

Data visualization

```
In [1]: import numpy as np
import pandas as pd
```

Import dataset

```
In [2]: data=pd.read_csv(r"C:\Users\user\Downloads\6_Salesworkload1.csv")
```

print data

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

Out[3]:

	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



print first 10 rows using head

In [4]: `data.head(10)`

Out[4]:

	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	31
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	0.0	1
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	0.0	41
3	10.2016	1.0	United Kingdom	88253.0	London (I)	4.0	Fish	1623.852	0.0	31
4	10.2016	1.0	United Kingdom	88253.0	London (I)	5.0	Fruits & Vegetables	1759.173	0.0	11
5	10.2016	1.0	United Kingdom	88253.0	London (I)	6.0	Meat	8270.316	0.0	17
6	10.2016	1.0	United Kingdom	88253.0	London (I)	13.0	Food	16468.251	0.0	311
7	10.2016	1.0	United Kingdom	88253.0	London (I)	7.0	Clothing	4698.471	0.0	2
8	10.2016	1.0	United Kingdom	88253.0	London (I)	8.0	Household	1183.272	0.0	1
9	10.2016	1.0	United Kingdom	88253.0	London (I)	9.0	Hardware	2029.815	0.0	1

print last 10 rows using tail

In [5]: `data.tail(5)`

Out[5]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLeas
7653	06.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	0.
7654	06.2017	9.0	Sweden	29650.0	Gothenburg	16.0	Customer Services	4270.479	0.
7655	06.2017	9.0	Sweden	29650.0	Gothenburg	11.0	Delivery	0	0.
7656	06.2017	9.0	Sweden	29650.0	Gothenburg	17.0	others	2224.929	0.
7657	06.2017	9.0	Sweden	29650.0	Gothenburg	18.0	all	39652.2	0.

print describe of dataset

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

Out[6]:

	Time index	StoreID	Dept_ID	HoursLease	Sales units	Turnover	Custome
count	7650.000000	7650.000000	7650.000000	7650.000000	7.650000e+03	7.650000e+03	(
mean	5.000000	61995.220000	9.470588	22.036078	1.076471e+06	3.721393e+06	N
std	2.582158	29924.581631	5.337429	133.299513	1.728113e+06	6.003380e+06	N
min	1.000000	12227.000000	1.000000	0.000000	0.000000e+00	0.000000e+00	N
25%	3.000000	29650.000000	5.000000	0.000000	5.457125e+04	2.726798e+05	N
50%	5.000000	75400.500000	9.000000	0.000000	2.932300e+05	9.319575e+05	N
75%	7.000000	87703.000000	14.000000	0.000000	9.175075e+05	3.264432e+06	N
max	9.000000	98422.000000	18.000000	3984.000000	1.124296e+07	4.271739e+07	N

Number elements in dataset

```
In [7]: data.size
```

Out[7]: 107212

print shape of dataset

```
In [8]: data.shape
```

Out[8]: (7658, 14)

print empty or not

In [9]: `data.isna()`

Out[9]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	Sales units
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...
7653	False	False	False	False	False	False	False	False	False	False
7654	False	False	False	False	False	False	False	False	False	False
7655	False	False	False	False	False	False	False	False	False	False
7656	False	False	False	False	False	False	False	False	False	False
7657	False	False	False	False	False	False	False	False	False	False

7658 rows × 14 columns



In [10]: `data.isnull().sum()`

Out[10]:

MonthYear	0
Time index	8
Country	8
StoreID	8
City	8
Dept_ID	8
Dept. Name	8
HoursOwn	8
HoursLease	8
Sales units	8
Turnover	8
Customer	7658
Area (m2)	8
Opening hours	8
dtype:	int64

```
In [16]: data1 = data.fillna(value=10)
data1
```

Out[16]:

	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 [17]: data1.isnull().sum()
```

```
Out[17]: MonthYear      0
Time index      0
Country         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
```

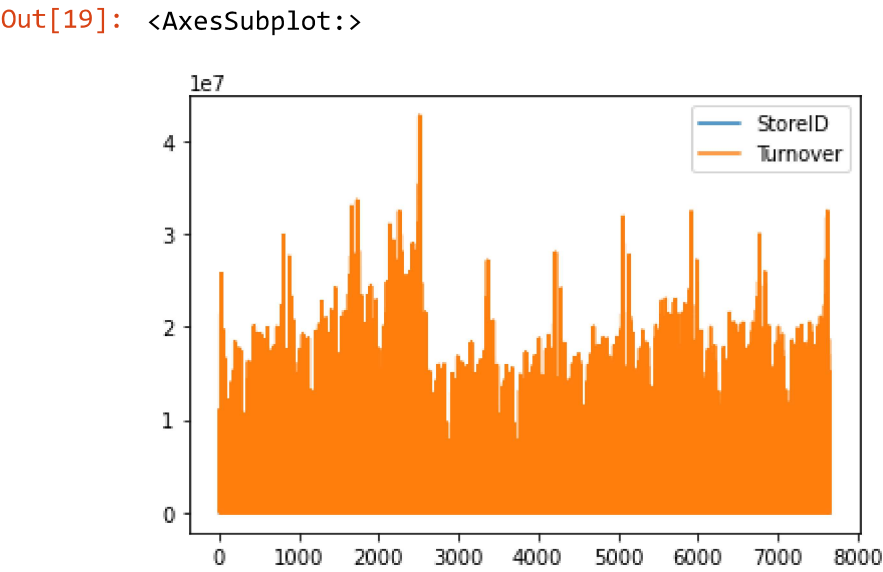
```
In [18]: data1 = data[["StoreID", "Turnover"]]  
data1
```

Out[18]:

	StoreID	Turnover
0	88253.0	1226244.0
1	88253.0	387810.0
2	88253.0	654657.0
3	88253.0	499434.0
4	88253.0	329397.0
...
7653	29650.0	14538825.0
7654	29650.0	0.0
7655	29650.0	0.0
7656	29650.0	0.0
7657	29650.0	15056214.0

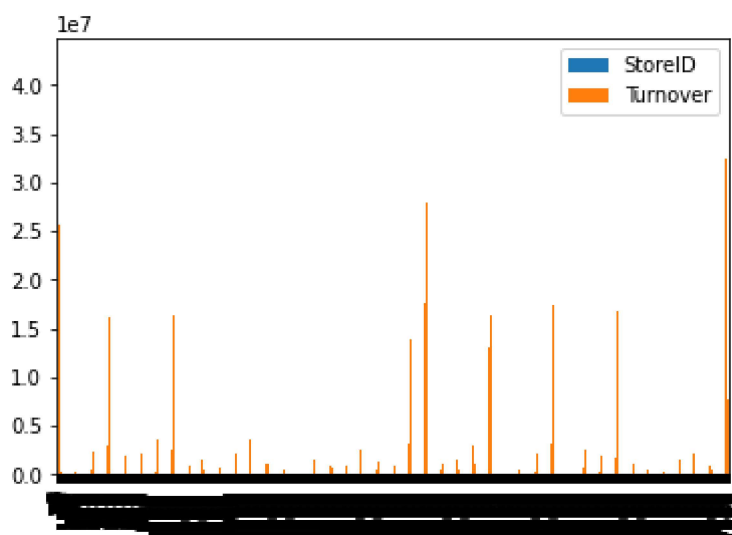
7658 rows × 2 columns

```
In [19]: data1.plot.line()
```



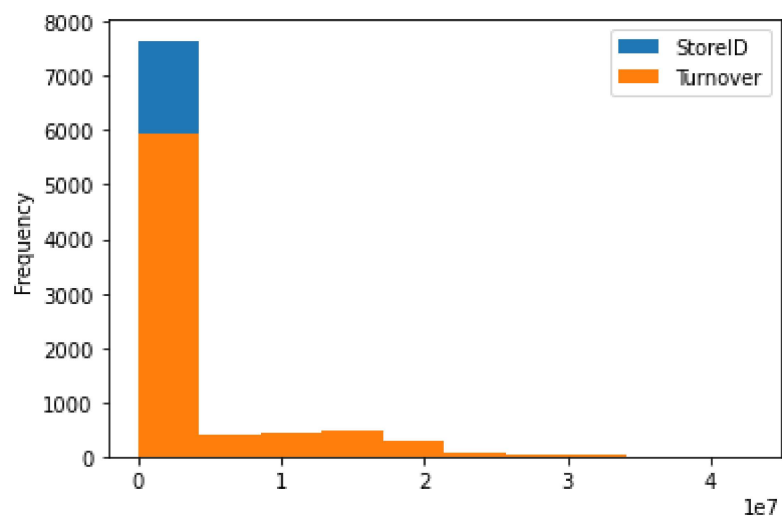
```
In [20]: data1.plot.bar()
```

```
Out[20]: <AxesSubplot:>
```



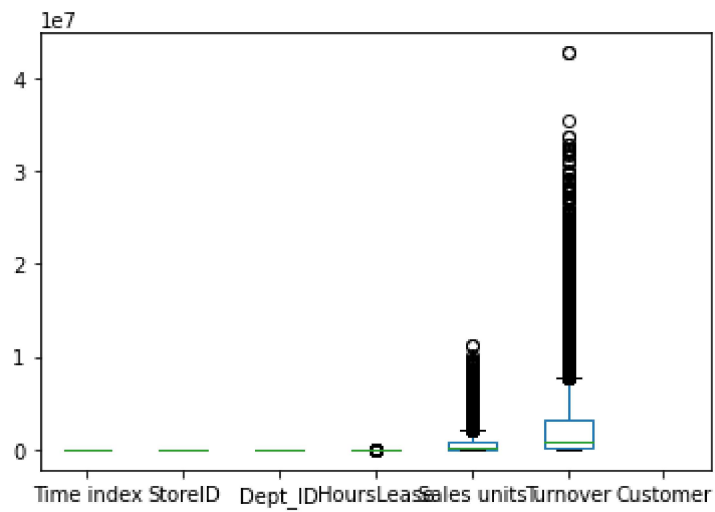
```
In [21]: data1.plot.hist()
```

```
Out[21]: <AxesSubplot:ylabel='Frequency'>
```



```
In [22]: data.plot.box('km')
```

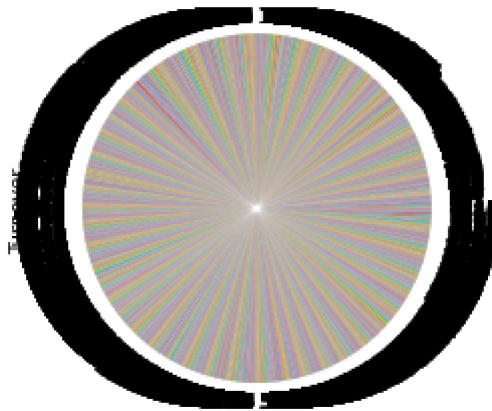
```
Out[22]: <AxesSubplot:>
```



```
In [24]: data2 = data1["Turnover"]
```

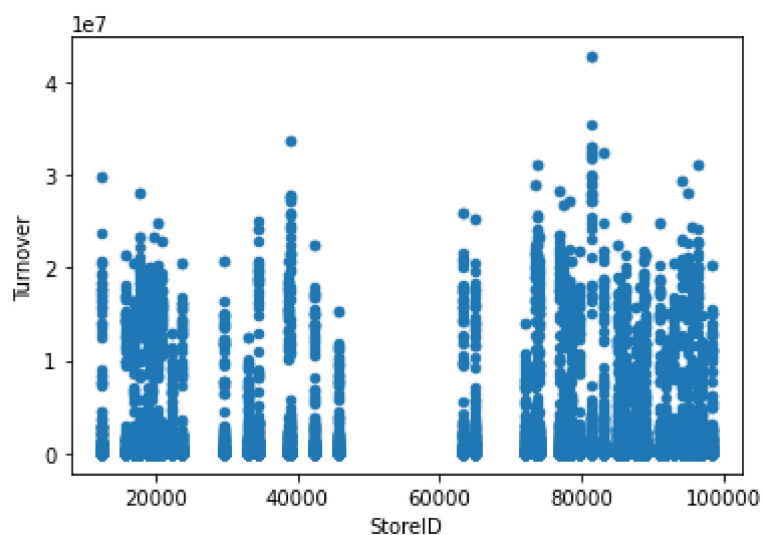
```
In [29]: data2.plot.pie()
```

```
Out[29]: <AxesSubplot:ylabel='Turnover'>
```



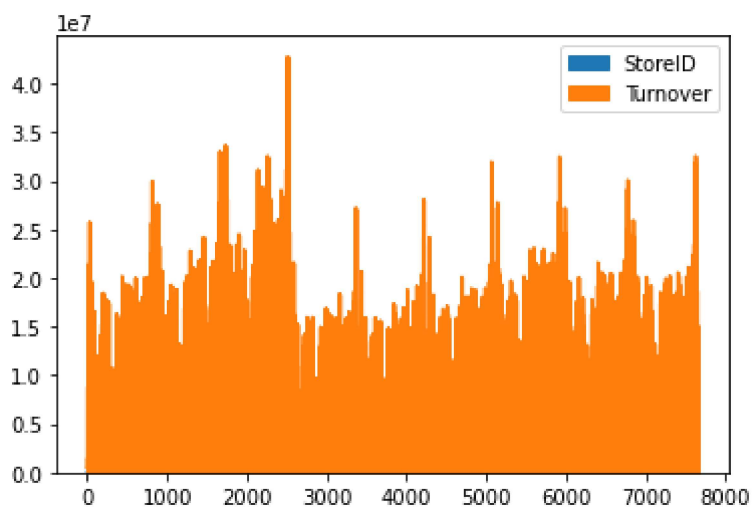

```
In [27]: data1.plot.scatter("StoreID", "Turnover")
```

```
Out[27]: <AxesSubplot:xlabel='StoreID', ylabel='Turnover'>
```



```
In [28]: data1.plot.area()
```

```
Out[28]: <AxesSubplot:>
```



```
In [ ]:
```