Oregon State University, School of EECS

CS 519 – Information Visualization

Version Control History Visualization

Ajinkya Patil and Pratik Krishnamurthy

Problem Description

In Open Source projects, large number of commits are made over time. Software Evolution Researchers studying different version control systems would need to analyze this data to draw conclusions and answer their research questions. Manually analyzing such data can be cumbersome. We present this visualization system to help researchers analyze such data easily.

Target Audience

Software Evolution Researchers.

Description of Data

- Commit ID – Nominal (Hash values)
- Commit Date – Interval (UNIX timestamp)
- Author – Nominal (John, Alex,…)
- Lines of Code (LOC) – Quantitative (100,55000,…)
- Source Lines of Code (SLOC) – Quantitative (5,7,10,…)
- Version Control Name – Nominal (Git or SVN)

Questions & Abstract Operations

❑ Is there a relation between the date and the commit frequency (number of commits) and commit size (LOC)?

Q1.1: How does the commit size and frequency vary with time throughout the project?

➢ **Abstract operations**: Retrieve the date, type of VCS and ratio (LOC / number of commits). Compare the ratios (LOC with the number of commits) for all the dates.

❑ Do developers commit more often through Git or SVN?

Q2.1: How many commits were made each day?
Q2.2: On what days did developers commit through GIT and SVN?

➢ **Abstract operations:** Retrieve the number of commits and type of VCS for each date. Compare the number of commits on each date.

❑ Is there a relation between LOC and SLOC over time?

Q3.1: How does LOC and SLOC vary with date throughout the project?

➢ **Abstract operations:** Retrieve the VCS type, ratio (LOC / SLOC) and date. Compare the ratios for all the dates.

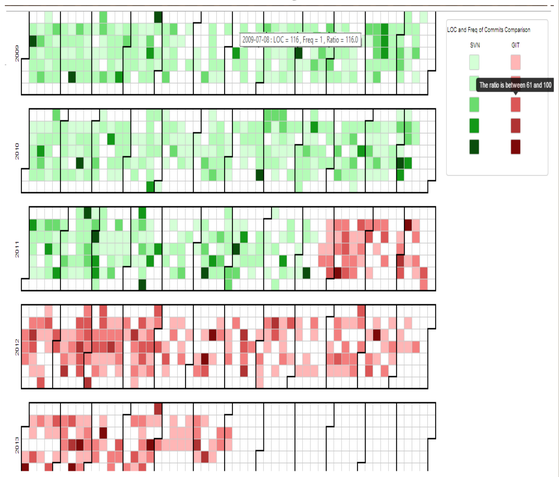
Encoding

Q1 – We represent the relationship between the number of commits and the Lines of Code (LOC) over each date as a ratio (LOC / number of commits) and this is the most important data for this question, followed by date and type of VCS. Ratio and Date are both quantitative variables while type of VCS is categorical. Using position, length, area, volume to encode a ratio doesn't make sense because the small variations in decimal values would not be accurately represented. Thus, we decided to represent ratio with 'Color Density' as it will help us compare the small changes in ratios over time. We chose 'Position' to encode Date because it is first on Mackinlay's ranking [1] for Quantitative data. We chose 'Color' for type of VCS as it is second on Mackinlay's ranking for categorical data [1].

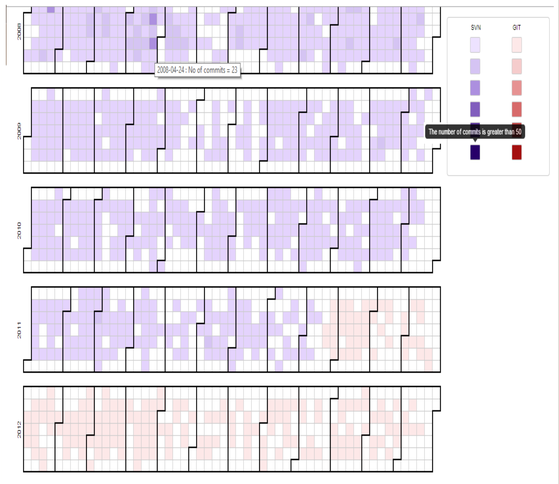
Q2 - Number of Commits is the most important variables in this question followed by Date and type of VCS. Date is Quantitative data and we represent it using position. Type of VCS is categorical and we represent it using 'Color' as it ranks second on Mackinlay's Ranking [1] for Categorical data. Number of commits is quantitative and it makes sense to choose Density for comparing this data over time to accommodate differences in the count.

Q3 - We express the relationship between LOC and SLOC using a Ratio (LOC / SLOC). Ratio is the most important variable followed by Date and type of VCS. Ratio is quantitative and we choose 'Density' to encode it since we want to compare the ratio over time. Type of VCS is categorical and we choose 'Color' to encode it. Date is quantitative and we choose to encode it using 'Position'.

Visualization for Q1



Visualization for Q2



The first visualization shows the variation of commit size and frequency throughout the project. The user can find out the variation looking at the color densities.

Similarly, the second question shows the number of commits on each day and the VCS used to make the commit.

Design Justification

Clustering methods provide a basis for data aggregation. Clustering relates to partitioning a data set into subsets exhibiting a certain similarity. The clustering process also provides an abstraction of the data. Concentrating on the clusters, rather than on individual data values, allows for an analysis of data sets with a much larger number of tuples.

A technique specifically designed for the analysis of clustered time-oriented data is the Cluster Calendar View.

The Cluster Calendar View facilitates the comparison of cluster representatives (overview), exploration of the values of a single cluster representative (abstract detail), and exploration of daily and monthly values of interest (specific details). Cluster affiliation is presented indirectly by color coding.

Hence, this visualization is perfect to answer the research questions since our data is spread across time spanning many years. It accommodates our encoding decisions and has a compact and extensive structure thereby allowing the user to differentiate and spot differences in the densities (ratio) across time (date with position encoding).

Interactions

We included tool-tip interactions over the legend for the visualization (to help interpret the color densities) and over each date in the calendar view (indicating the details for each date). A user can hover over these to get additional information. The legend has been intentionally kept fixed even if the user scrolls for ready reference.

References

[1] S.K. Card and Jay. Mackinlay, "The structure of the information visualization design space".
[2] https://github.com/mbostock/d3/wiki
[3] http://bost.ocks.org/mike/
[4] Aigner W et. Al, "Visual Methods for analyzing time-oriented data".