

Use Python for this.
will look much cleaner.

that

write a Python script,
able contains review id,
out any changes, and a
ve can use MATLAB or
to show the relationship
and date.

Zhimao Lin
Department of Computing Science
University of Alberta
 Edmonton, Canada
 zhimao@ualberta.ca

that end up

Don't use references on objects / subjects
According to previous work (2) ...

that

As soon as we generate the line diagram, we can identify the fluctuation of code review approval rate and sample some interesting points on the graph. Then, we can use the URL in the table to manually find out what actually happened during the code review activity. With this information, we can get some insight about why the code review approval rate changes.

VI. THREATS TO VALIDITY

The dataset of this project can be limited. We only have 11 GitHub repositories developed by two communities, Eclipse and Couchbase [1]. It provides data for 11 software systems, accounting for a total of 50,959 code reviews and 144,906 revisions [1]. Additionally, we have to manually evaluate some code reviews, which prevents us from generating a very objective conclusion.

- following questions after we finish this project:
- ~~What is the relationship between the code review approval rate and release deadline?~~
 - ~~Why the code review approval rate increases or decreases when the deadline is approaching?~~
- If a relationship is found in RQ1, why is there a change in approval rate ...

REFERENCES

since they are under pressure, resulting in the

This project will analyze the data provided on the Code Review Open Platform (CROP). CROP contains a dataset which includes all the code reviews in a certain time period of 11 open-source projects [1, 3]. Each code review records its time stamp and approval status [3].

you sure
that this is
approval w/o
changes? Does it
just record
approve/not approve
or does it
differentiate
between changes
requested
or not

There may some problems with Github API. Just double check with the release page. It can miss one or two.

→ How will you identify

release dates?

We can find the release date on the Github release page.

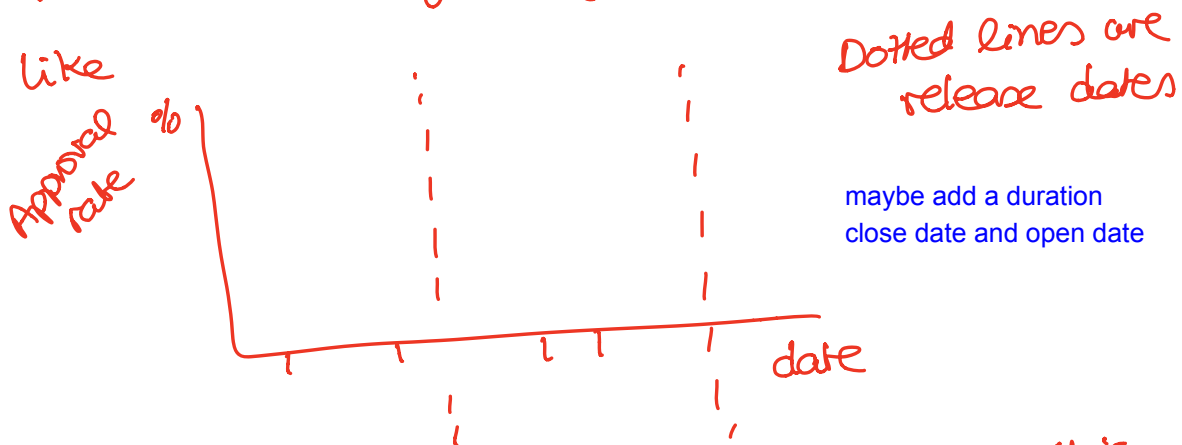
(need to clearly describe this in your methodology)

Define what is close to release date.

→ It might also make sense to compute if there is a correlation between "close to release date" and "approval status" you can use a chi-squared test for this. See ^{What is this?}

<https://datascience.stackexchange.com/questions/893/how-to-get-correlation-between-two-categorical-variable-and-a-categorical-variab>

→ In terms of your graph, I imagine something



Is this what you have in mind? Does this mean you are looking at all submitted reviews on a given day? or how exactly will you calculate approval rate? what is the denominator?

The number of code review in a day.

We can mine the data for the approval rate vs. number of reviews

| Date | How many code reviews are directly approved | Approved after changes | Rejected | Whether close to release date

Find the relationship between Approval rate W/o and Whether close to release date

Base on this table, it is easy to calculate the rate in any way that we want.

Questions:

Do I need to have issues/code review/test coverage/CI on our repo? -No