CIS4560 TERM PROJECT

Authors: Javier Nava, Oscar Munoz, Jesse Fabian, Ky Hoang
Instructor: Jongwook Woo
Date: 12/3/2018

Lab Tutorial:

ANALYZING NEW YORK'S PARKING TICKETS USING HADOOP, APACHE PIG, AND POWER BI

Objectives

In this hands-on-lab, students will:

- Download the data
- Upload data files
- Load data files into Apache Pig
- Filter data in Pig
- Store data into hdfs file
- Use Power BI to Visualize data

Platform Spec

- Oracle Big Data
- Cluster Version: 2.7.1
- Cluster number of nodes: 5 nodes
- OCPUs: 10
- Memory size: 150 GB
- Storage: 678 GB
- HDFS Capacity: 147 GB
- CPU speed: 2195 MHz

Commands for Cluster Versions and CPU speed are as followed:

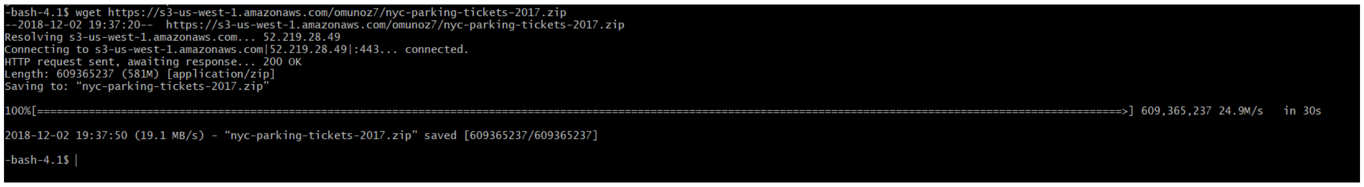
```
$ hdfs version
…
$ lscpu | grep MHz
```

Step 1: Download the data

You need to download the zip file from the following link: https://s3-us-west-1.amazonaws.com/omunoz7/nyc-parking-tickets-2017.zip . From the shell terminal, download the "NYC 2017's parking tickets" zip file by following the command below:

```
$ wget https://s3-us-west-1.amazonaws.com/omunoz7/nyc-parking-tickets-2017.zip
```

Output should look like:



```
-bash-4.1$ wget https://s3-us-west-1.amazonaws.com/omunoz7/nyc-parking-tickets-2017.zip
--2018-12-02 19:37:20--  https://s3-us-west-1.amazonaws.com/omunoz7/nyc-parking-tickets-2017.zip
Resolving s3-us-west-1.amazonaws.com... 52.219.28.49
Connecting to s3-us-west-1.amazonaws.com|52.219.28.49|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 609365237 (581M) [application/zip]
Saving to: "nyc-parking-tickets-2017.zip"

100%[==================================================================================================================================>] 609,365,237 24.9M/s   in 30s

2018-12-02 19:37:50 (19.1 MB/s) - "nyc-parking-tickets-2017.zip" saved [609365237/609365237]

-bash-4.1$ |
```

Step 2: Unzip and upload the data files

After you have successfully download the file, unzip the file to get the csv data file. Remove the zip file in the home directory to free up storage. Use the "ls" command to confirm that the zip file is removed and the unzipped file is present. Create a new directory called "parking" in Hadoop Distributed File System, then move the unzipped csv file to the new directory using the "-put" command. Follow the commands below:

```
$ unzip nyc-parking-tickets.zip

$ rm nyc-parking-tickets.zip

$ ls

$ hdfs dfs -mkdir parking

$ hdfs dfs -put  Parking_Violations_Issued_-_Fiscal_Year_2017.csv ./parking/

$ hdfs dfs -mv  ./parking/Parking_Violations_Issued_-

_Fiscal_Year_2017.csv./parking/tickets.csv

$ hdfs dfs -ls
```

```
Found 4 items
drwxr-xr-x   - khoang21 hdfs          0 2018-11-07 22:05 .hiveJars
drwx------   - khoang21 hdfs          0 2018-11-28 22:35 .staging
```

```
drwxr-xr-x   - khoang21 hdfs          0 2018-11-28 21:40 dualcore
```
drwxr-xr-x   - khoang21 hdfs   0 2018-12-03 00:15 parking

Check that the file is moved to the directory. There should be an item found. To check that the data file is correct, you can display the first 20 records of the data set by using the "-cat" command. Follow the steps by using the commands below:

```
$ hdfs dfs -ls parking
Found 1 items
-rw-r--r--   2 khoang21 hdfs 2086913576 2018-12-03 00:12 parking/tickets.csv
$ hdfs dfs -cat ./parking/tickets.csv | head -20
```

Results should look like the following:

```
…
…
1413656420,T672371C,NY,PAS,02/04/2017,40,TAXI,TOYOT,P,59630,73470,82230,20170531,
        0073,73,73,960758,0073,0000,0525A,,K,F,279,MCDOUGAL
        ST,,0,408,F1,,BBBBBBB,ALL,ALL,BLK,0,2015,-,0,,,,,
7959486440,GYF2052,NY,PAS,07/07/2016,71,4DSD,VOLKS,T,70407,17390,48890,20170527,
        120,120,120,356557,T303,E,0645P,,R,F,143,N St Austins Pl,,0,408,j6,,,,,GY,,2012,,0,C-
        PM,71-Insp. Sticker Missing (NYS,,,
5093620865,AD80228,AZ,PAS,09/24/2016,7,TK,FORD,V,0,0,0,0,,0,0,0,,,1122A,,QN,,,SO
        CONDUIT AVE (E/B),@ 89TH ST,0,1111,D,T,,,,,,2009,,0,,FAILURE TO STOP AT RED
        LIGHT,,,
8480309064,51771JW,NY,COM,01/26/2017,64,VAN,INTER,T,17850,10210,10110,88888888,0
        017,17,17,363557,T102,L,0256P,,NY,F,204,E 43rd
        St,,0,408,C8,,YYYYYYY,,,BROWN,,2007,,0,06,"64-No STD Ex Con/DPL, D/S Dec",,,
1416638830,GLP367,NY,PAS,04/30/2017,20,SUBN,DODGE,P,17650,10110,10010,20180304,
        0017,17,17,940179,0017,0000,1232A,,NY,O,330,E 33
        ST,,0,408,E2,,YYYYYYY,1200A,1159P,BLK,0,2009,-,0,,,,,
```

Step 3: Load data files into Pig

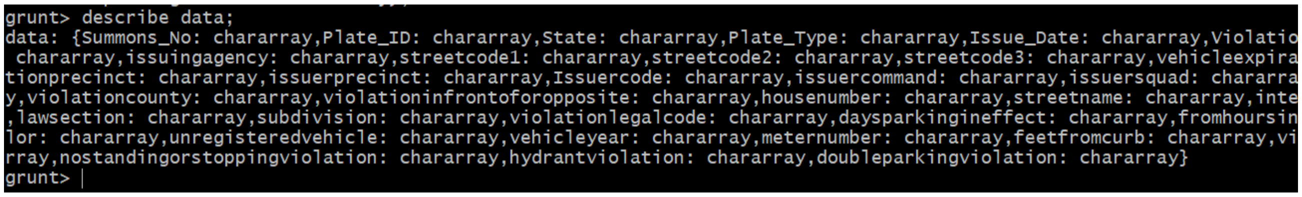
Once the data file has been move to the new directory and its records have been checked, you will need to use Apache pig to clean and filter the data set before visualizing the data. First off, you will need to go into pig by writing the following command:

```
-bash-4.1$ pig
```

Then you will load the dataset located in your hadoop directory using PigStorage. Be sure to include all the fields and their field type, which are listed below. After successfully running the LOAD command, use the DESCRIBE command to confirm all fields and their type.

```
grunt> data = LOAD './parking/tickets.csv' using PigStorage(',') AS
    (Summons_No:chararray,
      Plate_ID:chararray, State:chararray,
      Plate_Type:chararray, Issue_Date:chararray,
      Violation_code:int, Vehicle_Body_Type:chararray,
    Vehicle_make:chararray,
    issuingagency:chararray,
    streetcode1:chararray,
    streetcode2:chararray,
    streetcode3:chararray,
    vehicleexpirationdate:chararray,
    violationlocation:chararray,
    violationprecinct:chararray,
    issuerprecinct:chararray,
    Issuercode:chararray,
    issuercommand:chararray,
    issuersquad:chararray,
    violationtime:chararray,
    timefirstobserved:chararray,
    violationcounty:chararray,
    violationinfrontoforopposite:chararray,
    housenumber:chararray,
    streetname:chararray,
    intersectingstreet:chararray,
    datefirstobserved:chararray,
    lawsection:chararray,
    subdivision:chararray,
    violationlegalcode:chararray,
    daysparkingineffect:chararray,
    fromhoursineffect:chararray,
    tohoursineffect:chararray,
    vehiclecolor:chararray,
    unregisteredvehicle:chararray,
    vehicleyear:chararray,
    meternumber:chararray,
    feetfromcurb:chararray,
    violationpostcode:chararray,
    violationdescription:chararray,
    nostandingorstoppingviolation:chararray,
    hydrantviolation:chararray,
    doubleparkingviolation:chararray);
grunt> DESCRIBE data;
```

Result of the DESCRIBE command should look like the following:



```
grunt> describe data;
data: {Summons_No: chararray,Plate_ID: chararray,State: chararray,Plate_Type: chararray,Issue_Date: chararray,Violatio
 chararray,issuingagency: chararray,streetcode1: chararray,streetcode2: chararray,streetcode3: chararray,vehicleexpira
tionprecinct: chararray,issuerprecinct: chararray,Issuercode: chararray,issuercommand: chararray,issuersquad: chararra
y,violationcounty: chararray,violationinfrontoforopposite: chararray,housenumber: chararray,streetname: chararray,inte
,lawsection: chararray,subdivision: chararray,violationlegalcode: chararray,daysparkingineffect: chararray,fromhoursin
lor: chararray,unregisteredvehicle: chararray,vehicleyear: chararray,meternumber: chararray,feetfromcurb: chararray,vi
rray,nostandingorstoppingviolation: chararray,hydrantviolation: chararray,doubleparkingviolation: chararray}
grunt> |
```

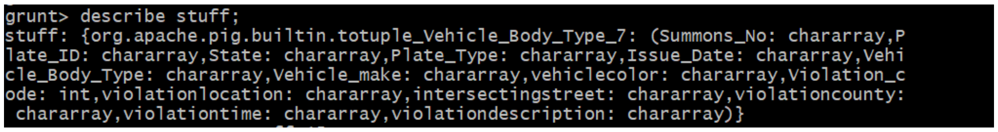
Step 4: Filter data

Now, you need to filter the data in Pig to show only the data necessary to answer the questions.

Some of these fields include the Plate_ID, vehicle_color, etc. This code will make a new record for each record in the ‘data’ table that only contains necessary information.

```
grunt> stuff = FOREACH data Generate
(Summons_No,
      Plate_ID,
    State,
    Plate_Type,
    Issue_Date,
    Vehicle_Body_Type,
    Vehicle_make,
    Vehiclecolor,
    Violation_code,
    Violationlocation,
    Intersectingstreet,
    Violationcounty,
    Violationtime,
    violationdescription);
grunt> DESCRIBE stuff;
```

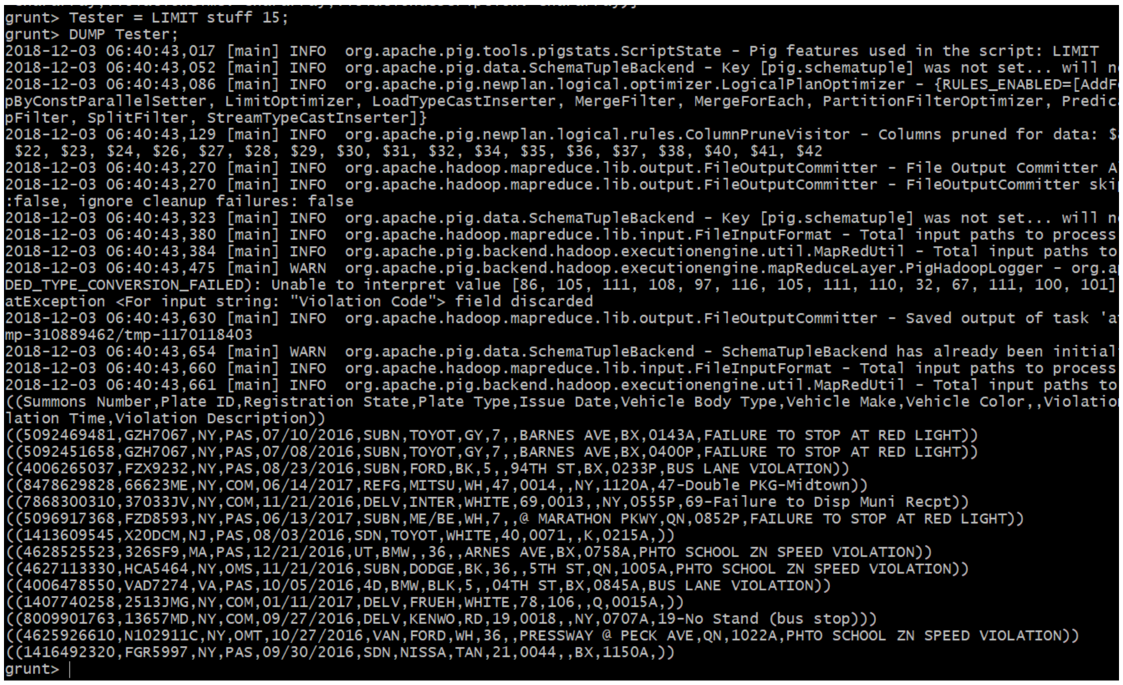
After running the code, use DESCRIBE to test that the new data set was created correctly with all necessary fields.



```
grunt> describe stuff;
stuff: {org.apache.pig.builtin.totuple_Vehicle_Body_Type_7: (Summons_No: chararray,P
late_ID: chararray,State: chararray,Plate_Type: chararray,Issue_Date: chararray,Vehi
cle_Body_Type: chararray,Vehicle_make: chararray,vehiclecolor: chararray,Violation_c
ode: int,violationlocation: chararray,intersectingstreet: chararray,violationcounty:
 chararray,violationtime: chararray,violationdescription: chararray)}
```

After checking that ‘stuff’ was created correctly with all proper fields, you want to test that the data is formatted correctly. For this reason, you will create ‘Tester’ that will copy a limited number of records from ‘stuff’ by using the LIMIT function. You then DUMP ‘Tester’ to check that the records are following the correct format.

```
grunt> Tester = LIMIT stuff 15;
grunt> DUMP Tester;
```



```
grunt> Tester = LIMIT stuff 15;
grunt> DUMP Tester;
2018-12-03 06:40:43,017 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: LIMIT
2018-12-03 06:40:43,052 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will n
2018-12-03 06:40:43,086 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddF
pByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, PartitionFilterOptimizer, Predic
pFilter, SplitFilter, StreamTypeCastInserter]}
2018-12-03 06:40:43,129 [main] INFO  org.apache.pig.newplan.logical.rules.ColumnPruneVisitor - Columns pruned for data: $
 $22, $23, $24, $26, $27, $28, $29, $30, $31, $32, $34, $35, $36, $37, $38, $40, $41, $42
2018-12-03 06:40:43,270 [main] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - File Output Committer A
2018-12-03 06:40:43,270 [main] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - FileOutputCommitter ski
:false, ignore cleanup failures: false
2018-12-03 06:40:43,323 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will n
2018-12-03 06:40:43,380 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process
2018-12-03 06:40:43,384 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to
2018-12-03 06:40:43,475 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.PigHadoopLogger - org.a
DED_TYPE_CONVERSION_FAILED): Unable to interpret value [86, 105, 111, 108, 97, 116, 105, 111, 110, 32, 67, 111, 100, 101]
atException <For input string: "Violation Code"> field discarded
2018-12-03 06:40:43,630 [main] INFO  org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter - Saved output of task 'a
mp-310889462/tmp-1170118403
2018-12-03 06:40:43,654 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initial
2018-12-03 06:40:43,660 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process
2018-12-03 06:40:43,661 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to
((Summons Number,Plate ID,Registration State,Plate Type,Issue Date,Vehicle Body Type,Vehicle Make,Vehicle Color,,Violatio
lation Time,Violation Description))
((5092469481,GZH7067,NY,PAS,07/10/2016,SUBN,TOYOT,GY,7,,BARNES AVE,BX,0143A,FAILURE TO STOP AT RED LIGHT))
((5092451658,GZH7067,NY,PAS,07/08/2016,SUBN,TOYOT,GY,7,,BARNES AVE,BX,0400P,FAILURE TO STOP AT RED LIGHT))
((4006265037,FZX9232,NY,PAS,08/23/2016,SUBN,FORD,BK,5,,94TH ST,BX,0233P,BUS LANE VIOLATION))
((8478629828,66623ME,NY,COM,06/14/2017,REFG,MITSU,WH,47,0014,,NY,1120A,47-Double PKG-Midtown))
((7868300310,37033JV,NY,COM,11/21/2016,DELV,INTER,WHITE,69,0013,,NY,0555P,69-Failure to Disp Muni Recpt))
((5096917368,FZD8593,NY,PAS,06/13/2017,SUBN,ME/BE,WH,7,,@ MARATHON PKWY,QN,0852P,FAILURE TO STOP AT RED LIGHT))
((1413609545,X20DCM,NJ,PAS,08/03/2016,SDN,TOYOT,WHITE,40,0071,,K,0215A,))
((4628525523,326SF9,MA,PAS,12/21/2016,UT,BMW,,36,,ARNES AVE,BX,0758A,PHTO SCHOOL ZN SPEED VIOLATION))
((4627113330,HCA5464,NY,OMS,11/21/2016,SUBN,DODGE,BK,36,,5TH ST,QN,1005A,PHTO SCHOOL ZN SPEED VIOLATION))
((4006478550,VAD7274,VA,PAS,10/05/2016,4D,BMW,BLK,5,,04TH ST,BX,0845A,BUS LANE VIOLATION))
((1407740258,2513JMG,NY,COM,01/11/2017,DELV,FRUEH,WHITE,78,106,,Q,0015A,))
((8009901763,13657MD,NY,COM,09/27/2016,DELV,KENWO,RD,19,0018,,NY,0707A,19-No Stand (bus stop)))
((4625926610,N102911C,NY,OMT,10/27/2016,VAN,FORD,WH,36,,PRESSWAY @ PECK AVE,QN,1022A,PHTO SCHOOL ZN SPEED VIOLATION))
((1416492320,FGR5997,NY,PAS,09/30/2016,SDN,NISSA,TAN,21,0044,,BX,1150A,))
grunt> |
```

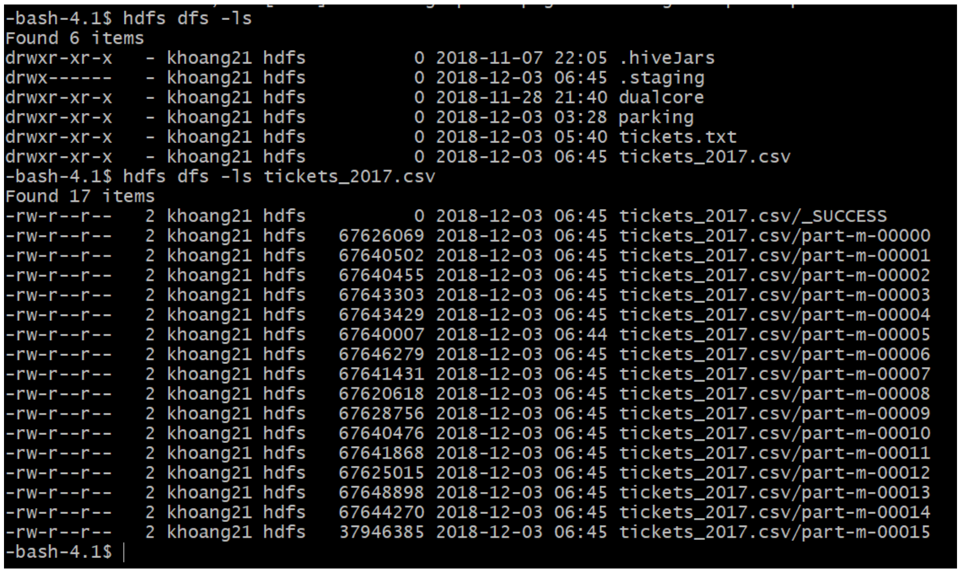
Step 5: Export data

After making sure that the data in ‘stuff’ contains all the necessary data, you want to store ‘stuff’ into the HDFS using the following command.

```
grunt> STORE stuff INTO ‘tickets_2017.csv’ USING Pigstorage(‘,’);
…
grunt> QUIT;
```

To make sure that Pig saved the results of the the filtering, you want to find your filename within our HDFS by using the -ls command. Once the file is found, you will want to see if the data is in the directory by using the second -ls command.

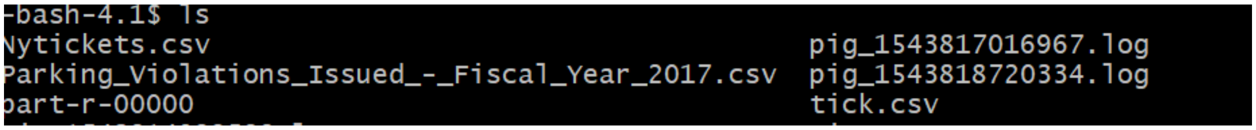
```
$ hdfs dfs -ls
…
$ hdfs dfs -ls tickets_2017.csv
```



```
-bash-4.1$ hdfs dfs -ls
Found 6 items
drwxr-xr-x   - khoang21 hdfs          0 2018-11-07 22:05 .hiveJars
drwx------   - khoang21 hdfs          0 2018-12-03 06:45 .staging
drwxr-xr-x   - khoang21 hdfs          0 2018-11-28 21:40 dualcore
drwxr-xr-x   - khoang21 hdfs          0 2018-12-03 03:28 parking
drwxr-xr-x   - khoang21 hdfs          0 2018-12-03 05:40 tickets.txt
drwxr-xr-x   - khoang21 hdfs          0 2018-12-03 06:45 tickets_2017.csv
-bash-4.1$ hdfs dfs -ls tickets_2017.csv
Found 17 items
-rw-r--r--   2 khoang21 hdfs          0 2018-12-03 06:45 tickets_2017.csv/_SUCCESS
-rw-r--r--   2 khoang21 hdfs   67626069 2018-12-03 06:45 tickets_2017.csv/part-m-00000
-rw-r--r--   2 khoang21 hdfs   67640502 2018-12-03 06:45 tickets_2017.csv/part-m-00001
-rw-r--r--   2 khoang21 hdfs   67640455 2018-12-03 06:45 tickets_2017.csv/part-m-00002
-rw-r--r--   2 khoang21 hdfs   67643303 2018-12-03 06:45 tickets_2017.csv/part-m-00003
-rw-r--r--   2 khoang21 hdfs   67643429 2018-12-03 06:45 tickets_2017.csv/part-m-00004
-rw-r--r--   2 khoang21 hdfs   67640007 2018-12-03 06:44 tickets_2017.csv/part-m-00005
-rw-r--r--   2 khoang21 hdfs   67646279 2018-12-03 06:45 tickets_2017.csv/part-m-00006
-rw-r--r--   2 khoang21 hdfs   67641431 2018-12-03 06:45 tickets_2017.csv/part-m-00007
-rw-r--r--   2 khoang21 hdfs   67620618 2018-12-03 06:45 tickets_2017.csv/part-m-00008
-rw-r--r--   2 khoang21 hdfs   67628756 2018-12-03 06:45 tickets_2017.csv/part-m-00009
-rw-r--r--   2 khoang21 hdfs   67640476 2018-12-03 06:45 tickets_2017.csv/part-m-00010
-rw-r--r--   2 khoang21 hdfs   67641868 2018-12-03 06:45 tickets_2017.csv/part-m-00011
-rw-r--r--   2 khoang21 hdfs   67625015 2018-12-03 06:45 tickets_2017.csv/part-m-00012
-rw-r--r--   2 khoang21 hdfs   67648898 2018-12-03 06:45 tickets_2017.csv/part-m-00013
-rw-r--r--   2 khoang21 hdfs   67644270 2018-12-03 06:45 tickets_2017.csv/part-m-00014
-rw-r--r--   2 khoang21 hdfs   37946385 2018-12-03 06:45 tickets_2017.csv/part-m-00015
-bash-4.1$ |
```

After finding that all the data has been saved as different part files, you want to merge all the data into one complete file of type CSV. After this is done, you will want to check that the file exists on your local file system, so run an ls command.

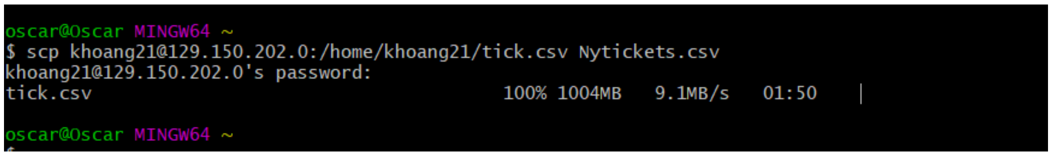
```
$ hdfs dfs -cat ./tickets_2017.csv/part* > tick.csv
$ ls
```



```
-bash-4.1$ ls
Nytickets.csv                                   pig_1543817016967.log
Parking_Violations_Issued_-_Fiscal_Year_2017.csv  pig_1543818720334.log
part-r-00000                                    tick.csv
```

In a new terminal Write out the following command to download the tick.csv file and rename it as Nytickets.csv. Make sure to change to your username when writing the command.

```
$ scp username@129.150.202.0:/home/username/tick.csv Nytickets.csv
```



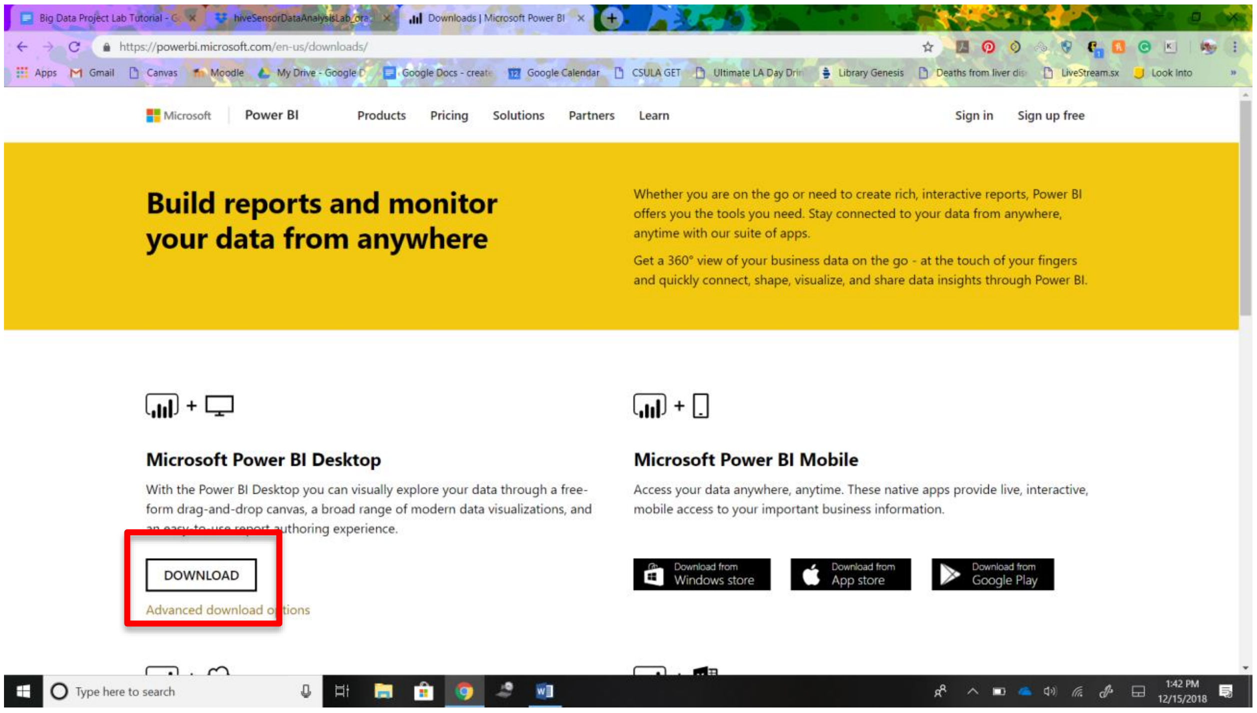
```
oscar@Oscar MINGW64 ~
$ scp khoang21@129.150.202.0:/home/khoang21/tick.csv Nytickets.csv
khoang21@129.150.202.0's password:
tick.csv                                      100% 1004MB   9.1MB/s   01:50    |

oscar@Oscar MINGW64 ~
```

Step 6: Visualize data using Power BI

Open the Power BI app or go to https://app.powerbi.com to download the application.

1. After downloading and installing your application, go ahead and open it.



Microsoft | Power BI | Products | Pricing | Solutions | Partners | Learn | Sign in | Sign up free

Build reports and monitor your data from anywhere

Whether you are on the go or need to create rich, interactive reports, Power BI offers you the tools you need. Stay connected to your data from anywhere, anytime with our suite of apps.

Get a 360° view of your business data on the go - at the touch of your fingers and quickly connect, shape, visualize, and share data insights through Power BI.

Microsoft Power BI Desktop

With the Power BI Desktop you can visually explore your data through a free-form drag-and-drop canvas, a broad range of modern data visualizations, and an easy-to-use report authoring experience.

DOWNLOAD

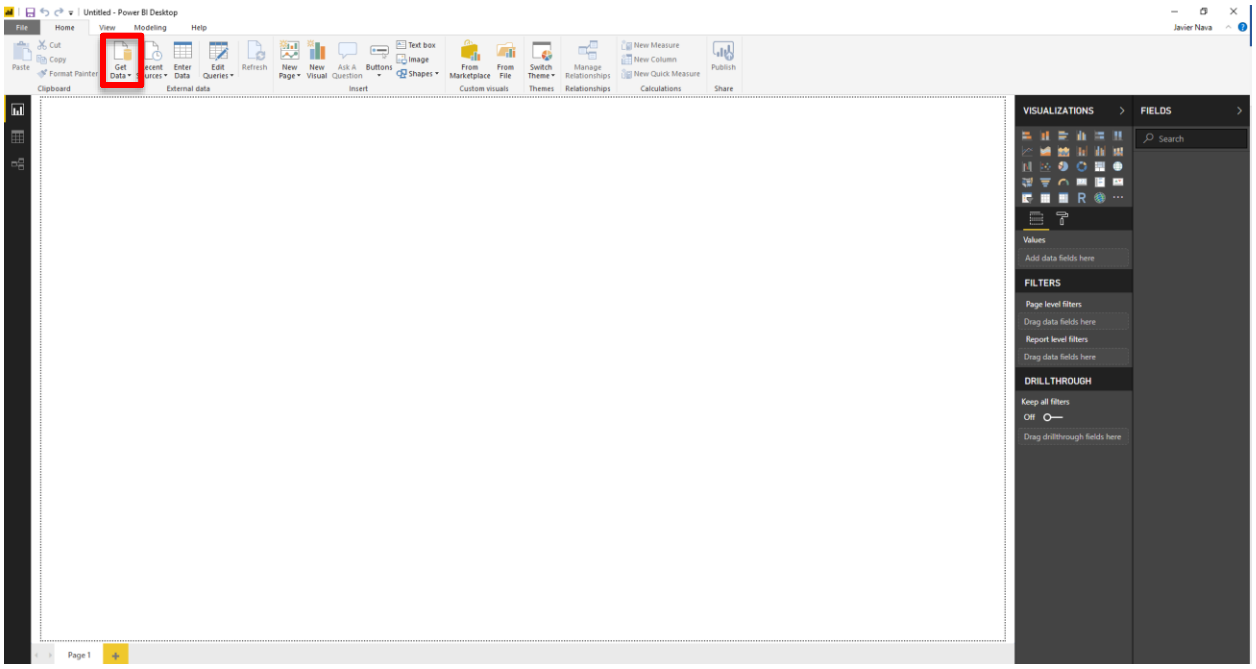
Advanced download options

Microsoft Power BI Mobile

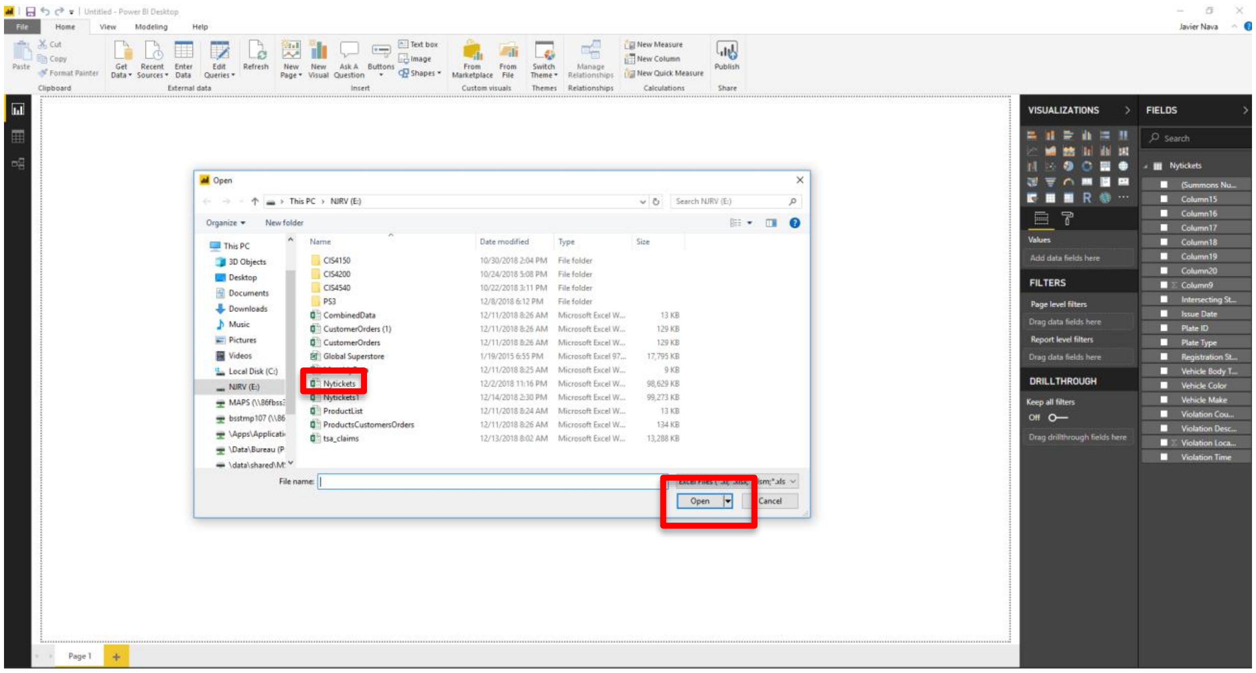
Access your data anywhere, anytime. These native apps provide live, interactive, mobile access to your important business information.

Download from Windows store | Download from App store | Download from Google Play

2. Once you open Power BI, it is time to import our dataset. To import your dataset, go ahead and click on the “Get Data” tab on the top and select on the excel file.

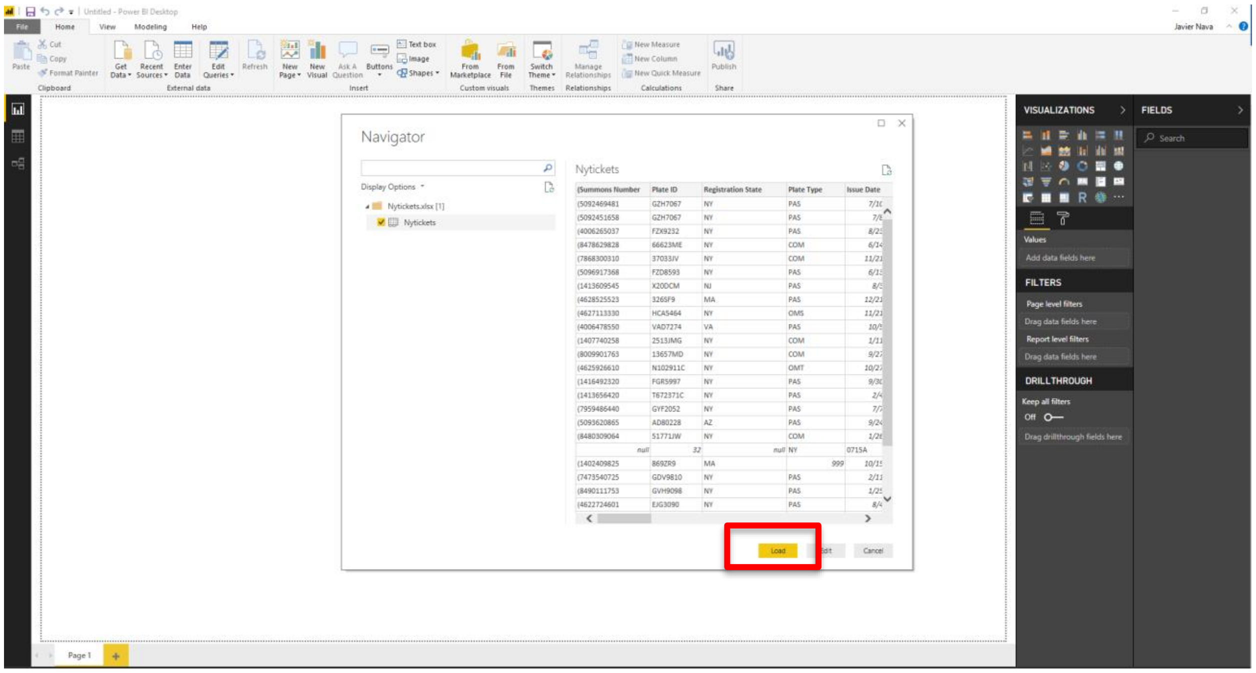


3. When clicking on ‘Get Data’ you will see this window. Once you see this window go ahead and select our dataset.



Name	Date modified	Type	Size
CIS4150	10/30/2018 2:04 PM	File folder	
CIS4200	10/24/2018 5:08 PM	File folder	
CIS4540	10/22/2018 3:11 PM	File folder	
PS3	12/8/2018 6:12 PM	File folder	
CombinedData	12/11/2018 8:26 AM	Microsoft Excel W...	13 KB
CustomerOrders (1)	12/11/2018 8:26 AM	Microsoft Excel W...	129 KB
CustomerOrders	12/11/2018 8:26 AM	Microsoft Excel W...	129 KB
Global Superstore	1/19/2015 6:55 PM	Microsoft Excel 97...	17,795 KB
[illegible]	12/11/2018 8:25 AM	Microsoft Excel W...	9 KB
Nytickets	12/2/2018 11:16 PM	Microsoft Excel W...	98,629 KB
Nytickets1	12/14/2018 2:30 PM	Microsoft Excel W...	99,273 KB
ProductList	12/11/2018 8:24 AM	Microsoft Excel W...	13 KB
ProductsCustomersOrders	12/11/2018 8:26 AM	Microsoft Excel W...	134 KB
tsa_claims	12/13/2018 8:02 AM	Microsoft Excel W...	13,288 KB

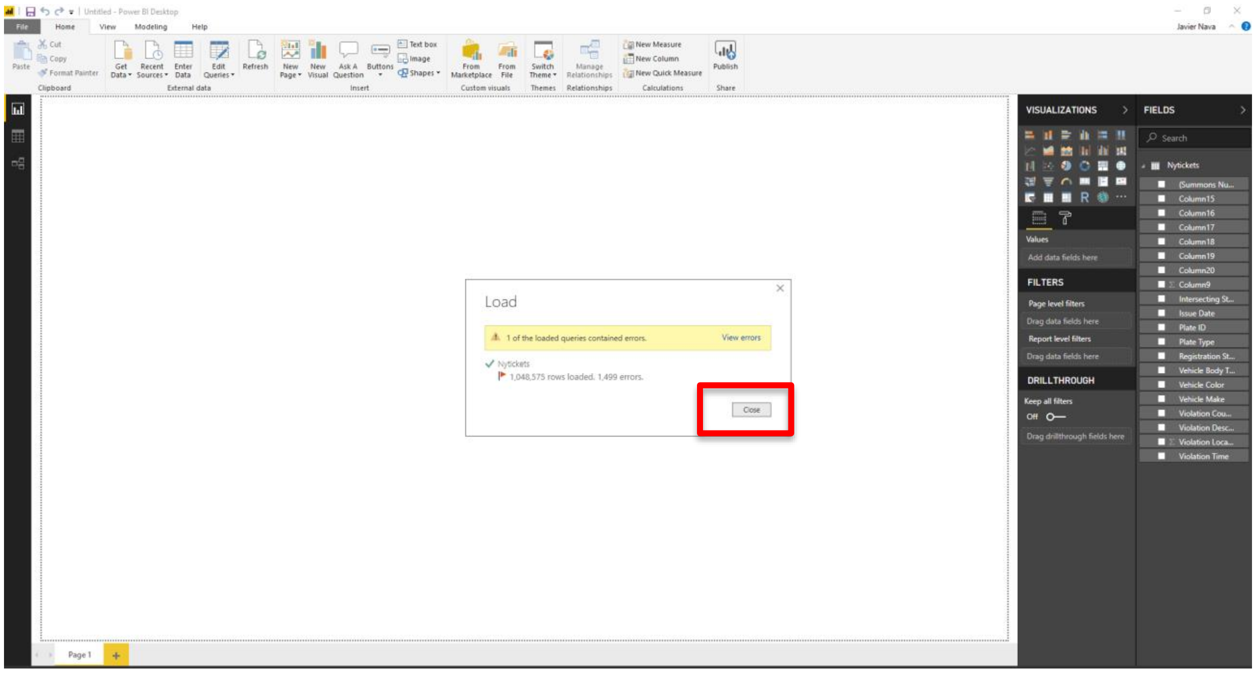
4. After doing so, go ahead and click on “Nytickets” to bring this data into our data visualization tool. Once doing so go ahead and click on the Yellow load button.



Nytickets

(Summons Number	Plate ID	Registration State	Plate Type	Issue Date
(5092469481	GZH7067	NY	PAS	7/1(
(5092451658	GZH7067	NY	PAS	7/8
(4006265037	FZX9232	NY	PAS	8/2
(8478629828	66623ME	NY	COM	6/14
(7868300310	37033JV	NY	COM	11/21
(5096917368	FZD8593	NY	PAS	6/1
(1413609545	X20DCM	NJ	PAS	8/
(4628525523	326SF9	MA	PAS	12/21
(4627113330	HCA5464	NY	OMS	11/21
(4006478550	VAD7274	VA	PAS	10/
(1407740258	2513JMG	NY	COM	1/11
(8009901763	13657MD	NY	COM	9/2
(4625926610	N102911C	NY	OMT	10/2
(1416492320	FGR5997	NY	PAS	9/3
(1413656420	T672371C	NY	PAS	2/
(7959486440	GYF2052	NY	PAS	7/
(5093620865	AD80228	AZ	PAS	9/2
(8480309064	51771JW	NY	COM	1/2
null	32	null	NY	0715A
(1402409825	869ZR9	MA	999	10/1
(7473540725	GDV9810	NY	PAS	2/11
(8490111753	GVH9098	NY	PAS	1/2
(4622724601	EJG3090	NY	PAS	8/

5. After loading the dataset, you will get this pop up window. Go ahead and click on the close option.



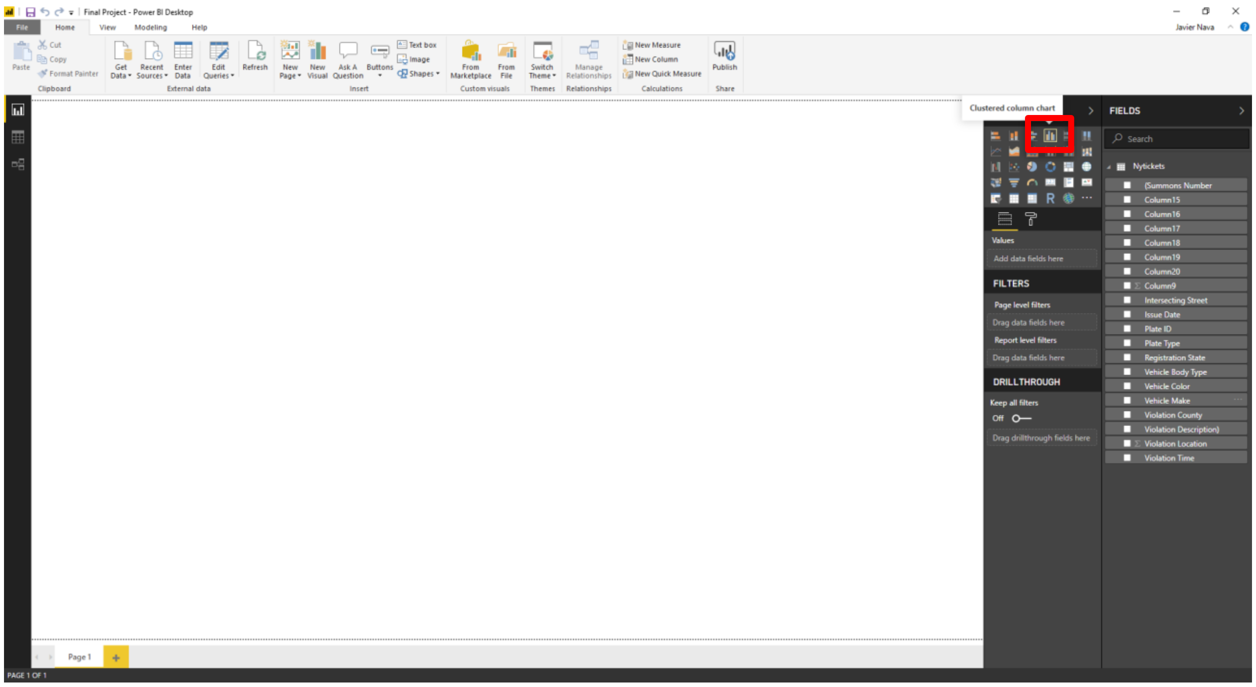
Load

1 of the loaded queries contained errors. View errors

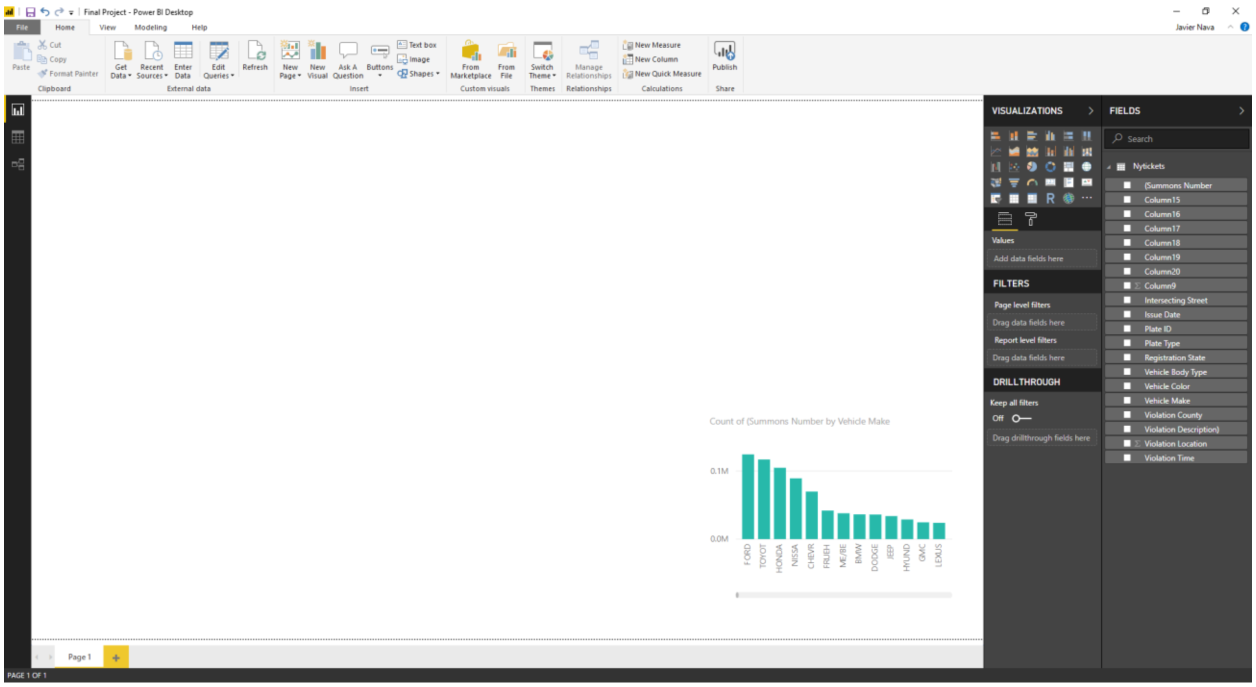
Nytickets

1,048,575 rows loaded. 1,499 errors.

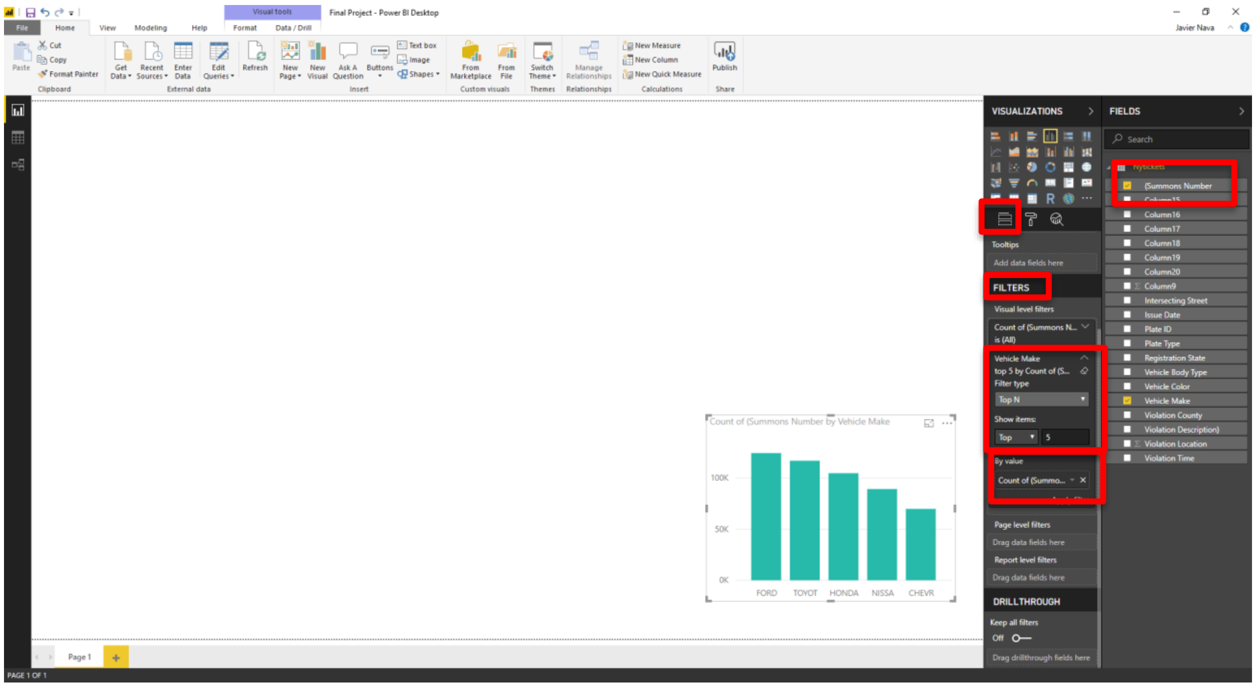
6. You are now ready to create the first visual. We will be creating a column chart. Click on the option in red.



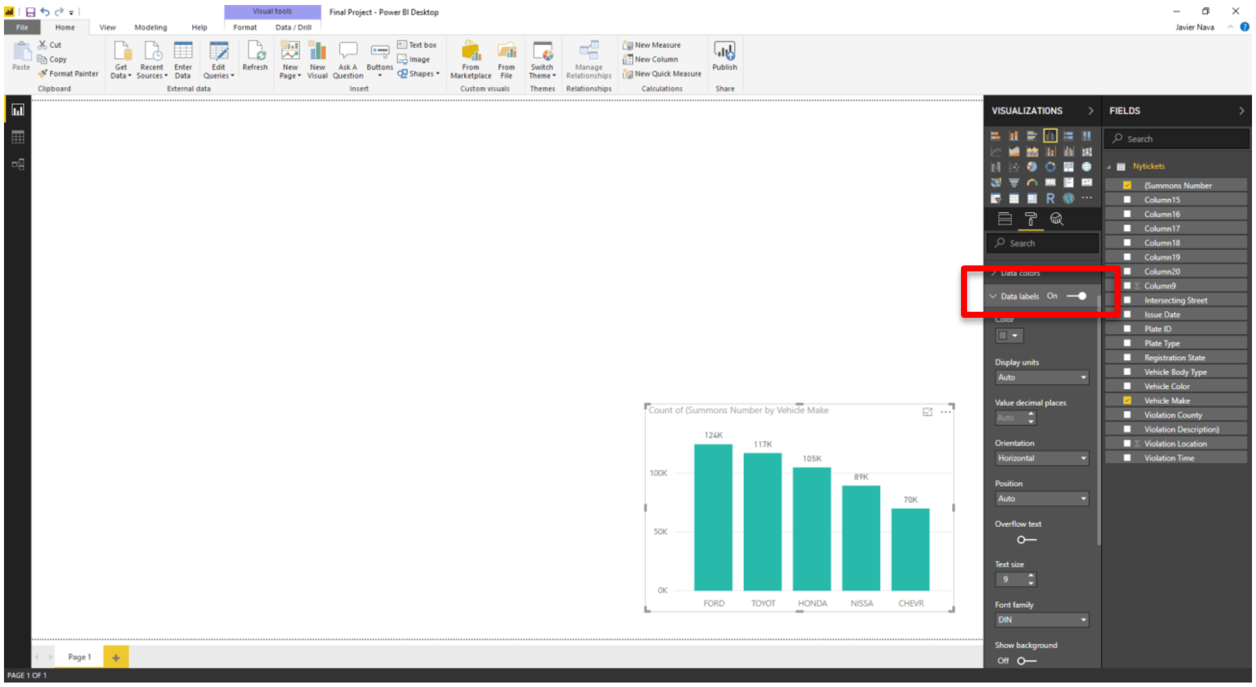
7. Next, we are going to



8. We will now Filter our data to display the top 5 values. To do this we are going to scroll down to the filters section. Once you do so click on **'Vehicle Made**. Next Change the filter type to **'top N'**. After doing so change **'show items'** to **'top' 5**. Lastly, drag **'(Summons Number'** to '**By Value'**. After doing so, change '**by value'** to **count of (summons number**.

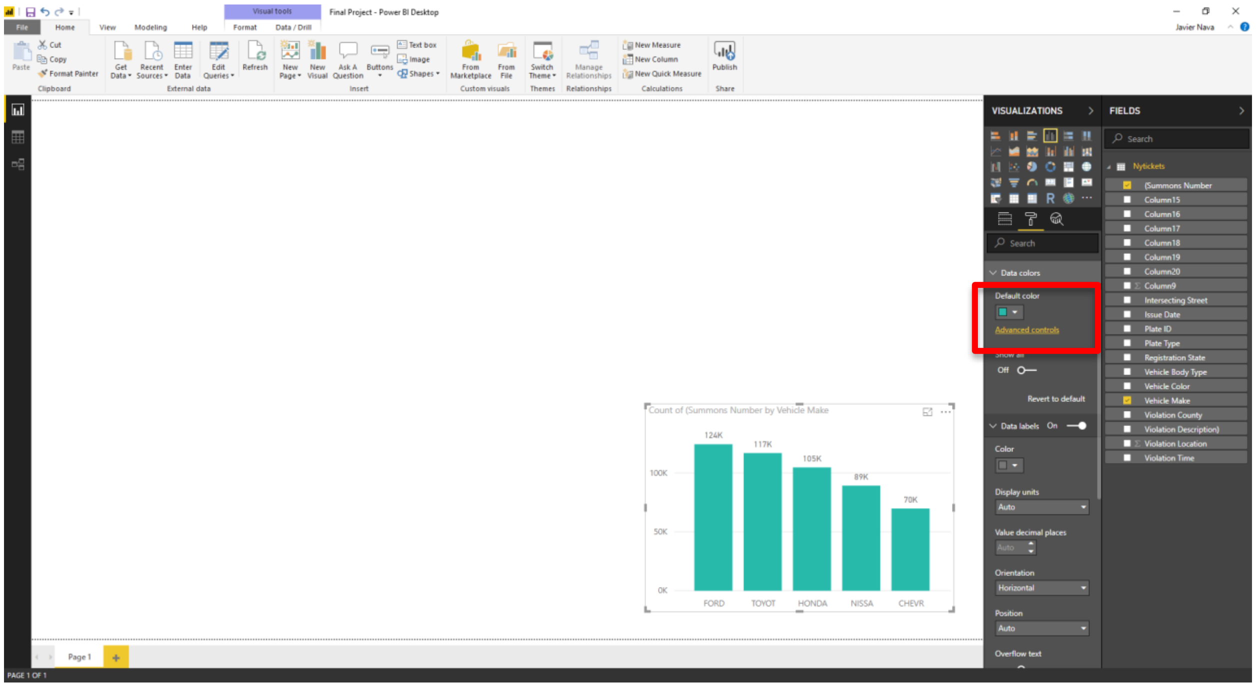

Vehicle Make
top 5 by Count of (S...
Filter type
Top N
Show items:
Top 5
By value
Count of (Summo...

9. After doing so go ahead and turn on **‘data labels’**. You may play around with the data labels to get them the way you like.



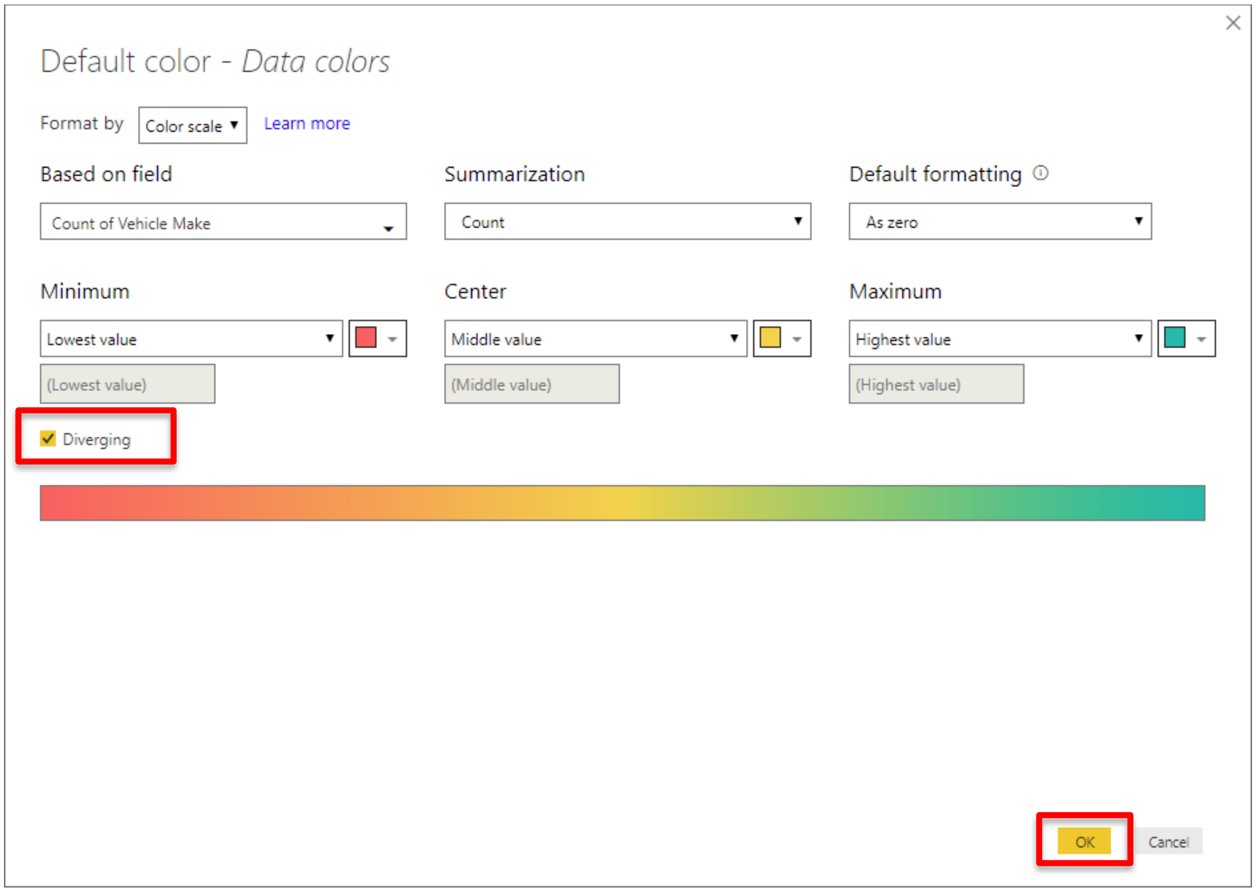
Data colors
Data labels On
Color
Display units: Auto
Value decimal places: Auto
Orientation: Horizontal
Position: Auto
Overflow text
Text size: 9
Font family: DIN
Show background: Off

10. Next, we will change the colors of our columns. Go to the **data colors** tab to change the colors. Click on **default color advanced controls.**



Data colors
Default color
Advanced controls
Show all: Off
Revert to default
Data labels On
Color
Display units: Auto
Value decimal places: Auto
Orientation: Horizontal
Position: Auto
Overflow text

11. That will bring up this pop up. This pop up allows us to color our columns by highest value to shortest value. Click on **diverging** and click ok.



Default color - *Data colors*

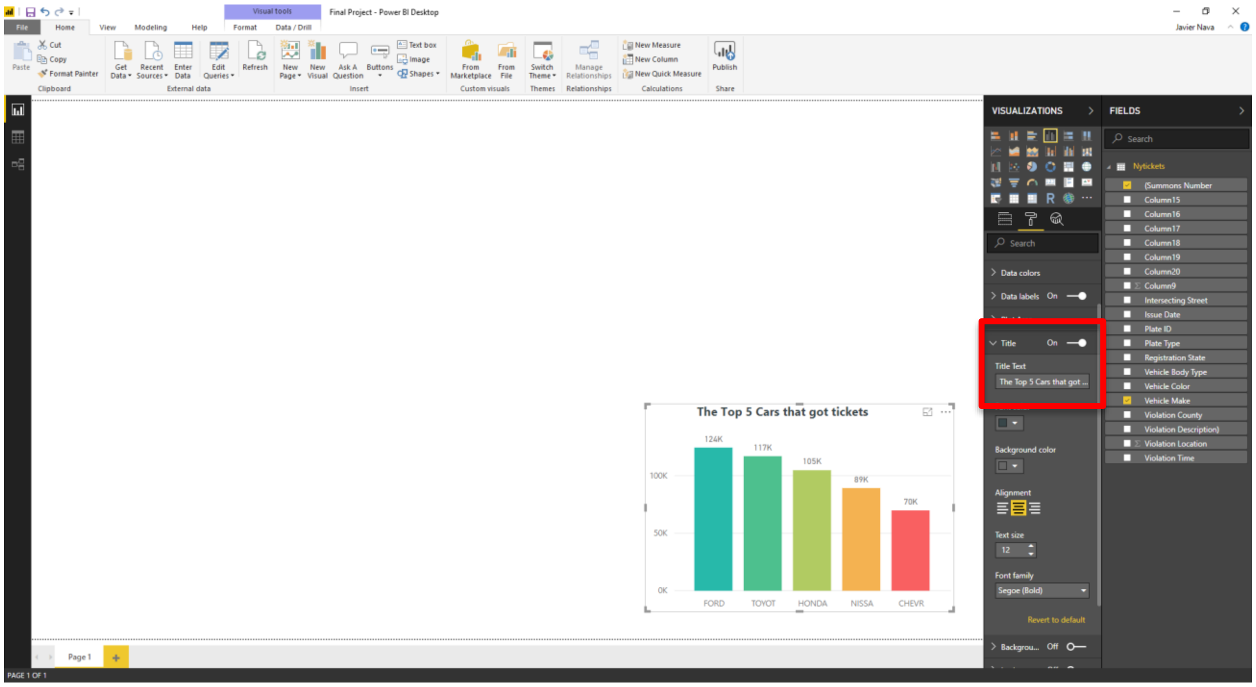
Format by: Color scale — Learn more

Based on field: Count of Vehicle Make
Summarization: Count
Default formatting: As zero

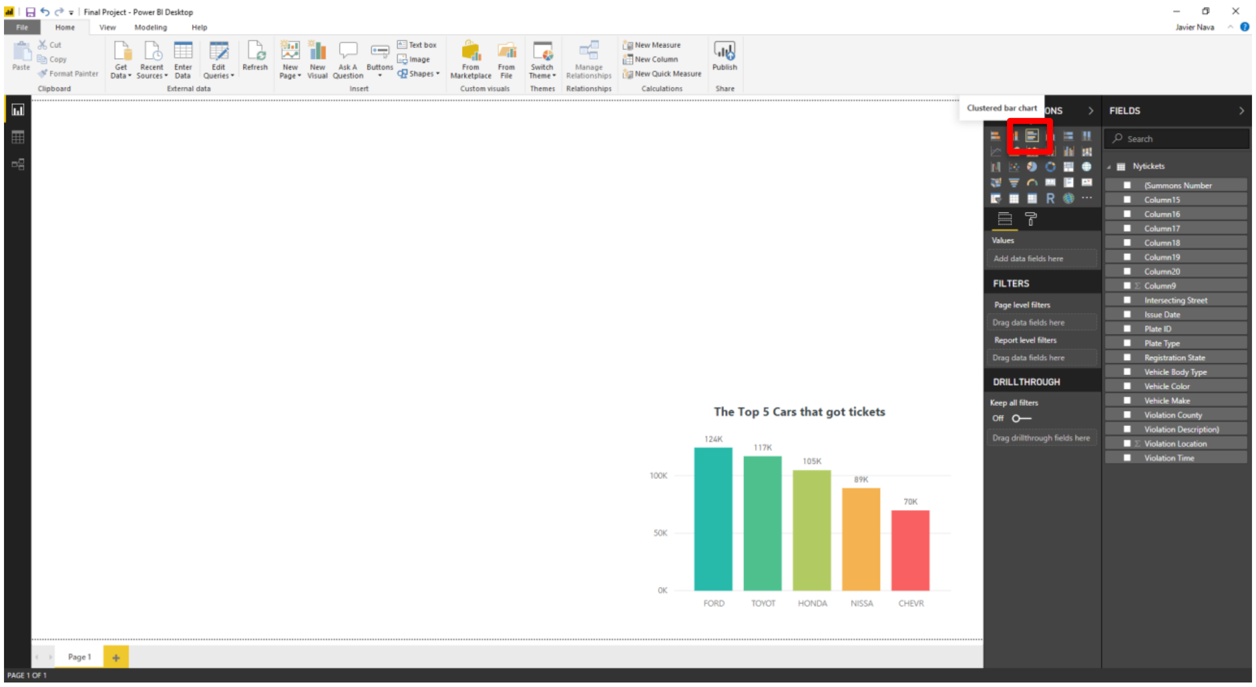
Minimum: Lowest value — (Lowest value)
Center: Middle value — (Middle value)
Maximum: Highest value — (Highest value)

☑ Diverging

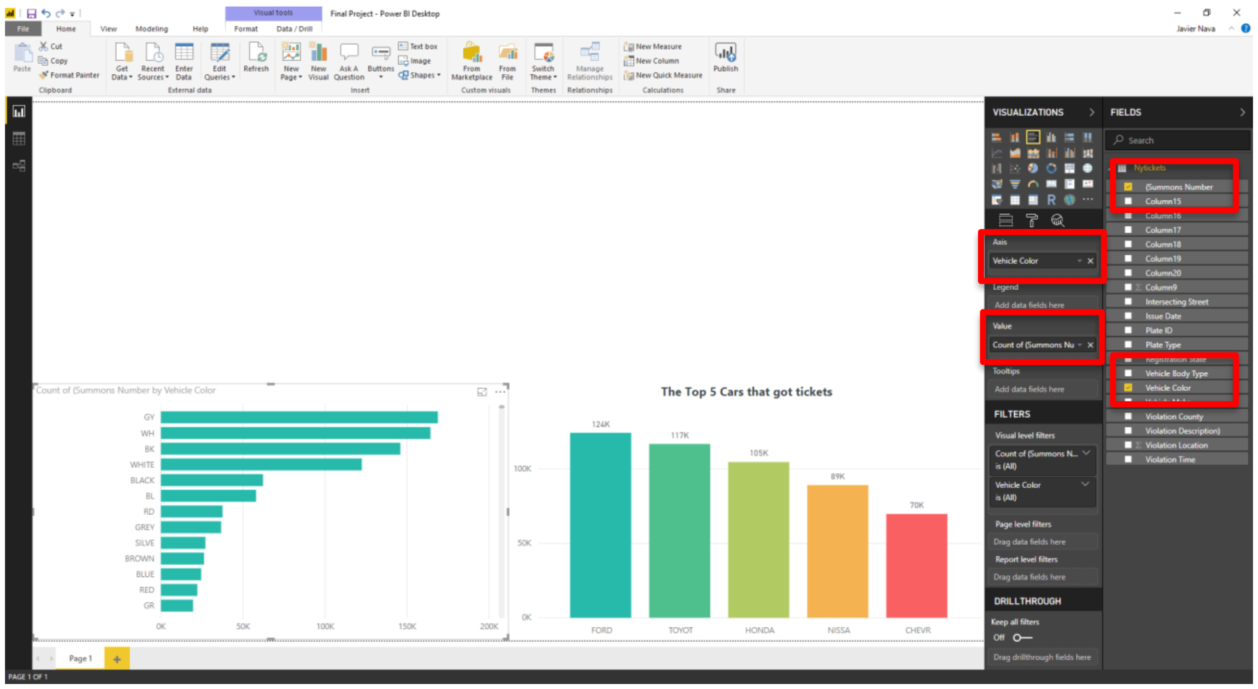
12. As you can see, our colors have changed. Lastly, we will add a title to our chart. To add a **title,** turn the title tab on and **on title text** add your preferred title.



13. Next, we are going to visualize the top 5 car colors attracted the most tickets. To do so we will use a **clustered bar chart**. Click on the **clustered bar chart.**



14. We will now add **(Summons Number** to “Value” and **Vehicle color** to “axis”.

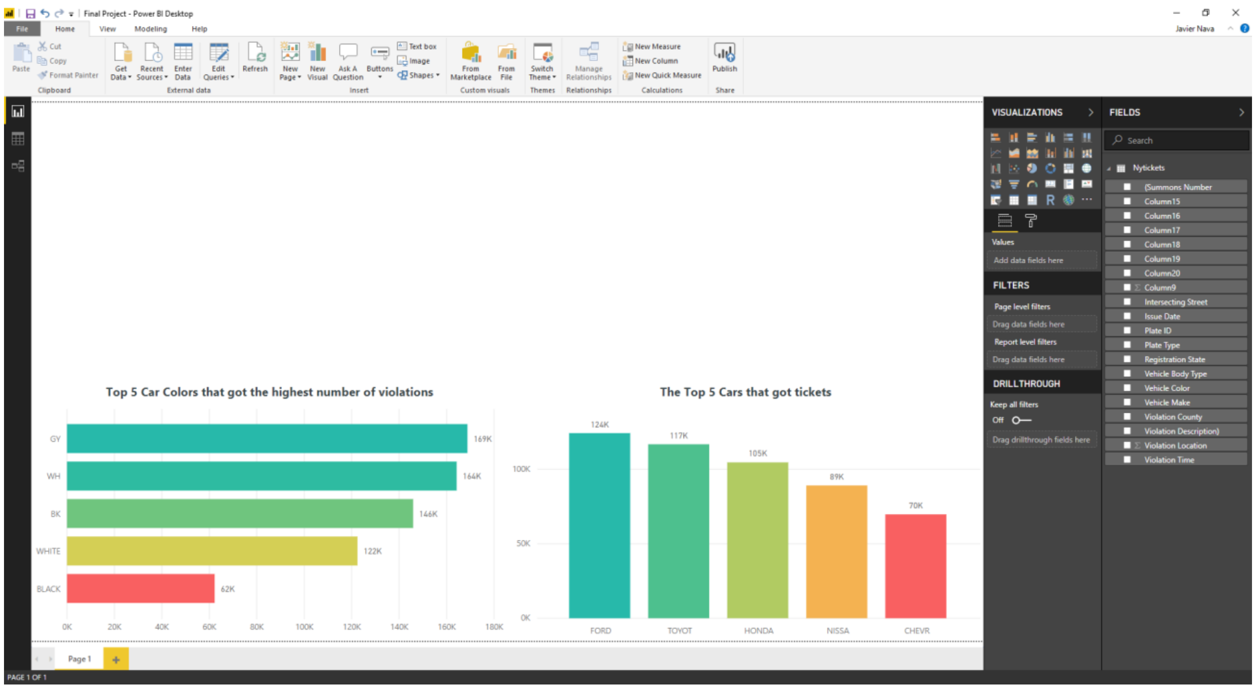

Final Project - Power BI Desktop
Count of (Summons Number by Vehicle Color
The Top 5 Cars that got tickets
Axis
Vehicle Color
Value
Count of (Summons Nu
(Summons Number
Vehicle Color

To limit to top 5 refer to step 8.

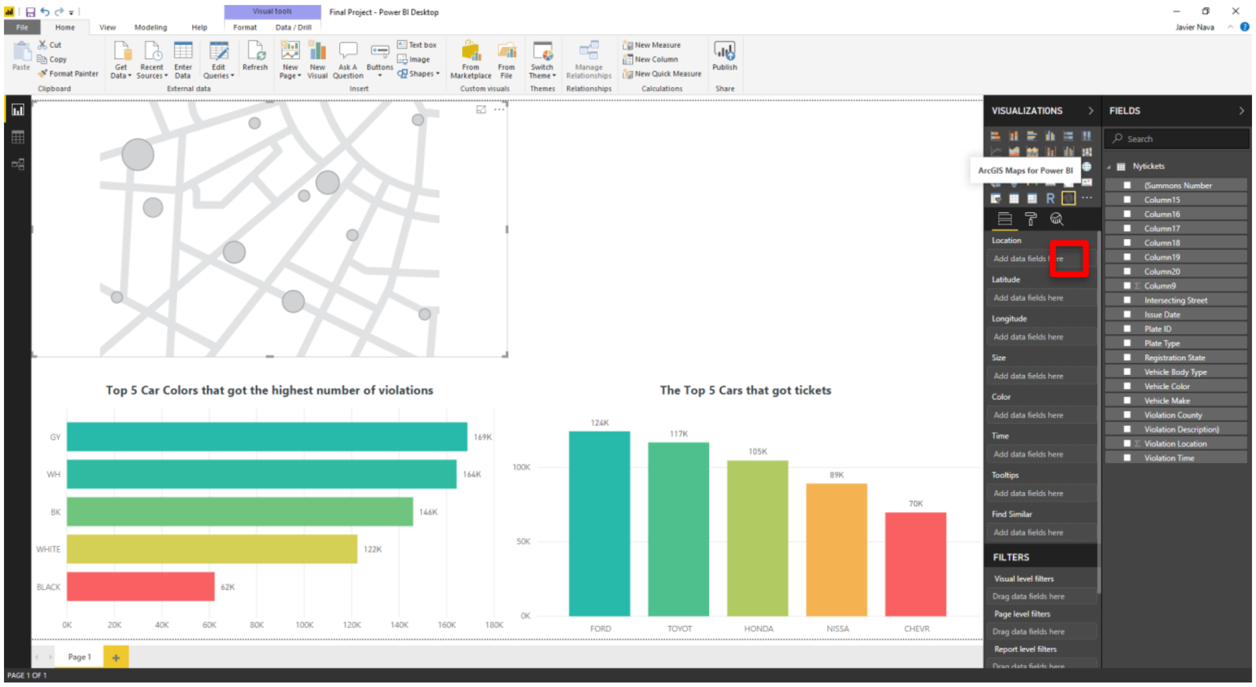
To change color of bars refer to steps 10 and 11.

To add the data labels refer to step 9.

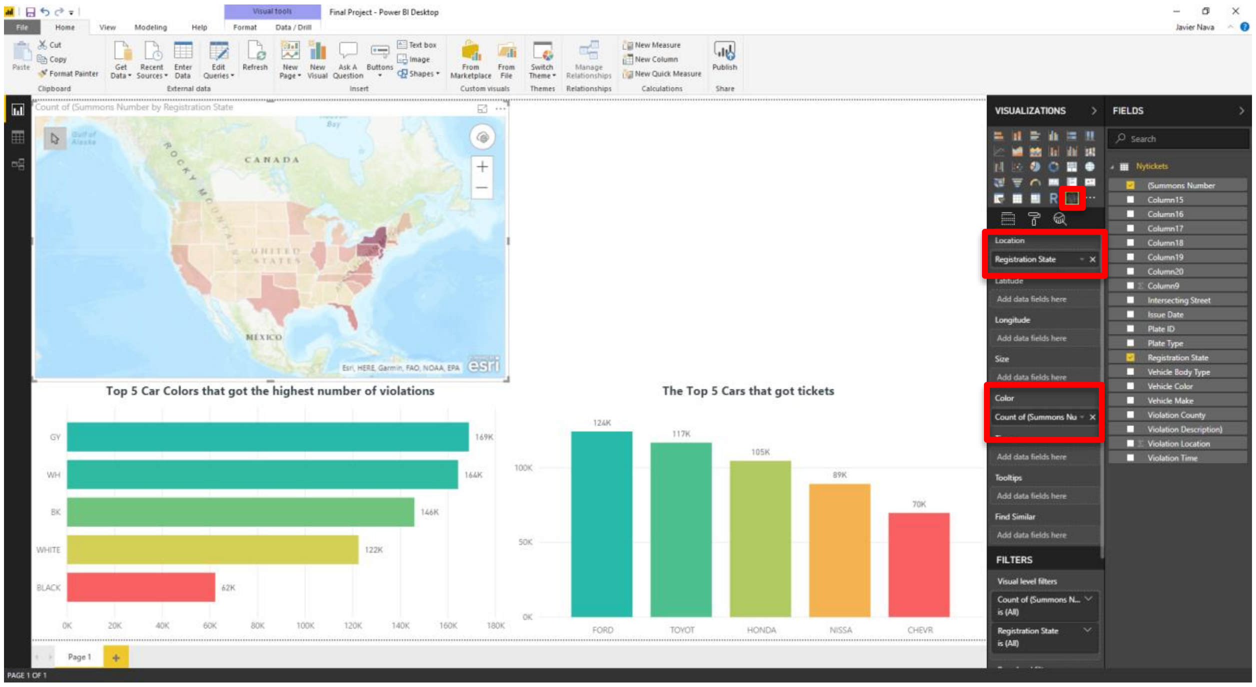
To add the title refer to step 12. (The title should be “Top 5 car colors that got the highest number of violations.



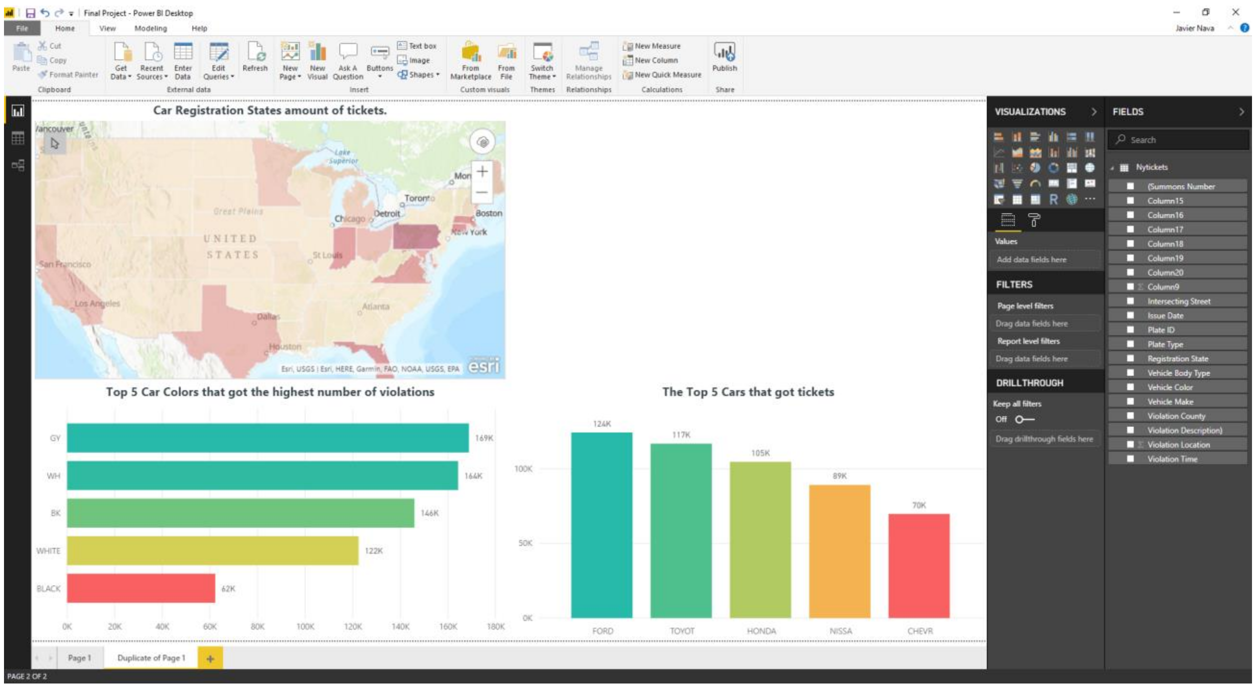
15. We are now going to visualize the states that had the highest number of cars registered that got the highest amount of tickets. First, we are going to use ArcGIS maps.



16. We are now going to drag **(Summons Number** to ‘color’ and **Registration Date** to ‘location’.

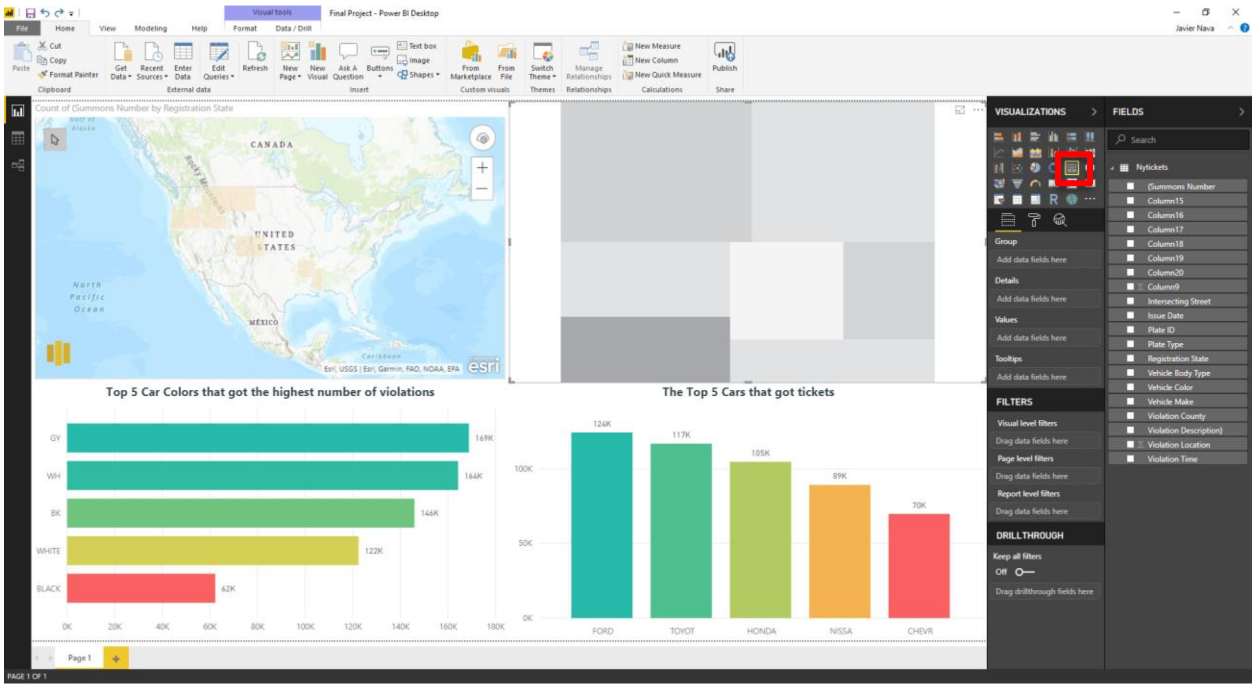

Count of (Summons Number by Registration State
Location
Registration State
Color
Count of (Summons Nu
CANADA
UNITED STATES
MEXICO
Top 5 Car Colors that got the highest number of violations
The Top 5 Cars that got tickets
Count of (Summons N... is (All)
Registration State is (All)

To change the title of our map to ‘Car Registration States amount of tickets” refer to step 12.

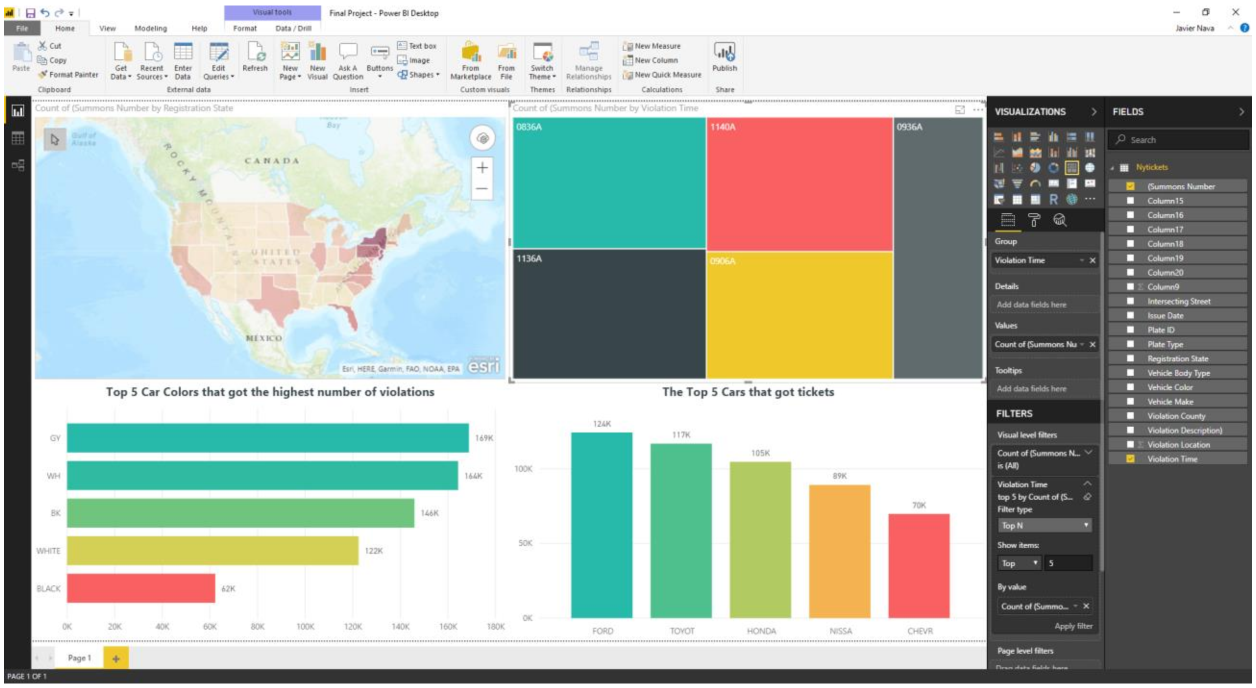

Car Registration States amount of tickets.
Top 5 Car Colors that got the highest number of violations
The Top 5 Cars that got tickets
VISUALIZATIONS
FIELDS

17. Moving forward we are going to visualize the times that had the highest amount of tickets.

Click on the Tree Map option.


Top 5 Car Colors that got the highest number of violations
The Top 5 Cars that got tickets
VISUALIZATIONS
FIELDS

18. Next drag **(Summons Number** to ‘Group’ and drag **Violation time** to ‘Values’.



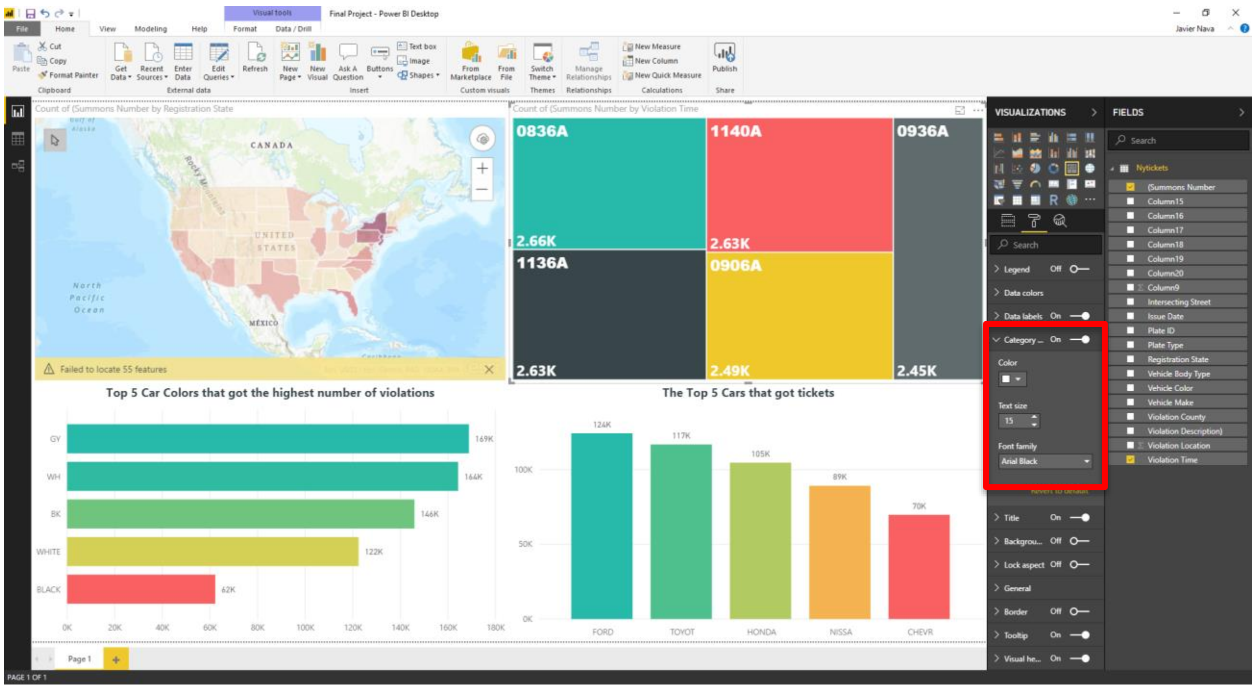
To limit to top 5 refer to step 8.

To change color of bars refer to steps 10 and 11.

To add the data labels refer to step 9.

To change the size of the values, go to category tab and increase the size.

To change the step refer to step 12. (Change title to “Rush Hour” Times)


Final Project - Power BI Desktop

Count of (Summons Number by Registration State

Failed to locate 55 features

Count of (Summons Number by Violation Time

0836A	2.66K
1140A	2.63K
0936A	2.45K
1136A	2.63K
0906A	2.49K

Top 5 Car Colors that got the highest number of violations

GY	169K
WH	164K
BK	146K
WHITE	122K
BLACK	62K

The Top 5 Cars that got tickets

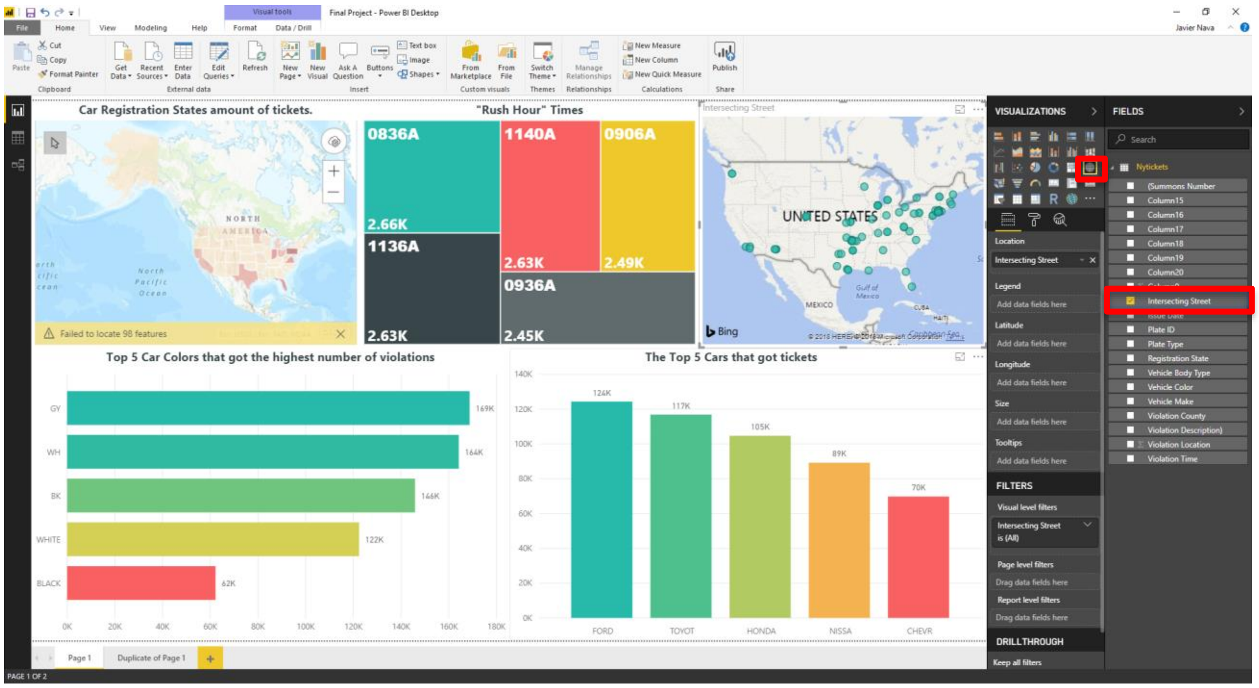
FORD	124K
TOYOT	117K
HONDA	105K
NISSA	89K
CHEVR	70K

Legend Off
Data colors
Data labels On
Category ... On
Color
Text size 15
Font family Arial Black
Title On
Backgrou... Off
Lock aspect Off
General
Border Off
Tooltip On
Visual he... On

FIELDS
Nytickets
- [✓] (Summons Number
- Column15
- Column16
- Column17
- Column18
- Column19
- Column20
- Column9
- Intersecting Street
- Issue Date
- Plate ID
- Plate Type
- Registration State
- Vehicle Body Type
- Vehicle Color
- Vehicle Make
- Violation County
- Violation Description)
- Violation Location
- [✓] Violation Time

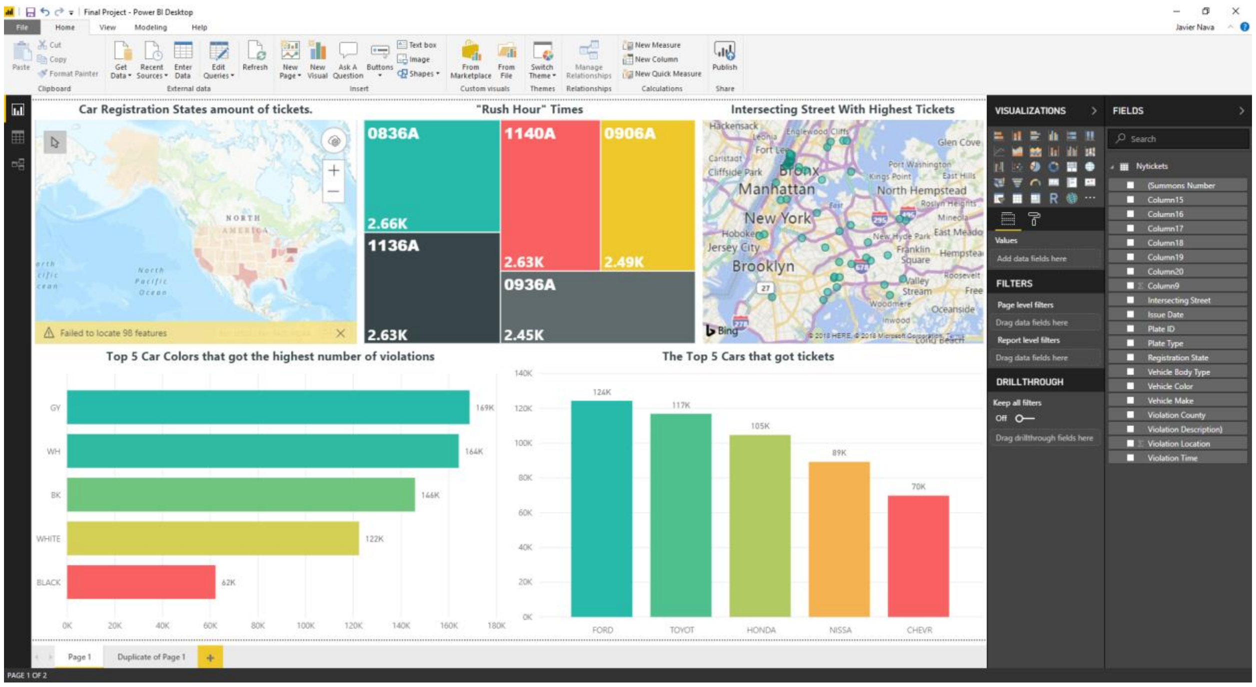
19. Lastly, we are going to visualize which intersections had the most tickets. Note that some of the intersections are getting picked up in different states. We are going to ignore those and only focus on the ones in New York.

Go ahead and click on the map to add. Next drag **intersecting Street** to location.



Final Project - Power BI Desktop
Car Registration States amount of tickets.
"Rush Hour" Times
Top 5 Car Colors that got the highest number of violations
The Top 5 Cars that got tickets
VISUALIZATIONS
FIELDS
Location
Intersecting Street
Intersecting Street is (All)

20. Next we are going to zoom in on the map to only intersections that are in New York. After zooming in we are finished. You have now successfully completed a dashboard.



Conclusion

In this tutorial, we learned how to create an alias using pig, extract useful information from a dataset, and how to export the alias into a CSV file. We then downloaded the new dataset to our local computer then used Power BI to analyze and visualize the data to answer our questions. Through our research and analysis, we discovered many interesting variables pertaining to the highest amount of parking violations in New York. First we started with the brands of cars, in which Ford was the highest. Then we filtered for color, where we discovered white cars had the highest amount of parking violations. For a more in practical analysis, we checked which times had the highest amount of parking tickets, and it turns out to be during the morning, specifically 8:36, 11:36, and 11:40 am. Along with the times we checked for the locations, and it turns out the highest concentration on the street Broadway. By looking at the results from our analysis and findings, we can eliminate or narrow down what Brands or car colors to choose from when purchasing a new vehicle. These analyses also help us locate the locations and time period with highest ticketing activities.

References

1. https://data.ny.gov/api/assets/83055271-29A6-4ED4-9374-E159F30DB5AE
2. https://www.kaggle.com/new-york-city/nyc-parking-tickets#Parking_Violations_Issued_-_Fiscal_Year_2017.csv
3. https://github.com/oscarmnz8/Big-Data-NYC-Parking-Tickets