Data Analyst Challenge

This challenge is comprised of two separate parts. The first task aims to assess core SQL competency and the second to assess coding, analysis and data wrangling skills. We expect this challenge will take between 4-6 hours, so please try to plan your work accordingly. If you have any concerns around the time involved then please get in contact with us as soon as possible.

Part 1: SQL

Scenario

One of our growth marketers would like to have a message conditionally inserted into a regular bulk email. The idea is to include a premium subscription upgrade CTA at the top of an email subject to the following criteria:

- Out of the users who have exported a design in the previous 7 days, the user is in the top decile by number of exports in that period.

The marketer also wants to use the category (birthday invitation, business card, etc) of the design that the user last exported.

The email marketer needs your help to identify the users in that top decile and the category of the last exported design.

Task

For an event table (called `design_exported`) containing the following columns, write an SQL query to determine the decile (1 is the lowest, 10 the highest) of each user by event count and the category of the user's last exported design.

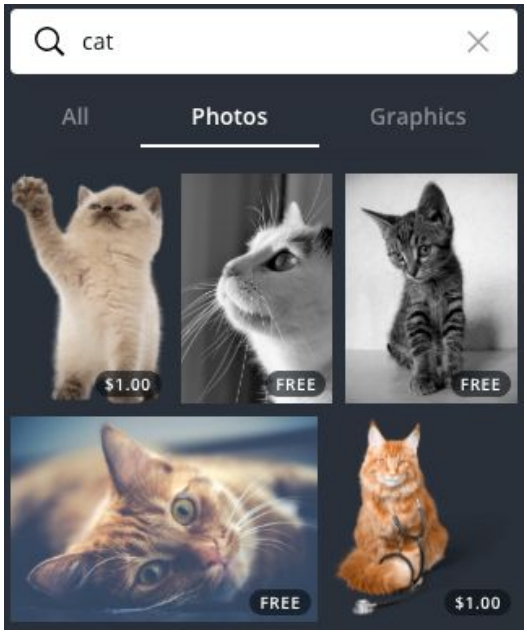
- `event_id`
    - unique identifier of each event (duplicates rows may exist in `design_exported`)
- `timestamp`
    - unix epoch time of the event (seconds)
- `user_id`
- `design_category`
    - English-language text label for the category of the exported design
    - example values: 'birthday invitation', 'business card', 'festival poster'

Please ensure that your SQL is roughly compatible with a PostgreSQL 9.5 database. We don't mind if there are minor syntax issues so you shouldn't need to install the database to test your query.

Part 2: Experiment Analysis

Scenario

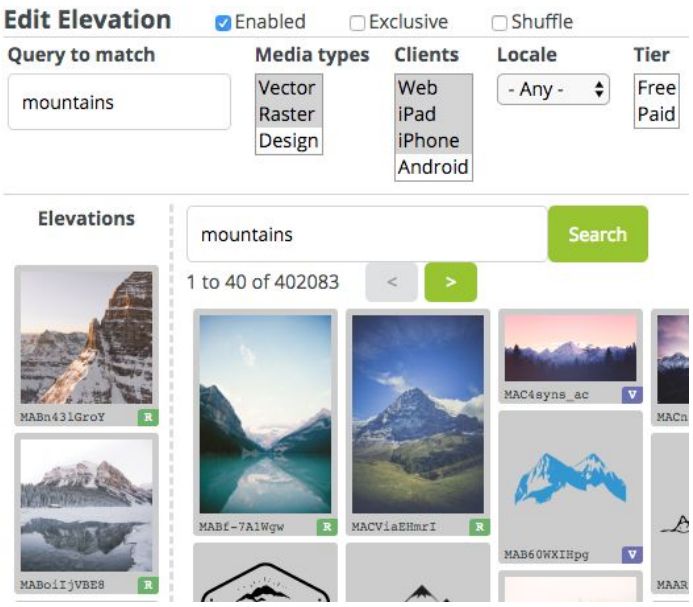
Canva's design editors have built-in image search functionality, allowing users to find a wide range of content to include in their designs.



Our search functionality is powered by an open source search engine and can provide results purely based on the similarity of image metadata with the user query string. Relying on this similarity works reasonably well but it's usually easy to find examples of queries with poor results.

To work around this our engineers developed a tool allowing people to administer "manual elevations". These are manually curated collections of search results which will appear first for any given query.

In particular, for any given combination of query string, media types (raster and/or vector), client (Web, iPad, iPhone, Android), locale and subscription tier (free or paid), a privileged user can find existing search results and drag them into the 'elevations' section on the left. The 'locale' and 'client' properties are primarily used for template search elevations and can be largely ignored here.



More recently a new mechanism has been developed that calculates elevations automatically by processing a large quantity of search data. The automated mechanism recalculates elevation sets for tens of thousands of queries daily and has demonstrated a strong overall positive impact on search results. The automated elevations mechanism calculates at most 50 elevations per query.

We have performed an experiment to assess the ongoing value of manual elevations. A subset of users were randomly assigned to one of the following experiment groups:

- Group A: Both automated and manual elevations were applied, with manual elevations taking priority
- Group B: Only automated elevations were applied to search results

Each user query was recorded along with each result click. In our data warehouse ETL process this has been aggregated and joined with publish and billing data to produce a single record per user query. This data is filtered so that only user queries that correspond to manual elevations are included and is supplied in `manual_elevations_experiment_data.csv.zip`. The fields in this file are:

- `search_id`
    - A unique identifier for each query
- `experiment_day`
    - 1 for the first day of the experiment, 2 for the second day, and so on
- `experiment_group`
    - The experiment group of the user at the time of the query

    - possible values: 'A' or 'B'
- `user_tier`
    - Indicates whether or not someone has a paid Canva subscription
    - possible values: 'free' or 'paid'
- `query`
    - The user query string
- `media_types`
    - The media type filters applied to the query (raster and/or vector filter)
    - These filters can be set in the UI with the All/Photos/Graphics selector visible in the example 'cat' query
    - possible values: 'R', 'V', 'RV'
- `num_elevations`
    - The number of search results with manual elevations (these are shown to users before any other result)
- `num_clicks`
    - The number of distinct images that were clicked on in the search results
- `num_exported_results`
    - The number of distinct images that were clicked on in the search results and then downloaded in a design
- `num_licenses`
    - The number of distinct images that were clicked on in the search results and then paid for while downloading a design

For your reference the Presto SQL query used to extract this data has been supplied in `manual_elevations_experiment_data.sql`. Looking at the query might help you understand the data a bit better.

Task

We would like to understand the ongoing value of manual elevations and what we should do with them. A full analysis would be too time-consuming for this challenge, so instead we'd like you to provide us with a report containing four components:

1. An analysis of the overall impact of manual elevations
2. A list of suggestions for how you would analyse the problem in the real-world
3. A completed analysis for the suggestion above that you believe would have the highest business impact (subject to reasonable time constraints - please don't spend more than 2 hours here)

4. A final recommendation

Please provide your report in an analytics notebook. If using Python please supply it in a Jupyter notebook. For R you can use Jupyter or R Markdown (RStudio has built-in support). You can use whatever packages you like for data wrangling, however please limit your usage of built-in or package-supplied statistics functions to very basic functions like mean, sum and variance. We're big fans of nice visualisations, however producing these can be time consuming. Feel free to skip them altogether (unless required for your chosen activity in task 3) or just keep them really simple (totally ugly is fine).