Git is popular distributed version control system, my team (20+ software engineers) use git to store all projects source code. I would like to explore the idea that whether I can measure engineers’ and projects’ contribution based on git commits and changes. I wrote a Python program to collect git commit (23K+ commits) from all my projects (6 projects use 50+ git repositories) and stored such info in MySql table with following key columns:

* user – user name;
* insert – number of insert operation for this commit;
* delete – number of delete operation for this commit;
* commit\_data – the time when this commit was made;
* project\_name – project name;

To measure contribution, I combined delete and insert to form a new column: change for easy comparison. I then output the results to csv file: gitwork\_mon.csv to be consumed by this D3 visualization project with following columns:

* user – user name;
* month – the month when the commit was made.;
* future\_store - change made for future\_store project;
* s2b - change made for s2b project;
* wumart-scm - change made for wumart-scm project;
* wz - change made for wz project;
* zt\_1 - change made for zt\_1 project;
* zt\_2 - change made for zt\_2 project;

This project follows interactive slideshow narrative visualization structure where user can check git change by 2 dimensions: by engineer and by project. 2 scenes are implemented respectively for this purpose. Using same CSS and layout, these 2 scenes share same template (month picker, D3 chart and legend) for visual consistency. Each scene has a month picker so that user can select which months of data she is interested to investigate (For simplification, I only collected data for 2018). This month picker shows the month from January till current month. Once the selection changed, the chart and legend will be refreshed with the appropriate data. User can change scene by click the tab “By Engineer” and “By Project” located on the top of the layout.

The scene “By Engineer” shows git changes made by individual engineer using bar chart. Since one engineer can work on multiple project, stacked bars are implemented, and different project is represented by different color. The metaphor is to use bar length to represent change volumes. Such design gives user an easy way to compare the contribution among engineers.

The scene “By Project” wants to compare the change volumes among projects and how much contribution an engineer made for a specific project. For this purpose, I implemented packed pie chart. Using size of circle to demonstrate the change volume and in one circle/project, use size of pie to differentiate the contribution among engineers. As you can see, the chart is very well suited for this purpose.

2 annotation approaches are implemented in this project: tooltip and legend. When mouse over a bar or pie, a tooltip with info: “Engineer”, “Project” and “Changes” will be shown to give user more info about this glyph. Legend is located on the right of the page to demonstrate which color is used for which project/engineer. These 2 annotations share same template for visual consistency.

2 set of parameters are used in this project: months and page tabs. Months is the parameter to filter in the data that the user is particularly interested. This parameter is global and controls the chart of both scenes. The chart will be refreshed once this parameter is changed. Page tabs control which scene is currently visible to the user.

3 triggers are implemented to response to user’s input: month selection checkbox change, page tabs click and mouse over/move/out a bar/pie chart. The first 2 triggers are connected to the 2 parameters mentioned above. The 3rd controls the show and hide of tooltip. D3 “on” function is used to handle these user input.

Files for this project,

* git.html – main file;
* gitwork\_mon – data file;
* month.js – month picker;
* month.css – style file for month picker;