Solutions:

* Final solutions located in standalone directory
* Initial analysis of problems in jupyter\_review directory

Task 1:

* Reviewed raw json to review data and format
* Used pandas to review data in jupyter notebook
  + Segmented data for review
* Spent time troubleshooting:
  + json conversion of datetime to string when writing to text file
  + Handling INF values in numeric field
* Total Time Spent: 1 hr and 10 minutes

Task 2:

* Reviewed raw json to review data and format
* Used pandas to review data in jupyter notebook
* Became familiar with nested file format
* Followed similar process as Task 1
* Explored usage of pandas json.normalize() to flatten nested structure
* Explored other ways to segment data rather than multiple for loops
* Total Time Spent: 1 hour and 35 minutes

Task 3:

* How to use datasets to generate insights:
  + Task 1:
    - I would review daily analytics data using a trend lien to explore days and times at which page views, visits, unique encounters were high and low to try to determine a correlation between the content at those times.
    - Try to understand what content draws viewership and enthusiasm in the community to help content producers understand their audience better.
  + Task 2:
    - I would use location information to determine which applications are popular and which are not in certain areas so that the relationship with those apps and the areas can be explored.
    - Determine if cultural differences attribute to the high or low usage of apps or if other factors such as slow services were a cause of issues.
    - Could use this data to determine which apps might make the most sense to releasee in certain areas and not others.
* Questions that can be answered:
  + Task 1:
    - What dates are high, low and average traffic days.
    - What content correlates to these highs and lows.
    - If more details on visitors was available:
      * Which visitors spend longer amounts of time on the content than others?
      * What specific content per page are they interacting with?
    - Would be useful to include timestamps, and pair with location and product information.
  + Task 2:
    - What locations use certain apps more frequently or less frequently than others.
    - How does proximity influence overlaps in application usage.
    - How do location demographics influence application use.
    - Which products have global utilization?

Dockerized Kafka Producer & Flask REST API:

Using Task 1 developed an initial concept to stream formatted data from source json file into postgres db for use in a flask web application API

To run:

1. docker network create kafka-network
2. docker-compose -f docker-compose.kafka.yml up -d
3. docker-compose up -d

* cd db && python review\_db.py # Check if data in postgres db
* Total Time Spent: 6 hours