Elaheh Aghaarabi , Student ID: 0728476

The dataset is imported and top 5 rows of dataset are displayed.

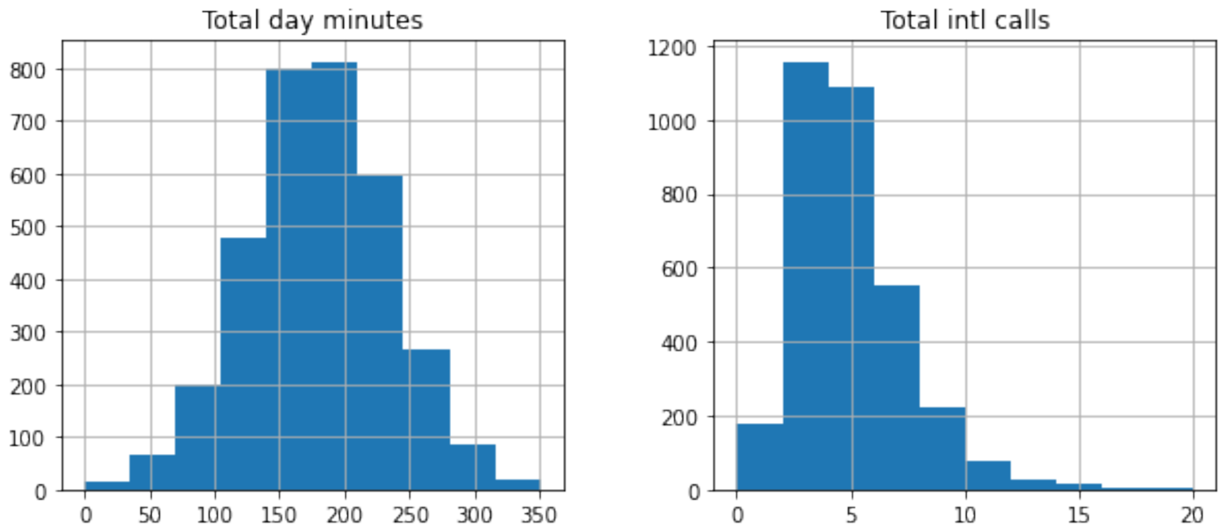
In [52]:
```
import numpy as np
import pandas as pd

import warnings
warnings.filterwarnings('ignore')

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read_csv('/Users/zzafari/Downloads/telecom_churn.csv')

df.head()
```

Out[52]:

	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	To e ca
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	1
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	1
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	1

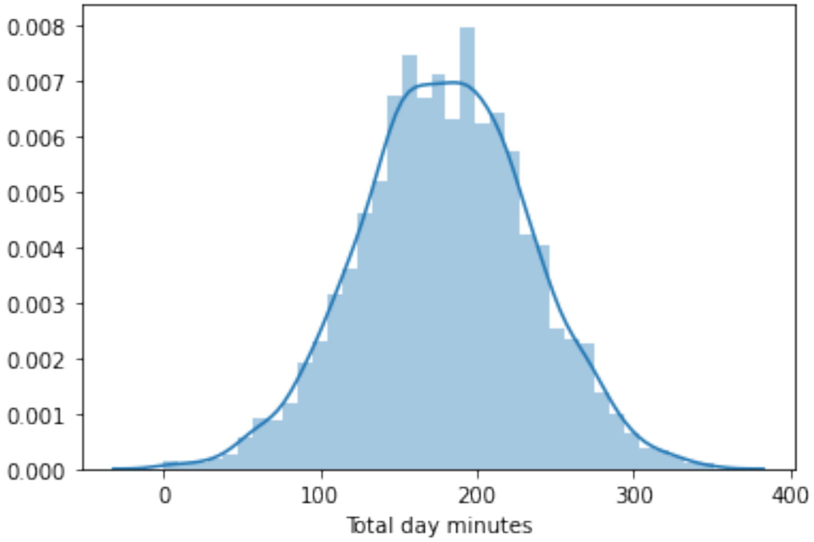
Distribution Charts: Now I will show the distribution of variables "Total day minutes" and "total intl calls" independently using histograms.

In [2]:
```
features = ['Total day minutes', 'Total intl calls']
df[features].hist(figsize=(10, 4));
```



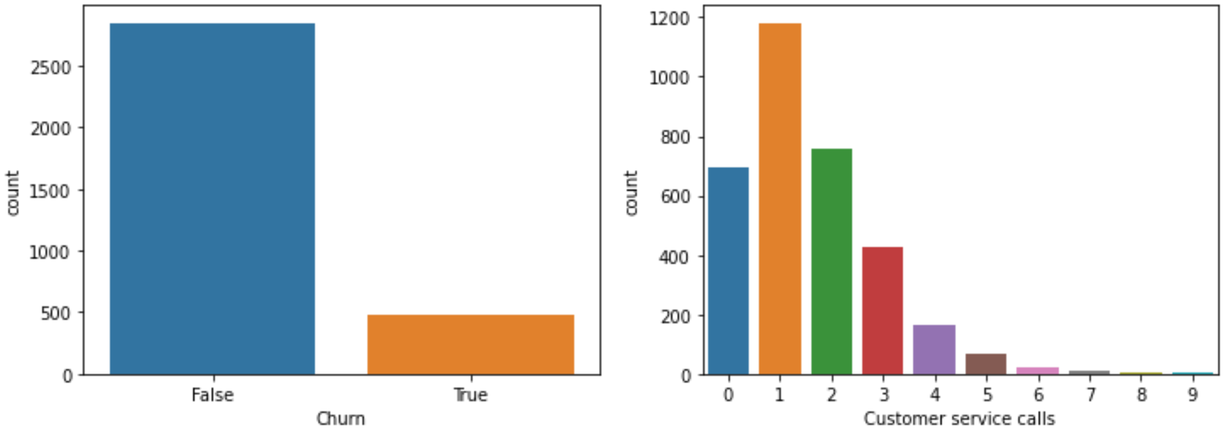
The "total day minutes" variable is normally distributed based on the figures. However, the "Total intl calls" has a longer tail at right. Below, I will show the normalized Kernel density estimate of the histogram chart of "total day minutes".

In [4]:
```
sns.distplot(df['Total day minutes']);
```



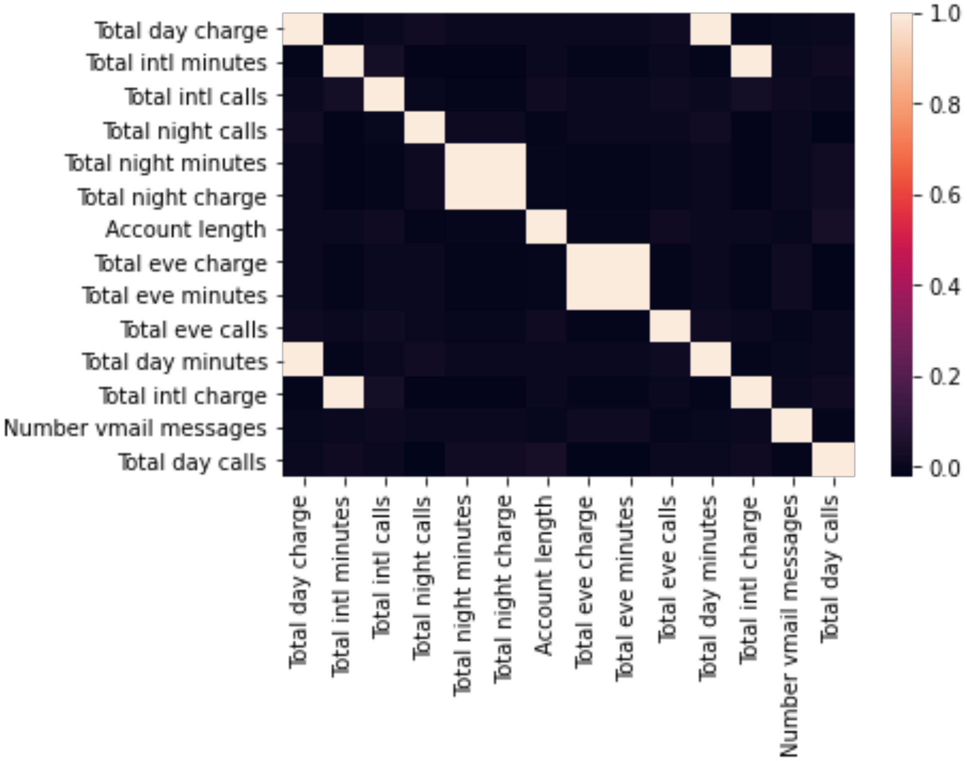
Comparison Chart: Bar plots are also useful to compare variables. Below the frequency of two categorial variables "customer service call" and "churn" are represented. The underlying distrubution can be shown using this chart too.

In [5]:
```
_, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))

sns.countplot(x='Churn', data=df, ax=axes[0]);
sns.countplot(x='Customer service calls', data=df, ax=axes[1]);
```



The "customer service calls" chart depicts that most of customers' concerns and complains are solved with less than 3 calls (0,1,2). Now I will investigate the correlation between numberical variables and will produce the correlation matrix.

In [6]:
```
numerical = list(set(df.columns) -
                 set(['State', 'International plan', 'Voice mail plan'
,
                      'Area code', 'Churn', 'Customer service calls'])
)


corr_matrix = df[numerical].corr()
sns.heatmap(corr_matrix);
```



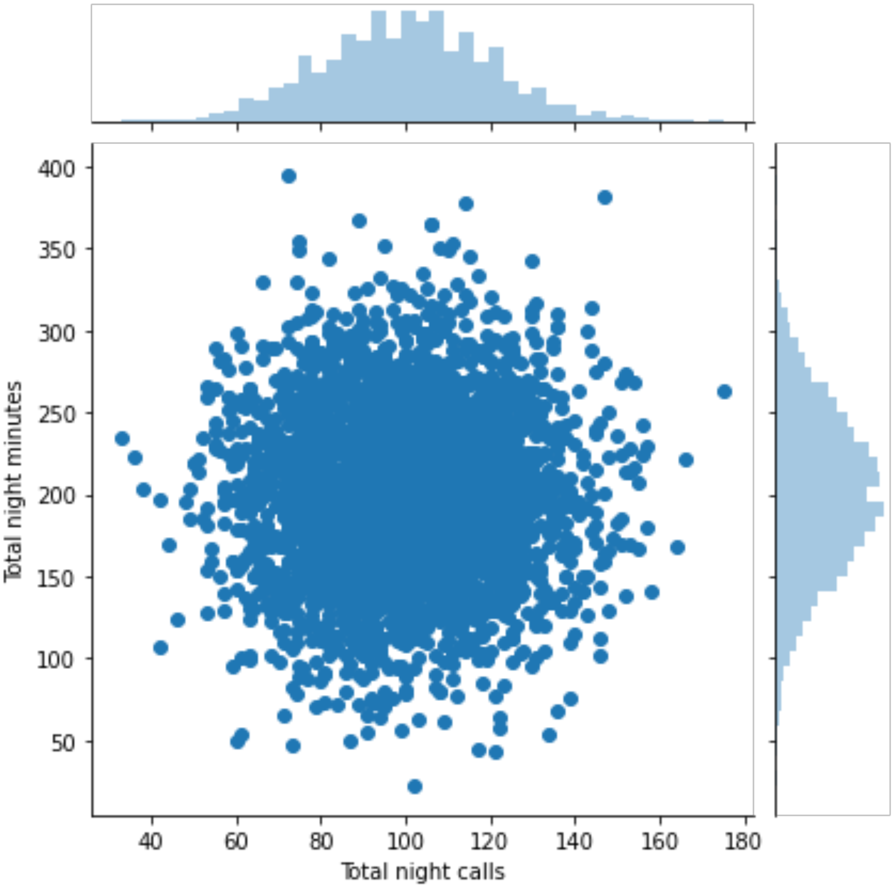
We can conclude that variables "total day charge, total, evening charge, total night chatge and total intl charge" are dependent variables and do not give us useful information. As a result, we can exclude them from dataset.

In [7]:
```
numerical = list(set(numerical) -
                 set(['Total day charge', 'Total eve charge', 'Total n
ight charge', 'Total intl charge']))
```

A scatter plot will be shown below to compare the distribution of two variables, "total night calls" and " total night minutes". Also, a scatterplot matrix containing all variables will be shown.

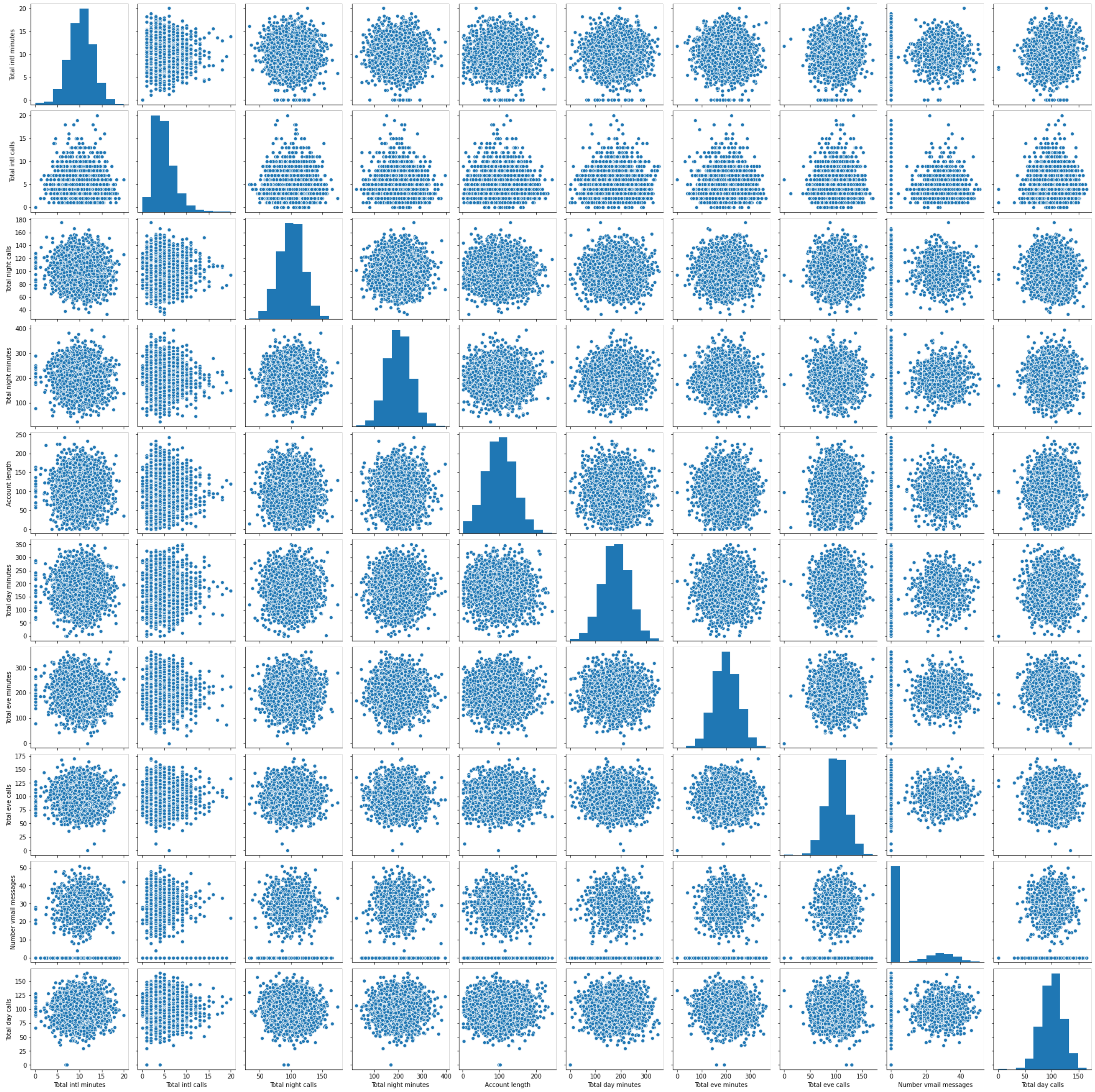
In [9]:

```
sns.jointplot(x='Total night calls', y='Total night minutes',
              data=df, kind='scatter');
```



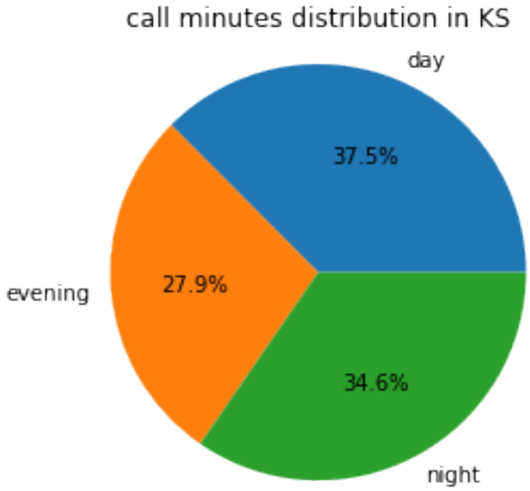
In [10]:

```
%config InlineBackend.figure_format = 'png'
sns.pairplot(df[numerical]);
```



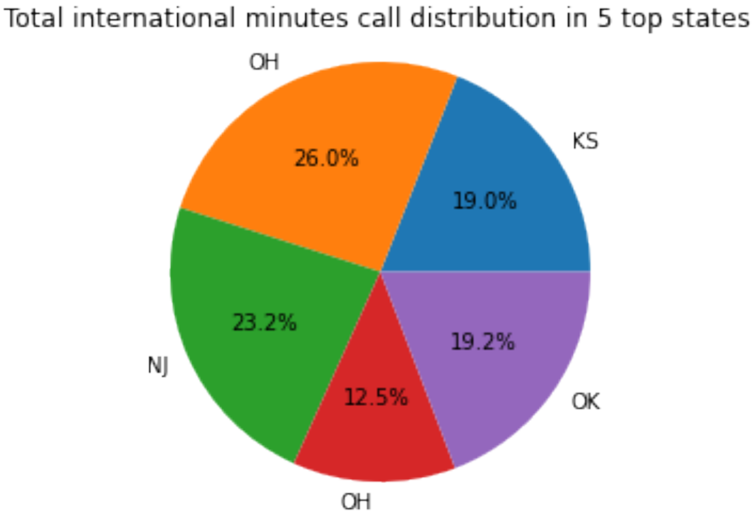
Composition charts: pie charts is shown below to display the time of call (a numerical variable) distribution in KS state.

```
In [34]: my_data = df.loc[0,['Total day minutes','Total eve minutes', 'Total night minutes']]
         my_labels = 'day','evening','night'
         plt.pie(my_data , labels=my_labels, autopct='%1.1f%%')
         plt.title('call minutes distribution in KS')
         plt.axis('equal')
         plt.show()
```



The pie chart can also show the states with most international minutes.

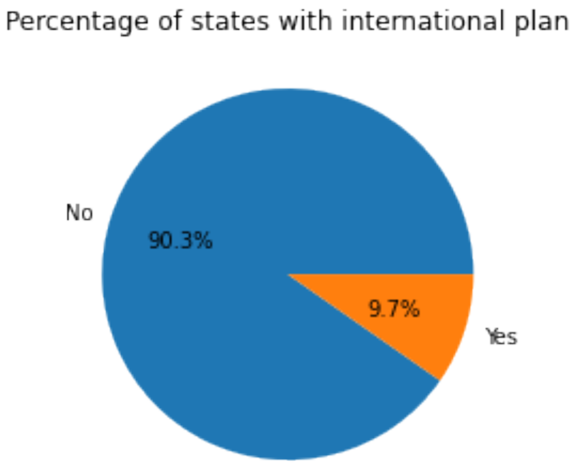
In [42]:
```
my_data = df.loc[0:4,['Total intl minutes']]
my_labels = 'KS','OH','NJ', 'OH', 'OK'
plt.pie(my_data , labels=my_labels,autopct='%1.1f%%')
plt.title('Total international minutes call distribution in 5 top states')
plt.axis('equal')
plt.show()
```



Pie chart can be used for categorical values as well. Below the percentage of states with international plan is shown.

In [50]:

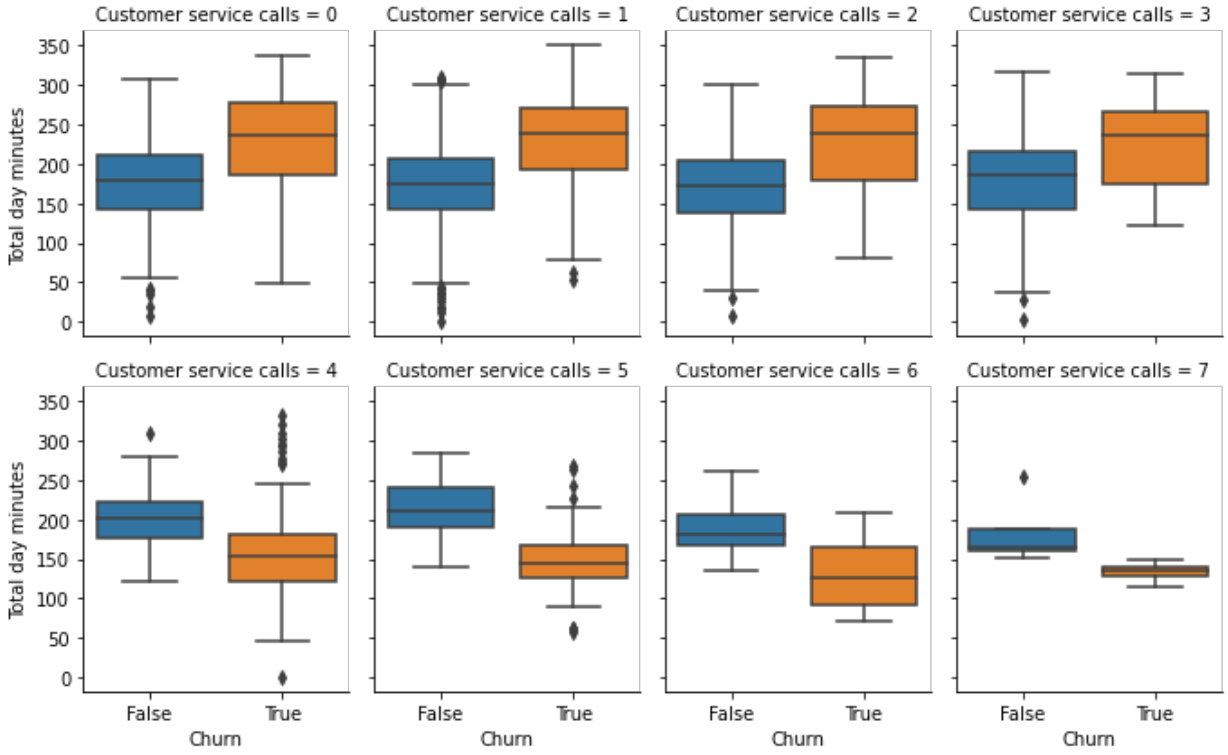
```
df_pie = df.groupby('International plan').size()

df_pie.plot(kind='pie', subplots=True, autopct='%1.1f%%')
plt.title("Percentage of states with international plan")
plt.ylabel("")
plt.show()
```



Distribution Chart: Box plot is a toll to find distribution in a dataset.

In [51]:

```
sns.catplot(x='Churn', y='Total day minutes', col='Customer service ca
lls',
               data=df[df['Customer service calls'] < 8], kind="box",
               col_wrap=4, height=3, aspect=.8);
```



In [ ]: