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 aspect 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 (a brute force attack that guesses all possible combinations).

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

Variable Name	Type	Description
num_digits	numeric	Number of digits in the password
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:

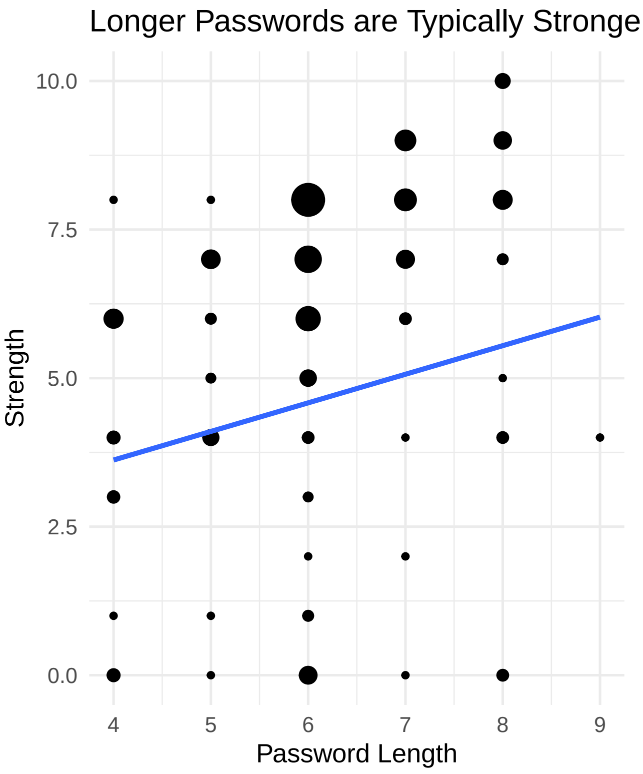

Figure 1

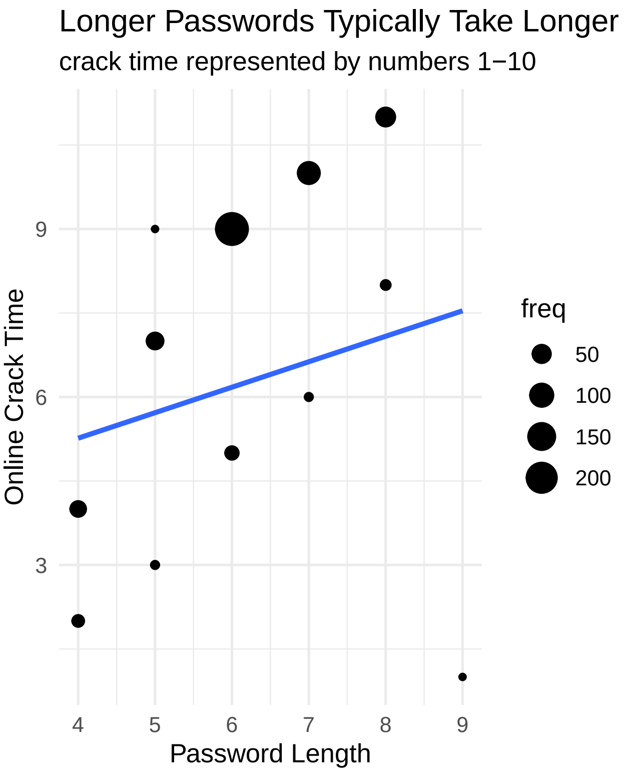

Figure 3

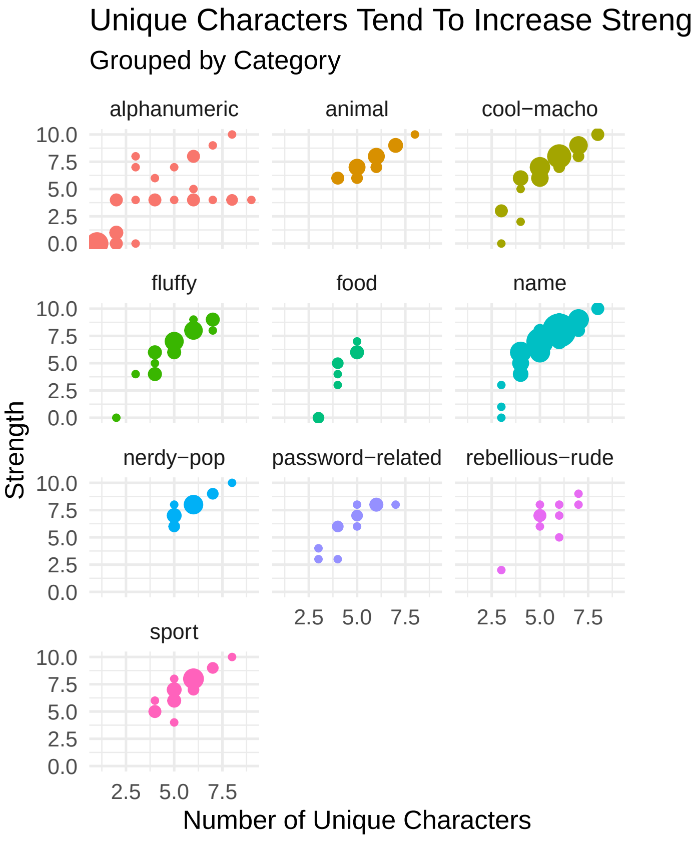

Figure 2

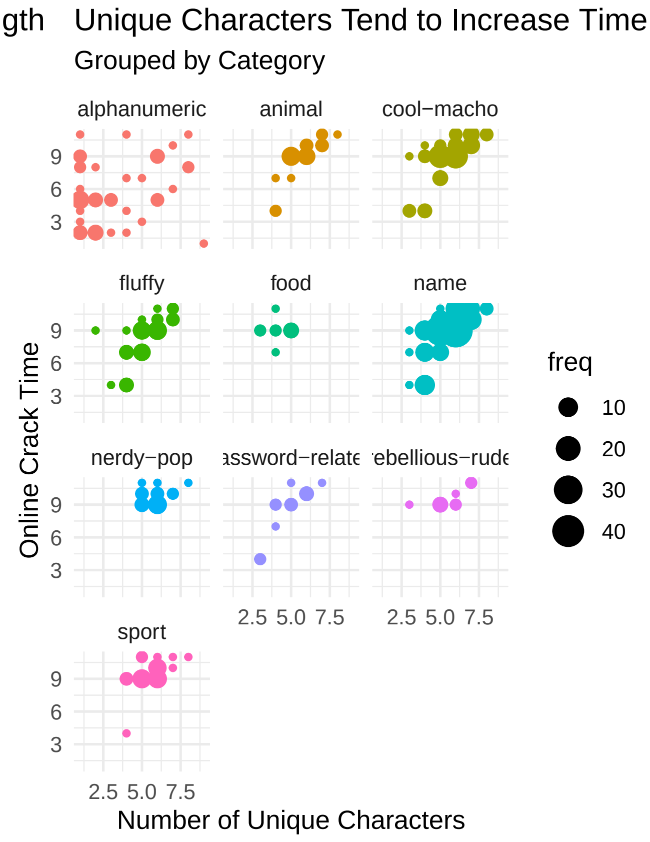

Figure 4

Figure 1 demonstrates that there appears to be a positive relationship between password length and the strength variable based on the line of best fit. 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. 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.

Figure 2 demonstrates that the number of unique characters appears to have a positive relationship with password strength. 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 (`true_val_strength`), as demonstrated by the positive slope of the line of best fit. In general, longer passwords take longer to crack, and the vast majority of passwords with online crack time categories of 9 or above are 6 characters or longer. Looking at the distribution of our data, it appears that there should be a stronger positive relationship between password between password length and online crack time, but the outlier at length 9 reduces the slope of our line of best fit. The passwords with length 9 and `true_val_strength` of 1 are really the only deviation from the trend that length increases strength.

Figure 4 demonstrates that there appears to be a positive relationship between number of unique characters and online crack time. However, whether this trend holds differs by password category. The passwords in the food, nerdy-pop, and sport categories are clustered around high online crack times (represented by values 1-10) and do not appear to have any clear pattern. For passwords in the alphanumeric category, the groups of points in the plot that look like two parallel lines with positive slopes are consistent with the general trend of positive relationship between number of unqiue characters and online crack time. However, many passwords with 2 or less unique characters have high online crack times, demonstrated by the vertical line at the left of the plot, and one point with 9 unique characters has a very low online crack time.

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.

add to this – add why true val is ordinal and also why

In terms of our outcome variables, we excluded `offline_crack_sec` becuase it is just a merely transformation of `online_crack_sec`. In terms of our predictors, we focused on pass_length, num_digits, num_letters, and num_unique variables. 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 did not end up using `num_letters` in our model because the ordinal model could not handle including more than 4 variables given that our dataset only had 500 observations. We reasoned that this should not affect our analysis, since the number of letters is correlated with the number of digits (which we did include). The number of letters can be derived from the number of digits, since all passwords in our dataset included either letters, digits, or some combination of both–no passwords included special characters.

Additionally, based on our research and prior knowledge, we believed it was reasonable to focus on password length, number of unique characters, number of digits, and category as the most important predictors of password strength. Longer passwords with more varied compositions (unique characters and digits) are typically harder to guess because that increases the options for what the password might look like. Category may also be an important indicator of strength when, instead of using brute force attacks, the person or program guesses the most common passwords–passwords that fall into certain categories may be more common and thus easier to crack.

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

Coefficients:
                          Value Std. Error t value
categoryanimal          -0.5422     0.7617 -0.7119
categorycool-macho      -0.8514     0.7015 -1.2137
categoryfluffy          -0.8548     0.7243 -1.1802
categoryfood            -3.0098     0.9140 -3.2929
categoryname            -0.6723     0.6809 -0.9873
categorynerdy-pop       -0.2455     0.7982 -0.3076
categorypassword-related -0.7074     0.8632 -0.8195
categoryrebellious-rude  -1.5309     0.9026 -1.6960
categorysport           -0.8894     0.7473 -1.1901
pass_length             -0.3116     0.1418 -2.1972
num_digits              -1.1207     0.2267 -4.9428
num_unique               3.6946     0.2155 17.1474

Intercepts:
     Value   Std. Error t value
0|1   4.4379  1.1224     3.9538
1|2   6.3768  1.1000     5.7971
2|3   7.2025  1.0380     6.9389
3|4   8.5633  1.0033     8.5348
4|5  11.4748  1.0175    11.2772
5|6  12.3062  1.0212    12.0507
6|7  15.0371  1.1057    13.5991
7|8  17.6989  1.1877    14.9012
8|9  22.0490  1.3242    16.6502
9|10 25.7696  1.4916    17.2765

Residual Deviance: 906.9067
AIC: 950.9067
```

```
          categoryanimal       categorycool-macho           categoryfluffy
              0.58146937               0.42682081               0.42536309
            categoryfood             categoryname        categorynerdy-pop
              0.04930122               0.51053601               0.78227931
categorypassword-related  categoryrebellious-rude            categorysport
              0.49292180               0.21634774               0.41089807
             pass_length               num_digits               num_unique
              0.73226425               0.32603949              40.22927327
```

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

Coefficients:
                           Value Std. Error t value
categoryanimal           -2.1008     1.7570 -1.1956
categorycool-macho       -2.2598     1.3952 -1.6197
categoryfluffy           -2.1499     1.4265 -1.5071
categoryfood             -1.3163     3.5464 -0.3712
categoryname             -2.0787     1.2922 -1.6087
categorynerdy-pop        -0.8051     3.3502 -0.2403
categorypassword-related -2.5701     1.9118 -1.3443
categoryrebellious-rude   3.1858     1.8991  1.6776
categorysport            -1.7996     2.0268 -0.8879
pass_length              12.2155     1.1713 10.4294
num_digits               -4.0217     0.4257 -9.4473
num_unique               -0.1844     0.1566 -1.1776

Intercepts:
      Value   Std. Error t value
1|2   30.0988  3.3669     8.9397
2|3   37.1905  4.2517     8.7471
3|4   41.7130  4.4530     9.3673
4|5   47.8291  4.6903    10.1974
5|6   53.0538  5.3539     9.9093
6|7   55.1551  5.6692     9.7289
7|8   62.3844  6.1357    10.1675
8|9   65.2460  6.6060     9.8768
9|10  77.7581  7.8402     9.9178
10|11 89.2016  8.9607     9.9548

Residual Deviance: 146.3203
AIC: 190.3203
```

```
          categoryanimal       categorycool-macho           categoryfluffy
            1.223622e-01             1.043671e-01             1.164979e-01
            categoryfood             categoryname        categorynerdy-pop
            2.681322e-01             1.250919e-01             4.470352e-01
categorypassword-related  categoryrebellious-rude            categorysport
            7.653060e-02             2.418719e+01             1.653680e-01
             pass_length               num_digits               num_unique
            2.018866e+05             1.792228e-02             8.315665e-01
```

```
[1] 201894.4
```

```
# A tibble: 13 x 5
   term                     estimate std.error statistic  p.value
   <chr>                       <dbl>     <dbl>     <dbl>    <dbl>
 1 (Intercept)               -24.9       2.88    -8.67   4.34e-18
 2 categoryanimal             -0.586     1.55    -0.378  7.05e- 1
 3 categorycool-macho          0.147     1.47     0.0999 9.20e- 1
 4 categoryfluffy              0.330     1.54     0.215  8.30e- 1
 5 categoryfood              -13.5     983.      -0.0138 9.89e- 1
 6 categoryname                0.483     1.43     0.337  7.36e- 1
 7 categorynerdy-pop           1.56      1.63     0.951  3.41e- 1
 8 categorypassword-related    2.04      1.79     1.14   2.56e- 1
 9 categoryrebellious-rude    -1.20      1.79    -0.674  5.00e- 1
10 categorysport               0.466     1.54     0.303  7.62e- 1
11 pass_length                -0.301     0.291   -1.03   3.01e- 1
```

```
12 num_digits                -1.76      0.295   -5.96   2.57e- 9
13 num_unique                 4.77      0.459   10.4    2.21e-25
```

```
# A tibble: 13 x 5
   term                     estimate std.error statistic      p.value
   <chr>                       <dbl>     <dbl>     <dbl>        <dbl>
 1 (Intercept)               -52.0       9.59   -5.42    0.0000000587
 2 categoryanimal             -1.11      4.57   -0.243   0.808
 3 categorycool-macho         -0.950     3.83   -0.248   0.804
 4 categoryfluffy             -1.86      3.72   -0.499   0.618
 5 categoryfood               -1.48      5.96   -0.248   0.804
 6 categoryname               -1.35      3.26   -0.414   0.679
 7 categorynerdy-pop          13.7    4963.      0.00276 0.998
 8 categorypassword-related   -1.72      6.53   -0.264   0.792
 9 categoryrebellious-rude    21.3    5643.      0.00378 0.997
10 categorysport               4.40     48.2     0.0913  0.927
11 pass_length                10.4       1.96    5.30    0.000000115
12 num_digits                 -3.31      0.779  -4.25    0.0000210
13 num_unique                 -0.592     0.502  -1.18    0.238
```

Figure 1: Predictions for Strength

	0	1
Not Strong	252	8
Strong	15	210

Figure 2: Predictions for True Strength

	0	1
Not Strong	103	2
Strong	1	379

Figure 3: AUC for ROC of Strength

.metric	.estimator	.estimate
roc_auc	binary	0.9772103

Figure 4: AUC for ROC of True Strength

.metric	.estimator	.estimate
roc_auc	binary	0.9961892

Results

Our first ordinal model shows the relationship between the predictors category, password length, number of digits, and number of unique characters and the log-odds of being in the next-highest strength category. The predictors with the largest impact on strength, as indicated by the magnitude of their slopes are number of unique characters and being categorized as food-related. The number of digits also had a relatively high slope magnitude, and password length had a small slope magnitude. It may seem strange that while controlling for the other predictors in the model, as the number of digits or password length increases, the odds of being in the next-highest strength category are predicted to decrease. However, this is because our model controls for the number of unique characters. Since our slope for number of unique characters is positive (and the exponentiated slope is greater than 1), a unique additional digit or character (which would make the password longer) is predicted to increase the odds of being in the next-highest strength category, but if the additional digit or character is not unique, it is predicted to decrease those odds. Although we do not conduct a formal hypothesis test, the high-magnitude t-values associated with the number of digits, being in the food category, and especially the number of unique characters (t value of 17.147) indicate that these predictors have a meaningful relationship with password strength and would be important to include in a model that predicts strength.

In terms of what the key coefficients from our model mean in context, the slope for `categoryfood` indicates that while controlling for all other predictors, our model predicts being in the food category to decrease a password's odds of being in the next-highest strength category (1 to 2, or 2 to 3, for example) by a multiplicative factor of 0.049. The slope for `num_unique` indicates that while controlling for all other predictors, as the number of unique characters in the password increases by 1, our model predicts the odds of being in the next-highest strength category to increase by 40.23 times.

Our second ordinal model shows the relationship between the same predictors and `true_val_strength`, which again represents an online crack time, represented by values 1-10. The predictors with the largest impact on strength, as indicated by the magnitude of their slopes are password length, number of digits, and being in the rebellious-rude category. The low magnitude slope for the number of unique characters makes sense in this model because the online crack time is determined by how long it takes to guess a password using brute force. If the computer is guessing every possible character every time, then uniqueness does not matter. The high-magnitude t-values associated with the number of digits (t value of -9.447) and password length (10.429) indicate that these predictors have a meaningful relationship with online crack time and would be important to include in a model that predicts online crack time.

In terms of what the key coefficients from our model mean in context, the slope for `categoryrebellious-rude` indicates that while controlling for all other predictors, our model predicts being in the rebellious-rude category to increase a password's odds of being in the next-highest strength category by 24.19 times. The slope for `pass_length` means that while controlling for all other predictors, as the password length increases by 1 character or digit, our model predicts the odds of being in the next-highest strength category to increase by 201,894.4 times.

Altogether, our two ordinal models suggest that different characteristics improve password strength depending on how password strength is defined. To increase the traditional, numeric measure of password strength, it may be most helpful to have more unique characters. Additionally, it may be helpful to not have a food-related password, as these passwords may be more easily guessed. To increase the time it takes to crack the password online, it appears most important to have a longer password, regardless of its composition. This makes sense based on the mechanism of online guessing, which is brute force, or trying all possible combinations.

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