
Group 4 Presentation

*Tanner Koscinski, Richard Tran, James Tang, Shu Jiang*

# Background

- Turn based Role Playing Game
- Pokémon can be thought of as animal-like creatures
  - Differentiated by stat points and type
- Pokémon Battles
  - When their health (HP) drops all the way down to zero, they faint

# Question

- Can we predict a Pokémon’s HP based on their stats?
- What factors determine a Pokémon's HP?

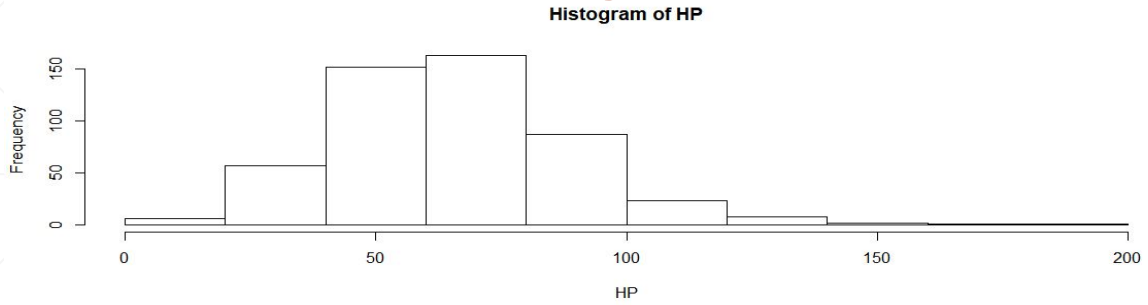
# Pokémon Statistics / Variables

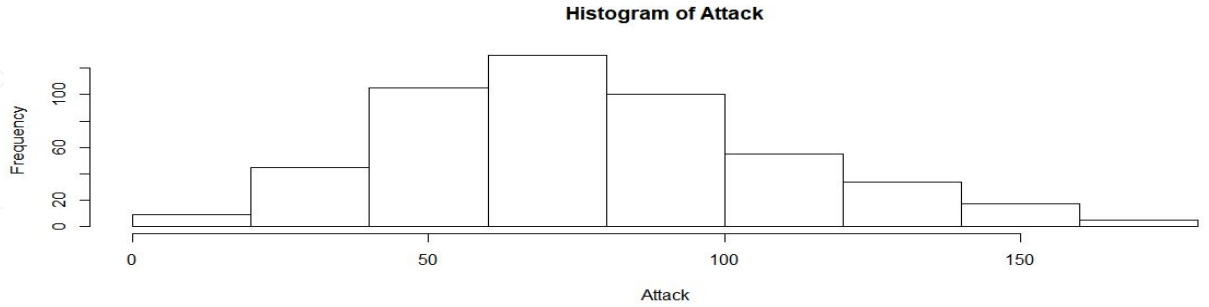
- **HP** determines the Hit Points of a Pokémon.
- **Attack** determines the physical power of a Pokémon.
- **Defense** determines how much damage a Pokémon can resist from Physical Attacks.
- **Special Attack** determines the special attack power of your Pokémon.
- **Special Defense** determines how much damage a Pokémon can resist from Special Attacks.
- **Speed** determines the order Pokémon will strike in Battle.

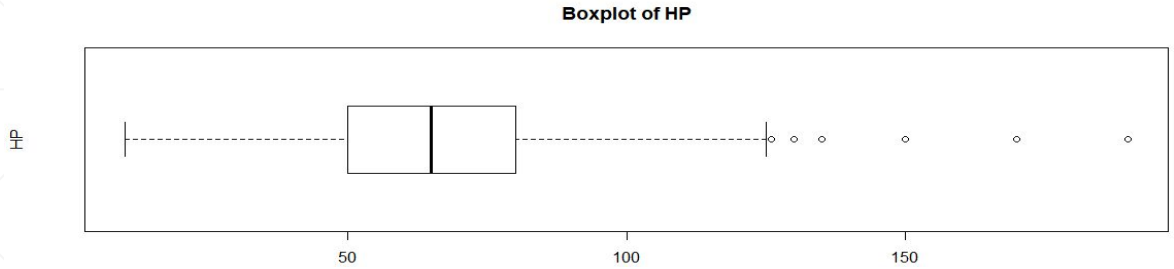
# Data Set

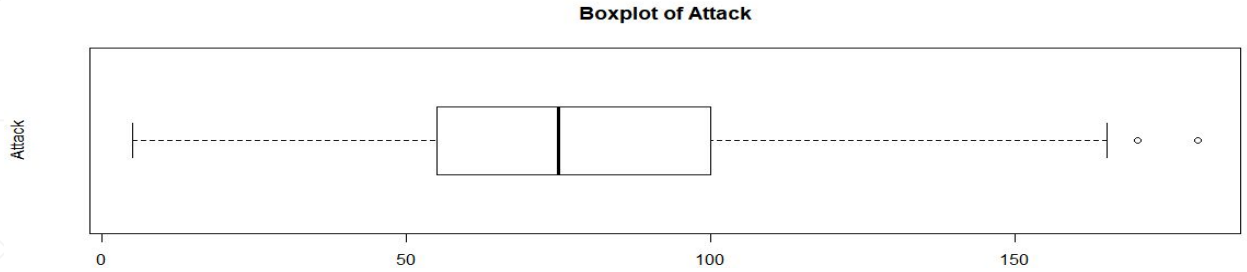
- 800 Pokémon across 6 generations
  - Each Pokémon has one or two of 18 different elemental types:
    - Normal, Fire, Water, Electric, Grass, Ice, Fighting, Poison, Ground, Flying, Psychic, Bug, Rock, Ghost, Dragon, Dark, Steel, Fairy
  - 6 base stats: HP, Attack, Defense, Special Attack, Special Defense, Speed
  - Legendary Identifier

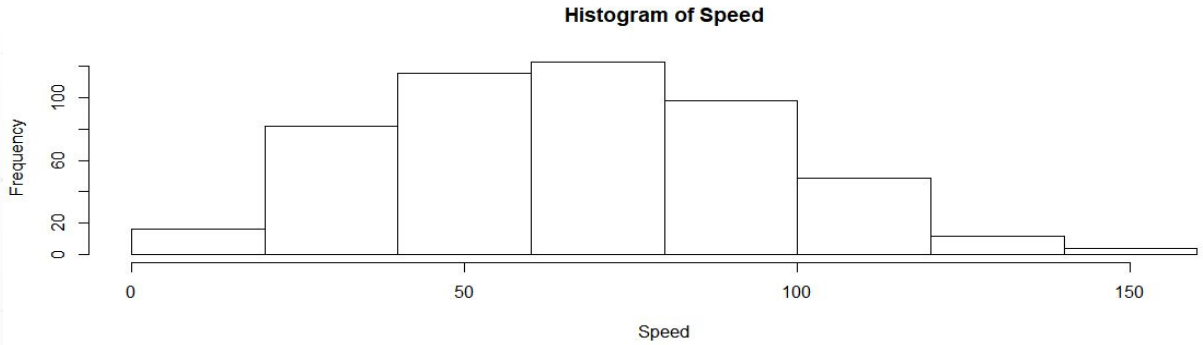
# Check normality of variables

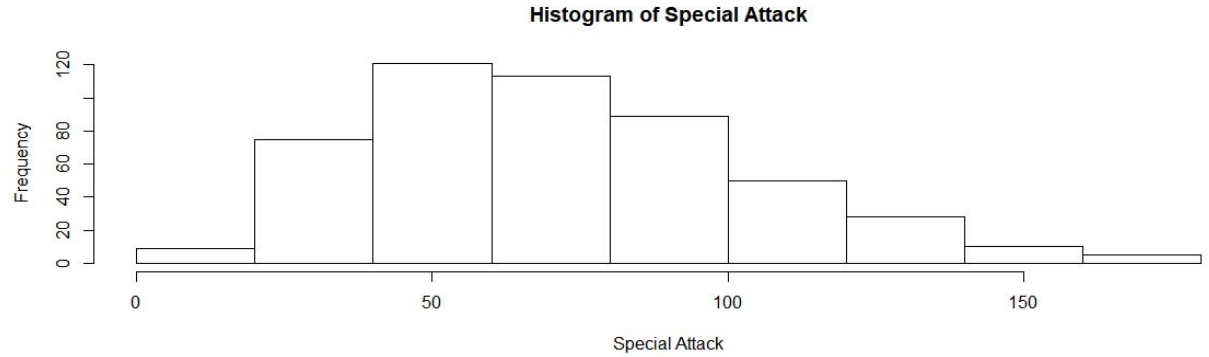


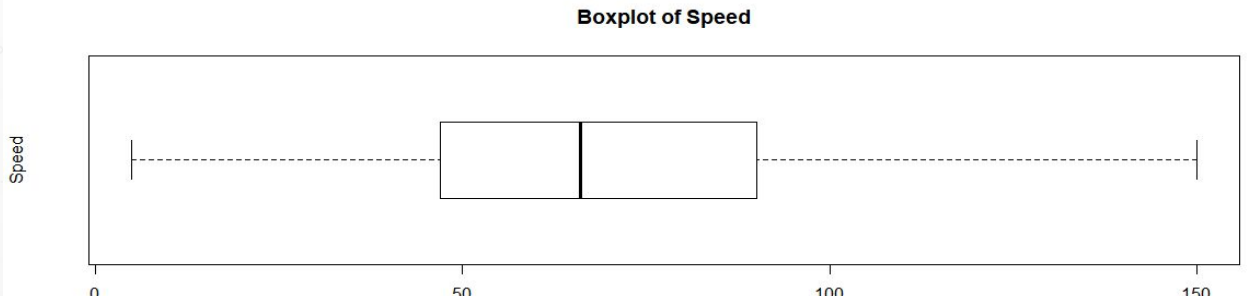



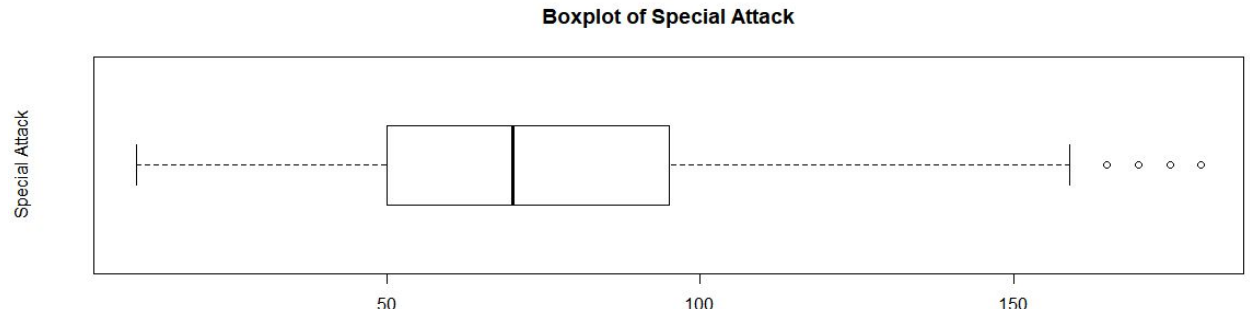



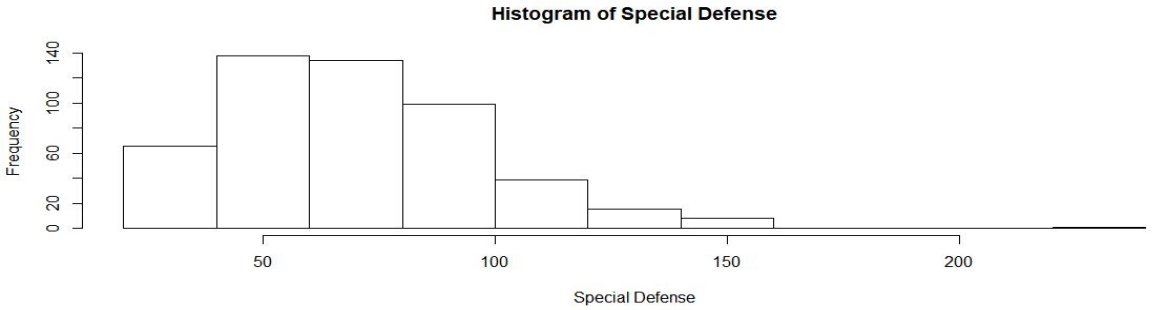



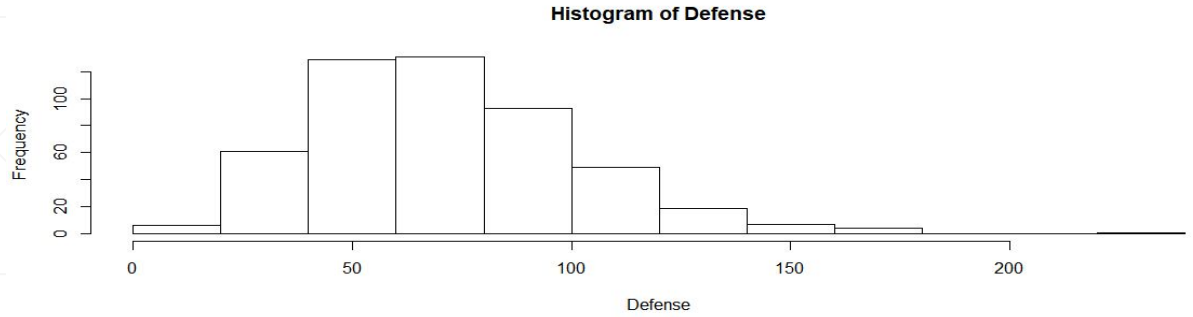



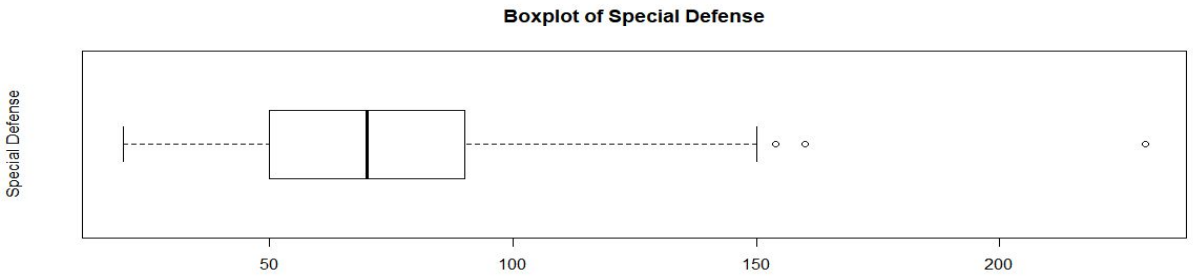



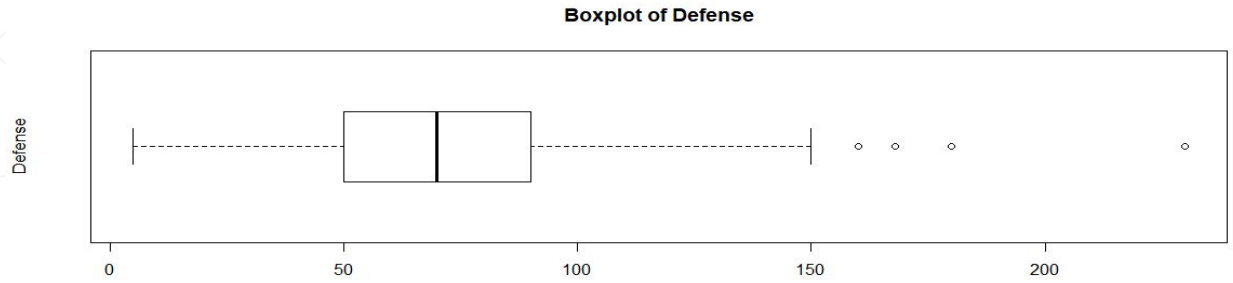





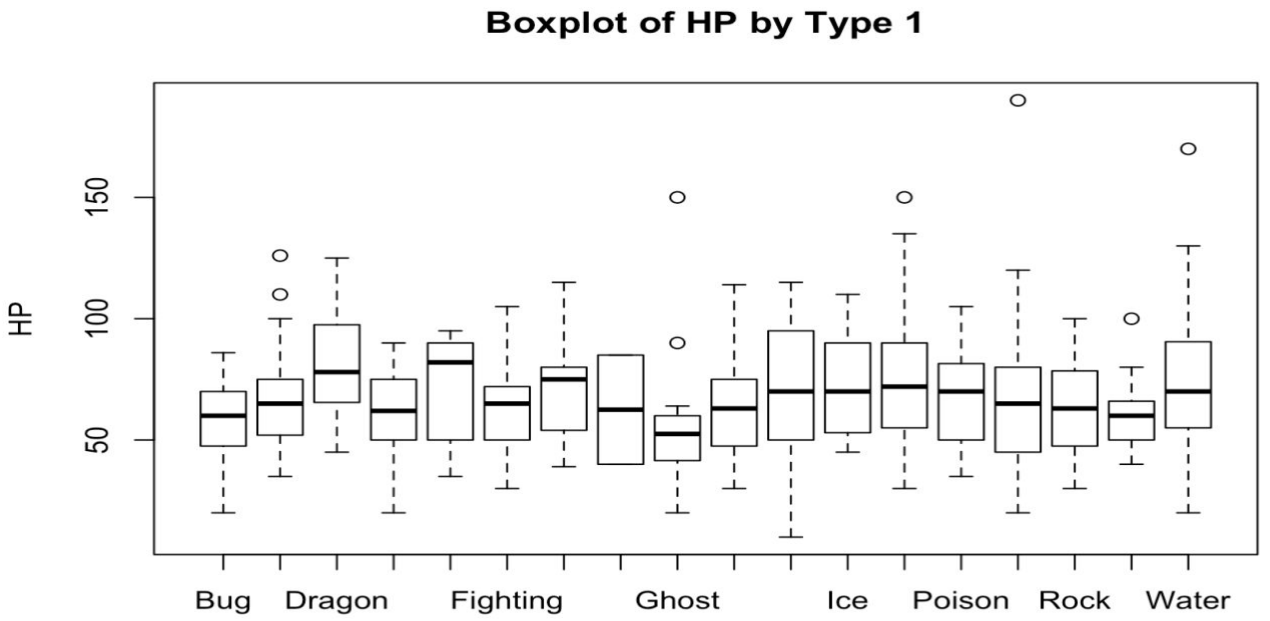



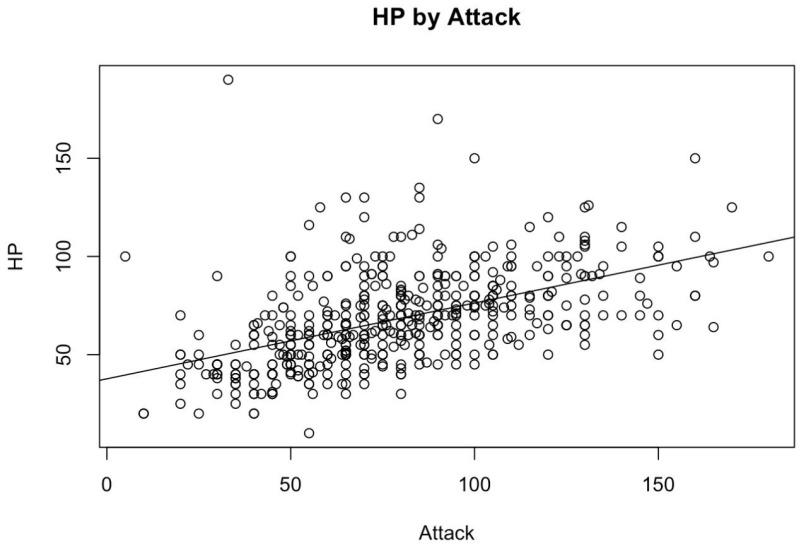



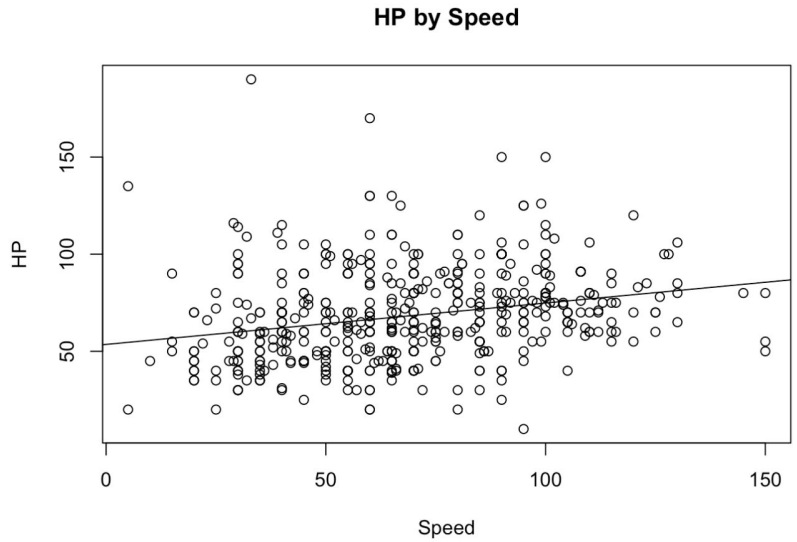

```
boxplot(data$hp ~ data$type1, main = 'Boxplot of HP by Type 1', ylab = 'HP')
```

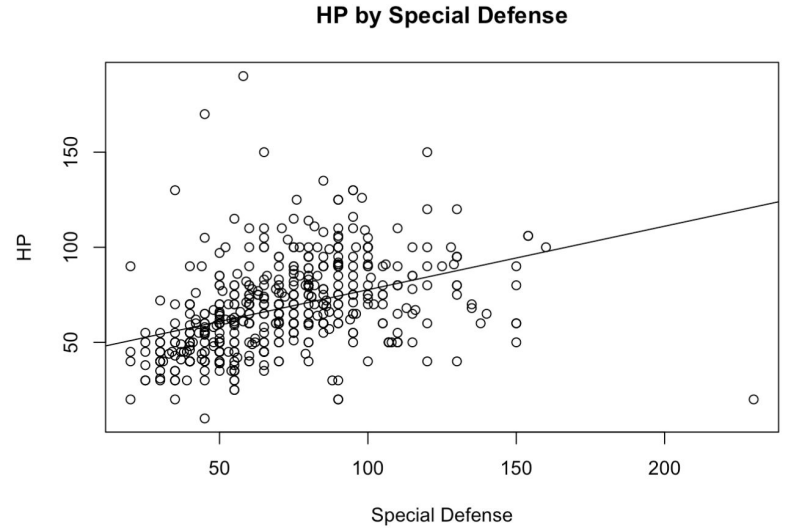
```
table(data$type1) + table(data$type2)[2:19]
```

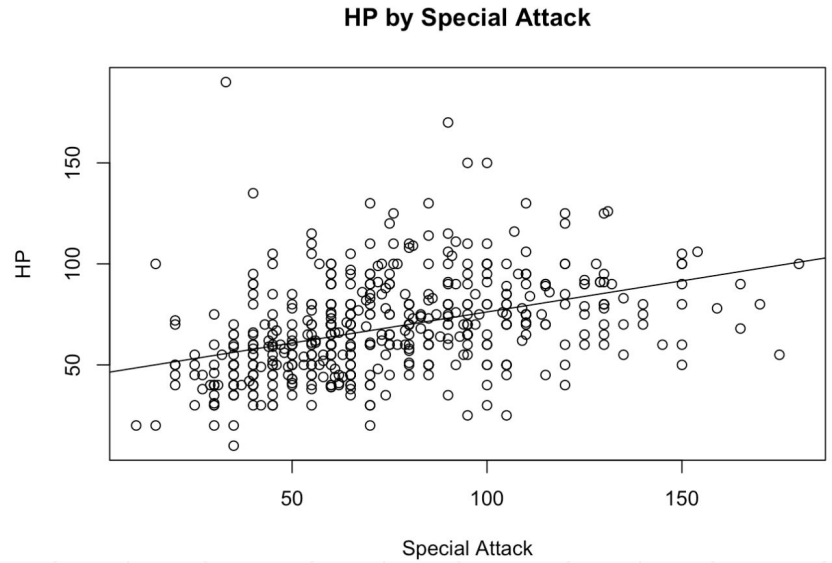
```
##
##      Bug     Dark   Dragon Electric     Fairy Fighting     Fire   Flying
##       44       39       33       30       24       36       35       67
##    Ghost    Grass   Ground      Ice   Normal   Poison  Psychic     Rock
##       24       58       42       29       56       43       52       41
##    Steel    Water
##       31       85
```

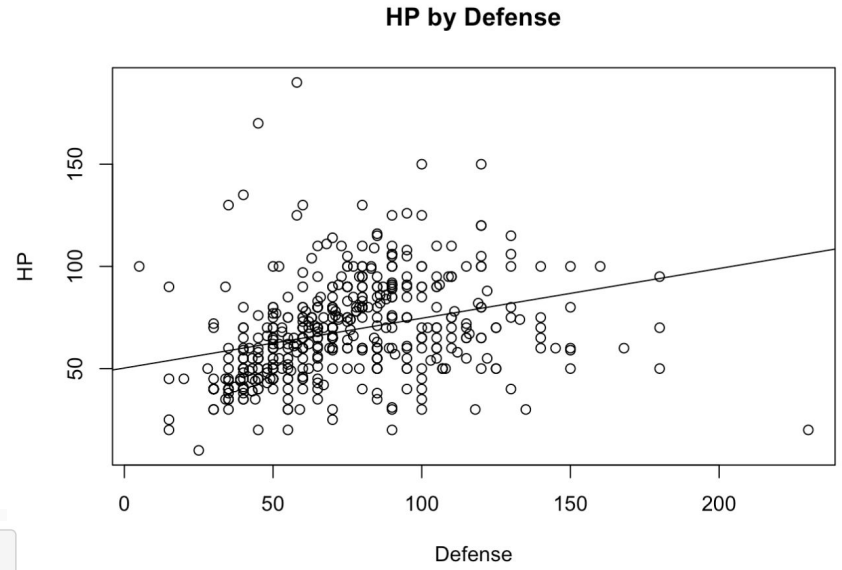




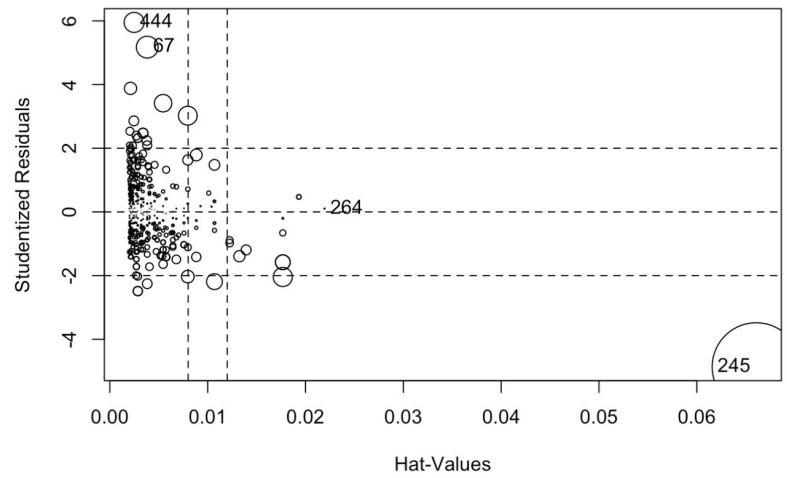




```
sort(tapply(data$hp, data$type1, mean, na.rm = T) - mean(data$hp))
```

```
##      Ghost        Bug      Steel   Electric   Fighting       Rock
## -11.460000  -9.372791  -7.397500  -6.970000  -6.114762  -5.817143
##     Flying      Grass     Poison       Dark    Psychic       Fire
##  -5.710000  -2.664545  -0.610000   0.830000   1.183939   2.754286
##      Fairy     Ground        Ice      Water     Normal     Dragon
##   3.567778   4.380909   4.526842   5.136667   5.641852  13.490000
```

```
influencePlot(lm(data$hp ~ data$spdefense))
```

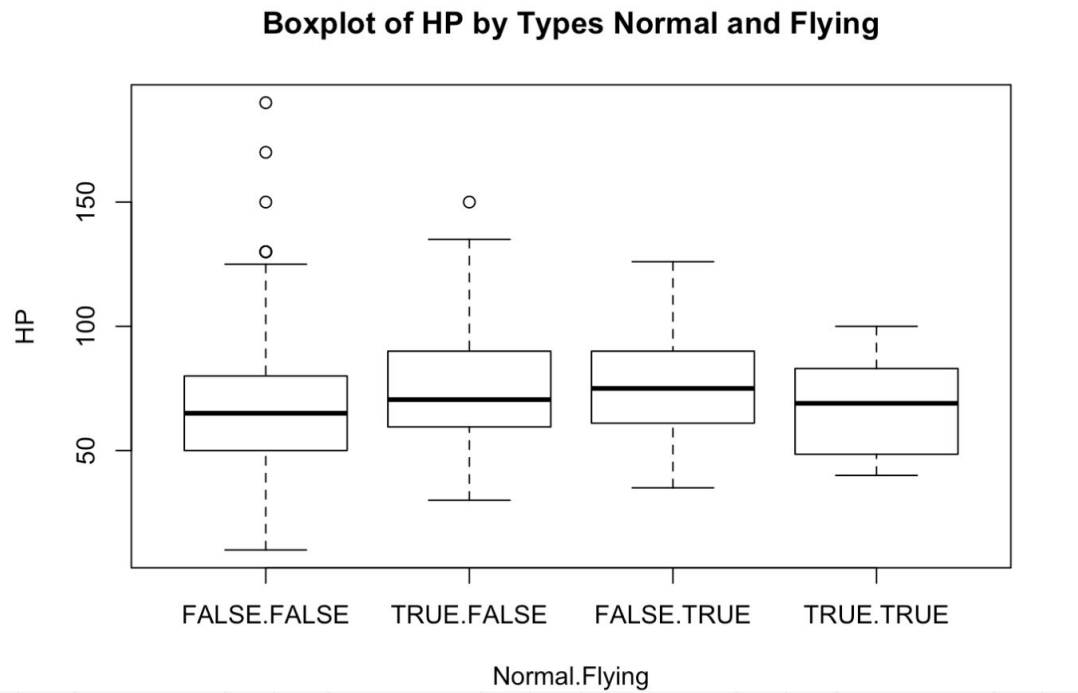


# Interaction effects

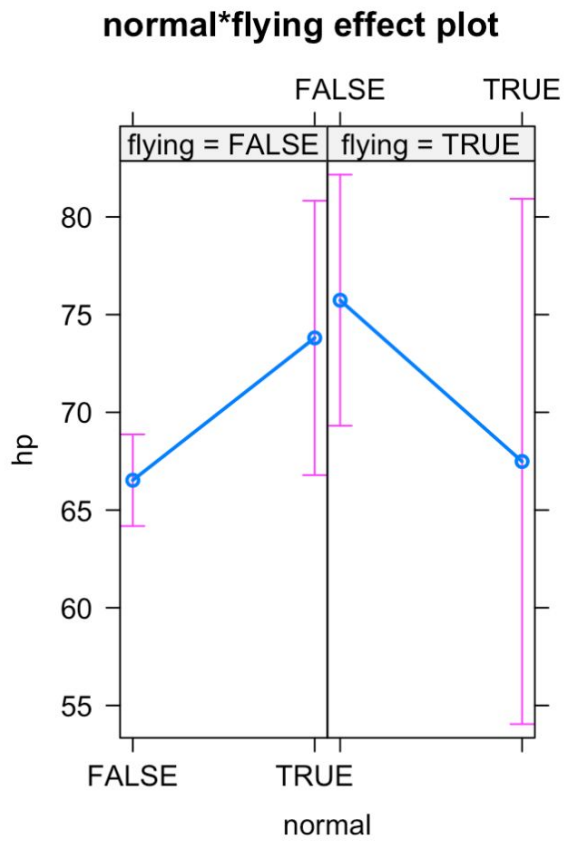
```
addmargins(table(data$normal, data$flying))
```

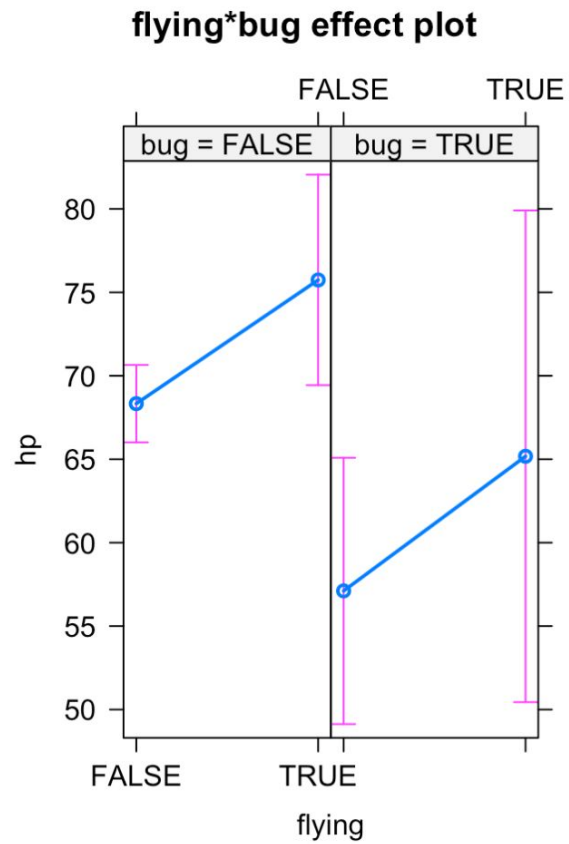
```
##
##         FALSE TRUE Sum
##   FALSE   389   55 444
##   TRUE     44   12  56
##   Sum     433   67 500
```

```
boxplot(data$hp ~ data$normal + data$flying, main = 'Boxplot of HP by Types Norm
al and Flying', xlab = 'Normal.Flying', ylab = 'HP')
```



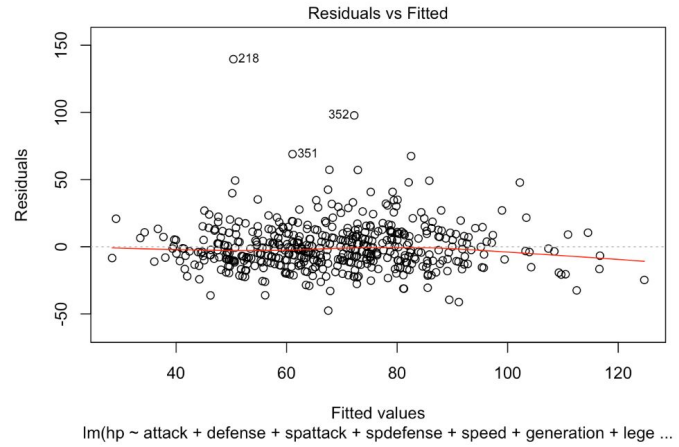

No statistically significant interaction between normal and flying types on HP or between flying and bug types on HP

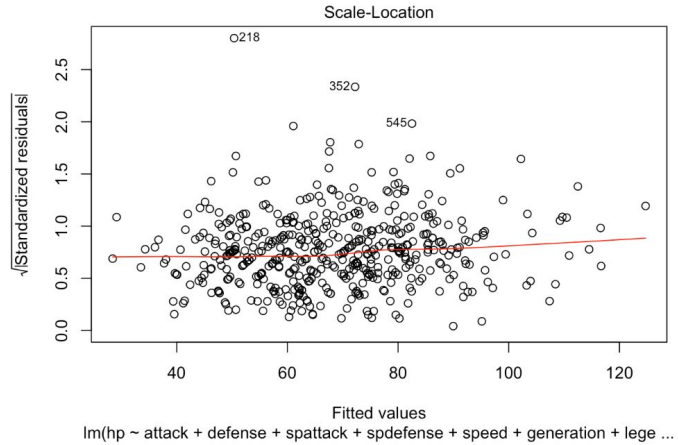


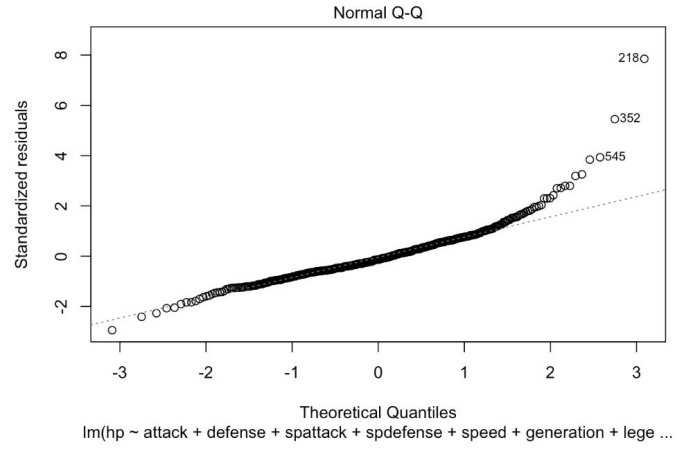



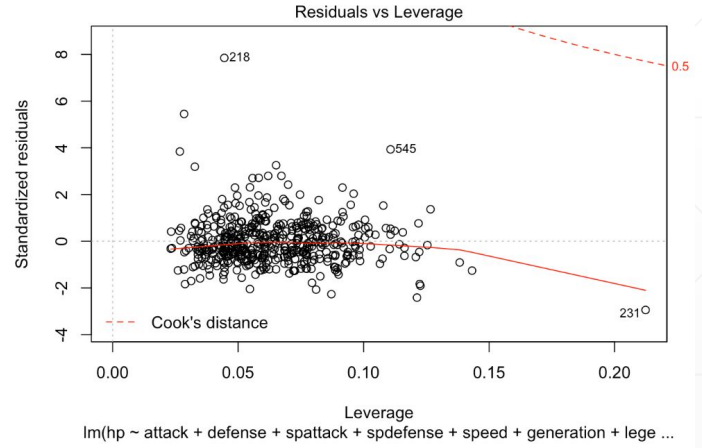
```
## Call:
## lm(formula = hp ~ attack + defense + spattack + spdefense + speed +
##     generation + legendary + bug + dark + dragon + electric +
##     fairy + fighting + fire + flying + ghost + grass + ground +
##     ice + normal + poison + psychic + rock + steel + water +
##     normal * flying + bug * flying, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.541 -10.379  -2.351   8.687 139.632
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           18.03419    4.48587   4.020 6.78e-05 ***
## attack                 0.30397    0.03653   8.321 9.64e-16 ***
## defense                0.04940    0.04130   1.196 0.232214
## spattack               0.15012    0.04015   3.739 0.000208 ***
## spdefense              0.15937    0.04136   3.853 0.000133 ***
## speed                 -0.08563    0.03788  -2.260 0.024251 *
## generation2            8.16726    2.97255   2.748 0.006236 **
## generation3            3.03058    2.65643   1.141 0.254518
## generation4            9.33817    2.95671   3.158 0.001690 **
## generation5            7.65517    2.68488   2.851 0.004548 **
## generation6            6.54225    3.40146   1.923 0.055040 .
## legendaryTrue          2.76070    3.59658   0.768 0.443117
## bugTRUE               -2.86478    3.76056  -0.762 0.446565
## darkTRUE              -5.73582    3.48824  -1.644 0.100780
## dragonTRUE            -0.20348    3.85993  -0.053 0.957982
## electricTRUE          -2.78959    4.21222  -0.662 0.508129
## fairyTRUE             -0.69349    4.28797  -0.162 0.871588
## fightingTRUE          -1.30853    3.77116  -0.347 0.728760
## fireTRUE              -4.94841    3.94232  -1.255 0.210031
## flyingTRUE             6.04851    3.17711   1.904 0.057553 .
## ghostTRUE            -10.16479    4.31876  -2.354 0.019003 *
## grassTRUE             -2.84799    3.18078  -0.895 0.371047
## groundTRUE             5.94677    3.34074   1.780 0.075713 .
## iceTRUE                5.44384    3.94553   1.380 0.168322
## normalTRUE            11.51018    3.77191   3.052 0.002406 **
## poisonTRUE             4.32315    3.50769   1.232 0.218389
## psychicTRUE           -0.10211    3.63534  -0.028 0.977604
## rockTRUE              -7.92604    3.55214  -2.231 0.026131 *
## steelTRUE            -12.45512    4.01574  -3.102 0.002041 **
## waterTRUE              6.04489    2.94185   2.055 0.040454 *
## flyingTRUE:normalTRUE -8.38795    6.69048  -1.254 0.210572
## bugTRUE:flyingTRUE    -3.17184    7.41655  -0.428 0.669088
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.2 on 468 degrees of freedom
## Multiple R-squared:  0.455,  Adjusted R-squared:  0.4189
## F-statistic:  12.6 on 31 and 468 DF,  p-value: < 2.2e-16
```

# Plot of the Full Model



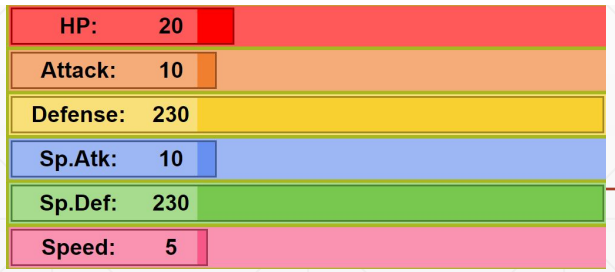

Multiple R-squared: 0.455, Adjusted R-squared: 0.4189

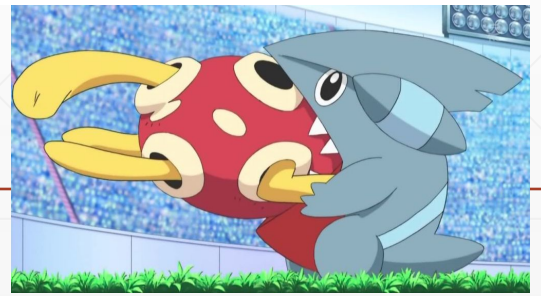
# Outliers / Leverage Points

- No transformations to data
  - Histograms and boxplots look relatively good and are not extremely skewed.
- Few outliers in the data set

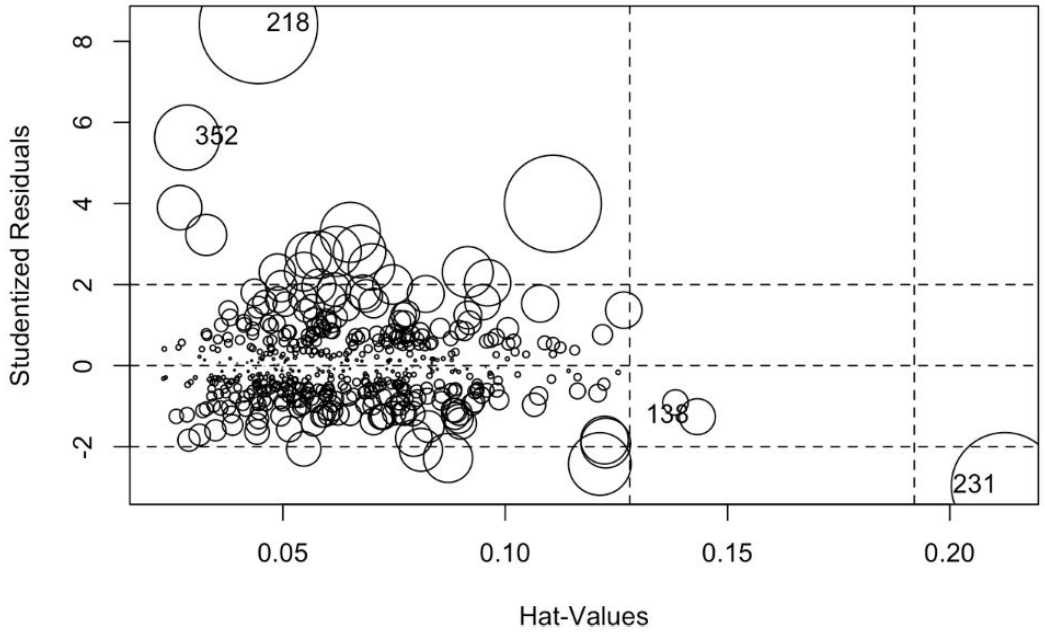
```
outlierTest(mfull)
```

```
##     rstudent unadjusted p-value Bonferonni p
## 218 8.415041         4.8362e-16   2.4181e-13
## 352 5.626309         3.1774e-08   1.5887e-05
## 545 3.994650         7.5255e-05   3.7627e-02
```



```
influencePlot(mfull)
```




```
##       StudRes        Hat       CookD
## 352  5.626309 0.02845789 0.027194658
## 231 -2.968200 0.21234868 0.073006903
## 138 -1.259418 0.14319834 0.008273783
## 218  8.415041 0.04449261 0.089666656
```

Removing the leverage point helped a little:
Before: Multiple R-squared: 0.455, Adjusted R-squared: 0.4189
After: Multiple R-squared: 0.4606, Adjusted R-squared: 0.4248

```
## Call:
## lm(formula = hp ~ attack + defense + spattack + spdefense + speed +
##     generation + legendary + bug + dark + dragon + electric +
##     fairy + fighting + fire + flying + ghost + grass + ground +
##     ice + normal + poison + psychic + rock + steel + water +
##     normal * flying + bug * flying, data = data[-which(t == 231),
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.218 -10.322  -2.081   8.288 138.814
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           17.07571    4.46060   3.828 0.000147 ***
## attack                 0.28576    0.03675   7.777 4.81e-14 ***
## defense                0.06896    0.04148   1.662 0.097116 .
## spattack               0.13972    0.03997   3.495 0.000519 ***
## spdefense              0.18971    0.04227   4.487 9.09e-06 ***
## speed                 -0.08951    0.03759  -2.381 0.017665 *
## generation2            8.71213    2.95376   2.950 0.003343 **
## generation3            3.24680    2.63555   1.232 0.218597
## generation4            9.25717    2.93247   3.157 0.001698 **
## generation5            7.74150    2.66291   2.907 0.003821 **
## generation6            6.49197    3.37347   1.924 0.054909 .
## legendaryTrue          2.53639    3.56774   0.711 0.477486
## bugTRUE               -1.36726    3.76353  -0.363 0.716551
## darkTRUE              -5.47848    3.46058  -1.583 0.114073
## dragonTRUE            -0.20747    3.82812  -0.054 0.956801
## electricTRUE          -2.95818    4.17789  -0.708 0.479264
## fairyTRUE             -1.39549    4.25920  -0.328 0.743329
## fightingTRUE          -1.11493    3.74065  -0.298 0.765792
## fireTRUE              -4.97363    3.90984  -1.272 0.203978
## flyingTRUE             5.96651    3.15105   1.893 0.058909 .
## ghostTRUE            -10.39395    4.28386  -2.426 0.015631 *
## grassTRUE             -3.14884    3.15619  -0.998 0.318955
## groundTRUE             5.79020    3.31363   1.747 0.081227 .
## iceTRUE                5.28003    3.91340   1.349 0.177920
## normalTRUE            11.45817    3.74087   3.063 0.002318 **
## poisonTRUE             3.97272    3.48078   1.141 0.254318
## psychicTRUE           -0.69124    3.61084  -0.191 0.848268
## rockTRUE              -7.36291    3.52797  -2.087 0.037429 *
## steelTRUE            -13.80663    4.00859  -3.444 0.000624 ***
## waterTRUE              5.73735    2.91944   1.965 0.049981 *
## flyingTRUE:normalTRUE -7.84262    6.63788  -1.181 0.238008
## bugTRUE:flyingTRUE    -4.82948    7.37660  -0.655 0.512981
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.05 on 467 degrees of freedom
## Multiple R-squared:  0.4606, Adjusted R-squared:  0.4248
## F-statistic: 12.87 on 31 and 467 DF,  p-value: < 2.2e-16
```

Backward Selection AIC produces this model:
Multiple R-squared: 0.4546, Adjusted R-squared: 0.4318

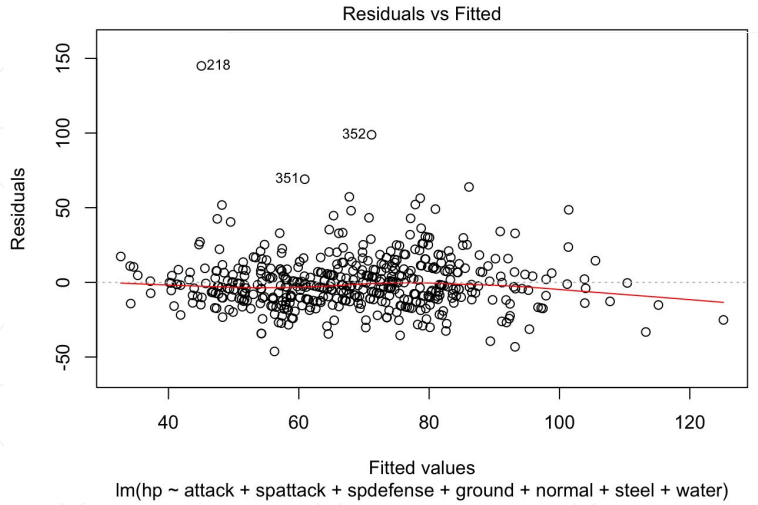
```
## Call:
## lm(formula = hp ~ attack + defense + spattack + spdefense + speed +
##     generation + dark + flying + ghost + ground + ice + normal +
##     poison + rock + steel + water, data = data[-which(t == 231),
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.650 -10.614  -2.575   8.437 140.565
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.97162    3.75668   3.719 0.000224 ***
## attack        0.29368    0.03291   8.925  < 2e-16 ***
## defense       0.06911    0.04098   1.687 0.092350 .
## spattack      0.13027    0.03467   3.758 0.000192 ***
## spdefense     0.20067    0.04088   4.909 1.26e-06 ***
## speed        -0.08354    0.03671  -2.276 0.023306 *
## generation2   8.58241    2.89323   2.966 0.003164 **
## generation3   3.37154    2.57415   1.310 0.190902
## generation4   9.24855    2.86732   3.226 0.001344 **
## generation5   7.80147    2.61605   2.982 0.003008 **
## generation6   6.37553    3.28490   1.941 0.052863 .
## darkTRUE     -4.52930    3.14007  -1.442 0.149839
## flyingTRUE    4.78515    2.48000   1.929 0.054261 .
## ghostTRUE    -9.33299    3.94397  -2.366 0.018360 *
## groundTRUE    6.94635    3.06142   2.269 0.023713 *
## iceTRUE       6.76176    3.50160   1.931 0.054068 .
## normalTRUE   11.65337    2.77907   4.193 3.28e-05 ***
## poisonTRUE    4.90115    3.13649   1.563 0.118803
## rockTRUE     -6.38275    3.29147  -1.939 0.053068 .
## steelTRUE   -13.09680    3.77089  -3.473 0.000561 ***
## waterTRUE     7.29443    2.27519   3.206 0.001436 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.94 on 478 degrees of freedom
## Multiple R-squared:  0.4546, Adjusted R-squared:  0.4318
## F-statistic: 19.92 on 20 and 478 DF,  p-value: < 2.2e-16
```

# Backward Selection BIC Model

We're going to go with the BIC model cause we'd rather have less variables

```
## Call:
## lm(formula = hp ~ attack + spattack + spdefense + ground + normal +
##     steel + water, data = data[-which(t == 231), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.271 -10.977  -2.112   8.264 144.950
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.35734    2.92848   6.269 7.99e-10 ***
## attack        0.28805    0.02905   9.916  < 2e-16 ***
## spattack      0.10706    0.03316   3.229  0.00133 **
## spdefense     0.23542    0.03680   6.396 3.71e-10 ***
## groundTRUE    7.72983    3.08949   2.502  0.01267 *
## normalTRUE   11.50007    2.72321   4.223 2.87e-05 ***
## steelTRUE   -10.69788    3.51821  -3.041  0.00249 **
## waterTRUE     6.65625    2.24886   2.960  0.00323 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.46 on 491 degrees of freedom
## Multiple R-squared:  0.4064, Adjusted R-squared:  0.398
## F-statistic: 48.03 on 7 and 491 DF,  p-value: < 2.2e-16
```

Multiple R-squared: 0.4064
Adjusted R-squared: 0.3980

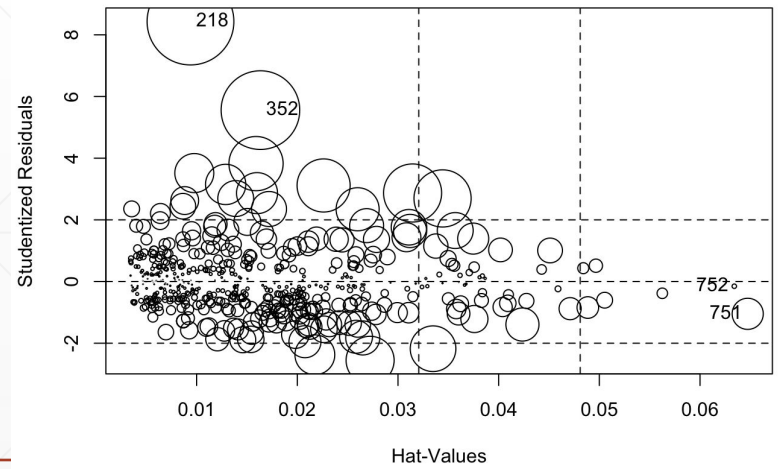



```
ncvTest(mbic)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.06875731, Df = 1, p = 0.79315
```

```
vif(mbic)
```

```
##    attack  spattack spdefense    ground    normal     steel     water
##  1.257610  1.599520  1.454313  1.076962  1.081476  1.055632  1.046175
```




NCV Test: Equality of error variance assumption holds. VIF: No problems with multicollinearity.

# Final Model?

```
## Call:
## lm(formula = hp ~ attack + spattack + spdefense + ground + normal +
##     steel + water, data = test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.731 -12.355  -3.645   6.510 169.453
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.68413    5.13633   4.806 2.47e-06 ***
## attack       0.16280    0.04880   3.336  0.00096 ***
## spattack     0.11692    0.05361   2.181  0.02999 *
## spdefense    0.29357    0.06153   4.771 2.89e-06 ***
## groundTRUE   9.57931    5.33305   1.796  0.07349 .
## normalTRUE  20.13220    4.16540   4.833 2.17e-06 ***
## steelTRUE   -3.96829    6.18219  -0.642  0.52145
## waterTRUE    2.32774    4.25522   0.547  0.58477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.59 on 292 degrees of freedom
## Multiple R-squared:  0.2492, Adjusted R-squared:  0.2312
## F-statistic: 13.85 on 7 and 292 DF,  p-value: 1.738e-15
```

Multiple R-squared: 0.2492, Adjusted R-squared: 0.2312

```
set.seed(1)
t = sample(1:800, 800)
train = data[t[1:500],]
test = data[t[501:800],]
data = train
```

- We used the predictor variables chosen from BIC Backward Selection to make a model with the testing data, and the R-squared dropped from 0.4064 to 0.2492, which is not very good.
- Note that the training data only has 500 Pokemon, and the testing data only has 300 Pokemon, so the models would benefit from a larger sample size.

# Difficulties

- Pokémon with abnormally high/low HP didn't fit well into the model
- Specialized Pokémon created outliers
- There are only 800 Pokémon in the first 6 generations. It would help if we had significantly more observations.

# Improvements

- It would be interesting to look into the difference between the lowest / middle / highest evolution forms, but we didn't have this data easily available.
- Eliminate the Pokémon with abnormally high or low HP values (+/- 2.5 standard deviations from the mean) from the dataset before creating our model.

# Final Model

BIC Backward Selection model on training data:
Multiple R-squared: 0.5047
Adjusted R-squared: 0.4964

```
## lm(formula = hp ~ attack + defense + spattack + spdefense + ghost +
##     ground + normal + steel, data = data[data$dex != 213 & data$hp <=
##     mean(data$hp) + sd(data$hp) * 2.5, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.814  -9.655  -1.544   8.564  63.819
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.34339    2.49922   6.940 1.27e-11 ***
## attack        0.25289    0.02630   9.616  < 2e-16 ***
## defense       0.08269    0.03231   2.559 0.010804 *
## spattack      0.12656    0.02771   4.567 6.28e-06 ***
## spdefense     0.19415    0.03403   5.705 2.03e-08 ***
## ghostTRUE   -12.07666    3.30021  -3.659 0.000281 ***
## groundTRUE    7.74515    2.57034   3.013 0.002720 **
## normalTRUE    9.61770    2.27191   4.233 2.76e-05 ***
## steelTRUE   -12.23566    3.11062  -3.934 9.60e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.27 on 482 degrees of freedom
## Multiple R-squared:  0.5047, Adjusted R-squared:  0.4964
## F-statistic: 61.39 on 8 and 482 DF,  p-value: < 2.2e-16
```

Same variables used on testing data:
Multiple R-squared: 0.3988
Adjusted R-squared: 0.3818

```
## Call:
## lm(formula = hp ~ attack + defense + spattack + spdefense + ghost +
##     ground + normal + steel, data = test[test$hp <= mean(data$hp) +
##     sd(data$hp) * 2.5, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.615  -9.256  -2.061   6.390  59.358
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 25.74911    3.46339   7.435 1.25e-12 ***
## attack       0.19425    0.03549   5.474 9.72e-08 ***
## defense      0.07101    0.04099   1.732  0.08431 .
## spattack     0.14975    0.03707   4.040 6.88e-05 ***
## spdefense    0.13650    0.04916   2.777  0.00586 **
## ghostTRUE   -7.62474    3.87030  -1.970  0.04981 *
## groundTRUE   7.97582    3.68522   2.164  0.03128 *
## normalTRUE   9.58041    2.89738   3.307  0.00107 **
## steelTRUE   -4.99902    4.34311  -1.151  0.25069
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.46 on 283 degrees of freedom
## Multiple R-squared:  0.3988, Adjusted R-squared:  0.3818
## F-statistic: 23.46 on 8 and 283 DF,  p-value: < 2.2e-16
```

# Final Model

BIC Backward Selection model on training data:
Multiple R-squared: 0.5047
Adjusted R-squared: 0.4964

```
## Call:
## lm(formula = hp ~ attack + defense + spattack + spdefense + ghost +
##     ground + normal + steel, data = data[data$dex != 213 & data$hp <=
##     mean(data$hp) + sd(data$hp) * 2.5, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.814  -9.655  -1.544   8.564  63.819
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.34339    2.49922   6.940 1.27e-11 ***
## attack        0.25289    0.02630   9.616  < 2e-16 ***
## defense       0.08269    0.03231   2.559 0.010804 *
## spattack      0.12656    0.02771   4.567 6.28e-06 ***
## spdefense     0.19415    0.03403   5.705 2.03e-08 ***
## ghostTRUE   -12.07666    3.30021  -3.659 0.000281 ***
## groundTRUE    7.74515    2.57034   3.013 0.002720 **
## normalTRUE    9.61770    2.27191   4.233 2.76e-05 ***
## steelTRUE   -12.23566    3.11062  -3.934 9.60e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.27 on 482 degrees of freedom
## Multiple R-squared:  0.5047, Adjusted R-squared:  0.4964
## F-statistic: 61.39 on 8 and 482 DF,  p-value: < 2.2e-16
```

# Interpretation of the coefficients

On average, while holding all else constant:

- A 1 point increase in attack increases the HP by 0.25
- A 1 point increase in defense increases HP by 0.08
- A 1 point increase in special attack increases HP by 0.13
- A 1 point increase in special defense increases HP by 0.19

On average, while holding all else constant:

- Ghost Pokémon have 12.08 less HP than other Pokémon
- Ground Pokémon have 7.75 more HP than other Pokémon
- Normal Pokémon have 9.62 more HP than other Pokémon
- Steel Pokémon have 12.24 less HP than other Pokémon

# R Code Appendix

```
data = read.csv('~/Documents/MAS/402/Project 1/pokemon.csv')

data$generation = as.factor(data$generation)

set.seed(1)

t = sample(1:800, 800)

train = data[t[1:500],]

test = data[t[501:800],]

data = train

hist(data$hp, main = 'Histogram of HP', xlab = 'HP')

boxplot(data$hp, main = 'Boxplot of HP', ylab = 'HP')

qqnorm(data$hp, main = 'Normal Quantile Plot of HP', ylab = 'HP')

hist(data$attack, main = 'Histogram of Attack', xlab = 'Attack')

boxplot(data$attack, main = 'Boxplot of Attack', ylab = 'Attack')

hist(data$spattack, main = 'Histogram of Special Attack', xlab = 'Special Attack')

boxplot(data$spattack, main = 'Boxplot of Special Attack', ylab = 'Special Attack')

hist(data$defense, main = 'Histogram of Defense', xlab = 'Defense')
```