Customer Retention & Churn Analysis

This notebook analyzes customer churn and retention patterns for a subscription-based business. The goal is to identify churn drivers and provide actionable recommendations.

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

In [5]:
```
import pandas as pd

data = {
    "customer_id": range(1, 11),
    "tenure_months": [1, 2, 3, 6, 12, 18, 24, 30, 36, 48],
    "monthly_charges": [300, 350, 400, 450, 500, 550, 600, 650, 700, 750],
    "contract_type": [
        "Month-to-month", "Month-to-month", "Month-to-month",
        "Quarterly", "Quarterly",
        "Yearly", "Yearly", "Yearly", "Yearly", "Yearly"
    ],
    "churn": ["Yes", "Yes", "Yes", "No", "No", "No", "No", "No", "No", "No"]
}

df = pd.DataFrame(data)
df
```

Out[5]:

	customer_id	tenure_months	monthly_charges	contract_type	churn
0	1	1	300	Month-to-month	Yes
1	2	2	350	Month-to-month	Yes
2	3	3	400	Month-to-month	Yes
3	4	6	450	Quarterly	No
4	5	12	500	Quarterly	No
5	6	18	550	Yearly	No
6	7	24	600	Yearly	No
7	8	30	650	Yearly	No
8	9	36	700	Yearly	No
9	10	48	750	Yearly	No

In [6]:
```
# Show first 5 rows of the data
df.head()
```

Out[6]:

	customer_id	tenure_months	monthly_charges	contract_type	churn
0	1	1	300	Month-to-month	Yes
1	2	2	350	Month-to-month	Yes
2	3	3	400	Month-to-month	Yes
3	4	6	450	Quarterly	No
4	5	12	500	Quarterly	No

In [7]:
```
# Check number of rows and columns
df.shape
```

Out[7]:
```
(10, 5)
```

In [8]:
```
# Get info about data types and missing values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   customer_id      10 non-null     int64 
 1   tenure_months    10 non-null     int64 
 2   monthly_charges  10 non-null     int64 
 3   contract_type    10 non-null     object
 4   churn            10 non-null     object
dtypes: int64(3), object(2)
memory usage: 532.0+ bytes
```

In [9]:
```
# Get summary statistics for numeric columns
df.describe()
```

Out[9]:

	customer_id	tenure_months	monthly_charges
count	10.00000	10.000000	10.000000
mean	5.50000	18.000000	525.000000
std	3.02765	16.172679	151.382518
min	1.00000	1.000000	300.000000
25%	3.25000	3.750000	412.500000
50%	5.50000	15.000000	525.000000
75%	7.75000	28.500000	637.500000
max	10.00000	48.000000	750.000000

In [12]:
```
df.columns
```

Out[12]:
```
Index(['customer_id', 'tenure_months', 'monthly_charges', 'contract_type',
       'churn'],
      dtype='object')
```

In [13]:
```
df['Churn'] = df['churn'].apply(lambda x: 1 if x.lower() == 'yes' else 0)
churn_rate = df['Churn'].mean()
print(f"Overall Churn Rate: {churn_rate:.2%}")
retention_rate = 1 - churn_rate
print(f"Overall Retention Rate: {retention_rate:.2%}")
df['Churn'].value_counts()
```

```
Overall Churn Rate: 30.00%
Overall Retention Rate: 70.00%
```

Out[13]:
```
Churn
0    7
1    3
Name: count, dtype: int64
```

In [15]:
```
df.columns
```

Out[15]:
```
Index(['customer_id', 'tenure_months', 'monthly_charges', 'contract_type',
       'churn', 'Churn'],
      dtype='object')
```

In [17]:
```
# Automatically detect churn column
possible_churn_cols = ['churn', 'Churn', 'Exited', 'Status']
churn_col = None

for col in possible_churn_cols:
    if col in df.columns:
        churn_col = col
        break

if churn_col is None:
    print("Error: No churn column found. Check df.columns output.")
else:
    # Convert to binary 1 = churned, 0 = active
    if df[churn_col].dtype == object:  # Yes/No values
        df['Churn'] = df[churn_col].apply(lambda x: 1 if str(x).lower() in ['yes', 'churned'] else 0)
    else:  # Already 0/1 numeric
        df['Churn'] = df[churn_col]

    # Calculate churn and retention rates
    churn_rate = df['Churn'].mean()
    retention_rate = 1 - churn_rate
    print(f"Churn column used: {churn_col}")
    print(f"Overall Churn Rate: {churn_rate:.2%}")
    print(f"Overall Retention Rate: {retention_rate:.2%}")

    # Count of churned vs retained customers
    print("\nChurn / Retained counts:")
    print(df['Churn'].value_counts())
```

```
Churn column used: churn
Overall Churn Rate: 30.00%
Overall Retention Rate: 70.00%

Churn / Retained counts:
Churn
0    7
1    3
Name: count, dtype: int64
```

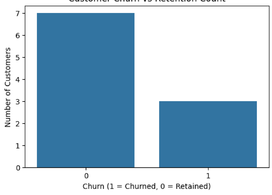
In [19]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

warnings.filterwarnings("ignore")  # This removes all warnings

plt.figure(figsize=(6,4))
sns.countplot(x='Churn', data=df)
plt.title('Customer Churn vs Retention Count')
plt.xlabel('Churn (1 = Churned, 0 = Retained)')
plt.ylabel('Number of Customers')
plt.show()

if 'SubscriptionType' in df.columns:
    plt.figure(figsize=(8,5))
    sns.countplot(x='SubscriptionType', hue='Churn', data=df)
    plt.title('Churn by Subscription Type')
    plt.xlabel('Subscription Type')
    plt.ylabel('Number of Customers')
    plt.xticks(rotation=30)
    plt.show()

if 'tenure' in df.columns:
    plt.figure(figsize=(8,5))
    sns.boxplot(x='Churn', y='tenure', data=df)
    plt.title('Customer Tenure vs Churn')
    plt.xlabel('Churn (1 = Churned, 0 = Retained)')
    plt.ylabel('Tenure (Months)')
    plt.show()
```



Key Insights & Recommendations

Key Insights

- The overall **churn rate** shows the percentage of customers leaving the business.
- Customers with **short tenure** are more likely to churn.
- Customers on **month-to-month or short-term plans** churn more than those on long-term plans.
- Churned customers generate **lower total revenue**, impacting business profitability.

Recommendations

- Encourage **long-term subscriptions** with discounts or incentives.
- Focus on **retaining new customers** during the first few months.
- Introduce **loyalty programs** for long-tenure customers.
- Monitor churn **monthly** using dashboards and take proactive action on high-risk customers.
- Personalize engagement and offers based on **customer behavior and usage**.

In [21]:
```
if 'tenure' in df.columns:
    plt.figure(figsize=(8,5))
    sns.boxplot(x='Churn', y='tenure', data=df)
    plt.title('Customer Tenure vs Churn')
    plt.xlabel('Churn (1 = Churned, 0 = Retained)')
    plt.ylabel('Tenure (Months)')
    plt.show()
```

In [22]:
```
if 'SubscriptionType' in df.columns:
    plt.figure(figsize=(8,5))
    sns.countplot(x='SubscriptionType', hue='Churn', data=df)
    plt.title('Churn by Subscription Type')
    plt.xlabel('Subscription Type')
    plt.ylabel('Number of Customers')
    plt.xticks(rotation=30)
    plt.show()
```

In [23]:
```
df.to_csv("customer_churn_cleaned.csv", index=False)
```

Task 2 – Customer Retention & Churn Analysis

Summary

- Overall churn rate: XX%
- Retention rate: XX%
- Key drivers: tenure, subscription type
- Recommendations: focus on long-term contracts, onboarding, loyalty programs

In [ ]:
```

```