# EDA

In [35]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')
```

In [36]:

```
df = pd.read_csv("C:/Data/Telco-Customer-Churn.csv")
df.head()
```

Out[36]:

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | Mul |
|---|---|---|---|---|---|---|---|---|
| **0** | 7590-<br>VHVEG | Female | 0 | Yes | No | 1 | No | |
| **1** | 5575- | Male | 0 | No | No | 34 | Yes | |
| **2** | 3668-<br>QPYBK | Male | 0 | No | No | 2 | Yes | |
| **3** | 7795- | Male | 0 | No | No | 45 | No | |
| **4** | 9237-<br>HQITU | Female | 0 | No | No | 2 | Yes | |

5 rows × 21 columns

In [37]:

```
df.shape
```

Out[37]:

```
(7043, 21)
```

In [38]:

```
df.isna().sum()
```

Out[38]:

```
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
```

# Analysis

In [39]:

```
df.drop(["customerID"], inplace = True, axis = 1)
```

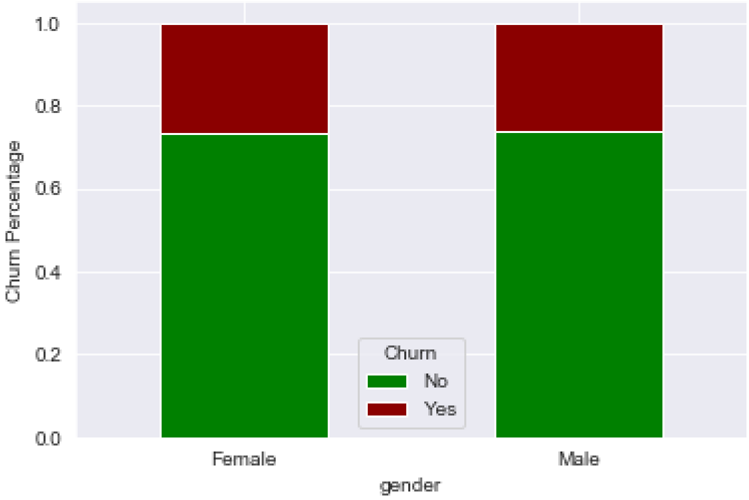
In [40]:

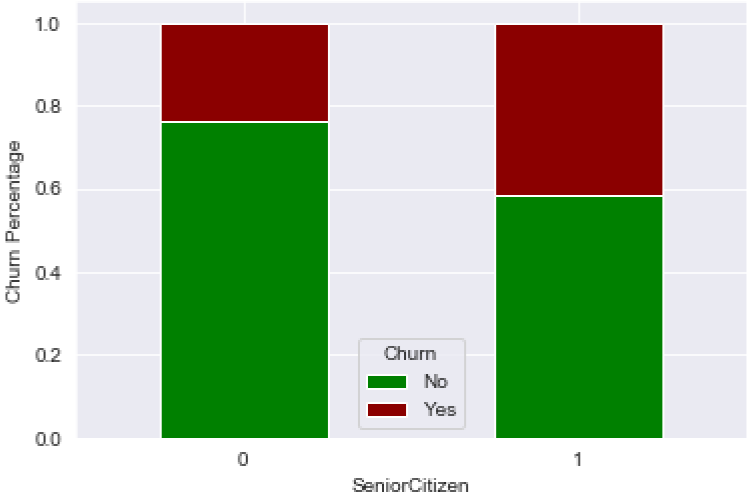
```
def stacked_plot(df, group, target):
    """
    Function to generate a stacked plots between two variables
    """
    fig, ax = plt.subplots(figsize = (6,4))
    temp_df = (df.groupby([group, target]).size()/df.groupby(group)[target].cour
    temp_df.plot(kind='bar', stacked=True, ax = ax, color = ["green", "darkred"]
    ax.xaxis.set_tick_params(rotation=0)
    ax.set_xlabel(group)
    ax.set_ylabel('Churn Percentage')
```

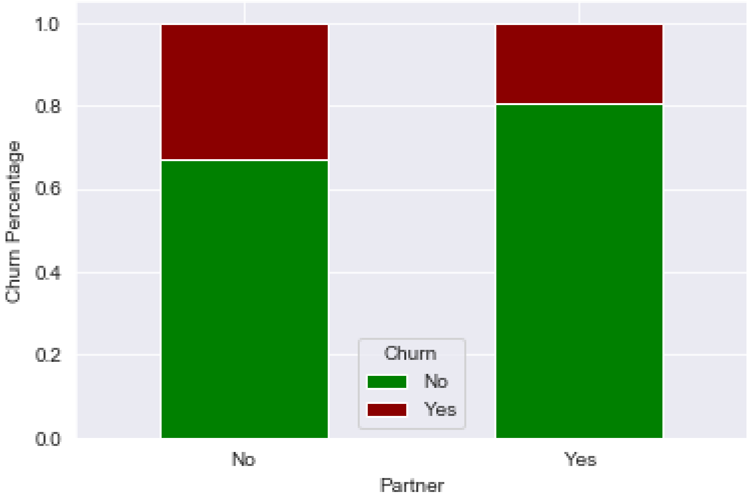
## Gender, SeniorCitizen, Partner, Dependents

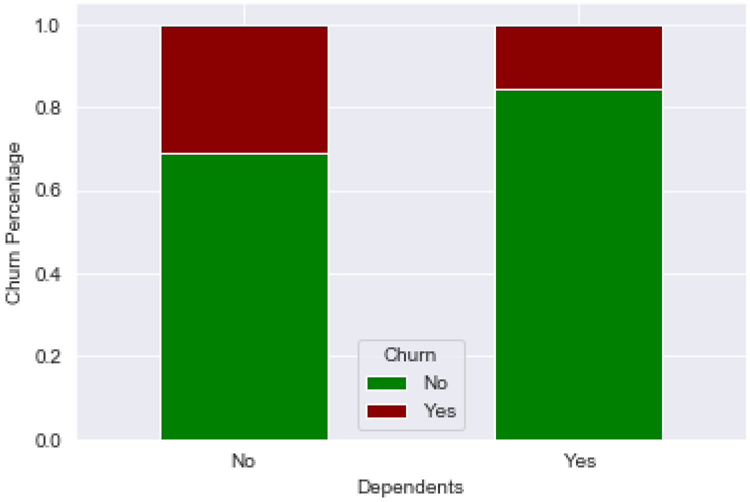
In [8]:

```
stacked_plot(df, "gender", "Churn")
stacked_plot(df, "SeniorCitizen", "Churn")
stacked_plot(df, "Partner", "Churn")
stacked_plot(df, "Dependents", "Churn")
```


1.0
0.8
0.6
0.4
0.2
0.0
Churn Percentage
Churn
No
Yes
Female
Male
gender


1.0
0.8
0.6
0.4
0.2
0.0
Churn Percentage
Churn
No
Yes
0
1
SeniorCitizen


1.0
0.8
0.6
0.4
0.2
0.0
Churn Percentage
Churn
No
Yes
No
Yes
Partner



In [9]:
```
df[(df.SeniorCitizen == 0) & (df.Partner == 'Yes') & (df.Dependents == 'Yes')].C
```

Out[9]:
```
No      1437
Yes      229
Name: Churn, dtype: int64
```

In [10]:
```
df[(df.SeniorCitizen == 0) & (df.Partner == 'Yes') & (df.Dependents == 'No')].Ch
```

Out[10]:
```
No      921
Yes     242
Name: Churn, dtype: int64
```

In [11]:
```
df[(df.SeniorCitizen == 0) & (df.Partner == 'No') & (df.Dependents == 'Yes')].Ch
```

Out[11]:
```
No      278
Yes      75
Name: Churn, dtype: int64
```

In [12]:
```
df[(df.SeniorCitizen == 0) & (df.Partner == 'No') & (df.Dependents == 'No')].Chu
```

Out[12]:
```
No      1872
Yes      847
Name: Churn, dtype: int64
```

## Tenure

In [10]:
```
df['tenure'].describe()
```

Out[10]:
```
count    7043.000000
mean       32.371149
std        24.559481
min         0.000000
25%         9.000000
50%        29.000000
75%        55.000000
max        72.000000
Name: tenure, dtype: float64
```

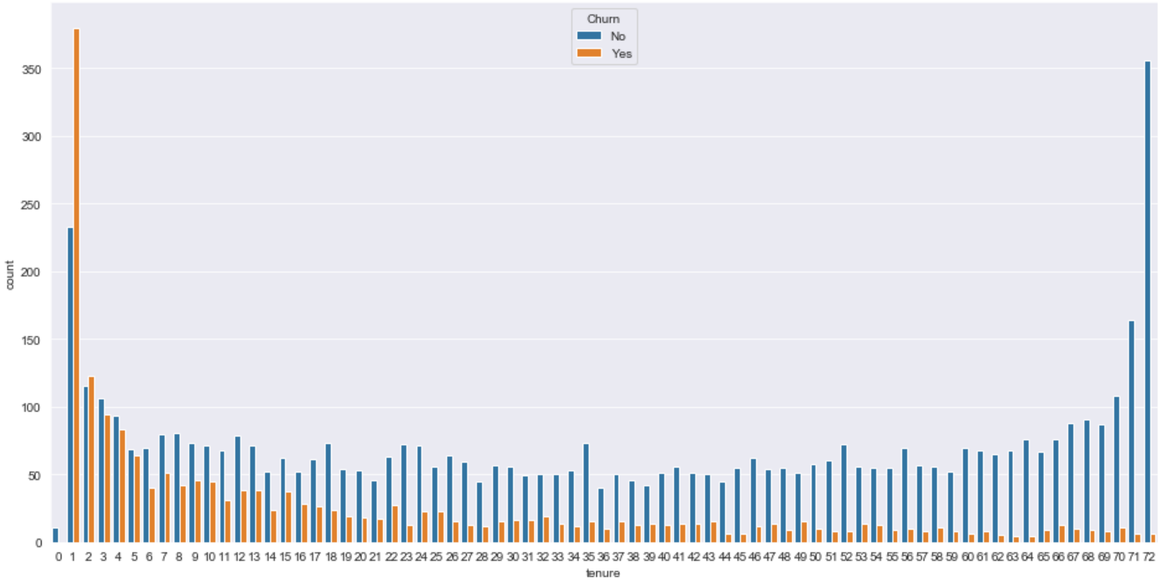
In [11]:
```
df['tenure'].value_counts().head(10)
```

Out[11]:

```
1     613
72    362
2     238
3     200
4     176
71    170
5     133
7     131
8     123
70    119
Name: tenure, dtype: int64
```

In [12]:

```
plt.figure(figsize=(16,8))
sns.countplot(x="tenure", hue="Churn", data=df)
plt.show()
```



In [41]:

```
def tenure(t):
    if t<=12:
        return 1
    elif t>12 and  t<=24:
        return 2
    elif t>24 and  t<=36:
        return 3
    elif t>36 and  t<=48:
        return 4
    elif t>48 and  t<=60:
        return 5
    else:
        return 6

df["tenure_group"]=df["tenure"].apply(lambda  x:  tenure(x))
```

In [42]:

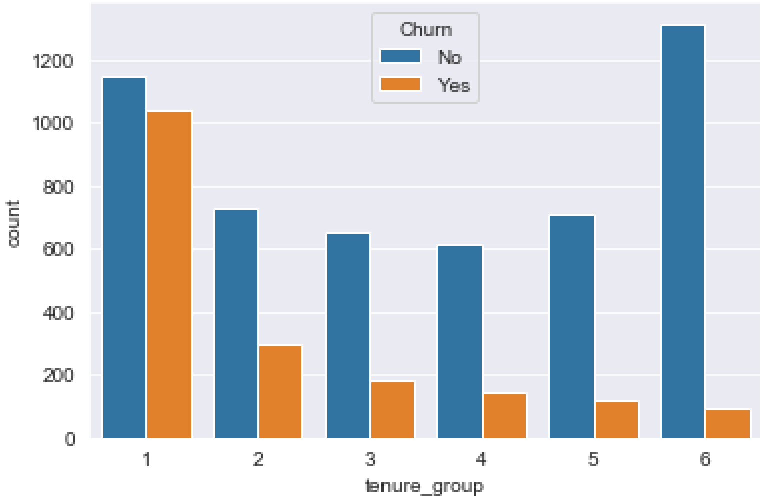
```
df["tenure_group"].value_counts()
```

Out[42]:

```
1    2186
6    1407
2    1024
3     832
5     832
4     762
Name: tenure_group, dtype: int64
```

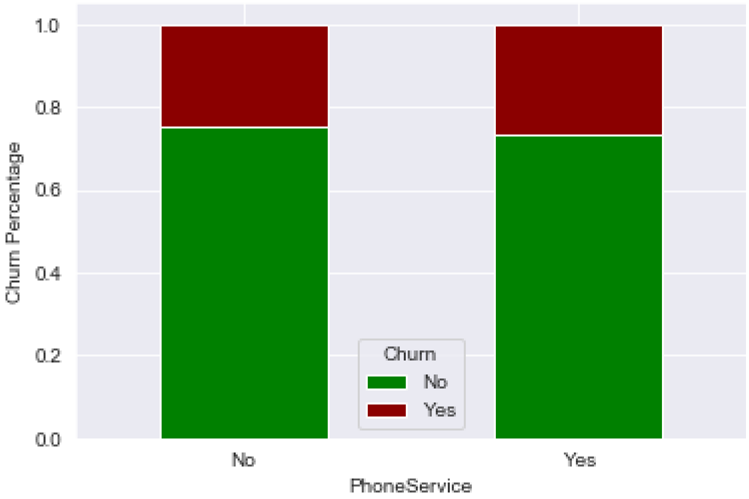
```
In [43]: sns.countplot(x="tenure_group", hue="Churn", data=df)
```

```
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x2128cfcf080>
```

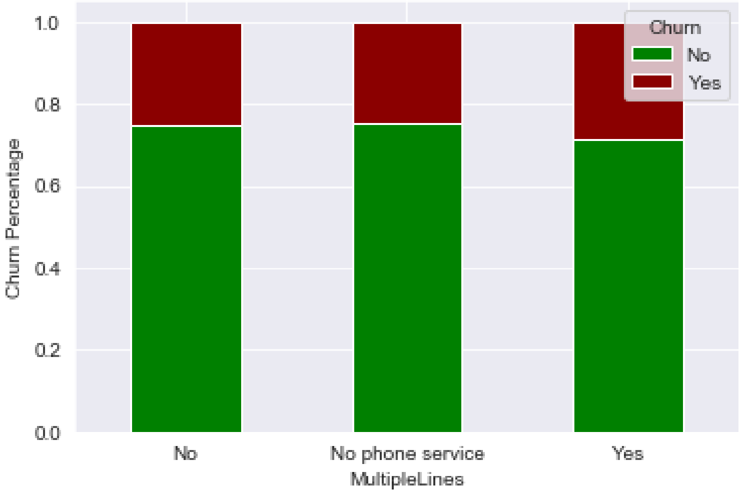


## Phone Service and MultipleLines

```
In [19]: stacked_plot(df, "PhoneService", "Churn")
```



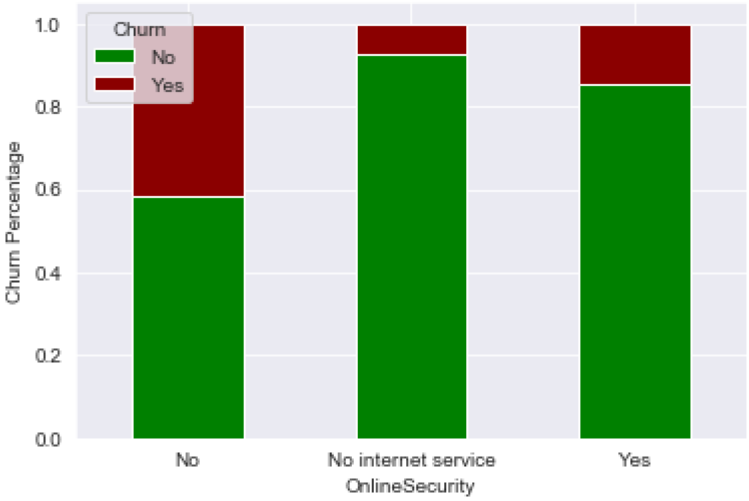
```
In [20]: stacked_plot(df, "MultipleLines", "Churn")
```

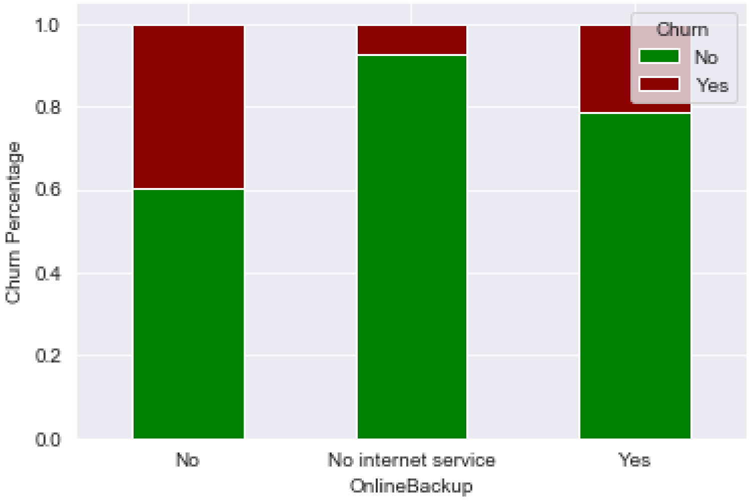


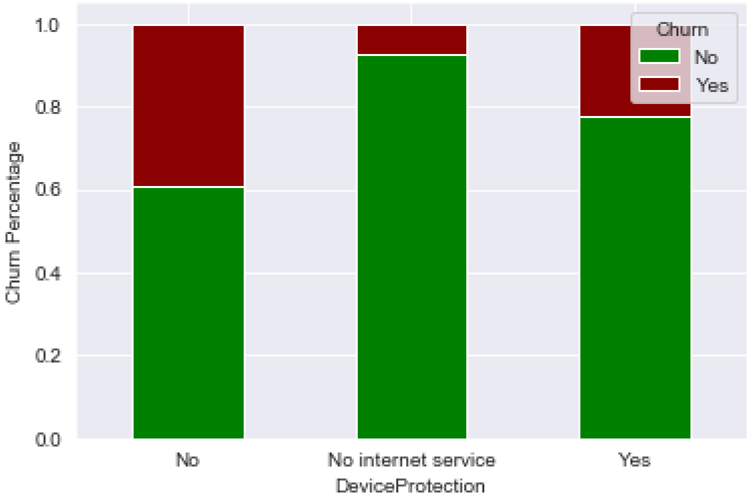
## OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies

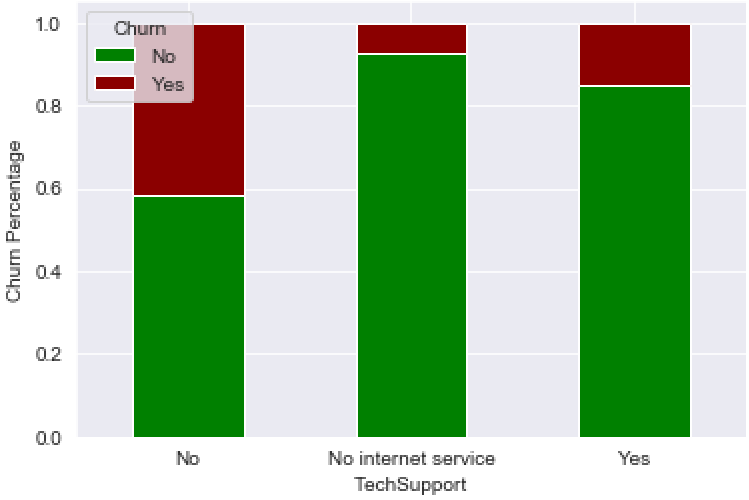
In [18]:

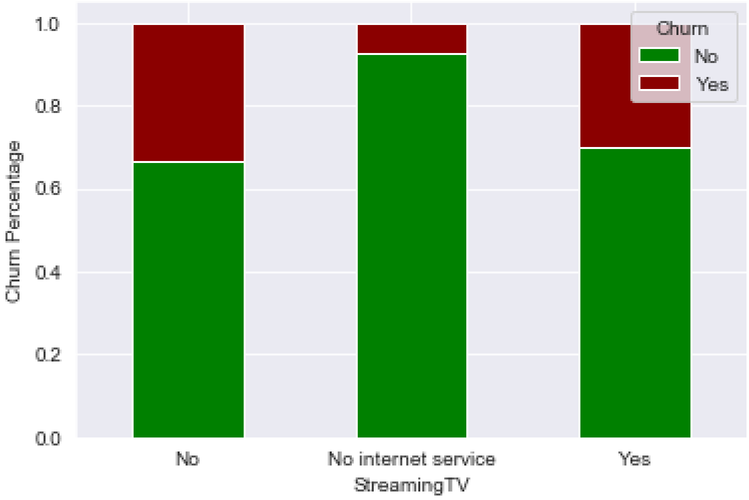
```
stacked_plot(df, "OnlineSecurity", "Churn")
stacked_plot(df, "OnlineBackup", "Churn")
stacked_plot(df, "DeviceProtection", "Churn")
stacked_plot(df, "TechSupport", "Churn")
stacked_plot(df, "StreamingTV", "Churn")
stacked_plot(df, "StreamingMovies", "Churn")
```

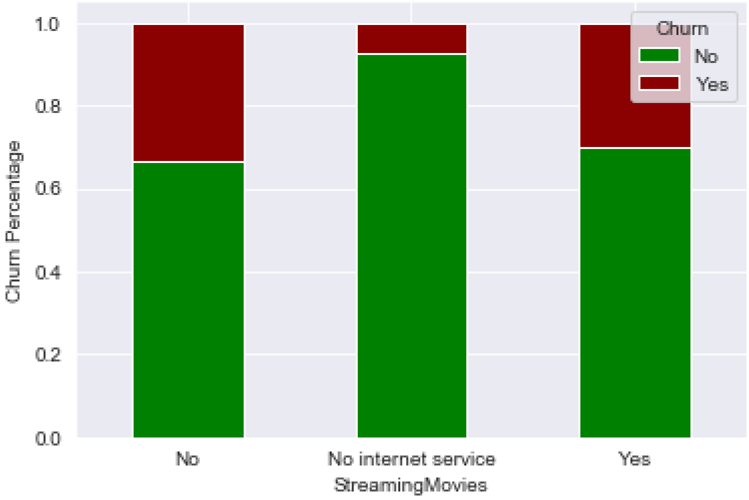






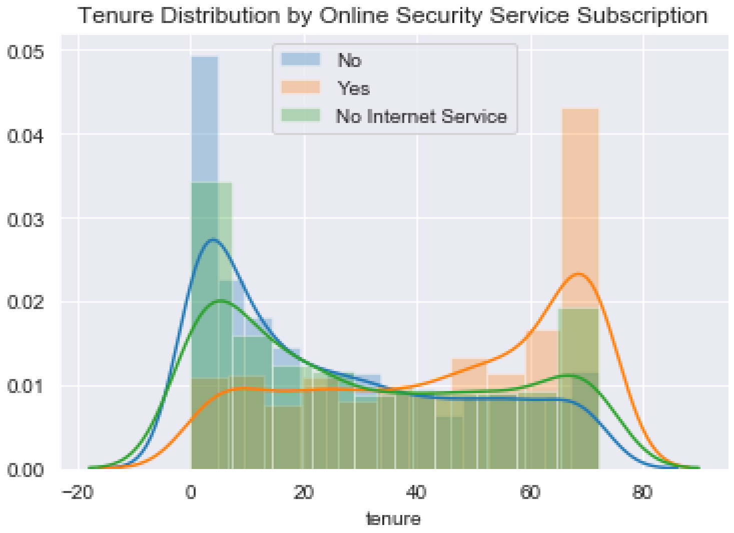




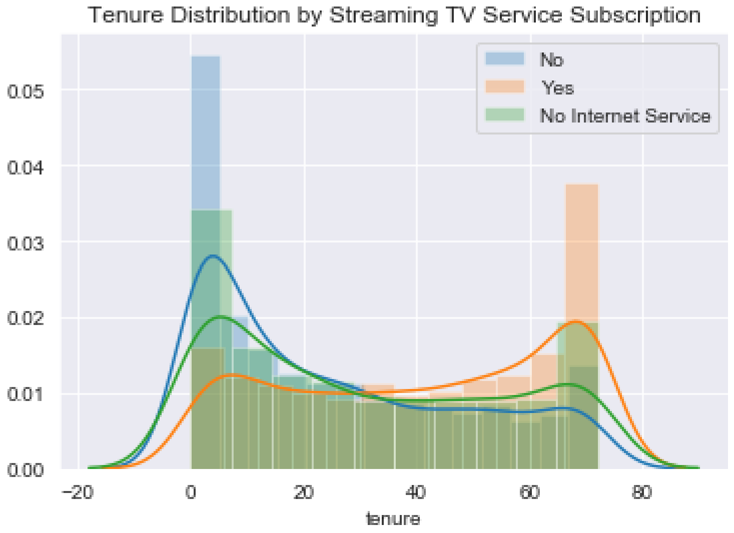


In [33]:

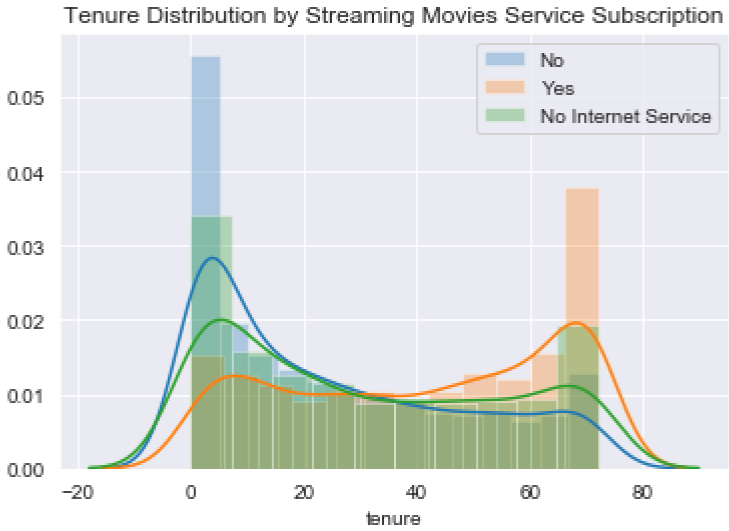
```
sns.distplot(df.tenure[df.OnlineSecurity == "No"], hist_kws=dict(alpha=0.3), lab
sns.distplot(df.tenure[df.OnlineSecurity == "Yes"], hist_kws=dict(alpha=0.3), la
sns.distplot(df.tenure[df.OnlineSecurity == "No internet service"], hist_kws=dic
plt.title("Tenure Distribution by Online Security Service Subscription")
plt.legend()
plt.show()
```



In [32]:
```
sns.distplot(df.tenure[df.StreamingTV == "No"], hist_kws=dict(alpha=0.3), label=
sns.distplot(df.tenure[df.StreamingTV == "Yes"], hist_kws=dict(alpha=0.3), label
sns.distplot(df.tenure[df.StreamingTV == "No internet service"], hist_kws=dict(a
plt.title("Tenure Distribution by Streaming TV Service Subscription")
plt.legend()
plt.show()
```

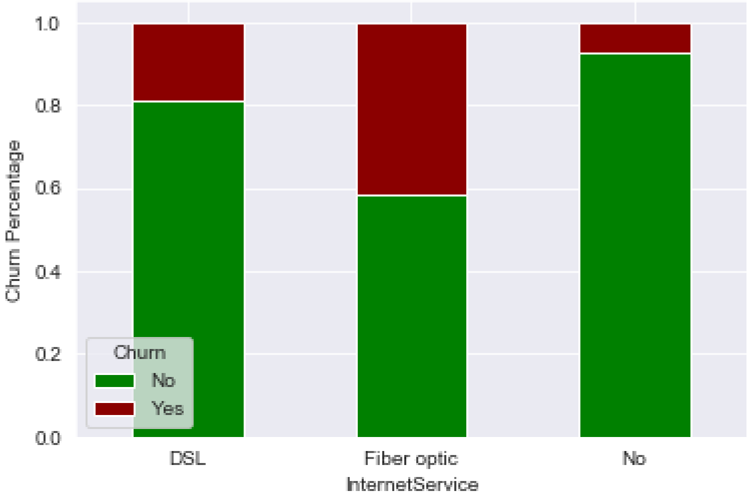


In [31]:
```
sns.distplot(df.tenure[df.StreamingMovies == "No"], hist_kws=dict(alpha=0.3), la
sns.distplot(df.tenure[df.StreamingMovies == "Yes"], hist_kws=dict(alpha=0.3), l
sns.distplot(df.tenure[df.StreamingMovies == "No internet service"], hist_kws=di
plt.title("Tenure Distribution by Streaming Movies Service Subscription")
plt.legend()
plt.show()
```



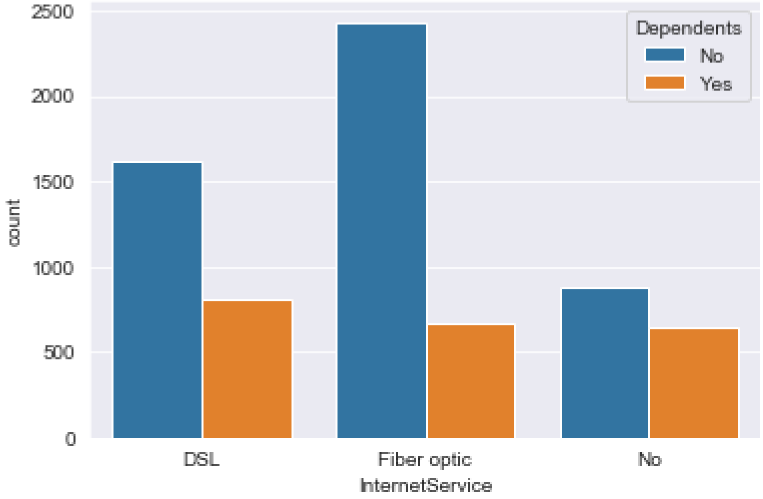
## InternetService

In [34]:
```
stacked_plot(df, "InternetService", "Churn")
```

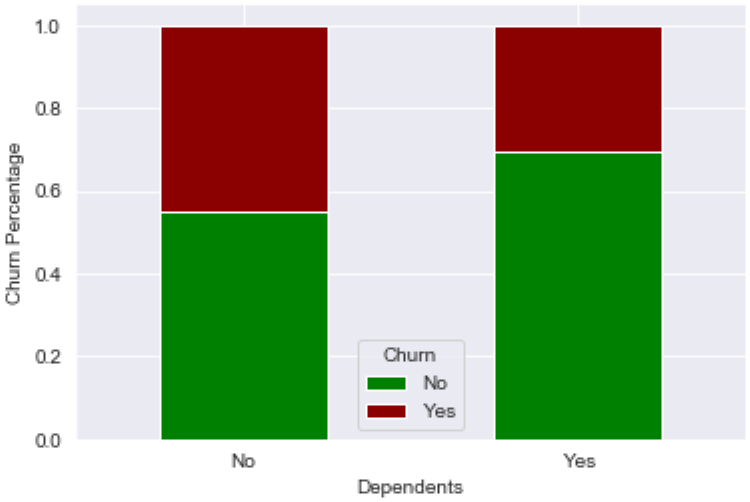


In [37]:
```
sns.countplot(df.InternetService, hue = df.Dependents)
```

Out[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ef073862e8>
```

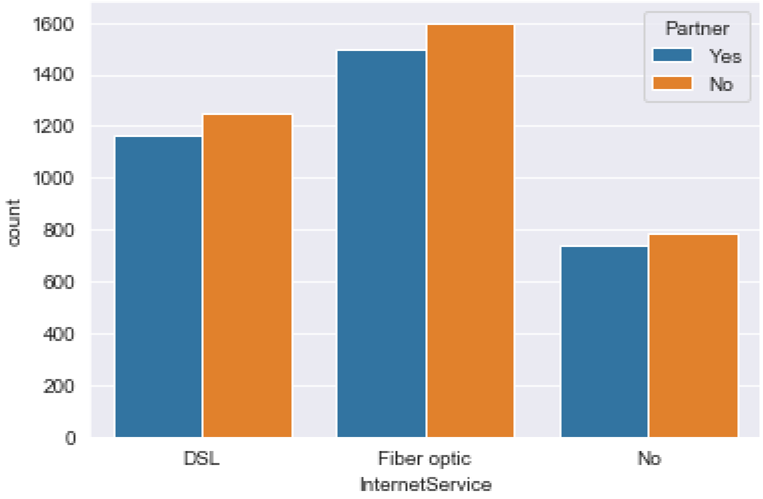


In [43]:
```
stacked_plot(df[df.InternetService == "Fiber optic"], "Dependents", "Churn")
```



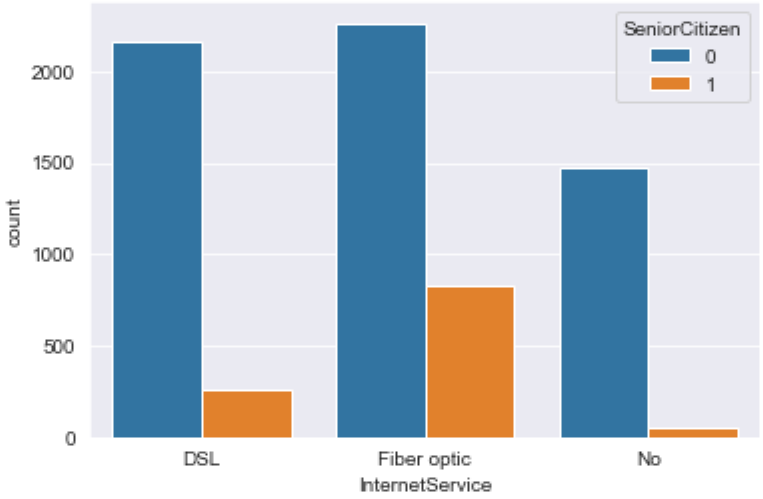
```
In [38]: sns.countplot(df.InternetService, hue = df.Partner)
```

```
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x1ef0755d390>
```

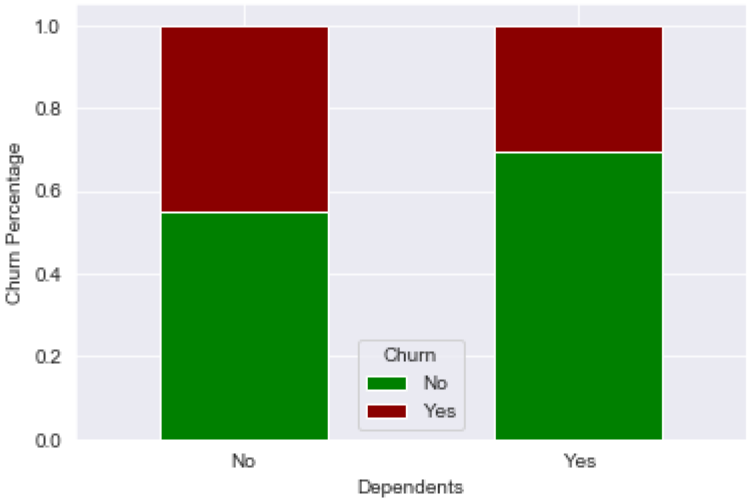


```
In [39]: sns.countplot(df.InternetService, hue = df.SeniorCitizen)
```

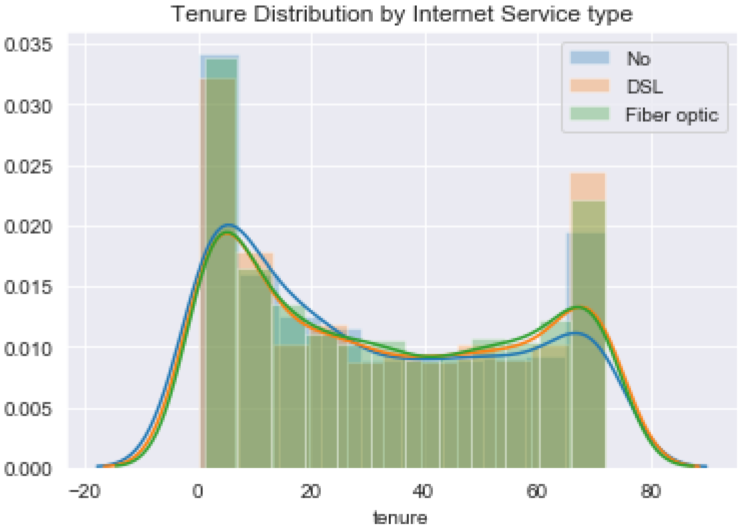
```
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x1ef081c9e80>
```



```
In [42]: stacked_plot(df[df.InternetService == "Fiber optic"], "SeniorCitizen", "Churn")
```



In [40]:
```
sns.distplot(df.tenure[df.InternetService == "No"], hist_kws=dict(alpha=0.3), la
sns.distplot(df.tenure[df.InternetService == "DSL"], hist_kws=dict(alpha=0.3), l
sns.distplot(df.tenure[df.InternetService == "Fiber optic"], hist_kws=dict(alpha
plt.title("Tenure Distribution by Internet Service type")
plt.legend()
plt.show()
```



In [44]:
```
df[df.InternetService == 'No'].head()
```

Out[44]:

| | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | I |
|---|---|---|---|---|---|---|---|---|
| **11** | Male | 0 | No | No | 16 | Yes | No | |
| **16** | Female | 0 | No | No | 52 | Yes | No | |
| **21** | Male | 0 | Yes | No | 12 | Yes | No | |
| **22** | Male | 0 | No | No | 1 | Yes | No | |
| **33** | Male | 0 | No | No | 1 | Yes | No | |

5 rows × 21 columns

In [45]:

```
df[df.InternetService == 'No'].OnlineSecurity.value_counts()
```

Out[45]:

```
No internet service    1526
Name: OnlineSecurity, dtype: int64
```

In [46]:

```
df[df.InternetService == 'No'].OnlineBackup.value_counts()
```

Out[46]:

```
No internet service    1526
Name: OnlineBackup, dtype: int64
```

In [48]:

```
df[df.InternetService == 'No'].DeviceProtection.value_counts()
```

Out[48]:

```
No internet service    1526
Name: DeviceProtection, dtype: int64
```

In [49]:

```
df[df.InternetService == 'No'].TechSupport.value_counts()
```

Out[49]:

```
No internet service    1526
Name: TechSupport, dtype: int64
```

In [50]:

```
df[df.InternetService == 'No'].StreamingMovies.value_counts()
```

Out[50]:

```
No internet service    1526
Name: StreamingMovies, dtype: int64
```

In [52]:

```
df[df.InternetService == 'No'].StreamingTV.value_counts()
```

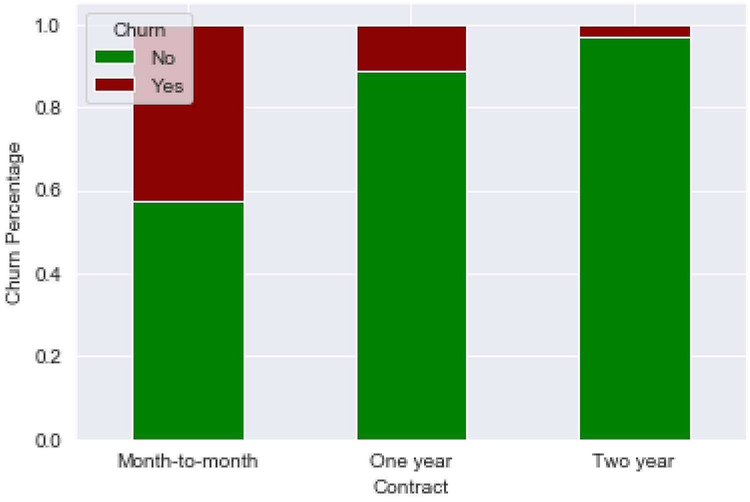
Out[52]:

```
No internet service    1526
Name: StreamingTV, dtype: int64
```

## Contract

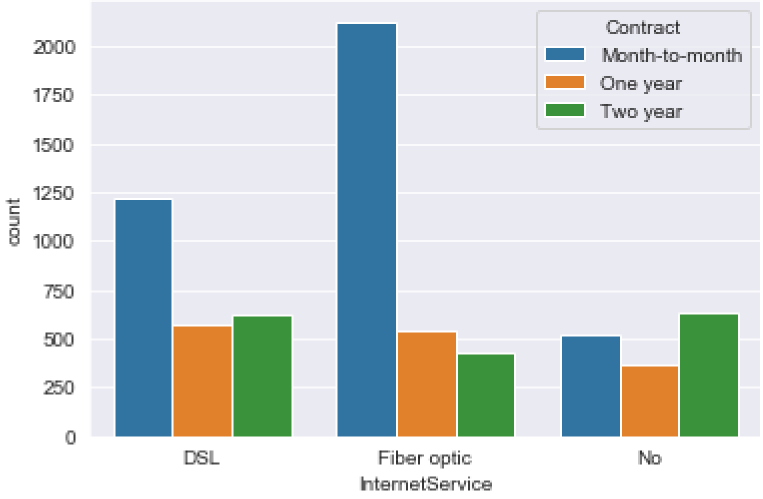
In [21]:

```
stacked_plot(df, "Contract", "Churn")
```



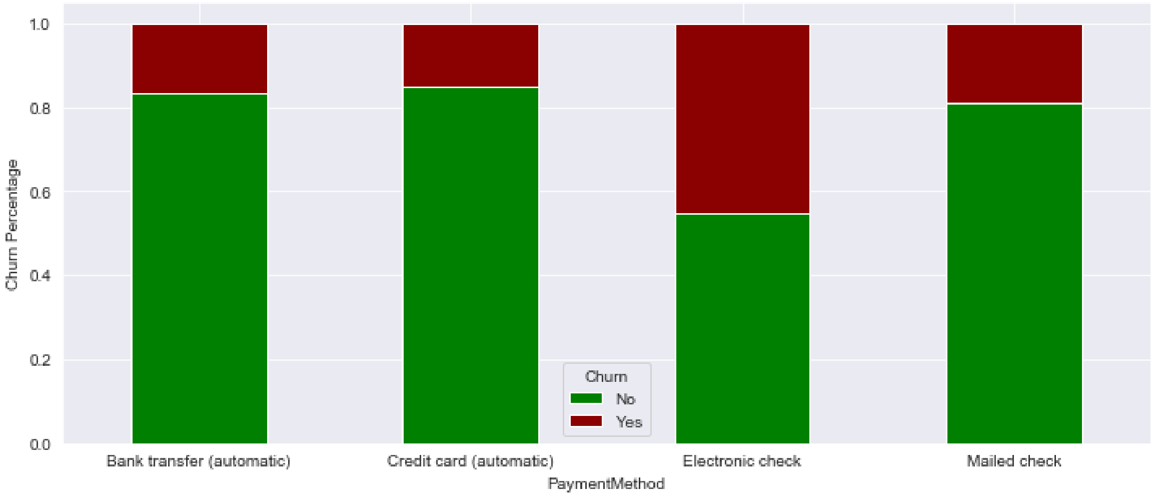
In [41]:
```
sns.countplot(df.InternetService, hue = df.Contract)
```

Out[41]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ef08c2cfd0>
```



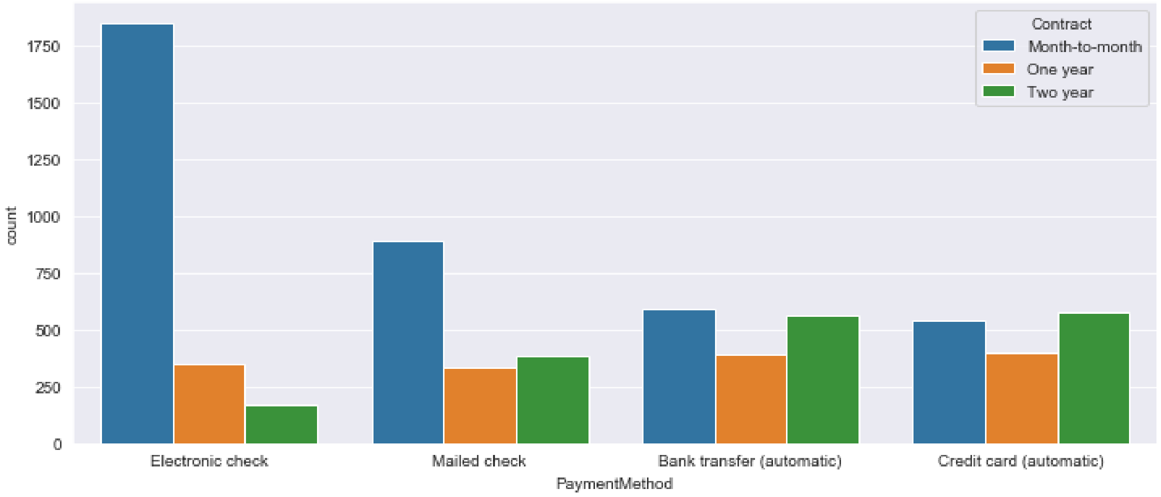
## PaymentMethod

In [44]:
```
group = "PaymentMethod"
target = "Churn"
fig, ax = plt.subplots(figsize = (12,5))
temp_df = (df.groupby([group, target]).size()/df.groupby(group)[target].count())
temp_df.plot(kind='bar', stacked=True, ax = ax, color = ["green", "darkred"])
ax.xaxis.set_tick_params(rotation=0)
ax.set_xlabel(group)
ax.set_ylabel('Churn Percentage');
```



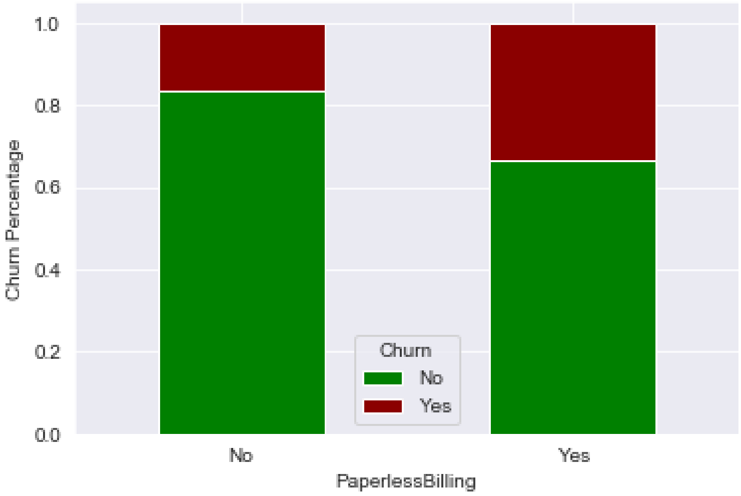
```
In [47]: fig, ax = plt.subplots(figsize = (12,5))
         sns.countplot(df.PaymentMethod, hue = df.Contract, ax = ax)
```

```
Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x1ef081edf60>
```



## PaperlessBilling

```
In [48]: stacked_plot(df, "PaperlessBilling", "Churn")
```



## TotalCharges

In [22]:
```
df.TotalCharges.describe()
```

Out[22]:
```
count      7043
unique     6531
top            
freq         11
Name: TotalCharges, dtype: object
```

In [53]:
```
df['TotalCharges'] = df["TotalCharges"].replace(" ",np.nan)
df['TotalCharges'].isna().sum()
```

Out[53]:
```
11
```

In [54]:
```
df[df["TotalCharges"].isnull()]
```

Out[54]:

| | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines |
|---|---|---|---|---|---|---|---|
| **488** | Female | 0 | Yes | Yes | 0 | No | No phone service |
| **753** | Male | 0 | No | Yes | 0 | Yes | No |
| **936** | Female | 0 | Yes | Yes | 0 | Yes | No |
| **1082** | Male | 0 | Yes | Yes | 0 | Yes | Yes |
| **1340** | Female | 0 | Yes | Yes | 0 | No | No phone service |
| **3331** | Male | 0 | Yes | Yes | 0 | Yes | No |
| **3826** | Male | 0 | Yes | Yes | 0 | Yes | Yes |
| **4380** | Female | 0 | Yes | Yes | 0 | Yes | No |
| **5218** | Male | 0 | Yes | Yes | 0 | Yes | No |
| **6670** | Female | 0 | Yes | Yes | 0 | Yes | Yes |
| **6754** | Male | 0 | No | Yes | 0 | Yes | Yes |

11 rows × 21 columns

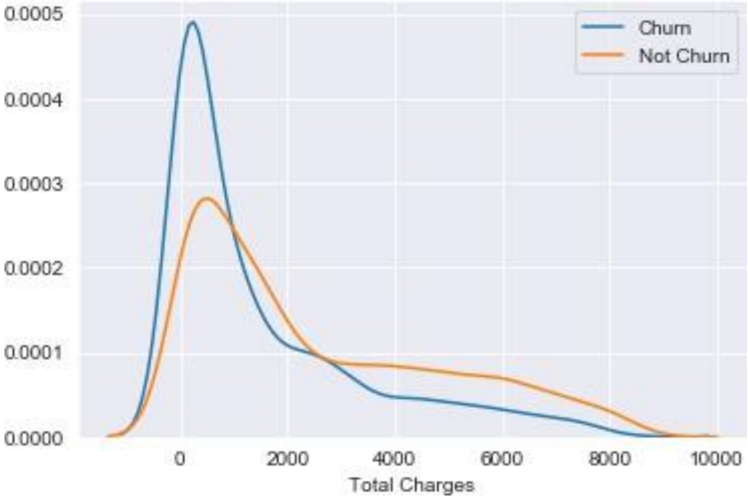
In [55]:
```
df.loc[df["TotalCharges"].isnull(), 'TotalCharges'] = 0
df.isnull().any().any()
```

Out[55]:
```
False
```

In [56]:
```
df['TotalCharges'] = df["TotalCharges"].astype(float)

Churn = df[df.Churn=="Yes"]
Not_Churn = df[df.Churn=="No"]
```

In [57]:
```
fig, ax = plt.subplots()
sns.kdeplot(Churn["TotalCharges"],label = "Churn", ax= ax)
sns.kdeplot(Not_Churn["TotalCharges"], label = "Not Churn", ax=ax)
ax.set_xlabel("Total Charges");
```



## Monthly Charges

In [18]:
```
df.MonthlyCharges.describe()
```

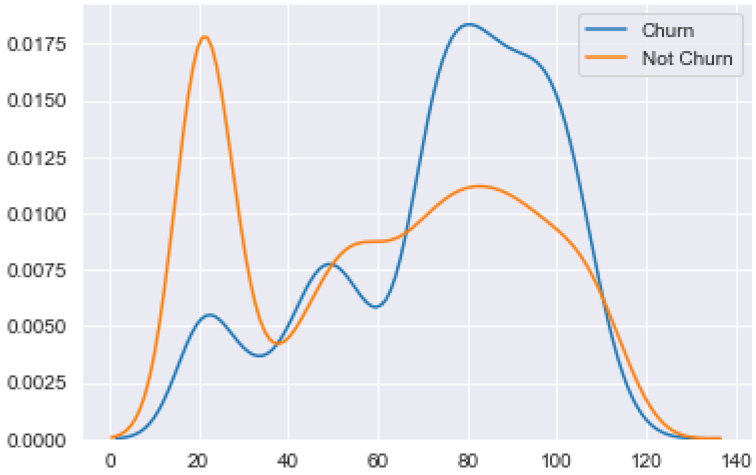
Out[18]:
```
count    7043.000000
mean       64.761692
std        30.090047
min        18.250000
25%        35.500000
50%        70.350000
75%        89.850000
max       118.750000
Name: MonthlyCharges, dtype: float64
```

In [19]:
```
df.MonthlyCharges.isna().sum()
```

Out[19]:
```
0
```

In [20]:
```
sns.kdeplot(Churn["MonthlyCharges"], label = "Churn")
sns.kdeplot(Not_Churn["MonthlyCharges"], label = "Not Churn")
```

Out[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x16f326cfcc0>
```



In [28]:

```
np.corrcoef(df.TotalCharges,  df.MonthlyCharges*df.tenure)
```

Out[28]:

```
array([[1.        , 0.99956055],
       [0.99956055, 1.        ]])
```

# Fucntion to prepare data for model building based on EDA

In [62]:

```
def datapreparation(filepath):
    
    df = pd.read_csv(filepath)
    df.drop(["customerID"], inplace = True, axis = 1)
    
    df.TotalCharges = df.TotalCharges.replace(" ",np.nan)
    df.TotalCharges.fillna(0, inplace = True)
    df.TotalCharges = df.TotalCharges.astype(float)
    
    cols1 = ['Partner', 'Dependents', 'PaperlessBilling', 'Churn', 'PhoneService
    for col in cols1:
        df[col] = df[col].apply(lambda x: 0 if x == "No" else 1)
    
    df.gender = df.gender.apply(lambda x: 0 if x == "Male" else 1)
    df.MultipleLines = df.MultipleLines.map({'No phone service': 0, 'No': 0, 'Ye
    
    cols2 = ['OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport'
    for col in cols2:
        df[col] = df[col].map({'No internet service': 0, 'No': 0, 'Yes': 1})
    
    df = pd.get_dummies(df, columns=['InternetService', 'Contract', 'PaymentMeth
    
    return df
```