Data Representation Take-home Tech Assessment (TTA)

BACKGROUND

This Take-home Tech Assessment (TTA) is designed based on real-life **Eon Labs Ltd** project so it is highly relevant to our job openings. It is in no way a definitive test of your enthusiasm for technical work. However, it has the potential to become a proxy for your working style and the way you approach the resolution of problems. Please share any thoughts you have or figure it out the way you always have so that we may get a better sense how you approach a problem.

TECH ASSESSMENT

Imagine that YOU are a **Machine Learning Research Scientist** in our data science team who is collaborating with our data engineering team.

Overview

The data engineering team has done some research and found that the Google Trends data is potentially beneficial to the data science team. YOU, as a data scientist, want a **time series of consistent Google Trends data from 2017 till the present with hourly interval**. YOU informed the engineering team of this requirement, but they said they could not fetch the hourly data directly. The reason why they are unable to fetch the hourly data directly is explained in the **Deep Dive** section. They may, however, fetch the following raw data from Google Trends:

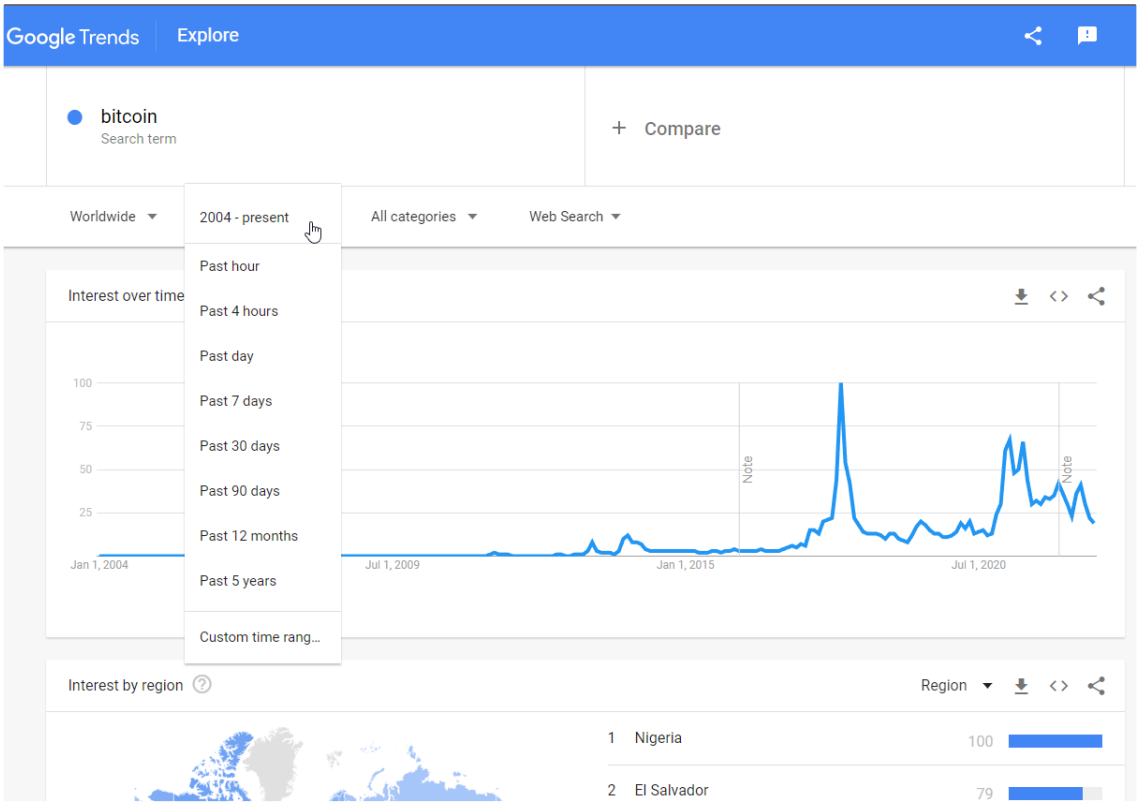
- `hourly_data.csv`: a time series of weekly-consistent Google Trends data starting in 2017 and continuing up to the present, with hourly intervals
- `weekly_data.csv`: a time series of yearly-consistent Google Trends data starting in 2017 and continuing up to the present, with weekly intervals
- `monthly_data.csv`: a time series of consistent Google Trends data starting in 2017 and continuing up to the present, with monthly intervals

What do YOU need to do?

- Carefully read the **Deep Dive** section.
- Write a Python script to solve the **Problem** using the time series files downloadable from the **Raw Data** section.

Deep Dive

The data engineering team fetched the raw data from Google Trends by way of web scraping from its website (as linked here).



As you can see, the Google Trends website offers an drop-down box for you to choose **Custom time range** (e.g. **From 2004 to present**")

The engineering team found that by choosing a time range of **2017-present**, they could only provide *time series of consistent Google Trends data with time interval of months* (downloadable as `monthly_data.csv` in the **Raw Data** section):

monthly_data.csv

E: > Downloads > monthly_data.csv

```
time_month,value_month,date
1483228800,6,2017-01-01
1485907200,6,2017-02-01
1488326400,7,2017-03-01
1491004800,6,2017-04-01
1493596800,15,2017-05-01
1496275200,14,2017-06-01
1498867200,13,2017-07-01
1501545600,20,2017-08-01
1504224000,21,2017-09-01
1506816000,23,2017-10-01
1509494400,44,2017-11-01
1512086400,100,2017-12-01
1514764800,57,2018-01-01
1517443200,40,2018-02-01
```

In order to get the time series of hourly interval, they were forced to work within a more constrained time range (i.e. a week). They are able to get a *time series of hourly data from 2017 up to the present* (downloadable as `hourly_data.csv` in the **Raw Data** section) by retrieving and concatenating week-range-data on a week-by-week basis.

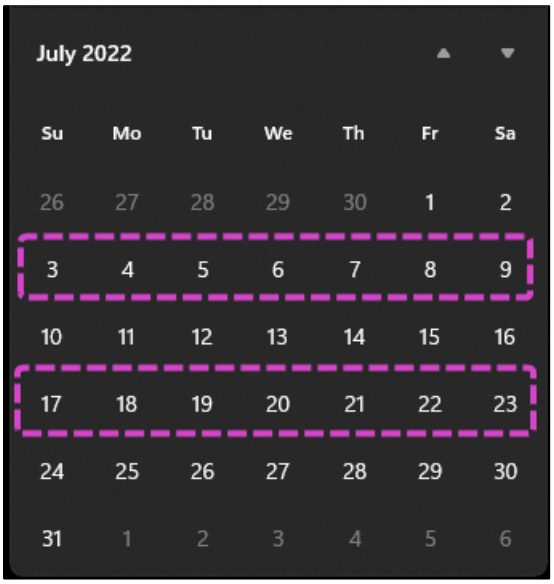
hourly_data.csv

E: > Downloads > hourly_data.csv

```csv
time_hour,value_hour,date
1483228800,30,2017-01-01 00:00:00
1483232400,34,2017-01-01 01:00:00
1483236000,33,2017-01-01 02:00:00
1483239600,43,2017-01-01 03:00:00
1483243200,32,2017-01-01 04:00:00
1483246800,32,2017-01-01 05:00:00
1483250400,32,2017-01-01 06:00:00
1483254000,28,2017-01-01 07:00:00
1483257600,25,2017-01-01 08:00:00
1483261200,22,2017-01-01 09:00:00
1483264800,19,2017-01-01 10:00:00
1483268400,27,2017-01-01 11:00:00
1483272000,26,2017-01-01 12:00:00
1483275600,29,2017-01-01 13:00:00
1483279200,34,2017-01-01 14:00:00
```

However, this hourly data are not what YOU want, since the data are not consistent!

Google scales the trends data within the window range you choose. In other words, say for example, a value_hour that equals '50' during the week from 2022-07-03 to 2022-07-09 are not the same as a value_hour that also equals '50' during the week from 2022-07-17 to 2022-07-23.



Only the value_hour numbers that sit within the same week are consistent.

Similarly, to get the *time series of weekly interval* (downloadable as `weekly_data.csv` in the **Raw Data** section), the engineering team used a time range of a year. They fetched year-range-data and concatenated them year by year from 2017 till the present.

weekly_data.csv

E: > Downloads > weekly_data.csv

```csv
time_week,value_week,date
1483228800,100,2017-01-01
1483833600,84,2017-01-08
1484438400,63,2017-01-15
1485043200,56,2017-01-22
1485648000,60,2017-01-29
1486252800,66,2017-02-05
1486857600,60,2017-02-12
1487462400,72,2017-02-19
1488067200,84,2017-02-26
1488672000,97,2017-03-05
1489276800,82,2017-03-12
1489881600,80,2017-03-19
```

By the same token, only the value_week in the same year are consistent.

Problem

With `monthly_data.csv`, `weekly_data.csv` and `hourly_data.csv` data files given to you by the engineering team, how do you use them to output **time series of consistent Google Trends data from 2017 till the present with time interval of hours**?

Write a Python script to solve this problem using the time series files downloadable from the **Raw Data** section below.

> 📢 **NOTE TO CANDIDATES WHO STARTED TO READ THIS ASSESSMENT EARLIER THAN @Last Saturday BUT DIDN'T QUITE UNDERSTAND WHAT THE WORD "NORMALIZE" MEANT.**
>
> Previously, when you saw "normalization," we meant "scaling." (How Google normalizes their trends data was unclear to us. What we knew was the data in the same fetching window was consistent but only within a range of 0-100). What we wanted you to do was not "normalization," but to make data in the 2017-present consistent. Perhaps you can understand "normalization" as in "some kind of scaling" and understand "normalized" as in "scaled and consistent." The current version of this assessment has been worded as such so that it is now more reflective of what we're trying to ask you to do.

Raw Data

hourly_data.csv 1643.0KB

weekly_data.csv 7.2KB

monthly_data.csv 1.7KB

SUBMIT YOUR ANSWER

Upload program code (or pseudo code) file(s) along with the README file for this TTA to GitHub, and send the repository link to `careers+data_representation_tta@eonlabs.com`

You are always welcomed to ask questions that you may have about this TTA by sending email to to `careers+data_representation_tta@eonlabs.com` so that our engineering team may answer your questions.

QUESTIONS & OPINIONS

Please don't hesitate to **Contact HR** for non-technical questions or express your opinions on the hiring process.

- EonLabs Job Board
  - Machine Learning Research Scientist
    - Data Representation Take-home Tech Assessment (TTA)
  - Data & Backend Software Engineer
    - Coding & Data Collection Take-home Tech Assessment (TTA)
- Eon Labs Ltd
- FAQ by Candidates
- Contact HR

ca.indeed.com

https://ca.indeed.com/cmp/Eonlabs

EonLabs - Financial Science Reimagined

Our scientists, researchers and engineers create AI-based trading models, algorithms and strategies for fund managers to achieve

https://www.eonlabs.com/