

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

```
In [2]: df=pd.read_csv(r"C:\Users\user\Downloads\6_Salesworkload1.csv")
df.fillna(0,inplace=True)
df
```

Out[2]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLea
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	06.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	
7654	06.2017	9.0	Sweden	29650.0	Gothenburg	16.0	Customer Services	4270.479	
7655	06.2017	9.0	Sweden	29650.0	Gothenburg	11.0	Delivery	0	
7656	06.2017	9.0	Sweden	29650.0	Gothenburg	17.0	others	2224.929	
7657	06.2017	9.0	Sweden	29650.0	Gothenburg	18.0	all	39652.2	

7658 rows × 14 columns



```
In [3]: df.head()
```

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 39:
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	0.0 8:
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	0.0 43:
3	10.2016	1.0	United Kingdom	88253.0	London (I)	4.0	Fish	1623.852	0.0 30:
4	10.2016	1.0	United Kingdom	88253.0	London (I)	5.0	Fruits & Vegetables	1759.173	0.0 16:



In [4]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7658 entries, 0 to 7657
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
0   MonthYear       7658 non-null   object
1   Time index      7658 non-null   float64
2   Country         7658 non-null   object
3   StoreID         7658 non-null   float64
4   City            7658 non-null   object
5   Dept_ID         7658 non-null   float64
6   Dept. Name      7658 non-null   object
7   HoursOwn        7658 non-null   object
8   HoursLease      7658 non-null   float64
9   Sales units     7658 non-null   float64
10  Turnover        7658 non-null   float64
11  Customer        7658 non-null   float64
12  Area (m2)       7658 non-null   object
13  Opening hours   7658 non-null   object
dtypes: float64(7), object(7)
memory usage: 837.7+ KB
```

In [5]: `import` seaborn `as` sns

In [6]: df.describe()

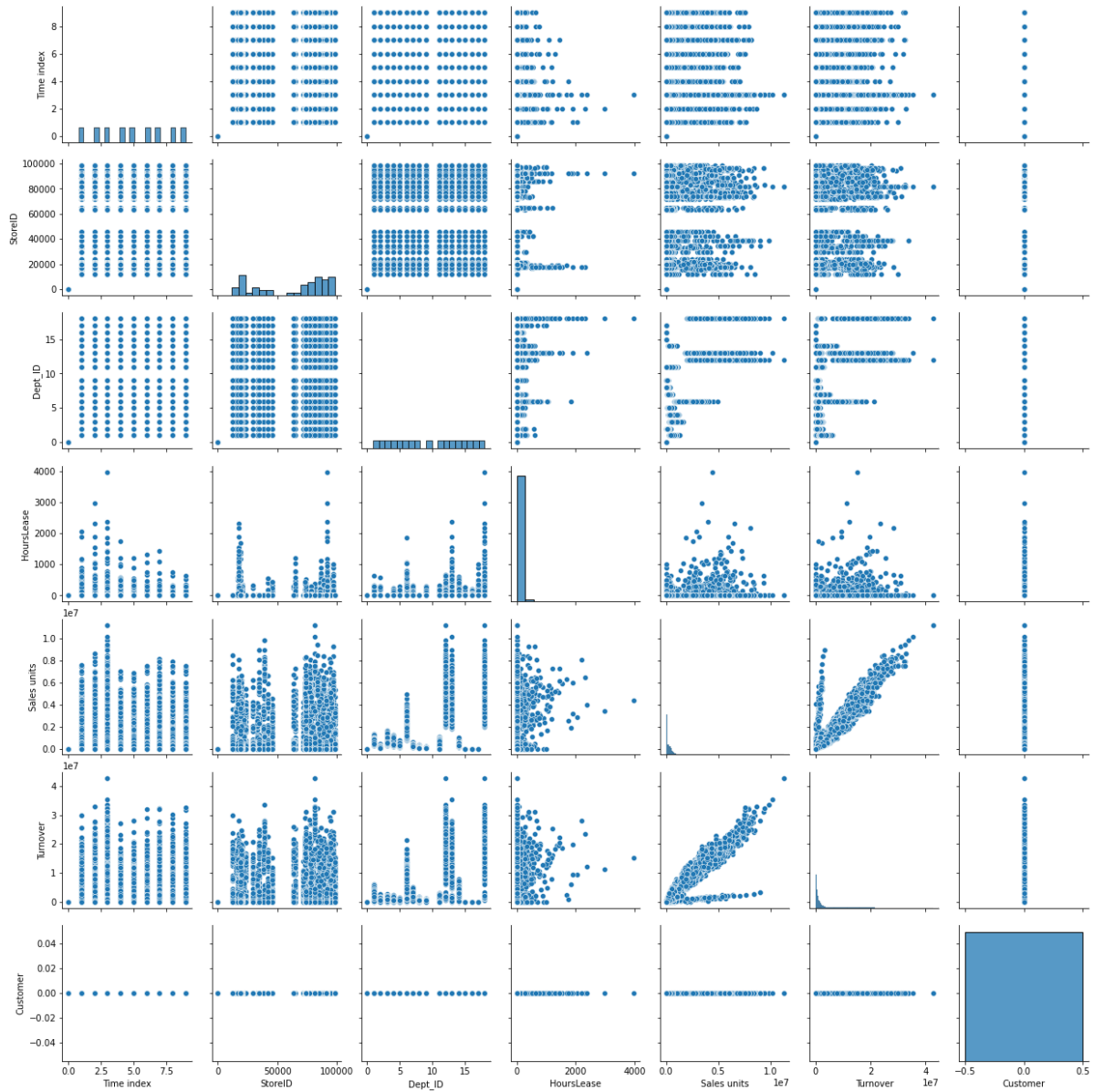
Out[6]:

	Time index	StoreID	Dept_ID	HoursLease	Sales units	Turnover	Customer
count	7658.000000	7658.000000	7658.000000	7658.000000	7.658000e+03	7.658000e+03	7658.000000
mean	4.994777	61930.456124	9.460695	22.013058	1.075346e+06	3.717505e+06	1.075346e+06
std	2.585859	29975.929873	5.343407	133.231761	1.727560e+06	6.001448e+06	1.727560e+06
min	0.000000	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00
25%	3.000000	29650.000000	5.000000	0.000000	5.441375e+04	2.720558e+05	5.441375e+04
50%	5.000000	73949.000000	9.000000	0.000000	2.927625e+05	9.300810e+05	2.927625e+05
75%	7.000000	87703.000000	14.000000	0.000000	9.154812e+05	3.251488e+06	9.154812e+05
max	9.000000	98422.000000	18.000000	3984.000000	1.124296e+07	4.271739e+07	1.124296e+07

In [ ]:

```
In [7]: sns.pairplot(df)
```

```
Out[7]: <seaborn.axisgrid.PairGrid at 0x151b2691550>
```

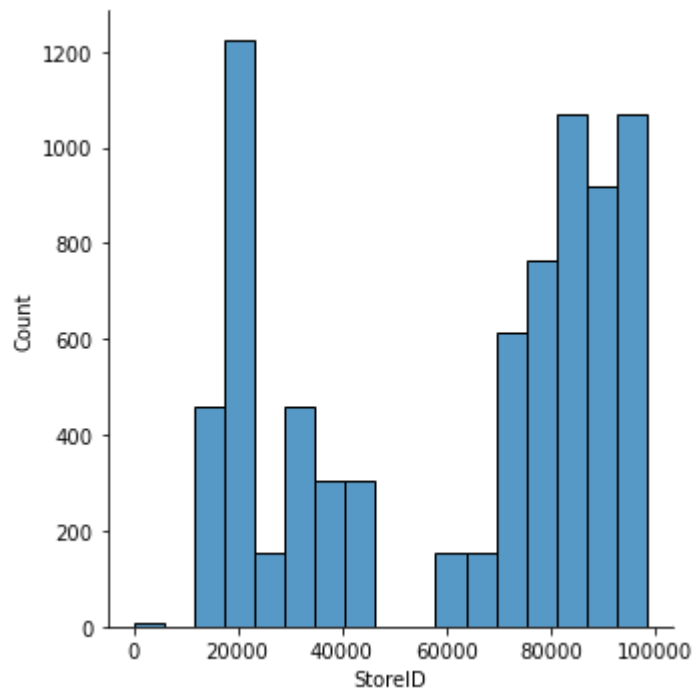


```
In [8]: df1=df.drop(['Country'],axis=1)
df1
df1=df1.drop(df1.index[1537:])
df1.isna().sum()
```

```
Out[8]: MonthYear      0
Time index      0
StoreID         0
City           0
Dept_ID        0
Dept. Name      0
HoursOwn        0
HoursLease      0
Sales units     0
Turnover        0
Customer        0
Area (m2)       0
Opening hours   0
dtype: int64
```

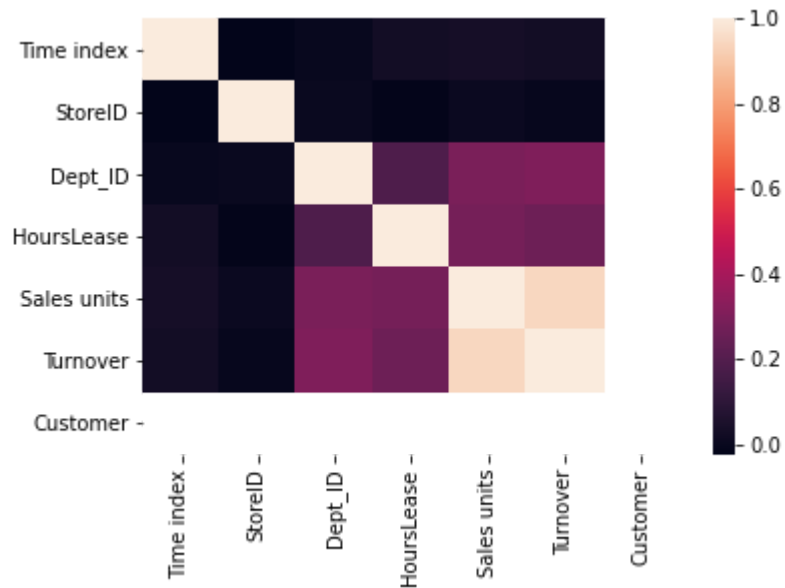
```
In [9]: sns.displot(df['StoreID'])
```

```
Out[9]: <seaborn.axisgrid.FacetGrid at 0x151b9ea71f0>
```



```
In [10]: sns.heatmap(df1.corr())
```

```
Out[10]: <AxesSubplot:>
```



```
In [11]: from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LinearRegression
```

```
In [12]: df1.isna().sum()
```

```
Out[12]: MonthYear      0  
Time index      0  
StoreID         0  
City            0  
Dept_ID         0  
Dept. Name      0  
HoursOwn        0  
HoursLease      0  
Sales units     0  
Turnover        0  
Customer        0  
Area (m2)       0  
Opening hours   0  
dtype: int64
```

```
In [24]: y=df1['Turnover']
x=df1.drop(['Turnover','MonthYear','City','Opening hours','Dept. Name','CustomerID'])
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

	Time index	StoreID	Dept_ID	HoursOwn	HoursLease	Sales units
196	1.0	95434.0	9.0	2158.842	0.0	63985.0
1191	2.0	87703.0	1.0	2627.745	0.0	498005.0
1216	2.0	19000.0	8.0	2102.196	0.0	87440.0
1189	2.0	96857.0	17.0	31.47	996.0	95.0
1352	2.0	64983.0	8.0	2778.801	0.0	113880.0
...	...	...	...	...	...	...
653	1.0	98422.0	7.0	5088.699	0.0	234040.0
167	1.0	19769.0	11.0	2895.24	0.0	465450.0
77	1.0	73949.0	9.0	874.866	0.0	53330.0
1410	2.0	83160.0	17.0	2416.896	0.0	160.0
556	1.0	83160.0	12.0	6303.441	0.0	5908995.0

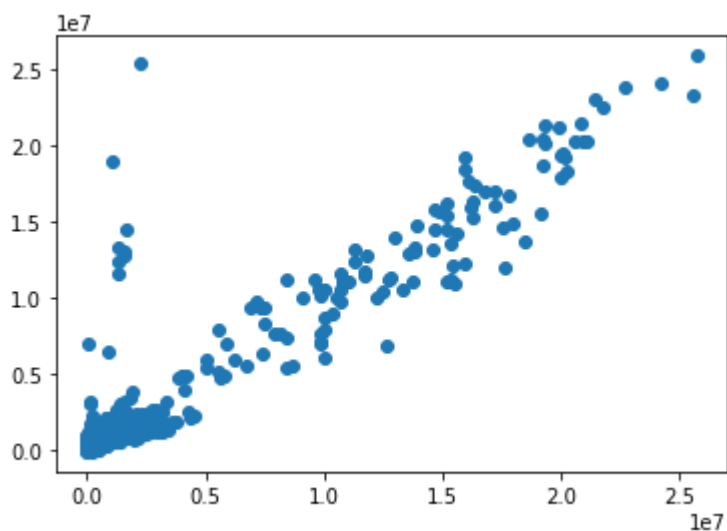
[1075 rows x 6 columns]

```
In [25]: model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

Out[25]: 304904.29054210894

```
In [26]: prediction=model.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[26]: <matplotlib.collections.PathCollection at 0x151bb81e070>



```
In [27]: model.score(x_test,y_test)
```

Out[27]: 0.8595662480283945

```
In [28]: from sklearn.linear_model import Ridge,Lasso
```

```
In [29]: rr=Ridge(alpha=10)  
rr.fit(x_train,y_train)
```

```
Out[29]: Ridge(alpha=10)
```

```
In [30]: rr.score(x_test,y_test)
```

```
Out[30]: 0.859566766825292
```

```
In [31]: la =Lasso(alpha=10)  
la.fit(x_train,y_train)
```

```
Out[31]: Lasso(alpha=10)
```

```
In [32]: la.score(x_test,y_test)
```

```
Out[32]: 0.8595662619232616
```