

```
In [26]: %matplotlib inline  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [27]: df = pd.read_csv("C:/Users/Kiran/OneDrive/Documents/Customer_Subscription_And_Tr
```

```
In [38]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 18106 entries, 0 to 18105  
Data columns (total 9 columns):  
 #   Column           Non-Null Count  Dtype     
 ---  --  
 0   cust_id          18106 non-null   int64    
 1   transaction_type 18106 non-null   object    
 2   transaction_date 18106 non-null   object    
 3   subscription_type 18106 non-null   object    
 4   subscription_price 18106 non-null   int64    
 5   customer_gender   18106 non-null   object    
 6   age_group         18106 non-null   object    
 7   customer_country  18106 non-null   object    
 8   referral_type    18106 non-null   object    
 dtypes: int64(2), object(7)  
 memory usage: 1.2+ MB
```

```
In [39]: df.describe()
```

```
Out[39]:
```

|              | cust_id      | subscription_price |
|--------------|--------------|--------------------|
| <b>count</b> | 18106.000000 | 18106.000000       |
| <b>mean</b>  | 5176.468298  | 73.340771          |
| <b>std</b>   | 2988.697910  | 27.902649          |
| <b>min</b>   | 1.000000     | 33.000000          |
| <b>25%</b>   | 2590.250000  | 53.000000          |
| <b>50%</b>   | 5157.500000  | 75.000000          |
| <b>75%</b>   | 7767.000000  | 99.000000          |
| <b>max</b>   | 10366.000000 | 119.000000         |

```
In [28]: df.head()
```

Out[28]:

|          | <b>cust_id</b> | <b>transaction_type</b> | <b>transaction_date</b> | <b>subscription_type</b> | <b>subscription_price</b> | <b>customer_name</b> |
|----------|----------------|-------------------------|-------------------------|--------------------------|---------------------------|----------------------|
| <b>0</b> | 1              | initial                 | 2020-09-01              | BASIC                    | 33                        | John Doe             |
| <b>1</b> | 2              | initial                 | 2022-03-01              | BASIC                    | 53                        | Jane Smith           |
| <b>2</b> | 3              | initial                 | 2020-10-01              | MAX                      | 99                        | Bob Johnson          |
| <b>3</b> | 3              | REDUCTION               | 2022-02-01              | BASIC                    | 53                        | Mike Williams        |
| <b>4</b> | 4              | initial                 | 2022-05-01              | PRO                      | 85                        | Sarah Davis          |

In [29]:

```
missing_data = df.isnull()
missing_data.head()
```

Out[29]:

|          | <b>cust_id</b> | <b>transaction_type</b> | <b>transaction_date</b> | <b>subscription_type</b> | <b>subscription_price</b> | <b>customer_name</b> |
|----------|----------------|-------------------------|-------------------------|--------------------------|---------------------------|----------------------|
| <b>0</b> | False          | False                   | False                   | False                    | False                     | False                |
| <b>1</b> | False          | False                   | False                   | False                    | False                     | False                |
| <b>2</b> | False          | False                   | False                   | False                    | False                     | False                |
| <b>3</b> | False          | False                   | False                   | False                    | False                     | False                |
| <b>4</b> | False          | False                   | False                   | False                    | False                     | False                |

In [30]:

```
for column in missing_data.columns.values.tolist():
    print(column)
    print(missing_data[column].value_counts())
    print("")
```

```
cust_id
cust_id
False    18106
Name: count, dtype: int64

transaction_type
transaction_type
False    18106
Name: count, dtype: int64

transaction_date
transaction_date
False    18106
Name: count, dtype: int64

subscription_type
subscription_type
False    18106
Name: count, dtype: int64

subscription_price
subscription_price
False    18106
Name: count, dtype: int64

customer_gender
customer_gender
False    18106
Name: count, dtype: int64

age_group
age_group
False    18106
Name: count, dtype: int64

customer_country
customer_country
False    18106
Name: count, dtype: int64

referral_type
referral_type
False    18106
Name: count, dtype: int64
```

In [32]: `df.notnull().sum()`

Out[32]:

|                    |       |
|--------------------|-------|
| cust_id            | 18106 |
| transaction_type   | 18106 |
| transaction_date   | 18106 |
| subscription_type  | 18106 |
| subscription_price | 18106 |
| customer_gender    | 18106 |
| age_group          | 18106 |
| customer_country   | 18106 |
| referral_type      | 18106 |
|                    |       |
| dtype:             | int64 |

In [33]: `df.shape`

Out[33]: (18106, 9)

In [36]: df.duplicated().sum()

Out[36]: np.int64(0)

In [43]: df['transaction\_date'] = pd.to\_datetime(df['transaction\_date'])

In [45]: df.drop\_duplicates(inplace = True)

In [46]: df['subscription\_price'] = pd.to\_numeric(df['subscription\_price'], errors='coerce')

In [ ]: df.tail()

|              | <b>cust_id</b> | <b>transaction_type</b> | <b>transaction_date</b> | <b>subscription_type</b> | <b>subscription_price</b> |
|--------------|----------------|-------------------------|-------------------------|--------------------------|---------------------------|
| <b>18101</b> | 10364          | UPGRADE                 | 2022-09-01              | MAX                      | 119                       |
| <b>18102</b> | 10365          | initial                 | 2021-05-01              | PRO                      | 75                        |
| <b>18103</b> | 10365          | REDUCTION               | 2021-07-01              | BASIC                    | 43                        |
| <b>18104</b> | 10366          | initial                 | 2021-01-01              | BASIC                    | 43                        |
| <b>18105</b> | 10366          | CHURN                   | 2022-02-01              | BASIC                    | 53                        |

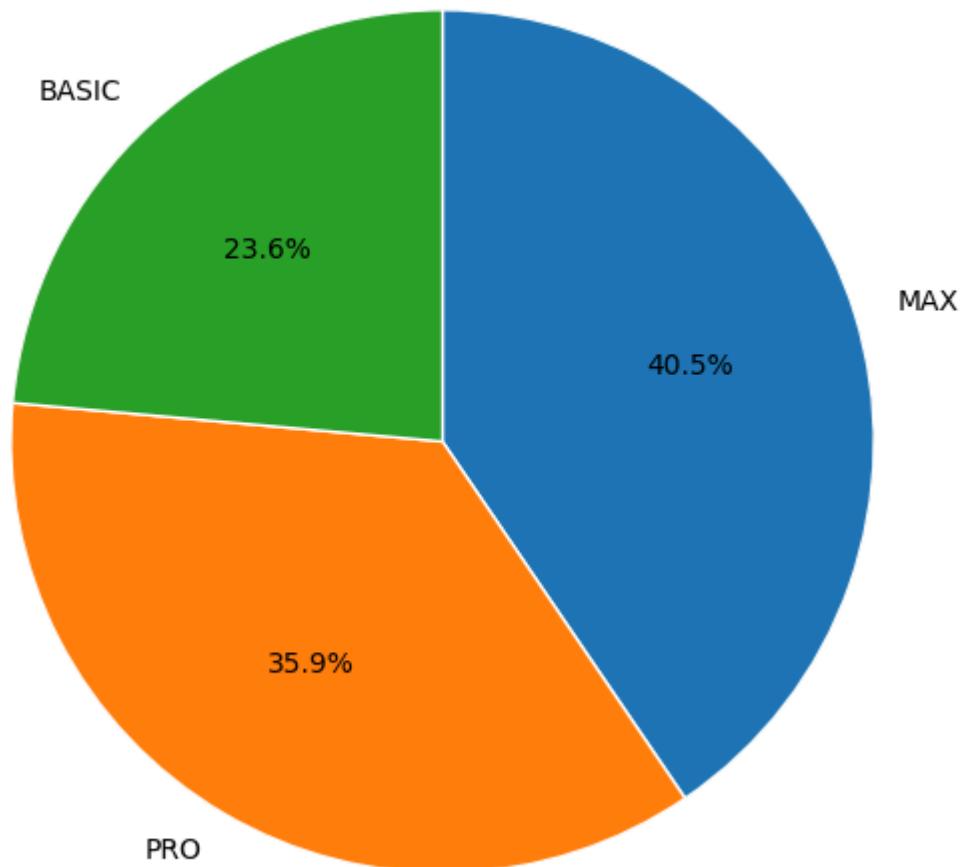


In [56]: revenue\_by\_plan = df.groupby('subscription\_type')['subscription\_price'].sum().reset\_index()
revenue\_by\_plan = revenue\_by\_plan.sort\_values(by='subscription\_price', ascending=False)

import seaborn as sns
import matplotlib.pyplot as plt

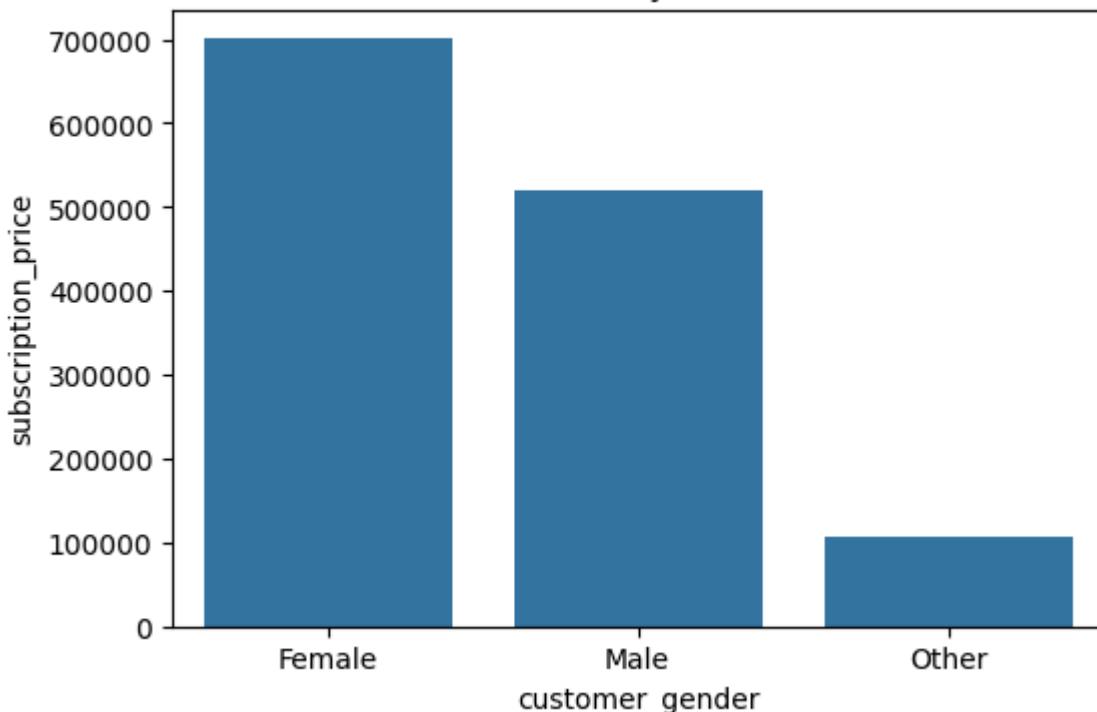
plt.figure(figsize=(7,7))
plt.pie(
 revenue\_by\_plan['subscription\_price'],
 labels=revenue\_by\_plan['subscription\_type'],
 autopct='%1.1f%%', # show percentage values
 startangle=90, # rotate start angle for better aesthetics
 counterclock=False,
 wedgeprops={'edgecolor': 'white'}
)
plt.title("Revenue Contribution by Subscription Type", fontsize=14)
plt.show()

## Revenue Contribution by Subscription Type



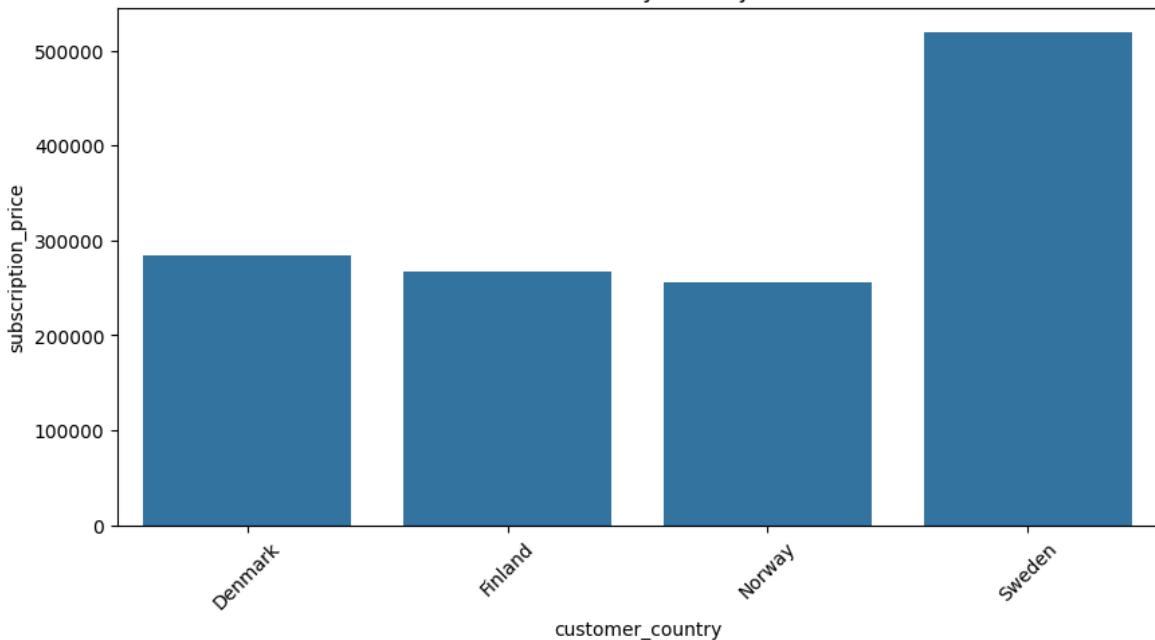
```
In [50]: gender_revenue = df.groupby('customer_gender')['subscription_price'].sum().reset_index()
plt.figure(figsize=(6,4))
sns.barplot(data=gender_revenue, x='customer_gender', y='subscription_price')
plt.title("Revenue by Gender")
plt.show()
```

### Revenue by Gender



```
In [51]: country_revenue = df.groupby('customer_country')['subscription_price'].sum().res
plt.figure(figsize=(10,5))
sns.barplot(data=country_revenue, x='customer_country', y='subscription_price')
plt.title("Revenue by Country")
plt.xticks(rotation=45)
plt.show()
```

### Revenue by Country



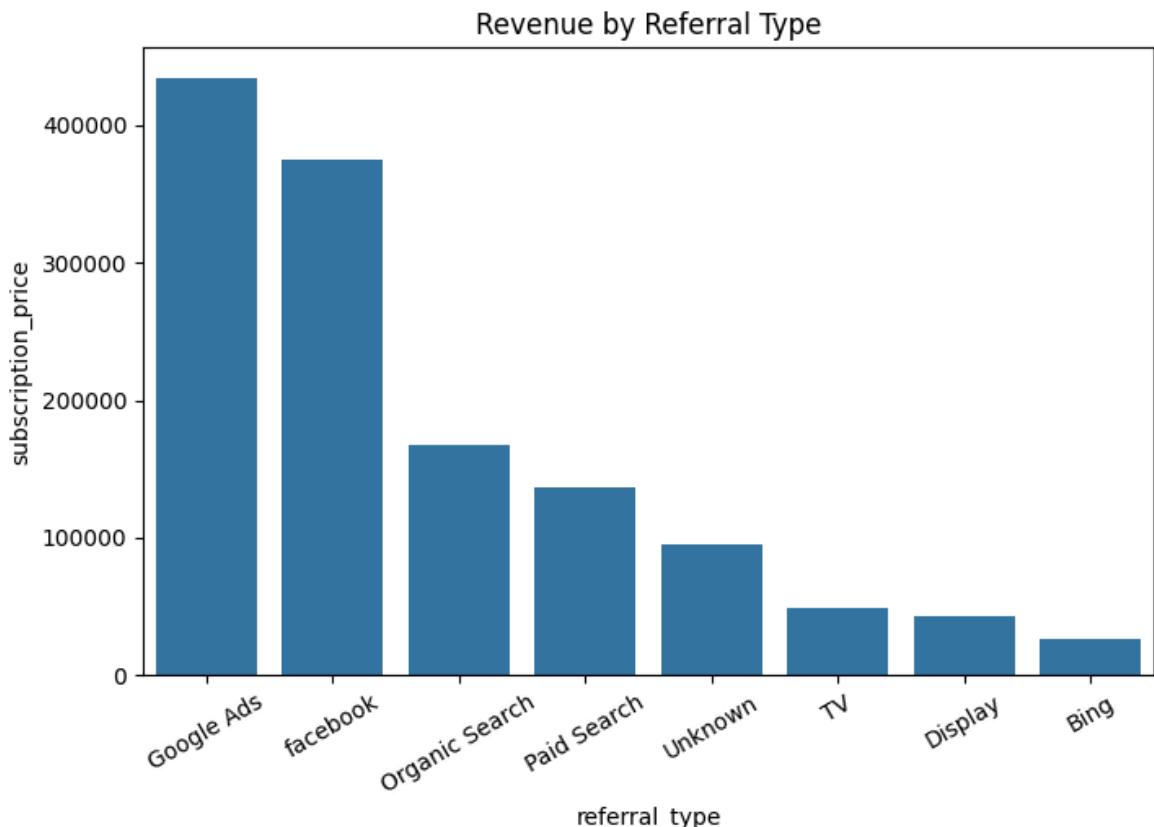
```
In [57]: df['month'] = df['transaction_date'].dt.to_period('M').dt.to_timestamp()
monthly_trend = df.groupby('month', as_index=False)[ 'subscription_price'].sum()
monthly_trend = monthly_trend.sort_values('month')
plt.figure(figsize=(10,5))
sns.lineplot(data=monthly_trend, x='month', y='subscription_price', marker='o')
plt.title("Monthly Subscription Revenue Trend")
plt.xlabel("Month")
```

```
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.show()
```



```
In [54]: referral_contribution = df.groupby('referral_type')[['subscription_price']].sum()
referral_contribution = referral_contribution.sort_values(by='subscription_price', ascending=False)

plt.figure(figsize=(8,5))
sns.barplot(data=referral_contribution, x='referral_type', y='subscription_price')
plt.title("Revenue by Referral Type")
plt.xticks(rotation=30)
plt.show()
```

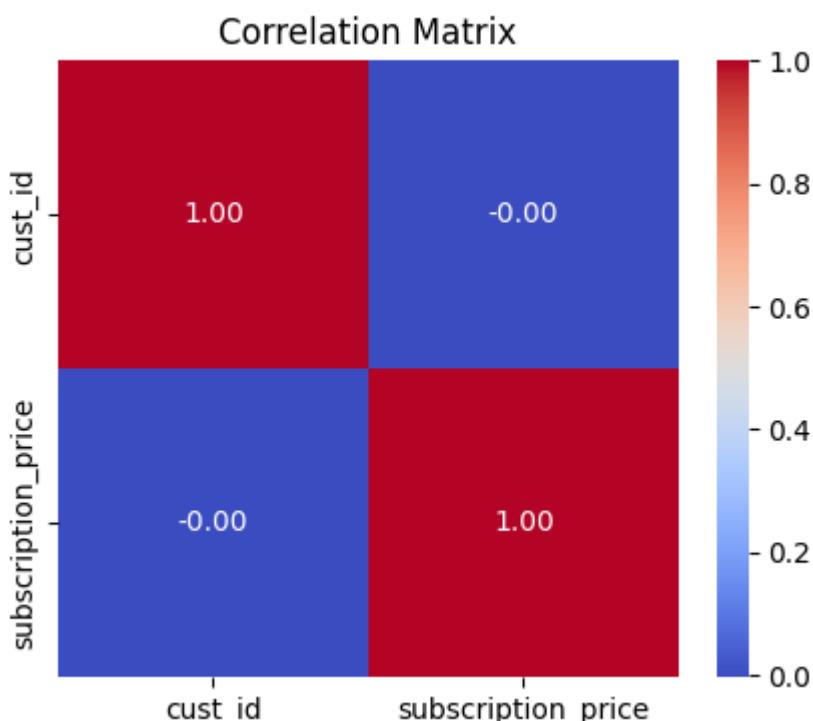


```
In [61]: corr = df.corr(numeric_only = True)
corr
```

```
Out[61]:
```

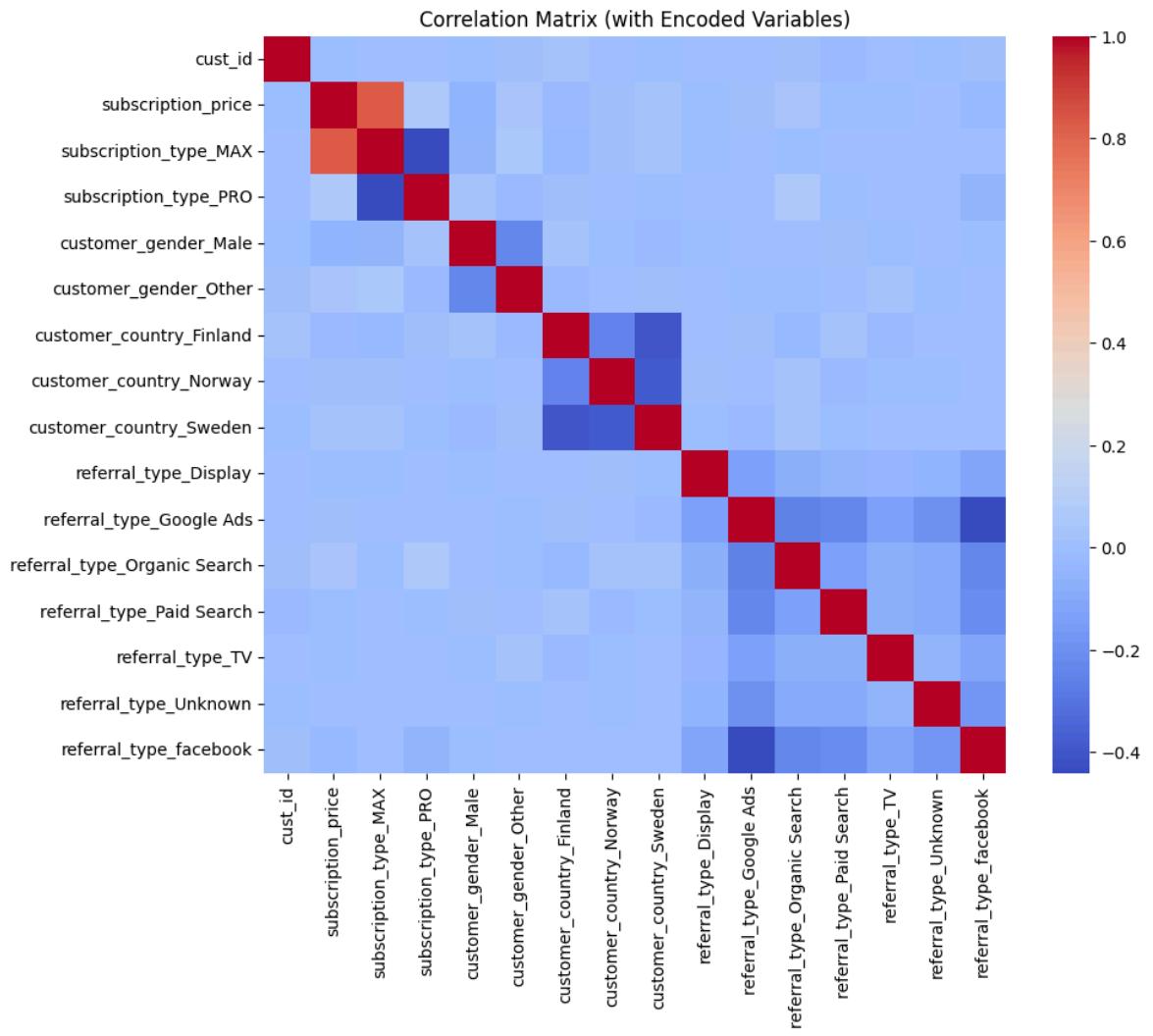
|                           | <b>cust_id</b> | <b>subscription_price</b> |
|---------------------------|----------------|---------------------------|
| <b>cust_id</b>            | 1.000000       | -0.003486                 |
| <b>subscription_price</b> | -0.003486      | 1.000000                  |

```
In [62]: plt.figure(figsize=(5,4))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```



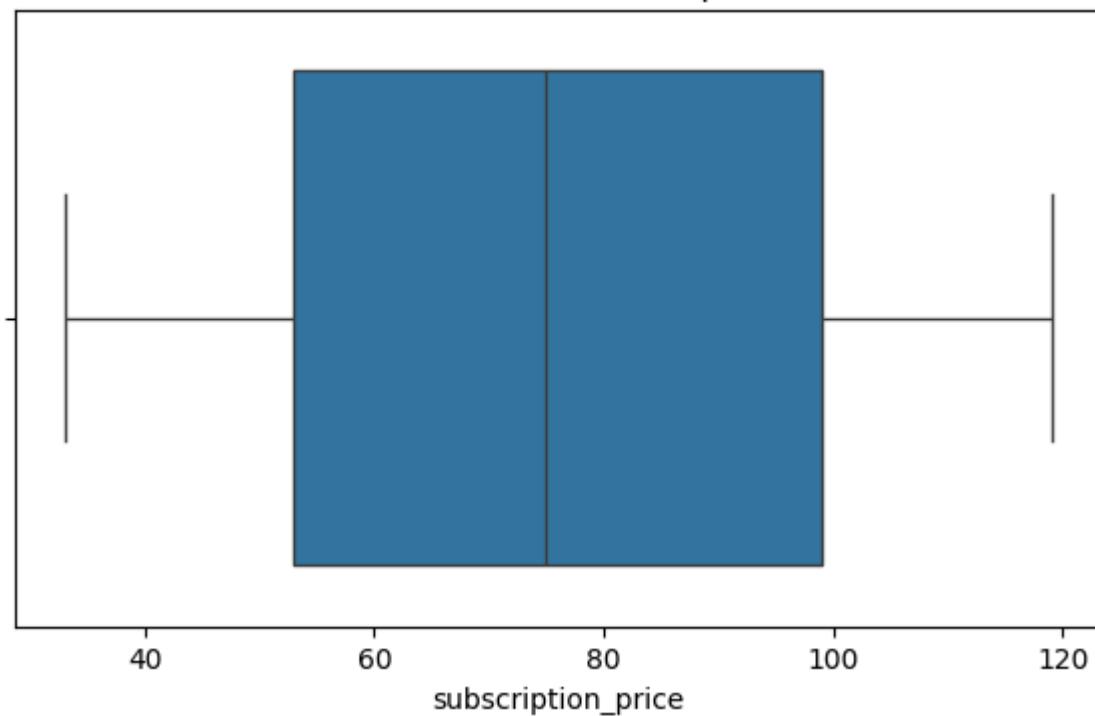
```
In [63]: encoded_df = df.copy()
encoded_df = pd.get_dummies(encoded_df, columns=['subscription_type', 'customer_'
                                                 _id'])

corr = encoded_df.corr(numeric_only=True)
plt.figure(figsize=(10,8))
sns.heatmap(corr, cmap="coolwarm", cbar=True)
plt.title("Correlation Matrix (with Encoded Variables)")
plt.show()
```



```
In [64]: plt.figure(figsize=(7,4))
sns.boxplot(x=df['subscription_price'])
plt.title("Outlier Detection – Subscription Price")
plt.show()
```

## Outlier Detection — Subscription Price



```
In [65]: Q1 = df['subscription_price'].quantile(0.25)
Q3 = df['subscription_price'].quantile(0.75)
IQR = Q3 - Q1

lower_limit = Q1 - 1.5 * IQR
upper_limit = Q3 + 1.5 * IQR

outliers = df[(df['subscription_price'] < lower_limit) | (df['subscription_price'] > upper_limit)]
print("Outliers detected:", outliers.shape[0])
outliers.head()
```

Outliers detected: 0

```
Out[65]: cust_id  transaction_type  transaction_date  subscription_type  subscription_price  cust...
```

```
In [70]: fig, axes = plt.subplots(2, 3, figsize=(18, 10))
fig.suptitle("📊 Customer Subscription Insights Dashboard", fontsize=18, fontweight='bold')

# Chart 1: Subscription Type Distribution
sns.countplot(data=df, x='subscription_type', ax=axes[0,0])
axes[0,0].set_title("Subscription Type Distribution")
axes[0,0].set_xlabel("")
axes[0,0].set_ylabel("Count")

# Chart 2: Total Revenue by Subscription Type
revenue = df.groupby('subscription_type')['subscription_price'].sum().reset_index()
sns.barplot(data=revenue, x='subscription_type', y='subscription_price', ax=axes[0,1])
axes[0,1].set_title("Revenue by Subscription Type")
axes[0,1].set_xlabel("")
axes[0,1].set_ylabel("Total Revenue ($)")

# Chart 3: Gender-wise Revenue
gender_rev = df.groupby('customer_gender')['subscription_price'].sum().reset_index()
```

```

sns.barplot(data=gender_rev, x='customer_gender', y='subscription_price', ax=axes[0,2])
axes[0,2].set_title("Revenue by Gender")
axes[0,2].set_xlabel("")
axes[0,2].set_ylabel("Total Revenue ($)")

# Chart 4: Monthly Revenue Trend
monthly_rev = df.groupby('month')['subscription_price'].sum().reset_index()
sns.lineplot(data=monthly_rev, x='month', y='subscription_price', marker='o', ax=axes[1,0])
axes[1,0].set_title("Monthly Revenue Trend")
axes[1,0].set_xlabel("")
axes[1,0].set_ylabel("Revenue ($)")
axes[1,0].tick_params(axis='x', rotation=45)

# Chart 5: Revenue by Referral Type
ref_rev = df.groupby('referral_type')['subscription_price'].sum().reset_index()
sns.barplot(data=ref_rev, x='referral_type', y='subscription_price', ax=axes[1,1])
axes[1,1].set_title("Revenue by Referral Type")
axes[1,1].set_xlabel("")
axes[1,1].set_ylabel("Revenue ($)")
axes[1,1].tick_params(axis='x', rotation=30)

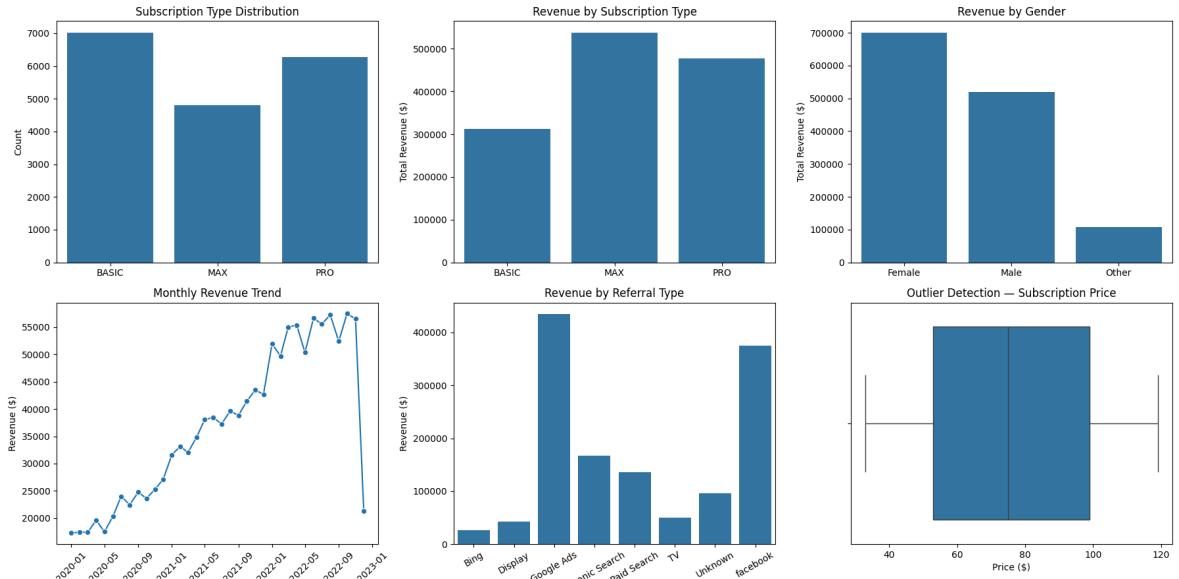
# Chart 6: Outlier Detection – Subscription Price
sns.boxplot(x=df['subscription_price'], ax=axes[1,2])
axes[1,2].set_title("Outlier Detection – Subscription Price")
axes[1,2].set_xlabel("Price ($)")
axes[1,2].set_ylabel("")

plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()

```

C:\Users\Kiran\AppData\Local\Temp\ipykernel\_14808\2708893583.py:47: UserWarning:  
 Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.  
 plt.tight\_layout(rect=[0, 0, 1, 0.96])  
C:\Users\Kiran\AppData\Roaming\Python\Python313\site-packages\IPython\core\pylabtools.py:170: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.  
fig.canvas.print\_figure(bytes\_io, \*\*kw)

Customer Subscription Insights Dashboard



## INSIGHTS

1. The company has a good balance between free and paid users — scope for upselling BASIC → PRO.
2. Higher-tier plans drive major revenue — suggest marketing focus on converting BASIC to MAX.
3. Targeting female users with retention campaigns may maximize ROI.
4. Consistent growth, but investigate reasons for the final-month drop (e.g., cancellations, promotions ending).
5. Digital marketing (Google, Facebook) is performing best — optimization or expansion here may improve ROI further.
6. No major data anomalies — pricing appears clean and consistent.

In [ ]: