2023-08-27, Sun

Participants: Jiasun and Mariia

Discussed:

1. Cleaned data – Mariia; Done
   1. Arachnid added as bot
   2. Duplicates removed
   3. New file Data\_AllYears\_Merged\_with core and bot\_v02.csv to use for visualization – Alina

2023-08-20, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Shiny app
   1. Server connection interrupts
2. Network visualization

2023-08-14, Mon

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Shiny app use and dataset collection whitepaper add:
   1. Describe Shiny app usage – Alina; Done.
2. Spikes in data
   1. Winsorize for regression analysis – ~99% one sided
3. Next steps:
   1. Network visualization

2023-08-07, Mon

Participants: Alina, Jiasun, and Mariia

Discussed:

* Web application <https://alinachen.shinyapps.io/App-1/>
* Anomalies – Alina
* Anomalies
  + Commits peak 2020-02-01, 02 by actor yann300: <https://github.com/yann300>

In GHArchive indeed as many commits (script for check below):

* SELECT \* ##count(id), count(distinct id)
* from (SELECT \* FROM `githubarchive.month.\*` where \_TABLE\_SUFFIX between '202002' and '202002')
* where actor.id=6940742 and type = 'PushEvent' and repo.name like 'ethereum/remix-plugins-directory'
  + - yann300 – not bot
  + Activites peak 2017-03-26 by actor Arachnid (2K) + many others (e.g., fjl, holiman, bas-vk, pipermerriam etc.) on that day in hundreds of activities
  + Spike in issues closed
  + Suggested solution: add Arachnid to bots; fix the scale of Y-axis

2023-07-30, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Shiny visualization tool - Alina
   1. Tooltips
   2. Time window feature
2. Extreme values
   1. Check on daily level
3. H2 not supported
   1. Add commentary

* Next meeting: Monday, 4:00 PM EST

2023-07-23, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Shiny program filters - Alina
   1. Data file in the backend - Done
   2. Aggregation period: month, week, year, day - Done
   3. Time period – in 2 weeks
2. Hypothesis testing - Mariia
   1. H2 testing – In progress
   2. H3 testing - In progress

To Do:

* Next meeting – 4:00 PM EST

2023-07-16, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Manuscript – Alina sent the file
   1. Complete with 2013-2014 description – Mariia. Done
2. Hypothesis testing – H2 testing; In progress
   1. Compute metrics – Mariia. Done

To Do:

* Visualization – Alina
* Next week – 9:30 PM EST - Done.

2023-07-09, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Manuscript:
   1. Data collection documentation, aka ‘readme’ – Alina; first draft by July 20.

E.g., <https://github.com/friedhelmvictor/lob-dex-wash-trading-paper>

* 1. Description of the data collection

1. Hypothesis testing:
   1. Causality – for future

2023-06-25, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Missing data in 2014 – In progress; waiting for Alina’s confirmation. – Done.

To Do:

1. Fix missing GhRepo, date1, and actor.login in 2014. – Alina, Mariia

# SELECT CONCAT('ethereum/', JSON\_EXTRACT\_SCALAR(t1.repo, '$.[repo.name](http://secure-web.cisco.com/1FheoUHDcnkUln4jB6nEhtx4ygjGfB--UIj0C987FQBNU6a2XyObUy8gRy9ErF_hfH_OcGXtuNCRPpyrtbLaqU0fOTFlPODH9Mgde6_J4qKeSU3xf0BCEfZ5CSUY87wz9fnRMnjUDZcQEk5glQPBRKefFilA4F0nm23AqK6q6Flgz0TmT8yYSWTTDm0o7CwNys6rfhlx0OVqu-hYCpAbSYy5WnWfK1RiI1dys2UHkYnzJ0xb0zCqDGUjUnEbl5yrz-ms_2GmQaF8rFxXE0rTQvHRWYQCiICm2-jtU30MDT3S-9rDG-r9FkW2izxMZD__1JlKAgwFQsR9NzUps4Z6XEN2zPIsBLB9piHoTPYdfcHLjnjmwJGFONzPU4ZySu9O_ermZMNM3wtFLMxCIWnxLEgwuKwz_zL4Bpvjt1v8vdeotjT1dm14tBrJn27tAToki/http%3A%2F%2Frepo.name)')) AS GhRepo,

SELECT CASE WHEN JSON\_EXTRACT\_SCALAR(t1.repo, '$.[repo.name](http://secure-web.cisco.com/1FheoUHDcnkUln4jB6nEhtx4ygjGfB--UIj0C987FQBNU6a2XyObUy8gRy9ErF_hfH_OcGXtuNCRPpyrtbLaqU0fOTFlPODH9Mgde6_J4qKeSU3xf0BCEfZ5CSUY87wz9fnRMnjUDZcQEk5glQPBRKefFilA4F0nm23AqK6q6Flgz0TmT8yYSWTTDm0o7CwNys6rfhlx0OVqu-hYCpAbSYy5WnWfK1RiI1dys2UHkYnzJ0xb0zCqDGUjUnEbl5yrz-ms_2GmQaF8rFxXE0rTQvHRWYQCiICm2-jtU30MDT3S-9rDG-r9FkW2izxMZD__1JlKAgwFQsR9NzUps4Z6XEN2zPIsBLB9piHoTPYdfcHLjnjmwJGFONzPU4ZySu9O_ermZMNM3wtFLMxCIWnxLEgwuKwz_zL4Bpvjt1v8vdeotjT1dm14tBrJn27tAToki/http%3A%2F%2Frepo.name)') IS NULL THEN JSON\_EXTRACT\_SCALAR(t1.payload, '$.[repo.name](http://secure-web.cisco.com/1FheoUHDcnkUln4jB6nEhtx4ygjGfB--UIj0C987FQBNU6a2XyObUy8gRy9ErF_hfH_OcGXtuNCRPpyrtbLaqU0fOTFlPODH9Mgde6_J4qKeSU3xf0BCEfZ5CSUY87wz9fnRMnjUDZcQEk5glQPBRKefFilA4F0nm23AqK6q6Flgz0TmT8yYSWTTDm0o7CwNys6rfhlx0OVqu-hYCpAbSYy5WnWfK1RiI1dys2UHkYnzJ0xb0zCqDGUjUnEbl5yrz-ms_2GmQaF8rFxXE0rTQvHRWYQCiICm2-jtU30MDT3S-9rDG-r9FkW2izxMZD__1JlKAgwFQsR9NzUps4Z6XEN2zPIsBLB9piHoTPYdfcHLjnjmwJGFONzPU4ZySu9O_ermZMNM3wtFLMxCIWnxLEgwuKwz_zL4Bpvjt1v8vdeotjT1dm14tBrJn27tAToki/http%3A%2F%2Frepo.name)') ELSE CONCAT('ethereum/', JSON\_EXTRACT\_SCALAR(t1.repo, '$.[repo.name](http://secure-web.cisco.com/1FheoUHDcnkUln4jB6nEhtx4ygjGfB--UIj0C987FQBNU6a2XyObUy8gRy9ErF_hfH_OcGXtuNCRPpyrtbLaqU0fOTFlPODH9Mgde6_J4qKeSU3xf0BCEfZ5CSUY87wz9fnRMnjUDZcQEk5glQPBRKefFilA4F0nm23AqK6q6Flgz0TmT8yYSWTTDm0o7CwNys6rfhlx0OVqu-hYCpAbSYy5WnWfK1RiI1dys2UHkYnzJ0xb0zCqDGUjUnEbl5yrz-ms_2GmQaF8rFxXE0rTQvHRWYQCiICm2-jtU30MDT3S-9rDG-r9FkW2izxMZD__1JlKAgwFQsR9NzUps4Z6XEN2zPIsBLB9piHoTPYdfcHLjnjmwJGFONzPU4ZySu9O_ermZMNM3wtFLMxCIWnxLEgwuKwz_zL4Bpvjt1v8vdeotjT1dm14tBrJn27tAToki/http%3A%2F%2Frepo.name)')) END AS GhRepo,

# CAST(created\_at AS DATE) AS date1,

CASE WHEN t1.created\_at IS NULL THEN CAST(SUBSTR(actor, 1, 10) AS DATE) ELSE CAST(created\_at AS DATE) END AS date1,

JSON\_EXTRACT(t1.actor, '$.[actor.id](http://secure-web.cisco.com/1q0S_i9lnlFaTS24gR8uuQcEwglM7CQt9vk3fwF4ymSUNfMc5Fz5EnBiaLuZluzhW3DX2ku1MhuQGvm_QfWAVcVdo9DlgmNtPCso2xAL-g1P5WjJKGzUY6MBn4yHMyhDopq4frslBnkw1m56bEOYW9vZHfgW2vYRbC5WOsGuc9wzp3lejFKrA0ZfbhpzLpV3ElaJH0oXRk9xoLlnR6pAdLBc-eWh0393QXXi-GwcvnqPxN1Us0f6P0i4f8EzB0wN_Aa32KElJ-My6QnTY2rXetqtyx3-zR4KLMBzs9oOH3c4VKeXUcHdn1x4gJ2JEnrWOMW3ieHag2wqmX7U2L8oNu_R37KVXeRFlR8n-5wy2XftL05eUMqOsX59WP-tguWCikMAhN2Q4UPEZqq7uSqE7kWSuKcZebmLAqtECB35yXKyqkFRY-_nKK5CM0tR6q3DP/http%3A%2F%2Factor.id)') AS actor\_id,

JSON\_EXTRACT\_SCALAR(t1.actor, '$.actor.login') AS actor\_login,

COUNT(JSON\_EXTRACT(t1.actor, '$.actor.login')) AS num\_activities,

COUNT(DISTINCT CASE WHEN type = 'PushEvent' THEN unique\_id ELSE NULL END) AS num\_dist\_commits,

COUNT(DISTINCT CASE WHEN type = 'CommitCommentEvent' THEN unique\_id ELSE NULL END) AS num\_dist\_commitcomments,

COUNT(DISTINCT CASE WHEN type = 'PushEvent' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_pushevents,

COUNT(DISTINCT CASE WHEN type = 'CommitCommentEvent' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_pusheventscomment,

COUNT(DISTINCT CASE WHEN type = 'PullRequestEvent' AND JSON\_EXTRACT(payload, '$.action') = '"opened"' THEN JSON\_EXTRACT(payload, '$.number') ELSE NULL END) AS num\_dist\_pullreqopened,

COUNT(DISTINCT CASE WHEN type = 'PullRequestEvent' AND JSON\_EXTRACT(payload, '$.action') = '"closed"' THEN JSON\_EXTRACT(payload, '$.number') ELSE NULL END) AS num\_dist\_pullreqclosed,

COUNT(DISTINCT CASE WHEN type = 'PullRequestEvent' THEN JSON\_EXTRACT(payload, '$.number') ELSE NULL END) AS num\_dist\_pullreqAll,

COUNT(DISTINCT CASE WHEN type = 'PullRequestReviewCommentEvent' THEN unique\_id ELSE NULL END) AS num\_dist\_pullreqcomments,

COUNT(DISTINCT CASE WHEN type = 'PullRequestEvent' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_pullreq,

COUNT(DISTINCT CASE WHEN type = 'PullRequestReviewCommentEvent' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_pullreqcomment,

COUNT(DISTINCT CASE WHEN type = 'PullRequestEvent' AND JSON\_EXTRACT(payload, '$.action') = '"opened"' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_pullreq\_opened,

COUNT(DISTINCT CASE WHEN type = 'PullRequestEvent' AND JSON\_EXTRACT(payload, '$.action') = '"closed"' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_pullreq\_closed,

COUNT(DISTINCT CASE WHEN type = 'IssuesEvent' AND JSON\_EXTRACT(payload, '$.action') = '"opened"' THEN JSON\_EXTRACT(JSON\_EXTRACT(payload, '$.issue'), '$.number') ELSE NULL END) AS num\_dist\_issuesopened,

COUNT(DISTINCT CASE WHEN type = 'IssuesEvent' AND JSON\_EXTRACT(payload, '$.action') = '"closed"' THEN JSON\_EXTRACT(JSON\_EXTRACT(payload, '$.issue'), '$.number') ELSE NULL END) AS num\_dist\_issuesclosed,

COUNT(DISTINCT CASE WHEN type = 'IssuesEvent' THEN JSON\_EXTRACT(JSON\_EXTRACT(payload, '$.issue'), '$.number') ELSE NULL END) AS num\_dist\_issuesAll,

COUNT(DISTINCT CASE WHEN type = 'IssueCommentEvent' THEN unique\_id ELSE NULL END) AS num\_dist\_issuecomments,

COUNT(DISTINCT CASE WHEN type = 'IssuesEvent' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_issues,

COUNT(DISTINCT CASE WHEN type = 'IssueCommentEvent' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_issuescomment,

COUNT(DISTINCT JSON\_EXTRACT(t1.actor, '$.actor.login') ) AS num\_actors\_allevents,

COUNT(DISTINCT CASE WHEN type = 'IssuesEvent' AND JSON\_EXTRACT(payload, '$.action') = '"opened"' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_issues\_opened,

COUNT(DISTINCT CASE WHEN type = 'IssuesEvent' AND JSON\_EXTRACT(payload, '$.action') = '"closed"' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_issues\_closed,

COUNT(DISTINCT CASE WHEN type = 'ForkEvent' THEN unique\_id ELSE NULL END) AS num\_forks\_event,

COUNT(DISTINCT CASE WHEN type = 'ForkEvent' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_forks,

COUNT(DISTINCT CASE WHEN type = 'WatchEvent' THEN unique\_id ELSE NULL END) AS num\_watch\_event,

COUNT(DISTINCT CASE WHEN type = 'WatchEvent' THEN JSON\_EXTRACT(t1.actor, '$.actor.login') ELSE NULL END) AS num\_actors\_watch,

COUNT(DISTINCT CASE WHEN type = 'ReleaseEvent' THEN unique\_id ELSE NULL END) AS num\_releases,

(CASE WHEN type = 'ReleaseEvent' THEN payload ELSE NULL END) AS release\_payload

from `ethereum-project-383415.Data.AllData20132014` t1

GROUP BY GhRepo, date1, actor\_id, actor\_login, release\_payload

ORDER BY GhRepo, date1, actor\_id, actor\_login, release\_payload

1. Descriptive data analysis – Alina, Mariia
2. Hypothesis testing – Mariia
3. Manuscript writing – Mariia, Jiasun
4. Next meeting: July 9, 2023 3 PM EST

2023-06-04, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Data availability from 2014

To Do:

1. Export data from BigQuery – Mariia. Done

SELECT \* FROM `ethereum-project-383415.Data.Data2017`

\*\*

SELECT \*

FROM (SELECT ROW\_NUMBER() OVER (

ORDER BY GhRepo,date1,actor\_id

) as row\_num,\*

FROM `ethereum-project-383415.Data.Data2017`)

WHERE row\_num>=33000

\*\*

SELECT \* FROM `ethereum-project-383415.Data.Data2022`

where date1>='2022-07-01'

1. Transform data 2013 – 2014 into panel format (like 2015) – Alina
2. Next meeting: June 16, 2023 8 PM EST – Done.

2023-05-21, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. Data availability from 2014
2. Maturity stages:
   1. Forks: London, Paris, Berlin etc.

To Do:

1. Check for anomalies of num activities in 2014 and 2015
   1. Compare average num activities per day in 2014 vs 2015
   2. Build charts of num\_activities, num\_actors, num\_releases by day
2. Upload all data 2016-2023 within 7 days - Alina
   1. Trial expiring in 7 days
3. Check 2014 data:
   1. ‘ethereum/mining’ repo July 2014, August 2014 – Mariia – Done.

SELECT \*

FROM `githubarchive.month.\*`

where \_TABLE\_SUFFIX between '201407' and '201408' and repo.name= 'ethereum/mining'

* 1. commit 637af12
  2. Vlad Zamfir committed on Jul 30, 2014
  3. Link: <https://github.com/ethereum/mining/commit/637af125e187c98305c6d675c3126b6acc4f72f2#diff-5628360e38236abc1097edcc3903a5436f995e640d63d8094e8b7442d9f1372b>
  4. Or check by names: Gavin Wood, Jack

Script for 2014:

SELECT type, payload, repo, actor, created\_at

FROM `githubarchive.month.\*`

where \_TABLE\_SUFFIX between '201401' and '201412'

and org.login='ethereum'

Note: be aware of the max size of csv export is 10MB. If dataset is larger, data will be lost in export

Notes:

* Need to find event: - Done. <https://github.com/ethereum/tryethereum/commit/ccf5e55a3c88a1250ce08c6b3f42a10904751b5a>
* Need to find event: - Done. Events before 2013-12-24 were created in vbuterin github

<https://github.com/ethereum/pyethereum/commit/6654544ca7245fa0f501057a94e74c8e331af2b0>

Solution:

* Activity archives are available starting 2/12/2011.
* Activity archives for dates between 2/12/2011-12/31/2014 was recorded from the (now deprecated) Timeline API.
* Activity archives for dates starting 1/1/2015 is recorded from the Events API.

Source: <https://www.gharchive.org/>

1. Next meeting: May 28, 2023, 8 PM EST - cancelled

2023-05-11, Thu

Participants: Alina, Jiasun, and Mariia

Discuss:

1. Data beginning from Jan 1 2015
2. num\_dist\_issuesopened, num\_dist\_issuesclosed, num\_dist\_pullreqopened, num\_dist\_pullreqclosed, num\_actors\_issues\_opened, num\_actors\_issues\_closed - all zeros

To Do:

1. Check the beginning of development of Etherum – Jiasun; Done.
   1. Very beginning, Dec 2013 – Miami Bitcoin conference
   2. Active development early 2014 (Spring)
2. Calculate fields: num\_dist\_issuesopened, num\_dist\_issuesclosed, num\_dist\_pullreqopened, num\_dist\_pullreqclosed, num\_actors\_issues\_opened, num\_actors\_issues\_closed, num\_actors\_pullreq\_opened, num\_actors\_pullreq\_closed – Alina; Done
3. Check 2014 data:
   1. ‘ethereum/mining’ repo July 2014, August 2014 – Alina;
4. Next meeting: May 21, 2023, 8 PM EST

2023-04-23, Sun

Participants: Alina, Jiasun, and Mariia

Discussed:

1. No meetings in the next 2 weeks
2. Next meeting – May 11, 2023 – Mariia confirm
3. Meeting minutes from past meeting

To Do:

1. Hypothesis to add: Presence of bots signals about the quality (?) of the project.
2. Collect data from 2014 – In progress.
3. Test H1 – Mariia
4. Send next meeting invite – Done.

2023-04-09, Sun

Discussed:

1. Dataset

To Do:

1. Collect data from 2014
2. Run 4 hypotheses on 2015 data locally - R
   1. H1:
      1. Locally construct indicator: is\_core/is\_peripheral – Alina; Done
      2. Dummy: is\_user\_bot
      3. Run regression
   2. H2:
      1. Network data
   3. H3:
      1. Collect dates of releases:
         1. Payload->release->created\_at - done
         2. Save fields: created\_at, payload - done
      2. Num activities between releases
   4. H4:
      1. Network data
3. Continue collecting data after check

Resources:

* <https://dept.stat.lsa.umich.edu/~jerrick/courses/stat701/notes/sql.html>
* <https://www.freecodecamp.org/news/connect-python-with-sql/>