Lab 09

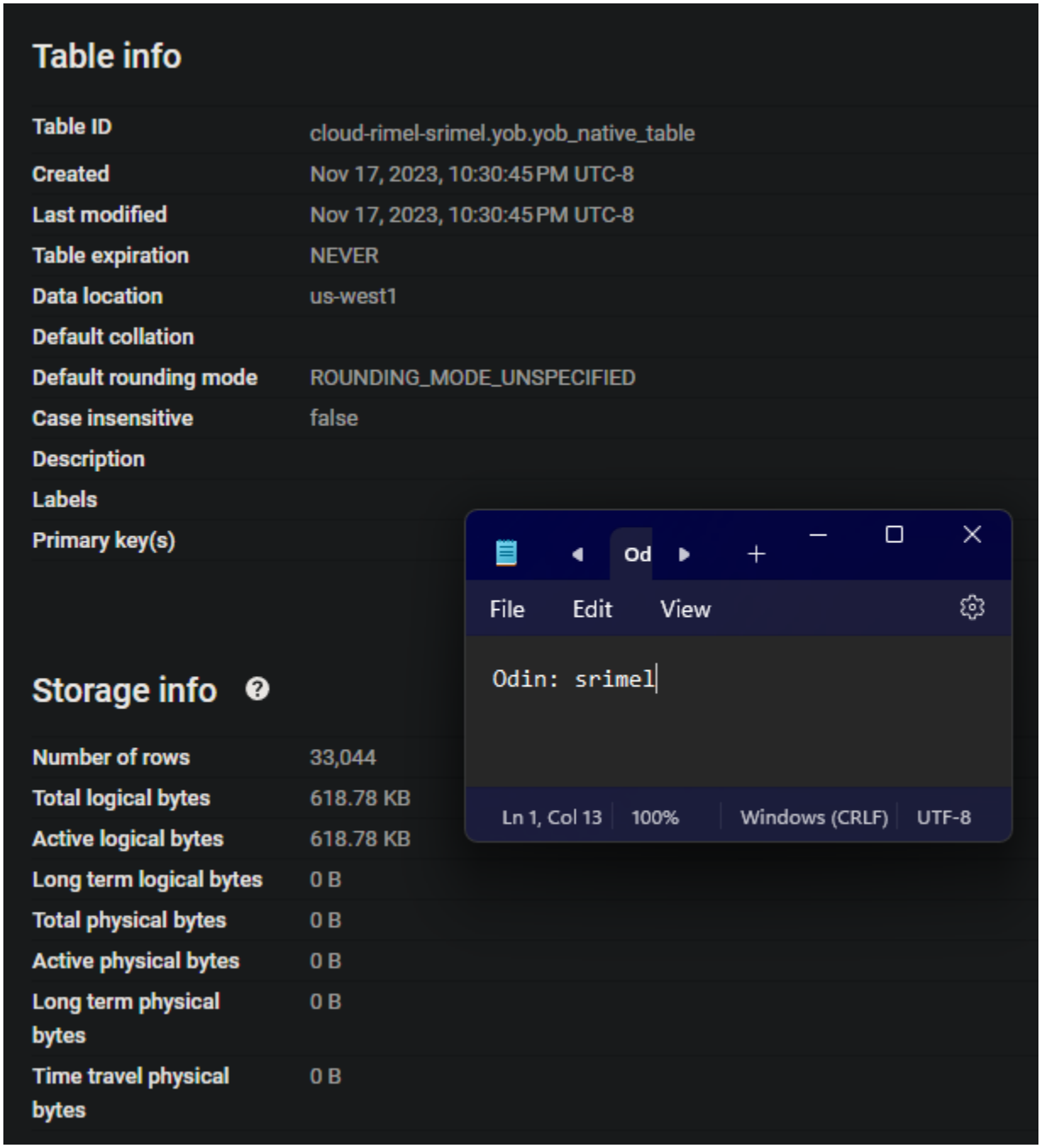
Stuart Rimel
CS530: Internet Web & Cloud, Fall 2023

Odin: srimel
** In all the terminal screenshots my Odin name is in the terminal prompt **

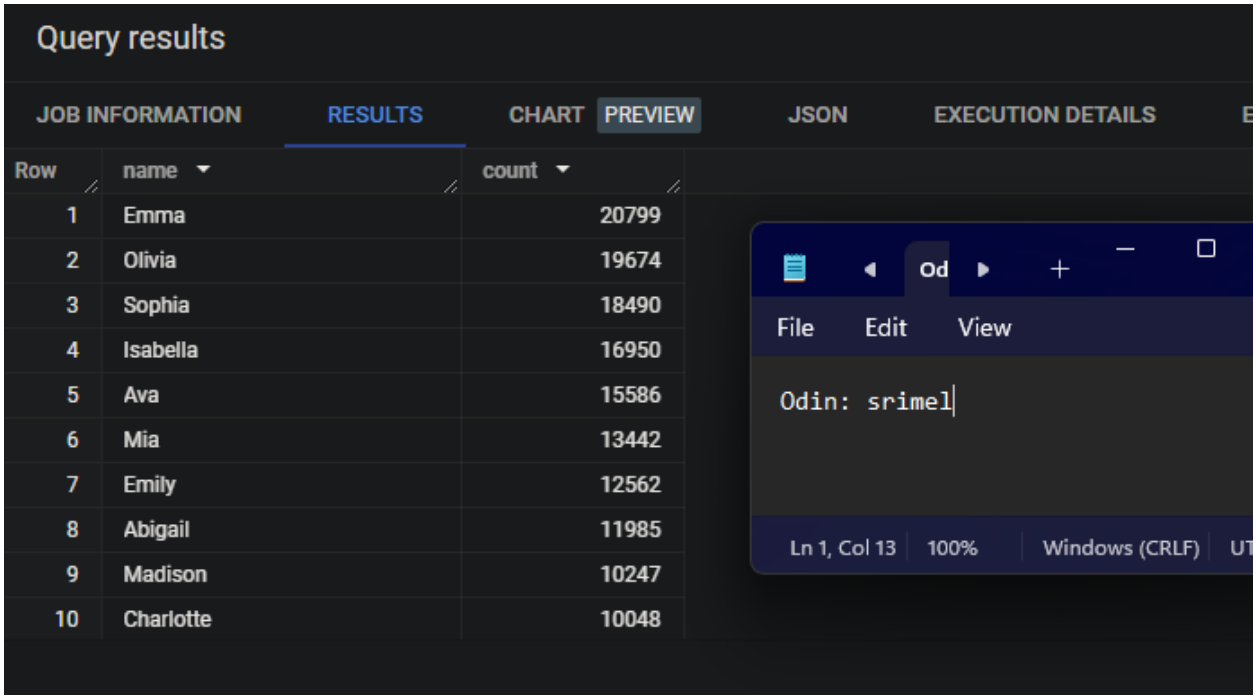
Table of contents:

9.1g: BigQuery, BigLake

9.1g.3: Create dataset



9.1g.4: Query data



Query results

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

Row	name	count
1	Emma	20799
2	Olivia	19674
3	Sophia	18490
4	Isabella	16950
5	Ava	15586
6	Mia	13442
7	Emily	12562
8	Abigail	11985
9	Madison	10247
10	Charlotte	10048

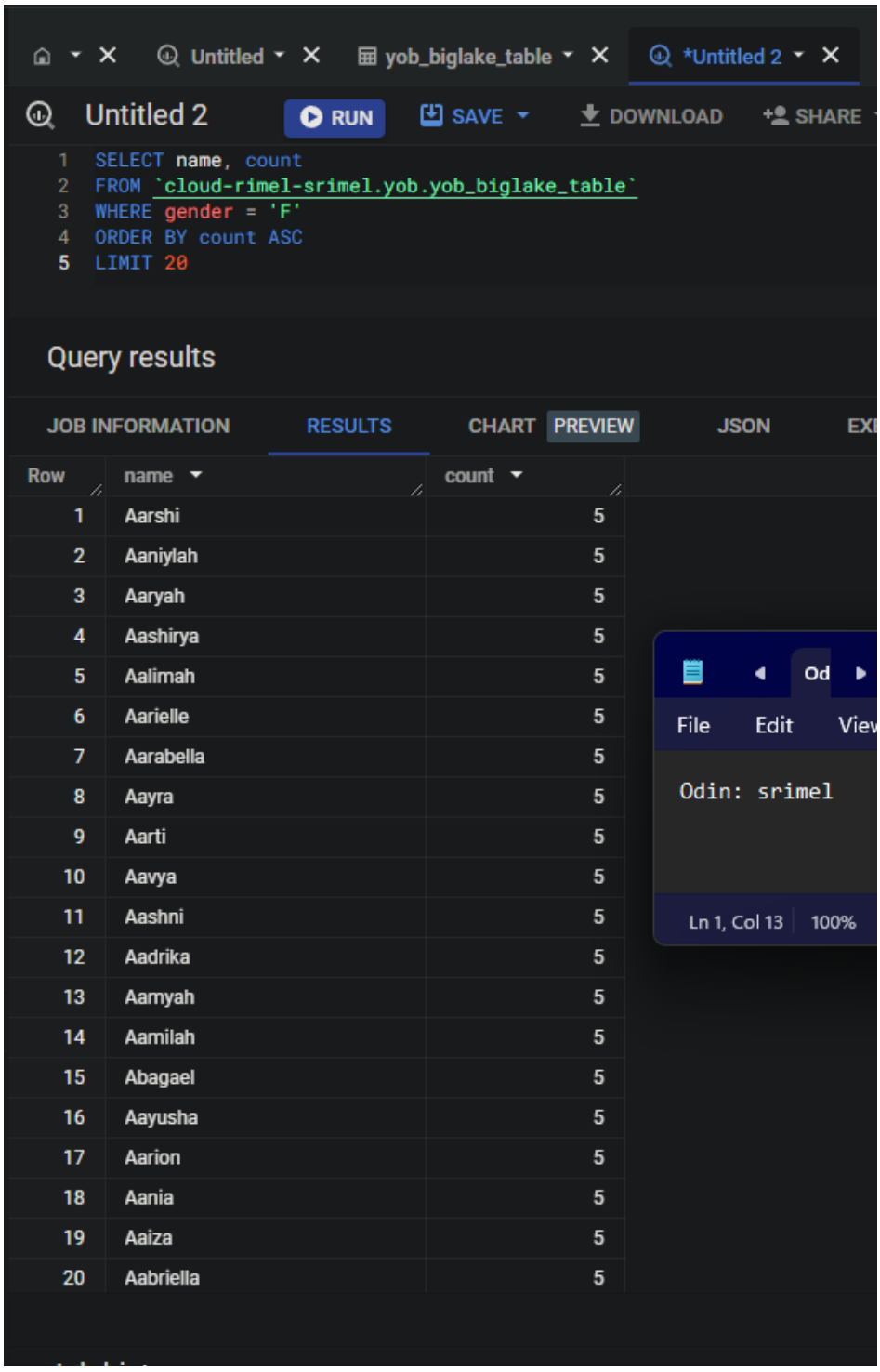

```
srimel@cloudshell:~ (cloud-rimel-srimel)$ bq query "SELECT name, count FROM [cloud-rimel-srimel.yob.yob_native_table] WHERE gender='M' ORDER BY count ASC LIMIT 10;"
+---------+-------+
|  name   | count |
+---------+-------+
| Aari    |     5 |
| Aaliyah |     5 |
| Aadian  |     5 |
| Aaroh   |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

```
cloud-rimel-srimel> select name, count from [cloud-rimel-srimel.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-rimel-srimel>
```

```
cloud-rimel-srimel> select name, count from [cloud-rimel-srimel.yob.yob_native_table] where name = 'Stuart';
+---------+-------+
|  name   | count |
+---------+-------+
| Stuart  |    82 |
+---------+-------+
cloud-rimel-srimel>
```

My name only had 82 :/

9.1g.9: Query data



9.2g: Jupyter Notebooks

9.2g.3: BigQuery query

How much less data: 18.89 GB

How many twins: 375,362

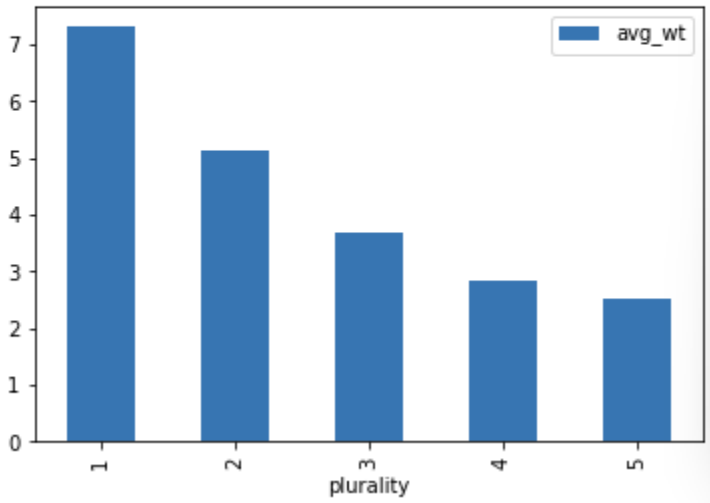
Lighter on average: 2.171

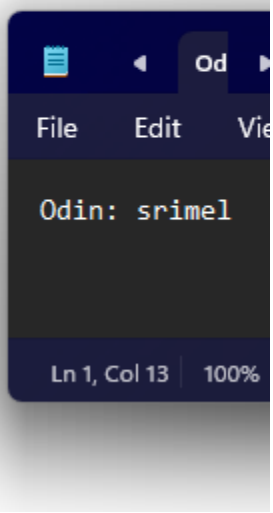
9.2g.6: Run queries

Two most important features:

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

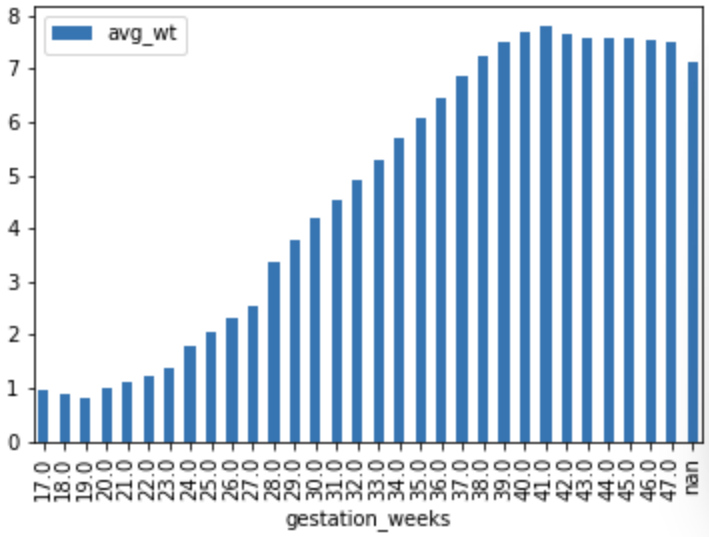
[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa72ae87dd0>
```

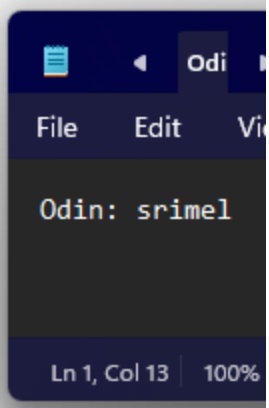




```
[8]: df = get_distinct_values('gestation_weeks')
     df.plot(x='gestation_weeks', y='avg_wt', kind='bar')
```

```
[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa72a156dd0>
```





9.2g.8: Mobility

Largest spike in trips to grocery/pharmacy: 2020-03-13

Workplace trips on stay-at-home order: -49%

9.2g.9: Airport traffic

Three airports impacted most in April 2020:

1. McCarran International
2. San Francisco International
3. Denver International

Three airports impacted most in August 2020:

1. McCarran International
2. Detroit Metropolitan Wayne County
3. San Francisco International

9.2g.10: Mortality

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

- Table: excess_deaths
    - Columns: placename, start_date, excess_deaths

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

- Table: us_counties
    - Columns: date, county, deaths

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

- Table: us_states
    - Columns: 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?

- Table: mask_use_by_county
    - Columns: county_fips_code, always

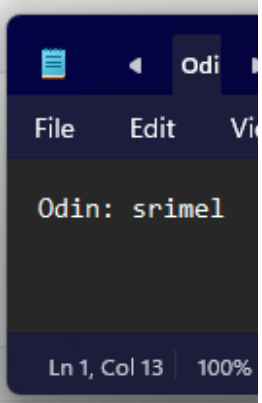
9.2g.11: Run example queries

[13]:
```
query_string = """
SELECT date, confirmed_cases
FROM `bigquery-public-data.covid19_nyt.us_states`
WHERE state_name = 'Oregon'
ORDER BY date ASC
"""
```

[14]:
```
from google.cloud import bigquery
df = bigquery.Client().query(query_string).to_dataframe()
df.head(3)
```

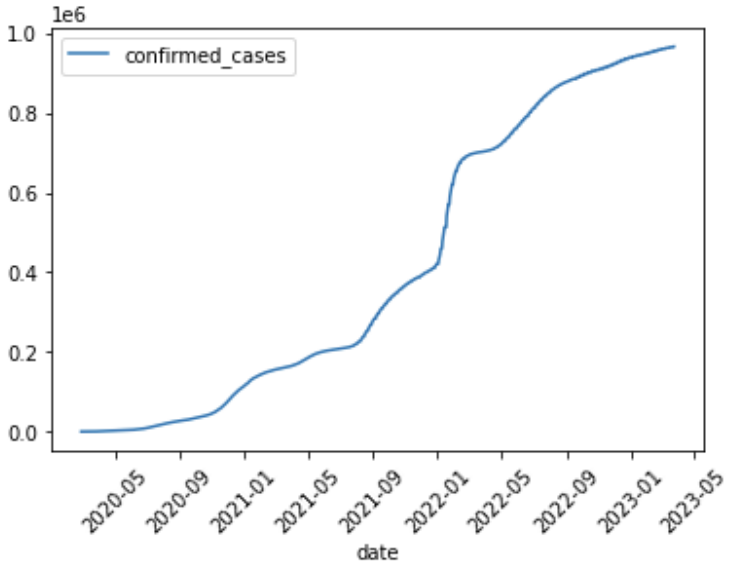
[14]:

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



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

[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa72a1fa2d0>
```



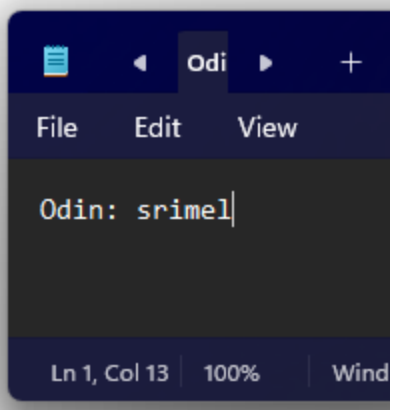
[16]:
```
def run_query(q_string):
    df = bigquery.Client().query(q_string).to_dataframe()
    return df
```

[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
"""

run_query(query_string).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	California	2020-04-17
7	Connecticut	2020-04-17
8	Pennsylvania	2020-04-17
9	Florida	2020-04-24

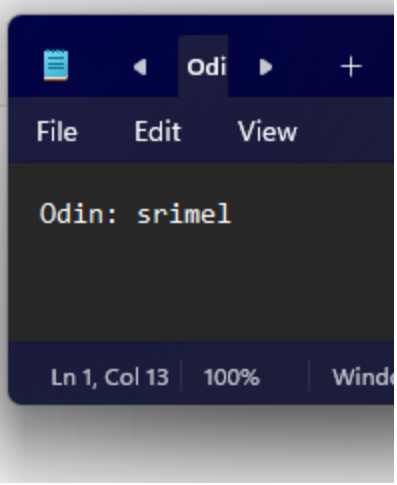


[20]:
```
query = """
SELECT DISTINCT mu.county_fips_code, mu.always, ct.county
FROM `bigquery-public-data.covid19_nyt.mask_use_by_county` as mu
LEFT JOIN `bigquery-public-data.covid19_nyt.us_counties` as ct
ON mu.county_fips_code = ct.county_fips_code
ORDER BY mu.always DESC
"""

run_query(query).head(5)
```

[20]:

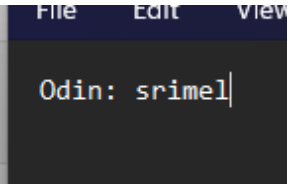
	county_fips_code	always	county
0	06027	0.889	Inyo
1	36123	0.884	Yates
2	48229	0.880	Hudspeth
3	06051	0.880	Mono
4	48141	0.877	El Paso



9.2g.12: Write queries

I created an abstraction ‘run_query’ to help with creating dataframes:

[16]:
```
def run_query(q_string):
    df = bigquery.Client().query(q_string).to_dataframe()
    return df
```

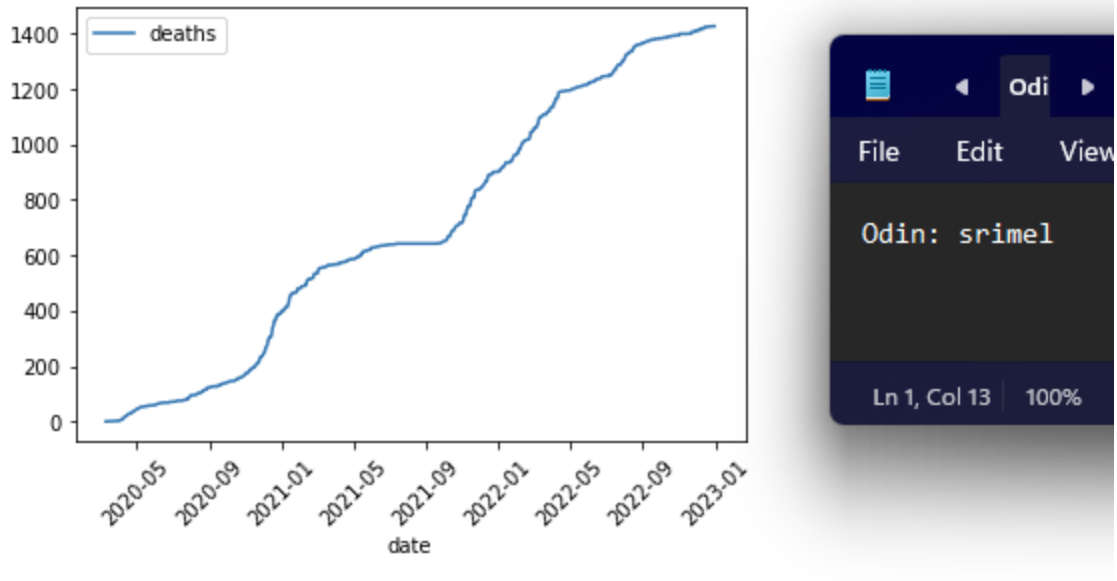


Deaths in Multnomah county
Construct a query string that obtains the number of deaths from COVID-19 that have occurred in Multnomah county for each day in the dataset, ensuring the data is returned in ascending order of date. Run the query and obtain the results.

[21]:
```
query = """
SELECT date, deaths
FROM `bigquery-public-data.covid19_nyt.us_counties`
WHERE county_fips_code = '41051' OR county = 'Multnomah'
ORDER BY date ASC
"""

run_query(query).plot(x='date', y='deaths', kind='line', rot=45)
```

[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa728575550>
```



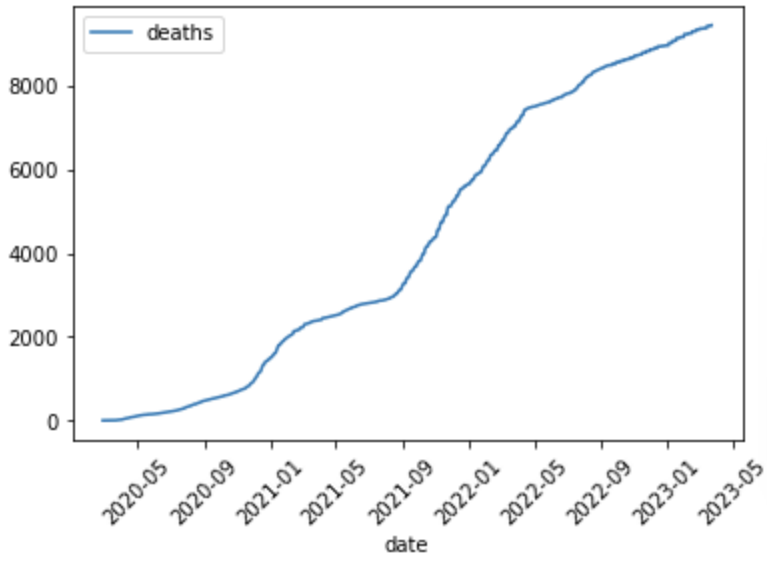
Deaths in Oregon

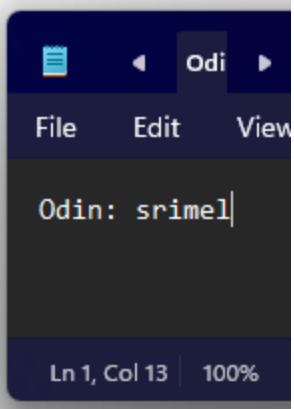
Construct a query string that obtains the number of deaths from COVID-19 that have occurred in Oregon for each day in the dataset, ensuring the data is returned in ascending order of date. Run the query and obtain the results.

[22]:
```
query = """
SELECT date, deaths
FROM `bigquery-public-data.covid19_nyt.us_states`
WHERE state_name = 'Oregon'
ORDER BY date ASC
"""

run_query(query).plot(x='date', y='deaths', kind='line', rot=45)
```

[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa72854d4d0>
```





9.3g: Dataproc

9.3g.6: Run computation

Job took around 2min to complete.

Pi is roughly 3.1415010714150107

9.3g.8: run computation again

Job took around 55 seconds to complete

Pi is roughly 3.1416717514167174

9.4g: Dataflow

9.4g.3: Beam code

1. Default input: ../javahelp/src/main/java/com/google/cloud/training/dataanalyst/javahelp/
2. Default output: /tmp/output
3. What operation does the 'PackageUse()' transform implement? I'm not sure exactly what this question is asking… 'PackageUse()' seems to be getting all the packages that were imported as 'import' is the keyword that it's looking for. This function calls getPackages which aggregates package names that were imported. Within getPackages, splitPackages is used to parse the package name out of the import statement.
4. TotalUse operation is implemented using CombinePerKey with the 'sum' operation
5. Which operations correspond to a "Map"?
    a. "GetJava", "GetImports", "PackageUse"
6. Which operation corresponds to a "Shuffle-Reduce"?
    a. "TotalUse"
7. Which operation corresponds to a "Reduce"?
    a. "Top_5"

9.4g.4: Run pipeline locally

```
(venv) srimel@cloudshell:~/source/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-rimel-srimel)$ python is_popular.py --output testrun
(venv) srimel@cloudshell:~/source/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-rimel-srimel)$ cat testrun-00000-of-00001
[('org', 45), ('org.apache', 44), ('org.apache.beam', 44), ('org.apache.beam.sdk', 43), ('org.apache.beam.sdk.transforms', 16)]
(venv) srimel@cloudshell:~/source/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-rimel-srimel)$
```

Results: [('org', 45), ('org.apache', 44), ('org.apache.beam', 44), ('org.apache.beam.sdk', 43), ('org.apache.beam.sdk.transforms', 16)]

The data returned represents a list of tuples consisting of package names and their total count found within the input directory. For example, 'org' is the most imported package with a total count of 45.

9.4g.5: Dataflow Lab #2 (Word count)

1. Stages:
    a. 'Read', 'Split', 'PairWithOne', 'GroupAndSum', 'Format', 'Write'
2. Descriptions
    a. 'Read'
        i. Reads the file specified by '--input'
    b. 'Split'
        i. Applies ParDo transformation, extracting words
    c. 'PairWithOne'

        i. Creates a key value pair of words with the number 1 as second tuple item
    d. ‘GroupAndSum’
        i. Aggregates work key-value pairs and takes a sum getting the word counts for each
    e. ‘Format’
        i. Formats the word key-value pairs into format specified by ‘format_result’. Format is '%s: %d'
    f. ‘Write’
        i. Writes the formatted words key-value pairs to ‘--output’ file.

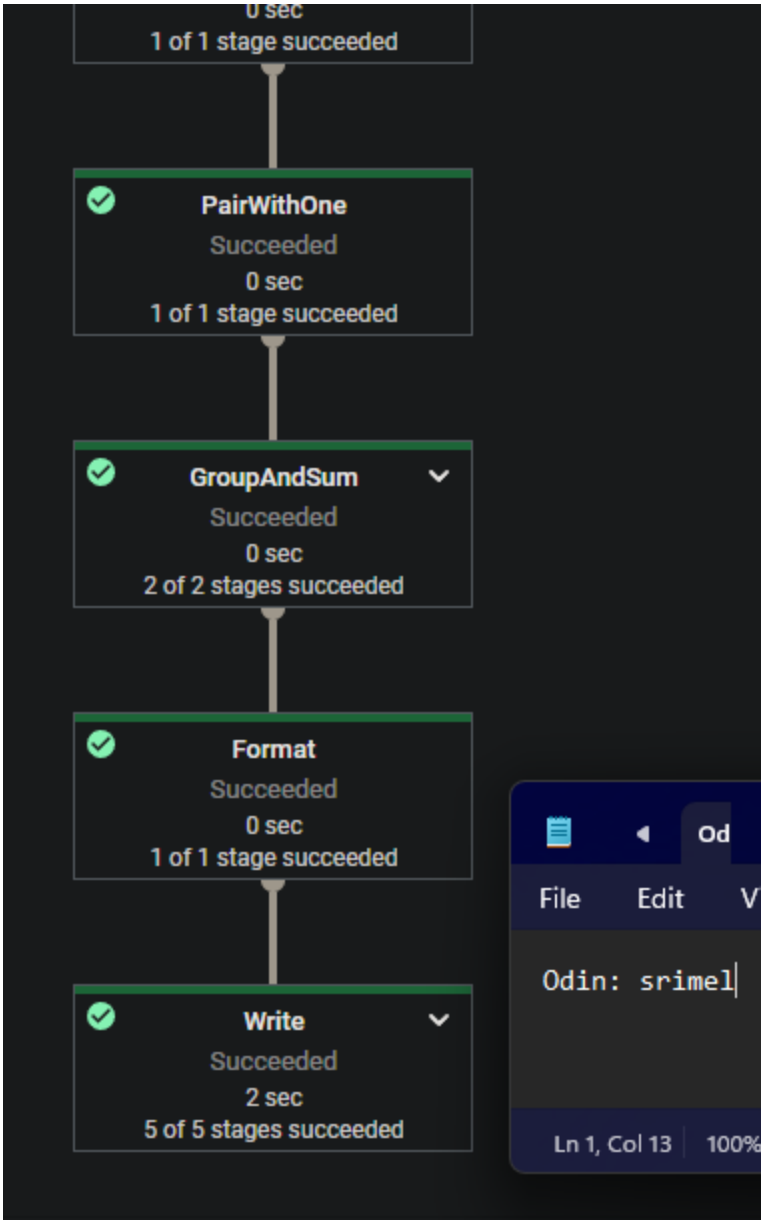
9.4g.6: Run code locally

1. Number of words in king lear: 4784
2. Command used: sort -t: -k2,2nr outputs-00000-of-00001 | head -n 3
    a. Results:
        the: 786
        I: 622
        and: 594
3. Added ‘lowercase’ stage and re-ran the command for sort
    a. Results:
        the: 908
        and: 738
        i: 622

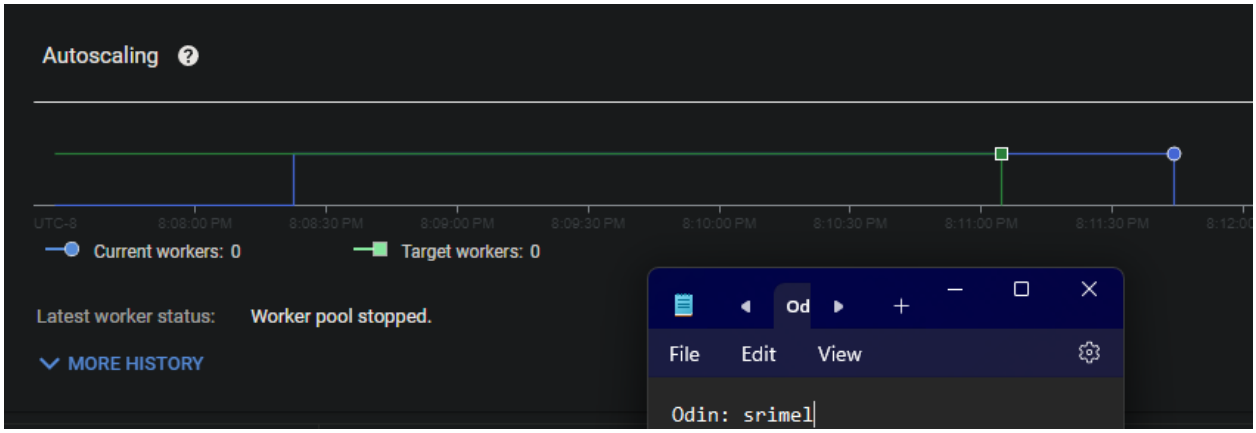
9.4g.9: Run code using Dataflow runner

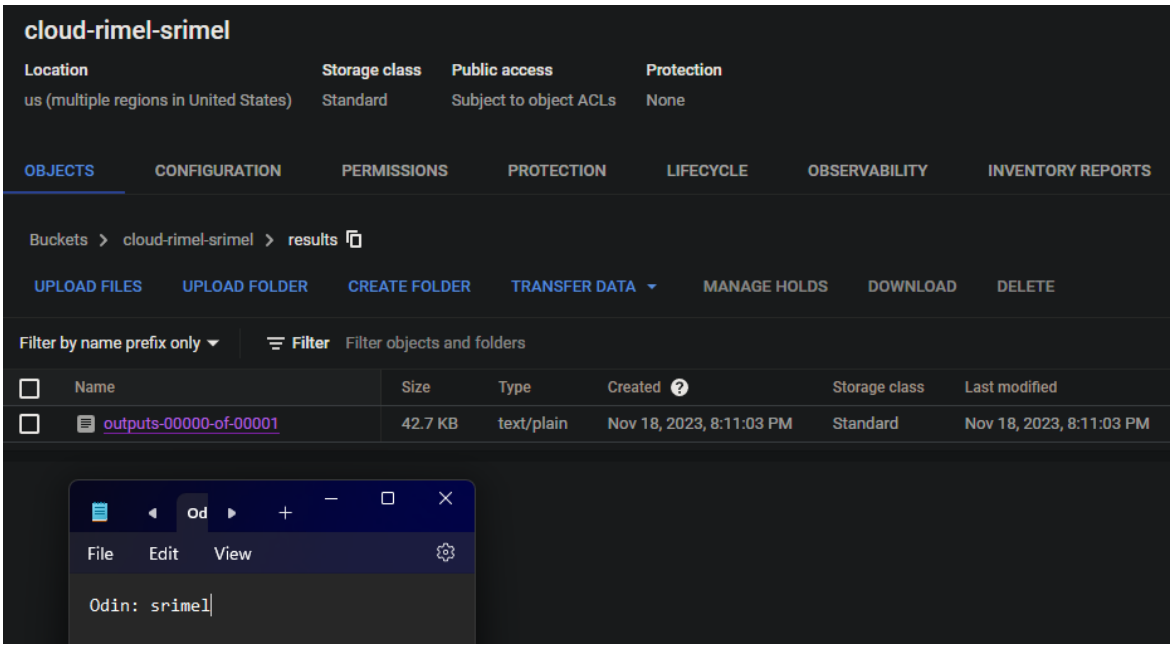
1. Part of job graph that took the longest:
    a. The ‘Write’ stage took the longest at 2 secs, all other stages show 0 secs

i.


0 sec
1 of 1 stage succeeded
PairWithOne
Succeeded
0 sec
1 of 1 stage succeeded
GroupAndSum
Succeeded
0 sec
2 of 2 stages succeeded
Format
Succeeded
0 sec
1 of 1 stage succeeded
Write
Succeeded
2 sec
5 of 5 stages succeeded
Od
File
Edit
Odin: srimel
Ln 1, Col 13

2.


Autoscaling
Current workers: 0
Target workers: 0
Latest worker status:
Worker pool stopped.
MORE HISTORY
Od
File
Edit
View
Odin: srimel

3. I'm only seeing 1 file in the 'output' ('results/' as specified in the dataflow runner command with '--output' flag) directory within my cloud storage:
    a. 

9.4g.12: View raw data from PubSub

```
srimel@cloudshell:~ (cloud-rimel-srimel)$ gcloud pubsub subscriptions pull taxisub --auto-ack
DATA: {"ride_id":"95460fa0-d998-4148-a749-96ebe210e9f2","point_idx":1801,"latitude":40.72784,"longi
tude":-73.88728,"timestamp":"2023-11-18T23:24:50.34888-05:00","meter_reading":36.9594,"meter_increm
ent":0.0205216,"ride_status":"enroute","passenger_count":1}
MESSAGE_ID: 9658710024556932
ORDERING_KEY:
ATTRIBUTES: ts=2023-11-18T23:24:50.34888-05:00
DELIVERY_ATTEMPT:
ACK_STATUS: SUCCESS
```

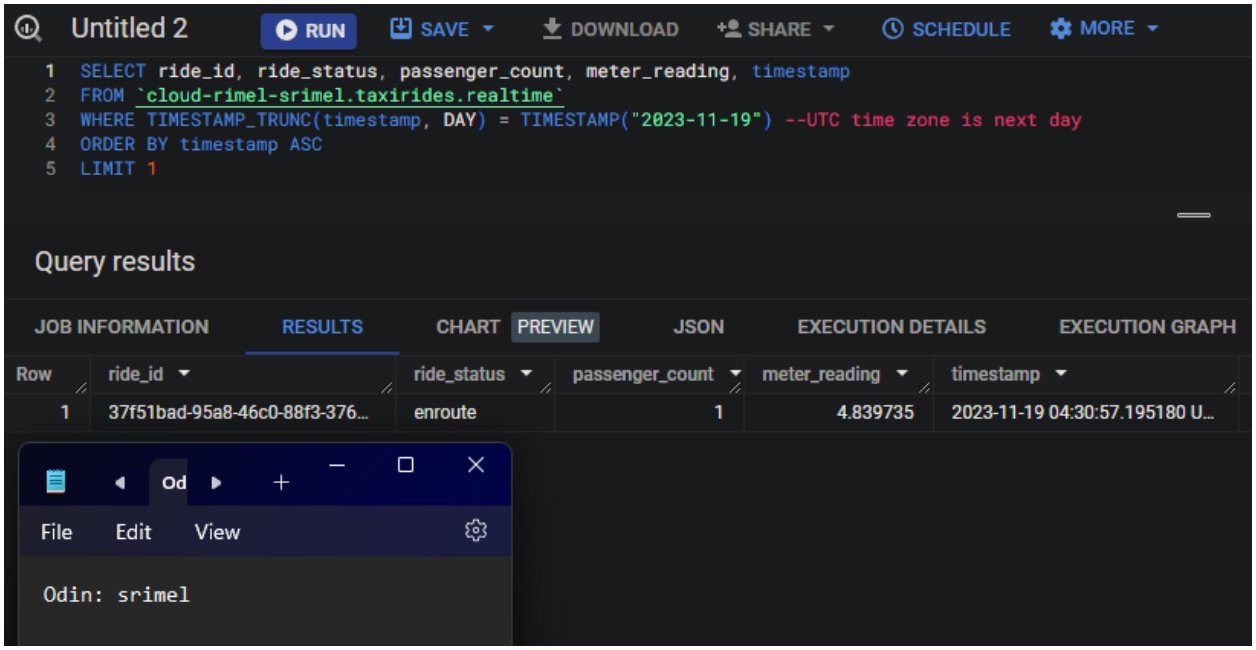
Fields for data object:

- ride_id, point_idx, latitude, longitude, timestamp, meter_reading, meter_increment, ride_status, passenger_count

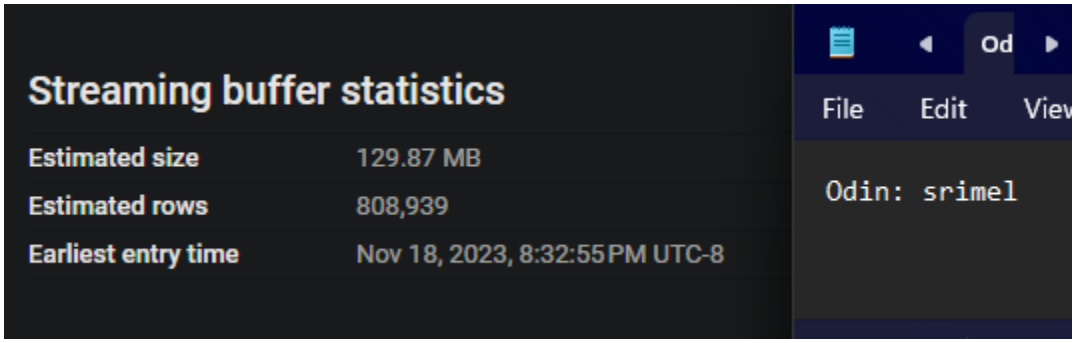
9.4g.14: Run Dataflow job from template

9.4g.15: Query data in BigQuery

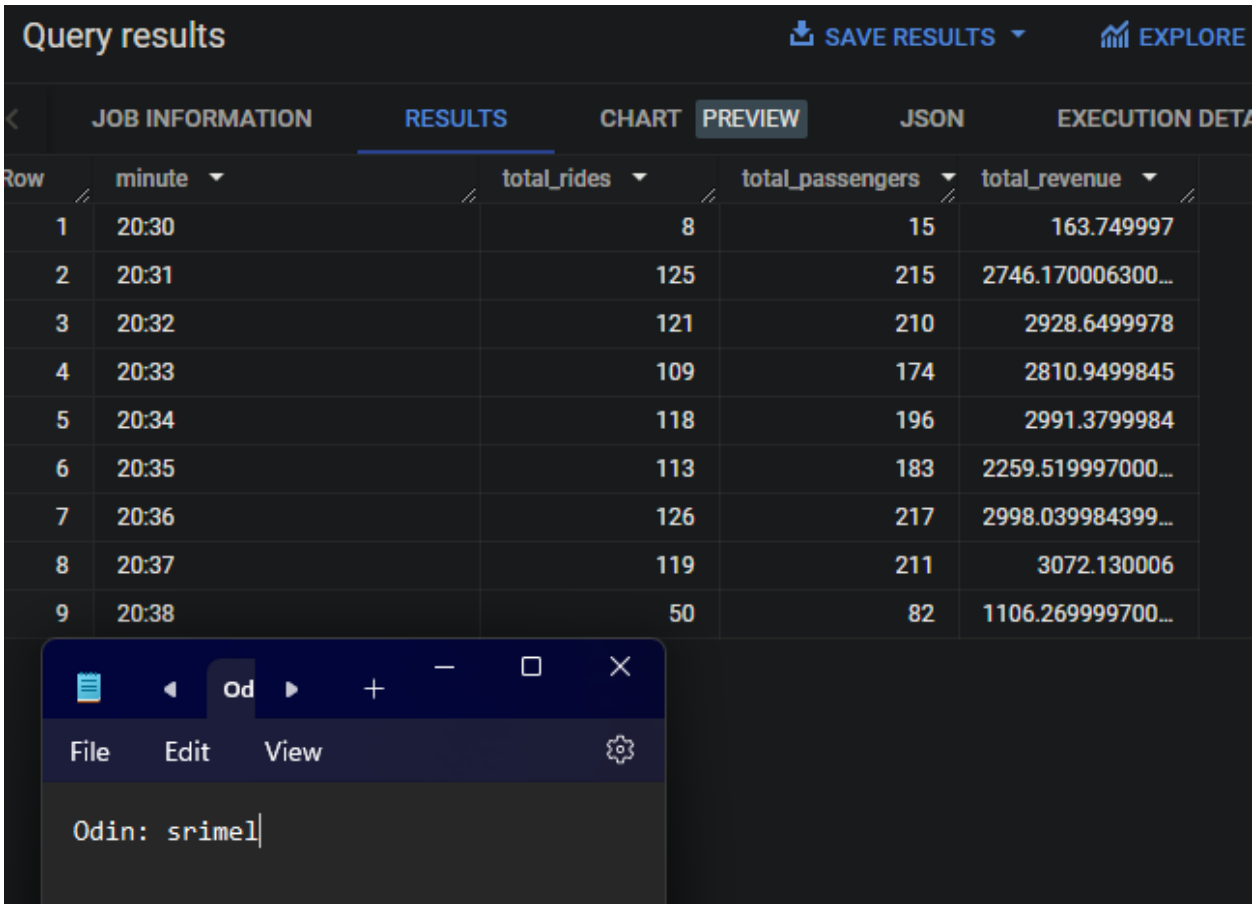
First ride query:



Estimated number of rows:



Rider per minute query:


Query results
SAVE RESULTS
EXPLORE
JOB INFORMATION
RESULTS
CHART
PREVIEW
JSON
EXECUTION DETA
Row
minute
total_rides
total_passengers
total_revenue
1 20:30 8 15 163.749997
2 20:31 125 215 2746.170006300...
3 20:32 121 210 2928.6499978
4 20:33 109 174 2810.9499845
5 20:34 118 196 2991.3799984
6 20:35 113 183 2259.519997000...
7 20:36 126 217 2998.039984399...
8 20:37 119 211 3072.130006
9 20:38 50 82 1106.269999700...
Od
File
Edit
View
Odin: srimel

9.4g.16: Data visualization