QLabs Overview

BA770 Lab Session

Questrom School of Business, Boston University

August 21, 2019

QLabs Overview

- You may follow the bullet points in this slides to review a QLab after you complete it.
- Key takeaways are marked with an asterisk before index.

Lecture1, QLabA
A Tour of Qwiklabs and the Google Cloud Platform

a) Gain an overall understanding of Qwiklabs platform and identify key features of a lab environment.

b) Access the GCP console with (temporary) specific credentials.

⋆ c) Know the definition of GCP projects.

⋆ d) Use the GCP navigation menu to identify types of GCP services.

e) Learn about primitive roles and use the Cloud IAM service to inspect actions available to specific users.

⋆ f) Learn basic Cloud Shell commands; run commands like touch, nano, and cat to create, edit, and output the content of files.

g) Learn about the API library and examine its chief features.

Lecture1, QLabA, Tips

a) Make sure you're using the temporary account to log in!

b) If you find something wrong or you get stuck in a session, don't be nervous - end the current session and open a new one. Redoing a lab will not influence your QLab grades.

c) Be careful with all keyboard and/or mouse operations when you are on the lab page. Do not click the End Lab button until you have completed all the tasks.

d) Learn more about applications provided by GCP, including virtual machine, storage, database, etc.

Lecture1, QLabB
Getting Started with Cloud Shell & gcloud

a) Gain an understanding of Google Cloud Shell, including definition, function, feature, etc.

⋆ b) Be familiar with **cd** command and **vi** editor.

c) Use gcloud commands to view configurations.

⋆ d) Use gsutil commands to manage Cloud Storage resources; know how to create a bucket and copy an existing file to a bucket.

Lecture1, QLabB, Tips

a) Bucket names are universally unique, so avoid using names like 'my_bucket' 'test_bucket'; otherwise you may receive 'ServiceException: 409 Bucket xxx already exists'.

b) You may use '**Ctrl+C**' to exit the current command.

c) Find **vi** documentation here.

d) Refer to documentation (you could just google it) to know more about **gcloud** and **gsutil** commands.

Lecture1, QLabC*
Creating a Virtual Machine

⋆ a) Create a virtual machine with 1) the GCP Console, 2) **gcloud** command line.

b) Check existing instances in Navigation menu - Compute Engine - VM instances.

⋆ c) Access the virtual machine by 1) launching a SSH client directly from browser, or 2) SSH'ing into the instance using gcloud.

Lecture1, QLabC*, Tips

a) Be aware of the window you're typing in:
   - When you are SSH'ing into an instance with a prompt window, make sure you execute the commands in this window.
   - If you are using **gcloud** commands, make sure the commands are running in the Cloud Shell.

b) If you get root access to your instance by mistake, press '**Ctrl+D**' to exit root user.

c) Learn more about SSH here.

d) Learn more about **sudo** command here.

e) Learn more about commonly used commands in GCP here.

Lecture2, QLabD
Introduction to SQL for BigQuery and Cloud SQL

- ⋆ a) Understand the relationship among project, database, and table.
- ⋆ b) Get familiar with BigQuery console, e.g. loading databases and tables into BigQuery.
- ⋆ c) Use SELECT, FROM, WHERE, COUNT, GROUP BY, AS, and ORDER BY keywords to fetch meaningful data from datasets.
- d) Export query results to local repository and upload them to Cloud Storage bucket.
- e) Create a new Cloud SQL instance and load files from Cloud Storage bucket as new tables.
- ⋆ f) Know basic knowledge of keywords CREATE, DELETE, INSERT INTO, and UNION. Be able to run queries in Cloud SQL using Cloud Shell.

Lecture2, QLabD, Tips

a) Make sure you have switched to the temporary account, and you are under the temporary project. Double check from time to time.

b) Leave enough time for file importing, instance creation and connecting, etc. You are strongly advised to read all instructions before starting the lab.

c) In Query editor, you could press '**Ctrl+Enter**' instead of clicking '**run**'.

d) Remember to rename the two csv documents.

e) After you enter '**gcloud sql connect qwiklabs-demo –user=root**' in Cloud Shell and wait for a while, you will see '**Connecting to database with SQL user [root]. Enter password:**'. Be aware that there's no flash cursor when you are typing. Enter your password and press 'Enter'.

f) When you are running SQL queries using Cloud Shell Command Line, remember to end each query with a semicolon '**;**'.

g) When importing files to tables in SQL instance, click Browse, and then double click the bucket name to find your files.

Lecture2, QLabD, SQL Keywords/Functions Summary

- **SELECT**: Specify the fields you want to pull from the dataset.
- **FROM**: Specify what table or tables to pull data from.
- **GROUP BY**: Aggregate result-set rows that share common criteria and return all of the unique entries found for such criteria.
- **COUNT**: Return the number of rows that share the same criteria.
- **AS**: Create an alias of a table or column.
- **ORDER BY**: Sort the returned data from a query in ascending or descending order based on a specified criteria or column value.
- **CREATE**: Create new databases or tables.
- **DELETE**: Delete existing databases or tables.
- **INSERT INTO**: Insert values into tables.
- **UNION**: Combine the output of two or more queries into a result set.

Lecture2, QLabE*
Using BigQuery in the GCP Console

⋆ a) Query a public dataset using aggregation functions.

⋆ b) Create a dataset in your project and load the data into a table.

c) Query your own table using GROUP BY and ORDER BY keywords.

Lecture2, QLabE*, Tips

a) After you click '**VIEW DATASET**', make sure you are still using the temporary google account and under the temporary project.

b) '**CREATE DATASET**' button is in blue font, right under the green check mark icon if your Query editor is not hidden. '**CREATE TABLE**' button is in a similar place.

Lecture3, QLabF
Analyzing Natality Data Using Datalab and BigQuery

- ⋆ a) Launch Cloud Datalab with Cloud Shell command.
- ⋆ b) Enable BigQuery client in Datalab notebook, execute SQL queries, and generate pandas dataframes.
- c) Create charts in Datalab.

Lecture3, QLabF, Tips

a) Be patient when waiting for the Datalab instance to launch. Remember to check the progress and to press Enter when responses are required.

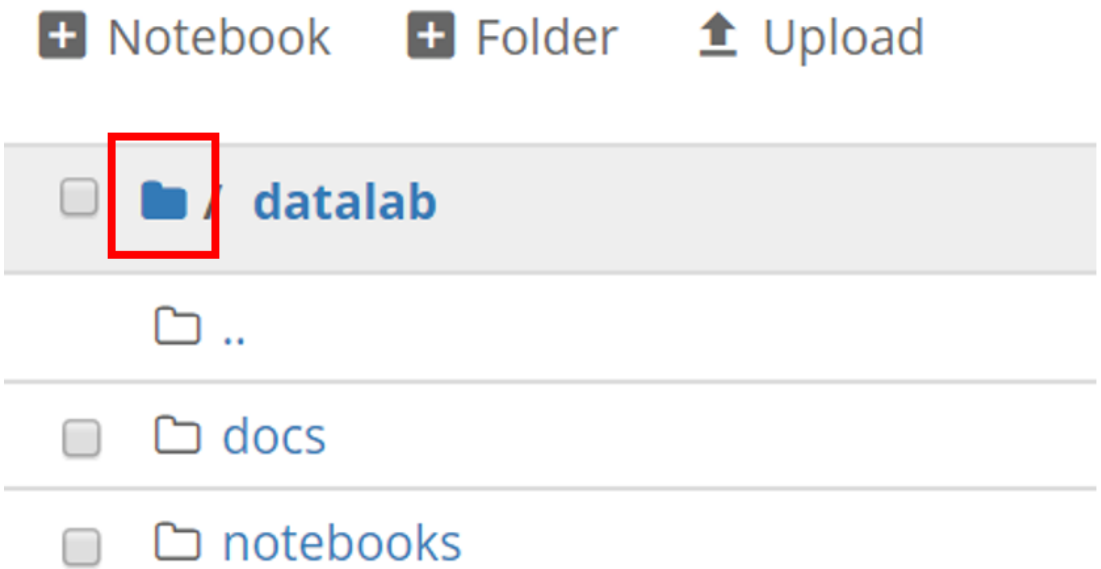
b) BigQuery result that you may refer to:

Row	plurality	num_babies	ave_weight
1	2	507706	5.16662858551256
2	3	27697	3.7188113817178317
3	5	325	2.625698693765901
4	1	15736332	7.336915793502335
5	4	1846	2.842509406912898

Lecture3, QLabF, Tips

c) A convenient way to open Datalab is to click the link 'http://localhost:8081/'. You can find the link in the output, right after 'Waiting for Datalab to be reachable at'. In this way you don't need to change port manually.

d) If you meet an error saying 'Creating Notebook Failed', click the solid folder icon to enter folder 'datalab' again, and then try starting a new notebook again.



Lecture3, QLabF, Tips

e) If you happen to lose connection to your Datalab instance (for example the error below):

Error: Could not connect to Cloud Shell on port 8081.

Ensure your server is listening on port 8081 and try again.

Dont worry, enter the following command in Cloud Shell:

datalab connect babyweight

Press Enter and wait for a short moment, and then youll be able to open Datalab again.

f) In case you meet errors when running cells in the notebook, shutdown the notebook by clicking Running Sessions in the upper right corner and open it again.

Lecture3, QLabF, Tips

g) You may use the following command in Cloud Shell to remove the passphrase of the SSH key:

rm .ssh/google_compute_engine*

You may press '**Tab**' to autocomplete the command.

h) Here is a cheat sheet of Notebook shortcuts. You may use these shortcuts to improve efficiency.

i) Download the notebooks in your datalab instance to your laptop from time to time, in case you may lose the instance by accident.

j) Use CAST, CONCAT and EXTRACT functions to create and manipulate SQL timestamps.

Lecture3, QLabG
Weather Data in BigQuery

a) Get a sense of GCP's great benefits for big data processing.

⋆ b) Combine and run analytics on multiple datasets.

⋆ c) Understand nested query statements with multiple functions; get familiar with table and column alias.

Lecture3, QLabG, Tips

a) COUNT(1): equivalent to COUNT(*), which returns all rows whether they are null or not null.
COUNT(Column): return all non-null rows.

b) Default JOIN type in BigQuery: INNER JOIN

c) Documentation for reference:
   1) JOIN
   2) Aggregate functions (e.g. AVG)
   3) Date functions (e.g. DATE)
   4) Mathematical functions (e.g. ABS)
   5) Statistical Aggregate Functions (e.g. CORR)
   6) Subquery

Lecture3, QLabH*
Exploring NCAA Data with BigQuery

a) Review how to view a table's basic information with Schema, Details, Preview panels.

b) Write and execute queries using WHERE, GROUP BY, and ORDER BY to answer specific questions.

Lecture4, QLabI
Exploring Your Ecommerce Dataset with SQL in BigQuery

a) Use Schema, Details, and Review tabs to get a sense of the dataset.

⋆ b) Detect duplicate records using COUNT, GROUP BY, and HAVING.

⋆ c) Write and execute basic SQL queries on ecommerce data using SELECT, FROM, WHERE, GROUP BY, ORDER BY, and LIMIT.

Lecture4, QLabI, Tips

a) NoOps: Short for No Operations.

b) We cannot use an aggregation function within a WHERE clause. We use HAVING clause instead.

c) GROUP BY 1: group by the first column regardless of what it's called. You may replace 1 with any number n indicating the n-th column. The same with ORDER BY.

d) We are going to learn more about WITH clause and Common Table Expression soon. For now, you may just get a sense of how WITH clause works.

e) Selecting aggregation results along with other columns using '**SELECT columns, AVG(column)**' will lead to error. This is because other columns must appear in a GROUP BY clause or be used in an aggregate function.

Lecture4, QLabJ
Troubleshooting Common SQL Errors with BigQuery

a) Look into a public dataset to gain some insights.

⋆ b) Use the BigQuery Query editor and Validator to troubleshoot SQL syntax errors.

⋆ c) Troubleshoot SQL logical errors: use of aggregation functions, GROUP BY, ORDER BY, WHERE vs HAVING.

Lecture4, QLabJ, Tips

a) Legacy SQL vs Standard SQL: Here we talk about the difference in syntax when referencing tables or project names.

   Legacy SQL: Use square brackets to start and end the table name, and use a colon (:) to delimit dataset and table names.
   Format: [bigquery-public-data:samples:natality]

   Standard SQL: Use the backtick character (`) to start and end the table name, and use the period character (.) to delimit dataset and table names.
   Format: `bigquery-public-data.samples.natality`

b) The AS keyword is optional.

Lecture5, QLabK*
Explore and Create Reports with Data Studio

a) Look into a public dataset to gain some insights.

⋆ b) Use the BigQuery Query editor and Validator to troubleshoot SQL syntax errors.

⋆ c) Troubleshoot SQL logical errors: use of aggregation functions, GROUP BY, ORDER BY, WHERE vs HAVING.

Lecture5, QLabK*, Tips

a) In the preview page, change the type of ratio field by selecting **'Numeric'–'Number'–'Percent'**.

b) '**Reporting tools tray**' is under the menu bar, beginning with the '**Add a page**' button.

c) Pay attention to the different fields (Data Source, Dimension, Metric, Sort, Filter) in Data tab when you are dragging columns to add them to corresponding fields.

d) More references:
   1) Data Studio Help Center.
   2) Tutorial: Create a new report.
   3) Connect Data Studio to BigQuery tables.
   4) Transform, categorize, and do math with your data.
   5) Video: Connect to Your Data.

Lecture5, QLabL*
How to Build a BI Dashboard Using Google Data Studio and BigQuery

a) Run a simple query against the trees table using SELECT, FROM, WHERE, IS (NOT) NULL, AND/OR, GROUP BY, and ORDER BY.

⋆ b) Create a new scheduled query to update data on a recurring basis.

c) Play with Data Studio Charts to find out more!

Important note: Save a PDF copy of your final report as the proof of a complete lab in order to get full credit for this Qlab.

Lecture5, QLabL*, Tips

a) **TIMESTAMP_TRUNC**(timestamp_expression, date_part[, time_zone]):
A timestamp function in Standard SQL that truncates a timestamp to the granularity of date_part. Commonly used date_part values include DAY, WEEK, MONTH, QUARTER, and YEAR.

b) **TIMESTAMP_SUB**(timestamp_expression, INTERVAL int64_expression date_part):
A timestamp function that subtracts int64_expression units of date_part from the timestamp, independent of any time zone. Commonly used date_part values include SECOND, MINUTE, and DAY.

c) **CURRENT_TIMESTAMP**():
Returns the current time in timestamp format.

d) Find timestamp functions reference here.

Lecture5, QLabM*
Creating Permanent Tables and Access-Controlled Views in BigQuery

⋆ a) Know six rules for creating tables in BigQuery.

⋆ b) Troubleshoot CREATE TABLE statements.

⋆ c) Understand the definition and usage of views, and create views for different cases.

Lecture5, QLabM*, Create Table Rules

a) Only one CREATE statement is allowed.

b) Either the specified column list or inferred columns from a query statement (or both) must be present.

c) When both the column list and the as query statement clause are present, BigQuery ignores the names in the as query statement clause and matches the columns with the column list by position.

d) When the as query statement clause is present and the column list is absent, BigQuery determines the column names and types from the as query statement clause.

e) Column names must be specified either through the column list or as query statement clause.

f) Duplicate column names are not allowed.

Lecture5, QLabM*, Tips

a) **Before you start writing queries, pin data-to-insights dataset to your project.** You may open this link in incognito mode, and log in with your temporary account.

b) Find documentation of STRING_AGG() function here.

Lecture6, QLabN*
Ingesting New Datasets into BigQuery

⋆ a) Create a dataset and load in a **CSV file** into a new BigQuery table.

⋆ b) Load data from **GCS bucket** to overwrite an existing table.

⋆ c) Save Data to **Google Sheets** and import back to BigQuery.

Lecture6, QLabN*, Tips

a) **SAFE_DIVIDE**(X, Y): Equivalent to the division operator (/), but returns NULL if an error occurs, such as a division by zero error.

b) **It's good practice to always wrap table names in backticks**. In this lab, the table name '**ecommerce.products**' contains no special characters nor reserved keywords, so we can safely leave the table name alone. However, if a name contains characters like '-', then omitting backticks would lead to error. A good example can be the dataset '**data-to-insights**' we used before (also in future Lab O); you may keep an eye on this. Also make sure not to mix up backticks with quotes.

c) If you have problems opening the dropdown menu of 'SAVE RESULTS' button, hide Query editor and then try again.

d) Reference to the last MCQ here.

Lecture7, QLabO
Troubleshooting and Solving Data Join Pitfalls

- ⋆ a) Understand and be able to perform five types of JOIN: CROSS JOIN, INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.
- ⋆ b) Be able to figure out the relationship (one-to-many, many-to-many, etc.) among multiple columns from different datasets.
- ⋆ c) Remember to examine datasets before performing joins, especially to inspect the key to check for potential duplication.

Lecture7, QLabO, Tips

a) In this lab, we review the procedures to create a new dataset and to pin a project in BigQuery. Recall how you did this before and try completing the steps without referring to instructions.

b) The 'ecommerce website catalog' link fails, but it doesn't matter. You can easily understand the one-to-many relationship between a product name and its SKUs (act as options), which could account for the difference in query results with/without the keyword DISTINCT.

c) Make a comparison between two queries in 'Examine relationship between SKU & Name' under 'Identify a key field in your ecommerce dataset' section. These two queries are in exactly the same format, but vary in the columns to aggregate, filter, group by, and order by.

d) Remember to try replacing STRING_AGG() with ARRAY_AGG()! Find out what your results look like.

Lecture7, QLabO, Tips

e) Recall the keyword LIKE you learned from DataCamp course Intro to SQL for Data Science.

f) Join pitfalls can happen in any type of JOIN. Recall the INNER JOIN example we saw in DataCamp, where we tried joining 'countries' table with 'populations' table on country code and ended up with multiple records of each country. The solution is to specify the 'year' variable as an additional key. Quick access to the DataCamp exercise here.

g) JOIN functions reference here (provided in the lab). You may simply focus on examples in '**Join Types**' section. If you are interested in more advanced JOIN functions, you may take a look at CROSS JOIN and fuzzy joins. More information on CROSS JOIN here.

h) WITH clause reference here.

Lecture7, QLabP*
Creating a Data Warehouse Through Joins and Unions

a) Gain a sense of Google's sentiment analysis API.

⋆ b) Be able to enrich the using JOINs and UNIONs.

⋆ c) Know how to write queries to CREATE a new table or INSERT data to an existing one.

Lecture7, QLabP*, Tips

a) The sentiment analysis is a method of Natural Language API. Find more information about Natural Language API here. Natural Language Processing (NLP) is a brunch of AI that helps computers understand, interpret and manipulate human language, and we can easily use the pre-trained models in Google to conduct text analysis.

b) LEFT JOIN allows us to keep the initial 'productSKU' records when joining sales data and inventory data.

c) A function starting with the **SAFE** will return **NULL** when an error occurs. Find more about **SATE.prefix** here.

d) A wildcard table represents a union of all the tables that match the wildcard expression, which is similar to a subquery using multiple UNION ALLs in the FROM clause.

Lecture7, QLabQ*
Creating Date-Partitioned Tables in BigQuery

⋆ a) Understand the definition of a partitioned table, and the benefits of querying a partitioned table compared to a non-partitioned one.

⋆ b) Create a date-partitioned table using PARTITION BY.

⋆ c) Create an auto-expiring partitioned table by specifying expiration days in OPTIONS clause.

d) Confirm the partition age by DATE_DIFF function.

Lecture7, QLabQ*, Tips

a) Keep in mind that LIMIT does not reduce the total amount of data processed.

b) For the query in section 'Creating an auto-expiring partitioned table', be aware of the filtering condition in the subquery:

WHERE stations.usaf = stn

It's a correlated subquery, which uses values from the outer query to generate the final result. In this case, for every row generated in the final dataset, the subquery is re-executed to match 'usaf' from table 'noaa_gsod.stations' with 'stn' from tables 'noaa_gsod.gsod*'.

Feel free to refer to the DataCamp course Intermediate SQL - You just learned correlated subquery from chapter 3.

Lecture7, QLabQ*, Tips

c) **DATE**(year, month, day): Constructs a DATE from INT64 values representing the year, month, and day.

d) **ANY_VALUE**(): Returns any value from the input or NULL if there are zero input rows. Find documentation and more examples here.

e) **_TABLE_SUFFIX**: Used for querying multiple tables using a wildcard expression. You can either filter specific tables or scan a range of tables. In this lab, we cast the table suffix to an integer and compare it with 2018.

   More information and examples here.

f) Documentation of creating and using partitioned tables here.

Lecture8, QLabR
Predict Visitor Purchases with a Classification Model in BQML

- ⋆ a) Know two major types of machine learning models and example cases they are used in.
- ⋆ b) Understand the difference and relationship between a training and an evaluation dataset.
- c) Query the dataset to prepare feature columns and the label column for model training.
- d) Create a classification model in BQML.
- ⋆ e) Evaluate model performance with ROC curve.
- f) Make predictions and rank the probability of your prediction.

Lecture8, QLabR, Tips

a) You may find detailed column descriptions of table 'ecommerce.web_analytics' here. This would help you better understand the query in section Improve model performance with Feature Engineering.

b) **COUNTIF**(expr): Returns the count of TRUE values for expression. Returns 0 if there are zero input rows or expression evaluates to FALSE for all rows.

c) **IFNULL**(expr, null_result): If expr is NULL, return null_result. Otherwise, return expr.

d) Recall Arrays and UNNEST() function we learned from the last lecture, and the CASE clause from DataCamp.

e) A deeper look at two CREATE OR REPLACE MODEL statements may help you better locate features and understand how these features are created using multiple aggregation functions and subqueries.

Lecture8, QLabS*
Predict Taxi Fare with a BigQuery ML Forecasting Model

⋆ a) Explore the dataset to select useful features for your machine learning model.

b) Define variables for the training set and an independent evaluation set for model training.

c) Create a linear regression model in BQML.

⋆ d) Use RMSE loss metric to evaluate your trained ML model.

e) Use the evaluated model to make predictions.

f) Perform feature engineering to improve the performance of your ML model.

Lecture8, QLabS*, Tips

a) **TIMESTAMP_DIFF**(X, Y, date): Returns the number of whole specified date intervals between two timestamps. Find more here.

b) **FARM_FINGERPRINT**(): Returns the fingerprint of the value. Find more here.

c) **MOD**(): Returns the remainder of dividing the first argument by the second argument. In this QLab, We use the modulo value as a filter to generate training and valuation sets.

d) By replacing **params.TRAIN** with **params.EVAL** in the query within the ML model, we are switching to the evaluation set, which is an independent dataset used for an out-of-sample model evaluation.

e) Under the workspace of Coursera, you can find many machine learning related courses. Here is one talking about the BGML.

Lecture9, QLabT
Extract, Analyze, and Translate Text from Images with the Cloud ML APIs

⋆ a) Create a cloud storage bucket and upload your own files to it.

b) Create a Vision API key and a request.

⋆ c) Get a sense of how to call the Vision API with 'curl'.

d) Use the Vision API's text detection (OCR) method.

e) Use the Translation API to translate text from the image.

f) Use the Natural Language API to analyze the text.

Lecture9, QLabU*
Working with JSON, Arrays, and Structs in BigQuery

a) Understand the definition of database normalization and de-normalization.

⋆ b) Understand Array and Struct data types in BigQuery; be able to recognize them through Type and Mode in Schema panel.

c) Get a sense of the benefits of using Arrays and Structs in a large reporting table.

d) Ingest a json file into a Bigquery table; be able to customize table schema and check it in JSON panel.

⋆ e) Practice working with Arrays using ARRAY_AGG() and ARRAY_LENGTH().

⋆ f) Query datasets containing Array variables using UNNEST().

⋆ g) Query datasets containing Struct variables; be able to explain the query statement following the idea of cross join.

⋆ h) Query datasets containing Arrays and Structs by unpacking both; use aliases appropriately to simplify your statement.