Part 4 | I Can't Bel-Eevee It's the End

Alex Gui and Lathan Liou

Date Submitted: May 9, 2018

Introduction

In July of 2016, Pokemon Go became an overnight sensation with hundreds of millions of people having downloaded the mobile game. The whole point of the game is to try to catch all the Pokemon available and train them (increase their combat power, which is abbreviated cp) so that you can battle other players with your strengthened Pokemon. A quick note about Pokemon is that they can evolve into stronger forms, so an evolved Pokemon will generally always have a higher cp than a non-evolved Pokemon. A number of people have tried to generate models in an attempt to predict the best way to maximize cp for their Pokemon. This is what we will attempt to do ourselves: create a model to predict combat power for an evolved Pokemon.

To refresh your memory, the dataset we are looking at is an original data set collected by $OpenIntro$[1], most likely by some individual who was playing Pokemon Go and decided to record data. The dataset contains 75 observations across 26 variables, with each observation representing a randomly generated Pokemon that the gamer caught. Four species are represented in this data, so the conclusions drawn from this modeling process will reflect the population of these 4 particular species: Eevee, Pidgey, Caterpie, and Weedle.

We avoid using "new" predictor variables in our modeling process because as a user, you wouldn't have access to any of the "new" information, but if you're interested in whether your pokemon will evolve into one with a high cp (cp_new), you would want to know which of the Pokemon's current stats can indicate a high cp_new. The variables that we are particularly interested in are cp, hp, and power_up_stardust. Our intuition is that a pokemon with a higher cp might evolve into a pokemon with a higher cp. Hp, or hit points, refers to the amount of damage a Pokemon can sustain in battle before fainting. It would be interesting to see whether a Pokemon with high hp will also have high cp. Power up stardust is used to raise cp of the pokemon, but the catch here is we don't know the ideal amount of power up stardust to max out the cp_new of the evolved pokemon.

You can follow our work here: https://github.com/alexaaag/math158-project.

Ridge Regression and Lasso
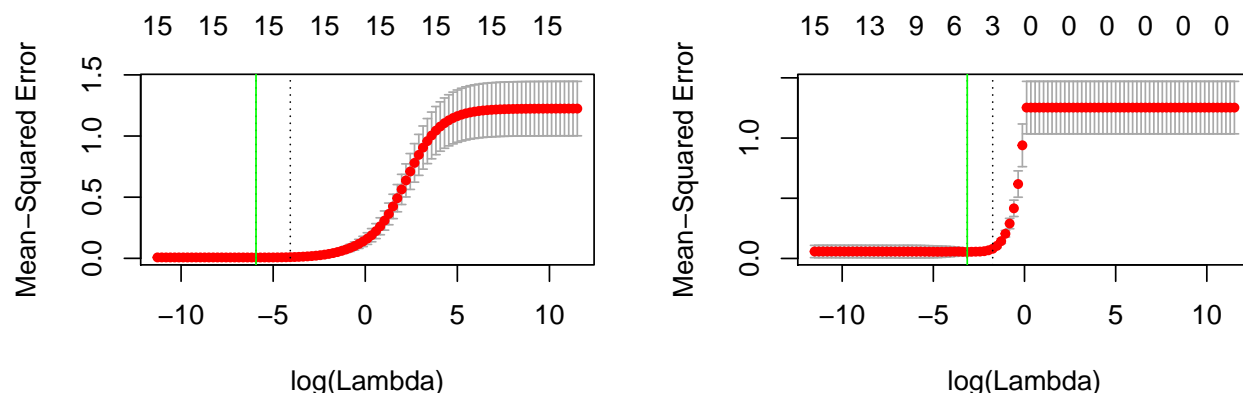
Ridge regression shrunk the coefficients closer to zero than multiple linear regression did. For instance, if we compare the coefficients of cp_{log}, it is 1.186 in our linear regression model, and 1.24e-5 in the ridge regression

model. On the other hand, lasso regression not only shrunk the coefficients but also performed variable selection, selecting cp_{log}, $attack\_strong\_value$ and $attack\_weak\_value$ to name a few.
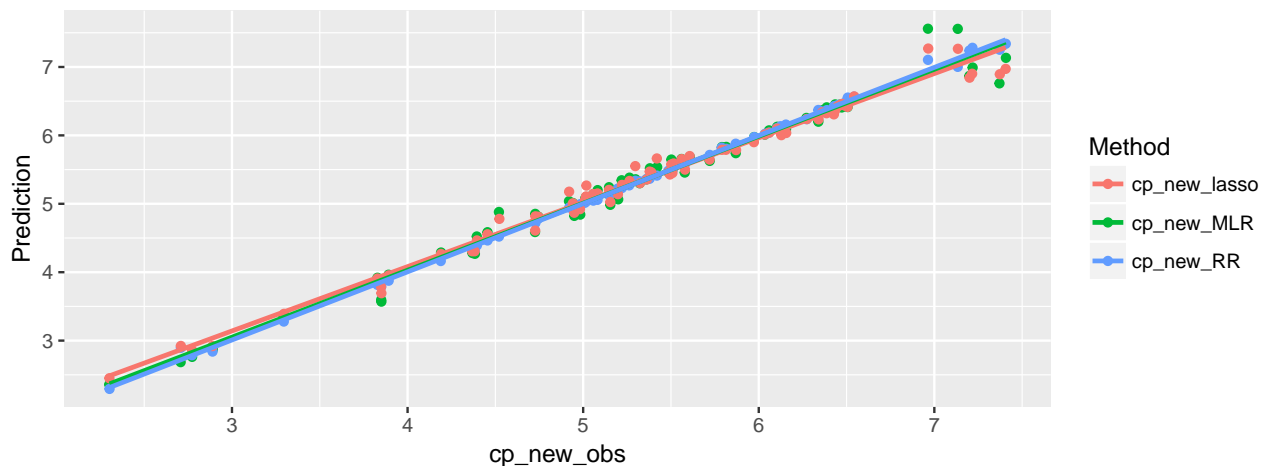
From this figure, it seems ridge regression and lasso seem to predict very similarly as multiple linear regression, since the slopes of each regression fit basically overlap each other.

Smoothing
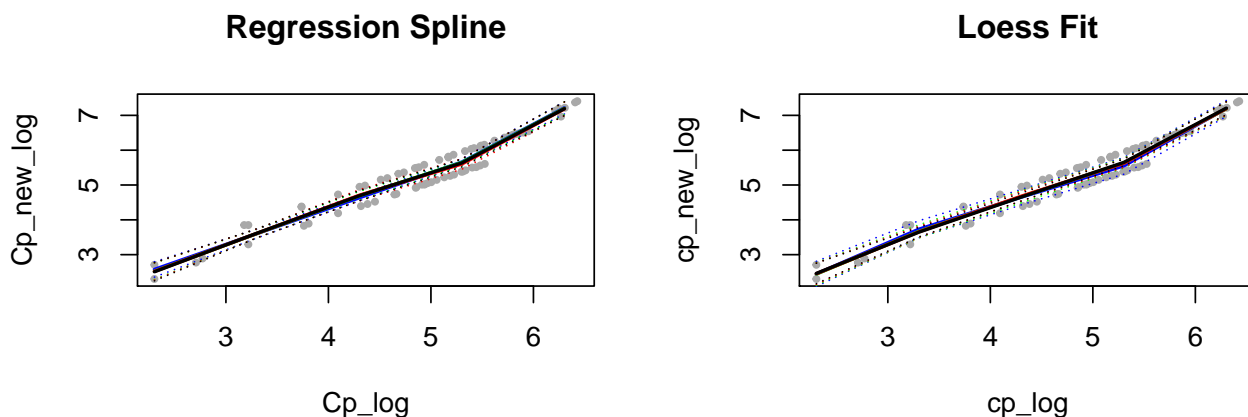
Both the smoothing spline and the loess curves fit the data extremely well. Changing the degrees of freedom, and hence the number of knots, for the smoothing splines improves the fit minimally to a certain point past which increasing the degrees of freedom actually begins to increase SSE. Likewise, increasing the span from 0.2 for loess actually increases SSE.

I would choose the regression spline model with 5 degrees of freedom because it fits the data very smoothly, and it still has a functional form which lends itself to interpretability.

Conclusion

Overall, running ridge regression, lasso, and smoothing methods such as regression splines and loess did not improve the model fit relative to multiple linear regression by very much. This is fairly unsurprising to us because we noticed how extremely well linear regression fit our dataset previously, and this can most likely be attributed to the nature of how cp_{new} is actually modeled in the game. We think cp_{new} is likely coded into the game as a function of a linear combination of certain predictors and our multiple linear regression model fairly closely matches the real model used in-game.

Something New

Q-Q Plot

A normal probability plot is used to identify substantial departures from normality in the data. We chose in particular to plot what is known as a normal quantile-quantile plot (Q-Q plot for short), which plots sample quantiles ordered and plotted in a continuous cumulative distribution function against theoretical quantiles from a standard normal distribution. A $y = x$ reference line is also plotted and if the sample data also come from a normal distribution, the points should fall roughly along this reference line. A q-q plot is important because it can provide more information about whether the normality assumption is violated.

Normal Q–Q Plot
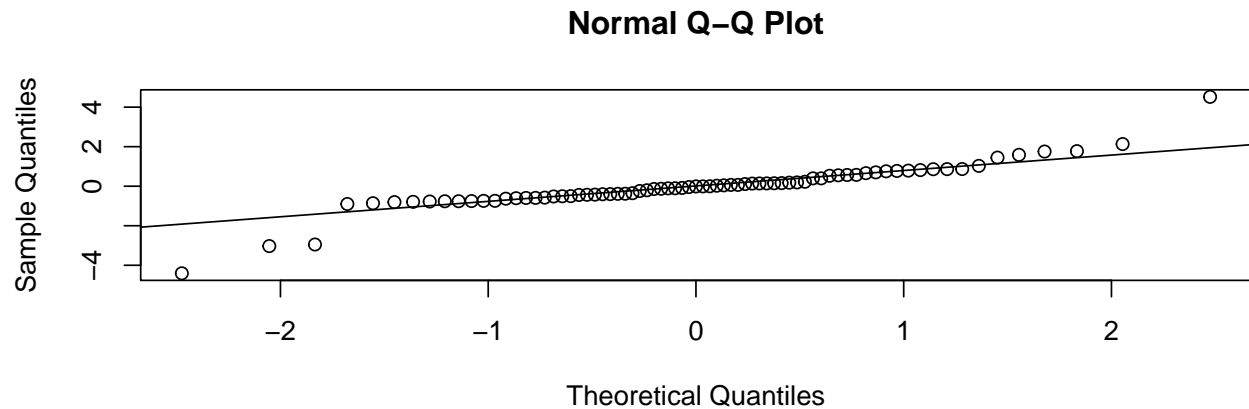
In our Q-Q plot, we observe that the outlying points do not fall on the line; however, we are not concerned with our ability to do inference since we did note a couple of outliers from before, and the majority of the data is fit by the model.

Added Variable Plots

Added variable plots (AV plots), also known as partial regression plot, attempt to show the marginal effect of adding another variable to a model already having one or more independent variables. Added variable plots are formed by first computing the residuals of regressing the response variable against the independent variable(s) but omitting the variable of interest, X_i. Let's call this $Y_{\cdot[i]}$ Next, the residuals are computed from regressing X_i against the remaining independent variables. Let's call this $X_{i\cdot[i]}$ The residuals from $Y_{\cdot[i]}$ and $X_{i\cdot[i]}$ are then plotted against each other. For this analysis, one underlying assumption is that the explanatory variables are not highly correlated with each other and that the explanatory variables have to be quantitative. One way to interpret AV plots is to compare the scatter of the points about the least squares lines and the scatter of the points around the horizontal line at 0. If the scatters are different, then we conclude that adding variable X to the model substantially reduced the error sum of squares.

Added−Variable Plots
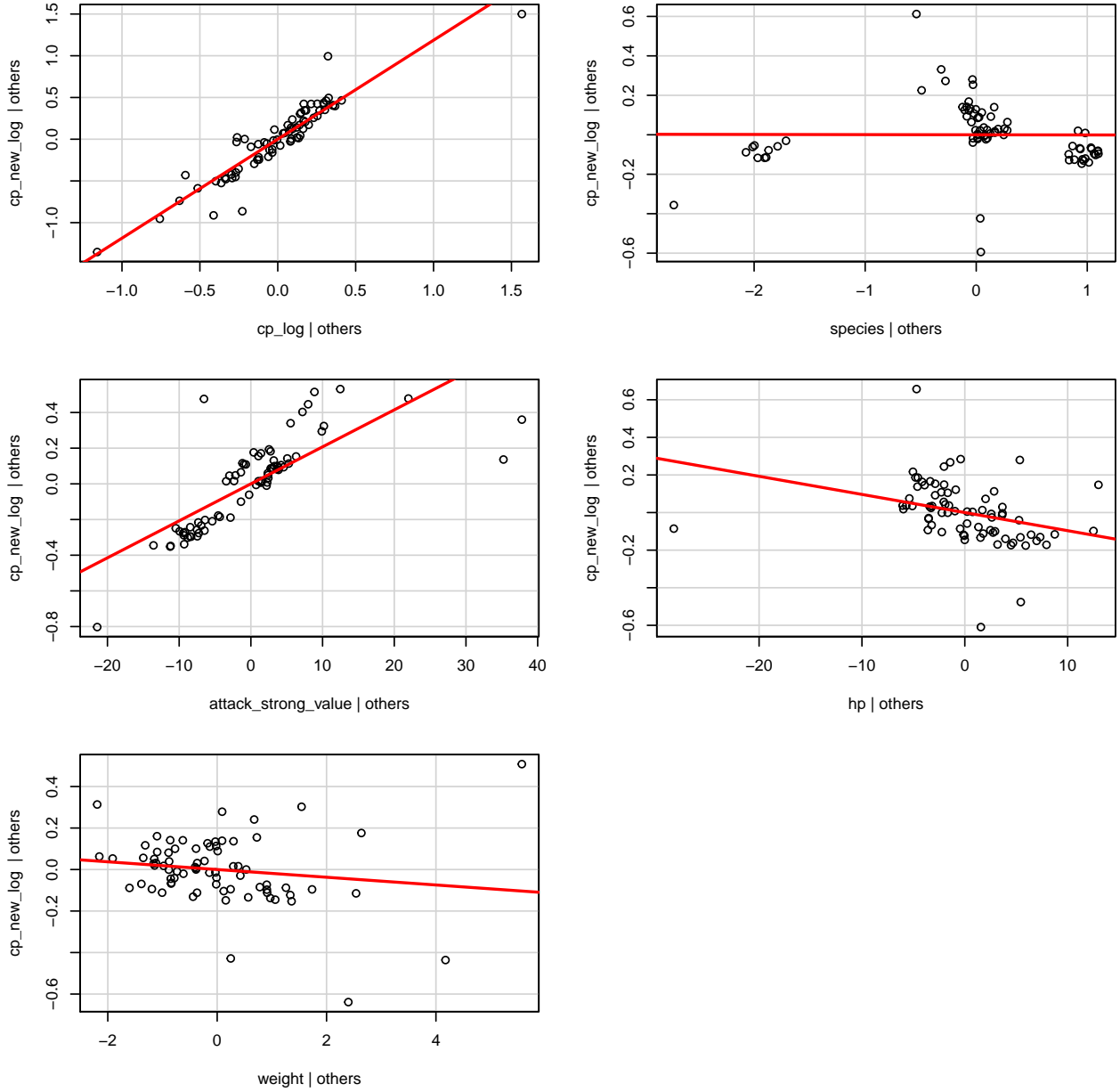
As shown, variables such as *cp_log* and *attack_strong_value* are highly correlated with *cp_new_log*, which indicate that adding *cp_log* or *attack_strong_value* substantially reduces the error sum of squares. On the other hand, variables such as weight were slightly correlated with *cp_new_log*, indicating that adding weight to the regression model does not substantially reduce the error sum of squares. In fact, the coefficient of partial determination for the linear effect of weight is $R^2_{Y\,weight|cp,species,attack\_strong\_value,hp} = 0.0283$. One thing to note is that the AV plot does not really make sense for species since species is a factor variable.

Another benefit of an added variable plot is it allows us to determine influential points, after accounting for the other variables in the model.

Principal Components Analysis (PCA)

Principal components analysis is widely used as an unsupervised learning method for feature extraction and data compression. In our analysis, we will apply principal components in our regression model as a dimensionality reduction technique. The intuition behind PCA is: given a set of highly correlated predictors, PCA will transform it into a smaller set of linearly independent variables called principal components. The transformation is defined such that the first principal component direction captures the greatest possible variability in the data, in others words, explain the most variability of the data. The succeeding principal components are linear combinations of the variables that is un- correlated with the preceding component and has largest variance subject to this constraint. The set of components constitute a basis for our data space.

Principal Components Regression

The principal components regression approach will first construct M principal components and then regress on the components instead of individual predictors. The underlying assumption of the model is "the directions in which $X_1, ... X_p$ shows the greatest variance are those associated with Y". Although this assumption is not guaranteed, it regardless provides a decent approximation that often yields good results. M, the number of principal components, is our tuning parameter that will be chosen by cross-validation.

We believe PCA works well with our pokemon data given the existence of strong correlation among our predictors. Our analysis shows that PCR greatly reduced variance and contributed to our model's predictive power.
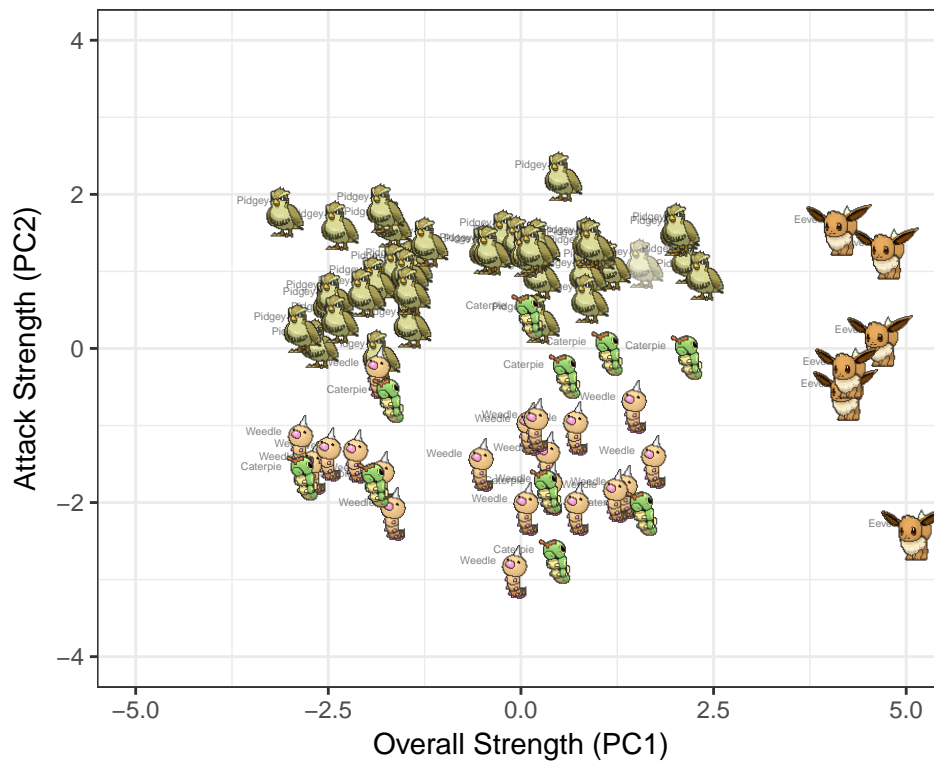
Principal Components

Our model first constructs 9 principal components(this makes sense since $p = 9$ and $M \leq p$).

To visualize it:

Look at PC1. We observe that the higher the performance metrics, the higher the PC1 value. Therefore we can interpret PC1 as a measurement of overall strength. As for PC2, we notice that higher PC2 is associated with higher attack value. Therefore we interpret PC2 as a measurement of attack strength.

We then plot our pokemon on our new principal components space:

Interesting Insights: Using principal components, we get to create two new powerful metrics to evaluate our pokemons. From the plot, you can observe that Eeeve in general has high overall strength and high attack strength. Pidgey has good attack stats but is weaker in general due to the creature's intrinsic nature. Caterpie and Weedle are weak on both metrics. Overall, this PCA gives you a high-level overview of our pokemon's strength. You should feel excited!

Regression:
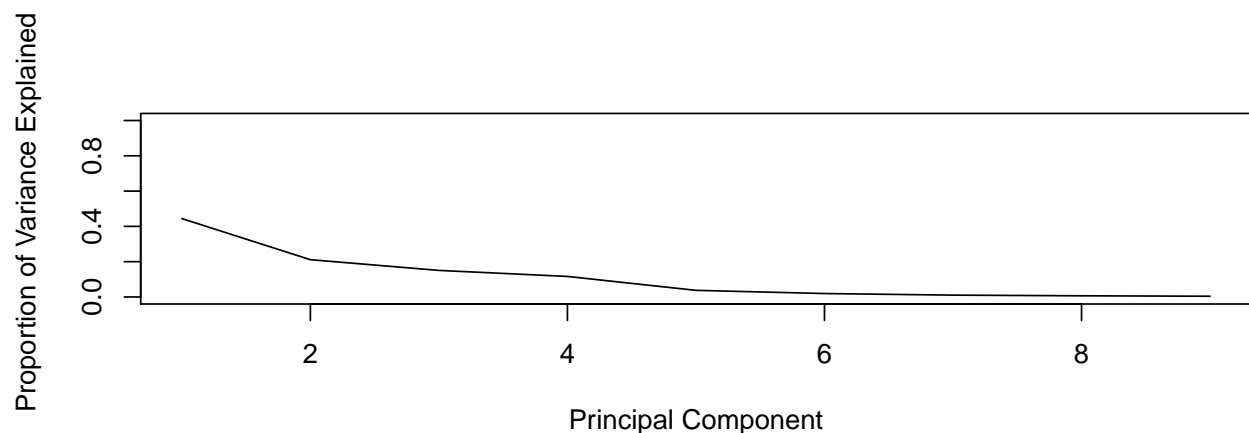
Observe that the first two principal components explain nearly 60% of the variability in the data. As $M \to 10$, the marginal contribution to variability explained decreases. Our regression model will use cross validation to tune M, the number of components as predictors.

```
## Data:     X dimension: 50 9
##  Y dimension: 50 1
```

```
## Fit method: svdpc
## Number of components considered: 9
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           397.1    224.9    203.8    185.4    182.3    213.3    213.4
## adjCV        397.1    223.8    202.3    184.0    179.9    209.4    208.5
##
##         7 comps  8 comps  9 comps
## CV          189    182.3    92.81
## adjCV       184    177.9    89.94
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X         45.77    65.32    80.62    91.87    96.18    98.04    98.98
## cpnew     70.69    79.25    84.37    87.68    88.05    90.30    93.22
##
##         8 comps  9 comps
## X         99.56   100.00
## cpnew     94.48    98.72
```

The metric of cross-validation is *root_mean_squared_error*. Ideally, we want to pick M where the CV score is minimized.
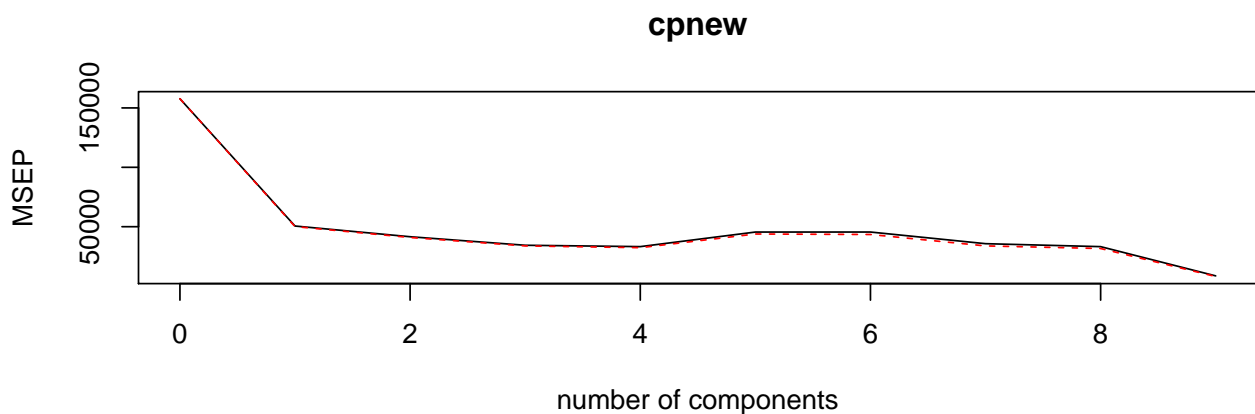
cpnew

We compute the test MSE as follows using $M = 9$ components:

```
pcr.pred = predict(pcr.fit, pokemondata[-train, ], ncomp = 9)
mean((pcr.pred - pokemondata[-train, ]$cpnew)^2)
```

```
## [1] 2827
```

Interpretation

Summary: How to get the best Pokemon?

Are you a Pokemon Go player? Have you been struggling with trying to identify the strong pokemon and winning your battles? Are you curious if your pokemon will be good enough after evolution?

These are the questions our report attempts to address. To kick start, we ran a comprehensive overview of the data at hand.
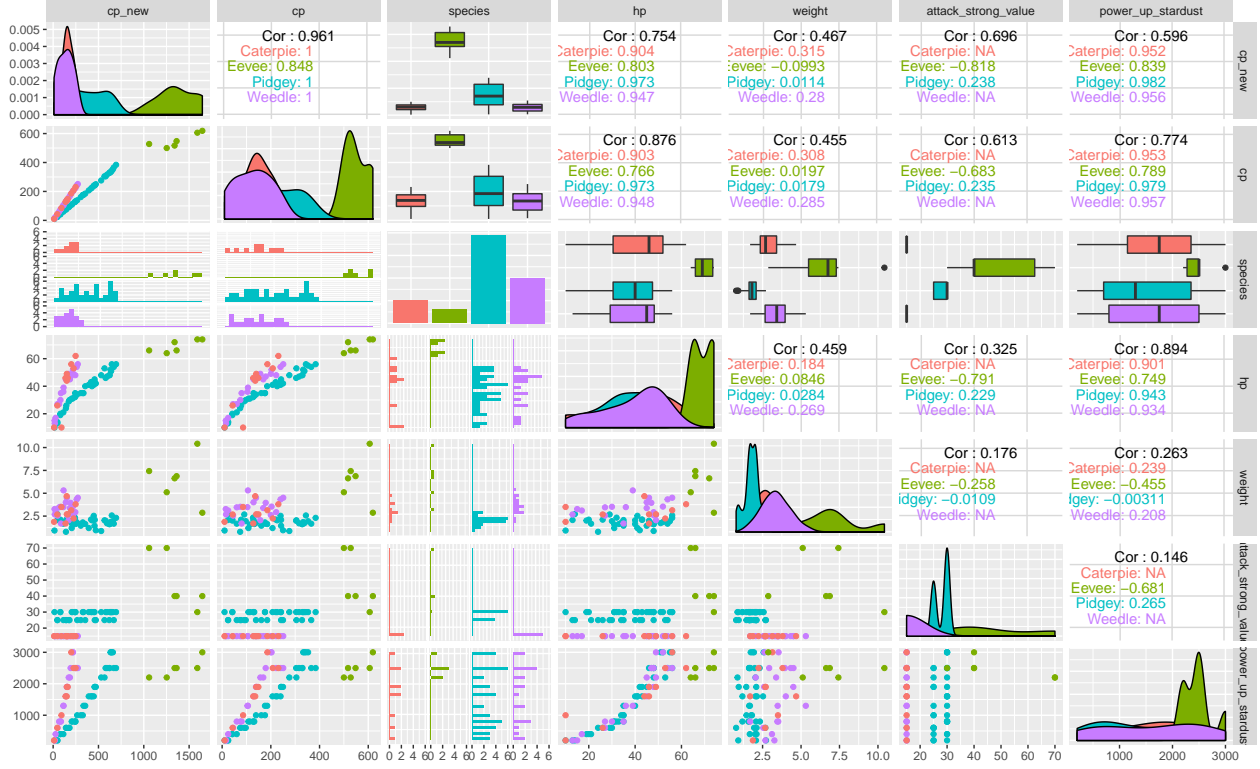
We observed that our response variable *cp_new* is strongly correlated with *cp*, indicating that *cp* might be a promising predictor. We also noticed the existence of correlation between pairs of explnatory variables and we kept that in mind in our model building process.

After substantial analysis, we decided to present a linear regression model using three of the most important features that will help users predict their pokemon's combat power post-evolution. To reiterate, since our data only includes four species, our inference only applies to the population of these four species. We used LASSO (Least Absolute Shrinkage and Selection Operator) to build our model. The decision is motivated by 1) high R^2 2) high interpretability 3) our residual plot and 4) correspondence with our PCA analysis that complements the LASSO model. The LASSO model works well with our data because our variables are predominantly quantitative and the model's feature selection capacity helps users understand what the characteristics are of a strong pokemon.

Our model is: $E[\log_{cp\_new}] = 0.64 + 0.89 \log cp + 0.012 attack\_weak + 0.0081 attack\_strong$. The cross-validated R^2 is 0.984 which means the model explains 98.4% of the variability in $\log(cp\_new)$. This model corresponds with our conclusion from PCA whose top two principal components indicate overall strength (cp) and attack strength ($attack\_strong$ and $attack\_weak$). Essentially, this means that what users should be primarily concerned with when determining whether their Pokemon is inherently strong or not is their base combat power prior to evolution as well its move strength. The residual plot further shows that 1) linearity is met and 2) there is a good scatter around the 0 line indicating constant variance. We do have influential points that correspond to the Eeeves but we decided not to remove them because we believe our users are equally interested in the performance of Eeeves as well. Overall, the technical conditions are met and we are comfortable with inference on the following four Pokemon: Caterpie, Weedle, Pidgey and Eevee.

residual plot
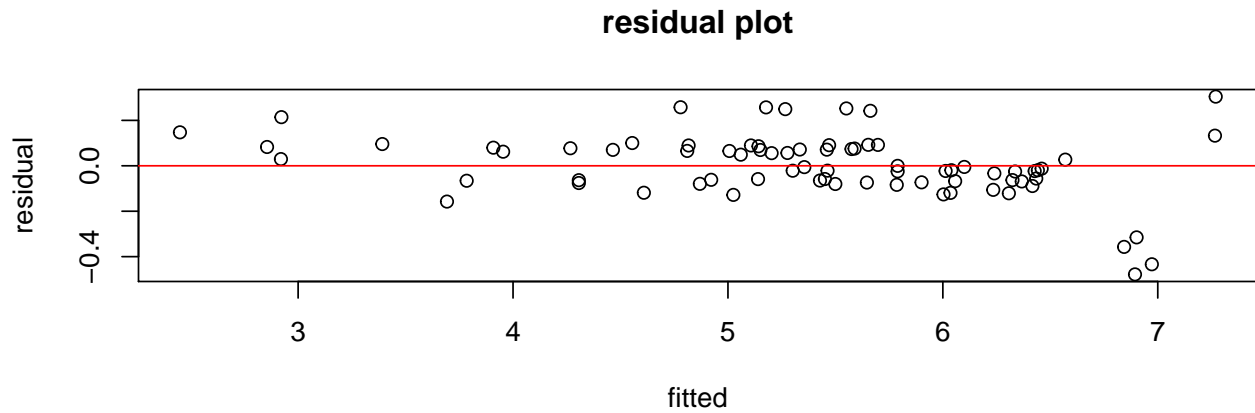
Future Direction

Overall, the most interesting insight that surprised us was the fact that cp_{new} could be explained primarily by cp, or combat power prior to evolution and attack value (both strong and weak). We did not expect this given that our exploratory data analysis did not indicate that attack value was highly correlated with cp_{new}. If we had more data of different species of Pokemon we would have liked to explore whether cp and species alone can predict cp_{new} since in our correlation plot above, there is almost a perfect correlation between cp and cp_{new}. Further, having more data on differen Pokemon species would have allowed our analysis to be inferred onto many many more Pokemon that a Pokemon Go player might want to know about. We also wish that there was data on battle statistics as well such as combat damage dealt to see whether that could factor in to explaining a Pokemon's strength.