GUIDELINES

1. Discard irrelevant or obviously erroneous data
    a. Most of the variable names should be self-explanatory, however data is deeply nested and will require detailed review in order to select the most appropriate data elements
2. Complete thorough EDA to identify which variables you can use to complete your analysis
    a. Any poorly populated or duplicate variables should be discarded
3. What is the **timeline** of the data? Do you see significant peaks and valleys?
    a. Do you see any data collection gaps?
    b. Do you see any outliers? Remove obvious outliers before plotting the timeline
    c. Do you see any spikes? Are these spikes caused by real activities / events?
4. What are the most popular **programming languages** on GitHub?
    a. Did the trend of most popular programming languages change over time?
5. What is the distribution of **licenses** across GitHub repositories?
    a. Any certain programming languages that are more likely to be associated with a particular license?
6. What can you tell about the most popular and most rapidly growing **repositories**?
    a. Is there certain technology that is driving popularity or explosive growth?
    b. Are these associated with Big TechLinks to an external site., who are open sourcing the technology?
    c. Are there any technological breakthroughs that are driving this brisk adoption?
7. Identify what **technologies** are most frequently associated with Data Science or AI projects. Did these technologies change over time?
8. What are the **most frequent reasons** for committing into GitHub repositories?
    a. Is this new technology development, bug fix, etc.
9. Identify the most prolific / influential **Committers**
    a. By commit volume
    b. Visualize the distribution of these commits
10. How unique are the **“subject” and “message”** values?
    a. Are they mostly unique? Or are people usually just copy-pasting the same text?
    b. You can use LSH to measure uniqueness / similarity
    c. Visualize “subject” and “message” duplication across all programming languages
    d. Visualize “subject” and “message” duplication for each of the top 5 programming languages