Pokemon Project

Kotomi Oda
Kaleb Cervantes
Nikhil Taringonda

October 7 2022

What attributes of Pokemon make them harder to catch?

Fortunately, Kaggle user Rounak Banik scraped data from serebii.net in 2017 to make "The Complete Pokemon Dataset". The dataset contains a lot of information for Pokemon in the mobile game *Pokemon Go*. Now, this dataset technically is not a "complete" Pokemon dataset as of the time this project was done. This is because since the data was scraped, there has been an entire Generation of Pokemon games that have been released.

We decided to clean the data based on the following:

- The resulting tibble should be in a tidy format.
- Only one unique identifier column is needed
- Categorical variables should be stored as factors with up to 15 different levels.

The resulting tibble had 800 rows and 19 columns. Seven of these columns were of interest to us for this project:

- `capture_rate`

  8-bit integer column that is used to calculate the probability that a Pokemon is caught.
- `name`

  Character column containing the Pokemon's official English name. This is also a unique identifier.
- `type1`, `type2`

  Character columns representing the primary and secondary types of the pokemon respectively.
- `generation`

  Factor representing the generation in which the Pokemon was introduced. This dataset only contains up to Generation 7 (i.e. Pokemon from *Ultra Sun*, *Ultra Moon*, and previous titles).
- `is_legendary`

  Logical column that is true when the Pokemon is legendary and false otherwise.
- `classification`

  Character column that describes biological characteristics of the Pokemon.

First, an additive ANOVA model was used to see if there were significant differences in the mean capture rate depending on different categorical attributes of the pokemon. The full model seemed to indiate that the interaction between primary and secondary types — and the effects of classification — were not signifcant in their difference of mean capture rate.

After this was done, a reduced ANOVA model was made to just include the additive effect of primary and secondary type, generation, and legendary status. In this model, all of these attributes seemed significant.

When the assumptions for ANOVA were checked, the constant variance and normality assumptions seemed to be violated for both models. In order to find an appropriate transformation, Box-Cox power transformations were used. These resulted in a rounded power of 0.09 and 0.13 for the first and second models respectively.

In order to still look at the impact of the columns on capture rate, box-plots were used.

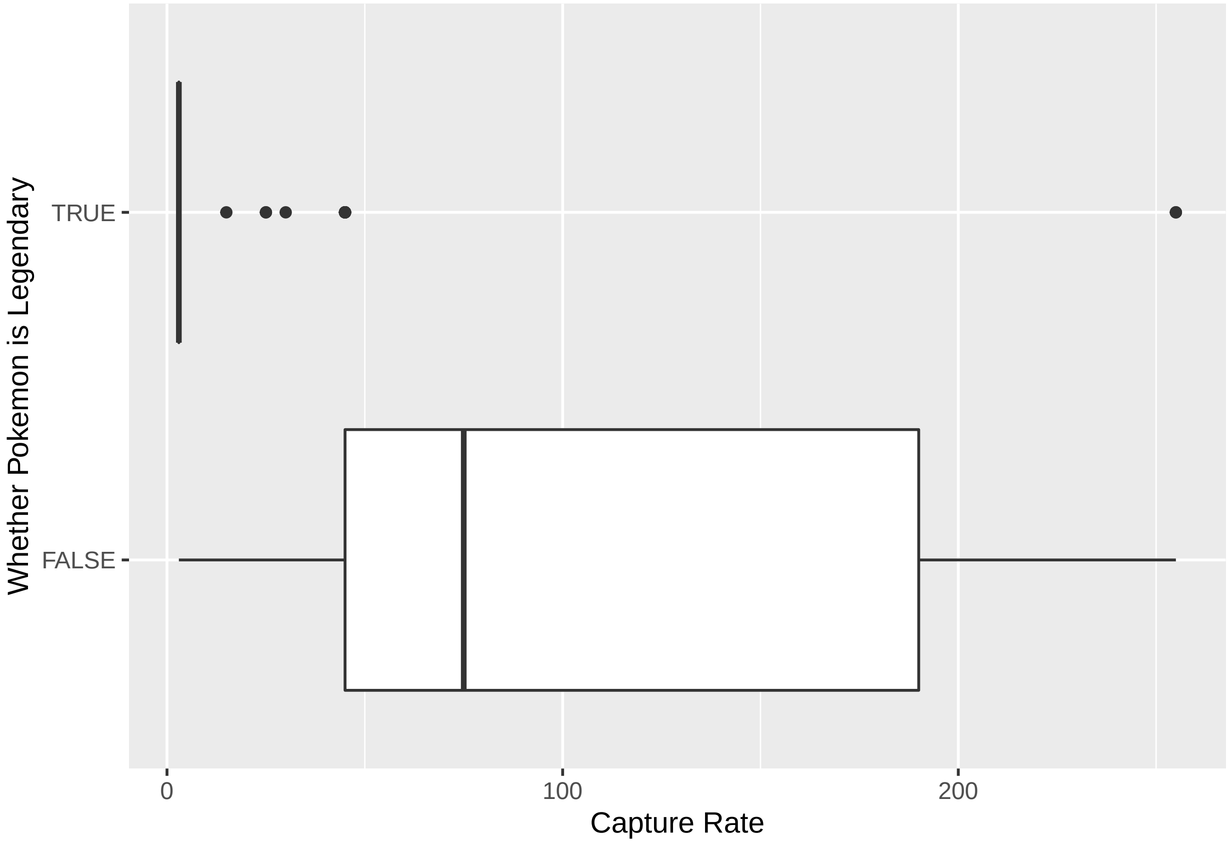

Figure 1: Legendary Status Distribution

Figure 1 seems to indicate that Legendary Pokemon tend to have significantly capture rates than non-Legendary Pokemon.

In figure 2, Pokemon classifications do not seem to have significant differences in capture rate. However, certain classifications such as Mouse and Mushroom Pokemon have significantly higher capture rates. Fox, Dragon, and Armor Pokemon seem to have lower capture rates than other classifications.

These indicate that although the ANOVA model may not have been conclusive, there still are significant differences in capture rate based on classification and legendary status.

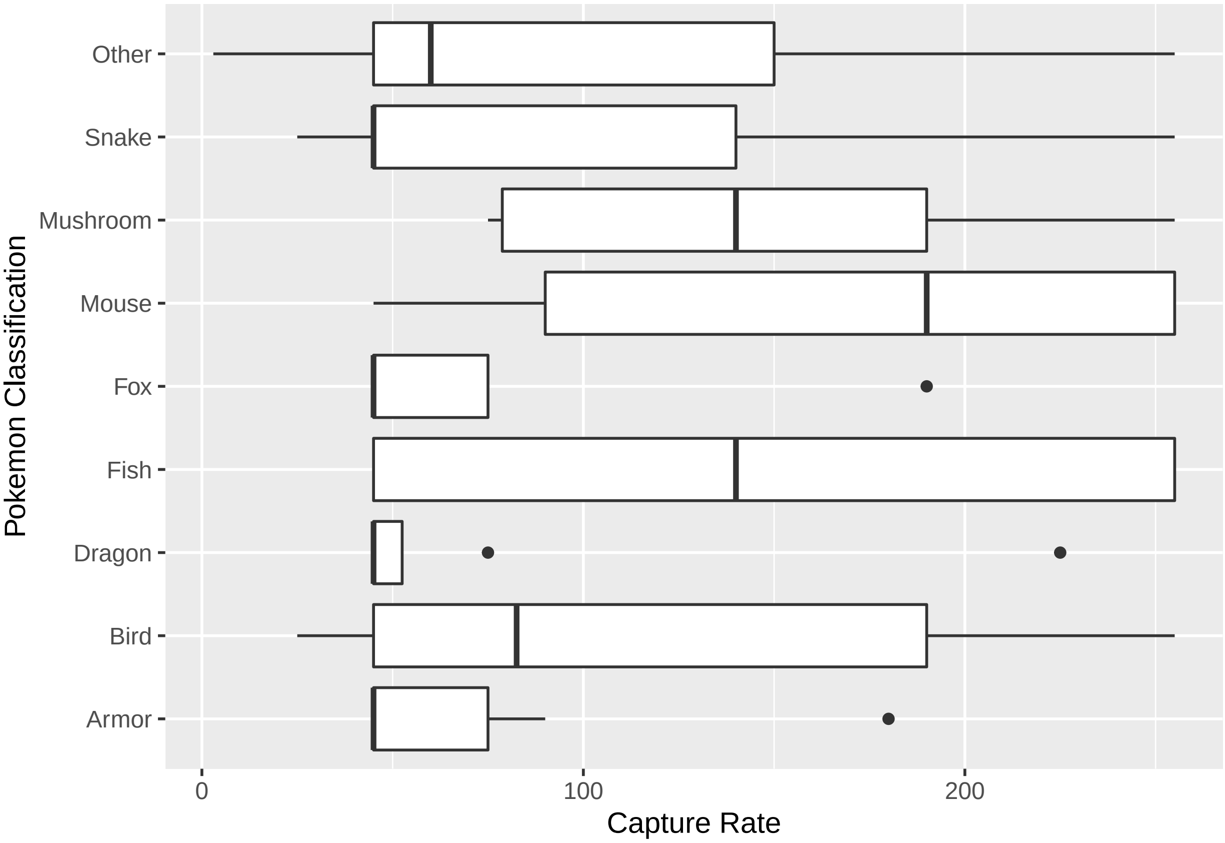

Figure 2: Classification Distribution

Code Appendix

The project can be found at the following link: https://github.com/KFCervantes/STAt650_Pokemon/

```
# sets up default settings for code chunks
knitr::opts_chunk$set(
  echo = F,
  message = F,
  warning = F
)

library(tidyverse)

pokemon <- read_csv("pokemon.csv") %>%

  select(

    # ignore type combat modifiers
    !contains("against"),

    # ignore untidy columns and other unique identifiers
    -c(abilities, japanese_name, pokedex_number, base_total)

  ) %>%

  # ignore rows with multiple values for the response
  filter(name != "Minior") %>%

  mutate(
    capture_rate = as.numeric(capture_rate),

    # simplifiy classification to main type and reduce levels
    classification = classfication %>%
      str_extract(r"([:alpha:]+(?= Pok))") %>%
      fct_lump_n(8),

    # reduce levels
    type1 = type1 %>%
      fct_lump_n(14),

    # fill NAs and redue levels
    type2 = type2 %>%
      fct_explicit_na("None") %>%
      fct_lump_n(13),

    generation = as.factor(generation),
    is_legendary = as.logical(is_legendary),

    # remove old classification column to correct spelling error
    .keep = "unused"
  )

# full anova model
aov1 <- aov(
```

```r
  capture_rate ~ type1 * type2 + generation + is_legendary + classification,
  pokemon
)

# summary and diagnostics of full model
summary(aov1)
plot(aov1, which = 1)
car::powerTransform(aov1) %>% summary()

# reduced anova model
aov2 <- aov(
  capture_rate ~ type1 + type2 + generation + is_legendary,
  pokemon
)

# summary and diagnostics of reduced model
summary(aov2)
plot(aov2, which = 1)
car::powerTransform(aov2) %>% summary()

# boxplots of columns that did not show significant differences
ggplot(pokemon, aes(capture_rate, type1)) + geom_boxplot()
ggplot(pokemon, aes(capture_rate, type2)) + geom_boxplot()
ggplot(pokemon, aes(capture_rate, generation)) + geom_boxplot()

# title is included in figure caption
ggplot(pokemon, aes(capture_rate, is_legendary)) +
  geom_boxplot() +
  labs(
    x = "Capture Rate",
    y = "Whether Pokemon is Legendary"
  )

# title is included in figure caption
ggplot(pokemon, aes(capture_rate, classification)) +
  geom_boxplot() +
  labs(
    x = "Capture Rate",
    y = "Pokemon Classification"
  )
```