In [1]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import metrics
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
data = pd.read_csv("DMV dataset/Telcom_Customer_Churn.csv")
print(data.index)
```

```
RangeIndex(start=0, stop=7043, step=1)
```

In [3]:

```
data
```

Out[3]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
0	7590-VHVEG	Female	0	Yes	No	1	No
1	5575-GNVDE	Male	0	No	No	34	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes
3	7795-CFOCW	Male	0	No	No	45	No
4	9237-HQITU	Female	0	No	No	2	Yes
...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes
7040	4801-JZAZL	Female	0	Yes	Yes	11	No
7041	8361-LTMKD	Male	1	Yes	No	4	Yes
7042	3186-AJIEK	Male	0	No	No	66	Yes

7043 rows × 21 columns

In [4]:

```
data.columns
```

Out[4]:
```
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 [5]:
```
data.shape
```

Out[5]:
```
(7043, 21)
```

In [6]:
```
data.head()
```

Out[6]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mul
0	7590-VHVEG	Female	0	Yes	No	1	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	
3	7795-CFOCW	Male	0	No	No	45	No	
4	9237-HQITU	Female	0	No	No	2	Yes	

5 rows × 21 columns

In [7]:
```
data.tail()
```

Out[7]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes
7040	4801-JZAZL	Female	0	Yes	Yes	11	No
7041	8361-LTMKD	Male	1	Yes	No	4	Yes
7042	3186-AJIEK	Male	0	No	No	66	Yes

5 rows × 21 columns

In [8]:
```
data.nunique()
```

Out[8]:
```
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
```

In [9]:
```
data.isna().sum()
```

Out[9]:
```
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 [10]:
```
data.isnull().sum()
```

Out[10]:

```
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 [11]:

```
print(len(data))
```

```
7043
```

In [12]:

```
data_cleaned = data.drop_duplicates()
```

In [13]:

```
data_cleaned = data.drop_duplicates()
```

In [14]:

```
data.describe()
```

Out[14]:

	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

In [15]:

```
unique, counts = np.unique(data['tenure'], return_counts=True)
print(unique, counts)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
 72] [ 11 613 238 200 176 133 110 131 123 119 116  99 117 109  76  99  80  87
  97  73  71  63  90  85  94  79  79  72  57  72  72  65  69  64  65  88
  50  65  59  56  64  70  65  65  51  61  74  68  64  66  68  68  80  70
  68  64  80  65  67  60  76  76  70  72  80  76  89  98 100  95 119 170
 362]
```

In [16]:
```
unique, counts = np.unique(data['MonthlyCharges'], return_counts=True)
print(unique, counts)
```

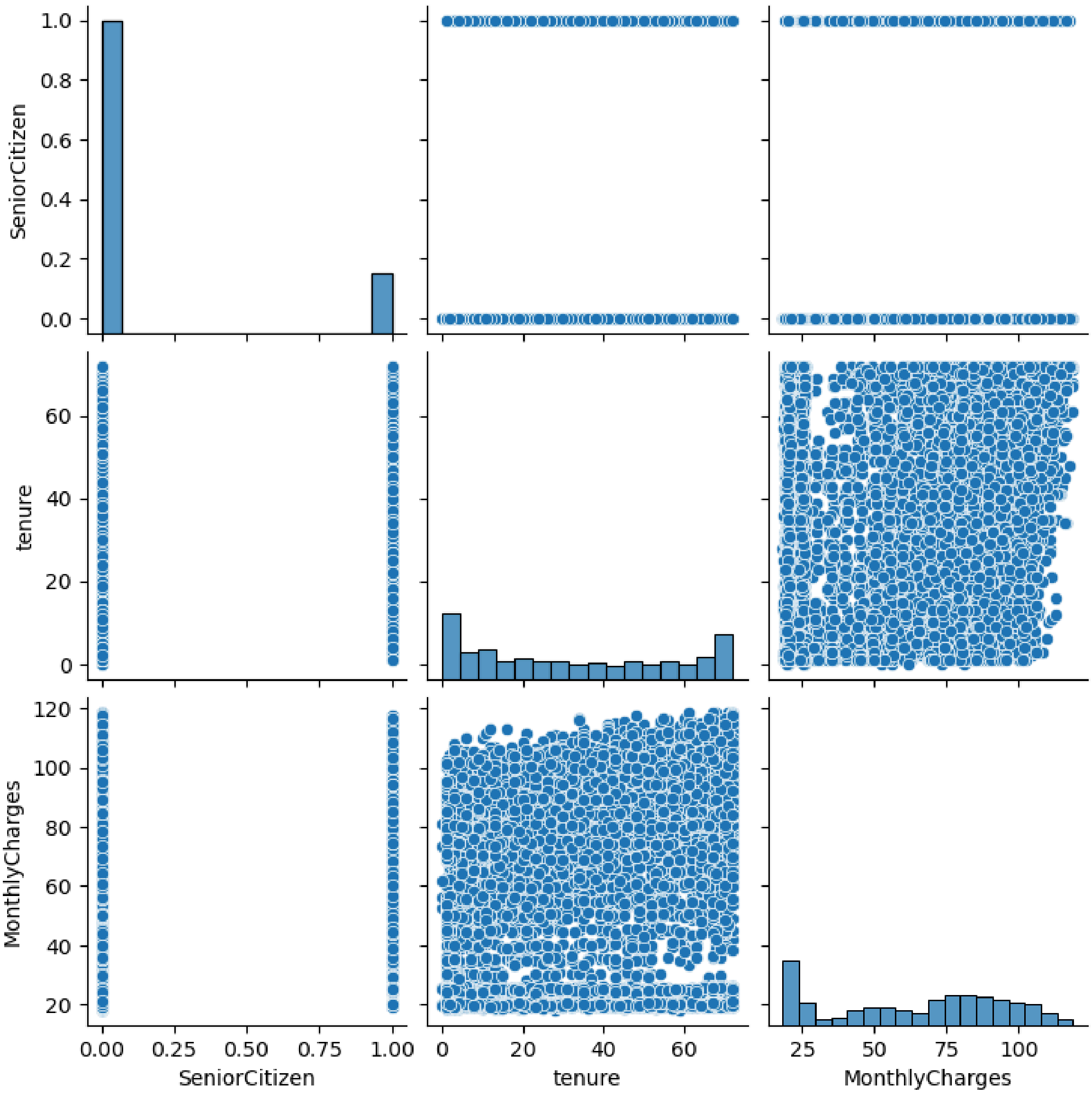
```
[ 18.25  18.4   18.55 ... 118.6  118.65 118.75] [1 1 1 ... 2 1 1]
```

In [17]:
```
unique, counts = np.unique(data['TotalCharges'], return_counts=True)
print(unique, counts)
```

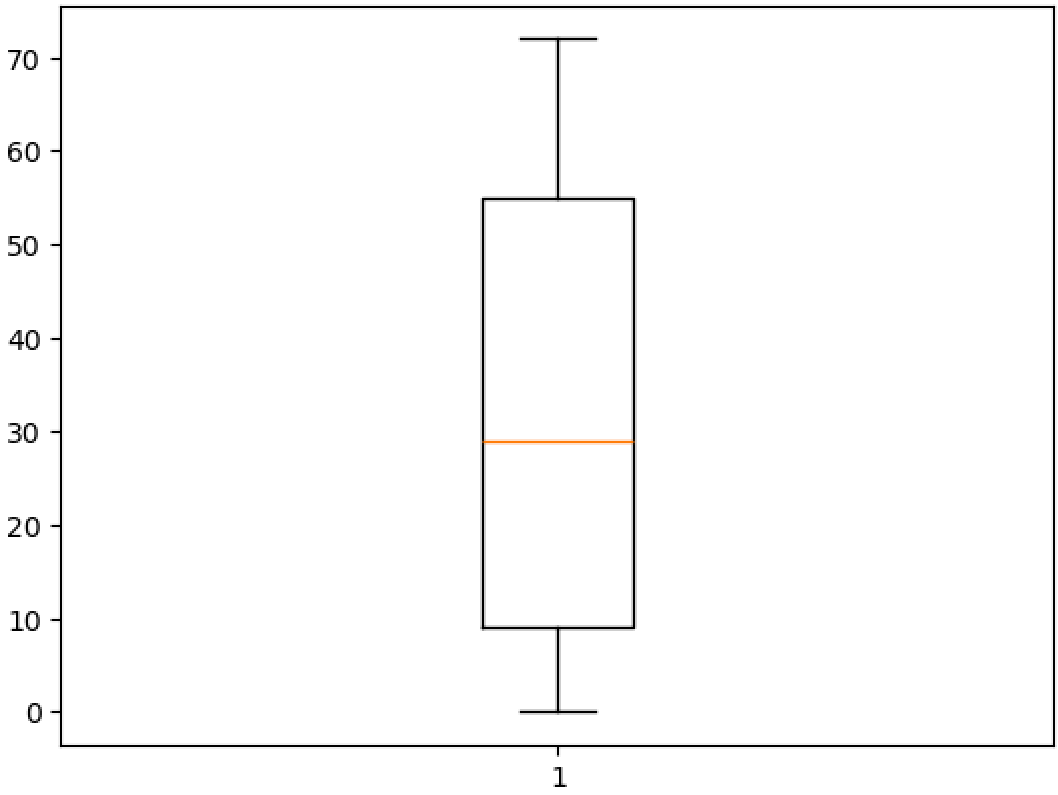
```
[' ' '100.2' '100.25' ... '999.45' '999.8' '999.9'] [11  1  1 ...  1  1  1]
```

In [18]:
```
sns.pairplot(data)
```

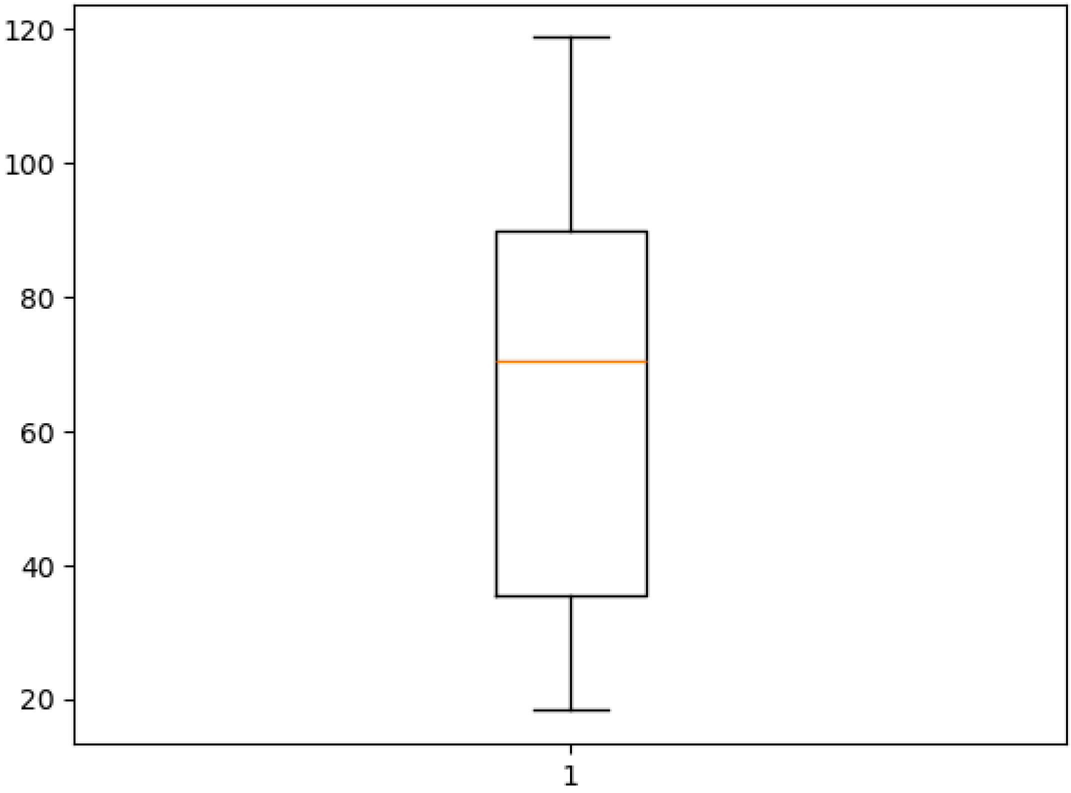
Out[18]:
```
<seaborn.axisgrid.PairGrid at 0x243cef77c50>
```



In [19]:
```
plt.boxplot(data['tenure'])
plt.show()
```



In [20]:
```
plt.boxplot(data['MonthlyCharges'])
plt.show()
```



In [23]:
```
# data.to_csv("DMV Dataset/Cleaned_Telecom_Customer_Churn.csv", index=False)
```