In [299…

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv(r"C:\Users\Cali\Downloads\datamarch24\churn_clean_208.csv")
df.shape
#Cleaning the data:
df.duplicated().any() #checking for duplicated records
print(df.isna().sum()) #checking for NaN values.
print(df['InternetService'].unique())
df['InternetService'] = df['InternetService'].fillna('None')
print(df.isna().sum())
df = df.drop(['CaseOrder','Customer_id','Interaction','UID','City','State','County'
z_children = np.abs(stats.zscore(df['Children']))
outliers_children = df[(z_children > 3)]
print(outliers_children[['Children']])
filter_children = df[df['Children'] > 12]
select_children = filter_children[['Age', 'Children']]
print(select_children) #shows how many customers have more than 12 children. None f
z_age = np.abs(stats.zscore(df['Age']))
outliers_age = df[(z_age > 3)]
print(outliers_age[['Age']])
filter_age = df[df['Age'] < 18]
select_age = filter_age[['Age']]
print("Under 18 Years Old:")
print(filter_age)
z_income = np.abs(stats.zscore(df['Income']))
outliers_income = df[(z_income > 3)]
print(outliers_income[['Income']])
filter_income = df[df['Income'] > 200000.00] #only 3 records return and seem reason
select_income = filter_income[['Income']]
print(select_income)
z_tenure = np.abs(stats.zscore(df['Tenure']))
outliers_tenure = df[(z_tenure > 3)]
print(outliers_tenure) #no outliers found
z_bandwidth = np.abs(stats.zscore(df['Bandwidth_GB_Year']))
outliers_bandwidth = df[z_bandwidth > 3]
print(outliers_bandwidth)
z_outage = np.abs(stats.zscore(df['Outage_sec_perweek']))
outlier_outage = df[z_outage > 5]
print(outlier_outage[['Outage_sec_perweek']])
z_email = np.abs(stats.zscore(df['Email']))
outlier_email = df[z_email > 3]
print(outlier_email[['Email']])
z_contacts = np.abs(stats.zscore(df['Contacts']))
outlier_contacts = df[z_contacts > 5]
print(outlier_contacts[['Contacts']])
z_eqfail = np.abs(stats.zscore(df['Yearly_equip_failure']))
outlier_eqfail = df[z_eqfail > 5]
print(outlier_eqfail[['Yearly_equip_failure']])
z_monthlycharge= np.abs(stats.zscore(df['MonthlyCharge']))
outlier_monthlycharge =df[z_monthlycharge > 3]
print(outlier_monthlycharge[['MonthlyCharge']])
```

```
print(df['Marital'].unique()) #Checking for spelling errors/unusual data
print(df['Gender'].unique())
print(df['Churn'].unique())
print(df['Techie'].unique())
print(df['Contract'].unique())
print(df['Port_modem'].unique())
print(df['Tablet'].unique())
print(df['InternetService'].unique())
print(df['Phone'].unique())
print(df['Multiple'].unique())
print(df['OnlineSecurity'].unique())
print(df['OnlineBackup'].unique())
print(df['DeviceProtection'].unique())
print(df['TechSupport'].unique())
print(df['StreamingTV'].unique())
print(df['StreamingMovies'].unique())
print(df['PaperlessBilling'].unique())
```

```
CaseOrder                0
Customer_id              0
Interaction              0
UID                      0
City                     0
State                    0
County                   0
Zip                      0
Lat                      0
Lng                      0
Population               0
Area                     0
TimeZone                 0
Job                      0
Children                 0
Age                      0
Income                   0
Marital                  0
Gender                   0
Churn                    0
Outage_sec_perweek       0
Email                    0
Contacts                 0
Yearly_equip_failure     0
Techie                   0
Contract                 0
Port_modem               0
Tablet                   0
InternetService       2129
Phone                    0
Multiple                 0
OnlineSecurity           0
OnlineBackup             0
DeviceProtection         0
TechSupport              0
StreamingTV              0
StreamingMovies          0
PaperlessBilling         0
PaymentMethod            0
Tenure                   0
MonthlyCharge            0
Bandwidth_GB_Year        0
Item1                    0
Item2                    0
Item3                    0
Item4                    0
Item5                    0
Item6                    0
Item7                    0
Item8                    0
dtype: int64
['Fiber Optic' 'DSL' nan]
CaseOrder            0
Customer_id          0
Interaction          0
UID                  0
```

```
City                    0
State                   0
County                  0
Zip                     0
Lat                     0
Lng                     0
Population              0
Area                    0
TimeZone                0
Job                     0
Children                0
Age                     0
Income                  0
Marital                 0
Gender                  0
Churn                   0
Outage_sec_perweek      0
Email                   0
Contacts                0
Yearly_equip_failure    0
Techie                  0
Contract                0
Port_modem              0
Tablet                  0
InternetService         0
Phone                   0
Multiple                0
OnlineSecurity          0
OnlineBackup            0
DeviceProtection        0
TechSupport             0
StreamingTV             0
StreamingMovies         0
PaperlessBilling        0
PaymentMethod           0
Tenure                  0
MonthlyCharge           0
Bandwidth_GB_Year       0
Item1                   0
Item2                   0
Item3                   0
Item4                   0
Item5                   0
Item6                   0
Item7                   0
Item8                   0
dtype: int64
      Children
30           9
97          10
144         10
329          9
334          9
...        ...
9623        10
9676         9
```

```
9790        10
9871        10
9901         9

[191 rows x 1 columns]
Empty DataFrame
Columns: [Age, Children]
Index: []
Empty DataFrame
Columns: [Age]
Index: []
Under 18 Years Old:
Empty DataFrame
Columns: [Children, Age, Income, Marital, Gender, Churn, Outage_sec_perweek, Email,
Contacts, Yearly_equip_failure, Techie, Contract, Port_modem, Tablet, InternetServic
e, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Str
eamingTV, StreamingMovies, PaperlessBilling, Tenure, MonthlyCharge, Bandwidth_GB_Yea
r]
Index: []

[0 rows x 27 columns]
         Income
46    132116.33
130   125814.88
186   135727.71
470   156740.67
511   146494.70
...         ...
9615  130319.30
9639  149952.70
9656  136818.50
9849  134443.30
9876  128468.00

[145 rows x 1 columns]
        Income
4249  258900.7
5599  220383.0
5801  212255.3
6649  231252.0
9180  256998.4
Empty DataFrame
Columns: [Children, Age, Income, Marital, Gender, Churn, Outage_sec_perweek, Email,
Contacts, Yearly_equip_failure, Techie, Contract, Port_modem, Tablet, InternetServic
e, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Str
eamingTV, StreamingMovies, PaperlessBilling, Tenure, MonthlyCharge, Bandwidth_GB_Yea
r]
Index: []

[0 rows x 27 columns]
Empty DataFrame
Columns: [Children, Age, Income, Marital, Gender, Churn, Outage_sec_perweek, Email,
Contacts, Yearly_equip_failure, Techie, Contract, Port_modem, Tablet, InternetServic
e, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Str
eamingTV, StreamingMovies, PaperlessBilling, Tenure, MonthlyCharge, Bandwidth_GB_Yea
r]
```

```
Index: []

[0 rows x 27 columns]
Empty DataFrame
Columns: [Outage_sec_perweek]
Index: []
      Email
795       2
1152      2
1381      1
1399      2
1473     23
1746     22
6320      1
7408      2
8365      1
8948      2
9248      2
9475     22
      Contacts
426          6
4673         6
4811         7
5840         6
7746         7
9380         7
9713         6
9750         6
      Yearly_equip_failure
1116                     4
1228                     4
5166                     4
5471                     6
6345                     4
9386                     4
9623                     4
9763                     4
Empty DataFrame
Columns: [MonthlyCharge]
Index: []
['Widowed' 'Married' 'Separated' 'Never Married' 'Divorced']
['Male' 'Female' 'Nonbinary']
['No' 'Yes']
['No' 'Yes']
['One year' 'Month-to-month' 'Two Year']
['Yes' 'No']
['Yes' 'No']
['Fiber Optic' 'DSL' 'None']
['Yes' 'No']
['No' 'Yes']
['Yes' 'No']
['Yes' 'No']
['No' 'Yes']
['No' 'Yes']
['No' 'Yes']
```

```
['Yes' 'No']
['Yes' 'No']
```

In [319…

```
print(df['Children'].describe()) #description statistics
print(df['Age'].describe())
print(df['Marital'].describe())
print(df['Gender'].describe())
print(df['Churn'].describe())
print(df['Outage_sec_perweek'].describe())
print(df['Email'].describe())
print(df['Contacts'].describe())
print(df['Yearly_equip_failure'].describe())
print(df['Contract'].describe())
print(df['Port_modem'].describe())
print(df['Tablet'].describe())
print(df['InternetService'].describe())
print(df['Phone'].describe())
print(df['Multiple'].describe())
print(df['OnlineSecurity'].describe())
print(df['OnlineBackup'].describe())
print(df['DeviceProtection'].describe())
print(df['TechSupport'].describe())
print(df['StreamingTV'].describe())
print(df['StreamingMovies'].describe())
print(df['Tenure'].describe())
print(df['Bandwidth_GB_Year'].describe())
print(df['MonthlyCharge'].describe())
```

```
count    10000.0000
mean         2.0877
std          2.1472
min          0.0000
25%          0.0000
50%          1.0000
75%          3.0000
max         10.0000
Name: Children, dtype: float64
count    10000.000000
mean        53.078400
std         20.698882
min         18.000000
25%         35.000000
50%         53.000000
75%         71.000000
max         89.000000
Name: Age, dtype: float64
count        10000
unique           5
top       Divorced
freq          2092
Name: Marital, dtype: object
count      10000
unique         3
top       Female
freq        5025
Name: Gender, dtype: object
count    10000.000000
mean         0.265000
std          0.441355
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: Churn, dtype: float64
count    10000.000000
mean        10.001848
std          2.976019
min          0.099747
25%          8.018214
50%         10.018560
75%         11.969485
max         21.207230
Name: Outage_sec_perweek, dtype: float64
count    10000.000000
mean        12.016000
std          3.025898
min          1.000000
25%         10.000000
50%         12.000000
75%         14.000000
max         23.000000
Name: Email, dtype: float64
count    10000.000000
```

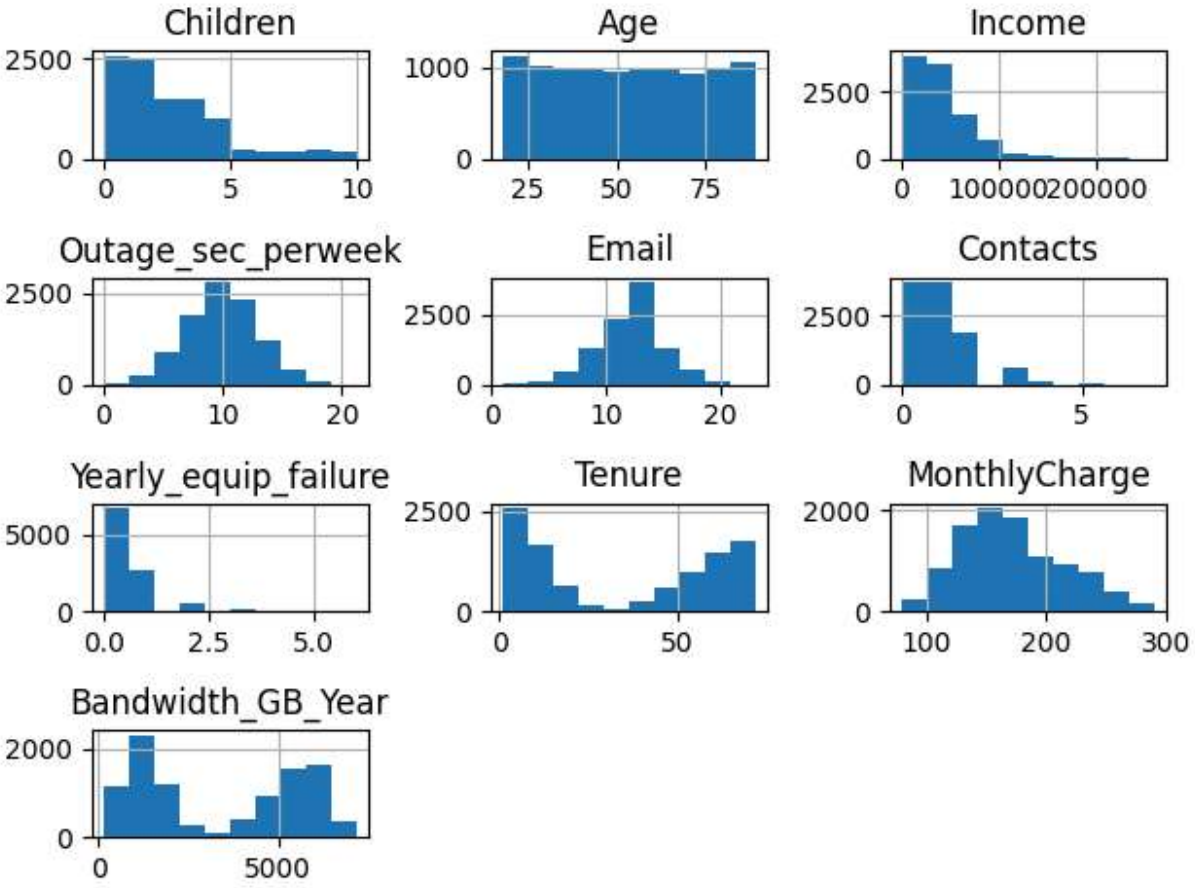
```
mean         0.994200
std          0.988466
min          0.000000
25%          0.000000
50%          1.000000
75%          2.000000
max          7.000000
Name: Contacts, dtype: float64
count    10000.000000
mean         0.398000
std          0.635953
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          6.000000
Name: Yearly_equip_failure, dtype: float64
count              10000
unique                 3
top       Month-to-month
freq                5456
Name: Contract, dtype: object
count    10000.000000
mean         0.483400
std          0.499749
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: Port_modem, dtype: float64
count    10000.000000
mean         0.299100
std          0.457887
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: Tablet, dtype: float64
count           10000
unique              3
top       Fiber Optic
freq             4408
Name: InternetService, dtype: object
count    10000.000000
mean         0.906700
std          0.290867
min          0.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          1.000000
Name: Phone, dtype: float64
count    10000.000000
mean         0.460800
```

```
std          0.498486
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: Multiple, dtype: float64
count    10000.000000
mean         0.357600
std          0.479317
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: OnlineSecurity, dtype: float64
count    10000.000000
mean         0.450600
std          0.497579
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: OnlineBackup, dtype: float64
count    10000.000000
mean         0.438600
std          0.496241
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: DeviceProtection, dtype: float64
count    10000.000000
mean         0.375000
std          0.484147
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: TechSupport, dtype: float64
count    10000.000000
mean         0.492900
std          0.499975
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: StreamingTV, dtype: float64
count    10000.000000
mean         0.489000
std          0.499904
min          0.000000
```

```
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: StreamingMovies, dtype: float64
count    10000.000000
mean        34.526188
std         26.443063
min          1.000259
25%          7.917694
50%         35.430507
75%         61.479795
max         71.999280
Name: Tenure, dtype: float64
count    10000.000000
mean      3392.341550
std       2185.294852
min        155.506715
25%       1236.470827
50%       3279.536903
75%       5586.141370
max       7158.981530
Name: Bandwidth_GB_Year, dtype: float64
count    10000.000000
mean       172.624816
std         42.943094
min         79.978860
25%        139.979239
50%        167.484700
75%        200.734725
max        290.160419
Name: MonthlyCharge, dtype: float64
```

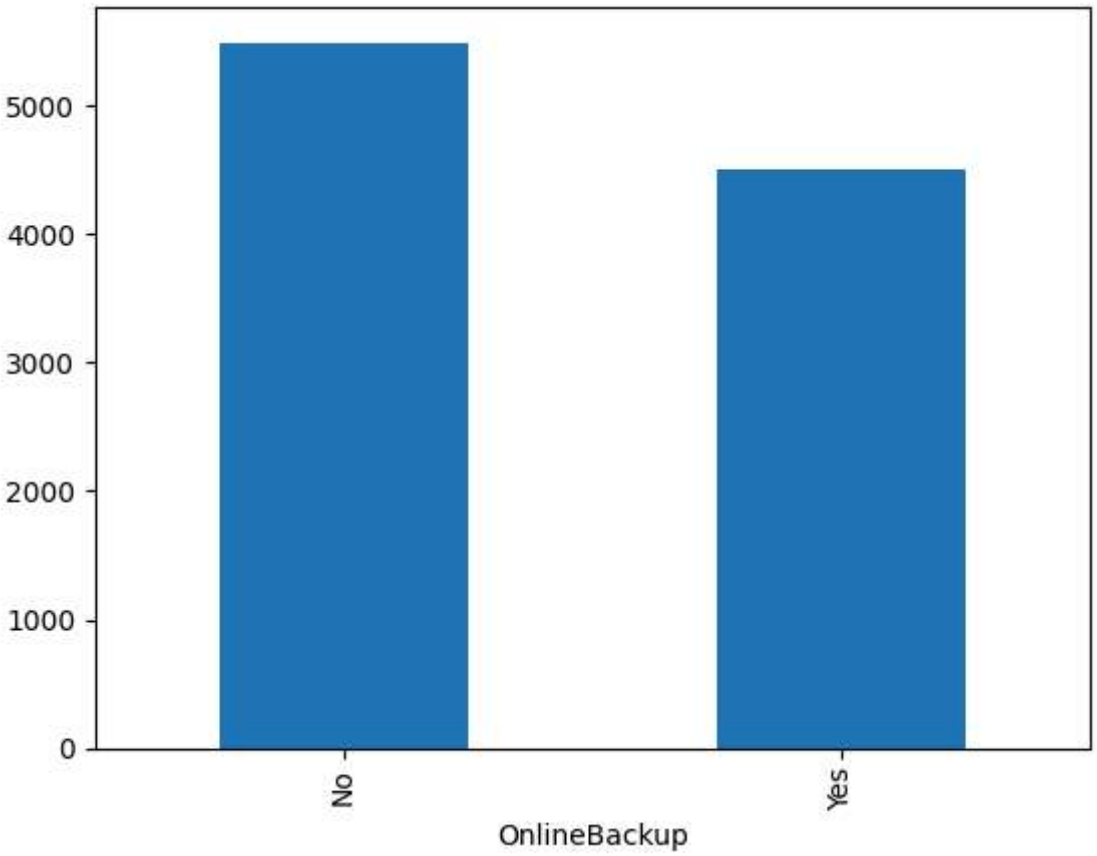
In [241…

```
df[['Children','Age', 'Income', 'Outage_sec_perweek','Email','Contacts','Yearly_equ
plt.savefig('churn_pyplot.jpg')
plt.tight_layout()
```



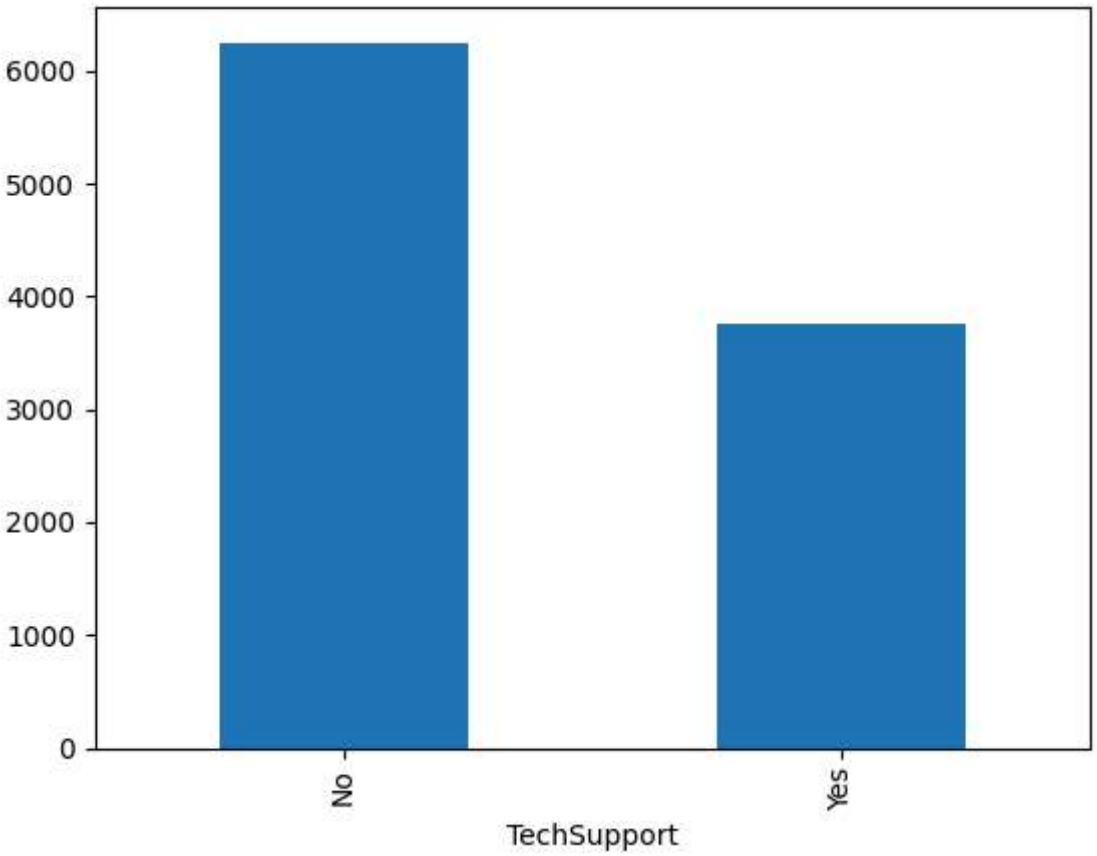
In [242…
```
df['OnlineBackup'].value_counts().plot(kind='bar') #Categorical Univariate visuals
```

Out[242…
```
<Axes: xlabel='OnlineBackup'>
```



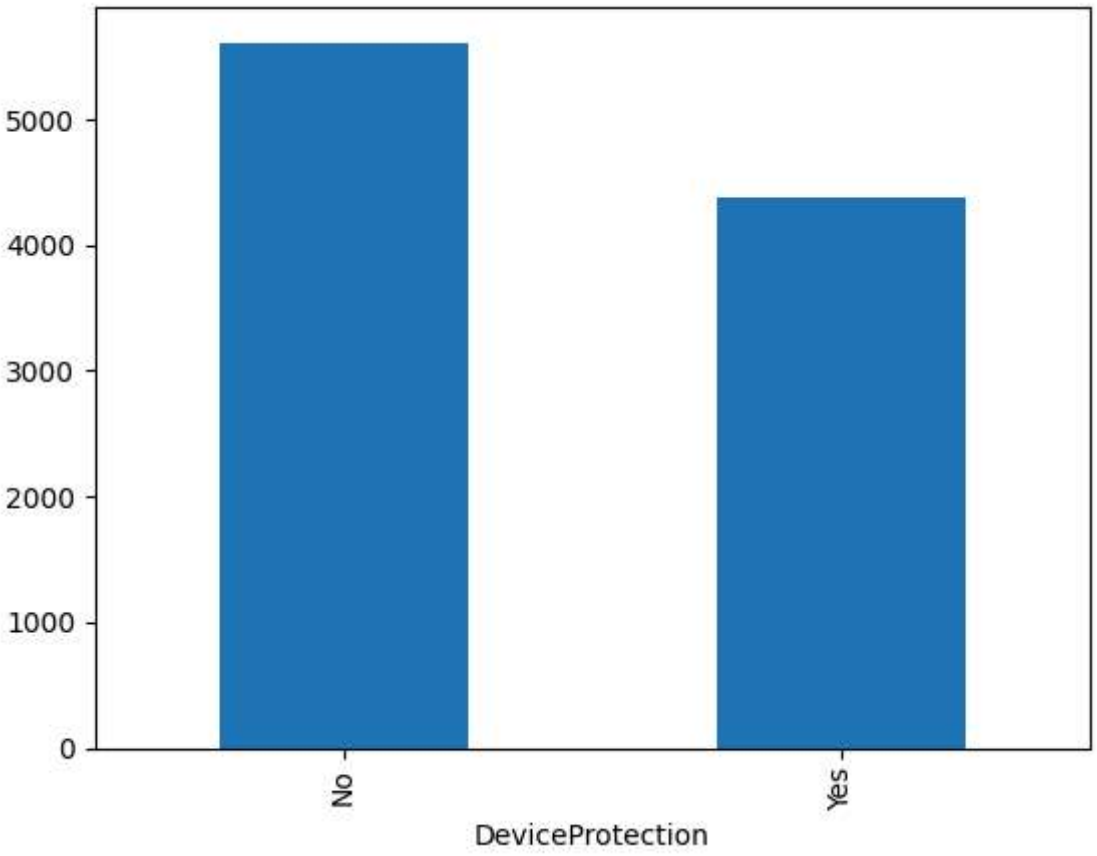
In [243…
```
df['TechSupport'].value_counts().plot(kind='bar')
```

Out[243…
```
<Axes: xlabel='TechSupport'>
```



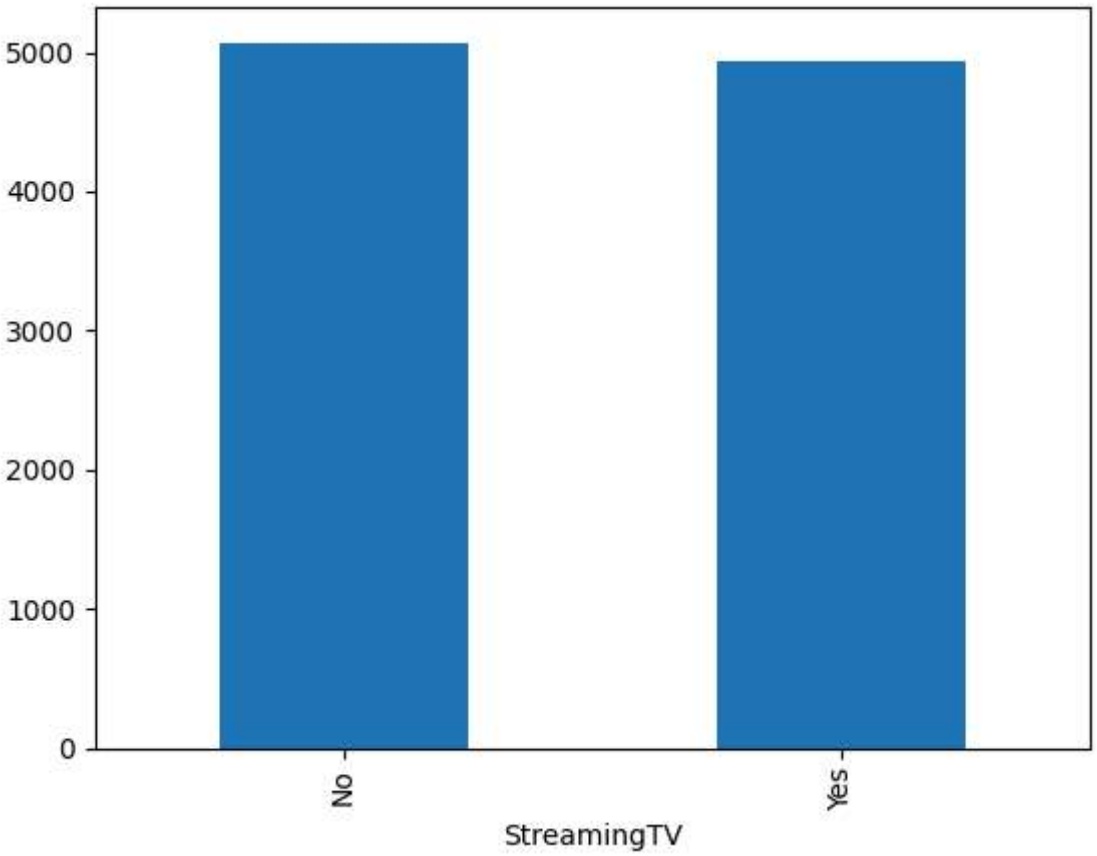
In [244…
```
df['DeviceProtection'].value_counts().plot(kind='bar')
```

Out[244…
```
<Axes: xlabel='DeviceProtection'>
```



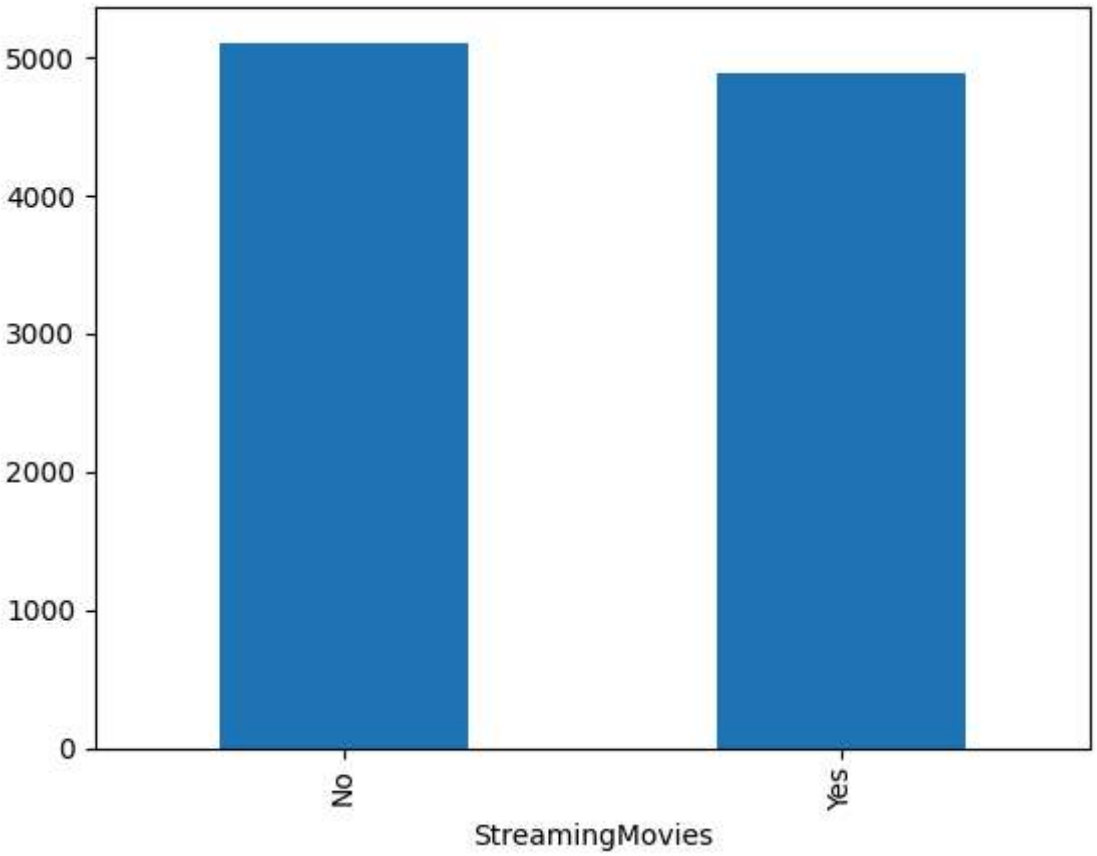
In [245…
```
df['StreamingTV'].value_counts().plot(kind='bar')
```

Out[245…
```
<Axes: xlabel='StreamingTV'>
```



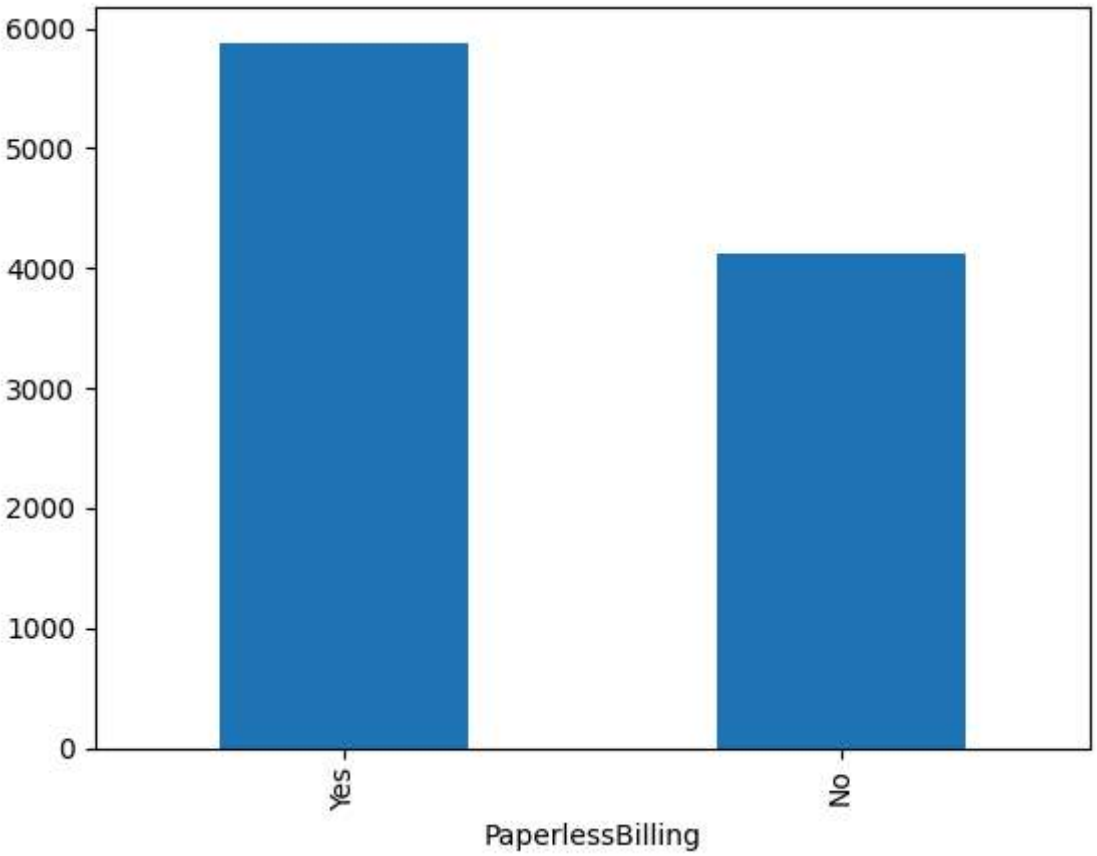
In [246…
```
df['StreamingMovies'].value_counts().plot(kind='bar')
```

Out[246…
```
<Axes: xlabel='StreamingMovies'>
```



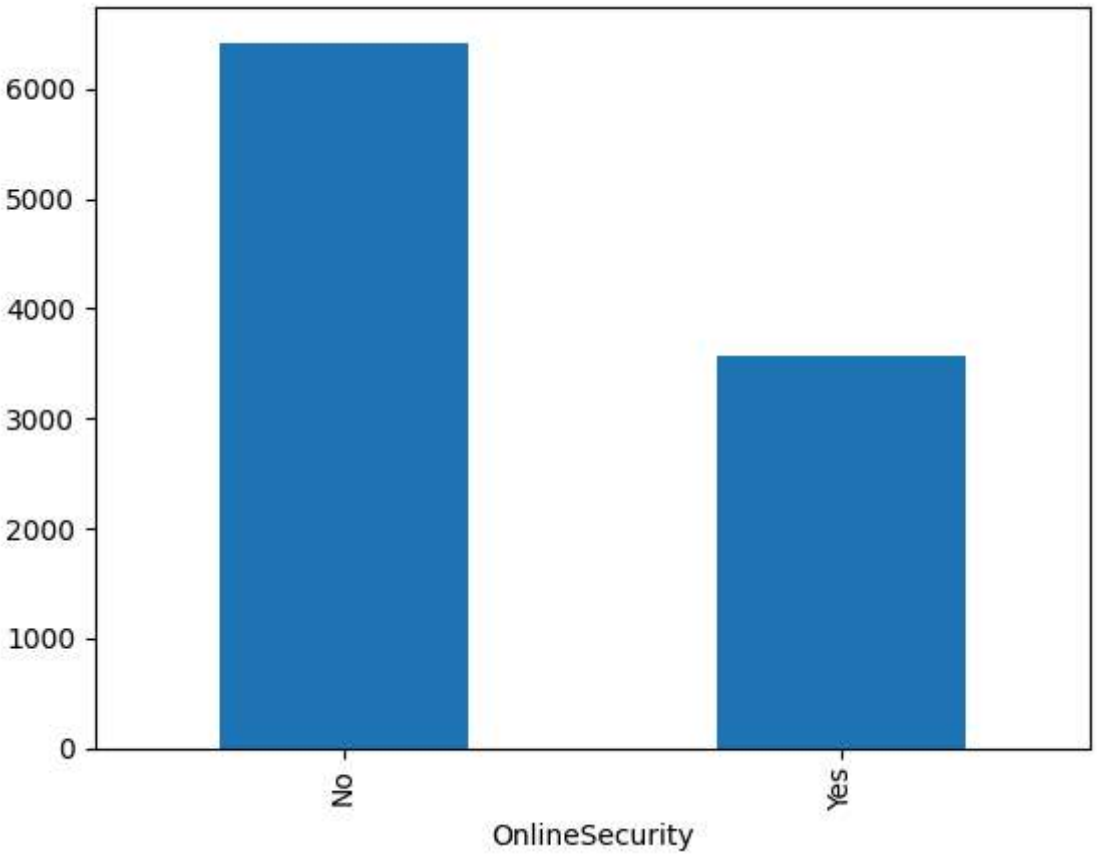
In [247…
```
df['PaperlessBilling'].value_counts().plot(kind='bar')
```

Out[247…
```
<Axes: xlabel='PaperlessBilling'>
```



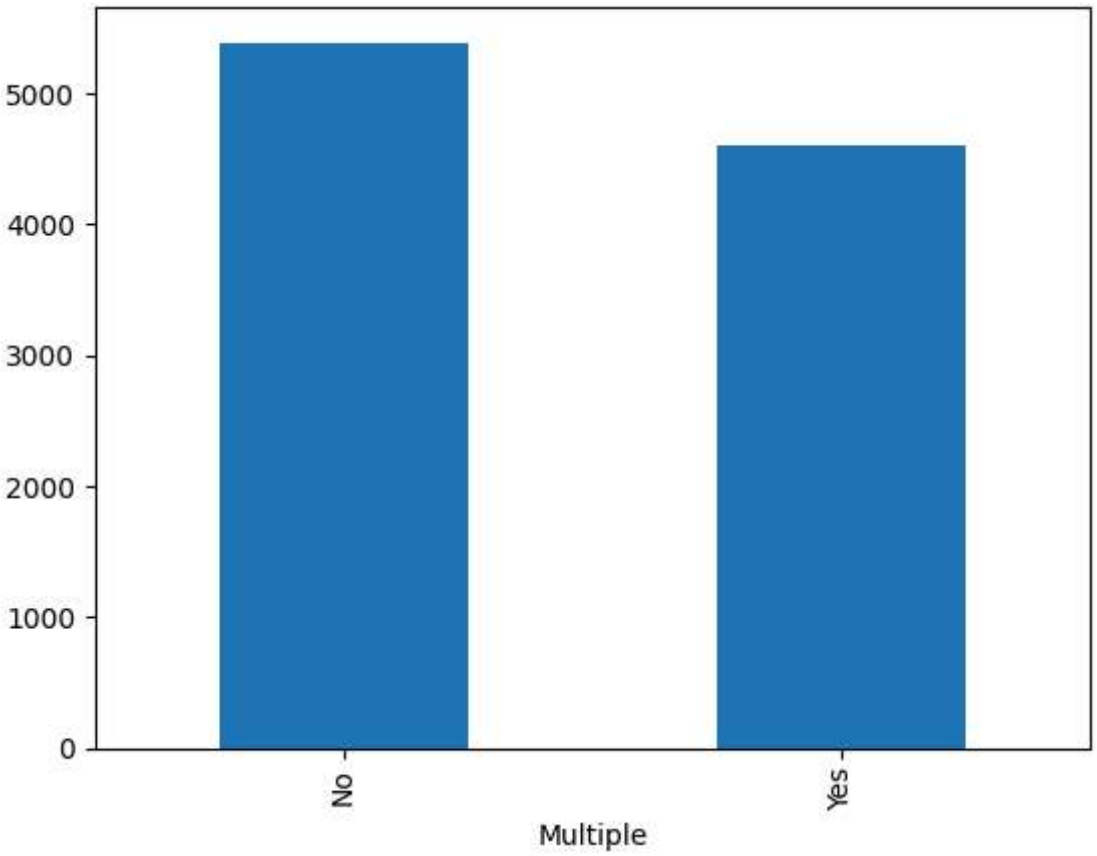
In [248…
```
df['OnlineSecurity'].value_counts().plot(kind='bar')
```

Out[248…
```
<Axes: xlabel='OnlineSecurity'>
```



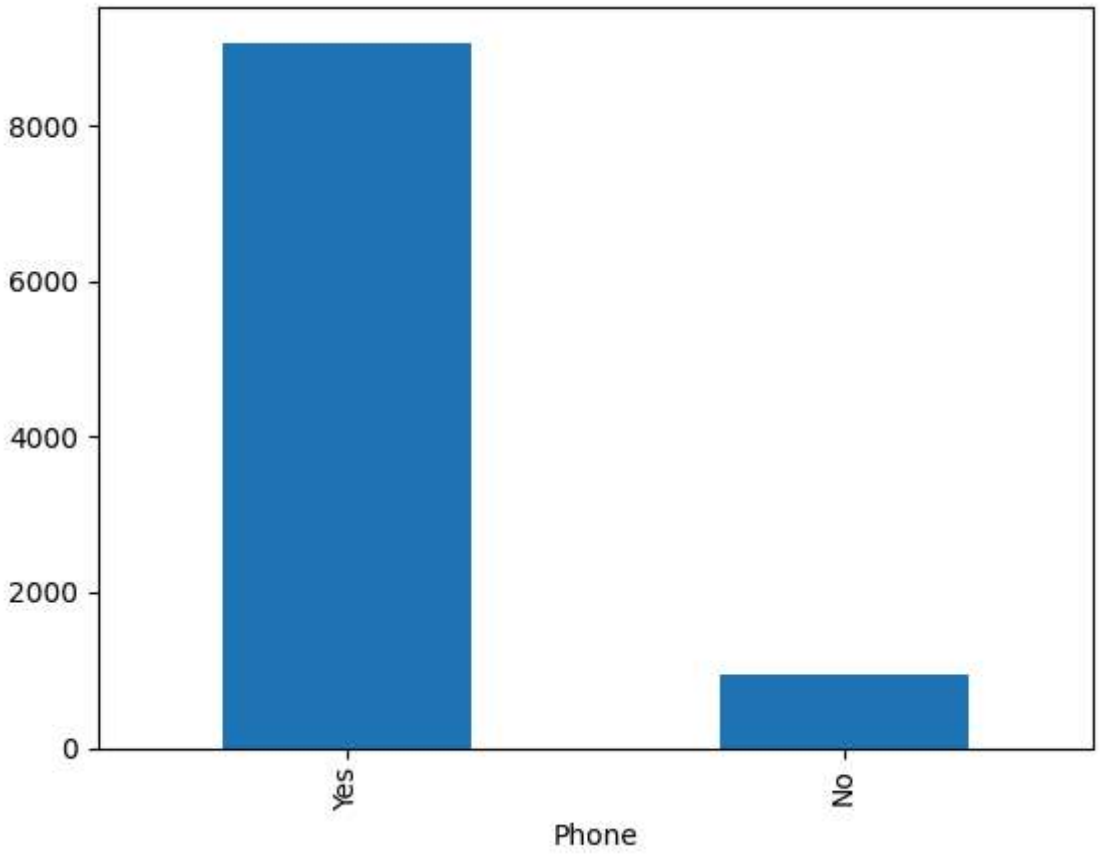
In [249…
```
df['Multiple'].value_counts().plot(kind='bar')
```

Out[249…
```
<Axes: xlabel='Multiple'>
```



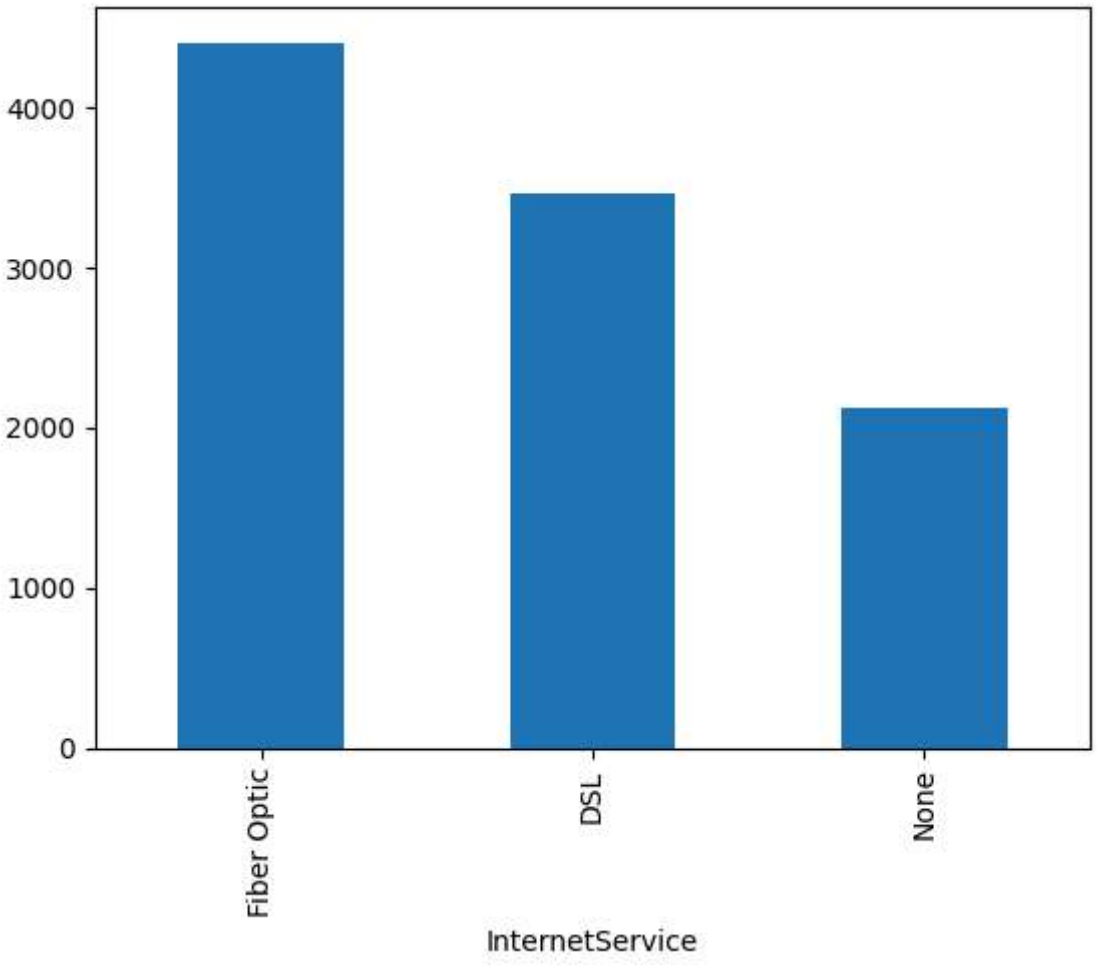
In [250…
```
df['Phone'].value_counts().plot(kind='bar')
```

Out[250…
```
<Axes: xlabel='Phone'>
```



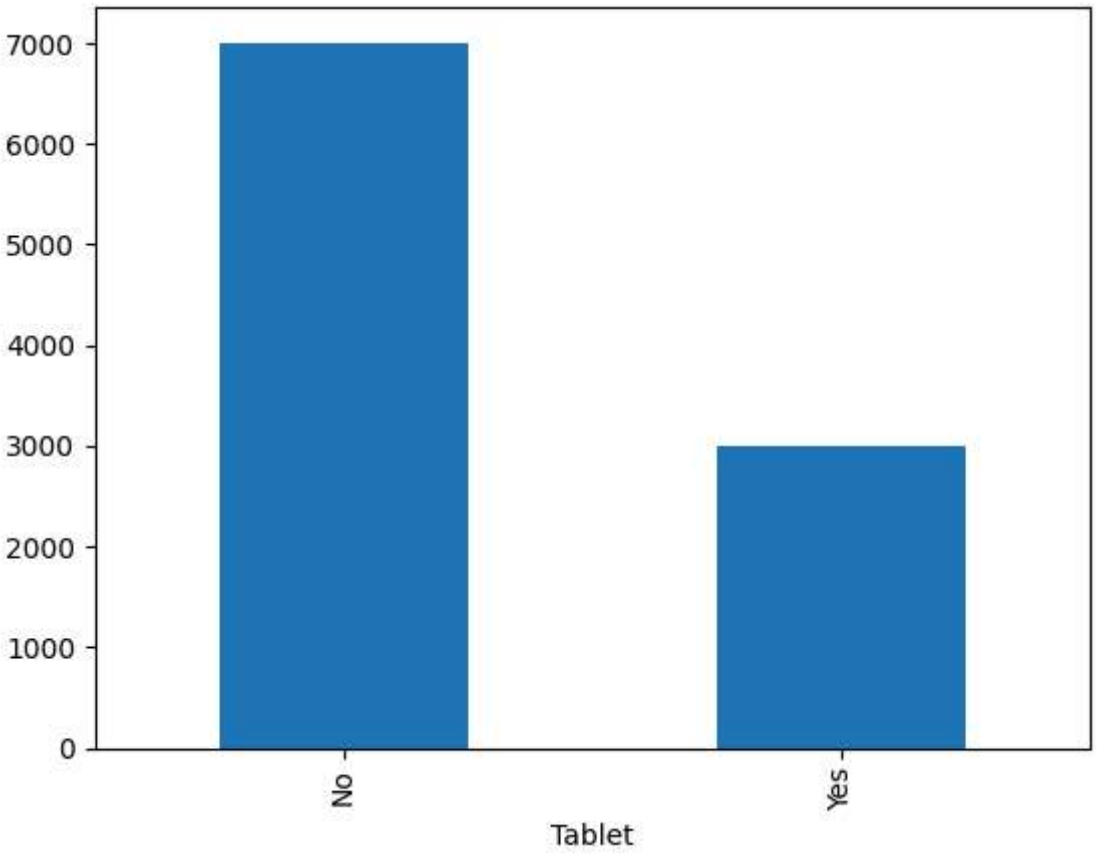
In [300…
```
df['InternetService'].value_counts().plot(kind='bar')
```

Out[300…
```
<Axes: xlabel='InternetService'>
```



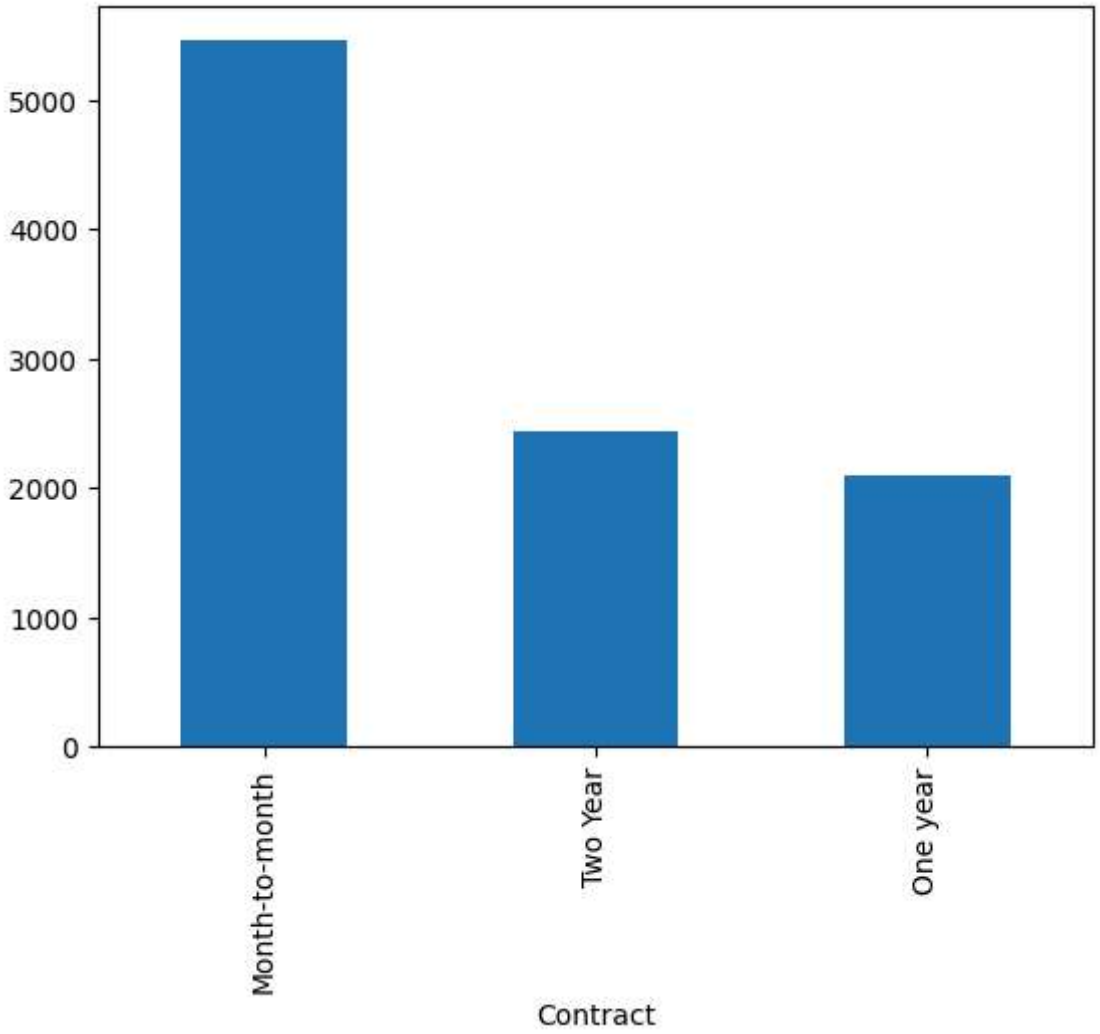
In [252…
```
df['Tablet'].value_counts().plot(kind='bar')
```

Out[252…
```
<Axes: xlabel='Tablet'>
```



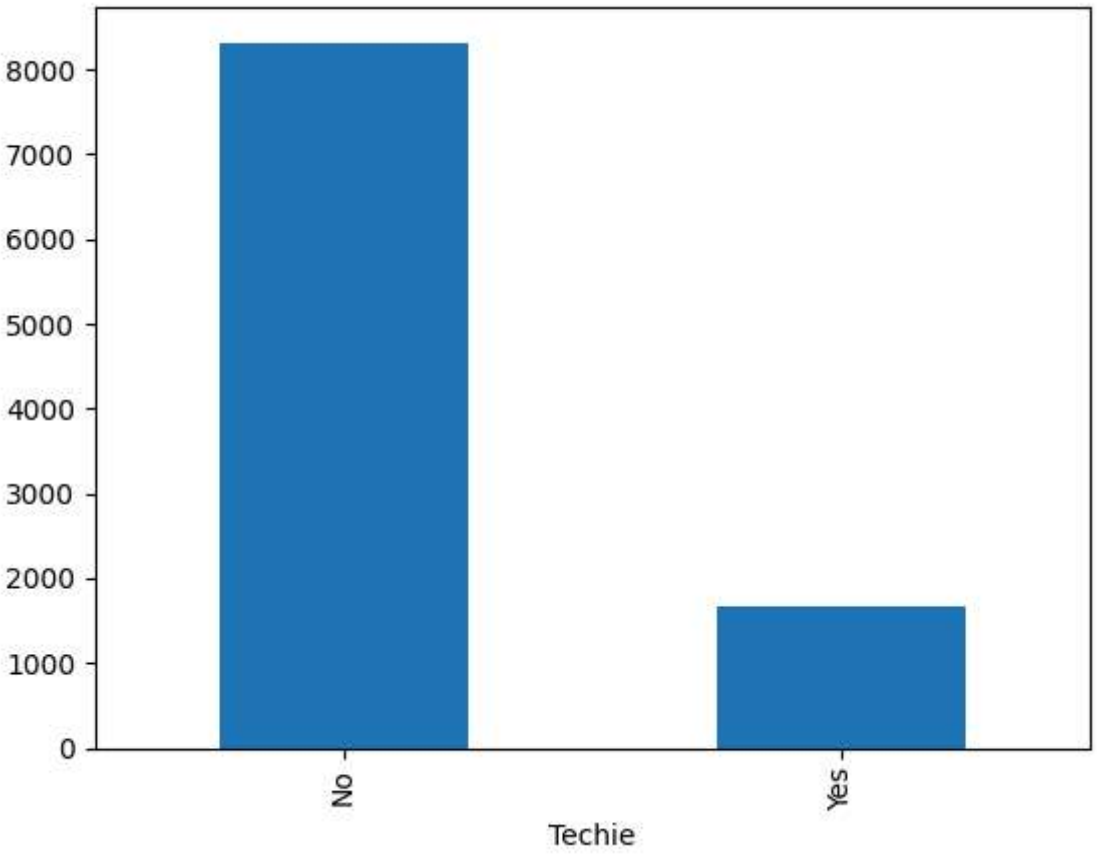
In [253…
```
df['Contract'].value_counts().plot(kind='bar')
```

Out[253…
```
<Axes: xlabel='Contract'>
```



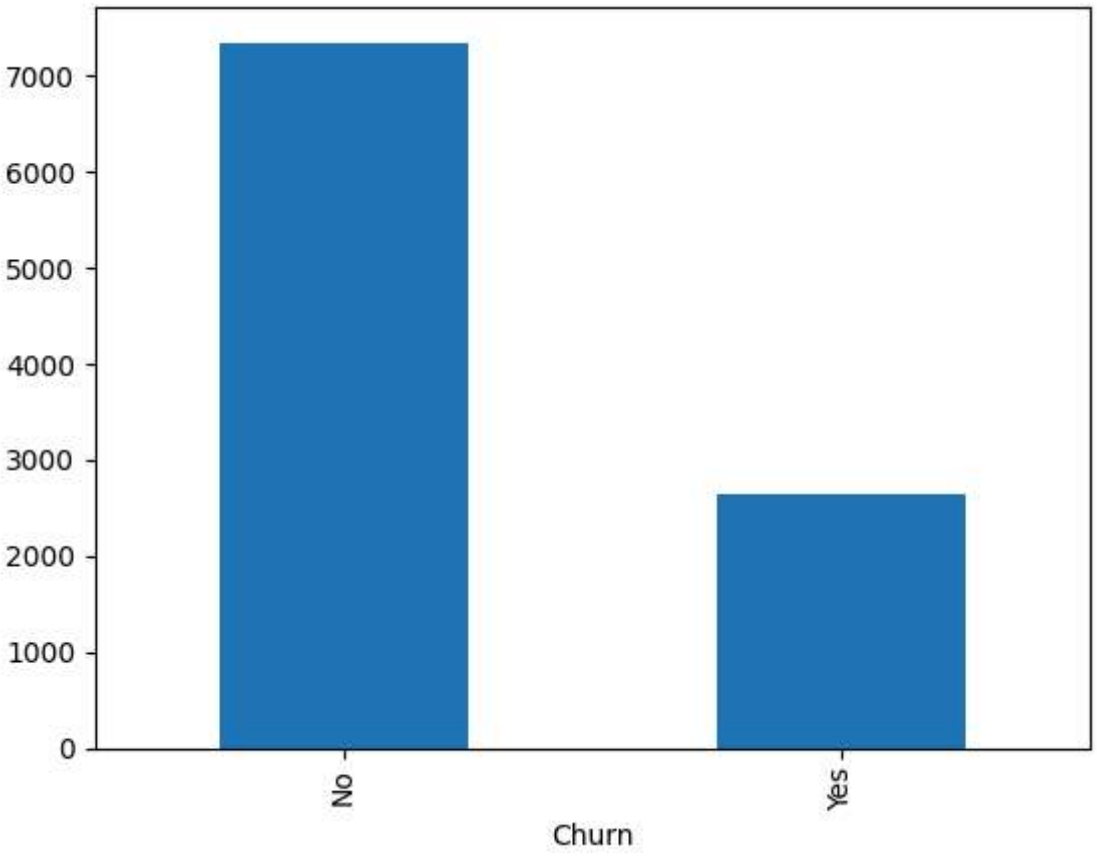
In [254…
```
df['Techie'].value_counts().plot(kind='bar')
```

Out[254…
```
<Axes: xlabel='Techie'>
```



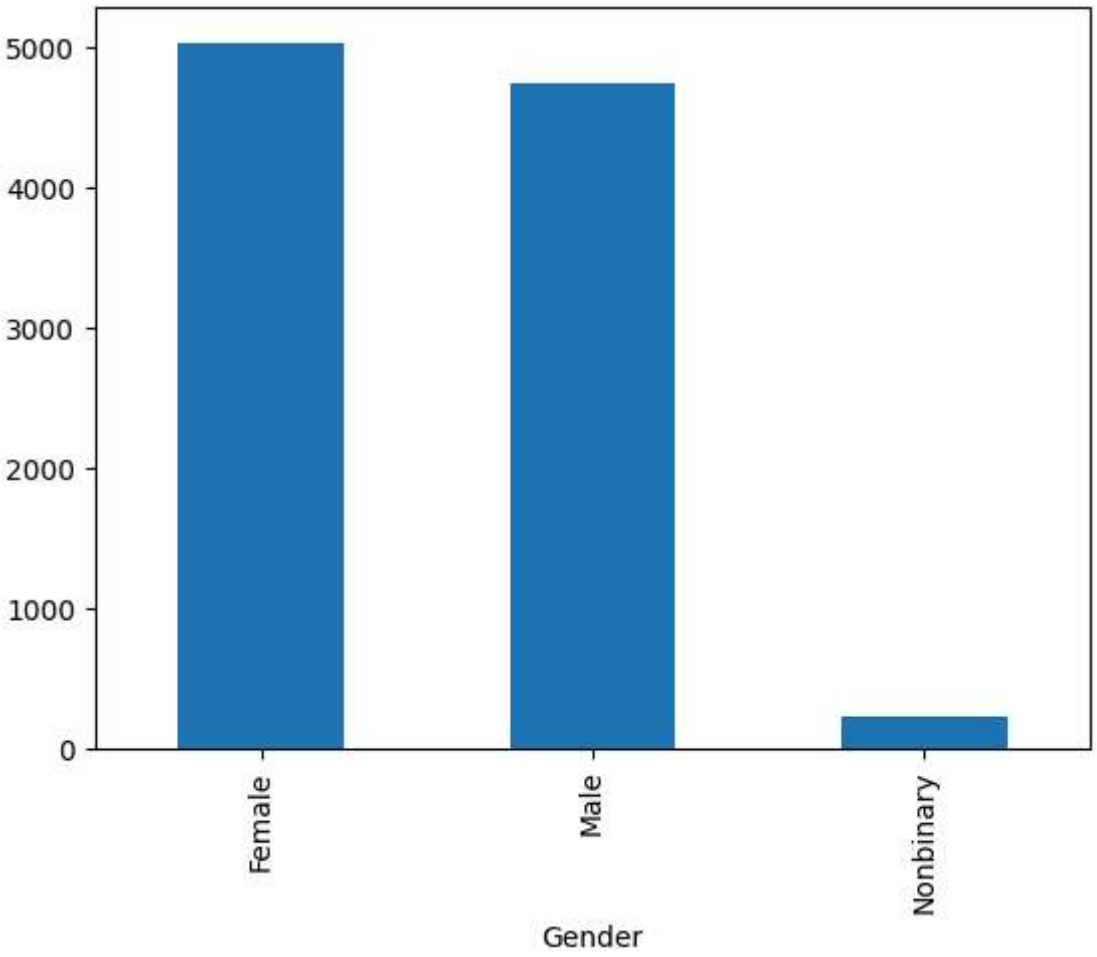
In [255…
```
df['Churn'].value_counts().plot(kind='bar')
```

Out[255…
```
<Axes: xlabel='Churn'>
```



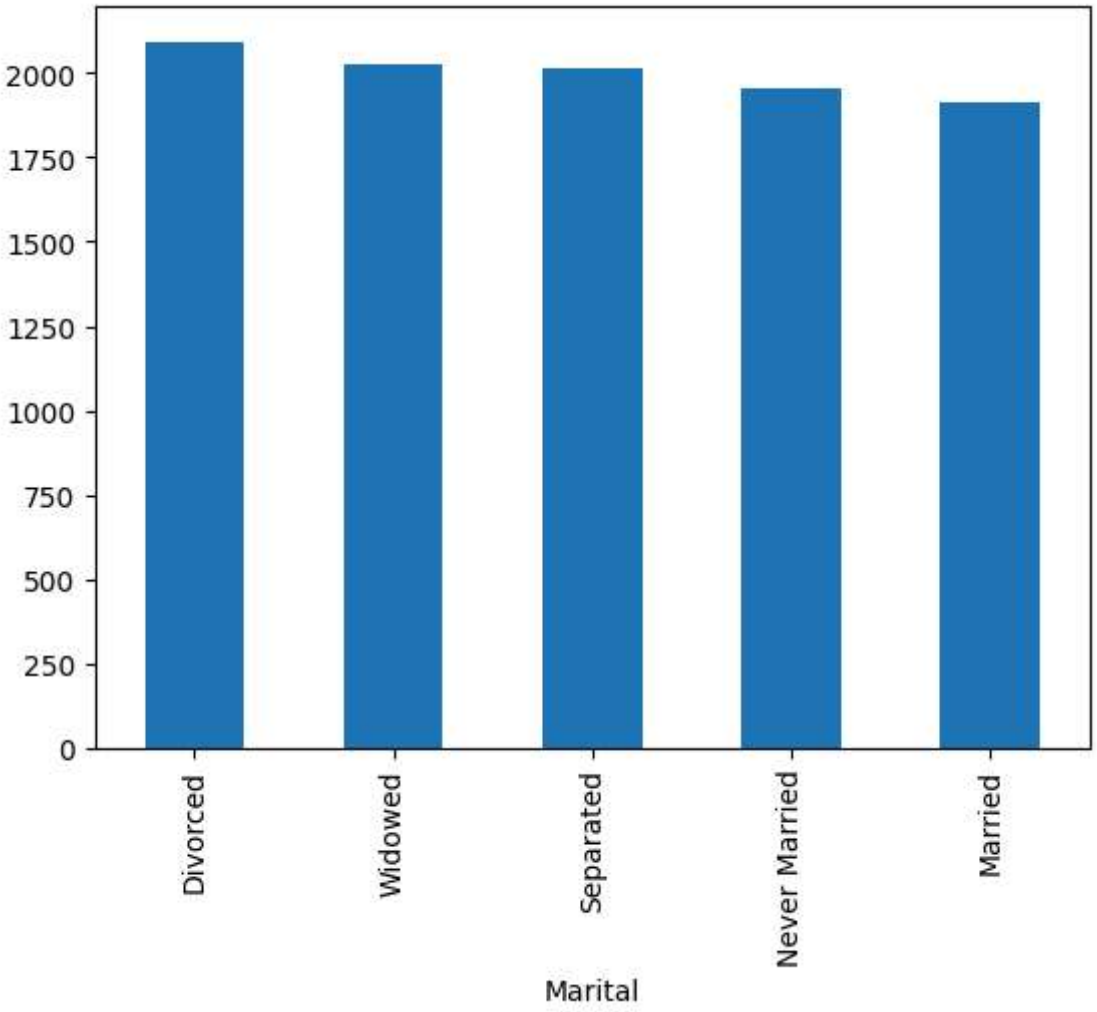
In [256…
```
df['Gender'].value_counts().plot(kind='bar')
```

Out[256…
```
<Axes: xlabel='Gender'>
```



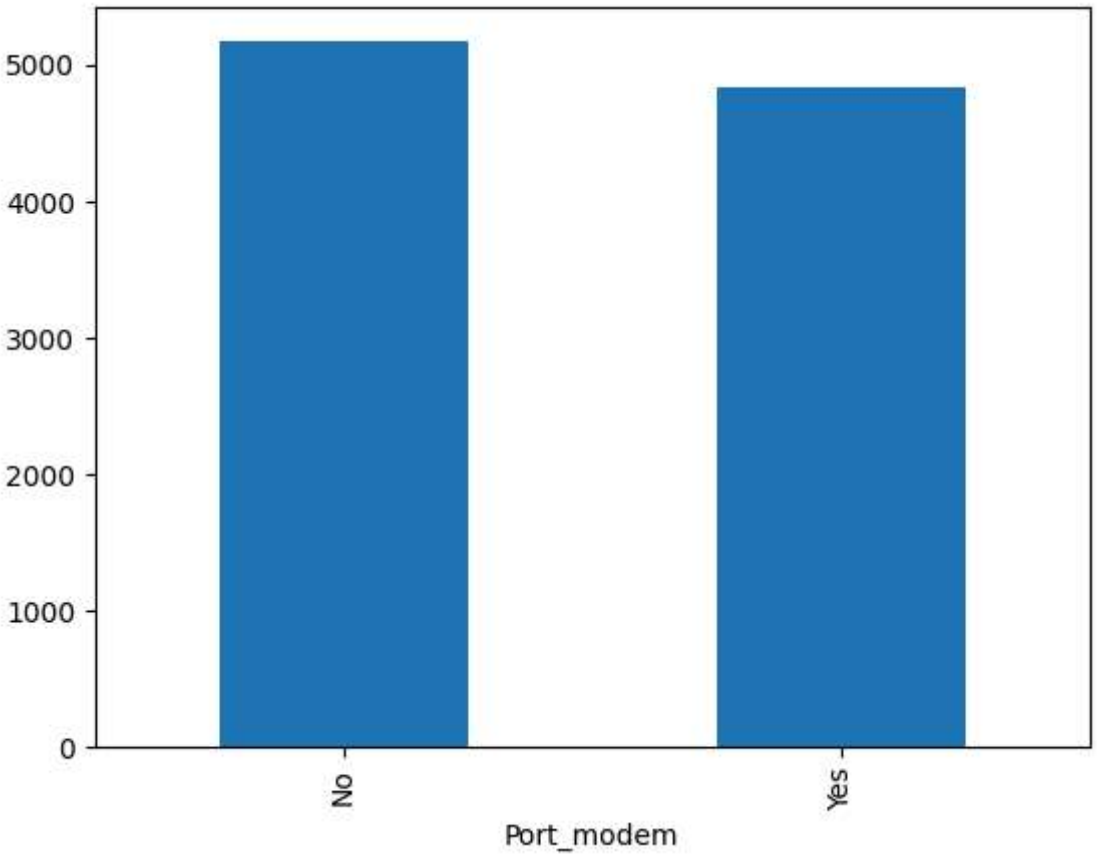
In [257…
```
df['Marital'].value_counts().plot(kind='bar')
```

Out[257…
```
<Axes: xlabel='Marital'>
```



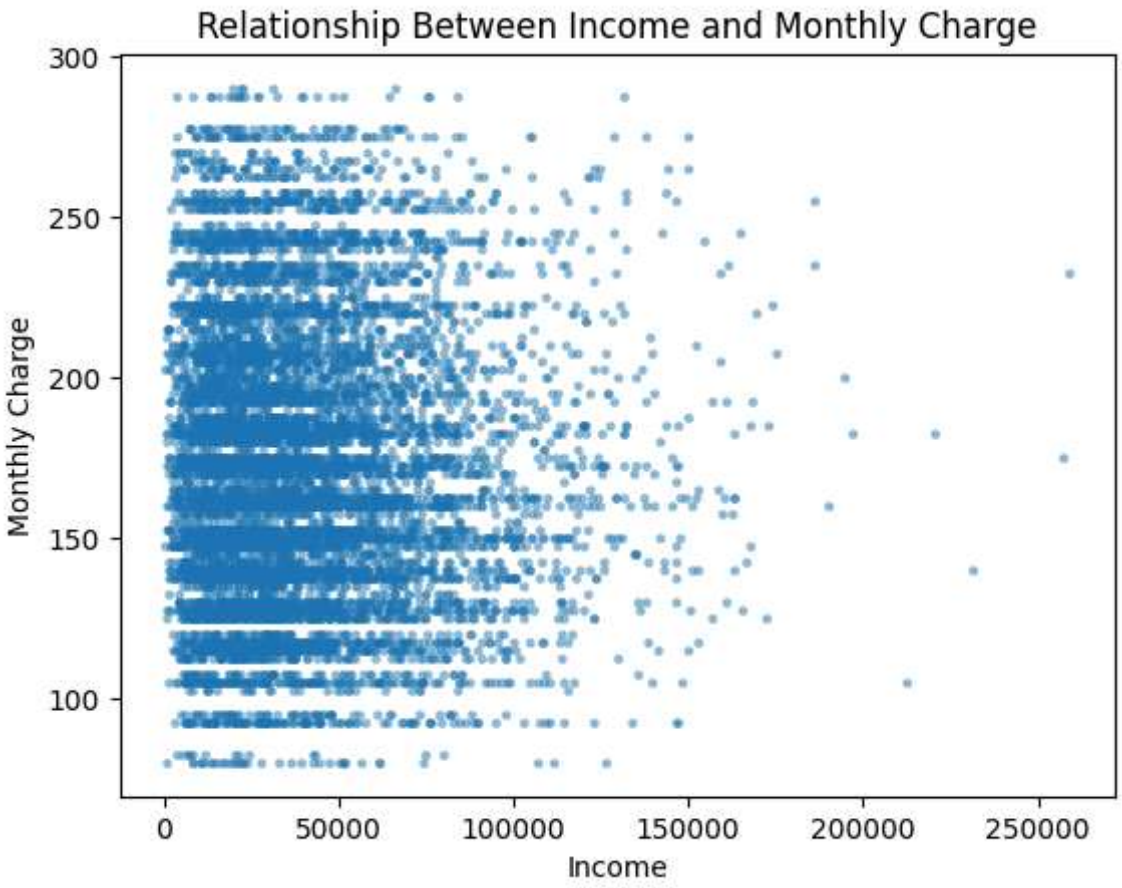
In [258…
```
df['Port_modem'].value_counts().plot(kind='bar')
```

Out[258…
```
<Axes: xlabel='Port_modem'>
```



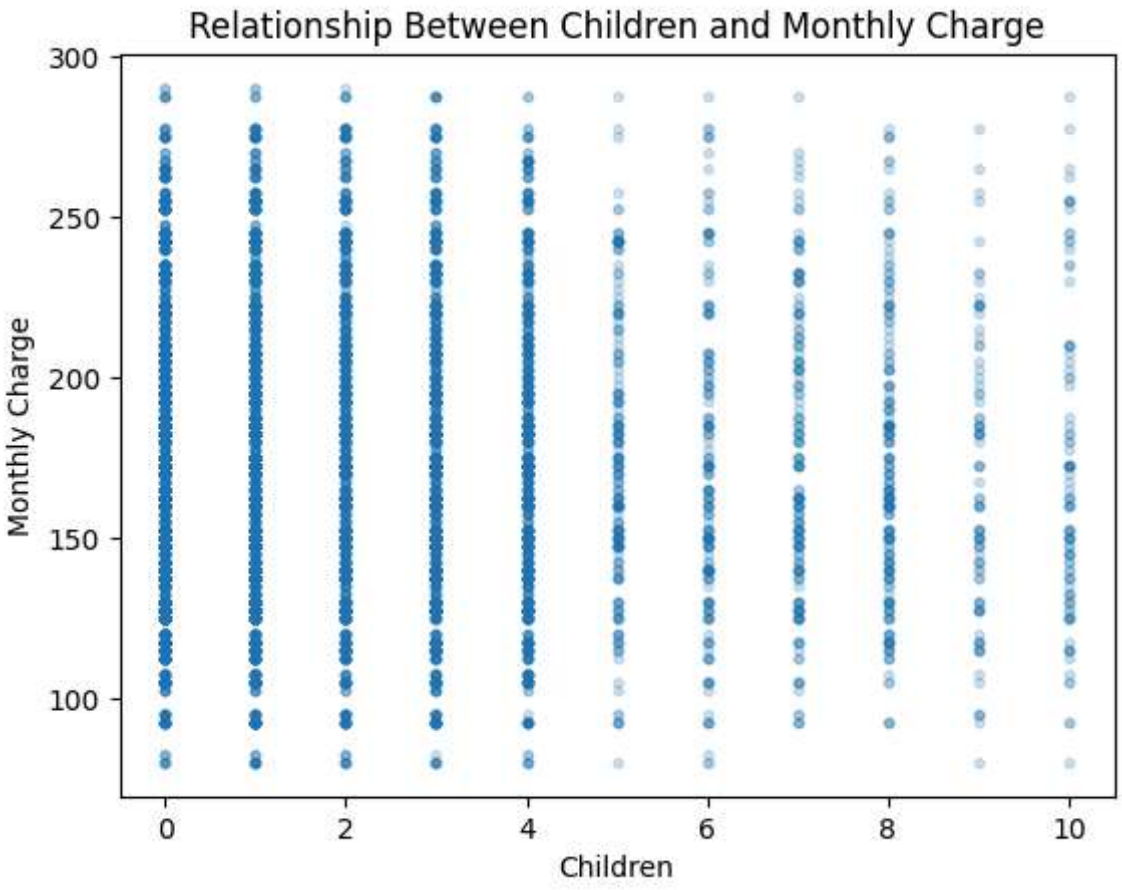
In [260…
```
plt.scatter(df.Income, df.MonthlyCharge, s=6, alpha=0.4) #Bivariate Statistic Visua
plt.title('Relationship Between Income and Monthly Charge')
plt.xlabel('Income')
plt.ylabel('Monthly Charge')
```

Out[260…
```
Text(0, 0.5, 'Monthly Charge')
```



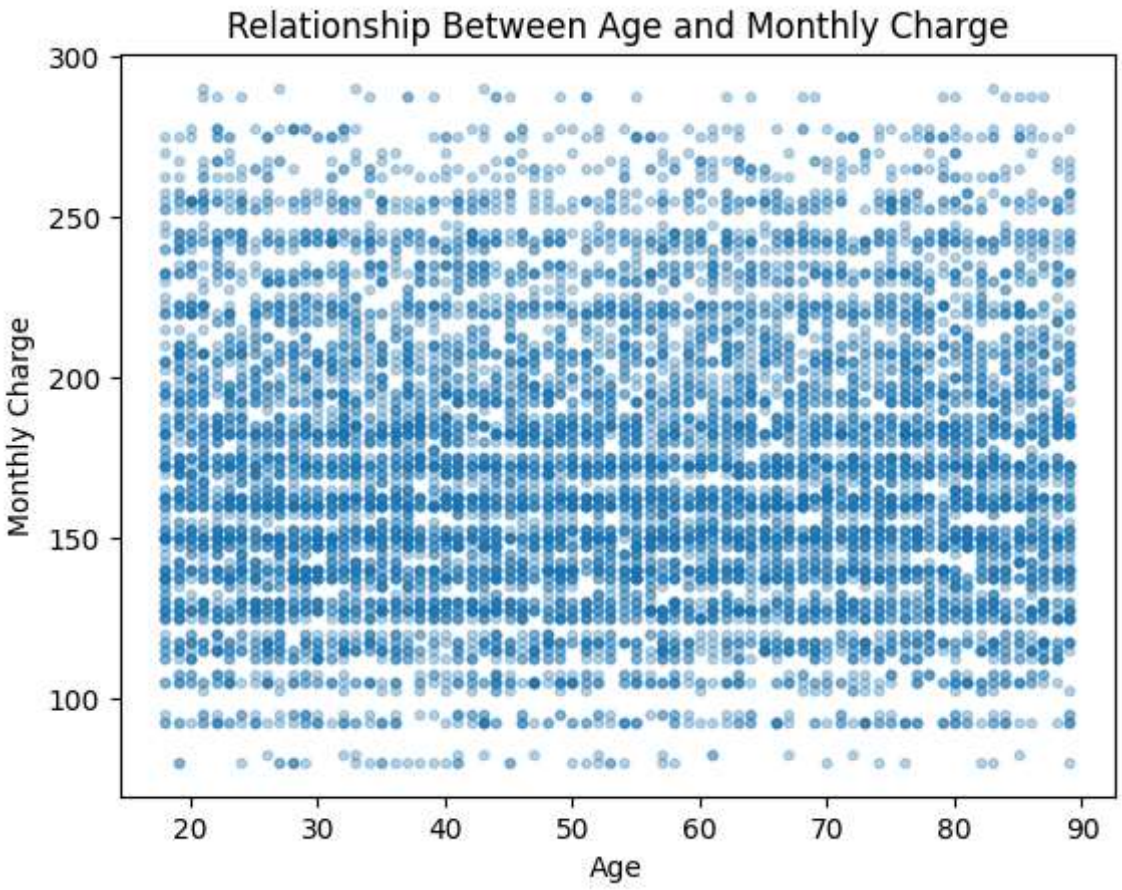
In [261…
```
plt.scatter(df.Children, df.MonthlyCharge, s=10, alpha=0.2)
plt.title('Relationship Between Children and Monthly Charge')
plt.xlabel('Children')
plt.ylabel('Monthly Charge')
```

Out[261…
```
Text(0, 0.5, 'Monthly Charge')
```



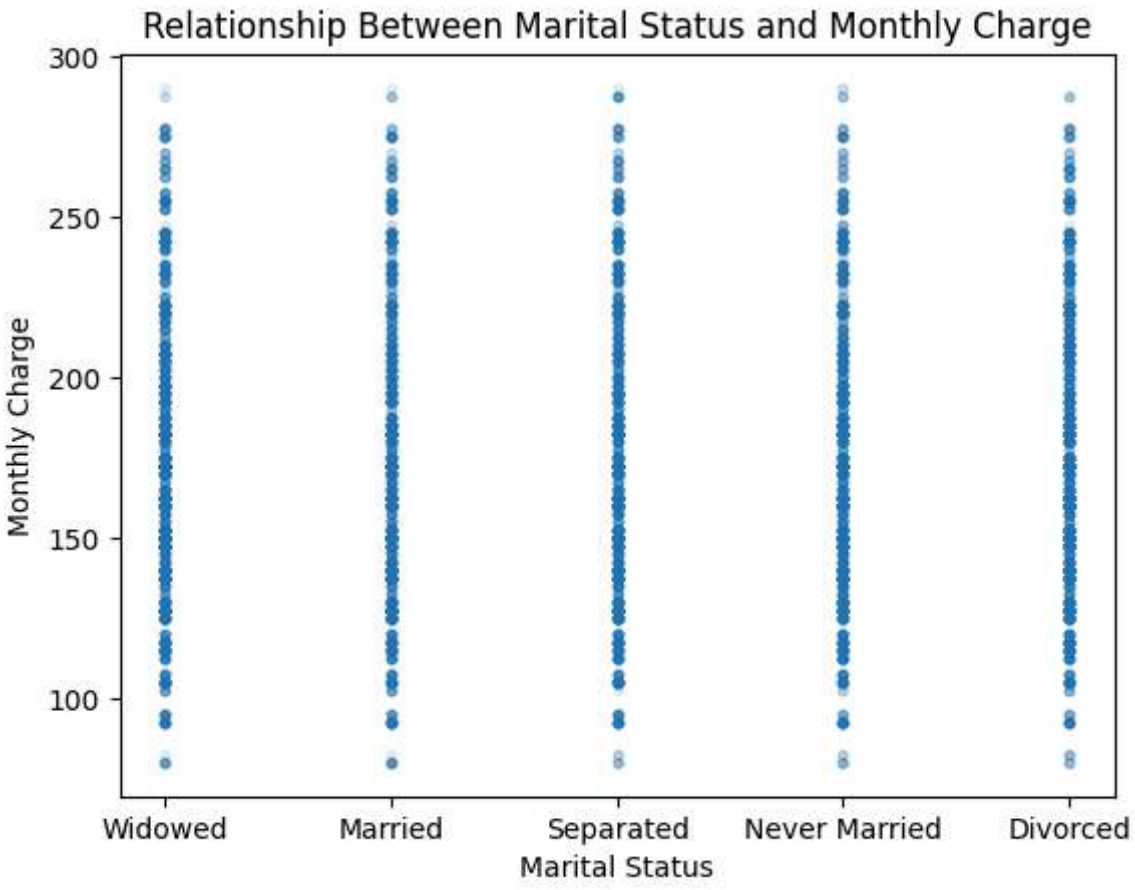
In [262…
```
plt.scatter(df.Age, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Age and Monthly Charge')
plt.xlabel('Age')
plt.ylabel('Monthly Charge')
```

Out[262…
```
Text(0, 0.5, 'Monthly Charge')
```



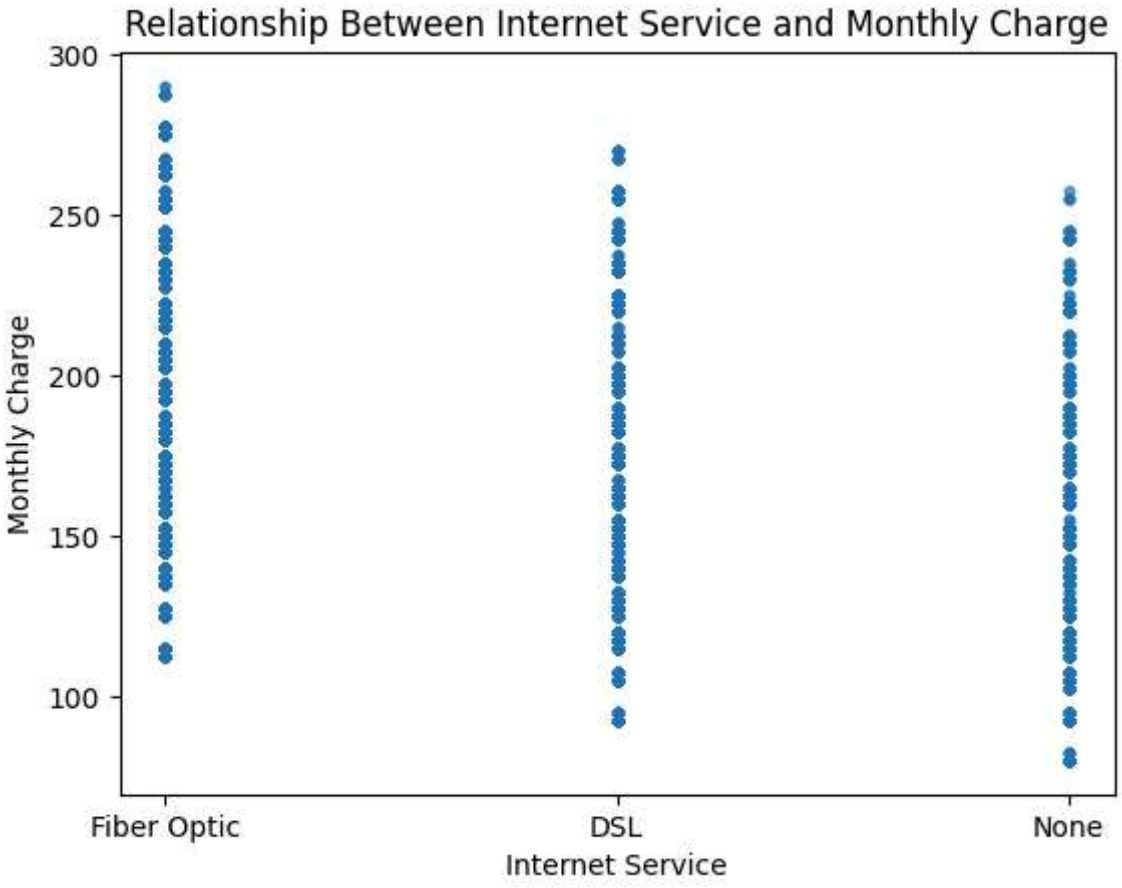
In [263…
```
plt.scatter(df.Marital, df.MonthlyCharge, s=10, alpha=0.1)
plt.title('Relationship Between Marital Status and Monthly Charge')
plt.xlabel('Marital Status')
plt.ylabel('Monthly Charge')
```

Out[263…
```
Text(0, 0.5, 'Monthly Charge')
```



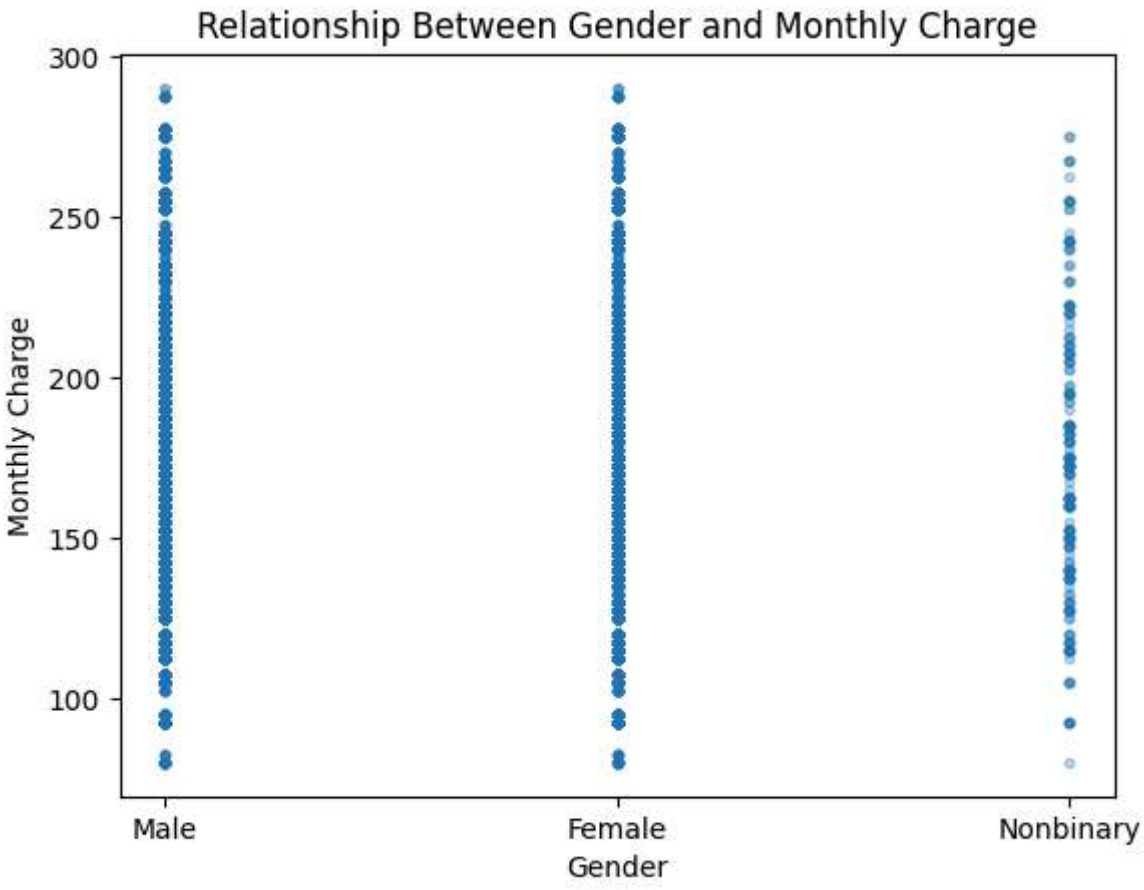
In [301…
```
plt.scatter(df.InternetService, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Internet Service and Monthly Charge')
plt.xlabel('Internet Service')
plt.ylabel('Monthly Charge')
```

Out[301…
```
Text(0, 0.5, 'Monthly Charge')
```



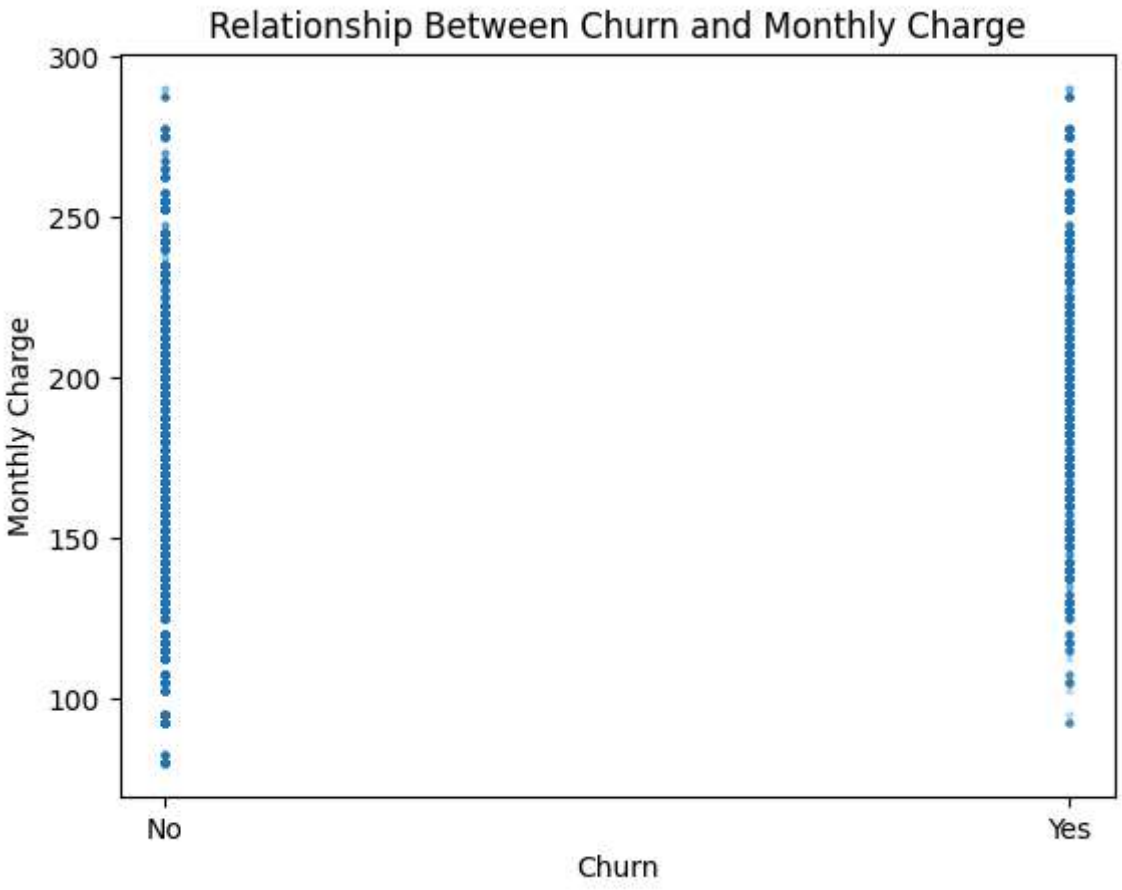
In [264…
```
plt.scatter(df.Gender, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Gender and Monthly Charge')
plt.xlabel('Gender')
plt.ylabel('Monthly Charge')
```

Out[264…
```
Text(0, 0.5, 'Monthly Charge')
```



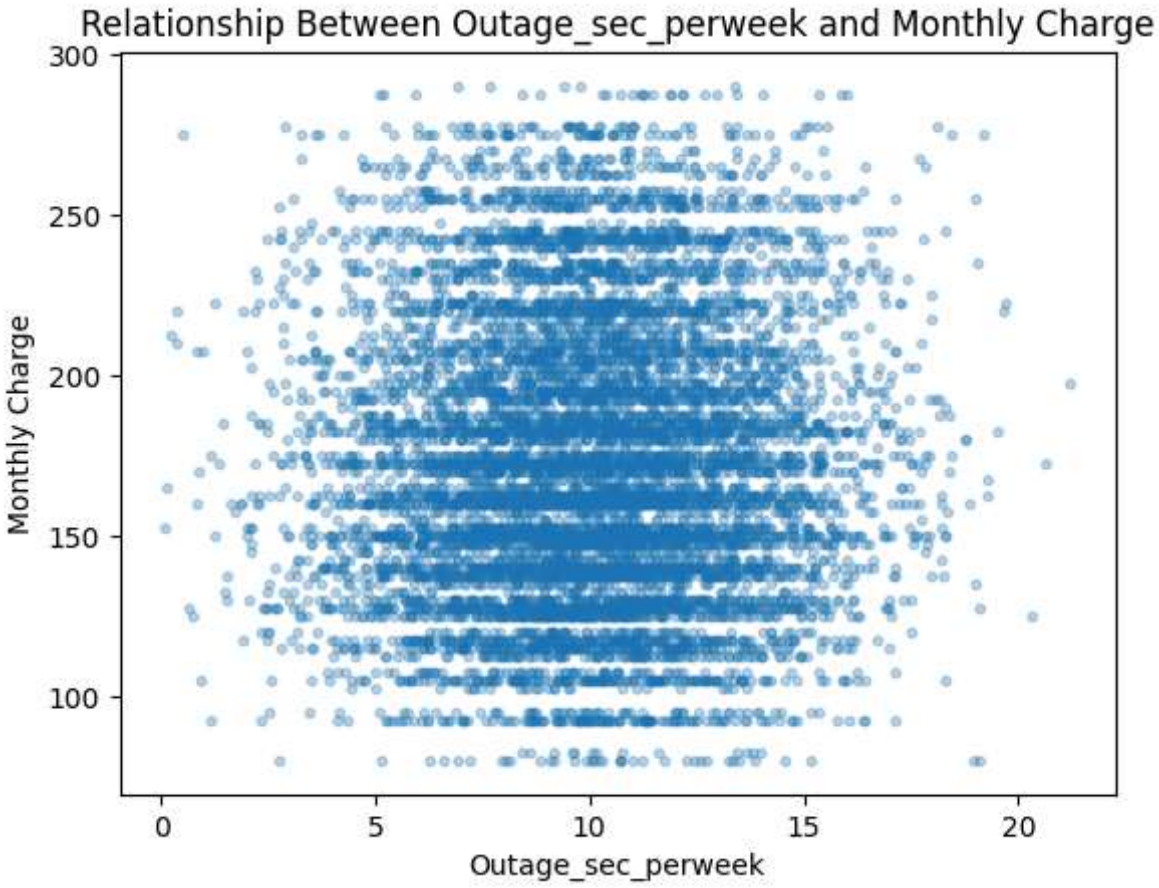
In [265…
```
plt.scatter(df.Churn, df.MonthlyCharge, s=4, alpha=0.2)
plt.title('Relationship Between Churn and Monthly Charge')
plt.xlabel('Churn')
plt.ylabel('Monthly Charge')
```

Out[265…
```
Text(0, 0.5, 'Monthly Charge')
```



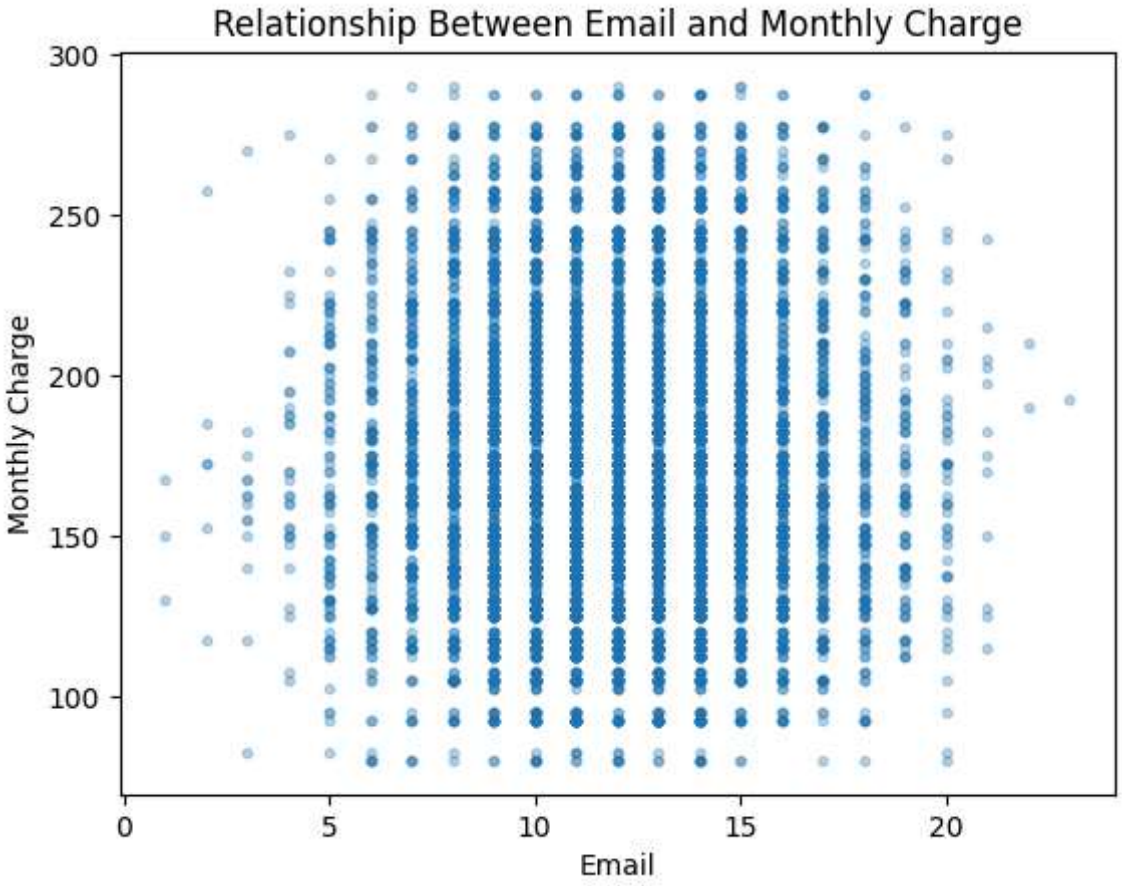
In [266…
```
plt.scatter(df.Outage_sec_perweek, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Outage_sec_perweek and Monthly Charge')
plt.xlabel('Outage_sec_perweek')
plt.ylabel('Monthly Charge')
```

Out[266…
```
Text(0, 0.5, 'Monthly Charge')
```



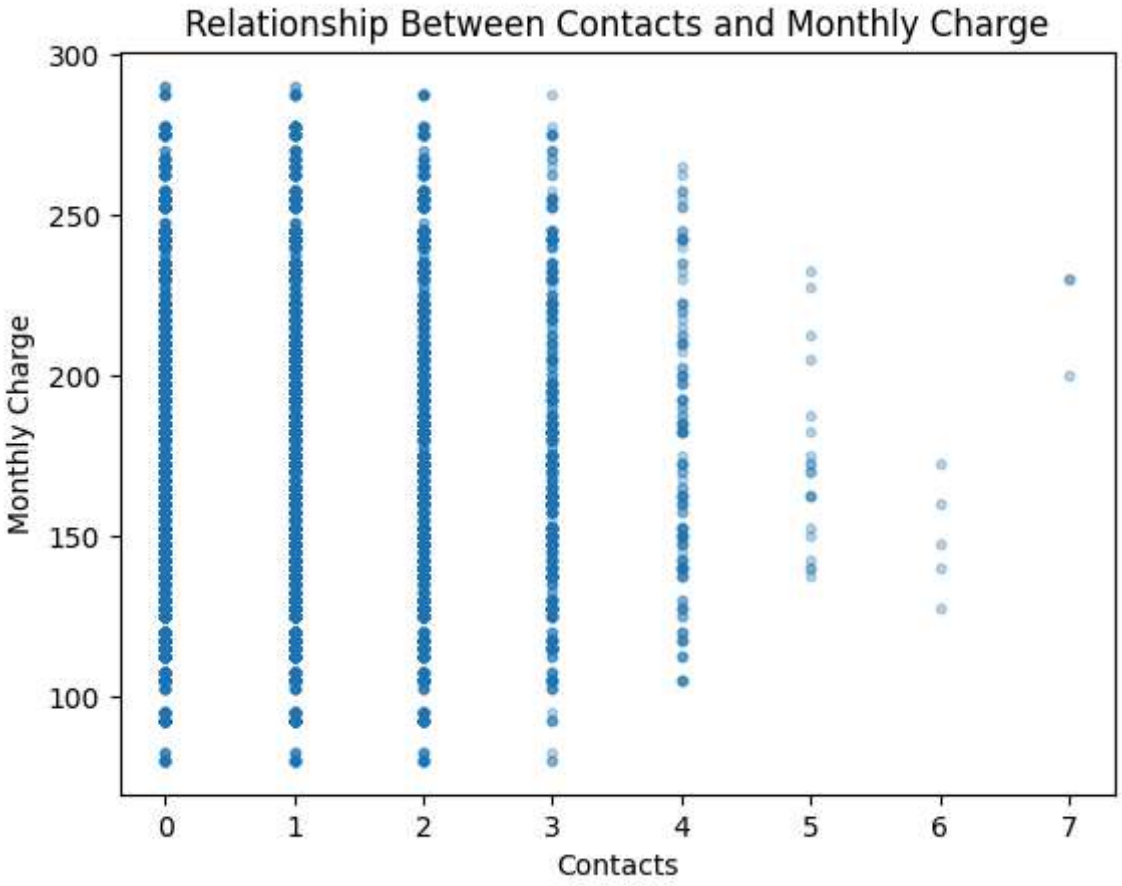
In [267…
```
plt.scatter(df.Email, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Email and Monthly Charge')
plt.xlabel('Email')
plt.ylabel('Monthly Charge')
```

Out[267…
```
Text(0, 0.5, 'Monthly Charge')
```



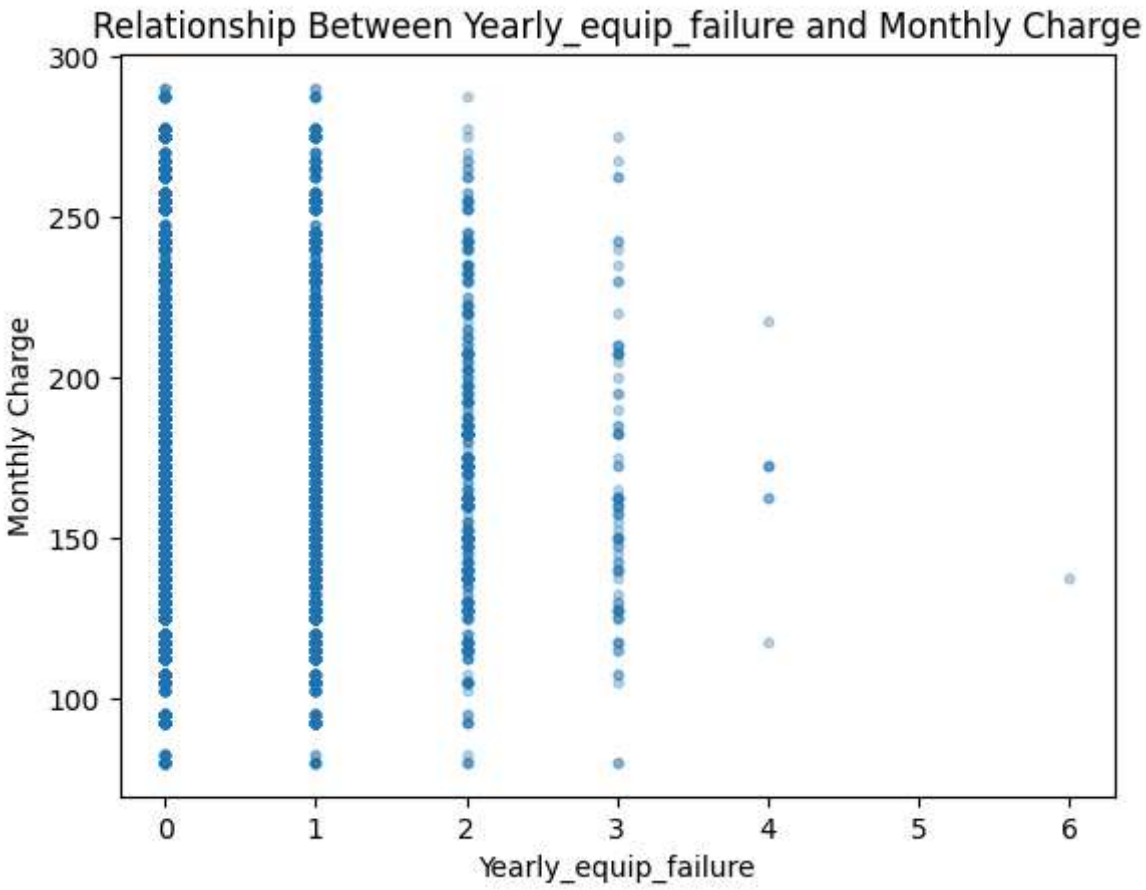
In [268…
```
plt.scatter(df.Contacts, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Contacts and Monthly Charge')
plt.xlabel('Contacts')
plt.ylabel('Monthly Charge')
```

Out[268…
```
Text(0, 0.5, 'Monthly Charge')
```



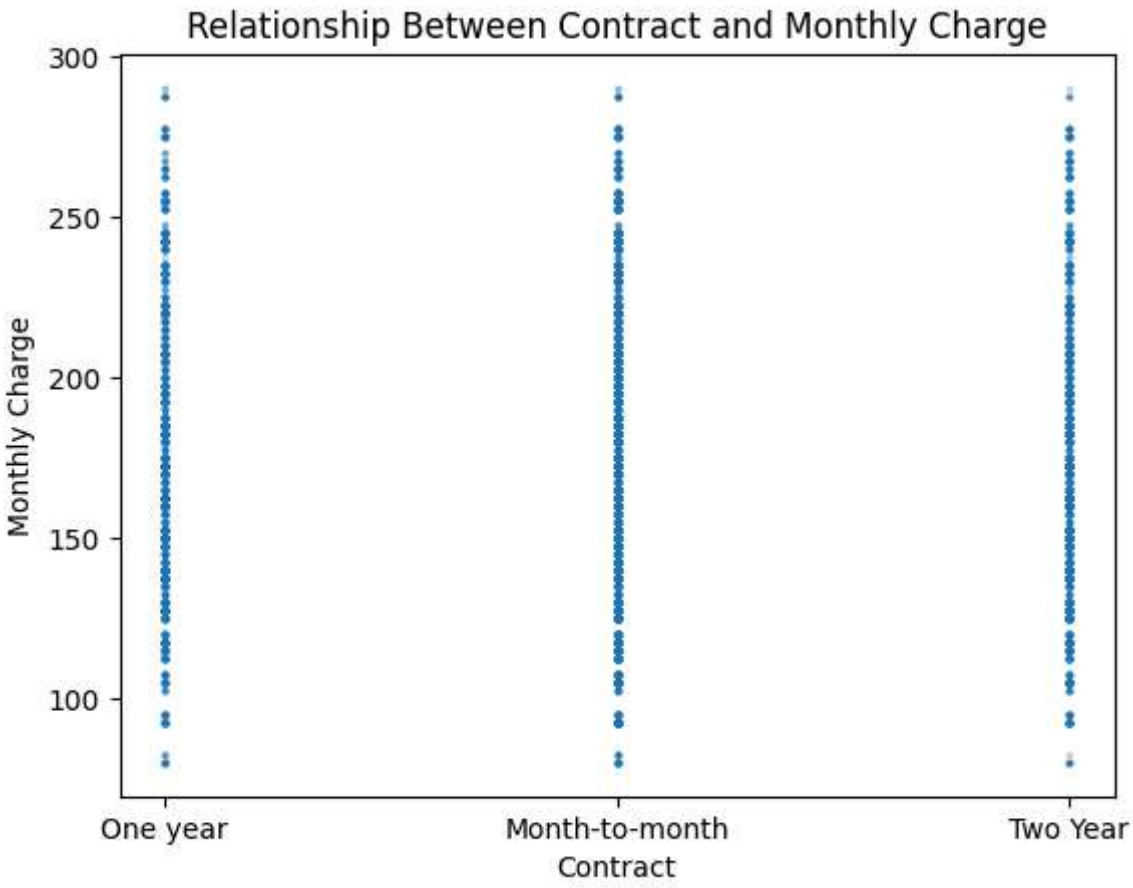
In [269…
```
plt.scatter(df.Yearly_equip_failure, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Yearly_equip_failure and Monthly Charge')
plt.xlabel('Yearly_equip_failure')
plt.ylabel('Monthly Charge')
```

Out[269…
```
Text(0, 0.5, 'Monthly Charge')
```



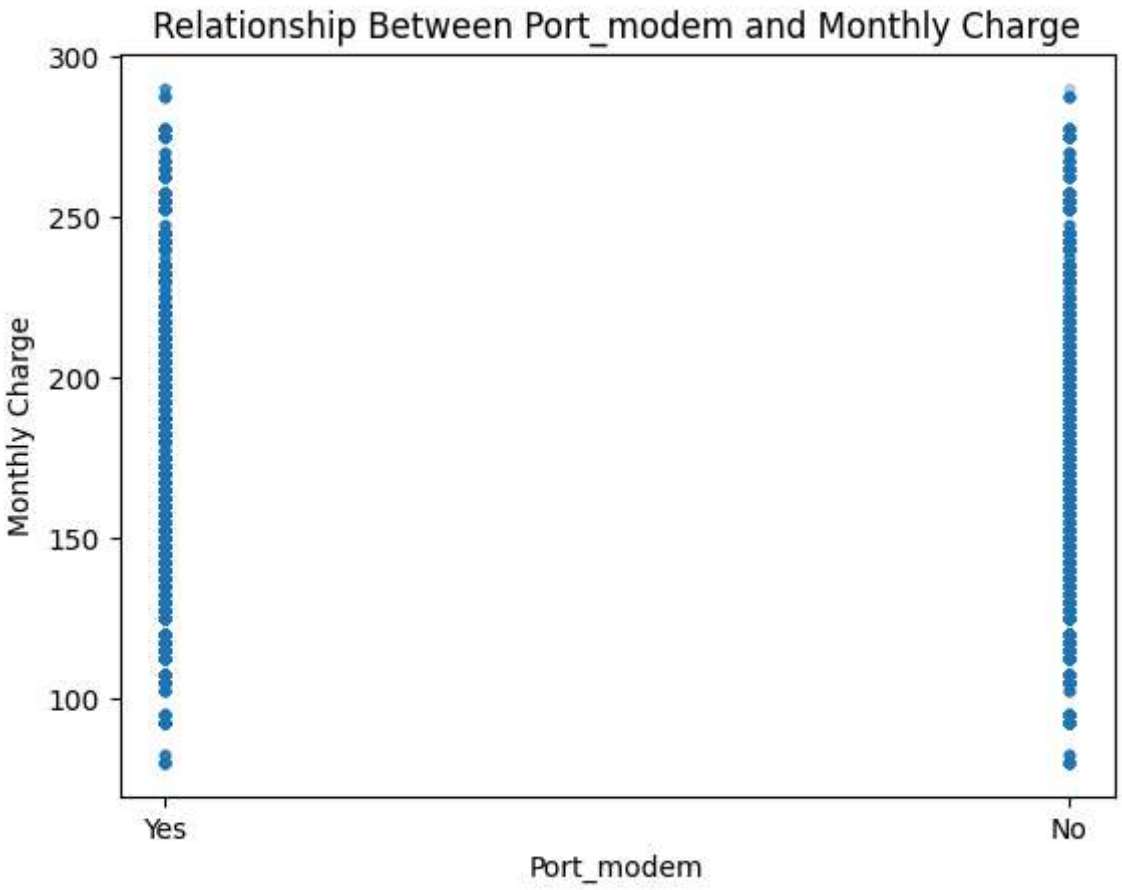
In [270…
```
plt.scatter(df.Contract, df.MonthlyCharge, s=3, alpha=0.2)
plt.title('Relationship Between Contract and Monthly Charge')
plt.xlabel('Contract')
plt.ylabel('Monthly Charge')
```

Out[270…
```
Text(0, 0.5, 'Monthly Charge')
```



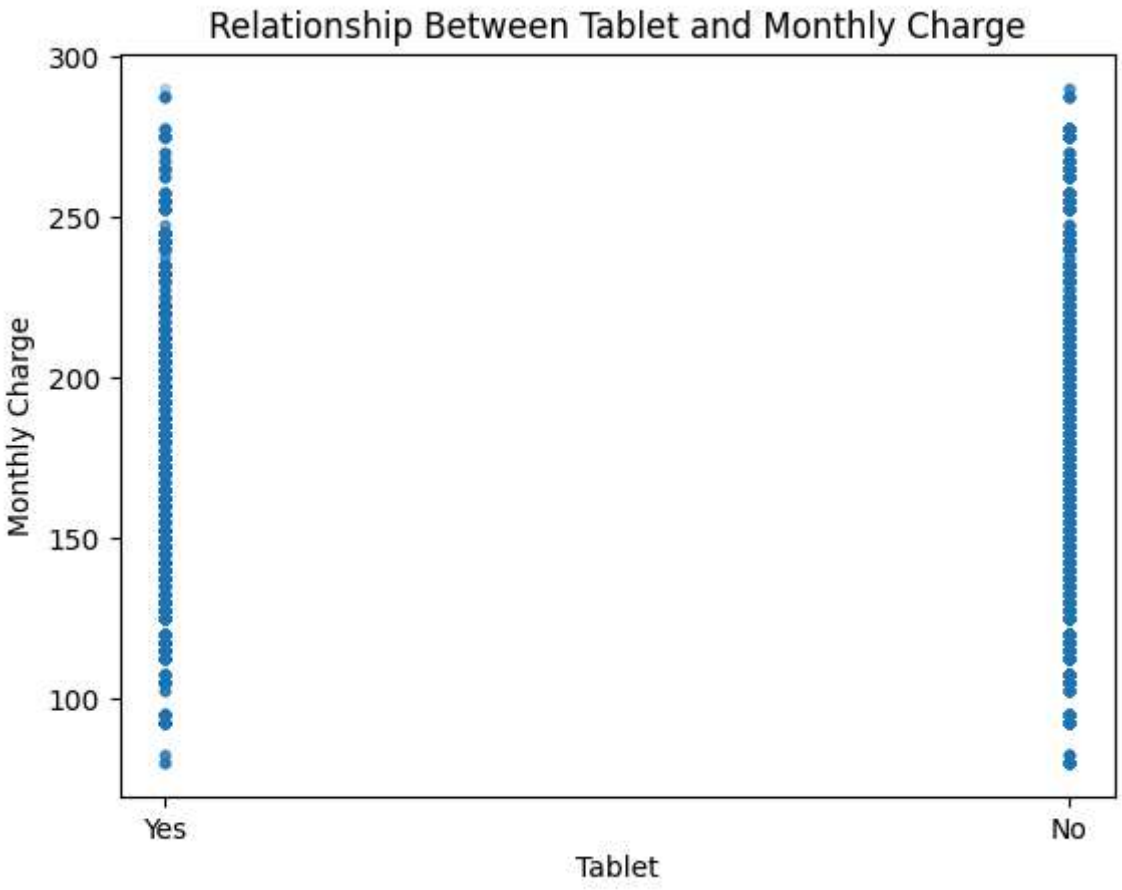
In [271…
```
plt.scatter(df.Port_modem, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Port_modem and Monthly Charge')
plt.xlabel('Port_modem')
plt.ylabel('Monthly Charge')
```

Out[271…
```
Text(0, 0.5, 'Monthly Charge')
```



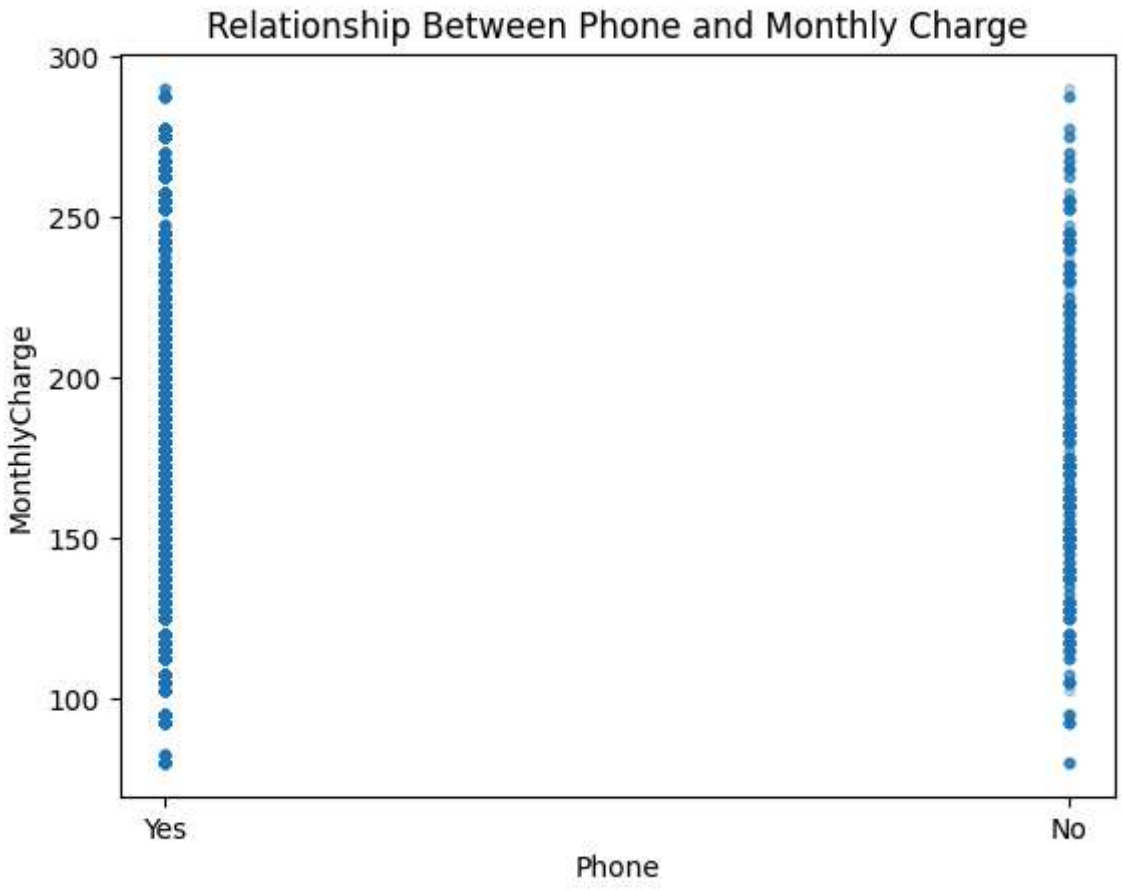
In [272…
```
plt.scatter(df.Tablet, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Tablet and Monthly Charge')
plt.xlabel('Tablet')
plt.ylabel('Monthly Charge')
```

Out[272…
```
Text(0, 0.5, 'Monthly Charge')
```



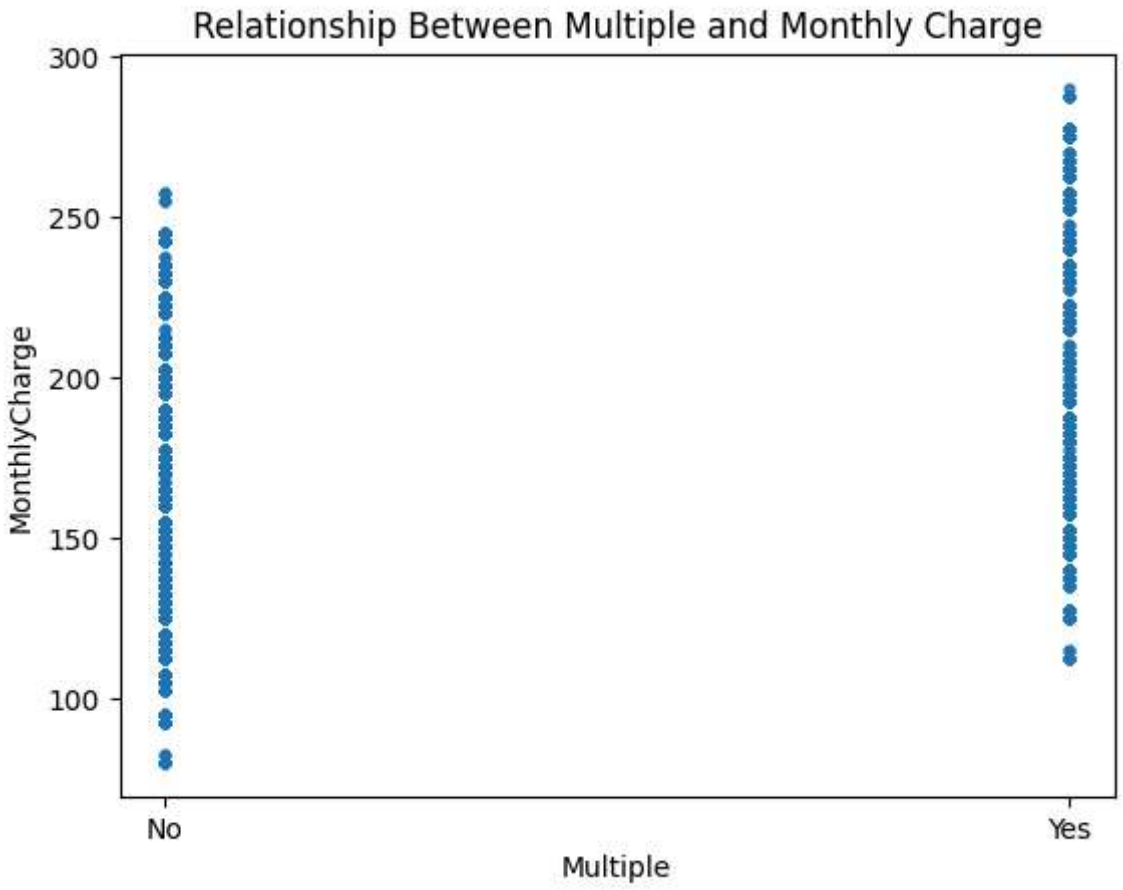
In [273…
```
plt.scatter(df.Phone, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Phone and Monthly Charge')
plt.xlabel('Phone')
plt.ylabel('MonthlyCharge')
```

Out[273…
```
Text(0, 0.5, 'MonthlyCharge')
```



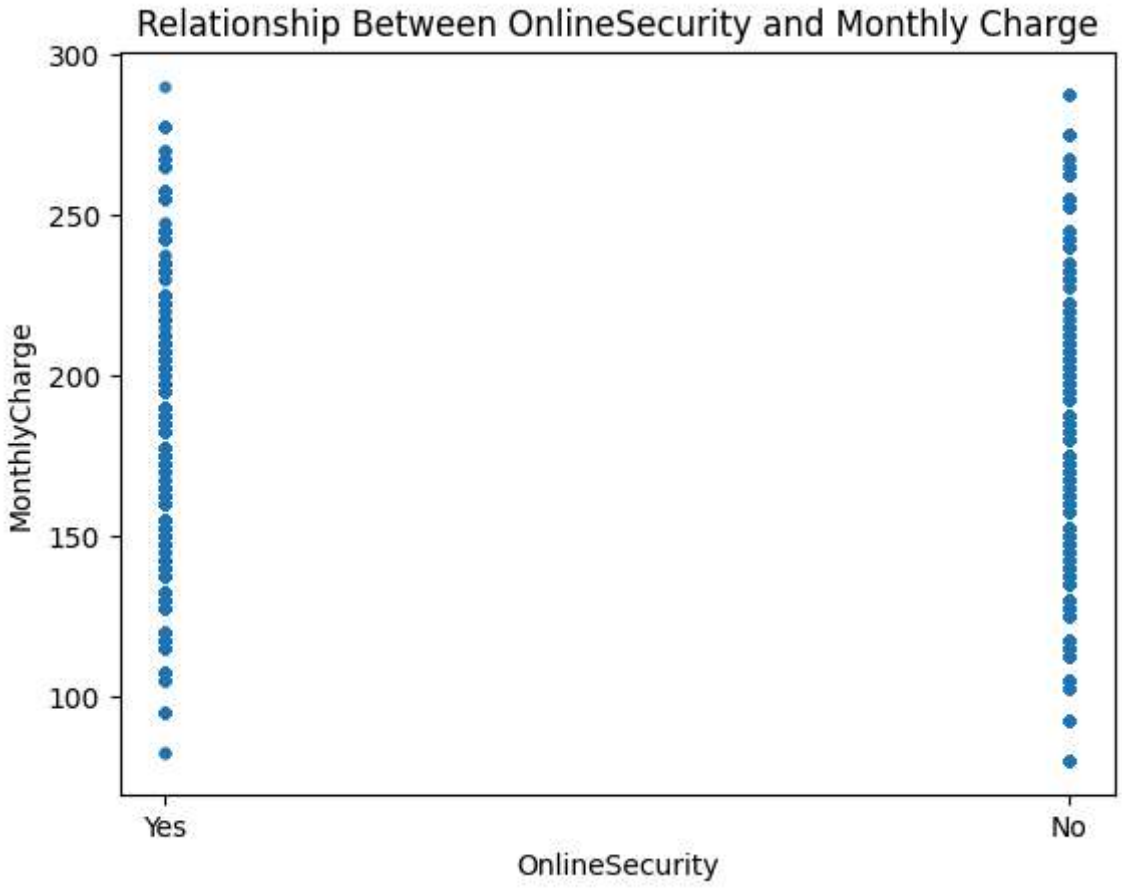
In [274…
```
plt.scatter(df.Multiple, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between Multiple and Monthly Charge')
plt.xlabel('Multiple')
plt.ylabel('MonthlyCharge')
```

Out[274…
```
Text(0, 0.5, 'MonthlyCharge')
```



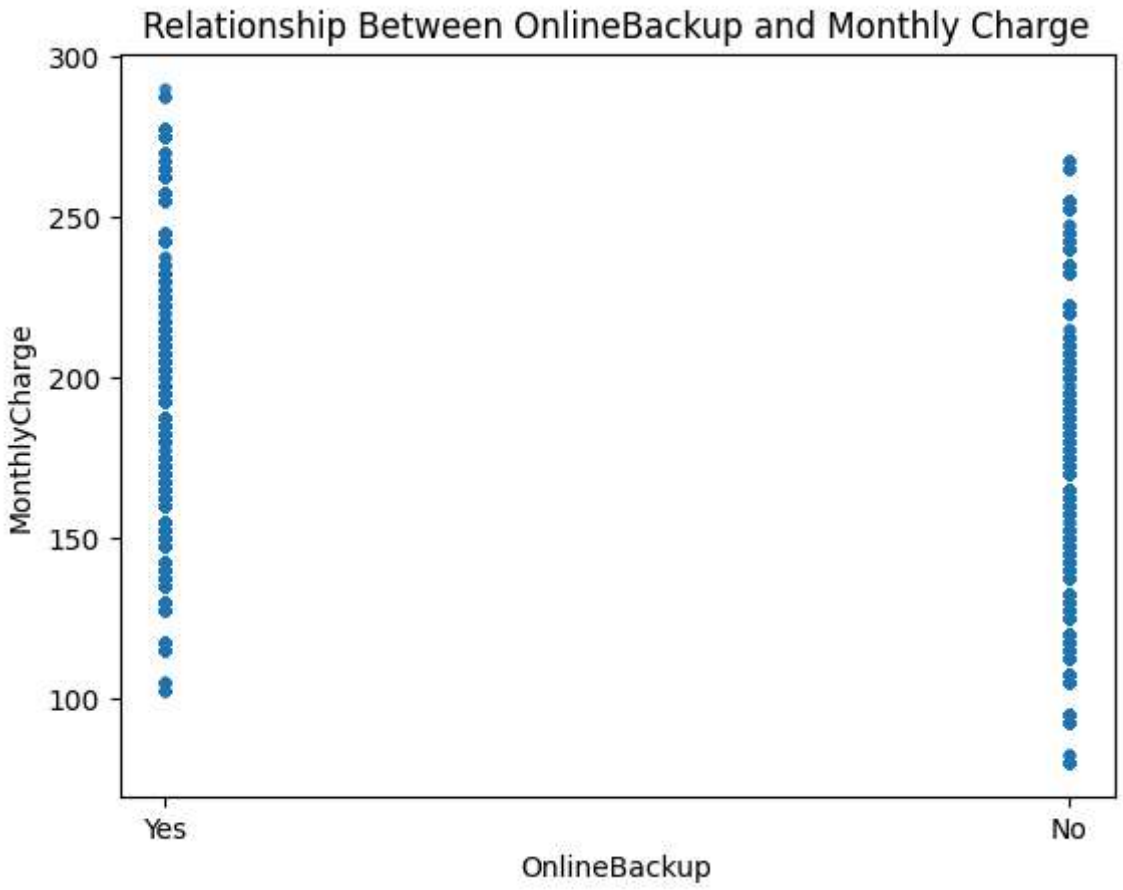
In [275…
```
plt.scatter(df.OnlineSecurity, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between OnlineSecurity and Monthly Charge')
plt.xlabel('OnlineSecurity')
plt.ylabel('MonthlyCharge')
```

Out[275…
```
Text(0, 0.5, 'MonthlyCharge')
```



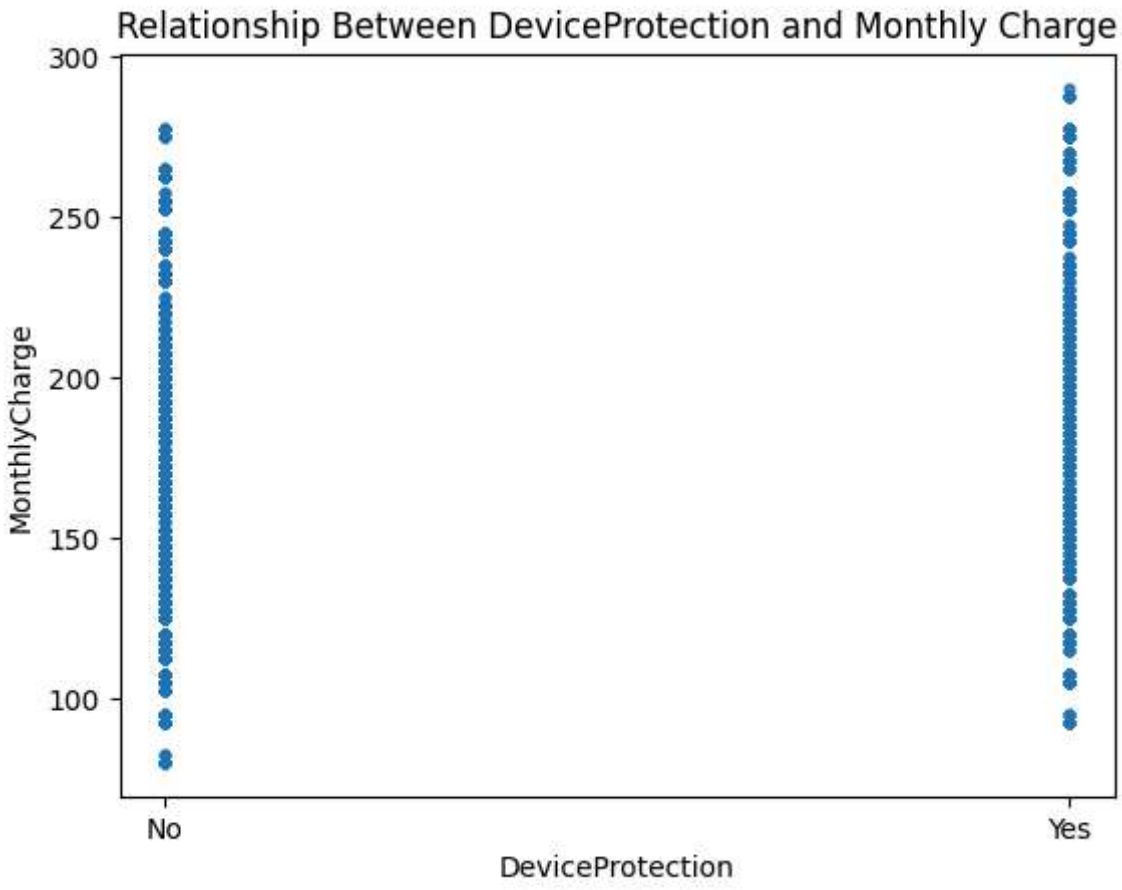
In [276…
```
plt.scatter(df.OnlineBackup, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between OnlineBackup and Monthly Charge')
plt.xlabel('OnlineBackup')
plt.ylabel('MonthlyCharge')
```

Out[276…
```
Text(0, 0.5, 'MonthlyCharge')
```



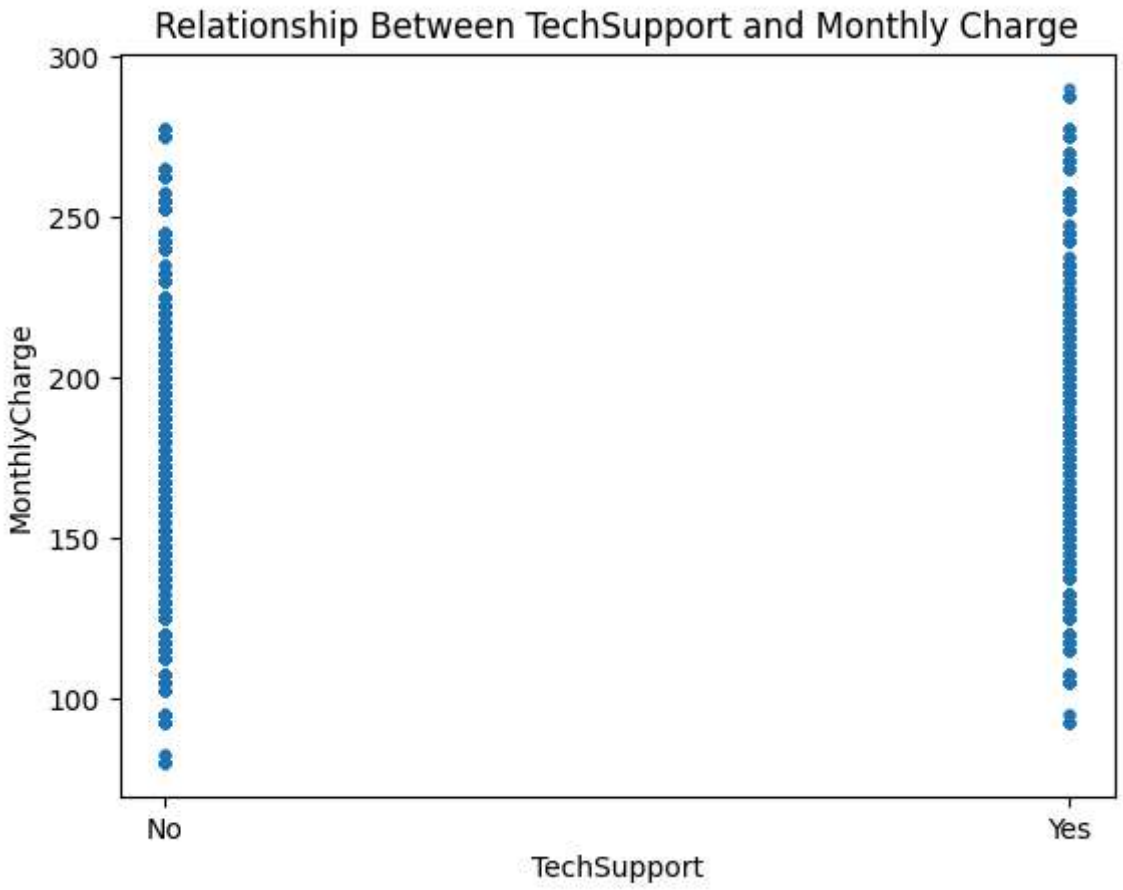
In [277…
```
plt.scatter(df.DeviceProtection, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between DeviceProtection and Monthly Charge')
plt.xlabel('DeviceProtection')
plt.ylabel('MonthlyCharge')
```

Out[277…
```
Text(0, 0.5, 'MonthlyCharge')
```



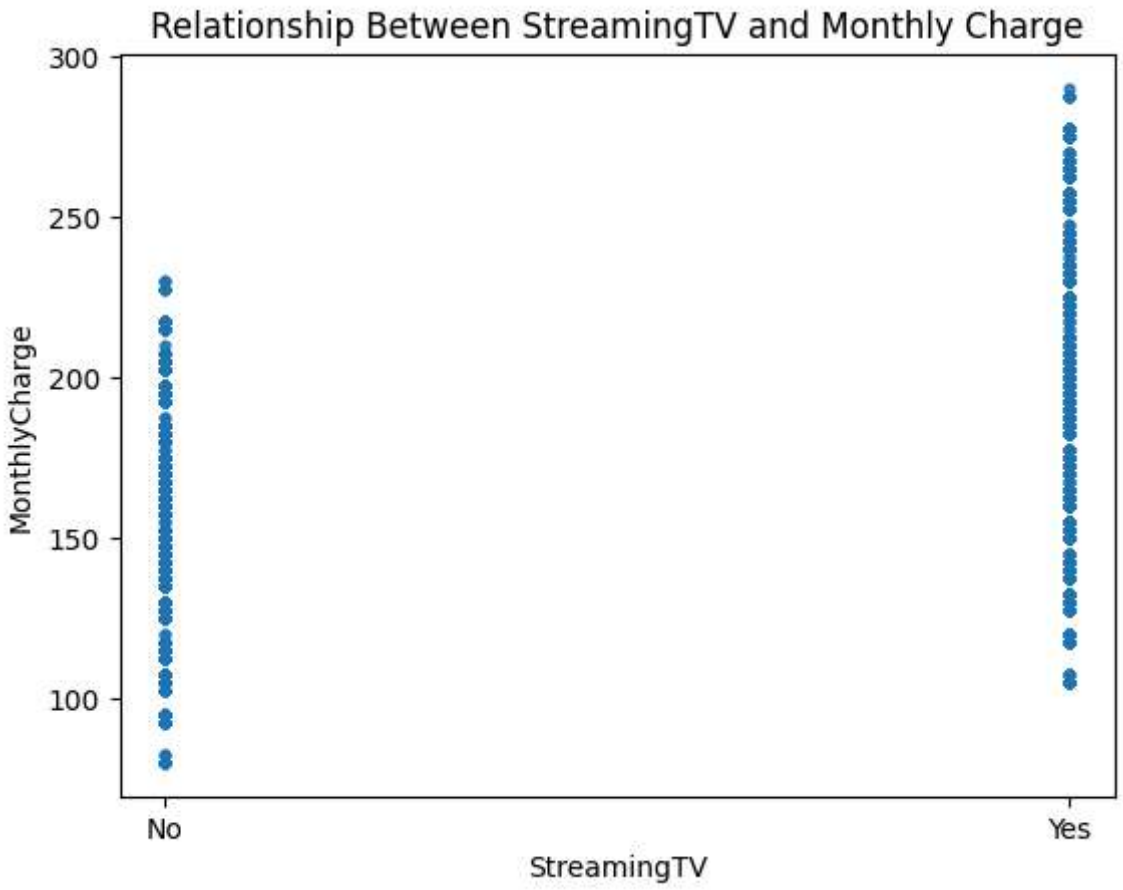
In [278…
```
plt.scatter(df.TechSupport, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between TechSupport and Monthly Charge')
plt.xlabel('TechSupport')
plt.ylabel('MonthlyCharge')
```

Out[278…
```
Text(0, 0.5, 'MonthlyCharge')
```



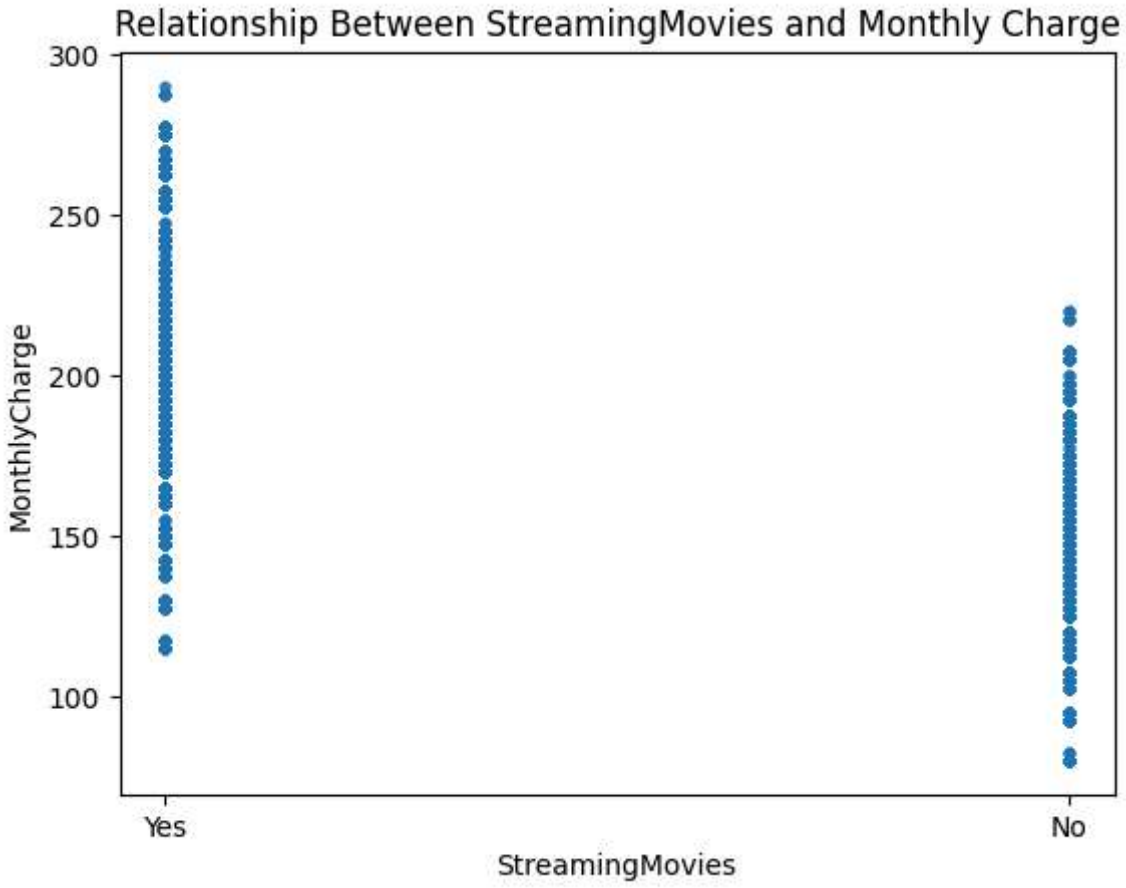
In [279…
```
plt.scatter(df.StreamingTV, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between StreamingTV and Monthly Charge')
plt.xlabel('StreamingTV')
plt.ylabel('MonthlyCharge')
```

Out[279…
```
Text(0, 0.5, 'MonthlyCharge')
```



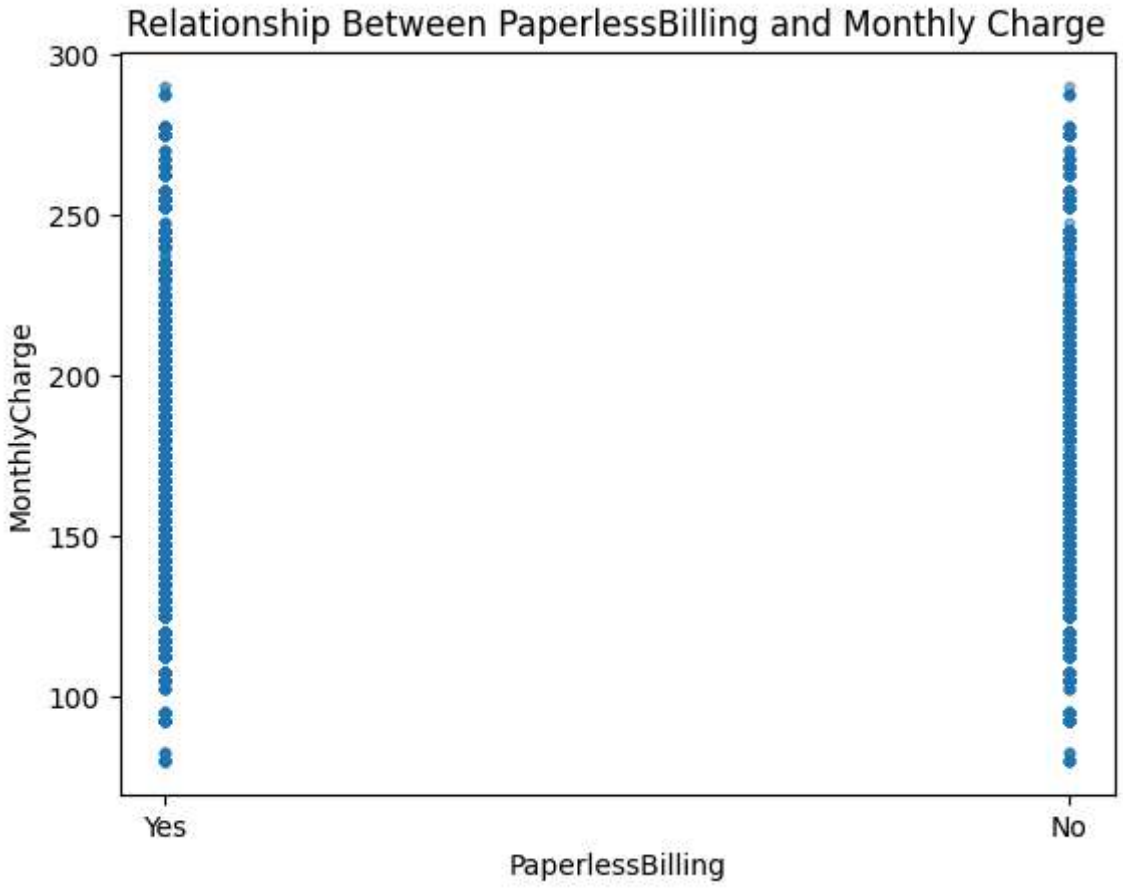
In [280…
```
plt.scatter(df.StreamingMovies, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between StreamingMovies and Monthly Charge')
plt.xlabel('StreamingMovies')
plt.ylabel('MonthlyCharge')
```

Out[280…
```
Text(0, 0.5, 'MonthlyCharge')
```



In [281…
```
plt.scatter(df.PaperlessBilling, df.MonthlyCharge, s=10, alpha=0.3)
plt.title('Relationship Between PaperlessBilling and Monthly Charge')
plt.xlabel('PaperlessBilling')
plt.ylabel('MonthlyCharge')
```

Out[281…
```
Text(0, 0.5, 'MonthlyCharge')
```



In [306…

```
df_clean = pd.get_dummies(df, columns = ['Gender', 'Marital', 'Contract', 'Internet
df_clean = df_clean.rename(columns=lambda x: x.replace(' ', '_'))
print(df_clean.head(3))
```

```
   Children  Age     Income  Churn  Outage_sec_perweek  Email  Contacts  \
0         0   68   28561.99      0            7.978323     10         0
1         1   27   21704.77      1           11.699080     12         0
2         4   50    9609.57      0           10.752800      9         0

   Yearly_equip_failure  Techie  Port_modem  ...  dmy_Male  dmy_Nonbinary  \
0                     1       0           1  ...      True          False
1                     1       1           0  ...     False          False
2                     1       1           1  ...     False          False

   dmy_Married  dmy_Never_Married  dmy_Separated  dmy_Widowed  dmy_One_year  \
0        False              False          False         True          True
1         True              False          False        False         False
2        False              False          False         True         False

   dmy_Two_Year  dmy_Fiber_Optic  dmy_None
0         False             True     False
1         False             True     False
2          True            False     False

[3 rows x 33 columns]
```

In [307…

```
df['Churn'] = df['Churn'].replace({'Yes': 1, 'No': 0}) #Replacing yes with 1 and no
df['Techie'] = df['Techie'].replace({'Yes': 1, 'No': 0})
```

```
df['Port_modem'] = df['Port_modem'].replace({'Yes': 1, 'No': 0})
df['Tablet'] = df['Tablet'].replace({'Yes': 1, 'No': 0})
df['Phone'] = df['Phone'].replace({'Yes': 1, 'No': 0})
df['Multiple'] = df['Multiple'].replace({'Yes': 1, 'No': 0})
df['OnlineSecurity'] = df['OnlineSecurity'].replace({'Yes': 1, 'No': 0})
df['OnlineBackup'] = df['OnlineBackup'].replace({'Yes': 1, 'No': 0})
df['DeviceProtection'] = df['DeviceProtection'].replace({'Yes': 1, 'No': 0})
df['TechSupport'] = df['TechSupport'].replace({'Yes': 1, 'No': 0})
df['StreamingMovies'] = df['StreamingMovies'].replace({'Yes': 1, 'No': 0})
df['StreamingTV'] = df['StreamingTV'].replace({'Yes': 1, 'No': 0})
df['PaperlessBilling'] = df['PaperlessBilling'].replace({'Yes': 1, 'No': 0})
```

In [323…

```
df_clean.corr()['MonthlyCharge']  #showing correlations between independent variabl
File_Path = r'C:\Users\Cali\Documents\Cleaned_Data_CSV_D208.csv'
df_clean.to_csv(File_Path, index = False)
```

In [310…

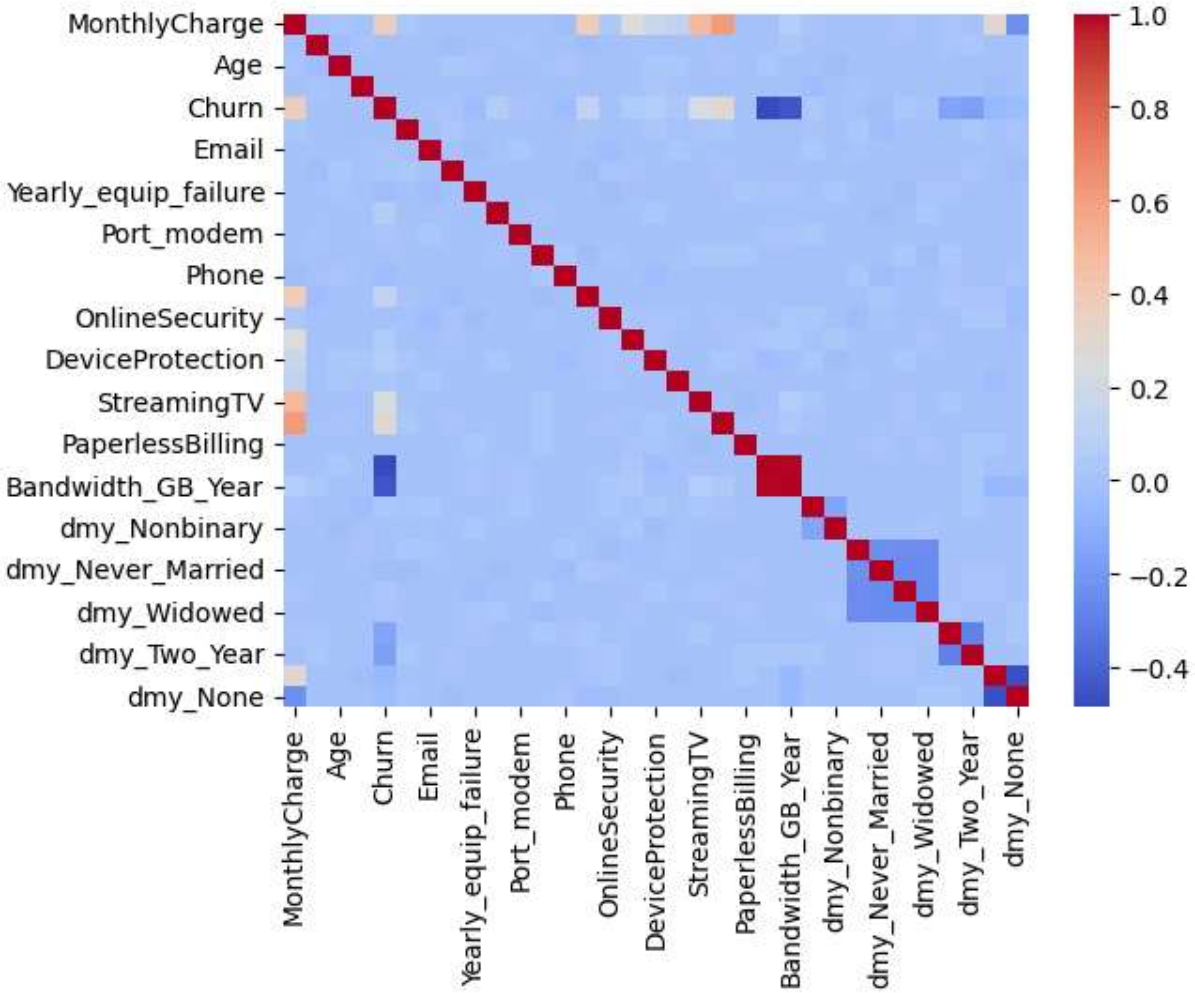
```
import statsmodels.formula.api as smf #Initial model
model_formula = 'MonthlyCharge ~ Children + Age + Income + Churn + Outage_sec_perwe
model = smf.ols(model_formula, data=df_clean).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          MonthlyCharge   R-squared:                       0.996
Model:                            OLS   Adj. R-squared:                  0.995
Method:                 Least Squares   F-statistic:                 6.912e+04
Date:                Sun, 07 Jul 2024   Prob (F-statistic):               0.00
Time:                        14:34:15   Log-Likelihood:                -24754.
No. Observations:               10000   AIC:                         4.957e+04
Df Residuals:                    9967   BIC:                         4.981e+04
Df Model:                          32
Covariance Type:            nonrobust
================================================================================
=========
                               coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---------
Intercept                   -88.1637      0.646   -136.569      0.000     -89.429
-86.898
dmy_Male[T.True]            -20.0815      0.091   -221.481      0.000     -20.259
-19.904
dmy_Nonbinary[T.True]         6.5391      0.196     33.396      0.000       6.155
6.923
dmy_Married[T.True]          -0.0190      0.091     -0.208      0.835      -0.198
0.160
dmy_Never_Married[T.True]    -0.0596      0.091     -0.657      0.511      -0.237
0.118
dmy_Separated[T.True]         0.0578      0.090      0.641      0.521      -0.119
0.234
dmy_Widowed[T.True]          -0.0298      0.090     -0.332      0.740      -0.206
0.146
dmy_One_year[T.True]          0.1533      0.077      1.988      0.047       0.002
0.304
dmy_Two_Year[T.True]          0.1724      0.074      2.344      0.019       0.028
0.317
dmy_Fiber_Optic[T.True]     147.7851      0.452    326.885      0.000     146.899
148.671
dmy_None[T.True]            115.0267      0.454    253.273      0.000     114.136
115.917
Children                     -9.4997      0.036   -264.432      0.000      -9.570
-9.429
Age                           1.0118      0.004    266.194      0.000       1.004
1.019
Income                    7.241e-07   1.02e-06      0.708      0.479   -1.28e-06
2.73e-06
Churn                         0.4639      0.091      5.082      0.000       0.285
0.643
Outage_sec_perweek            0.0057      0.010      0.585      0.559      -0.013
0.025
Email                        -0.0018      0.010     -0.191      0.848      -0.021
0.017
Contacts                     -0.0232      0.029     -0.795      0.427      -0.080
0.034
Yearly_equip_failure         -0.0064      0.045     -0.140      0.888      -0.095
0.083
Techie                        0.0187      0.077      0.242      0.809      -0.133
```

```
0.171
Port_modem                    0.0411      0.058      0.712      0.477      -0.072
0.154
Tablet                        0.0010      0.063      0.016      0.988      -0.123
0.125
Phone                        -0.0078      0.099     -0.078      0.938      -0.202
0.187
Multiple                     10.3047      0.097    106.622      0.000      10.115
10.494
OnlineSecurity              -20.7318      0.102   -203.456      0.000     -20.932
-20.532
OnlineBackup                 -6.5450      0.117    -56.007      0.000      -6.774
-6.316
DeviceProtection            -13.7524      0.109   -126.666      0.000     -13.965
-13.540
TechSupport                  11.1079      0.060    185.691      0.000      10.991
11.225
StreamingTV                 -28.3284      0.252   -112.442      0.000     -28.822
-27.835
StreamingMovies             -12.6747      0.233    -54.385      0.000     -13.132
-12.218
PaperlessBilling             -0.0807      0.059     -1.377      0.169      -0.196
0.034
Tenure                      -25.2955      0.089   -285.009      0.000     -25.469
-25.122
Bandwidth_GB_Year             0.3088      0.001    285.350      0.000       0.307
0.311
==============================================================================
Omnibus:                    48640.338   Durbin-Watson:                   2.012
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1085.432
Skew:                          -0.028   Prob(JB):                    2.00e-236
Kurtosis:                       1.387   Cond. No.                     1.65e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 1.65e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

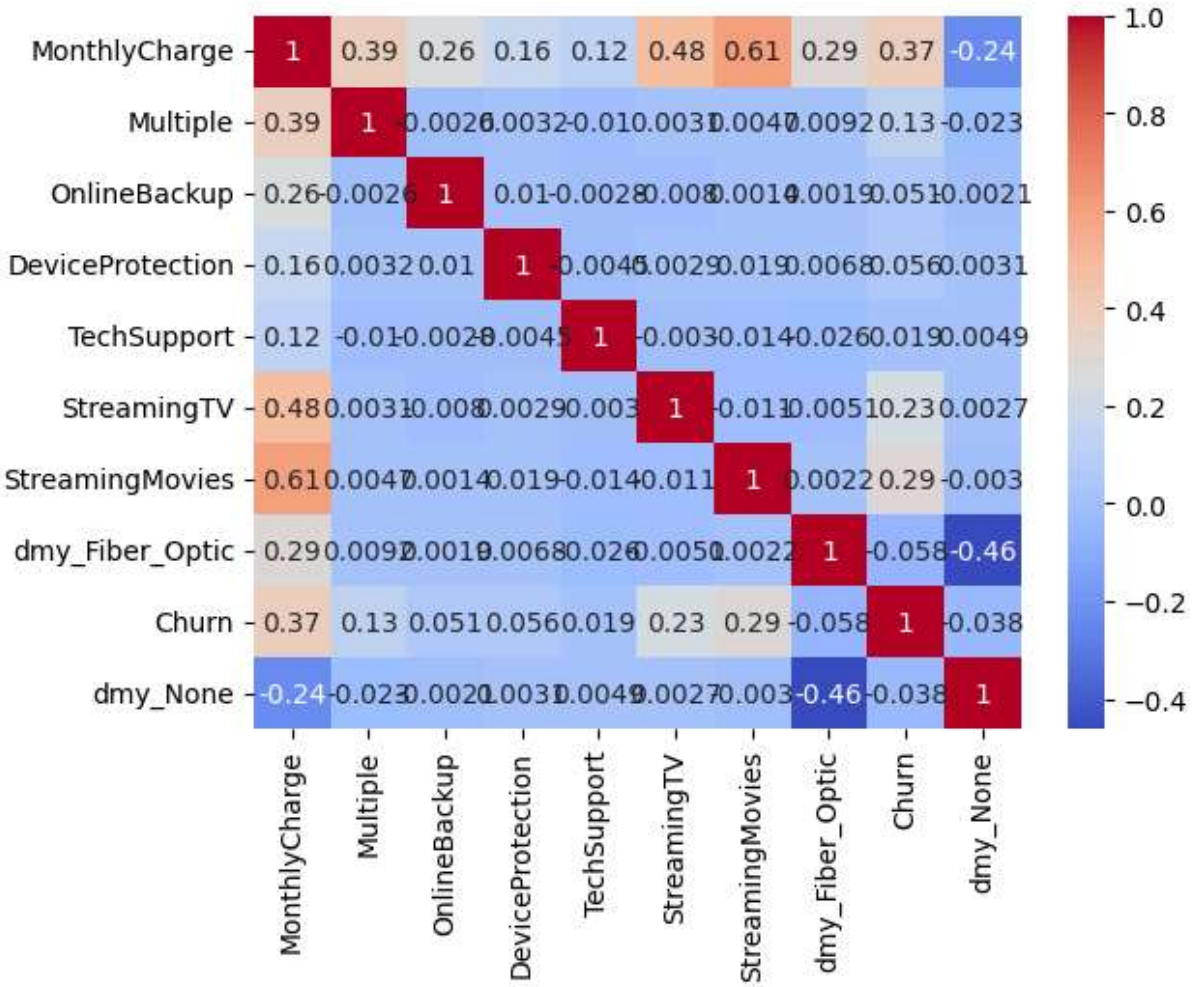
In [312…

```
# Create dataframe for heatmap bivariate analysis of correlation
cols = [
    'MonthlyCharge','Children', 'Age', 'Income', 'Churn', 'Outage_sec_perweek', 'Em
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'Bandwidth_GB_Year', 'dmy_Male
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
]
df_bivariate = df_clean[cols]
ax = sns.heatmap(df_bivariate.corr(), annot=False, cmap="coolwarm")

plt.show()
```



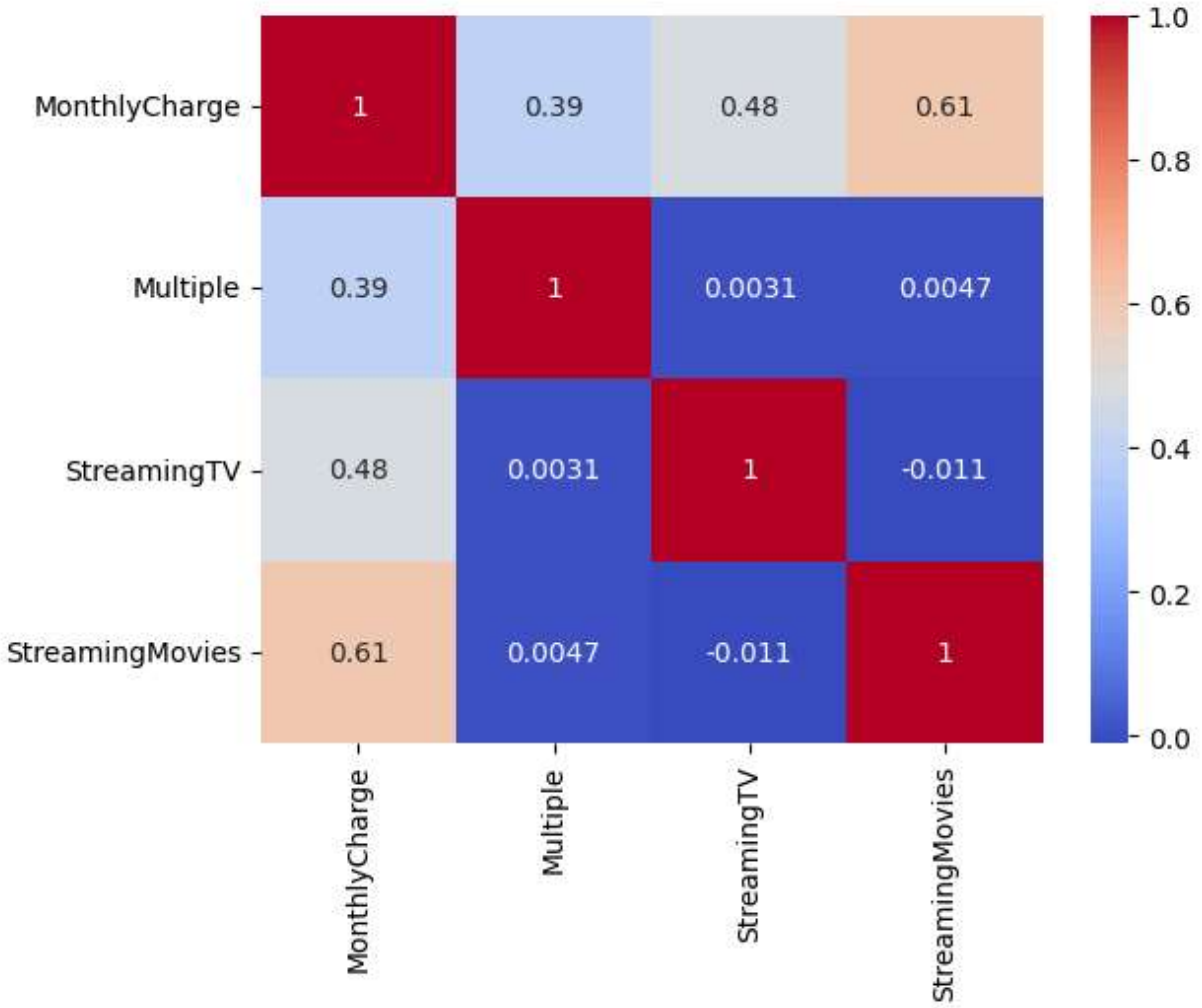
In [315…

```
lesscols = [
    'MonthlyCharge', 'Multiple',
    'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
    'StreamingMovies', 'dmy_Fiber_Optic', 'Churn', 'dmy_None'
]
df_bivariate = df_clean[lesscols]
ax = sns.heatmap(df_bivariate.corr(), annot=True, cmap="coolwarm")

plt.show()
```



In [316…

```
lesscols = [
    'MonthlyCharge', 'Multiple', 'StreamingTV',
    'StreamingMovies'
]
df_bivariate = df_clean[lesscols]
ax = sns.heatmap(df_bivariate.corr(), annot=True, cmap="coolwarm")

plt.show() #final variables, no multicollinearity!
```



In [ ]:

In [317…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.formula.api import ols
model_formula = 'MonthlyCharge ~ StreamingMovies + StreamingTV + Multiple'
model_reduced = smf.ols(model_formula, data=df_clean).fit() #Reduced Model
print(model_reduced.summary())

# modify figure size
fig = plt.figure(figsize=(14, 8))

# creating regression plots
fig = sm.graphics.plot_regress_exog(model_reduced, 'StreamingTV', fig=fig) #regress
```

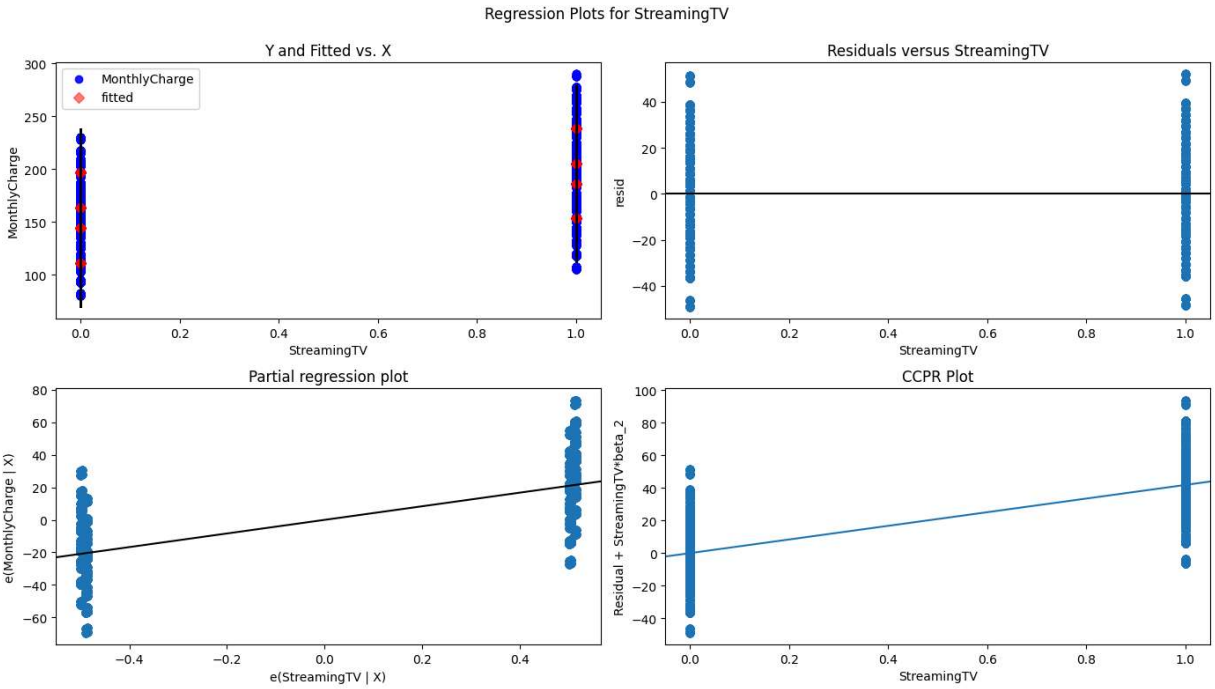
OLS Regression Results

Dep. Variable:	MonthlyCharge	R-squared:	0.754
Model:	OLS	Adj. R-squared:	0.754
Method:	Least Squares	F-statistic:	1.023e+04
Date:	Sun, 07 Jul 2024	Prob (F-statistic):	0.00
Time:	14:36:28	Log-Likelihood:	-44770.
No. Observations:	10000	AIC:	8.955e+04
Df Residuals:	9996	BIC:	8.958e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	111.1506	0.414	268.177	0.000	110.338	111.963
StreamingMovies	52.5254	0.426	123.318	0.000	51.690	53.360
StreamingTV	41.8768	0.426	98.332	0.000	41.042	42.712
Multiple	32.8738	0.427	76.965	0.000	32.037	33.711

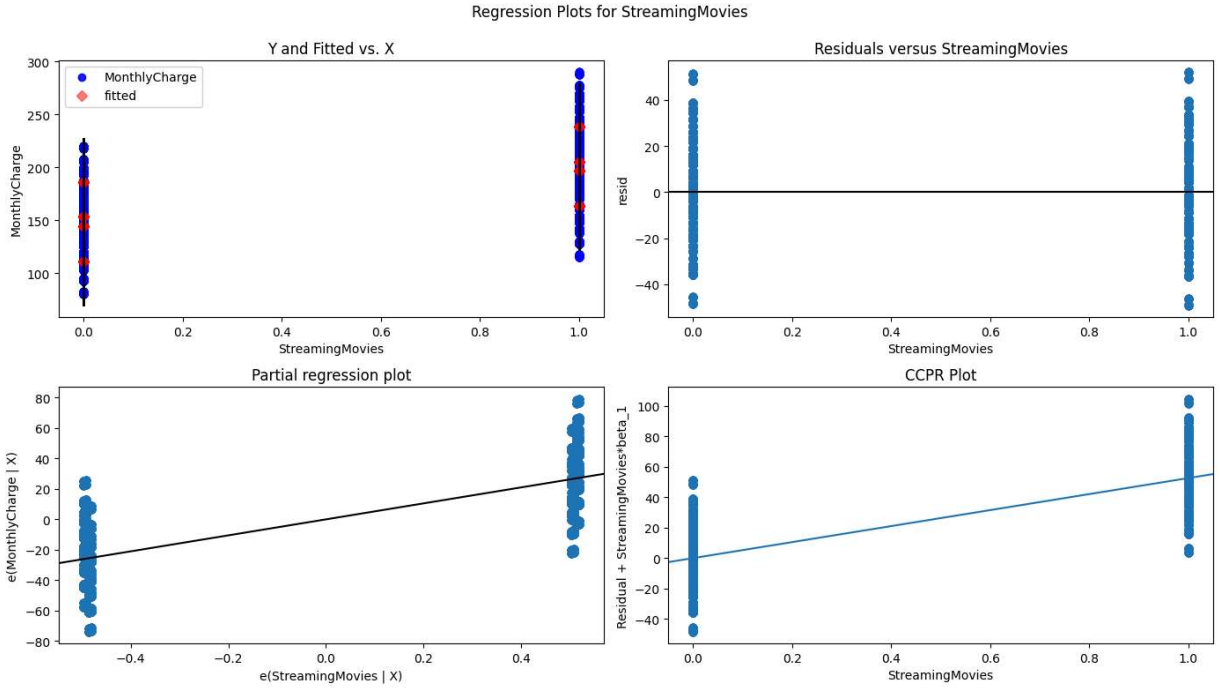
Omnibus:	185.821	Durbin-Watson:	1.981
Prob(Omnibus):	0.000	Jarque-Bera (JB):	102.213
Skew:	-0.005	Prob(JB):	6.38e-23
Kurtosis:	2.505	Cond. No.	3.62

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.

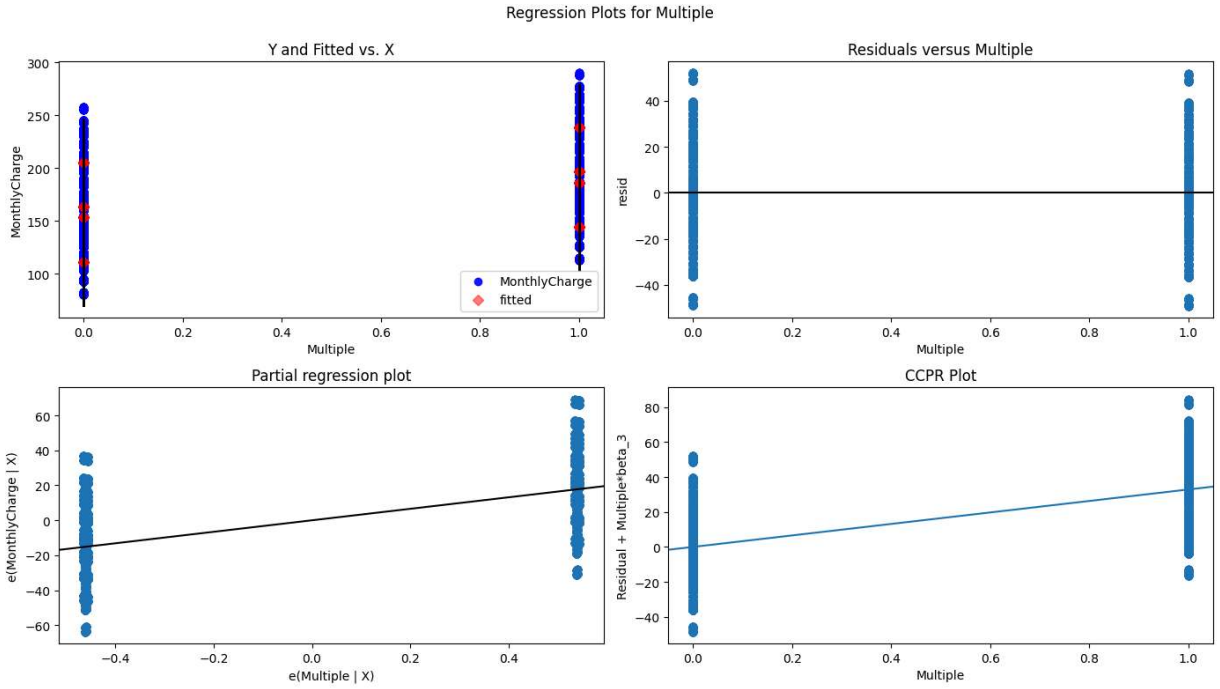


In [318…

```
fig2 = plt.figure(figsize=(14, 8))

fig2 = sm.graphics.plot_regress_exog(model_reduced, 'StreamingMovies', fig=fig2)
```



In [294…
```
fig3 = plt.figure(figsize=(14, 8))

fig3 = sm.graphics.plot_regress_exog(model_reduced, 'Multiple', fig=fig3)
```



In [ ]: