

## IMPORT CSV FILE:-

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

```
data = pd.read_csv('Telco-Customer-churn.csv')
```

```
data.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure
0	7590-VHVEG	Female	0	Yes	No	1
1	5575-GNVDE	Male	0	No	No	34
2	3668-QPYBK	Male	0	No	No	2
3	7795-CF0CW	Male	0	No	No	45
4	9237-HQITU	Female	0	No	No	2

	MultipleLines	InternetService	OnlineSecurity	...
0	No phone service	DSL	No	...
1	No	DSL	Yes	...
2	No	DSL	Yes	...
3	No phone service	DSL	Yes	...
4	No	Fiber optic	No	...

	TechSupport	StreamingTV	StreamingMovies	Contract
0	No	No	No	Month-to-month
1	No	No	No	One year
2	No	No	No	Month-to-month
3	Yes	No	No	One year
4	No	No	No	Month-to-month

Yes

	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	Electronic check	29.85	29.85	No
1	Mailed check	56.95	1889.5	No
2	Mailed check	53.85	108.15	Yes
3	Bank transfer (automatic)	42.30	1840.75	No
4	Electronic check	70.70	151.65	Yes

[5 rows x 21 columns]

## CLEAN DATA:-

### Checking data info

```
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
```

## Replacing blanks with 0 as tenure is 0 and no total charges are recorded

```
data["TotalCharges"] = data["TotalCharges"].replace(" ", "0")
data["TotalCharges"] = data["TotalCharges"].astype("float")
```

## Checking data info after making some changes

```
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   float64
20   Churn                 7043 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

## Checking for null values

```
data.isnull().sum()
```

```
customerID    0
gender        0
```

```
SeniorCitizen      0
Partner            0
Dependents         0
tenure             0
PhoneService       0
MultipleLines      0
InternetService    0
OnlineSecurity     0
OnlineBackup       0
DeviceProtection   0
TechSupport        0
StreamingTV        0
StreamingMovies    0
Contract           0
PaperlessBilling   0
PaymentMethod      0
MonthlyCharges     0
TotalCharges       0
Churn              0
dtype: int64
```

## Describe our data

```
data.describe()
```

	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

## Checking for duplicate data

```
data.duplicated().sum()
```

```
0
```

```
data["customerID"].duplicated().sum()
```

```
0
```

Convert the values of SeniorCitizen from (0,1) to (no,yes)

```
def conv(value):  
    if value == 1:  
        return "yes"  
    else:  
        return "no"  
  
data["SeniorCitizen"] = data["SeniorCitizen"].apply(conv)
```

After convert the values of SeniorCitizen from (0,1) to (no,yes) checking the data

```
data.head(30)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure
0	7590-VHVEG	Female	no	Yes	No	1
1	5575-GNVDE	Male	no	No	No	34
2	3668-QPYBK	Male	no	No	No	2
3	7795-CF0CW	Male	no	No	No	45
4	9237-HQITU	Female	no	No	No	2
5	9305-CDSKC	Female	no	No	No	8
6	1452-KI0VK	Male	no	No	Yes	22
7	6713-OK0MC	Female	no	No	No	10
8	7892-P00KP	Female	no	Yes	No	28
9	6388-TABGU	Male	no	No	Yes	62
10	9763-GRSKD	Male	no	Yes	Yes	13
11	7469-LKBCI	Male	no	No	No	16
12	8091-TTVAX	Male	no	Yes	No	58
13	0280-XJGEX	Male	no	No	No	49

14	5129-JLPIS	Male	no	No	No	25
Yes						
15	3655-SNQYZ	Female	no	Yes	Yes	69
Yes						
16	8191-XWSZG	Female	no	No	No	52
Yes						
17	9959-W0FKT	Male	no	No	Yes	71
Yes						
18	4190-MFLUW	Female	no	Yes	Yes	10
Yes						
19	4183-MYFRB	Female	no	No	No	21
Yes						
20	8779-QRDMV	Male	yes	No	No	1
No						
21	1680-VDCWW	Male	no	Yes	No	12
Yes						
22	1066-JKSGK	Male	no	No	No	1
Yes						
23	3638-WEABW	Female	no	Yes	No	58
Yes						
24	6322-HRPFA	Male	no	Yes	Yes	49
Yes						
25	6865-JZNK0	Female	no	No	No	30
Yes						
26	6467-CHFZW	Male	no	Yes	Yes	47
Yes						
27	8665-UTDHz	Male	no	Yes	Yes	1
No						
28	5248-YGIJN	Male	no	Yes	No	72
Yes						
29	8773-HHU0Z	Female	no	No	Yes	17
Yes						
	MultipleLines	InternetService	OnlineSecurity		...	\
0	No phone service	DSL	No		...	
1	No	DSL	Yes		...	
2	No	DSL	Yes		...	
3	No phone service	DSL	Yes		...	
4	No	Fiber optic	No		...	
5	Yes	Fiber optic	No		...	
6	Yes	Fiber optic	No		...	
7	No phone service	DSL	Yes		...	
8	Yes	Fiber optic	No		...	
9	No	DSL	Yes		...	
10	No	DSL	Yes		...	
11	No	No	No internet service		...	
12	Yes	Fiber optic	No		...	
13	Yes	Fiber optic	No		...	
14	No	Fiber optic	Yes		...	

15		Yes	Fiber optic		Yes	...
16		No	No	No internet service		...
17		Yes	Fiber optic		Yes	...
18		No	DSL		No	...
19		No	Fiber optic		No	...
20	No phone service		DSL		No	...
21		No	No	No internet service		...
22		No	No	No internet service		...
23		Yes	DSL		No	...
24		No	DSL		Yes	...
25		No	DSL		Yes	...
26		Yes	Fiber optic		No	...
27	No phone service		DSL		No	...
28		Yes	DSL		Yes	...
29		No	DSL		No	...

	DeviceProtection		TechSupport		StreamingTV	\
0	No		No		No	
1	Yes		No		No	
2	No		No		No	
3	Yes		Yes		No	
4	No		No		No	
5	Yes		No		Yes	
6	No		No		Yes	
7	No		No		No	
8	Yes		Yes		Yes	
9	No		No		No	
10	No		No		No	
11	No internet service	No internet service	No internet service	No internet service	No internet service	
12	Yes		No		Yes	
13	Yes		No		Yes	
14	Yes		Yes		Yes	
15	Yes		Yes		Yes	
16	No internet service	No internet service	No internet service	No internet service	No internet service	
17	Yes		No		Yes	
18	Yes		Yes		No	
19	Yes		No		No	
20	Yes		No		No	
21	No internet service	No internet service	No internet service	No internet service	No internet service	
22	No internet service	No internet service	No internet service	No internet service	No internet service	
23	No		Yes		No	
24	No		Yes		No	
25	No		No		No	
26	No		No		Yes	
27	No		No		No	
28	Yes		Yes		Yes	
29	No		No		Yes	

StreamingMovies

Contract PaperlessBilling \

0	No	Month-to-month	Yes
1	No	One year	No
2	No	Month-to-month	Yes
3	No	One year	No
4	No	Month-to-month	Yes
5	Yes	Month-to-month	Yes
6	No	Month-to-month	Yes
7	No	Month-to-month	No
8	Yes	Month-to-month	Yes
9	No	One year	No
10	No	Month-to-month	Yes
11	No internet service	Two year	No
12	Yes	One year	No
13	Yes	Month-to-month	Yes
14	Yes	Month-to-month	Yes
15	Yes	Two year	No
16	No internet service	One year	No
17	Yes	Two year	No
18	No	Month-to-month	No
19	Yes	Month-to-month	Yes
20	Yes	Month-to-month	Yes
21	No internet service	One year	No
22	No internet service	Month-to-month	No
23	No	Two year	Yes
24	No	Month-to-month	No
25	No	Month-to-month	Yes
26	Yes	Month-to-month	Yes
27	No	Month-to-month	No
28	Yes	Two year	Yes
29	Yes	Month-to-month	Yes

	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	Electronic check	29.85	29.85	No
1	Mailed check	56.95	1889.50	No
2	Mailed check	53.85	108.15	Yes
3	Bank transfer (automatic)	42.30	1840.75	No
4	Electronic check	70.70	151.65	Yes
5	Electronic check	99.65	820.50	Yes
6	Credit card (automatic)	89.10	1949.40	No
7	Mailed check	29.75	301.90	No
8	Electronic check	104.80	3046.05	Yes
9	Bank transfer (automatic)	56.15	3487.95	No
10	Mailed check	49.95	587.45	No
11	Credit card (automatic)	18.95	326.80	No
12	Credit card (automatic)	100.35	5681.10	No
13	Bank transfer (automatic)	103.70	5036.30	Yes
14	Electronic check	105.50	2686.05	No
15	Credit card (automatic)	113.25	7895.15	No
16	Mailed check	20.65	1022.95	No



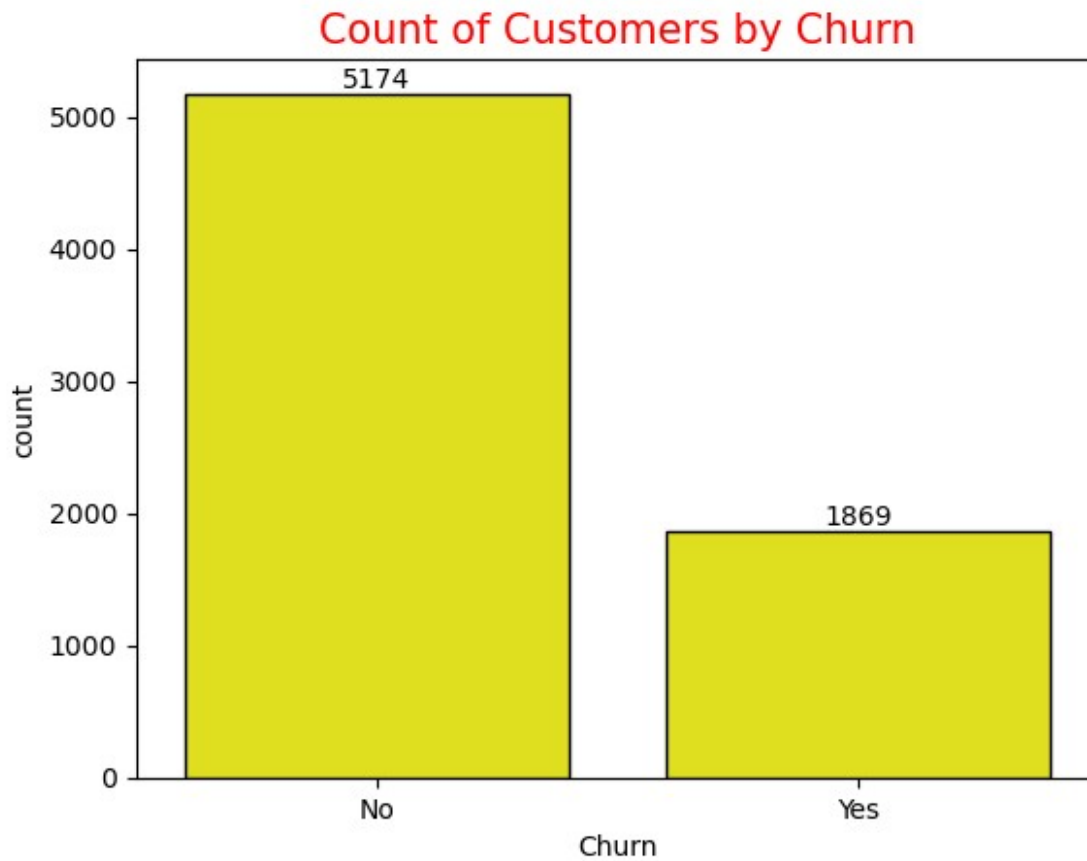
17	Bank transfer (automatic)	106.70	7382.25	No
18	Credit card (automatic)	55.20	528.35	Yes
19	Electronic check	90.05	1862.90	No
20	Electronic check	39.65	39.65	Yes
21	Bank transfer (automatic)	19.80	202.25	No
22	Mailed check	20.15	20.15	Yes
23	Credit card (automatic)	59.90	3505.10	No
24	Credit card (automatic)	59.60	2970.30	No
25	Bank transfer (automatic)	55.30	1530.60	No
26	Electronic check	99.35	4749.15	Yes
27	Electronic check	30.20	30.20	Yes
28	Credit card (automatic)	90.25	6369.45	No
29	Mailed check	64.70	1093.10	Yes

[30 rows x 21 columns]

## PERFORM ANALYSIS:-

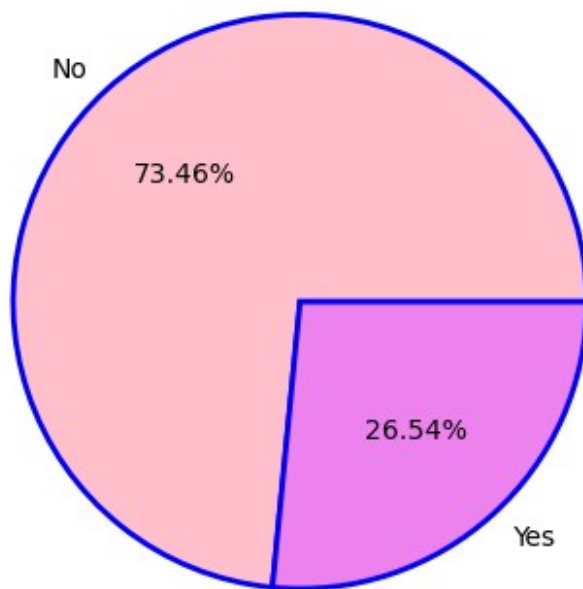
Checking for the number of customer who had churned

```
ax = sns.countplot(x = 'Churn', data = data,color = "yellow",edgecolor
= "black")
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Churn", fontsize = 15, color = 'red')
plt.show()
```



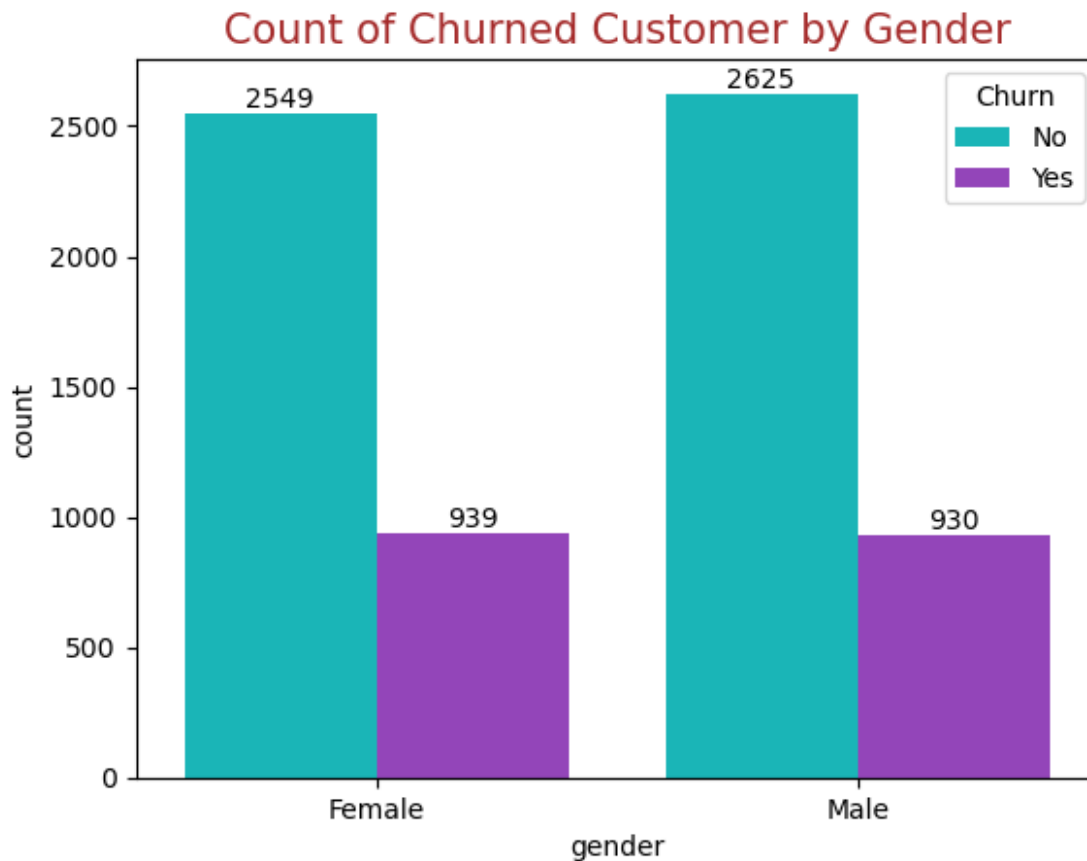
```
gb = df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%", colors =
["pink", "violet"], wedgeprops = {'edgecolor':'blue','linewidth':2})
plt.title("Percentage of Churned Customers",fontsize = 15, color =
'darkblue',)
plt.show()
```

## Percentage of Churned Customers



## Checking for the number of churned customers on the basis of gender

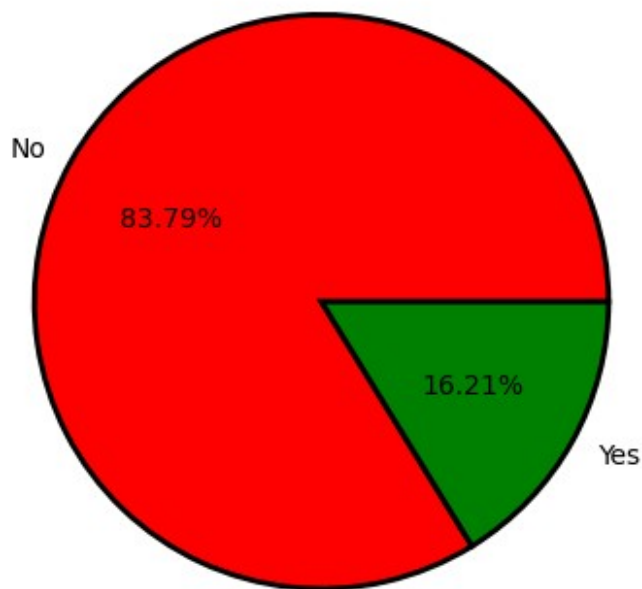
```
colors = {'Yes': 'darkorchid', 'No': 'darkturquoise'}
ax = sns.countplot(x = 'gender', data = data, hue = 'Churn', palette =
colors)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Count of Churned Customer by Gender", color = 'brown',
fontsize = 15)
plt.show()
```



Percentage of customers on the basis of senior citizens

```
gb = df.groupby("SeniorCitizen").agg({'SeniorCitizen':"count"})
plt.pie(gb['SeniorCitizen'], labels = ['No', 'Yes'], autopct = "%1.2f%",
        colors = ["red", "green"], wedgeprops = {'edgecolor':'black','linewidth':2})
plt.title("Percentage of Senior-Citizen Customers", fontsize = 15, color = 'blue',)
plt.show()
```

## Percentage of Senior-Citizen Customers



Checking for the number of churned customers on the basis of seniorcitizens

```
# Calculate percentage values
data_grouped = data.groupby('SeniorCitizen')
['Churn'].value_counts(normalize=True).unstack() * 100

# Create figure and axis
fig, ax = plt.subplots(figsize=(5,5))

# Plot the data with stacking
data_grouped.plot(kind='bar', stacked=True, ax=ax, color=['green',
'lightgreen'])

# Add percentage labels to the bars
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.text(x + width / 2, y + height / 2, f'{height:.1f}%',
ha='center', va='center')

# Customize the plot
plt.title("Percentage of Churned Customers by Senior Citizen Status",
```

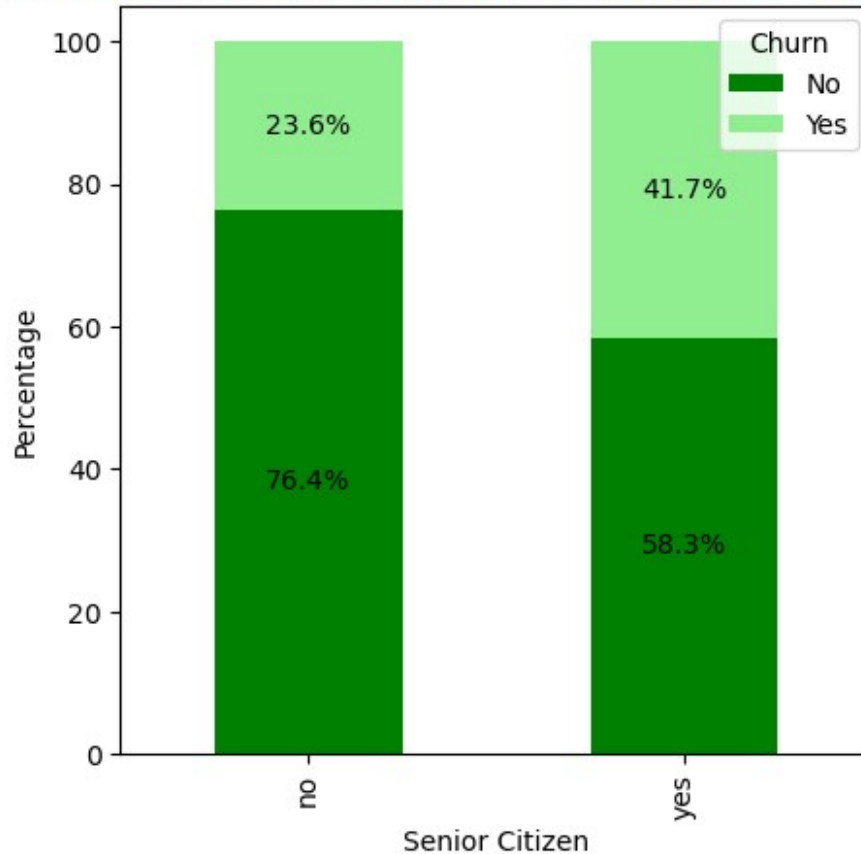
```

color='brown', fontsize=15)
plt.ylabel("Percentage")
plt.xlabel("Senior Citizen")
plt.legend(title="Churn", loc = 'upper right')

# Show the plot
plt.show()

```

## Percentage of Churned Customers by Senior Citizen Status

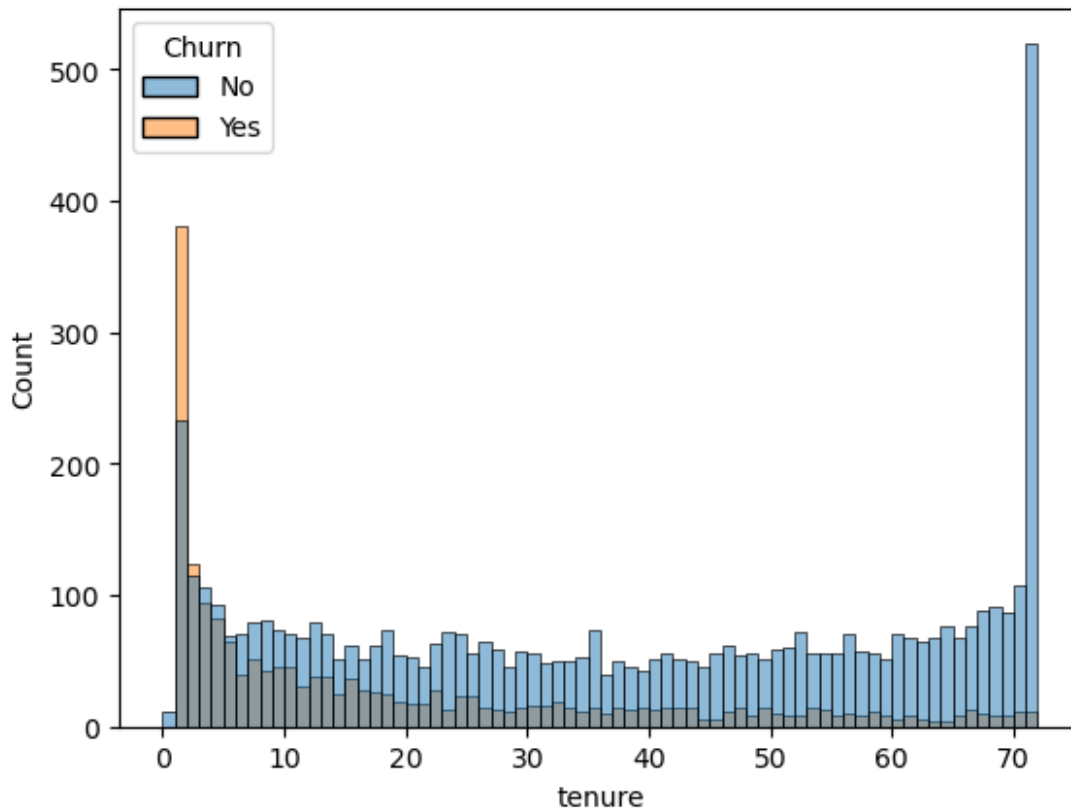


Checking for number of churned customers on the basis of tenure

```

plt.figure(figsize = (9,4))
sns.histplot(x = "tenure", data = data, bins = 72, hue = "Churn",)
plt.show()

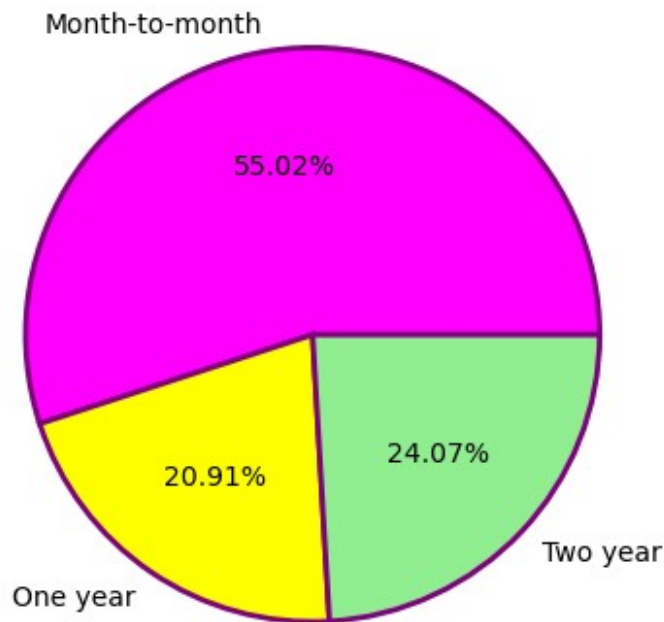
```



## Percentage of customers on the basis of different contracts

```
gb = df.groupby("Contract").agg({'Contract': "count"})
plt.pie(gb['Contract'], labels = gb.index, autopct = "%1.2f%%", colors = ["magenta", "yellow", "lightgreen"], wedgeprops = {'edgecolor': 'purple', 'linewidth': 2})
plt.title("Percentage of Customers of Different Contracts", fontsize = 15, color = 'aqua',)
plt.show()
```

## Percentage of Customers of Different Contracts



Checking for the number of churned customers on the basis of different contracts

```
# Calculate percentage values
data_grouped = data.groupby('Contract')
['Churn'].value_counts(normalize=True).unstack() * 100

# Create figure and axis
fig, ax = plt.subplots(figsize=(10,5))

# Plot the data with stacking
data_grouped.plot(kind='bar', stacked=True, ax=ax,
color=['orchid','mediumaquamarine'])

# Add percentage labels to the bars
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.text(x + width / 2, y + height / 2, f'{height:.1f}%',
ha='center', va='center')

# Customize the plot
plt.title("Percentage of Churned Customers by Senior Citizen Status",
```

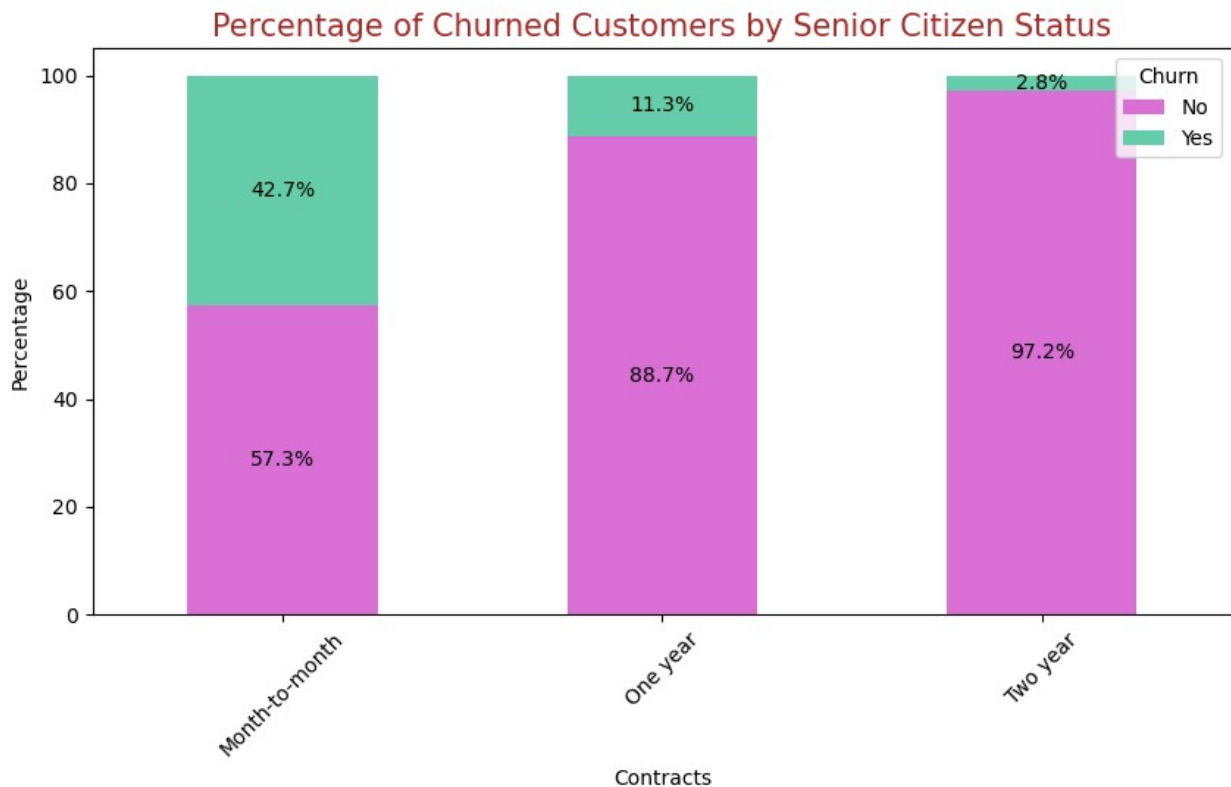


```

color='brown', fontsize=15)
plt.ylabel("Percentage")
plt.xlabel("Contracts")
plt.legend(title="Churn", loc = 'upper right')
plt.xticks(rotation = 45)

# Show the plot
plt.show()

```



## Display all the columns

```

data.columns.values

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

```

## Counting the number of churned customers on the basis of PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies

```
# Define the columns to plot
columns = ['PhoneService', 'MultipleLines', 'InternetService',
           'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
           'TechSupport', 'StreamingTV', 'StreamingMovies']

# Create subplots
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 12))
fig.suptitle("Count Plots of Different Services on the Basis of Churn",
             fontsize=18, color='blue')

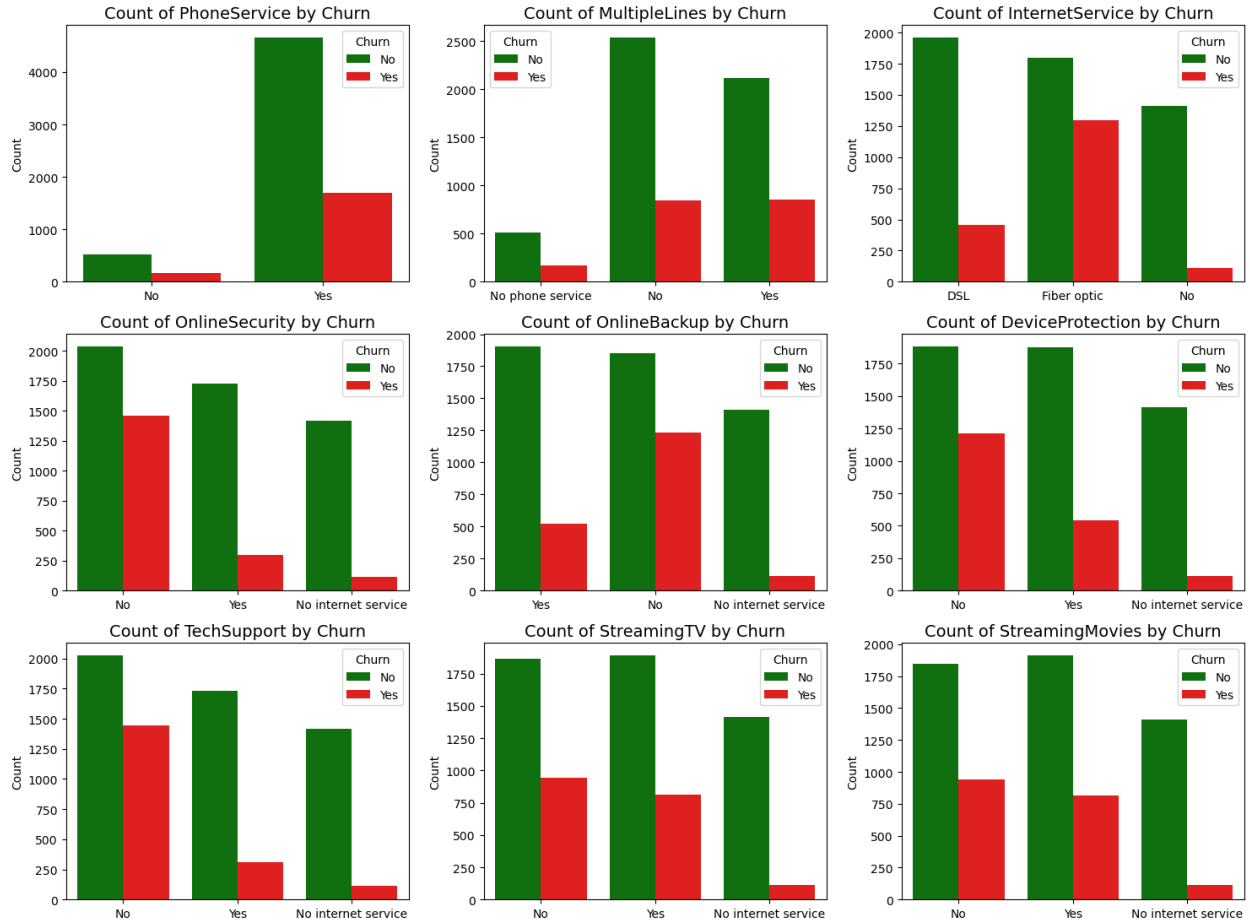
# Flatten the axes array for easy indexing
axes = axes.flatten()

# Define custom colors for Churn categories
churn_colors = {'Yes': 'red', 'No': 'green'} # You can change these colors

# Loop through each column and create a count plot with Churn hue
for i, col in enumerate(columns):
    sns.countplot(x=data[col], hue=data['Churn'], ax=axes[i],
                  palette=churn_colors)
    axes[i].set_title(f"Count of {col} by Churn", fontsize=14)
    axes[i].set_xlabel("")
    axes[i].set_ylabel("Count")

# Adjust layout
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```

### Count Plots of Different Services on the Basis of Churn



Checking for the number of churned customers on the basis of payment method

```
colors = {'Yes': 'goldenrod', 'No': 'lawngreen'}
ax = sns.countplot(x = 'PaymentMethod', data = data, hue = 'Churn',
palette = colors)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Count of Churned Customer by Payment Method", color =
'brown', fontsize = 15)
plt.xticks(rotation = 45)
plt.show()
```

Count of Churned Customer by Payment Method

