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

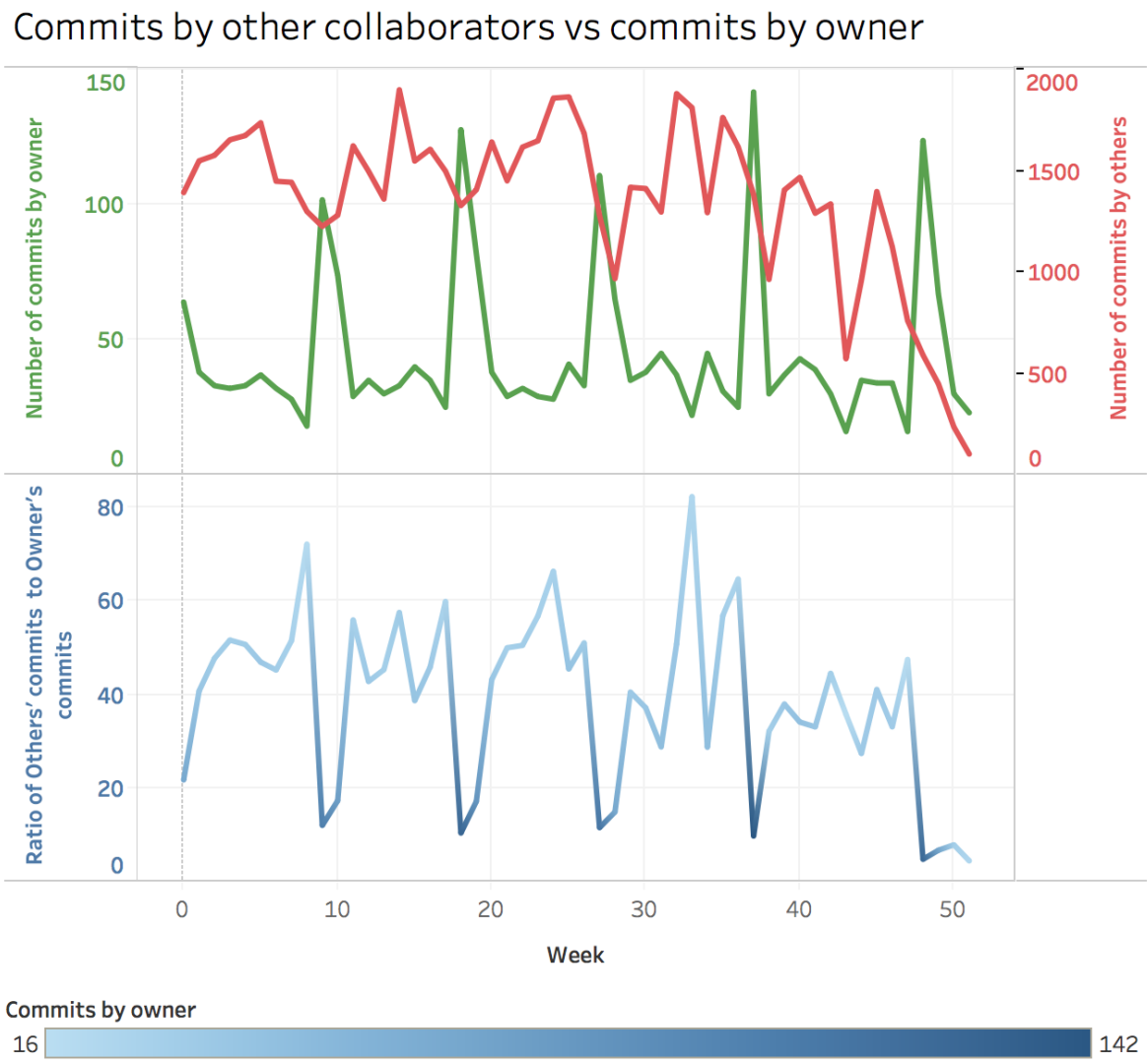
Visualization 1 - Commits by other collaborators vs commits by owner

The following 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)
```

Using Tableau, a calculated field, "Others/Owners", was created by dividing 'others' column by 'owners' column. The index (0 to 51) which represents the weeks were set to be the horizontal axis of the line chart. On the top half of the line chart, the number of commits by the owner and others were shown together using the dual axis feature. The owner commits line was coloured green and others' commits line was coloured red. On the bottom half, the ratio of others' commits to owner's commits across weeks was shown. The line was coloured

according to the number of commits made by the owner. The visualization achieved is shown below:



We felt that the ratio was an important component in showing 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 less than the collaborators'.

Visualization 2 - Number of commits across days and working hours

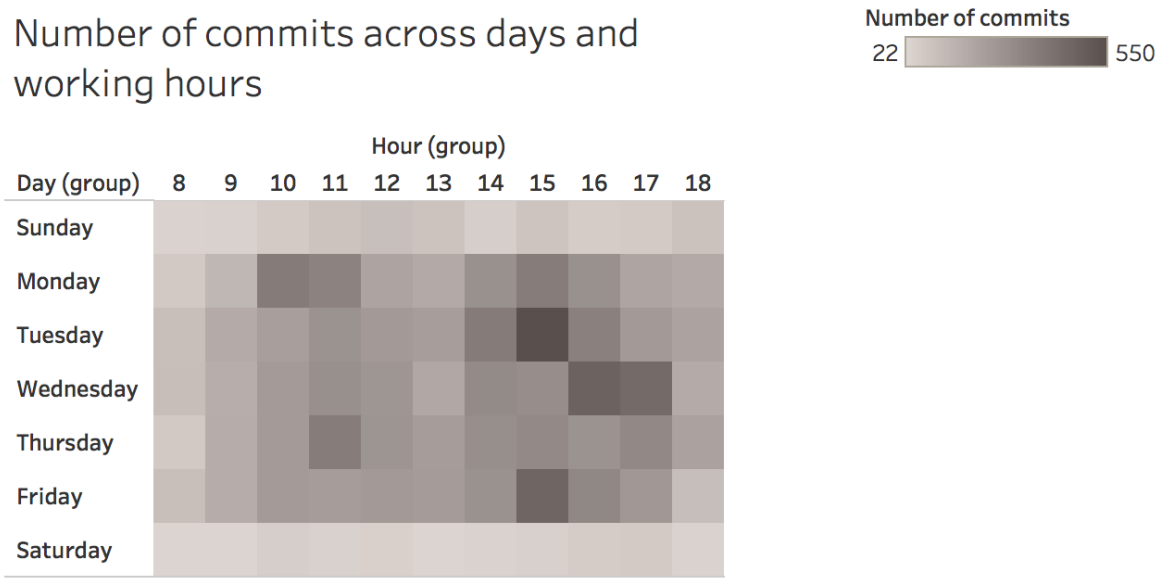
The following python script was used to extract and format the data:

```
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)
```

Using Tableau, the data was grouped according to day into the corresponding day's string name. Using the day group, hour attribute and commits, the heat map visualization was selected. A filter was applied to the hour attribute to only include commits from 8am to 6pm. The days were distributed along the vertical axis and the working hours were distributed along the horizontal axis. A grey-toned single colour scale was included to show how the number of commits were distributed across the days and hours. 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

Where owner = 'Torvalds' and repo = the name of the 6 repos owned by Torvalds.

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

```
#read in data
libdc_path = "data_q3_libdc-for-dirk.json"
linux_path = "data_q3_linux.json"
pesconvert_path = "data_q3_pesconvert.json"
subsurface_path = "data_q3_subsurface-for-dirk.json"
testtlb_path = "data_q3_test-tlb.json"
uemacs_path = "data_q3_uemacs.json"

libdc = json.load(open(libdc_path))
linux = json.load(open(linux_path))
pesconvert = json.load(open(pesconvert_path))
subsurface = json.load(open(subsurface_path))
testtlb = json.load(open(testtlb_path))
uemacs = json.load(open(uemacs_path))

repo_trend = pd.DataFrame(index=range(50), columns=["name","language","size"])

repo_list = [libdc, linux, pesconvert, subsurface, testtlb, uemacs]
repo_dict = {0:"libdc", 1:"linux", 2:"pesconvert", 3:"subsurface", 4:"testtlb", 5:"uemacs"}
repo_dict[0]
row_count = 0
for i,repo in enumerate(repo_list):
    for k,v in repo.items():
        repo_trend.iloc[row_count,0] = repo_dict[i]
        repo_trend.iloc[row_count,1] = k
        repo_trend.iloc[row_count,2] = v
        row_count = row_count + 1
repo_trend.to_csv("qn3_rawdata.csv", index=True)
```

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

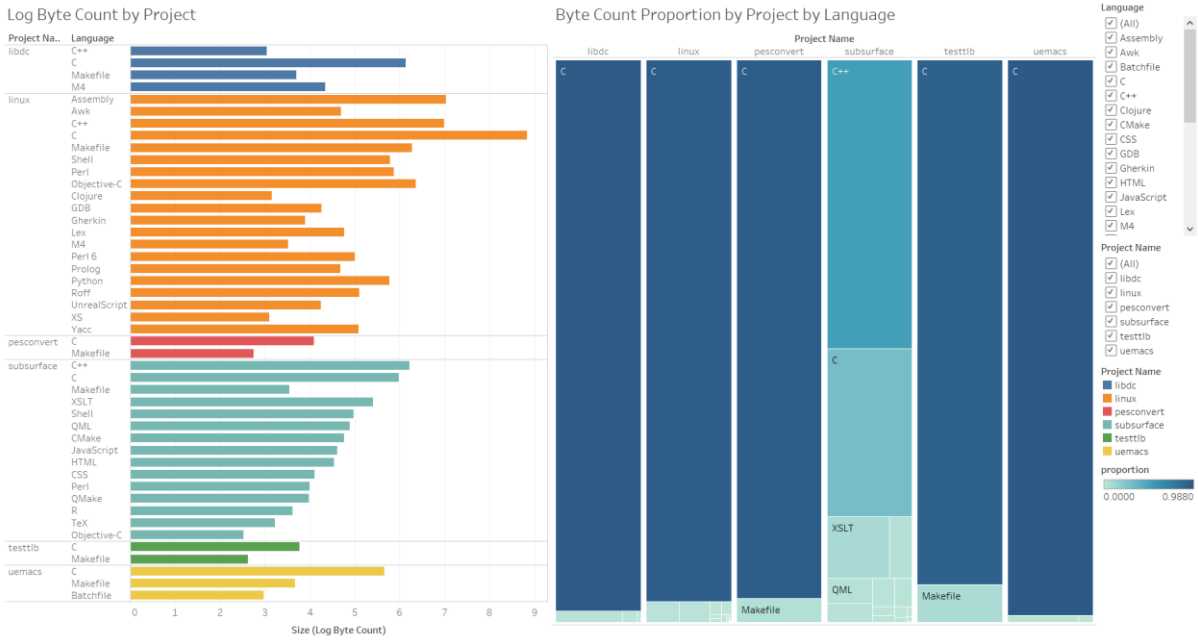
arranged in alphabetical order. A horizontal bar chart was used due to the large number of languages involved.

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

2) Byte Count Proportion by Project by Language

This is a tree map visualization used 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.



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