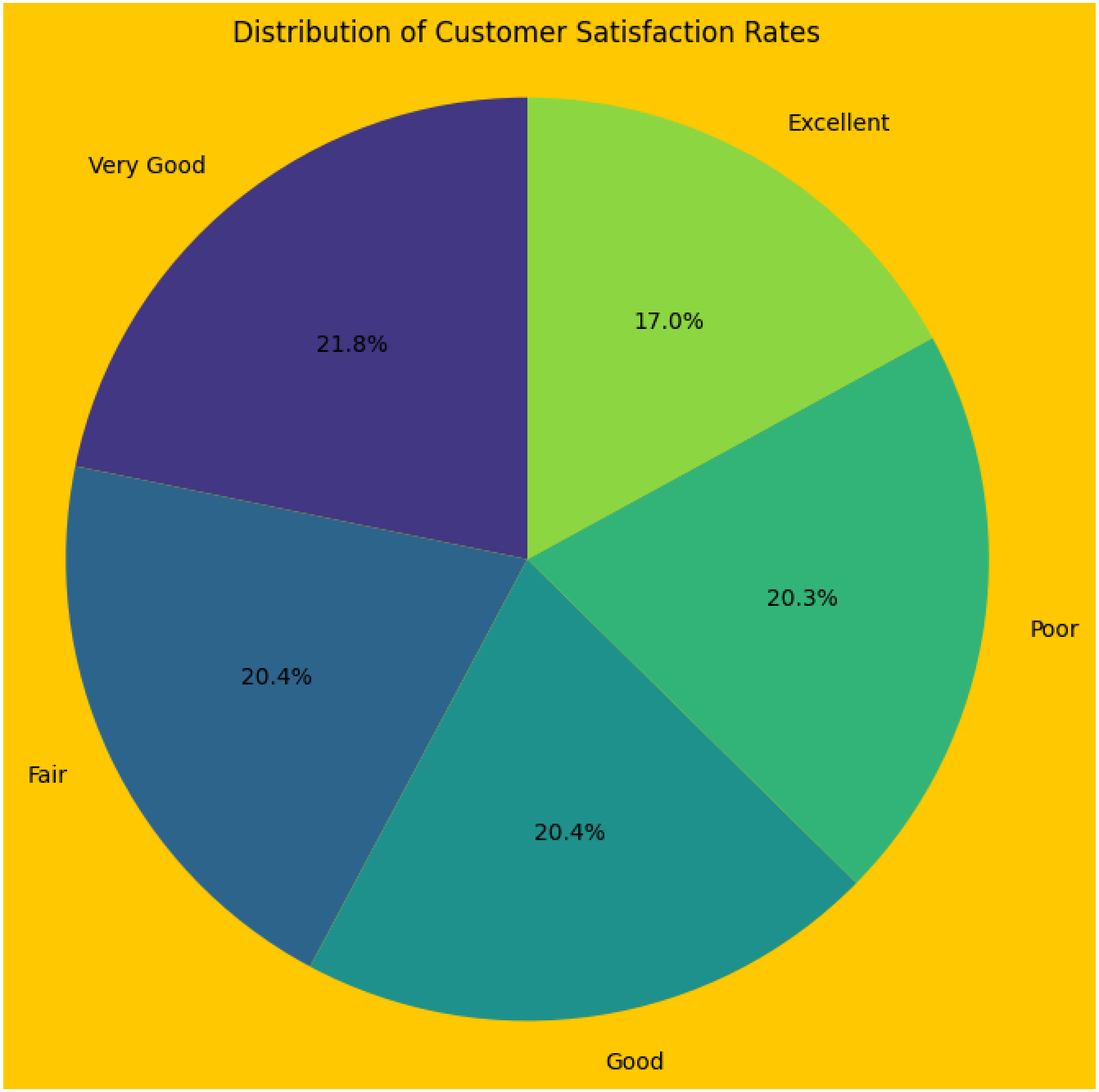


CUSTOMER REVIEW ANALYSIS

```
In [ ]: mtn["Customer Review"].value_counts(normalize=True) * 100
```

```
Out[ ]:
                      proportion
 Customer Review
       Very Good       21.765914
            Fair       20.431211
            Good       20.431211
            Poor       20.328542
       Excellent       17.043121

dtype: float64
```

```
In [ ]: customer_review_counts = mtn["Customer Review"].value_counts(normalize=True) * 100

        plt.figure(figsize=(8, 8), facecolor='#FFCC00')
        plt.pie(customer_review_counts, labels=customer_review_counts.index, autopct='%1.1f%%', startangle=90, colors=sns.col
        plt.title("Distribution of Customer Satisfaction Rates")
        plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
        plt.show()
```

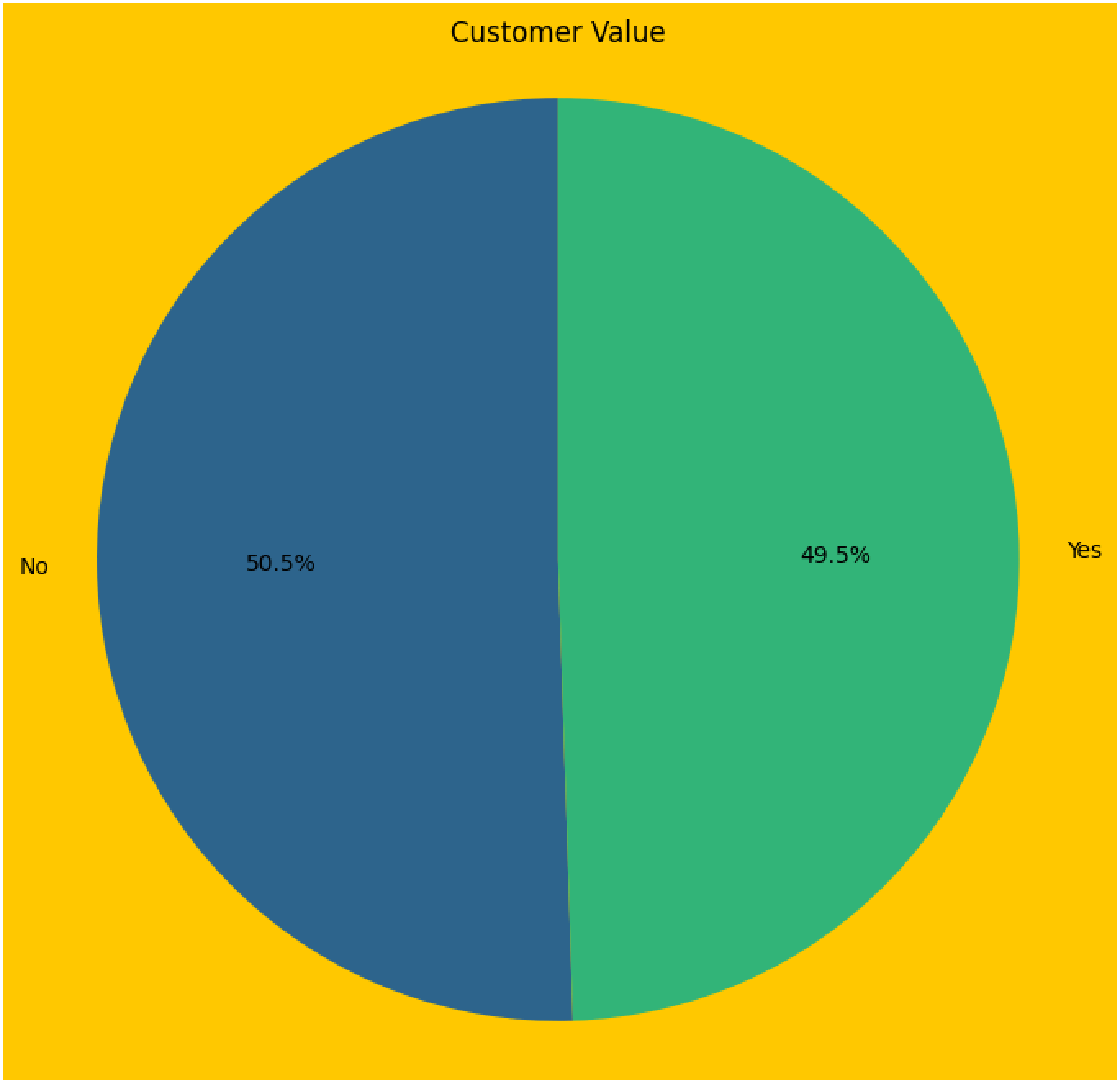


CUSTOMER VALUE ANALYSIS

```
In [ ]: mtn["High Value Customer"].value_counts(normalize=True) * 100
```

```
Out[ ]:
                      proportion
High Value Customer
                 No    50.513347
                Yes    49.486653

dtype: float64
```

```
In [ ]: High_value_customer = mtn["High Value Customer"].value_counts(normalize=True) * 100

        plt.figure(figsize=(8, 8), facecolor='#FFCC00')
        plt.pie(High_value_customer, labels=High_value_customer.index, autopct='%1.1f%%', startangle=90, colors=sns.color_pal
        plt.title("Customer Value")
        plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
        plt.show()
```



GENDER ANALYSIS BY COUNT AND TOTAL REVENUE

In [ ]:

```
(mtn.drop_duplicates(subset=['Customer ID']).groupby('Gender').agg(Total_Revenue=("Total Revenue", "sum"),Number=("Ge
    .sort_values(by="Total_Revenue", ascending=False))
```

Out[ ]:

	Total_Revenue	Number
Gender		
Male	49667200	246
Female	45890250	250

REASONS FOR CUSTOMER CHURN

In [ ]:

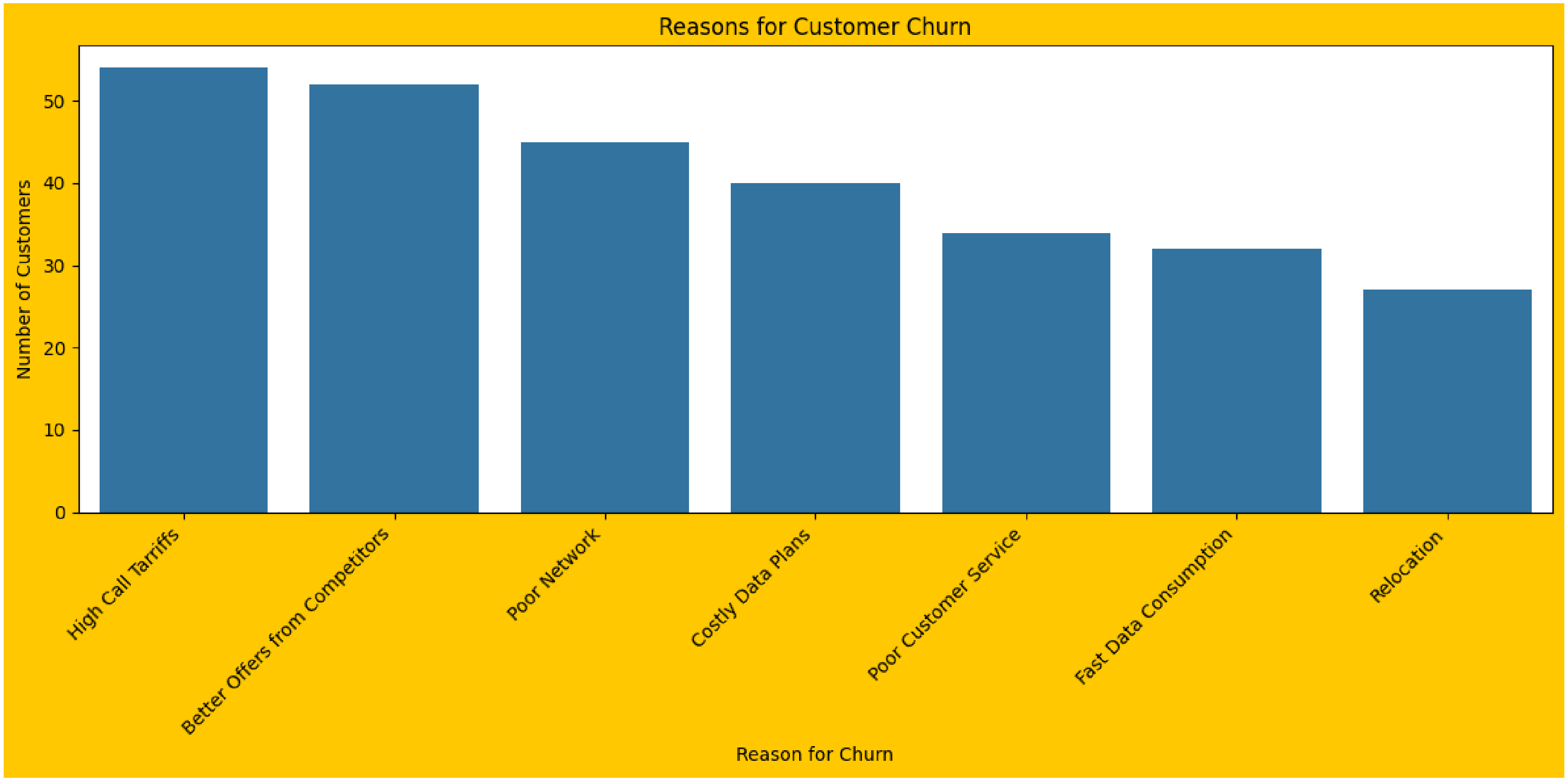
```
pd.crosstab(mtn["Customer Churn Status"], mtn["Reasons for Churn"])
```

Out[ ]:

Reasons for Churn	Better Offers from Competitors	Costly Data Plans	Fast Data Consumption	High Call Tarriffs	Poor Customer Service	Poor Network	Relocation
Customer Churn Status							
Yes	52	40	32	54	34	45	27

In [ ]:

```
churn_reasons_crosstab = pd.crosstab(mtn["Customer Churn Status"], mtn["Reasons for Churn"])
churn_reasons_yes = churn_reasons_crosstab.loc["Yes"].sort_values(ascending=False)
plt.figure(figsize=(12, 6), facecolor='#FFCC00')
sns.barplot(x=churn_reasons_yes.index, y=churn_reasons_yes.values)
plt.title("Reasons for Customer Churn")
plt.xlabel("Reason for Churn")
plt.ylabel("Number of Customers")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



```
In [ ]: #CHURN STATUS BASED ON DEVICE
```

```
In [ ]: pd.crosstab(mtn["Customer Churn Status"], mtn["Device Category"], margins=True)
```

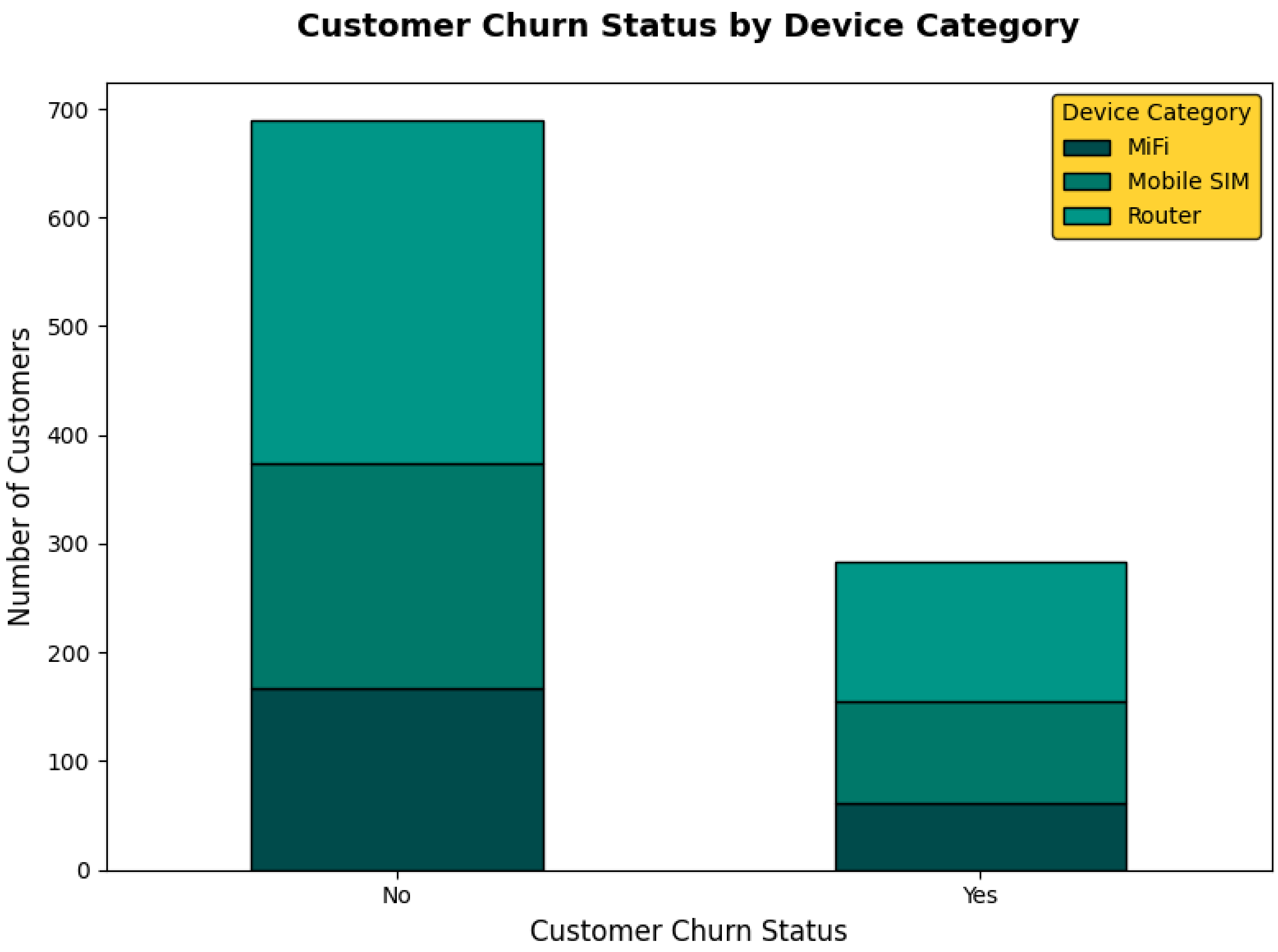
Out[ ]:

Device Category	MiFi	Mobile SIM	Router	All
Customer Churn Status				
No	167	207	316	690
Yes	61	94	129	284
All	228	301	445	974

In [ ]:

```
churn_device = pd.crosstab(mtn["Customer Churn Status"], mtn["Device Category"])

# Plot
plt.figure(figsize=(8, 6), facecolor="#FFCC00")  # MTN Yellow background

churn_device.plot(
    kind="bar",
    stacked=True,
    color=["#004C4C", "#00796B", "#009688"],  # MTN dark to medium teals
    edgecolor="black",
    figsize=(8, 6)
)

# Title & labels
plt.title("Customer Churn Status by Device Category", fontsize=14, fontweight="bold", color="black", pad=20)
plt.xlabel("Customer Churn Status", fontsize=12, color="black")
plt.ylabel("Number of Customers", fontsize=12, color="black")
plt.xticks(rotation=0, color="black")
plt.yticks(color="black")

# Legend
plt.legend(title="Device Category", facecolor="#FFCC00", edgecolor="black", labelcolor="black")

# Layout
plt.tight_layout()
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



```
In [ ]: mtn.to_csv("mtn_custmer_churn",index=False)
```