TBD

Zaid and Zoe

```
library(tidyverse)
library(tidymodels)
library(stringr)
library(stringi)
library(leaps)
library(MASS)
library(glmnet)
library(caret)
library(Matrix)
pass <- read.csv("data/pass.csv")
```

```
uniqchars <- function(x) unique(strsplit(x, "")[[1]])

pass_more <- pass |>
  mutate(true_val = ifelse(time_unit == "years", 31536000*value,
                           ifelse(time_unit == "months", 2592000*value,
                          ifelse(time_unit == "days", 86400*value,
                          ifelse(time_unit == "hours", 3600*value,
                          ifelse(time_unit =="minutes", 60*value,value))))),
         true_val = jitter(true_val),
         pass_length = nchar(password),
         font_size = NULL,
         value = NULL,
         time_unit = NULL,
         rank_alt = NULL,
         num_digits = str_count(password, "[0-9]"),
         num_letters = str_count(password, "[a-z]"),
         num_vowels = str_count(password, "[a,e,i,o,u]"),
         num_unique = sapply(strsplit(password, ""),
                             function(x) length(unique(x)))) |>
```

```
  filter(!is.na(rank)) |>
  filter(strength < 11)

nrow(pass_more)
```

```
[1] 485
```

```
#this is just to organize categories for EDA

pass_ordered <- pass_more

pass_ordered$category <- with(pass_ordered,
                              reorder(category, strength, median, na.rm=T))
```

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: What characteristics yield strong passwords? 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. We decide which passwords are the strongest overall by looking at the characteristics that appear to impact both measures of password strength.

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/tree/master/data/2020/2020-01-14 https://docs.google.com/spreadsheets/d/1cz7TDhm0ebVpySqbTvrHrD3WpxeyE4hLZtifWSnoNTQ/edit#

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
offline_crack_sec	double	Time to crack offline in seconds
strength	ordinal	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
num_letters	numeric	Number of letters in the password
num_vowels	numeric	Number of vowels 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.

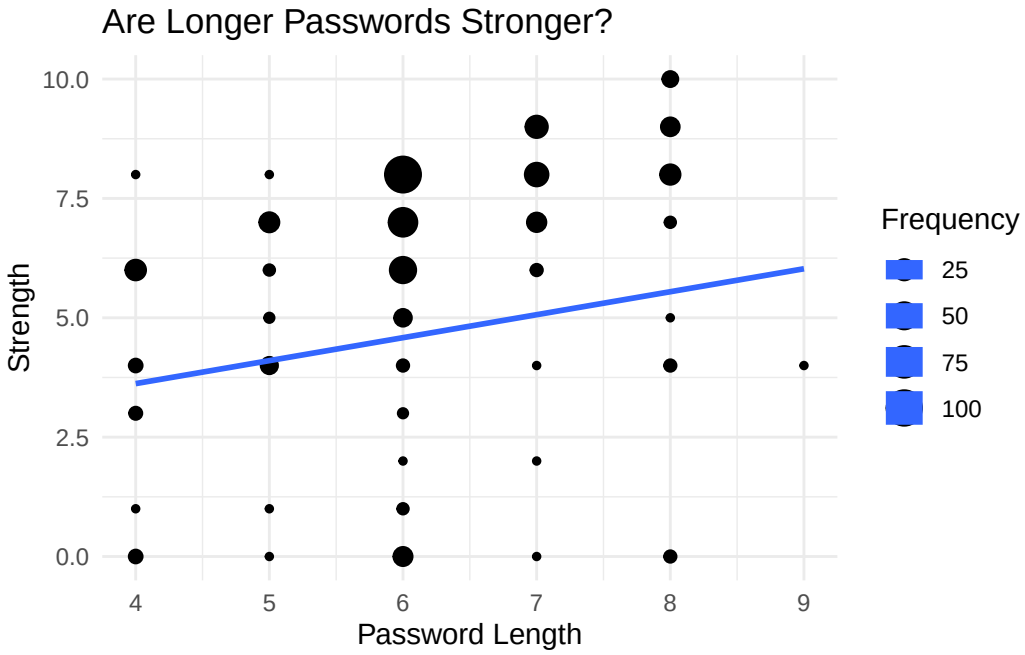
In the cleaning process, we removed the last seven observations, as all their values were "NA." We also removed several variables that appeared in the original dataset. First, we removed the variables that had a strength recorded over ten as those may have been miscalculations or strengths that were not standardized to values 1 through 10. Second, we removed the `rank_alt` variable because we wanted to focus on the passwords' first ranks of popularity, as opposed to their secondary ranks, because their first ranks were a clearer indicator of how common they were. Third, also removed the font_size variable, as the font sizes were chosen arbitrarily to display passwords in a graphic on the Knowledge is Beautiful webiste. Fourth and finally, 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). We added noise to our new `true_val` variable because the time to crack by online guessing only included discrete values (2.17 years, 0.00321 days, etc.). From there,

we were left with 485 observations. Additionally, we added five new variables: pass_length, num_digits, num_letters, num_vowels, and num_unique. We added these variables because we believe that the length of the password, as well as its composition could impact password strength.

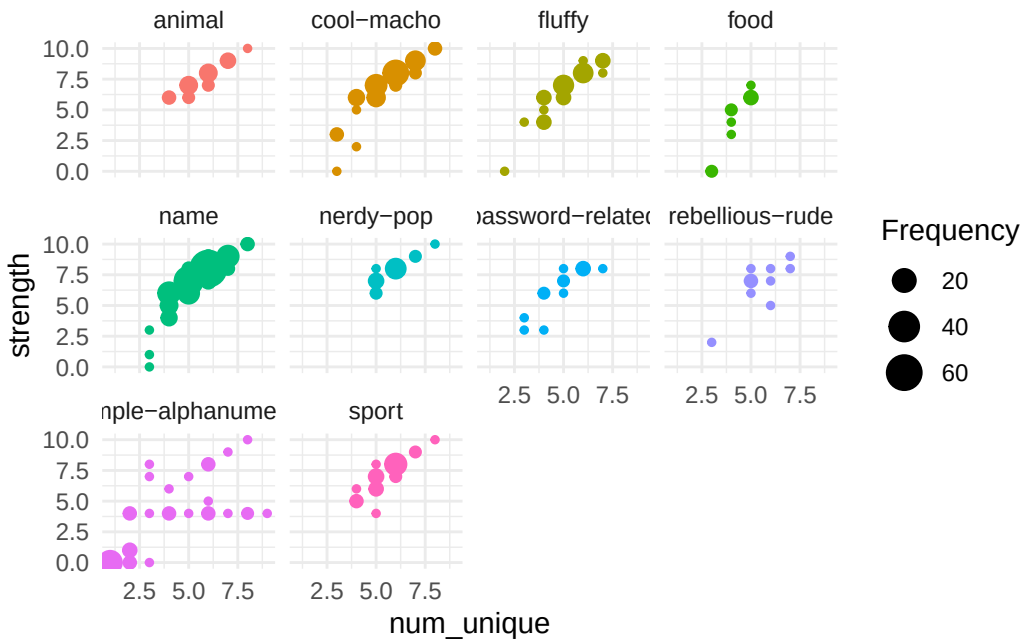
we treat as ordinal

Exploratory Data Analysis

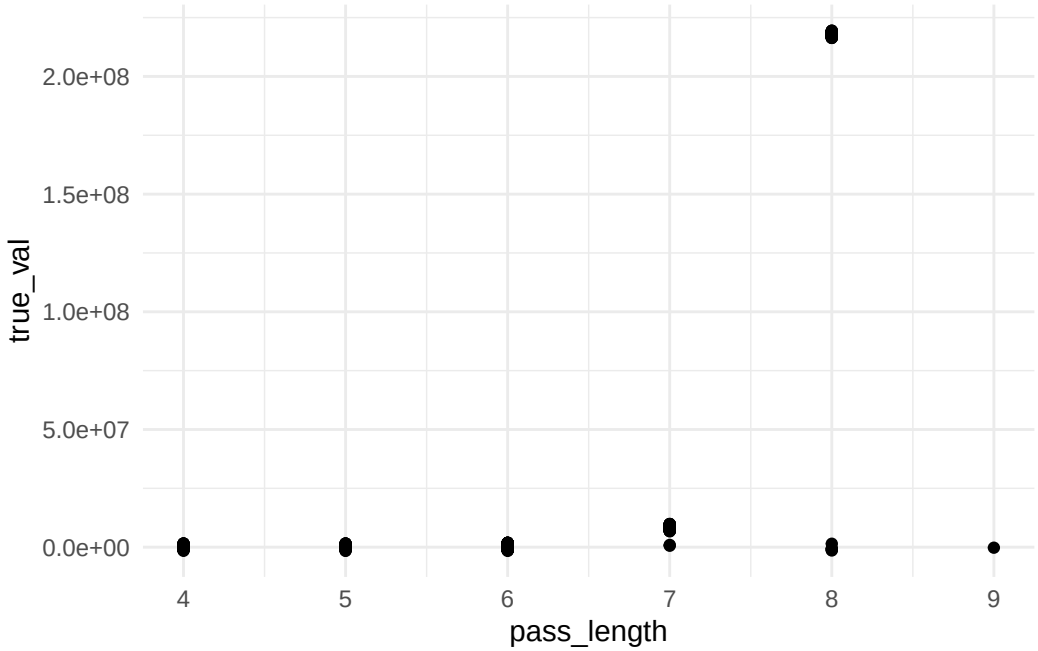
```
pass_more %>%
  group_by(strength, pass_length) %>%
  summarize(freq=n()) |>
  ggplot(aes(x = pass_length, y = strength, size = freq)) +
  geom_point() +
  geom_smooth(method = "lm", se = F) +
  theme_minimal() +
  scale_fill_viridis_d() +
  labs(x = "Password Length", y = "Strength",
       title = "Are Longer Passwords Stronger?", size = "Frequency")
```



```
pass_more %>%
  group_by(num_unique, strength, category) %>%
  summarize(freq=n()) |>
  ggplot(aes(x = num_unique, y = strength, color = category, size = freq)) +
  facet_wrap(~category) +
  geom_point() +
  theme_minimal() +
  scale_fill_viridis_d() +
  guides(size=guide_legend("Frequency"), color = "none")
```




```
pass_more |>
  group_by(pass_length, true_val) |>
  ggplot(aes(x = pass_length, y = true_val))+
  geom_point() +
  theme_minimal() +
  scale_fill_viridis_d()
```




```
pass_ordered %>%
  summarize(mean(strength))
```

```
  mean(strength)
1       6.606186
```

```
pass_ordered %>%
  summarize(mean(num_digits))
```

```
  mean(num_digits)
1        0.4639175
```

```
pass_ordered %>%
  summarize(mean(num_letters))
```

```
  mean(num_letters)
1          5.717526
```

```
  pass_ordered %>%
    summarize(mean(num_unique))
```

```
  mean(num_unique)
1         5.191753
```

The first visualization demonstrates that at password length increases,

Preliminary findings: strength - most common strength is 8, but again, note that this is in relation to the generally bad passwords in the dataset.

The explanatory data analysis helps the reader better understand the observations in the data along with interesting and relevant relationships between the variables. It incorporates appropriate visualizations and summary statistics.

Methodology

```
  y1 <- pass_more$strength
  x1 <- model.matrix(strength ~ . - password - true_val - offline_crack_sec - rank,
              data = pass_more)

  y2 <- pass_more$true_val
  x2 <- model.matrix(true_val ~ . - password - offline_crack_sec - strength - rank,
              data = pass_more)

  m_lasso_strength <- cv.glmnet(x1, y1, alpha = 1)
  best_lambda <- m_lasso_strength$lambda.min
  best_lambda
```

```
[1] 0.02824799
```

```
  m_best <- glmnet(x1, y1, alpha = 1, lambda = best_lambda)
  m_best$beta
```

```
15 x 1 sparse Matrix of class "dgCMatrix"
                                    s0
(Intercept)                 .
categorycool-macho          .
```

```
categoryfluffy                .
categoryfood                -1.15374041
categoryname                 0.03618070
categorynerdy-pop            0.13009429
categorypassword-related     .
categoryrebellious-rude     -0.18828472
categorysimple-alphanumeric -0.37854184
categorysport                .
pass_length                 -0.08570262
num_digits                  -0.30586615
num_letters                  .
num_vowels                   .
num_unique                   1.20276041
```

```
m_lasso_strength <- cv.glmnet(x2, y2, alpha = 1)
best_lambda <- m_lasso_strength$lambda.min
best_lambda
```

```
[1] 677900.9
```

```
m_best <- glmnet(x2, y2, alpha = 1, lambda = best_lambda)
m_best$beta
```

```
15 x 1 sparse Matrix of class "dgCMatrix"
                                    s0
(Intercept)                          .
categorycool-macho           3343384.3
categoryfluffy               3702398.0
categoryfood                -2283700.4
categoryname                         .
categorynerdy-pop          -11176402.9
categorypassword-related             .
categoryrebellious-rude       241836.4
categorysimple-alphanumeric  34865630.8
categorysport                -5237624.3
pass_length                  34950690.4
num_digits                           .
num_letters                  10832765.2
num_vowels                           .
num_unique                   -3585163.3
```

Based on prior knowledge / research, the variables we think will be important to include are pass_length, num_digits, and num_unique - *do we think category is important?* - since online cracking is partially done by guessing common passwords…

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