**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

#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,

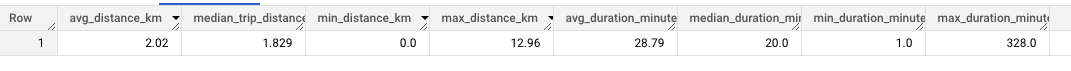
ROUND(AVG(duration\_in\_minutes),2) as avg\_duration\_minutes,

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;



#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;



#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

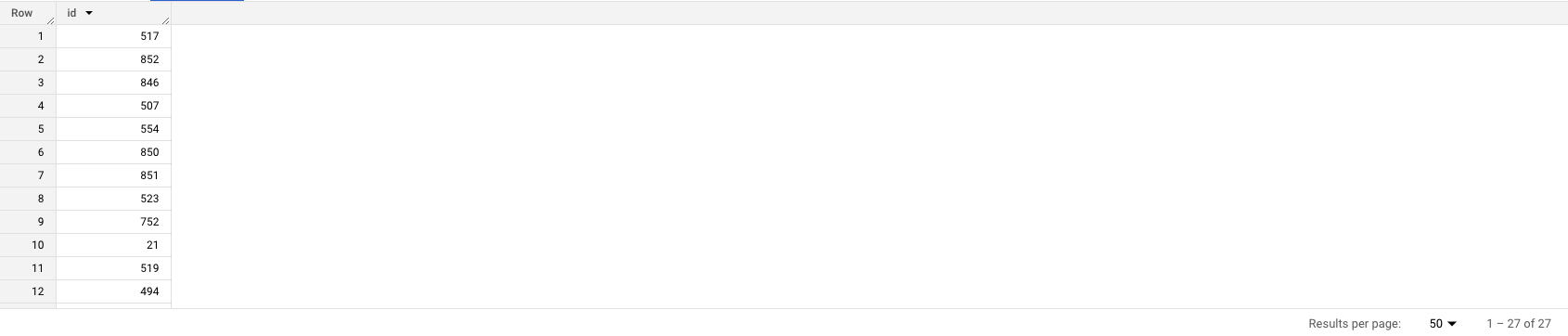
FROM

`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.



#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

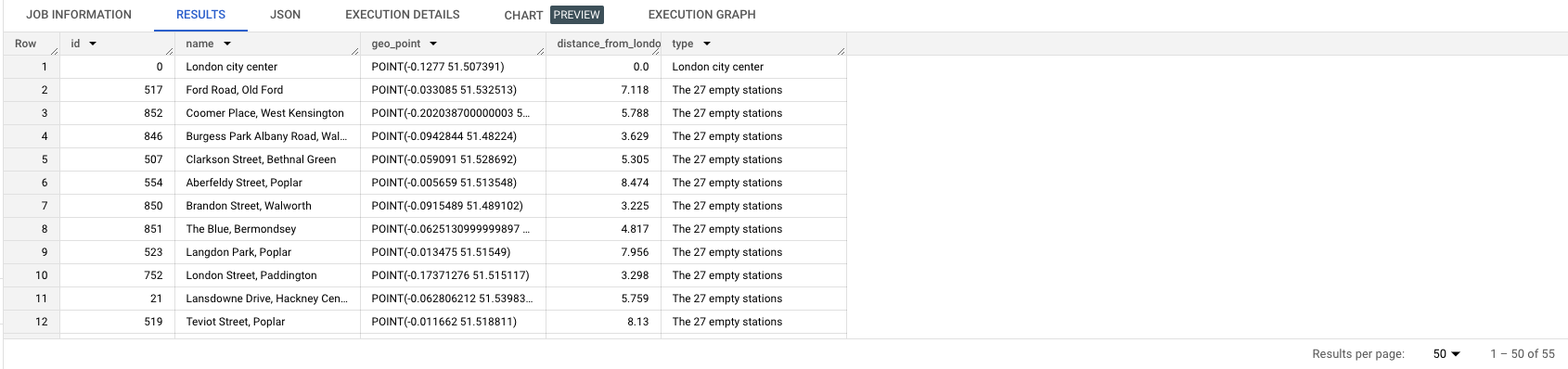
JOIN

`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";



#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

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

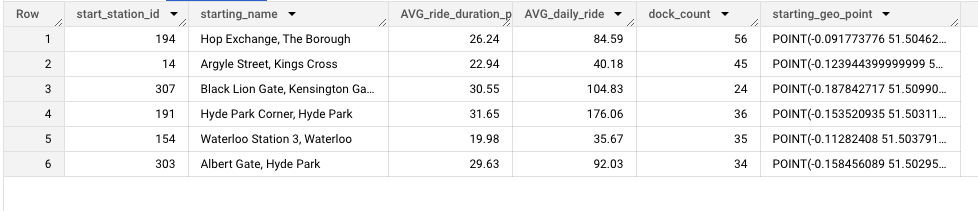
ORDER BY start\_station\_id,MONTH\_start,DAY\_start DESC)

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;



#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,

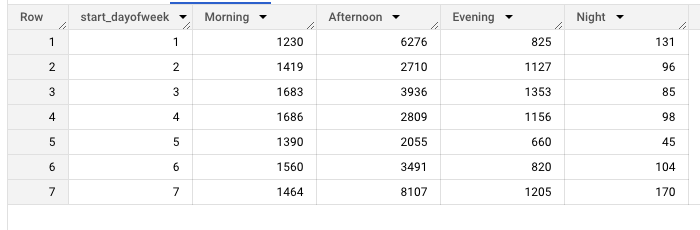
COUNT(CASE WHEN start\_hour+1 IN (23,0,1,2,3,4,5) THEN 1 END) AS Night

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;



#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,

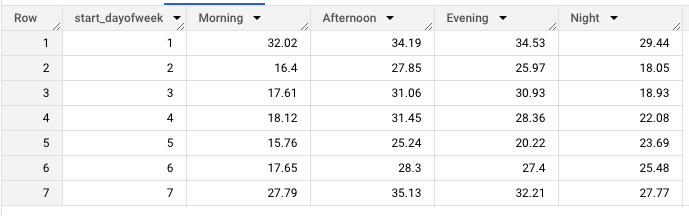
#Morning - 6 to 12 am, Afternoon - 1 to 6 pm, Evening - 7 to 10 pm, night - 11 pm to 5 am

FROM

table\_cleaned

GROUP BY start\_dayofweek

ORDER BY start\_dayofweek;



#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

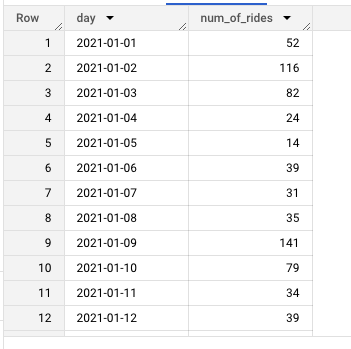
FROM

`data-analysis-389112.Project\_Google.cycle\_stations\_pro`

WHERE name LIKE '%Albert Gate, Hyde Park%')

GROUP BY day

ORDER BY day;



#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

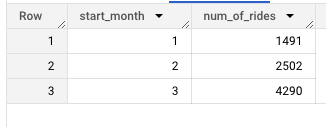
FROM

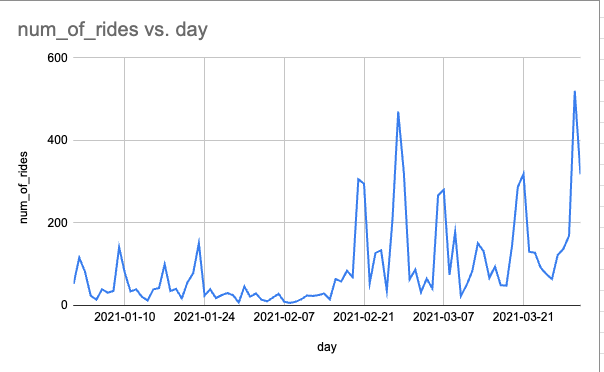
`data-analysis-389112.Project\_Google.cycle\_stations\_pro`

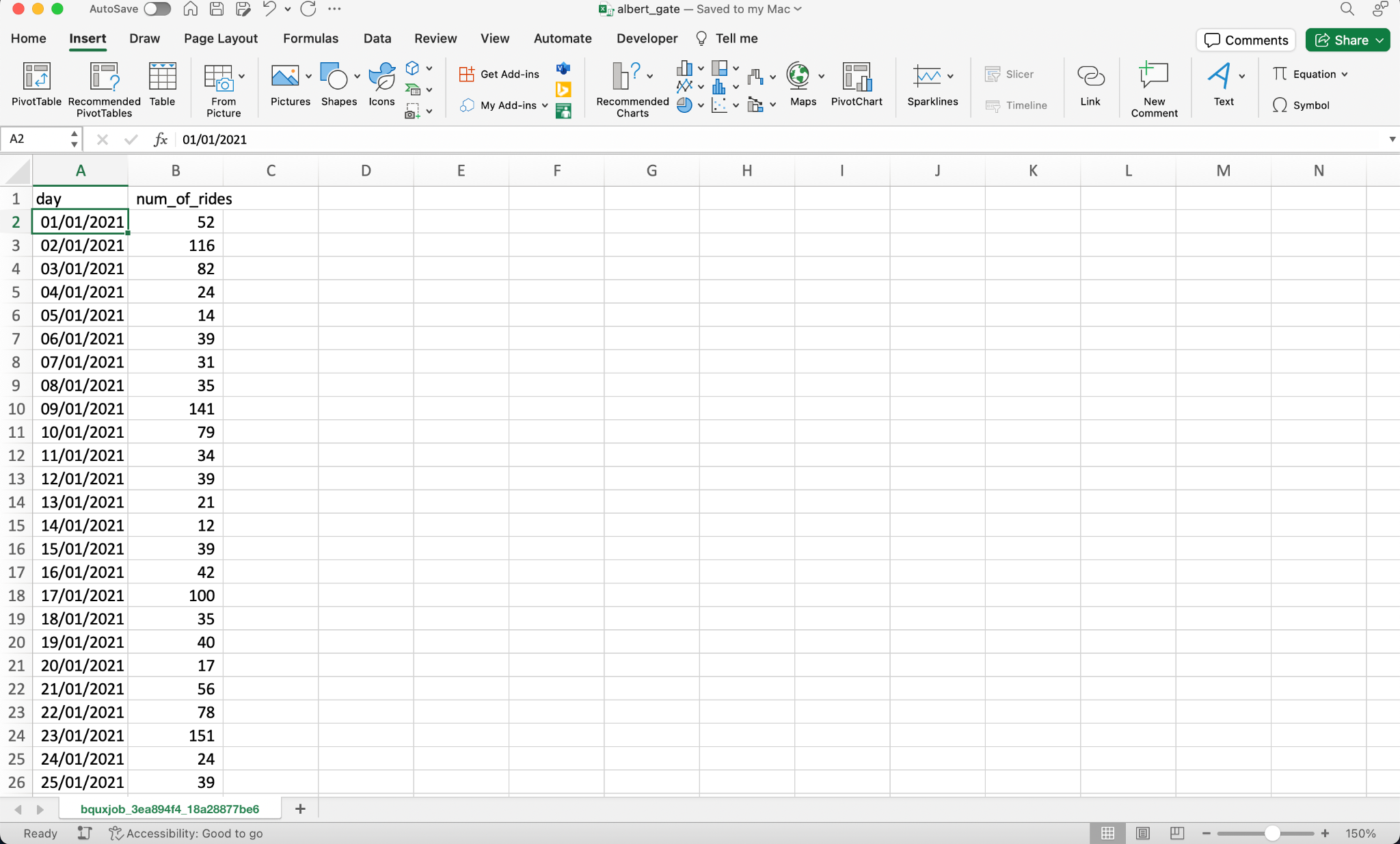
WHERE name LIKE '%Albert Gate, Hyde Park%')

GROUP BY start\_month

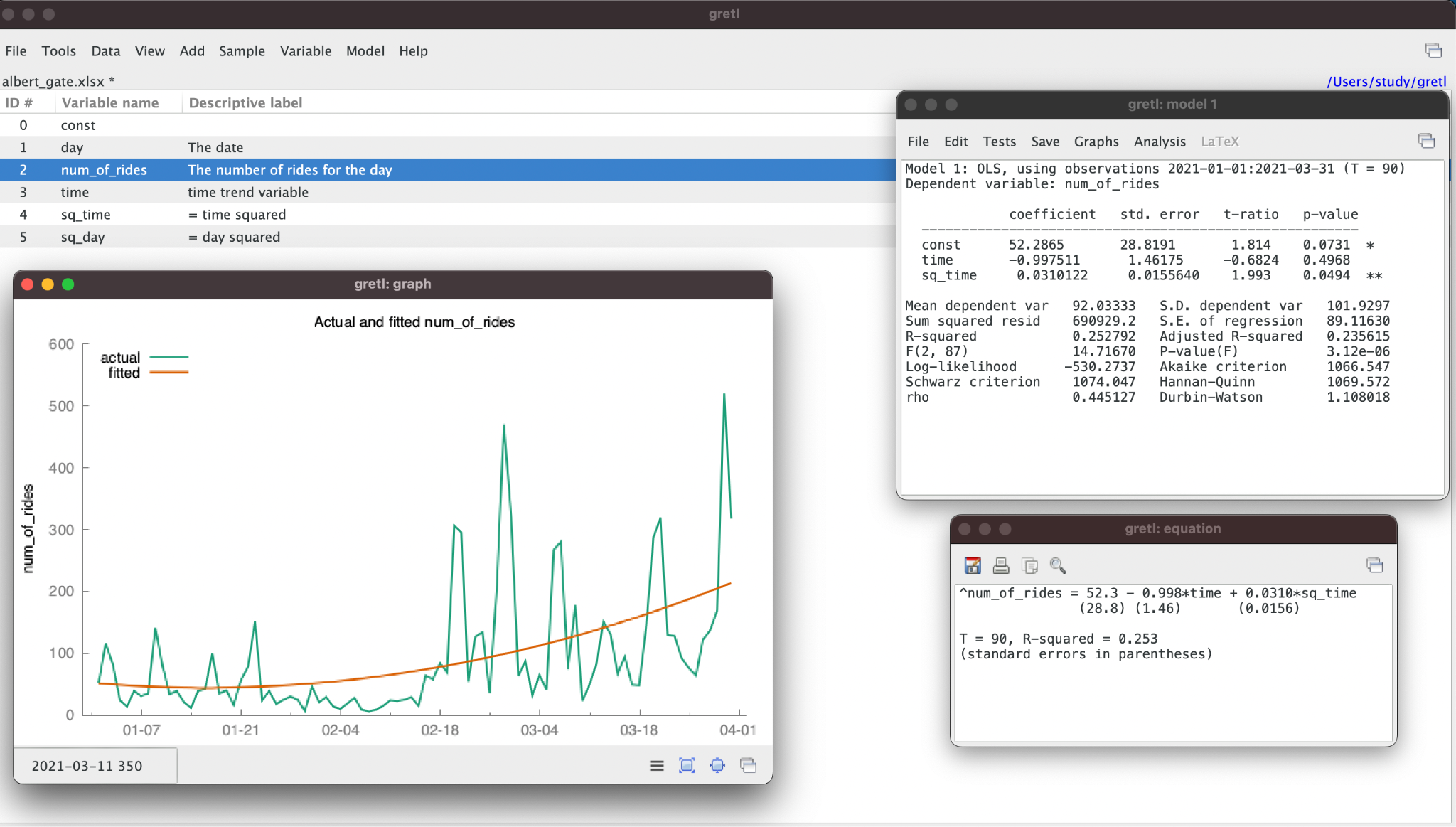
ORDER BY start\_month;



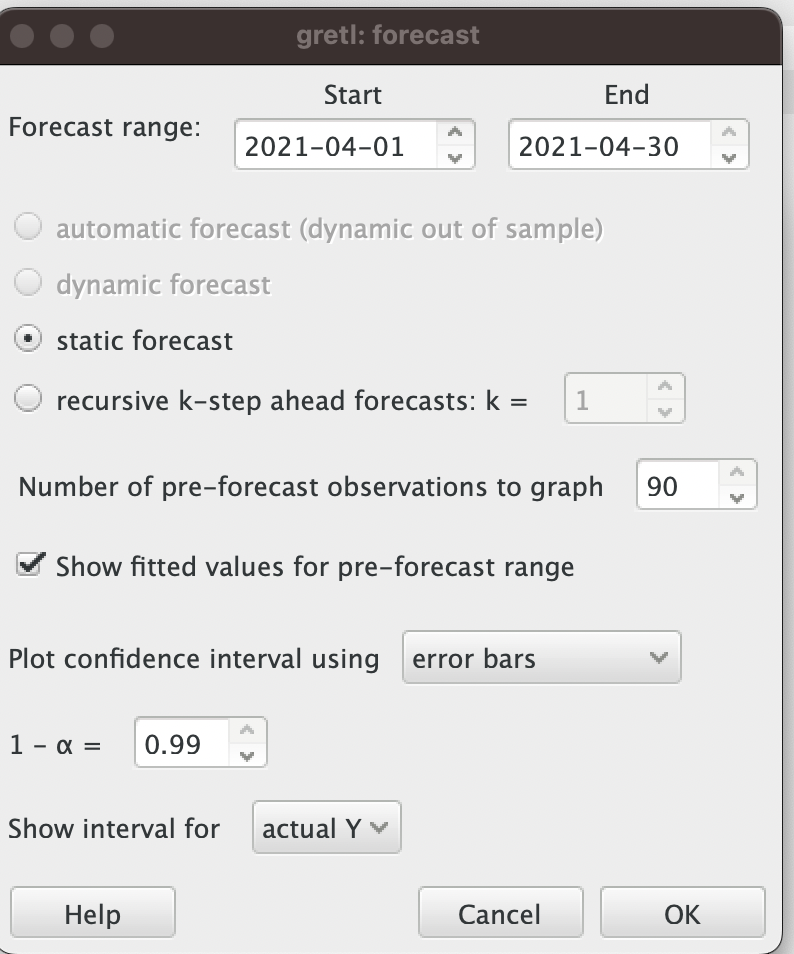


#We will move this table to sheets, and download is as an excel file so we could load it into gretl - a statistical package able to run a linear regression (ordinary least squares (OLS) model): 

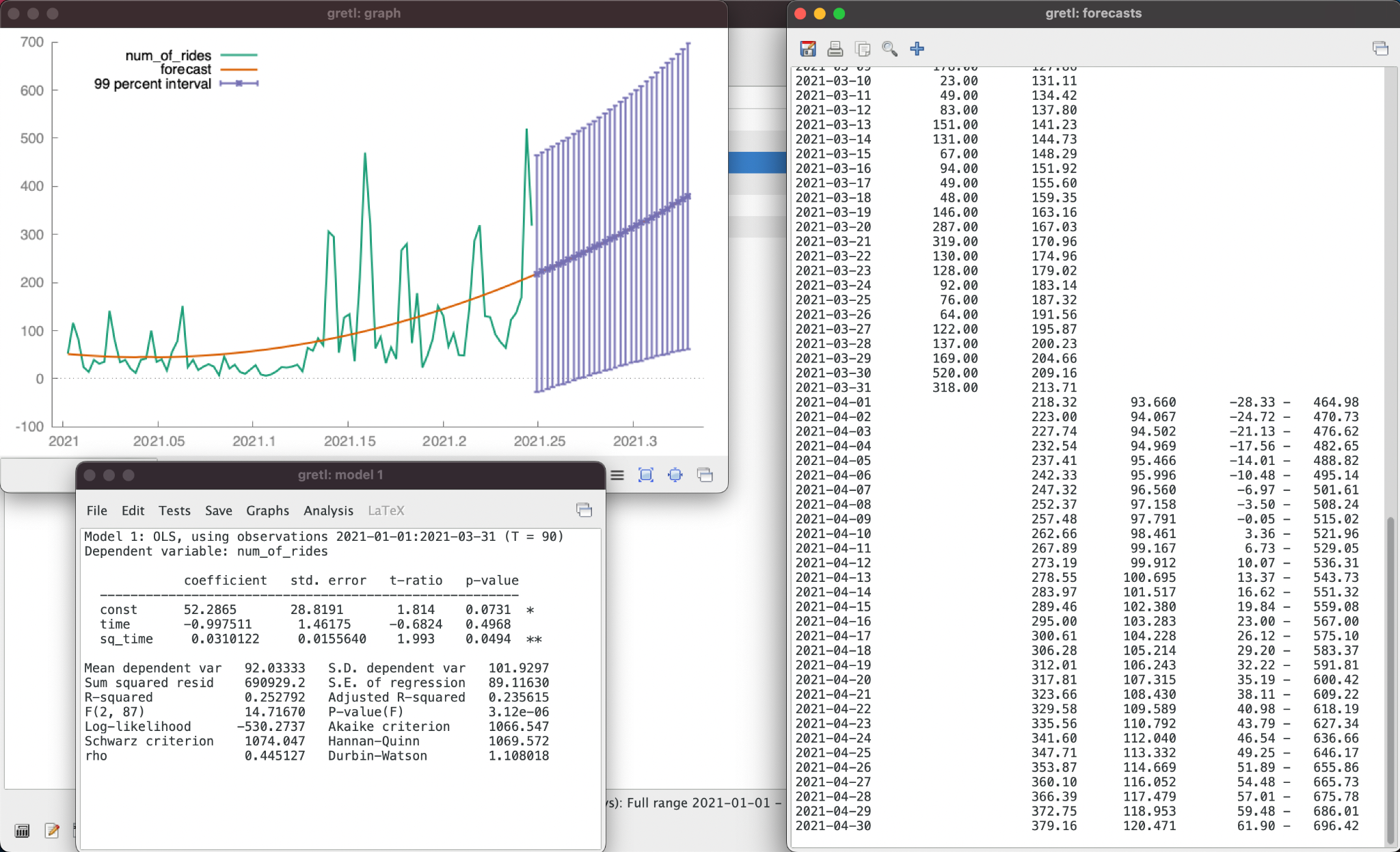
The equation: num\_of\_rides = α + β1 \* time + β2 \* time ^ 2



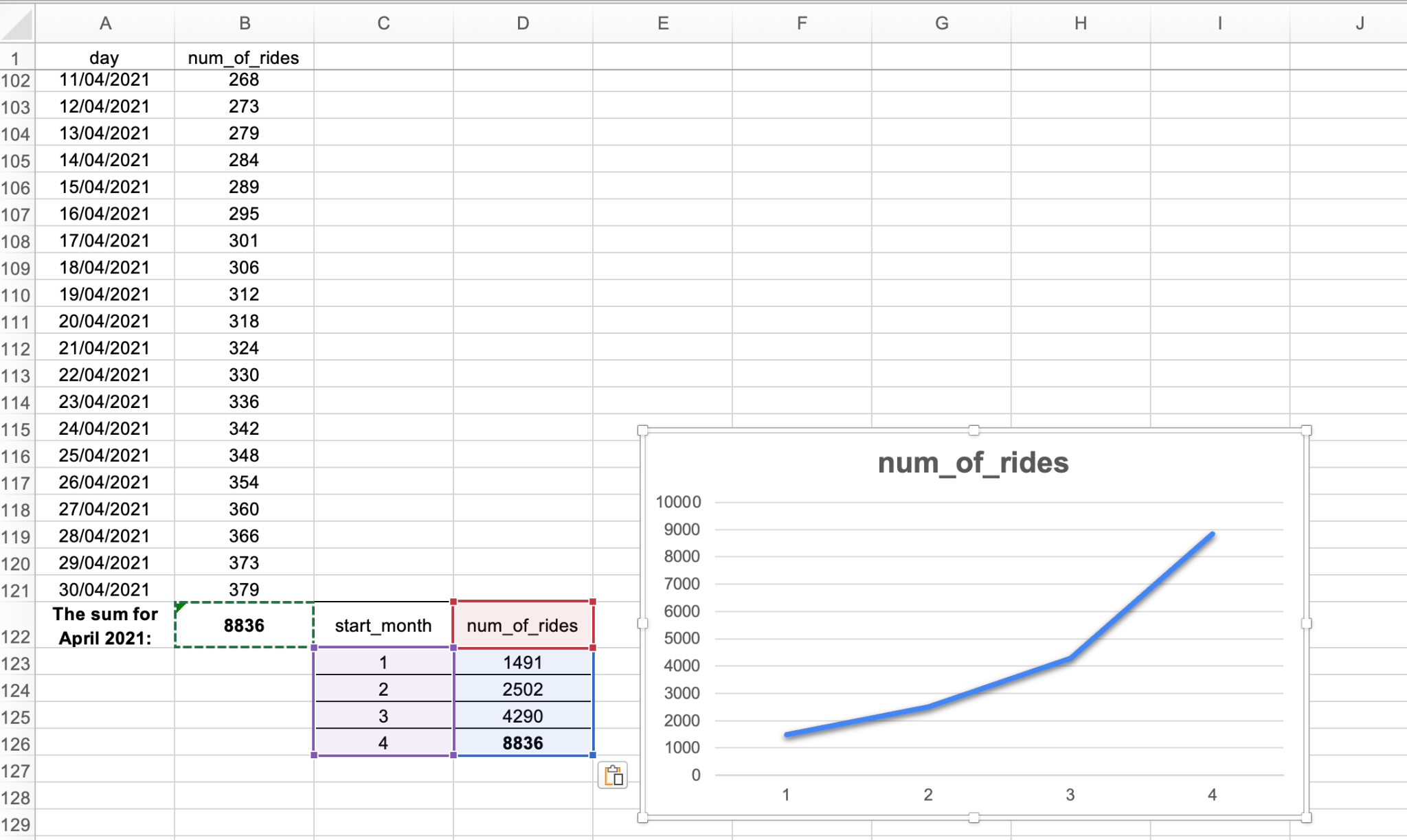
now, we use of model to predict - forecast the next 30 days: the month of April 2020:



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



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!



