

A PROJECT REPORT ON
COVID-19 STATS
FOR
AISSCE 2021 EXAMINATION
**[AS A PART OF INFORMATICS
PRACTISES COURSE (065)]**

SUBMITTED BY:

---ADITYA MEENA---

SEAT NO:-

UNDER THE GUIDANCE OF
MRS. JULIET BELLY



Central Railway School & Jr. College
Murbad Road, Kalyan(W), Thane Dist, Maharashtra-421301
(Affiliated to CBSE, New Delhi, Affiliation No 1180013)



CERTIFICATE

This is to certify that the project entitled COVID-19 STATS is a bonafide work done by Aditya Meena of class XII Science, session 2020–2021 in partial fulfillment of CBSE's AISSCE Examination 2021 and has been carried out under my direct supervision and guidance. He has taken proper care and shown utmost sincerity in completion of this project. I certify this project is up to my expectation and as per the guidelines issued by the CBSE.

Internal Examiner

External Examiner

Principal

School Stamp

ACKNOWLEDGEMENT

I WOULD LIKE TO EXPRESS MY SPECIAL THANKS OF GRATITUDE TO OUR PRINCIPAL MR. JACOB THOMAS AS WELL AS MY TEACHER MRS. JULIET BELLY WHO GAVE ME THE GOLDEN OPPORTUNITY TO DO THIS WONDERFUL PROJECT ON THE TOPIC COVID-19 STATS WHICH ALSO HELPED ME IN DOING A LOT OF RESEARCH AND I CAME TO KNOW ABOUT SO MANY NEW THINGS. I AM REALLY THANKFUL TO THEM.

SECONDLY I WOULD ALSO LIKE TO THANK MY PARENTS AND FRIENDS WHO HELPED ME A LOT IN FINALIZING THIS PROJECT WITHIN A LIMITED TIME FRAME.

CONTENTS

1. INTRODUCTION:

- **AIM**
- **OBJECTIVE**

2. REQUIREMENT ANALYSIS:

- **TECHNOLOGIES USED**

3. CODING AND OUTPUT:

- **CODING**
- **OUTPUT**

4. CONCLUSION

5. BIBLIOGRAPHY

**AIM: TO MAKE AN APPLICATION OF
COVID-STATS USING PYTHON**

OBJECTIVE:

- **The main objective is to create a program on 'COVID-19 STATS'.**
- **To show the data in discrete and continuous graphical form which helps in reading data.**

TECHNOLOGY USED

Python (programming language)

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms,

including [structured](#) (particularly, [procedural](#)), [object-oriented](#) and [functional programming](#). Python is often described as a "batteries included" language due to its comprehensive [standard library](#).

Python was created in the late 1980s, and first released in 1991, by [Guido van Rossum](#) as a successor to the [ABC programming language](#). Python 2.0, released in 2000, introduced new features, such as [list comprehensions](#), and a garbage collection system with [reference counting](#), and was discontinued with version 2.7 in 2020. Python 3.0, released in 2008, was a major revision of the language that is not completely [backward-compatible](#) and much Python 2 code does not run unmodified on Python 3. With Python 2's [end-of-life](#) (and [pip](#) having dropped support in 2021), only Python 3.6.x and later are supported, with older versions still supporting e.g. [Windows 7](#) (and old installers not restricted to 64-bit Windows).

Python [interpreters](#) are supported for mainstream [operating systems](#) and available for a few more (and in the past supported many more). A global community of programmers develops and maintains [CPython](#), a [free and open-source reference implementation](#). A non-profit organization, the [Python Software Foundation](#), manages and directs resources for Python and CPython development.

As of February 2021, Python ranks third in [TIOBE](#)'s index of most popular programming languages, behind [C](#) and [Java](#), having previously gained second place and their award for the most popularity gain for 2020.

PYTHON LIBRARIES USED

1.PANDAS:

In [computer programming](#), **pandas** is a [software library](#) written for the [Python programming language](#) for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and [time series](#). It is [free software](#) released under the [three-clause BSD license](#). The name is derived from the term "[panel data](#)", an [econometrics](#) term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. [Wes McKinney](#) started building what would become pandas at [AQR Capital](#) while he was a researcher there from 2007 to 2010.

2.MATPLOTLIB:

Matplotlib is a [plotting library](#) for the [Python](#) programming language and its numerical mathematics extension [NumPy](#). It provides an [object-oriented API](#) for embedding plots into applications using general-purpose [GUI toolkits](#) like [Tkinter](#), [wxPython](#), [Qt](#), or [GTK+](#). There is also a [procedural](#) "pylab" interface based on a [state machine](#) (like [OpenGL](#)), designed to closely resemble that of [MATLAB](#), though its use is discouraged. [SciPy](#) makes use of Matplotlib.

Matplotlib was originally written by [John D. Hunter](#). Since then it has an active development community and is

distributed under a [BSD-style license](#). Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell.

3.CSV(Comma Separated Values):

CSV file – It is called Comma Separated Values file which allows data to be saved in a tabular format. A simple file such as spreadsheet or database. Files in the csv format can be exported and imported from program that store data in table such as MS excel or open office.

CODING AND CORRESPONDING RESULT

MAIN MENU:

```
~~~~~*****~~~~~
~~~~~WELCOME DEAR USER~~~~~
~~~~~*****~~~~~
COMPREHEND THE DATA IN DIFFERENT WAYS
1.TO READ THE DATA IN TABULAR FORM.
2.RETRIVE DATA IN LINE GRAPH FORM.
3.RETRIVE DATA IN BAR GRAPH FORM.
4.RETRIVE DATA FOR INDIA IN DIFFERENT WAYS.
5.READING DATA IN CONDITIONAL WAYS.
6.EXIT
~~~~~*****~~~~~
~~~~~*****~~~~~
~~~~~NOTE:YOU WILL GET THE DATA UPTO 1-JANUARY 2021~~~~~
Enter an option for which you are looking:|
```

IF OPTION CHOSEN IS 1:

READ THE CSV:

```
print("Whole data")
dfl=pd.read_csv("C:\\Users\\Aditya meena\\Desktop\\corona project\\ORIGINAL.csv")
print(dfl)
```

Whole data

	State	Confirmed	Recovered	Deaths	Tested
0	Maharashtra	1992683	1890323	50473	14265878
1	Karnataka	932432	912205	12175	16032142
2	Andhra Pradesh	886066	877212	7141	13214568
3	Kerala	851195	779097	3481	8874963
4	Tamil Nadu	831323	813326	12272	1532064
5	Delhi	632590	619501	10754	9926544
6	Uttar Pradesh	596904	579693	8580	26789521
7	West Bengal	565661	548705	10063	76745126
8	Odisha	333444	329801	1954	7446162
9	Rajasthan	315394	308010	2750	5627829
10	Chhattisgarh	293972	284412	3565	394516
11	Telangana	291872	286244	1579	7481236
12	Haryana	266428	261510	2989	4966522
13	Bihar	258883	253913	1460	21312561
14	Gujarat	256367	245907	4367	10205632
15	Madhya Pradesh	251882	241966	3756	5093612
16	Assam	216864	212975	1075	6264521
17	Punjab	170729	162762	5509	4251366
18	Jammu and Kashmir	123425	120392	1922	4271346
19	Jharkhand	117786	115542	1054	5051203

IF OPTION CHOSEN IS 2:

READ THE DATA IN LINE GRAPH FORM:

```
print("2.RETRIVE DATA IN LINE GRAPH FORM.")

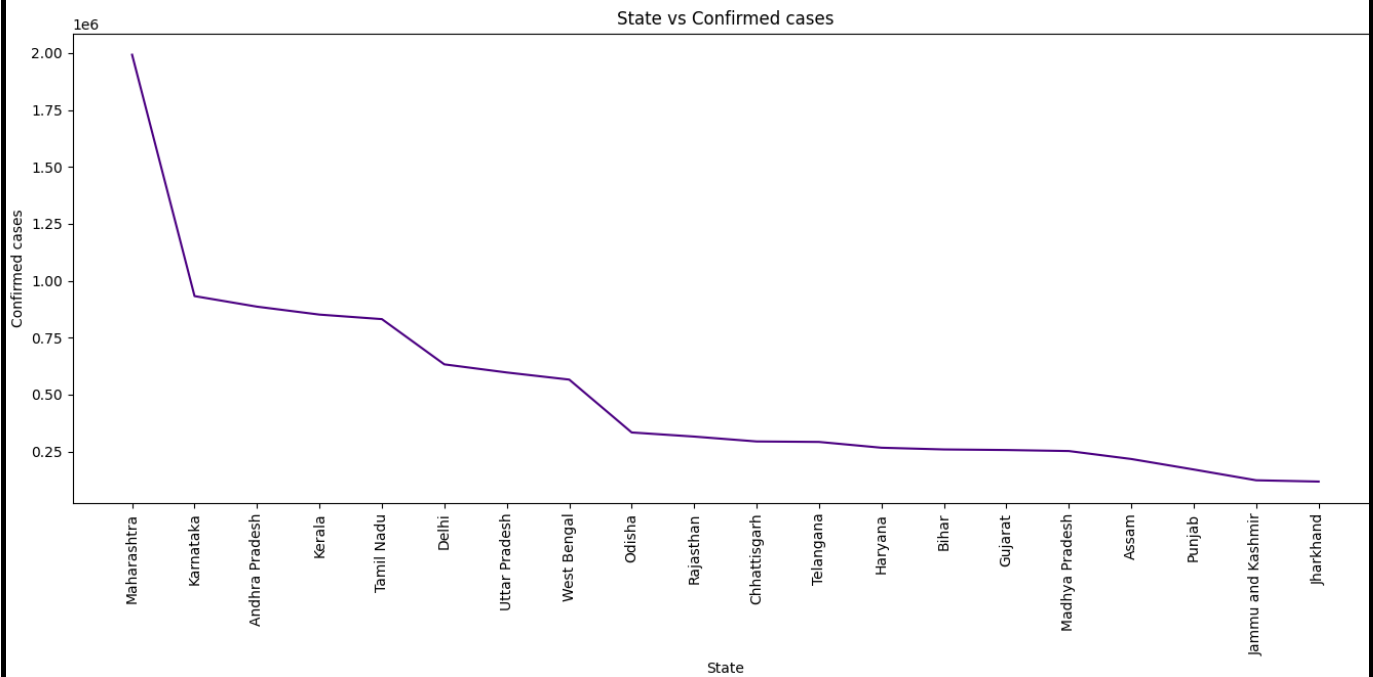
elif x==2:
    dfl=pd.read_csv("C:\\Users\\Aditya meena\\Desktop\\corona project\\ORIGINAL.csv")
    print("USING LINE GRAPHS")
    st=df1["State"]
    conf=df1["Confirmed"]
    rec=df1["Recovered"]
    dc=df1["Deaths"]
    ts=df1["Tested"]
    plt.xlabel("State")
    plt.xticks(rotation="vertical")
    print("SELECT AN OPTION FROM THE FOLLOWING:")
    print("1.RETRIVE DATA FOR STATE VS CONFIRMED CASES.")
    print("2.RETRIVE DATA FOR STATE VS RECOVERED CASES.")
    print("3.RETRIVE DATA FOR STATE VS DEATHS.")
    print("4.RETRIVE DATA FOR STATE VS PEOPLE TESTED.")
    print("5.DISPLAY ALL THE LINE GRAPH.")
    print("*****")
    xl=int(input("Enter an option for which you are looking:"))
```

IF SUB-OPTION CHOSEN IS 1:

STATE VS CONFIRMED CASES:

```
if xl==1:
    plt.ylabel("Confirmed cases")
    plt.plot(st,conf,color='indigo')
    plt.title("State vs Confirmed cases")
    plt.show()
```

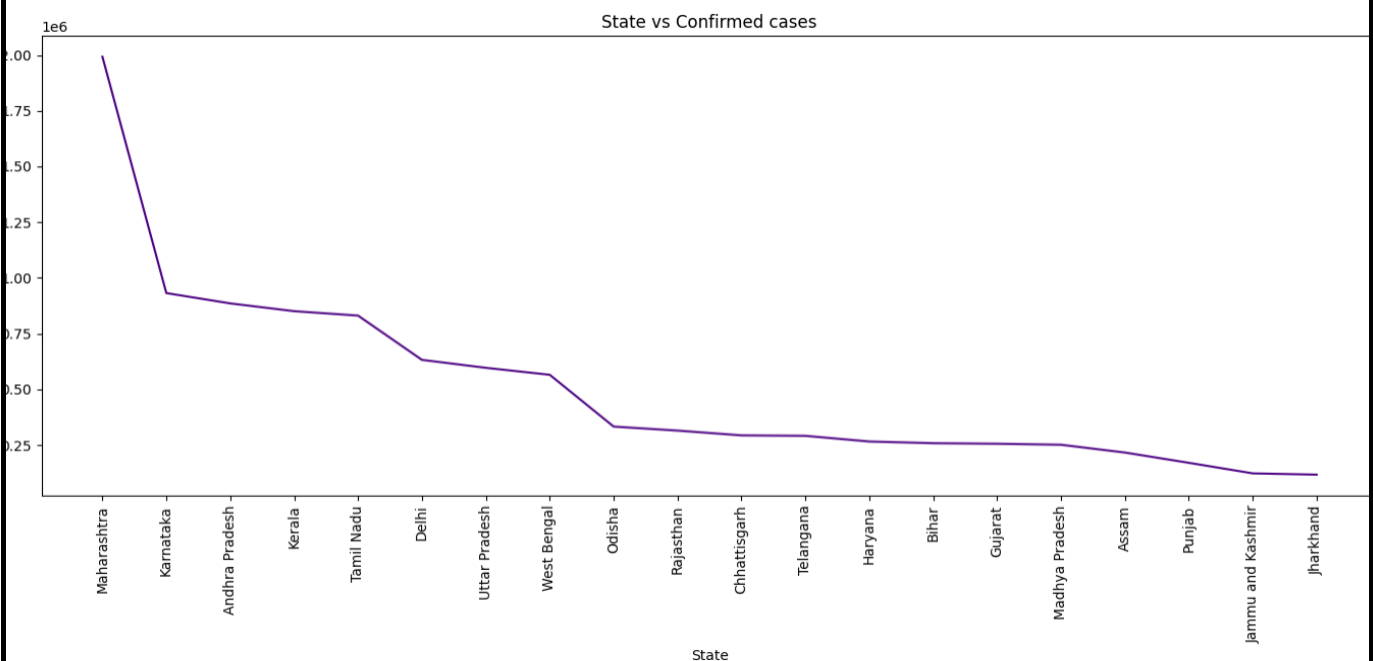
OUTPUT:



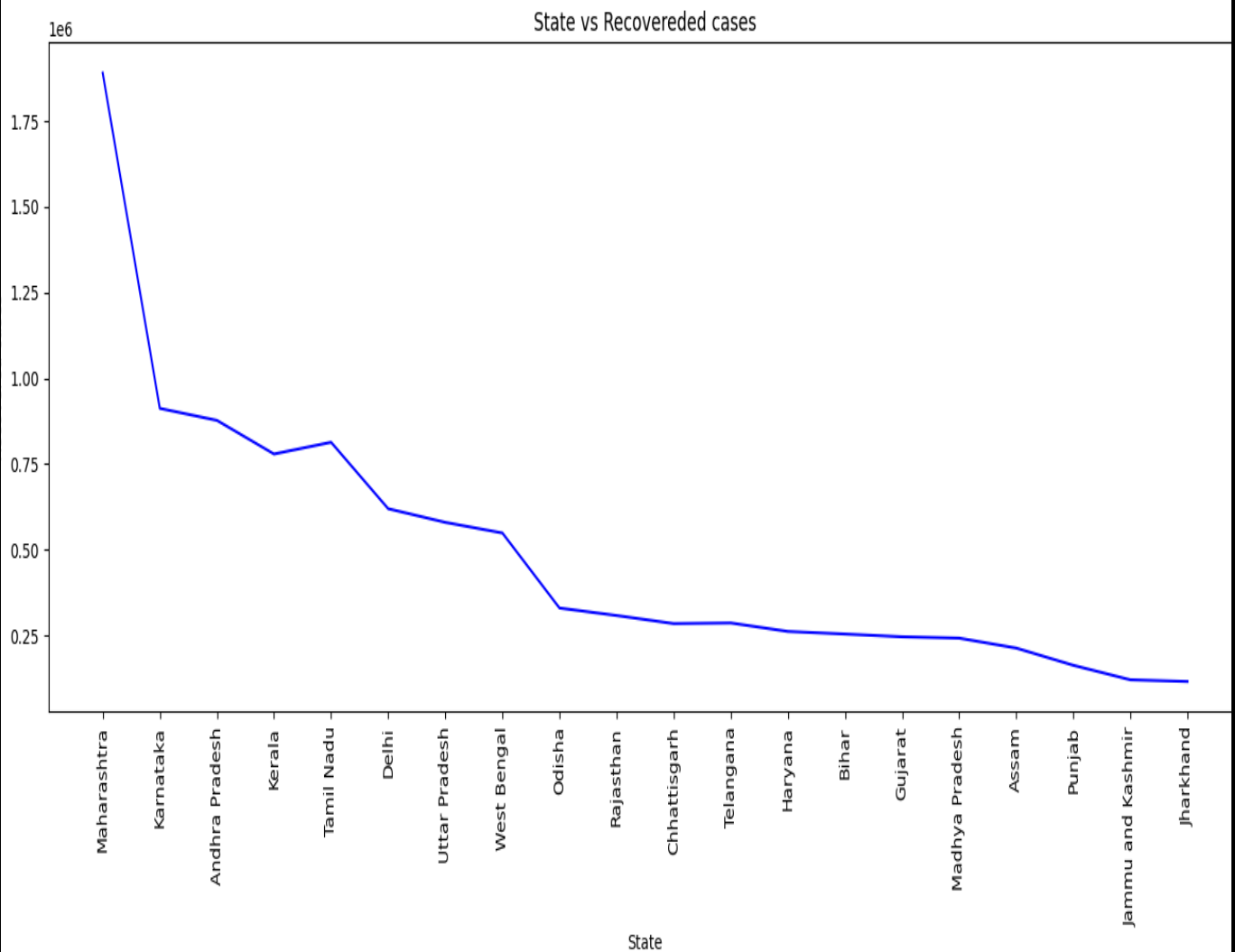
IF SUB-OPTION CHOSEN2 IS 2:

STATE VS RECOVERED CASES:

```
elif xl==2:  
    plt.ylabel("Recovered cases")  
    plt.title("State vs Recovered cases")  
    plt.plot(st,rec,color='blue')  
    plt.show()
```



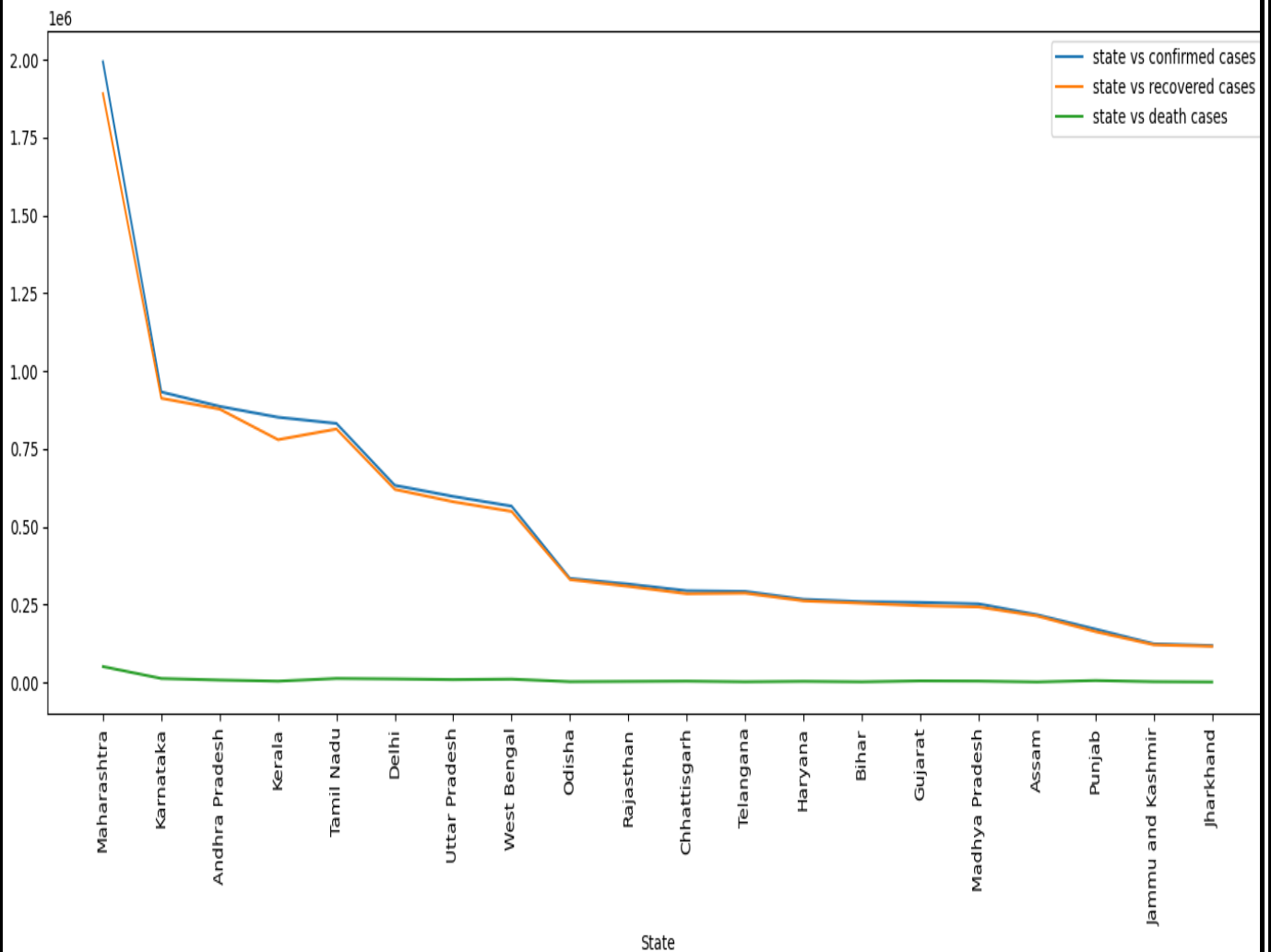
OUTPUT:



IF SUB-OPTION CHOSEN IS 5:

```
elif xl==5:
    plt.ylabel("Number of cases")
    plt.plot(st,conf,label='state vs confirmed cases')
    plt.plot(st,rec,label="state vs recovered cases")
    plt.plot(st,dc,label='state vs death cases')
    plt.legend()
    plt.show()
```

OUTPUT:



IF OPTION CHOSEN IS 3:

READ THE DATA IN BAR GRAPH FORM

```
print("3.RETRIVE DATA IN BAR GRAPH FORM.")
```

```

elif x==3:
    dfl=pd.read_csv("C:\\Users\\Aditya meena\\Desktop\\corona project\\ORIGINAL.csv")
    st=df1["State"]
    conf=df1["Confirmed"]
    rec=df1["Recovered"]
    dc=df1["Deaths"]
    ts=df1["Tested"]
    plt.xlabel("State")
    plt.xticks(rotation="vertical")
    print("1.RETRIVE THE DATA FOR STATE VS CONFIRMED CASES.")
    print("2.RETRIVE THE DATA FOR STATE VS RECOVERED CASES.")
    print("3.RETRIVE THE DATA FOR STATE VS DEATHS.")
    print("4.RETRIVE THE DATA FOR STATE VS PEOPLE TESTED.")
    print("5.RETRIVE THE DATA FOR STATE VS ALL CASES IN STACK BAR FORM.")
    print("USING BAR GRAPH:")
    print("~~~~~")

```

IF SUB-OPTION CHOSEN IS 1:

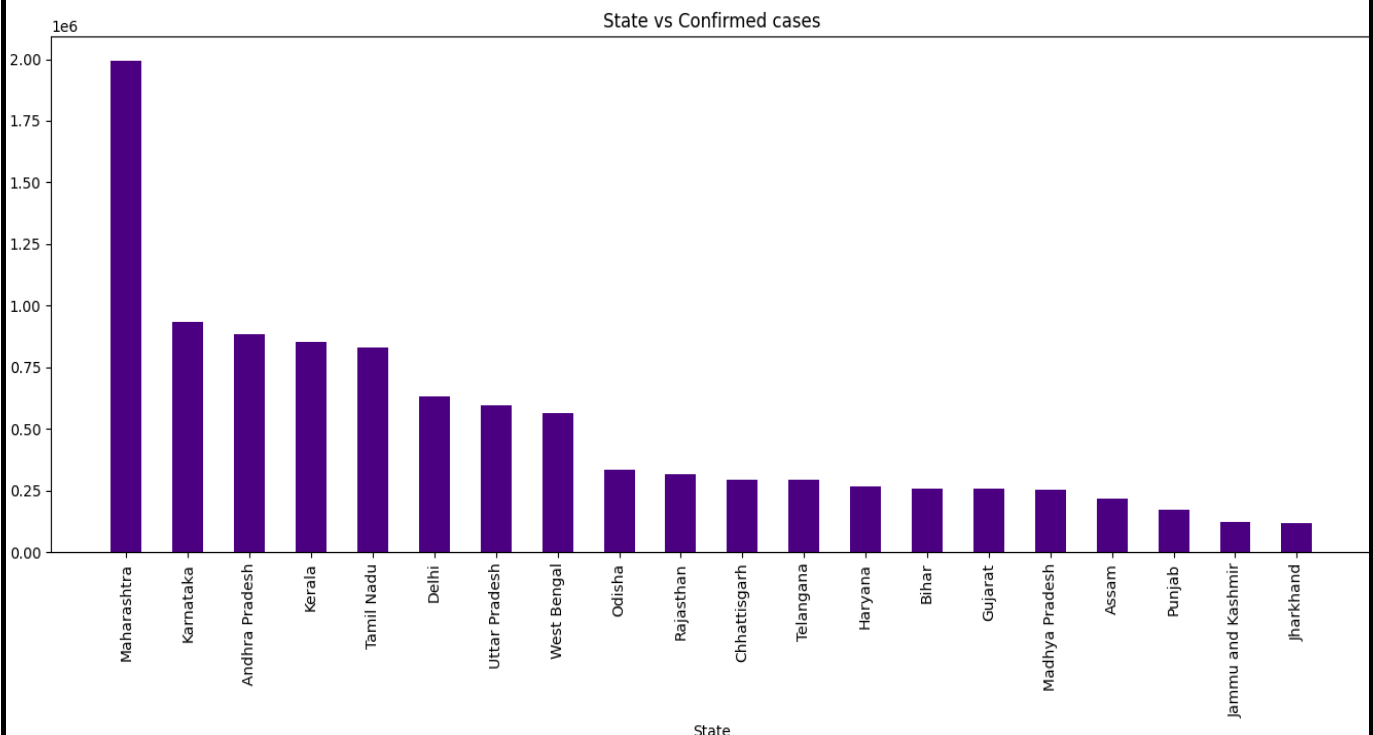
STATE VS CONFIRMED CASES

```

if x2==1:
    plt.ylabel("Confirmed cases")
    plt.title("State vs Confirmed cases")
    plt.bar(st,conf,color='indigo',width=.50)
    plt.show()

```

OUTPUT:

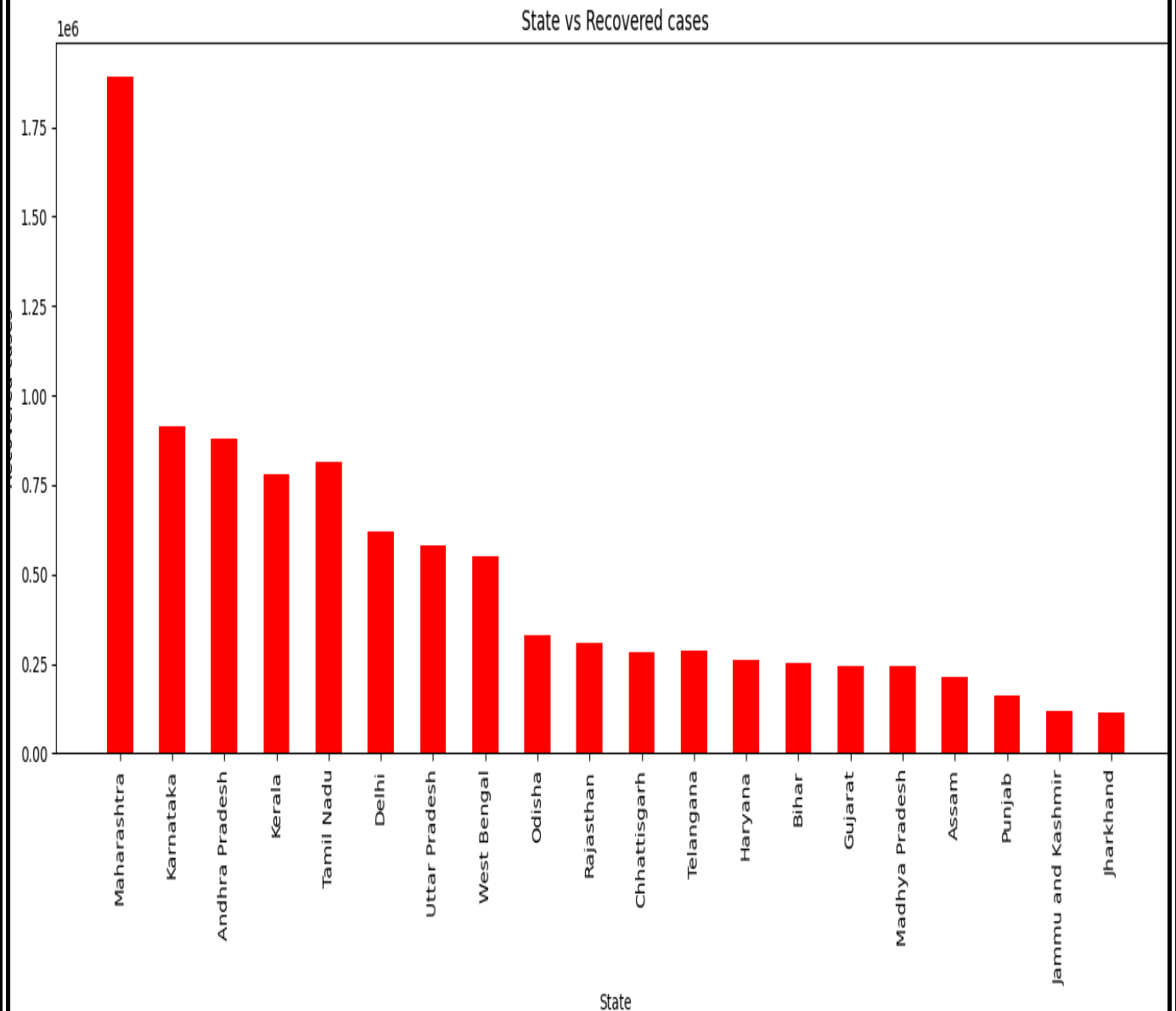


IF SUB-OPTION CHOSEN IS 2:

STATE VS RECOVERED CASES

```
elif x2==2:
    plt.ylabel("Recovered cases")
    plt.title("State vs Recovered cases")
    plt.bar(st,rec,color='red',width=.50)
    plt.show()
```

OUTPUT:



IF OPTION CHOSEN IS 4:

```
print("4.RETRIVE DATA FOR INDIA IN DIFFERENT WAYS.")
```

READ THE DATA OF INDIA

```
print("1.READ THE CSV FILE")
print("2.READ THE GRAPH OF NUMBER OF CONFIRMED CASES IN INDIA.")
print("3.READ THE GRAPH OF NUMBER OF ACTIVE CASES IN INDIA.")
print("4.READ THE GRAPH OF NUMBER OF RECOVERED CASES IN INDIA.")
print("5.READ THE GRAPH OF NUMBER OF DEATHS CASES IN INDIA.")
print("6.READ THE GRAPH OF NUMBER OF PEOPLE TESTED IN INDIA.")
print("7.DISPLAY ALL THE LINE GRAPH.")
print("~~~~~*****~")
xx=int(input('Enter an option for which you are looking:'))
```

IF SUB-OPTION CHOSEN IS 1:

TO READ CSV

```
if xx==1:
    df2=pd.read_csv("C:\\Users\\Aditya meena\\Desktop\\corona project\\activemh.csv")
    print(df2)
```

	Date	Confirmed cases	Active cases	Recovered cases	Deaths	Tested
0	01-Feb	2	2	0	0	0
1	01-Mar	5	2	3	0	0
2	01-Apr	2059	1829	169	69	51151
3	01-May	37263	26007	10021	1231	976363
4	01-Jun	198372	97009	95744	5606	3966075
5	01-Jul	605221	227401	359905	17848	9056173
6	01-Aug	1752171	567417	1146917	37410	19821831
7	01-Sep	3766108	799530	2899528	66462	44337201
8	01-Oct	6392049	9425887	5348746	99807	76717728
9	01-Nov	8229322	562329	7542905	122642	110743103
10	01-Dec	9499728	427528	8931803	138160	142445952
11	01-Jan	10306469	247427	9905570	149255	174899783

IF SUB-OPTION CHOSEN IS 2:

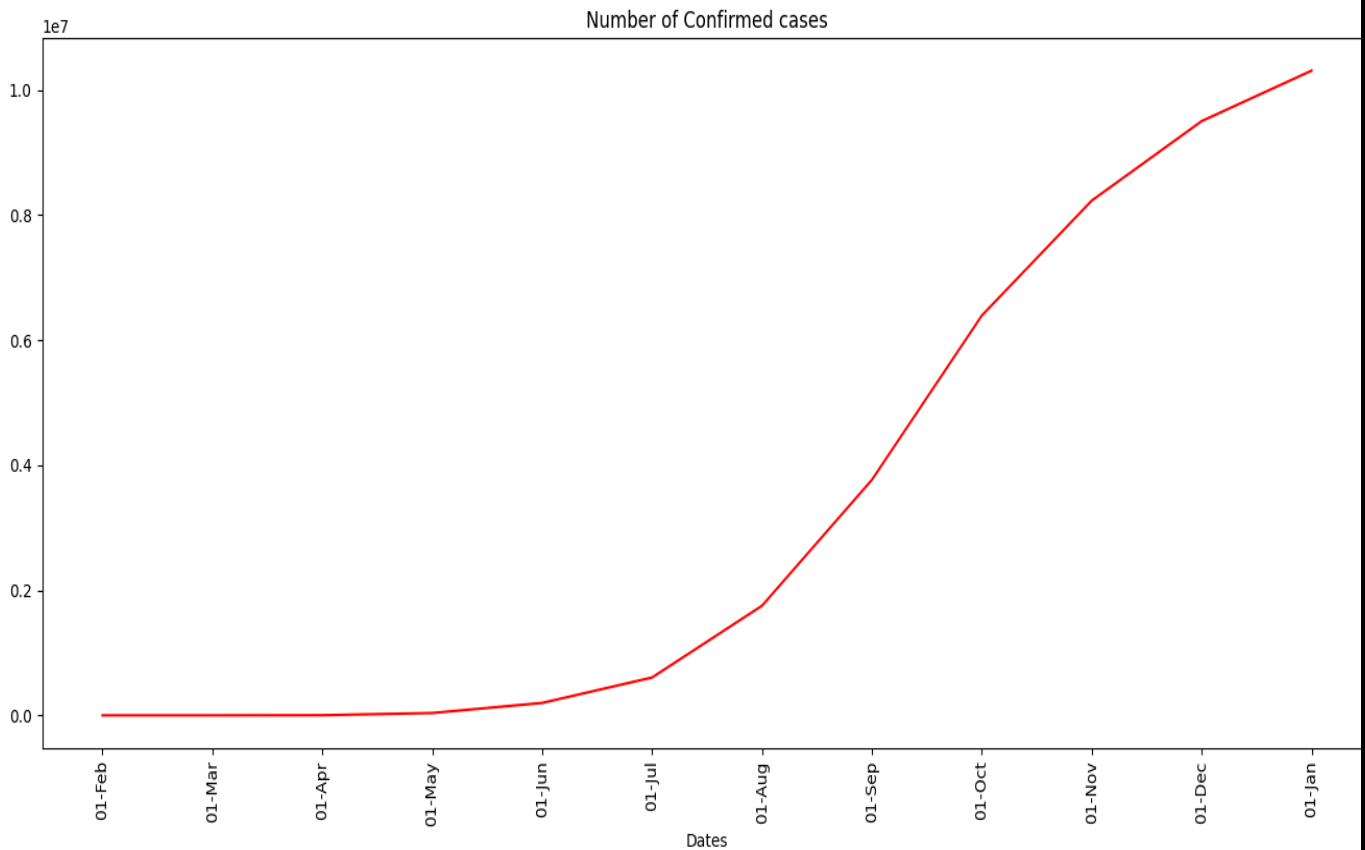
DATE VS CONFIRMED CASES


```

if xx==2:
    df2=pd.read_csv("C:\\Users\\Aditya meena\\Desktop\\corona project\\activemh.csv")
    stl=df2["Date"]
    confl=df2["Confirmed cases"]
    plt.xlabel("Dates")
    plt.ylabel("Confirmed cases")
    plt.xticks(rotation="vertical")
    plt.plot(stl,confl,color='red')
    plt.title("Number of Confirmed cases")
    plt.show()

```

OUTPUT:



IF OPTION CHOSEN IS 5:

```

print("5.READING DATA IN CONDITIONAL WAYS.")

```

RETRIEVE DATA IN CONDITIONAL WAYS

```
elif x==5:
    print("1.RETRIVE DATA FOR TOP VALUES")
    print("2.RETRIVE DATA FOR BOTTOM VALUES")
```

IF SUB-OPTION CHOSEN IS 1:

```
if op==1:
    print("1.STATE WITH MOST CONFIRMED CASES")
    print("2.STATE WITH MOST RECOVERED CASES")
    print("3.STATE WITH MOST DEATHS")
    print("4.STATE WITH MOST PEOPLE TESTED")
```

OUTPUT:

```
Enter an option for which you are looking:1
1.STATE WITH MOST CONFIRMED CASES
2.STATE WITH MOST RECOVERED CASES
3.STATE WITH MOST DEATHS
4.STATE WITH MOST PEOPLE TESTED
Enter an option for which you are looking:
```

IF SUB-OPTION CHOSEN IS 1:

```
if x==1:
    print("\n\nTHE STATE HAVING HIGHEST NUMBER OF CONFIRMED CASES:")
    print(dff.head(1))
elif x>1 and x<=20:
    print("\n\nThe top",x,"State having highest number of Confirmed cases")
    dff1=dff.head(x)
    print(dff1)
else:
    print("Enter a valid option")
```

THE STATE WITH HIGHEST NUMBER OF CONFIRMED CASES:

```
if x==1:
    print("\n\nTHE STATE HAVING HIGHEST NUMBER OF CONFIRMED CASES:")
    print(dff.head(1))
```

```
NOTE:MAXIMUM ROWS ARE 20
Enter the number of state u want from top:1
```

```
THE STATE HAVING HIGHEST NUMBER OF CONFIRMED CASES:
      Confirmed
State
Maharashtra    1992683
```

10 STATES WITH HIGHEST NUMBER OF CONFIRMED CASES:

```
elif x>1 and x<=20:  
    print("\n\nThe top",x,"State having highest number of Confirmed cases")  
    dff1=dff.head(x)  
    print(dff1)
```

OUTPUT:

Enter the number of state u want from top:10

The top 10 State having highest number of Confirmed cases

State	Confirmed
Maharashtra	1992683
Karnataka	932432
Andhra Pradesh	886066
Kerala	851195
Tamil Nadu	831323
Delhi	632590
Uttar Pradesh	596904
West Bengal	565661
Odisha	333444
Rajasthan	315394

IF SUB-OPTION IS 2:

```
print("1.STATE WITH LEAST CONFIRMED CASES")
print("2.STATE WITH LEAST RECOVERED CASES")
print("3.STATE WITH LEAST DEATHS")
print("4.STATE WITH LEAST PEOPLE TESTED")
```

OUTPUT:

```
1.STATE WITH LEAST CONFIRMED CASES
2.STATE WITH LEAST RECOVERED CASES
3.STATE WITH LEAST DEATHS
4.STATE WITH LEAST PEOPLE TESTED
```

IF SUB-OPTION CHOSEN IS 1:

```
dff=pd.read_csv('Book1.csv',index_col=0)
x=int(input("Enter the number of state u want from Bottom:"))
if x==1:
    print("\n\nTHE STATE HAVING LOWEST NUMBER OF CONFIRMED CASES:")
    print(dff.tail(1))
elif x>1 or x<=19:
    print("\n\nThe bottom",x,"State having lowest confirmed cases")
    dffl=dff.tail(x)
    print(dffl)
else:
    print("Enter a valid option")
```

THE STATE HAVING LOWEST NUMBER OF CONFIRMED CASES:

```
print("\n\nTHE STATE HAVING LOWEST NUMBER OF CONFIRMED CASES:")
print(dff.tail(1))
```

OUTPUT:

```
THE STATE HAVING LOWEST NUMBER OF CONFIRMED CASES:
      Confirmed
State
Jharkhand      117786
```

10 STATES WITH LEAST NUMBER OF CONFIRMED CASES:

```
print("1.STATE WITH LEAST CONFIRMED CASES")
print("2.STATE WITH LEAST RECOVERED CASES")
print("3.STATE WITH LEAST DEATHS")
print("4.STATE WITH LEAST PEOPLE TESTED")
opl=int(input("Enter an option for which you are looking:"))
if opl==1:
    dff=pd.read_csv('Book1.csv',index_col=0)
    x=int(input("Enter the number of state u want from Bottom:"))

    elif x>1 or x<=19:
        print("\n\nThe bottom",x,"State having lowest confirmed cases")
        dffl=dff.tail(x)
        print(dffl)
    else:
        print("Enter a valid option")
```

OUTPUT:

The bottom 10 State having lowest confirmed cases

State	Confirmed
-------	-----------

Chhattisgarh	293972
Telangana	291872
Haryana	266428
Bihar	258883
Gujarat	256367
Madhya Pradesh	251882
Assam	216864
Punjab	170729
Jammu and Kashmir	123425
Jharkhand	117786

1	State	Confirmed				
2	Maharashtra	1992683				
3	Karnataka	932432				
4	Andhra Pradesh	886066				
5	Kerala	851195				
6	Tamil Nadu	831323				
7	Delhi	632590				
8	Uttar Pradesh	596904				
9	West Bengal	565661				
10	Odisha	333444				
11	Rajasthan	315394				
12	Chhattisgarh	293972				
13	Telangana	291872				
14	Haryana	266428				
15	Bihar	258883				
16	Gujarat	256367				
17	Madhya Pradesh	251882				
18	Assam	216864				
19	Punjab	170729				
20	Jammu and Kashmir	123425				
21	Jharkhand	117786				

1	State	Deaths				
2	Maharashtra	50473				
3	Tamil Nadu	12272				
4	Karnataka	12175				
5	Delhi	10754				
6	West Bengal	10063				
7	Uttar Pradesh	8580				
8	Andhra Pradesh	7141				
9	Punjab	5509				
10	Gujarat	4367				
11	Madhya Pradesh	3756				
12	Chhattisgarh	3565				
13	Kerala	3481				
14	Haryana	2989				
15	Rajasthan	2750				
16	Odisha	1954				
17	Jammu and Kashmir	1922				
18	Telangana	1579				
19	Bihar	1460				
20	Assam	1075				
21	Jharkhand	1054				

1	Date	Confirmed cases	Active cases	Recovered cases	Deaths	Tested			
2	1-Feb	2	2	0	0	0			
3	1-Mar	5	2	3	0	0			
4	1-Apr	2059	1829	169	69	51151			
5	1-May	37263	26007	10021	1231	976363			
6	1-Jun	198372	97009	95744	5606	3966075			
7	1-Jul	605221	227401	359905	17848	9056173			
8	1-Aug	1752171	567417	1146917	37410	19821831			
9	1-Sep	3766108	799530	2899528	66462	44337201			
10	1-Oct	6392049	9425887	5348746	99807	76717728			
11	1-Nov	8229322	562329	7542905	122642	1.11E+08			
12	1-Dec	9499728	427528	8931803	138160	1.42E+08			
13	1-Jan	10306469	247427	9905570	149255	1.75E+08			

1	State	Recovered				
2	Maharashtra	1890323				
3	Karnataka	912205				
4	Andhra Pradesh	877212				
5	Tamil Nadu	813326				
6	Kerala	779097				
7	Delhi	619501				
8	Uttar Pradesh	579693				
9	West Bengal	548705				
10	Odisha	329801				
11	Rajasthan	308010				
12	Telangana	286244				
13	Chhattisgarh	284412				
14	Haryana	261510				
15	Bihar	253913				
16	Gujarat	245907				
17	Madhya Pradesh	241966				
18	Assam	212975				
19	Punjab	162762				
20	Jammu and Kashmir	120392				
21	Jharkhand	115542				

1	State	Tested					
2	West Bengal	76745126					
3	Uttar Pradesh	26789521					
4	Bihar	21312561					
5	Karnataka	16032142					
6	Maharashtra	14265878					
7	Andhra Pradesh	13214568					
8	Gujarat	10205632					
9	Delhi	9926544					
10	Kerala	8874963					
11	Telangana	7481236					
12	Odisha	7446162					
13	Assam	6264521					
14	Rajasthan	5627829					
15	Madhya Pradesh	5093612					
16	Jharkhand	5051203					
17	Haryana	4966522					
18	Jammu and Kashmir	4271346					
19	Punjab	4251366					
20	Tamil Nadu	1532064					
21	Chhattisgarh	394516					

1	State	Confirmed	Recovered	Deaths	Tested			
2	Maharashtra	1992683	1890323	50473	14265878			
3	Karnataka	932432	912205	12175	16032142			
4	Andhra Pradesh	886066	877212	7141	13214568			
5	Kerala	851195	779097	3481	8874963			
6	Tamil Nadu	831323	813326	12272	1532064			
7	Delhi	632590	619501	10754	9926544			
8	Uttar Pradesh	596904	579693	8580	26789521			
9	West Bengal	565661	548705	10063	76745126			
10	Odisha	333444	329801	1954	7446162			
11	Rajasthan	315394	308010	2750	5627829			
12	Chhattisgarh	293972	284412	3565	394516			
13	Telangana	291872	286244	1579	7481236			
14	Haryana	266428	261510	2989	4966522			
15	Bihar	258883	253913	1460	21312561			
16	Gujarat	256367	245907	4367	10205632			
17	Madhya Pradesh	251882	241966	3756	5093612			
18	Assam	216864	212975	1075	6264521			
19	Punjab	170729	162762	5509	4251366			
20	Jammu and Kashmir	123425	120392	1922	4271346			
21	Jharkhand	117786	115542	1054	5051203			

CONCLUSION

**IN THE LIGHT OF THE ABOVE,WE
CAN SEE THAT THE GRAPHICAL
METHOD IS USED TO REPRESENT
DATA BASED ON PAST AND
CURRENT INFORMATION.IT MORE
ACCURATELY DESCRIBES THE
COUNTINOUS NATURE OF INCREASE
AND DECREASE IN EACH CASES
THAT IS IN LINE GRAPH FORM AND
DISCRETE IN BAR GRAPH FORM.**

**THIS METHOD IS PERFORMED TO
GET A MORE ACCURATE ESTIMATE.**

BIBLIOGRAPHY

- **NCERT CLASS 11 I.P. TEXTBOOK**

- [HTTPS://WWW.COVID19INDIA.ORG/](https://www.covid19india.org/)
- WIKIPEDIA/PYTHON