Importing Pandas

```
import pandas as pd
```

Importing file and some basic checks

```
df = pd.read_csv("clean_telecom.csv")

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 496 entries, 0 to 495
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   customer_id         496 non-null    object 
 1   gender              496 non-null    object 
 2   senior_citizens     496 non-null    object 
 3   partner             496 non-null    object 
 4   dependents          496 non-null    object 
 5   contract_type       496 non-null    object 
 6   monthly_charges     496 non-null    float64
 7   total_charges       496 non-null    float64
 8   active_months       496 non-null    int64  
 9   total_gb_used       496 non-null    float64
 10  total_minutes_used  496 non-null    int64  
 11  churn_flag          496 non-null    bool   
dtypes: bool(1), float64(3), int64(2), object(6)
memory usage: 43.2+ KB
```

```
df.describe()
```

```
       monthly_charges  total_charges  active_months  total_gb_used  \
count       496.000000     496.000000     496.000000     496.000000   
mean         75.437258    2627.699435       6.641129    1968.334415   
std          25.686423    1787.667418       3.468217    1251.196141   
min          30.140000      42.600000       1.000000     122.670000   
25%          54.887500    1152.132500       4.000000     923.782500   
50%          75.805000    2390.950000       7.000000    1843.125000   
75%          97.672500    3759.655000      10.000000    2833.912500   
max         119.590000    8188.600000      12.000000    4996.500000   

       total_minutes_used  
count          496.000000  
mean         51460.360887  
std          32409.308363  
min            345.000000  
```

```
25%          24439.500000
50%          43860.000000
75%          76712.250000
max         127305.000000
```

```
df.head()
```

```
  customer_id  gender senior_citizens partner dependents contract_type
\
0    CUST0001    Male              No      No        Yes      Two year

1    CUST0002  Female              No     Yes         No      Two year

2    CUST0003    Male              No     Yes         No      One year

3    CUST0004    Male             Yes     Yes        Yes      Two year

4    CUST0005    Male              No     Yes        Yes      One year


   monthly_charges  total_charges  active_months  total_gb_used  \
0           118.32        5561.04              4         332.31
1            53.09         637.08             10        2514.78
2            88.88        5510.56             12        2311.71
3            47.83         382.64              4        1576.35
4            80.88        1698.48              4        1569.57

   total_minutes_used  churn_flag
0               19026        True
1               68958       False
2               81813       False
3               34950       False
4               46464       False
```

```
df.describe(include="all")
```

```
       customer_id  gender senior_citizens partner dependents
contract_type  \
count          496     496             496     496        496
496
unique         496       2               2       2          2
3
top       CUST0001  Female             Yes     Yes        Yes      One
year
freq             1     253             252     262        261
178
mean           NaN     NaN             NaN     NaN        NaN
NaN
std            NaN     NaN             NaN     NaN        NaN
NaN
min            NaN     NaN             NaN     NaN        NaN
```

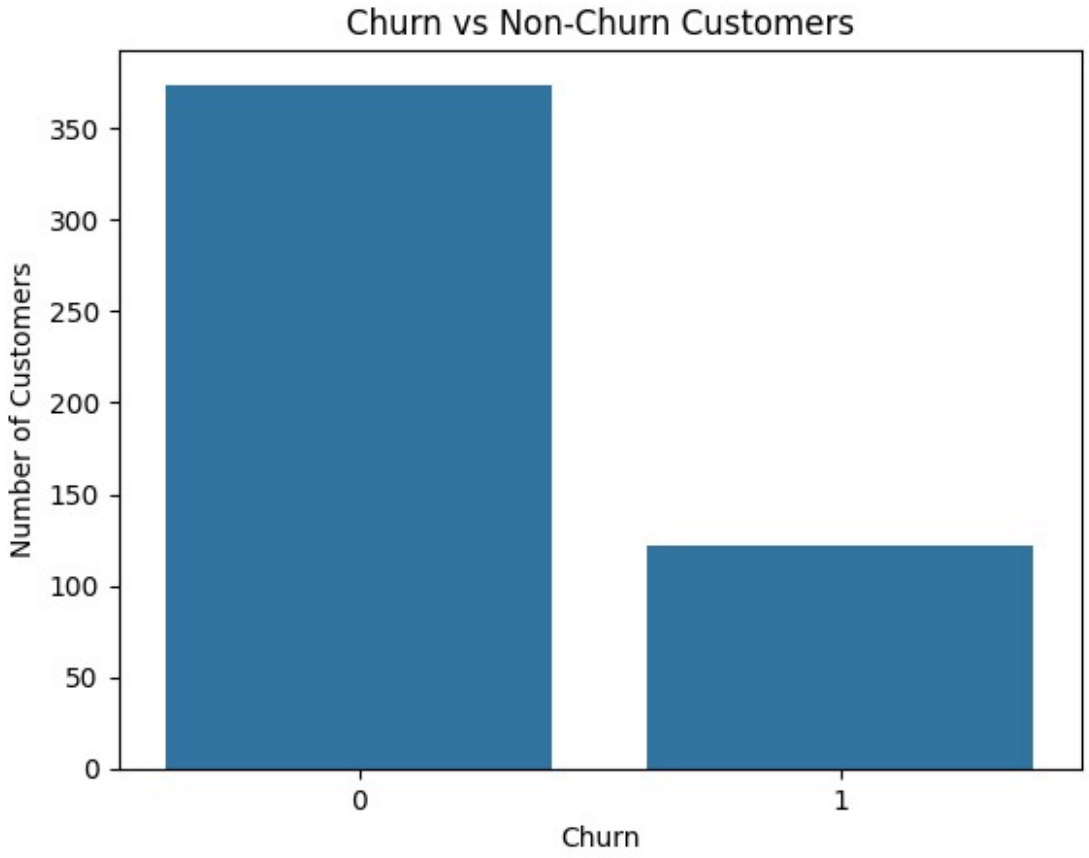
```
NaN
25%             NaN      NaN              NaN      NaN         NaN
NaN
50%             NaN      NaN              NaN      NaN         NaN
NaN
75%             NaN      NaN              NaN      NaN         NaN
NaN
max             NaN      NaN              NaN      NaN         NaN
NaN

        monthly_charges  total_charges  active_months
total_gb_used  \
count        496.000000     496.000000     496.000000     496.000000

unique              NaN            NaN            NaN            NaN

top                 NaN            NaN            NaN            NaN

freq                NaN            NaN            NaN            NaN

mean          75.437258    2627.699435       6.641129    1968.334415

std           25.686423    1787.667418       3.468217    1251.196141

min           30.140000      42.600000       1.000000     122.670000

25%           54.887500    1152.132500       4.000000     923.782500

50%           75.805000    2390.950000       7.000000    1843.125000

75%           97.672500    3759.655000      10.000000    2833.912500

max          119.590000    8188.600000      12.000000    4996.500000


        total_minutes_used churn_flag
count           496.000000        496
unique                 NaN          2
top                    NaN      False
freq                   NaN        374
mean          51460.360887        NaN
std           32409.308363        NaN
min             345.000000        NaN
25%           24439.500000        NaN
50%           43860.000000        NaN
75%           76712.250000        NaN
max          127305.000000        NaN

df
```

```
    customer_id  gender senior_citizens partner dependents
contract_type  \
0      CUST0001    Male              No      No         Yes         Two
year
1      CUST0002  Female              No     Yes          No         Two
year
2      CUST0003    Male              No     Yes          No         One
year
3      CUST0004    Male             Yes     Yes         Yes         Two
year
4      CUST0005    Male              No     Yes         Yes         One
year
..          ...     ...             ...     ...         ...
...
491    CUST0496    Male              No     Yes         Yes  Month-to-
month
492    CUST0497    Male              No      No          No  Month-to-
month
493    CUST0498  Female             Yes      No         Yes  Month-to-
month
494    CUST0499    Male             Yes     Yes         Yes  Month-to-
month
495    CUST0500  Female              No     Yes         Yes         One
year

     monthly_charges  total_charges  active_months  total_gb_used  \
0             118.32        5561.04              4         332.31
1              53.09         637.08             10        2514.78
2              88.88        5510.56             12        2311.71
3              47.83         382.64              4        1576.35
4              80.88        1698.48              4        1569.57
..               ...            ...            ...            ...
491            56.95        1651.55              7        2775.12
492            62.28         311.40              8         605.04
493           102.40        4812.80              2         644.91
494            55.09        1873.06             11        2493.03
495            48.96        3182.40              4         730.65

     total_minutes_used  churn_flag
0                 19026        True
1                 68958       False
2                 81813       False
3                 34950       False
4                 46464       False
..                  ...         ...
491               71685       False
492               28785        True
493               17355       False
494               69987        True
495               28236       False
```

```
[496 rows x 12 columns]
```

Importing Matplotlib and Seaborn

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(data=df, x='churn_flag')
plt.title("Churn vs Non-Churn Customers")
plt.xlabel("Churn")
plt.ylabel("Number of Customers")
plt.show()
```

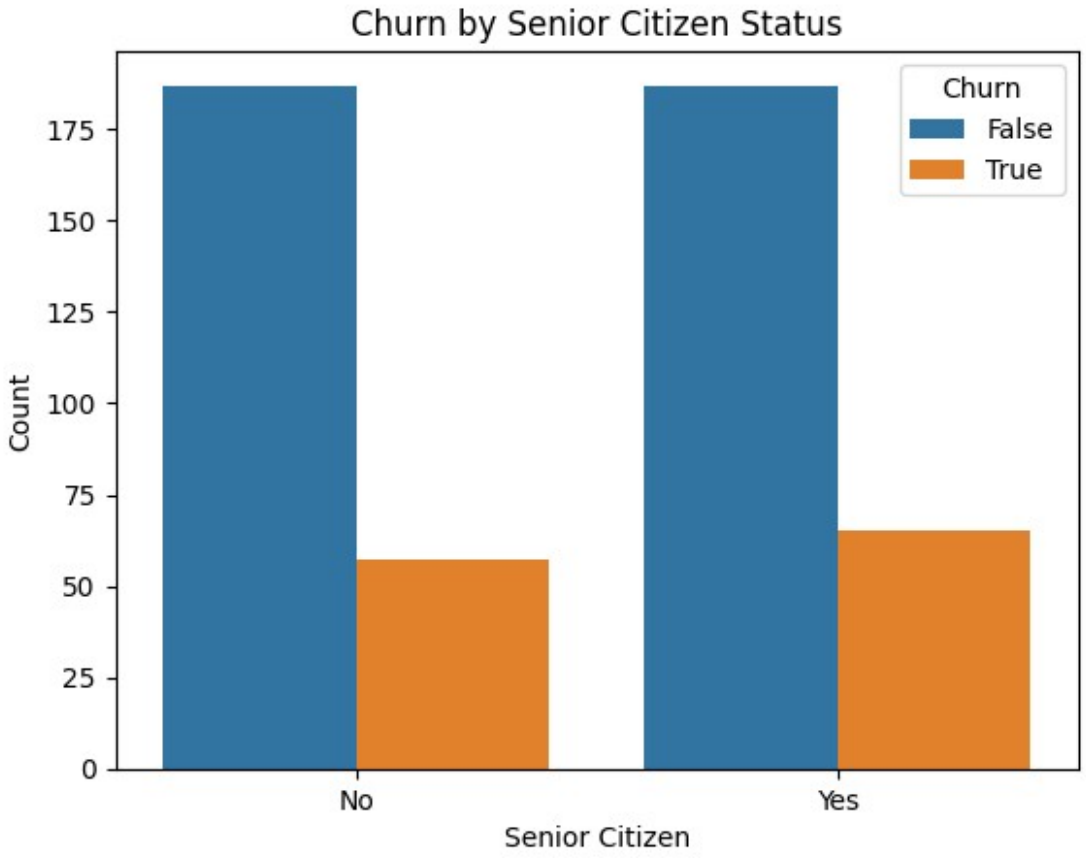


- significant difference between churn and no churn customers

```
churn_rate = df['churn_flag'].value_counts(normalize=True) * 100
churn_rate

churn_flag
False    75.403226
```

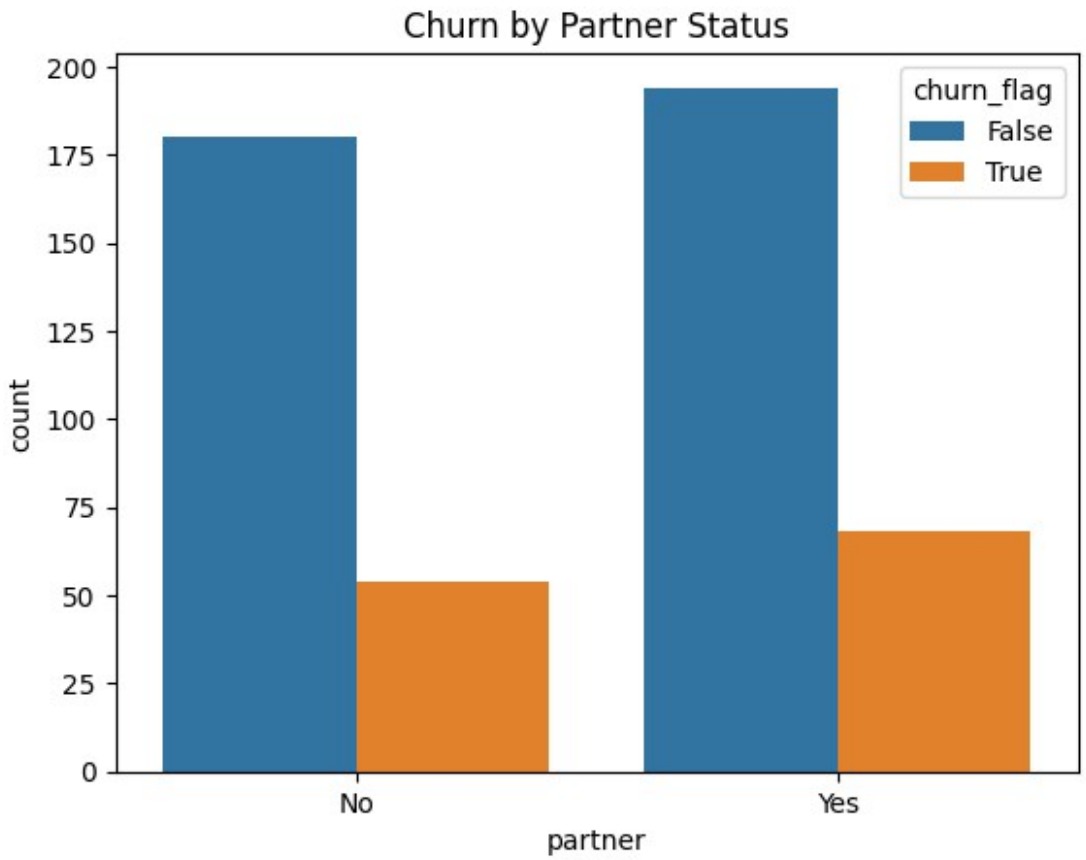
```
True     24.596774
Name: proportion, dtype: float64
```

```
sns.countplot(data=df, x='senior_citizens', hue='churn_flag')
plt.title("Churn by Senior Citizen Status")
plt.xlabel("Senior Citizen")
plt.ylabel("Count")
plt.legend(title='Churn')
plt.show()
```

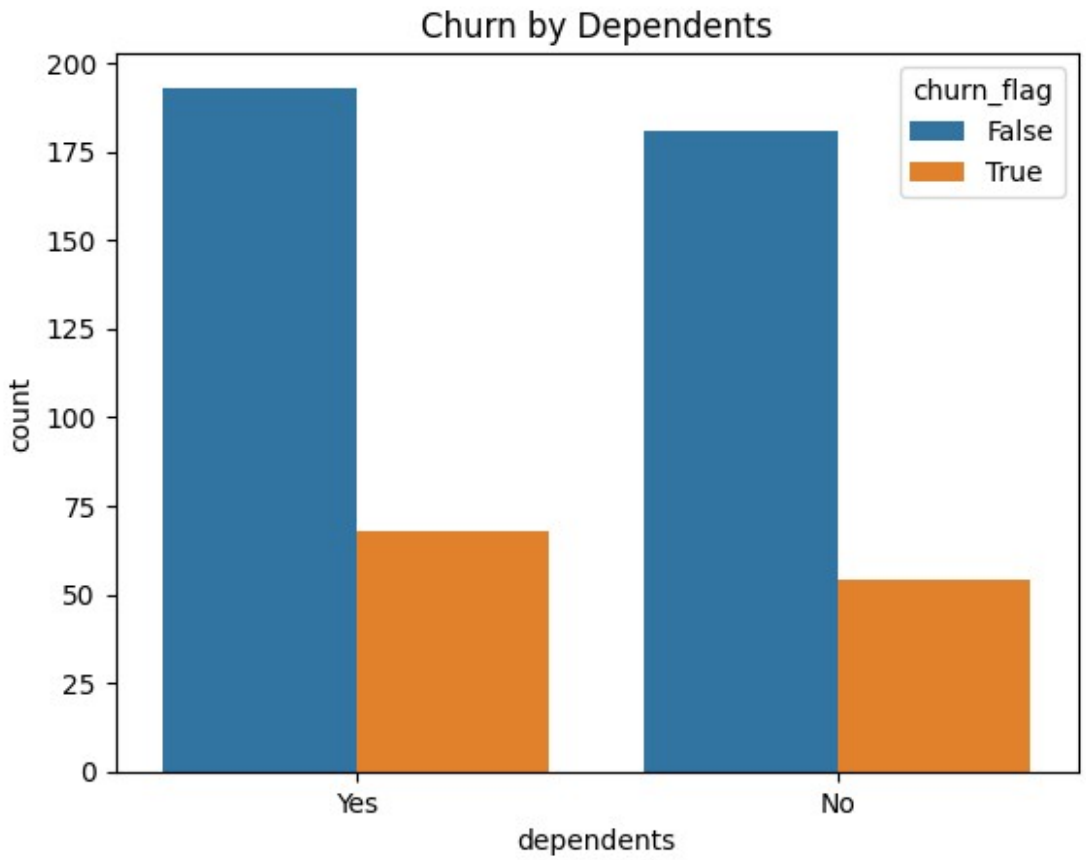


- balanced chart

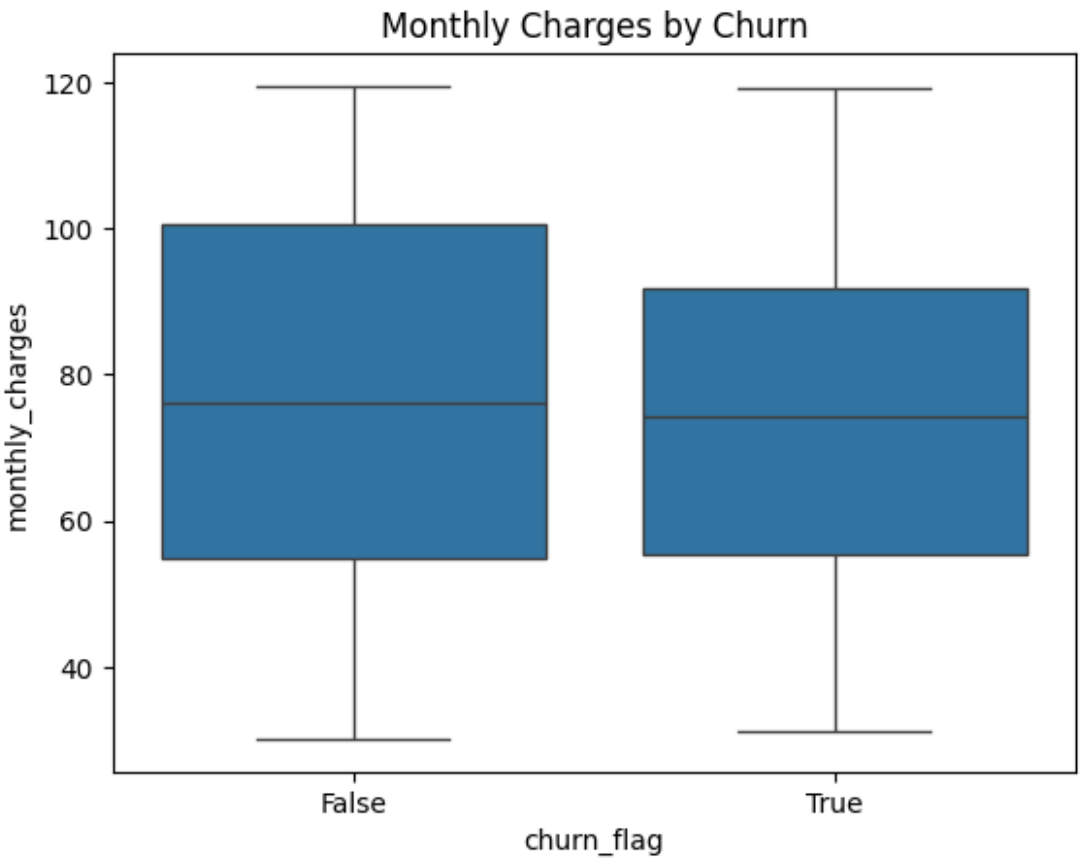
```
sns.countplot(data=df, x='partner', hue='churn_flag')
plt.title("Churn by Partner Status")
plt.show()
```



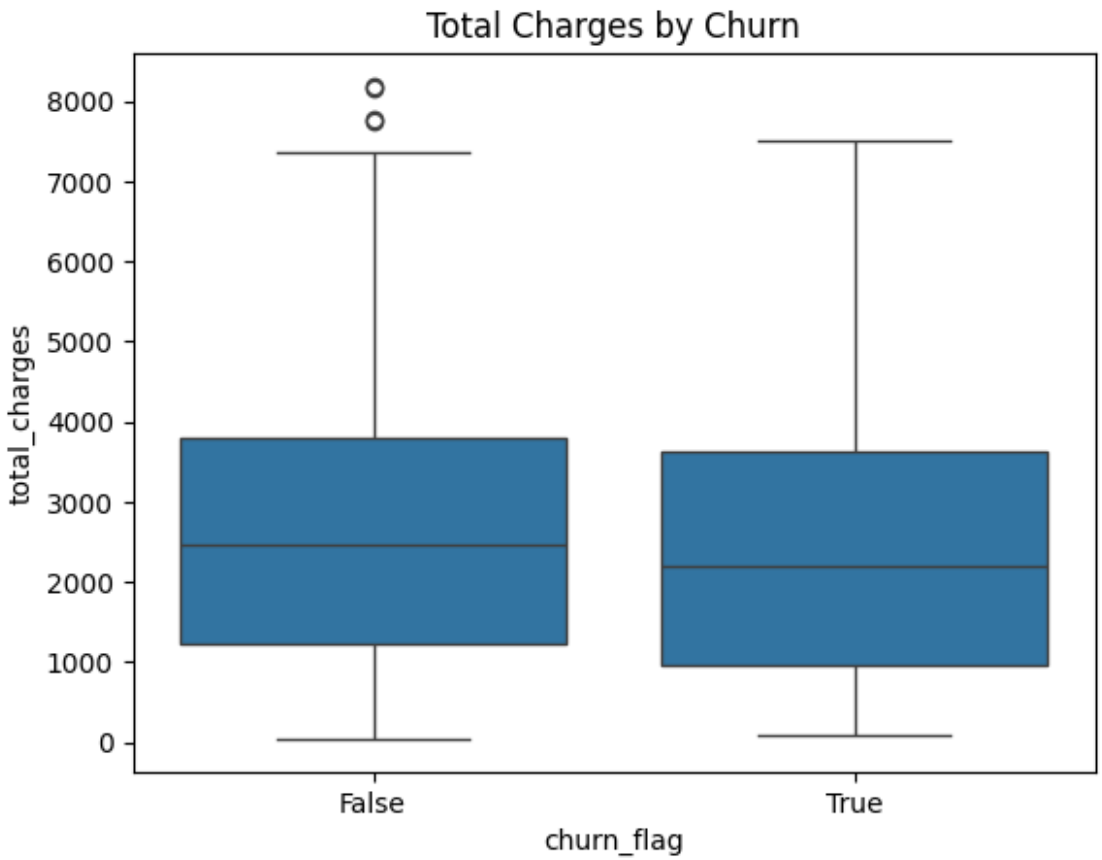

```
sns.countplot(data=df, x='dependents', hue='churn_flag')
plt.title("Churn by Dependents")
plt.show()
```



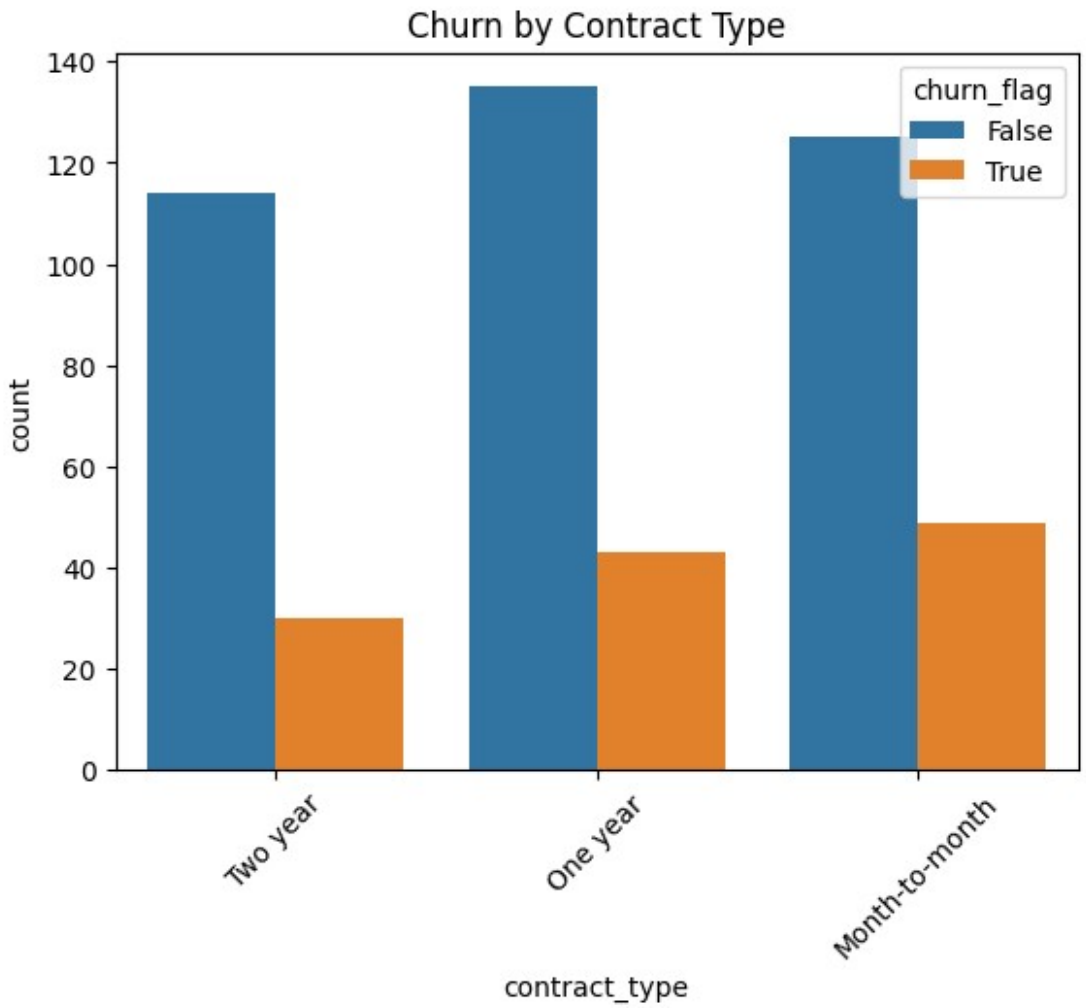

```
sns.boxplot(data=df, x='churn_flag', y='monthly_charges')
plt.title("Monthly Charges by Churn")
plt.show()
```



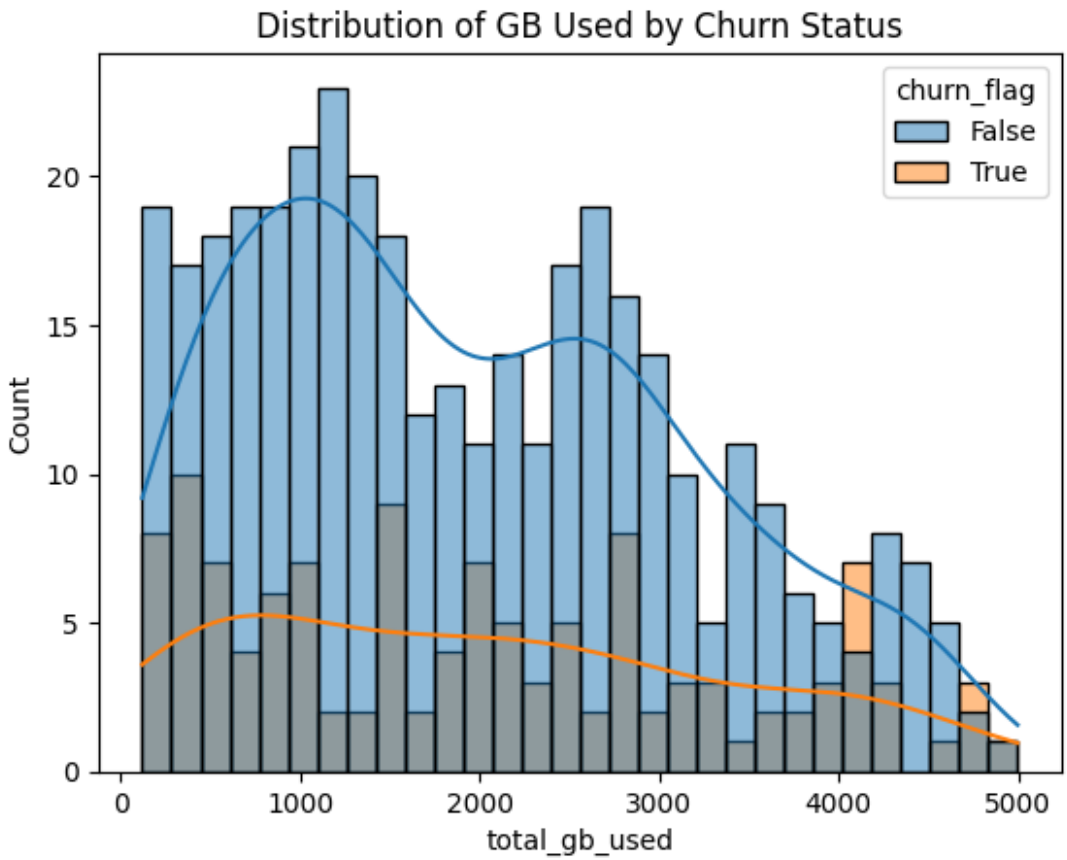

```
sns.boxplot(data=df, x='churn_flag', y='total_charges')
plt.title("Total Charges by Churn")
plt.show()
```



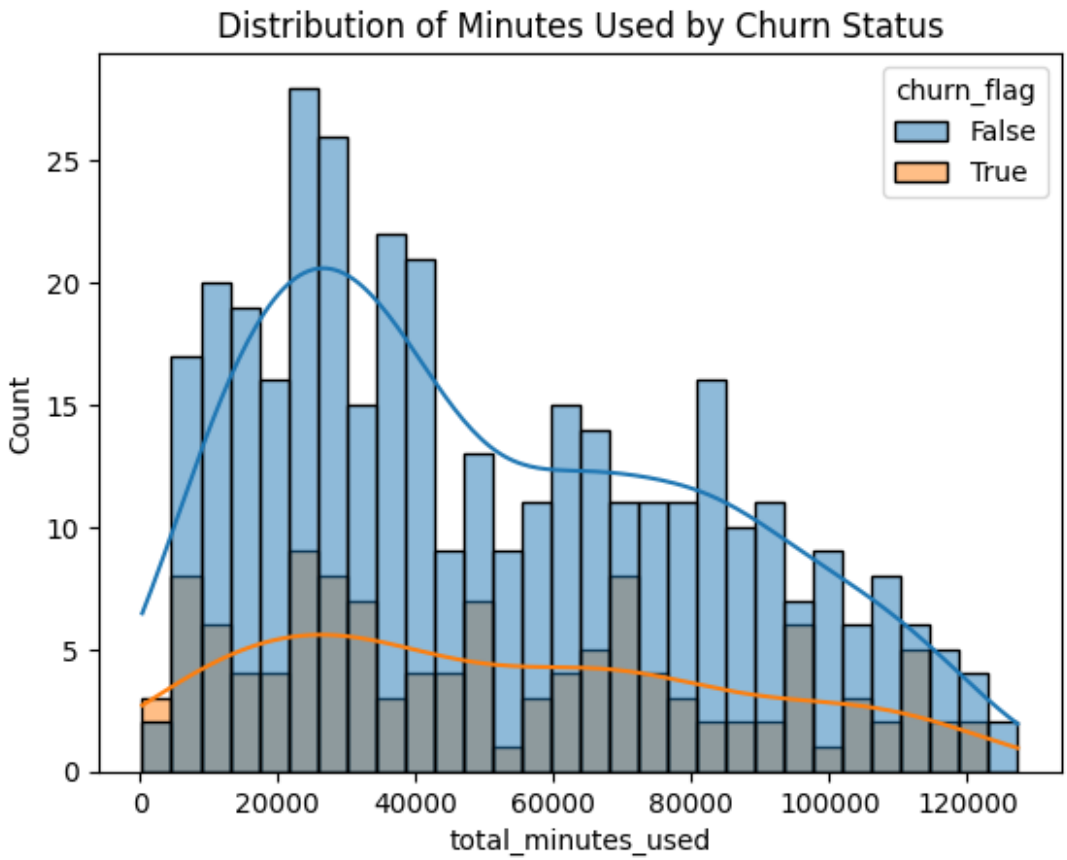

```
sns.countplot(data=df, x='contract_type', hue='churn_flag')
plt.title("Churn by Contract Type")
plt.xticks(rotation=45)
plt.show()
```



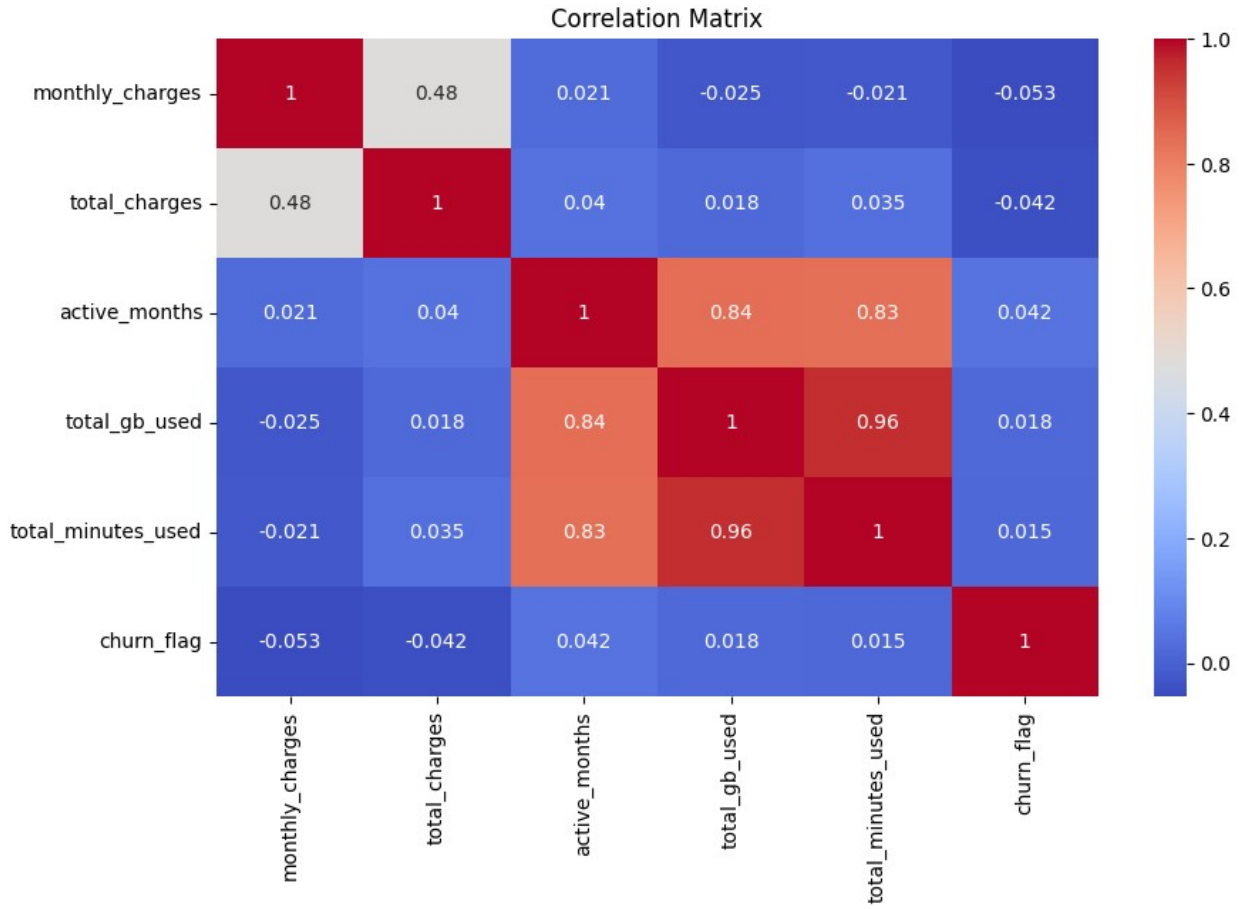

```
sns.histplot(data=df, x='total_gb_used', hue='churn_flag', kde=True,
bins=30)
plt.title("Distribution of GB Used by Churn Status")
plt.show()
```




```
sns.histplot(data=df, x='total_minutes_used', hue='churn_flag',
kde=True, bins=30)
plt.title("Distribution of Minutes Used by Churn Status")
plt.show()
```




```
# Encode churn_flag to 0/1 if it's still True/False
df['churn_flag'] = df['churn_flag'].map({True: 1, False: 0})

# Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```


Correlation Matrix
monthly_charges
total_charges
active_months
total_gb_used
total_minutes_used
churn_flag
1
0.48
0.021
-0.025
-0.021
-0.053
0.48
1
0.04
0.018
0.035
-0.042
0.021
0.04
1
0.84
0.83
0.042
-0.025
0.018
0.84
1
0.96
0.018
-0.021
0.035
0.83
0.96
1
0.015
-0.053
-0.042
0.042
0.018
0.015
1
1.0
0.8
0.6
0.4
0.2
0.0