BigQuery GIS:
A GeoVisual Exploration

Google Cloud

This session will be 🎥recorded🎥

You mute yourself when not speaking. Turn on/off camera if you want

Google Cloud

BigQuery GIS:
A GeoVisual Exploration

Google Cloud

Agenda

- BigQuery GIS Intro
- GIS Data Types
- Demo

Google Cloud

Speaker

Razvan CULEA
Strategic Cloud Engineer
Data & Analytics

Google Cloud

What is BigQuery?

- Fully Managed, Zero-Ops Data Warehouse
- Petabyte-Scale
- Industry-Standard SQL
- Automatically Encrypted, Durable, and Highly Available
- Virtually Unlimited Resources

Google Cloud

What is BigQuery **GIS**?

- Geospatial Information Systems
- BigQuery is the first MPP data warehouse to support geospatial data types and functions
- GEOGRAPHY data type represents a pointset on Earth
- Same S2 library as Google Maps
- BQ Geo Viz visualization tool

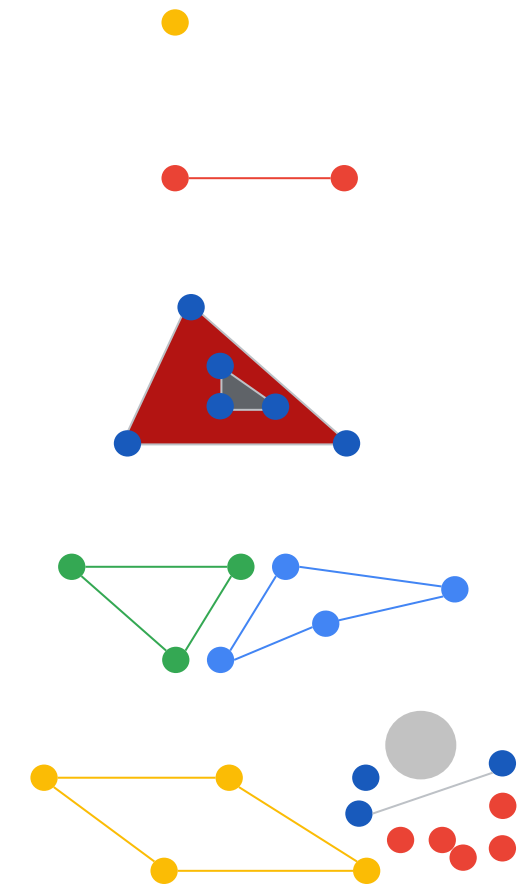
Google Cloud

GIS **Data Types**

The `GEOGRAPHY` data type represents a pointset on the Earth's surface.

A pointset is a set of points, lines and polygons.

- Point
- Linestring
- Polygon
- Multi-polygon
- Collections



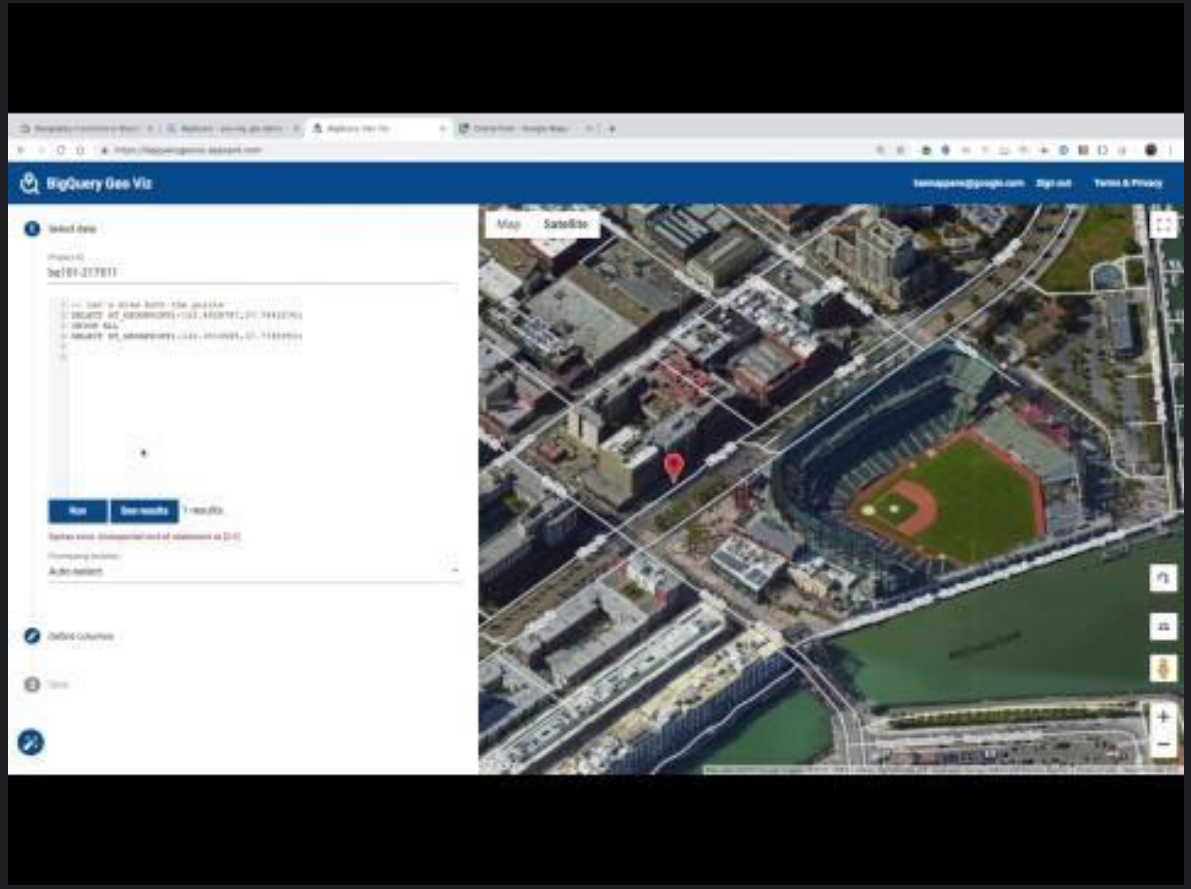
Google Cloud

GIS Functions Types and Functions

Constructors	ST_GEOGPOINT(longitude, latitude)
Transformations	ST_CLOSESTPOINT(geography_1,geography_2)
Predicates	ST_CONTAINS(geography_1,geography_2)
Measures	ST_DISTANCE(geography_1,geography_2)
Parsers	ST_GEOGFROMTEXT(wkt_string)

DEMO!!

Google Cloud



Google Cloud

BigQuery Geo Viz tool

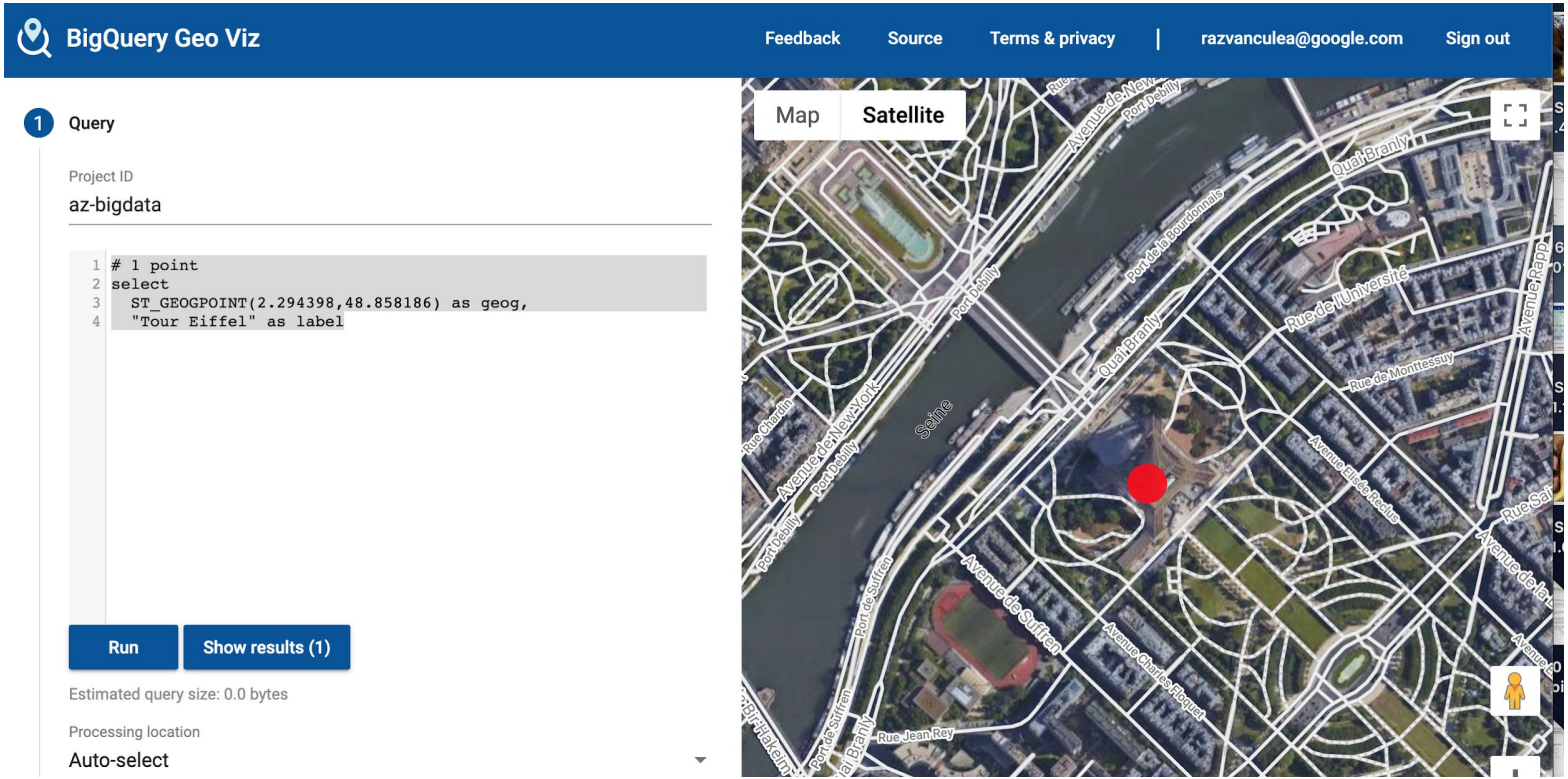
Go to : https://bigquerygeoviz.appspot.com/

Type a project id where you have BigQuery access to run queries.

Get latitude and longitude coordinates (from maps.google.com)

```
# 1 point
select
  ST_GEOGPOINT(2.294398,48.858186) as geog,
  "Tour Eiffel" as label
union
select
  ST_GEOGPOINT(2.2948335,48.8737917) as geog,
  "Arc de Triomphe" as label
```

https://bigquerygeoviz.appspot.com/



BigQuery Geo Viz tool

Go to : https://bigquerygeoviz.appspot.com/

Type a project id where you have BigQuery access to run queries.

Get latitude and longitude coordinates (from maps.google.com)

```
# 2 points
SELECT
  ST_GEOGPOINT(2.294398,48.858186) AS geog,
  "Tour Eiffel" AS label
UNION ALL
SELECT
  ST_GEOGPOINT(2.2948335,48.8737917) AS geog,
  "Arc de Triomphe" AS label
```

Google Cloud

BigQuery Geo Viz tool

```
# 2 points and 1 line
SELECT
  ST_GEOGPOINT(2.294398,48.858186) AS geog,
  "Tour Eiffel" AS label
UNION ALL
SELECT
  ST_GEOGPOINT(2.2948335,48.8737917) AS geog,
  "Arc de Triomphe" AS label
UNION ALL
SELECT
  ST_MAKELINE(ST_GEOGPOINT(2.294398,48.858186),
ST_GEOGPOINT(2.2948335,48.8737917)) AS geog, "Arc-Tour" AS label
```

BigQuery Geo Viz tool

```
SELECT ST_GEOGPOINT(2.294398,48.858186) AS geog, "Tour Eiffel" AS label
UNION ALL
SELECT ST_GEOGPOINT(2.2948335,48.8737917) AS geog, "Arc de Triomphe" AS label
UNION ALL
SELECT ST_GEOGPOINT(2.3122604,48.8661091) AS geog, "Grand Palais" AS label
UNION ALL
SELECT
  ST_MAKEPOLYGON(ST_GEOGFROMTEXT('LINESTRING(2.294398 48.858186, 2.2948335 48.8737917, 2.3122604
48.8661091)')) AS geog, "Arc-Tour-Palais" AS label
UNION ALL
SELECT
ST_CENTROID(ST_MAKEPOLYGON(ST_GEOGFROMTEXT('LINESTRING(2.294398 48.858186, 2.2948335 48.8737917,
2.3122604 48.8661091)'))) AS geog,
  "Centr Arc-Tour-Palais" AS label
```

Custom maps & BigQuery import

Sign in https://mymaps.google.com/
Create a layer with multiple polygons. Ex: parks & landmarks in Paris

Export to KMZ

Use GDAL to convert KMZ to CSV for BigQuery
ogr2ogr -f csv **parc.csv** -nln output -nlt GEOMETRY -dialect sqlite -sql "select AsGeoJSON(geometry) geom,* from **parc**" Paris_Geo.kmz

ogr2ogr -f csv **poi.csv** -nln output -nlt GEOMETRY -dialect sqlite -sql "select AsGeoJSON(geometry) geom,* from **poi**" Paris_Geo.kmz

BigQuery - import CSV & test

Go to https://cloud.google.com/bigquery
Select your BQ project and import the CSVs in a dataset

```
SELECT ST_GEOGFROMGEOJSON(replace(geom,",0]","]")) as g, Name
FROM `az-bigdata.geo_demo.poi`
```

Explore the data in Geo Viz

Google Cloud

BigQuery - import CSV & test

Go to https://cloud.google.com/bigquery
Select your BQ project and import the CSVs in a dataset

Test:

```
SELECT poi.Name poi_name, parc.Name parc_name, poi.g poi_g, parc.g parc_g
FROM (
  #1
  SELECT ST_GEOGFROMGEOJSON(replace(geom,",0]","]")) as g, Name
  FROM `az-bigdata.geo_demo.poi`
) poi
JOIN (
  #2
  SELECT ST_GEOGFROMGEOJSON(replace(geom,",0]","]")) as g, Name
  FROM `az-bigdata.geo_demo.parc`
) parc
ON ST_CONTAINS(parc.g,poi.g);
```

BigQuery - Public Datasets - Covid-19

Public dataset :
https://console.cloud.google.com/marketplace/details/bigquery-public-datasets/covid19-public-data-program

Go to https://cloud.google.com/bigquery

SELECT * FROM `bigquery-public-data.covid19_jhu_csse_eu.summary`

London Bikes

Dataset :
https://console.cloud.google.com/marketplace/details/greater-london-authority/london-bicycles

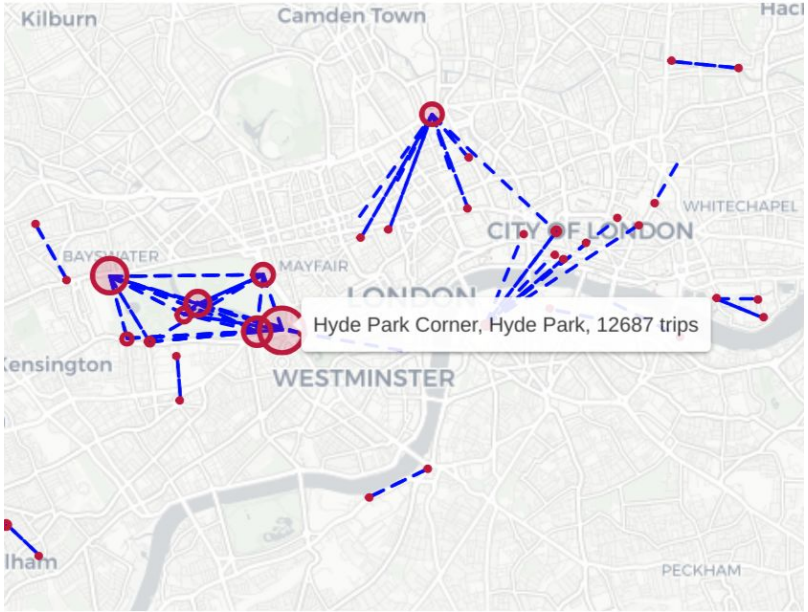
```
SELECT ST_GEOGPOINT(longitude,latitude)
geog,id,name,docks_count,bikes_count
FROM `bigquery-public-data.london_bicycles.cycle_stations`
```

Most Common Bicycle Paths

Use ST_MAKELINE() to draw a line between the start and stop destination in the most common bicycle paths.

Most cycle trips start and end around Hyde Park!

```
  stations AS (
  SELECT
    name,
    ST_GEOGPOINT(longitude,latitude) AS geometry
  FROM
    `google.com:pso-bq-gis-demo.gis_demo.cycle_stations` )
SELECT
  ST_ASGEOJSON(ST_MAKELINE(b.geometry,c.geometry)) line,
  ST_ASGEOJSON(c.geometry) AS end_geog,
  a.*
FROM
  most_popular_paths a
JOIN
  stations b
```



Google Cloud

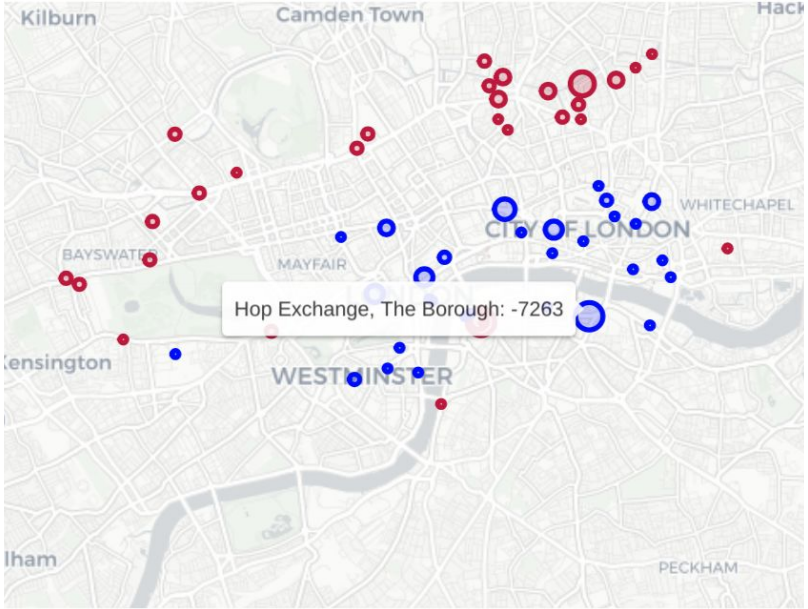
Docking Capacity Analysis

To analyse station docking capacity, we subtract the number of trips that started from the station by the number of trips that ended at the station.

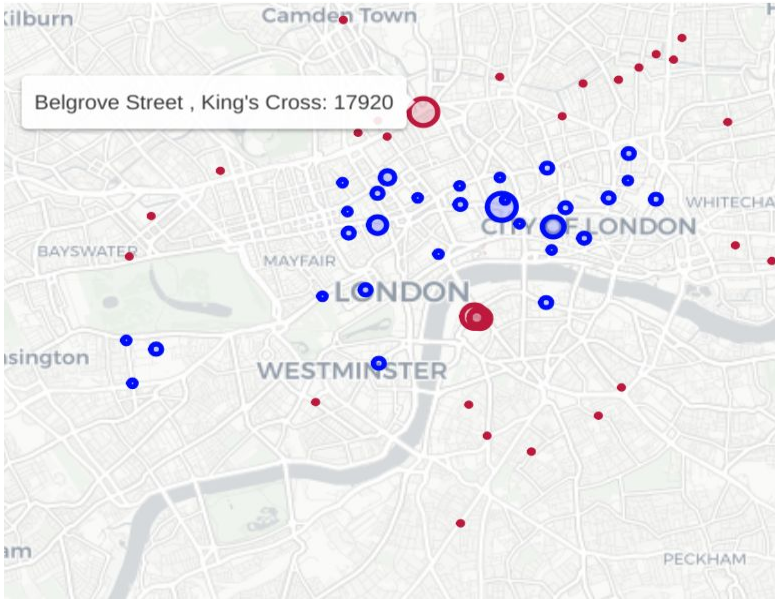
Looks like there are more trips ending at Borough Market in the weekend!

Google Cloud

```
    end_station_name,
    COUNT(1) as end_count
  FROM
    `google.com:pso-bq-gis-demo.gis_demo.london_cycle`
  WHERE
    EXTRACT(DAYOFWEEK FROM start_date) IN (1,7)
  GROUP BY 1
 )
SELECT
  a.start_station_name AS station_name,
  ST_ASGEOJSON(ST_GEOGPOINT(c.longitude, c.latitude)) geog,
  a.start_count - b.end_count AS difference
FROM
  start_count a
```



Morning vs Evening Commute Traffic



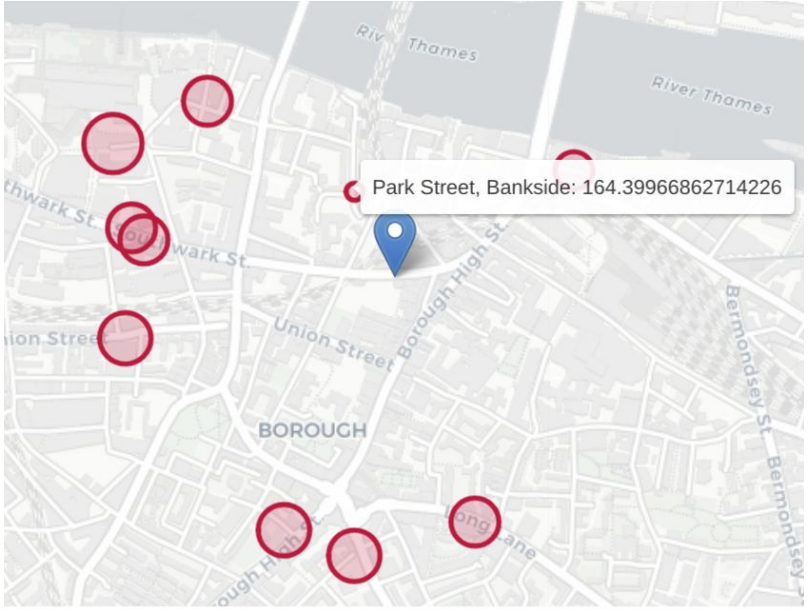
Google Cloud

Closest Docking Stations to Borough Market

I also like to go to Borough Market in the weekend, however there is unlikely to be docking space available.

What are some closest stations I can head to instead?

```
      name = 'Hop Exchange, The Borough'
  )
SELECT
  a.name source_station,
  ST_ASGEOJSON(a.geog) source_geog,
  b.name destination_station,
  ST_ASGEOJSON(b.geog) destination_geog,
  ST_DISTANCE(a.geog, b.geog) distance
FROM
  stations a
```



Google Cloud

Appendix

Give it a go!

GIS Hurricane Tutorial

GIS NYC Bikes

GIS London Bikes

References:

BigQuery GIS Documentation

Using GIS with Geo Viz and Google Earth Engine

Geography Functions

BigQuery Public Datasets

Thank you!