

# Week#9 Labs

*Neha Agrawal*

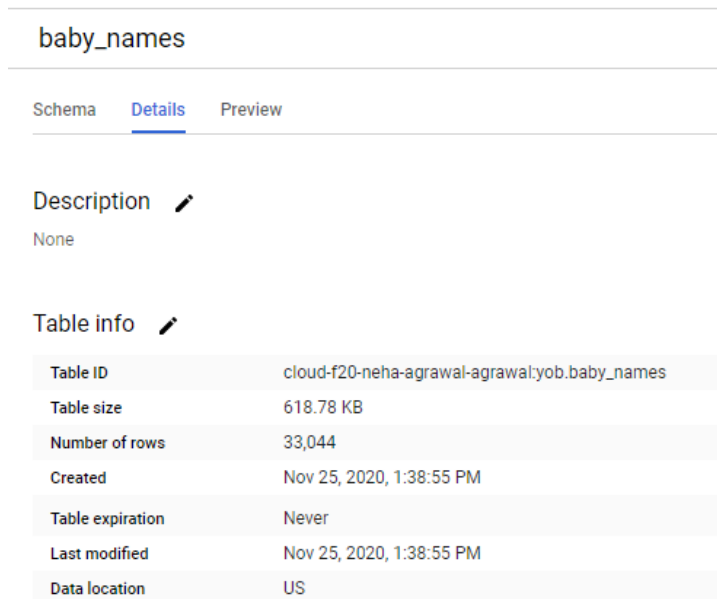
## Table of Contents



<b>BigQuery, JupyterLab .....</b>	<b>1</b>
BigQuery, Notebooks Lab #1 (Ingestion data) .....	1
Examine dataset .....	1
Query Data .....	1
BigQuery, Notebooks Lab #2 (Natality) .....	3
BigQuery query .....	3
Jupyter notebook query .....	3
Exploring the dataset .....	3
Run queries .....	3
Bigquery, Notebooks Lab #4 (COVID-19 Mobility) .....	4
Bigquery, Notebooks Lab #4 (COVID-19 NYT) .....	5
Run example queries .....	5
Write queries .....	8
Clean up.....	10

## 1. BigQuery, Notebooks Lab #1 (Ingesting data)

## 2. Examine dataset

- Take a screenshot of the table's details that includes the number of rows in the table.



baby_names	
Schema	Details Preview
Description 	
None	
Table info 	
Table ID	cloud-f20-neha-agrawal-agrawal:yob.baby_names
Table size	618.78 KB
Number of rows	33,044
Created	Nov 25, 2020, 1:38:55 PM
Table expiration	Never
Last modified	Nov 25, 2020, 1:38:55 PM
Data location	US

## 3. Query data

In the Query editor for your table, run a query that lists the 10 most popular female names in 2014.

- Screenshot your results and include it in your lab notebook


cloud-f20-Neha-Agrawal-agrawal


& INFO

SHORTCUT

Query editor

Query results

 SAVE RESULTS

 EXPLORE DATA

Query complete (0.2 sec elapsed, 618.8 KB processed)

Job information

Results

JSON

Execution details

Row	name	count
1	Emma	20799
2	Olivia	19674
3	Sophia	18490
4	Isabella	16950
5	Ava	15586
6	Mia	13442
7	Emily	12562
8	Abigail	11985
9	Madison	10247
10	Charlotte	10048

Bring up a Cloud Shell session and run a query to get the 10 least popular boys names in 2014.

- Screenshot your results and include it in your lab notebook

```
agrawal@cloudshell:~ (cloud-f20-neha-agrawal-agrawal) $  
limit 10"  
Waiting on bqjob_r6abca403c18a61d7_0000017601602ac8_1  
+-----+-----+  
| name | count |  
+-----+-----+  
| Aari | 5 |  
| Aaliyah | 5 |  
| Aadian | 5 |  
| Aaroh | 5 |  
| Aarit | 5 |  
| Aadiiv | 5 |  
| Aadhi | 5 |  
| Aarohan | 5 |  
| Aariyan | 5 |  
| Aamer | 5 |  
+-----+-----+
```

### Via interactive bq session

At the prompt, you can then enter your query. Run a query to find the 10 most popular male names in 2014.

- Screenshot your results and include it in your lab notebook

```
cloud-f20-neha-agrawal-agrawal> SELECT name, count  
Waiting on bqjob_r7138ee27320fa7d5_0000017601629  
+-----+-----+  
| name | count |  
+-----+-----+  
| Noah | 19144 |  
| Liam | 18342 |  
| Mason | 17092 |  
| Jacob | 16712 |  
| William | 16687 |  
| Ethan | 15619 |  
| Michael | 15323 |  
| Alexander | 15293 |  
| James | 14301 |  
| Daniel | 13829 |  
+-----+-----+  
cloud-f20-neha-agrawal-agrawal> █
```

Finally, run a query on your name. How popular was it?

- Screenshot your results and include it in your lab notebook

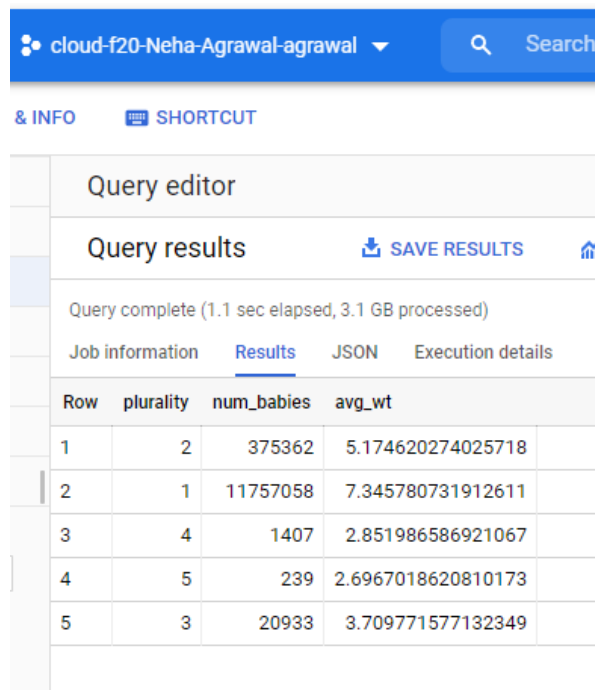
```
cloud-f20-neha-agrawal-agrawal> Select name, count from [cloud-f20-neha-agrawal-agrawal.yob.baby_names] where name='Neha'  
Waiting on bqjob_r3214b182b15ec88d_000001760165b52c_1 ... (0s) Current status: DONE  
+-----+-----+  
| name | count |  
+-----+-----+  
| Neha | 67 |  
+-----+-----+
```

#### 4. BigQuery, Notebooks Lab #2 (Natality)

#### 5. BigQuery query

Answer the following question for your lab notebook:

- How many twins were born during this time?



The screenshot shows the Google Cloud BigQuery interface. At the top, there's a blue header with the user 'cloud-f20-Neha-Agrawal-agrawal' and a search bar. Below the header, there are tabs for '& INFO' and 'SHORTCUT'. The main area is divided into two sections: 'Query editor' and 'Query results'. The 'Query results' section shows a message 'Query complete (1.1 sec elapsed, 3.1 GB processed)' and a 'SAVE RESULTS' button. Below this, there are tabs for 'Job information', 'Results', 'JSON', and 'Execution details'. The 'Results' tab is active, displaying a table with 5 rows and 4 columns: 'Row', 'plurality', 'num\_babies', and 'avg\_wt'. The first row shows 2 twins with 375362 babies and an average weight of 5.174620274025718. The second row shows 1 twin with 11757058 babies and an average weight of 7.345780731912611. The third row shows 4 twins with 1407 babies and an average weight of 2.851986586921067. The fourth row shows 5 twins with 239 babies and an average weight of 2.6967018620810173. The fifth row shows 3 twins with 20933 babies and an average weight of 3.709771577132349.

Row	plurality	num_babies	avg_wt
1	2	375362	5.174620274025718
2	1	11757058	7.345780731912611
3	4	1407	2.851986586921067
4	5	239	2.6967018620810173
5	3	20933	3.709771577132349

*375362 twins were born between 2001 and 2003*

#### 6. Jupyter notebook query

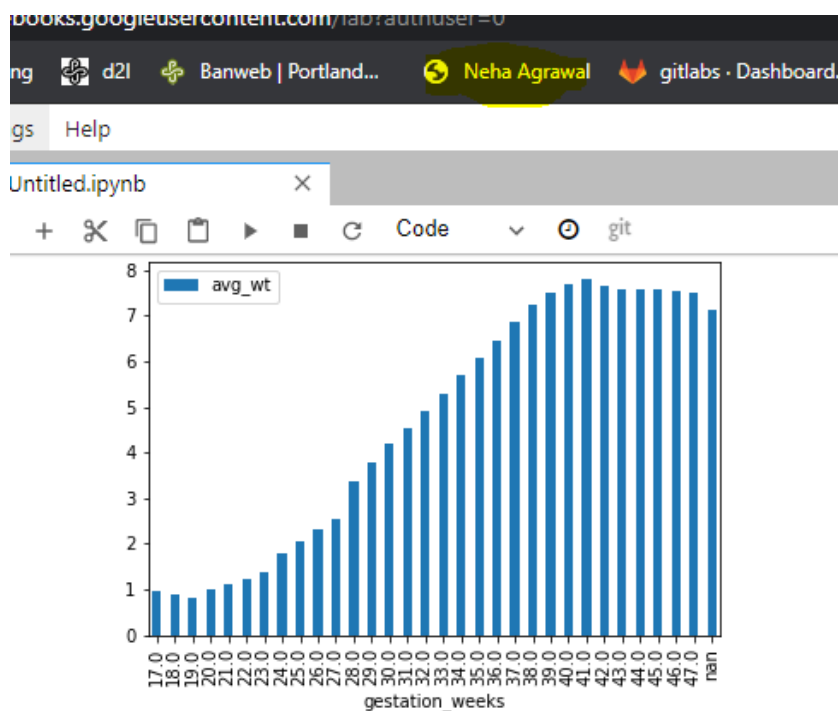
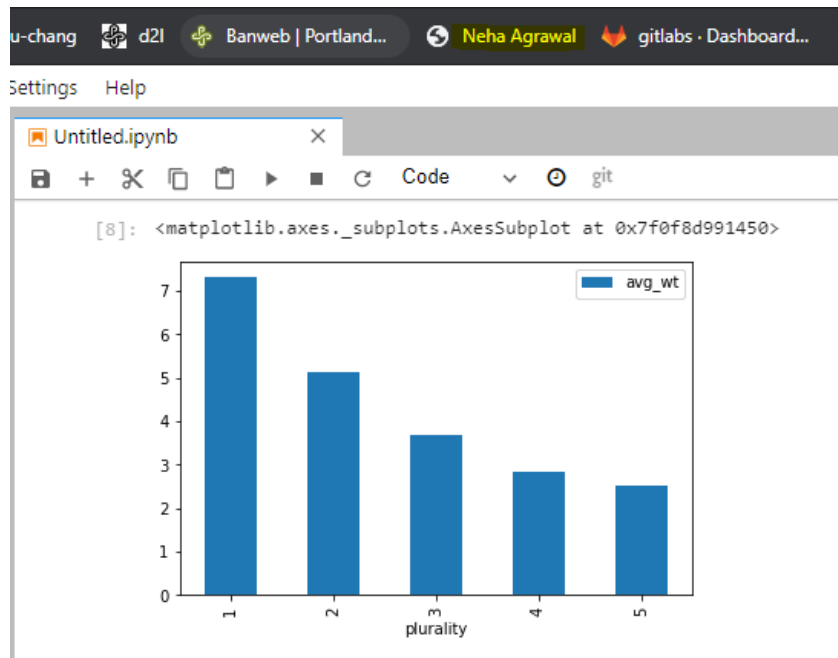
#### 7. Exploring the dataset

#### 8. Run queries

In examining the plots, which two features are the strongest predictors for a newborn baby's weight?

- Show the plots generated for the two most important features for your lab notebook

*The two most important features to determine the newborn baby's weight are plurality and gestation weeks.*



## 9. BigQuery, Notebooks Lab #3 (COVID-19 Mobility)

Find the link that documents what the dataset measures and answer the following question:

- **What dates are used as a baseline for the mobility data?**

Baseline Period: Jan 3–Feb 6, 2020.

- **What day saw the largest spike in trips to grocery and pharmacy stores?**  
13 March, 2020
- **On the day the stay-at-home order took effect (3/23/2020), what was the total impact on workplace trips?**  
We saw a sudden decline in the workplace trips. On 22 March it was -34, but on 23 March it dropped further down to -49.
- **Which three airports were impacted the most in April 2020 (the month when lockdowns became widespread)?**  
McCarran International, San Francisco International, Denver International
- **Run the query again using the month of August 2020. Which three airports were impacted the most?**  
McCarran International, Detroit Metropolitan Wayne County, San Francisco International

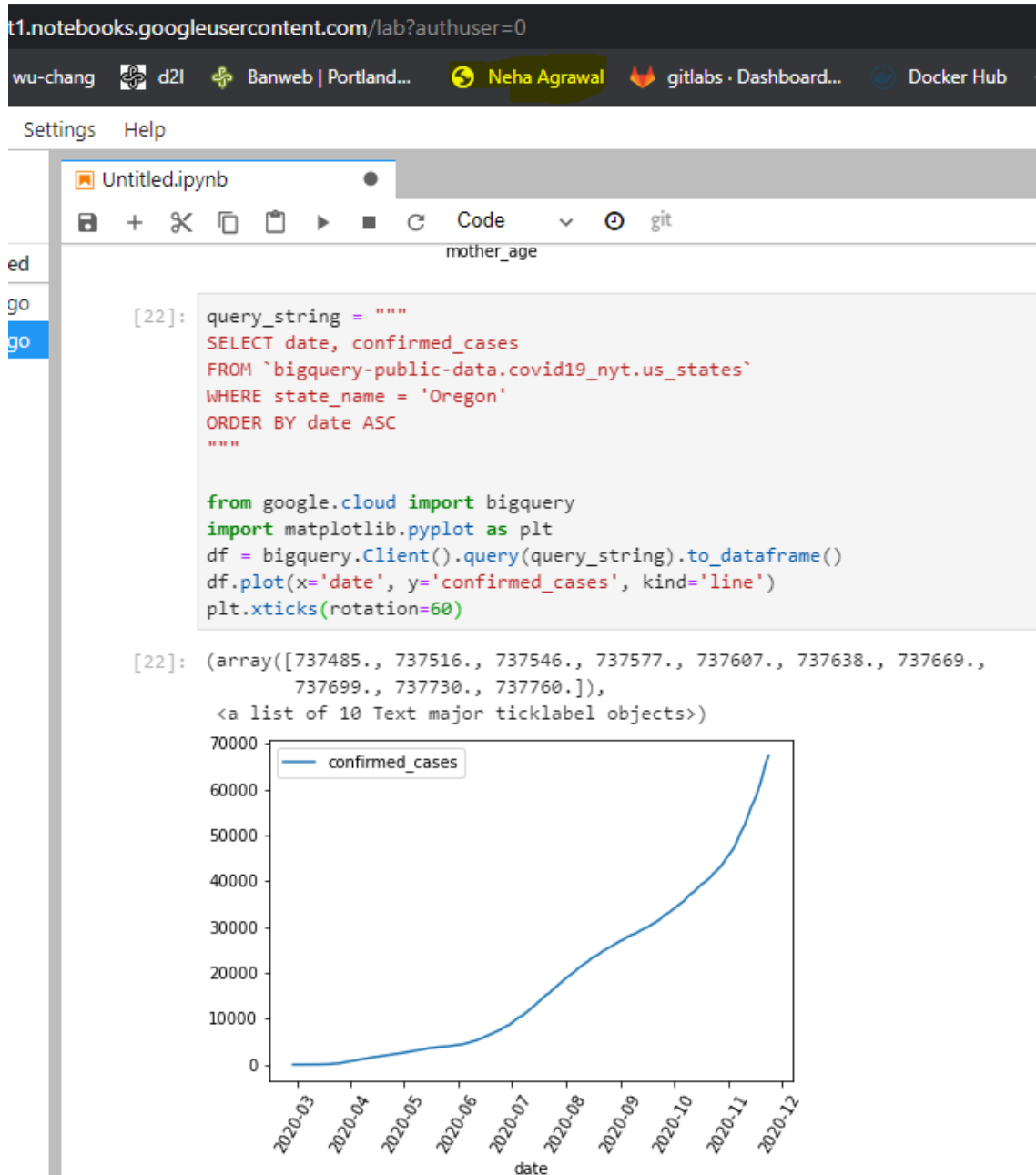
## 10. BigQuery, Notebooks Lab #4 (COVID-19 NYT)

There are four tables within the dataset. Click on each and view their schemas to see the columns they have. Find the following for subsequent queries that you will need to run

- **What table and columns identify the place name, the starting date, and the number of excess deaths from COVID-19?**  
excess\_deaths
- **What table and columns identify the date, county, and deaths from COVID-19?**  
us\_counties
- **What table and columns identify the date, state, and confirmed cases of COVID-19?**  
us\_states
- **What table and columns identify a county code and the percentage of its residents that report they always wear masks?**  
mask\_use\_by\_county

## 11. Run example queries

- **Show a screenshot of the plot and the code used to generate it for your lab notebook**



- From within your Jupyter notebook, run the query and write code that shows the first 10 states that reached 1000 deaths from COVID-19. Take a screenshot for your lab notebook.

d2lBanweb | Portland...Neha Agrawalgitlabs · Dashboard...

Help

Intitled.ipynb

+✂️📄📌▶️■🔄Code⌵🕒git

```
[23]: query_string = """
      SELECT state_name, MIN(date) as date_of_1000
      FROM `bigquery-public-data.covid19_nyt.us_states`
      WHERE deaths > 1000
      GROUP BY state_name
      ORDER BY date_of_1000 ASC
      """

      from google.cloud import bigquery
      import matplotlib.pyplot as plt
      df = bigquery.Client().query(query_string).to_dataframe()
      df.head(10)
```

```
[23]:
```

	state_name	date_of_1000
0	New York	2020-03-29
1	New Jersey	2020-04-06
2	Michigan	2020-04-09
3	Louisiana	2020-04-14
4	Massachusetts	2020-04-15
5	Illinois	2020-04-16
6	California	2020-04-17
7	Connecticut	2020-04-17
8	Pennsylvania	2020-04-17
9	Florida	2020-04-24

- Take a screenshot for your lab notebook of the Top 5 counties and the states they are located in.



The screenshot shows a Jupyter Notebook window titled 'Untitled.ipynb'. The code cell [24]: contains a SQL query string and Python code to execute it. The query selects distinct county\_fips\_code, always, and county from a BigQuery table, joined with another table, ordered by always in descending order. The results are displayed as a table with 5 rows and 4 columns: county\_fips\_code, always, county, and an unnamed column (likely date). The results show the top 5 counties by 'always' value.

```
[24]: query_string = """
SELECT DISTINCT mu.county_fips_code, mu.always,
FROM `bigquery-public-data.covid19_nyt.mask_u
LEFT JOIN `bigquery-public-data.covid19_nyt.u
ON mu.county_fips_code = ct.county_fips_code
ORDER BY mu.always DESC
"""

from google.cloud import bigquery
import matplotlib.pyplot as plt
df = bigquery.Client().query(query_string).to_dataframe()
df.head(5)
```

	county_fips_code	always	county	
0	06027	0.889	Inyo	
1	36123	0.884	Yates	
2	48229	0.880	Hudspeth	
3	06051	0.880	Mono	
4	48141	0.877	El Paso	

[ ]:

## 12. Write queries

Using your Jupyter notebook, perform the following...

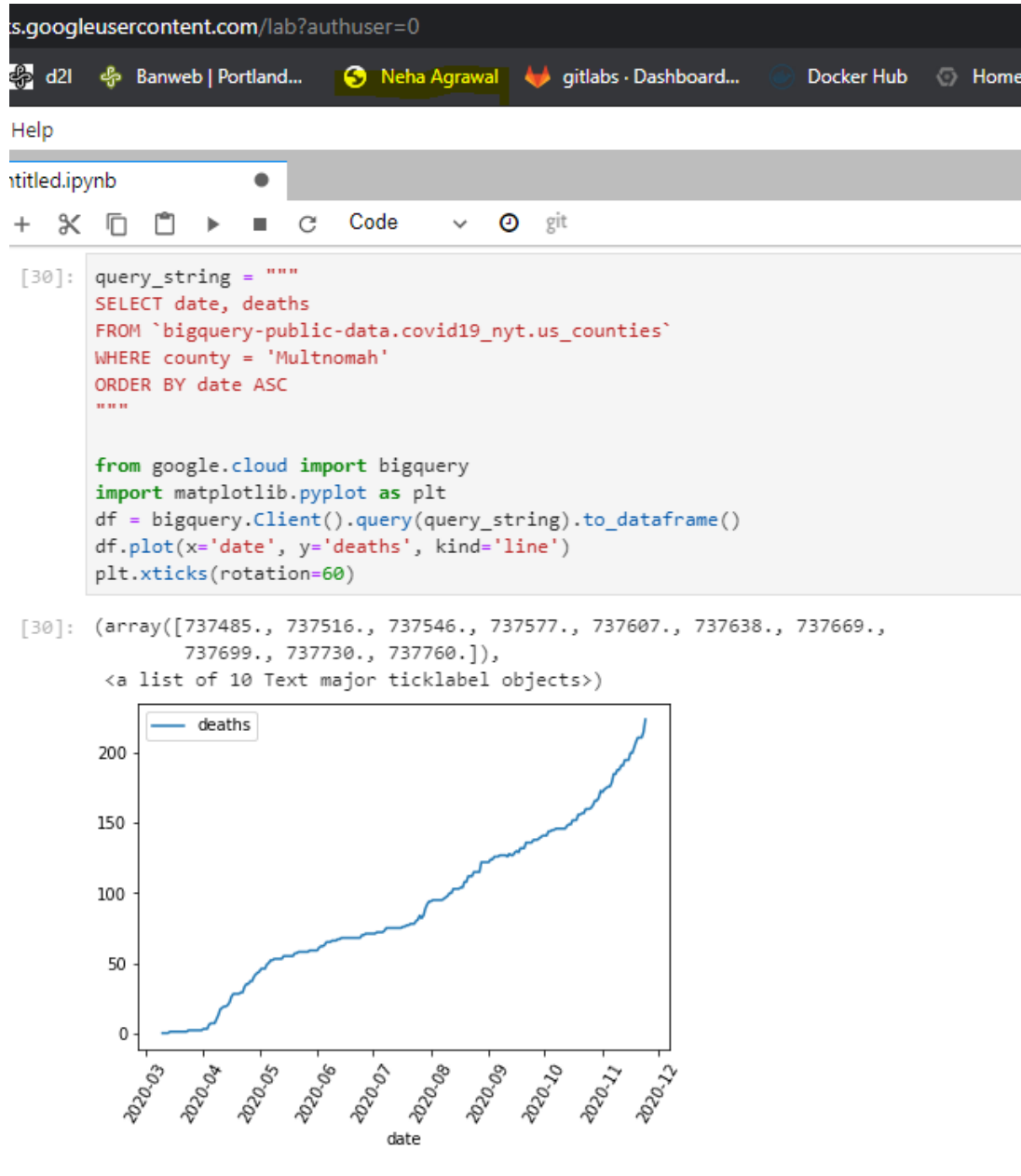
### Deaths in Multnomah county

Construct a query string that obtains the number of deaths from COVID-19 that have occurred in Multnomah county for each day in the dataset, ensuring the data is returned in ascending order of date. Run the query and obtain the results.

- Plot the results and take a screenshot for your lab notebook.

QUERY:

```
SELECT date, deaths FROM `bigquery-public-data.covid19_nyt.us_counties` WHERE  
county = 'Multnomah' ORDER BY date ASC
```



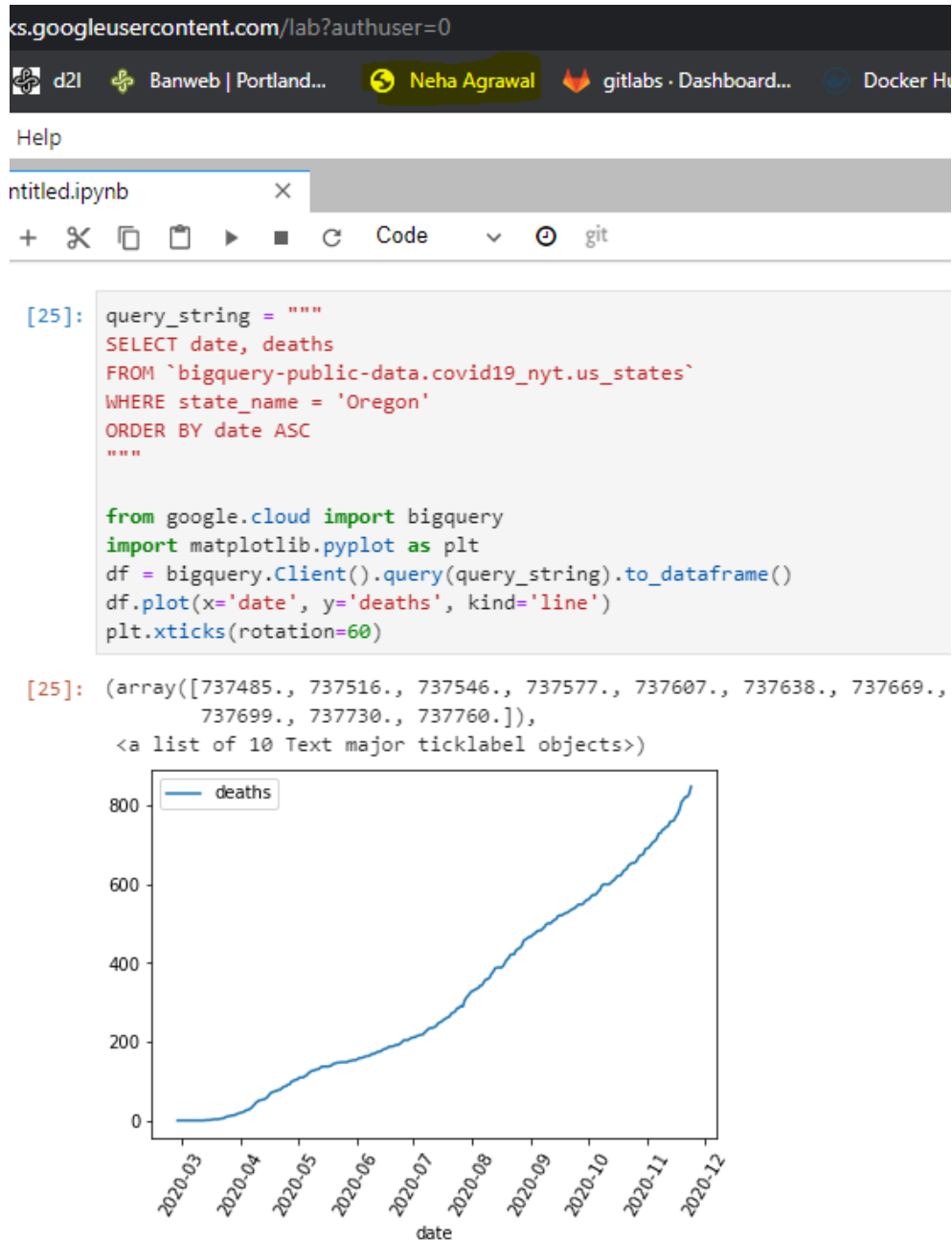
## Deaths in Oregon

Construct a query string that obtains the total number of deaths from COVID-19 that have occurred in Oregon for each day in the dataset, ensuring the data is returned in ascending order of date. Run the query and obtain the results.

- Plot the results and take a screenshot for your lab notebook.

QUERY:

```
SELECT date, deaths FROM `bigquery-public-data.covid19_nyt.us_states` WHERE
state_name = 'Oregon' ORDER BY date ASC
```



### 13. Clean up