Data Mining - Lab 1

Huỳnh Thị Thắm - 18110209

In [14]:
```
path = "G:\\University_Learning\\data-mining\\Lab01\\"
import os
os.chdir(path)
currentWorkingDir = %pwd
currentWorkingDir
```

Out[14]:
```
'G:\\University_Learning\\data-mining\\Lab01'
```

In [15]:
```
# Read data and look statistics
import pandas as pd
data = pd.read_csv("Dataset\\Telco Customer Churn.csv")

print("Display all first of 5 rows :")
display(data.head())
print("The shape of data in (nrows,ncols)")
print(data.shape)
```

```
Display all first of 5 rows :
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtectior
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No

5 rows × 21 columns

```
The shape of data in (nrows,ncols)
(7043, 21)
```

In [16]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from IPython.display import display, Image
import warnings
warnings.filterwarnings("ignore")
```

In [17]:
```
print("Range Index of Dataframe : \n\t", data.index)
print("\nColumn of Dataframe in the list type : \n\t", list(data.columns))
print("\nNumber of Internet Service in the data ? \n\t", data["InternetService"].unique())
print(f"\nThe min max value of Monthly Charges in the data ? \n\t  \
      from min : {np.min(data['MonthlyCharges'])} to max : {np.max(data['MonthlyCharges'])}")
print(f"\nThe mean std value of tenure in the data ? \n\t  \
      mean : {data['tenure'].mean()} std : {data['tenure'].std()}")
print(f"\nThe sum and median value of tenure in the data ? \n\t  \
      sum : {data['tenure'].sum()} median : {data['tenure'].median()}")
```

```
Range Index of Dataframe :
         RangeIndex(start=0, stop=7043, step=1)

Column of Dataframe in the list type :
         ['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines',
'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovie
s', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn']

Number of Internet Service in the data ?
         ['DSL' 'Fiber optic' 'No']

The min max value of Monthly Charges in the data ?
                from min : 18.25 to max : 118.75

The mean std value of tenure in the data ?
                mean : 32.37114865824223 std : 24.55948102309423

The sum and median value of tenure in the data ?
                sum : 227990 median : 29.0
```

In [18]:
```
print("How many cases of Churn ?")
print(data["Churn"].value_counts())
print("\n")

print("Can we see the statistics table of the whole data ?")
display(data.describe())
print("\n")

print("Is there any missing value at all columns ?")
display(data.isnull().sum())
print("\n")
```

```
How many cases of Churn ?
No     5174
Yes    1869
Name: Churn, dtype: int64


Can we see the statistics table of the whole data ?
```

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

```
Is there any missing value at all columns ?
```

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

In [19]:
```
print("Number of type of contracts in the table :")
print(len(data["Contract"].unique()))
print("\n")
print("The rows from index 10 to 15 :")
display(data.loc[10:15, :])
print("\n")
print("Reset index of the above results in a new table : ")
df = data.loc[10:15, :]
df = df.reset_index(drop = True)
display(df)
print("\n")
print("The rows from index 10 to 15 of columns tenure, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCh
display(data.loc[10:15, ['customerID', 'tenure', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'Tota
print("\n")
print("The rows from index 10 of columns tenure, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges
display(data.loc[10, ['customerID', 'tenure', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCh
```

```
Number of type of contracts in the table :
3


The rows from index 10 to 15 :
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtectio
10	9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	...	N
11	7469-LKBCI	Male	0	No	No	16	Yes	No	No	No internet service	...	No intern servi
12	8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	...	Y
13	0280-XJGEX	Male	0	No	No	49	Yes	Yes	Fiber optic	No	...	Y
14	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	...	Y
15	3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	...	Y

6 rows × 21 columns

Reset index of the above results in a new table :

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtectior
0	9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	...	No
1	7469-LKBCI	Male	0	No	No	16	Yes	No	No	No internet service	...	No interne service
2	8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	...	Yes
3	0280-XJGEX	Male	0	No	No	49	Yes	Yes	Fiber optic	No	...	Yes
4	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	...	Yes
5	3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	...	Yes

6 rows × 21 columns

The rows from index 10 to 15 of columns tenure, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges, Churn :

	customerID	tenure	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
10	9763-GRSKD	13	Month-to-month	Yes	Mailed check	49.95	587.45	No
11	7469-LKBCI	16	Two year	No	Credit card (automatic)	18.95	326.8	No
12	8091-TTVAX	58	One year	No	Credit card (automatic)	100.35	5681.1	No

	customerID	tenure	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
13	0280-XJGEX	49	Month-to-month	Yes	Bank transfer (automatic)	103.70	5036.3	Yes
14	5129-JLPIS	25	Month-to-month	Yes	Electronic check	105.50	2686.05	No
15	3655-SNQYZ	69	Two year	No	Credit card (automatic)	113.25	7895.15	No

```
The rows from index 10 of columns tenure, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges, Chur
n :

customerID              9763-GRSKD
tenure                          13
Contract           Month-to-month
PaperlessBilling              Yes
PaymentMethod         Mailed check
MonthlyCharges               49.95
TotalCharges                587.45
Churn                           No
Name: 10, dtype: object
```

In [20]:

```
data = pd.read_csv("Dataset\\Telco Customer Churn.csv")

print("The type of all columns in data :")
display(data.info())
print("\n")

print("Change SeniorCitizen from int64 to object :")
print("Original Type of SeniorCitizen :", data["SeniorCitizen"].dtypes)
MapDict = {1 : "Yes", 0 : "No"}
data["SeniorCitizen"] = data["SeniorCitizen"].map(MapDict)
print("New Type of SeniorCitizen :", data["SeniorCitizen"].dtypes)
print("\n")

print("Extract the categorical and numeric columns :")
CatFeatures = [col for col in data.columns if data[col].dtypes in ["object", "bool"]]
NumFeatures = [col for col in data.columns if data[col].dtypes in ["int64", "float64"]]
print("Categorical Features :", CatFeatures)
print("Numeric Features :", NumFeatures)
print("\n")

print("Show the all statistics of Numeric Features :")
display(data.describe())
print("\n")

print("Show the all statistics of Categorical Features :")
display(data[CatFeatures].describe(include='all'))
print("\n")

print("Get data from describe table :")
NumStats = data[NumFeatures].describe(include='all')
CatStats = data[CatFeatures].describe(include='all')
MonthlyCharges_50 = NumStats.loc["50%", "MonthlyCharges"]
Churn_top_freq = CatStats.loc[["top", "freq"], "Churn"]
print("Monthly Charges at 50 %(median) : \n", MonthlyCharges_50)
print("Top and Frequency of Top in Churn : \n", Churn_top_freq)
```

```
The type of all columns in data :
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
```

```
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB

None
```

```
Change SeniorCitizen from int64 to object :
Original Type of SeniorCitizen : int64
New Type of SeniorCitizen : object


Extract the categorical and numeric columns :
Categorical Features : ['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService', 'MultipleLi
nes', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'Streami
ngMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'TotalCharges', 'Churn']
Numeric Features : ['tenure', 'MonthlyCharges']
```

Show the all statistics of Numeric Features :

	tenure	MonthlyCharges
count	7043.000000	7043.000000
mean	32.371149	64.761692
std	24.559481	30.090047
min	0.000000	18.250000
25%	9.000000	35.500000
50%	29.000000	70.350000
75%	55.000000	89.850000
max	72.000000	118.750000

Show the all statistics of Categorical Features :

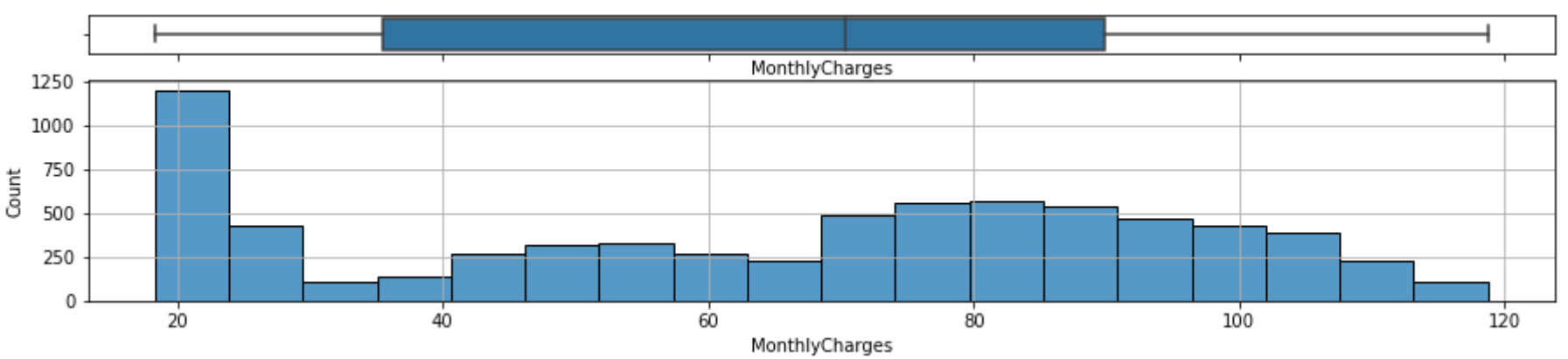
	customerID	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	Device
count	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043	
unique	7043	2	2	2	2	2	3	3	3	3	
top	7590-VHVEG	Male	No	No	No	Yes	No	Fiber optic	No	No	
freq	1	3555	5901	3641	4933	6361	3390	3096	3498	3088	

Get data from describe table :
Monthly Charges at 50 %(median) :
 70.35
Top and Frequency of Top in Churn :
 top   No

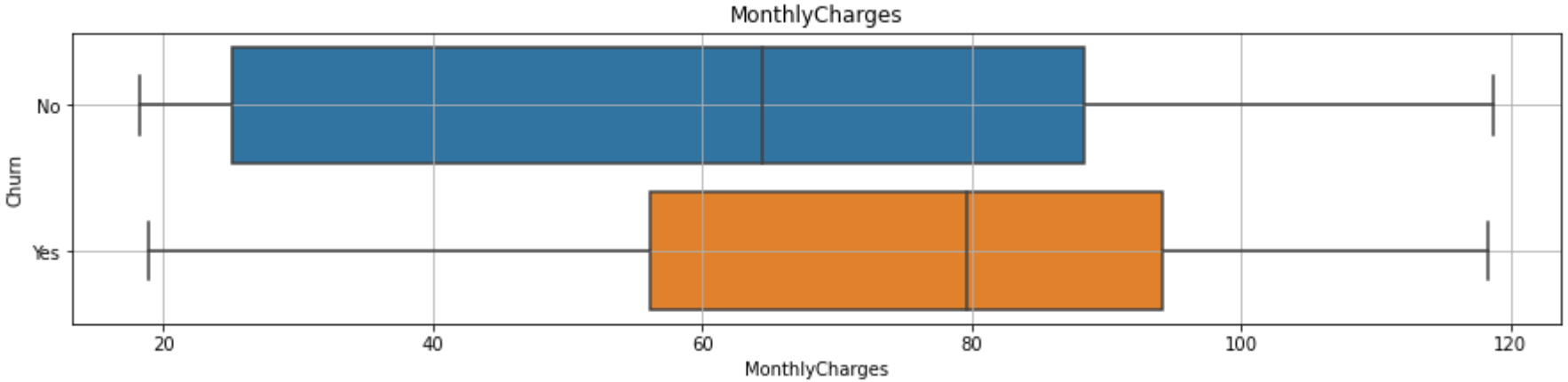
freq 5174

In [21]:
```
data = pd.read_csv("Dataset\\Telco Customer Churn.csv")

print("Draw chart for a numeric features :")
feature = "MonthlyCharges"
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85)})
f.set_figheight(3)
f.set_figwidth(15)
sns.boxplot(data[feature], ax=ax_box)
sns.histplot(data=data, x=feature, ax=ax_hist)
plt.grid()
plt.show()

print("Draw chart for a numeric feature according to a categorical feature :")
feature = "MonthlyCharges"
plt.figure(figsize = (15,3))
sns.boxplot(y ='Churn', x = feature, data = data)
plt.title(feature)
plt.grid()
plt.show()

print("Draw chart for two numeric features according to a categorical feature :")
plt.figure(figsize=(15,5))
feature_x = "MonthlyCharges"
feature_y = "tenure"
feature_hue = "Churn"
sns.scatterplot(x = feature_x, y= feature_y, hue=feature_hue, data = data, legend='full')
plt.grid()
plt.show()
```

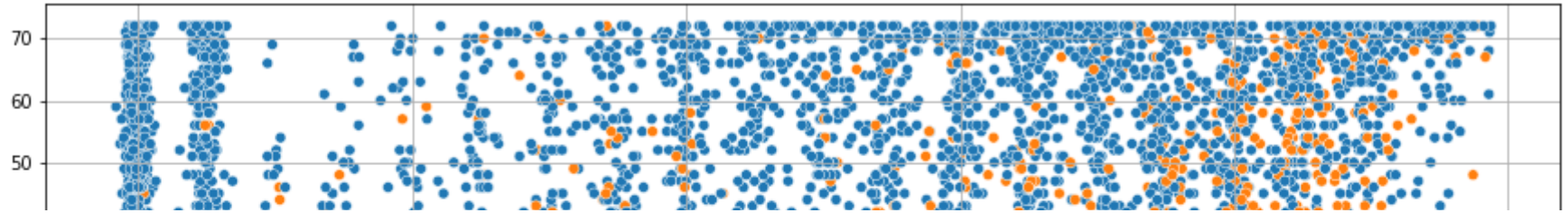
```
Draw chart for a numeric features :
```



Draw chart for a numeric feature according to a categorical feature :

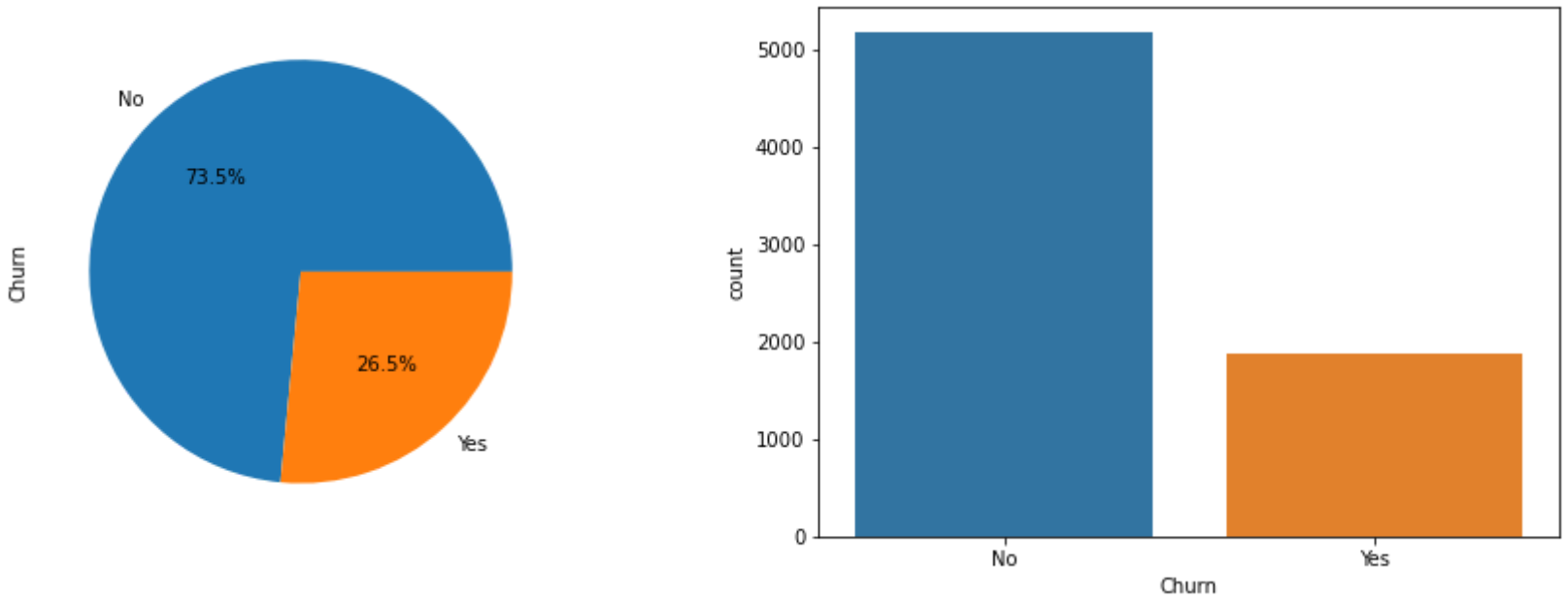


Draw chart for two numeric features according to a categorical feature :

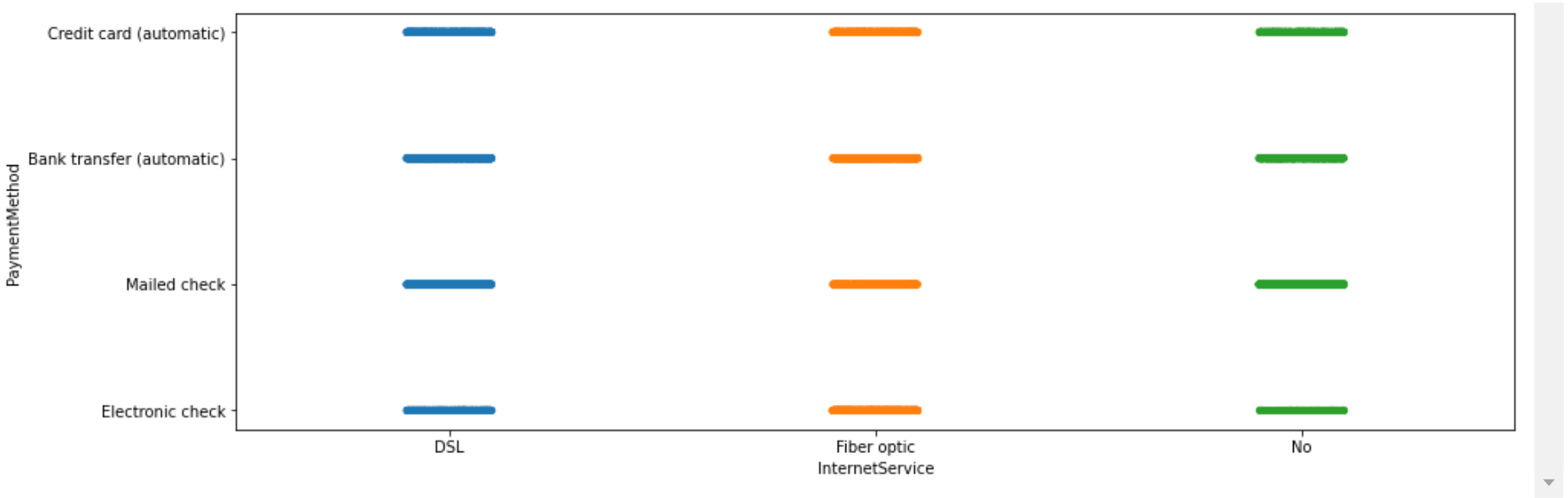

70
60
50

In [22]:
```
print("Draw chart for a categorical feature :")
feature = "Churn"
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
data[feature].value_counts().plot.pie(autopct='%1.1f%%')
plt.subplot(1,2,2)
sns.countplot(data[feature])
plt.show()

print("Draw chart for a categorical feature according to another categorical feature :")
plt.figure(figsize=(15,5))
feature_x = "InternetService"
feature_y = "PaymentMethod"
sns.stripplot(data[feature_x],data[feature_y])
plt.show()
```

Draw chart for a categorical feature :



Draw chart for a categorical feature according to another categorical feature :



In [23]:
```
data.columns
```

Out[23]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

In [24]:

```
s', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovi

lessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges']

feature_ser) - set(feature_info)) + ["Churn"]


fo], axis = 1)
```

Split data into many data :

	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies
0	No	No phone service	DSL	No	Yes	No	No	No	No
1	Yes	No	DSL	Yes	No	Yes	No	No	No
2	Yes	No	DSL	Yes	Yes	No	No	No	No
3	No	No phone service	DSL	Yes	No	Yes	Yes	No	No
4	Yes	No	Fiber optic	No	No	No	No	No	No

	tenure	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges
0	1	Month-to-month	Yes	Electronic check	29.85	29.85
1	34	One year	No	Mailed check	56.95	1889.5
2	2	Month-to-month	Yes	Mailed check	53.85	108.15
3	45	One year	No	Bank transfer (automatic)	42.30	1840.75
4	2	Month-to-month	Yes	Electronic check	70.70	151.65

	Partner	Churn	SeniorCitizen	Dependents	gender	customerID	Churn
0	Yes	No	0	No	Female	7590-VHVEG	No
1	No	No	0	No	Male	5575-GNVDE	No
2	No	Yes	0	No	Male	3668-QPYBK	Yes
3	No	No	0	No	Male	7795-CFOCW	No
4	No	Yes	0	No	Female	9237-HQITU	Yes

Merge two data into one by cols :

	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	tenure
0	No	No phone service	DSL	No	Yes	No	No	No	No	1
1	Yes	No	DSL	Yes	No	Yes	No	No	No	34
2	Yes	No	DSL	Yes	Yes	No	No	No	No	2
3	No	No phone service	DSL	Yes	No	Yes	Yes	No	No	45
4	Yes	No	Fiber optic	No	No	No	No	No	No	2

In [25]:
```
print("Filter data by condition :")
Condition1 = data["MonthlyCharges"] > 80
Condition2 = data["SeniorCitizen"] == 1
data_over80_SeniorCitizen = data[Condition1 & Condition2].copy()
display(data_over80_SeniorCitizen.head())
print(data_over80_SeniorCitizen.shape)

value1, value2 = 100 , 1
data_less100_SeniorCitizen = data.query("`MonthlyCharges` < @value1 and `SeniorCitizen` == @value2")
display(data_less100_SeniorCitizen.head())
print(data_less100_SeniorCitizen.shape)

print("Merge two data into one by rows :")
data_merge = pd.concat([data_over80_SeniorCitizen, data_less100_SeniorCitizen])
display(data_merge.head())
print(data_merge.shape)
```

```
Filter data by condition :
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtectio
30	3841-NFECX	Female	1	Yes	No	71	Yes	Yes	Fiber optic	Yes	...	Y
31	4929-XIHVW	Male	1	Yes	No	2	Yes	No	Fiber optic	No	...	Y
50	8012-SOUDQ	Female	1	No	No	43	Yes	Yes	Fiber optic	No	...	N
53	7495-OOKFY	Female	1	Yes	No	8	Yes	Yes	Fiber optic	No	...	N
55	1658-BYGOY	Male	1	No	No	18	Yes	Yes	Fiber optic	No	...	N

5 rows × 21 columns

```
(657, 21)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtectio

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtectio
20	8779-QRDMV	Male	1	No	No	1	No	No phone service	DSL	No	...	Y
30	3841-NFECX	Female	1	Yes	No	71	Yes	Yes	Fiber optic	Yes	...	Y
31	4929-XIHVW	Male	1	Yes	No	2	Yes	No	Fiber optic	No	...	Y
34	3413-BMNZE	Male	1	No	No	1	Yes	No	DSL	No	...	N
50	8012-SOUDQ	Female	1	No	No	43	Yes	Yes	Fiber optic	No	...	N

5 rows × 21 columns

(906, 21)
Merge two data into one by rows :

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtectio
30	3841-NFECX	Female	1	Yes	No	71	Yes	Yes	Fiber optic	Yes	...	Y
31	4929-XIHVW	Male	1	Yes	No	2	Yes	No	Fiber optic	No	...	Y
50	8012-SOUDQ	Female	1	No	No	43	Yes	Yes	Fiber optic	No	...	N
53	7495-OOKFY	Female	1	Yes	No	8	Yes	Yes	Fiber optic	No	...	N
55	1658-BYGOY	Male	1	No	No	18	Yes	Yes	Fiber optic	No	...	N

5 rows × 21 columns

(1563, 21)

In [26]:
```
print("Group by gender and count on 'PhoneService', 'MultipleLines', 'InternetService', \
      'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', \
      sum on tenure and Average on MonthlyCharges :")
feature_ser = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
genderDF = pd.DataFrame()
series = data[(data['PhoneService'] == "Yes") & (data['MultipleLines'] == "Yes") & (data['InternetService'] != "No")
            & (data['OnlineSecurity'] == "Yes") & (data['OnlineBackup'] == "Yes") & (data['DeviceProtection'] == "Yes")
            & (data['TechSupport'] == "Yes") & (data['StreamingTV'] == "Yes") & (data['StreamingMovies'] == "Yes")]
series = series.groupby("gender")[feature_ser].count()
genderDF.index  = series.index
genderDF[["Count on "+feature for feature in feature_ser]] = series
genderDF["Sum on tenure"] = data.groupby("gender")["tenure"].sum()
genderDF["Average on MonthlyCharges"] = data.groupby("gender")["MonthlyCharges"].mean()
display(genderDF.head())

print("Apply a function to create a new column Total Services")
temp_data = data.copy()
DictReplace = {'No phone service': 0, 'No internet service': 0, 'No': 0, 'Yes': 1, 'DSL': 1, 'Fiber optic': 1}
temp_data.replace(DictReplace, inplace=True)
data['Total Services'] = temp_data[feature_ser].apply(lambda fea: np.sum(fea), axis=1)
display(data[['customerID'] + feature_ser + ["Total Services"]].head())

print("Join data with gender on the gender information to create new information about gender")
genderDF["gender"] = genderDF.index
genderDF = genderDF.reset_index(drop = True)
display(genderDF.head())
data = pd.merge(data, genderDF, left_on='gender', right_on='gender')
display(data.head())
```

```
Group by gender and count on 'PhoneService', 'MultipleLines', 'InternetService',       'OnlineSecurity', 'OnlineBacku
p', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',       sum on tenure and Average on MonthlyChar
ges :
```

gender	Count on PhoneService	Count on MultipleLines	Count on InternetService	Count on OnlineSecurity	Count on OnlineBackup	Count on DeviceProtection	Count on TechSupport	Count on StreamingTV	Count on StreamingMovie
Female	114	114	114	114	114	114	114	114	11

gender	Count on PhoneService	Count on MultipleLines	Count on InternetService	Count on OnlineSecurity	Count on OnlineBackup	Count on DeviceProtection	Count on TechSupport	Count on StreamingTV	Count on StreamingMovie
Male	94	94	94	94	94	94	94	94	9

Apply a function to create a new column Total Services

	customerID	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingM
0	7590-VHVEG	No	No phone service	DSL	No	Yes	No	No	No	
1	5575-GNVDE	Yes	No	DSL	Yes	No	Yes	No	No	
2	3668-QPYBK	Yes	No	DSL	Yes	Yes	No	No	No	
3	7795-CFOCW	No	No phone service	DSL	Yes	No	Yes	Yes	No	
4	9237-HQITU	Yes	No	Fiber optic	No	No	No	No	No	

Join data with gender on the gender information to create new information about gender

	Count on PhoneService	Count on MultipleLines	Count on InternetService	Count on OnlineSecurity	Count on OnlineBackup	Count on DeviceProtection	Count on TechSupport	Count on StreamingTV	Count on StreamingMovies	Sum of tenure
0	114	114	114	114	114	114	114	114	114	11246
1	94	94	94	94	94	94	94	94	94	11552

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	Count on MultipleLines

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	Count on MultipleLines
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	114
1	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	114
2	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	...	114
3	6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL	Yes	...	114
4	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	...	114

5 rows × 33 columns