Lab Assignment 02

The objective of this lab assignment is to explore a dataset that contains information from customers of a telephone company (`data_lab_02.csv`). We will analyze the features in the dataset and try to determine which of these features are good indicators of customer churn (that is, loss of customers).

Instructions:

Complete each task and question by filling in the blanks (`...`) with one or more lines of code or text. Each task and question is worth **0.5 points** (out of **10 points**).

Submission:

This assignment is due **Tuesday, September 22, at 11:59PM (Central Time)**.

This assignment must be submitted on Gradescope as a **PDF file** containing the completed code for each task and the corresponding output. To save your Jupyter notebook as a PDF file, go to `File > Export Notebook As > HTML` or `File > Download As > HTML`, open the HTML file and print it as a PDF file. Additionally, this assignment has a single question on Gradescope and **all pages of the PDF file** must be assigned to this question. A **0.5-point (5%) penalty** will be applied to submissions that do not follow these guidelines. For more instructions on how to submit assignments on Gradescope, see this guide.

Late submissions will be accepted within **0-12** hours after the deadline with a **0.5-point (5%) penalty** and within **12-24** hours after the deadline with a **2-point (20%) penalty**. No late submissions will be accepted more than 24 hours after the deadline.

This assignment is individual. Offering or receiving any kind of unauthorized or unacknowledged assistance is a violation of the University's academic integrity policies, will result in a grade of zero for the assignment, and will be subject to disciplinary action.

Part 1: Exploring the Dataset

In [128]:
```
# Load libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [129]:
```
# Load dataset
data = pd.read_csv('data_lab_02.csv')
```

In [130]:
```
# Display the first three rows of the dataset
data.head(3)
```

Out[130]:

	State	Account length	Area code	International plan	Voice mail plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Tot i minut
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	110	10.30	162.6	104	7.32	12

Task 01 (of 15): Display the first three rows and the first three columns of the dataset using the `iloc` and `loc` methods. *Hint:* Remember that the `iloc` method is used for indexing by integer position and the `loc` method is used for indexing by label.

In [131]:
```
data.iloc[[0,1,2], [0,1,2]]
```

Out[131]:

	State	Account length	Area code
0	KS	128	415
1	OH	107	415
2	NJ	137	415

In [132]:
```
data.loc[[0,1,2],['State', 'Account length', 'Area code']]
```

Out[132]:

	State	Account length	Area code
0	KS	128	415
1	OH	107	415
2	NJ	137	415

Task 02 (of 15): Determine the dimensionality of the dataset. Then, display information (data types, number of values) about the features in the dataset. *Hint:* Use methods `shape` and `info`.

In [133]:
```
data.shape
```

Out[133]: `(3333, 20)`

In [134]:
```
data.info
```

Out[134]:
```
<bound method DataFrame.info of      State  Account length  Area code International plan Voice mail p
lan  \
0       KS             128        415                 No             Yes
1       OH             107        415                 No             Yes
2       NJ             137        415                 No              No
3       OH              84        408                Yes              No
4       OK              75        415                Yes              No
...    ...             ...        ...                ...             ...
3328    AZ             192        415                 No             Yes
3329    WV              68        415                 No              No
3330    RI              28        510                 No              No
3331    CT             184        510                Yes              No
3332    TN              74        415                 No             Yes

      Number voice mail messages  Total day minutes  Total day calls  \
0                             25              265.1              110
1                             26              161.6              123
2                              0              243.4              114
3                              0              299.4               71
4                              0              166.7              113
...                          ...                ...              ...
3328                          36              156.2               77
3329                           0              231.1               57
3330                           0              180.8              109
3331                           0              213.8              105
3332                          25              234.4              113

      Total day charge  Total eve minutes  Total eve calls  Total eve charge  \
0                45.07              197.4               99             16.78
1                27.47              195.5              103             16.62
2                41.38              121.2              110             10.30
3                50.90               61.9               88              5.26
4                28.34              148.3              122             12.61
...                ...                ...              ...               ...
3328             26.55              215.5              126             18.32
3329             39.29              153.4               55             13.04
3330             30.74              288.8               58             24.55
3331             36.35              159.6               84             13.57
3332             39.85              265.9               82             22.60

      Total night minutes  Total night calls  Total night charge  \
0                   244.7                 91               11.01
1                   254.4                103               11.45
2                   162.6                104                7.32
3                   196.9                 89                8.86
4                   186.9                121                8.41
...                   ...                ...                 ...
3328                279.1                 83               12.56
3329                191.3                123                8.61
3330                191.9                 91                8.64
3331                139.2                137                6.26
3332                241.4                 77               10.86

      Total intl minutes  Total intl calls  Total intl charge  \
0                   10.0                 3               2.70
1                   13.7                 3               3.70
2                   12.2                 5               3.29
3                    6.6                 7               1.78
4                   10.1                 3               2.73
...                  ...               ...                ...
3328                 9.9                 6               2.67
3329                 9.6                 4               2.59
3330                14.1                 6               3.81
3331                 5.0                10               1.35
3332                13.7                 4               3.70

      Customer service calls  Churn
0                          1  False
1                          1  False
2                          0  False
3                          2  False
4                          3  False
...                      ...    ...
3328                       2  False
3329                       3  False
3330                       2  False
3331                       2  False
3332                       0  False

[3333 rows x 20 columns]>
```

Question 01 (of 05): How many observations and how many features are in the dataset? What are the data types of the features? Are there any missing values?

Answer: There are 3333 observations and 20 features in the dataset. The data types are boolean, integer, float and String. There are NOT missing values

Part 2: Transforming the Features

Task 03 (of 15): Change the data type of feature 'Churn' from bool to int64 and change the values of feature 'International plan' from Yes/No to True/False. *Hint:* Use methods `astype` and `map`.

In [135]:
```
data['Churn'] = data['Churn'].astype('int64', copy=True, errors='raise')
change_values = {'No' : False, 'Yes' : True}
data['International plan'] = data['International plan'].map(change_values)
data.head(3)
```

Out[135]:

	State	Account length	Area code	International plan	Voice mail plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Tot i minut
0	KS	128	415	False	Yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10
1	OH	107	415	False	Yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13
2	NJ	137	415	False	No	0	243.4	114	41.38	121.2	110	10.30	162.6	104	7.32	12

Task 04 (of 15): Create a new numerical feature named 'Total charge' that contains the sum of the day, evening, and night charges. Then, sort the dataset in descending order by total charge. *Hint:* Use method `sort_values`.

In [136]:
```
data['Total charge'] = data['Total day charge'] + data['Total eve charge'] + data['Total night charge'
]
data.sort_values(by=['Total charge'], ascending=False, inplace=True)
data.head(3)
```

Out[136]:

	State	Account length	Area code	International plan	Voice mail plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	...	Total eve charge	Total night minutes	Total night calls	Total night charge	Tot in minute
985	NY	64	415	True	No	0	346.8	55	58.96	249.5	...	21.21	275.4	102	12.39	13
15	NY	161	415	False	No	0	332.9	67	56.59	317.8	...	27.01	160.6	128	7.23	5
365	CO	154	415	False	No	0	350.8	75	59.64	216.5	...	18.40	253.9	100	11.43	10

3 rows × 21 columns

Part 3: Summarizing the Features

Task 05 (of 15): Compute summary statistics for all numerical features and all non-numerical features. *Hint:* Use method `describe` with the appropriate parameters.

In [137]:
```
import numpy as np
data.describe(include=np.number)
```

Out[137]:

	Account length	Area code	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total nig minut
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.00000
mean	101.064806	437.182418	8.099010	179.775098	100.435644	30.562307	200.980348	100.114311	17.083540	200.8720
std	39.822106	42.371290	13.688365	54.467389	20.069084	9.259435	50.713844	19.922625	4.310668	50.5738
min	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	23.2000
25%	74.000000	408.000000	0.000000	143.700000	87.000000	24.430000	166.600000	87.000000	14.160000	167.0000
50%	101.000000	415.000000	0.000000	179.400000	101.000000	30.500000	201.400000	100.000000	17.120000	201.2000
75%	127.000000	510.000000	20.000000	216.400000	114.000000	36.790000	235.300000	114.000000	20.000000	235.3000
max	243.000000	510.000000	51.000000	350.800000	165.000000	59.640000	363.700000	170.000000	30.910000	395.0000

In [138]:
```
data.describe(exclude=np.number)
```

Out[138]:

	State	International plan	Voice mail plan
count	3333	3333	3333
unique	51	2	2
top	WV	False	No
freq	106	3010	2411

Task 06 (of 15): Group the data by feature 'Churn' and compute summary statistics for all numerical variables again. *Hint:* Use method `groupby`.

In [139]:
```
data.groupby('Churn').describe()
```

Out[139]:

	Account length								Area code		...	Customer service calls		Total charge			
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max	count	mean	std	m
Churn																	
0	2850.0	100.793684	39.88235	1.0	73.0	100.0	127.0	243.0	2850.0	437.074737	...	2.0	8.0	2850.0	55.705404	9.454475	1
1	483.0	102.664596	39.46782	1.0	76.0	103.0	127.0	225.0	483.0	437.817805	...	4.0	9.0	483.0	62.466418	13.887371	2

2 rows × 136 columns

Task 07 (of 15): Compute the percentage of churned and non-churned customers. *Hint:* Use method `value_counts` with the appropriate parameters.

In [140]:
```
data['Churn'].value_counts(normalize=True)
```

Out[140]:
```
0    0.855086
1    0.144914
Name: Churn, dtype: float64
```

Task 08 (of 15): Compute the mean values of all numerical features for churned and non-churned customers. Notice the differences and similarities between both groups.

In [141]:
```
data.groupby('Churn').mean()
```

Out[141]:

	Account length	Area code	International plan	Number voice mail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total nigh minutes
Churn											
0	100.793684	437.074737	0.065263	8.604561	175.175754	100.283158	29.780421	199.043298	100.038596	16.918909	200.1331
1	102.664596	437.817805	0.283644	5.115942	206.914079	101.335404	35.175921	212.410145	100.561077	18.054969	205.2316

Question 02 (of 05): What is the percentage of churned customers? What is the mean total charge for churned customers? What is the percentage of non-churned customers? What is the mean total charge for non-churned customers

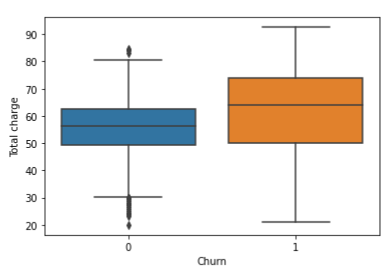
Answer: 14.49% are churned customer, the mean of Total charge is 55.71. 85.5% are non-churned customers, the mean of Total charge is 62.47

Part 4: Visualizing the Features

Task 09 (of 15): Visualize the summary statistics of churned and non-churned customers for feature 'Total charge'. *Hint:* Use function `seaborn.boxplot()` with the apropriate parameters. Make sure you group customers by feature 'Churn'!

In [142]:
```
sns.boxplot(x = 'Churn', y = 'Total charge', data=data)
```

Out[142]: `<AxesSubplot:xlabel='Churn', ylabel='Total charge'>`



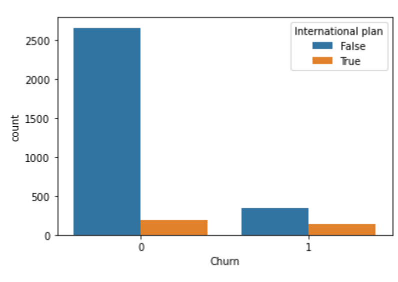
Question 03 (of 05): What do you observe in the plot?

Answer: Churn customer has wider range in between lower quartile to upper quartile. There are 50% of non-churn customer tatal charge in between 50 to 65. There are few outliers churn customer has extreme low tatal charge and extreme high tatal charge.

Task 10 (of 15): Visualize the number of churned and non-churned customers in each category of feature 'International plan'. *Hint:* Use function `seaborn.countplot()` with the apropriate parameters. Make sure you group customers by feature 'Churn'!

In [143]:
```
sns.countplot(x='Churn', hue='International plan', data=data)
```

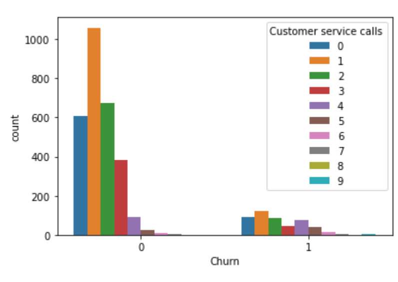
Out[143]: `<AxesSubplot:xlabel='Churn', ylabel='count'>`



Task 11 (of 15): Visualize the number of churned and non-churned customers in each category of feature 'Customer service calls'. *Hint:* Use function `seaborn.countplot()` with the apropriate parameters. Make sure you group customers by feature 'Churn'!

In [144]:
```
sns.countplot(x='Churn', hue='Customer service calls', data=data)
```

Out[144]: `<AxesSubplot:xlabel='Churn', ylabel='count'>`



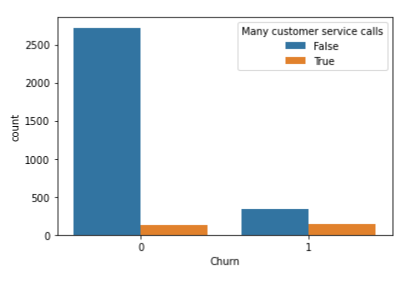
Task 12 (of 15): Create a new Boolean feature named 'Many customer service calls' that indicates whether a user has made more than 3 customer service calls.

In [145]:
```
data['Many customer service calls'] = data.apply (lambda row:True if row['Customer service calls'] > 3
 else False, axis=1)
```

Task 13 (of 15): Visualize the number of churned and non-churned customers in each category of feature 'Many customer service calls'. *Hint:* Use function `seaborn.countplot()` with the apropriate parameters. Make sure you group customers by feature 'Churn'!

In [146]:
```
sns.countplot(x='Churn', hue='Many customer service calls', data=data)
```

Out[146]: `<AxesSubplot:xlabel='Churn', ylabel='count'>`



Question 04 (of 05): What do you observe in the plots?

Answer: The most non-churn customer called customer service about 1 time. Non-churn customer made less 'Many Customer Service Calls' then churn customer

Part 5: Making Conclusions

Task 14 (of 15): Compute the churn rate (percentage of churned customers) for customers without international plan and for customers with international plan. *Hint:* Use method `value_counts`.

In [147]:
```
# Compute churn rate for customers without international plan
tempData = data.loc[data['International plan']==False]
num_churned = tempData['Churn'].value_counts()[1]
num_nonchurned = tempData['Churn'].value_counts()[0]
churn_rate = num_churned/(num_nonchurned + num_churned)
print(churn_rate)
```

```
0.11495016611295682
```

In [148]:
```
# Compute churn rate for customers with international plan
dataWithInternationalPlan = data.loc[data['International plan']==True]
num_churned = dataWithInternationalPlan['Churn'].value_counts()[1]
num_nonchurned = dataWithInternationalPlan['Churn'].value_counts()[0]
churn_rate = num_churned/(num_nonchurned + num_churned)
print(churn_rate)
```

```
0.4241486068111455
```

Task 15 (of 15): Compute the churn rate (percentage of churned customers) for customers with 3 customer service calls or less and for customers with more than 3 service calls. *Hint:* Use method `value_counts`.

In [149]:
```
# Compute churn rate for customers with 3 customer service calls or less
dataLessCalls = data.loc[data['Many customer service calls']==False]
num_churned = dataLessCalls['Churn'].value_counts()[1]
num_nonchurned = dataLessCalls['Churn'].value_counts()[0]
churn_rate = num_churned/(num_nonchurned + num_churned)
print(churn_rate)
```

```
0.11252446183953033
```

In [150]:
```
# Compute churn rate for customers with more than 3 customer service calls
dataManyCalls = data.loc[data['Many customer service calls']==True]
num_churned = dataManyCalls['Churn'].value_counts()[1]
num_nonchurned = dataManyCalls['Churn'].value_counts()[0]
churn_rate = num_churned/(num_nonchurned + num_churned)
print(churn_rate)
```

```
0.5168539325842697
```

Question 05 (of 05): What are your final conclusions from the exploration of features 'International plan' and 'Many customer service calls'? What other tasks would you perform to explore this dataset?

Answer: There are more churn customers has international plans than churn customers has NOT international plans.There are more churn customer made many customer calls than churn customer made less customer calls