Pokemon

2022-10-05

In this project, we were interested in seeing what factors affect how easily a pokemon is caught.

In order to do this, a pokemon dataset from Kaggle was used.

The dataset contains a variable called `capture_rate`. It is important to understand what this does. It is encoded as a 8-bit unsigned integer. It may help to understand how this is used. First, a capture value is calculated

$$\text{Capture Value} = \frac{3\text{HP}_{\text{max}} - 2\text{HP}_{\text{current}}}{3\text{HP}_{\text{max}}} \cdot \text{Catch Rate} \cdot \text{Modifier}_{\text{status}} \cdot \text{Modifier}_{\text{ball}}$$

Now, If the capture value is at least 255, the catch is guaranteed. Otherwise, the formula below is used to compute some number.

$$\text{Catch Value} = \left(2^{20} - 2^{4}\right) \left(\frac{\text{Capture Value}}{2^{24} - 2^{16}}\right)^{1/4}$$

In order to calculate whether a pokemon is captured, a random 16-bit unsigned integer is generated. If it is less than or equal to the catch value, then the pokemon is caught. Otherwise this is repeat up to two more times. Each of these is called a “shake”.

Assuming the randomly generated integer follow a uniform distribution $\text{Catch Value} \cdot 2^{-16}$, the number of failed shakes before a pokemon is caught follows a geometric distribution with probability of success $\frac{\text{Catch Rate}}{2^{16}}$. In this case, a pokemon is caught if the number of failed shakes is at most 2.

Using this process, min and max probabilities of catch were added to the data.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
catch_value <- function(max_hp, curr_hp, catch_rate, ball_modifier = 1, status_modifier = 1)
   (( 3 * max_hp - 2 * curr_hp ) * (catch_rate * ball_modifier) / (3 * max_hp) ) * status_modifier

p_shake <- function(x)
  ifelse(
    x < 255, # check if x < 255 element wise
    (65535 * sqrt(sqrt(x / 255)) * 2^-16), # if so compute probability of success with formula
    1 # otherwise success
  )
```

```
pokemon <- read_csv("pokemon.csv") %>%
  filter(name != "Minior") %>%
  mutate(
    katakana_name = japanese_name %>% str_extract(r"([^[a-zA-Z0-9?]]+)"),
    romaji_name = japanese_name %>% str_extract(r"([a-zA-Z0-9?]+)"),
    capture_rate = as.numeric(capture_rate),
    p_catch_min = catch_value(hp, hp, capture_rate) %>% p_shake() %>% pgeom(2, .),
    p_catch_max = catch_value(hp, 1, capture_rate) %>% p_shake() %>% pgeom(2, .),
    japanese_name = NULL
  )
```

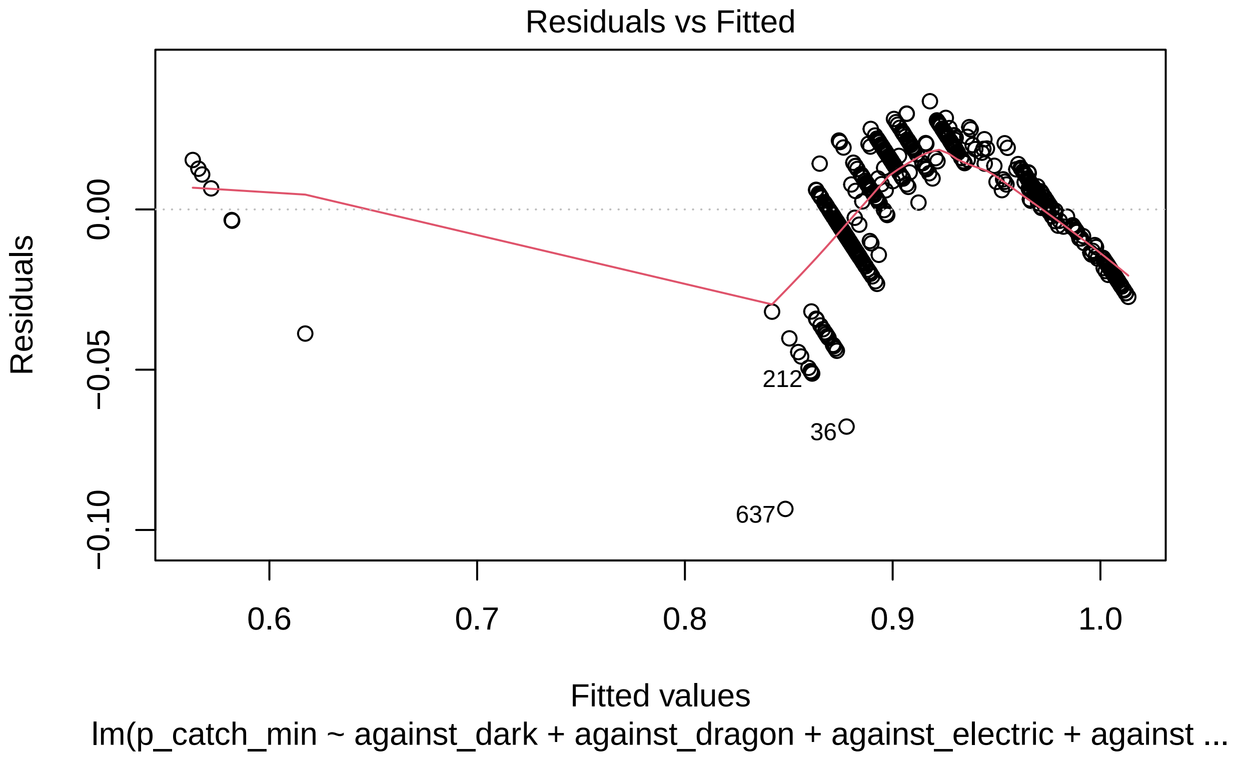
```
## Rows: 801 Columns: 41
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): abilities, capture_rate, classfication, japanese_name, name, type1...
## dbl (34): against_bug, against_dark, against_dragon, against_electric, again...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

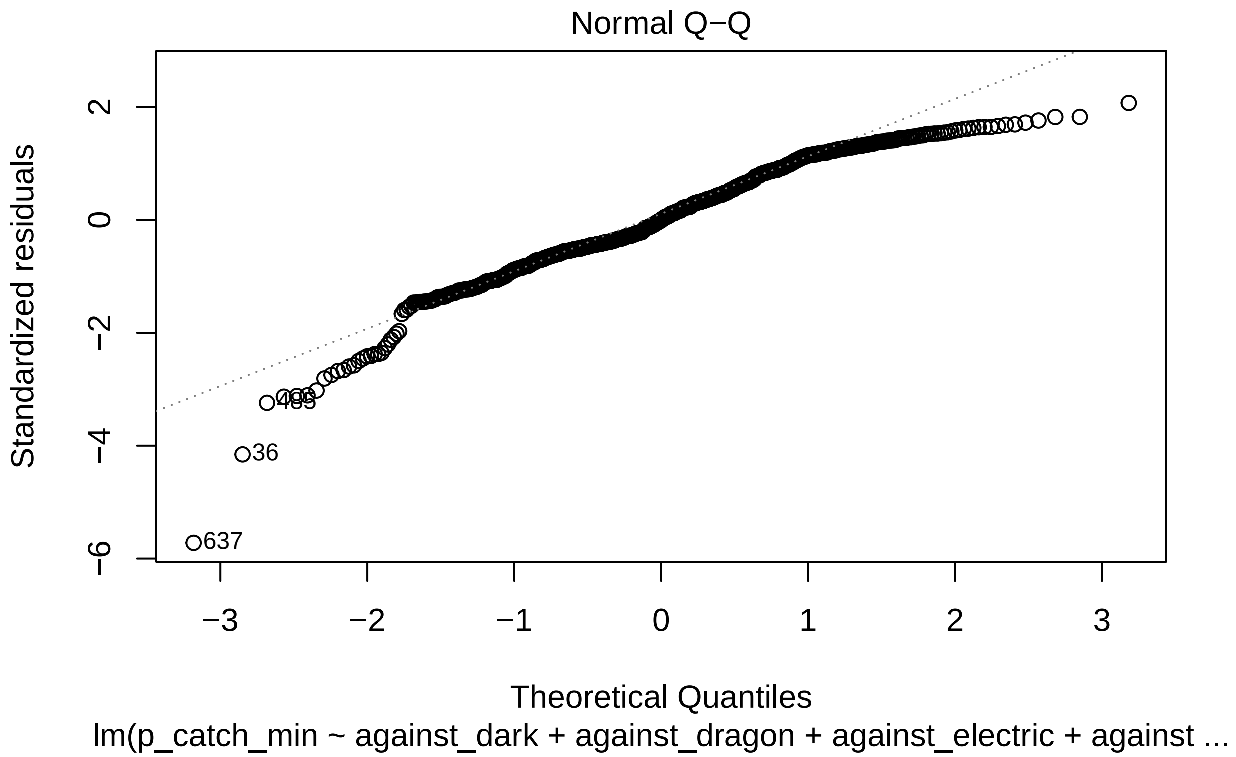
Now that this is done, it may help to look at relations between variables.

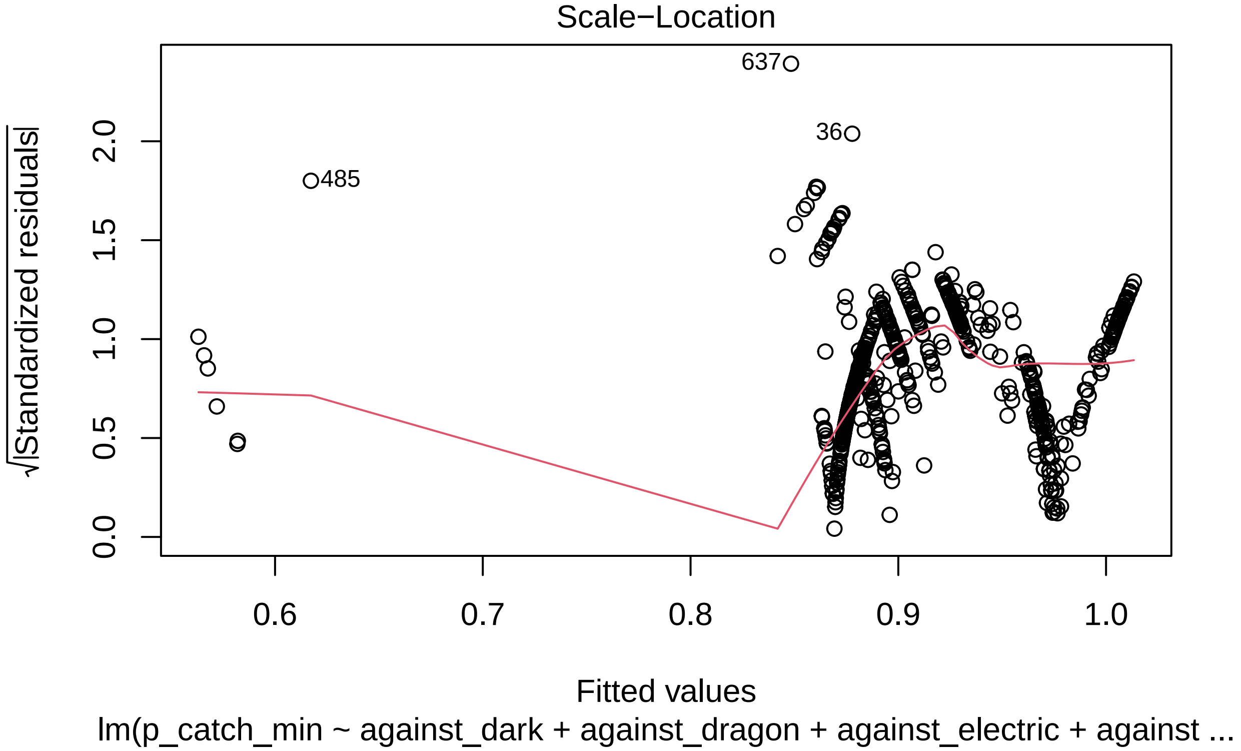
```
lm1 <- lm(
  p_catch_min ~ . - p_catch_max,
  data = pokemon %>% select_if(is.numeric)
) %>%
  step(trace = 0)

summary(lm1)
```

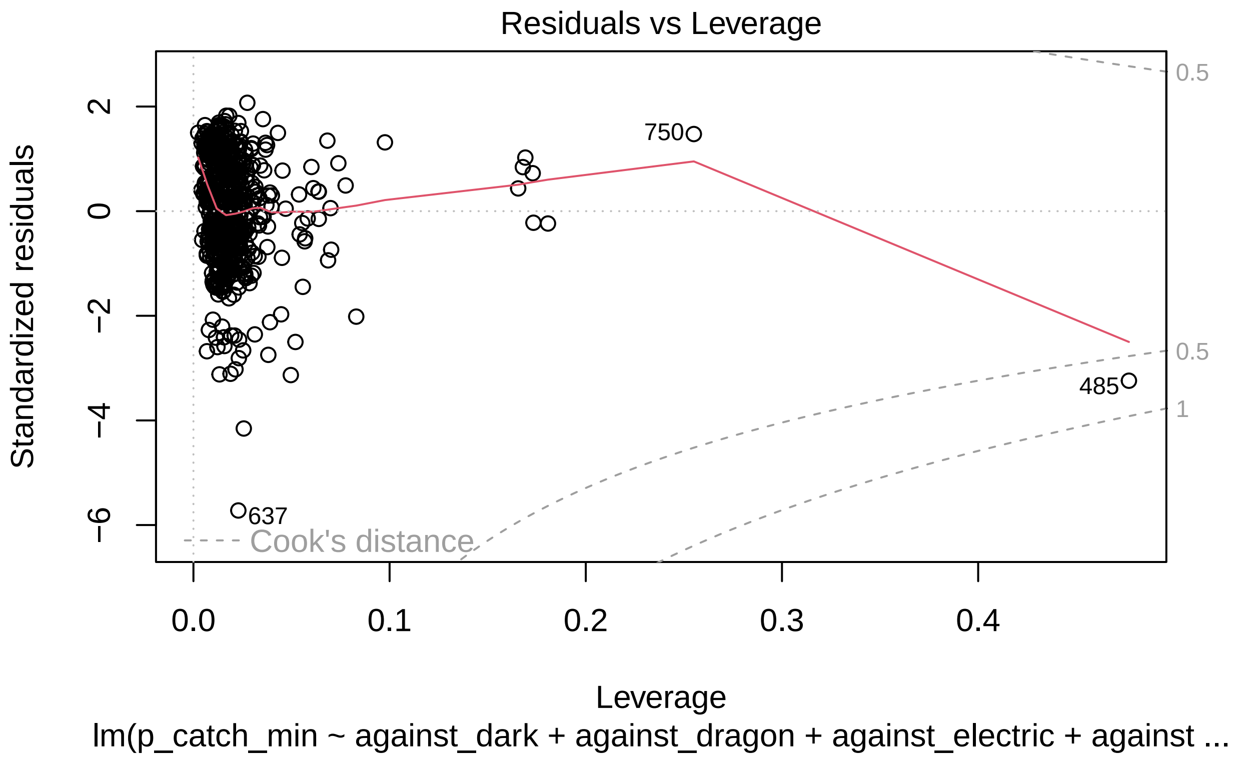
```
##
## Call:
## lm(formula = p_catch_min ~ against_dark + against_dragon + against_electric +
##     against_fairy + against_ghost + against_ground + against_psychic +
##     base_egg_steps + capture_rate + percentage_male + sp_defense +
##     weight_kg + is_legendary, data = pokemon %>% select_if(is.numeric))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.093434 -0.009324 -0.000088  0.012959  0.033750
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.640e-01  5.582e-03 154.792  < 2e-16 ***
## against_dark      5.138e-03  2.817e-03   1.824 0.068615 .
## against_dragon   -5.599e-03  2.595e-03  -2.157 0.031340 *
## against_electric -1.918e-03  1.008e-03  -1.903 0.057508 .
## against_fairy     5.587e-03  1.793e-03   3.115 0.001916 **
## against_ghost     2.789e-03  1.745e-03   1.599 0.110396
## against_ground    1.930e-03  9.861e-04   1.957 0.050730 .
## against_psychic   3.206e-03  1.451e-03   2.210 0.027477 *
## base_egg_steps   -1.747e-06  3.822e-07  -4.571 5.78e-06 ***
## capture_rate      5.889e-04  1.065e-05  55.308  < 2e-16 ***
## percentage_male  -1.206e-04  3.323e-05  -3.629 0.000306 ***
## sp_defense       -7.730e-05  2.858e-05  -2.705 0.007014 **
## weight_kg        -1.766e-05  9.683e-06  -1.824 0.068669 .
## is_legendary     -2.362e-01  1.001e-02 -23.591  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01652 on 670 degrees of freedom
##   (116 observations deleted due to missingness)
## Multiple R-squared:  0.9245, Adjusted R-squared:  0.9231
## F-statistic: 631.5 on 13 and 670 DF,  p-value: < 2.2e-16
```

```
plot(lm1)
```




Normal Q−Q
Standardized residuals
2
0
−2
−4
−6
−3
−2
−1
0
1
2
3
485
36
637
Theoretical Quantiles
lm(p_catch_min ~ against_dark + against_dragon + against_electric + against ...


Scale−Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
0.6
0.7
0.8
0.9
1.0
637
36
485
Fitted values
lm(p_catch_min ~ against_dark + against_dragon + against_electric + against ...



```
lm2 <- lm(
  p_catch_min ~
    poly(against_dark, 4, raw = T) +
    poly(against_dragon, 4, raw = T) +
    poly(against_electric, 4, raw = T) +
    poly(against_fairy, 4, raw = T) +
    poly(against_ghost, 4, raw = T) +
    poly(against_ground, 4, raw = T) +
    poly(against_psychic, 4, raw = T) +
    poly(base_egg_steps, 4, raw = T) +
    poly(capture_rate, 4, raw = T) +
    poly(percentage_male, 4, raw = T) +
    poly(sp_defense, 4, raw = T) +
    poly(weight_kg, 4, raw = T) +
    is_legendary,
  data = pokemon %>% select_if(is.numeric) %>% remove_missing()
) %>%
  step(trace = 0)
```

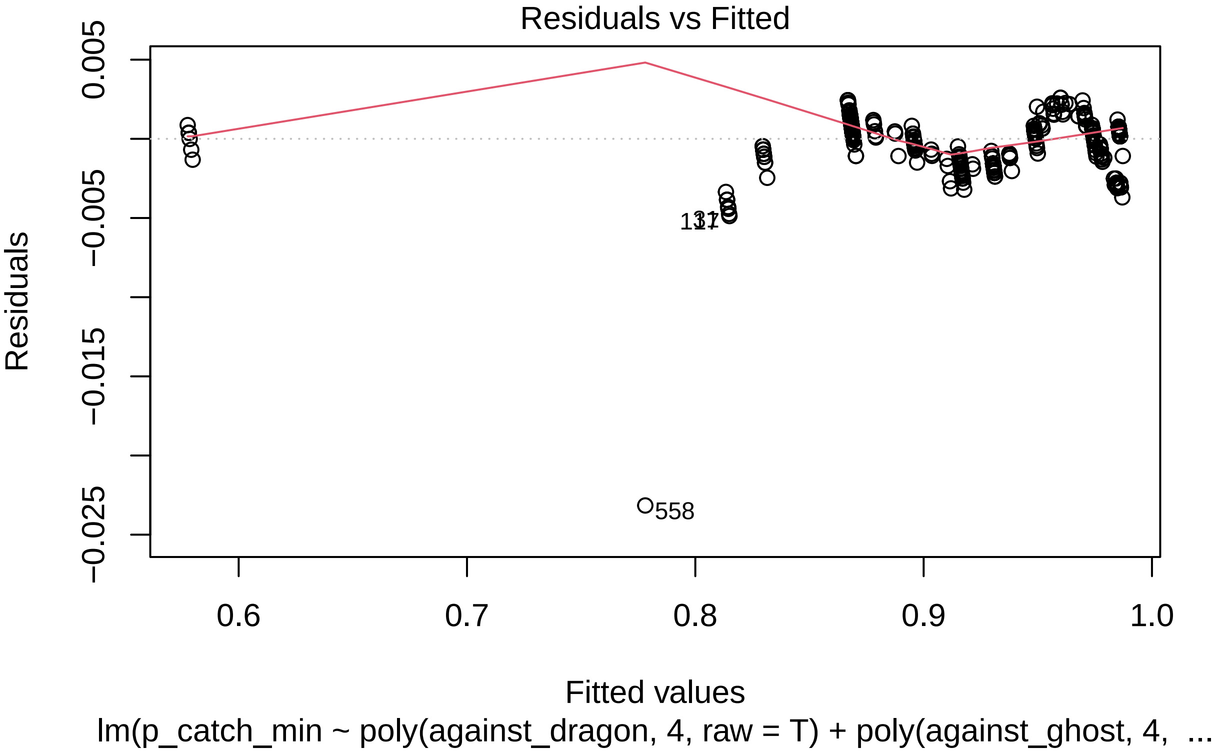
```
## Warning: Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
```

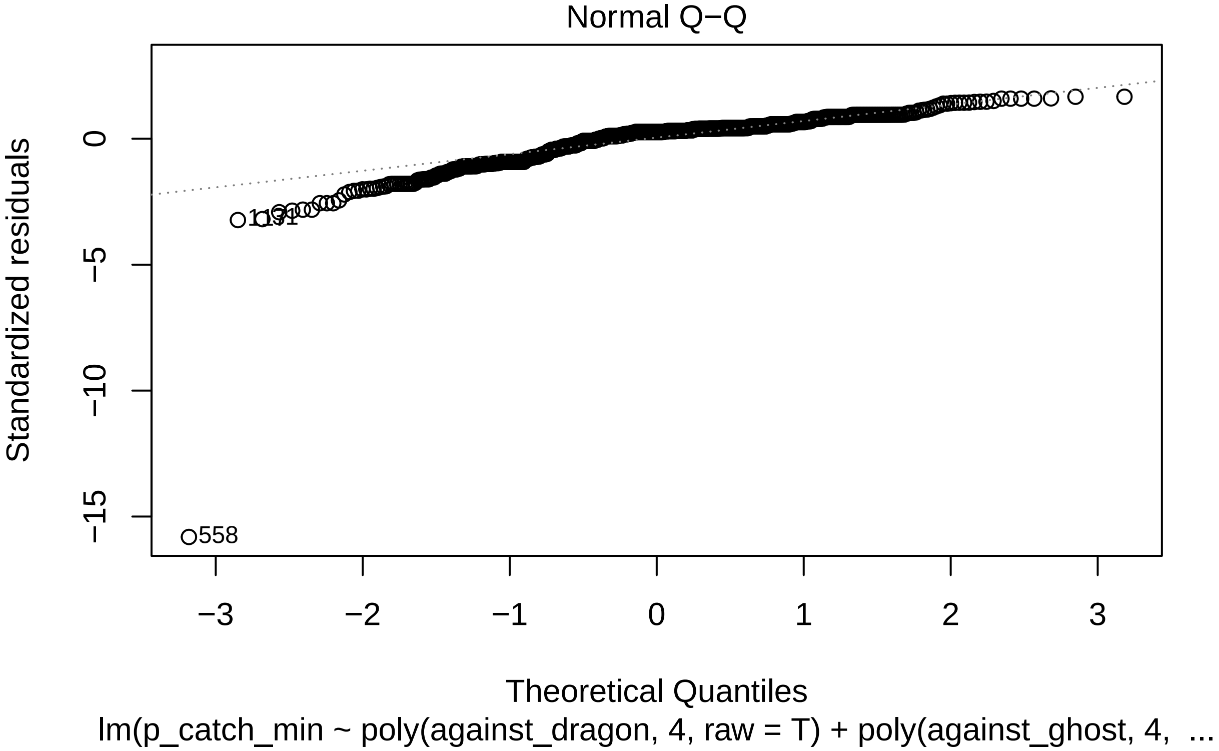
```
summary(lm2)
```

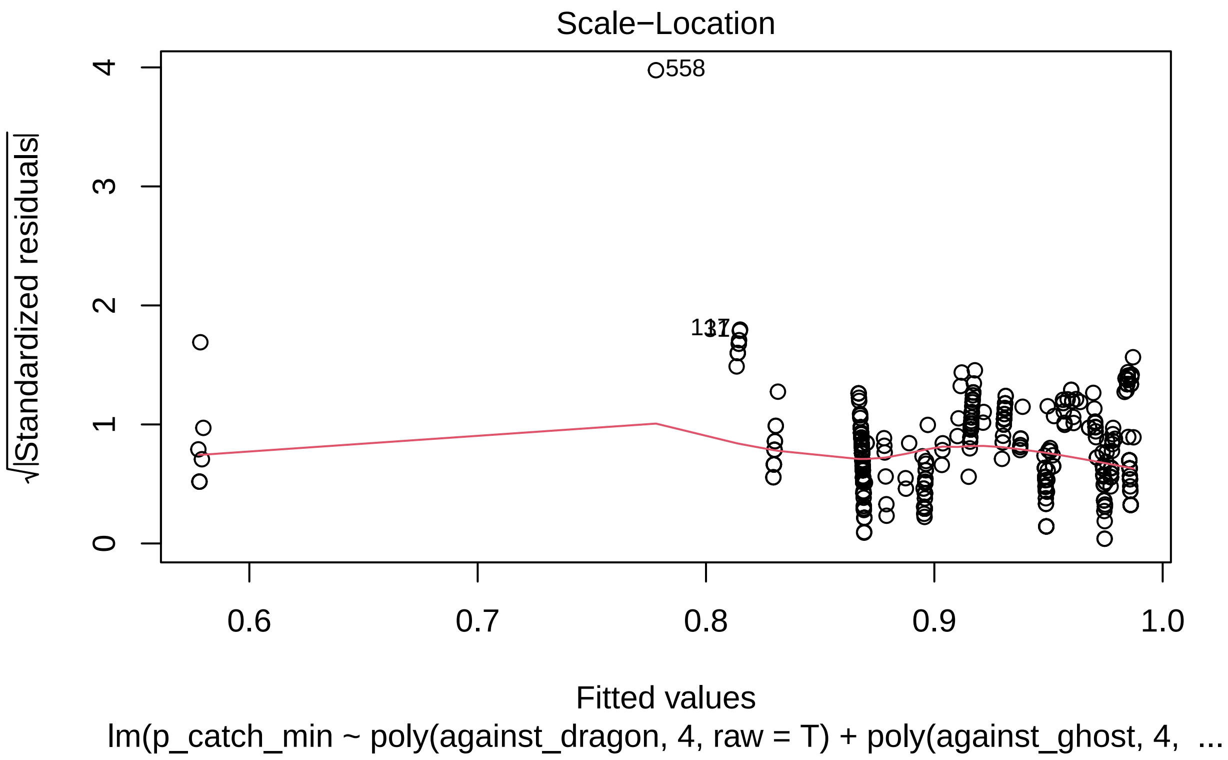
```
##
```

```
## Call:
## lm(formula = p_catch_min ~ poly(against_dragon, 4, raw = T) +
##     poly(against_ghost, 4, raw = T) + poly(base_egg_steps, 4,
##     raw = T) + poly(capture_rate, 4, raw = T) + poly(percentage_male,
##     4, raw = T) + is_legendary, data = pokemon %>% select_if(is.numeric) %>%
##     remove_missing())
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0231553 -0.0006214  0.0004166  0.0007566  0.0025921
##
## Coefficients: (2 not defined because of singularities)
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        7.176e-01  1.962e-03 365.710   <2e-16 ***
## poly(against_dragon, 4, raw = T)1  -2.364e-03  2.023e-03  -1.169   0.2429
## poly(against_dragon, 4, raw = T)2   3.068e-03  2.859e-03   1.073   0.2836
## poly(against_dragon, 4, raw = T)3  -7.549e-04  9.481e-04  -0.796   0.4262
## poly(against_dragon, 4, raw = T)4         NA         NA      NA       NA
## poly(against_ghost, 4, raw = T)1   -1.565e-04  1.477e-03  -0.106   0.9156
## poly(against_ghost, 4, raw = T)2   -5.968e-04  2.136e-03  -0.279   0.7801
## poly(against_ghost, 4, raw = T)3    2.987e-04  7.116e-04   0.420   0.6748
## poly(against_ghost, 4, raw = T)4          NA         NA      NA       NA
## poly(base_egg_steps, 4, raw = T)1  -2.163e-06  1.030e-06  -2.099   0.0362 *
## poly(base_egg_steps, 4, raw = T)2   4.079e-10  2.057e-10   1.983   0.0478 *
## poly(base_egg_steps, 4, raw = T)3  -3.016e-14  1.520e-14  -1.984   0.0477 *
## poly(base_egg_steps, 4, raw = T)4   6.183e-19  3.110e-19   1.988   0.0472 *
## poly(capture_rate, 4, raw = T)1     5.000e-03  4.155e-05 120.325   <2e-16 ***
## poly(capture_rate, 4, raw = T)2    -4.145e-05  5.496e-07 -75.410   <2e-16 ***
## poly(capture_rate, 4, raw = T)3     1.634e-07  2.844e-09  57.433   <2e-16 ***
## poly(capture_rate, 4, raw = T)4    -2.403e-10  4.986e-12 -48.198   <2e-16 ***
## poly(percentage_male, 4, raw = T)1 -3.425e-05  5.857e-05  -0.585   0.5589
## poly(percentage_male, 4, raw = T)2  2.448e-07  2.413e-06   0.101   0.9192
## poly(percentage_male, 4, raw = T)3  1.151e-08  3.411e-08   0.337   0.7359
## poly(percentage_male, 4, raw = T)4 -9.902e-11  1.608e-10  -0.616   0.5383
## is_legendary                       -1.491e-01  1.835e-03 -81.265   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001566 on 664 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.9993
## F-statistic: 5.199e+04 on 19 and 664 DF,  p-value: < 2.2e-16
```

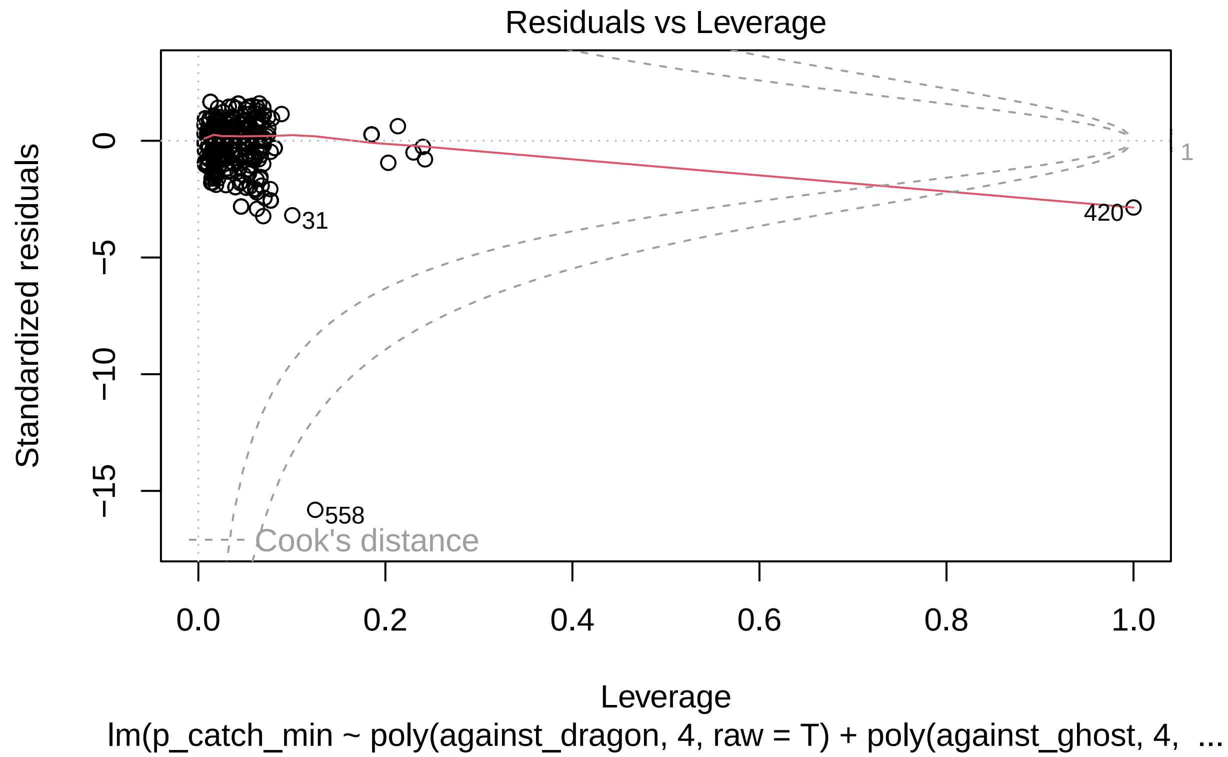
```
plot(lm2)
```


Residuals vs Fitted
Residuals
0.005
−0.005
−0.015
−0.025
0.6
0.7
0.8
0.9
1.0
131
11
558
Fitted values
lm(p_catch_min ~ poly(against_dragon, 4, raw = T) + poly(against_ghost, 4, ...


Normal Q−Q
Standardized residuals
0
−5
−10
−15
−3
−2
−1
0
1
2
3
11
131
558
Theoretical Quantiles
lm(p_catch_min ~ poly(against_dragon, 4, raw = T) + poly(against_ghost, 4, ...



```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



```
lm2 <- lm(
  p_catch_min ~
```

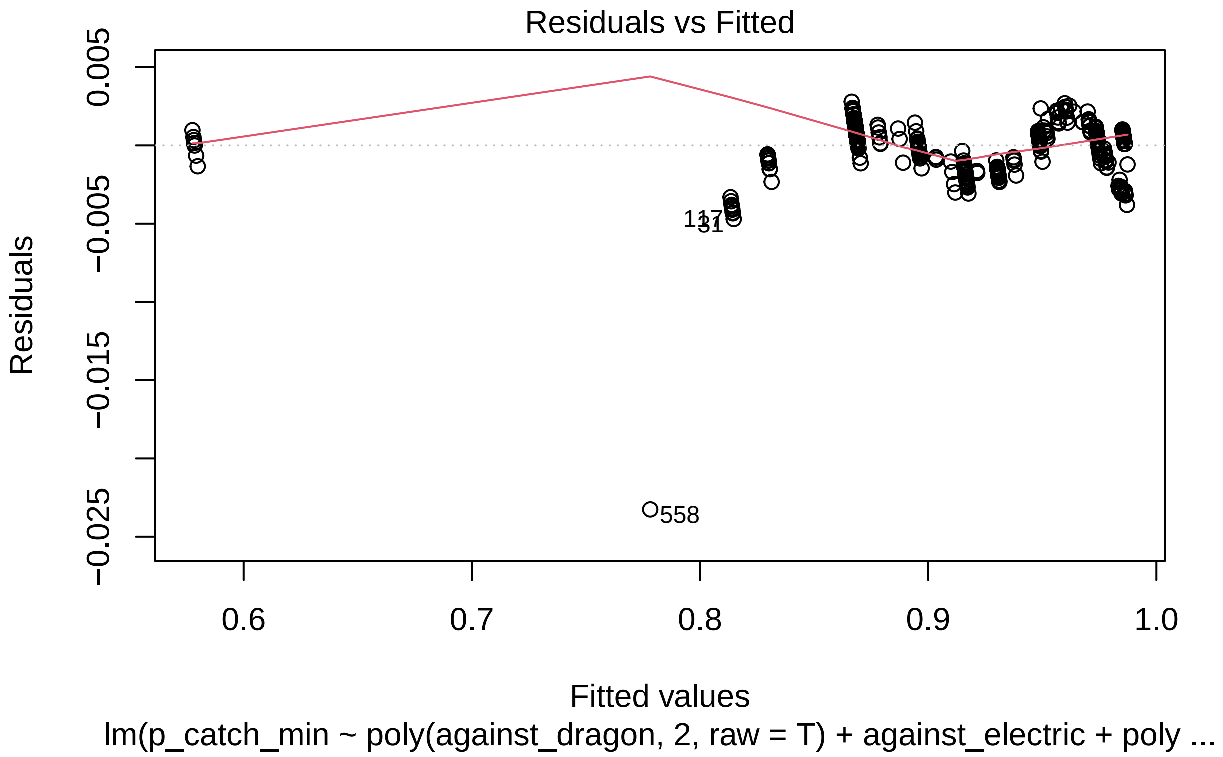
```
    against_dark +
    poly(against_dragon, 2, raw = T) +
    against_electric +
    against_fairy +
    poly(against_ghost, 2, raw = T) +
    against_ground +
    against_psychic +
    poly(base_egg_steps, 4, raw = T) +
    poly(capture_rate, 4, raw = T) +
    poly(percentage_male, 3, raw = T) +
    sp_defense +
    weight_kg +
    is_legendary,
  data = pokemon %>% select_if(is.numeric) %>% remove_missing()
) %>%
  step(trace = 0)
```

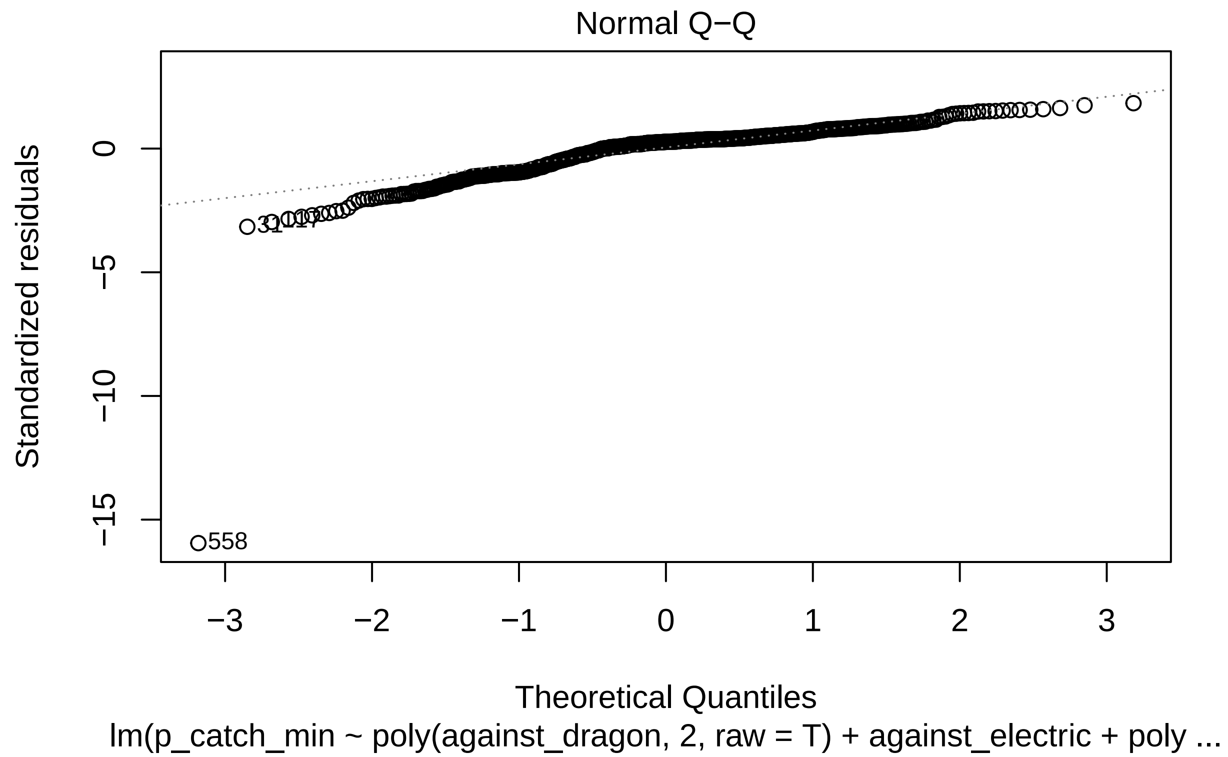
```
## Warning: Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
## Removed 116 rows containing missing values.
```

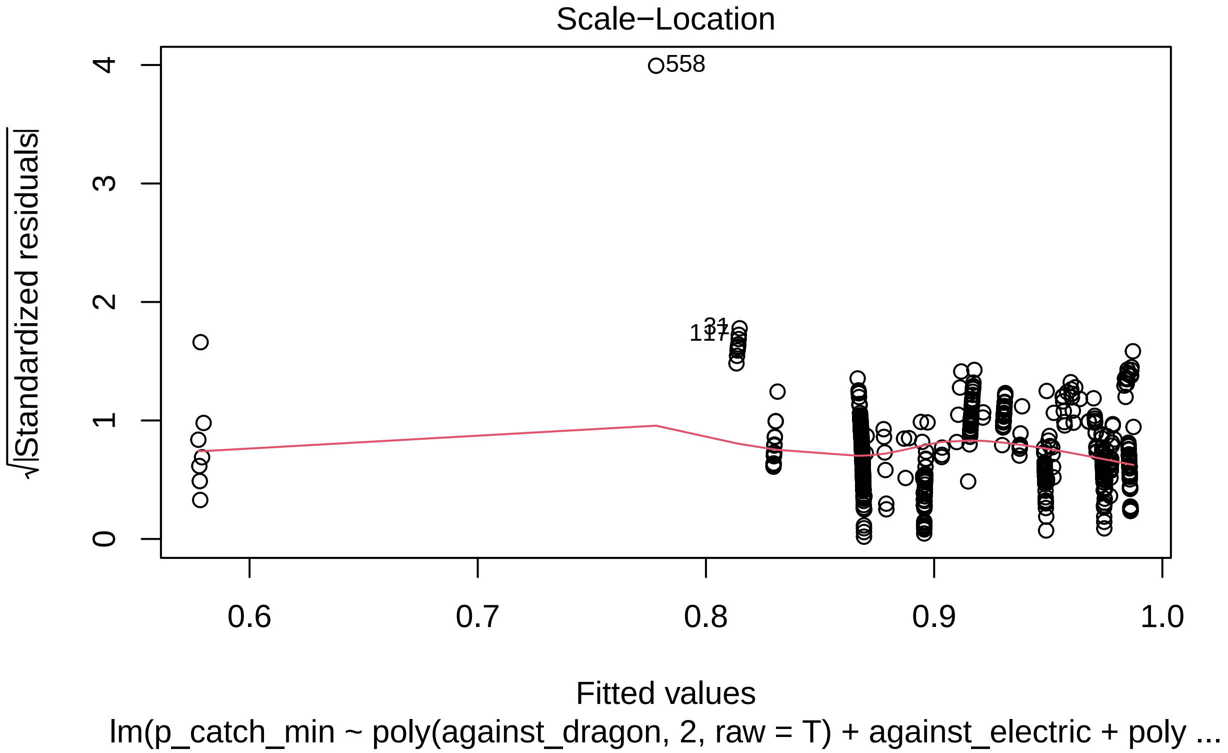
```
summary(lm2)
```

```
## 
## Call:
## lm(formula = p_catch_min ~ poly(against_dragon, 2, raw = T) + 
##     against_electric + poly(against_ghost, 2, raw = T) + poly(base_egg_steps, 
##     4, raw = T) + poly(capture_rate, 4, raw = T) + poly(percentage_male, 
##     3, raw = T) + weight_kg + is_legendary, data = pokemon %>% 
##     select_if(is.numeric) %>% remove_missing())
## 
## Residuals:
##        Min         1Q     Median         3Q        Max 
## -0.0232575 -0.0006257  0.0004201  0.0007763  0.0027886 
## 
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                          7.185e-01  1.947e-03 368.980  < 2e-16 ***
## poly(against_dragon, 2, raw = T)1  -6.726e-04  4.490e-04  -1.498  0.13467    
## poly(against_dragon, 2, raw = T)2   7.106e-04  2.233e-04   3.181  0.00153 ** 
## against_electric                   -1.944e-04  9.406e-05  -2.066  0.03917 *  
## poly(against_ghost, 2, raw = T)1   -8.002e-04  3.101e-04  -2.581  0.01008 *  
## poly(against_ghost, 2, raw = T)2    3.044e-04  1.452e-04   2.096  0.03648 *  
## poly(base_egg_steps, 4, raw = T)1  -2.500e-06  1.024e-06  -2.443  0.01484 *  
## poly(base_egg_steps, 4, raw = T)2   4.654e-10  2.042e-10   2.279  0.02300 *  
## poly(base_egg_steps, 4, raw = T)3  -3.389e-14  1.508e-14  -2.247  0.02498 *  
## poly(base_egg_steps, 4, raw = T)4   6.897e-19  3.086e-19   2.235  0.02576 *  
## poly(capture_rate, 4, raw = T)1     5.004e-03  4.132e-05 121.100  < 2e-16 ***
## poly(capture_rate, 4, raw = T)2    -4.153e-05  5.465e-07 -75.999  < 2e-16 ***
## poly(capture_rate, 4, raw = T)3     1.639e-07  2.828e-09  57.958  < 2e-16 ***
## poly(capture_rate, 4, raw = T)4    -2.414e-10  4.957e-12 -48.693  < 2e-16 ***
## poly(percentage_male, 3, raw = T)1 -6.287e-05  2.848e-05  -2.207  0.02764 *
## poly(percentage_male, 3, raw = T)2  1.614e-06  7.744e-07   2.084  0.03758 *
## poly(percentage_male, 3, raw = T)3 -9.137e-09  5.338e-09  -1.712  0.08742 .
## weight_kg                          -1.473e-06  9.321e-07  -1.581  0.11443
## is_legendary                       -1.490e-01  1.836e-03 -81.144  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001559 on 665 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.9993
## F-statistic: 5.537e+04 on 18 and 665 DF,  p-value: < 2.2e-16
```

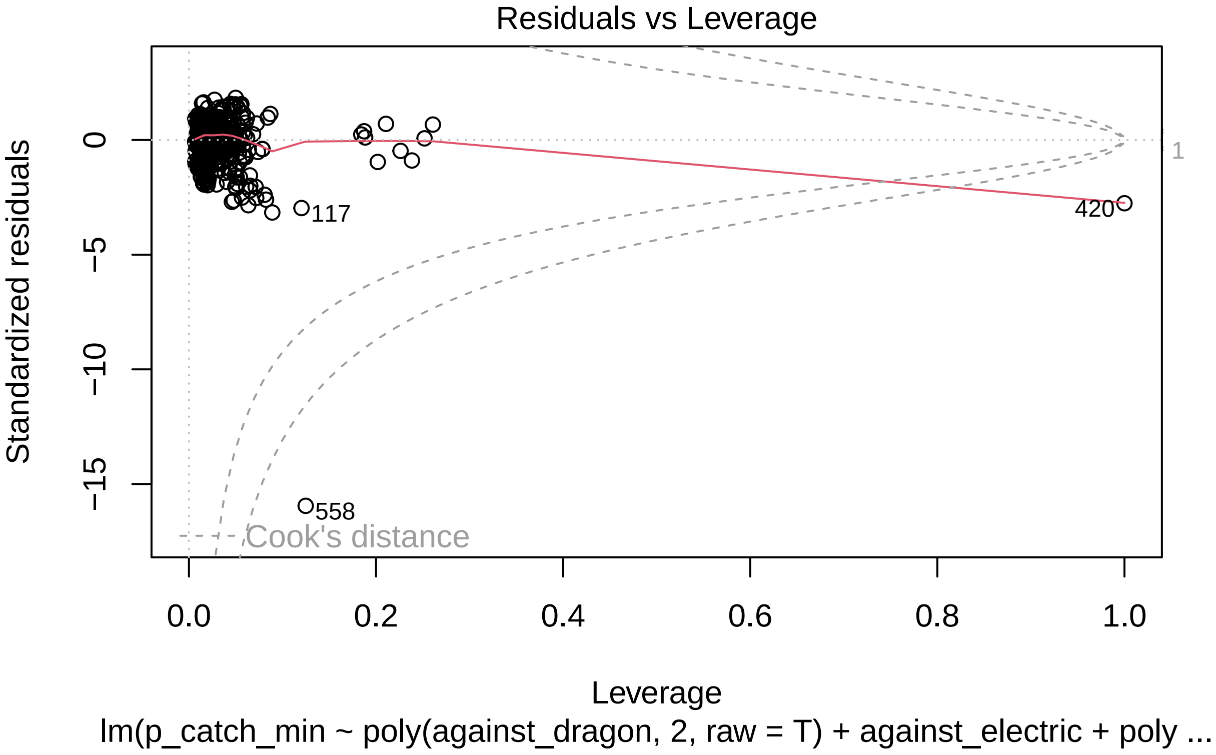
```
plot(lm2)
```







```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```


Residuals vs Leverage
Standardized residuals
0
−5
−10
−15
0.0
0.2
0.4
0.6
0.8
1.0
Leverage
lm(p_catch_min ~ poly(against_dragon, 2, raw = T) + against_electric + poly ...
117
420
558
1
Cook's distance