Addison Wurtz
CS 530
notebooks/Week9

Lab Notebook – Week 9

Table of Contents:

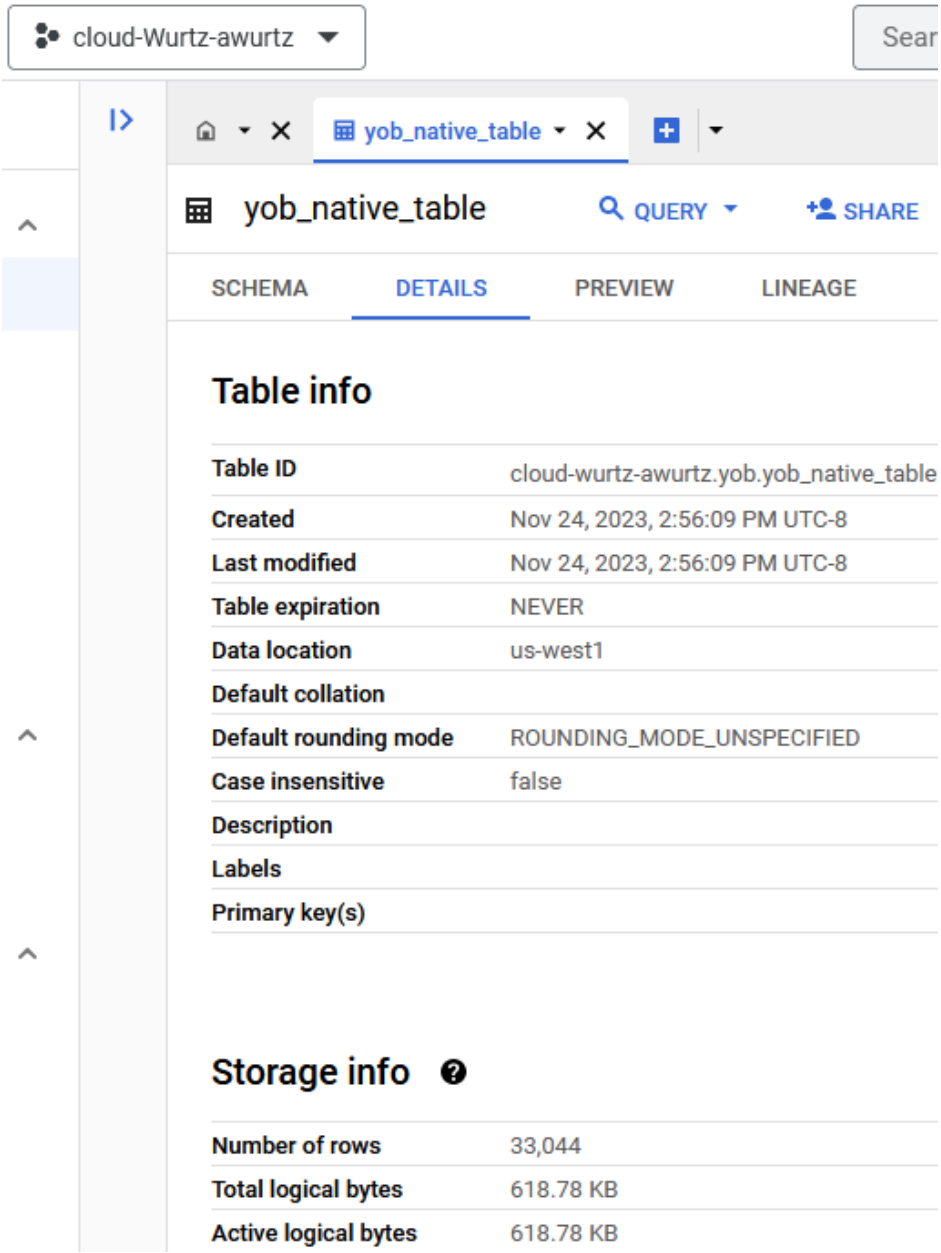
Addison Wurtz
CS 530
notebooks/Week9

Addison Wurtz
CS 530
notebooks/Week9

09.1g: Big Query, BigLake

BigQuery

Create dataset



Addison Wurtz
CS 530
notebooks/Week9

Query data

```
SELECT name, count
FROM `cloud-wurtz-awurtz.yob.yob_native_table`
WHERE gender='F'
ORDER BY count DESC
LIMIT 20
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW

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

Load more

Addison Wurtz
CS 530
notebooks/Week9

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ bq query
+---------+-------+
|  name   | count |
+---------+-------+
| Aari    |     5 |
| Aaliyah |     5 |
| Aadian  |     5 |
| Aaroh   |     5 |
| Aarit   |     5 |
| Aadiv   |     5 |
| Aadhi   |     5 |
| Aarohan |     5 |
| Aariyan |     5 |
| Aamer   |     5 |
+---------+-------+
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$
```

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ bq query "SELECT name
, gender, count FROM [cloud-wurtz-awurtz.yob.yob_native_table]
WHERE name='Addison' ORDER BY count DESC"

+---------+--------+-------+
|  name   | gender | count |
+---------+--------+-------+
| Addison | F      |  6950 |
| Addison | M      |   129 |
+---------+--------+-------+
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$
```

Addison Wurtz
CS 530
notebooks/Week9

BigLake

Query Data

```
SELECT name, count
FROM `cloud-wurtz-awurtz.yob.yob_biglake_table`
WHERE gender='F'
ORDER BY count ASC
LIMIT 20
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW

Row	name	count
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

Addison Wurtz
CS 530
notebooks/Week9

09.2g: Jupyter Notebooks

BigQuery query

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

The table is 21.94 GB. This query processes 3.05 GB. It processes 18.89 GB less data.

How many twins were born during this time range?

375,362

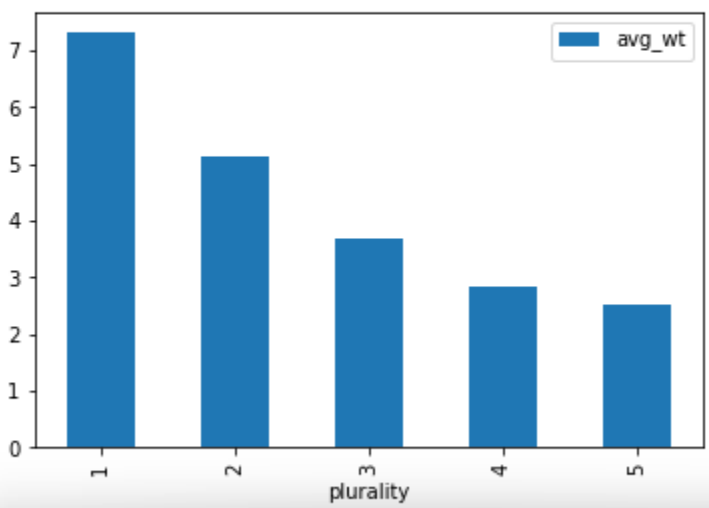
How much lighter on average are they compared to single babies?

The twins are 2.17 pounds lighter than single babies on average.

Run queries

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

[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc310440990>
```



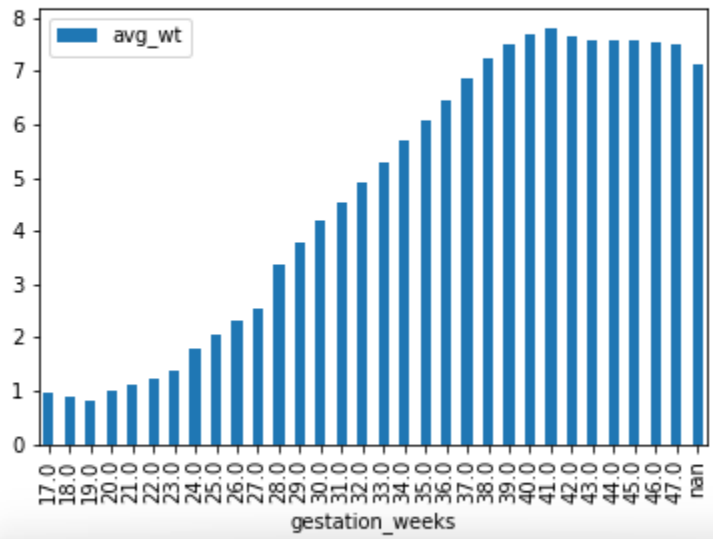
[9]:

ODIN ID awurtz

Addison Wurtz
CS 530
notebooks/Week9

[10]:
```
df = get_distinct_values('gestation_weeks')
df.plot(x='gestation_weeks', y='avg_wt', kind='bar')
```

[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc30af957d0>
```



[11]: ODIN ID awurtz

Mobility

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

2020-03-13

On the day stay-at-home order took effect (3/23/3030), what was the total impact on workplace trips?

-49%

Addison Wurtz
CS 530
notebooks/Week9

Airport traffic

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

Detroit Metropolitan Wayne County, McCarran International, San Francisco International

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

McCarran International, Detroit Metropolitan Wayne County, San Francisco International

---

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 contains the columns `placename`, `start_date`, and `excess_deaths`.

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

The `us_counties` table contains the columns `date`, `county`, and `deaths`

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

The `us_states` table contains the columns `date`, `state_name`, and `confirmed_cases` columns.

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 contains the columns `county_fips_code` and `always`.

Addison Wurtz
CS 530
notebooks/Week9

Run example queries

Confirmed cases in Oregon

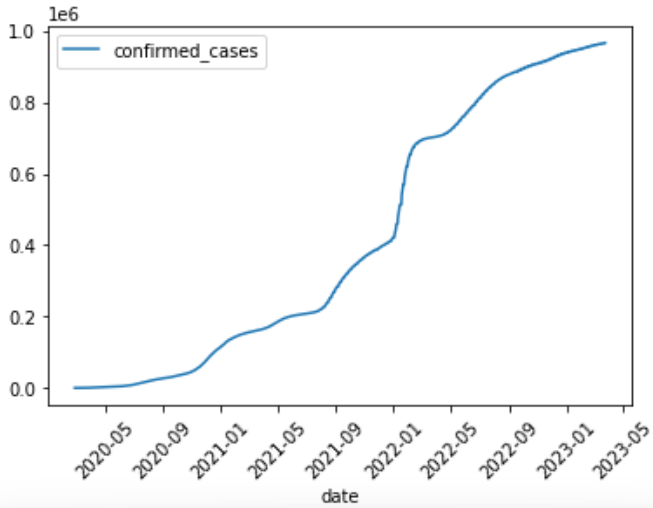
[16]:
```
query_string = """
SELECT date, confirmed_cases
FROM `bigquery-public-data.covid19_nyt.us_states`
WHERE state_name = 'Oregon'
ORDER BY date ASC
"""

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

[16]:

	date	confirmed_cases
0	2020-02-28	1
1	2020-02-29	1
2	2020-03-01	2
3	2020-03-02	2
4	2020-03-03	2

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

[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc30ac582d0>
```



[ ]:

ODIN ID awurtz

Addison Wurtz
CS 530
notebooks/Week9

Date when states reached 1000 deaths

[19]:
```
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).to_dataframe()
df.head(10)
```

[19]:

	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

ODIN ID awurtz

Addison Wurtz
CS 530
notebooks/Week9

Mask usage per county

[21]:
```
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
"""

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

[21]:

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

[ ]:

ODIN ID awurtz

Addison Wurtz
CS 530
notebooks/Week9

Write queries

Deaths in Multnomah county

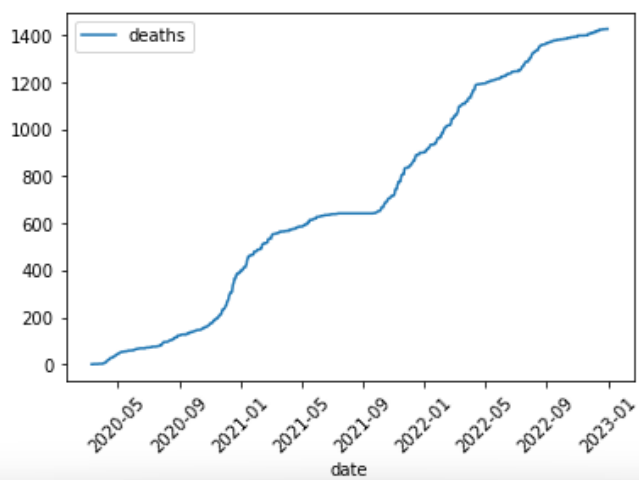
[22]:
```
query_string = """
SELECT date, deaths
FROM `bigquery-public-data.covid19_nyt.us_counties`
WHERE county = 'Multnomah'
ORDER BY date ASC
"""

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

[22]:

	date	deaths
0	2020-03-10	0
1	2020-03-11	0
2	2020-03-12	0
3	2020-03-13	0
4	2020-03-14	1

[23]:
```
df.plot(x='date', y='deaths', kind='line', rot=45)
```

[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc310439850>
```



[ ] ODIN ID awurtz

Addison Wurtz
CS 530
notebooks/Week9

Deaths in Oregon

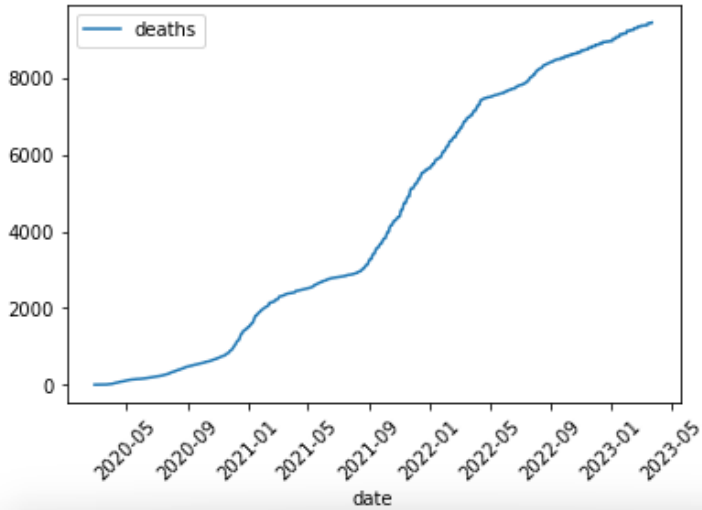
[24]:
```
query_string = """
SELECT date, deaths
FROM `bigquery-public-data.covid19_nyt.us_states`
WHERE state_name = 'Oregon'
ORDER BY date ASC
"""

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

[24]:

	date	deaths
0	2020-02-28	0
1	2020-02-29	0
2	2020-03-01	0
3	2020-03-02	0
4	2020-03-03	0

[25]:
```
df.plot(x='date', y='deaths', kind='line', rot=45)
```

[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc30a680d50>
```



[ ]:

ODIN ID awurtz

Addison Wurtz
CS 530
notebooks/Week9

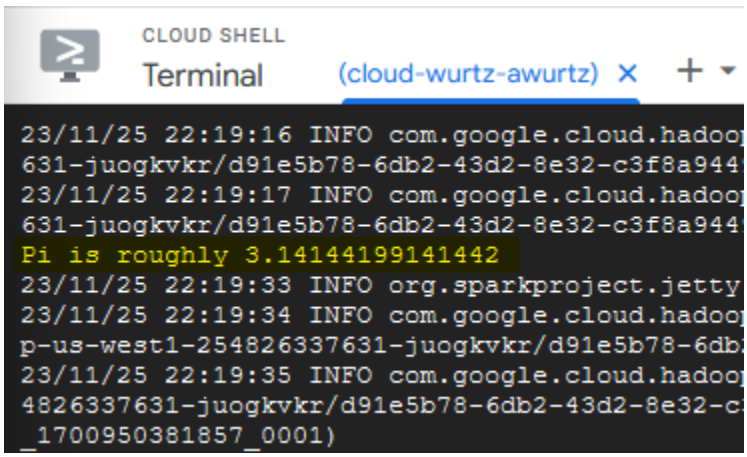
09.3g: Dataproc

Run computation

How long did the job take to execute?

It took about 37 seconds.

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

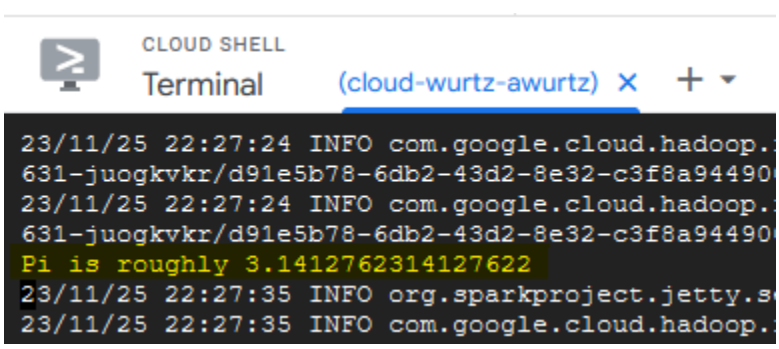


Run computation again

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

It took ~7 seconds to execute. That is 30 seconds faster than the previous execution.

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



Addison Wurtz
CS 530
notebooks/Week9

09.4g: Dataflow

Dataflow Lab #1 (Java package popularity)

Where is the input taken from by default?

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

Where does the output go by default?

```
default='/tmp/output'
```

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

PackageUse() implements the ParDo operation. It extracts all the packages that are used.

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

TotalUse implements the GroupByKey and Combine operations. It finds the sum of the number of times each package was used.

Which operations correspond to a "Map"?

beam.FlatMap corresponds to map.

Which operation corresponds to a "Shuffle-Reduce"?

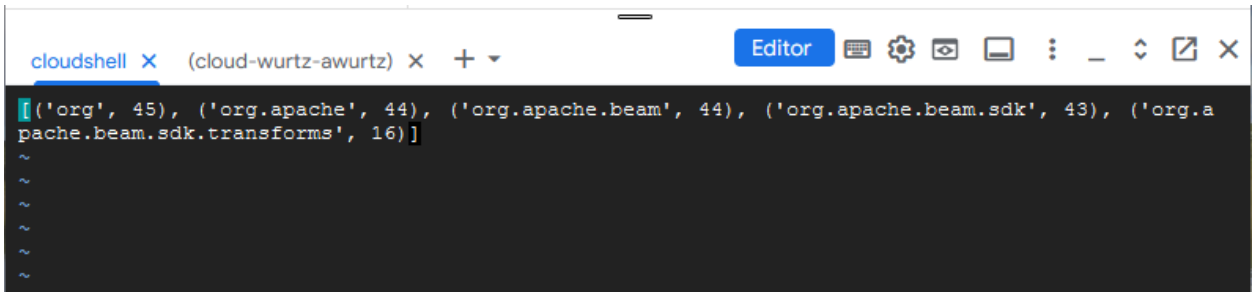
beam.CombinePerKey corresponds to shuffle-reduce.

Addison Wurtz
CS 530
notebooks/Week9

Which operation corresponds to a "Reduce"?

beam.transforms.combiners.Top.Of corresponds to reduce.

Run pipeline locally



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

The file contains the 5 most common import packages. Each entry shows the package name and the number of times it was found. So there were 45 imports that included org, 44 that included org.apache, 44 that included org.apache.beam, 43 that included org.apache.beam.sdk and 16 that included org.apache.beam.sdk.transforms.

Dataflow Lab #2 (Word count)

What are the names of the stages in the pipeline?

Read, Split, PairWithOne, GroupAndSum, Format, Write

Describe what each stage does.

Read: parses the input into lines of text
Split: parses each line of text into words
PairWithOne: Converts each word into a tuple containing the word and the integer 1
GroupAndSum: A shuffle-reduce that combines all the instances of each word and sums numbers resulting a tuple (word, wordcount)
Format: formats the tuple as a string “{word}: {wordcount}”
Write: Writes the formatted output to a file

Addison Wurtz
CS 530
notebooks/Week9

Run code locally

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

```
(env) awurtz@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/d
ataflow/python (cloud-wurtz-awurtz)$ wc --lines outputs-00000-of-00001
4784 outputs-00000-of-00001
(env) awurtz@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/d
(env) awurtz@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/d
ataflow/python (cloud-wurtz-awurtz)$
```

4784 different words

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) awurtz@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/d
ataflow/python (cloud-wurtz-awurtz)$ sort -n -r --key=2 outputs-00000-of-00001 | head --lines=3
the: 786
I: 622
and: 594
(env) awurtz@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/d
```

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

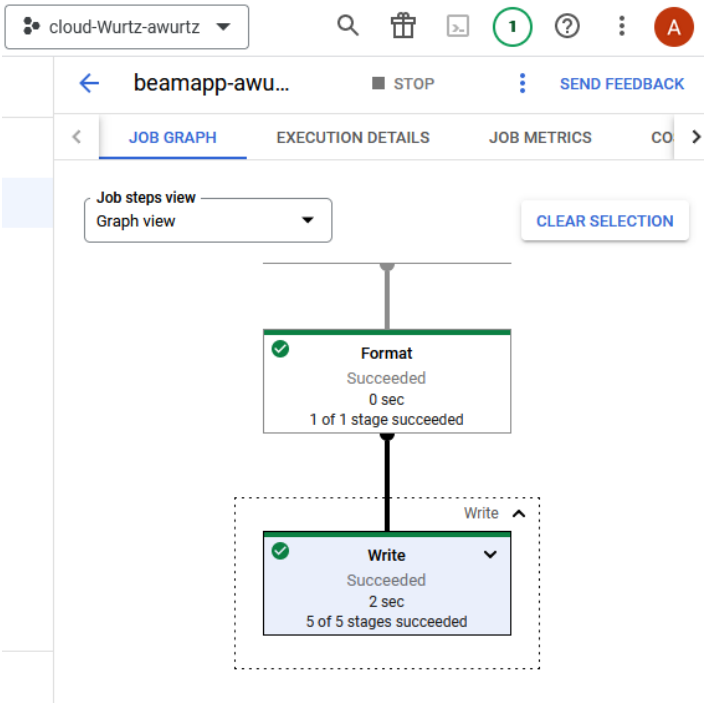
```
(env) awurtz@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/d
ataflow/python (cloud-wurtz-awurtz)$ s
ort -n -r --key=2 outputs-00000-of-00001 | head --lines=3
the: 908
and: 738
i: 622
(env) awurtz@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/d
ataflow/python (cloud-wurtz-awurtz)$
```

Run code using Dataflow runner

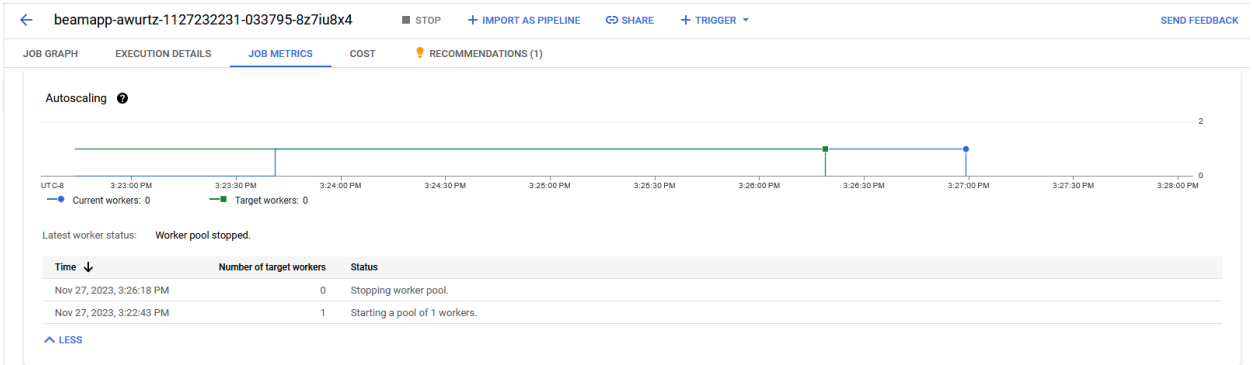
The part of the job graph that has taken the longest time to complete.

The Write step took the longest time to complete.

Addison Wurtz
CS 530
notebooks/Week9



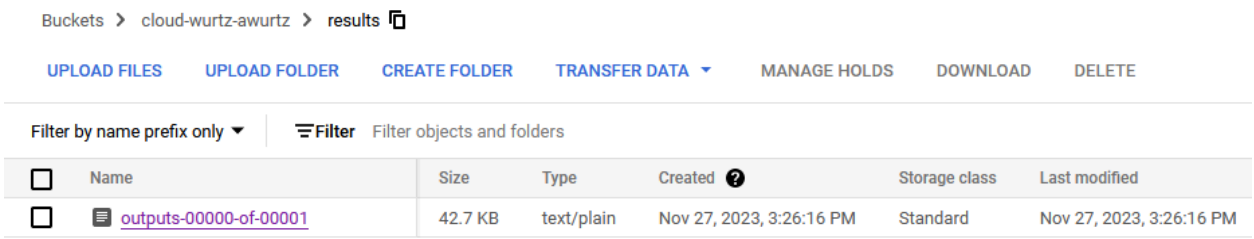
The autoscaling graph showing when the worker was created and stopped.



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

1

Addison Wurtz
CS 530
notebooks/Week9



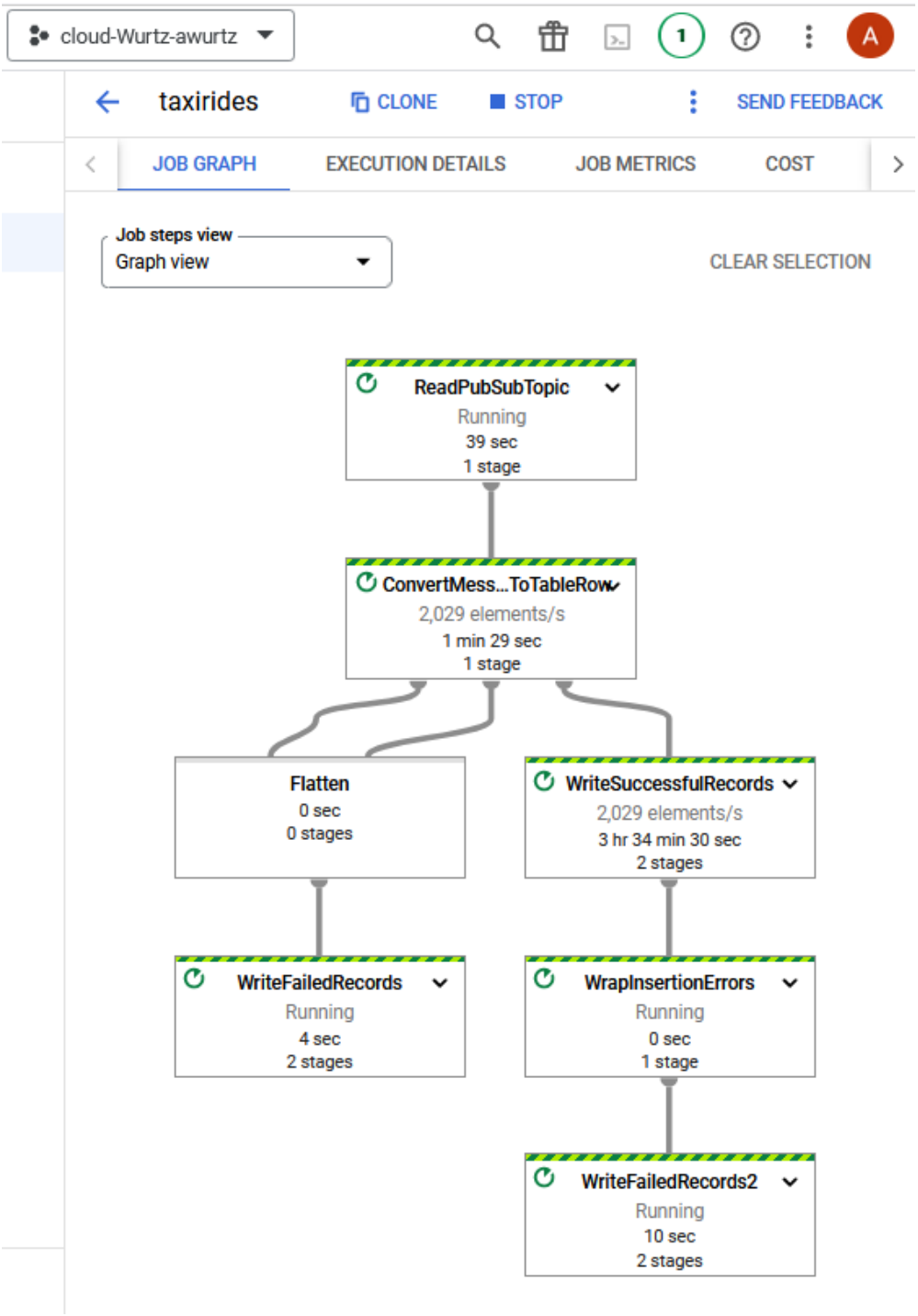
Dataflow Lab #3 (Taxi ETL pipeline)

View raw data from PubSub

```
(env) awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ gcloud pubsub subscriptions pull taxisub --auto
-ack
DATA: {"ride_id":"5e4a0c7f-dc01-426f-b6aa-2e3ceaad86d7","point_idx":1287,"latitude":40.71242,"l
ongitude":-73.72745,"timestamp":"2023-11-27T18:37:07.81108-05:00","meter_reading":24.146969,"me
ter_increment":0.018762214,"ride_status":"enroute","passenger_count":2}
MESSAGE_ID: 9718173923881906
ORDERING_KEY:
ATTRIBUTES: ts=2023-11-27T18:37:07.81108-05:00
DELIVERY_ATTEMPT:
ACK_STATUS: SUCCESS
(env) awurtz@cloudshell:~ (cloud-wurtz-awurtz)$
```

Addison Wurtz
CS 530
notebooks/Week9

Run Dataflow job from template



Addison Wurtz
CS 530
notebooks/Week9

Query data in BigQuery

SCHEMA | DETAILS | PREVIEW | LINEAGE | DATA PROFILE NEW | DATA QUALITY NEW

Row	ride_id	point_idx	latitude	longitude	timestamp	meter_reading	meter_incremen	ride_status	passenger_coun
1	e37d5368-a679-4671-8e79-6c3...	1217	40.6789000...	-73.881650...	2023-11-27 23:39:29.221530 U...	24.503355	0.020134227	enroute	5
2				0...	2023-11-27 23:39:29.236270 U...	18.029907	0.04953271	enroute	5
3				0...	2023-11-27 23:39:34.564520 U...	8.713513	0.035135135	enroute	6

ODIN ID awurtz

Streaming buffer statistics

Estimated size	222.88 MB
Estimated rows	1,368,259
Earliest entry time	Nov 27, 2023, 3:41:31 PM UTC-8

ODIN ID awurtz

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS

Row	minute	total_rides	total_passengers	total_revenue
1	15:39	151	274	2159.7900005
2	15:40	304	525	3958.530001000...
3	15:41	315	471	4341.0400103
4	15:42	327	557	4868.339987400...
5	15:43	317	507	4075.739997200...
6	15:44	297	499	4524.310005100...
7	15:45	297	473	4015.230003300...
8	15:46	315	512	4154.4100039
			96	4646.890005900...

ODIN ID awurtz

Addison Wurtz
CS 530
notebooks/Week9

Data Visualization