

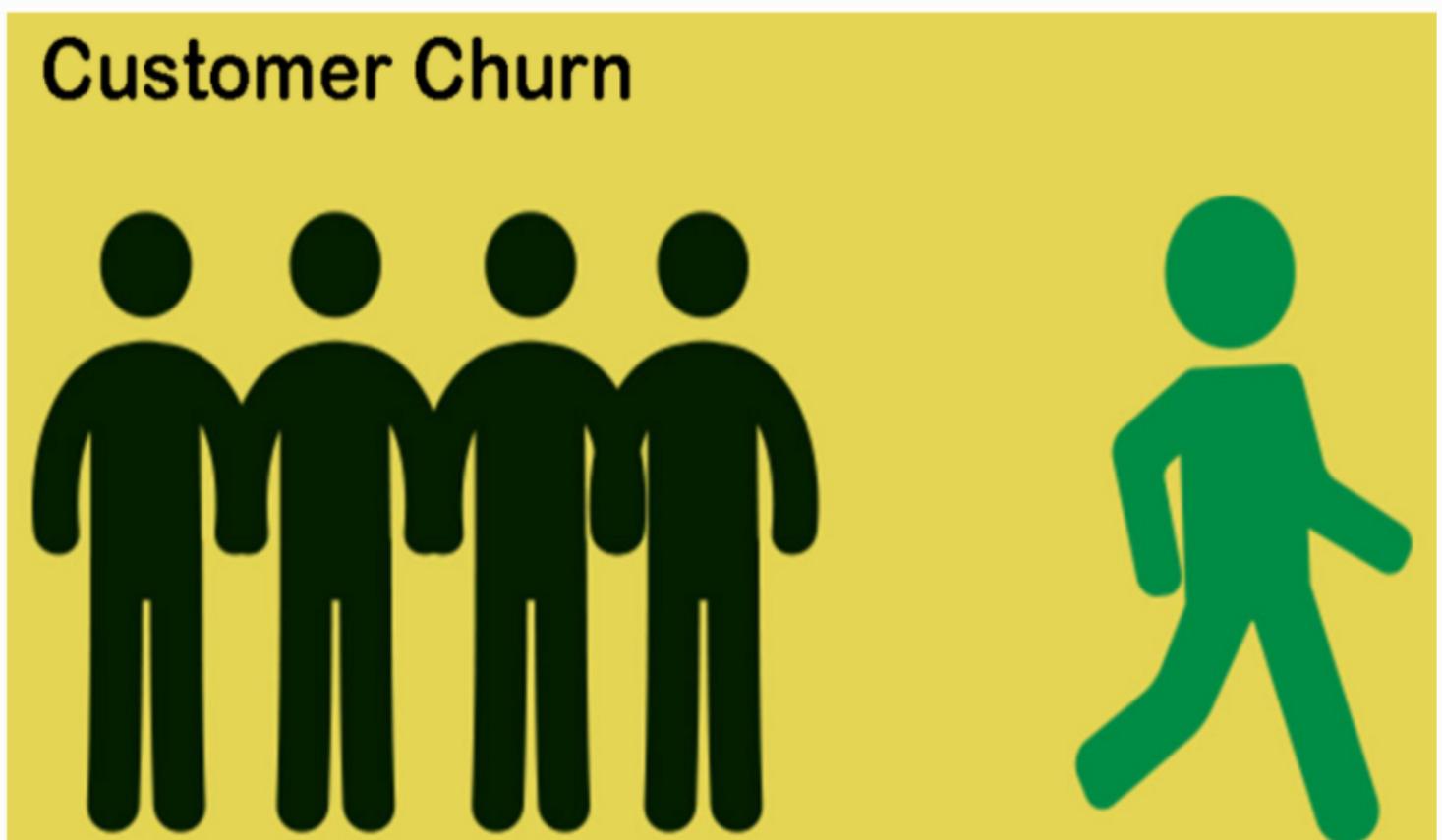
CUSTOMER CHURN PREDICTION

Name : S.Sheeba

Naa mudalvan id : au920821104044

Phase 4 Project : developement part 2

Phase 4



DATA SOURCE

Churn prediction relies on data from various sources, including seniorcitizen, gender , techsupport, phoneservice, multiple lines, internet service and customer feedback.

DATA LINK

A	B	C	D	E	F	G	H	I	J	K	L	M	N
customerID	gender	SeniorCitizen	Partner	Dependents	Tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV
7590-VHV	Female	0	Yes	No	1	No	No phone	DSL	No	Yes	No	No	No
5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No
3668-QPY	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No
7795-CFO	Male	0	No	No	45	No	No phone	DSL	Yes	No	Yes	Yes	No
9237-HQI	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No
9305-CDS	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No	Yes
1452-KIO	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No	Yes
6713-OKO	Female	0	No	No	10	No	No phone	DSL	Yes	No	No	No	No

<https://www.kaggle.com/datasets/blastchar/telco-customer-churn>

VISUALIZATION

Visualization refers to the process of creating graphical representations of data or information to make it more understandable and accessible. These visual representations can include charts, graphs, maps, diagrams, or other images that help convey complex information in a clear and intuitive manner.

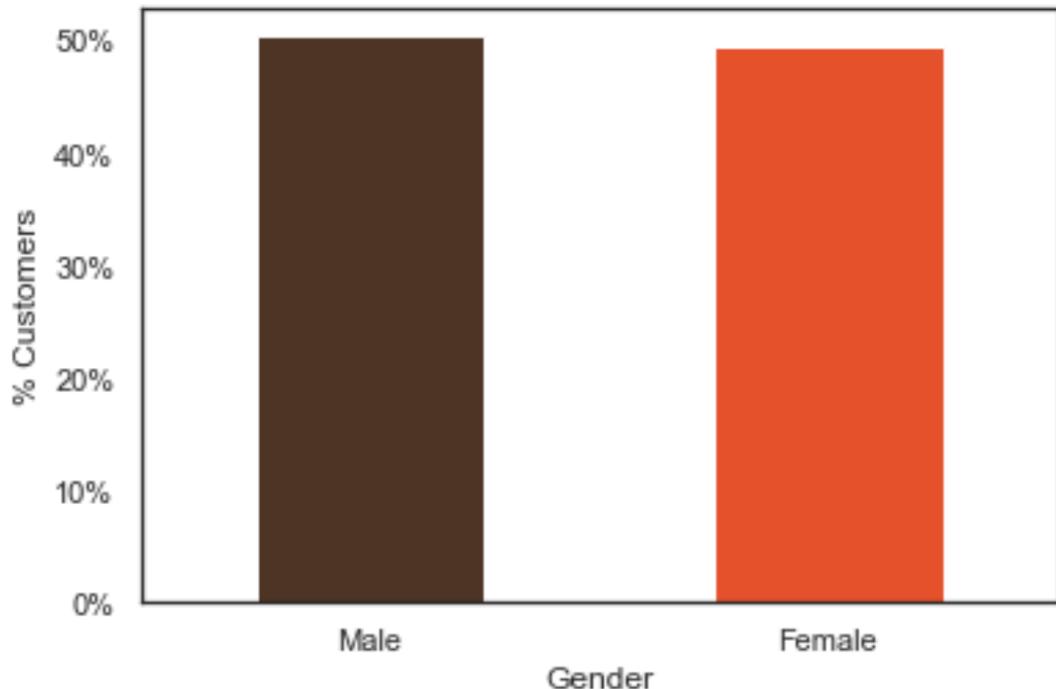
Visualization is commonly used in various fields, including data analysis, science, business, and education, to aid in understanding patterns, trends, relationships, and insights within data or concepts. It can be a powerful tool for conveying information and supporting decision-making.

PROGRAM

```
42 # Data exploration
43
44 colors = ['#4D3425', '#E4512B']
45 ax = (df['gender'].value_counts() * 100.0 / len(df)).plot(kind='bar',
46                         stacked=True, rot=0, color=colors)
47
48 ax.yaxis.set_major_formatter(mtick.PercentFormatter())
49 ax.set_ylabel('% Customers')
50 ax.set_xlabel('Gender')
51 ax.set_ylabel('% Customers')
52 ax.set_title('Gender Distribution')
53
```

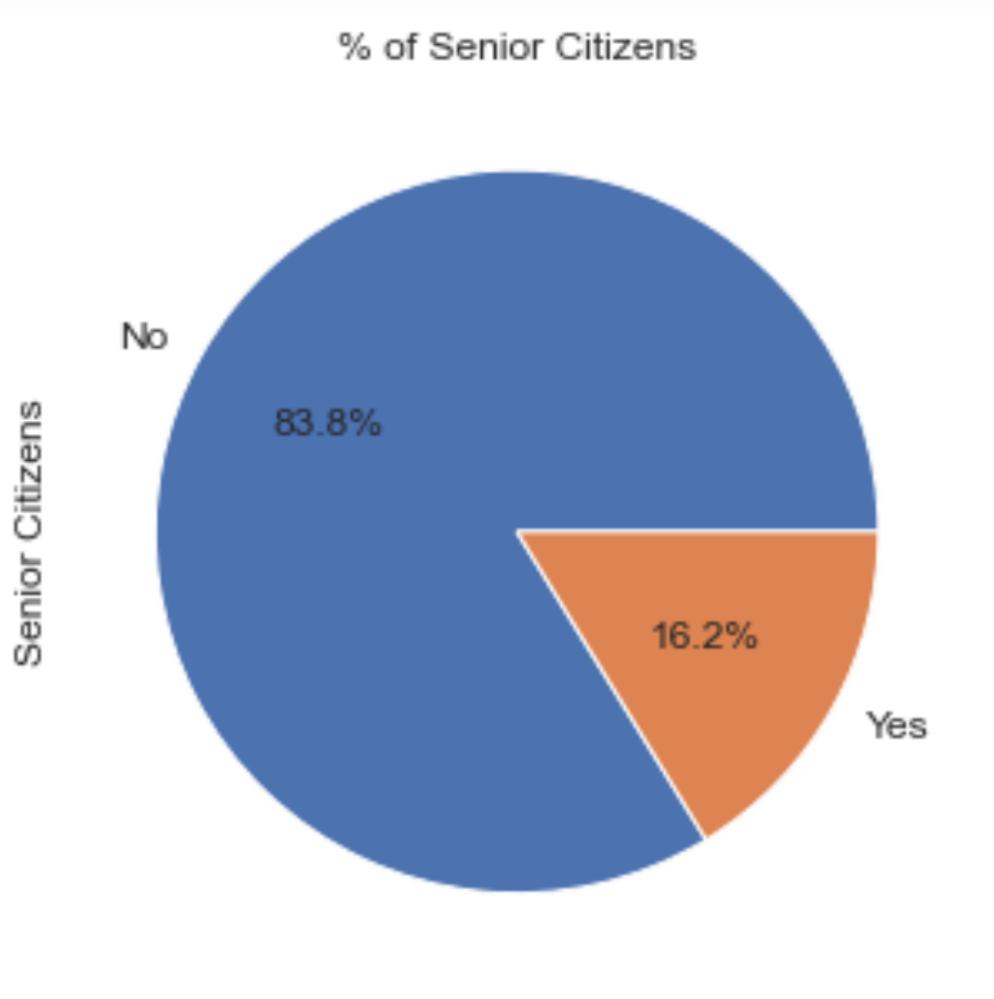
Output:

Gender Distribution



```
75 # Senior citizen  
76  
77 ax = (df['SeniorCitizen'].value_counts() * 100.0 / len(df)).plot.pie(autopct='%.1f%%',  
78 labels=['No', 'Yes'], figsize=(5, 5), fontsize=12)  
79  
80  
81 ax.yaxis.set_major_formatter(mtick.PercentFormatter())  
82 ax.set_ylabel('Senior Citizens', fontsize=12)  
83 ax.set_title('% of Senior Citizens', fontsize=12)  
84
```

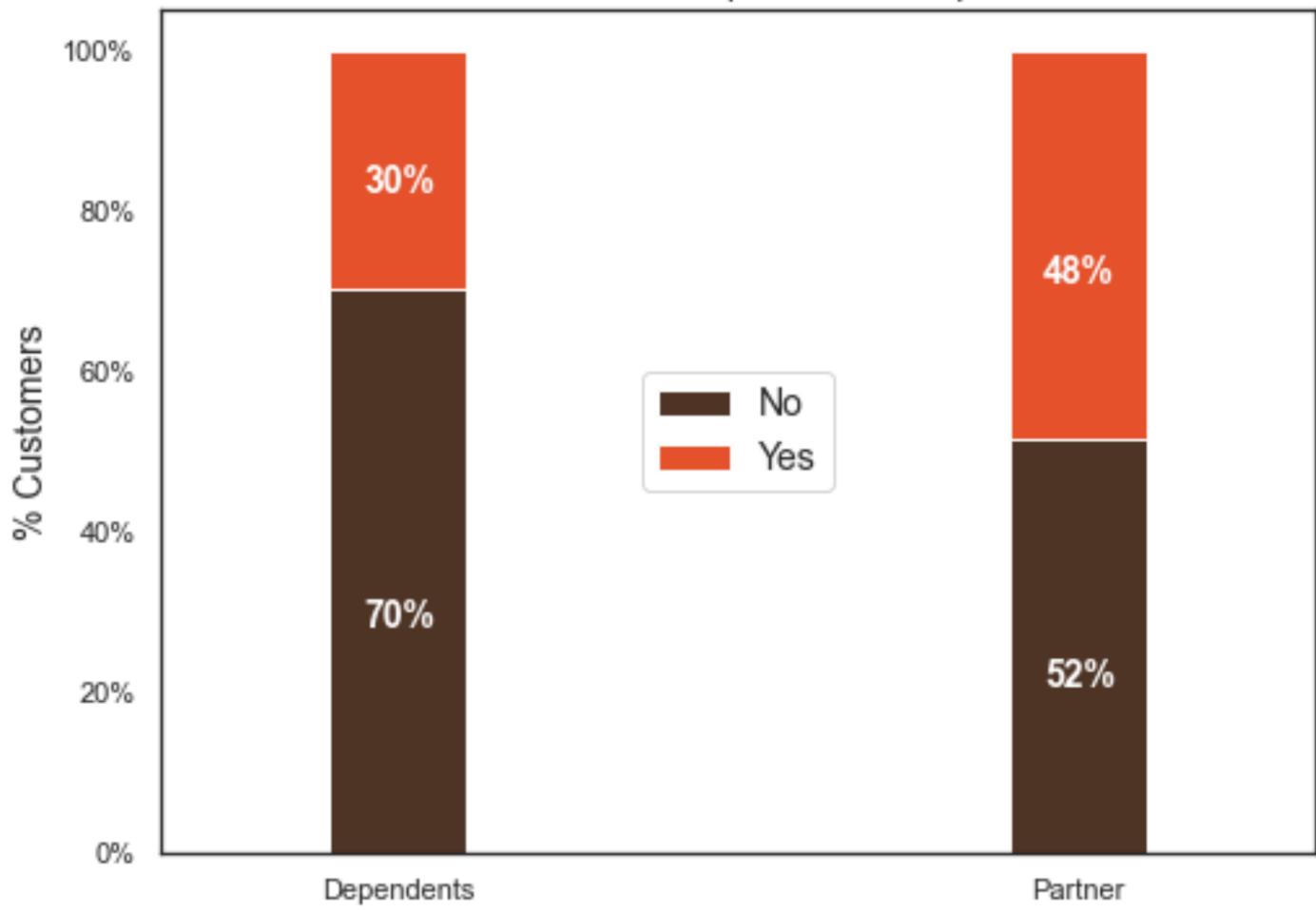
Output:



```
85 # Partner and dependent status
86
87 df2 = pd.melt(df, id_vars=['customerID'], value_vars=['Dependents', 'Partner'])
88 df3 = df2.groupby(['variable', 'value']).count().unstack()
89 df3 = df3 * 100 / len(df)
90 colors = ['#4D3425', '#E4512B']
91 ax = df3.loc[:, 'customerID'].plot.bar(stacked=True, color=colors, figsize=(8, 6), rot=0, width=0.2)
92
93 ax.yaxis.set_major_formatter(mtick.PercentFormatter())
94 ax.set_ylabel('% Customers', size=14)
95 ax.set_xlabel('')
96 ax.set_title('% Customers with dependents and partners', size=14)
97 ax.legend(loc='center', prop={'size': 14})
98
99 for p in ax.patches:
100     width, height = p.get_width(), p.get_height()
101     x, y = p.get_xy()
102     ax.annotate('{:.0f}%'.format(height), (x + 0.25 * width, y + 0.4 * height),
103                 color='white',
104                 weight='bold',
105                 size=14)
```

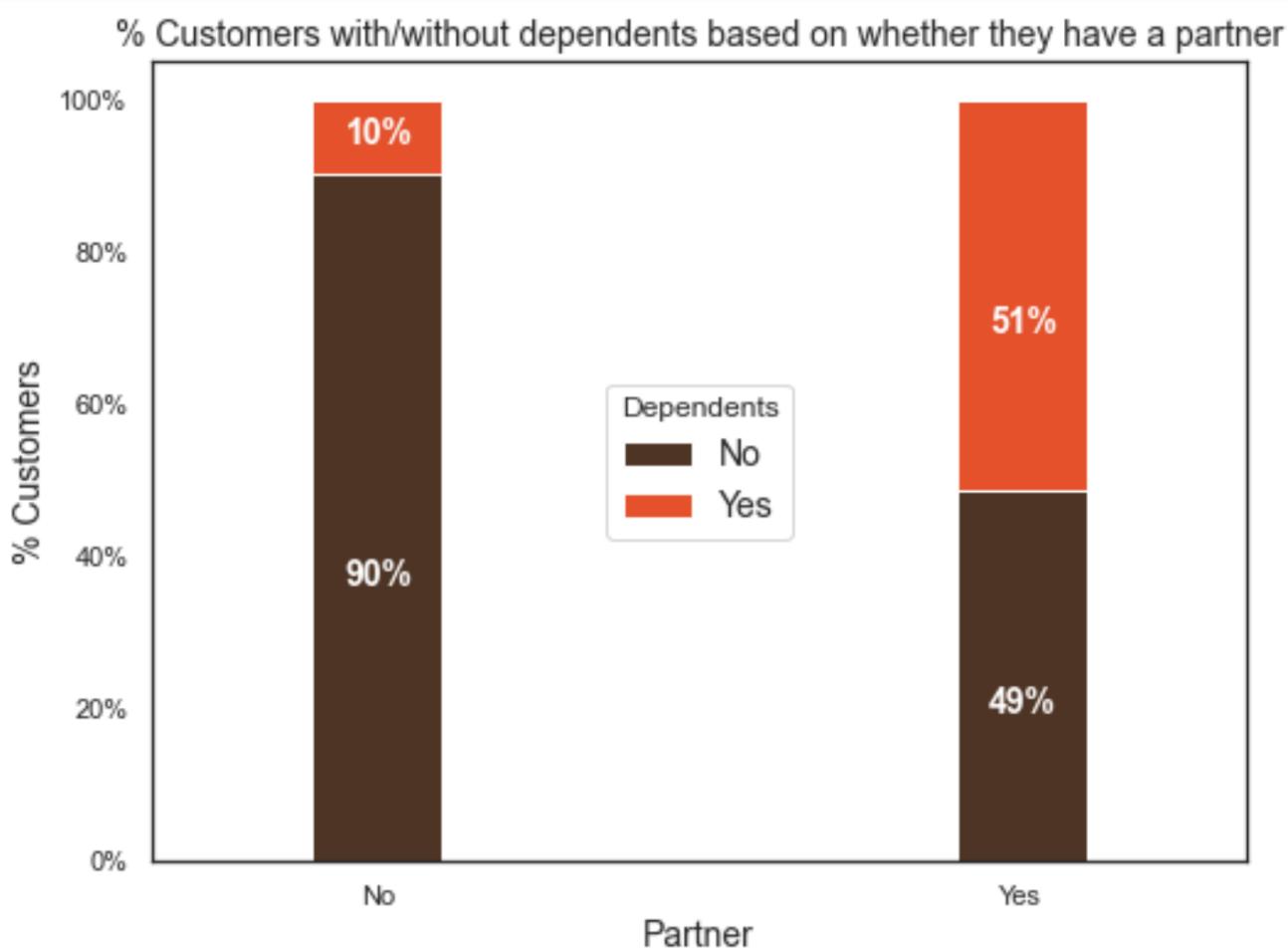
Output:

% Customers with dependents and partners



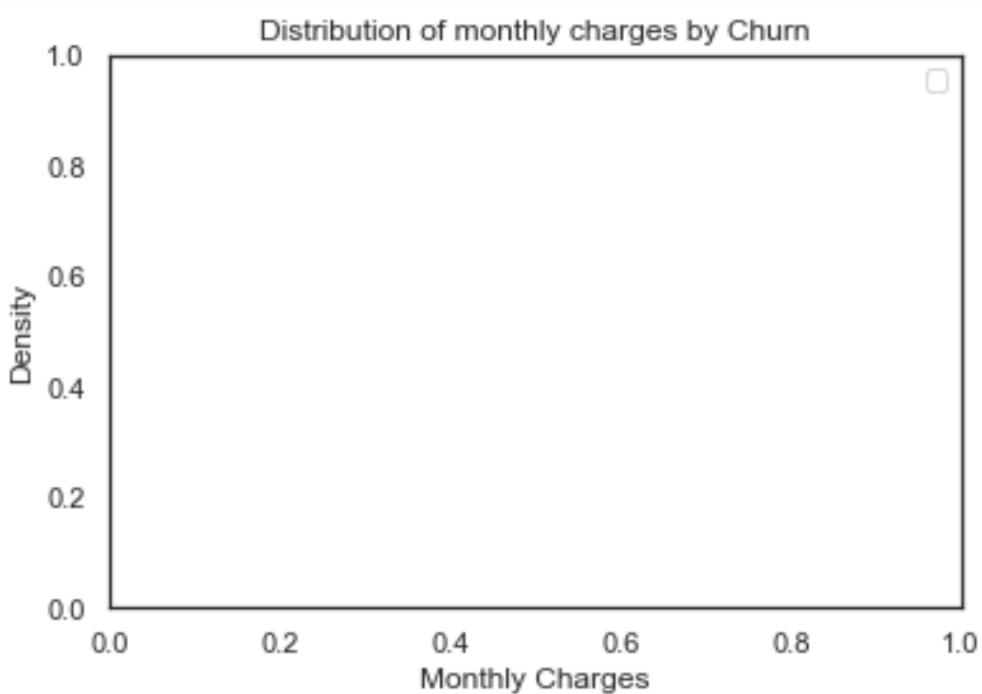
```
109 # Customers with or without dependents
110
111 colors = ['#4D3425', '#E4512B']
112 partner_dependents = df.groupby(['Partner', 'Dependents']).size().unstack()
113
114 ax = (partner_dependents.T * 100.0 / partner_dependents.T.sum()).T.plot(kind='bar',
115                                         width=0.2,
116                                         stacked=True,
117                                         rot=0,
118                                         figsize=(8, 6),
119                                         color=colors)
120
121 ax.yaxis.set_major_formatter(mtick.PercentFormatter())
122
123 ax.legend(loc='center', prop={'size': 14}, title='Dependents', fontsize=14)
124 ax.set_ylabel('% Customers', size=14)
125 ax.set_title('% Customers with/without dependents based on whether they have a partner', size=14)
126 ax.xaxis.label.set_size(14)
127
```

Output:



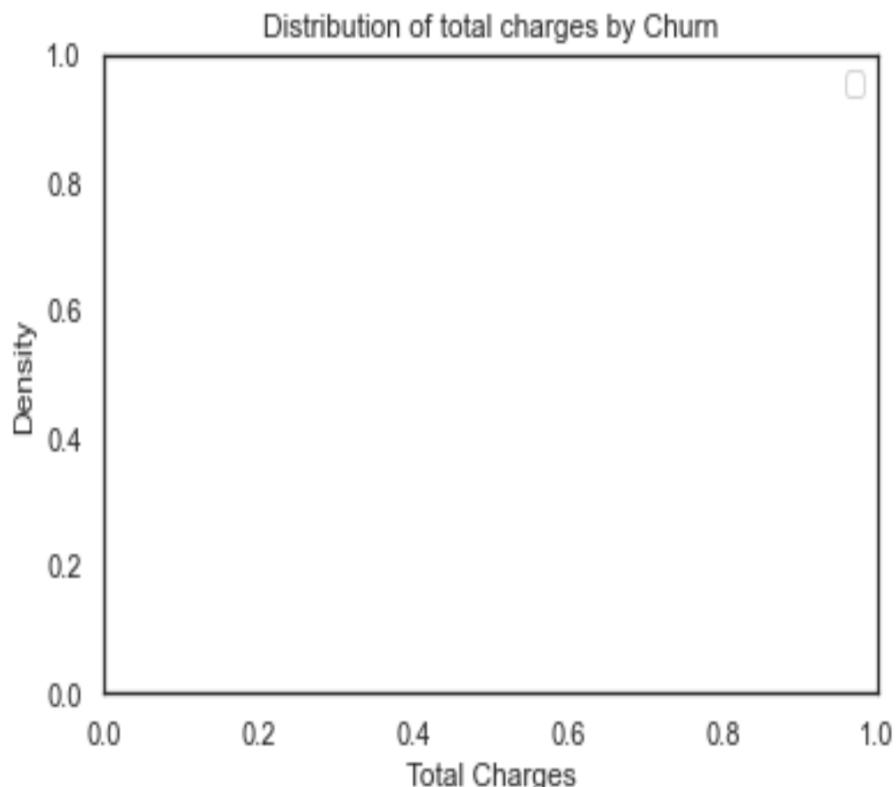
```
281  
282 # Churn by Monthly Charges  
283  
284 ax = sns.kdeplot(df.MonthlyCharges[(df[ "Churn" ] == 0)],  
285                     color="Red", shade=True)  
286 ax = sns.kdeplot(df.MonthlyCharges[(df[ "Churn" ] == 1)],  
287                     ax=ax, color="Blue", shade=True)  
288 ax.legend([ "Not Churn", "Churn"], loc='upper right')  
289 ax.set_ylabel('Density')  
290 ax.set_xlabel('Monthly Charges')  
291 ax.set_title('Distribution of monthly charges by Churn')  
292
```

Output:



```
293     # Churn by Total Charges
294
295     ax = sns.kdeplot(df.TotalCharges[(df[ "Churn" ] == 0)],
296                         color="Red", shade=True)
297     ax = sns.kdeplot(df.TotalCharges[(df[ "Churn" ] == 1)],
298                         ax=ax, color="Blue", shade=True)
299     ax.legend([ "Not Churn", "Churn"], loc='upper right')
300     ax.set_ylabel('Density')
301     ax.set_xlabel('Total Charges')
302     ax.set_title('Distribution of total charges by Churn')
303
```

Output:



```

136 # Customer info
137
138 ax = sns.distplot(df['tenure'], hist=True, kde=False,
139                     bins=int(180 / 5), color='darkblue', |
140                     hist_kws={'edgecolor': 'black'},
141                     kde_kws={'linewidth': 4})
142 ax.set_ylabel('# of Customers')
143 ax.set_xlabel('Tenure (months)')
144 ax.set_title('# of Customers by their tenure')
145 ax = df['Contract'].value_counts().plot(kind='bar', rot=0, width=0.3)
146 ax.set_ylabel('# of Customers')
147 ax.set_title('# of Customers by Contract Type')
148 fig, (ax1, ax2, ax3) = plt.subplots(nrows=1, ncols=3, sharey=True, figsize=(20, 6))
149 ax = sns.distplot(df[df['Contract'] == 'Month-to-month']['tenure'],
150                     hist=True, kde=False,
151                     bins=int(180 / 5), color='turquoise',
152                     hist_kws={'edgecolor': 'black'},
153                     kde_kws={'linewidth': 4},
154                     ax=ax1)
155 ax.set_ylabel('# of Customers')
156 ax.set_xlabel('Tenure (months)')
157 ax.set_title('Month to Month Contract')
158
159
160 ax = sns.distplot(df[df['Contract'] == 'One year']['tenure'],
161                     hist=True, kde=False,
162                     bins=int(180 / 5), color='steelblue',
163                     hist_kws={'edgecolor': 'black'},
164                     kde_kws={'linewidth': 4},
165                     ax=ax2)
166 ax.set_xlabel('Tenure (months)', size=14)
167 ax.set_title('One Year Contract', size=14)
168
169 ax = sns.distplot(df[df['Contract'] == 'Two year']['tenure'],
170                     hist=True, kde=False,
171                     bins=int(180 / 5), color='darkblue',
172                     hist_kws={'edgecolor': 'black'},
173                     kde_kws={'linewidth': 4},
174                     ax=ax3)
175 ax.set_xlabel('Tenure (months)')
176 ax.set_title('Two Year Contract')
177
178 services = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
179             'OnLineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']
180
181 fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 12))
182 for i, item in enumerate(services):
183     if i < 3:
184         ax = df[item].value_counts().plot(kind='bar', ax=axes[i, 0], rot=0)
185
186     elif i >= 3 and i < 6:
187         ax = df[item].value_counts().plot(kind='bar', ax=axes[i - 3, 1], rot=0)
188
189     elif i < 9:
190         ax = df[item].value_counts().plot(kind='bar', ax=axes[i - 6, 2], rot=0)
191     ax.set_title(item)
192
193 df[['MonthlyCharges', 'TotalCharges']].plot.scatter(x='MonthlyCharges', y='TotalCharges')
194

```

Output:

