

In [1]:

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

# DATA COLLECTION

In [2]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\6_Salesworkload1 - 6_Salesworkload1.csv")
```

In [3]:

```
b=a.head(10)
b
```

Out[3]:

	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
5	10.2016	1.0	United Kingdom	88253.0	London (I)	6.0	Meat	8270.316	0.0
6	10.2016	1.0	United Kingdom	88253.0	London (I)	13.0	Food	16468.251	0.0
7	10.2016	1.0	United Kingdom	88253.0	London (I)	7.0	Clothing	4698.471	0.0
8	10.2016	1.0	United Kingdom	88253.0	London (I)	8.0	Household	1183.272	0.0
9	10.2016	1.0	United Kingdom	88253.0	London (I)	9.0	Hardware	2029.815	0.0

# DATA CLEANING AND PRE-PROCESSING

In [4]:

```
b.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   MonthYear       10 non-null    object
 1   Time index      10 non-null    float64
 2   Country         10 non-null    object
 3   StoreID         10 non-null    float64
 4   City            10 non-null    object
 5   Dept_ID         10 non-null    float64
 6   Dept. Name      10 non-null    object
 7   HoursOwn        10 non-null    object
 8   HoursLease      10 non-null    float64
 9   Sales units     10 non-null    float64
10   Turnover        10 non-null    float64
11   Customer        0 non-null     float64
12   Area (m2)       10 non-null    object
13   Opening hours   10 non-null    object
dtypes: float64(7), object(7)
memory usage: 1.2+ KB
```

In [5]:

```
b.describe()
```

Out[5]:

	Time index	StoreID	Dept_ID	HoursLease	Sales units	Turnover	Customer
count	10.0	10.0	10.000000	10.0	1.000000e+01	1.000000e+01	0.0
mean	1.0	88253.0	5.800000	0.0	6.543725e+05	1.978511e+06	NaN
std	0.0	0.0	3.614784	0.0	9.914003e+05	2.861420e+06	NaN
min	1.0	88253.0	1.000000	0.0	5.491500e+04	2.904000e+05	NaN
25%	1.0	88253.0	3.250000	0.0	1.034225e+05	4.033612e+05	NaN
50%	1.0	88253.0	5.500000	0.0	2.615525e+05	5.770455e+05	NaN
75%	1.0	88253.0	7.750000	0.0	4.284400e+05	1.518067e+06	NaN
max	1.0	88253.0	13.000000	0.0	3.107935e+06	8.714679e+06	NaN

In [6]:

```
c=b.dropna(axis=1)
c
```

Out[6]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease
0	10.2016	1.0	United Kingdom	88253.0	London (l)	1.0	Dry	3184.764	0.0
1	10.2016	1.0	United Kingdom	88253.0	London (l)	2.0	Frozen	1582.941	0.0
2	10.2016	1.0	United Kingdom	88253.0	London (l)	3.0	other	47.205	0.0
3	10.2016	1.0	United Kingdom	88253.0	London (l)	4.0	Fish	1623.852	0.0
4	10.2016	1.0	United Kingdom	88253.0	London (l)	5.0	Fruits & Vegetables	1759.173	0.0
5	10.2016	1.0	United Kingdom	88253.0	London (l)	6.0	Meat	8270.316	0.0
6	10.2016	1.0	United Kingdom	88253.0	London (l)	13.0	Food	16468.251	0.0
7	10.2016	1.0	United Kingdom	88253.0	London (l)	7.0	Clothing	4698.471	0.0
8	10.2016	1.0	United Kingdom	88253.0	London (l)	8.0	Household	1183.272	0.0
9	10.2016	1.0	United Kingdom	88253.0	London (l)	9.0	Hardware	2029.815	0.0

In [7]:

```
c.columns
```

Out[7]:

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

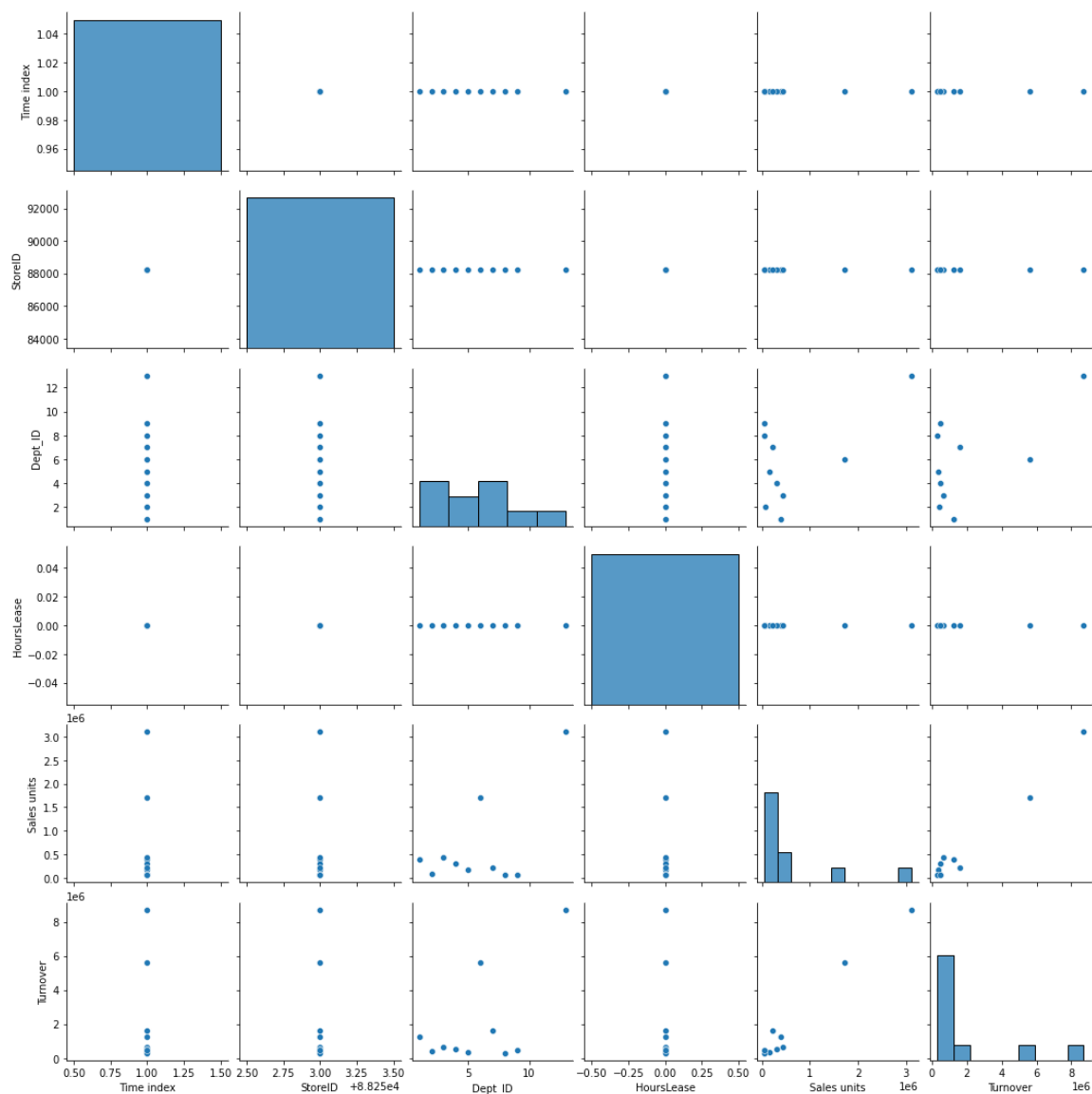
# EDA AND VISUALIZATION

In [8]:

```
sns.pairplot(c)
```

Out[8]:

&lt;seaborn.axisgrid.PairGrid at 0x2019259daf0&gt;

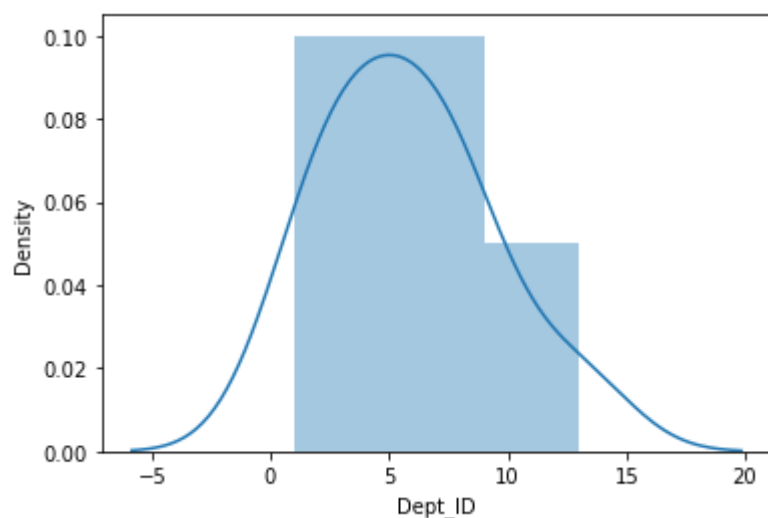


In [10]:

```
sns.distplot(c['Dept_ID'])
```

Out[10]:

<AxesSubplot:xlabel='Dept\_ID', ylabel='Density'>



In [11]:

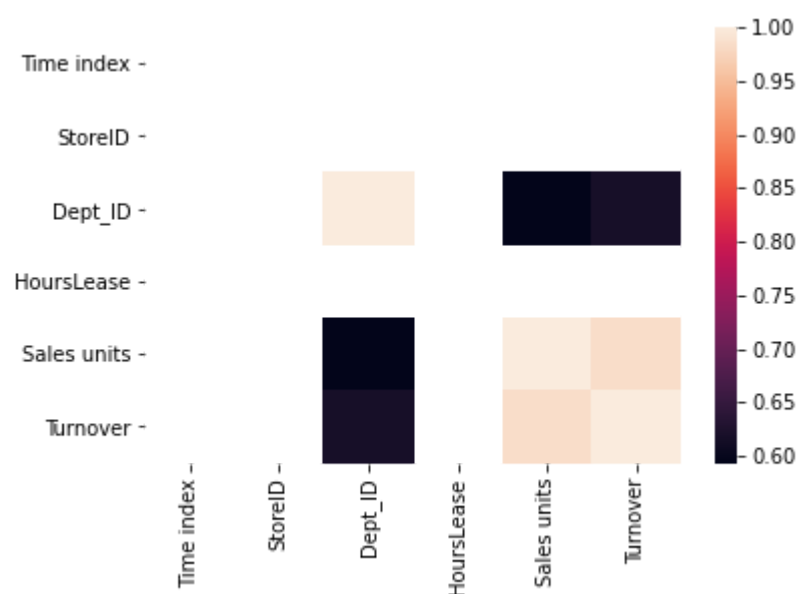
```
f=c[['MonthYear', 'Time index', 'StoreID', 'Dept_ID',  
     'HoursOwn', 'HoursLease', 'Sales units', 'Turnover',  
     'Area (m2)']]
```

In [12]:

```
sns.heatmap(f.corr())
```

Out[12]:

<AxesSubplot:>



In [13]:

```
x=f[['MonthYear', 'Time index', 'StoreID',  
     'HoursOwn', 'HoursLease', 'Sales units', 'Turnover',  
     'Area (m2)']]  
y=f['Dept_ID']
```

In [14]:

```
from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5)
```

In [15]:

```
from sklearn.linear_model import LinearRegression  
  
lr=LinearRegression()  
lr.fit(x_train,y_train)
```

Out[15]:

LinearRegression()

In [16]:

```
print(lr.intercept_)
```

0.9606107485042035

In [17]:

```
r=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])  
r
```

Out[17]:

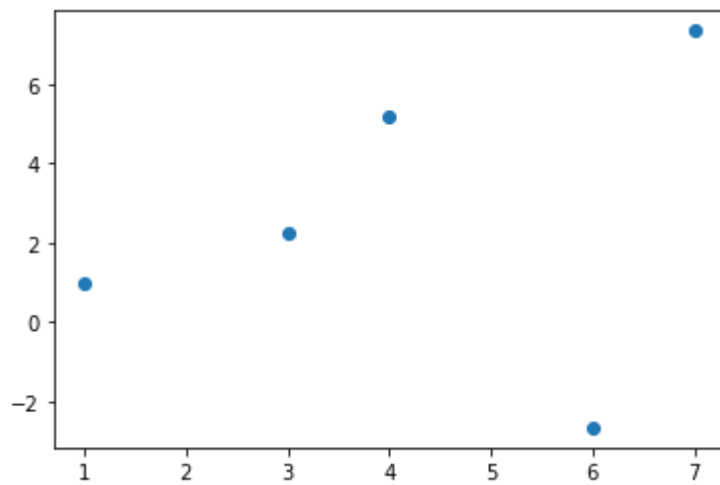
	Co-efficient
<b>MonthYear</b>	-2.000878e-16
<b>Time index</b>	1.988495e-17
<b>StoreID</b>	0.000000e+00
<b>HoursOwn</b>	2.020036e-03
<b>HoursLease</b>	0.000000e+00
<b>Sales units</b>	1.746637e-05
<b>Turnover</b>	-1.204883e-05
<b>Area (m2)</b>	1.484486e-03

In [18]:

```
u=lr.predict(x_test)
plt.scatter(y_test,u)
```

Out[18]:

<matplotlib.collections.PathCollection at 0x20195de2ca0>



In [19]:

```
print(lr.score(x_test,y_test))
```

-2.38650401823929

In [ ]: