Data collection from BigQuery

1. Decide the Time Unit of Analysis, i.e., week, month, etc. – Day
2. Unit of data collection: Repo – day – actor.id
3. Shortlist the list of activities we are calculating:

* num\_dist\_commits
* num\_dist\_commitcomments
* num\_actors\_pushevents
* num\_actors\_pusheventscomment
* num\_dist\_pullreqopened
* num\_dist\_pullreqclosed
* num\_dist\_pullreqAll
* num\_dist\_pullreqcomments
* num\_actors\_pullreq
* num\_actors\_pullreqcomment
* num\_dist\_issuesopened
* num\_dist\_issuesclosed
* num\_dist\_issuesAll
* num\_dist\_issuecomments
* num\_actors\_issues
* num\_actors\_issuescomment
* num\_actors\_allevents
* num\_actors\_issues\_opened
* num\_actors\_issues\_closed
* num\_forks\_event
* num\_actors\_forks
* num\_watch\_event
* num\_actors\_watch

1. Collect data from GH Archive tables 201508-today
   1. Collect and save in .csv separately by years:
      1. Stage I: 2015
      2. Stage II and on: 2016, 2017, 2018, 2019Q1, 2019H2, 2020H1, 2020H2, 2021H1, 2021H2, 2022H1, 2022H2, 2023H1.
   2. Break down in quarters (3 months) if # of records larger than 1,000,000.
   3. It’s going to be a big database!

**Table 1. Database for Ethereum GitHub Project**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Time Unit of Analysis | Repo\_name | Actor.id | Actor\_login | Num\_activities | Num\_activities | ... | Num\_activities |
| Time1 | Repo.name1 | actorid001 |  |  |  |  |  |
| Time1 | Repo.name1 | actorid002 |  |  |  |  |  |
| Time1 | Repo.name1 | ... |  |  |  |  |  |
| Time1 | Repo.name2 | actorid001 |  |  |  |  |  |
| Time1 | Repo.name2 | actorid002 |  |  |  |  |  |
| Time1 | Repo.name2 | ... |  |  |  |  |  |
| ... | ... | ... |  |  |  |  |  |
|  | Repo.nameN | actorid002 |  |  |  |  |  |
|  | Repo.nameN | ... |  |  |  |  |  |

Other interesting analyses that require additional computation may be performed within a shorter timeframe.