jupyter Churn analysis Last Checkpoint: 11 seconds ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

```
# describe statics
```

[9]:
```
df.describe()
```

[9]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.734304
std	0.368612	24.559481	30.090047	2266.794470
min	0.000000	0.000000	18.250000	0.000000
25%	0.000000	9.000000	35.500000	398.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

[11]:
```
df.duplicated().sum()
```

[11]:
```
0
```

[13]:
```
df["customerID"].duplicated().sum()
```

[13]:
```
0
```

```
#convert senior citizen value to yes and no
```

[7]:
```
df.isnull()
```

[7]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport
0	False	False	False	False	False	False	False	False	False	False	...	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7038	False	False	False	False	False	False	False	False	False	False	...	False	False
7039	False	False	False	False	False	False	False	False	False	False	...	False	False
7040	False	False	False	False	False	False	False	False	False	False	...	False	False
7041	False	False	False	False	False	False	False	False	False	False	...	False	False
7042	False	False	False	False	False	False	False	False	False	False	...	False	False

7043 rows × 21 columns

```
# all over null vales ( no null values are there
```

[8]:
```
df.isnull().sum().sum()
```

[8]:
```
0
```

[5]:
```
df["TotalCharges"] = df["TotalCharges"].replace(" ","0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

```
# checking information
```

[6]:
```
df.info()
```

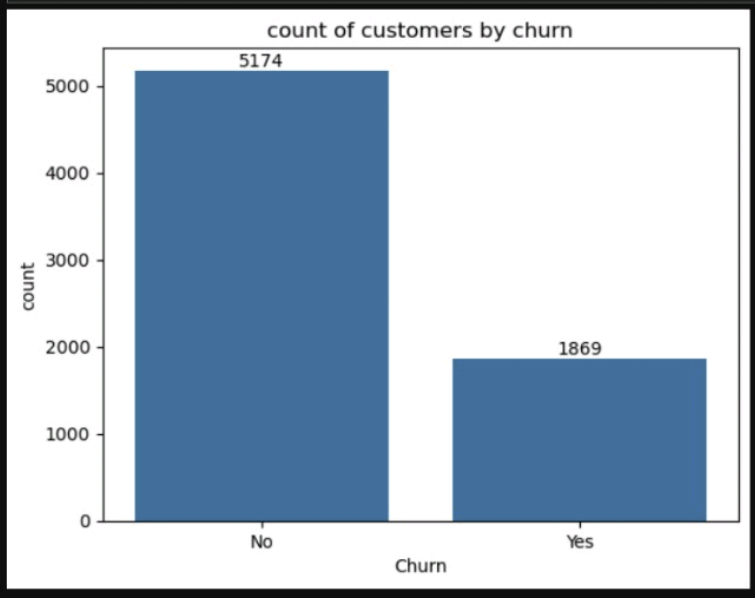
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   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   float64
 20  Churn             7043 non-null   object 
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

[4]:
```
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
```

```
#replace blanks and convert datatype
```

[32]:
```
ax = sns.countplot( x = 'Churn' , data = df )


ax.bar_label(ax.containers[0])
plt.title("count of customers by churn")
plt.show()
```

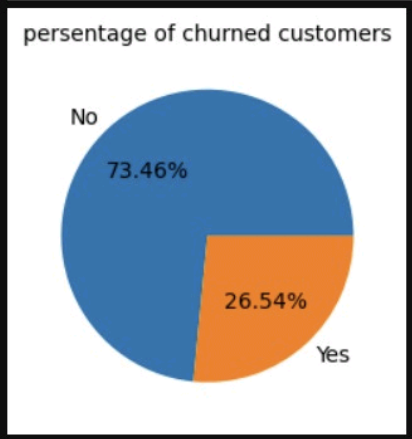


jupyter **Churn analysis** Last Checkpoint: 11 seconds ago

File Edit View Run Kernel Settings Help

Code JupyterLab

[33]:
```
plt.figure(figsize =(3,4))
gb = df.groupby("Churn").agg({ 'Churn' : "count"})
plt.pie(gb['Churn'] , labels = gb.index , autopct = "%1.2f%%")
plt.title("persentage of churned customers" , fontsize = 10)
plt.show()
gb
```

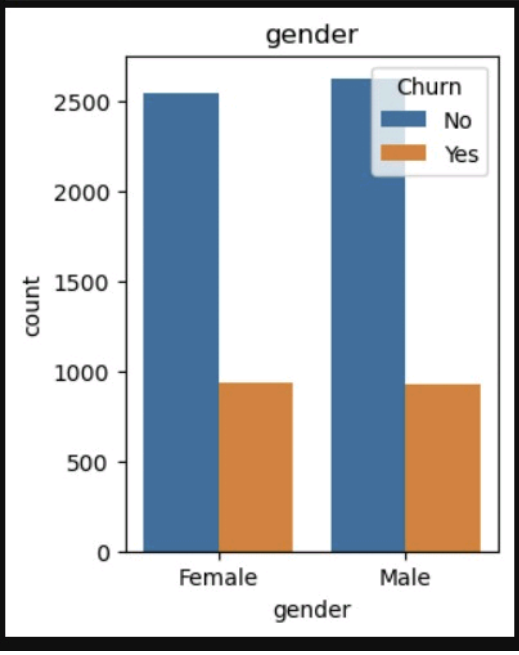


[33]:

Churn	
No	5174
Yes	1869

[ ]:
```
# from the given pie chart we can conclude that 26.54% of our customers out .
# now explore the reason behind it.
```

[41]:
```
plt.figure(figsize = (3,4))

sns.countplot(x = "gender" , data = df , hue = "Churn")
plt.title("gender")
plt.show()
```

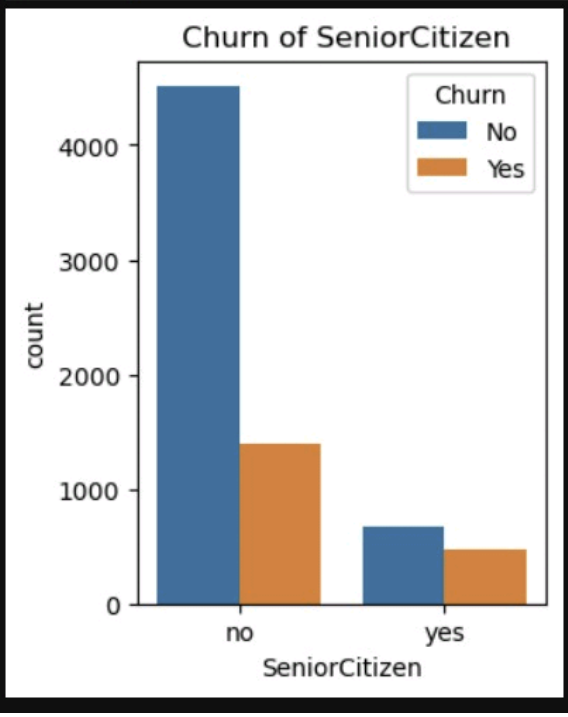


jupyter Churn analysis Last Checkpoint: 11 seconds ago

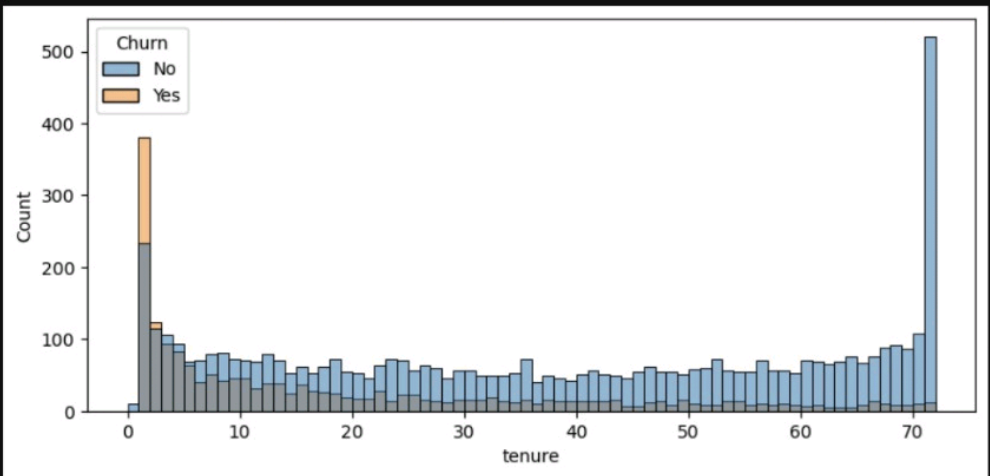
File Edit View Run Kernel Settings Help

Code

[43]:
```
plt.figure(figsize = (3,4))

sns.countplot(x = "SeniorCitizen" , data = df , hue = "Churn")
plt.title(" Churn of SeniorCitizen")
plt.show()
```

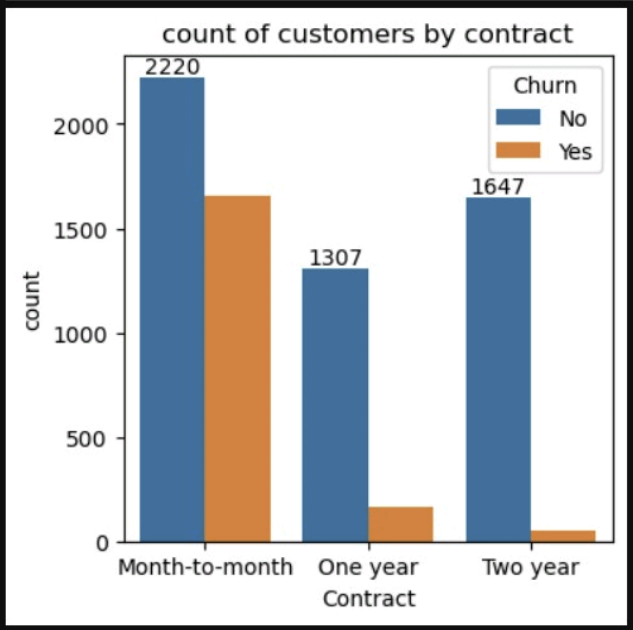


[47]:
```
plt.figure(figsize = (9,4))
sns.histplot(x = "tenure" , data = df , bins = 72 , hue = "Churn")
plt.show()
```




```
#pepole who have used our servies for a long time have stayed and people who have used our servies
```

[49]:
```
plt.figure(figsize =(4,4))
ax = sns.countplot( x = 'Contract' , data = df , hue = "Churn" )


ax.bar_label(ax.containers[0])
plt.title("count of customers by contract")
plt.show()
```



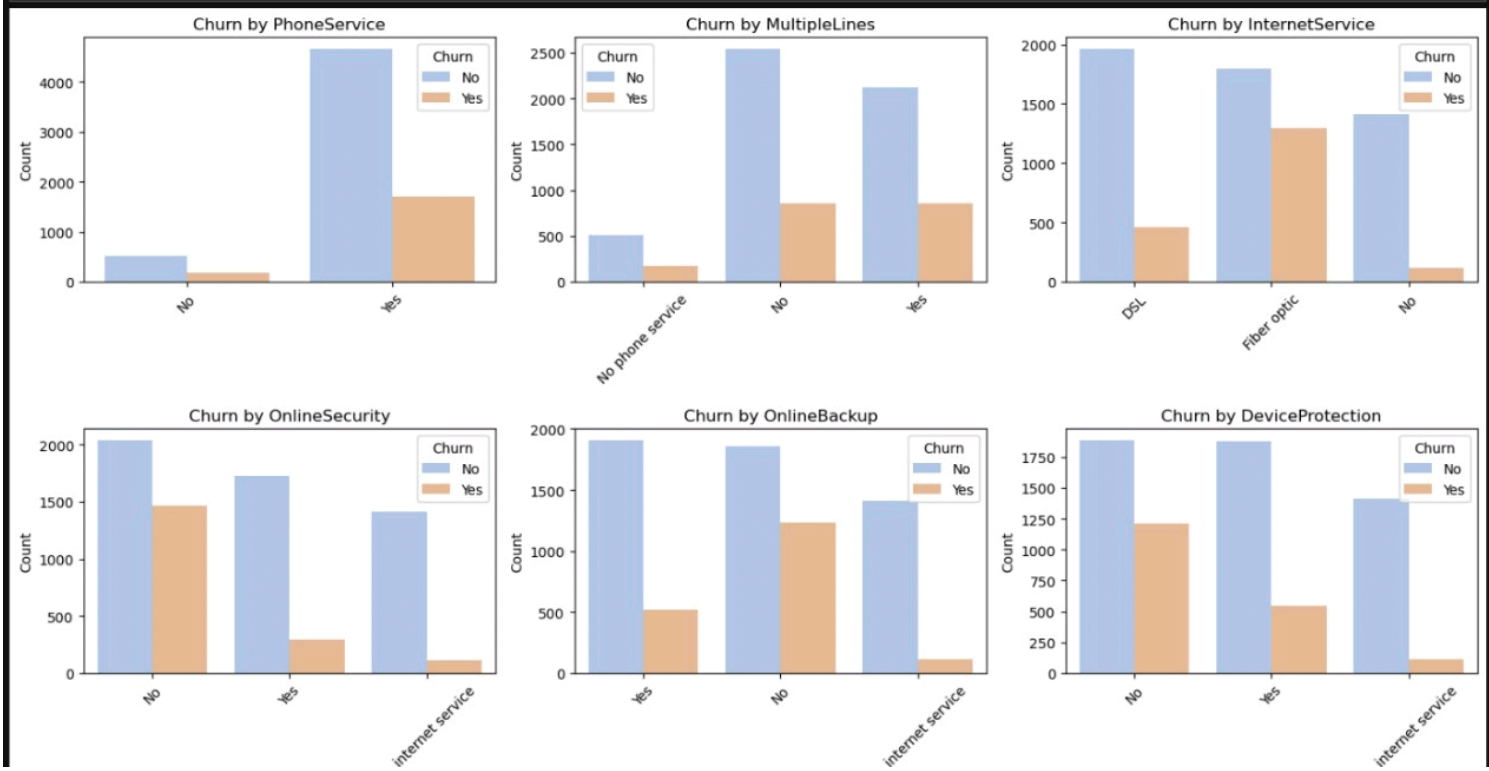
[ ]:
```
#people who have month to month are likely to churn  then from those who have one or two years of cintract
```

[50]:
```
df.columns.values
```

[50]:
```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

[53]:
```
cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

# Number of rows and columns for subplots
n_cols = 3
n_rows = (len(cols) + n_cols - 1) // n_cols  # ceiling division

fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 12))
axes = axes.flatten()

for i, col in enumerate(cols):
    sns.countplot(x=col, data=df, hue="Churn", ax=axes[i], palette="pastel")
    axes[i].set_title(f"Churn by {col}")
    axes[i].set_xlabel("")
    axes[i].set_ylabel("Count")
    axes[i].tick_params(axis='x', rotation=45)

# Remove extra empty subplots if any
for j in range(len(cols), len(axes)):
    fig.delaxes(axes[j])
```

```
plt.tight_layout()
plt.show()
```



```
]: plt.figure(figsize =(6,4))
   ax = sns.countplot( x = 'PaymentMethod' , data = df , hue = "Churn" )


   ax.bar_label(ax.containers[0])
   ax.bar_label(ax.containers[1])
   plt.title("count of customers by Payment Method")
   plt.xticks(rotation = 40)
   plt.show()
```