Grant Hicks

ECON860 Midterm

See repository's 'README' for details about the program & how data was obtained, split into parts 1-3, including both bonuses.

Datasets (csv) are included in the repository. html files scraped from charcoal also included – the first html (20231026081808) was the one used for part 1, the others were scraped in AWS EC2 for the bonus users. Program also scraped github API (json) & html – I did not upload these to the repository, as it would have been over a thousand files, but I can if requested.

Written report prompts follow below:

Part 1

– Summary statistics of the dataset you download.

Charcoal 2022 Data

	N	Minimum	Maximum	Mean	Median	Std. Dev.
Repo Count	448	0	8,082	137.62	42.5	536.71
Follower Count	448	0	12,816	422.27	72	1,290.54
Member Since	448	1/22/2008	8/1/2021	11/30/2010	3/18/2010	1,029.37

Charcoal html used for this summary was scraped on 10/26/23 at 8:18am ET.

See next question for more details about this summary data.

– Sample size of the dataset, number of unique login IDs, number of invalid login IDs, number of login IDs with invalid/missing information.

693 non-duplicate login IDs were returned from parsing the charcoal site. Of these 693 login IDs, 9 of them had invalid or missing information (5 users had -1 repo/followers count AND were missing 'member since', 1 user was missing 'member since', and 3 users had invalid dates, such as Feb 30).

When the unique user list from the charcoal dataset (parsed_files/dataset.csv) was fed into run_requests_API.py in part 2, only 448 of the 693 login IDs returned a valid github API, meaning 245 of the 693 unique charcoal IDs were invalid. The summary statistics shown above are for the charcoal data of only the 448 valid IDs (for part 2 comparison purposes).

– ”Bonus GitHub Data” section – reported separately from “GitHub Data” section

Program run_requests.py was run for 24 hours in AWS EC2, beginning at UTC 10/26/23 16:51:00. It saved one html file from the charcoalpaper exam link every 10 minutes.

Program run_parse_part1_bonus.py was run to parse the html files returned by the request for bonus user occurrences during the 24 hour period. Results summarized below.

scrape_time_date	scrape_time_UTC	Login_ID2
10/26/2023	23:01	misbahsy
10/26/2023	23:21	halit-vural
10/26/2023	23:41	yuhao-git-star
10/27/2023	3:01	misbahsy
10/27/2023	3:21	halit-vural
10/27/2023	3:31	esin
10/27/2023	3:41	yuhao-git-star
10/27/2023	3:51	ilyakatz
10/27/2023	5:01	misbahsy
10/27/2023	5:11	zishon89us
10/27/2023	5:31	esin
10/27/2023	5:41	yuhao-git-star
10/27/2023	8:01	misbahsy
10/27/2023	8:11	zishon89us
10/27/2023	8:21	halit-vural
10/27/2023	8:41	yuhao-git-star
10/27/2023	8:51	ilyakatz
10/27/2023	9:01	misbahsy
10/27/2023	9:41	yuhao-git-star
10/27/2023	10:01	misbahsy
10/27/2023	14:31	esin
10/27/2023	14:41	yuhao-git-star
10/27/2023	14:51	ilyakatz

Part 2

– Summary statistics of the dataset you download.

– The information you get in Part 1 is information of the users in 2022, while the information you get in this part is current information of the users. Compare the two datasets.

Github API 2023 Data (as of 10/28/23)

	N	Minimum	Maximum	Mean	Median	Std. Dev.
Repo Count	448	0	8,485	134.58	41	547.31
Follower Count	448	0	15,121	425.42	65	1,374.57
Member Since	448	1/22/2008	9/13/2023	4/18/2011	5/30/2010	1,179.78

Variance (Github API 2023 minus Charcoal 2022)

	N	Minimum	Maximum	Mean	Median
Repo Count	-	-	403	-3.04	-1.5
Follower Count	-	-	2,305	3.16	-7
Member Since (var in days)	-	-	773	140	73

Notably, several users 'member since' date increased in the 2023 github data when compared to the 2022 charcoal data. I would have expected all 'member since' data to be static since it is the same list of users.

I suppose user accounts could have been deactivated then reactivated under the same ID, resetting the date, as well as repo & follower data. This can explain our unexpected variance results in mean & median repo/follower data.

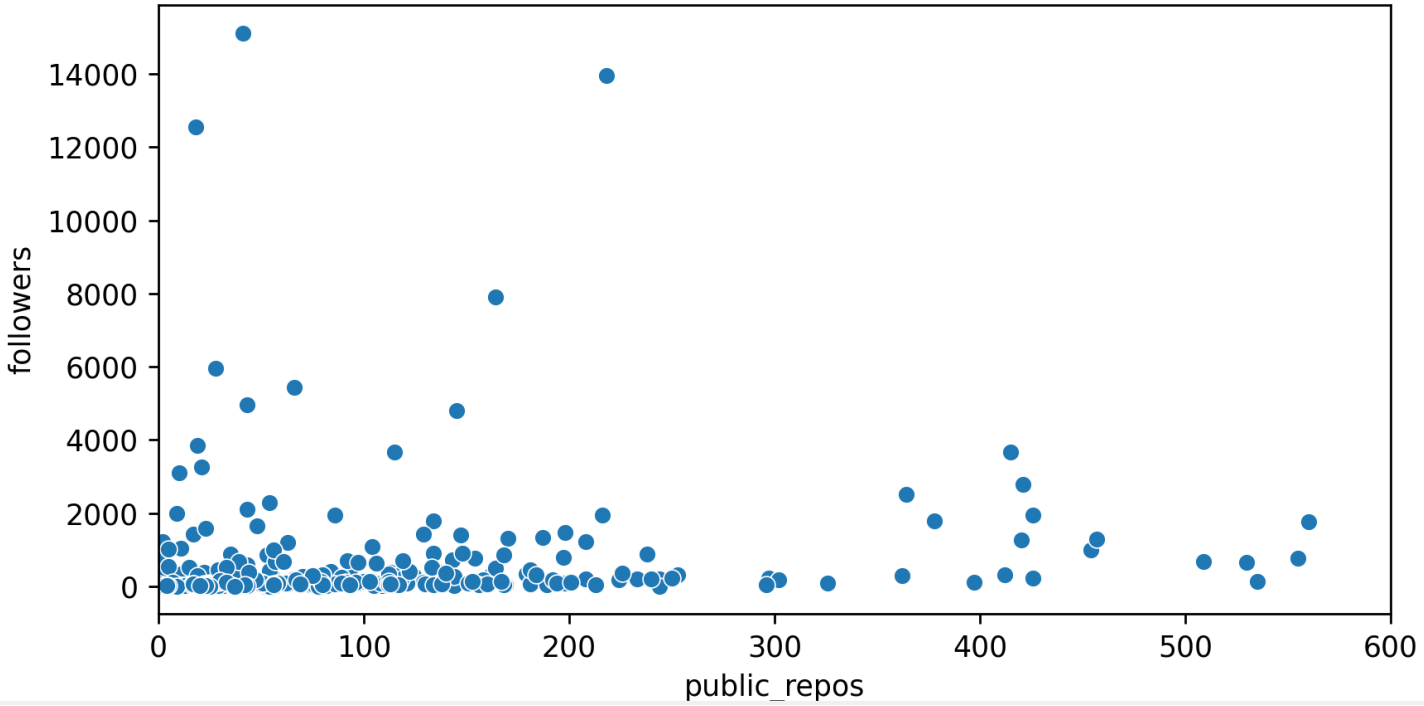
A few examples of user github data appearing to reset:

	Github API 2023			Charcoal 2022		
gh_id	**public_repos**	**followers**	**created_at**	**public_repos**	**followers**	**created_at**
iExk	0	0	2023-01-25	4	17	10/27/2009
j9h1	0	0	2017-05-30	197	32	9/12/2010
abnn	0	0	2015-11-04	27	37	4/30/2012
otzm	0	0	2023-09-13	132	111	11/1/2008

In all 4 of these examples, I have confirmed that the ID is currently a valid github account (with 0 followers/public repos), and I have confirmed that the exact IDs/data did appear in the charcoal html I scraped on 10/26/2023 at 8:18am ET.

When I look at the charcoal site on 10/30/23, none of these 4 users are present, so I suppose the charcoal dataset can change over time (aside from just the bonus data).

Part 3:



I plotted the relationship between public repositories & followers. I set the scatter plot program to limit the x value to 600, as there were only 10 users in the dataset with >600 repositories. I consider those 10 users to be outliers and do not think their data adds any value. Therefore, the sample size is 438.

I would expect a positive correlation between these two variables, as a larger collection of repositories would illicit a wider following. The plot supports this, though it would have a low coefficient of determination if a linear regression were run. This is reasonable, as the plot does not account for the content or quality of the repositories. One user may have only 2 repositories, both very robust, while another user may have 500 repositories, filled with nonsense. Additionally, social factors and other variables affect follower count.