Pre-workshop instructions:

In the upcoming workshop we are trying to tackle challenges that could be grouped into 3 buckets: (I) access to BigQuery and the related stuff, (II) setting up a programming environment that allows one to analyze the data obtained, and (III) the actual process of analysis.

Ideally, we would be able to take care of (I) & (II) before the workshop, and focus more on (III) during the workshop. Even if we can't get steps (I) & (II) done for everyone before the workshop, we'll be able to address the sticking points. With that in mind, I created this document to lay out the challenges in front of us. It would help us to get the most out of the workshop if you could go through the document before the workshop.

Section I: BigQuery-related points

1. **Getting access to BigQuery:**
    a. This step is important if you need access to BigQuery. In case you have a course team member (or me) who will provide you with the downloaded files, you may not need to go through this step.

    b. To get access to the datasets of your courses, you can go directly to Jon Daries, or work with us (we are still trying to figure out the best way to go about it).

    In either case, you'll need a Gmail id, and in the first case you will probably need to set up a payment methods (credit card/free trial credit, if using the Google Cloud for the first time).

2. **Interacting with BigQuery and downloading raw data:**
    a. If you are able to access edx/mitx course datasets in your repository and know how to download files, go to https://bigquery.cloud.google.com to download the following tables from any *one* course.
        i. Person_course
        ii. Problem_analysis
        iii. Course_axis
    If you have these tables, you can perform analyses on your actual data in the workshop, rather than working with the fake data I'll provide.

    b. One way to download these tables is via creating a "bucket" in Google Cloud storage (https://console.cloud.google.com/storage/), and exporting the tables a-c into that bucket. To do that,
        i. First make sure that you create/choose the right "Project", and create a new storage "bucket", say "test_bucket".
        ii. Within the "test_bucket", create a new folder, e.g., "test_folder"

iii. From BigQuery, open a table and click on the “Export table” button.
    1. Choose the “csv” for files without nested format (e.g., person_course), and “JSON” for those with nested data (e.g., course_axis, problem_analysis)
    2. No need to use compression unless the file is really large (>1 GB)
    3. In the address bar, after gs://, type “test_bucket/test_folder/test_file”
    4. Hit export, and refresh your folder in the Storage console to ensure that you see the new file, and the file size is what you expect from the BigQuery Table Details -> Table Size.
    5. Once satisfied, just click on the file in storage, and it’ll be downloaded in your computer.

iv. If you do a lot of table exporting and downloading, you’d perhaps want to perform steps 1-5 via their API. Let me know if that’s what you’d want, and we can set you up for that process.

2. Running SQL jobs in BigQuery:

a. I’m no expert in SQL, and I avoid using this feature in BigQuery for a number of reasons:
    i. I’m no expert in SQL...
    ii. From a research point of view, having access to raw data is more versatile: you can do more analyses without having to write new SQL queries.
    iii. Visualization is also easier, as you can use your favorite programs in many different ways
    iv. It’s easy to create expensive queries by mistake (it’s possible to set up a max cost per query somewhere to avoid running into such a scenario.)

b. However, direct SQL queries could make more sense if
    i. You are planning on combing through, say, all the log files from one course for each student’s activities, or
    ii. You are making a real-time app, and low latency is the goal
    iii. want to look at some specific numbers every day, and have not much use for the raw data beyond those numbers

Let’s figure out what kind of analysis you have in mind, and if running SQL queries (on BigQuery or via API) would make more sense for your project. If so, I can provide you with starter codes written by Ike and Jon (and possibly others), and from there we can possibly figure out the queries you need to write.

Section II: Python-setup and the related points:

We are going to use python-based tools for this workshop. However you are welcome to use other programming languages/tools that can perform similar tasks. Even in the latter case, hopefully the workshop will give you an idea of how to translate the basic analysis steps in your favorite language.

In python, we would need the following packages for the workshop:
For analysis: Numpy, Scipy, Pandas, matplotlb, jupyter notebook (the last one is convenient, but not essential)

For generating plots, I use plotly. You can use Bokeh, Seaborn, or stick to matplotlib, for the workshop, if you don't want to use plotly.

If you have these packages (and/or know how to install them), you are good to go. Otherwise, please read on.

Setting up a python eco-system in your computer:

When one is creating a python development environment for the first time, unfortunately there are a number of decisions one needs to make regarding python version, package managing system etc.

TL;DR: Consider installing anaconda, if you don't have it. On MacOSx you may already have it pre-installed. You can install Conda on Windows as well. After installation, it takes about 3GB.

I recommend python version>=3.6. It's fine for the workshop if you are running python 2.7, but edx is moving on to python 3, and it's a good time that we all do so!

The article below has step-by-step guidance on installing conda (unlike in the article, I recommend you install full anaconda, and not miniconda). This will hopefully clarify the landscape a bit.
https://medium.freecodecamp.org/why-you-need-python-environments-and-how-to-manage-them-with-conda-85f155f4353c

For the masochists: If you do not want to install Anaconda, another option is to use pip (a python package installer). More here. As explained in this link, If you are going to take this route (and use, python2.7), it's strongly recommended that you also install something called *virtualenv*, which creates a separate installation of python in an isolated environment, so that

you don't have to "Marie Kondo" your laptop at a future date. Here is a very quick summary of what virtualenv does.

For python>=3.3 you don't need to install virtualenv separately -- there's an equivalent in-built program called venv that you can use to create a virtual environment.

Here is a semi-high-level view of the process:

1. We create a virtualenv (or venv), which makes a copy of system python installation inside it.
   "`python3 -m /path/to/venv`"
2. Whenever we want to use this isolated environment (which could be just one environment for all similar projects), we activate it with the following command (for bash shell in MacOSx or Linux):
   "`source /path/to/venv/bin/activate`"
3. We check that the venv is activated (there will be a "(venv)" in the terminal console). We then install requisite python packages by using something called "pip" -- package installer for python. The command for installing a python package `xyz` is as simple as:
   "`pip install xyz`"
   We also run our programs in this environment (via scripts or in jupyter notebook).
4. Finally, deactivate the virtual environment by simply typing "`deactivate`".

The process in windows and other shells is similar.

Section III: Actual analysis

If you are not super familiar with python, this 5-minute overview could be a good starting point.

For the workshop, we'll use Pandas package for analysis, which has its own set of "commands". I'll have our tasks available through jupyter notebook and standalone scripts (in case you don't have jupyter notebook installed, but have the other python packages installed).

If you are itching to start with Pandas, this official "10-minute" guide is safe for me to recommend. Although I doubt if all different sections of the guide are of equal use, and it'd be very impressive if you can make sense of the whole guide in 10 minutes!

I'll make the scripts/notebook and the data available on 4/10, before the workshop.

Questions/comments? Email anindyar@mit.edu

The dropbox folder containing the notebook and the data is here:
https://www.dropbox.com/sh/caqjwx2sak7p8aw/AABiKGP37x83KhjJKTvgo-tQa?dl=0

If you don’t have jupyter notebook installed, you can use google colaboratory:
The equivalent notebook is at
https://drive.google.com/file/d/1pR9KmjUfyCmXsk9PWAGqZFomsAFnz2H2/view?usp=sharing

Download this notebook, and upload it in your own Google Drive, connect colab, and then we’ll see how to run it.

Other resources:

1. Jolyon has written a package that could be used to catch errors in edx courses and create a pretty pdf outline of the course index: https://github.com/jolyonb/edx-xml-clean

2. Connor has a repo that could help you get an idea of how to download datasets from BigQuery: https://github.com/connor-makowski/BQ_Data_Generic

3. Martin has a code (in R) that could create pseudonymized usernames/user_id lists. Contact Martin directly if you need it.

4. Dan has a wealth of jupyter notebooks, one for each paper that he has written (and that’s a lot!). Ask him if you need more real-life examples of edx data analysis.

5. Here is the edx2bigquery repository, that generates the tables in BigQuery, and the data reference for the tables is at:
http://edx.readthedocs.io/projects/devdata/en/latest/internal_data_formats/