In [32]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [33]:
```
data=pd.read_csv(r"C:\Users\user\Downloads\6_Salesworkload1 - 6_Salesworkload1
data
```

Out[33]:

| | MonthYear | Time index | Country | StoreID | City | Dept_ID | Dept. Name | HoursOwn | HoursLe |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 1.0 | Dry | 3184.764 | |
| **1** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 2.0 | Frozen | 1582.941 | |
| **2** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 3.0 | other | 47.205 | |
| **3** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 4.0 | Fish | 1623.852 | |
| **4** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 5.0 | Fruits & Vegetables | 1759.173 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **7653** | 6.2017 | 9.0 | Sweden | 29650.0 | Gothenburg | 12.0 | Checkout | 6322.323 | |
| **7654** | 6.2017 | 9.0 | Sweden | 29650.0 | Gothenburg | 16.0 | Customer Services | 4270.479 | |
| **7655** | 6.2017 | 9.0 | Sweden | 29650.0 | Gothenburg | 11.0 | Delivery | 0 | |
| **7656** | 6.2017 | 9.0 | Sweden | 29650.0 | Gothenburg | 17.0 | others | 2224.929 | |
| **7657** | 6.2017 | 9.0 | Sweden | 29650.0 | Gothenburg | 18.0 | all | 39652.2 | |

7658 rows × 14 columns

In [34]:
```
df=data.head(100)
df
```

Out[34]:

| | MonthYear | Time index | Country | StoreID | City | Dept_ID | Dept. Name | HoursOwn | HoursLease |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 1.0 | Dry | 3184.764 | 0.0 |
| **1** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 2.0 | Frozen | 1582.941 | 0.0 |
| **2** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 3.0 | other | 47.205 | 0.0 |
| **3** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 4.0 | Fish | 1623.852 | 0.0 |
| **4** | 10.2016 | 1.0 | United Kingdom | 88253.0 | London (I) | 5.0 | Fruits & Vegetables | 1759.173 | 0.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **95** | 10.2016 | 1.0 | United Kingdom | 18808.0 | London (II) | 14.0 | Non Food | 7817.148 | 0.0 |
| **96** | 10.2016 | 1.0 | United Kingdom | 18808.0 | London (II) | 15.0 | Admin | 5110.728 | 0.0 |
| **97** | 10.2016 | 1.0 | United Kingdom | 18808.0 | London (II) | 12.0 | Checkout | 6209.031 | 0.0 |
| **98** | 10.2016 | 1.0 | United Kingdom | 18808.0 | London (II) | 16.0 | Customer Services | 3115.53 | 0.0 |
| **99** | 10.2016 | 1.0 | United Kingdom | 18808.0 | London (II) | 11.0 | Delivery | 7209.777 | 246.0 |

100 rows × 14 columns

In [35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   MonthYear      100 non-null    object 
 1   Time index     100 non-null    float64
 2   Country        100 non-null    object 
 3   StoreID        100 non-null    float64
 4   City           100 non-null    object 
 5   Dept_ID        100 non-null    float64
 6   Dept. Name     100 non-null    object 
 7   HoursOwn       100 non-null    object 
 8   HoursLease     100 non-null    float64
 9   Sales units    100 non-null    float64
 10  Turnover       100 non-null    float64
 11  Customer       0 non-null      float64
 12  Area (m2)      100 non-null    object 
 13  Opening hours  100 non-null    object 
dtypes: float64(7), object(7)
memory usage: 11.1+ KB
```

In [36]:
```
df.describe()
```

Out[36]:

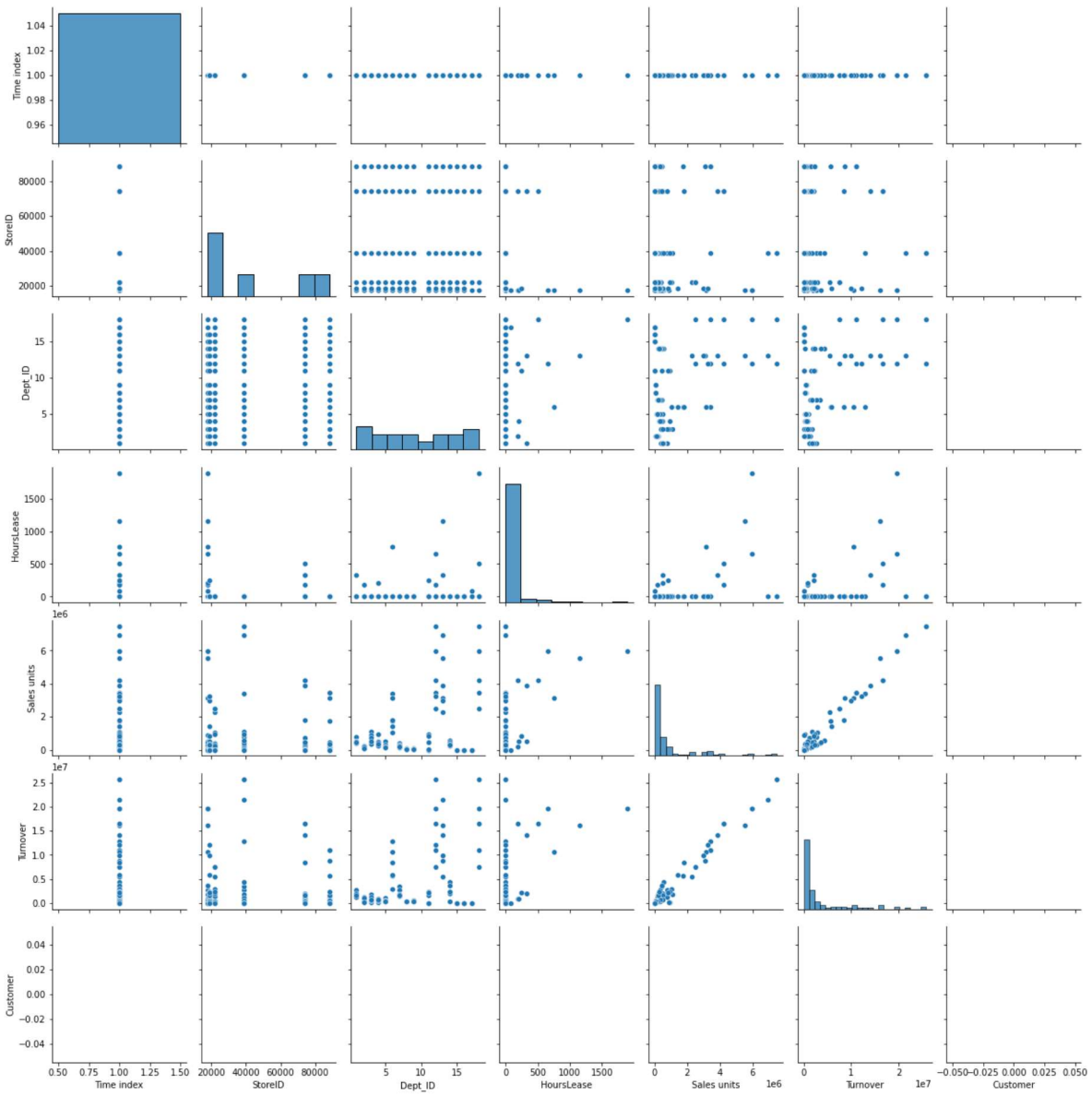
| | Time index | StoreID | Dept_ID | HoursLease | Sales units | Turnover | Customer |
|---|---|---|---|---|---|---|---|
| **count** | 100.0 | 100.000000 | 100.000000 | 100.000000 | 1.000000e+02 | 1.000000e+02 | 0.0 |
| **mean** | 1.0 | 43781.340000 | 9.310000 | 65.340000 | 1.063110e+06 | 3.590811e+06 | NaN |
| **std** | 0.0 | 28146.505061 | 5.292829 | 249.349222 | 1.769242e+06 | 5.968009e+06 | NaN |
| **min** | 1.0 | 17647.000000 | 1.000000 | 0.000000 | 0.000000e+00 | 0.000000e+00 | NaN |
| **25%** | 1.0 | 18808.000000 | 5.000000 | 0.000000 | 5.300125e+04 | 2.702460e+05 | NaN |
| **50%** | 1.0 | 38976.000000 | 9.000000 | 0.000000 | 3.072850e+05 | 8.339250e+05 | NaN |
| **75%** | 1.0 | 73949.000000 | 14.000000 | 0.000000 | 9.195138e+05 | 2.966010e+06 | NaN |
| **max** | 1.0 | 88253.000000 | 18.000000 | 1896.000000 | 7.476680e+06 | 2.571973e+07 | NaN |

In [37]:
```
df.columns
```

Out[37]:
```
Index(['MonthYear', 'Time index', 'Country', 'StoreID', 'City', 'Dept_ID',
       'Dept. Name', 'HoursOwn', 'HoursLease', 'Sales units', 'Turnover',
       'Customer', 'Area (m2)', 'Opening hours'],
      dtype='object')
```

In [38]:
```
sns.pairplot(df)
```

Out[38]:
```
<seaborn.axisgrid.PairGrid at 0x1d790da5700>
```



In [10]:
```
da=data[['Time index', 'StoreID', 'Dept_ID', 'HoursOwn', 'HoursLease', 'Sales
       'Customer']]
da
```

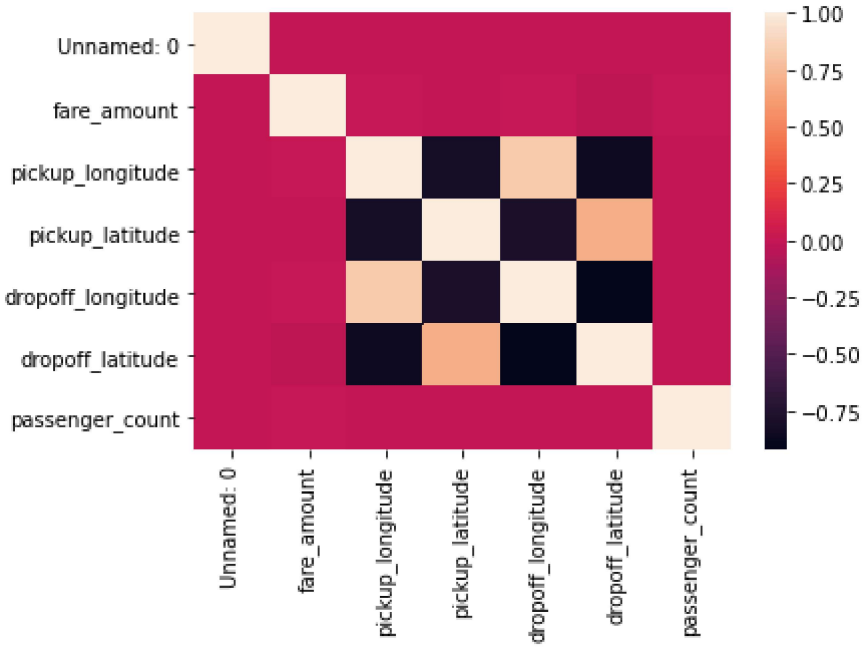
Out[10]:

| | Unnamed: 0 | fare_amount | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_lat |
|---|---|---|---|---|---|---|
| **0** | 24238194 | 7.5 | -73.999817 | 40.738354 | -73.999512 | 40.7 |
| **1** | 27835199 | 7.7 | -73.994355 | 40.728225 | -73.994710 | 40.7 |
| **2** | 44984355 | 12.9 | -74.005043 | 40.740770 | -73.962565 | 40.7 |
| **3** | 25894730 | 5.3 | -73.976124 | 40.790844 | -73.965316 | 40.8 |
| **4** | 17610152 | 16.0 | -73.925023 | 40.744085 | -73.973082 | 40.7 |
| **...** | ... | ... | ... | ... | ... | |
| **199995** | 42598914 | 3.0 | -73.987042 | 40.739367 | -73.986525 | 40.7 |
| **199996** | 16382965 | 7.5 | -73.984722 | 40.736837 | -74.006672 | 40.7 |
| **199997** | 27804658 | 30.9 | -73.986017 | 40.756487 | -73.858957 | 40.6 |
| **199998** | 20259894 | 14.5 | -73.997124 | 40.725452 | -73.983215 | 40.6 |
| **199999** | 11951496 | 14.1 | -73.984395 | 40.720077 | -73.985508 | 40.7 |

200000 rows × 7 columns

In [40]:
```
sns.heatmap(da.corr())
```

Out[40]: <AxesSubplot:>



In [43]:
```
x=da[['Unnamed: 0', 'fare_amount']]
y=da['passenger_count']
```

In [44]:
```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [45]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[45]:
```
LinearRegression()
```

In [46]:
```
print(lr.intercept_)
```

```
1.6656992560926966
```

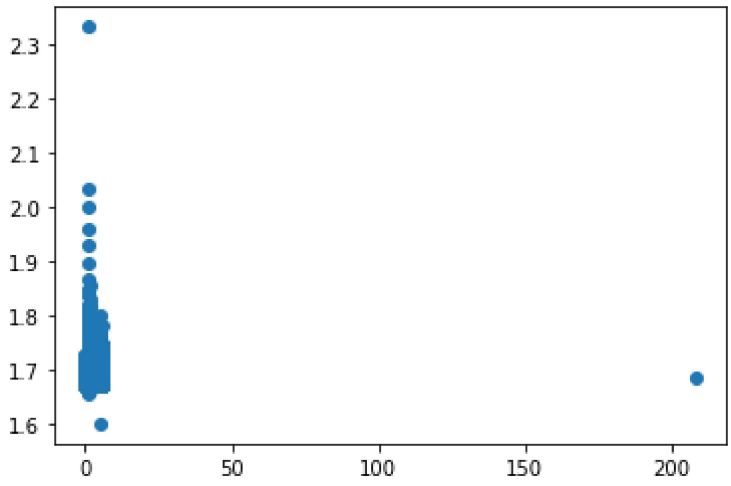
In [47]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[47]:

| | Co-efficient |
|---|---|
| **Unnamed: 0** | 7.037157e-11 |
| **fare_amount** | 1.331269e-03 |

In [48]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[48]:
```
<matplotlib.collections.PathCollection at 0x1d794f36550>
```



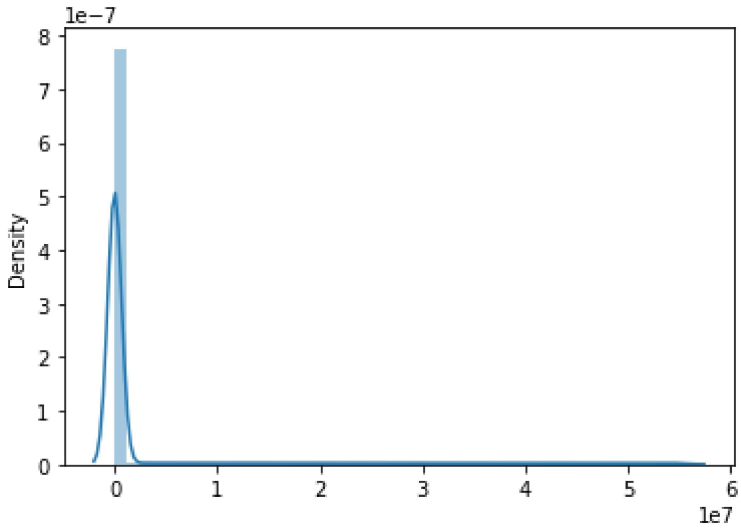

In [49]:
```
print(lr.score(x_test,y_test))
```

```
9.845960967613 81e-05
```

In [50]:
```
sns.distplot(da)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

Out[50]: <AxesSubplot:ylabel='Density'>




In [ ]: