```
import pandas as pd
```

Start coding or generate with AI.

```
df = pd.read_csv('/content/WA_Fn-UseC_-Telco-Customer-Churn.csv')
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DevicePro
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	

5 rows × 21 columns

```
df.info()
```

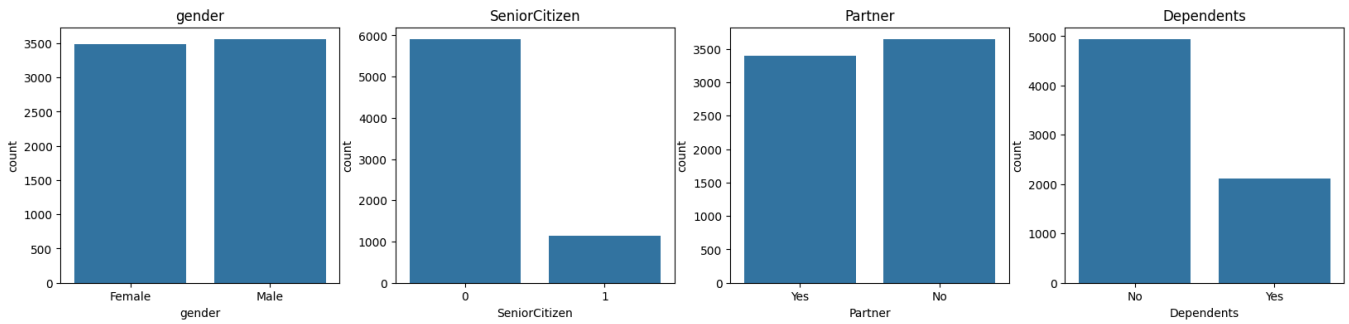
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

```
df['Churn'].value_counts()
```

Churn	count
No	5174
Yes	1869

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

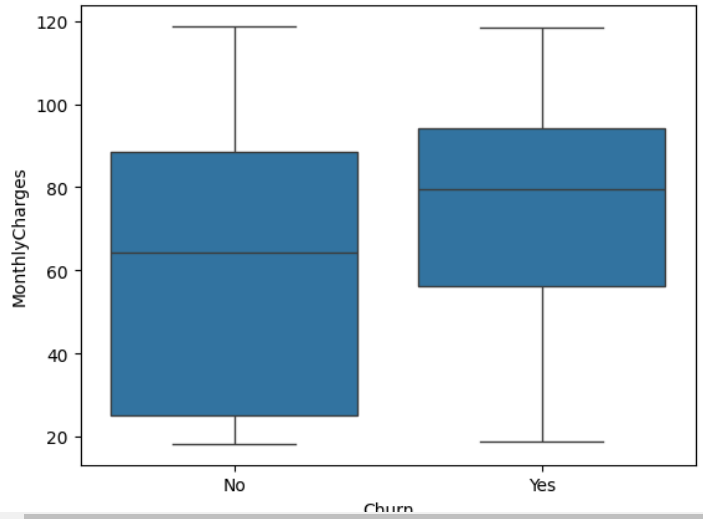
```
cols = ['gender','SeniorCitizen','Partner','Dependents']
numerical = cols
plt.figure(figsize=(20,4))
for i,col in enumerate(numerical):
  ax = plt.subplot(1,len(numerical),i+1)
  sns.countplot(x=str(col),data=df)
  ax.set_title(f'{col}')
```



- Most customers are younger individuals with no dependents
- Equal distribution of gender and marital status

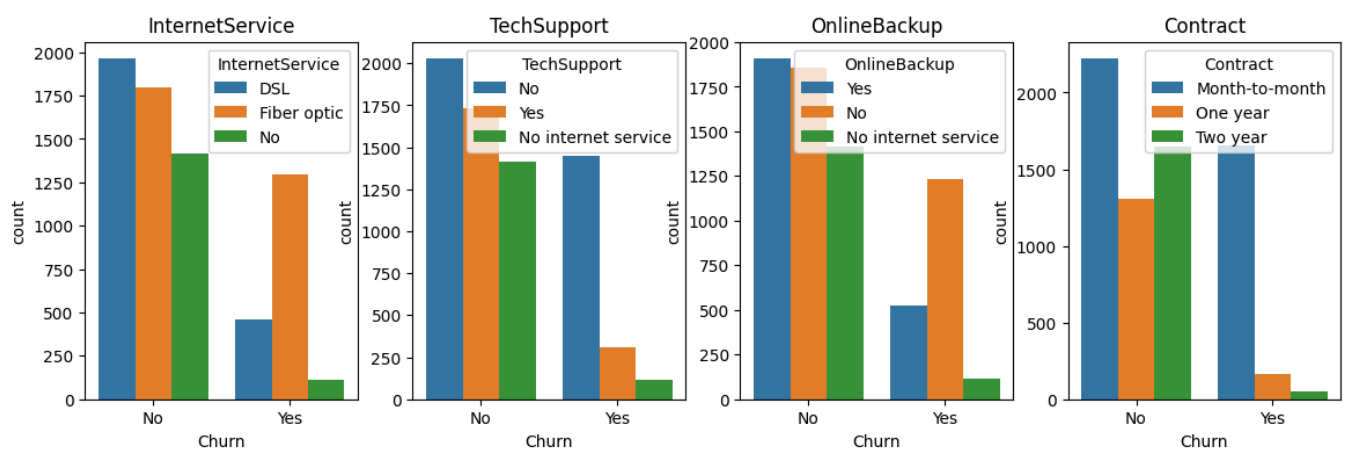
```
sns.boxplot(x='Churn',y='MonthlyCharges',data=df)
```

```
<Axes: xlabel='Churn', ylabel='MonthlyCharges'>
```



- Customers who churned have a higher median monthly charge

```
cols = ['InternetService','TechSupport','OnlineBackup','Contract']
plt.figure(figsize=(14,4))
for i,col in enumerate(cols):
  ax = plt.subplot(1,len(cols),i+1)
  sns.countplot(x='Churn',hue=str(col),data=df)
  ax.set_title(f'{col}')
```



- Internet Service: Customers using Fiber Optic cable churn more often than others
- Tech Support: Many users who churned did not sign up for any tech support
- Online Backup: Many customers who churned did not hsigned up for online backup
- Contract: Users who churned were almost always on a monthly contract.

```
df['TotalCharges'] = df['TotalCharges'].apply(lambda x:pd.to_numeric(x,errors='coerce')).dropna()

cat_features = df.drop(['customerID','TotalCharges','MonthlyCharges','SeniorCitizen','tenure'],axis=1)
cat_features.head()
```

	gender	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Stre
0	Female	Yes	No	No	No phone service	DSL	No	Yes	No	No	
1	Male	No	No	Yes	No	DSL	Yes	No	Yes	No	
2	Male	No	No	Yes	No	DSL	Yes	Yes	No	No	
3	Male	No	No	No	No phone service	DSL	Yes	No	Yes	Yes	
4	Female	No	No	Yes	No	Fiber optic	No	No	No	No	

```
from sklearn import preprocessing

le = preprocessing.LabelEncoder()
df_cat = cat_features.apply(le.fit_transform)
df_cat.head()
```

	gender	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Stre
0	0	1	0	0	1	0	0	2	0	0	
1	1	0	0	1	0	0	2	0	2	0	
2	1	0	0	1	0	0	2	2	0	0	
3	1	0	0	0	1	0	2	0	2	2	
4	0	0	0	1	0	1	0	0	0	0	

```
num_features = df[['customerID','TotalCharges','MonthlyCharges','SeniorCitizen','tenure']] # Changed 'customerId' to 'customerID
finaldf = pd.merge(num_features,df_cat,left_index=True,right_index=True)

from sklearn.model_selection import train_test_split

finaldf = finaldf.dropna()
finaldf = finaldf.drop(['customerID'],axis=1)

X = finaldf.drop(['Churn'],axis=1)
y = finaldf['Churn']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)

from imblearn.over_sampling import SMOTE

oversample = SMOTE(k_neighbors=5)
X_smote,y_smote = oversample.fit_resample(X_train,y_train)
X_train,y_train = X_smote,y_smote

y_train.value_counts()
```

	count
Churn	
1	3583
0	3583

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(random_state=46)
rf.fit(X_train,y_train)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=46)
```

```
from sklearn.metrics import accuracy_score


preds = rf.predict(X_test)
print(accuracy_score(preds,y_test))
```

```
0.7706161137440758
```