How Do Password Characteristics Affect Password Strength?

Zaid Muqsit and Zoe Spicer

Introduction

Research Question and Motivation

In our increasingly technology-oriented world, data security is a pressing and essential topic. As cybercriminals' hacking tools have improved, data leaks at major companies such as Yahoo, Facebook, LinkedIn, Mariott International, Adobe, Bank of America, British Airways, and CVS have compromised billions of users' personal information. In 2022, IBM found that the average data breach in the U.S. cost companies an average of $9.44 million in lost business, crisis management efforts, and ransom payments. Data breaches can also allow hackers to access users' personal information such as names, addresses, credit card details, and Social Security numbers, which can be used for financial fraud or identity theft. One critical aspect of data security is password strength, which can reduce the risk of cybercriminals guessing users' passwords and acesssing personal information. Given our interset in datasecurity and the topcality of password strength as a key facet of this subject area, we wanted to explore password data for our project.

Our research question is: How do various password characteristics affect password strength? We measure password strength in two ways: "strength" (which is calculated by an algorithm based on the password's length and complexity and is comparative to the generally bad passwords in the dataset) and the time the password takes to crack by online guessing.

Data Description

Variable Name	Type	Description
rank	numeric	Popularity in their database of released passwords
password	character	Actual text of password
category	categorical	Classification of type of password
true_val	double	Time to crack by online guessing standardized to seconds
true_val_strength	double	true_val made numeric where 11 is most crack time, 1 is lowest
offline_crack_sec	double	Time to crack offline in seconds
rank_alt	numeric	Secondary popularity rank in database of released passwords
font_size	numeric	Arbitrary font size Knowledge Is Beautiful used in graphic
strength	numeric	Quality of password where 10 is highest, 1 is lowest
pass_length	numeric	Length of the password
num_digits	numeric	Number of digits in the password

Variable Name	Type	Description
num_letters	numeric	Number of letters in the password
num_unique	numeric	Number of unique characters (letters or numbers in the password)

Our data come from Tidy Tuesday, originally sourced from Information is Beautiful, a design company that distills data into visualizations and infographics. Information is Beautiful acquired its data on passwords by deep-mining 20 separate data breaches in 2017, including breaches of Facebook, Sony, and Yahoo. The data only includes the 500 most popular passwords, which also tended to be low-strength. Therefore, the `strength` variable indicates password strength in relation to these generally weak passwords.

In the cleaning process, we removed the last seven observations, as all their values were "NA." We also removed observations that had a strength recorded over ten as those may have been miscalculations or strengths that were not standardized to values 1 through 10. From there, we were left with 485 observations. Additionally, we combined the `value` and `time_unit` variables into one time standardized to seconds called `true_val`. Previously, `value` referred to the time to crack by online guessing, and time unit was the time unit to match with that value (seconds, minutes, hours, days, months, or years). Based on `true_val`, we made a new variable called `true_val_strength` for use in ordinal regression. This variable translated `true_val` values to numbers 1-10, since `true_val` values were not actually continuous but rather discrete values (2.17 years, 0.00321 days, etc.). Standardizing these times to 1-10 also allowed us to better visualize our data, since there was a large gap between observations—some took only seconds to crack, while others took years. Finally, we added four new variables: `pass_length`, `num_digits`, `num_letters`, and `num_unique`. We added these variables because we believe that password length and composition could impact strength.

Exploratory Data Analysis

Given our prior knowledge of what makes passwords stronger, we chose to focus our exploratory data analysis on the predictors password length and number of unique characters, along with their relationships with other variables in the dataset.

Summary Statistics:

```
           Variable Mean Median  Sd Min Max
1          strength  6.6      7 2.3   0  10
2 true_val_strength  8.6      9 2.1   1  11
3       pass_length  6.2      6 1.1   4   9
4        num_digits 0.46      0 1.6   0   9
5       num_letters  5.7      6 1.9   0   8
6        num_unique  5.2      5 1.5   1   9
```

From the table, the average number of digits in a password are 0.464, the average number of letters is 5.718, the average number of unique characters is 5.192, and the average password length is 6.181. In general, this indicates that the most popular passwords in the data leaks used all unique letters and rarely used numbers. In terms of our predictors, the average strength was 6.6, and the average true_val_strength was 8.6, representing an online crack time of about two and a half days. This indicates that the compared to generally weak passwords, the average password in this dataset had a higher-than-average "strength" by both measures. In other words, the distribution of our data under `strength` and `true_val_strength` are left-skewed. An explanation of why we focused on these variables can be found in the methodology section.

Plots:

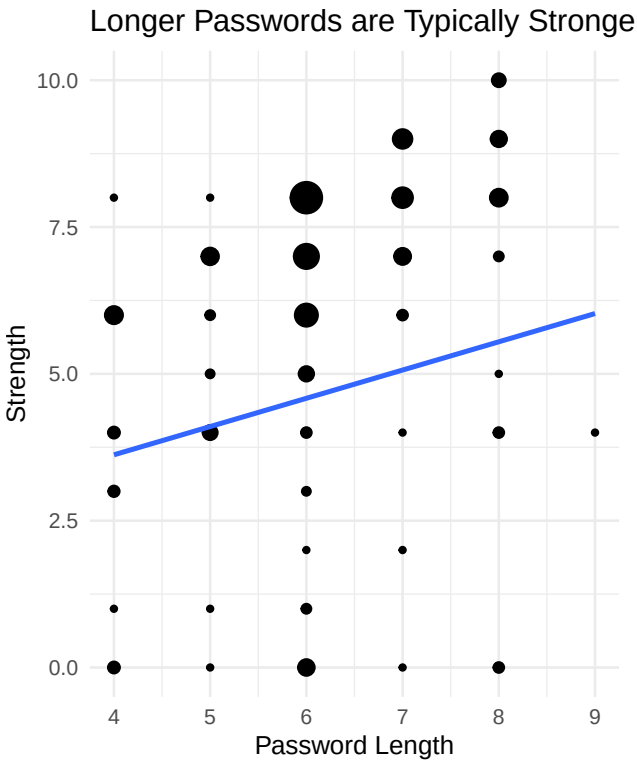

Figure 1

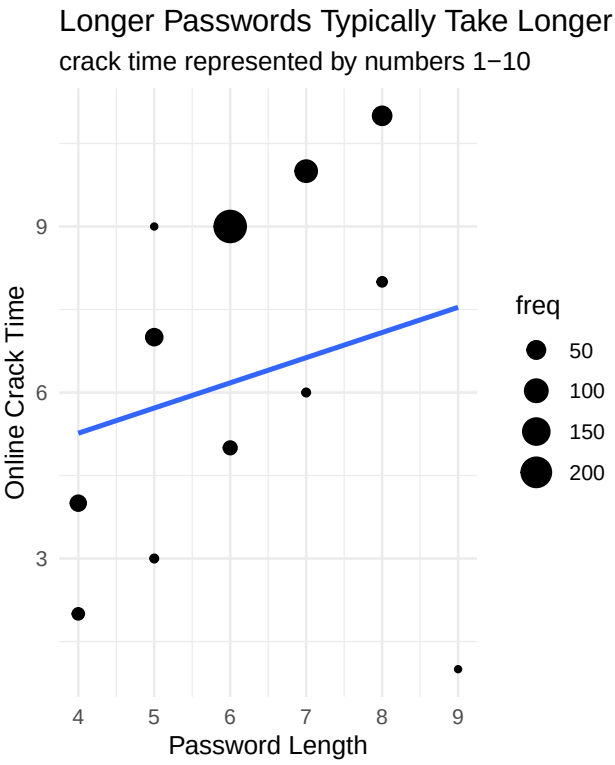

Figure 3

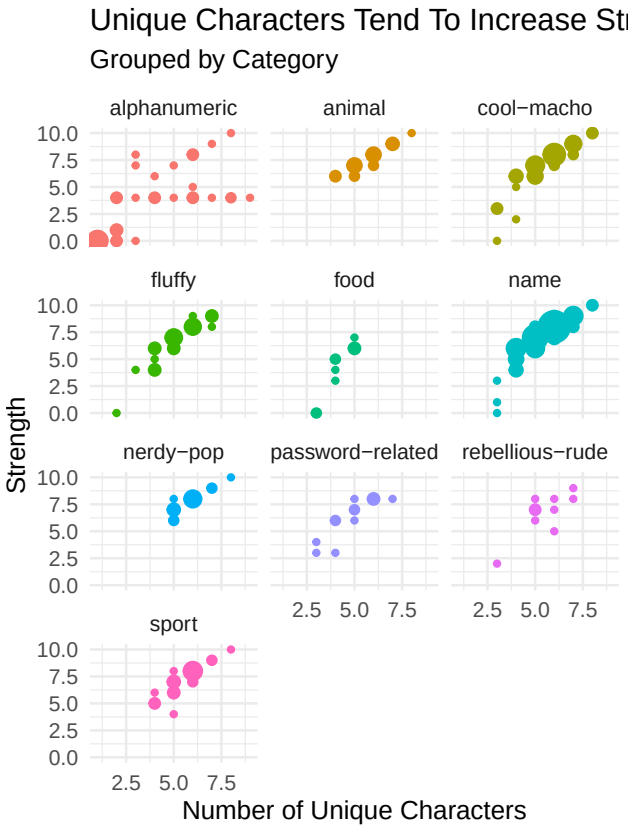

Figure 2

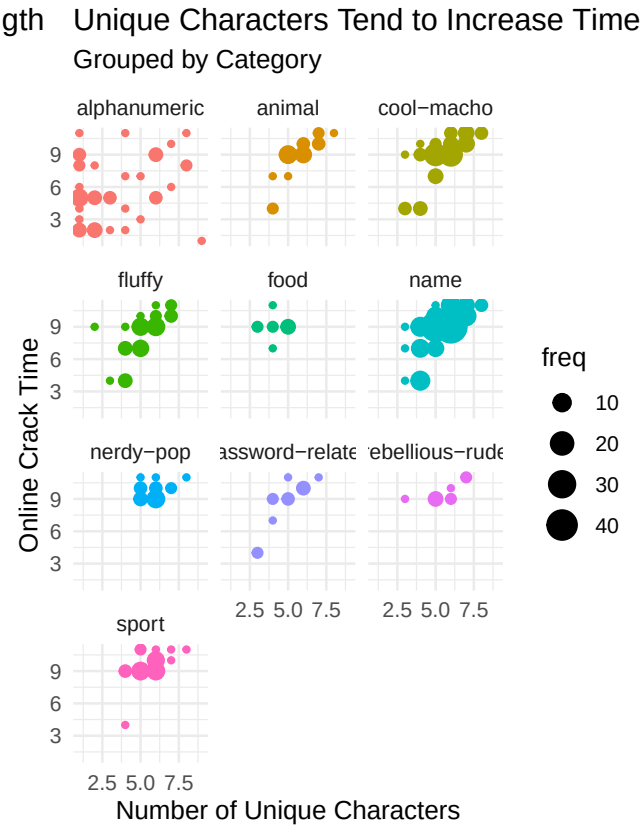

Figure 4

Figure 1 demonstrates that there appears to be a positive relationship between password length and the strength variable. The line of best fit shows that as password length increases, the strength of the password also tends to increase. We can also see this from the data themselves based on how the size of the points change as strength increases. For passwords of length 6, 7, and 8, most passwords have strengths of above 5. For passwords of length 4 and 5, there are some passwords with strengths above 5, but there appear to be a similar number of passwords with these lengths with strengths below 5, as well. This graph also helps visualize the composition of the data itself. The large size of several points associated with password length 6 indicates that, by far, most passwords in this dataset have length 6. Passwords of length 7 appear to be the next-most frequent, passwords of length 4, 5, and 8 appear to be about equally frequent, and passwords of length 9 are very infrequent. It is helpful to understand for the following data analysis that our sample size of long passwords is very small, which may limit the conclusions we can draw for that group.

Figure 2 demonstrates that the number of unique characters appears to have a positive relationship with password strength. Within each password category, as the number of unique characters increases, the strength of the password also tends to increase. This relationship holds for all categories, except simple-alphanumeric. Although there appears to be a positive relationship between number of unique terms and password strengths for some passwords in this category, the horizontal line in this plot also shows that some passwords with varying numbers of unique terms have the same password strength. Additionally, the size of the points in the plot demonstrates that some categories of passwords were more popular in our data, especially name, cool-macho, fluffy, and sport.

Figure 3 shows a positive relationship between password length and online crack time, as demonstrated by the positive slope of the line of best fit.

Figure 4 demonstrates that there appears to be a positive relationship between number of unique terms and online crack time. As the number of unique terms increases, the online crack time remains similar for many passwords. However, those passwords with the greatest online crack seconds ($> 10\hat{}\{8\}$) are typically those with at least 5 characters. There are, however, some exceptions with some passwords with few unique characters in the simple-alphanumeric and food categories still acheiving high online crack times.

Methodology

In our analysis, we treat our two outcome variables, strength and online crack time, as ordinal outcomes.

The strength variable is a number 1-10, in order of increasing password strength, making ordinal a good fit. The strength variable meets the ordinal assumption of proportional odds, since it is reasonable to assume that one-unit changes in each predictor have the same conditional relationship with being in each strength category. For example, the strength variable is calculated in part based on password length, and each one character increase in password length has the same conditional relationship with being in each strength category.

We focused on the strength, true_val_strength, pass_length, num_digits, num_letters, and num_unique variables, as we determined these would be our primary variables of interest. We excluded the rank and rank_alt variables because our research question explores what characteristics of passwords make them stronger, and their popularity in these data leaks did not have to do with their actual composition and is likely not representative of how popular these passwords are on the whole. We excluded the password variable, since the actual text of the password could not be used as a predictor–the composition of the password is encompassed in our num_digits, num_letters, num_unique, and pass_length variables. We excluded online_crack_sec

Online crack time

Based on our research and prior knowledge, the variables we believe will be important to include are `pass_length`, `num_digits`, `num_unique`, and XX…

We treat strength as an ordinal variable…

- *do we think category is important?* - since online cracking is partially done by guessing common passwords…

```
Call:
polr(formula = factor(true_val_strength) ~ . - password - num_letters -
    offline_crack_sec - strength - rank - true_val, data = pass_more)

Coefficients:
                            Value Std. Error t value
categoryanimal           -2.4073288   1.780308 -1.3522
categorycool-macho       -2.4970830   1.411714 -1.7688
categoryfluffy           -2.4119193   1.440887 -1.6739
categoryfood             -1.3852755   3.513409 -0.3943
categoryname             -2.3427836   1.312286 -1.7853
categorynerdy-pop        -1.0969569   3.353646 -0.3271
categorypassword-related -2.7557164   1.974544 -1.3956
categoryrebellious-rude   2.9555240   1.901654  1.5542
categorysport            -2.0673252   2.038791 -1.0140
rank_alt                  0.0002755   0.001723  0.1599
font_size                 0.1465549   0.156208  0.9382
pass_length              12.3361920   1.189674 10.3694
num_digits               -3.9894507   0.425417 -9.3777
num_unique               -0.4026950   0.286247 -1.4068

Intercepts:
      Value   Std. Error t value
1|2   30.9227  3.5535     8.7021
2|3   38.0604  4.4155     8.6197
3|4   42.5212  4.6420     9.1600
4|5   48.6971  4.8886     9.9613
5|6   53.9431  5.5435     9.7309
6|7   56.0261  5.8552     9.5686
7|8   63.2754  6.3311     9.9943
8|9   66.1508  6.8007     9.7270
9|10  78.6836  8.0235     9.8066
10|11 90.1927  9.1445     9.8631

Residual Deviance: 145.4019
AIC: 193.4019
```

```
# A tibble: 15 x 5
   term                    estimate std.error   statistic p.value
   <chr>                      <dbl>     <dbl>       <dbl>   <dbl>
 1 (Intercept)             -7.68e15 25651015. -299511062.       0
 2 categoryanimal          -1.63e15 21603611.  -75658073.       0
 3 categorycool-macho      -5.90e14 19000084.  -31054276.       0
```

```
 4 categoryfluffy           -5.31e14 19939353.  -26655778.       0
 5 categoryfood             -7.40e14 26058703.  -28388868.       0
 6 categoryname             -6.57e14 18124427.  -36237979.       0
 7 categorynerdy-pop        -5.02e14 22439712.  -22390475.       0
 8 categorypassword-related -4.73e14 24835080.  -19053449.       0
 9 categoryrebellious-rude  -2.19e14 26599317.   -8215697.       0
10 categorysport            -5.12e14 20748629.  -24689026.       0
11 rank_alt                  8.19e11    21807.   37578413.       0
12 font_size                 4.66e14  2481721.  187636119.       0
13 pass_length              -5.98e13  3953935.  -15131956.       0
14 num_digits               -2.35e14  3366514.  -69739344.       0
15 num_unique                5.89e14  4871430.  120896331.       0
```

```
# A tibble: 15 x 5
   term                      estimate  std.error statistic    p.value
   <chr>                        <dbl>      <dbl>     <dbl>      <dbl>
 1 (Intercept)              -56.5       12.6      -4.48    0.00000735
 2 categoryanimal            -1.59       4.30     -0.370   0.712
 3 categorycool-macho        -1.24       3.73     -0.331   0.740
 4 categoryfluffy            -2.08       3.67     -0.567   0.571
 5 categoryfood              -1.26       5.90     -0.213   0.831
 6 categoryname              -1.60       3.24     -0.494   0.621
 7 categorynerdy-pop         13.4     4816.        0.00278 0.998
 8 categorypassword-related  -1.96       6.86     -0.285   0.776
 9 categoryrebellious-rude   20.6     5730.        0.00360 0.997
10 categorysport              4.34      55.3       0.0786  0.937
11 rank_alt                   0.00281    0.00487   0.577   0.564
12 font_size                  0.319      0.755     0.423   0.673
13 pass_length               10.9        2.25      4.85    0.00000125
14 num_digits                -3.26       0.887    -3.68    0.000235
15 num_unique                -1.02       1.24     -0.828   0.408
```

to-dos - LASSO for online crack sec (and offline if we decide to do that) - hypothesis test for idk yet - should we plug in our LASSO variables into a regression (e.g. ordinal for password strength?, OLS for crack time?) or should we just use the LASSO coefficients themselves - would it be too much to see whether these strong passwords are the most common?

Results

Discussion

In terms of the research question. In terms of limitations, to reiterate, this data holds the 500 most common passwords from a data leak - and so since they are the most common, they might all also just not be strong to begin with. The strength variable that was attached, therefore, did not reference all passwords, just the ones in the data set, so super strength should not be correlated with saying that the password is very good, it is just good in comparison to the rest of the passwords in the data set. Additionally, the passwords in this data set did not have special characters or capital letters. Again this is because, being the most common, they have to be relatively simple, so we were not able to analyze these characteristics and see what influence they

have. Most passwords nowadays are forced to be inherently strong (with a minimum character, digit, and special character limit), and a lot of these leaked passwords did not follow these rules, but the analysis still confirms the belief that usually, with more unique characters and numbers, passwords tend to get stronger. To improve upon analysis, (although this may not be ethically valid) it would probably help to have a more representative idea of how passwords are in a breach (as opposed to just the most popular ones), and from there, we can test their online and offline guess time, which we can again, correlate to strength.

Sources

External research: https://www.keepersecurity.com/blog/2022/09/14/why-is-password-security-important/ https://www.bleepingcomputer.com/news/security/the-benefits-of-making-password-strength-more-transparent/ https://www.ibm.com/downloads/cas/3R8N1DZJ

Data source: https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-01-14/readme.md