In [36]:
```
import numpy as np
import pandas as pd
```

In [41]:
```
x=pd.read_csv(r"C:\Users\user\Downloads\6_Salesworkload1 - 6_Salesworkload1.csv
print(x)
```

```
     MonthYear  Time index         Country  StoreID        City  Dept_ID  \
0      10.2016         1.0  United Kingdom  88253.0  London (I)      1.0
1      10.2016         1.0  United Kingdom  88253.0  London (I)      2.0
2      10.2016         1.0  United Kingdom  88253.0  London (I)      3.0
3      10.2016         1.0  United Kingdom  88253.0  London (I)      4.0
4      10.2016         1.0  United Kingdom  88253.0  London (I)      5.0
...        ...         ...             ...      ...         ...      ...
7653    6.2017         9.0          Sweden  29650.0  Gothenburg     12.0
7654    6.2017         9.0          Sweden  29650.0  Gothenburg     16.0
7655    6.2017         9.0          Sweden  29650.0  Gothenburg     11.0
7656    6.2017         9.0          Sweden  29650.0  Gothenburg     17.0
7657    6.2017         9.0          Sweden  29650.0  Gothenburg     18.0

                Dept. Name  HoursOwn  HoursLease  Sales units    Turnover  \
0                      Dry  3184.764         0.0     398560.0   1226244.0
1                   Frozen  1582.941         0.0      82725.0    387810.0
2                    other    47.205         0.0     438400.0    654657.0
3                     Fish  1623.852         0.0     309425.0    499434.0
4     Fruits & Vegetables  1759.173         0.0     165515.0    329397.0
...                   ...       ...         ...          ...         ...
7653             Checkout  6322.323         0.0    3886530.0  14538825.0
7654    Customer Services  4270.479         0.0        245.0         0.0
7655             Delivery         0         0.0          0.0         0.0
7656               others  2224.929         0.0        245.0         0.0
7657                  all   39652.2         0.0    3886530.0  15056214.0

      Customer Area (m2) Opening hours
0          NaN    953.04        Type A
1          NaN    720.48        Type A
2          NaN    966.72        Type A
3          NaN   1053.36        Type A
4          NaN   1053.36        Type A
...        ...       ...           ...
7653       NaN       #NV        Type A
7654       NaN       #NV        Type A
7655       NaN       #NV        Type A
7656       NaN       #NV        Type A
7657       NaN       #NV        Type A

[7658 rows x 14 columns]
```

In [43]:
```
x=x.head(100)
x
```

Out[43]:

	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	3
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 [44]:
```
x.tail(6)
```

Out[44]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	
94	10.2016	1.0	United Kingdom	18808.0	London (II)	9.0	Hardware	1970.022	0.0	
95	10.2016	1.0	United Kingdom	18808.0	London (II)	14.0	Non Food	7817.148	0.0	3
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	32
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	8

In [45]:
```
x.dtypes
```

Out[45]:
```
MonthYear         object
Time index       float64
Country           object
StoreID          float64
City              object
Dept_ID          float64
Dept. Name        object
HoursOwn          object
HoursLease       float64
Sales units      float64
Turnover         float64
Customer         float64
Area (m2)         object
Opening hours     object
dtype: object
```

In [46]:
```
x.index
```

Out[46]:
```
RangeIndex(start=0, stop=100, step=1)
```

In [47]:
```
x.describe
```

Out[47]:
```
<bound method NDFrame.describe of    MonthYear  Time index         Country  S
toreID         City  Dept_ID  \
0    10.2016         1.0  United Kingdom  88253.0   London (I)      1.0
1    10.2016         1.0  United Kingdom  88253.0   London (I)      2.0
2    10.2016         1.0  United Kingdom  88253.0   London (I)      3.0
3    10.2016         1.0  United Kingdom  88253.0   London (I)      4.0
4    10.2016         1.0  United Kingdom  88253.0   London (I)      5.0
..       ...         ...             ...      ...          ...      ...
95   10.2016         1.0  United Kingdom  18808.0  London (II)     14.0
96   10.2016         1.0  United Kingdom  18808.0  London (II)     15.0
97   10.2016         1.0  United Kingdom  18808.0  London (II)     12.0
98   10.2016         1.0  United Kingdom  18808.0  London (II)     16.0
99   10.2016         1.0  United Kingdom  18808.0  London (II)     11.0

              Dept. Name  HoursOwn  HoursLease  Sales units    Turnover  \
0                    Dry  3184.764         0.0     398560.0   1226244.0
1                 Frozen  1582.941         0.0      82725.0    387810.0
2                  other    47.205         0.0     438400.0    654657.0
3                   Fish  1623.852         0.0     309425.0    499434.0
4   Fruits & Vegetables  1759.173         0.0     165515.0    329397.0
..                   ...       ...         ...          ...         ...
95              Non Food  7817.148         0.0     301500.0   2319717.0
96                 Admin  5110.728         0.0         25.0         0.0
97              Checkout  6209.031         0.0    3262240.0  12161196.0
98    Customer Services   3115.53         0.0         25.0         0.0
99              Delivery  7209.777       246.0     843615.0   2204589.0

    Customer Area (m2) Opening hours
0         NaN    953.04        Type A
1         NaN    720.48        Type A
2         NaN    966.72        Type A
3         NaN   1053.36        Type A
4         NaN   1053.36        Type A
..        ...       ...           ...
95        NaN  18930.84        Type B
96        NaN         0        Type B
97        NaN  32052.24        Type B
98        NaN         0        Type B
99        NaN         0        Type B

[100 rows x 14 columns]>
```

In [48]:
```
x["Saves"]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get
_loc(self, key, method, tolerance)
   3079             try:
-> 3080                 return self._engine.get_loc(casted_key)
   3081             except KeyError as err:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHas
hTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHas
hTable.get_item()

KeyError: 'Saves'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
<ipython-input-48-37ced1545842> in <module>
----> 1 x["Saves"]

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem_
_(self, key)
   3022             if self.columns.nlevels > 1:
   3023                 return self._getitem_multilevel(key)
-> 3024             indexer = self.columns.get_loc(key)
   3025             if is_integer(indexer):
   3026                 indexer = [indexer]

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get
_loc(self, key, method, tolerance)
   3080                 return self._engine.get_loc(casted_key)
   3081             except KeyError as err:
-> 3082                 raise KeyError(key) from err
   3083
   3084         if tolerance is not None:

KeyError: 'Saves'
```

In [ ]:
```
x.loc[1:7]
```

In [49]:
```
x.fillna(value=100)
```

Out[49]:

	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	3
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 [50]:
```
x.dropna()
```

Out[50]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	Sales units	Turn

In [51]:
```
x.columns
```

Out[51]:
```
Index(['MonthYear', 'Time index', 'Country', 'StoreID', 'City', 'Dept_ID',
       'Dept. Name', 'HoursOwn', 'HoursLease', 'Sales units', 'Turnover',
       'Customer', 'Area (m2)', 'Opening hours'],
      dtype='object')
```

In [52]:
```
x.dropna(axis=1,how="any")
```

Out[52]:

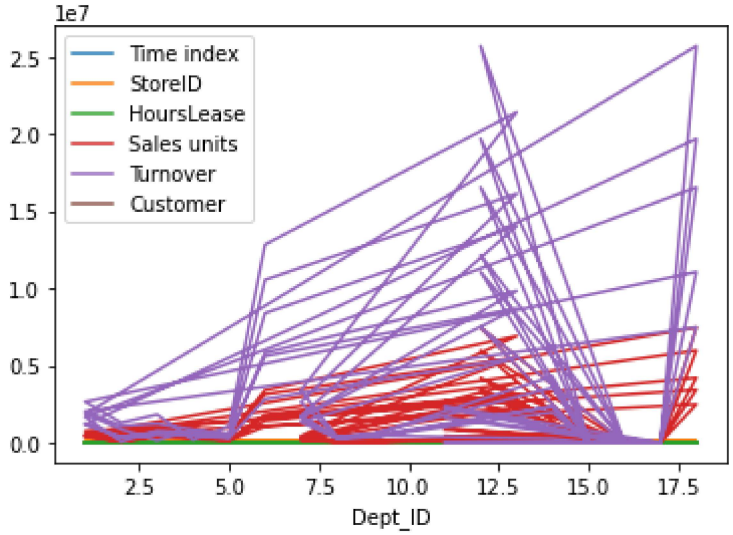
	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease
0	10.2016	1.0	United Kingdom	88253.0	London (I)	1.0	Dry	3184.764	0.0
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	0.0
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	0.0
3	10.2016	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 × 13 columns

In [53]:
```
import matplotlib.pyplot as pp
```

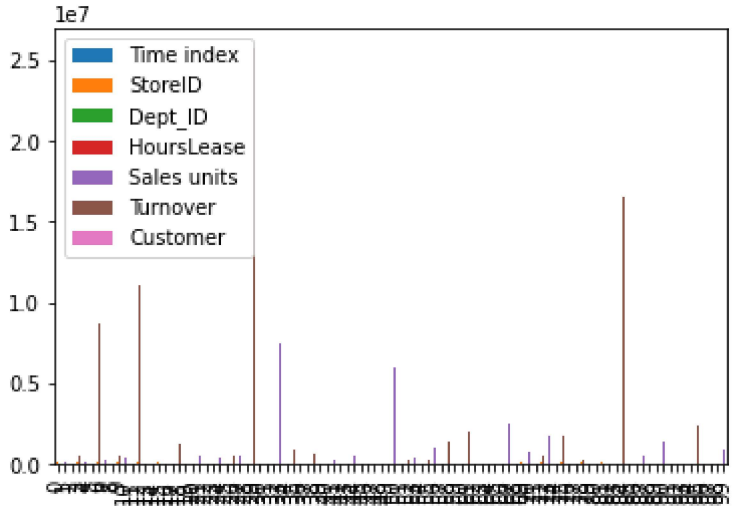
In [55]:
```
x.plot.line("Dept_ID")
```

Out[55]: <AxesSubplot:xlabel='Dept_ID'>



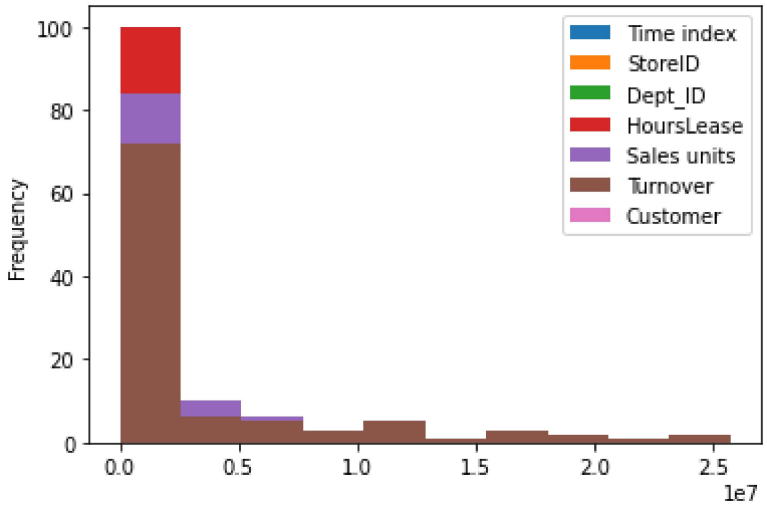
In [56]:
```
x.plot.bar()
```

Out[56]: <AxesSubplot:>



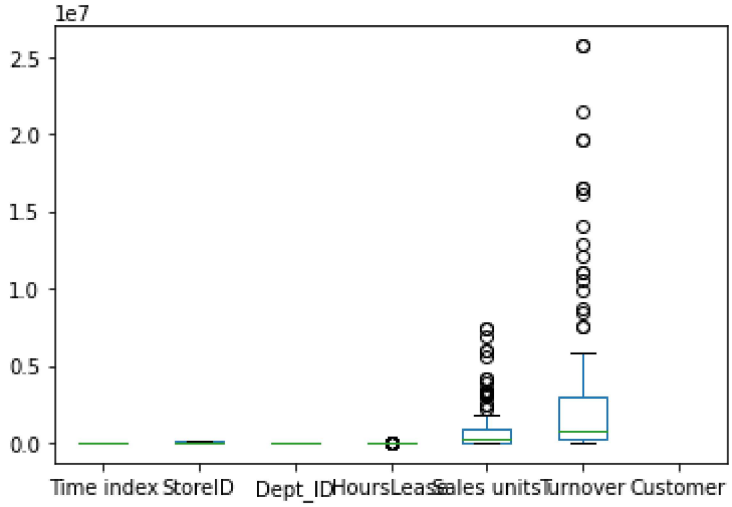
In [57]:
```
x.plot.hist()
```

Out[57]: <AxesSubplot:ylabel='Frequency'>



In [58]:
```
x.plot.box()
```

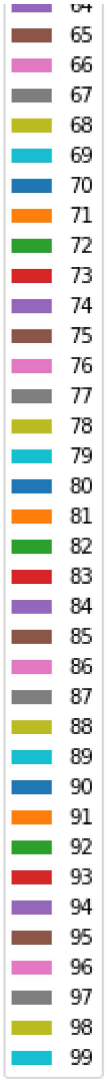
Out[58]: <AxesSubplot:>



In [74]:
```
x.plot.pie(y="StoreID")
```

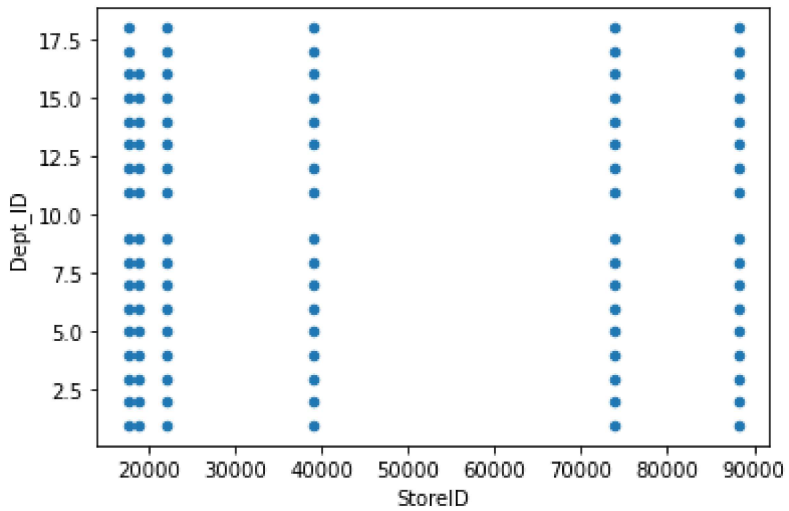
Out[74]:
```
<AxesSubplot:ylabel='StoreID'>
```


StoreID
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63



In [75]:
```
x.plot.scatter(x='StoreID',y='Dept_ID')
```

Out[75]: <AxesSubplot:xlabel='StoreID', ylabel='Dept_ID'>



In [60]:
```
y=x[["Dept_ID"]]
y
```

Out[60]:

	Dept_ID
0	1.0
1	2.0
2	3.0
3	4.0
4	5.0
...	...
95	14.0
96	15.0
97	12.0
98	16.0
99	11.0

100 rows × 1 columns

In [61]:
```
x.mean()
```

Out[61]:
```
Time index           1.00
StoreID          43781.34
Dept_ID              9.31
HoursLease          65.34
Sales units    1063110.45
Turnover       3590810.91
Customer              NaN
dtype: float64
```

In [62]:
```
x.median()
```

Out[62]:
```
MonthYear          10.2016
Time index          1.0000
StoreID         38976.0000
Dept_ID             9.0000
HoursOwn         3252.4245
HoursLease          0.0000
Sales units    307285.0000
Turnover       833925.0000
Customer               NaN
Area (m2)       1320.1200
dtype: float64
```

In [63]:
```
x.mode()
```

Out[63]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease
0	10.2016	1.0	United Kingdom	17647.0	Birmingham	1.0	Admin	47.205	0.(
1	NaN	NaN	NaN	22117.0	Leicester	2.0	Checkout	NaN	NaN
2	NaN	NaN	NaN	38976.0	Liverpool	3.0	Clothing	NaN	NaN
3	NaN	NaN	NaN	73949.0	London (I)	4.0	Customer Services	NaN	NaN
4	NaN	NaN	NaN	88253.0	Manchester	5.0	Delivery	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	6.0	Dry	NaN	NaN
6	NaN	NaN	NaN	NaN	NaN	7.0	Fish	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	8.0	Food	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	9.0	Frozen	NaN	NaN
9	NaN	NaN	NaN	NaN	NaN	11.0	Fruits & Vegetables	NaN	NaN
10	NaN	NaN	NaN	NaN	NaN	12.0	Hardware	NaN	NaN
11	NaN	NaN	NaN	NaN	NaN	13.0	Household	NaN	NaN
12	NaN	NaN	NaN	NaN	NaN	14.0	Meat	NaN	NaN
13	NaN	NaN	NaN	NaN	NaN	15.0	Non Food	NaN	NaN
14	NaN	NaN	NaN	NaN	NaN	16.0	other	NaN	NaN

In [64]:
```
x.sum()
```

Out[64]:
```
MonthYear        10.201610.201610.201610.201610.201610.201610.2...
Time index                                                 100.0
Country          United KingdomUnited KingdomUnited KingdomUnit...
StoreID                                                4378134.0
City             London (I)London (I)London (I)London (I)London...
Dept_ID                                                    931.0
Dept. Name       DryFrozenotherFishFruits & VegetablesMeatFoodC...
HoursOwn         3184.7641582.94147.2051623.8521759.1738270.316...
HoursLease                                                6534.0
Sales units                                           106311045.0
Turnover                                              359081091.0
Customer                                                    0.0
Area (m2)        953.04720.48966.721053.361053.3611735.1619865....
Opening hours    Type AType AType AType AType AType AType AType...
dtype: object
```

In [65]:
```
x.count()
```

Out[65]:
```
MonthYear        100
Time index       100
Country          100
StoreID          100
City             100
Dept_ID          100
Dept. Name       100
HoursOwn         100
HoursLease       100
Sales units      100
Turnover         100
Customer           0
Area (m2)        100
Opening hours    100
dtype: int64
```

In [66]:
```
x.min()
```

Out[66]:
```
MonthYear               10.2016
Time index                  1.0
Country          United Kingdom
StoreID                 17647.0
City                 Birmingham
Dept_ID                     1.0
Dept. Name                Admin
HoursOwn                      0
HoursLease                  0.0
Sales units                 0.0
Turnover                    0.0
Customer                    NaN
Area (m2)                     0
Opening hours            Type A
dtype: object
```

In [67]:
```
x.max()
```

Out[67]:
```
MonthYear               10.2016
Time index                  1.0
Country          United Kingdom
StoreID                 88253.0
City                 Manchester
Dept_ID                    18.0
Dept. Name               others
HoursOwn               9654.996
HoursLease               1896.0
Sales units           7476680.0
Turnover             25719732.0
Customer                    NaN
Area (m2)               9959.04
Opening hours            Type B
dtype: object
```

In [68]:
```
from numpy import cov
from scipy.stats import pearsonr
from scipy.stats import spearmanr
```

In [70]:
```
d1=x["Dept_ID"]
d2=x["StoreID"]
cov(d1,d2)
```

Out[70]:
```
array([[2.80140404e+01, 4.05093394e+03],
       [4.05093394e+03, 7.92225747e+08]])
```

In [71]:
```
print(pearsonr(d1,d2))
```

```
(0.027192103790695, 0.788274475472634)
```

In [72]:
```
print(spearmanr(d1,d2))
```

```
SpearmanrResult(correlation=0.02743319542997965, pvalue=0.7864418629068988)
```

In [ ]: