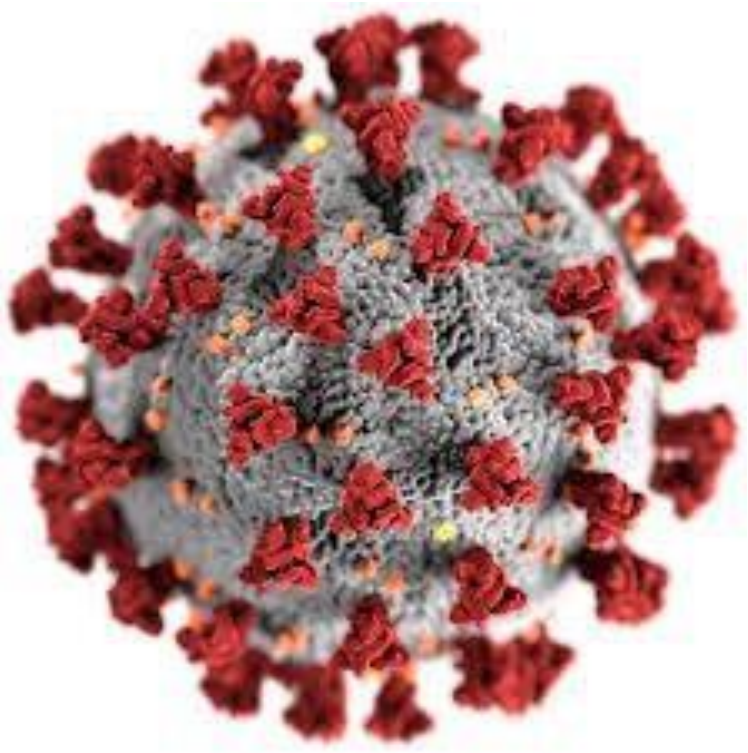
COVID-19

Group-6



Group Members

Name	CWID	HAWK ID
Sibo Bai	A20434344	sbai8@hawk.iit.edu
Sarthak Agrawal	A20444355	sagrawal4@hawk.iit.edu
Shivani Agrawal	A20443602	sagrawal3@hawk.iit.edu
Vaibhavi kulkarni	A20452251	vkulkarni3@hawk.iit.edu
Shreeya Deshpande	A20438928	sdeshpande4@hawk.iit.edu
Ramakant Chaudhari	A20446183	rchaudhari1@hawk.iit.edu

Index

Abstract

Nowadays, COVID-19 is a global pandemic according to the assessment. In this report, our group is trying to find out how much impact and casualty the United States have faced state-wise till now. Here, with the help of data warehousing, SQL, and Pentaho transformation knowledge we are building a cumulative study model and displaying it with the help of some visualization graphs using tableau. Here, we discuss the design of data warehouses for COVID-19. We have collected an extensive set of data on COVID-19 from the Kaggle website. We have done all the necessary pre-processing on this data. We have implemented ETL to simplify source code conversion to realize the diversity of data points for reporting. Then we design the type of visualization to attract audiences by using interactive elements. We believe that our prediction and analysis of the data is very useful for COVID-19.

Introduction

A new coronavirus designated 2019-nCoV was first identified in Wuhan, the capital of China's Hubei province. People developed pneumonia without a clear cause and for which existing vaccines or treatments were not effective. The virus has shown evidence of human-to-human transmission. In the main page of Our World in Data, we can find a daily updated comprehensive overview of the data and research on the pandemic of COVID-19. The visualizations of confirmed cases and confirmed deaths are updated daily and published by the European centers for disease control, the best global data set on the epidemic.

We started by collecting the data and then Cleaning it so that we could proceed with transforming it. The data cleaning process involved the following steps:

Data Preparation

```
1  setwd("C:\\Users\\13128\\Downloads\\us-counties-covid-19-dataset", header = T)
2  MyData = read.csv("us-counties.csv", header = T)
3  str(MyData)
4
```

```
> setwd("C:/Users/13128/Downloads/us-counties-covid-19-dataset")
> MyData = read.csv("us-counties.csv", header = T)
> str(MyData)
'data.frame':   53848 obs. of  6 variables:
 $ date  : Factor w/ 83 levels "1/21/2020","1/22/2020",..: 1 2 3 4 4 5 5 5 6 6 ...
 $ county: Factor w/ 1608 levels "Abbeville","Acadia",..: 1325 1325 1325 348 1325 1066 348 1325 886 850 ...
 $ state : Factor w/ 55 levels "Alabama","Alaska",..: 52 52 52 15 52 5 15 52 3 5 ...
 $ fips  : int  53061 53061 53061 17031 53061 6059 17031 53061 4013 6037 ...
 $ cases : Factor w/ 1579 levels "0","1","10","100",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ deaths: int  0 0 0 0 0 0 0 0 0 0 ...
>
```

Using "***setwd***" to get the path and working directory of our folder in which our data is present.

Using "***MyData***" we are loading our .csv file into the data frame.

Using "***str***" displaying various columns with their attributes on which we have to work.

```
1  setwd("C:\\Users\\13128\\Downloads\\us-counties-covid-19-dataset", header = T)
2  MyData = read.csv("us-counties.csv", header = T)
3  str(MyData)
4  View(MyData)
5
```

```
> setwd("C:/Users/13128/Downloads/us-counties-covid-19-dataset")
> MyData = read.csv("us-counties.csv", header = T)
> str(MyData)
'data.frame':   53848 obs. of  6 variables:
 $ date  : Factor w/ 83 levels "1/21/2020","1/22/2020",..: 1 2 3 4 4 5 5 5 6 6 ...
 $ county: Factor w/ 1608 levels "Abbeville","Acadia",..: 1325 1325 1325 348 1325 1066 348 1325 886 850 ...
 $ state : Factor w/ 55 levels "Alabama","Alaska",..: 52 52 52 15 52 5 15 52 3 5 ...
 $ fips  : int  53061 53061 53061 17031 53061 6059 17031 53061 4013 6037 ...
 $ cases : Factor w/ 1579 levels "0","1","10","100",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ deaths: int  0 0 0 0 0 0 0 0 0 0 ...
> View(MyData)
>
```

Using "***View***" function, we find whether the correct file is loaded or not.

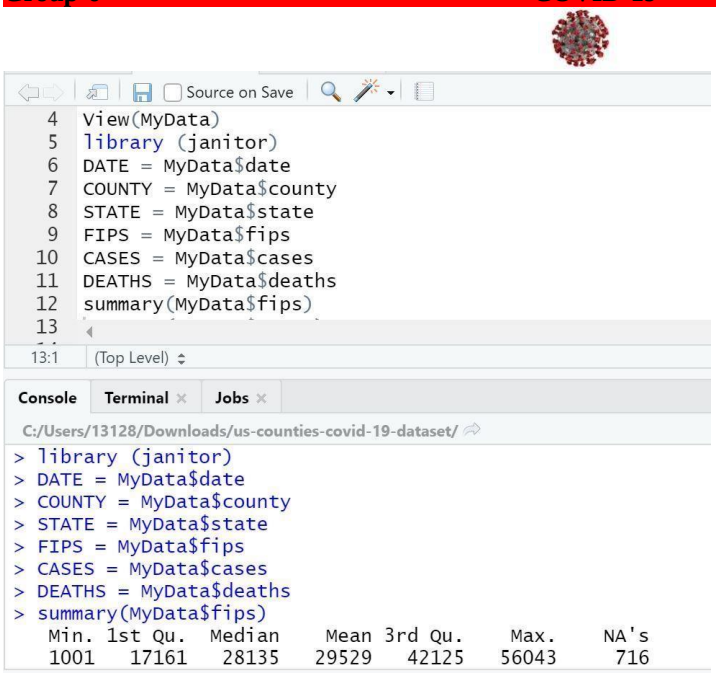
Our data looks like this:

trial1.R* | trial1.R* | MyData

Filter

	date	county	state	fips	cases	deaths
1	1/21/2020	Snohomish	Washington	53061	1	0
2	1/22/2020	Snohomish	Washington	53061	1	0
3	1/23/2020	Snohomish	Washington	53061	1	0
4	1/24/2020	Cook	Illinois	17031	1	0
5	1/24/2020	Snohomish	Washington	53061	1	0
6	1/25/2020	Orange	California	6059	1	0
7	1/25/2020	Cook	Illinois	17031	1	0
8	1/25/2020	Snohomish	Washington	53061	1	0
9	1/26/2020	Maricopa	Arizona	4013	1	0
10	1/26/2020	Los Angeles	California	6037	1	0
11	1/26/2020	Orange	California	6059	1	0
12	1/26/2020	Cook	Illinois	17031	1	0

Showing 1 to 14 of 53,848 entries, 6 total columns



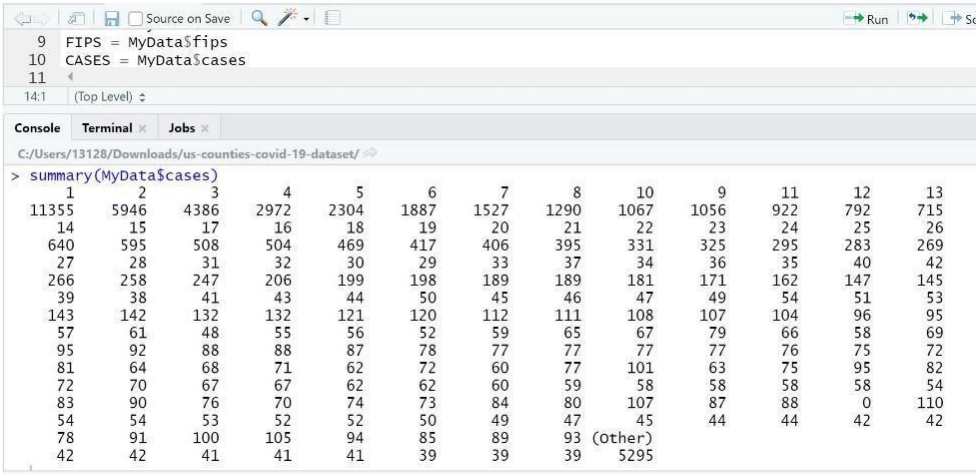
```
Source on Save
 4  View(MyData)
 5  library (janitor)
 6  DATE = MyData$date
 7  COUNTY = MyData$county
 8  STATE = MyData$state
 9  FIPS = MyData$fips
10  CASES = MyData$cases
11  DEATHS = MyData$deaths
12  summary(MyData$fips)
13
13:1  (Top Level)

Console  Terminal  Jobs
C:/Users/13128/Downloads/us-counties-covid-19-dataset/
> library (janitor)
> DATE = MyData$date
> COUNTY = MyData$county
> STATE = MyData$state
> FIPS = MyData$fips
> CASES = MyData$cases
> DEATHS = MyData$deaths
> summary(MyData$fips)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   1001   17161   28135   29529   42125   56043     716
```

Using the library "***Janitor***" we load the attributes for the pre-processing that needs to be performed.

Using "***Summary***" we get the various factors with null values present in the dataset.

We can see that in the "***fips***" column there are 700+ missing values.

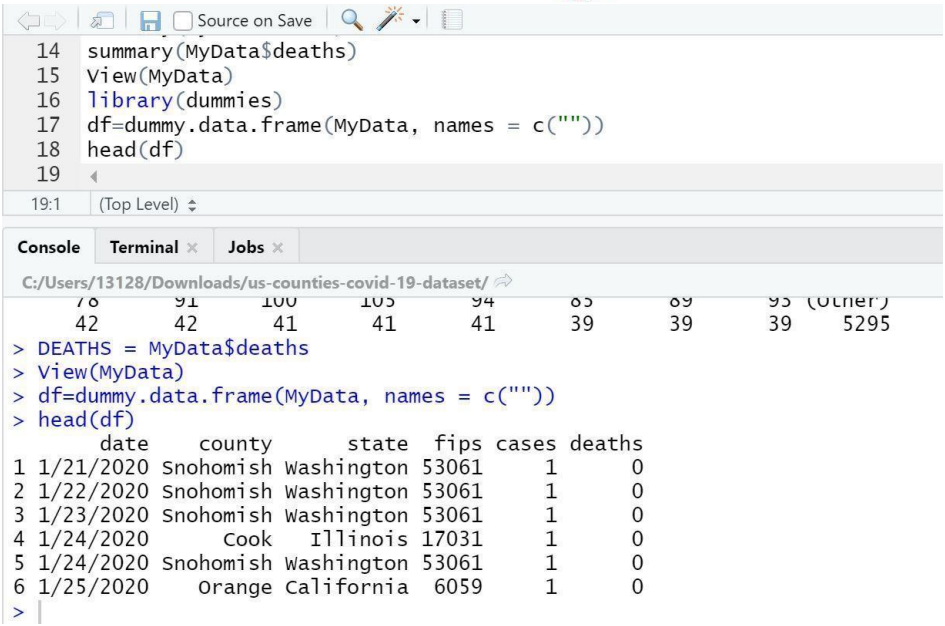


```
Source on Save  Run
 9  FIPS = MyData$fips
10  CASES = MyData$cases
11
14:1  (Top Level)

Console  Terminal  Jobs
C:/Users/13128/Downloads/us-counties-covid-19-dataset/
> summary(MyData$cases)
     1      2      3      4      5      6      7      8     10      9     11     12     13
 11355   5946   4386   2972   2304   1887   1527   1290   1067   1056    922    792    715
    14     15     17     16     18     19     20     21     22     23     24     25     26
   640    595    508    504    469    417    406    395    331    325    295    283    269
    27     28     31     32     30     29     33     37     34     36     35     40     42
   266    258    247    206    199    198    189    189    181    171    162    147    145
    39     38     41     43     44     50     45     46     47     49     54     51     53
   143    142    132    132    121    120    112    111    108    107    104     96     95
    57     61     48     55     56     52     59     65     67     79     66     58     69
    95     92     88     88     87     78     77     77     77     77     76     75     72
    81     64     68     71     62     72     60     77    101     63     75     95     82
    72     70     67     67     62     62     60     59     58     58     58     58     54
    83     90     76     70     74     73     84     80    107     87     88      0    110
    54     54     53     52     52     50     49     47     45     44     44     42     42
    78     91    100    105     94     85     89     93 (Other)
    42     42     41     41     41     39     39     39   5295
```

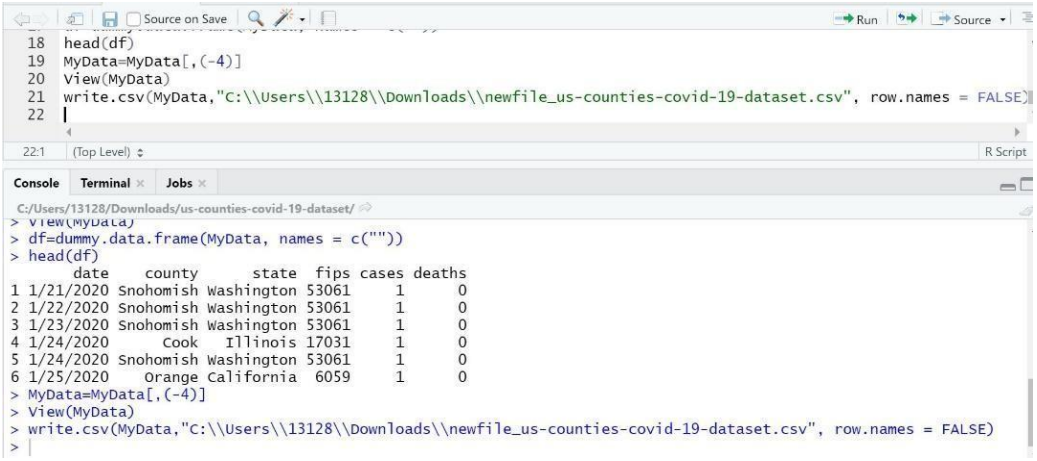
Performing the "***summary***" function on the attributes and finding if any null values.

In cases and death attributes there were no missing values and hence no pre-processing has been performed on them.



Here, we have created a dummy frame of the dataset to remove or to perform cleaning on the dataset.

"***df***" is the name of the variable set for the functions to be performed in the MyData dataset.



Using the "***head***" function our first 6 records are displayed.

As we went through our data, column "***fips***" made no contribution in the further process, and also was not contributing to this process. Hence we have removed it from the dataset using "***-4***" as it is the fourth column in our dataset.

Using "***write.csv***" we have created our new dataset file which is ready for the further process of transformation.

trial1.R* trial1.R* MyData

Filter

	date	county	state	cases	deaths
1	1/21/2020	Snohomish	Washington	1	0
2	1/22/2020	Snohomish	Washington	1	0
3	1/23/2020	Snohomish	Washington	1	0
4	1/24/2020	Cook	Illinois	1	0
5	1/24/2020	Snohomish	Washington	1	0
6	1/25/2020	Orange	California	1	0
7	1/25/2020	Cook	Illinois	1	0
8	1/25/2020	Snohomish	Washington	1	0
9	1/26/2020	Maricopa	Arizona	1	0
10	1/26/2020	Los Angeles	California	1	0
11	1/26/2020	Orange	California	1	0
12	1/26/2020	Cook	Illinois	1	0

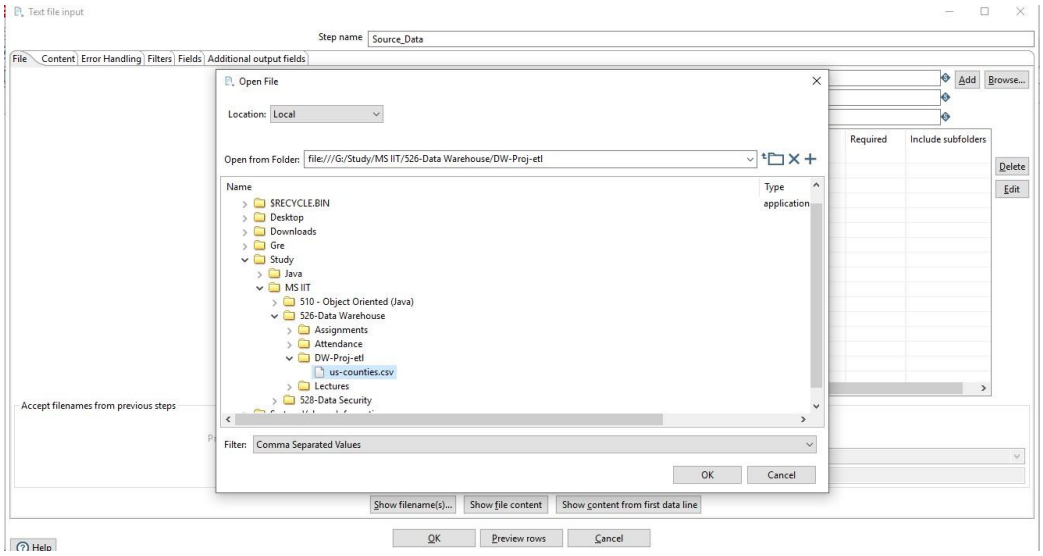
Showing 1 to 14 of 53,848 entries, 5 total columns

Using "***view***" we checked whether the column has been eliminated or not.

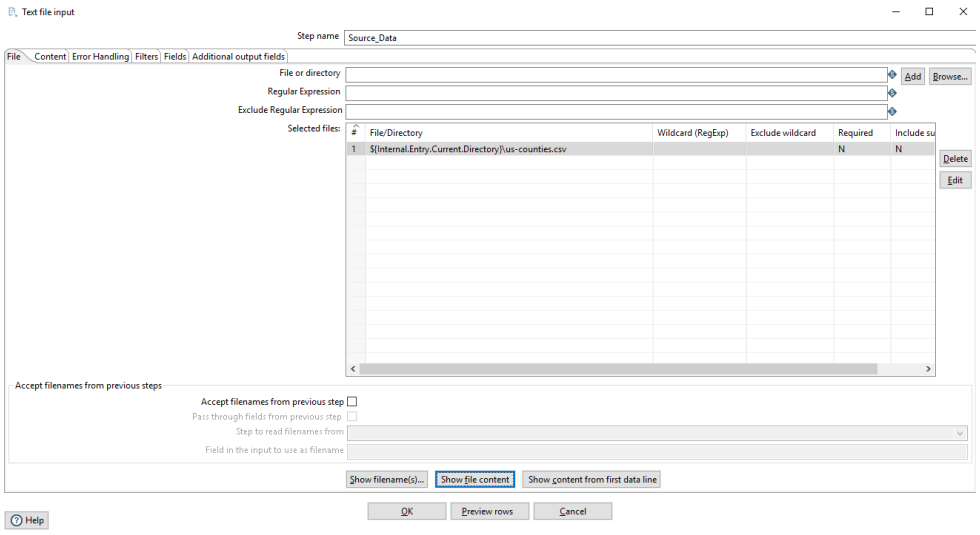
So, finally the fourth column of attribute "***fips***" has been removed and our preprocessed and cleaning a new .csv file has been completed.

Provisioning and Transformation

Steps performed for the Provisioning and Transformation of COVID-19 data.

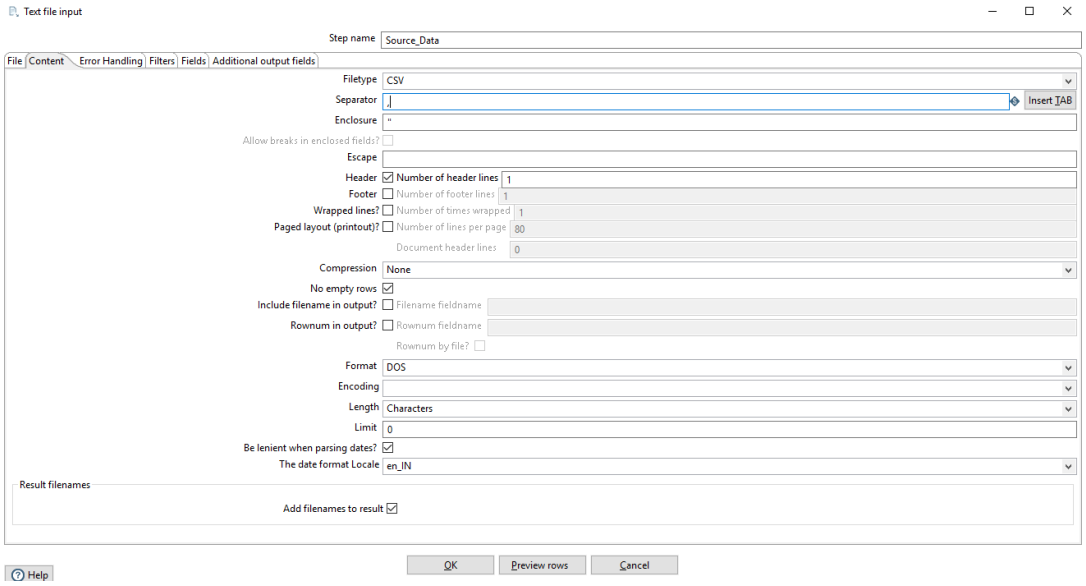


Text file input i.e. Source_Data

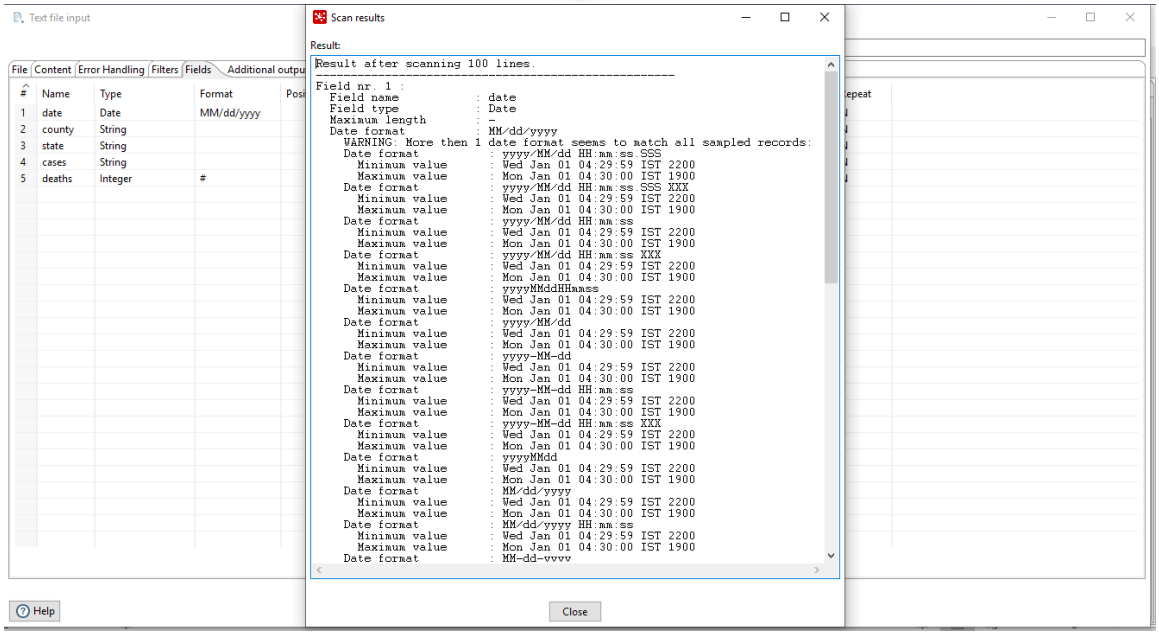


File Path replaced with an internal variable

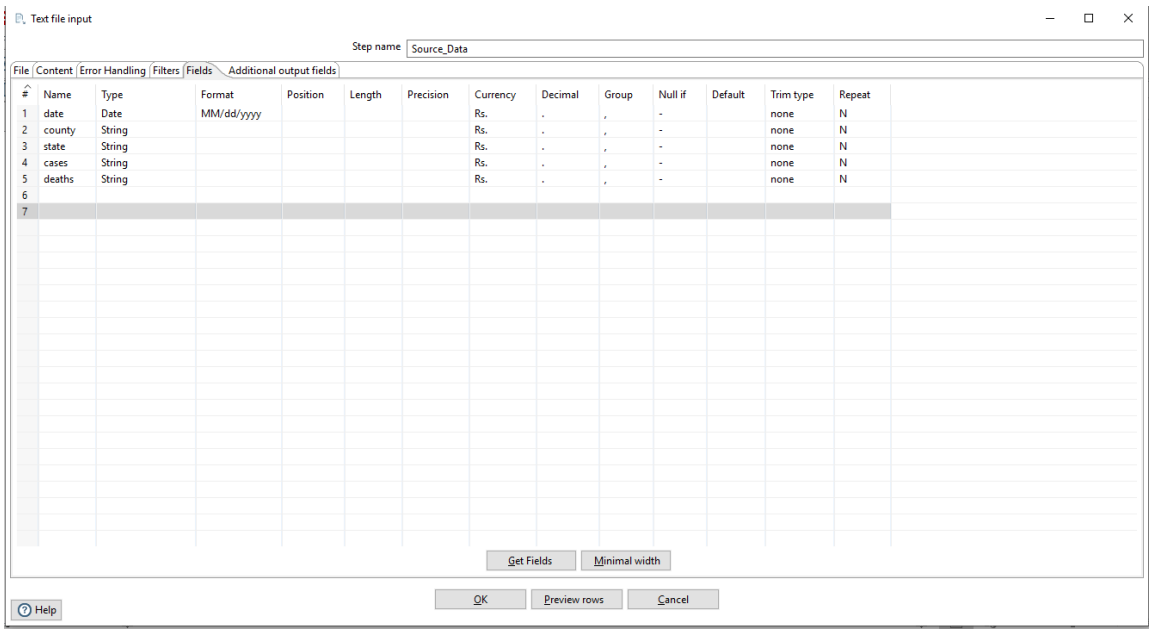
Preview if the file is loaded properly.



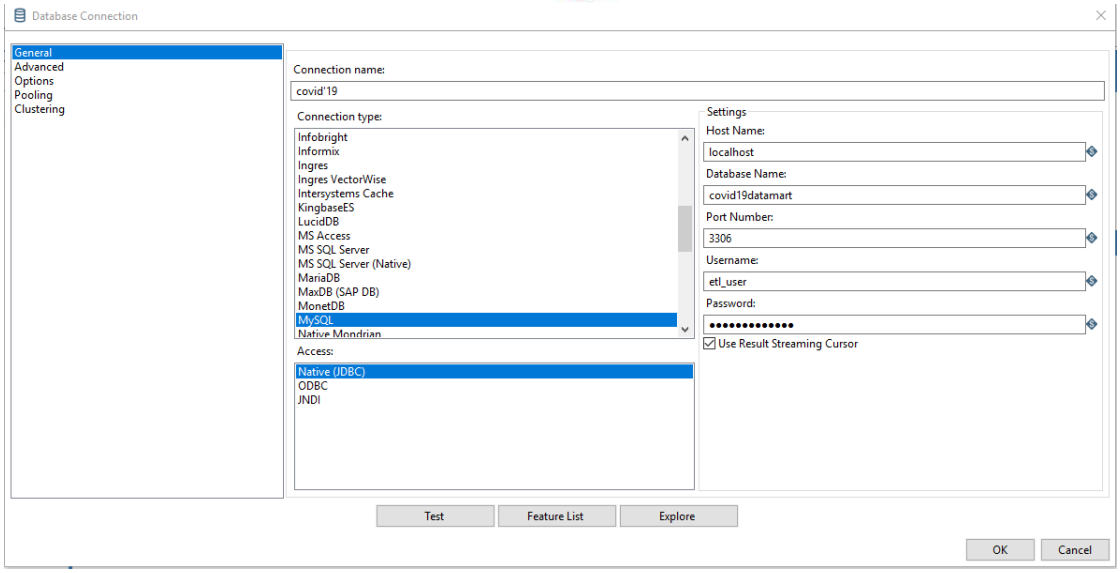
As the file is CSV format need “,” separator.



Default Field types of columns.



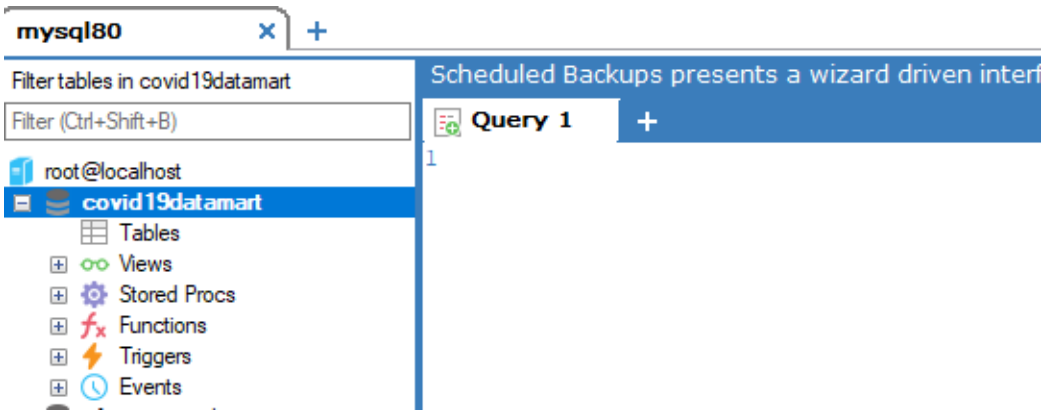
Changed type to string and removed other parameters w.r.t position, length, and precision.



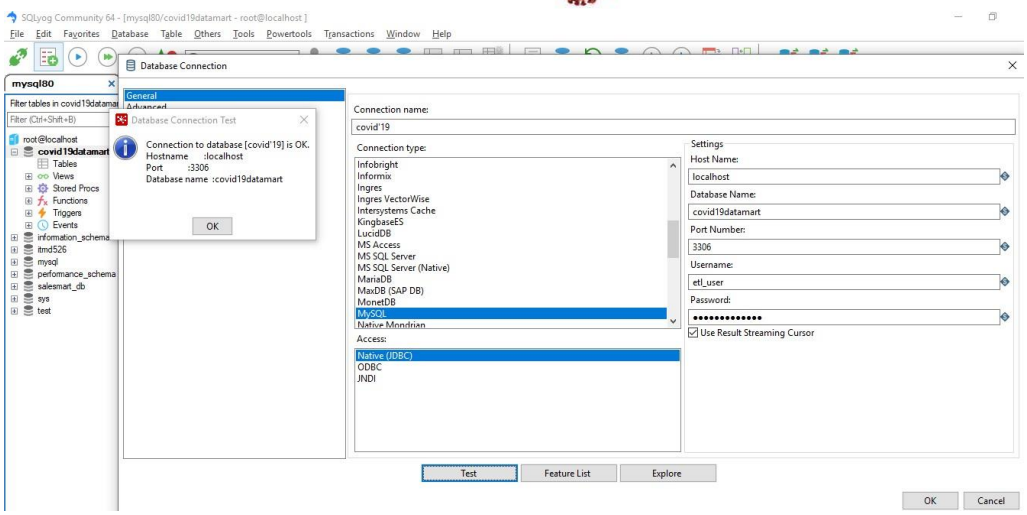
Then Created a new MySQL database connection for table output and covid19datamart in SQLYog.

Connection name: COVID'19

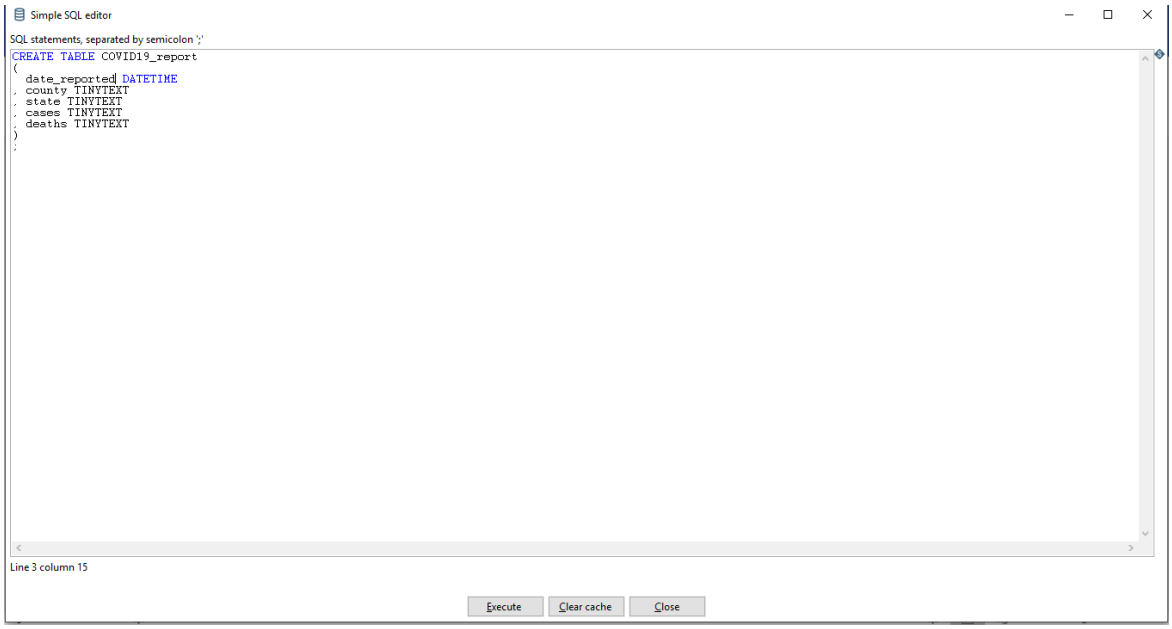
Database name: covid19datamart database.



Before creating a DB connection, we need DB in SQLyog as below:

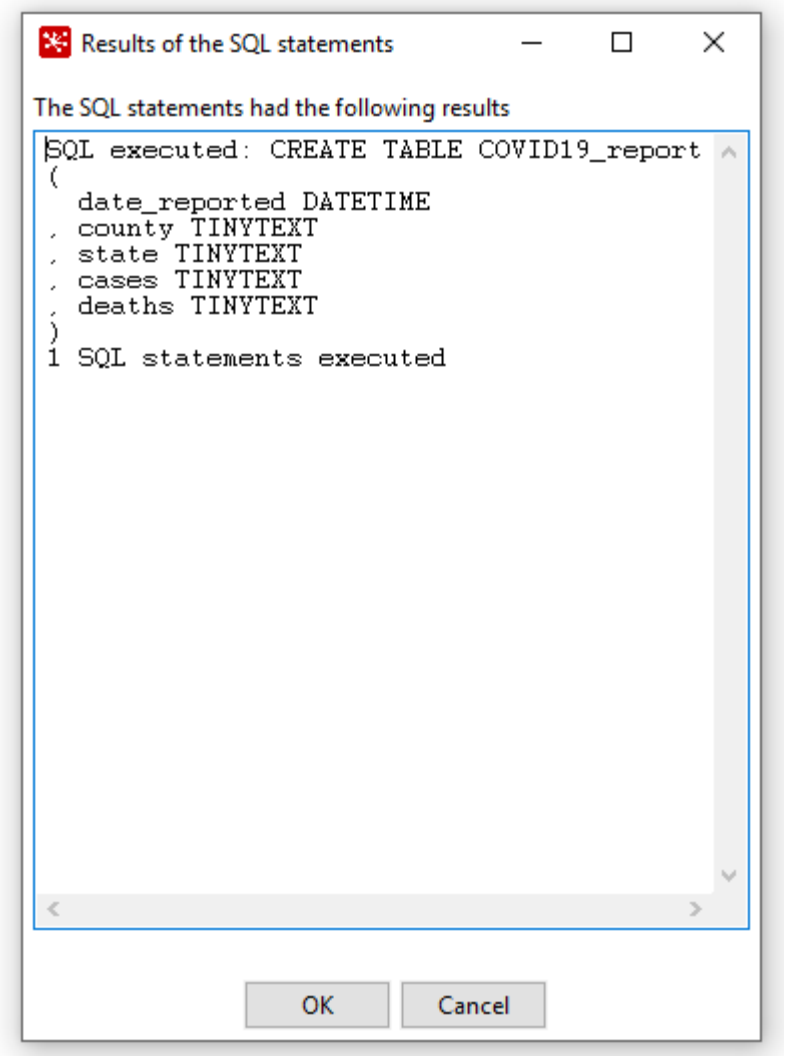


Tested Connection: OK

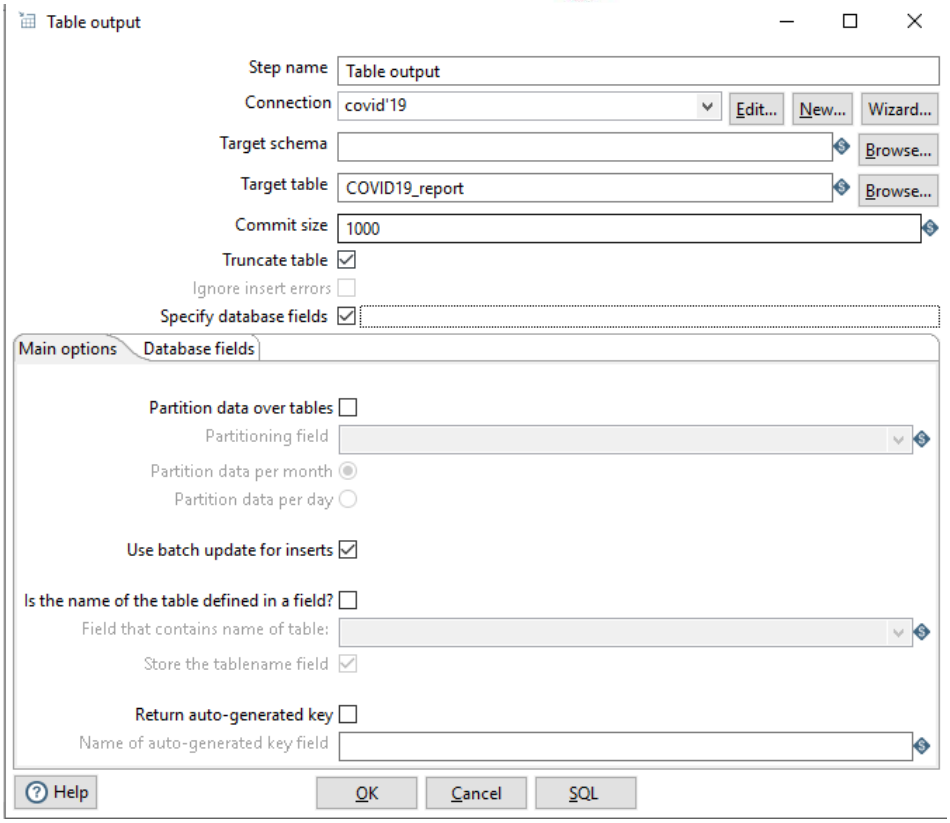


Creation of Table: covid19_report

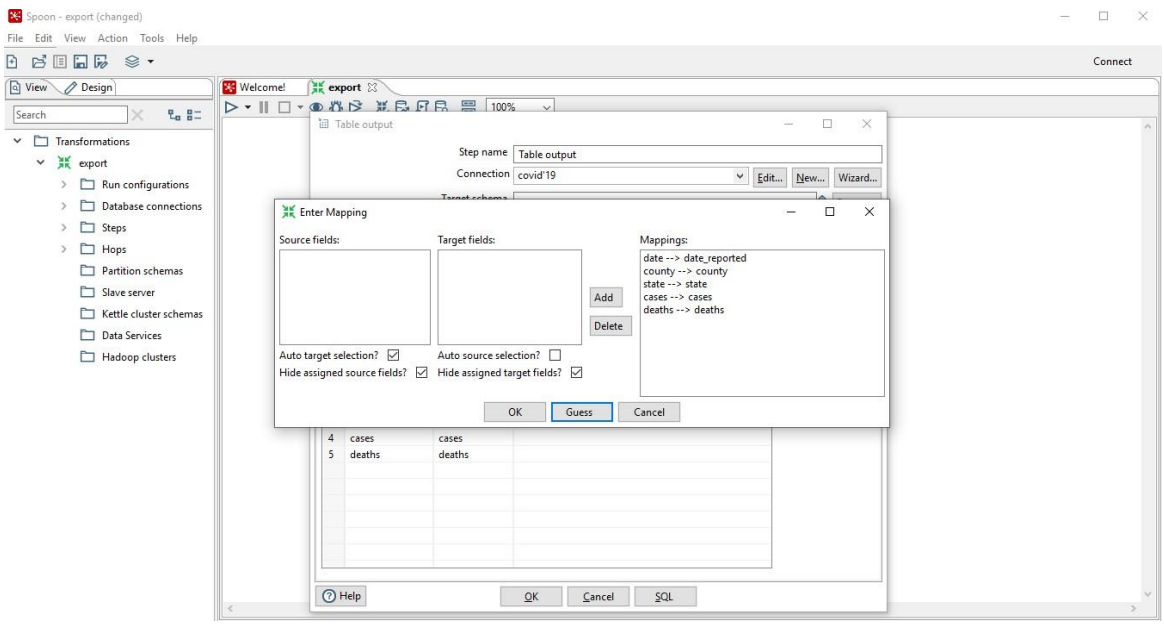
Here, date to date_reported to avoid sql_query error.



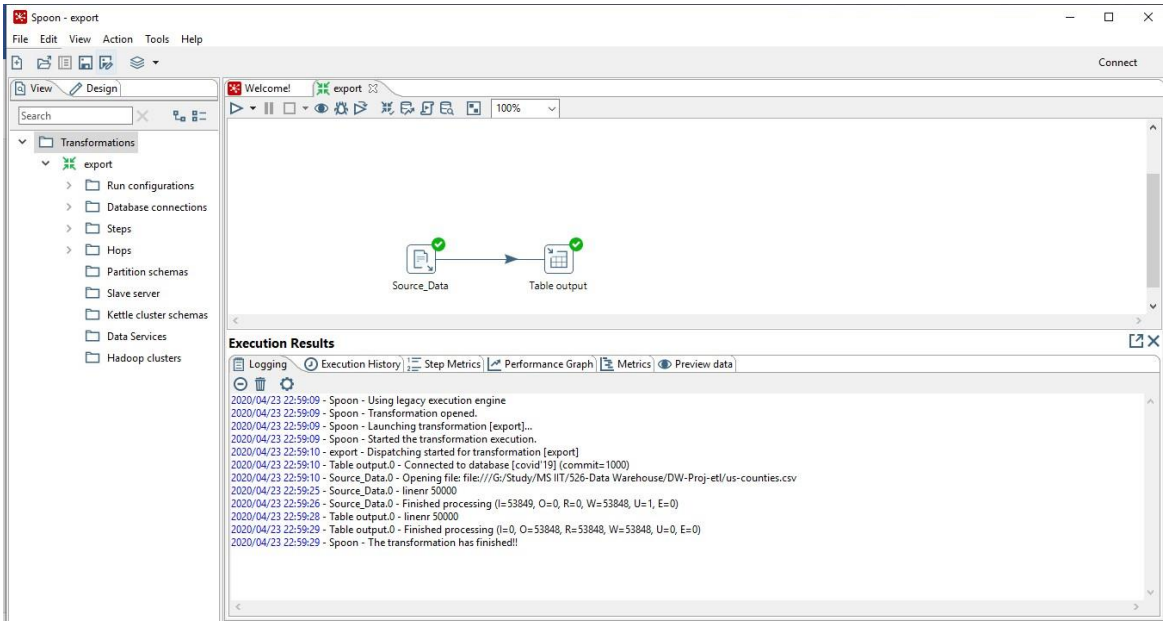
SQL query executed successfully.



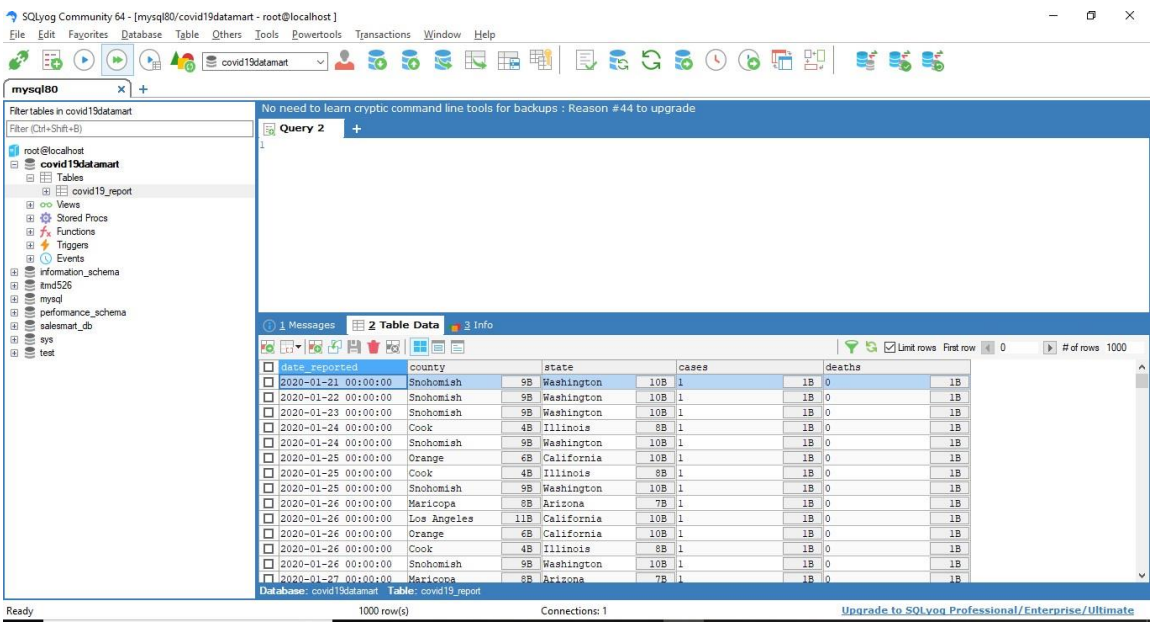
Check the Truncate Table so that every run will have fresh data.



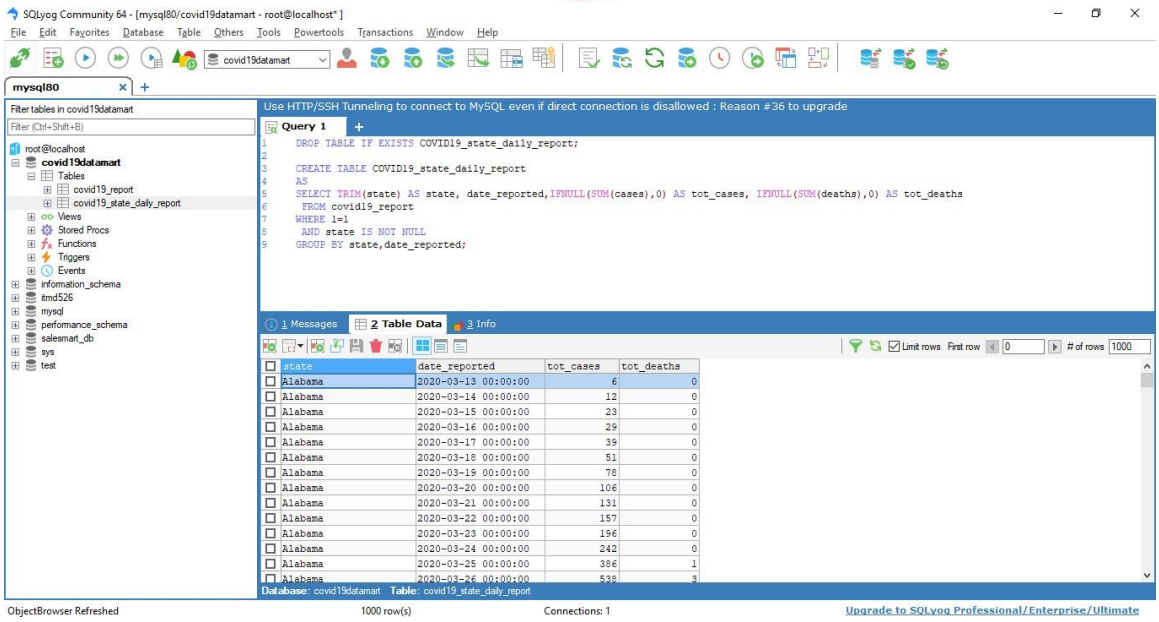
Mapped Source fields and Sql_Table fields i.e. target table “COVID19_report”



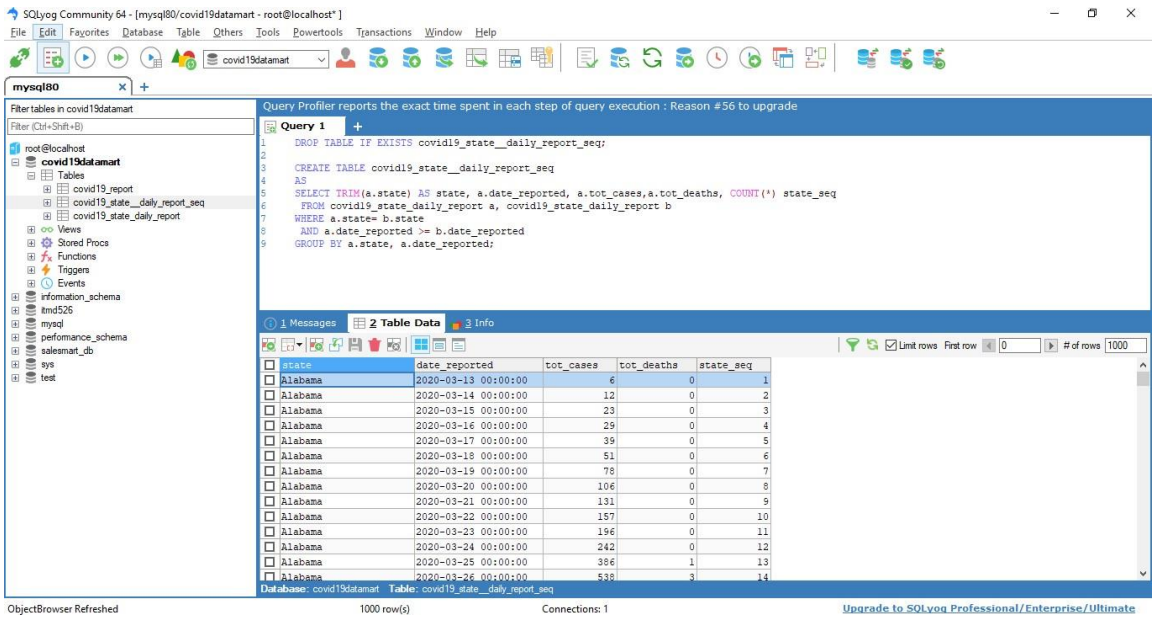
After running successful transformation source data is pushed into sql table.



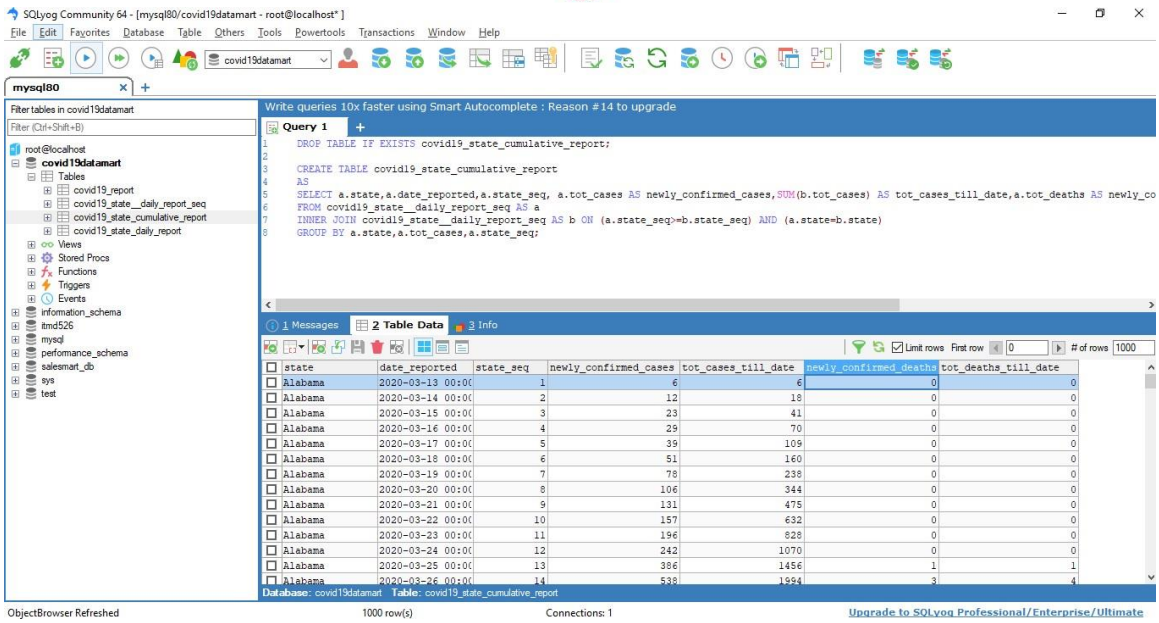
Data pushed into table using KTR file.



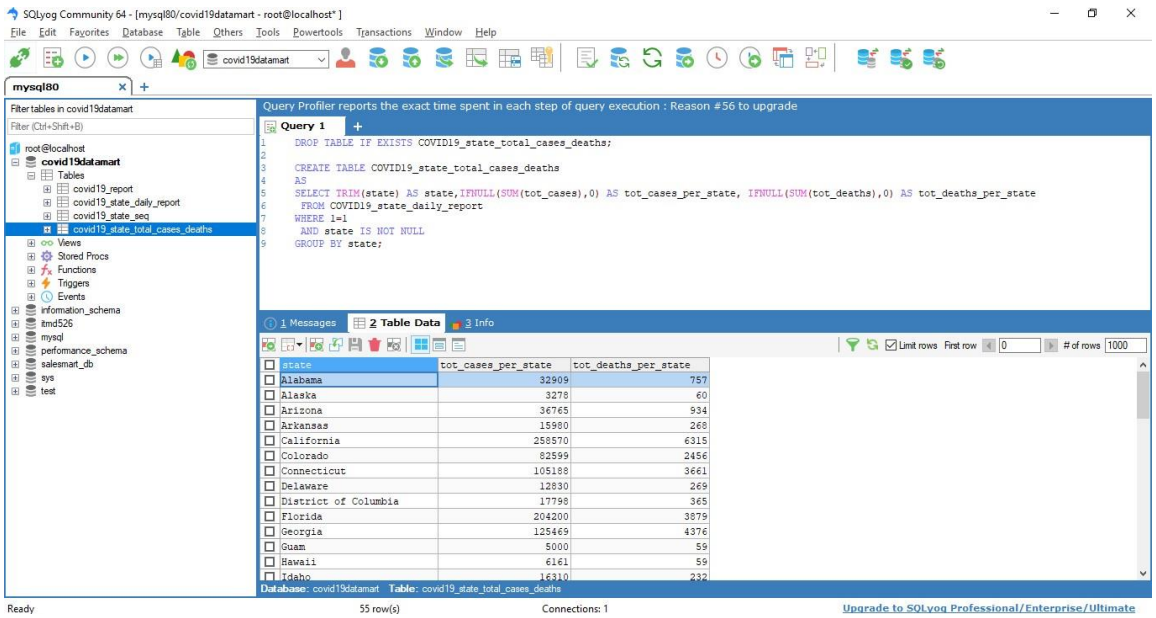
Above query will create new table “COVID19_state_daily_report” and grouped cases according to state and date_reported to give total count of cases/deaths confirmed on single day.



Above query will generate state_seq in covid19_state_daily_report_seq table.



With help of seq generated above query will help to give cumulative_report per state.Which is helpful for visualization purpose.



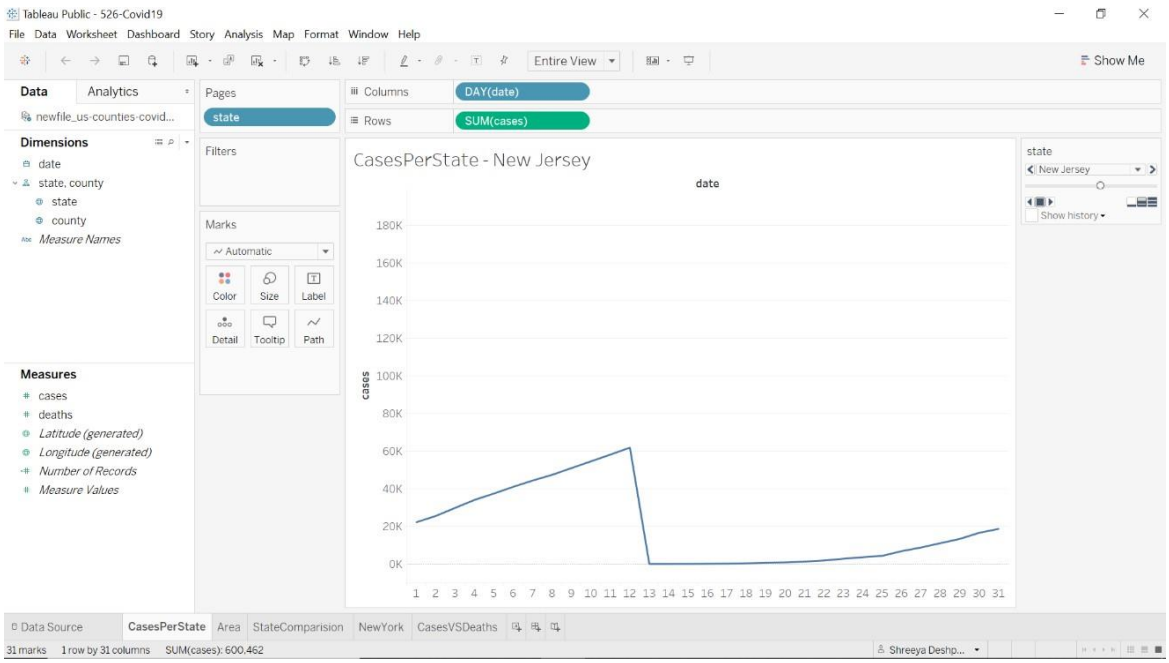
Above query will give total number of cases/deaths for each state till date.

This, data will also be required for Visualization purpose.

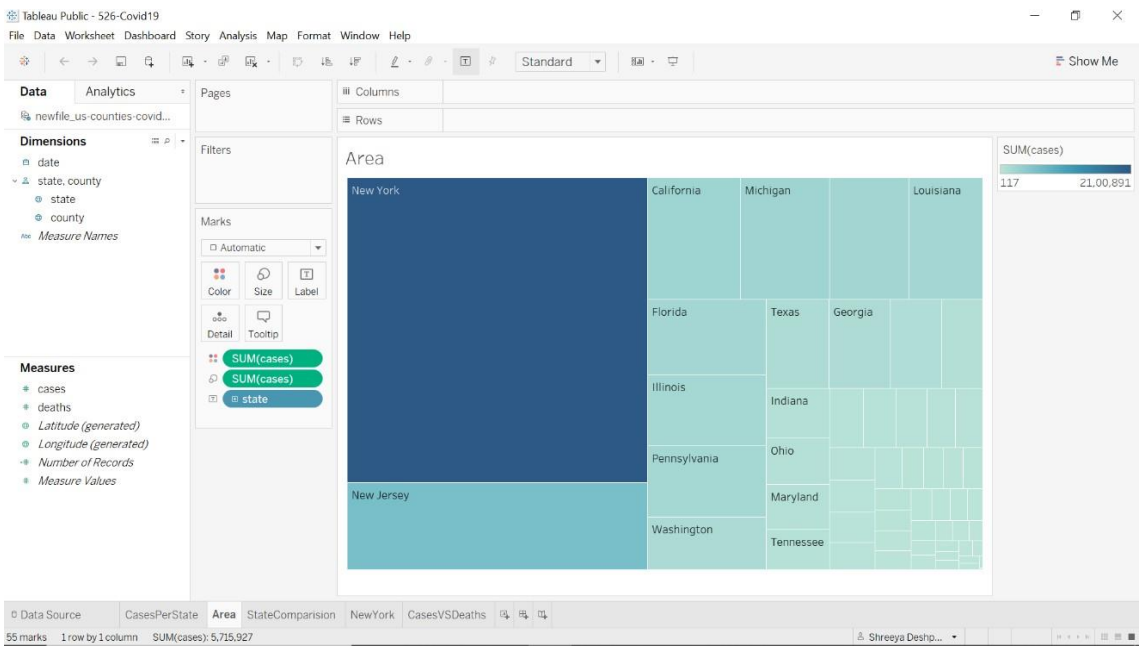
Dashboard Implementation (Visualization)

The visualization has been done in tableau. The CSV file generated after preprocessing was used for visualizing. The different graphs generated are shown below:

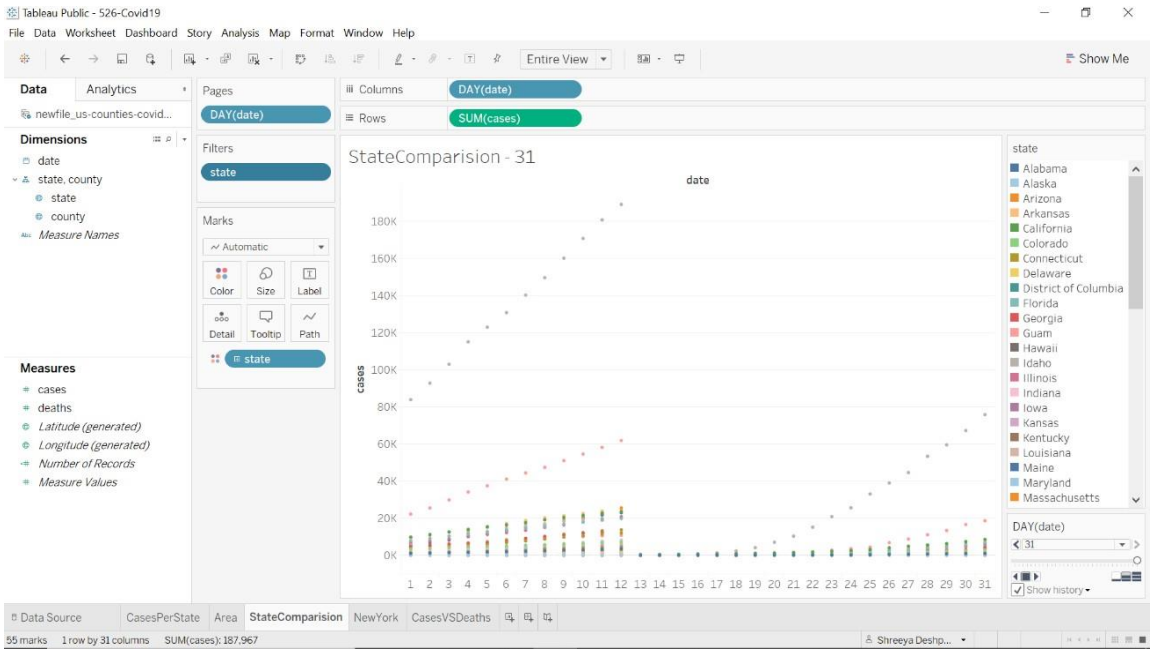
- Sum of Cases day-wise in different states:



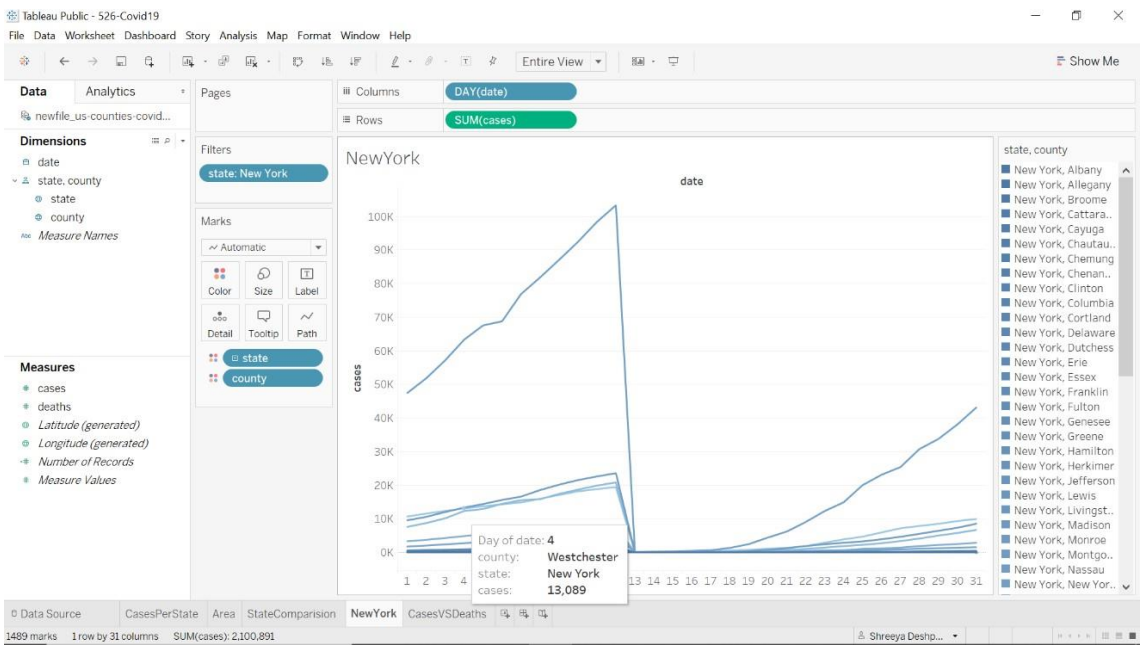
- Area Graph for comparing state-wise total cases:



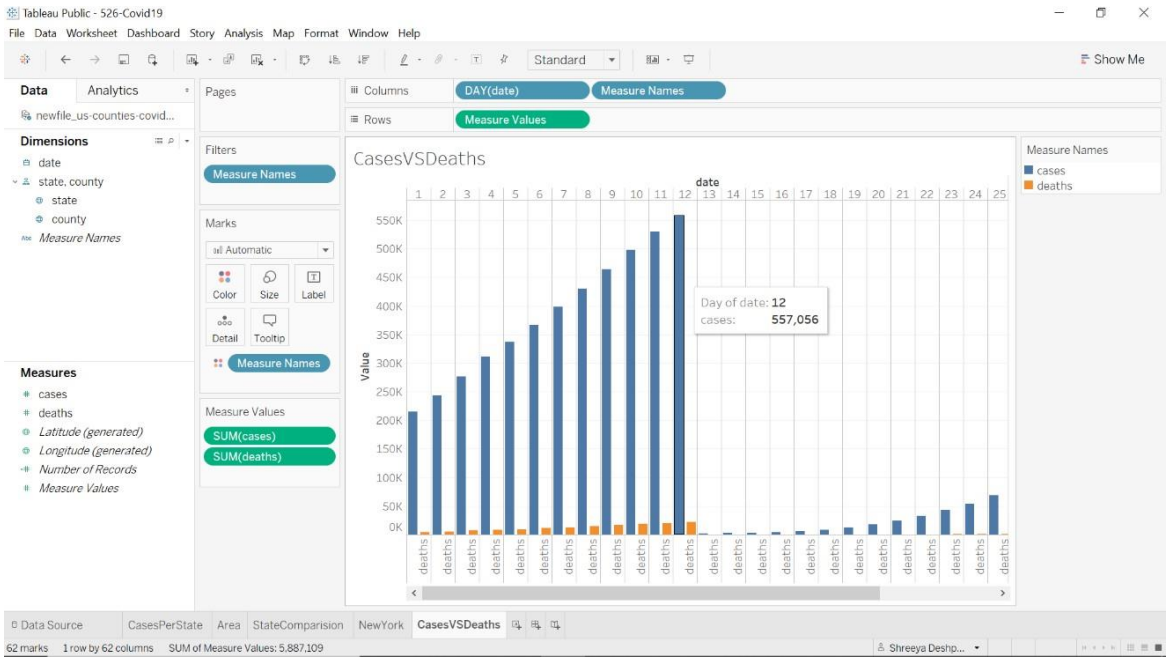
- Comparison of cases in all States day-wise:



- Countywise cases in New York State:



- Comparison of total cases and deaths day-wise:



Analysis of data

- Deaths across America spiked as Covid-19 began its spread, and many were never attributed to the new coronavirus. Our analysis includes the data between 21st January to12th April 2020.The data is from Kaggle website.
- Notable increases in deaths were seen in March and early April. This was especially true in New York and New Jersey, states hard-hit by the pandemic. Using data from this website we found that 15,000 excess deaths from March 1 to April 4.
- In the state of New York in county Westchester on day 4, we could see total 13,089 cases which are highest in comparison with all other county.
- After visualizing the data, we could see the number of cases day by day.
- We could measure and compare number of deaths per identified positive cases which is not too much.
- That means people are recovering from this covid 19 even though the cases are increasing on daily basis.