

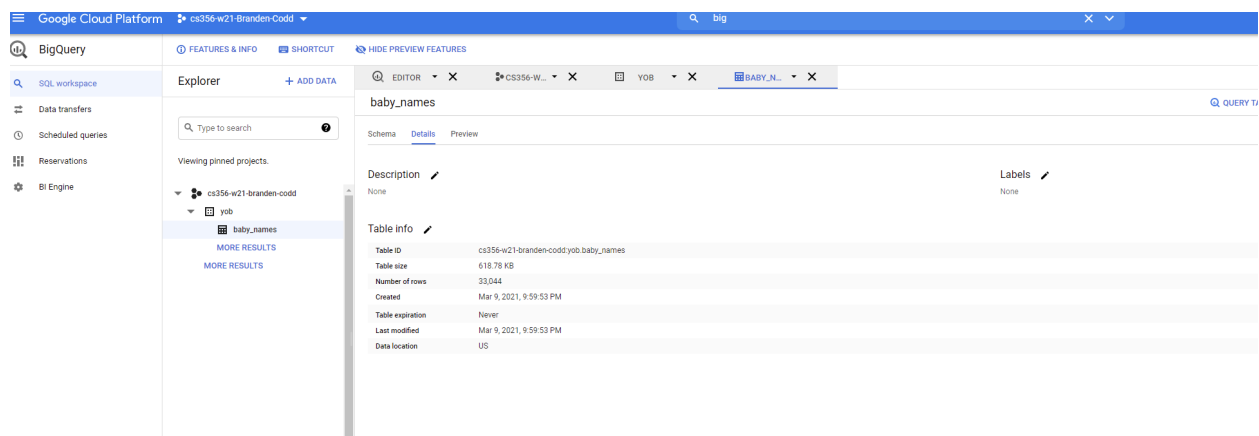
## 1. BigQuery, Notebooks Lab #1 (Ingesting data)

- No screenshot or observations

## 2. Examine dataset

Then, click on the "Details" tab.

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



## 3. Query data

Run the query, then hide the editor to see your query results.

- Screenshot your results and include it in your lab notebook

Explorers + ADD DATA

Q EDITOR X CS356-W... X YOB X BABY\_N... X Q UNSAVE... 2 X

1 SELECT name, count  
2 FROM cs356-w21-branden-codd.yob.baby\_names  
3 where sex = 'M'  
4 order by count desc  
5 LIMIT 20

Query results SAVE RESULTS EXPLORE DATA

Query complete (0.5 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
11	Harper	9564
12	Sofia	9542
13	Avery	9517
14	Elizabeth	9492
15	Amelia	8727
16	Evelyn	8692
17	Ella	8489
18	Chloe	8469
19	Victoria	7955
20	Aubrey	7589

- Screenshot your results and include it in your lab notebook

```
codd@cloudshell:~ (cs356-w21-branden-codd) $ bq query "select name, count FROM [cs356-w21-branden-codd.yob.baby_names] where sex='M' order by count asc limit 10"
Waiting on bqjob_r2bde96fe44cc2069_000001781ac74d92_1 ... (0s) Current status: DONE
```

name	count
Aari	5
Aaliyah	5
Aadian	5
Aaroh	5
Aarit	5
Aadiv	5
Aadhi	5
Aarohan	5
Aariyan	5
Aamer	5

```
codd@cloudshell:~ (cs356-w21-branden-codd) $
```

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

```
cs356-w21-branden-codd> select name, count from [cs356-w21-branden-codd.yob.baby_names] where sex='M' order by count desc limit 10
Waiting on bqjob_r5a53b249d074ce32_000001781aca1963_1 ... (0s) Current status: DONE
```

name	count
Noah	19144
Liam	18342
Mason	17092
Jacob	16712
William	16687
Ethan	15619
Michael	15323
Alexander	15293
James	14301
Daniel	13829

```
cs356-w21-branden-codd>
```

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

- Screenshot your results and include it in your lab notebook

```
cs356-w21-branden-codd> select name, count from [cs356-w21-branden-codd.yob.baby_names] where name="Branden"
Waiting on bqjob_r6791cbfdaa046429_000001781acb4cd3_1 ... (0s) Current status: DONE
+-----+-----+
| name | count |
+-----+-----+
| Branden | 202 |
+-----+-----+
cs356-w21-branden-codd> █
```

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

- No screenshot or observations

## 5. BigQuery query

Answer the following question for your lab notebook:

- How many twins were born during this time?
  - 375362

## 6. Jupyter notebook query

- No screenshot or observations

## 7. Exploring the dataset

- No screenshot or observations

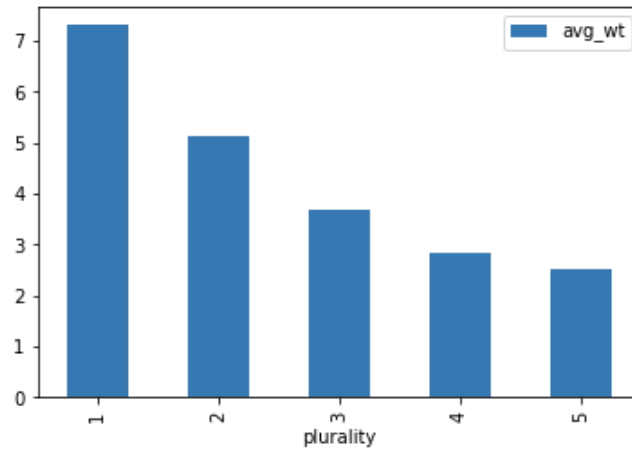
## 8. Run queries

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

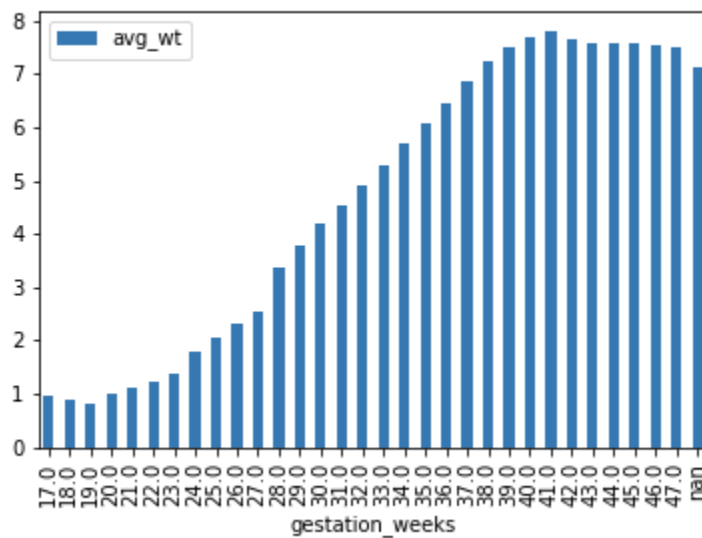
gestation time, plurality

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

```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd4788f6990>
```



```
[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd4787f2550>
```



## 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?
  - Changes for a given date as compared to baseline. Baseline is the median value, for the corresponding day of the week, during the 5-week period Jan 3–Feb 6, 2020.

- What day saw the largest spike in trips to grocery and pharmacy stores?
  - 2020-03-13
- On the day the stay-at-home order took effect (3/23/2020), what was the total impact on workplace trips?
  - -49
  - Large drop
- Which three airports were impacted the most in April 2020 (the month when lockdowns became widespread)?

Row	airport_name	traffic_fraction
1	McCarran International	32.666666666666666
2	San Francisco International	38.599999999999999
3	Denver International	38.633333333333326

- Run the query again using the month of August 2020. Which three airports were impacted the most?

Row	airport_name	traffic_fraction
1	McCarran International	40.933333333333333
2	Detroit Metropolitan Wayne County	46.133333333333334
3	San Francisco International	51.333333333333336


## 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?

<div> <div>Run</div> <div>Save query</div> <div>Save view</div> <div>Schedule query</div> <div>More</div> </div>				
excess_deaths				
<div> <div>Schema</div> <div>Details</div> <div>Preview</div> </div>				
Field name	Type	Mode	Policy tags	Description
country	STRING	NULLABLE		
placename	STRING	NULLABLE		
frequency	STRING	NULLABLE		
start_date	DATE	NULLABLE		
end_date	DATE	NULLABLE		
year	STRING	NULLABLE		
month	INTEGER	NULLABLE		
week	INTEGER	NULLABLE		
deaths	INTEGER	NULLABLE		
expected_deaths	INTEGER	NULLABLE		
excess_deaths	INTEGER	NULLABLE		
baseline	STRING	NULLABLE		
<div>Edit schema</div>				

- What table and columns identify the date, county, and deaths from COVID-19?

<div> <div>Run</div> <div>Save query</div> <div>Save view</div> <div>Schedule query</div> <div>More</div> </div>				
us_counties				
<div> <div>Schema</div> <div>Details</div> <div>Preview</div> </div>				
Field name	Type	Mode	Policy tags 	Description
date	DATE	NULLABLE		
county	STRING	NULLABLE		
state_name	STRING	NULLABLE		
county_fips_code	STRING	NULLABLE		
confirmed_cases	INTEGER	NULLABLE		
deaths	INTEGER	NULLABLE		
<div>Edit schema</div>				


- What table and columns identify the date, state, and confirmed cases of COVID-19?

## us\_states

Schema

Details

Preview

Field name	Type	Mode	Policy tags 	Descr
date	DATE	NULLABLE		
state_name	STRING	NULLABLE		
state_fips_code	STRING	NULLABLE		
confirmed_cases	INTEGER	NULLABLE		
deaths	INTEGER	NULLABLE		

Edit schema

- What table and columns identify a county code and the percentage of its residents that report they always wear masks?



<div> <div>Run</div> <div>Save query</div> <div>Save view</div> <div>Schedule query</div> <div>More</div> </div>				
mask_use_by_county				
<div> <div>Schema</div> <div>Details</div> <div>Preview</div> </div>				
Field name	Type	Mode	Policy tags	Description
county_fips_code	STRING	NULLABLE		
never	FLOAT	NULLABLE		
rarely	FLOAT	NULLABLE		
sometimes	FLOAT	NULLABLE		
frequently	FLOAT	NULLABLE		
always	FLOAT	NULLABLE		
<div>Edit schema</div>				

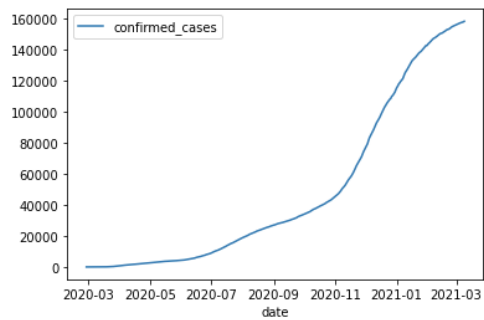
11. Run example queries

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

```
[27]: def get_distinct_values(column_name):
      query_string = """
      SELECT date,
      confirmed_cases
      FROM `bigquery-public-data.covid19_nyt.us_states`
      WHERE state_name = 'Oregon'
      ORDER BY date ASC
      """
      return bigquery.Client().query(query_string).to_dataframe().sort_values(column_name)
```

```
[28]: df = get_distinct_values('date')
      df.plot(x='date', y='confirmed_cases', kind='line')
```

```
[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd478581c10>
```



- 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.

```
[37]: 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
      """
```

```
[38]: from google.cloud import bigquery
      df = bigquery.Client().query(query_string).to_dataframe()
      df.head(10)
```

```
[38]:
```

	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.

```
[41]: query_string = """
      SELECT DISTINCT mu.county_fips_code, mu.always, ct.county, state_name
      FROM `bigquery-public-data.covid19_nyt.mask_use_by_county` as mu
      LEFT JOIN `bigquery-public-data.covid19_nyt.us_counties` as ct
      ON mu.county_fips_code = ct.county_fips_code
      ORDER BY mu.always DESC
      """
```

```
[42]: from google.cloud import bigquery
      df = bigquery.Client().query(query_string).to_dataframe()
      df.head(5)
```

```
[42]:
```

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

```
[ ]:
```

## 12. Write queries

### Deaths in Jackson county

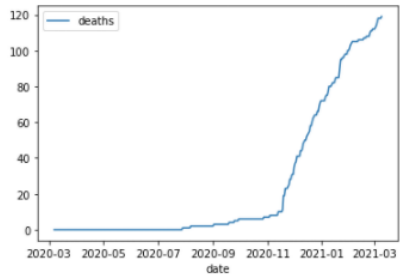
Construct a query string that obtains the number of deaths from COVID-19 that have occurred in Jackson county for each day in the dataset, ensuring the data is returned in ascending order of date. Run the query and obtain the results.

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

```
[58]: def get_distinct_values(column_name):
      query_string = """
      SELECT
        deaths,
        date
      FROM
        `bigquery-public-data.covid19_nyt.us_counties`
      WHERE
        county = 'Jackson' and state_name = 'Oregon'
      """
      return bigquery.Client().query(query_string).to_dataframe().sort_values(column_name)
```

```
[59]: df = get_distinct_values('date')
      df.plot(x='date', y='deaths', kind='line')
```

```
[59]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd478212ed0>
```



```
[ ]:
```

## 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.

```

date
[62]: def get_distinct_values(column_name):
      query_string = """
      SELECT
        deaths,
        date
      FROM
        `bigquery-public-data.covid19_nyt.us_states`
      WHERE
        state_name = 'Oregon'
      """
      return bigquery.Client().query(query_string).to_dataframe().sort_values(column_name)

```

```

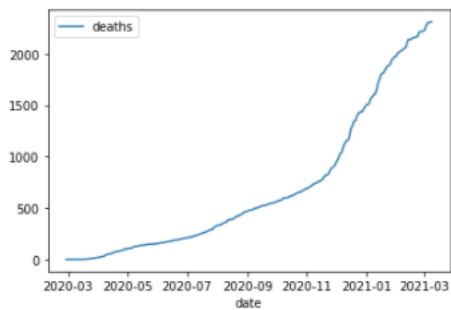
[63]: df = get_distinct_values('date')
      df.plot(x='date', y='deaths', kind='line')

```

```

[63]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd47341f210>

```



```

[ ]:

```