Data Mining - Lab 4

Huỳnh Thị Thắm - 18110209

In [1]:
```
# import basic libraries
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
# import plot libraries
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
path='Dataset/telecom_churn.csv'
df_customer=pd.read_csv(path)
df_customer.head(10)
```

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	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls	c
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	
5	AL	118	510	Yes	No	0	223.4	98	37.98	220.6	101	18.75	203.9	118	9.18	6.3	6	
6	MA	121	510	No	Yes	24	218.2	88	37.09	348.5	108	29.62	212.6	118	9.57	7.5	7	
7	MO	147	415	Yes	No	0	157.0	79	26.69	103.1	94	8.76	211.8	96	9.53	7.1	6	
8	LA	117	408	No	No	0	184.5	97	31.37	351.6	80	29.89	215.8	90	9.71	8.7	4	
9	WV	141	415	Yes	Yes	37	258.6	84	43.96	222.0	111	18.87	326.4	97	14.69	11.2	5	

In [4]:
```
path='Dataset/BigMartSales.csv'
df_mart=pd.read_csv(path)
df_mart.head(10)
```

Out[4]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Out
0	FDA15	9.300	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	
1	DRC01	5.920	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	
2	FDN15	17.500	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	
3	FDX07	19.200	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	
4	NCD19	8.930	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	
5	FDP36	10.395	Regular	0.000000	Baking Goods	51.4008	OUT018	2009	Medium	
6	FDO10	13.650	Regular	0.012741	Snack Foods	57.6588	OUT013	1987	High	
7	FDP10	NaN	Low Fat	0.127470	Snack Foods	107.7622	OUT027	1985	Medium	
8	FDH17	16.200	Regular	0.016687	Frozen Foods	96.9726	OUT045	2002	NaN	
9	FDU28	19.200	Regular	0.094450	Frozen Foods	187.8214	OUT017	2007	NaN	

In [5]:
```
#Consider dataset BigMart Sales
print('Columns s name of BigMart Sales dataset: \n',df_mart.columns)
print('Columns s name of Customer Churn dataset: \n',df_customer.columns)
print('Shape of BigMart Sales dataset before drop null values: ',df_mart.shape)
print('Shape of Customer Churn dataset before drop null values: ',df_customer.shape)
df_mart=df_mart.dropna()
df_customer=df_customer.dropna()
print('Shape of BigMart Sales dataset after drop null values: ',df_mart.shape)
print('Shape of Customer Churn dataset afterr drop null values: ',df_customer.shape)
```

```
Columns s name of BigMart Sales dataset:
 Index(['Item_Identifier', 'Item_Weight', 'Item_Fat_Content', 'Item_Visibility',
       'Item_Type', 'Item_MRP', 'Outlet_Identifier',
       'Outlet_Establishment_Year', 'Outlet_Size', 'Outlet_Location_Type',
       'Outlet_Type', 'Item_Outlet_Sales'],
      dtype='object')
Columns s name of Customer Churn dataset:
 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')
Shape of BigMart Sales dataset before drop null values:  (8523, 12)
Shape of Customer Churn dataset before drop null values:  (3333, 20)
Shape of BigMart Sales dataset after drop null values:  (4650, 12)
Shape of Customer Churn dataset afterr drop null values:  (3333, 20)
```

Với mỗi tiêu chí/ thuộc tính của dữ liệu CustomerChurn hay BigMartSales chọn một hình vẽ EDA phù hợp kèm theo nhận xét của bạn về tiêu chí/thuộc tính đó:

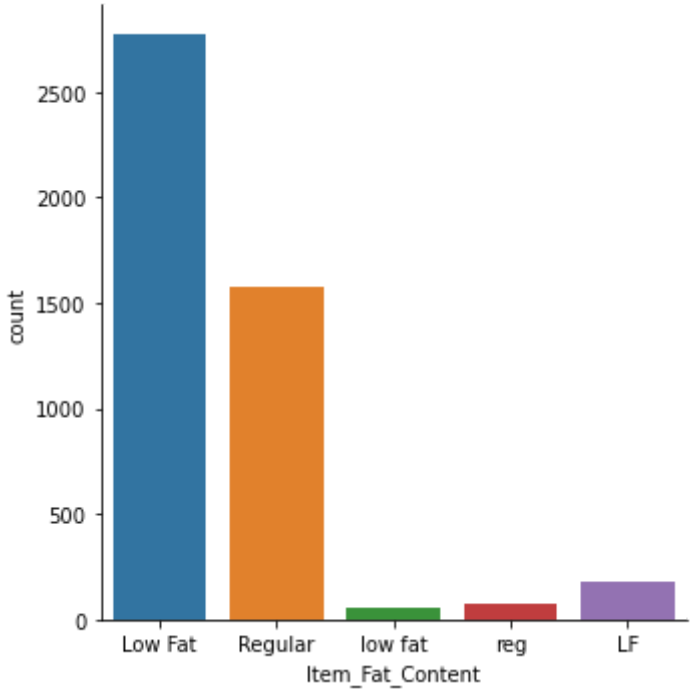
BigMart Sales

In [6]:
```
print('Type of each features of BigMart Sales: \n',df_mart.dtypes)
```

```
Type of each features of BigMart Sales:
 Item_Identifier               object
Item_Weight                   float64
Item_Fat_Content               object
Item_Visibility               float64
Item_Type                      object
Item_MRP                      float64
Outlet_Identifier              object
Outlet_Establishment_Year       int64
Outlet_Size                    object
Outlet_Location_Type           object
Outlet_Type                    object
Item_Outlet_Sales             float64
dtype: object
```

In [7]:
```
sns.catplot(x='Item_Fat_Content',kind='count',data=df_mart)
```

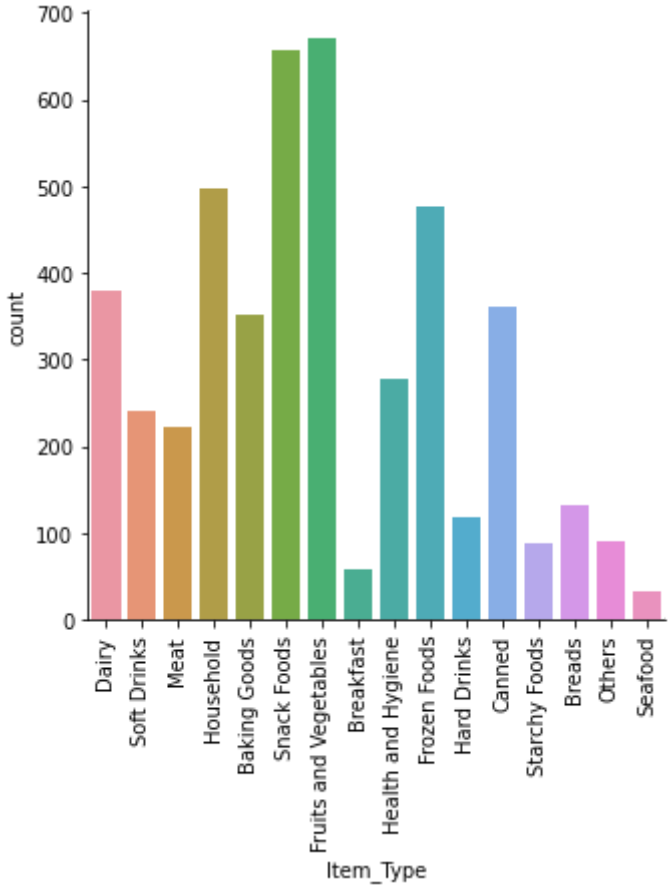
Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c0a40b20>
```



Ta thấy, Item fat content có số lượng Low Fat là chiếm nhiều nhất và nhỏ nhất là low fat.

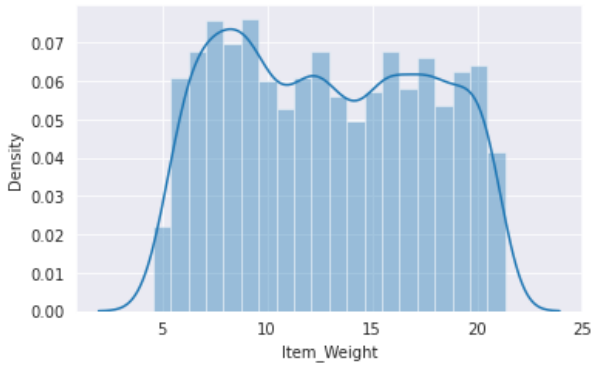
In [8]:
```
sns.catplot(x='Item_Type',kind='count',data=df_mart).set_xticklabels(rotation=90)
```

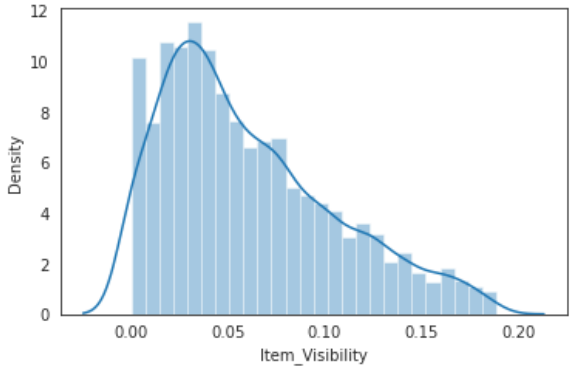
Out[8]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c64c7fa0>
```

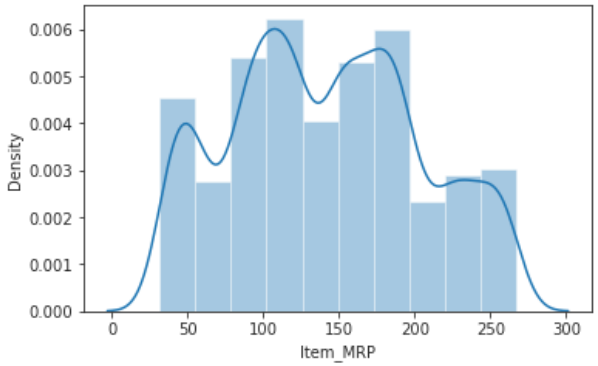


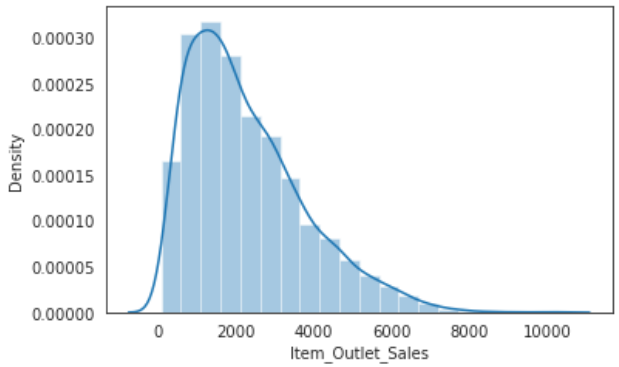
Item type chiếm số lượng lớn nhất là Snack Foods và Fruits and Vegetables

In [14]:
```
f = plt.figure(figsize=(20,8))
gs = f.add_gridspec(2, 3)

with sns.axes_style("darkgrid"):
    ax = f.add_subplot(gs[0, 0])
    sns.distplot(df_mart.Item_Weight,bins=20)

with sns.axes_style("white"):
    ax = f.add_subplot(gs[0, 1])
    sns.distplot(df_mart.Item_Visibility)

with sns.axes_style("ticks"):
    ax = f.add_subplot(gs[0, 2])
    sns.distplot(df_mart.Item_MRP,bins=10)

with sns.axes_style("white"):
    ax = f.add_subplot(gs[1, 0])
    sns.distplot(df_mart.Item_Outlet_Sales,bins=20)
```





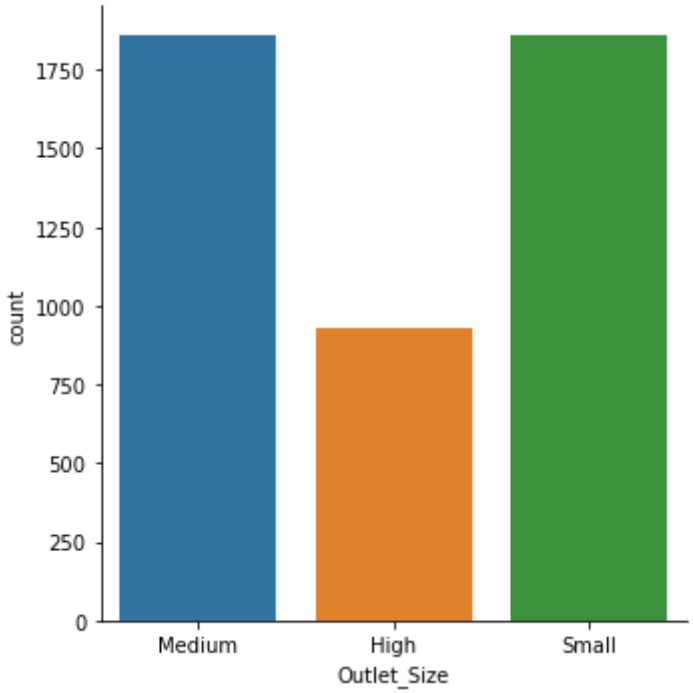


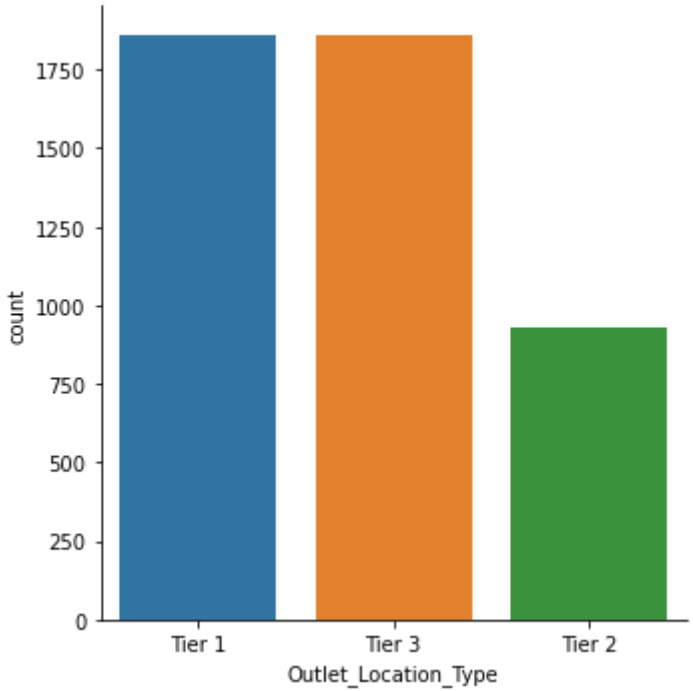


Theo distplot thì Item_weight trông không tuân theo phân phối chuẩn và tập trung chủ yếu là mức từ 5 tới 10. Ta thấy item_visibility tuân theo phân phối chuẩn nhưng hơi lệch phải và tập trung chủ yếu ở mức từ 0.03 tới 0.05 Item_MRP tập trung nhiều ở 75-100 và từ 150 - 200 Item_Outlet_sales tuân theo phân phối chuẩn và hơi lệch phải với đỉnh ở 1500 - 2000

In [15]:
```
sns.catplot(x='Outlet_Size',kind='count',data=df_mart)
sns.catplot(x='Outlet_Location_Type',kind='count',data=df_mart)
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x1d7c67e1610>

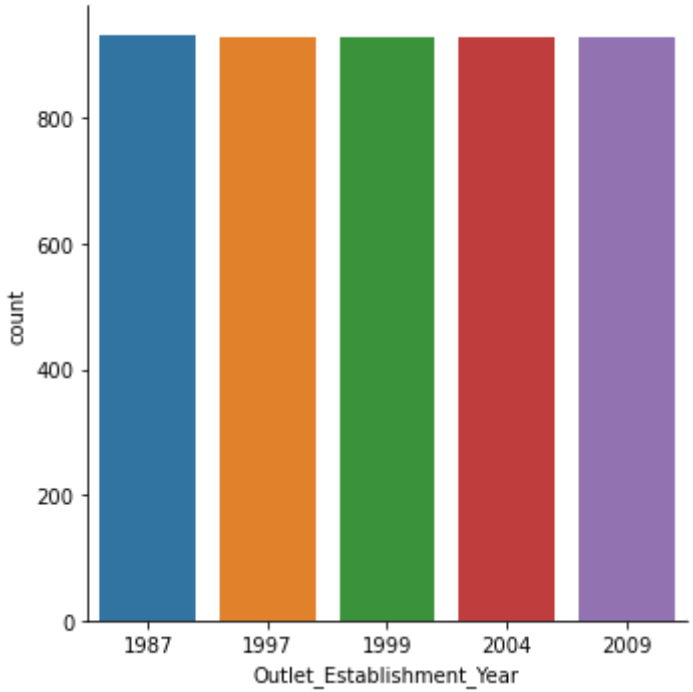




Số lượng outlet_size của Medium và Small khá đều nhau và tương tự cho Tier 1, tier 2 cho outlet_location_type

In [16]:
```
#Outlet_Establishment_Year
sns.catplot(x='Outlet_Establishment_Year',kind='count',data=df_mart)
```

Out[16]: <seaborn.axisgrid.FacetGrid at 0x1d7c6b3b970>



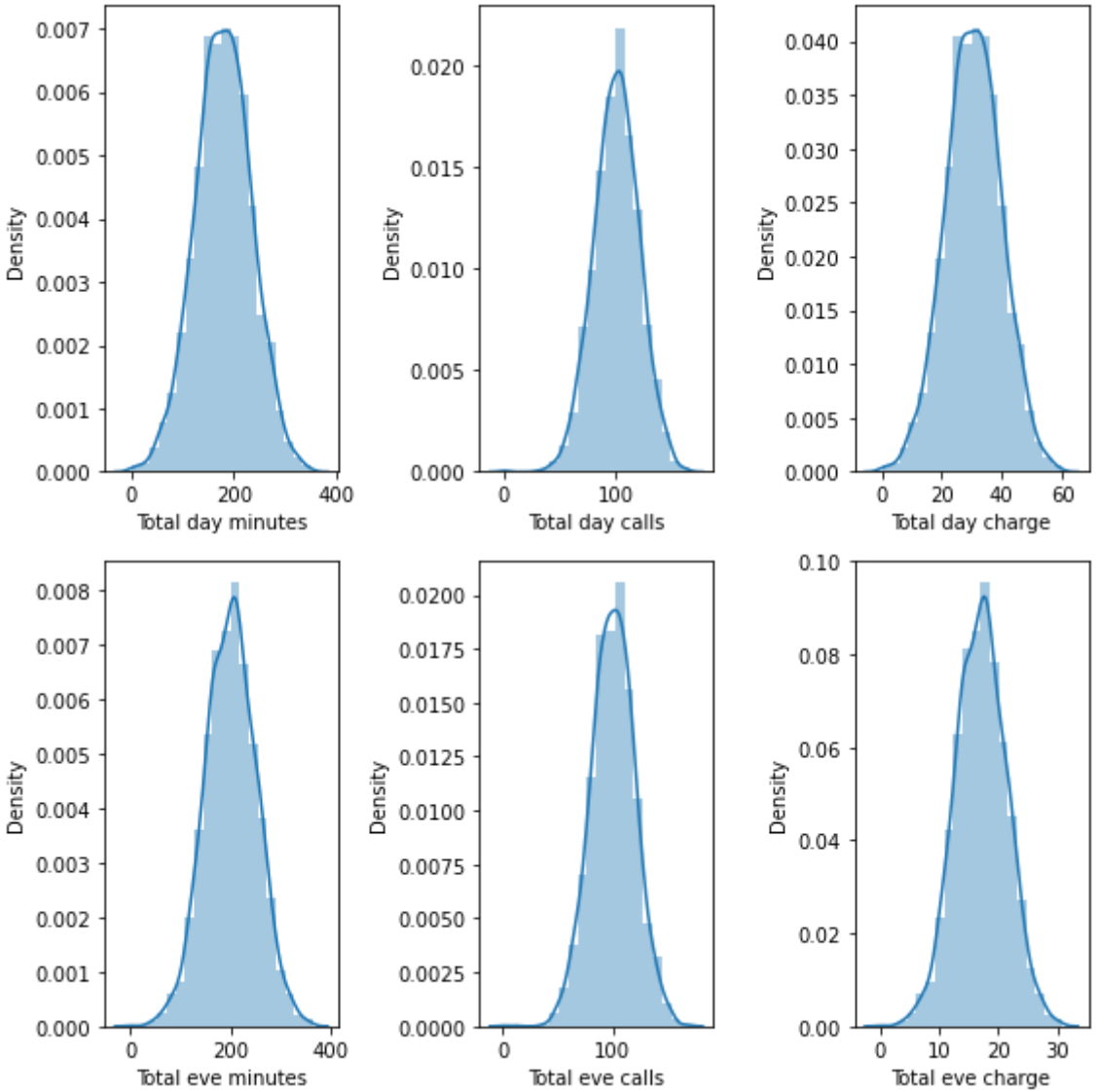
Năm establish outlet là đều nhau với các năm là 1987, 1997, 1999, 2004, 2009

Customer Churn

In [17]:
```
print('Type of each features of Customer Churn: \n',df_customer.dtypes)
```

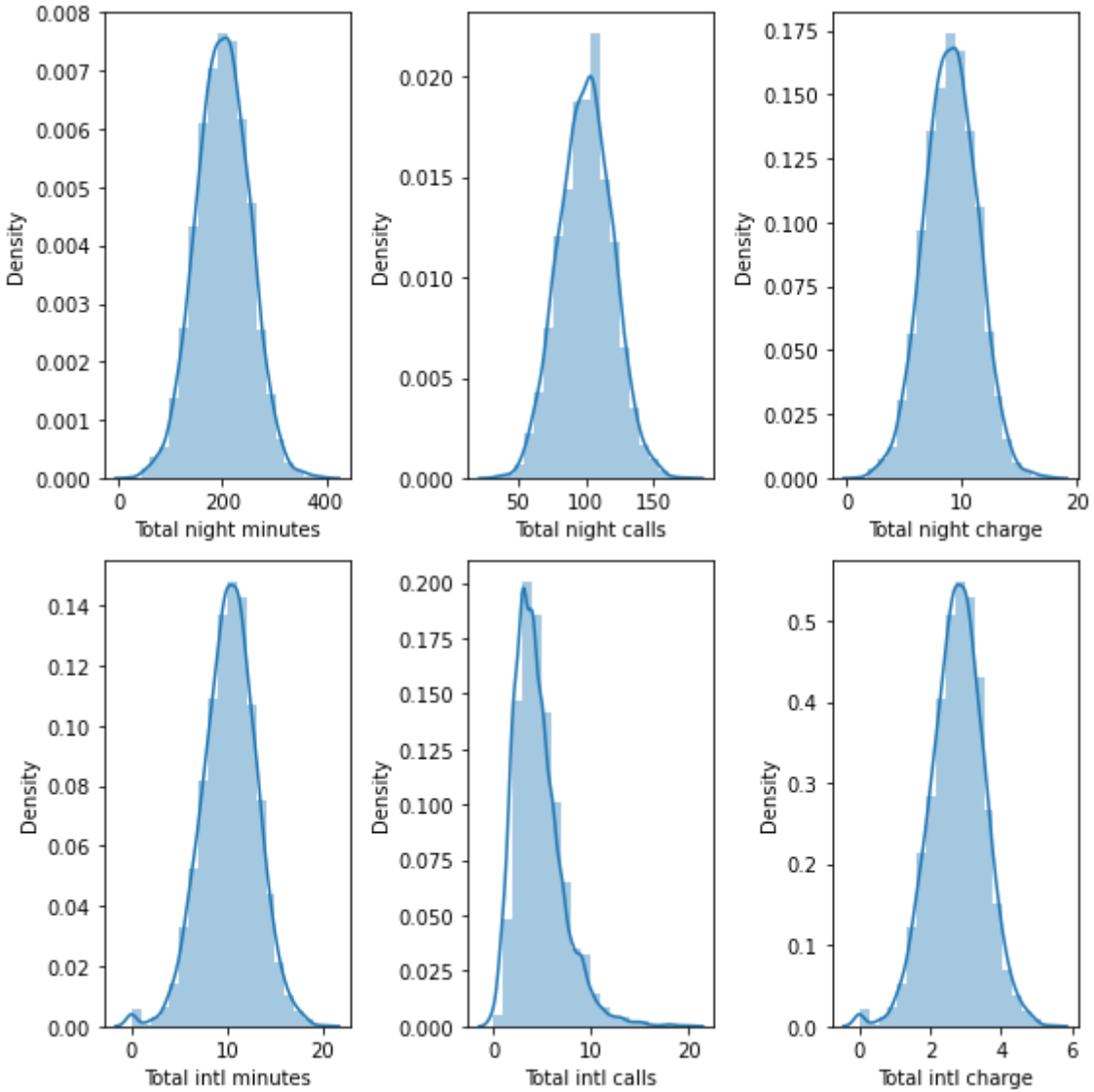
```
Type of each features of Customer Churn:
 State                    object
Account length            int64
Area code                 int64
International plan       object
Voice mail plan          object
Number vmail messages     int64
Total day minutes       float64
Total day calls           int64
Total day charge        float64
Total eve minutes       float64
Total eve calls           int64
Total eve charge        float64
Total night minutes     float64
Total night calls         int64
Total night charge      float64
Total intl minutes      float64
Total intl calls          int64
Total intl charge       float64
Customer service calls    int64
Churn                      bool
dtype: object
```

In [18]:
```
f = plt.figure(figsize=(8,8))
gs = f.add_gridspec(2, 3)
ax = f.add_subplot(gs[0, 0])
sns.distplot(df_customer['Total day minutes'],bins=20)
ax = f.add_subplot(gs[0, 1])
sns.distplot(df_customer['Total day calls'],bins=20)
ax = f.add_subplot(gs[0, 2])
sns.distplot(df_customer['Total day charge'],bins=20)
ax = f.add_subplot(gs[1, 0])
sns.distplot(df_customer['Total eve minutes'],bins=20)
ax = f.add_subplot(gs[1, 1])
sns.distplot(df_customer['Total eve calls'],bins=20)
ax = f.add_subplot(gs[1, 2])
sns.distplot(df_customer['Total eve charge'],bins=20)
f.tight_layout()
```



Tất cả các plot đều tuân theo dạng chuẩn trong đó đỉnh nằm ở giữa đồ thị

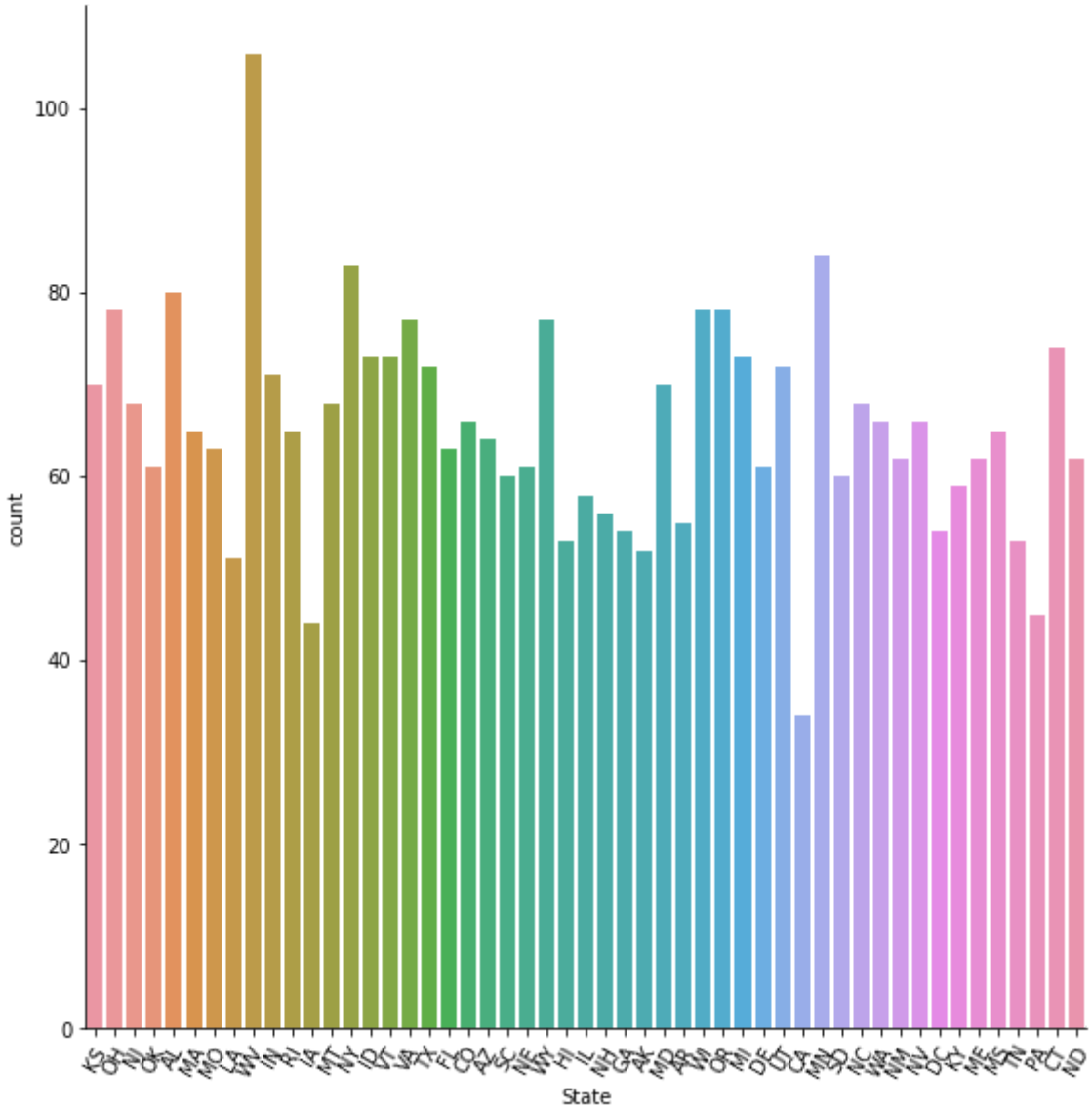
In [19]:
```
f = plt.figure(figsize=(8,8))
gs = f.add_gridspec(2, 3)
ax = f.add_subplot(gs[0, 0])
sns.distplot(df_customer['Total night minutes'],bins=20)
ax = f.add_subplot(gs[0, 1])
sns.distplot(df_customer['Total night calls'],bins=20)
ax = f.add_subplot(gs[0, 2])
sns.distplot(df_customer['Total night charge'],bins=20)
ax = f.add_subplot(gs[1, 0])
sns.distplot(df_customer['Total intl minutes'],bins=20)
ax = f.add_subplot(gs[1, 1])
sns.distplot(df_customer['Total intl calls'],bins=20)
ax = f.add_subplot(gs[1, 2])
sns.distplot(df_customer['Total intl charge'],bins=20)
f.tight_layout()
```



Tất cả đều dạng chuẩn và có đỉnh ở giữa đồ thị. Trừ total_intl call bị lệch phải

In [20]:
```
sns.catplot(x='State',kind='count',data=df_customer,height=8).set_xticklabels(rotation=60)
```

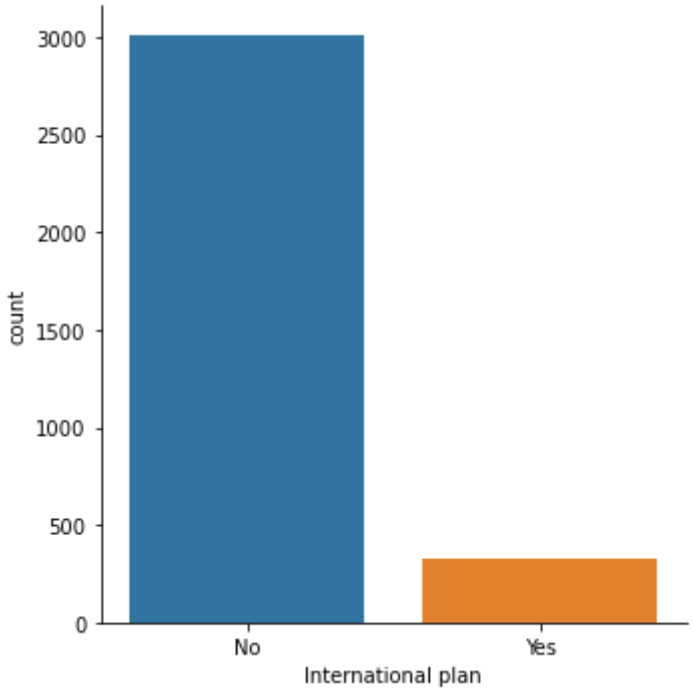
Out[20]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c6a971c0>
```



WV là state có số lần xuất hiện nhiều nhất

In [21]:
```
sns.catplot(x='International plan',kind='count',data=df_customer)
```

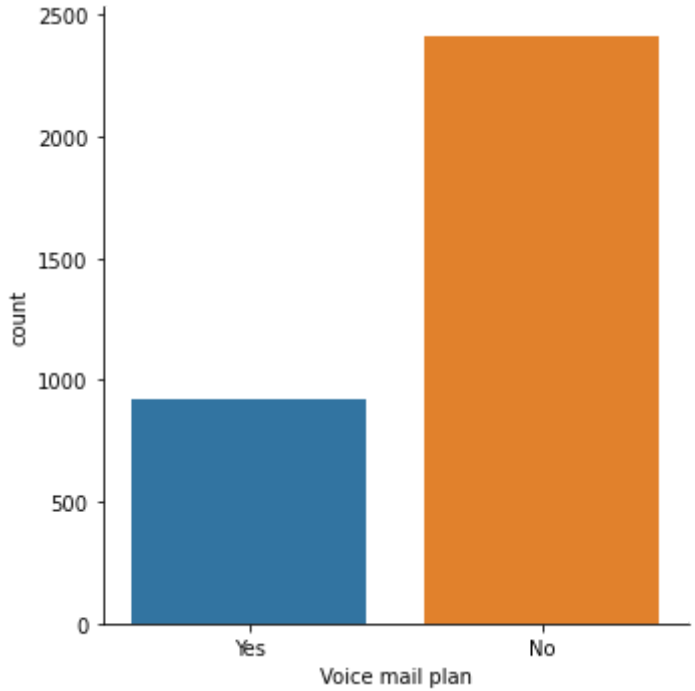
Out[21]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c6baadc0>
```



Đa số có international plan là No với khoảng 3000

In [22]:
```
sns.catplot(x='Voice mail plan',kind='count',data=df_customer)
```

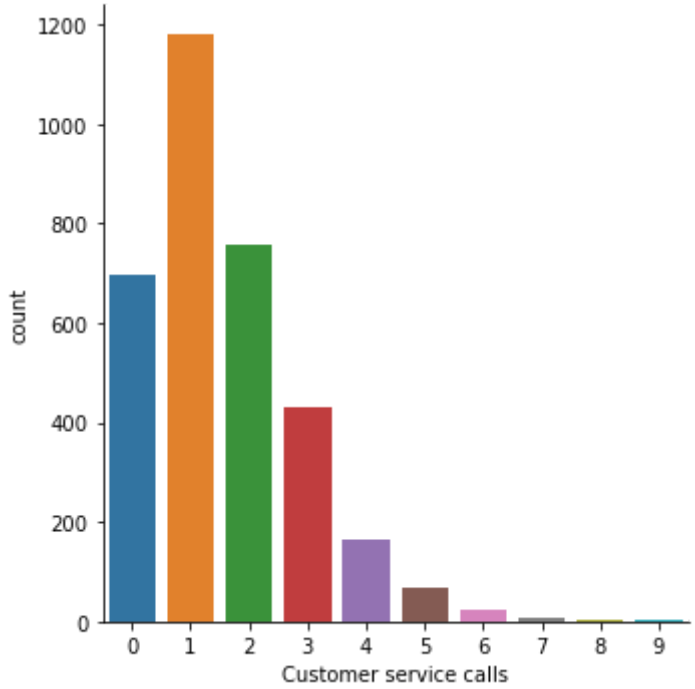
Out[22]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c7dc5130>
```



Đa số là No với số lượng gần 2500

In [23]:
```
sns.catplot(x='Customer service calls',kind='count',data=df_customer)
```

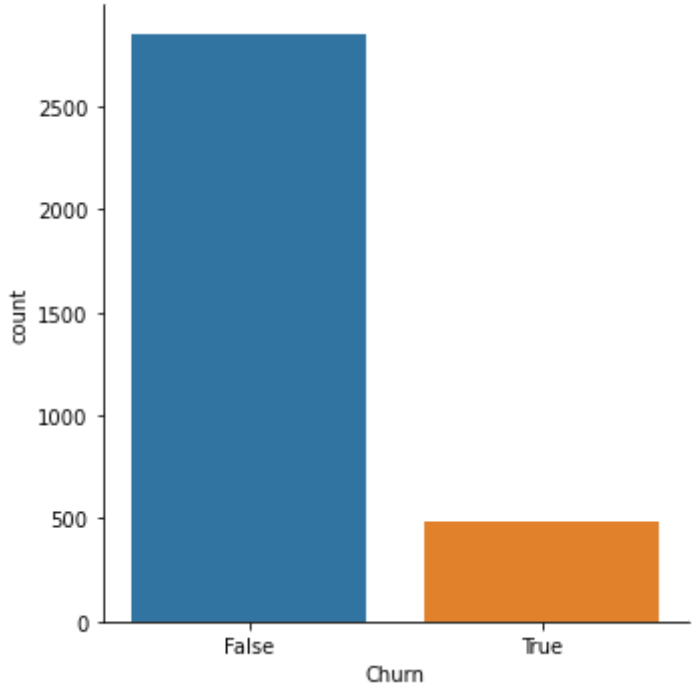
Out[23]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c675c2e0>
```



Số lượng trông giống phân phối chuẩn và có sốlượng nhiều nhất ở 1 với khoảng 1200

In [24]:
```
sns.catplot(x='Churn',kind='count',data=df_customer)
```

Out[24]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c6977580>
```



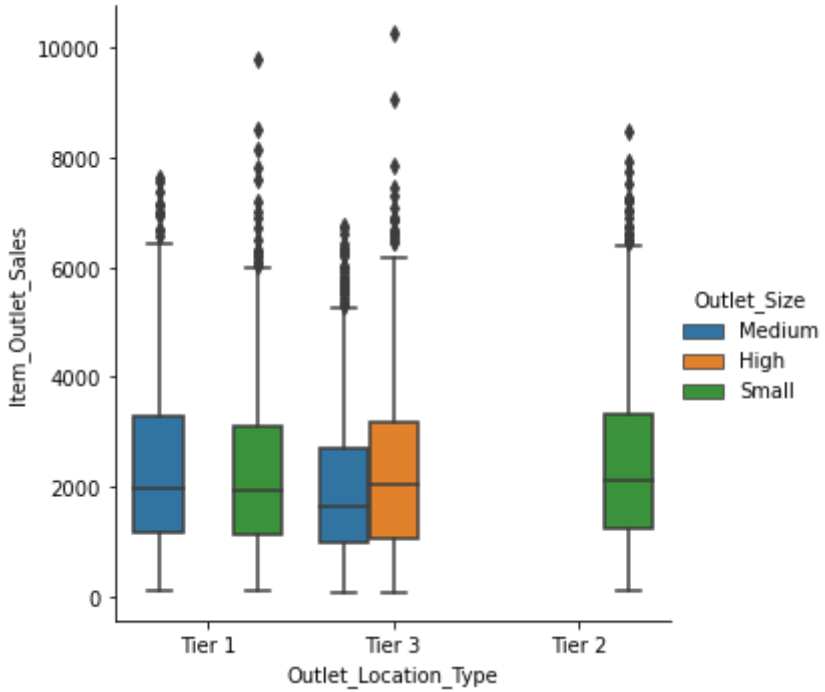
False chiếm đa số với hơn 2500

Chọn 2, 3, hay 4 tiêu chí bạn nghi ngờ có mối quan hệ với nhau mật thiết và biểu diễn chúng lên một hình EDA sau đó cho nhận xét về mối quan hệ (Mỗi dữ liệu CustomerChurn hay BigMartSales cho 3 TH này)

BigMartSales

In [25]:
```
sns.catplot(x='Outlet_Location_Type',y='Item_Outlet_Sales',hue='Outlet_Size',kind='box',data=df_mart)
```

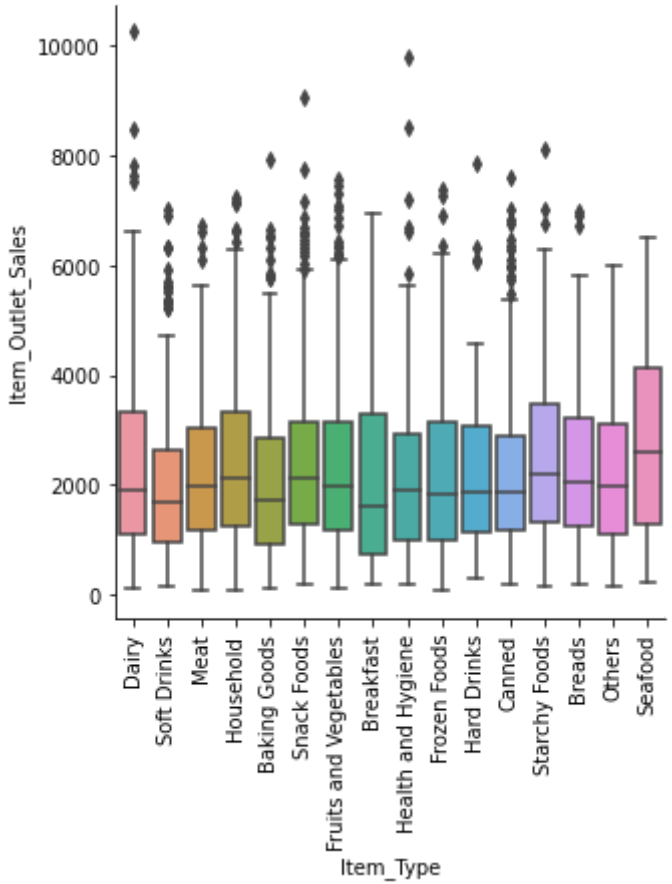
Out[25]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c7d644c0>
```



Có vẽ như Tier 2 chỉ có Outlet_Size ở dạng Small, Tier 1 không có Outlet có kích cỡ lớn và tier 3 không có outlet cỡ n

In [26]:
```
sns.catplot(x='Item_Type',y='Item_Outlet_Sales',kind='box',data=df_mart).set_xticklabels(rotation=90)
```

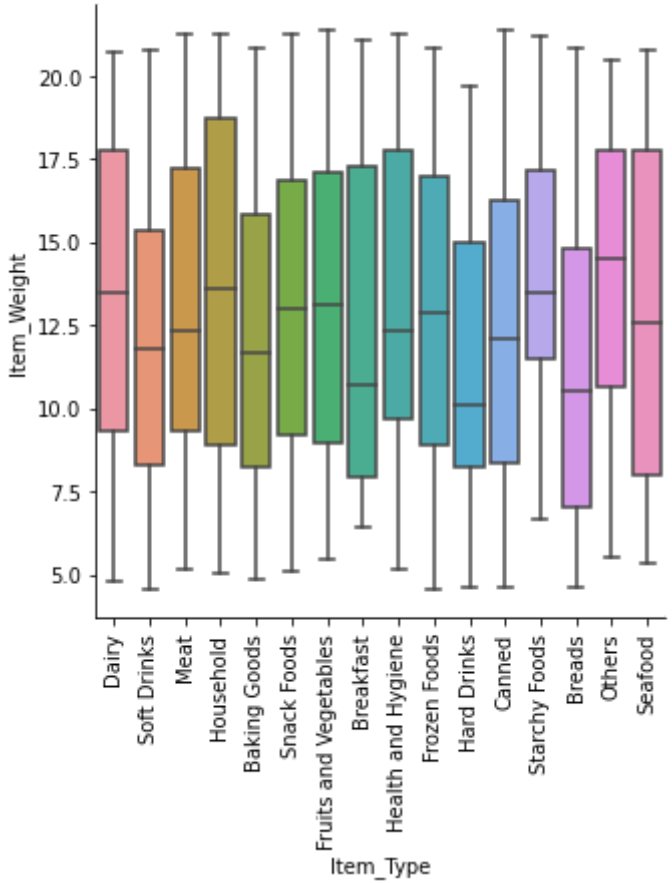
Out[26]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c80227f0>
```



Có vẻ như Hard Drink có lượng Item_Outlet_Sales thấp hơn cả. Trong khi đó với Item_type là Seafood thì người ta chi tiêu nhiều tiền hơn cho Item_Outlet

In [27]:
```
sns.catplot(x='Item_Type',y='Item_Weight',kind='box',data=df_mart).set_xticklabels(rotation=90)
```

Out[27]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c65d6130>
```

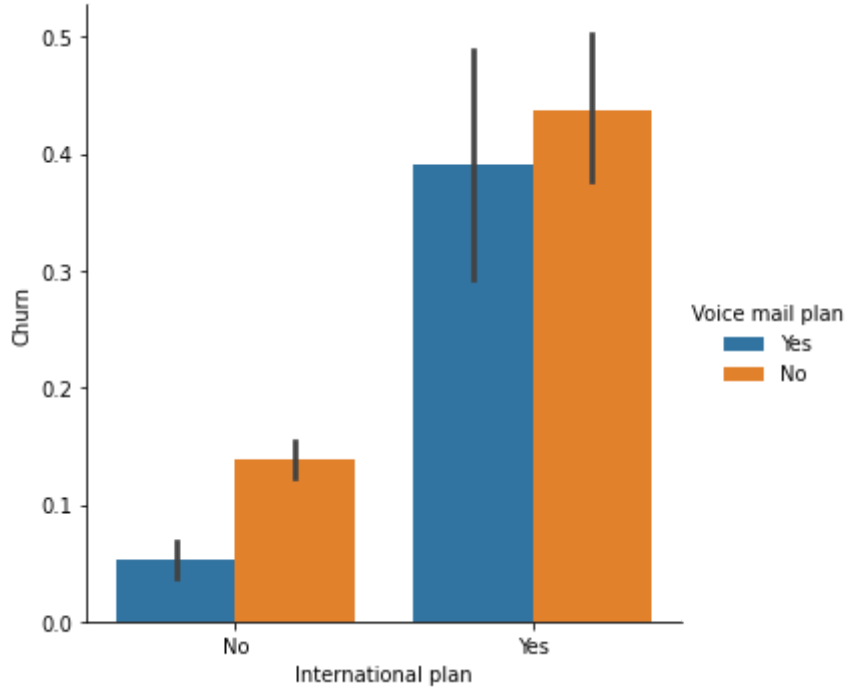


Những khách hàng mua household, Dairy, Health and Hygiene, Seafood và Others có xu hướng có cân nặng Item cao hơn những hàng mua Item type khác.

Customer Churn

In [28]:
```
sns.catplot(data=df_customer,x='International plan',y='Churn',hue='Voice mail plan',kind='bar')
```

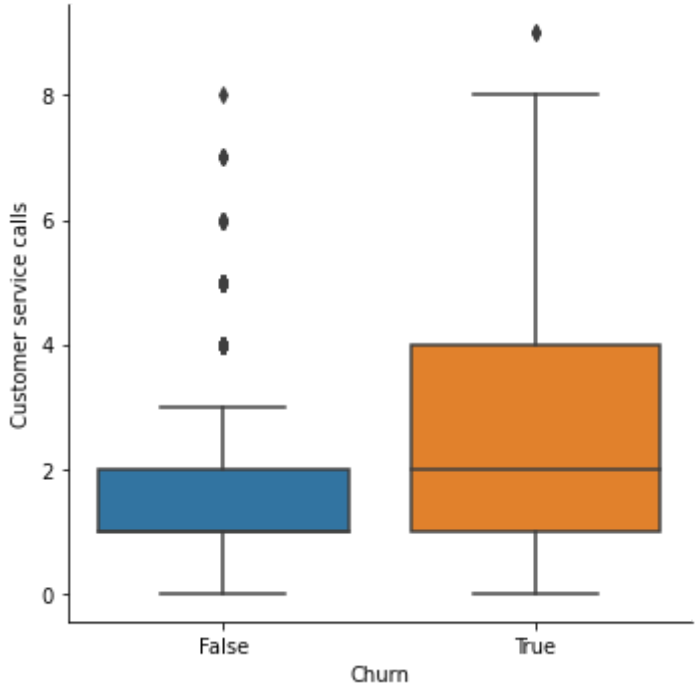
Out[28]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c7f1f160>
```



Với international plan No thì số lượng Voice mail plan với Churn đều thấp. Trong khi đó, tí lệ Voice mail plan và Chu Yes hay no thì đều đa số nằm ở International plan là Yes.

In [29]:
```
sns.catplot(data=df_customer,x='Churn',y='Customer service calls',kind='box')
```

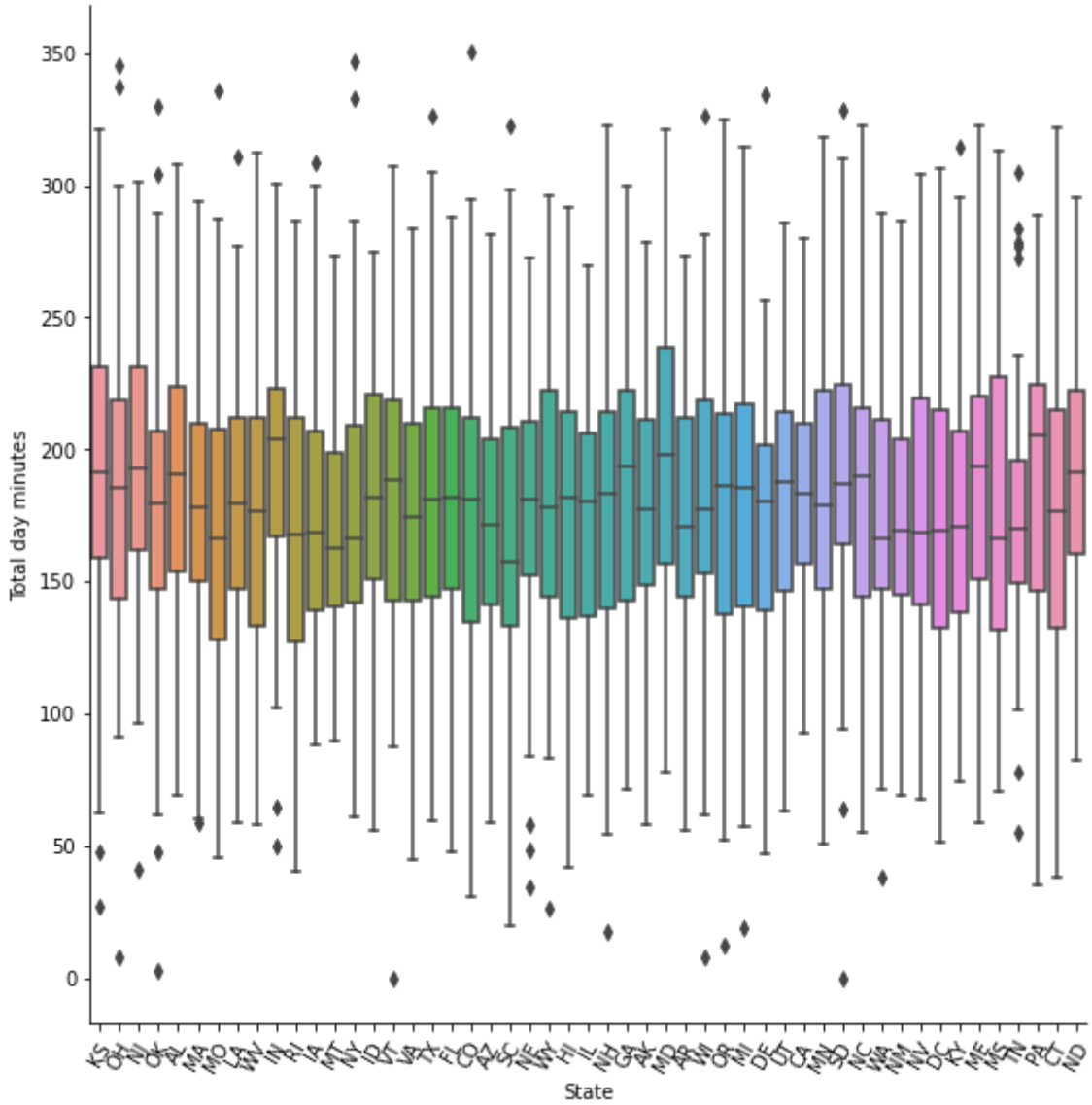
Out[29]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c694cd00>
```



Với những khách hàng có Churn là True thì trung bình số cuộc gọi dịch vụ mà họ thực hiện Nhiều hơn rất nhiều so với khách hàng không Churn.

In [30]:
```
sns.catplot(data=df_customer,x='State',y='Total day minutes',kind='box',height=8).set_xticklabels(rotation=60)
```

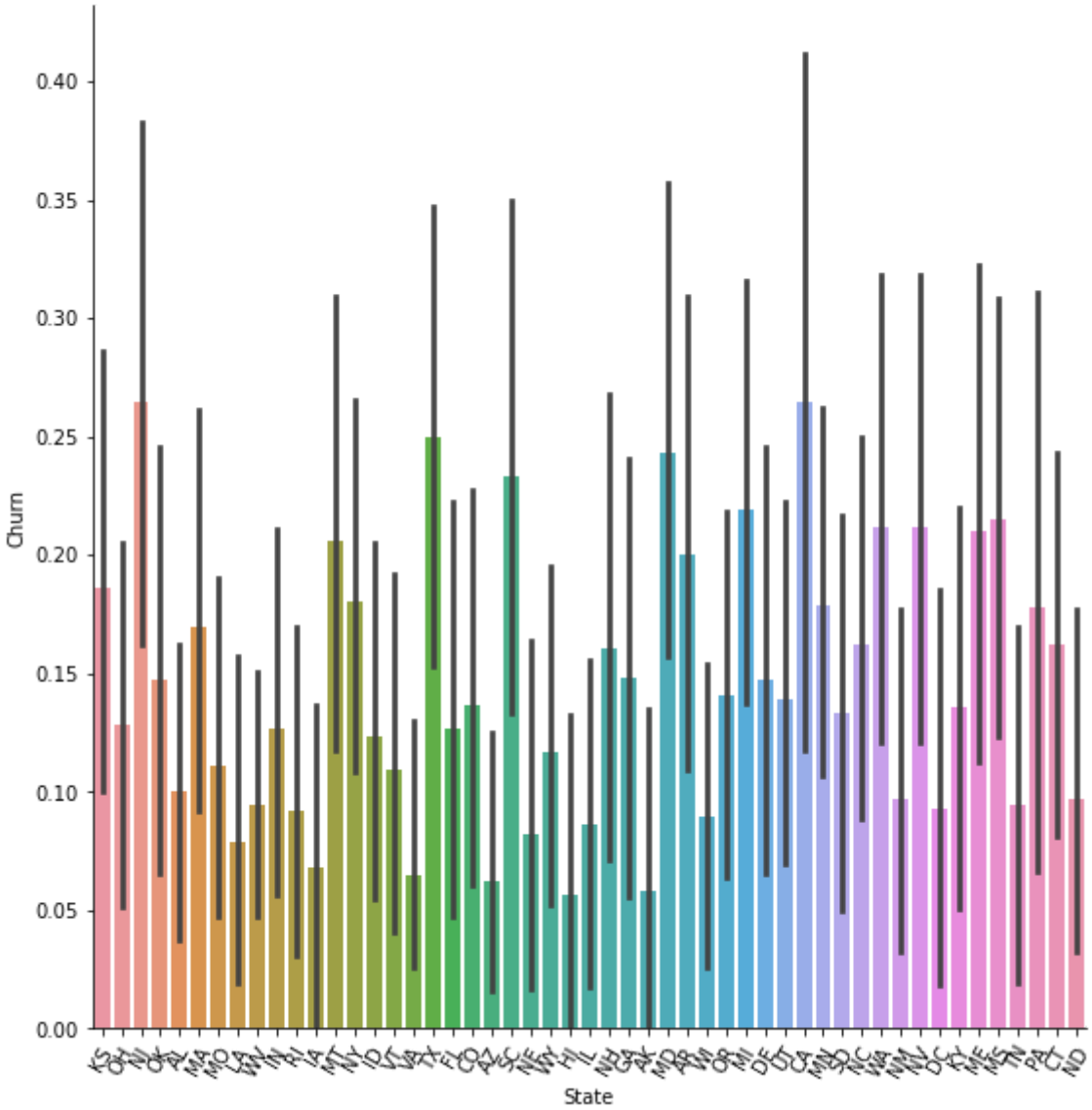
Out[30]: <seaborn.axisgrid.FacetGrid at 0x1d7c69ccf70>



Các bang IN, PA, MD, ME có tổng số phút gọi ban ngày trung bình là cao hơn các bang còn lại.

In [31]:
```
sns.catplot(data=df_customer,x='State',y='Churn',kind='bar',height=8).set_xticklabels(rotation=60)
```

Out[31]:
```
<seaborn.axisgrid.FacetGrid at 0x1d7c822b640>
```



Bang NJ, TX, CA, MD là các bang có tỉ lệ Churn cao nhất.