Pokemon Project

Kotomi Oda
Kaleb Cervantes
Nikhil Taringonda

October 7, 2022

What categorical 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 about 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.

Initially, the data contained 801 rows and 41 columns. With the exception of one column and two rows, the data was in a tidy format. Not all of the columns would be necessary for this project, so 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.

  Higher capture rates correspond to a Pokemon being easier to catch while lower capture rates correspond to a Pokemon being more difficult to catch.

- `name`

  Character column containing the Pokemon's official English name. This is also a unique identifier.

- `type1`, `type2`

  Character columns represent 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 the biological characteristics of the Pokemon.

  The raw data contained many classifications that were sub-classifications of others. In these cases, the primary classification was used. Pokemon with multiple classifications had their first listed classification used for this project. Fortunately, all of these situations were able to be simplified with a regular expression.

Table 1 provides some summary statistics for the capture rates in the data.

Table 1: Summary Table for `pokemon$capture_rate`

Min.	3.00000
1st Qu.	45.00000
Median	60.00000
Mean	98.76125
3rd Qu.	170.00000
Max.	255.00000

These summary statistics are relevant to why the cleaned dataset has one less observation than the raw data. The row that was removed corresponded to the Pokemon Minior. Minior is a Pokemon with two forms. Each form has a different capture rate. In one form, Minior had a capture rate of 30, which is below the first quartile. In its other form, Minior had a capture rate of 255, which is the maximum value for capture rate. Since this Pokemon had extremely different capture rates depending on form, it was decided that this observation would be removed.

Different methods were used to find which categorical variables have a significant effect on capture rate. First, ANOVA models — with significance level $\alpha = 0.05$ — were used to see if there were significant differences in the mean capture rate depending on categorical variables. The first model used the additive effects of:

- Pokemon generation
- legendary status
- primary and secondary type, as well as their interaction effect

The full model seemed to indicate that the interaction between primary and secondary types — and the effects of classification — were not significant in their difference of mean capture rate for $\alpha = 0.05$.

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 for $\alpha = 0.05$.

When the assumptions for ANOVA were checked, the constant variance and normality assumptions seemed to be violated for both models. 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. This indicated that the results of the ANOVA models may not have been valid.

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

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

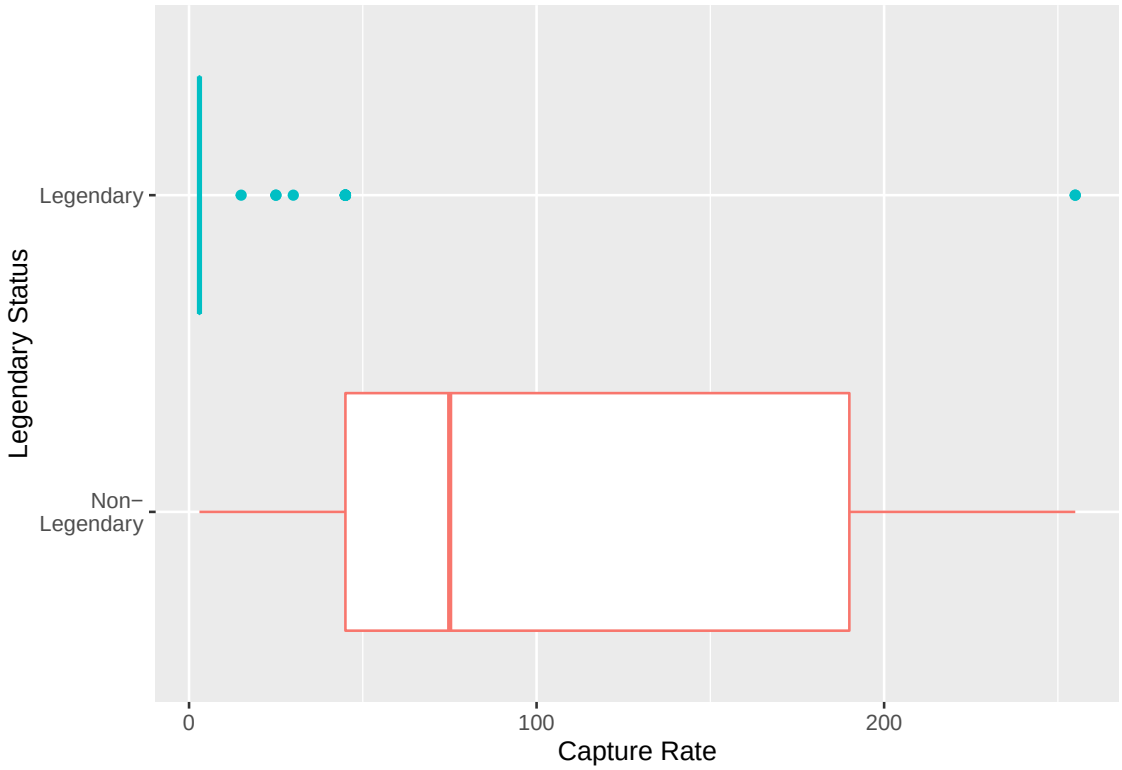

Figure 1: Distributions of Capture Rates for Legendary and Non-Legendary Pokemon

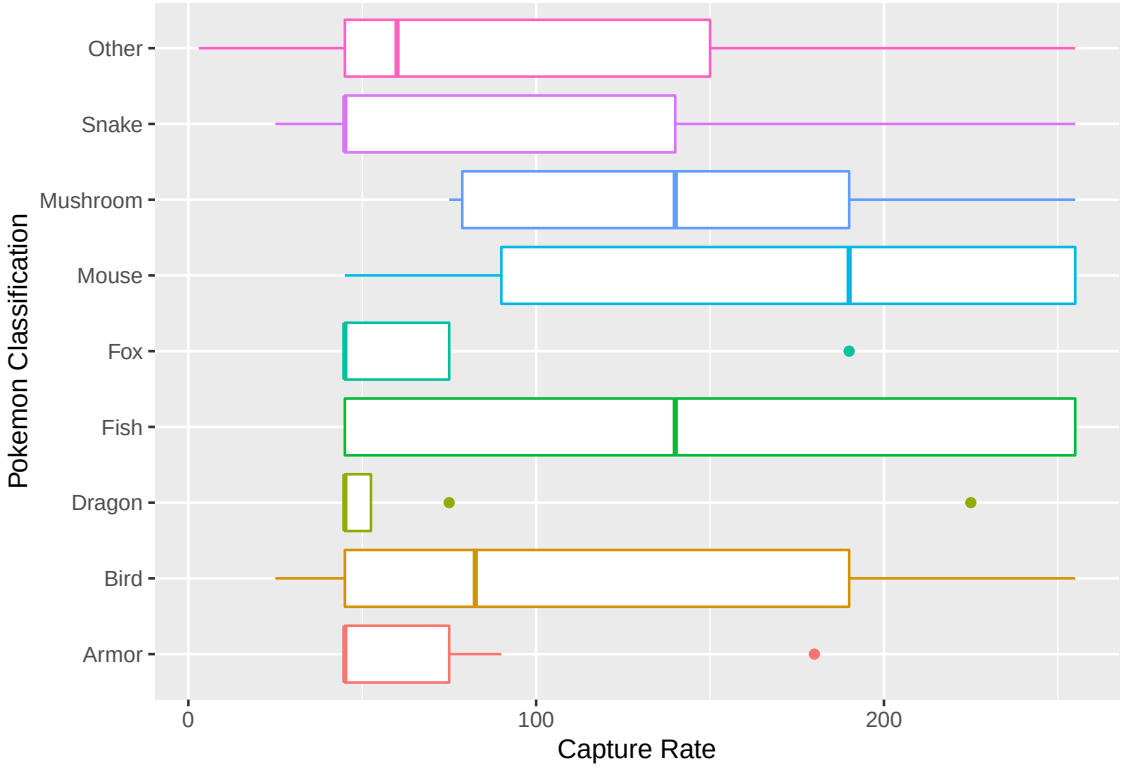

Figure 2: Distributions of Capture Rates within Classifications

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.

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"
  )

# gets summary of capture rate column
pokemon %>%
```

```
  pull(capture_rate) %>%
  summary() %>%

  {

    # adds output from summary function as columns in a dataframe
    data.frame(
      names(.),
      as.vector(.)
    )

  } %>%

  # converts dataframe to table
  knitr::kable(
    col.names = c("", ""),
    caption = r"(Summary Table for `pokemon$capture_rate`)"
  )

# row that was removed from the raw data
# shows capture rate from this row
read_csv("pokemon.csv") %>%
  filter(name == "Minior") %>%
  pull(capture_rate)

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

```r
pokemon %>%

  # change logical to factor to add factor values to plot labels
  mutate(
    is_legendary = as.factor(is_legendary) %>%
      fct_recode(
        Legendary = "TRUE",
        `Non-\nLegendary` = "FALSE"
      )
  ) %>%

  # plot modified version of column
  ggplot(aes(capture_rate, is_legendary, col = is_legendary)) +
  geom_boxplot() +
  theme(legend.position = "none") +
  labs(
    x = "Capture Rate",
    y = "Legendary Status"
  )

# title is included in figure caption
ggplot(pokemon, aes(capture_rate, classification, col = classification)) +
  geom_boxplot() +
  theme(legend.position = "none") +
  scale_fill_brewer(palette = "Bupu") +
  labs(
    x = "Capture Rate",
    y = "Pokemon Classification"
  )
```