Learning Pandas and Matplotlib

Pandas is pythons library that enables broad possibilities for data analysis. By using Pandas it is very easy to upload, manage and analyse data from different tables by using SQL-like commands. Moreover, in connection with the libraries Matplotlib and Seaborn, Pandas gives broad opportunities to visualise the data.

In [1]:

```
import pandas as pd
import numpy as np

from matplotlib import pyplot as plt
from matplotlib.pyplot import rcParams
rcParams['figure.figsize'] = 8, 5
import seaborn as sns
```

We study the main methods of packages Pandas and Matplotlib by working with the dataset that describes the churn rate of the customers of a telecom company.

Exercise 1:

Read the data from the file 'telecom_churn.csv' and display the first 5 rows by using the method 'head'.

Hint : method read_csv may be useful

In [2]:

```
### Write Your code here ###
df= pd.read_csv('/Users/jincheong-a/Desktop/Uni-Dataanalysis/Lab01_Pandas_And
```

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	T c
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	

Exercise 2:

Display the size of the data array, information about it, and its main statistical characteristics. **Hint:** use methods shape, info, describe.

In [3]:

```
### Write Your code here ###
print(np.shape(df))
print(np.info(df))
print(df.describe)
```

```
(3333, 20)
Two-dimensional size-mutable, potentially heterogeneous tabular data
structure with labeled axes (rows and columns). Arithmetic operation
s
align on both row and column labels. Can be thought of as a dict-lik
e
container for Series objects. The primary pandas data structure.

Parameters
----------
data : numpy ndarray (structured or homogeneous), dict, or DataFrame
    Dict can contain Series, arrays, constants, or list-like objects

    .. versionchanged :: 0.23.0
       If data is a dict, argument order is maintained for Python 3.
6
       and later.

index : Index or array-like
```

Exercise 3:

Convert column 'Churn' to the int64 type. **Hint:** use method astype. Plot the distribution of the churn and loyal clients in the bar plot. The figure should look like this:

In [4]:

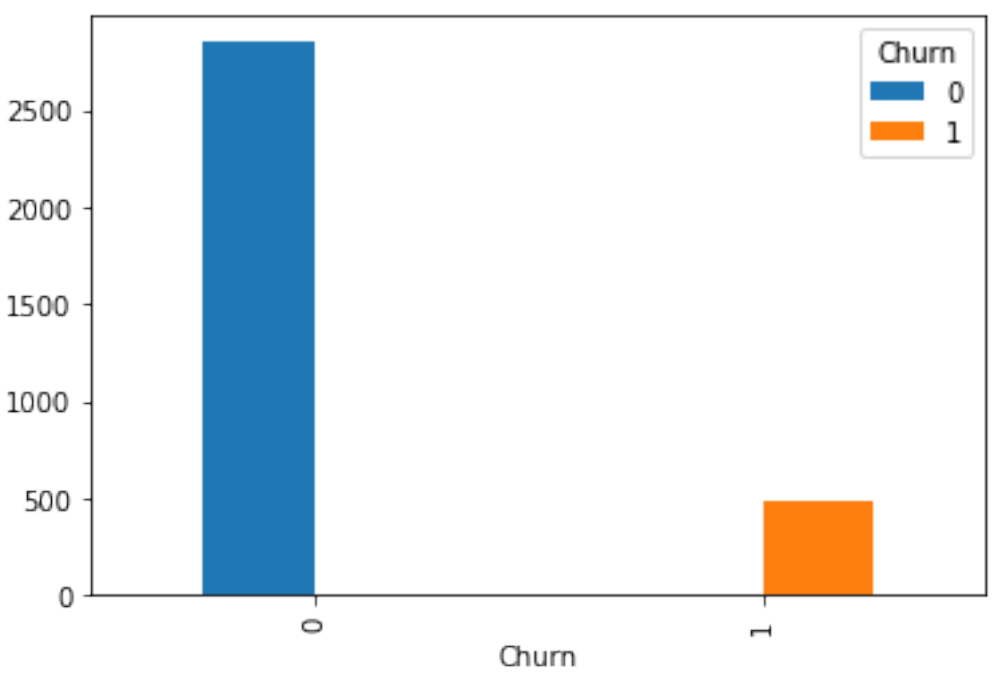
```
### Write Your code here ###
churn = df['Churn'].astype(int)
df['Churn'] = churn
```

In [13]:

```
# df['Churn'].plot(kind = 'bar')
pd.crosstab(df.Churn, df.Churn).plot(kind = 'bar')
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a25d7dc88>
```



Sorting

Exercise 4:

Sort the dataframe you have obtained in exercise 3 by the value in 'Total day charge' in descending/ascending order. Also sort it by using the column 'Churn' as the primary key and 'Total eve calls' as the secondary key. Try different combinations of ordering. **Hint:** use the method sort_values

In [6]:

```
### Write Your code here ###

#1.by the value in 'Total day charge' in descending/ascending order.

df.sort_values(by=['Total day charge'], ascending=[False]).head()
```

Out[6]:

	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
365	CO	154	415	No	No	0	350.8	75	59.64	216.5
985	NY	64	415	Yes	No	0	346.8	55	58.96	249.5
2594	OH	115	510	Yes	No	0	345.3	81	58.70	203.4
156	OH	83	415	No	No	0	337.4	120	57.36	227.4
605	MO	112	415	No	No	0	335.5	77	57.04	212.5

In [7]:

```
df.sort_values(['Churn', 'Total eve calls'], ascending=[True, False]).head()
```

Out[7]:

	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
3219	NY	150	415	No	Yes	35	139.6	72	23.73	332.8
58	WI	68	415	No	No	0	148.8	70	25.30	246.5
1020	HI	115	415	No	Yes	33	145.0	72	24.65	194.5
1706	MD	54	415	No	No	0	273.8	113	46.55	119.6
1071	PA	134	408	No	No	0	205.3	122	34.90	240.5

Indexing and extracting information from the dataframe

By using Pandas dataframes we are able to index and extract information from the dataset. You can index the information stored in dataframe either by names or by indices. In the first case you use the command loc, in the second iloc. **Hint:** Use logical indexing for the columns and the groupby method to solve the tasks.

Exercise 5:

- Display the mean churn rate of the clients.
- Now we want to analyse statistical information only for the clients, which are or aren't loyal to their telecom-company (field 'Churn' in dataframe). Extract the loyal and non-loyal clients from the table seperately and display the means of their charateristics in a single dataframe.
- How long do the non-loyal users talk during the day (on average)?
- What is the maximum length of the international calls for the loyal users that do not use the international plan?

In [ ]:

```
### Write Your code here ###

#1. Display the mean churn rate of the clients.



#2.

df_churn = df[df['Churn'] == 0]
df_loyal = df[df['Churn'] == 1]

df_churn.describe()
df_loyal.describe()


#3. Average 175 minutes for non-loyal users during the day

df_loyal.groupby(['International plan']).max()

#4. 15 is the maximal length
```

Distribution of the features

Exercise 6:

Plot the distribution of the features that have numerical values. **Hint:** use the method hist which can also be applied from the pandas dataframe. It should look like that:

What do you observe? From which probability distribution could each feature be generated?

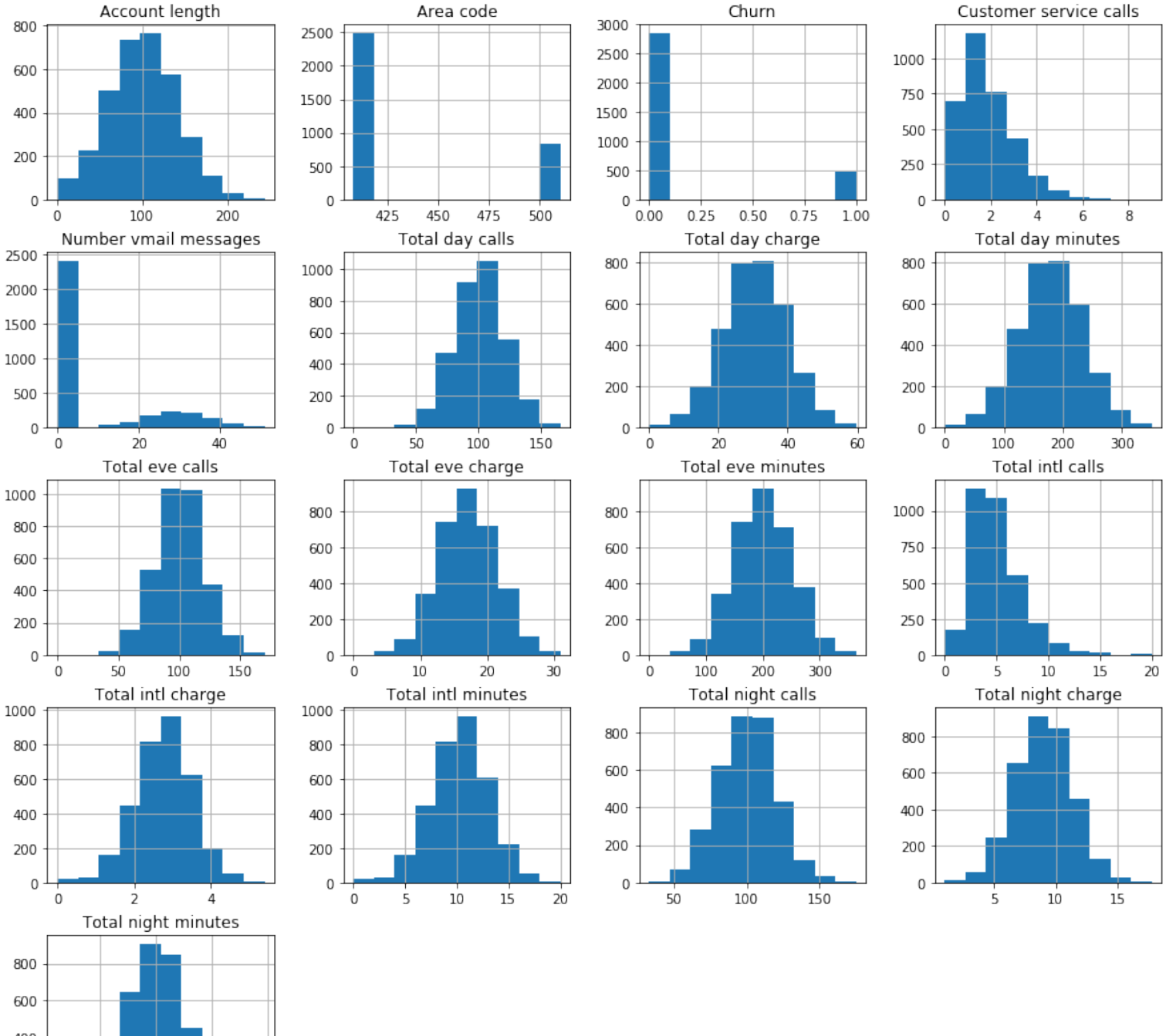
In [9]:

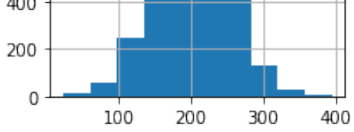
```
### Write Your code here ###
df.hist(figsize=(15, 15))
```

Out[9]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x1a218da9e
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a21ac616
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a21bbc12
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a238e9c5
0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x1a238d5b7
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a236f90f
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a241982e
8>,
```

```
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a241d558
8>],
      [<matplotlib.axes._subplots.AxesSubplot object at 0x1a241d55c
0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a241f3a2
0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a2386a16
0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a2383fef
0>],
      [<matplotlib.axes._subplots.AxesSubplot object at 0x1a2429c19
8>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a2427f40
0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a242a066
8>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a242f08d
0>],
      [<matplotlib.axes._subplots.AxesSubplot object at 0x1a242d6b3
8>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a23814da
0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a2381304
8>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a238322b
0>]],
      dtype=object)
```





In [10]:

```
# Customer service calls and Total intl calls have right skewed distribution.
# Area code,Number vmail messages, or Churn have binary classification.
# The rest of the features have more or less normal distribution. But, Number
```

Pivot tables and graphics

We want to see how the instances are distributed between two categories: 'International plan' and 'Churn'.

Exercise 7:

- Build the cross table between the features using the method *crosstab*.
- Visualize the distribution for the feature 'Churn', depending on the value of the features 'International plan', 'Voice mail plan', and 'Customer service calls'. **Hint:** Use commands plt.sublot and sns.subplots.

Your plot should look something like this:

What do you see? What conclusions can be drawn? What feature (intuitively) can be more important for Churn prediction?

In [11]:

```
### Write Your code here ###

pd.crosstab(df['International plan'], df.Churn)

```

Out[11]:

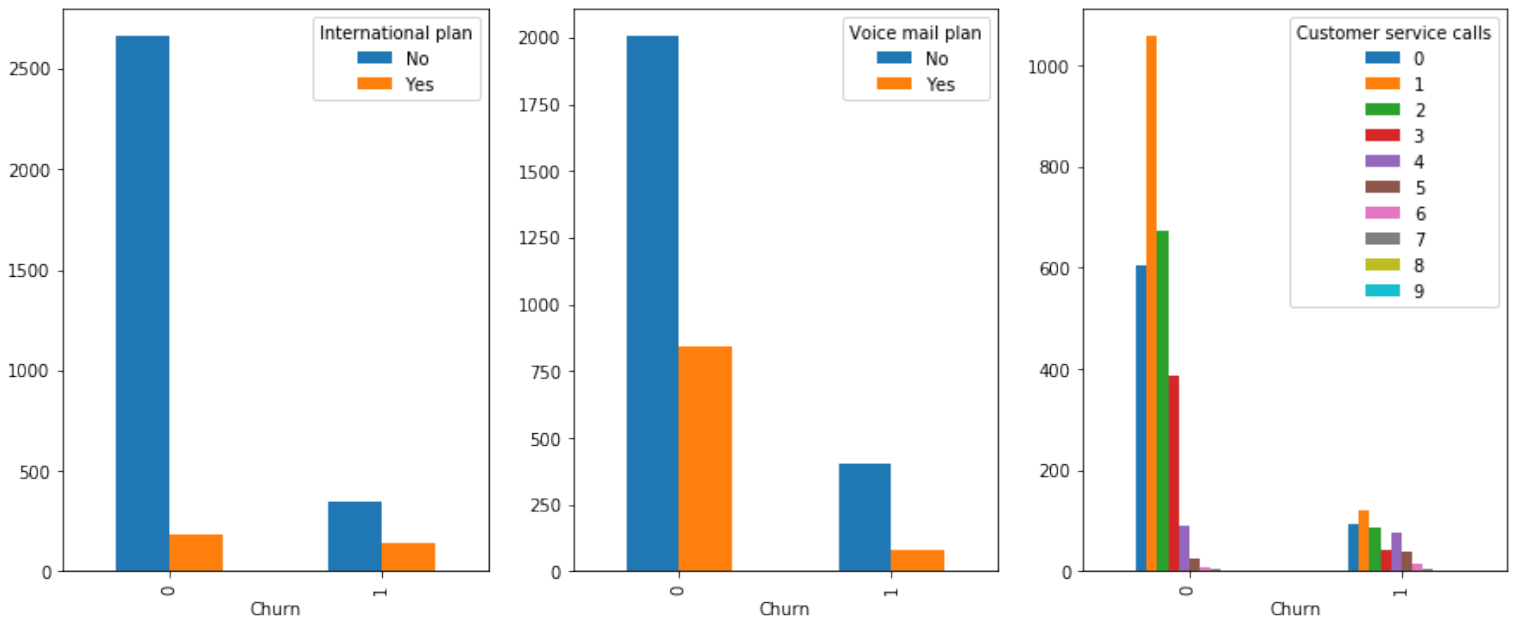
Churn	0	1
International plan		
No	2664	346
Yes	186	137

In [12]:

```
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(15, 6))
pd.crosstab( df.Churn, df['International plan']).plot(ax=axes[0], kind='bar')
pd.crosstab( df.Churn, df['Voice mail plan']).plot(ax=axes[1], kind='bar')
pd.crosstab( df.Churn, df['Customer service calls']).plot(ax=axes[2], kind='b
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a24ee8be0>
```



Exercise 8:

Add a new feature to the dataframe which will describe whether or not the user has done more than 3 service calls. Investigate how this feature describes the churn rate.

In [13]:

```
### Write Your code here ###

df.loc[df['Customer service calls'] > 3, 'more_than_3_service_calls'] = True
df.loc[df['Customer service calls'] <=3 ,'more_than_3_service_calls'] = False
```

In [ ]:

```

```

Learning Pandas and Matplotlib

Pandas is pythons library that enables broad possibilities for data analysis. By using Pandas it is very easy to upload, manage and analyse data from different tables by using SQL-like commands. Moreover, in connection with the libraries Matplotlib and Seaborn, Pandas gives broad opportunities to visualise the data.

In [1]:

We study the main methods of packages Pandas and Matplotlib by working with the dataset that describes the churn rate of the customers of a telecom company.

Exercise 1:

Read the data from the file 'telecom_churn.csv' and display the first 5 rows by using the method 'head'.
Hint : method read_csv may be useful

In [2]:

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	T c
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	

Exercise 2:

Display the size of the data array, information about it, and its main statistical characteristics. **Hint:** use methods shape, info, describe.

In [3]:

```
(3333, 20)
Two-dimensional size-mutable, potentially heterogeneous tabular data
structure with labeled axes (rows and columns). Arithmetic operation
s
align on both row and column labels. Can be thought of as a dict-lik
e
container for Series objects. The primary pandas data structure.

Parameters
----------
data : numpy ndarray (structured or homogeneous), dict, or DataFrame
    Dict can contain Series, arrays, constants, or list-like objects

    .. versionchanged :: 0.23.0
       If data is a dict, argument order is maintained for Python 3.
6
       and later.

index : Index or array-like
```

Exercise 3:

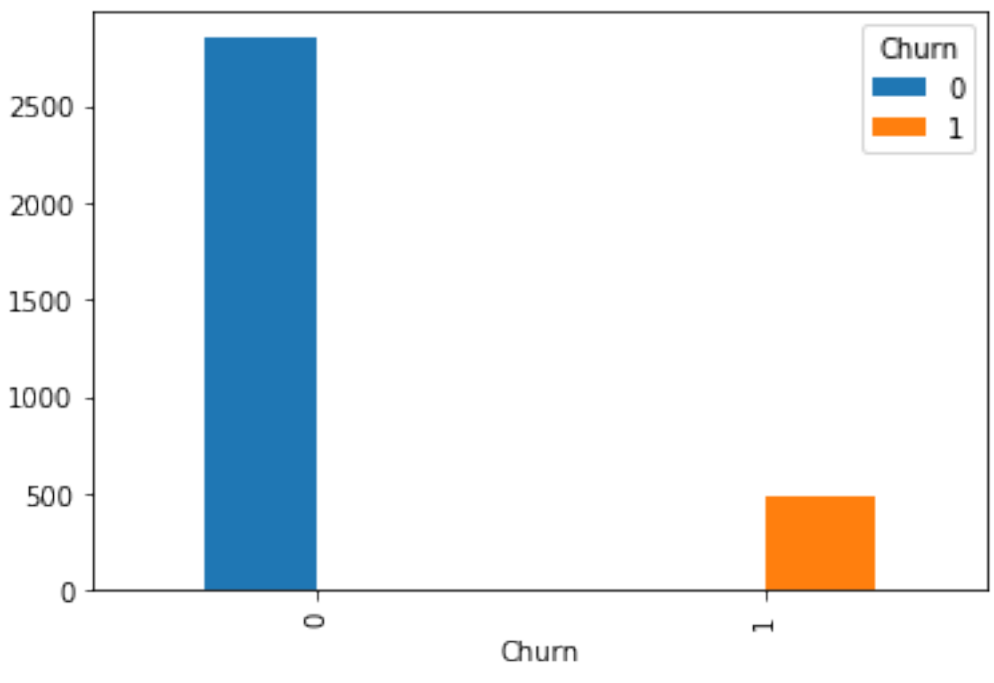
Convert column 'Churn' to the int64 type. **Hint:** use method astype. Plot the distribution of the churn and loyal clients in the bar plot. The figure should look like this:

In [4]:

In [13]:

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a25d7dc88>
```



Sorting

Exercise 4:

Sort the dataframe you have obtained in exercise 3 by the value in 'Total day charge' in descending/ascending order. Also sort it by using the column 'Churn' as the primary key and 'Total eve calls' as the secondary key. Try different combinations of ordering. **Hint:** use the method sort_values

In [6]:

Out[6]:

	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
365	CO	154	415	No	No	0	350.8	75	59.64	216.5
985	NY	64	415	Yes	No	0	346.8	55	58.96	249.5
2594	OH	115	510	Yes	No	0	345.3	81	58.70	203.4
156	OH	83	415	No	No	0	337.4	120	57.36	227.4
605	MO	112	415	No	No	0	335.5	77	57.04	212.5

In [7]:

Out[7]:

	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
3219	NY	150	415	No	Yes	35	139.6	72	23.73	332.8
58	WI	68	415	No	No	0	148.8	70	25.30	246.5
1020	HI	115	415	No	Yes	33	145.0	72	24.65	194.5
1706	MD	54	415	No	No	0	273.8	113	46.55	119.6
1071	PA	134	408	No	No	0	205.3	122	34.90	240.5

Indexing and extracting information from the dataframe

By using Pandas dataframes we are able to index and extract information from the dataset. You can index the information stored in dataframe either by names or by indices. In the first case you use the command loc, in the second iloc. **Hint:** Use logical indexing for the columns and the groupby method to solve the tasks.

Exercise 5:

- Display the mean churn rate of the clients.
- Now we want to analyse statistical information only for the clients, which are or aren't loyal to their telecom-company (field 'Churn' in dataframe). Extract the loyal and non-loyal clients from the table seperately and display the means of their charateristics in a single dataframe.
- How long do the non-loyal users talk during the day (on average)?
- What is the maximum length of the international calls for the loyal users that do not use the international plan?

In [ ]:

Distribution of the features

Exercise 6:

Plot the distribution of the features that have numerical values. **Hint:** use the method hist which can also be applied from the pandas dataframe. It should look like that:

What do you observe? From which probability distribution could each feature be generated?

In [9]:

Out[9]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x1a218da9e
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a21ac616
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a21bbc12
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a238e9c5
0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x1a238d5b7
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a236f90f
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a241982e
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a241d558
8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x1a241d55c
0>,
```

In [10]:

Pivot tables and graphics

We want to see how the instances are distributed between two categories: 'International plan' and 'Churn'.

Exercise 7:

- Build the cross table between the features using the method *crosstab*.
- Visualize the distribution for the feature 'Churn', depending on the value of the features 'International plan', 'Voice mail plan', and 'Customer service calls'. **Hint:** Use commands plt.sublot and sns.subplots.

Your plot should look something like this:

What do you see? What conclusions can be drawn? What feature (intuitively) can be more important for Churn prediction?

In [11]:

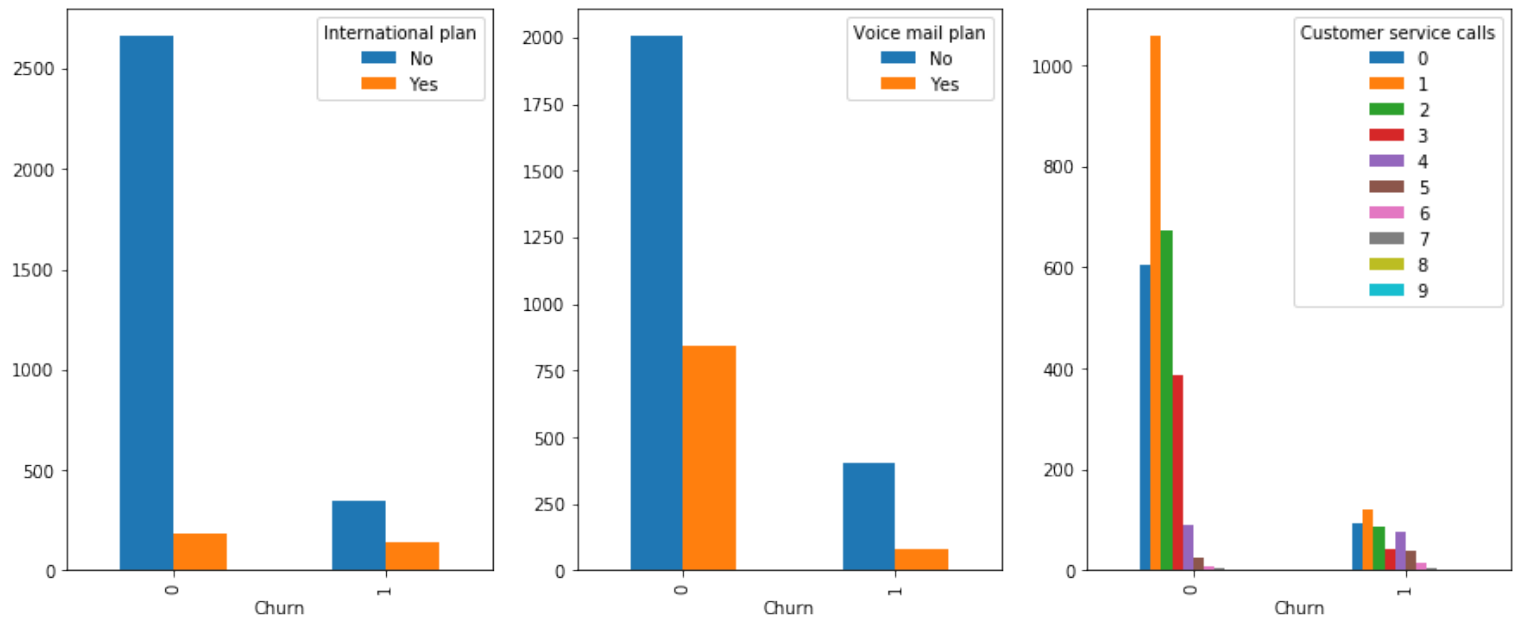
Out[11]:

Churn	0	1
International plan		
No	2664	346
Yes	186	137

In [12]:

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a24ee8be0>
```



Exercise 8:

Add a new feature to the dataframe which will describe whether or not the user has done more than 3 service calls. Investigate how this feature describes the churn rate.

In [13]:

In [ ]: