

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import warnings
warnings.filterwarnings("ignore")
```

```
In [7]: from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from sklearn.preprocessing import LabelEncoder
```

```
In [5]: pd.options.display.max_columns=None
```

```
In [13]: data=pd.read_csv(r"C:\Users\nalla\Desktop\projects\Telco-Customer-Churn.csv")
```

```
In [14]: data.head()
```

Out[14]:

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

```
In [16]: data.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
```

```
In [18]: data.drop('customerID',axis=1,inplace=True)
```

```
In [19]: set(''.join(data["TotalCharges"].tolist()))
```

```
Out[19]: {' ', '.', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'}
```

```
In [21]: data['TotalCharges']=data['TotalCharges'].replace(' ',np.nan)
```

```
In [22]: data['TotalCharges']=data['TotalCharges'].astype('float')
```

```
In [23]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   gender                 7043 non-null  object
1   SeniorCitizen          7043 non-null  int64
2   Partner                7043 non-null  object
3   Dependents             7043 non-null  object
4   tenure                 7043 non-null  int64
5   PhoneService           7043 non-null  object
6   MultipleLines          7043 non-null  object
7   InternetService        7043 non-null  object
8   OnlineSecurity         7043 non-null  object
9   OnlineBackup           7043 non-null  object
10  DeviceProtection       7043 non-null  object
11  TechSupport            7043 non-null  object
12  StreamingTV            7043 non-null  object
13  StreamingMovies        7043 non-null  object
14  Contract               7043 non-null  object
15  PaperlessBilling       7043 non-null  object
16  PaymentMethod          7043 non-null  object
17  MonthlyCharges         7043 non-null  float64
18  TotalCharges           7032 non-null  float64
19  Churn                  7043 non-null  object
dtypes: float64(2), int64(2), object(16)
memory usage: 1.1+ MB
```

```
In [24]: #EDA
data.columns=data.columns.str.lower()
```

```
In [26]: data.columns
```

```
Out[26]: Index(['gender', 'seniorcitizen', 'partner', 'dependents', 'tenure',
        'phoneservice', 'multiplelines', 'internetservice', 'onlinesecurity',
        'onlinebackup', 'deviceprotection', 'techsupport', 'streamingtv',
        'streamingmovies', 'contract', 'paperlessbilling', 'paymentmethod',
        'monthlycharges', 'totalcharges', 'churn'],
        dtype='object')
```

```
In [27]: num_cols=['tenure','monthlycharges','totalcharges']
```

```
In [28]: data[num_cols].describe().T
```

```
Out[28]:
```

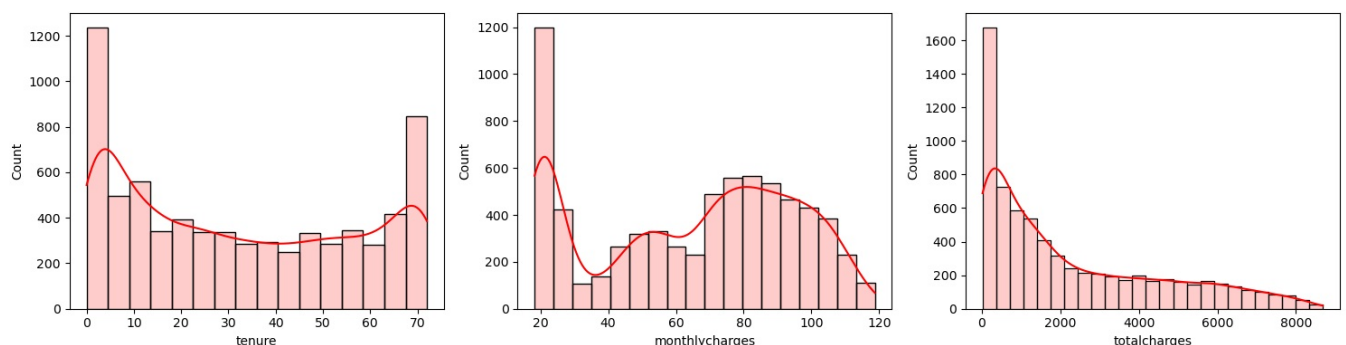
	count	mean	std	min	25%	50%	75%	max
tenure	7043.0	32.371149	24.559481	0.00	9.00	29.000	55.0000	72.00
monthlycharges	7043.0	64.761692	30.090047	18.25	35.50	70.350	89.8500	118.75
totalcharges	7032.0	2283.300441	2266.771362	18.80	401.45	1397.475	3794.7375	8684.80

```
In [29]: data['seniorcitizen'].value_counts()
```

```
Out[29]: seniorcitizen
0      5901
1      1142
Name: count, dtype: int64
```

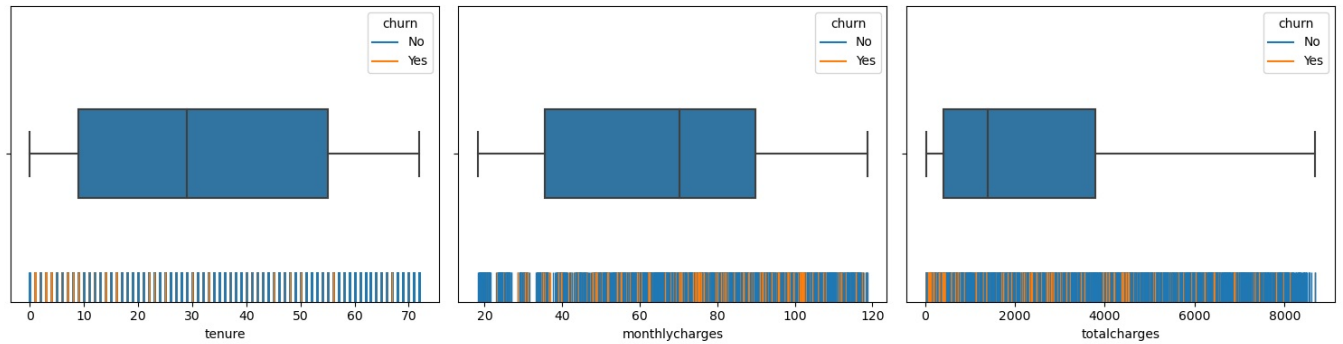
```
In [30]: ord_cols=["dependents","gender","paperlessbilling",'partner','phoneservice']
label='churn'
cat_cols=['seniorcitizen','multiplelines','internetservice','onlinesecurity','onlinebackup','deviceprotection',
```

```
In [34]: plt.figure(figsize=(15,4))
for i,col in enumerate(num_cols):
    plt.subplot(1,3,i+1)
    sns.histplot(data, x=col,color='red',alpha=0.2,kde=True)
plt.tight_layout()
plt.show()
```

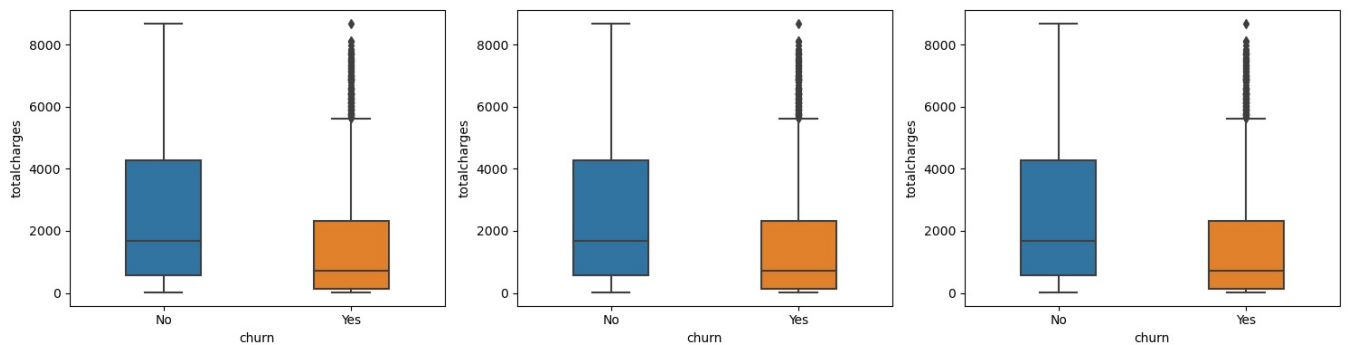


```
In [36]: plt.figure(figsize=(15,4))
for i,col in enumerate(num_cols):
```

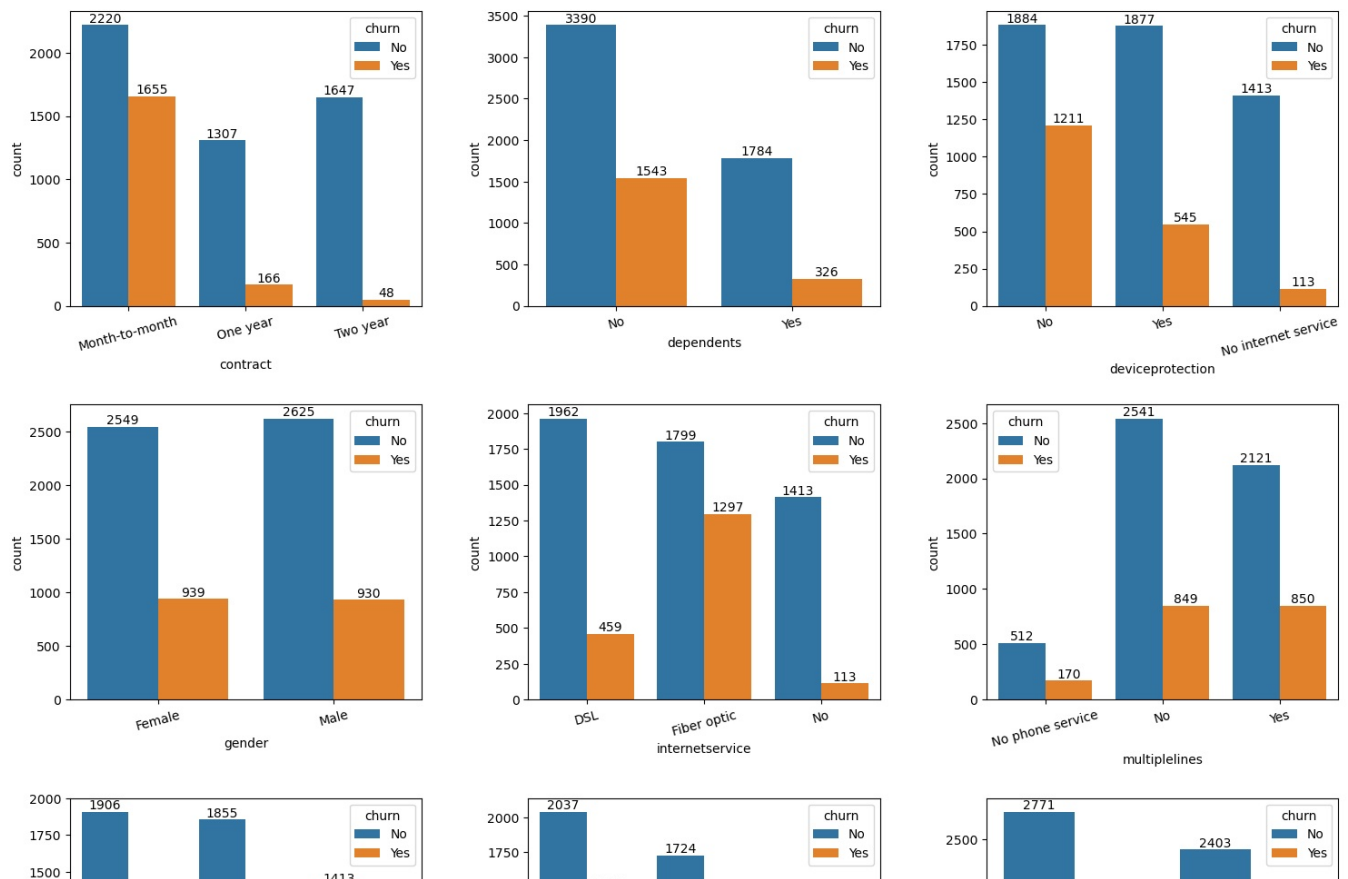
```
plt.subplot(1,3,i+1)
sns.rugplot(data, x=col,hue=label,height=0.1)
sns.boxplot(data,x=col,width=0.3)
plt.tight_layout()
plt.show()
```

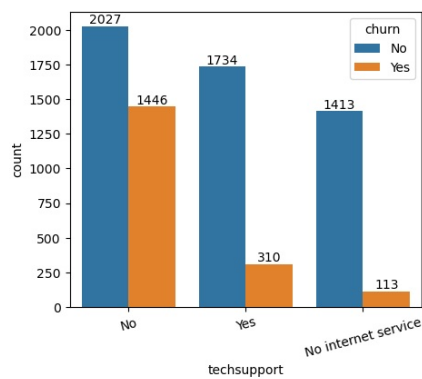
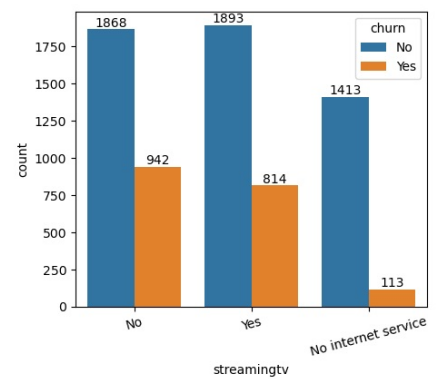
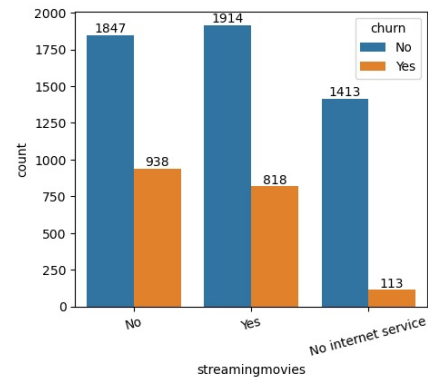
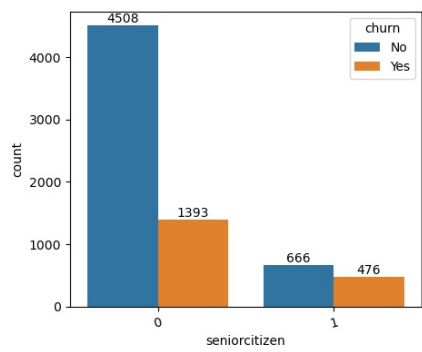
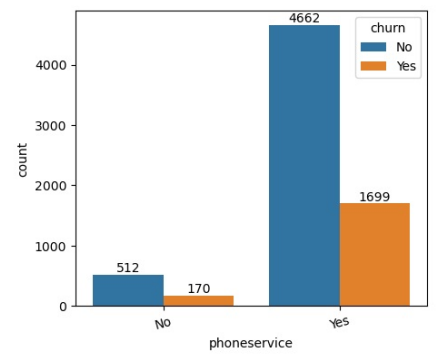
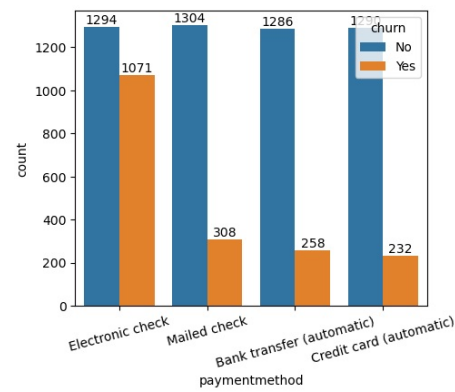
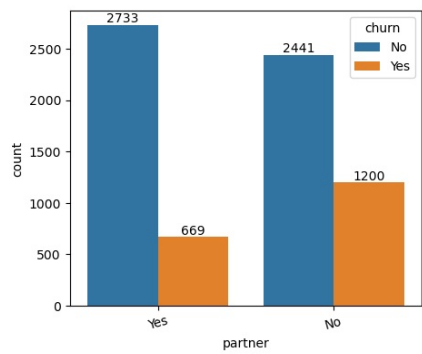
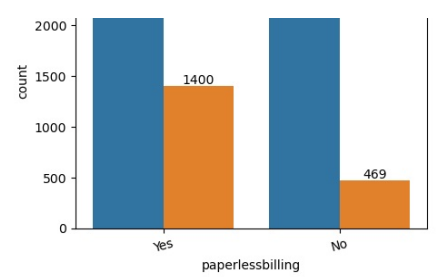
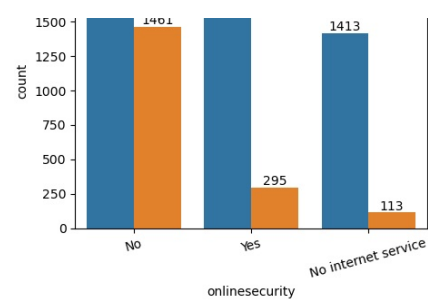
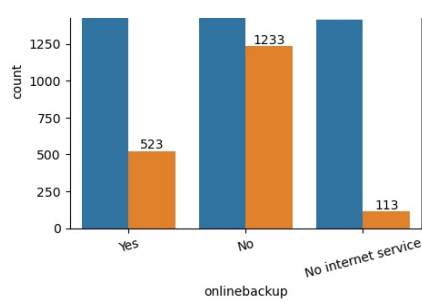


```
In [38]: plt.figure(figsize=(15,4))
for i,columns in enumerate(num_cols):
    plt.subplot(1,3,i+1)
    sns.boxplot(data,x=label,y=col,width=0.4)
plt.tight_layout()
plt.show()
```

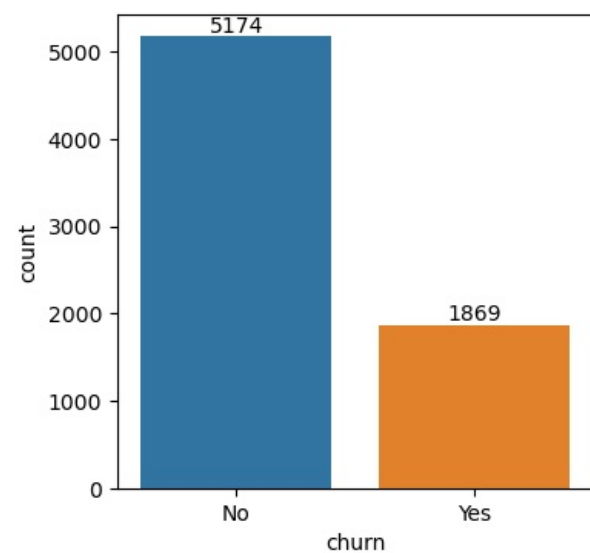


```
In [41]: plt.figure(figsize=(15,26))
for i,col in enumerate(data.columns.difference(num_cols)[1:]):
    plt.subplot(6,3,i+1)
    ax=sns.countplot(data,x=col,hue=label)
    ax.bar_label(ax.containers[0])
    ax.bar_label(ax.containers[1])
    plt.xticks(rotation=15)
plt.tight_layout()
plt.show()
```





```
In [42]: plt.figure(figsize=(4,4))
ax=sns.countplot(data,x=label)
ax.bar_label(ax.containers[0])
plt.show()
```



In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js