# Data Mining - Lab 01

- Full name: Đinh Anh Huy
- Student ID: 18110103

In [1]:
```
from google.colab import drive
drive.mount('/content/gdrive')

!ln -s /content/gdrive/My\ Drive/ /mydrive
```

```
Mounted at /content/gdrive
ln: failed to create symbolic link '/mydrive/My Drive': File exists
```

In [2]:
```
path = "/mydrive/Colab Notebooks/Data Mining/Lab01"
import os
os.chdir(path)
```

In [3]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.display import display, Image
import warnings
warnings.filterwarnings('ignore')
```

### Read data and display the shape of data.

In [4]:
```
# Read data
data = pd.read_csv('Dataset/Telco Customer Churn.csv')

print(">> Display the first 5 rows of data:")
display(data.head())
print(">> Shape of data: ", data.shape)
print("   * Number of rows: ", data.shape[0])
print("   * Number of columns: ", data.shape[1])
```

```
>> Display the first 5 rows of data:
```

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

5 rows × 21 columns

```
>> Shape of data:  (7043, 21)
   * Number of rows:  7043
   * Number of columns:  21
```

## Description of Telco Customer Churn Dataset

### Context

"Predict behavior to retain customers. You can analyze all relevant customer data and develop focused customer retention programs." [IBM Sample Data Sets]

**Content**

Each row represents a customer, each column contains customer’s attributes described on the column Metadata.

The data set includes information about:

- Customers who left within the last month – the column is called Churn.
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.
- Customer account information – how long they’ve been a customer, contract, payment method, paperless billing, monthly charges, and total charges.
- Demographic info about customers – gender, age range, and if they have partners and dependents.

## Statistics of Data

In [5]:
```
print(">> The information of data (name/type/number of values/check missing value of each columns in data): ")
display(data.info())
```

```
>> The information of data (name/type/number of values/check missing value of each 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
```

In [6]:
```
print(">> Number of unique values in each feature: ")
display(data.nunique())
print("* Comment: TotalCharges Feature is an object type but has many values.")
```

```
>> Number of unique values in each feature:

customerID          7043
gender                 2
SeniorCitizen          2
Partner                2
Dependents             2
tenure                73
PhoneService           2
MultipleLines          3
InternetService        3
OnlineSecurity         3
OnlineBackup           3
DeviceProtection       3
TechSupport            3
StreamingTV            3
StreamingMovies        3
Contract               3
PaperlessBilling       2
PaymentMethod          4
MonthlyCharges      1585
TotalCharges        6531
Churn                  2
dtype: int64

* Comment: TotalCharges Feature is an object type but has many values.
```

In [7]:
```
# change the values of the TotalCharges Feature to numeric values and replace empty to np.nan
data['TotalCharges'] = data['TotalCharges'].replace(' ', np.nan, regex=True)
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'])
```

In [8]:
```
# extract the categorical and numeric columns
CatFeatures = [col for col in data.columns if data[col].dtypes in ['object']]
NumFeatures = [col for col in data.columns if data[col].dtypes in ['int64', 'float64']]

print(">> Categorical Features: ", CatFeatures)
print("\n>> Numeric Features: ", NumFeatures)
```

```
>> Categorical Features:  ['customerID', 'gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetS
ervice', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contra
ct', 'PaperlessBilling', 'PaymentMethod', 'Churn']

>> Numeric Features:  ['SeniorCitizen', 'tenure', 'MonthlyCharges', 'TotalCharges']
```

In [9]:
```
print(">> The unique values of each categorical feature: ")
for feature in CatFeatures:
  print('{}:\t{}'.format(feature, data[feature].unique()))
```

```
>> The unique values of each categorical feature:
customerID:     ['7590-VHVEG' '5575-GNVDE' '3668-QPYBK' ... '4801-JZAZL' '8361-LTMKD'
 '3186-AJIEK']
gender: ['Female' 'Male']
Partner:        ['Yes' 'No']
Dependents:     ['No' 'Yes']
PhoneService:   ['No' 'Yes']
MultipleLines:  ['No phone service' 'No' 'Yes']
InternetService:        ['DSL' 'Fiber optic' 'No']
OnlineSecurity: ['No' 'Yes' 'No internet service']
OnlineBackup:   ['Yes' 'No' 'No internet service']
DeviceProtection:       ['No' 'Yes' 'No internet service']
TechSupport:    ['No' 'Yes' 'No internet service']
StreamingTV:    ['No' 'Yes' 'No internet service']
StreamingMovies:        ['No' 'Yes' 'No internet service']
Contract:       ['Month-to-month' 'One year' 'Two year']
PaperlessBilling:       ['Yes' 'No']
PaymentMethod:  ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
Churn:  ['No' 'Yes']
```

In [10]:
```
# describe data by type of features
print(">> All statistics of Numeric Features: ")
display(data[NumFeatures].describe(include='all'))
print("\n>> All statistics of Categorical Features: ")
display(data[CatFeatures].describe(include='all'))
```

```
>> All statistics of Numeric Features:
```

| | SeniorCitizen | tenure | MonthlyCharges | TotalCharges |
|---|---|---|---|---|
| **count** | 7043.000000 | 7043.000000 | 7043.000000 | 7032.000000 |
| **mean** | 0.162147 | 32.371149 | 64.761692 | 2283.300441 |
| **std** | 0.368612 | 24.559481 | 30.090047 | 2266.771362 |
| **min** | 0.000000 | 0.000000 | 18.250000 | 18.800000 |
| **25%** | 0.000000 | 9.000000 | 35.500000 | 401.450000 |
| **50%** | 0.000000 | 29.000000 | 70.350000 | 1397.475000 |
| **75%** | 0.000000 | 55.000000 | 89.850000 | 3794.737500 |
| **max** | 1.000000 | 72.000000 | 118.750000 | 8684.800000 |

```
>> All statistics of Categorical Features:
```

| | customerID | gender | Partner | Dependents | PhoneService | MultipleLines | InternetService | OnlineSecurity | OnlineBackup | DeviceProtection | Tec |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | |
| **unique** | 7043 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | |
| **top** | 7590-VHVEG | Male | No | No | Yes | No | Fiber optic | No | No | No | |
| **freq** | 1 | 3555 | 3641 | 4933 | 6361 | 3390 | 3096 | 3498 | 3088 | 3095 | |

In [11]:
```
# check missing value
print(">> Check if any column has missing value: ")
display(data.isnull().sum())
print("\n* Comment: There are 11 customers that have no the infomation about TotalCharges feature.")
```

```
>> Check if any column has missing value:

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       11
Churn               0
dtype: int64


* Comment: There are 11 customers that have no the infomation about TotalCharges feature.
```

In [12]:
```
# Check duplicate values
print(">> Number of duplicate values existing in data: ", data.duplicated().sum())
```

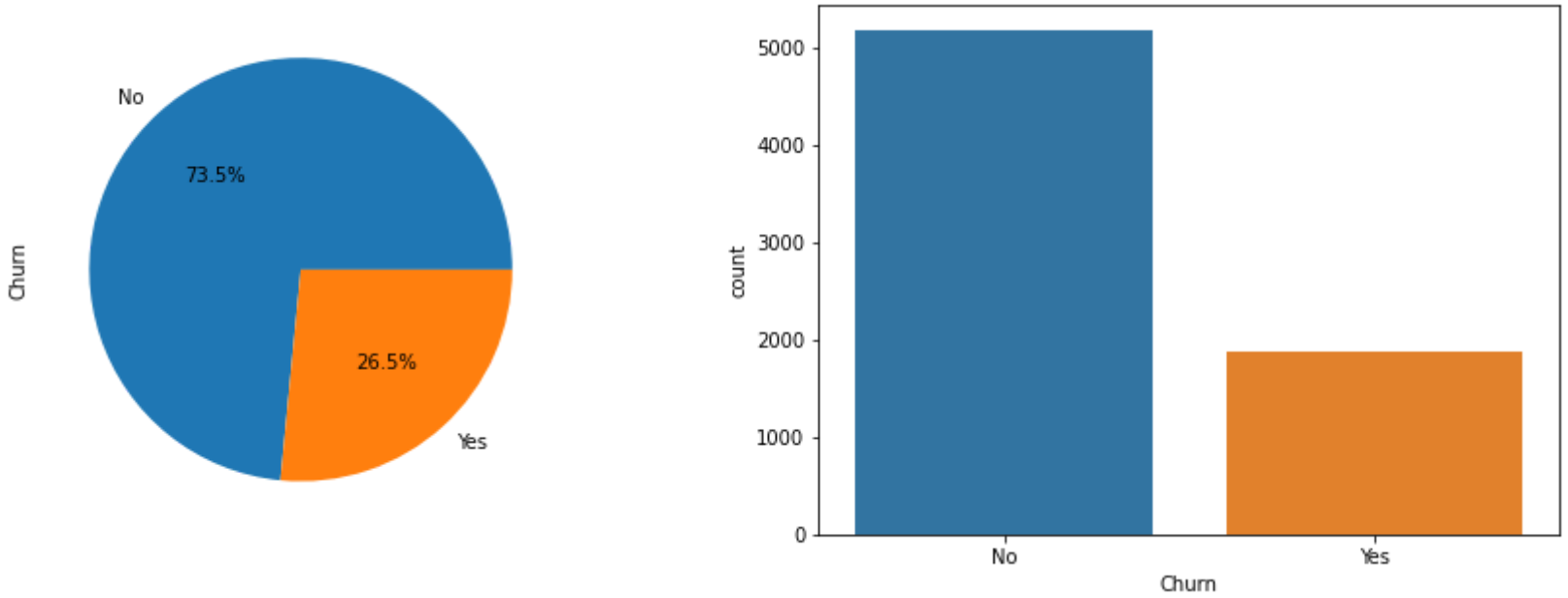
```
>> Number of duplicate values existing in data:  0
```

## Visualize Data

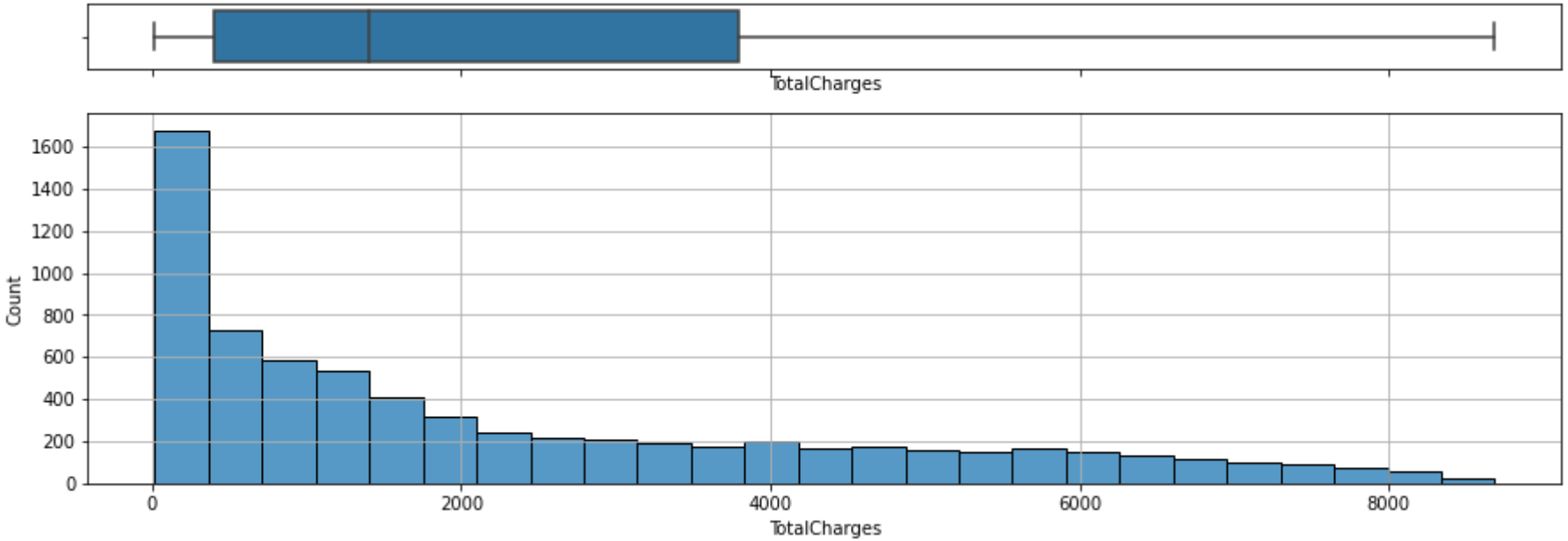
In [13]:
```
print(">> Display the quanlity of each values in Churn Feature:")

fig, ax = plt.subplots(1, 2, figsize=(15,5))
data['Churn'].value_counts().plot.pie(autopct='%1.1f%%', ax=ax[0])
sns.countplot(data['Churn'], ax=ax[1])
plt.show()
```

```
>> Display the quanlity of each values in Churn Feature:
```



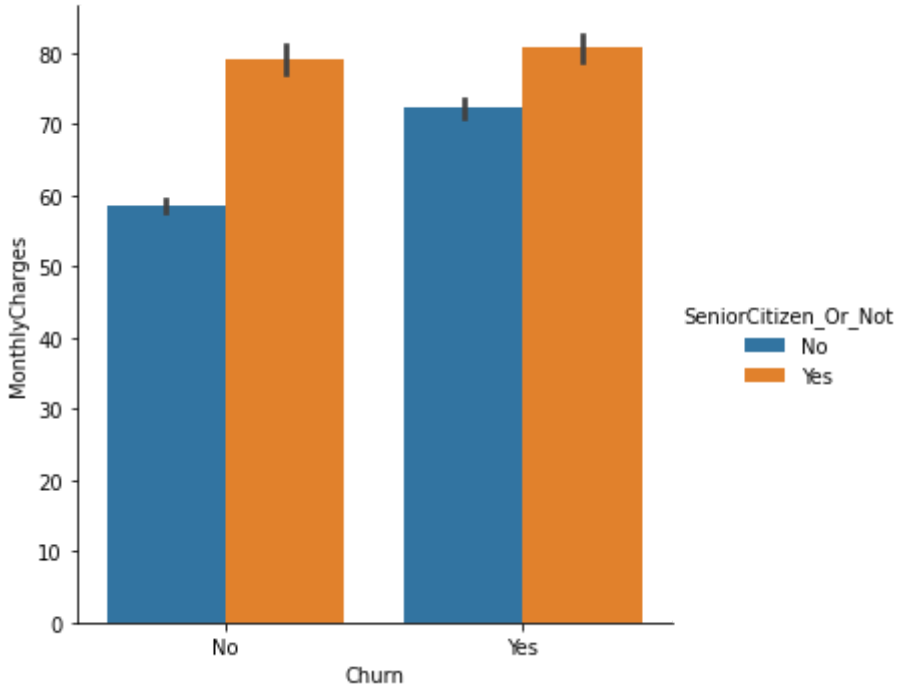
In [14]:
```
fig, ax = plt.subplots(2, sharex=True, gridspec_kw={'height_ratios': (.15, .85)})
fig.set_figheight(5)
fig.set_figwidth(15)
sns.boxplot(data['TotalCharges'], ax=ax[0])
sns.histplot(data=data, x='TotalCharges', ax=ax[1])
plt.grid()
plt.show()
```



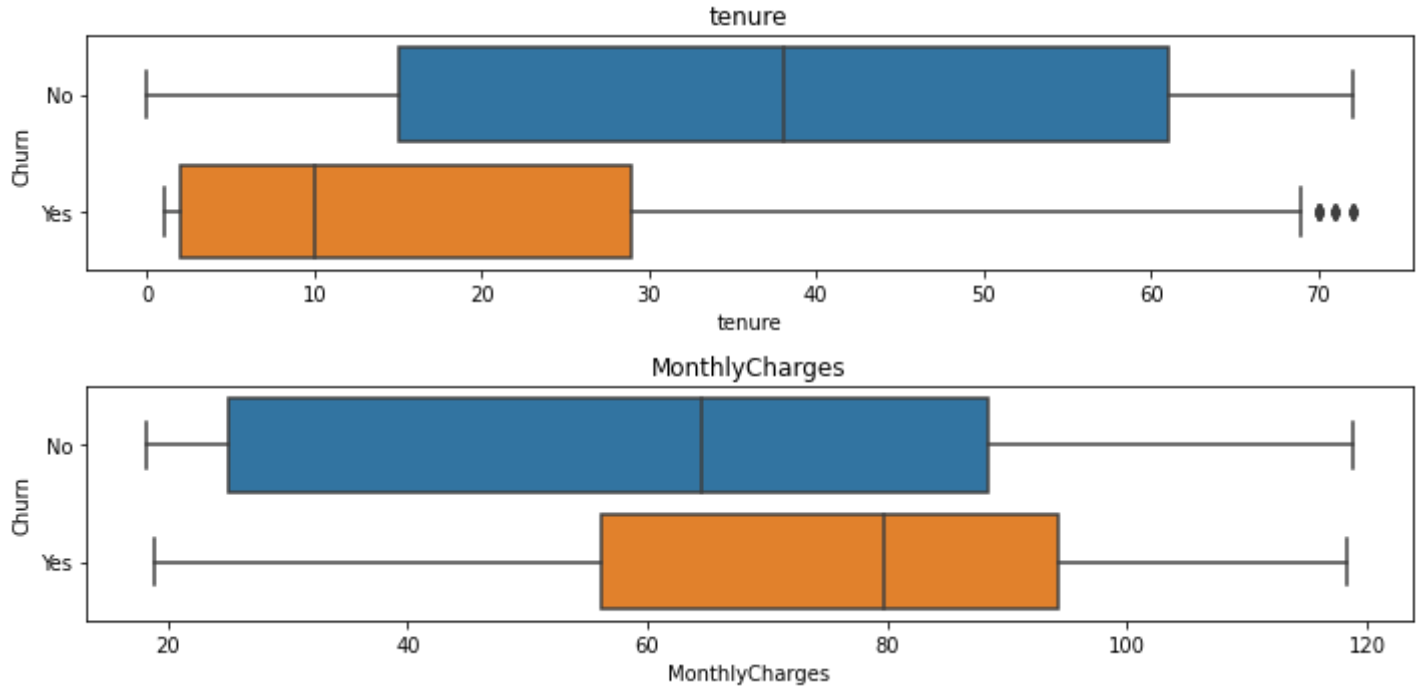
In [15]:
```
# map 0/1 to No/Yes in SeniorCitizen feature
MapDict = {0: 'No', 1: 'Yes'}
data['SeniorCitizen_Or_Not'] = data['SeniorCitizen'].map(MapDict)

plt.figure(figsize=(10,5))
sns.catplot(x='Churn', y='MonthlyCharges', hue='SeniorCitizen_Or_Not', kind='bar', data=data)
plt.show()
```

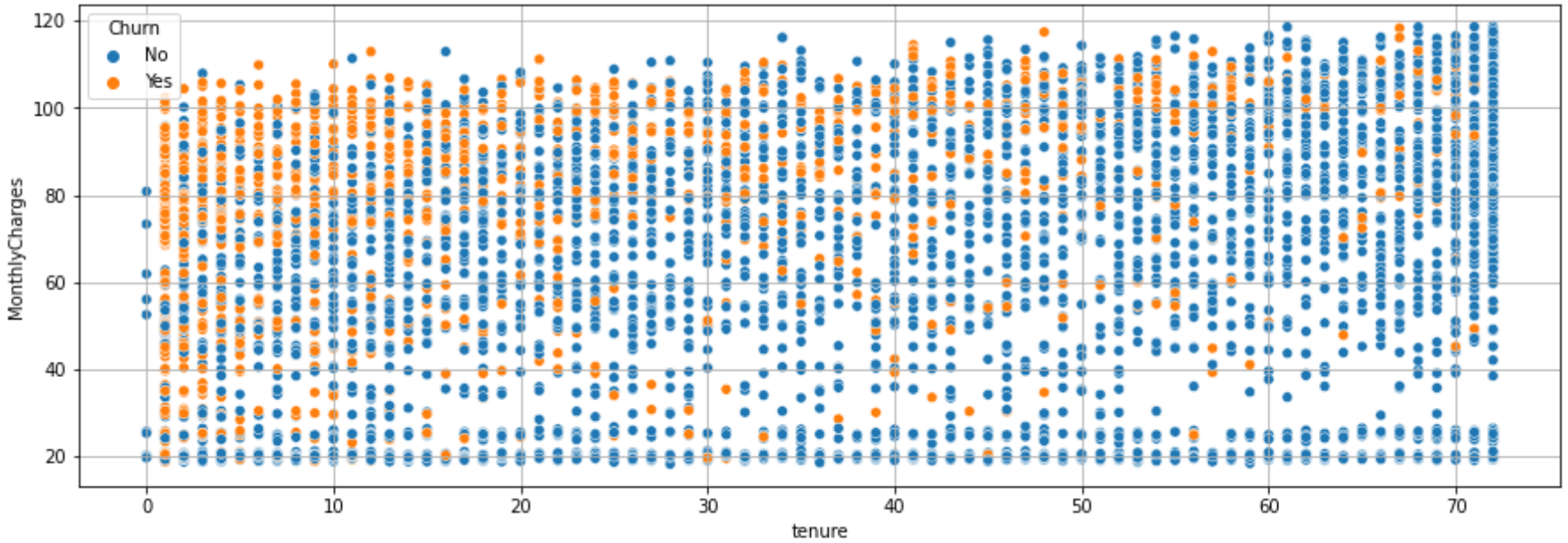
```
<Figure size 720x360 with 0 Axes>
```



In [16]:
```
feature1 = 'tenure'
feature2 = 'MonthlyCharges'
fig, ax = plt.subplots(2, 1, figsize=(10,5))
sns.boxplot(y='Churn', x=feature1, data=data, ax=ax[0])
sns.boxplot(y='Churn', x=feature2, data=data, ax=ax[1])
ax[0].set_title(feature1)
ax[1].set_title(feature2)
plt.tight_layout()
plt.show()
```



In [17]:
```
plt.figure(figsize=(15,5))
feature_x = "tenure"
feature_y = "MonthlyCharges"
feature_hue = "Churn"
sns.scatterplot(x=feature_x, y=feature_y, hue=feature_hue, data=data, legend='full')
plt.grid()
plt.show()
```




## Split Data into many Data

In [18]:
```
print(">> The table data is about services that each customer has signed up for: ")
services = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'T
data_services = data[services].copy()
display(data_services.head())

print("\n>> The table data is about customers account information: ")
customer_info = ['customerID', 'tenure', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharge
data_customer_info = data[customer_info].copy()
display(data_customer_info.head())

print("\n>> The table data is about demographic info about customers: ")
demographic_info = list(set(data.columns) - set(services) - set(customer_info))
data_demographic_info = data[demographic_info].copy()
display(data_demographic_info.head())
```

```
>> The table data is about services that each customer has signed up for:
```

| | 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 |

```
>> The table data is about customers account information:
```

| | customerID | tenure | Contract | PaperlessBilling | PaymentMethod | MonthlyCharges | TotalCharges |
|---|---|---|---|---|---|---|---|
| **0** | 7590-VHVEG | 1 | Month-to-month | Yes | Electronic check | 29.85 | 29.85 |
| **1** | 5575-GNVDE | 34 | One year | No | Mailed check | 56.95 | 1889.50 |
| **2** | 3668-QPYBK | 2 | Month-to-month | Yes | Mailed check | 53.85 | 108.15 |

| | customerID | tenure | Contract | PaperlessBilling | PaymentMethod | MonthlyCharges | TotalCharges |
|---|---|---|---|---|---|---|---|
| **3** | 7795-CFOCW | 45 | One year | No | Bank transfer (automatic) | 42.30 | 1840.75 |
| **4** | 9237-HQITU | 2 | Month-to-month | Yes | Electronic check | 70.70 | 151.65 |

>> The table data is about demographic info about customers:

| | Partner | gender | SeniorCitizen | Dependents | SeniorCitizen_Or_Not | Churn |
|---|---|---|---|---|---|---|
| **0** | Yes | Female | 0 | No | No | No |
| **1** | No | Male | 0 | No | No | No |
| **2** | No | Male | 0 | No | No | Yes |
| **3** | No | Male | 0 | No | No | No |
| **4** | No | Female | 0 | No | No | Yes |

## Filter Data By Condition

**Thống kê số lượng khách hàng sử dụng các dịch vụ theo giới tính.**

In [19]:
```
service_features = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtect
services_by_gender = data[['gender']+service_features].copy()
services_by_gender = services_by_gender.groupby('gender')[service_features].count()
display(services_by_gender)
```

| gender | PhoneService | MultipleLines | InternetService | OnlineSecurity | OnlineBackup | DeviceProtection | TechSupport | StreamingTV | StreamingMovies |
|---|---|---|---|---|---|---|---|---|---|
| **Female** | 3488 | 3488 | 3488 | 3488 | 3488 | 3488 | 3488 | 3488 | 3488 |
| **Male** | 3555 | 3555 | 3555 | 3555 | 3555 | 3555 | 3555 | 3555 | 3555 |

**Thống kê các khách hàng là người cao tuổi nhưng vẫn tiếp tục sử dụng dịch vụ với phí hàng tháng lớn hơn 100.**

In [20]:
```
value1, value2, value3 = 100, 'Yes', 1
data_more100_MonthlyCharges = data.query("`MonthlyCharges` > @value1 and `Churn` == @value2 and `SeniorCitizen` == @valu
display(data_more100_MonthlyCharges.head())
```

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | ... | TechSupport |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **261** | 3606-TWKGI | Male | 1 | No | No | 13 | Yes | Yes | Fiber optic | No | ... | No |
| **609** | 3419-SNJJD | Female | 1 | Yes | No | 65 | Yes | Yes | Fiber optic | Yes | ... | No |
| **630** | 5099-BAILX | Male | 1 | Yes | Yes | 43 | Yes | Yes | Fiber optic | No | ... | Yes |
| **638** | 4913-EHYUI | Male | 1 | Yes | Yes | 56 | Yes | Yes | Fiber optic | Yes | ... | No |
| **785** | 0691-IFBQW | Female | 1 | No | No | 46 | Yes | Yes | Fiber optic | Yes | ... | No |

5 rows × 22 columns

**Thống kê tổng chi phí và chi phí hàng tháng theo loại hình hợp đồng.**

In [21]:
```
charges = ['MonthlyCharges', 'TotalCharges']
charges_by_contract = data[['Contract']+charges].copy()
charges_by_contract = charges_by_contract.groupby('Contract')[charges].count()
display(charges_by_contract)
```

| Contract | MonthlyCharges | TotalCharges |
|---|---|---|
| **Month-to-month** | 3875 | 3875 |
| **One year** | 1473 | 1472 |
| **Two year** | 1695 | 1685 |

**Thống kê tổng số dịch vụ đã sử dụng của từng khách hàng.**

In [22]:
```
encode_data = data.copy()
replace_map = {"No phone service": "No", "No internet service": "No", "DSL": "Yes", "Fiber optic": "Yes"}
encode_data.replace(replace_map, inplace=True)
for ft in service_features:
  encode_data[ft].replace({'Yes': 1, "No": 0}, inplace=True)

data['TotalServices'] = encode_data[service_features].apply(lambda features: np.sum(features), axis=1)
display(data.head())
```

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | ... | StreamingTV | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **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 | ... | No | |
| **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 | ... | No | |
| **4** | 9237-HQITU | Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | ... | No | |

5 rows × 23 columns