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

*Alex Gui and Lathan Liou*

*Date Submitted: May 9, 2018*

## Introduction

In July 2016, Pokemon Go became an overnight sensation with hundreds of millions of people downloading the mobile game. For most players, this game represents a nice coffee break. However, for more serious players, the objective is to try to obtain the strongest Pokemon possible available, which is indicated by the highest combat power, abbreviated cp, so that you can battle other players' pokemon and win. Usually players can catch weaker forms of Pokemon that, through training, can evolve into stronger forms, so an evolved Pokemon will generally always have a higher cp than a non-evolved Pokemon. In the search for the strongest Pokemon, players have wished to determine what characteristics would indicate that the pokemon they have is stronger relative to other players' pokemon. For instance, if a player had pokemon X and trained it to its maximum potential but pokemon X was still weaker than pokemon Y, the player would want to know why pokemon Y was still stronger. 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.

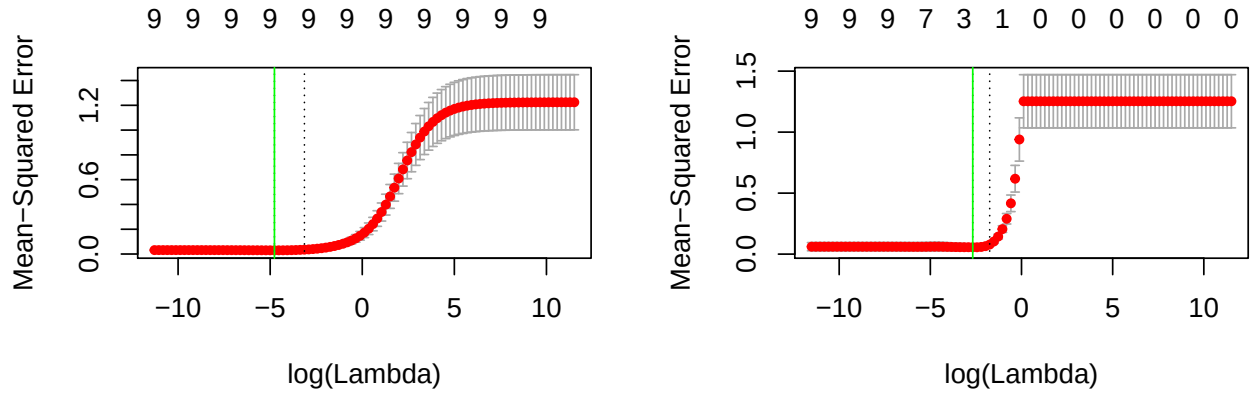
The dataset we are using to build our model is an original data set collected by *OpenIntro*[1]. The dataset contains 75 observations across 26 variables, with each observation representing a randomly generated Pokemon that the gamer caught. Four species are represented in this data, so the conclusions drawn from this modeling process can only be inferred onto the population of these 4 particular species: Eevee, Pidgey, Caterpie, and Weedle.

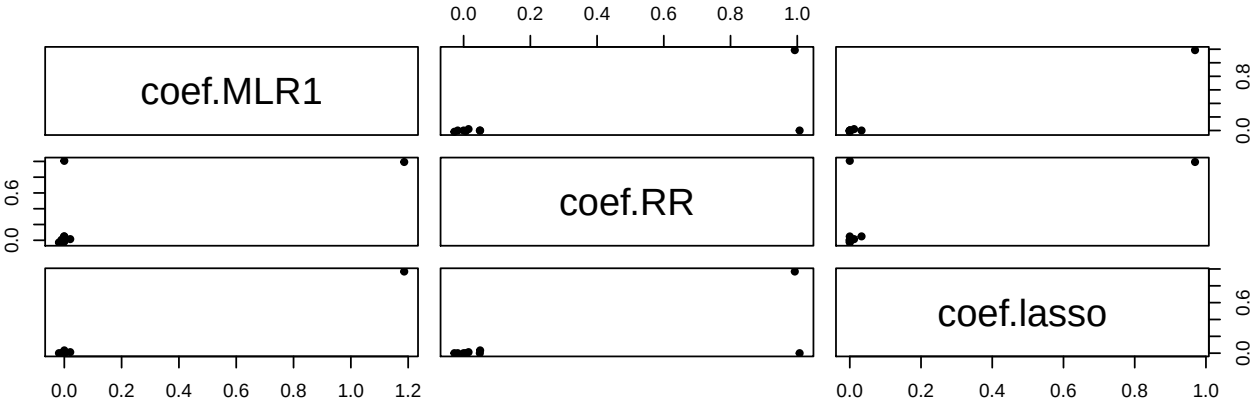
We avoid using "new" predictor variables in our modeling process because as a user, you wouldn't have access to any of the "new" information, but if you're interested in whether your pokemon will evolve into one with a high cp (*cp_new*), you would want to know which of the Pokemon's current stats could indicate a high *cp_new*. We also avoid categorical variables such as the name of the attack and attack type because they don't inherently contain any information that users would be concerned with; it's the attack value that matters. In our exploratory analysis, we also log-transformed cp and *cp_new* because we had noticed issues of nonconstant variance in the residual plot.

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

## Ridge Regression and Lasso

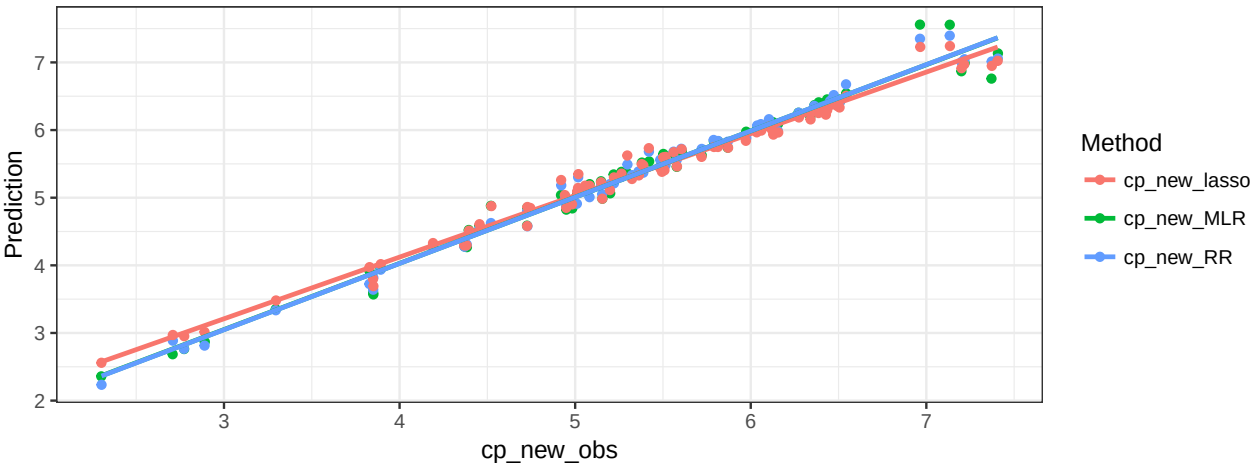
We ran ridge regression and LASSO on our explanatory variables of interest.



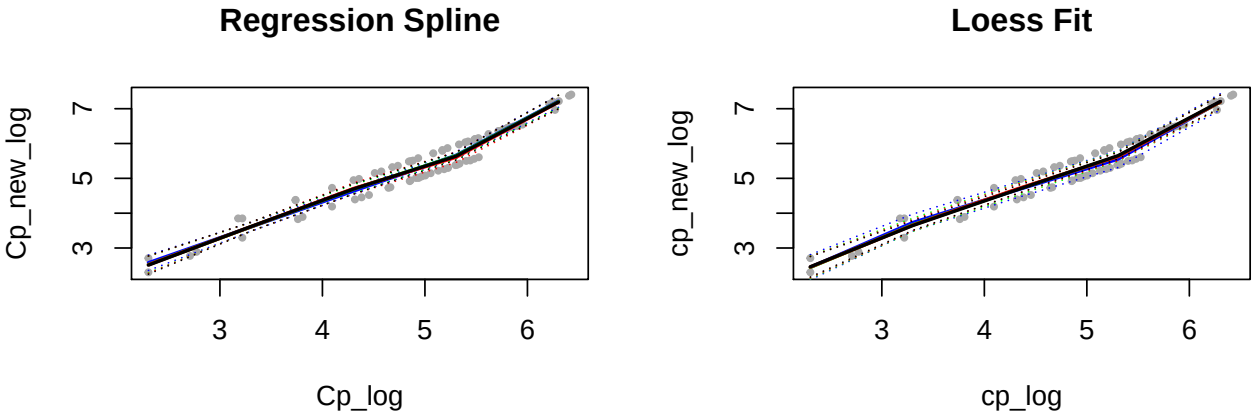

As shown in the pairs plot, ridge regression shrunk the coefficients closer to zero than multiple linear regression did. On the other hand, lasso regression not only shrunk the coefficients but also performed variable selection more selectively than our stepwise regression did previously, selecting *cp_log*, *attack_strong_value* and *attack_weak_value* at our optimal lambda value.

Next, we wanted to see how the predictions from our multiple linear regression, ridge regression, and our LASSO models would fare against each other.




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




We chose \$cp_{log} to run our smoothing spline and the loess regressions. The smoothing splines and loess fit the data extremely well. Changing the degrees of freedom, and hence the number of knots, for the smoothing splines improves the fit minimally. At a certain point, increasing the degrees of freedom actually begins to increase SSE. Increasing the span from 0.2 for loess actually increases SSE.

The spline and loess models are all extremely similar due to the nature of the data being perfectly correlated, but I would choose the spline model since it fits the data very smoothly, and it still has a functional form which lends itself to interpretability.

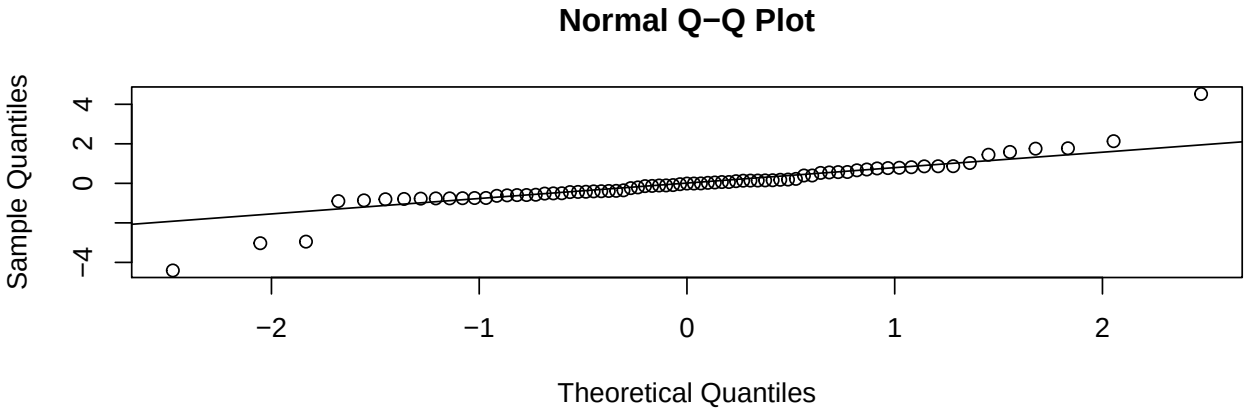
### Conclusion

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

\subsection*{Q-Q Plot^2}

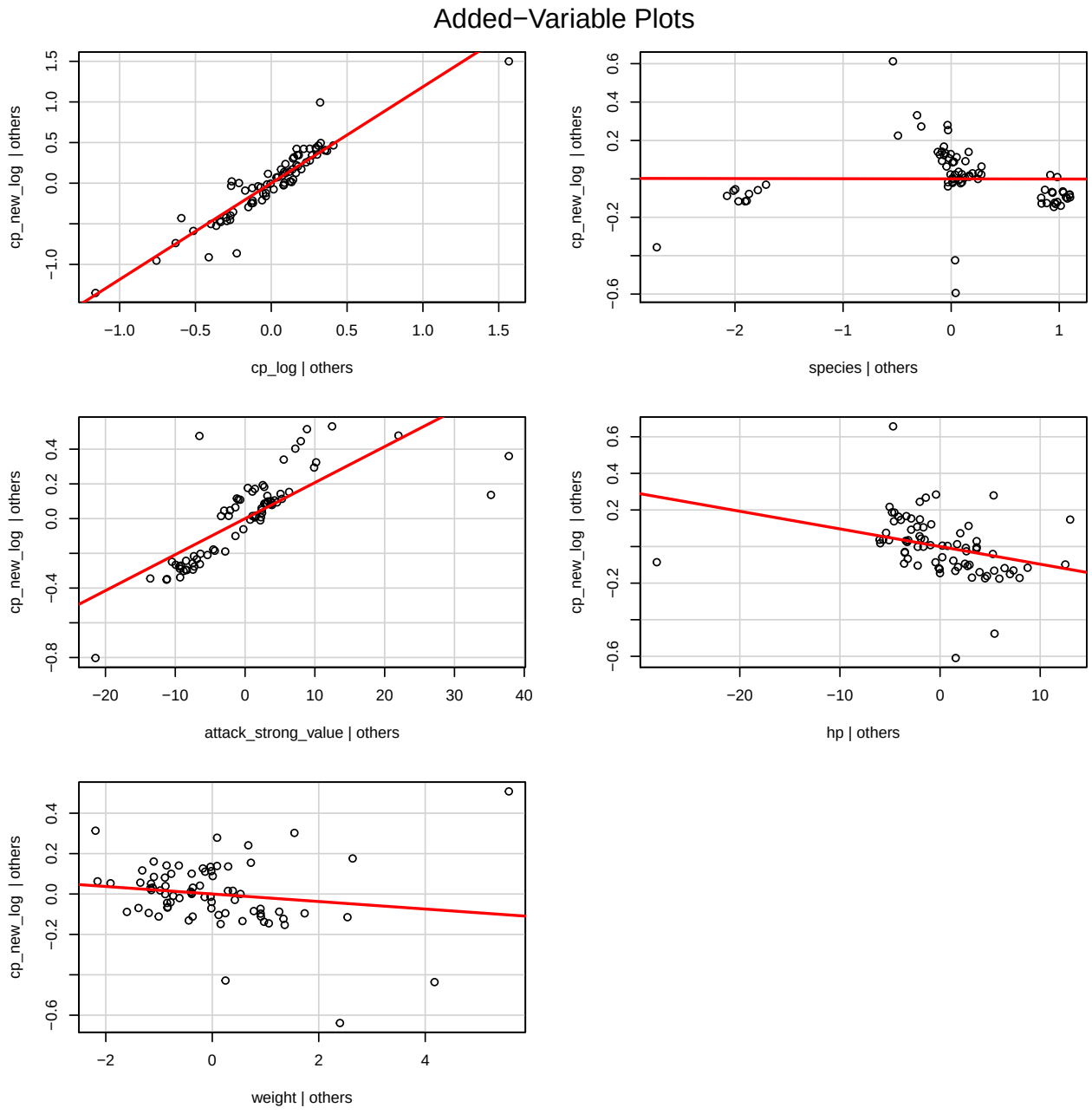
A normal probability plot is used to identify substantial departures from normality in the data. We chose in particular to plot what is known as a normal quantile-quantile plot (Q-Q plot for short), which plots sample quantiles ordered and plotted in a continuous cumulative distribution function against 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 information about whether the normality technical condition of the residuals is violated.




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.

\subsection*{Added Variable Plots^3}

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 $X_{i.[i]}$ The residuals from $Y_{\cdot[i]}$ and $X_{i.[i]}$ 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.




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 $R^2_{Yweight|cp,species,attack\_strong\_value,hp} = 0.0283$. 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 Regression (PCR)

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 other words, explains the greatest 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 constitutes a basis for our data space.

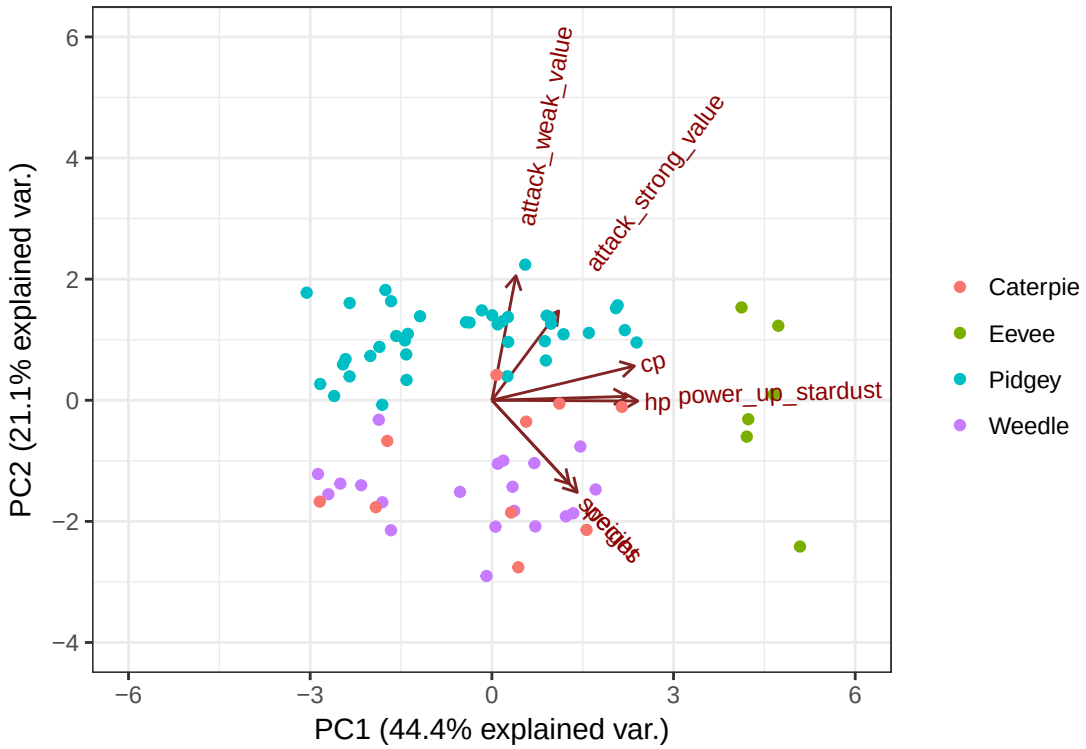
The principal components regression approach will first construct $M$ principal components and then regress on the components instead of individual predictors. The underlying assumption of the model is "the directions in which $X_1, ...X_p$ shows the greatest variance are those associated with Y" (ISLR). 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 PCR works well with our pokemon data given the existence of strong correlation among our predictors (check out our correlation plot down below!).

## Principal Components

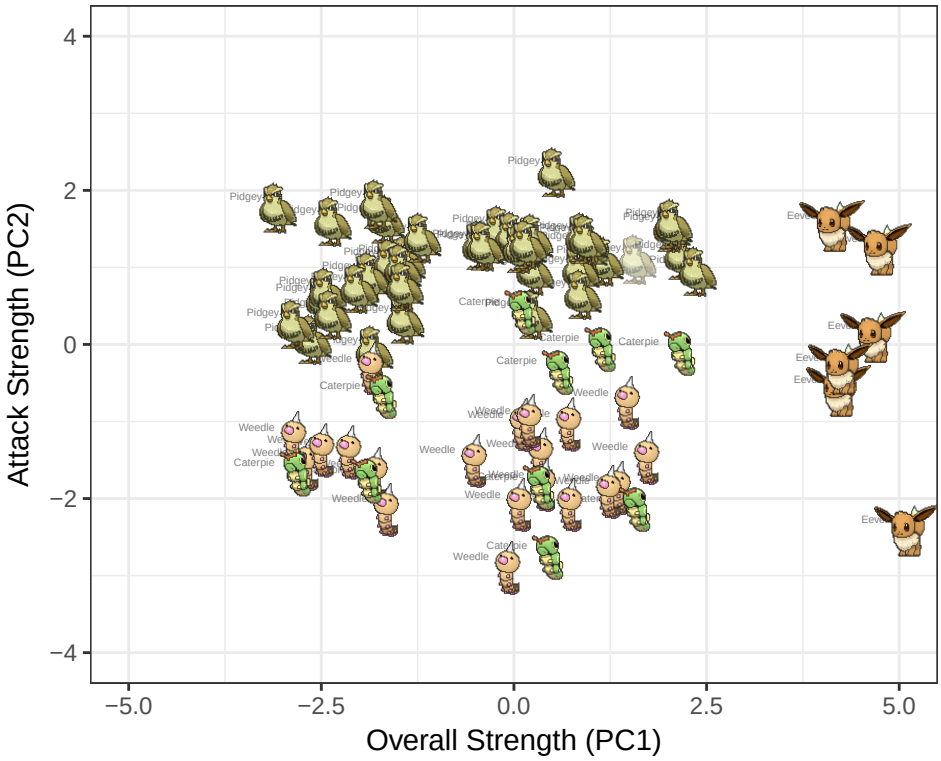
Our model first constructs 9 principal components(this makes sense since $p = 9$ and $M \leq p$).

To visualize it (we only picked a subset of the variables to avoid over-crowding the plot):



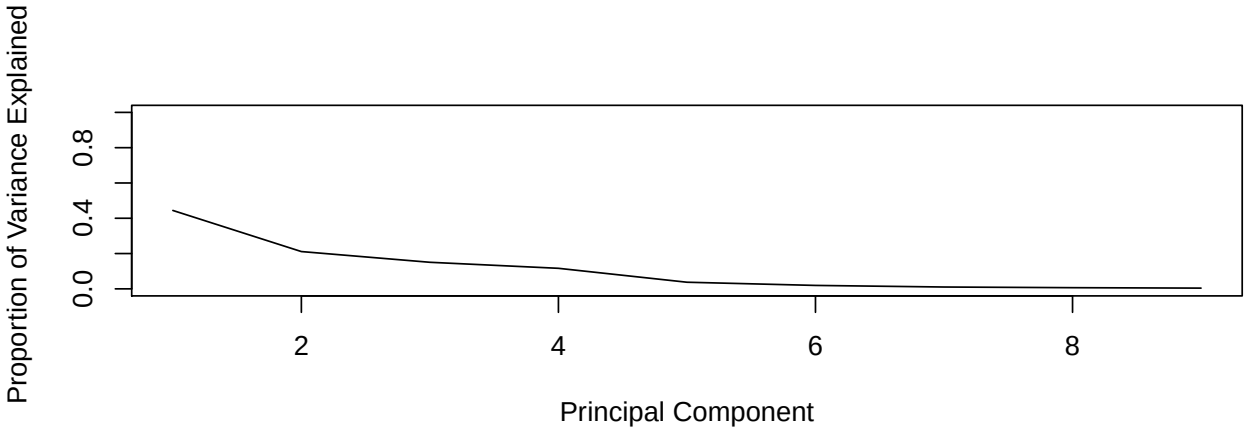

Let's look at PC1. We observe that the higher the performance metrics, the higher the PC1 value. Therefore we can interpret PC1 as a measurement of overall strength. As for PC2, we notice that higher PC2 is associated with higher attack value. Therefore we interpret PC2 as a measurement of attack strength.

We then plot our pokemon on our PC1 and PC2 space:



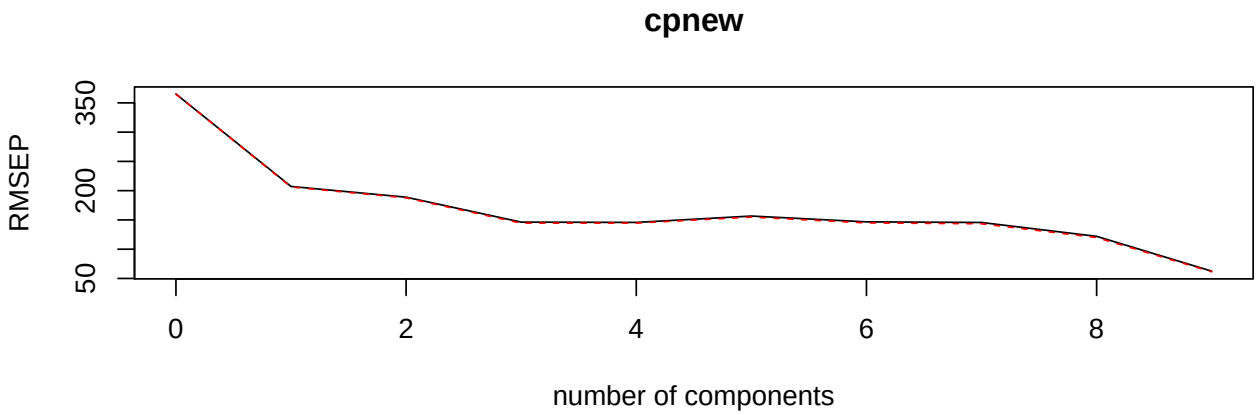

**Interesting Insights:** Using principal components, we identify two new powerful metrics to evaluate our pokemons. From the plot, you can observe that Eeeve in general has high overall strength and high attack strength. Pidgey has good attack strength but is weaker that Eevees in general. Caterpie and Weedle are weak on both metrics. Overall, this PCA gives you a high-level overview of our pokemon's strength. If you own a Eevee, you should feel excited about evolving it because you will probably get a really strong evolution of Eevee!

## Regression




Observe that the first two principal components explain more than 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.

The cross-validation uses *root_mean_squared_error* as the metric. $M = 9$ returns the smallest cv score.




Therefore we used all the components to build the linear regression model. Our result shows that all regression coefficients are signigicant and the adjusted $R^2 = 0.983$.

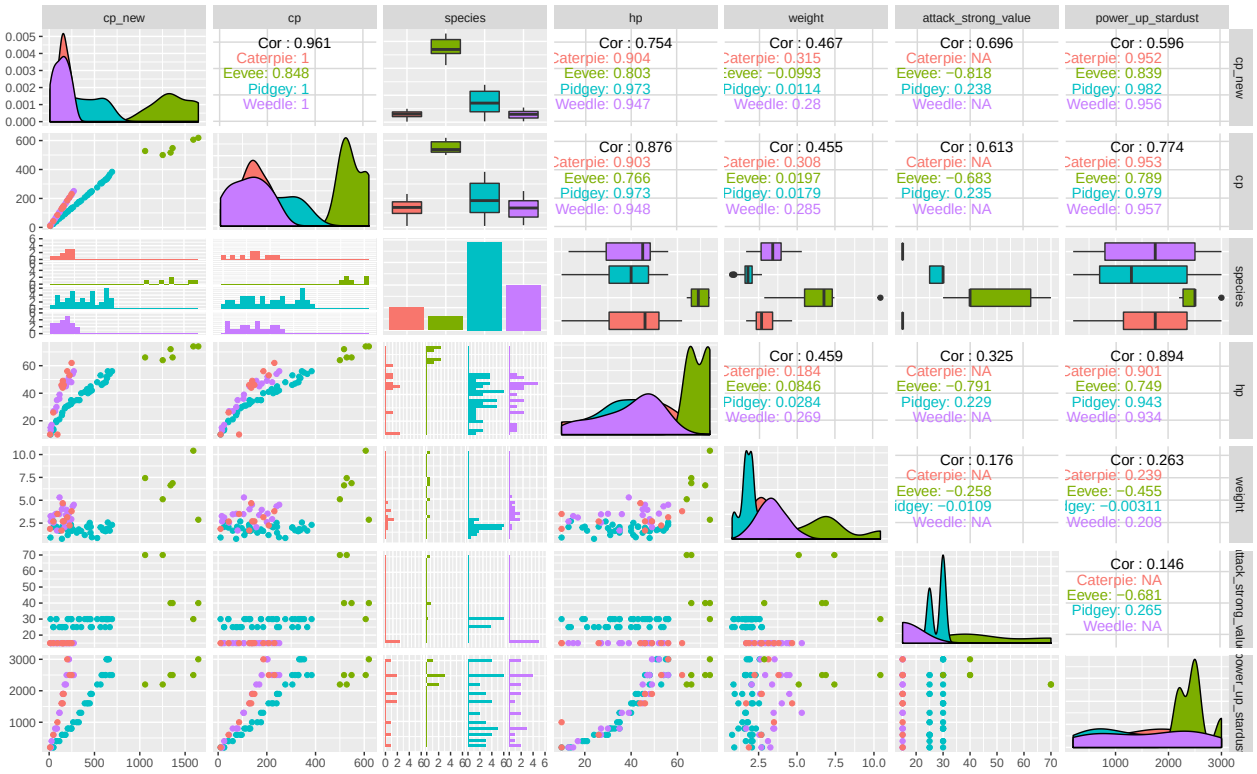
### Interpretation

Our principal component regression model performs really well in predicting *cp_new*. We also observe that our top two principal components (overall strength, attack strength) correspond to the predictors selected by LASSO (cp,attack_strong and attak_weak). However, we decide not to go with PCR model because it is not interpretable.

## Summary: How to get the best Pokemon?

Are you a Pokemon Go player? Have you been struggling trying to identify whether your Pokemon are strong and how to win your battles? Are you curious if your Pokemon will be strong enough after evolution?

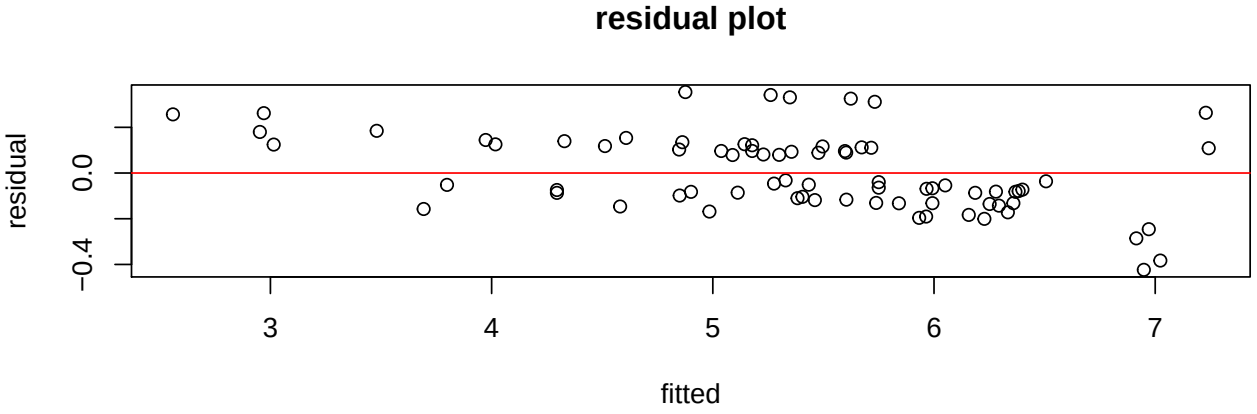
These are the questions our report attempts to address. The first thing we will show you is a comprehensive overview of all the variables that you could possibly consider as a Pokemon Go player.




We observed that our response variable *cp_new* is strongly correlated with *cp*, indicating that *cp* might be a promising predictor.

After substantial analysis, we decided to present a linear regression model using three of the most important features that will help users like yourself predict their Pokemons' 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 distills what the important characteristics are of a strong pokemon for users. One drawback is that we cannot obtain p-values of tests of significance because LASSO does not have a closed form tht allows us to calculate variance easily.

Our final 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 $\log_{cp\_new}$. The residual plot further shows that 1) linearity is met and 2) there is a good scatter around the 0 line indicating constant variance.



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.** 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.

# Future Directions

Overall, the most interesting insight we obtained was the fact that $cp_{new}$ could be well explained by attack value (both strong and weak). We did not expect this since 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 different 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. Ultimately, we built a clean model that has high predictive power and that very clearly indicates what things a Pokemon Go player should pay attention to. We beleive with more data, we can extend our analysis such that we can inform users about the ins-and-outs of every Pokemon in the Pokemon Go universe.

# Sources


- OpenIntro (https://www.openintro.org/stat/data/?data=pokemon). This is where downloaded the csv for our data.
- baptiste (https://stackoverflow.com/questions/30299529/ggplot2-define-plot-layout-with-grid-arrange-as-argument-of-do-call). How we make nicely arranged graphs.
- sape research group (http://sape.inf.usi.ch/quick-reference/ggplot2/colour) Colors for ggplot are great
- Marc Böttinger (https://stats.stackexchange.com/questions/266592/how-to-calculate-r2-for-lasso-glmnet?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa) neat way of finding CV R-squared of lasso regression
- Online Stat book (http://onlinestatbook.com/2/advanced_graphs/q-q_plots.html) Info on QQplots
- Silverfish (https://stats.stackexchange.com/questions/125561/what-does-an-added-variable-plot-partial-regression-plot-explain-in-a-multiple?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa) Great explanation of Added Variable plots
- ISLR, PCA
- Ahn Le (http://people.duke.edu/ aql3/gotta-plot-them-all/) inspiration for awesome way to use ggplot to incorporate sprites