Ethereum Project: Data Collection Manuscript

Alina Chen, Mariia Petryk, Jiasun Li

# All Events

You will need access to [BigQuery](https://cloud.google.com/bigquery), a data warehouse created by Google to manage and analyze data, and [GH Archive](https://www.gharchive.org/), a project that records and archives the public GitHub timeline.

## 2015 to Present

On BigQuery, run the following script: <https://docs.google.com/document/d/1EgzNXdJjzMDhnUz9BjEuB-uMK0V9h5Z47QJSQcJSVnY/edit?usp=sharing>.

This script calculates and returns the frequency of twenty-eight different events: num\_activities, 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\_actors\_pullreq\_opened, num\_actors\_pullreq\_closed, 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, num\_releases, release\_payload.

BigQuery should return a table with entries that specify repositories, dates, actor IDs, actor logins, and the different recorded events for the year 2015.

Save the results as a local CSV file. Repeat the script for 2016, 2017, 2018, 2019, 2020, 2021, 2022, and 2023 by changing the line “FROM (SELECT \* FROM `githubarchive.month.\*` WHERE \_TABLE\_SUFFIX BETWEEN '201501' AND '201512') t1 WHERE t1.repo.name LIKE 'ethereum/%'” accordingly. Save each result as a CSV file.

## 2013 and 2014

Due to information shortages, we will begin to work locally on BigQuery, instead of relying on GitHub Archive. Prior to 2015, only actor logins were used to identify actors. Actor IDs had not been assigned to users. This resulted in slightly different changes in the scripts we used.

\*mariia explain how you retrieved the individual data files for each month of 2013 and 2014\*

On any IDE, such as Visual Studio Code, run the following Python script:

<https://docs.google.com/document/d/1XePirIttPQVDBOA5PNg7YpgknRt947mQkitOEz4IyB4/edit?usp=sharing>.

This script combines each CSV file into one called AllData20132014.csv.

\*mariia used her own SQL script to generate another merged file maybe have that instead\*

On BigQuery, click add to upload a local file. Select AllData20132014.csv and a dataset to create the table. Then, run the following script: <https://docs.google.com/document/d/1at-KWEDrOUjcGVi8TT8X3ndWlIEPeA4fP4e1ztx6Yj8/edit?usp=sharing>.

Replace this line, from ethereum-project-383415.Data.Data20132014Merged, with your project and dataset name. My project is named ethereum-project-383415 and my dataset is named Data. This can vary, depending on user preferences.

This should generate twenty-eight different events for the years 2013 and 2014. Save the result as a CSV file.

# Core Users

## 2015 to Present

To test our first hypothesis, run the following Python script using an IDE: <https://docs.google.com/document/d/1ZHnmQ78TM00MtUnYASRu1zEaS27ve95XuVL7YlXi97k/edit?usp=sharing>.

Change the following lines: df = pd.read\_csv('Data2015.csv', low\_memory = False), merged\_df[columns\_to\_display].to\_csv('Data2015CoreUser.csv', index = False) to match the corresponding year. Repeat the script for 2016, 2017, 2018, 2019, 2020, 2021, 2022, and 2023. A file containing the results for each year should be generated in the chosen directory.

## 2013 and 2014

In the CSV file that contains all the events for 2013 and 2014, create an empty column called actor\_id to the right of date1 and to the left of actor\_login.

Now, run the following Python script using an IDE: <https://docs.google.com/document/d/18EIGAKRlG2rcNQChECNuSr52YKeBc5bCjYg3vyx2ibw/edit?usp=sharing>.

actor\_login is used as the unique identifier instead of actor\_id.

# Shiny Visualization

The website can be found at the following URL: <https://alinachen.shinyapps.io/App-1/>. The code for the application can be found in Project Ethereum – Visualization.R.

The website allows the user to choose how they want the data to be aggregated (days, weeks, months, and years). The user can also select a custom date range and the variable the user wants to compare time to. If the custom date range selected cannot be aggregated, for example trying to aggregate five days monthly, no data will be displayed.

The website also allows the user to customize the graph, giving them an option to change the color of the line and whether it is solid or dashed. You can also choose to filter out bots from the displayed results by clicking the checkmark next to “Exclude bots?”.

For a working website, you must define the user interface and the server. When writing the script for the website, remember that the CSV file must be read outside of the definitions for the user interface and the server. The Shiny tutorial found at <https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html> can be completed for assistance. Website details can be found within the script.

The CSV file is kept locally, so the user does not need to upload a file themselves. The CSV should be kept in the same directory as your R script. It can be found under the name Data\_AllYears\_Merged.csv.

Please email [alinachen2028@gmail.com](mailto:alinachen2028@gmail.com) if you encounter any issues.