Mini Project 1 Report – Git(Hub) Viz

|  |  |  |
| --- | --- | --- |
| Student Name | Calvin Tan | Muthiah Nachiappan |
| Matriculation Number | A0040230H | A0094582W |

# Introduction

This visualization intends to provide a quick overview of the activities within a GitHub repository. Specifically, it focusses on

1. the comparison between the number of commits by collaborators and owners for the past 52 weeks,
2. the number of commits during different times of the day, and
3. the preferences in programming languages within an owner’s repositories.

The data for creating this visualization was readily available through GitHub’s API. For demonstration purposes, we used repositories from the owner ‘torvalds’, especially the ‘linux’ repository. Calvin handled the extraction and formatting of the data into csv files using Python script. We decided to use Tableau for creating the 3 visualizations. Nachiappan came up with 2 of the visualizations and Calvin handled the other.

# Visualizations – Purpose & Method

## Types of visualization

|  |  |
| --- | --- |
| **Objective** | **Visualization** |
| 1 | Line chart |
| 2 | Heat map |
| 3 | Bar plot, Tree map |

## Visualization 1 - Commits by Other Collaborators vs Commits by Owner

The following GitHub API was used:

https://api.github.com/repos/:owner/:repo/stats/participation

The following Python script was used to extract and format the data:

r = requests.get(<https://api.github.com/repos/torvalds/linux/stats/participation>')

data = r.json()

output = pd.DataFrame(index=range(52), columns = ["total","owner", "others"])

output.loc[:,"total"] = data['all']

output.loc[:,"owner"] = data['owner']

output.loc[:,"others"] = output.loc[:,"total"] - output.loc[:,"owner"]

output = output.sort\_index(0, ascending = False)

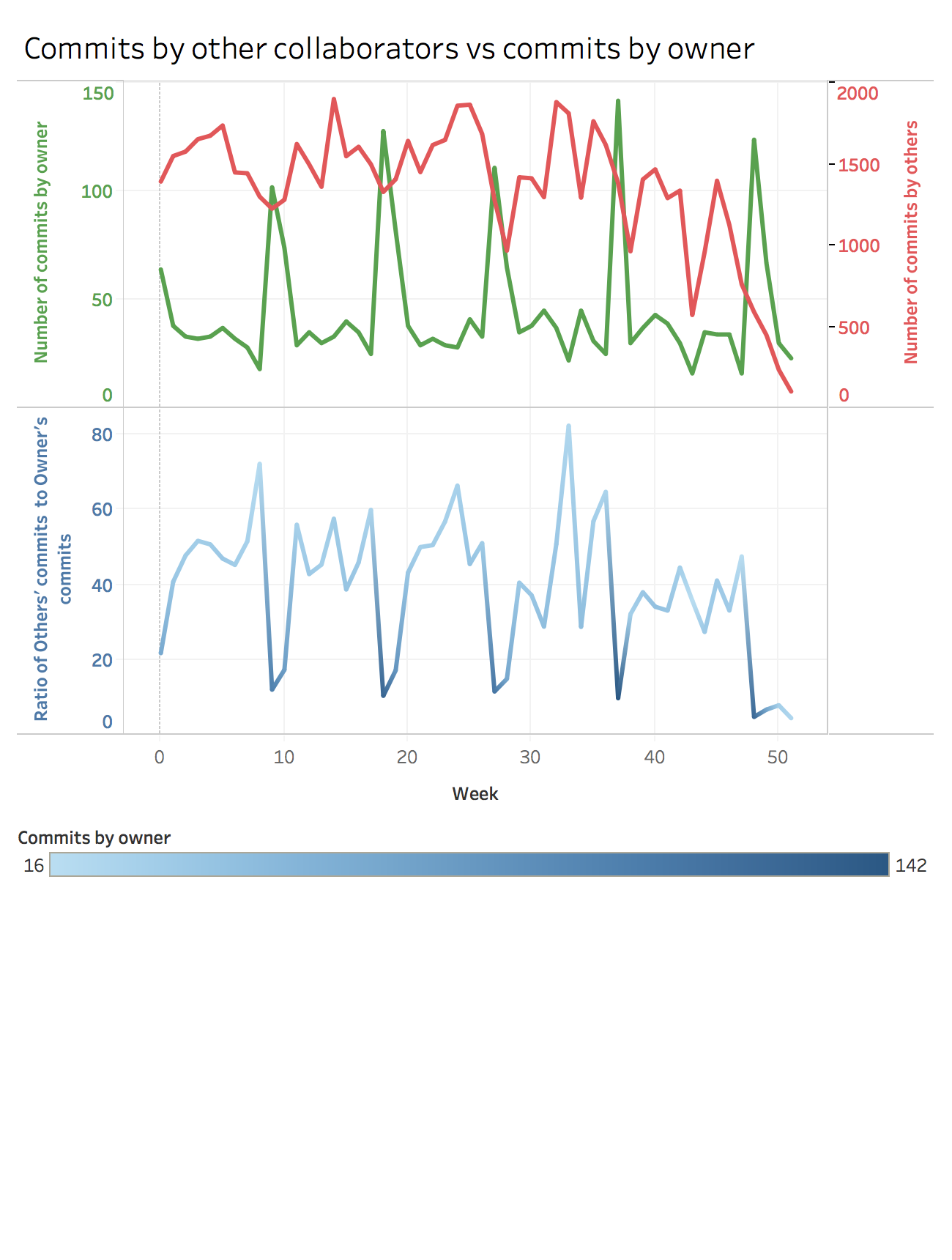
output.to\_csv("qn1\_rawdata.csv", index = True)

Resulting format of the data is a table with:

* Week (0 to 51)
* Total number of commits
* Commits by owner
* Commits by others

Further pre-processing was done to calculate “Others/Owners” by dividing number of commits by others by number of commits by owner.

This visualization is a 2-part line chart. A line chart was selected due to the temporal nature of the data. The week is shown as the horizontal axis for both parts. On the first part, the number of commits by the owner and others are shown together along the vertical axis. Owner’s commits are shown in green and others’ commits are shown in red. On the second part, the ratio of others’ commits to owner’s commits is shown along the vertical axis. The ratio line is coloured according to the number of commits made by the owner for correlation purposes. The visualization achieved is shown below:



We felt that the ratio is an important derived value because we can see how much more commits were made by other collaborators throughout the year. Interestingly, every 9 weeks, the owner made more commits and the ratio dipped. However, throughout the year, the owner’s commits were always less than the collaborators’.

## Visualization 2 - Number of Commits Across Days and Working Hours

The following GitHub API was used:

https://api.github.com/repos/:owner/:repo/stats/punch\_card

The following python script was used to extract and format the data:

r = requests.get(https://api.github.com/repos/torvalds/linux/stats/punch\_card')

data = r.json()

total\_rows = len(data)

output = pd.DataFrame(index=range(total\_rows), columns = ["day", "hour", "commits"])

for x in range(total\_rows):

for col in range(3):

output.iloc[x,col] = data[x][col]

output.to\_csv("qn2\_rawdata.csv", index = True)

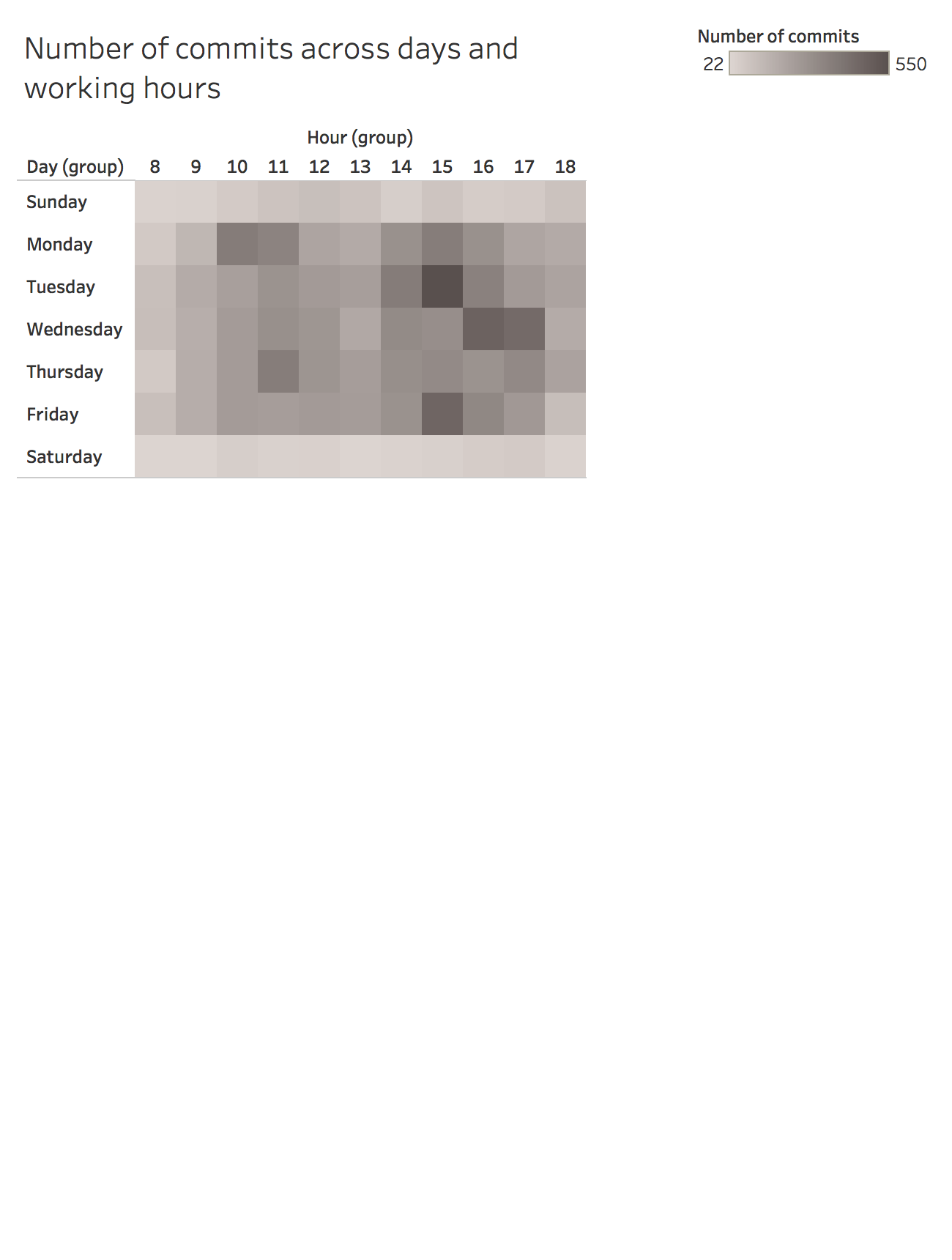
The resulting format of the data is a table with:

* Day (encoded from 0 to 6)
* Hour (0 to 23)
* Number of commits

Further pre-processing was done to create the following groups:

* Number of commits by day (‘Sunday’ to ‘Saturday’)
* Number of commits by hour (filtered to only include data from 8 to 18)

The heat map visualization was selected in Tableau because the number of commits (quantitative measure) is spread across two discrete time dimensions. This allows the user to observe commits within days and across the week. The days were distributed along the vertical axis and the working hours were distributed along the horizontal axis. Number of commits was encoded using a single colour scale. Mouse-over shows the exact number of commits. The visualization achieved is shown below:



## Visualization 3 – Byte Count Per Language Per Project for User (Torvalds)

The following GitHub APIs were used:

https://api.github.com/repos/:owner/:repo/languages

Here, the owner is ‘torvalds’ and repo is each of the 6 repos owned by Torvalds.

After extracting the required information, the following python script was used to format the data:



The resulting format of the data is a table with:

* name of repository
* language
* byte count

Further preprocessing was done to calculate the following:

* Total byte count per project
* Proportion of byte count per language per project
* Log of byte count

Using Tableau, a total of 2 visualizations were created to answer this question.

1. Log Byte Count by Project

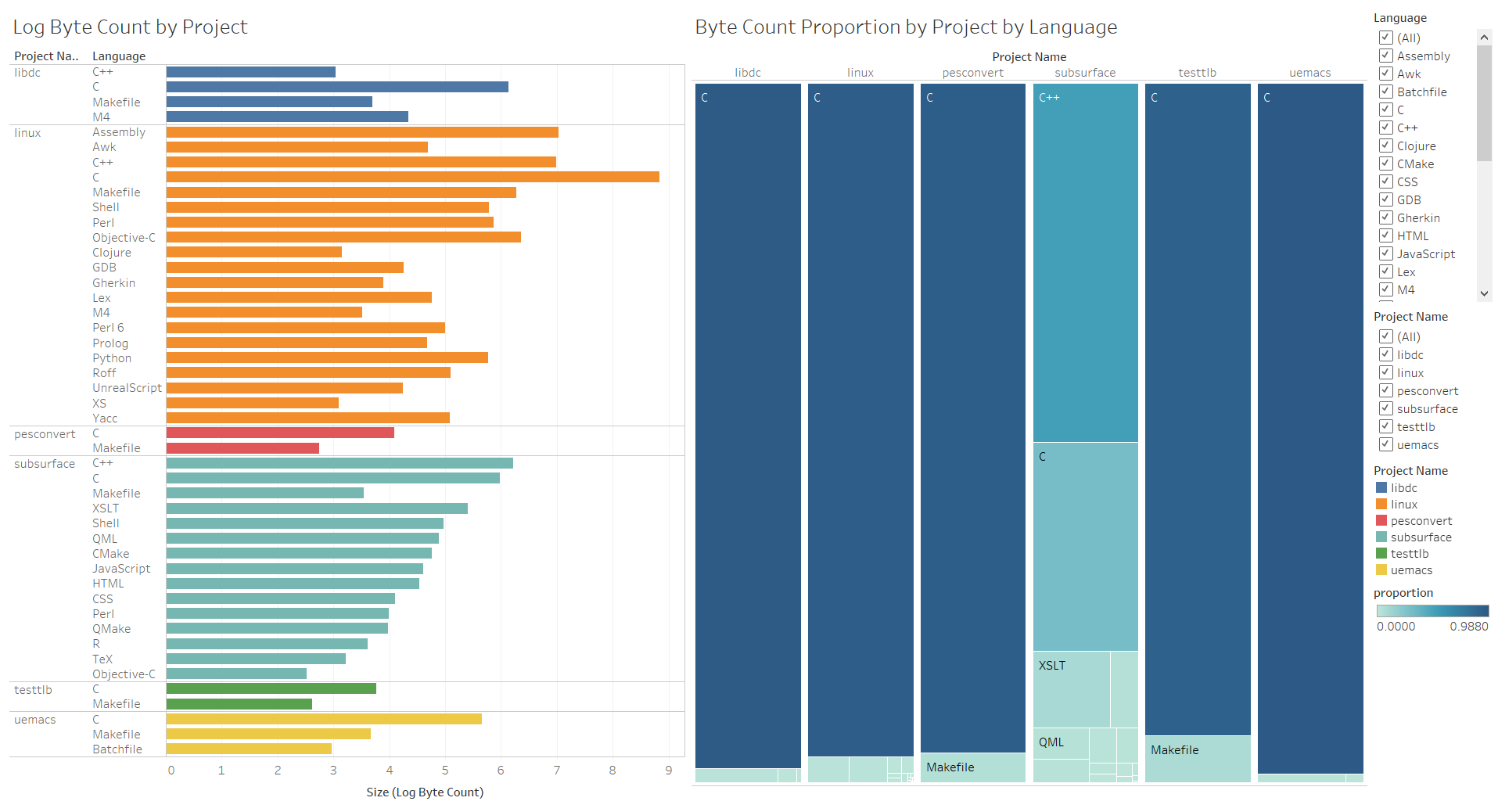
This visualization shows the languages used in each project and the corresponding byte count. Byte count was shown using the log scale because one of the project’s size is significantly larger than the others. Hence, a log scale was better suited. Language names are arranged in alphabetical order. A horizontal bar chart was used due to the large number of languages involved.

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Data Type | Encoding | Note |
| Project name | Categorical | Position, Colour | Each project forms an outer group on the y-axis and is colour coded |
| Language | Categorical | Position | Each language, nested within each project, is illustrated on each row |
| Log byte count | Quantitative | Length | The length of the bar shows the log byte count per language per project |

1. Byte Count Proportion by Project by Language

A treemap visualization was chosen to illustrate the proportion of language used in each project by byte count. It allows the user to identify the dominant languages used in each project and how that compares across projects. Mouseover shows the exact proportion figure and language name.

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Data Type | Encoding | Note |
| Project name | Categorical | Position | Each project occupies a vertical space |
| Language | Categorical | Position | Each language is represented as a separate box within each project |
| Proportion of byte count per project | Quantitative | Area, colour | The proportion of language used per project is illustrated using both area and colour. High proportions correspond to large areas and deeper blue colour. |



Due to the large amount of data, to facilitate exploratory analysis, filters were used to aid in performing the following tasks:

* Analyze and filter data based on project name
* Compare languages used across 2 or more projects using the project name filter
* Identify which projects use a language or set of languages
* Identify anomalies in terms of language usage