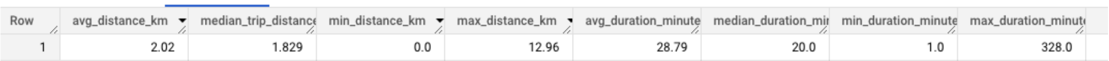
Zen City's Journey through London's bike rental data – BigQuery SQL script:

```
□#Data Cleaning & Data Wrangling:

SELECT
*
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
LIMIT 1000;

SELECT
COUNT(*)
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`; --49015

SELECT
COUNT(DISTINCT rental_id)
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`; --49015

SELECT
*
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
LIMIT 1000;

SELECT
COUNT(*)
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`; --795


SELECT
COUNT(DISTINCT id)
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`; --795
```

```sql
#Check if we have duplicate station id's:
SELECT
id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
GROUP BY id
HAVING COUNT(id) > 1; #no

#num of bikes:
SELECT COUNT(DISTINCT bike_id)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`; #11185

#Checking if the values in column duration are correct:
SELECT rental_id
FROM
(
SELECT
rental_id,
duration,
TIMESTAMP_DIFF(end_date, start_date, SECOND) AS calculated_difference
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`)
WHERE duration != calculated_difference; --there are no issues in terms of duration

#Checking if we have invalid rides in terms of station, rides that are in stations
which have already been removed:
SELECT *
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
end_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL)
OR
start_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL); --127 invalid rides that must
be removed

#Check if we have 2 stations with the same location:
SELECT latitude, longitude, COUNT(*)
FROM
(
SELECT
id, latitude, longitude
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`)
GROUP BY latitude, longitude
HAVING COUNT(*) > 1; --No!


#Handle station names with double spaces:

SELECT
name,
replace (name,'  ',' ')
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE INSTR(name,'  ') > 0 ; #3 stations that should be fixed


#Outliers in terms of ride duration:
-- Assuming outlier values are outside the range of mean +/- 3 standard deviations.
SELECT *
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
duration >=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
+ 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
OR
duration <=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
- 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new` ); --112 outliers

#Check for stations that exists in the rides table but not in the stations table:
SELECT
DISTINCT r.end_station_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
LEFT JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
WHERE e.id IS NULL; --15 invalid stations

#There are 774 rides that are invalid in terms of invalid ending station:
SELECT rental_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE end_station_id IN
(SELECT
DISTINCT r.end_station_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
LEFT JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
WHERE e.id IS NULL);

#Check for nulls in new table
SELECT DISTINCT
bike_model,end_station_logical_terminal,start_station_logical_terminal,end_station_priority_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`; -- all this columns are irrelevant

#Check values in column Locked:
SELECT DISTINCT locked
FROM`data-analysis-389112.Project_Google.cycle_stations_pro`; --all stations are unlocked!

#Check for duration miss calculation
SELECT rental_id
```

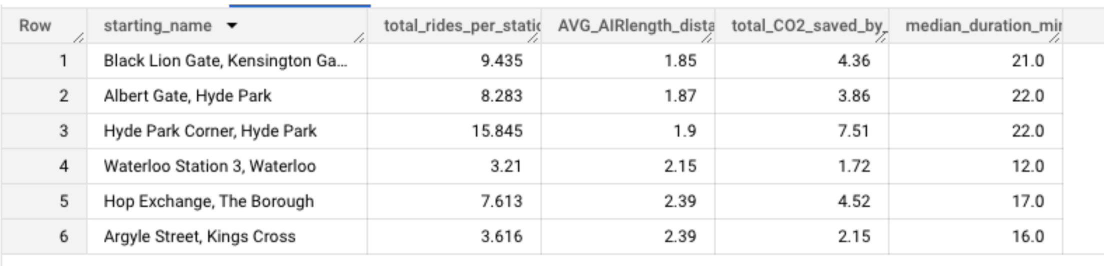
```
FROM
(
SELECT
rental_id,
duration,
TIMESTAMP_DIFF(end_date, start_date, SECOND) AS calculated_difference
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`)
WHERE duration != calculated_difference; #Duration values are valid!

# Ensure data integrity for the "start_station_id" and"end_station_id" columns?
SELECT COUNT(*) AS missing_station_id_count
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE start_station_id IS NULL OR end_station_id IS NULL; --there are no rows with
null values for those columns

#The cte + Staistics:
#Used Inner Join to remove the 15 ending stations that appear in ride table but are
missing from the station table (*removed 774)
#Overall removed 1013 rides (We also removed outliers, and the 127 that pass
through stations that have already been removed = installed is false or there is a
value for the removal date column), we returned - 48002 rides:
WITH table_cleaned AS
(SELECT
rental_id, bike_id, duration AS duration_in_seconds, duration / 60 AS
duration_in_minutes,
start_date, EXTRACT(MONTH FROM start_date) start_month, EXTRACT(DAYOFWEEK FROM
start_date) start_dayofweek,EXTRACT(HOUR FROM start_date) start_hour,
start_station_id, replace (s.name,'  ',' ') starting_name, s.docks_count
starting_dock_count,
ST_GEOGPOINT (s.longitude, s.latitude) starting_geo_point,
end_date, EXTRACT(MONTH FROM end_date) end_month, EXTRACT(DAYOFWEEK FROM end_date)
end_dayofweek,EXTRACT(HOUR FROM end_date) end_hour, end_station_id, replace (e.name
,'  ',' ')ending_name, e.docks_count ending_dock_count,
ST_GEOGPOINT(e.longitude, e. latitude) ending_geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(s.longitude, s.latitude), ST_GEOGPOINT(e.longitude,
e.latitude))) / 1000 AS trip_distance_km
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS s
ON r.start_station_id = s.id
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
WHERE
rental_id NOT IN (# remove invalid stations
SELECT rental_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
end_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL)
OR
start_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL))
AND
rental_id NOT IN (# remove outliers
SELECT rental_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
duration >=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
+ 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
OR
duration <=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
- 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new` )))
SELECT
```

```
ROUND(AVG(trip_distance_km),2) as avg_distance_km,
APPROX_QUANTILES(trip_distance_km, 2)[OFFSET(1)] AS median_trip_distance_km,
ROUND(MIN(trip_distance_km),2) as min_distance_km,
ROUND(MAX(trip_distance_km),2) as max_distance_km,
ROUND(AVG(duration_in_minutes),2) as avg_duration_minutes,
APPROX_QUANTILES(duration_in_minutes, 2)[OFFSET(1)] AS median_duration_minutes,
ROUND(MIN(duration_in_minutes),2) as min_duration_minutes,
ROUND(MAX(duration_in_minutes),2) as max_duration_minutes
FROM table_cleaned;
```



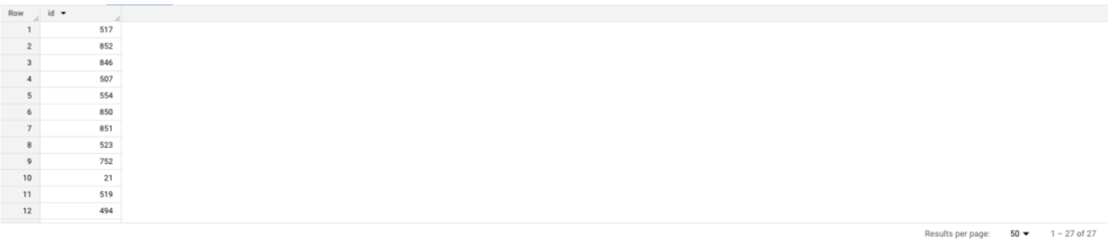
Row	avg_distance_km	median_trip_distance	min_distance_km	max_distance_km	avg_duration_minute	median_duration_mi	min_duration_minute	max_duration_minut
1	2.02	1.829	0.0	12.96	28.79	20.0	1.0	328.0

```
#Basic Statistics to start the presentaion:
WITH table_cleaned AS
(SELECT
rental_id, bike_id, duration AS duration_in_seconds, duration / 60 AS
duration_in_minutes,
start_date, EXTRACT(MONTH FROM start_date) start_month, EXTRACT(DAYOFWEEK FROM
start_date) start_dayofweek,EXTRACT(HOUR FROM start_date) start_hour,
start_station_id, replace (s.name,'  ',' ') starting_name, s.docks_count
starting_dock_count,
ST_GEOGPOINT (s.longitude, s.latitude) starting_geo_point,
end_date, EXTRACT(MONTH FROM end_date) end_month, EXTRACT(DAYOFWEEK FROM end_date)
end_dayofweek,EXTRACT(HOUR FROM end_date) end_hour, end_station_id, replace (e.name
,'  ',' ')ending_name, e.docks_count ending_dock_count,
ST_GEOGPOINT(e.longitude, e. latitude) ending_geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(s.longitude, s.latitude), ST_GEOGPOINT(e.longitude,
e.latitude))) / 1000 AS trip_distance_km
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS s
ON r.start_station_id = s.id
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
WHERE
rental_id NOT IN (
SELECT rental_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
end_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL)
OR
start_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL))
AND
rental_id NOT IN (
SELECT rental_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
duration >=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
+ 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
OR
duration <=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
- 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new` )))

SELECT
starting_name,
COUNT(*) / 1000 AS total_rides_per_station_in_thousands,
ROUND(AVG(trip_distance_km),2) AS AVG_AIRlength_distance,
ROUND((SUM(trip_distance_km) * 0.249) / 1000, 2) AS
total_CO2_saved_by_station_in_ton,
APPROX_QUANTILES(duration_in_minutes, 2)[OFFSET(1)] AS median_duration_minutes
FROM table_cleaned
GROUP BY starting_name
ORDER BY AVG_AIRlength_distance;
```



Row	starting_name	total_rides_per_static	AVG_AIRlength_dista	total_CO2_saved_by_	median_duration_mi
1	Black Lion Gate, Kensington Ga...	9.435	1.85	4.36	21.0
2	Albert Gate, Hyde Park	8.283	1.87	3.86	22.0
3	Hyde Park Corner, Hyde Park	15.845	1.9	7.51	22.0
4	Waterloo Station 3, Waterloo	3.21	2.15	1.72	12.0
5	Hop Exchange, The Borough	7.613	2.39	4.52	17.0
6	Argyle Street, Kings Cross	3.616	2.39	2.15	16.0

```
#First Question: Does the distance from the center of London have an affect on the utilization of the station?
#London city center location: (latitude and longitude values)
https://www.findlatitudeandlongitude.com/l/London+city+centre/5715707/

#First of all, here are all the stations that exist but have had no rides during Q1 of 2021:
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE id NOT IN(
SELECT DISTINCT s.id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro` AS s
JOIN
`data-analysis-389112.Project_Google.cycle_hire_new` AS r
ON s.id = r.start_station_id OR s.id = r.end_station_id);--there are 27 stations that have no usage.
```



Row	id
1	517
2	852
3	846
4	507
5	554
6	850
7	851
8	523
9	752
10	21
11	519
12	494

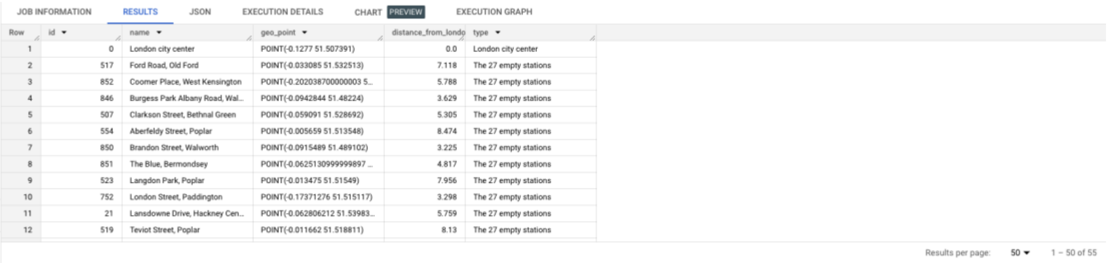
```
#The average distance from London city center of the 27 stations that have no rides:
SELECT AVG(distance_from_london_center_in_km) AS
average_distance_from_london_center_in_km
FROM
(
SELECT *, ST_GEOGPOINT(longitude, latitude) AS geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(longitude, latitude), ST_GEOGPOINT(-0.1277,
51.507391))) / 1000 AS distance_from_london_center_in_km
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE id NOT IN(
SELECT DISTINCT s.id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro` AS s
JOIN
`data-analysis-389112.Project_Google.cycle_hire_new` AS r
ON s.id = r.start_station_id OR s.id = r.end_station_id));#5.68 km

#Now, let's find the average distance from London city center of the top 27
stations (by the number of rides during Q1 2021):
#The subquery: *select the top 27 stations by the num of rides:
SELECT end_station_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
GROUP BY end_station_id
ORDER BY COUNT(*) DESC
LIMIT 28)
#The query:
SELECT AVG(distance_from_london_center_in_km) AS
average_distance_from_london_center_in_km
FROM
(
SELECT *, ST_GEOGPOINT(longitude, latitude) AS geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(longitude, latitude), ST_GEOGPOINT(-0.1277,
51.507391))) / 1000 AS distance_from_london_center_in_km
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE id IN(
SELECT end_station_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
GROUP BY end_station_id
ORDER BY COUNT(*) DESC
LIMIT 28)); #2.35 km
```

#Next, we visualized our results using BigQuery Geo Viz: a web tool for visualization of geospatial data in BigQuery using Google Maps APIs.

https://cloud.google.com/bigquery/docs/geospatial-get-started

#For GeoViz:
#For each of the 54 (27 worst + 27 best) stations, we will return it id, name, geopoint and distance from the center of London
*Using UNION ALL i've also added the Center of London as a point
*I couldn't use the CTE for this query because we're also checking for the 27 worst station's without any rides, which means they won't show up (i'm using inner JOIN and not an OUTER (left / right) JOIN:

```
SELECT id, name, ST_GEOGPOINT(longitude, latitude) AS geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(longitude, latitude), ST_GEOGPOINT(-0.1277,
51.507391))) / 1000 AS distance_from_london_center_in_km, "Top 27 Stations" AS type
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE id IN(
SELECT end_station_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
GROUP BY end_station_id
ORDER BY COUNT(*) DESC
LIMIT 28)

UNION ALL

SELECT id, name, ST_GEOGPOINT(longitude, latitude) AS geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(longitude, latitude), ST_GEOGPOINT(-0.1277,
51.507391))) / 1000 AS distance_from_london_center_in_km, "The 27 empty stations"
AS type
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE id NOT IN(
SELECT DISTINCT s.id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro` AS s
JOIN
`data-analysis-389112.Project_Google.cycle_hire_new` AS r
ON s.id = r.start_station_id OR s.id = r.end_station_id)

UNION ALL

SELECT 0, "London city center", ST_GEOGPOINT(-0.1277, 51.507391) AS geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(-0.1277, 51.507391), ST_GEOGPOINT(-0.1277,
51.507391))) / 1000 AS distance_from_london_center_in_km, "London city center";
```



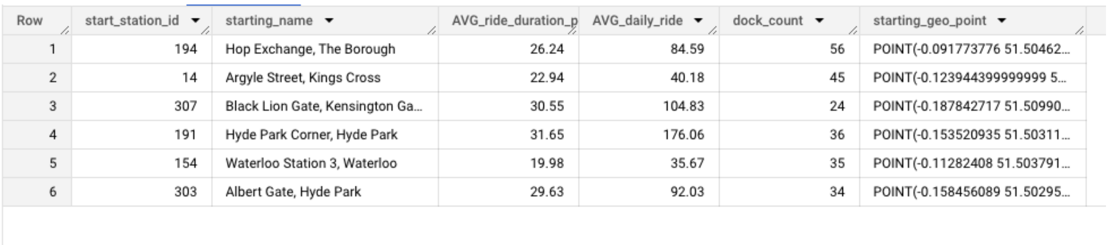
Row	id	name	geo_point	distance_from_londo	type
1	0	London city center	POINT(-0.1277 51.507391)	0.0	London city center
2	517	Ford Road, Old Ford	POINT(-0.033085 51.532513)	7.118	The 27 empty stations
3	852	Coomer Place, West Kensington	POINT(-0.202038700000003 5...	5.788	The 27 empty stations
4	846	Burgess Park Albany Road, Wal...	POINT(-0.0942844 51.48224)	3.629	The 27 empty stations
5	507	Clarkson Street, Bethnal Green	POINT(-0.059091 51.528692)	5.305	The 27 empty stations
6	554	Aberfeldy Street, Poplar	POINT(-0.005659 51.513548)	8.474	The 27 empty stations
7	850	Brandon Street, Walworth	POINT(-0.0915489 51.489102)	3.225	The 27 empty stations
8	851	The Blue, Bermondsey	POINT(-0.0625130999999897 ...	4.817	The 27 empty stations
9	523	Langdon Park, Poplar	POINT(-0.013475 51.51549)	7.956	The 27 empty stations
10	752	London Street, Paddington	POINT(-0.17371276 51.515117)	3.298	The 27 empty stations
11	21	Lansdowne Drive, Hackney Cen...	POINT(-0.062806212 51.53983...	5.759	The 27 empty stations
12	519	Teviot Street, Poplar	POINT(-0.011662 51.518811)	8.13	The 27 empty stations

Results per page: 50 ▾ 1 – 50 of 55

```
    #Second Question: Find the best and worst Starting stations in terms of average
  amount of daily rides, the average duration of those rides, and the dock count of
                                                            each station(Multivariate)
#Using our findings we are able to find the differences in utilization between each
of the 6 starting stations:
WITH table_cleaned AS
(SELECT
rental_id, bike_id, duration AS duration_in_seconds, duration / 60 AS
duration_in_minutes,
start_date, EXTRACT(MONTH FROM start_date) start_month, EXTRACT(DAYOFWEEK FROM
start_date) start_dayofweek,EXTRACT(HOUR FROM start_date) start_hour,
start_station_id, replace (s.name,'  ',' ') starting_name, s.docks_count
starting_dock_count,
ST_GEOGPOINT (s.longitude, s.latitude) starting_geo_point,
end_date, EXTRACT(MONTH FROM end_date) end_month, EXTRACT(DAYOFWEEK FROM end_date)
end_dayofweek,EXTRACT(HOUR FROM end_date) end_hour, end_station_id, replace (e.name
,'  ',' ')ending_name, e.docks_count ending_dock_count,
ST_GEOGPOINT(e.longitude, e. latitude) ending_geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(s.longitude, s.latitude), ST_GEOGPOINT(e.longitude,
e.latitude))) / 1000 AS trip_distance_km
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS s
ON r.start_station_id = s.id
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
```

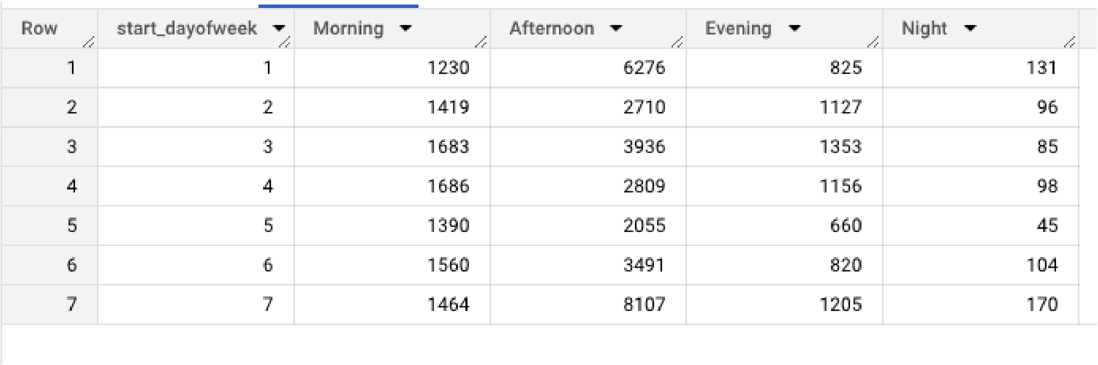
```
WHERE
rental_id NOT IN (
SELECT rental_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
end_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL)
OR
start_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL))
AND
rental_id NOT IN (
SELECT rental_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
duration >=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
+ 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
OR
duration <=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
- 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new` )))
,
#another CTE to get the geoPoint for each starting station:
geo_for_each_starting_station AS
(SELECT
```

```sql
start_station_id,
ST_GEOGPOINT (MAX(s.longitude), MAX(s.latitude)) starting_geo_point,
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS s
ON r.start_station_id = s.id
GROUP BY start_station_id
)
SELECT the_table.*, starting_geo_point
FROM
(
SELECT
start_station_id,
starting_name,
ROUND(AVG(SUM_daily_rides_minutes_per_station) / AVG(count_rides),2) AS
AVG_ride_duration_per_ride_daily_minutes, #the calculation is = the average total duration per each day / the average number of rides per each day
ROUND(AVG(count_rides),2) AS AVG_daily_ride, #the average daily number of rides
MAX(docks_count) AS dock_count #because were using GROUP BY we must aggregate, MAX has no effect because the value for docks_count will be the same for each row with this station id
FROM(
SELECT
table_cleaned.start_station_id,
starting_name,
SUM(duration_in_minutes) AS SUM_daily_rides_minutes_per_station,
starting_dock_count AS docks_count,
EXTRACT(DAY FROM start_date) DAY_start,
EXTRACT(MONTH FROM start_date) MONTH_start,
COUNT(*) AS count_rides
FROM table_cleaned
GROUP BY start_station_id,starting_name,DAY_start,MONTH_start, docks_count #Group by each starting station and day -> we want to calculate daily values
ORDER BY start_station_id,MONTH_start,DAY_start DESC)
GROUP BY start_station_id,starting_name #Group by each starting station
) AS the_table
JOIN geo_for_each_starting_station #So we can get the GeoPoint for each starting station (each row)
ON the_table.start_station_id = geo_for_each_starting_station.start_station_id;
```



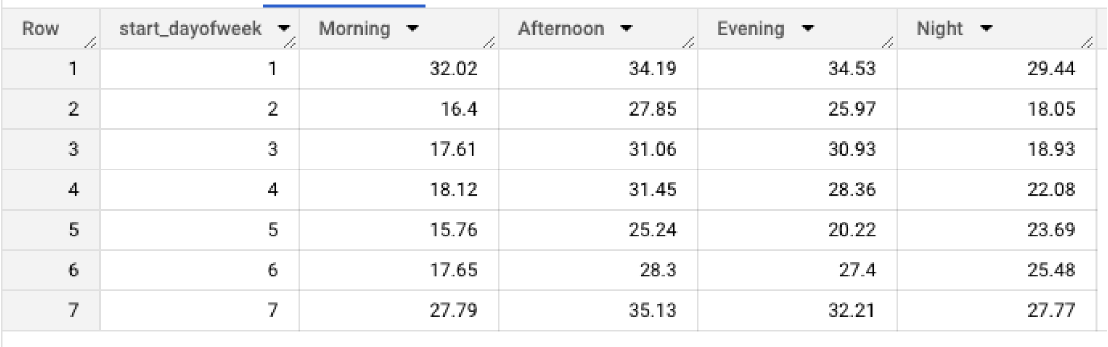
Row	start_station_id	starting_name	AVG_ride_duration_p	AVG_daily_ride	dock_count	starting_geo_point
1	194	Hop Exchange, The Borough	26.24	84.59	56	POINT(-0.091773776 51.50462...
2	14	Argyle Street, Kings Cross	22.94	40.18	45	POINT(-0.123944399999999 5...
3	307	Black Lion Gate, Kensington Ga...	30.55	104.83	24	POINT(-0.187842717 51.50990...
4	191	Hyde Park Corner, Hyde Park	31.65	176.06	36	POINT(-0.153520935 51.50311...
5	154	Waterloo Station 3, Waterloo	19.98	35.67	35	POINT(-0.11282408 51.503791...
6	303	Albert Gate, Hyde Park	29.63	92.03	34	POINT(-0.158456089 51.50295...

```
#Third Question: Analyze rental patterns by day of the week and hour
#The first query will return the total number of rides for each day of the week and
time of day:
WITH table_cleaned AS
(SELECT
rental_id, bike_id, duration AS duration_in_seconds, duration / 60 AS
duration_in_minutes,
start_date, EXTRACT(MONTH FROM start_date) start_month, EXTRACT(DAYOFWEEK FROM
start_date) start_dayofweek,EXTRACT(HOUR FROM start_date) start_hour,
start_station_id, replace (s.name,'  ',' ') starting_name, s.docks_count
starting_dock_count,
ST_GEOGPOINT (s.longitude, s.latitude) starting_geo_point,
end_date, EXTRACT(MONTH FROM end_date) end_month, EXTRACT(DAYOFWEEK FROM end_date)
end_dayofweek,EXTRACT(HOUR FROM end_date) end_hour, end_station_id, replace (e.name
,'  ',' ')ending_name, e.docks_count ending_dock_count,
ST_GEOGPOINT(e.longitude, e. latitude) ending_geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(s.longitude, s.latitude), ST_GEOGPOINT(e.longitude,
e.latitude))) / 1000 AS trip_distance_km
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS s
ON r.start_station_id = s.id
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
WHERE
rental_id NOT IN (
SELECT rental_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
end_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL)
OR
start_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL))
AND
rental_id NOT IN (
SELECT rental_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
duration >=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
+ 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
OR
duration <=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
- 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new` )))

SELECT
start_dayofweek, #+1 beacuse the date value is in UTC, and London is one hour
ahead:
COUNT(CASE WHEN start_hour+1 IN (6,7,8,9,10,11,12) THEN 1 END) AS Morning,
COUNT(CASE WHEN start_hour+1 IN (13,14,15,16,17,18) THEN 1 END) AS Afternoon,
COUNT(CASE WHEN start_hour+1 IN (19,20,21,22) THEN 1 END) AS Evening,
COUNT(CASE WHEN start_hour+1 IN (23,0,1,2,3,4,5) THEN 1 END) AS Night
FROM #Morning - 6 to 12 am, Afternoon - 1 to 6 pm, Evening - 7 to 10 pm, night - 11
pm to 5 am
table_cleaned
GROUP BY start_dayofweek
ORDER BY start_dayofweek;
```



Row	start_dayofweek	Morning	Afternoon	Evening	Night
1	1	1230	6276	825	131
2	2	1419	2710	1127	96
3	3	1683	3936	1353	85
4	4	1686	2809	1156	98
5	5	1390	2055	660	45
6	6	1560	3491	820	104
7	7	1464	8107	1205	170

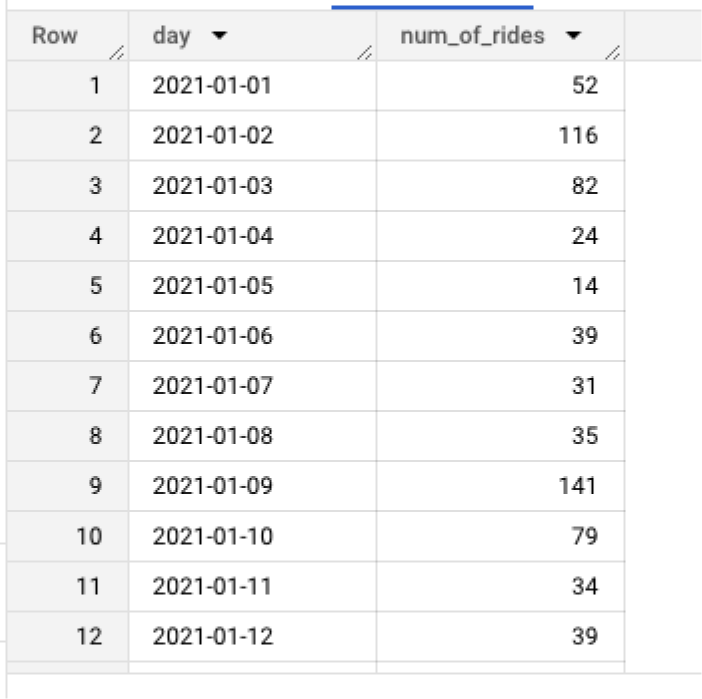
```
#The second query will return the average ride duration in minutes for each day of
the week and time of day:
WITH table_cleaned AS
(SELECT
rental_id, bike_id, duration AS duration_in_seconds, duration / 60 AS
duration_in_minutes,
start_date, EXTRACT(MONTH FROM start_date) start_month, EXTRACT(DAYOFWEEK FROM
start_date) start_dayofweek,EXTRACT(HOUR FROM start_date) start_hour,
start_station_id, replace (s.name,'  ',' ') starting_name, s.docks_count
starting_dock_count,
ST_GEOGPOINT (s.longitude, s.latitude) starting_geo_point,
end_date, EXTRACT(MONTH FROM end_date) end_month, EXTRACT(DAYOFWEEK FROM end_date)
end_dayofweek,EXTRACT(HOUR FROM end_date) end_hour, end_station_id, replace (e.name
,'  ',' ')ending_name, e.docks_count ending_dock_count,
ST_GEOGPOINT(e.longitude, e. latitude) ending_geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(s.longitude, s.latitude), ST_GEOGPOINT(e.longitude,
e.latitude))) / 1000 AS trip_distance_km
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS s
ON r.start_station_id = s.id
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
WHERE
rental_id NOT IN (
SELECT rental_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
end_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL)
OR
start_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL))
AND
rental_id NOT IN (
SELECT rental_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
duration >=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
+ 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
OR
duration <=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
- 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new` )))

SELECT
start_dayofweek, #+1 beacuse the date value is in UTC, and London is one hour
ahead:
ROUND(AVG(CASE WHEN start_hour+1 IN (6,7,8,9,10,11,12) THEN duration_in_minutes
END),2) AS Morning,
ROUND(AVG(CASE WHEN start_hour+1 IN (13,14,15,16,17,18) THEN duration_in_minutes
END),2) AS Afternoon,
ROUND(AVG(CASE WHEN start_hour+1 IN (19,20,21,22) THEN duration_in_minutes END),2)
AS Evening,
ROUND(AVG(CASE WHEN start_hour+1 IN (23,0,1,2,3,4,5) THEN duration_in_minutes
END),2) AS Night,
#Morning - 6 to 12 am, Afternoon - 1 to 6 pm, Evening - 7 to 10 pm, night - 11 pm
to 5 am
FROM
table_cleaned
GROUP BY start_dayofweek
ORDER BY start_dayofweek;
```



Row	start_dayofweek	Morning	Afternoon	Evening	Night
1	1	32.02	34.19	34.53	29.44
2	2	16.4	27.85	25.97	18.05
3	3	17.61	31.06	30.93	18.93
4	4	18.12	31.45	28.36	22.08
5	5	15.76	25.24	20.22	23.69
6	6	17.65	28.3	27.4	25.48
7	7	27.79	35.13	32.21	27.77

```
#Prediction: Predict how many rentals will be made in the next month (April 2021)
in “Albert Gate, Hyde Park” bike station.
#Albert Gate, Hyde Park - ID: 303
#For station - Albert Gate, Hyde Park, return the number of rides per each day in
Q1 2021:
WITH table_cleaned AS
(SELECT
rental_id, bike_id, duration AS duration_in_seconds, duration / 60 AS
duration_in_minutes,
start_date, EXTRACT(MONTH FROM start_date) start_month, EXTRACT(DAYOFWEEK FROM
start_date) start_dayofweek,EXTRACT(HOUR FROM start_date) start_hour,
start_station_id, replace (s.name,' ',' ') starting_name, s.docks_count
starting_dock_count,
ST_GEOGPOINT (s.longitude, s.latitude) starting_geo_point,
end_date, EXTRACT(MONTH FROM end_date) end_month, EXTRACT(DAYOFWEEK FROM end_date)
end_dayofweek,EXTRACT(HOUR FROM end_date) end_hour, end_station_id, replace (e.name
,' ',' ') ending_name, e.docks_count ending_dock_count,
ST_GEOGPOINT(e.longitude, e. latitude) ending_geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(s.longitude, s.latitude), ST_GEOGPOINT(e.longitude,
e.latitude))) / 1000 AS trip_distance_km
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS s
ON r.start_station_id = s.id
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
WHERE
rental_id NOT IN (
SELECT rental_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
end_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL)
OR
start_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL))
AND
rental_id NOT IN (
SELECT rental_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
duration >=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
+ 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
OR
duration <=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
- 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new` )))
SELECT
EXTRACT(DATE FROM start_date) AS day, COUNT(*) AS num_of_rides
FROM
```

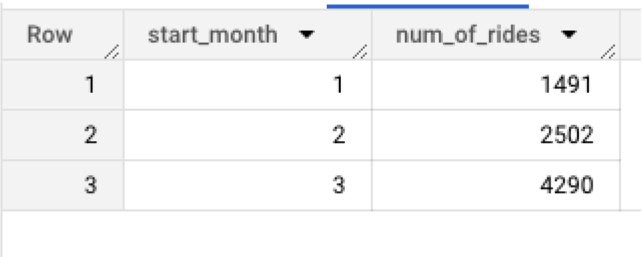
```
table_cleaned
WHERE start_station_id IN(
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE name LIKE '%Albert Gate, Hyde Park%')
GROUP BY day
ORDER BY day;
```



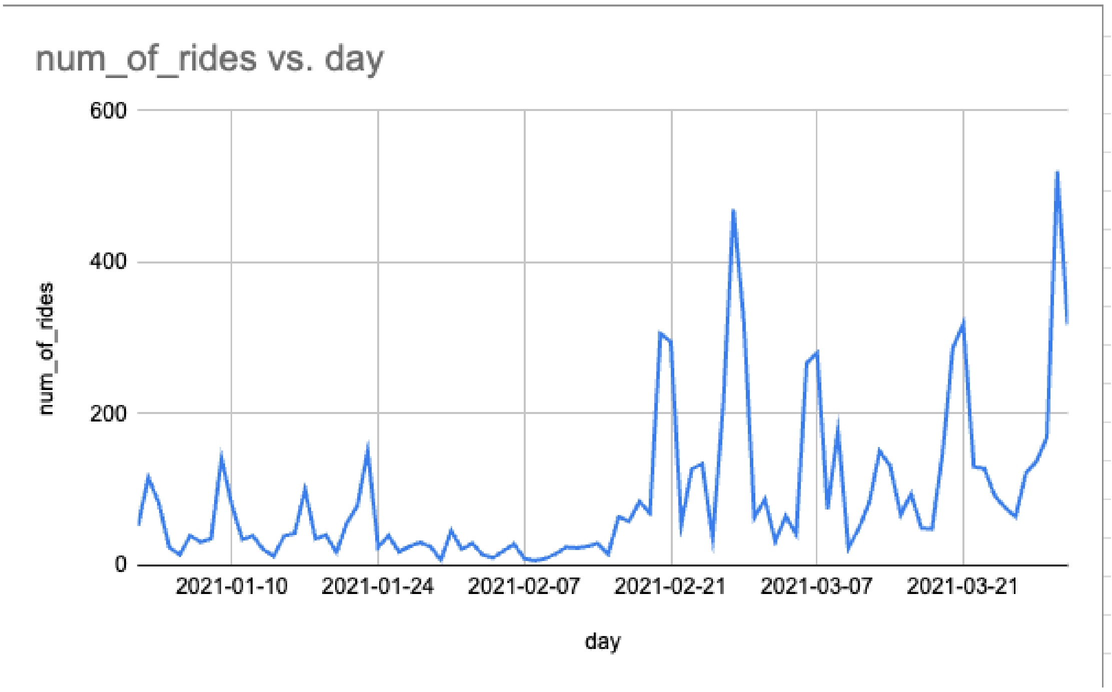
Row	day	num_of_rides
1	2021-01-01	52
2	2021-01-02	116
3	2021-01-03	82
4	2021-01-04	24
5	2021-01-05	14
6	2021-01-06	39
7	2021-01-07	31
8	2021-01-08	35
9	2021-01-09	141
10	2021-01-10	79
11	2021-01-11	34
12	2021-01-12	39

```
#Query for the number of rides in the staion for each month:
WITH table_cleaned AS
(SELECT
rental_id, bike_id, duration AS duration_in_seconds, duration / 60 AS
duration_in_minutes,
start_date, EXTRACT(MONTH FROM start_date) start_month, EXTRACT(DAYOFWEEK FROM
start_date) start_dayofweek,EXTRACT(HOUR FROM start_date) start_hour,
start_station_id, s.name starting_name, s.docks_count starting_dock_count,
ST_GEOGPOINT (s.longitude, s.latitude) starting_geo_point,
end_date, EXTRACT(MONTH FROM end_date) end_month, EXTRACT(DAYOFWEEK FROM end_date)
end_dayofweek,EXTRACT(HOUR FROM end_date) end_hour, end_station_id, e.name
ending_name, e.docks_count ending_dock_count,
ST_GEOGPOINT(e.longitude, e. latitude) ending_geo_point,
ROUND(ST_DISTANCE(ST_GEOGPOINT(s.longitude, s.latitude), ST_GEOGPOINT(e.longitude,
e.latitude))) / 1000 AS trip_distance_km
FROM `data-analysis-389112.Project_Google.cycle_hire_new` AS r
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS s
ON r.start_station_id = s.id
JOIN `data-analysis-389112.Project_Google.cycle_stations_pro` AS e
ON r.end_station_id = e.id
WHERE
rental_id NOT IN (
SELECT rental_id
FROM
`data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
end_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL)
OR
start_station_id IN (
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE installed = false OR removal_date IS NOT NULL))
AND
rental_id NOT IN (
SELECT rental_id
FROM `data-analysis-389112.Project_Google.cycle_hire_new`
WHERE
duration >=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
+ 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
OR
duration <=
(SELECT
AVG(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new`)
- 3 * (SELECT STDDEV(duration)
FROM `data-analysis-389112.Project_Google.cycle_hire_new` )))
```

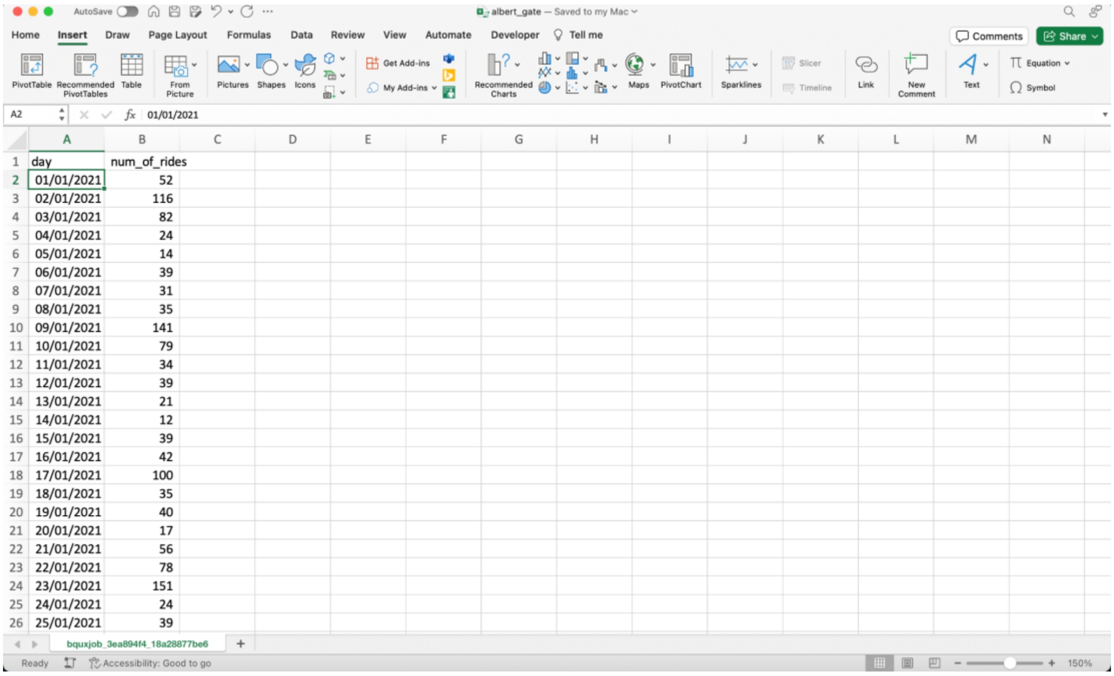
```
SELECT
start_month, COUNT(*) AS num_of_rides
FROM
table_cleaned
WHERE start_station_id IN(
SELECT id
FROM
`data-analysis-389112.Project_Google.cycle_stations_pro`
WHERE name LIKE '%Albert Gate, Hyde Park%')
GROUP BY start_month
ORDER BY start_month;
```



Row	start_month	num_of_rides
1	1	1491
2	2	2502
3	3	4290



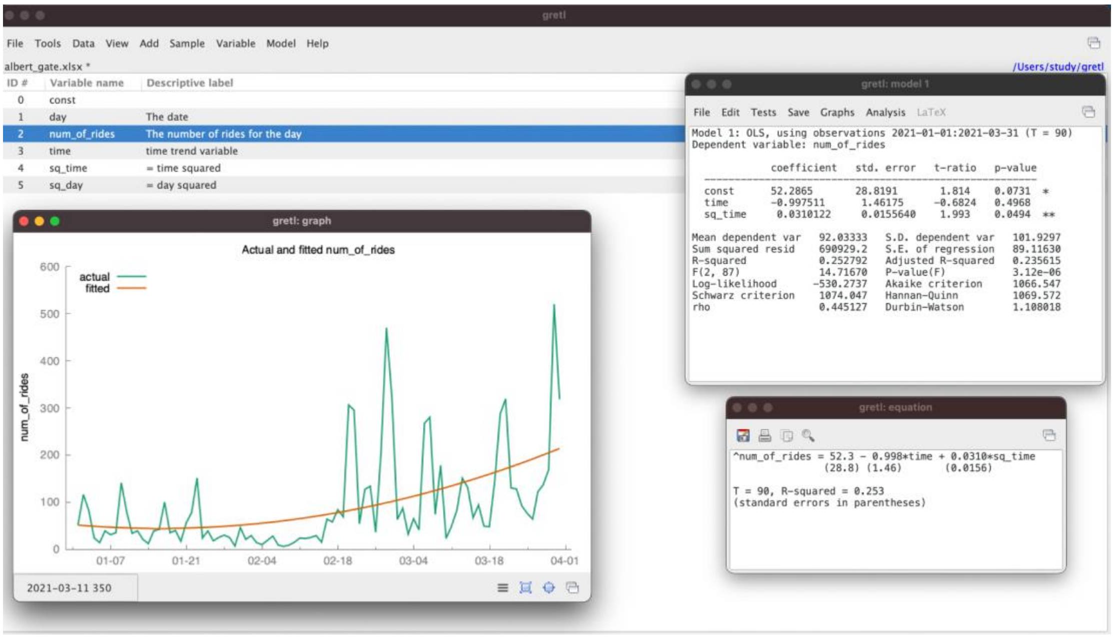
#We will move this table to sheets, and download is as an excel file so we could load it into gretl - a statistical package able to run a linear regression (ordinary least squares (OLS) model):



albert_gate — Saved to my Mac
Home Insert Draw Page Layout Formulas Data Review View Automate Developer Tell me
Comments Share
A2 01/01/2021
bquxjob_3ea894f4_18a28877be6
Ready Accessibility: Good to go 150%

	A	B
1	day	num_of_rides
2	01/01/2021	52
3	02/01/2021	116
4	03/01/2021	82
5	04/01/2021	24
6	05/01/2021	14
7	06/01/2021	39
8	07/01/2021	31
9	08/01/2021	35
10	09/01/2021	141
11	10/01/2021	79
12	11/01/2021	34
13	12/01/2021	39
14	13/01/2021	21
15	14/01/2021	12
16	15/01/2021	39
17	16/01/2021	42
18	17/01/2021	100
19	18/01/2021	35
20	19/01/2021	40
21	20/01/2021	17
22	21/01/2021	56
23	22/01/2021	78
24	23/01/2021	151
25	24/01/2021	24
26	25/01/2021	39

The equation: num_of_rides = α + β1 * time + β2 * time ^ 2

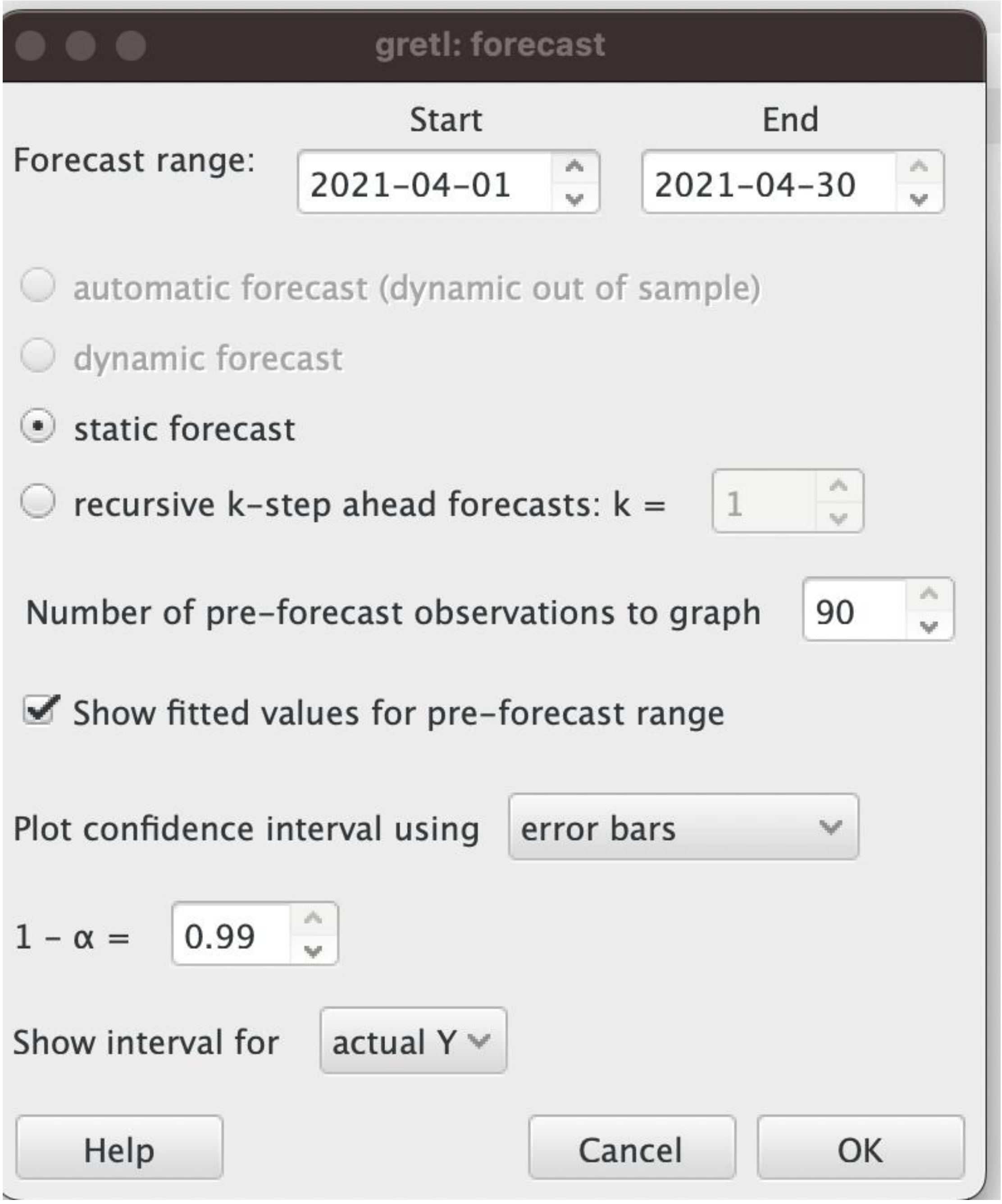


ID #	Variable name	Descriptive label
0	const	
1	day	The date
2	num_of_rides	The number of rides for the day
3	time	time trend variable
4	sq_time	= time squared
5	sq_day	= day squared

```
Model 1: OLS, using observations 2021-01-01:2021-03-31 (T = 90)
Dependent variable: num_of_rides

             coefficient   std. error   t-ratio   p-value
  ---------------------------------------------------------
  const      52.2865       28.8191       1.814    0.0731   *
  time       -0.997511      1.46175     -0.6824   0.4968
  sq_time     0.0310122     0.0155640    1.993    0.0494   **

Mean dependent var   92.03333   S.D. dependent var   101.9297
Sum squared resid    690929.2   S.E. of regression   89.11630
R-squared            0.252792   Adjusted R-squared   0.235615
F(2, 87)             14.71670   P-value(F)           3.12e-06
Log-likelihood      -530.2737   Akaike criterion     1066.547
Schwarz criterion    1074.047   Hannan-Quinn         1069.572
rho                  0.445127   Durbin-Watson        1.108018
```

```
^num_of_rides = 52.3 - 0.998*time + 0.0310*sq_time
               (28.8) (1.46)        (0.0156)

T = 90, R-squared = 0.253
(standard errors in parentheses)
```

now, we use of model to predict - forecast the next 30 days: the month of April 2020:



gretl: forecast

Forecast range: Start 2021-04-01 End 2021-04-30

○ automatic forecast (dynamic out of sample)

○ dynamic forecast

● static forecast

○ recursive k-step ahead forecasts: k = 1

Number of pre-forecast observations to graph 90

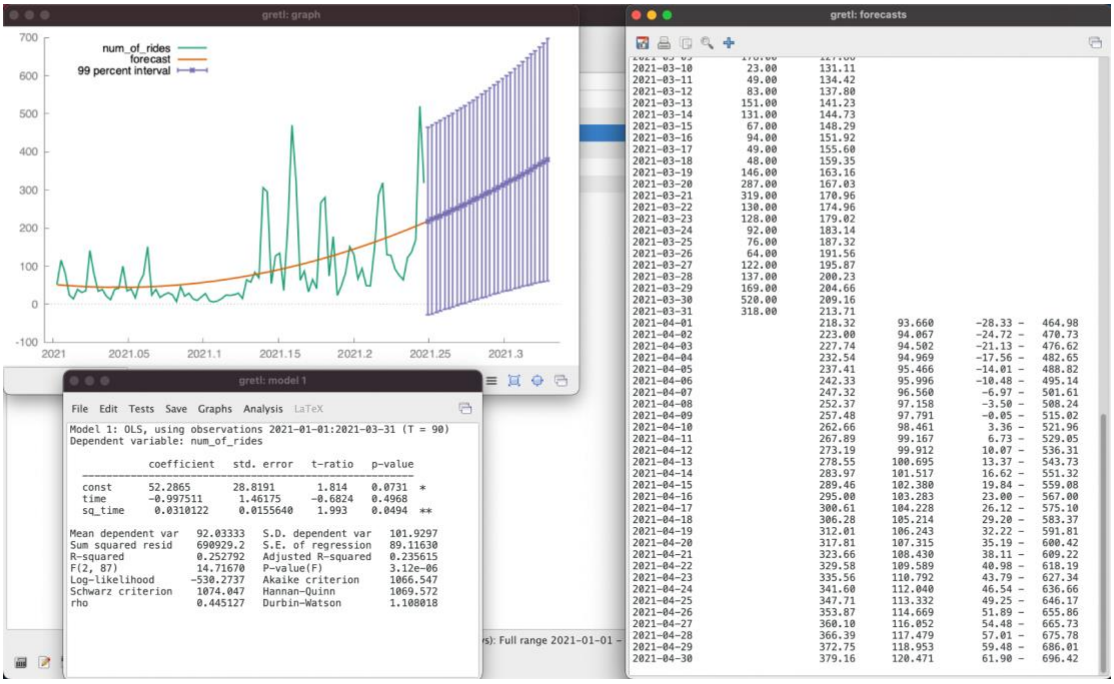
☑ Show fitted values for pre-forecast range

Plot confidence interval using error bars

1 − α = 0.99

Show interval for actual Y

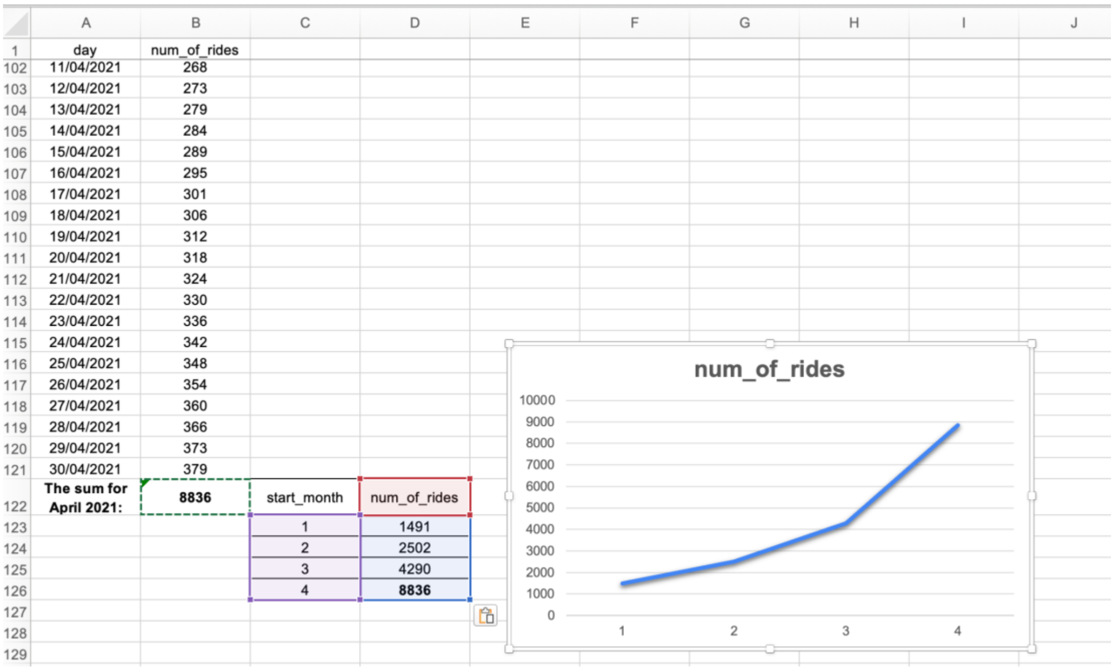
`(with a significance level of α = 0.01 -> 1-α = 99%)`



```
Model 1: OLS, using observations 2021-01-01:2021-03-31 (T = 90)
Dependent variable: num_of_rides

             coefficient   std. error   t-ratio   p-value
  ---------------------------------------------------------
  const       52.2865       28.8191       1.814    0.0731  *
  time        -0.997511      1.46175     -0.6824   0.4968
  sq_time      0.0310122     0.0155640    1.993    0.0494  **

Mean dependent var   92.03333   S.D. dependent var   101.9297
Sum squared resid    690929.2   S.E. of regression   89.11630
R-squared            0.252792   Adjusted R-squared   0.235615
F(2, 87)             14.71670   P-value(F)           3.12e-06
Log-likelihood      -530.2737   Akaike criterion     1066.547
Schwarz criterion    1074.047   Hannan-Quinn         1069.572
rho                  0.445127   Durbin-Watson        1.108018
```

```
2021-03-10        23.00      131.11
2021-03-11        49.00      134.42
2021-03-12        83.00      137.80
2021-03-13       151.00      141.23
2021-03-14       131.00      144.73
2021-03-15        67.00      148.29
2021-03-16        94.00      151.92
2021-03-17        49.00      155.60
2021-03-18        48.00      159.35
2021-03-19       146.00      163.16
2021-03-20       287.00      167.03
2021-03-21       319.00      170.96
2021-03-22       130.00      174.96
2021-03-23       128.00      179.02
2021-03-24        92.00      183.14
2021-03-25        76.00      187.32
2021-03-26        64.00      191.56
2021-03-27       122.00      195.87
2021-03-28       137.00      200.23
2021-03-29       169.00      204.66
2021-03-30       520.00      209.16
2021-03-31       318.00      213.71
2021-04-01                   218.32      93.660     -28.33 -   464.98
2021-04-02                   223.00      94.067     -24.72 -   470.73
2021-04-03                   227.74      94.502     -21.13 -   476.62
2021-04-04                   232.54      94.969     -17.56 -   482.65
2021-04-05                   237.41      95.466     -14.01 -   488.82
2021-04-06                   242.33      95.996     -10.48 -   495.14
2021-04-07                   247.32      96.560      -6.97 -   501.61
2021-04-08                   252.37      97.158      -3.50 -   508.24
2021-04-09                   257.48      97.791      -0.05 -   515.02
2021-04-10                   262.66      98.461       3.36 -   521.96
2021-04-11                   267.89      99.167       6.73 -   529.05
2021-04-12                   273.19      99.912      10.07 -   536.31
2021-04-13                   278.55     100.695      13.37 -   543.73
2021-04-14                   283.97     101.517      16.62 -   551.32
2021-04-15                   289.46     102.380      19.84 -   559.08
2021-04-16                   295.00     103.283      23.00 -   567.00
2021-04-17                   300.61     104.228      26.12 -   575.10
2021-04-18                   306.28     105.214      29.20 -   583.37
2021-04-19                   312.01     106.243      32.22 -   591.81
2021-04-20                   317.81     107.315      35.19 -   600.42
2021-04-21                   323.66     108.430      38.11 -   609.22
2021-04-22                   329.58     109.589      40.98 -   618.19
2021-04-23                   335.56     110.792      43.79 -   627.34
2021-04-24                   341.60     112.040      46.54 -   636.66
2021-04-25                   347.71     113.332      49.25 -   646.17
2021-04-26                   353.87     114.669      51.89 -   655.86
2021-04-27                   360.10     116.052      54.48 -   665.73
2021-04-28                   366.39     117.479      57.01 -   675.78
2021-04-29                   372.75     118.953      59.48 -   686.01
2021-04-30                   379.16     120.471      61.90 -   696.42
```

now, all that’s left is to sum the predicted values of April 2021 and we’ll receive an answer: 8,836 rides during April 2021!



	A	B	C	D
1	day	num_of_rides		
102	11/04/2021	268		
103	12/04/2021	273		
104	13/04/2021	279		
105	14/04/2021	284		
106	15/04/2021	289		
107	16/04/2021	295		
108	17/04/2021	301		
109	18/04/2021	306		
110	19/04/2021	312		
111	20/04/2021	318		
112	21/04/2021	324		
113	22/04/2021	330		
114	23/04/2021	336		
115	24/04/2021	342		
116	25/04/2021	348		
117	26/04/2021	354		
118	27/04/2021	360		
119	28/04/2021	366		
120	29/04/2021	373		
121	30/04/2021	379		
122	**The sum for April 2021:**	**8836**	start_month	num_of_rides
123			1	1491
124			2	2502
125			3	4290
126			4	**8836**
127				
128				
129				

□