Untitled4 (1)

April 1, 2020

[3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.offline as py
import plotly.graph_objs as go
```

[6]:
```
f = pd.read_csv('C:/R Analysis/Churn_Telecom.csv')
```

[7]:
```
f.head()
```

[7]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No	One year	No	Mailed check	56.95	1889.5	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

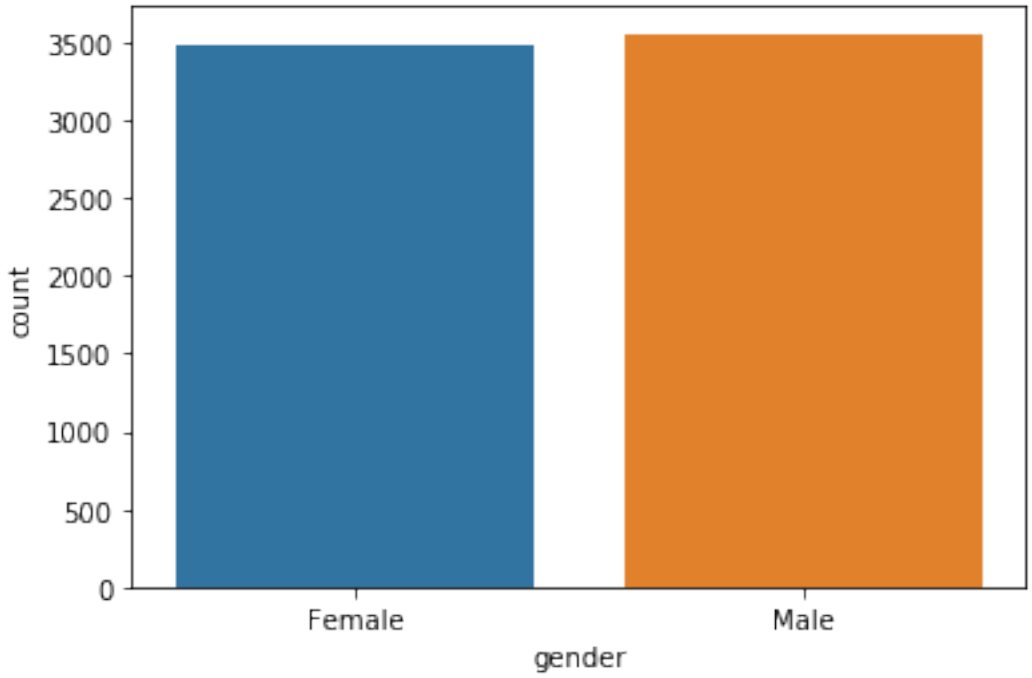
```
[5 rows x 21 columns]
```

[8]:
```
f.columns
```

[8]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

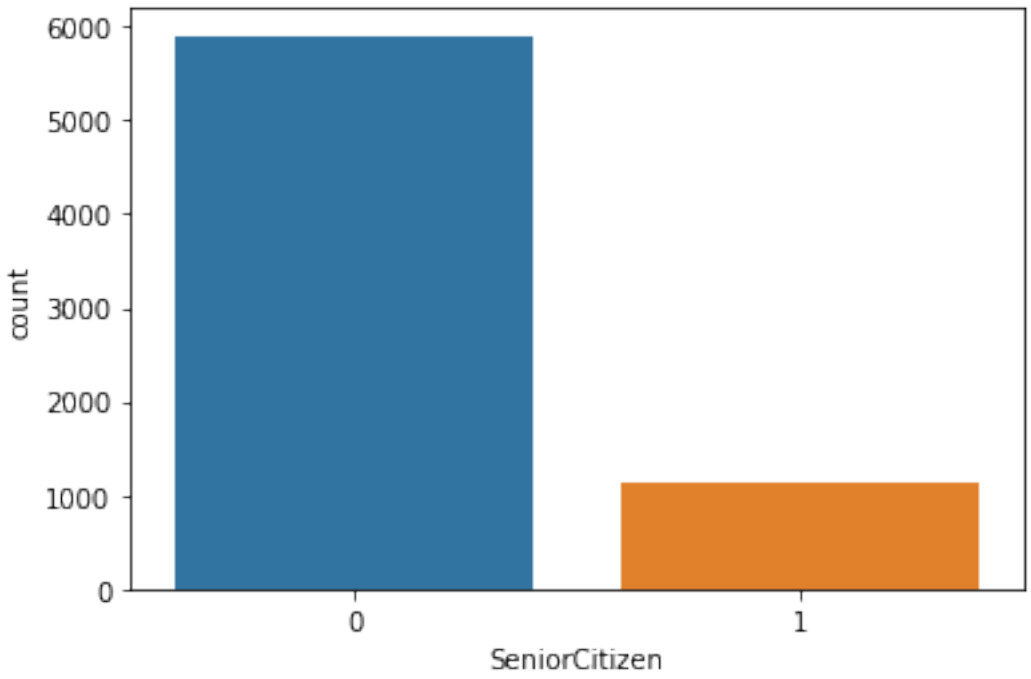
[9]:
```
sns.countplot(f.gender)
```

[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x210944b9668>
```



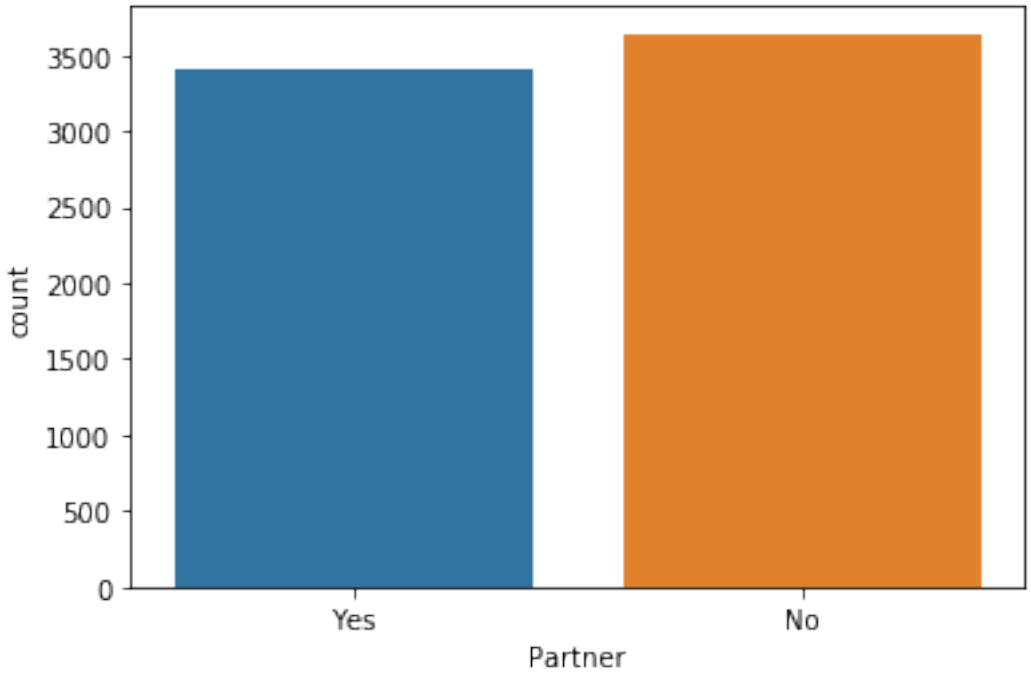
[10]:
```
sns.countplot(f.SeniorCitizen)
```

[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2109480dcc0>
```



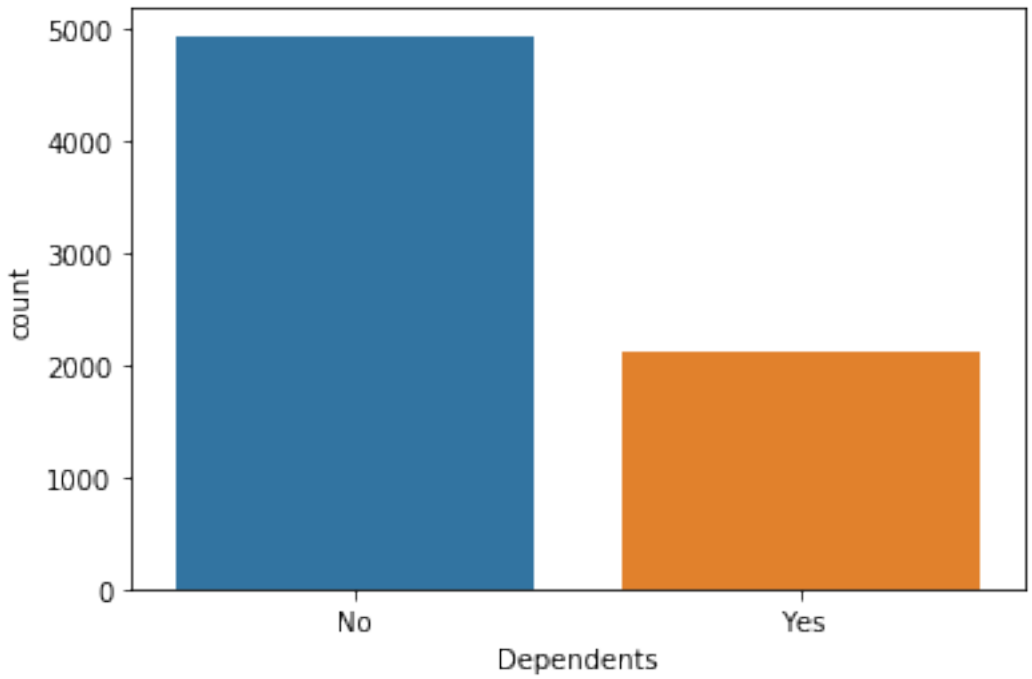
[11]:
```
sns.countplot(f.Partner)
```

[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21094853208>
```



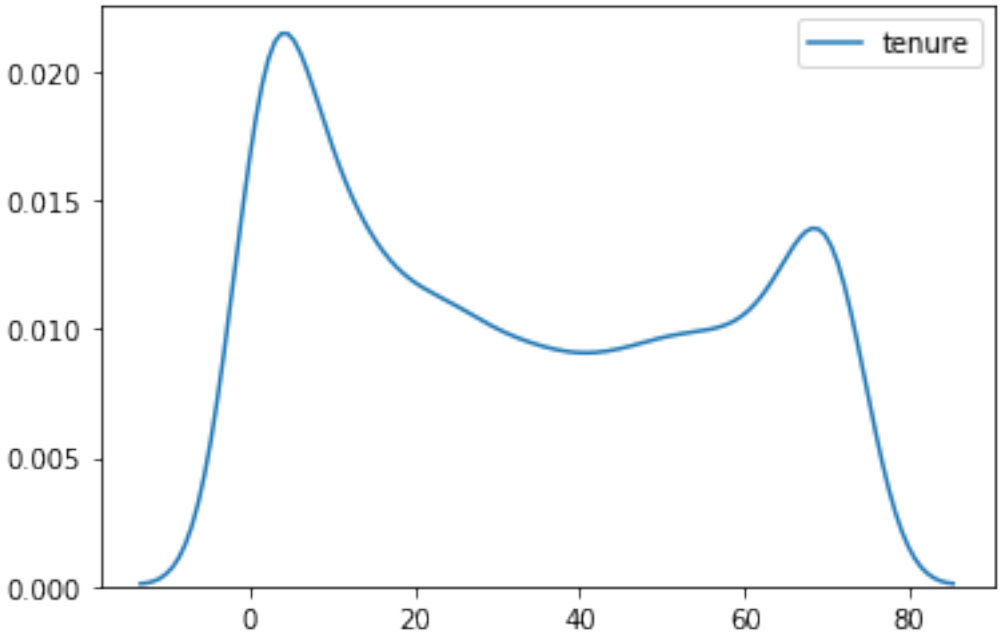
[12]:
```
sns.countplot(f.Dependents)
```

[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2109489eb00>
```



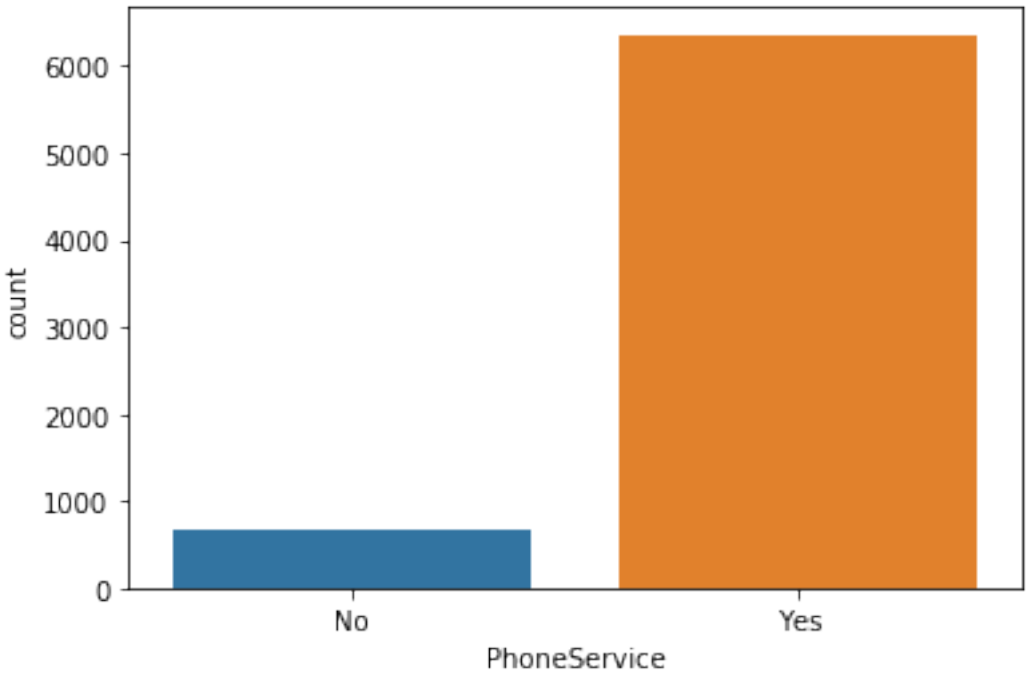
[13]:
```
sns.kdeplot(f.tenure)
```

[13]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21094918978>
```



[14]:
```
sns.countplot(f.PhoneService)
```

[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x210948b9198>
```



[15]:
```
print('MultipleLines :', set(f.MultipleLines))
print('InternetService :', set(f.InternetService))
print('OnlineSecurity :', set(f.OnlineSecurity))
print('DeviceProtection :', set(f.DeviceProtection))
print('TechSupport :', set(f.TechSupport))
print('StreamingTV :', set(f.StreamingTV))
print('StreamingMovies :', set(f.StreamingMovies))
print('Contract :', set(f.Contract))
print('PaperlessBilling', set(f.PaperlessBilling))
print('PaymentMethod', set(f.PaymentMethod))
```

```
MultipleLines : {'No phone service', 'No', 'Yes'}
InternetService : {'Fiber optic', 'DSL', 'No'}
OnlineSecurity : {'No internet service', 'No', 'Yes'}
DeviceProtection : {'No internet service', 'No', 'Yes'}
TechSupport : {'No internet service', 'No', 'Yes'}
StreamingTV : {'No internet service', 'No', 'Yes'}
StreamingMovies : {'No internet service', 'No', 'Yes'}
Contract : {'One year', 'Two year', 'Month-to-month'}
```

```
PaperlessBilling {'No', 'Yes'}
PaymentMethod {'Credit card (automatic)', 'Bank transfer (automatic)',
'Electronic check', 'Mailed check'}
```

[16]:
```
# label
lab = f["Churn"].value_counts().keys().tolist()
print(lab)

# Values
val = f["Churn"].value_counts().values.tolist()
print(val)

# colors
cmap = plt.get_cmap('Spectral')
colors = [cmap(i) for i in np.linspace(0, 1, 8)]

trace = go.Pie(labels = lab,
               values = val,
               marker = dict(colors = colors,
                             line = dict(color = "white",
                                         width = 1.3)
                            ),
               rotation = 45,
              hoverinfo = "label+value+text",
              hole = .5
              )

layout = go.Layout(dict(title = "Customer attrition in data",
                        plot_bgcolor = "rgb(243, 243, 243)",
                        paper_bgcolor = "rgb(243, 243, 243)",
                        )
                   )

data = [trace]
print(data)

fig = go.Figure(data = data, layout = layout)
py.iplot(fig)
```

```
['No', 'Yes']
[5174, 1869]
[Pie({
    'hole': 0.5,
    'hoverinfo': 'label+value+text',
    'labels': [No, Yes],
    'marker': {'colors': [[0.6196078431372549, 0.00392156862745098,
                           0.25882352941176473, 1.0], [0.8853517877739331,
```

```
                        0.3190311418685121, 0.29042675893886966, 1.0],
                        [0.9873125720876587, 0.6473663975394078,
                        0.3642445213379469, 1.0], [0.9971549404075356,
                        0.9118031526336025, 0.6010765090349866, 1.0],
                        [0.9288735101883892, 0.9715494040753557,
                        0.6380622837370243, 1.0], [0.6334486735870821,
                        0.8521337946943485, 0.6436755094194541, 1.0],
                        [0.2800461361014994, 0.6269896193771626,
                        0.7024221453287197, 1.0], [0.3686274509803922,
                        0.30980392156862746, 0.6352941176470588, 1.0]],
              'line': {'color': 'white', 'width': 1.3}},
    'rotation': 45,
    'values': [5174, 1869]
})]
```

[17]:
```
churn     = f[f["Churn"] == "Yes"]
not_churn = f[f["Churn"] == "No"]

Id_col     = ['customerID']
target_col = ["Churn"]
cat_cols   = f.nunique()[f.nunique() < 6].keys().tolist()
cat_cols   = [x for x in cat_cols if x not in target_col]
num_cols   = [x for x in f.columns if x not in cat_cols + target_col + Id_col]

def plot_pie(column) :

    trace1 = go.Pie(values  = churn[column].value_counts().values.tolist(),
                    labels  = churn[column].value_counts().keys().tolist(),
                    hoverinfo = "label+percent+name",
                    domain  = dict(x = [0,.48]),
                    name    = "Churn Customers",
                    marker  = dict(line = dict(width = 2,
                                               color = "rgb(243,243,243)")
                                   ),
                    hole    = .6
                   )
    trace2 = go.Pie(values  = not_churn[column].value_counts().values.tolist(),
                    labels  = not_churn[column].value_counts().keys().tolist(),
                    hoverinfo = "label+percent+name",
                    marker  = dict(line = dict(width = 2,
                                               color = "rgb(243,243,243)")
                                   ),
                    domain  = dict(x = [.52,1]),
                    hole    = .6,
                    name    = "Non churn customers"
                   )
```

```
    layout = go.Layout(dict(title = column + " distribution in customer␣
 ↪attrition ",
                            plot_bgcolor  = "rgb(243,243,243)",
                            paper_bgcolor = "rgb(243,243,243)",
                            annotations = [dict(text = "churn customers",
                                                font = dict(size = 13),
                                                showarrow = False,
                                                x = .15, y = .5),
                                           dict(text = "Non churn customers",
                                                font = dict(size = 13),
                                                showarrow = False,
                                                x = .88,y = .5
                                               )
                                          ]
                           )
                      )
    data = [trace1,trace2]
    fig  = go.Figure(data = data,layout = layout)
    py.iplot(fig)

def histogram(column) :
    trace1 = go.Histogram(x  = churn[column],
                          histnorm= "percent",
                          name = "Churn Customers",
                          marker = dict(line = dict(width = .5,
                                                    color = "black"
                                                    )
                                        ),
                          opacity = .9
                          )

    trace2 = go.Histogram(x  = not_churn[column],
                          histnorm = "percent",
                          name = "Non churn customers",
                          marker = dict(line = dict(width = .5,
                                                    color = "black"
                                                   )
                                        ),
                          opacity = .9
                          )

    data = [trace1,trace2]
    layout = go.Layout(dict(title =column + " distribution in customer␣
 ↪attrition ",
                            plot_bgcolor  = "rgb(243,243,243)",
                            paper_bgcolor = "rgb(243,243,243)",
                            xaxis = dict(gridcolor = 'rgb(255, 255, 255)',
```

```
                                            title = column,
                                            zerolinewidth=1,
                                            ticklen=5,
                                            gridwidth=2
                                           ),
                            yaxis = dict(gridcolor = 'rgb(255, 255, 255)',
                                            title = "percent",
                                            zerolinewidth=1,
                                            ticklen=5,
                                            gridwidth=2
                                           ),
                           )
                       )
    fig  = go.Figure(data=data,layout=layout)

    py.iplot(fig)

for i in cat_cols :
    plot_pie(i)

for i in num_cols :
    histogram(i)
```

[18]:
```
churn     = f[f["Churn"] == "Yes"]
not_churn = f[f["Churn"] == "No"]

Id_col     = ['customerID']
target_col = ["Churn"]
cat_cols   = f.nunique()[f.nunique() < 6].keys().tolist()
print(cat_cols)
cat_cols   = [x for x in cat_cols if x not in target_col]
print(cat_cols)
num_cols   = [x for x in f.columns if x not in cat_cols + target_col + Id_col]
print(num_cols)
```

```
['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService',
'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup',
'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
'PaperlessBilling', 'PaymentMethod', 'Churn']
['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService',
'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup',
'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
'PaperlessBilling', 'PaymentMethod']
['tenure', 'MonthlyCharges', 'TotalCharges']
```

[19]:
```
churn     = f[f["Churn"] == "Yes"]
not_churn = f[f["Churn"] == "No"]
```

```
trace1 = go.Pie(values  = churn.SeniorCitizen.value_counts().values.tolist(),
                labels  = churn.SeniorCitizen.value_counts().keys().
 ↪tolist(),
                hoverinfo = "label+percent+name",
                domain  = dict(x = [0,.48]),
                name    = "Churn Customers",
                marker  = dict(line = dict(width = 2,
                                           color = "rgb(243,243,243)")
                               ),
                hole    = .7
               )

trace2 = go.Pie(values  = not_churn.SeniorCitizen.value_counts().values.
 ↪tolist(),
                labels  = not_churn.SeniorCitizen.value_counts().keys().
 ↪tolist(),
                hoverinfo = "label+percent+name",
                marker  = dict(line = dict(width = 2,
                                           color = "rgb(243,243,243)")
                               ),
                domain  = dict(x = [.52,1]),
                hole    = .7,
                name    = "Non churn customers"
               )

layout = go.Layout(dict(title = "Gender distribution in customer attrition ",
                        plot_bgcolor  = "rgb(243,243,243)",
                        paper_bgcolor = "rgb(243,243,243)",
                        annotations = [dict(text = "churn customers",
                                            font = dict(size = 13),
                                            showarrow = False,
                                            x = .15, y = .5),
                                       dict(text = "Non churn customers",
                                            font = dict(size = 13),
                                            showarrow = False,
                                            x = .88,y = .5
                                           )
                                      ]
                       )
                  )

data = [trace1,trace2]
fig  = go.Figure(data = data,layout = layout)
py.iplot(fig)
```

[20]:
```
print(churn.SeniorCitizen.value_counts().values.tolist())
```

```
[1393, 476]
```

[21]:
```
print(churn.SeniorCitizen.value_counts().values)
```

```
[1393  476]
```

[22]:
```
print(churn.SeniorCitizen.value_counts())
```

```
0    1393
1     476
Name: SeniorCitizen, dtype: int64
```

[ ]:
```

```