TBD

Zaid and Zoe

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

```r
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,val
         pass_length = nchar(password),
         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)
```

Introduction

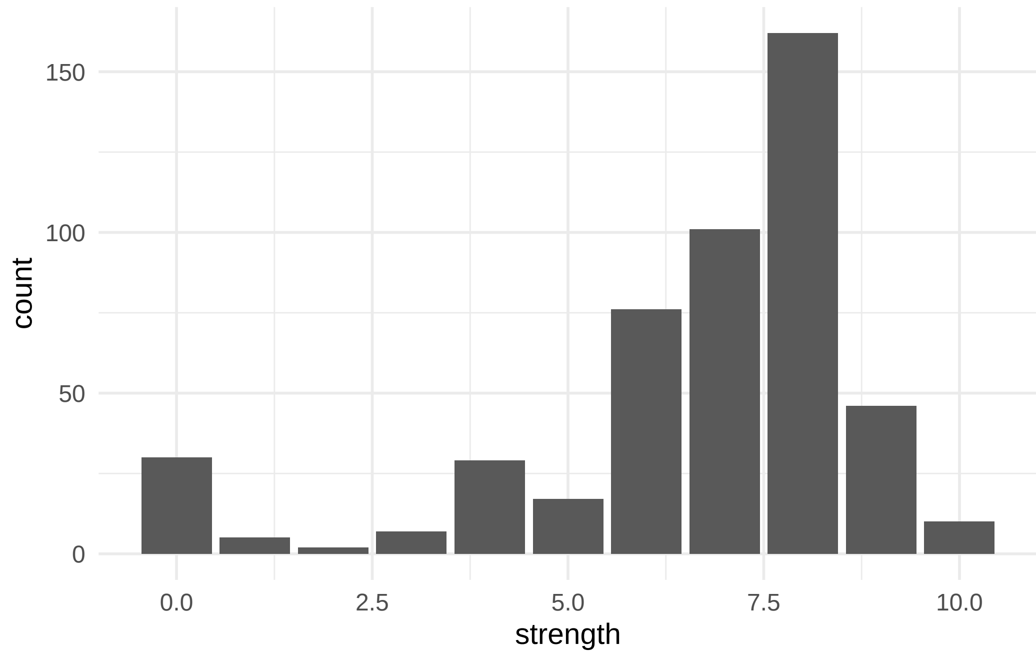
Variable Name	Type	Description
rank	double	Popularity in their database of released passwords
password	character	Actual text of password
category	character	Classification of type of password
value	double	Time to crack by online guessing
time_unit	character	Time unit to match with value
true_val	double	Time to crack by online guessing standardized to seconds
offline_crack_sec	double	Time to crack offline in seconds
rank_alt	double	Rank 2
strength	double	Quality of password where 10 is highest, 1 is lowest
font_size	double	Used to create the graphic for KIB
pass_length	double	Length of the password
num_digits	double	Number of digits in the password
num_letters	double	Number of letters in the password
num_vowels	double	Number of vowels in the password
num_unique	double	Number of unique characters (letters or numbers in the password)

In the cleaning process, we removed the last seven observations, as all their values were "NA." Additionally, we removed the variables that had a strength recorded over ten as those may have been miscalculations. 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 believe that the length of the password, as well as its composition could possibly impact the strength of the password, and so we decided to add them in to investigate their various relationships.

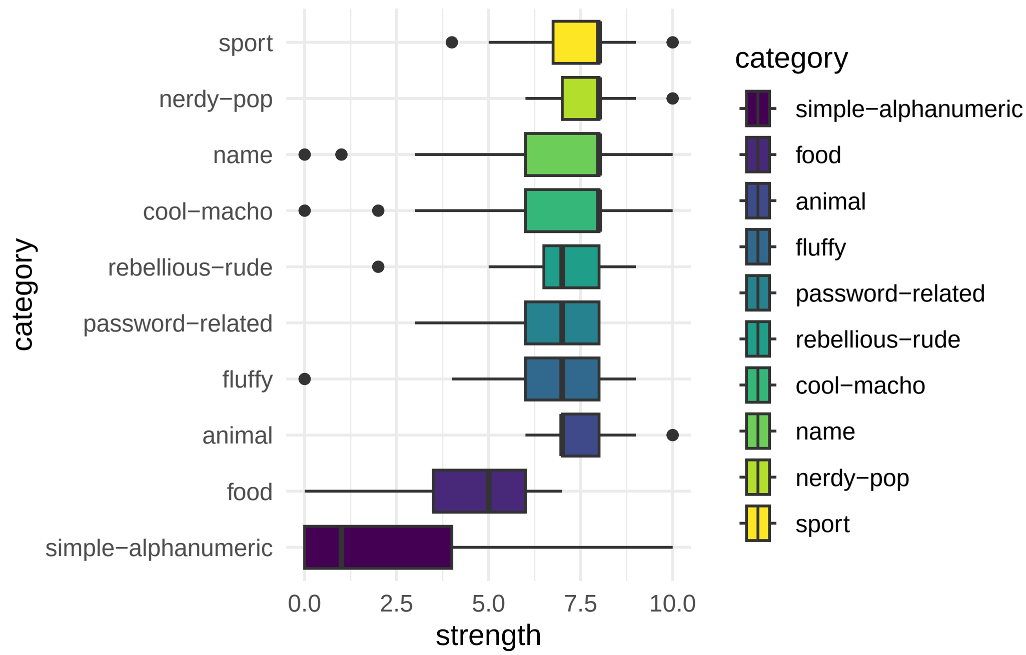
The research question and motivation are clearly stated in the introduction, including citations for the data source and any external research. The data are clearly described, including a description about how the data were originally collected and a concise definition of the variables relevant to understanding the report. The data cleaning process is clearly described, including any decisions made in the process (e.g., creating new variables, removing observations, etc.) 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.

Exploratory Data Analysis

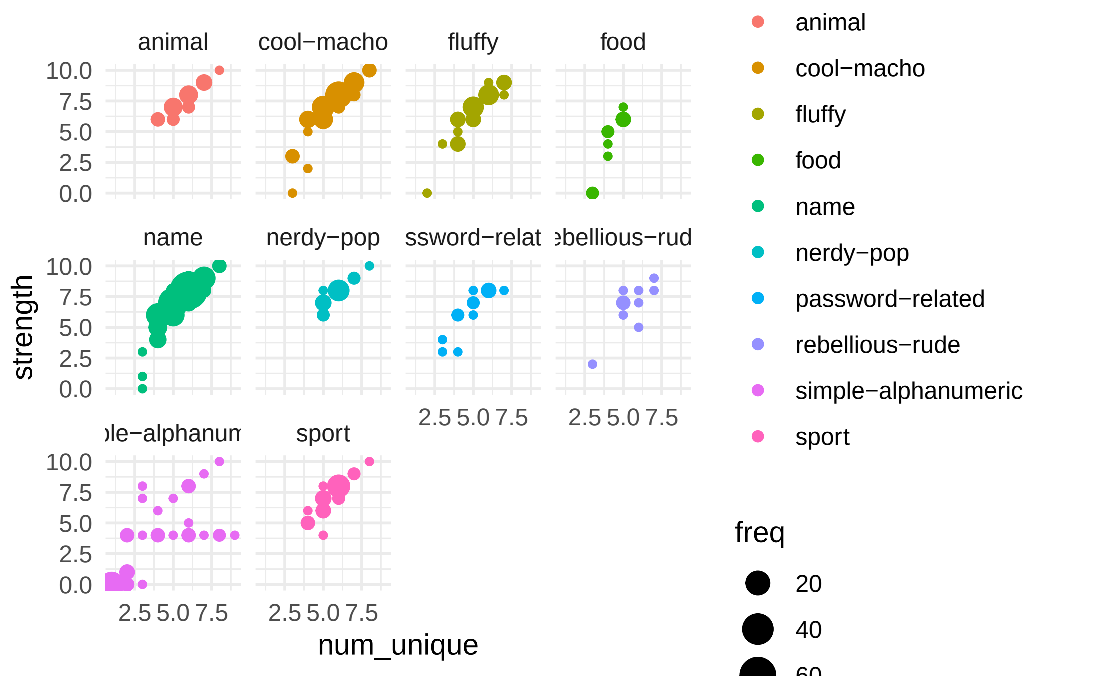
```
#im gonna comment some of these out because i feel like we have to be picky with which gra

pass_ordered <- pass_more

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

# pass_more %>%
#   ggplot(aes(x = pass_length)) +
#   geom_bar()

# pass_more %>%
#   ggplot(aes(x = category)) +
#   geom_bar()

pass_more %>%
  ggplot(aes(x = strength)) +
  geom_bar() +
  theme_minimal() +
  scale_fill_viridis_d()
```



```
# pass_more %>%
#   ggplot(aes(x = num_digits)) +
#   geom_bar()
#
# pass_more %>%
#   ggplot(aes(x = num_letters)) +
#   geom_bar()
#
# pass_more %>%
#   ggplot(aes(x = num_vowels)) +
#   geom_bar()
#
# pass_more %>%
#   ggplot(aes(x = num_unique)) +
#   geom_bar()

pass_ordered |>
  ggplot(aes(x = strength, y = category, fill = category)) +
  geom_boxplot() +
  theme_minimal() +
  scale_fill_viridis_d()
```



```
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()
```



Preliminary findings: - rank - (from dataset) top 5 most popular passwords were password, 123456, 12345678, 1234, and qwerty - category - name passwords most common - length - passwords of length 6 most common - strength - only looking at strength ratings 1-10, passwords with 8 rating most popular (these are relative to generally bad passwords tho) - number digits: 0 digits most popular - number letters: 6 most common (makes sense if digits aren't common and passwords are usually of length 6) - number vowels: 2 most common - num unique: 6 most common, 5 not too far off

Conclusions: - seeming like name passwords of length 6 and relatively high strength compared to generally bad passwords, without numbers are the most common. These are probably actual words and not just random repeated letters since the num unique is typically equal to the length (also just looking at the passwords)

Methodology

```
y <- pass_more$strength
x <- model.matrix(strength ~ category + font_size + pass_length + num_letters + num_digits
            data = pass_more)
m_lasso_strength <- cv.glmnet(x, y, alpha = 1)
best_lambda <- m_lasso_strength$lambda.min
best_lambda
```

```
[1] 0.03034234
```

```
m_best <- glmnet(x, y, alpha = 1, lambda = best_lambda)
m_best$beta
```

```
16 x 1 sparse Matrix of class "dgCMatrix"
                                     s0
(Intercept)                  .
categorycool-macho           .
categoryfluffy               .
categoryfood                -0.48298160
categoryname                 .
categorynerdy-pop            0.05467965
categorypassword-related     .
categoryrebellious-rude     -0.10793719
categorysimple-alphanumeric  .
categorysport                .
font_size                    0.53783506
pass_length                  0.03477627
num_letters                  0.08087130
num_digits                   .
num_vowels                   .
num_unique                   0.41472087
```

1. run regressions to see which help predict password strength most? what is our response variable tho

- can do LASSO and stepwise and compare models ?
- fit model with interaction terms maybe?

2. hypothesis test

3. we could prolly run a logistic regression but not sure why we'd want to predict the odds? idk

not seeing use for multinomial or ordinal regression bc i doubt our outcome will be categorical...

don't think need mixed model bc not seeing any grouping

Results

Discussion