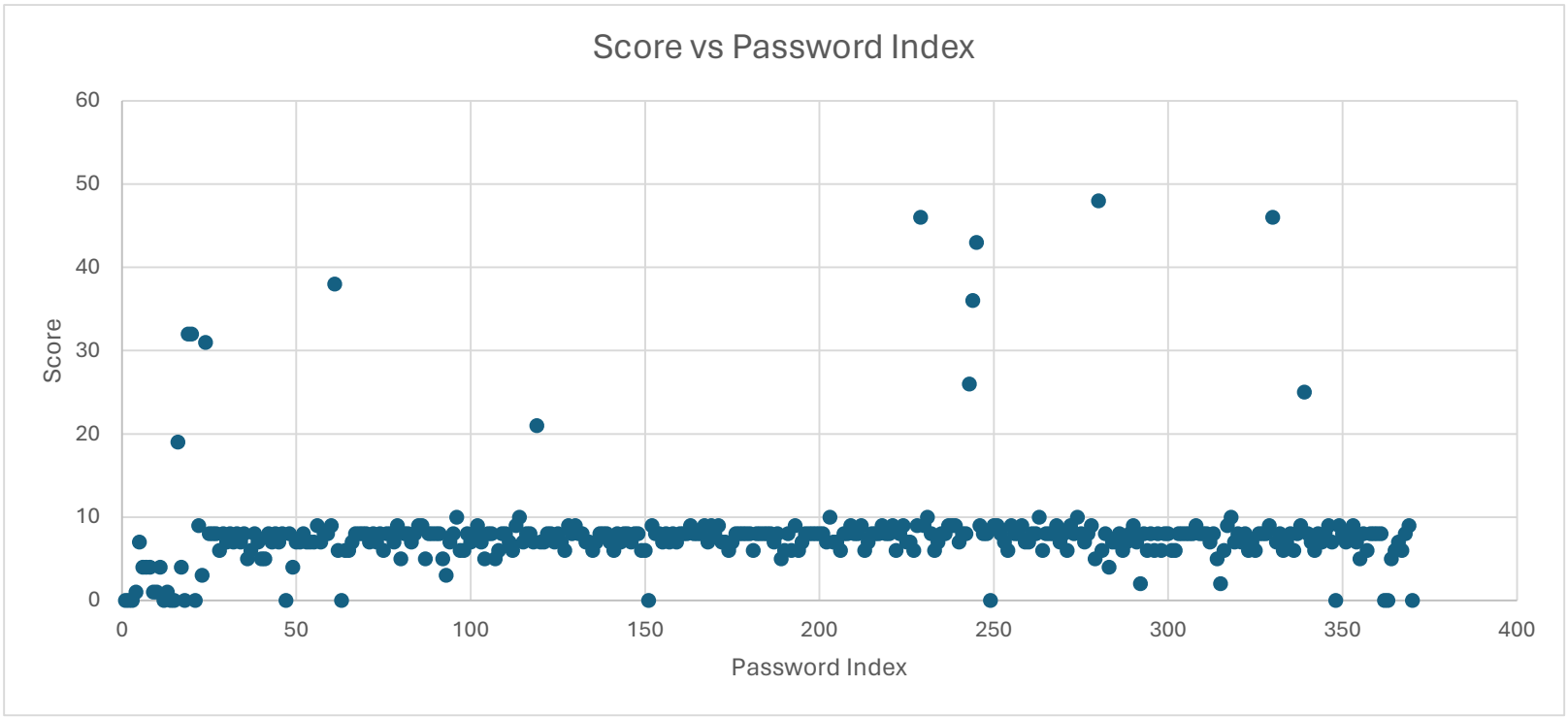
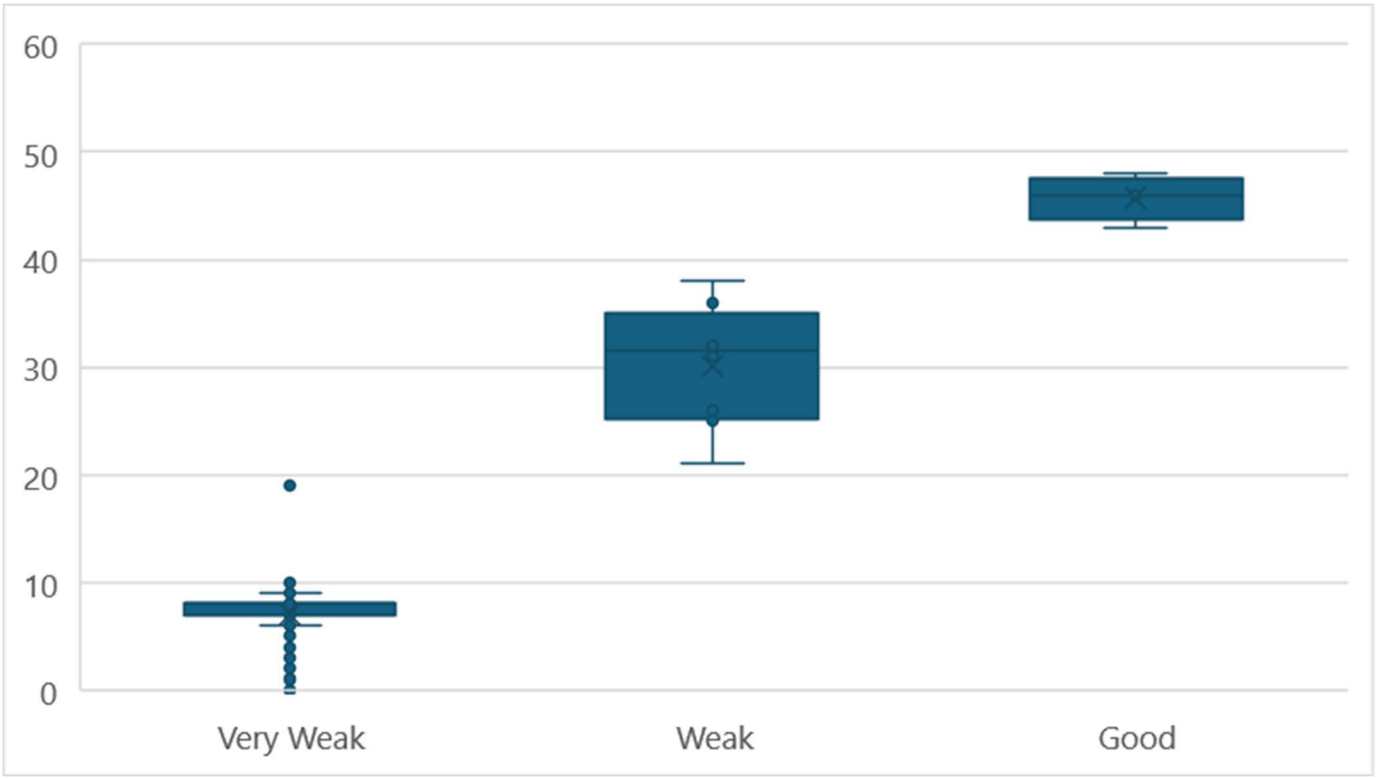
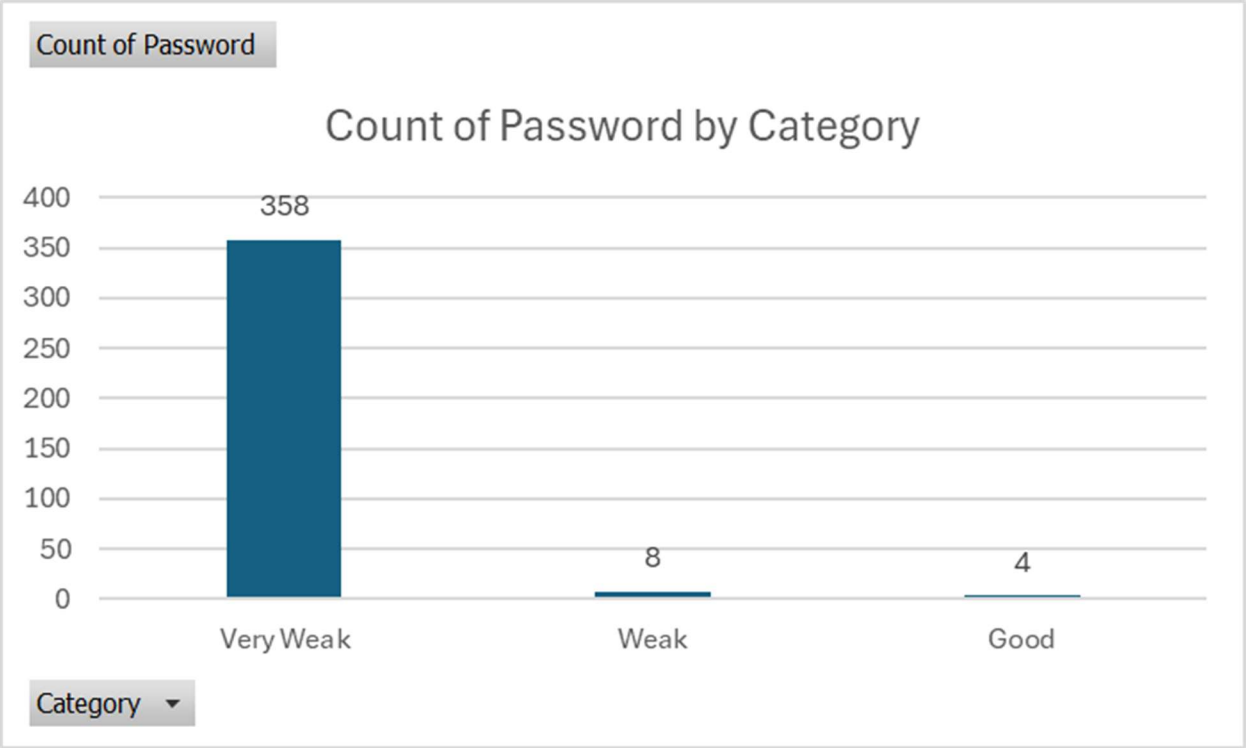
1.



Based on the requested chart, the score for most banned passwords were in the 7-9 range. I didn't feel like this chart was a great representation for the data (at least on its own), so here's a few more to look at.



This box and whiskers blot shows the distribution of data inside of the three categories. And the histogram on the next page shows how many passwords are in each category.

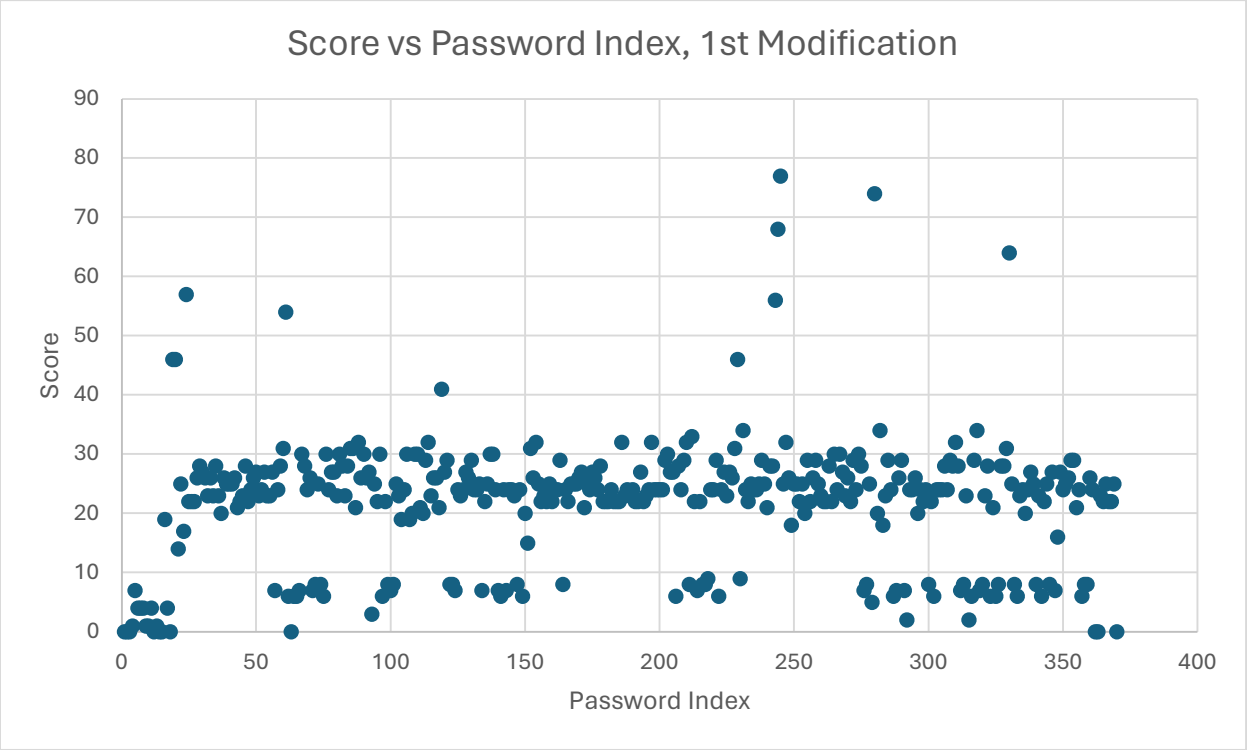


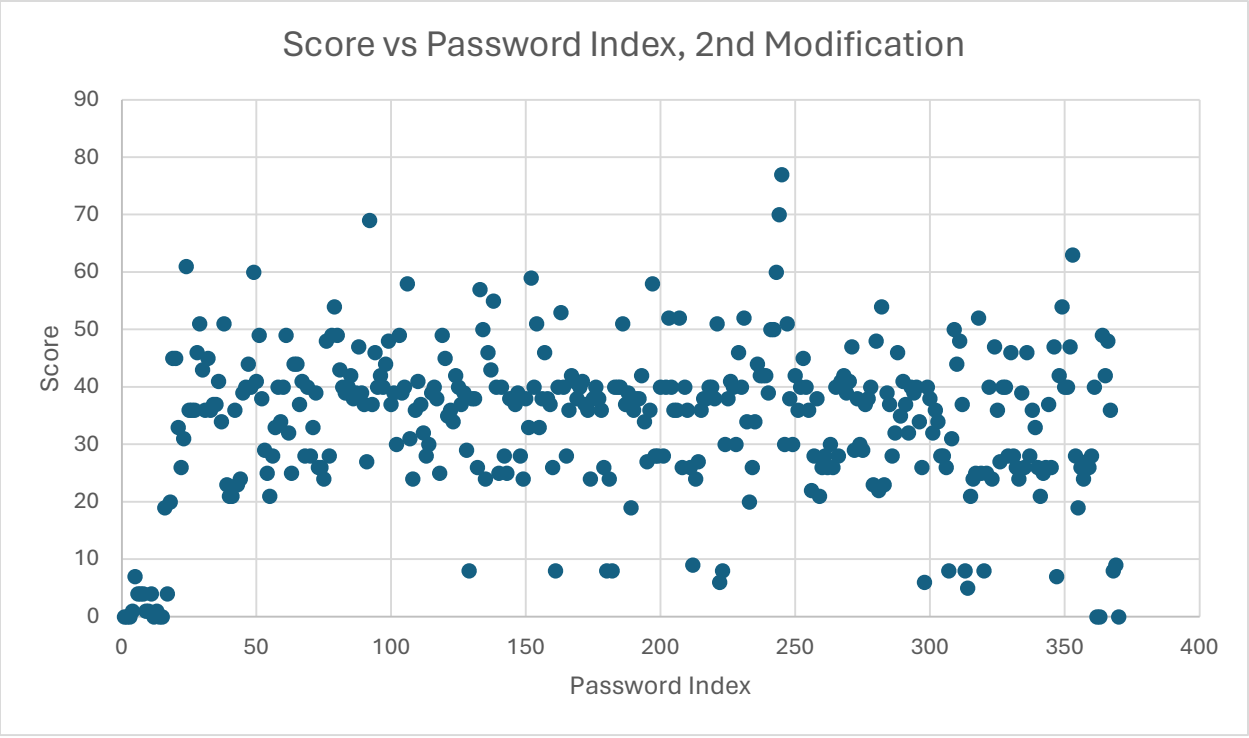
2. Factors that increased a password's score include the length of the password, the number of uppercase and lowercase letters, the number of numbers and symbols, the number of numbers or symbols in the middle of the password, and a bonus if all the requirements have been met. The requirements were that the password had to be at least eight characters in length and contain at least one character from three of the four following categories; uppercase letters, lowercase letters, numbers, and symbols.

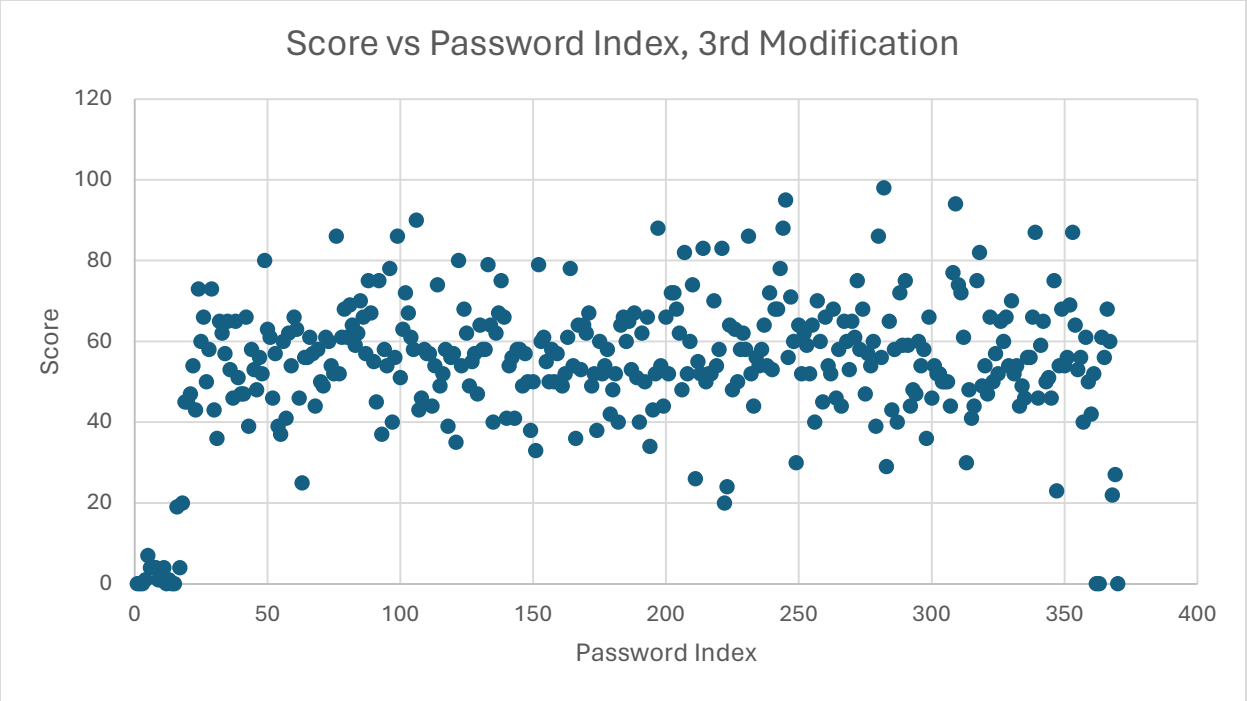
Factors that reduced a passwords score include whether the password was entirely letters or entirely numbers, whether there were any repeat characters, or having consecutive uppercase letters, lowercase letters, or numbers. There were additional penalties applied for passwords that contained sequential (defined as 3+) letters, numbers, or symbols.

The most interesting calculation is from the number of upper- and lower-case letters, as it is not simply a count of letters. Rather, it is based on the proportion of the total number of characters to the number of upper- and lower-case letters. As an example, a123 would receive 6 points for this category, whereas aa12 would receive only 4.

3. Charts as requested!







The score improved noticeably with each modification. The average of the unmodified password list was ~8, however the first modification (simple capitalization) resulted in an average score of almost 22. The third and fourth modifications continued to improve on this, with averages of 34 and 54 respectively. With all four data sets, number only passwords struggled (this is the clump of data in the bottom left of each chart). Outside of those instances, with each modification the passwords become more and more random looking, and had a better mix of the four categories of characters (uppercase letters, lowercase letters, numbers, and symbols). One thing that stood out was the rather unexpected success of “password123”. In all four data sets, it is one of the highest scoring passwords, starting with a score of 43 (good) in the unmodified dataset, and progressively improving. In the second dataset, it transformed into “PAsworD123” and had one of the first strong scores, clocking in at 77. It then scored another 77 (having transformed into “p@ssw0rd123”) in the third dataset, and a whopping 95 in the final dataset, having been transformed into “P@55W0RD123”. I highlight this mostly to point out that password strength scores are not infallible. Each of these passwords is likely to be well known and would be checked early with a dictionary attack.

4. JTR’s incremental mode is the classic brute force approach. It may take a long time to run, as it tries every possible combination of characters until it has found a result. The wordlist method is a dictionary attack. It compares the hash to a list of potential matches.

5.

Password File	Incremental Method	Wordlist john.txt Wordlist None Rules	Wordlist rockyou.txt Wordlist None Rules	Wordlist john.txt Wordlist All Rules	Wordlist rockyou.txt Wordlist All Rules
(1) - No change	345	315	369	341	369
(2) – Various letters capitalized	74	72	137	326	172
(3) - Small number of replacements	182	29	258	161	306
(4) - Large number of replacements	32	15	35	36	62

6. The first main takeaway from the experiment was that for the brute force attacks (the incremental method) capitalization was more effective than symbol replacement. Speculating for a bit, this likely has to do with the order that the different characters were utilized in generating a password. Another observation is that dictionary attacks were more likely to succeed when there were more passwords to choose from (a longer wordlist). Dictionary attacks were also more effective after swapping from no modification rules to utilizing all modification rules, although this came at the cost of run time. Finally, the large number of replacements file did reasonably well against all five attack methods.

7.
A) The two most used passwords are “june1097” and “t8Ky/<^mTB” and are rated as good and very strong respectively.

B) Speaking from personal experience, one reason is that it’s simply much easier to remember one good password that can be used on multiple sites than it is to remember a good password for each individual account. Additionally, people may use the same password because they either don’t know or they don’t care about the dangers of shared passwords.

C) I suppose that depends on one’s definition of “necessary”. People are certainly able to make accounts using common passwords, and frankly, the vast majority of people who do this will face no consequences as a result of it. However, adding a restriction on common passwords would increase account security. And at that point, it’s going to be a fight between customer service (presumably the folks who will have to deal with people being unable to access their accounts due to the increase in security measures) and the IT/cybersecurity office (the folks who will need to regularly mine for and restrict common passwords) in order to generate the cost-benefit analysis for this restriction.

8. The JTR analysis showed how vulnerable common and/or simple passwords are to even basic attacks. One thing that I took away was that different passwords took a different amount of time to crack, based on length and complexity. With the understanding that no password is truly uncrackable, security at a personal level can be approached from the idiom of running away from a bear. We don't have to be faster than the bear (the hacker in this case), we just need to be faster than the other guys. Hackers will spend their time exploiting easy to crack passwords, and by the time they can crack the more complex ones, the breach should have had time to be detected and sealed.

Based on the exercise, password strength is largely tied to how random the password is or at least appears to be. Utilizing common patterns or simple patterns as part of the password makes it far easier for attacks to succeed.