

# NYC Automated Traffic Volume Counts

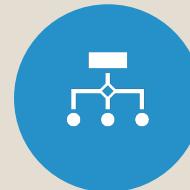
- Pratishtha Soni
- Saurabh Jain

CPSC 531-01  
Advance Database Management  
Instructor- Tseng-Ching James Shen,  
PhD

# Agenda



Why analysis of traffic data?



Dataset



Methodology



Actual Implementation



Visualizing the analysis



Future Scope

# Why Analysis ?

- New York City transportation use automated traffic recorders to collect traffic volume counts at bridge crossings and roadways.
- Traffic data **provides information about how traffic count on specific road segments change over time**. It is important in network analysis because traffic affects travel times, which in turn affect results.

# Applications

- Future Lane Roadway Design Process
- Future Employment in Automobile Industry
- Future Population
- Future Housing

# Dataset

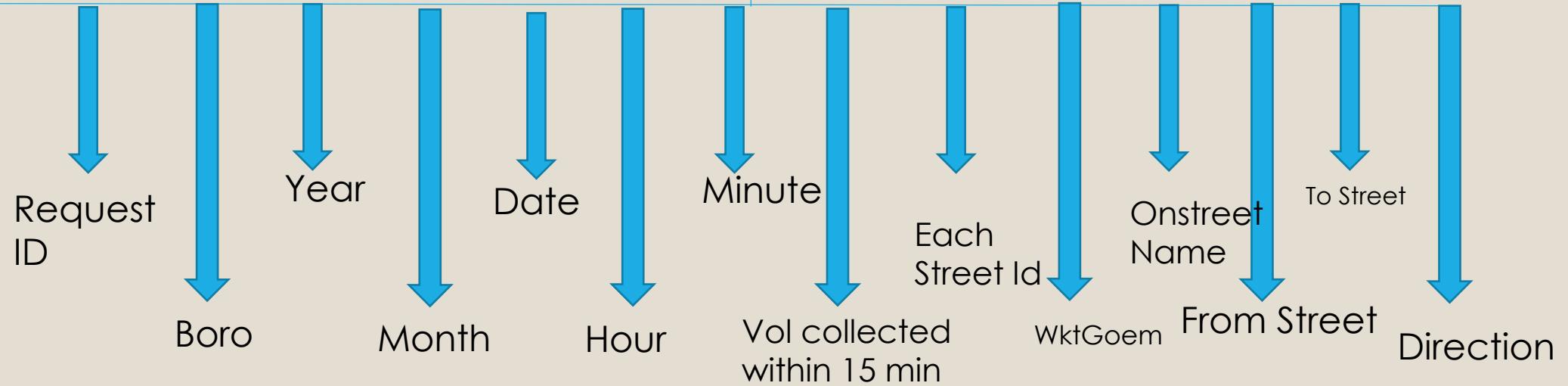
Collected from Kaggle

1. Total 14 columns

2. Total 27190511 rows

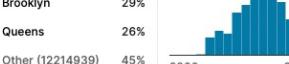
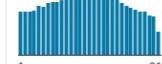
Total Data size is 3.3 GB

## Columns Names



Total 21 columns

# Dataset Snapshot

<code>RequestID</code>	<code>Boro</code>	# Yr	# M	# D	# HH	# MM	# Vol	<code>SegmentID</code>	<code>WktGeom</code>	<code>street</code>	<code>fromSt</code>	<code>toSt</code>	<code>Direction</code>
A unique ID that is generated for each counts request.	Lists which of the five administrative divisions of New York City the location is within, written as a word	The two digit year portion of the date when the count was conducted.	The two digit month portion of the date when the count was conducted.	The two digit day portion of the date when the count was conducted.	The two digit hour portion of the time when the count was conducted.	The two digit start minute portion of the time when the count was conducted.	The total sum of count collected within a 15 minute increments.	The ID that identifies each segment of a street in the LION street network version 14.	A text markup language for representing vector geometry objects on a map and spatial reference systems of spatial	The 'On Street' where the count took place.	The 'From Street' where the count took place.	The 'To Street' where the count took place.	The text-based direction of traffic where the count took place.
 1100 32.4k	Brooklyn 29% Queens 26% Other (12214939) 45%								20463 unique values	BROADWAY 2% QUEENS BOULEVARD 1% Other (26427057) 97%	Dead End 8% Dead end 2% Other (24552570) 90%	Dead end 8% 8 Avenue Line 1% Other (24776351) 91%	NB 26% SB 26% Other (13114795) 48%
20856	Queens	2015	6	23	23	30	9	171896	POINT (1052296.600156678 199785.26932711253)	94 AVENUE	207 Street	Francis Lewis Boulevard	WB
21231	Staten Island	2015	9	14	4	15	6	9896	POINT (942668.0589589147 171441.21296926)	RICHMOND TERRACE	Wright Avenue	Emeric Court	WB
29279	Bronx	2017	10	19	4	30	85	77817	POINT (1016508.0034850211 235221.59092266942)	HUNTS POINT AVENUE	Whittier Street	Randall Avenue	NB
27019	Brooklyn	2017	11	7	18	30	168	188823	POINT (992925.4316854962 184116.82855457635)	FLATBUSH AVENUE	Brighton Line	Brighton Line	NB
26734	Manhattan	2017	11	3	22	8	355	137516	POINT (1004175.9505178436 247779.63624949602)	WASHINGTON BRIDGE	Harlem River Shoreline	Harlem River Shoreline	EB

# Tools

- Java Spark
- Google Collab Notebook
- Google Cloud Platform
- Google Data Proc
- Python Matplotlib

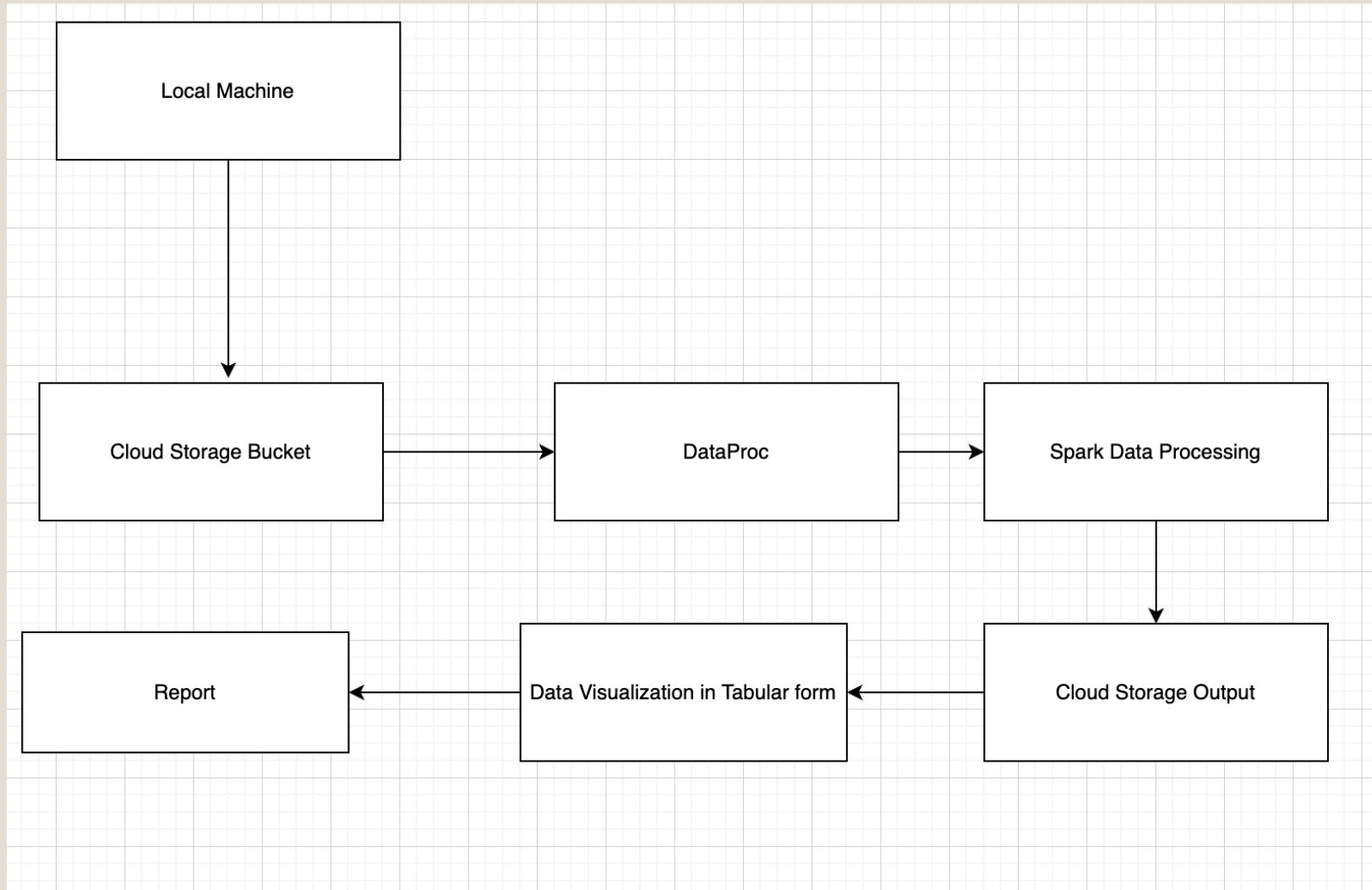




# Google DataProc

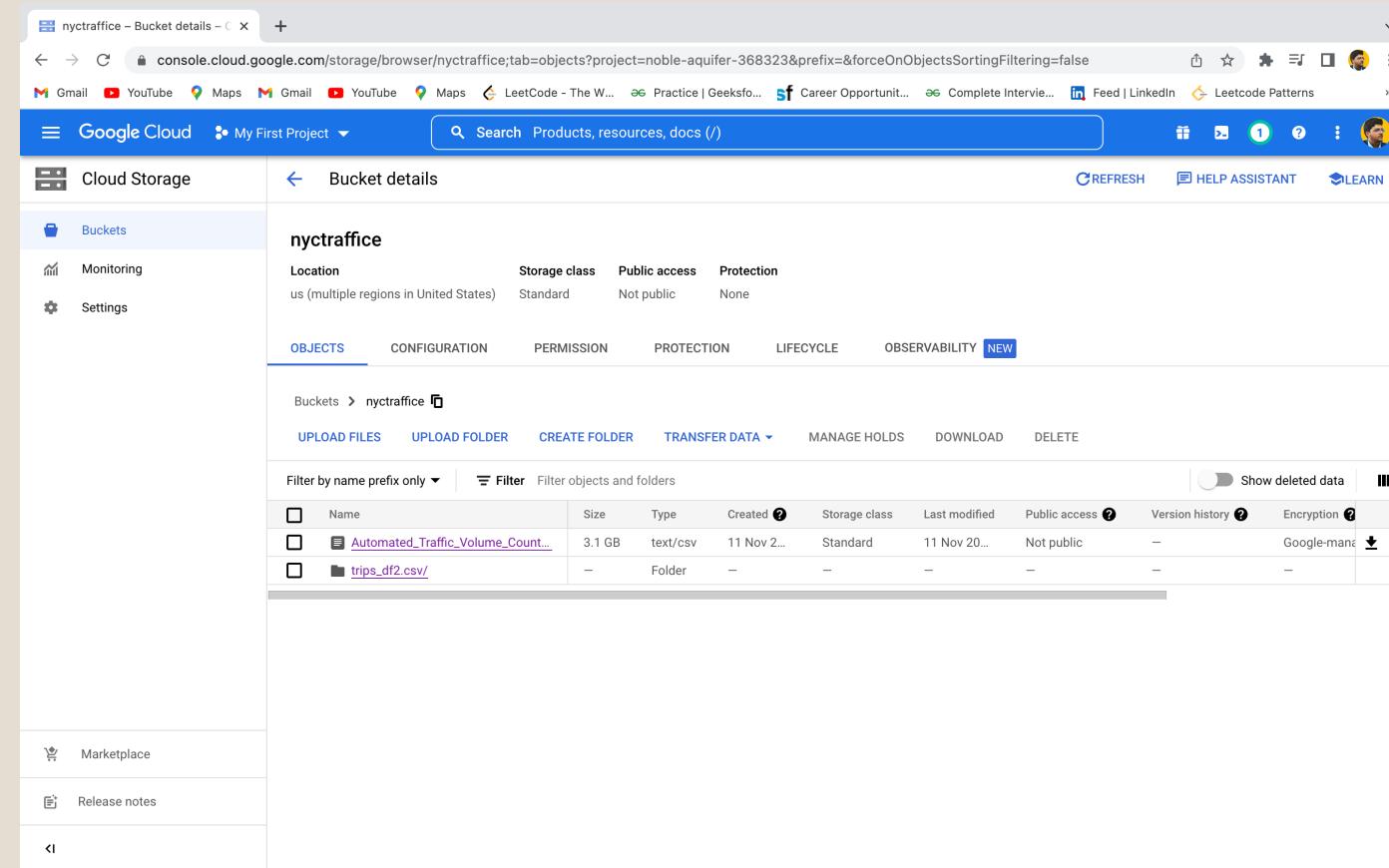
- Cost Effective
- Supports different big data tools. Ex:- Spark, Hive etc.
- Also supports Python, Scala, R, Java and lambda function ( $\lambda$ )
- Scales resources according to business needs.
- Pre-installed software

# Methodology



# Actual Implementation

- Create Buckets and Upload Data



# Actual Implementation

- Create Clusters

The image shows two side-by-side screenshots of the Google Cloud DataProc interface.

**Left Screenshot (Cluster Details):**

- Header:** Google Cloud, My First Project, Search: dataproc
- Left Sidebar:** Dataproc, Clusters, Jobs, Workflows, Auto-scaling policies, Serverless, Batches, Metastore services, Metastore, Federation, Utilities, Component exchange, Workbench, Release notes.
- Center Content:**
  - Jobs on clusters:** Cluster details for "cluster-7886".
    - Name: cluster-7886
    - Cluster UUID: 31e2bca3-54b6-4340-8255-d45a67a68bb6
    - Type: Dataproc cluster
    - Status: Running
  - VM Instances:** Shows three instances: cluster-7886-m (Master), cluster-7886-w-0, and cluster-7886-w-1 (Worker). Role: Master, Worker, Worker. SSH access is available.
  - EQUIVALENT REST:** A section showing equivalent REST API endpoints.
- Bottom:** Message: Successfully started cluster-7886.

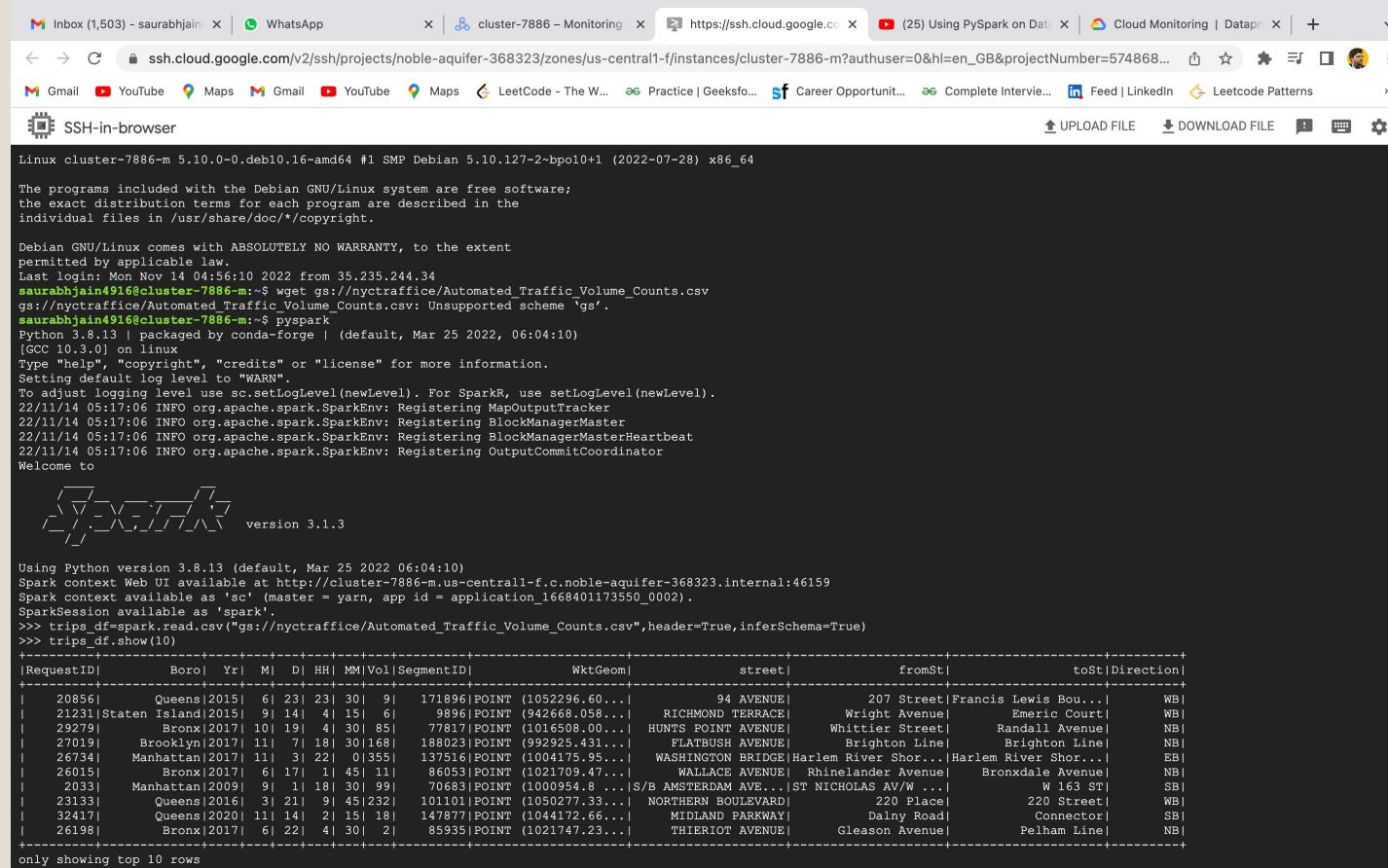
**Right Screenshot (Configuration):**

- Header:** Google Cloud, My First Project, Search: dataproc
- Left Sidebar:** Dataproc, Clusters, Jobs, Workflows, Auto-scaling policies, Serverless, Batches, Metastore services, Metastore, Federation, Utilities, Component exchange, Workbench, Release notes.
- Center Content:** Detailed configuration settings for the cluster.

Setting	Value
Region	us-central1
Zone	us-central1-f
Auto-scaling	Off
DataProc Metastore	None
Scheduled deletion	Off
Master node	Standard (1 master, N workers)
Machine type	n1-standard-4
Number of GPUs	0
Primary disk type	pd-standard
Primary disk size	50GB
Local SSDs	0
Worker nodes	2
Machine type	n1-standard-2
Number of GPUs	0
Primary disk type	pd-standard
Primary disk size	50GB
Local SSDs	0
Secondary worker nodes	0
Secure Boot	Disabled
VTPEM	Disabled
Integrity Monitoring	Disabled
Cloud Storage staging bucket	dataproc-staging-us-central1-57486208620-icvp9ar
Network	default
Network tags	None
Internal IP only	No
Image version	2.0.50-debian10
Created	10 Nov 2022, 16:06:16
Properties	Show more properties

# Actual Implementation

- Run Spark command in SSH Consol.



The screenshot shows a web browser window with multiple tabs open. The active tab is titled "cluster-7886 - Monitoring" and displays a terminal session titled "SSH-in-browser". The terminal output is as follows:

```
Linux cluster-7886-m 5.10.0-0.deb10.16-amd64 #1 SMP Debian 5.10.127-2-bpo10+1 (2022-07-28) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov 14 04:56:10 2022 from 35.235.244.34
saurabhjain@9156@cluster-7886-m:~$ wget gs://nyctraffice/Automated_Traffic_Volume_Counts.csv
gs://nyctraffice/Automated_Traffic_Volume_Counts.csv: Unsupported scheme `gs'.
saurabhjain@9156@cluster-7886-m:~$ pyspark
Python 3.8.13 | packaged by conda-forge | (default, Mar 25 2022, 06:04:10)
[GCC 10.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/14 05:17:06 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
22/11/14 05:17:06 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
22/11/14 05:17:06 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
22/11/14 05:17:06 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Welcome to
    / \   _ \ / \   _ \ / \
   / \ / \ / \ / \ / \ / \ version 3.1.8
   / \ / \ / \ / \ / \ / \
   / \ / \ / \ / \ / \ / \
   / \ / \ / \ / \ / \ / \
Using Python version 3.8.13 (default, Mar 25 2022 06:04:10)
Spark context Web UI available at http://cluster-7886-m.us-central1-f.c.noble-aquifer-368323.internal:46159
Spark context available as 'sc' (master = yarn, app id = application_1668401173550_0002).
SparkSession available as 'spark'.
>>> trips_df=spark.read.csv("gs://nyctraffice/Automated_Traffic_Volume_Counts.csv",header=True,inferSchema=True)
>>> trips_df.show(10)
+-----+-----+-----+-----+-----+-----+
|RequestID|Boro|Yr|M|D|HH|MM|Vol|SegmentID|WktGeom|street|fromSt|toSt|Direction|
+-----+-----+-----+-----+-----+-----+
| 20856|Queens|2015|6|23|23|30|9|171896|POINT (1052296.60...|94 AVENUE|207 Street|Francis Lewis Bou...|WB|
| 21231|Staten Island|2015|9|14|4|15|6|9896|POINT (942668.058...|RICHMOND TERRACE|Wright Avenue|Emeric Court|WB|
| 29279|Bronx|2017|10|19|4|30|85|77817|POINT (1016500.0...|HUNTS POINT AVENUE|Whittier Street|Randall Avenue|NB|
| 27019|Brooklyn|2017|11|7|18|30|168|188023|POINT (992925.431...|FLATBUSH AVENUE|Brighton Line|Brighton Line|NB|
| 26734|Manhattan|2017|11|3|22|0|355|137516|POINT (1004175.95...|WASHINGTON BRIDGE|Harlem River Shor...|Harlem River Shor...|EB|
| 26015|Bronx|2017|6|17|1|45|11|86052|POINT (1021709.47...|WALLACE AVENUE|Rhinelander Avenue|Bronxdale Avenue|NB|
| 2033|Manhattan|2009|9|1|18|30|99|70683|POINT (1000954.8...|S/B AMSTERDAM AVE...|ST NICHOLAS AV/W ...|W 163 ST|SB|
| 23133|Queens|2016|3|21|9|45|232|101101|POINT (1050277.33...|NORTHERN BOULEVARD|220 Place|220 Street|WB|
| 32417|Queens|2020|11|14|2|15|18|147877|POINT (1044172.66...|MIDLAND PARKWAY|Daly Road|Connector|SB|
| 26198|Bronx|2017|6|22|4|30|2|85935|POINT (1021747.23...|THIERIOT AVENUE|Gleason Avenue|Pelham Line|NB|
+-----+
only showing top 10 rows
```

# Actual Implementation

- ## ◦ Spark SQL Commands .

```
>>> new_results = spark.sql("select Boro,vol/15 as volper from Traffic").show()  
+-----+  
| Boro | volper |  
+-----+  
| Queens| 4.06666666666666 |  
| Staten Island| 3.86666666666667 |  
| Queens| 3.33333333333335 |  
| Brooklyn| 10.8 |  
| Queens| 5.6 |  
| Bronx| 0.0 |  
| Manhattan| 11.5333333333333 |  
| Queens| 0.666666666666666 |  
| Bronx| 1.2 |  
| Queens| 5.86666666666666 |  
| Manhattan| 7.2 |  
| Queens| 1.33333333333333 |  
| Manhattan| 0.86666666666667 |  
| Manhattan| 2.6 |  
| Manhattan| 1.26666666666666 |  
| Brooklyn| 4.1333333333334 |  
| Manhattan| 12.0666666666666 |  
| Manhattan| 0.93333333333333 |  
| Queens| 0.666666666666666 |  
| Manhattan| 6.0 |  
+-----+  
only showing top 20 rows
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2276 | Manhattan|2000| 1| 5| 7| 30| 216| 32956|POINT (986884.7 2...|W/B UNION SQUARE ...| E 14 ST/BROADWAY| 4 AV| WB\|
| 7625 | Queens|2007| 4| 5| 23| 0| 148| 156596|POINT (1023975.5 ...| LINDEN BLVD| 78 ST| AMBER ST| WB\|
| 7627 | Brooklyn|2007| 4| 24| 5| 0| 972| 141367|POINT (1013833.7 ...| JACKIE ROBINSON PY| MILLER AV| VERNONT PL| EB\|
| 7618 | Brooklyn|2007| 4| 12| 22| 45| 11| 45393|POINT (1010051.3 ...| CORNELIA ST| WYCKOFF AV| CYPRESS AV| NB\|
| 7624 | Queens|2007| 4| 18| 4| 0| 4| 190387|POINT (1023782.6 ...| LINDEN BLVD| 78 ST| AMBER ST| WB\|
| 3442 | Bronx|2008| 4| 13| 8| 0| 164| 88874|POINT (1020685.6 ...|S/B VAN CORTLANDT...| E 242 ST/KATONAH AVI KATONAH AVENUE| SB\|
| 3473 | Queens|2008| 6| 5| 11| 0| 3291| 138592|POINT (1060197.4 ...|E/B LAURELTON @ C...| DEAD END|BELT PKWY EB EXIT...| EB\|
| 7489 | Queens|2008| 5| 26| 15| 45| 759| 153405|POINT (1004881.5 ...|E/B KOSCIUSZKO BR...| 54 RD| BQE EB EXIT 35 E-W| EB\|
| 7499 | Queens|2008| 3| 17| 10| 45| 119| 156954|POINT (1066765.1 ...|E/B UNION TURNPIK...| LANGDALE ST| LANGDALE ST| EB\|
| 3479 | Manhattan|2008| 8| 17| 22| 15| 320| 36414|POINT (994948.7 2...|S/B 2 AVENUE @ EA...| E 65 ST| E 66 ST| SB\|
| 7521 | Queens|2008| 3| 17| 11| 30| 128| 156095|POINT (1059170.9 ...|E/B MERRICK BLVD@...| 244 ST| 243 ST| EB\|
| 7463 | Bronx|2008| 5| 13| 8| 45| 694| 190807|POINT (1031782.6 ...|N/B HUTCHINSON RI...| NEW ENGLAND THRwy| NEW ENGLAND THRwy| NB\|
| 7464 | Bronx|2008| 5| 18| 17| 30| 0| 154955|POINT (1033736.9 ...|N/B HUTCHINSON RI...|HUTCHINSON RIVER ...|HUTCHINSON RIVER ...| NB\|
| 3475 | Manhattan|2008| 7| 23| 14| 15| 91| 23255|POINT (981449.3 2...|E/B CHAMBERS ST @...|HUDSON ST/W BROADWAY| GREENWICH ST| EB\|
| 7600 | Bronx|2008| 5| 5| 11| 0| 2069| 135125|POINT (1015681.1 ...|S/B HENRY HUDSON ...| DEAD END| MOSHOLU PKWY| SB\|
| 3434 | Bronx|2008| 6| 23| 22| 0| 2013| 95879|POINT (1038049.4 ...|S/B NEW ENGLAND T...| CONNECTOR| DEAD END| SB\|
| 3450 | Brooklyn|2008| 5| 10| 15| 30| 106| 105839|POINT (987431.6 1...|N/B 3 AVE BRIDGE ...| 3 ST| 6 ST| NB\|
| 7464 | Bronx|2008| 5| 7| 16| 30|1006| 154955|POINT (1033736.9 ...|N/B HUTCHINSON RI...|HUTCHINSON RIVER ...|HUTCHINSON RIVER ...| NB\|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 20 rows

```

```

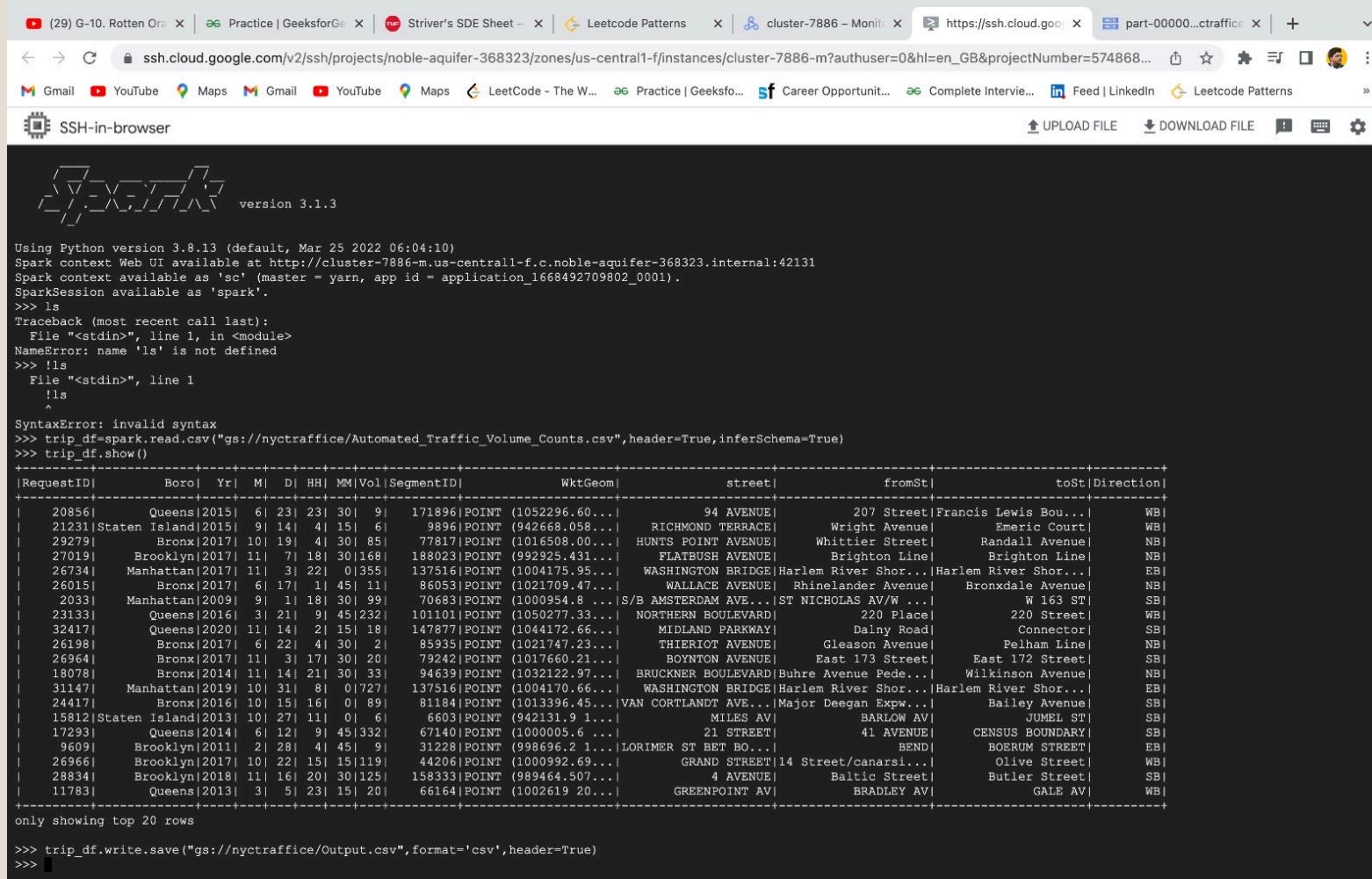
>>> new_results = spark.sql("select * from Traffic order by Yr DESC").show()
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| RequestID| Boro| Yr| M| D| HH| MM|Vol|SegmentID| WktGeom| Street| fromSt| toSt| Direction|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|\f0\f24 \cf0 Req...| Boro| Yr| M| D| HH| MM|Vol|SegmentID| WktGeom| street| fromSt| toSt|Direction\|
| 32384|Manhattan|2020| 10| 20| 11| 30| 81| 158808|POINT (989240.431...| BROADWAY| West 60 Street| West 61 Street| NB\|
| 32407 | Bronx|2020| 10| 19| 11| 0|140| 69836|POINT (1005248.95...| EAST 149 STREET| Park Avenue| Morris Avenue| WB\|
| 32384|Manhattan|2020| 11| 1| 10| 0| 42| 34257|POINT (989423.209...| CENTRAL PARK WEST| 8 Avenue Line| 8 Avenue Line| NB\|
| 32417 | Queens|2020| 10| 16| 16| 15|152| 191270|POINT (1010338.18...| QUEENS BOULEVARD| Dead End| Dead end| WB\|
| 32407 | Bronx|2020| 10| 13| 3| 30| 10| 184562|POINT (1008564.77...| GRAND CONCOURSE| Concourse Line| Concourse Line| SB\|
| 32417 | Queens|2020| 11| 16| 8| 45| 99| 156873|POINT (1041559.46...| UTOPIA PARKWAY| 65 Avenue| 67 Avenue| SB\|
| 32417 | Queens|2020| 10| 13| 15| 30|194| 150549|POINT (1031410.03...|CROSS BAY BOULEVARD| Dead End| Connector| NB\|
| 32384|Manhattan|2020| 10| 28| 11| 30|195| 34338|POINT (987878.166...| AMSTERDAM AVENUE| West 60 Street| West 61 Street| NB\|
| 32407 | Bronx|2020| 10| 13| 8| 0|220| 78877|POINT (1015489.01...| WESTCHESTER AVENUE|East 167 Street| Home Street| WB\|
| 32417 | Queens|2020| 11| 18| 18| 15|167| 150346|POINT (1044907.43...| MERRICK BOULEVARD| 111 Avenue| Dead end| SB\|
| 31960 | Queens|2020| 1| 26| 0| 0| 0| 171277|POINT (1054176.62...| BEACH 14 STREET| Heyson Road| Seagirt Boulevard| SB\|
| 32406 | Queens|2020| 9| 8| 11| 45| 53| 91184|POINT (1031689.11...| KISSENA BOULEVARD| 41 Avenue| Barclay Avenue| EB\|
| 32417 | Queens|2020| 11| 22| 19| 15| 61| 150039|POINT (1038323.08...| 164 STREET| 65 Avenue| 67 Avenue| SB\|
| 32417 | Queens|2020| 10| 17| 22| 30| 75| 57192|POINT (1038292.60...| LIBERTY AVENUE| Waltham Street| 146 Street| EB\|
| 32407 | Bronx|2020| 10| 18| 4| 45| 34| 72026|POINT (1007749.58...| JEROME AVENUE| Goble Place| East Mt Eden Avenue| NB\|
| 32417 | Queens|2020| 10| 18| 11| 0| 57| 146374|POINT (1066838.41...| UNION TURNPIKE| Hewlett Street| Hewlett Street| WB\|
| 32407 | Bronx|2020| 10| 20| 4| 45| 22| 70376|POINT (1008111.29...| 3 AVENUE|East 154 Street| East 155 Street| NB\|
| 32384|Manhattan|2020| 10| 27| 2| 0| 27| 173249|POINT (1002839.60...| 145 STREET BRIDGE| Dead End| Harlem River Shor...| EB\|
| 32407 | Bronx|2020| 10| 15| 13| 0| 75| 121069|POINT (1013077.48...| CROTONA AVENUE| Dead End| Crotona Park North| SB\|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 20 rows

```

&gt;&gt;&gt;

# Actual Implementation

- Save our Output in Cloud Bucket



The screenshot shows an SSH-in-browser session running on a Google Cloud terminal. The title bar indicates the session is connected to a cluster-7886 instance. The terminal window displays Python code for reading a CSV file and writing it back to Google Storage.

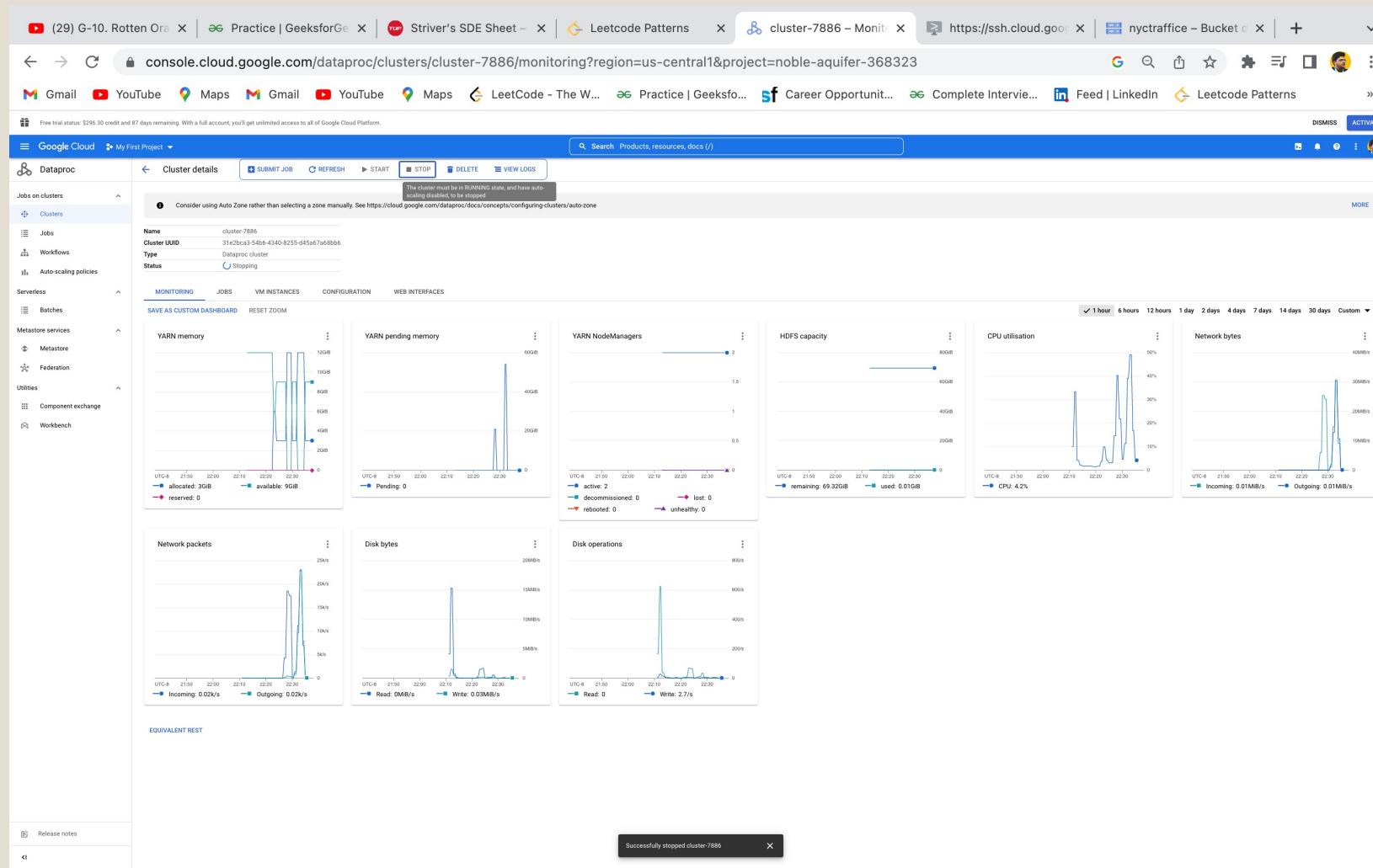
```
Using Python version 3.8.13 (default, Mar 25 2022 06:04:10)
Spark context Web UI available at http://cluster-7886-m.us-central1-f.c.noble-aquifer-368323.internal:42131
Spark context available as 'sc' (master = yarn, app id = application_1668492709802_0001).
SparkSession available as 'spark'.
>>> ls
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'ls' is not defined
>>> !ls
File "<stdin>", line 1
!ls
^
SyntaxError: invalid syntax
>>> trip_df=spark.read.csv("gs://nyctraffice/Automated_Traffic_Volume_Counts.csv",header=True,inferSchema=True)
>>> trip_df.show()
+-----+-----+-----+-----+-----+-----+
|RequestID|Boro|Yr|M|D|HH|MM|Vol|SegmentID|WktGeom|street|fromSt|toSt|Direction|
+-----+-----+-----+-----+-----+-----+
| 20856|Queens|2015|6|23|23|30|9| 171896|POINT (1052296.60...)| 94 AVENUE| 207 Street|Francis Lewis Bou...| WB|  
| 21231|Staten Island|2015|9|14|4|15|6| 98961|POINT (942668.058...)| RICHMOND TERRACE| Wright Avenue| Emeric Court| WB|  
| 29279|Bronx|2017|10|19|4|30|85| 77817|POINT (1016508.00...)| HUNTS POINT AVENUE| Whittier Street| Randall Avenue| NB|  
| 27019|Brooklyn|2017|11|7|18|30|168| 188023|POINT (992925.431...)| FLATBUSH AVENUE| Brighton Line| Brighton Line| NB|  
| 26734|Manhattan|2017|11|31|22|0|355| 137516|POINT (1004175.95...)| WASHINGTON BRIDGE|Harlem River Shor...|Harlem River Shor...| EB|  
| 26015|Bronx|2017|6|17|1|45|11| 86053|POINT (1021709.47...)| WALLACE AVENUE| Rhinelander Avenue| Bronxdale Avenue| NB|  
| 2033|Manhattan|2009|9|1|18|30|99| 70683|POINT (1000954.8...)|S/B AMSTERDAM AVE...|ST NICHOLAS AV/W ...| W 163 ST| SB|  
| 23133|Queens|2016|3|21|9|45|232| 101101|POINT (1050277.33...)| NORTHERN BOULEVARD| 220 Place| 220 Street| WB|  
| 32417|Queens|2020|11|14|2|15|18| 147877|POINT (1044172.66...)| MIDLAND PARKWAY| Dalny Road| Connector| SB|  
| 26198|Bronx|2017|6|22|4|30|2| 85935|POINT (1021747.23...)| THIEROT AVENUE| Gleason Avenue| Pelham Line| NB|  
| 26964|Bronx|2017|11|3|17|30|20| 79242|POINT (1017660.21...)| BOYNTON AVENUE| East 173 Street| East 172 Street| SB|  
| 18078|Bronx|2014|11|14|21|30|33| 94639|POINT (1032122.97...)| BRUCKNER BOULEVARD|Buhré Avenue Pede...| Wilkinson Avenue| NB|  
| 31147|Manhattan|2019|10|31|8|0|127| 137516|POINT (1004170.66...)| WASHINGTON BRIDGE|Harlem River Shor...|Harlem River Shor...| EB|  
| 24417|Bronx|2016|10|15|16|0|89| 81184|POINT (1013396.45...)|VAN CORTLANDT AVE...|Major Deegan Expw...| Bailey Avenue| SB|  
| 15812|Staten Island|2013|10|27|11|0|6| 6603|POINT (942131.9 1...)| MILES AV| BARLOW AV| JUMEL ST| SB|  
| 17293|Queens|2014|6|12|9|45|332| 67140|POINT (1000005.6 ...)| 21 STREET| 41 AVENUE| CENSUS BOUNDARY| SB|  
| 9609|Brooklyn|2011|2|28|4|45|9| 31228|POINT (998696.2 1...)|LORIMER ST BET BO...| BEND| BOERUM STREET| EB|  
| 26966|Brooklyn|2017|10|22|15|15|119| 44206|POINT (1000992.69...)| GRAND STREET|14 Street/canarsi...| Olive Street| WB|  
| 28834|Brooklyn|2018|11|16|20|30|125| 158333|POINT (989464.507...)| 4 AVENUE| Baltic Street| Butler Street| SB|  
| 11783|Queens|2013|3|5|23|15|20| 66164|POINT (1002619.20...)| GREENPOINT AV| BRADLEY AV| GALE AV| WB|  
+-----+-----+-----+-----+-----+-----+  
only showing top 20 rows  
  
>>> trip_df.write.save("gs://nyctraffice/Output.csv",format='csv',header=True)  
>>>
```

# Partitions of the data on cloud

The screenshot shows the Google Cloud Storage interface for the 'nyctraffice' bucket. The left sidebar has 'Cloud Storage' selected under 'Buckets'. The main area shows 'Bucket details' for 'nyctraffice' with 'us (multiple regions in United States)' as the location, 'Standard' as the storage class, and 'NOT public' as the public access level. The 'OBJECTS' tab is active, displaying a list of approximately 26 objects. Each object entry includes columns for Name, Size, Type, Created, Storage class, Last modified, Public access, Version history, Encryption, Retention expiry date, and Holds. Most objects have a size of 128 MB and are of type 'application/octet-stream'. The 'Created' column shows dates from November 2022. The 'Encryption' column indicates 'Google-managed key' for all entries. The 'Retention expiry date' and 'Holds' columns show 'None' for all objects.

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiry date	Holds
__SUCCESS	0 B	application/octet-stream	14 Nov 2022, 22:33:04	Standard	14 Nov 2022, 22:33:04	Not public	—	Google-managed key	—	None
part-00000-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:31:33	Standard	14 Nov 2022, 22:31:33	Not public	—	Google-managed key	—	None
part-00001-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:31:48	Standard	14 Nov 2022, 22:31:48	Not public	—	Google-managed key	—	None
part-00002-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:31:51	Standard	14 Nov 2022, 22:31:51	Not public	—	Google-managed key	—	None
part-00003-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:31:52	Standard	14 Nov 2022, 22:31:52	Not public	—	Google-managed key	—	None
part-00004-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:31:49	Standard	14 Nov 2022, 22:31:49	Not public	—	Google-managed key	—	None
part-00005-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:03	Standard	14 Nov 2022, 22:32:03	Not public	—	Google-managed key	—	None
part-00006-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:04	Standard	14 Nov 2022, 22:32:04	Not public	—	Google-managed key	—	None
part-00007-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:06	Standard	14 Nov 2022, 22:32:06	Not public	—	Google-managed key	—	None
part-00008-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:07	Standard	14 Nov 2022, 22:32:07	Not public	—	Google-managed key	—	None
part-00009-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:18	Standard	14 Nov 2022, 22:32:18	Not public	—	Google-managed key	—	None
part-00010-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:18	Standard	14 Nov 2022, 22:32:18	Not public	—	Google-managed key	—	None
part-00011-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:21	Standard	14 Nov 2022, 22:32:21	Not public	—	Google-managed key	—	None
part-00012-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:21	Standard	14 Nov 2022, 22:32:21	Not public	—	Google-managed key	—	None
part-00013-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:33	Standard	14 Nov 2022, 22:32:33	Not public	—	Google-managed key	—	None
part-00014-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:33	Standard	14 Nov 2022, 22:32:33	Not public	—	Google-managed key	—	None
part-00015-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:34	Standard	14 Nov 2022, 22:32:34	Not public	—	Google-managed key	—	None
part-00016-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:35	Standard	14 Nov 2022, 22:32:35	Not public	—	Google-managed key	—	None
part-00017-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:48	Standard	14 Nov 2022, 22:32:48	Not public	—	Google-managed key	—	None
part-00018-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:49	Standard	14 Nov 2022, 22:32:49	Not public	—	Google-managed key	—	None
part-00019-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:49	Standard	14 Nov 2022, 22:32:49	Not public	—	Google-managed key	—	None
part-00020-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:32:49	Standard	14 Nov 2022, 22:32:49	Not public	—	Google-managed key	—	None
part-00021-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:33:01	Standard	14 Nov 2022, 22:33:01	Not public	—	Google-managed key	—	None
part-00022-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:33:02	Standard	14 Nov 2022, 22:33:02	Not public	—	Google-managed key	—	None
part-00023-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	128 MB	application/octet-stream	14 Nov 2022, 22:33:03	Standard	14 Nov 2022, 22:33:03	Not public	—	Google-managed key	—	None
part-00024-98171302-d7a6-4126-8cc4-9add37c3207d-c000.c...	99.9 MB	application/octet-stream	14 Nov 2022, 22:33:01	Standard	14 Nov 2022, 22:33:01	Not public	—	Google-managed key	—	None

# Artifacts of Cluster Monitoring



# Data Visualization Using Matplotlib

DatabaseProjectipynb

All changes saved

Table of contents

New Section

{x} Section

```
f,ax=plt.subplots(1,2,figsize=(18,8))
df['Boro'].value_counts().plot.pie(autopct='%.1f%%',ax=ax[0],shadow=True)
ax[0].set_title('Boro')
ax[0].set_ylabel('')
sns.countplot('Boro',data=df,ax=ax[1])
ax[1].set_title('Boro')
plt.show()
```

Boro

Borough	Percentage
Brooklyn	30.3%
Queens	25.7%
Manhattan	22.5%
Bronx	16.8%
Staten Island	4.6%

Boro

Borough	Count
Queens	~2500
Staten Island	~500
Brooklyn	~2900
Bronx	~1600
Manhattan	~2200

0s completed at 12:56 AM

# Future Scope

- Analysis of the traffic count data.
- Using Classification Models to analyze data.
- Predicting the best locations for your residential based on traffic counts.
- Predicting the locations for business purposes.