

Zen City's Journey through London's bike rental data

Unset

--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_s
tation_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 + Statistics:
--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_mjr	min_duration_minute	max_duration_minuti
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_station	AVG_AIRlength_distance	total_CO2_saved_by_station	median_duration_minutes
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

Results per page: 50 1 - 27 of 27

--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";
```

JOB INFORMATION					
RESULTS					
JSON					
EXECUTION DETAILS					
CHART					
EXECUTION GRAPH					
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 S...	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
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

```



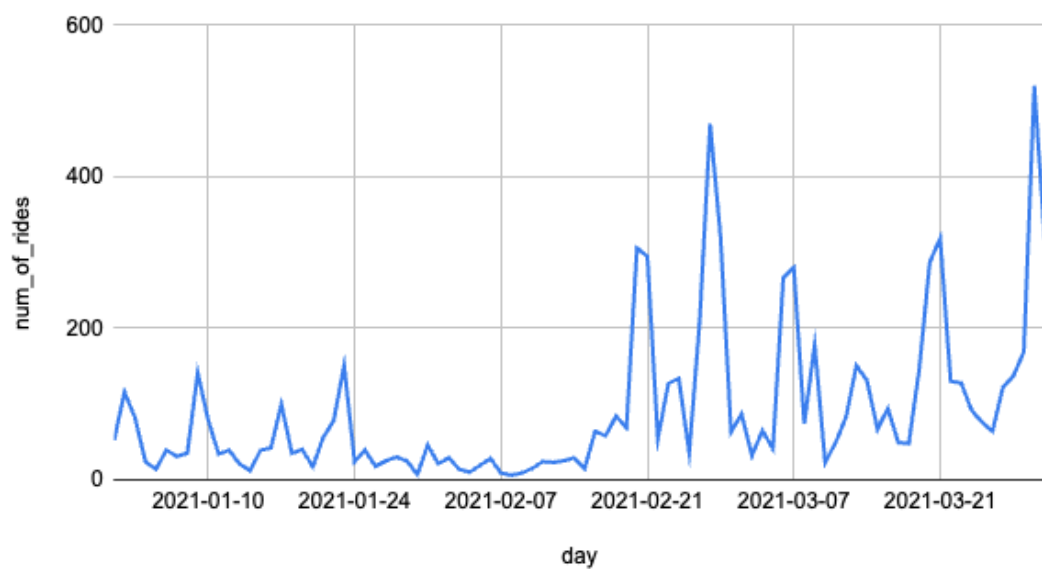
```

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;

```

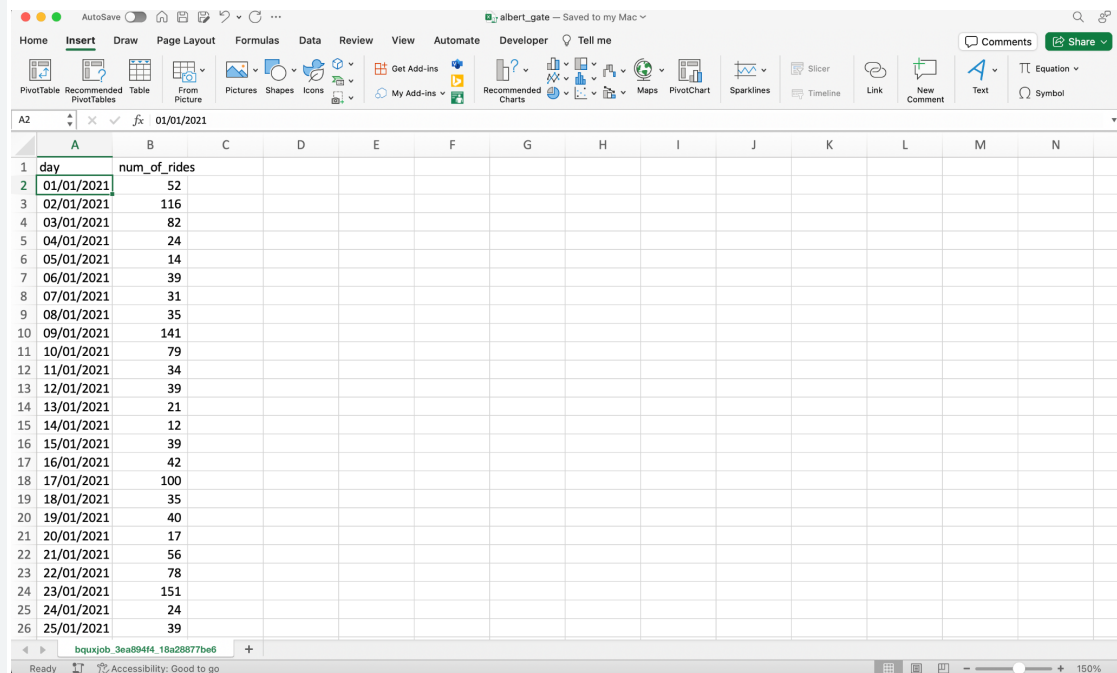
Row	start_month	num_of_rides
1	1	1491
2	2	2502
3	3	4290

num_of_rides vs. day

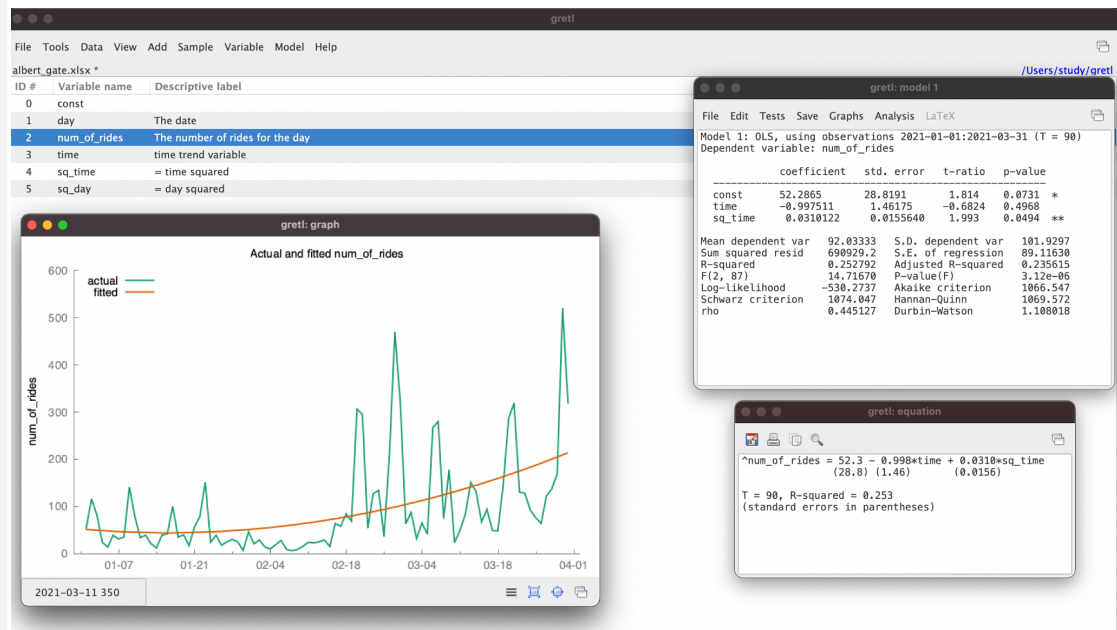


--We will move this table to sheets, and download it as an excel file so we could load it into gretl - a statistical package able to run a linear

regression (ordinary least squares (OLS) model):



The equation: $\text{num_of_rides} = \alpha + \beta_1 * \text{time} + \beta_2 * \text{time}^2$



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

gretl: forecast

Forecast range: Start End

2021-04-01 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

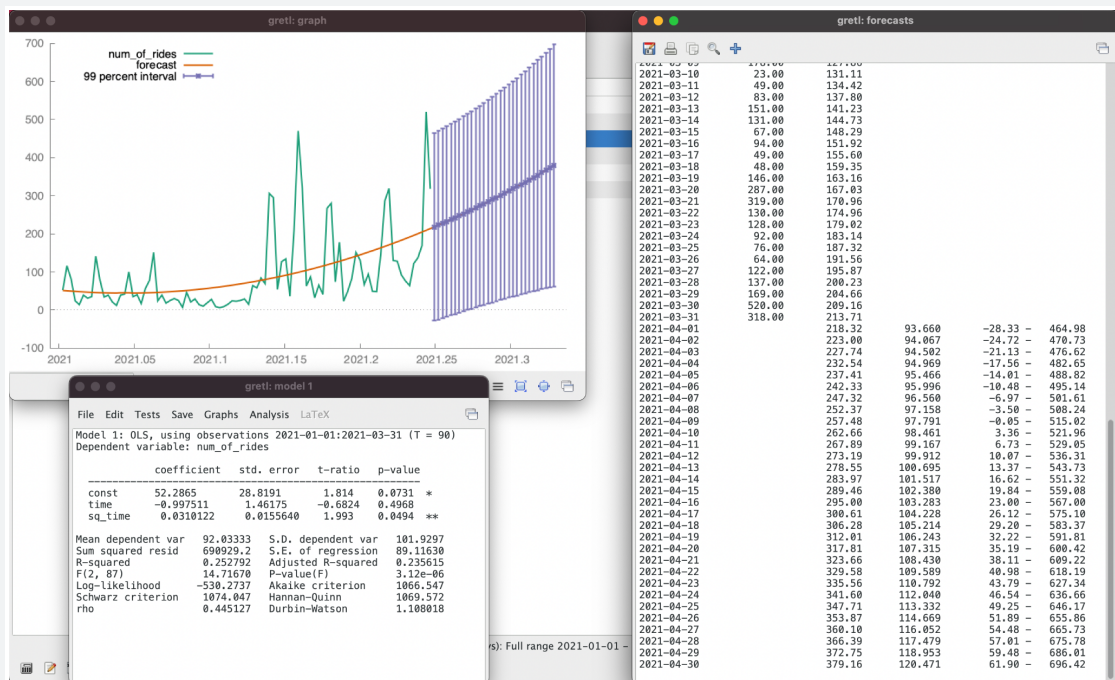
Plot confidence interval using error bars

1 - α = 0.99

Show interval for actual Y

Help Cancel OK

(with a significance level of $\alpha = 0.01 \rightarrow 1-\alpha = 99\%$)



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!

