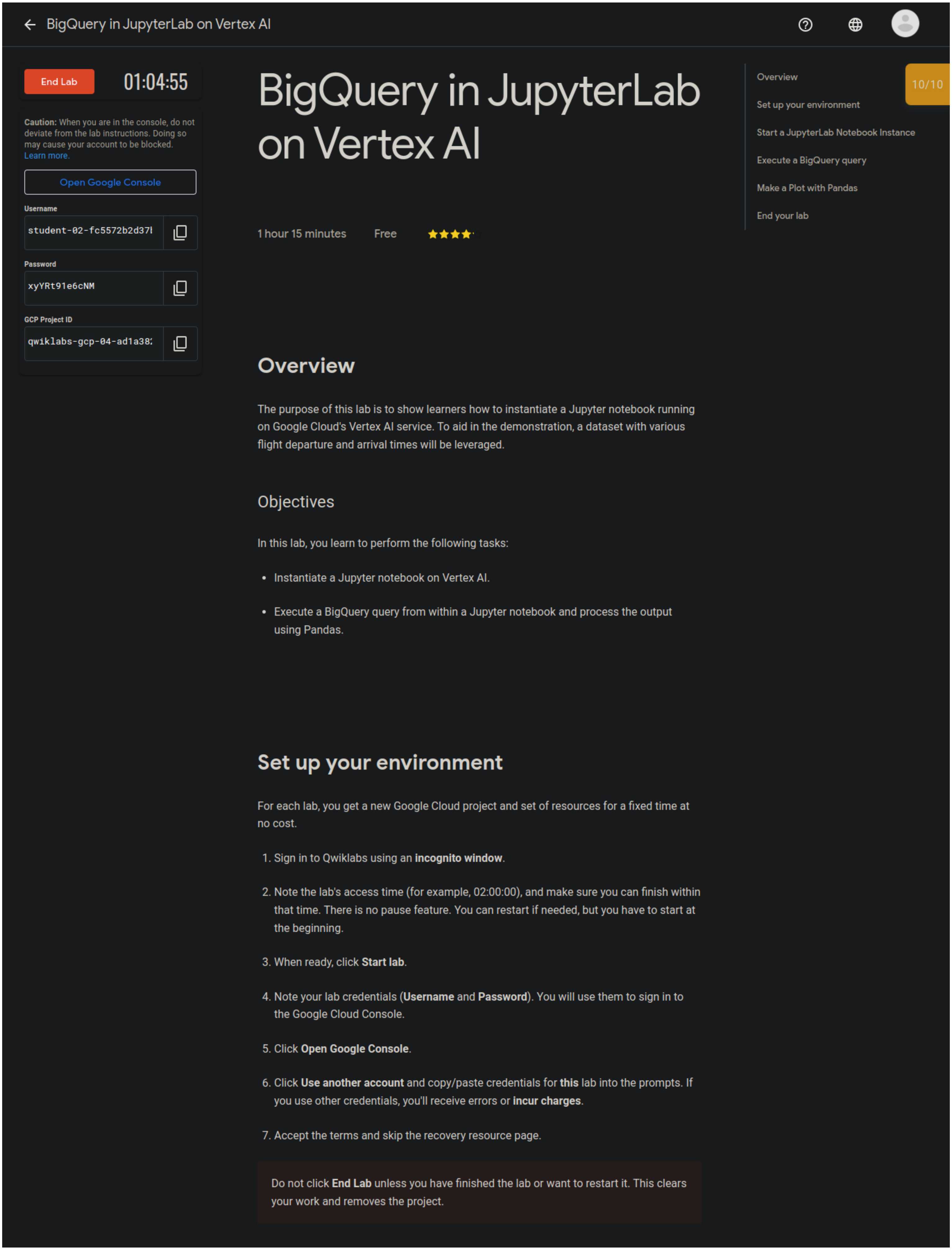


Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Username: student-02-fc5572b2d37

Password: xyYRt91e6cNM

GCP Project ID: qwiklabs-gcp-04-ad1a38

BigQuery in JupyterLab on Vertex AI

1 hour 15 minutes Free ★★★★

Overview

The purpose of this lab is to show learners how to instantiate a Jupyter notebook running on Google Cloud's Vertex AI service. To aid in the demonstration, a dataset with various flight departure and arrival times will be leveraged.

Objectives

In this lab, you learn to perform the following tasks:

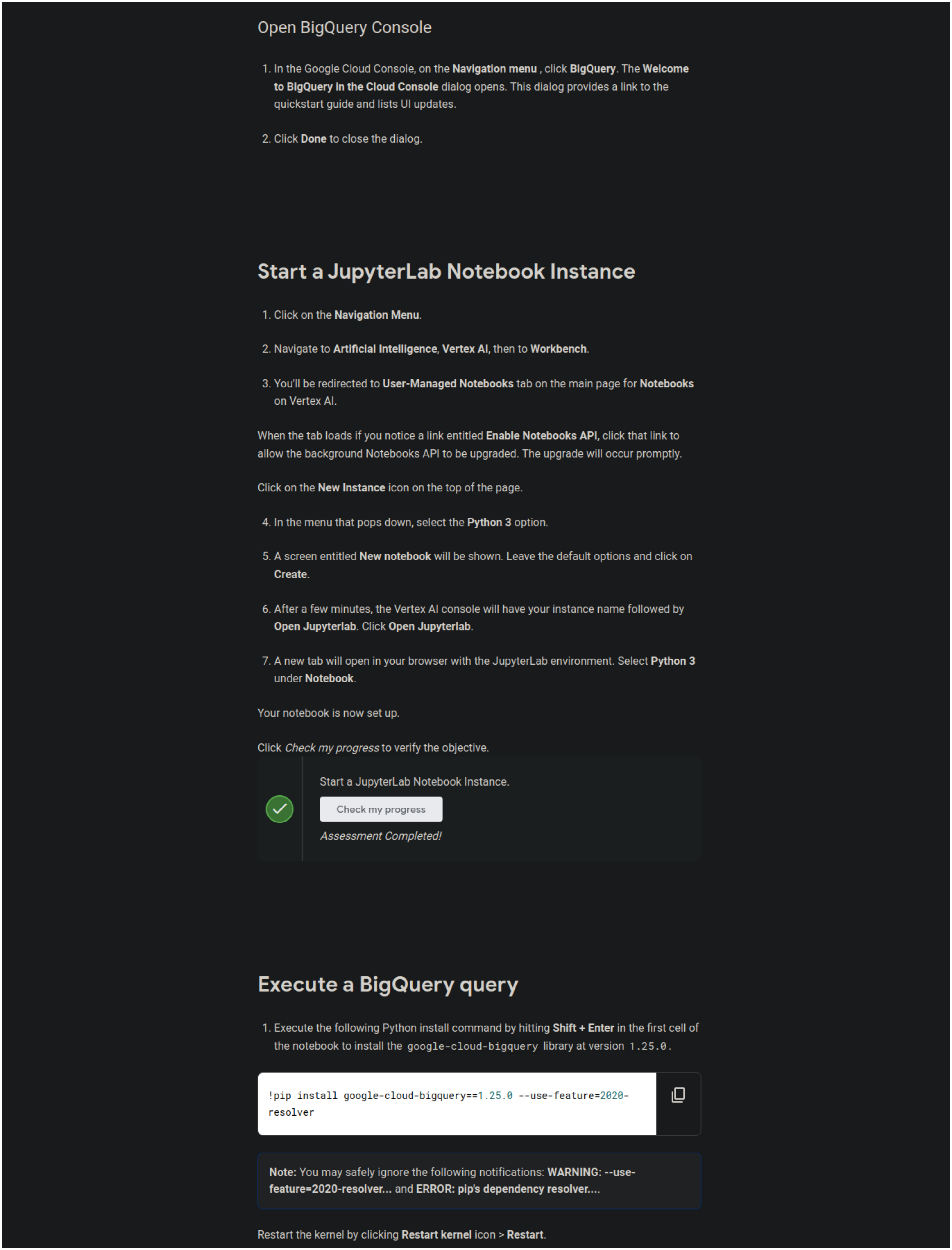
- Instantiate a Jupyter notebook on Vertex AI.
- Execute a BigQuery query from within a Jupyter notebook and process the output using Pandas.

Set up your environment

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, 02:00:00), and make sure you can finish within that time. There is no pause feature. You can restart if needed, but you have to start at the beginning.
3. When ready, click **Start lab**.
4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.
5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts. If you use other credentials, you'll receive errors or **incur charges**.
7. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.



Open BigQuery Console

1. In the Google Cloud Console, on the **Navigation menu** , click **BigQuery**. The **Welcome to BigQuery in the Cloud Console** dialog opens. This dialog provides a link to the quickstart guide and lists UI updates.
2. Click **Done** to close the dialog.

Start a JupyterLab Notebook Instance

1. Click on the **Navigation Menu**.
2. Navigate to **Artificial Intelligence**, **Vertex AI**, then to **Workbench**.
3. You'll be redirected to **User-Managed Notebooks** tab on the main page for **Notebooks** on Vertex AI.

When the tab loads if you notice a link entitled **Enable Notebooks API**, click that link to allow the background Notebooks API to be upgraded. The upgrade will occur promptly.

Click on the **New Instance** icon on the top of the page.

4. In the menu that pops down, select the **Python 3** option.
5. A screen entitled **New notebook** will be shown. Leave the default options and click on **Create**.
6. After a few minutes, the Vertex AI console will have your instance name followed by **Open Jupyterlab**. Click **Open Jupyterlab**.
7. A new tab will open in your browser with the JupyterLab environment. Select **Python 3** under **Notebook**.

Your notebook is now set up.

Click *Check my progress* to verify the objective.

✓ Start a JupyterLab Notebook Instance.

Check my progress

Assessment Completed!

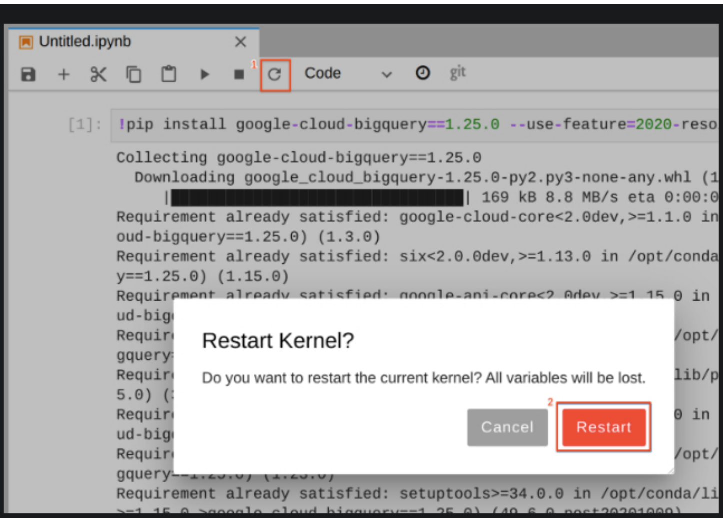
Execute a BigQuery query

1. Execute the following Python install command by hitting **Shift + Enter** in the first cell of the notebook to install the `google-cloud-bigquery` library at version `1.25.0`.

```
!pip install google-cloud-bigquery==1.25.0 --use-feature=2020-resolver
```

Note: You may safely ignore the following notifications: **WARNING: --use-feature=2020-resolver...** and **ERROR: pip's dependency resolver....**

Restart the kernel by clicking **Restart kernel** icon > **Restart**.



```
[1]: !pip install google-cloud-bigquery==1.25.0 --use-feature=2020-reso
     Collecting google-cloud-bigquery==1.25.0
       Downloading google_cloud_bigquery-1.25.0-py2.py3-none-any.whl (1
          |████████████████████████████████| 169 kB 8.8 MB/s eta 0:00:0
     Requirement already satisfied: google-cloud-core<2.0dev,>=1.1.0 in
     oud-bigquery==1.25.0) (1.3.0)
     Requirement already satisfied: six<2.0.0dev,>=1.13.0 in /opt/conda
     y==1.25.0) (1.15.0)
     Requirement already satisfied: google-api-core<2.0dev,>=1.15.0 in
     ud-big                                                            
     Requir                                                       /opt/
     gquery
     Requir                                                       lib/p
     5.0) (
     Requir                                                       0 in
     ud-big
     Requir                                                       /opt/
     gquery
     Requirement already satisfied: setuptools>=34.0.0 in /opt/conda/li
```

Restart Kernel?

Do you want to restart the current kernel? All variables will be lost.

2. Enter the following query in the second cell of the notebook.

```
%%bigquery df
SELECT
  depdelay as departure_delay,
  COUNT(1) AS num_flights,
  APPROX_QUANTILES(arrdelay, 10) AS arrival_delay_deciles
FROM
  `cloud-training-demos.airline_ontime_data.flights`
WHERE
 depdelay is not null
GROUP BY
  depdelay
HAVING
  num_flights > 100
ORDER BY
  depdelay ASC
```

The command makes use of the magic function `%%bigquery`. Magic functions in notebooks provide an alias for a system command. In this case, `%%bigquery` runs the query in the cell in BigQuery and stores the output in a Pandas DataFrame object named `df`.

3. Run the cell by hitting **Shift + Enter**, when the cursor is in the cell. Alternatively, if you navigate to the **Run** tab you can click on **Run Selected Cells**. Note the keyboard shortcut for this action in case it is not Shift + Enter. There should be no output when executing the command.

Click *Check my progress* to verify the objective.

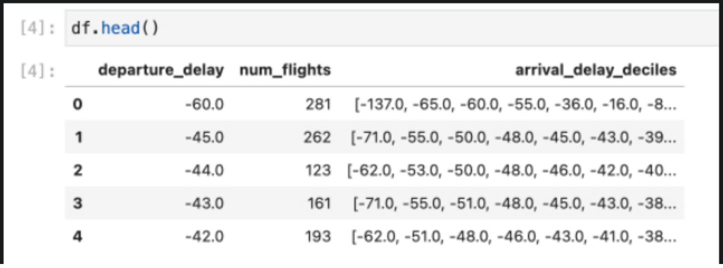
✓ Execute a BigQuery query

Check my progress

Assessment Completed!

4. View the first five rows of the query's output by executing the following code in a new cell:

```
df.head()
```



```
[4]: df.head()
```

[4]:

	departure_delay	num_flights	arrival_delay_deciles
0	-60.0	281	[-137.0, -65.0, -60.0, -55.0, -36.0, -16.0, -8...
1	-45.0	262	[-71.0, -55.0, -50.0, -48.0, -45.0, -43.0, -39...
2	-44.0	123	[-62.0, -53.0, -50.0, -48.0, -46.0, -42.0, -40...
3	-43.0	161	[-71.0, -55.0, -51.0, -48.0, -45.0, -43.0, -38...
4	-42.0	193	[-62.0, -51.0, -48.0, -46.0, -43.0, -41.0, -38...

Make a Plot with Pandas

We're going to use the Pandas DataFrame containing our query output to build a plot that depicts how arrival delays correspond to departure delays. Before continuing, if you are unfamiliar with Pandas the Ten Minute Getting Started Guide is recommended reading.

1. To get a DataFrame containing the data we need we first have to wrangle the raw query output. Enter the following code in a new cell to convert the list of `arrival_delay_deciles` into a Pandas Series object. The code also renames the resulting columns.

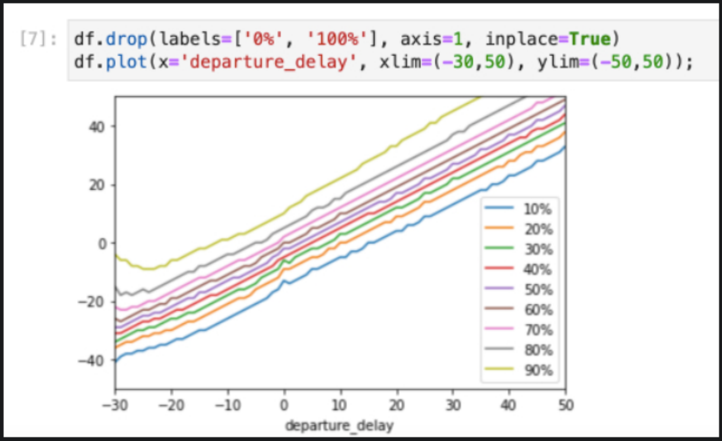
```
import pandas as pd
percentiles = df['arrival_delay_deciles'].apply(pd.Series)
percentiles.rename(columns = lambda x : '{0}%'.format(x*10),
inplace=True)
percentiles.head()
```

2. Since we want to relate departure delay times to arrival delay times we have to concatenate our `percentiles` table to the `departure_delay` field in our original DataFrame. Execute the following code in a new cell:

```
df = pd.concat([df['departure_delay'], percentiles], axis=1)
df.head()
```

3. Before plotting the contents of our DataFrame, we'll want to drop extreme values stored in the `0%` and `100%` fields. Execute the following code in a new cell:

```
df.drop(labels=['0%', '100%'], axis=1, inplace=True)
df.plot(x='departure_delay', xlim=(-30,50), ylim=(-50,50));
```

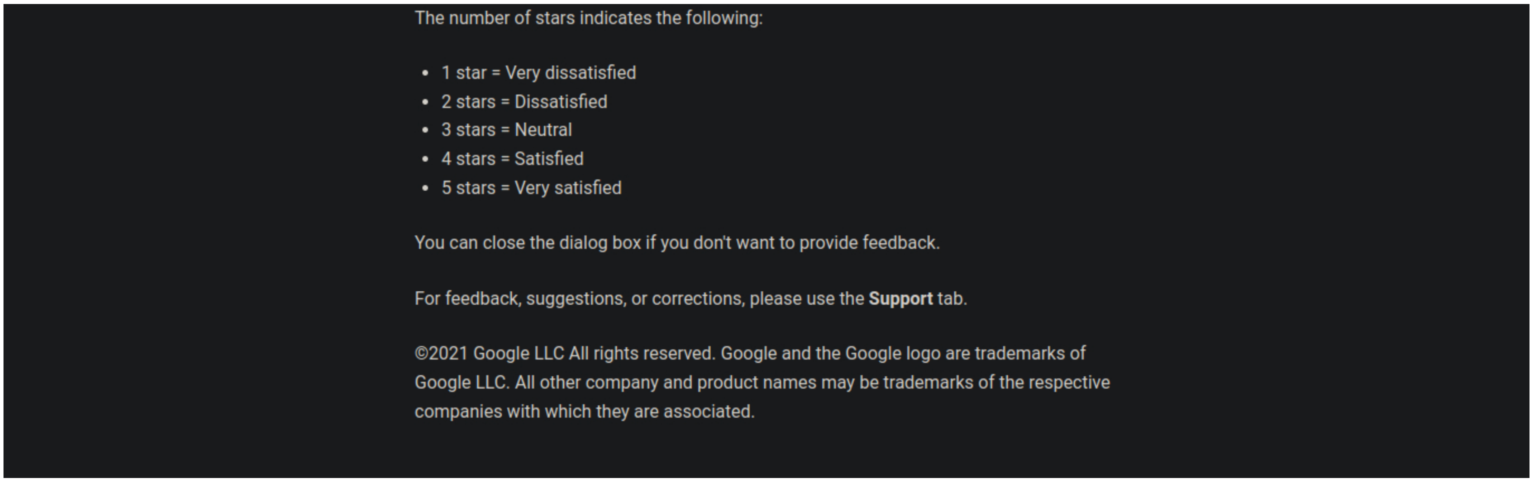


```
[7]: df.drop(labels=['0%', '100%'], axis=1, inplace=True)
     df.plot(x='departure_delay', xlim=(-30,50), ylim=(-50,50));
```

End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you've used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.



The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.