

09.1g: BigQuery, BigLake	1
Create dataset	1
Query data	2
Query data	4
09.2g: Jupyter Notebooks	5
BigQuery query	5
Run queries	6
Mobility	6
Airport traffic	7
Mortality	7
Run example queries	8
Write queries	10
09.3g: Dataproc	11
Run computation	11
Run computation again	12
09.4g: Dataflow	13
Beam code	13
Run pipeline locally	14
Dataflow Lab #2 (Word count)	14
Run code locally	14
Run code using Dataflow runner	15
View raw data from PubSub	17
Run Dataflow job from template	18
Query data in BigQuery	18
Data visualization	20

09.1g: BigQuery, BigLake

Create dataset

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

cloud-Tran-hali5 big qu Search

Explorer + ADD

Viewing workspace resources.
SHOW STARRED ONLY

cloud-tran-hali5

External connections

yob

yob_native_table

SHOW MORE

odin... File Edit Format View Help

hali5

Windows (CRLF) UTF-8

yob_native_table QUERY SHARE COPY SNAPSHOT DELETE REFRESH

SCHEMA DETAILS PREVIEW LINEAGE

Table ID cloud-tran-hali5.yob.yob_native_table

Created May 29, 2023, 10:53:00 AM UTC-7

Last modified May 29, 2023, 10:53:00 AM UTC-7

Table expiration NEVER

Data location us-west1

Default collation

Default rounding mode ROUNDING_MODE_UNSPECIFIED

Case insensitive false

Description

Labels

Primary key(s)

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

Total physical bytes 0 B

Active physical bytes 0 B

Long term physical bytes 0 B

Time travel physical bytes 0 B

Query data

Screenshot the query results and include it in your lab notebook

yob_native_table *Untitled

RUN SAVE SHARE SCHEDULE MORE

Query completed.

```
1 SELECT name, count
2 FROM `cloud-tran-hali5.yob.yob_native_table`
3 WHERE gender='F'
4 ORDER BY count DESC
5 LIMIT 20
```

odin... File Edit Format View Help

hali5

The screenshot shows the Google Cloud BigQuery console. On the left, the Explorer pane shows the workspace resources, including 'cloud-tran-hali5' and 'yob'. The main pane displays a query titled 'Untitled' with the following SQL:

```
1 SELECT name, count
2 FROM `cloud-tran-hali5.yob.yob_native_table`
```

The query results are displayed in a table with 20 rows. The columns are 'name' and 'count'.

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

A small terminal window titled 'odin...' is overlaid on the right side of the console, showing the text 'hali5'.

Screenshot your results and include it in your lab notebook

```
cloud-tran-hali5 x
hali5@cloudshell:~ (cloud-tran-hali5)$ bq query "SELECT name, count FROM [cloud-tran-hali5.yob.yob_native_table] WHERE gender='M' ORDER BY count ASC LIMIT 10"
I0529 18:16:57.609214 139793241921344 bigquery_client.py:730] There is no apilog flag so non-critical logging is disabled.
+-----+
| name | count |
+-----+
| Aari | 5 |
| Aaliyah | 5 |
| Aadian | 5 |
| Aaroh | 5 |
| Aarit | 5 |
| Aadiv | 5 |
| Aadhi | 5 |
| Aarohan | 5 |
| Aariyan | 5 |
| Amer | 5 |
+-----+
hali5@cloudshell:~ (cloud-tran-hali5)$
```

Screenshot your results and include it in your lab notebook

```

cloud-tran-hali5 x
hali5@cloudshell:~ (cloud-tran-hali5)$ bq shell
I0529 18:28:17.552992 140271381657408 bigquery_client.py:730] There is no apilog flag so non-critical logging is disabled.
Welcome to BigQuery! (Type help for more information.)
cloud-tran-hali5> SELECT name, count FROM [cloud-tran-hali5.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-tran-hali5>

```

Screenshot your results and include it in your lab notebook

```

cloud-tran-hali5 x
hali5@cloudshell:~ (cloud-tran-hali5)$ bq shell
I0529 18:34:14.258896 140603490658112 bigquery_client.py:730] There is no apilog flag so non-critical logging is disabled.
Welcome to BigQuery! (Type help for more information.)
cloud-tran-hali5> SELECT name, count FROM [cloud-tran-hali5.yob.yob_native_table] WHERE name='Hali'
+-----+-----+
| name | count |
+-----+-----+
| Hali | 24 |
+-----+-----+
cloud-tran-hali5>

```

Query data

Screenshot the query results and include it in your lab notebook

The screenshot shows a BigQuery web interface. The main editor, titled 'Untitled 2', contains the following SQL query:

```

1 SELECT name, count
2 FROM `cloud-tran-hali5.yob.yob_biglake_table`
3 WHERE gender='F'
4 ORDER BY count ASC
5 LIMIT 20

```

Below the query, there are buttons for 'RUN', 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE'. A status message on the right indicates 'Query completed.' with a green checkmark.

In the foreground, there is a small terminal window titled 'odinl...' with a menu bar (File, Edit, Format, View, Help) and the text 'hali5' entered.

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the Explorer pane shows the project 'cloud-tran-hali5' with external connections 'us-west1.biglake' and 'yob'. The 'yob' dataset contains tables 'yob_biglake_table' and 'yob_native_table'. The main editor shows a query in 'Untitled 2' with the following SQL:

```
2 FROM `cloud-tran-hali5.yob.yob_biglake_table`
3 WHERE gender='F'
```

The query has been executed successfully, as indicated by the 'Query completed' status. The 'Query results' pane shows a table with 20 rows and 2 columns: 'name' and 'count'. All counts are 5. An 'odin' terminal window is overlaid on the results, displaying the text 'hal15'.

Row	name	count
1	Aarshi	5
2	Aaniyah	5
3	Aaryah	5
4	Aashirya	5
5	Aalimah	5
6	Aarielle	5
7	Aarabella	5
8	Aayra	5
9	Aarti	5
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

09.2g: Jupyter Notebooks

BigQuery query

How much less data does this query process compare to the size of the table?

There is roughly 7 times less data in this query process compared to the size of the table. The query will process 3.05 GB compared to the size of the table which is 21.94 GB.

How many twins were born during this time range?

There were 375362 twins born during this time range.

How much lighter on average are they compared to single babies?

On average, twins are roughly 2 pounds lighter than single babies.

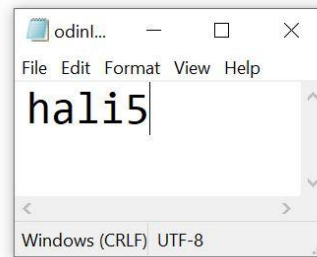
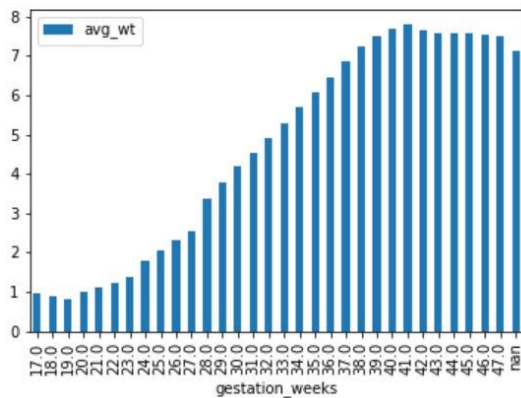
Run queries

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

The two strongest predictors for a newborn baby's weight are plurality and gestation weeks.

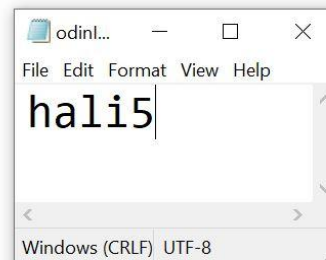
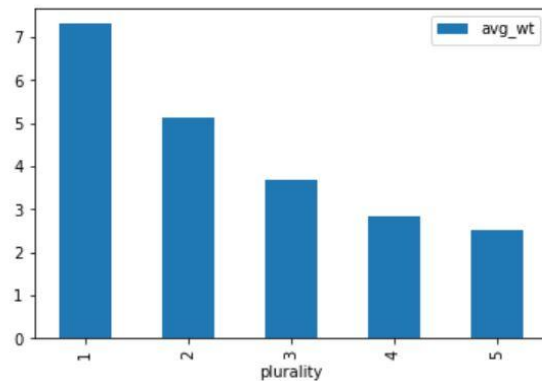
```
[8]: df = get_distinct_values('gestation_weeks')
df.plot(x='gestation_weeks', y='avg_wt', kind='bar')

[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc2d2ed6d90>
```



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

[6]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc2d806e490>
```



Mobility

What day saw the largest spike in trips to grocery and pharmacy stores?

2020/03/13 saw the largest spike to grocery and pharmacy stores.

On the day the stay-at-home order took effect (3/23/2020), what was the total impact on workplace trips?

The total impact on workplace trips was -49.

Airport traffic

Which three airports were impacted the most in April 2020 (the month when lockdowns became widespread)?

Detroit Metropolitan Wayne County, McCarran International, and San Francisco International were impacted the most in April 2020.

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

The three airports from the previous question Detroit Metropolitan Wayne County, McCarran International, and San Francisco International were impacted the most in August 2020.

Mortality

What table and columns identify the place name, the starting date, and the number of excess deaths from COVID-19?

The `excess_deaths` table and its corresponding columns *placename*, *start_date*, and *excess_deaths* identify this.

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

The `us_counties` table and its corresponding columns *date*, *county*, and *deaths* identify this.

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

The `us_states` table and its corresponding columns *date*, *state_name*, and *deaths* identify this.

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

The `mask_use_by_county` table and its corresponding columns `country_fips_code` and `always` identify this.

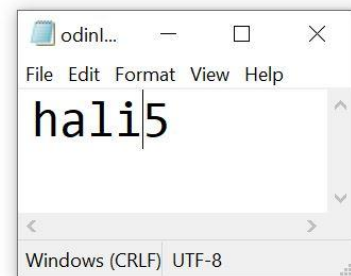
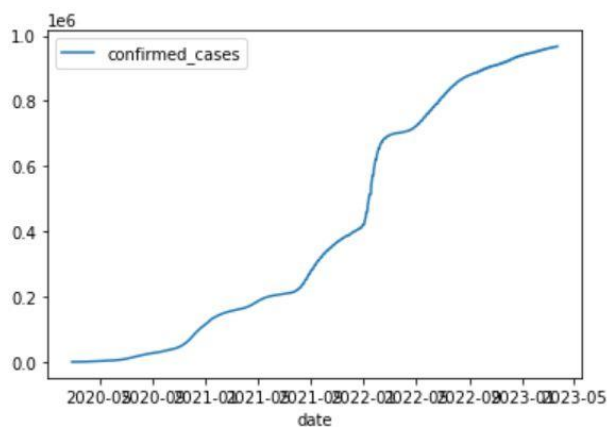
Run example queries

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

```
[16]: query_string = """
      SELECT date, confirmed_cases
      FROM `bigquery-public-data.covid19_nyt.us_states`
      WHERE state_name = 'Oregon'
      ORDER BY date ASC
      """

      from google.cloud import bigquery
      df = bigquery.Client().query(query_string).to_dataframe()
      df.plot(x='date', y='confirmed_cases', kind='line')
```

```
[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc2d13b6750>
```



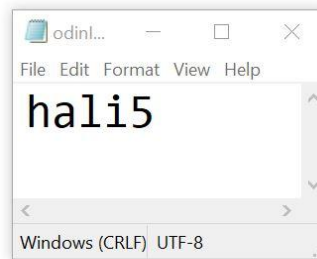
From within your Jupyter notebook, run the query and write code that shows the first 10 states that reached 1000 deaths from COVID-19. Take a screenshot for your lab notebook.


```
[18]: 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
LIMIT 10
"""

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

```
[18]:
```

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



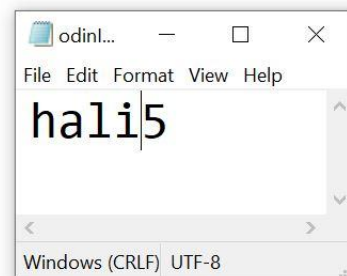
Take a screenshot for your lab notebook of the Top 5 counties and the states they are located in.

```
[19]: query_string = """
SELECT DISTINCT mu.county_fips_code, mu.always, ct.county, ct.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
LIMIT 5
"""

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

```
[19]:
```

	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



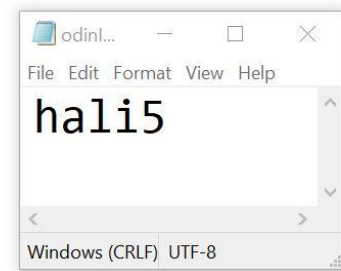
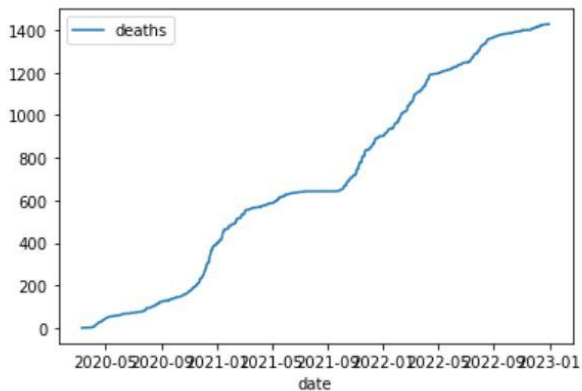
Write queries

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

```
[21]: query_string = """
      SELECT date, deaths
      FROM `bigquery-public-data.covid19_nyt.us_counties`
      WHERE county='Multnomah'
      ORDER BY date ASC
      """

      from google.cloud import bigquery
      df = bigquery.Client().query(query_string).to_dataframe()
      df.plot(x='date', y='deaths', kind='line')
```

```
[21]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc2d0e3bf90>
```

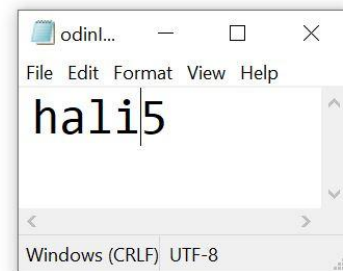
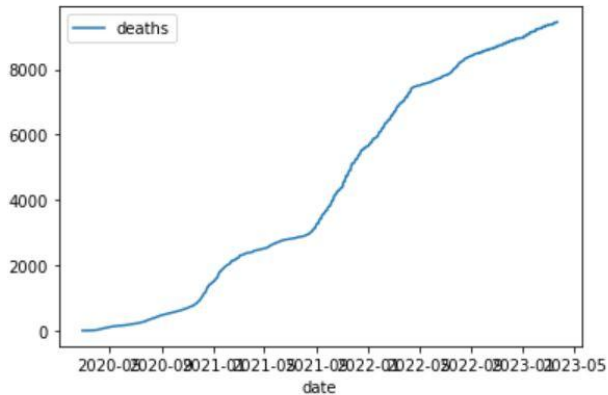


Plot the results and take a screenshot for your lab notebook.

```
[24]: query_string = """
SELECT date, deaths
FROM `bigquery-public-data.covid19_nyt.us_states`
WHERE state_name='Oregon'
ORDER BY date ASC
"""

from google.cloud import bigquery
df = bigquery.Client().query(query_string).to_dataframe()
df.plot(x='date', y='deaths', kind='line')
```

```
[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc2d05f4e10>
```



09.3g: Dataproc

Run computation

How long did the job take to execute?

The job took 1 minute, 13 seconds to execute.

Examine output.txt and show the estimate of π calculated.

```

hali5@cloudshell:~ (cloud-tran-hali5) $ cat output.txt
Job [ab15ef93313348c69960ba4a2d068c4a] submitted.
Waiting for job output...
23/05/30 05:58:22 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/05/30 05:58:22 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/05/30 05:58:22 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/05/30 05:58:22 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
23/05/30 05:58:22 INFO org.sparkproject.jetty.util.log: Logging initialized @3767ms to c
23/05/30 05:58:22 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; bui
23/05/30 05:58:22 INFO org.sparkproject.jetty.server.Server: Started @3873ms
23/05/30 05:58:22 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerCo
23/05/30 05:58:23 INFO org.apache.hadoop.yarn.client.RMPProxy: Connecting to ResourceMana
23/05/30 05:58:23 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application
23/05/30 05:58:24 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not four
23/05/30 05:58:24 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to fir
23/05/30 05:58:26 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
23/05/30 05:58:27 INFO org.apache.hadoop.yarn.client.RMPProxy: Connecting to ResourceMana
23/05/30 05:58:29 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gc
already exists with desired state.
Pi is roughly 3.1416515514165155

```

The estimate of Pi calculated was 3.1416515514165155.

Run computation again

How long did the job take to execute? How much faster did it take?

The job took 37 seconds to execute. It was nearly 2 times faster than the previous.

Examine output2.txt and show the estimate of π calculated.

```

hali5@cloudshell:~ (cloud-tran-hali5) $ cat output2.txt
Job [656ee2f82afa43cfba775904ba4c847b] submitted.
Waiting for job output...
23/05/30 06:15:45 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/05/30 06:15:45 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster

23/05/30 06:15:46 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/05/30 06:15:46 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
23/05/30 06:15:46 INFO org.sparkproject.jetty.util.log: Logging initialized @3232ms to
23/05/30 06:15:46 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; bu
23/05/30 06:15:46 INFO org.sparkproject.jetty.server.Server: Started @3322ms
23/05/30 06:15:46 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerCo
23/05/30 06:15:47 INFO org.apache.hadoop.yarn.client.RMPProxy: Connecting to ResourceMan
23/05/30 06:15:47 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Applicatio
23/05/30 06:15:47 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not fou
23/05/30 06:15:47 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to fi
23/05/30 06:15:48 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
23/05/30 06:15:49 INFO org.apache.hadoop.yarn.client.RMPProxy: Connecting to ResourceMan
23/05/30 06:15:51 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gc
already exists with desired state.
Pi is roughly 3.1416726314167263

```

The estimate of Pi calculated was 3.1416726314167263.

09.4g: Dataflow

Beam code

Where is the input taken from by default?

The input by default is taken from

```
../javahelp/src/main/java/com/google/cloud/training/dataanalyst/  
javahelp/
```

Where does the output go by default?

The output by default is taken from `/tmp/output`

Examine both the `getPackages()` function and the `splitPackageName()` function. What operation does the 'PackageUse' transform implement?

The PackageUse transform takes a line of a code and a keyword and applies the `getPackages` function to extract the package name. It then yields (package, 1) pairs for each package used in the line of code.

Look up Beam's `CombinePerKey`. What operation does the `TotalUse` operation implement?

The `TotalUse` implements the operation to calculate the total count of each package used in the Java source code by taking the output from the `PackageUse` transformation. It utilizes `CombinePerKey(sum)` to calculate the total count of packages.

Which operations correspond to a "Map"?

`GetJava`, `GetImports`, and `PackageUse` operations correspond to a Map.

Which operation corresponds to a "Shuffle-Reduce"?

`CombinePerKey` operation corresponds to Shuffle-Reduce.

Which operation corresponds to a "Reduce"?

Top_5 operation corresponds to Reduce.

Run pipeline locally

Take a screenshot of its contents

```
(env) hali5@cloudshell:/tmp (cloud-tran-hali5)$ cat output-00000-of-00001  
[('org', 45), ('org.apache', 44), ('org.apache.beam', 44), ('org.apache.beam.sdk', 43), ('org.apache.beam.sdk.transforms', 16)]  
(env) hali5@cloudshell:/tmp (cloud-tran-hali5)$
```

Explain what the data in this output file corresponds to based on your understanding of the program.

The data in the output file corresponds to the top 5 Java packages that are used the most frequently in the Java source code that is inputted into the pipeline.

Dataflow Lab #2 (Word count)

What are the names of the stages in the pipeline?

The names of the states in the pipeline are Read, Split, PairWithOne, GroupAndSum, and Format.

Describe what each stage does.

Read: Reads the input text file

Split: Applies WordExtractingDoFn function transformation to split each line of the input into individual words.

PairWithOne: Uses Map transform to convert each individual word to a key value pair.

GroupAndSum: Uses the CombinePerKey transformation to group the key-value pairs by key and sum the corresponding values.

Format: Uses the MapTuple function to format the counts into a PCollection of strings. It formats the key-value pairs.

Write: Writes the PCollection to an output file.

Run code locally

Use wc with an appropriate flag to determine the number of unique words in King Lear.


```
(env) hali5@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-tran-hali5)$ wc -l outputs-00000-of-00001
4784 outputs-00000-of-00001
(env) hali5@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-tran-hali5)$
```

Use sort with appropriate flags to perform a *numeric* sort on the *key field* containing the count for each word in *descending* order. Pipe the output into head to show the top 3 words in King Lear and the number of times they appear.

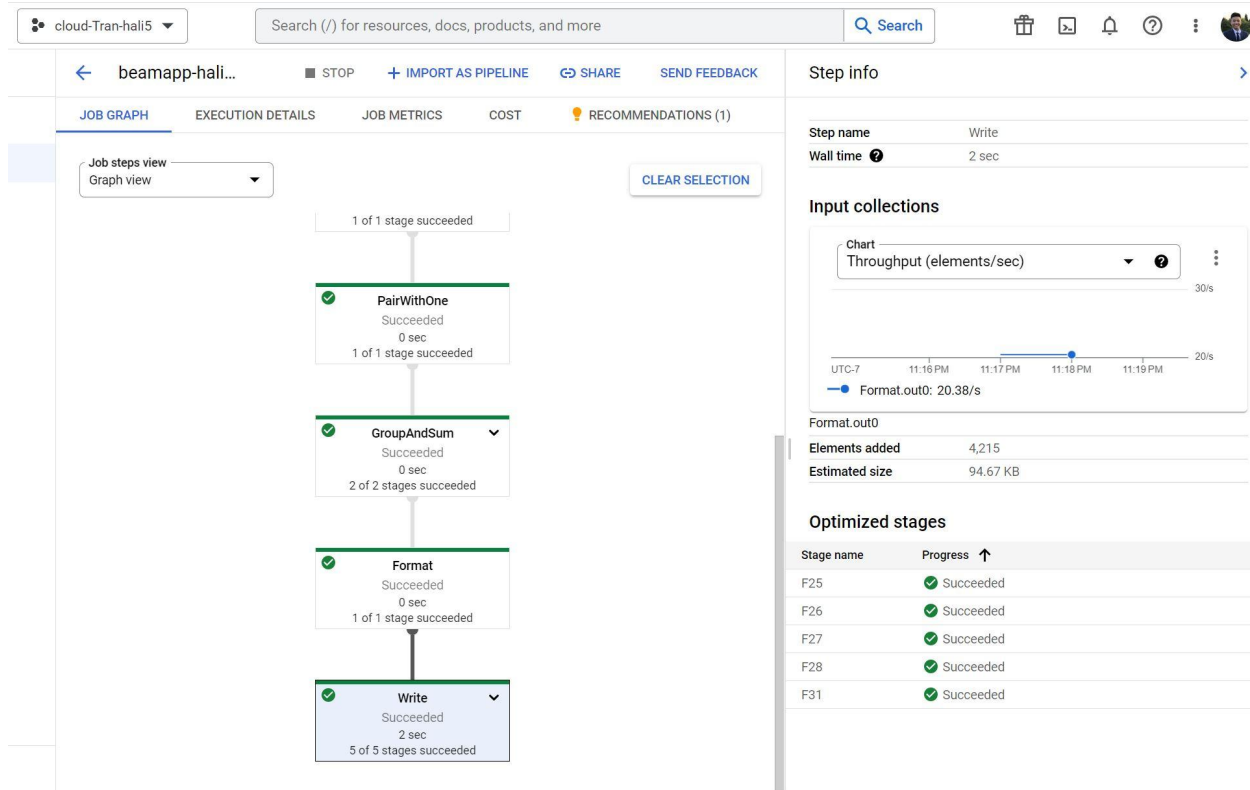
```
(env) hali5@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-tran-hali5)$ sort -k2 -nr outputs-00000-of-00001 | head -n 3
the: 786
I: 622
and: 594
(env) hali5@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-tran-hali5)$
```

Use the previous method to show the top 3 words in King Lear, case-insensitive, and the number of times they appear.

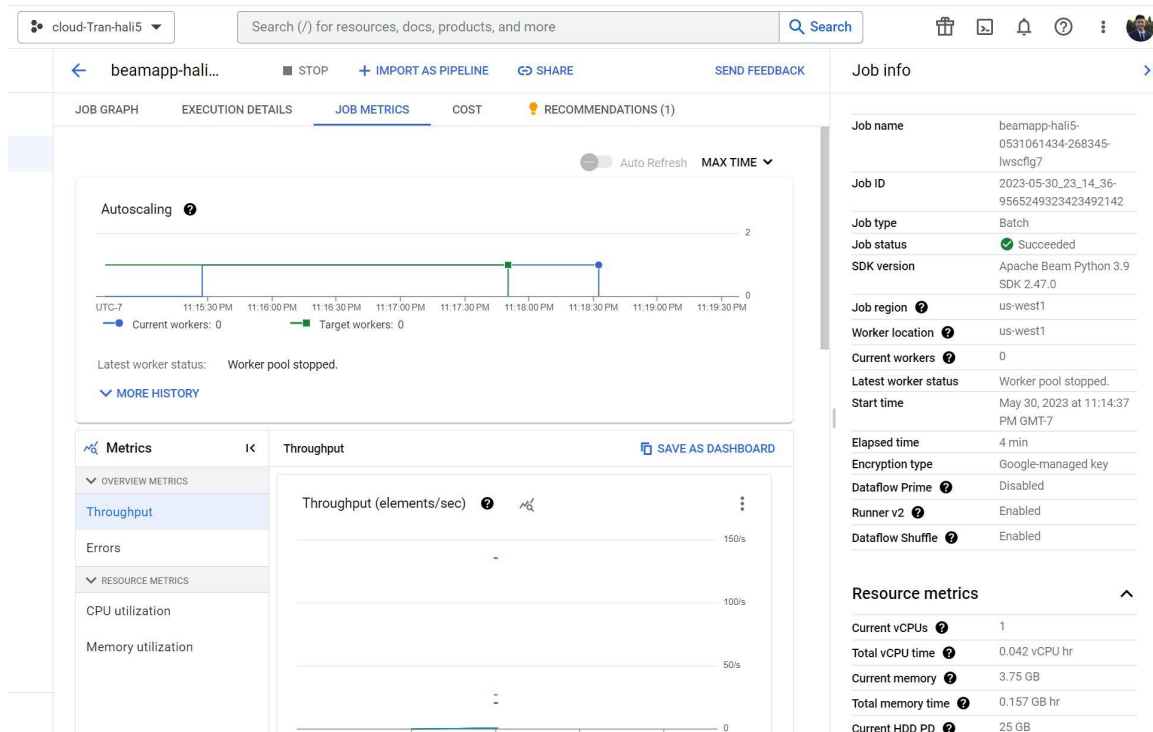
```
(env) hali5@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-tran-hali5)$ sort -k2 -nr outputs-00000-of-00001 | head -n 3
the: 908
and: 738
i: 622
(env) hali5@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-tran-hali5)$
```

Run code using Dataflow runner

The part of the job graph that has taken the longest time to complete.



The auto scaling graph shows when the worker was created and stopped.



Examine the output directory in Cloud Storage. How many files has the final write stage in the pipeline created?

The final write stage in the pipeline created a single file named outputs.

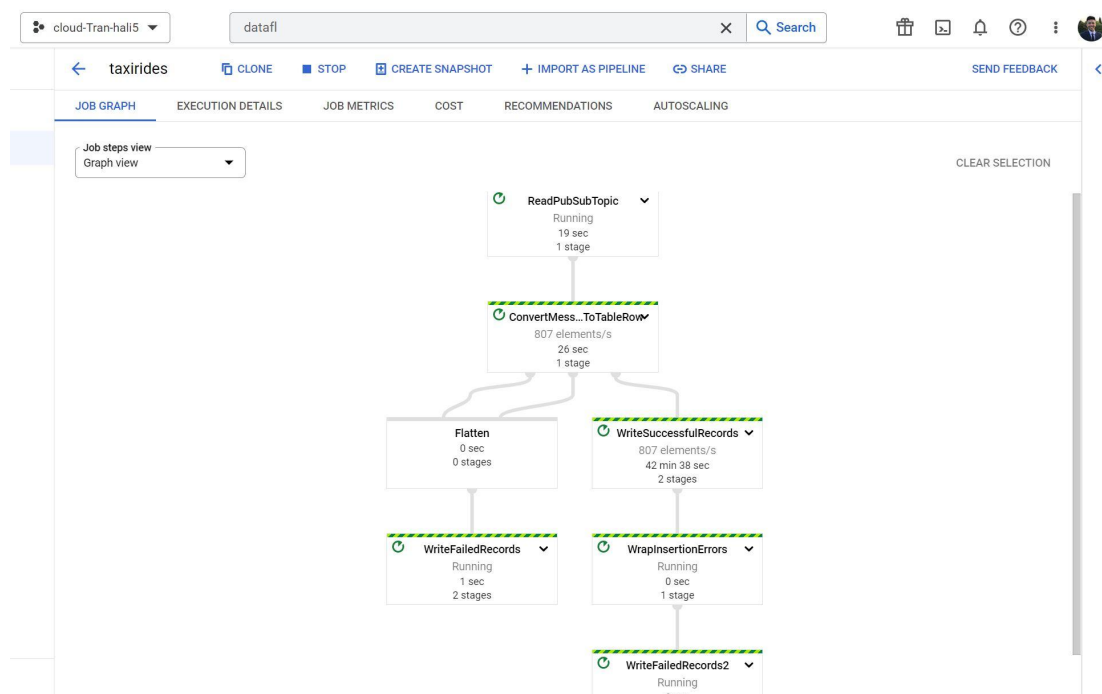
View raw data from PubSub

Take a screenshot listing the different fields of this object.

```
hali5@cloudshell:~ (cloud-tran-hali5)$ gcloud pubsub subscriptions pull taxisub --auto-ack
DATA: {"ride_id":"7f9aba63-f93d-4cf9-89eb-23770b940bd9","point_idx":683,"latitude":40.736850000000004,"longitude":-73.931220000000001,"timestamp":"2023-05-31T02:33:07.46506-04:00","meter_reading":12.459263,"meter_increment":0.018241966,"ride_status":"enroute","passenger_count":1}
MESSAGE_ID: 7861017045290965
ORDERING_KEY:
ATTRIBUTES: ts=2023-05-31T02:33:07.46506-04:00
DELIVERY_ATTEMPT:
ACK_STATUS: SUCCESS
hali5@cloudshell:~ (cloud-tran-hali5)$
```

Run Dataflow job from template

Take a screenshot of the pipeline that includes its stages and the number of elements per second being handled by individual stages.



Query data in BigQuery

Take a screenshot showing the number of passengers and the amount paid for the first ride

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, a project selector (cloud-Tran-hali5), a search bar (bigqu), and various utility icons. The main workspace displays the 'realtime' table. A message indicates it is a partitioned table. The 'PREVIEW' tab is active, showing a single row of data. The table schema includes columns for ride_id, point_idx, latitude, longitude, timestamp, meter_reading, meter_increment, ride_status, and passenger_count.

Row	ride_id	point_idx	latitude	longitude	timestamp	meter_reading	meter_increment	ride_status	passenger_count
1	8426bac6-e5bb-4128-8a09-1c5ec3e3f939	318	40.756...	-73.95410000000000...	2023-05-31 0...	7.378645	0.023203285	enroute	1

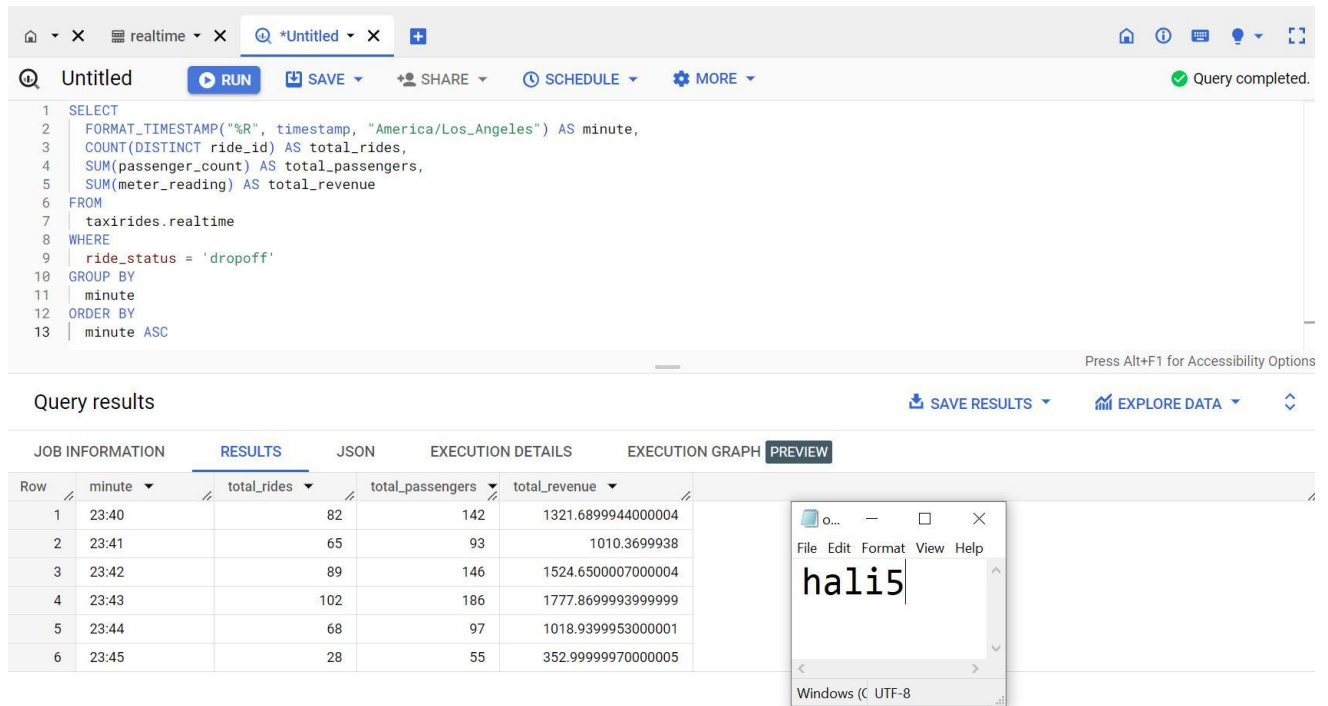
Take a screenshot showing the estimated number of rows in the table.

The screenshot shows the Google Cloud BigQuery console interface. The 'realtime' table is selected in the Explorer on the left. The 'PREVIEW' tab is active, displaying the first 20 rows of data. The table schema includes columns for ride_id, point_idx, latitude, longitude, timestamp, meter_reading, meter_increment, and ride_status. The bottom of the interface shows 'Results per page: 50' and '1 - 50 of many'.

Row	ride_id	point_idx	latitude	longitude	timestamp	meter_reading	meter_increment	ride_status
1	8426bac6-e5bb-4128-8a09-1c5...	318	40.7567800...	-73.9541000...	2023-05-31 06:40:12.941050 U...	7.378645	0.023203285	enroute
2	7be2787a-3aff-4ec5-86b4-446...	154	40.7408600...	-73.99429	2023-05-31 06:40:28.788200 U...	12.353208	0.08021563	enroute
3	adf79f96-cbff-4c57-8869-c5c1...	225	40.72527	-73.9963700...	2023-05-31 06:40:29.101920 U...	6.104022	0.027128987	enroute
4	d8f2730c-abb3-4829-8412-d38...	161	40.74034	-73.98202	2023-05-31 06:40:15.457420 U...	3.237167	0.020106627	enroute
5	7386578a-be93-4144-9f14-c88...	397	40.7520300...	-73.98617	2023-05-31 06:40:17.795840 U...	10.031263	0.025267666	enroute
6	9ae431c6-0349-4e6c-a7c1-a32...	882	40.81181	-73.9661900...	2023-05-31 06:40:26.170080 U...	18.194818	0.020629046	enroute
7	7723a60f-792d-427e-92ce-798...	460	40.6974800...	-74.0143500...	2023-05-31 06:40:28.958590 U...	9.679754	0.021042945	enroute
8	b3e3e5b2-63a1-4655-b48b-0d3...	412	40.70172	-73.9554900...	2023-05-31 06:40:26.055490 U...	12.159838	0.02951417	enroute
9	2ad65b64-2842-40ed-b97a-4b...	318	40.71226	-74.01439	2023-05-31 06:40:07.790070 U...	7.0772076	0.02225537	enroute
10	1aed8dfe-41ef-439a-9d37-159...	79	40.7579600...	-73.98554	2023-05-31 06:40:12.198030 U...	2.2918375	0.029010601	enroute
11	5d0bf3dd-1bc3-4077-90c3-f9f4...	214	40.74906	-74.00318	2023-05-31 06:40:15.703130 U...	5.445196	0.025444841	enroute
12	1a57d33a-09af-4f8f-9f59-0e7f...	52	40.73602	-73.98519	2023-05-31 06:40:29.614240 U...	1.6248411	0.031246943	enroute
13	073a2e07-b750-4695-a364-c97...	107	40.7215400...	-73.94565	2023-05-31 06:40:15.545160 U...	5.667373	0.052966103	enroute
14	796e46bb-a9bd-47df-9e5d-a67...	1394	40.66483	-73.99687	2023-05-31 06:40:15.990560 U...	25.54741	0.018326692	enroute
15	1b03dced-29ce-4daa-b05c-fc6...	132	40.74105	-73.98152	2023-05-31 06:40:08.923420 U...	3.8879795	0.02945439	enroute
16	9f27dcec-4bd7-419a-8ca1-f2d...	401	40.74765	-73.9702500...	2023-05-31 06:40:28.860800 U...	10.917737	0.027226277	enroute
17	9a300759-8c4a-4b6c-9f71-665...	222	40.71685	-73.9833300...	2023-05-31 06:40:15.057660 U...	5.301492	0.023880595	enroute
18	f4dd7b59-5864-46a8-94de-759...	1014	40.76436	-73.90291	2023-05-31 06:40:28.653780 U...	29.529036	0.029121337	enroute
19	74bdeb86-85f5-43c1-9c02-e0b...	847	40.65084	-73.9762100...	2023-05-31 06:40:12.199090 U...	19.984003	0.023593863	enroute
20	cf29aa86-344b-47e6-b877-f07...	1353	40.6704400...	-73.9973500...	2023-05-31 06:40:28.826870 U...	31.442581	0.023239158	enroute

Take
a

screenshot showing the per-minute number of rides, passengers, and revenue for the data collected



Data visualization

Take a screenshot showing the plot for your data for your lab notebook

