

Week #9 Labs

9.1g: BigQuery, BigLake	2
3. Create dataset	2
4. Query data	3
9. Query data	5
9.2g: Jupyter Notebooks	5
3. BigQuery query	5
6. Run queries	6
8. Mobility	7
9. Airport traffic	7
10. Mortality	8
11. Run example queries	8
12. Write queries	10
9.3g: Dataproc	12
6. Run computation	12
8. Run computation again	12
9.4g: Dataflow	13
3. Beam code	13
4. Run pipeline locally	14
5. Dataflow Lab #2 (Word count)	14
6. Run code locally	15
9. Run code using Dataflow runner	15
12. View raw data from PubSub	16
14. Run Dataflow job from template	16
15. Query data in BigQuery	17
16. Data visualization	19

9.1g: BigQuery, BigLake

3. Create dataset

The screenshot shows the Google Cloud Platform BigQuery interface. At the top, there is a navigation bar with the Google Cloud logo, the project name "cloud-metens", and a search bar. Below the navigation bar is the "Explorer" sidebar, which includes a search bar for BigQuery resources and a list of resources under the "cloud-metens" project. The "yob" dataset is expanded, showing its tables: "yob_native_table" and "yod". The "yob_native_table" table is currently selected. To the right of the sidebar is the main content area, which displays the details for the "yob_native_table" table. The "DETAILS" tab is selected, showing storage information: Number of rows (33,044), Total logical bytes (618.78 KB), Active logical bytes (618.78 KB), Long term logical bytes (0 B), Current physical bytes (0 B), Total physical bytes (0 B), and Active physical bytes (0 B). Below the storage info is a "Job history" section. At the bottom of the interface is a "CLOUD SHELL" terminal window.

Google Cloud cloud-metens Search (/) for resources, docs, products, etc.

Explorer + ADD 🔍

Search BigQuery resources ?

Viewing resources.
SHOW STARRED ONLY

cloud-metens star ⋮

- Queries ⋮
- Notebooks ⋮
- Data canvases ⋮
- Data preparations ⋮
- Workflows ⋮
- External connections ⋮

yob star ⋮

- yob_native_table star ⋮
- yod star ⋮

SUMMARY ^

CLOUD SHELL

Terminal (cloud-metens) X + ▾

cloudshell:~ (cloud-metens)\$ ls

Details for yob_native_table

SCHEMA DETAILS PREVIEW

Labels

Primary key(s)

Tags

Storage info ?

Number of rows	33,044
Total logical bytes	618.78 KB
Active logical bytes	618.78 KB
Long term logical bytes	0 B
Current physical bytes	0 B
Total physical bytes	0 B
Active physical bytes	0 B

Job history

4. Query data

The screenshot shows the Google Cloud BigQuery interface. At the top, there are navigation links for 'Google Cloud' and 'cloud-metens'. A search bar is also present. The main area is divided into two sections: 'Explorer' on the left and 'Untitled query' on the right.

Explorer: This section shows a tree view of resources under 'cloud-metens'. The 'yob' folder is expanded, showing its contents: 'yob_native_table' and 'yod'. There is a 'SHOW STARRED ONLY' filter applied.

Untitled query: This section contains a query editor and a results table.

Query Editor: The query is:1 SELECT name, count
2 FROM `cloud-metens.yob.yob_native_table`
3 WHERE gender='F'
4 ORDER BY count DESC
5 LIMIT 20

Results Table: The results table has four columns: Row, name, count, and a header row. The data is as follows:

Row	name	count
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

```

metens@cloudshell:~ (cloud-metens)$ wc -l yob2014.csv
33044 yob2014.csv
metens@cloudshell:~ (cloud-metens)$ bq query "SELECT name, count
FROM [cloud-metens.yob.yob_native_table]
WHERE gender='M'
ORDER BY count ASC
LIMIT 10"
+-----+-----+
| name | count |
+-----+-----+
| Aari | 5 |
| Aaliyah | 5 |
| Aadian | 5 |
| Aaroh | 5 |
| Aarit | 5 |
| Aadiv | 5 |
| Aadhi | 5 |
| Aarohan | 5 |
| Aariyan | 5 |
| Aamer | 5 |
+-----+-----+
metens@cloudshell:~ (cloud-metens)$ █

```

```

cloud-metens> SELECT name, count FROM [cloud-metens.yob.yob_native_table] WHERE gender='M' ORDER BY count DESC LIMIT 10
+-----+-----+
| name | count |
+-----+-----+
| Noah | 19144 |
| Liam | 18342 |
| Mason | 17092 |
| Jacob | 16712 |
| William | 16687 |
| Ethan | 15619 |
| Michael | 15323 |
| Alexander | 15293 |
| James | 14301 |
| Daniel | 13829 |
+-----+-----+
cloud-metens> █

```

```

cloud-metens> SELECT name, count FROM [cloud-metens.yob.yob_native_table] WHERE name='Nathan'
+-----+-----+
| name | count |
+-----+-----+
| Nathan | 8 |
| Nathan | 8902 |
+-----+-----+
cloud-metens> █

```

9. Query data

The screenshot shows the BigQuery web interface. On the left, the 'Explorer' pane displays a tree view of resources under the project 'cloud-metens'. It includes sections for 'Queries', 'Notebooks', 'Data canvases', 'Data preparations', 'Workflows', 'External connections', and two datasets: 'us-west1.biglake' and 'yob'. Under 'yob', there are tables 'yob_biglake_table' and 'yob_native_table'. The right pane is titled 'Untitled query' and contains a SQL query to select names and counts from the 'yob_biglake_table' where gender is 'F', ordered by count asc, and limited to 20 rows. Below the query is a 'Query results' table with 20 rows, each containing a name and a count of 5.

Row	name	count
10	Aavya	5
11	Aashni	5
12	Aadrika	5
13	Aamyah	5
14	Aamilah	5
15	Abagael	5
16	Aayusha	5
17	Aarion	5
18	Aania	5
19	Aaiza	5
20	Aabriella	5

9.2g: Jupyter Notebooks

3. BigQuery query

The size of the table is about 22GB and the size of the query is about 3GB. So the query processes about 7 times less data than the size of the table.

Between 2001 and 2003, there were 375362 twins born.

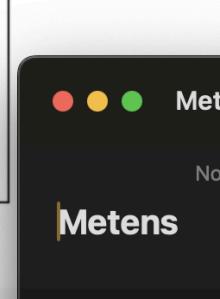
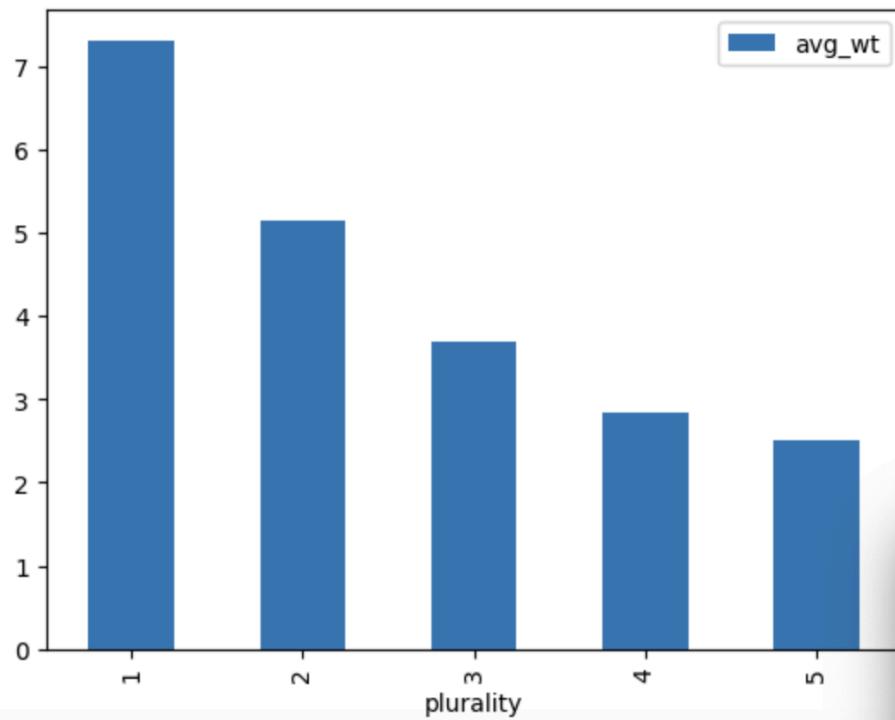
On average, twins are about 5 pounds each while single babies are about 7 pounds in weight.

6. Run queries

The two features with the strongest prediction of a babies weight are: **plurality** and **gestation weeks**.

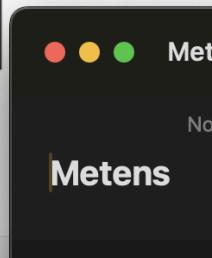
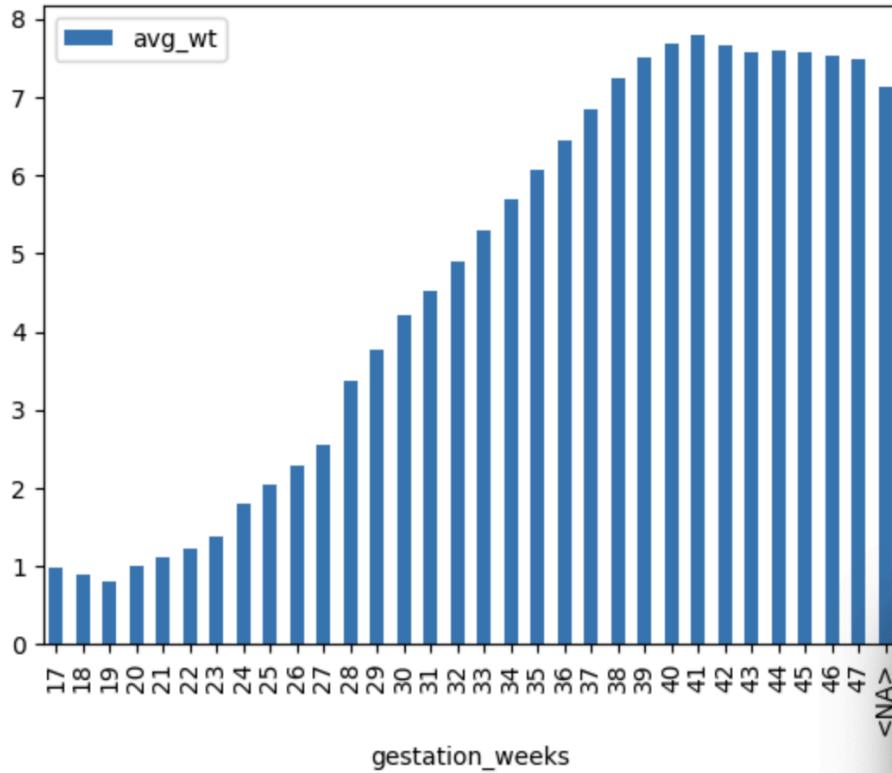
```
.....
return bigquery.Client().query(query_string).to_dataframe().sort_values(column_n
[7]: df = get_distinct_values('plurality')
df.plot(x='plurality', y='avg_wt', kind='bar')

[7]: <Axes: xlabel='plurality'>
```



```
[9]: df = get_distinct_values('gestation_weeks')
df.plot(x='gestation_weeks', y='avg_wt', kind='bar')
```

```
[9]: <Axes: xlabel='gestation_weeks'>
```



8. Mobility

The day that had the largest spike in trips to grocery and pharmacy stores (17) was **2020-03-13**. On the day the stay-at-home order took effect (3/23/2020), the total impact on workplace trips was **-49 from baseline**.

9. Airport traffic

The three airports that were impacted the most in April were:

- 1) Detroit Metropolitan Wayne County with 45 % or normal traffic
- 2) McCarran International with 45 %
- 3) San Francisco International with 47 %

In August, the three airports were the same, but their percentages were only slightly different:

- 1) Detroit Metropolitan Wayne County with 45 % or normal traffic
- 2) McCarran International with 44 %
- 3) San Francisco International with 53 %

10. Mortality

The table **excess_deaths** has columns: **placename**, **start_date**, and **excess_deaths**.

The table **us_counties** has columns: **date**, **county**, and **deaths**.

The table **us_states** has columns: **date**, **state_name**, and **confirmed_cases**.

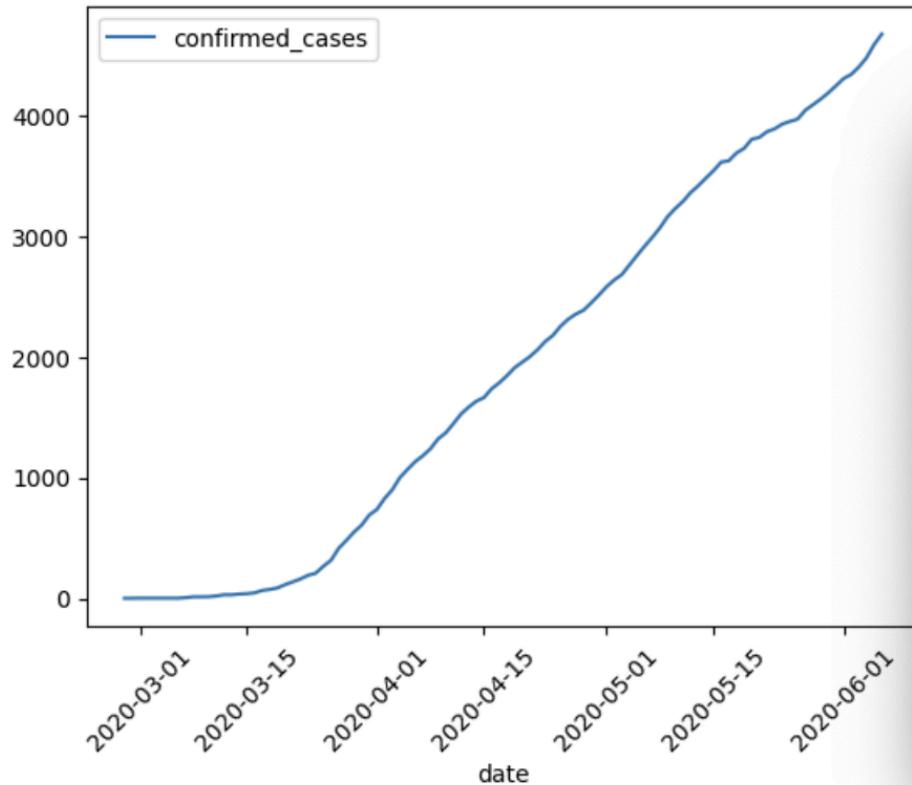
The table **mask_use_by_county** has columns: **county_fips_code** and **never, rarely, sometimes, frequently, always**.

11. Run example queries

```
[16]: query_string = """SELECT date, confirmed_cases  
FROM `bigquery-public-data.covid19_nyt.us_states`  
WHERE state_name = 'Oregon'  
ORDER BY date ASC"""\n\nfrom google.cloud import bigquery  
df = bigquery.Client().query(query_string + " LIMIT 100").to_dataframe()
```

```
[17]: df.plot(x='date', y='confirmed_cases', kind='line', rot=45)
```

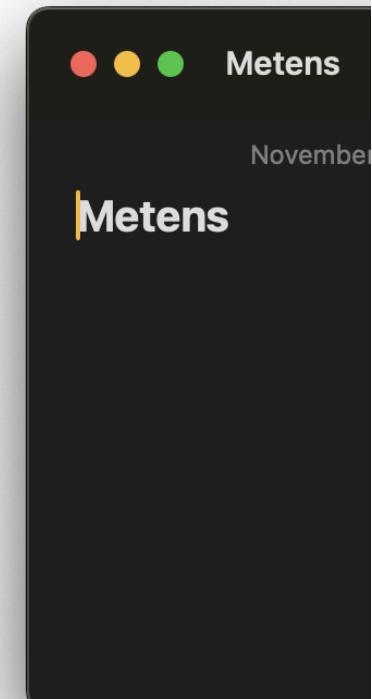
```
[17]: <Axes: xlabel='date'>
```



Metens

```
[20]: 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
df = bigquery.Client().query(query_string + " LIMIT 100").to_dataframe()
df.head(10)
```

	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	Connecticut	2020-04-17
7	Pennsylvania	2020-04-17
8	California	2020-04-17
9	Florida	2020-04-24



```
[27]: 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"""
from google.cloud import bigquery
df = bigquery.Client().query(query_string + " LIMIT 100").to_dataframe()
df.head()
```

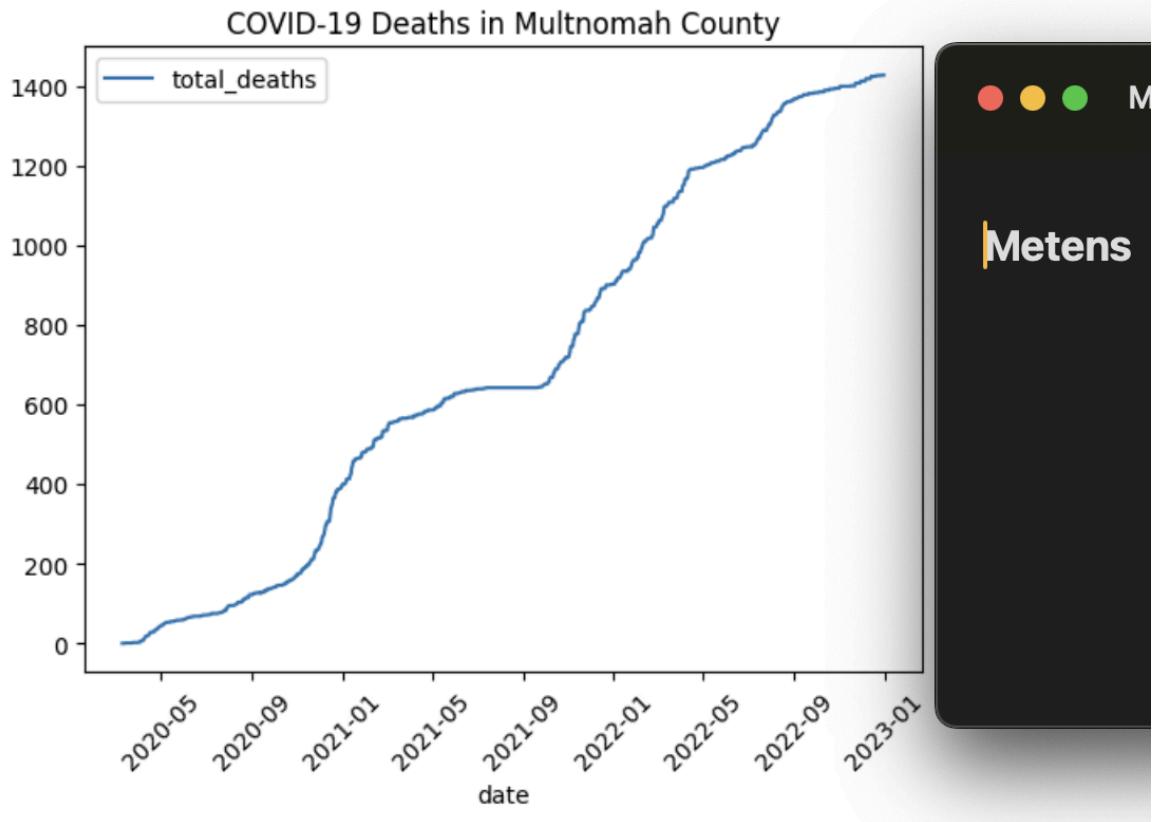
	county_fips_code	always	county	state_name
0	06027	0.889	Inyo	California
1	36123	0.884	Yates	New York
2	48229	0.880	Hudspeth	Texas
3	06051	0.880	Mono	California
4	48141	0.877	El Paso	Texas

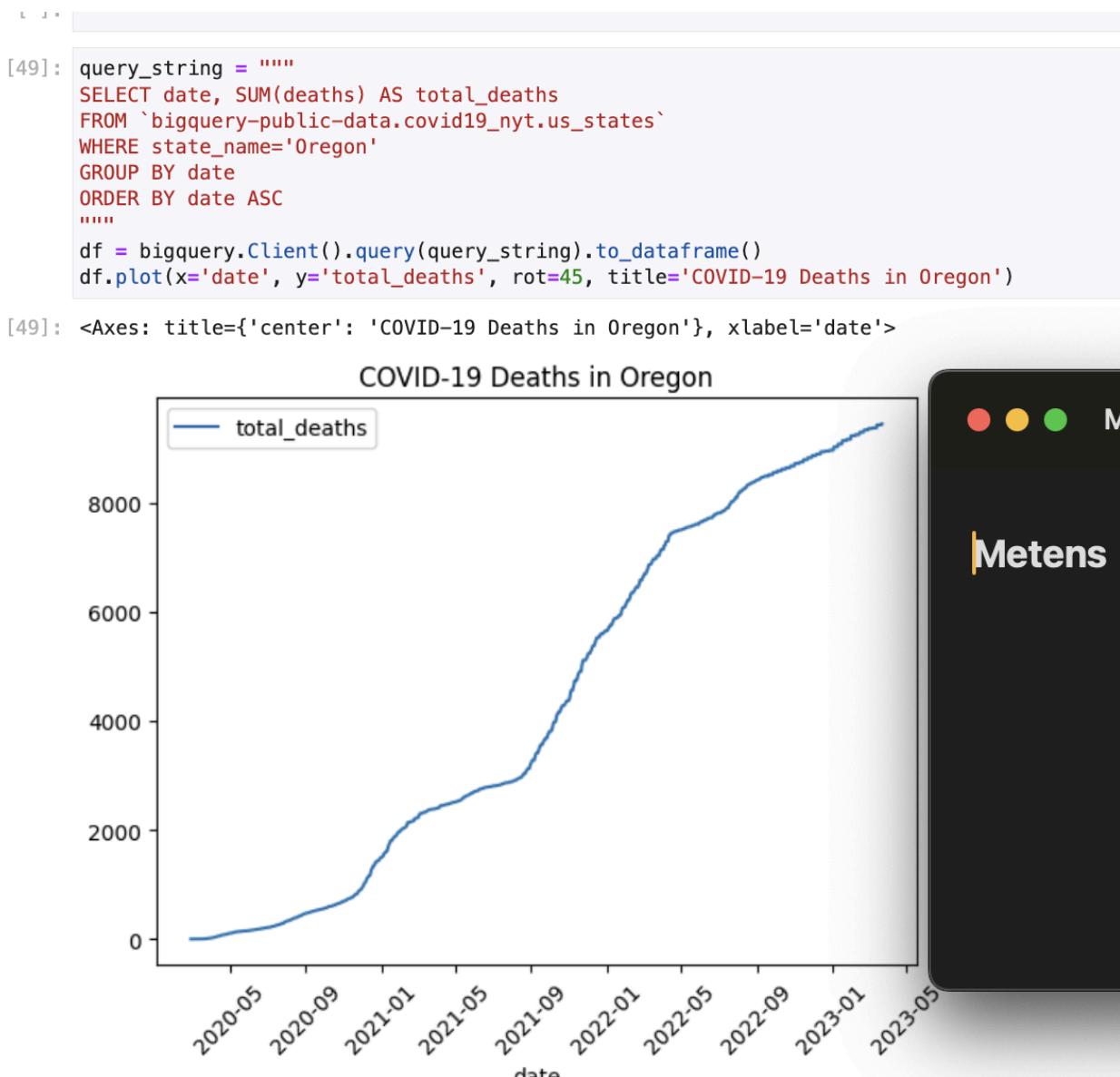


12. Write queries

```
[47]: query_string = """  
SELECT date, SUM(deaths) AS total_deaths  
FROM `bigquery-public-data.covid19_nyt.us_counties`  
WHERE county='Multnomah'  
GROUP BY date  
ORDER BY date ASC  
"""  
df = bigquery.Client().query(query_string).to_dataframe()  
df.plot(x='date', y='total_deaths', rot=45, title='COVID-19 Deaths in Multnomah County')
```

```
[47]: <Axes: title={'center': 'COVID-19 Deaths in Multnomah County'}, xlabel='date'>
```





9.3g: Dataproc

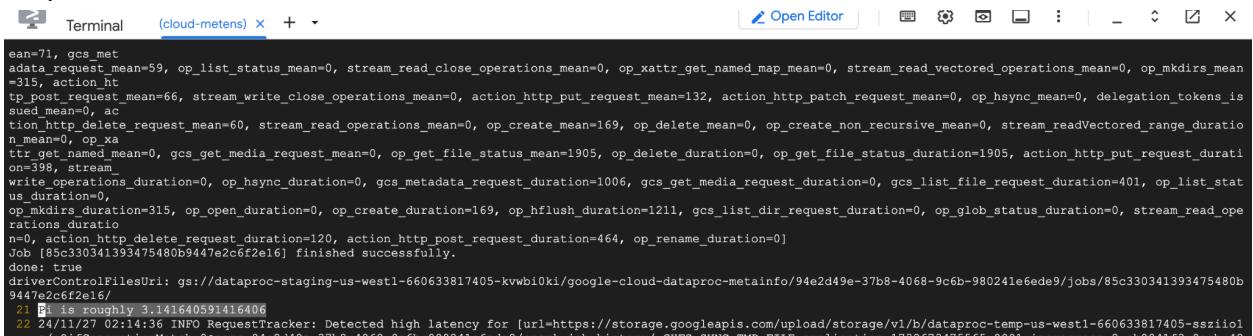
6. Run computation

```
Waiting for cluster creation operation...done.
Created [https://dataproc.googleapis.com/v1/projects/cloud-metens/regions/us-west1/clusters/metens-dplab] Cluster placed in zone [us-west1-b].
metens@cloudshell:~ (cloud-metens)$

gcloud dataproc jobs submit spark --cluster ${CLUSTERNAME} \
--class org.apache.spark.examples.SparkPi \
--jars file:///usr/lib/spark/examples/jars/spark-examples.jar -- 1000 \
> output.txt &
Wed Nov 27 02:13:40 AM UTC 2024
[1] 2100
metens@cloudshell:~ (cloud-metens)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME} ; date
JOB ID: 85c330341393475480b9447e2c6f2e16
TYPE: spark
STATUS: SETUP_DONE
Wed Nov 27 02:13:49 AM UTC 2024
metens@cloudshell:~ (cloud-metens)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME} ; date
JOB ID: 85c330341393475480b9447e2c6f2e16
TYPE: spark
STATUS: SETUP_DONE
Wed Nov 27 02:14:00 AM UTC 2024
metens@cloudshell:~ (cloud-metens)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME} ; date
JOB ID: 85c330341393475480b9447e2c6f2e16
TYPE: spark
STATUS: SETUP_DONE
Wed Nov 27 02:14:13 AM UTC 2024
metens@cloudshell:~ (cloud-metens)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME} ; date
JOB ID: 85c330341393475480b9447e2c6f2e16
TYPE: spark
STATUS: RUNNING
Wed Nov 27 02:14:29 AM UTC 2024
metens@cloudshell:~ (cloud-metens)$ gcloud dataproc jobs list --cluster ${CLUSTERNAME} ; date
JOB ID: 85c330341393475480b9447e2c6f2e16
TYPE: spark
STATUS: DONE
[1]+ Done                  gcloud dataproc jobs submit spark --cluster ${CLUSTERNAME} --class org.apache.spark.examples.SparkPi --jars file:///usr/lib/spark/examples/jars/spark-exa
mples.jar -- 1000 >output.txt
Wed Nov 27 02:15:17 AM UTC 2024
```

The job started at 02:13:40 and ended at 02:15:17. It took less than 1 minute and 37 seconds to finish.

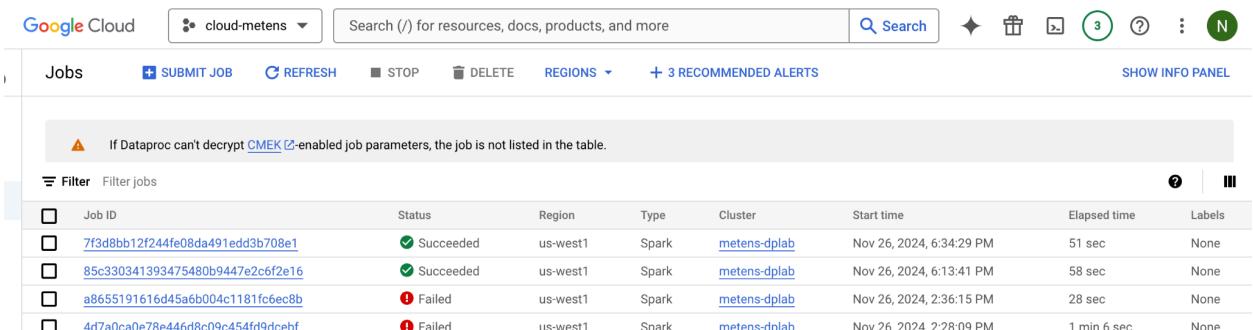
Output.txt:



```
ean=71, gcs met
data_request_mean=59, op_list_status_mean=0, stream_read_close_operations_mean=0, op_xattr_get_named_map_mean=0, stream_read_vectored_operations_mean=0, op_mkdirs_mean
=315, action ht
tp_post_request_mean=66, stream_write_close_operations_mean=0, action_http_put_request_mean=132, action_http_patch_request_mean=0, op_hsync_mean=0, delegation_tokens_is
sued_mean=0, ac
tion http_delete_request_mean=60, stream_read_operations_mean=0, op_create_mean=169, op_delete_mean=0, op_create_non_recursive_mean=0, stream_readVectored_range_dura
tion_mean=0, op_xa
ttr_get_named_mean=0, gcs_get_media_request_mean=0, op_get_file_status_mean=1905, op_delete_duration=0, op_get_file_status_duration=1905, action_http_put_request_dura
tion=398, stream
write_operations_duration=0, op_hsync_duration=0, gcs_metadata_request_duration=1006, gcs_get_media_request_duration=0, gcs_list_file_request_duration=401, op_list_stat
us_duration=0,
op_mkdirs_duration=315, op_open_duration=0, op_create_duration=169, op_hflush_duration=1211, gcs_list_dir_request_duration=0, op_glob_status_duration=0, stream_read_o
perations_dura
tio
n=0, action http_delete_request_duration=120, action http_post_request_duration=464, op_rename_duration=0]
Job [85c330341393475480b9447e2c6f2e16] finished successfully.
done: true
driverControlFileUri: gs://dataproc-staging-us-west1-660633817405-kvwbio1ki/google-cloud-dataproc-metainfo/94e2d49e-37b8-4068-9c6b-980241e6ede9/jobs/85c330341393475480b
9447e2c6f2e16/
21 Pi is roughly 3.141640591416406
22 24/11/27 02:14:36 INFO RequestTracker: Detected high latency for [url=https://storage.googleapis.com/upload/storage/v1/b/dataproc-temp-us-west1-660633817405-ssziol
r/2f69...]. Will retry 21 more times after 0.6s. 2024-11-27T02:14:36Z
```

Pi is roughly 3.141640591416406.

8. Run computation again



Job ID	Status	Region	Type	Cluster	Start time	Elapsed time	Labels
7f3d8bb12f244fe08da491edd3b708e1	Succeeded	us-west1	Spark	metens-dplab	Nov 26, 2024, 6:34:29 PM	51 sec	None
85c330341393475480b9447e2c6f2e16	Succeeded	us-west1	Spark	metens-dplab	Nov 26, 2024, 6:13:41 PM	58 sec	None
a8655191616d45a6b004c1181fc6ec8b	Failed	us-west1	Spark	metens-dplab	Nov 26, 2024, 2:36:15 PM	28 sec	None
4d7a0ca0e78e446d8c09c454fd9dcebfb	Failed	us-west1	Spark	metens-dplab	Nov 26, 2024, 2:28:09 PM	1 min 6 sec	None

The first run took 58 seconds, and this run took 51 seconds. It was 7 seconds faster.

CLOUD SHELL

Terminal (cloud-metens) x + -

```
1 less [CONTEXT ratelimit_period="1 MINUTES" ]
2 Waiting for job output...
3 24/11/27 02:34:38 INFO SparkEnv: Registering MapOutputTracker
4 24/11/27 02:34:38 INFO SparkEnv: Registering BlockManagerMaster
5 24/11/27 02:34:38 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
21 Pi is roughly 3.1414132314141323
   @
   @
   @
10  @
   @
   @
   @
13  @
14  @
   @
   @
q
a
s
t
3
s
```

Pi is roughly 3.1414132314141323.

9.4g: Dataflow

3. Beam code

The input variable: `input = '{0}*.java'`.format(options.input) comes from options, `pipeline_args = parser.parse_known_args()` which has `parser = argparse.ArgumentParser(description='Find the most used Java packages')`. So the input is taken from java files in the directory `(..javahelp/src/main/java/com/google/cloud/training/dataanalyst/javahelp/)` which is captured by the parser variable.

The call: `'write' >> beam.io.WriteToText(output_prefix)` uses the variable: `output_prefix = options.output_prefix` which is derived from this: `output_prefix = options.output_prefixble` which holds the output prefix is this:
`parser.add_argument('--output_prefix', default='/tmp/output', help='Output prefix')` Thus, the output goes to the default '`/tmp/output`' directory.

The **packageUse** method calls the **getPacKes** method which calls the **splitPackageName** method. The **packageUse** method gets back a packages variable, which is a list of split packages (e.g. ["com", "com.example", "com.example.appname", "com.example.appname.library", "com.example.appname.library.widgetname"]) and then creates tuples using yield, which outputs all the split packages in tuple form: ("com", 1), ("com.example", 1), etc.

The **totalUse** operation found on line 68: `TotalUse' >> beam.CombinePerKey(sum)`

uses **beam.CombinePerKey** method passing a sum function that essentially calculates how many times the packages appear in the input dataset.

Operations that correspond to a Map are those who implement a one to one operation for each element in a list with all the elements receiving the same treatment. Both the **GetImports** and **PackageUse** operations correspond to a map:

```
'GetImports' >> beam.FlatMap(lambda line: startsWith(line, keyword))
'PackageUse' >> beam.FlatMap(lambda line: packageUse(line, keyword))
```

The **TotalUse** operation corresponds to a shuffle-reduce because each package (key) is assigned a value (1) in the tuple creations and then the TotalUse operation sums the total usage of each package pair using the beam's combinePerKey method.

The operation that corresponds to a reduce is **Top_5** operation. This operation transforms and combines, then captures the 5 most used packages.

4. Run pipeline locally

```
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python [cloud-metens]$ python is_popular.py
WARNING:apache_beam.transforms.core:('No iterator is returned by the process method in %s.', <class 'apache_beam.transforms.combiners.TopPerBundle'>
WARNING:apache_beam.io.filebasedsource:Deleting existing files under target path matching: '-*-output_shards*)0$cd
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python [cloud-metens]$ ls /tmp
tmp..1M1Tz20FK vscode-ipc-bd93abf48f.sock
tmp..hp1DD0dfq vscode-ipc-4d0331d9-1211-4ble-814e-3c23c4dbc7ef.sock vscode-typescript1000
cloudcode-tempVmMe jupyter55af7a69f137230a3e8180b259d10800ea1 tmp..WuKaNlupQ1 vscode-ipc-5534ab72-a236-477c-b043-906ad6fcda3b.sock
cloudcode_tempFppGzZ output-00000-00001 tmp..SubpthfvD vscode-skaffold-events-logs
cloudcode_tempjTlRzZ tmp..SubpthfvD tmux-1000
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python [cloud-metens]$ cat /tmp/output-00000-00001
[('org', 45), ('org.apache', 44), ('org.apache.beam', 44), ('org.apache.beam.sdk', 43), ('org.apache.beam.sdk.transforms', 16)]
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python [cloud-metens]$
```

The output: `[('org', 45), ('org.apache', 44), ('org.apache.beam', 44), ('org.apache.beam.sdk', 43), ('org.apache.beam.sdk.transforms', 16)]` corresponds to the total times that the packages in each tuple in the list were used. The '**org**' Java package was used **45** times. The '**org.apache.beam.sdk.transforms**' package was used **16** times in the local java files.

5. Dataflow Lab #2 (Word count)

The names of the stages in the pipeline are **Read**, **Split**, **PairWithOne**, and **GroupAndSum**, **Format**, and **Write**.

Read: Read the text file[pattern] into a PCollection.

Split: Parse each line of input text into words. Using a regular expression and turns each word into a PCollection of strings.

PairWithOne: Map each string to a key value pair tuple ('string', 1).

GroupAndSum: Sums the total usage of each string into the tuple for that string.

Format: Output each tuple into a PCollection of strings "word: count" format for easier readability.

Write: Outputs each PCollection of "word: count" into a file with each string in its own line.

6. Run code locally

Since the wc command outputs Lines, Words, and Characters, we can specify words:

```
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-metens)$ cat outputs-00000-of-00001 | wc  
4784      9568    48944  
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-metens)$ cat outputs-00000-of-00001 | wc -w  
9568  
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-metens)$  
  
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-metens)$ cat outputs-00000-of-00001 | sort -k2,2nr -t':'|head -n 3  
the: 786  
I: 622  
and: 594  
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-metens)$
```

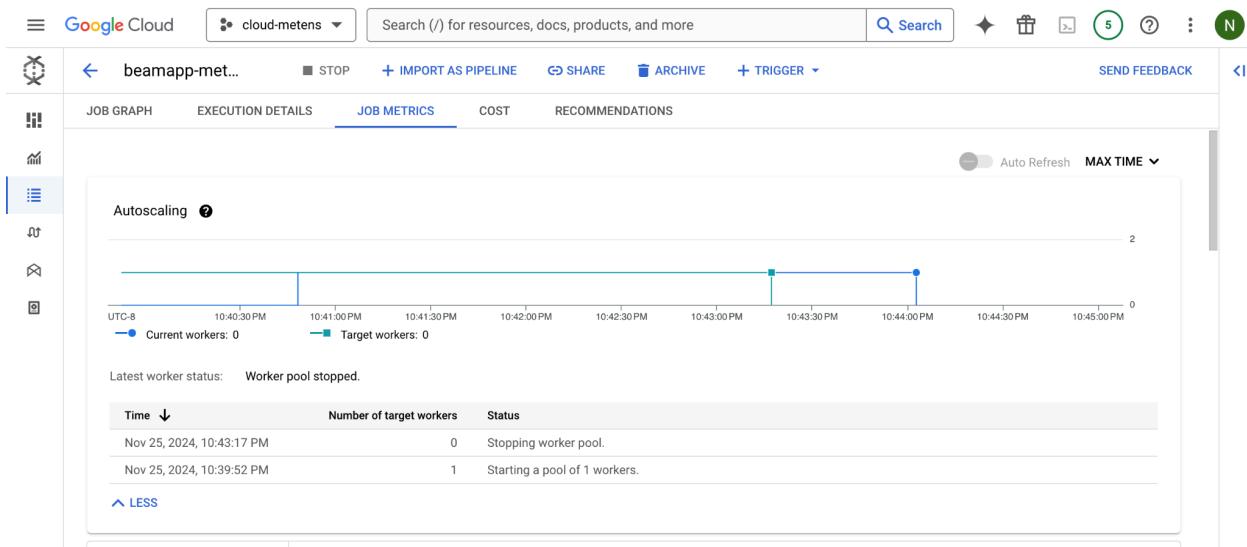
After adding `| 'lowercase' >> beam.Map(lambda x: x.lower())` to the pipeline, we get all words to lowercase:

```
(env) metens@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-metens)$ cat outputs-00000-of-00001 | sort -k2,2nr -t':'|  
head -n 3  
the: 908  
and: 738  
i: 622
```

9. Run code using Dataflow runner

The part of the job graph that has taken the longest time to complete was the **Write** operation which took 1 second but had 5 stages, the most out of the other operations.

The autoscaling graph showing when the worker was created and stopped:



The final write stage in the pipeline created 4 files:

Bucket details

cloud-metens

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Subject to object ACLs	Soft Delete

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE OBSERVABILITY INVENTORY REPORTS OPERATIONS

Folder browser

Buckets > cloud-metens > tmp > beamapp-metens-1126063946-642058-csn8mnd.1732603186.642214

CREATE FOLDER UPLOAD TRANSFER DATA OTHER SERVICES

Filter by name prefix only Filter objects and folders Show Live objects only

Name	Size	Type	Created	Storage class
pickled_main_session	4.2 KB	application/octet-stream	Nov 25, 2024, 10:39:46 PM	Standard
pipeline.pb	49.1 KB	application/octet-stream	Nov 25, 2024, 10:39:47 PM	Standard
submission_environment_depend...	2.3 KB	application/octet-stream	Nov 25, 2024, 10:39:47 PM	Standard
tmp-4fedaa9134054709-00000-of...	1.2 KB	application/octet-stream	Nov 25, 2024, 10:43:03 PM	Standard

12. View raw data from PubSub

```
(env) metens@cloudshell:~ (cloud-metens)$ gcloud pubsub subscriptions pull taxisub --auto-ack
DATA: {"ride_id":"32140e60-e0f8-4fe8-9b45-861f2013a3d","point_idx":371,"latitude":40.7657,"longitude":-73.96317,"timestamp":"2024-11-26T02:28:53.01506-05:00","meter_reading":0.516536,"meter_increment":0.028346457,"ride_status":"enroute","passenger_count":1}
MESSAGE_ID: 12752572141366057
ORDERING_KEY:
ATTRIBUTES: ts=2024-11-26T02:28:53.01506-05:00
DELIVERY_ATTEMPT:
ACK_STATUS: SUCCESS
```

14. Run Dataflow job from template

taxirides

JOB GRAPH EXECUTION DETAILS JOB METRICS COST RECOMMENDATIONS (1) AUTOSCALING

Job steps view Graph view

CLEAR SELECTION

Job info

Job name	taxirides
Job ID	2024-11-25_23_38_44-7428301323522357626
Job type	Streaming
Job status	Running
SDK version	Apache Beam SDK for Java 2.60.0
Job region	us-west1
Service zones	us-west1-a
Worker location	us-west1
Current workers	1
Latest worker status	Worker pool started.
Straggler status	No active straggler
Start time	November 25, 2024 at 11:38:44 PM GMT-8
Elapsed time	21 min 7 sec
Encryption type	Google-managed
Dataflow Prime	Disabled
Runner v2	Disabled
Streaming Engine	Disabled
Vertical Autoscaling	Disabled
Streaming Mode	Exactly once

15. Query data in BigQuery

The screenshot shows the Google Cloud BigQuery interface. The project is set to 'cloud-metens'. A search bar at the top right contains the query 'taxirides:realtime'. The main view displays the 'realtime' table from the 'taxirides' dataset. The table has 13 columns: point_idx, latitude, longitude, timestamp, meter_reading, meter_increments, ride_status, and passenger_count. The first 13 rows of data are shown, with the 14th row being a summary row. The interface includes tabs for Schema, Details, Preview, Table Explorer, Insights, Lineage, Data Profile, and Data Quality.

Assuming meter_count is the amount paid for the first ride:

The screenshot shows the Google Cloud BigQuery interface with a query editor. The query is:# SELECT SUM(passenger_count) FROM `cloud-metens.taxirides.realtime`
SELECT meter_reading, passenger_count FROM `cloud-metens.taxirides.realtime`
WHERE passenger_count > 0The results show the total passenger count and the first 12 rows of data from the realtime table. The results table has columns for meter_reading and passenger_count.

Google Cloud cloud-metens big

Explorer Untitled query

Search BigQuery resources

taxirides:realtime

Found 3 results.

SEARCH ALL PROJECTS

cloud-metens

taxirides

realtime

Untitled query

RUN SAVE DOWNLOAD SHARE SCHEDULE OPEN IN

4 -- SELECT * FROM `cloud-metens.taxirides.realtime`;

5 SELECT COUNT(*) AS total_rows FROM `cloud-metens.taxirides.realtime`;

Query results

JOB INFORMATION RESULTS CHART JSON EXECUTION DETAILS EXECUTION GRAPH

Row	total_rows
1	6143052

SAVE RESULT

Google Cloud cloud-metens big

Explorer Untitled query

Search BigQuery resources

taxirides:realtime

Found 3 results.

SEARCH ALL PROJECTS

cloud-metens

taxirides

realtime

Untitled query

RUN SAVE DOWNLOAD SHARE SCHEDULE OPEN IN MORE

3 COUNT(DISTINCT ride_id) AS total_rides,

4 SUM(passenger_count) AS total_passengers,

5 SUM(meter_reading) AS total_revenue

6 FROM

7 taxirides.realtime

8 WHERE

9 | ride_status = 'droppedoff'

Query results

JOB INFORMATION RESULTS CHART JSON EXECUTION DETAILS EXECUTION GRAPH

Row	minute	total_rides	total_passengers	total_revenue
1	00:00	387	664	4563.119991899...
2	00:01	371	642	4663.039994999...
3	00:02	343	588	4340.9399943...
4	00:03	378	635	4477.260008599...
5	00:04	396	678	5373.0399974...
6	00:05	345	573	4246.4300047...
7	00:06	401	674	4838.6600063...
8	00:07	414	723	4977.6700009...
9	00:08	378	652	5028.349994599...
10	00:09	408	725	5026.039999400...

SAVE RESULTS EXPLORE DATA

Results per page: 50 1 - 44 of 44 < > >>

16. Data visualization

