Customer Data Analysis (Task 2)

Revision (2)

In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:

```
data = pd.read_csv('dataset_nan (1).csv')
```

In [8]:

```
data.head()
```

Out[8]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	7590-VHVEG	Female	NaN	Yes	No	NaN	No	No phone service
1	5575-GNVDE	Male	NaN	No	No	34.0	Yes	No
2	3668-QPYBK	Male	NaN	No	No	2.0	Yes	No
3	7795-CFOCW	Male	NaN	No	No	45.0	No	No phone service
4	9237-HQITU	Female	NaN	No	No	2.0	Yes	No

5 rows × 21 columns

In [9]:

```
data.info()
```

```
<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     7000 non-null   float64
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            6896 non-null   float64
 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   float64
 20  Churn             7043 non-null   object 
dtypes: float64(4), object(17)
memory usage: 1.1+ MB
```

In [10]:

```
data.isnull().sum()
```

Out[10]:

```
customerID            0
gender                0
SeniorCitizen        43
Partner               0
Dependents            0
tenure              147
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 [12]:

```
data.shape
```

Out[12]:

```
(7043, 21)
```

In [28]:

```
data.describe()
```

Out[28]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7000.000000	6896.000000	7043.000000	7043.000000
mean	0.163143	33.041473	64.761692	2283.300440
std	0.369522	24.382260	30.090047	2265.000258
min	0.000000	1.000000	18.250000	18.800000
25%	0.000000	10.000000	35.500000	402.225000
50%	0.000000	30.000000	70.350000	1400.550000
75%	0.000000	56.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

In [43]:

```
data['TotalCharges'].value_counts().sort_values().tail(10)
```

Out[43]:

```
19.45       6
20.15       6
19.55       7
45.30       7
19.90       8
19.65       8
20.05       8
19.75       9
20.20      11
2283.30    12
Name: TotalCharges, dtype: int64
```

In [46]:

```
data['MonthlyCharges'].value_counts().sort_values().tail(20)
```

Out[46]:

```
20.40    30
20.45    31
19.40    31
19.50    32
20.20    35
20.10    37
19.60    37
19.80    38
20.35    38
20.25    39
19.75    39
20.15    40
19.55    40
19.65    43
19.70    43
20.00    43
19.90    44
19.95    44
19.85    45
20.05    61
Name: MonthlyCharges, dtype: int64
```

In [49]:

```python
data['tenure'].value_counts().sort_values().tail(20)
```

Out[49]:

```
18.0     97
67.0     98
11.0     99
15.0     99
68.0    100
13.0    109
6.0     110
10.0    116
12.0    117
70.0    119
9.0     119
8.0     123
7.0     131
5.0     133
71.0    170
4.0     176
3.0     200
2.0     238
72.0    362
1.0     477
Name: tenure, dtype: int64
```

In [29]:

```python
data['Contract'].value_counts()
```

Out[29]:

```
Month-to-month    3875
Two year          1695
One year          1473
Name: Contract, dtype: int64
```

In [30]:

```python
data.groupby(['Contract']).mean()
```

Out[30]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
Contract				
Month-to-month	0.209665	18.637457	66.398490	1369.254581
One year	0.129781	42.185286	65.048608	3034.172980
Two year	0.085951	57.071810	60.770413	3720.405133

In [50]:

```
data.groupby(['Contract']).std()
```

Out[50]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
Contract				
Month-to-month	0.407122	17.701433	26.926599	1613.879008
One year	0.336178	18.915116	31.840539	2229.058541
Two year	0.280375	17.729130	34.678865	2566.043071

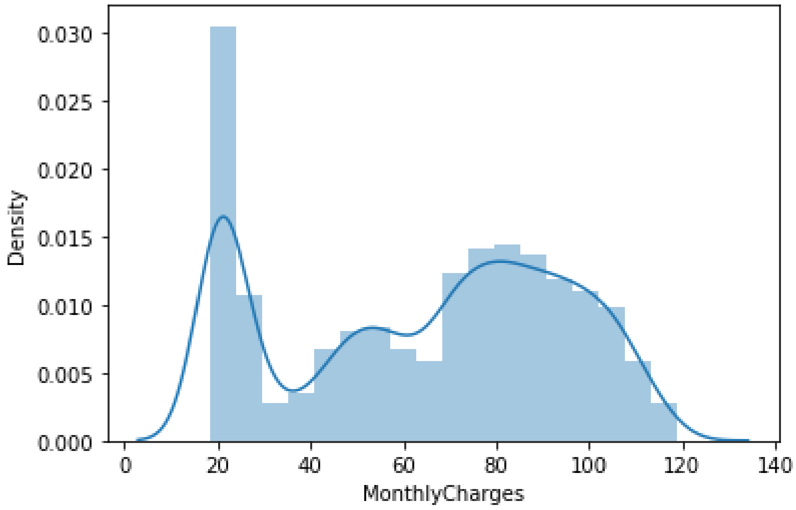
In [51]:

```
sns.distplot(data['MonthlyCharges'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[51]:

```
<AxesSubplot:xlabel='MonthlyCharges', ylabel='Density'>
```



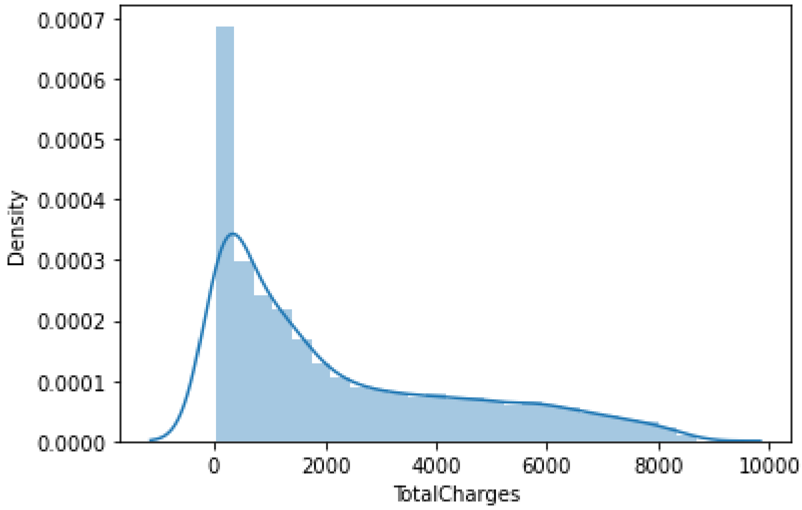
In [52]:

```
sns.distplot(data['TotalCharges'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[52]:

```
<AxesSubplot:xlabel='TotalCharges', ylabel='Density'>
```



In [58]:

```
x=np.log(data['TotalCharges'])
x
```

Out[58]:

```
0       3.396185
1       7.544068
2       4.683519
3       7.517928
4       5.021575
          ...
7038    7.596141
7039    8.904209
7040    5.847739
7041    5.725544
7042    8.831201
Name: TotalCharges, Length: 7043, dtype: float64
```

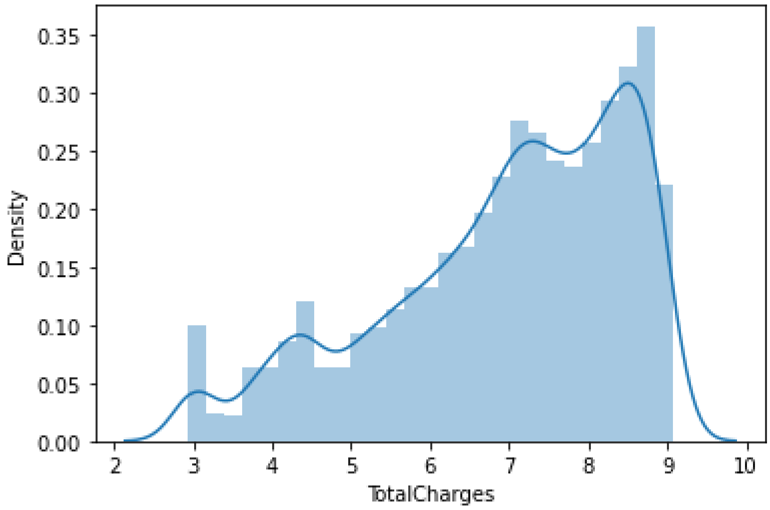
In [59]:

```
sns.distplot(x)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[59]:

```
<AxesSubplot:xlabel='TotalCharges', ylabel='Density'>
```



In [61]:

```
data.corr()
```

Out[61]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
SeniorCitizen	1.000000	0.013521	0.221101	0.102831
tenure	0.013521	1.000000	0.238635	0.822171
MonthlyCharges	0.221101	0.238635	1.000000	0.650468
TotalCharges	0.102831	0.822171	0.650468	1.000000

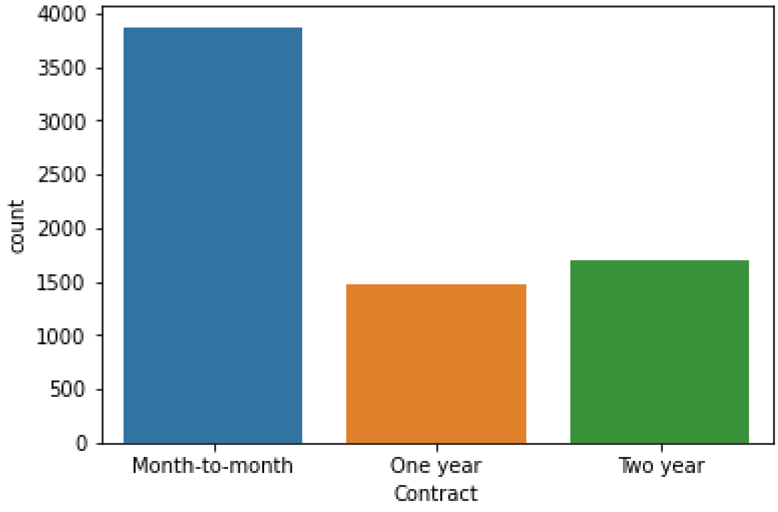
In [62]:

```
sns.countplot(data['Contract'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[62]:

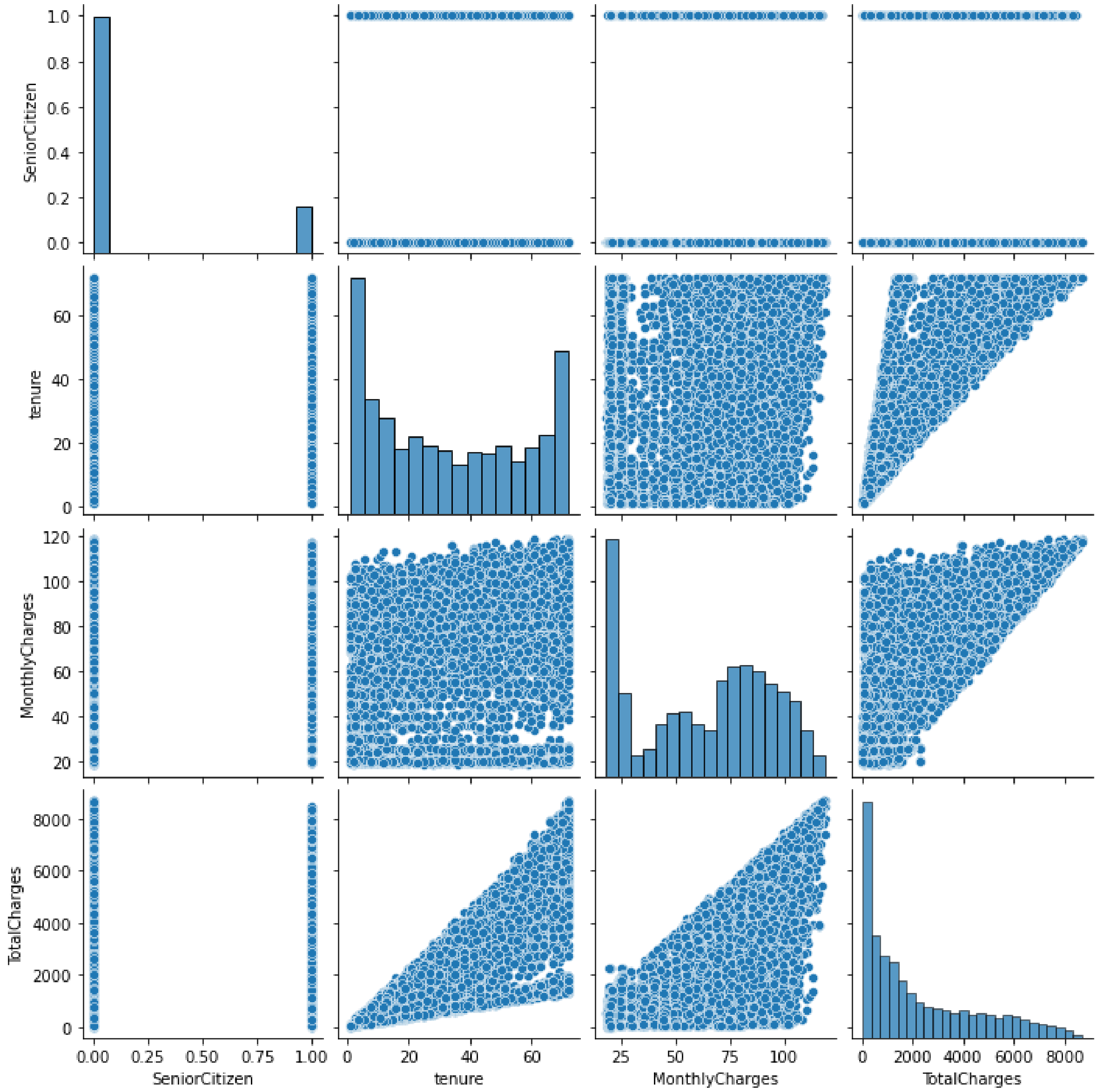
```
<AxesSubplot:xlabel='Contract', ylabel='count'>
```



In [74]:

```
sns.pairplot(data, size=2.5)
plt.show();
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\axisgrid.py:1912: UserWar
ning: The `size` parameter has been renamed to `height`; please update your
code.
  warnings.warn(msg, UserWarning)
```



In [85]:

```
df=data.groupby(['gender','Contract','StreamingTV']).sum()
df
```

Out[85]:

			SeniorCitizen	tenure	MonthlyCharges	TotalCharges
gender	**Contract**	**StreamingTV**				
Female	**Month-to-month**	**No**	218.0	16492.0	66951.55	1154480.40
		No internet service	7.0	2592.0	5225.85	54220.25
		Yes	174.0	15462.0	56128.90	1397275.15
	One year	**No**	23.0	9637.0	14964.45	663014.05
		No internet service	6.0	5094.0	3490.20	107882.60
		Yes	72.0	15744.0	29537.65	1477329.25
	Two year	**No**	14.0	7847.0	9275.40	544931.60
		No internet service	8.0	15473.0	7053.65	344740.85
		Yes	46.0	24073.0	34804.75	2219896.55
Male	**Month-to-month**	**No**	218.0	16727.0	66348.05	1166864.15
		No internet service	5.0	2152.0	5468.75	44952.50
		Yes	185.0	16335.0	57171.05	1488069.05
	One year	**No**	31.0	10112.0	15150.35	691204.65
		No internet service	10.0	5676.0	4088.10	121039.00
		Yes	48.0	15665.0	28585.85	1408867.25
	Two year	**No**	14.0	9316.0	11049.60	656548.95
		No internet service	16.0	15587.0	6840.30	351999.50
		Yes	47.0	23870.0	33982.15	2187969.25

In [89]:

```
df=data.groupby(['gender','Contract','InternetService']).mean()
df
```

Out[89]:

```
                                   SeniorCitizen     tenure  MonthlyCharges  TotalCharges
gender  Contract  InternetService
Female  Month-to-             DSL       0.129195  16.988034       50.615698    885.677990
           month
                      Fiber optic       0.297170  21.047801       86.794564   1891.825117
                               No       0.027344  10.800000       20.413477    211.797852
        One year              DSL       0.083032  39.660650       62.285612   2494.652158
                      Fiber optic       0.263736  52.728938       99.584982   5299.743590
                               No       0.036364  30.502994       20.899401    646.003593
        Two year              DSL       0.075410  59.378289       70.843344   4264.237987
                      Fiber optic       0.175355  65.112676      104.508920   6814.285681
                               No       0.024691  47.904025       21.770525   1064.014969
  Male  Month-to-             DSL       0.133117  15.736395       49.835427    776.099919
           month
                      Fiber optic       0.304265  22.937380       87.249105   2048.044439
                               No       0.018797   8.747967       20.405784    167.733209
        One year              DSL       0.093750  41.407534       60.550514   2556.429966
                      Fiber optic       0.196226  51.451128       97.952820   5088.700564
                               No       0.051020  29.409326       20.751777    614.411168
        Two year              DSL       0.084639  59.846395       70.096875   4263.484375
                      Fiber optic       0.158140  65.254630      104.633102   6852.792593
                               No       0.051118  50.280645       21.784395   1121.017516
```

In [91]:

```
df=data.groupby(['gender','Contract','PaymentMethod']).mean()
df
```

Out[91]:

			SeniorCitizen	tenure	MonthlyCharges	TotalCharges
gender	**Contract**	**PaymentMethod**				
Female	**Month-to-month**	**Bank transfer (automatic)**	0.172078	25.091803	70.094355	1883.260968
		Credit card (automatic)	0.197761	24.525926	68.736481	1845.695556
		Electronic check	0.278261	17.765816	74.510572	1408.485491
		Mailed check	0.088942	10.794937	45.327512	521.916388
	One year	**Bank transfer (automatic)**	0.160622	46.165803	67.863472	3311.482383
		Credit card (automatic)	0.147208	46.469388	71.164213	3553.013198
		Electronic check	0.203704	46.814815	79.819136	3973.195370
		Mailed check	0.049080	29.355422	47.859036	1599.449096
	Two year	**Bank transfer (automatic)**	0.098592	61.904930	64.880877	4179.465088
		Credit card (automatic)	0.099644	59.542254	64.407368	4005.764561
		Electronic check	0.086420	60.493827	81.613580	5082.524074
		Mailed check	0.025773	41.895288	39.566753	1881.928351
Male	**Month-to-month**	**Bank transfer (automatic)**	0.212996	25.480000	67.964337	1892.105376
		Credit card (automatic)	0.210332	23.640741	66.632784	1766.514286
		Electronic check	0.279956	19.366404	75.470477	1534.443933
		Mailed check	0.074310	10.559091	46.685895	575.659263
	One year	**Bank transfer (automatic)**	0.174359	45.873737	67.114394	3314.492424
		Credit card (automatic)	0.095000	44.731343	64.844030	3165.395771
		Electronic check	0.162162	45.864130	78.461351	3748.259730
		Mailed check	0.035503	29.404762	40.857602	1375.256140
	Two year	**Bank transfer (automatic)**	0.100719	61.737410	65.149821	4138.369355
		Credit card (automatic)	0.119048	60.263514	64.436318	4106.247973
		Electronic check	0.126437	61.689655	86.870115	5450.290805
		Mailed check	0.015957	45.679348	37.576596	1873.872074

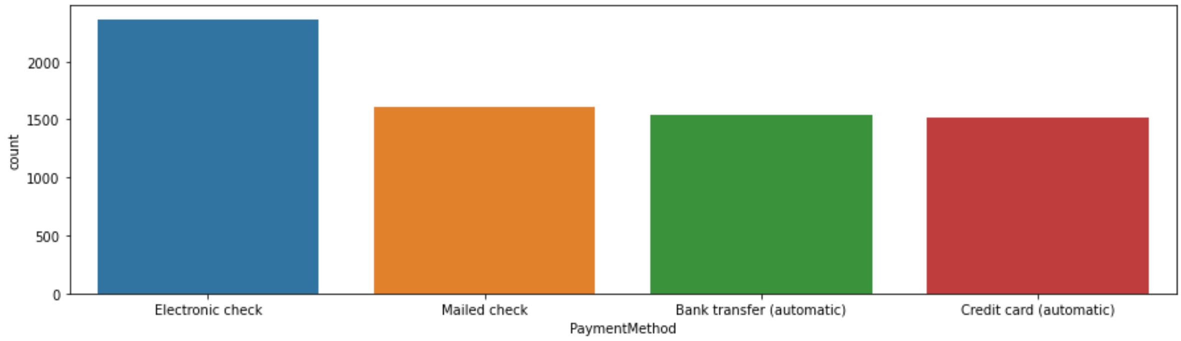
In [100]:

```
plt.figure(figsize=(15,4))
sns.countplot(data['PaymentMethod'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

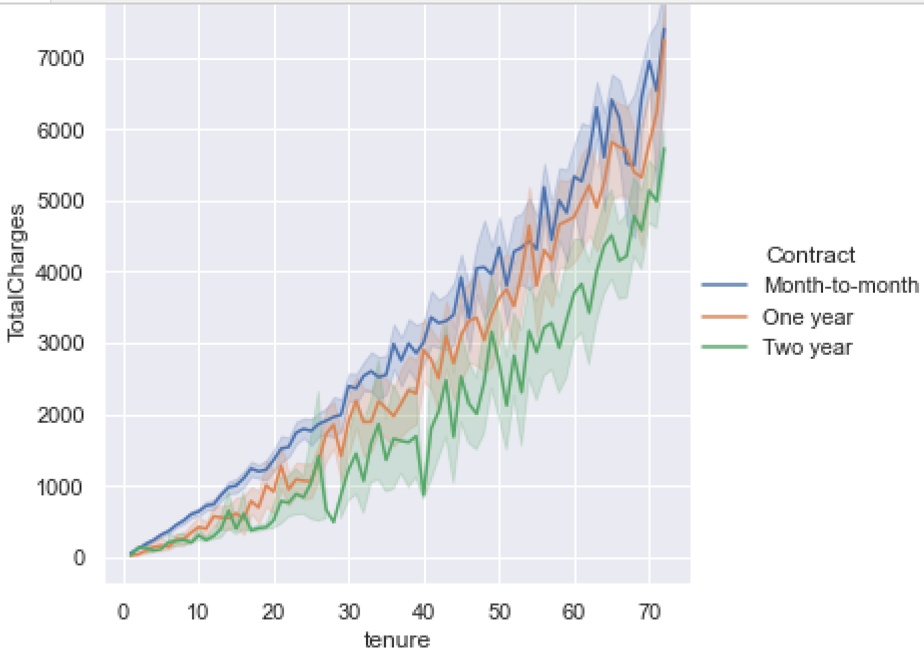
Out[100]:

```
<AxesSubplot:xlabel='PaymentMethod', ylabel='count'>
```



In [23]:

```
sns.relplot(hue=data['Contract'], x=data['tenure'], y=data['TotalCharges'], kind="line'
```



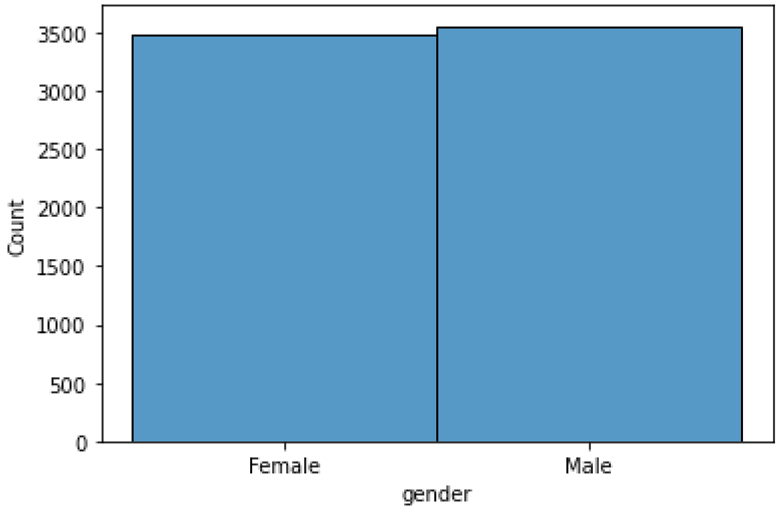
The contract that company make more profite is Month-to-month contract

In [106]:

```python
sns.histplot(data['gender'])
```

Out[106]:

```
<AxesSubplot:xlabel='gender', ylabel='Count'>
```

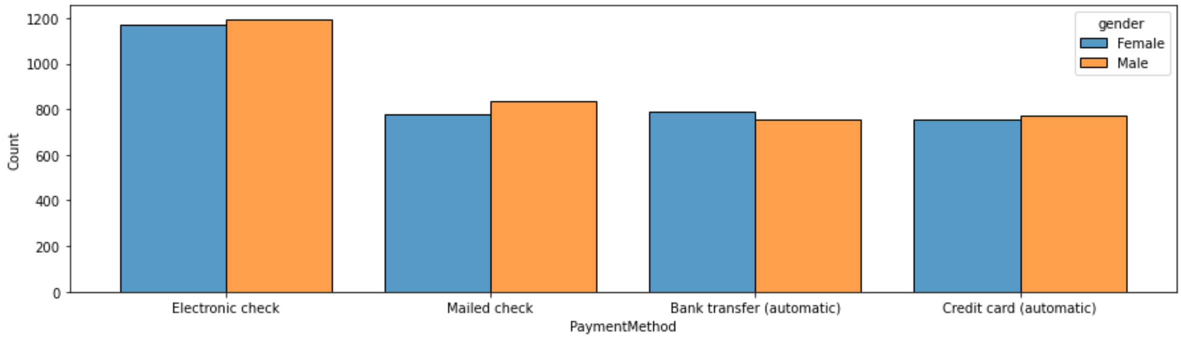


In [109]:

```python
plt.figure(figsize=(15,4))
sns.histplot(data=data, x=data['PaymentMethod'], hue=data['gender'], multiple="dodge",
```

Out[109]:

```
<AxesSubplot:xlabel='PaymentMethod', ylabel='Count'>
```

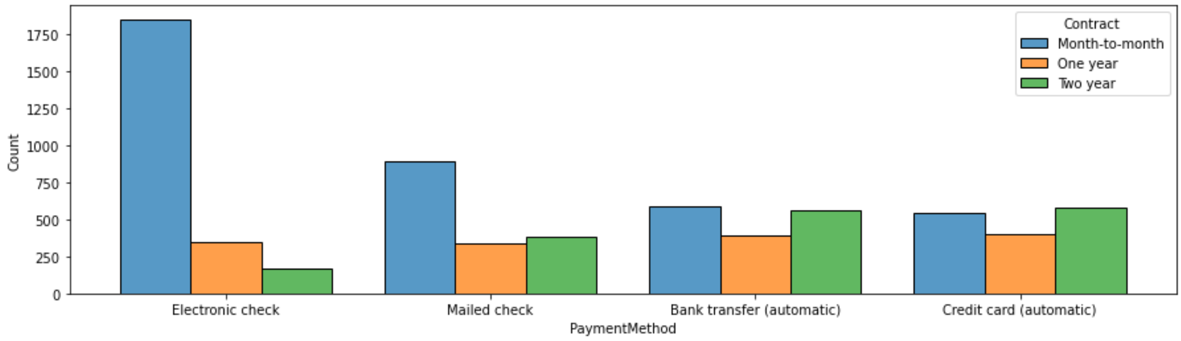


In [110]:

```python
plt.figure(figsize=(15,4))
sns.histplot(data=data, x=data['PaymentMethod'], hue=data['Contract'], multiple="dodge'
```

Out[110]:

```
<AxesSubplot:xlabel='PaymentMethod', ylabel='Count'>
```



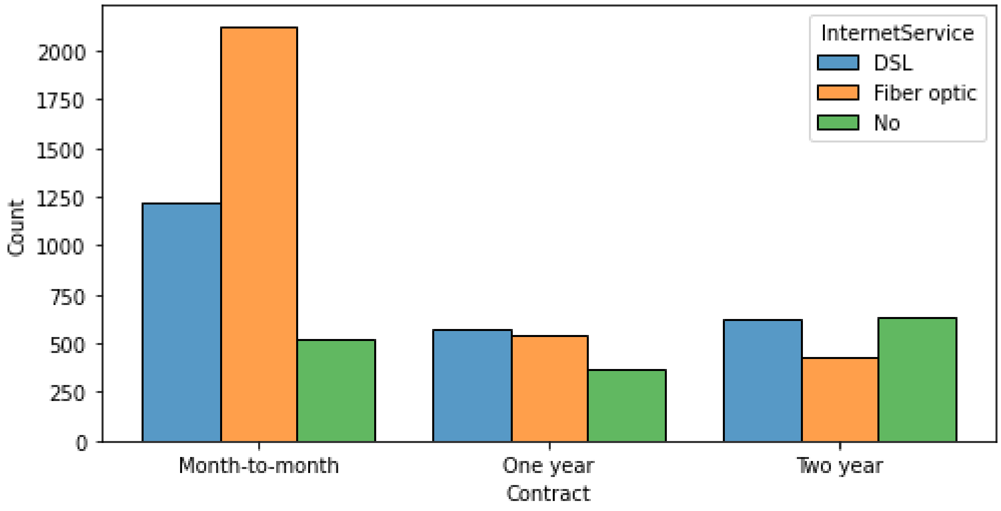
From the above table the {Month-to-month} contract is the most contract used with [Electronic check] method

In [113]:

```python
plt.figure(figsize=(8,4))
sns.histplot(data=data, x=data['Contract'], hue=data['InternetService'], multiple="dodg
```

Out[113]:

```
<AxesSubplot:xlabel='Contract', ylabel='Count'>
```

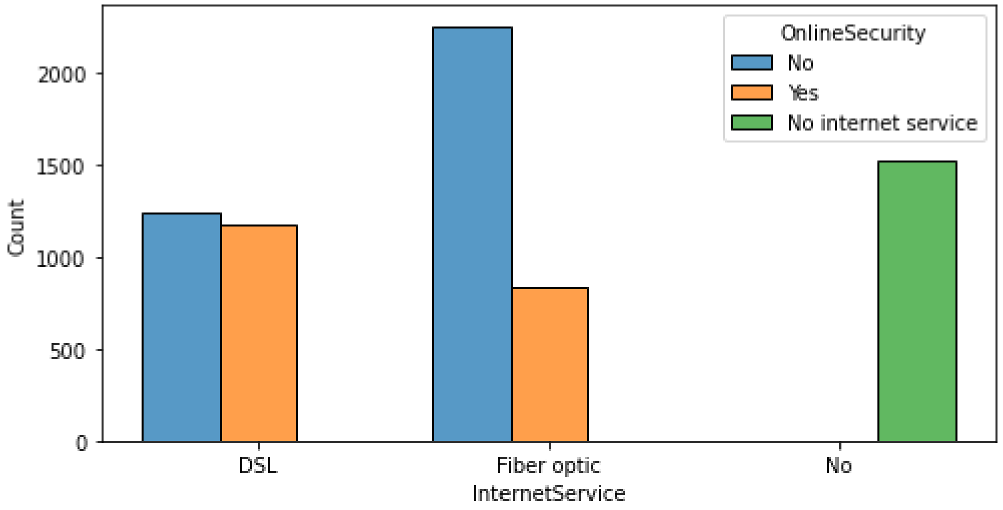


In [117]:

```
plt.figure(figsize=(8,4))
sns.histplot(data=data, x=data['InternetService'], hue=data['OnlineSecurity'], multiple
```

Out[117]:

```
<AxesSubplot:xlabel='InternetService', ylabel='Count'>
```



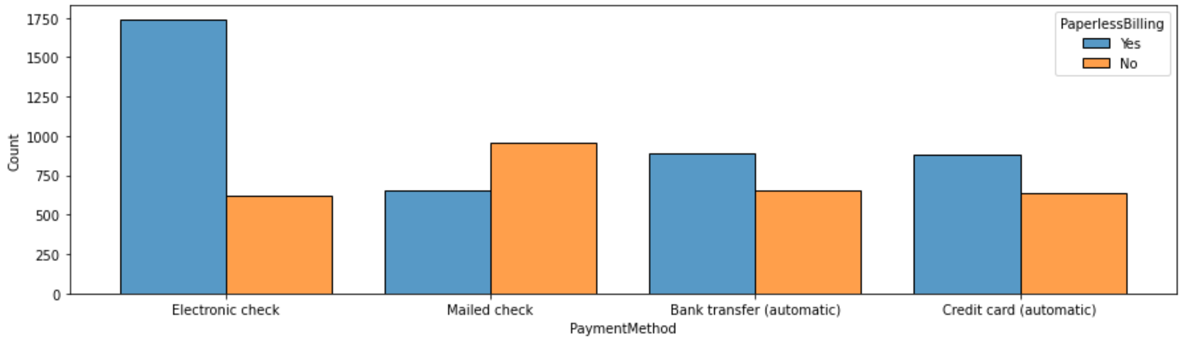
As showing above graph, about half the DSL servises are online secured while the Fiber optic service is most of it have no security

In [118]:

```
plt.figure(figsize=(15,4))
sns.histplot(data=data, x=data['PaymentMethod'], hue=data['PaperlessBilling'], multiple
```

Out[118]:

```
<AxesSubplot:xlabel='PaymentMethod', ylabel='Count'>
```

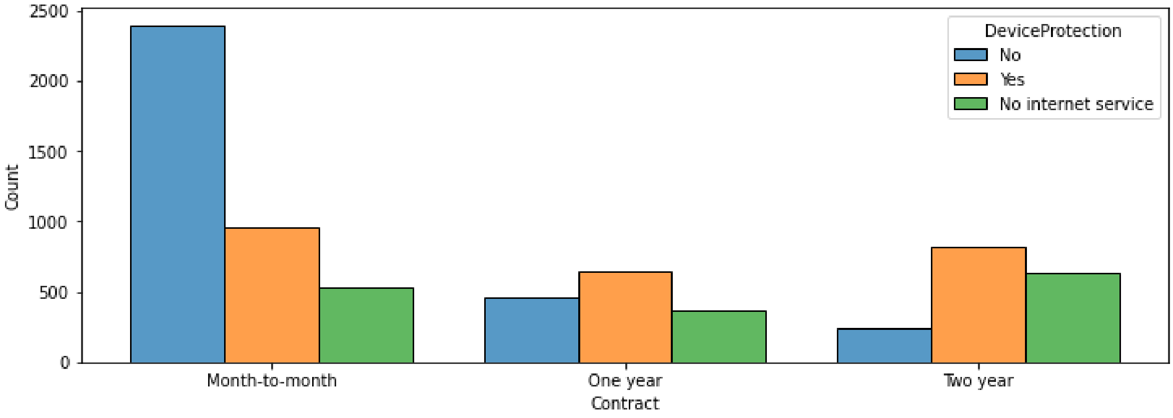


In [127]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['Contract'], hue=data['DeviceProtection'], multiple="doc
```

Out[127]:

```
<AxesSubplot:xlabel='Contract', ylabel='Count'>
```



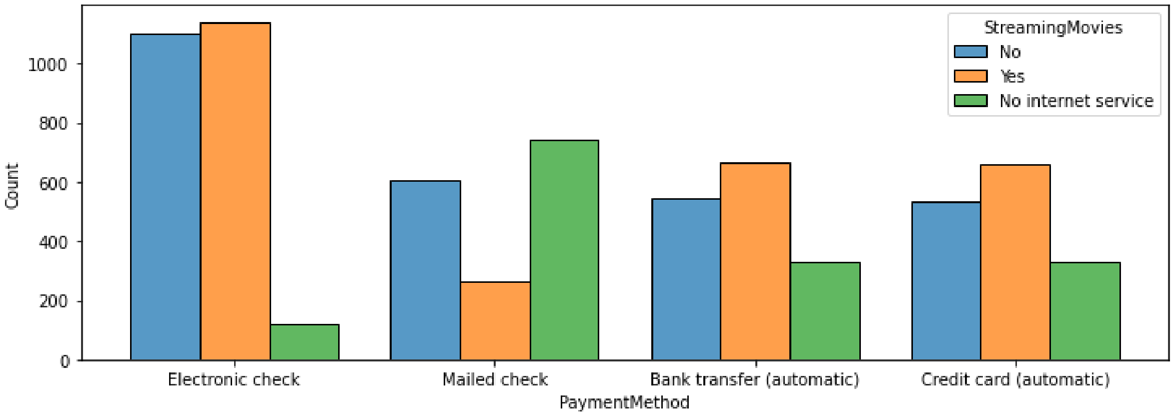
Mailed check are the only methode has less paperlessBilling

In [121]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['PaymentMethod'], hue=data['StreamingMovies'], multiple=
```

Out[121]:

```
<AxesSubplot:xlabel='PaymentMethod', ylabel='Count'>
```

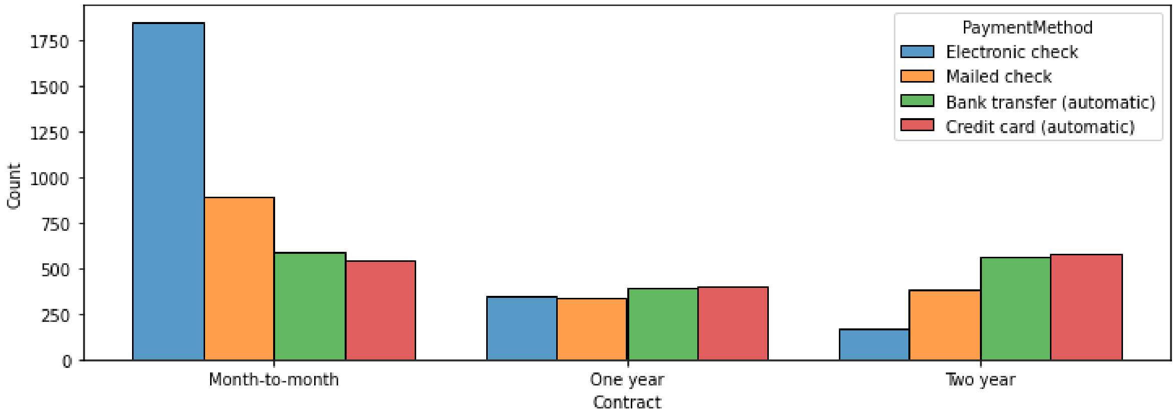


In [205]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['Contract'], hue=data['PaymentMethod'], multiple="dodge'
```

Out[205]:

```
<AxesSubplot:xlabel='Contract', ylabel='Count'>
```

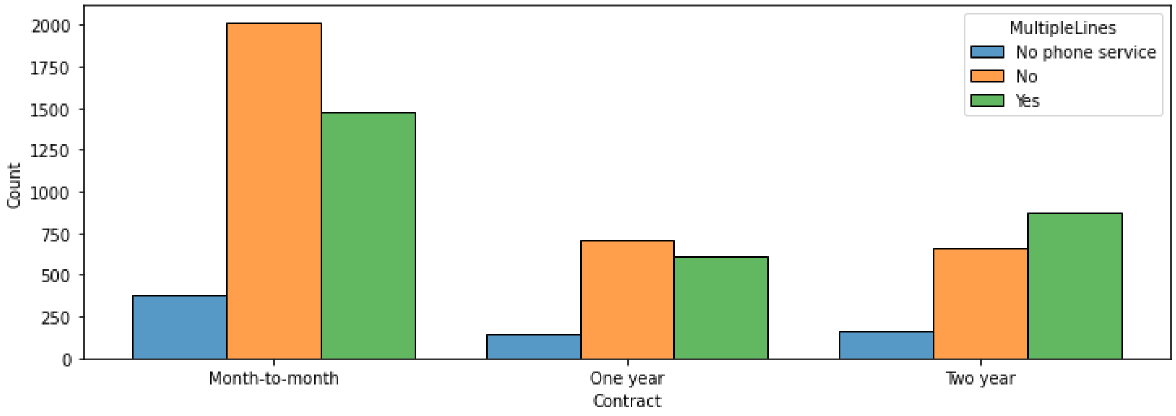


In [126]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['Contract'], hue=data['MultipleLines'], multiple="dodge'
```

Out[126]:

```
<AxesSubplot:xlabel='Contract', ylabel='Count'>
```

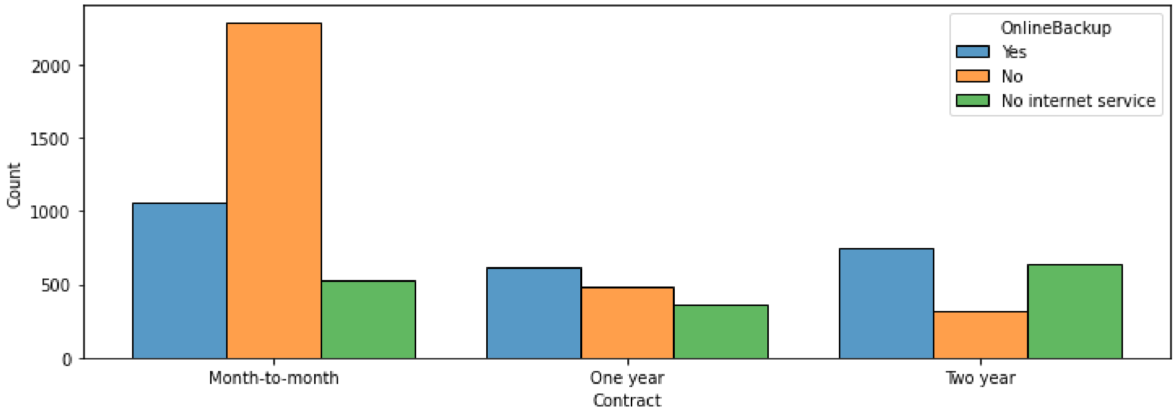


In [189]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['Contract'], hue=data['OnlineBackup'], multiple="dodge",
```

Out[189]:

```
<AxesSubplot:xlabel='Contract', ylabel='Count'>
```



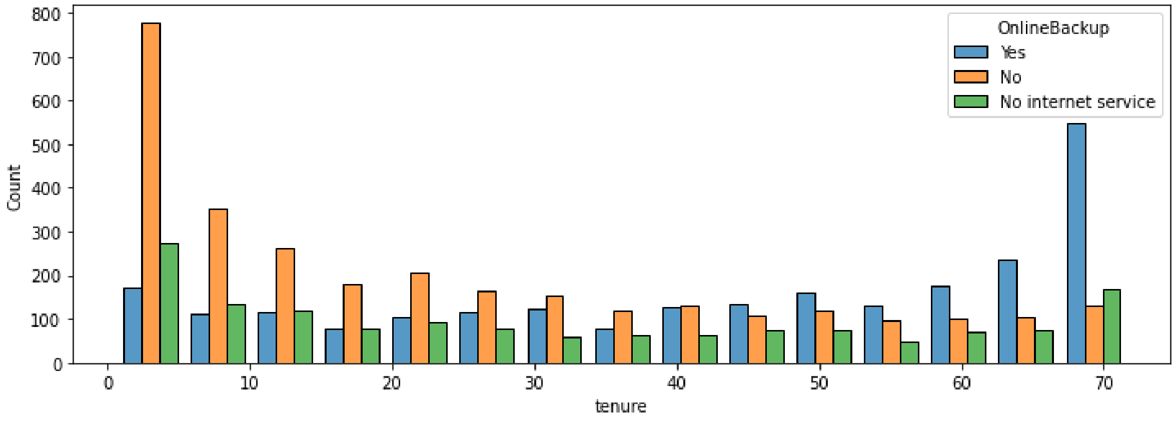
Month-to-month payment method need to inhance the online backup

In [220]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['tenure'], hue=data['OnlineBackup'], multiple="dodge", s
```

Out[220]:

```
<AxesSubplot:xlabel='tenure', ylabel='Count'>
```

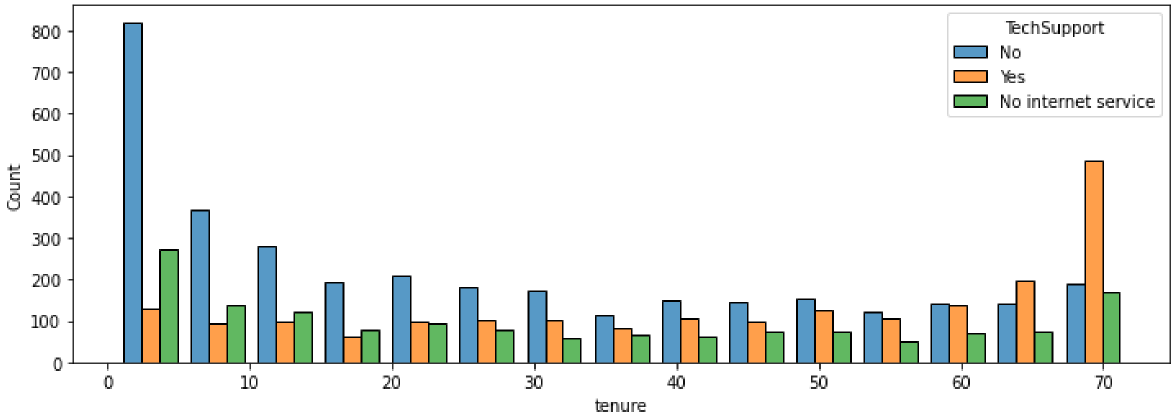


In [191]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['tenure'], hue=data['TechSupport'], multiple="dodge", sh
```

Out[191]:

```
<AxesSubplot:xlabel='tenure', ylabel='Count'>
```

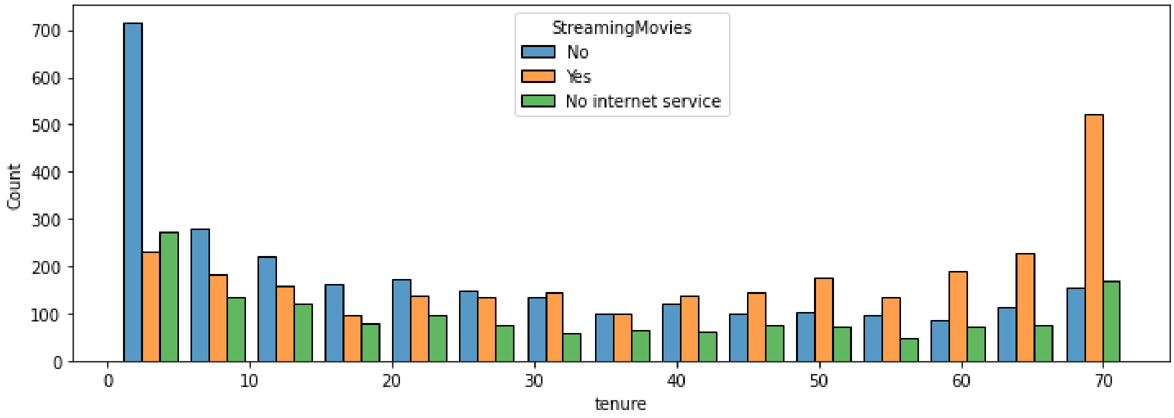


In [193]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['tenure'], hue=data['StreamingMovies'], multiple="dodge"
```

Out[193]:

```
<AxesSubplot:xlabel='tenure', ylabel='Count'>
```

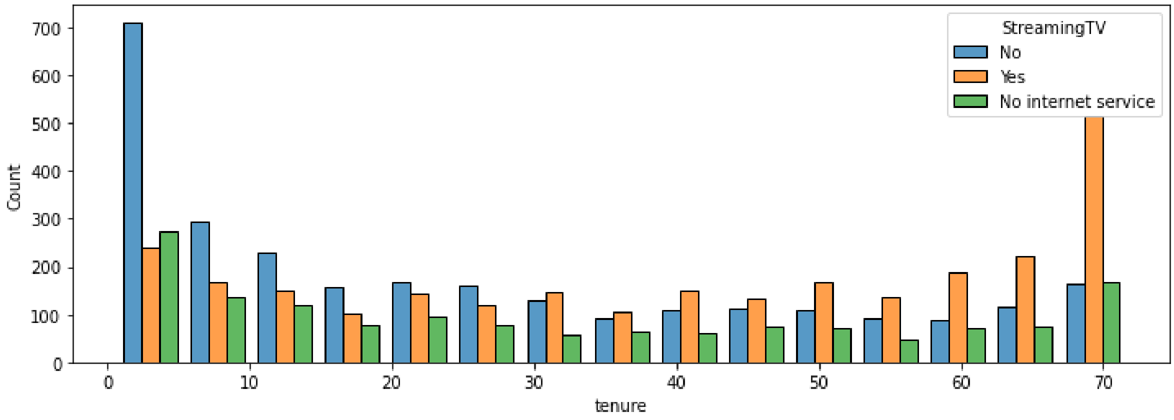


In [194]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['tenure'], hue=data['StreamingTV'], multiple="dodge", sh
```

Out[194]:

```
<AxesSubplot:xlabel='tenure', ylabel='Count'>
```

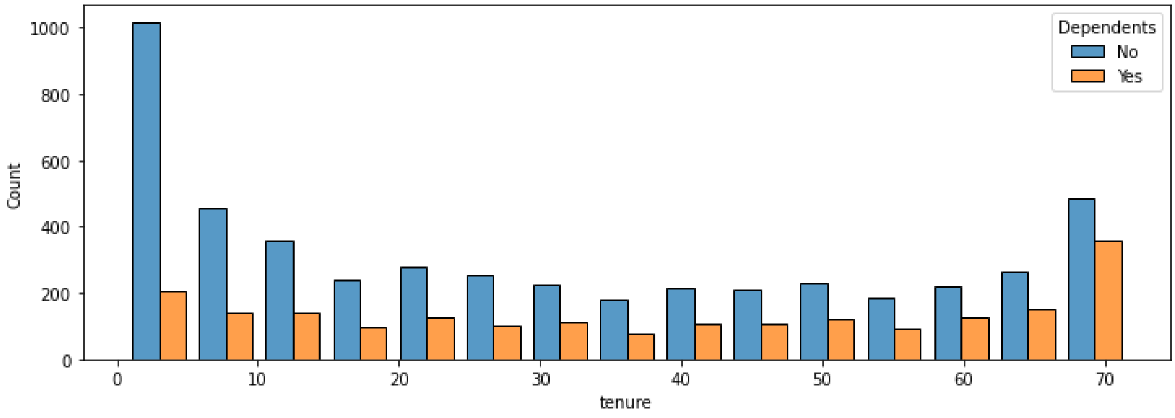


In [213]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['tenure'], hue=data['Dependents'], multiple="dodge", shr
```

Out[213]:

```
<AxesSubplot:xlabel='tenure', ylabel='Count'>
```

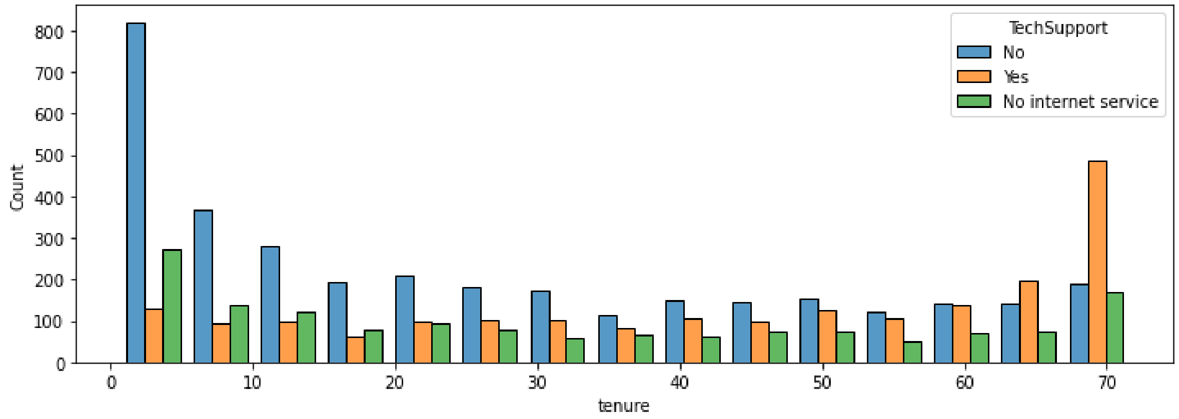


In [225]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['tenure'], hue=data['TechSupport'], multiple="dodge", sh
```

Out[225]:

```
<AxesSubplot:xlabel='tenure', ylabel='Count'>
```



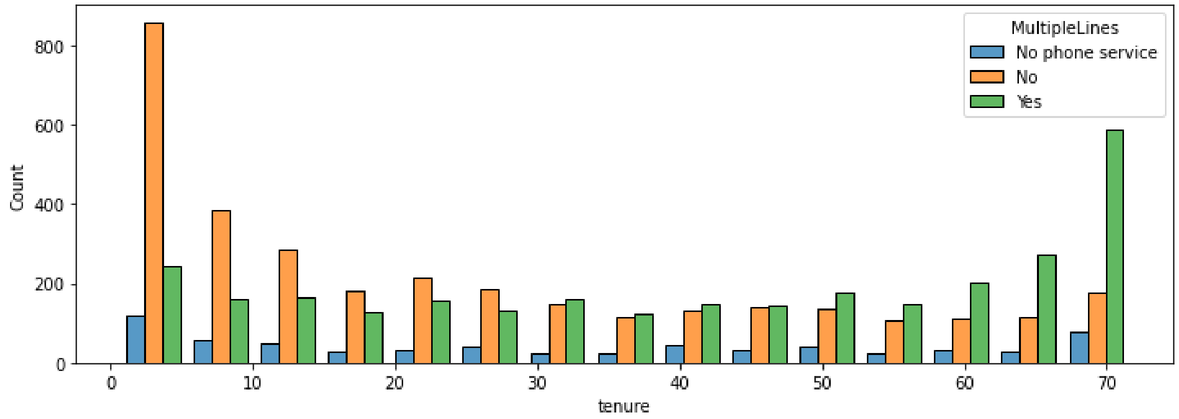
Most of the staying customer in the company have partners

In [208]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['tenure'], hue=data['MultipleLines'], multiple="dodge",
```

Out[208]:

```
<AxesSubplot:xlabel='tenure', ylabel='Count'>
```



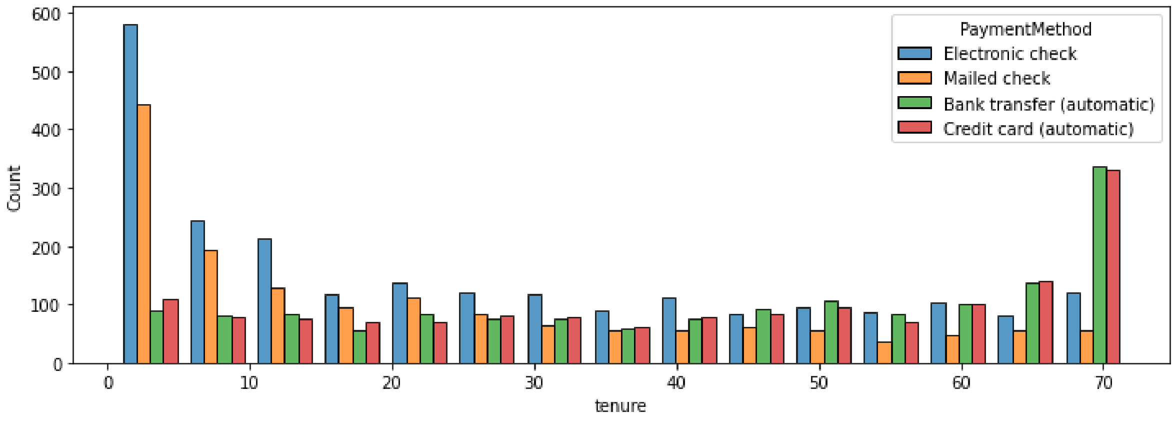
The most of customer staying in company who have multiple lines

In [198]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['tenure'], hue=data['PaymentMethod'], multiple="dodge",
```

Out[198]:

```
<AxesSubplot:xlabel='tenure', ylabel='Count'>
```



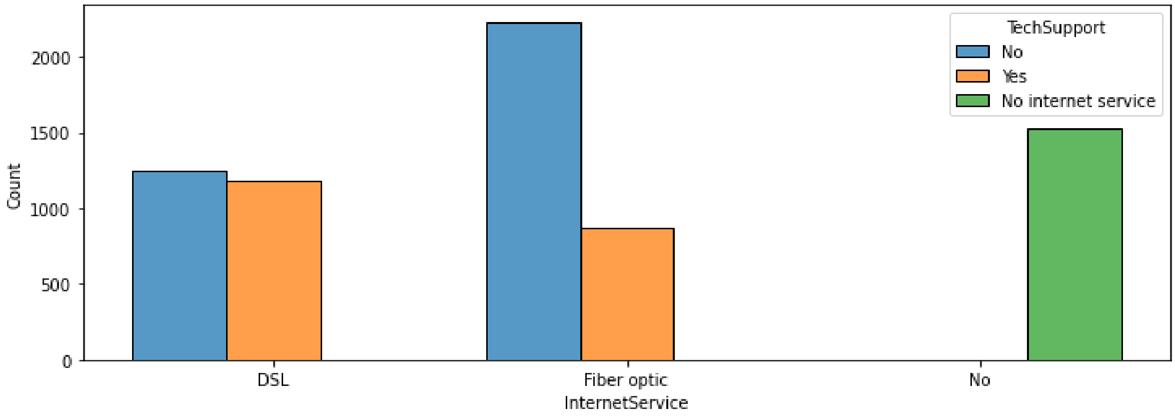
The customers who pay by Bank transfer and credit cards are the most customer are staying in company

In [130]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['InternetService'], hue=data['TechSupport'], multiple="d
```

Out[130]:

```
<AxesSubplot:xlabel='InternetService', ylabel='Count'>
```



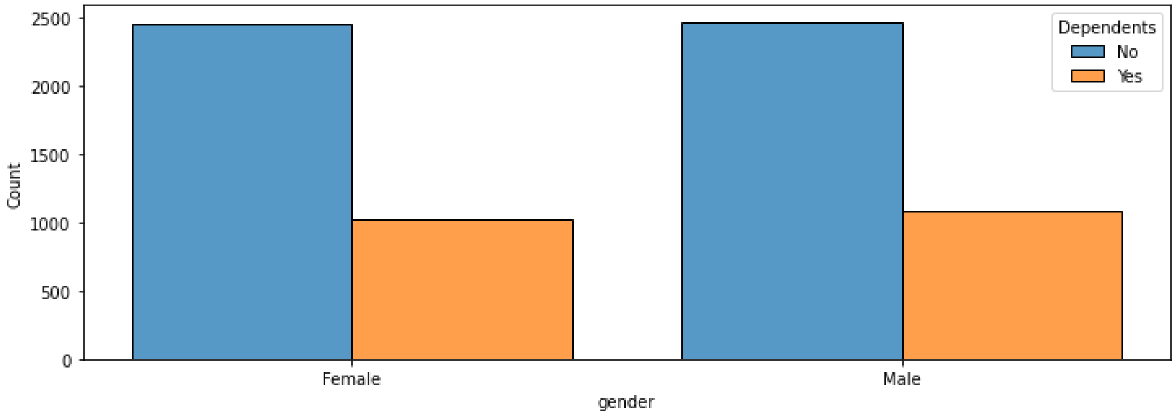
Fiber optic service need more techSupport

In [132]:

```python
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['gender'], hue=data['Dependents'], multiple="dodge", shr
```

Out[132]:

```
<AxesSubplot:xlabel='gender', ylabel='Count'>
```

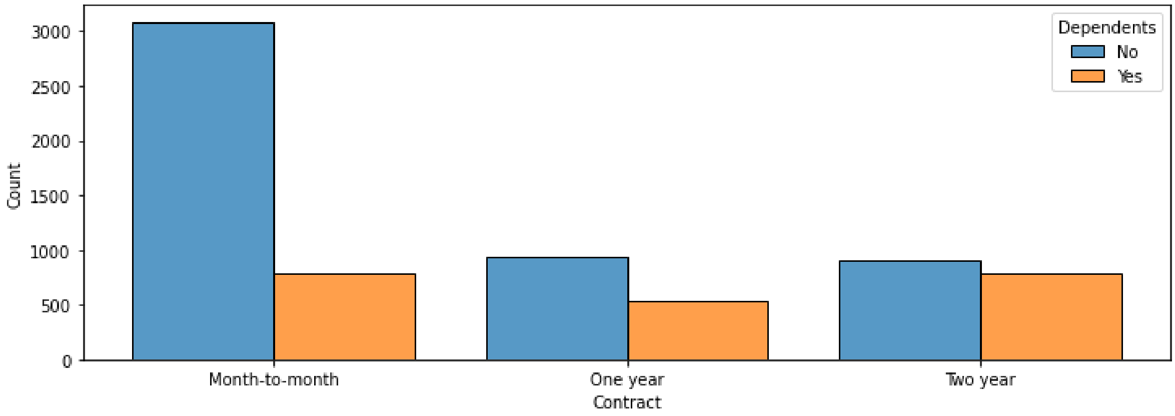


In [133]:

```python
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['Contract'], hue=data['Dependents'], multiple="dodge", s
```

Out[133]:

```
<AxesSubplot:xlabel='Contract', ylabel='Count'>
```

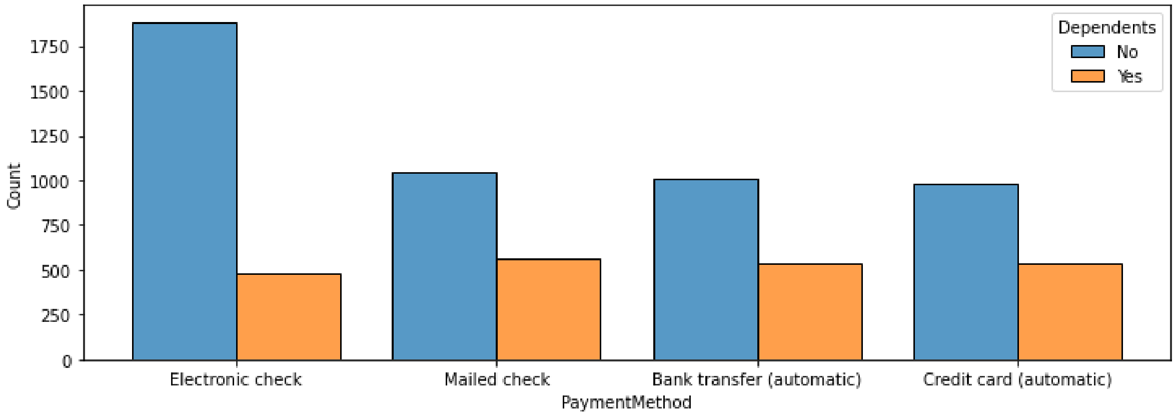


In [134]:

```
plt.figure(figsize=(12,4))
sns.histplot(data=data, x=data['PaymentMethod'], hue=data['Dependents'], multiple="dodg
```

Out[134]:

```
<AxesSubplot:xlabel='PaymentMethod', ylabel='Count'>
```



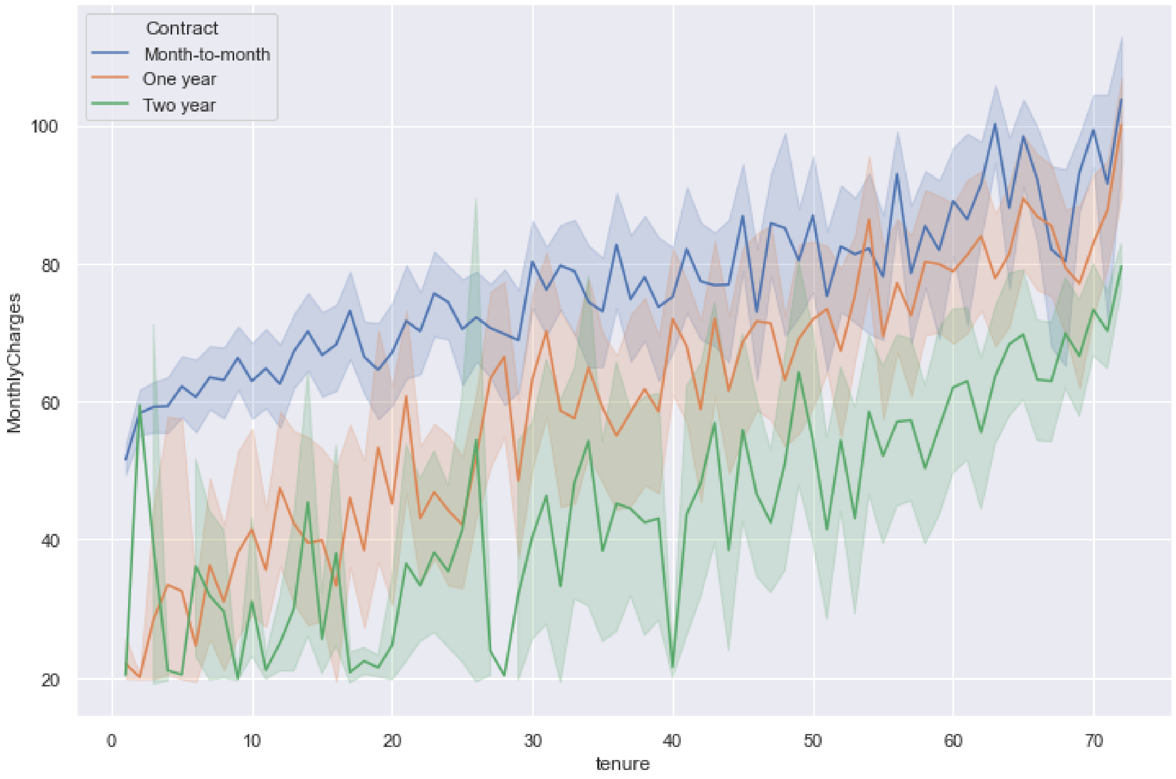
The number of dependents accounts are approximate equal for all payment methods

In [25]:

```
plt.figure(figsize=(12,8))
sns.set_theme(style="darkgrid")
sns.lineplot(data=data, x=data['tenure'], y=data['MonthlyCharges'],hue=data['Contract'])
```

Out[25]:

```
<AxesSubplot:xlabel='tenure', ylabel='MonthlyCharges'>
```

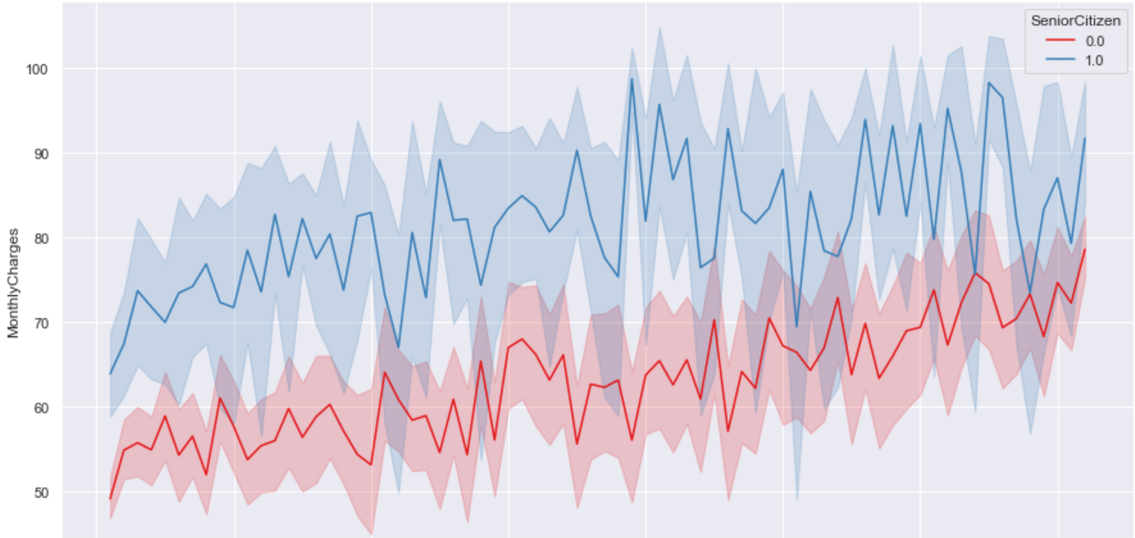


In [57]:

```
plt.figure(figsize=(15,8))
sns.set_theme(style="darkgrid")
sns.lineplot(data=data, x=data['tenure'], y=data['MonthlyCharges'],hue=data['SeniorCiti
```

Out[57]:

```
<AxesSubplot:xlabel='tenure', ylabel='MonthlyCharges'>
```

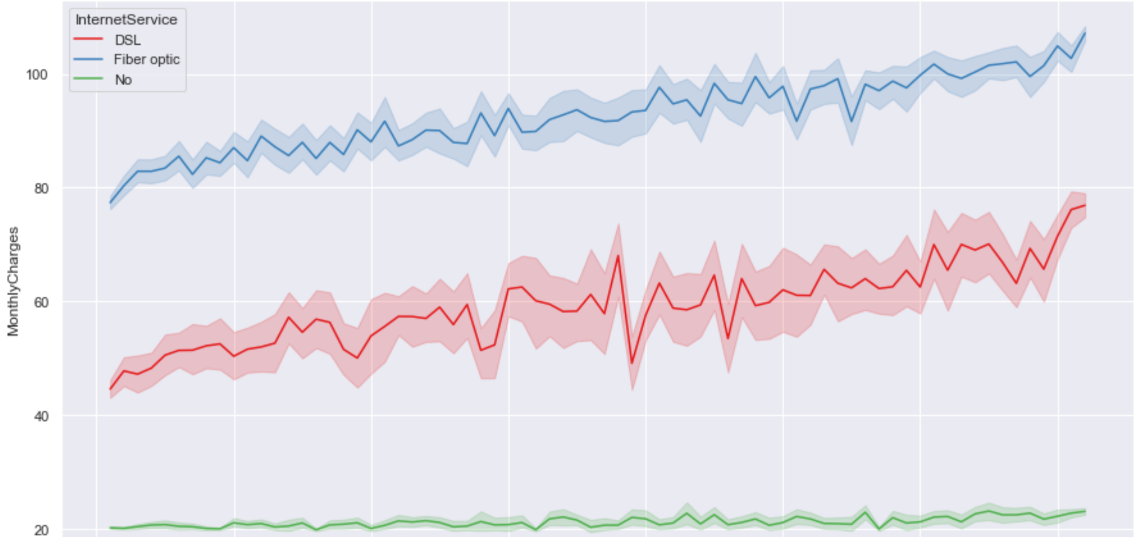


In [44]:

```
,8))
arkgrid")
, x=data['tenure'], y=data['MonthlyCharges'],hue=data['InternetService'], palette = "Set1")
```

Out[44]:

```
<AxesSubplot:xlabel='tenure', ylabel='MonthlyCharges'>
```

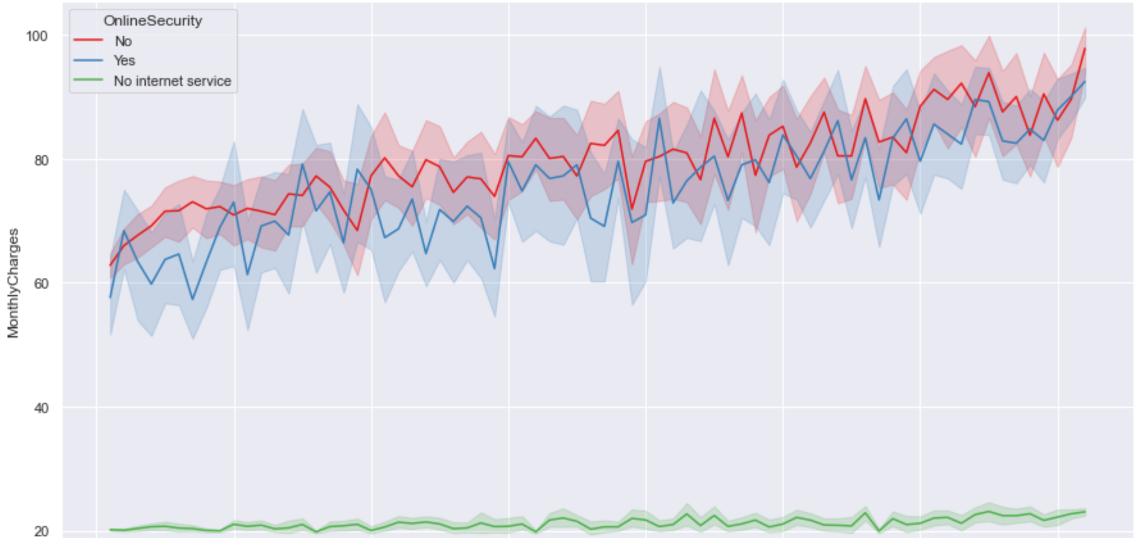


In [47]:

```
figsize=(15,8))
ne(style="darkgrid")
:(data=data, x=data['tenure'], y=data['MonthlyCharges'],hue=data['OnlineSecurity'], palette
```

Out[47]:

```
<AxesSubplot:xlabel='tenure', ylabel='MonthlyCharges'>
```

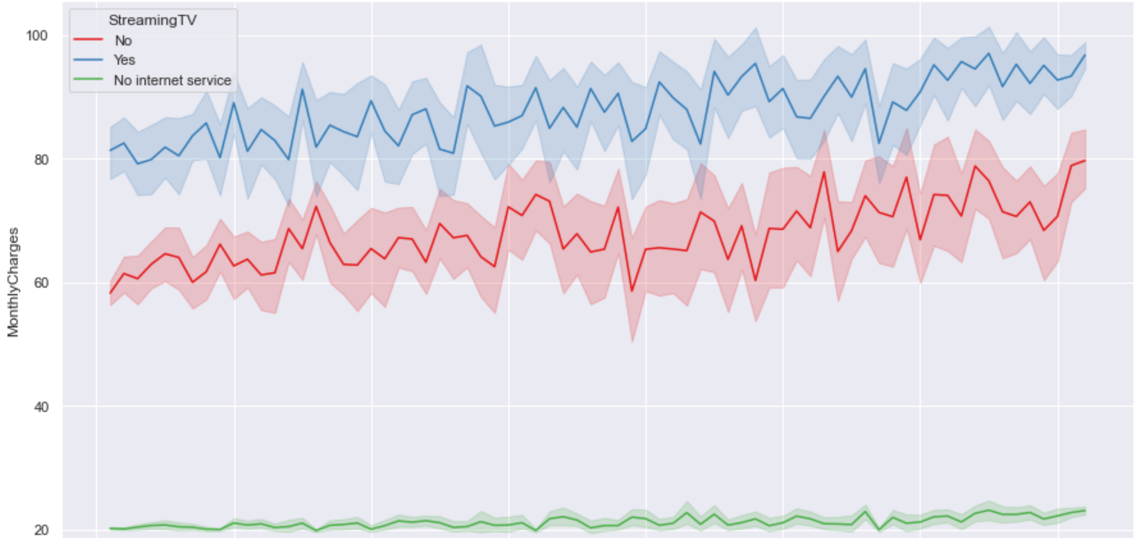


In [50]:

```
ze=(15,8))
yle="darkgrid")
a=data, x=data['tenure'], y=data['MonthlyCharges'],hue=data['StreamingTV'], palette = "Set1
```

Out[50]:

```
<AxesSubplot:xlabel='tenure', ylabel='MonthlyCharges'>
```

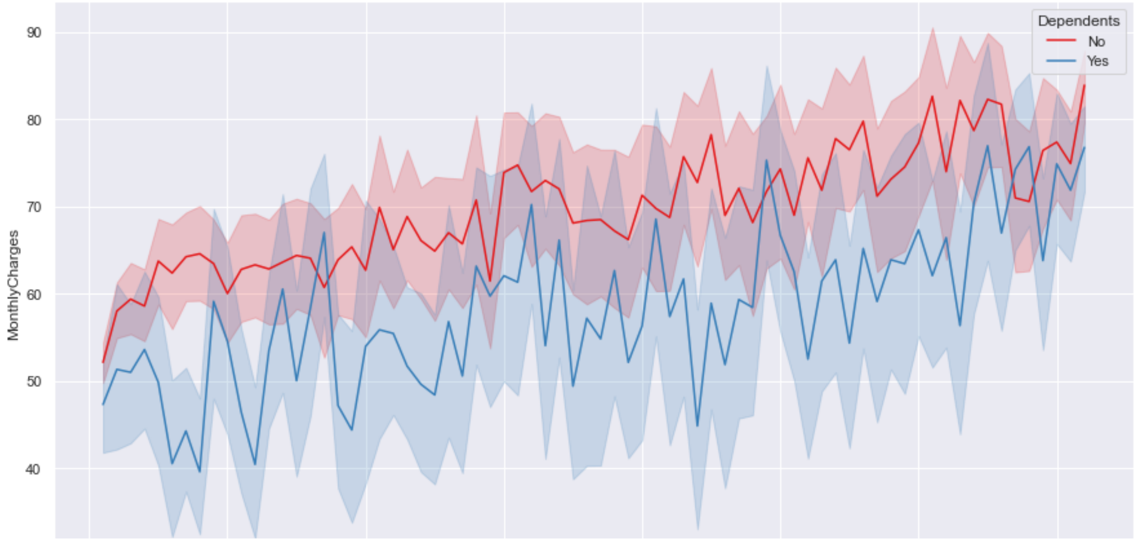


In [51]:

```
e=(15,8))
le="darkgrid")
=data, x=data['tenure'], y=data['MonthlyCharges'],hue=data['Dependents'], palette = "Set1")
```

Out[51]:

```
<AxesSubplot:xlabel='tenure', ylabel='MonthlyCharges'>
```

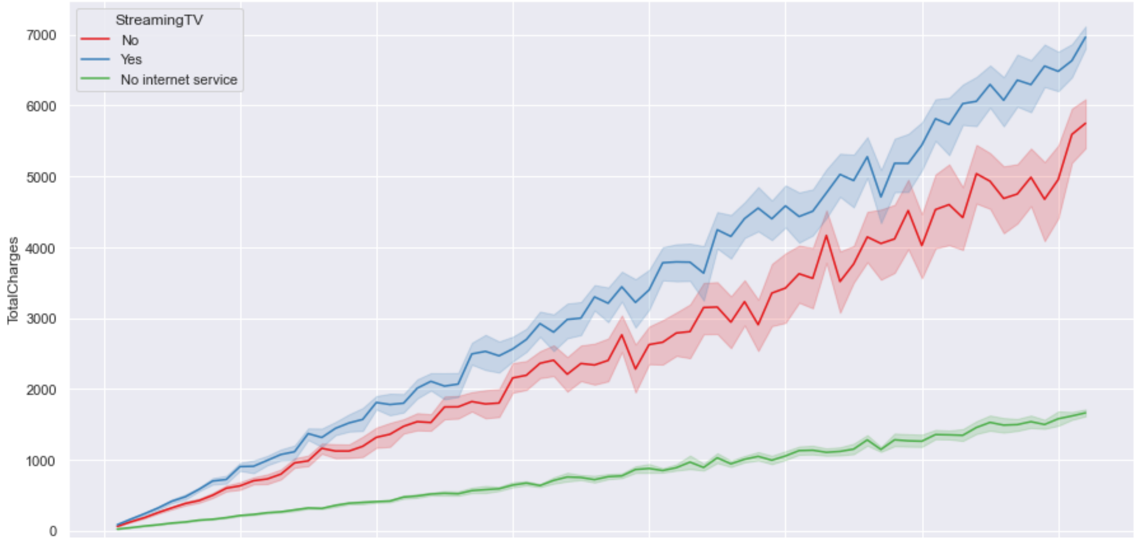


In [55]:

```
ze=(15,8))
yle="darkgrid")
a=data, x=data['tenure'], y=data['TotalCharges'],hue=data['StreamingTV'], palette = "Set1")
```

Out[55]:

```
<AxesSubplot:xlabel='tenure', ylabel='TotalCharges'>
```

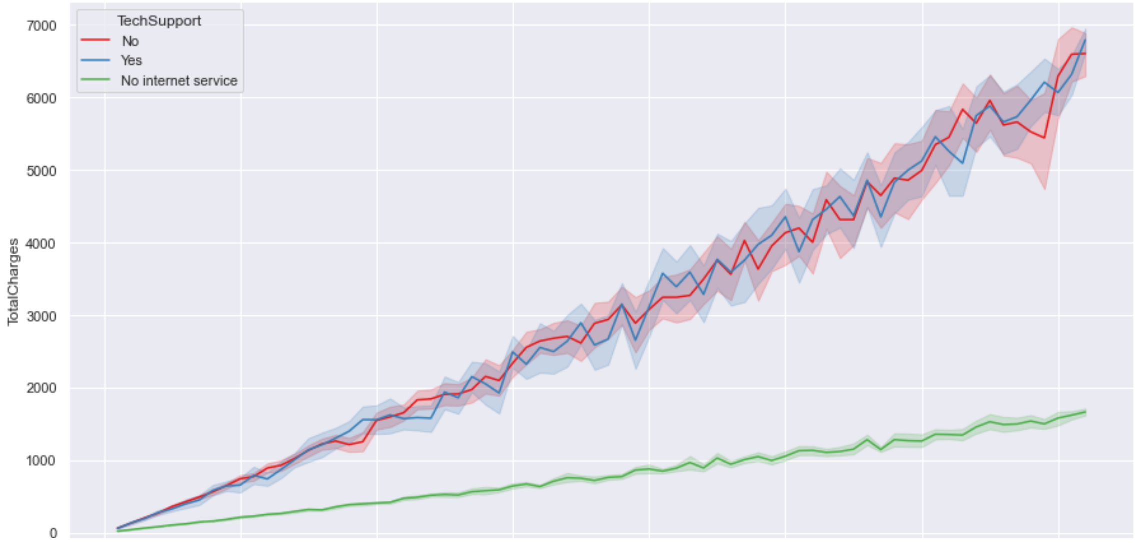


In [54]:

```
ze=(15,8))
yle="darkgrid")
a=data, x=data['tenure'], y=data['TotalCharges'],hue=data['TechSupport'], palette = "Set1")
```

Out[54]:

```
<AxesSubplot:xlabel='tenure', ylabel='TotalCharges'>
```



In [53]:

```
data.groupby(['Contract','InternetService']).sum()
```

Out[53]:

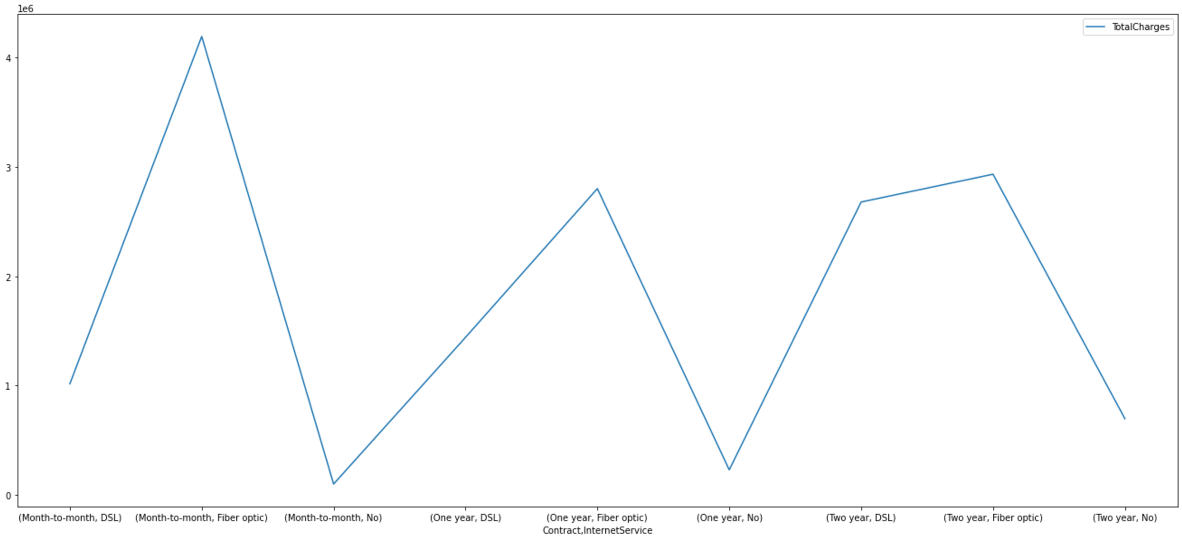
		SeniorCitizen	tenure	MonthlyCharges	TotalCharges
Contract	**InternetService**				
Month-to-month	**DSL**	159.0	19191.0	61418.45	1015136.20
	Fiber optic	636.0	45825.0	185181.10	4191552.55
	No	12.0	4744.0	10694.60	99172.75
One year	**DSL**	50.0	23077.0	34996.15	1439990.85
	Fiber optic	124.0	28081.0	53242.15	2800424.35
	No	16.0	10770.0	7578.30	228921.60
Two year	**DSL**	50.0	37142.0	44250.75	2677700.30
	Fiber optic	71.0	27964.0	44861.15	2931646.05
	No	24.0	31060.0	13893.95	696740.35

In [168]:

```
n=data.groupby(['Contract','InternetService']).sum().plot(y='TotalCharges',figsize=(23,
n
```

Out[168]:

```
<AxesSubplot:xlabel='Contract,InternetService'>
```

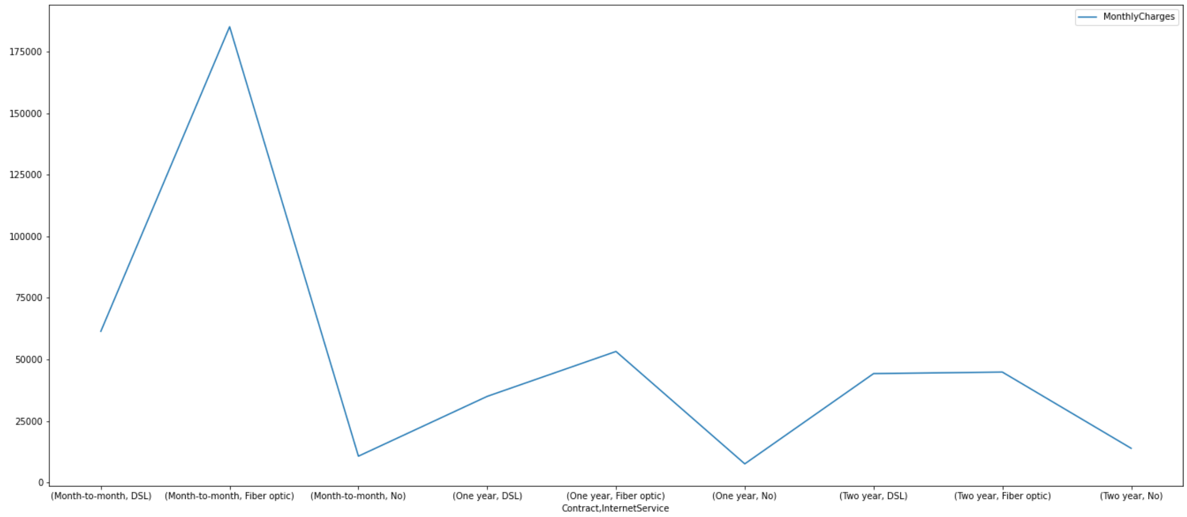


In [167]:

```
n=data.groupby(['Contract','InternetService']).sum().plot(y='MonthlyCharges',figsize=(2
n
```

Out[167]:

```
<AxesSubplot:xlabel='Contract,InternetService'>
```

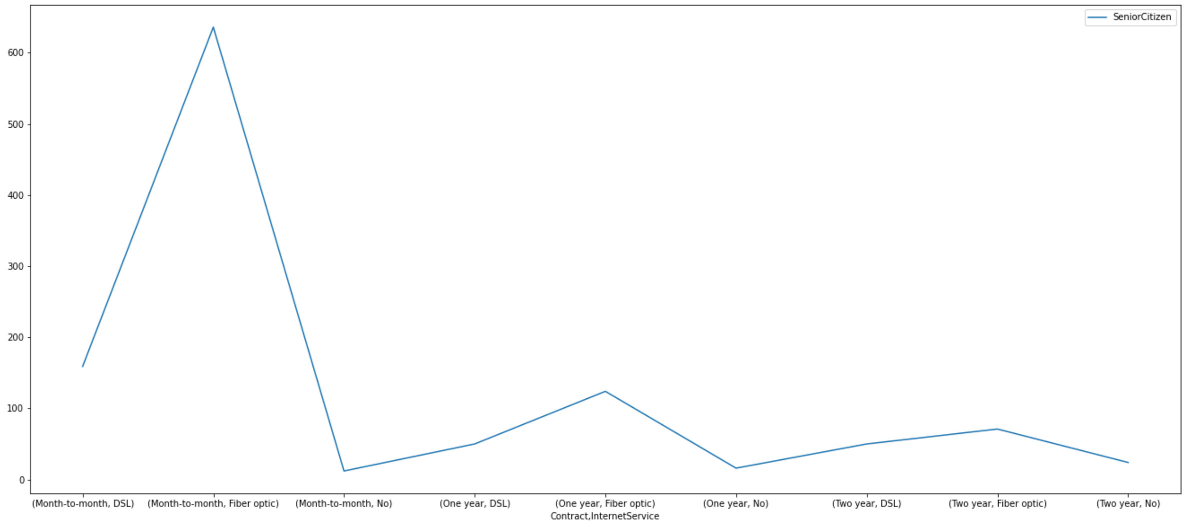


In [169]:

```
n=data.groupby(['Contract','InternetService']).sum().plot(y='SeniorCitizen',figsize=(2
n
```

Out[169]:

```
<AxesSubplot:xlabel='Contract,InternetService'>
```



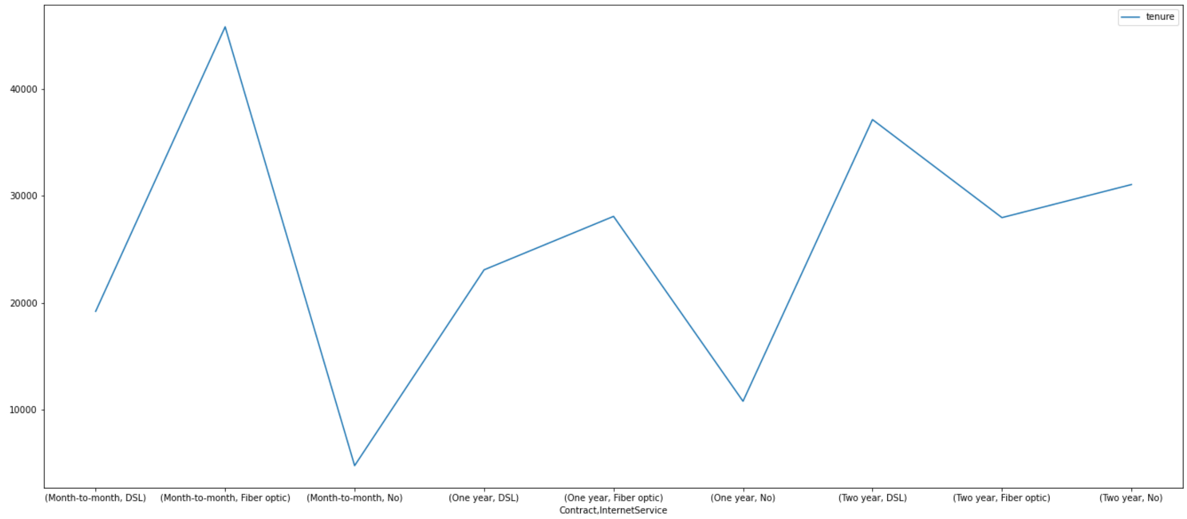
Most of senior citizens are pay by month-to-month method with fibar optic as internit service

In [170]:

```
n=data.groupby(['Contract','InternetService']).sum().plot(y='tenure',figsize=(23,10))
n
```

Out[170]:

```
<AxesSubplot:xlabel='Contract,InternetService'>
```

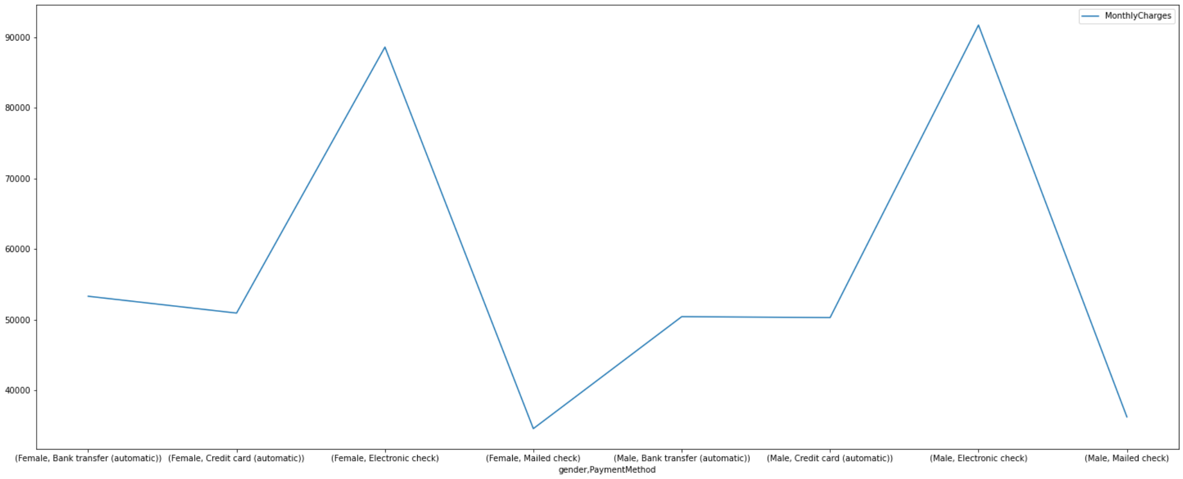


In [184]:

```
df=data.groupby(['gender','PaymentMethod']).sum().plot(y='MonthlyCharges',figsize=(25,1
df
```

Out[184]:

```
<AxesSubplot:xlabel='gender,PaymentMethod'>
```

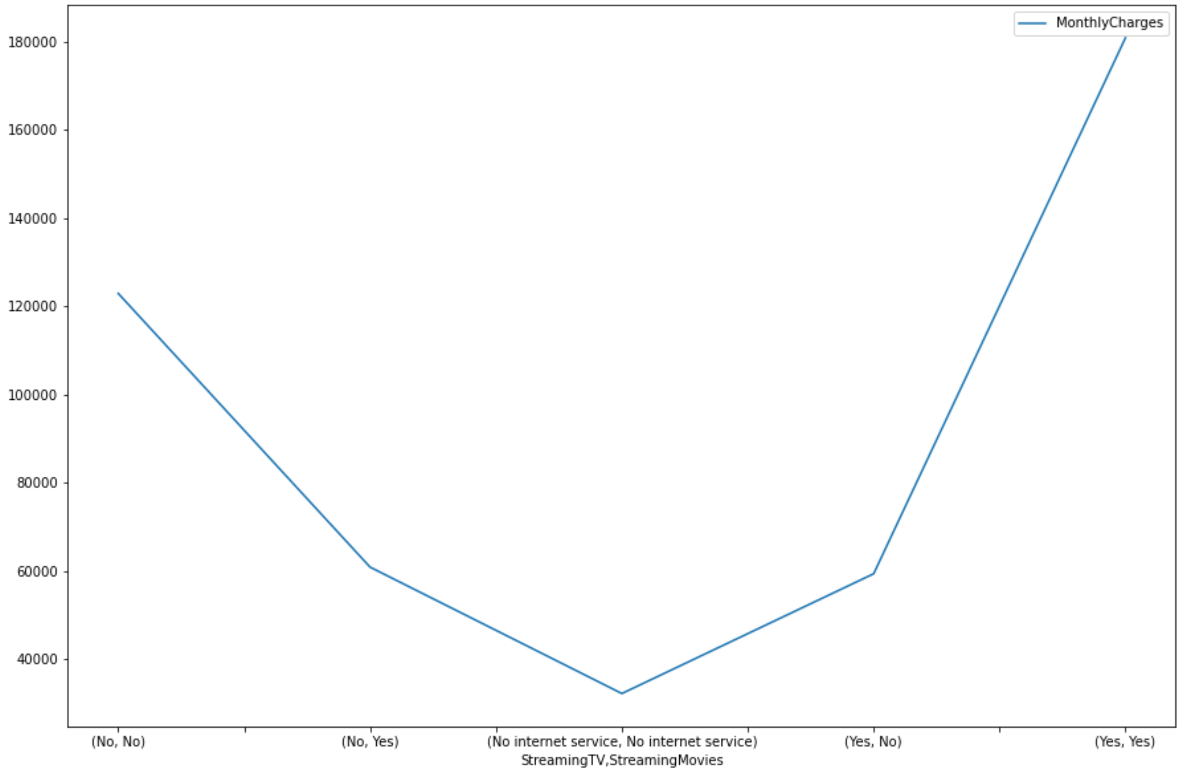


In [196]:

```
df=data.groupby(['StreamingTV','StreamingMovies']).sum().plot(y='MonthlyCharges',figsiz
df
```

Out[196]:

```
<AxesSubplot:xlabel='StreamingTV,StreamingMovies'>
```



The income from customers without both the streaming of tv and movies is more than the customers participant in only one of them

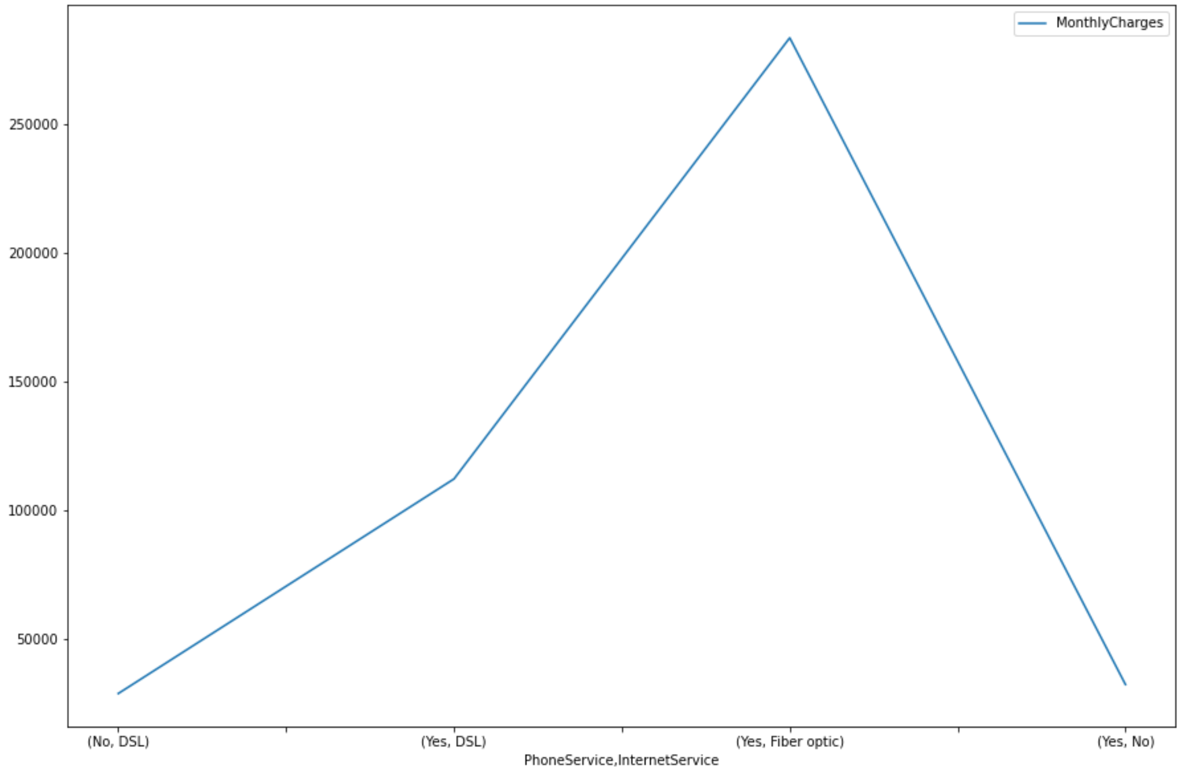
Need to inhance the streaming services

In [204]:

```
df=data.groupby(['PhoneService','InternetService']).sum().plot(y='MonthlyCharges',figsi
df
```

Out[204]:

```
<AxesSubplot:xlabel='PhoneService,InternetService'>
```

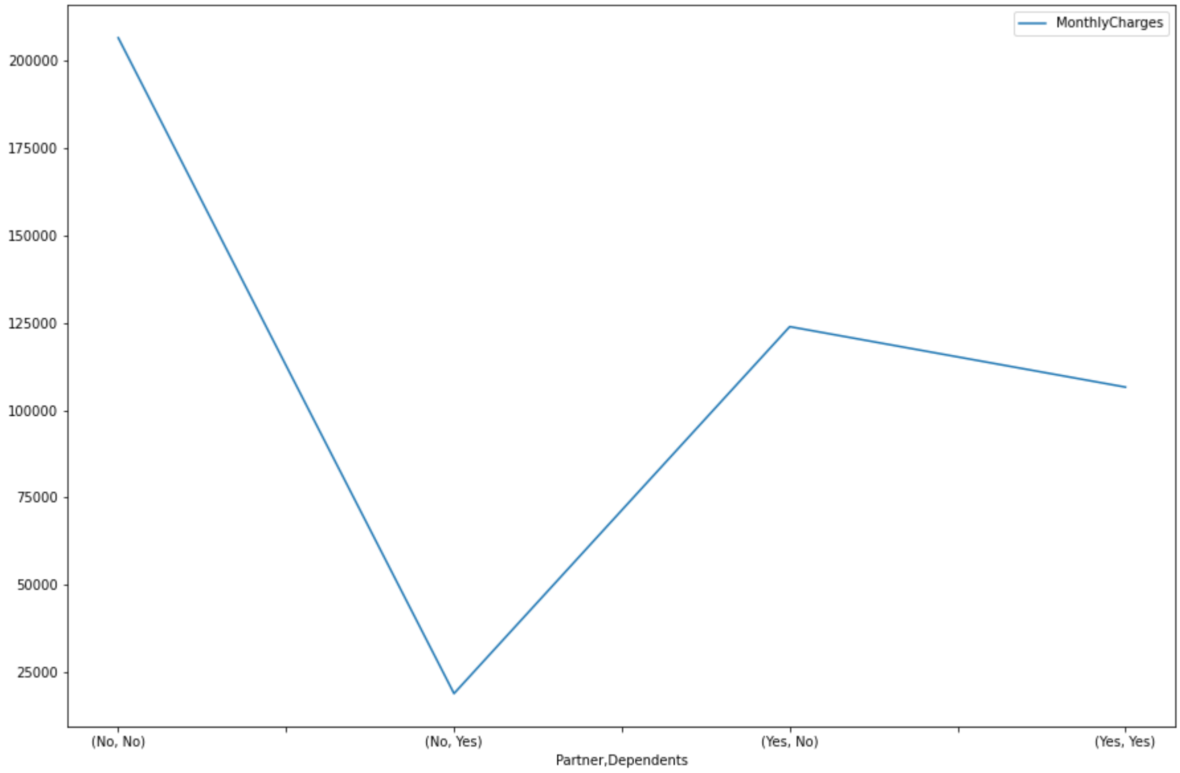


In [218]:

```python
df=data.groupby(['Partner','Dependents']).sum().plot(y='MonthlyCharges',figsize=(15,10)
df
```

Out[218]:

```
<AxesSubplot:xlabel='Partner,Dependents'>
```

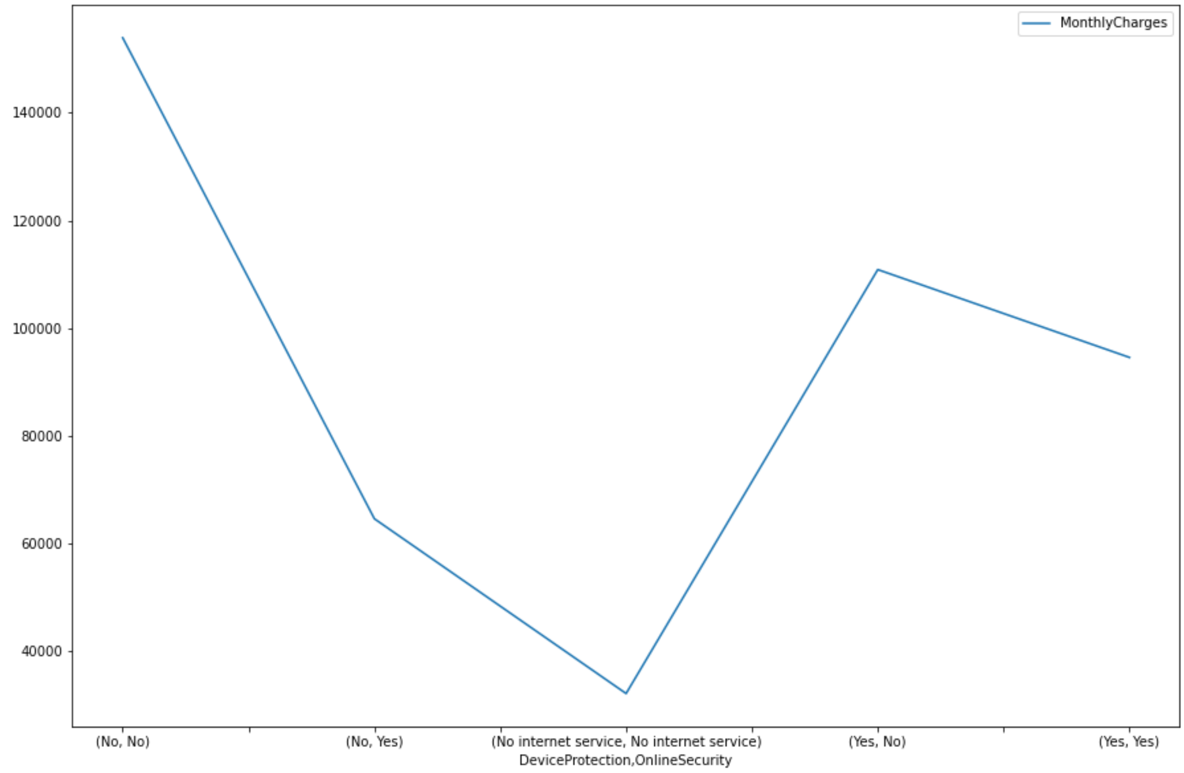


In [223]:

```
df=data.groupby(['DeviceProtection','OnlineSecurity']).sum().plot(y='MonthlyCharges',fi
df
```

Out[223]:

```
<AxesSubplot:xlabel='DeviceProtection,OnlineSecurity'>
```

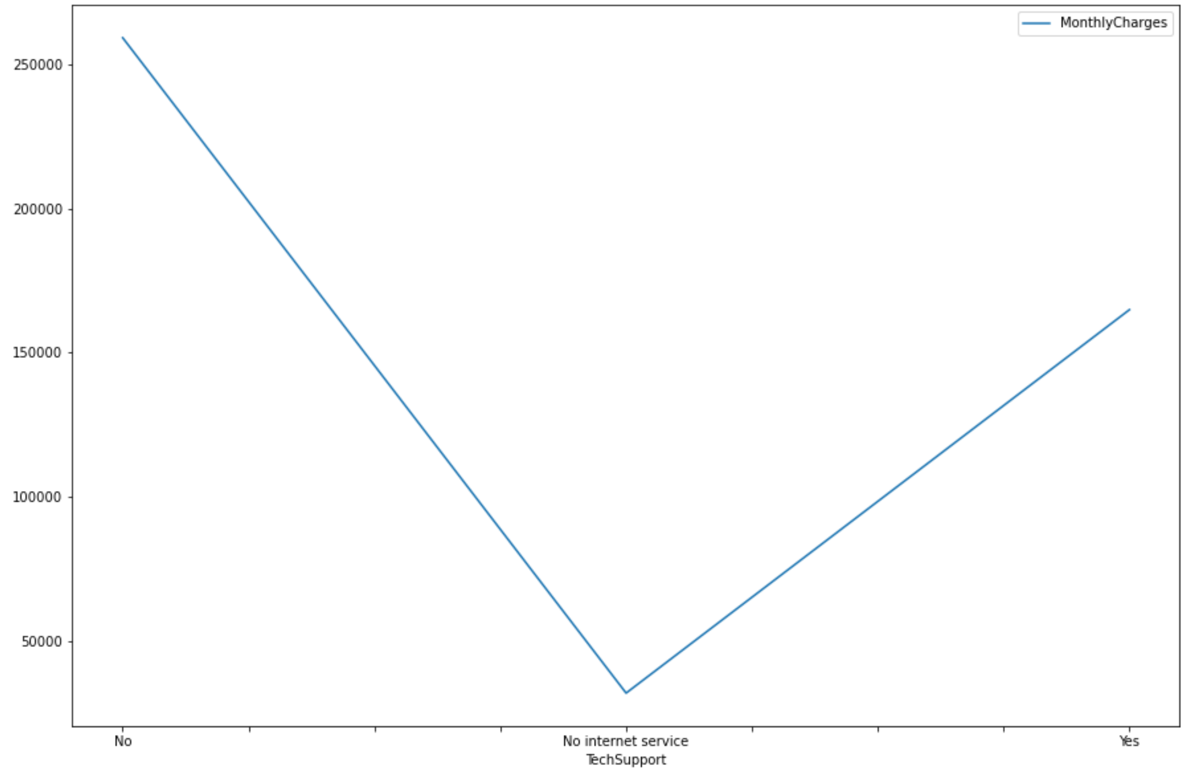


In [232]:

```
df=data.groupby(['TechSupport']).sum().plot(y='MonthlyCharges',figsize=(15,10))
df
```

Out[232]:

```
<AxesSubplot:xlabel='TechSupport'>
```

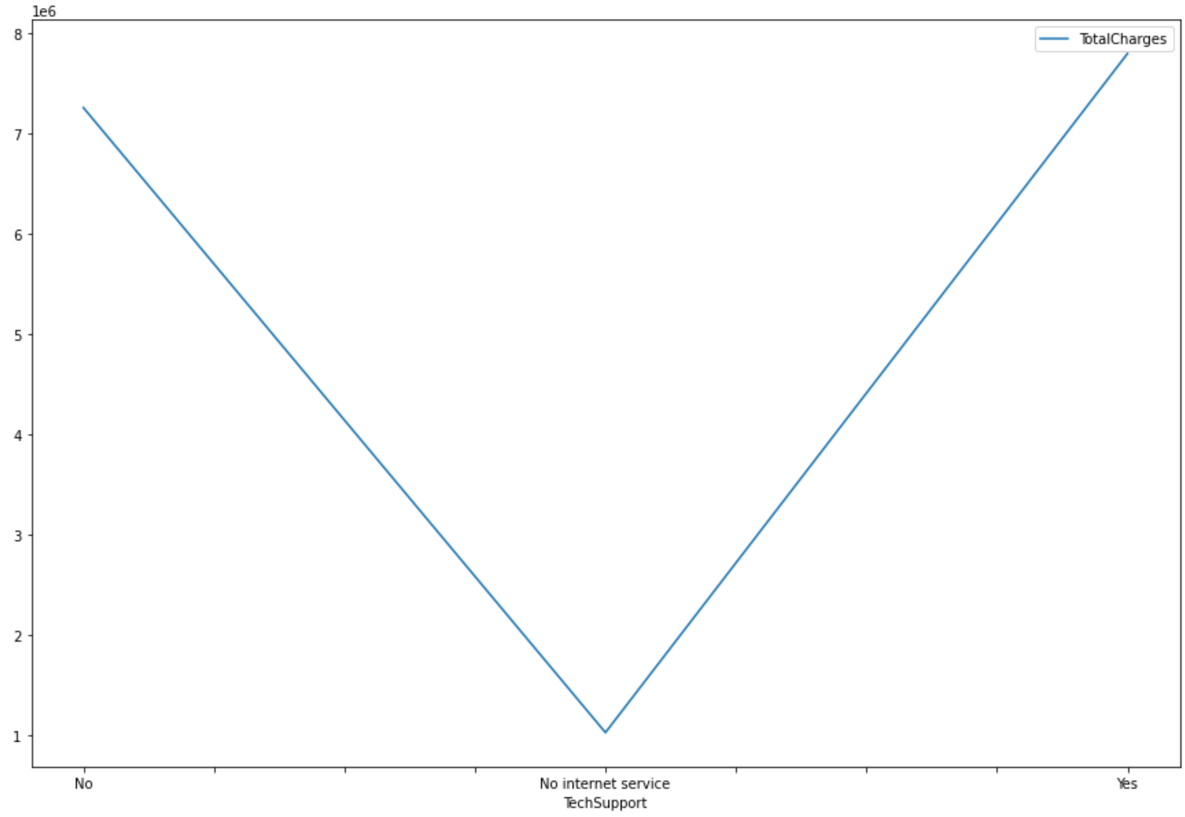


In [234]:

```
df=data.groupby(['TechSupport']).sum().plot(y='TotalCharges',figsize=(15,10))
df
```

Out[234]:

```
<AxesSubplot:xlabel='TechSupport'>
```



Conclusion

Most of the revenues are from customers contracting by Month-to-month method, using a fiber optic service and containg most of senior citizens.

However that the most income comes from customes paying by Electronic check, the most stable customer who stay in the company are the customers pay by Bank transfer and credit cards.

The customers who get the streaming services take the two (tv and movie) services or neither of the two

two.

All the fiber optic services has a phone services.

Most of the staying customer in the company have partners

Customers with no partner or dependents are the most value monthly charged.

The majorty of month-to-month contract has no online backup.

According to monthly revenue, the customer with no device protection or online security have the most monthly charged value.

Using Mailed check as payment method is has less probability for paper billing than any other method.

The income from customer with no tech support is higher than customer with tech support for monthly revenues, although the total revenue is higher for customer with tech support.