EDA on Telecom Churn

Project Summary

Telecom churn analysis is the process of identifying customers who are likely to cancel their service or switch to a different service provider. This is an important problem for telecom companies, as churn can have a significant impact on their revenue and profitability. Orange Telecom's Churn Dataset consists of cleaned customer activity data and a label specifying whether a customer has churned. The goal of the analysis is to understand the factors that contribute to churn and develop strategies to reduce churn by targeting those factors. The first step in conducting an exploratory data analysis (EDA) for telecom churn analysis is data acquisition, which involves obtaining a representative sample of data from the telecom company including customer demographic information, usage patterns, and churn status. The next step is data cleaning, which involves removing any missing or incomplete data and ensuring that the data is in a format that can be easily analyzed. Data visualization involves using plots and charts to visualize the data and identify trends and patterns, while data summarization involves using statistical techniques to summarize the data and understand the relationships between different variables.

To reduce churn and improve customer retention, it is important to take a proactive approach. One effective strategy is to modify the International Plan so that the charges are the same as the normal plan. This will help address any potential dissatisfaction with higher charges for international usage. Communication and asking for feedback often can help to identify and address any issues that may lead to churn. Offering promotions periodically can also help to retain customers, as can focusing on customers experiencing problems in states with high churn rates. Paying attention to your best customers and ensuring they receive the support they need is also important. Regular server maintenance and addressing poor network connectivity issues can also help to reduce churn. Developing a roadmap for onboarding new customers can help to ensure a smooth onboarding process and reduce the likelihood of churn. Analyzing churn when it occurs can provide valuable insights into the factors contributing to churn, which can inform strategies for reducing churn. Finally, it is important to stay competitive by keeping up with industry trends and continuously improving the customer experience.

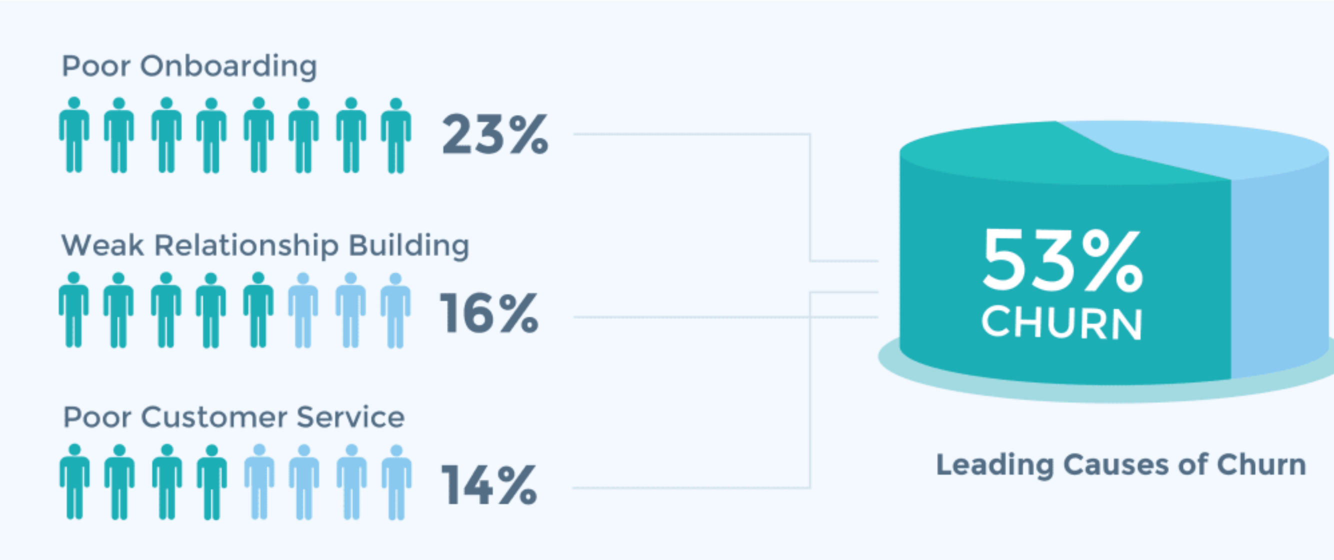
Through EDA of the churn dataset, it was found that the charge fields are directly related to the minute fields, the area code may not be relevant and can be excluded, customers with the International Plan tend to churn more often, customers who have had four or more customer service calls churn significantly more than other customers, customers with high day and evening minute usage tend to churn at a higher rate, and there is no clear relationship between churn and the number of calls made or received. In conclusion, to reduce churn and improve customer retention, telecom companies should focus on modifying the International Plan, being proactive with communication and asking for feedback, offering promotions, focusing on customers experiencing problems in states with high churn rates, paying attention to their best customers, and continuously improving the customer experience.

Problem Statement

Telecom churn analysis is the process of identifying customers who are likely to cancel their service or switch to a different service provider. This is an important problem for telecom companies, as churn can have a significant impact on their revenue and profitability.

GitHub Link

https://github.com/rahulinchal/Telecom-Churn (https://github.com/rahulinchal/Telecom-Churn)



Importing Important Packages

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
%matplotlib inline

#filtering out the warnings
import warnings
warnings.filterwarnings('ignore')
```

Loading the dataset

In [2]:

```
data = pd.read_csv("https://raw.githubusercontent.com/rahulinchal/Telecom-Churn/main/Tel
data.head()
```

Out[2]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

Making a copy of data

In [3]:

```
df = data.copy()
df.head()
```

Out[3]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

Data Description

Name	Description	Value Type	Statistical Type
State	State abbreviation (like KS = Kansas)	String	Categorical
Account length	How long the client has been with the company	Numerical	Quantitative
Area code	Phone number prefix	Numerical	Categorical
International plan	International plan (on/off)	String, "Yes"/"No"	Categorical/Binary
Voice mail plan	Voicemail (on/off)	String, "Yes"/"No"	Categorical/Binary
Number vmail messages	Number of voicemail messages	Numerical	Quantitative
Total day minutes	Total duration of daytime calls	Numerical	Quantitative
Total day calls	Total number of daytime calls	Numerical	Quantitative
Total day charge	Total charge for daytime services	Numerical	Quantitative
Total eve minutes	Total duration of evening calls	Numerical	Quantitative
Total eve calls	Total number of evening calls	Numerical	Quantitative
Total eve charge	Total charge for evening services	Numerical	Quantitative
Total night minutes	Total duration of nighttime calls	Numerical	Quantitative
Total night calls	Total number of nighttime calls	Numerical	Quantitative
Total night charge	Total charge for nighttime services	Numerical	Quantitative
Total intl minutes	Total duration of international calls	Numerical	Quantitative
Total intl calls	Total number of international calls	Numerical	Quantitative
Total intl charge	Total charge for international calls	Numerical	Quantitative
Customer service calls	Number of calls to customer service	Numerical	Categorical/Ordinal

Finding the null values

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
State                     0
Account length            0
Area code                 0
International plan        0
Voice mail plan           0
Number vmail messages     0
Total day minutes         0
Total day calls           0
Total day charge          0
Total eve minutes         0
Total eve calls           0
Total eve charge          0
Total night minutes       0
Total night calls         0
Total night charge        0
Total intl minutes        0
Total intl calls          0
Total intl charge         0
Customer service calls    0
Churn                     0
dtype: int64
```

There are no null values

Finding the duplcates

In [5]:

```
df.duplicated().value_counts()
```

Out[5]:

```
False    3333
dtype: int64
```

There are no duplicates

In [6]:

```
df.shape
```

Out[6]:

```
(3333, 20)
```

The data is clean and has no duplicate values.

In [7]:

```
# Getting the Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   State                   3333 non-null   object 
 1   Account length          3333 non-null   int64  
 2   Area code               3333 non-null   int64  
 3   International plan      3333 non-null   object 
 4   Voice mail plan         3333 non-null   object 
 5   Number vmail messages   3333 non-null   int64  
 6   Total day minutes       3333 non-null   float64
 7   Total day calls         3333 non-null   int64  
 8   Total day charge        3333 non-null   float64
 9   Total eve minutes       3333 non-null   float64
 10  Total eve calls         3333 non-null   int64  
 11  Total eve charge        3333 non-null   float64
 12  Total night minutes     3333 non-null   float64
 13  Total night calls       3333 non-null   int64  
 14  Total night charge      3333 non-null   float64
 15  Total intl minutes      3333 non-null   float64
 16  Total intl calls        3333 non-null   int64  
 17  Total intl charge       3333 non-null   float64
 18  Customer service calls  3333 non-null   int64  
 19  Churn                   3333 non-null   bool   
dtypes: bool(1), float64(8), int64(8), object(3)
memory usage: 498.1+ KB
```

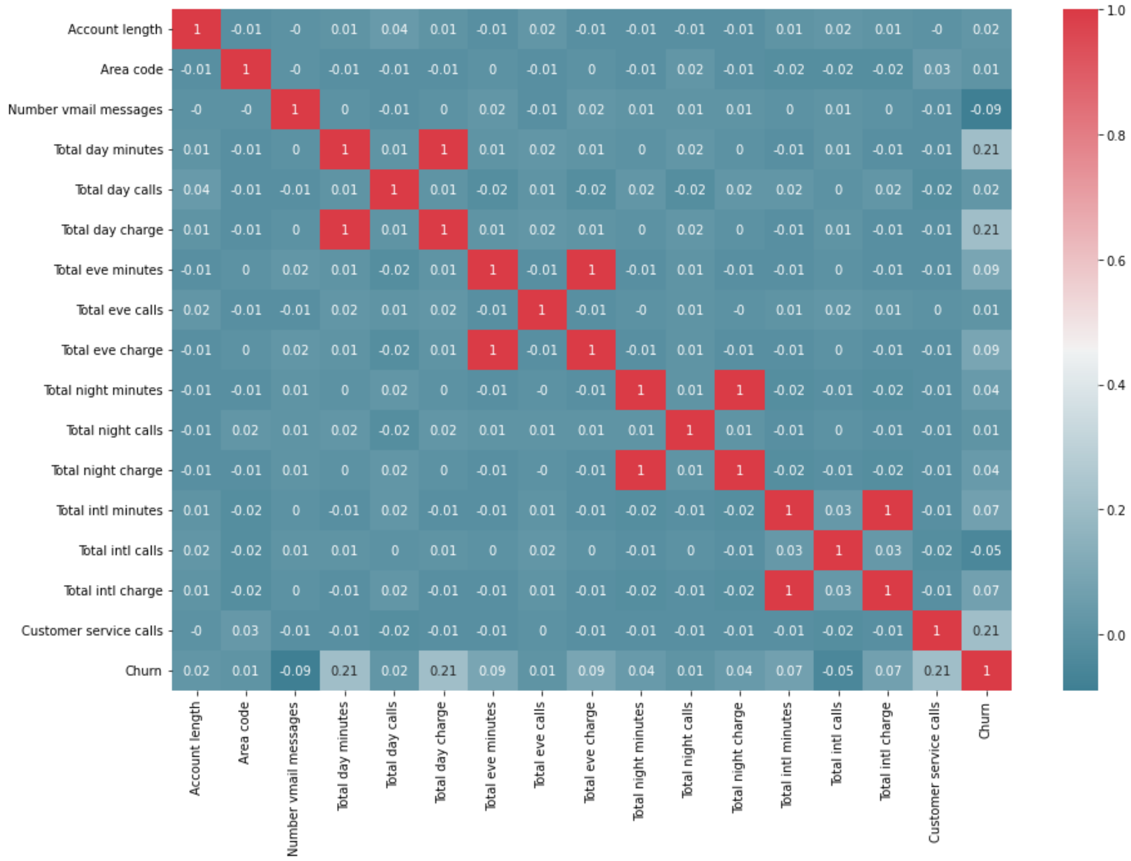
Heatmap for correlation

In [8]:

```
plt.figure(figsize=(15,10))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
sns.heatmap(np.round(df.corr(),2),annot=True, cmap=cmap)
```

Out[8]:

```
<AxesSubplot:>
```



Observation

From the above correlation heatmap, we can see total day charge & total day minute, total evening charge & total evening minute, total night charge & total night minute are positiveliy highly correlated with a value of 1.

Customer service call is positively correlated only with area code and negative correlated with rest variables.

Rest all correlation can be depicted from the above chart.

Data Visualization

1. Checking for how many customer leave the company (Churn).

In [9]:

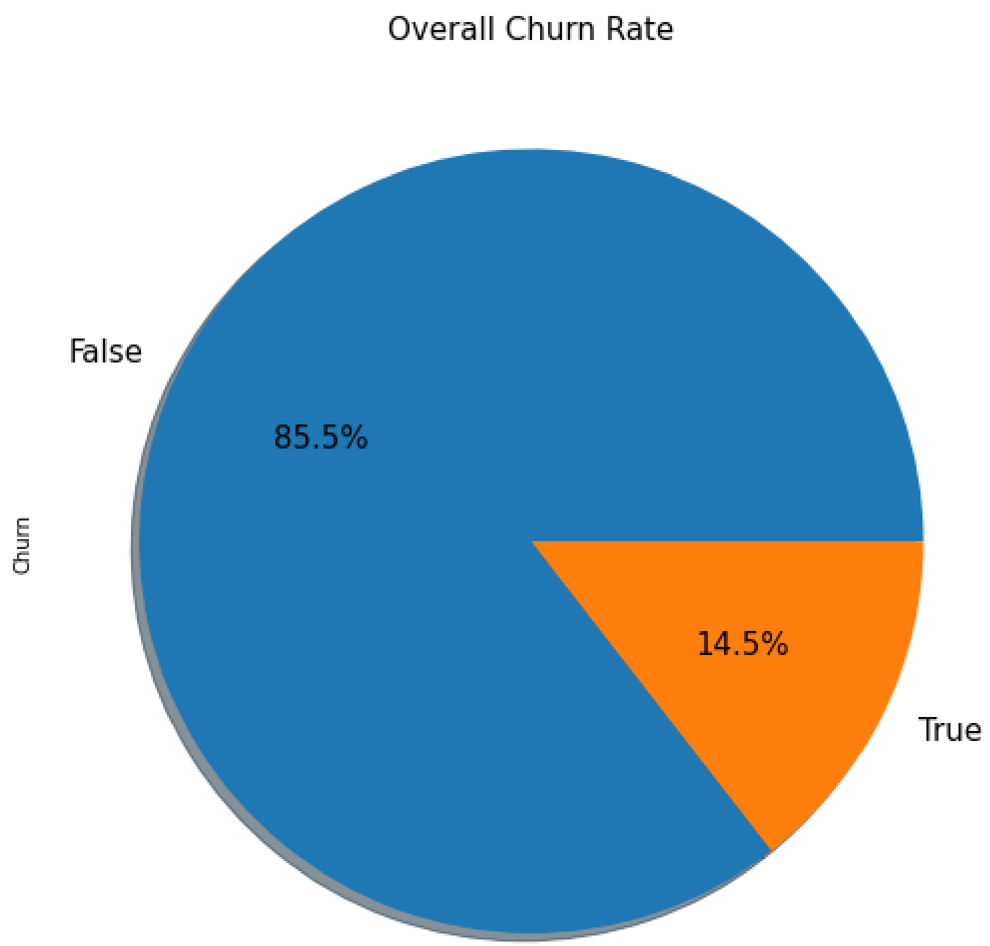
```
df['Churn'].value_counts()
```

Out[9]:

```
False    2850
True      483
Name: Churn, dtype: int64
```

In [10]:

```
# Visualizing using pie chart.
textprops = {"fontsize":15} # Font size of text in pie chart
plt.figure(figsize = (9,9)) # fixing pie chart size
df['Churn'].value_counts().plot(kind = 'pie', autopct = '%1.1f%%', shadow = True, textpr
plt.title("Overall Churn Rate", fontsize = 15)
plt.show()
```



Observation

The retention value is only 14.4% that counts to only 483 out of 3333 people.

2. Checking for churn rate area code wise

In [11]:

```
df.head()
```

Out[11]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [12]:

```
# Checking for unique values
df['Area code'].unique()
```

Out[12]:

```
array([415, 408, 510], dtype=int64)
```

In [13]:

```
# Grouping the churn with area code
area_churn = df.groupby(['Area code'])['Churn'].value_counts().unstack()
area_churn
```

Out[13]:

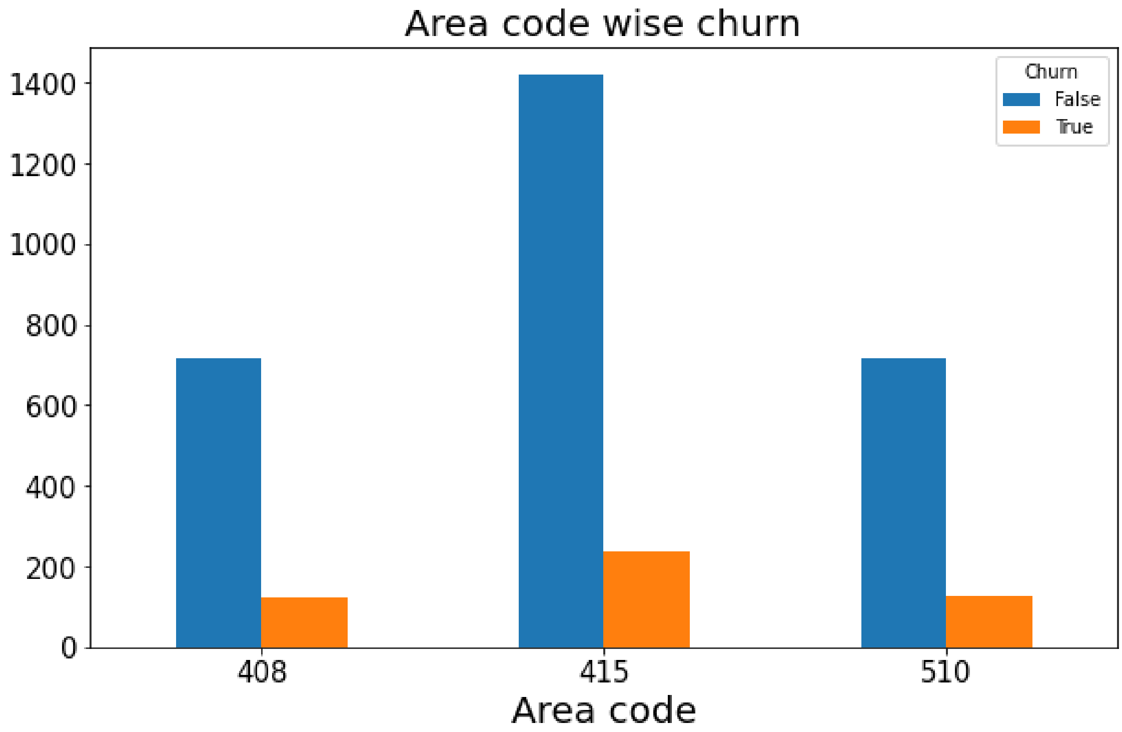
Churn	False	True
Area code		
408	716	122
415	1419	236
510	715	125

In [14]:

```
# Visulaizing the area wise churn
area_churn.plot(kind = 'bar', figsize = (10,6), fontsize = 15)
plt.xticks(rotation = 360)
plt.xlabel("Area code", fontsize = 20)
plt.title("Area code wise churn", fontsize = 20)
```

Out[14]:

```
Text(0.5, 1.0, 'Area code wise churn')
```



Observation

Area code 415 has the highest Churn rate comapred to other areas.

3. Checking for churn rate State wise

In [15]:

```
df.head()
```

Out[15]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [16]:

```
df['State'].unique()
```

Out[16]:

```
array(['KS', 'OH', 'NJ', 'OK', 'AL', 'MA', 'MO', 'LA', 'WV', 'IN', 'RI',
       'IA', 'MT', 'NY', 'ID', 'VT', 'VA', 'TX', 'FL', 'CO', 'AZ', 'SC',
       'NE', 'WY', 'HI', 'IL', 'NH', 'GA', 'AK', 'MD', 'AR', 'WI', 'OR',
       'MI', 'DE', 'UT', 'CA', 'MN', 'SD', 'NC', 'WA', 'NM', 'NV', 'DC',
       'KY', 'ME', 'MS', 'TN', 'PA', 'CT', 'ND'], dtype=object)
```

In [17]:

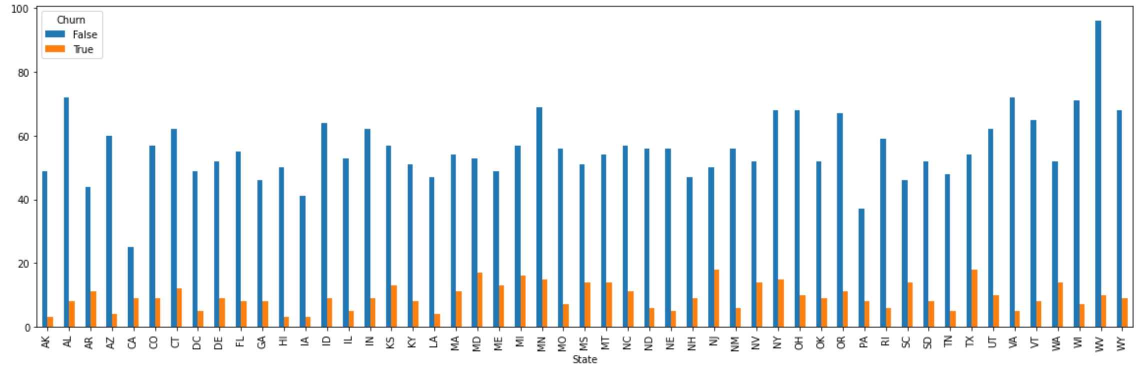
```
state_churn = df.groupby(['State'])['Churn'].value_counts().unstack()
state_churn.head()
```

Out[17]:

Churn	False	True
State		
AK	49	3
AL	72	8
AR	44	11
AZ	60	4
CA	25	9

In [18]:

```
state_churn.plot(kind = 'bar', figsize = (20,6))
plt.show()
```



4. International Plans

In [19]:

```
df.head()
```

Out[19]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [20]:

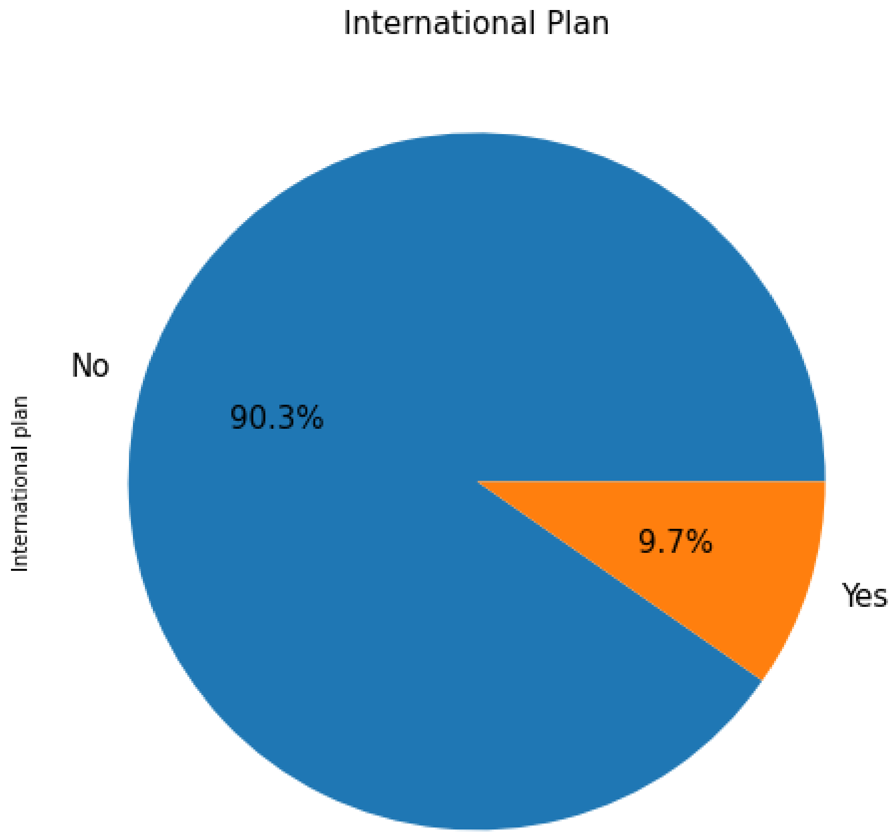
```
int_plan = df['International plan'].value_counts()
int_plan
```

Out[20]:

```
No     3010
Yes     323
Name: International plan, dtype: int64
```

In [21]:

```
int_plan.plot(kind = 'pie', figsize = (8,8), fontsize = 15, autopct = '%1.1f%%')
plt.title("International Plan", fontsize = 15)
plt.show()
```



Observation

90.3% of the people have no international plans and only 9.7% of the people have intenational plan.

5. Checking the churn on the basis of International plan

In [22]:

```
df.head()
```

Out[22]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [23]:

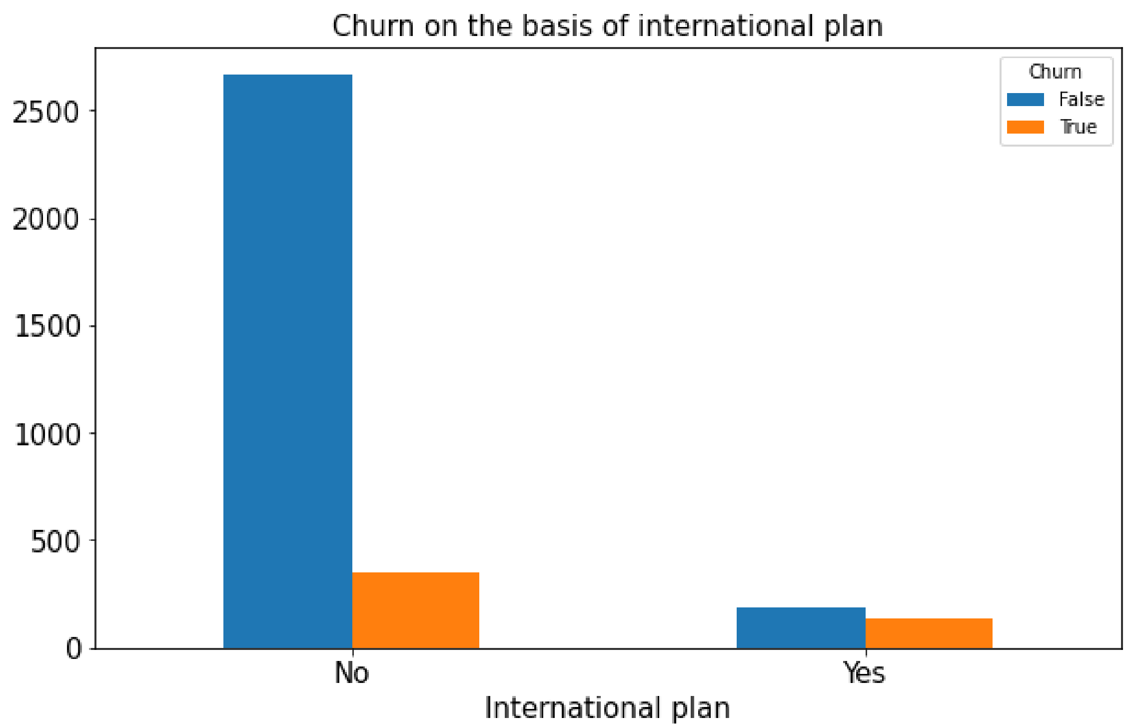
```
international_churn = df.groupby(['International plan'])['Churn'].value_counts().unstack
international_churn
```

Out[23]:

Churn	False	True
International plan		
No	2664	346
Yes	186	137

In [24]:

```
plot2 = international_churn.plot(kind = 'bar', figsize = (10,6), fontsize = 15)
plt.title("Churn on the basis of international plan", fontsize = 15)
plt.xlabel("International plan", fontsize = 15)
plt.xticks(rotation = 360)
plt.show()
```



Observation

Looks like the people who dont have international plan have more churn compared to those who have international plan.

6. Checking churn on the basis of Voice mail plan

In [25]:

```
df.head()
```

Out[25]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [26]:

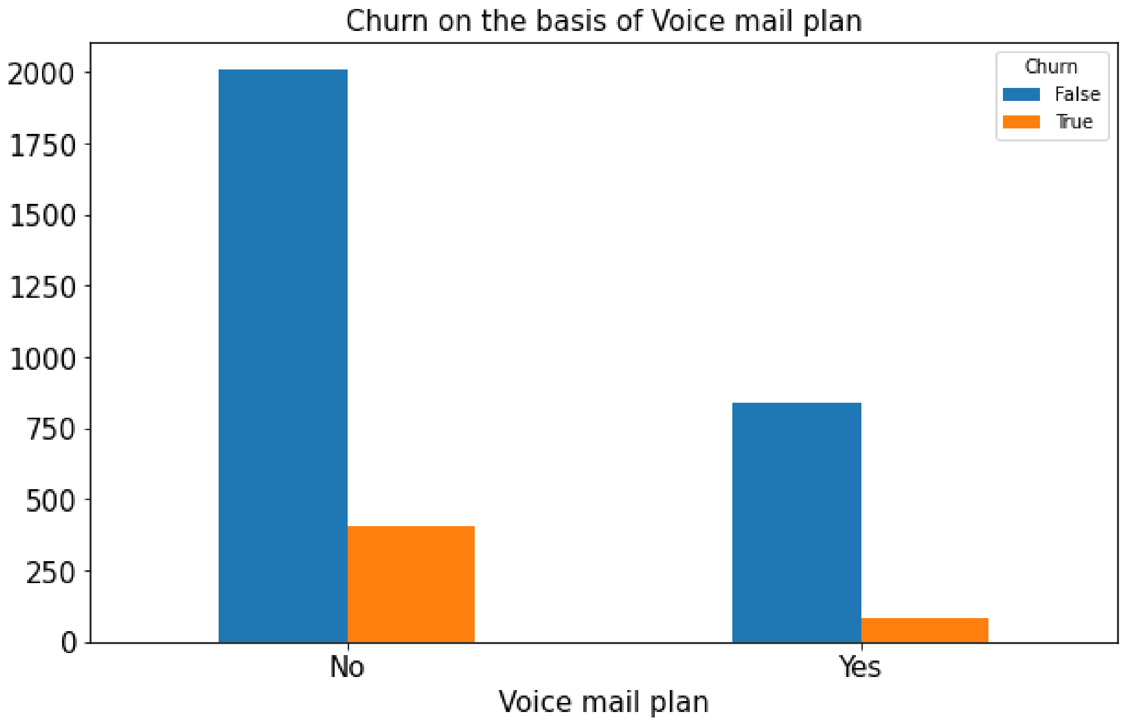
```
voice_churn = df.groupby(['Voice mail plan'])['Churn'].value_counts().unstack()
voice_churn
```

Out[26]:

Churn	False	True
Voice mail plan		
No	2008	403
Yes	842	80

In [27]:

```
voice_churn.plot(kind = 'bar', figsize = (10,6), fontsize = 15)
plt.title("Churn on the basis of Voice mail plan", fontsize = 15)
plt.xlabel("Voice mail plan", fontsize = 15)
plt.xticks(rotation = 360)
plt.show()
```



Observation

Churn is more to the people who has no voice mail plan compared to the people who has voice mail plan.

7. Checking the churn customers who are having both International and voice mail plan

In [28]:

```
df.head()
```

Out[28]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [29]:

```
# Getting those data who has international plan and voice mail plan
voice_and_int = df[df['International plan'] == 'Yes']
voice_and_int = voice_and_int[voice_and_int['Voice mail plan']  == 'Yes']

# Getting those data who has international plan but no voice mail plan
voice_and_int2 = df[df['International plan'] == 'Yes']
voice_and_int2 = voice_and_int2[voice_and_int2['Voice mail plan']  == 'No']

# Getting those data who dont have international plan but have voice mail plan
voice_and_int3 = df[df['International plan'] == 'No']
voice_and_int3 = voice_and_int3[voice_and_int3['Voice mail plan']  == 'Yes']

# Getting those data who dont have international plan and voice mail plan
voice_and_int4 = df[df['International plan'] == 'No']
voice_and_int4 = voice_and_int4[voice_and_int4['Voice mail plan']  == 'No']
```

In [30]:

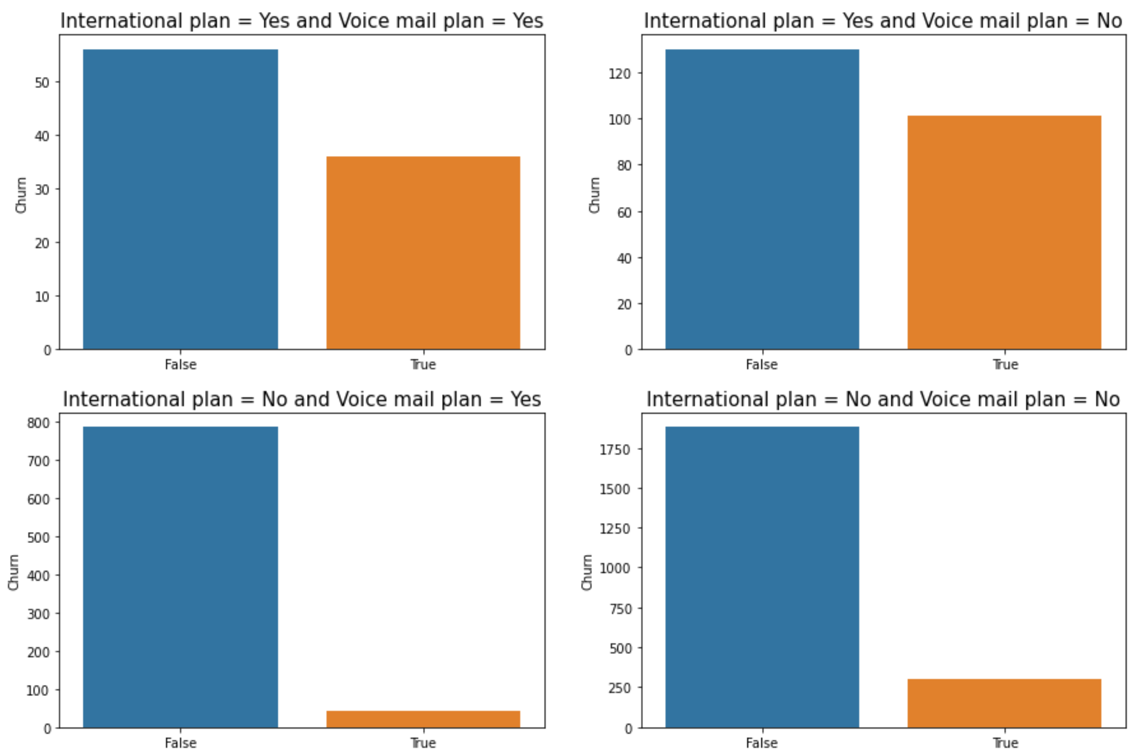
```
plt.figure(figsize = (15,10))

# Checking the churn customers who are having both International and voice mail plan
plt.subplot(2,2,1)
sns.barplot(data = voice_and_int, x = voice_and_int['Churn'].value_counts().keys(),
           y = voice_and_int['Churn'].value_counts())
plt.title("International plan = Yes and Voice mail plan = Yes", fontsize = 15)

# Checking the churn customers who are having International plan but not the voice mail
plt.subplot(2,2,2)
sns.barplot(data = voice_and_int2, x = voice_and_int2['Churn'].value_counts().keys(),
           y = voice_and_int2['Churn'].value_counts())
plt.title("International plan = Yes and Voice mail plan = No", fontsize = 15)

# Checking the churn customers who are not having International plan but have voice mail
plt.subplot(2,2,3)
sns.barplot(data = voice_and_int3, x = voice_and_int3['Churn'].value_counts().keys(),
           y = voice_and_int3['Churn'].value_counts())
plt.title("International plan = No and Voice mail plan = Yes", fontsize = 15)

# Checking the churn customers who are neither having International plan nor have voice
plt.subplot(2,2,4)
sns.barplot(data = voice_and_int4, x = voice_and_int4['Churn'].value_counts().keys(),
           y = voice_and_int4['Churn'].value_counts())
plt.title("International plan = No and Voice mail plan = No", fontsize = 15)
plt.show()
```



Observation

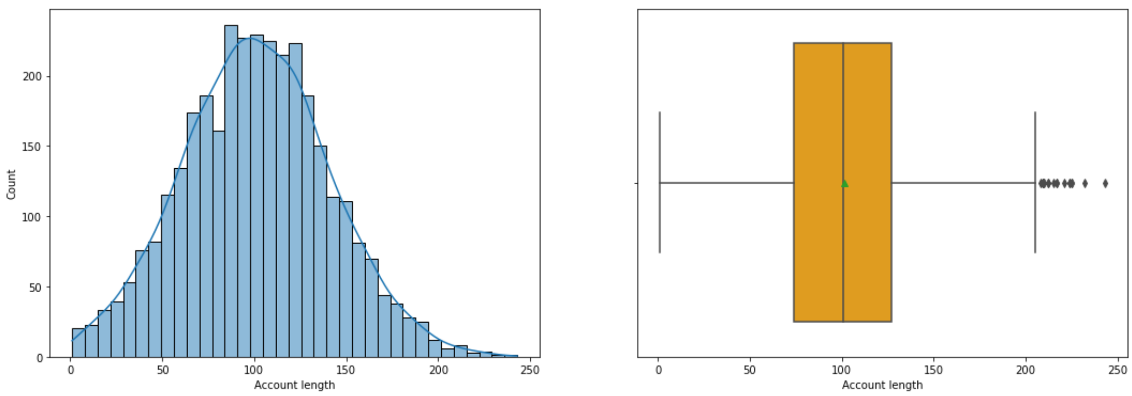
The churn is most to the people who dont have international plan but have voice mail plan The ch

8. Column wise Histogram and Box Plot (Univariate)

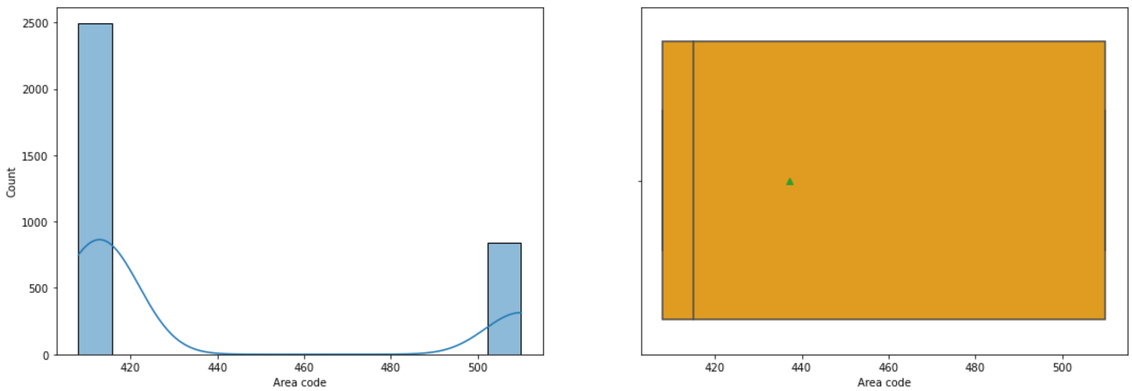
In [31]:

```
# Visualizing code of histogram plot & boxplot for each columns to know the data distrib
for col in df.describe().columns:
    fig,axes = plt.subplots(nrows=1,ncols=2,figsize=(18,6))
    sns.histplot(df[col], ax = axes[0],kde = True)
    sns.boxplot(df[col], ax = axes[1],orient='h',showmeans=True,color='orange')
    fig.suptitle("Distribution plot of "+ col, fontsize = 15)
    plt.show()
```

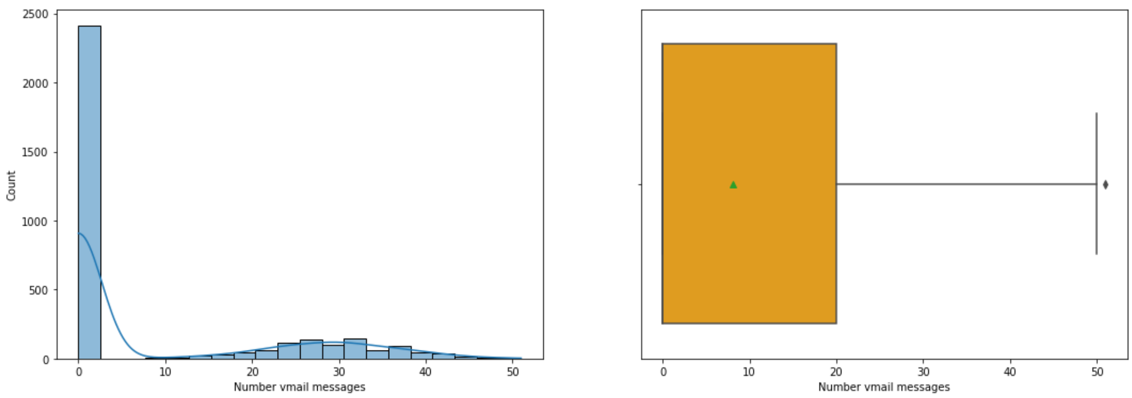
Distribution plot of Account length



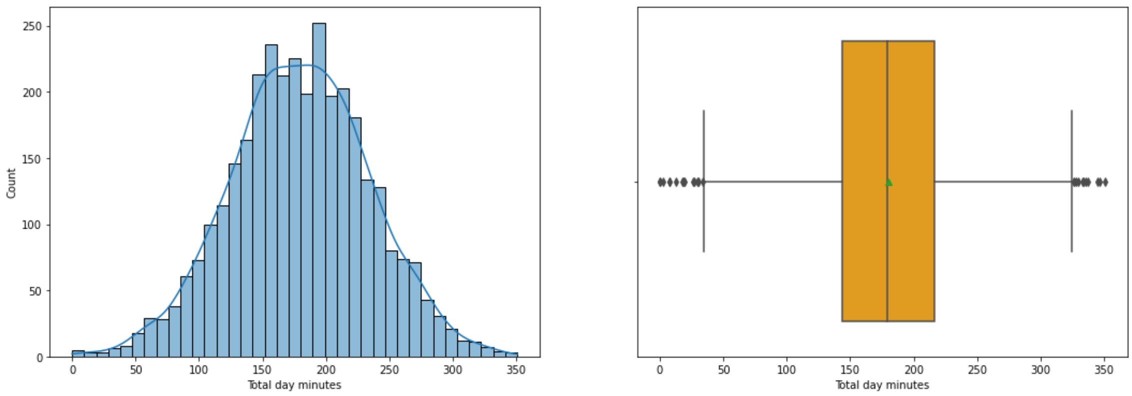
Distribution plot of Area code



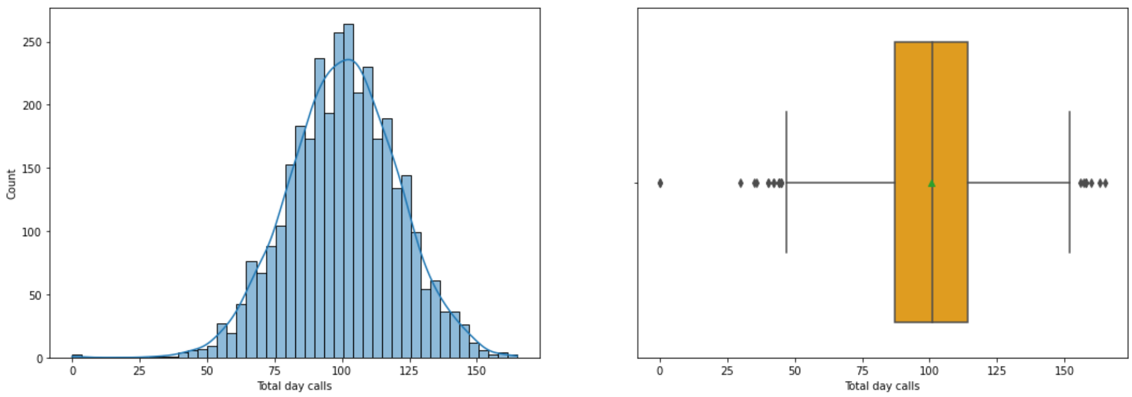
Distribution plot of Number vmail messages



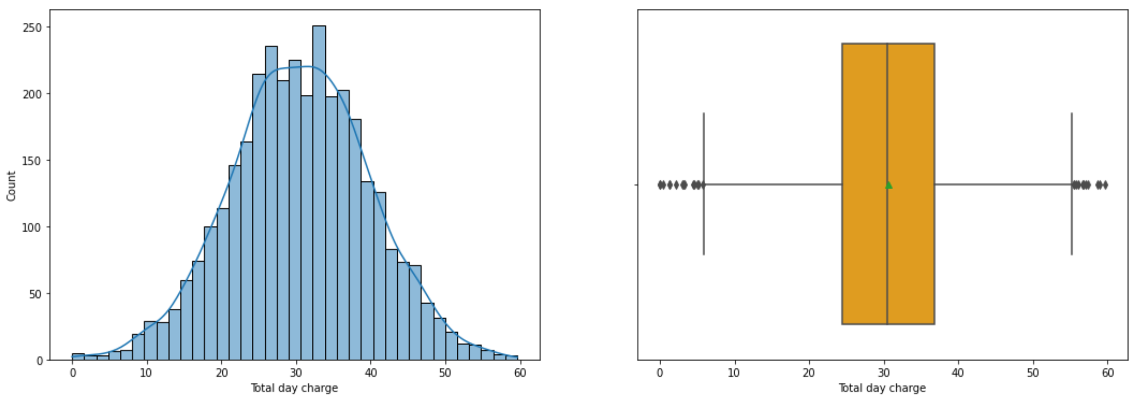
Distribution plot of Total day minutes



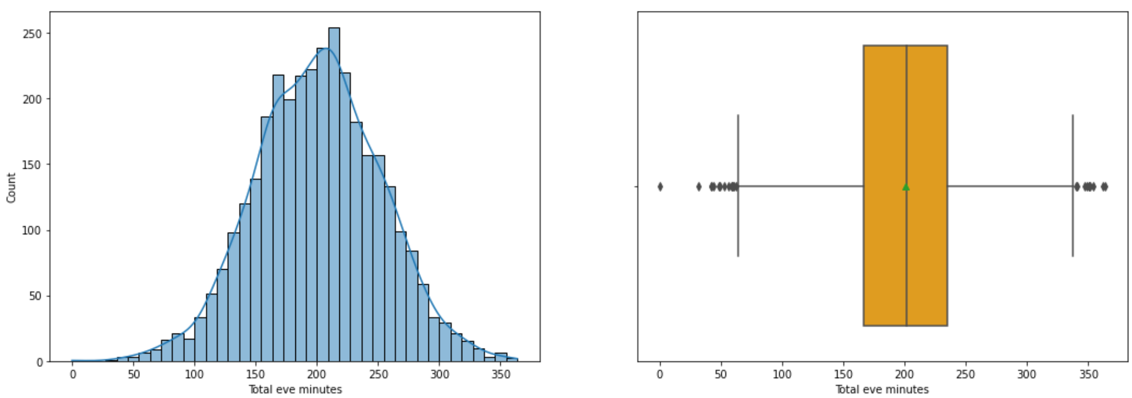
Distribution plot of Total day calls



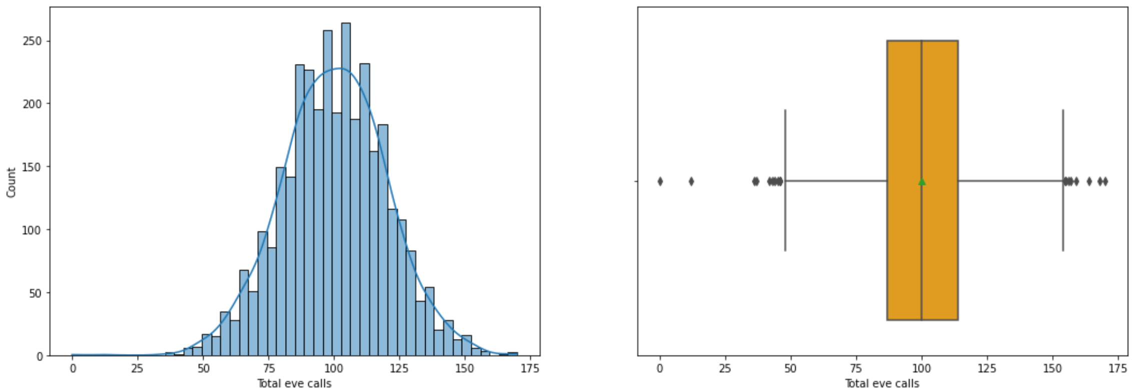
Distribution plot of Total day charge



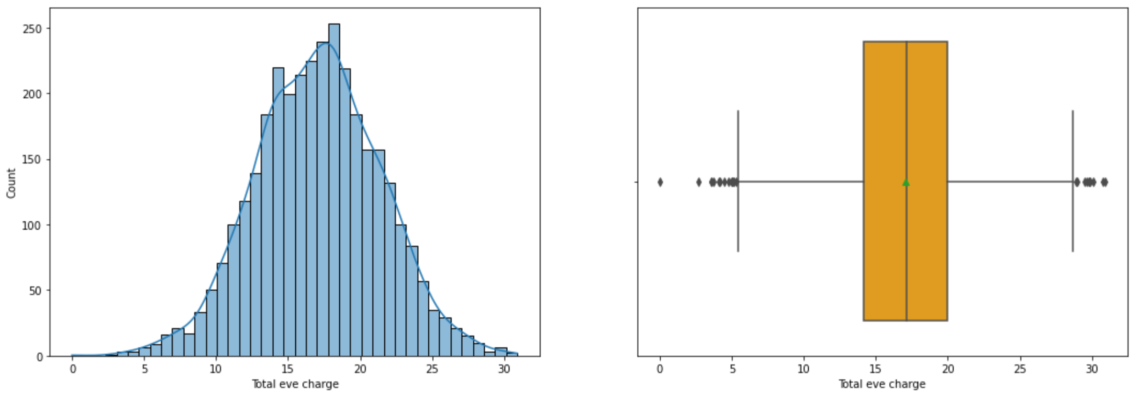
Distribution plot of Total eve minutes



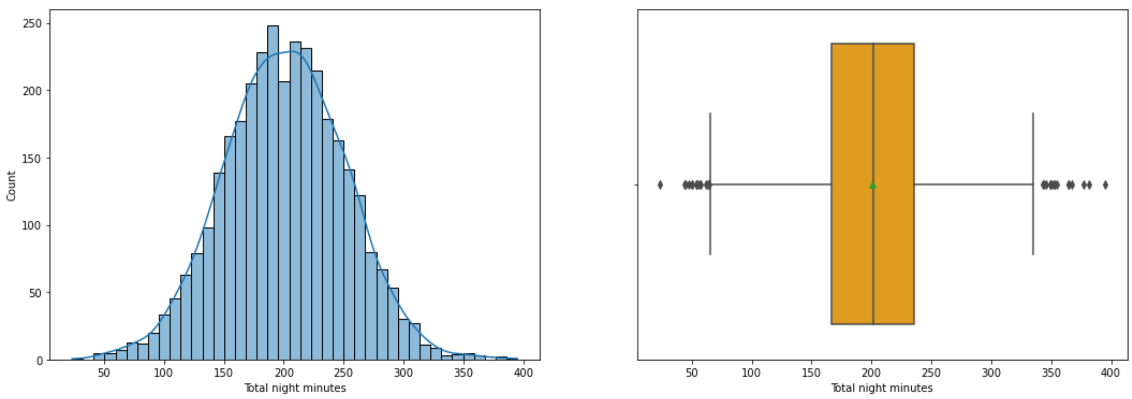
Distribution plot of Total eve calls



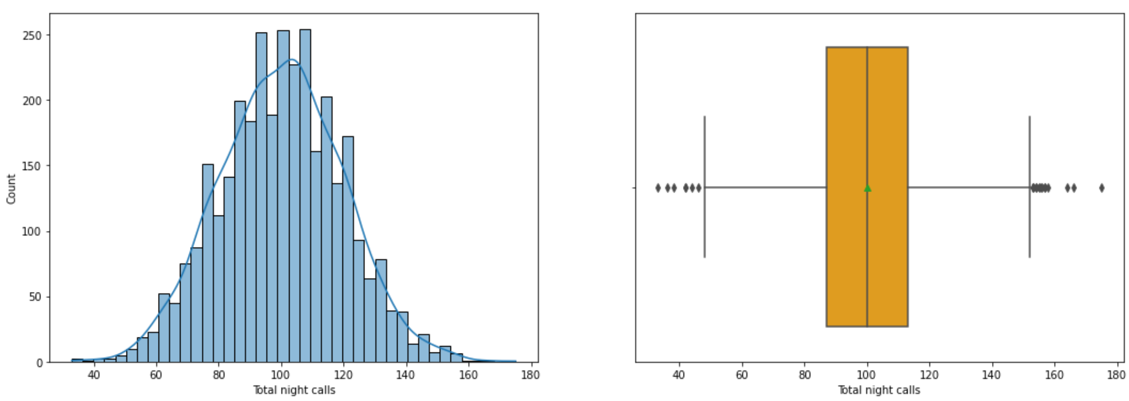
Distribution plot of Total eve charge



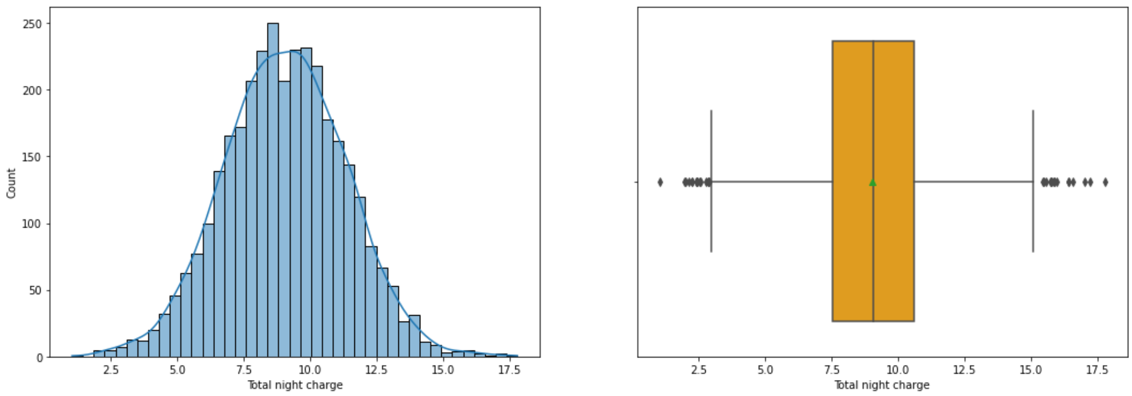
Distribution plot of Total night minutes



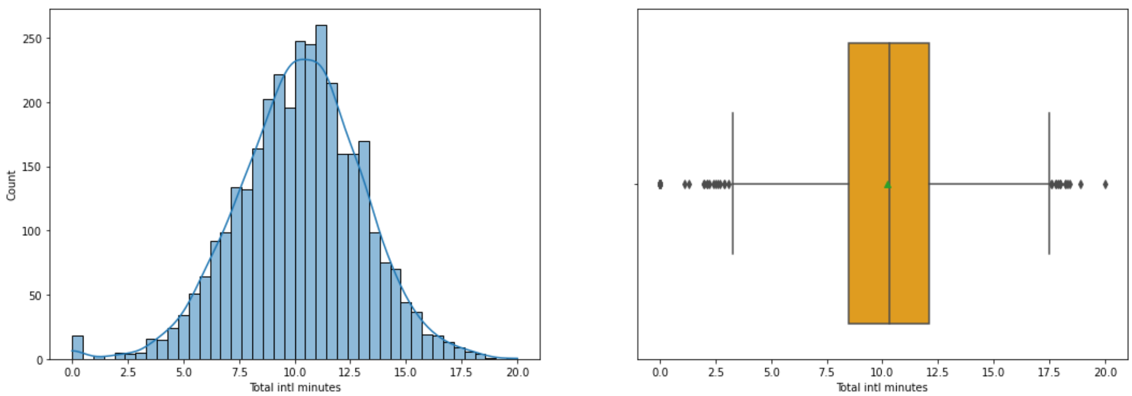
Distribution plot of Total night calls



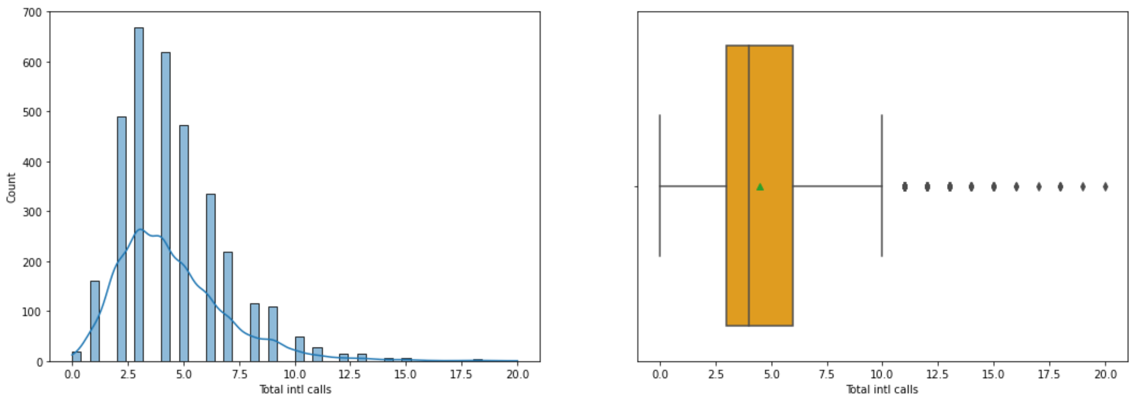
Distribution plot of Total night charge



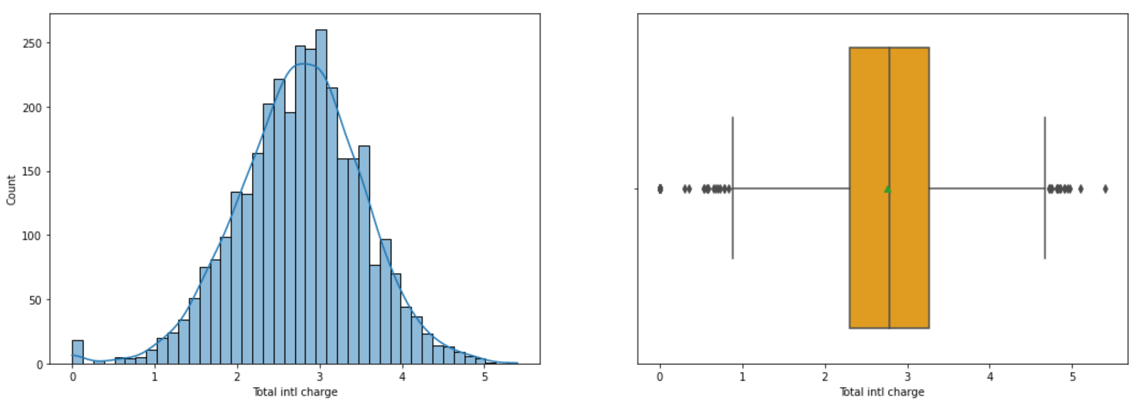
Distribution plot of Total intl minutes

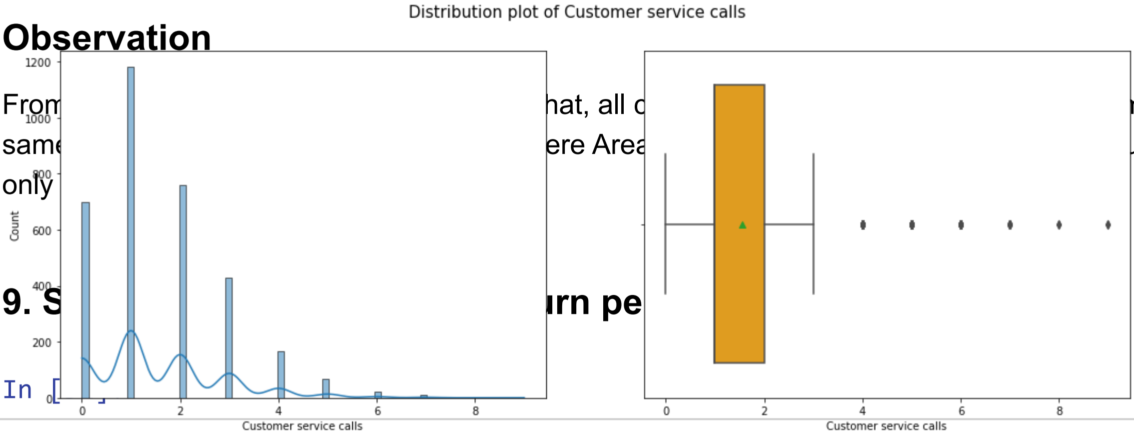


Distribution plot of Total intl calls



Distribution plot of Total intl charge





Observation

From ... hat, all c ... nean is nearly same ... ere Area ... ue as there are only

9. S ... urn pe

In [ ]:

```
df.head()
```

Out[32]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [33]:

```
df['Churn'].value_counts()
```

Out[33]:

```
False    2850
True      483
Name: Churn, dtype: int64
```

In [34]:

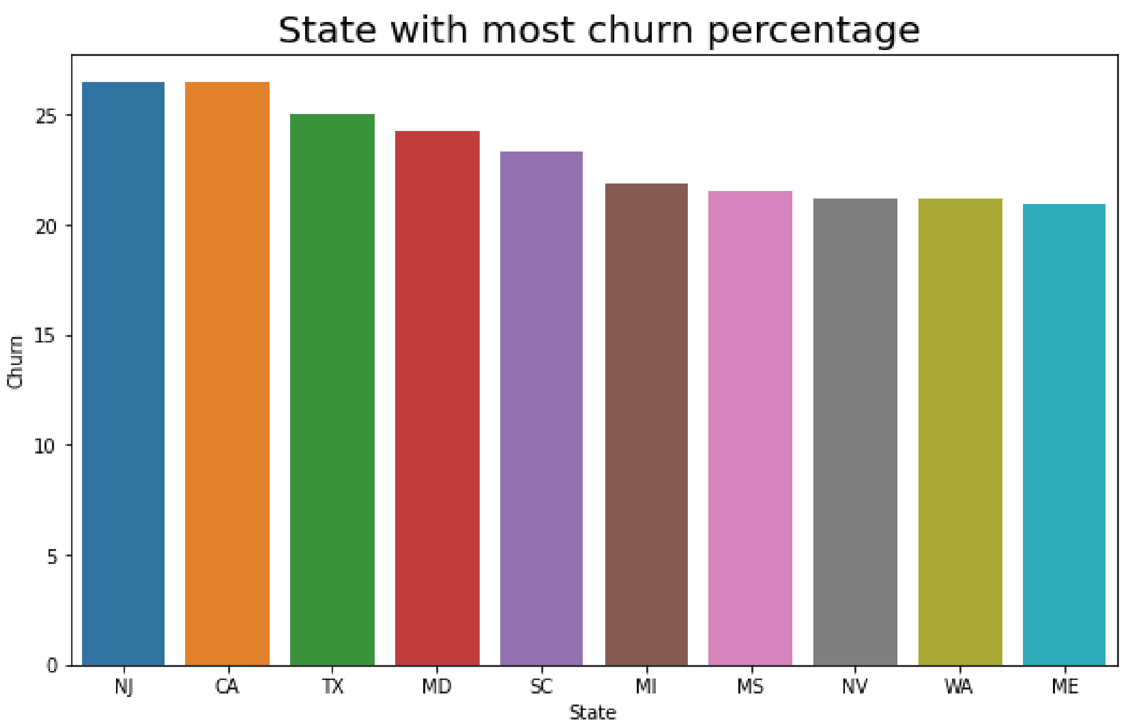
```
to_state_churn = ((df.groupby(['State'])['Churn'].mean()*100).sort_values(ascending = Fa
to_state_churn
```

Out[34]:

	State	Churn
0	NJ	26.470588
1	CA	26.470588
2	TX	25.000000
3	MD	24.285714
4	SC	23.333333
5	MI	21.917808
6	MS	21.538462
7	NV	21.212121
8	WA	21.212121
9	ME	20.967742

In [35]:

```
plt.figure(figsize = (10,6))
sns.barplot(data = to_state_churn, x = to_state_churn['State'], y = to_state_churn['Chur
plt.title(" State with most churn percentage", fontsize = 20)
plt.show()
```



10. State vs Average bottom true churn percentage

In [36]:

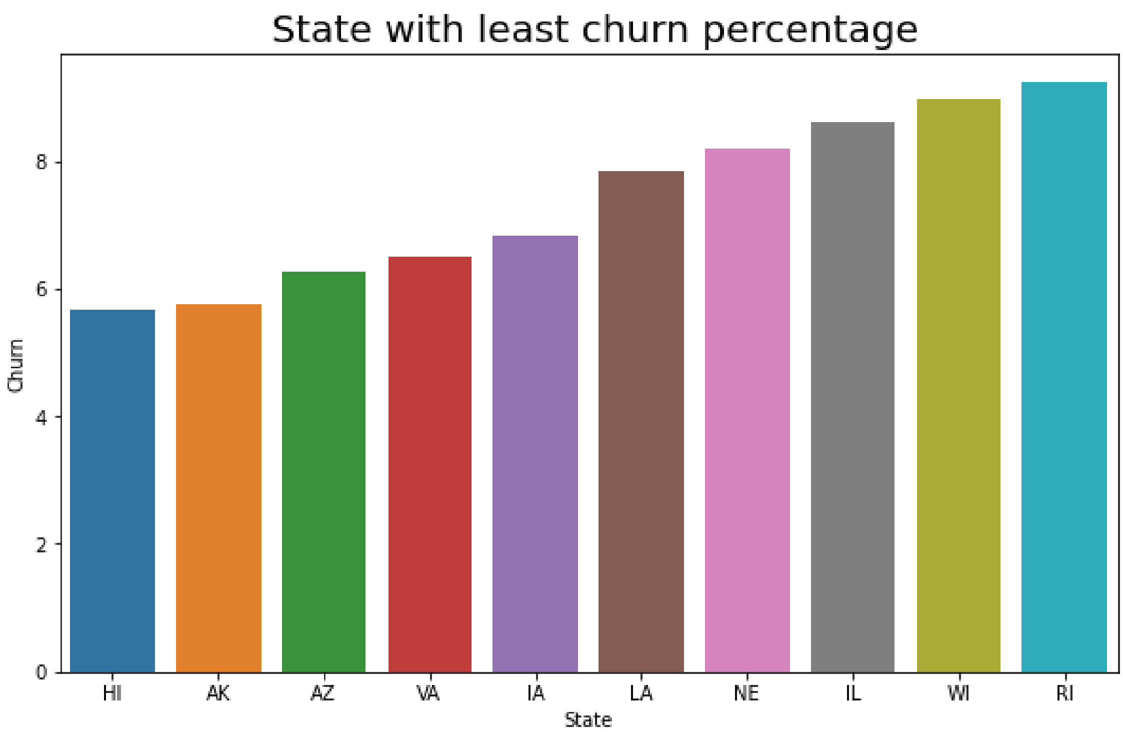
```
bottom_state_churn = ((df.groupby(['State'])['Churn'].mean()*100).sort_values(ascending
bottom_state_churn
```

Out[36]:

	State	Churn
0	HI	5.660377
1	AK	5.769231
2	AZ	6.250000
3	VA	6.493506
4	IA	6.818182
5	LA	7.843137
6	NE	8.196721
7	IL	8.620690
8	WI	8.974359
9	RI	9.230769

In [37]:

```
plt.figure(figsize = (10,6))
sns.barplot(data = bottom_state_churn, x = bottom_state_churn['State'], y = bottom_state
plt.title(" State with least churn percentage", fontsize = 20)
plt.show()
```



Observation

There are 51 states having different churn rates .

CA, NJ ,TX , MD ,SC ,MI, MS, NV, WA, ME are the ones who have higher churn rate more than 20% which is more than 50% of average churn rate.

And **HI, AK, AZ, VA, IA, LA, NE, IL, WI, RI** are the ones who have lower churn rate which is less than 10%.

11. One Digit Account Length

In [38]:

```
df.head()
```

Out[38]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [39]:

```
one_digit = df[df['Account length'] <= 9]
one_digit.head()
```

Out[39]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
158	MO	6	510	No	No	0	183.6	117	31.21	256.7
671	CT	3	415	No	Yes	36	118.1	117	20.08	221.5
923	MS	1	415	No	No	0	144.8	107	24.62	112.5
960	AR	5	415	No	No	0	199.2	106	33.86	187.3
964	NY	9	408	No	Yes	31	193.8	130	32.95	202.6

In [40]:

```
one_digit['Churn'].value_counts()
```

Out[40]:

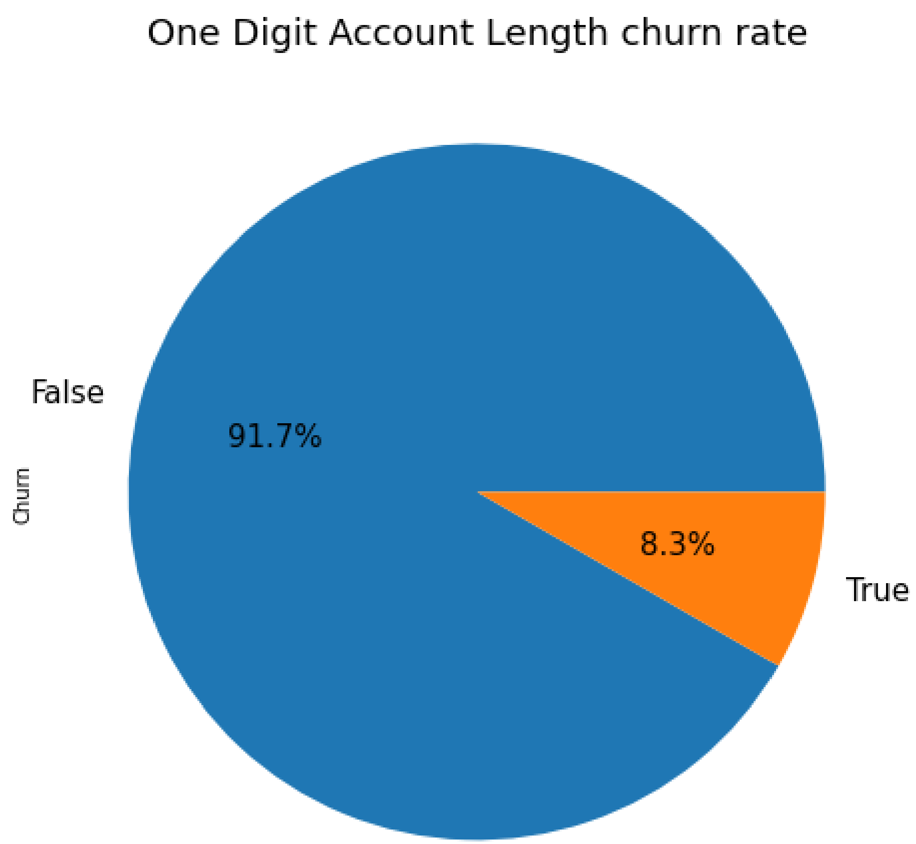
```
False    22
True      2
Name: Churn, dtype: int64
```

In [41]:

```
one_digit['Churn'].value_counts().plot(kind = 'pie', figsize = (8,8), autopct = '%1.1f%%
plt.title('One Digit Account Length churn rate', fontsize=18)
```

Out[41]:

```
Text(0.5, 1.0, 'One Digit Account Length churn rate')
```



Observation

Account length is the noumber of days the customers are active. So for the new customers those churning rate is too low around 8.3% in percentage. They might be just using the telecom service to experience the benefits and they might not be satisfied with the service provided and churned.

12. Two Digit Account Length

In [42]:

```
two_digit = df[(df['Account length'] > 9) & (df['Account length'] <= 99)]
two_digit.head()
```

Out[42]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3
10	IN	65	415	No	No	0	129.1	137	21.95	228.5
11	RI	74	415	No	No	0	187.7	127	31.91	163.4
13	MT	95	510	No	No	0	156.6	88	26.62	247.6

In [43]:

```
two_digit['Churn'].value_counts()
```

Out[43]:

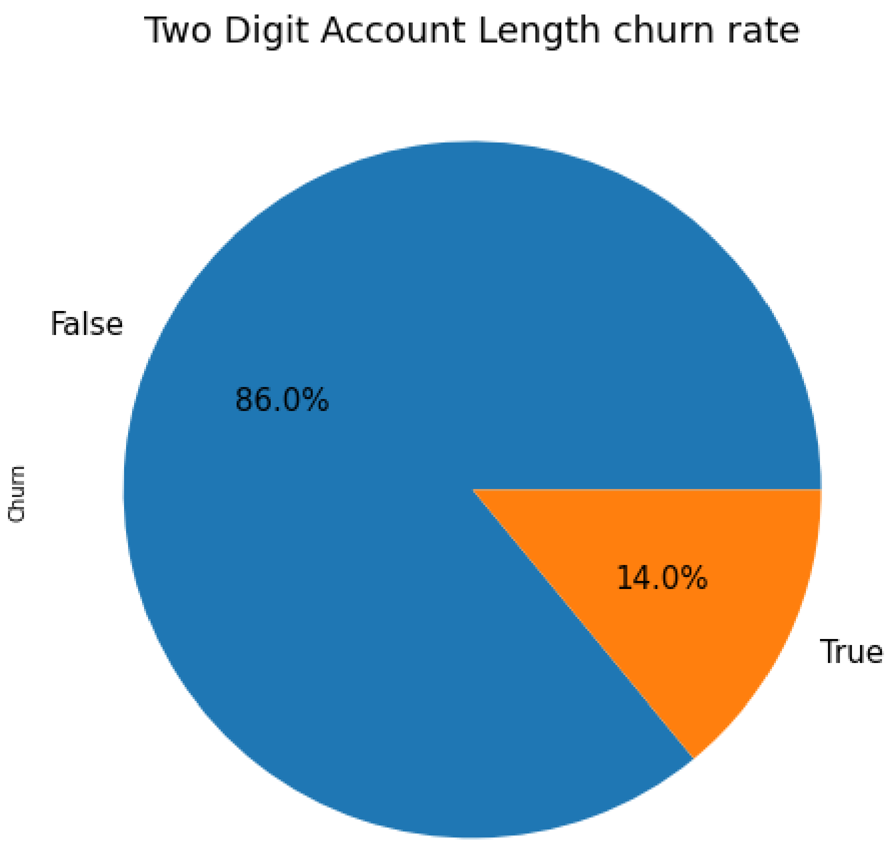
```
False    1378
True      225
Name: Churn, dtype: int64
```

In [44]:

```
two_digit['Churn'].value_counts().plot(kind = 'pie', figsize = (8,8), autopct = '%1.1f%
plt.title('Two Digit Account Length churn rate', fontsize=18)
```

Out[44]:

```
Text(0.5, 1.0, 'Two Digit Account Length churn rate')
```



Observation

Those people whose account length are between 10 to 99 are having a churning rate of 14%. The customers below 50 might be treated as new customers and more than 55 and less than 99 they might not be geting benefits from plan taken.

13. Three Digit Account Length

In [45]:

```
three_digit = df[(df['Account length'] > 99)]
three_digit.head()
```

Out[45]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
5	AL	118	510	Yes	No	0	223.4	98	37.98	220.6
6	MA	121	510	No	Yes	24	218.2	88	37.09	348.5

In [46]:

```
three_digit['Churn'].value_counts()
```

Out[46]:

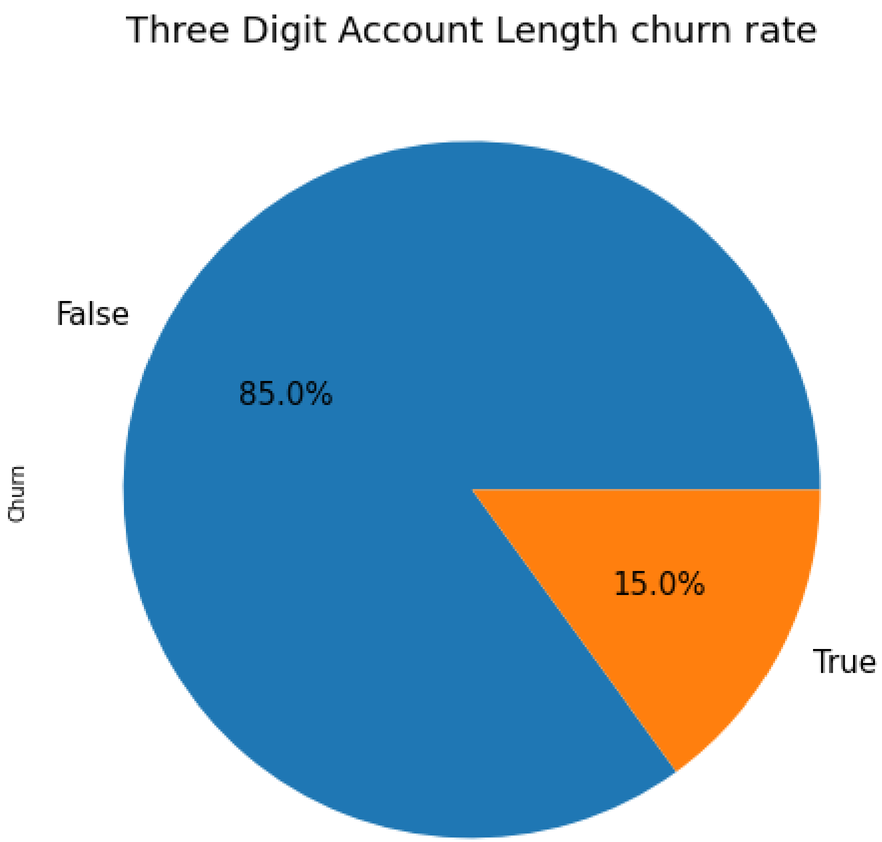
```
False    1450
True      256
Name: Churn, dtype: int64
```

In [47]:

```
three_digit['Churn'].value_counts().plot(kind = 'pie', figsize = (8,8), autopct = '%1.1f
plt.title('Three Digit Account Length churn rate', fontsize=18)
```

Out[47]:

```
Text(0.5, 1.0, 'Three Digit Account Length churn rate')
```



Observation

Those people whose account length is more than 100 are like old customers and they might be churning due to no additional offers given to them like power plus plan or other benefits.

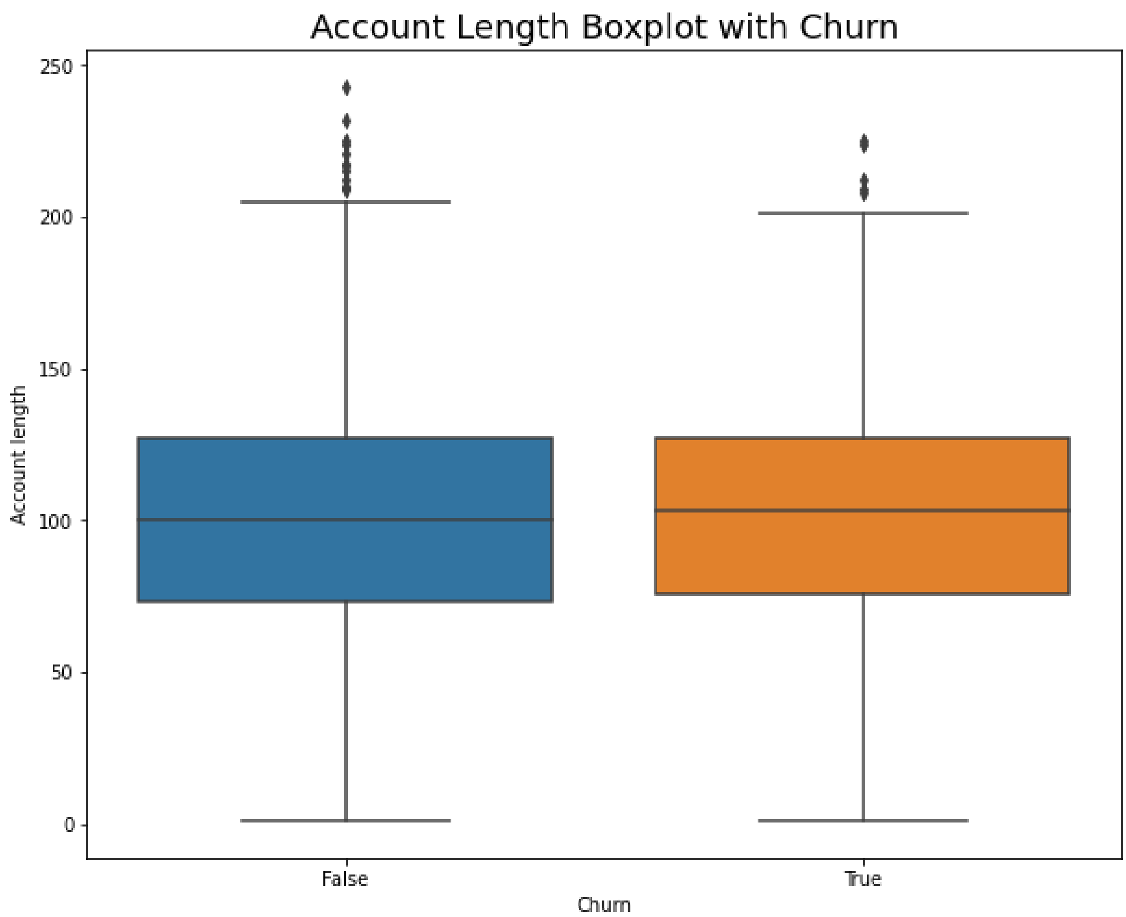
14. Box Plot for Account Length attribute

In [48]:

```
plt.figure(figsize=(10,8))
sns.boxplot(data = df, x ='Churn', y = 'Account length')
plt.title('Account Length Boxplot with Churn', fontsize=18)
```

Out[48]:

```
Text(0.5, 1.0, 'Account Length Boxplot with Churn')
```



Observation

Account length is the no. of days the customers are active. So for the new customers those churning rate is too low around 8.3% in percentage and number is 2. They might be just using the telecom service to experience the benefits and they might not be satisfied with the service provided and churned.

Those people whose account length are between 10 to 99 are having a churning rate of 14%. The customers below 50 might be treated as new customers and more than 55 and less than 99 they might not be geting benefits from plan taken.

Those people whose account length is more than 100 are like old customers and they might be churning due to no additional offers given to them like power plus plan or other benefits.

So, yes Account Length is also depicting a clear view of churing reasons and insights.

15. Voice Mail

In [49]:

```
df.head()
```

Out[49]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [50]:

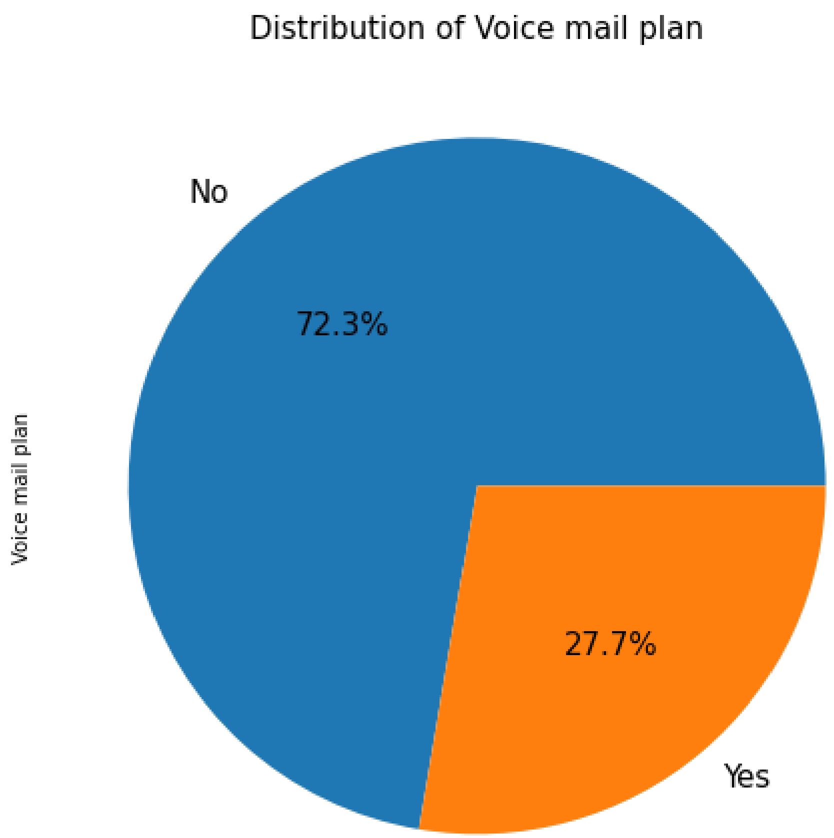
```
voice_mail_plan = df['Voice mail plan'].value_counts()
voice_mail_plan
```

Out[50]:

```
No     2411
Yes     922
Name: Voice mail plan, dtype: int64
```

In [51]:

```
voice_mail_plan.plot(kind = 'pie', autopct = '%1.1f%%', figsize = (8,8), fontsize = 15)
plt.title('Distribution of Voice mail plan', fontsize = 15)
plt.show()
```



16. Voice mail plan with respect to churn

In [52]:

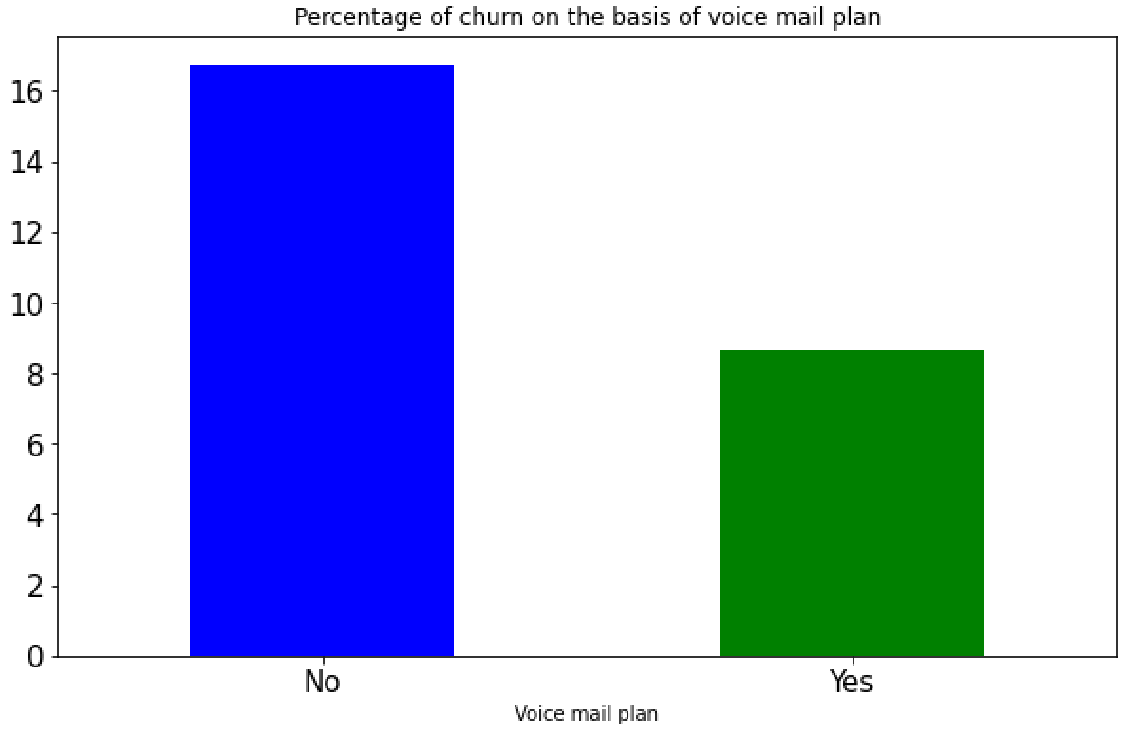
```
voice_churn = ((df.groupby(['Voice mail plan'])['Churn'].mean())*100)
voice_churn
```

Out[52]:

```
Voice mail plan
No     16.715056
Yes     8.676790
Name: Churn, dtype: float64
```

In [53]:

```
voice_churn.plot(kind = 'bar', figsize = (10,6), fontsize = 15, color = ['b','g'])
plt.xticks(rotation = 360)
plt.title("Percentage of churn on the basis of voice mail plan")
plt.show()
```



Observation

2411 dont have a voice mail plan.
922 have a voice mail plan.
Among those who dont have a voice mail 16.7 % people churn.
Whereas among those who have a voice mail plan only 8.7 % people churn.

Customers with the Voice Mail Plan tend to churn less frequently

17. Area code with respect to churn

In [54]:

```
df.head()
```

Out[54]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [55]:

```
df['Area code'].value_counts()
```

Out[55]:

```
415    1655
510     840
408     838
Name: Area code, dtype: int64
```

In [56]:

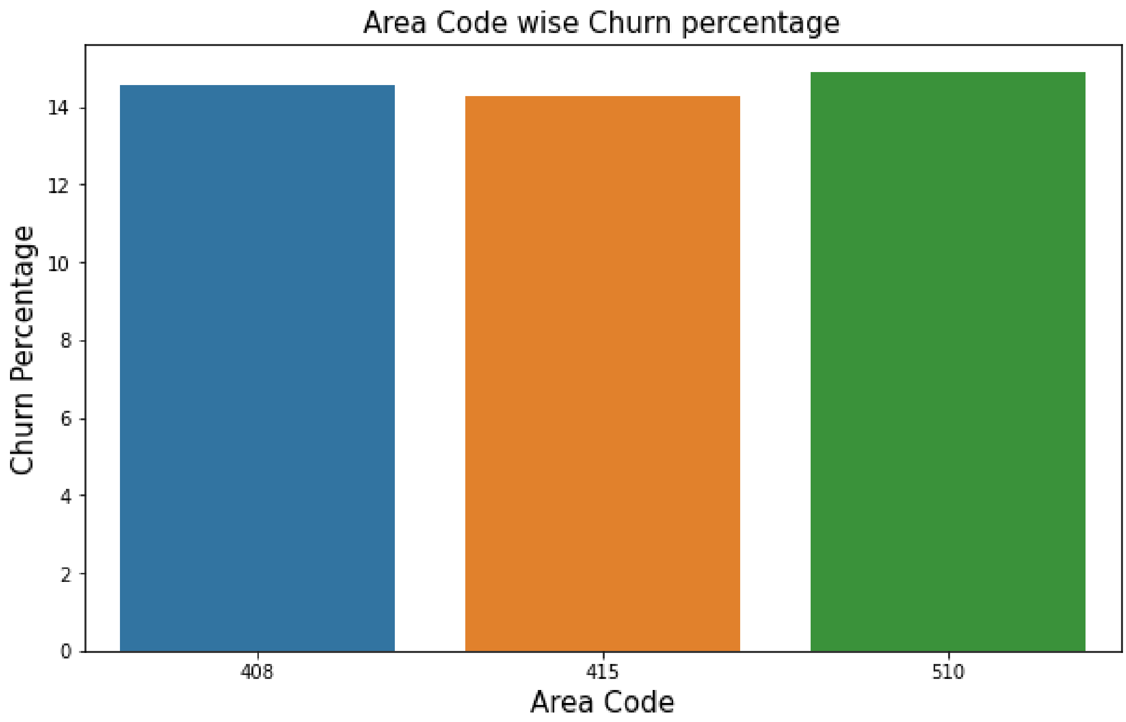
```
area_churn = ((df.groupby(['Area code'])['Churn'].mean())*100).reset_index()
area_churn
```

Out[56]:

	Area code	Churn
0	408	14.558473
1	415	14.259819
2	510	14.880952

In [57]:

```
plt.figure(figsize = (10,6))
sns.barplot(data = area_churn, x = 'Area code', y = 'Churn')
plt.title('Area Code wise Churn percentage', fontsize = 15)
plt.ylabel('Churn Percentage', fontsize = 15)
plt.xlabel("Area Code", fontsize = 15)
plt.show()
```



Observation

All Area Code have around 14% Churn rate. So, Area Code doesn't matter.

18. Overall Calls in day

In [58]:

```
df.head()
```

Out[58]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [59]:

```
overall_calls_day = df.groupby(['Churn'])['Total day minutes', 'Total day calls', 'Total
overall_calls_day
```

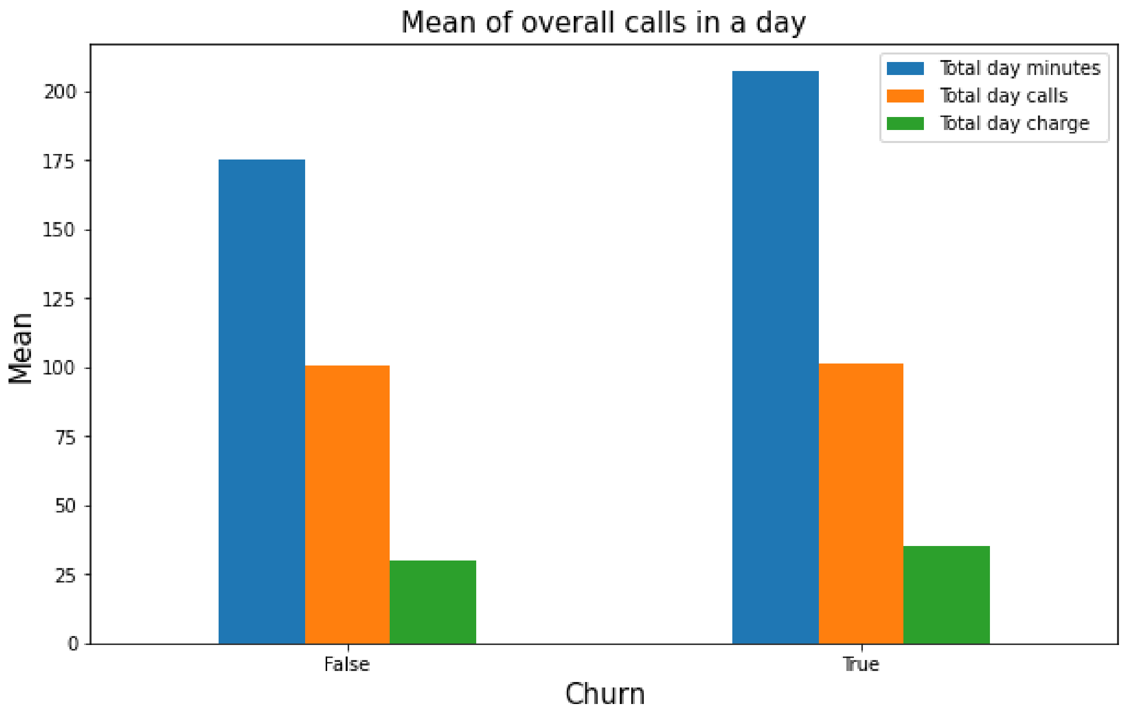
Out[59]:

	Total day minutes	Total day calls	Total day charge
Churn			
False	175.175754	100.283158	29.780421
True	206.914079	101.335404	35.175921

In [60]:

```
plt.figure(figsize = (10,6))
overall_calls_day.plot(kind = 'bar', figsize = (10,6))
plt.title('Mean of overall calls in a day', fontsize = 15)
plt.xticks(rotation = 360)
plt.ylabel('Mean', fontsize = 15)
plt.xlabel("Churn", fontsize = 15)
plt.show()
```

<Figure size 720x432 with 0 Axes>



19. Overall Calls in evening

In [61]:

```
df.head()
```

Out[61]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [62]:

```
overall_calls_eve = df.groupby(['Churn'])['Total eve minutes', 'Total eve calls', 'Total
overall_calls_eve
```

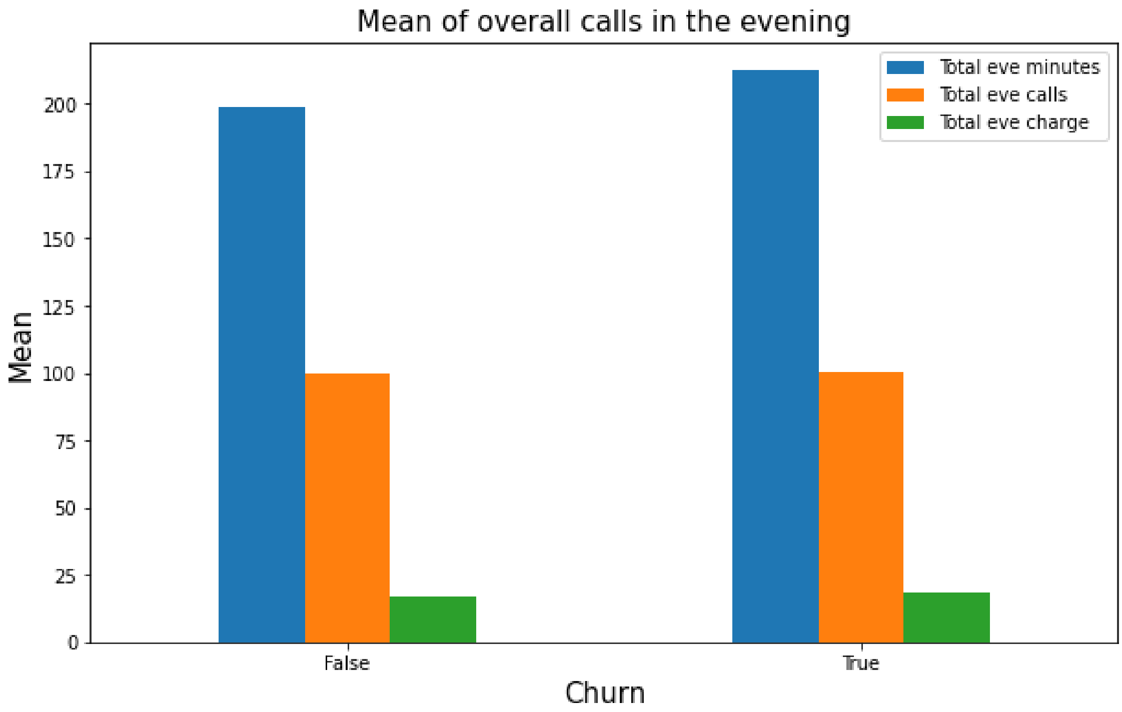
Out[62]:

Churn	Total eve minutes	Total eve calls	Total eve charge
False	199.043298	100.038596	16.918909
True	212.410145	100.561077	18.054969

In [63]:

```
plt.figure(figsize = (10,6))
overall_calls_eve.plot(kind = 'bar', figsize = (10,6))
plt.title('Mean of overall calls in the evening', fontsize = 15)
plt.xticks(rotation = 360)
plt.ylabel('Mean', fontsize = 15)
plt.xlabel("Churn", fontsize = 15)
plt.show()
```

<Figure size 720x432 with 0 Axes>



20. Overall Calls in night

In [64]:

```
df.head()
```

Out[64]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [65]:

```
overall_calls_night = df.groupby(['Churn'])['Total night minutes', 'Total night calls',
overall_calls_night
```

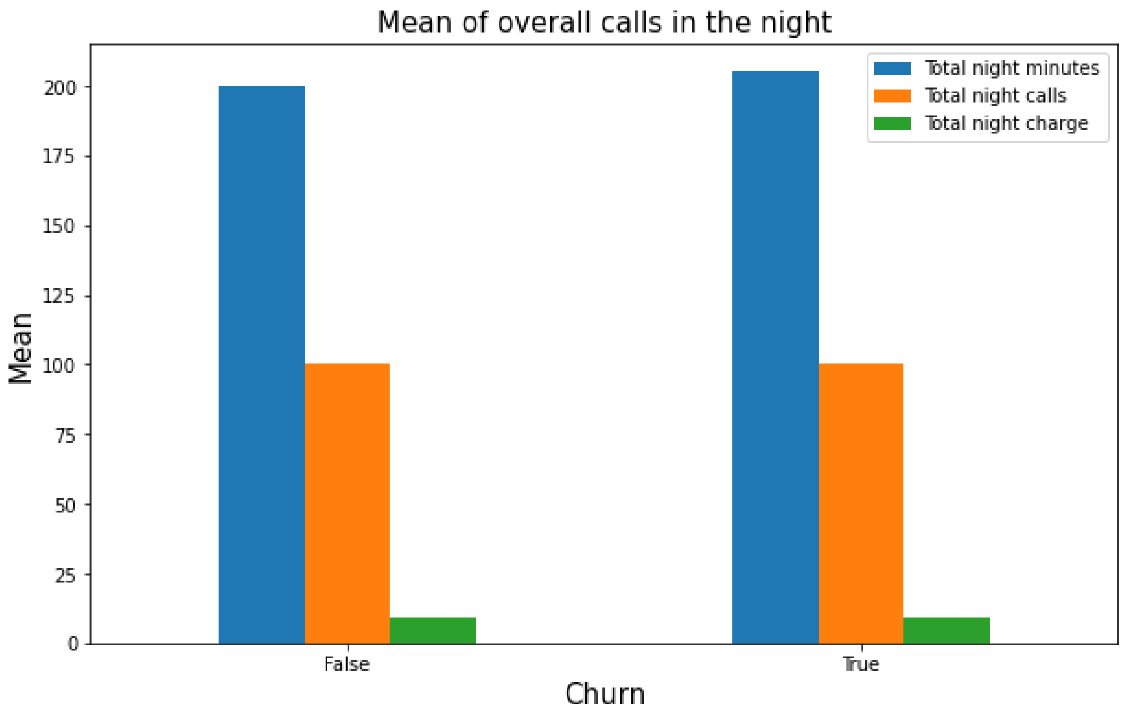
Out[65]:

Churn	Total night minutes	Total night calls	Total night charge
False	200.133193	100.058246	9.006074
True	205.231677	100.399586	9.235528

In [66]:

```
plt.figure(figsize = (10,6))
overall_calls_night.plot(kind = 'bar', figsize = (10,6))
plt.title('Mean of overall calls in the night', fontsize = 15)
plt.xticks(rotation = 360)
plt.ylabel('Mean', fontsize = 15)
plt.xlabel("Churn", fontsize = 15)
plt.show()
```

<Figure size 720x432 with 0 Axes>



21. Average calls of total day calls, evening calls & night calls on basis of churn

In [67]:

```
df.head()
```

Out[67]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [68]:

```
avg_calls = df.groupby(['Churn'])['Total day calls', 'Total eve calls', 'Total night cal
avg_calls
```

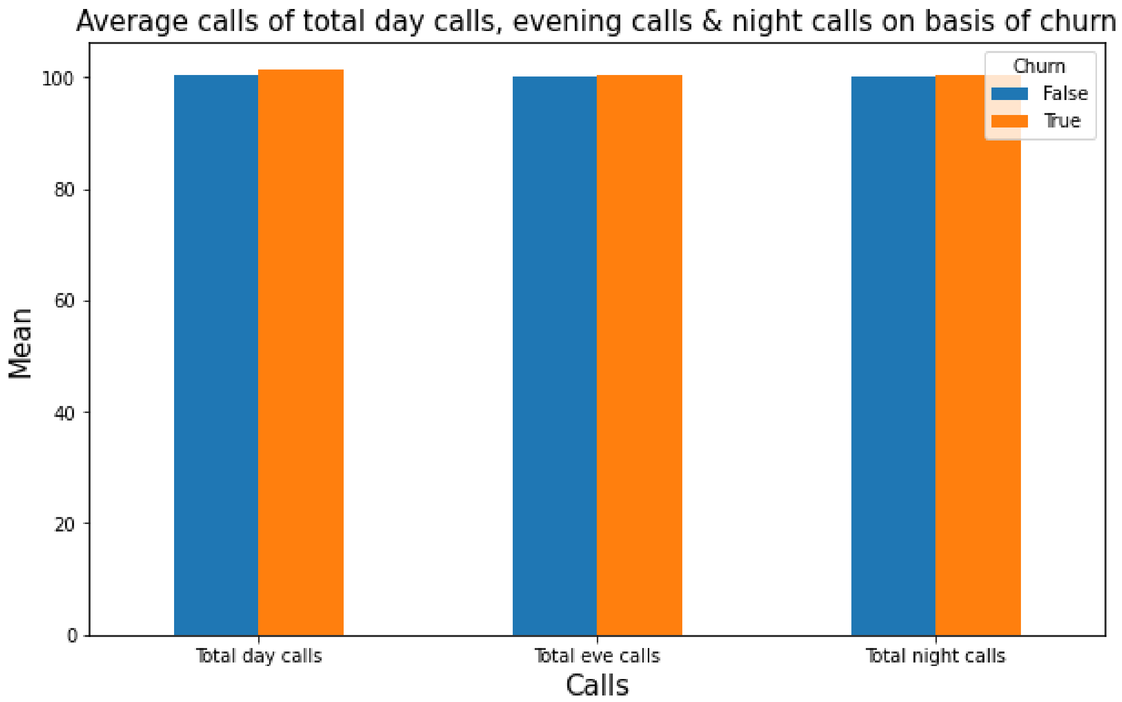
Out[68]:

Churn	False	True
Total day calls	100.283158	101.335404
Total eve calls	100.038596	100.561077
Total night calls	100.058246	100.399586

In [69]:

```
plt.figure(figsize = (10,6))
avg_calls.plot(kind = 'bar', figsize = (10,6))
plt.title('Average calls of total day calls, evening calls & night calls on basis of chu
plt.xticks(rotation = 360)
plt.ylabel('Mean', fontsize = 15)
plt.xlabel("Calls", fontsize = 15)
plt.show()
```

<Figure size 720x432 with 0 Axes>



22. Average calls of total day minutes, evening minutes & night minutes on basis of churn

In [70]:

```
df.head()
```

Out[70]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [71]:

```
avg_minutes = df.groupby(['Churn'])['Total day minutes', 'Total eve minutes', 'Total nig
avg_minutes
```

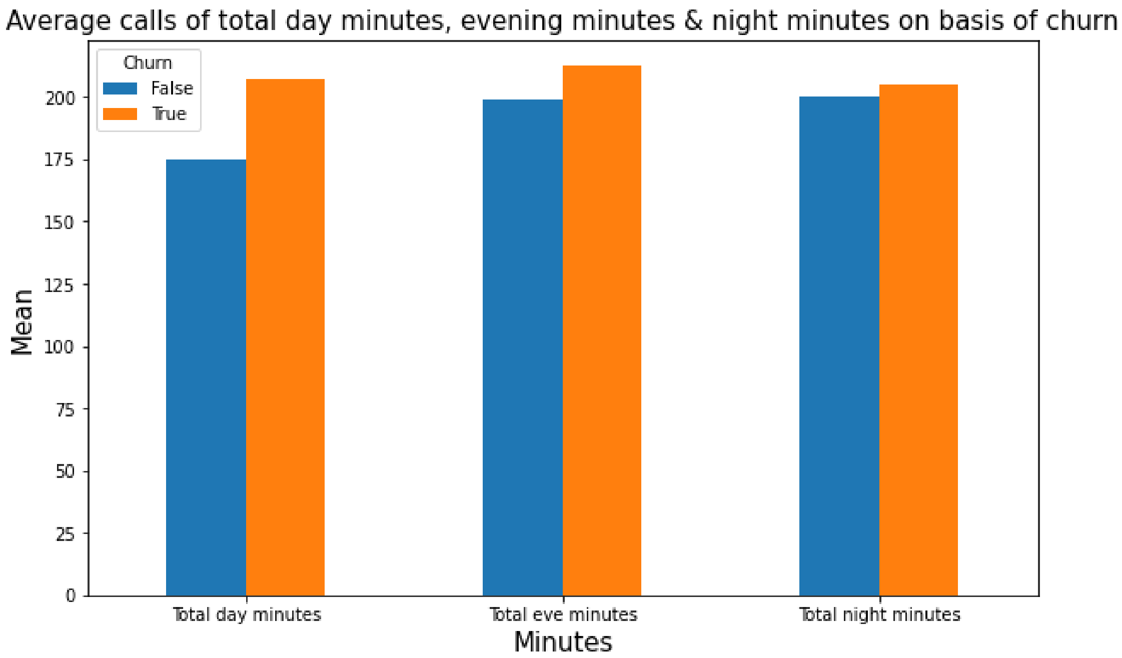
Out[71]:

Churn	False	True
Total day minutes	175.175754	206.914079
Total eve minutes	199.043298	212.410145
Total night minutes	200.133193	205.231677

In [72]:

```
plt.figure(figsize = (10,6))
avg_minutes.plot(kind = 'bar', figsize = (10,6))
plt.title('Average calls of total day minutes, evening minutes & night minutes on basis
plt.xticks(rotation = 360)
plt.ylabel('Mean', fontsize = 15)
plt.xlabel("Minutes", fontsize = 15)
plt.show()
```

```
<Figure size 720x432 with 0 Axes>
```



23. Average calls of total day Charges, evening Charges & night Charges on basis of churn

In [73]:

```
df.head()
```

Out[73]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [74]:

```
avg_charges = df.groupby(['Churn'])['Total day charge', 'Total eve charge', 'Total night
avg_charges
```

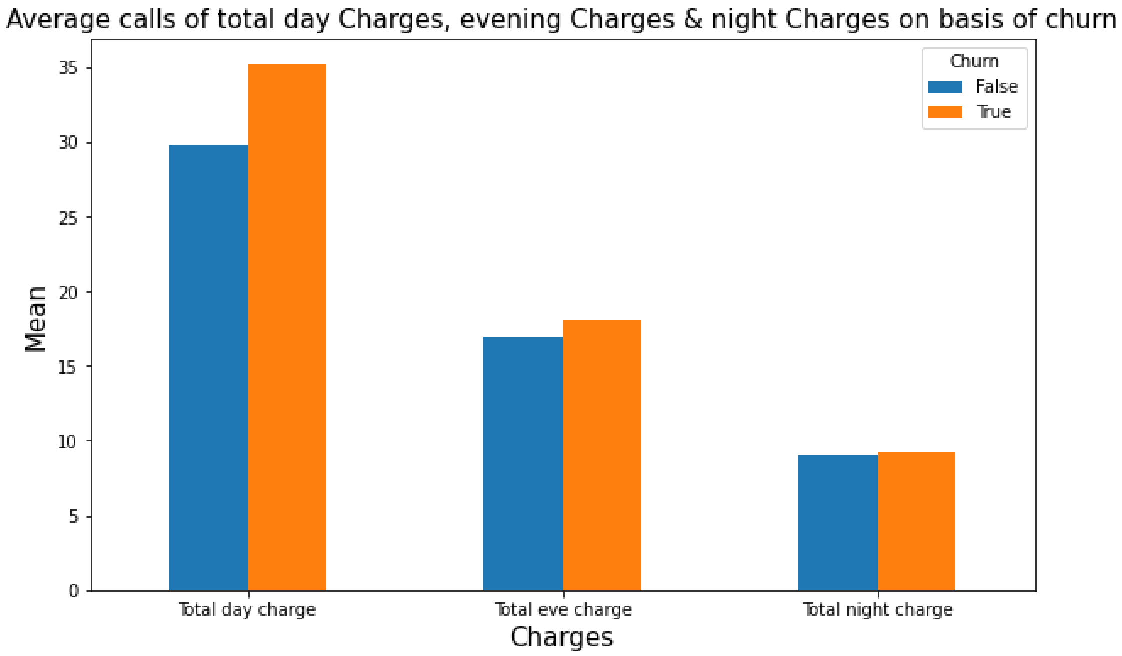
Out[74]:

Churn	False	True
Total day charge	29.780421	35.175921
Total eve charge	16.918909	18.054969
Total night charge	9.006074	9.235528

In [75]:

```
plt.figure(figsize = (10,6))
avg_charges.plot(kind = 'bar', figsize = (10,6))
plt.title('Average calls of total day Charges, evening Charges & night Charges on basis
plt.xticks(rotation = 360)
plt.ylabel('Mean', fontsize = 15)
plt.xlabel("Charges", fontsize = 15)
plt.show()
```

<Figure size 720x432 with 0 Axes>



Observation

Churn customers speak more minutes that non-churn customers at day,evening and night. Hence they pay more charge that non-churn customers.

We can retain churn customers if we include master plan. In master plan if a customer is talking more minutes then we can charge a little less amount from him or he can get discount or additional few free minutes to talk.

This will make customers who are going to churn happy and they will not leave the company.

24. Customer Service calls

In [76]:

```
df.head()
```

Out[76]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3

In [77]:

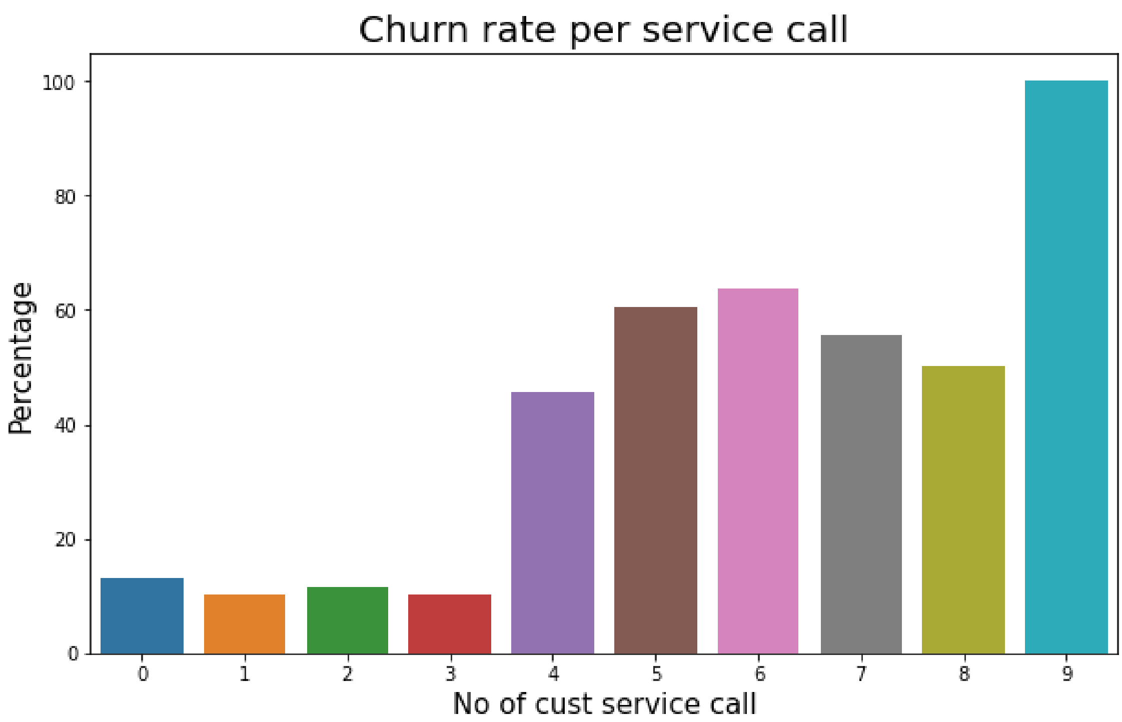
```
customer_churn = ((df.groupby(['Customer service calls'])['Churn'].mean())*100).reset_ir
customer_churn
```

Out[77]:

	Customer service calls	Churn
0	0	13.199426
1	1	10.330229
2	2	11.462451
3	3	10.256410
4	4	45.783133
5	5	60.606061
6	6	63.636364
7	7	55.555556
8	8	50.000000
9	9	100.000000

In [78]:

```
plt.figure(figsize = (10,6))
sns.barplot(data = customer_churn, x = customer_churn['Customer service calls'], y = cus
plt.title("Churn rate per service call", fontsize = 20)
plt.xlabel('No of cust service call', fontsize = 15)
plt.ylabel('Percentage', fontsize = 15)
plt.show()
```



Observation

The service calls of customers varies from 0 to 9 .

Those customers who make more service calls they have a high probability of leaving.

As we can see from graph , customers with more than 5, their churning rate is more.

Hence customers who make more than 5 service calls, their queries should be solved immediately and they should be given better service so that they dont leave the company.

Customers with four or more customer service calls churn more than four times as often as do the other customers

Conclusion

The telecommunications market is already well-established, and the rate of new customers is slow. As a result, companies in this industry prioritize retention and reducing customer churn. This project analyzed a churn dataset to identify the main factors contributing to churn and gain valuable insights. Through exploratory data analysis, we were able to gain insight into the churn dataset, listed below:

1. The four charge fields are directly related to the minute fields.
2. The area code may not be relevant and can be excluded.
3. Customers with the International Plan tend to churn more often.
4. Customers who have had four or more customer service calls churn significantly more than other customers.
5. Customers with high day and evening minute usage tend to churn at a higher rate.
6. There is no clear relationship between churn and the variables such as day calls, evening calls, night calls, international calls, night minutes, international minutes, account length, or voice mail messages.