CIS 5200 – PROJECT TUTORIAL

BIG DATA DRIVEN – CONSUMER COMPLAINTS ANALYSIS

PROJECT TUTORIAL

INSTRUCTOR: Dr. Jongwook Woo

AUTHORS:

Dhwani Vaishnav

Manimozhi Neethinayagam

Akanksha S Khaire

Mansi Vivekanand Dhoke

Consumer Complaints Analysis Using Hive

Objectives:

The CFPB, U.S. government agency dedicated to making sure consumers are treated fairly by banks, lenders, and other financial institutions. It also maintains a public database of consumer complaints related to financial products and services, which can be used to inform policy decisions and regulatory actions.

We have analyzed and studied the complaint data from all companies in the USA, understood the year-on-year statistics and filtered the CFPB complaint data to include only complaints from companies in California. We also Analyzed sentiment and Ngram for consumer Narrative field. This tutorial explains the steps to perform the ETL processing using Hive and create visualizations to get detailed insights. The analysis can provide valuable insights into potential areas of concern for both consumers and financial institutions, leading to improved transparency and trust in the financial industry.

Introduction:

This tutorial will provide you with instructions on using HiveQL to analyze the CFPB dataset. Through this analysis, you can gain valuable insights into the issues that consumers face in the financial marketplace. These insights can be used to regulate financial products and services effectively and ensure that they comply with existing federal consumer financial laws.

In this tutorial, you'll learn how to use HADOOP CLUSTER to:

- Download and upload JSON file to HDFS
- Create Hive tables in HDFS
- Create Hive queries to analyze data
- Use Tableau to visualize the analyzed data
- Use Excel 3D Map for 3D visualization

Platform Spec:

- CLUSTER VERSION: Hadoop 3.1.2
- CLUSTER NODES: 5 (2 master nodes, 3 worker nodes)
- MEMORY SIZE: Memory Used – 388.64 (Memory in use - 367.68 GB, Memory Remaining – 20.96 GB
- CPU SPEED: 1995.312 MHz

To Get the cluster details, execute below commands:

- To know the CLUSTER VERSION: hdfs version

```
-bash-4.2$ hdfs version
Hadoop 3.1.2
Source code repository ssh://git@bitbucket.oci.oraclecorp.com:7999/bdcs/apache_bigtop.git -r 4100eb8d8581c4328601079ff5af522f95e9977f
Compiled by root on 2023-02-27T08:26Z
Compiled with protoc 2.5.0
From source with checksum b367ca15864aef16725a3035859c9ece
This command was run using /usr/odh/1.1.5/hadoop/hadoop-common-3.1.2.jar
```

- To know the CLUSTER NODES: yarn node –list –all

```
-bash-4.2$ yarn node --list --all
23/05/04 21:47:48 INFO client.RMProxy: Connecting to ResourceManager at bigdaimn0.sub03291929060.trainingvcn.oraclevcn.com/10.1.0.38:8050
23/05/04 21:47:48 INFO client.AHSProxy: Connecting to Application History server at bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com/10.1.0.149:10200
Total Nodes:3
         Node-Id             Node-State Node-Http-Address       Number-of-Running-Containers
bigdaiwn2.sub03291929060.trainingvcn.oraclevcn.com:45454              RUNNING bigdaiwn2.sub03291929060.trainingvcn.oraclevcn.com:8042                    0
bigdaiwn1.sub03291929060.trainingvcn.oraclevcn.com:45454              RUNNING bigdaiwn1.sub03291929060.trainingvcn.oraclevcn.com:8042                    5
bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:45454              RUNNING bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:8042                    1
```

- To know the MEMORY SIZE: hdfs dfsadmin -report

```
-bash-4.2$ hdfs dfsadmin -report
Configured Capacity: 419520548352 (390.71 GB)
Present Capacity: 417301338222 (388.64 GB)
DFS Remaining: 22508090343 (20.96 GB)
DFS Used: 394793247879 (367.68 GB)
DFS Used%: 94.61%
```

- To know the CPU SPEED: lscpu | grep ‘MHz”

```
-bash-4.2$ lscpu | grep "MHz"
CPU MHz:                       1995.312
```

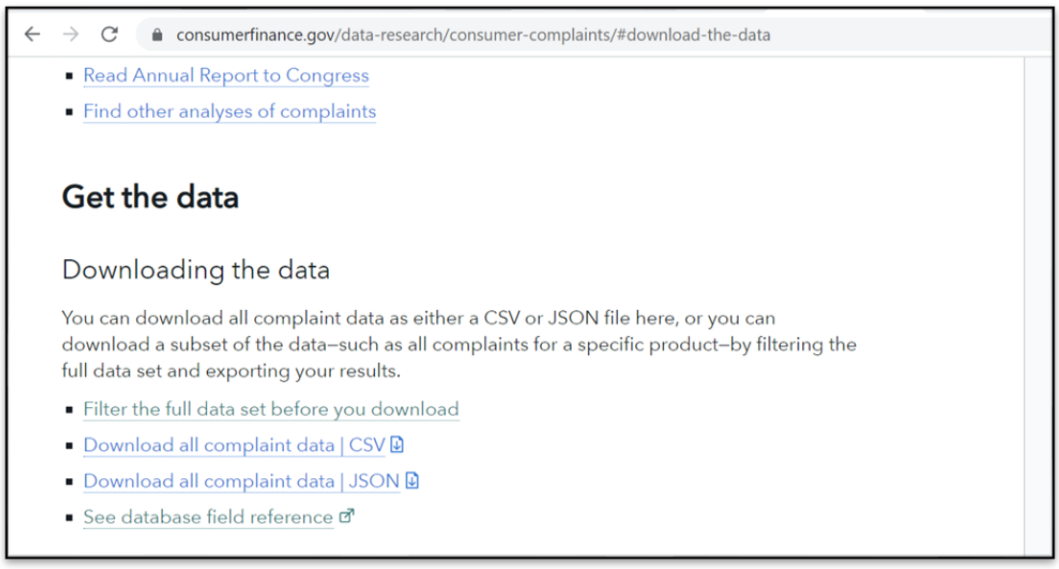
Dataset Details:

- DATASET NAME: Consumer Financial Protection Bureau Dataset
- DATASET URL: https://catalog.data.gov/dataset/consumer-complaint-database
- TOTAL SIZE: 3.6 GB
- COUNTRY CONSIDERED: USA
- NUMBER OF FILES: 1
- FILE FORMAT: JSON

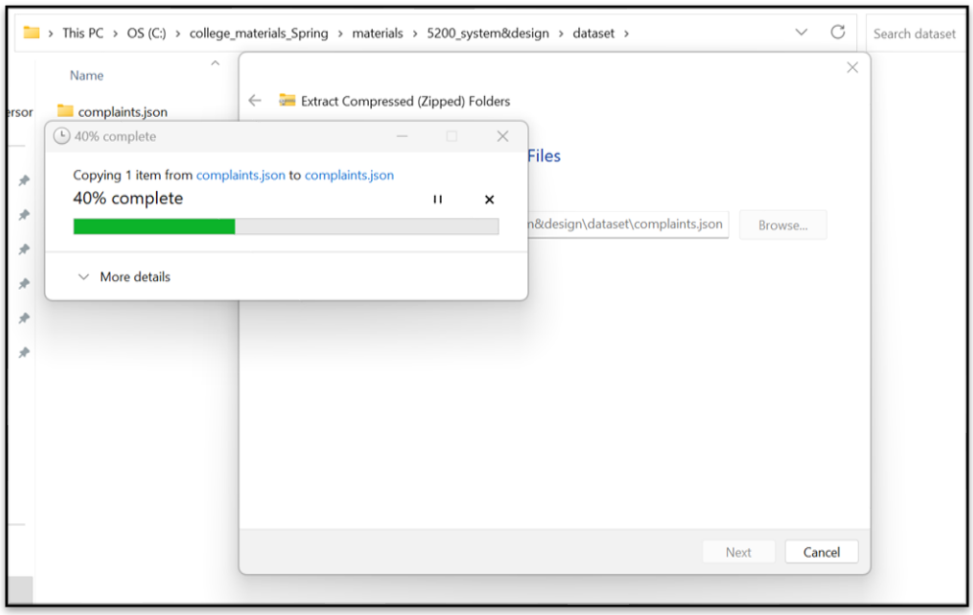
Step 1: Download the Dataset

This step is to get data manually to the local system. Following are the steps to download:

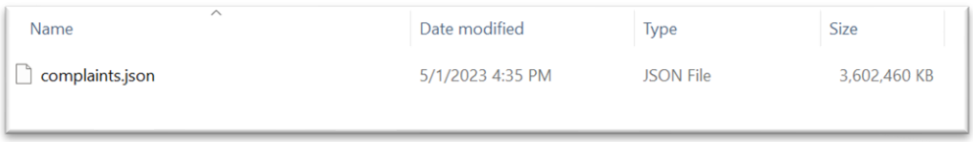
1. CFPB Dataset – Download Dataset to local machine. You should have complaints.json in your local machine.
   Note: This file contents can change depending on when it was downloaded since the complaints are registered daily.



1. Extract the complaints.json using zip(Extract All)



2. We have now extracted the json file



3. Now, you have to transfer the single json file from the local machine to HADOOP Cluster, which is shown in next section.

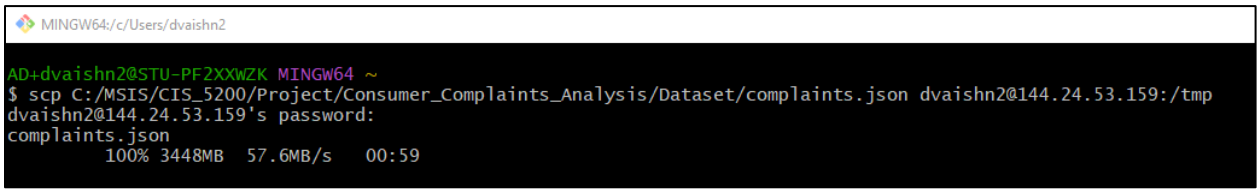
Step 2: Upload Files to Hadoop File system (HDFS)

a. Copy dataset using scp to the linux “tmp” directory.

Open a Gitbash session (Session 1) and execute the following command:

Remember to use your username and IP address of the cluster given.

```
scp C:/MSIS/CIS_5200/Project/Consumer_Complaints_Analysis/Dataset/complaints.json dvaishn2@144.24.53.159:/tmp
```



b. Additionally, for the CFPB complaints narrative sentiment analysis, we have used dictionary.tsv file which we have downloaded from amazon S3 using wget shell command as following:
Open new Gitbash session (session 2)
Note: You need to remotely access your Oracle Big Data server of Oracle Cloud using ssh. Your CalStateLA username(dvaishn2) should be a username & password to connect to the Hadoop cluster as follows:

ssh dvaishn2@144.24.53.159

wget -O dictionary.tsv https://github.com/dalgual/aidatasci/raw/master/data/bigdata/dictionary.tsv

c. Run the following HDFS commands to create and list "**5200_Complaints_DS” & “5200_Project_Tables**" directory in HDFS:

```
hdfs dfs -mkdir 5200_Complaints_DS

hdfs dfs -mkdir 5200_Project_Tables

hdfs dfs -mkdir 5200_Project_Tables/DV

hdfs dfs -mkdir 5200_Project_Tables/DV/Dictionary

hdfs dfs -mkdir /user/dvaishn2/5200_Project_Tables/Mani
```

```
hdfs dfs -mkdir /user/dvaishn2/5200_Project_Tables/AK
hdfs dfs -mkdir /user/dvaishn2/5200_Project_Tables/MD
hdfs dfs -ls
```

```
-bash-4.2$ hdfs dfs -ls
Found 8 items
drwx------   - dvaishn2 hdfs          0 2023-04-16 00:28 .Trash
drwxr-xr-x   - dvaishn2 hdfs          0 2023-04-06 01:04 .hiveJars
drwxr-xrwx   - dvaishn2 hdfs          0 2023-04-16 00:44 5200_Complaints_DS
drwxr-xr-x   - dvaishn2 hdfs          0 2023-04-16 00:47 5200_Project_Tables
drwxr-xr-x   - dvaishn2 hdfs          0 2023-04-13 01:44 dualcore
drwxr-xr-x   - dvaishn2 hdfs          0 2023-04-13 01:44 ratings
drwxr-xr-x   - dvaishn2 hdfs          0 2023-04-06 01:46 tmp
drwxr-xr-x   - dvaishn2 hdfs          0 2023-04-13 21:25 user
```

```
hdfs dfs -ls 5200_Project_Tables/
```

```
-bash-4.2$ hdfs dfs -ls 5200_Project_Tables
Found 5 items
drwxr-xr-x   - akhaire3 hdfs          0 2023-04-21 19:26 5200_Project_Tables/AK
drwxr-xr-x   - akhaire3 hdfs          0 2023-04-21 18:13 5200_Project_Tables/AK
drwxr-xrwx   - dvaishn2 hdfs          0 2023-05-02 00:40 5200_Project_Tables/DV
drwxr-xr-x   - mdhoke   hdfs          0 2023-04-21 18:36 5200_Project_Tables/MD
drwxr-xrwx   - mneethi  hdfs          0 2023-04-21 22:52 5200_Project_Tables/Mani
```

d. Once files are transferred to linux in step a, the json file can be transferred from linux “tmp” location to the "5200_Complaints_DS” directory in HDFS using the "-put" command.

```
hdfs dfs -put /tmp/complaints.json 5200_Complaints_DS
hdfs dfs -ls 5200_Complaints_DS/
```

Confirming the files are transferred to HDFS:

```
-bash-4.2$ hdfs dfs -ls 5200_Complaints_DS/
Found 1 items
-rw-r--rw-   3 dvaishn2 hdfs 3615569889 2023-04-15 20:11 5200_Complaints_DS/complaints.json
```

e. Also, lets transfer the dictionary.tsv from linux to the "5200_Project_Tables/DV/dictionary” directory in HDFS using the "-put" command.

```
hdfs dfs -put dictionary.tsv 5200_Project_Tables/DV/dictionary/
```

```
-bash-4.2$ hdfs dfs -put dictionary.tsv 5200_Project_Tables/DV/dictionary/
```

Confirming the files are transferred to HDFS:

```
hdfs dfs -ls 5200_Project_Tables/DV/dictionary
```

```
-bash-4.2$ hdfs dfs -ls 5200_Project_Tables/DV/dictionary
Found 1 items
-rw-r--r--   3 dvaishn2 hdfs     308921 2023-05-02 00:41 5200_Project_Tables/DV/dictionary/dictionary.tsv
```

f. It is then necessary to grant permission to the "5200_Complaints_DS, 5200_Project_Tables" directory to other team members, ensuring that everyone can access and collaborate on the dataset.
By following these steps, all team members can work collaboratively on the dataset in a secure and efficient manner.

Changing the permission for the use of teammates:

```
hdfs dfs -chmod -R o+rw /user/dvaishn2/5200_Complaints_DS/
```

```
-bash-4.2$ hdfs dfs -chmod -R o+rw /user/dvaishn2/5200_Complaints_DS/
-bash-4.2$
```

```
hdfs dfs -chmod -R o+rw /user/dvaishn2/5200_Project_Tables/
```

```
-bash-4.2$ hdfs dfs -chmod -R o+rw /user/dvaishn2/5200_Project_Tables/
```

Confirming the access by other teammates:

```
-bash-4.2$ pwd
/home/mneethi
-bash-4.2$ hdfs dfs -cat /user/dvaishn2/5200_Complaints_DS/complaints.json | head -2;
[
{"date_received": "2022-01-26", "product": "Credit reporting, credit repair services, or other personal consumer reports", "sub_product": "Credit reporting", "issue": "Inco
rrect information on your report", "sub_issue": "Information belongs to someone else", "complaint_what_happened": "", "company_public_response": "Company has responded to t
he consumer and the CFPB and chooses not to provide a public response", "company": "TRANSUNION INTERMEDIATE HOLDINGS, INC.", "state": "VA", "zip_code": "20166", "tags": "",
 "consumer_consent_provided": "Consent not provided", "submitted_via": "Web", "date_sent_to_company": "2022-01-26", "company_response": "Closed with explanation", "timely":
 "Yes", "consumer_disputed": "N/A", "complaint_id": "5152476"},
```

Step 3: Creating Hive Queries

1. Setting up the Beeline environment to run the hive queries

Instructions:
Open another Gitbash session to get into **beeline** Command Line Interface using the following command to run hive queries. Beeline is for multiple users' access to Hive Server 2 of a Hadoop cluster.

Beeline command:

a. Open new Gitbash session (session 3)

Note: You need to remotely access your Oracle Big Data server of Oracle Cloud using ssh. Your CalStateLA username(dvaishn2) should be a username & password to connect to the Hadoop cluster as follows:

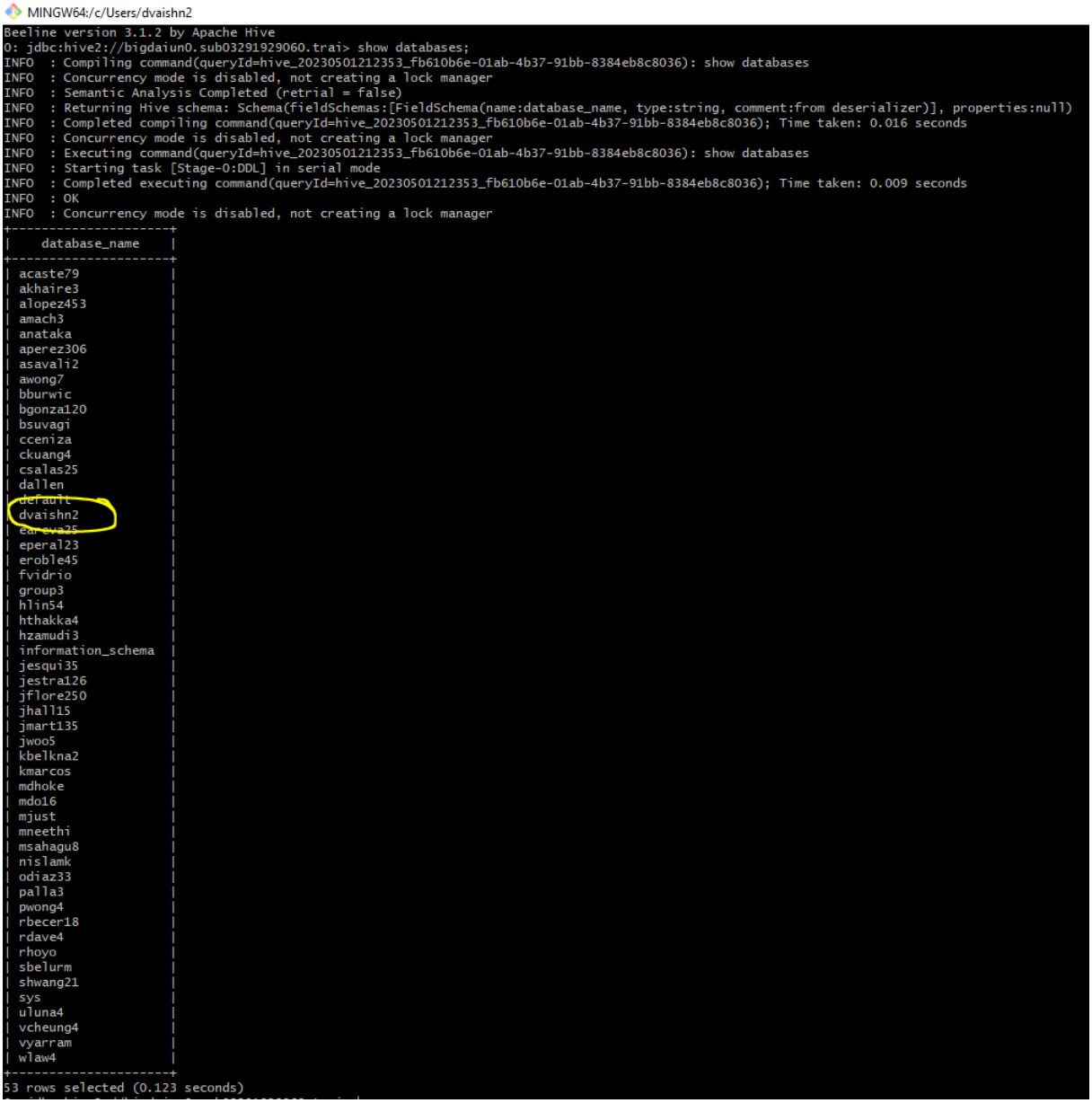
ssh dvaishn2@144.24.53.159

$ beeline

b. Now you must create your database with your username to separate your tables from other users. For example, the user dvaishn2 should run the following:

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE DATABASE IF NOT EXISTS dvaishn2;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show databases;



0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use dvaishn2;

Note: use your database name instead of dvaishn2

2. Creating an External Table in Hive to Query Data Stored in HDFS

Description:

The following Hive statement creates an **External Table** that allows Hive to query data stored in HDFS. External tables preserve the data in the original file format while allowing the Hive to perform queries against the data within the file.

The Hive statements below create an external table that allows the user to query and analyze the JSON responses stored in text files in HDFS.

The table is named "raw_complaints" and has one column named "json_response" with data type STRING.

The location is "/user/dvaishn2/5200_Complaints_DS/", which is the directory path where the text files containing the JSON responses will be stored.

Instructions:

a. Run following hive query on Beeline:

```
CREATE EXTERNAL TABLE IF NOT EXISTS raw_complaints(json_response STRING)

STORED AS TEXTFILE

LOCATION '/user/dvaishn2/5200_Complaints_DS/';
```

b. Confirm if the table is created using following command:

```
show tables;
```

c. Execute following query to display the first 2 rows of "raw_complaints" table:

```
SELECT * FROM raw_complaints LIMIT 2;
```

```
+----------------------------------------------------+

|          raw_complaints.json_response              |

+----------------------------------------------------+

| [                                                  |
```

| {"date_received": "2022-01-26", "product": "Credit reporting, credit repair services, or other personal consumer reports", "sub_product": "Credit reporting", "issue": "Incorrect information on your report", "sub_issue": "Information belongs to someone else", "complaint_what_happened": "", "company_public_response": "Company has responded to the consumer and the CFPB and chooses not to provide a public response", "company": "TRANSUNION INTERMEDIATE HOLDINGS, INC.", "state": "VA", "zip_code": "20166", "tags": "", "consumer_consent_provided": "Consent not provided", "submitted_via": "Web", "date_sent_to_company": "2022-01-26", "company_response": "Closed with explanation", "timely": "Yes", "consumer_disputed": "N/A", "complaint_id": "5152476"}, |

+--+

2 rows selected (0.328 seconds)

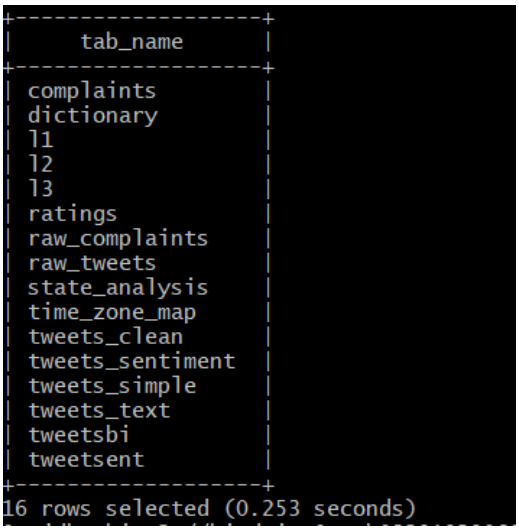
d. Execute following query to create the base table to which the data from raw_complaints are inserted:

Description:
The purpose of this table is to select data from the "json_response" column of the "raw_complaints" table and transform it into structured data using the "get_json_object" function to extract specific fields from the JSON data.

```
DROP TABLE IF EXISTS complaints;


CREATE TABLE complaints(
c_date_received string, c_product string, c_sub_product string,
c_issue string, c_sub_issue string, c_complaint_narrative string, c_company_public_response string,
c_company string, c_state string, c_zip int,
c_tags string, c_consumer_consent string, c_submitted_via string,
c_date_sent_to_company string, c_company_response_to_consumer string,
c_timely_response string, c_consumer_disputed string, c_complaint_id bigint
);
```

e. Confirm if the table is created using following command:

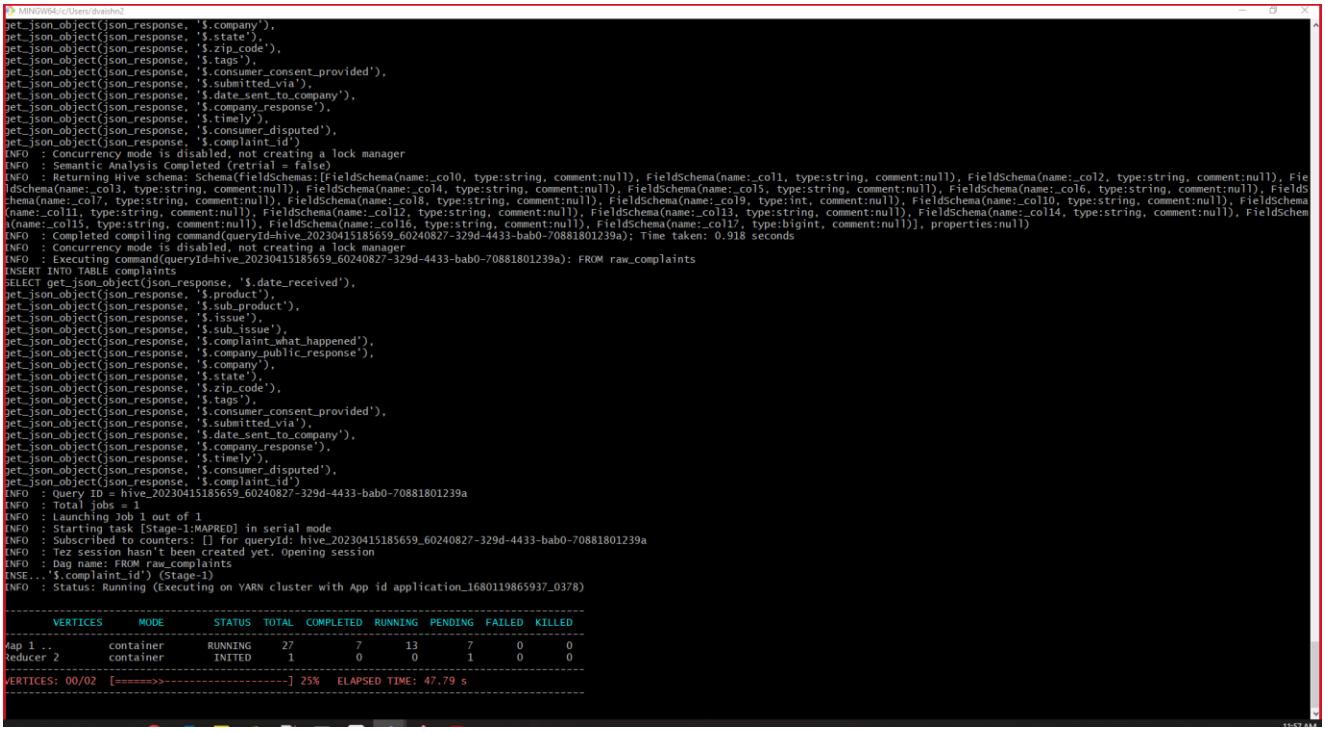
```
show tables;
```



f. **Execute the following query to insert the structured data into the "complaints" table with each field mapped to its corresponding column:**

```
FROM raw_complaints

INSERT INTO TABLE complaints

SELECT CAST(to_date(from_unixtime(unix_timestamp(get_json_object(json_response,
'$.date_received'), 'yyyy-MM-dd'))) AS date),

get_json_object(json_response, '$.product'),

get_json_object(json_response, '$.sub_product'),

get_json_object(json_response, '$.issue'),

get_json_object(json_response, '$.sub_issue'),

get_json_object(json_response, '$.complaint_what_happened'),

get_json_object(json_response, '$.company_public_response'),

get_json_object(json_response, '$.company'),

get_json_object(json_response, '$.state'),

get_json_object(json_response, '$.zip_code'),

get_json_object(json_response, '$.tags'),

get_json_object(json_response, '$.consumer_consent_provided'),

get_json_object(json_response, '$.submitted_via'),

get_json_object(json_response, '$.date_sent_to_company'),

get_json_object(json_response, '$.company_response'),
```

```
get_json_object(json_response, '$.timely'),
get_json_object(json_response, '$.consumer_disputed'),
get_json_object(json_response, '$.complaint_id');
```




```
SELECT * FROM complaints LIMIT 2;
```

```
+------------------------------+-------------------------------------------------------+-----------------------------------------
----+-----------------------------------------------+-----------------------------------------------+---------------
-----------------------------------+-----------------------------------------------+-------------------------------
--+--------------------+------------------+------------------+------------------------------+-----------------------
----+-----------------------------------+-------------------------------------------+------------------------------+-----
--------------------------+--------------------------+
```

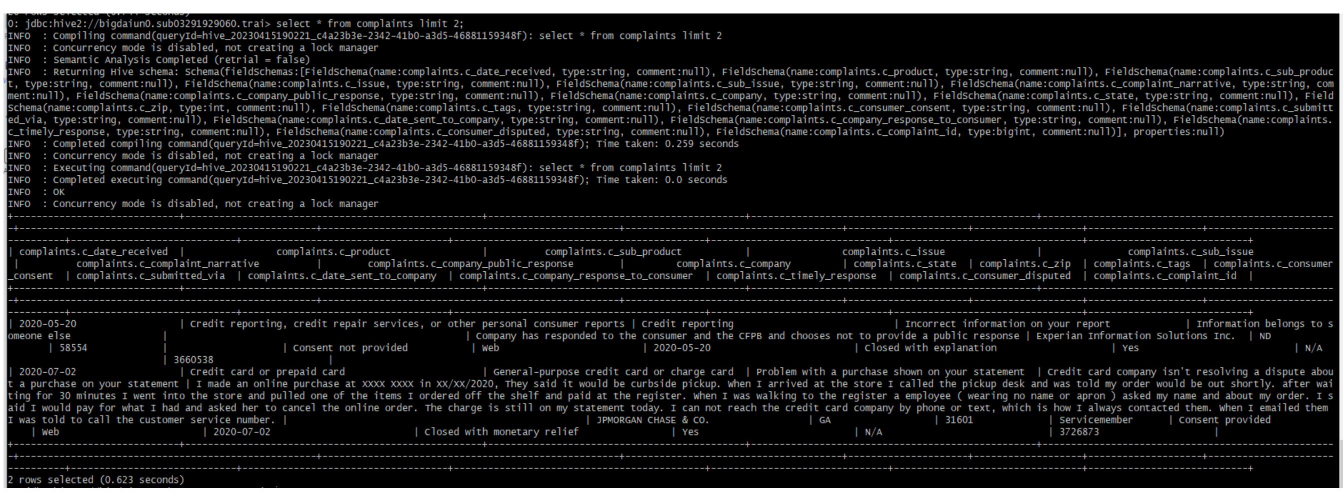
| complaints.c_date_received | complaints.c_product | complaints.c_sub_product | complaints.c_issue | complaints.c_sub_issue | complaints.c_complaint_narrative | complaints.c_company_public_response | complaints.c_company | complaints.c_state | complaints.c_zip | complaints.c_tags | complaints.c_consumer_consent | complaints.c_submitted_via | complaints.c_date_sent_to_company | complaints.c_company_response_to_consumer | complaints.c_timely_response | complaints.c_consumer_disputed | complaints.c_complaint_id |

```
+----------------------------+-------------------------------------------------+----------------------------------------
----+-----------------------------------------------+-------------------------------------------------+---------------
---------------------------------+-------------------------------------------------+----------------------------------
--+---------------------+------------------+-------------------+------------------------------+------------------------
----+-----------------------------------+------------------------------------------+-----------------------------+-----
---------------------------+---------------------------+
```

| 2020-05-20 | Credit reporting, credit repair services, or other personal consumer reports | Credit reporting | Incorrect information on your report | Information belongs to someone else | | Company has responded to the consumer and the CFPB and chooses not to provide a public response | Experian Information Solutions Inc. | ND | 58554 | | Consent not provided | Web | 2020-05-20 | Closedwith explanation | Yes | N/A | 3660538 |

| 2020-07-02 | Credit card or prepaid card | General-purpose credit card or charge card | Problem with a purchase shown on your statement | Credit card company isn't resolving a dispute about a purchase on your statement | I made an online purchase at XXXX XXXX in XX/XX/2020, They said it would be curbside pickup. When I arrived at the store I called the pickup desk and was told my order would be out shortly. after waiting for 30 minutes I went into the store and pulled one of the items I ordered off the shelf and paid at the register. When I was walking to the register aemployee (wearing no name or apron) asked my name and about my order. I said I would pay for what I had and asked her to cancel the online order. The charge is still on my statement today. I can not reachthe credit card company by phone or text, which is how I always contacted them. When I emailed them I was told to call the customer service number. | | JPMORGAN CHASE & CO. | GA | 31601 | Servicemember | Consent provided | Web | 2020-07-02 | Closed with monetary relief | Yes | N/A | 3726873 |

```
+----------------------------+-------------------------------------------------+----------------------------------------
----+-----------------------------------------------+-------------------------------------------------+---------------
---------------------------------+-------------------------------------------------+----------------------------------
--+---------------------+------------------+-------------------+------------------------------+------------------------
----+-----------------------------------+------------------------------------------+-----------------------------+-----
---------------------------+---------------------------+
```



```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> select * from complaints limit 2;
INFO  : Compiling command(queryId=hive_20230415190221_c4a23b3e-2342-41b0-a3d5-46881159348f): select * from complaints limit 2
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:complaints.c_date_received, type:string, comment:null), FieldSchema(name:complaints.c_product, type:string, comment:null), FieldSchema(name:complaints.c_sub_produc
t, type:string, comment:null), FieldSchema(name:complaints.c_issue, type:string, comment:null), FieldSchema(name:complaints.c_sub_issue, type:string, comment:null), FieldSchema(name:complaints.c_complaint_narrative, type:string, com
ment:null), FieldSchema(name:complaints.c_company_public_response, type:string, comment:null), FieldSchema(name:complaints.c_company, type:string, comment:null), FieldSchema(name:complaints.c_state, type:string, comment:null), Field
Schema(name:complaints.c_zip, type:int, comment:null), FieldSchema(name:complaints.c_tags, type:string, comment:null), FieldSchema(name:complaints.c_consumer_consent, type:string, comment:null), FieldSchema(name:complaints.c_submitt
ed_via, type:string, comment:null), FieldSchema(name:complaints.c_date_sent_to_company, type:string, comment:null), FieldSchema(name:complaints.c_company_response_to_consumer, type:string, comment:null), FieldSchema(name:complaints.
c_timely_response, type:string, comment:null), FieldSchema(name:complaints.c_consumer_disputed, type:string, comment:null), FieldSchema(name:complaints.c_complaint_id, type:bigint, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20230415190221_c4a23b3e-2342-41b0-a3d5-46881159348f); Time taken: 0.259 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20230415190221_c4a23b3e-2342-41b0-a3d5-46881159348f): select * from complaints limit 2
INFO  : Completed executing command(queryId=hive_20230415190221_c4a23b3e-2342-41b0-a3d5-46881159348f); Time taken: 0.0 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+----------------------------+-------------------------------------------------+----------------------------------------
-+-------------------------------------------+-------------------------------------------+---------------------
----------+-------------------------+------------------------+-------------------------+
| complaints.c_date_received |                complaints.c_product                 |         complaints.c_sub_product          |              complaints.c_issue              |              complaints.c_sub_issue
 |          complaints.c_complaint_narrative           |          complaints.c_company_public_response          |          complaints.c_company         | complaints.c_state  | complaints.c_zip  | complaints.c_tags  | complaints.c_consumer
_consent  | complaints.c_submitted_via  | complaints.c_date_sent_to_company  | complaints.c_company_response_to_consumer  | complaints.c_timely_response  | complaints.c_consumer_disputed  | complaints.c_complaint_id  |
+----------------------------+-------------------------------------------------+----------------------------------------
-+-------------------------------------------+-------------------------------------------+---------------------
----------+-------------------------+------------------------+-------------------------+
| 2020-05-20                 | Credit reporting, credit repair services, or other personal consumer reports | Credit reporting                 | Incorrect information on your report         | Information belongs to s
omeone else           |                                                   | Company has responded to the consumer and the CFPB and chooses not to provide a public response | Experian Information Solutions Inc.  | ND
       | 58554       |                    | Consent not provided        | Web                          | 2020-05-20                   | Closed with explanation                    | Yes                          | N/A
                     | 3660538                    |
| 2020-07-02                 | Credit card or prepaid card                           | General-purpose credit card or charge card  | Problem with a purchase shown on your statement  | Credit card company isn't resolving a dispute abou
t a purchase on your statement | I made an online purchase at XXXX XXXX in XX/XX/2020, They said it would be curbside pickup. When I arrived at the store I called the pickup desk and was told my order would be out shortly. after wai
ting for 30 minutes I went into the store and pulled one of the items I ordered off the shelf and paid at the register. When I was walking to the register a employee ( wearing no name or apron ) asked my name and about my order. I s
aid I would pay for what I had and asked her to cancel the online order. The charge is still on my statement today. I can not reach the credit card company by phone or text, which is how I always contacted them. When I emailed them
I was told to call the customer service number. |                                          | JPMORGAN CHASE & CO.                | GA                 | 31601              | Servicemember      | Consent provided
    | Web                        | 2020-07-02                         | Closed with monetary relief                 | Yes                          | N/A                          | 3726873                    |
+----------------------------+-------------------------------------------------+----------------------------------------
-+-------------------------------------------+-------------------------------------------+---------------------
----------+-------------------------+------------------------+-------------------------+
2 rows selected (0.623 seconds)
```

3. Hive Query to get the number of complaints filed for each company for the year 2019 to 2023

Description:
The following Hive statement creates a Hive table **company_data** using data from the "complaints" table.

The SELECT statement within the CREATE TABLE statement is used to populate the company_data table with specific columns from the "complaints" table, including c_complaint_id, c_date_received, and a cleaned-up version of c_company.

The HAVING clause filters the data to only include complaints with a date_received between 2019 and 2023.

Finally, the ORDER BY clause sorts the resulting data by c_complaint_id.

NOTE: We have cleaned the c_company column to have no special characters **except** comma(,), colon(:), inverted comma (‘), double quotes (“), semi colon (;), Dollar ($), brackets (()), and space.

```
DROP TABLE IF EXISTS company_data;


CREATE TABLE company_data

row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES

("separatorChar" = ","quoteChar" = "\"")

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/DV/company_data'

AS

SELECT c_complaint_id, c_date_received, regexp_replace(c_company,'[^a-zA-Z0-9,:\'\"`;\$\(\)\\s]', ') company

FROM complaints

GROUP BY c_complaint_id, c_date_received, c_company

HAVING year(c_date_received) BETWEEN 2019 AND 2023

ORDER BY c_complaint_id;
```

4. Hive Query to get the total complaints based on Products & Subproducts for the year 2019 to 2023

Description:
The following Hive statement creates **product_data** data from the "complaints" table.

It selects the columns c_complaint_id, c_date_received, c_product, and c_sub_product from the "complaints" table.

The resulting data is grouped by c_complaint_id, c_date_received, c_product, and c_sub_product.

It filters the data based on the year of c_date_received being between 2019 and 2023.

The result is sorted by c_complaint_id.

NOTE: We have cleaned the c_product & c_sub_product columns to have no special characters **except** comma(,), colon(:), inverted comma (‘), double quotes (“), semi colon (;), Dollar ($), brackets (()), and space.

```
DROP TABLE IF EXISTS product_data;


CREATE TABLE product_data

row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES

("separatorChar" = ","quoteChar" = "\"")

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/DV/product_data'

AS

SELECT c_complaint_id, c_date_received, regexp_replace(c_product,'[^a-zA-Z0-9,:\'\"`;\$\(\)\\s]', '') product, regexp_replace(c_sub_product,'[^a-zA-Z0-9,:\'\"`;\$\(\)\\s]', '') sub_product

FROM complaints

GROUP BY c_complaint_id, c_date_received, c_product, c_sub_product

HAVING year(c_date_received) BETWEEN 2019 AND 2023

ORDER BY c_complaint_id;
```

5. Hive query to get the total complaints based on issues & sub-issues for the year 2019 to 2023

Description:

The following Hive statement creates **issue_data** data from the "complaints" table.

The SELECT statement will extract the values of the c_complaint_id, c_date_received, c_issue, and c_sub_issue columns from the "complaints" table.

The resulting data will be grouped by c_complaint_id, c_date_received, c_issue, and c_sub_issue, and only data with a c_date_received year between 2019 and 2023 will be included.

The result set will be ordered by c_complaint_id.

NOTE: We have cleaned the c_issue & c_sub_issue columns to have no special characters **except** comma(,), colon(:), inverted comma (‘), double quotes (“), semi colon (;), Dollar ($), brackets ((()), and space.

```
DROP TABLE IF EXISTS issue_data;


CREATE TABLE issue_data

row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES

("separatorChar" = ","."quoteChar" = "\"")

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/DV/issue_data'

AS

SELECT c_complaint_id, c_date_received, regexp_replace(c_issue,'[^a-zA-Z0-9,:\'\"`;\$\(\)\\s]', '') issue,
regexp_replace(c_sub_issue,'[^a-zA-Z0-9,:\'\"`;\$\(\)\\s]', '') sub_issue

FROM complaints

GROUP BY c_complaint_id, c_date_received, c_issue, c_sub_issue

HAVING year(c_date_received) BETWEEN 2019 AND 2023

ORDER BY c_complaint_id;
```

6. Hive query to get top 10 issues based on number of complaints for the year 2019 to 2023

Description:

This query creates a Hive table named **top_issues** with the data of the top 10 issues that customers complained about.

The data for the table is obtained using a Common Table Expression (CTE) named "top_10_issues". This CTE selects the "issue" column from the "issue_data" table and counts the number of complaints for each issue.

It then orders the results in descending order by the count and limits the results to the top 10 issues.

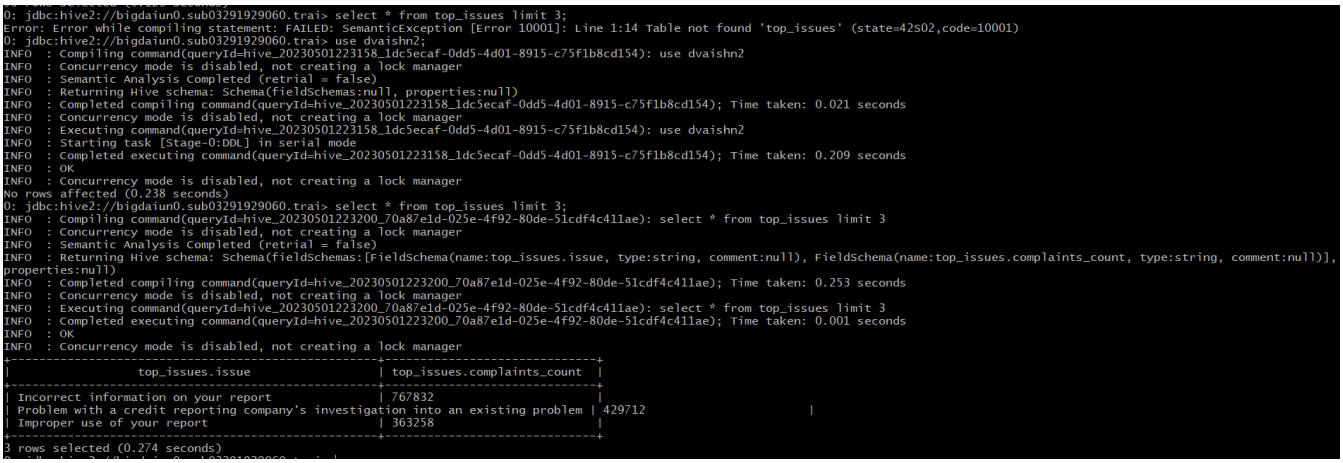
The outer SELECT statement then selects all the columns from the CTE and sorts the results in descending order by the complaint count.

```
DROP TABLE IF EXISTS top_issues;


CREATE TABLE top_issues

row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES
```

```
("separatorChar" = ","quoteChar" = "\"")
STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/DV/top_issues'
AS
with top_10_issues AS
(
SELECT issue, COUNT(*) AS complaints_count
FROM issue_data
GROUP BY issue
ORDER BY complaints_count DESC
LIMIT 10)
select * from top_10_issues
ORDER BY complaints_count DESC;
```

Selecting top 3 rows from the table to confirm:

```
SELECT * FROM top_issues LIMIT 3;
```



Get the file on Linux from HDFS:

Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS:

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/DV/top_issues/00000*_0
```

Prepare the file(s) in Linux to download to local PC:
Run the following command to check if files are present:

```
ls –al
```

We need to save the csv file with different name into the local Linux file system before downloading:

```
cat 000000_0 > top_issues.csv
```

Since all the output files will have similar name file(s) 00000*_0, lets remove this file from the Linux

```
rm 000000_0
```

```
-bash-4.2$ ls -al
total 28
drwx------   6 dvaishn2 dvaishn2   106 Apr 19 04:00 .
drwxr-xr-x. 70 root     root      4096 Apr  4 22:10 ..
-rw-------   1 dvaishn2 dvaishn2 18051 Apr 19 03:08 .bash_history
drwxr-xr-x   2 root     root        40 Apr  4 22:10 .beeline
drwxrwxr-x   3 dvaishn2 dvaishn2    18 Apr  6 00:58 .cache
drwxrwxr-x   3 dvaishn2 dvaishn2    18 Apr  6 00:58 .config
drwx------   2 dvaishn2 dvaishn2    25 Apr 12 03:09 .ssh
-rw-rw-r--   1 dvaishn2 dvaishn2   490 Apr 19 04:00 top_issues.csv
```

Copy the file to local PC:
Open another terminal with git bash to download the file to local PC.

Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password

```
scp dvaishn2@144.24.53.159:/home/dvaishn2/top_issues.csv top_issues.csv
```

```
AD+dvaishn2@STU-PF2XXWZK MINGW64 ~
$ scp dvaishn2@144.24.53.159:/home/dvaishn2/top_issues.csv top_issues.csv
dvaishn2@144.24.53.159's password:
top_issues.csv
              100%  490    17.4KB/s   00:00
```

7. Hive Query to get the number of complaints received by CFPB for the year 2019 to 2023

Description:
The following Hive statement creates a Hive table **complaint_date_analysis** using data from the "complaints" table.

The SELECT statement within the CREATE TABLE statement is used to populate the company_data table with specific columns from the "complaints" table, including c_complaint_id, c_date_received, and a cleaned-up version of c_company.

The HAVING clause filters the data to only include complaints with a date_received between 2019 and 2023.

Finally, the ORDER BY clause sorts the resulting data by c_date_received.

NOTE: We have cleaned the c_company column to have no special characters except comma(,), colon(:), inverted comma (‘), double quotes (“), semi colon (;), Dollar ($), brackets (()), and space.

Creating table using database “mdhoke”:

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use mdhoke;

```
DROP TABLE IF EXISTS complaints_date_analysis;


CREATE TABLE complaints_date_analysis

row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES

("separatorChar" = ","quoteChar" = "\"")

STORED AS TEXTFILE LOCATION
'/user/dvaishn2/5200_Project_Tables/MD/complaints_date_analysis'

AS

SELECT c_date_received ,COUNT(c_complaint_id)

FROM dvaishn2.complaints

GROUP BY  c_date_received

Having (YEAR(c_date_received) BETWEEN 2021 AND 2023) AND  c_date_received IS NOT NULL

ORDER BY c_date_received ;


select * from complaints_date_analysis limit 10;
```

```
INFO  : Concurrency mode is disabled, not creating a lock manager
+-------------------------------------------+-------------------------------+
| complaints_date_analysis.c_date_received  | complaints_date_analysis._c1  |
+-------------------------------------------+-------------------------------+
| 2021-01-01                                | 624                           |
| 2021-01-02                                | 904                           |
| 2021-01-03                                | 675                           |
| 2021-01-04                                | 1991                          |
| 2021-01-05                                | 2310                          |
| 2021-01-06                                | 2002                          |
| 2021-01-07                                | 1570                          |
| 2021-01-08                                | 1659                          |
| 2021-01-09                                | 720                           |
| 2021-01-10                                | 510                           |
+-------------------------------------------+-------------------------------+
```

Get the file on Linux from HDFS:

Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS:

```
hdfs dfs -get /user/mdhoke/5200_Project_Tables/MD/complaints_date_analysis/00000*_0
```

Prepare the file(s) in Linux to download to local PC:

Run the following command to check if files are present:

```
ls –al
```

We need to save the csv file with different name into the local Linux file system before downloading

```
cat 000000_0 > complaints_date_analysis.csv
```

Since all the output files will have similar name file(s) 00000*_0, lets remove this file from the Linux

```
rm 000000_0
```

```
drwx------   6 dvaishn2 dvaishn2       129 Apr 17 20:50 .
drwxr-xr-x. 70 root     root         4096 Apr  4 22:10 ..
-rw-------   1 dvaishn2 dvaishn2     14613 Apr 17 19:14 .bash_history
drwxr-xr-x   2 root     root           40 Apr  4 22:10 .beeline
drwxrwxr-x   3 dvaishn2 dvaishn2       18 Apr  6 00:58 .cache
-rw-rw-r--   1 dvaishn2 dvaishn2 121781358 Apr 17 20:50 company_data.csv
drwxrwxr-x   3 dvaishn2 dvaishn2       18 Apr  6 00:58 .config
-rw-rw-r--   1 dvaishn2 dvaishn2  10190981 Apr 17 01:00 loan_data.csv
drwx------   2 dvaishn2 dvaishn2       25 Apr 12 03:09 .ssh
```

Copy the file to local PC:

Open another terminal with git bash to download the file to local PC.

Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password

```
scp mdhoke@144.24.53.159:/home/mdhoke/complaints_date_analysis.csv complaints_date_analysis.csv
```

```
MINGW64:/c/Users/V P DHOKE

V P DHOKE@DESKTOP-JOBKIO5 MINGW64 /
$ cd

V P DHOKE@DESKTOP-JOBKIO5 MINGW64 ~
$ scp mdhoke@144.24.53.159:/home/mdhoke/complaints_date_analysis.csv complaints_
date_analysis.csv
mdhoke@144.24.53.159's password:
complaints_date_analysis.csv                  100%   78KB 241.8KB/s   00:00

V P DHOKE@DESKTOP-JOBKIO5 MINGW64 ~
$
```

8. Display companies with their Product and respective issues for each state

Description:

The following Hive statement creates **state_table** data from the "complaints, company_data, product_data, issue_data" table.

It selects the columns c_complaint_id , c_date_received, company FROM “company_data”, product FROM “product_data”, issue FROM “issue_data”, and c_state, c_zip FROM the "complaints" table.

The resulting data is grouped by c_complaint_id, c_date_received, product, company, issue, c_state, c_zip

It filters the data based on the year of c_date_received being between 2021 and 2023 and c_state is not equal to null.

The result is ordered by c_state.

NOTE: We have joined the four tables, (where we already created the table with cleaned data) with c_complaint_id field (The tables: “company_data”, “product_data”, “issue_data”, "complaints").

Creating table using database “mneethi”:

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use mneethi;

```
DROP TABLE IF EXISTS state_table;

CREATE TABLE state_table row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES ("separatorChar" = ","quoteChar" = "\"")

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/Mani/state_table'

AS

SELECT cd.c_complaint_id AS complaint_id, cd.c_date_received AS date_received, cd.company AS company,

pd.product AS Product, id.issue AS issue, master.c_state AS state, master.c_zip AS zip

FROM dvaishn2.complaints master

JOIN dvaishn2.company_data cd ON cd.c_complaint_id =  master.c_complaint_id

JOIN dvaishn2.product_data pd ON pd.c_complaint_id = master.c_complaint_id

JOIN dvaishn2.issue_data id ON id.c_complaint_id= master.c_complaint_id

GROUP BY cd.c_complaint_id, cd.c_date_received, cd.company, pd.product, id.issue, master.c_state, master.c_zip

HAVING year(cd.c_date_received) BETWEEN 2021 AND 2023 AND master.c_state != ''

ORDER BY state;
```

```
+--------------------------+--------------------------+--------------------------------------+------------------------
-------------------------+--------------------------------------+-------------------+-----------------+
```

| state_table.complaint_id | state_table.date_received | state_table.company | state_table.product | state_table.issue | state_table.state | state_table.zip |

+---------------------------+----------------------------+---------------------------------------+--+-------------------------------------+--------------------+------------------+

| 6680780 | 2023-03-11 | Paypal Holdings, Inc | Money transfer, virtual currency, or money service | Fraud or scam | AA | 10001 |

| 4060539 | 2021-01-11 | TRANSUNION INTERMEDIATE HOLDINGS, INC | Credit reporting, credit repair services, or other personal consumer reports | Incorrect information on your report | AA | 34004 |

+---------------------------+----------------------------+---------------------------------------+--+-------------------------------------+--------------------+------------------+

Prepare the file(s) in Linux to download to local PC:

Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS:

```
hdfs dfs -ls /user/dvaishn2/5200_Project_Tables/Mani/state_table
```

```
-bash-4.2$ hdfs dfs -ls /user/dvaishn2/5200_Project_Tables/Mani/state_table
Found 1 items
-rw-r--r--   3 mneethi hdfs  276781169 2023-04-20 00:30 /user/dvaishn2/5200_Project_Tables/Mani/state_table/000000_0
-bash-4.2$
```

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/Mani/state_table/00000*_0
```

Download the file(s) from Linux to local PC:

Run the following command to check if files are present:

```
ls –al

cat 000000_0 > state_table.csv
```

```
-bash-4.2$ ls -al
total 270312
drwx------   7 mneethi mneethi       119 Apr 20 00:37 .
drwxr-xr-x. 70 root    root         4096 Apr  4 22:10 ..
-rw-------   1 mneethi mneethi      9087 Apr 19 23:45 .bash_history
drwxr-xr-x   2 root    root           40 Apr  4 22:10 .beeline
drwxrwxr-x   3 mneethi mneethi        18 Apr  6 00:45 .cache
drwxrwxr-x   3 mneethi mneethi        18 Apr  6 00:45 .config
drwxr-xr-x   4 mneethi mneethi        56 Apr 18 00:59 Mani
drwx------   2 mneethi mneethi        25 Apr 20 00:34 .ssh
-rw-rw-r--   1 mneethi mneethi 276781169 Apr 20 00:34 state_table.csv
```

Since all the output files will have similar name file(s) 00000*_0, lets remove this file from the Linux

```
rm 000000_0
```

Copy the file to local PC:

Open another terminal with git bash to download the file to local PC.

Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password

```
scp mneethi@144.24.53.159:/home/mneethi/state_table.csv state_table.csv
```

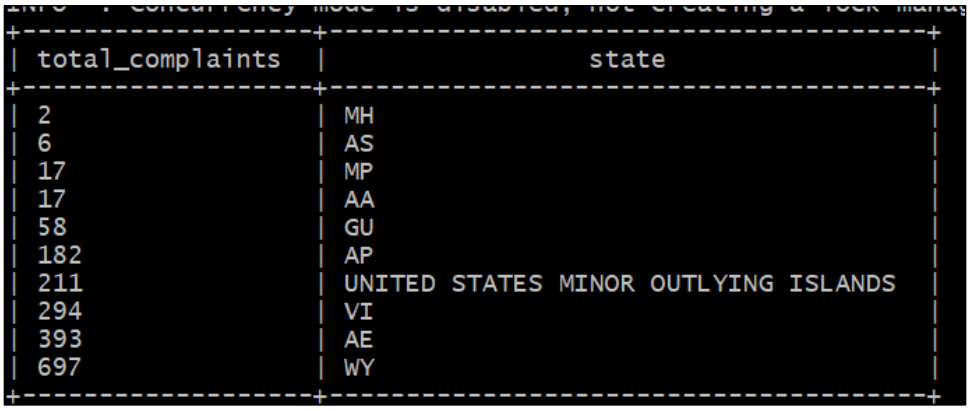
Once, the state_table is created, extracting the details to show the highest and lowest state which received the complaints.

We have grouped the result by state and order the result by total count of complaints.

```
SELECT count(*) AS total_complaints, state FROM state_table
GROUP BY state ORDER BY total_complaints desc limit 10;
```

total_complaints	state
187825	FL
173795	TX
166793	CA
123727	GA
98073	NY
92829	PA
66856	IL
56146	NC
52291	NJ
44324	MD

```
SELECT count(*) AS total_complaints, state FROM state_table
GROUP BY state ORDER BY total_complaints asc limit 10;
```



total_complaints	state
2	MH
6	AS
17	MP
17	AA
58	GU
182	AP
211	UNITED STATES MINOR OUTLYING ISLANDS
294	VI
393	AE
697	WY

9. Find out top 3 products the consumers identified in the complaint

Description:

This query creates a Hive table named **top_products** with the data of the top products that customers complained about.

The data for the table is obtained using the table created by another user (dvaishn2) dvaishn2.product_data and named new table as " top_products ". This selects the "product" column from the "product_data" table and counts the number of complaints for each product.

It then orders the results in descending order by the count and limits the results to the top 3 products.

The outer SELECT statement then selects all the columns from the top_products and sorts the results in descending order by the complaint count.

```
DROP TABLE IF EXISTS top_products;

CREATE TABLE top_products

row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES

("separatorChar" = ","quoteChar" = "\"")

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/Mani/top_products'

AS

with top_3_products AS

(SELECT product, COUNT(*) AS complaints_count

FROM dvaishn2.product_data

GROUP BY product

ORDER BY complaints_count DESC

LIMIT 3)

SELECT * FROM top_3_products

ORDER BY complaints_count DESC;
```

Once the table is created, using SELECT query checking the table.

```
SELECT * FROM top_products;
```

```
+-------------------------------------------------+------------------------------+
|               top_products.product              | top_products.complaints_count |
+-------------------------------------------------+------------------------------+
| Credit reporting, credit repair services, or other personal consumer reports | 1581329
| Debt collection                                 | 246934                       |
| Credit card or prepaid card                     | 143837                       |
+-------------------------------------------------+------------------------------+
```

Prepare the file(s) in Linux to download to local PC:

Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS:

```
hdfs dfs -ls /user/dvaishn2/5200_Project_Tables/Mani/top_products
```

```
-bash-4.2$ hdfs dfs -ls /user/dvaishn2/5200_Project_Tables/Mani/top_products
Found 1 items
-rw-r--r--   3 mneethi hdfs        155 2023-04-20 00:46 /user/dvaishn2/5200_Project_Tables/Mani/top_products/000000_0
-bash-4.2$
```

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/Mani/top_products/00000*_0
```

Download the file(s) from Linux to local PC:

Run the following command to check if files are present:

```
ls –al
```

We need to save the csv file with different name into the local Linux file system before downloading

```
cat 000000_0 > top_products.csv
```

```
-bash-4.2$ cat 000000_0 > top_products.csv
-bash-4.2$ ls -al
total 5148
drwx------   9 mneethi mneethi    4096 May  2 18:09 .
drwxr-xr-x. 70 root    root       4096 Apr  4 22:10 ..
-rw-r--r--   1 mneethi mneethi     155 May  2 18:09 000000_0
-rw-r--r--   1 mneethi mneethi 1960945 Apr 27 02:39 000001_0
-rw-------   1 mneethi mneethi   14035 Apr 27 05:56 .bash_history
drwxr-xr-x   2 root    root         40 Apr  4 22:10 .beeline
drwxrwxr-x   3 mneethi mneethi      18 Apr  6 00:45 .cache
drwxrwxr-x   3 mneethi mneethi      18 Apr  6 00:45 .config
drwxr-xr-x   2 mneethi mneethi     101 Aug  9  2016 driver_data
-rw-rw-r--   1 mneethi mneethi  273059 Apr 27 01:32 driver_data.zip
-rw-rw-r--   1 mneethi mneethi 3001419 Apr 27 02:40 driver_event.txt
drwxrwxr-x   3 mneethi mneethi      25 Sep  8  2016 __MACOSX
drwxr-xr-x   4 mneethi mneethi      56 Apr 18 00:59 Mani
drwx------   2 mneethi mneethi      25 Apr 20 00:34 .ssh
-rw-rw-r--   1 mneethi mneethi     155 May  2 18:09 top_products.csv
```

Since all the output files will have similar name file(s) 00000*_0, lets remove this file from the Linux

```
rm 000000_0
```

Copy the file to local PC:

Open another terminal with git bash to download the file to local PC.

Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password

```
scp mneethi@144.24.53.159:/home/mneethi/top_products.csv top_products.csv
```

10. Show the statistics for California about most complaints received based on Products

Description:

This query creates a Hive table named **ca_product** with the data of the top products in California that customers complained about.

The data for the table is obtained from state_table and named new table as " ca_product ". This selects the "state" column from the "state_table" table and counts the number of complaints for each product.

The result is grouped by state and product and then filtered the state which has “CA” and order the result in descending order by the count

```
DROP TABLE IF EXISTS ca_product;

CREATE TABLE ca_product

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/Mani/ca_product'

AS

SELECT state, count(*) AS total, product

FROM state_table

GROUP BY product, state

HAVING state = 'CA'

ORDER BY total DESC;
```

Once the table is created using the SELECT query checking the table.

```
SELECT * FROM ca_product;
```

```
+--------------------+--------------------+-----------------------------------------------+
| ca_complaints.state | ca_complaints.total |             ca_complaints.product             |
+--------------------+--------------------+-----------------------------------------------+
| CA                 | 111046             | Credit reporting, credit repair services, or other personal consumer reports |
| CA                 | 14635              | Debt collection                               |
| CA                 | 12116              | Credit card or prepaid card                   |
| CA                 | 12042              | Checking or savings account                   |
| CA                 | 7931               | Mortgage                                      |
| CA                 | 4242               | Money transfer, virtual currency, or money service |
| CA                 | 2118               | Vehicle loan or lease                         |
| CA                 | 1348               | Student loan                                  |
| CA                 | 1315               | Payday loan, title loan, or personal loan     |
+--------------------+--------------------+-----------------------------------------------+
9 rows selected (0.272 seconds)
```

Prepare the file(s) in Linux to download to local PC:

Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS:

```
hdfs dfs -ls /user/dvaishn2/5200_Project_Tables/Mani/ ca_product
```

```
-bash-4.2$ hdfs dfs -ls /user/dvaishn2/5200_Project_Tables/Mani/ca_product
Found 1 items
-rw-r--r--   3 mneethi hdfs        363 2023-04-21 22:52 /user/dvaishn2/5200_Project_Tables/Mani/ca_product/000000_0
-bash-4.2$
```

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/Mani/ca_product/00000*_0
```

Download the file(s) from Linux to local PC:

Run the following command to check if files are present:

```
ls –al
```

We need to save the csv file with different name into the local Linux file system before downloading

```
cat 000000_0 > ca_product.csv
```

```
-bash-4.2$ hdfs dfs -get /user/dvaishn2/5200_Project_Tables/Mani/ca_product/00000*_0
-bash-4.2$ cat 000000_0 > ca_product.csv
-bash-4.2$ ls -al
total 5152
drwx------   9 mneethi mneethi    4096 May  2 18:21 .
drwxr-xr-x. 70 root    root       4096 Apr  4 22:10 ..
-rw-r--r--   1 mneethi mneethi     363 May  2 18:20 000000_0
-rw-r--r--   1 mneethi mneethi 1960945 Apr 27 02:39 000001_0
-rw-------   1 mneethi mneethi   14035 Apr 27 05:56 .bash_history
drwxr-xr-x   2 root    root         40 Apr  4 22:10 .beeline
drwxrwxr-x   3 mneethi mneethi      18 Apr  6 00:45 .cache
-rw-rw-r--   1 mneethi mneethi     363 May  2 18:21 ca_product.csv
drwxrwxr-x   3 mneethi mneethi      18 Apr  6 00:45 .config
drwxr-xr-x   2 mneethi mneethi     101 Aug  9  2016 driver_data
-rw-rw-r--   1 mneethi mneethi  273059 Apr 27 01:32 driver_data.zip
-rw-rw-r--   1 mneethi mneethi 3001419 Apr 27 02:40 driver_event.txt
drwxrwxr-x   3 mneethi mneethi      25 Sep  8  2016 __MACOSX
drwxr-xr-x   4 mneethi mneethi      56 Apr 18 00:59 Mani
drwx------   2 mneethi mneethi      25 Apr 20 00:34 .ssh
-rw-rw-r--   1 mneethi mneethi     155 May  2 18:09 top_products.csv
```

Since all the output files will have similar name file(s) 00000*_0, lets remove this file from the Linux

```
rm 000000_0
```

Copy the file to local PC:

Open another terminal with git bash to download the file to local PC.

Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password

```
scp mneethi@144.24.53.159:/home/mneethi/ca_product.csv ca_product.csv
```

```
manim@manimozhi MINGW64 ~
$ scp mneethi@144.24.53.159:/home/mneethi/ca_product.csv ca_product.csv
mneethi@144.24.53.159's password:
ca_product.csv                                              100%  363     3.6KB/s   00:00
```

11. Find out Company with highest complaints for California

Description:

This query creates a Hive table named **ca_company** with the data of the top company in California that customers complained about.

The data for the table is obtained from state_table and named new table as " ca_company". This selects the "company " column from the "state_table" table and counts the number of complaints for each company.

The result is grouped by state and company and then filtered the state which has “CA”, company is not equal to null and order the result in descending order by the count amd limit the result by 5.

```
DROP TABLE IF EXISTS ca_company

CREATE TABLE ca_company

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/Mani/ca_company'

AS

SELECT company AS company, count(complaint_id) AS total

FROM state_table

WHERE state = 'CA'

GROUP BY company, state

HAVING company!=''

ORDER BY total DESC LIMIT 5;
```

Once the table is created, using SELECT query checking the table.

```
SELECT * FROM ca_company;
```

```
+----------------------------------------+-------------------+
|           ca_company.company           | ca_company.total  |
+----------------------------------------+-------------------+
| EQUIFAX, INC                           | 36746             |
| TRANSUNION INTERMEDIATE HOLDINGS, INC  | 30858             |
| Experian Information Solutions Inc     | 27775             |
| BANK OF AMERICA, NATIONAL ASSOCIATION  | 6414              |
| WELLS FARGO  COMPANY                   | 5260              |
+----------------------------------------+-------------------+
```

Prepare the file(s) in Linux to download to local PC:

Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS:

```
hdfs dfs -ls /user/dvaishn2/5200_Project_Tables/Mani/ca_company
```

```
-bash-4.2$ hdfs dfs -ls /user/dvaishn2/5200_Project_Tables/Mani/ca_company
Found 1 items
-rw-r--r--   3 mneethi hdfs        173 2023-04-21 22:46 /user/dvaishn2/5200_Project_Tables/Mani/ca_company/000000_0
-bash-4.2$
```

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/Mani/ca_company/00000*_0
```

Download the file(s) from Linux to local PC:

Run the following command to check if files are present:

```
ls –al
```

We need to save the csv file with different name into the local Linux file system before downloading

```
cat 000000_0 > ca_company.csv
```

```
-bash-4.2$ hdfs dfs -get /user/dvaishn2/5200_Project_Tables/Mani/ca_company/00000*_0
-bash-4.2$ cat 000000_0 > ca_company.csv
-bash-4.2$ ls -al
total 5156
drwx------   9 mneethi mneethi    4096 May  2 18:30 .
drwxr-xr-x. 70 root    root       4096 Apr  4 22:10 ..
-rw-r--r--   1 mneethi mneethi     173 May  2 18:30 000000_0
-rw-r--r--   1 mneethi mneethi 1960945 Apr 27 02:39 000001_0
-rw-------   1 mneethi mneethi   14035 Apr 27 05:56 .bash_history
drwxr-xr-x   2 root    root         40 Apr  4 22:10 .beeline
drwxrwxr-x   3 mneethi mneethi      18 Apr  6 00:45 .cache
-rw-rw-r--   1 mneethi mneethi     173 May  2 18:30 ca_company.csv
-rw-rw-r--   1 mneethi mneethi     363 May  2 18:21 ca_product.csv
drwxrwxr-x   3 mneethi mneethi      18 Apr  6 00:45 .config
drwxr-xr-x   2 mneethi mneethi     101 Aug  9  2016 driver_data
-rw-rw-r--   1 mneethi mneethi  273059 Apr 27 01:32 driver_data.zip
-rw-rw-r--   1 mneethi mneethi 3001419 Apr 27 02:40 driver_event.txt
drwxrwxr-x   3 mneethi mneethi      25 Sep  8  2016 __MACOSX
drwxr-xr-x   4 mneethi mneethi      56 Apr 18 00:59 Mani
drwx------   2 mneethi mneethi      25 Apr 20 00:34 .ssh
-rw-rw-r--   1 mneethi mneethi     155 May  2 18:09 top_products.csv
```

Copy the file to local PC:

Open another terminal with git bash to download the file to local PC.

Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password

```
scp mneethi@144.24.53.159:/home/mneethi/ca_company.csv ca_company.csv
```

12. Hive query to show the analysis on the company response to the complaints for last 3 years for the state CA

Description:

This query creates a table **CA_company_response** that stores information about the responses of the company "EQUIFAX, INC." to consumer complaints received in California between 2021 and 2023 related to the product "Credit reporting, credit repair services, or other personal consumer reports" and the issue "Incorrect information on your report".

The c_company_response_to_consumer field is checked to make sure it is not empty before being stored on the new table.

Creating table using database "dvaishn2":

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use dvaishn2;

```
DROP TABLE IF EXISTS CA_company_response;

CREATE TABLE CA_company_response

row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES

("separatorChar" = ","quoteChar" = "\"")

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/DV/CA_company_response'

AS

SELECT c_company, c_company_response_to_consumer as company_response

FROM mneethi.top_products tp, top_issues ti, complaints

WHERE tp.product = 'Credit reporting, credit repair services, or other personal consumer reports'

AND ti.issue = 'Incorrect information on your report'

AND c_company = 'EQUIFAX, INC.'

AND c_state = 'CA'

AND c_company_response_to_consumer != "
```

```
AND YEAR(c_date_received) BETWEEN 2021 AND 2023;
```

Selecting the data from the table:

```
select distinct c_company, company_response from CA_company_response;
```

Note: Selecting "distinct" companies to avoid displaying all 36,746 rows repeating the company & its response.

```
+----------------+-----------------------------------+
|   c_company    |         company_response          |
+----------------+-----------------------------------+
| EQUIFAX, INC.  | In progress                       |
| EQUIFAX, INC.  | Closed with explanation           |
| EQUIFAX, INC.  | Closed with non-monetary relief   |
+----------------+-----------------------------------+
```

Get the file on Linux from HDFS:

Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS:

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/DV/CA_company_response/00000*_0
```

Prepare the file(s) in Linux to download to local PC:

Run the following command to check if files are present:

```
ls –al
```

We need to save the csv file with different name into the local Linux file system before downloading

```
cat 000000_0 > CA_company_response.csv
```

Since all the output files will have similar name file(s) 00000*_0, lets remove this file from the Linux

```
rm 000000_0
```

```
-bash-4.2$ ls -al
total 1572
drwx------   6 dvaishn2 dvaishn2     135 Apr 21 02:12 .
drwxr-xr-x. 70 root     root        4096 Apr  4 22:10 ..
-rw-------   1 dvaishn2 dvaishn2   20818 Apr 21 00:00 .bash_history
drwxr-xr-x   2 root     root          40 Apr  4 22:10 .beeline
drwxrwxr-x   3 dvaishn2 dvaishn2      18 Apr  6 00:58 .cache
-rw-rw-r--   1 dvaishn2 dvaishn2 1508116 Apr 21 02:12 CA_company_response.csv
drwxrwxr-x   3 dvaishn2 dvaishn2      18 Apr  6 00:58 .config
-rw-rw-r--   1 dvaishn2 dvaishn2   66827 Apr 20 00:35 products.tsv
drwx------   2 dvaishn2 dvaishn2      25 Apr 12 03:09 .ssh
```

Copy the file to local PC:

Open another terminal with git bash to download the file to local PC.

Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password

```
scp dvaishn2@144.24.53.159:/home/dvaishn2/CA_company_response.csv CA_company_response.csv
```

```
AD+dvaishn2@STU-PF2XXWZK MINGW64 ~
$ scp dvaishn2@144.24.53.159:/home/dvaishn2/CA_company_response.csv CA_company_response.csv
dvaishn2@144.24.53.159's password:
CA_company_response.csv                                     100% 1473KB   8.9MB/s   00:00
```

13. Find out the top 5 companies which have received the highest number of complaints

Description:

Create a table called **top_companies_complaints** using the CSV file stored in the HDFS location "/user/dvaishn2/5200_Project_Tables/AK/top_companies_complaints". The table is defined using the OpenCSVSerde, which is a SerDe (serializer/deserializer) that can read and write CSV files. Use a common table expression (CTE) to define a subquery that counts the number of complaints for each company in the "company_data" table and groups them by company. The results are sorted in descending order by the number of complaints and limited to the top 5 companies.

Creating table using database "akhaire3":

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use akhaire3;

```
CREATE TABLE top_companies_complaints
row format SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES
("separatorChar" = ","quoteChar" = "\"")
STORED AS TEXTFILE LOCATION
'/user/dvaishn2/5200_Project_Tables/AK/top_companies_complaints'
AS
with top_companies_complaints AS
(
SELECT company, count(*) as complaints FROM dvaishn2. company_data master GROUP BY
company ORDER BY complaints DESC LIMIT 5)
select * FROM top_companies_complaints ORDER BY complaints DESC;
```

Select all columns from the "top_companies_complaints" and order the results by the number of complaints in descending order.

```
select * from top_companies_complaints;
```

top_companies_complaints.company	top_companies_complaints.complaints
EQUIFAX, INC	531523
TRANSUNION INTERMEDIATE HOLDINGS, INC	457249
Experian Information Solutions Inc	415544
CAPITAL ONE FINANCIAL CORPORATION	47447
BANK OF AMERICA, NATIONAL ASSOCIATION	44899

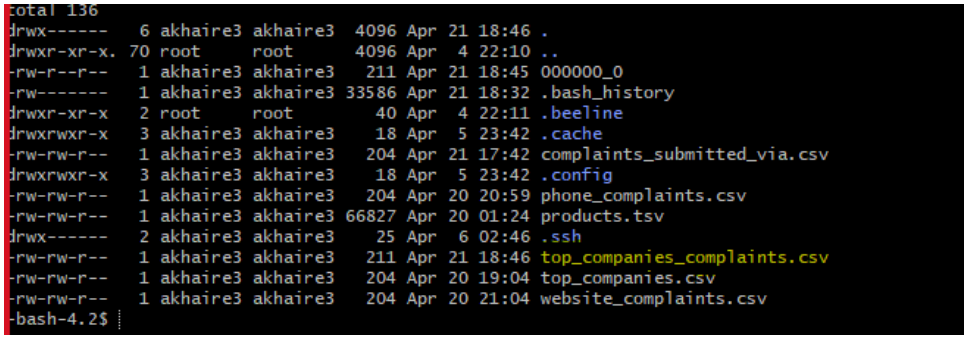
Retrieve the data from the "top_companies_complaints" table stored in HDFS by using the "hdfs dfs -get" command in Linux. This command copies the file from HDFS to the local file system.

Get the file on Linux from HDFS:

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/AK/top_companies_complaints/00000*_0
```

Get the file from linux to local PC:

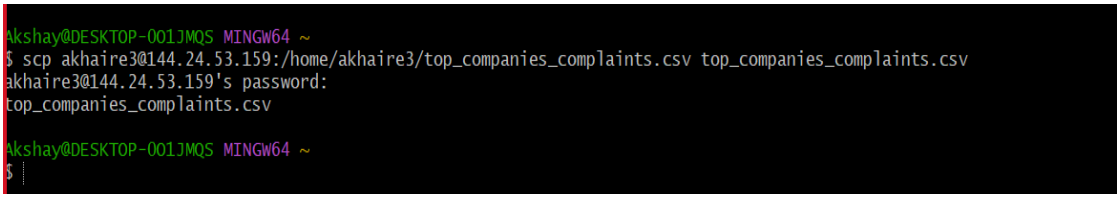
```
ls –al
cat 000000_0 > top_companies_complaints.csv
ls –al
rm 000000_0
```



Copy the file to local PC:

Copy the file from the Linux machine to a local PC using the "scp" command. This command securely copies files between hosts on a network using SSH.

```
scp akhaire3@144.24.53.159:/home/akhaire3/top_companies_complaints.csv
top_companies_complaints.csv
```



14. How did the consumer submit the complaint to the CFPB?

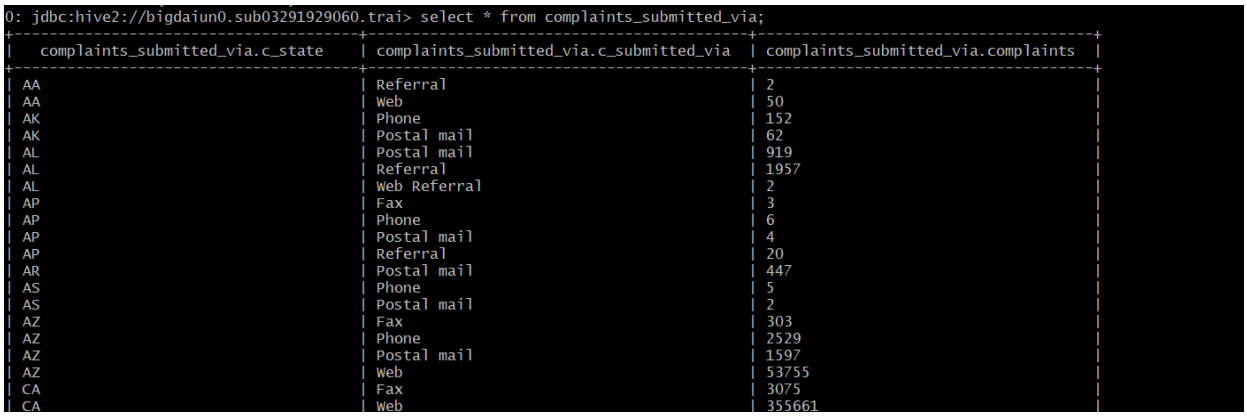
Description

Create a table called **complaints_submitted_via** using the CSV file stored in the HDFS location "/user/dvaishn2/5200_Project_Tables/AK/complaints_submitted_via". The table is defined using the OpenCSVSerde, which is a SerDe (serializer/deserializer) that can read and write CSV files. Use a SELECT statement to define a subquery that counts the number of complaints for each channel and state in the "complaints" table and groups them by state and submitted via channel. The results are filtered to exclude any rows where the state or submitted via column is empty.

```
CREATE TABLE complaints_submitted_via
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES ("separatorChar" = ",", "quoteChar" = "\"")
STORED AS TEXTFILE
LOCATION '/user/dvaishn2/5200_Project_Tables/AK/complaints_submitted_via'
AS
SELECT c_state, c_submitted_via, COUNT(c_complaint_id) AS complaints
FROM dvaishn2.complaints
GROUP BY c_state, c_submitted_via
HAVING c_state!='' AND c_submitted_via!='';
```

Insert the results of the SELECT statement into the "complaints_submitted_via" table.

```
select * from complaints_submitted_via;
```



```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> select * from complaints_submitted_via;
+-------------------------------------+-----------------------------------------+-------------------------------------+
|   complaints_submitted_via.c_state  | complaints_submitted_via.c_submitted_via | complaints_submitted_via.complaints |
+-------------------------------------+-----------------------------------------+-------------------------------------+
| AA                                  | Referral                                | 2                                   |
| AA                                  | Web                                     | 50                                  |
| AK                                  | Phone                                   | 152                                 |
| AK                                  | Postal mail                             | 62                                  |
| AL                                  | Postal mail                             | 919                                 |
| AL                                  | Referral                                | 1957                                |
| AL                                  | Web Referral                            | 2                                   |
| AP                                  | Fax                                     | 3                                   |
| AP                                  | Phone                                   | 6                                   |
| AP                                  | Postal mail                             | 4                                   |
| AP                                  | Referral                                | 20                                  |
| AR                                  | Postal mail                             | 447                                 |
| AS                                  | Phone                                   | 5                                   |
| AS                                  | Postal mail                             | 2                                   |
| AZ                                  | Fax                                     | 303                                 |
| AZ                                  | Phone                                   | 2529                                |
| AZ                                  | Postal mail                             | 1597                                |
| AZ                                  | Web                                     | 53755                               |
| CA                                  | Fax                                     | 3075                                |
| CA                                  | Web                                     | 355661                              |
```

Retrieve the data from the "complaints_submitted_via" table stored in HDFS by using the "hdfs dfs -get" command in Linux. This command copies the file from HDFS to the local file system.

Get the file on Linux from HDFS:

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/AK/complaints_submitted_via/00000*_0
```

Get the file from linux to local PC:

```
ls –al
cat 000000_0 > complaints_submitted_via.csv
ls –al
rm 000000_0
```

```
total 136
drwx------   6 akhaire3 akhaire3  4096 Apr 21 18:46 .
drwxr-xr-x. 70 root     root      4096 Apr  4 22:10 ..
-rw-r--r--   1 akhaire3 akhaire3   211 Apr 21 18:45 000000_0
-rw-------   1 akhaire3 akhaire3 33586 Apr 21 18:32 .bash_history
drwxr-xr-x   2 root     root        40 Apr  4 22:11 .beeline
drwxrwxr-x   3 akhaire3 akhaire3    18 Apr  5 23:42 .cache
-rw-rw-r--   1 akhaire3 akhaire3   204 Apr 21 17:42 complaints_submitted_via.csv
drwxrwxr-x   3 akhaire3 akhaire3    18 Apr  5 23:42 .config
-rw-rw-r--   1 akhaire3 akhaire3   204 Apr 20 20:59 phone_complaints.csv
-rw-rw-r--   1 akhaire3 akhaire3 66827 Apr 20 01:24 products.tsv
drwx------   2 akhaire3 akhaire3    25 Apr  6 02:46 .ssh
```

Copy the file to local PC:

Copy the file from the Linux machine to a local PC using the "scp" command. This command securely copies files between hosts on a network using SSH.

```
scp akhaire3@144.24.53.159:/home/akhaire3/complaints_submitted_via.csv
complaints_submitted_via.csv
```

```
akshay@DESKTOP-OO1JMQS MINGW64 ~
$ scp akhaire3@144.24.53.159:/home/akhaire3/complaints_submitted_via.csv complaints_submitted_via.csv
akhaire3@144.24.53.159's password:
complaints_submitted_via.csv                                  100% 7996    71.0KB/s   00:00

akshay@DESKTOP-OO1JMQS MINGW64 ~
```

15. Hive Queries to perform Sentiment Analysis on Complaints Narrative

Description:

We need to create a table dictionary, which has polarity to show each word's meaning implied as positive or negative. Execute the following command to create the table:

Creating table using database "dvaishn2":

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;

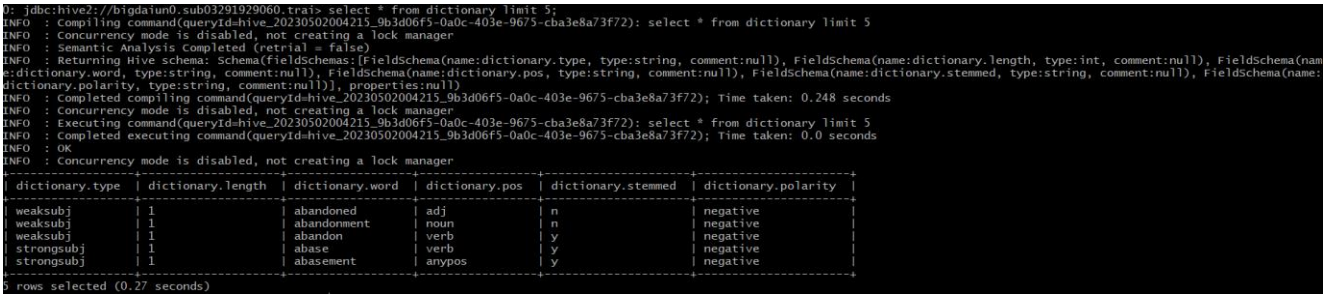
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use dvaishn2;

```
CREATE EXTERNAL TABLE dictionary (

type string,

length int,

word string,

pos string,

stemmed string,
```

```
polarity string )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/user/dvaishn2/5200_Project_Tables/DV/dictionary';
```

You must query some data from to see if it has the correct data and values:

```
SELECT * FROM dictionary LIMIT 5;
```



```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> select * from dictionary limit 5;
INFO  : Compiling command(queryId=hive_20230502004215_9b3d06f5-0a0c-403e-9675-cba3e8a73f72): select * from dictionary limit 5
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:dictionary.type, type:string, comment:null), FieldSchema(name:dictionary.length, type:int, comment:null), FieldSchema(nam
e:dictionary.word, type:string, comment:null), FieldSchema(name:dictionary.pos, type:string, comment:null), FieldSchema(name:dictionary.stemmed, type:string, comment:null), FieldSchema(name:
dictionary.polarity, type:string, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20230502004215_9b3d06f5-0a0c-403e-9675-cba3e8a73f72); Time taken: 0.248 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20230502004215_9b3d06f5-0a0c-403e-9675-cba3e8a73f72): select * from dictionary limit 5
INFO  : Completed executing command(queryId=hive_20230502004215_9b3d06f5-0a0c-403e-9675-cba3e8a73f72); Time taken: 0.0 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+-------------------+--------------------+-------------------+-----------------+--------------------+---------------------+
| dictionary.type   | dictionary.length  | dictionary.word   | dictionary.pos  | dictionary.stemmed | dictionary.polarity |
+-------------------+--------------------+-------------------+-----------------+--------------------+---------------------+
| weaksubj          | 1                  | abandoned         | adj             | n                  | negative            |
| weaksubj          | 1                  | abandonment       | noun            | n                  | negative            |
| weaksubj          | 1                  | abandon           | verb            | y                  | negative            |
| strongsubj        | 1                  | abase             | verb            | y                  | negative            |
| strongsubj        | 1                  | abasement         | anypos          | y                  | negative            |
+-------------------+--------------------+-------------------+-----------------+--------------------+---------------------+
5 rows selected (0.27 seconds)
```

Description:

This query creates a view **complaint_clean**.

The view selects columns c_complaint_id, c_date_received, and c_state from the complaints table

It applies a regular expression to the c_complaint_narrative column to remove any non-alphanumeric characters (excluding whitespace).

The WHERE clause filters the rows to only include complaints received in the years 2022 and 2023.

```
CREATE VIEW IF NOT EXISTS complaints_clean AS
SELECT
c_complaint_id,
c_date_received,
regexp_replace(c_complaint_narrative, '[^a-zA-Z0-9\\s]', '') narrative,
c_state
FROM complaints
WHERE YEAR(c_date_received) BETWEEN 2022 AND 2023;
```

Following views will split the consumer narrative value to each word and explode in multiple rows:

```
CREATE VIEW IF NOT EXISTS complaints_v1 as
```

```
SELECT c_complaint_id, c_date_received, words
FROM complaints
LATERAL VIEW EXPLODE(sentences(lower(c_complaint_narrative))) dummy as words
WHERE YEAR(c_date_received) BETWEEN 2022 AND 2023;


CREATE VIEW IF NOT EXISTS complaints_v2 as
SELECT c_complaint_id, c_date_received, word
FROM complaints_v1
LATERAL VIEW EXPLODE ( words ) dummy as word
WHERE YEAR(c_date_received) BETWEEN 2022 AND 2023;
```

Following view is created to map the polarity value to each word of complaint narrative:

```
create view IF NOT EXISTS complaints_v3 as select
c_complaint_id, c_date_received,
complaints_v2.word,
case d.polarity
when 'negative' then -1
when 'positive' then 1
else 0 end as polarity
from complaints_v2 left outer join dictionary d on complaints_v2.word = d.word
WHERE YEAR(c_date_received) BETWEEN 2022 AND 2023
ORDER BY polarity ASC;
```

The following query calculates the overall sentiment for each consumer narrative for each complaint id:

```
create table IF NOT EXISTS complaint_sentiment
stored as orc as
SELECT c_complaint_id, c_date_received,
case
when sum( polarity ) > 0 then 'positive'
when sum( polarity ) < 0 then 'negative'
```

```
else 'neutral' end as sentiment
from complaints_v3
group by c_complaint_id, c_date_received, YEAR(c_date_received)
HAVING YEAR(c_date_received) BETWEEN 2022 AND 2023;
```

The following query assigns values to polarity for proper computation of the data:

```
CREATE TABLE IF NOT EXISTS complaints_bi
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE
LOCATION "/user/dvaishn2/5200_Project_Tables/DV/complaints_bi"
AS SELECT
cc.*,
case cs.sentiment
when 'positive' then 2
when 'neutral' then 1
when 'negative' then 0
end as sentiment
FROM complaints_clean cc LEFT OUTER JOIN complaint_sentiment cs
on cc.c_complaint_id = cs.c_complaint_id
WHERE YEAR(c_date_received) BETWEEN 2022 AND 2023;
```

This is the final sentiment table containing the clean complaints narrative & sentiment values:

```
CREATE TABLE IF NOT EXISTS complaints_sentiment
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE
LOCATION "/user/dvaishn2/5200_Project_Tables/DV/complaints_sentiment"
AS
select c_complaint_id, c_date_received, narrative, c_state, sentiment from complaints_bi
where c_complaint_id IS NOT NULL
```

```
AND ( sentiment =0 or sentiment =1 or sentiment =2)
AND (YEAR(c_date_received) BETWEEN 2022 AND 2023);
```

Get the file on Linux from HDFS:

Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS:

```
hdfs dfs -get /user/dvaishn2/5200_Project_Tables/DV/complaints_sentiment/000*_0
```

Prepare the file(s) in Linux to download to local PC:

Run the following command to check if files are present:

```
ls -al
```

```
-bash-4.2$ hdfs dfs -ls 5200_Project_Tables/DV/complaints_sentiment
Found 24 items
-rw-r--r--   3 dvaishn2 hdfs   10110926 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000000_0
-rw-r--r--   3 dvaishn2 hdfs   10477613 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000001_0
-rw-r--r--   3 dvaishn2 hdfs   10281637 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000002_0
-rw-r--r--   3 dvaishn2 hdfs   10481322 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000003_0
-rw-r--r--   3 dvaishn2 hdfs   10178584 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000004_0
-rw-r--r--   3 dvaishn2 hdfs   10190589 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000005_0
-rw-r--r--   3 dvaishn2 hdfs   10258631 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000006_0
-rw-r--r--   3 dvaishn2 hdfs   10140501 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000007_0
-rw-r--r--   3 dvaishn2 hdfs   10221101 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000008_0
-rw-r--r--   3 dvaishn2 hdfs   10264192 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000009_0
-rw-r--r--   3 dvaishn2 hdfs   10259779 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000010_0
-rw-r--r--   3 dvaishn2 hdfs   10209037 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000011_0
-rw-r--r--   3 dvaishn2 hdfs   10700719 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000012_0
-rw-r--r--   3 dvaishn2 hdfs   10729281 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000013_0
-rw-r--r--   3 dvaishn2 hdfs   10601167 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000014_0
-rw-r--r--   3 dvaishn2 hdfs   10793585 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000015_0
-rw-r--r--   3 dvaishn2 hdfs   10658036 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000016_0
-rw-r--r--   3 dvaishn2 hdfs   10525571 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000017_0
-rw-r--r--   3 dvaishn2 hdfs   10749574 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000018_0
-rw-r--r--   3 dvaishn2 hdfs   10553509 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000019_0
-rw-r--r--   3 dvaishn2 hdfs   11009857 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000020_0
-rw-r--r--   3 dvaishn2 hdfs   10767765 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000021_0
-rw-r--r--   3 dvaishn2 hdfs   10728074 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000022_0
-rw-r--r--   3 dvaishn2 hdfs   10854408 2023-04-18 03:30 5200_Project_Tables/DV/complaints_sentiment/000023_0
```

We need to concatenate the csv files into the local Linux file system before downloading:

```
cat 00000{0..9}_0 00001{0..9}_0 00002{0..4}_0 > senti_out.csv
```

Since all the output files will have similar name file(s) 00000*_0, lets remove this file from the Linux

```
rm 00*_0
```

Copy the file to local PC:

Open another terminal with git bash to download the file to local PC.

Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password

```
scp dvaishn2@144.24.53.159:/home/dvaishn2/senti_out.csv senti_out.csv
```

```
AD+dvaishn2@STU-PF2XXWZK MINGW64 ~
$ scp dvaishn2@144.24.53.159:/home/dvaishn2/senti_out.csv senti_out.csv
dvaishn2@144.24.53.159's password:
senti_out.csv                                                    100%  152MB  62.8MB/s   00:02
```

16. NGram Sentiment analysis - Text Processing for Consumer Complaints Narrative

Description:
This query creates a table called **raw_narrative** with four columns: product, issue, company, and narrative.

The data in the text file is expected to be delimited by tabs (\t).

The purpose of this table is to be a starting point for storing and processing NGrams for the complaint's narrative.

Run the query in Hive:

```
DROP TABLE IF EXISTS raw_narrative;

CREATE TABLE raw_narrative

(

product string,

issue string,

company string,

narrative string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY "\t"

STORED AS TEXTFILE LOCATION '/user/dvaishn2/5200_Project_Tables/DV/raw_narrative';
```

Description:
This query inserts data into the table **raw_narrative** by selecting columns from three other tables - top_products, top_issues, and complaints that match certain criteria.

The SELECT statement concatenates values from the top_products, top_issues, and complaints tables into JSON-formatted strings for the product, issue, company, and narrative columns of raw_narrative.

Based on earlier observations, it was found that the consumers are most dissatisfied with the following product, issue, and company:

- Top company: "EQUIFAX, INC." with the maximum number of complaints.
- Top product: "Credit reporting, credit repair services, or other personal consumer reports" with the maximum number of complaints.
- Top issue: "Incorrect information on your report" with the maximum number of complaints.

Since consumers' personal details like card number, date of birth, account number, etc. are represented as "XXXX" in the c_complaint_narrative column, the regexp_replace function is used to remove all

occurrences of the character 'x' or 'X' from the c_complaint_narrative column before inserting it into the narrative column of raw_narrative.

Run the query in Hive:

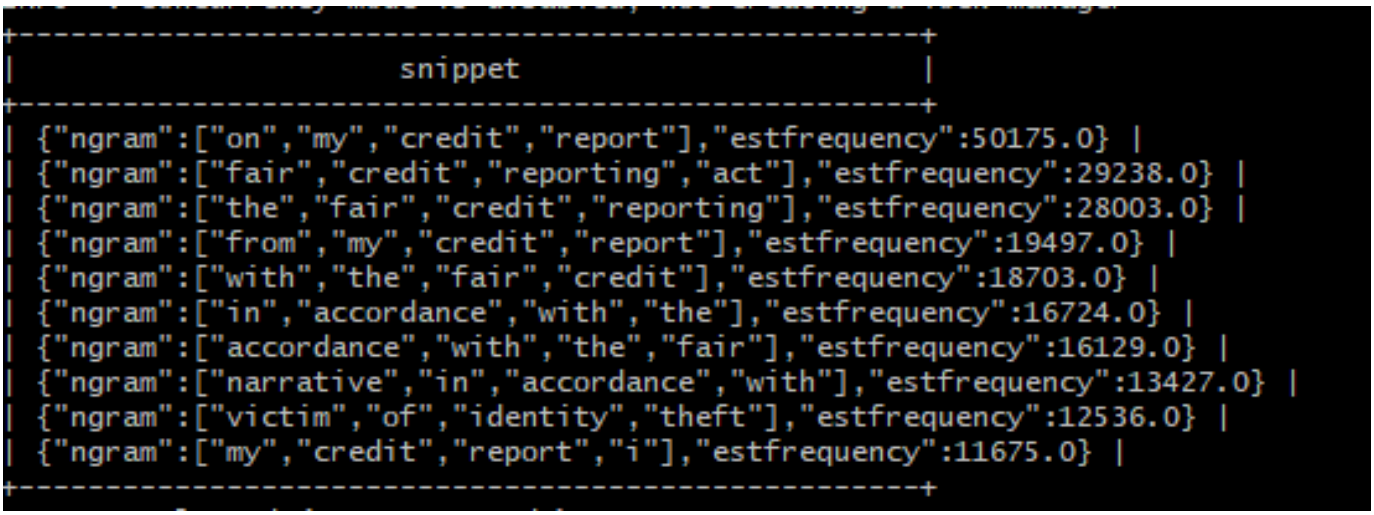
```
INSERT OVERWRITE table raw_narrative

SELECT CONCAT('{ "product": "', tp.product, '"}'), CONCAT('{"issue": "', ti.issue, '"}'),
CONCAT('{"company": "', c_company, '"}'), CONCAT('{"narrative": "',
regexp_replace(c_complaint_narrative,'[xX]', ''), '"}')

FROM mneethi.top_products tp, top_issues ti, complaints

WHERE tp.product = 'Credit reporting, credit repair services, or other personal consumer reports' AND
ti.issue = 'Incorrect information on your report' AND c_company = 'EQUIFAX, INC.';
```

Description:

The following query normalizes all consumer complaint narratives on EQUIFAX, INC's product to lowercase, breaks them into individual words using the SENTENCES function, and passes those to the NGRAMS function to find the 10 most common snippets (four---word combinations).

Run the query in Hive:

```
SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(narrative)), 4, 10))

AS snippet

FROM raw_narrative;
```



The list of words or terms analyzed does not offer much useful information or unique insights. Most of the words on the list are commonly used, and their appearance in the complaints doesn't reveal much about the issues faced by consumers.

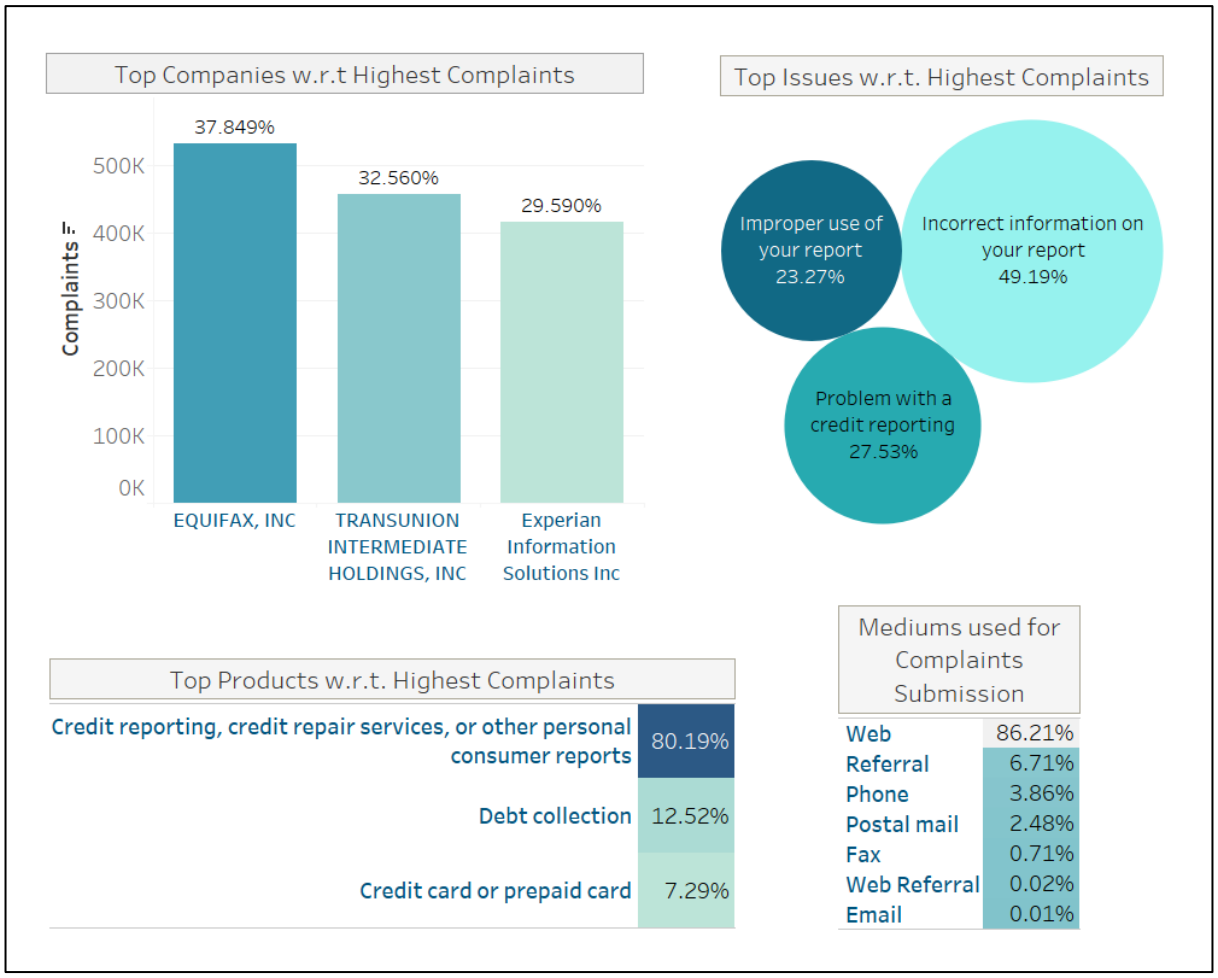
However, "victim of identity theft" stands out in the list. This means that this particular phrase appears more frequently than other phrases or words in the list. This is significant because it suggests that a significant number of consumers who have complained about EQUIFAX, INC. and its credit reporting services have been victims of identity theft.

Therefore, this insight indicates that EQUIFAX, INC. needs to address the issue of identity theft in its credit reporting services. It is an important issue that is affecting many of its consumers and causing them to complain. By addressing this issue, EQUIFAX, INC. can improve its credit reporting services and better serve its customers.

Step 3: Data Visualization Using Tableau

Visualization 1:

A visual representation of Overall Complaints Status based on Highest Number of Complaints received per Company, per Product, Per Issue and how the complaints are submitted (A DASHBOARD)

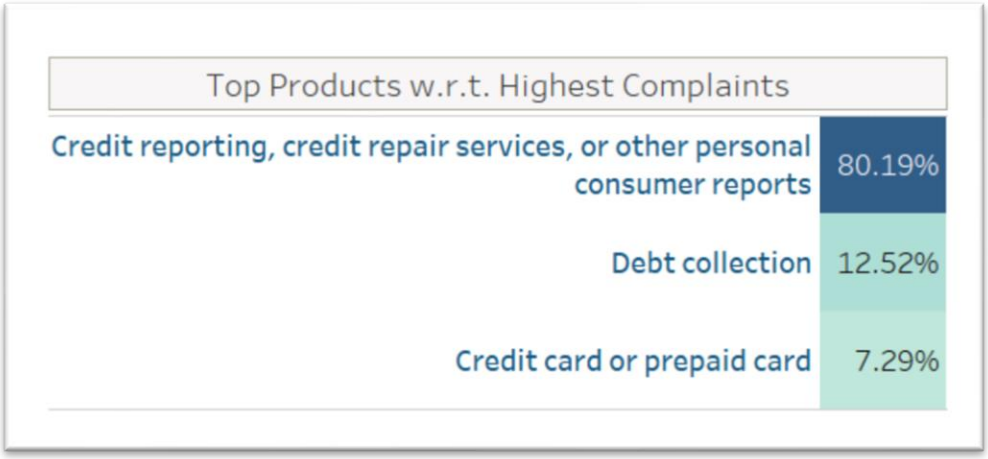


Here are in detail steps to get the above DASHBOARD visualizations:

A visual representation of the PRODUCTS for which the highest number of complaints have received

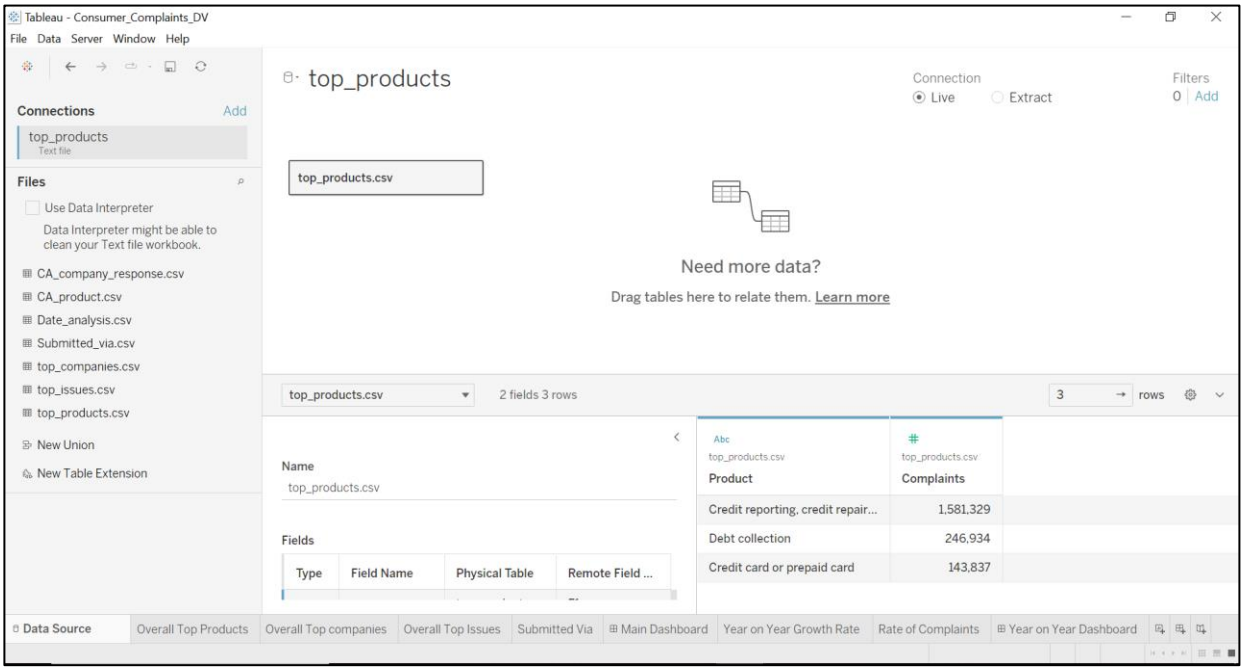
Below is the dataset csv file content:

Credit reporting, credit repair services, or other personal consumer reports	1581329
Debt collection	246934
Credit card or prepaid card	143837

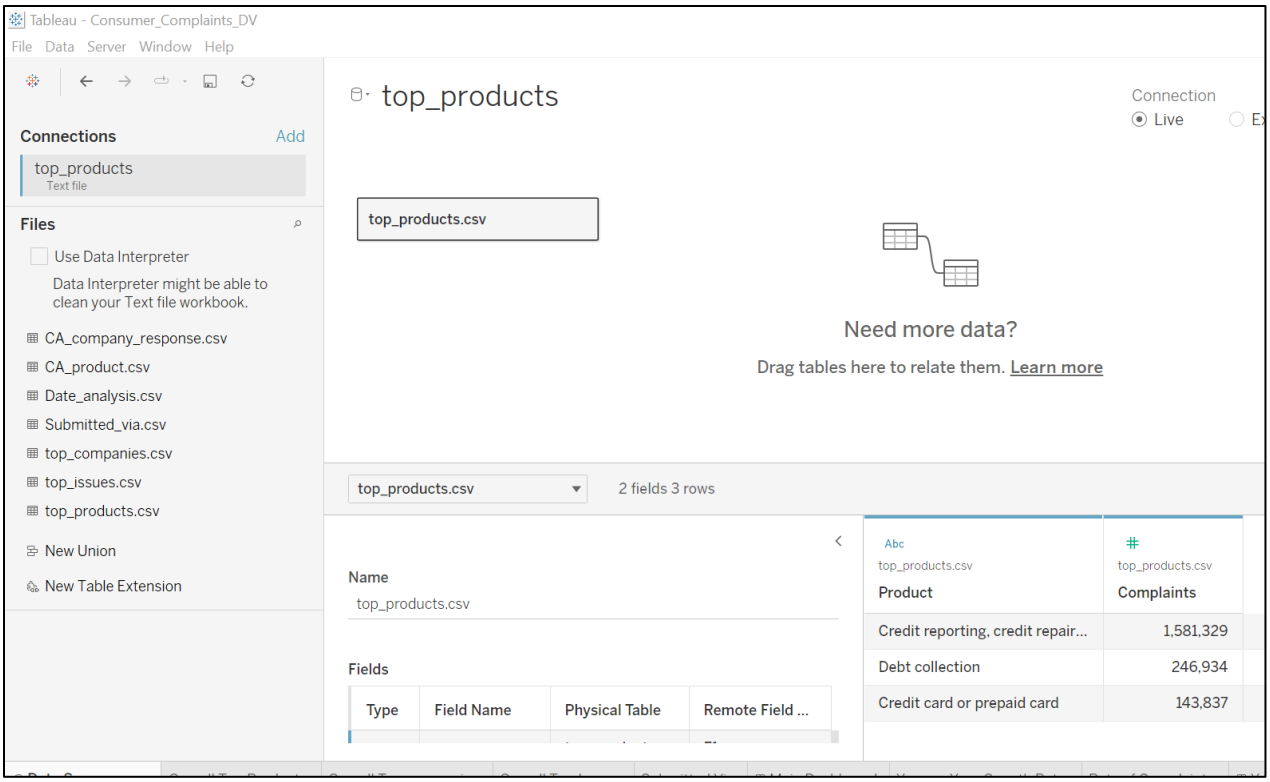


Here are the steps you can take to create a visual representation of the top products based on complaints received:

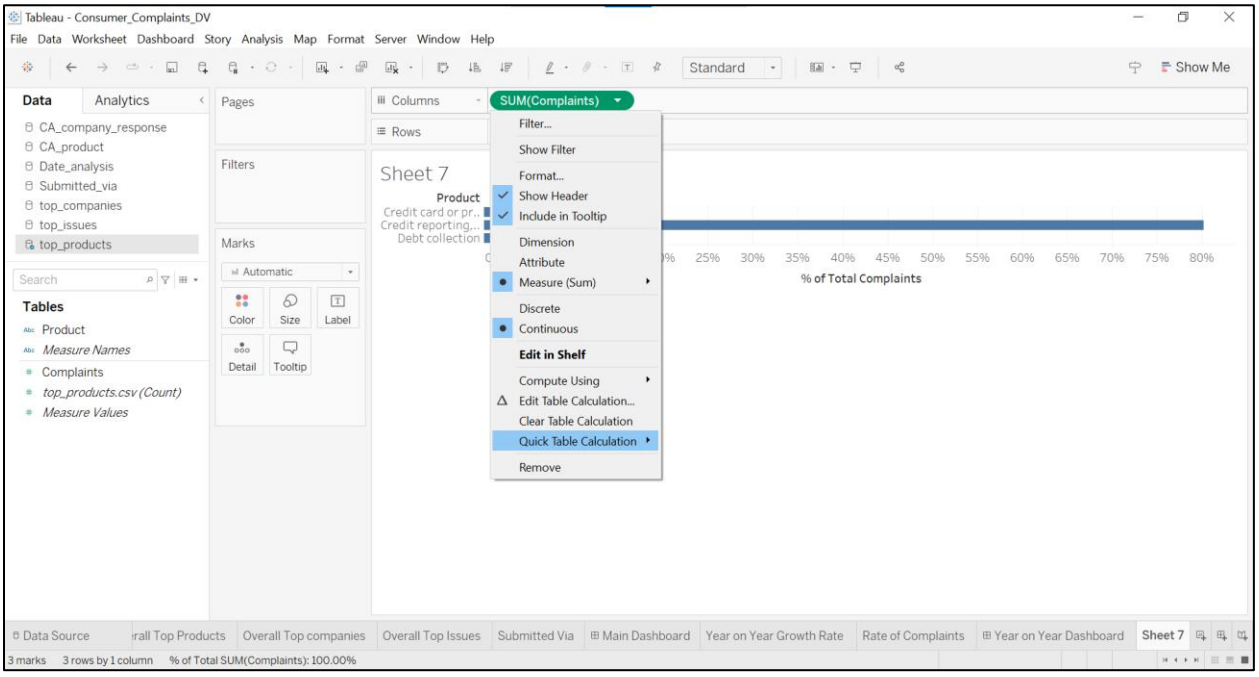
i) Connect to your data source: Open Tableau and connect to **top_products.csv**



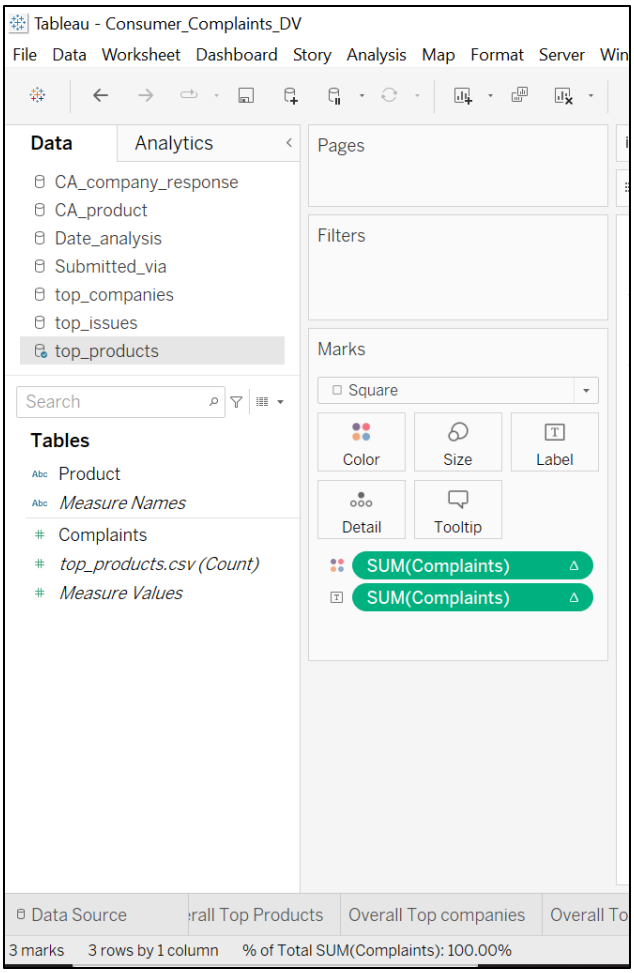
ii) Rename F1, F2 by right clicking each value as F1: Product, F2: Complaints



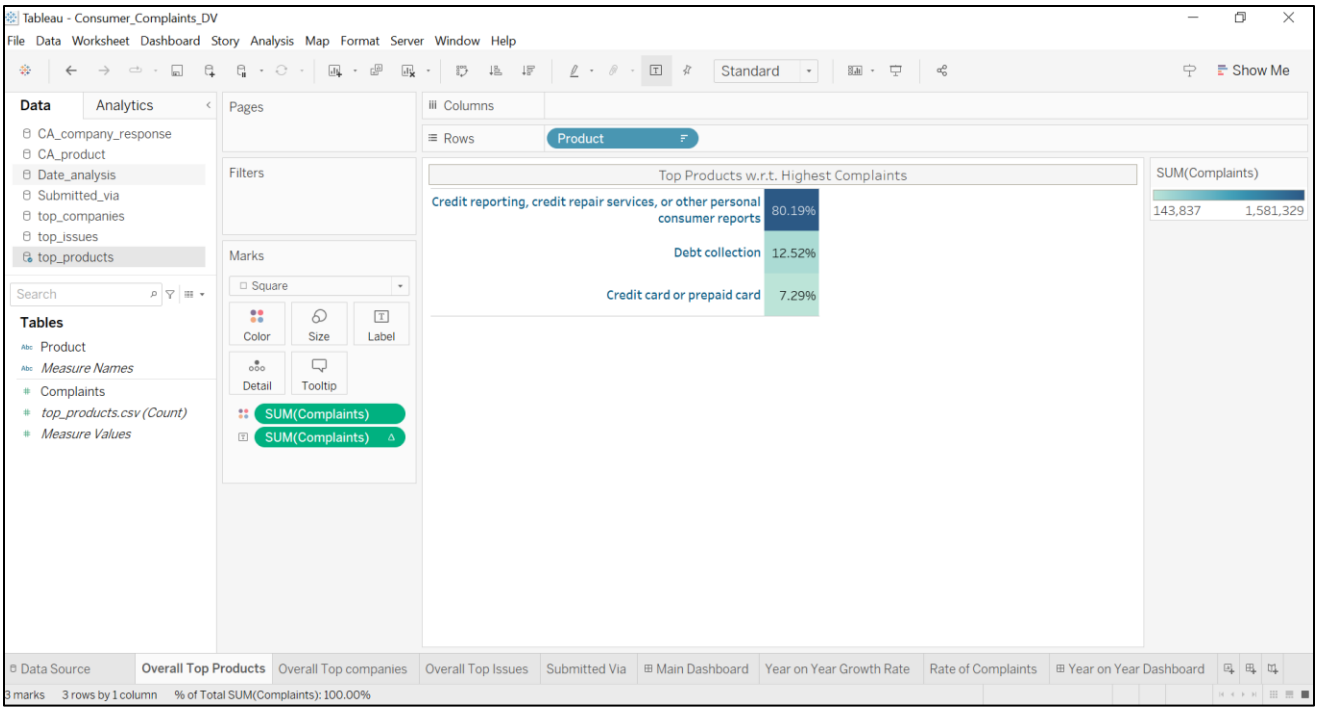
iii) Then you will need to drag **Product** to Rows and **Complaints** to columns. Then you have to click on SUM(Complaint) dropdown menu and select: Quick Table Calculation > Percent of Total



iv) For changing the color of the highlight table, you have to drag SUM(Complaints) to the Marks>Color. And for the Labels, you need to drag SUM(Complaints) to Marks > Label



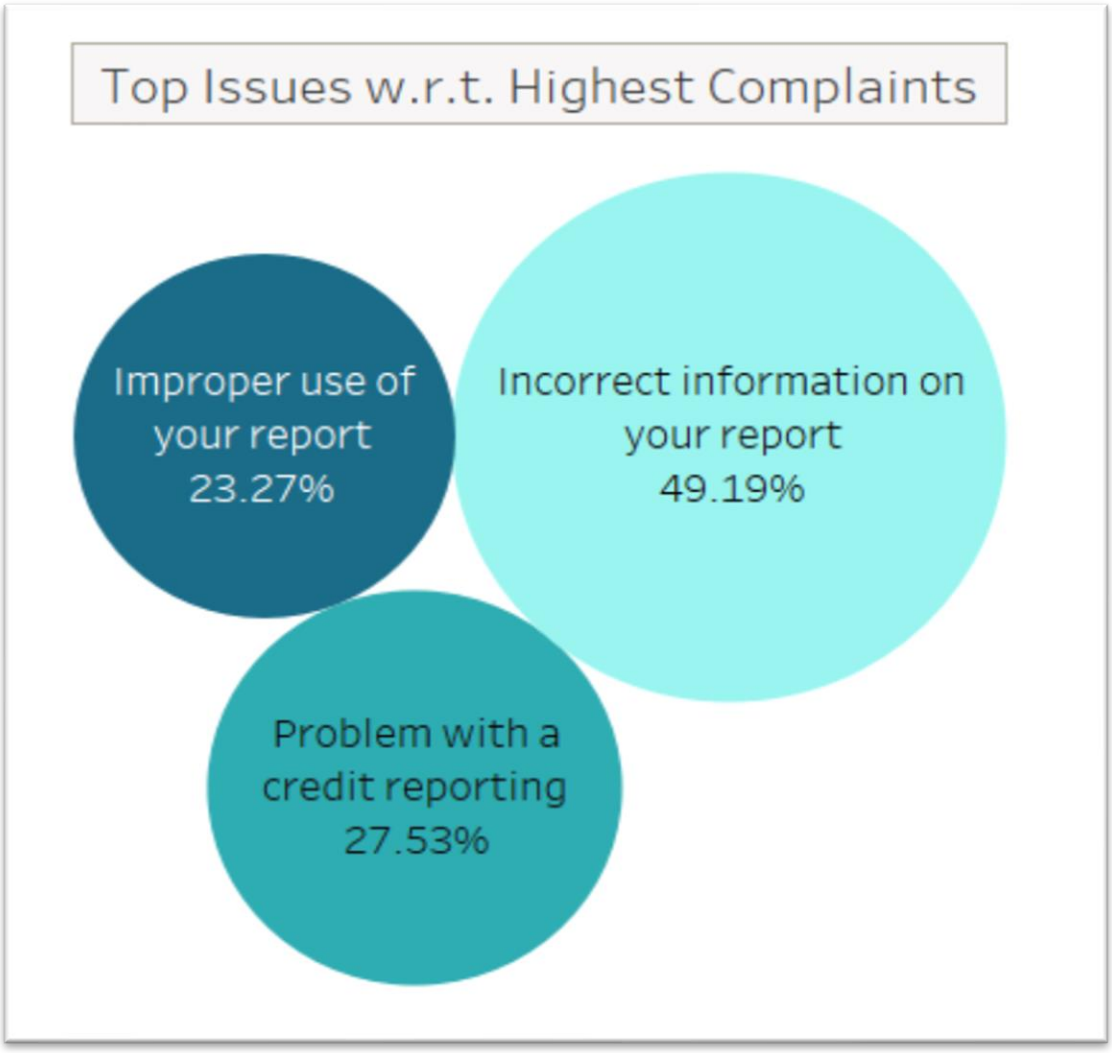
v) Once you are satisfied with your visualization, save it as a Tableau workbook with .Twbx extension.



A visual representation of the ISSUES for which the highest number of complaints have received

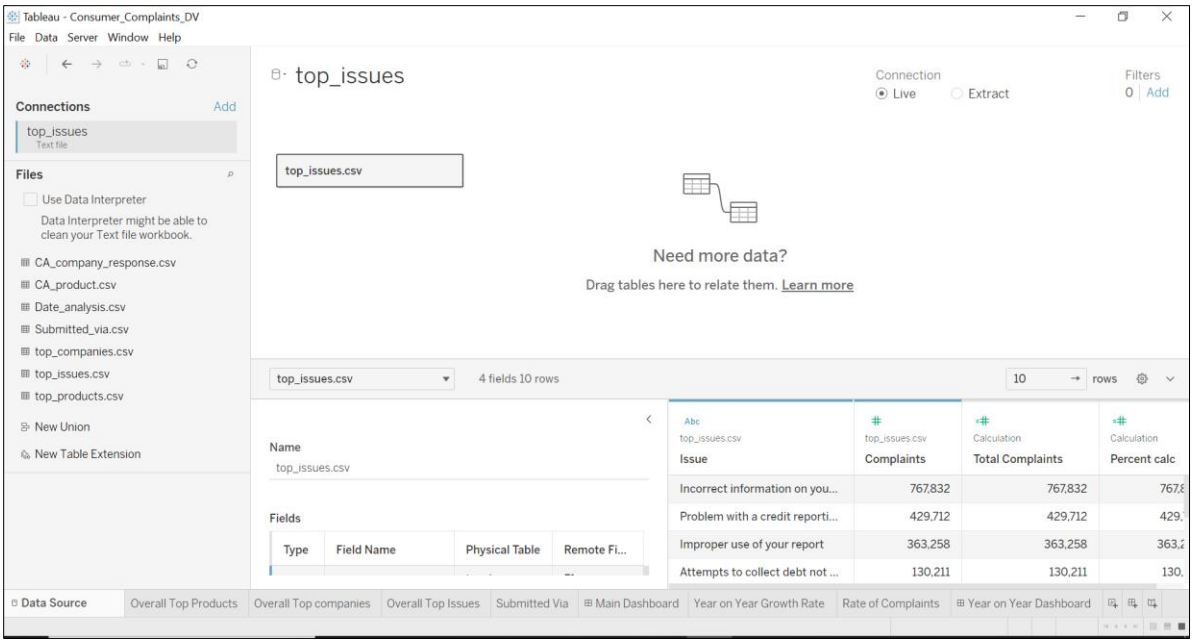
Below is the dataset csv file content:

Incorrect information on your report	767832
Problem with a credit reporting company's investigation into an existing problem	429712
Improper use of your report	363258
Attempts to collect debt not owed	130211
Managing an account	78599
Written notification about debt	53172
Trouble during payment process	49547
Problem with a purchase shown on your statement	36615
Struggling to pay mortgage	25080
Took or threatened to take negative or legal action	20817

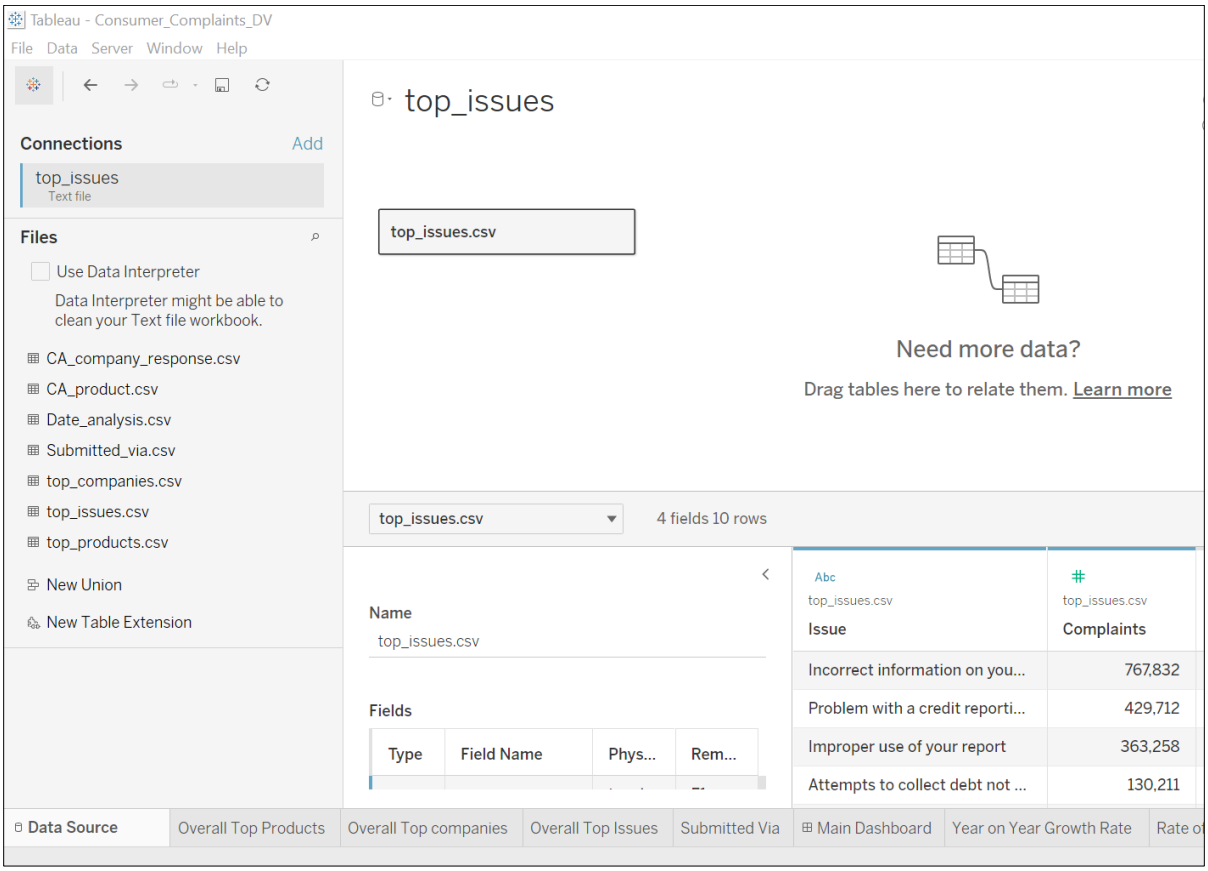


Here are the steps you can take to create a visual representation of the top ISSUES based on complaints received:

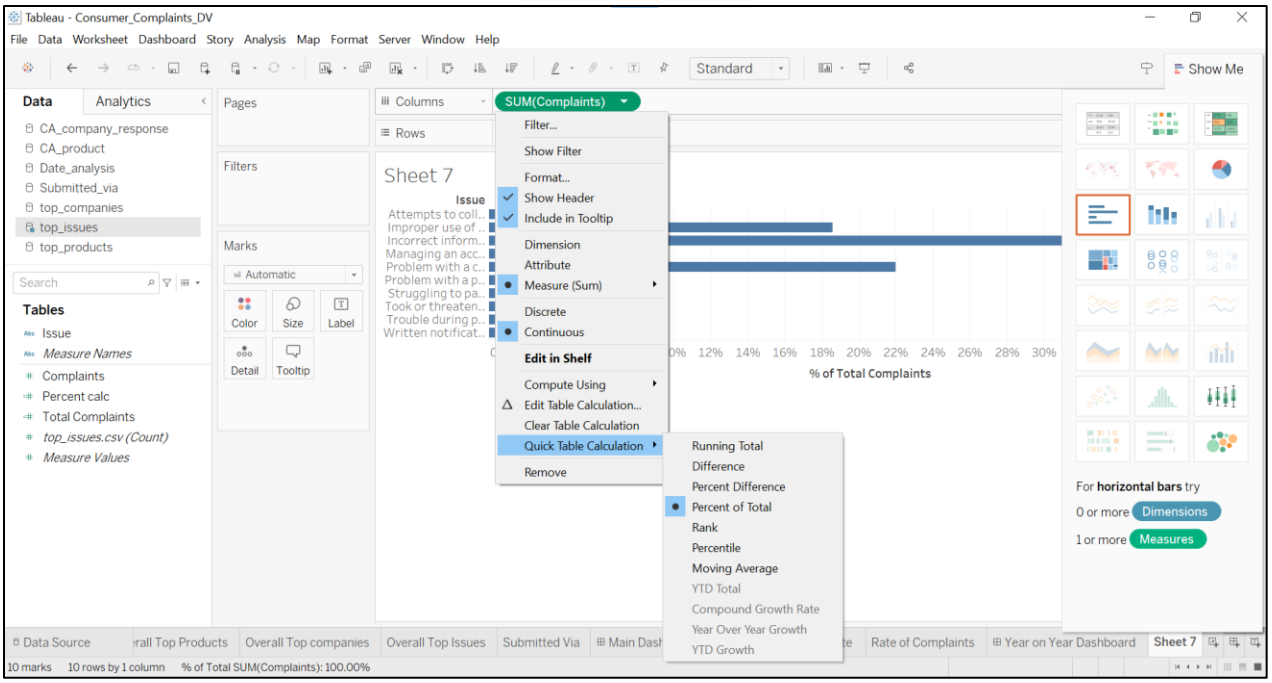
i) Connect to your data source: Open Tableau and connect to **top_issues.csv**



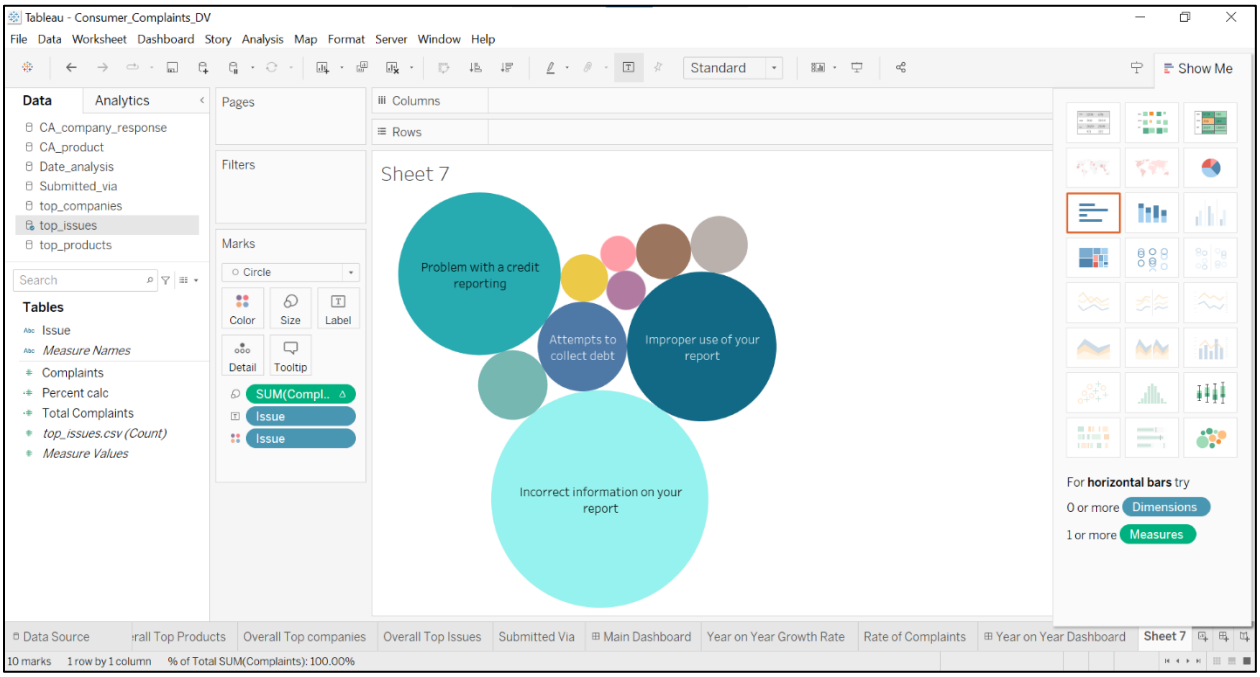
ii) Rename F1, F2 by right clicking each value as F1: Issue, F2: Complaints



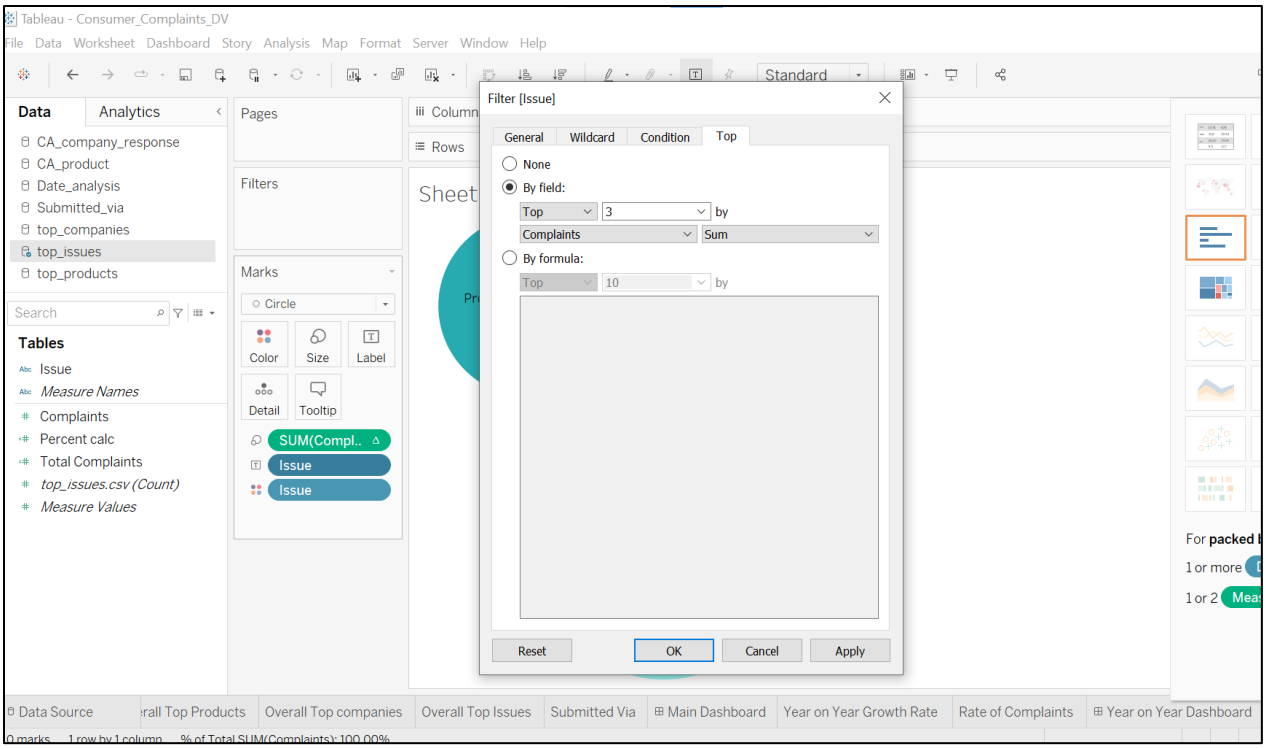
iii) Then you will need to drag **Issue** to Rows and **Complaints** to columns. Then you have to click on SUM(Complaint) dropdown menu and select: Quick Table Calculation > Percent of Total



iv) Change the chart type to **packed bubbles** from **Show Me** toolbar to the right.

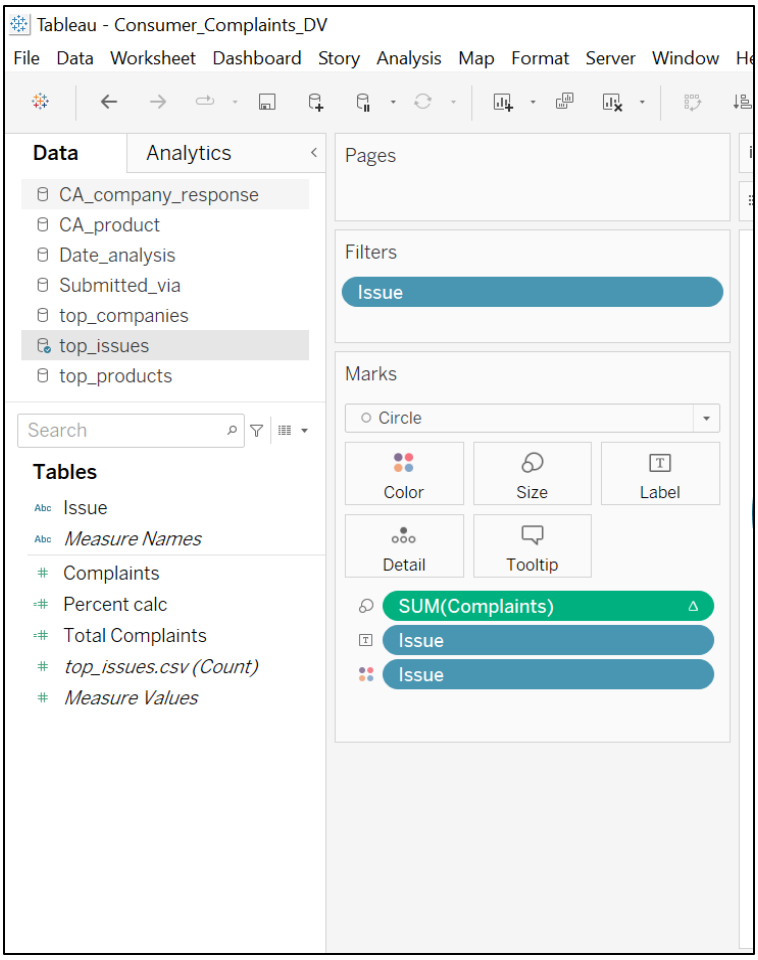


v) Filter the **top 3 issues** to properly show the insights by clicking on Issue > Filter

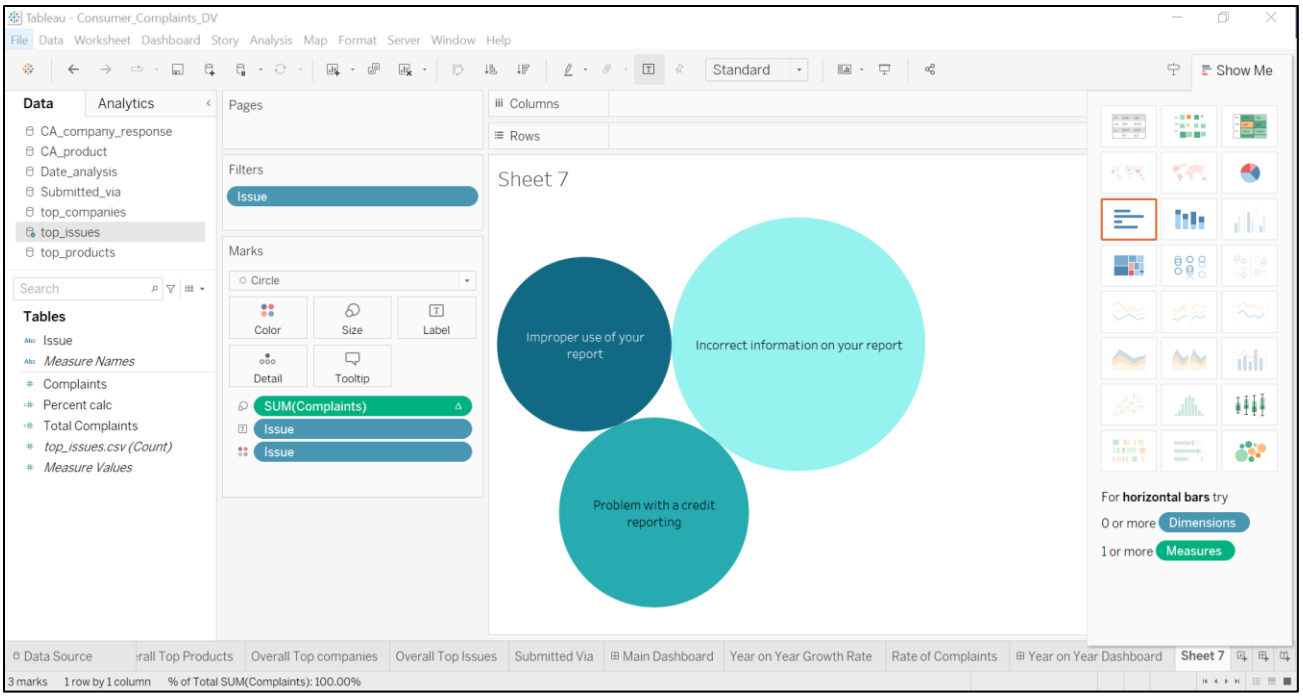


vi) For changing the color of the highlight table, you have to drag **Issue** to the Marks>Color. And for the Labels, you need to drag **Issue** to Marks > Label.

Note: The Size of the bubbles are automatically adjusted based on SUM(Complaints) once we selected the chart type to packed bubbles.



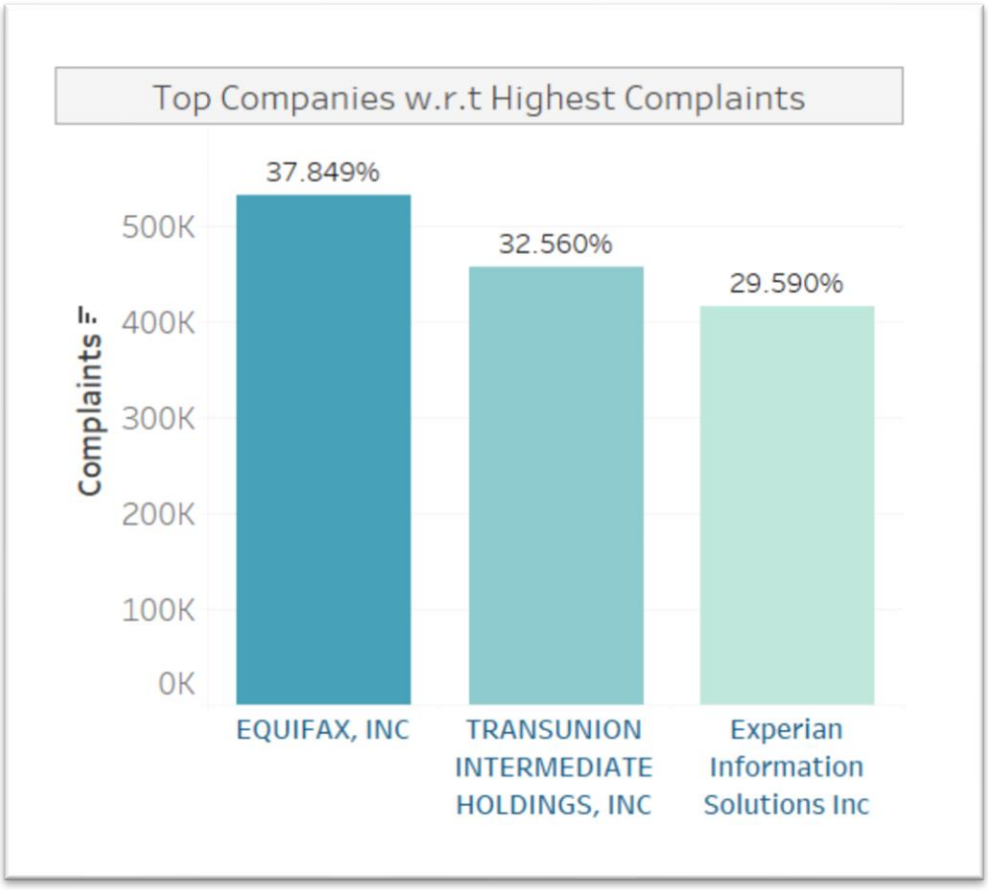
vii) Once you are satisfied with your visualization, save it as a Tableau workbook with .Twbx extension.



A visual representation of the COMPANIES for which the highest number of complaints have received

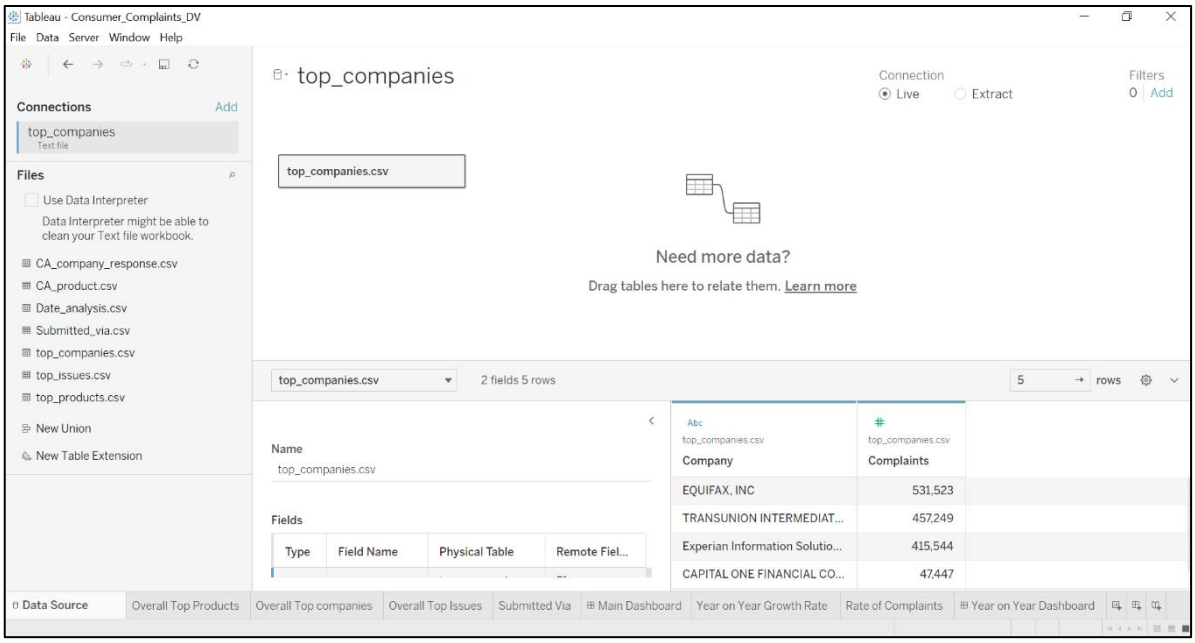
Below is the dataset csv file content:

EQUIFAX, INC	531523
TRANSUNION INTERMEDIATE HOLDINGS, INC	457249
Experian Information Solutions Inc	415544
CAPITAL ONE FINANCIAL CORPORATION	47447
BANK OF AMERICA, NATIONAL ASSOCIATION	44899

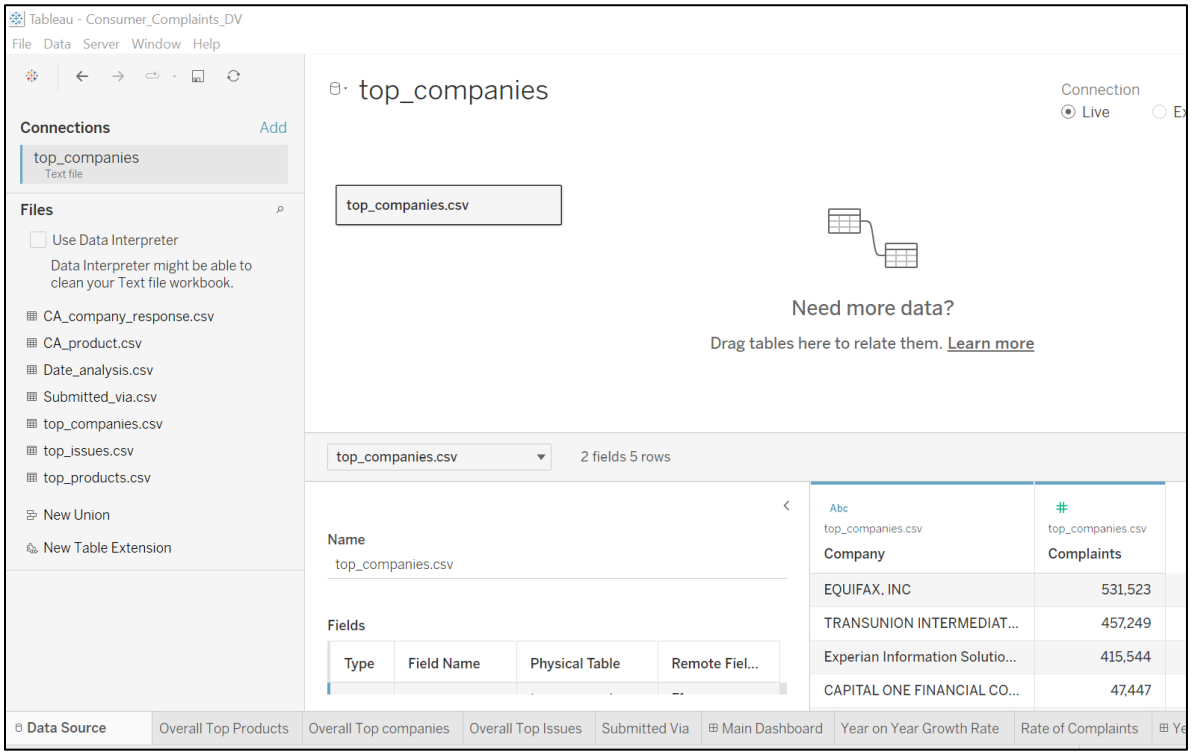


Here are the steps you can take to create a visual representation of the top COMPANIES based on complaints received:

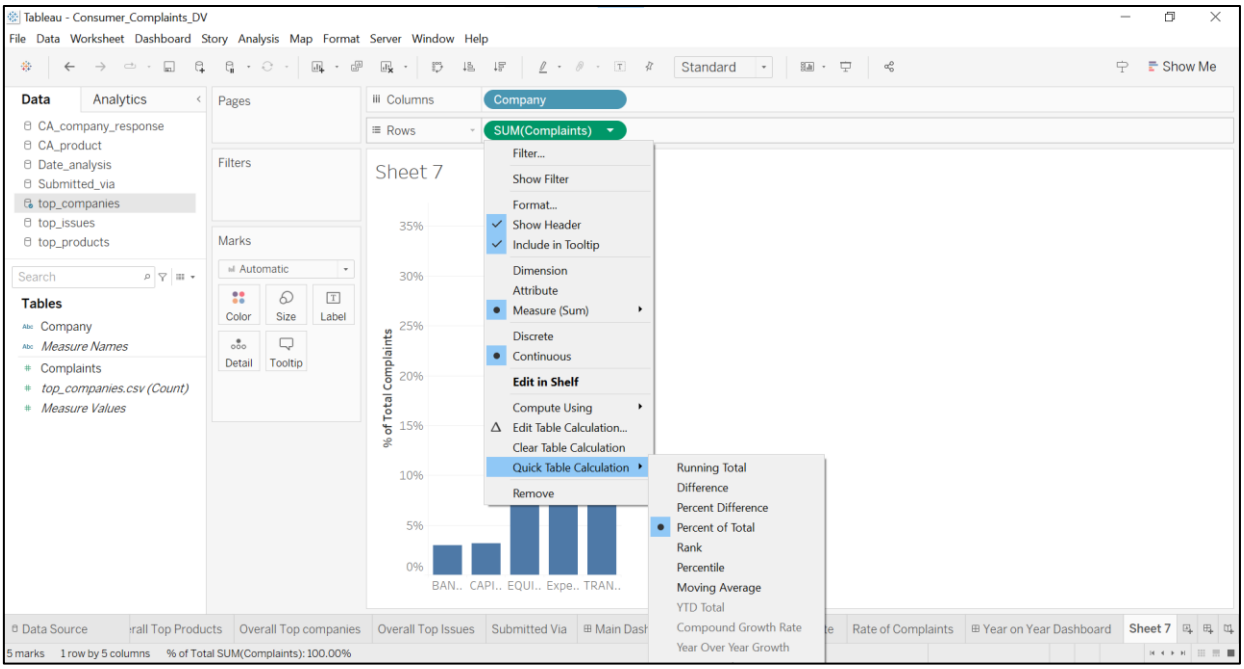
i) Connect to your data source: Open Tableau and connect to **top_companies.csv**



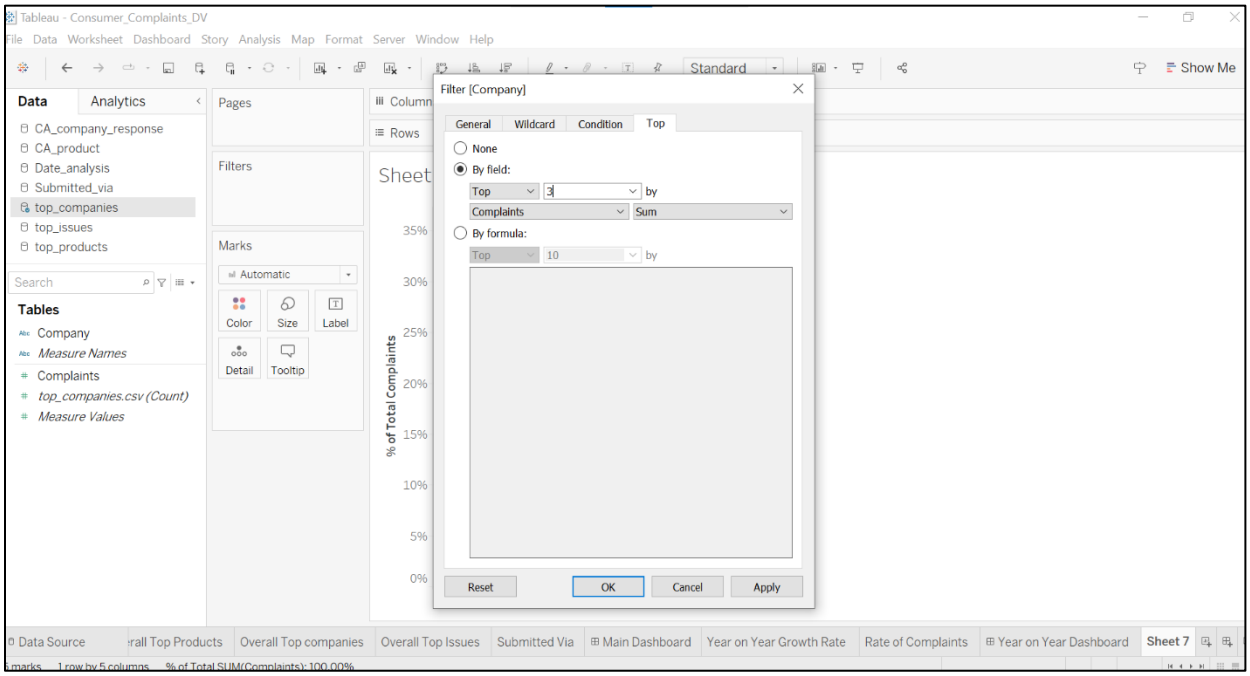
ii) Rename F1, F2 by right clicking each value as F1: Company, F2: Complaints



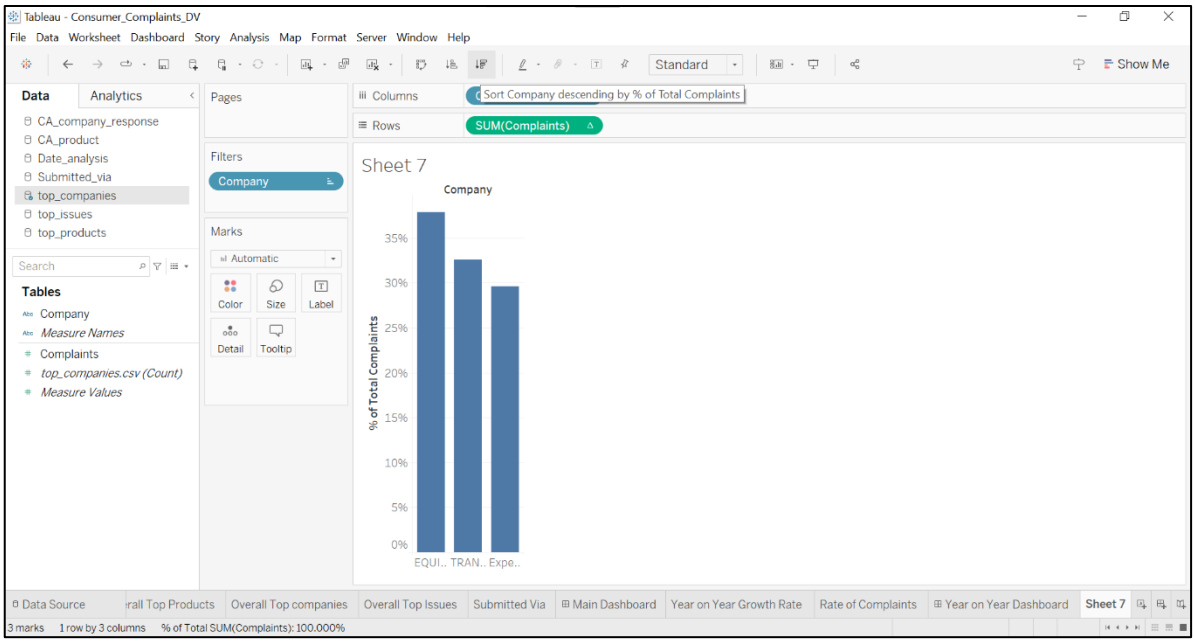
iii) Then you will need to drag **Company** to Rows and **Complaints** to columns. Then you have to click on SUM(Complaint) dropdown menu and select: Quick Table Calculation > Percent of Total



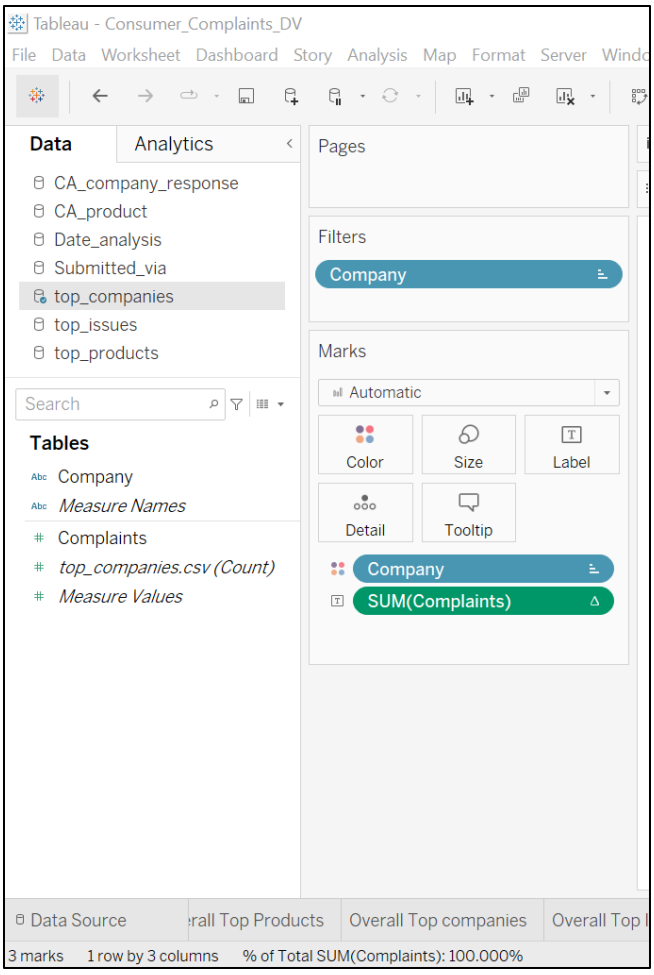
iv) Filter the **top 3 companies** to properly show the insights by clicking on Company > Filter



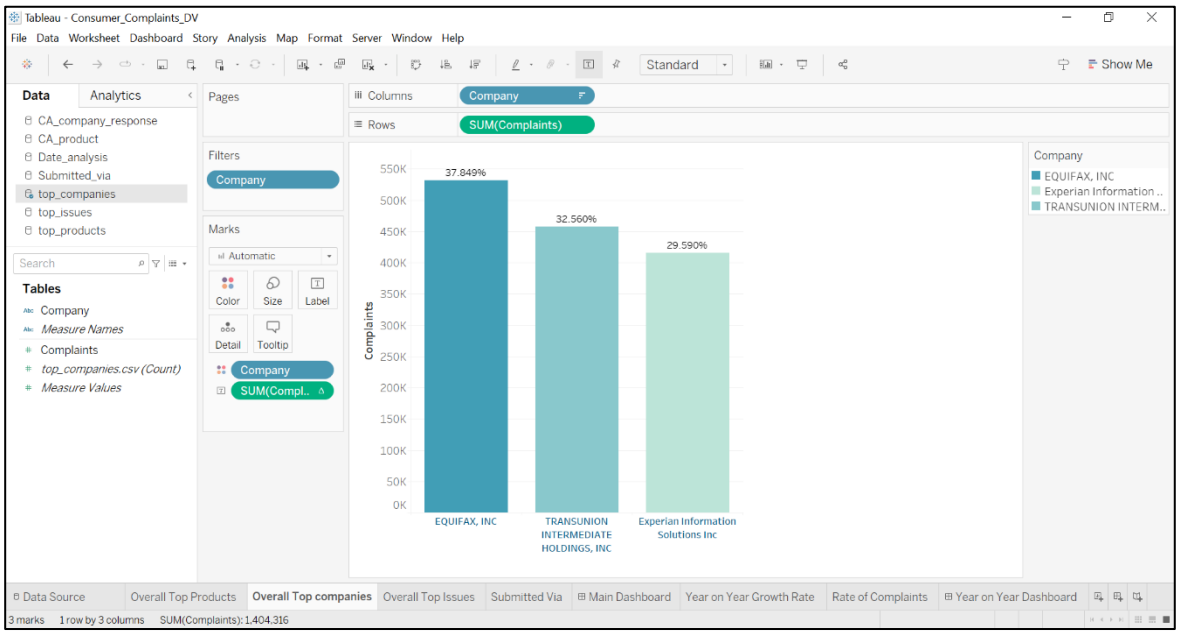
v) **Sort** the companies in descending order by clicking on the button as shown below



vi) For changing the color of the bar chart, you have to drag **Company** to the Marks>Color. And for the Labels, you need to drag **SUM(Complaints)** to Marks > Label. To get the Percentage, you need to click on SUM(Complaints) > Quick Table Calculation > Percent of Total as shown in previous steps.



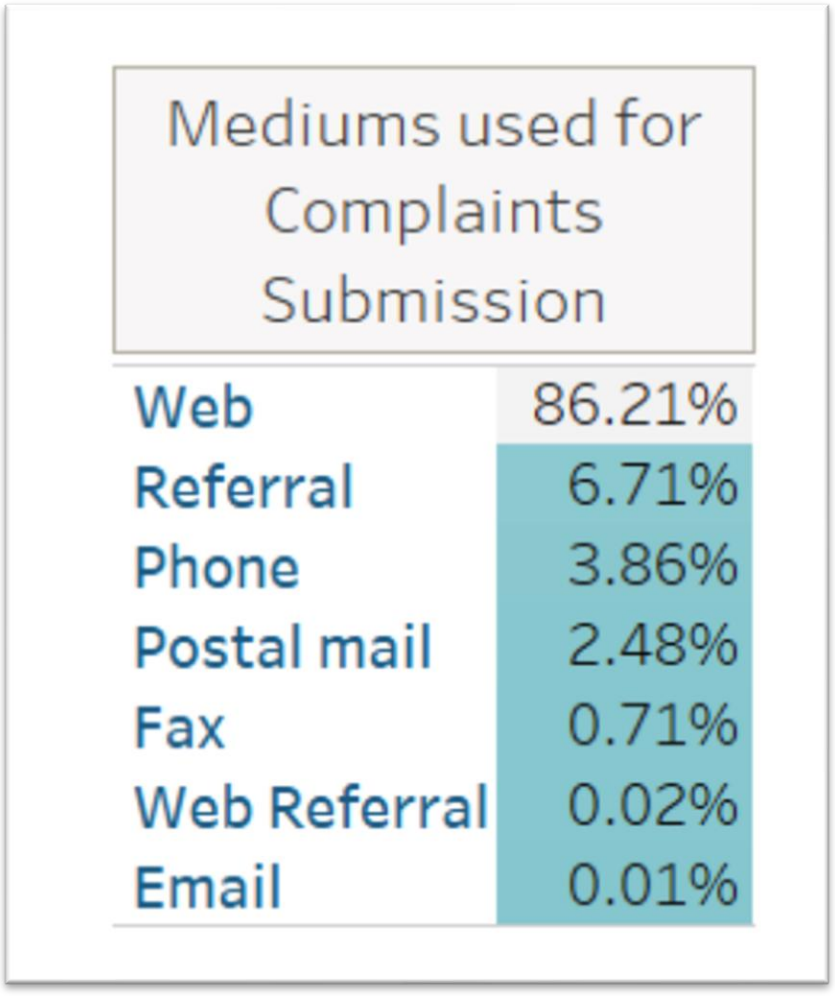
vii) Once you are satisfied with your visualization, save it as a Tableau workbook with. Twbx extension.



A visual representation of how many complaints have been SUBMITTED VIA WHICH MODE

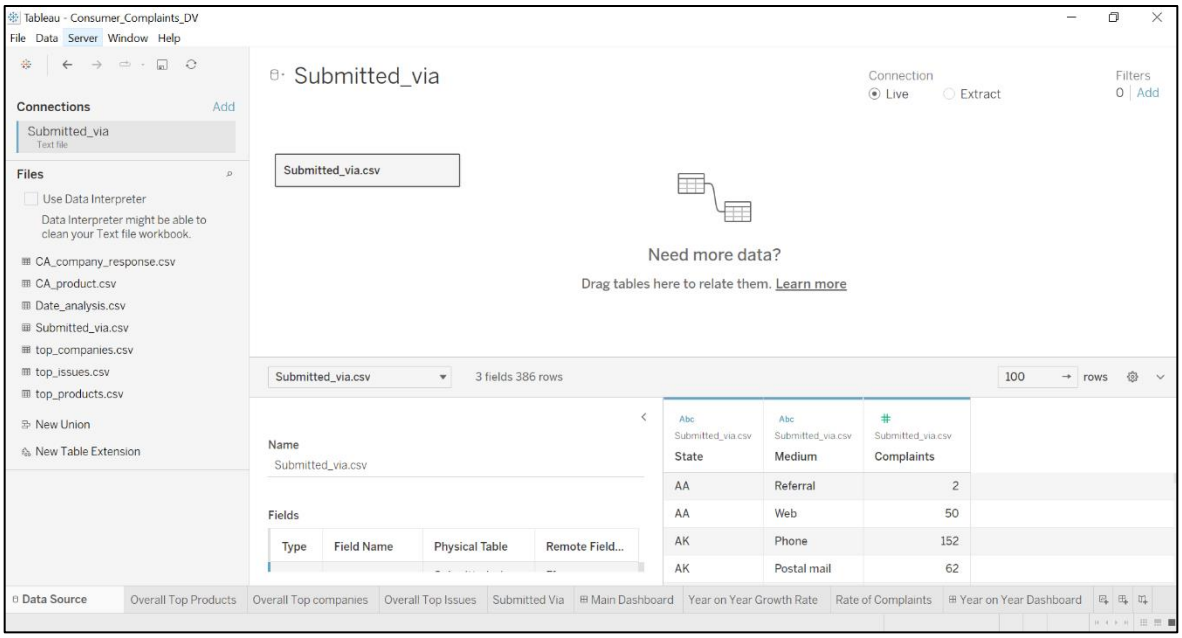
Below is the dataset csv file content:

AA	Referral	2
AA	Web	50
AK	Phone	152
AK	Postal mail	62
AL	Postal mail	919
AL	Referral	1957
AL	Web Referral	2
AP	Fax	3
AP	Phone	6
AP	Postal mail	4
AP	Referral	20
AR	Postal mail	447
AS	Phone	5
AS	Postal mail	2
AZ	Fax	303
AZ	Phone	2529
AZ	Postal mail	1597
AZ	Web	53755
CA	Fax	3075
CA	Web	355661
CA	Web Referral	3
CO	Email	4
CO	Phone	1796
CO	Postal mail	1112
CO	Referral	3741
CO	Web	34338
CO	Web Referral	1
CT	Phone	1831
CT	Web Referral	144
DC	Phone	668
DC	Postal mail	455
DC	Referral	1076
DC	Web Referral	1

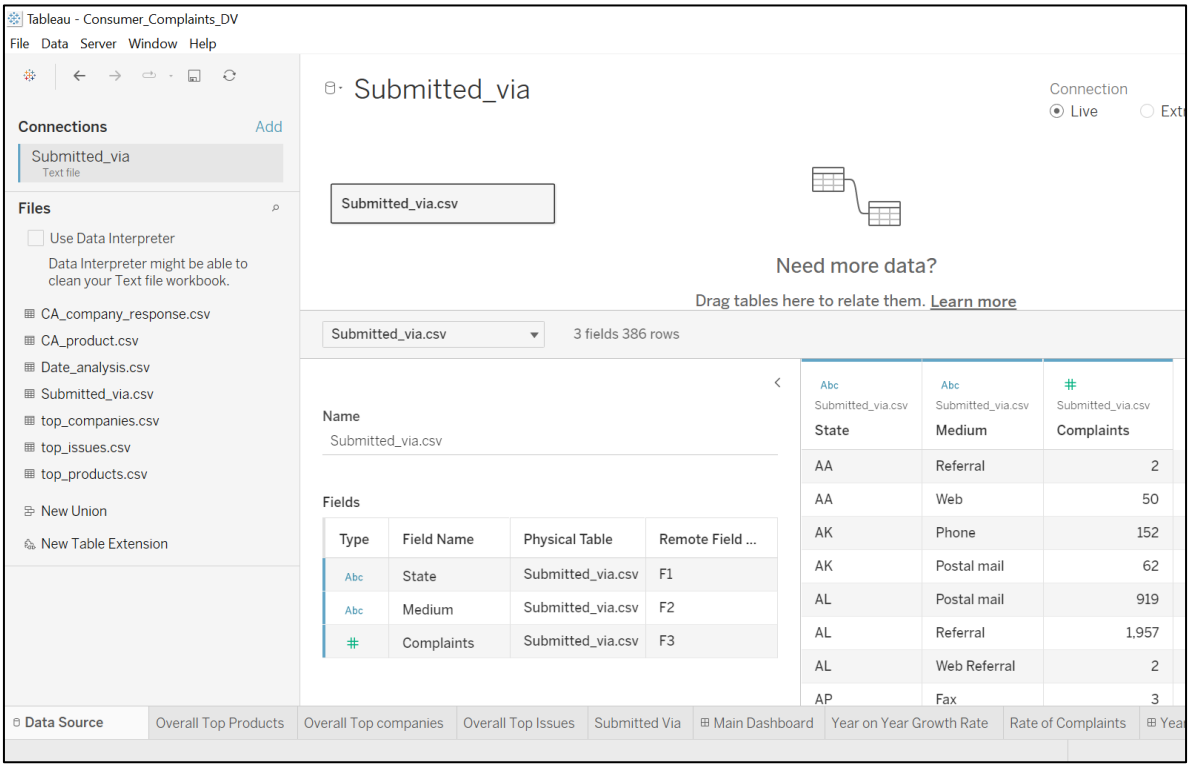


Here are the steps you can take to create a visual representation of the SUBMITTED VIA different modes:

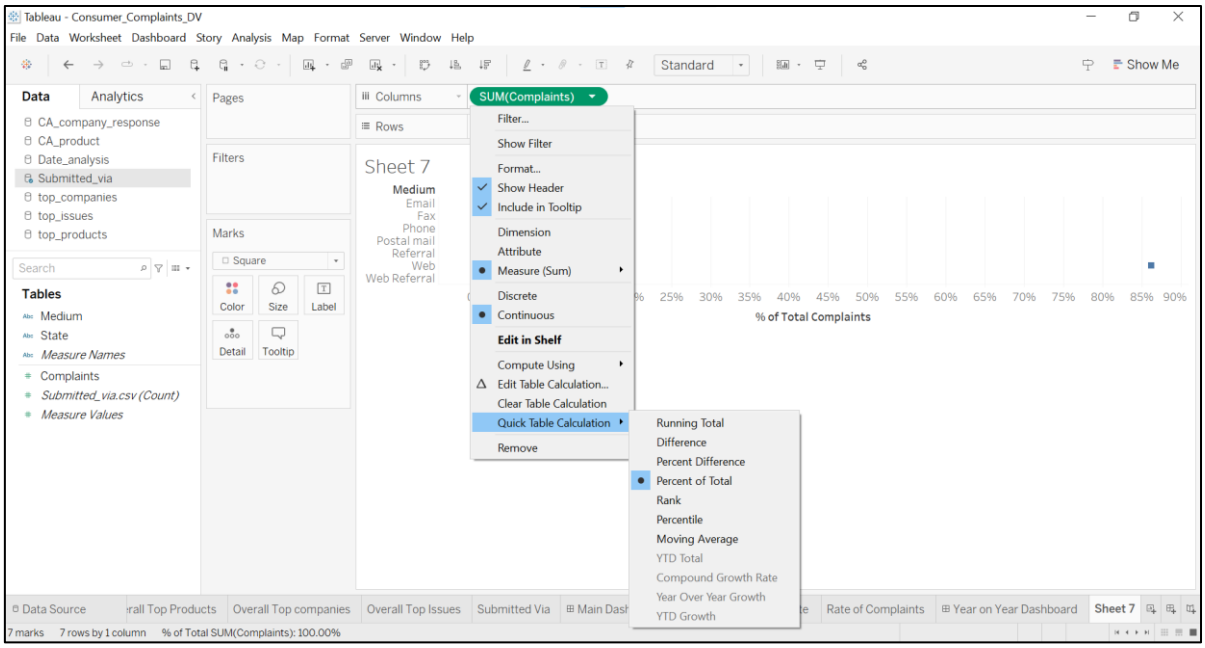
i) Connect to your data source: Open Tableau and connect to **submitted_via.csv**



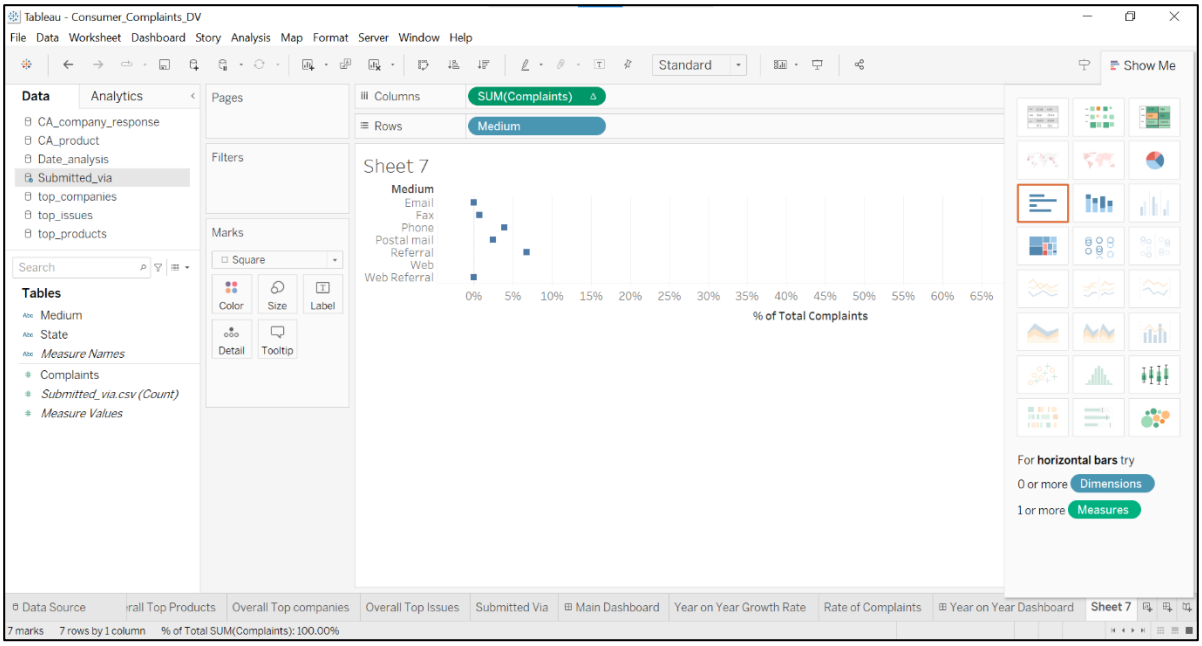
ii) Rename F1, F2, F3 by right clicking each value as F1: State, F2: Medium, F3: Complaints



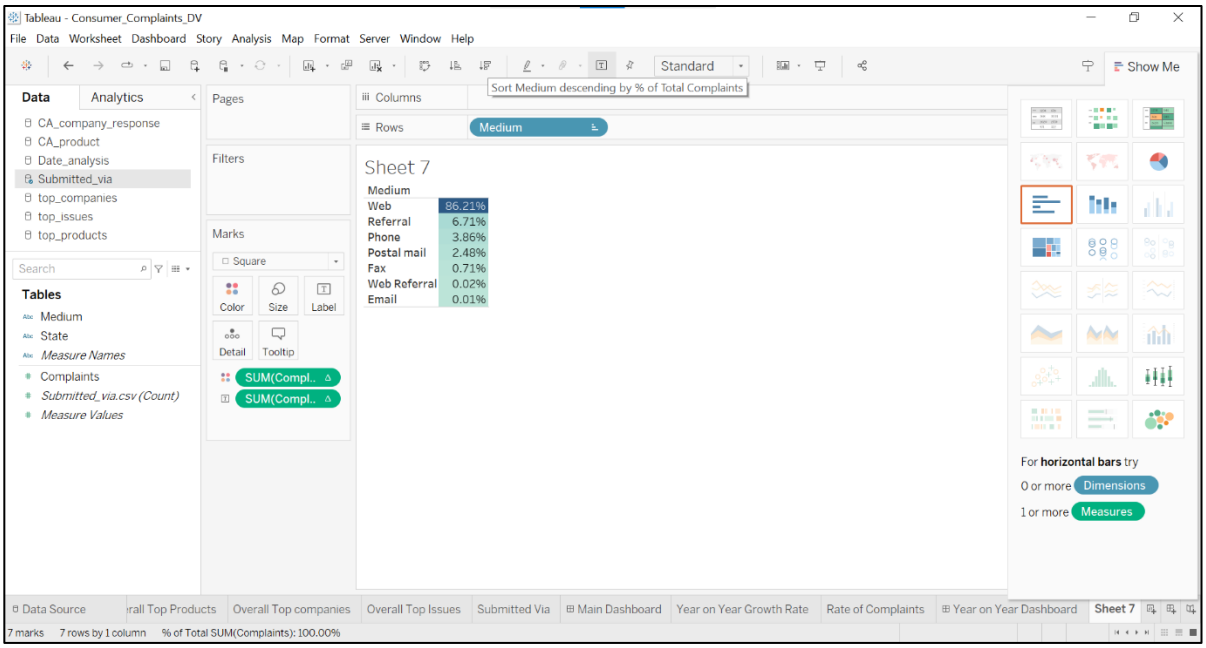
iii) Then you will need to drag **Medium** to Rows and **Complaints** to columns. Then you have to click on SUM(Complaint) dropdown menu and select: Quick Table Calculation > Percent of Total



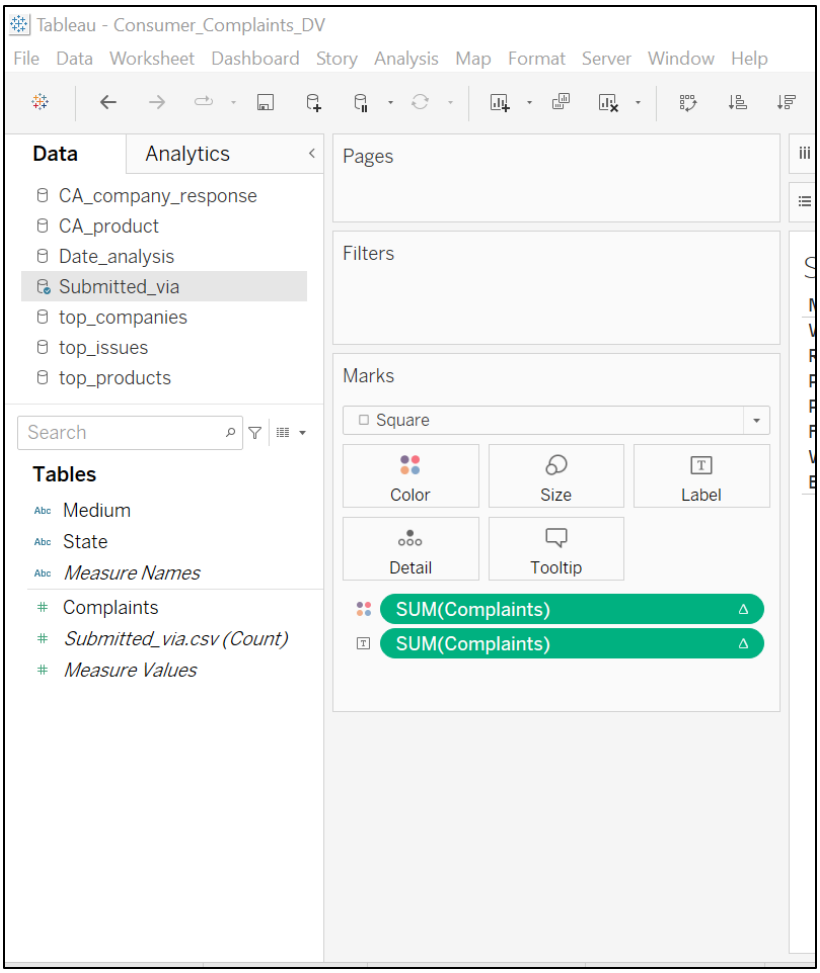
iv) Change the type of chart to Highlight Tables by clicking on Show Me toolbar to the right



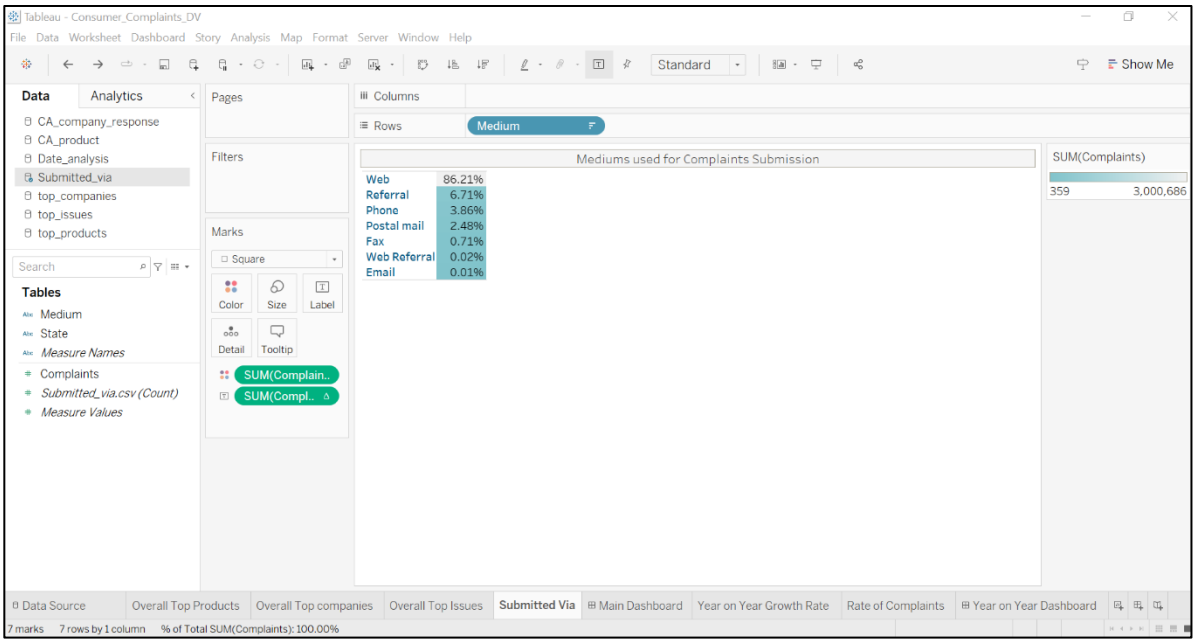
v) **Sort** the table based on % of total values in descending order by clicking on the button as shown below



vi) For changing the color of the bar chart, you have to drag **SUM(Complaints)** to the Marks>Color. And for the Labels, you need to drag **SUM(Complaints)** to Marks > Label. To get the Percentage, you need to click on SUM(Complaints) > Quick Table Calculation > Percent of Total as shown in previous steps.

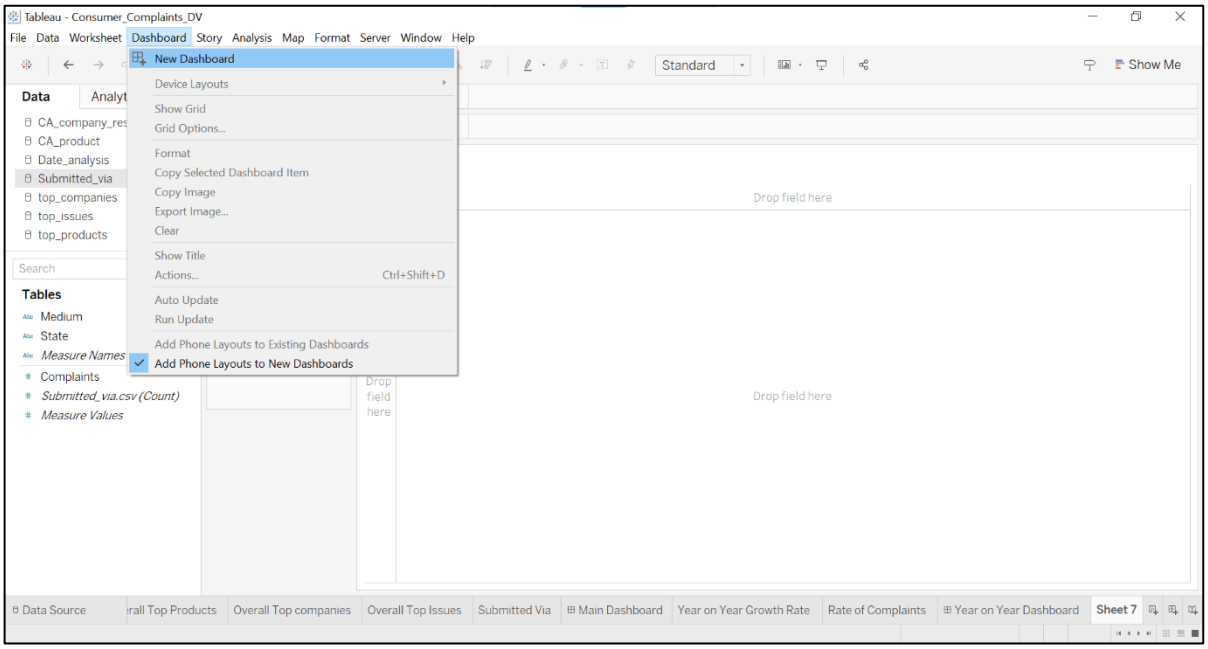


vii) Once you are satisfied with your visualization, save it as a Tableau workbook with. Twbx extension.

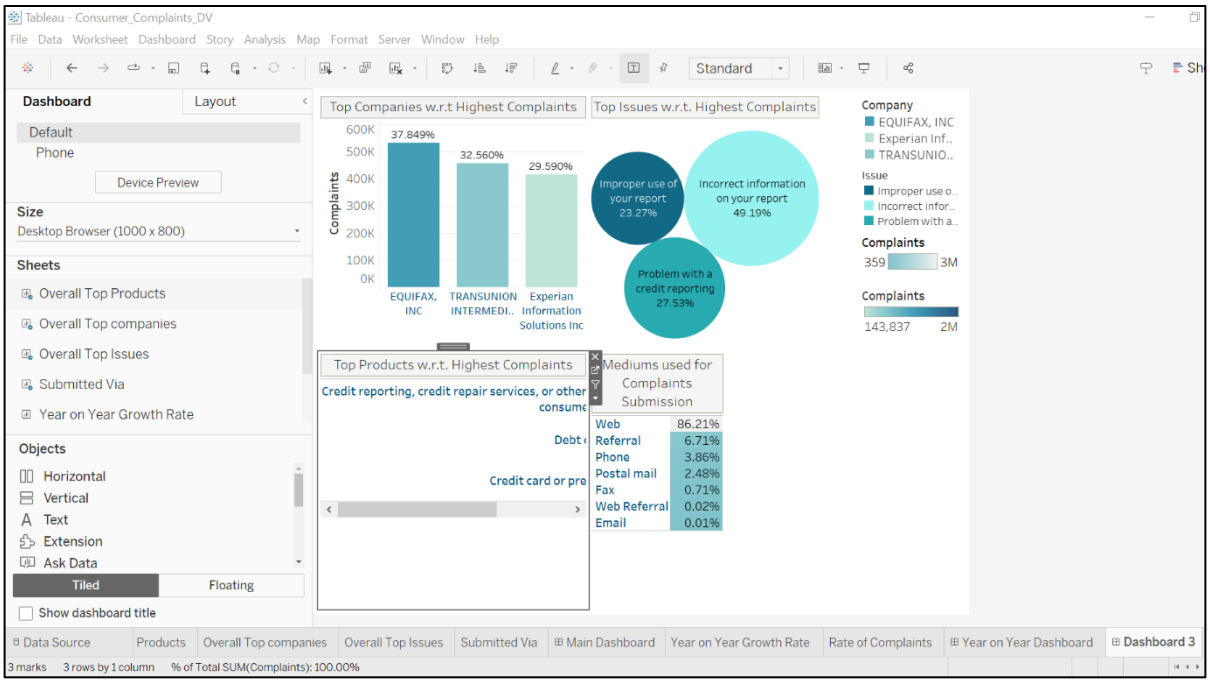


Now, we will put together all 4 visualizations shown above into a DASHBOARD

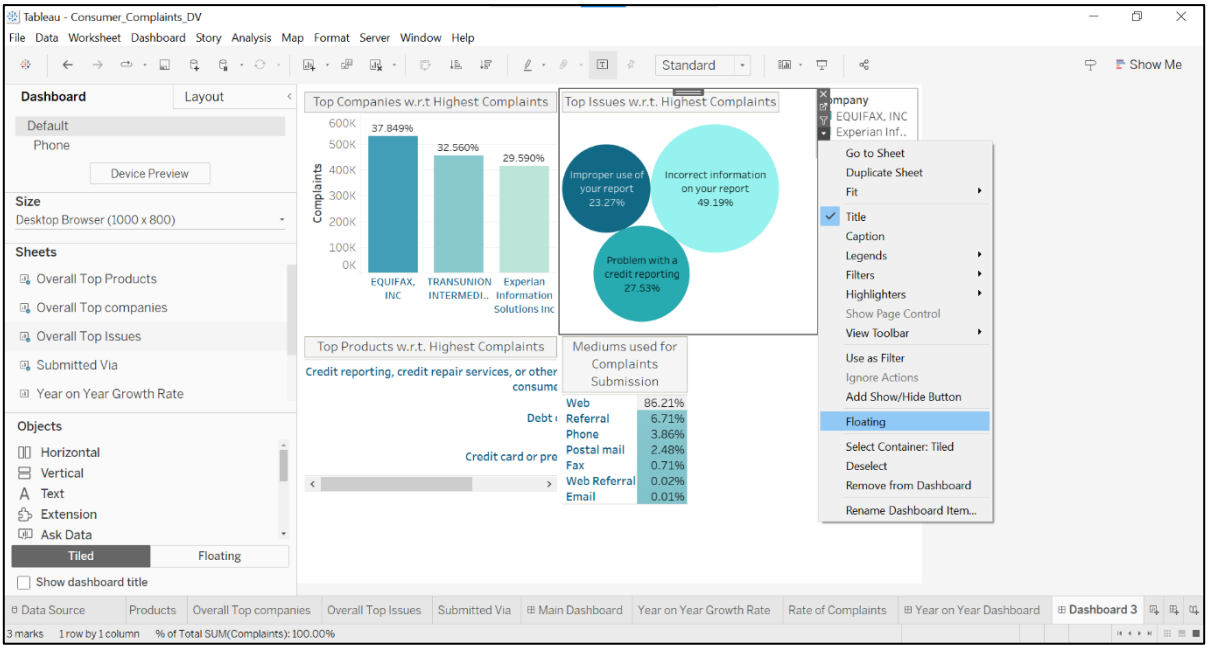
i) In Tableau, create a new Dashboard as shown below



ii) From the left side **Sheets tool bar**, drag all 4 sheets one by one as shown below



iii) To place them to fit in the window as per your need, make the windows **floating** by clicking on each component and select floating as shown below



iv) Change the positions, format the views, and save the tableau file with. twbx extension with the desired Dashboard once done

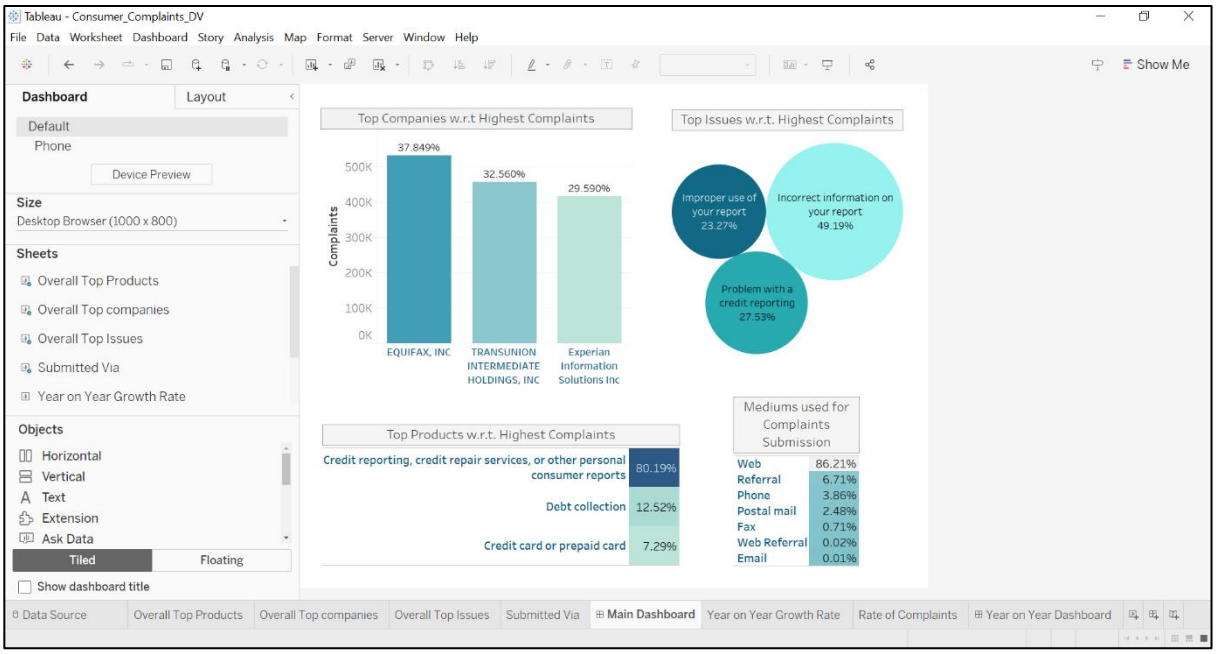
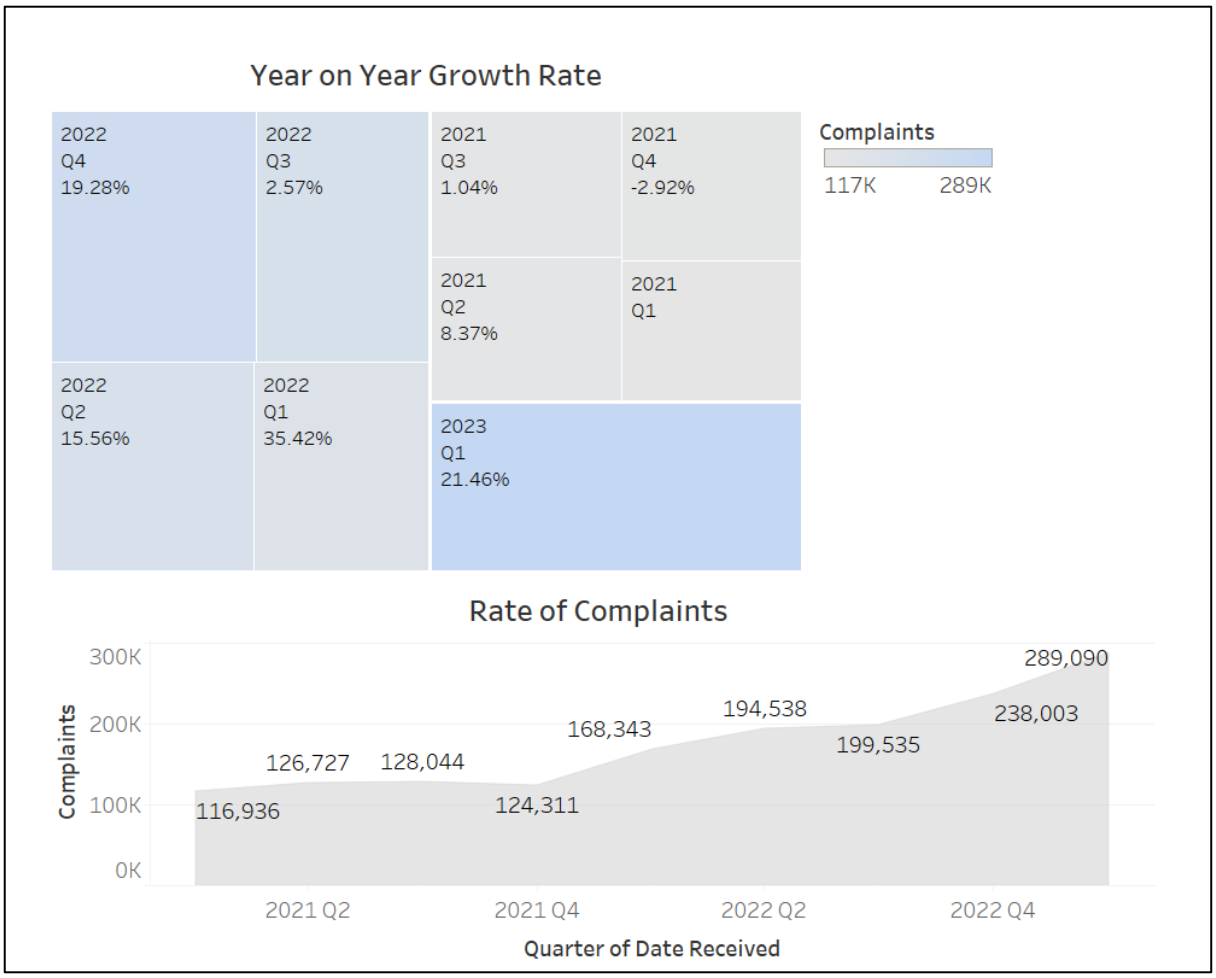


Tableau - Consumer_Complaints_DV
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Show Me
Dashboard
Layout
Default
Phone
Device Preview
Size
Desktop Browser (1000 x 800)
Sheets
Overall Top Products
Overall Top companies
Overall Top Issues
Submitted Via
Year on Year Growth Rate
Objects
Horizontal
Vertical
Text
Extension
Ask Data
Tiled
Floating
Show dashboard title
Top Companies w.r.t Highest Complaints
37.849%
32.560%
29.590%
Complaints
500K
400K
300K
200K
100K
0K
EQUIFAX, INC
TRANSUNION INTERMEDIATE HOLDINGS, INC
Experian Information Solutions Inc
Top Issues w.r.t. Highest Complaints
Improper use of your report 23.27%
Incorrect information on your report 49.19%
Problem with a credit reporting 27.53%
Top Products w.r.t. Highest Complaints
Credit reporting, credit repair services, or other personal consumer reports 80.19%
Debt collection 12.52%
Credit card or prepaid card 7.29%
Mediums used for Complaints Submission
Web 86.21%
Referral 6.71%
Phone 3.86%
Postal mail 2.48%
Fax 0.71%
Web Referral 0.02%
Email 0.01%
Data Source
Overall Top Products
Overall Top companies
Overall Top Issues
Submitted Via
Main Dashboard
Year on Year Growth Rate
Rate of Complaints
Year on Year Dashboard

Visualization 2:

A visual representation of YEAR-ON-YEAR growth in number of Complaints over a period (A DASHBOARD)

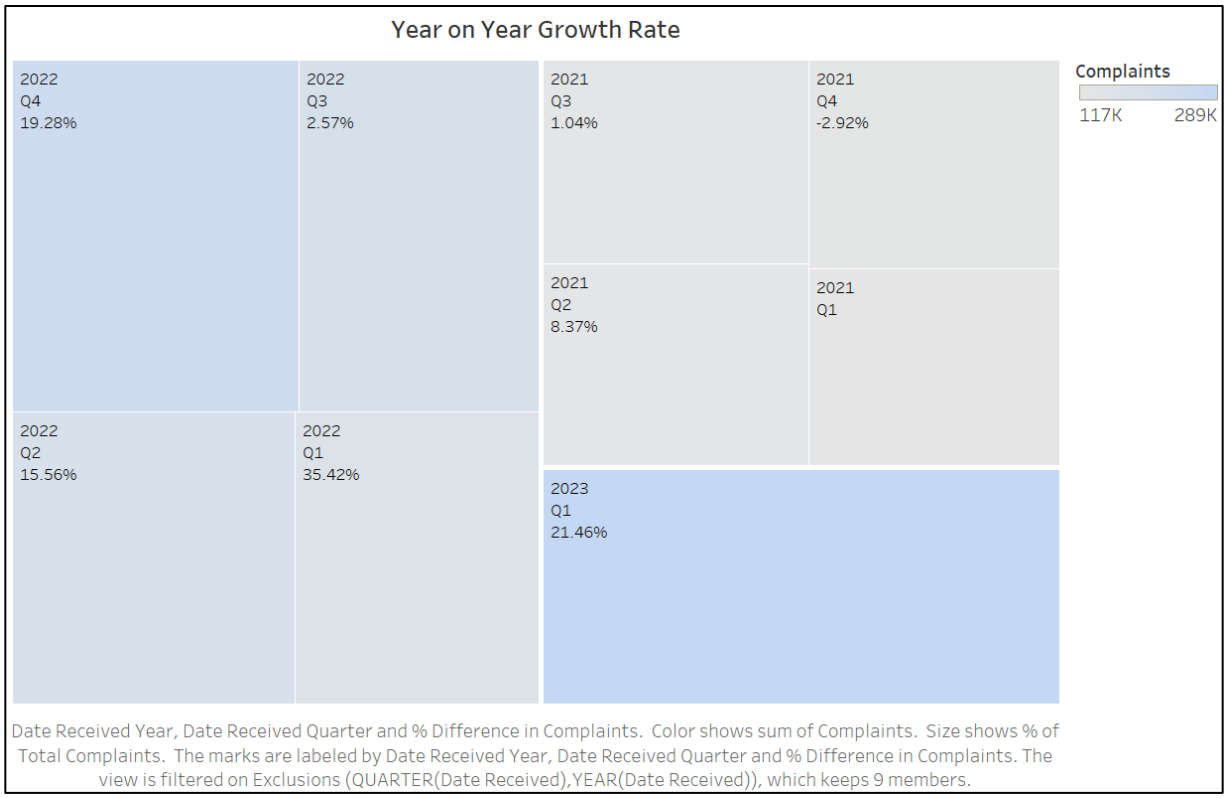


Here are in details steps to get the above DASHBOARD visualizations:

A visual representation of the growth rate of complaints per quarter from year 2021 to 2023

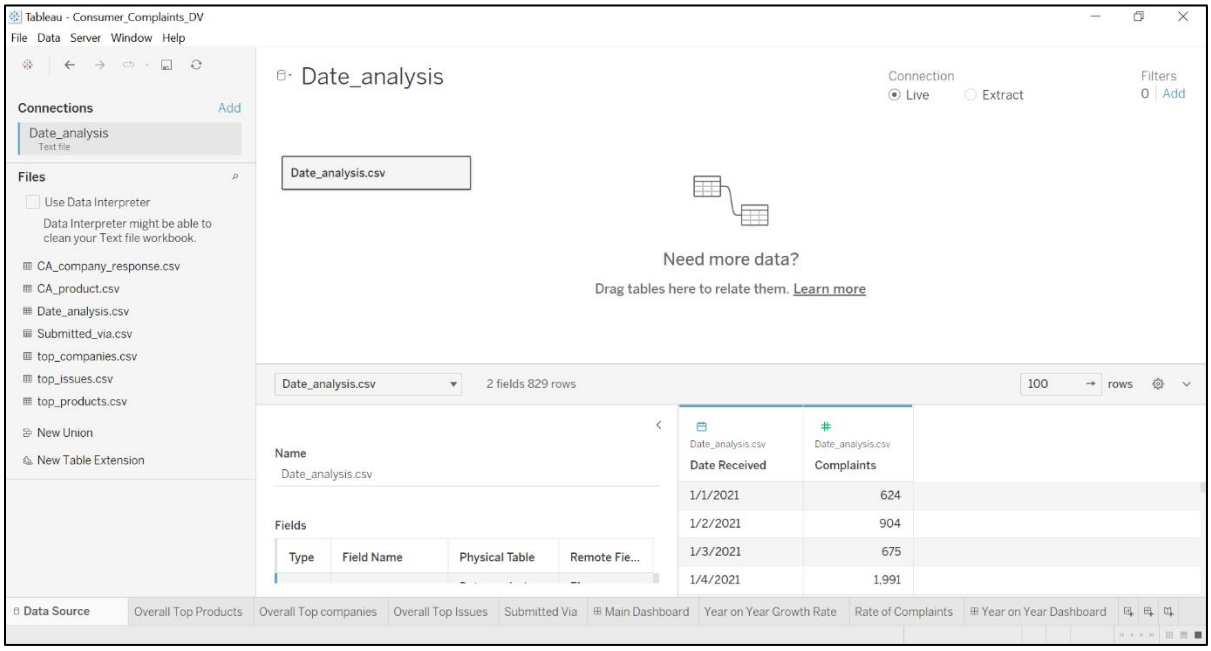
Below is the dataset csv file content:

1/1/2021	624	
1/2/2021	904	
1/3/2021	675	
1/4/2021	1991	
1/5/2021	2310	
1/6/2021	2002	
1/7/2021	1570	
1/8/2021	1659	
1/9/2021	720	
1/10/2021	510	
1/11/2021	1602	
1/12/2021	1786	
1/13/2021	1752	
1/14/2021	1552	
1/15/2021	1649	
1/16/2021	830	
1/17/2021	664	
1/18/2021	1120	
1/19/2021	1696	
1/20/2021	1598	
1/21/2021	2029	
1/22/2021	1842	
1/23/2021	914	
1/24/2021	640	
1/25/2021	1632	
1/26/2021	1808	
1/27/2021	1837	
1/28/2021	1775	

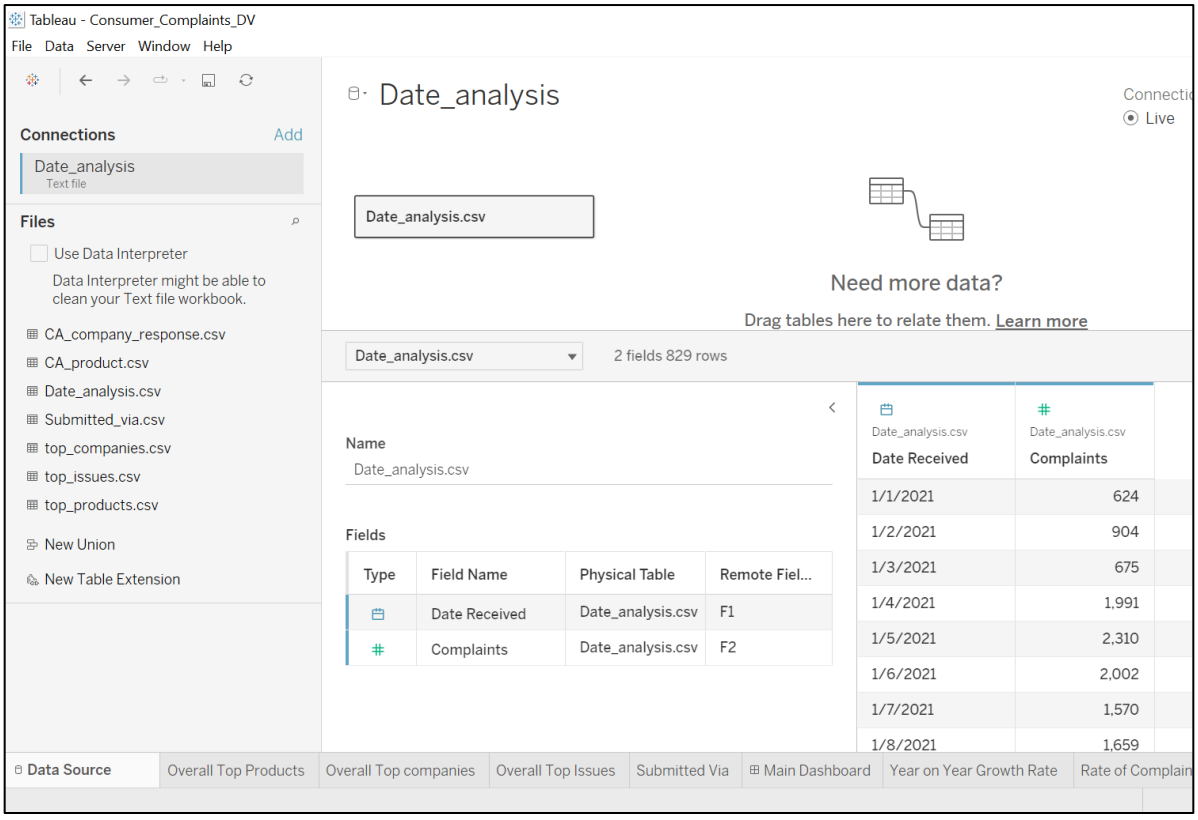


Here are the steps you can take to create a visual representation of the growth rate of complaints per quarter from year 2021 to 2023

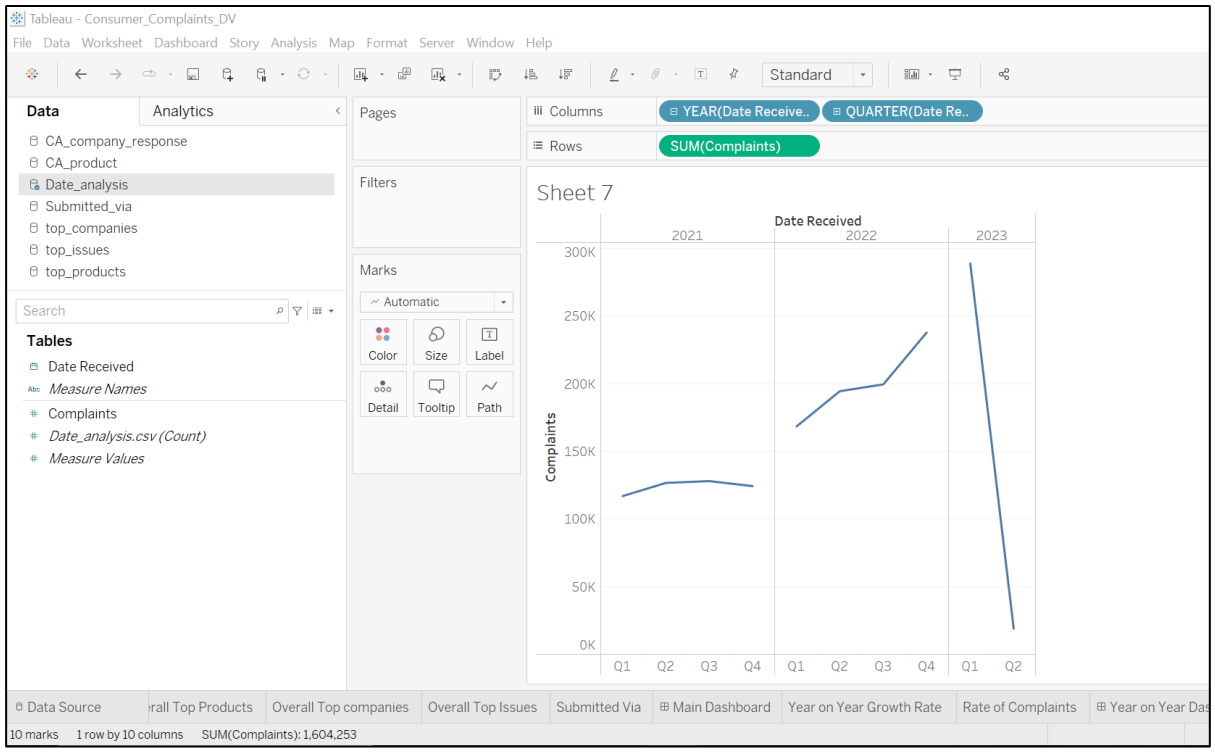
i) Connect to your data source: Open Tableau and connect to **Date_analysis.csv**



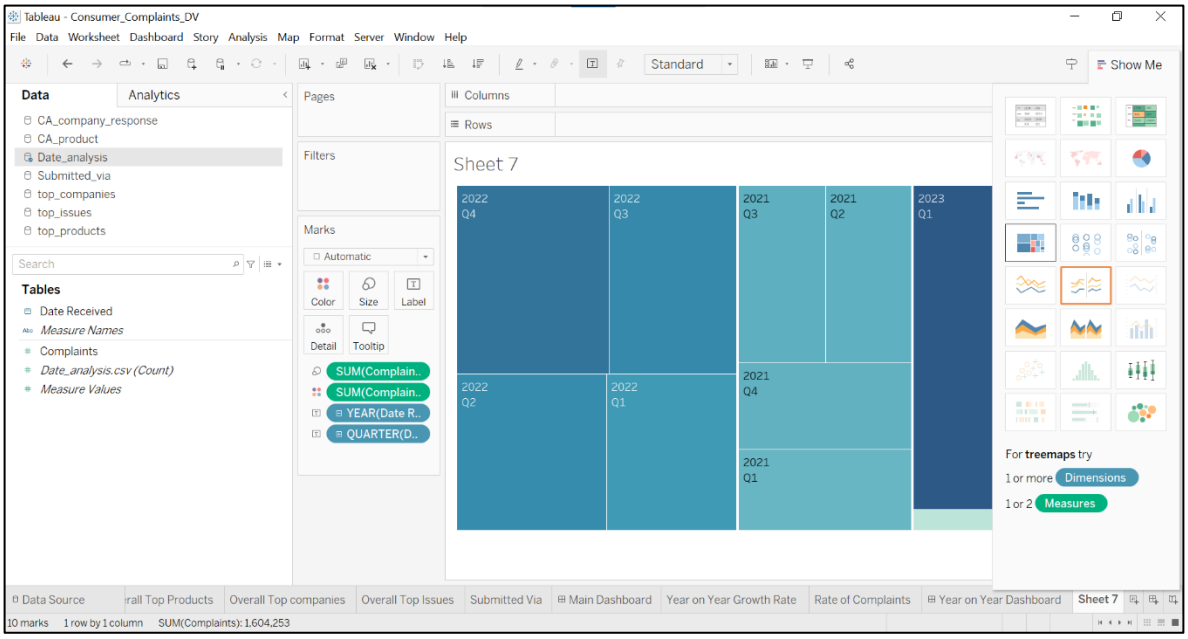
ii) Rename F1, F2 by right clicking each value as F1: Date Received, F2: Complaints



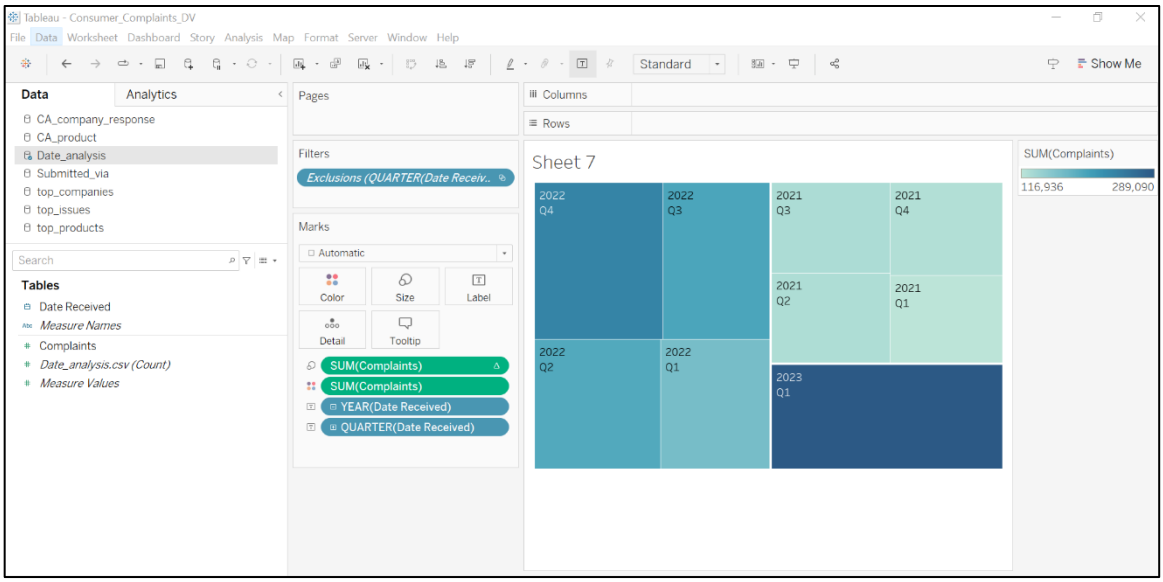
iii) Then you will need to drag **Date Received** to Rows and **Complaints** to columns. Then you have to expand YEAR(Date Received) to one level down at Quarters level by clicking little + sign on YEAR(Date Received)



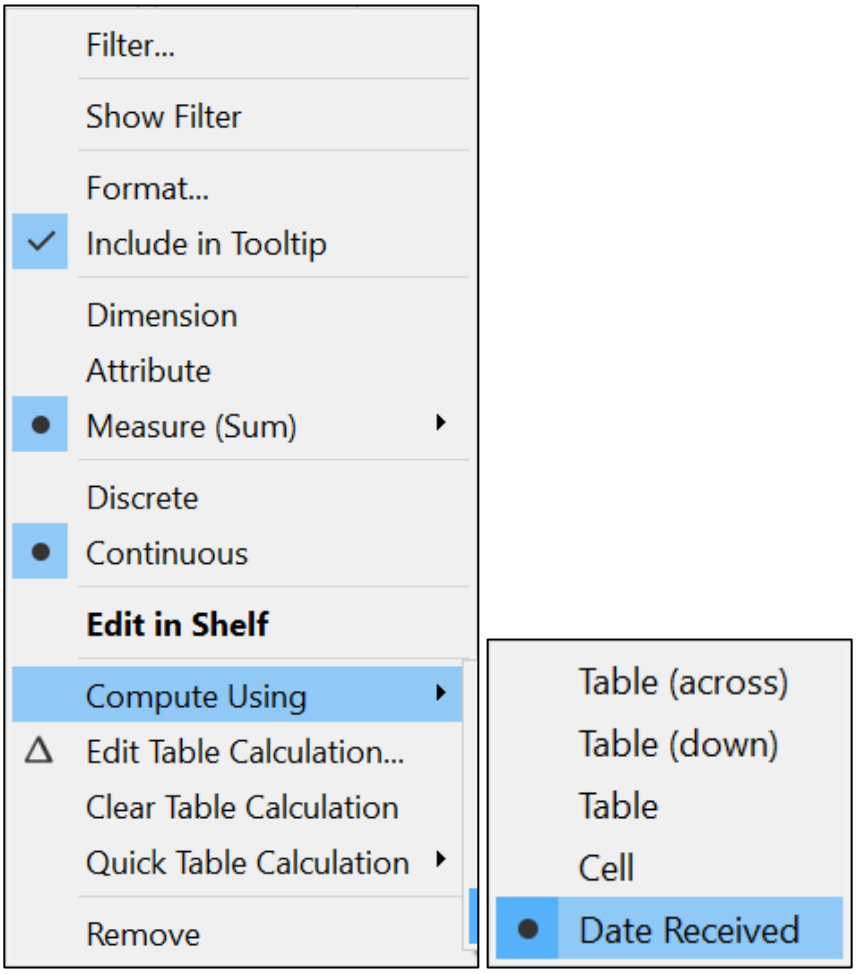
iv) Change the chart type to **Treemaps** from **Show Me** toolbar to the right as shown below

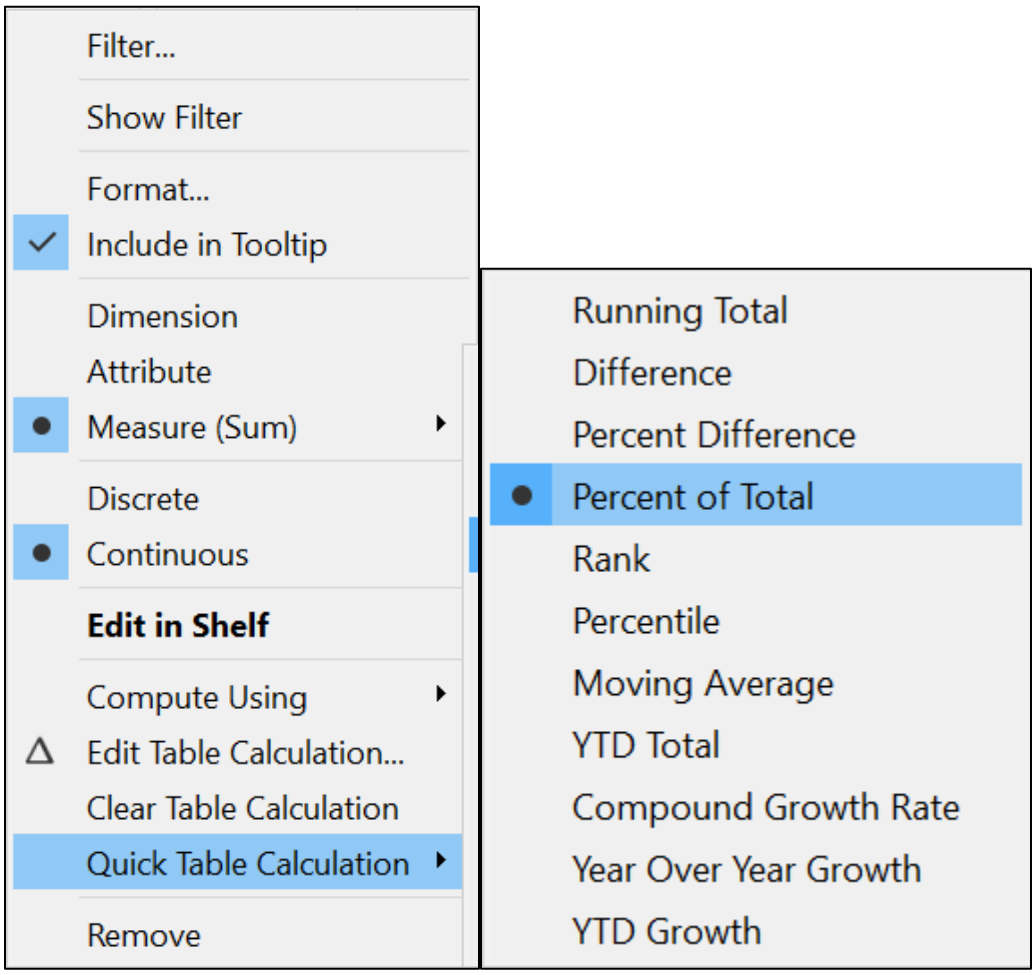


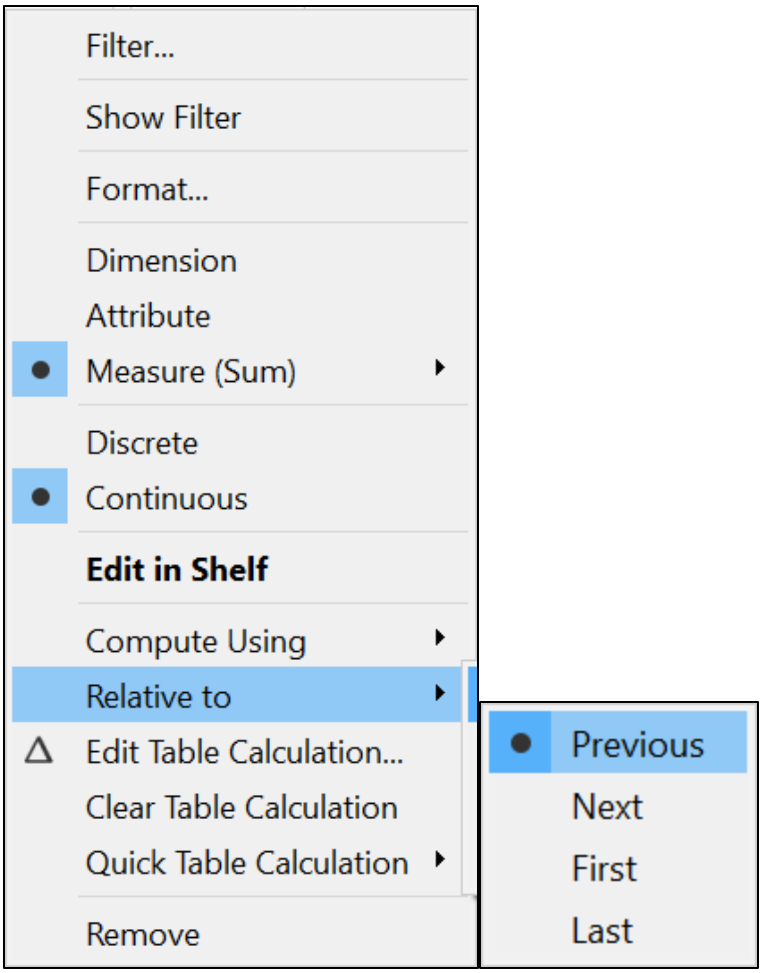
v) Exclude Q2 of 2023 by right clicking on the quarter 2 of 2023 > Exclude



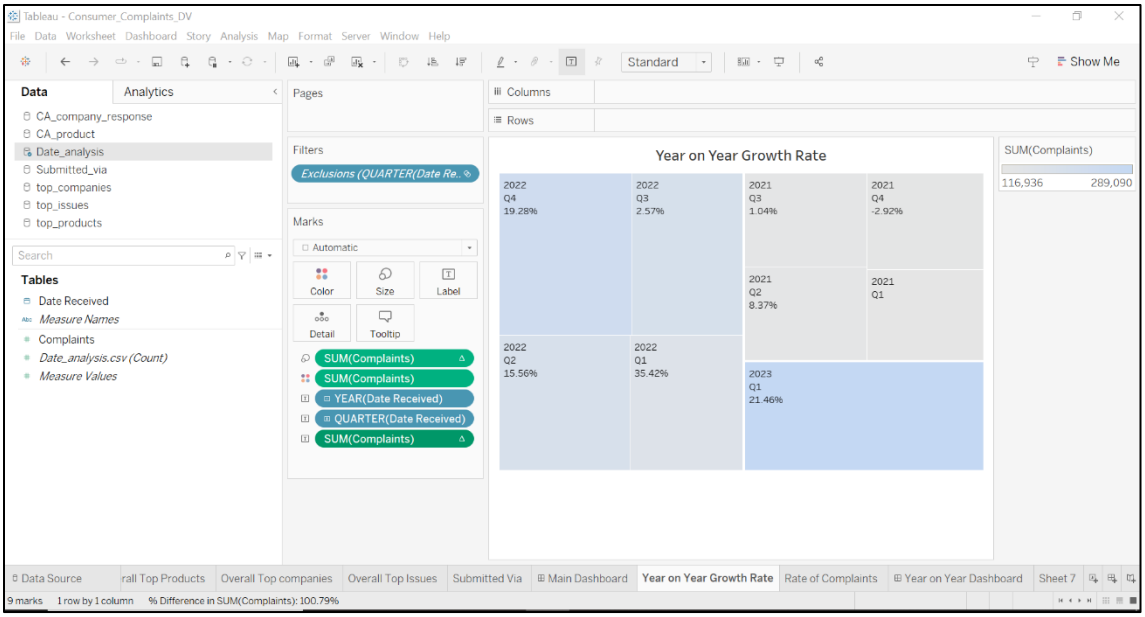
vi) **To show the % difference in complaints compared to previous quarter**, drag the Complaints to Marks > Label. Click on SUM(Complaint) and select options as shown below one by one.




Filter...
Show Filter
Format...
Include in Tooltip
Dimension
Attribute
Measure (Sum)
Discrete
Continuous
Edit in Shelf
Compute Using
Edit Table Calculation...
Clear Table Calculation
Quick Table Calculation
Remove
Running Total
Difference
Percent Difference
Percent of Total
Rank
Percentile
Moving Average
YTD Total
Compound Growth Rate
Year Over Year Growth
YTD Growth



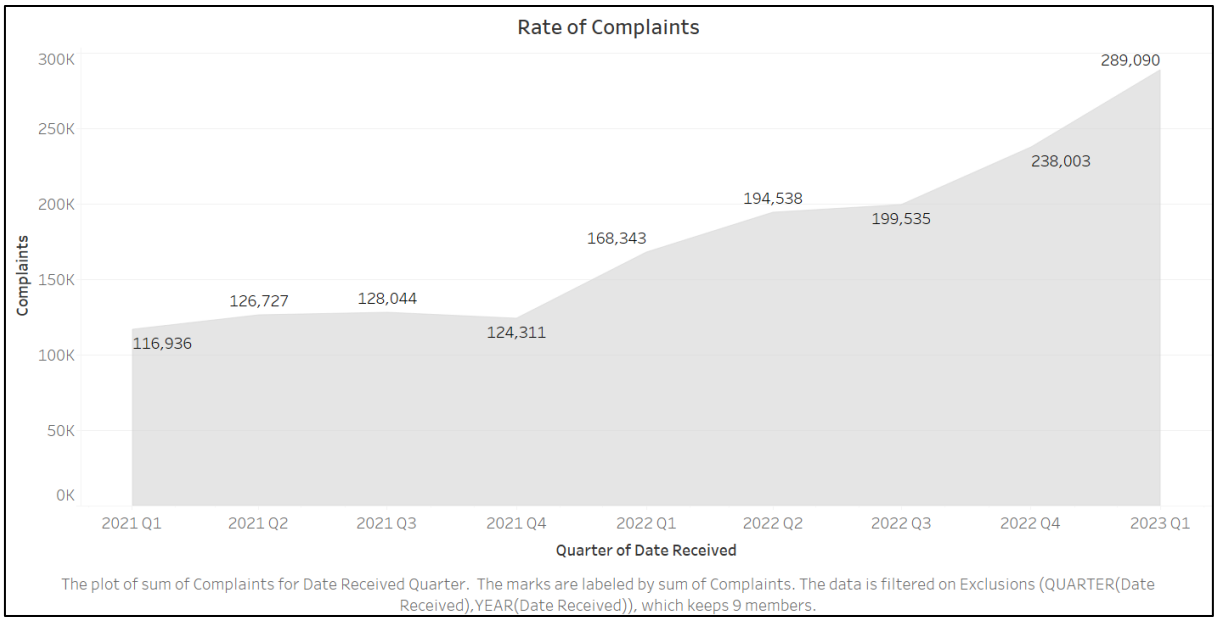
vii) Change the color by dragging **SUM(Complaints)** to Marks > Color. Once you are satisfied with your visualization, save it as a Tableau workbook with. Twbx extension.



A visual representation of the growth rate of complaints per quarter from year 2021 to 2023 – an area chart with complaints statistics

Below is the dataset csv file content (same data source as above visual):

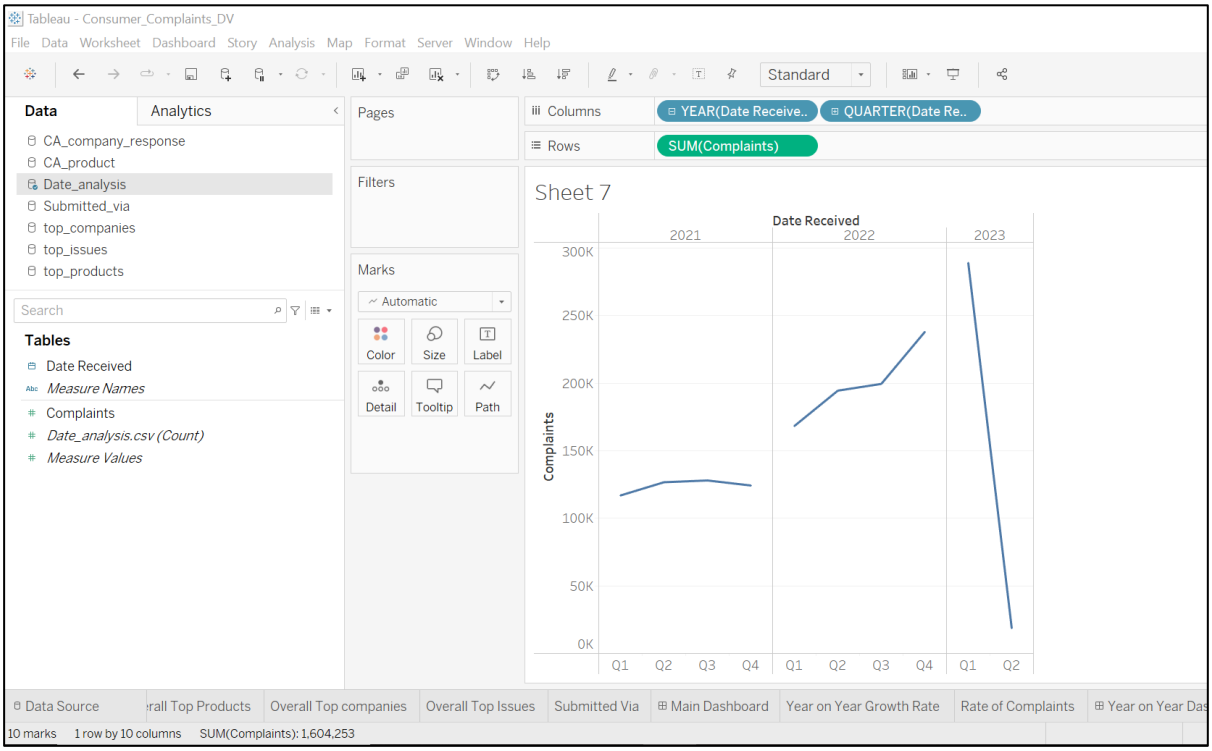
1/1/2021	624
1/2/2021	904
1/3/2021	675
1/4/2021	1991
1/5/2021	2310
1/6/2021	2002
1/7/2021	1570
1/8/2021	1659
1/9/2021	720
1/10/2021	510
1/11/2021	1602
1/12/2021	1786
1/13/2021	1752
1/14/2021	1552
1/15/2021	1649
1/16/2021	830
1/17/2021	664
1/18/2021	1120
1/19/2021	1696
1/20/2021	1598
1/21/2021	2029
1/22/2021	1842
1/23/2021	914
1/24/2021	640
1/25/2021	1632
1/26/2021	1808
1/27/2021	1837
1/28/2021	1775



Here are the steps you can take to create a visual representation of the complaints per quarter from year 2021 to 2023

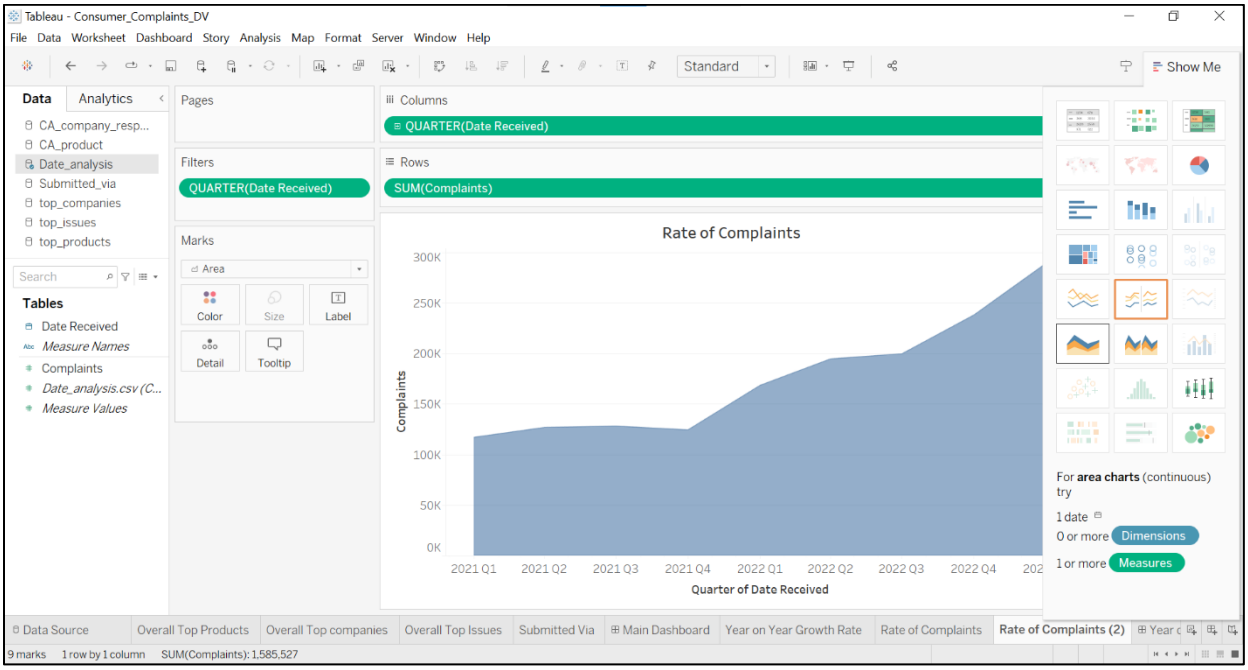
i) Connect to your data source: Open Tableau and connect to **Date_analysis.csv**

ii) Then you will need to drag **Date Received** to Rows and **Complaints** to columns. Then you have to expand YEAR(Date Received) to one level down at Quarters level by clicking little + sign on YEAR(Date Received)

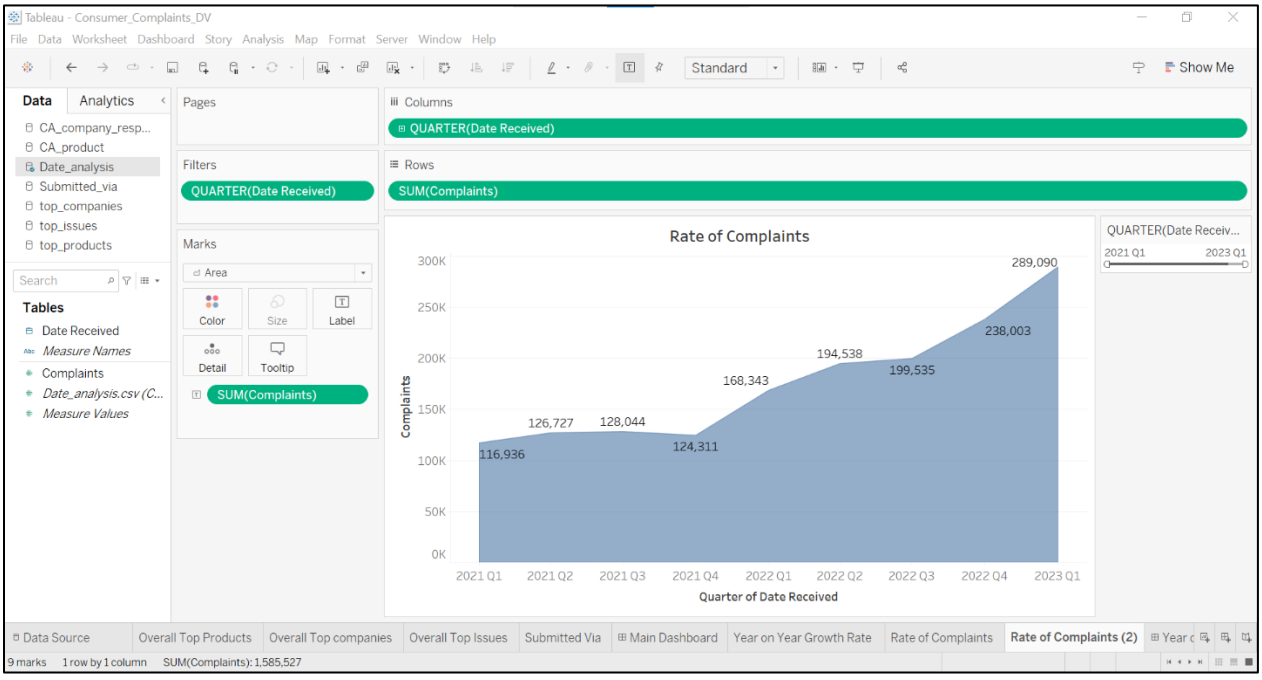


iii) Exclude Q2 of 2023 by right clicking on the quarter 2 of 2023 > Exclude

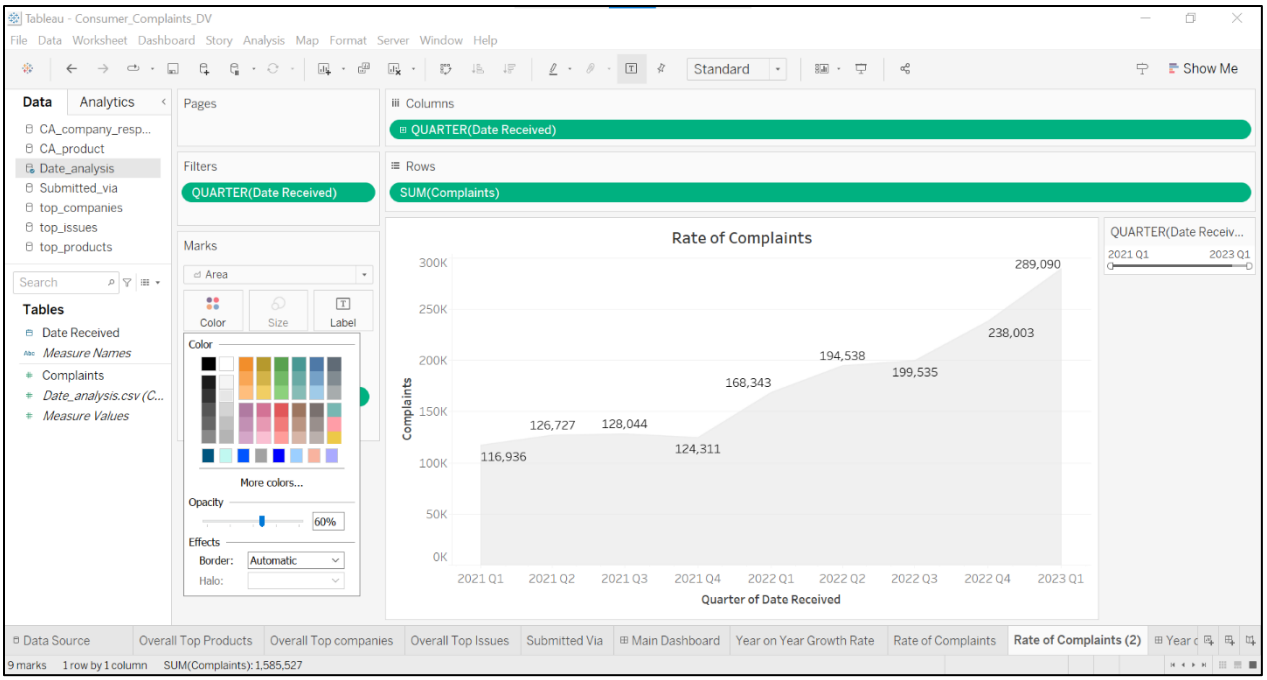
iv) Change the chart type to **Area Charts** from **Show Me** toolbar to the right as shown below



v) Drag the #complaints to Marks -> Label to show the labels on the chart

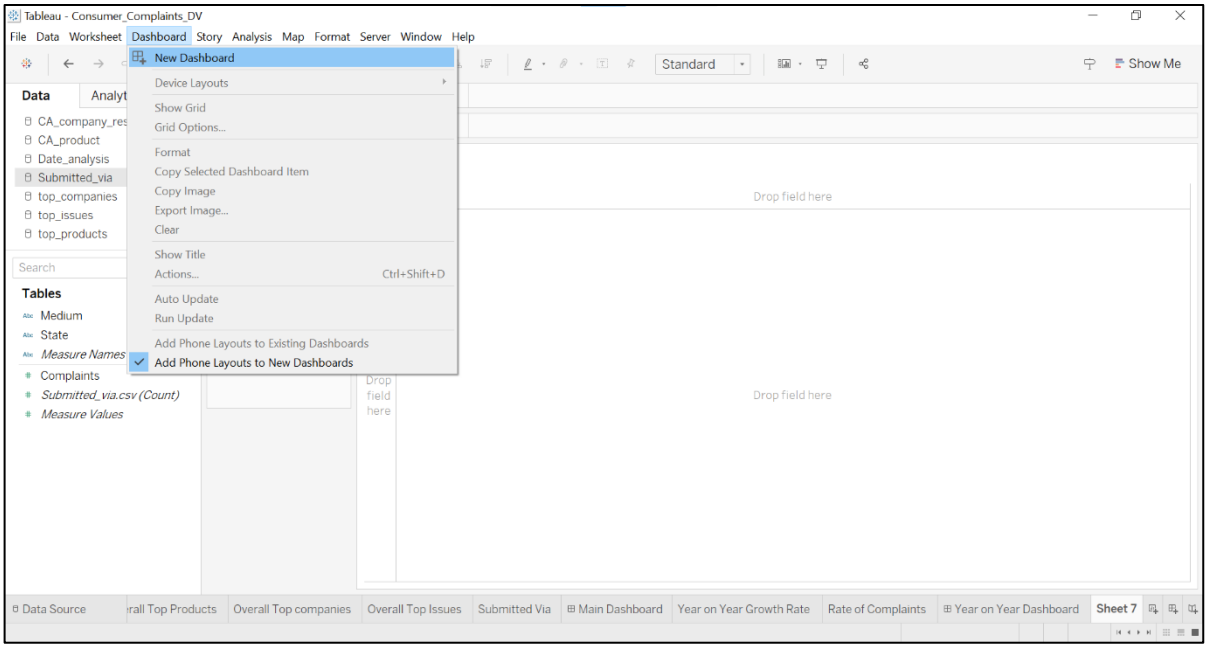


vi) Change the color by clicking on to Marks > Color. Once you are satisfied with your visualization, save it as a Tableau workbook with. Twbx extension

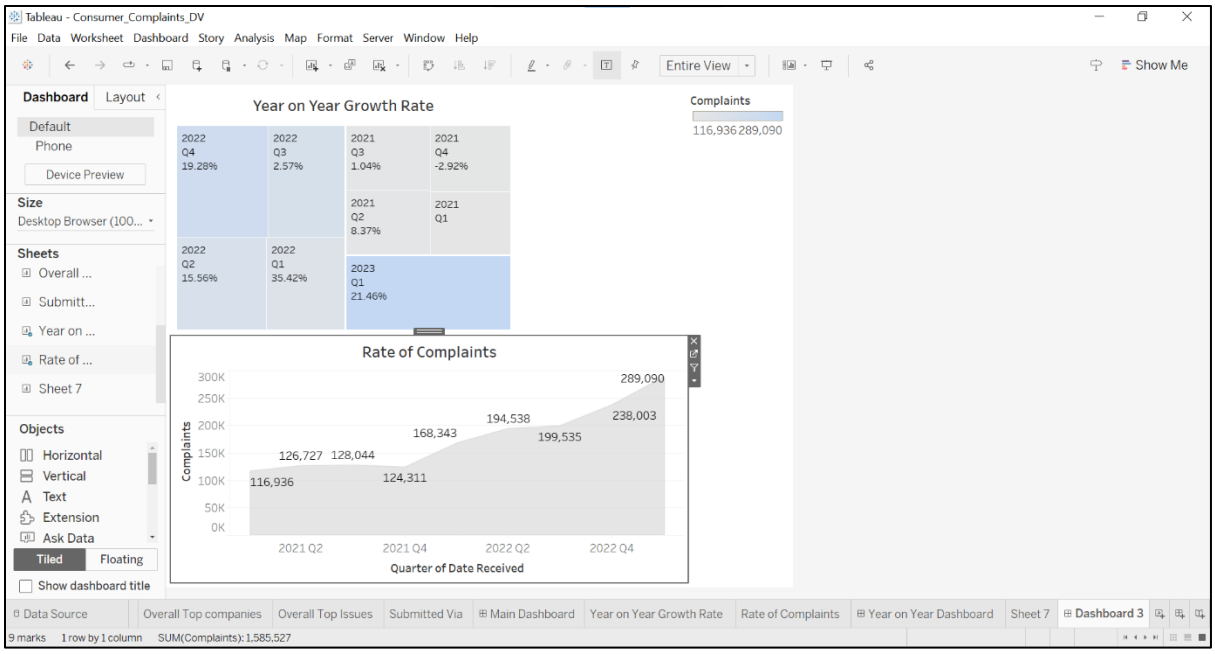


Now, we will put together all 2 visualizations shown above into a DASHBOARD

v) In Tableau, create a new Dashboard as shown below



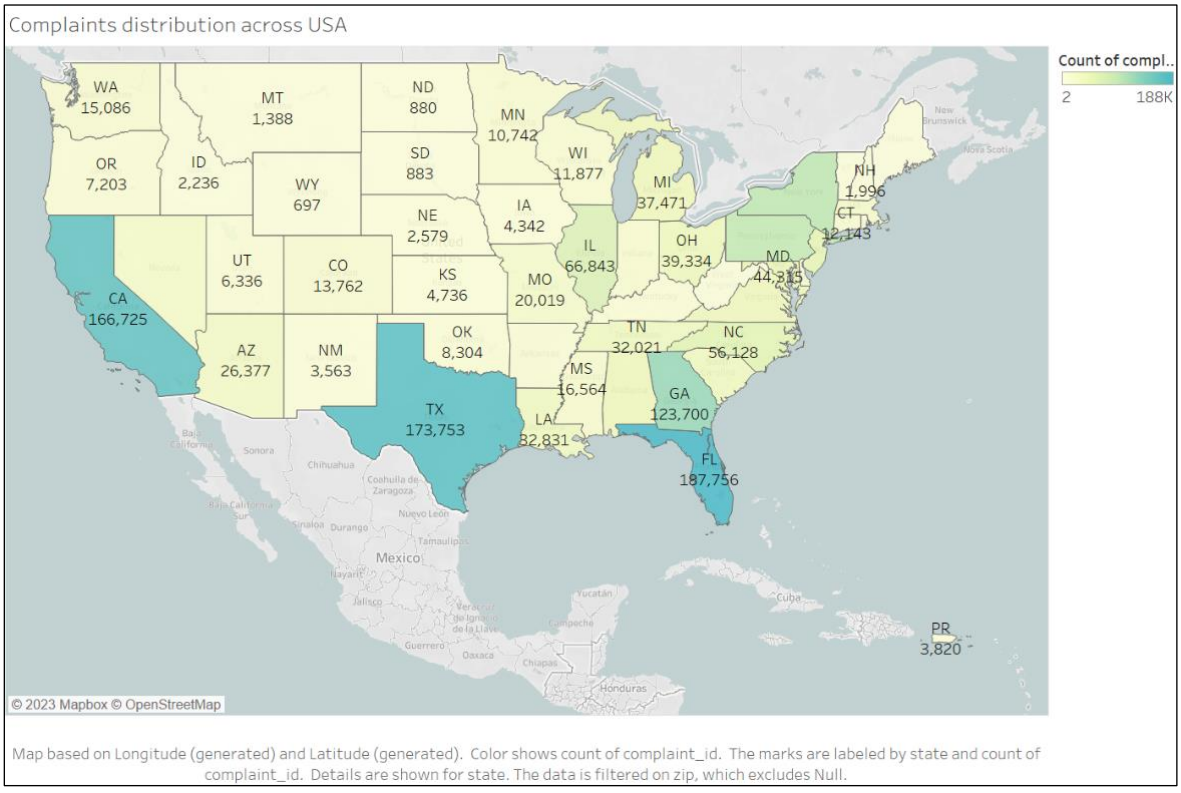
vi) From the left side **Sheets tool bar**, drag all 4 sheets one by one as shown below



vii) Change the positions, format the views and save the tableau file with .twbx extension with the desired Dashboard once done

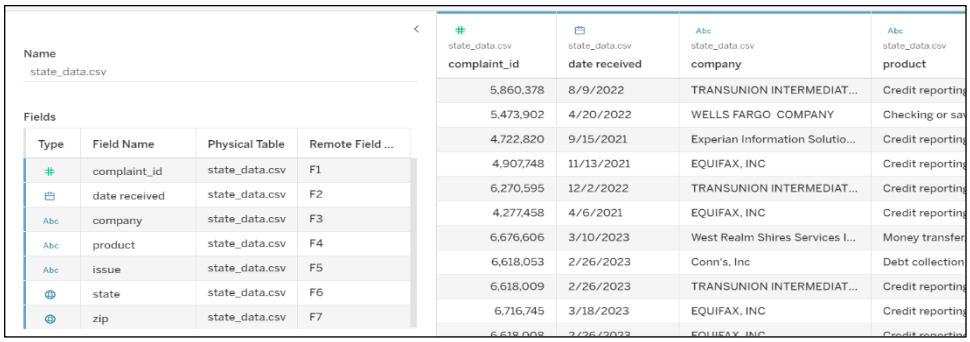
Visualization 3

A visual representation of the complaint's distribution by state



Here are the steps you can take to create a visual representation of the complaint's distribution by state:

i) Connect to your data source: Open your Tableau to connect your server. You need to select Text File state-table.csv file and rename the fields by right click and then click rename F1 to complaint_id, F2 to date received, F3 to company, F4 to product, F5 to issue, F6 to state, F7 to zip.



Name
state_data.csv

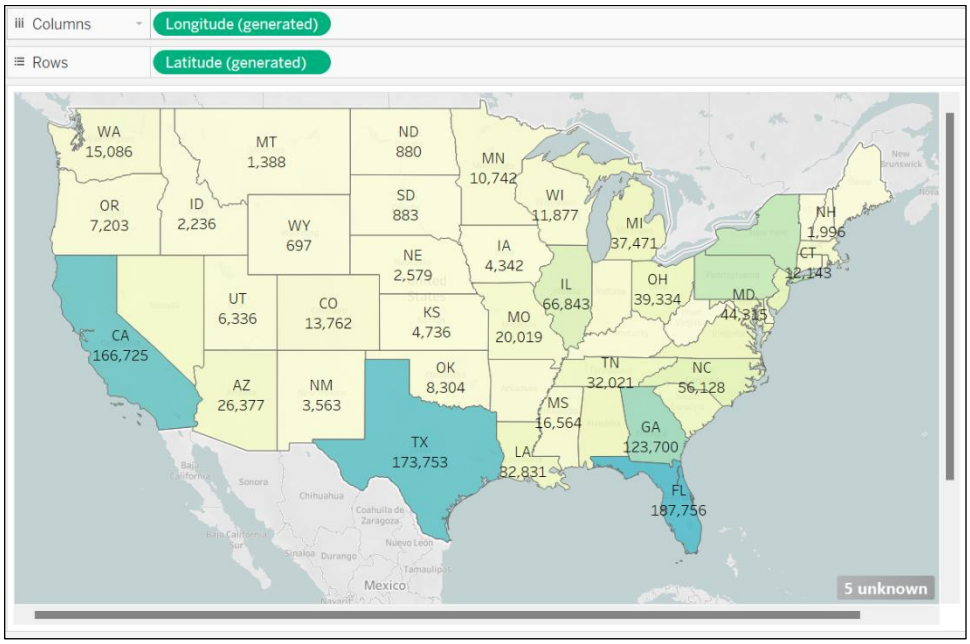
Fields

Type	Field Name	Physical Table	Remote Field ...
#	complaint_id	state_data.csv	F1
📅	date received	state_data.csv	F2
Abc	company	state_data.csv	F3
Abc	product	state_data.csv	F4
Abc	issue	state_data.csv	F5
🌐	state	state_data.csv	F6
🌐	zip	state_data.csv	F7

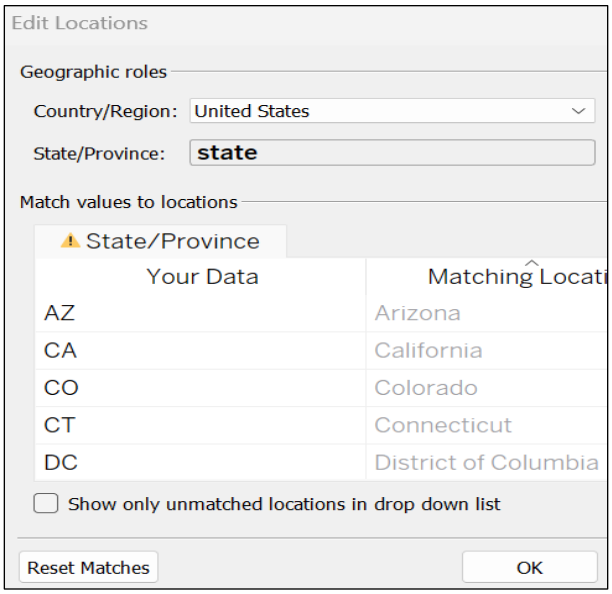
state_data.csv complaint_id	state_data.csv date received	state_data.csv company	state_data.csv product
5,860,378	8/9/2022	TRANSUNION INTERMEDIAT...	Credit reporting
5,473,902	4/20/2022	WELLS FARGO COMPANY	Checking or sav
4,722,820	9/15/2021	Experian Information Solutio...	Credit reporting
4,907,748	11/13/2021	EQUIFAX, INC	Credit reporting
6,270,595	12/2/2022	TRANSUNION INTERMEDIAT...	Credit reporting
4,277,458	4/6/2021	EQUIFAX, INC	Credit reporting
6,676,606	3/10/2023	West Realm Shires Services I...	Money transfer
6,618,053	2/26/2023	Conn's, Inc	Debt collection
6,618,009	2/26/2023	TRANSUNION INTERMEDIAT...	Credit reporting
6,716,745	3/18/2023	EQUIFAX, INC	Credit reporting

ii) Create a new worksheet: Once you have connected to your data source, create a new worksheet by selecting "sheet1" next to Data Source, which will present the following frame. Later rename the “sheet1” to statemap.

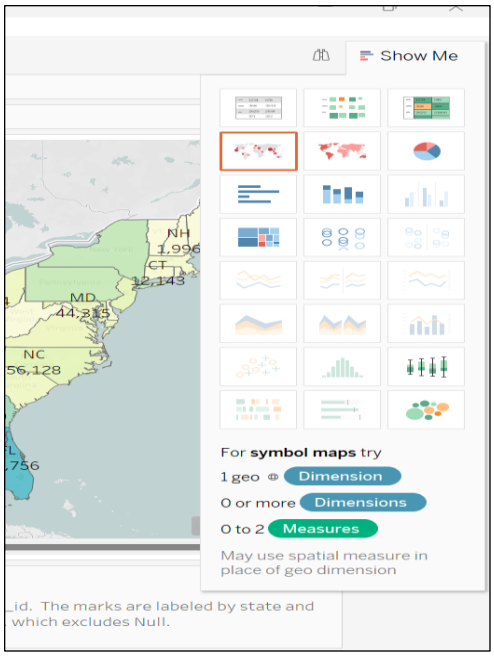
- Choose the relevant columns: Drag longitude to Columns and Latitude to Rows to create map chart.



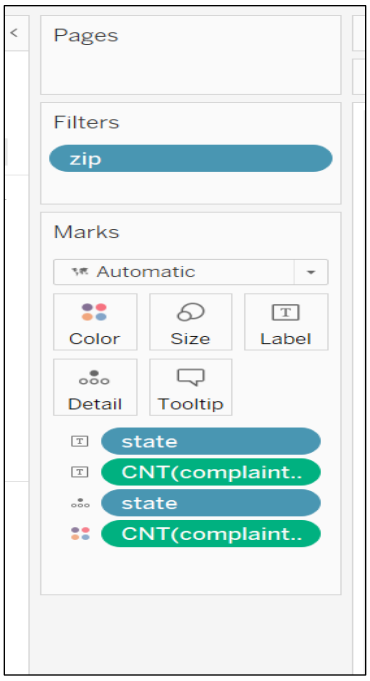
Then go to Maps in the menu bar and click on Edit Locations and make sure country/Region is selected as United States.



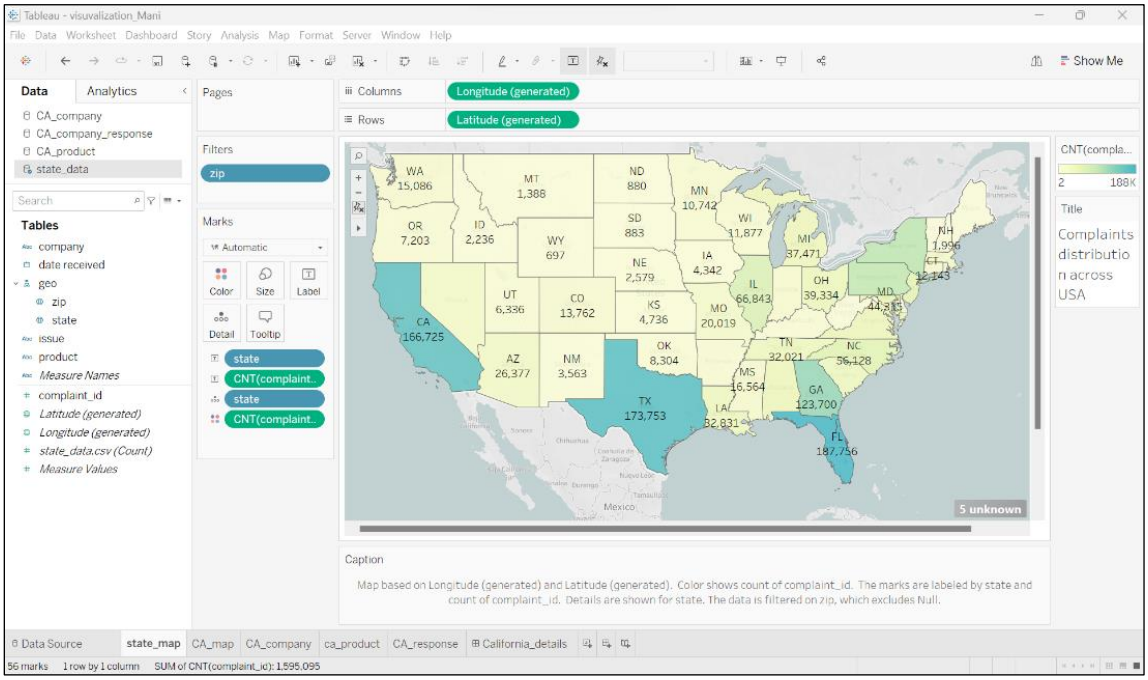
- Choose the chart type: You can select Show Me and its Geo Map to display



- For the changing the color of the map, you need to drag the CNT(Compalint_id) to color. Drag zip and put it into the filters. To display the state name and number of count, drag the state column and put it into the to Label and drag CNT(complaint_id) to Label as well.

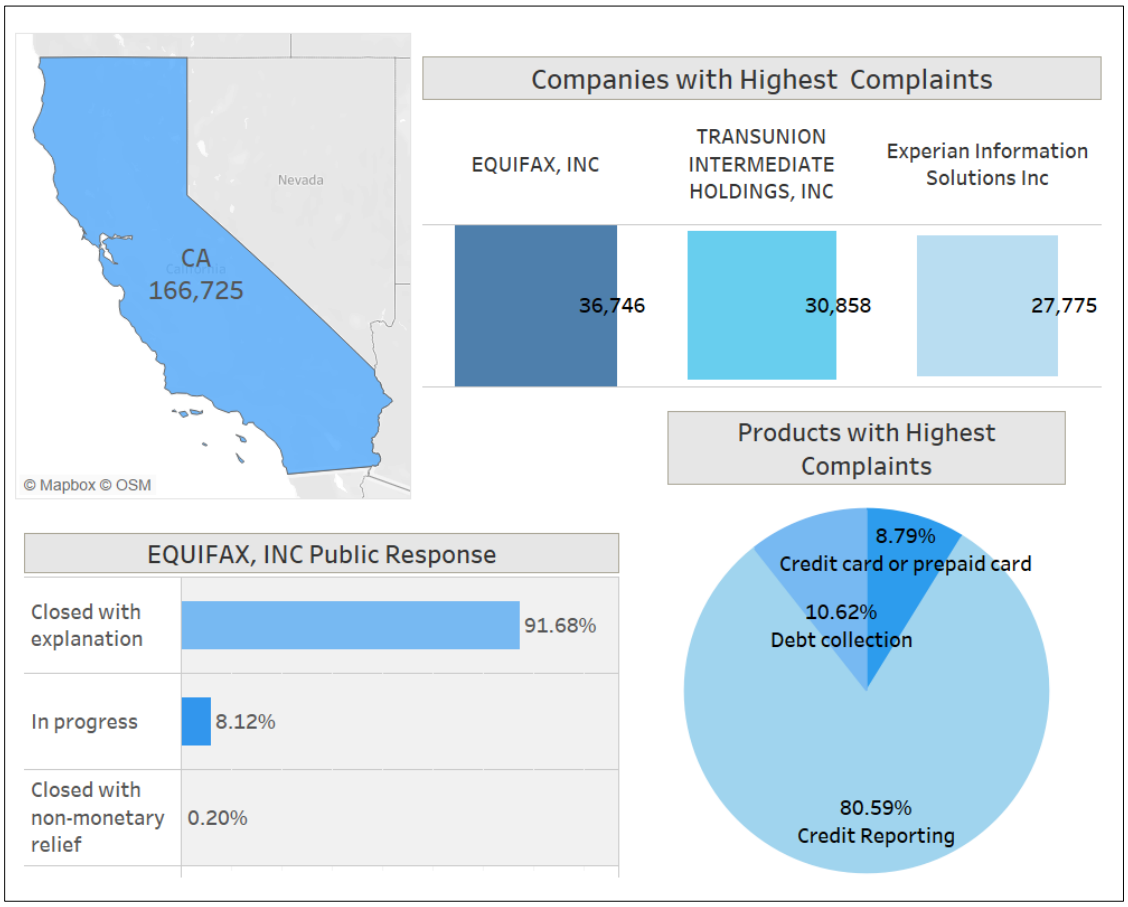


vii) Once you are satisfied with your visualization, save it as a Tableau workbook with .Twbx extension.



Visualization 4- Visualization of the state CA

A visual representation of Overall Complaints Status based on CALIFORNIA STATE by Number of Complaints received per Company, per Product, Per Issue and how the complaints are submitted (A DASHBOARD)

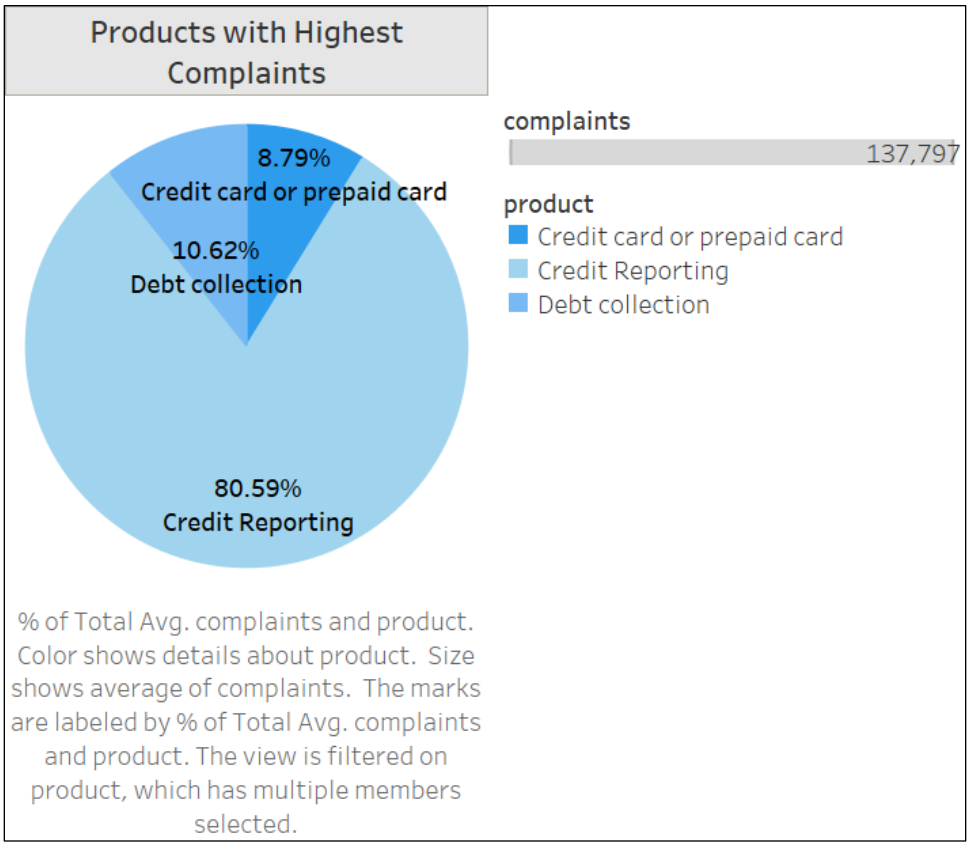


Here are the steps you can take to create a visual representation of the above Dashboard.

Top products based on complaints in California state:

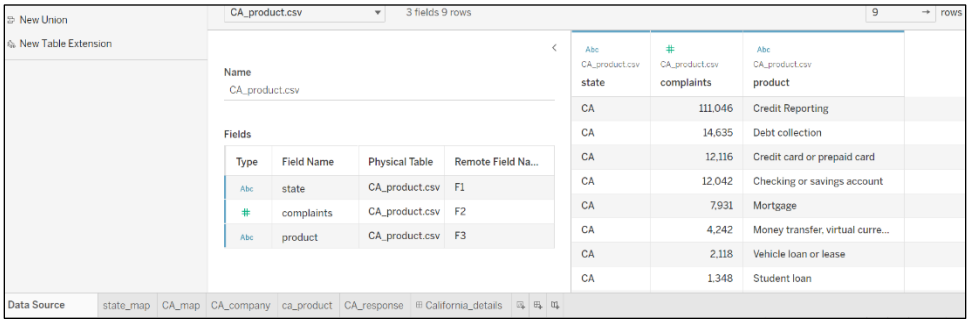
Below is the file in tsv(tab delimited format)

CA	111046	Credit reporting, credit repair services, or other personal consumer reports
CA	14635	Debt collection
CA	12116	Credit card or prepaid card
CA	12042	Checking or savings account
CA	7931	Mortgage
CA	4242	Money transfer, virtual currency, or money service
CA	2118	Vehicle loan or lease
CA	1348	Student loan
CA	1315	Payday loan, title loan, or personal loan



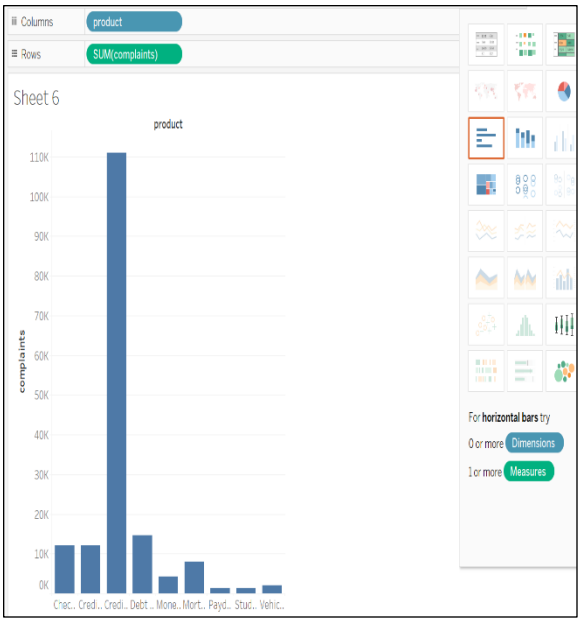
Here are the steps you can take to create a visual representation of the CA_products distribution in California state:

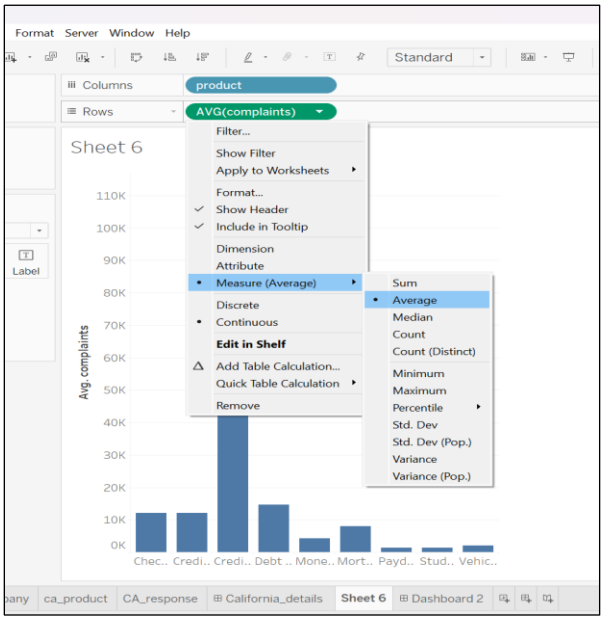
i) Connect to your data source: Open your Tableau to connect your server. You need to select Text File **ca_product.csv** file and rename the fields by right click and then click rename F1 to state, F2 to complaints, F3 to product.



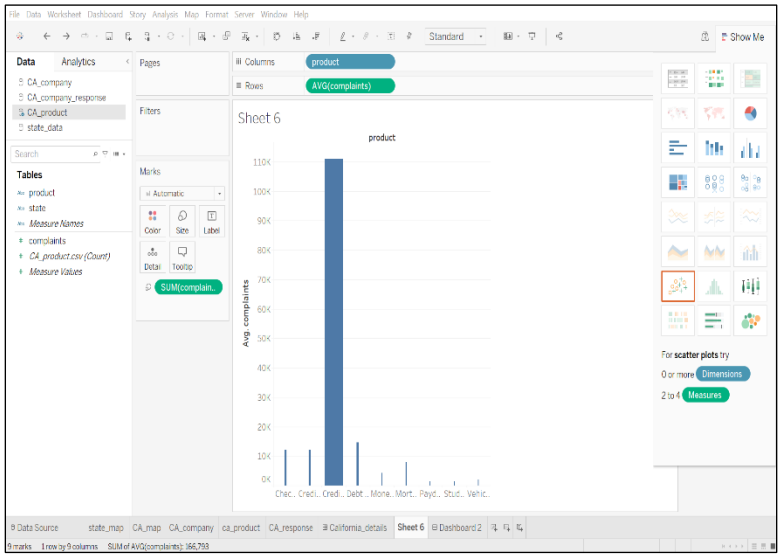
ii) Once you have connected to your data source, create a new worksheet by selecting "sheet2" next to the Data Source. Later rename the sheet2 to **ca_product**.

iii) Drag the **product** to columns and **complaints** and **SUM(complaints) to** Rows shelves to create the chart.

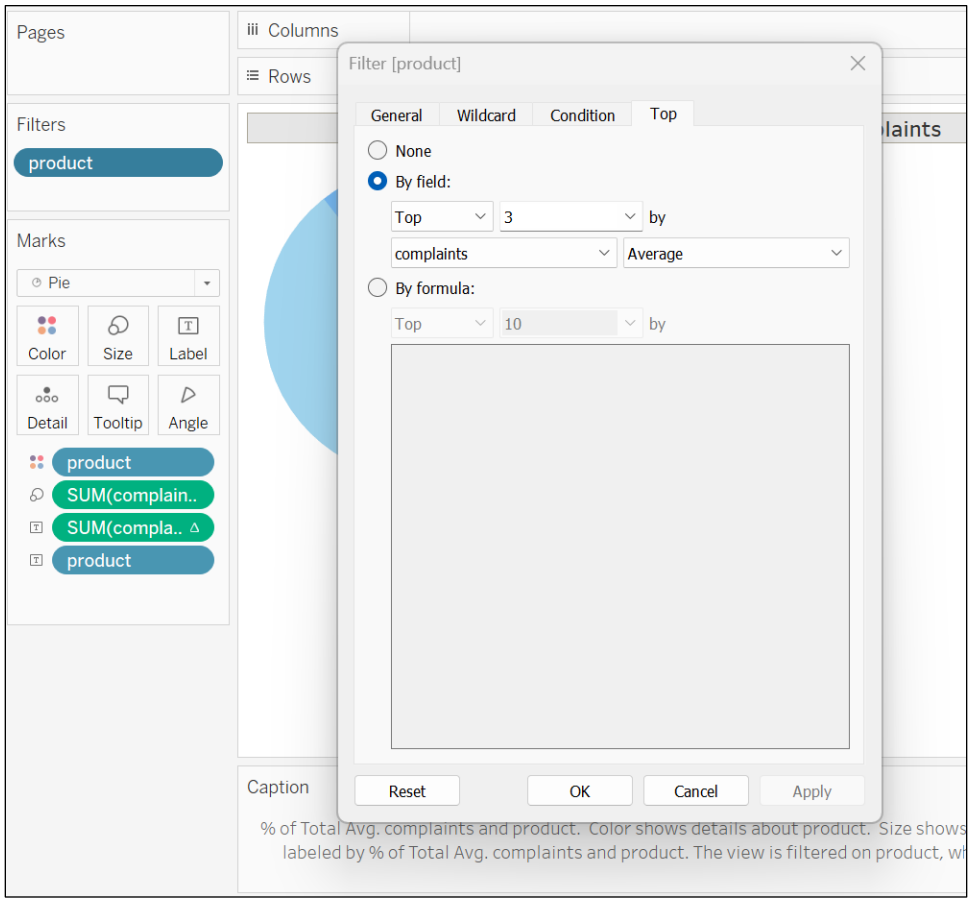




iv) You can select Show Me and change the type of chart to pie chart to display visualization chart.

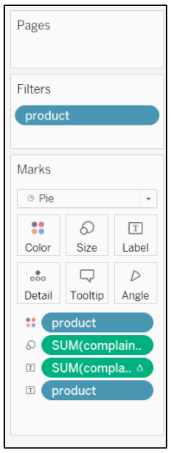


v) After changing the chart type to pie, drag the **Product to Filters** and then right click filters and select the tab top and enable the **BY FIELD** and select top 3 by Complaints and sum.

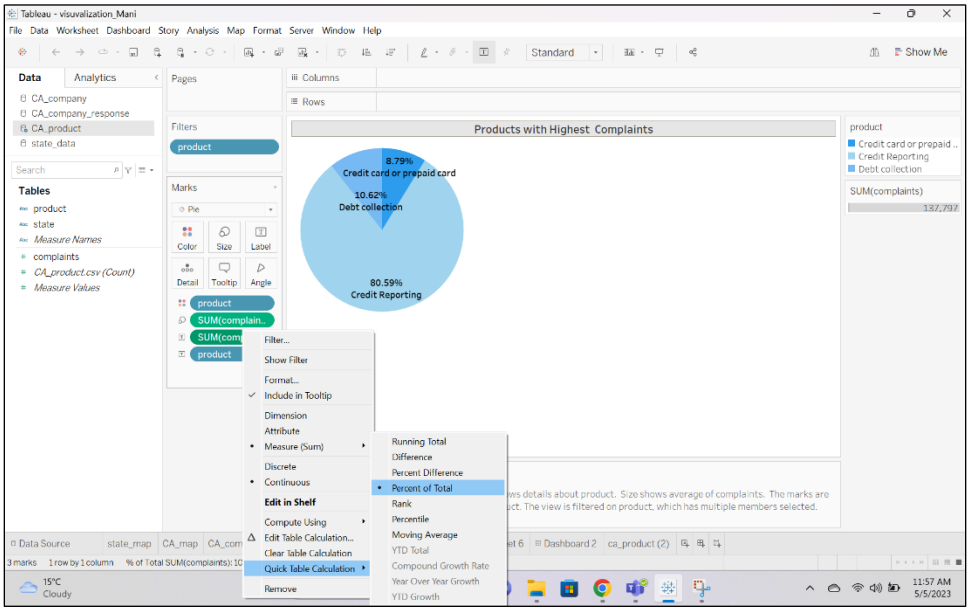


vi) For the changing the color of the pie chart, you need to drag the product to color. Drag SUM(complaint) and put it into the size (to adjust the size of the chart) and Label. To display the product name , drag the product column and put it into the to Label.

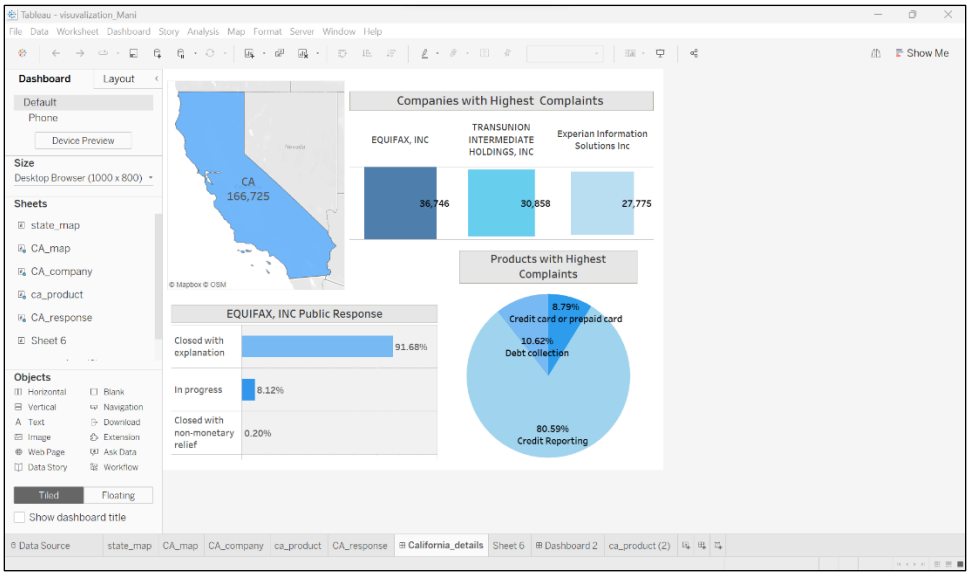
Note: The Size of the pie are automatically adjusted based on SUM(Complaints) once we selected the chart type to pie chart.



vii) After placing the filed SUM(complaint) in details, right click and select quick Table Calculation and then click Percent of Total to get the percentage.



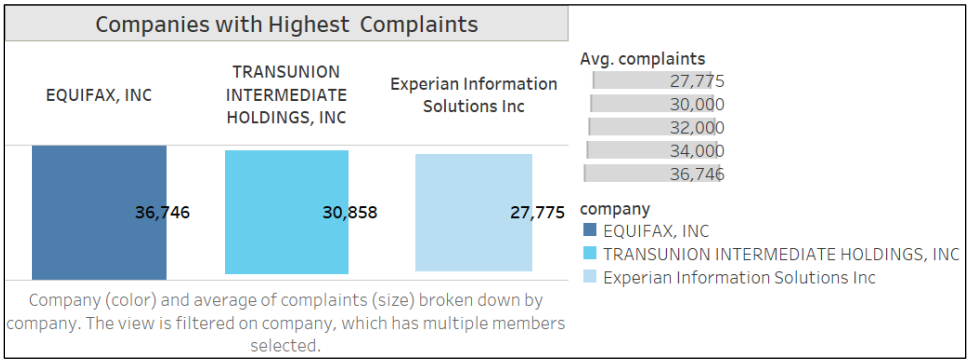
viii) Once you are satisfied with your visualization, save it as a Tableau workbook with. Twbx extension.



Top companies based on complaints

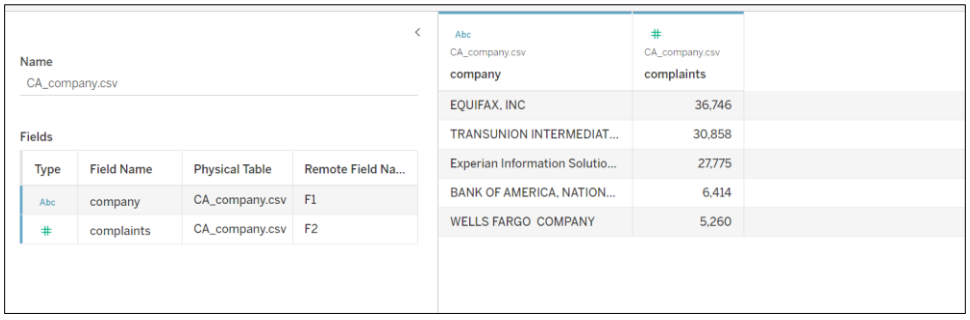
Below is the dataset file in tsv format (tab delimited format):

```
EQUIFAX, INC    36746
TRANSUNION INTERMEDIATE HOLDINGS, INC   30858
Experian Information Solutions Inc  27775
BANK OF AMERICA, NATIONAL ASSOCIATION   6414
WELLS FARGO  COMPANY    5260
```



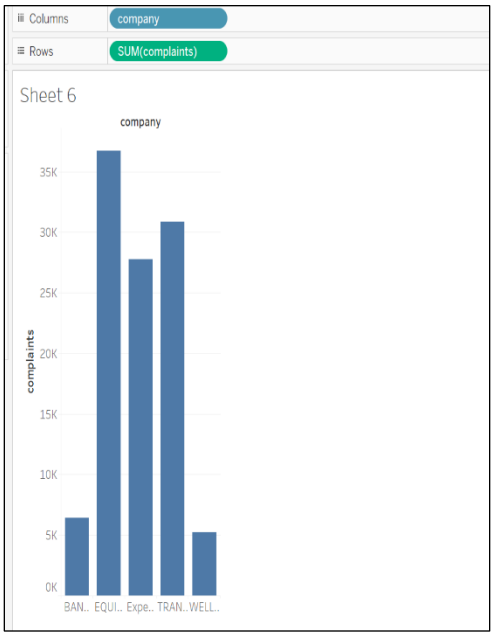
Here are the steps you can take to create a visualization of the top companies based on complaints in the state of CA using Tableau:

i) Open your Tableau to connect your server. You need to select Text File **CA_company.csv** file and rename the fields by right click and then click rename F1 to company, F2 to complaints.

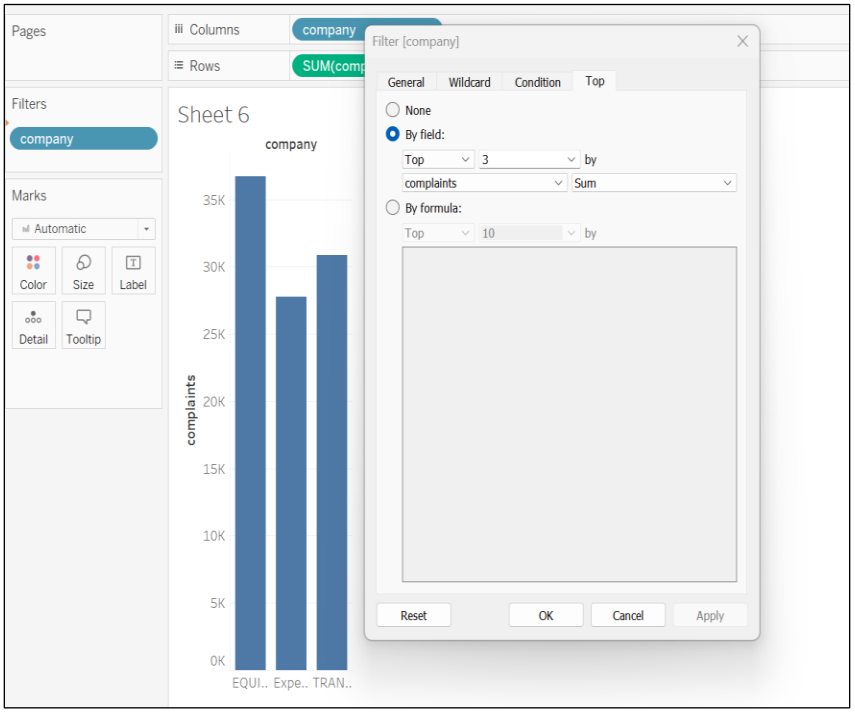


ii) Create a new worksheet: Once you have connected to your data source, create a new worksheet by selecting "sheet3" next to Data Source, which will present the following frame. Later rename the “sheet1” to CA_company.

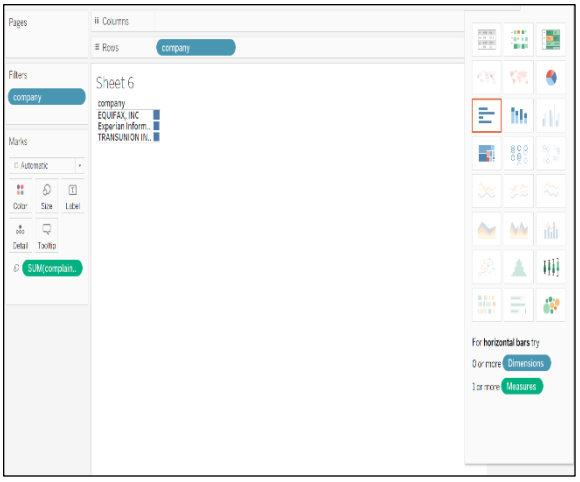
iii) Drag **company** to Columns and SUM**(complaints)** to Rows to create chart.

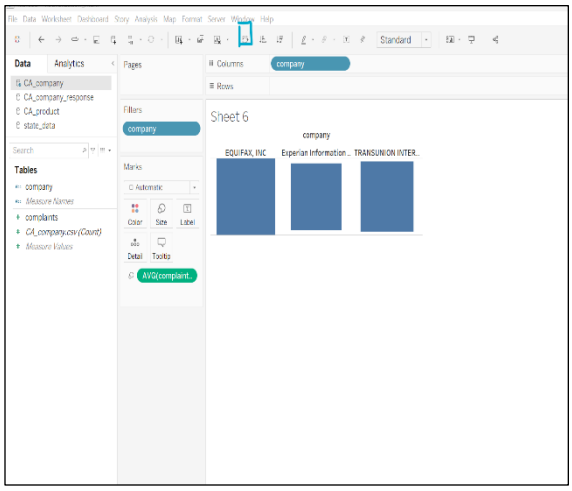


iv) Drag the "**company**" field onto the "Filters" shelf , then right click and then click on Edit filters and select “top 3" from the list of company.



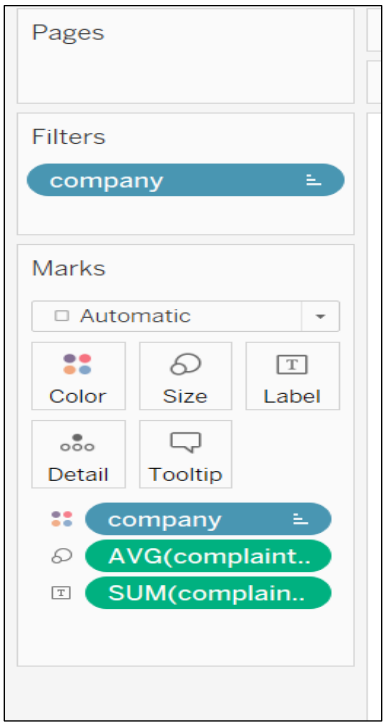
v) You can select Show Me and select the heat map chart to display. After selecting the heat map, click on the swap rows and columns and enlarge the size of the chart.



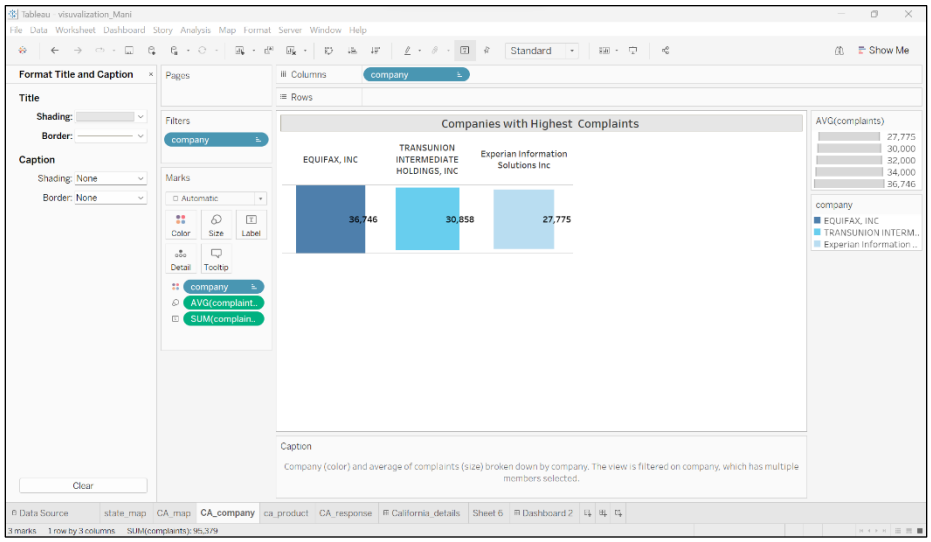


vi) For the changing the color of the heat map chart, you need to drag the company to color. Drag complaints and put it into the size , by default it will be as SUM(complaints), now change the SUM(complaints) to AVG(complaints) by right click and click on measures change SUM to AVG and to display the count, drag the SUM(complaint) to Label.

Note: The Size of the heat map are automatically adjusted based on AVG(Complaints).



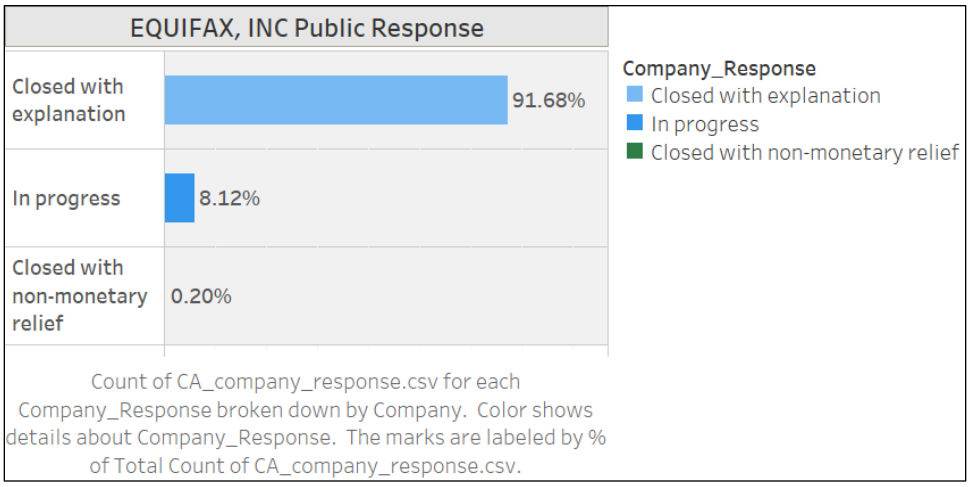
vii) Once you are satisfied with your visualization, save it as a Tableau workbook with .Twbx extension.



A company public response for Equifax based on complaints

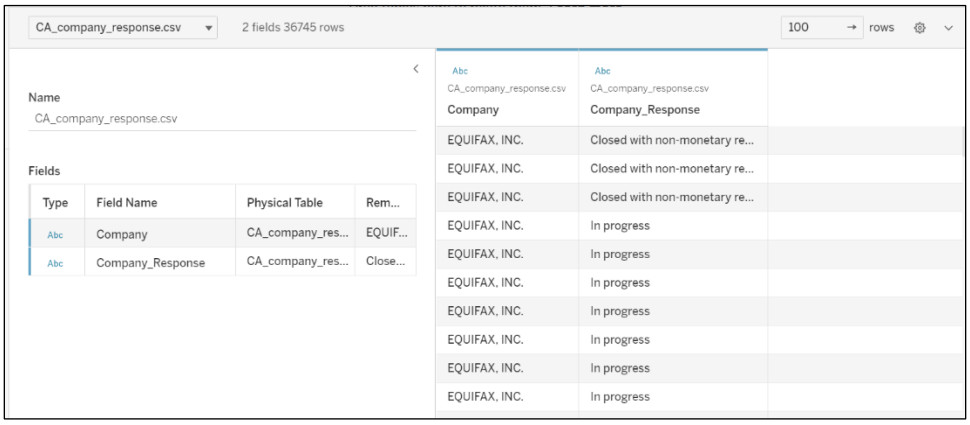
Below is the dataset in csv(comma delimited format)

EQUIFAX,	Closed with non-monetary relief
EQUIFAX,	Closed with non-monetary relief
EQUIFAX,	Closed with non-monetary relief
EQUIFAX,	Closed with non-monetary relief
EQUIFAX,	In progress
EQUIFAX,	In progress
EQUIFAX,	In progress
EQUIFAX,	In progress
EQUIFAX,	In progress
EQUIFAX,	In progress
EQUIFAX,	In progress
EQUIFAX,	In progress
EQUIFAX,	In progress
EQUIFAX,	In progress



Here are the steps you can take to create a visualization of the public response of Equifax based on complaints in the state of CA using Tableau:

i) Open your Tableau to connect your server. You need to select Text File **CA_company_response.csv** file and rename the fields by right click and then click rename F1 to company, F2 to complany_Response.

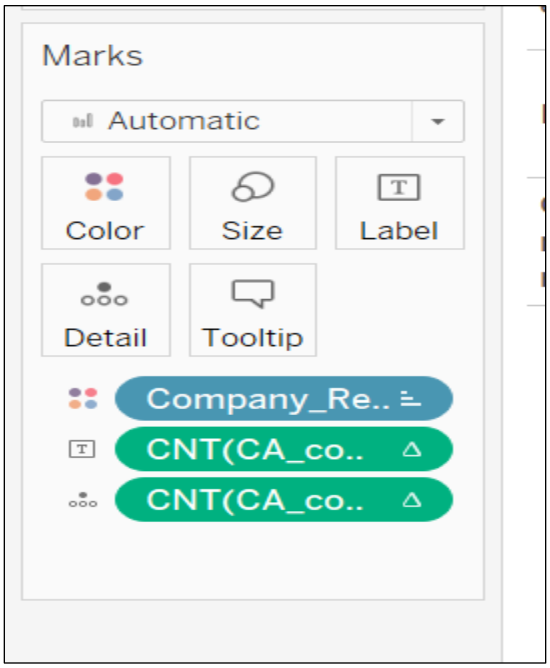


ii) Once you have connected to your data source, create a new worksheet rename the “sheet” to **CA_response**.

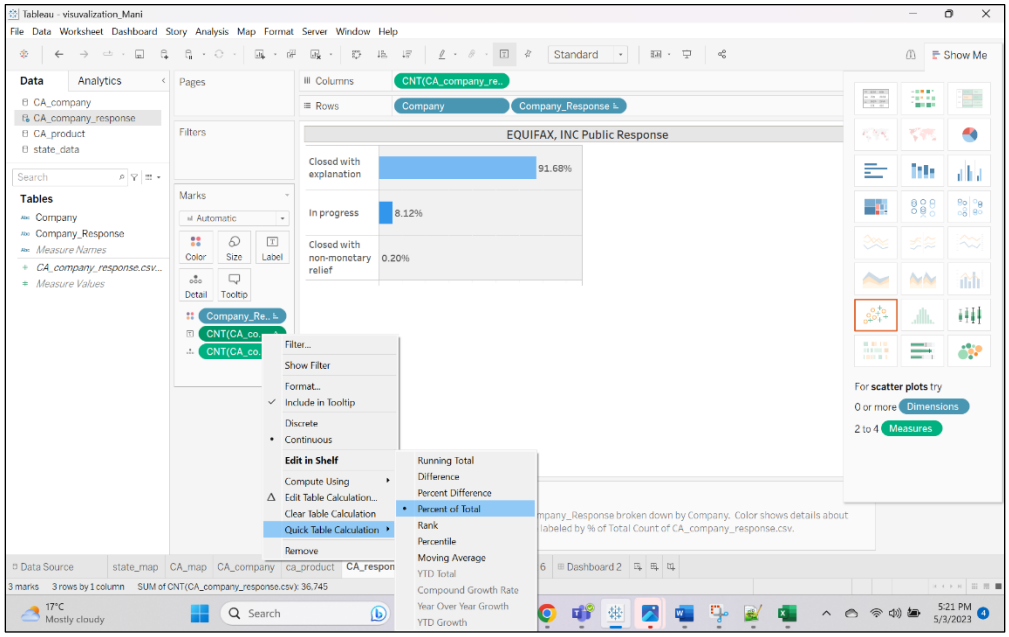
iii) Drag **CNT(CA_company_response)**to Columns and company , **company_response** to Rows to create bar chart.

iv) You can select Show Me and select bar chart to display.

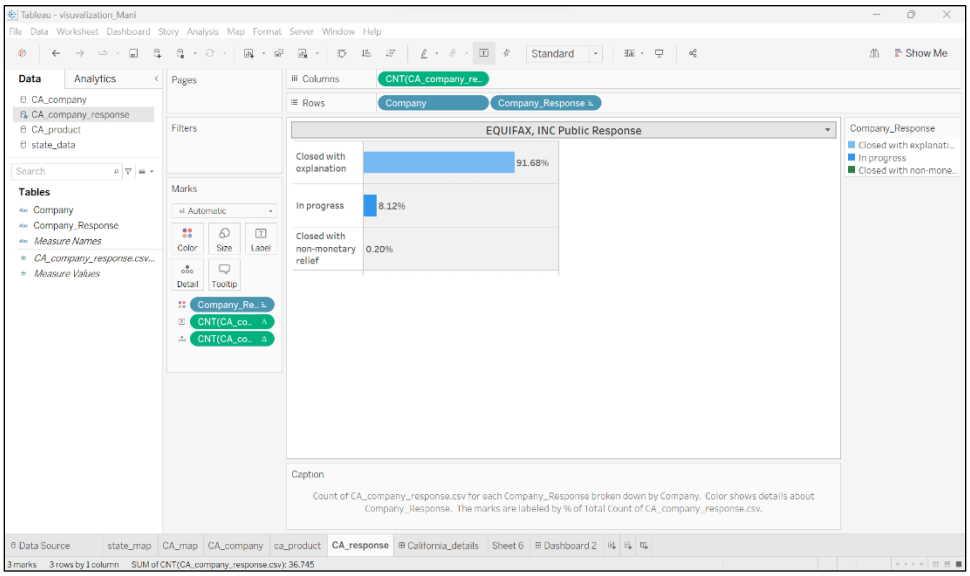
v) For the changing the color of the map, you need to drag the Company_Response to color. Drag CNT(CA_company_Response) and put it into the details. To display the count of Company_Response in the chart , drag the CNT(CA_Compnay_Response) field and put it into the to Label.



vi) After placing the filed **CNT(company_response) in details**, right click and select quick Table Calculation and then click Percent of Total to get the percentage.



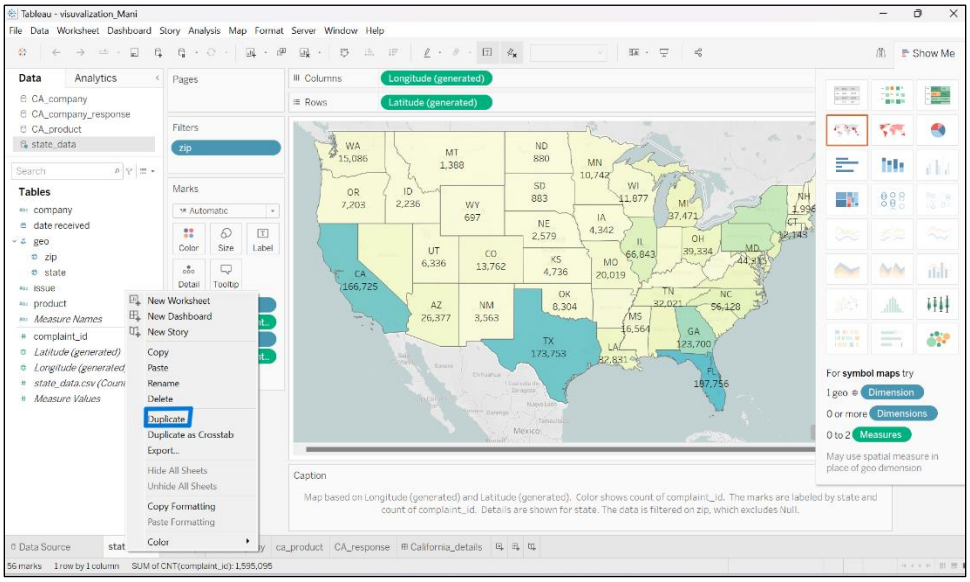
vii) Once you are satisfied with your visualization, save it as a Tableau workbook with .Twbx extension.



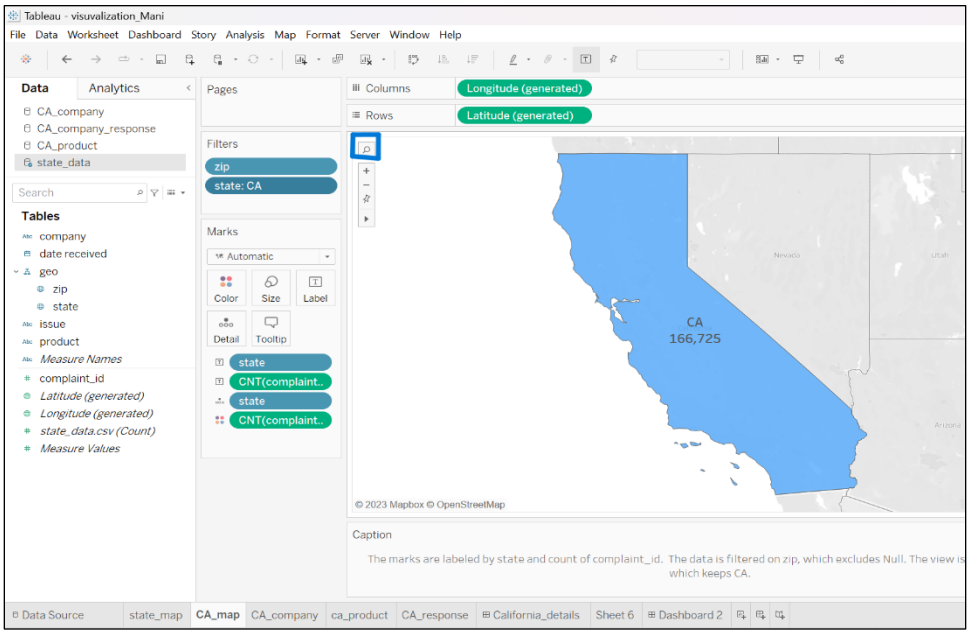
To Create the CA map:

Here are the steps you need to follow to get CA Map separately.

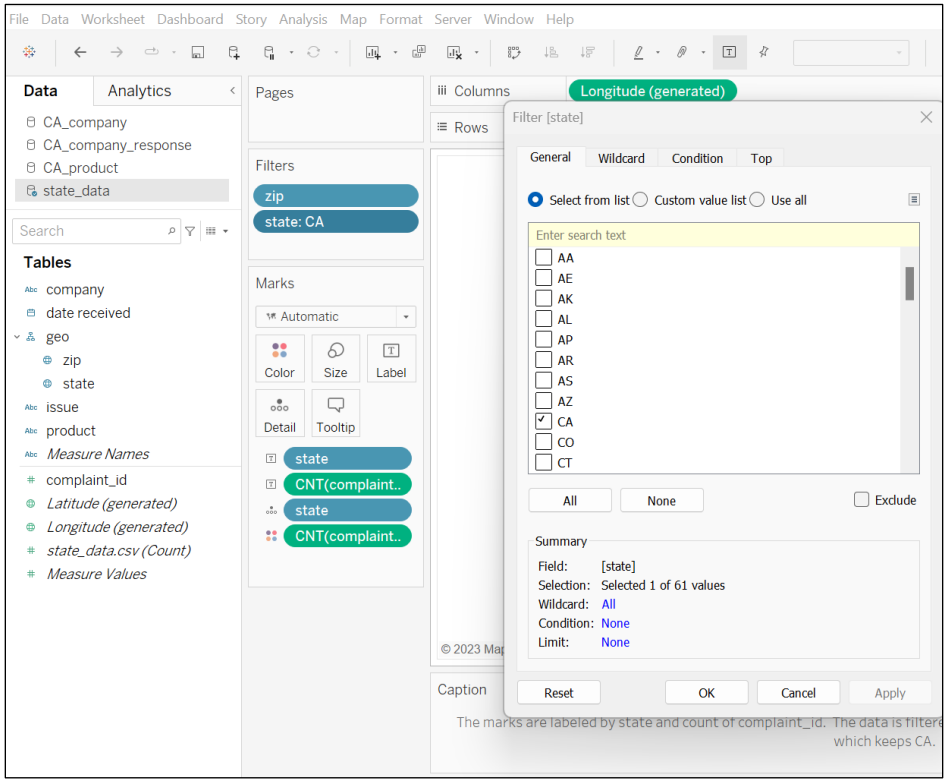
i) Right click on the state_map already created and click on duplicate.



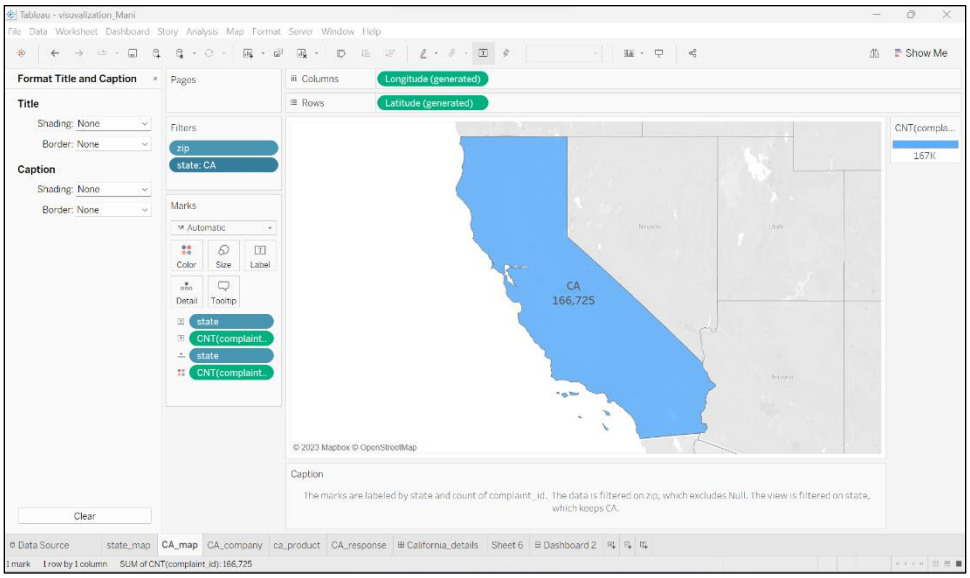
ii) After duplicating the sheet, click on zoom in the map.



iii) Drag state filed to the Filter and select edit filters and then click on select from list and select only CA to show only the California State.

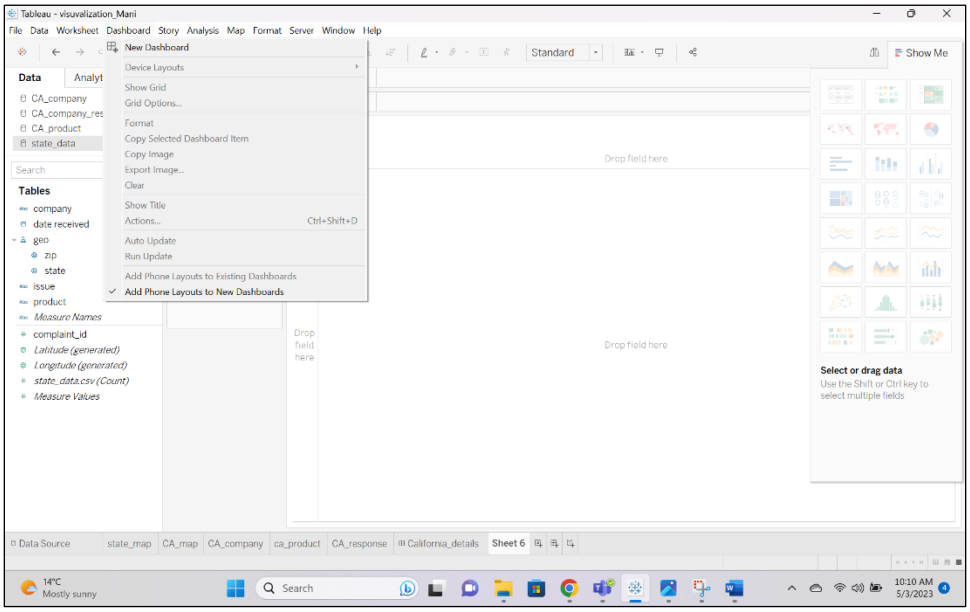


iv) Once you are satisfied with your visualization, save it as a Tableau workbook with .Twbx extension.

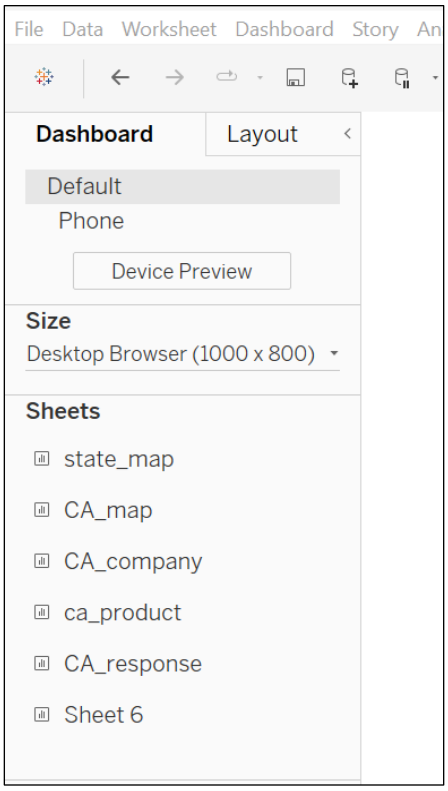


TO CREATE THE DASHBOARD:

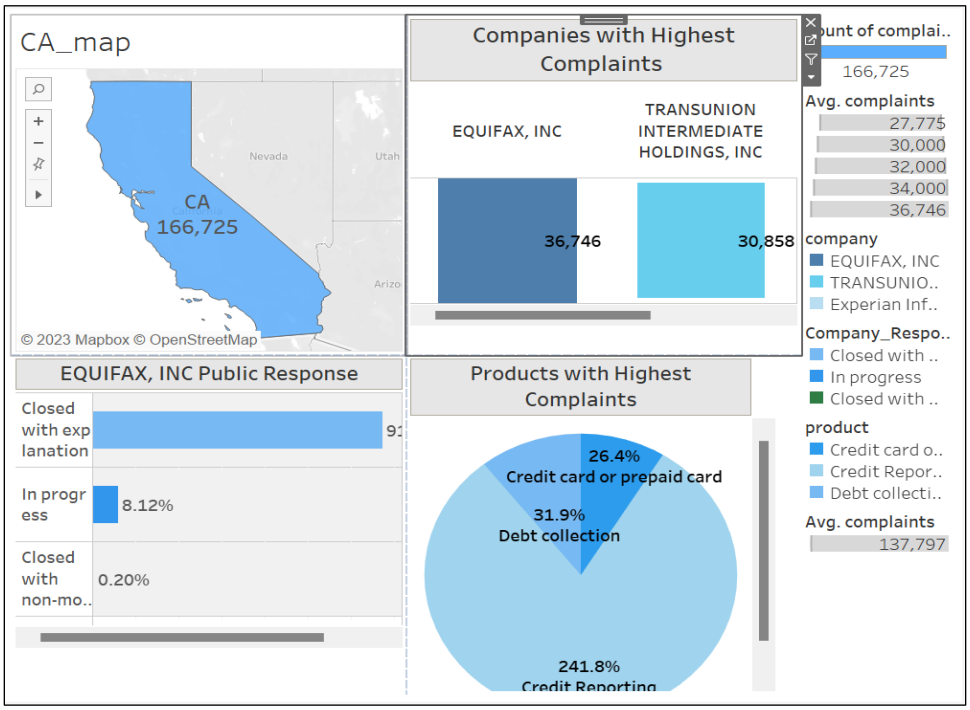
i) In Tableau, create a new Dashboard as shown below



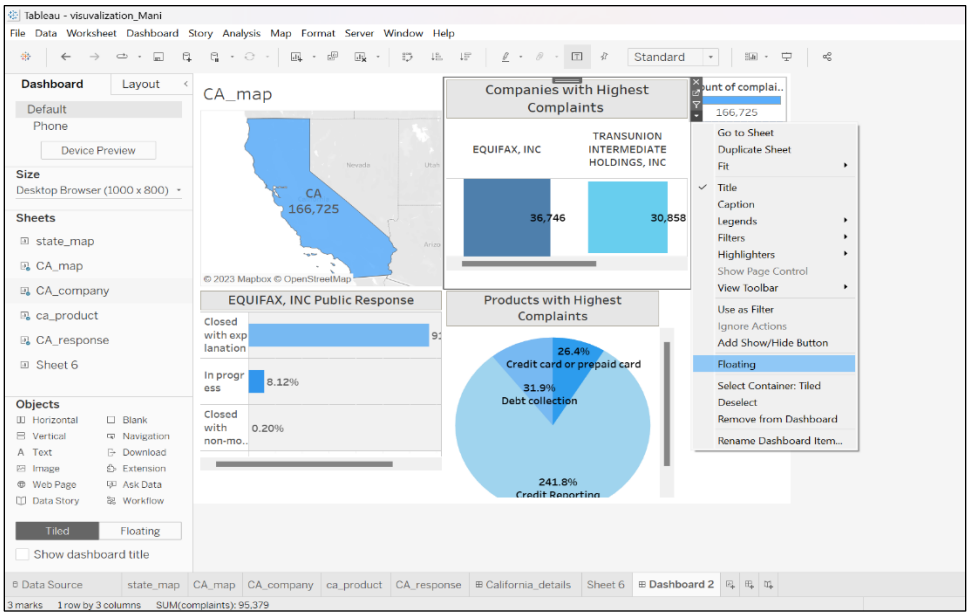
ii) From the left side **Sheets tool bar**, drag CA_map, CA_company, ca_product, CA_response sheets one by one to the “drop sheet here” area.



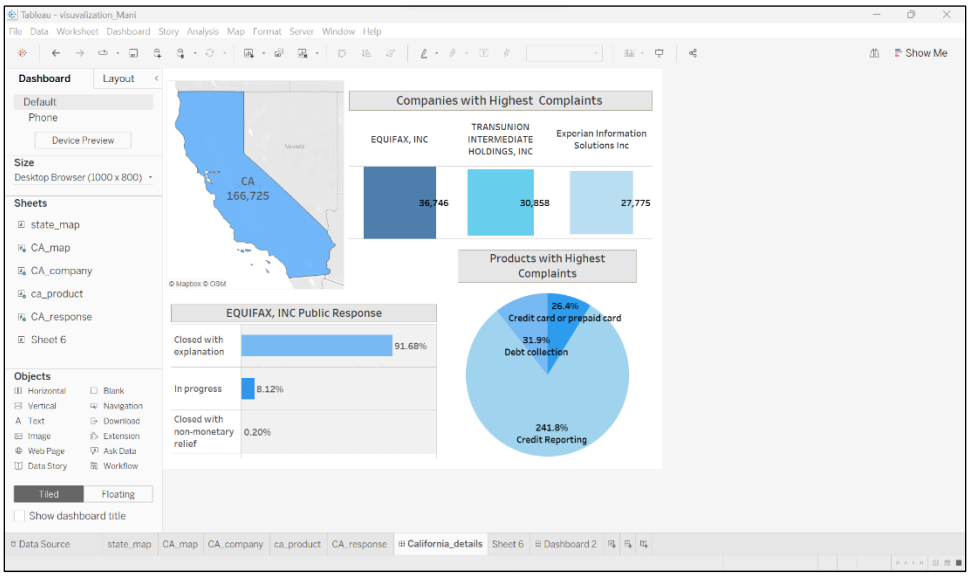
iii) From the left side **Sheets tool bar**, drag all 4 sheets one by one as shown below



iv) To place them to fit in the window as per your need, make the windows **floating** by clicking on each component and select floating as shown below



v) Change the positions, format the views and save the tableau file with .twbx extension with the desired Dashboard once done.



Step 5: Visualization using Power Map in Excel

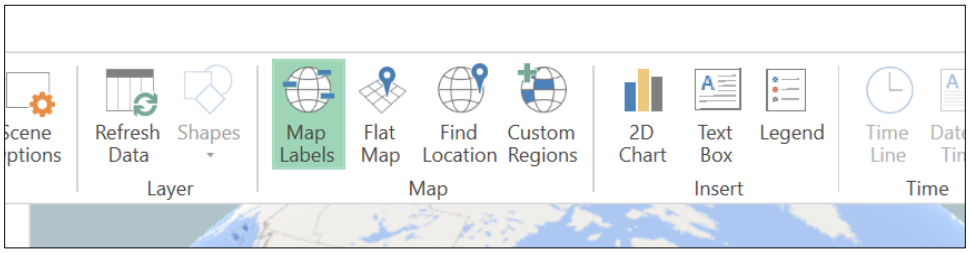
Complaints Narrative Sentiment Analysis Visualization:

i) Open the senti_out.csv in excel

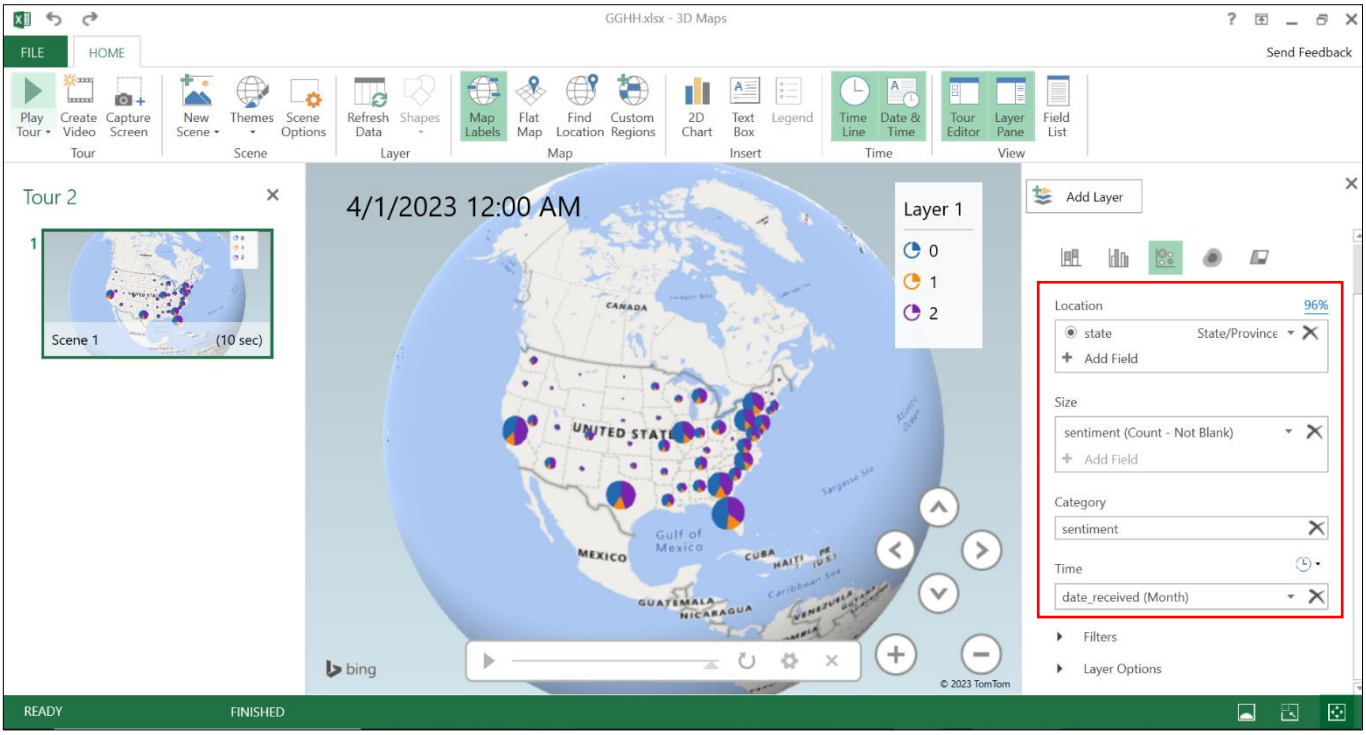
ii) Save the file as excel file (.xlxs) as shown

iii) Go to Insert -> 3d MAP -> Open 3D Maps

iv) Select the “Map Labels” as shown below



v) You need to select the properties and values in the layer as follows. Then, you can drag the earth and rotate it to observe the sentiment of the USA



vi) Now you can kill “tour2” frame in the left. You may move or resize the Layer menu too.

vii) You can change colors from Layer Options for each sentiment type

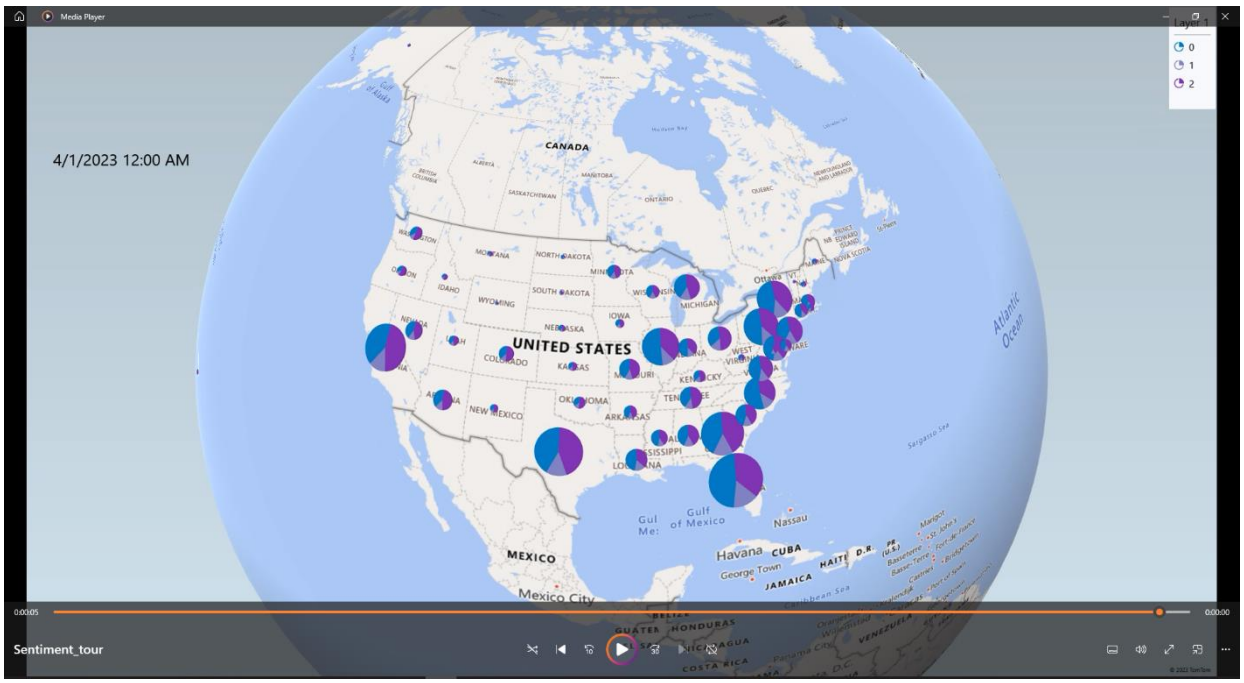
Just for the reference, the numbers are mapped with polarity:

'positive' = 2, 'neutral' = 1, ‘negative' = 0

viii) Click on play button to observe how much complaints have been generating

ix) You can create a video using “Create Video” and save it in your local PC.

x) This is how it looks finally



Conclusion

To sum up, this tutorial introduced you to the utilization of Hadoop Cluster for analyzing consumer complaints' statistics via Apache Hive. The tutorial walked you through the process of uploading the raw data to HDFS, followed by loading it onto Hive tables for executing queries. Lastly, you were taught how to import the query results and create visualizations utilizing tools like Tableau, as well as 3D Map charts within Microsoft Excel.