Tutorial 1. Updated.

The purpose of this tutorial is to give you experience of getting data via BigQuery. We will export our data and push it into git for use with Google Colab, however, we can use BigQuery directly. We will not go into using BigQuery directly with Google Colab.

1.0

Let’s go to BigQuery using you UHI google account: <https://console.cloud.google.com/home>

Now, scroll down on the left hand menu and click on BigQuery, then BigQuery Studio:

A screenshot of a cloud overview

Description automatically generated

You should now see something like the following if you have done Tutorial 0:

A screenshot of a computer

Description automatically generated

In the explorer pane, type in Big, into the Search Bar:

A screenshot of a computer

Description automatically generated

Then click on SEARCH ALL PROJECTS:

A screenshot of a search box

Description automatically generated

Then click on the star next to bigquery-public-data.

A close up of a text

Description automatically generated

You can now clear the search box using the X:

A screenshot of a search engine

Description automatically generated

You will see your project and the bigquery-public-data dataset.

1.1

The good thing is, this will allow you to have all of the data you need for your assignment as well.

Let’s discuss why we are doing this. We want to find out if weather has an effect on the number of taxi trips taken in New York City. To carry out this analysis, we need all of the taxi trips taken in New York and the weather data per day over the same period of time. Thankfully we have access to both.

**For the purposes of the tutorial, we are only going to use 2009 and make an assumption about where the weather is measured from i.e. Central Park. In the assignment, it is ok to use Central Park for the weather data, but the date range will be much larger.**

Now, why don’t we explore the data we need.

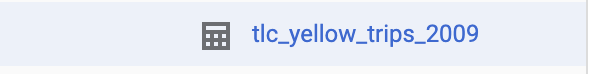
The Taxi data is actually stored here:

A screenshot of a phone

Description automatically generated

bigquery-public-data > new\_york > tlc\_yellow\_trips\_2009

If you select it



You will get the following:

A screenshot of a computer

Description automatically generated

Here we can see the schema. Next, we are going to query this dataset:

Click on Query:

A screenshot of a computer

Description automatically generated

Then, “In a new tab”:

A screenshot of a computer

Description automatically generated

And you get this:

A screenshot of a computer

Description automatically generated

As you can see, we have an error, perhaps let’s make a change. Copy and paste this in to the query window:

SELECT \* FROM `bigquery-public-data.new\_york.tlc\_yellow\_trips\_2009` LIMIT 10

This means, select everything from the tlc\_yellow\_trips\_2009 but only show me 10. Notice the backticks around the table used and the use of dots between the different parts of the structure i.e.

`bigquery-public-data.new\_york.tlc\_yellow\_trips\_2009`

Now, let’s “Run” this query by clicking the Run button:

A picture containing ball, person, player, drawing

Description automatically generated

And we get:

A screenshot of a computer

Description automatically generated

You can’t see all of the columns, but you can move around in here to see some more of the data.

Now, let’s have a look at the weather data. Luckily for us, it’s in here too.

bigquery-public-data > noaa\_gsod > gsod2009

A screenshot of a computer

Description automatically generated

Scroll down until we get to 2009.

There are many years of weather data in here:

A close-up of a message

Description automatically generated

Even as far back as the 1929. Again, selecting this you will get some information about the dataset.

A screenshot of a computer

Description automatically generated

In exactly the same way as we did just a moment ago, click on Query and open a new Query window and copy and paste this in:

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2009`

Now, let’s have a look at the data:

A screenshot of a data

Description automatically generated

This query will take a while and will return weather for the whole of America.

We want to narrow it down to Central Park in New York. Thankfully, I did this for you using this query:

SELECT \* FROM `bigquery-public-data.noaa\_gsod.stations` where country = "US" and state = "NY" and name = 'NYC CENTRAL PARK'

Which gives a wban of 94728, let’s test it with this query. Again, copy, paste and run:

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2009` where wban = '94728'

As you will see, you get 365 days returned, so the weather for each day of that year:

A screenshot of a phone

Description automatically generated

1.2 Collate the data

What we need to do, for our little example, is collate the weather data for each day and find the number of taxi trips for each day. We also want to add a day of the week numerical field (1 Monday – 7 Sunday) as it is likely that the day of the week will have an influence on taxi usage.

On the right of your project in the explorer pane, click on the 3 dots and then Create Dataset:

A screenshot of a computer

Description automatically generated

In the side drawer, add a Dataset ID (you will need to make another for the assignment):

A screenshot of a computer

Description automatically generated

Next, click “Create Dataset”

Now, under your project, you will see:

A close-up of a white background

Description automatically generated

We will now create a view called taxi\_data\_count by copy and pasting this query into a Query Window:

CREATE VIEW `hip-plexus-741.tutorial\_1.taxi\_data\_count`

AS SELECT CAST(pickup\_datetime as DATE) as pickup\_date, COUNT(CAST(pickup\_datetime as DATE)) AS NUM\_TRIPS

FROM `bigquery-public-data.new\_york.tlc\_yellow\_trips\_2009`

GROUP BY pickup\_date

What this will do is create a view with the name “taxi\_data\_count” that will change the date in the “tlc\_yellow\_trips\_2009” to a DATE format rather than a timestamp. This will help us later. It will also count the number of trips for a particular day i.e. there will be lots of trips in one day, this will group them together for each date, then return the length of each of these groups.

After running this query, you will see the following:

A screenshot of a computer

Description automatically generated

Now, let’s check out our data:

We can do this by copy and pasting the query into a query pane as we have done before:

SELECT \* FROM `hip-plexus-741.tutorial\_1.taxi\_data\_count`

Run this query and you can see:

A screenshot of a data

Description automatically generated

The dates aren’t ordered, but you can note there is 365 rows i.e. 1 for every day in the year and that NUM\_TRIPS shows the number of taxi journeys on that particular day.

A screenshot of a computer

Description automatically generated

Next, we are going to add a new day of the week integer field. To do this we will use the view we just made, to make another view. Again, in the query pane:

CREATE VIEW `hip-plexus-741.tutorial\_1.taxi\_data\_count\_final`

AS SELECT FORMAT\_DATE("%u", pickup\_date) as day, pickup\_date, NUM\_TRIPS

FROM`hip-plexus-741.tutorial\_1.taxi\_data\_count`

Which creates another View:

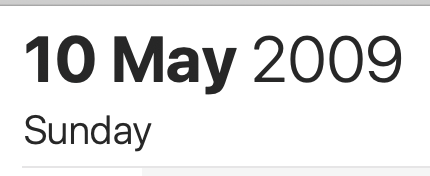
A screenshot of a computer

Description automatically generated

Now, let’s see the data:

SELECT \* FROM `hip-plexus-741.tutorial\_1.taxi\_data\_count\_final`

As you can see, the 10th of May 2009 was a Sunday (7):



And the 7th of September is a Monday (1):

A close up of a logo

Description automatically generated

Next, we need to deal with our weather data. Firstly, create another view:

CREATE VIEW `hip-plexus-741.tutorial\_1.weather\_data`

AS SELECT DATE(CAST(year AS INT64), CAST(mo AS INT64), CAST(da AS INT64)) AS date, year, mo, da, temp, dewp, slp, visib, wdsp, mxpsd, gust, max, min, prcp, sndp, fog

FROM `bigquery-public-data.noaa\_gsod.gsod2009`

WHERE wban='94728'

ORDER BY mo, da;

Let’s have a look at it:

A screenshot of a computer code

Description automatically generated

SELECT \* FROM `hip-plexus-741.tutorial\_1.weather\_data`

We get:

A screenshot of a computer

Description automatically generated

As you can see, I have created a date field that is of the type DATE. This will allow us to join with the taxi data.

Lastly, let’s create a table with our final data. We do this using a query like before, in the Query Pane:

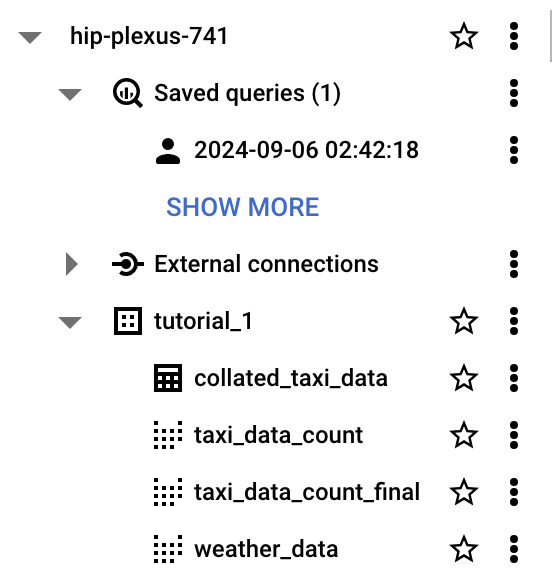
CREATE TABLE `hip-plexus-741.tutorial\_1.collated\_taxi\_data`

AS SELECT day, year, mo, da, pickup\_date, temp, dewp, slp, visib, wdsp, mxpsd, gust, max, min, prcp, sndp, fog, NUM\_TRIPS

FROM `hip-plexus-741.tutorial\_1.weather\_data` AS weather, `hip-plexus-741.tutorial\_1.taxi\_data\_count\_final` AS complaints

WHERE complaints.pickup\_date = weather.date;

And check the data:



SELECT \* FROM `hip-plexus-741.tutorial\_1.collated\_taxi\_data` order by mo, da

And we get:

A screenshot of a computer

Description automatically generated

1.3 Download the data as a CSV

With the returned data above, click on:

A close up of a logo

Description automatically generated

You can see something like this:  
  
A screenshot of a computer

Description automatically generated

Then, click on CSV local file:

A screenshot of a cell phone

Description automatically generated

The file should be downloaded and you can check it in a text editor:

A picture containing computer

Description automatically generated

At this point, you should upload this file to your git repository that you are using for this course. If you haven’t done it already from tutorial 0:

Go to the URL: <https://github.com/>

On the left-hand column menu, click New:

A close up of a logo

Description automatically generated

Add a repository name:

A screenshot of a cell phone

Description automatically generated

As you can see, I have used my made-up student number, but you should use yours.

Add a description:

A screenshot of a cell phone

Description automatically generated

Notice that I have selected “Initialize this repository with a README”.

Click “Create repository”.

You should see something like this:

A screenshot of a social media post

Description automatically generated

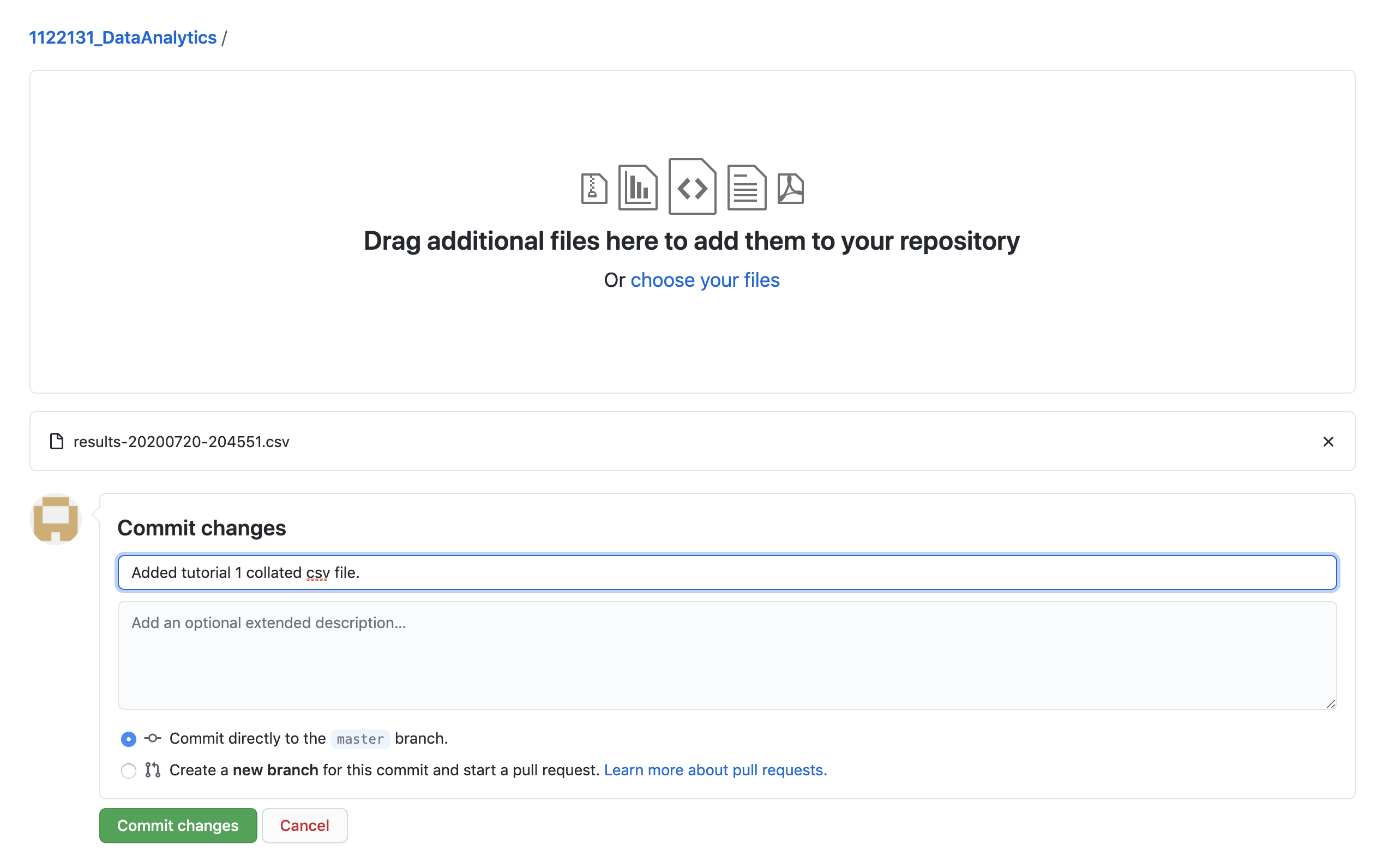
You can add the csv file in a number of different ways but we will just use the web interface. Firstly, save your finished csv file as <whateveryoulike>.csv to somewhere familiar i.e. your Desktop.

On github, on the repo page shown above, click “Add file”:

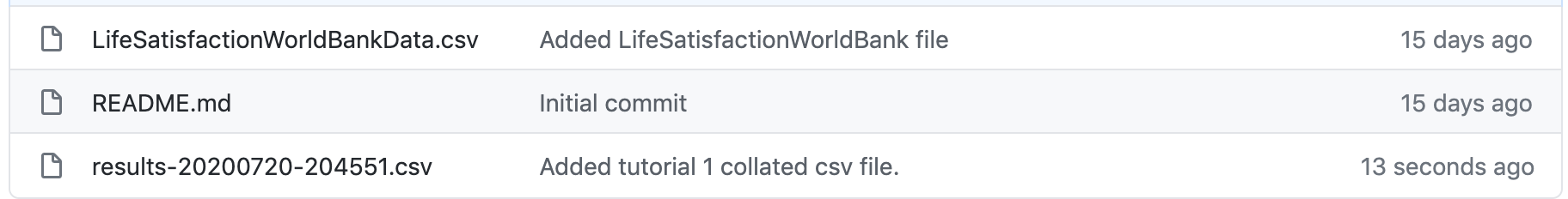
A screenshot of a cell phone

Description automatically generated

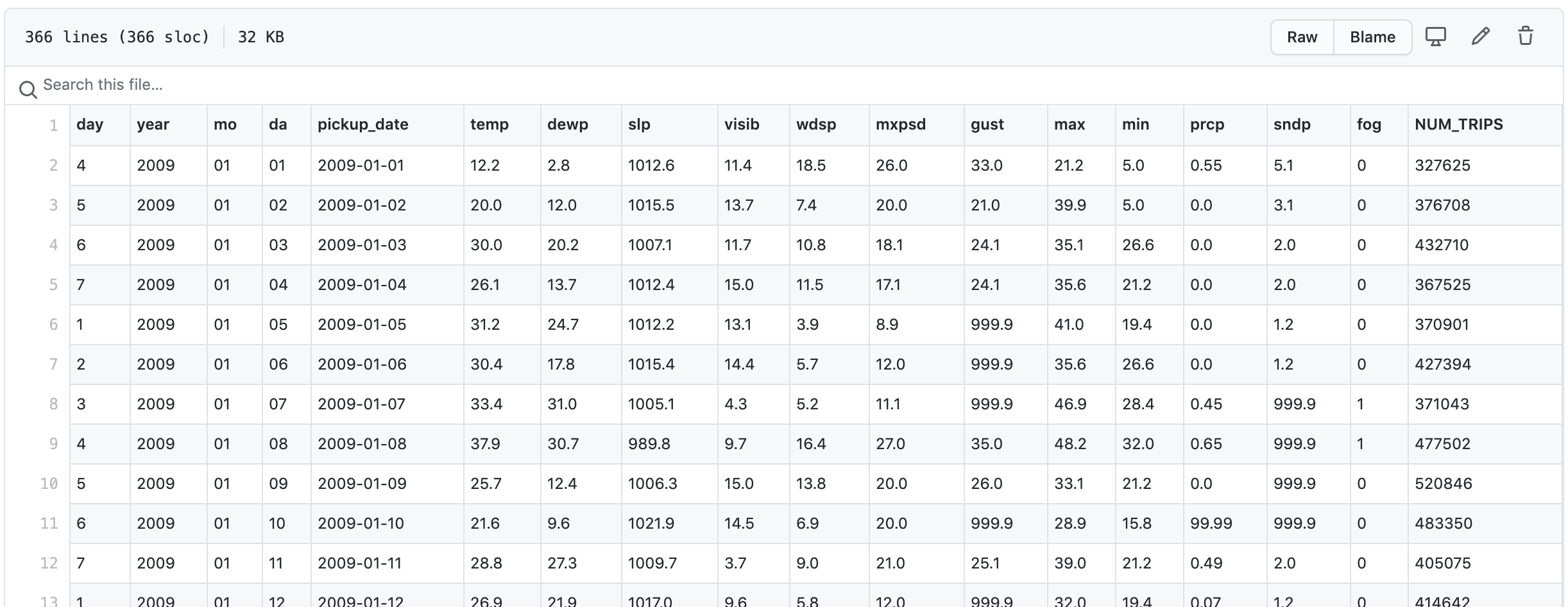
Select the file and write a commit message:



Click “Commit changes”:



You should now see your file. Click on it:



Click on Raw:

A screenshot of a computer

Description automatically generated

Now, take note of the url, copy it somewhere, we will need it.

(Here is mine just in case you are having problems: <https://raw.githubusercontent.com/1122131uhi/1122131_DataAnalytics/master/results-20200720-204551.csv>)

1.4 Known issues and things for the assignment:

In the assignment, you most likely want to download the full dataset that you are looking at (not the weather). This way you can do data science on it to see if there are mitigating factors or data that needs cleansed as shown in the main course document. In this case we jumped straight to collating data. The collated data will also require to be analysed and cleansed.

Also, the assignment data has a data range over a number of years. You will need to join the data for each year in the weather data, together.

Let’s explore this:

To get all of the data for 2009 you will need to use the query:

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2009` WHERE wban='94728'

As you can see, this returns 365 days. If we want 2010, should be simple, we just put in 2010:

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2010` WHERE wban='94728'

Next, for 2011 onwards:

Returns 365 days.

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2011` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2012` WHERE wban='94728'

Returns 366 days (leap year)

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2013` WHERE wban='94728'

Return 365

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2014` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2015` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2016` WHERE wban='94728'

Returns 366 days (leap year)

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2017` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2018` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2019` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2020` WHERE wban='94728'

Returns 366 days (leap year)

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2021` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2022` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2023` WHERE wban='94728'

Returns 365 days

SELECT \* FROM `bigquery-public-data.noaa\_gsod.gsod2024` WHERE wban='94728'

Returns only part of the year as we are not complete with 2024 yet.

Now, I am going to show you how to merge three of these weather tables together into one table. Again, we use a query for this in the Query Pane:

CREATE VIEW `hip-plexus-741.tutorial\_1.weather\_2009\_to\_2011` AS

SELECT DATE(CAST(year AS INT64), CAST(mo AS INT64), CAST(da AS INT64)) AS date, year, mo, da, temp, dewp, slp, visib, wdsp, mxpsd, gust, max, min, prcp, sndp, fog

FROM `bigquery-public-data.noaa\_gsod.gsod2009` WHERE wban='94728'

UNION ALL

SELECT DATE(CAST(year AS INT64), CAST(mo AS INT64), CAST(da AS INT64)) AS date, year, mo, da, temp, dewp, slp, visib, wdsp, mxpsd, gust, max, min, prcp, sndp, fog

FROM `bigquery-public-data.noaa\_gsod.gsod2010` WHERE wban='94728'

UNION ALL

SELECT DATE(CAST(year AS INT64), CAST(mo AS INT64), CAST(da AS INT64)) AS date, year, mo, da, temp, dewp, slp, visib, wdsp, mxpsd, gust, max, min, prcp, sndp, fog

FROM `bigquery-public-data.noaa\_gsod.gsod2011` WHERE wban='94728'

ORDER BY year, mo, da;

Running this we get:

A screenshot of a phone

Description automatically generated

Let’s look at the results:

SELECT \* FROM `hip-plexus-741.tutorial\_1.weather\_2009\_to\_2011`

Run this and we get:

A screenshot of a computer

Description automatically generated

As you can see, we get 1095 results for 3 years (no leap years) between 2009 – 2011.

This method will be useful for your assignment.