DSS Data Product Engineer Code Test

Instructions:

Programming language: Python, you may use non-standard libraries if needed.

Test Duration: Please time your task 1 and 2 (ideally the total time should be within 2 hours but feel free to spend more time if needed) and include the total time in your response.

Task 1

The attached file ***key_metrics.json*** is the json response for a web analytics reporting API. Specifically, it includes key metrics (page views, visits, unique visitors and bounce rate) data for one of our sites.

The structure of the json is irregular. The metric counts for each day can be found in

```
json["report"]["data"][i]["counts"]
```

and the metrics definitions are in

```
json["report"]["metrics"]
```

We need convert this json into a more useful format:

```
[
    {
        "date": datetime(2016, 11, 13),
        "page_views": 6209372,
        "visits": 2326077,
        // other key metrics
    },

    {
    // other dates to follow
    }
    ...
]
```

Basically a list of dictionaries, with metric definitions as the key, and metric counts as value. Note we also expect the "date" string field to be converted to Python datetime object.

Output: print the final data structure or write to a json file

Task 2

Similarly, the second attachment **products.json** includes the products traffic for top countries / cities (for a single day) - in this web analytics API, these dimensions (e.g. product, country and city) are referred to as "elements", see *json["report"]["elements"]* . The data for each element is embedded - in this specific case, the data is grouped by "country" first, then each record has "breakdown" for "city", then breakdown by "product"

we would like to "flatten" the json into the following format:

```
[
        {
                "country": "United States",
                "city": "san francisco (California, United States)",
                "product_name": "AVG AntiVirus Free 2014",
                "page_views": 215,
                "visits": 77,
        },

        // other country/city/product_name entries to follow, no need to add aggregations

]
```

This way it's easy to store the data in a database or feed it into statistics / machine learning programs.

Ideally your program should be able to handle any number of dimensions (e.g. when the report response is breakout by Country + City + County + Product + Product Category).

Output: print the final data structure or write to a json file

Task 3

Open end questions (no time limit): How would you use this dataset (top products per country/city) to generate insights that could help the business? What questions you can answer? What's your hypothesis and how would you test it?

Prepare to discuss these questions during the interview. Bonus point if you can provide visualizations, code samples and statistic test results.