

# City\_Wise

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib as plt
```

```
In [6]: xls = pd.ExcelFile(r"C:\Users\dell\Downloads\App Analytics Report-06.05.2023 (1).xlsx")
city_wise = pd.read_excel(xls, 'Citiwise Report')
```

```
In [7]: city_wise
```

```
Out[7]:
```

	Town/City	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Conversions	Total revenue
0	Bengaluru	6097	5685	15013	0.769385	2.462359	762.20550	607200	62939	0
1	Patna	1594	1467	2127	0.440646	1.334379	98.22208	38830	6980	0
2	Hyderabad	1038	920	1578	0.569264	1.520231	243.69080	96826	34103	0
3	Indore	983	915	1241	0.426460	1.262462	67.89115	21383	4121	0
4	Lucknow	897	839	1125	0.450180	1.254181	83.40580	21041	3650	0
...	...	...	...	...	...	...	...	...	...	...
569	Titwala	0	0	0	0.000000	0.000000	0.00000	1	1	0
570	Vagator	0	0	0	0.000000	0.000000	0.00000	16	16	0
571	Vellakoil	0	0	0	0.000000	0.000000	0.00000	1	0	0
572	Wardha	0	0	0	0.000000	0.000000	0.00000	1	1	0
573	Washington	0	0	0	0.000000	0.000000	0.00000	1	1	0

574 rows × 10 columns

```
In [8]: city_wise.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574 entries, 0 to 573
Data columns (total 10 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Town/City                            574 non-null    object
1   Users                                574 non-null    int64
2   New users                            574 non-null    int64
3   Engaged sessions                     574 non-null    int64
4   Engagement rate                      574 non-null    float64
5   Engaged sessions per user            574 non-null    float64
6   Average engagement time              574 non-null    float64
7   Event count                          574 non-null    int64
8   Conversions                          574 non-null    int64
9   Total revenue                       574 non-null    int64
dtypes: float64(3), int64(6), object(1)
memory usage: 45.0+ KB
```

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

Out[9]: Town/City  
Users 0  
New users 0  
Engaged sessions 0  
Engagement rate 0  
Engaged sessions per user 0  
Average engagement time 0  
Event count 0  
Conversions 0  
Total revenue 0  
dtype: int64

In [10]: city\_wise["Town/City"].nunique()

Out[10]: 574

In [11]: city\_wise.describe().transpose()

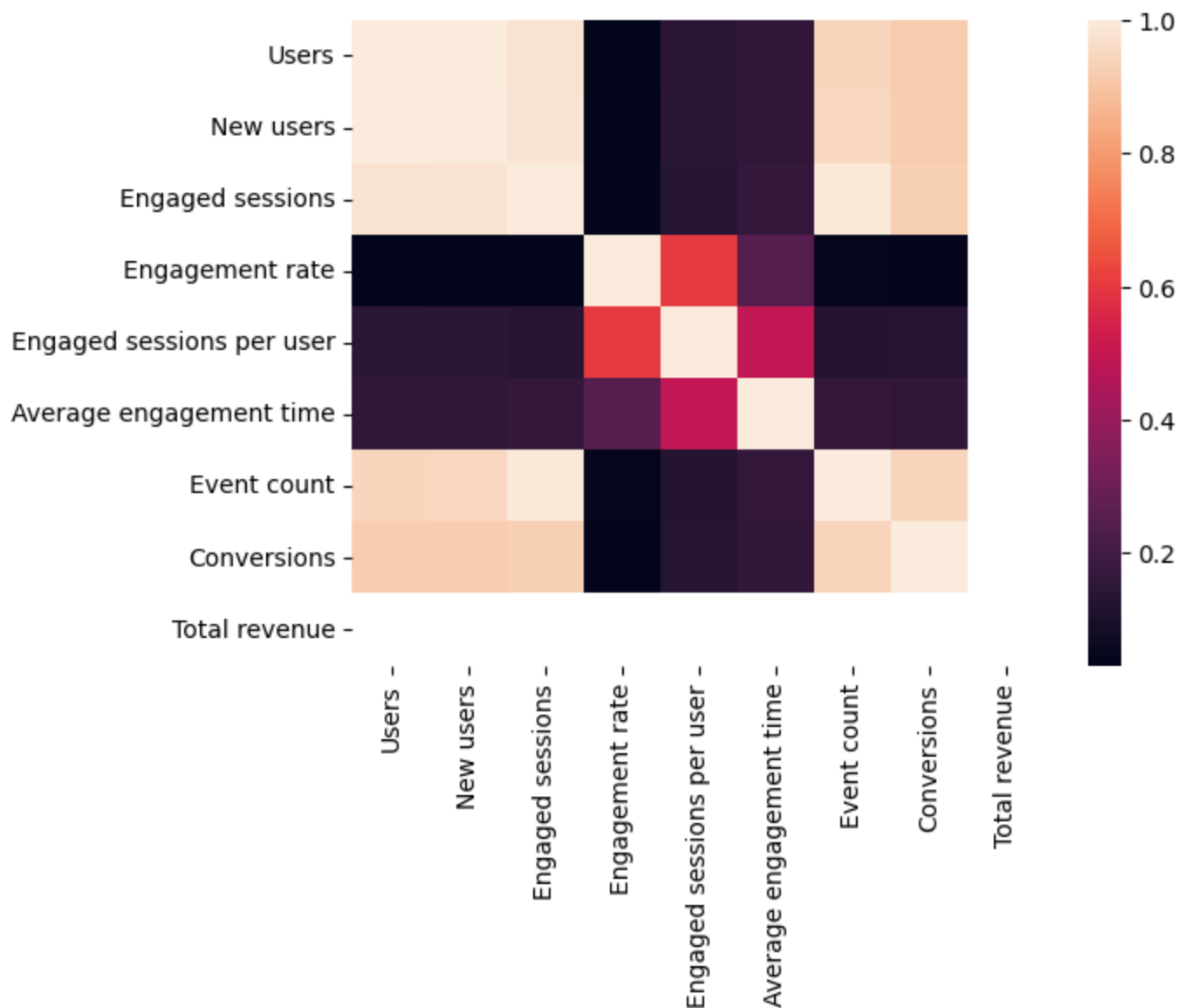
	count	mean	std	min	25%	50%	75%	max
Users	574.0	46.203833	290.091708	0.0	1.00	2.000000	7.750000	6097.0
New users	574.0	39.846690	267.115856	0.0	1.00	1.000000	5.000000	5685.0
Engaged sessions	574.0	71.613240	656.033770	0.0	1.00	2.000000	7.000000	15013.0
Engagement rate	574.0	0.489662	0.341660	0.0	0.25	0.497547	0.724026	1.0
Engaged sessions per user	574.0	0.884915	0.750856	0.0	0.50	1.000000	1.000000	7.0
Average engagement time	574.0	94.093768	202.850678	0.0	7.00	27.500000	98.402298	2721.0
Event count	574.0	2297.088850	26179.153773	1.0	10.00	35.000000	162.750000	607200.0
Conversions	574.0	337.829268	3074.301547	0.0	2.00	5.000000	23.000000	62939.0
Total revenue	574.0	0.000000	0.000000	0.0	0.00	0.000000	0.000000	0.0

In [12]: city\_wise.corr()

	Users	New users	Engaged sessions	Engagement rate	Engaged sessions per user	Average engagement time	Event count	Conversions	Total revenue
Users	1.000000	0.998784	0.975866	0.031740	0.141313	0.155377	0.946402	0.917091	NaN
New users	0.998784	1.000000	0.979186	0.031042	0.137042	0.152543	0.950213	0.917985	NaN
Engaged sessions	0.975866	0.979186	1.000000	0.037906	0.131520	0.159986	0.991068	0.925880	NaN
Engagement rate	0.031740	0.031042	0.037906	1.000000	0.599869	0.249160	0.040164	0.036949	NaN
Engaged sessions per user	0.141313	0.137042	0.131520	0.599869	1.000000	0.492213	0.122799	0.127281	NaN
Average engagement time	0.155377	0.152543	0.159986	0.249160	0.492213	1.000000	0.161539	0.152925	NaN
Event count	0.946402	0.950213	0.991068	0.040164	0.122799	0.161539	1.000000	0.942964	NaN
Conversions	0.917091	0.917985	0.925880	0.036949	0.127281	0.152925	0.942964	1.000000	NaN
Total revenue	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

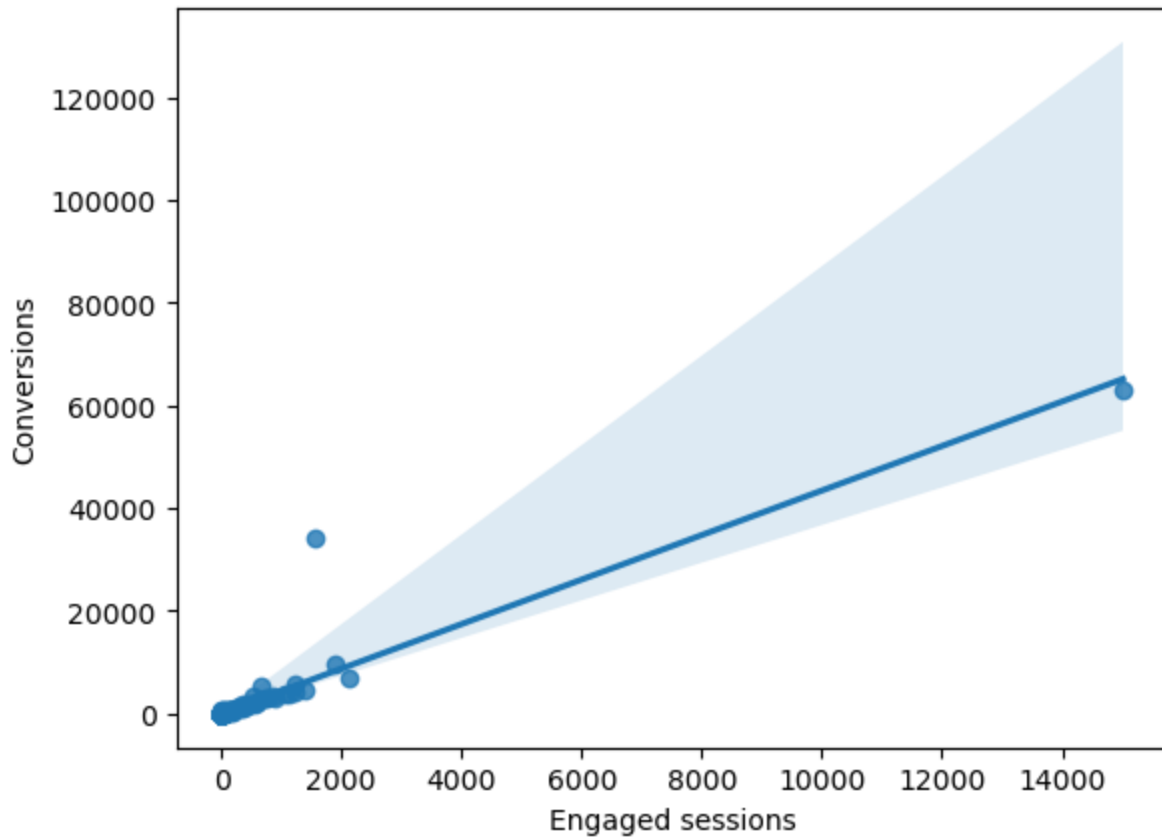
In [13]: one\_hotmen/city\_wise.corr()

Out[13]: <AxesSubplot:>



```
In [14]: sns.regplot(x = "Engaged sessions" , y = "Conversions" , data = city_wise)
```

```
Out[14]: <AxesSubplot:xlabel='Engaged sessions', ylabel='Conversions'>
```



## Observations

### analysis and findings

- 1) Total number of cities, where the company is running are 574.
- 2) No null values present in the given data.
- 3) The features 'Users','New users','Engaged sessions' and 'Event count' have a very high co-relation and these features may be use to build a predictive model for conversions.
- 4) Bangalore is the highest city in terms of conversions followed by Patna, Hyderabad and Indore.