In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv(r'D:\Data Analytics\LetsUpgrade\Day 19\auto_insurance_payment.csv')
```

In [3]:
```
data
```

Out[3]:

	Number of claims	Total Payment
0	108	3925
1	19	462
2	13	157
3	124	4222
4	40	1194
...	...	...
58	9	874
59	31	2098
60	14	955
61	53	2446
62	26	1875

63 rows × 2 columns

In [4]:
```
data.tail()
```

Out[4]:

	Number of claims	Total Payment
58	9	874
59	31	2098
60	14	955
61	53	2446
62	26	1875

In [5]:
```
data.dtypes
```

Out[5]:
```
Number of claims    int64
 Total Payment      int64
dtype: object
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63 entries, 0 to 62
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Number of claims  63 non-null     int64
 1    Total Payment    63 non-null     int64
dtypes: int64(2)
memory usage: 1.1 KB
```

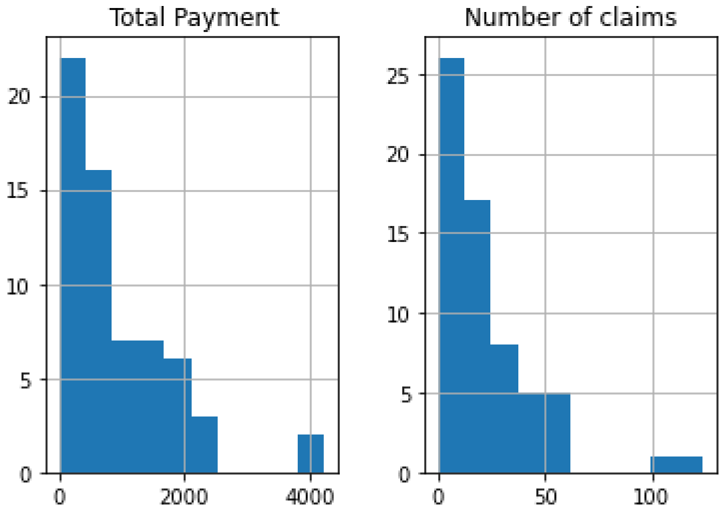
In [7]:
```
data.describe()
```

Out[7]:

	Number of claims	Total Payment
count	63.000000	63.000000
mean	22.904762	921.873016
std	23.351946	878.266281
min	0.000000	0.000000
25%	7.500000	299.000000
50%	14.000000	596.000000
75%	29.000000	1364.000000
max	124.000000	4222.000000

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

Out[8]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000023B9EB04B38>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000023BA0B5FE10>]],
      dtype=object)
```



In [9]:

```
data.corr()
```

Out[9]:

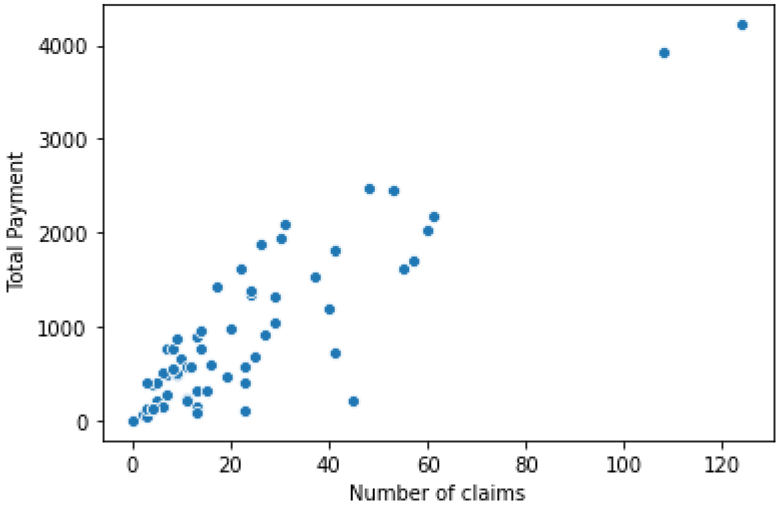
	Number of claims	Total Payment
Number of claims	1.000000	0.880668
Total Payment	0.880668	1.000000

In [13]:

```
sns.scatterplot(data['Number of claims'],data[' Total Payment'])
```

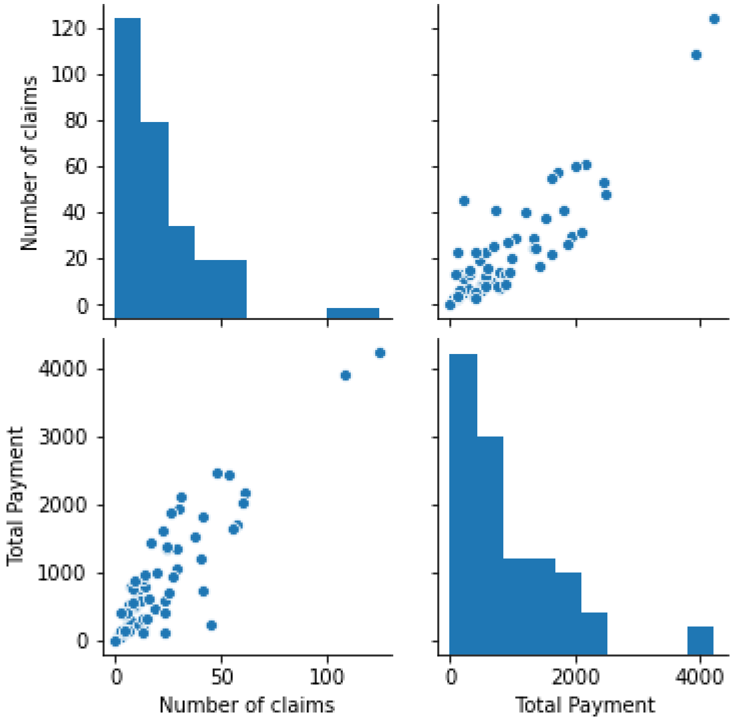
Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23ba0e6c0b8>
```



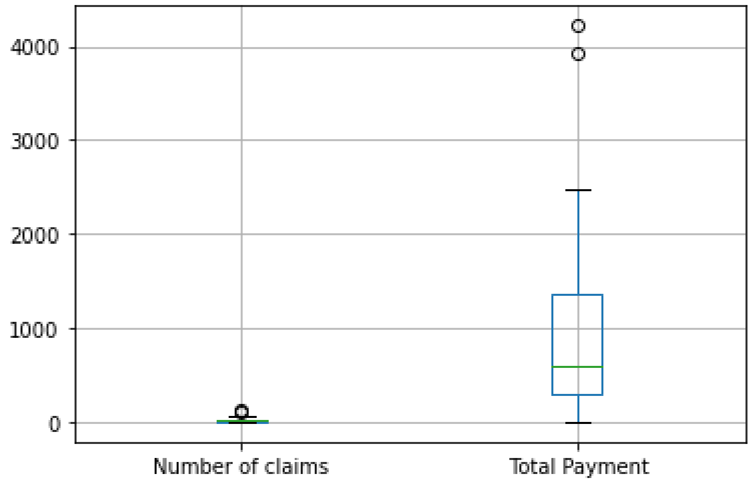
In [14]:
```
sns.pairplot(data)
```

Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x23ba0ecfe48>
```



In [15]:
```
data.boxplot()
```

Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x23ba0fe53c8>
```



In [16]:
```
data.head(1)
```

Out[16]:

	Number of claims	Total Payment
0	108	3925

In [45]:
```
x=data.iloc[:,:1]
```

In [46]:
```
x.head(1)
```

Out[46]:

	Number of claims
0	108

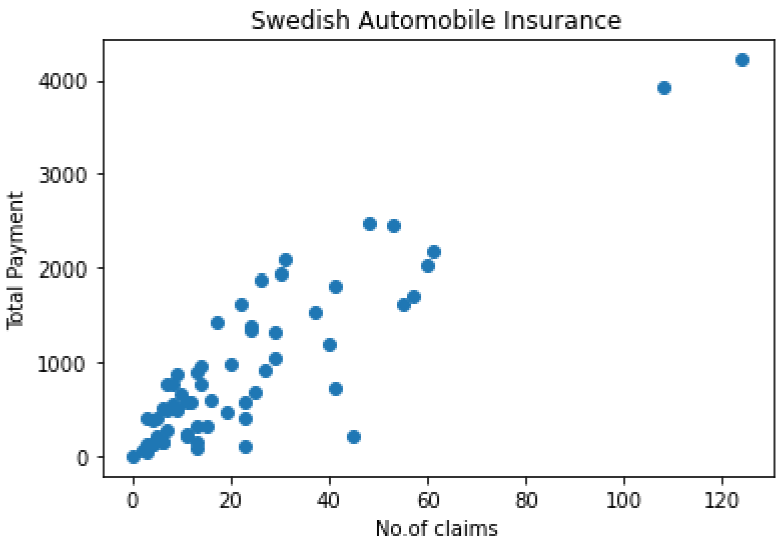
In [47]:
```
y=data.iloc[:,1:2]
```

In [48]:
```
y.head(1)
```

Out[48]:

	Total Payment
0	3925

In [49]:
```
plt.scatter(x,y)
plt.title('Swedish Automobile Insurance')
plt.xlabel('No.of claims')
plt.ylabel('Total Payment')
plt.show()
```



In [50]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [51]:
```
data.shape
```

Out[51]:
```
(63, 2)
```

In [52]:
```
X_train.shape
```

Out[52]:
```
(50, 1)
```

In [53]:
```
X_test.shape
```

Out[53]:
```
(13, 1)
```

In [54]:
```
y_test.shape
```

Out[54]:
```
(13, 1)
```

In [55]:
```
y_train.shape
```

Out[55]:
```
(50, 1)
```

In [56]:
```
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
```

In [57]:
```
#Train the model
lin_reg.fit(X_train,y_train)
```

Out[57]:
```
LinearRegression()
```

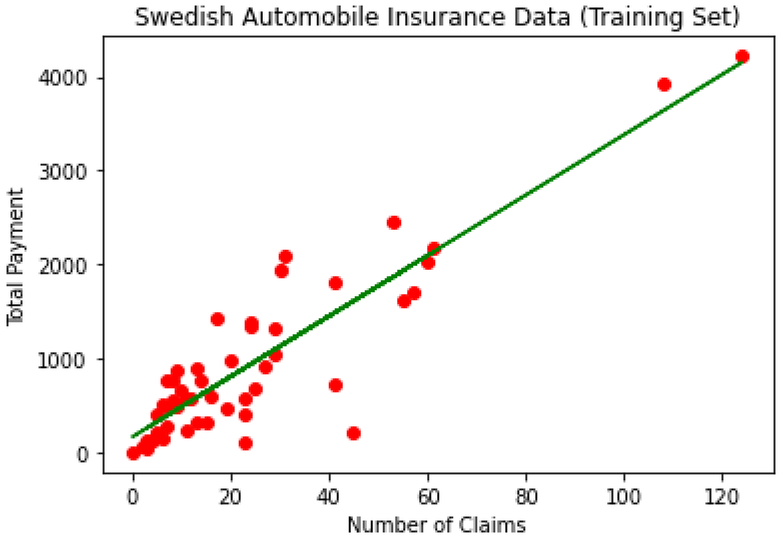
In [58]:
```
lin_reg.coef_
```

Out[58]:
```
array([[32.18742386]])
```

In [59]:
```
lin_reg.intercept_
```

Out[59]:
```
array([164.67433032])
```

In [60]:
```
plt.scatter(X_train,y_train,color='Red')
plt.plot(X_train,lin_reg.predict(X_train),color='green')
plt.title('Swedish Automobile Insurance Data (Training Set)')
plt.xlabel('Number of Claims')
plt.ylabel('Total Payment')
plt.show()
```



In [61]:
```
#Test the model
ypred=lin_reg.predict(X_test)
ypred
```

Out[61]:
```
array([[1355.60901327],
       [ 261.23660191],
       [1001.54735077],
       [ 583.11084055],
       [ 357.7988735 ],
       [ 872.79765532],
       [ 293.42402578],
       [ 518.73599282],
       [1709.67067577],
       [ 583.11084055],
       [ 293.42402578],
       [ 615.29826441],
       [1452.17128486]])
```

In [62]:
```
X_test
```

Out[62]:

	Number of claims
43	37
49	3
62	26
51	13
32	6
34	22
29	4
46	11
11	48
2	13
26	4
60	14
4	40

In [63]:

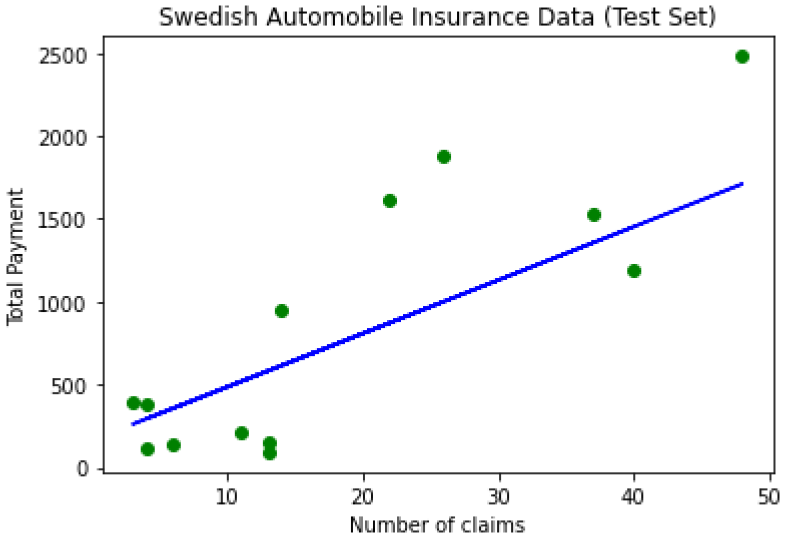
```
data.head()
```

Out[63]:

	Number of claims	Total Payment
0	108	3925
1	19	462
2	13	157
3	124	4222
4	40	1194

In [64]:

```
plt.scatter(X_test,y_test,color='green')
plt.plot(X_test,lin_reg.predict(X_test),color='blue')
plt.title('Swedish Automobile Insurance Data (Test Set)')
plt.xlabel('Number of claims')
plt.ylabel('Total Payment')
plt.show()
```



In [65]:

```
#Estimate the cost
from sklearn.metrics import mean_squared_error, r2_score
RMSE=np.sqrt(mean_squared_error(y_test,ypred))
r_square=r2_score(y_test,ypred)
print('The R-square value is: ',r_square)
print('The RMSE value is: ',RMSE)
```

```
The R-square value is:  0.6488832597282386
The RMSE value is:  458.93148052499384
```

In [74]:

```
#How to predict for unseen value
unseen_pred=lin_reg.predict(np.array([[3234]]))
print('The unseen for the given x is: ',unseen_pred)
```

```
The unseen for the given x is:  [[104258.80310468]]
```

In [ ]: