Pokemon Analysis

Connor Bryson, Nathen Byford, Miguel Iglesias

10/12/2021

Introduction

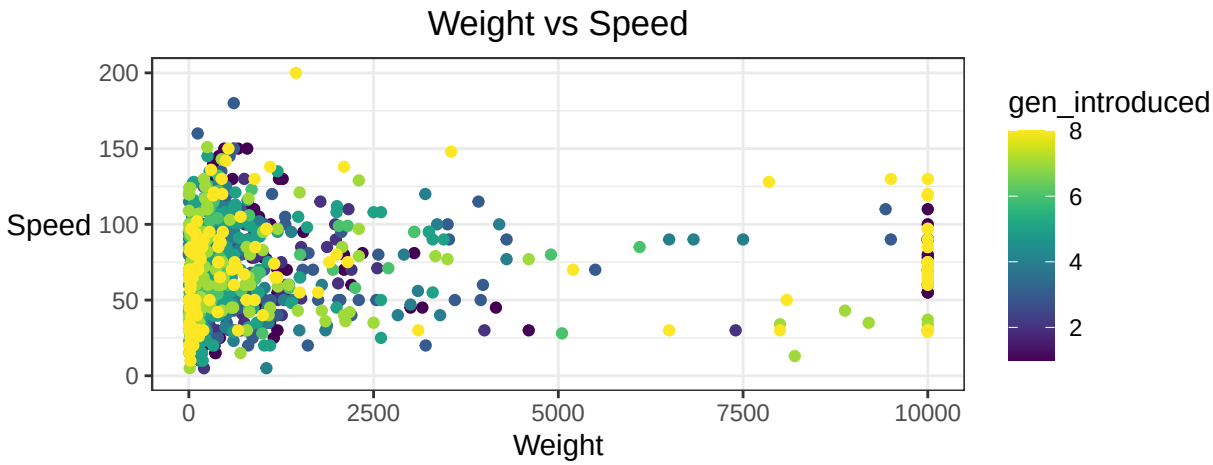
Pokémon, a Japanese card and video game, revolves around a fantasy world where people fight each other with creatures they find and domesticate. Each creature, or Pokémon, can be characterized by certain attributes from health points (hp), to their typing which includes fire-Pokémon, water-Pokémon, and grass-Pokémon among others. Our dataset is a CSV that takes all the defined characteristics of the Pokémon and collects into one useable file. Though the dataset includes 49 variables, the ones we are using include: _____.

Questions and Findings

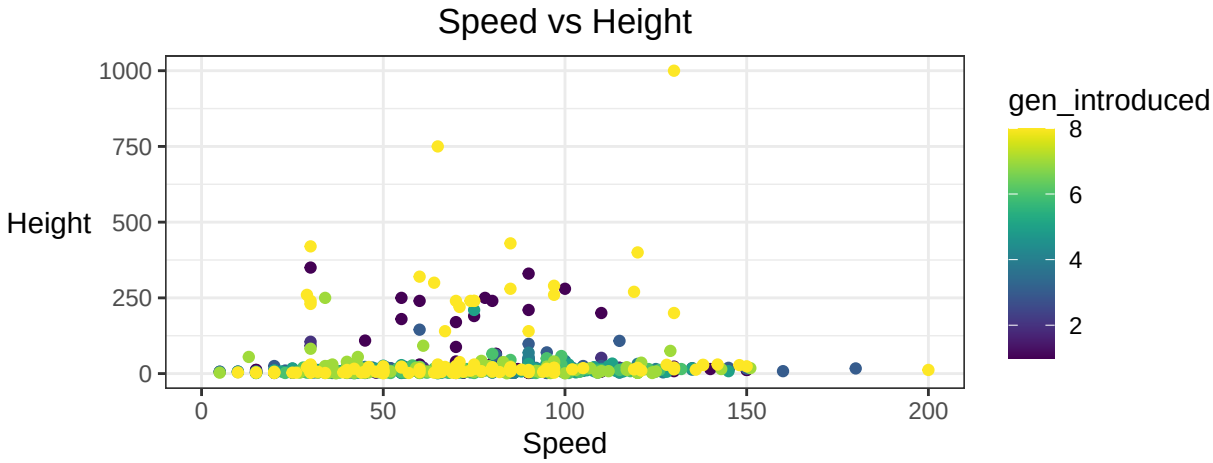
Speed

We want to begin with looking into what factors effect the speed of the Pokemon. This included testing whether real life factors such as weight and height affected the speed of the Pokemon. At first we believed they would, but as seen by the graphs below we were wrong.

```
#Weight vs Speed (organized by Generation)
spdvswgt <- ggplot(pokemon, aes(x =weight, y =speed, color= gen_introduced)) +
  geom_point()+
  labs(title="Weight vs Speed", x= "Weight", y="Speed")+
  xlim(c(0, max(pokemon$weight))) +
  ylim(c(0, max(pokemon$speed)))+
  viridis::scale_color_viridis()


#Speed vs Height (organized by Generation)
spdvshgt <- ggplot(pokemon, aes(x =speed, y =height, color= gen_introduced)) +
  geom_point()+
  labs(title="Speed vs Height", x= "Speed", y="Height")+
  xlim(c(0, max(pokemon$speed))) +
  ylim(c(0, max(pokemon$height)))+
  viridis::scale_color_viridis()+
  theme(plot.title = element_text(hjust = 0.5))

spdvswgt
```




```
spdvshgt
```

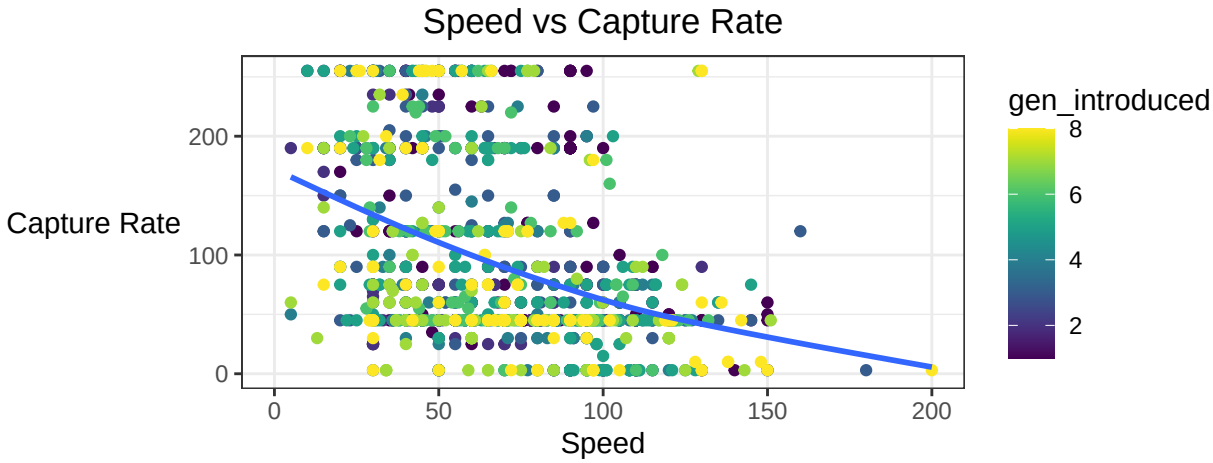


Looking at the graph for weight and speed, we noticed that there are outliers (on the right side of the graph) that do not make sense for speed. To see whether this was a mistake or not, we looked at the generation of each Pokemon and found that the majority of the outliers came from the most recent generation rather than the earlier generations when Pokemon's system of attributes were still in the works. In addition to weight, height did not have an affect on the speed of a Pokemon, thus defying all of our expectations of this data set.

Afterwards, we wanted to see if the speed of a Pokemon affected its overall capture rate and whether it is harder to catch a Pokemon if it has a higher speed or not.

```
#Speed vs Capture Rate
ggplot(pokemon, aes(x =speed, y =capture_rate,  color= gen_introduced)) +
  geom_point()+
  labs(title="Speed vs Capture Rate", x= "Speed", y="Capture Rate")+
  xlim(c(0, max(pokemon$speed))) +
  ylim(c(0, max(pokemon$capture_rate)))+
  geom_smooth(se=FALSE)+
  viridis::scale_color_viridis()+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

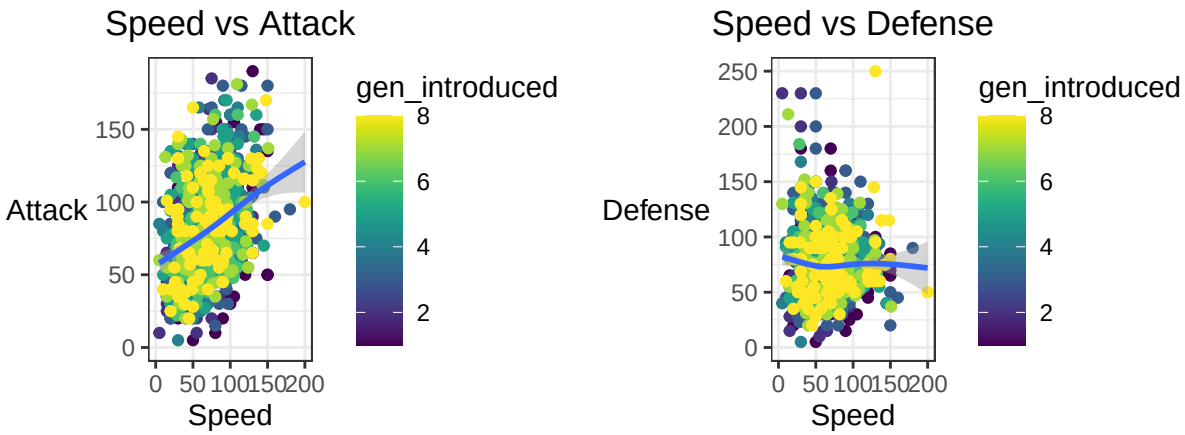


The plot shows a general negative trend as the speed increases, the capture rate decreases.

Finally with speed, since there was no realistic interpretation of speed for a given Pokemon, we decided to see if there was a relationship between a Pokemon's speed of attack and defense.

```
#Speed vs Attack
spdvsatk <-ggplot(pokemon, aes(x =speed, y =attack,  color= gen_introduced)) +
  geom_point()+
  labs(title="Speed vs Attack", x= "Speed", y="Attack")+
  xlim(c(0, max(pokemon$speed))) +
  ylim(c(0, max(pokemon$attack)))+
  geom_smooth()+
  viridis::scale_color_viridis()+
  theme(plot.title = element_text(hjust = 0.5))

#Speed vs Defense
spdvsdef <-ggplot(pokemon, aes(x =speed, y =defense,  color= gen_introduced)) +
  geom_point()+
  labs(title="Speed vs Defense", x= "Speed", y="Defense")+
  xlim(c(0, max(pokemon$speed))) +
  ylim(c(0, max(pokemon$defense)))+
  geom_smooth()+
  viridis::scale_color_viridis()+
  theme(plot.title = element_text(hjust = 0.5))

spdvsatk+spdvsdef
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



The results found that attack tends to have a slight positive correlation with speed while defense does not.

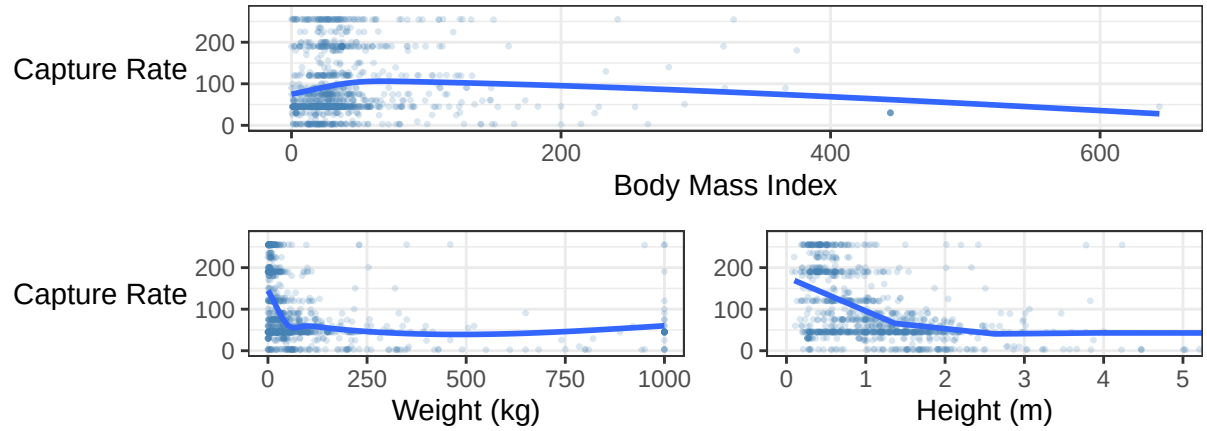
Capture Rate

We also wanted to look into how capture rate is effected by variables such as height, weight, gender, and legendary. Capture rate describes how easy it is to capture a Pokémon found in the wild. The lower the capture rate, the harder the Pokémon is to capture. Initially our belief was that the larger the Pokémon, the harder it would be to capture so we decided to compare capture rate to BMI, Height and Weight. We then wanted to see how certain categorical variables affected capture rate in the hopes of finding a pattern in how the game developers determined capture rate.

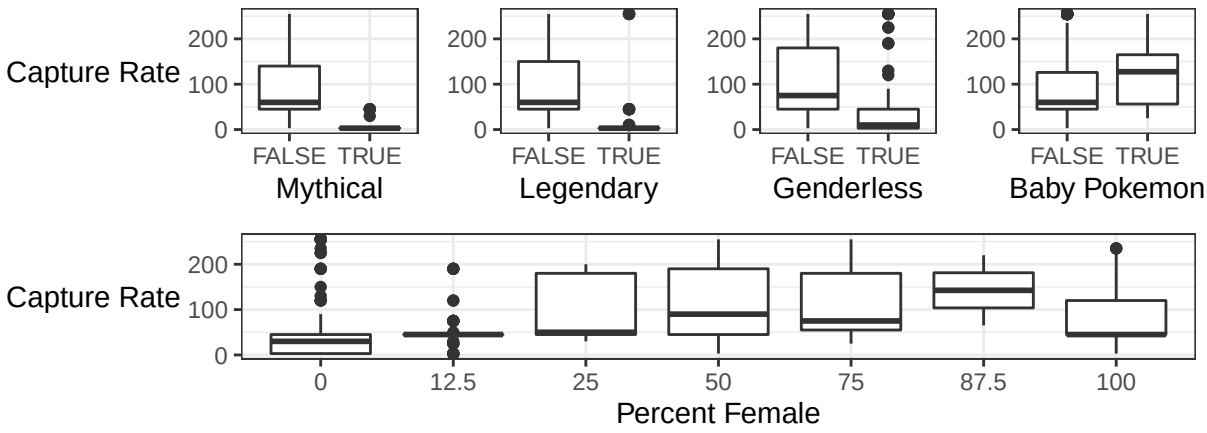
```
#Create a plot that shows the relationship between capture_rate and BMI
#Limit to not include a lot of empty space
BMI <- ggplot(pokemon_new, aes(BMI, capture_rate)) +
  geom_jitter(alpha = 0.2, color = "Steel Blue", size = .5) +
  geom_smooth(se = FALSE) +
  coord_cartesian(ylim = c(0,275)) +
  labs(x = "Body Mass Index", y = "Capture Rate")


#Create a plot that shows the relationship between capture_rate and Height
#Limit included to allow for best view of relationship
Height <- ggplot(pokemon_new, aes(height/10, capture_rate)) +
  geom_jitter(alpha = 0.2, color = "Steel Blue", size = .5) +
  geom_smooth(se = FALSE) +
  coord_cartesian(xlim = c(0,5), ylim = c(0,275)) +
  labs(x = "Height (m)", y = NULL)

#Create a plot that shows the relationship between capture_rate and weight
#Limit included to allow for best view of relationship

Weight <- ggplot(pokemon_new, aes(weight/10, capture_rate)) +
  geom_jitter(alpha = 0.2, color = "Steel blue", size = .5) +
  geom_smooth(alpha = 0.5, se = FALSE) +
  coord_cartesian(ylim = c(0,275)) +
  labs(x = "Weight (kg)", y = "Capture Rate")
```

```
BMI / (Weight | Height)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
p1 <- ggplot(pokemon_new) +
  geom_boxplot(aes(reorder(shape, capture_rate, FUN = median), capture_rate)) +
  labs(x = "Shape", y = "Capture Rate")

p2 <- ggplot(pokemon_new, aes(mythical, capture_rate)) +
  geom_boxplot() +
  labs(x = "Mythical", y = "Capture Rate")

p3 <- ggplot(pokemon_new, aes(legendary, capture_rate)) +
  geom_boxplot() +
  labs(x = "Legendary", y = NULL)

p4 <- ggplot(pokemon_new, aes(genderless, capture_rate)) +
  geom_boxplot() +
  labs(x = "Genderless", y = NULL)

p5 <- ggplot(pokemon_new, aes(baby_pokemon, capture_rate)) +
  geom_boxplot() +
  labs(x = "Baby Pokemon", y = NULL)

p6 <- ggplot(pokemon_new, aes(as.factor(female_rate*100), capture_rate)) +
  geom_boxplot() +
  labs(x = "Percent Female", y = "Capture Rate")

(p2 | p3 | p4| p5) / p6
```

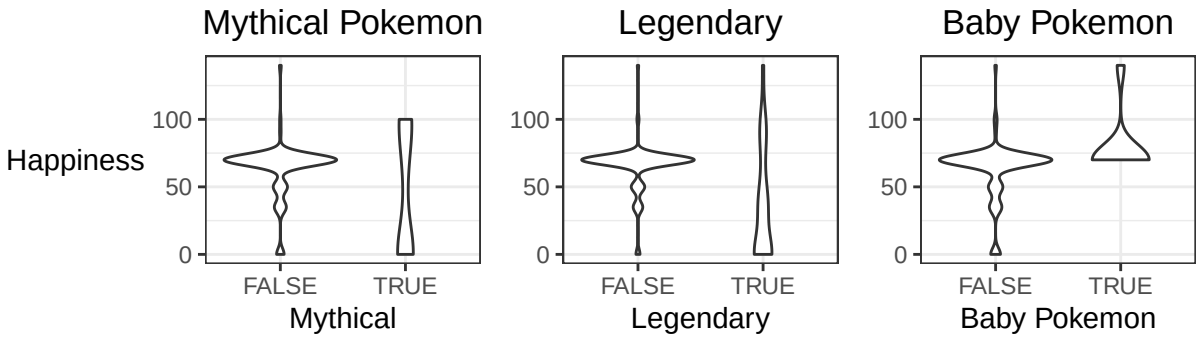


Happiness

Lastly we wanted to see what factors effected the happiness of the Pokemon. This variable was just something we thought would be fun to look into. Our main questions were what other variables effect the Pokemon's happiness. We began with looking at if the Pokemon is a legendary and if it's a mythical Pokemon.

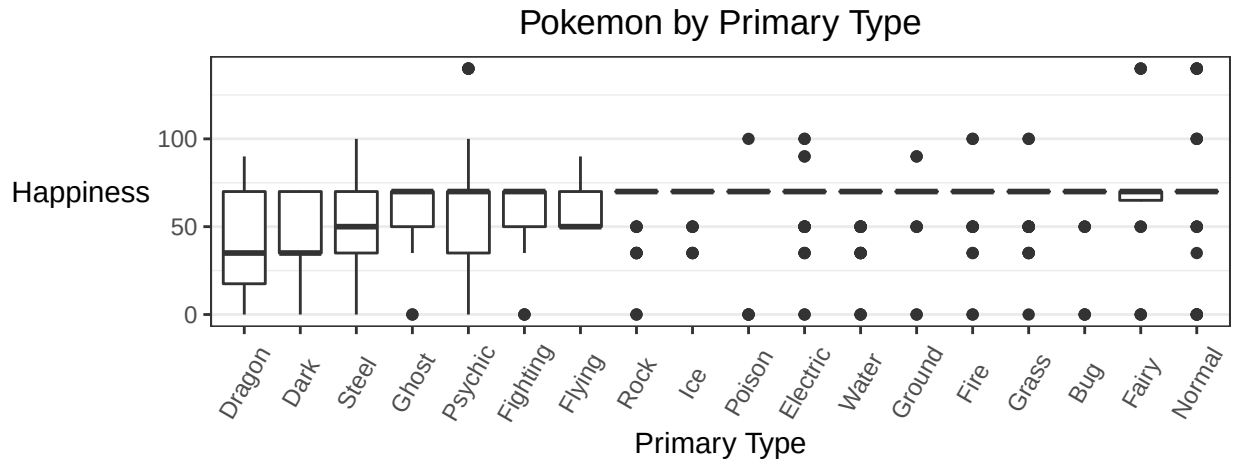
To get a grasp of what effects different variables had on the happiness of the Pokemon we took a look at how different characteristics of the Pokemon had an effect. Bellow are three violin plots looking at mythical, legendary, and baby Pokemon and comparing them to Pokemon without that attribute.

```
v1 <- pokemon_new |> ggplot(aes(mythical, base_happiness)) +
  geom_violin() +
  labs(title = "Mythical Pokemon", x = "Mythical", y = "Happiness")

v2 <- pokemon_new |> ggplot(aes(legendary, base_happiness)) +
  geom_violin() +
  labs(title = "Legendary", x = "Legendary", y = NULL)

v3 <- pokemon_new |> ggplot(aes(baby_pokemon, base_happiness)) +
  geom_violin() +
  labs(title = "Baby Pokemon", x = "Baby Pokemon", y = NULL)

(v1 + v2 + v3)
```



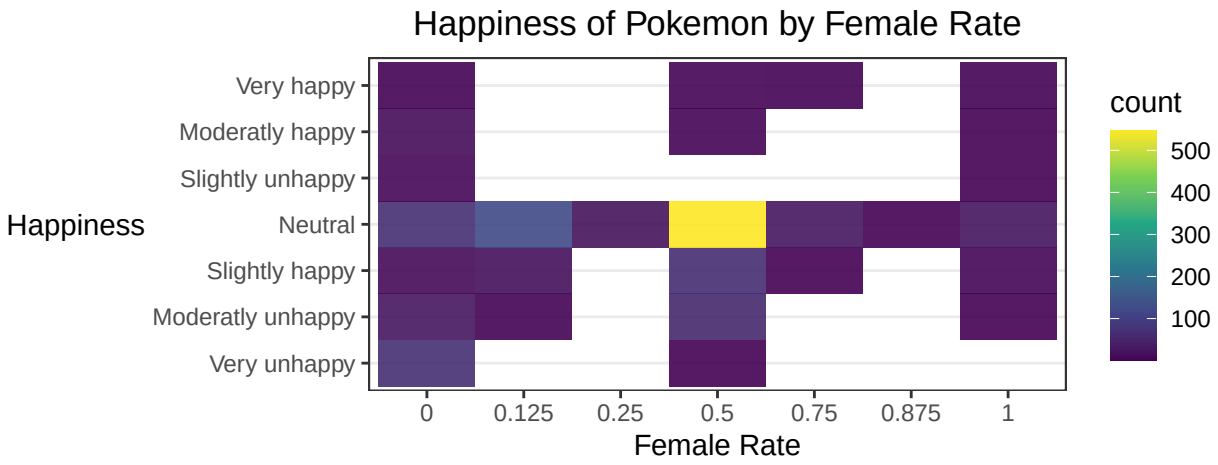
We also looked at how the Pokemon primary type effects it's happiness. Looking at the box plot below we can see that there are diferent distributions of happiness bases upon primary type. Notable the only type with all possible values for happiness is Normal type, and the type with the least variability in all data points is flying type, but they still aren't very happy.

```
b4 <- pokemon_new |> ggplot(aes(reorder(primary, base_happiness, mean), base_happiness)) +
  geom_boxplot() +
  labs(title = "Pokemon by Primary Type", x = "Primary Type",
       y = "Happiness") +
  theme(
    axis.text.x.bottom = element_text(angle = 60, vjust = .5, hjust = 0.5),
    panel.grid.major.x = element_blank()
    )

b4
```



We were also curious if the female rate effected happiness of the Pokemon. Converting the happiness variable to a likert scale we can easily see the amount of data points in each bin of female rate in the following plot. As seen in the plots before there is a smaller amount of Pokemon with a high female rate. Notice how it appears that depending of the female rate there are different levels of happiness that are observed. What stood out to us what that Pokemon with a female rate of 0.25 or 0.875 are only observed with neutral happiness and there are no very unhappy Pokemon with a rate of 1. There is no apparent correlation between happiness and female rate.

```
pokemon_new |> ggplot(aes(as.factor(female_rate), happiness)) +
  geom_bin2d(alpha = .9) +
  scale_fill_viridis_c() +
  labs(title = "Happiness of Pokemon by Female Rate", x = "Female Rate", y = "Happiness") +
  theme(panel.grid.major.x = element_blank())
```


Happiness of Pokemon by Female Rate
Happiness
Very happy
Moderatly happy
Slightly unhappy
Neutral
Slightly happy
Moderatly unhappy
Very unhappy
0
0.125
0.25
0.5
0.75
0.875
1
Female Rate
count
500
400
300
200
100