

In [2]:

```
import pandas as pd
import numpy as np
```

In [5]:

```
pd.set_option('display.max_columns',80)
df1 = pd.read_csv('covid_19_india.csv')
```

In [6]:

df1

Out[6]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirm
0	1	30/01/20	6:00 PM	Kerala	1	0	0	0	
1	2	31/01/20	6:00 PM	Kerala	1	0	0	0	
2	3	01/02/20	6:00 PM	Kerala	2	0	0	0	
3	4	02/02/20	6:00 PM	Kerala	3	0	0	0	
4	5	03/02/20	6:00 PM	Kerala	3	0	0	0	
...	...	...	...	...	...	...	...	...	
8481	8482	16/11/20	8:00 AM	Telengana	-	-	242084	1407	257876
8482	8483	16/11/20	8:00 AM	Tripura	-	-	30555	364	31111
8483	8484	16/11/20	8:00 AM	Uttarakhand	-	-	62543	1107	68654
8484	8485	16/11/20	8:00 AM	Uttar Pradesh	-	-	480965	7372	511111
8485	8486	16/11/20	8:00 AM	West Bengal	-	-	394576	7661	431111

8486 rows x 9 columns



In [7]:

```
df2=df1.drop(['Time','ConfirmedIndianNational','ConfirmedForeignNational'],axis=1)
df2
```

Out[7]:

	Sno	Date	State/UnionTerritory	Cured	Deaths	Confirmed
0	1	30/01/20	Kerala	0	0	1
1	2	31/01/20	Kerala	0	0	1
2	3	01/02/20	Kerala	0	0	2
3	4	02/02/20	Kerala	0	0	3
4	5	03/02/20	Kerala	0	0	3
...	...	...	...	...	...	...
8481	8482	16/11/20	Telengana	242084	1407	257876

8482	8483	16/11/20	State/UnionTerritory	Cases	Deaths	Confirmed
8483	8484	16/11/20	Uttarakhand	62543	1107	68215
8484	8485	16/11/20	Uttar Pradesh	480965	7372	511304
8485	8486	16/11/20	West Bengal	394576	7661	431551

8486 rows x 6 columns

In [8]:

```
df2=df2.rename(columns={'State/UnionTerritory':'State'})
```

In [9]:

```
grouped = df2.groupby(df2.State)
```

In [10]:

```
uniqueValues = (df2['State']).unique()
```

In [11]:

```
uniqueValues
```

Out[11]:

```
array(['Kerala', 'Telengana', 'Delhi', 'Rajasthan', 'Uttar Pradesh',  
      'Haryana', 'Ladakh', 'Tamil Nadu', 'Karnataka', 'Maharashtra',  
      'Punjab', 'Jammu and Kashmir', 'Andhra Pradesh', 'Uttarakhand',  
      'Odisha', 'Puducherry', 'West Bengal', 'Chhattisgarh',  
      'Chandigarh', 'Gujarat', 'Himachal Pradesh', 'Madhya Pradesh',  
      'Bihar', 'Manipur', 'Mizoram', 'Andaman and Nicobar Islands',  
      'Goa', 'Unassigned', 'Assam', 'Jharkhand', 'Arunachal Pradesh',  
      'Tripura', 'Nagaland', 'Meghalaya', 'Dadar Nagar Haveli',  
      'Cases being reassigned to states', 'Sikkim', 'Daman & Diu',  
      'Dadra and Nagar Haveli and Daman and Diu', 'Telangana',  
      'Telangana***', 'Telengana***', 'Maharashtra***', 'Chandigarh***',  
      'Punjab***'], dtype=object)
```

In [12]:

```
df_State=pd.DataFrame(uniqueValues)  
df_State
```

Out[12]:

	0
0	Kerala
1	Telengana
2	Delhi
3	Rajasthan
4	Uttar Pradesh
5	Haryana
6	Ladakh
7	Tamil Nadu
8	Karnataka
9	Maharashtra
10	Punjab
11	Jammu and Kashmir
12	Andhra Pradesh
13	Uttarakhand

14	Odisha
15	Puducherry
16	West Bengal
17	Chhattisgarh
18	Chandigarh
19	Gujarat
20	Himachal Pradesh
21	Madhya Pradesh
22	Bihar
23	Manipur
24	Mizoram
25	Andaman and Nicobar Islands
26	Goa
27	Unassigned
28	Assam
29	Jharkhand
30	Arunachal Pradesh
31	Tripura
32	Nagaland
33	Meghalaya
34	Dadar Nagar Haveli
35	Cases being reassigned to states
36	Sikkim
37	Daman & Diu
38	Dadra and Nagar Haveli and Daman and Diu
39	Telangana
40	Telangana***
41	Telangana***
42	Maharashtra***
43	Chandigarh***
44	Punjab***

In [14]:

```
df_S1=df_State.rename(columns={0:'State'})
df_S1
```

Out[14]:

	State
0	Kerala
1	Telangana
2	Delhi
3	Rajasthan
4	Uttar Pradesh
5	Haryana
6	Ladakh

7	Tamil Nadu
8	Karnataka
9	Maharashtra
10	Punjab
11	Jammu and Kashmir
12	Andhra Pradesh
13	Uttarakhand
14	Odisha
15	Puducherry
16	West Bengal
17	Chhattisgarh
18	Chandigarh
19	Gujarat
20	Himachal Pradesh
21	Madhya Pradesh
22	Bihar
23	Manipur
24	Mizoram
25	Andaman and Nicobar Islands
26	Goa
27	Unassigned
28	Assam
29	Jharkhand
30	Arunachal Pradesh
31	Tripura
32	Nagaland
33	Meghalaya
34	Dadar Nagar Haveli
35	Cases being reassigned to states
36	Sikkim
37	Daman & Diu
38	Dadra and Nagar Haveli and Daman and Diu
39	Telangana
40	Telangana***
41	Telangana***
42	Maharashtra***
43	Chandigarh***
44	Punjab***

In [16]:

```

updat_df=df_S1.drop([39,40,41,42,43,44])
updat_df

```

Out[16]:

	State
0	...

0	Kerala
1	Telengana
2	Delhi
3	Rajasthan
4	Uttar Pradesh
5	Haryana
6	Ladakh
7	Tamil Nadu
8	Karnataka
9	Maharashtra
10	Punjab
11	Jammu and Kashmir
12	Andhra Pradesh
13	Uttarakhand
14	Odisha
15	Puducherry
16	West Bengal
17	Chhattisgarh
18	Chandigarh
19	Gujarat
20	Himachal Pradesh
21	Madhya Pradesh
22	Bihar
23	Manipur
24	Mizoram
25	Andaman and Nicobar Islands
26	Goa
27	Unassigned
28	Assam
29	Jharkhand
30	Arunachal Pradesh
31	Tripura
32	Nagaland
33	Meghalaya
34	Dadar Nagar Haveli
35	Cases being reassigned to states
36	Sikkim
37	Daman & Diu
38	Dadra and Nagar Haveli and Daman and Diu

In [14]:

```
df_S=df_State[:39]
```

In [17]:

```
l1=[]
l2=[]
```

```
l3=[]
for i in uniqueValues:
    g = grouped.get_group(i)
    l1.append(g)
```

In [18]:

```
Telangana=grouped.get_group('Telengana')
Telangana
```

Out[18]:

	Sno	Date	State	Cured	Deaths	Confirmed	
	32	33	02/03/20	Telengana	0	0	1
	35	36	03/03/20	Telengana	0	0	1
	43	44	04/03/20	Telengana	0	0	1
	49	50	05/03/20	Telengana	0	0	1
	56	57	06/03/20	Telengana	0	0	1
	...	...	...	...	...	...	...
	8341	8342	12/11/20	Telengana	235950	1393	254666
	8376	8377	13/11/20	Telengana	237172	1397	255663
	8411	8412	14/11/20	Telengana	238908	1401	256713
	8446	8447	15/11/20	Telengana	240545	1404	257374
	8481	8482	16/11/20	Telengana	242084	1407	257876

213 rows × 6 columns

In [19]:

```
Kerala=grouped.get_group('Kerala')
Delhi=grouped.get_group('Delhi')
Rajasthan=grouped.get_group('Rajasthan')
UttarPradesh=grouped.get_group('Uttar Pradesh')
Haryana=grouped.get_group('Haryana')
Ladakh=grouped.get_group('Ladakh')
TamilNadu=grouped.get_group('Tamil Nadu')
Karnataka=grouped.get_group('Karnataka')
Maharashtra=grouped.get_group('Maharashtra')
Punjab=grouped.get_group('Punjab')
JammuandKashmir=grouped.get_group('Jammu and Kashmir')
```

In [20]:

```
AndhraPradesh=grouped.get_group('Andhra Pradesh')
Uttarakhand=grouped.get_group('Uttarakhand')
Odisha=grouped.get_group('Odisha')
Puducherry=grouped.get_group('Puducherry')
WestBengal=grouped.get_group('West Bengal')
Chhattisgarh=grouped.get_group('Chhattisgarh')
Chandigarh=grouped.get_group('Chandigarh')
Gujarat=grouped.get_group('Gujarat')
HimachalPradesh=grouped.get_group('Himachal Pradesh')
MadhyaPradesh=grouped.get_group('Madhya Pradesh')
Bihar=grouped.get_group('Bihar')
```

In [21]:

```
Manipur=grouped.get_group('Manipur')
Mizoram=grouped.get_group('Mizoram')
AndamanandNicobarIslands=grouped.get_group('Andaman and Nicobar Islands')
Goa=grouped.get_group('Goa')
Assam=grouped.get_group('Assam')
Jharkhand=grouped.get_group('Jharkhand')
```

```
ArunachalPradesh=grouped.get_group('Arunachal Pradesh')
Tripura=grouped.get_group('Tripura')
Nagaland=grouped.get_group('Nagaland')
Meghalaya=grouped.get_group('Meghalaya')
Sikkim=grouped.get_group('Sikkim')
```

In [22]:

```
Telangana_1=grouped.get_group('Telangana***')
```

In [25]:

```
Delhi_cured=Delhi['Cured'].tolist()
Delhi_confirmed=Delhi['Confirmed'].tolist()
Delhi_deaths=Delhi['Deaths'].tolist()
```

In [40]:

```
from scipy.stats import f_oneway
stat, p = f_oneway(Delhi_cured, Delhi_confirmed, Delhi_deaths)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=110.839, p=0.000
Probably different distributions
```

In [30]:

```
Maharashtra_cured=Maharashtra['Cured'].tolist()
Maharashtra_confirmed=Maharashtra['Confirmed'].tolist()
Maharashtra_deaths=Maharashtra['Deaths'].tolist()
```

In [42]:

```
stat, p = f_oneway(Maharashtra_cured, Maharashtra_confirmed, Maharashtra_deaths)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=89.194, p=0.000
Probably different distributions
```

In [29]:

```
Kerala_cured=Kerala['Cured'].tolist()
Kerala_confirmed=Kerala['Confirmed'].tolist()
Kerala_deaths=Kerala['Deaths'].tolist()
```

In [43]:

```
stat, p = f_oneway(Kerala_cured, Kerala_confirmed, Kerala_deaths)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=45.663, p=0.000
Probably different distributions
```

In [44]:

```
stat, p = f_oneway(Kerala_cured, Maharashtra_cured, Delhi_cured)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
```

```
else:
    print('Probably different distributions')
```

stat=106.496, p=0.000  
Probably different distributions

In [45]:

```
from scipy.stats import ttest_rel
stat, p = ttest_rel(Delhi_cured, Delhi_deaths)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

stat=14.590, p=0.000  
Probably different distributions

In [46]:

```
stat, p = ttest_rel(Delhi_confirmed, Delhi_deaths)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

stat=15.467, p=0.000  
Probably different distributions

In [47]:

```
stat, p = ttest_rel(Delhi_cured, Delhi_confirmed)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

stat=-20.749, p=0.000  
Probably different distributions

In [35]:

```
from scipy.stats import pearsonr
stat, p = pearsonr(Delhi_cured, Delhi_confirmed)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

stat=0.998, p=0.000  
Probably dependent

In [49]:

```
from scipy.stats import normaltest
stat, p = normaltest(Delhi_deaths)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably Gaussian')
else:
    print('Probably not Gaussian')
```

stat=1489.452, p=0.000  
Probably not Gaussian

In [28]:

```
from scipy.stats import f_oneway
```



```
from scipy import stats
```

In [42]:

```
F, p = stats.f_oneway(Kerala_deaths, Maharashtra_deaths, Delhi_deaths)
print("p-value for significance is: ", p)
if p<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

p-value for significance is: 1.7208038714642608e-77  
reject null hypothesis

In [43]:

```
F, p = stats.f_oneway(Kerala_cured, Maharashtra_cured, Delhi_cured)
print("p-value for significance is: ", p)
if p<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

p-value for significance is: 9.727791295363627e-42  
reject null hypothesis

In [44]:

```
F, p = stats.f_oneway(Kerala_confirmed, Maharashtra_confirmed, Delhi_confirmed)
print("p-value for significance is: ", p)
if p<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

p-value for significance is: 1.8860363747946795e-50  
reject null hypothesis

In [37]:

```
pip install statsmodels
```

Collecting statsmodels

Downloading statsmodels-0.12.1-cp36-cp36m-manylinux1\_x86\_64.whl (9.5 MB)

|██| 9.5 MB 4.7 MB/s eta 0:00:01

Requirement already satisfied: numpy>=1.15 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from statsmodels) (1.19.4)

Collecting patsy>=0.5

Downloading patsy-0.5.1-py2.py3-none-any.whl (231 kB)

|██| 231 kB 51.6 MB/s eta 0:00:01

Requirement already satisfied: pandas>=0.21 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from statsmodels) (1.1.4)

Requirement already satisfied: scipy>=1.1 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from statsmodels) (1.5.3)

Requirement already satisfied: six in /srv/conda/envs/notebook/lib/python3.6/site-packages (from patsy>=0.5->statsmodels) (1.15.0)

Requirement already satisfied: python-dateutil>=2.7.3 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from pandas>=0.21->statsmodels) (2.8.1)

Requirement already satisfied: pytz>=2017.2 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from pandas>=0.21->statsmodels) (2020.4)

Installing collected packages: patsy, statsmodels

Successfully installed patsy-0.5.1 statsmodels-0.12.1

Note: you may need to restart the kernel to use updated packages.

In [26]:

```
from statsmodels.stats import weightstats as stests
```

In [31]:

```
ztest ,pval1= stests.ztest(Delhi_confirmed, x2=Maharashtra_confirmed, value=0,alternati
```

```
ve='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

7.466727334332615e-27  
reject null hypothesis

In [32]:

```
ztest ,pval1 = stats.ztest(Delhi_confirmed, x2=Kerala_confirmed, value=0,alternative='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

1.1103686154825099e-05  
reject null hypothesis

In [33]:

```
ztest ,pval1 = stats.ztest(Maharashtra_confirmed, x2=Kerala_confirmed, value=0,alternative='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

5.141444147202287e-37  
reject null hypothesis

In [34]:

```
ztest ,pval1 = stats.ztest(Delhi_deaths, x2=Kerala_deaths, value=0,alternative='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

4.6994084033363774e-67  
reject null hypothesis

In [37]:

```
ztest ,pval1 = stats.ztest(Delhi_deaths, x2=Maharashtra_deaths, value=0,alternative='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

1.7466227224765882e-38  
reject null hypothesis

In [38]:

```
ztest ,pval1 = stats.ztest(Maharashtra_deaths, x2=Kerala_deaths, value=0,alternative='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

7.610510121878501e-62  
reject null hypothesis

In [39]:

```
ztest ,pval1 = stests.ztest(Delhi_cured, x2=Kerala_cured, value=0,alternative='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

5.1059581124681983e-08  
reject null hypothesis

In [40]:

```
ztest ,pval1 = stests.ztest(Maharashtra_cured, x2=Kerala_cured, value=0,alternative='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

5.23820797942999e-31  
reject null hypothesis

In [41]:

```
ztest ,pval1 = stests.ztest(Delhi_cured, x2=Maharashtra_cured, value=0,alternative='two-sided')
print(float(pval1))
if pval1<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

1.3158632604991941e-20  
reject null hypothesis

In [ ]: