Telecom Customer Churn Analysis

```
1. Domain Exploration
    - Understand the buiness process
    - Identify common challenges, business beleifs, losses, solutions, data flow

2. Data Collection and Data Exploration
    - Collect data from multiple verticals in business, prepare a dataset
    - perform general data exploration to understand data quality, identify common issues

3. Data Cleaning
    - Handle missing values
    - Handle duplicate entries
    - Handle unwanted columns - identifiers
    - Handle outliers

4. Analysis on data
    - Descriptive Analysis - analyse each variable individually
    - Exploratory Analysis - Analyse each variable with respect to the target KPI (churn)
        - Statistics methods
        - Data visualization

5. Prepare reports to communicate the results
```

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Data Exploration

In [10]:

```
# Load data
df = pd.read_excel(r"E:\MLIoT\ML\dataset\telecom\telecom_churn_modelling.xlsx",
                   na_values=['#','NA','--','Not Available','?'])
df.shape
```

Out[10]:

```
(3333, 20)
```

In [4]:

```
df.head()
```

Out[4]:

	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	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10.0	3
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13.7	3
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	110	10.30	162.6	104	7.32	12.2	5
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9	88	5.26	196.9	89	8.86	6.6	7
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3	122	12.61	186.9	121	8.41	10.1	3

In [5]:

```
len(df.State.unique())
```

Out[5]:

```
51
```

Observations -

```
- State is a categorical attribute with 51 unique value - high cardinality
- Voice mail plan and Number vmail messages represent relative information
- Total xxx charge should be related/multiplier of total xxx minutes or total xxx calls
```

In [6]:

```
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
```

In [7]:

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

Out[7]:

```
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 [8]:

```
df['International plan'].unique()
```

Out[8]:

```
array(['No', 'Yes'], dtype=object)
```

In [9]:

```
df['Voice mail plan'].unique()
```

Out[9]:

```
array(['Yes', 'No'], dtype=object)
```

Data Cleaning

In [11]:

```
# check for duplicate entries
df.duplicated().sum()
```

Out[11]:

```
0
```

In [12]:

```
#check for missing values
df.isnull().sum()
```

Out[12]:

```
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
```

In [13]:

```
# check for unwanted columns - identifiers
# need to analyse - State
```

In [14]:
```
# check for outliers
df.skew()
```

Out[14]:
```
Account length            0.096606
Area code                 1.126823
Number vmail messages     1.264824
Total day minutes        -0.029077
Total day calls          -0.111787
Total day charge         -0.029083
Total eve minutes        -0.023877
Total eve calls          -0.055563
Total eve charge         -0.023858
Total night minutes       0.008921
Total night calls         0.032500
Total night charge        0.008886
Total intl minutes       -0.245136
Total intl calls          1.321478
Total intl charge        -0.245287
Customer service calls    1.091359
Churn                     2.018356
dtype: float64
```

Descriptive Analysis

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	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10.0	3
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13.7	3
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	110	10.30	162.6	104	7.32	12.2	5
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9	88	5.26	196.9	89	8.86	6.6	7
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3	122	12.61	186.9	121	8.41	10.1	3

In [19]:
```
df.describe()
```

Out[19]:

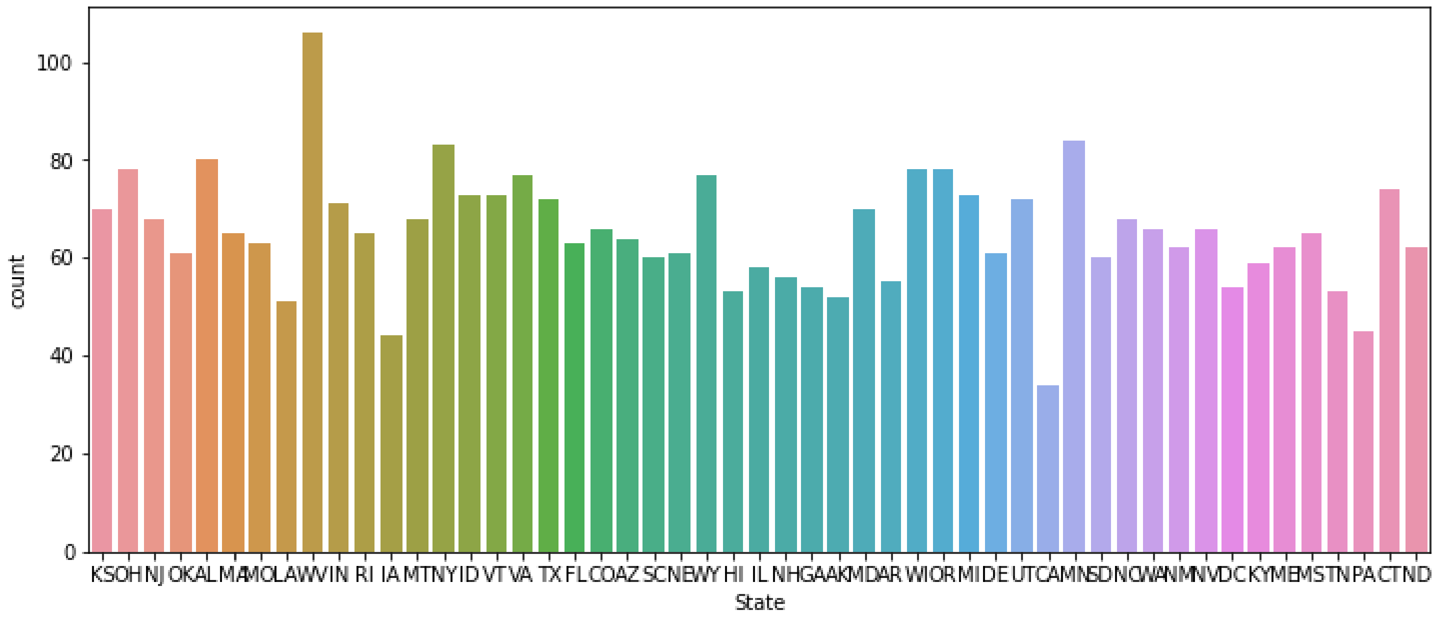
	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Tota
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.
mean	101.064806	437.182418	8.099010	179.775098	100.435644	30.562307	200.980348	100.114311	17.083540	200.872037	100
std	39.822106	42.371290	13.688365	54.467389	20.069084	9.259435	50.713844	19.922625	4.310668	50.573847	19.
min	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	23.200000	33.
25%	74.000000	408.000000	0.000000	143.700000	87.000000	24.430000	166.600000	87.000000	14.160000	167.000000	87.
50%	101.000000	415.000000	0.000000	179.400000	101.000000	30.500000	201.400000	100.000000	17.120000	201.200000	100.
75%	127.000000	510.000000	20.000000	216.400000	114.000000	36.790000	235.300000	114.000000	20.000000	235.300000	113.
max	243.000000	510.000000	51.000000	350.800000	165.000000	59.640000	363.700000	170.000000	30.910000	395.000000	175.

State

In [16]:
```
df['State'].value_counts()
```

Out[16]:
```
WV    106
MN     84
NY     83
AL     80
OR     78
OH     78
WI     78
VA     77
WY     77
CT     74
VT     73
ID     73
MI     73
TX     72
UT     72
IN     71
MD     70
KS     70
MT     68
NJ     68
NC     68
NV     66
WA     66
CO     66
RI     65
MS     65
MA     65
AZ     64
FL     63
MO     63
NM     62
ME     62
ND     62
OK     61
DE     61
NE     61
SC     60
SD     60
KY     59
IL     58
NH     56
AR     55
GA     54
DC     54
TN     53
HI     53
AK     52
LA     51
PA     45
IA     44
CA     34
Name: State, dtype: int64
```

In [17]:
```
plt.figure(figsize=(12,5))
sns.countplot(df['State'])
plt.show()
```



Observation -

```
- On a whole, states have some variation present in the number of customers from each state
```

Analysing numeric attributes

In [20]:
```
df.columns
```

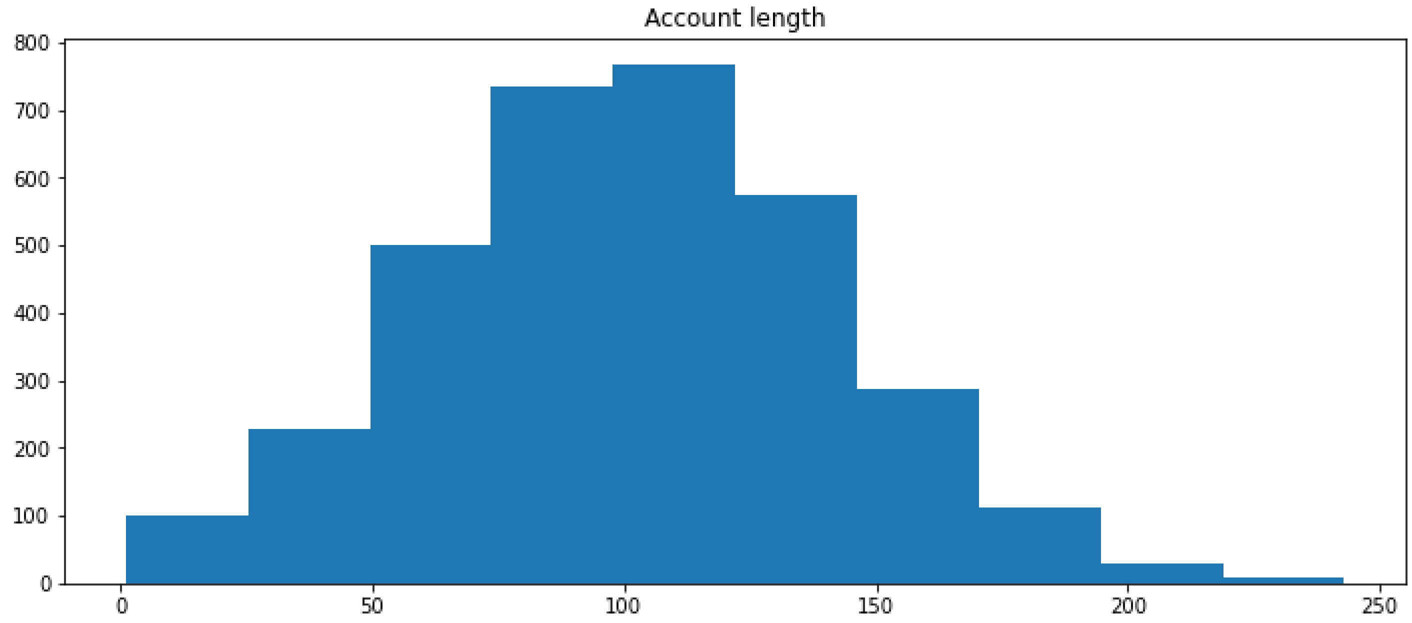
Out[20]:
```
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day minutes',
       'Total day calls', 'Total day charge', 'Total eve minutes',
       'Total eve calls', 'Total eve charge', 'Total night minutes',
       'Total night calls', 'Total night charge', 'Total intl minutes',
       'Total intl calls', 'Total intl charge', 'Customer service calls',
       'Churn'],
      dtype='object')
```

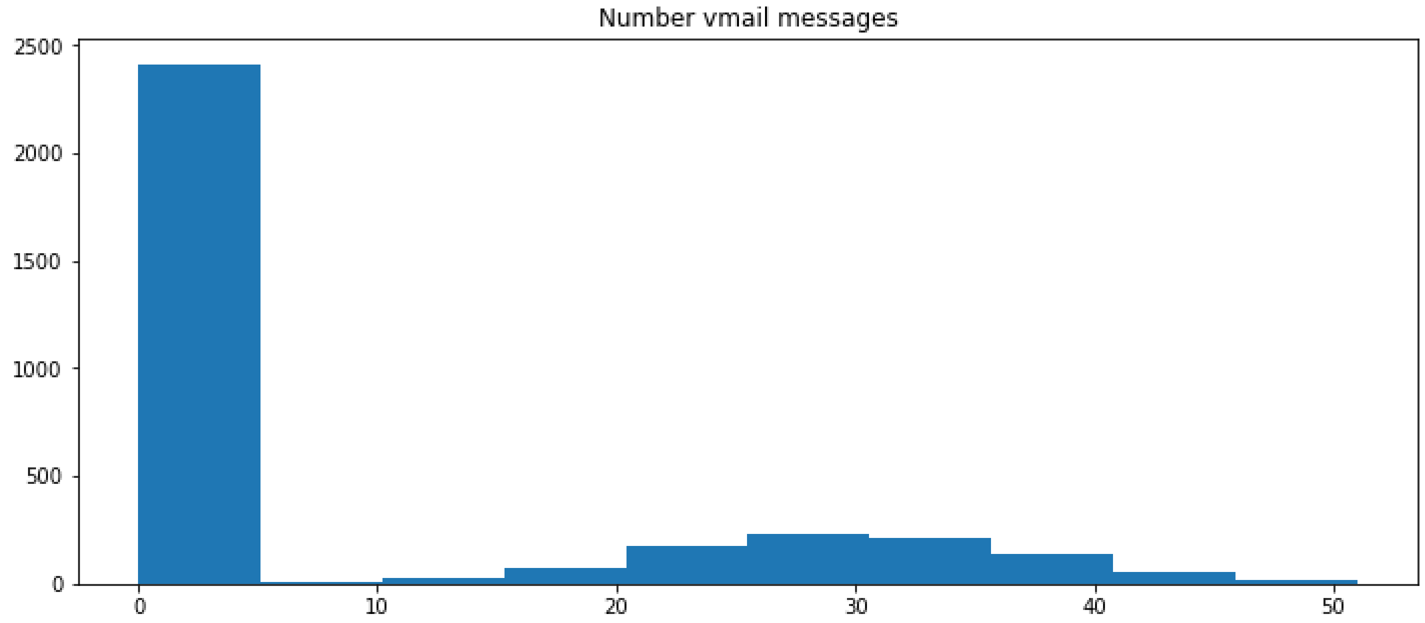
In [21]:
```
df.describe()
```

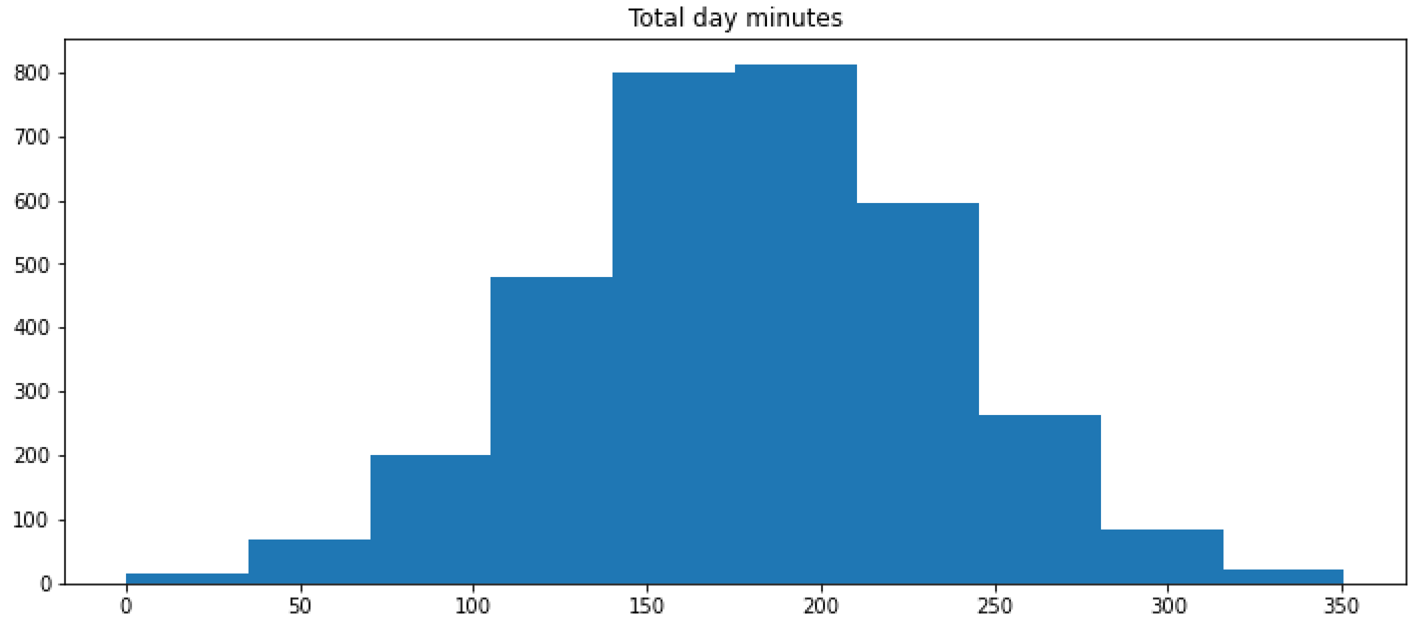
Out[21]:

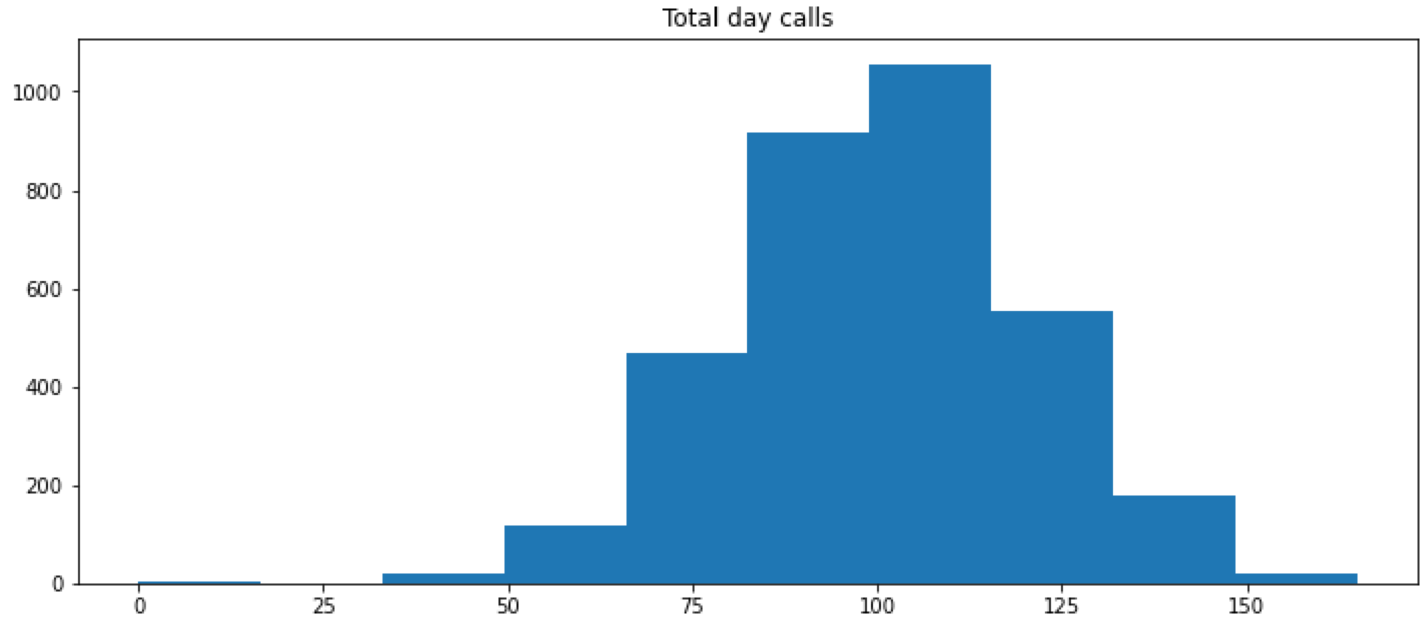
	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Tot
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.
mean	101.064806	437.182418	8.099010	179.775098	100.435644	30.562307	200.980348	100.114311	17.083540	200.872037	100
std	39.822106	42.371290	13.688365	54.467389	20.069084	9.259435	50.713844	19.922625	4.310668	50.573847	19.
min	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	23.200000	33.
25%	74.000000	408.000000	0.000000	143.700000	87.000000	24.430000	166.600000	87.000000	14.160000	167.000000	87.
50%	101.000000	415.000000	0.000000	179.400000	101.000000	30.500000	201.400000	100.000000	17.120000	201.200000	100.
75%	127.000000	510.000000	20.000000	216.400000	114.000000	36.790000	235.300000	114.000000	20.000000	235.300000	113.
max	243.000000	510.000000	51.000000	350.800000	165.000000	59.640000	363.700000	170.000000	30.910000	395.000000	175.

```
In [22]: nums = ['Account length', 'Number vmail 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', 'Total intl minutes',
                'Total intl calls', 'Total intl charge', 'Customer service calls']

         for col in nums:
             plt.figure(figsize=(12,5))
             plt.hist(df[col])
             plt.title(col)
             plt.show()
```

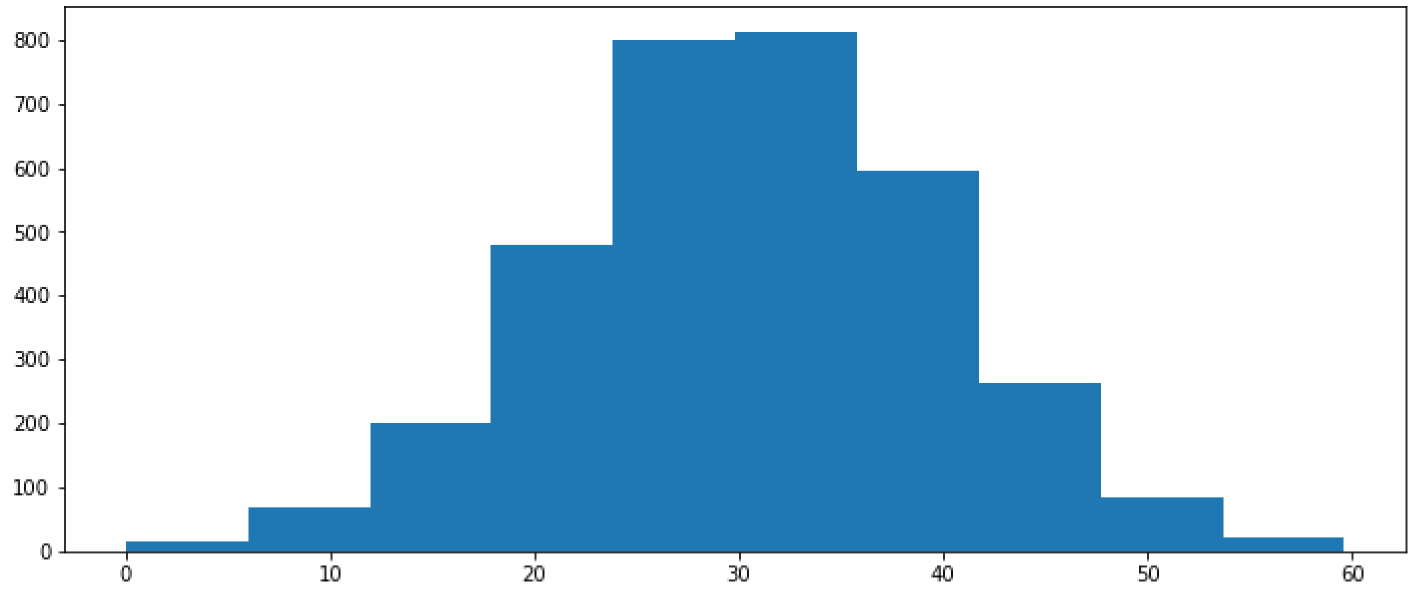

Account length
0
100
200
300
400
500
600
700
800
0
50
100
150
200
250


Number vmail messages
0
500
1000
1500
2000
2500
0
10
20
30
40
50

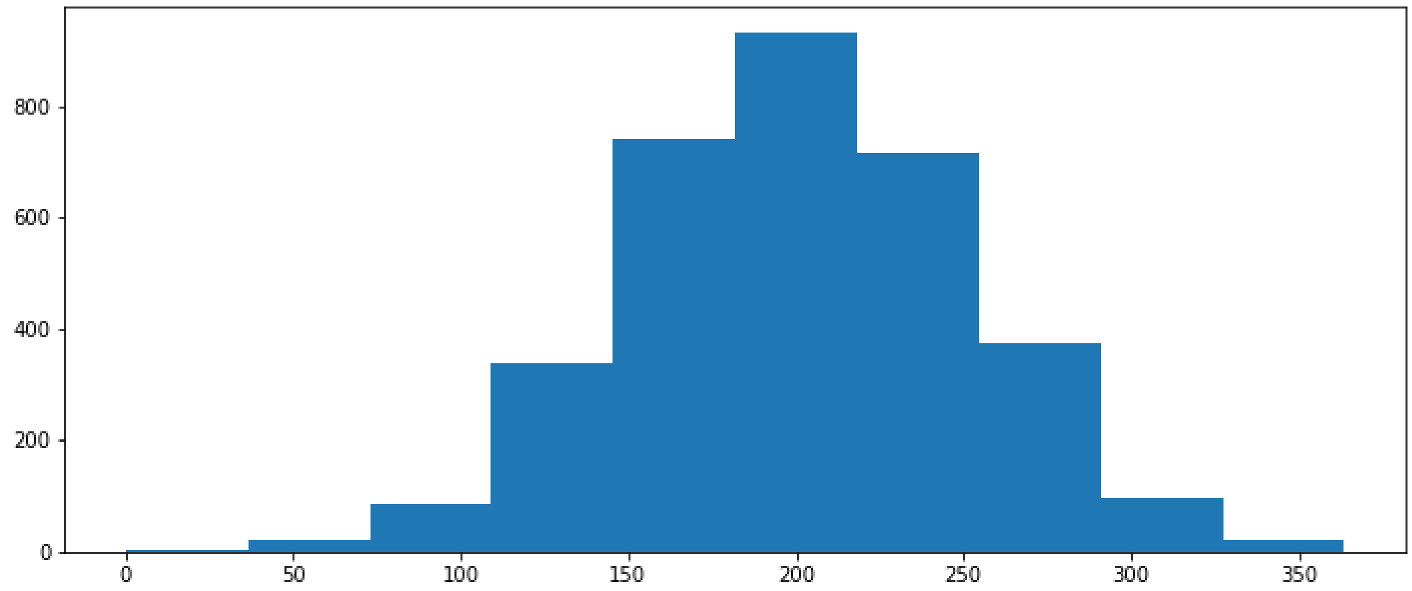

Total day minutes
0
100
200
300
400
500
600
700
800
0
50
100
150
200
250
300
350


Total day calls
0
200
400
600
800
1000
0
25
50
75
100
125
150

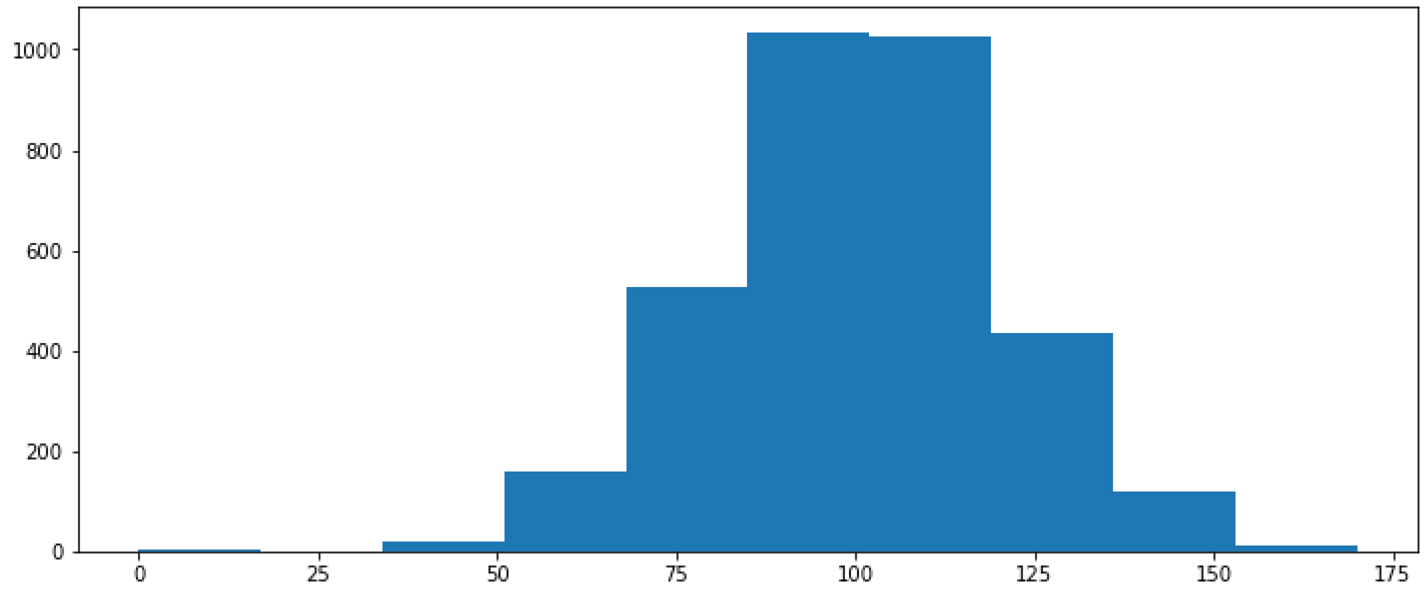
Total day charge



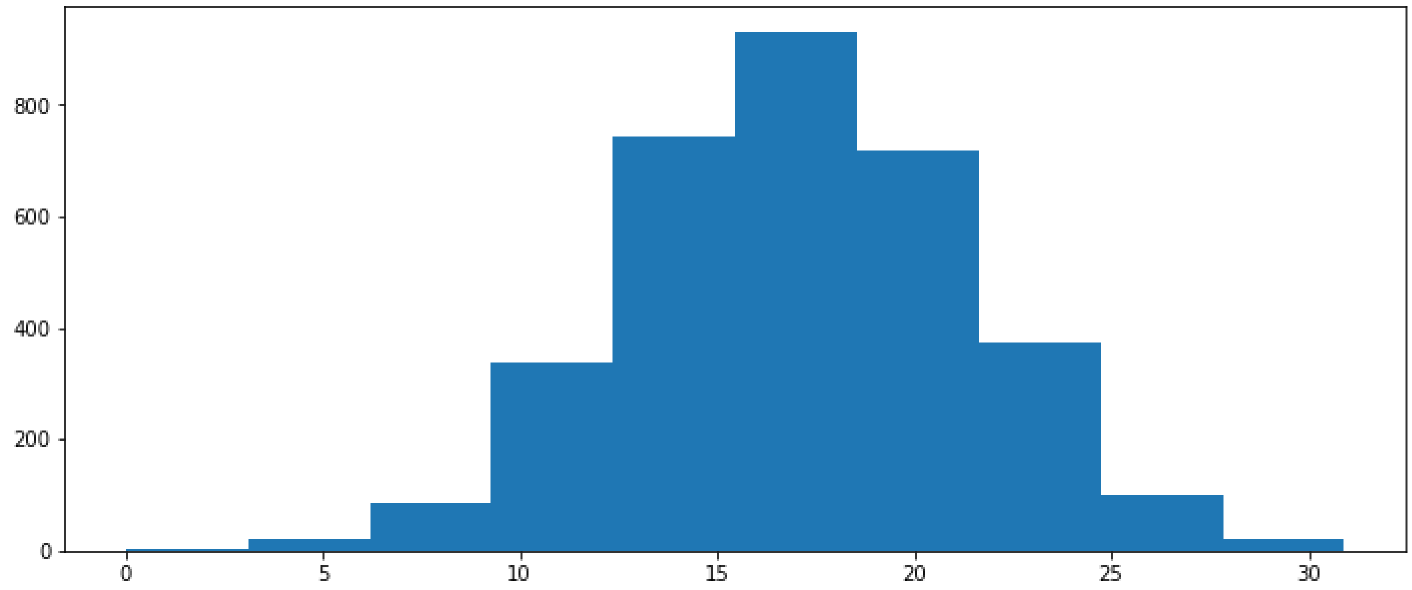
Total eve minutes

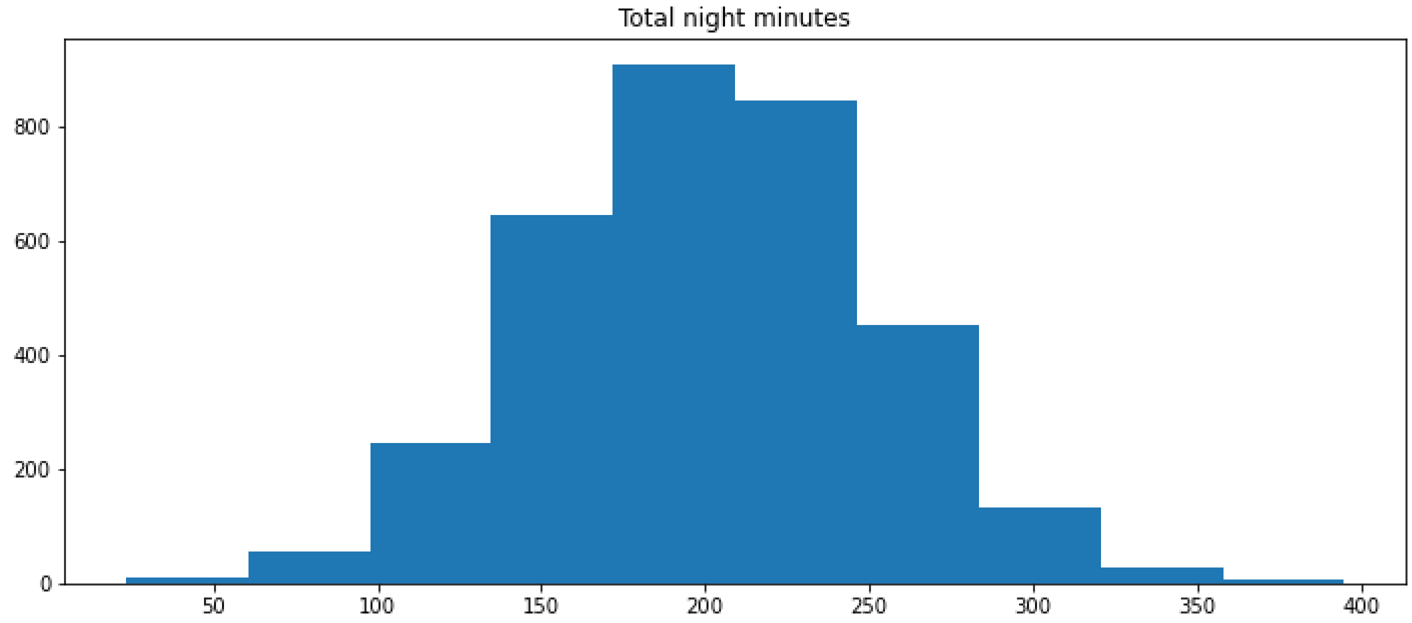


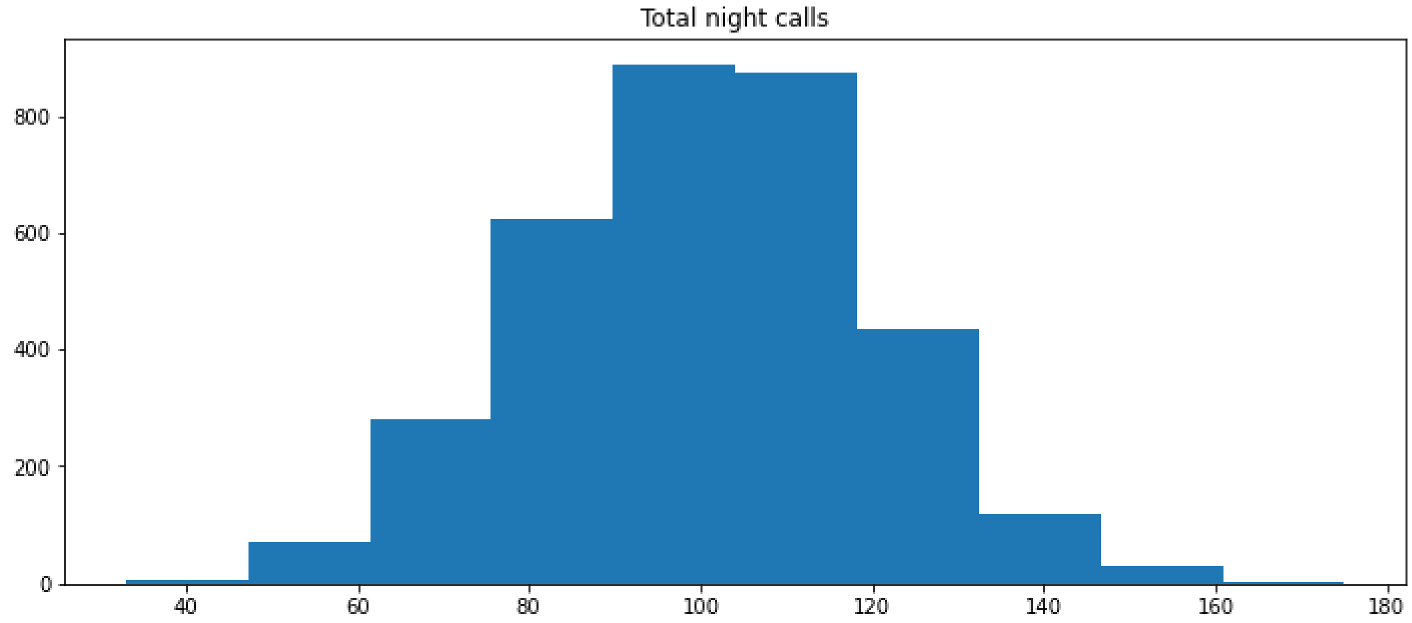
Total eve calls

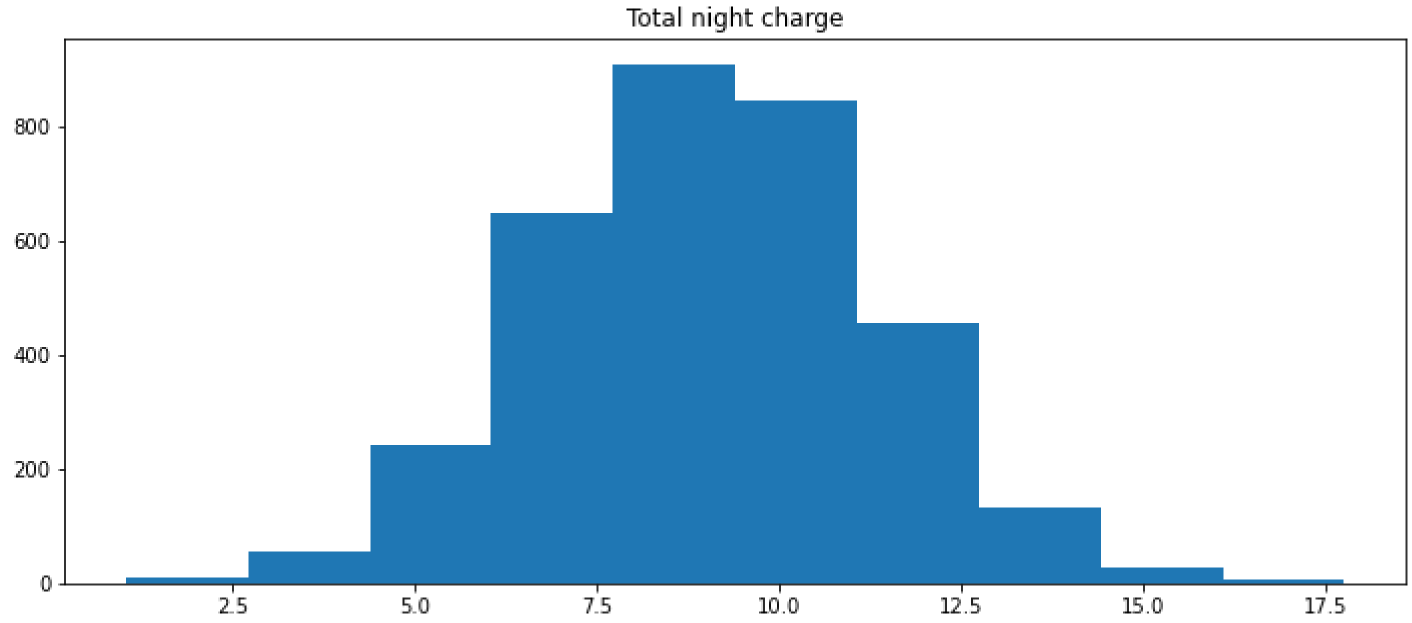


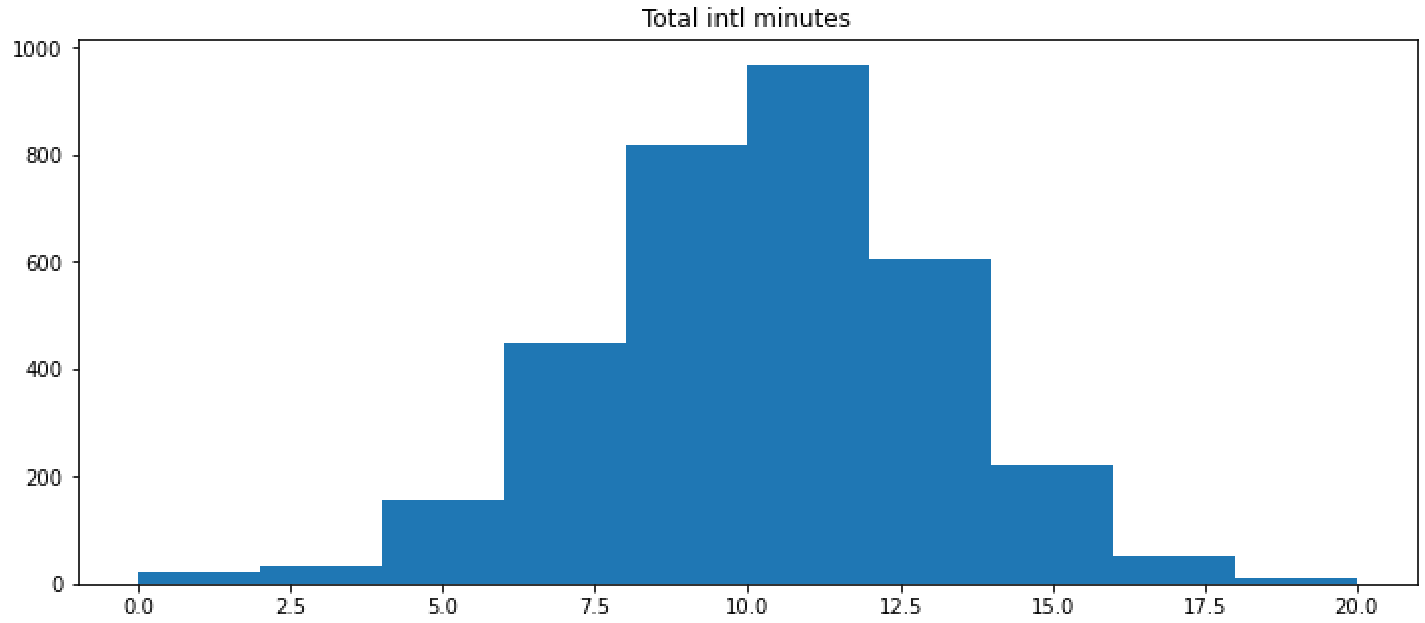
Total eve charge

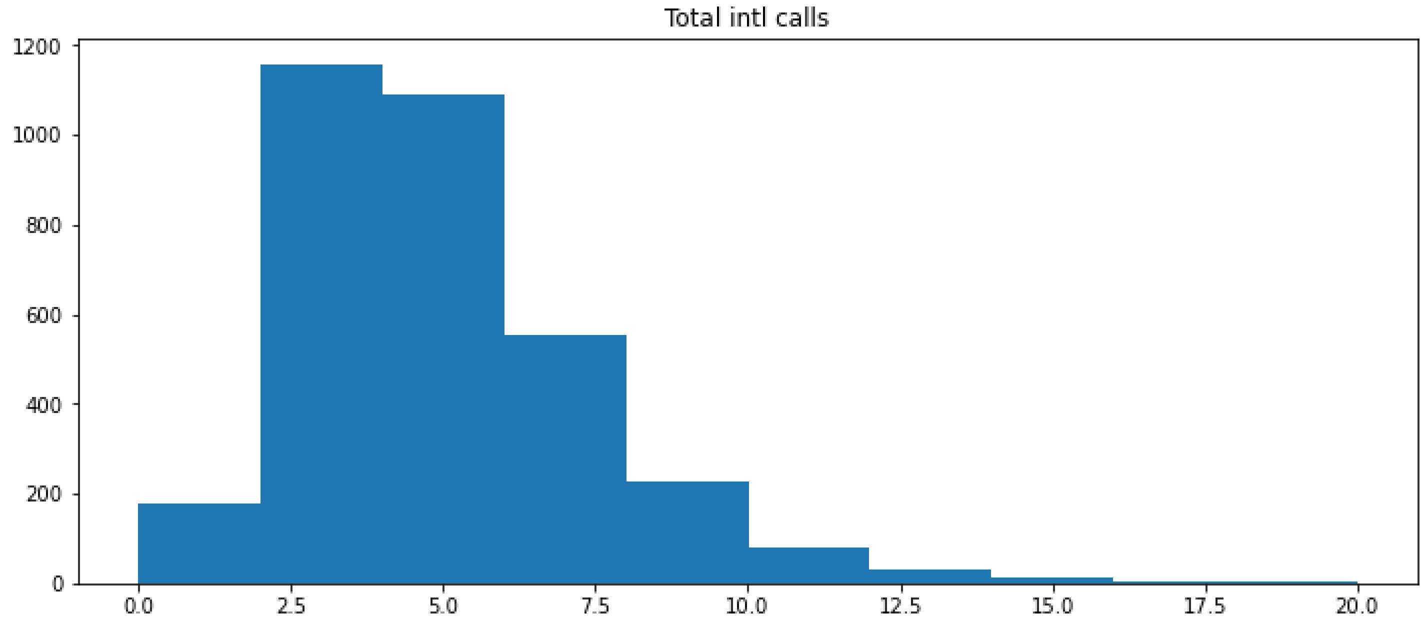


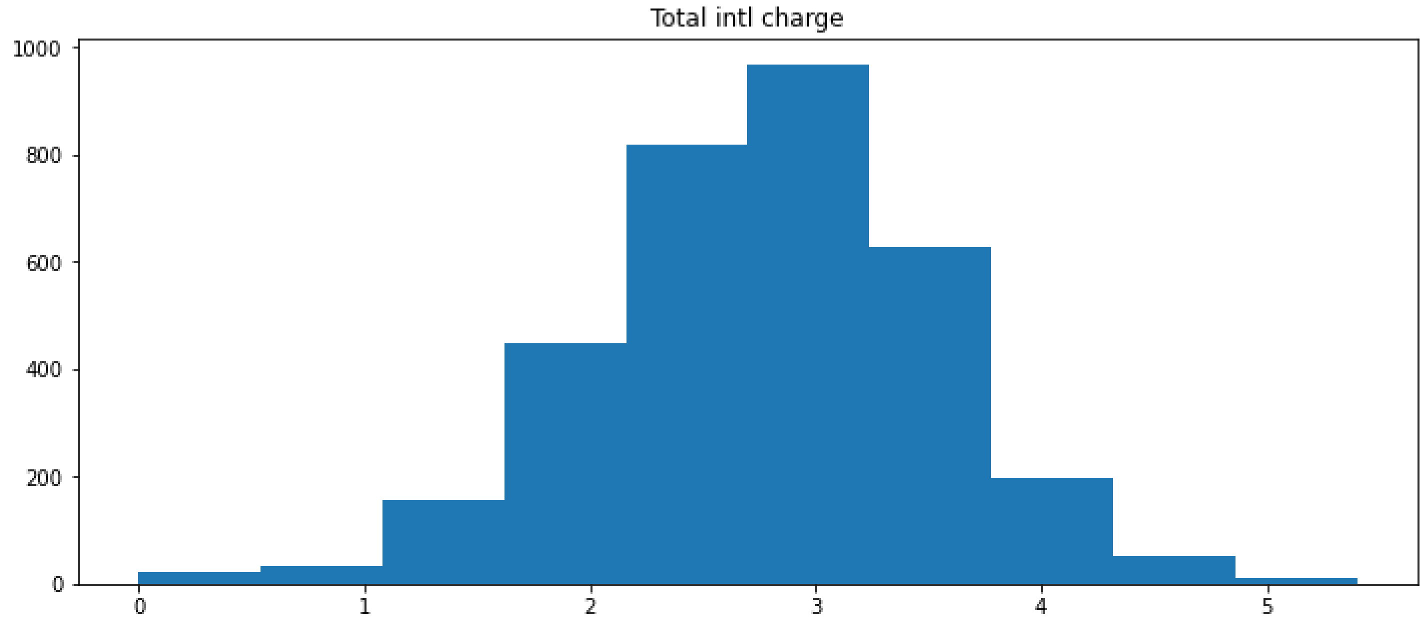

Total night minutes
0
200
400
600
800
50
100
150
200
250
300
350
400

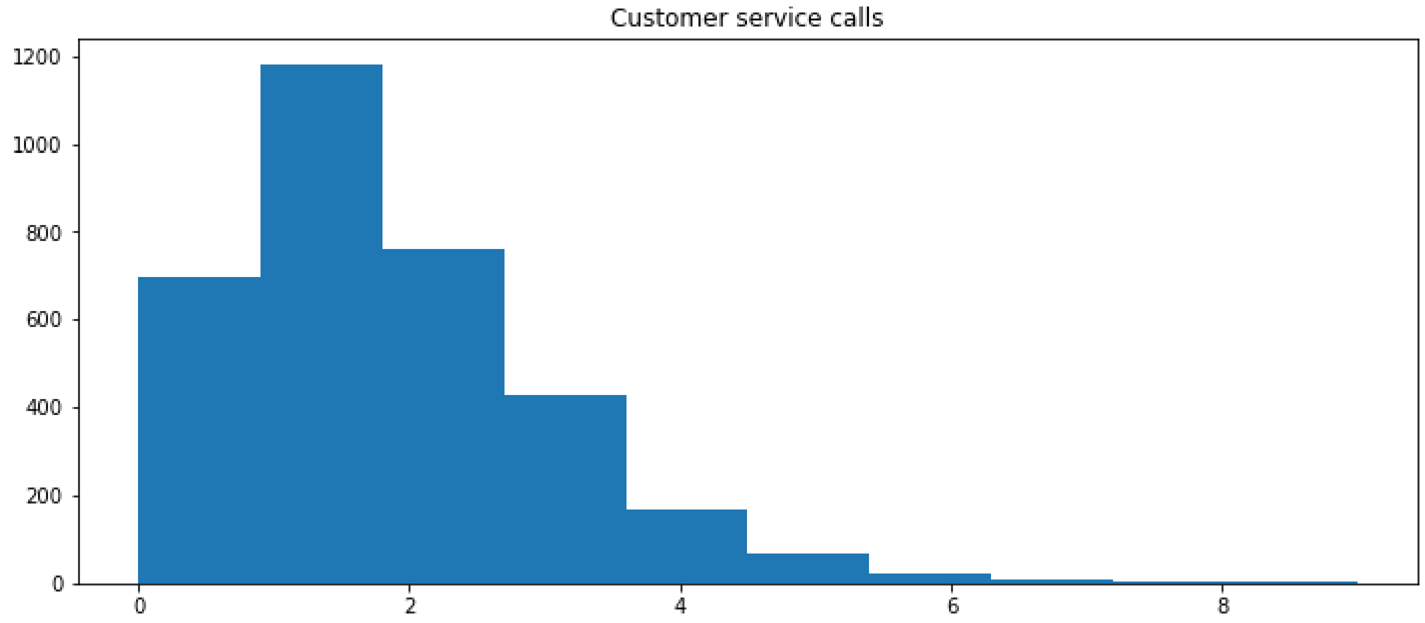

Total night calls
0
200
400
600
800
40
60
80
100
120
140
160
180


Total night charge
0
200
400
600
800
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Total intl minutes
0
200
400
600
800
1000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0







Observations -

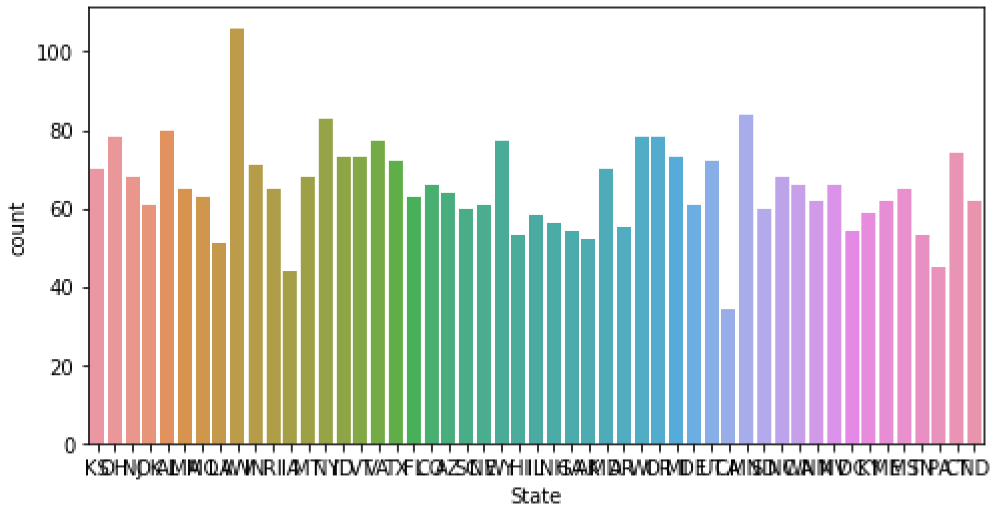
```
- total xxx minutes seems to be correlated to total xxx charge
- customer service calls, total intl calls, total day calls, total eve calls, seems to have outliers
- Number of vmail messages seems to have multimodal distribution
```

In [24]:
```
cats = ['State','Area code', 'International plan',
       'Voice mail plan','Churn']
```

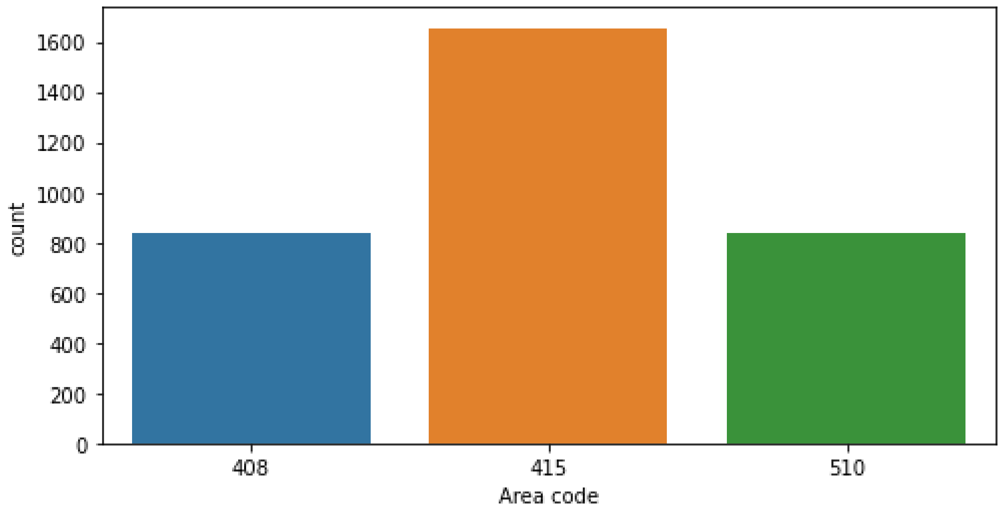
In [25]:

```
for col in cats:
    print(df[col].value_counts())

    plt.figure(figsize=(8,4))
    sns.countplot(df[col])
    plt.show()
```

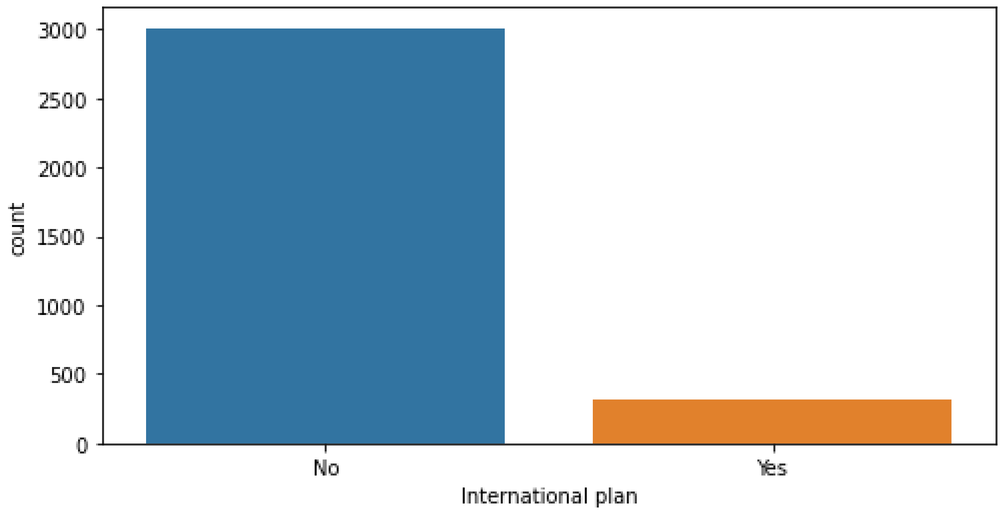
```
WV    106
MN     84
NY     83
AL     80
OR     78
OH     78
WI     78
VA     77
WY     77
CT     74
VT     73
ID     73
MI     73
TX     72
UT     72
IN     71
MD     70
KS     70
MT     68
NJ     68
NC     68
NV     66
WA     66
CO     66
RI     65
MS     65
MA     65
AZ     64
FL     63
MO     63
NM     62
ME     62
ND     62
OK     61
DE     61
NE     61
SC     60
SD     60
KY     59
IL     58
NH     56
AR     55
GA     54
DC     54
TN     53
HI     53
AK     52
LA     51
PA     45
IA     44
CA     34
Name: State, dtype: int64
```



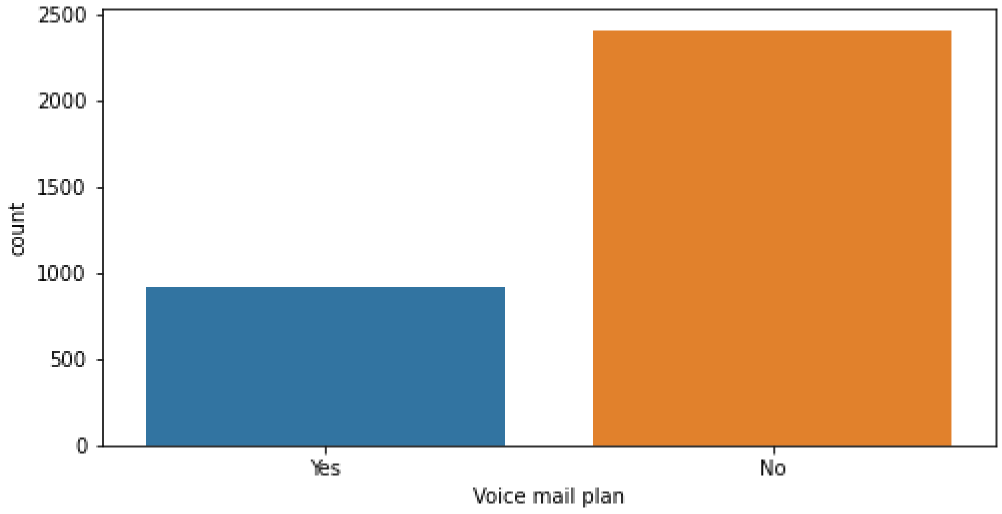
```
415    1655
510     840
408     838
Name: Area code, dtype: int64
```



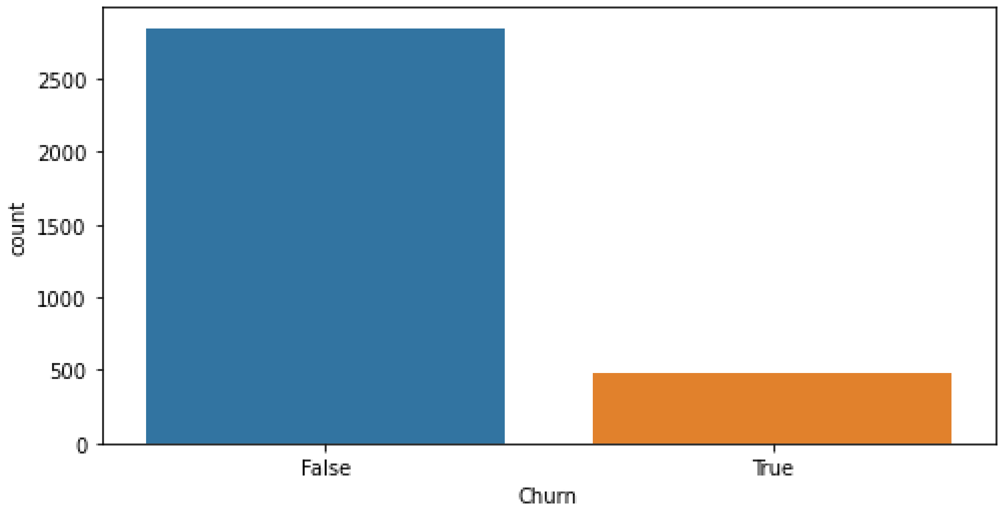
```
No     3010
Yes     323
Name: International plan, dtype: int64
```



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



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



Observation -

```
- Area Code - Almost half of customers are from area code 415, rest 1/4 from each area code
- International Plan - almost 90% of customers do not have international plans
- Approx 30% of customers have opted for voice mail plan
- Churn - almost 14% of customers have left the telecom company
```

Exploratory Analysis

```
- Correlation Analysis
- ANOVA
- Chi Square analysis
```

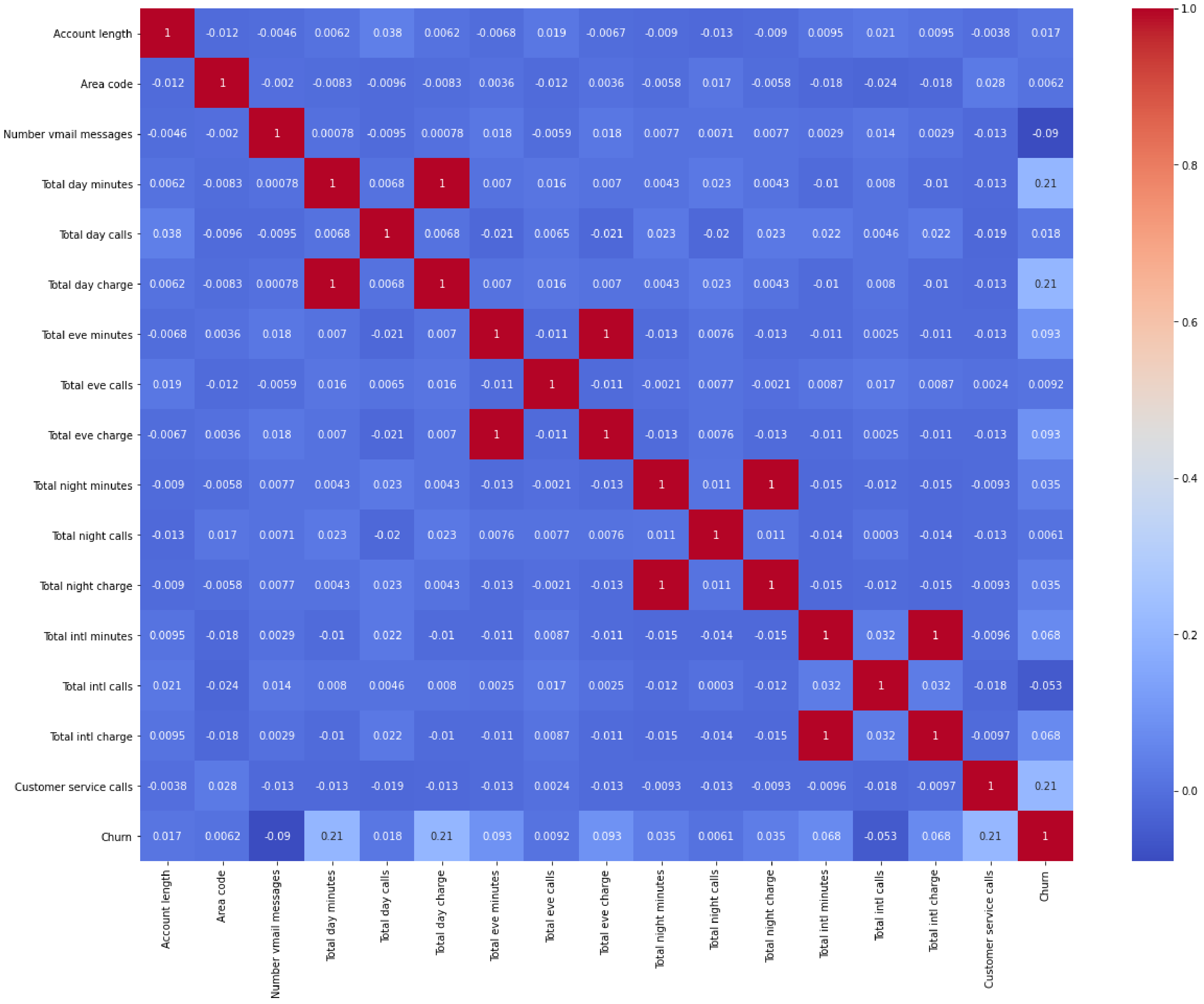
Correlation Analysis

In [26]:
```
corr = df.corr()
corr
```

Out[26]:

	Account length	Area code	Number vmail 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	Tota mi
Account length	1.000000	-0.012463	-0.004628	0.006216	0.038470	0.006214	-0.006757	0.019260	-0.006745	-0.008955	-0.013176	-0.008960	0.00
Area code	-0.012463	1.000000	-0.001994	-0.008264	-0.009646	-0.008264	0.003580	-0.011886	0.003607	-0.005825	0.016522	-0.005845	-0.01
Number vmail messages	-0.004628	-0.001994	1.000000	0.000778	-0.009548	0.000776	0.017562	-0.005864	0.017578	0.007681	0.007123	0.007663	0.00
Total day minutes	0.006216	-0.008264	0.000778	1.000000	0.006750	1.000000	0.007043	0.015769	0.007029	0.004323	0.022972	0.004300	-0.01
Total day calls	0.038470	-0.009646	-0.009548	0.006750	1.000000	0.006753	-0.021451	0.006462	-0.021449	0.022938	-0.019557	0.022927	0.02
Total day charge	0.006214	-0.008264	0.000776	1.000000	0.006753	1.000000	0.007050	0.015769	0.007036	0.004324	0.022972	0.004301	-0.01
Total eve minutes	-0.006757	0.003580	0.017562	0.007043	-0.021451	0.007050	1.000000	-0.011430	1.000000	-0.012584	0.007586	-0.012593	-0.01
Total eve calls	0.019260	-0.011886	-0.005864	0.015769	0.006462	0.015769	-0.011430	1.000000	-0.011423	-0.002093	0.007710	-0.002056	0.00
Total eve charge	-0.006745	0.003607	0.017578	0.007029	-0.021449	0.007036	1.000000	-0.011423	1.000000	-0.012592	0.007596	-0.012601	-0.01
Total night minutes	-0.008955	-0.005825	0.007681	0.004323	0.022938	0.004324	-0.012584	-0.002093	-0.012592	1.000000	0.011204	0.999999	-0.01
Total night calls	-0.013176	0.016522	0.007123	0.022972	-0.019557	0.022972	0.007586	0.007710	0.007596	0.011204	1.000000	0.011188	-0.01
Total night charge	-0.008960	-0.005845	0.007663	0.004300	0.022927	0.004301	-0.012593	-0.002056	-0.012601	0.999999	0.011188	1.000000	-0.01
Total intl minutes	0.009514	-0.018288	0.002856	-0.010155	0.021565	-0.010157	-0.011035	0.008703	-0.011043	-0.015207	-0.013605	-0.015214	1.00
Total intl calls	0.020661	-0.024179	0.013957	0.008033	0.004574	0.008032	0.002541	0.017434	0.002541	-0.012353	0.000305	-0.012329	0.03
Total intl charge	0.009546	-0.018395	0.002884	-0.010092	0.021666	-0.010094	-0.011067	0.008674	-0.011074	-0.015180	-0.013630	-0.015186	0.99
Customer service calls	-0.003796	0.027572	-0.013263	-0.013423	-0.018942	-0.013427	-0.012985	0.002423	-0.012987	-0.009288	-0.012802	-0.009277	-0.00
Churn	0.016541	0.006174	-0.089728	0.205151	0.018459	0.205151	0.092796	0.009233	0.092786	0.035493	0.006141	0.035496	0.06

In [27]:
```
# heatmap for correlation matrix
plt.figure(figsize=(20,15))
sns.heatmap(corr,annot=True,cmap="coolwarm")
plt.show()
```



Observations -

```
- total xx minutes are correlated to total xx charge, total xx charge is a multiplier of total xx minutes
- Total day charge has slightly good correlation with churn - customers paying higher charges are more likely to leave
- Customer service calls, seems to have high correlation with churn - customers making more calls, have higher chances to l
eave the telecom service provider.
```

ANOVA - Analysis of variance - f test

In [28]:
```
nums
```

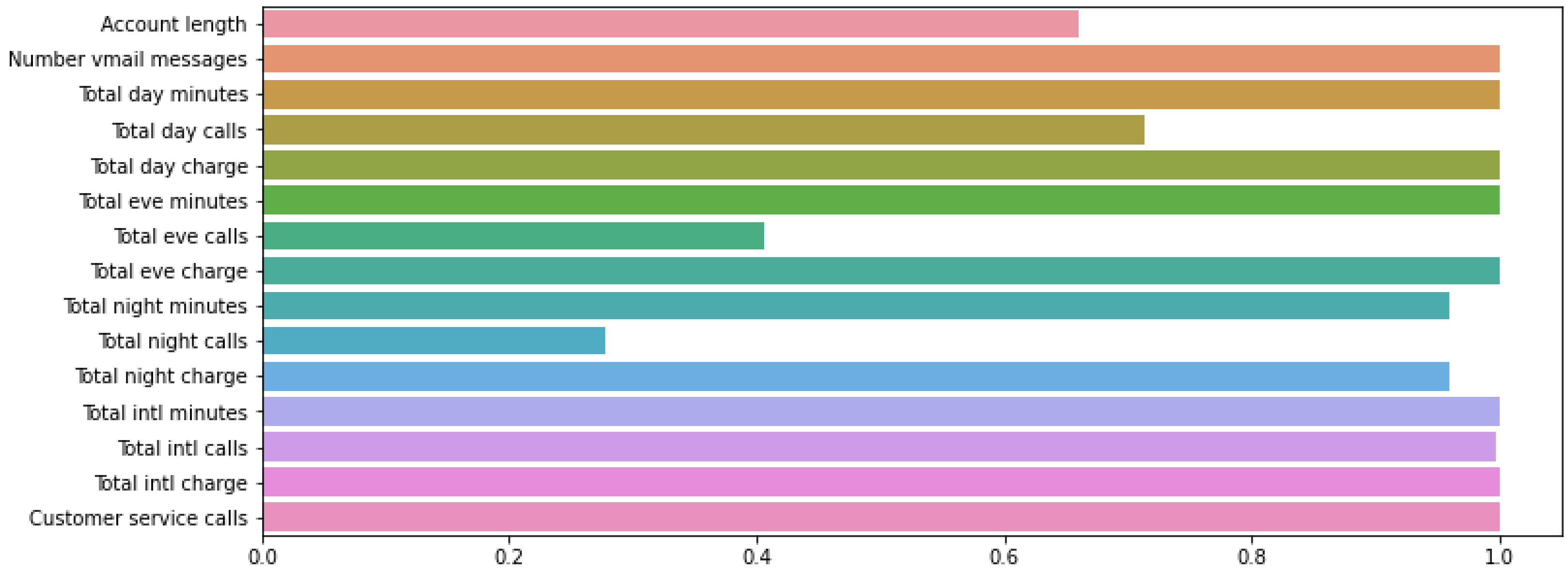
Out[28]:
```
['Account length',
 'Number vmail messages',
 'Total day minutes',
 'Total day calls',
 'Total day charge',
 'Total eve minutes',
 'Total eve calls',
 'Total eve charge',
 'Total night minutes',
 'Total night calls',
 'Total night charge',
 'Total intl minutes',
 'Total intl calls',
 'Total intl charge',
 'Customer service calls']
```

In [29]:
```
xnum = df[nums]
y = df['Churn']

from sklearn.feature_selection import f_classif
fvalue, pvalue = f_classif(xnum,y)
```

In [30]:
```
for i in range(len(nums)):
    print(nums[i],pvalue[i])
```

```
Account length 0.33976000705720666
Number vmail messages 2.1175218402696038e-07
Total day minutes 5.300278227509361e-33
Total day calls 0.28670102402211844
Total day charge 5.30060595239102e-33
Total eve minutes 8.011338561256927e-08
Total eve calls 0.5941305829720491
Total eve charge 8.0365242277544477e-08
Total night minutes 0.040466484637588881
Total night calls 0.7230277872081609
Total night charge 0.040451218769160205
Total intl minutes 8.05731126549437e-05
Total intl calls 0.002274701409850077
Total intl charge 8.0187535830472557e-05
Customer service calls 3.900360240185746e-34
```

In [49]:
```
pvalue2 = 1-pvalue

plt.figure(figsize=(12,5))
sns.barplot(x=pvalue2,y=nums)
#plt.yticks(ticks=np.arange(len(nums)),labels=nums,rotation=90)
plt.show()
```



Observation -

```
- important features - Number vmail messages, total day minutes, total eve mins, total int mins, customer service calls, to
tal intl calls
```

Chi Square test

In [51]:
```
cats
```

Out[51]:
```
['State', 'Area code', 'International plan', 'Voice mail plan', 'Churn']
```

In [52]:
```
cats = ['State', 'Area code', 'International plan', 'Voice mail plan']
xcat = df[cats]
y = df['Churn']
```

In [53]:

```
from sklearn.preprocessing import LabelEncoder

xcat['State'] = LabelEncoder().fit_transform(xcat['State'])
xcat['International plan'] = LabelEncoder().fit_transform(xcat['International plan'])
xcat['Voice mail plan'] = LabelEncoder().fit_transform(xcat['Voice mail plan'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports until
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  after removing the cwd from sys.path.
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  """
```

In [54]:

```
from sklearn.feature_selection import chi2
chival, pvalue = chi2(xcat,y)
```

In [55]:

```
for i in range(len(cats)):
    print(cats[i],pvalue[i])
```

```
State 0.19214978695607624
Area code 0.47015277286099566
International plan 4.091734729415479e-46
Voice mail plan 5.28486023170551e-07
```

Observation -

```
- Important features - International Plan, Voice mail plan"
```

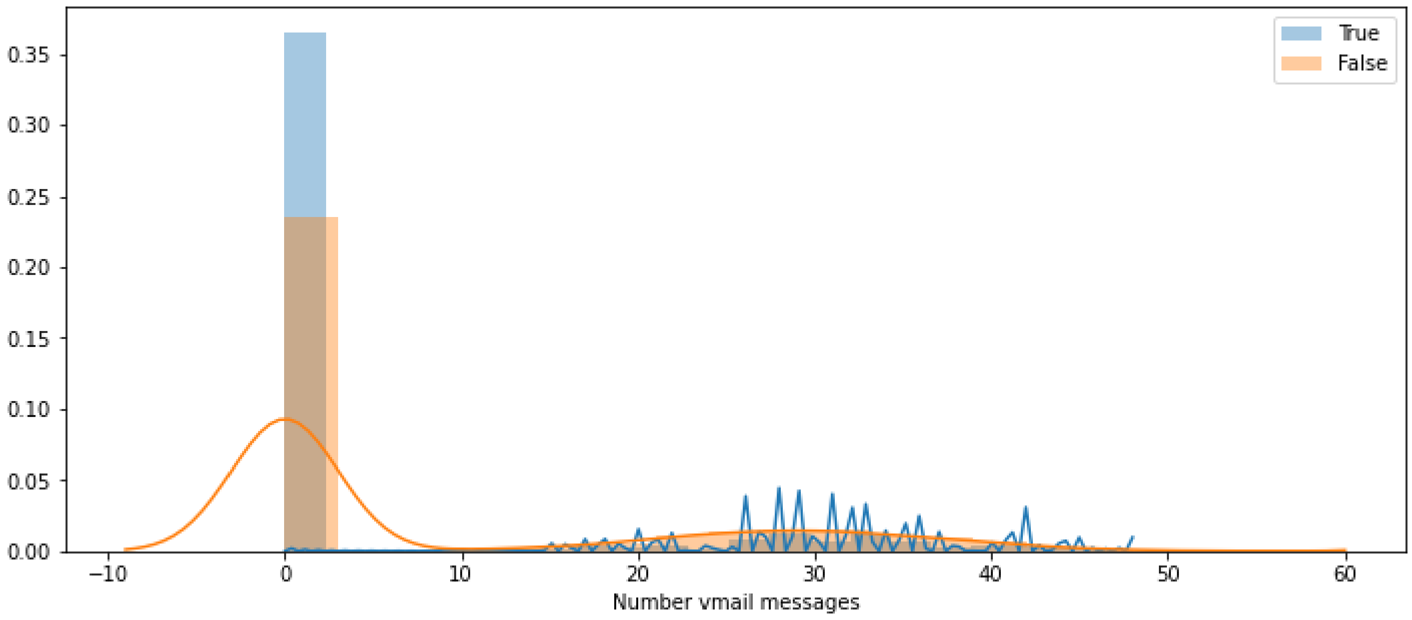
Data Visualization

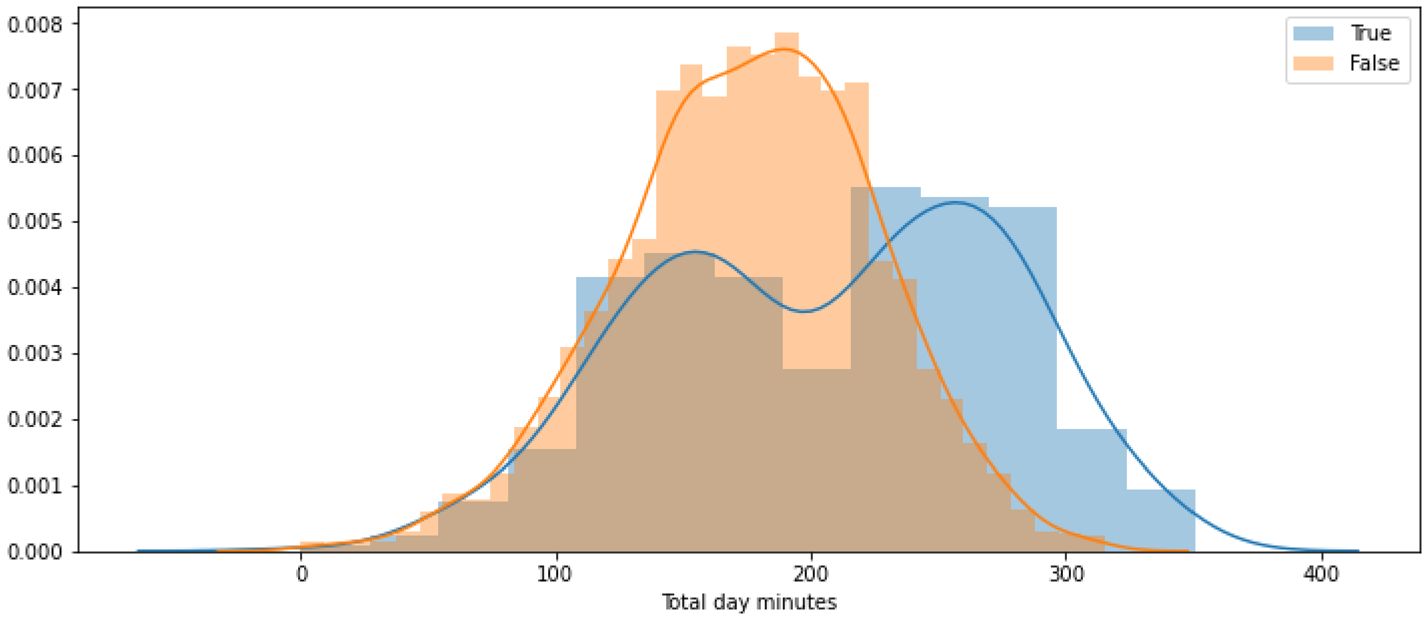
In [56]:

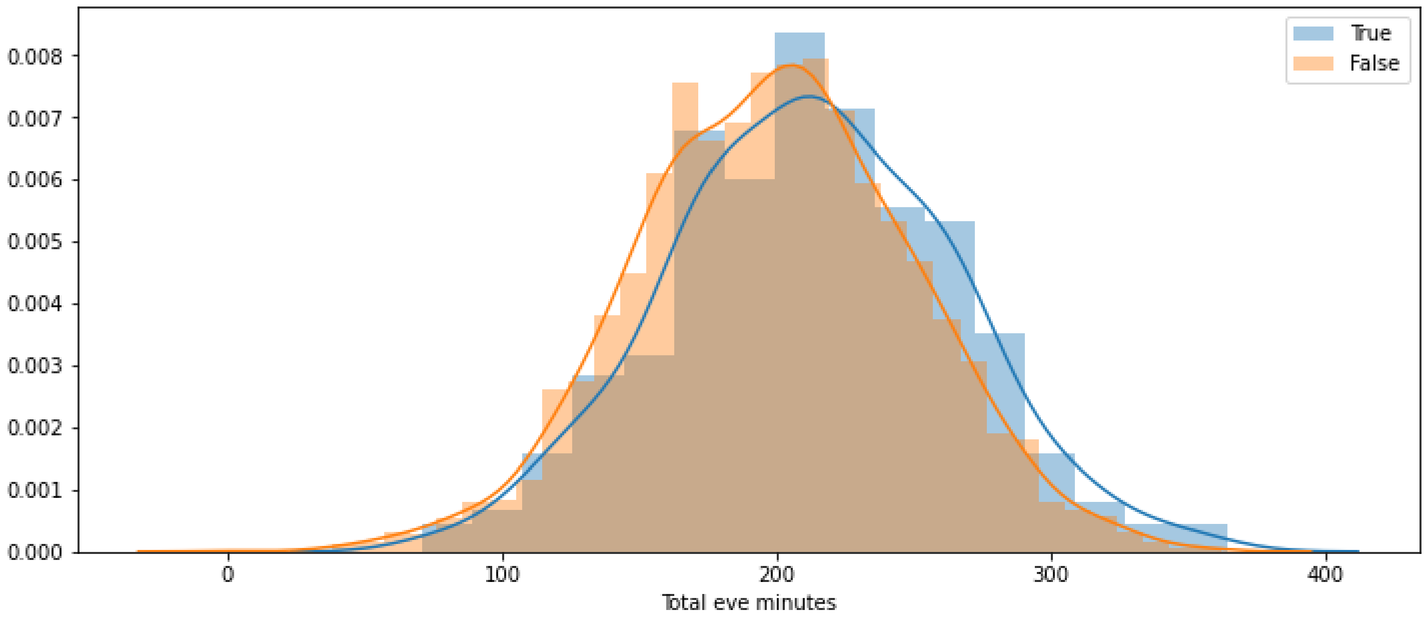
```
nums = ['Number vmail messages', 'Total day minutes','Total eve minutes',
       'Total night minutes','Total intl minutes',
       'Total intl calls','Customer service calls']
```

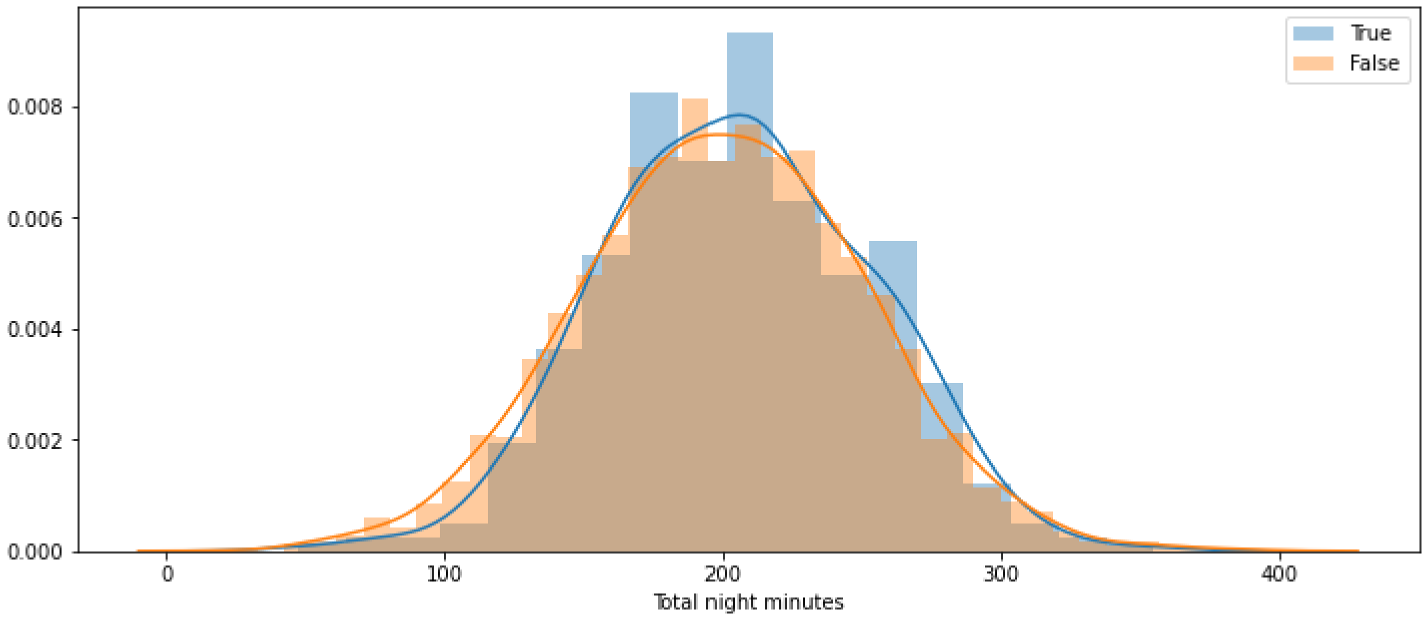
In [57]:

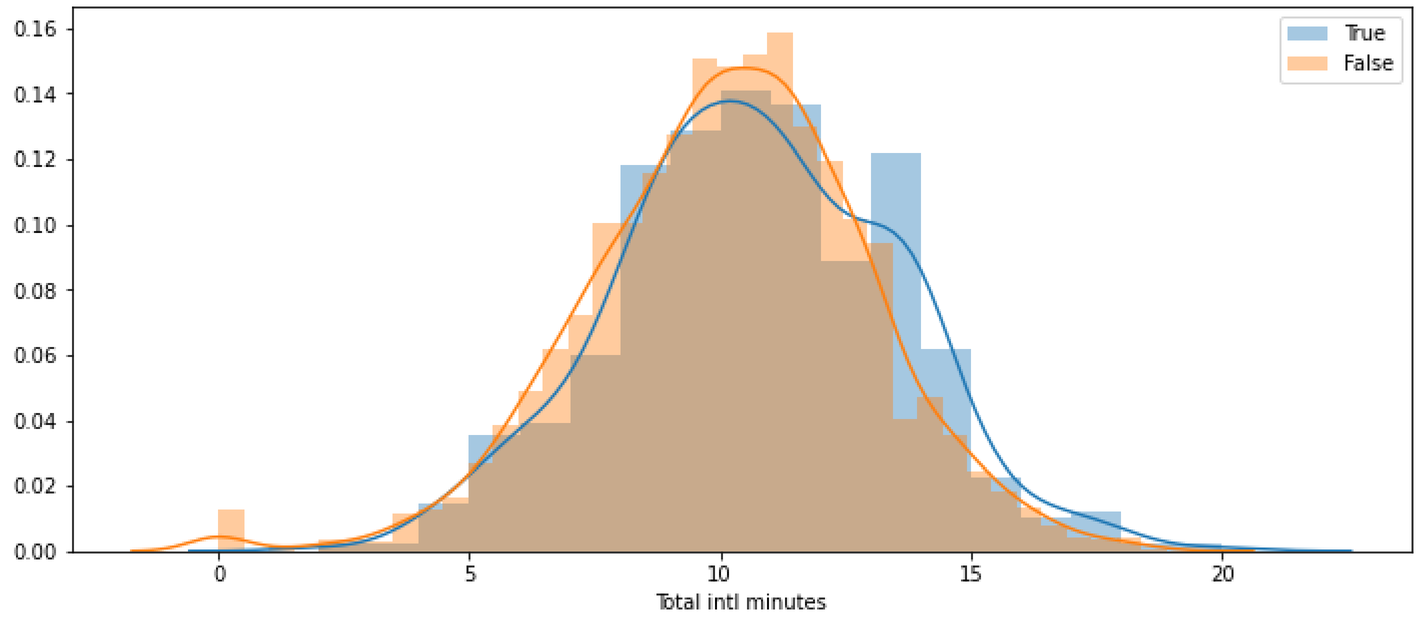
```
for col in nums:
    plt.figure(figsize=(12,5))
    sns.distplot(df[col][df.Churn==True])
    sns.distplot(df[col][df.Churn==False])
    plt.legend([True,False])
    plt.show()
```

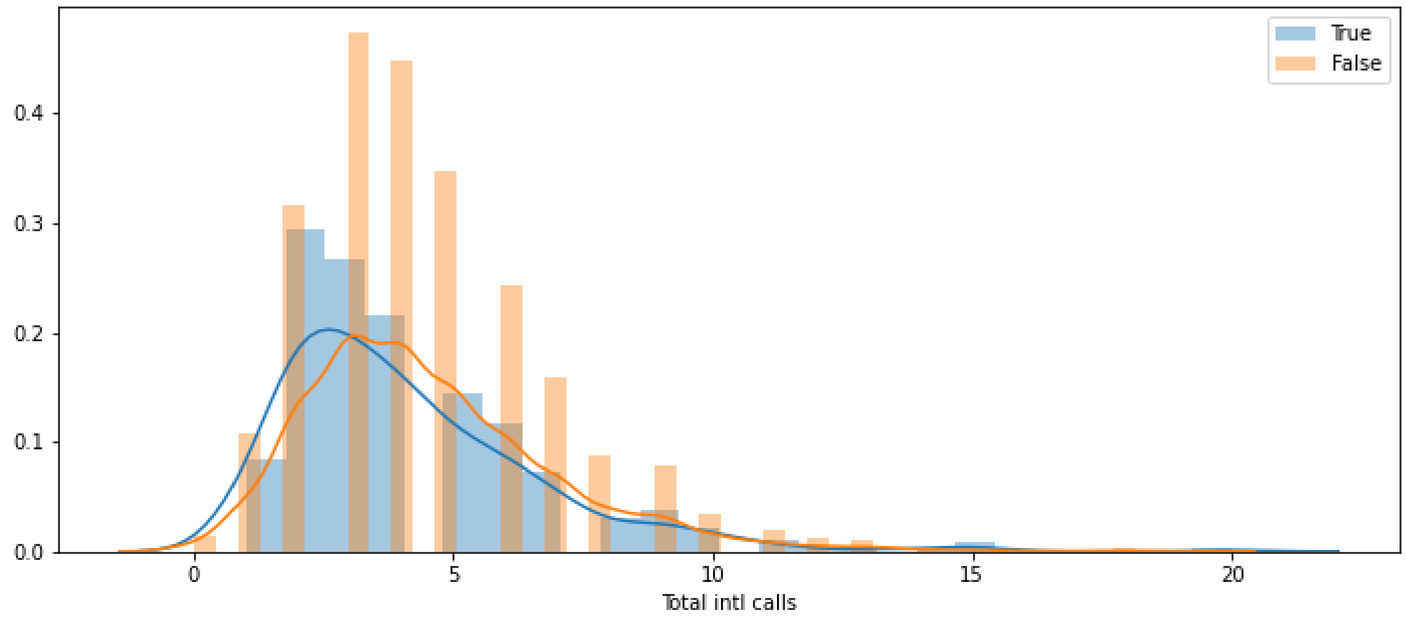

True
False
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−10
0
10
20
30
40
50
60
Number vmail messages

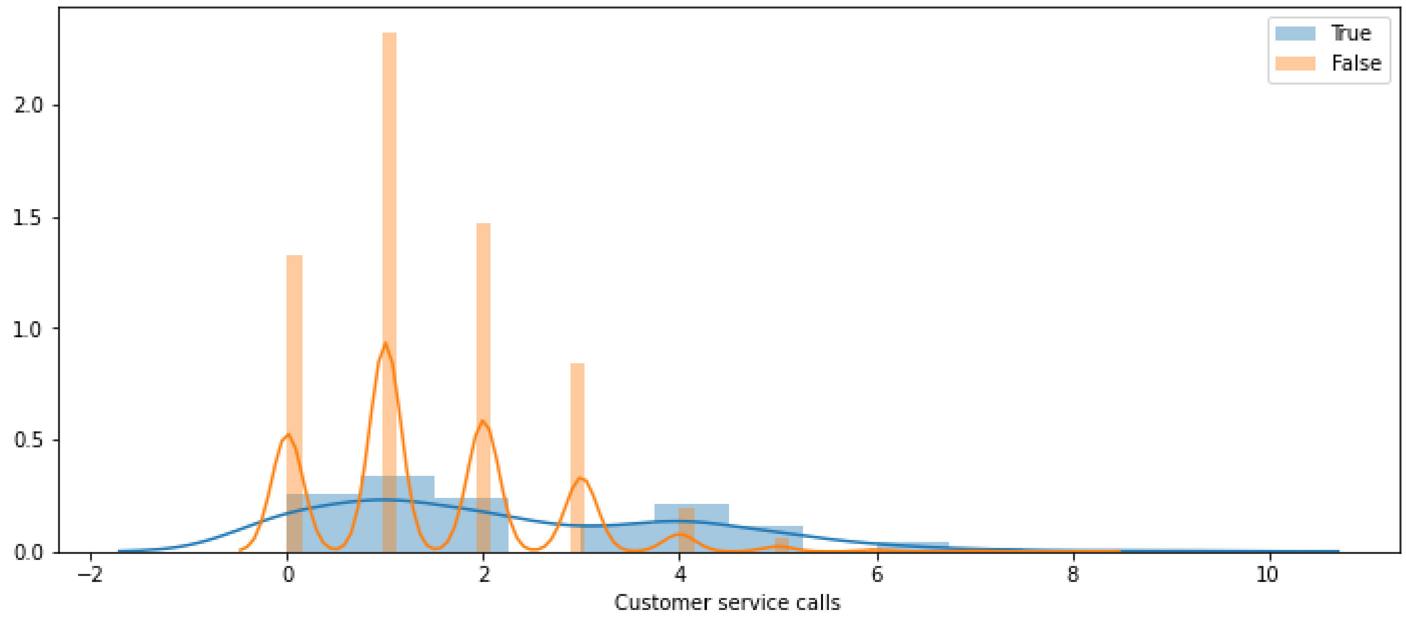

True
False
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
100
200
300
400
Total day minutes


True
False
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
100
200
300
400
Total eve minutes


True
False
0.008
0.006
0.004
0.002
0.000
0
100
200
300
400
Total night minutes





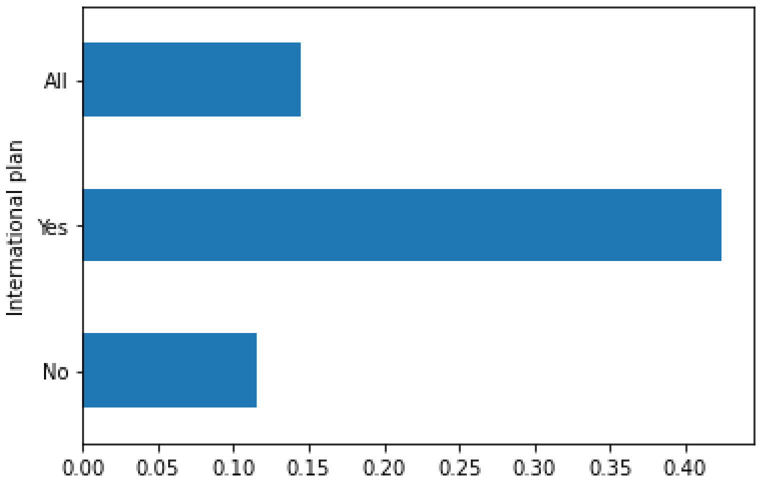


```
In [58]: cats = ['International plan', 'Voice mail plan']
```

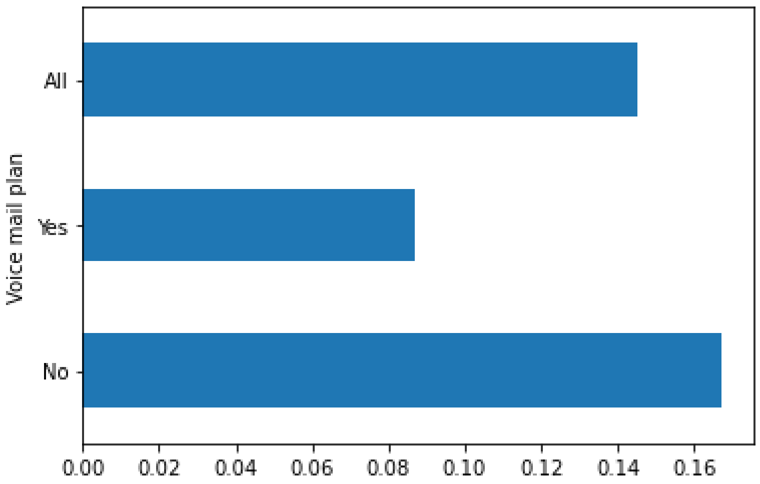
In [62]:

```
for col in cats:
    pivot = pd.crosstab(df[col],df['Churn'],margins=True)
    print(pivot)
    ratio = pivot[True]/pivot['All']
    print(ratio)
    ratio.plot(kind='barh')
    plt.show()
```

```
Churn               False  True   All
International plan
No                   2664   346  3010
Yes                   186   137   323
All                  2850   483  3333
International plan
No      0.114950
Yes     0.424149
All     0.144914
dtype: float64
```



```
Churn           False  True   All
Voice mail plan
No               2008   403  2411
Yes               842    80   922
All              2850   483  3333
Voice mail plan
No      0.167151
Yes     0.086768
All     0.144914
dtype: float64
```



In [ ]: