Predictors of High CP in Four Species of Pokemon in Pokemon GO.

Suzanna S.

2024-04-05

Introduction

Pokemon Go is an augmented-reality mobile game where players can catch hundreds of Pokemon by moving through their physical environment. A key element of the game is the ability to evolve Pokemon, unlocking a new aesthetic as well as increasing their fighting ability.

We will be analyzing the 'pokemon_go' data set from openintro.org/data. This data set provides evolutions of 75 pokemon of 4 species and their characteristics. We seek to answer the following four questions about the data:

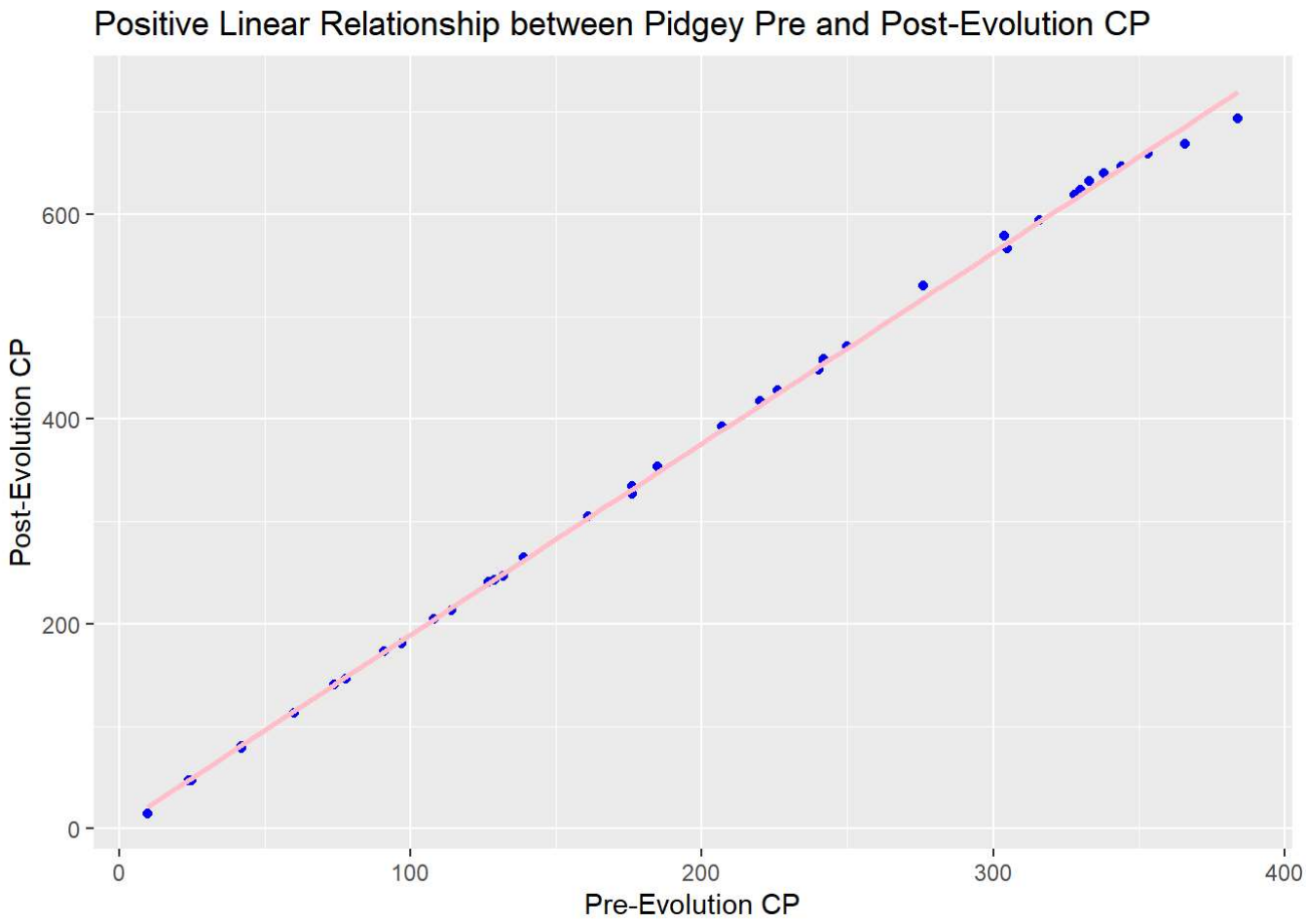
1. What characteristics are important in predicting a Pokémon's combat power (CP) following the evolution?
2. How reliable is the prediction?
3. Is post-evolution CP linearly related to the Pokémon's pre-evolution CP?
4. Is the model the same across different Pokémon species?

Pidgey Analysis

Pidgey is a pigeon-like pokemon that evolves into Pidgeotto. To test which numerical variables are related to final CP, we generate scatter plots and test a linear model. CP, HP, and stardust required to level up are numerical variables unrelated to attacks, so these make the most sense to test for relation to higher post-evolution CP. Let's see if these have a correlation to evolved CP:

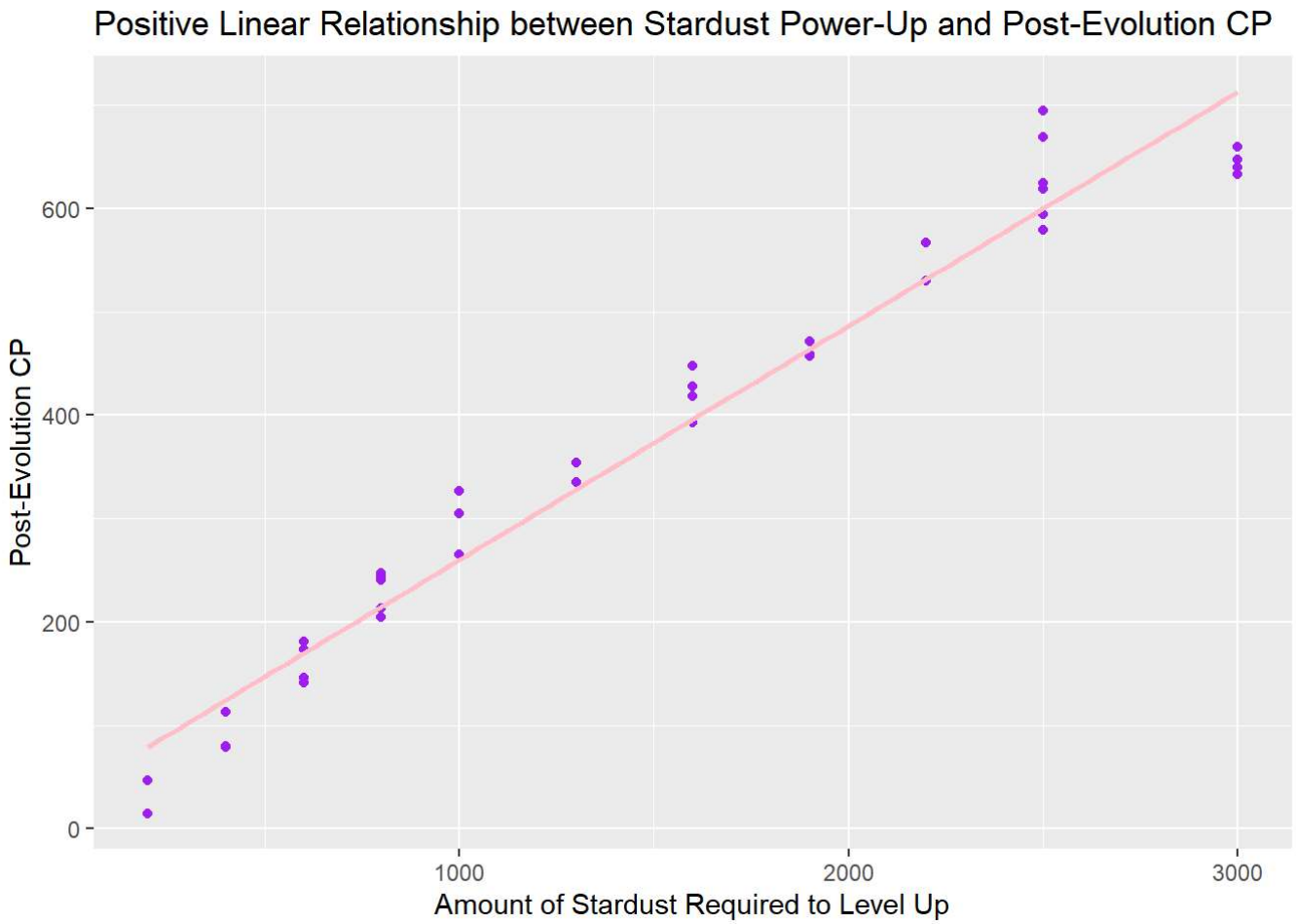
```
# Dataframe of only pidgey pokemon

pidgey <- pokemon_go |> filter(species=='Pidgey')

# Plots for EDA

ggplot(pidgey, aes(x = cp, y = cp_new)) +
  geom_point(shape = "circle", color = "blue") +
  labs(
    title = "Positive Linear Relationship between Pidgey Pre and Post-Evolution CP",
    x = "Pre-Evolution CP",
    y = "Post-Evolution CP"
  ) +
  geom_smooth(method=lm, se = FALSE, color = "pink")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



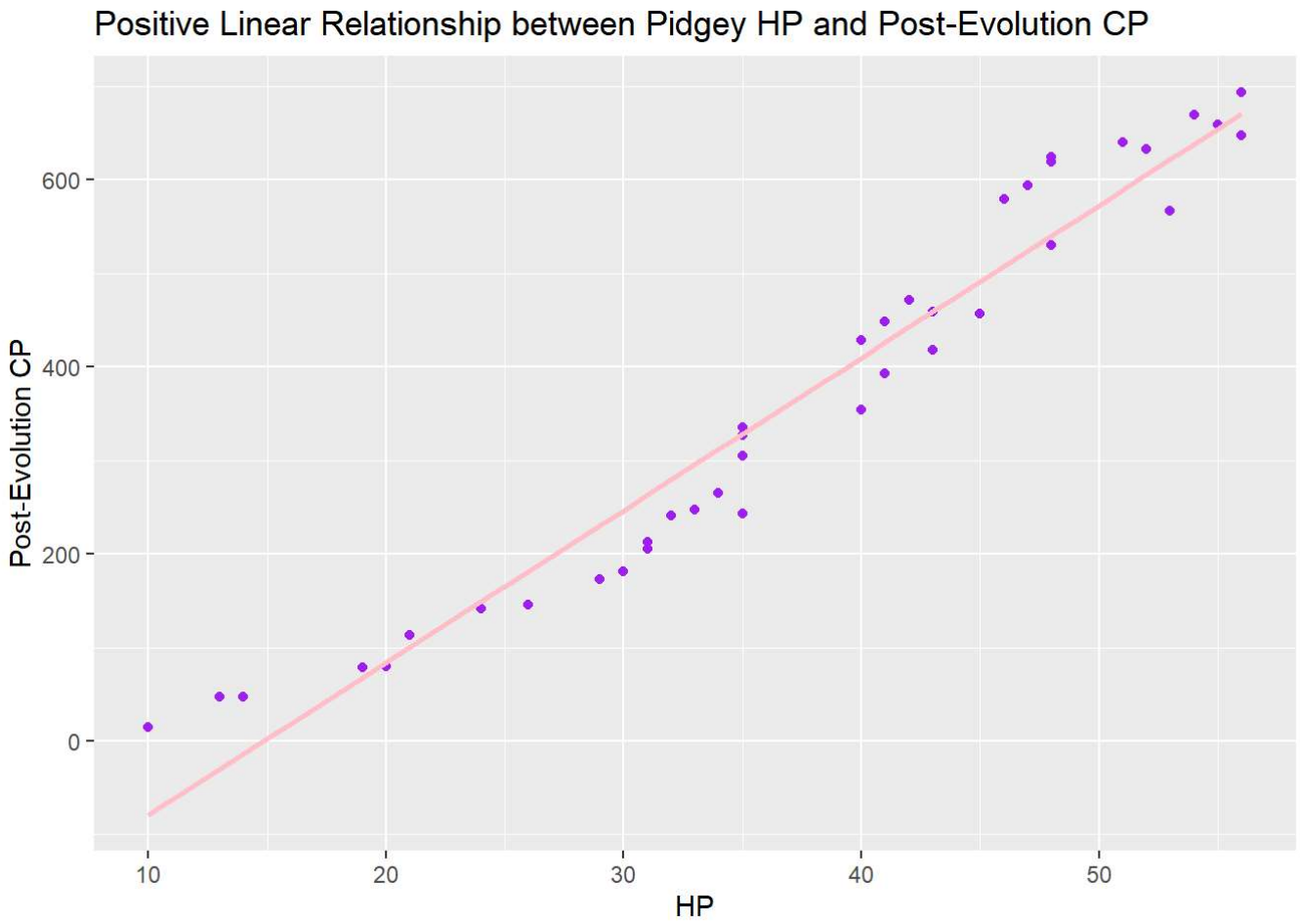

```
ggplot(pidgey, aes(x = power_up_stardust, y = cp_new)) +
  geom_point(shape = "circle", color = "purple") +
  labs(
    title = "Positive Linear Relationship between Stardust Power-Up and Post-Evolution CP",
    x = "Amount of Stardust Required to Level Up",
    y = "Post-Evolution CP"
  ) +
  geom_smooth(method=lm, se = FALSE, color = "pink")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

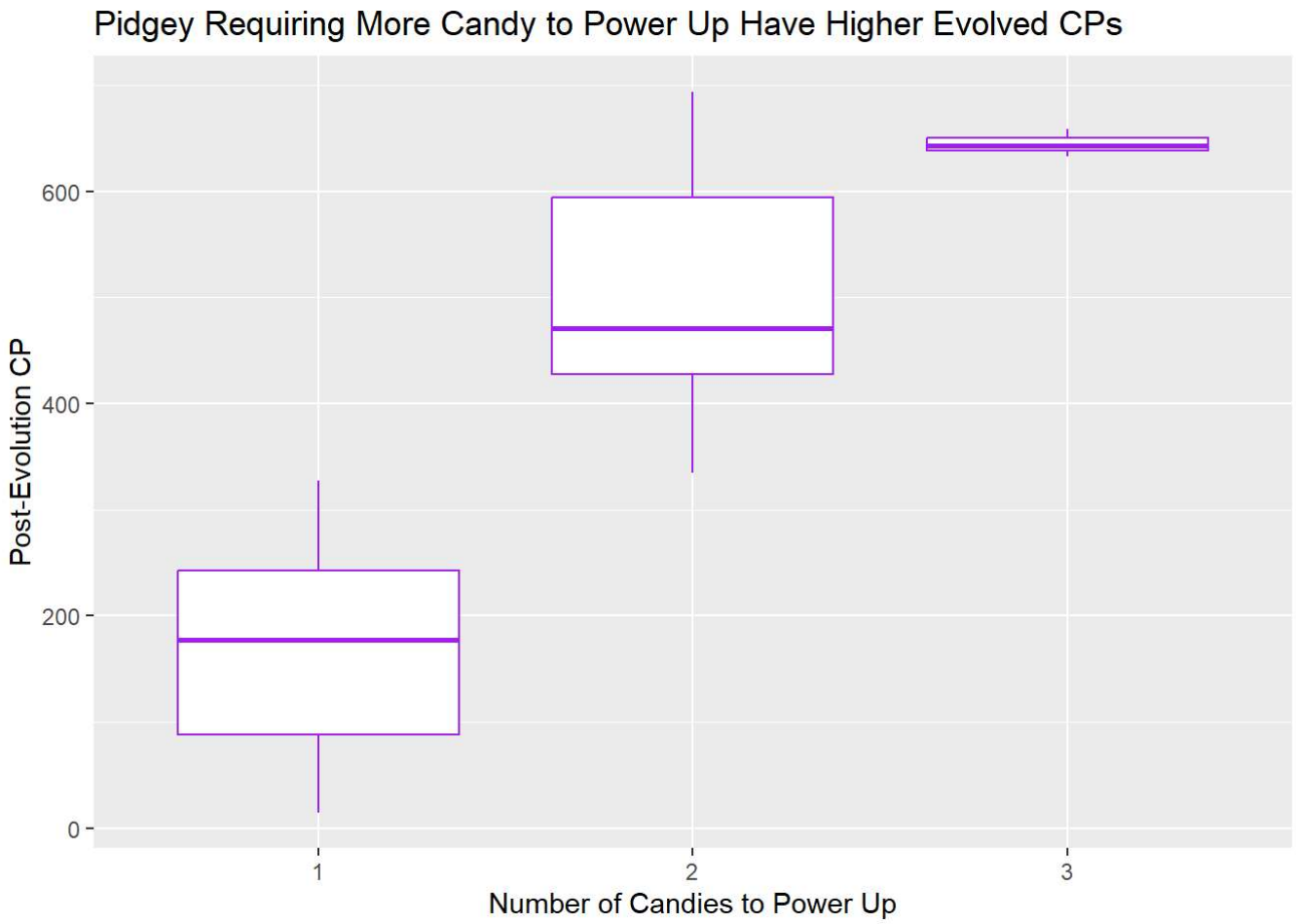



```
ggplot(pidgey, aes(x = hp, y = cp_new)) +
  geom_point(shape = "circle", color = "purple") +
  labs(
    title = "Positive Linear Relationship between Pidgey HP and Post-Evolution CP",
    x = "HP",
    y = "Post-Evolution CP"
  ) +
  geom_smooth(method=lm, se = FALSE, color = "pink")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



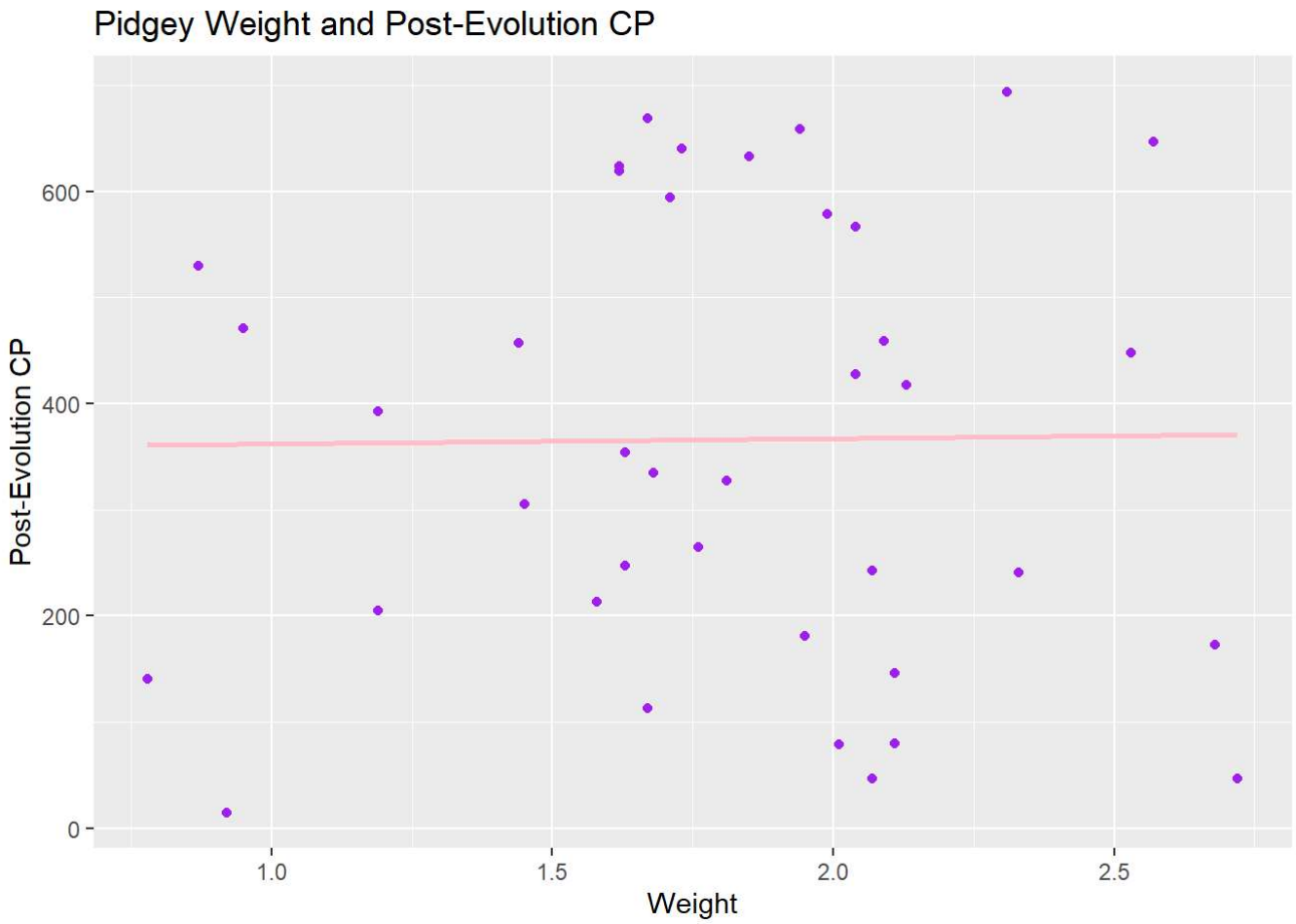

```
# make power up candy into a categorical variable

pidgey$candy <- as.factor(ifelse(pidgey$power_up_candy == 1, '1',
                                 ifelse(pidgey$power_up_candy == 2, '2',
                                        ifelse(pidgey$power_up_candy == 3,'3','0'))))

ggplot(pidgey, aes(x = candy, y = cp_new)) +
  geom_boxplot(shape = "circle", color = "purple") +
  labs(
    title = "Pidgey Requiring More Candy to Power Up Have Higher Evolved CPs",
    x = "Number of Candies to Power Up",
    y = "Post-Evolution CP")
```



For fun, let's check if larger Pidgey have higher CP post-evolution; if true, this would incentivize catching the largest Pidgey possible!

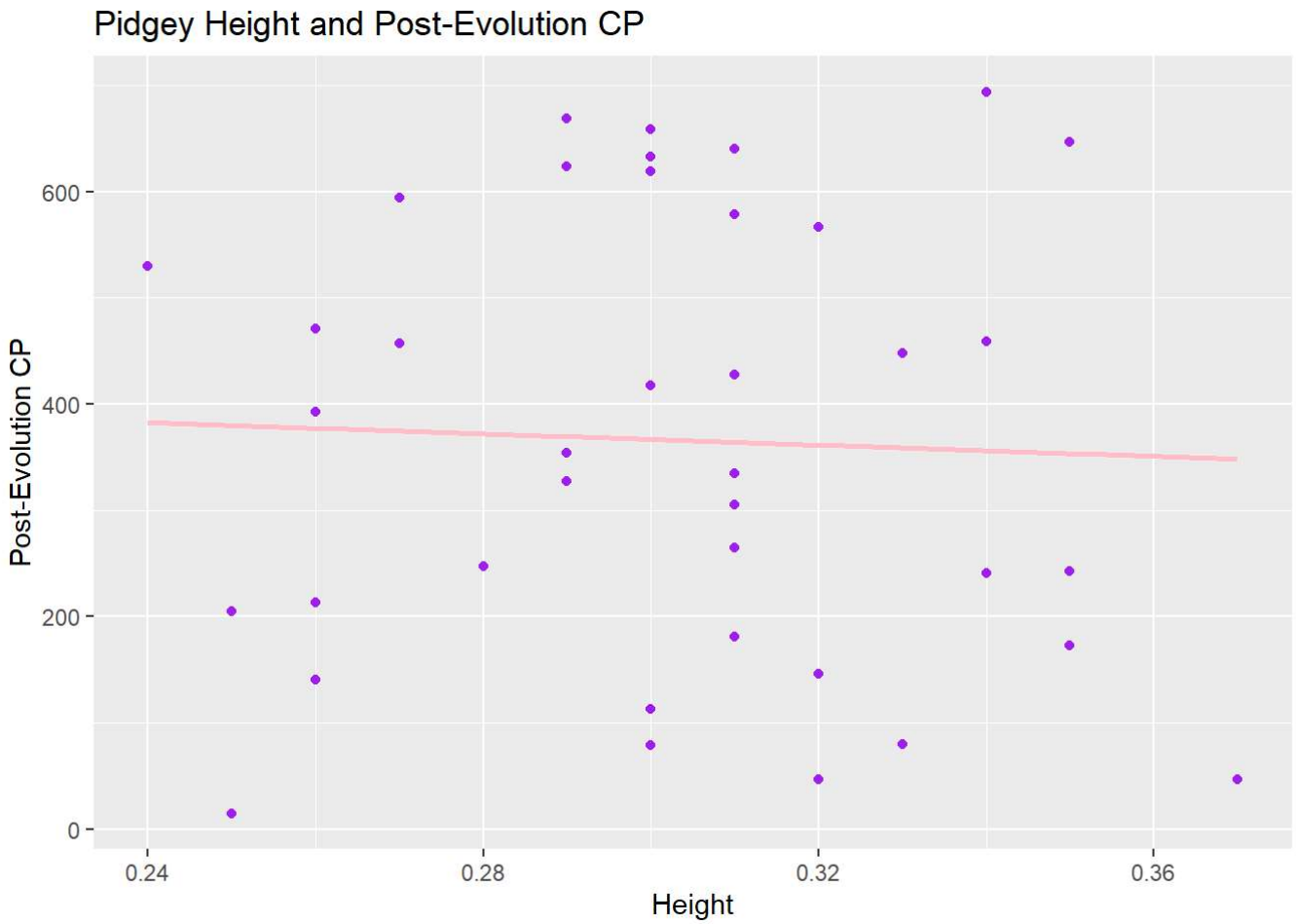
```
ggplot(pidgey, aes(x = weight, y = cp_new)) +
  geom_point(shape = "circle", color = "purple") +
  labs(
    title = "Pidgey Weight and Post-Evolution CP",
    x = "Weight",
    y = "Post-Evolution CP",
    color = "Species"
  ) +
  geom_smooth(method=lm, se = FALSE, color = "pink")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(pidgey, aes(x = height, y = cp_new)) +
  geom_point(shape = "circle", color = "purple") +
  labs(
    title = "Pidgey Height and Post-Evolution CP",
    x = "Height",
    y = "Post-Evolution CP",
    color = "Species"
  ) +
  geom_smooth(method=lm, se = FALSE, color = "pink")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



There seems to be no correlation between Pidgey size and post-evolution CP.

Multiple Regression Model for Pidgey

We return to our hypothesis about HP, CP, candies and stardust in predicting final CP. We perform a p-value analysis to determine which of these explanatory variables is most important in determining post-evolution CP. We begin with a backward-elimination approach.

```
lm_pidgey_mult <- lm(pidgey$cp_new ~ pidgey$cp + pidgey$hp + pidgey$power_up_stardust + pidge
y$power_up_candy)
summary(lm_pidgey_mult)
```

```
##
## Call:
## lm(formula = pidgey$cp_new ~ pidgey$cp + pidgey$hp + pidgey$power_up_stardust +
##     pidgey$power_up_candy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7099  -3.1944  -0.3228   2.9648   8.8052
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -5.197717   5.734859  -0.906  0.37114
## pidgey$cp                 1.629354   0.060549  26.910  < 2e-16 ***
## pidgey$hp                 0.614149   0.306101   2.006  0.05282 .
## pidgey$power_up_stardust  0.022105   0.007127   3.101  0.00386 **
## pidgey$power_up_candy    -0.709929   3.897251  -0.182  0.85654
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.375 on 34 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9993
## F-statistic: 1.456e+04 on 4 and 34 DF,  p-value: < 2.2e-16
```

The power-up candy variable has a p-value of 0.85 > 0.05, so we remove it from the model.

```
lm_pidgey_new <- lm(pidgey$cp_new ~ pidgey$cp + pidgey$hp + pidgey$power_up_stardust)
summary(lm_pidgey_new)
```

```
##
## Call:
## lm(formula = pidgey$cp_new ~ pidgey$cp + pidgey$hp + pidgey$power_up_stardust)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7457  -3.1083  -0.3387   2.8066   8.9386
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -5.645515   5.109212  -1.105   0.2767
## pidgey$cp                 1.634010   0.054127  30.188  < 2e-16 ***
## pidgey$hp                 0.608975   0.300541   2.026   0.0504 .
## pidgey$power_up_stardust  0.021132   0.004649   4.545 6.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.3 on 35 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9994
## F-statistic: 1.997e+04 on 3 and 35 DF,  p-value: < 2.2e-16
```

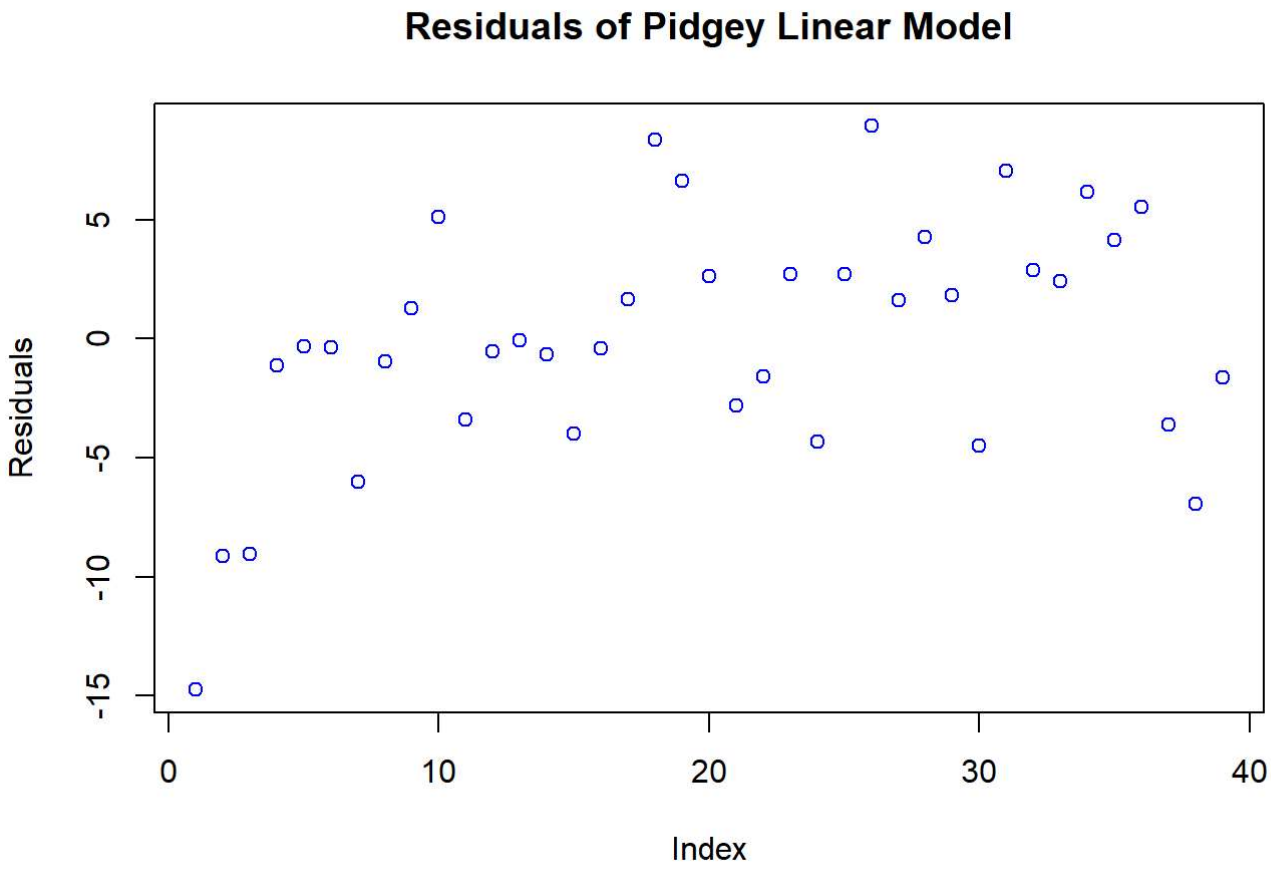
This improved the adjusted R-squared, so we are justified in removing it. Since the Hit Point (HP) variable has a p-value of 0.0504 > 0.05, we remove this next.

```
lm_pidgey_new <- lm(pidgey$cp_new ~ pidgey$cp + pidgey$power_up_stardust)
summary(lm_pidgey_new)$adj.r.squared
```

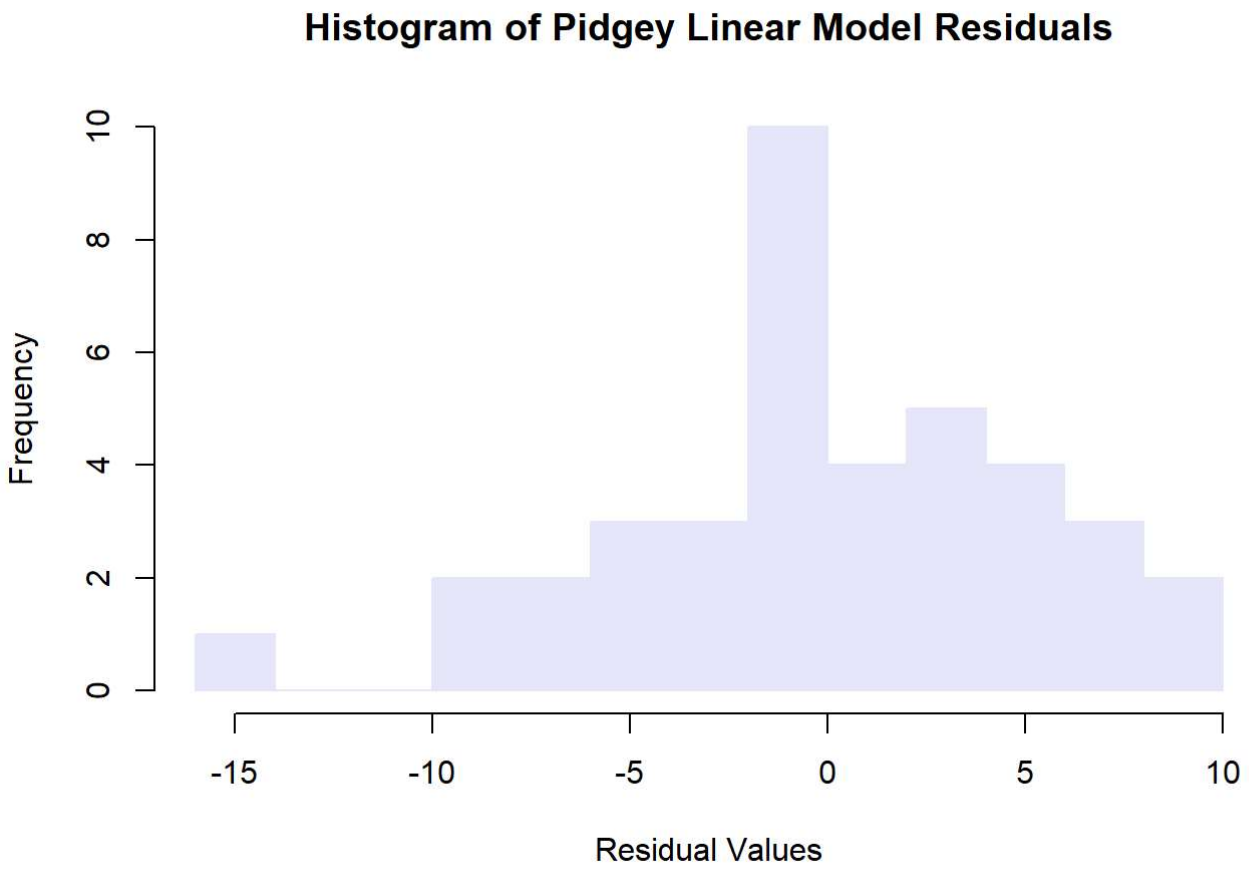
```
## [1] 0.9993113
```

Note that we see a decrease in adjusted R-squared, so we keep the HP variable in the model. We find the best linear model from incorporating CP, power up stardust, and HP. The R-squared value tells us that 99.9% of variability in evolved CP can be explained by variations in original CP. Let's test if a least squares model is appropriate for CP, stardust, and HP vs New CP:

```
# linear model for pidgey cp pre & post evolution
lm_pidgey <- lm(pidgey$cp_new ~ pidgey$cp + pidgey$power_up_stardust + pidgey$hp)

# plot of pidgey residuals
plot(lm_pidgey$residuals, ylab = "Residuals", main = "Residuals of Pidgey Linear Model", col
= 'blue')
```




```
hist(lm_pidgey$residuals, breaks = 10, main = "Histogram of Pidgey Linear Model Residuals", x
lab = "Residual Values", ylab = "Frequency", col = 'lavender', border = 'lavender')
```



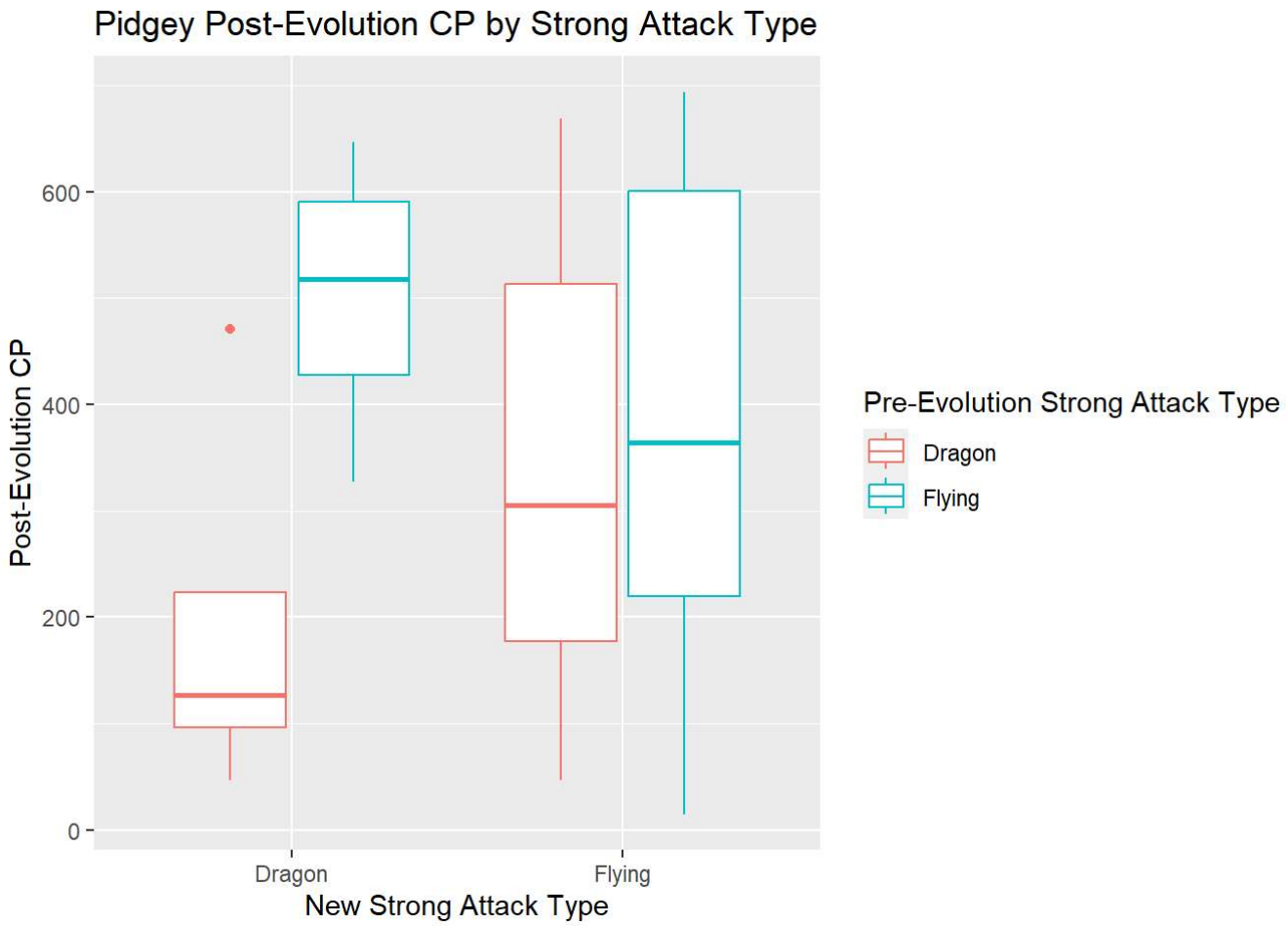
With the linearity condition already satisfied, we check the residual plot, which shows a slight curve not apparent from the data. More advanced techniques should be used to further analyze this. The histogram shows that residuals are nearly normal, though there is one outlier with a residual of -15. Since the CP of one Pidgey caught in the wild is independent of another, observations are independent.

Overall, the best fitting multiple regression model for predicting post-evolution CP of Pidgey depends on pre-evolution CP, HP, and stardust required to power up. With this model we found that 99.9% of variation in post-evolution CP could be explained by variation in these predictor variables. However, we are skeptical of the validity of a linear model due to curvature in the residual plot. More advanced work is needed for a definitive answer, though for our purposes we will default to the results from this model as 1) the adjusted R-squared is incredibly high, and 2) we are saving more advanced techniques for a more advanced class in statistics.

Pidgey Attack Type Analysis

We can also consider if strong attack types can influence the final CP of Pidgey Pokemon. Let's make a plot to look for a correlation:

```
ggplot(pidgey, aes(x = attack_strong_type_new, y = cp_new, color = attack_strong_type)) +
  geom_boxplot() +
  labs(
    title = "Pidgey Post-Evolution CP by Strong Attack Type",
    x = "New Strong Attack Type",
    y = "Post-Evolution CP",
    color = "Pre-Evolution Strong Attack Type"
  )
```



Pidgey with a new strong attack type of “Flying” have similar final CPs whether their pre-evolution strong attack type was “Flying” or “Dragon.” However, we see that new strong attack type of “Dragon” has more variation depending on what the pre-evolution strong attack type was. Since the IQRs do not overlap, we predict a significant difference between Dragon-type evolved Pidgey CP depending on pre-evolution strong attack type.

We can use a t-distribution to create a hypothesis test for difference of means to quantitatively answer the question: Is there a significant difference between final CPs of evolved “Dragon” type Pidgey Pokemon depending on pre-evolution strong attack types?

First, we check conditions: We require that observations are independent between each other and within groups and that normality condition is satisfied for each group.

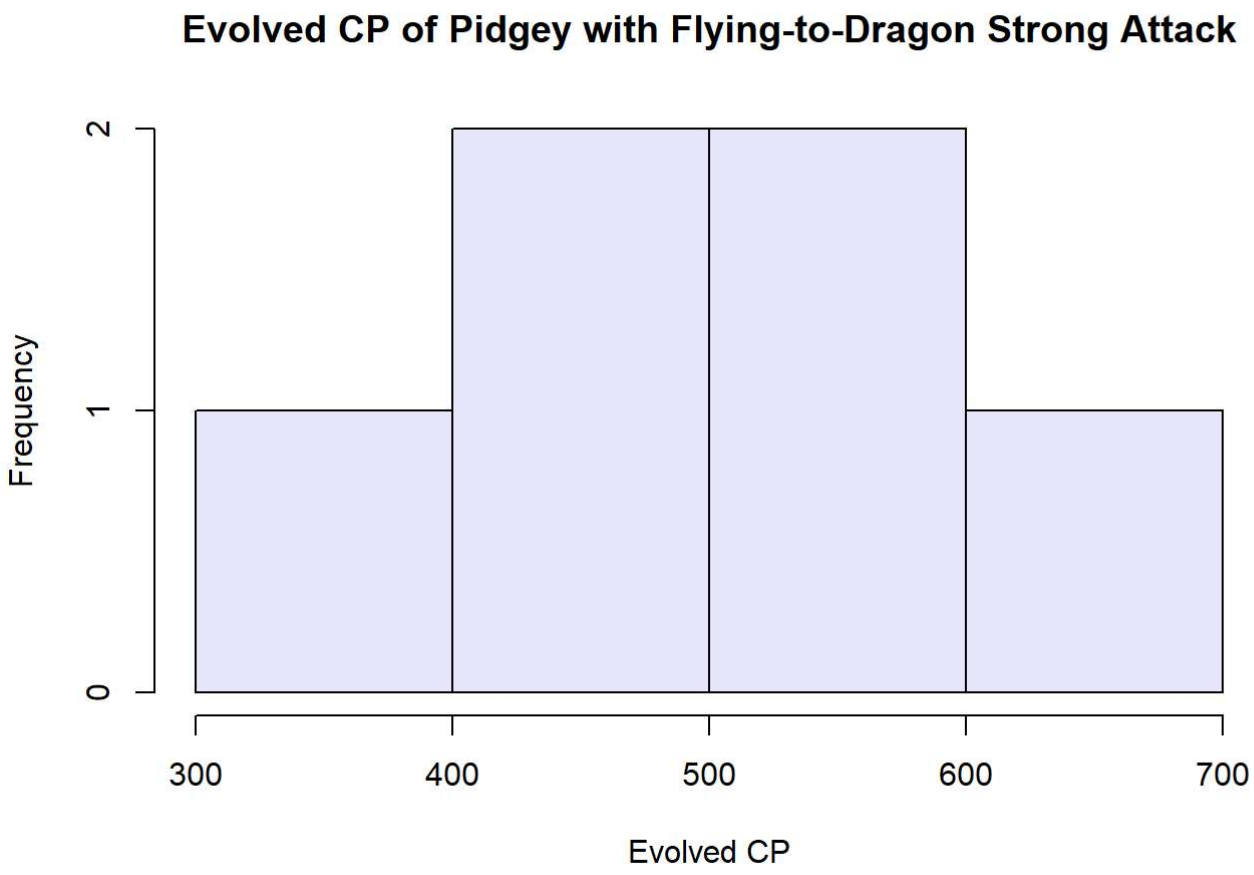
```
# create new dataframes
drag_pidgey <- pidgey |> filter(attack_strong_type_new == "Dragon", attack_strong_type == "Dr
agon")
fly_pidgey <- pidgey |> filter(attack_strong_type_new == "Dragon", attack_strong_type == "Fly
ing")

# normality condition varies depending on if n > 30
count(fly_pidgey) # 6
```

```
##   n
## 1 6
```

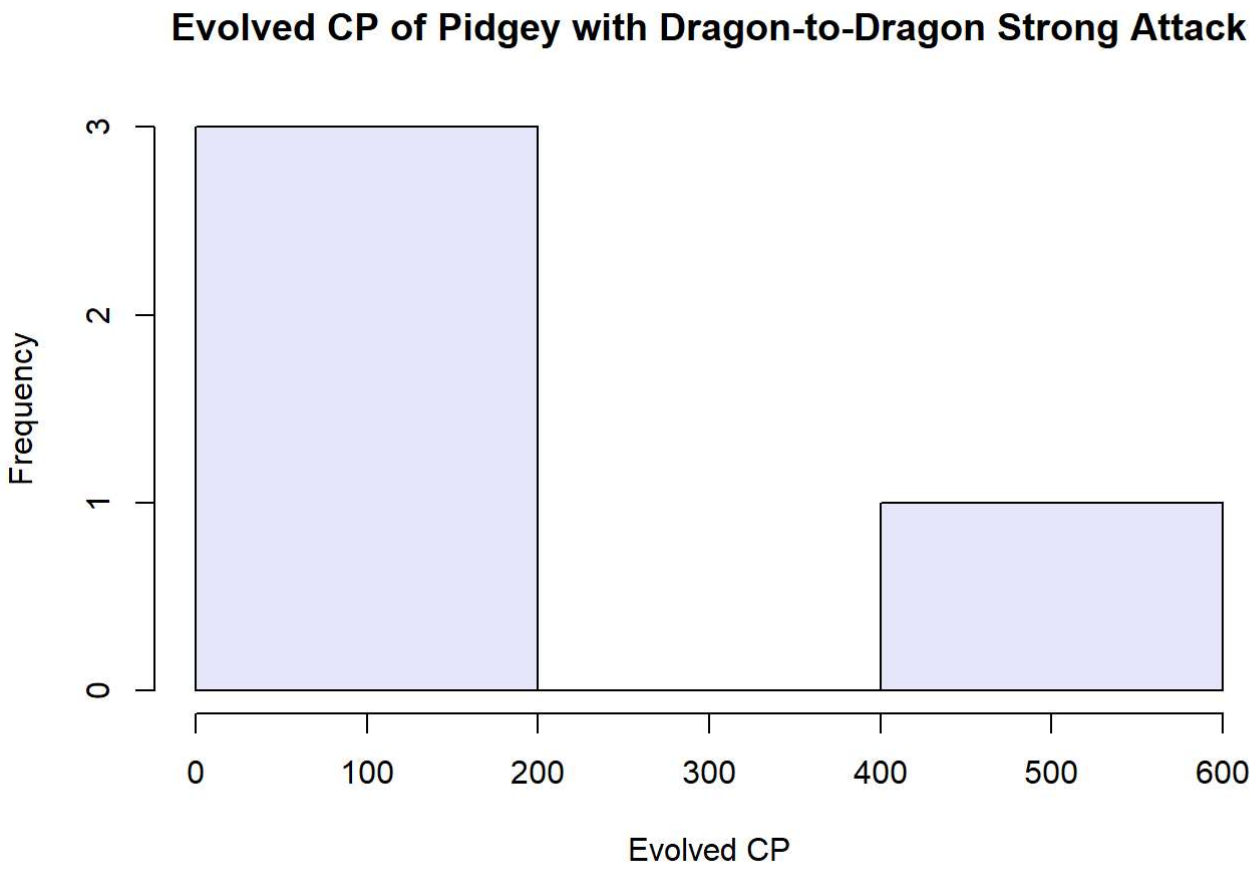
```
count(drag_pidgey) # 4
```

```
##   n
## 1 4
```

```
# check for outliers
hist(fly_pidgey$cp_new, main = "Evolved CP of Pidgey with Flying-to-Dragon Strong Attack", xl
ab = "Evolved CP", col = 'lavender')
```




```
hist(drag_pidgey$cp_new, main = "Evolved CP of Pidgey with Dragon-to-Dragon Strong Attack", x
lab = "Evolved CP", col = 'lavender')
```



We don't see any clear outliers in the data for the flying-to-dragon Pidgey. There is one outlier for the dragon-to-dragon Pidgey, and considering that there are only four observations this could cause problems with the normality condition using the t-distribution. However, consider that this outlier has a higher CP than the other observations, but generally the dragon-to-dragon Pidgey have a lower CP than the flying-to-dragon Pidgey. This outlier will increase the average dragon-to-dragon Pidgy CP, bringing the two means closer together. If we perform a t-test and still find significance, than we can be sure that this outlier does not impact the validity of the results. If we do not find significance, we can remove the outlier and perform the test again.

Since our sample was random, independence is satisfied.

We proceed with a difference in means hypothesis test using the t-distribution.

H_0 : There is no difference in mean CPs for Pidgey with a Dragon-type strong attack depending on pre-evolution strong attack type - ($\mu_{flying} - \mu_{dragon} = 0$).

H_a : There is a difference in mean CPs for Pidgey with a Dragon-type strong attack depending on pre-evolution strong attack type - ($\mu_{flying} - \mu_{dragon} \neq 0$).

```
# mean post-evolution CPs for both
fly_avgcp <- mean(fly_pidgey$cp_new)
drag_avgcp <-mean(drag_pidgey$cp_new)

# standard deviations
fly_SD <- sd(fly_pidgey$cp_new)
drag_SD <- sd(drag_pidgey$cp_new)

# counts
n_fly <- count(fly_pidgey)
n_drag <- count(drag_pidgey)

# sample proportion and standard error
p_hat <- fly_avgcp - drag_avgcp
pidgey_SE <- sqrt((fly_SD^2/n_fly) + drag_SD^2/n_drag)

# degrees of freedom
df_fly <- n_fly - 1
df_drag <- n_drag - 1
df <- min(df_fly, df_drag)

# calculate t-score and p-value
T_score <- p_hat/pidgey_SE
p_val <- 2*(1-pt(2.899642,3))

# output
if (p_val < 0.05){
  print("Reject the null: For Dragon-type evolved Pidgey, there is a significant difference b
etween post-evolution CPs depending on pre-evolution strong attack type.")
} else{
  print("Cannot reject the null: For Dragon-type evolved Pidgey, there is not a significant d
ifference between post-evolution CPs depending on pre-evolution strong attack type.")
}
```

```
## [1] "Cannot reject the null: For Dragon-type evolved Pidgey, there is not a significant di
fference between post-evolution CPs depending on pre-evolution strong attack type."
```

Our hypothesis test shows that we cannot reject the null hypothesis, but observe that the p-value is not very far from 0.05. We can try this again and remove the outlier Dragon-to-Dragon Pidgey to see if this impacted our results at all.

```
drag_pidgey <- pidgey |> filter(attack_strong_type_new == "Dragon", attack_strong_type == "Dr
agon", cp_new < 400)

# mean post-evolution CPs for both
fly_avgcp <- mean(fly_pidgey$cp_new)
drag_avgcp <-mean(drag_pidgey$cp_new)

# standard deviations
fly_SD <- sd(fly_pidgey$cp_new)
drag_SD <- sd(drag_pidgey$cp_new)

# counts
n_fly <- count(fly_pidgey)
n_drag <- count(drag_pidgey)

# sample proportion and standard error
p_hat <- fly_avgcp - drag_avgcp
pidgey_SE <- sqrt((fly_SD^2/n_fly) + drag_SD^2/n_drag)

# degrees of freedom
df_fly <- n_fly - 1
df_drag <- n_drag - 1
df <- min(df_fly, df_drag)

# calculate t-score and p-value
T_score <- p_hat/pidgey_SE
p_val <- 2*(1-pt(7.042482,2))
p_val
```

```
## [1] 0.01957268
```

```
# output
if (p_val < 0.05){
  print("Reject the null: For Dragon-type evolved Pidgey, there is a significant difference b
etween post-evolution CPs depending on pre-evolution strong attack type.")
} else{
  print("Cannot reject the null: For Dragon-type evolved Pidgey, there is not a significant d
ifference between post-evolution CPs depending on pre-evolution strong attack type.")
}
```

```
## [1] "Reject the null: For Dragon-type evolved Pidgey, there is a significant difference be
tween post-evolution CPs depending on pre-evolution strong attack type."
```

After removing this outlier, we have found a significant correlation between post-evolution CP and pre-evolution strong attack type for Pidgey with a post-evolution Dragon-type strong attack. We should be skeptical of results since the sample size is quite small, but for our purposes, we continue with this conclusion.

Players of the original Pokemon games may be familiar with the “same attack bonus,” an increase in CP for Pokemon who use an attack of the same type as their species. Since Pidgey is a normal and flying-type Pokemon, we predict that the Pidgey’s CP would increase due to this same-attack bonus when the Pidgey has a Flying strong attack type. From our data, it appears that for an evolved Pidgey with a flying strong attack

type, it doesn't matter what its strong attack type was pre-evolution. For evolved Pidgey with a dragon strong attack type, there is a significant increase in post-evolution CP associated with a Flying pre-evolution strong attack type. This implies that pre-evolution Pidgey of Flying type strong attacks have higher CPs.

In conjunction with our model, we recommend evolving Pidgey with high CP, HP, and stardust requirements, and add that the Pidgey should have a Flying-type strong attack.

With this advice in mind, we would like to know the probability of a Pidgey having an above average CP provided that they have a Flying-type strong attack. Our previous test implied a difference in CPs between Pidgey with Dragon-type and Flying-type strong attacks. If true, we should see a higher probability of a high CP provided that the Pidgey has a flying-type strong attack.

```
# add a row to the table for CP above-average

pidgey <- pidgey |>
  mutate(above_avg = cp >= mean(cp))

# making contingency table

flying_above <- data.frame(pidgey$attack_strong_type, pidgey$above_avg)
flying_above <- table(pidgey$attack_strong_type, pidgey$above_avg)
kable(flying_above, align = 'r',col.names = c('Strong Attack Type', 'Count Below Average', 'C
ount Above Average'), caption = 'Pidgey Strong Attack Types Compared to CP')
```

Pidgey Strong Attack Types Compared to CP

Strong Attack Type	Count Below Average	Count Above Average
Dragon	10	5
Flying	10	14

```
# calculate probabilities

n_pidgey <- count(pidgey) # count Pidgey
n_above_avg <- sum(pidgey$above_avg) # count Pidgey with above-average CP
n_flying <- sum(pidgey$attack_strong_type == "Flying") # count flying Pidgey
n_both <- sum(pidgey$attack_strong_type == "Flying" & pidgey$above_avg)

p_flying <- n_flying / n_pidgey
p_both <- n_both / n_pidgey
p_final <- p_both / p_flying

print(paste("The probability that a Pidgey has an above-average CP provided that it has a fly
ing-type strong attack is", p_final))
```

```
## [1] "The probability that a Pidgey has an above-average CP provided that it has a flying-t
ype strong attack is 0.583333333333333"
```

If there were no correlation between high CP and strong attack type, we would expect this probability to be around 0.5; since the probability is larger, there could be a correlation. But this result may be due to chance, given that the sample size is relatively small with $n = 40$. We will construct a confidence interval using the Central Limit Theorem to quantify our uncertainty with this estimate. First, we check the conditions:

observations are all independent since wild Pidgey do not influence each others' attack type, and success-failure is verified below. Note that for the sample size, we use the n_{flying} variable, since the probability we calculated is only for Pidgey with Flying-type strong attacks.

```
# verifying success-failure condition

n_flying*p_final # n*p
```

```
##    n
## 1 14
```

```
n_flying*(1-p_final) #n*(1-p)
```

```
##    n
## 1 10
```

Both np and $n(1-p)$ are greater than or equal to 10, so the Central Limit Theorem holds. The sample proportion p_{final} will follow a normal distribution with a mean of p_{final} and a standard error of $\sqrt{\frac{p_{final}(1-p_{final})}{n_{flying}}}$. We use this to build a 95% confidence interval around the sample proportion.

```
SE_flying <- sqrt((p_final*(1-p_final))/n_flying) # standard error
conf_int95 <- c(p_final - 1.96*SE_flying, p_final + 1.96*SE_flying) # 1.96 = Z-score for 95% confidence

print(conf_int95)
```

```
## $n
## [1] 0.3860896
##
## $n
## [1] 0.7805771
```

We are 95% confident that the proportion of Pidgey with a Flying-type strong attack who have an above-average CP is between 38.6% and 78.1%. This is a large range due to the small sample size, and since it contains the null value 50% we cannot be certain that Pidgey with a Flying-type strong attack necessarily have a higher CP than those with a Dragon-type strong attack. With a larger sample size, we could find a smaller confidence interval with greater accuracy to determine whether or not there is a correlation between CP and strong attack type in Pidgey.

Though our results imply there is no direct significant benefit to evolved CP, there is still benefit to evolving a Flying-type Pidgey: if the Pidgey evolves to have a Dragon-type strong attack, its CP will be significantly higher than a Pidgey who had a Dragon-type pre-evolution strong attack.

Battling in Pokemon Go isn't only about having a Pokemon with an above-average CP. More important to an individual player might be the prevalence of an extremely high CP. The largest CP in our data set for a Pidgey pre-evolution is 384, so let's consider any CP over 300. A player might assume that CP follows a normal distribution due to the sheer number of Pokemon available to catch. We can use the sample mean and the standard deviation to calculate the probability that a Pidgey with a Flying-type strong attack has a CP above 300, assuming that CP has a normal distribution. We will compare this to actual probabilities calculated from our data and compare to determine whether CP is normally distributed in Pokemon GO.

```
# create data frame of flying-type strong attack pidgey
flying_type <- pidgey |> filter(attack_strong_type == "Flying")

# standard deviation and mean of their CPs
sd_flying <- sd(flying_type$cp)
mean_flying <- mean(flying_type$cp)

# find probability
print(paste("The probability that a Pidgey with a Flying-type strong attack has a CP above 30
0 is",pnorm(-300, mean = mean_flying, sd = sd_flying)))
```

```
## [1] "The probability that a Pidgey with a Flying-type strong attack has a CP above 300 is
1.41161070876722e-06"
```

```
# compare to general Pidgey population
sd <- sd(pidgey$cp)
mean <- mean(pidgey$cp)

print(paste("The probability that any Pidgey will have a CP above 300 is",pnorm(-300, mean =
mean, sd = sd)))
```

```
## [1] "The probability that any Pidgey will have a CP above 300 is 5.63008482191353e-06"
```

Both of these probabilities are quite small, but observe that the actual probability from our dataset of a Pidgey having a CP above 300 is

```
# count how many Pidgey have CP above 300

above_300_flying <- sum(flying_type$cp >= 300)
above_300_all <- sum(pidgey$cp >= 300)

# calculate proportion of Pidgey with CP above 300

above_300_flying / n_flying
```

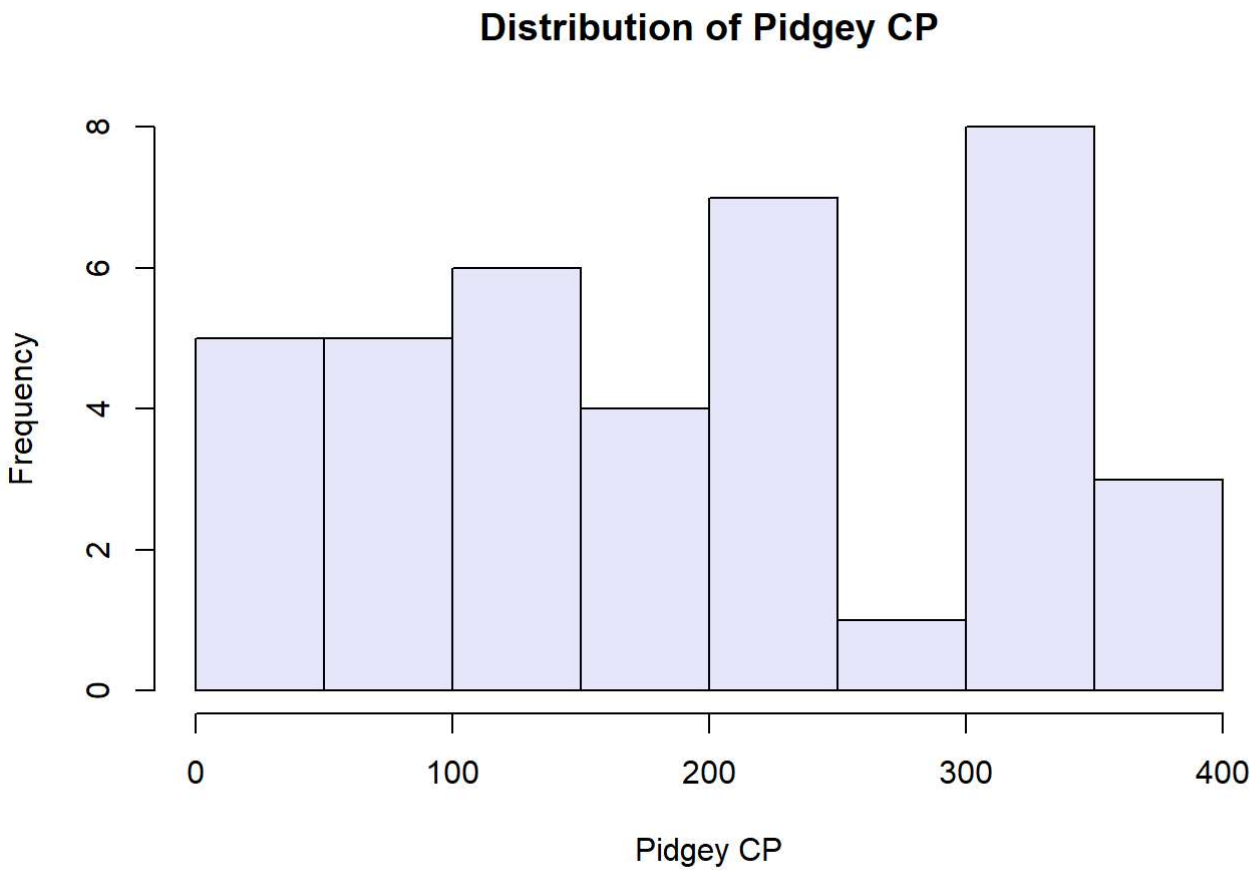
```
## [1] 0.3333333
```

```
above_300_all / count(pidgey)
```

```
##           n
## 1 0.2820513
```

Such a high discrepancy between the sample probability of about 30% compared to the normal distribution probability of less than 1% implies that CP does not take a normal distribution. We can see this graphically:

```
hist(pidgey$cp, main = "Distribution of Pidgey CP", xlab = "Pidgey CP", col = 'lavender')
```



It could be possible that CP is normally distributed (or close enough to it) in-game, since there are more Pokemon available to catch. However, it also seems that some Pokemon tend toward different CP ranges. From our work with this sample, we found that CP is not normally distributed, with openings for further research.

We will next analyze the effect of weak attacks. Since all pre-evolved Pidgey have normal weak attack types, we might ask if the specific attack they perform (Quick Attack vs. Tackle) influences their CP. Since we already established that pre-evolution CP is linearly related to post-evolution CP, we proceed with an independence test between pre-evolution CP and Pidgey Weak Attack. We choose not to test against the post-evolution CP because evolved Pidgey have different weak attacks (Steel Wing and Wing Attack) and weak attack types (Steel vs. Flying), which could influence the results. If there is a correlation between pre-evolution CP and Weak Attack, it will translate linearly to post-evolution CP.

```
# create a new data frame
weak_attacks <- pidgey |>
  group_by(attack_weak) |>
  mutate(above_avg = cp >= mean(cp), below_avg = cp < mean(cp)) |>
  summarize(
    above_avg = sum(above_avg),
    below_avg = sum(below_avg)
   )

# print table
kable(weak_attacks, align = 'r',col.names = c('Weak Type', 'Count Above Average', 'Count Belo
w Average'), caption = 'Pidgey Weak Attack Types Compared to CP')
```

Pidgey Weak Attack Types Compared to CP

Weak Type	Count Above Average	Count Below Average
Quick Attack	10	7
Tackle	9	13

We can proceed with a chi-square test since data are independent due to random sampling and each cell has a count greater than 5.

```
# create df and table for chi-square test
weak_attacks <- data.frame(pidgey$attack_weak, pidgey$above_avg)
weak_attacks <- table(pidgey$attack_weak, pidgey$above_avg)

# chi square test
chisq.test(weak_attacks,correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  weak_attacks
## X-squared = 1.2319, df = 1, p-value = 0.267
```

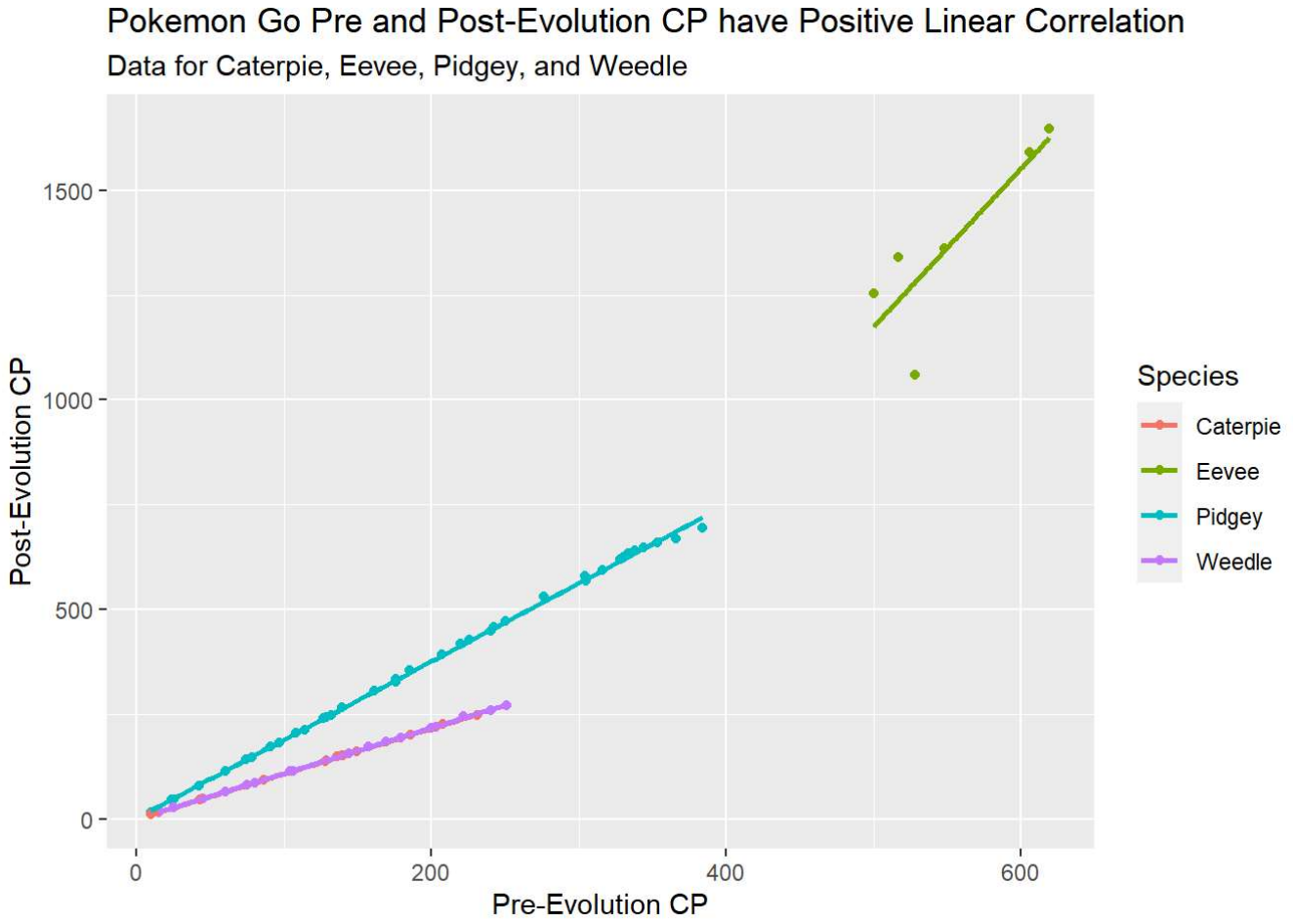
Since the p-value is 0.267 > 0.05, we conclude that Weak Attack and having an above or below average CP are independent. There is no correlation between Weak Attack and pre-evolution CP. Returning to the idea of the “same-attack bonus,” this result makes sense since both Quick Attack and Tackle are normal-type attacks, meaning they provide equal advantage to a Pidgey’s CP. In conclusion, we have found that Pidgey with higher CP, HP, and stardust power-up stats who have a Flying Type Strong Attack are most likely to have the highest post-evolution CP.

Generalization to other Pokemon Species

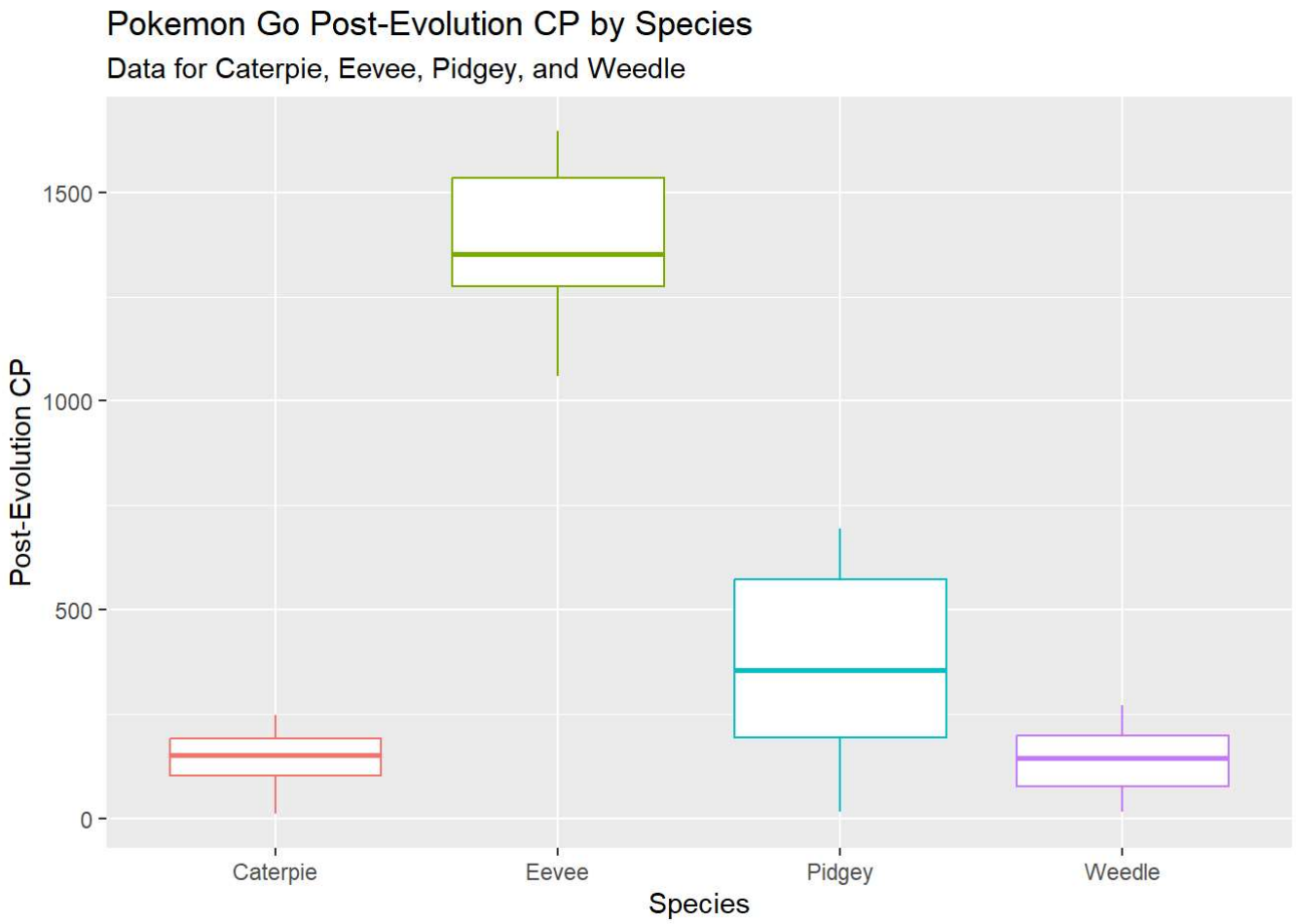
We can make a plot to determine whether we should consider the relationship between pre- and post-evolution CP for the other Pokemon series in the pokemon_go data set:

```
ggplot(pokemon_go, aes(x = cp, y = cp_new, color = species)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE) +
   labs(
    title = "Pokemon Go Pre and Post-Evolution CP have Positive Linear Correlation",
    subtitle = "Data for Caterpie, Eevee, Pidgey, and Weedle",
    x = "Pre-Evolution CP",
    y = "Post-Evolution CP",
    color = "Species"
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(pokemon_go, aes(x = species, y = cp_new, color = species)) +
  geom_boxplot(show.legend = FALSE) +
  labs(
    title = "Pokemon Go Post-Evolution CP by Species",
    subtitle = "Data for Caterpie, Eevee, Pidgey, and Weedle",
    x = "Species",
    y = "Post-Evolution CP"
  )
```



We see that the Weedle and Caterpie have nearly identical median post-evolution CPs with slight differences in IQR. As the whiskers completely overlap, we can say that CP levels of these two Pokemon are very similar. Pidgey has a higher median CP with a larger IQR, though its whiskers overlap with Caterpie and Weedle slightly. Notably Eevee has a significantly higher median CP and IQR. We also see that Eevee pre-and post-evolution CP has more outliers in the linear model compared to other species, with generally much higher CPs. We should keep this in mind as we continue with the multiple regression model.

To conclude our analysis of the pokemon_go data set, we will build linear models for the three remaining Pokemon species and compare it to the model we have built for Pidgey to determine if pre-evolution CP, HP, and power-up stardust are reliable predictor variables for post-evolution CP.

```
# create tables for each species
eevee <- pokemon_go |> filter(species == "Eevee")
weedle <- pokemon_go |> filter(species == "Weedle")
caterpie <- pokemon_go |> filter(species == "Caterpie")

# linear models similar to best-fitting Pidgey models
lm_eevee <- lm(eevee$cp_new ~ eevee$cp + eevee$hp + eevee$power_up_stardust)
summary(lm_eevee)
```

```
##
## Call:
## lm(formula = eevee$cp_new ~ eevee$cp + eevee$hp + eevee$power_up_stardust)
##
## Residuals:
##       1       2       3       4       5       6
##  -32.49  105.44   66.53   30.35 -159.60  -10.24
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1036.8405  1063.8007  -0.975    0.433
## eevee$cp                    1.6916     2.4258   0.697    0.558
## eevee$hp                   11.8384    24.3731   0.486    0.675
## eevee$power_up_stardust     0.2641     0.3924   0.673    0.570
##
## Residual standard error: 146.8 on 2 degrees of freedom
## Multiple R-squared:  0.8176, Adjusted R-squared:  0.5439
## F-statistic: 2.988 on 3 and 2 DF,  p-value: 0.2608
```

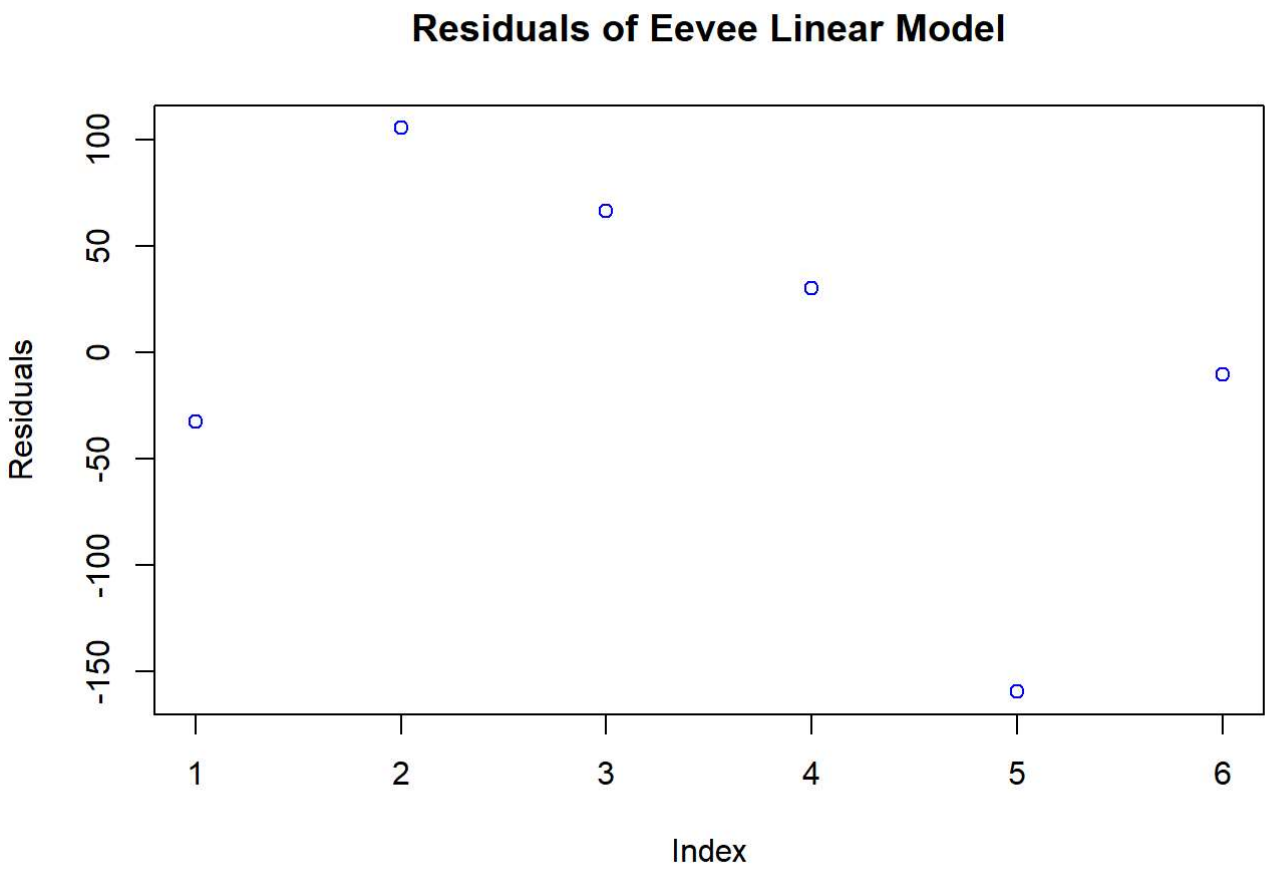
```
lm_caterpie <- lm(caterpie$cp_new ~ caterpie$cp + caterpie$hp + caterpie$power_up_stardust)
summary(lm_caterpie)
```

```
##
## Call:
## lm(formula = caterpie$cp_new ~ caterpie$cp + caterpie$hp + caterpie$power_up_stardust)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.85753 -0.36616 -0.29674  0.08161  3.07621
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -0.550216   1.327854  -0.414    0.693
## caterpie$cp                 1.087593   0.028091  38.717 1.98e-08 ***
## caterpie$hp                 0.020868   0.073944   0.282    0.787
## caterpie$power_up_stardust -0.000848   0.002188  -0.388    0.712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.674 on 6 degrees of freedom
## Multiple R-squared:  0.9997, Adjusted R-squared:  0.9995
## F-statistic:  6105 on 3 and 6 DF,  p-value: 7.681e-11
```

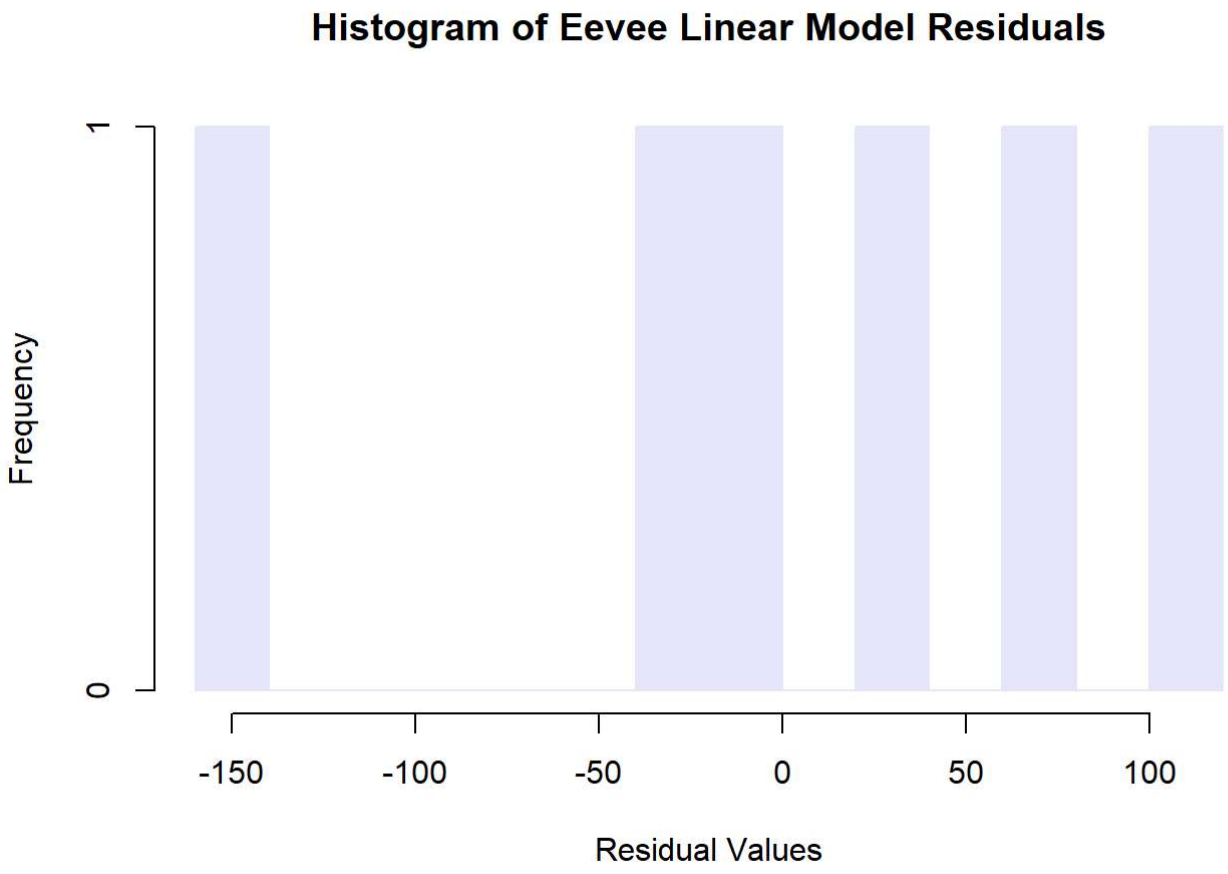
```
lm_weedle <- lm(weedle$cp_new ~ weedle$cp + weedle$hp + weedle$power_up_stardust)
summary(lm_weedle)
```

```
##
## Call:
## lm(formula = weedle$cp_new ~ weedle$cp + weedle$hp + weedle$power_up_stardust)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5492 -0.6453 -0.0440  0.3781  3.2796
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.8376094  1.3453412   0.623    0.542
## weedle$cp                  1.0980951  0.0156298  70.256   <2e-16 ***
## weedle$hp                 -0.0469369  0.0716218  -0.655    0.522
## weedle$power_up_stardust -0.0002565  0.0011175  -0.230    0.821
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.264 on 16 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
## F-statistic: 2.624e+04 on 3 and 16 DF,  p-value: < 2.2e-16
```

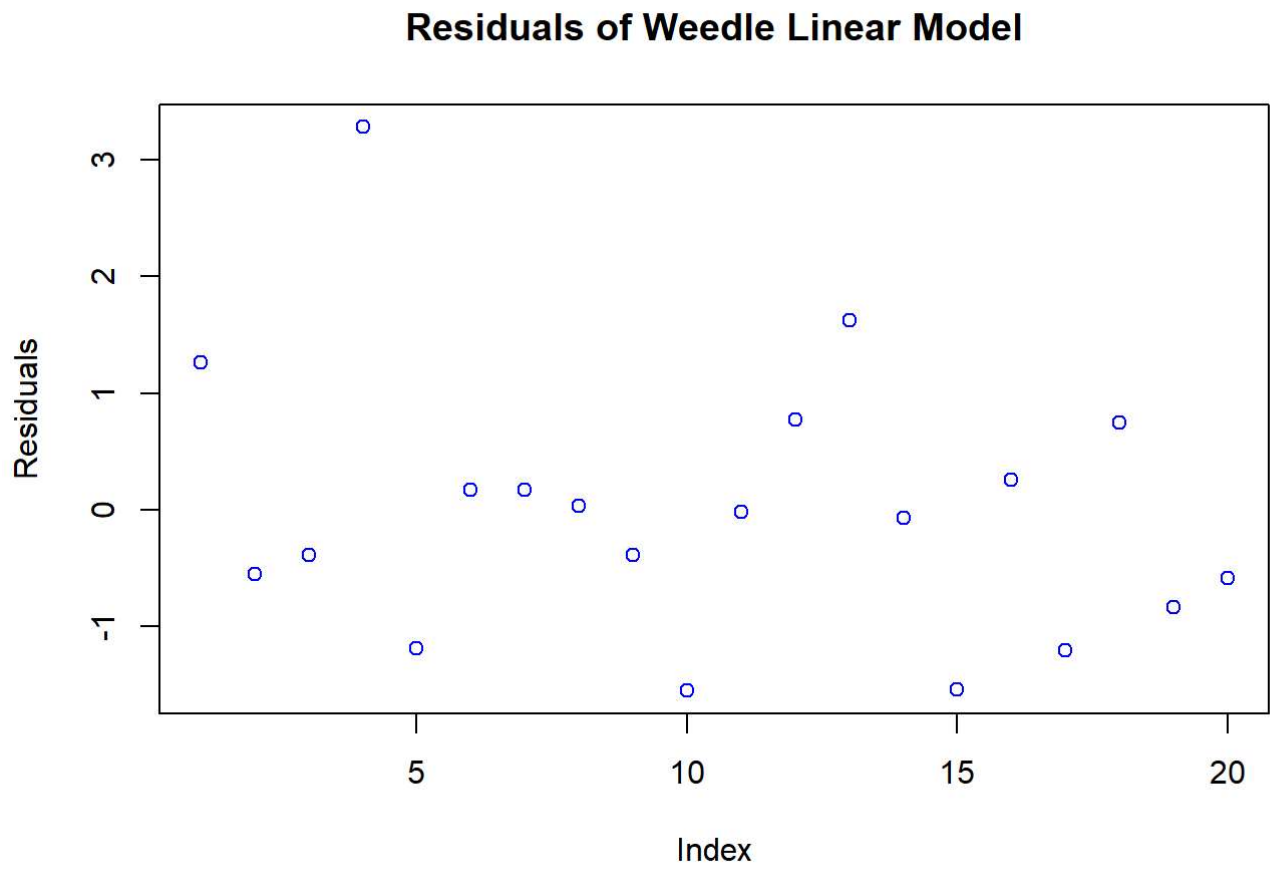
```
# checking conditions
# eevee
plot(lm_eevee$residuals, ylab = "Residuals", main = "Residuals of Eevee Linear Model", col =
'blue')
```



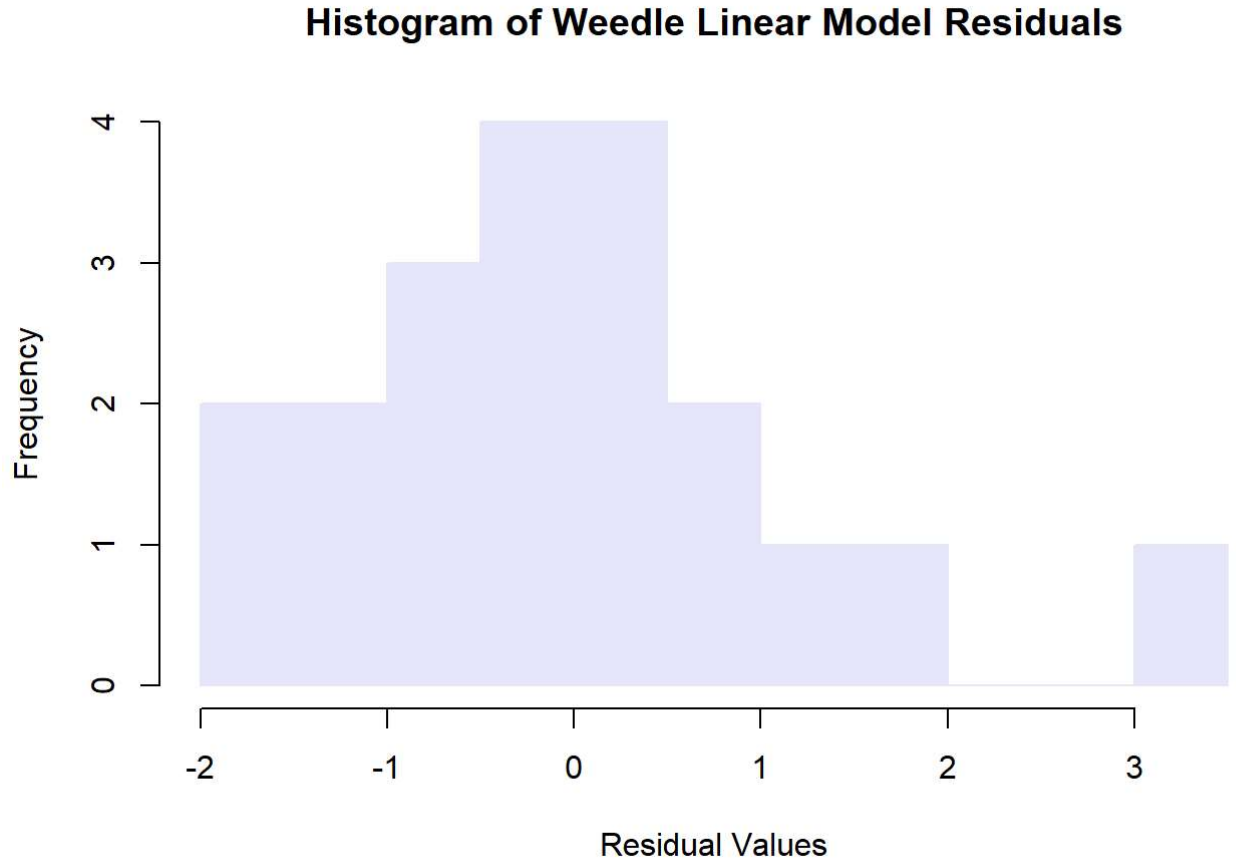
```
hist(lm_eevee$residuals, breaks = 10, main = "Histogram of Eevee Linear Model Residuals", xla
b = "Residual Values", ylab = "Frequency", col = 'lavender', border = 'lavender')
```



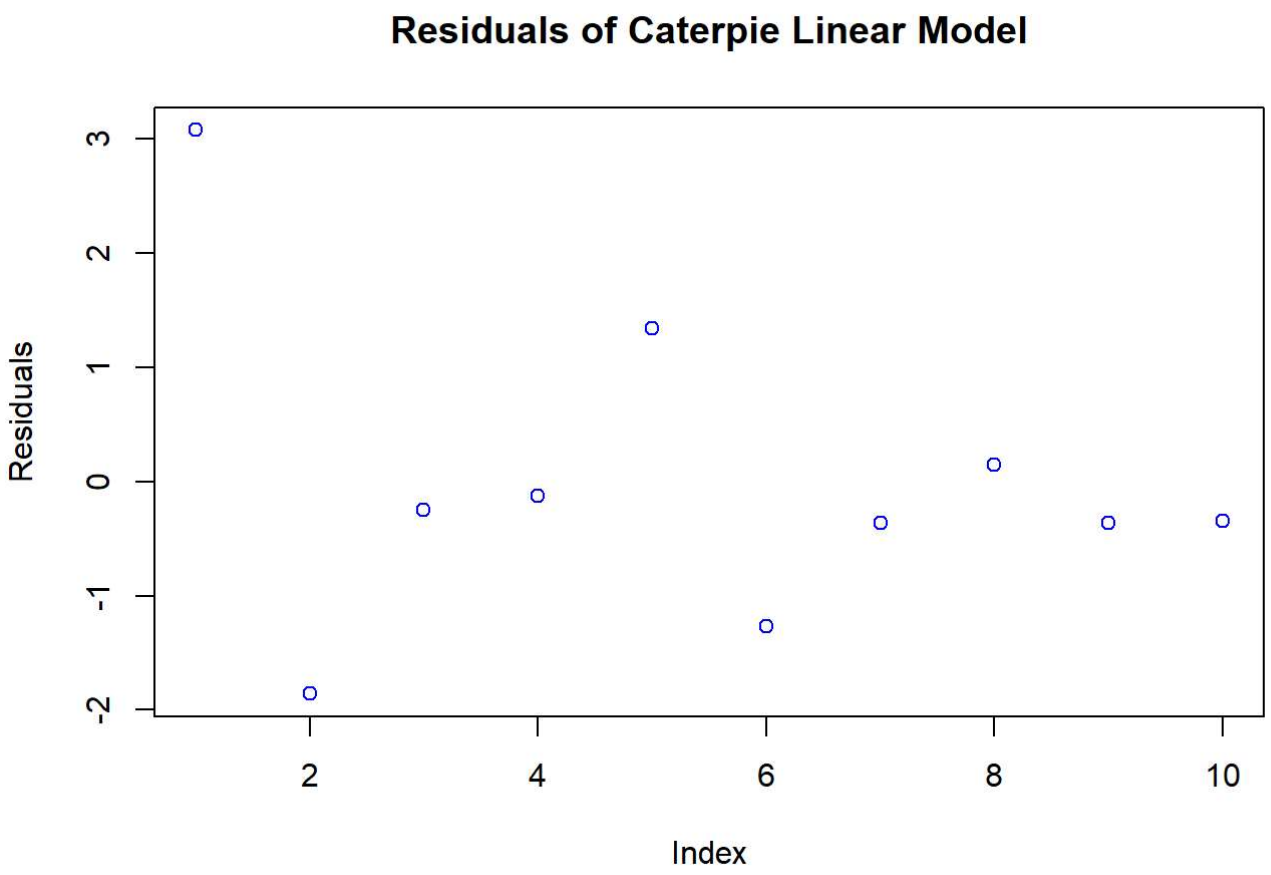

```
# weedle
plot(lm_weedle$residuals, ylab = "Residuals", main = "Residuals of Weedle Linear Model", col
= 'blue')
```



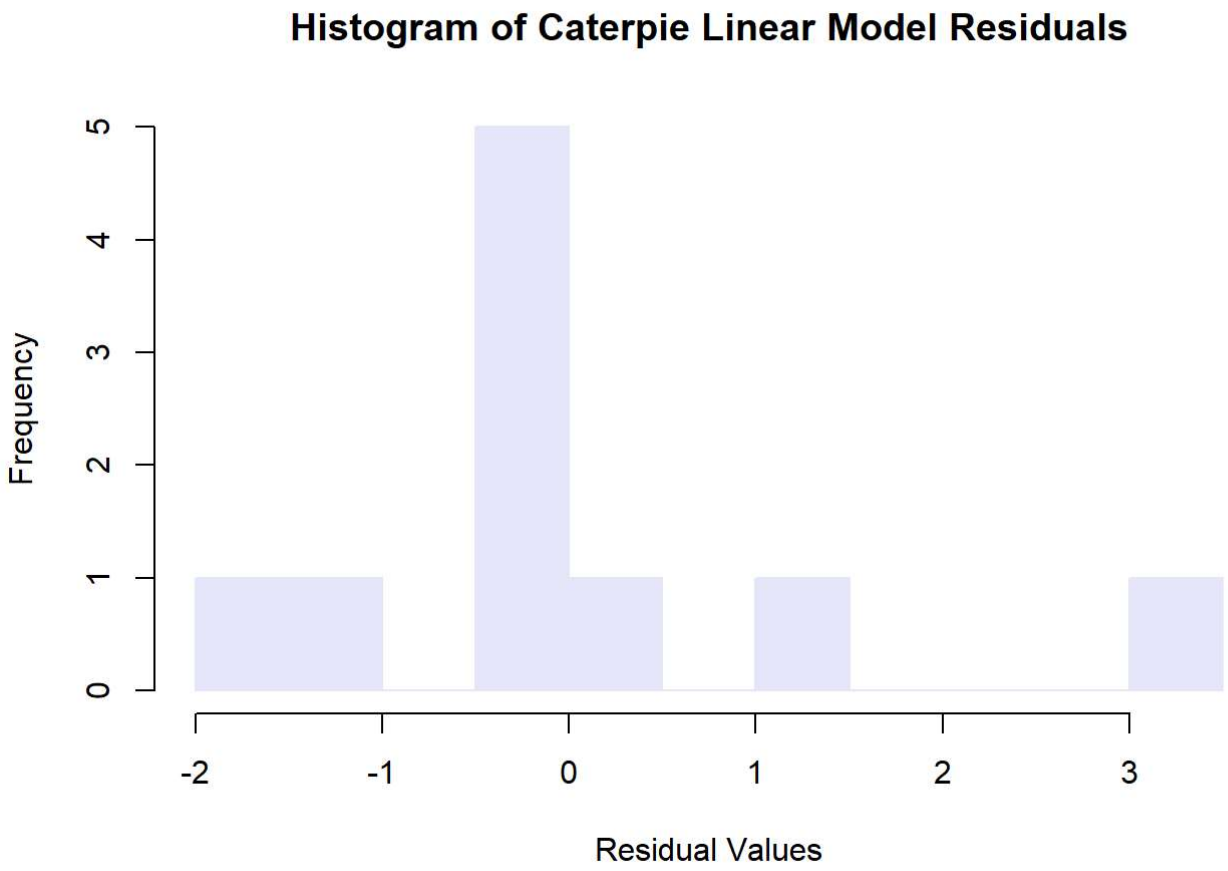

```
hist(lm_weedle$residuals, breaks = 10, main = "Histogram of Weedle Linear Model Residuals", x
lab = "Residual Values", ylab = "Frequency", col = 'lavender', border = 'lavender')
```



```
# caterpie
plot(lm_caterpie$residuals, ylab = "Residuals", main = "Residuals of Caterpie Linear Model",
col = 'blue')
```




```
hist(lm_caterpie$residuals, breaks = 10, main = "Histogram of Caterpie Linear Model Residual
s", xlab = "Residual Values", ylab = "Frequency", col = 'lavender', border = 'lavender')
```



From these linear models we can observe the following:

1. For Caterpie and Weedle, only the pre-evolution CP has a significant impact on post-evolution CP, according to the p-value analysis. Variables like HP and stardust have p-values larger than 0.05 and therefore are not significant.
2. Caterpie and Weedle have large adjusted R-square values, implying that 99.95% and 99.98% of variation in post-evolution CP can be explained by variations in pre-evolution CP, HP, and power-up stardust. This implies that the regression model is a good fit.
3. Eevee, having a general trend of a higher CP and higher variation than Pidgey, Caterpie, or Weedle, has a model that does not fit well, with an adjusted R-squared of only 0.54. The p-value analysis shows that CP, HP, and power-up stardust are not significant predictors of post-evolution CP. We should keep in mind that this may also be due to the extremely small sample size, which can impact the fit of the model.

These results make sense since the residual plots show that a linear model is not the best fit: the small sample sizes for Eevee and Caterpie mean that residuals are not nearly normal. The only other Pokemon for which multiple regression is a good candidate might be Weedle, since the conditions are satisfied.

Overall, we cannot draw any definitive conclusions that generalize to all Pokemon regarding a model for predicting post-evolution CP. We postulate that pre-evolution CP is the most significant predictor, with possibilities of HP, CP, and stardust power-up requirements impacting it as well. We consider the effects of the “same attack bonus” for weak and strong attacks of the same type as the Pokemon, and leave detailed analysis to further research.