

## ▼ Before you begin

1. Use the [Cloud Resource Manager](#) to Create a Cloud Platform project if you do not already have one.
2. [Enable billing](#) for the project.
3. [Enable BigQuery](#) APIs for the project.

## ▼ Provide your credentials to the runtime

```
from google.colab import auth
auth.authenticate_user()
print('Authenticated')
```

## ▼ Optional: Enable data table display

Colab includes the `google.colab.data_table` package that can be used to display large pandas dataframes as an interactive data table. It can be enabled with:

```
%load_ext google.colab.data_table
```

If you would prefer to return to the classic Pandas dataframe display, you can disable this by running:

```
%unload_ext google.colab.data_table
```

## ▼ Use BigQuery via magics

The `google.cloud.bigquery` library also includes a magic command which runs a query and either displays the result or saves it to a variable as a `DataFrame`.

```
# Display query output immediately

%%bigquery --project yourprojectid
SELECT
  COUNT(*) as total_rows
FROM `bigquery-public-data.samples.gsod`
```

1 entry Filter

index	total_rows
0	114420316

Show 25 per page

# Save output in a variable `df`

```
%bigquery --project yourprojectid df
SELECT
  COUNT(*) as total_rows
FROM `bigquery-public-data.samples.gsod`
```

df

1 entry Filter ?

index	total_rows
0	114420316

Show 25 per page

## ▼ Use BigQuery through google-cloud-bigquery

See [BigQuery documentation](#) and [library reference documentation](#).

The [GSOD sample table](#) contains weather information collected by NOAA, such as precipitation amounts and wind speeds from late 1929 to early 2010.

## ▼ Declare the Cloud project ID which will be used throughout this notebook

```
project_id = '[your project ID]'
```

## ▼ Sample approximately 2000 random rows

```
from google.cloud import bigquery

client = bigquery.Client(project=project_id)

sample_count = 2000
row_count = client.query('''
  SELECT
    COUNT(*) as total
  FROM `bigquery-public-data.samples.gsod`''').to_dataframe().total[0]
```

```
df = client.query('''
SELECT
*
FROM
`bigquery-public-data.samples.gsod`
WHERE RAND() < %d/%d
''' % (sample_count, row_count)).to_dataframe()
```

```
print('Full dataset has %d rows' % row_count)
```

Full dataset has 114420316 rows

## ▼ Describe the sampled data

```
df.describe()
```

1 to 8 of 8 entries Filter ?

index	station_number	wban_number	year	month	day
count	1979.0	1979.0	1979.0	1979.0	
mean	505585.599292572	89647.44113188479	1987.1814047498738	6.525517938352704	15.7155128
std	302491.1873178059	27088.238467141528	15.993487610677022	3.419259475533833	8.6615625
min	10100.0	13.0	1933.0	1.0	
25%	238255.0	99999.0	1978.0	4.0	
50%	538980.0	99999.0	1990.0	7.0	
75%	725273.5	99999.0	2000.0	10.0	
max	999999.0	99999.0	2010.0	12.0	

Show 25 per page

## ▼ View the first 10 rows

```
df.head(10)
```

1 to 10 of 10 entries

Filter



index	station_number	wban_number	year	month	day	mean_temp	num_mean_temp_sam
0	105780	99999	1968	9	13	46	

# 10 highest total\_precipitation samples

df.sort\_values('total\_precipitation', ascending=False).head(10)[['station\_number', 'year', 'm

1 to 10 of 10 entries

Filter



index	station_number	year	month	day	total_precipitation
644	230220	1964	7	15	5.909999847412109
1155	985430	2008	12	8	3.4600000381469727
1196	248260	1961	11	1	2.950000047683716
1588	257670	1959	8	9	2.950000047683716
980	299150	1962	3	1	2.950000047683716
1325	470250	1965	11	25	2.950000047683716
1917	288380	1994	8	6	2.319999933242798
1211	585190	1995	4	14	2.319999933242798
250	647000	2005	8	19	2.200000047683716
1418	964710	1975	9	8	1.9700000286102295

Show 25 per page

## ▼ Use BigQuery through pandas-gbq

The `pandas-gbq` library is a community led project by the pandas community. It covers basic functionality, such as writing a DataFrame to BigQuery and running a query, but as a third-party library it may not handle all BigQuery features or use cases.

[Pandas GBQ Documentation](#)

```
import pandas as pd
```

```
sample_count = 2000
```

```
df = pd.io.gbq.read_gbq('''
    SELECT name, SUM(number) as count
    FROM `bigquery-public-data.usa_names.usa_1910_2013`
    WHERE state = 'TX'
    GROUP BY name
    ORDER BY count DESC
    LIMIT 100
''', project_id=project_id, dialect='standard')
```

```
df.head()
```

1 to 5 of 5 entries

Filter



index	name	count
0	James	272793
1	John	235139
2	Michael	225320
3	Robert	220399

## ▼ Syntax highlighting

`google.colab.syntax` can be used to add syntax highlighting to any Python string literals which are used in a query later.

```
from google.colab import syntax
query = syntax.sql('''
SELECT
  COUNT(*) as total_rows
FROM
  `bigquery-public-data.samples.gsod`
''')
```

```
pd.io.gbq.read_gbq(query, project_id=project_id, dialect='standard')
```

1 entry

Filter



index	total_rows
0	114420316

Show  per page

