Statistics 101A Project - Group 5: Pokemon Weights

Ryan Chu, Samvit Garg, Nicole Lee, Naren Prakash, Nathaniel Pranoto Tjong, Oliver Siu

Introduction

In our research, we wanted to answer the question "How does a Pokémon's stats in-game predict its "real-life" weight?" We used a dataset called "The Complete Pokemon Dataset" pulled from Kaggle by contributor Rounak Banik. It contains the information of 801 Pokémon from the first 7 generations of the Pokémon games. Originally, the dataset contained 41 variables, but we decided to eliminate the ones that were not relevant to our research. This included variables like the names of Pokémon, their affinities or weaknesses to elements, whether they were legendary or not, etc. After all this, we landed on the following variables: weight_kg, base_egg_steps, attack, defense, hp, sp_attack, sp_defense, capture_rate, experience_growth. We made weight_kg the response variable and the other eight variables its predictors. We then plotted these variables against each other and found a somewhat positive relationship against weight_kg. From this observation, we decided to fit a linear multiple regression model to try and predict weight from our given variables. After testing, we found that we needed to perform a transformation to correctly fit our data. After transformation, we found that our transformed model was valid, and gave us a good fit for the data.

In this paper, we're going to start with our original regression model and why it violated the model assumptions, then go over how we transformed the model and chose the best possible transformation, then how we chose the best predictive model, and finally how we interpreted the results.

Data Description

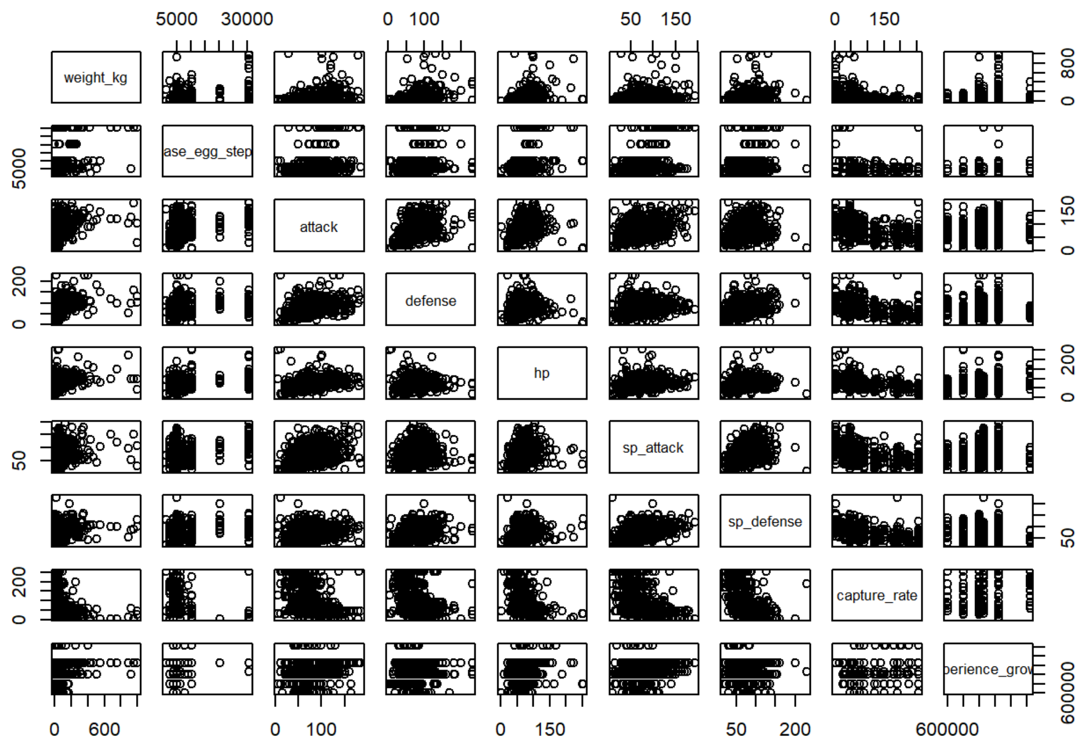
Description of variables:

- weight_kg - weight of the Pokémon (kg)
- base_egg_steps - steps it takes for a Pokémon egg to hatch (distance traveled counted in "steps" where each step is moving on tile in-game)
- attack - base stat that determines the attack value of a Pokémon
- defense - base stat that determines the defense value of a Pokémon
- hp - hit points of a Pokémon (how much damage it can sustain before it faints)
- sp_attack - base stat that determines the special power of your Pokémon
- sp_defense - base stat that determines the special defense of your Pokémon
- capture_rate - base chance of catching a Pokémon
- experience_growth - experience points needed to reach level 100

```
General Summary of Variables:

weight_kg      base_egg_steps      attack           defense            hp
 Min.   :  0.10   Min.   : 1280   Min.   :  5.00   Min.   :  5.00   Min.   :  1.00
 1st Qu.:  9.00   1st Qu.: 5120   1st Qu.: 55.00   1st Qu.: 50.00   1st Qu.: 50.00
 Median : 27.65   Median : 5120   Median : 75.00   Median : 70.00   Median : 65.00
 Mean   : 61.35   Mean   : 7190   Mean   : 77.89   Mean   : 72.99   Mean   : 68.95
 3rd Qu.: 64.35   3rd Qu.: 6400   3rd Qu.:100.00   3rd Qu.: 90.00   3rd Qu.: 80.00
 Max.   :999.90   Max.   :30720   Max.   :185.00   Max.   :230.00   Max.   :255.00
   sp_attack        sp_defense       capture_rate     experience_growth
 Min.   : 10.00   Min.   : 20.00   Min.   :  3.00   Min.   : 600000
 1st Qu.: 45.00   1st Qu.: 50.00   1st Qu.: 45.00   1st Qu.:1000000
 Median : 65.00   Median : 65.50   Median : 60.00   Median :1000000
 Mean   : 71.34   Mean   : 70.90   Mean   : 98.87   Mean   :1055002
 3rd Qu.: 91.75   3rd Qu.: 89.75   3rd Qu.:170.00   3rd Qu.:1059860
 Max.   :194.00   Max.   :230.00   Max.   :255.00   Max.   :1640000
NA's   :20

Standard Deviation:
weight_kg    base_egg_steps    attack        defense                hp
```

```
109.28741     6570.22627        32.21298      30.78979         26.60664
sp_attack     sp_defense        capture_rate  experience_growth
32.17934      27.99371          76.03634      161712.37247

Correlation:

                   weight_kg base_egg_steps     attack    defense         hp  sp_attack
weight_kg          1.0000000      0.4433020  0.3818831  0.4241136  0.4277867  0.2448696
base_egg_steps     0.4433020      1.0000000  0.3273998  0.2735761  0.3619379  0.3960974
attack             0.3818831      0.3273998  1.0000000  0.4684594  0.4082482  0.3671467
defense            0.4241136      0.2735761  0.4684594  1.0000000  0.2390407  0.2516834
hp                 0.4277867      0.3619379  0.4082482  0.2390407  1.0000000  0.3615414
sp_attack          0.2448696      0.3960974  0.3671467  0.2516834  0.3615414  1.0000000
sp_defense         0.3065837      0.3300221  0.2599450  0.5374384  0.3615289  0.5041236
capture_rate      -0.3266001     -0.3542185 -0.4920778 -0.4486342 -0.4686823 -0.5098808
experience_growth 0.2460957       0.3713009  0.2384658  0.1297488  0.2123994  0.1941638
                   sp_defense capture_rate experience_growth
weight_kg          0.30658367   -0.3266001        0.24609566
base_egg_steps     0.33002211   -0.3542185        0.37130089
attack             0.25994505   -0.4920778        0.23846579
defense            0.53743844   -0.4486342        0.12974882
hp                 0.36152892   -0.4686823        0.21239944
sp_attack          0.50412361   -0.5098808        0.19416376
sp_defense         1.00000000   -0.5148140        0.08542536
capture_rate      -0.51481397    1.0000000       -0.16979354
experience_growth  0.08542536   -0.1697935        1.00000000
```

Some correlations between predictors are a bit high, reaching 0.5 and higher.



Looking at the summary statistics and pair plots, we can see most of the variables are right-skewed. Density plots are included in the appendix for further analysis ("Original Data Density Plots").

Taking the linear model of all our untransformed variables, R gives us the following:

```
##
## Call:
## lm(formula = weight_kg ~ capture_rate + base_egg_steps + attack +
##     defense + hp + sp_attack + sp_defense + capture_rate + experience_growth,
##     data = pokemon_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -255.83  -31.79   -5.96   18.68  814.30
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.680e+02  2.785e+01  -6.034 2.48e-09 ***
## capture_rate       5.073e-02  5.705e-02   0.889   0.3742
## base_egg_steps     4.291e-03  5.738e-04   7.478 2.05e-13 ***
## attack             2.868e-01  1.267e-01   2.263   0.0239 *
## defense            9.785e-01  1.359e-01   7.201 1.42e-12 ***
## hp                 1.051e+00  1.427e-01   7.367 4.47e-13 ***
## sp_attack         -1.314e-01  1.266e-01  -1.038   0.2996
## sp_defense        -3.199e-02  1.581e-01  -0.202   0.8397
## experience_growth  3.664e-05  2.121e-05   1.728   0.0844 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87.68 on 773 degrees of freedom
##   (20 observations deleted due to missingness)
## Multiple R-squared:  0.3629, Adjusted R-squared:  0.3563
## F-statistic: 55.04 on 8 and 773 DF,  p-value: < 2.2e-16
```

From this summary, we can see that capture rate, sp_attack, and sp_defense are not significant predictors. The adjusted R-squared is also rather low, with a value of 0.3563. When looking at the residual plots of the full model (Appendix “Original Model Diagnostic Plots”), some problems immediately stood out. The residual plot had a fan shape indicating heterogeneous variance, the Q-Q residuals didn’t follow a 45-degree line at all, violating the normality of errors assumption, and there were some clear bad leverage points. This led us to believe our model required some sort of transformation.

Keeping this in mind, two methods of transformation were utilized to provide alternative models. The first method was using the box-cox method to jointly normalize the predictor variables while performing an inverse response plot to transform the response variable. Using this transformation (Appendix “Inverse Response Results”), the diagnostic plots of this model (Appendix “Inverse Response Model Diagnostic Plots”) show noticeable improvement in both residual plots, the Q-Q plot, and the residuals vs leverage plot. The second method chosen for an alternative model was a box-cox transformation of the response variable and the predictor variables together. Using the results of this transformation (Appendix “Box-Cox Transformation Results”), we once more observe noticeable improvements in both residual plots, the Q-Q plot, and the residuals vs leverage plot. Both of these proposed models are very similar in adjusted R^2, F statistics, and significance of predictor variables. Ultimately, the box-cox transformation was chosen due to a better Q-Q plot (indicating a more normal distribution of the error term). This is the model we will refer to as the transformed model.

VIFs of Box-Cox Transformation

```
   t_bes        t_attack        t_defense             t_hp
1.469546        1.828095         1.910672         1.575185
    t_sa            t_sd             t_cp experience_growth
1.755954        2.225710         2.304678         1.190213
```

When taking the VIFs of the full transformed model, we found that none of the VIFs were over 5. However, when looking at the added-variable plots (see Appendix "Box-Cox Transformation Added-Variable Plots") we can see that the transformed special attack and special defense seem to not have an impact, and attack and base egg steps seem to have very little impact.

Variable selection was deemed worthwhile for investigation, so tests were performed to compare models using all possible subsets, backward AIC & BIC stepwise regression, and forward AIC & BIC stepwise regression. From all possible subsets, the best adjusted R-squared, AIC, and AICc model was the full transformed model. From all possible subsets, the best BIC model was a reduced model with 4 predictors. From stepwise regression, the full transformed model was chosen from Backwards AIC & Forwards AIC selection, and the reduced model with 4 predictors was chosen from Backwards BIC & Forwards BIC selection.

Backwards AIC

```
## Start:  AIC=-1895.3
## t_weight ~ t_bes + t_attack + t_defense + t_hp + t_sa + t_sd +
##     t_cp + experience_growth
##
##                     Df Sum of Sq    RSS     AIC
## <none>                           67.706 -1895.3
## - t_cp               1    0.2836 67.990 -1894.0
## - t_bes              1    0.3034 68.010 -1893.8
## - t_sa               1    0.3277 68.034 -1893.5
## - t_sd               1    0.4135 68.120 -1892.5
## - experience_growth  1    0.5518 68.258 -1891.0
## - t_attack           1    2.0724 69.779 -1873.7
## - t_defense          1    3.2823 70.989 -1860.3
## - t_hp               1    8.2417 75.948 -1807.5
```

Backwards BIC

```
## Step:  AIC=-1864.04
## t_weight ~ t_bes + t_attack + t_defense + t_hp
##
##             Df Sum of Sq    RSS     AIC
## <none>                   69.091 -1864.0
## - t_bes      1    1.1165 70.207 -1858.2
## - t_attack   1    2.2128 71.303 -1846.1
## - t_defense  1    6.8326 75.923 -1797.0
## - t_hp       1   11.9179 81.009 -1746.3
```

Forwards AIC

```
## Step:  AIC=-1895.3
## t_weight ~ t_hp + t_defense + t_attack + t_bes + experience_growth +
##     t_sd + t_sa + t_cp
```

Forwards BIC

```
## Step:  AIC=-1864.04
## t_weight ~ t_hp + t_defense + t_attack + t_bes
##
##                     Df Sum of Sq    RSS     AIC
## <none>                           69.091 -1864.0
## + experience_growth  1   0.45888 68.632 -1862.6
## + t_cp               1   0.39669 68.694 -1861.8
## + t_sd               1   0.32795 68.763 -1861.1
## + t_sa               1   0.02015 69.070 -1857.6
```

Therefore, our 2 candidate models from variable selection were:

1. Reduced model (4 predictors)
2. Full transformed model (8 predictors)

A partial F-test via ANOVA test was employed to compare the two models, giving p-value = 0.003502. Since the p-value of the partial F-test was less than 0.05, we reject the null hypothesis, and there is sufficient evidence against the reduced model in favor of the full model. Therefore, we still chose the full transformed model.

Partial F-Test

```
## Analysis of Variance Table
##
## Model 1: t_weight ~ t_bes + t_attack + t_defense + t_hp
## Model 2: t_weight ~ t_bes + t_attack + t_defense + t_hp + t_sa + t_sd +
##     t_cp + experience_growth
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    777 69.091
## 2    773 67.706  4    1.3843 3.9513 0.003502 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Therefore, we have our final model:

$$\begin{aligned} \text{weight}^{0.16} = & -0.71529 - 27.3651527 * \text{base egg steps}^{-0.67} + 0.0373754 * \text{attack}^{0.5} + 0.1586544 * \text{defense}^{0.33} \\ & + 0.2643399 * \text{hp}^{0.33} - 0.0444752 * \text{sp\_attack}^{0.33} + 0.3048735 * \text{sp\_defense}^{0.15} \\ & - 0.0230343 * \text{capture\_rate}^{0.33} + 1.7932379 \times 10^{-7} * \text{experience\_growth} \end{aligned}$$

```
## Call:
## lm(formula = t_weight ~ t_bes + t_attack + t_defense + t_hp +
##     t_sa + t_sd + t_cp + experience_growth)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29697 -0.14664  0.00619  0.17558  1.21196
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -7.153e-01  2.815e-01  -2.541   0.0112 *
## t_bes             -2.737e+01  1.470e+01  -1.861   0.0631 .
## t_attack           3.738e-02  7.684e-03   4.864 1.39e-06 ***
## t_defense          1.587e-01  2.592e-02   6.122 1.47e-09 ***
## t_hp               2.643e-01  2.725e-02   9.700  < 2e-16 ***
## t_sa              -4.448e-02  2.299e-02  -1.934   0.0534 .
## t_sd               3.049e-01  1.403e-01   2.173   0.0301 *
## t_cp              -2.303e-02  1.280e-02  -1.799   0.0724 .
## experience_growth  1.793e-07  7.144e-08   2.510   0.0123 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.296 on 773 degrees of freedom
##   (20 observations deleted due to missingness)
## Multiple R-squared:  0.4637, Adjusted R-squared:  0.4581
## F-statistic: 83.54 on 8 and 773 DF,  p-value: < 2.2e-16
```

Since the response variables and the predictors were transformed independently with different powers, a direct interpretation of each coefficient is not convenient. However, we can still interpret their general influence on the weight of a Pokémon, i.e. their positive or negative influence on weight:

- base_egg_steps has a negative influence on weight
- attack has a positive influence on weight
- defense has a positive influence on weight
- hp has a positive influence on weight
- sp_attack has a negative influence on weight
- sp_defense has a positive influence on weight
- capture_rate has a negative influence on weight
- experience_growth has a positive influence on weight

These influences for these slopes were verified via the summary statistics and added-variable plots (see Appendix "Box-Cox Transformation Added-Variable Plots"), and they mostly make sense in the context of the Pokémon games. Increased stats of a Pokémon generally lead to a heavier Pokémon. While the negative influence of sp_attack on weight may seem curious, heavier Pokémon have increased base attack stats, so they utilize their base attack stats more and rely less on their sp_attack. capture_rate having a negative influence on weight also makes sense as a heavier Pokémon has more mass and is more difficult to capture. base_egg_steps having a negative influence on weight indicates that heavier Pokémon usually hatch faster. This is the best model we could develop to predict the weight of a Pokémon.

Discussion

In summary, our project used various base stats built into the Pokémon games to predict a physical characteristic of the Pokémon, in this case, weight. After testing several transformed and altered models, we found a model that seems to do the best job showcasing the relationship between the selected base stats and the weight of a given Pokémon. We found that in general, stronger Pokémon are physically heavier as well. Because of the significance of the slopes, there is evidence to suggest that this is a deliberate decision on the part of the developers.

There are certain limitations to the scope of this data. Firstly, this data was only representative of the first seven generations of Pokémon (out of a current nine). This means the more recent changes in game direction may not be represented in this analysis. Second, this is only one game. While Pokémon has been around for decades and has inspired many other developers, trends in Pokémon may not be representative of the gaming industry as a whole, and certainly not real-life animals.

This study could hence be improved in the future with some changes. Data from other games as well as from developers from various global regions could be considered. More recent data could also be brought in. Finally, data resembling more diverse trends would greatly increase the scope of the analysis (could focus on specific representations of marginalized groups).

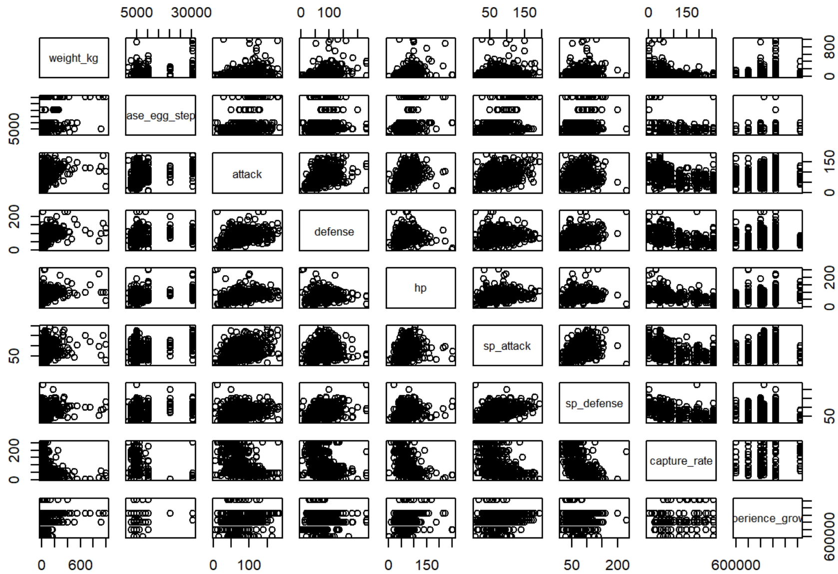
With this in mind, we will discuss some possible implications seen throughout the gaming industry. In fictional settings, developers may tend to mimic the perception that physically bigger entities are stronger in the games they develop. This graphical design choice makes sense, but it is still interesting to note. These fictional settings often still fit our physical perceptions of reality, which may make them easier to understand, but are still a bound on our creativity.

More broadly, in other contexts, perpetuating real-life perceptions can serve to make games less inclusive. Stereotypes and biases held by developers clearly can make their way into the games they produce. It is important to recognize these biases in the media. These details are genuinely critical to the interpretation. While Pokémon is not intended to do more than provide an immersive journey, and the strength scale of the Pokémon makes it easy to understand which are strong, we can still recognize the fallacious "the strong survive" mindset it implies. Similarly, in other media, while they may succeed in entertaining and providing positive messaging, we can still point out their representations and treatment of minority and marginalized groups.

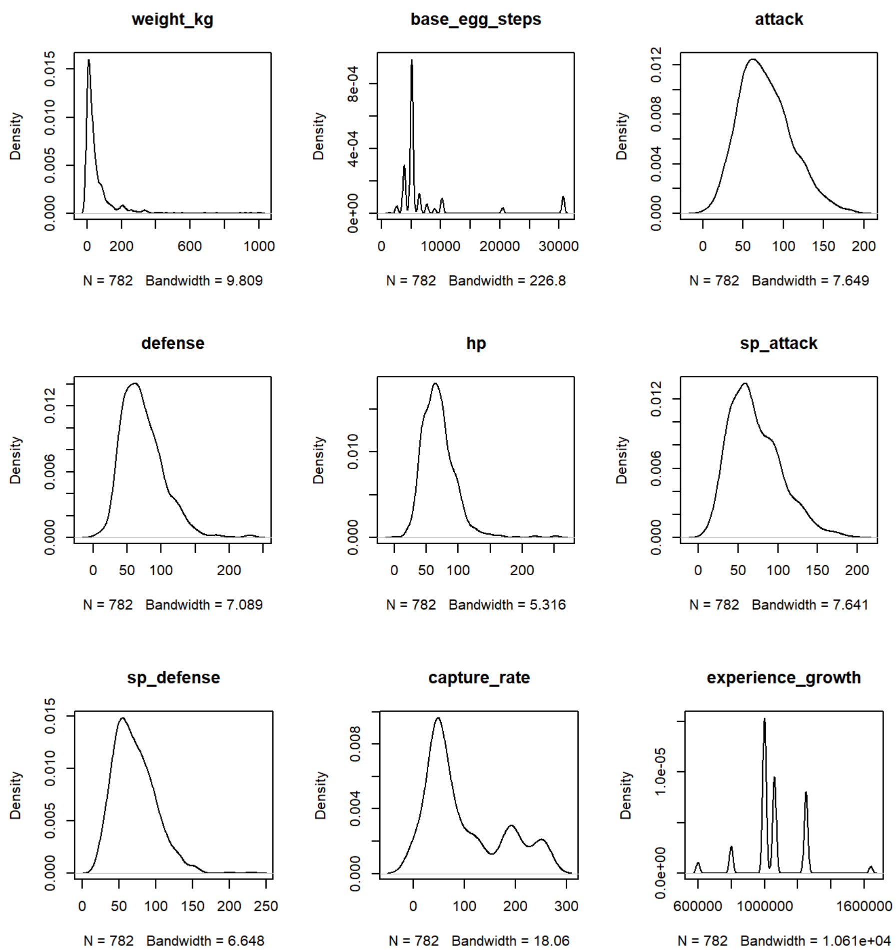
Overall, analyses like this can be used to more specifically study such trends in media. It is important to do so, as more inclusive media is crucial to the natural development of a more inclusive society.

Appendix

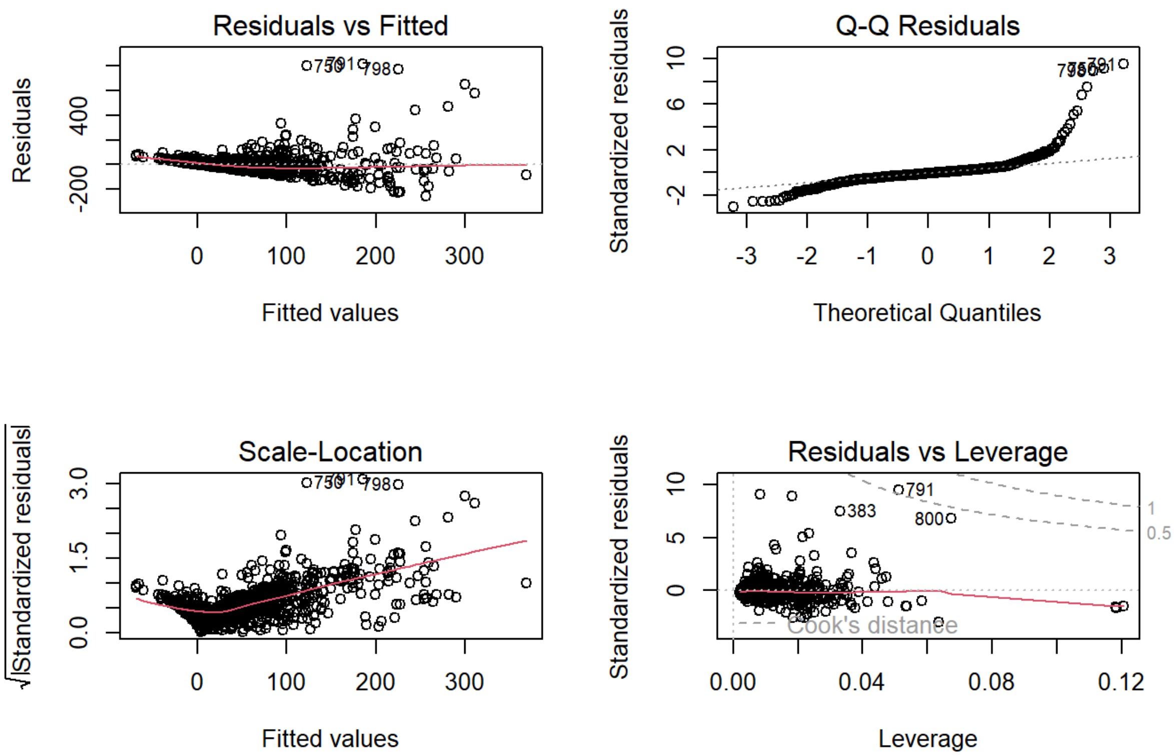
Original Data Pair Plots



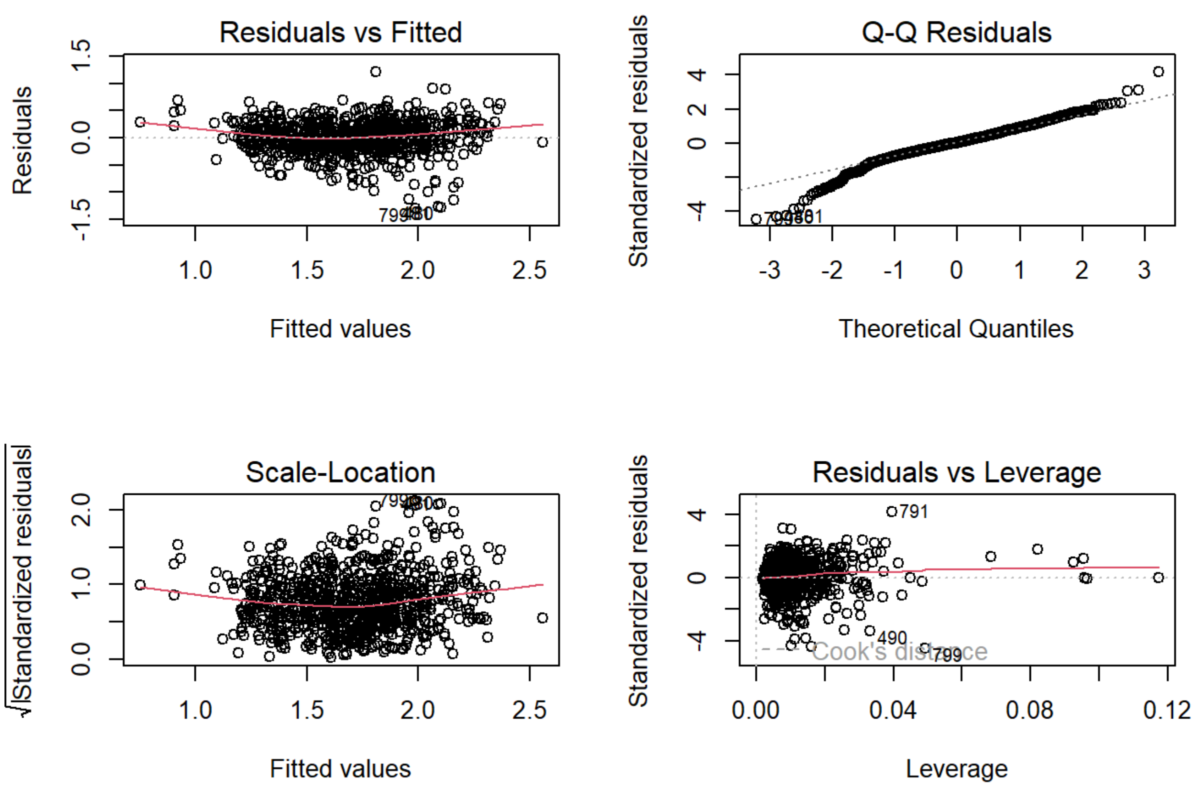
Original Data Density Plots



Original Model Diagnostic Plots

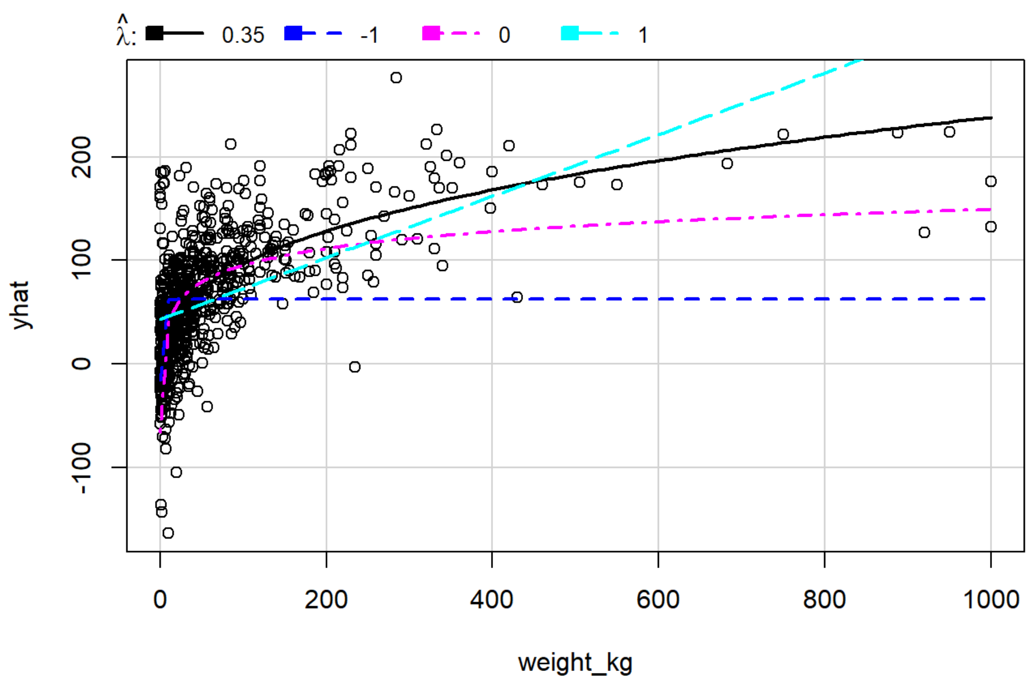


Full Box-Cox Transformation Diagnostic Plots



Inverse Response Results

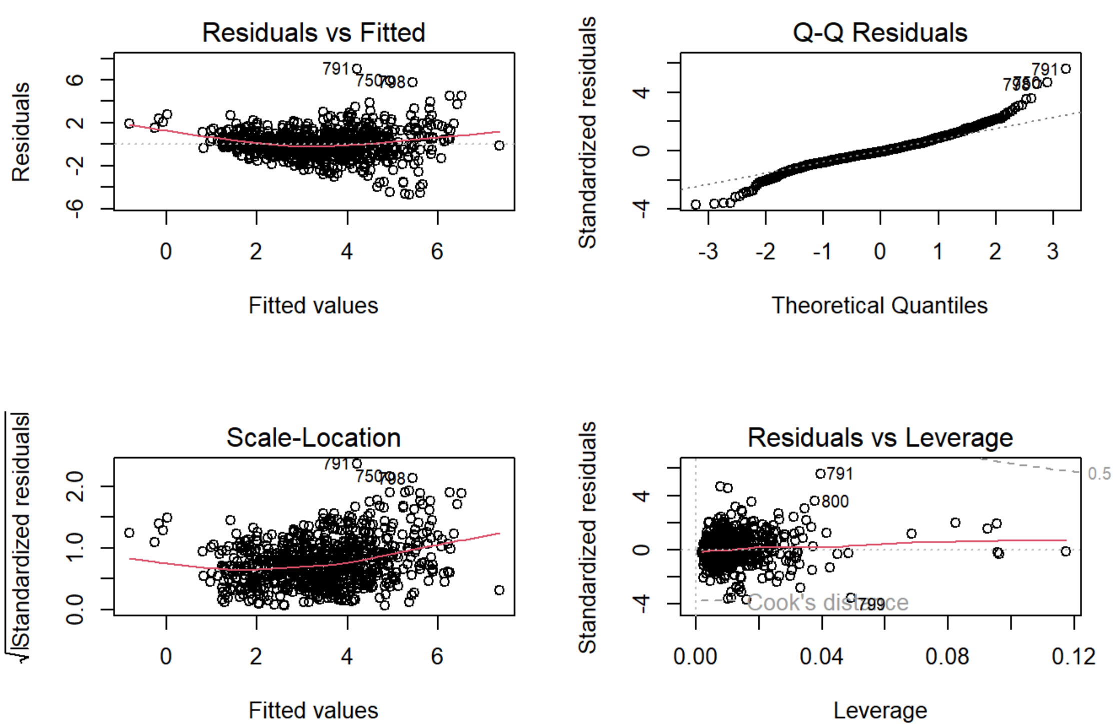
```
## bcPower Transformations to Multinormality
##                    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## base_egg_steps       -0.6725       -0.67      -0.7544      -0.5905
## attack                0.5034        0.50       0.3978       0.6089
## defense               0.2889        0.33       0.1973       0.3804
## hp                    0.3133        0.33       0.2382       0.3884
## sp_attack             0.3380        0.33       0.2306       0.4453
## sp_defense            0.1412        0.14       0.0186       0.2639
## capture_rate          0.3519        0.33       0.3017       0.4022
## experience_growth     0.8826        1.00       0.6416       1.1235
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                        LRT df       pval
## LR test, lambda = (0 0 0 0 0 0 0 0) 748.9334  8 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                        LRT df       pval
## LR test, lambda = (1 1 1 1 1 1 1 1) 2641.379  8 < 2.22e-16
```




```
##        lambda     RSS
## 1   0.3510711 1531643
## 2 -1.0000000 2741779
## 3   0.0000000 1758910
## 4   1.0000000 1951684
```

```
summary(t2_pokemon_model)
Call:
lm(formula = t2_weight ~ t2_bes + t2_attack + t2_defense + t2_hp +
    t2_sa + t2_sd + t2_cp + experience_growth)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7010 -0.6883 -0.1001  0.6250  7.0203

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -6.637e+00  1.278e+00  -5.193 2.65e-07 ***
t2_bes            -2.105e+02  6.368e+01  -3.305 0.000993 ***
t2_attack          1.392e-01  3.327e-02   4.184 3.19e-05 ***
t2_defense         7.719e-01  1.122e-01   6.877 1.26e-11 ***
t2_hp              1.142e+00  1.180e-01   9.674  < 2e-16 ***
t2_sa             -1.890e-01  9.960e-02  -1.897 0.058186 .
t2_sd              1.155e+00  6.790e-01   1.701 0.089413 .
t2_cp             -9.349e-02  5.544e-02  -1.686 0.092163 .
experience_growth  8.836e-07  3.094e-07   2.856 0.004410 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.282 on 773 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared:  0.4748,   Adjusted R-squared:  0.4694
F-statistic: 87.36 on 8 and 773 DF,  p-value: < 2.2e-16
```

Inverse Response Model Diagnostic Plots



Box-Cox Transformation Results

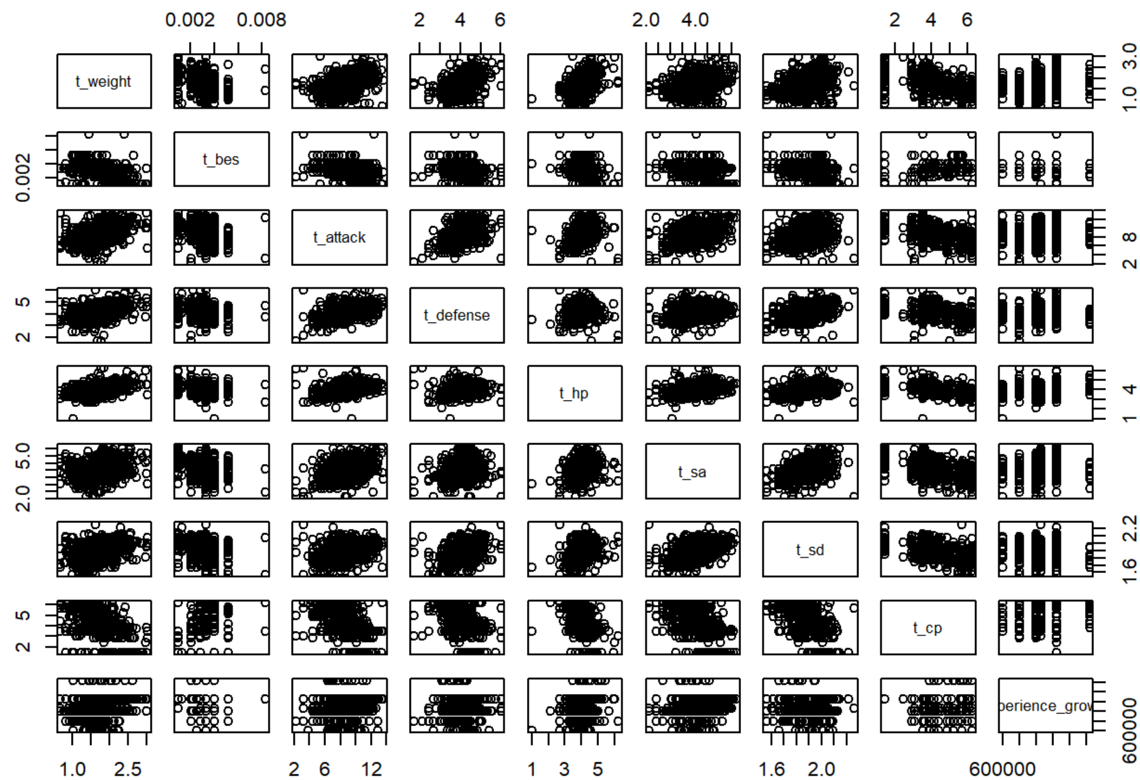
```
summary(pokemon_full_model_pt)
## bcPower Transformations to Multinormality
##                    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## weight_kg             0.1561        0.16       0.1241       0.1882
## base_egg_steps       -0.6690       -0.67      -0.7515      -0.5864
## attack                0.5202        0.50       0.4142       0.6261
## defense               0.3161        0.33       0.2241       0.4080
## hp                    0.3195        0.33       0.2447       0.3943
## sp_attack             0.3243        0.33       0.2132       0.4354
## sp_defense            0.1471        0.15       0.0234       0.2708
## capture_rate          0.3507        0.33       0.2999       0.4015
## experience_growth     0.8937        1.00       0.6504       1.1370
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                       LRT df       pval
## LR test, lambda = (0 0 0 0 0 0 0 0 0) 836.6689  9 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                       LRT df       pval
## LR test, lambda = (1 1 1 1 1 1 1 1 1) 4490.133  9 < 2.22e-16
```

```
attach(pokemon_subset)
t_weight <- weight_kg^0.16
t_bes <- base_egg_steps^-0.67
t_attack <- attack^0.5
t_defense <- defense^0.33
t_hp <- hp^0.33
t_sa <- sp_attack^0.33
t_sd <- sp_defense^0.15
t_cp <- capture_rate^0.33
t_pokemon_model <- lm(t_weight ~ t_bes + t_attack + t_defense + t_hp +
                        t_sa + t_sd + t_cp + experience_growth)
summary(t_pokemon_model)
```

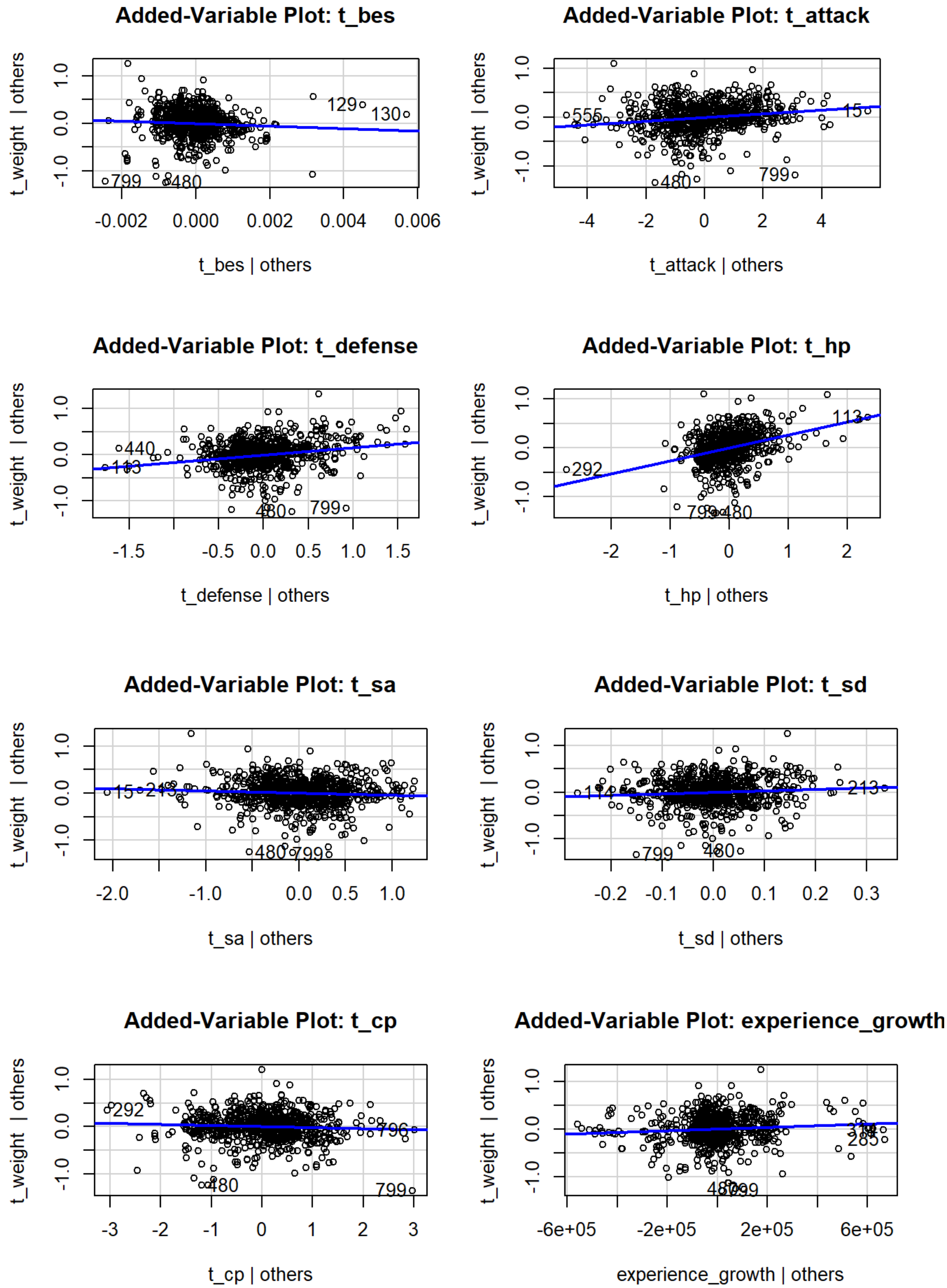
```
Call:
lm(formula = t_weight ~ t_bes + t_attack + t_defense + t_hp +
    t_sa + t_sd + t_cp + experience_growth)

Residuals:
     Min       1Q   Median       3Q      Max
-1.29697 -0.14664  0.00619  0.17558  1.21196

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -7.153e-01  2.815e-01  -2.541   0.0112 *
t_bes             -2.737e+01  1.470e+01  -1.861   0.0631 .
t_attack           3.738e-02  7.684e-03   4.864 1.39e-06 ***
t_defense          1.587e-01  2.592e-02   6.122 1.47e-09 ***
t_hp               2.643e-01  2.725e-02   9.700  < 2e-16 ***
t_sa              -4.448e-02  2.299e-02  -1.934   0.0534 .
```

```
t_sd                3.049e-01  1.403e-01   2.173   0.0301 *
t_cp               -2.303e-02  1.280e-02  -1.799   0.0724 .
experience_growth   1.793e-07  7.144e-08   2.510   0.0123 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.296 on 773 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared:  0.4637,   Adjusted R-squared:  0.4581
F-statistic: 83.54 on 8 and 773 DF,  p-value: < 2.2e-16
```

Box-Cox Transformation Pair Plots



Box-Cox Transformation Added-Variable Plots



Reduced Model Diagnostic Plots (4 predictor)