

In [7]:

```
import pandas as pd
df = pd.read_csv('Record.csv')
df
```

Out[7]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	15822
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	3283
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	9339
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	8102	9.33	6.92	75591.66	560
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3296425.02	26573
5	Australia and Oceania	Solomon Islands	Baby Food	Online	C	2/4/2015	547995746	2/21/2015	2974	255.28	159.42	759202.72	4741
6	Sub-Saharan Africa	Angola	Household	Offline	M	4/23/2011	135425221	4/27/2011	4187	668.27	502.54	2798046.49	21041
7	Sub-Saharan Africa	Burkina Faso	Vegetables	Online	H	7/17/2012	871543967	7/27/2012	8082	154.06	90.93	1245112.92	7348
8	Sub-Saharan Africa	Republic of the Congo	Personal Care	Offline	M	7/14/2015	770463311	8/25/2015	6070	81.73	56.67	496101.10	3439
9	Sub-Saharan Africa	Senegal	Cereal	Online	H	4/18/2014	616607081	5/30/2014	6593	205.70	117.11	1356180.10	7721
10	Asia	Kyrgyzstan	Vegetables	Online	H	6/24/2011	814711606	7/12/2011	124	154.06	90.93	19103.44	112
11	Sub-Saharan Africa	Cape Verde	Clothes	Offline	H	8/2/2014	939825713	8/19/2014	4168	109.28	35.84	455479.04	1493
12	Asia	Bangladesh	Clothes	Online	L	1/13/2017	187310731	3/1/2017	8263	109.28	35.84	902980.64	2961
13	Central America and the Caribbean	Honduras	Household	Offline	H	2/8/2017	522840487	2/13/2017	8974	668.27	502.54	5997054.98	45097
14	Asia	Mongolia	Personal Care	Offline	C	2/19/2014	832401311	2/23/2014	4901	81.73	56.67	400558.73	2777
15	Europe	Bulgaria	Clothes	Online	M	4/23/2012	972292029	6/3/2012	1673	109.28	35.84	182825.44	599
16	Asia	Sri Lanka	Cosmetics	Offline	M	11/19/2016	419123971	12/18/2016	6952	437.20	263.33	3039414.40	18306
17	Sub-Saharan Africa	Cameroon	Beverages	Offline	C	4/1/2015	519820964	4/18/2015	5430	47.45	31.79	257653.50	1726
18	Asia	Turkmenistan	Household	Offline	L	12/30/2010	441619336	1/20/2011	3830	668.27	502.54	2559474.10	19247
19	Australia and Oceania	East Timor	Meat	Online	L	7/31/2012	322067916	9/11/2012	5908	421.89	364.69	2492526.12	21545
20	Europe	Norway	Baby Food	Online	L	5/14/2014	819028031	6/28/2014	7450	255.28	159.42	1901836.00	11876
21	Europe	Portugal	Baby Food	Online	H	7/31/2015	860673511	9/3/2015	1273	255.28	159.42	324971.44	2029
22	Central America and the Caribbean	Honduras	Snacks	Online	L	6/30/2016	795490682	7/26/2016	2225	152.58	97.44	339490.50	2168

23	Australia Region Oceania	New Zealand	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Profit
24	Europe	Moldova	Personal Care	Online	L	5/7/2016	740147912	5/10/2016	5070	81.73	56.67	414371.10	2873	
25	Europe	France	Cosmetics	Online	H	5/22/2017	898523128	6/5/2017	1815	437.20	263.33	793518.00	4779	
26	Australia and Oceania	Kiribati	Fruits	Online	M	10/13/2014	347140347	11/10/2014	5398	9.33	6.92	50363.34	373	
27	Sub- Saharan Africa	Mali	Fruits	Online	L	5/7/2010	686048400	5/10/2010	5822	9.33	6.92	54319.26	402	
28	Europe	Norway	Beverages	Offline	C	7/18/2014	435608613	7/30/2014	5124	47.45	31.79	243133.80	1628	
29	Sub- Saharan Africa	The Gambia	Household	Offline	L	5/26/2012	886494815	6/9/2012	2370	668.27	502.54	1583799.90	11910	
...	
70	Asia	Turkmenistan	Office Supplies	Online	M	4/23/2013	462405812	5/20/2013	5010	651.21	524.96	3262562.10	26300	
71	Middle East and North Africa	Libya	Fruits	Online	L	8/14/2015	816200339	9/30/2015	673	9.33	6.92	6279.09	46	
72	Sub- Saharan Africa	Democratic Republic of the Congo	Beverages	Online	C	5/26/2011	585920464	7/15/2011	5741	47.45	31.79	272410.45	1825	
73	Sub- Saharan Africa	Djibouti	Cereal	Online	H	5/20/2017	555990016	6/17/2017	8656	205.70	117.11	1780539.20	10137	
74	Middle East and North Africa	Pakistan	Cosmetics	Offline	L	7/5/2013	231145322	8/16/2013	9892	437.20	263.33	4324782.40	26048	
75	North America	Mexico	Household	Offline	C	11/6/2014	986435210	12/12/2014	6954	668.27	502.54	4647149.58	34946	
76	Australia and Oceania	Federated States of Micronesia	Beverages	Online	C	10/28/2014	217221009	11/15/2014	9379	47.45	31.79	445033.55	2981	
77	Asia	Laos	Vegetables	Offline	C	9/15/2011	789176547	10/23/2011	3732	154.06	90.93	574951.92	3393	
78	Europe	Monaco	Baby Food	Offline	H	5/29/2012	688288152	6/2/2012	8614	255.28	159.42	2198981.92	13732	
79	Australia and Oceania	Samoa	Cosmetics	Online	H	7/20/2013	670854651	8/7/2013	9654	437.20	263.33	4220728.80	25421	
80	Europe	Spain	Household	Offline	L	10/21/2012	213487374	11/30/2012	4513	668.27	502.54	3015902.51	22679	
81	Middle East and North Africa	Lebanon	Clothes	Online	L	9/18/2012	663110148	10/8/2012	7884	109.28	35.84	861563.52	2825	
82	Middle East and North Africa	Iran	Cosmetics	Online	H	11/15/2016	286959302	12/8/2016	6489	437.20	263.33	2836990.80	17087	
83	Sub- Saharan Africa	Zambia	Snacks	Online	L	1/4/2011	122583663	1/5/2011	4085	152.58	97.44	623289.30	3980	
84	Sub- Saharan Africa	Kenya	Vegetables	Online	L	3/18/2012	827844560	4/7/2012	6457	154.06	90.93	994765.42	5871	
85	North America	Mexico	Personal Care	Offline	L	2/17/2012	430915820	3/20/2012	6422	81.73	56.67	524870.06	3639	
86	Sub- Saharan Africa	Sao Tome and Principe	Beverages	Offline	C	1/16/2011	180283772	1/21/2011	8829	47.45	31.79	418936.05	2806	
87	Sub- Saharan Africa	The Gambia	Baby Food	Offline	M	2/3/2014	494747245	3/20/2014	5559	255.28	159.42	1419101.52	8862	
88	Middle East and North Africa	Kuwait	Fruits	Online	M	4/30/2012	513417565	5/18/2012	522	9.33	6.92	4870.26	36	
89	Europe	Slovenia	Beverages	Offline	C	10/23/2016	245740560	11/05/2016	4000	47.45	31.79	204447.00	1404	

89	Europe	Slovenia	Beverages	Offline	C	10/23/2016	345718562	11/25/2016	4660	47.45	31.79	221117.00	1481
	Region Sub-	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total
90	Saharan Africa	Sierra Leone	Office Supplies	Offline	H	12/6/2016	621386563	12/14/2016	948	651.21	524.96	617347.08	4976
91	Australia and Oceania	Australia	Beverages	Offline	H	7/7/2014	240470397	7/11/2014	9389	47.45	31.79	445508.05	2984
92	Middle East and North Africa	Azerbaijan	Office Supplies	Online	M	6/13/2012	423331391	7/24/2012	2021	651.21	524.96	1316095.41	10609
93	Europe	Romania	Cosmetics	Online	H	11/26/2010	660643374	12/25/2010	7910	437.20	263.33	3458252.00	20829
94	Central America and the Caribbean	Nicaragua	Beverages	Offline	C	2/8/2011	963392674	3/21/2011	8156	47.45	31.79	387002.20	2592
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	888	109.28	35.84	97040.64	318
96	Asia	Malaysia	Fruits	Offline	L	11/11/2011	810711038	12/28/2011	6267	9.33	6.92	58471.11	433
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	6/1/2016	728815257	6/29/2016	1485	154.06	90.93	228779.10	1350
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	8/8/2015	5767	81.73	56.67	471336.91	3268
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2/10/2012	665095412	2/15/2012	5367	668.27	502.54	3586605.09	26971

100 rows × 14 columns

We have read the data from "Record.csv" file

In [10]:

```
df.to_csv('new.csv')#it will load data with index value
#if we dont' want index
df.to_csv('new_noIndex.csv',index = False)
```

We have created the weather_data with features day,temperature,windspeed and event

In [4]:

```
import pandas as pd
weather_data = [('1/1/2017',32,6,'Rain'),
                ('1/2/2017',35,7,'Sunny'),
                ('1/3/2017',28,2,'Snow'),
                ('1/4/2017',24,2,'Snow'),
                ('1/5/2017',32,4,'Rain'),
                ('1/6/2017',31,2,'Sunny')]
df = pd.DataFrame(weather_data,columns=['day','temperature','windspeed','event'])
df
```

Out[4]:

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow
3	1/4/2017	24	2	Snow
4	1/5/2017	32	4	Rain
5	1/6/2017	31	2	Sunny

In [5]:

```
df.shape
```

Out[5]:

(6, 4)

There are sixth rows and 4th columns

In [8]:

```
df.head(3)
```

Out[8]:

	day	temprature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow

We have fetched the 3 top data from "weather_data" dataframe

In [10]:

```
df.tail()
```

Out[10]:

	day	temprature	windspeed	event
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow
3	1/4/2017	24	2	Snow
4	1/5/2017	32	4	Rain
5	1/6/2017	31	2	Sunny

We got last 5th rows by default as we don't mention any number of rows

In [11]:

```
df[2:5]
```

Out[11]:

	day	temprature	windspeed	event
2	1/3/2017	28	2	Snow
3	1/4/2017	24	2	Snow
4	1/5/2017	32	4	Rain

We have feteched the data from row 2 to row 4 based on silce

In [13]:

```
df.columns
```

Out[13]:

```
Index(['day', 'temprature', 'windspeed', 'event'], dtype='object')
```

We got all columns using df.columns

In [14]:

```
df.day
```

Out[14]:

```
0    1/1/2017
1    1/2/2017
2    1/3/2017
3    1/4/2017
4    1/5/2017
5    1/6/2017
Name: day, dtype: object
```

In [15]:

```
df['day']
```

Out[15]:

```
0    1/1/2017
1    1/2/2017
2    1/3/2017
3    1/4/2017
4    1/5/2017
5    1/6/2017
Name: day, dtype: object
```

In [19]:

```
df['temperature'].max()
```

Out[19]:

```
35
```

We used max() to get maximum temprature

In [20]:

```
df.temperature.min()
```

Out[20]:

```
24
```

min() to get minimum temprature

In [22]:

```
df[['day', 'event']]
```

Out[22]:

	day	event
0	1/1/2017	Rain
1	1/2/2017	Sunny
2	1/3/2017	Snow
3	1/4/2017	Snow
4	1/5/2017	Rain
5	1/6/2017	Sunny

In [23]:

```
df['temperature'].describe()
```

Out[23]:

```
count      6.000000
mean       30.333333
std        3.829708
min        24.000000
25%        28.750000
50%        31.500000
75%        32.000000
max        35.000000
Name: temprature, dtype: float64
```

Using describe function,we get statistics details

In [25]:

```
df[df.temprature == df.temprature.max()]
```

Out[25]:

	day	temprature	windspeed	event
1	1/2/2017	35	7	Sunny

In [26]:

```
df.day[df.temprature == df.temprature.max()]
```

Out[26]:

```
1      1/2/2017
Name: day, dtype: object
```

We found the date on which having the maximum temprature

In [12]:

```
import pandas as pd
df= pd.read_excel('Book1.xlsx')
```

Numpy Basic codes

We have created the 3 dimensional arrat

In [19]:

```
import numpy as np
a = np.array([[[0,1],[2,3]],[[4,5],[6,7]]])
a
```

Out[19]:

```
array([[[0, 1],
        [2, 3]],
       [[4, 5],
        [6, 7]]])
```

In [17]:

```
a.ndim
```

Out[17]:

ndim is used to get dimensionality of array

In [18]:

```
a.shape
```

Out [18]:

 $(2, 2, 2)$

We found the shape of array

In [6]:

```
l = [1,2,3,45]
%timeit [i**2 for i in l]
```

1.12 μ s \pm 61 ns per loop (mean \pm std. dev. of 7 runs, 1000000 loops each)

In [8]:

```
import numpy as np
a = np.array(1000)
%timeit a**2
```

488 ns \pm 58.1 ns per loop (mean \pm std. dev. of 7 runs, 1000000 loops each)

In [4]:

```
import numpy as np
a = np.arange(1,10,3)
a
```

Out[4]:

```
array([1, 4, 7])
```

`arange(start,end,step)` is used to arrange the data

In [7]:

```
import numpy as np

a = np.linspace(0,1,6)
print(a)
```

[0. 0.2 0.4 0.6 0.8 1.]

Using the linspace, we will get value between 0 to 1 and six is showing the number of values

In [27]:

```
a = np.ones((10,10))
a
```

Out[27]:

[illegible]

```
[1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],  
[1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]])
```

numpy.ones fill all values as 1

In [29]:

```
a = np.zeros((4,4))  
a
```

Out[29]:

```
array([[0., 0., 0., 0.],  
       [0., 0., 0., 0.],  
       [0., 0., 0., 0.],  
       [0., 0., 0., 0.]])
```

In [33]:

```
a = np.eye(7)  
a
```

Out[33]:

```
array([[1., 0., 0., 0., 0., 0., 0.],  
       [0., 1., 0., 0., 0., 0., 0.],  
       [0., 0., 1., 0., 0., 0., 0.],  
       [0., 0., 0., 1., 0., 0., 0.],  
       [0., 0., 0., 0., 1., 0., 0.],  
       [0., 0., 0., 0., 0., 1., 0.],  
       [0., 0., 0., 0., 0., 0., 1.]])
```

In [34]:

```
a = np.eye(3,3)  
a
```

Out[34]:

```
array([[1., 0., 0.],  
       [0., 1., 0.],  
       [0., 0., 1.]])
```

In [36]:

```
a = np.diag([1,2,3,4,5])  
a
```

Out[36]:

```
array([[1, 0, 0, 0, 0],  
       [0, 2, 0, 0, 0],  
       [0, 0, 3, 0, 0],  
       [0, 0, 0, 4, 0],  
       [0, 0, 0, 0, 5]])
```

In [38]:

```
a = np.random.rand(4)  
a
```

Out[38]:

```
array([0.86033351, 0.45154492, 0.8654467 , 0.04231998])
```

got positive random value

In [39]:


```
a = np.random.randn(4)
a
```

Out[39]:

```
array([-0.5568228 , -0.31951956, -0.18035875, -1.71528343])
```

We have got negative random value

In [42]:

```
a = np.arange(10)
a.dtype
```

Out[42]:

```
dtype('int32')
```

In [44]:

```
a = np.arange(10, dtype = 'float64')
a
```

Out[44]:

```
array([0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])
```

In [48]:

```
import numpy as np
a = np.arange(10)
print(a[5])
```

5

In [49]:

```
a = np.arange(10)
a
```

Out[49]:

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [52]:

```
b= a[1:8:2]
b
```

Out[52]:

```
array([1, 3, 5, 7])
```

In [53]:

```
a = np.arange(10)
a[5:] = 10
a
```

Out[53]:

```
array([ 0,  1,  2,  3,  4, 10, 10, 10, 10, 10])
```

We have set value 10 after position 5th

In [56]:

```
b = np.arange(5)
a[5:] = b[::-1]
a
```

Out[56]:

```
array([0, 1, 2, 3, 4, 4, 3, 2, 1, 0])
```

We have reversed the values after position 5th

In [72]:

```
import numpy as np
a = np.arange(10)
a
b = a[::2]
print(b)
print(np.shares_memory(a,b))
```

```
[0 2 4 6 8]
True
```

In [73]:

```
a[0] = 10
print(a)
print(b)
```

```
[10  1  2  3  4  5  6  7  8  9]
[10  2  4  6  8]
```

In [65]:

```
a = np.arange(10)
b = a[::2].copy() #force a copy
print(b)
print(np.shares_memory(a,b))
```

```
[0 2 4 6 8]
False
```

In [74]:

```
a[0] = 10
print(b)
print(a)
```

```
[10  2  4  6  8]
[10  1  2  3  4  5  6  7  8  9]
```

In [75]:

```
import numpy as np

a = np.random.randint(0,20,15)
a
```

Out[75]:

```
array([19,  1, 12, 17,  8,  8,  0, 16, 14,  6, 12, 12,  0, 14, 12])
```

In [76]:

```
mask = (a%2==0)
extract_from_a = a[mask]
extract from a
```

Out[76]:

```
array([12,  8,  8,  0, 16, 14,  6, 12, 12,  0, 14, 12])
```

In [78]:

```
a[mask] = -1
a
```

Out[78]:

```
array([19,  1, -1, 17, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1])
```

In [79]:

```
a = np.arange(0,100,10)
a
```

Out[79]:

```
array([ 0, 10, 20, 30, 40, 50, 60, 70, 80, 90])
```

In [81]:

```
a[[9,7]] = -200
a
```

Out[81]:

```
array([  0,  10,  20,  30,  40,  50,  60, -200,  80, -200])
```

In [85]:

```
import numpy as np
a = np.array([1,2,3,2])
b = np.array([2,2,3,2])
c = np.array([6,4,4,5])
d = (a<=b)&(a<=c)
print(d.all())
```

True

In [90]:

```
import numpy as np
x = np.array([1,2,3,1])
y = np.array([[1,2,3],[5,6,7]])
print(x.mean())
print(np.median(x))
print(np.median(y,axis=-1))
print(x.std())
```

1.75

1.5

[2. 6.]

0.82915619758885

In [112]:

```
import numpy as np
a= np.tile(np.arange(0,40,10),(3,1))
a = a.T
b = np.array([0,1,2])
a+b
```

Out[112]:

```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```

In [100]:

```
a= a.T
a
```

Out[100]:

```
array([[ 0,  0,  0],
       [10, 10, 10],
       [20, 20, 20],
       [30, 30, 30]])
```

In [95]:

```
b= np.array([0,1,2])
a+b
```

Out[95]:

```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```

In [111]:

```
import numpy as np
a= np.arange(0,40,10)
a = a[:,np.newaxis]
b= np.array([0,1,2])
a+b
```

Out[111]:

```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```

In [121]:

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
print(a.ravel())
```

[1 2 3 4 5 6]

In [122]:

```
a=a.T
a
```

Out[122]:

```
array([[1, 4],
       [2, 5],
       [3, 6]])
```

In [123]:

```
a=a.ravel()
a
```

Out[123]:

```
array([1, 4, 2, 5, 3, 6])
```

```
In [124]:
```

```
print(a.shape)
```

```
(6,)
```

```
In [126]:
```

```
a = a.reshape((2,3))  
a
```

```
Out[126]:
```

```
array([[1, 4, 2],  
       [5, 3, 6]])
```

```
In [ ]:
```

```
a[0,0]= 100  
a
```

```
In [129]:
```

```
a = np.array([[5,4,6],[2,3,2]])  
a.sort(axis=1)  
a
```

```
Out[129]:
```

```
array([[4, 5, 6],  
       [2, 2, 3]])
```

```
In [128]:
```

```
a = np.array([[5,4,6],[2,3,2]])  
b = np.sort(a,axis=0)  
b
```

```
Out[128]:
```

```
array([[2, 3, 2],  
       [5, 4, 6]])
```

```
In [130]:
```

```
a = np.array([4,3,1,2])  
j = np.argsort(a)  
j
```

```
Out[130]:
```

```
array([2, 3, 1, 0], dtype=int64)
```

```
In [131]:
```

```
a[j]
```

```
Out[131]:
```

```
array([1, 2, 3, 4])
```

```
In [ ]:
```

