In [1]:

```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
!wget -q http://mirror.its.dal.ca/apache/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz
!tar xvf spark-2.4.0-bin-hadoop2.7.tgz
!pip install -q findspark
```

```
The system cannot find the path specified.
'wget' is not recognized as an internal or external command,
operable program or batch file.
tar: Error opening archive: Failed to open 'spark-2.4.0-bin-hadoop2.7.tgz'
```

In [2]:

```
import pyspark
```

In [3]:

```
print(dir(pyspark))
```

```
['Accumulator', 'AccumulatorParam', 'BarrierTaskContext', 'BarrierTaskInfo',
'BasicProfiler', 'Broadcast', 'HiveContext', 'InheritableThread', 'MarshalSe
rializer', 'PickleSerializer', 'Profiler', 'RDD', 'RDDBarrier', 'Row', 'SQLC
ontext', 'SparkConf', 'SparkContext', 'SparkFiles', 'SparkJobInfo', 'SparkSt
ageInfo', 'StatusTracker', 'StorageLevel', 'TaskContext', '_NoValue', '__all
__', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__n
ame__', '__package__', '__path__', '__spec__', '__version__', '_globals', 'a
ccumulators', 'broadcast', 'cloudpickle', 'conf', 'context', 'copy_func', 'f
iles', 'find_spark_home', 'java_gateway', 'join', 'keyword_only', 'profile
r', 'rdd', 'rddsampler', 'resource', 'resultiterable', 'serializers', 'shuff
le', 'since', 'sql', 'statcounter', 'status', 'storagelevel', 'taskcontext',
'traceback_utils', 'types', 'util', 'version', 'wraps']
```

In [4]:

```
import pyspark.sql
```

In [5]:

```
print(dir(pyspark.sql))
```

```
['Catalog', 'Column', 'DataFrame', 'DataFrameNaFunctions', 'DataFrameReade
r', 'DataFrameStatFunctions', 'DataFrameWriter', 'GroupedData', 'HiveContex
t', 'PandasCogroupedOps', 'Row', 'SQLContext', 'SparkSession', 'UDFRegistrat
ion', 'Window', 'WindowSpec', '__all__', '__builtins__', '__cached__', '__do
c__', '__file__', '__loader__', '__name__', '__package__', '__path__', '__sp
ec__', 'catalog', 'column', 'conf', 'context', 'dataframe', 'group', 'panda
s', 'readwriter', 'session', 'streaming', 'types', 'udf', 'utils', 'window']
```

In [3]:

```
from pyspark.sql import SparkSession
```

In [4]:

```
spark=SparkSession.builder.master("local[*]").getOrCreate()
```

In [5]:

```
spark.conf.set('spark.sql.repl.eagerEval.enabled', True)
```

In [6]:

```
data_frame=spark.read.csv("C:\\Users\\Anandarshan\\Downloads\\covid_19_india.csv", inferSch
```

In [7]:

```
data_frame
```

Out[7]:

Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cure
1	2020-01-30	6:00 PM	Kerala	1	0	
2	2020-01-31	6:00 PM	Kerala	1	0	
3	2020-02-01	6:00 PM	Kerala	2	0	
4	2020-02-02	6:00 PM	Kerala	3	0	
5	2020-02-03	6:00 PM	Kerala	3	0	
6	2020-02-04	6:00 PM	Kerala	3	0	
7	2020-02-05	6:00 PM	Kerala	3	0	
8	2020-02-06	6:00 PM	Kerala	3	0	
9	2020-02-07	6:00 PM	Kerala	3	0	
10	2020-02-08	6:00 PM	Kerala	3	0	
11	2020-02-09	6:00 PM	Kerala	3	0	
12	2020-02-10	6:00 PM	Kerala	3	0	
13	2020-02-11	6:00 PM	Kerala	3	0	
14	2020-02-12	6:00 PM	Kerala	3	0	
15	2020-02-13	6:00 PM	Kerala	3	0	
16	2020-02-14	6:00 PM	Kerala	3	0	
17	2020-02-15	6:00 PM	Kerala	3	0	
18	2020-02-16	6:00 PM	Kerala	3	0	
19	2020-02-17	6:00 PM	Kerala	3	0	
20	2020-02-18	6:00 PM	Kerala	3	0	

only showing top 20 rows

Summary of the data_set

In [8]:

```
data_frame.select('*').describe().show(vertical=True)
```

```
-RECORD 0-----------------------------------------
 summary                  | count
 Sno                      | 16526
 Date                     | 16526
 Time                     | 16526
 State/UnionTerritory     | 16526
 ConfirmedIndianNational  | 16526
 ConfirmedForeignNational | 16526
 Cured                    | 16526
 Deaths                   | 16526
 Confirmed                | 16526
-RECORD 1-----------------------------------------
 summary                  | mean
 Sno                      | 8263.5
 Date                     | null
 Time                     | null
 State/UnionTerritory     | null
 ConfirmedIndianNational  | 12.188340807174887
 ConfirmedForeignNational | 1.4955156950672646
 Cured                    | 224545.9802735084
 Deaths                   | 3335.16053491468
 Confirmed                | 246823.90312235267
-RECORD 2-----------------------------------------
 summary                  | stddev
 Sno                      | 4770.789609697749
 Date                     | null
 Time                     | null
 State/UnionTerritory     | null
 ConfirmedIndianNational  | 21.582253392789077
 ConfirmedForeignNational | 3.57629242180641
 Cured                    | 495453.45797039923
 Deaths                   | 8890.528988545258
 Confirmed                | 541404.365345987
-RECORD 3-----------------------------------------
 summary                  | min
 Sno                      | 1
 Date                     | 2020-01-30
 Time                     | 10:00 AM
 State/UnionTerritory     | Andaman and Nicob...
 ConfirmedIndianNational  | -
 ConfirmedForeignNational | -
 Cured                    | 0
 Deaths                   | 0
 Confirmed                | 0
-RECORD 4-----------------------------------------
 summary                  | max
 Sno                      | 16526
 Date                     | 2021-06-28
 Time                     | 9:30 PM
 State/UnionTerritory     | West Bengal
 ConfirmedIndianNational  | 9
 ConfirmedForeignNational | 9
 Cured                    | 5790113
 Deaths                   | 121286
 Confirmed                | 6036821
```

In [9]:

```
data_frame.show()
```

```
+---+----------+-------+--------------------+----------------------+-------
----------------+-----+------+---------+
|Sno|      Date|   Time|State/UnionTerritory|ConfirmedIndianNational|Confirm
edForeignNational|Cured|Deaths|Confirmed|
+---+----------+-------+--------------------+----------------------+-------
----------------+-----+------+---------+
|  1|2020-01-30|6:00 PM|              Kerala|                     1|
0|    0|     0|        1|
|  2|2020-01-31|6:00 PM|              Kerala|                     1|
0|    0|     0|        1|
|  3|2020-02-01|6:00 PM|              Kerala|                     2|
0|    0|     0|        2|
|  4|2020-02-02|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
|  5|2020-02-03|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
|  6|2020-02-04|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
|  7|2020-02-05|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
|  8|2020-02-06|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
|  9|2020-02-07|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 10|2020-02-08|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 11|2020-02-09|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 12|2020-02-10|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 13|2020-02-11|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 14|2020-02-12|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 15|2020-02-13|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 16|2020-02-14|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 17|2020-02-15|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 18|2020-02-16|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 19|2020-02-17|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
| 20|2020-02-18|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|
+---+----------+-------+--------------------+----------------------+-------
----------------+-----+------+---------+
only showing top 20 rows
```

In [10]:

```
data_frame2=spark.read.csv("C:\\Users\\Anandarshan\\Downloads\\covid_vaccine_statewise.csv"
```

In [11]:

```
data_frame3=spark.read.csv("C:\\Users\\Anandarshan\\Downloads\\StatewiseTestingDetails.csv"
```

In [12]:

```
data_frame2
```

Out[12]:

Updated On	State	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)
16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	23757.0
17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	27348.0
18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	41361.0
19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	81901.0
20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	98111.0
21/01/2021	India	365965.0	32226.0	12600.0	365965.0	0.0	132784.0
22/01/2021	India	549381.0	36988.0	14115.0	549381.0	0.0	193899.0
23/01/2021	India	759008.0	43076.0	15605.0	759008.0	0.0	267856.0
24/01/2021	India	835058.0	49851.0	18111.0	835058.0	0.0	296283.0
25/01/2021	India	1277104.0	55151.0	19682.0	1277104.0	0.0	444137.0
26/01/2021	India	1293784.0	60821.0	21467.0	1293784.0	0.0	449119.0
27/01/2021	India	1726490.0	69495.0	23737.0	1726490.0	0.0	586081.0
28/01/2021	India	2295491.0	78523.0	25610.0	2295491.0	0.0	771229.0
29/01/2021	India	2814803.0	83664.0	26219.0	2814803.0	0.0	939069.0
30/01/2021	India	3067736.0	87822.0	26643.0	3067736.0	0.0	1022380.0
31/01/2021	India	3127107.0	91593.0	27011.0	3127107.0	0.0	1061307.0
01/02/2021	India	3350265.0	97432.0	27751.0	3350265.0	0.0	1152344.0
02/02/2021	India	3527971.0	106461.0	29522.0	3527971.0	0.0	1218507.0
03/02/2021	India	3825835.0	116568.0	31167.0	3825835.0	0.0	1324273.0
04/02/2021	India	4314304.0	126714.0	32505.0	4314304.0	0.0	1504527.0

only showing top 20 rows

In [13]:

```
print(dir(data_frame))
```

```
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__
format__', '__ge__', '__getattr__', '__getattribute__', '__getitem__', '__gt
__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__mod
ule__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__s
etattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_col
lect_as_arrow', '_jcols', '_jdf', '_jmap', '_jseq', '_lazy_rdd', '_repr_html
_', '_sc', '_schema', '_sort_cols', '_support_repr_html', '_to_corrected_pan
das_type', 'agg', 'alias', 'approxQuantile', 'cache', 'checkpoint', 'coalesc
e', 'colRegex', 'collect', 'columns', 'corr', 'count', 'cov', 'createGlobalT
empView', 'createOrReplaceGlobalTempView', 'createOrReplaceTempView', 'creat
eTempView', 'crossJoin', 'crosstab', 'cube', 'describe', 'distinct', 'drop',
'dropDuplicates', 'drop_duplicates', 'dropna', 'dtypes', 'exceptAll', 'expla
in', 'fillna', 'filter', 'first', 'foreach', 'foreachPartition', 'freqItem
s', 'groupBy', 'groupby', 'head', 'hint', 'inputFiles', 'intersect', 'inters
ectAll', 'isLocal', 'isStreaming', 'is_cached', 'join', 'limit', 'localCheck
point', 'mapInPandas', 'na', 'orderBy', 'persist', 'printSchema', 'randomSpl
it', 'rdd', 'registerTempTable', 'repartition', 'repartitionByRange', 'repla
ce', 'rollup', 'sameSemantics', 'sample', 'sampleBy', 'schema', 'select', 's
electExpr', 'semanticHash', 'show', 'sort', 'sortWithinPartitions', 'sql_ct
x', 'stat', 'storageLevel', 'subtract', 'summary', 'tail', 'take', 'toDF',
'toJSON', 'toLocalIterator', 'toPandas', 'transform', 'union', 'unionAll',
'unionByName', 'unpersist', 'where', 'withColumn', 'withColumnRenamed', 'wit
hWatermark', 'write', 'writeStream', 'writeTo']
```

In [14]:

```
data_frame3
```

Out[14]:

Date	State	TotalSamples	Negative	Positive
2020-04-17	Andaman and Nicob...	1403.0	1210	12.0
2020-04-24	Andaman and Nicob...	2679.0	null	27.0
2020-04-27	Andaman and Nicob...	2848.0	null	33.0
2020-05-01	Andaman and Nicob...	3754.0	null	33.0
2020-05-16	Andaman and Nicob...	6677.0	null	33.0
2020-05-19	Andaman and Nicob...	6965.0	null	33.0
2020-05-20	Andaman and Nicob...	7082.0	null	33.0
2020-05-21	Andaman and Nicob...	7167.0	null	33.0
2020-05-22	Andaman and Nicob...	7263.0	null	33.0
2020-05-23	Andaman and Nicob...	7327.0	null	33.0
2020-05-24	Andaman and Nicob...	7327.0	null	33.0
2020-05-25	Andaman and Nicob...	7363.0	null	33.0
2020-05-26	Andaman and Nicob...	7448.0	null	33.0
2020-05-27	Andaman and Nicob...	7499.0	null	33.0
2020-05-28	Andaman and Nicob...	7519.0	null	33.0
2020-05-29	Andaman and Nicob...	7567.0	null	33.0
2020-05-30	Andaman and Nicob...	7567.0	null	33.0
2020-05-31	Andaman and Nicob...	7706.0	null	33.0
2020-06-01	Andaman and Nicob...	7805.0	null	33.0
2020-06-02	Andaman and Nicob...	8086.0	null	33.0

only showing top 20 rows

In [15]:

```
#dataframe has 16527 rows and 9 columns
#dataframe2 has 6032 rows and 18 columns
#dataframe3 has 14801 rows and 5 columns
len(data_frame.columns)
len(data_frame2.columns)
```

Out[15]:

```
18
```

In [16]:

```
df=data_frame.filter(data_frame["Date"]>='2021-06-01')
```

In [17]:

```
print("showing the latest data_set")
df.show()
```

```
showing the latest data_set
+-----+----------+-------+--------------------+----------------------+-----
------------------+-------+------+---------+
|  Sno|      Date|   Time|State/UnionTerritory|ConfirmedIndianNational|Confi
rmedForeignNational|  Cured|Deaths|Confirmed|
+-----+----------+-------+--------------------+----------------------+-----
------------------+-------+------+---------+
|15519|2021-06-01|8:00 AM|Andaman and Nicob...|                     -|
-|   6719|   115|     7005|
|15520|2021-06-01|8:00 AM|      Andhra Pradesh|                     -|
-|1528360| 10930|  1693085|
|15521|2021-06-01|8:00 AM|   Arunachal Pradesh|                     -|
-|  23402|   115|    27272|
|15522|2021-06-01|8:00 AM|               Assam|                     -|
-| 354810|  3365|   411216|
|15523|2021-06-01|8:00 AM|               Bihar|                     -|
-| 685362|  5163|   706761|
|15524|2021-06-01|8:00 AM|          Chandigarh|                     -|
-|  57526|   753|    60046|
|15525|2021-06-01|8:00 AM|        Chhattisgarh|                     -|
-| 922674| 13048|   971463|
|15526|2021-06-01|8:00 AM|Dadra and Nagar H...|                     -|
-|   9957|     4|    10286|
|15527|2021-06-01|8:00 AM|               Delhi|                     -|
-|1390963| 24237|  1426240|
|15528|2021-06-01|8:00 AM|                 Goa|                     -|
-| 140254|  2649|   155666|
|15529|2021-06-01|8:00 AM|             Gujarat|                     -|
-| 766991|  9833|   809169|
|15530|2021-06-01|8:00 AM|             Haryana|                     -|
-| 729752|  8303|   756635|
|15531|2021-06-01|8:00 AM|    Himachal Pradesh|                     -|
-| 173566|  3143|   190330|
|15532|2021-06-01|8:00 AM|   Jammu and Kashmir|                     -|
-| 251463|  3907|   290465|
|15533|2021-06-01|8:00 AM|           Jharkhand|                     -|
-| 323876|  4991|   337774|
|15534|2021-06-01|8:00 AM|           Karnataka|                     -|
-|2261590| 29090|  2604431|
|15535|2021-06-01|8:00 AM|              Kerala|                     -|
-|2310385|  8815|  2526579|
|15536|2021-06-01|8:00 AM|              Ladakh|                     -|
-|  16859|   189|    18662|
|15537|2021-06-01|8:00 AM|         Lakshadweep|                     -|
-|   6242|    33|     8077|
|15538|2021-06-01|8:00 AM|      Madhya Pradesh|                     -|
-| 748573|  8067|   780030|
+-----+----------+-------+--------------------+----------------------+-----
------------------+-------+------+---------+
only showing top 20 rows
```

Finding Null values

In [18]:

```
#Step1:- Data Analysis
 #       showing no null values
print("No null values in the data_frame")
from pyspark.sql.functions import isnan, when, count, col
data_frame.select([count(when(col('Date').isNull(),True))])
data_frame.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in data_frame.


#data_frame2.select([count(when(col('Date').isNull(),True))]).show()
data_frame2.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in data_frame


#data_frame3.select([count(when(col('Date').isNull(),True))]).show()
data_frame3.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in data_frame
```

```
No null values in the data_frame
+---+----+----+-------------------+----------------------+-------------
----------+-----+------+---------+
|Sno|Date|Time|State/UnionTerritory|ConfirmedIndianNational|ConfirmedForei
gnNational|Cured|Deaths|Confirmed|
+---+----+----+-------------------+----------------------+-------------
----------+-----+------+---------+
|  0|   0|   0|                  0|                     0|
0|    0|     0|        0|
+---+----+----+-------------------+----------------------+-------------
----------+-----+------+---------+

+----------+-----+---------------------------+-----------------------+--
----------+----------------------+----------------------+-------------
--------------+-----------------------------+--------------------------
-------+--------------------------+----------------------------+---------
------------------+----+----------------+----------------+-------------
--+------------------------+
|Updated On|State|Total Individuals Vaccinated|Total Sessions Conducted|To
tal Sites |First Dose Administered|Second Dose Administered|Male(Individua
ls Vaccinated)|Female(Individuals Vaccinated)|Transgender(Individuals Vacc
inated)|Total Covaxin Administered|Total CoviShield Administered|Total Spu
tnik V Administered|AEFI|18-45 years (Age)|45-60 years (Age)|60+ years (Ag
e)|Total Doses Administered|
+----------+-----+---------------------------+-----------------------+--
----------+----------------------+----------------------+-------------
--------------+-----------------------------+--------------------------
-------+--------------------------+----------------------------+---------
------------------+----+----------------+----------------+-------------
--+------------------------+
|         0|    0|                          1|                      1|
1|                     1|                         1|
1|                           1|                                  1|
1|                          1|                      4627|2184|
2187|             2186|           2186|                      0|
+----------+-----+---------------------------+-----------------------+--
----------+----------------------+----------------------+-------------
--------------+-----------------------------+--------------------------
-------+--------------------------+----------------------------+---------
------------------+----+----------------+----------------+-------------
--+------------------------+
```

```
+----+-----+------------+--------+--------+
|Date|State|TotalSamples|Negative|Positive|
+----+-----+------------+--------+--------+
|   0|    0|           0|    8232|    9237|
+----+-----+------------+--------+--------+
```

In [19]:

```
# Using select() to Add Multiple Column
data_frame.select('*', ((data_frame.Confirmed-(data_frame.Cured + data_frame.Deaths)).alias
```

```
+---+----------+-------+--------------------+----------------------+-------
-----------------+-----+------+---------+------+
|Sno|      Date|   Time|State/UnionTerritory|ConfirmedIndianNational|Confirm
edForeignNational|Cured|Deaths|Confirmed|active|
+---+----------+-------+--------------------+----------------------+-------
-----------------+-----+------+---------+------+
|  1|2020-01-30|6:00 PM|              Kerala|                     1|
0|    0|     0|        1|     1|
|  2|2020-01-31|6:00 PM|              Kerala|                     1|
0|    0|     0|        1|     1|
|  3|2020-02-01|6:00 PM|              Kerala|                     2|
0|    0|     0|        2|     2|
|  4|2020-02-02|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
|  5|2020-02-03|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
|  6|2020-02-04|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
|  7|2020-02-05|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
|  8|2020-02-06|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
|  9|2020-02-07|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 10|2020-02-08|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 11|2020-02-09|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 12|2020-02-10|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 13|2020-02-11|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 14|2020-02-12|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 15|2020-02-13|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 16|2020-02-14|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 17|2020-02-15|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 18|2020-02-16|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 19|2020-02-17|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
| 20|2020-02-18|6:00 PM|              Kerala|                     3|
0|    0|     0|        3|     3|
+---+----------+-------+--------------------+----------------------+-------
-----------------+-----+------+---------+------+
only showing top 20 rows
```

DATA VISUALIZATION

In [20]:

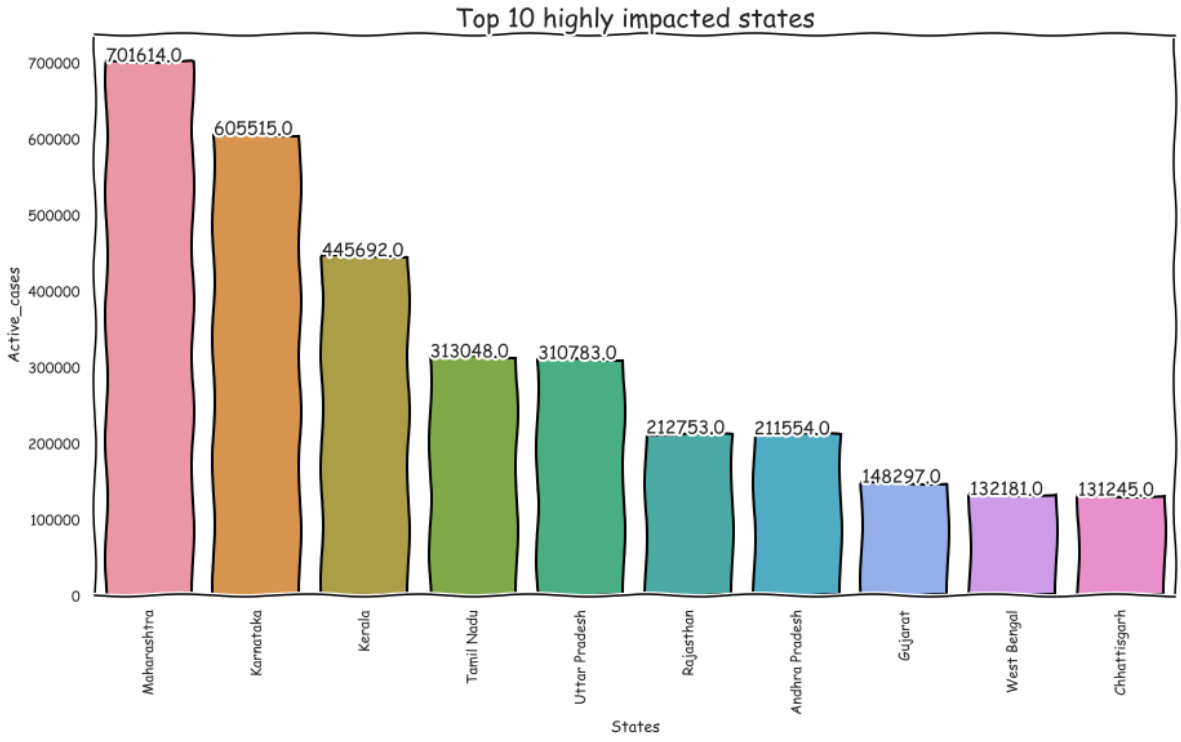
```
import matplotlib.pyplot as plt
import matplotlib.dates as mtd
import seaborn as sns
from matplotlib.ticker import ScalarFormatter
colors=['#0C68C7','#3A6794','#00FAF3','#FA643C','#C71D12']
sns.set(palette=colors, style='white')

sns.palplot(colors)
df=data_frame
```

In [21]:

```
import pandas as pd
df=pd.read_csv("C:\\Users\\Anandarshan\\Downloads\\covid_19_india.csv")
df.rename(columns={'State/UnionTerritory':'States'}, inplace=True)
df['Active_cases']=df['Confirmed']-(df['Cured']+df['Deaths'])
```

In [26]:

```
top_10=df.groupby(by='States').max()[['Active_cases','Date']].sort_values(by=['Active_cases
with plt.xkcd():
    fig=plt.figure(figsize=(15,8))
    plt.title("Top 10 highly impacted states", size=20)
    ax=sns.barplot(data=top_10.iloc[:10],y='Active_cases',x='States', linewidth=2, edgecolo
    ax.set_xticklabels(labels=ax.get_xticklabels(),rotation=90)
    for i in ax.patches:
        ax.text(x=i.get_x(),y=i.get_height(),s=i.get_height())
```



CONCLUSION

This bar plot shows us that, there are three states namely Maharashtra, Karnataka, Kerala which is exceeding 3lakhs, 50 thousand active number of mean cases. These states should prioritize adopting strict covid19 curbs such as imposing lockdowns, section 144 and improve their health infrastructure.

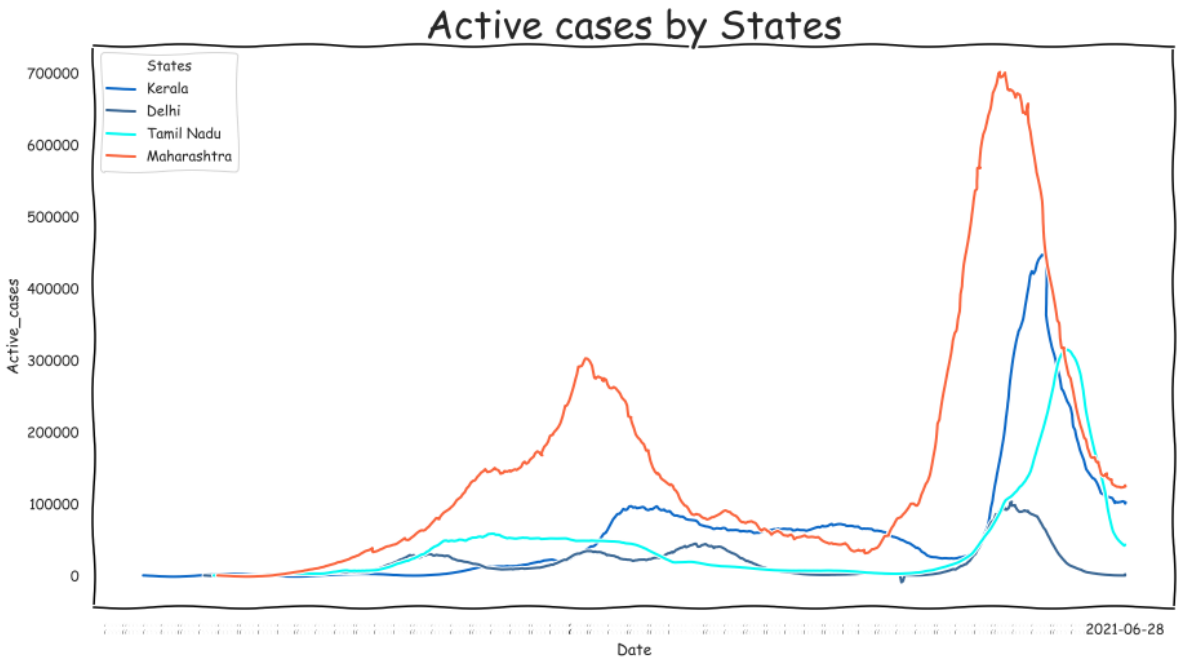
There is a high number of chances that third wave might emerge or start from these three states if these following curbs are not followed.

In [2]:

```
import pandas as pd
dataframe2=pd.read_csv("C:\\Users\\Anandarshan\\Downloads\\covid_vaccine_statewise.csv")
```

In [22]:

```
fig=plt.figure(figsize=(15,8))
with plt.xkcd():
    ax=sns.lineplot(data=df[df['States'].isin(['Kerala','Tamil Nadu','Delhi','Maharashtra']
    ax.set_title("Active cases by States", size=30)
```



In [ ]:

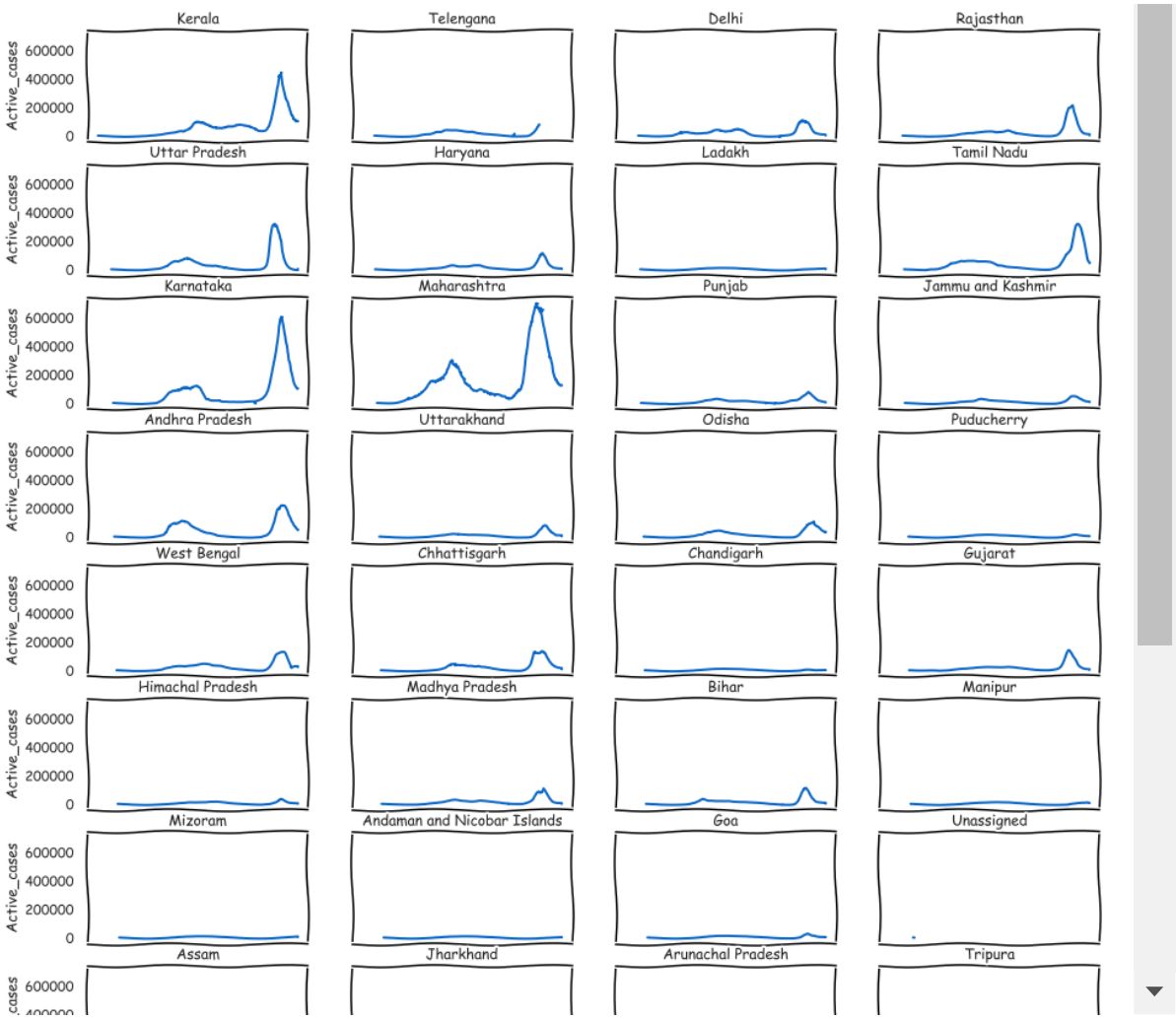
In [23]:

```
#Lets convert the Date feature to Date&time datatype
df['Date']=pd.to_datetime(df['Date'],format='%Y-%m-%d')

#Time is not required as it doesnt make much difference
df.drop(['Time'],axis=1, inplace=True)

#Renaming State/UnionTerritory to States for easy reference
df.rename(columns={'State/UnionTerritory':'States'}, inplace=True)
```

In [24]:

```
with plt.xkcd():
    fig, ax=plt.subplots(nrows=10, ncols=4, figsize=[15,20], sharex=True, sharey=True)
    ax=ax.flatten()

    for i,s in enumerate(df['States'].unique()):
        data1=df[df['States']==s][['Date','Active_cases']]
        sns.lineplot(data=data1, x='Date',y='Active_cases', ax=ax[i])
        ax[i].set_title(s)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-24-00365d9169ff> in <module>
      5     for i,s in enumerate(df['States'].unique()):
      6         data1=df[df['States']==s][['Date','Active_cases']]
----> 7         sns.lineplot(data=data1, x='Date',y='Active_cases', ax=ax[i]
)
      8         ax[i].set_title(s)

IndexError: index 40 is out of bounds for axis 0 with size 40
```



CONCLUSION

We can see a steep rise in active cases for the following states - Maharashtra, Karnataka, Uttar Pradesh, Andhra Pradesh, Uttarakhand, West Bengal and Delhi.

The major factors that influenced such a drastic situation were:

1)Maharashtra saw a steep rise in active cases since it's a densely populated state.

2)Kerela had high active cases due to the Indian nationals returning back from foreign countries.

3)Karnataka - Since Kerela and Maharashtra share boundaries with Karnataka, due to the exchange of goods and public transportation, led to an increase in number of cases in Karnataka.

4)Uttrakhand - Events like Kumbh Mela served as the major factor contributing to high corona cases.

5)Delhi - The rising cases in Delhi are attributed to social gatherings during the festivities, deteriorating air quality, increasing incidences of respiratory disorders and clusters of positive cases at workplaces. The fatigue among frontline workers also served as a reason for a high number of cases.

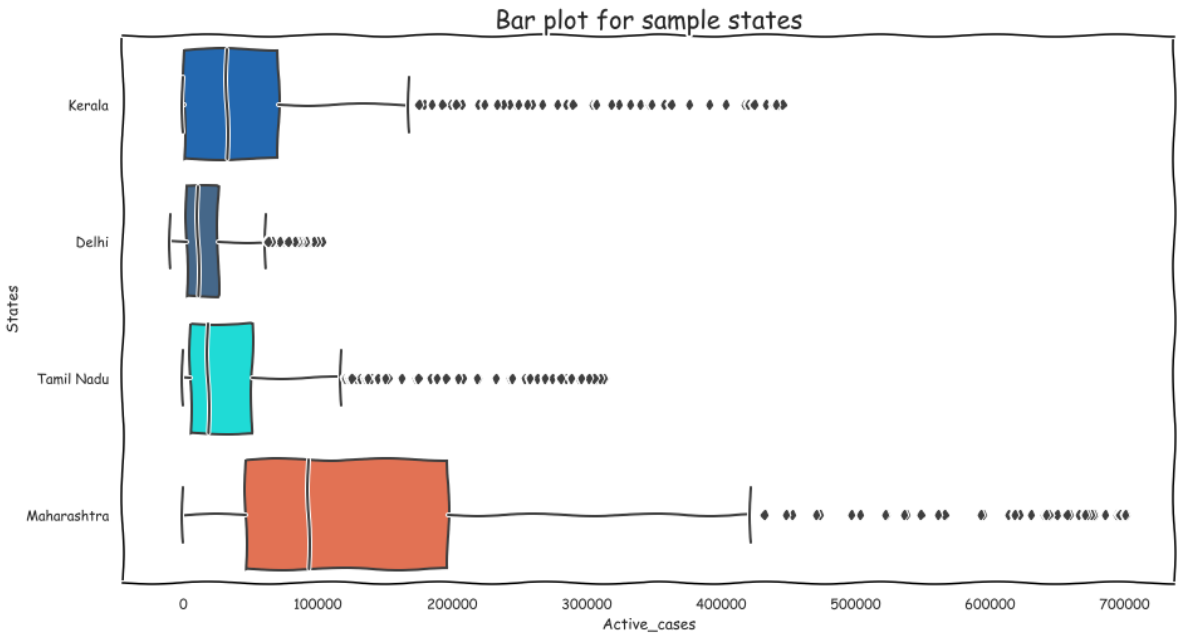
6)West Bengal - The eight-phase assembly election had seen all political parties and their leaders hold massive rallies and public gatherings.This was a major factor influencing the number of cases in the state.

Outliners Indentification

In [25]:

```
print("Outliners for states vs active_cases")
with plt.xkcd():
    fig=plt.figure(figsize=(15,8))
    sns.boxplot(data=df[df['States'].isin(['Kerala','Tamil Nadu','Delhi','Maharashtra'])],x
    plt.title("Bar plot for sample states" ,size=20)
```

```
Outliners for states vs active_cases
```



In [26]:

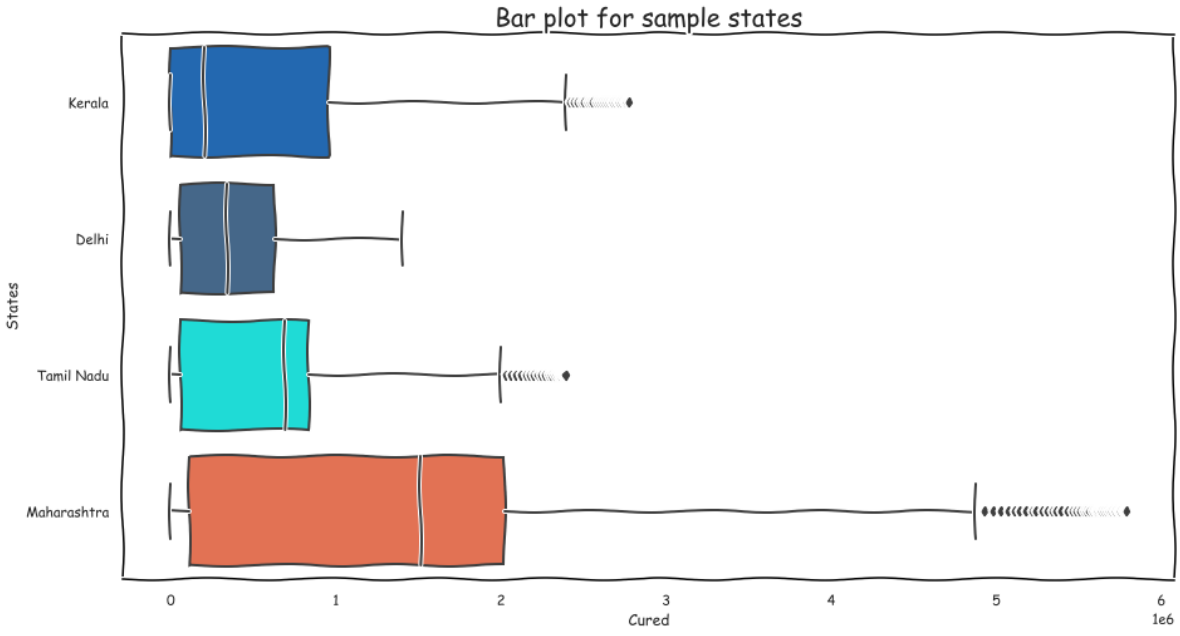
```
median_states=df[df['States'].isin(['Kerala','Tamil Nadu','Delhi','Maharashtra'])]
median_states.groupby(by=['States']).median().style.bar(['Active_cases'])
```

Out[26]:

States	Sno	Cured	Deaths	Confirmed	Active_cases
Delhi	7847.500000	336309.000000	6409.500000	372883.500000	10984.500000
Kerala	7295.500000	203495.500000	1035.500000	299514.000000	32623.000000
Maharashtra	7981.000000	1514079.000000	44024.000000	1683775.000000	93400.000000
Tamil Nadu	7956.000000	691236.000000	11122.000000	724522.000000	18395.000000

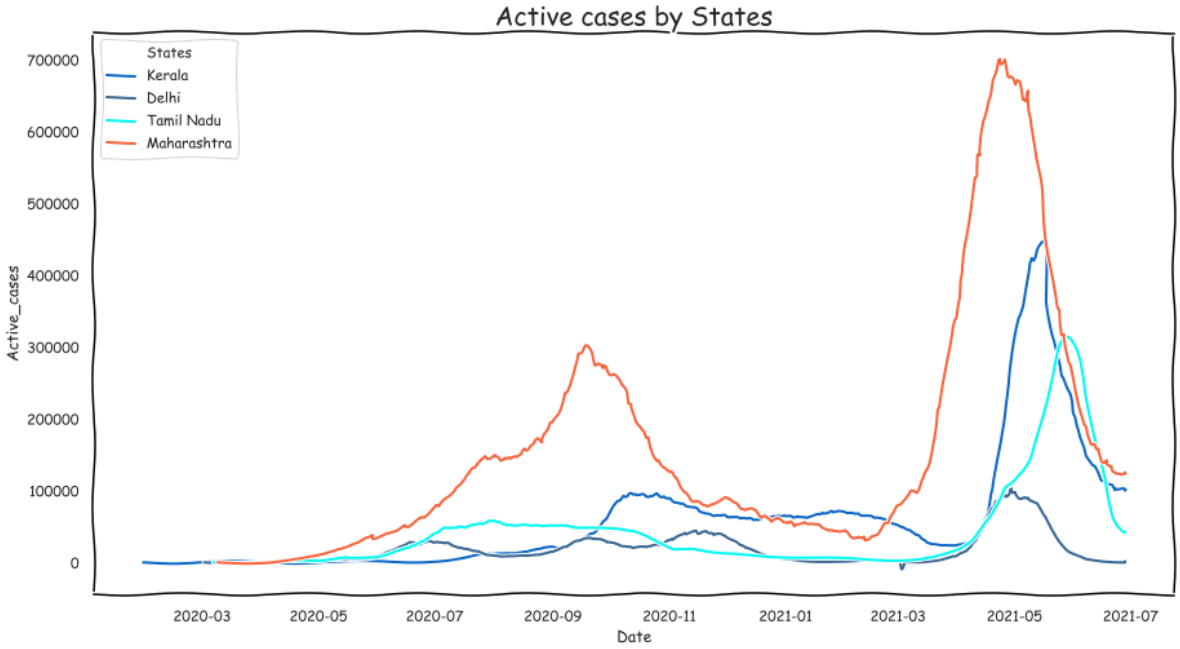
In [27]:

```
with plt.xkcd():
    fig=plt.figure(figsize=(15,8))
    sns.boxplot(data=df[df['States'].isin(['Kerala','Tamil Nadu','Delhi','Maharashtra'])],x
    plt.title("Bar plot for sample states" ,size=20)
```



In [28]:

```
fig=plt.figure(figsize=(15,8))
with plt.xkcd():
    ax=sns.lineplot(data=df[df['States'].isin(['Kerala','Tamil Nadu','Delhi','Maharashtra']
    ax.set_title("Active cases by States", size=20)
```



TITLE:

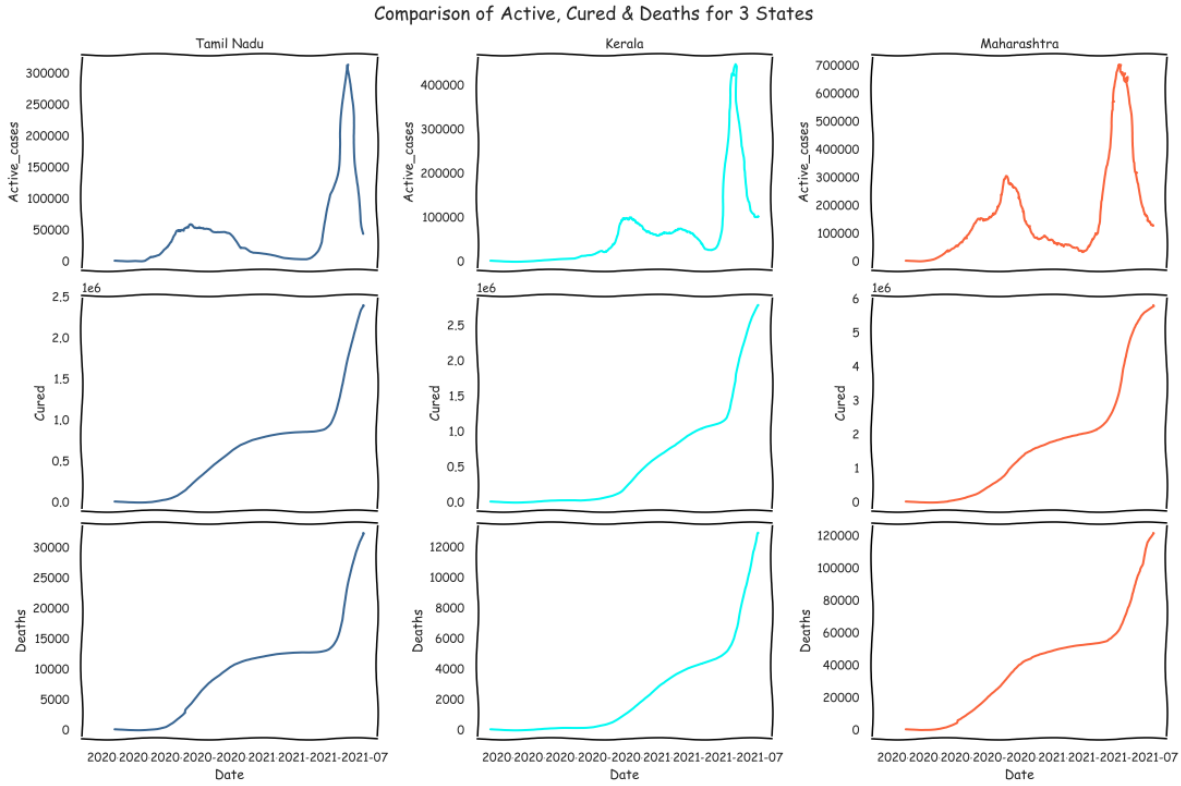
Drawing a comparization between the active cases of 4 major states- Kerela,Delhi,Maharashtra,Tamil Nadu against theTime line March 2020 to July 2021

CONCLUSION:

This line plot shows a sudden rise in active cases in Kerala which proves that it was hiding the actual number of reported cases. The authenticity of the official information provided by Kerala in the future must be investigated and the Centre must pay proper heed to the occurrence of such events and ensure no state repeats this in future. If such things continue, the accuracy of statistics such as the total number of vaccinated people, the number of deaths and current active cases will be affected. Due to this kind of inaccurate date, the public masses discontinue the practicing of covid19 precautions which has a recursive effect.

In [29]:

```
states=['Kerala', 'Tamil Nadu', 'Maharashtra']
tn=df[df['States']=='Tamil Nadu']
kl=df[df['States']=='Kerala']
mh=df[df['States']=='Maharashtra']

with plt.xkcd():

    fig, ax=plt.subplots(nrows=3, ncols=3, figsize=(15,10), squeeze=False, sharex=True, sha
    plt.suptitle("Comparison of Active, Cured & Deaths for 3 States")
    sns.lineplot(data=tn, x='Date',y='Active_cases', ax=ax[0,0], color=colors[1])
    ax[0,0].set_title("Tamil Nadu")
    sns.lineplot(data=tn, x='Date',y='Cured', ax=ax[1,0], color=colors[1])
    sns.lineplot(data=tn, x='Date',y='Deaths', ax=ax[2,0], color=colors[1])

    sns.lineplot(data=kl, x='Date',y='Active_cases', ax=ax[0,1], color=colors[2])
    ax[0,1].set_title("Kerala")
    sns.lineplot(data=kl, x='Date',y='Cured', ax=ax[1,1], color=colors[2])
    sns.lineplot(data=kl, x='Date',y='Deaths', ax=ax[2,1], color=colors[2])

    sns.lineplot(data=mh, x='Date',y='Active_cases', ax=ax[0,2], color=colors[3])
    ax[0,2].set_title("Maharashtra")
    sns.lineplot(data=mh, x='Date',y='Cured', ax=ax[1,2], color=colors[3])
    sns.lineplot(data=mh, x='Date',y='Deaths', ax=ax[2,2], color=colors[3])
```



Conclusion

We see that the states - Tamil Nadu, Kerala and Maharashtra share similar trends in the number of active_cases, cured cases and deaths during the first and second peaks.

With the help of this, we can conclude that if one of the states starts observing a steep rise, the probability of the other 2 states to observe a similar pattern is very high and in this case the states can adopt strict measures to control the rising cases.

In addition to this, the states can in advance prepare for a sufficient amount of oxygen supply and focus on improving their health infrastructure.

In [30]:

```
tn=df[df['States']=='Tamil Nadu']['Cured']
mh=df[df['States']=='Maharashtra']['Cured']
kl=df[df['States']=='Kerala']['Cured']

from scipy.stats import ttest_ind
```

In [ ]:

```

```

Hypothesis Testing

In [31]:

```
st,p_value=ttest_ind(tn,kl)
if p_value <0.05:
    print("Both states {} & {} have significant difference in Cure rate".format('Tamil Nadu
else:
    print("Both states {} & {} have no significant difference in Cure rate".format('Tamil N

st,p_value=ttest_ind(tn,mh)
if p_value <0.05:
    print("Both states {} & {} have significant difference in Cure rate".format('Tamil Nadu
else:
    print("Both states {} & {} have no significant difference in Cure rate".format('Tamil N

st,p_value=ttest_ind(kl,mh)
if p_value <0.05:
    print("Both states {} & {} have significant difference in Cure rate".format('Kerala','M
else:
    print("Both states {} & {} have no significant difference in Cure rate".format('Kerala'
```

```
Both states Tamil Nadu & Kerala have no significant difference in Cure rate
Both states Tamil Nadu & Maharashtra have significant difference in Cure rat
e
Both states Kerala & Maharashtra have significant difference in Cure rate
```

In [32]:

```
    #lets take mean impacted vs mean cured
tn_cured = df[df['States']=='Tamil Nadu']['Cured'].max()
mh_cured=df[df['States']=='Maharashtra']['Cured'].max()
kl_cured=df[df['States']=='Kerala']['Cured'].max()

tn_active = df[df['States']=='Tamil Nadu']['Confirmed'].max()
mh_active=df[df['States']=='Maharashtra']['Confirmed'].max()
kl_active=df[df['States']=='Kerala']['Confirmed'].max()
```

Cured Cases Proportions

In [33]:

```
print([tn_cured, mh_cured] , [tn_active, mh_active])
print(f' Proportion of cured cases in Tamil Nadu, Maharastra = {round(tn_cured/tn_active,2)

print([tn_cured, kl_cured] , [tn_active, kl_active])
print(f' Proportion of cured cases in Tamil Nadu, Kerala = {round(tn_cured/tn_active,2)}%,

print([mh_cured, kl_cured] , [mh_active, kl_active])
print(f' Proportion of cured cases in Tamil Nadu, Kerala = {round(mh_cured/mh_active,2)}%,
```

```
[2390783, 5790113] [2465874, 6036821]
 Proportion of cured cases in Tamil Nadu, Maharastra = 0.97%, 0.96% respecti
vely

[2390783, 2775967] [2465874, 2888894]
 Proportion of cured cases in Tamil Nadu, Kerala = 0.97%, 0.96% respectively

[5790113, 2775967] [6036821, 2888894]
 Proportion of cured cases in Tamil Nadu, Kerala = 0.96%, 0.96% respectively
```

Propotions_zTest

In [52]:

```
from statsmodels.stats.proportion import proportions_ztest
tn_cured = df[df['States']=='Tamil Nadu']['Deaths'].max()
mh_cured=df[df['States']=='Maharashtra']['Deaths'].max()
kl_cured=df[df['States']=='Kerala']['Deaths'].max()

tn_active = df[df['States']=='Tamil Nadu']['Confirmed'].max()
mh_active=df[df['States']=='Maharashtra']['Confirmed'].max()
kl_active=df[df['States']=='Kerala']['Confirmed'].max()

print([tn_cured, mh_cured] , [tn_active, mh_active])
print(f' Proportion of Death cases in Tamil Nadu, Maharastra = {round(tn_cured/tn_active,2)

print([tn_cured, kl_cured] , [tn_active, kl_active])
print(f' Proportion of Death cases in Tamil Nadu, Kerala = {round(tn_cured/tn_active,2)}%,

print([mh_cured, kl_cured] , [mh_active, kl_active])
print(f' Proportion of Death cases in Tamil Nadu, Kerala = {round(mh_cured/mh_active,2)}%,


stat, p_value = proportions_ztest([tn_cured, mh_cured] , [tn_active, mh_active])

if p_value <0.05:
    print("Both states {} & {} have significant difference in Death rate".format('Tamil Nad
else:
    print("Both states {} & {} have no significant difference in Death rate".format('Tamil

stat, p_value = proportions_ztest([tn_cured, kl_cured] , [tn_active, kl_active])

if p_value <0.05:
    print("Both states {} & {} have significant difference in Death rate".format('Tamil Nad
else:
    print("Both states {} & {} have no significant difference in Death rate".format('Tamil

stat, p_value = proportions_ztest([kl_cured, mh_cured] , [kl_active, mh_active])

if p_value <0.05:
    print("Both states {} & {} have significant difference in Death rate".format('Kerala','
else:
    print("Both states {} & {} have no significant difference in Death rate".format('Kerala
```

```
[32290, 121286] [2465874, 6036821]
 Proportion of Death cases in Tamil Nadu, Maharastra = 0.01%, 0.02% respecti
vely

[32290, 12879] [2465874, 2888894]
 Proportion of Death cases in Tamil Nadu, Kerala = 0.01%, 0.0% respectively

[121286, 12879] [6036821, 2888894]
 Proportion of Death cases in Tamil Nadu, Kerala = 0.02%, 0.0% respectively

Both states Tamil Nadu & Maharashtra have significant difference in Death ra
te
Both states Tamil Nadu & Kerala have significant difference in Death rate
Both states Kerala & Maharashtra have significant difference in Death rate
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```