

```
1 # import basic libraries
2 import numpy as np
3 import pandas as pd
4 import warnings
5 warnings.filterwarnings('ignore')
6 # import plot libraries
7 import seaborn as sns
8 import matplotlib.pyplot as plt
```

```
1 path='/content/CustomerChurn.csv'
2 df_customer=pd.read_csv(path)
3 df_customer.head(10)
```

	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
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	110	10.30	162.6	104
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9	88	5.26	196.9	89
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3	122	12.61	186.9	121
5	AL	118	510	Yes	No	0	223.4	98	37.98	220.6	101	18.75	203.9	118
6	MA	121	510	No	Yes	24	218.2	88	37.09	348.5	108	29.62	212.6	118
7	MO	147	415	Yes	No	0	157.0	79	26.69	103.1	94	8.76	211.8	96
8	LA	117	408	No	No	0	184.5	97	31.37	351.6	80	29.89	215.8	90
9	WV	141	415	Yes	Yes	37	258.6	84	43.96	222.0	111	18.87	326.4	97

```
1 path='/content/BigMartSales.csv'
2 df_mart=pd.read_csv(path)
3 df_mart.head(10)
```

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Estimated_Sales
0	FDA15	9.300	Low Fat	0.016047	Dairy	249.8092	OUT049	1363.2682
1	DRC01	5.920	Regular	0.019278	Soft Drinks	48.2692	OUT018	191.9362
2	FDN15	17.500	Low Fat	0.016760	Meat	141.6180	OUT049	1038.5495
3	FDX07	19.200	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1495.0170
4	NCD19	8.930	Low Fat	0.000000	Household	53.8614	OUT013	291.2790
5	FDP36	10.395	Regular	0.000000	Baking Goods	51.4008	OUT018	205.2940
6	FDO10	13.650	Regular	0.012741	Snack Foods	57.6588	OUT013	312.3446
7	FDP10	NaN	Low Fat	0.127470	Snack Foods	107.7622	OUT027	1184.2060
8	FDH17	16.200	Regular	0.016687	Frozen Foods	96.9726	OUT045	554.5092
9	FDU28	19.200	Regular	0.094450	Frozen Foods	187.8214	OUT017	1504.2892

```
1 #Consider dataset BigMart Sales
2 print('Columns s name of BigMart Sales dataset: \n',df_mart.columns)
3 print('Columns s name of Customer Churn dataset: \n',df_customer.columns)
4 print('Shape of BigMart Sales dataset before drop null values: ',df_mart.shape)
5 print('Shape of Customer Churn dataset before drop null values: ',df_customer.shape)
6
7 df_mart=df_mart.dropna()
8 df_customer=df_customer.dropna()
9
```

```

9
10 print('Shape of BigMart Sales dataset after drop null values: ',df_mart.shape)
11 print('Shape of Customer Churn dataset after 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 after 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

```
1 print('Type of each features of BigMart Sales: \n',df_mart.dtypes)
```

Type of each features of BigMart Sales:

```

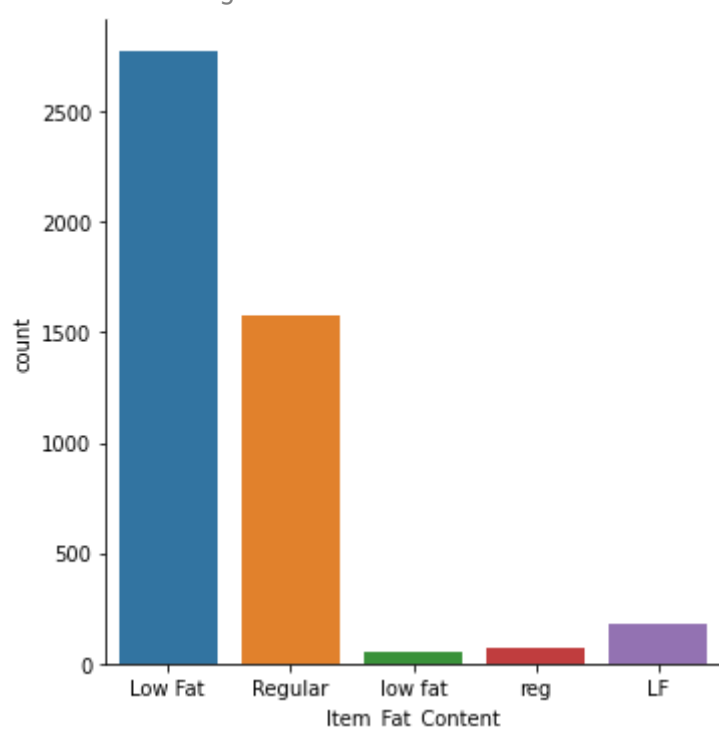
Item_Identifier      object
Item_Weight          float64
Item_Fat_Content      object
Item_Visibility      float64
Item_Type            object
Item_MRP             float64
Outlet_Identifier     object
Outlet_Establishment_Year  int64
Outlet_Size          object
Outlet_Location_Type  object
Outlet_Type          object
Item_Outlet_Sales    float64
dtype: object

```

```
1 sns.catplot(x='Item_Fat_Content',kind='count',data=df_mart)
```

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

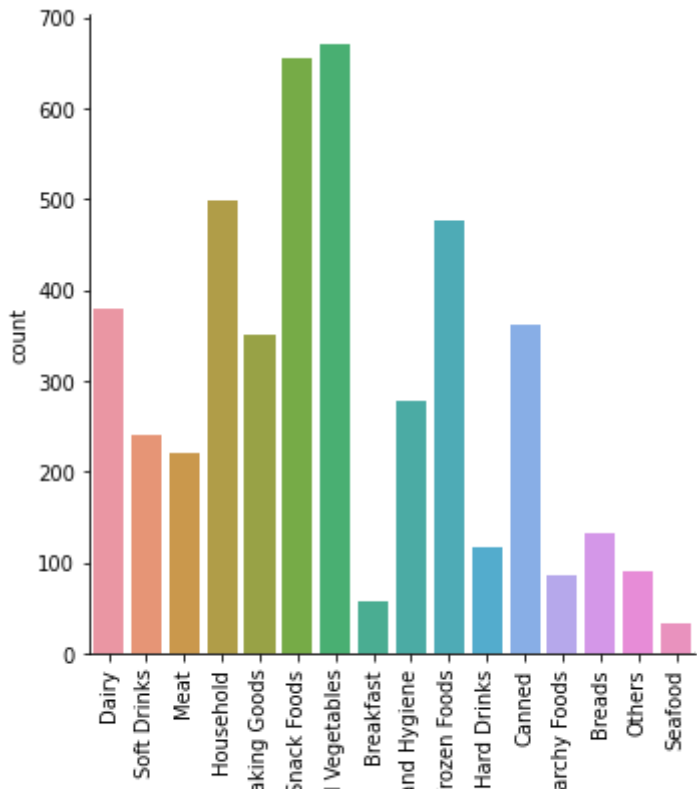
<seaborn.axisgrid.FacetGrid at 0x7fac7e1a1550>



```
1 sns.catplot(x='Item_Type',kind='count',data=df_mart).set_xticklabels(rotation=90)
```

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

<seaborn.axisgrid.FacetGrid at 0x7fac7e12a9d0>



```
1 f = plt.figure(figsize=(8,8))
2 gs = f.add_gridspec(2, 3)
3
4 with sns.axes_style("darkgrid"):
5     ax = f.add_subplot(gs[0, 0])
6     sns.distplot(df_mart.Item_Weight,bins=20)
7
8
9 with sns.axes_style("white"):
10    ax = f.add_subplot(gs[0, 1])
11    sns.distplot(df_mart.Item_Visibility)
12
13 with sns.axes_style("ticks"):
14    ax = f.add_subplot(gs[0, 2])
15    sns.distplot(df_mart.Item_MRP,bins=10)
16
17 with sns.axes_style("white"):
18    ax = f.add_subplot(gs[1, 0])
19    sns.distplot(df_mart.Item_Outlet_Sales,bins=20)
20
21 #Theo distplot thì Item_weight trông không tuân theo phân phối chuẩn và tập trung chủ yếu là mức từ 5 tới 10.
22 #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
23 #Item_MRP tập trung nhiều ở 75-100 và từ 150 - 200
24 #Item_Outlet_sales tuân theo phân phối chuẩn và hơi lệch phải với đỉnh ở 1500 - 2000
```

```
<seaborn.axisgrid.FacetGrid at 0x7fac73279d90>
```



```

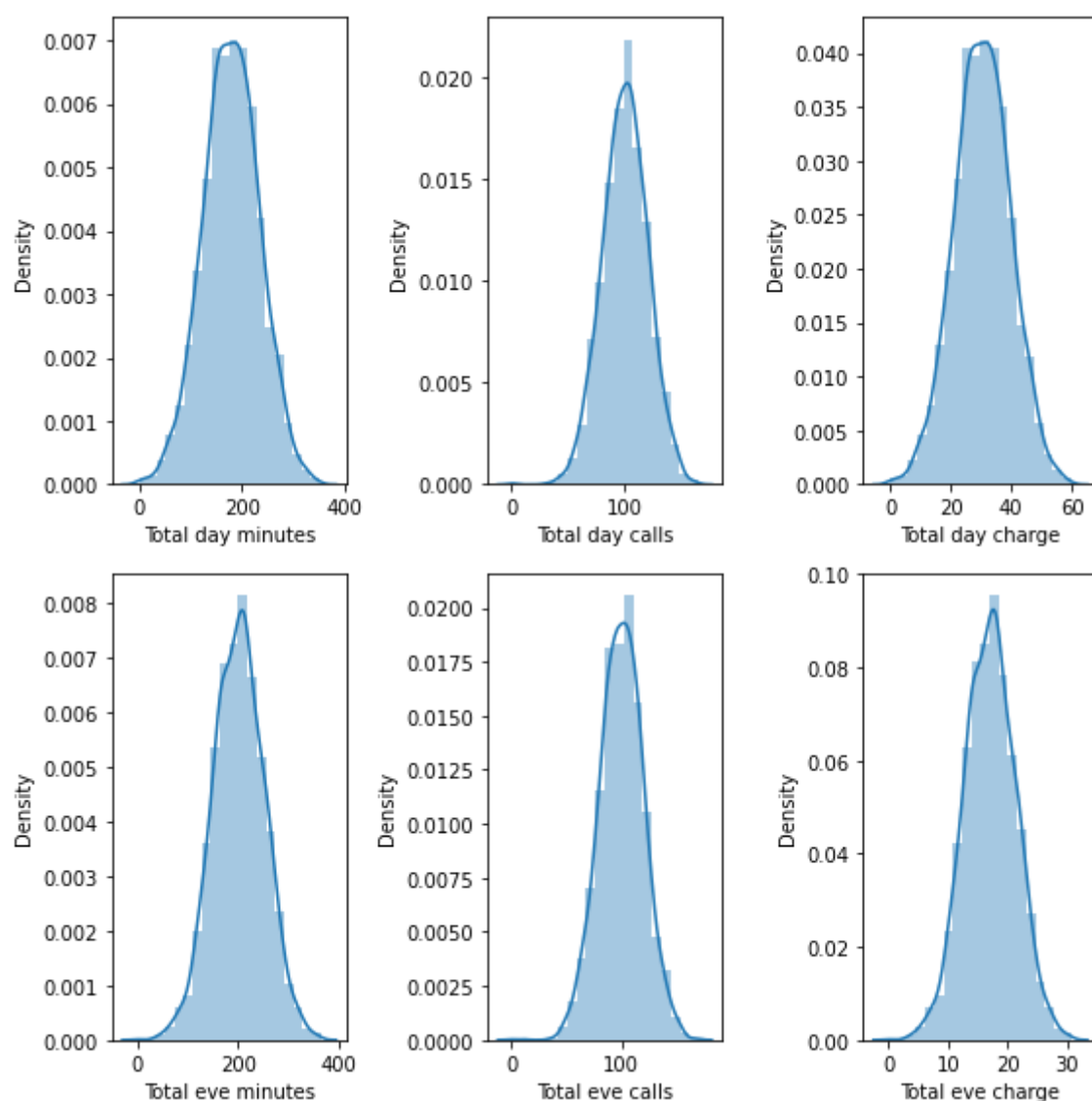
Account length          int64
Area code               int64
International plan      object
Voice mail plan         object
Number vmail messages   int64
Total day minutes       float64
Total day calls         int64
Total day charge        float64
Total eve minutes       float64
Total eve calls         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

```

```

1 f = plt.figure(figsize=(8,8))
2 gs = f.add_gridspec(2, 3)
3
4 ax = f.add_subplot(gs[0, 0])
5 sns.distplot(df_customer['Total day minutes'],bins=20)
6
7 ax = f.add_subplot(gs[0, 1])
8 sns.distplot(df_customer['Total day calls'],bins=20)
9
10 ax = f.add_subplot(gs[0, 2])
11 sns.distplot(df_customer['Total day charge'],bins=20)
12
13 ax = f.add_subplot(gs[1, 0])
14 sns.distplot(df_customer['Total eve minutes'],bins=20)
15
16 ax = f.add_subplot(gs[1, 1])
17 sns.distplot(df_customer['Total eve calls'],bins=20)
18
19 ax = f.add_subplot(gs[1, 2])
20 sns.distplot(df_customer['Total eve charge'],bins=20)
21
22 f.tight_layout()
23 #Tất cả các plot đều tuân theo dạng chuẩn trong đó đỉnh nằm ở giữa đồ thị

```



```

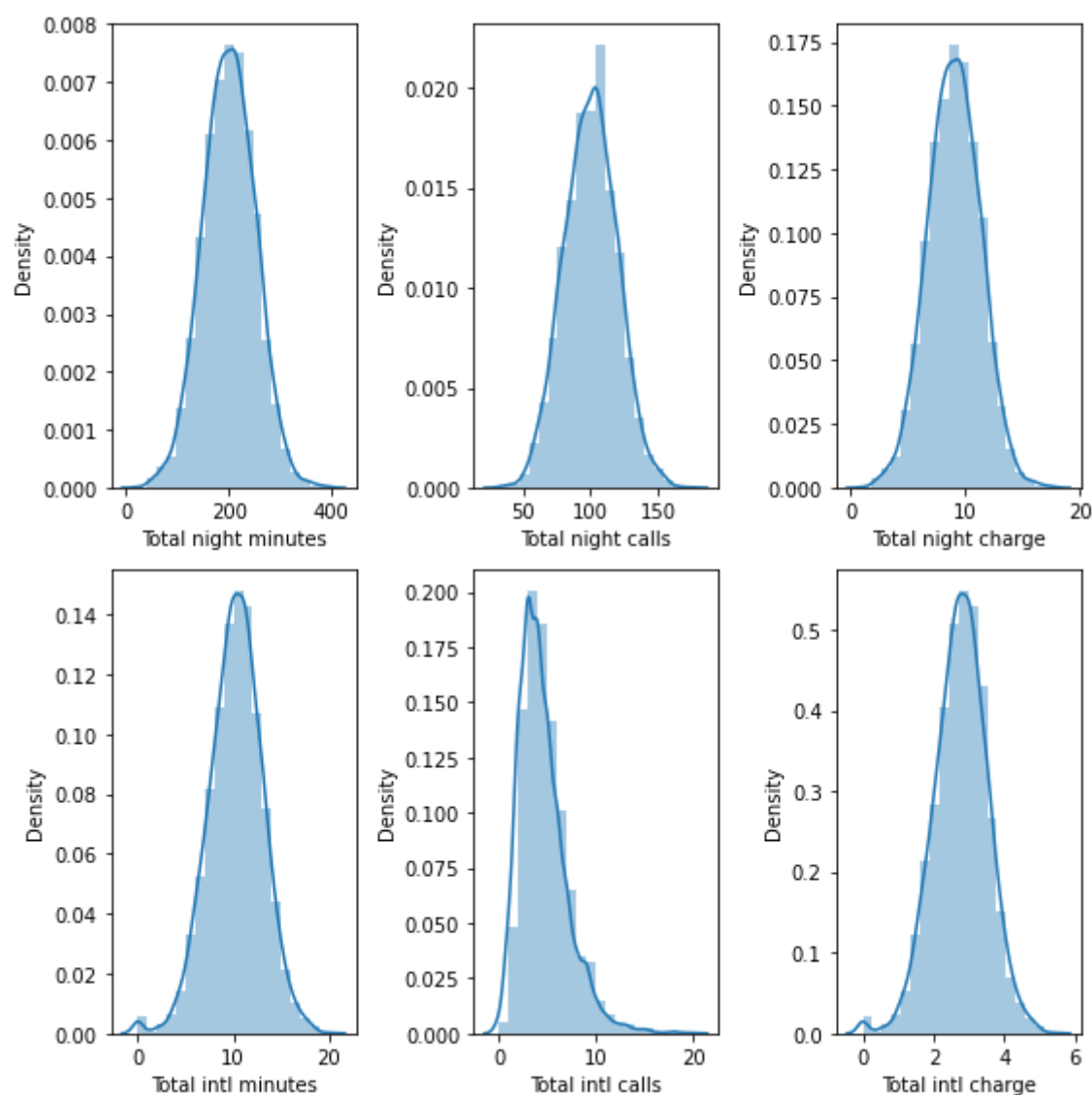
1 f = plt.figure(figsize=(8,8))
2 gs = f.add_gridspec(2, 3)
3
4 ax = f.add_subplot(gs[0, 0])
5 sns.distplot(df_customer['Total night minutes'],bins=20)

```

```

5 sns.distplot(df_customer['Total night minutes'],bins=20)
6
7 ax = f.add_subplot(gs[0, 1])
8 sns.distplot(df_customer['Total night calls'],bins=20)
9
10 ax = f.add_subplot(gs[0, 2])
11 sns.distplot(df_customer['Total night charge'],bins=20)
12
13 ax = f.add_subplot(gs[1, 0])
14 sns.distplot(df_customer['Total intl minutes'],bins=20)
15
16 ax = f.add_subplot(gs[1, 1])
17 sns.distplot(df_customer['Total intl calls'],bins=20)
18
19 ax = f.add_subplot(gs[1, 2])
20 sns.distplot(df_customer['Total intl charge'],bins=20)
21
22 f.tight_layout()
23 #Tất cả đều dạng chuẩn và có đỉnh ở giữa đồ thị. Trừ total_intl call bị lệch phải.

```

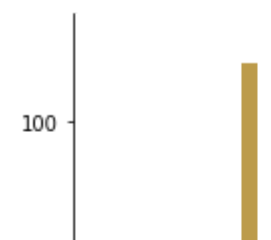


```

1 sns.catplot(x='State',kind='count',data=df_customer,height=8).set_xticklabels(rotation=60)
2 #WV là state có số lần xuất hiện nhiều nhất

```

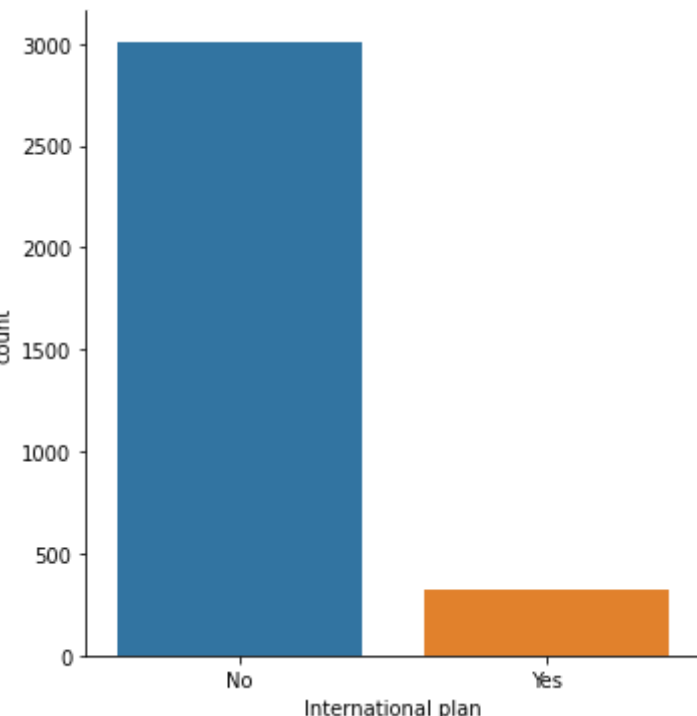
```
<seaborn.axisgrid.FacetGrid at 0x7fac72c53fd0>
```



```
1 sns.catplot(x='International plan',kind='count',data=df_customer)
```

2 #Đa số có international plan là No với khoảng 3000

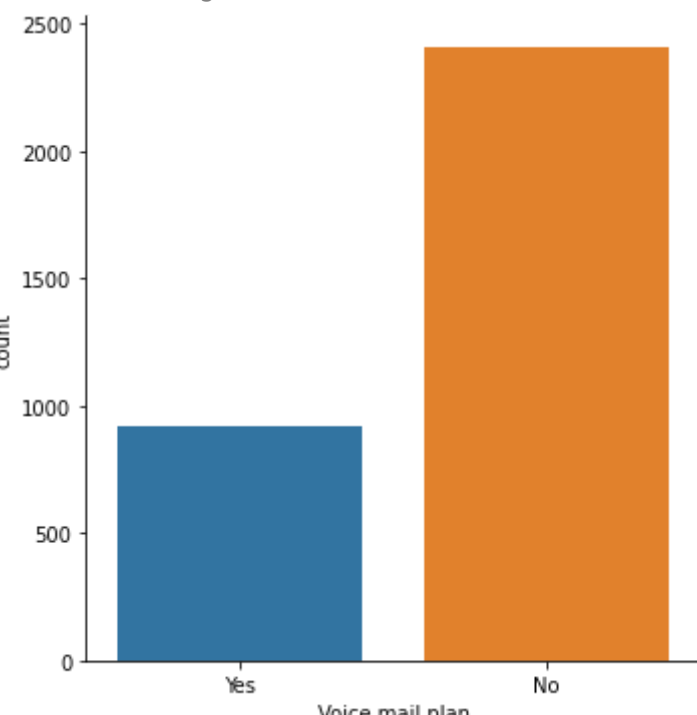
```
<seaborn.axisgrid.FacetGrid at 0x7fac72bca0d0>
```



```
1 sns.catplot(x='Voice mail plan',kind='count',data=df_customer)
```

2 #Đa số là No với số lượng gần 2500

```
<seaborn.axisgrid.FacetGrid at 0x7fac732739d0>
```

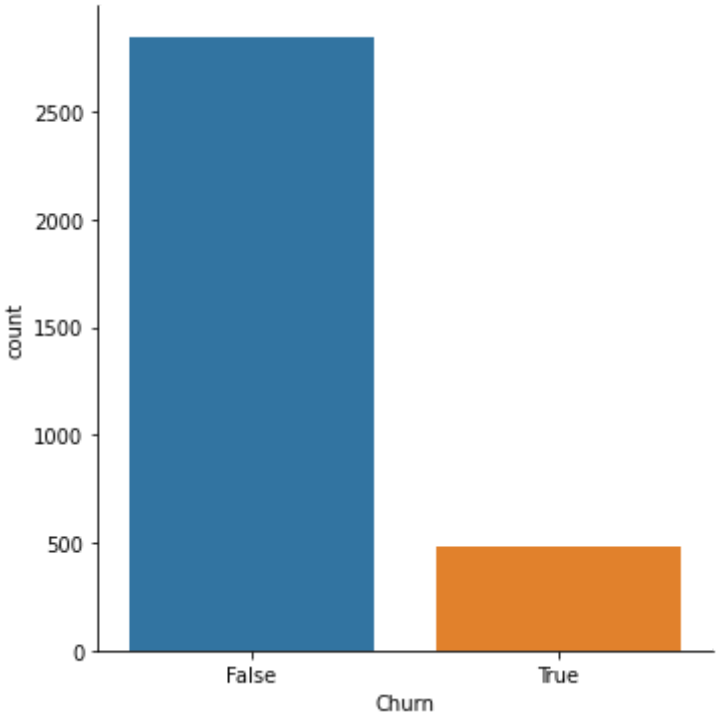


```
1 sns.catplot(x='Customer service calls',kind='count',data=df_customer)
```

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

```
<seaborn.axisgrid.FacetGrid at 0x7fac732e3b90>
1200 |
1 sns.catplot(x='Churn',kind='count',data=df_customer)
2 #False chiếm đa số với hơn 2500
```

<seaborn.axisgrid.FacetGrid at 0x7fac75405290>

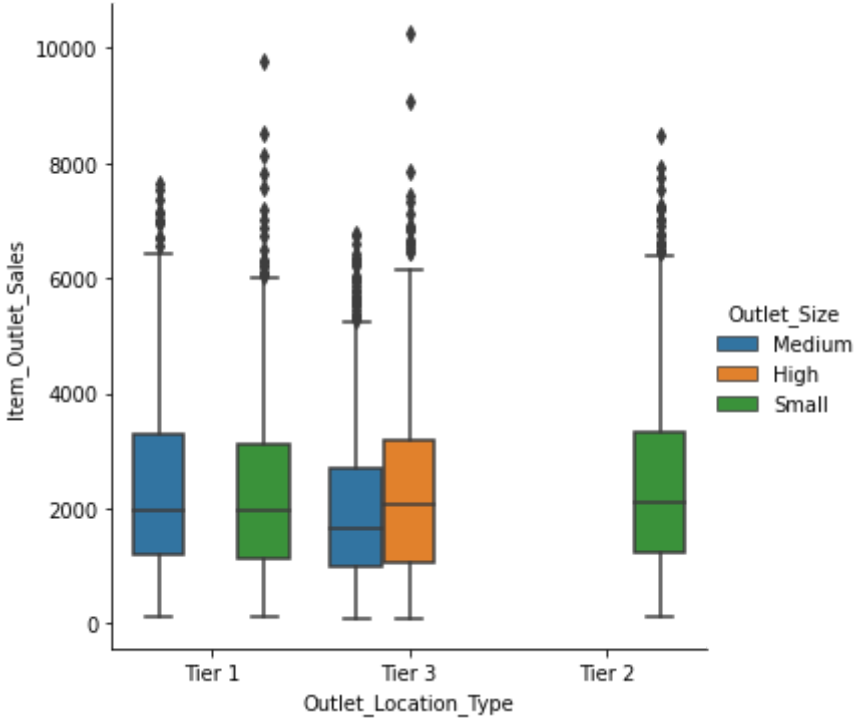


Chọn 2, 3, hay 4 tiêu chí bạn nghi ngờ có mối quan hệ với nhau mật thiết và biểu diễn chúng lên một hình EDA sau đó cho nhận xét về mối quan hệ (Mỗi dữ liệu CustomerChurn hay BigMartSales cho 3 TH này)

▼ BigMart Sales

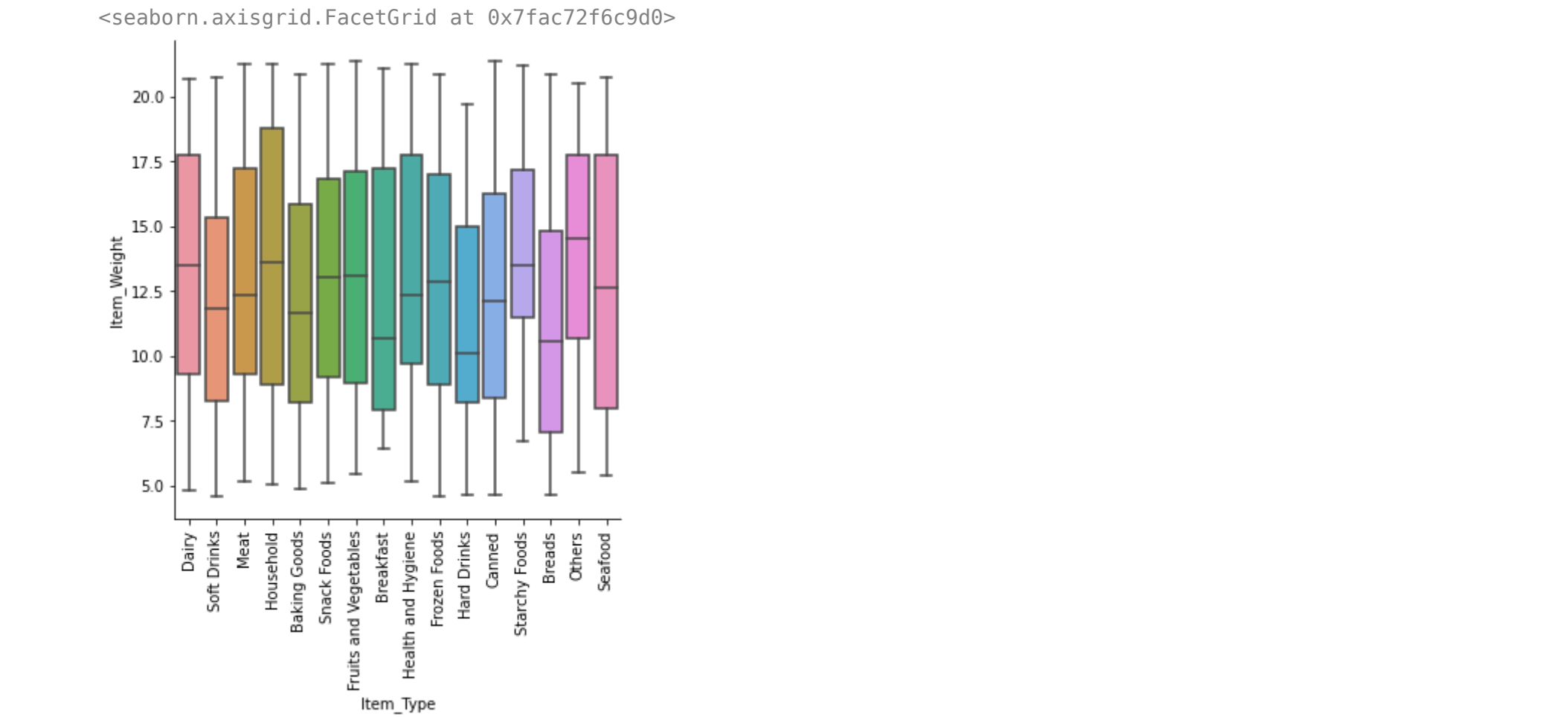
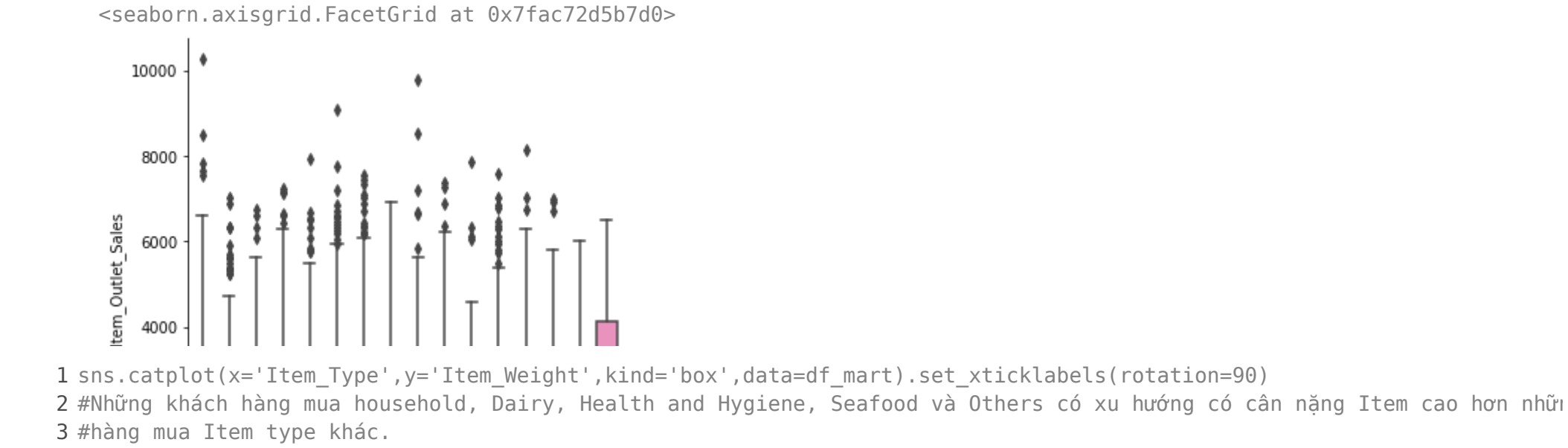
```
1 sns.catplot(x='Outlet_Location_Type',y='Item_Outlet_Sales',hue='Outlet_Size',kind='box',data=df_mart)
2 #Có vẻ như Tier 2 chỉ có Outlet_Size ở dạng Small, Tier 1 không có Outlet có kích cỡ lớn và tier 3 không có outlet cỡ l
```

<seaborn.axisgrid.FacetGrid at 0x7fac7539ad50>



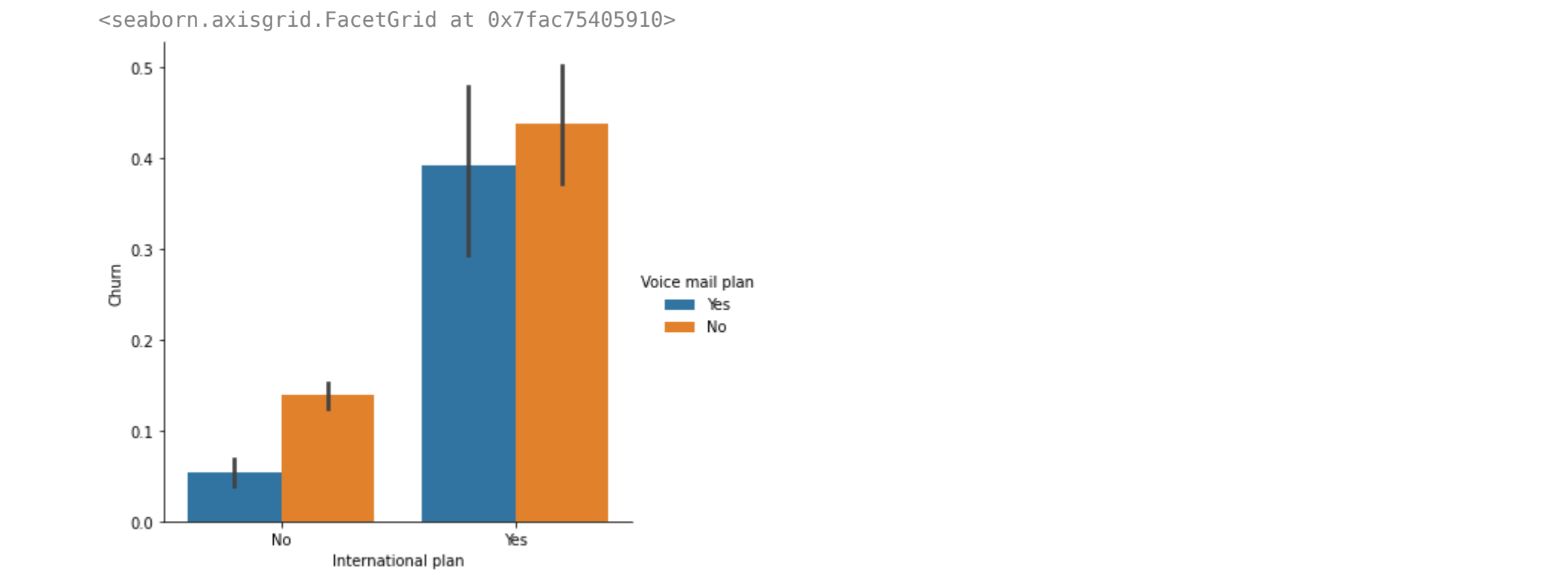
```
1 sns.catplot(x='Item_Type',y='Item_Outlet_Sales',kind='box',data=df_mart).set_xticklabels(rotation=90)
2 #Có vẻ như Hard Drink có lượng Item_Outlet_Sales thấp hơn cả. Trong khi đó với Item_type là Seafood thì người ta chi t
3 #bình nhiều tiền hơn cho Item_Outlet
```





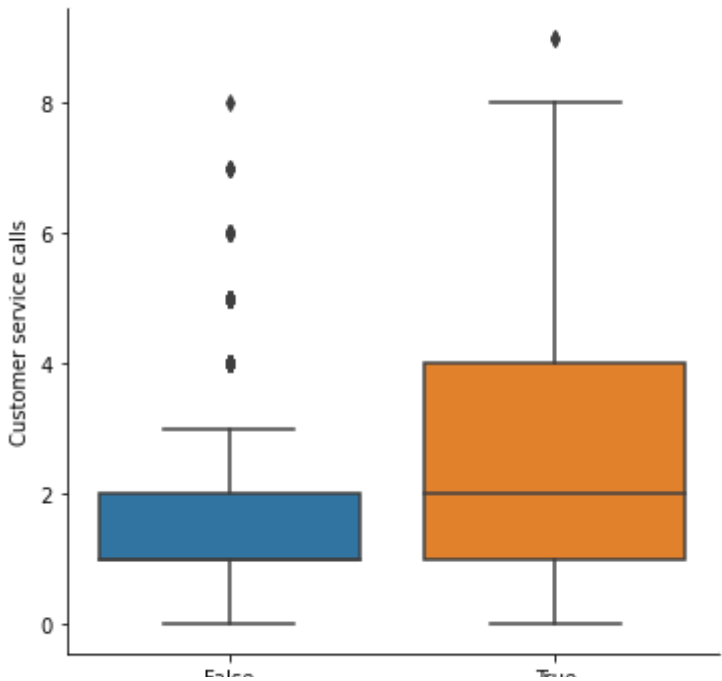
▼ Customer Churn

1 sns.catplot(data=df\_customer,x='International plan',y='Churn',hue='Voice mail plan',kind='bar')  
2 #Với international plan No thì số lượng Voice mail plan với Churn đều thấp. Trong khi đó, tỉ lệ Voice mail plan và Chu  
3 #Yes hay no thì đều đa số nằm ở International plan là Yes.



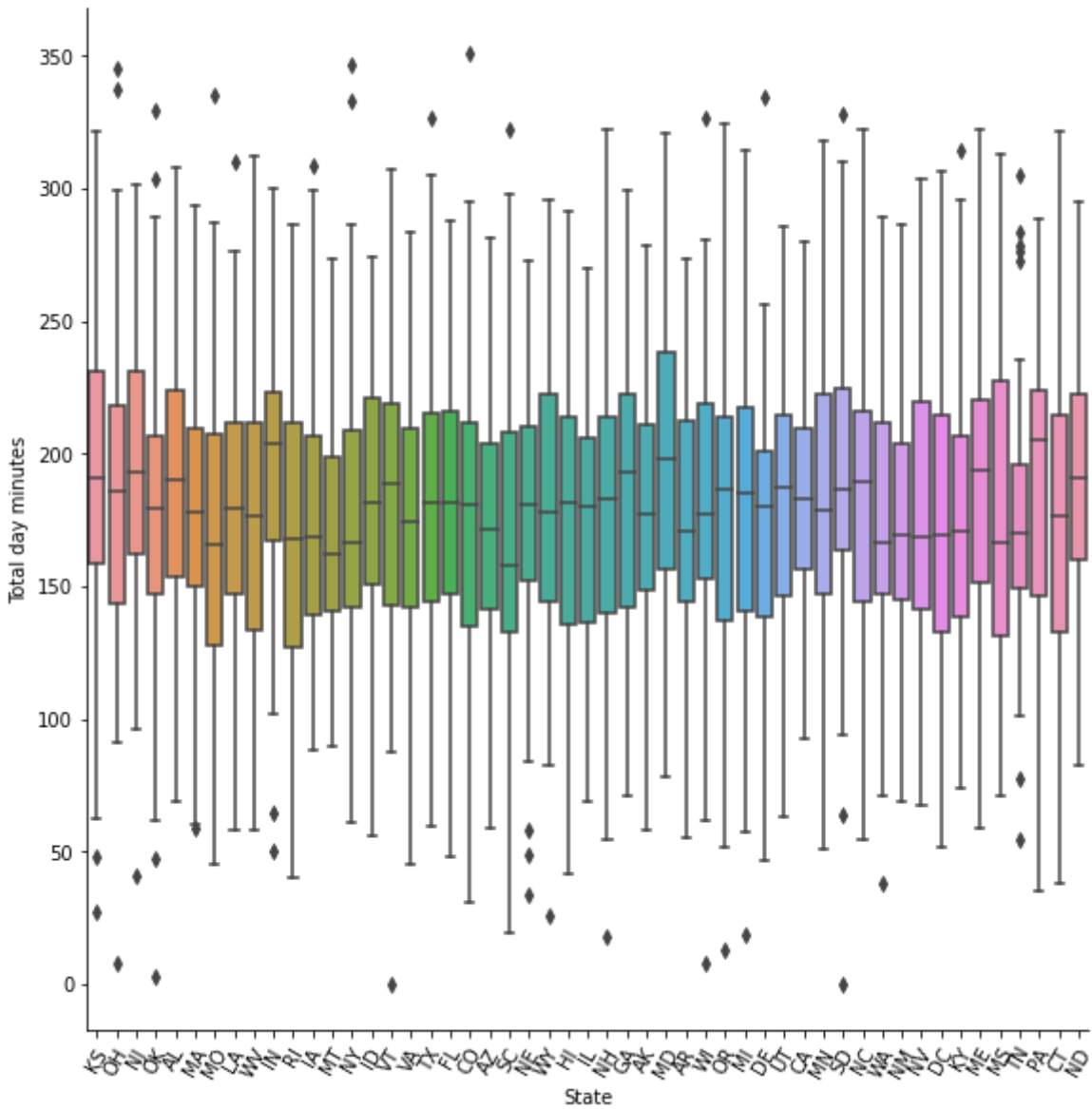
1 sns.catplot(data=df\_customer,x='Churn',y='Customer service calls',kind='box')  
2 #Với những khách hàng có Churn là True thì trung bình số cuộc gọi dịch vụ mà họ thực hiện  
3 #Nhiều hơn rất nhiều so với khách hàng không Churn.

<seaborn.axisgrid.FacetGrid at 0x7fac731b7890>



```
1 sns.catplot(data=df_customer,x='State',y='Total day minutes',kind='box',height=8).set_xticklabels(rotation=60)
2 #Các bang IN, PA, MD, ME có tổng số phút gọi ban ngày trung bình là cao hơn các bang còn lại.
```

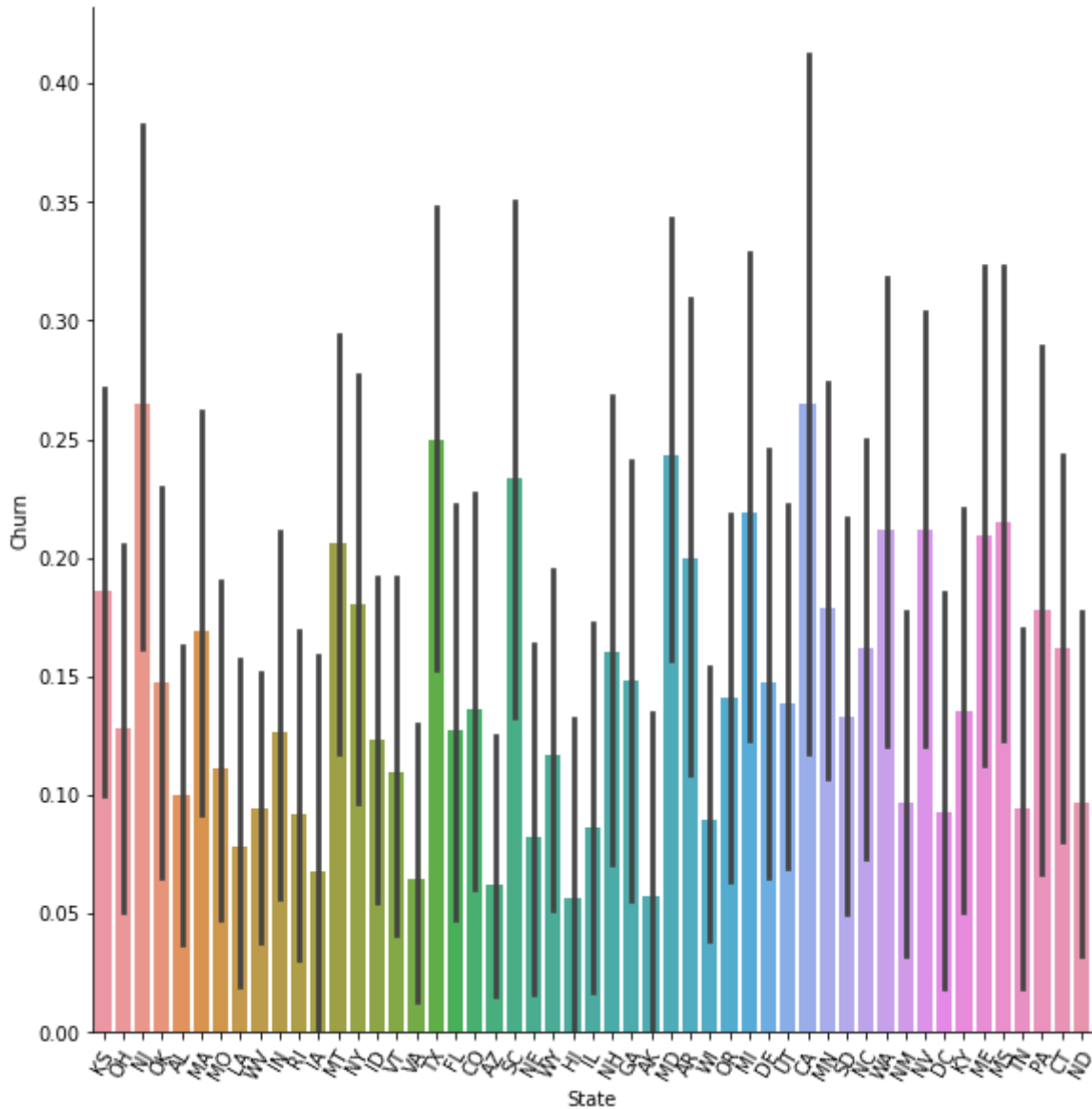
<seaborn.axisgrid.FacetGrid at 0x7fac72573750>



```
1 sns.catplot(data=df_customer,x='State',y='Churn',kind='bar',height=8).set_xticklabels(rotation=60)
2 #Bang NJ, TX, CA, MD là các bang có tỉ lệ Churn cao nhất.
```



<seaborn.axisgrid.FacetGrid at 0x7fac726b1a50>



✓ 2s completed at 12:31 AM

