Part 1 - BigQuery Questions

1- How many sessions are there?

211,904 sessions

Query used:

```
SELECT
        COUNT( DISTINCT visitId  ||  fullvisitorid )
FROM
        `dhh-analytics-hiringspace.GoogleAnalyticsSample.ga_sessions_export`;
```

- According to the docs combining visitId and fullvisitorid should get a unique ID , however if Distinct is removed then there can be multiple rows with this combination (max 2) and they are not duplicates
- The extra records are for sessions that moved from one day to another (started before 12 am and ended after it) and they are completion of the old sessions and should be merged into those rows

2- How many sessions does each visitor create?

- To get number of sessions created IN THE DATASET for each user by *fullvisitorid* :

```
SELECT
        COUNT(DISTINCT visitId),
        fullvisitorid
FROM
        `dhh-analytics-hiringspace.GoogleAnalyticsSample.ga_sessions_export`
GROUP BY 2
ORDER BY 1 DESC;
```

- To get number of sessions created in general by the user

```
SELECT
        MAX(visitNumber),
        fullvisitorid
FROM
        `dhh-analytics-hiringspace.GoogleAnalyticsSample.ga_sessions_export`
GROUP BY 2
ORDER BY 1 DESC;
```

- To switch between numbers in the dataset vs numbers in general we switch COUNT with MAX(visitNumber) in all the following queries
- To get average of the number of sessions : 2.127 = **2 sessions on average**
- 1 is min number of sessions and 80 is max

```
WITH sessions AS (
SELECT
        COUNT(DISTINCT visitId) AS cnt,
        fullvisitorid
FROM
        `dhh-analytics-hiringspace.GoogleAnalyticsSample.ga_sessions_export`
GROUP BY 2 ORDER BY 1 DESC
)
SELECT AVG(cnt), min(cnt), max(cnt) FROM sessions;
```

- percentiles, 1st quantile and median are (**1 session**)
- third quantile (75%) is **2 sessions**

Query:

```
WITH sessions AS (
SELECT
        COUNT(DISTINCT visitId) AS cnt, fullvisitorid
FROM
        `dhh-analytics-hiringspace.GoogleAnalyticsSample.ga_sessions_export`
GROUP BY 2 ORDER BY 1 DESC
),
quantiles as (
SELECT approx_quantiles(cnt,100) percentiles from sessions
)
SELECT
        percentiles[offset(25)] as p25,percentiles[offset(50)] as median,
        percentiles[offset(75)] as p75,
        percentiles[offset(100)] as max
FROM quantiles;
```

3. How much time does it take on average to reach the order_confirmation screen per session (in minutes)?

34.5 Minutes (seems too big)

Query used:

```
SELECT
avg(time/(60*1000))
FROM
`dhh-analytics-hiringspace.GoogleAnalyticsSample.ga_sessions_export`
,UNNEST (hit) AS h,
UNNEST (h.customDimensions) AS cd
WHERE
cd.value = 'order_confirmation'
AND cd.index = 11 ;
```

- this query's average calculation can possibly have a reduced average than the real one, because of those sessions split on two rows would have time start at 0 from the second row , so this would be problematic for sessions where order_confirmation is reached in second screen

4. By using the GoogleAnalyticsSample data and BackendDataSample tables,

analyse how often users tend to change their location in the beginning of their journey (screens like home and listing) versus in checkout and on order placement

Users tend to change their location in the earlier screens like shop_list screen (85k times), then home screen (75k times) then the next biggest number is in later screens like checkout (68k times)
and rest of screens have negligible numbers compared to those for geolocation.requested event

This event was chosen out of all location events because it was the only event having Latitude and Longitude data in each event firing so that we could compare the change
other events like ‘other_location.clicked’ and ‘Change Location’ have very little data for screens and no data for coordinates (and even so they confirm the conclusion of having it changed in earlier screens)

Queries used:

```
SELECT

landingScreenName,

h.eventAction AS eventName,

count(*)

FROM

`dhh-analytics-hiringspace.GoogleAnalyticsSample.ga_sessions_export`

,UNNEST (hit) AS h,

UNNEST (h.customDimensions) AS cd

WHERE

LOWER(h.eventAction) LIKE '%location%'

group by 1,2

order by 2 asc ,3 desc;
```

```
SELECT

CASE WHEN cd.index = 11 THEN  cd.value ELSE 'NOT SCREEN' END AS screen ,

CASE WHEN cd.index = 11 THEN 'SCREEN TYPE' WHEN cd.index = 18 THEN 'LONGITUDE'

  WHEN cd.index = 19 THEN 'LATITUDE' ELSE 'OTHER' END,

h.eventAction AS eventName,

count(*)

FROM

`dhh-analytics-hiringspace.GoogleAnalyticsSample.ga_sessions_export`

,UNNEST (hit) AS h,

UNNEST (h.customDimensions) AS cd

WHERE

LOWER(h.eventAction) LIKE '%location%'

and ( cd.index = 11 OR cd.index = 18 OR cd.index = 19)

group by 1,2,3

order by 3 asc ,4 desc;
```

4.2. and demonstrate the the deviation between earlier and later inputs (if any) in terms of coordinates change.

In terms of actual coordinates changes I have the exact opposite conclusion, checkout screen has 4552 location changes, then followed by shop_list and home screens having their combined numbers less than checkout location changes.

Which shows that the users only attempt to change their locations in earlier screens but more actual changes to location happen at checkout !

The SQL query is too big so it will be attached in 4.location_change_event_counts.sql file

4.3. Then, using the BackendDataSample table, see if those customers who changed their address ended placing orders and if those orders were delivered successfully,

Yes all customers who changed their locations ended up placing orders and all of them were delivered.

4.4. if so, did they match their destination ?

None of the delivery coordinates did exactly match the customer coordinates which makes sense
But to measure if they matched or not we must put a sensible margin of error
One latitude degree equals about 69 miles which equals about 111 kilometers, so if we make our margin of error equals 0.0003 latitude and longitude degrees in difference from the customers' set location so we allow 300 square meters difference then almost half of all deliveries match their destination
And if we allowed 500 square meters then only about 30% of deliveries do not match while the remaining 70% matches
Finally if the threshold was decreased to 100 meters then most of the deliveries will not match destinations set by users (about 90%)

So the answer to this question depends on the acceptable margin of error in coordinates.

Part 2 - Python Questions

All questions are answered in the iPython notebook attached (DHH.ipynb)

To view and run directly in Colab: link

Extra Tableau Visualization