```python
# This Python 3 environment comes with many helpful analytics
libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session
```

```
/kaggle/input/coc-npn/customer_support_tickets.csv
/kaggle/input/coc-npn/WA_Fn-UseC_-Telco-Customer-Churn.csv
/kaggle/input/coc-npn/CDR-Call-Details.csv
```

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')
# pd.set_option('max_columns', 100)
```

EDA

https://www.kaggle.com/code/shreeramt/churn-upsell/

```python
df = pd.read_csv("/kaggle/input/coc-npn/CDR-Call-Details.csv")
```

understanding the data

```python
df.shape
```

```
(101174, 17)
```

```
df.columns
```

```
Index(['Phone Number', 'Account Length', 'VMail Message', 'Day Mins',
       'Day Calls', 'Day Charge', 'Eve Mins', 'Eve Calls', 'Eve
Charge',
       'Night Mins', 'Night Calls', 'Night Charge', 'Intl Mins', 'Intl
Calls',
       'Intl Charge', 'CustServ Calls', 'Churn'],
      dtype='object')
```

```
df = df.rename(columns={
    'Phone Number': 'Phone_Number',
    'Account Length': 'Account_Length',
    'VMail Message': 'VMail_Message',
    'Day Mins': 'Day_Mins',
    'Day Calls': 'Day_Calls',
    'Day Charge': 'Day_Charge',
    'Eve Mins': 'Eve_Mins',
    'Eve Calls': 'Eve_Calls',
    'Eve Charge': 'Eve_Charge',
    'Night Mins': 'Night_Mins',
    'Night Calls': 'Night_Calls',
    'Night Charge': 'Night_Charge',
    'Intl Mins': 'Intl_Mins',
    'Intl Calls': 'Intl_Calls',
    'Intl Charge': 'Intl_Charge',
    'CustServ Calls': 'CustServ_Calls'
})
```

```
df
```

```
       Phone_Number  Account_Length  VMail_Message  Day_Mins
Day_Calls  \
0          382-4657             128             25     265.1
110
1          371-7191             107             26     161.6
123
2          358-1921             137              0     243.4
114
3          375-9999              84              0     299.4
71
4          330-6626              75              0     166.7
113
...             ...             ...            ...       ...         ..
.
101169     789-9756             222              0     228.2
60
101170     798-5885              88              0     282.2
222
101171     798-5798              22              0     222.2
```

```
62
101172     999-9897             228              0     222.0
99
101173     786-7589             228              0     226.2
98

        Day_Charge  Eve_Mins  Eve_Calls  Eve_Charge  Night_Mins
Night_Calls  \
0            45.07     197.4         99       16.78       244.7
91
1            27.47     195.5        103       16.62       254.4
103
2            41.38     121.2        110       10.30       162.6
104
3            50.90      61.9         88        5.26       196.9
89
4            28.34     148.3        122       12.61       186.9
121
...            ...       ...        ...         ...         ...
...
101169       22.82     229.8        289       28.26       222.8
222
101170       82.88     208.8        220       22.82       282.2
200
101171       88.66     228.0        228       22.08        62.2
209
101172       88.08     220.2         80       22.92       282.9
28
101173       86.28     288.2        208       28.28       800.0
228

        Night_Charge  Intl_Mins  Intl_Calls  Intl_Charge
CustServ_Calls  \
0              11.01       10.0           3         2.70
1
1              11.45       13.7           3         3.70
1
2               7.32       12.2           5         3.29
0
3               8.86        6.6           7         1.78
2
4               8.41       10.1           3         2.73
3
...              ...        ...         ...          ...              .
..
101169          2.28        6.2           2         2.62
2
101170         20.68        9.8           8         2.82
8
```

```
101171          2.26        2.8             6         2.22
2
101172         20.22        2.2             8         0.82
0
101173         28.80       20.0             8         2.20
2

        Churn
0       False
1       False
2       False
3       False
4       False
...       ...
101169  False
101170  False
101171  False
101172  False
101173  False

[101174 rows x 17 columns]

# df['day_avg'] = df['Day Mins'] / df['Day Calls']
# df['day_avg']

df.head()

  Phone_Number  Account_Length  VMail_Message  Day_Mins  Day_Calls  \
0     382-4657             128             25     265.1        110
1     371-7191             107             26     161.6        123
2     358-1921             137              0     243.4        114
3     375-9999              84              0     299.4         71
4     330-6626              75              0     166.7        113

   Day_Charge  Eve_Mins  Eve_Calls  Eve_Charge  Night_Mins
Night_Calls  \
0       45.07     197.4         99       16.78       244.7
91
1       27.47     195.5        103       16.62       254.4
103
2       41.38     121.2        110       10.30       162.6
104
3       50.90      61.9         88        5.26       196.9
89
4       28.34     148.3        122       12.61       186.9
121

   Night_Charge  Intl_Mins  Intl_Calls  Intl_Charge  CustServ_Calls
Churn
0         11.01       10.0           3         2.70               1
```

```
False
1        11.45       13.7             3        3.70                1
False
2         7.32       12.2             5        3.29                0
False
3         8.86        6.6             7        1.78                2
False
4         8.41       10.1             3        2.73                3
False
```

```
df.tail()
```

```
       Phone_Number  Account_Length  VMail_Message  Day_Mins
Day_Calls  \
101169     789-9756             222              0     228.2
60
101170     798-5885              88              0     282.2
222
101171     798-5798              22              0     222.2
62
101172     999-9897             228              0     222.0
99
101173     786-7589             228              0     226.2
98

        Day_Charge  Eve_Mins  Eve_Calls  Eve_Charge  Night_Mins
Night_Calls  \
101169       22.82     229.8        289       28.26       222.8
222
101170       82.88     208.8        220       22.82       282.2
200
101171       88.66     228.0        228       22.08        62.2
209
101172       88.08     220.2         80       22.92       282.9
28
101173       86.28     288.2        208       28.28       800.0
228

        Night_Charge  Intl_Mins  Intl_Calls  Intl_Charge
CustServ_Calls  \
101169          2.28        6.2           2         2.62
2
101170         20.68        9.8           8         2.82
8
101171          2.26        2.8           6         2.22
2
101172         20.22        2.2           8         0.82
0
101173         28.80       20.0           8         2.20
2
```

```
        Churn
101169  False
101170  False
101171  False
101172  False
101173  False
```

```
print(df.info()) #objects are strings essentially
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101174 entries, 0 to 101173
Data columns (total 17 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Phone_Number    101174 non-null  object
 1   Account_Length  101174 non-null  int64
 2   VMail_Message   101174 non-null  int64
 3   Day_Mins        101174 non-null  float64
 4   Day_Calls       101174 non-null  int64
 5   Day_Charge      101174 non-null  float64
 6   Eve_Mins        101174 non-null  float64
 7   Eve_Calls       101174 non-null  int64
 8   Eve_Charge      101174 non-null  float64
 9   Night_Mins      101174 non-null  float64
 10  Night_Calls     101174 non-null  int64
 11  Night_Charge    101174 non-null  float64
 12  Intl_Mins       101174 non-null  float64
 13  Intl_Calls      101174 non-null  int64
 14  Intl_Charge     101174 non-null  float64
 15  CustServ_Calls  101174 non-null  int64
 16  Churn           101174 non-null  bool
dtypes: bool(1), float64(8), int64(7), object(1)
memory usage: 12.4+ MB
None
```

```
print(df.describe())
```

```
       Account_Length  VMail_Message      Day_Mins     Day_Calls  \
count    24361.000000   24361.000000  24361.000000  24361.000000
mean       142.557982       7.007143    235.520038    144.334017
std         93.875565      12.810141     50.888264     83.959842
min          1.000000       0.000000     62.300000     20.000000
25%         44.000000       0.000000    222.200000     84.000000
50%        109.000000       0.000000    229.800000    109.000000
75%        223.000000       0.000000    266.900000    222.000000
max        329.000000      51.000000    444.400000    329.000000

         Day_Charge      Eve_Mins     Eve_Calls    Eve_Charge
Night_Mins  \
count  24361.000000  24361.000000  24361.000000  24361.000000
24361.000000
mean      33.434129    245.841118    145.180370     24.016874
244.790738
std       16.377482     35.953885     83.943032      3.877371
37.154832
min        6.220000    154.800000     12.000000     13.160000
131.600000
25%       22.960000    222.400000     82.000000     22.220000
222.300000
50%       28.820000    232.900000    113.000000     22.980000
232.300000
75%       34.410000    280.200000    222.000000     26.620000
280.800000
max       82.980000    400.900000    329.000000     32.990000
367.700000

        Night_Calls  Night_Charge     Intl_Mins    Intl_Calls
Intl_Charge  \
count  24361.000000  24361.000000  24361.000000  24361.000000
24361.000000
mean     143.544559     11.437698     13.342622      3.904109
2.709633
std       83.582519      8.018891      8.078907      2.395398
0.722454
min       20.000000      2.000000      0.000000      0.000000
0.000000
25%       82.000000      4.240000      6.600000      2.000000
2.240000
50%      108.000000      9.220000      9.800000      3.000000
2.490000
75%      222.000000     20.290000     22.000000      6.000000
2.920000
max      329.000000     32.980000     32.900000     16.000000
5.400000

       CustServ_Calls
count    24361.000000
mean         2.046673
std          1.760284
min          0.000000
25%          2.000000
50%          2.000000
75%          2.000000
max         11.000000
```

dropping the churn column

From what I've seen account_length seems to indicate the duration the account was open so given the stat summary of it I assume it is measured in days which validates the presence of other features

```
# df = df.drop(['Churn'], axis=1).copy()
```

Null & NaN

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

```
Phone_Number      0
Account_Length    0
VMail_Message     0
Day_Mins          0
Day_Calls         0
Day_Charge        0
Eve_Mins          0
Eve_Calls         0
Eve_Charge        0
Night_Mins        0
Night_Calls       0
Night_Charge      0
Intl_Mins         0
Intl_Calls        0
Intl_Charge       0
CustServ_Calls    0
Churn             0
dtype: int64
```

Duplicates

```
df.duplicated().sum()
# duplicated_rows = df[df.duplicated()]
# print(duplicated_rows)
```

```
40729
```

```
df.loc[df.duplicated()]
```

```
      Phone_Number  Account_Length  VMail_Message  Day_Mins  Day_Calls  \
9999      785-9657             228             28     268.2        220

10000     779-7999             202             26     262.6        228

10001     758-9959             282              0     228.2        222
```

```
10002     775-9999               82             0     299.2          22

10003     778-6656               28             0     266.2         228

...            ...              ...           ...       ...         ...

99985     775-9988              292            86     286.2          22

99986     757-7869               68             0     282.2          82

99987     797-5759               28             0     280.8         209

99988     765-9799              282             0     228.8         208

99989     786-8989               22            28     282.2         228


       Day_Charge  Eve_Mins  Eve_Calls  Eve_Charge  Night_Mins
Night_Calls  \
9999        28.02     292.2         99       26.28       222.2
92
10000       22.22     298.8        208       26.62       282.2
208
10001       22.88     222.2        220       20.80       262.6
202
10002       80.90      62.9         88        8.26       296.9
89
10003       28.82     228.8        222       22.62       286.9
222
...           ...       ...        ...         ...         ...
...
99985       26.88     228.8        226       28.82       229.2
88
99986       89.29     288.2         88       28.02       292.8
228
99987       80.22     288.8         88       22.88       292.9
92
99988       86.88     289.6         82       28.82       289.2
282
99989       89.88     268.9         82       22.60       222.2
22

       Night_Charge  Intl_Mins  Intl_Calls  Intl_Charge
CustServ_Calls  Churn
9999          22.02       20.0           8         2.20
2  False
10000         22.28       28.2           8         8.20
2  False
10001          2.82       22.2           8         8.29
0  False
```

```
10002          8.86       6.6          2         2.28
2  False
10003          8.22      20.2          8         2.28
8  False
...             ...       ...        ...          ...               ..
.    ...
99985         22.86       9.9          6         2.62
2  False
99986          8.62       9.6          2         2.89
8  False
99987          8.62      22.2          6         8.82
2  False
99988          6.26       8.0         20         2.88
2  False
99989         20.86      28.2          2         8.20
0  False

[40729 rows x 17 columns]
```

```
df.query('Phone_Number == "779-7999"').head()
```

```
      Phone_Number  Account_Length  VMail_Message  Day_Mins  Day_Calls
\
3334      779-7999             407             46     464.6        445

6667      779-7999             202             26     262.6        228

10000     779-7999             202             26     262.6        228

13333     779-7999             202             26     262.6        228

16666     779-7999             202             26     262.6        228


       Day_Charge  Eve_Mins  Eve_Calls  Eve_Charge  Night_Mins
Night_Calls  \
3334       47.47     495.5        405       46.64       454.4
405
6667       22.22     298.8        208       26.62       282.2
208
10000      22.22     298.8        208       26.62       282.2
208
13333      22.22     298.8        208       26.62       282.2
208
16666      22.22     298.8        208       26.62       282.2
208

       Night_Charge  Intl_Mins  Intl_Calls  Intl_Charge
CustServ_Calls  Churn
3334          44.45       45.7           5          5.7
```

```
4  False
6667          22.28       28.2            8          8.2
2  False
10000         22.28       28.2            8          8.2
2  False
13333         22.28       28.2            8          8.2
2  False
16666         22.28       28.2            8          8.2
2  False
```

```
df = df.loc[~df.duplicated()].reset_index(drop=True).copy()
df
```

```
      Phone_Number  Account_Length  VMail_Message  Day_Mins  Day_Calls
\
0         382-4657             128             25     265.1        110

1         371-7191             107             26     161.6        123

2         358-1921             137              0     243.4        114

3         375-9999              84              0     299.4         71

4         330-6626              75              0     166.7        113

...            ...             ...            ...       ...        ...

60440     789-9756             222              0     228.2         60

60441     798-5885              88              0     282.2        222

60442     798-5798              22              0     222.2         62

60443     999-9897             228              0     222.0         99

60444     786-7589             228              0     226.2         98


       Day_Charge  Eve_Mins  Eve_Calls  Eve_Charge  Night_Mins
Night_Calls  \
0           45.07     197.4         99       16.78       244.7
91
1           27.47     195.5        103       16.62       254.4
103
2           41.38     121.2        110       10.30       162.6
104
3           50.90      61.9         88        5.26       196.9
89
4           28.34     148.3        122       12.61       186.9
121
...           ...       ...        ...         ...         ...
```

```
...
60440       22.82     229.8        289       28.26       222.8
222
60441       82.88     208.8        220       22.82       282.2
200
60442       88.66     228.0        228       22.08        62.2
209
60443       88.08     220.2         80       22.92       282.9
28
60444       86.28     288.2        208       28.28       800.0
228

       Night_Charge  Intl_Mins  Intl_Calls  Intl_Charge  
CustServ_Calls  Churn  
0             11.01       10.0           3         2.70  
1  False  
1             11.45       13.7           3         3.70  
1  False  
2              7.32       12.2           5         3.29  
0  False  
3              8.86        6.6           7         1.78  
2  False  
4              8.41       10.1           3         2.73  
3  False  
...             ...        ...         ...          ...              ..
.   ...  
60440          2.28        6.2           2         2.62  
2  False  
60441         20.68        9.8           8         2.82  
8  False  
60442          2.26        2.8           6         2.22  
2  False  
60443         20.22        2.2           8         0.82  
0  False  
60444         28.80       20.0           8         2.20  
2  False  

[60445 rows x 17 columns]
```

unique counts in a feature

```
df.nunique()

Phone_Number      7467
Account_Length     322
VMail_Message       72
Day_Mins          2548
Day_Calls          221
Day_Charge        2873
Eve_Mins          2523
```

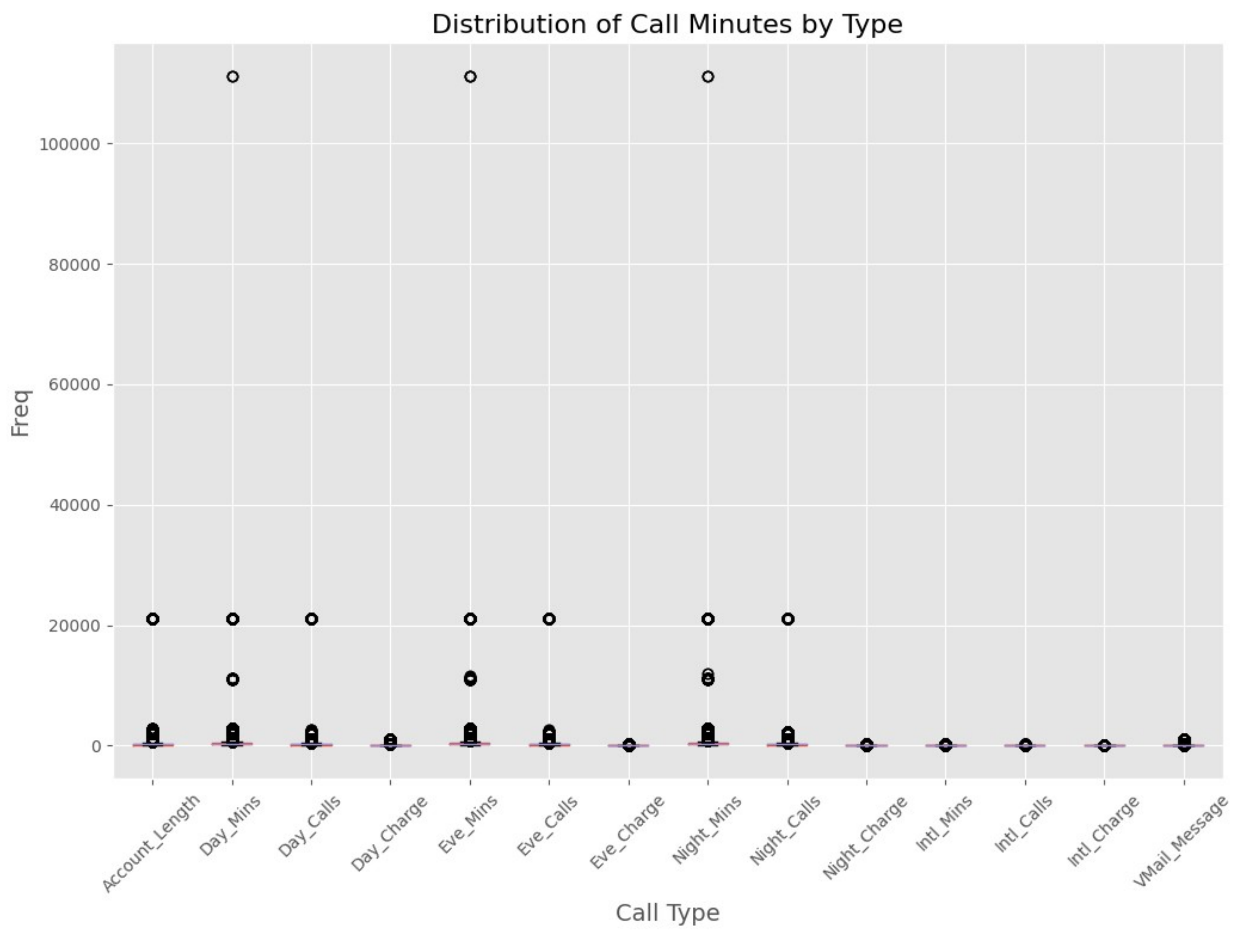
```
Eve_Calls          224
Eve_Charge        2221
Night_Mins        2464
Night_Calls        218
Night_Charge      1470
Intl_Mins          267
Intl_Calls          39
Intl_Charge        339
CustServ_Calls      11
Churn                2
dtype: int64
```

```
df.shape
```

```
(60445, 17)
```

Outliers

```
fig, ax = plt.subplots(figsize=(12, 8))

numeric_cols = ['Account_Length','Day_Mins','Day_Calls','Day_Charge',
'Eve_Mins', 'Eve_Calls','Eve_Charge',
'Night_Mins','Night_Calls','Night_Charge','Intl_Mins','Intl_Calls','In
tl_Charge', 'VMail_Message']

# Create a box plot
df.boxplot(column= numeric_cols,
           ax=ax)

# Add titles and labels
ax.set_title('Distribution of Call Minutes by Type', fontsize=16)
ax.set_xlabel('Call Type', fontsize=14)
ax.set_ylabel('Freq', fontsize=14)
ax.set_xticklabels(numeric_cols, rotation=45)

plt.show()
```

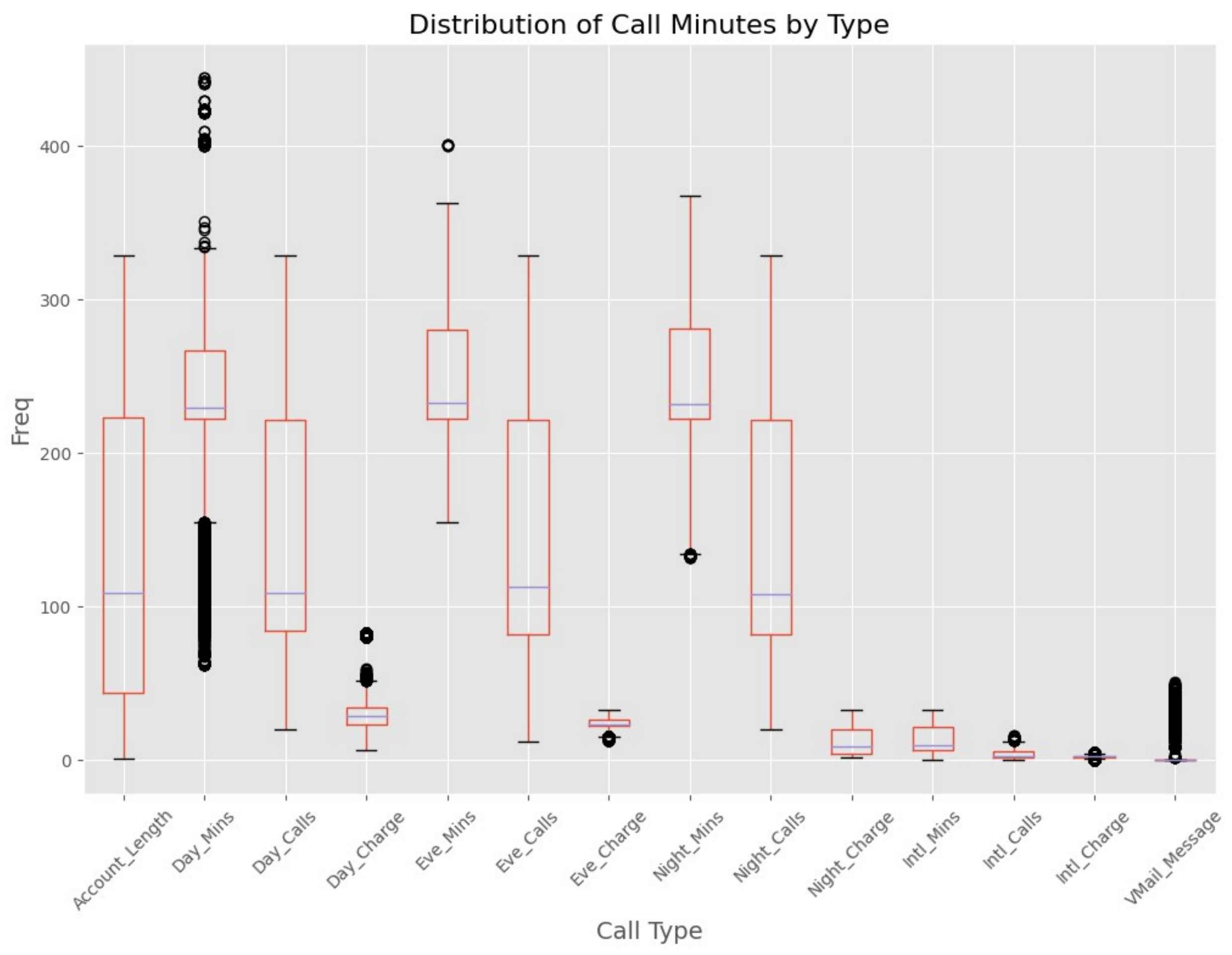


```
df.describe()

       Account_Length  VMail_Message      Day_Mins     Day_Calls  \
count    60445.000000   60445.000000  60445.000000  60445.000000
mean       329.385541      18.101613    596.555649    274.752138
std       1436.763064      76.859936   2231.437206    997.047766
min          1.000000       0.000000      0.000000      0.000000
25%         68.000000       0.000000    222.200000     88.000000
50%        200.000000       0.000000    262.200000    202.000000
75%        243.000000      22.000000    404.400000    226.000000
max      21111.000000    1111.000000 111111.110000  21111.000000

         Day_Charge      Eve_Mins     Eve_Calls    Eve_Charge
Night_Mins  \
count  60445.000000  60445.000000  60445.000000  60445.000000
60445.000000
mean      64.880486    669.288486    272.647911     34.894168
661.395415
std      134.695269   2476.768325    956.454815     36.584588
2422.994289
min        0.000000      0.000000      0.000000      0.000000
20.300000
```

```
25%       24.220000     223.400000     88.000000     22.220000
223.200000
50%       32.920000     268.800000    200.000000     24.900000
269.110000
75%       80.480000     440.400000    226.000000     32.620000
440.400000
max     1111.990000  111111.200000  21111.000000    211.990000
111111.110000

        Night_Calls  Night_Charge     Intl_Mins    Intl_Calls
Intl_Charge  \
count  60445.000000  60445.000000  60445.000000  60445.000000
60445.000000
mean     267.133377     14.066812     20.571371      5.728894
4.315376
std      913.240573     16.532483     25.494288      8.271802
2.711543
min       20.000000      1.040000      0.000000      0.000000
0.000000
25%       88.000000      6.220000      8.800000      2.000000
2.280000
50%      200.000000      9.220000     20.200000      4.000000
2.920000
75%      226.000000     20.400000     22.800000      8.000000
4.940000
max    21111.000000    211.920000    211.900000    211.000000
11.920000

       CustServ_Calls
count    60445.000000
mean         2.563438
std          2.376449
min          0.000000
25%          2.000000
50%          2.000000
75%          4.000000
max         11.000000


# df = df[
#     (df['Account_Length'] <= 20000) &
#     (df['Day_Mins'] <= 20000) &
#     (df['Eve_Mins'] <= 20000) &
#     (df['Night_Mins'] <= 20000) &
#     (df['Intl_Mins'] <= 20000) &
#     (df['Day_Calls'] <= 20000) &
#     (df['Eve_Calls'] <= 20000) &
#     (df['Night_Calls'] <= 20000) &
#     (df['Intl_Calls'] <= 20000) &
#     (df['CustServ_Calls'] <= 20000)
# ].copy()

def remove_outliers_iqr(df, column_name):
    Q1 = df[column_name].quantile(0.25)
    Q3 = df[column_name].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column_name] >= lower_bound) & (df[column_name] <=
upper_bound)]


# Loop through each numerical column
for col in numeric_cols:
    df = remove_outliers_iqr(df, col)

# Reset index after filtering
df.reset_index(drop=True, inplace=True)

df.shape
```

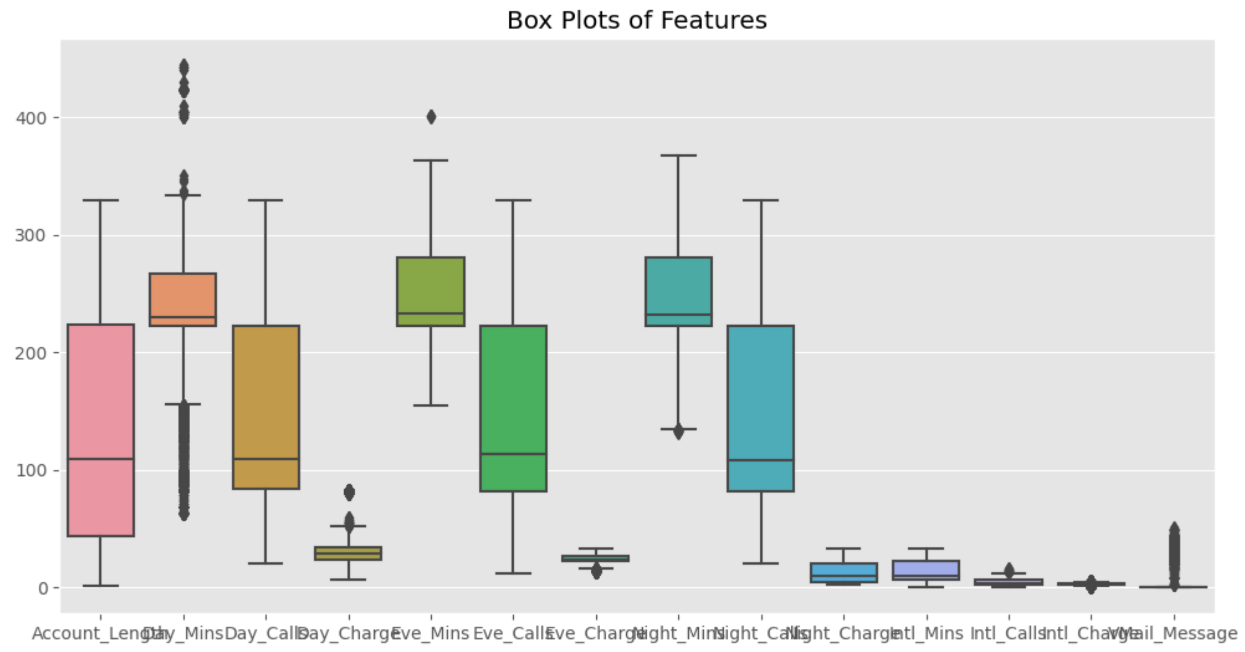
```
(24361, 17)
```

```
fig, ax = plt.subplots(figsize=(12, 8))

# box plot
df.boxplot(column=numeric_cols,
           ax=ax)

# titles and labels
ax.set_title('Distribution of Call Minutes by Type', fontsize=16)
ax.set_xlabel('Call Type', fontsize=14)
ax.set_ylabel('Freq', fontsize=14)
ax.set_xticklabels(numeric_cols, rotation=45)

plt.show()
```



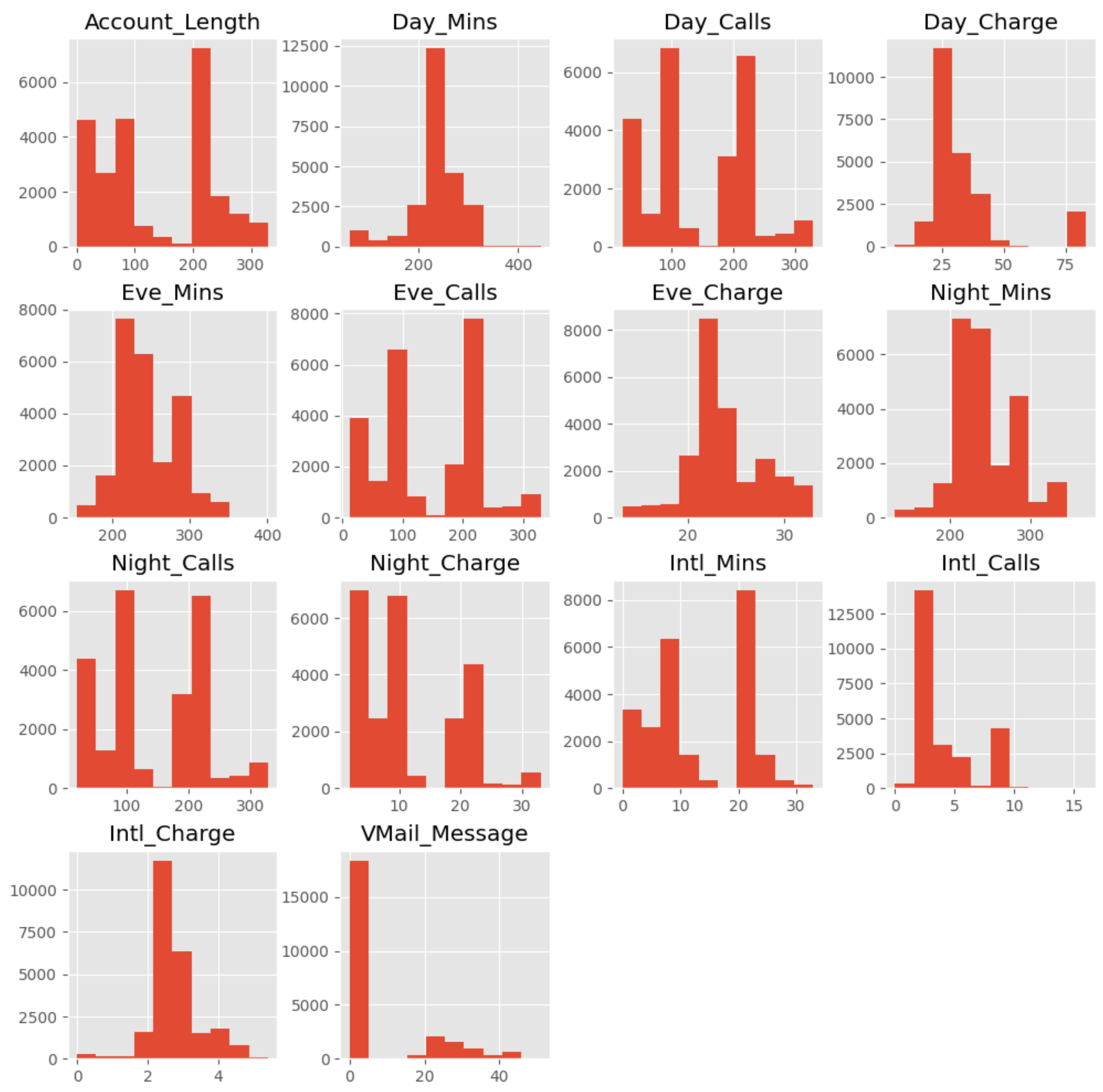
other visualizations

```
# boxplot
plt.figure(figsize=(12,6))
sns.boxplot(data=df[numeric_cols])
plt.title('Box Plots of Features')
plt.show()
```

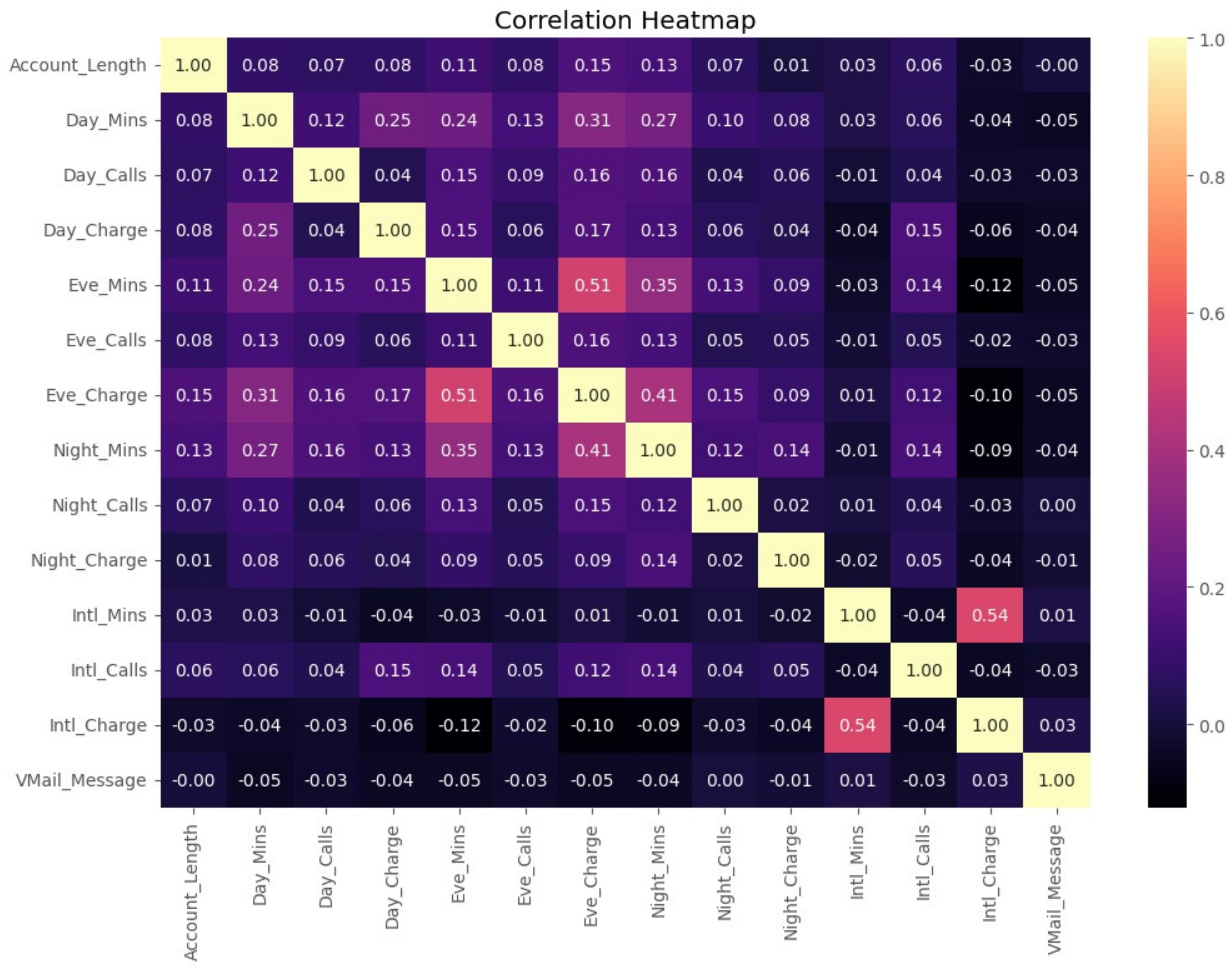


```
df.describe()

       Account_Length  VMail_Message      Day_Mins     Day_Calls  \
count    24361.000000   24361.000000  24361.000000  24361.000000
mean       142.557982       7.007143    235.520038    144.334017
std         93.875565      12.810141     50.888264     83.959842
min          1.000000       0.000000     62.300000     20.000000
25%         44.000000       0.000000    222.200000     84.000000
50%        109.000000       0.000000    229.800000    109.000000
75%        223.000000       0.000000    266.900000    222.000000
max        329.000000      51.000000    444.400000    329.000000

         Day_Charge      Eve_Mins     Eve_Calls    Eve_Charge
Night_Mins  \
count  24361.000000  24361.000000  24361.000000  24361.000000
24361.000000
mean      33.434129    245.841118    145.180370     24.016874
244.790738
std       16.377482     35.953885     83.943032      3.877371
37.154832
min        6.220000    154.800000     12.000000     13.160000
131.600000
25%       22.960000    222.400000     82.000000     22.220000
222.300000
50%       28.820000    232.900000    113.000000     22.980000
232.300000
75%       34.410000    280.200000    222.000000     26.620000
280.800000
max       82.980000    400.900000    329.000000     32.990000
367.700000
```

```
        Night_Calls  Night_Charge     Intl_Mins    Intl_Calls
Intl_Charge  \
count  24361.000000  24361.000000  24361.000000  24361.000000
24361.000000
mean     143.544559     11.437698     13.342622      3.904109
2.709633
std       83.582519      8.018891      8.078907      2.395398
0.722454
min       20.000000      2.000000      0.000000      0.000000
0.000000
25%       82.000000      4.240000      6.600000      2.000000
2.240000
50%      108.000000      9.220000      9.800000      3.000000
2.490000
75%      222.000000     20.290000     22.000000      6.000000
2.920000
max      329.000000     32.980000     32.900000     16.000000
5.400000

       CustServ_Calls
count    24361.000000
mean         2.046673
std          1.760284
min          0.000000
25%          2.000000
50%          2.000000
75%          2.000000
max         11.000000

# histograms
df[numeric_cols].hist(figsize=(12, 12))
plt.show()
```

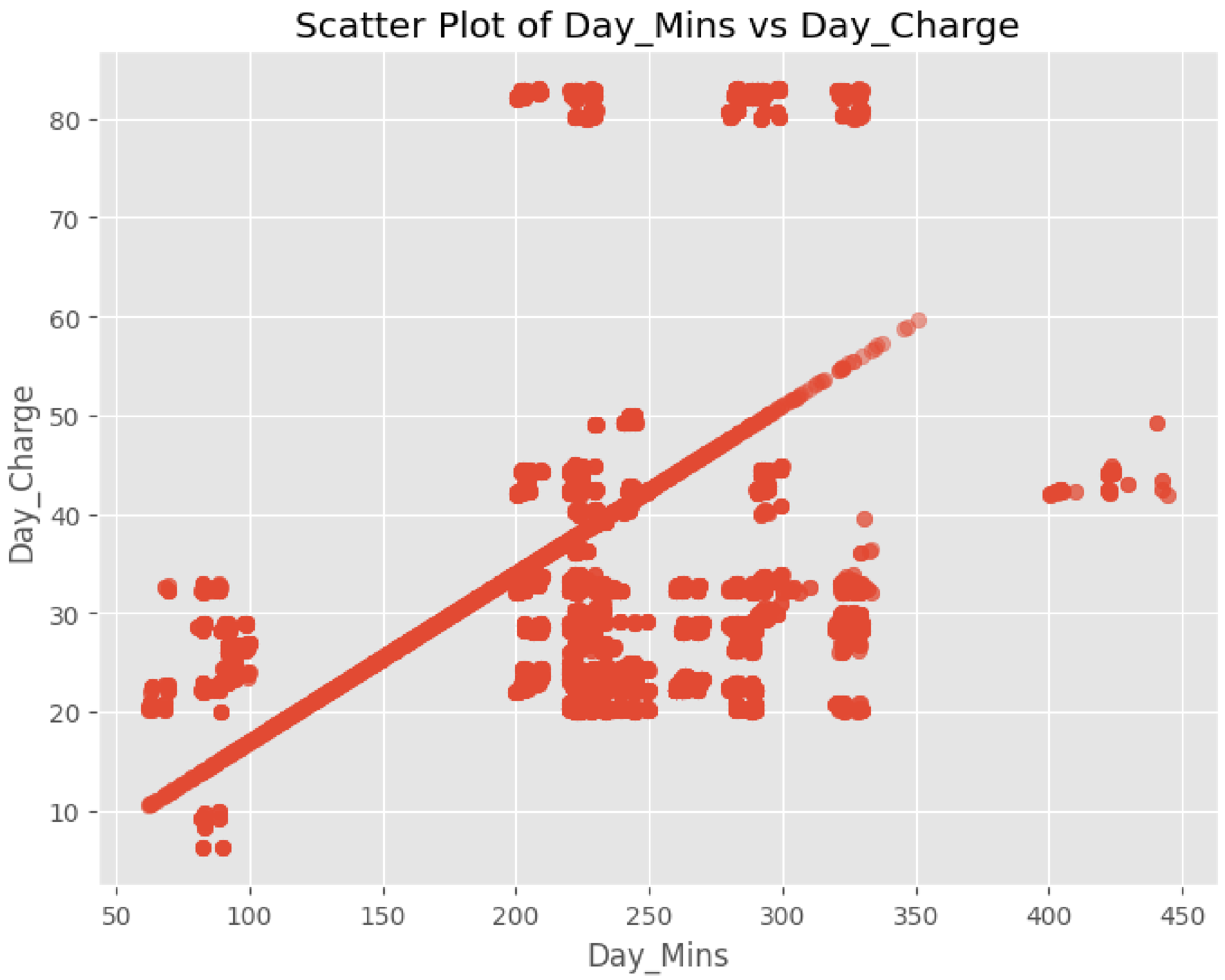


```
# correlation heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(df[numeric_cols].corr(), annot=True, fmt='.2f',
cmap='magma')
plt.title('Correlation Heatmap')
plt.show()
```

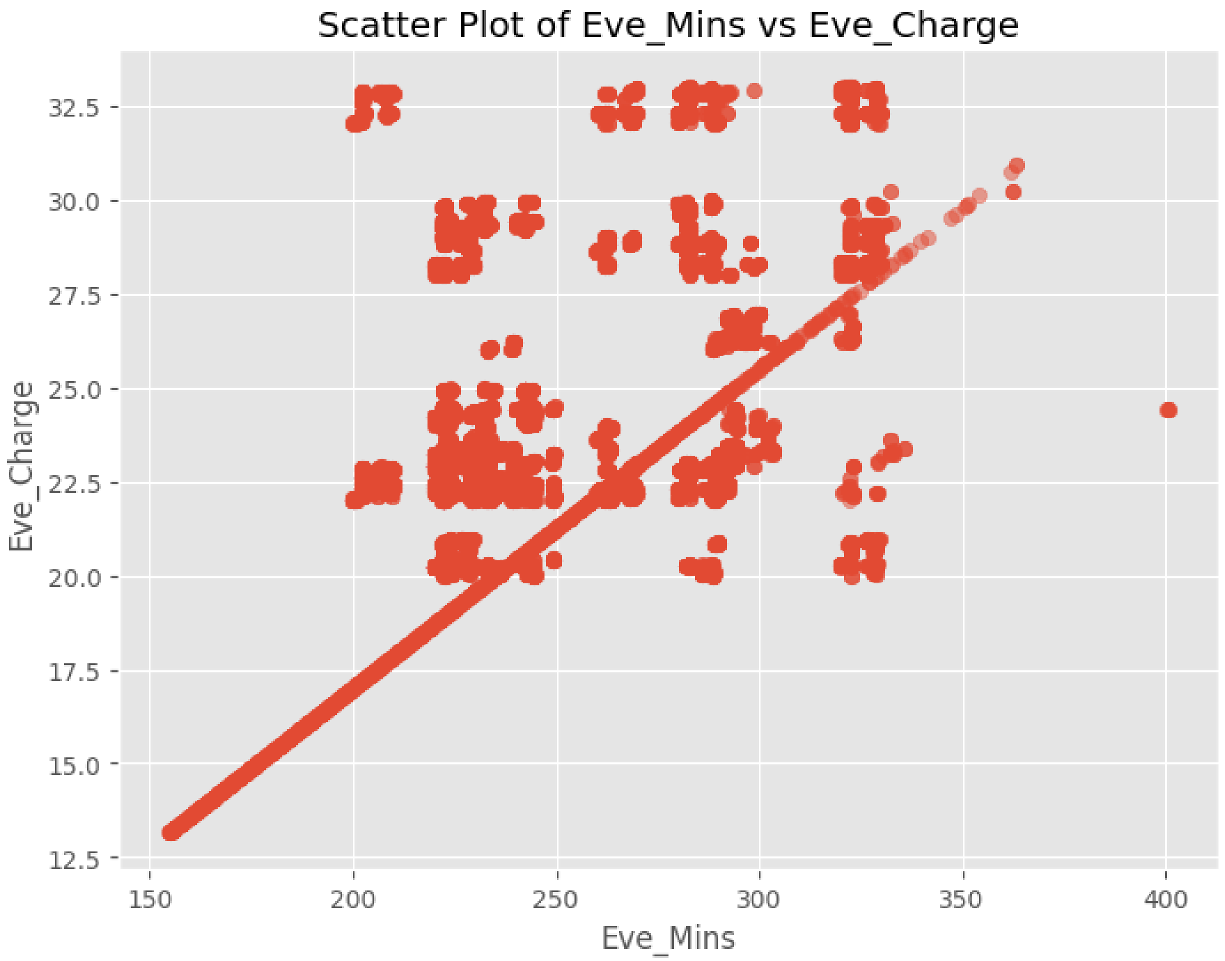


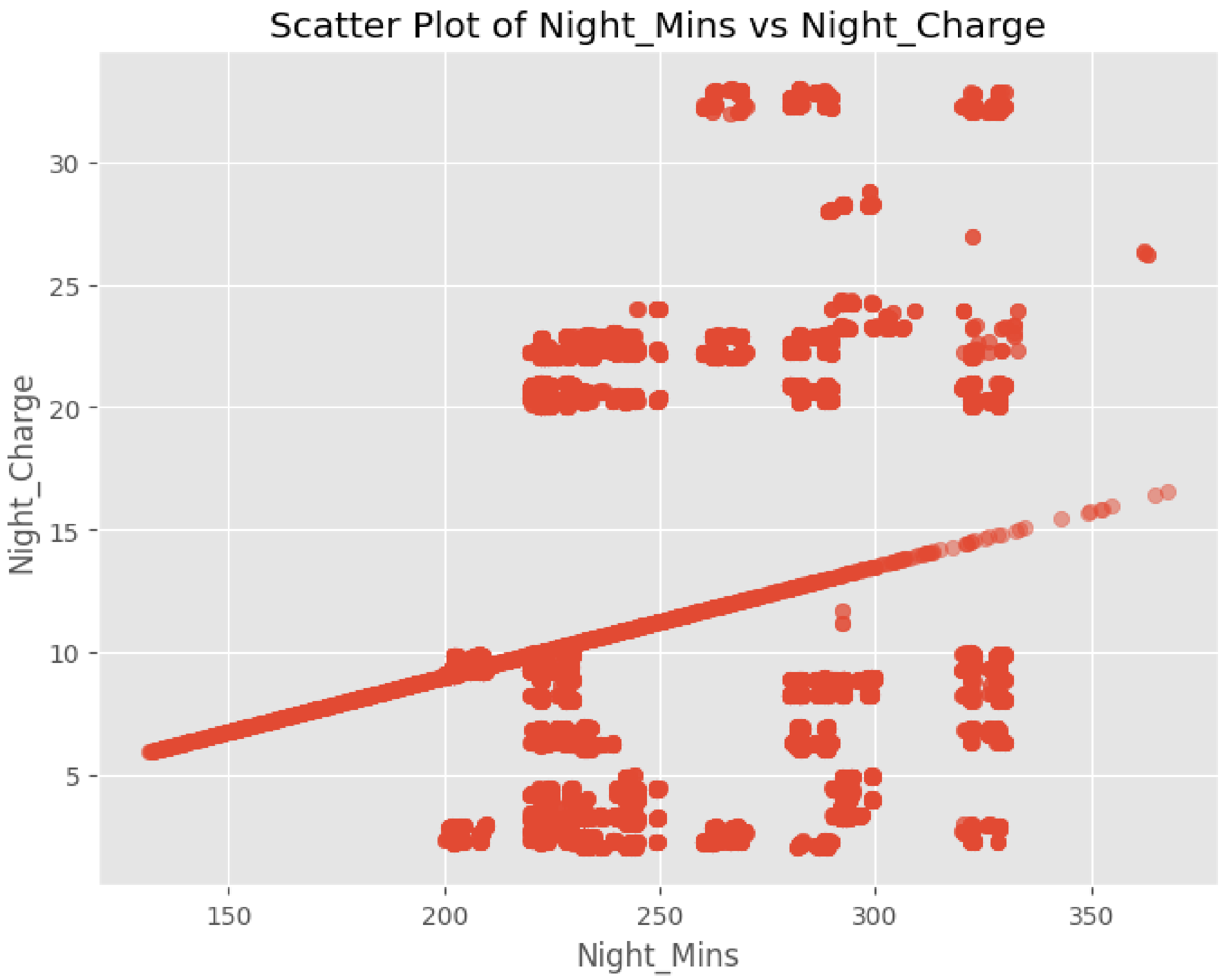
	Account_Length	Day_Mins	Day_Calls	Day_Charge	Eve_Mins	Eve_Calls	Eve_Charge	Night_Mins	Night_Calls	Night_Charge	Intl_Mins	Intl_Calls	Intl_Charge	VMail_Message
Account_Length	1.00	0.08	0.07	0.08	0.11	0.08	0.15	0.13	0.07	0.01	0.03	0.06	-0.03	-0.00
Day_Mins	0.08	1.00	0.12	0.25	0.24	0.13	0.31	0.27	0.10	0.08	0.03	0.06	-0.04	-0.05
Day_Calls	0.07	0.12	1.00	0.04	0.15	0.09	0.16	0.16	0.04	0.06	-0.01	0.04	-0.03	-0.03
Day_Charge	0.08	0.25	0.04	1.00	0.15	0.06	0.17	0.13	0.06	0.04	-0.04	0.15	-0.06	-0.04
Eve_Mins	0.11	0.24	0.15	0.15	1.00	0.11	0.51	0.35	0.13	0.09	-0.03	0.14	-0.12	-0.05
Eve_Calls	0.08	0.13	0.09	0.06	0.11	1.00	0.16	0.13	0.05	0.05	-0.01	0.05	-0.02	-0.03
Eve_Charge	0.15	0.31	0.16	0.17	0.51	0.16	1.00	0.41	0.15	0.09	0.01	0.12	-0.10	-0.05
Night_Mins	0.13	0.27	0.16	0.13	0.35	0.13	0.41	1.00	0.12	0.14	-0.01	0.14	-0.09	-0.04
Night_Calls	0.07	0.10	0.04	0.06	0.13	0.05	0.15	0.12	1.00	0.02	0.01	0.04	-0.03	0.00
Night_Charge	0.01	0.08	0.06	0.04	0.09	0.05	0.09	0.14	0.02	1.00	-0.02	0.05	-0.04	-0.01
Intl_Mins	0.03	0.03	-0.01	-0.04	-0.03	-0.01	0.01	-0.01	0.01	-0.02	1.00	-0.04	0.54	0.01
Intl_Calls	0.06	0.06	0.04	0.15	0.14	0.05	0.12	0.14	0.04	0.05	-0.04	1.00	-0.04	-0.03
Intl_Charge	-0.03	-0.04	-0.03	-0.06	-0.12	-0.02	-0.10	-0.09	-0.03	-0.04	0.54	-0.04	1.00	0.03
VMail_Message	-0.00	-0.05	-0.03	-0.04	-0.05	-0.03	-0.05	-0.04	0.00	-0.01	0.01	-0.03	0.03	1.00

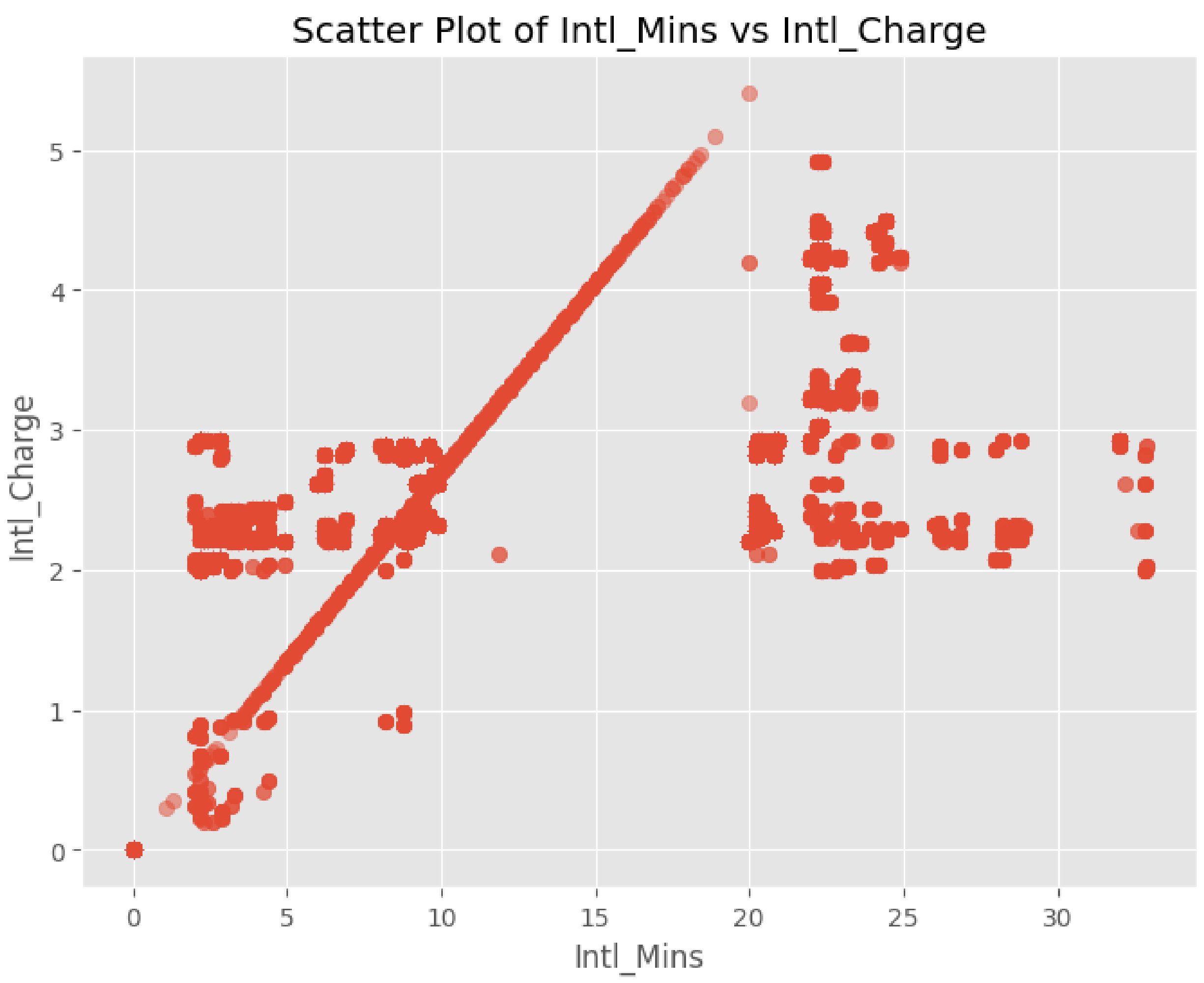
```python
# List of feature pairs
feature_pairs = [
    ('Day_Mins', 'Day_Charge'),
    ('Eve_Mins', 'Eve_Charge'),
    ('Night_Mins', 'Night_Charge'),
    ('Intl_Mins', 'Intl_Charge'),
    # ...
]

# scatter plots for each pair
for x, y in feature_pairs:
    plt.figure(figsize=(8, 6))
    plt.scatter(df[x], df[y], alpha=0.5)
    plt.xlabel(x)
    plt.ylabel(y)
    plt.title(f'Scatter Plot of {x} vs {y}')
    plt.show()
```



Scatter Plot of Day_Mins vs Day_Charge
Day_Charge
Day_Mins


Scatter Plot of Eve_Mins vs Eve_Charge
Eve_Charge
Eve_Mins
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0
32.5
150
200
250
300
350
400


Scatter Plot of Night_Mins vs Night_Charge
Night_Charge
Night_Mins
5
10
15
20
25
30
150
200
250
300
350

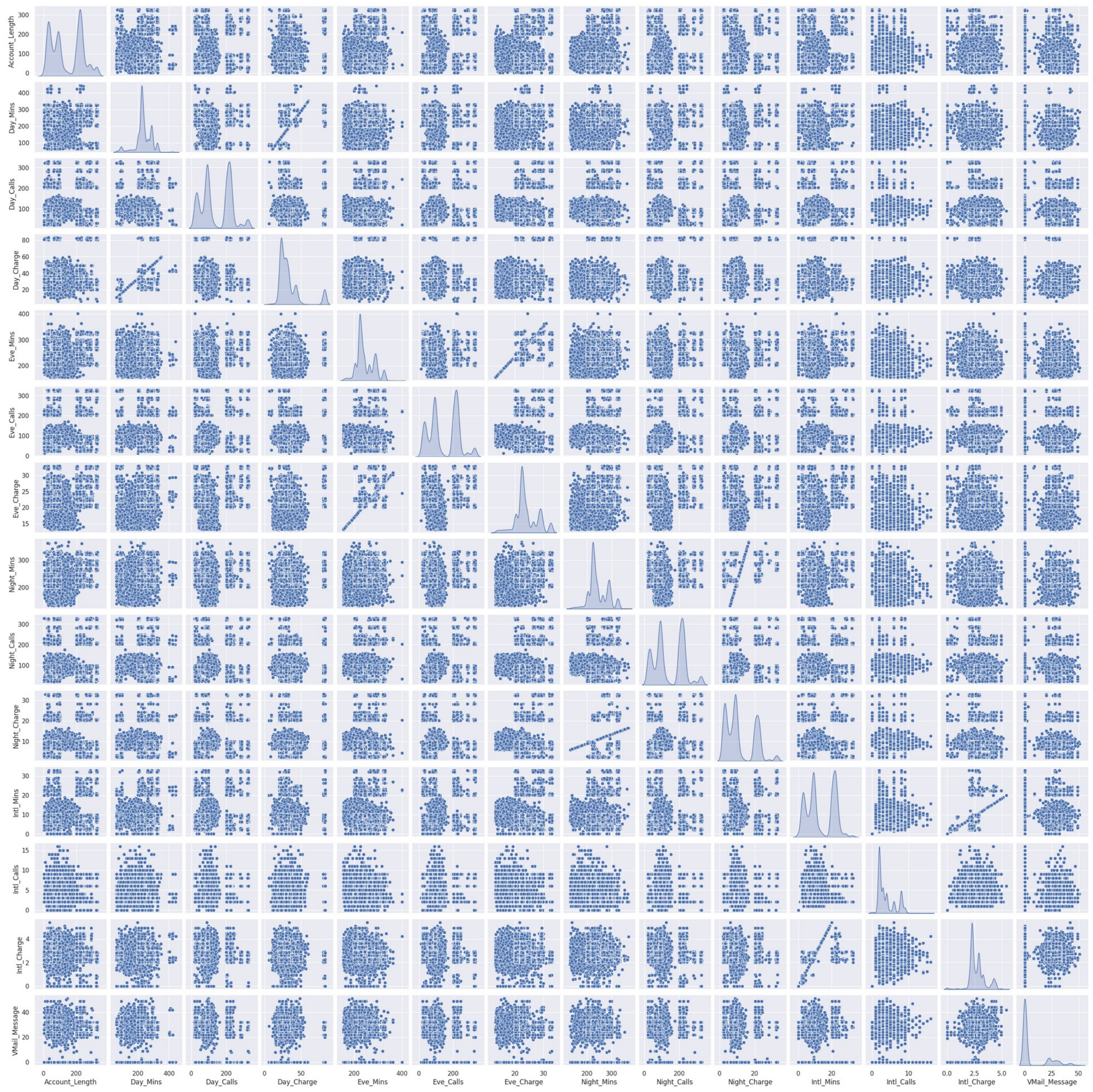


```
# pairplot
sns.set()
sns.pairplot(df[numeric_cols],size = 2 ,kind
='scatter',diag_kind='kde')
plt.show()
```

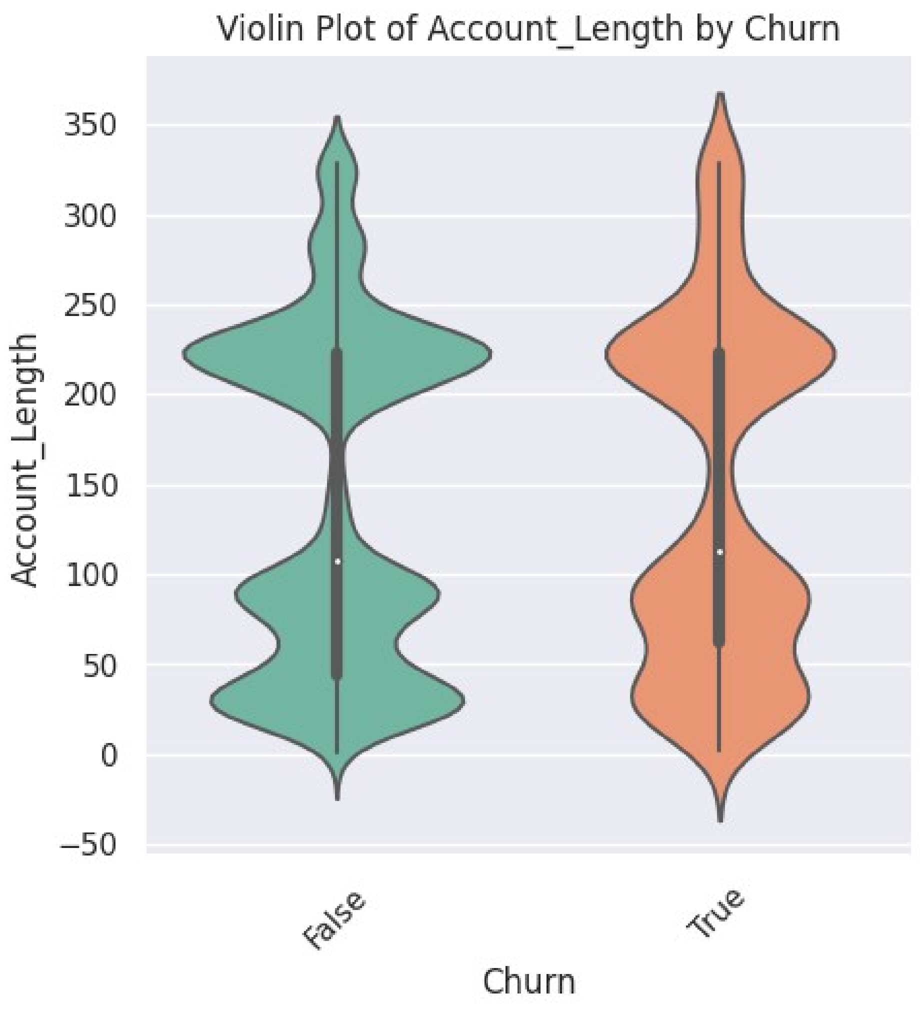
```
/opt/conda/lib/python3.10/site-packages/seaborn/axisgrid.py:2095:
UserWarning: The `size` parameter has been renamed to `height`; please
update your code.
  warnings.warn(msg, UserWarning)
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
```

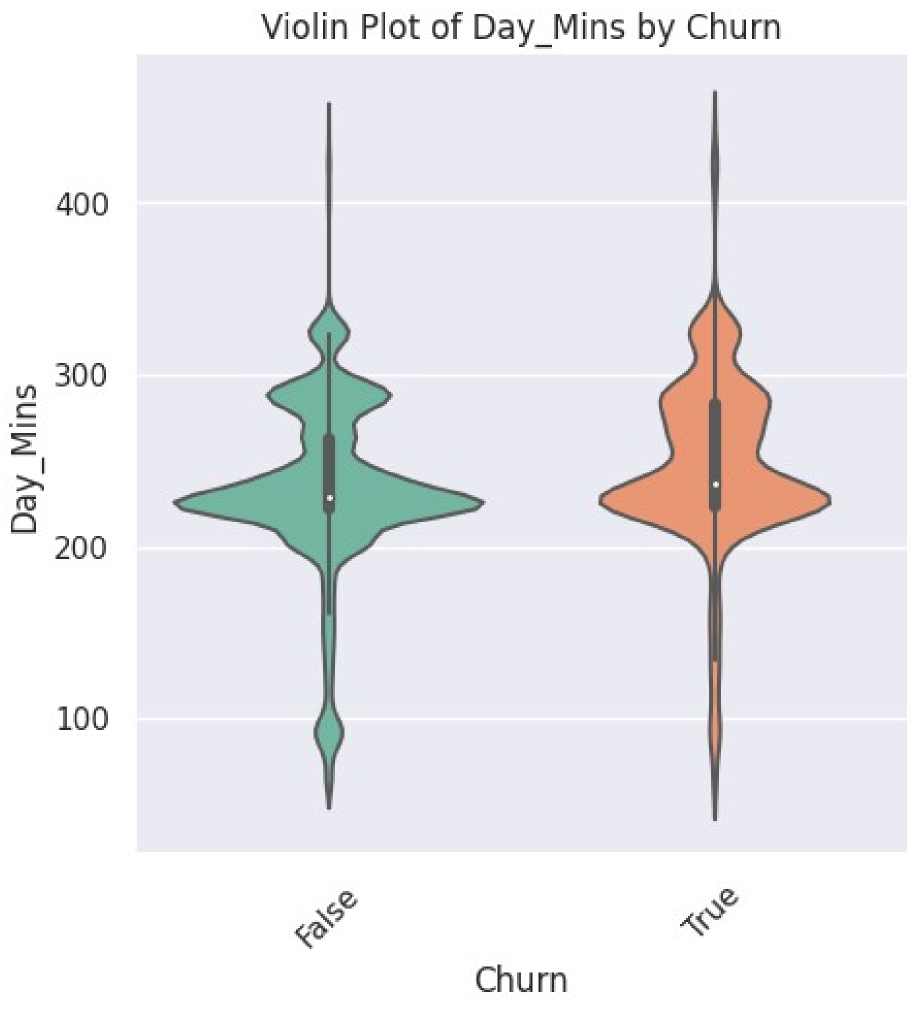
```
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
```

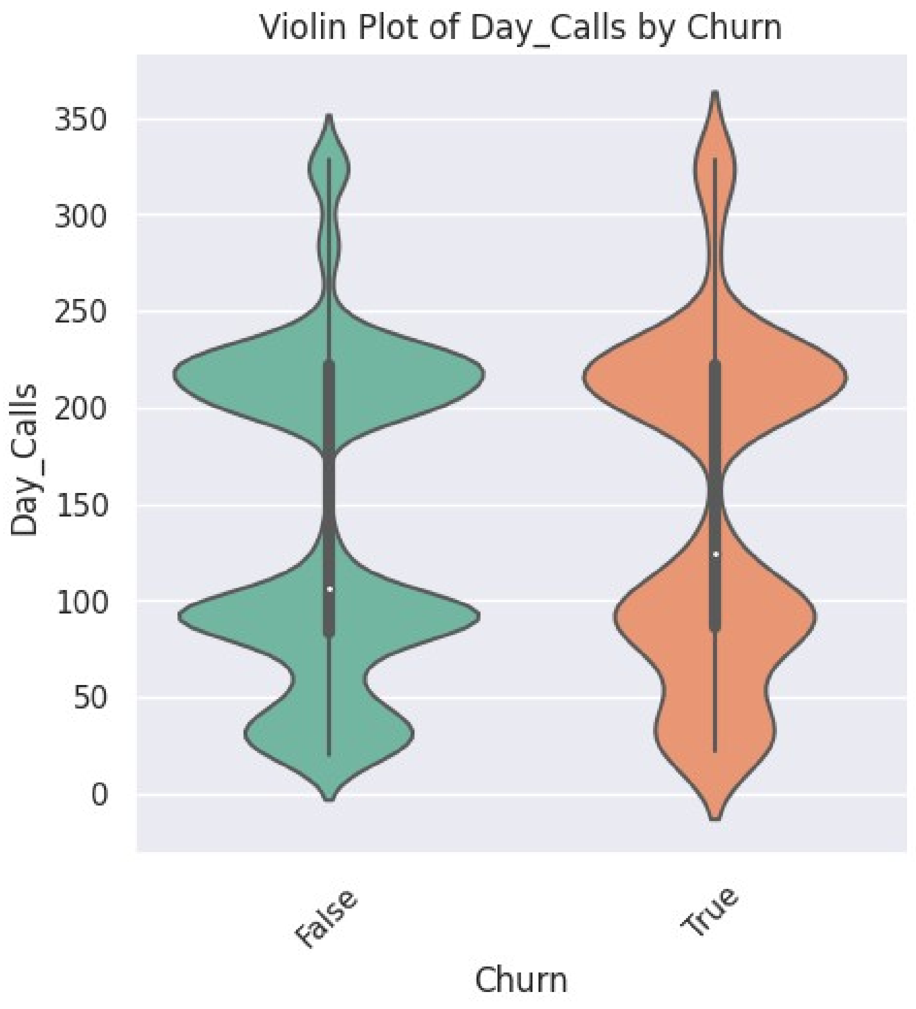
```
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

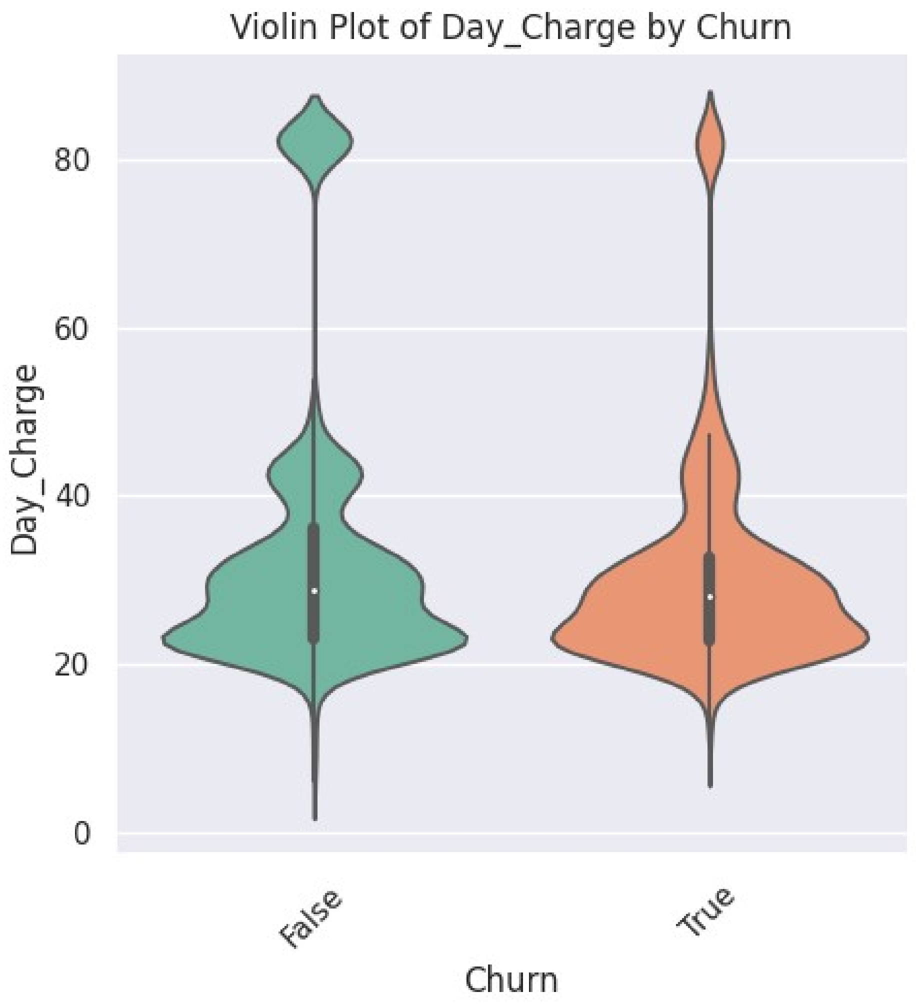


```
# violin plot
for num_col in numeric_cols:
    sns.catplot(x='Churn', y=num_col, kind='violin', data=df,
palette='Set2')
    plt.title(f'Violin Plot of {num_col} by Churn')
    plt.xticks(rotation=45)  # Rotate x labels if they overlap
    plt.show()
```

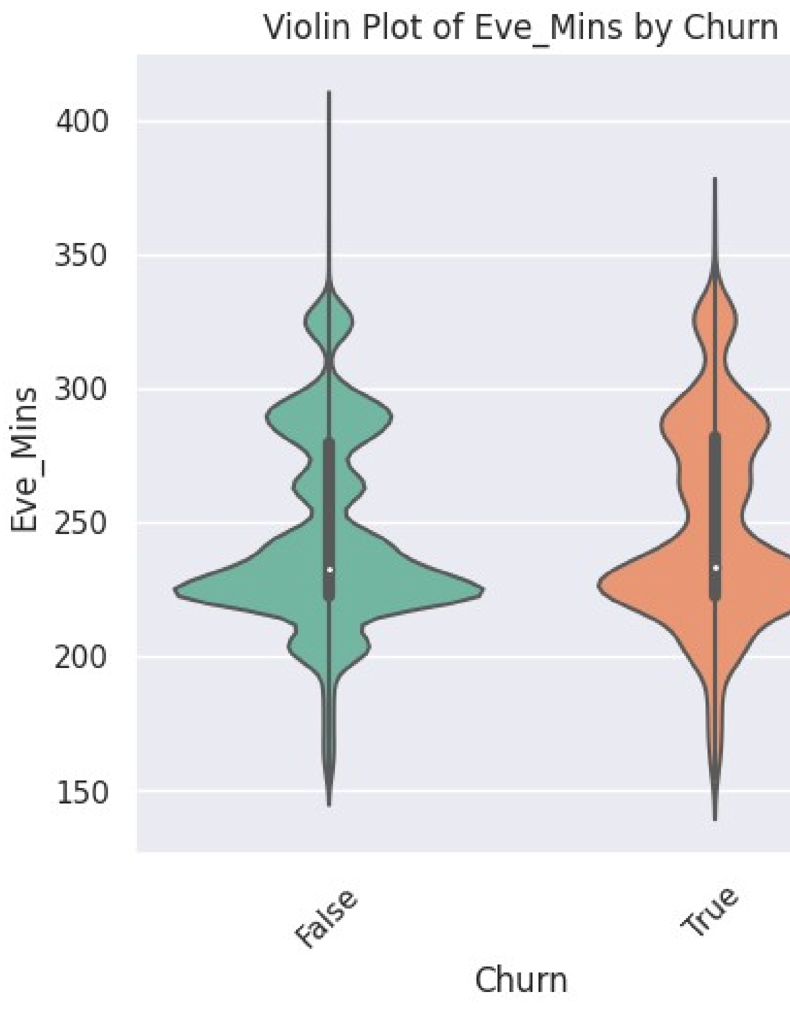


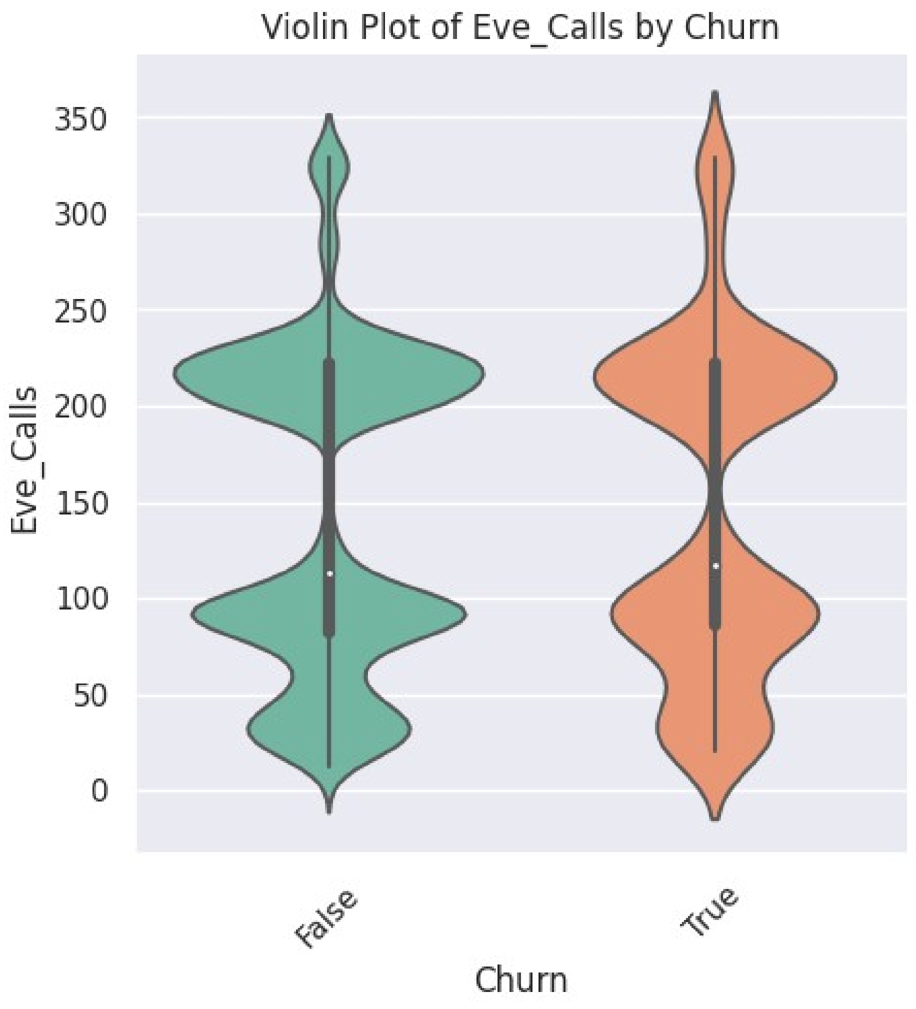

Violin Plot of Day_Mins by Churn
Day_Mins
400
300
200
100
False
True
Churn


Violin Plot of Day_Calls by Churn
350
300
250
200
150
100
50
0
Day_Calls
False
True
Churn

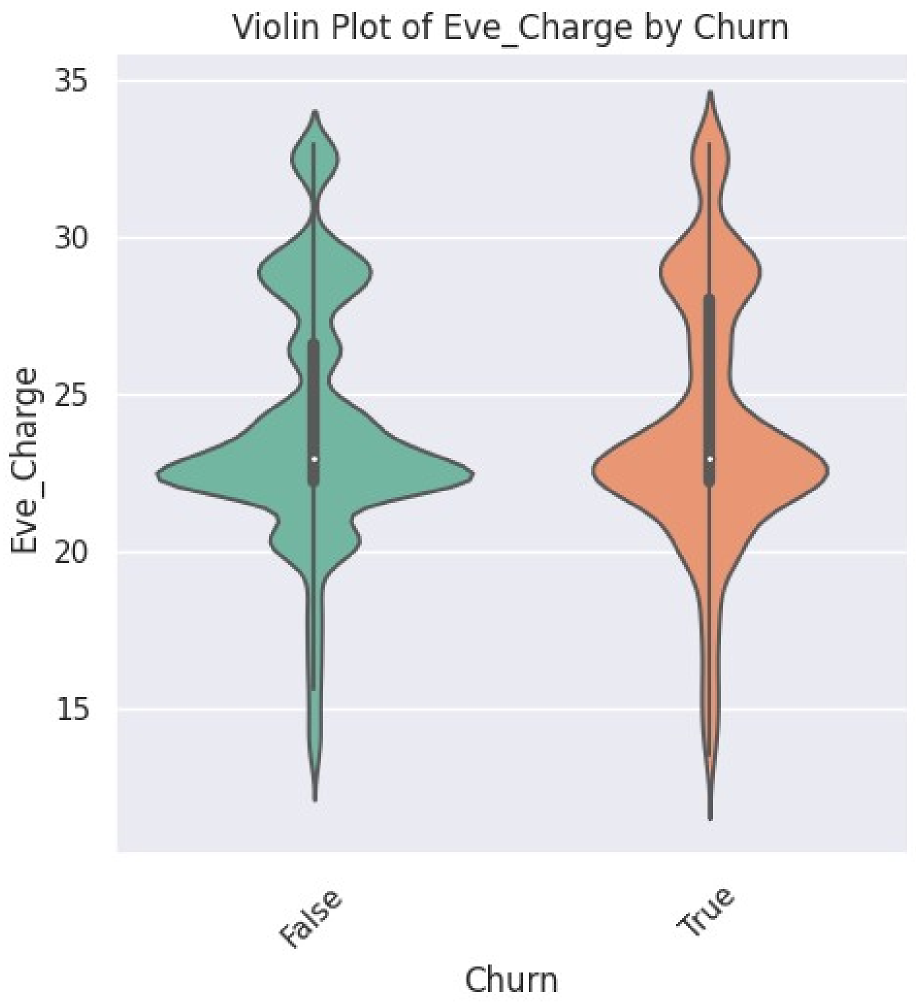


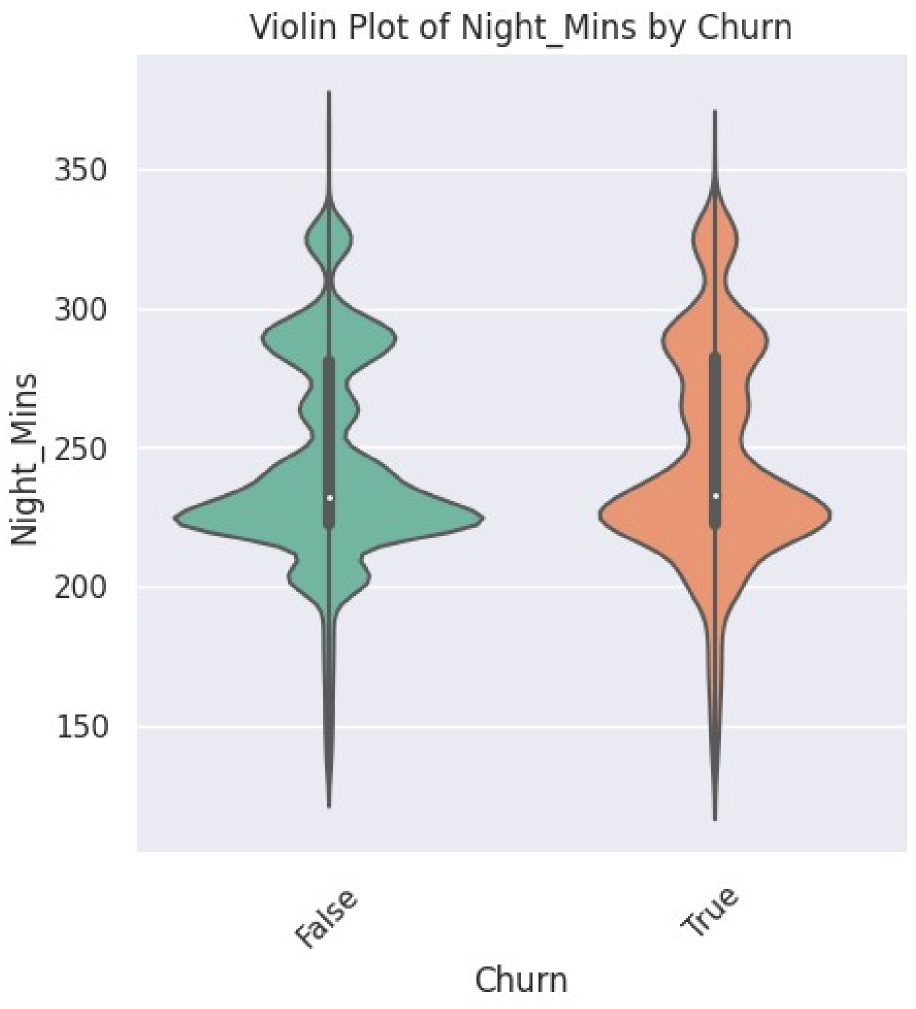
Violin Plot of Day_Charge by Churn
Day_Charge
0
20
40
60
80
False
True
Churn


Violin Plot of Eve_Mins by Churn
400
350
300
250
200
150
Eve_Mins
False
True
Churn

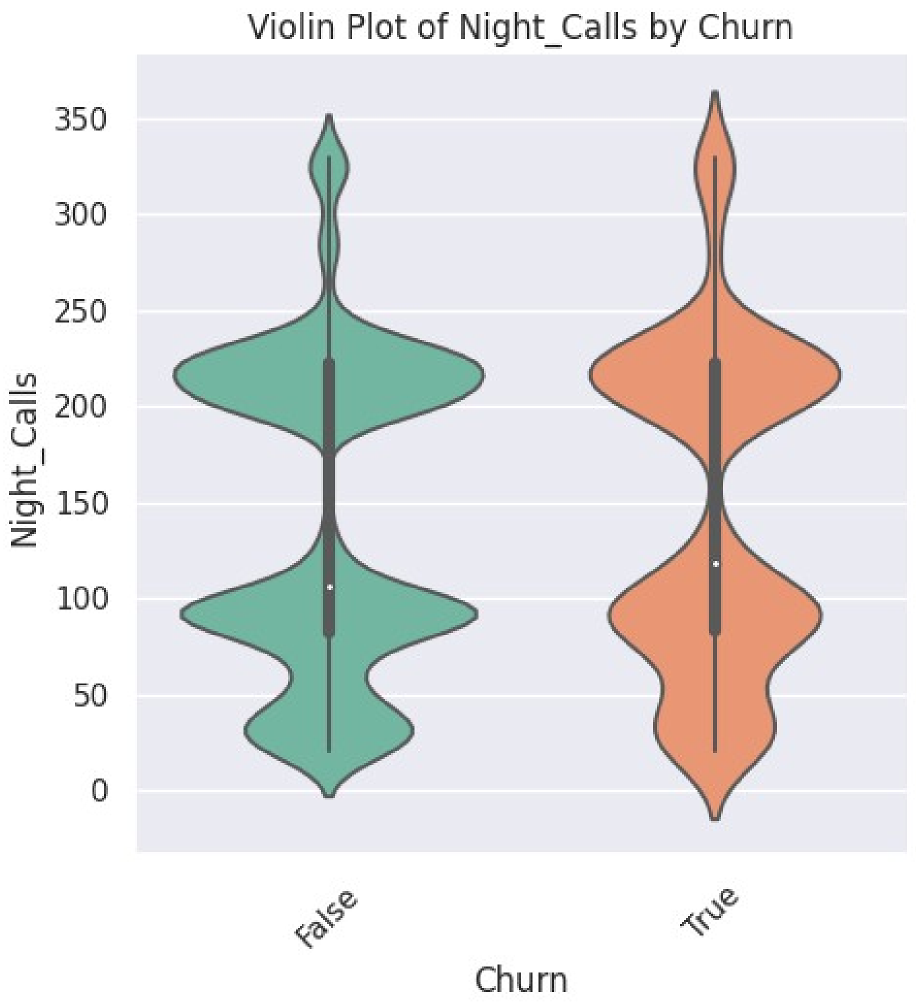


Violin Plot of Eve_Calls by Churn
Eve_Calls
350
300
250
200
150
100
50
0
False
True
Churn

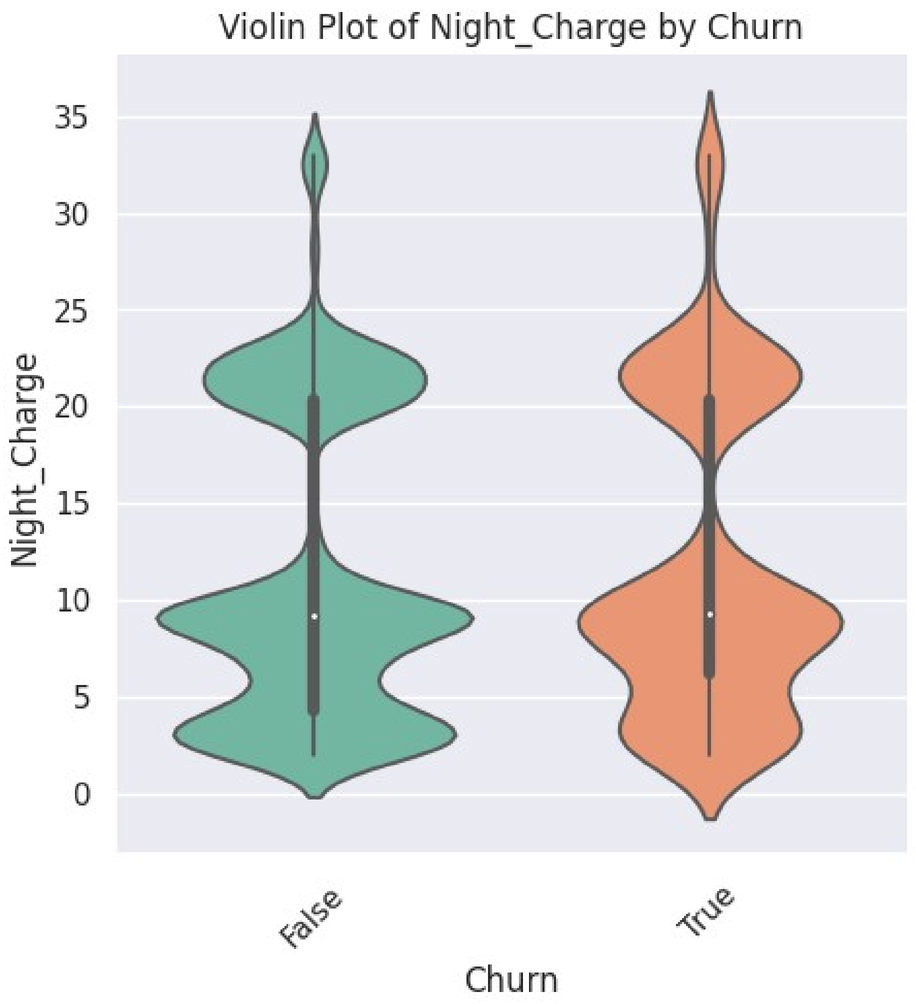

Violin Plot of Eve_Charge by Churn
35
30
25
20
15
Eve_Charge
False
True
Churn

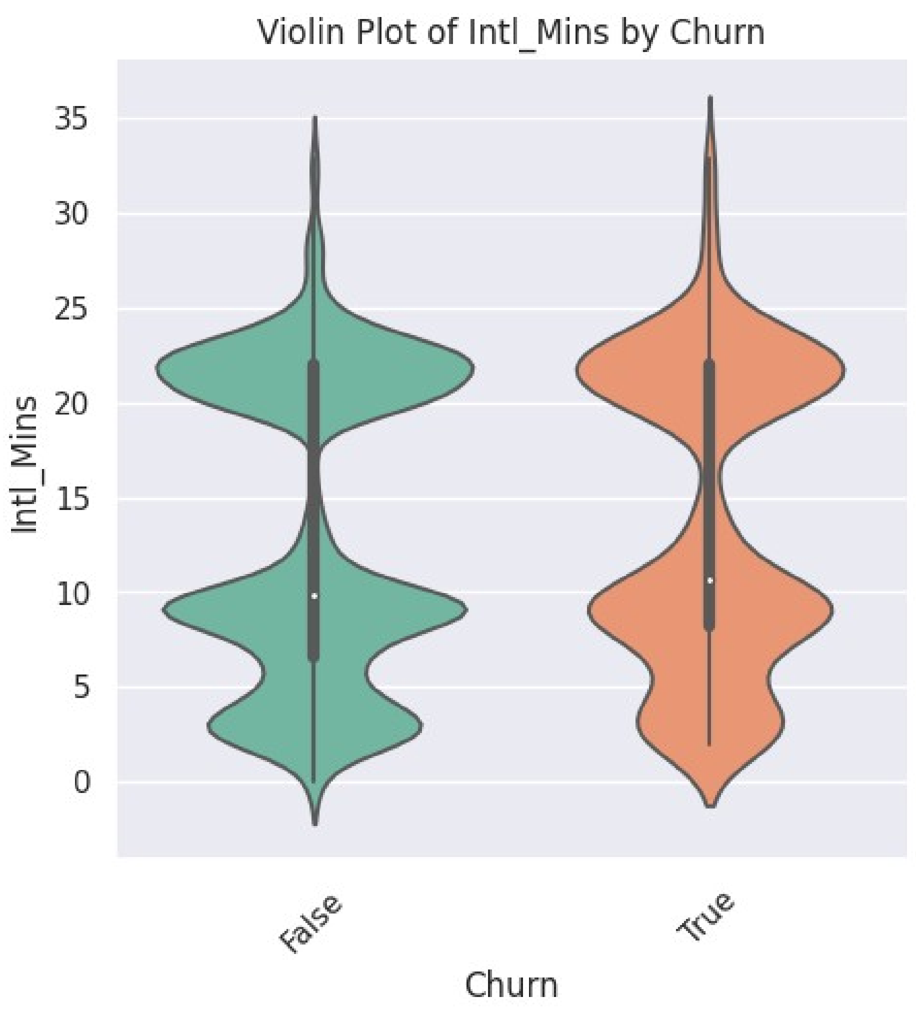


Violin Plot of Night_Mins by Churn
Night_Mins
350
300
250
200
150
False
True
Churn

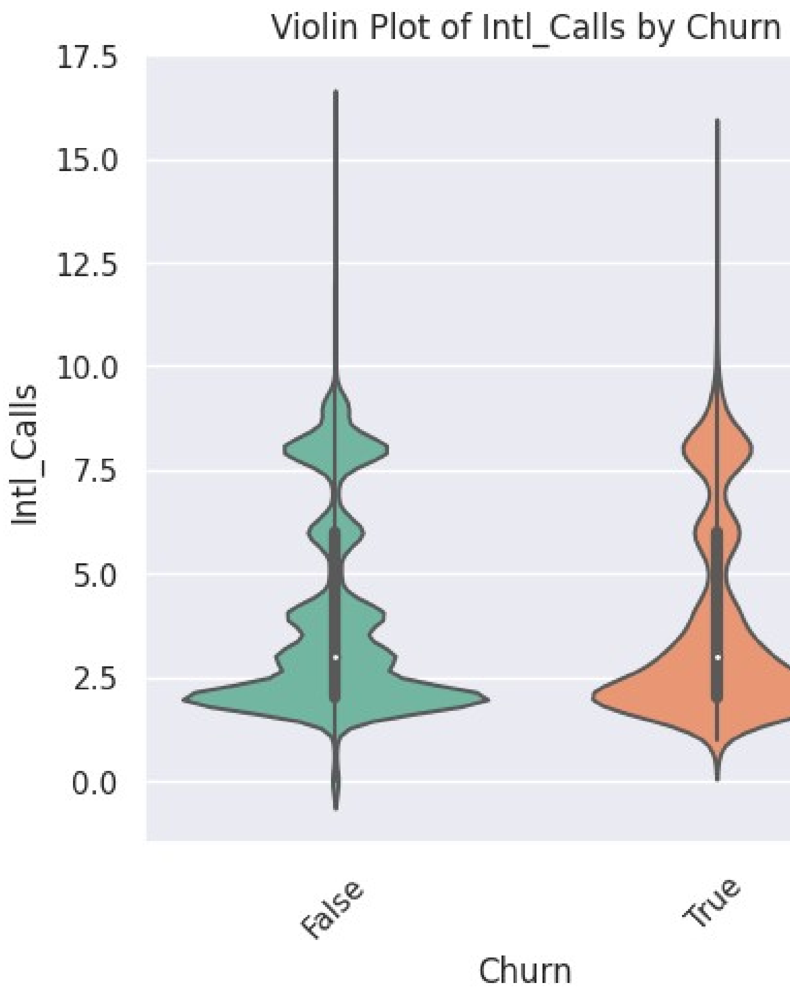


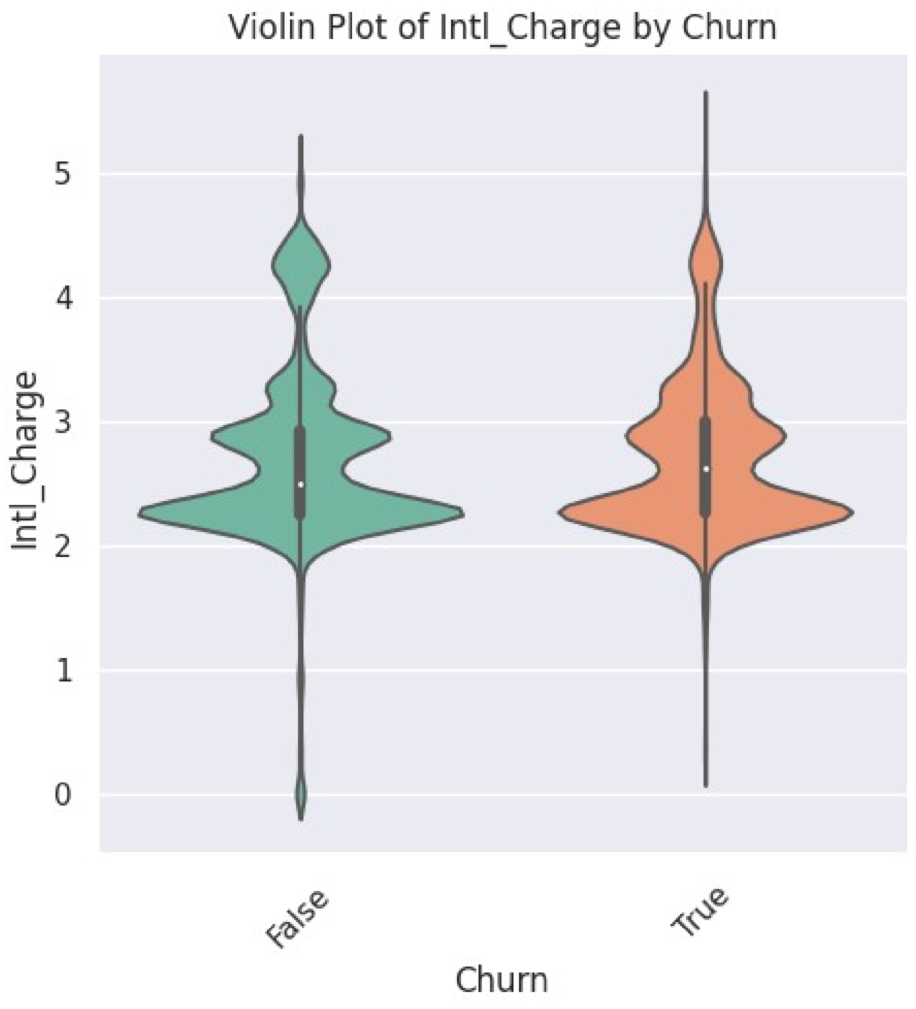
Violin Plot of Night_Calls by Churn
Night_Calls
350
300
250
200
150
100
50
0
False
True
Churn

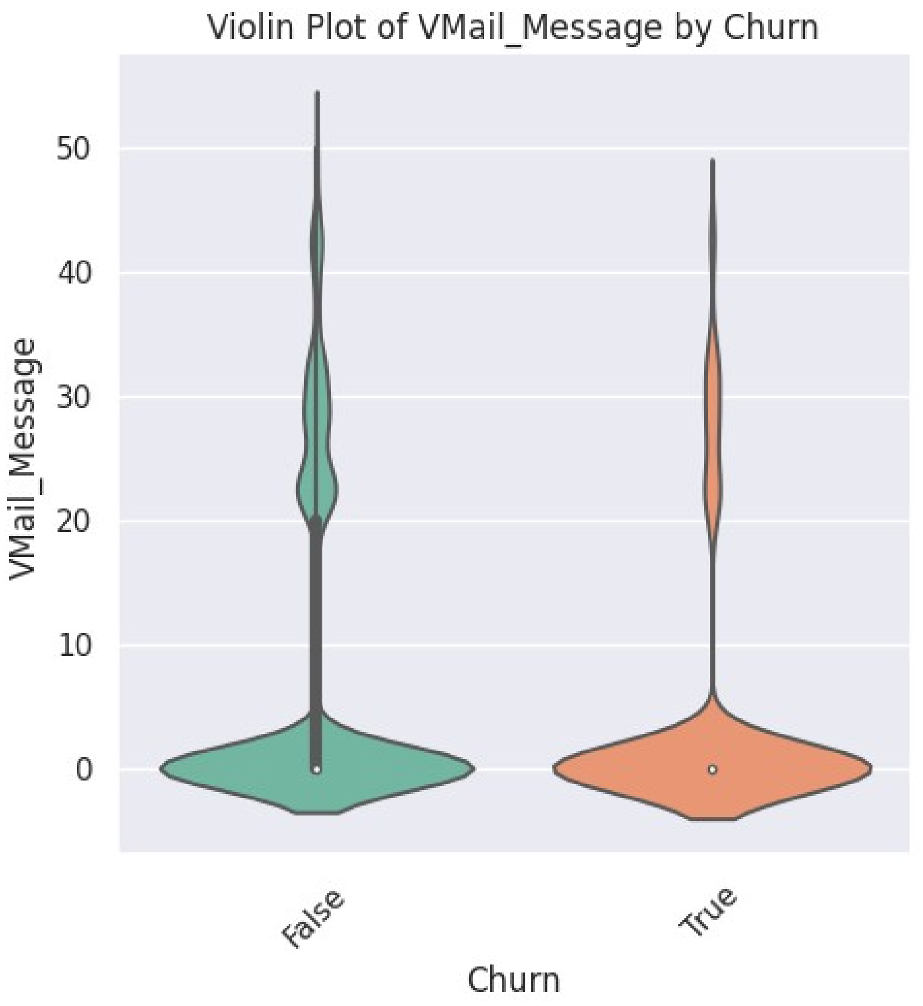

Violin Plot of Night_Charge by Churn
Night_Charge
0
5
10
15
20
25
30
35
False
True
Churn



Violin Plot of Intl_Mins by Churn
Intl_Mins
35
30
25
20
15
10
5
0
False
True
Churn


Violin Plot of Intl_Calls by Churn
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Intl_Calls
False
True
Churn


Violin Plot of Intl_Charge by Churn
Intl_Charge
5
4
3
2
1
0
False
True
Churn



Violin Plot of VMail_Message by Churn
VMail_Message
0
10
20
30
40
50
False
True
Churn