Lumo Data Exercise

The Lumo Lift sensor collects posture data, step data, and other activity data while being worn. We provide you with a Redshift cluster containing the datasets for this exercise. If you've ever wanted to try Redshift or if this your first time hearing about it then this is a great chance to try it out! This exercise is divided into two parts, a **required** SQL proficiency section and an **optional** open-ended data challenge. **You must share all your source code with us either through GitHub or email.**

Amazon Redshift

There is plenty of disk space available in the cluster but be very aware of the amount of working memory your queries use.

Cluster Configuration:
single-node dc1.large, 7 ECUs (2 virtual cores), **15GB RAM**, 160GB SSD

The following info will allow you to connect to the cluster using any method you want. You can use your favorite compatible SQL client, programming language, or any other piece of software.

Credentials

JDBC URL:
`jdbc:redshift://data-challenge-chai.cvq7kcm4gnyy.us-west-2.redshift.amazonaws.com:5439/lumochallenge`
Endpoint:
`data-challenge-chai.cvq7kcm4gnyy.us-west-2.redshift.amazonaws.com:5439`
username:
`challenger`
password:
`AvoirMangeDuLion99`
database:
`lumochallenge`
port:
`5439`

The cluster will be taken down on 2016-04-07 at 23:59:59 (pacific time)

The Dataset

The datasets are split into user profile data and user activity data. The data is a sample of the type of data we collect at Lumo. **We purposely included outliers, bad values, and null values in both datasets so try your best to quantify them and implement a strategy to best exclude them.**

User Profile Data: All this is stored in the `owner_metadata` table. There are 12,708 rows where each is a profile.

column name	notes
owner	the anonymized unique id associated with a user. use this to join on activity data
date_started	first day this user started using the Lumo Lift YYYY-MM-DD
age	
gender	
height	measured in meters
weight	measured in kilograms
client_os	client OS used to register account. android, ios, windows, etc
locale	primary locale. follows ISO 15897

Activity Data Format: All this data is stored in the `activity_data` table in “long” format. The data is a sample of roughly 3 months of Lumo Lift data [2015-10-01, 2016-01-01) from all users. There a total of **180,710,380 rows**.

column name	notes
owner	Anonymized unique id of the user this piece of data belongs to
sensor_id	ID associated with the Lumo Lift sensor that generated this piece data
act_type	A code representing a specific activity (see data dictionary)
act_value	The value associated with the activity (see data dictionary)
act_time_gmt	UTC of when this user performed the activity
act_time_local	Local time of when this user performed the activity
upload_time_gmt	UTC time our servers received this piece of data

Each row represents a single activity type performed by an owner and sensor_id within a 5 minute block of time. Therefore, a single 5 minute block for a given owner and sensor_id can have multiple rows. This is an example of a 5 minute block of data for a single user:

```
owner          act_type    act_value  act_time_gmt         act_time_local       upload_time_gmt
3833934073541  STG         1          2015-10-07 12:10:00  2015-10-07 14:10:00  2015-10-07 12:16:59
3833934073541  STBF        2          2015-10-07 12:10:00  2015-10-07 14:10:00  2015-10-07 12:16:59
3833934073541  SG          18         2015-10-07 12:10:00  2015-10-07 14:10:00  2015-10-07 12:16:59
3833934073541  SBF                    2015-10-07 12:10:00  2015-10-07 14:10:00  2015-10-07 12:16:59
3833934073541  C_STEPS     34         2015-10-07 12:10:00  2015-10-07 14:10:00  2015-10-07 12:16:59
3833934073541  C_DIST      1420       2015-10-07 12:10:00  2015-10-07 14:10:00  2015-10-07 12:16:59
```

The activity data table has an interleaved sort key on the `owner, act_type, act_time_local, and act_time_gmt` columns. For the curious, this article explains the algorithms / math behind interleaved sorting (it’s very cool).

Part 1: Redshift PostgreSQL

Redshift is based on PostgreSQL 8.0.2 and supports most of its features and data types. The solution to each problem must be in its own file. **Solutions must only be a single query but can include subqueries, and ‘with’ statements.** Each problem comes with example output.

Problem 1: DAU/MAU

Calculate the DAU/MAU for every day in the month of October.

local_date	dau	mau
2015-10-10	1879	7040
2015-10-11	1757	7040
2015-10-12	2302	7040

In the above sample output, we see that on October 10: 1879 (dau) unique users had data and 7040 (mau) unique users in the month of october had data.

Problem 2: Correlation

The Lumo Lift has a coaching feature that actively vibrates whenever a user begins to slouch. Calculate the correlation between coach vibration buzzes (C_CVBUZZ) and good posture time (SG + STG + CG) per user per day in the month of December.

owner	local_date	corr_ab
3952609873541	2015-12-22	-0.213
2409509873541	2015-12-11	-0.2229
4358509873541	2015-12-14	0.0166

Problem 3: Confidence Intervals

Posture Score is defined as the following proportion of activity types (good posture time/ total posture time):

$$\frac{SG + STG + CG}{R + W + CG + CBS + CBF + STBS + STBF + STG + SBS + SBF + S + SG}$$

Using activity data from the month of October calculate the average daily posture score per gender as well as the upper and lower bounds of the 95% confidence interval for each gender.

gender	lower_bound	avg_posture	upper_bound
f	0.3874	0.4015	0.4156
m	0.4549	0.4691	0.4833

Problem 4: Counting Streaks

For every user between November 15 (inclusive) and December 15 (exclusive), calculate their longest streak of continuous daily usage AND count the total number of streaks. A streak is 2 days or more of *continuous* use. Only include days when a user has >= 30 minutes of good posture OR has >= 500 steps.

owner	longest_streak	n_streaks
9468609873541	23	3
1451609873541	23	2
5711609873541	23	2

In the above example, user 9468609873541 has 3 occurrences when they used the product more than 2 days continuously and the longest of the streaks was 23 continuous days.

Part 2: Data Challenge (optional)

Consider this portion a mini-hackathon where you build something using the datasets we provide. This part is purposely open-ended so create something that demonstrates what you're passionate about. **Any language, framework, toolset, and platform can be used**. You are free to use additional open datasets to supplement our data. The best place to start is simply to explore the dataset, hopefully you got a sense of the data from part 1.

- The code you develop here must be shared either through GitHub or email.
- **If you perform a statistical analysis or build a model then your results must be reproducible[1].**
- if you use third-party software libraries or packages then they must be easily accessible and installed. proprietary and/or paid packages are highly discouraged.
- If you use additional datasets or files then you must include them in your submission. Please don't use any proprietary or private datasets.

Some examples:
Imputation: Did you think we would give you perfect data? Ha ha! no. Do you know a cool technique to fill in missing data? How accurate is it?

Slouch Predictor: Based on a user's historical data, can you forecast when a user is about to slouch? When is the best time to send a notification to a user to be more aware of their posture?

User Clustering: other than the profile data we give you, are there ways to group people by the data we collect on them?

Retention: Is there a specific segment of users who use the Lumo Lift longer? more continuously?

Visualization: Come up with an engaging and insightful method of simply showing the data. There must be a better way than bar charts and scatter plots!

[1] if running your code doesn't reproduce your results then your submission will not be accepted.