

THE ASSAM KAZIRANGA UNIVERSITY

# Data Analytics Using Python Laboratory

**NAME:JINTU GOGOI  
ID: ET21BTHCS059  
B. Tech , CSE,  
6th semester, Sec-A  
DAP Lab Copy**

# INDEX

Experiments	Signatures
1. Weather Data Analysis	
2. Car Data Analysis	
3. Police Data Analysis	
4. Covid Data Analysis	
5. London Housing Data Analysis	
6. Census Data Analysis	
7. Udemy Course Data Analysis	
8. Netflix Data Analysis	

## Weather Data

```
In [1]: import pandas as pd #importing Library files
```

```
In [3]: data=pd.read_csv('Weather.csv') #reading data
```

```
In [48]: data.head(1) #displaying only head of the data i.e only first row
```

```
Out[48]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog

```
In [41]: #Q. 1) Find all the unique 'Wind Speed' values in the data.
```

```
data['Wind Speed_km/h'].unique()
```

```
Out[41]: array([ 4,  7,  6,  9, 15, 13, 20, 22, 19, 24, 30, 35, 39, 32, 33, 26, 44,  
        43, 48, 37, 28, 17, 11,  0, 83, 70, 57, 46, 41, 52, 50, 63, 54,  2],  
       dtype=int64)
```

```
In [54]: #Q. 2) Find the number of times when the 'Weather is exactly Clear'.
```

```
data[data['Weather']=='Clear']
```

```
Out[54]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
67	1/3/2012 19:00	-16.9	-24.8	50	24	25.0	101.74	Clear
114	1/5/2012 18:00	-7.1	-14.4	56	11	25.0	100.71	Clear
115	1/5/2012 19:00	-9.2	-15.4	61	7	25.0	100.80	Clear
116	1/5/2012 20:00	-9.8	-15.7	62	9	25.0	100.83	Clear
117	1/5/2012 21:00	-9.0	-14.8	63	13	25.0	100.83	Clear
...	...	...	...	...	...	...	...	...
8646	12/26/2012 6:00	-13.4	-14.8	89	4	25.0	102.47	Clear
8698	12/28/2012 10:00	-6.1	-8.6	82	19	24.1	101.27	Clear
8713	12/29/2012 1:00	-11.9	-13.6	87	11	25.0	101.31	Clear
8714	12/29/2012 2:00	-11.8	-13.1	90	13	25.0	101.33	Clear
8756	12/30/2012 20:00	-13.8	-16.5	80	24	25.0	101.52	Clear

1326 rows × 8 columns

```
In [56]: #Q. 3) Find the number of times when the 'Wind Speed was exactly 4 km/h'.  
data[data['Wind Speed_km/h']==4]
```

```
Out[56]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
96	1/5/2012 0:00	-8.8	-11.7	79	4	9.7	100.32	Snow
101	1/5/2012 5:00	-7.0	-9.5	82	4	4.0	100.19	Snow
146	1/7/2012 2:00	-8.1	-11.1	79	4	19.3	100.15	Cloudy
...	...	...	...	...	...	...	...	...
8768	12/31/2012 8:00	-8.6	-10.3	87	4	3.2	101.14	Snow Showers
8769	12/31/2012 9:00	-8.1	-9.6	89	4	2.4	101.09	Snow
8770	12/31/2012 10:00	-7.4	-8.9	89	4	6.4	101.05	Snow,Fog
8772	12/31/2012 12:00	-5.8	-7.5	88	4	12.9	100.78	Snow
8773	12/31/2012 13:00	-4.6	-6.6	86	4	12.9	100.63	Snow

474 rows × 8 columns

```
In [58]: #Q. 4) Find out all the Null Values in the data.  
data.isnull().sum()
```

```
Out[58]:
```

Date/Time	Temp_C
0	0

```
Dew Point Temp_C    0  
Rel Hum_%         0  
Wind Speed_km/h   0  
Visibility_km      0  
Press_kPa          0  
Weather            0  
dtype: int64
```

```
In [60]: #Rename the column name 'Weather' of the dataframe to 'Weather Condition'.
```

```
data.rename(columns={'Weather':'Weather Condition'}, inplace=True)  
data
```

```
Out[60]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
2	1/1/2012 2:00	-1.8	-3.4	89	7	4.0	101.26	Freezing Drizzle,Fog
3	1/1/2012 3:00	-1.5	-3.2	88	6	4.0	101.27	Freezing Drizzle,Fog
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fog
...	...	...	...	...	...	...	...	...
8779	12/31/2012 19:00	0.1	-2.7	81	30	9.7	100.13	Snow
8780	12/31/2012 20:00	0.2	-2.4	83	24	9.7	100.03	Snow
8781	12/31/2012 21:00	-0.5	-1.5	93	28	4.8	99.95	Snow
8782	12/31/2012 22:00	-0.2	-1.8	89	28	9.7	99.91	Snow
8783	12/31/2012 23:00	0.0	-2.1	86	30	11.3	99.89	Snow

8784 rows × 8 columns

```
In [64]: #Q. 6) What is the mean 'Visibility' ?
```

```
data['Visibility_km'].mean()
```

```
Out[64]: 27.664446721311478
```

```
In [65]: #Q. 7) What is the Standard Deviation of 'Pressure' in this data?
```

```
data['Press_kPa'].std() #std() is the function use to find standard deviation
```

```
Out[65]: 0.8440047459486474
```

```
In [67]: #Q. 8) What is the Variance of 'Relative Humidity' in this data ?
```

```
data['Rel Hum_%'].var() #var is the function use to find the variance
```

```
Out[67]: 286.2485501984998
```

```
In [68]: #Q. 9) Find all instances when 'Snow' was recorded.
```

```
data[data['Weather Condition']=='Snow']
```

```
Out[68]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
55	1/3/2012 7:00	-14.0	-19.5	63	19	25.0	100.95	Snow
84	1/4/2012 12:00	-13.7	-21.7	51	11	24.1	101.25	Snow
86	1/4/2012 14:00	-11.3	-19.0	53	7	19.3	100.97	Snow
87	1/4/2012 15:00	-10.2	-16.3	61	11	9.7	100.89	Snow
88	1/4/2012 16:00	-9.4	-15.5	61	13	19.3	100.79	Snow
...	...	...	...	...	...	...	...	...
8779	12/31/2012 19:00	0.1	-2.7	81	30	9.7	100.13	Snow
8780	12/31/2012 20:00	0.2	-2.4	83	24	9.7	100.03	Snow
8781	12/31/2012 21:00	-0.5	-1.5	93	28	4.8	99.95	Snow
8782	12/31/2012 22:00	-0.2	-1.8	89	28	9.7	99.91	Snow
8783	12/31/2012 23:00	0.0	-2.1	86	30	11.3	99.89	Snow

390 rows × 8 columns

```
In [69]: #Q. 10) Find all instances when 'Wind Speed is above 24' and 'Visibility is 25'.
```

```
data[(data['Wind Speed_km/h']>24) & (data['Visibility_km']==25)]
```

```
data[(data['Wind Speed_km/h']>24) & (data['Visibility_km']==25)]
```

**Out[69]:**

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
23	1/1/2012 23:00	5.3	2.0	79	30	25.0	99.31	Cloudy
24	1/2/2012 0:00	5.2	1.5	77	35	25.0	99.26	Rain Showers
25	1/2/2012 1:00	4.6	0.0	72	39	25.0	99.26	Cloudy
26	1/2/2012 2:00	3.9	-0.9	71	32	25.0	99.26	Mostly Cloudy
27	1/2/2012 3:00	3.7	-1.5	69	33	25.0	99.30	Mostly Cloudy
...	...	...	...	...	...	...	...	...
8705	12/28/2012 17:00	-8.6	-12.0	76	26	25.0	101.34	Mainly Clear
8753	12/30/2012 17:00	-12.1	-15.8	74	28	25.0	101.26	Mainly Clear
8755	12/30/2012 19:00	-13.4	-16.5	77	26	25.0	101.47	Mainly Clear
8759	12/30/2012 23:00	-12.1	-15.1	78	28	25.0	101.52	Mostly Cloudy
8760	12/31/2012 0:00	-11.1	-14.4	77	26	25.0	101.51	Cloudy

308 rows × 8 columns

**In [70]:** #Q. 11) What is the Mean value of each column against each 'Weather Condition' ?

```
data.groupby(data['Weather Condition']).mean()
```

```
C:\Users\Dibyajyoti\AppData\Local\Temp\ipykernel_18632\3119569782.py:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
```

```
data.groupby(data['Weather Condition']).mean()
```

**Out[70]:**

Weather Condition	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Clear	6.825716	0.089367	64.497738	10.557315	30.153243	101.587443
Cloudy	7.970544	2.375810	69.592593	16.127315	26.625752	100.911441
Drizzle	7.353659	5.504878	88.243902	16.097561	17.931707	100.435366
Drizzle,Fog	8.067500	7.033750	93.275000	11.862500	5.257500	100.786625
Drizzle,Ice Pellets,Fog	0.400000	-0.700000	92.000000	20.000000	4.000000	100.790000
Drizzle,Snow	1.050000	0.150000	93.500000	14.000000	10.500000	100.890000
Drizzle,Snow,Fog	0.693333	0.120000	95.866667	15.533333	5.513333	99.281333
Fog	4.303333	3.159333	92.286667	7.946667	6.248000	101.184067
Freezing Drizzle	-5.667143	-8.000000	83.571429	16.571429	9.200000	100.202857
Freezing Drizzle,Fog	-2.533333	-4.183333	88.500000	17.000000	5.266667	100.441667
Freezing Drizzle,Haze	-5.433333	-8.000000	82.000000	10.333333	2.666667	100.316667
Freezing Drizzle,Snow	-5.109091	-7.072727	86.090909	16.272727	5.872727	100.520909
Freezing Fog	-7.575000	-9.250000	87.750000	4.750000	0.650000	102.320000
Freezing Rain	-3.885714	-6.078571	84.642857	19.214286	8.242857	99.647143
Freezing Rain,Fog	-2.225000	-3.750000	89.500000	15.500000	7.550000	99.945000
Freezing Rain,Haze	-4.900000	-7.450000	82.500000	7.500000	2.400000	100.375000
Freezing Rain,Ice Pellets,Fog	-2.600000	-3.700000	92.000000	28.000000	8.000000	100.950000
Freezing Rain,Snow Grains	-5.000000	-7.300000	84.000000	32.000000	4.800000	98.560000
Haze	-0.200000	-2.975000	81.625000	10.437500	7.831250	101.482500
Mainly Clear	12.558927	4.581671	60.667142	14.144824	34.264862	101.248832
Moderate Rain,Fog	1.700000	0.800000	94.000000	17.000000	6.400000	99.980000
Moderate Snow	-5.525000	-7.250000	87.750000	33.750000	0.750000	100.275000
Moderate Snow,Blowing Snow	-5.450000	-6.500000	92.500000	40.000000	0.600000	100.570000
Mostly Cloudy	10.574287	3.131174	62.102465	15.813920	31.253842	101.025288
Rain	9.786275	7.042810	83.624183	19.254902	18.856536	100.233333
Rain Showers	13.722340	9.187766	75.159574	17.132979	22.816489	100.404043
Rain Showers,Fog	12.800000	12.100000	96.000000	13.000000	6.400000	99.830000
Rain Showers,Snow Showers	2.150000	-1.500000	76.500000	22.500000	21.700000	101.100000
Rain,Fog	8.273276	7.219828	93.189655	14.793103	6.873276	100.500862
Rain,Haze	4.633333	2.066667	83.333333	11.666667	6.700000	100.540000
Rain,Ice Pellets	0.600000	-0.600000	92.000000	24.000000	9.700000	100.120000

Rain,Snow	1.055556	-0.566667	89.000000	28.388889	11.672222	99.951111
Rain,Snow,Grains	1.900000	-2.100000	75.000000	26.000000	25.000000	100.600000
Rain,Snow,Fog	0.800000	0.300000	96.000000	9.000000	6.400000	100.730000
Rain,Snow,Ice Pellets	1.100000	-0.175000	91.500000	23.250000	6.000000	100.105000
Snow	-4.524103	-7.623333	79.307692	20.038462	11.171795	100.536103
Snow Pellets	0.700000	-6.400000	59.000000	35.000000	2.400000	99.700000
Snow Showers	-3.506667	-7.866667	72.350000	19.233333	20.158333	100.963500
Snow Showers,Fog	-10.675000	-11.900000	90.750000	13.750000	7.025000	101.292500
Snow,Blowing Snow	-5.410526	-7.621053	84.473684	34.842105	4.105263	99.704737
Snow,Fog	-5.075676	-6.364865	90.675676	17.324324	4.537838	100.688649
Snow,Haze	-4.020000	-6.860000	80.600000	5.000000	4.640000	100.782000
Snow,Ice Pellets	-1.883333	-3.666667	87.666667	23.833333	7.416667	100.548333
Thunderstorms	24.150000	19.750000	77.000000	7.500000	24.550000	100.230000
Thunderstorms,Heavy Rain Showers	10.900000	9.000000	88.000000	9.000000	2.400000	100.260000
Thunderstorms,Moderate Rain Showers,Fog	19.600000	18.500000	93.000000	15.000000	3.200000	100.010000
Thunderstorms,Rain	20.433333	18.533333	89.000000	15.666667	19.833333	100.420000
Thunderstorms,Rain Showers	20.037500	17.618750	86.375000	18.312500	15.893750	100.233750
Thunderstorms,Rain Showers,Fog	21.600000	18.700000	84.000000	19.666667	9.700000	100.063333
Thunderstorms,Rain,Fog	20.600000	18.600000	88.000000	19.000000	4.800000	100.080000

In [71]: #Q. 11) What is the Mean value of each column against each 'Weather Condition' ?

#Minimum value of each column of weather condition

data.groupby(data['Weather Condition']).min()

Out[71]:

Weather Condition	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Clear	1/11/2012 1:00	-23.3	-28.5	20	0	11.3	99.52
Cloudy	1/1/2012 17:00	-21.4	-26.8	18	0	11.3	98.39
Cloudy	1/1/2012 17:00	-21.4	-26.8	18	0	11.3	98.39
Drizzle	1/23/2012 21:00	1.1	-0.2	74	0	6.4	97.84
Drizzle,Fog	1/23/2012 20:00	0.0	-1.6	85	0	1.0	98.65
Drizzle,Ice Pellets,Fog	12/17/2012 9:00	0.4	-0.7	92	20	4.0	100.79
Drizzle,Snow	12/17/2012 15:00	0.9	0.1	92	9	9.7	100.63
Drizzle,Snow,Fog	12/18/2012 21:00	0.3	-0.1	92	7	2.4	97.79
Fog	1/1/2012 0:00	-16.0	-17.2	80	0	0.2	98.31
Freezing Drizzle	1/13/2012 10:00	-9.0	-12.2	78	6	4.8	98.44
Freezing Drizzle,Fog	1/1/2012 2:00	-6.4	-9.0	82	6	3.6	98.74
Freezing Drizzle,Haze	2/1/2012 11:00	-5.8	-8.3	81	9	2.0	100.28
Freezing Drizzle,Snow	1/13/2012 3:00	-8.3	-10.4	79	6	2.4	99.19
Freezing Fog	1/22/2012 6:00	-19.0	-22.9	71	0	0.2	101.97
Freezing Rain	1/13/2012 11:00	-6.5	-9.0	81	7	2.8	98.22
Freezing Rain,Fog	1/17/2012 23:00	-6.1	-8.7	82	7	2.8	98.32
Freezing Rain,Haze	2/1/2012 14:00	-4.9	-7.5	82	6	2.0	100.34
Freezing Rain,Ice Pellets,Fog	12/17/2012 3:00	-2.6	-3.7	92	28	8.0	100.95
Freezing Rain,Snow Grains	1/13/2012 9:00	-5.0	-7.3	84	32	4.8	98.56
Haze	1/22/2012 12:00	-11.5	-16.0	68	0	4.8	100.35
Mainly Clear	1/10/2012 11:00	-22.8	-28.0	20	0	12.9	98.67
Moderate Rain,Fog	12/10/2012 8:00	1.7	0.8	94	17	6.4	99.98
Moderate Snow	1/12/2012 15:00	-6.3	-7.6	83	26	0.6	99.88
Moderate Snow,Blowing Snow	12/27/2012 10:00	-5.5	-6.6	92	39	0.6	100.50
Mostly Cloudy	1/1/2012 16:00	-23.2	-28.5	18	0	11.3	98.36
Rain	1/1/2012 18:00	0.3	-5.7	40	0	4.0	97.52
Rain Showers	1/1/2012 22:00	1.6	-7.2	37	0	6.4	98.51
Rain Showers,Fog	10/20/2012 3:00	12.8	12.1	96	13	6.4	99.83
Rain Showers,Snow Showers	11/4/2012 8:00	2.1	-1.8	75	17	19.3	101.09
Rain,Fog	1/23/2012 18:00	0.0	-1.2	83	0	2.0	98.61

	Rain,Fog	1/23/2012 18:00	0.0	-1.2	83	0	2.0	98.61
	Rain,Haze	3/13/2012 7:00	4.0	1.0	81	7	4.0	100.50
	Rain,Ice Pellets	12/18/2012 5:00	0.6	-0.6	92	24	9.7	100.12
	Rain,Snow	1/10/2012 5:00	0.6	-1.7	81	13	2.4	98.18
	Rain,Snow Grains	12/21/2012 0:00	1.9	-2.1	75	26	25.0	100.60
	Rain,Snow,Fog	12/8/2012 21:00	0.8	0.3	96	9	6.4	100.73
	Rain,Snow,Ice Pellets	12/21/2012 1:00	0.9	-0.7	88	17	4.8	99.85
	Snow	1/10/2012 1:00	-16.7	-24.6	41	0	1.0	97.75
	Snow Pellets	11/24/2012 15:00	0.7	-6.4	59	35	2.4	99.70
	Snow Showers	1/12/2012 7:00	-13.3	-19.3	52	0	2.4	99.49
	Snow Showers,Fog	12/26/2012 9:00	-11.3	-12.7	89	7	4.0	100.63
	Snow,Blowing Snow	1/13/2012 21:00	-12.0	-16.2	70	24	0.6	98.11
	Snow,Fog	12/16/2012 15:00	-10.1	-12.0	77	4	1.2	99.38
	Snow,Haze	2/1/2012 17:00	-4.3	-7.2	80	0	4.0	100.61
	Snow,Ice Pellets	12/10/2012 3:00	-4.3	-5.9	76	19	2.8	99.40
	Thunderstorms	7/16/2012 1:00	21.6	19.4	67	0	24.1	99.84
	Thunderstorms,Heavy Rain Showers	5/29/2012 6:00	10.9	9.0	88	9	2.4	100.26
	Thunderstorms,Moderate Rain Showers,Fog	7/17/2012 6:00	19.6	18.5	93	15	3.2	100.01
	Thunderstorms,Rain	5/25/2012 20:00	19.4	18.2	83	4	16.1	100.19
	Thunderstorms,Rain Showers	5/29/2012 16:00	11.0	7.0	68	7	6.4	99.65
	Thunderstorms,Rain Showers,Fog	6/29/2012 3:00	19.5	16.1	80	7	9.7	99.71
	Thunderstorms,Rain,Fog	7/17/2012 5:00	20.6	18.6	88	19	4.8	100.08

In [72]: #Maximum value of each column of weather condition

```
data.groupby(data['Weather Condition']).max()
```

Out[72]:

Weather Condition	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Clear	9/9/2012 5:00	32.8	20.4	99	33	48.3	103.63
Cloudy	9/9/2012 23:00	30.5	22.6	99	54	48.3	103.65
Drizzle	9/30/2012 3:00	18.8	17.7	96	30	25.0	101.56
Drizzle,Fog	9/30/2012 2:00	19.9	19.1	100	28	9.7	102.07
Drizzle,Ice Pellets,Fog	12/17/2012 9:00	0.4	-0.7	92	20	4.0	100.79
Drizzle,Snow	12/19/2012 18:00	1.2	0.2	95	19	11.3	101.15
Drizzle,Snow,Fog	12/22/2012 3:00	1.1	0.6	98	32	9.7	100.15
Fog	9/22/2012 0:00	20.8	19.6	100	22	9.7	103.04
Freezing Drizzle	2/1/2012 5:00	-2.3	-3.3	93	26	12.9	101.02
Freezing Drizzle,Fog	12/10/2012 5:00	-0.3	-2.3	94	33	8.0	101.27
Freezing Drizzle,Haze	2/1/2012 13:00	-5.0	-7.7	83	11	4.0	100.36
Freezing Drizzle,Snow	3/2/2012 12:00	-3.3	-4.6	94	24	12.9	101.18
Freezing Fog	3/17/2012 6:00	-0.1	-0.3	99	9	0.8	102.85
Freezing Rain	2/1/2012 7:00	0.3	-1.7	92	28	16.1	101.00
Freezing Rain,Fog	12/17/2012 1:00	0.1	-0.9	93	26	9.7	101.01
Freezing Rain,Haze	2/1/2012 15:00	-4.9	-7.4	83	9	2.8	100.41
Freezing Rain,Ice Pellets,Fog	12/17/2012 3:00	-2.6	-3.7	92	28	8.0	100.95
Freezing Rain,Snow Grains	1/13/2012 9:00	-5.0	-7.3	84	32	4.8	98.56
Haze	3/13/2012 23:00	14.1	11.1	86	17	9.7	102.97
Mainly Clear	9/9/2012 9:00	33.0	21.2	99	63	48.3	103.59
Moderate Rain,Fog	12/10/2012 8:00	1.7	0.8	94	17	6.4	99.98
Moderate Snow	12/27/2012 9:00	-4.9	-6.7	93	39	0.8	100.67
Moderate Snow,Blowing Snow	12/27/2012 12:00	-5.4	-6.4	93	41	0.6	100.64
Mostly Cloudy	9/9/2012 2:00	32.4	24.4	100	83	48.3	103.65
Rain	9/5/2012 2:00	22.8	20.4	99	52	48.3	102.26
Rain Showers	9/8/2012 16:00	26.4	23.0	97	41	48.3	102.31
Rain Showers,Fog	10/20/2012 3:00	12.8	12.1	96	13	6.4	99.83

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
Rain Showers,Fog	10/20/2012 0:00	12.0	12.1	86	10	0.7	100.00	
Rain Showers,Snow Showers	12/5/2012 10:00	2.2	-1.2	78	28	24.1	101.11	
Rain,Fog	9/30/2012 23:00	21.7	19.5	100	46	9.7	101.77	
Rain,Haze	3/13/2012 9:00	5.5	2.9	86	17	9.7	100.61	
Rain,Ice Pellets	12/18/2012 5:00	0.6	-0.6	92	24	9.7	100.12	
Rain,Snow	4/23/2012 3:00	1.7	0.5	94	52	25.0	101.07	
Rain,Snow,Grains	12/21/2012 0:00	1.9	-2.1	75	26	25.0	100.60	
Rain,Snow,Fog	12/8/2012 21:00	0.8	0.3	96	9	6.4	100.73	
Rain,Snow,Ice Pellets	12/21/2012 5:00	1.3	0.1	94	28	6.4	100.47	
Snow	4/27/2012 9:00	3.7	0.3	96	57	25.0	102.73	
Snow Pellets	11/24/2012 15:00	0.7	-6.4	59	35	2.4	99.70	
Snow Showers	3/4/2012 21:00	2.9	-0.7	94	37	48.3	102.50	
Snow Showers,Fog	12/29/2012 13:00	-10.0	-11.1	92	22	9.7	102.52	
Snow,Blowing Snow	2/25/2012 9:00	-1.4	-2.9	91	48	9.7	100.62	
Snow,Fog	3/14/2012 19:00	1.1	0.8	99	35	9.7	102.07	
Snow,Haze	2/1/2012 21:00	-3.6	-6.4	81	15	6.4	100.99	
Snow,Ice Pellets	3/3/2012 4:00	0.8	-1.7	92	33	11.3	100.96	
Thunderstorms	7/4/2012 16:00	26.7	20.1	87	15	25.0	100.62	
Thunderstorms,Heavy Rain Showers	5/29/2012 6:00	10.9	9.0	88	9	2.4	100.26	
Thunderstorms,Moderate Rain Showers,Fog	7/17/2012 6:00	19.6	18.5	93	15	3.2	100.01	
Thunderstorms,Rain	7/23/2012 18:00	21.3	19.1	93	30	24.1	100.83	
Thunderstorms,Rain Showers	9/8/2012 4:00	25.5	23.1	98	32	25.0	101.06	
Thunderstorms,Rain Showers,Fog	7/31/2012 20:00	22.9	21.3	91	35	9.7	100.64	
Thunderstorms,Rain,Fog	7/17/2012 5:00	20.6	18.6	88	19	4.8	100.08	

In [73]: #Q. 13) Show all the Records where Weather Condition is Fog.

```
data[data['Weather Condition']=='Fog']
```

Out[73]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fog
5	1/1/2012 5:00	-1.4	-3.3	87	9	6.4	101.27	Fog
6	1/1/2012 6:00	-1.5	-3.1	89	7	6.4	101.29	Fog
...	...	...	...	...	...	...	...	...
8716	12/29/2012 4:00	-16.0	-17.2	90	6	9.7	101.25	Fog
8717	12/29/2012 5:00	-14.8	-15.9	91	4	6.4	101.25	Fog
8718	12/29/2012 6:00	-13.8	-15.3	88	4	9.7	101.25	Fog
8719	12/29/2012 7:00	-14.8	-16.4	88	7	8.0	101.22	Fog
8722	12/29/2012 10:00	-12.0	-13.3	90	7	6.4	101.15	Fog

150 rows × 8 columns

In [74]: #Q. 14) Find all instances when 'Weather is Clear' or 'Visibility is above 40'

```
data[(data['Weather Condition']=='Clear') | (data['Visibility_km']>40)] # / is the symbol for OR
```

Out[74]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
67	1/3/2012 19:00	-16.9	-24.8	50	24	25.0	101.74	Clear
106	1/5/2012 10:00	-6.0	-10.0	73	17	48.3	100.45	Mainly Clear
107	1/5/2012 11:00	-5.6	-10.2	70	22	48.3	100.41	Mainly Clear
108	1/5/2012 12:00	-4.7	-9.6	69	20	48.3	100.38	Mainly Clear
109	1/5/2012 13:00	-4.4	-9.7	66	26	48.3	100.40	Mainly Clear
...	...	...	...	...	...	...	...	...
8749	12/30/2012 13:00	-12.4	-16.2	73	37	48.3	100.92	Mostly Cloudy
8750	12/30/2012 14:00	-11.8	-16.1	70	37	48.3	100.96	Mainly Clear
8751	12/30/2012 15:00	-11.3	-15.6	70	32	48.3	101.05	Mainly Clear
8752	12/30/2012 16:00	-11.4	-15.5	72	26	48.3	101.15	Mainly Clear

```
8752 12/30/2012 10:00 -11.4 -15.5 72 20 48.3 101.10 Mainly Clear  
8756 12/30/2012 20:00 -13.8 -16.5 80 24 25.0 101.52 Clear
```

3027 rows × 8 columns

```
In [76]: #Q. 15) Find all instances when :  
# A. 'Weather is Clear' and 'Relative Humidity is greater than 50' or B. 'Visibility is above 40'  
data[((data['Weather Condition']=='Clear') & (data['Rel Hum_%']>50)) | (data['Visibility_km']>40)]
```

```
Out[76]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
106	1/5/2012 10:00	-6.0	-10.0	73	17	48.3	100.45	Mainly Clear
107	1/5/2012 11:00	-5.6	-10.2	70	22	48.3	100.41	Mainly Clear
108	1/5/2012 12:00	-4.7	-9.6	69	20	48.3	100.38	Mainly Clear
109	1/5/2012 13:00	-4.4	-9.7	66	26	48.3	100.40	Mainly Clear
110	1/5/2012 14:00	-5.1	-10.7	65	22	48.3	100.46	Mainly Clear
...	...	...	...	...	...	...	...	...
8749	12/30/2012 13:00	-12.4	-16.2	73	37	48.3	100.92	Mostly Cloudy
8750	12/30/2012 14:00	-11.8	-16.1	70	37	48.3	100.96	Mainly Clear
8751	12/30/2012 15:00	-11.3	-15.6	70	32	48.3	101.05	Mainly Clear
8752	12/30/2012 16:00	-11.4	-15.5	72	26	48.3	101.15	Mainly Clear
8756	12/30/2012 20:00	-13.8	-16.5	80	24	25.0	101.52	Clear

2921 rows × 8 columns

```
In [ ]:
```

```
In [ ]: # Cars Dataset
# Q. 1) Instruction ( For Data Cleaning ) - Find all Null Values in the dataset. If there is any null value in any column, then find it.
# Q. 2) Question ( Based on Value Counts )- Check what are the different types of Make are there in our dataset. And, what is the count of each make?
# Q. 3) Instruction ( Filtering ) - Show all the records where Origin is Asia or Europe.
# Q. 4) Instruction ( Removing unwanted records ) - Remove all the records (rows) where Weight is above 4000.
# Q. 5) Instruction ( Applying function on a column ) - Increase all the values of 'MPG_City' column by 3.
```

```
In [2]: import pandas as pd
df = pd.read_csv(r"cars.csv");
print(df);
```

```
      Make          Model  Type  Origin DriveTrain    MSRP \
0   Acura           MDX   SUV   Asia     All $36,945
1   Acura        RSX Type S 2dr Sedan  Asia   Front $23,820
2   Acura          TSX 4dr Sedan  Asia   Front $26,990
3   Acura          TL 4dr Sedan  Asia   Front $33,195
4   Acura         3.5 RL 4dr Sedan  Asia   Front $43,755
..   ...
427  Volvo       C70 LPT convertible 2dr Sedan  Europe   Front $40,565
428  Volvo       C70 HPT convertible 2dr Sedan  Europe   Front $42,565
429  Volvo          S80 Tc 4dr Sedan  Europe   Front $45,210
430  Volvo            V40 Wagon  Europe   Front $26,135
431  Volvo          XC70 Wagon  Europe     All $35,145

      Invoice  EngineSize  Cylinders  Horsepower  MPG_City  MPG_Highway \
0    $33,337        3.5         6.0       265.0      17.0        23.0
1    $21,761        2.0         4.0       200.0      24.0        31.0
2    $24,647        2.4         4.0       200.0      22.0        29.0
3    $30,299        3.2         6.0       270.0      20.0        28.0
4    $39,014        3.5         6.0       225.0      18.0        24.0
..   ...
427   $38,203        2.4         5.0       197.0      21.0        28.0
428   $40,083        2.3         5.0       242.0      20.0        26.0
429   $42,573        2.9         6.0       268.0      19.0        26.0
430   $24,641        1.9         4.0       170.0      22.0        29.0
431   $33,112        2.5         5.0       208.0      20.0        27.0
```

```
      Weight  Wheelbase  Length
0    4451.0     106.0    189.0
1    2778.0     101.0    172.0
2    3230.0     105.0    183.0
3    3575.0     108.0    186.0
4    3880.0     115.0    197.0
..   ...
427   3450.0     105.0    186.0
428   3450.0     105.0    186.0
429   3653.0     110.0    190.0
430   2822.0     101.0    180.0
431   3823.0     109.0    186.0
```

[432 rows x 15 columns]

```
In [12]: df.isnull().sum()
```

```
# print(my_df['Age'].mean())
```

```
Out[12]: Make      4
Model      4
Type       4
Origin     4
DriveTrain 4
MSRP       4
Invoice    4
EngineSize 4
Cylinders  6
Horsepower 4
MPG_City   4
MPG_Highway 4
Weight     4
Wheelbase  4
Length     4
dtype: int64
```

```
In [44]: df['Cylinders'].fillna(df['Cylinders'].mean(), inplace=True)
```

```
In [35]: df.fillna(df.mean(), inplace=True)

# df['Cylinders'].fillna(df['Cylinders'].mean(), inplace=True)

df.isnull().sum()

C:\Users\Dibyajyoti\AppData\Local\Temp\ipykernel_20992\1995966899.py:1: FutureWarning: The default value of numeric_only in Dataframe.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
df.fillna(df.mean(), inplace=True)
```

```
Out[35]: Make      4
Model      4
Type       4
Origin     4
DriveTrain 4
MSRP       4
Invoice    4
EngineSize 0
Cylinders  0
Horsepower 0
MPG_City   0
MPG_Highway 0
Weight     0
Wheelbase  0
Length     0
dtype: int64
```

```
In [43]: df['Make'].value_counts()
```

```
Out[43]: Toyota      28
Chevrolet   27
Mercedes-Benz 26
Ford        23
BMW         20
Audi        19
Honda       17
...
Nissan      17
Volkswagen  15
Chrysler    15
Dodge        13
Mitsubishi  13
Volvo        12
Jaguar       12
Hyundai     12
Subaru      11
Pontiac     11
Mazda        11
Lexus        11
Kia          11
Buick        9
Mercury      9
Lincoln     9
Saturn       8
Cadillac    8
Suzuki      8
Infiniti    8
GMC          8
Acura        7
Porsche      7
Saab          7
Land Rover   3
Oldsmobile   3
Jeep          3
Scion         2
Isuzu         2
MINI          2
Hummer        1
Name: Make, dtype: int64
```

```
In [50]: df[df['Origin'].isin(["Asia", "Europe"])]
```

```
Out[50]:   Make   Model   Type   Origin   DriveTrain   MSRP   Invoice   EngineSize   Cylinders   Horsepower   MPG_City   MPG_Highway   Weight   Wheelbase   Length
0   Acura   MDX   SUV   Asia     All   $36,945   $33,337      3.5       6.0      265.0      17.0       23.0     4451.0      106.0      189.0
1   Acura   RSX Type S 2dr   Sedan   Asia   Front   $23,820   $21,761      2.0       4.0      200.0      24.0       31.0     2778.0      101.0      172.0
2   Acura   TSX 4dr   Sedan   Asia   Front   $26,990   $24,647      2.4       4.0      200.0      22.0       29.0     3230.0      105.0      183.0
```

2	Acura	TSX 4dr	Sedan	Asia	Front	\$26,990	\$24,647	2.4	4.0	200.0	22.0	29.0	3230.0	105.0	183.0
3	Acura	TL 4dr	Sedan	Asia	Front	\$33,195	\$30,299	3.2	6.0	270.0	20.0	28.0	3575.0	108.0	186.0
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	\$43,755	\$39,014	3.5	6.0	225.0	18.0	24.0	3880.0	115.0	197.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
427	Volvo	C70 LPT 2dr	Sedan	Europe	Front	\$40,565	\$38,203	2.4	5.0	197.0	21.0	28.0	3450.0	105.0	186.0
428	Volvo	C70 HPT 2dr	Sedan	Europe	Front	\$42,565	\$40,083	2.3	5.0	242.0	20.0	26.0	3450.0	105.0	186.0
429	Volvo	S80 T6 4dr	Sedan	Europe	Front	\$45,210	\$42,573	2.9	6.0	268.0	19.0	26.0	3653.0	110.0	190.0
430	Volvo	V40	Wagon	Europe	Front	\$26,135	\$24,641	1.9	4.0	170.0	22.0	29.0	2822.0	101.0	180.0
431	Volvo	XC70	Wagon	Europe	All	\$35,145	\$33,112	2.5	5.0	208.0	20.0	27.0	3823.0	109.0	186.0

281 rows × 15 columns

4

In [53]: `df[~(df['Weight'] > 4000)]`

Out[53]:	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	\$23,820	\$21,761	2.0	4.0	200.0	24.0	31.0	2778.0	101.0	172.
2	Acura	TSX 4dr	Sedan	Asia	Front	\$26,990	\$24,647	2.4	4.0	200.0	22.0	29.0	3230.0	105.0	183.
3	Acura	TL 4dr	Sedan	Asia	Front	\$33,195	\$30,299	3.2	6.0	270.0	20.0	28.0	3575.0	108.0	186.
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	\$43,755	\$39,014	3.5	6.0	225.0	18.0	24.0	3880.0	115.0	197.
5	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	\$46,100	\$41,100	3.5	6.0	225.0	18.0	24.0	3893.0	115.0	197.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
427	Volvo	C70 LPT 2dr	Sedan	Europe	Front	\$40,565	\$38,203	2.4	5.0	197.0	21.0	28.0	3450.0	105.0	186.
428	Volvo	C70 HPT convertible	Sedan	Europe	Front	\$42,565	\$40,083	2.3	5.0	242.0	20.0	26.0	3450.0	105.0	186.

329 rows × 15 columns

4

In [60]: `df['MPG_City'] = df['MPG_City'].apply(lambda x:x+3)`

In [61]: `df`

Out[61]:	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	MDX	SUV	Asia	All	\$36,945	\$33,337	3.5	6.0	265.0	38.0	23.0	4451.0	106.0	189.0
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	\$23,820	\$21,761	2.0	4.0	200.0	45.0	31.0	2778.0	101.0	172.0
2	Acura	TSX 4dr	Sedan	Asia	Front	\$26,990	\$24,647	2.4	4.0	200.0	43.0	29.0	3230.0	105.0	183.0
3	Acura	TL 4dr	Sedan	Asia	Front	\$33,195	\$30,299	3.2	6.0	270.0	41.0	28.0	3575.0	108.0	186.0
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	\$43,755	\$39,014	3.5	6.0	225.0	39.0	24.0	3880.0	115.0	197.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
427	Volvo	C70 LPT 2dr	Sedan	Europe	Front	\$40,565	\$38,203	2.4	5.0	197.0	42.0	28.0	3450.0	105.0	186.0
428	Volvo	C70 HPT convertible	Sedan	Europe	Front	\$42,565	\$40,083	2.3	5.0	242.0	41.0	26.0	3450.0	105.0	186.0
429	Volvo	S80 T6 4dr	Sedan	Europe	Front	\$45,210	\$42,573	2.9	6.0	268.0	40.0	26.0	3653.0	110.0	190.0
430	Volvo	V40	Wagon	Europe	Front	\$26,135	\$24,641	1.9	4.0	170.0	43.0	29.0	2822.0	101.0	180.0
431	Volvo	XC70	Wagon	Europe	All	\$35,145	\$33,112	2.5	5.0	208.0	41.0	27.0	3823.0	109.0	186.0

432 rows × 15 columns

4

In [ ]:

### Police Data

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

In [2]: df = pd.read_csv(r'PoliceData.csv');

In [3]: #showing the first N rows in the data (by default, N=5)
df.head(5)

Out[3]:   stop_date stop_time country_name driver_gender driver_age_raw driver_age driver_race violation_raw violation search_conducted search_type stop_outcome
0 1/2/2005    1:55      NaN          M       1985.0     20.0    White    Speeding    Speeding        False      NaN
1 1/18/2005   8:15      NaN          M       1965.0     40.0    White    Speeding    Speeding        False      NaN
2 1/23/2005   23:15     NaN          M       1972.0     33.0    White    Speeding    Speeding        False      NaN
3 2/20/2005   17:15     NaN          M       1986.0     19.0    White Call for Service      Other        False      NaN      Arrest Dr
4 3/14/2005   10:00     NaN          F       1984.0     21.0    White    Speeding    Speeding        False      NaN      Cita

In [4]: #detecting the missing values from each column of the dataframe.
df.isnull().sum()

Out[4]: stop_date      0
stop_time      0
country_name  65535
driver_gender  4061
driver_age_raw 4054
driver_age     4387
driver_race    4060
violation_raw  4060
violation      4060
search_conducted 0
search_type    63056
stop_outcome   4060
stop_outcome   4060
is_arrested    4060
stop_duration  4060
drugs_related_stop 0
dtype: int64

In [5]: #To drop a column from dataframe.
df.drop(['country_name'], axis = 1)

Out[5]:   stop_date stop_time driver_gender driver_age_raw driver_age driver_race violation_raw violation search_conducted search_type stop_outcome
0 1/2/2005    1:55          M       1985.0     20.0    White    Speeding    Speeding        False      NaN      Cita
1 1/18/2005   8:15          M       1965.0     40.0    White    Speeding    Speeding        False      NaN      Cita
2 1/23/2005   23:15         M       1972.0     33.0    White    Speeding    Speeding        False      NaN      Cita
3 2/20/2005   17:15         M       1986.0     19.0    White Call for Service      Other        False      NaN      Arrest Dr
4 3/14/2005   10:00         F       1984.0     21.0    White    Speeding    Speeding        False      NaN      Cita
... ... ... ... ... ... ... ... ... ... ... ...
65530 12/6/2012  17:54         F       1987.0     25.0    White    Speeding    Speeding        False      NaN      Cita
65531 12/6/2012  22:22         M       1954.0     58.0    White    Speeding    Speeding        False      NaN      Warr
65532 12/6/2012  23:20         M       1985.0     27.0    Black Equipment/Inspection Violation Equipment        False      NaN      Cita
65533 12/7/2012  0:23        NaN      NaN      NaN      NaN      NaN      NaN        False      NaN      NaN
65534 12/7/2012  0:30         F       1985.0     27.0    White    Speeding    Speeding        False      NaN      Cita

65535 rows × 14 columns

In [6]: #In a column, show all the unique values with their count.
df['driver_age'].value_counts()

Out[6]: 22.0    2912
21.0    2886
20.0    2783
23.0    2725
24.0    2582
```

```
15.0      5
84.0      3
88.0      2
83.0      2
85.0      1
Name: driver_age, Length: 73, dtype: int64
```

```
In [7]: #df.groupby('Col_1')[['Col_2']].sum() - To create groups - Two Keys - Apply on Col_2 grouped by Col_1.
```

```
df.groupby('driver_gender').search_conducted.sum()
```

```
Out[7]: driver_gender
```

```
F    366
M   2113
Name: search_conducted, dtype: int64
```

```
In [8]: df.stop_duration.value_counts()
```

```
Out[8]: 0-15 Min    47379
16-30 Min    11448
30+ Min     2647
2            1
Name: stop_duration, dtype: int64
```

```
In [9]: #df['Column_name'].map( { old1:new1 , old2:new2} ) - Change the all values of a column from old to new. We have to write for all
```

```
df['stop_duration']= df['stop_duration'].map({'0-15 Min' : 7.5, '16-30 Min' : 24, '30+ Min' : 45})
```

```
In [10]: df
```

```
Out[10]:
```

	stop_date	stop_time	country_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conducted	search_t
0	1/2/2005	1:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	False	N
1	1/18/2005	8:15	NaN	M	1985.0	40.0	White	Speeding	Speeding	False	N
2	1/23/2005	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	False	N
65530	12/6/2012	17:54	NaN	F	1987.0	25.0	White	Speeding	Speeding	False	N
65531	12/6/2012	22:22	NaN	M	1954.0	58.0	White	Speeding	Speeding	False	N
65532	12/6/2012	23:20	NaN	M	1985.0	27.0	Black	Equipment/Inspection Violation	Equipment	False	N
65533	12/7/2012	0:23	NaN	NaN	NaN	NaN	NaN	NaN	NaN	False	N
65534	12/7/2012	0:30	NaN	F	1985.0	27.0	White	Speeding	Speeding	False	N

65535 rows × 15 columns

```
In [11]: #df['Column_name'].mean() - To show Mean value of a column.
```

```
df['stop_duration'].mean()
```

```
Out[11]: 12.187420698181345
```

```
In [12]: #df.groupby('Column_1').Column_2.describe() - To create groups based on Column1 and show statistics summary based on Column2.
```

```
df.groupby('stop_date').driver_gender.describe()
```

```
Out[12]:
```

stop_date	count	unique	top	freq
1/1/2006	17	2	M	14
1/1/2007	12	1	M	12
1/1/2008	21	2	F	11
1/1/2009	6	2	M	5
1/1/2010	12	2	M	11
...	...	...	...	...
9/9/2008	14	2	M	9
9/9/2009	17	2	M	16
9/9/2010	28	2	M	21
9/9/2011	24	2	M	19
...	...	...	...	...

## COVID ANALYSIS

```
In [1]: import pandas as pd  
  
import seaborn as sns  
  
import matplotlib.pyplot as plt
```

```
In [2]: df = pd.read_csv('COVID.csv')  
  
df
```

```
Out[2]:
```

	Date	State	Region	Confirmed	Deaths	Recovered
0	4/29/2020	NaN	Afghanistan	1939	60	252
1	4/29/2020	NaN	Albania	766	30	455
2	4/29/2020	NaN	Algeria	3848	444	1702
3	4/29/2020	NaN	Andorra	743	42	423
4	4/29/2020	NaN	Angola	27	2	7
...	...	...	...	...	...	...
316	4/29/2020	Wyoming	US	545	7	0
317	4/29/2020	Xinjiang	Mainland China	76	3	73
318	4/29/2020	Yukon	Canada	11	0	0
319	4/29/2020	Yunnan	Mainland China	185	2	181
320	4/29/2020	Zhejiang	Mainland China	1268	1	1263

321 rows × 6 columns

```
In [3]: #sns.heatmap(df.isnull()) - It will show the all columns & missing values in them in heat map form.  
#plt.show() - To show the plot.  
  
sns.heatmap(df.isnull())  
  
plt.show()
```



```
In [4]: #Q. 1) Show the number of Confirmed, Deaths and Recovered cases in each Region.  
  
df.groupby('Region')[['Confirmed', 'Recovered']].sum()
```

Out[4]:

Region	Confirmed	Recovered
Afghanistan	1939	252
Albania	766	455
Algeria	3848	1702
Andorra	743	423
Angola	27	7
...	...	...
West Bank and Gaza	344	71
Western Sahara	6	5
Yemen	6	1
Zambia	97	54
Zimbabwe	32	5

187 rows × 2 columns

In [5]: #Q. 2) Remove all the records where the Confirmed Cases is Less Than 10.

```
df = df[~(df.Confirmed < 10)]  
df
```

Out[5]:

	Date	State	Region	Confirmed	Deaths	Recovered
0	4/29/2020	NaN	Afghanistan	1939	60	252
1	4/29/2020	NaN	Albania	766	30	455
2	4/29/2020	NaN	Algeria	3848	444	1702
3	4/29/2020	NaN	Andorra	743	42	423
4	4/29/2020	NaN	Angola	27	2	7
...	...	...	...	...	...	...
316	4/29/2020	WuyiJing,...	US	545	7	0
317	4/29/2020	Xinjiang	Mainland China	76	3	73
318	4/29/2020	Yukon	Canada	11	0	0
319	4/29/2020	Yunnan	Mainland China	185	2	181
320	4/29/2020	Zhejiang	Mainland China	1268	1	1263

304 rows × 6 columns

In [6]: #Q. 3) In which Region, maximum number of Confirmed cases were recorded ?

```
df.groupby('Region').Confirmed.sum().sort_values(ascending = False).head(1)
```

Out[6]:

```
Region  
US    1039909  
Name: Confirmed, dtype: int64
```

In [7]: #Q. 4) In which Region, minimum number of Deaths cases were recorded ?

```
df.groupby('Region').Deaths.sum().sort_values(ascending = True).head(1)
```

Out[7]:

```
Region  
Cambodia    0  
Name: Deaths, dtype: int64
```

In [8]: #Q. 5) How many Confirmed, Deaths & Recovered cases were reported from India till 29 April 2020 ?

```
df[df.Region == 'India']
```

Out[8]:

	Date	State	Region	Confirmed	Deaths	Recovered
74	4/29/2020	NaN	India	33062	1079	8437

In [9]: #Q. 6-A ) Sort the entire data wrt No. of Confirmed cases in ascending order.

```
df.sort_values(by = 'Confirmed', ascending = True)
```

Out[9]:

	Date	State	Region	Confirmed	Deaths	Recovered
156	4/29/2020	NaN	Suriname	10	1	8

	Date	Country	Subname	No.	I	O
70	4/29/2020	Nan	Holy See	10	0	2
59	4/29/2020	Nan	Gambia	10	1	8
318	4/29/2020	Yukon	Canada	11	0	0
217	4/29/2020	Greenland	Denmark	11	0	11
...	...	...	...	...	...	...
57	4/29/2020	Nan	France	165093	24087	48228
168	4/29/2020	Nan	UK	165221	26097	0
80	4/29/2020	Nan	Italy	203591	27682	71252
153	4/29/2020	Nan	Spain	236899	24275	132929
265	4/29/2020	New York	US	299691	29477	0

304 rows × 6 columns

In [10]: #Q. Q-10 ) sort the entire data wrt No. of Recovered cases in descending order

df.sort\_values(by=['Recovered'], ascending=False)

Out[10]:

	Date	State	Region	Confirmed	Deaths	Recovered
153	4/29/2020	Nan	Spain	236899	24275	132929
61	4/29/2020	Nan	Germany	161539	6467	120400
76	4/29/2020	Nan	Iran	93657	5957	73791
80	4/29/2020	Nan	Italy	203591	27682	71252
229	4/29/2020	Hubel	Mainland China	68128	4512	63616
...	...	...	...	...	...	...
258	4/29/2020	Nevada	US	4934	230	0
257	4/29/2020	Nebraska	US	3851	56	0
255	4/29/2020	Montana	US	451	16	0
254	4/29/2020	Missouri	US	7660	338	0
274	4/29/2020	Ohio	US	17303	937	0

## LONDON HOUSING ANALYSIS

```
In [1]: import pandas as pd  
  
import seaborn as sns  
  
import matplotlib.pyplot as plt
```

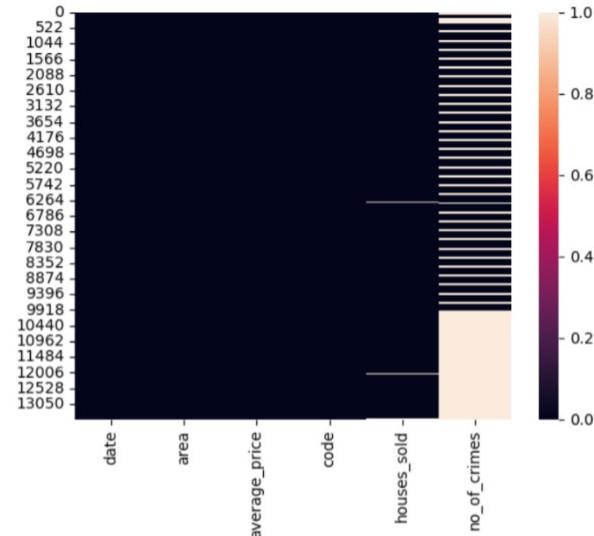
```
In [2]: df = pd.read_csv(r'London_Housing.csv')  
  
df
```

```
Out[2]:
```

	date	area	average_price	code	houses_sold	no_of_crimes
0	1/1/1995	city of london	91449	E09000001	17.0	NaN
1	2/1/1995	city of london	82203	E09000001	7.0	NaN
2	3/1/1995	city of london	79121	E09000001	14.0	NaN
3	4/1/1995	city of london	77101	E09000001	7.0	NaN
4	5/1/1995	city of london	84409	E09000001	10.0	NaN
...	...	...	...	...	...	...
13544	9/1/2019	england	249942	E92000001	64605.0	NaN
13545	10/1/2019	england	249376	E92000001	68677.0	NaN
13546	11/1/2019	england	248515	E92000001	67814.0	NaN
13547	12/1/2019	england	250410	E92000001	NaN	NaN
13548	1/1/2020	england	247355	E92000001	NaN	NaN

13549 rows × 6 columns

```
In [3]: #* sns.heatmap(df.isnull()) - It will show the all columns & missing values in them in heat map form.  
#* plt.show() - To show the plot.  
  
sns.heatmap(df.isnull())  
  
plt.show()
```



**Q. 1) Convert the Datatype of 'Date' column to Date-Time format.**

```
In [4]: df.dtypes
```

```
Out[4]: date      object  
area       object  
.....
```

```
dtype: object

In [5]: df['date'] = pd.to_datetime(df.date)

In [6]: df.dtypes
```

```
Out[6]: date      datetime64[ns]
area        object
average_price    int64
code         object
houses_sold    float64
no_of_crimes   float64
dtype: object
```

## Q. 2) Add a new column "year" in the dataframe, which contains years only.

```
In [7]: df['year'] = df.date.dt.year
```

```
In [8]: df
```

```
Out[8]:   date      area  average_price      code  houses_sold  no_of_crimes  year
0  1995-01-01  city of london      91449  E09000001     17.0       NaN  1995
1  1995-02-01  city of london      82203  E09000001      7.0       NaN  1995
2  1995-03-01  city of london      79121  E09000001     14.0       NaN  1995
3  1995-04-01  city of london      77101  E09000001      7.0       NaN  1995
4  1995-05-01  city of london      84409  E09000001     10.0       NaN  1995
...
13544 2019-09-01      england      249942  E92000001    64605.0       NaN  2019
13549 rows × 7 columns
```

## (2.B) Add a new column "month" as 2nd column in the dataframe, which contains month only.

```
In [9]: df.insert(1, 'month', df.date.dt.month)
```

```
In [10]: df
```

```
Out[10]:   date  month      area  average_price      code  houses_sold  no_of_crimes  year
0  1995-01-01     1  city of london      91449  E09000001     17.0       NaN  1995
1  1995-02-01     2  city of london      82203  E09000001      7.0       NaN  1995
2  1995-03-01     3  city of london      79121  E09000001     14.0       NaN  1995
3  1995-04-01     4  city of london      77101  E09000001      7.0       NaN  1995
4  1995-05-01     5  city of london      84409  E09000001     10.0       NaN  1995
...
13544 2019-09-01     9      england      249942  E92000001    64605.0       NaN  2019
13545 2019-10-01    10      england      249376  E92000001    68677.0       NaN  2019
13546 2019-11-01    11      england      248515  E92000001    67814.0       NaN  2019
13547 2019-12-01    12      england      250410  E92000001       NaN       NaN  2019
13548 2020-01-01     1      england      247355  E92000001       NaN       NaN  2020
13549 rows × 8 columns
```

## Q. 3) Remove the columns 'year' and 'month' from the dataframe.

```
In [11]: df.drop(['month', 'year'], axis = 1, inplace = True)
```

### Q. 3) Remove the columns 'year' and 'month' from the dataframe.

```
In [11]: df.drop(['month', 'year'], axis = 1, inplace = True)
```

```
In [12]: df
```

```
Out[12]:
```

	date	area	average_price	code	houses_sold	no_of_crimes
0	1995-01-01	city of london	91449	E09000001	17.0	NaN
1	1995-02-01	city of london	82203	E09000001	7.0	NaN
2	1995-03-01	city of london	79121	E09000001	14.0	NaN
3	1995-04-01	city of london	77101	E09000001	7.0	NaN
4	1995-05-01	city of london	84409	E09000001	10.0	NaN
...	...	...	...	...	...	...
13544	2019-09-01	england	249942	E92000001	64605.0	NaN
13545	2019-10-01	england	249376	E92000001	68677.0	NaN
13546	2019-11-01	england	248515	E92000001	67814.0	NaN
13547	2019-12-01	england	250410	E92000001	NaN	NaN
13548	2020-01-01	england	247355	E92000001	NaN	NaN

13549 rows × 6 columns

### Q. 4) Show all the records where 'No. of Crimes' is 0. And, how many such records are there ?

```
In [13]: df[df.no_of_crimes == 0]
```

```
Out[13]:
```

	date	area	average_price	code	houses_sold	no_of_crimes
72	2001-01-01	city of london	284262	E09000001	24.0	0.0
73	2001-02-01	city of london	198137	E09000001	37.0	0.0
74	2001-03-01	city of london	189033	E09000001	44.0	0.0
75	2001-04-01	city of london	205494	E09000001	38.0	0.0
76	2001-05-01	city of london	223459	E09000001	30.0	0.0
...	...	...	...	...	...	...
178	2009-11-01	city of london	397909	E09000001	11.0	0.0
179	2009-12-01	city of london	411955	E09000001	16.0	0.0
180	2010-01-01	city of london	464436	E09000001	20.0	0.0
181	2010-02-01	city of london	490525	E09000001	9.0	0.0
182	2010-03-01	city of london	498241	E09000001	15.0	0.0

104 rows × 6 columns

### Q. 5) What is the maximum & minimum 'average\_price' per year in england ?



```
In [14]: # Maximum average Price
```

```
df.groupby(by = df.date.dt.year).average_price.max().sort_values(ascending = False).head(1)
```

```
Out[14]:
```

date	average_price
2018	1463378
	Name: average_price, dtype: int64

```
In [18]: # Minimum Average Price
```

```
df.groupby(by = df.date.dt.year).average_price.min().sort_values(ascending = True).head(1)
```

```
Out[18]:
```

date	average_price
1996	40722
	Name: average_price, dtype: int64

```
In [19]: # Maximum No_of_crimes  
df.groupby(by=['area']).no_of_crimes.max().sort_values(ascending = False).head(1)
```

```
Out[19]: area  
westminster    7461.0  
Name: no_of_crimes, dtype: float64
```

```
In [20]: # Minimum No_of_crimes  
df.groupby(by=['area']).no_of_crimes.min().sort_values(ascending = True).head(1)
```

```
Out[20]: area  
city of london    0.0  
Name: no_of_crimes, dtype: float64
```

**Q. 7) Show the total count of records of each area, where average price is less than 100000.**

```
In [ ]: df[df.average_price < 100000].groupby('area').count()
```

```
In [ ]:
```

## India Census of each district 2011

```
In [1]: #import data
```

```
import pandas as pd
```

```
In [2]: df = pd.read_csv("file.csv")
df.head()
```

Out[2]:

	District_code	State_name	District_name	Population	Male	Female	Literate	Workers	Male_Workers	Female_Workers	Christians	Sikhs	Buddhists	Others
0	1	JAMMU AND KASHMIR	Kupwara	870354	474190	396164	439654	229064	190899	38165	...	1700	5600	66
1	2	JAMMU AND KASHMIR	Badgam	753745	398041	355704	335649	214866	162578	52288	...	1489	5559	47
2	3	JAMMU AND KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	75079	53265	21814	...	658	1092	88635
3	4	JAMMU AND KASHMIR	Kargil	140802	77785	63017	86236	51873	39839	12034	...	604	1171	20126
4	5	JAMMU AND KASHMIR	Punch	476835	251899	224936	261724	161393	117677	43716	...	958	11188	83

5 rows × 25 columns

### 1. How will you hide the indexes of the dataframe

```
In [3]: df.style.hide_index()
```

District_code	State_name	District_name	Population	Male	Female	Literate	Workers	Male_Workers	Female_Workers	Cultivator_Workers	Agriculture
1	JAMMU AND KASHMIR	Kupwara	870354	474190	396164	439654	229064	190899	38165	...	34680
2	JAMMU AND KASHMIR	Badgam	753745	398041	355704	335649	214866	162578	52288	...	55299
3	JAMMU AND KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	75079	53265	21814	...	20869
4	JAMMU AND KASHMIR	Kargil	140802	77785	63017	86236	51873	39839	12034	...	8266
5	JAMMU AND KASHMIR	Punch	476835	251899	224936	261724	161393	117677	43716	...	54264
6	JAMMU AND KASHMIR	Rainuri	642415	345351	297064	364109	290912	184752	106160	...	136527

### 2. How can we set the caption/heading of the dataframe

```
In [6]: df.style.set_caption('India Census 2011')
```

```
Out[6]: India Census 2011
```

District_code	State_name	District_name	Population	Male	Female	Literate	Workers	Male_Workers	Female_Workers	Cultivator_Workers	Agriculture
0	1	JAMMU AND KASHMIR	Kupwara	870354	474190	396164	439654	229064	190899	38165	34680
1	2	JAMMU AND KASHMIR	Badgam	753745	398041	355704	335649	214866	162578	52288	55299
2	3	JAMMU AND KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	75079	53265	21814	20869
3	4	JAMMU AND KASHMIR	Kargil	140802	77785	63017	86236	51873	39839	12034	8266
4	5	JAMMU AND KASHMIR	Punch	476835	251899	224936	261724	161393	117677	43716	54264

	5	6	JAMMU AND KASHMIR	Rajouri	642415	345351	297064	364109	290912	184752	106160	136527
--	---	---	-------------------	---------	--------	--------	--------	--------	--------	--------	--------	--------

click to unscroll output; double click to hide

### 3. show the records related with the districts - New Delhi, Lucknow, Jaipur

In [7]: df[df['District\_name'].isin(['New Delhi', 'Lucknow', 'Jaipur'])]

Out[7]:	District_code	State_name	District_name	Population	Male	Female	Literate	Workers	Male_Workers	Female_Workers	...	Christians	Sikhs	Buddhi
	93	94	NCT OF DELHI	New Delhi	142004	77942	64062	114179	59541	46940	12601	...	4852	2933
	109	110	RAJASTHAN	Jaipur	6626178	3468507	3157671	4300965	2464893	1714947	749946	...	12708	18782
	156	157	UTTAR PRADESH	Lucknow	4589838	2394476	2195362	3127260	1542806	1226399	316407	...	20493	23883

3 rows × 25 columns

### 4. Calculate State-wise

Total number of population

In [11]: df.groupby('State\_name').Population.sum().sort\_values()

Out[11]: State_name	
LAKSHADWEEP	64473
DAMAN AND DIU	243247
DADRA AND NAGAR HAVELI	343709
ANDAMAN AND NICOBAR ISLANDS	380581
SIKKIM	610577
CHANDIGARH	1055450
MIZORAM	1097286
PONDICHERRY	1247953
ARUNACHAL PRADESH	1383727
GOA	1458545
NAGALAND	1978502
MANIPUR	2855794
MEGHALAYA	2966889
TRIPURA	3673917
HIMACHAL PRADESH	6864602
UTTARAKHAND	10086292
JAMMU AND KASHMIR	12541302
NCT OF DELHI	16787941
HARYANA	25351462
CHHATTISGARH	25545198
PUNJAB	27743338
ASSAM	31205576
JHARKHAND	32988134
KERALA	33406061
ORISSA	41974218
GUJARAT	60439692
KARNATAKA	61095297
RAJASTHAN	68548437
TAMIL NADU	72147030
MADHYA PRADESH	72626899
ANDHRA PRADESH	84580777
WEST BENGAL	91276115
BIHAR	104099452
MAHARASHTRA	112374333
UTTAR PRADESH	199812341

Name: Population, dtype: int64

Total no. of population with different religions

In [13]: df.groupby('State\_name')[['Hindus', 'Muslims', 'Christians', 'Sikhs', 'Buddhists', 'Jains']].sum()

Out[13]:

State_name	Hindus	Muslims	Christians	Sikhs	Buddhists	Jains
ANDAMAN AND NICOBAR ISLANDS	264296	32413	80984	1286	338	31
ANDHRA PRADESH	74824149	8082412	1129784	40244	36692	53849
ARUNACHAL PRADESH	401876	27045	418732	3287	162815	771
ASSAM	19180759	10679345	1165867	20672	54993	25949
BIHAR	86078686	17557809	129247	23779	25453	18914
CHANDIGARH	852574	51447	8720	138329	1160	1960
CHHATTISGARH	23819789	514998	490542	70036	70467	61510
DADRA AND NAGAR HAVELI	322857	12922	5113	217	634	1186
DAMAN AND DIU	220150	19277	2820	172	217	287
GOA	963877	121564	366130	1473	1095	1109
GUJARAT	53533988	5846761	316178	58246	30483	579654
HARYANA	22171128	1781342	50353	1243752	7514	52613
HIMACHAL PRADESH	6532765	149881	12646	79896	78659	1805
JAMMU AND KASHMIR	3566674	8567485	35631	234848	112584	2490
JHARKHAND	22376051	4793994	1418608	71422	8956	14974
KARNATAKA	51317472	7893065	1142647	28773	95710	440280
KERALA	18282492	8873472	6141269	3814	4752	4489
LAKSHADWEEP	1788	62268	317	8	10	11
MADHYA PRADESH	66007121	4774695	213282	151412	216052	567028
MAHARASHTRA	89703057	12971152	1080073	223247	6531200	1400349
MANIPUR	1181876	239836	1179043	1527	7084	1692
MEGHALAYA	342078	130399	2213027	3045	9864	627
MIZORAM	30136	14832	956331	286	93411	376
NAGALAND	173054	48963	1739651	1890	6759	2655
NOT IN DELHI	4216160	2156281	1162000	572501	10140	162001

## 5. how many male workers were there in Maharashtra state

```
In [25]: df[df.State_name == 'MAHARASHTRA']['Male_Workers'].sum()
Out[25]: 32616875
```

## 6. How to set a column as index of the dataframe

In [26]: df.set\_index('District\_code')

Out[26]:

District_code	State_name	District_name	Population	Male	Female	Literate	Workers	Male_Workers	Female_Workers	Cultivator_Workers	...	Christian
1	JAMMU AND KASHMIR	Kupwara	870354	474190	396164	439654	229064	190899	38165	34680	...	170
2	JAMMU AND KASHMIR	Badgam	753745	398041	355704	335649	214866	162578	52288	55299	...	148
3	JAMMU AND KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	75079	53265	21814	20869	...	61
4	JAMMU AND KASHMIR	Kargil	140802	77785	63017	86236	51873	39839	12034	8266	...	61
5	JAMMU AND KASHMIR	Punch	476835	251899	224936	261724	161393	117677	43716	54264	...	91
...	...	...	...	...	...	...	...	...	...	...	...	...
636	PONDICHERRY	Mahe	41816	19143	22673	36470	11802	9093	2709	43	...	91

## 7. Adding a Suffix to the column name

```
In [28]: df.add_suffix('_suffix')
Out[28]: District_code_suffix State_name_suffix District_name_suffix Population_suffix Male_suffix Female_suffix Literate_suffix Workers_suffix Male_Workers_su
```

## 7. Adding a Suffix to the column name

```
In [28]: df.add_suffix('_suffix')
```

```
Out[28]:
```

	District_code_suffix	State_name_suffix	District_name_suffix	Population_suffix	Male_suffix	Female_suffix	Literate_suffix	Workers_suffix	Male_Workers_su
0	1	JAMMU AND KASHMIR	Kupwara	870354	474190	396164	439654	229064	1908
1	2	JAMMU AND KASHMIR	Badgam	753745	398041	355704	335649	214866	1628
2	3	JAMMU AND KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	75079	534
3	4	JAMMU AND KASHMIR	Kargil	140802	77785	63017	66236	51873	398
4	5	JAMMU AND KASHMIR	Punch	476835	251899	224936	261724	161393	1176
...	...	...	...	...	...	...	...	...	...
635	636	PONDICHERRY	Mahe	41816	19143	22673	36470	11802	96
636	637	PONDICHERRY	Karikal	200222	97809	102413	154916	68301	53
637	638	ANDAMAN AND NICOBAR ISLANDS	Nicobars	36842	20727	16115	25332	17125	125
638	639	ANDAMAN AND NICOBAR ISLANDS	North AND Middle Andaman	105597	54861	50736	78683	38579	306
639	640	ANDAMAN AND NICOBAR ISLANDS	South Andaman	238142	127283	110859	190266	96631	771

640 rows × 25 columns

## 8. Adding a prefix to the column name

```
In [29]: df.add_prefix('prefix_')
```

- -

```
In [1]: #UDEMY DATA ANALYSIS
#Importing

import pandas as pd
```

```
In [2]: df = pd.read_csv('file.csv')
df.head()
```

	course_id	course_title	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject
0	288942	#1 Piano Hand Coordination: Play 10th Ballad ...	True	35	3137	18	68	All Levels	1.5 hours	2014-09-18T05:07:05Z	Musical Instruments
1	1170074	#10 Hand Coordination - Transfer Chord Ballad ...	True	75	1593	1	41	Intermediate Level	1 hour	2017-04-12T19:06:34Z	Musical Instruments
2	1193886	#12 Hand Coordination: Let your Hands dance w/...	True	75	482	1	47	Intermediate Level	1.5 hours	2017-04-26T18:34:57Z	Musical Instruments
3	1116700	#4 Piano Hand Coordination: Fun Piano Runs in ...	True	75	850	3	43	Intermediate Level	1 hour	2017-02-21T23:48:18Z	Musical Instruments
4	1120410	#5 Piano Hand Coordination: Piano Runs in 2 ...	True	75	940	3	32	Intermediate Level	37 mins	2017-02-21T23:44:49Z	Musical Instruments

## 1. what are the different subjects for which udemy is offering courses

```
In [6]: df['subject'].unique()
```

```
Out[6]: array(['Musical Instruments', 'Business Finance', 'Graphic Design', 'Web Development'], dtype=object)
```

## 2. which subjects has the maximum number of courses

```
In [12]: df['subject'].value_counts()
```

```
Out[12]: Web Development      1200
Business Finance        1199
Musical Instruments     680
Graphic Design          603
Name: subject, dtype: int64
```

## 3. Show all the courses which are free of cost

```
In [14]: df[df.is_paid == False]
```

	course_id	course_title	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject
41	286070	5 lecciones que todo guitarrista debe tomar	False	Free	4452	263	14	Beginner Level	1 hour	2014-08-23T05:08:14Z	Musical Instruments
49	696630	7 Ways A Beginner Guitarist Can Sound Better, ...	False	Free	4529	193	7	Beginner Level	36 mins	2015-12-21T18:50:50Z	Musical Instruments
57	955914	A beginner's guide to fingerpicking and strummm...	False	Free	3481	29	20	Beginner Level	2 hours	2016-09-13T21:51:59Z	Musical Instruments
67	270976	A how to guide in HTML	False	Free	7318	205	8	Beginner Level	35 mins	2014-08-10T20:19:10Z	Web Development
74	1214144	¡Triunfar en La Bolsa de Valores No Requiere d...	False	Free	338	7	6	Beginner Level	1 hour	2017-05-30T14:30:12Z	Business Finance
...	...	...	...	...	...	...	...	...	...	...	...
3620	552598	WordPress in under an hour - Create/manage you...	False	Free	6474	86	15	All Levels	1 hour	2015-07-15T00:21:56Z	Web Development

3638	1068590	Wordpress Website Mastery 2017	False	Free	5181	64	22	Beginner Level	2.5 hours	2017-01-18T17:01:40Z	Web Development
3652	752706	Write quicker HTML5 and CSS 3; productivity ha...	False	Free	13016	286	15	All Levels	1.5 hours	2016-03-03T17:45:31Z	Web Development
3674	1257976	Your First 10 Guitar Lessons - Learn how to pl...	False	Free	924	4	18	Beginner Level	1 hour	2017-06-29T00:29:36Z	Musical Instruments
3680	1225194	Zoho Books Gestion Financière d'Entreprise pas...	False	Free	229	0	33	All Levels	2 hours	2017-05-26T16:45:55Z	Business Finance

310 rows × 11 columns

#### 4. Show all the courses which are paid

In [15]:	df[df.is_paid == True]										
Out[15]:	course_id	course_title	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject
0	288942	#1 Piano Hand Coordination: Play 10th Ballad i...	True	35	3137	18	68	All Levels	1.5 hours	2014-09-18T05:07:05Z	Musical Instruments
1	1170074	#10 Hand Coordination - Transfer Chord Ballad ...	True	75	1593	1	41	Intermediate Level	1 hour	2017-04-12T19:06:34Z	Musical Instruments
2	1193886	#12 Hand Coordination: Let your Hands dance wi...	True	75	482	1	47	Intermediate Level	1.5 hours	2017-04-26T18:34:57Z	Musical Instruments
3	1116700	#4 Piano Hand Coordination: Fun Piano Runs in ...	True	75	850	3	43	Intermediate Level	1 hour	2017-02-21T23:48:18Z	Musical Instruments
4	1120410	#5 Piano Hand Coordination: Piano Runs in 2 ...	True	75	940	3	32	Intermediate Level	37 mins	2017-02-21T23:44:49Z	Musical Instruments
...	...	...	...	...	...	...	...	...	...	...	...
3676	498488	Your First Successful Forex Trades - With Case...	True	200	1079	34	16	All Levels	2.5 hours	2015-05-26T20:48:48Z	Business Finance
3677	328960	Your Own Site In 45 Min: The Complete Wordpress...	True	120	1566	29	36	All Levels	4 hours	2015-04-20T22:15:17Z	Web Development
3678	552700	Your Second Course on Piano: Two Handed Playing	True	70	1018	12	22	Beginner Level	5 hours	2015-10-26T20:04:21Z	Musical Instruments
3679	631754	Zend Framework 2: Learn the PHP framework ZF2 ...	True	40	723	130	37	All Levels	6.5 hours	2015-11-11T18:55:45Z	Web Development
3681	964478	Zombie Apocalypse Photoshop Actions	True	50	12	1	15	All Levels	1.5 hours	2016-09-26T22:19:48Z	Graphic Design

3372 rows × 11 columns

#### 5. what are the top selling courses

In [16]: df.sort_values('num_subscribers', ascending = False)												
Out[16]:	course_id	course_title	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject	
	2230	Learn HTML5 Programming From Scratch	False	Free	268923	8629	45	All Levels	10.5 hours	2013-02-14T07:03:41Z	Web Development	
	776	Coding for Entrepreneurs Basic	False	Free	161029	279	27	Beginner Level	3.5 hours	2013-06-09T15:51:55Z	Web Development	
	3385	The Web Developer Bootcamp	True	200	121584	27445	342	All Levels	43 hours	2015-11-02T21:13:27Z	Web Development	
	640	Build Your First Website in 1 Week with HTML5	False	Free	120291	5924	30	Beginner Level	3 hours	2014-04-08T16:21:30Z	Web Development	
	...	...	...	...	...	...	...	...	...	...	...	
	3316	The Complete Web Developer Course 2.0	True	200	114512	22412	304	All Levels	30.5 hours	2016-03-08T22:28:36Z	Web Development	
	...	...	...	...	...	...	...	...	...	...	...	
	2290	Learn Pirates of the Caribbean by Ear on the P...	True	20	0	0	6	All Levels	32 mins	2017-05-22T17:14:43Z	Musical Instruments	
	1267	Financial Statement Auditing Cycles	True	50	0	0	9	Intermediate Level	2 hours	2017-06-29T23:20:10Z	Business Finance	
	2070	Kickstarter success in 5 easy steps	True	20	0	0	12	All Levels	31 mins	2017-05-16T14:55:28Z	Business Finance	
	1931	Introduction to Project Management for Finance...	True	50	0	0	9	Beginner Level	2 hours	2017-07-03T21:40:32Z	Business Finance	
	3165	Harassing you and Avoid Ba...	True	20	0	0	7	Beginner Level	37 mins	2017-02-02T16:22:37Z	Business Finance	
	3682 rows × 11 columns											

## 6. what are the least selling courses

In [19]: df.sort_values('num_subscribers')												
Out[19]:	course_id	course_title	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject	
	649	Building a Balanced Scorecard	True	50	0	0	11	Intermediate Level	2 hours	2017-07-03T21:38:22Z	Business Finance	
	3259	The Cash Flow Statement - An Introduction	True	50	0	0	10	Beginner Level	1.5 hours	2017-06-28T16:05:51Z	Business Finance	
	3261	The Complete Adobe Spark Course	True	150	0	0	29	All Levels	2 hours	2017-07-03T17:39:57Z	Graphic Design	
	910	Create Beautiful Image Maps for Your Website	True	20	0	0	8	Intermediate Level	37 mins	2016-01-18T17:56:36Z	Graphic Design	
	3328	The Cuckoo Strategy to get European Funding	True	160	0	0	6	All Levels	1 hour	2016-01-06T02:38:23Z	Business Finance	
	...	...	...	...	...	...	...	...	...	...	...	
	3316	The Complete Web Developer Course 2.0	True	200	114512	22412	304	All Levels	30.5 hours	2016-03-08T22:28:36Z	Web Development	
	640	Build Your First Website in 1 Week	False	Free	120291	5924	30	Beginner Level	3 hours	2014-04-08T16:21:30Z	Web Development	

3385	625204	The Web Developer Bootcamp	True	200	121584	27445	342	All Levels	43 hours	2015-11-02T21:13:27Z	Web Development
776	59014	Coding for Entrepreneurs Basic	False	Free	161029	279	27	Beginner Level	3.5 hours	2013-06-09T15:51:55Z	Web Development
2230	41295	Learn HTML5 Programming From Scratch	False	Free	268923	8629	45	All Levels	10.5 hours	2013-02-14T07:03:41Z	Web Development

3682 rows × 11 columns

## 7. show all the courses of graphic design where the price is below 100

```
In [22]: df[(df.subject == 'Graphic Design') & (df.price < '100')]
```

```
Out[22]:   course_id  course_title  is_paid  price  num_subscribers  num_reviews  num_lectures  level  content_duration  published_timestamp  subject
```

## 8. list out all the courses that are related with python

```
In [23]: df[df.course_title.str.contains('Python')]
```

course_id	course_title	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject
216	Advanced Scalable Python Web Development Using...	True	120	1299	56	71	Intermediate Level	14 hours	2016-08-11T22:09:24Z	Web Development
777	Coding for Entrepreneurs: Learn Python, Django...	True	195	23412	799	251	All Levels	45 hours	2013-04-08T00:46:14Z	Web Development
814	Complete Python Web Course: Build	True	110	7489	941	173	All Levels	16 hours	2015-11-08T20:57:35Z	Web Development

## 9. what are the courses that are published in 2015

```
In [24]: df.dtypes
```

```
Out[24]: course_id          int64
course_title        object
is_paid            bool
price             object
num_subscribers    int64
num_reviews        int64
num_lectures       int64
level              object
content_duration   object
published_timestamp object
subject            object
dtype: object
```

```
In [25]: df['published_timestamp'] = pd.to_datetime(df.published_timestamp)
```

```
In [26]: df.dtypes
```

```
Out[26]: course_id          int64
course_title        object
is_paid            bool
price             object
num_subscribers    int64
num_reviews        int64
num_lectures       int64
level              object
content_duration   object
published_timestamp datetime64[ns, UTC]
subject            object
dtype: object
```

```
In [27]: df['Year'] = df['published_timestamp'].dt.year
```

In [28]: df

	course_id	course_title	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject
0	288942	#1 Piano Hand Coordination: Play 10th Ballad ...	True	35	3137	18	68	All Levels	1.5 hours	2014-09-18 05:07:05+00:00	Musical Instruments
1	1170074	#10 Hand Coordination - Transfer Chord Ballad ...	True	75	1593	1	41	Intermediate Level	1 hour	2017-04-12 19:06:34+00:00	Musical Instruments
2	1193886	#12 Hand Coordination: Let your Hands dance w/...	True	75	482	1	47	Intermediate Level	1.5 hours	2017-04-26 18:34:57+00:00	Musical Instruments
3	1116700	#4 Piano Hand Coordination: Fun Piano Runs in ...	True	75	850	3	43	Intermediate Level	1 hour	2017-02-21 23:48:18+00:00	Musical Instruments
4	1120410	#5 Piano Hand Coordination: Piano Runs in 2 ...	True	75	940	3	32	Intermediate Level	37 mins	2017-02-21 23:44:49+00:00	Musical Instruments
...	...	...	...	...	...	...	...	...	...	...	...
3677	328960	Your Own Site in 45 Min: The Complete Wordpress...	True	120	1566	29	36	All Levels	4 hours	2015-04-20 22:15:17+00:00	Web Development
3678	552700	Your Second Course on Piano: Two Handed Playing	True	70	1018	12	22	Beginner Level	5 hours	2015-10-26 20:04:21+00:00	Musical Instruments
3679	631754	Zend Framework 2: Learn the PHP	True	40	723	130	37	All Levels	6.5 hours	2015-11-11 18:55:45+00:00	Web Development
3680	1225194	Zoho Books Gestion Financière d'Entreprise pas...	False	Free	229	0	33	All Levels	2 hours	2017-05-26 16:45:55+00:00	Business Finance
3681	964478	Zombie Apocalypse Photoshop Actions	True	50	12	1	15	All Levels	1.5 hours	2016-09-26 22:19:48+00:00	Graphic Design

3682 rows × 12 columns

In [29]: df[df.Year == 2015]

	course_id	course_title	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject
10	591880	1 - Concepts of Statistics For Beginners Step ...	True	200	273	4	15	Beginner Level	31 mins	2015-08-30 22:48:34+00:00	Business Finance
16	302450	10 Numbers Every Business Owner Should Know	True	20	13	1	9	All Levels	1 hour	2015-03-08 19:11:24+00:00	Business Finance
19	384928	101 Blues riffs - learn how the harmonica super...	True	200	1350	65	55	Intermediate Level	6.5 hours	2015-01-04 21:14:31+00:00	Musical Instruments
23	550842	16 Guitar Chords to Jam With - (Beginner - Int...	True	20	1224	19	20	Beginner Level	1 hour	2015-07-10 19:53:56+00:00	Musical Instruments

26	486240	6 Easy Steps To Investment And Avoiding Traps	True	20	828	1	20	All Levels	1 hour	2015-04-27 23:18:59+00:00	Business Finance	2
3675	551170	Your First Course on Piano	True	70	2002	34	33	Beginner Level	4 hours	2015-07-16 17:07:49+00:00	Musical Instruments	2
3676	498488	Your First Successful Forex Trades - With Case...	True	200	1079	34	16	All Levels	2.5 hours	2015-05-26 20:48:48+00:00	Business Finance	2
3677	328860	Your Own Site in 45 Min: The Complete Wordpress...	True	120	1566	29	36	All Levels	4 hours	2015-04-20 22:15:17+00:00	Web Development	2
3678	552700	Your Second Course on Piano: Two Handed Playing	True	70	1018	12	22	Beginner Level	5 hours	2015-10-26 20:04:21+00:00	Musical Instruments	2
3679	631754	Zend Framework 2: Learn the PHP framework ZF2... [4]	True	40	723	130	37	All Levels	6.5 hours	2015-11-11 18:35:48+00:00	Web Development	2

1014 rows × 12 columns

## 10. What are the max. number of subscribers for Each level of courses

```
In [34]: df.groupby("level")['num_subscribers'].max()
```

```
Out[34]: level
```

```
In [10]: #importing Libraries
import pandas as pd
import seaborn as sns
```

```
In [2]: #importing dataset
df = pd.read_csv('file.csv')
df.head()
```

```
Out[2]:
```

	Show_Id	Category	Title	Director	Cast	Country	Release_Date	Rating	Duration	Type	Description
0	s1	TV Show	3%	NaN	João Miguel, Bianca Comparato, Michel Gomes, R...	Brazil	August 14, 2020	TV-MA	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi &...	In a future where the elite inhabit an island ...
1	s2	Movie	07:19	Jorge Michel Grau	Demián Bichir, Héctor Bonilla, Oscar Serrano, ...	Mexico	December 23, 2016	TV-MA	93 min	Dramas, International Movies	After a devastating earthquake hits Mexico Citt...
2	s3	Movie	23:59	Gilbert Chan	Tedd Chan, Stella Chung, Henley Hill, Lawrence ...	Singapore	December 20, 2018	R	78 min	Horror Movies, International Movies	When an army recruit is found dead, his fellow...
3	s4	Movie	9	Shane Acker	Elijah Wood, John C. Reilly, Jennifer Connelly...	United States	November 16, 2017	PG-13	80 min	Action & Adventure, Independent Movies, Sci-Fi...	In a postapocalyptic world, rag-doll robots hi...
4	s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin Spacey, Kate Bosworth, Aar...	United States	January 1, 2020	PG-13	123 min	Dramas	A brilliant group of students become card-coun...

## removing duplicate records if any

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

```
Out[3]:
```

	Show_Id	Category	Title	Director	Cast	Country	Release_Date	Rating	Duration	Type	Description
6300	s684	Movie	Backfire	Dave Patten	Black Deniro, Byron "Squally" Vinson, Dominic ...	United States	April 5, 2019	TV-MA	97 min	Dramas, Independent Movies, Thrillers	When two would-be robbers accidentally kill a ...
	6622	s6621	The Lost Okoroshi	Abba T. Makama	Seun Ajayi, Judith Audu, Tope Tedela, Ifi Enana...	Nigeria	September 4, 2020	TV-MA	94 min	Comedies, Dramas, Independent Movies	A disillusioned security guard transforms into...

```
In [5]: df.drop_duplicates( inplace = True )
```

```
In [6]: df[df.duplicated()]
```

```
Out[6]:
```

	Show_Id	Category	Title	Director	Cast	Country	Release_Date	Rating	Duration	Type	Description

```
In [7]: df
```

```
Out[7]:
```

	Show_Id	Category	Title	Director	Cast	Country	Release_Date	Rating	Duration	Type	Description
0	s1	TV Show	3%	NaN	João Miguel, Bianca Comparato, Michel Gomes, R...	Brazil	August 14, 2020	TV-MA	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi &...	In a future where the elite inhabit an island ...
1	s2	Movie	07:19	Jorge Michel Grau	Demián Bichir, Héctor Bonilla, Oscar Serrano, ...	Mexico	December 23, 2016	TV-MA	93 min	Dramas, International Movies	After a devastating earthquake hits Mexico Citt...
2	s3	Movie	23:59	Gilbert Chan	Tedd Chan, Stella Chung, Henley Hill, Lawrence ...	Singapore	December 20, 2018	R	78 min	Horror Movies, International Movies	When an army recruit is found dead, his fellow...
3	s4	Movie	9	Shane Acker	Elijah Wood, John C. Reilly, Jennifer Connelly...	United States	November 16, 2017	PG-13	80 min	Action & Adventure, Independent Movies, Sci-Fi...	In a postapocalyptic world, rag-doll robots hi...
4	s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin Spacey, Kate Bosworth, Aar...	United States	January 1, 2020	PG-13	123 min	Dramas	A brilliant group of students become card-coun...
...	...	...	...	...	...	...	...	...	...	...	...
7784	s7783	Movie	Zozo	Josef Fares	Imad Creidi, Antoinette Turk, Elias Gergi, Car...	Sweden, Czech Republic, United Kingdom, Denmark...	October 19, 2020	TV-MA	99 min	Dramas, International Movies	When Lebanon's Civil War deprives Zozo of his ...
					Vicky Kaushal, Sarah Jane Dias					Dramas,	A scrappy but poor

7785	s7784	Movie	Zubaan	Mozez Singh	Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	TV-14	111 min	Documentaries, International Movies, Music & Musicals	Scrappy but poor boy worms his way into a ty...
7786	s7785	Movie	Zulu Man in Japan	NaN	Nasty C	NaN	September 25, 2020	TV-MA	44 min	Documentaries, International Movies, Music & M...	In this documentary, South African rapper Nast...
7787	s7786	TV Show	Zumbo's Just Desserts	NaN	Adriano Zumbo, Rachel Khoo	Australia	October 31, 2020	TV-PG	1 Season	International TV Shows, Reality TV	Dessert wizard Adriano Zumbo looks for the nex...
7788	s7787	Movie	ZZ TOP: THAT LITTLE OL' BAND FROM TEXAS	Sam Dunn	NaN	United Kingdom, Canada, United States	March 1, 2020	TV-MA	90 min	Documentaries, Music & Musicals	This documentary delves into the mystique behi...

7787 rows × 11 columns

## showing Null values using the heat map if any

In [9]: `df.isnull().sum()`

```
Out[9]: Show_Id      0
Category      0
Title         0
Director     2388
Cast          718
Country       507
Release_Date  10
Rating         7
Duration       0
Type          0
Description    0
dtype: int64
```

In [11]: `sns.heatmap(df.isnull())`

Out[11]: <Axes: >



In which year highest number of TV shows & movies were released? Show with bar graph

In [12]: `df.dtypes`

```
Out[12]: Show_Id      object
Category      object
Title         object
Director     object
Cast          object
Country       object
Release_Date  object
Rating        object
Duration      object
Type          object
Description   object
dtype: object
```

In [13]: `df['Date_N'] = pd.to_datetime(df['Release_Date'])`

In [14]: `df.dtypes`

```
Out[14]: Show_Id      object
Category      object
Title         object
Director     object
Cast          object
Country       object
Release_Date  object
Rating        object
Duration      object
Type          object
Description   object
Date_N        datetime64[ns]
dtype: object
```

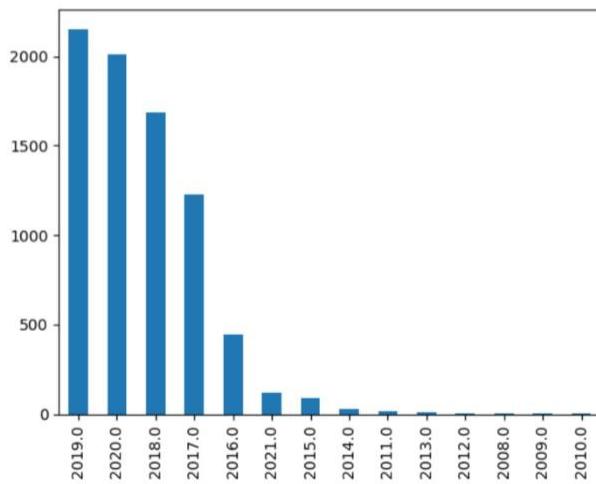
In [15]: `df['Date_N'].dt.year.value_counts()`

```
Out[15]: 2019.0    2153
2020.0    2009
```

```
2014.0      25
2011.0      13
2013.0      11
2012.0       3
2008.0       2
2009.0       2
2010.0       1
Name: Date_N, dtype: int64
```

```
In [18]: df['Date_N'].dt.year.value_counts().plot( kind = 'bar' )
```

```
Out[18]: <Axes: >
```



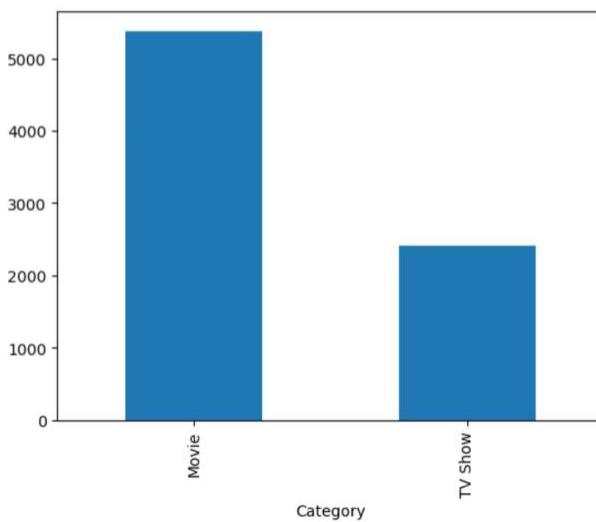
How many movies and TV shows are there in the dataset? show with Bar graph

```
In [25]: df.groupby('Category').Category.count()
```

```
Out[25]: Category
Movie      5377
TV Show    2410
Name: Category, dtype: int64
```

```
In [23]: df.groupby('Category').Category.count().plot( kind = 'bar' )
```

```
Out[23]: <Axes: xlabel='Category'>
```



## Show all the movies that were released in the year 2020

```
In [27]: df['Year'] = df['Date_N'].dt.year
```

```
In [28]: df
```

	Show_Id	Category	Title	Director	Cast	Country	Release_Date	Rating	Duration	Type	Description	Date_N	Year
0	s1	TV Show	3%	NaN	João Miguel, Bianca Comparato, Michel Gomes, R...	Brazil	August 14, 2020	TV-MA	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi &...	In a future where the elite inhabit an island ...	2020-08-14	2020.0
1	s2	Movie	07:19	Jorge Michel Grau	Demién Bichir, Héctor Bonilla, Oscar Serrano, ...	Mexico	December 23, 2016	TV-MA	93 min	Dramas, International Movies	After a devastating earthquake hits Mexico City...	2016-12-23	2016.0
2	s3	Movie	23:59	Gilbert Chan	Tedd Chan, Stella Chung, Henley Hill, Lawrence ...	Singapore	December 20, 2018	R	78 min	Horror Movies, International Movies	When an army recruit is found dead, his fellow...	2018-12-20	2018.0
3	s4	Movie	9	Shane Acker	Elijah Wood, John C. Reilly, Jennifer Connolly...	United States	November 16, 2017	PG-13	80 min	Action & Adventure, Independent Movies, Sci-Fi...	In a postapocalyptic world, rag-doll robots hi...	2017-11-16	2017.0
4	s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin Spacey, Kate Bosworth, Aar...	United States	January 1, 2020	PG-13	123 min	Dramas	A brilliant group of students become card-cou...	2020-01-01	2020.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7784	s7783	Movie	Zozo	Josef Fares	Imad Creidi, Antoinette Turk, Elias Gergi, Car...	Sweden, Czech Republic, United Kingdom, Denmark...	October 19, 2020	TV-MA	99 min	Dramas, International Movies	When Lebanon's Civil War deprives Zozo of his ...	2020-10-19	2020.0
7785	s7784	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...	2019-03-02	2019.0
7786	s7785	Movie	Zulu Man in Japan	NaN	Nasty C	NaN	September 25, 2020	TV-MA	44 min	Documentaries, International Movies, Music & M...	In this documentary, South African rapper Nast...	2020-09-25	2020.0
7787	s7786	TV Show	Zumbo's Just Desserts	NaN	Adriano Zumbo, Rachel Khoo	Australia	October 31, 2020	TV-PG	1 Season	International TV Shows, Reality TV	Dessert wizard Adriano Zumbo looks for the nex...	2020-10-31	2020.0
7788	s7787	Movie	ZZ TOP: THAT LITTLE OL' BAND FROM TEXAS	Sam Dunn	NaN	United Kingdom, Canada, United States	March 1, 2020	TV-MA	90 min	Documentaries, Music & Musicals	This documentary delves into the mystique behi...	2020-03-01	2020.0

7787 rows × 13 columns

```
In [32]: df[(df['Category'] == 'Movie') & (df['Year'] == 2020)]
```

	Show_Id	Category	Title	Director	Cast	Country	Release_Date	Rating	Duration	Type	Description	Date_N	Year
4	s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin Spacey, Kate Bosworth, Aar...	United States	January 1, 2020	PG-13	123 min	Dramas	A brilliant group of students become card-cou...	2020-01-01	2020.0
6	s7	Movie	122	Yasir Al Yasiri	Amina Khalil, Ahmed Dawood, Tarek Lotfy, Ahmed...	Egypt	June 1, 2020	TV-MA	95 min	Horror Movies, International Movies	After an awful accident, a couple admitted to ...	2020-06-01	2020.0
14	s15	Movie	3022	John Sults	Omar Epps, Kate Walsh, Miranda Cosgrove, ...	United States	March 19, 2020	R	91 min	Independent Movies, Sci-Fi & Fantasy, Thrillers	Stranded when the Earth is suddenly destroyed ...	2020-03-19	2020.0

14	s15	Movie	3022	John Suits	Omar Epps, Kate Walsh, Miranda Cosgrove, Angus...	United States	March 19, 2020	R	91 min	Independent Movies, Sci-Fi & Fantasy, Thrillers	Stranded when the Earth is suddenly destroyed ...	2020-03-19	2020.0
27	s28	Movie	#Alive	Cho Il	Yoo Ah-in, Park Shin-hye	South Korea	September 8, 2020	TV-MA	99 min	Horror Movies, International Movies, Thrillers	As a grisly virus rampages a city, a lone man ...	2020-09-08	2020.0
28	s29	Movie	#AnneFrank - Parallel Stories	Sabina Fedeli, Anna Migotto	Helen Mirren, Gengher Gatti	Italy	July 1, 2020	TV-14	95 min	Documentaries, International Movies	Through her diary, Anne Frank's story is retold...	2020-07-01	2020.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7762	s7761	Movie	Zaki Chan	Wael Ihsan	Ahmed Helmy, Yasmin Abdulaziz, Hassan Hosny, H...	Egypt	May 19, 2020	TV-PG	109 min	Comedies, International Movies, Romantic Movies	An unqualified young man has his work cut out ...	2020-05-19	2020.0
7783	s7782	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...	2020-01-11	2020.0
7784	s7783	Movie	Zozo	Josef Fares	Imad Creidl, Antoinette Turk, Elias Gergl, Car...	Sweden, Czech Republic, United Kingdom, Denmar...	October 19, 2020	TV-MA	99 min	Dramas, International Movies	When Lebanon's Civil War deprives Zozo of his ...	2020-10-19	2020.0
7786	s7785	Movie	Zulu Man in Japan	NaN	Nasty C	NaN	September 25, 2020	TV-MA	44 min	Documentaries, International Movies, Music & M...	In this documentary, South African rapper Nast...	2020-09-25	2020.0
7788	s7787	Movie	ZZ TOP: THAT LITTLE OL' BAND FROM TEXAS	Sam Dunn	NaN	United Kingdom, Canada, United States	March 1, 2020	TV-MA	90 min	Documentaries, Music & Musicals	This documentary delves into the mystique behi...	2020-03-01	2020.0

show only the titles of all the TV shows that were released in india only

```
In [33]: df[(df['Category'] == 'TV Show') & (df['Country'] == 'India')]['Title']
```

```
Out[33]: 86          21 Sarfarosh: Saragarhi 1897
132                  7 (Seven)
340          Agent Raghav
364          Akbar Birbal
533          Anjaan: Rural Myths
...
6249      The Creative Indians
6400  The Golden Years with Javed Akhtar
6469      The House That Made Me
7294          Typewriter
7705      Yeh Meri Family
Name: Title, Length: 71, dtype: object
```

```
In [ ]:
```