Exploring_Pokemon_Stats - Final Project

Pokemon have stats associated with their "permanent stats", those being"

Hit points (HP), Attack, Defense, Special Attack, Special Defense, Speed

For each permanent stat, each pokemon has a base stat that raises with leveling through battle, items, or stat-raising moves.

The "Total" column is: The sum of the base of each permanent stat and a general guide to how strong a pokemon is.

"Stats" = base stat for the referred pokemon or group of pokemon

```
#Load all libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------------- tidyverse 
```

```
## v ggplot2 3.3.3     v purrr   0.3.3
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------ tidyverse_confli
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(rstatix)
```

```
##
## Attaching package: 'rstatix'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(ggpubr)
library(ggplot2)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

Read in dataframe:

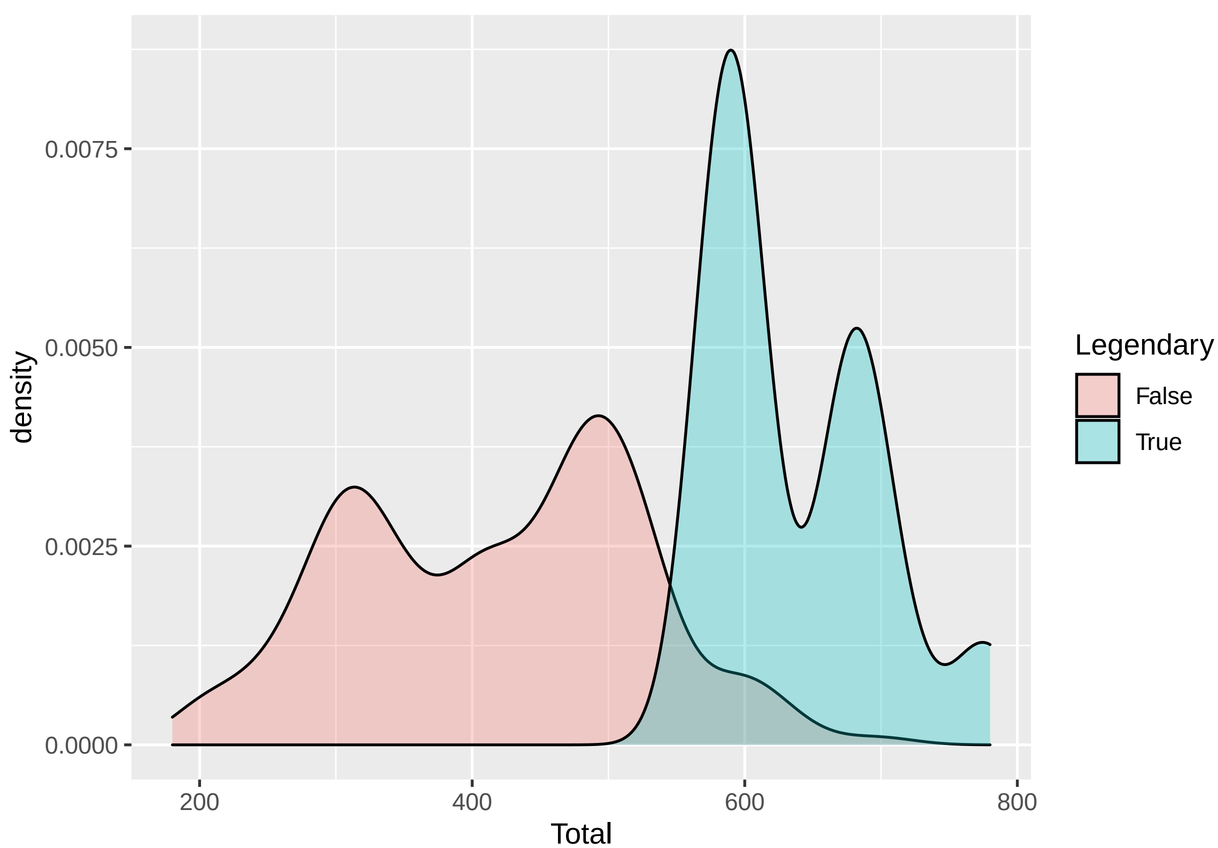
```
pkmn=read.csv("Pokemon.csv")
```

Overall question: Does the mean Total change significantly per gen?

First, a note. I hypothesize that legendary pokemon have significantly higher mean totals compared to non-legendary pokemon. I will test that.

Let's look at the density plot first:

```
ggplot(pkmn, aes(x=Total, fill=Legendary)) +
  geom_density(alpha=0.3)
```



There appear to be a higher density of legendary pokemon with high total stats. Let's test our hypothesis:

HO: There is no difference in Pokemon 'Total' between legendary and non-legendary pokemon. alpha = 0.05

First, let's check assumptions for a t-test:

Random Sampling

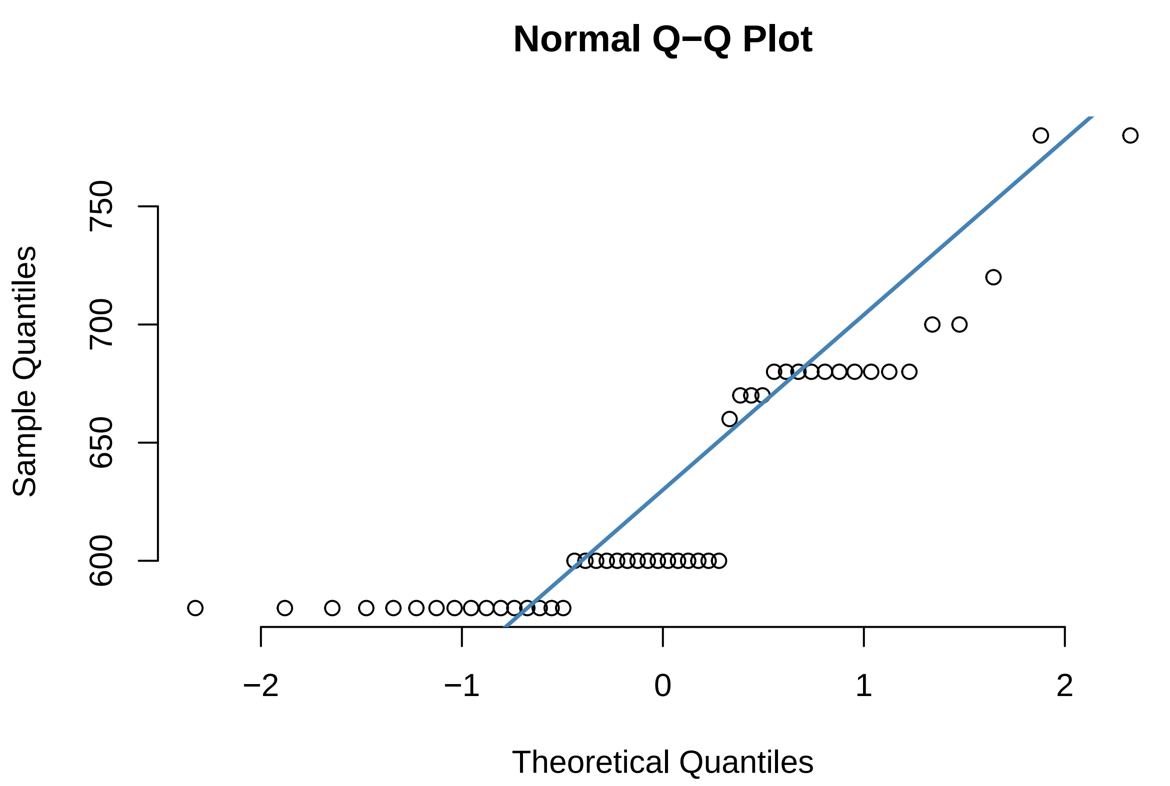
I'll take a random sample of 50 legendary and non-legendary pokemon

```
pkmn=read.csv("Pokemon.csv")
set.seed(123)
Legendary=pkmn %>%
  filter(Legendary=="True") %>%
  sample_n(50)
NonLegendary=pkmn %>%
  filter(Legendary=="False") %>%
  sample_n(50)
```

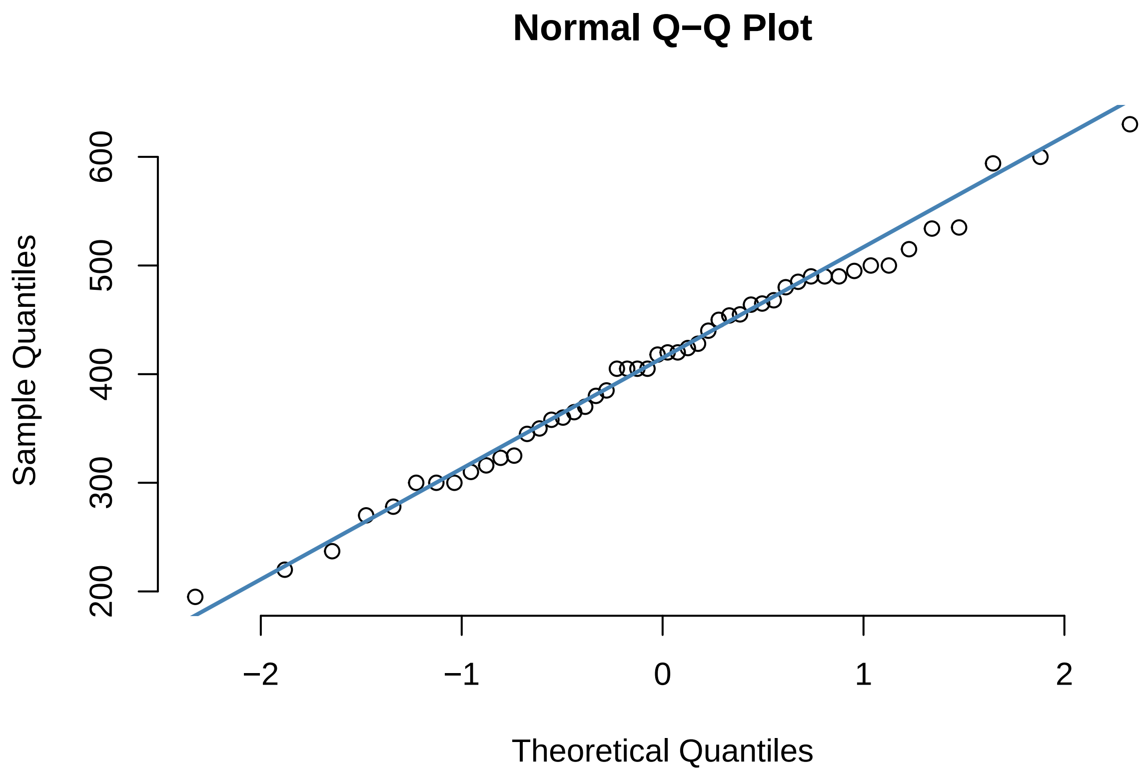
Normality

I'll make a QQ plot of the Total Score for each group

```
qqnorm(Legendary$Total, pch = 1, frame = FALSE)
qqline(Legendary$Total, col = "steelblue", lwd = 2)
```




```
qqnorm(NonLegendary$Total, pch = 1, frame = FALSE)
qqline(NonLegendary$Total, col = "steelblue", lwd = 2)
```



What is the mean/median Total for legendary vs nonlegendary pokemon? What does the boxlot look like?

```r
# summary stats for sample of 50

Legendary %>% filter(Legendary=="True") %>% get_summary_stats(Total, type="full")
```

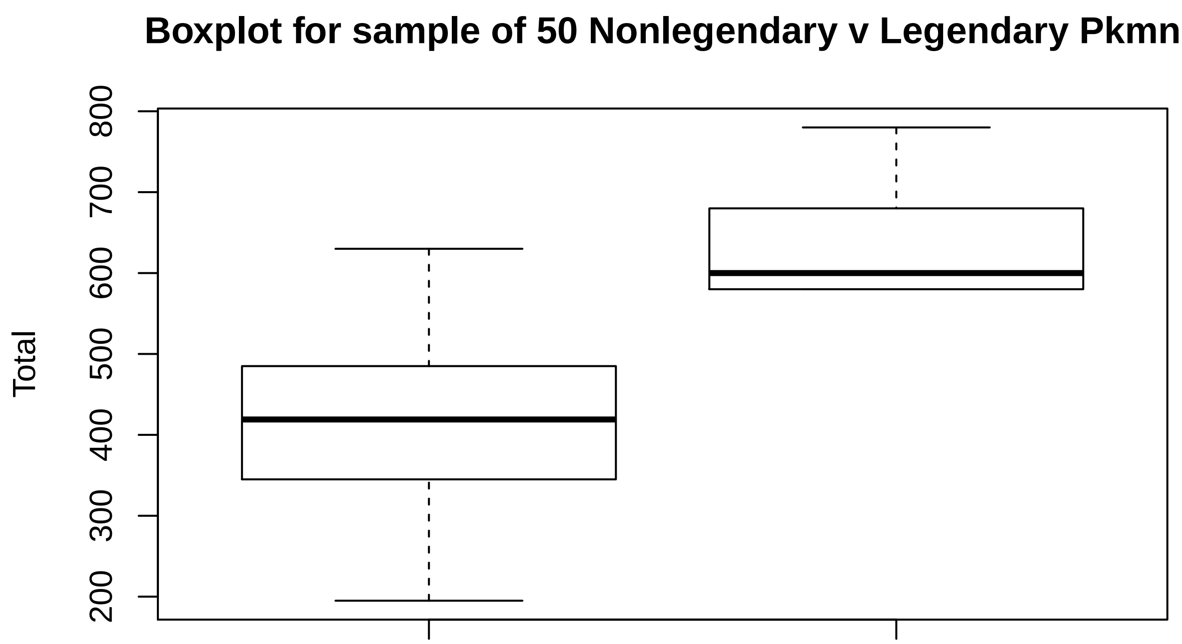
```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total       50   580   780    600   580   680   100  29.7  629.  54.8  7.75
## # ... with 1 more variable: ci <dbl>
```

```r
NonLegendary %>% filter(Legendary=="False") %>% get_summary_stats(Total, type="full")
```

```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total       50   195   630    419  346.  484.  138.  104.  411.  97.7  13.8
## # ... with 1 more variable: ci <dbl>
```

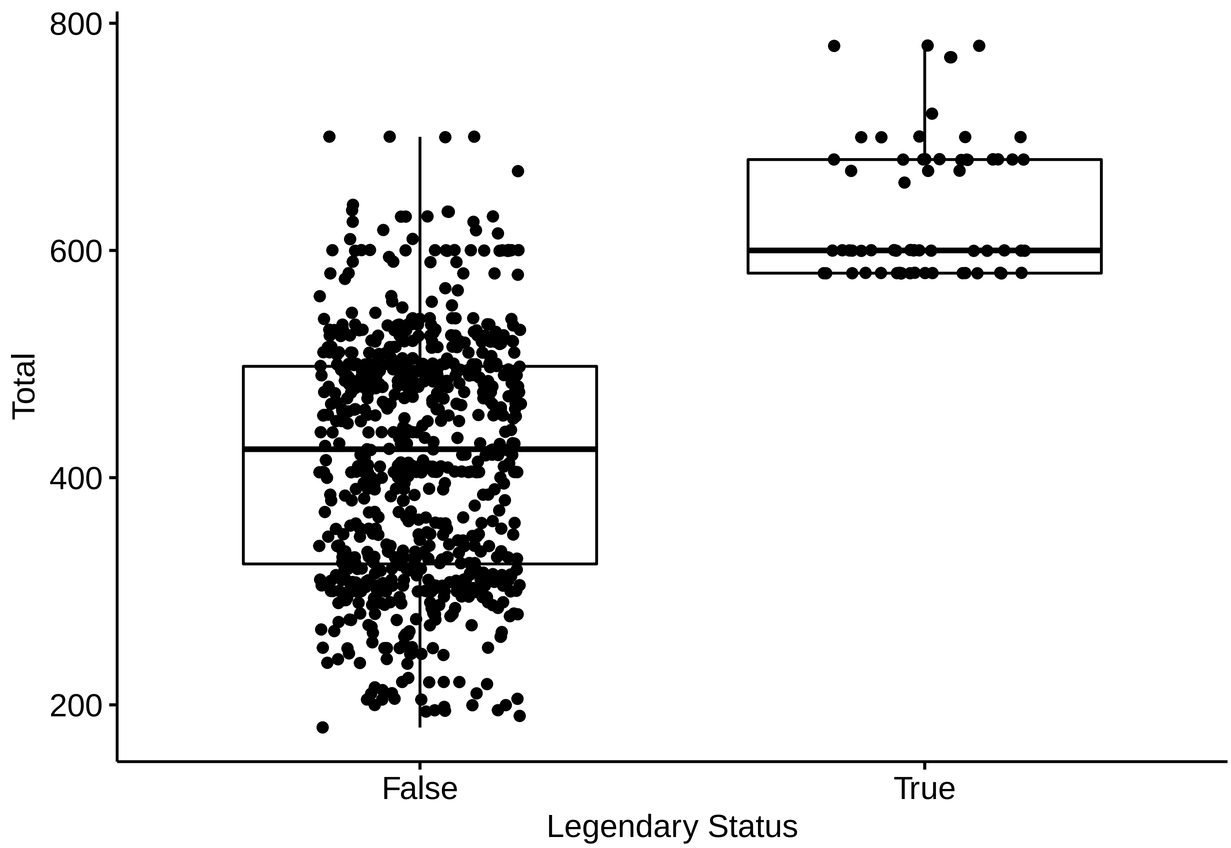
```r
#boxplot for sample of 50

boxplot(NonLegendary$Total,Legendary$Total,main="Boxplot for sample of 50 Nonlegendary v Legendary Pkmn
```




```r
#boxplot for total df

ggboxplot(
  pkmn, x = "Legendary", y = "Total",
  ylab = "Total", xlab = "Legendary Status", add = "jitter"
  )
```



The QQ plots do not show normality, there's a backwards S shape for the nonlegendary and legendary, therefore, I'll use a Wilcox test.

The mean Total for Legendary as well as the median Total for Legendary is higher than the mean and median Total for Nonlegendary. The boxplot shows the same.

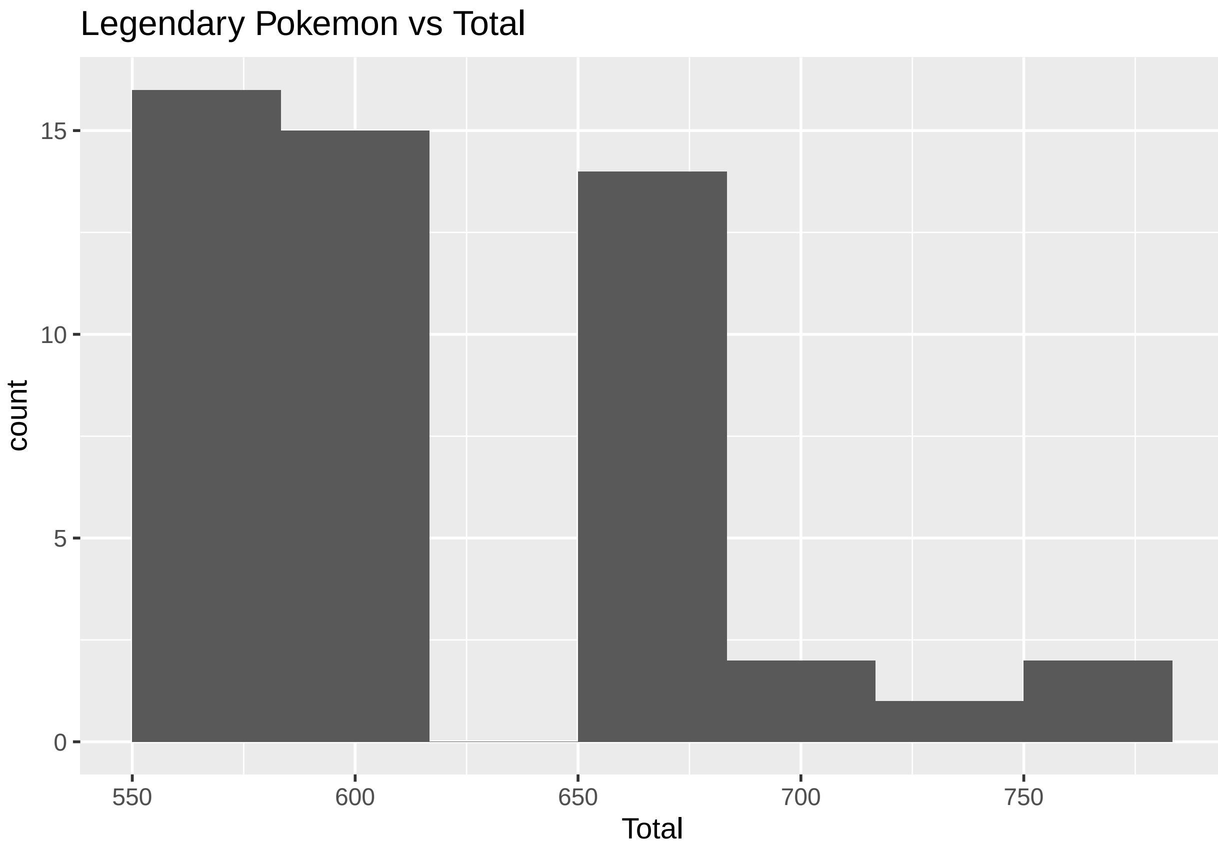
Independent Samples

The samples are independent from each other.

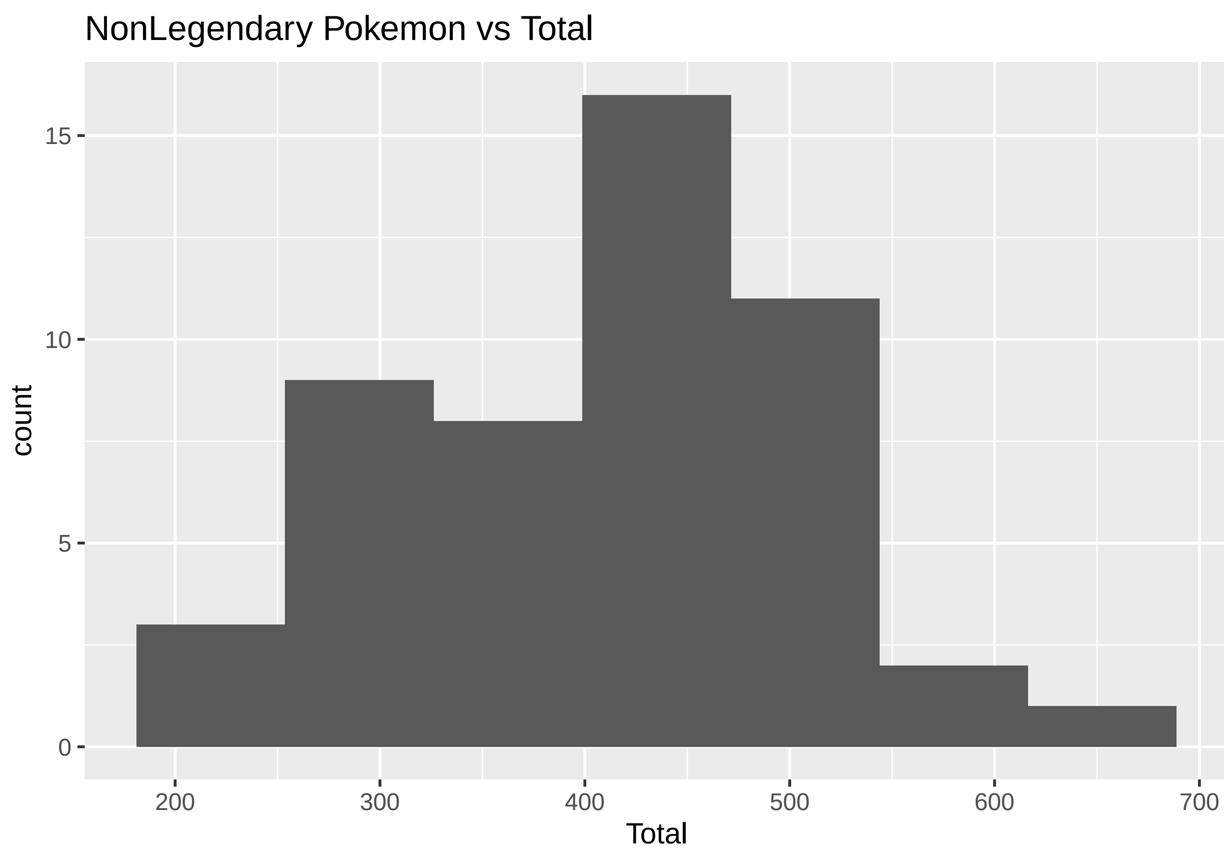
Equal variance

I'll use Bartlett's test for the entire dataframe and also make histograms to test equal variance

```
ggplot(Legendary, aes(x=Total))+geom_histogram(bins = 7)+ggtitle("Legendary Pokemon vs Total")
```




```
ggplot(NonLegendary, aes(x=Total))+geom_histogram(bins = 7)+ggtitle("NonLegendary Pokemon vs Total")
```




```
bartlett.test(Total~Legendary, pkmn) #Alpha = 0.05
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Total by Legendary
## Bartlett's K-squared = 27.272, df = 1, p-value = 1.767e-07
```

Bartlett's test shows a p-value (1.767e-07) less than alpha (set to 0.05). This, along with the histograms, indicates that there is an unequal variance between the legendary and Non-legendary pokemon.

Since the assumptions are not met, and there is unequal variance, I will perform a Wilcox test on the sample.

```
wilcox.test(Legendary$Total, NonLegendary$Total)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Legendary$Total and NonLegendary$Total
## W = 2429.5, p-value = 3.19e-16
## alternative hypothesis: true location shift is not equal to 0
```

According to the Wilcoxon rank sum test, the mean Totals between Legendary and NonLegendary appear to be significantly different with a p-value <3.19e-16.

According to this, we conclude legendary pokemon are on average more powerful than nonlegendary

Finally, What is the mean total of legendary vs nonlegendary pokemon Across the whole dataset?

```
pkmn %>% filter(Legendary=="False") %>% get_summary_stats(Total, type="full")
```

```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total      735   180   700    425   324   498   174  126.  417.  107.  3.94
## # ... with 1 more variable: ci <dbl>
```

```
pkmn %>% filter(Legendary=="True") %>% get_summary_stats(Total, type="full")
```

```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total       65   580   780    600   580   680   100  29.7  637.  60.9  7.56
## # ... with 1 more variable: ci <dbl>
```

Legendary pokemon have a mean total of 637 vs 417 for Non-legendary pokemon.

Despite this, I'll continue on to answer the Overall question:

Does the mean total change significantly per gen?

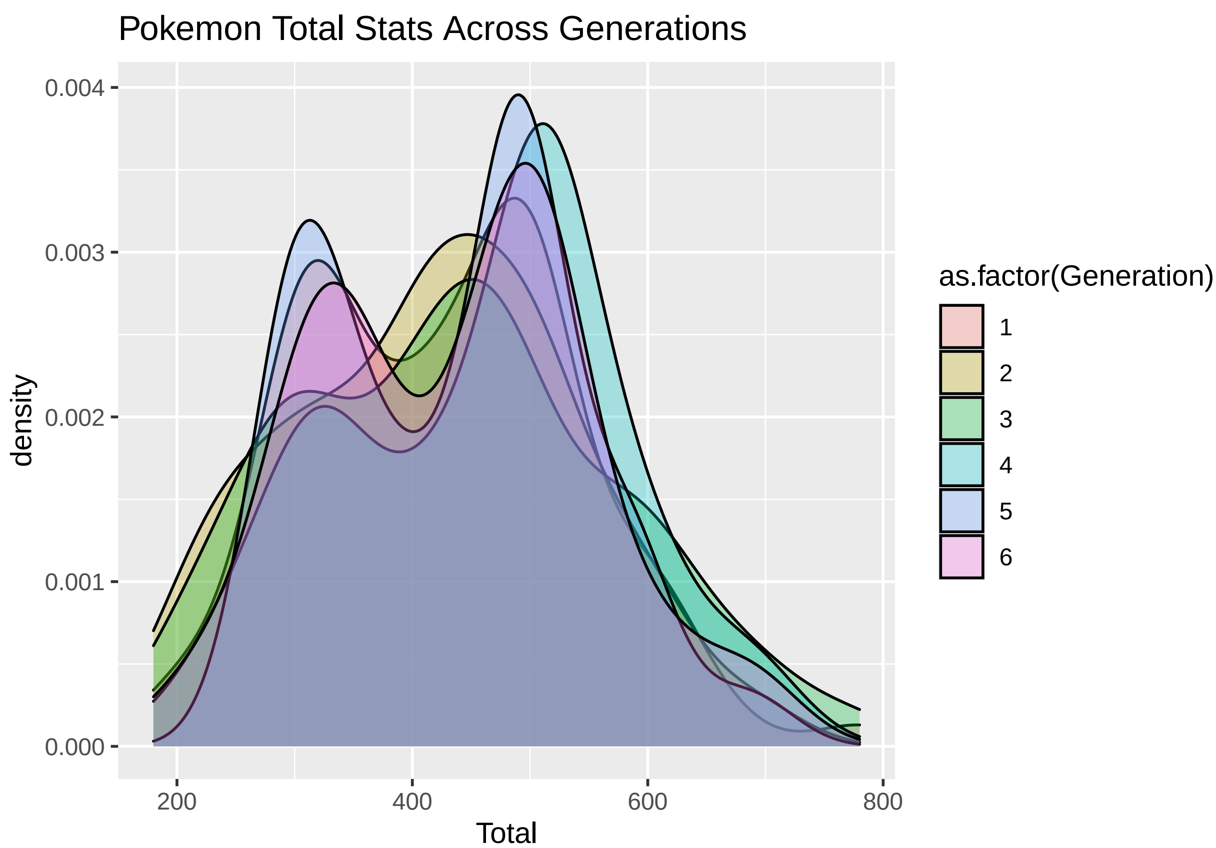
HO: There is no difference in Pokemon 'Total' across generations. alpha = 0.05 To answer this, we'll use an ANOVA to test the significance of the mean totals across gens.

Examining the relationship between Total and Gens

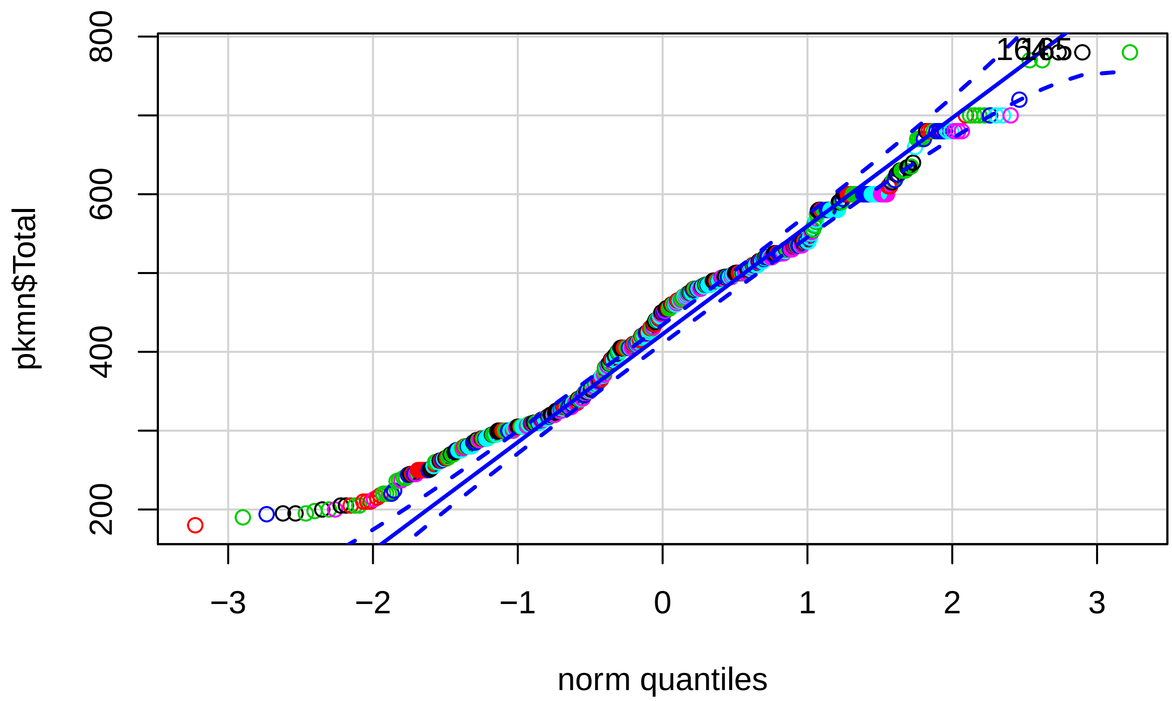
Let's make a histogram and a QQ plot

```
#save generation as a factor in new pkmn DF

pkmnGen <- pkmn %>%
   ggplot(aes(x=Total, fill=as.factor(Generation))) +
  geom_density(alpha=0.3) +
  ggtitle("Pokemon Total Stats Across Generations")
pkmnGen
```




```
qqPlot(pkmn$Total, col = pkmn$Generation)
```




```
## [1] 164 165
```

```
# summary stats for Totals across Gens

Gen1 = pkmn %>% filter(Generation==1) %>% get_summary_stats(Total, type="full")
Gen2 = pkmn %>% filter(Generation==2) %>% get_summary_stats(Total, type="full")
Gen3 = pkmn %>% filter(Generation==3) %>% get_summary_stats(Total, type="full")
Gen4 = pkmn %>% filter(Generation==4) %>% get_summary_stats(Total, type="full")
Gen5 = pkmn %>% filter(Generation==5) %>% get_summary_stats(Total, type="full")
Gen6 = pkmn %>% filter(Generation==6) %>% get_summary_stats(Total, type="full")

Gen1
```

```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total      166   195   780   436.   325   500   175  131.  427.  116.  8.99
## # ... with 1 more variable: ci <dbl>
```

```
Gen2
```

```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total      106   180   700   422.   330   500   170  133.  418.  120.  11.7
## # ... with 1 more variable: ci <dbl>
```

```
Gen3
```

```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total      160   190   780    435   310   530   220  152.  436.  136.  10.8
## # ... with 1 more variable: ci <dbl>
```

```
Gen4
```

```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total      121   194   720    485   350   530   180  119.  459.  120.  10.9
## # ... with 1 more variable: ci <dbl>
```

```
Gen5
```

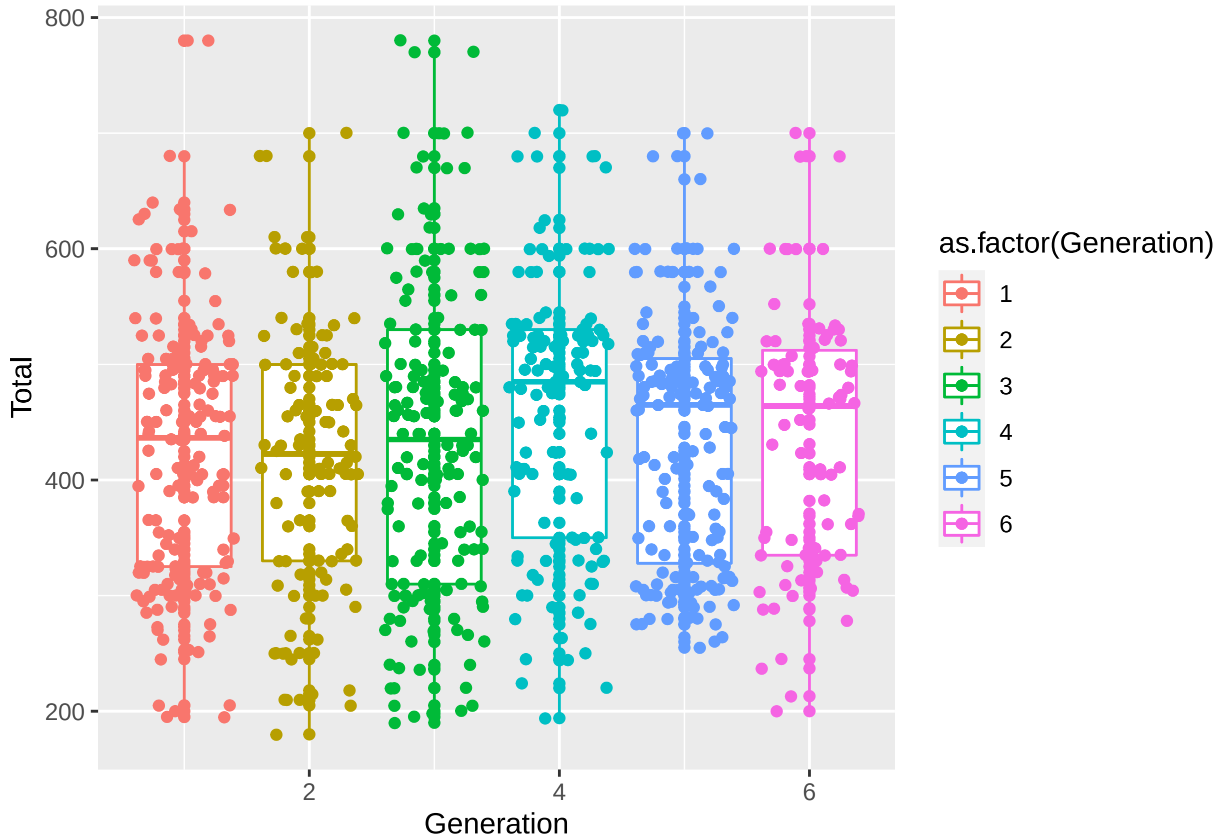
```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total      165   255   700    465   328   505   177  126.  435.  108.  8.42
## # ... with 1 more variable: ci <dbl>
```

```
Gen6
```

```
## # A tibble: 1 x 13
##   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Total       82   200   700    464   335  512.  177.  126.  436.  115.  12.7
## # ... with 1 more variable: ci <dbl>
```

Boxplots to examine Variance

```
#Take a look at variance
pkmn %>% ggplot(aes(Generation, Total, color = as.factor(Generation))) +
  geom_boxplot() + geom_point() + geom_jitter()
```



Variance appears to be relatively equal among all generations according to the boxplot and mean/median shown in summary statistics

Upon first glance at the density plots, there doesn't appear to be differences between the gens. The density appear to be bimodial and right skewed. The QQ plot doesn't appear to show normality, and based on the boxplots, variance does not appear to greatly skew between generations.

Kruskal-Wallis' ANOVA

As I didn't see normality, I'll use a Kruskal-Wallis' ANOVA to compare the Totals across Generations

```
pkmnGen=kruskal.test(Total~Generation,data=pkmn)
pkmnGen
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Total by Generation
## Kruskal-Wallis chi-squared = 9.2316, df = 5, p-value = 0.1002
```

The p value is above 0.05, at 0.1002, and therefore I accept the null hypothesis that there are no significant differences in mean Total across generations.

Concluding that mean Total does not change significantly across generations reaffirms that power scaling across games is consistent

##Are certain types of pokemon more represented as legendary?

H0: Pokemon Type and Legendary status are independent. alpha = 0.05

```
#Create table of legendary pokemon and their type
Leg_Type <-xtabs(~Legendary + Type.1, data = pkmn)
Leg_Type
```

```
##          Type.1
## Legendary Bug Dark Dragon Electric Fairy Fighting Fire Flying Ghost Grass
##     False  69   29     20       40    16       27   47      2    30    67
##     True    0    2     12        4     1        0    5      2     2     3
##          Type.1
## Legendary Ground Ice Normal Poison Psychic Rock Steel Water
##     False     28  22     96     28      43   40    23   108
##     True       4   2      2      0      14    4     4     4
```

```
prop.table(Leg_Type, margin = 2) #View proportions
```

```
##          Type.1
## Legendary        Bug       Dark     Dragon   Electric      Fairy   Fighting
##     False 1.00000000 0.93548387 0.62500000 0.90909091 0.94117647 1.00000000
##     True  0.00000000 0.06451613 0.37500000 0.09090909 0.05882353 0.00000000
##          Type.1
## Legendary       Fire     Flying      Ghost      Grass     Ground        Ice
##     False 0.90384615 0.50000000 0.93750000 0.95714286 0.87500000 0.91666667
##     True  0.09615385 0.50000000 0.06250000 0.04285714 0.12500000 0.08333333
##          Type.1
## Legendary     Normal     Poison    Psychic       Rock      Steel      Water
##     False 0.97959184 1.00000000 0.75438596 0.90909091 0.85185185 0.96428571
##     True  0.02040816 0.00000000 0.24561404 0.09090909 0.14814815 0.03571429
```

Perform Chi-Square test to test H0 Assumptions: Randomly sampled? Yes : Expected values >5? Yes

```
chisq.test(Leg_Type)$expected #About 9% in each type
```

```
## Warning in chisq.test(Leg_Type): Chi-squared approximation may be incorrect
```

```
##          Type.1
## Legendary      Bug     Dark Dragon Electric    Fairy Fighting   Fire Flying
##     False 63.39375 28.48125   29.4   40.425 15.61875 24.80625 47.775  3.675
##     True   5.60625  2.51875    2.6    3.575  1.38125  2.19375  4.225  0.325
##          Type.1
## Legendary Ghost   Grass Ground   Ice  Normal Poison  Psychic   Rock    Steel
##     False  29.4 64.3125   29.4 22.05 90.0375 25.725 52.36875 40.425 24.80625
##     True    2.6  5.6875    2.6  1.95  7.9625  2.275  4.63125  3.575  2.19375
##          Type.1
## Legendary Water
##     False 102.9
##     True    9.1
```

```r
chisq.test(Leg_Type)
```

```
## Warning in chisq.test(Leg_Type): Chi-squared approximation may be incorrect
```

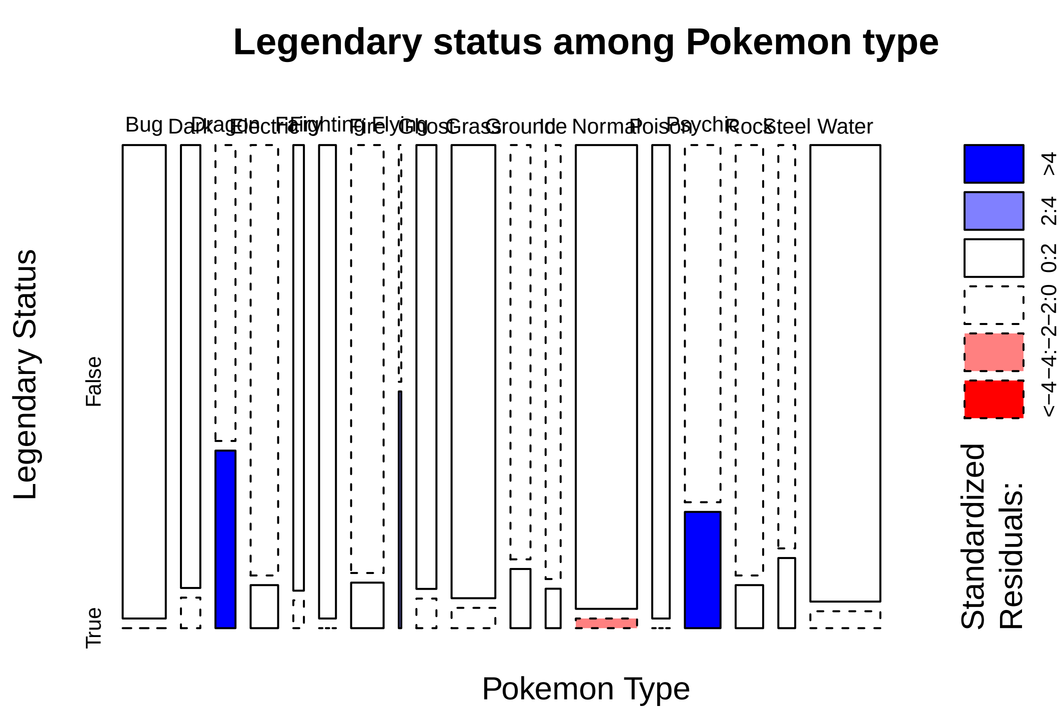
```
##
##  Pearson's Chi-squared test
##
## data:  Leg_Type
## X-squared = 90.42, df = 17, p-value = 5.119e-12
```

Expected: Roughly 9% legendary for each pokemon type.

p-value = 5.119e-12, therefore, we can reject null hypothesis and that legendary type pokemon are independent of type of pokemon

Strikingly, when looking back at proportions table, only 2% of normal pokemon are legendary, and there are absolutely no bug, fighting or poison legendary pokemon. In contrast, 24.6% of all psychic pokemon are legendary, 37.5% of all dragon pokemon are legendary, and 50% of all flying type are legendary!

```r
mosaicplot(t(Leg_Type),
           main = "Legendary status among Pokemon type",
           ylab = "Legendary Status",
           xlab = "Pokemon Type",
           shade = TRUE)
```

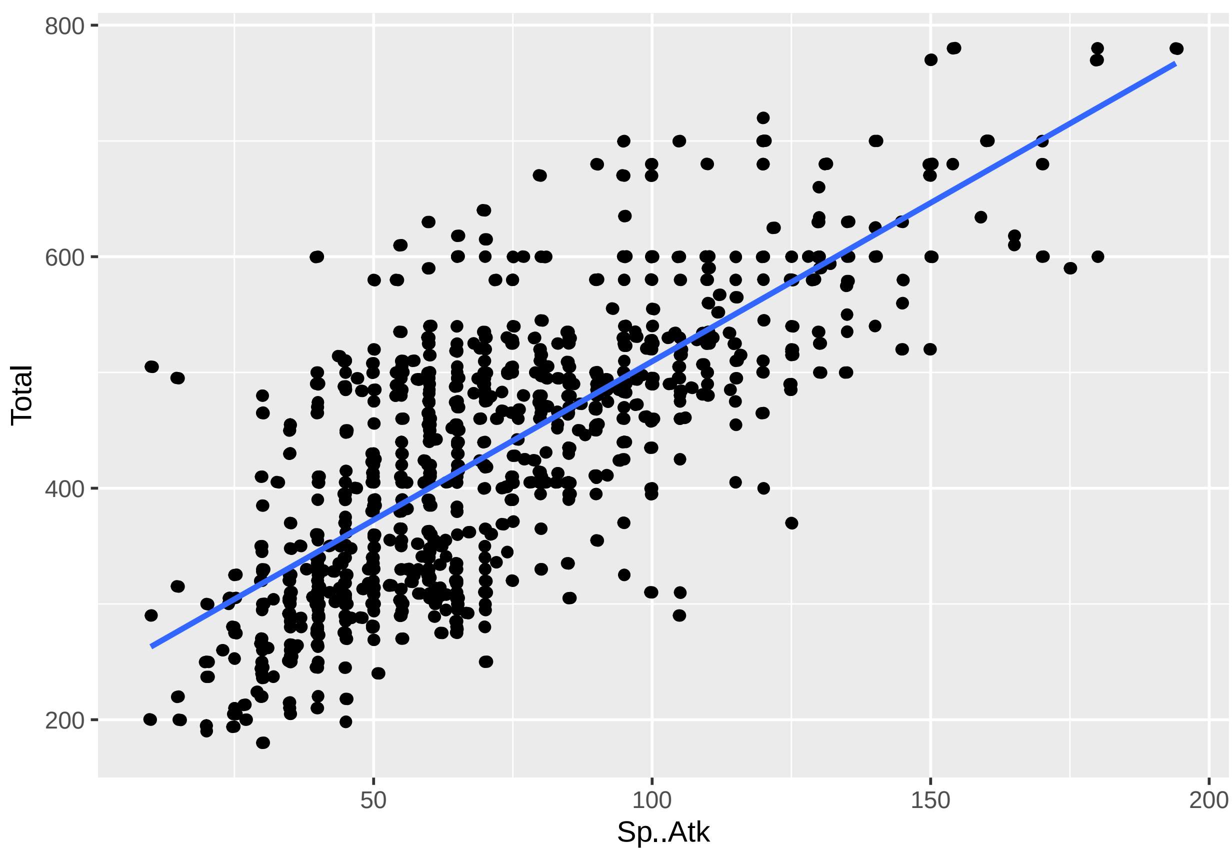


Based on the mosaic plot, Legendary type pokemon are overrepresented as Dragon, Flying, and Psychic type pokemon, while underrepresented as Normal type.

###Can a linear model predict Total based on a Pokemons Special Attack stats?

```
#Exploratory plot plotting Total and Special Attack
ggplot(pkmn, aes(x = Sp..Atk, y = Total)) +
  geom_point() +
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE)
```

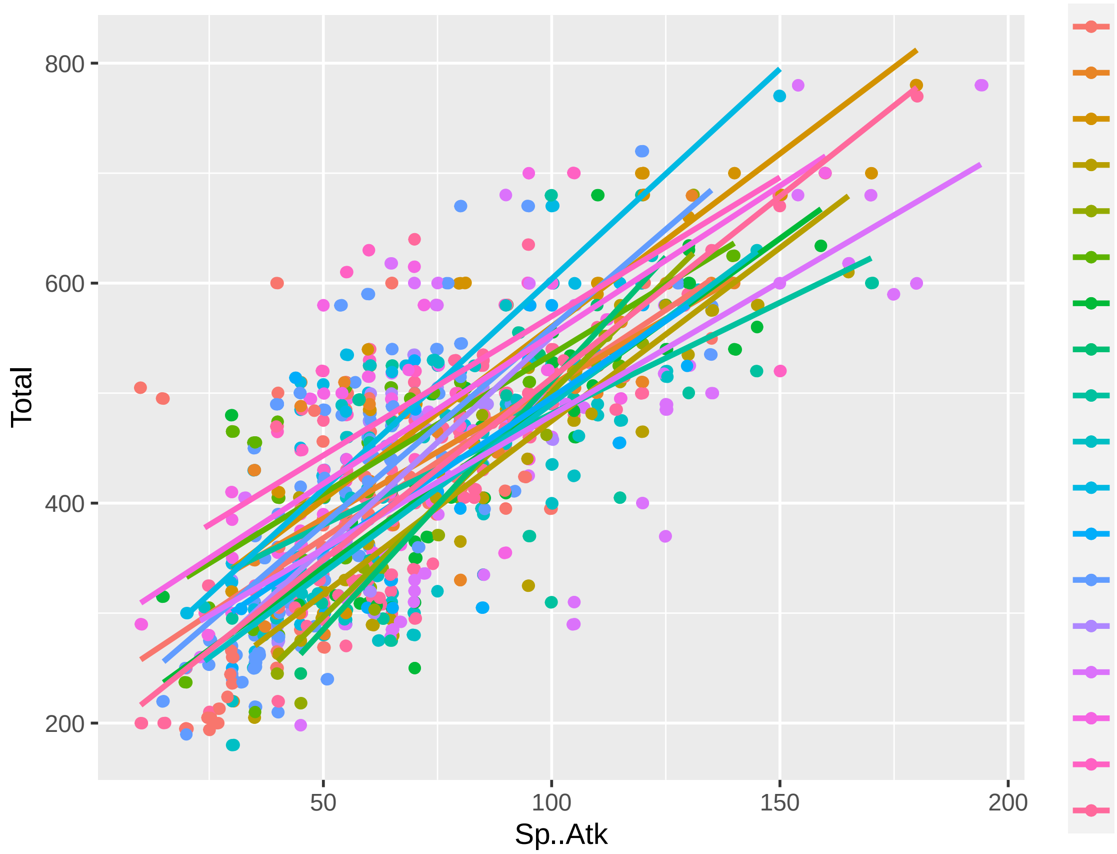
```
## `geom_smooth()` using formula 'y ~ x'
```



This model looks good, but let's see if there is a difference based on Pokemon Type

```
ggplot(pkmn, aes(x = Sp..Atk, y = Total, color = Type.1)) +
  geom_point() +
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE)
```

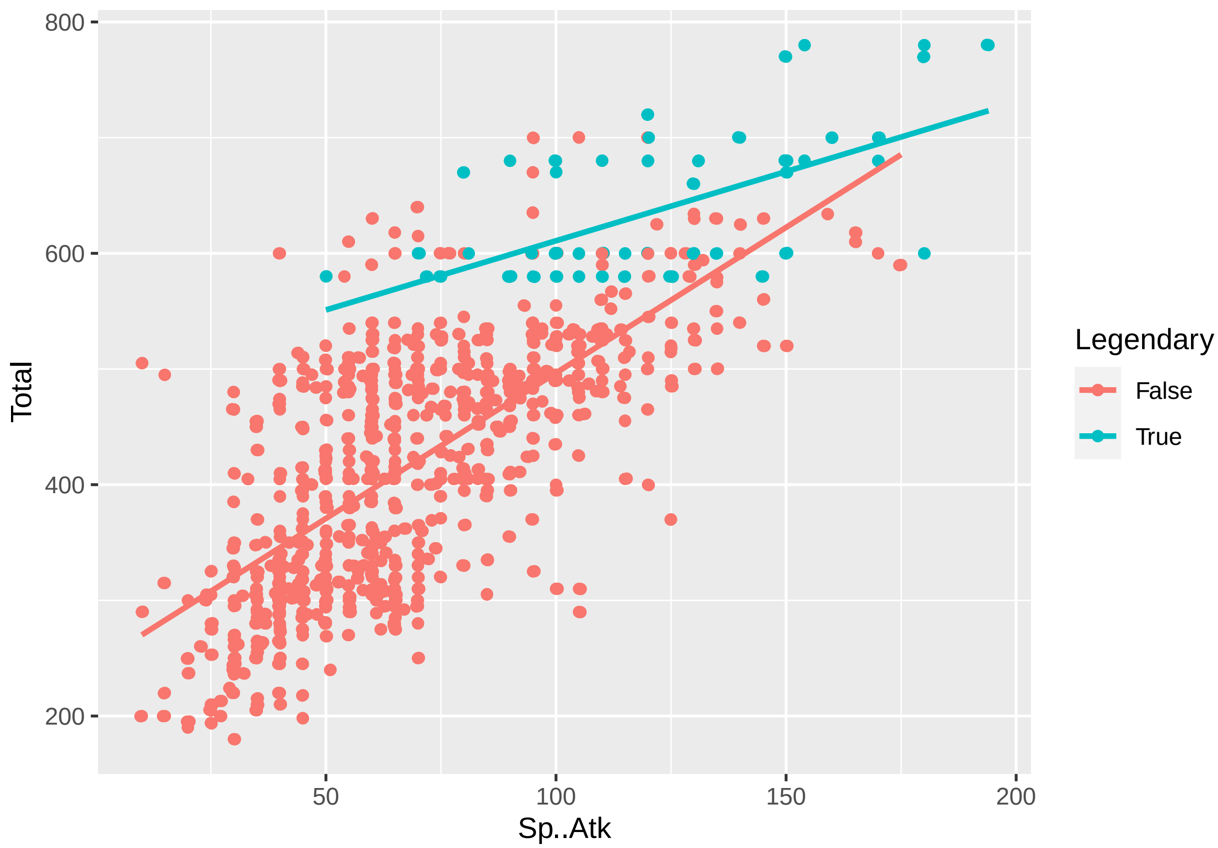
```
## `geom_smooth()` using formula 'y ~ x'
```



All of the lines appear to follow the same general trend despite their pokemon Type, but now, lets look at Legendary status.

```
ggplot(pkmn, aes(x = Sp..Atk, y = Total, color = Legendary)) +
  geom_point() +
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE)
```

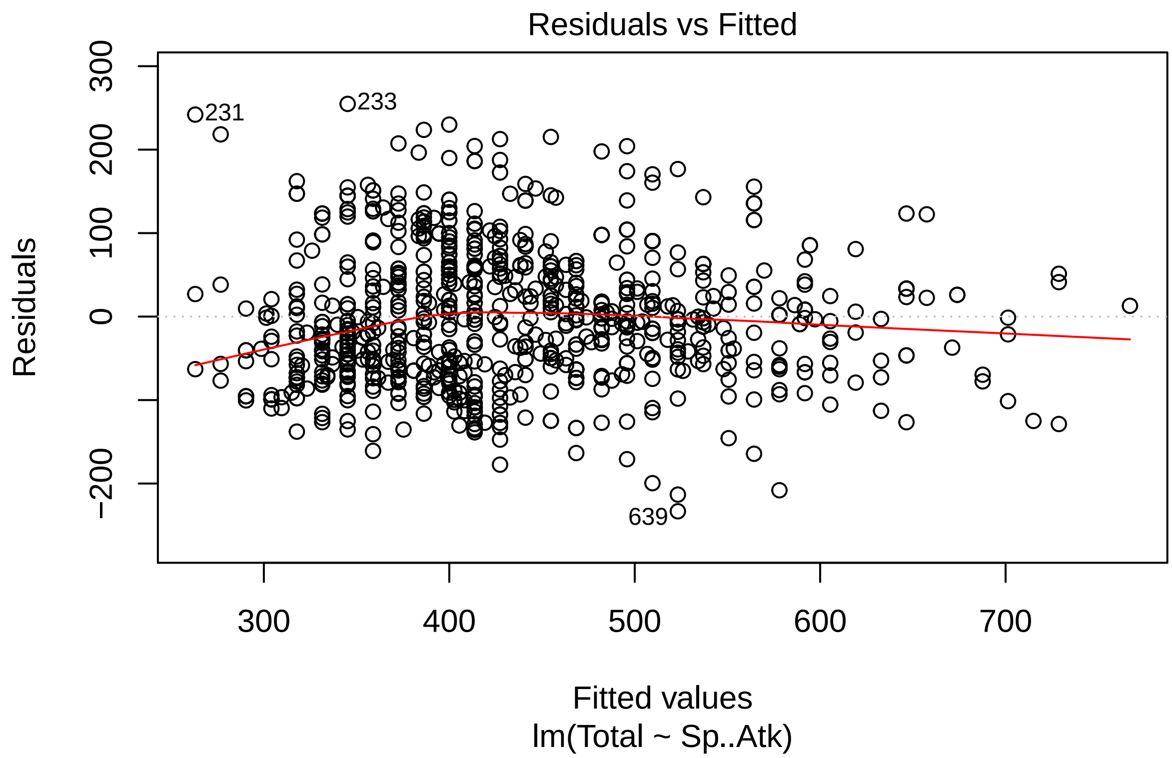
```
## `geom_smooth()` using formula 'y ~ x'
```

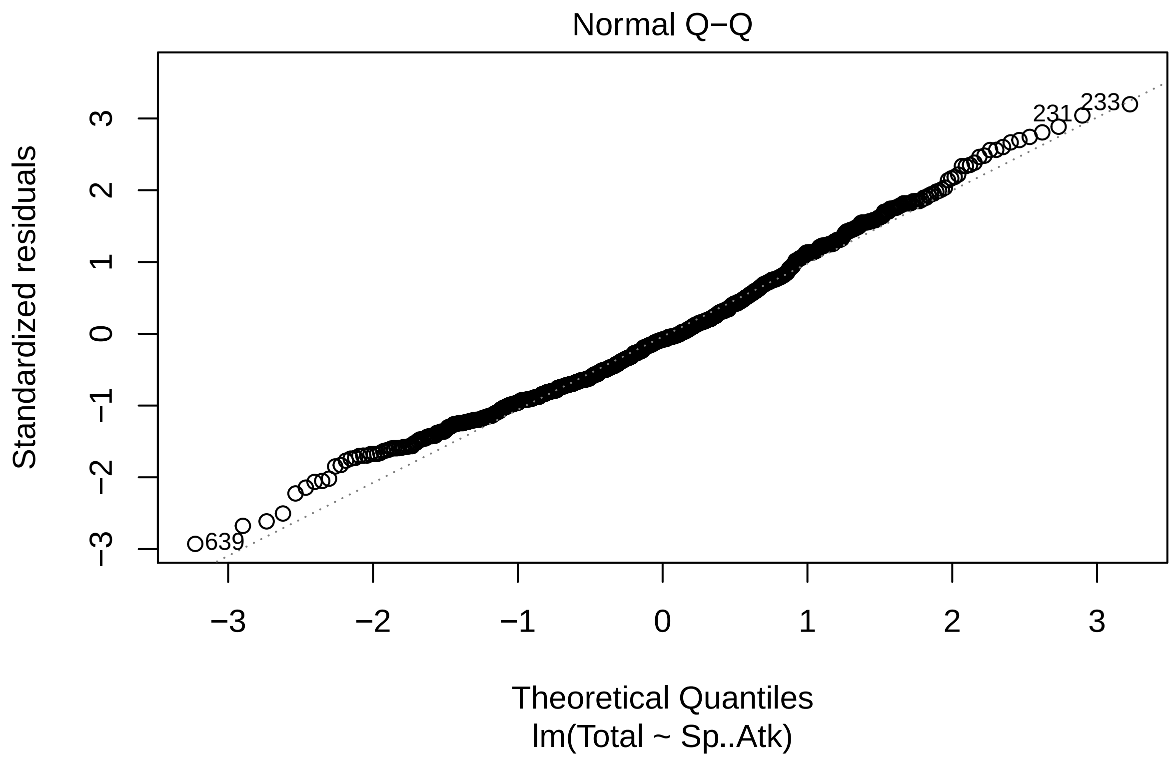


Legendary pokemon appear to overall have higher scores, higher special attacks and have slopes are different than non-Legendary Pokemon.

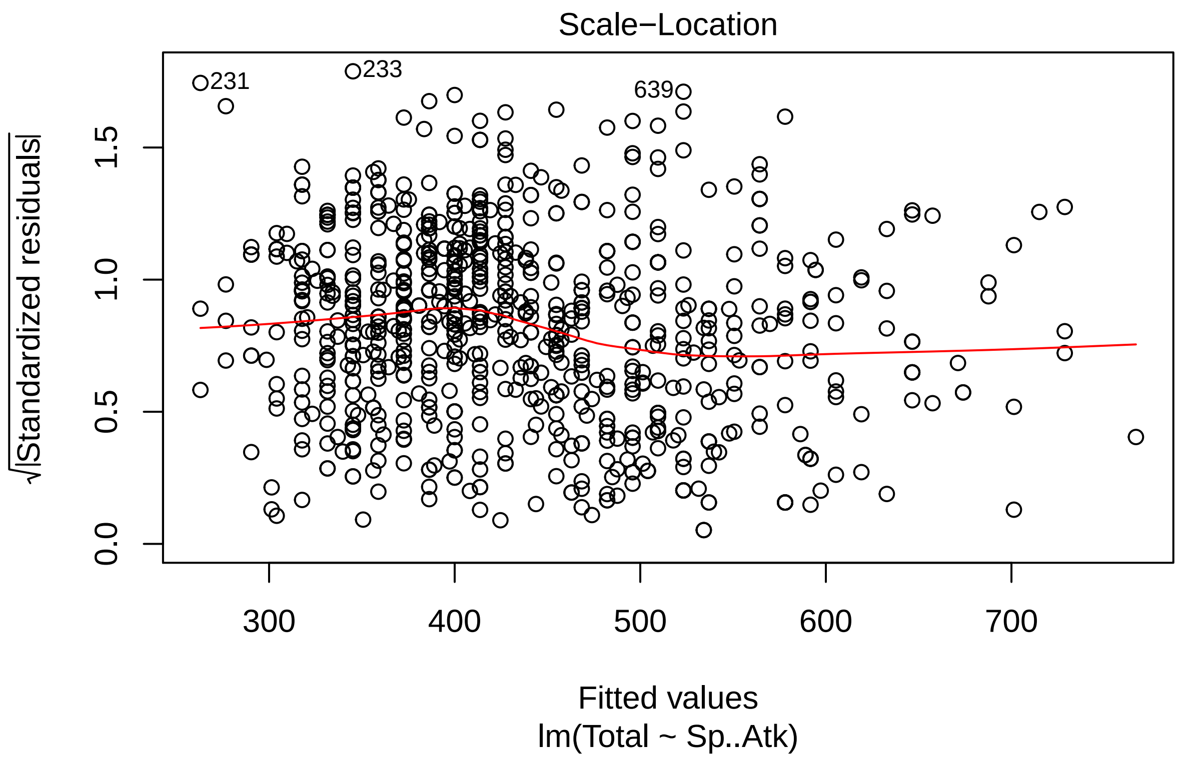
```
#Simple Linear model of Total as predicted by Special Attack alone
SpAtkmod1 <- lm(Total ~ Sp..Atk, data = pkmn)
#Linear model integrating Legendary Status to Sp.Atk
SpAtkmod2 <- lm(Total ~ Sp..Atk * Legendary, data = pkmn)
```

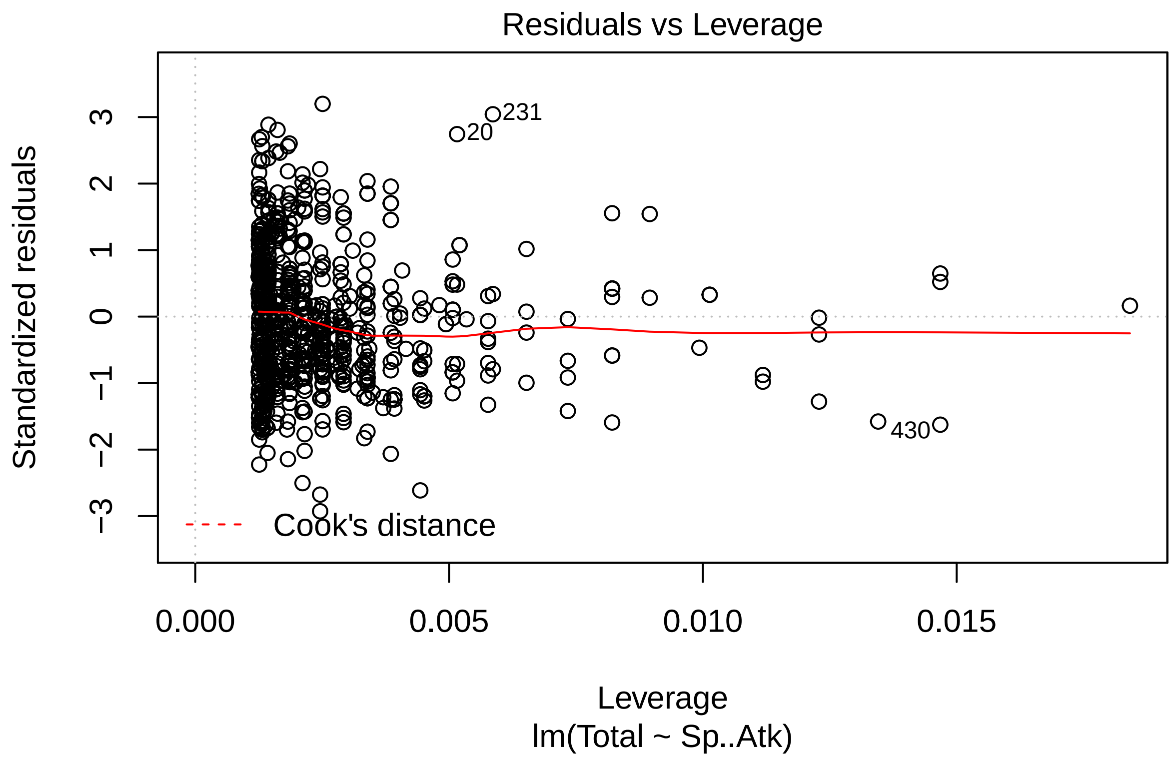
```
#Diagnostics and analysis of both models
plot(SpAtkmod1)
```


Residuals vs Fitted
Residuals
300
200
100
0
−200
231
233
639
300
400
500
600
700
Fitted values
lm(Total ~ Sp..Atk)



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Total ~ Sp..Atk)
639
231
233
−3
−2
−1
0
1
2
3

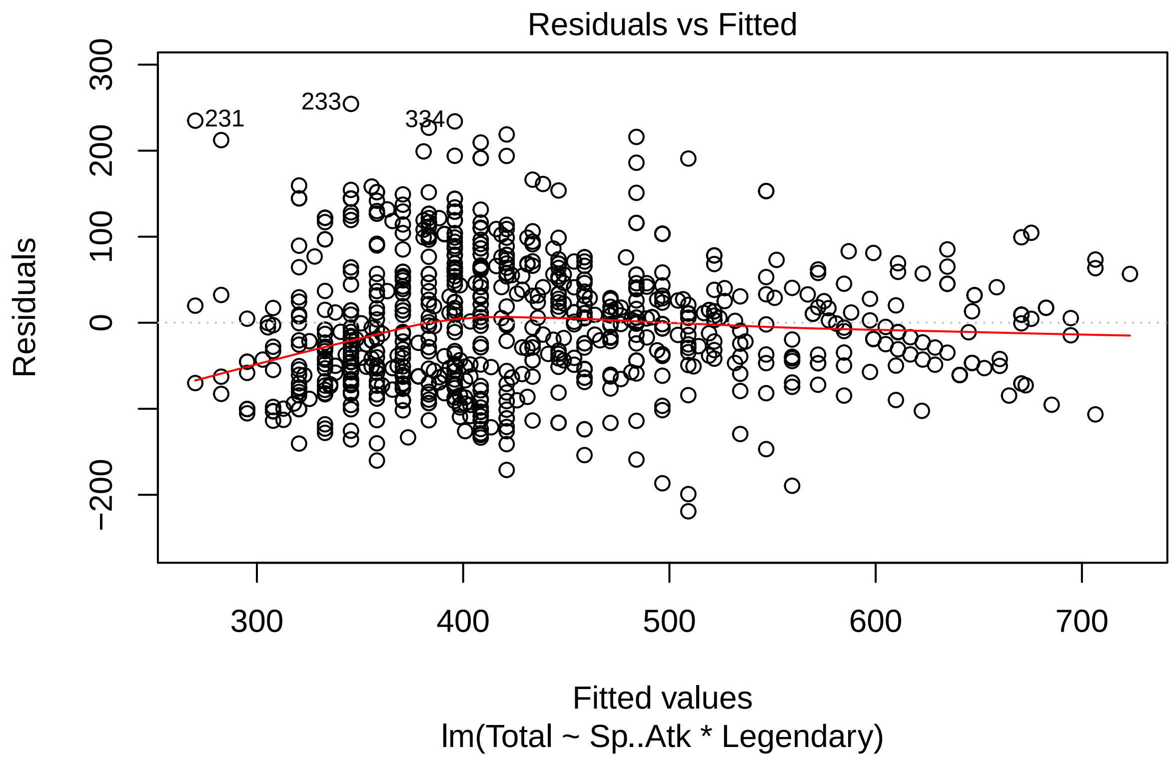

Scale−Location
231
233
639
√|Standardized residuals|
0.0
0.5
1.0
1.5
300
400
500
600
700
Fitted values
lm(Total ~ Sp..Atk)

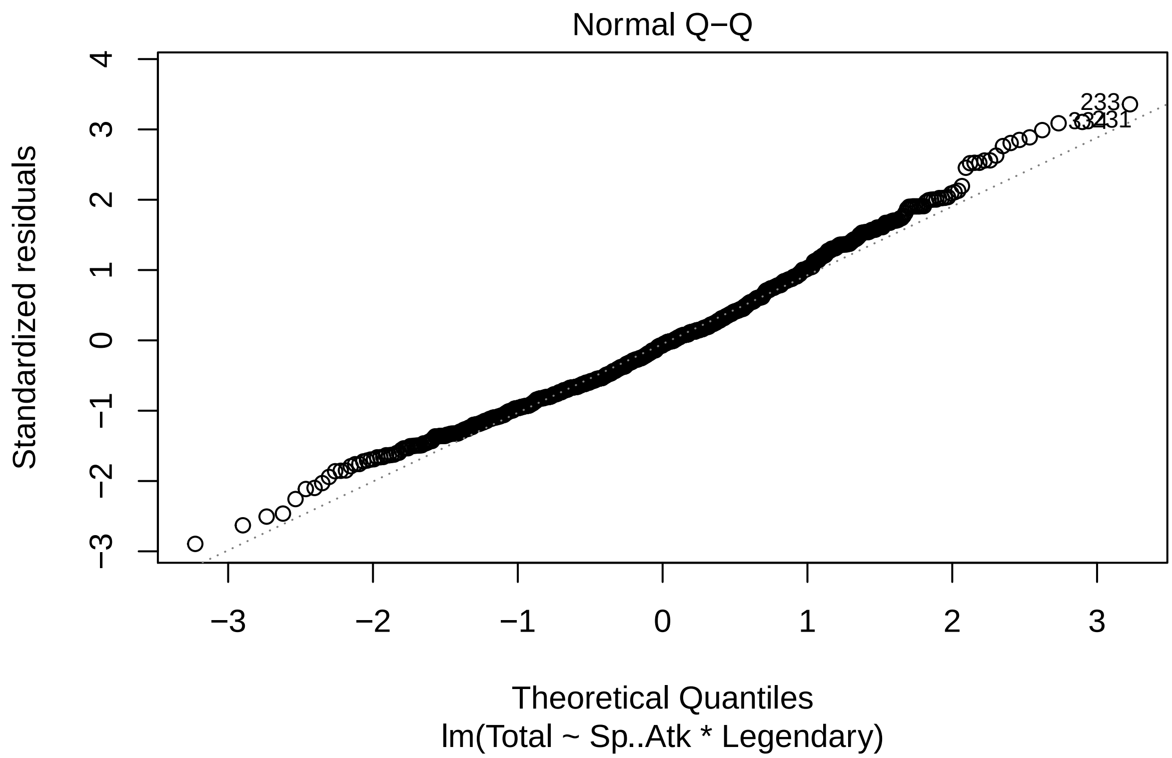


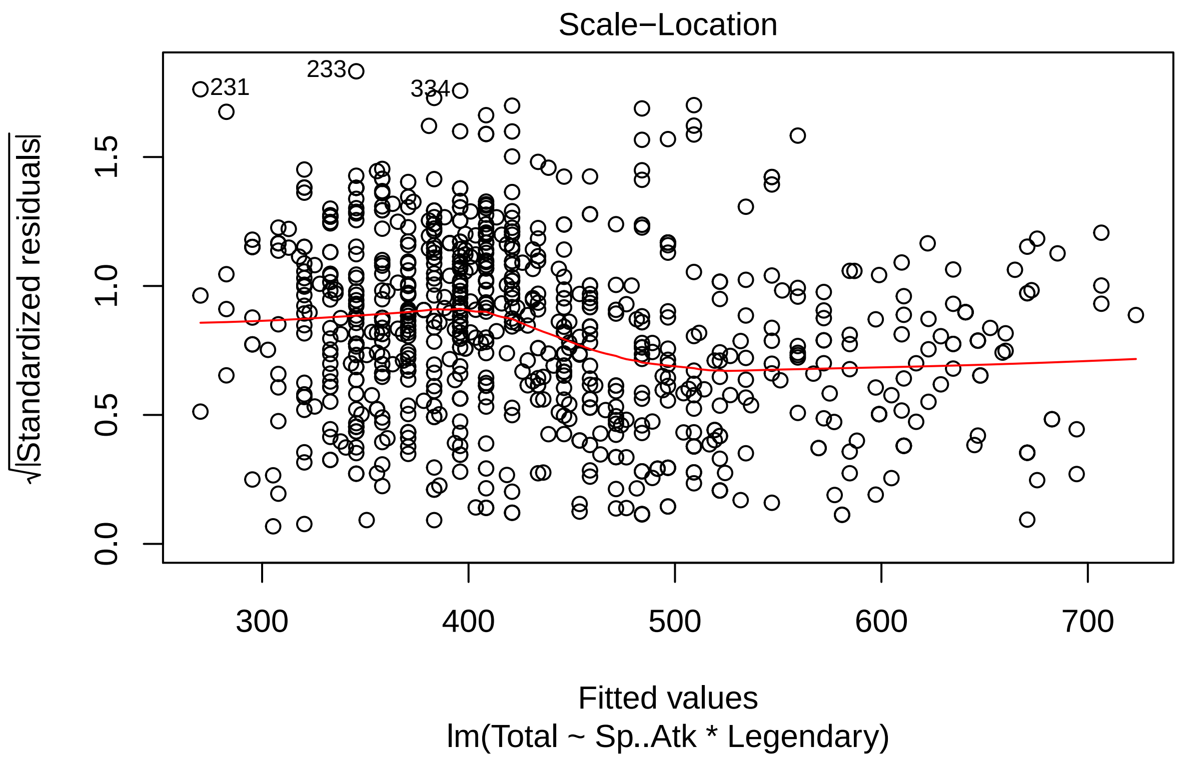

```
summary(SpAtkmod1)
```

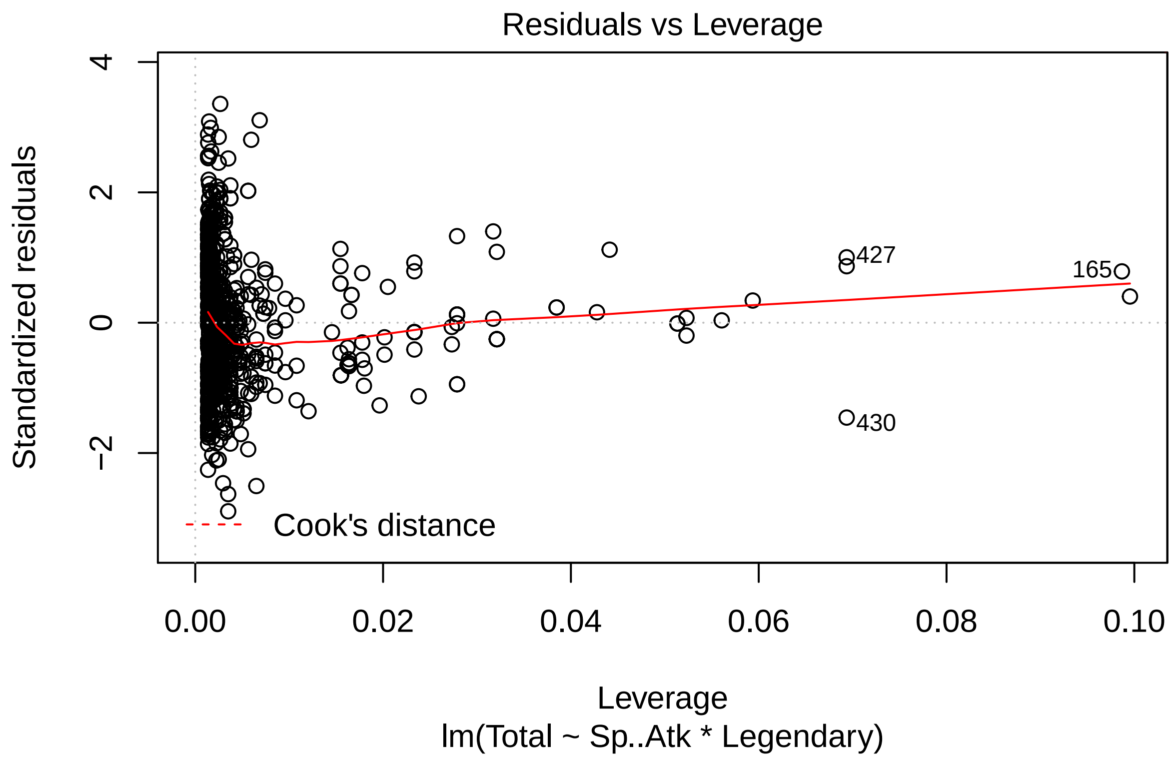
```
##
## Call:
## lm(formula = Total ~ Sp..Atk, data = pkmn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -233.26  -57.85   -6.39   51.37  254.81
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 235.61291    6.88444   34.22   <2e-16 ***
## Sp..Atk       2.73949    0.08624   31.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.77 on 798 degrees of freedom
## Multiple R-squared:  0.5584, Adjusted R-squared:  0.5578
## F-statistic:  1009 on 1 and 798 DF,  p-value: < 2.2e-16
```

```
plot(SpAtkmod2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Total ~ Sp..Atk * Legendary)
231
233
334
300
200
100
0
−200
300
400
500
600
700


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Total ~ Sp..Atk * Legendary)
233
332
231


Scale−Location
231
233
334
√|Standardized residuals|
0.0
0.5
1.0
1.5
300
400
500
600
700
Fitted values
lm(Total ~ Sp..Atk * Legendary)




```
summary(SpAtkmod2)
```

```
##
## Call:
## lm(formula = Total ~ Sp..Atk * Legendary, data = pkmn)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -219.176  -53.848   -5.108   46.168  254.388
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            244.95739    7.15879  34.218  < 2e-16 ***
## Sp..Atk                  2.51636    0.09626  26.142  < 2e-16 ***
## LegendaryTrue          246.26419   39.08333   6.301 4.89e-10 ***
## Sp..Atk:LegendaryTrue   -1.32012    0.31972  -4.129 4.03e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.87 on 796 degrees of freedom
## Multiple R-squared:  0.6016, Adjusted R-squared:  0.6001
## F-statistic: 400.6 on 3 and 796 DF,  p-value: < 2.2e-16
```

Both models are randomly sampled, Residuals are independent from one another, Residuals are normally distributed based on both QQ plots, there does appear to be funneling of residuals Based on the summary of *the data, the better model is SpAtkmod2, with a higher R-squared value, the Multiple R-squared value for SpAtkmod1 = 0.5584, while the Adjusted R-squared value for SpAtkmod2 = 0.6001, indicating that this model is a better predictor of Pokemon Total score.*

Total (Non-Legendary) = 244.95839 + 2.51636 (Special Attack) Total (Legendary) = 491.22158 + 1.19624 (Special Attack)