December 10, 2021

Predicting The Height and Weight Of A Pokemon

Multiple Linear Regression and Using The SignifReg Package for Model Building

Chance Vandergeugten
Marguerite Mourad
Leo Jacinto Magtibay

We examine the predictability of some key variables in the Pokemon data set. We identify and build models from the data set that showcase Multiple Linear Regression in action and using SignifReg package to build the models from the ground up while keeping the fundamental assumptions of Linear Regression intact. We further show that the Multiple Linear Regression might be useful, but this infact is not enough to keep the assumptions of Linear Regression, thus we apply Boxcox transformation to the full model. By using the Boxcox transformation and building the model from the new full model we highlight the improvements that the Boxcox transformation has done and provide exposition of the viability of the models that comes from this approach. We thus show the effectiveness of Boxcox transformation as a supplement for algorithmic model building using the SignifReg package.

1 Introduction

Using the Pokemon data set, we built multiple linear regression models in order to predict our response variables, height and weight (with metrics in meters and kilograms, respectively). The PokeDex is a website that revolves around a Pokemon's strengths, weaknesses as as well as their overall stats and types (if they are legendary or not) [1]. This website was used to help with imputations due to missing data that occurred within our data set. The predictors that we will be utilizing are each individual Pokemon's abilities which corresponds to their varied characteristics obtained from the data set together with the information obtained from PokeDex.
Throughout this report, we will employ the packages in R, such as the SignifReg package to build and compare models according to the normality assumptions, which is homoscedasticity along with a normal distribution. The SignifReg package is a package created by Zambom and Kim with the use of choosing a direction, which is stepwise and forward in our models, and choosing a criterion, which is Adjusted-R^2 [2]. In addition, the data analysed in this report stems from the diagnostics which the models produced when using the full and reduced model as well as the models with transformations through the plots, Normal Q-Q, Residuals vs Fitted, Scale-Location and Residuals vs Leverage. Further, for our transformations, we will be applying the techniques of Box-cox transformations on our regression models to observe the impact it has on our response variables (in different models) and our predictors, while also keeping in mind of our assumptions. These cumulatively work to describe the overall results which we are attempting to achieve without violating the aforementioned normality assumptions which correlates with Multiple Linear Regression. Additionally, we will also be observing the summaries which R will produce in order to find significance among our response variables.
Overall, with the use of the SignifReg package as well as the other packages in R, such as the cars package which has the variance inflation factor, we will determine the model that best describes the relationship of a Pokemon's height and weight individually in correspondence of their attributes illustrated in the Pokemon data set.

2 Methodology

Model 1 (Predicting Weight)

In our first model, we used a multiple linear regression approach to predict a Pokemon's weight from other predictors in the data set. Using the Signifreg package, a model was constructed starting from a null model with the direction specified being "both". This is a ground up approach with a step-wise algorithm. The criterion selected was "r-adj". The resulting model took the form

$$weight\_kg \sim \beta_0 + \beta_1 base\_total + \beta_2 base\_happiness + \beta_3 speed + \beta_4 hp + \beta_5 against\_poison$$

$$+\beta_6 is_legendary + \beta_7 sp\_attack + \beta_8 against\_water$$

A Boxcox transformation was then applied to our response variable (weight_kg) in order to correct some diagnostics and improve the r-adj value. This would result in our final model to predict weight_kg. The resulting model took the form

$$(weight\_kg^{\lambda}-1)/\lambda \sim \beta_0+\beta_1 base\_total+\beta_2 base\_happiness+\beta_3 speed+\beta_4 hp+\beta_5 against\_poison$$
$$+\beta_6 is_legendary + \beta_7 sp\_attack + \beta_8 against\_water$$

$\lambda = 0.1818182$

The results of this model are later discussed in the analysis portion of the report.

Model 2 (Predicting Height)

Using the same methods as Model 1 we've constructed models that would predict the height of a Pokemon. Using the SignifReg function we've built a model from the ground up. As with Model 1 we've used the same criterion, Adjusted R^2. First, we ran stepwise selection of the predictors, producing the model:

$$height_m \sim \beta_0+\beta\_1 weight\_kg+\beta_2 base\_total+\beta_3 hp+\beta_4 generation+\beta_5 base\_happiness$$
$$+\beta_6 type2+\beta_7 against\_electric+\beta_8 against\_psychic+\beta_9 sp\_defense+\beta_{10} is\_legendary$$
$$+\beta_{11} experience\_growth + \beta_{12} against\_ice + \beta_{13} against\_grass$$

We also built a model using forward selection which produced the following model:

$$height_m \sim \beta_0+\beta\_1 weight\_kg+\beta_2 base\_total+\beta_3 hp+\beta_4 generation+\beta_5 base\_happiness$$
$$+\beta_6 type2+\beta_7 against\_electric+\beta_8 against\_psychic+\beta_9 sp\_defense+\beta_{10} is\_legendary$$
$$+\beta_{11} experience\_growth + \beta_{12} against\_ice + \beta_{13} against\_grass$$

By running the SignifReg function with both stepwise and forward directions, we were able to obtain a comparative view of the models and check the diagnostics of each. We have discovered that the stewise and forward ended up with exactly the same model, which we will go in detail in Data Analysis section. As with Model 1 we've taken the liberty to improve the diagnostics of Model 2 by using a Boxcox transformation. By using the same methods as before, we then include the Boxcox transformation to the model, such that height_m$^{\lambda}$ where $\lambda = 0.1414141$ which was attained using the Boxcox function in R.

$$(height_m^{\lambda}-1)/\lambda \sim \beta_0+\beta\_1 weight\_kg+\beta_2 base\_total+\beta_3 hp+\beta_4 generation+\beta_5 base\_happiness$$
$$+\beta_6 type2+\beta_7 against\_electric+\beta_8 against\_psychic+\beta_9 sp\_defense+\beta_{10} is\_legendary$$
$$+\beta_{11} experience\_growth + \beta_{12} against\_ice + \beta_{13} against\_grass$$

3 Data Analysis

Model 1 (Predicting Weight)

After our model for predicting weight was constructed, we obtained the following summary from R:

```
Call:
lm(formula = (weight_kg^lambda - 1)/lambda ~ base_total + base_happiness +
    speed + hp + against_poison + is_legendary + sp_attack +
    against_water)

Residuals:
    Min      1Q  Median      3Q     Max
-9.2779 -0.9389  0.0680  1.1076  7.9498

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.867022   0.432409   2.005 0.045292 *
base_total      0.015589   0.001157  13.478  < 2e-16 ***
base_happiness -0.030654   0.003724  -8.232 7.56e-16 ***
speed          -0.018254   0.002834  -6.441 2.06e-10 ***
hp              0.020295   0.003345   6.068 2.01e-09 ***
against_poison -0.515880   0.133468  -3.865 0.000120 ***
is_legendary   -0.954047   0.282701  -3.375 0.000775 ***
sp_attack      -0.014056   0.003132  -4.488 8.25e-06 ***
against_water   0.439130   0.117977   3.722 0.000211 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.848 on 792 degrees of freedom
Multiple R-squared:  0.5345,    Adjusted R-squared:  0.5298
F-statistic: 113.7 on 8 and 792 DF,  p-value: < 2.2e-16
```

Figure 1: Weight Model Summary

From the summary, we were able to obtain an adjusted-R^2 value of 0.5298. This means that our model is able to explain 52.89% of the variability in a Pokemon's weight (in kilograms) from the 8 predictors. Over 50% explanation of the response variable is quite desirable and we were pleased with the results from our model.

Taking a look at the predictors, we noticed that all of the predictors were found to be significant with a p-value less than .001. Although we can contribute the significance of the predictors to the criterion selection in the SignifReg package, it is important to note that a p-value less than .001 marks very strong evidence in favor of rejecting the assumption that each $\beta_n = 0$. This suggests that the method used to build this model was effective at finding the most valuable explanatory variables associated with a Pokemon's weight. An interesting observation is that the most significant predictor in this model was "base_total" which is the total of a Pokemon's base stats (strength, defense, speed, etc). Although this gave us the most explanation for a Pokemon's weight, our model

still included some of the base stats on their own (speed, hp, sp_attack). Because these base stats are included in the "base_total" predictor, we checked the VIF to ensure that multicollinearity was not an issue:

```
> vif(bc_m1)
    base_total base_happiness          speed             hp against_poison   is_legendary
      4.451091       1.247195       1.571697       1.850303       1.258990       1.494428
     sp_attack  against_water
      2.404283       1.199146
```

Figure 2: Weight Model VIFs

In order to be happy with our final model, we had to make sure that the residuals were in alignment with our assumptions (residuals are homoscedastic and normally distributed). Before applying the Boxcox transformation, the residual plot was less than stellar:

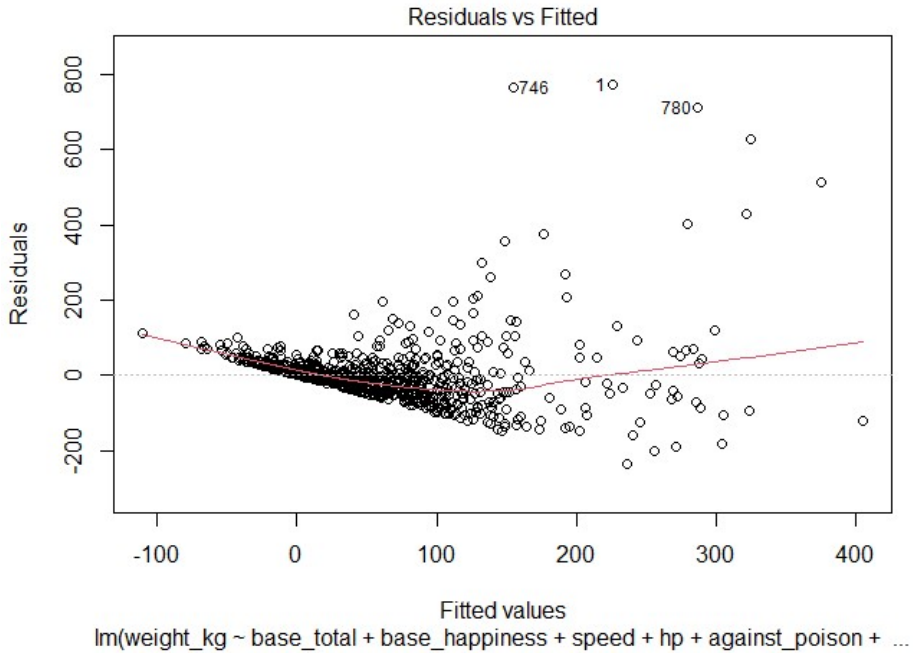

Figure 3: Untransformed Residual Plot

This residual plot shows a noticeable quadratic pattern in the residuals as well as an increase in variance as our fitted values increase (heteroscedastic). This kind of residual plot suggests that a new model must be built or that transformations must be applied to the variables of the model.

We also checked the Normal QQ plot to see if the residuals were approximately normally distributed:

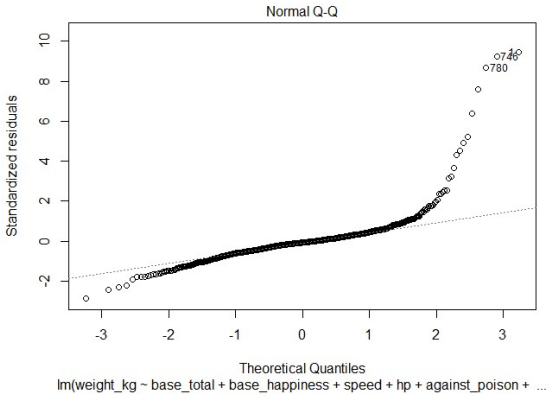

Figure 4: Untransformed Normal QQ Plot

Most of the QQ plot seemed to fit well except for the right tail which had residuals much larger than expected. For these reasons we decided to apply a Boxcox transformation to our response variable. The transformation helped us achieve residuals that adhered closer to our assumptions in order to have a valid model:

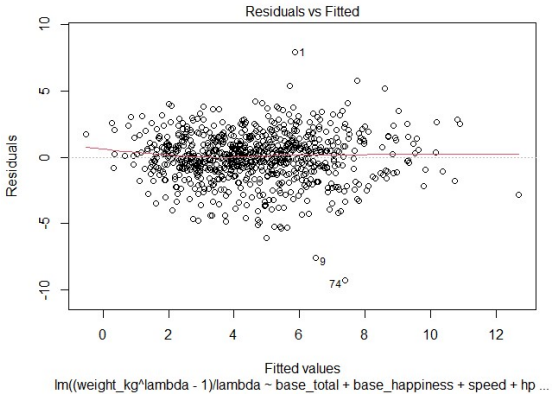

Figure 5: Transformed Residual Plot

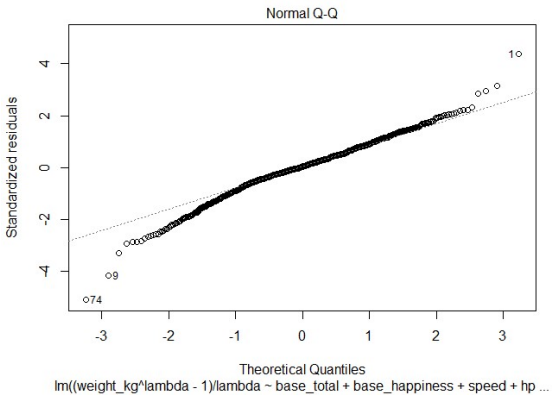

Figure 6: Transformed Normal QQ Plot

After looking at the transformed residuals, we concluded that the model was now ready to be used for predicting a Pokemon's weight.

Model 2 (Predicting Height)

As we've covered in the methodology, we've built a model from the ground up using the SignifReg function. By applying the summary function to the first model we obtained an adjusted-R^2 of 0.5061. Meaning that this model is able to explain 50.61% of the variability for a Pokemon's height (in meters) from the 13 predictors. (7)

```
Call:
lm(formula = height_m ~ weight_kg + base_total + hp + generation +
    base_happiness + type2 + against_electric + against_psychic +
    sp_defense + is_legendary + experience_growth + against_ice +
    against_grass)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7427 -0.2525 -0.0574  0.1675 10.8331

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -5.193e-01  2.936e-01  -1.769 0.077328 .
weight_kg          4.262e-03  3.163e-04  13.476  < 2e-16 ***
base_total         2.628e-03  4.225e-04   6.222 8.07e-10 ***
hp                 5.509e-03  1.359e-03   4.055 5.53e-05 ***
generation        -4.954e-02  1.483e-02  -3.340 0.000877 ***
base_happiness    -6.172e-03  1.641e-03  -3.761 0.000182 ***
type2bug           2.239e-01  3.419e-01   0.655 0.512834
type2dark          1.339e-02  1.816e-01   0.074 0.941222
type2dragon        2.380e-01  2.034e-01   1.170 0.242208
type2electric     -1.212e-01  2.569e-01  -0.472 0.637235
type2fairy        -4.588e-02  1.485e-01  -0.309 0.757508
type2fighting     -1.681e-01  1.739e-01  -0.967 0.333932
type2fire         -3.446e-02  2.169e-01  -0.159 0.873806
type2flying       -1.312e-01  1.595e-01  -0.823 0.410895
type2ghost         3.383e-01  2.076e-01   1.630 0.103557
type2grass         1.853e-01  1.786e-01   1.038 0.299804
type2ground        4.700e-01  1.726e-01   2.723 0.006615 **
type2ice           3.696e-02  2.007e-01   0.184 0.853907
type2normal        1.565e-01  3.816e-01   0.410 0.681908
type2poison       -8.357e-02  1.570e-01  -0.532 0.594709
type2psychic      -1.004e-01  1.523e-01  -0.659 0.510209
type2rock         -2.726e-01  2.170e-01  -1.256 0.209349
type2steel        -2.064e-01  1.769e-01  -1.167 0.243732
type2water        -1.404e-01  1.990e-01  -0.706 0.480692
against_electric   2.278e-01  7.674e-02   2.968 0.003092 **
against_psychic    1.192e-01  7.451e-02   1.599 0.110199
sp_defense        -2.035e-03  1.438e-03  -1.415 0.157333
is_legendary      -1.937e-01  1.207e-01  -1.605 0.108874
experience_growth  2.446e-07  1.875e-07   1.305 0.192427
against_ice        7.776e-02  6.312e-02   1.232 0.218296
against_grass     -5.171e-02  4.775e-02  -1.083 0.279199
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.752 on 770 degrees of freedom
Multiple R-squared:  0.5246,    Adjusted R-squared:  0.5061
F-statistic: 28.32 on 30 and 770 DF,  p-value: < 2.2e-16
```

Figure 7: Height Model Summary

Analyzing the predictors, there are a lot of insignificant predictors that are caused by the dummy variables, mainly, attributes of type 2. Although, there are some significant predictors which are weight_kg, base_total, generation, and

hp. To check for multicollinearity we apply the vif() function to the model, as we've discovered there is a level of multicollinearity with variable type2. This is not surprising as it consists of multiple parts, as it is a dummy variable which can have dependecies within that variable (8).

	GVIF	Df	GVIF^(1/(2*Df))
weight_kg	1.666503	1	1.290931
base_total	3.587816	1	1.894153
hp	1.844631	1	1.358172
generation	1.159752	1	1.076918
base_happiness	1.463788	1	1.209871
type2	22.736268	18	1.090653
against_electric	3.574088	1	1.890526
against_psychic	1.925939	1	1.387782
sp_defense	2.283432	1	1.511103
is_legendary	1.645449	1	1.282751
experience_growth	1.277726	1	1.130365
against_ice	3.047598	1	1.745737
against_grass	2.007451	1	1.416846

Figure 8: Height Model VIFs

Studying the diagnostics (9), we can see that the Residuals vs Fitted plot shows an obvious pattern, a certain concentration then slowly spreads out, an obvious showing of heteroscedacity. This violates the variance assumption of Linear Regression. The Q-Q plot shows that there are trailing off at the ends, which is not so much of a violation of the normality assumption but is still questionable. The Standardized Residuals also behaves in a similar way of the Residuals, a concentration then spreads out. Also, there is a leverage point that goes beyond Cook's distance which would be considered an influential point. We chose not to remove this point since our sample size was quite large, but we wanted to keep in mind that it does have a slight effect on the overall predicted linear relationship.

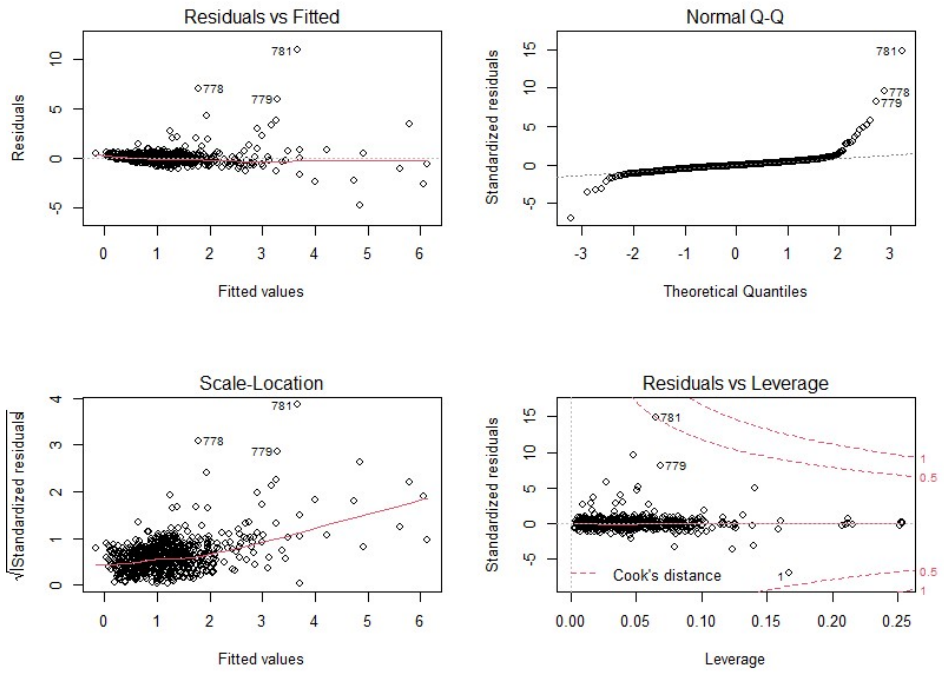

Figure 9: Height Model Diagnostics

With the goal of a comparative view we've also done the diagnostics to the model that used the "forward" as a direction of the SignifReg function. The forward model has produced the same exact model as the stepwise model. This is not uncommon. Each algorithm, both forward and stepwise, follows the path of maximizing adjusted-R^2, as that is our criterion of choice. At each step of the algorithm, it looks at the possible predictors to add, so that depending on the variables added they can stop at different final models because at each entry of a predictor, the entire model changes, which can render completely different produced models, yet here is an example of both approaches ending up with the same model. Thus, the diagnostics for the forward model is exactly the same as the stepwise model.

As with the model that predicts the weight, we've also taken the liberty to use the Boxcox transformation for the model that predicts the height. Our goal here is to improve the diagnostics of the model such that it adheres to the fundamental assumptions of Linear Regression.

```
Call:
lm(formula = (height_m^lambda - 1)/lambda ~ weight_kg + base_total +
    hp + generation + base_happiness + type2 + against_electric +
    against_psychic + sp_defense + is_legendary + experience_growth +
    against_ice + against_grass)

Residuals:
     Min       1Q   Median       3Q      Max
-3.01899 -0.21426  0.01966  0.22289  2.44252

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1.567e+00  1.695e-01  -9.246  < 2e-16 ***
weight_kg          1.815e-03  1.826e-04   9.940  < 2e-16 ***
base_total         3.089e-03  2.439e-04  12.665  < 2e-16 ***
hp                 4.411e-03  7.845e-04   5.623 2.62e-08 ***
generation        -5.330e-02  8.564e-03  -6.224 7.97e-10 ***
base_happiness    -4.822e-03  9.476e-04  -5.089 4.53e-07 ***
type2bug           2.354e-01  1.974e-01   1.193 0.233402
type2dark          1.429e-01  1.048e-01   1.363 0.173203
type2dragon        2.811e-01  1.174e-01   2.394 0.016911 *
type2electric     -2.255e-01  1.483e-01  -1.520 0.128802
type2fairy        -1.820e-01  8.577e-02  -2.122 0.034179 *
type2fighting     -1.010e-02  1.004e-01  -0.101 0.919853
type2fire          4.564e-03  1.252e-01   0.036 0.970932
type2flying       -5.985e-02  9.210e-02  -0.650 0.515975
type2ghost         2.846e-01  1.199e-01   2.374 0.017825 *
type2grass         9.544e-02  1.031e-01   0.926 0.354952
type2ground        1.233e-01  9.967e-02   1.237 0.216548
type2ice           1.389e-01  1.159e-01   1.199 0.230945
type2normal        1.728e-01  2.203e-01   0.784 0.433090
type2poison       -1.221e-01  9.066e-02  -1.347 0.178507
type2psychic       5.724e-02  8.795e-02   0.651 0.515360
type2rock         -1.404e-01  1.253e-01  -1.121 0.262807
type2steel        -8.243e-02  1.021e-01  -0.807 0.419872
type2water        -2.580e-03  1.149e-01  -0.022 0.982094
against_electric   8.564e-02  4.431e-02   1.933 0.053622 .
against_psychic    1.641e-01  4.302e-02   3.814 0.000147 ***
sp_defense        -1.200e-03  8.302e-04  -1.445 0.148745
is_legendary      -3.922e-01  6.968e-02  -5.629 2.55e-08 ***
experience_growth  9.054e-08  1.083e-07   0.836 0.403285
against_ice        3.810e-02  3.644e-02   1.045 0.296163
against_grass     -1.578e-02  2.757e-02  -0.572 0.567267
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4342 on 770 degrees of freedom
Multiple R-squared:  0.6292,    Adjusted R-squared:  0.6147
F-statistic: 43.55 on 30 and 770 DF,  p-value: < 2.2e-16
```

Figure 10: Boxcox Transformed Height Model Diagnostics

By applying the summary function (11) to the transformed model we obtained an Adjusted-R^2 of 0.6147. Meaning that this model is able to explain 61.54% of the variability for a Pokemon's height (in meters) from the 13 predictors. A significant increase in the Adjusted-R^2, a more than 10% improvement with just using the Boxcox transformation. Also, some significance of some predictors have changed. Weight_kg, base_total, hp, generation, base_happiness are still significant as with the previous model but some variables of type2 have changed. Also, against_psychic and is_legendary has now become significant as the transformation has been applied. Showing that the model as a whole has better explanation.

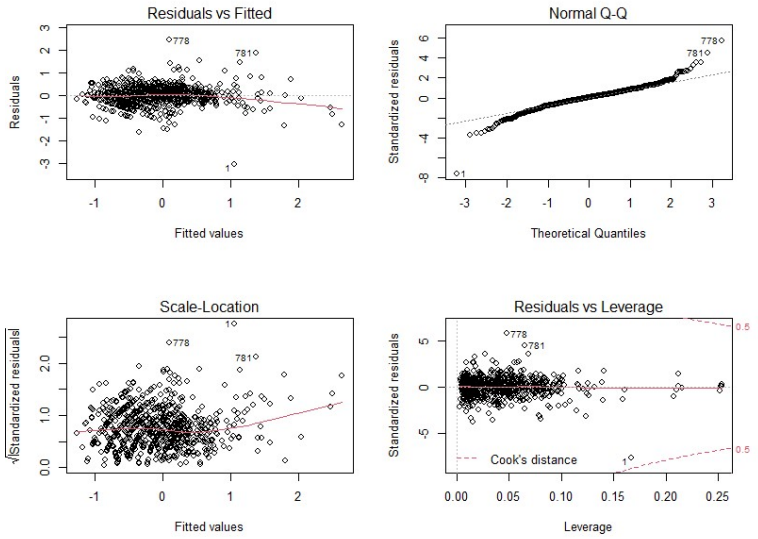

Figure 11: Boxcox Transformed Stepwise Model Diagnostics

Looking at the diagnostics of the transformed model (11) we see a lot of improvements. Although there is still some concentration in the Residuals vs Fitted plot that to a degree still violates the variance assumption of Linear Regression, it can be argued that this can be a satisfactory model. This can be backed up by the improvement seen in the standardized residuals, showing an apparent cloudy points of data that seems to suggests a degree of homoscedacity. The Q-Q plot also has improved to a degree which is a better version of the previous model, but the trailing off in the ends of tail can be still be questionable. Through the transformed model the point that is beyond the Cook's distance has been eliminated.

We argue that through the Boxcox transformation, the model is a valid model since it provides a set of satisfactory diagnostics that adheres to the fundamental assumptions of Linear Regression with a strong adjusted-R^2 such that it is able to explain 61.54% of the variability for a Pokemon's height.

4 Conclusion

Using the Pokemon data set, our goal was to create multiple linear regression models to predict both a Pokemon's height and weight. This process is achieved by determining our base models, applying our methods and then using Boxcox transformation. Once we use transformations, we utilized a similar application to what was carried out in our base models then compare our directions (stepwise and/or forward). Hence, this led to our observations gathered from diagnostics, while also checking for any violations of the fundamental assumptions of Linear Regression.
Therefore, by running Signifreg, we were able to gather the following. We perceived that the regression model predicting height with Boxcox transformation as well as using both stepwise and forward, provided results left for further augmentation. As mentioned before, running the Signifreg function as well as checking for multicollinearity using VIF, the most notable value for our height and weight variables is Adjusted-R^2. Due to having our criterion set to "r-adj", we were able to gather that an Adjusted R^2 value for our weight model explained variability over 50% while our height model with transformations explained variability over 60%.

References

[1] *The Official Pokémon Website*. Pokemon.com. (n.d.). Retrieved December 9, 2021, from https://www.pokemon.com/us/pokedex/.

[2] Kim, J., & Zambom, Z. A. Package signifreg. CRAN. Retrieved December 9, 2021, from https://cran.r-project.org/web/packages/SignifReg/index.html.

5 Appendix

R Code:

```
### Pokemon Project Code (Model 1) ###

## Setup Variables / Load Data ##

library(SignifReg)
library(MASS)
library(car)

pokemon = read.csv("pokemon_updated.csv",header=TRUE)
attach(pokemon)

scope1 = ~ attack + defense + sp_attack + sp_defense +  base_egg_steps
+ base_happiness + base_total + cap_rate +
```

```
 experience_growth + hp + speed + is_legendary + against_bug + against_dark
+ against_dragon + against_electric +
 against_fairy + against_fight +against_fire +against_flying +against_ghost
+ against_grass + against_ground +
 against_ice + against_normal + against_poison + against_psychic +against_rock
+against_steel +against_water

## Building the model ##

m1 = SignifReg(null_model,direction="both",criterion="r-adj",scope=scope1)


## Boxcox Transformation ##

bc = boxcox(m1)
lambda <- bc$x[which.max(bc$y)]

bc_m1 = lm((weight_kg^lambda - 1)/lambda ~ base_total + base_happiness
+ speed + hp + against_poison +
 is_legendary + sp_attack + against_water)


## Check Diagnostics ##

summary(bc_m1)
vif(bc_m1)
plot(bc_m1)



heightmodel <- lm(height_m ~ against_bug + against_dark + against_dragon
+ against_electric + against_fairy + against_fight + against_fire +
against_flying + against_ghost + against_grass + against_ground + against_ice
+ against_normal + against_poison + against_psychic + against_rock
+ against_steel + against_water + attack + base_egg_steps + base_happiness
+ base_total + cap_rate + hp + sp_defense + speed + type1+ type2 +
generation + is_legendary + weight_kg + experience_growth + defense
+ sp_attack)
nullmodel <- lm(height_m ~ 1)
scope = list(lower=formula(nullmodel), upper=formula(heightmodel) )

select = SignifReg(nullmodel, scope = scope, direction = "both", trace
= FALSE, criterion = "r-adj", alpha = 1)
vif(select)
AIC(select)
par(mfrow=c(2,2))
```

```
plot(select)
summary(select)

selectF = SignifReg(nullmodel, scope = scope, direction = "both", trace
= FALSE, criterion = "r-adj", alpha = 1)
vif(selectF)
AIC(selectF)
par(mfrow=c(2,2))
plot(selectF)
summary(selectF)


bc = boxcox(select) ##since select and selectF are the same model
lambda <- bc$x[which.max(bc$y)]
bc_height = lm((height_m^lambda - 1)/lambda ~ weight_kg + base_total
+ hp + generation + base_happiness + type2 +against_electric + against_psychic
+ sp_defense + is_legendary + experience_growth + against_ice + against_grass)

summary(bc_height)
vif(bc_height)
par(mfrow=c(2,2))
plot(bc_height)
vif(bc_height)
```