Team: Data Science, Economics and Research
Date: 08/22/2018

SQL Test:
Q1) Think about ways in which cross-dispatch might make querying Uber's data complicated. Write a paragraph or two about common types of analysis that might fail if the researcher didn't think carefully enough about cross-dispatch.

A cross-dispatch might make querying Uber's data complicated because you cannot directly aggregate number of driver vehicles across products. For example, to get a total count of active UberX drivers for a given day in New York City, you would have to account for the regular UberX driver vehicles plus the UberXL/UberSUV/UberBlack driver vehicles who accepted UberX rides for that day in New York City as well. If this complication is not accounted for, you could have an incorrect aggregate number of driver vehicles across Uber products.

Cross-dispatch practices can also complicate or fail an analysis if the researcher does not account for them. That is because riders could have a wrong impression of product experiences, and if that is not accounted for then the results can have misleading interpretations. These misleading interpretations are driven by cross-dispatch factors not controlled for in the analysis, which often leads to biases in the research.

For example, suppose a client selects an UberX vehicle for a night out with his wife in downtown New York City and instead an UberXL vehicle arrives without the clients knowing. Both clients are enjoying the extra space that comes with the UberXL vehicle. In fact, the clients enjoyed that extra room so much that because of it, they gave the driver a 5 star rating. Say this cross-dispatch event is not a one-time event and happens frequently with other clients as well. Assume now that an Uber researcher wants to evaluate client ratings by product without controlling for cross-dispatch events. The effect of each product on rider ratings could be off because the ratings are partially due to the cross-dispatch event (client selects a UberX vehicle but receives a UberXL vehicle) that is left out of the analysis.

Q2) Please write a query that answers the following: For each driver in New York City, report the number of trips completed by week for the weeks of March, 2014.

```
SELECT
   tt.drive_id,
   count(tt.drive_id) AS n_trips,
   DATEPART(week, tt.request_at) AS week
FROM trips tt
INNER JOIN cities cc ON cc.id = tt.city_id
WHERE 1=1
   AND tt.request_at >= '2014_03_01'
   AND tt.request_at < '2014_04_01'
   AND cc.name = 'new_york'
   AND tt.status = 'completed
GROUP BY tt.driver_id, week;
```

Q3) Comparing the results of this query to counts of drivers and supply hours in another tool generally considered reliable, we find that this query's results for these quantities are inflated by some factor. Why? Are *aggregate_fares* and *avg_fares_per_hour* also wrong? In what direction?

One of the columns for this query is inflated because the query, *driver_fares*, has a one-to-many relationship. Since the *driver_fares* query has two columns in its *GROUP BY* function, *driver_id* and *vehicle_view_name*, a unique *driver_id* can have more than one row in the dataset. That is because there can be multiple unique products the *driver_id* can have transactions for (e.g. UberX, UberXL, UberBLACK). Therefore, the following piece of code in the SQL query "*COUNT(driver_id) AS n_drivers*" could be counting *driver_id* more than once since it is not counting each *driver_id* uniquely. The direction of the count of inflated *n_drivers* will have an upward inflated factor.

Both *aggregate_fares* and *avg_fares_per_hour* are not inflated. The reason why *aggregate_fares* is not inflated by some factor is because the query "*SUM(total_fares) AS aggregate_fares*" is a query of the summation for all *total_fares* transactions. Regardless of the *GROUP BY driver_id* and *vehicle_view_name*, the *total_fares* column is a unique transaction for each observation. For example, suppose a *driver_id* 1111 exists and has two driver vehicle transactions, UberX and UberXL. Because of the GROUP BY function on *driver_id* and *vehicle_view_name*, *driver_id* 1111 will appear twice (UberX and UberXL), but each driver vehicle transaction will have its own fare. Thus, there are no duplicate transactions of fares occurring when summing *total_fares*.

Similarly, *avg_fares_per_hour* is also not inflated. Since the aggregation is occurring within the *driver_times* query and the *GROUP BY* function is only on one column, *driver_id*, the relationship is one-to-one. Therefore, getting the *avg_fares_per_hour* will be calculated using a unique *total_fares* amount and a unique *hours_on_shift* amount.

Analysis Test:

Q1) Using your analysis tool of choice (R), generate a graph showing an hourly breakdown of client login behavior.

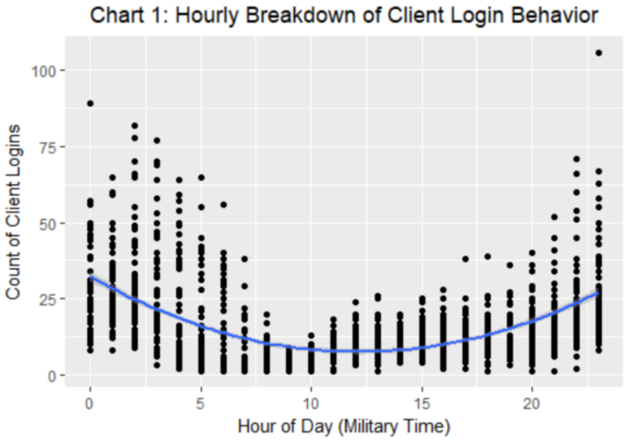


Chart 1 is a distribution of count of client logins by hour of day throughout all weeks and weekdays.

Since the raw format of the data is in a data-time format (e.g. 2012-03-01 00:05:55) — I extracted month, week, day, and hour date components. Subsequently, I graphed count of client logins by hour versus averaging count of hourly client logins within all weeks and days across the data points. This method exposed the hourly distribution across the weeks and weekdays. Noticeably, there are hours with high variability of client logins (0 through 5 AM, midnight to early morning) and hours with low variability (8 AM to 8 PM, morning to evening). Also, it appears that two outliers exist in the dataset. Both outliers are occurring around midnight.

Q2) Add a best fit line or curve to this graph and include any relevant metrics/statistics to quantify the quality of the fit.

To determine the best fit line or curve, a series of regression models were constructed from the login dataset. Noticeably in Chart 1, the data has some curvature to it. Therefore, almost immediately I identified a simple linear regression wouldn't suffice.

$$[1] \qquad client\_logins_t = \beta_0 + \beta_1 login\_hour_t + \varepsilon_t$$

where *client_logins* represents the count of client logins within each hour of the day and week, and *login_hour* represents the time of the day (e.g. 0, 1, …, 24).

This simple regression has two obvious flaws: (1) There is a possibility of selection bias of clients logging on more frequently during specific days of the week. If this is true, then the independence in the error term is violated. This violation can bias the effect of *login_hour* on count of *client_logins*.

$$E(\varepsilon \mid login\_hour) \neq 0$$

And (2), the data reveals curvature. Since equation 1 is only a simple linear regression, the model will not be able to pick up the dynamic curvature in the data series. Therefore, I developed a polynomial regression.

$$Y_t = \beta_0 + \beta_1 X_t + \beta_1 X^2_t + \ldots + \sum \beta_k X^n_t + \varepsilon_t$$

where X_t is the *login_hour*$_t$, and k is the number of power terms in the polynomial regression. I turned away from building a higher order of power polynomial regression model because as terms were added, those terms quickly resulted in high multicollinearity. Therefore, I stayed with a 2nd power polynomial regression model.

$$[2] \qquad client\_logins_t = \beta_0 + \beta_1 login\_hour_t + \beta_2 login\_hour^2_t + \varepsilon_t$$

I considered that equation 2 might be able to capture the curvature in the data series, but with such low dimensions (2nd power) it might not be flexible enough to capture sudden changes in slope. This led me to thinking about other types of regression models.

As stated earlier, clearly in Chart 1, there is curvature in the data series. It is also noticeable that there are sudden changes in slope. Specifically, at 8 AM and 8 PM (hour 20 in military time). Thinking about the sudden slope changes from an Uber client perspective, 8 AM and 8 PM can be considered as rush hour times. Knowing that 8 AM and 8 PM are hours with sudden changes in count of client logins, I moved into building a piecewise/spline regression with two knots. One at 8 AM and the other at 8 PM.

$$Y_t = \beta_0 + \beta_1 X_t + \beta_2 Z_{1t} + \beta_3 Z_{2t} + \varepsilon_t$$

where,

$$Z_{1t} = D_{1t}(X_t - 8); \; Z_{2t} = D_{2t}(X_t - 20)$$

Both D_{1t} & D_{2t} are dummy variables based on the value of hour in the day (e.g. 0, 1, …, 24). More specifically, since 8 AM and 8 PM are knots known in advance, then for the first dummy variable, when $X_t \leq 8$, $D_{1t} = 0$ and when $X_t > 8$, $D_{1t} = 1$. For the second dummy variable, when $X_t \leq 20$, $D_{2t} = 0$ and when $X_t > 20$, $D_{2t} = 1$.

Using the first dummy variable, D_{1t}, I created the corresponding spline adjustment variable Z_{1t} as $Z_{1t} = D_{1t}(X_t - 8)$. Notice that whenever X_t is less than 8, $D_{1t} = 0$, so Z_{1t} can never be negative. Furthermore, Z_{1t} is equal to 0 at $X_t = 8$, but takes on values 1, 2, 3… as X_t takes on values 9, 10, 11…. Thus, the effect of Z_{1t} is introduced gradually as X_t moves beyond 8. In a similar way, I created Z_{2t} in the same manner relative to its spline knot value of 20 (8 PM). Therefore, the following piecewise regression was fit.

$$[3] \qquad client\_logins_t = \beta_0 + \beta_1 login\_hour_t + \underbrace{\beta_2 Z_{1t}}_{Knot\ 1} + \underbrace{\beta_3 Z_{2t}}_{Knot\ 2} + \varepsilon_t$$

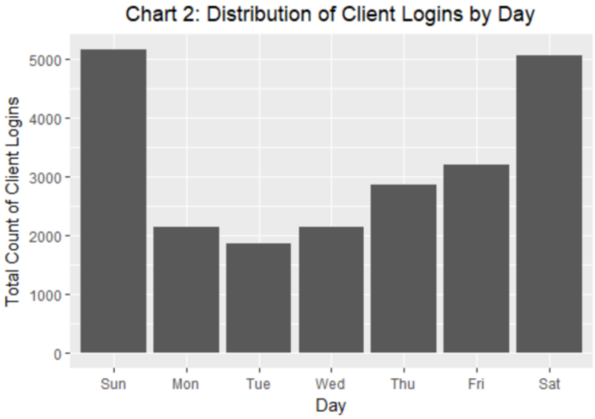
By substituting into equation 3 for Z_{1t} and Z_{2t} and using the two spine knots, 8 and 20, I evaluated each fitted line across three different hourly periods.

Hours 0 through 8: $Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t$

Hours 9 through 20: $Y_t = (\beta_0 - 8\beta_2) + (\beta_1 + \beta_2)X_t + \varepsilon_t$

Hours 21 through 23: $Y_t = (\beta_0 - 8\beta_2 - 20\beta_3) + (\beta_1 + \beta_2 + \beta_3)X_t + \varepsilon_t$

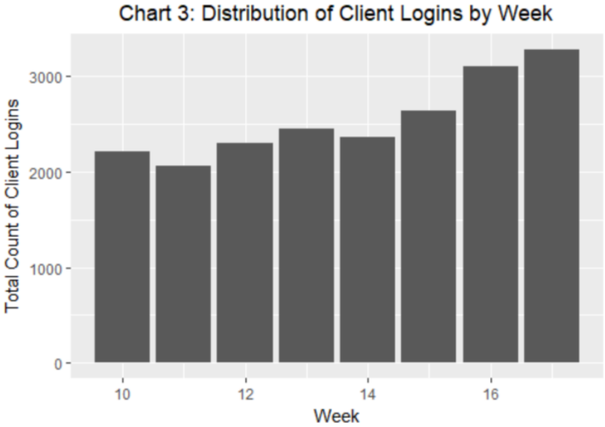
I mentioned earlier in equation 1 that there is a possibility of selection bias occurring where clients might be logging on more frequently during specific days of the week. Observing Chart 2, we can see that there are more count of client logins during the weekends over weekdays.



To control for this, I added a weekend dummy variable to equation 3. More specifically, if day equals 'Sunday' or 'Saturday', *weekend* = 1. For all other day's *weekend* = 0.

[4] $$client\_logins_t = \beta_0 + \beta_1 login\_hour_t + \beta_2 Z_{1t} + \beta_3 Z_{2t} + \beta_4 weekend_t + \varepsilon_t$$

Lastly, I noticed that there is an upward trend across the weeks of client logins. Note in Chart 3, I excluded weeks 9 and 18 because they were not full weeks. Week 9 only had data points for Thursday, Friday, and Saturday. While week 18 only had data points for Sunday, and Monday.



Consequently, the final model I created included fixed effect (FE) week variables.

$$[5] \qquad client\_logins_t = \beta_0 + \beta_1 login\_hour_t + \beta_2 Z_{1t} + \beta_3 Z_{2t} + \beta_4 weekend_t + \sum_{j=1}^{9} \theta_j week_t + \varepsilon_t$$

where *j*, represents the week dummy variables. There are 10 weeks within the dataset. Week 9 was withheld and used as the reference point. Note, most FE week variables were not statistically significant (based on p-value of 5%). So, I tested for exclusion restrictions by using the R^2 form of the F-statistic.

$$F = \frac{(R^2_{UR} - R^2_R)/q}{(1 - R^2_{UR})/n - k - 1}$$

$$F = \frac{(.54^2_{UR} - .51^2_R)/9}{(1 - .54^2_{UR})/1{,}422} = 9.96$$

where R^2_{UR} includes all FE week variables, R^2_R excludes all FE week variables, and *q* is the number of restrictions imposed in the moving from the unrestricted to the restricted model. With a *5%* significance level, $q = 9$, and $n - k - 1 = 1{,}422$, the critical value is *1.88*. Therefore, *9.96 > 1.88,* and lead me on to choosing equation 5 as the best fit model with a 54% R^2. All 5 regression models are included in Table 1.

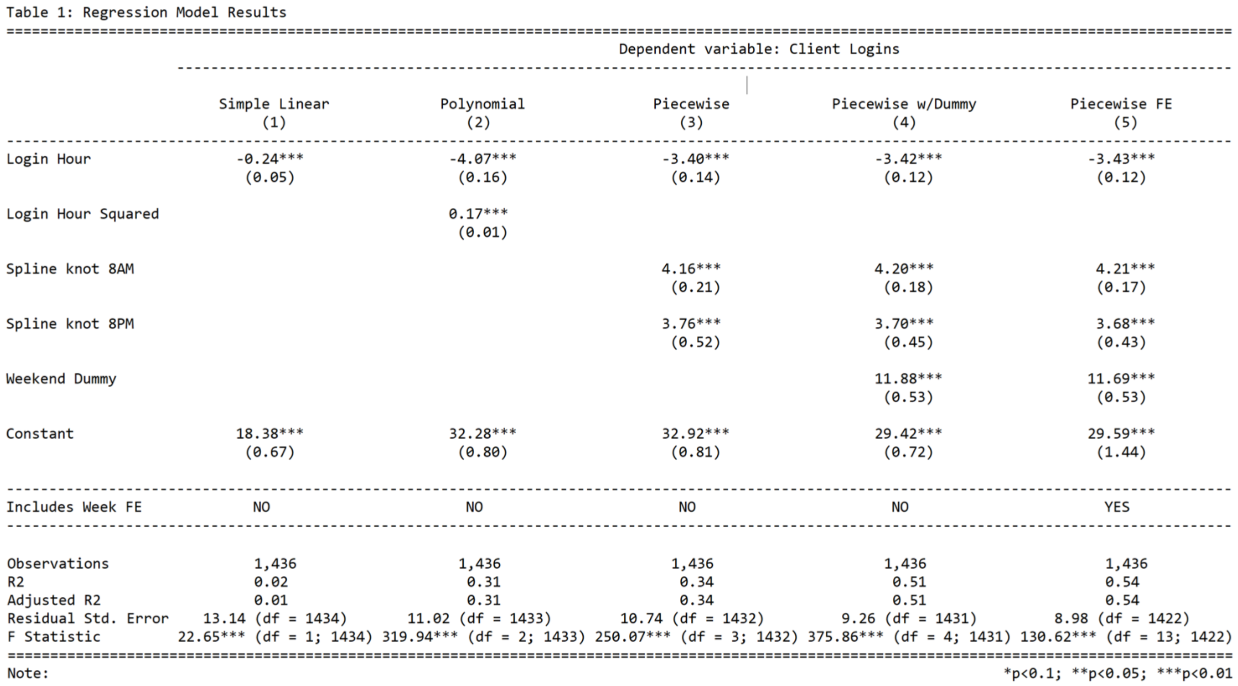


Table 1: Regression Model Results

	Dependent variable: Client Logins				
	Simple Linear (1)	Polynomial (2)	Piecewise (3)	Piecewise w/Dummy (4)	Piecewise FE (5)
Login Hour	-0.24*** (0.05)	-4.07*** (0.16)	-3.40*** (0.14)	-3.42*** (0.12)	-3.43*** (0.12)
Login Hour Squared		0.17*** (0.01)			
Spline knot 8AM			4.16*** (0.21)	4.20*** (0.18)	4.21*** (0.17)
Spline knot 8PM			3.76*** (0.52)	3.70*** (0.45)	3.68*** (0.43)
Weekend Dummy				11.88*** (0.53)	11.69*** (0.53)
Constant	18.38*** (0.67)	32.28*** (0.80)	32.92*** (0.81)	29.42*** (0.72)	29.59*** (1.44)
Includes Week FE	NO	NO	NO	NO	YES
Observations	1,436	1,436	1,436	1,436	1,436
R2	0.02	0.31	0.34	0.51	0.54
Adjusted R2	0.01	0.31	0.34	0.51	0.54
Residual Std. Error	13.14 (df = 1434)	11.02 (df = 1433)	10.74 (df = 1432)	9.26 (df = 1431)	8.98 (df = 1422)
F Statistic	22.65*** (df = 1; 1434)	319.94*** (df = 2; 1433)	250.07*** (df = 3; 1432)	375.86*** (df = 4; 1431)	130.62*** (df = 13; 1422)

Note: *p<0.1; **p<0.05; ***p<0.01

Other models were considered however not included in the analysis. The classical regression models I used could be insufficient for explaining all of the interesting dynamics of the time series. I considered using time series models such as an autoregressive (AR) or autoregressive moving average (ARMA) model. Which then lead me to consider nonstationary models, such as an autoregressive integrated moving average (ARIMA) model. It is not obvious that there is a sort of regularity happening over time in the behavior of the time series, especially with the upward trend occurring. However, I am willing to assume that regularity does exist over time and that the time series is stationary. Thus, ARIMA models were excluded from the analysis.

Q3) in a short write up, discuss any significant trends or deviations you observe in the dataset.

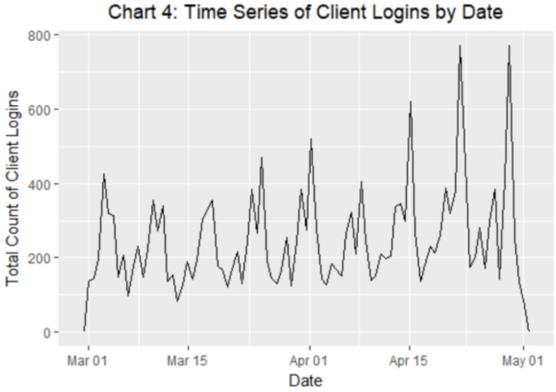


Chart 4 is a time series plot on the raw format of the data (data-time format, e.g. 2012-03-01 00:05:55) of client logins. The behavior of time series plot introduces a few visible anecdotes.

First, there is an upward trend in the series during the latter part of April. This gradual increase seems to be deviating away from the mean. A motivating question to ask is if the deviation from the mean is caused organically, say from the demand and popularity of Uber rides, or is it related to a change in policy or some other factor that caused a shift in the time series. In addition to the upward trend in the latter part of April, the volatility of client logins is also increasing during that time. The increase in volatility could be driven by the increase in variation of types of clients. Having additional characteristics of the time series data can be used to investigate both questions.

Another insight is that the regular variation imposed on the time series appears to have a cyclical pattern. In Chart 2, it is evident that weekends have higher client logins compared to weekdays. That pattern is consistent across the entire time series of data points. Therefore, there are cyclical patterns of demand throughout the week for clients. This insight can be used in various ways. For one, operations can be impacted by the cyclical patterns of the week. There might not be enough drivers on the road to maximize the transactions of client logins. Additionally, the cyclical spikes can affect servers. If servers go down because the cyclical spikes were not prepared for, many clients might not be able to complete a rider transaction.

Q4) Repeat this analysis by graphing client logins by week and by hour of day, noting any interesting finding. Based on what you find what you think it is.

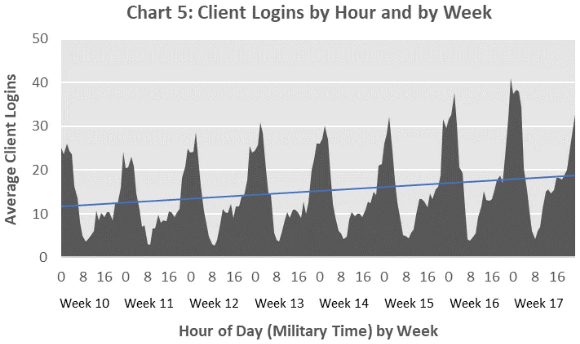


Chart 5 is a graph of average hourly client logins within the days of a week by weeks of the dataset.

I mentioned earlier that there seems to be a cyclical pattern of client logins. Chart 5 reveals the cyclical pattern much more clearly. Regardless of the day within a week (e.g. Monday vs Wednesday vs Saturday), there seems to be a pattern of hourly client logins within the dataset. Being an Uber client myself as well as having countless of friends who are Uber clients, we tend to login onto Uber and request for rides in the evening, when I need to get home from work or when I am going out to dinner with my wife.

The peaks and valleys of the plot in Chart 5 appears to be following rush hour times for clients. 6 to 8 AM are typical hours when people need to get to work and the evenings (8 PM) are typical hours when people are getting back home from work or going out. The highest point (midnight) in the plot also makes sense. These might be people who went out with friends for a couple drinks and played it safe by catching an Uber ride.

Another noticeable insight is the fitted linear line appears to have an upward trend across the weeks of the data series. I also saw this pattern in Chart 4. However, I am not entirely convinced that the upward trend is permanently deviating away from the mean. It is reasonable to believe the deviation from the mean can be caused organically from the demand and popularity of Uber rides. But there are not enough consistent upward trending weeks in the data series to persuade me. In fact, only two of the eight weeks (note, weeks 9 and 18 were excluded because they were not full weeks) appear to have a significant upward lift. If additional weeks of data have the similar upward trend that exists in week 16 and 17, then the series is more likely to be permanently deviating away from the mean.