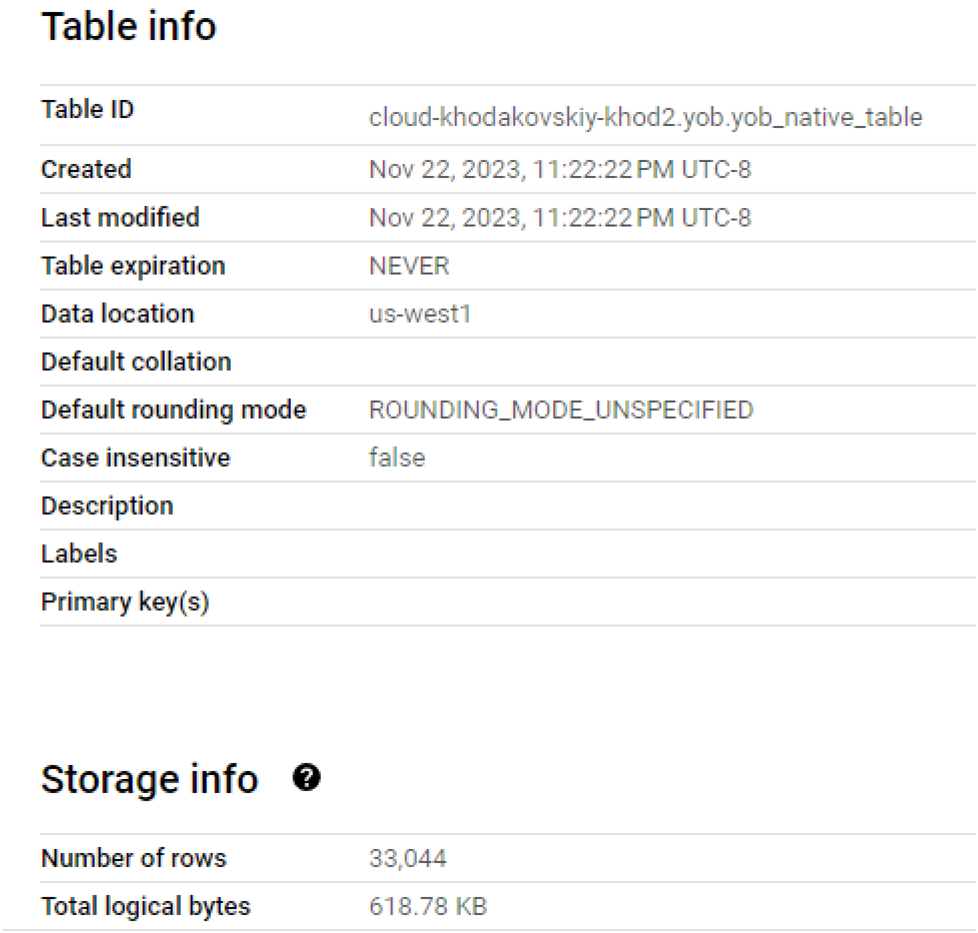
Lab 9 Notebook

## 9.1g

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



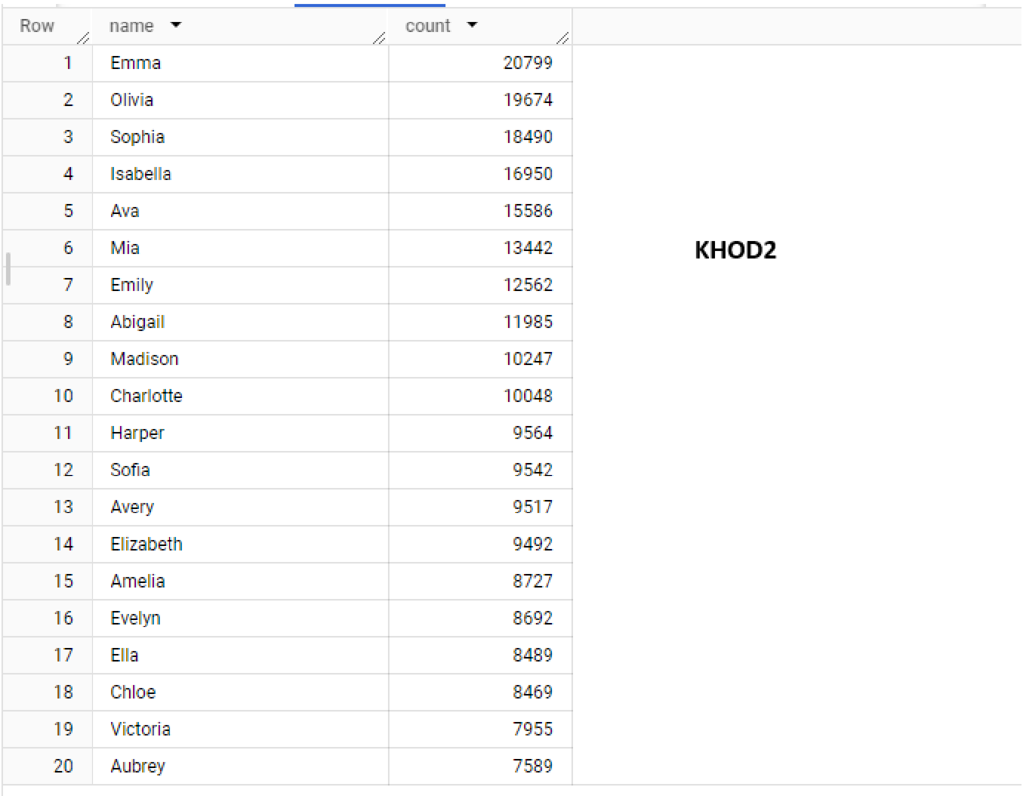
### Table info

| | |
|---|---|
| Table ID | cloud-khodakovskiy-khod2.yob.yob_native_table |
| Created | Nov 22, 2023, 11:22:22 PM UTC-8 |
| Last modified | Nov 22, 2023, 11:22:22 PM UTC-8 |
| 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 |

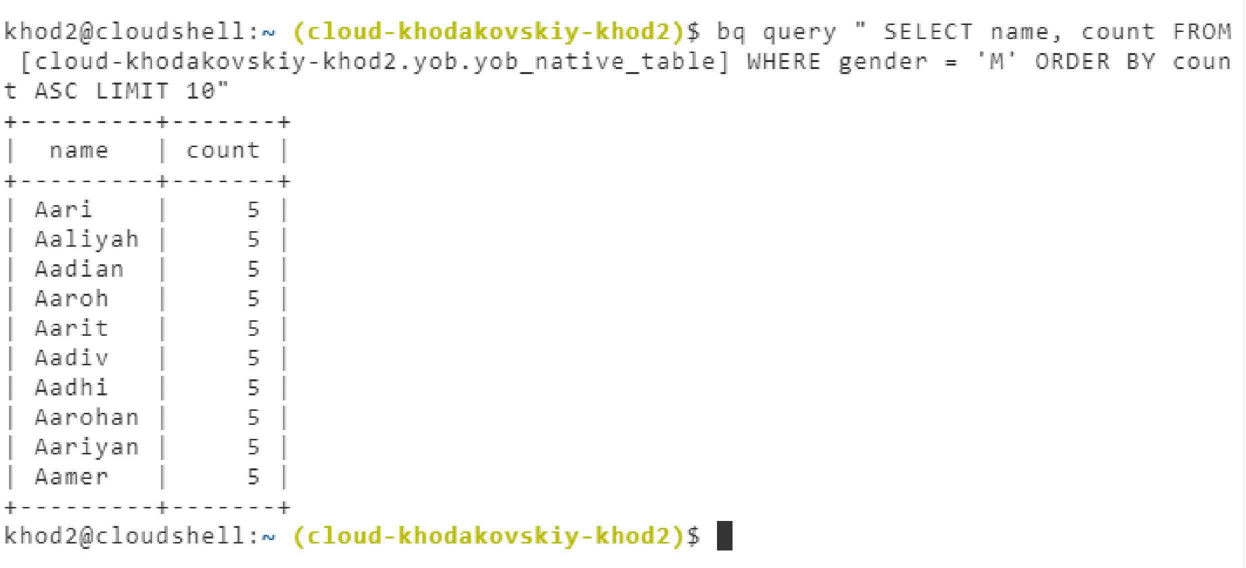
**Screenshot the query results and include it in your lab notebook.**



| 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 |

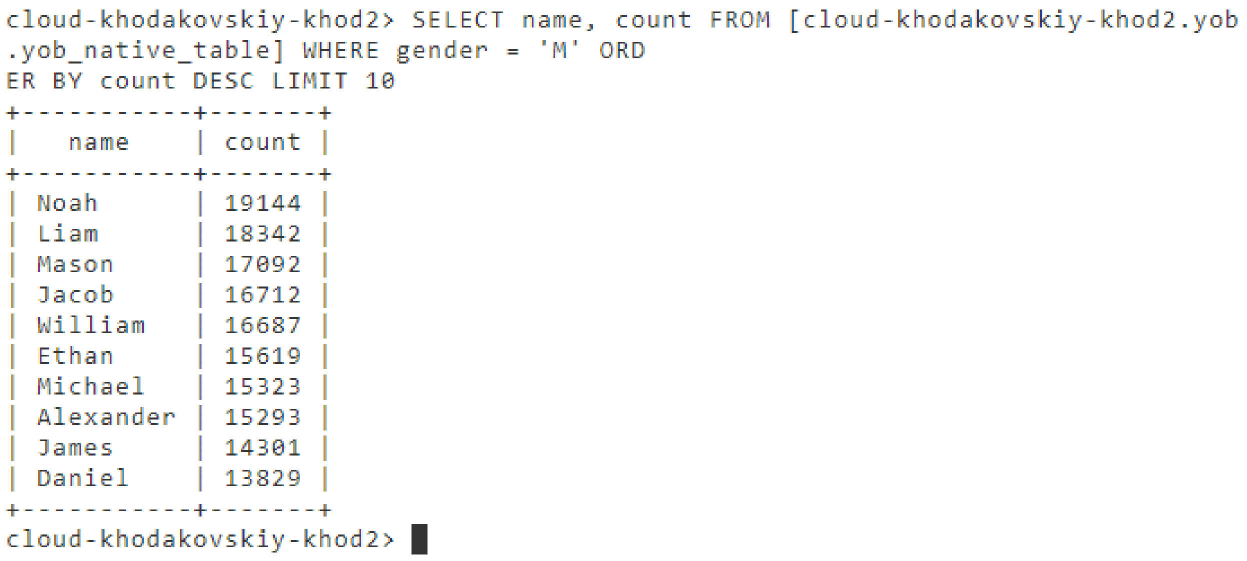
**KHOD2**

**Screenshot your results and include it in your lab notebook.**



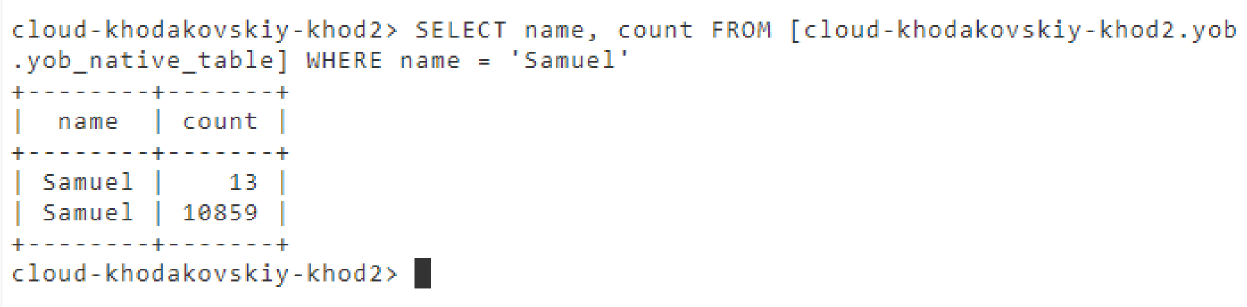
```
khod2@cloudshell:~ (cloud-khodakovskiy-khod2)$ bq query " SELECT name, count FROM
 [cloud-khodakovskiy-khod2.yob.yob_native_table] WHERE gender = 'M' ORDER BY coun
t ASC LIMIT 10"
+---------+-------+
|  name   | count |
+---------+-------+
| Aari    |     5 |
| Aaliyah |     5 |
| Aadian  |     5 |
| Aaroh   |     5 |
| Aarit   |     5 |
| Aadiv   |     5 |
| Aadhi   |     5 |
| Aarohan |     5 |
| Aariyan |     5 |
| Aamer   |     5 |
+---------+-------+
khod2@cloudshell:~ (cloud-khodakovskiy-khod2)$
```

**Screenshot your results and include it in your lab notebook.**



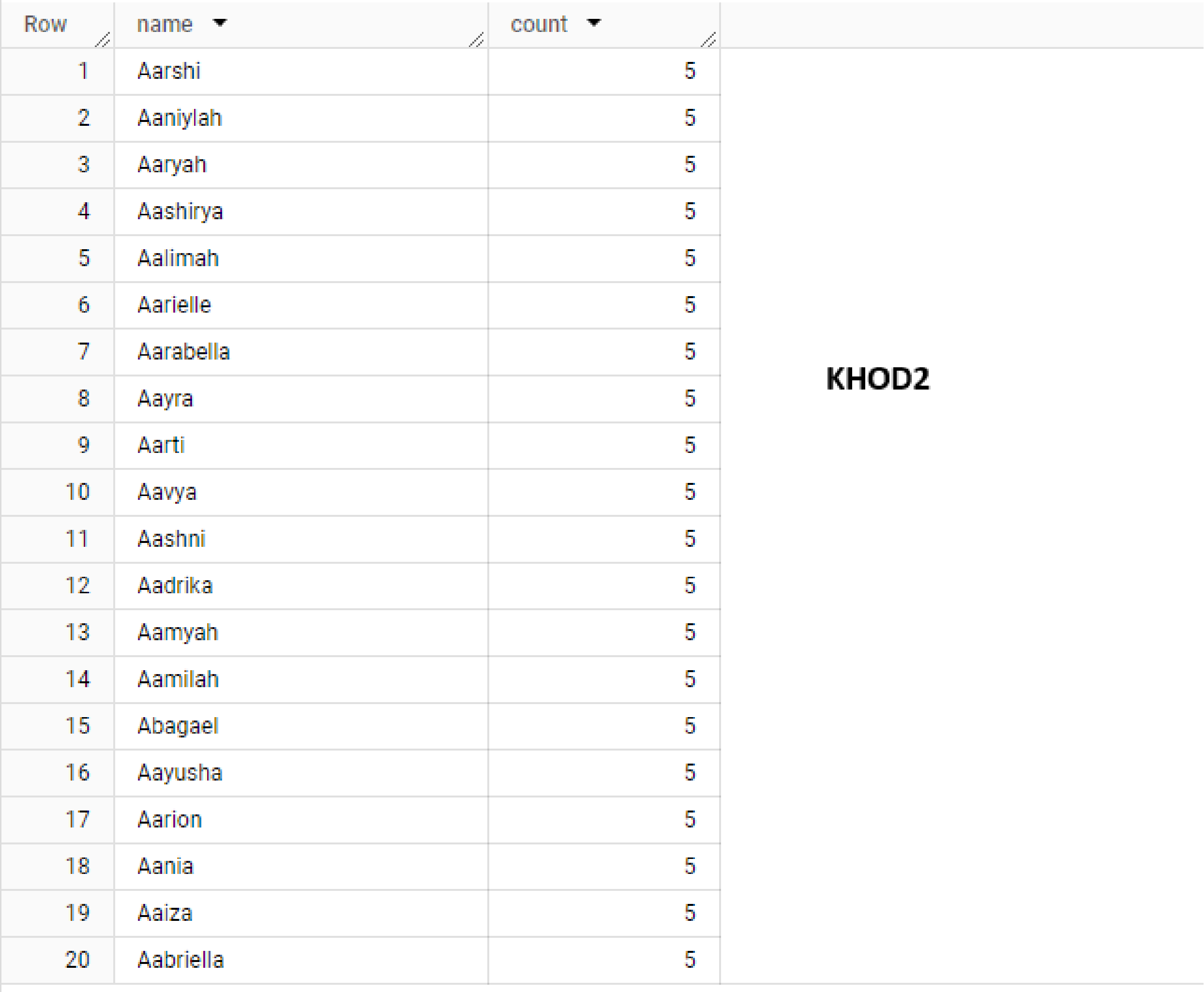
```
cloud-khodakovskiy-khod2> SELECT name, count FROM [cloud-khodakovskiy-khod2.yob
.yob_native_table] WHERE gender = 'M' ORD
ER BY count DESC LIMIT 10
+-----------+-------+
|   name    | count |
+-----------+-------+
| Noah      | 19144 |
| Liam      | 18342 |
| Mason     | 17092 |
| Jacob     | 16712 |
| William   | 16687 |
| Ethan     | 15619 |
| Michael   | 15323 |
| Alexander | 15293 |
| James     | 14301 |
| Daniel    | 13829 |
+-----------+-------+
cloud-khodakovskiy-khod2>
```

**Screenshot your results and include it in your lab notebook.**



```
cloud-khodakovskiy-khod2> SELECT name, count FROM [cloud-khodakovskiy-khod2.yob
.yob_native_table] WHERE name = 'Samuel'
+--------+-------+
|  name  | count |
+--------+-------+
| Samuel |    13 |
| Samuel | 10859 |
+--------+-------+
cloud-khodakovskiy-khod2>
```

**9. Screenshot the query results and include it in your lab notebook.**



| Row | name | count |
|---|---|---|
| 1 | Aarshi | 5 |
| 2 | Aaniylah | 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 |

**KHOD2**

## 9.2g

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

   I'm assuming the size of the table is the size of the original query since it's selecting all. The original query was about 10 gb, this query is about three. Seven less gigabytes.

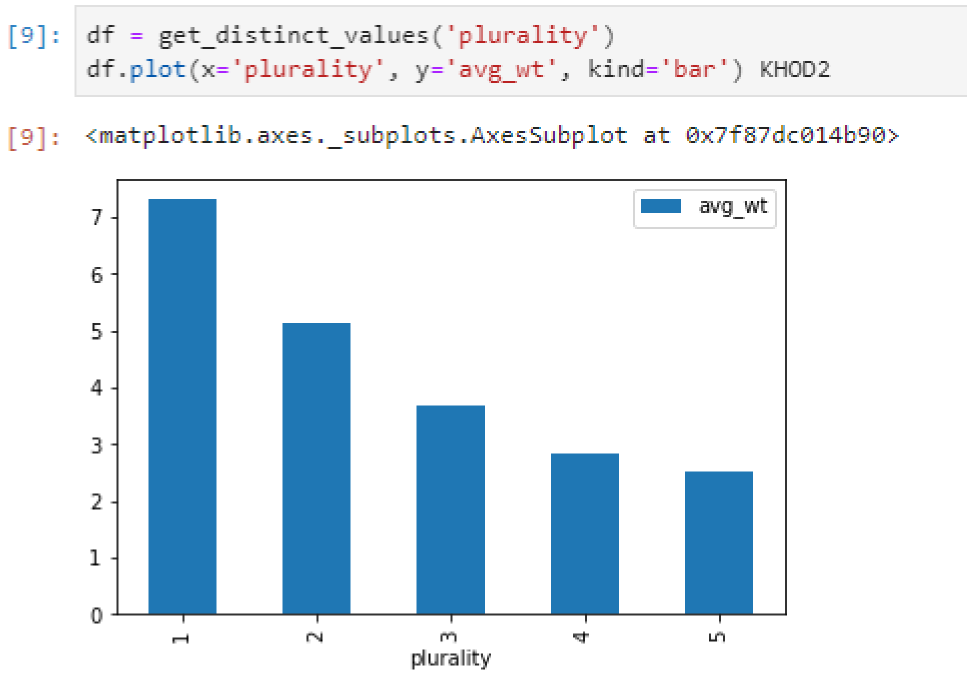
   **How many twins were born during this time range?**

   375,362.

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

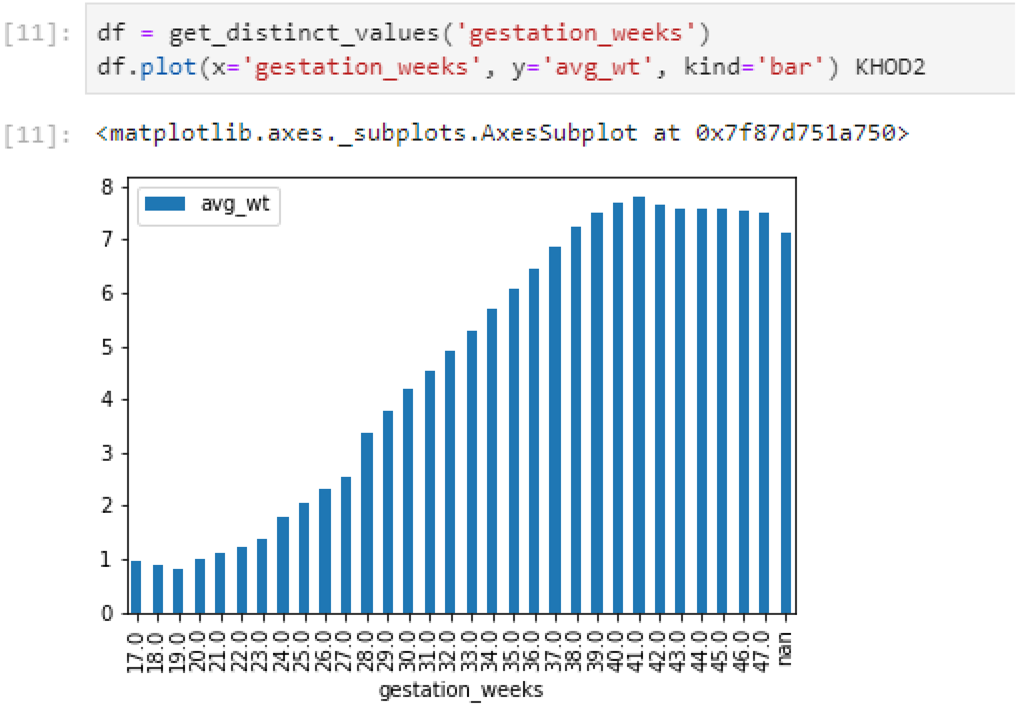
   They're about 2.2 lbs (?) lighter than single babies, on average.

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



```
[9]: df = get_distinct_values('plurality')
     df.plot(x='plurality', y='avg_wt', kind='bar') KHOD2

[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f87dc014b90>
```



```
[11]: df = get_distinct_values('gestation_weeks')
      df.plot(x='gestation_weeks', y='avg_wt', kind='bar') KHOD2

[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f87d751a750>
```

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

   March 3rd, 2020.

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

   The number of trips to work decreased by 49% off the baseline.

9. **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.

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

   The same three.

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

    excess_deaths: placename, start_date, excess_deaths.

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

    us_counties: date, county, deaths.

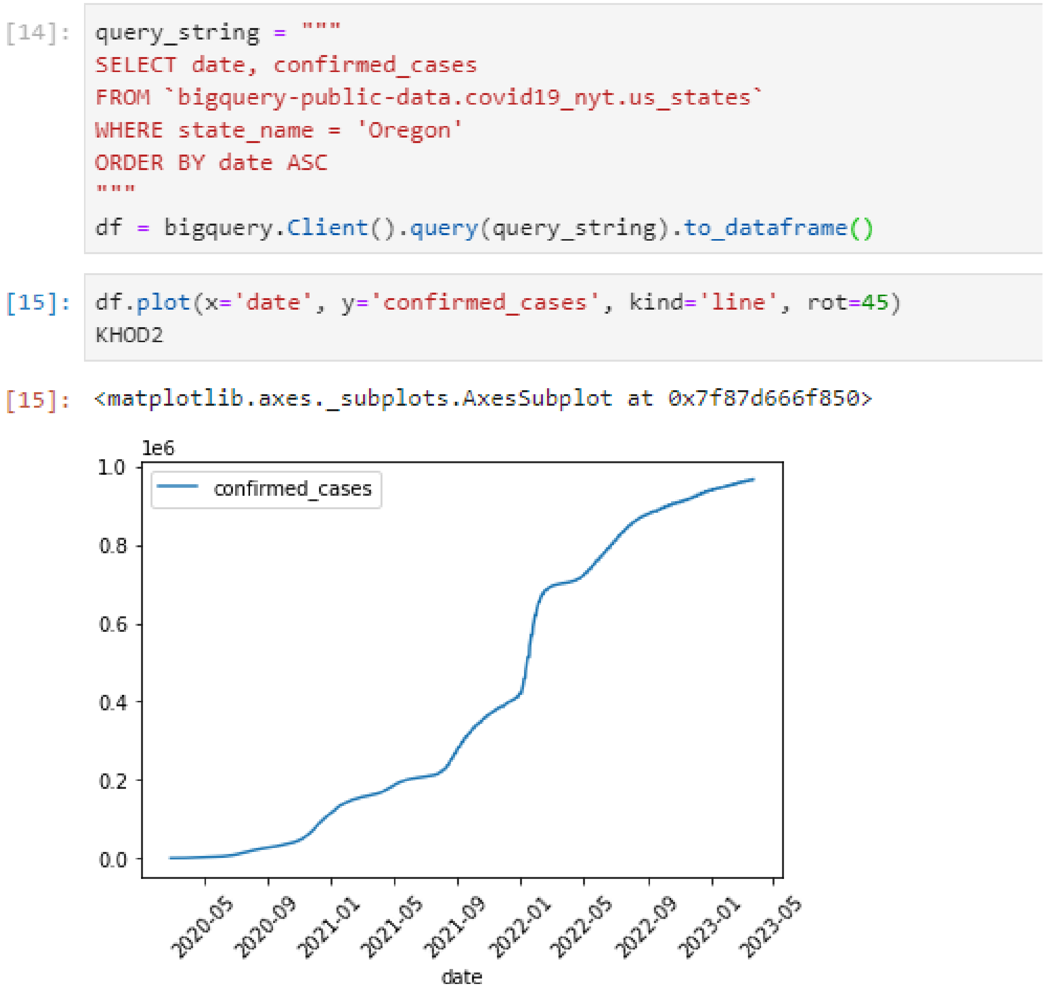
    **What table and columns identify the date, state, and confirmed cases of COVID-19?**

    us_states: date, state_name, confirmed_cases.

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

    mask_use_by_county: county_fips_code, always.

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

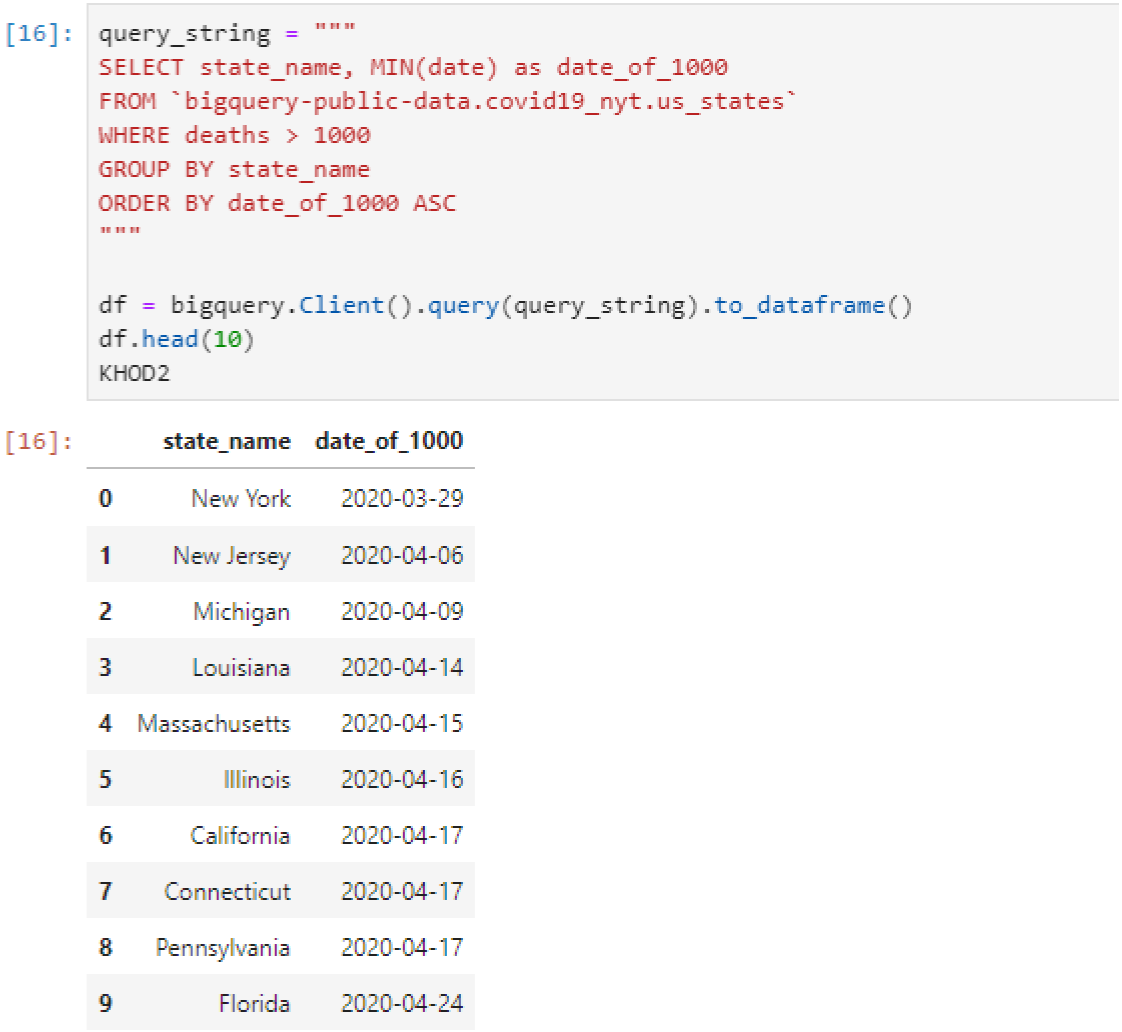


```
[14]: query_string = """
      SELECT date, confirmed_cases
      FROM `bigquery-public-data.covid19_nyt.us_states`
      WHERE state_name = 'Oregon'
      ORDER BY date ASC
      """
      df = bigquery.Client().query(query_string).to_dataframe()

[15]: df.plot(x='date', y='confirmed_cases', kind='line', rot=45)
      KHOD2

[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7f87d666f850>
```

**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.**



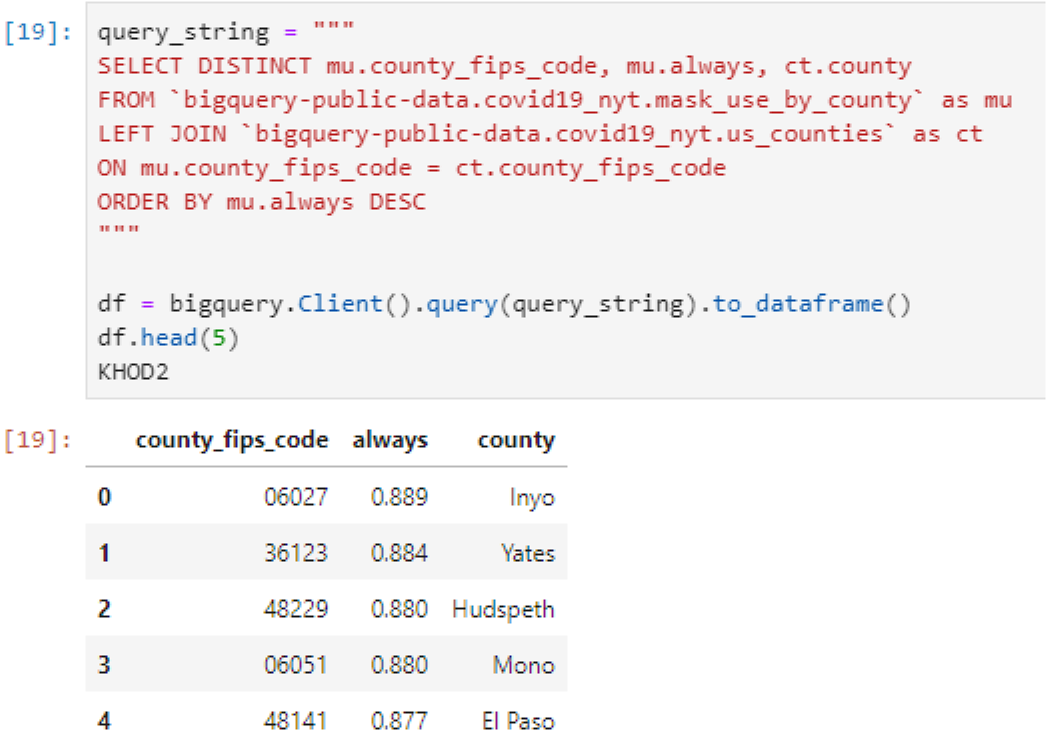
```
[16]: query_string = """
      SELECT state_name, MIN(date) as date_of_1000
      FROM `bigquery-public-data.covid19_nyt.us_states`
      WHERE deaths > 1000
      GROUP BY state_name
      ORDER BY date_of_1000 ASC
      """

      df = bigquery.Client().query(query_string).to_dataframe()
      df.head(10)
      KHOD2
```

[16]:

| | 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.**



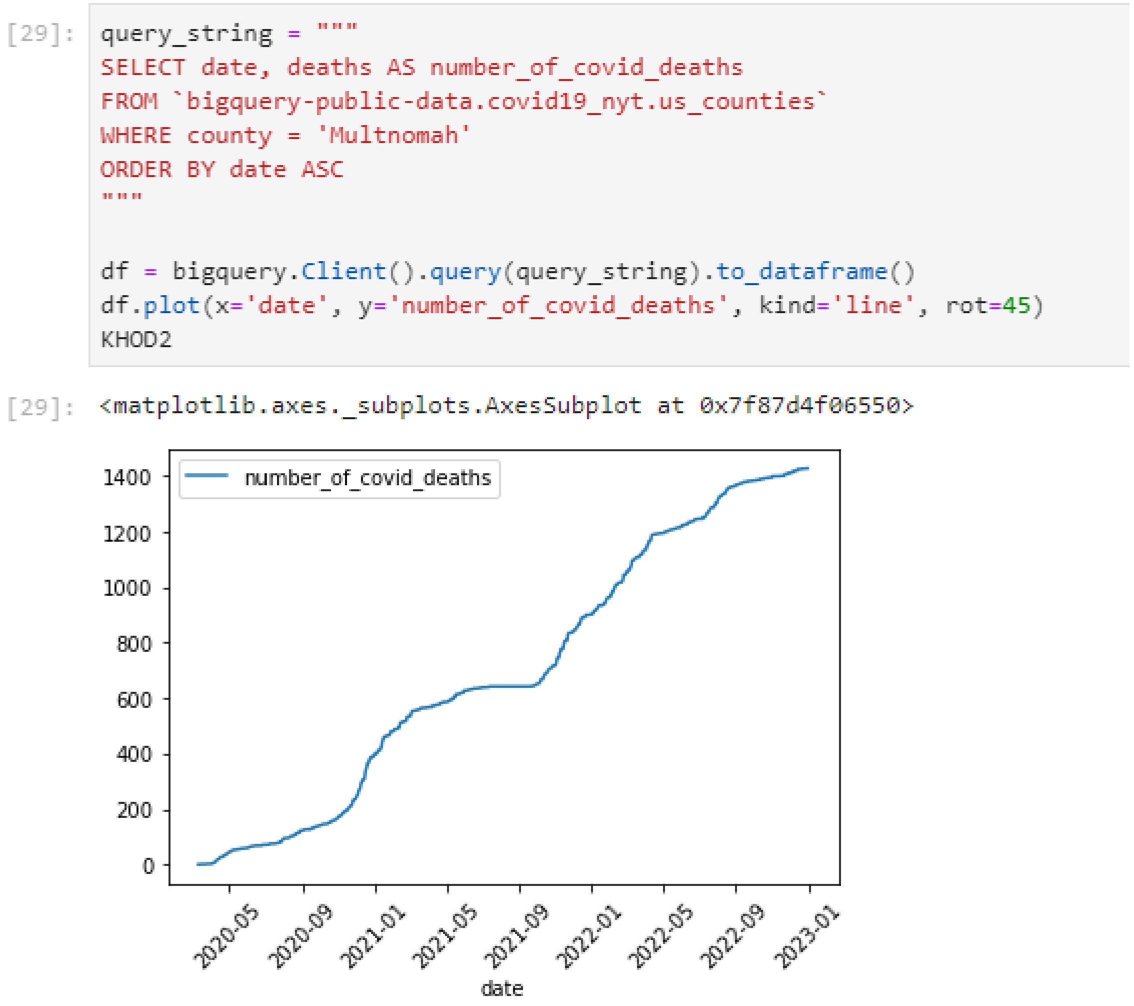
```
[19]: query_string = """
      SELECT DISTINCT mu.county_fips_code, mu.always, ct.county
      FROM `bigquery-public-data.covid19_nyt.mask_use_by_county` as mu
      LEFT JOIN `bigquery-public-data.covid19_nyt.us_counties` as ct
      ON mu.county_fips_code = ct.county_fips_code
      ORDER BY mu.always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

      df = bigquery.Client().query(query_string).to_dataframe()
      df.head(5)
      KHOD2
```

[19]:

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

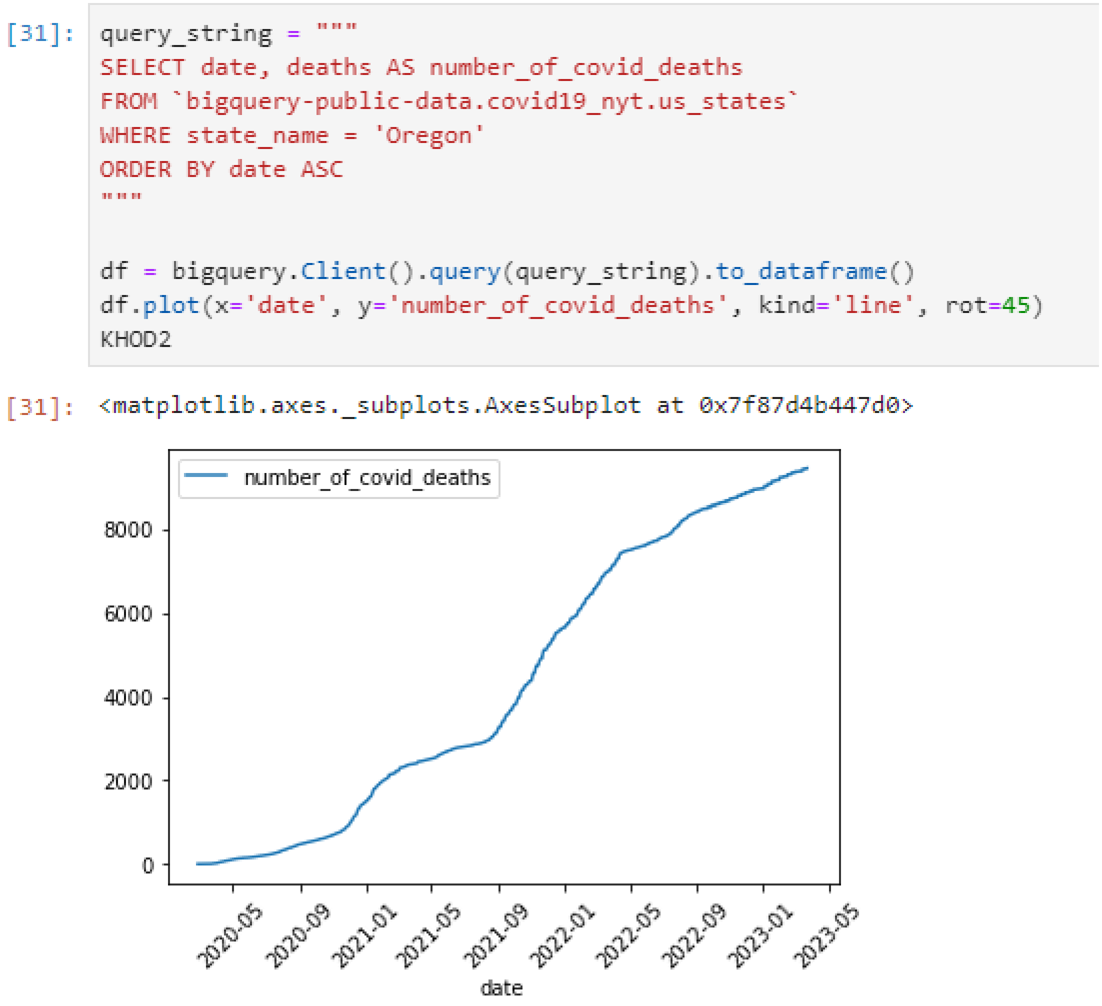


```
[29]: query_string = """
      SELECT date, deaths AS number_of_covid_deaths
      FROM `bigquery-public-data.covid19_nyt.us_counties`
      WHERE county = 'Multnomah'
      ORDER BY date ASC
      """

      df = bigquery.Client().query(query_string).to_dataframe()
      df.plot(x='date', y='number_of_covid_deaths', kind='line', rot=45)
      KHOD2
```

```
[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7f87d4f06550>
```

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



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

      df = bigquery.Client().query(query_string).to_dataframe()
      df.plot(x='date', y='number_of_covid_deaths', kind='line', rot=45)
      KHOD2
```

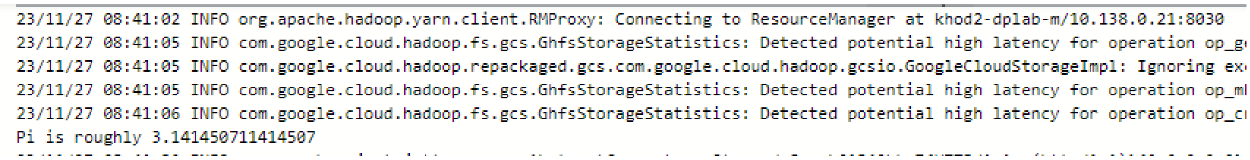
```
[31]: <matplotlib.axes._subplots.AxesSubplot at 0x7f87d4b447d0>
```

## 9.3g

6. **How long did the job take to execute?**

It took about a minute and 10 seconds.

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

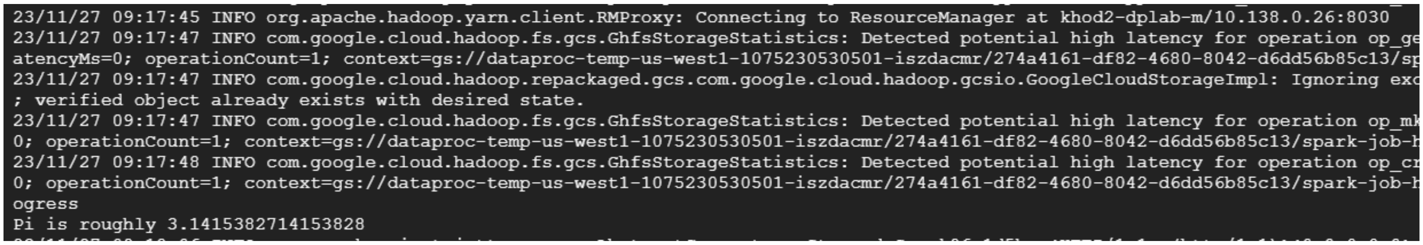


```
23/11/27 08:41:02 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at khod2-dplab-m/10.138.0.21:8030
23/11/27 08:41:05 INFO com.google.cloud.hadoop.fs.gcs.GhfsStorageStatistics: Detected potential high latency for operation op_g
23/11/27 08:41:05 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring ex
23/11/27 08:41:05 INFO com.google.cloud.hadoop.fs.gcs.GhfsStorageStatistics: Detected potential high latency for operation op_m
23/11/27 08:41:06 INFO com.google.cloud.hadoop.fs.gcs.GhfsStorageStatistics: Detected potential high latency for operation op_c
Pi is roughly 3.141450711414507
```

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

   The job took about 30 seconds to execute, about 40 seconds faster.

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



```
23/11/27 09:17:45 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at khod2-dplab-m/10.138.0.26:8030
23/11/27 09:17:47 INFO com.google.cloud.hadoop.fs.gcs.GhfsStorageStatistics: Detected potential high latency for operation op_ge
atencyMs=0; operationCount=1; context=gs://dataproc-temp-us-west1-1075230530501-iszdacmr/274a4161-df82-4680-8042-d6dd56b85c13/sp
23/11/27 09:17:47 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring exc
; verified object already exists with desired state.
23/11/27 09:17:47 INFO com.google.cloud.hadoop.fs.gcs.GhfsStorageStatistics: Detected potential high latency for operation op_mk
0; operationCount=1; context=gs://dataproc-temp-us-west1-1075230530501-iszdacmr/274a4161-df82-4680-8042-d6dd56b85c13/spark-job-h
23/11/27 09:17:48 INFO com.google.cloud.hadoop.fs.gcs.GhfsStorageStatistics: Detected potential high latency for operation op_cr
0; operationCount=1; context=gs://dataproc-temp-us-west1-1075230530501-iszdacmr/274a4161-df82-4680-8042-d6dd56b85c13/spark-job-h
ogress
Pi is roughly 3.1415382714153828
```

## 9.4g

3. **Where is the input taken from by default?**

   The input is taken from all the java files in the javahelp directory. The parser specifies that by adding an –input option to its arguments.

**Where does the output go by default?**

The output goes to /tmp/output, using the writeToText function.

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

The PackageUse() function implements a transform to take an “import *origin.packagename.etc*” line and return a list of the pieces of the package name.

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

The TotalUse transform counts the occurrences of each package found in the files and returns that.

**Which operations correspond to a "Map"?**

GetImports and PackageUse, and maybe GetJava? I’m not sure about the last one.

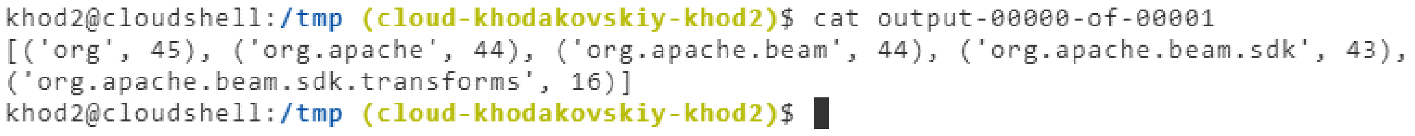
**Which operation corresponds to a "Shuffle-Reduce"?**

None that I can see.

**Which operation corresponds to a "Reduce"?**

TotalUse and Top_5.

4. **Take a screenshot of its contents.**



```
khod2@cloudshell:/tmp (cloud-khodakovskiy-khod2)$ 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)]
khod2@cloudshell:/tmp (cloud-khodakovskiy-khod2)$ 
```

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

The data in this output file corresponds to the most popular packages that are imported in the javahelp directory. After the input of all *import* statements in the files, the program transforms the listed packages into an aggregated count of each package, then lists the top 5.

5. **What are the names of the stages in the pipeline?**

    Split, PairWithOne, and GroupAndSum.

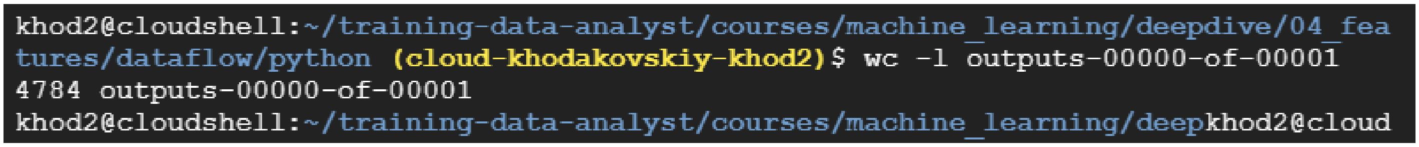
    **Describe what each stage does.**

    The Split stage takes a list of lines from the pipeline and performs the WordExtractingDoFN() function on each one, returning a list of matching words.

    The PairWithOne stage maps each word as a tuple of (word, 1 count) and returns that to the next one.

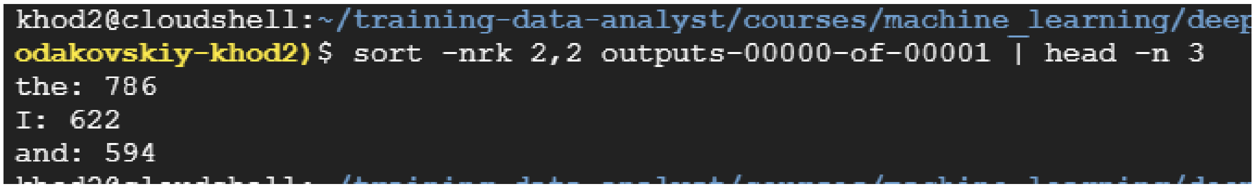
    GroupAndSum counts each word's occurrences and returns a list of words and their count.

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



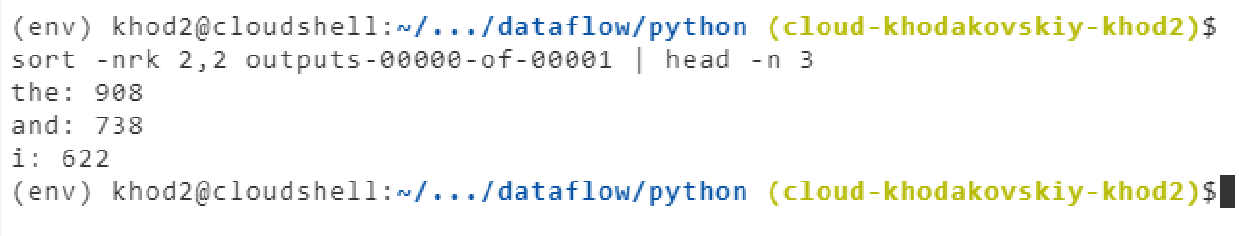
```
khod2@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_fea
tures/dataflow/python (cloud-khodakovskiy-khod2)$ wc -l outputs-00000-of-00001
4784 outputs-00000-of-00001
khod2@cloudshell:~/training-data-analyst/courses/machine_learning/deepkhod2@cloud
```

**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.**



```
khod2@cloudshell:~/training-data-analyst/courses/machine_learning/deep
odakovskiy-khod2)$ sort -nrk 2,2 outputs-00000-of-00001 | head -n 3
the: 786
I: 622
and: 594
khod2@cloudshell:~/training-data-analyst/courses/machine_learning/deep
```

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

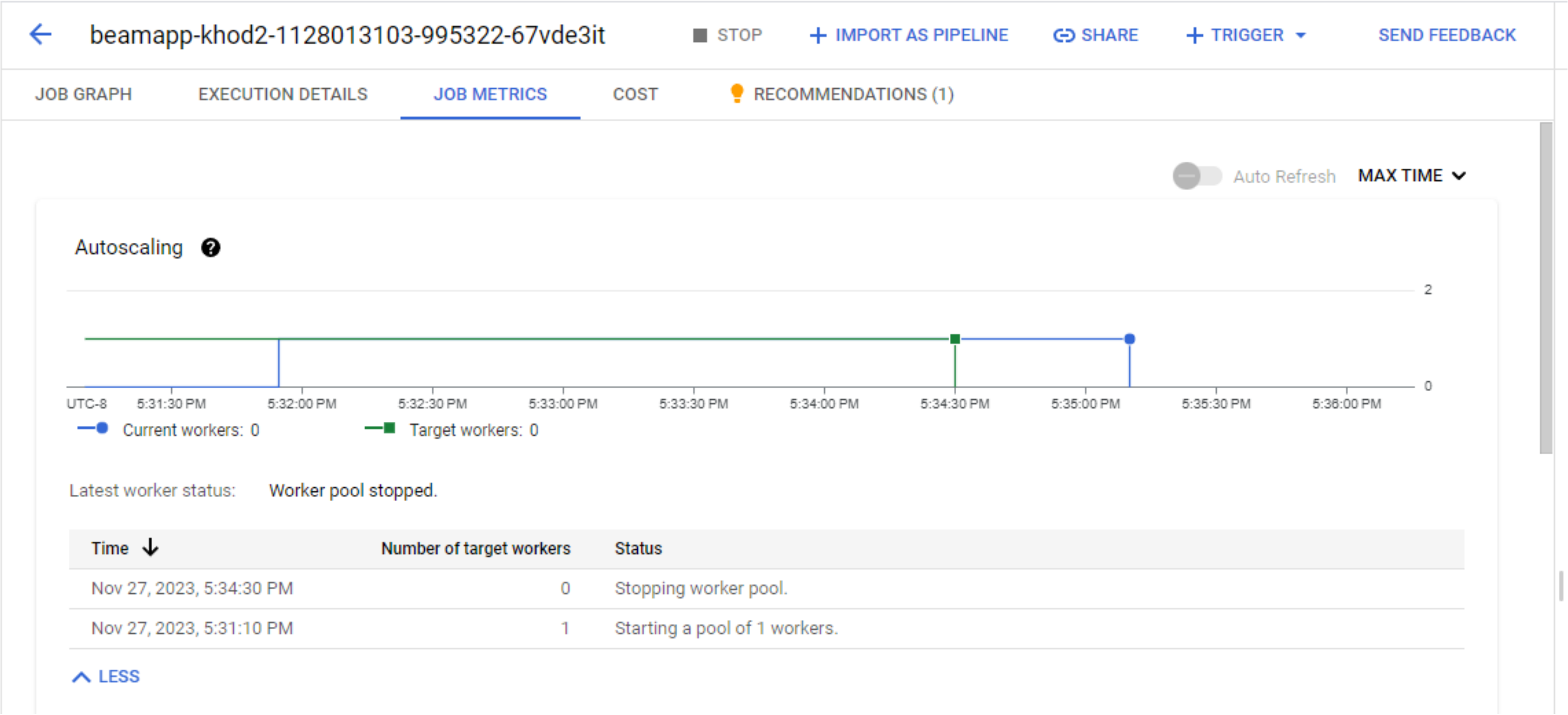


```
(env) khod2@cloudshell:~/.../dataflow/python (cloud-khodakovskiy-khod2)$
sort -nrk 2,2 outputs-00000-of-00001 | head -n 3
the: 908
and: 738
i: 622
(env) khod2@cloudshell:~/.../dataflow/python (cloud-khodakovskiy-khod2)$
```

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

According to the graph, the write section took the longest.

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



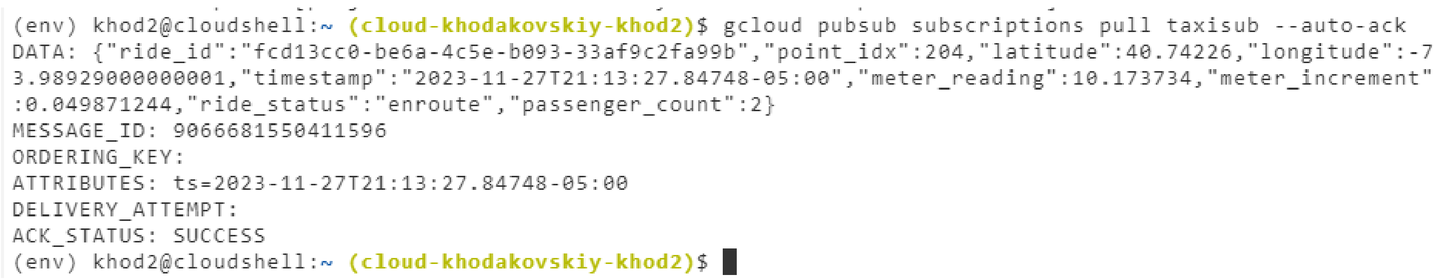
Latest worker status: Worker pool stopped.

| Time ↓ | Number of target workers | Status |
|---|---|---|
| Nov 27, 2023, 5:34:30 PM | 0 | Stopping worker pool. |
| Nov 27, 2023, 5:31:10 PM | 1 | Starting a pool of 1 workers. |

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

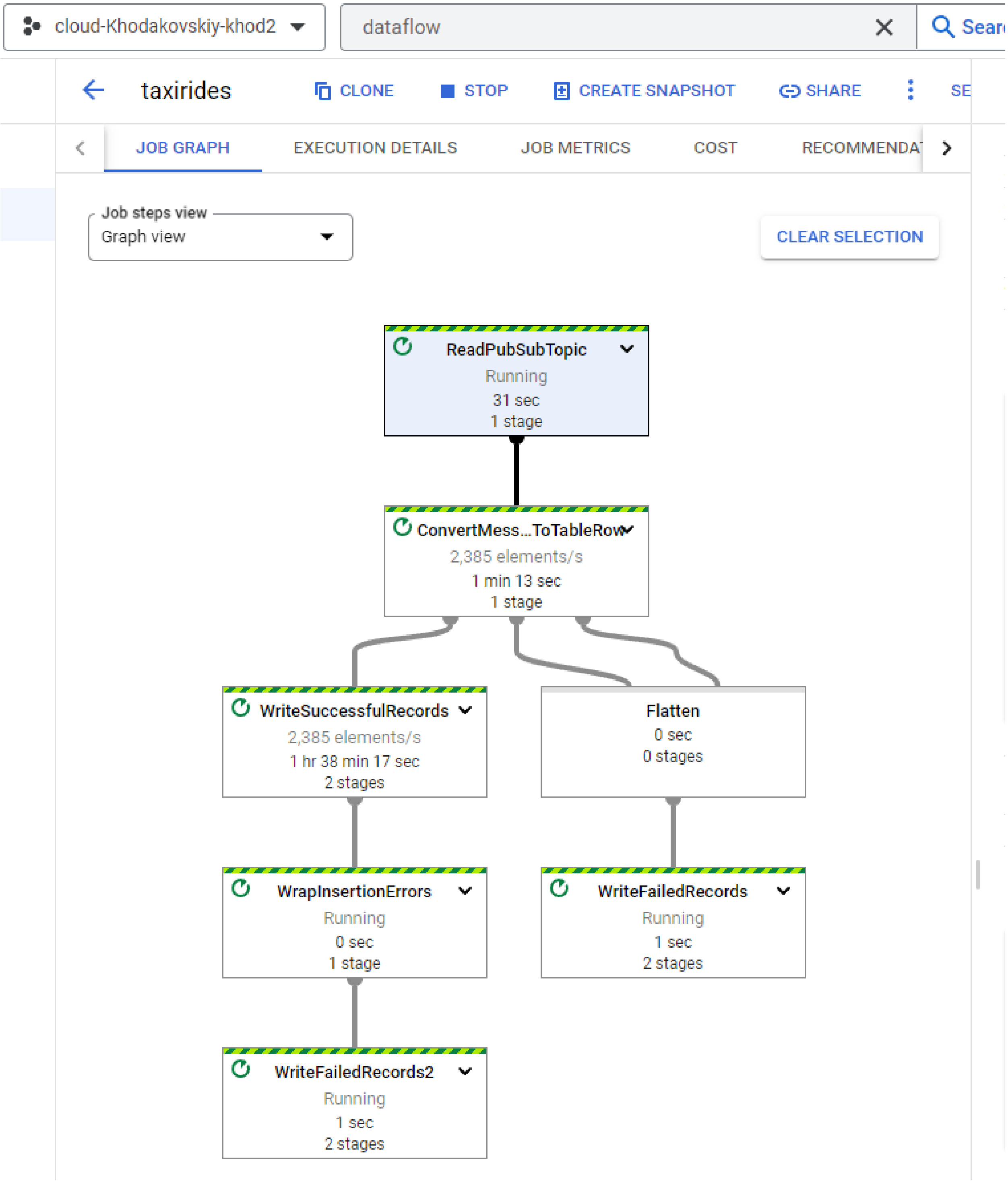
It created one file, outputs-00000-of-00001.

**12. Take a screenshot listing the different fields of this object.**

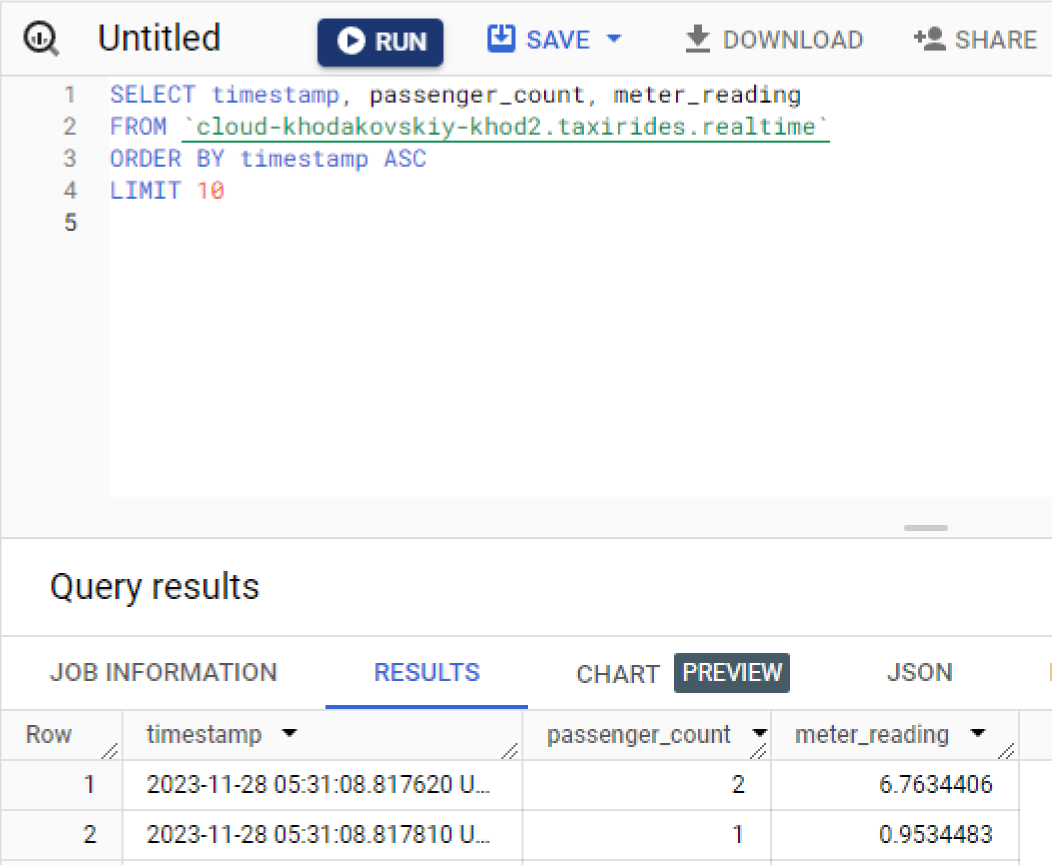


```
(env) khod2@cloudshell:~ (cloud-khodakovskiy-khod2)$ gcloud pubsub subscriptions pull taxisub --auto-ack
DATA: {"ride_id":"fcd13cc0-be6a-4c5e-b093-33af9c2fa99b","point_idx":204,"latitude":40.74226,"longitude":-7
3.98929000000001,"timestamp":"2023-11-27T21:13:27.84748-05:00","meter_reading":10.173734,"meter_increment"
:0.049871244,"ride_status":"enroute","passenger_count":2}
MESSAGE_ID: 9066681550411596
ORDERING_KEY:
ATTRIBUTES: ts=2023-11-27T21:13:27.84748-05:00
DELIVERY_ATTEMPT:
ACK_STATUS: SUCCESS
(env) khod2@cloudshell:~ (cloud-khodakovskiy-khod2)$
```

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



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



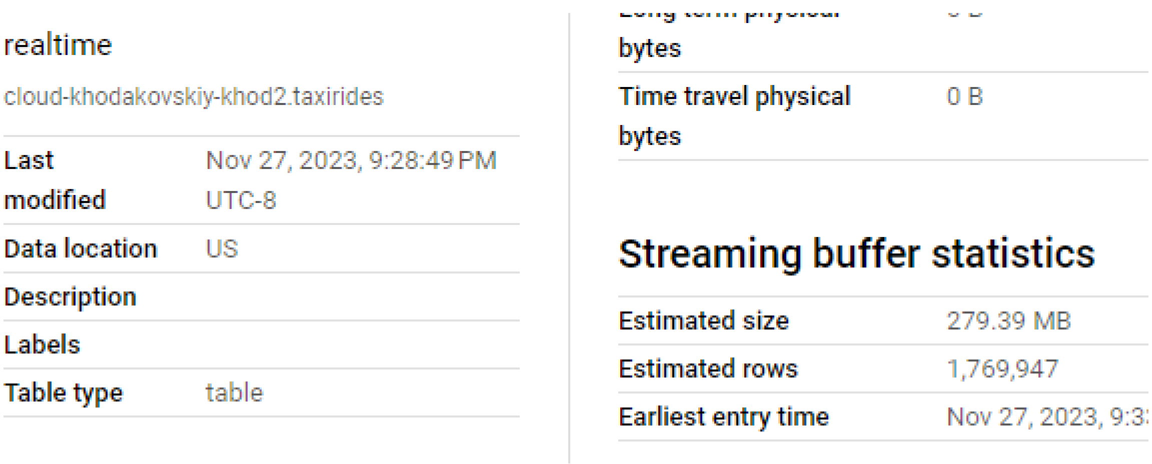
```
SELECT timestamp, passenger_count, meter_reading
FROM `cloud-khodakovskiy-khod2.taxirides.realtime`
ORDER BY timestamp ASC
LIMIT 10

```

Query results

| Row | timestamp | passenger_count | meter_reading |
|---|---|---|---|
| 1 | 2023-11-28 05:31:08.817620 U... | 2 | 6.7634406 |
| 2 | 2023-11-28 05:31:08.817810 U... | 1 | 0.9534483 |

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



realtime

cloud-khodakovskiy-khod2.taxirides

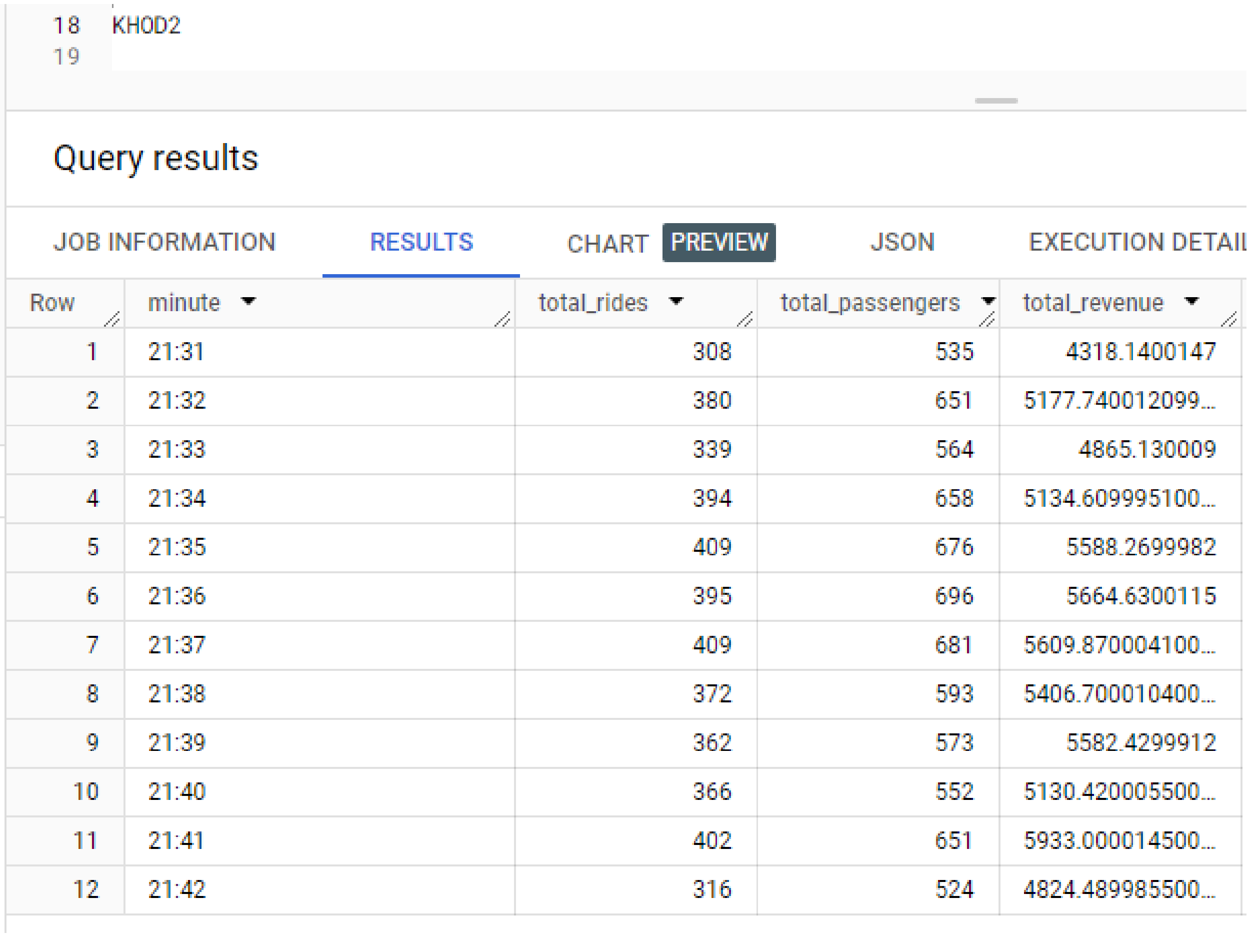
| | |
|---|---|
| Last modified | Nov 27, 2023, 9:28:49 PM UTC-8 |
| Data location | US |
| Description | |
| Labels | |
| Table type | table |

| | |
|---|---|
| Long term physical bytes | 0 B |
| Time travel physical bytes | 0 B |

Streaming buffer statistics

| | |
|---|---|
| Estimated size | 279.39 MB |
| Estimated rows | 1,769,947 |
| Earliest entry time | Nov 27, 2023, 9:3 |

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

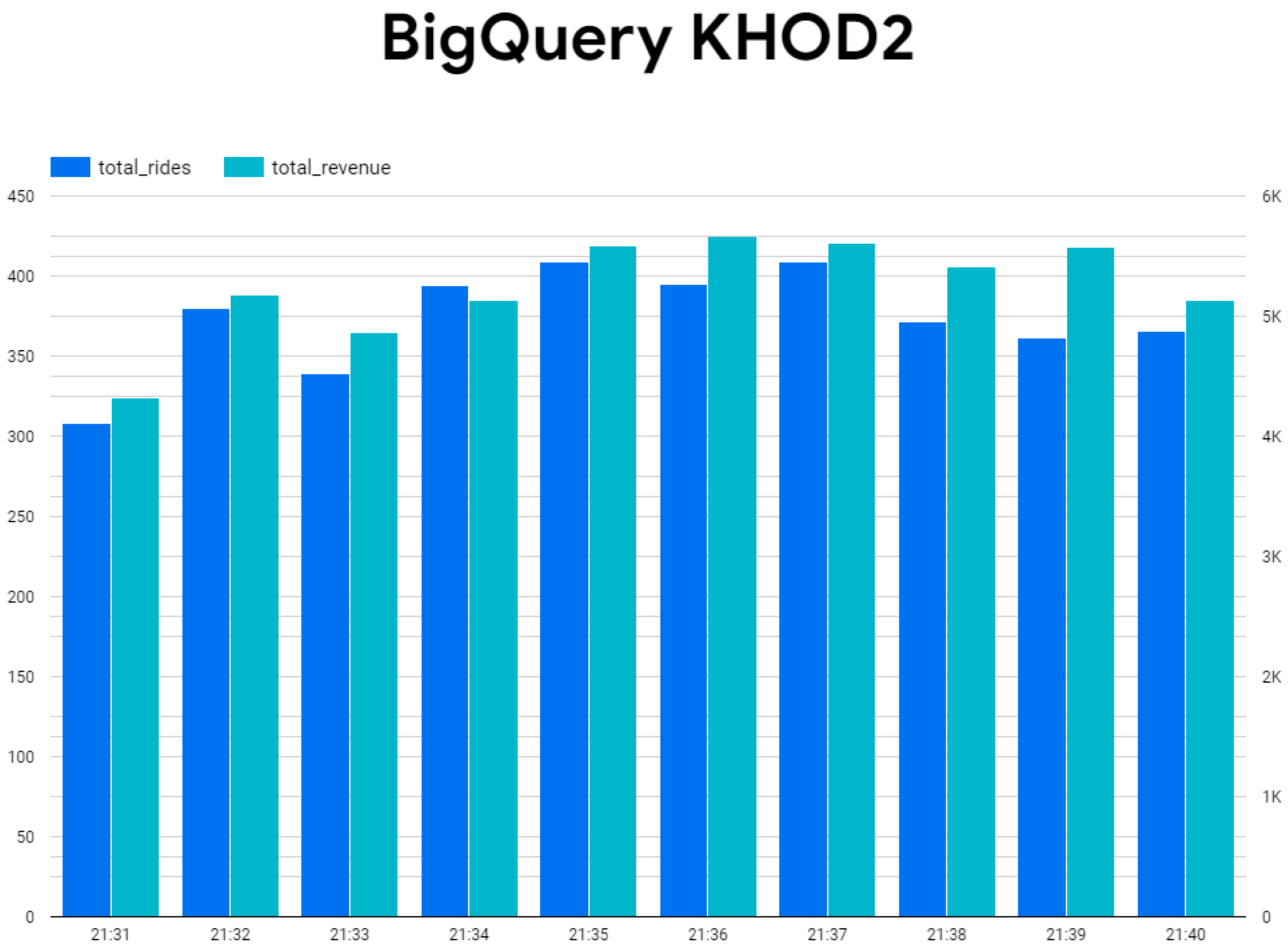


```
KHOD2
```

Query results

| Row | minute | total_rides | total_passengers | total_revenue |
|---|---|---|---|---|
| 1 | 21:31 | 308 | 535 | 4318.1400147 |
| 2 | 21:32 | 380 | 651 | 5177.740012099... |
| 3 | 21:33 | 339 | 564 | 4865.130009 |
| 4 | 21:34 | 394 | 658 | 5134.609995100... |
| 5 | 21:35 | 409 | 676 | 5588.2699982 |
| 6 | 21:36 | 395 | 696 | 5664.6300115 |
| 7 | 21:37 | 409 | 681 | 5609.870004100... |
| 8 | 21:38 | 372 | 593 | 5406.700010400... |
| 9 | 21:39 | 362 | 573 | 5582.4299912 |
| 10 | 21:40 | 366 | 552 | 5130.420005500... |
| 11 | 21:41 | 402 | 651 | 5933.000014500... |
| 12 | 21:42 | 316 | 524 | 4824.489985500... |

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


BigQuery KHOD2
total_rides
total_revenue
6K
5K
4K
3K
2K
1K
21:31
21:32
21:33
21:34
21:35
21:36
21:37
21:38
21:39
21:40