Question 1 - SQL

Part A

You have a table populated with trip information (named Company_trip) table with a rider_id (unique per rider), trip_id (unique per trip), trip_timestamp_utc (the UTC timestamp for when the trip began), and trip_status, which can either be ‘completed’ or ‘not completed’.

```
rider_id , trip_id, begintrip_timestamp_utc, trip_status
```

Write a query to return the trip_id for the 5th completed trip for each rider. If a rider has completed fewer than five trips, then don’t include them in the results.

```
SELECT
        RIDER_ID
        , TRIP_ID

FROM Company_TRIP

WHERE TRIP_STATUS = 'completed'

QUALIFY DENSE_RANK() OVER(PARTITION BY RIDER_ID ORDER BY BEGINTRIP_TIMESTAMP_UTC ASC) = 5
```

Part B

You are given three separate tables (named trip_initiated, trip_cancel, and trip_complete) of the form:

```
trip_initiated | trip_id, rider_id, driver_id, timestamp
trip_cancel | trip_id, rider_id, driver_id, timestamp
trip_complete | trip_id, rider_id, driver_id, timestamp
```

Each trip_id in these tables will be unique and only appear once, and a trip will only ever result in a single cancel event or it will be completed. Write a query to create a single table with one row per trip event sequence (trip initiated →cancel/complete):

```
dispatch_events | trip_id, rider_id, driver_id, initiated_ts, cancel_ts, complete_ts
```

There should only be a single row per trip with a unique trip_id.

```
CREATE TABLE DISPATCH_EVENTS AS
(
SELECT
        init.TRIP_ID
        , init.RIDER_ID
        , init.DRIVER_ID
        , init.TIMESTAMP_ as INITIATED_TS –'TIMESTAMP' is a reserved word
        , complete.TIMESTAMP_ as COMPLETE_TS
        , cancel.TIMESTAMP_ as CANCEL_TS

FROM TRIP_INITIATED init

LEFT JOIN TRIP_COMPLETE
        ON init.TRIP_ID = complete.TRIP_ID

LEFT JOIN TRIP_CANCEL
        ON init.TRIP_ID = cancel.TRIP_ID
) WITH DATA PRIMARY INDEX (TRIP_ID, RIDER_ID, DRIVER_ID)
```

Part C

Write at least one test query to validate the data in the table you created in Part B. Indicate what you would expect the query to return if the data were valid.

```
– This query should return blank
SELECT
        CANCEL_TS

FROM DISPATCH_EVENTS

WHERE TRIP_ID = '16';


– This query should return '2018-02-17 14:00:00'
SELECT
        COMPLETE_TS

FROM DISPATCH_EVENTS

WHERE TRIP_ID = '16'
```

Question 2 - Experimental Design

Part A

The Driver team is planning to test a new incentive structure in which they will offer drivers an extra $5 per hour if they choose to drive during times of peak demand (4pm - 8pm) to increase the available supply. Drivers will be required to complete at least 5 trips during this window to qualify for the new incentive. As the data scientist on the team:

1) Propose and define the primary success metric of this test. In addition, propose and define 2 or 3 tracking metrics that will be important to monitor in addition to the success metric you have defined.
2) Outline an experimentation plan to evaluate the effect of this incentive, according to the metrics you outlined.
    a) What would be the rollout schedule, and how would you balance this with statistical rigor?
    b) What type of data analysis would you perform? Please explain why you chose that method over possible alternatives.

1. Primary Success Metric: average number of trips an active driver makes during peak demand.

The goal of this metric is to see if drivers that already drive at least one trip during peak demand tend to drive more trips during that time with the incentive in place. An increase in this metric would mean a driver that normally drives X trips during peak demand now drives more than X trips.

Tracking Metric: percent of active drivers that make >= 5 trips during peak demand.

Since the incentive is paid out with at least 5 trips, I'd like to track what proportion of the driver supply meets that threshold. If very few meet the threshold, it may be set too high, and if too many meet the threshold it could get very expensive.

Tracking Metric: number of new active drivers for the peak demand period

This metric would count all drivers that previously did not drive during the peak demand period but now drive during the peak demand period. The idea for this metric is to see if the incentive brings in new drivers and/or shifts existing driver's behaviors into the peak demand period.

2. This type of experiment compares drivers that got the incentive and drivers that did not.

- Define the hypothesis: Providing a $5 per hour incentive will increase the number of trips a driver makes during peak demand hours.
- Identify good locations to test the incentive. A good location would be one with more demand than supply, so increasing the supply of drivers would be beneficial. It would also need sufficient overall demand and supply, for example, I wouldn't want to pick a location with only a few riders and drivers.
- Generate a test and control group. The test group would be given the incentive and the control group would not. The sample size in each group would need to be big enough to create good statistics. I can use statistical power calculators to determine the sample size needed to achieve statistical significance levels.
- Decide how long to run the experiment. This would depend somewhat on how many drivers are needed in the two groups, but I would want to make sure the experiment runs for at least a full week, since I expect there might be differences in drivers on different days of the week.
- Schedule the roll-out. I would want at least 3 separate areas to give multiple data points and see if there are any differences in the results based on geographic location. I would also want to repeat the experiment at the same location more than once, because sometimes with these types of incentives there is a “novelty effect” where it really only works the first time. I would also want to schedule the roll-outs starting at different times of the month, because I expect there may be more of a difference at the end of the month before people have rent and bills due soon.
- Analyze the data. I would calculate the primary success metric to find the averages for the test groups and control groups for all experiments in the roll-out and count the sample sizes. Using the resulting data I can use ANOVA analysis to test of all the averages are the same. The resulting analysis will give me the probability that the test and control groups have the same average (p-value), as well as p-values for the effects of geographic location and starting time of month. If the test and control p-value is low, it gives support to my hypothesis for the experiment. If the p-values for geographic location and starting time of month are low, then the effect of this incentive varies depending on the value of these variables.

Part B

A marketing team is planning a campaign to attract new riders to Company in which they will put billboards up across a city, and they'll be up for several weeks. As the data scientist on the team, what metrics would you be interested in when analyzing the impact of this campaign and how would you go about quantifying any effect on these?

Interesting Metrics:

- Number of new app downloads in the city
- Number of first rides taken in the city
- Number of Google searches for "Company" in the city

Quantifying the Effects:

It's not possible to have a true test and control group in this case, because we can't show the billboards to some people and not show the billboards to other people. Because of this, it would be better to create a counterfactual to compare the metrics against. This can be done by using time series analysis of these metrics over time before the billboards were put up to forecast an expected value of the metrics after the billboards were put up. The difference between the true metric values and the counterfactual is the effect of the billboards.

Question 3 - Modeling

Company's Driver team is interested in predicting which driver signups are most likely to start driving. To help explore this question, we have provided a sample dataset of a cohort of driver signups. The data was pulled a some time after they signed up to include the result of whether they actually completed their first trip. It also includes several pieces of background information gathered about the driver and their car.

We would like you to use this data set to help understand what factors are best at predicting whether a signup will start to drive within 30 days of signing up, and offer suggestions to operationalize those insights to help Company.

See below for a description of the dataset. Please include any code you wrote for the analysis and delete the dataset when you have finished with the challenge. Please also call out any data related assumptions or issues that you encounter.

Data set description

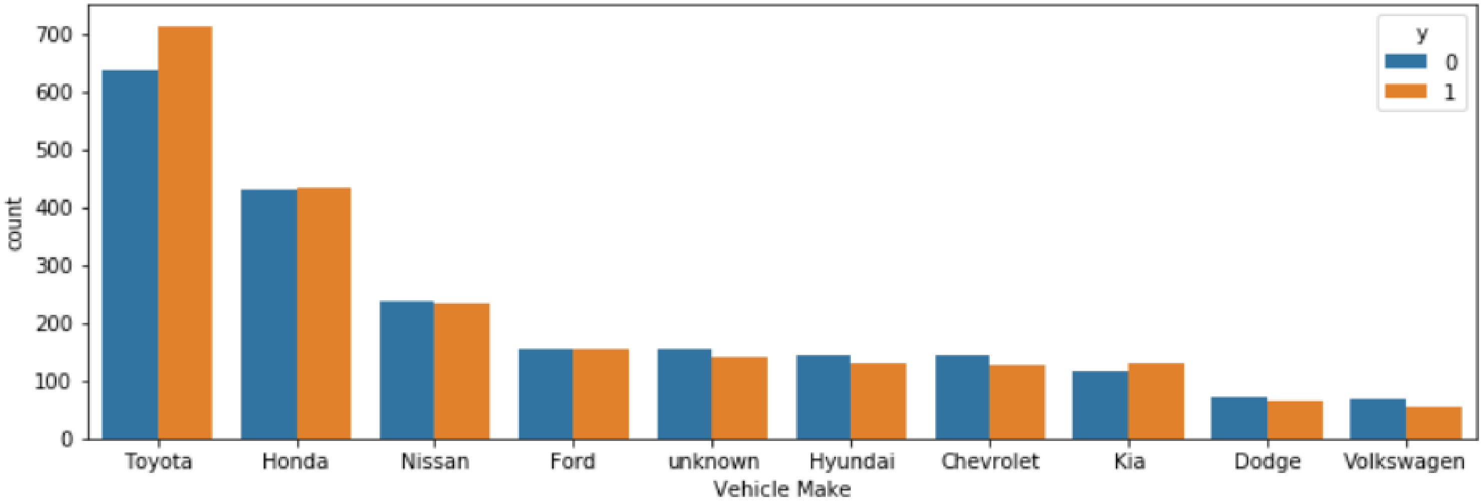
id: driver_id
city_name*: city that this user signed up in*
signup_os*: signup device of the user*
signup_channel*: what channel did the driver sign up from*
signup_timestamp*: timestamp of account creation*
bgc_date*: timestamp when driver consented to background check*
vehicle_added_date*: timestamp when driver's vehicle information was uploaded*
vehicle_make: *make of vehicle uploaded*
vehicle_model: *model of vehicle uploaded*
vehicle_year: *year that the car was made*
first_trip_date*: timestamp of the first trip as a driver*

Part A

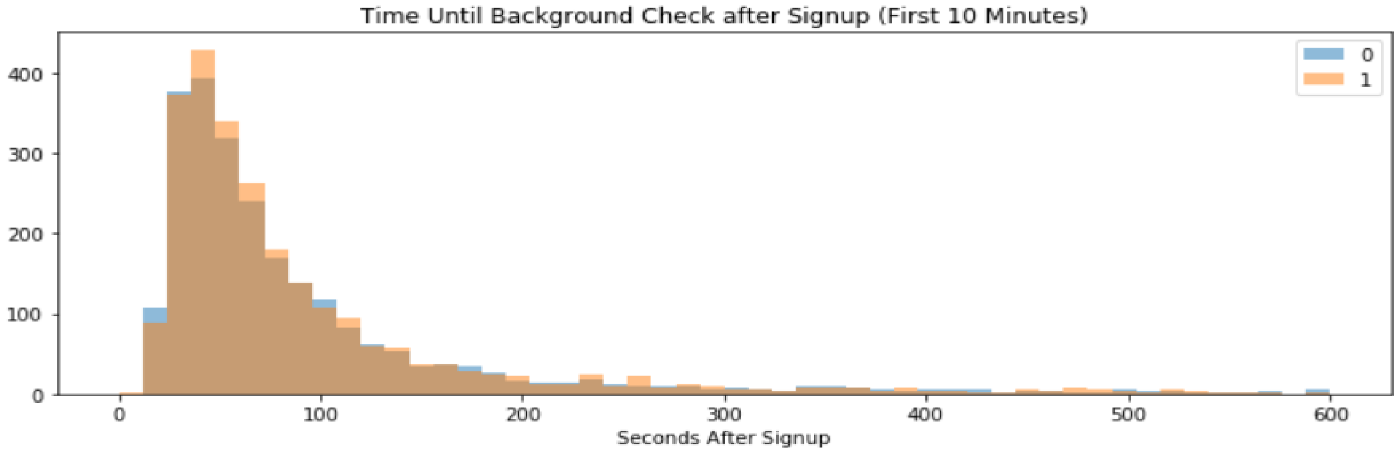
Perform any cleaning, exploratory analysis, and/or visualizations to use the provided data for this analysis (a few sentences/plots describing your approach will suffice). What fraction of the driver signups took a first trip within 30 days of signing up?

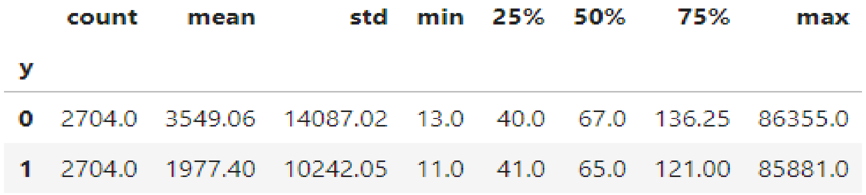
My approach to cleaning was to remove any records that appear to not be serious signups. I assumed Company needs to have the driver get a background check and input their vehicle information before allowing them to drive, so I removed all records that had an incomplete signup. After cleaning the data set, the fraction of driver signups that took a first trip within 30 days is 56%. I did some sampling to down select the data to get a balanced 50-50 split for my modeling data set.

I explored some features that might be predictive, such as vehicle make and time between signup and background check.



More Toyota cars appear in the successful first trip dataset, so this data may be useful to predict successful first trips.





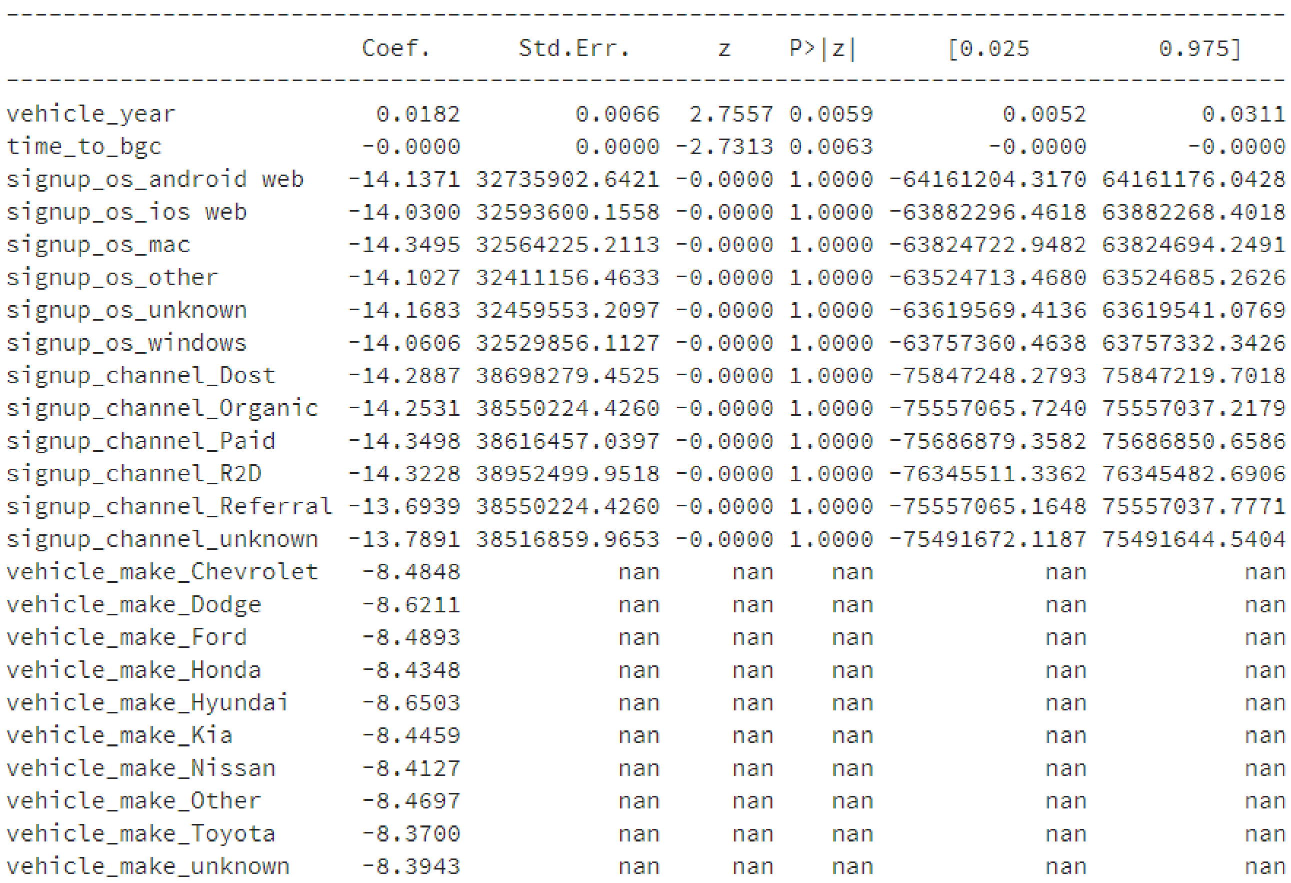
	count	mean	std	min	25%	50%	75%	max
y								
0	2704.0	3549.06	14087.02	13.0	40.0	67.0	136.25	86355.0
1	2704.0	1977.40	10242.05	11.0	41.0	65.0	121.00	85881.0

For successful and non-successful first trips, they both appear very similar for time until background check. However, there are more large outliers in the non-successful first trip groups, shown by the larger mean for y=0 than y=1.

Part B

Build a predictive model to help Company determine whether or not a driver signup will start driving within 30 days of signing up. Discuss why you chose your approach, what alternatives you considered, and any concerns you have. How valid is your model? Include any key indicators of model performance.

Since my response data is binary, I need to use a modeling approach for classification. I will use logistic regression for this model, because it provides estimates for each variable that can be interpreted and used by the business to target more first trips. There are other alternatives for classification problems, such as Random Forest, Support Vector Machines, and Neural Networks. These alternatives may perform better at classification; however, they are more difficult models to interpret.

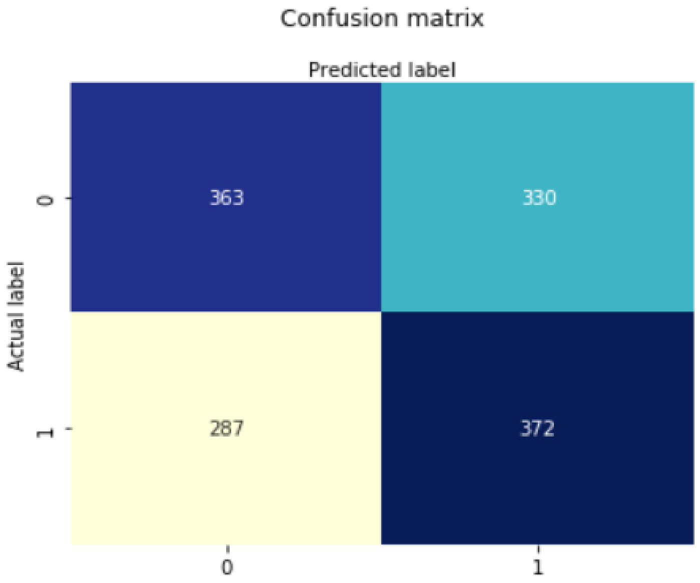


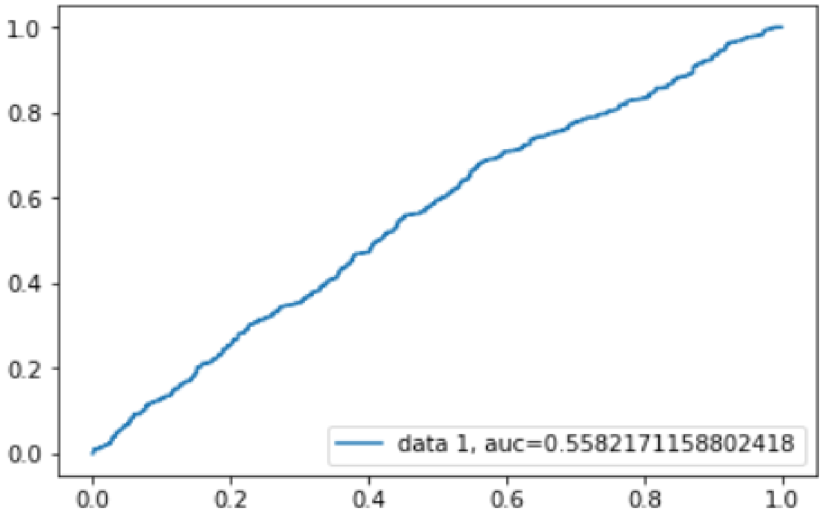
```
-----------------------------------------------------------------------------------------
                         Coef.        Std.Err.       z     P>|z|       [0.025        0.975]
-----------------------------------------------------------------------------------------
vehicle_year              0.0182        0.0066  2.7557 0.0059         0.0052         0.0311
time_to_bgc              -0.0000        0.0000 -2.7313 0.0063        -0.0000        -0.0000
signup_os_android web   -14.1371 32735902.6421 -0.0000 1.0000 -64161204.3170 64161176.0428
signup_os_ios web       -14.0300 32593600.1558 -0.0000 1.0000 -63882296.4618 63882268.4018
signup_os_mac           -14.3495 32564225.2113 -0.0000 1.0000 -63824722.9482 63824694.2491
signup_os_other         -14.1027 32411156.4633 -0.0000 1.0000 -63524713.4680 63524685.2626
signup_os_unknown       -14.1683 32459553.2097 -0.0000 1.0000 -63619569.4136 63619541.0769
signup_os_windows       -14.0606 32529856.1127 -0.0000 1.0000 -63757360.4638 63757332.3426
signup_channel_Dost     -14.2887 38698279.4525 -0.0000 1.0000 -75847248.2793 75847219.7018
signup_channel_Organic  -14.2531 38550224.4260 -0.0000 1.0000 -75557065.7240 75557037.2179
signup_channel_Paid     -14.3498 38616457.0397 -0.0000 1.0000 -75686879.3582 75686850.6586
signup_channel_R2D      -14.3228 38952499.9518 -0.0000 1.0000 -76345511.3362 76345482.6906
signup_channel_Referral -13.6939 38550224.4260 -0.0000 1.0000 -75557065.1648 75557037.7771
signup_channel_unknown  -13.7891 38516859.9653 -0.0000 1.0000 -75491672.1187 75491644.5404
vehicle_make_Chevrolet   -8.4848           nan     nan    nan            nan            nan
vehicle_make_Dodge       -8.6211           nan     nan    nan            nan            nan
vehicle_make_Ford        -8.4893           nan     nan    nan            nan            nan
vehicle_make_Honda       -8.4348           nan     nan    nan            nan            nan
vehicle_make_Hyundai     -8.6503           nan     nan    nan            nan            nan
vehicle_make_Kia         -8.4459           nan     nan    nan            nan            nan
vehicle_make_Nissan      -8.4127           nan     nan    nan            nan            nan
vehicle_make_Other       -8.4697           nan     nan    nan            nan            nan
vehicle_make_Toyota      -8.3700           nan     nan    nan            nan            nan
vehicle_make_unknown     -8.3943           nan     nan    nan            nan            nan
```

My first model iteration included variables: vehicle_year, time_to_bgc, and dummy variables for signup_os, signup_channel, and vehicle_make. To create dummy variables, I create a new column for each level of the categorical variable filled with 0's and 1's to represent that row having that level.

The model did not perform well with a pseudo R-squared of only 0.023 (1 is the best model possible). This model is barely better than a random guess. The vehicle_year is significant; however, the coefficient is very low, saying that for each increase in year, the odds of completing a first trip only improve by 0.018 (with everything else constant). I have serious concerns using this model to predict the likelihood of completing a first trip, because a random number generator would do nearly just as good.

I created a confusion matrix and ROC curve as further performance metrics:





AUC: 0.558

Accuracy: 0.544 (True Positives + True Negatives / All Values)

Precision: 0.530 (True Positives / Total Predicted Positives)

Recall: 0.564 (True Positives / Total Actual Positives)

The ROC curve is nearly diagonal with an AUC of 0.558, which is not much better than random guessing, the best would be a curve near the top left corner.

Part C

Briefly discuss how Company might leverage the insights gained from the model to generate more first trips (again, a few ideas/sentences will suffice).

Based off this model, I would not suggest Company leverage the results of this particular model. However, if the coefficient for vehicle_year was larger, Company could leverage this insight to target new car buyers, or buyers of used cars that are only a few years old. Also, if the coefficient for having people approve a background check was larger, Company could use this info to send a push notification to the phone of the person that signed up.