

# Basic Analysis using Numpy and Pandas

## Sales Dataset

To import library

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

```
In [2]: import pandas as pd
```

To import dataset

```
In [3]: data=pd.read_csv(r"C:\Users\user\Downloads\sales.csv")
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



To get Top 20 record

In [4]: `data.head(20)`

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
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
10	10.2016	1.0	United Kingdom	88253.0	London (I)	14.0	Non Food	7911.558	0.0
11	10.2016	1.0	United Kingdom	88253.0	London (I)	15.0	Admin	4308.243	0.0
12	10.2016	1.0	United Kingdom	88253.0	London (I)	12.0	Checkout	5825.097	0.0
13	10.2016	1.0	United Kingdom	88253.0	London (I)	16.0	Customer Services	3320.085	0.0
14	10.2016	1.0	United Kingdom	88253.0	London (I)	11.0	Delivery	0	0.0
15	10.2016	1.0	United Kingdom	88253.0	London (I)	17.0	others	2253.252	0.0
16	10.2016	1.0	United Kingdom	88253.0	London (I)	18.0	all	40086.486	0.0
17	10.2016	1.0	United Kingdom	38976.0	Manchester	1.0	Dry	2583.687	0.0
18	10.2016	1.0	United Kingdom	38976.0	Manchester	2.0	Frozen	5145.345	0.0
19	10.2016	1.0	United Kingdom	38976.0	Manchester	3.0	other	47.205	0.0

To get last 20 record

In [5]: `data.tail(20)`

Out[5]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLea
7638	06.2017	9.0	Sweden	90992.0	Malmö	11.0	Delivery	0	(
7639	06.2017	9.0	Sweden	90992.0	Malmö	17.0	others	1935.405	(
7640	06.2017	9.0	Sweden	90992.0	Malmö	18.0	all	40133.691	(
7641	06.2017	9.0	Sweden	29650.0	Gothenburg	1.0	Dry	1929.111	(
7642	06.2017	9.0	Sweden	29650.0	Gothenburg	2.0	Frozen	2457.807	(
7643	06.2017	9.0	Sweden	29650.0	Gothenburg	3.0	other	47.205	(
7644	06.2017	9.0	Sweden	29650.0	Gothenburg	4.0	Fish	1689.939	(
7645	06.2017	9.0	Sweden	29650.0	Gothenburg	5.0	Fruits & Vegetables	2567.952	(
7646	06.2017	9.0	Sweden	29650.0	Gothenburg	6.0	Meat	8210.523	(
7647	06.2017	9.0	Sweden	29650.0	Gothenburg	13.0	Food	16902.537	(
7648	06.2017	9.0	Sweden	29650.0	Gothenburg	7.0	Clothing	3587.58	(
7649	06.2017	9.0	Sweden	29650.0	Gothenburg	8.0	Household	1312.299	(
7650	06.2017	9.0	Sweden	29650.0	Gothenburg	9.0	Hardware	1598.676	(
7651	06.2017	9.0	Sweden	29650.0	Gothenburg	14.0	Non Food	6498.555	(
7652	06.2017	9.0	Sweden	29650.0	Gothenburg	15.0	Admin	3433.377	(
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	(

Statistical Analysis

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

Out[6]:

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

To get row and column

In [7]: `print(np.shape(data))`

(7658, 14)

Find Number of Elements

In [8]: `np.size(data)`

Out[8]: 107212

Find Missing values

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



To drop the missing values

In [10]: `data.dropna(axis=1,how="any")`

Out[10]:

	MonthYear
0	10.2016
1	10.2016
2	10.2016
3	10.2016
4	10.2016
...	...
7653	06.2017
7654	06.2017
7655	06.2017
7656	06.2017
7657	06.2017

7658 rows × 1 columns

```
In [11]: data["Time index"]
```

```
Out[11]: 0      1.0
          1      1.0
          2      1.0
          3      1.0
          4      1.0
          ...
          7653    9.0
          7654    9.0
          7655    9.0
          7656    9.0
          7657    9.0
          Name: Time index, Length: 7658, dtype: float64
```

```
In [12]: data1=data[['Time index','Dept_ID']]
          data1
```

```
Out[12]:
```

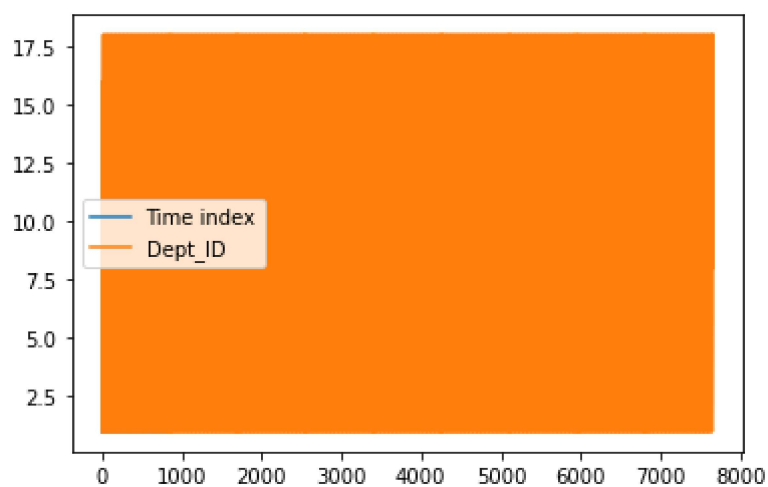
	Time index	Dept_ID
0	1.0	1.0
1	1.0	2.0
2	1.0	3.0
3	1.0	4.0
4	1.0	5.0
...	...	...
7653	9.0	12.0
7654	9.0	16.0
7655	9.0	11.0
7656	9.0	17.0
7657	9.0	18.0

7658 rows × 2 columns

```
In [13]: import matplotlib as mp
```

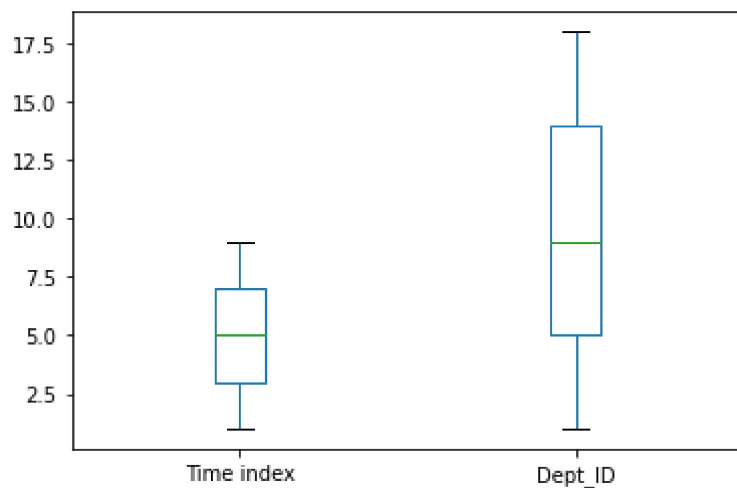
```
In [14]: data1.plot.line()
```

```
Out[14]: <AxesSubplot:>
```



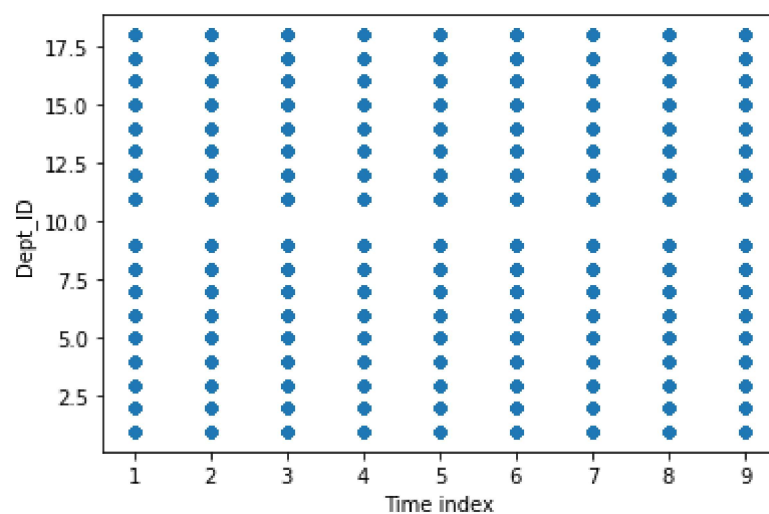
```
In [15]: data1.plot.box()
```

```
Out[15]: <AxesSubplot:>
```



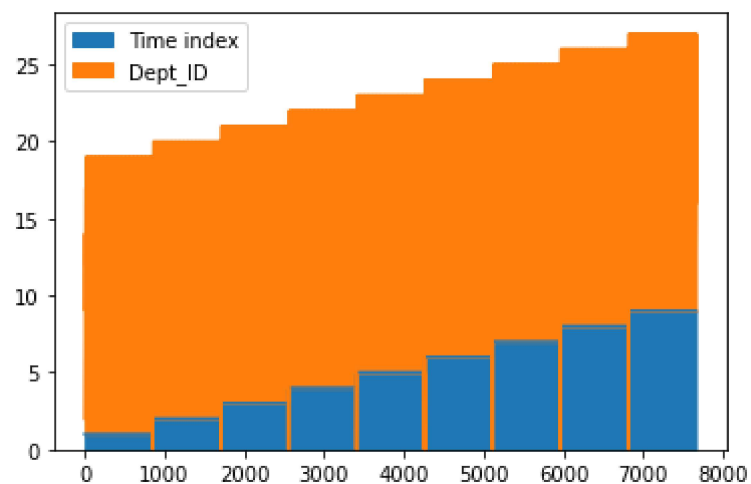
```
In [16]: data1.plot.scatter(x="Time index",y="Dept_ID")
```

```
Out[16]: <AxesSubplot:xlabel='Time index', ylabel='Dept_ID'>
```



```
In [17]: data1.plot.area()
```

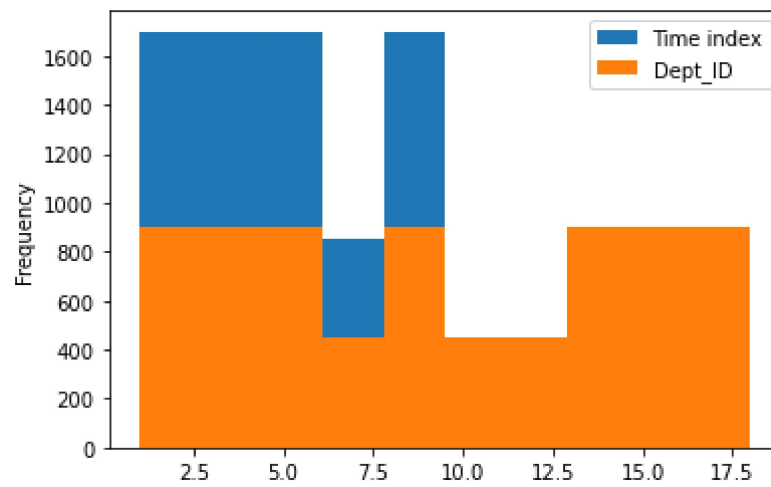
```
Out[17]: <AxesSubplot:>
```





```
In [18]: data1.plot.hist()
```

```
Out[18]: <AxesSubplot:ylabel='Frequency'>
```



```
In [19]: data1.plot.pie(y="Dept_ID")
```

```
Out[19]: <AxesSubplot:ylabel='Dept_ID'>
```

```

-----
ValueError                                Traceback (most recent call last)
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\formatters.py in __call__(self, obj)
    339                 pass
    340             else:
--> 341                 return printer(obj)
    342             # Finally look for special method names
    343             method = get_real_method(obj, self.print_method)

C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\pylabtools.py in <lambda>(fig)
    246
    247     if 'png' in formats:
--> 248         png_formatter.for_type(Figure, lambda fig: print_figure(fig,
'png', **kwargs))
    249     if 'retina' in formats or 'png2x' in formats:
    250         png_formatter.for_type(Figure, lambda fig: retina_figure(fig,
**kwargs))

C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\pylabtools.py in print_figure(fig, fmt, bbox_inches, **kwargs)
    130         FigureCanvasBase(fig)
    131
--> 132     fig.canvas.print_figure(bytes_io, **kw)
    133     data = bytes_io.getvalue()
    134     if fmt == 'svg':

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientation, format, bbox_inches, pad_inches, bbox_extra_artists, backend, **kwargs)
    2208
    2209         try:
-> 2210             result = print_method(
    2211                 filename,
    2212                 dpi=dpi,

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in wrapper(*args, **kwargs)
    1637         kwargs.pop(arg)
    1638
-> 1639     return func(*args, **kwargs)
    1640
    1641     return wrapper

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *args)
    507         *metadata*, including the default 'Software' key.
    508         """
--> 509     FigureCanvasAgg.draw(self)
    510     mpl.image.imsave(
    511         filename_or_obj, self.buffer_rgba(), format="png", origin
="upper",

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in draw(self)
    400     def draw(self):

```

```

401         # docstring inherited
--> 402     self.renderer = self.get_renderer(cleared=True)
403     # Acquire a lock on the shared font cache.
404     with RendererAgg.lock, \

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py
in get_renderer(self, cleared)
416         and getattr(self, "_lastKey", None) == key)
417     if not reuse_renderer:
--> 418         self.renderer = RendererAgg(w, h, self.figure.dpi)
419         self._lastKey = key
420     elif cleared:

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py
in __init__(self, width, height, dpi)
94         self.width = width
95         self.height = height
---> 96     self._renderer = _RendererAgg(int(width), int(height), dpi)
97     self._filter_renderers = []
98

```

**ValueError:** Image size of 259x114892 pixels is too large. It must be less than  $2^{16}$  in each direction.

<Figure size 432x288 with 1 Axes>

In [ ]: