Project 4 158: Beyond Lineary / Something New / Summary

Tesfa Asmara and Kevin Loun

5/10/2022

Contents

1 Introduction

The dataset for this project contains 10,000 League of Legends ranked matches from the North American region with 775 variables offered through the Riot Games API, provided on Kaggle (Games 2021a)(James 2020). Each match is pulled from players who rank Gold in the League system, a ranking system that matches players of a similar skill level to play with and against each other. Amongst North American players, the Gold skill level was the second most common tier, achieved by 27.7 percent of players, or approximately 49.86 million players when considered against Riot Games' player base of 180 million ("Distribution of League of Legends (Lol) Summoners in North America as of October 2021, by Tier" 2021)(Games 2021b). This dataset will be referred to as `lol10`.

For this project, the following variables are of interest: time spent crowd controlling others, map side, longest time spent living, kills, gold earned, and total damage dealt. A figure including all the relevant variables and their description is attached at the end.

2 Shrinkage & Smoothing Models

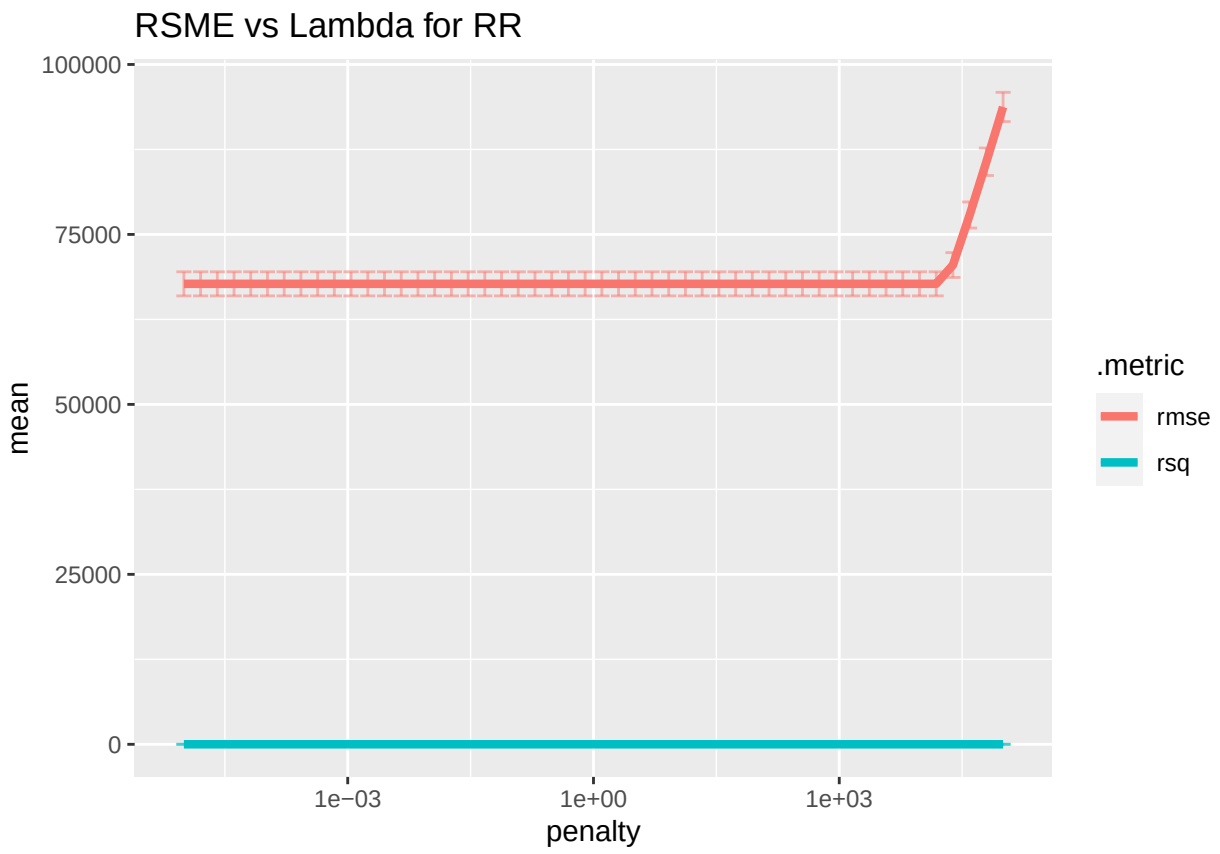
2.1 Normalizing Data

Since we are running a Ridge Regression and LASSO model on our data we need to ensure that our data is normalized to ensure that all variables contribute equally to the penalized coefficients in our models.

2.2 Ridge Regression

Ridge Regression optimization provides a trade-off between two different criteria: variance and bias.It seeks to find coefficients that minimize the SSE of the data set. Ridge Regression attempts to shink the coeffcients in our model close to zero but does not actually remove any coeffcients. This is done using a tuning parameter $\lambda > 0$. To develop a Ridge Regression model for our data we created a recipe that normalized our data and then used cross validation to find the penalty, λ value that would best minimize the SSE of our data. After using cross validatation we found that the value of λ that best minimized the SSE and our coefficients was 0.000001. The plot below showcases how our MSE and R^2 value change as a function of λ and our final ridge regression model can also be found below.

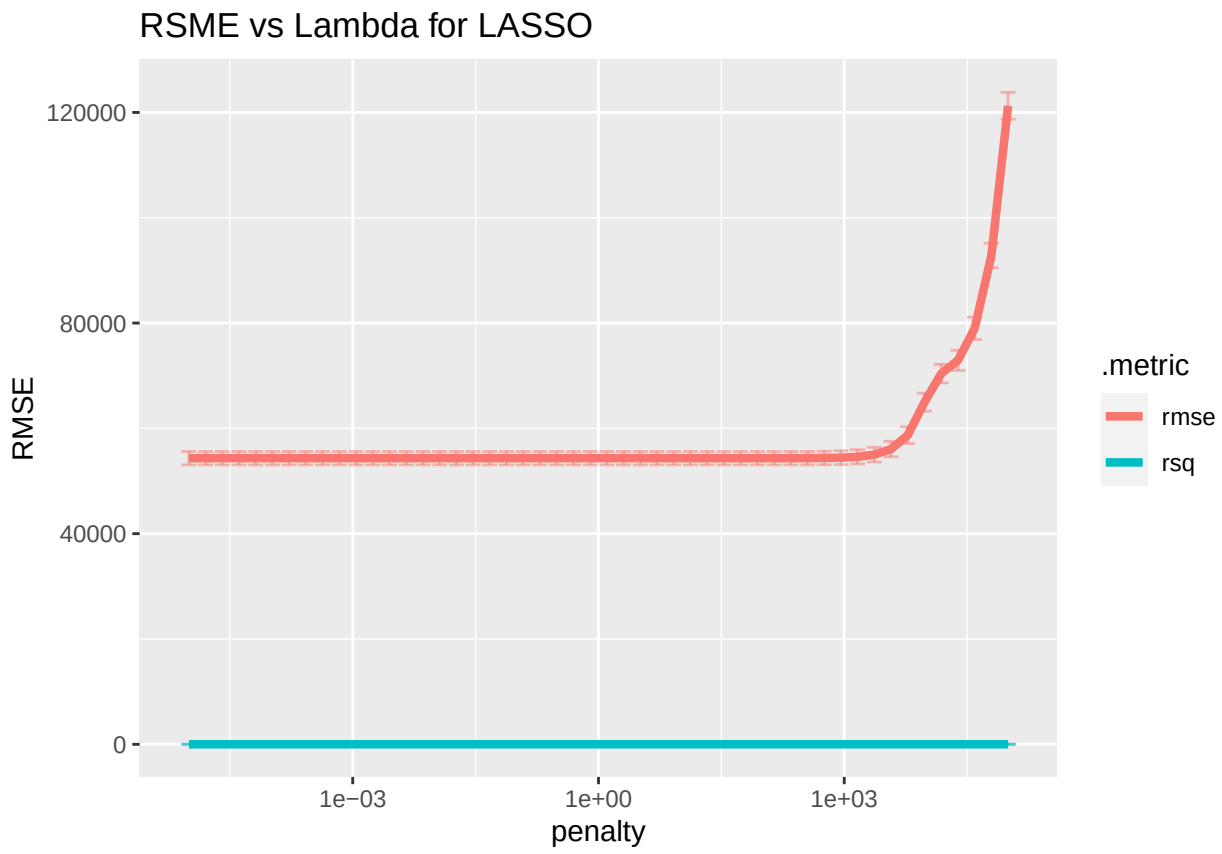
```
## # A tibble: 5 x 3
##   term                         estimate penalty
##   <chr>                           <dbl>   <dbl>
## 1 (Intercept)                   551340. 0.00001
## 2 b_gold_earned                 180281. 0.00001
## 3 b_kills                       -17192. 0.00001
## 4 b_time_c_cing_others           16638. 0.00001
## 5 b_longest_time_spent_living   15542. 0.00001
```



2.3 LASSO Regression

Similar to Ridge Regression, LASSO optimization provides a trade-off between two different criteria: variance and bias.It seeks to find coefficients that minimize the SSE of the data set. LASSO attempts to shink the coeffcients in our model to zero but does not keep all predictors, only important ones. This is also done using a tuning parameter $\lambda > 0$. To develop a LASSO model for our data we created a recipe that normalized our data and then used cross validation to find the penalty, λ value that would best minimize the SSE of our data. After using cross validatation we found that the value of λ that best minimized the SSE and our coefficients was also 0.000001. The plot below showcases how our MSE and R^2 value change as a function of λ and our final ridge regression model can also be found below.

```
## # A tibble: 5 x 3
##   term                        estimate penalty
##   <chr>                          <dbl>   <dbl>
## 1 (Intercept)                  551340. 0.00001
## 2 b_gold_earned                269336. 0.00001
## 3 b_kills                      -79530. 0.00001
## 4 b_time_c_cing_others              0  0.00001
## 5 b_longest_time_spent_living   -659. 0.00001
```

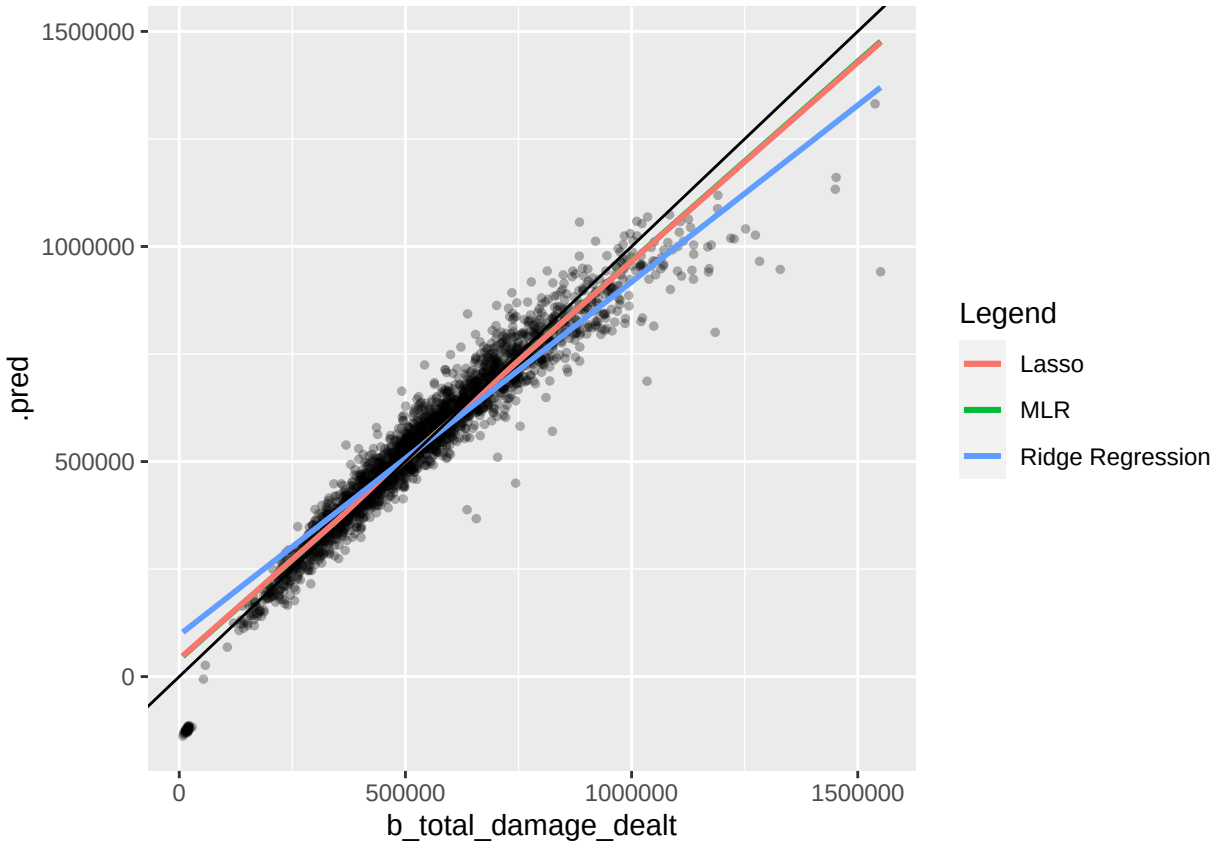


2.4 Comparing Models

When comparing MLR, RR, and LASSO we can look at the coefficients of the individual models. In the LASSO Model the coeffcient of `b_time_c_cing_others` has been penalized to 0 while in MLR it is a negative value with and in RR it is a large coefficient. In the MLR and LASSO model `b_kills` has a relatively large coefficient while it becomes more penalized in the RR model. Finally, the coefficient estimate for `b_longest_time_spent_living` in MLR and LASSO are relatively small compared to RR which is odd because of how RR and LASSO penalized the coefficient differently despite using the same λ value. The remaining variables appear to remain similar throughout the three models.

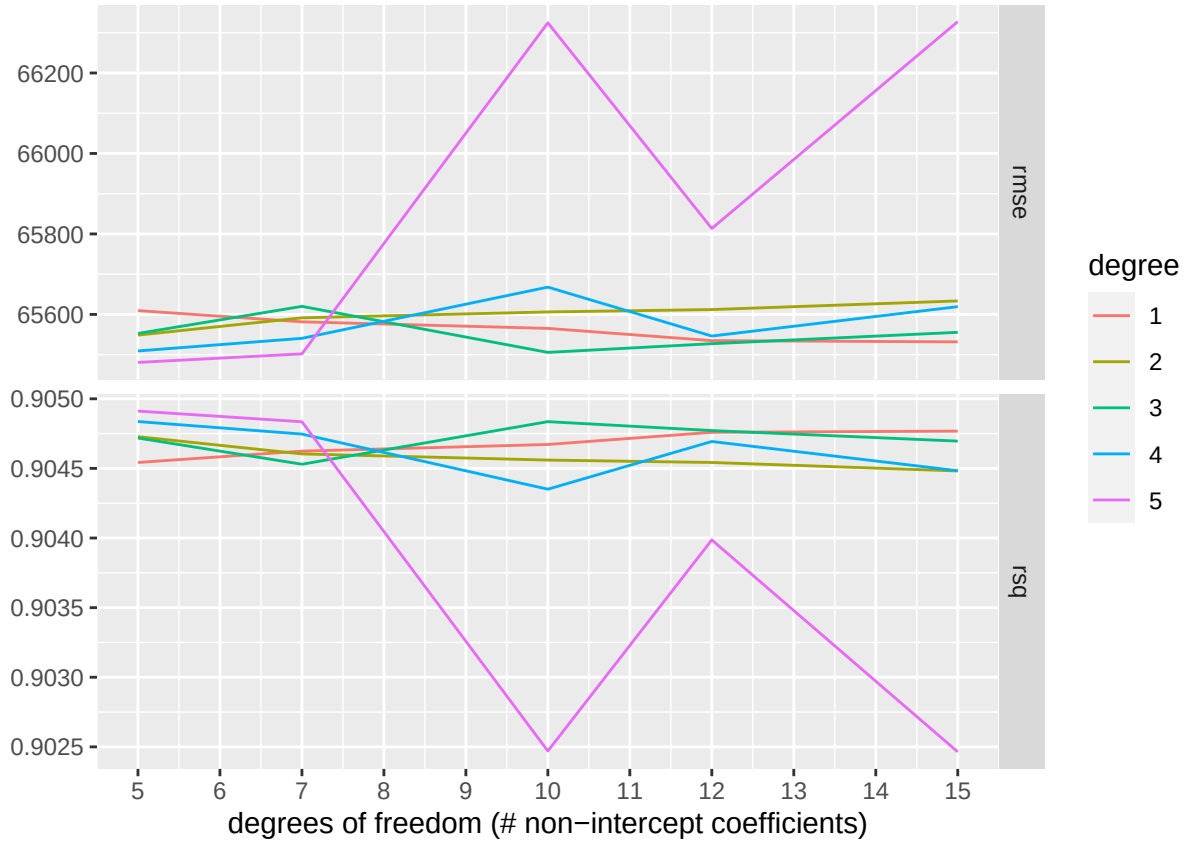
2.5 Plotting Predicted vs Actual for 3 Models

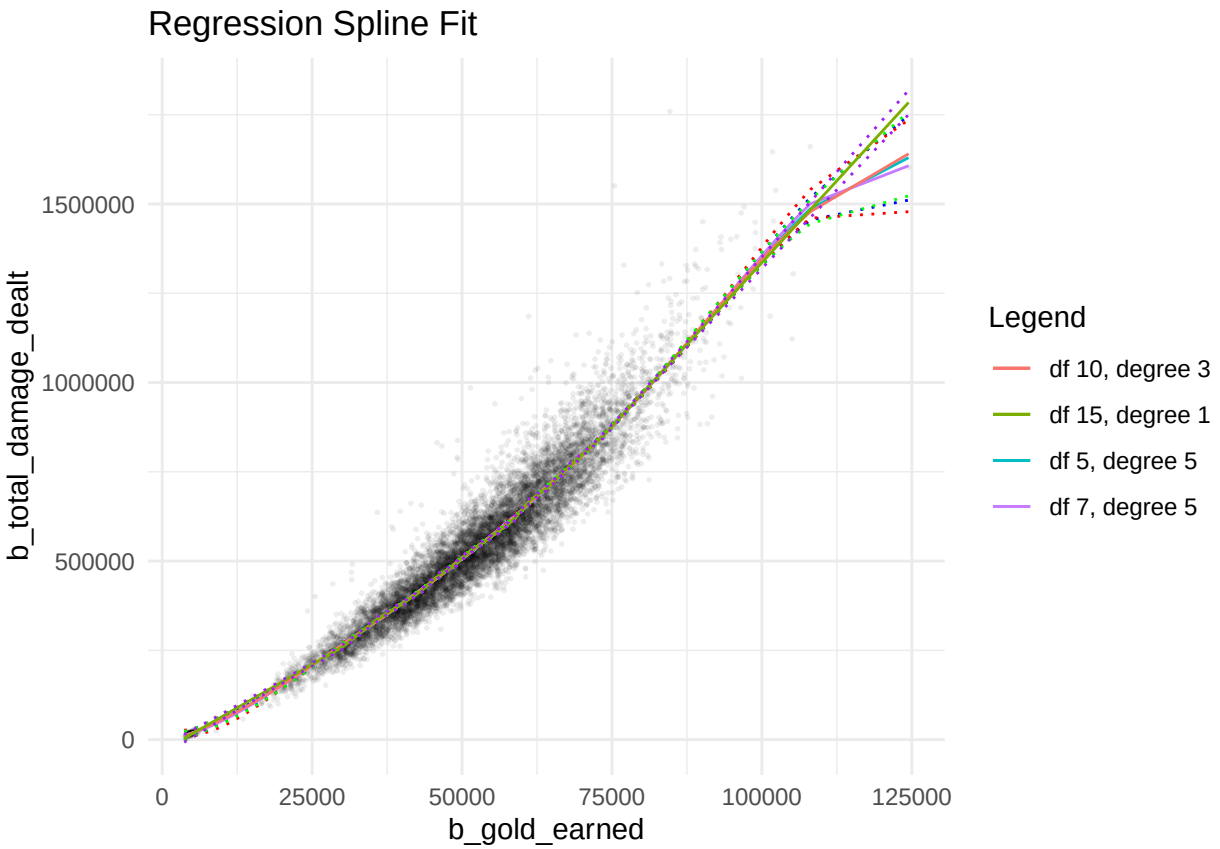
When comparing the predictions of the MLR, Ridge Regression, and LASSO models visually we can see that the MLR and LASSO model produce predictions that are closest to the actual value of the test set while the Ridge Regression model is the least accurate. However, it should be noted that this could be due to overfitting of the models, so while the models may appear to have more predictive accuracy they may not have the same accuracy on new data.



2.6 Regression Spline

We chose to apply the Regression Spline smoothing technique on `b_gold_earned` plotted against `b_total_damage_dealt`, where we earlier observed the nonlinearity of the regression model and the nonconstancy of the error terms. We select the combinations of degrees of freedom and degree based on their having a high coefficient of determination and low MSE.

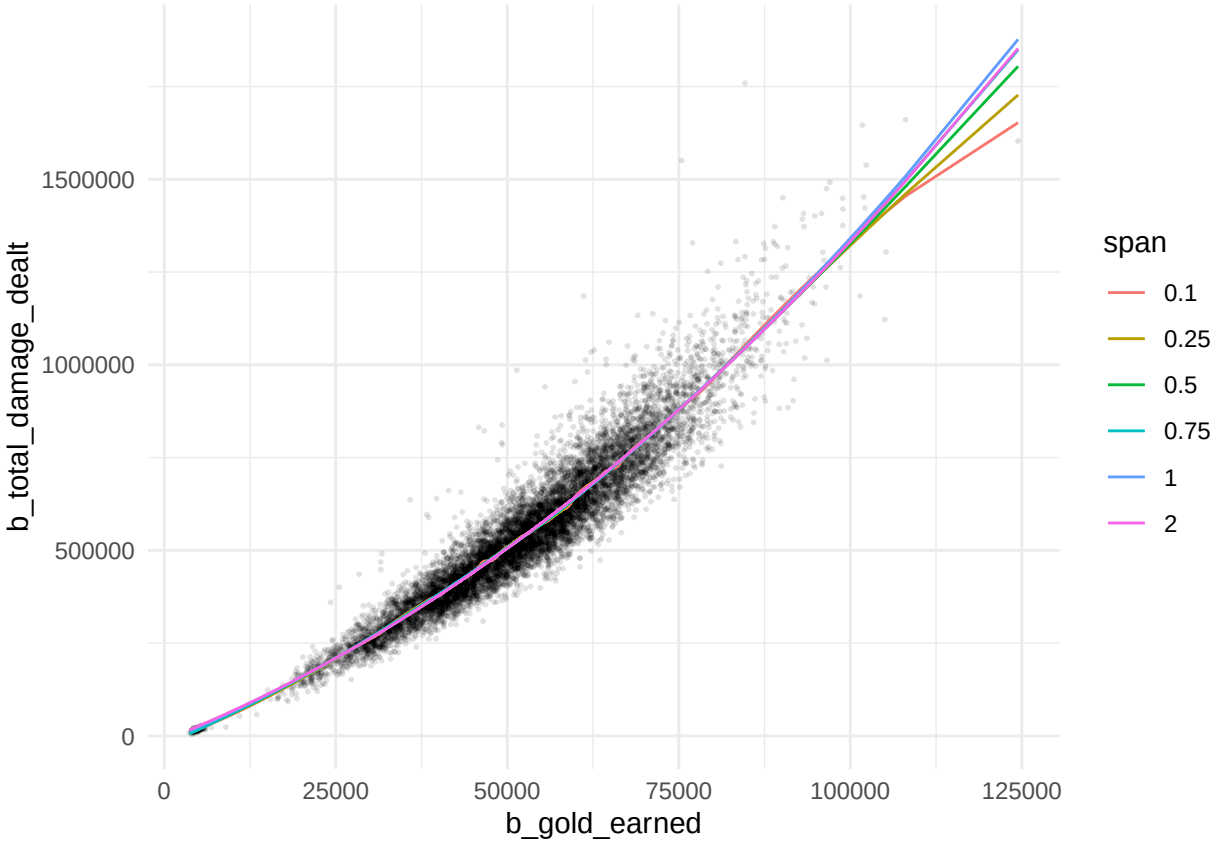

66200
66000
65800
65600
rmse
0.9050
0.9045
0.9040
0.9035
0.9030
0.9025
rsq
5
6
7
8
9
10
11
12
13
14
15
degrees of freedom (# non−intercept coefficients)
degree
1
2
3
4
5



In the graph, the average difference between the upper standard error and lower standard error is the smallest for the blue line. This corresponds to employing 5 degrees and 5 knots. So, for our comparisons to the loess smoothing technique, we will use the 5 degrees and 5 knots Regression Spline.

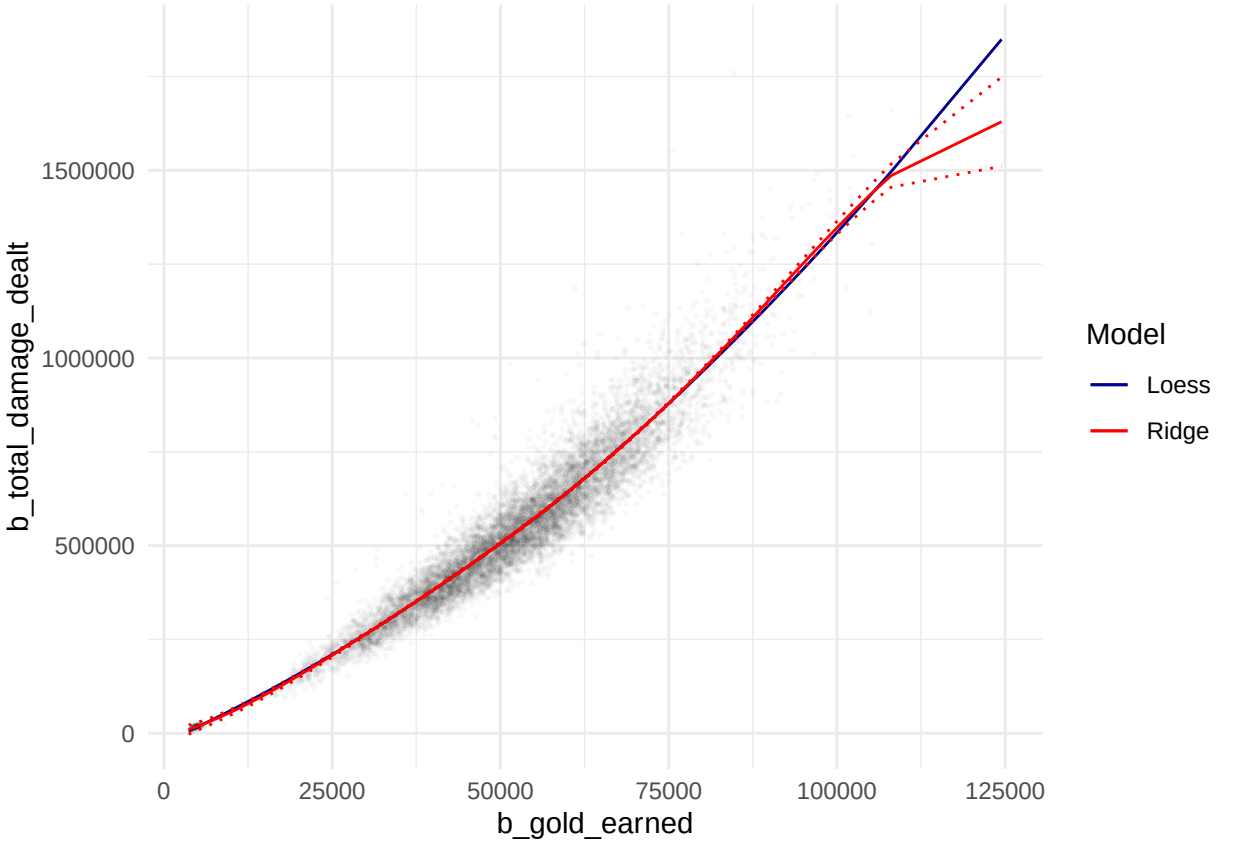
2.7 Loess

In our Loess model we chose to run a model on `b_gold_earned` and `b_total_damage_dealt`. Since Loess models use span as a parameter to determine weights of points, we choose to use six span values to create six different models. Our span values are .1, .25, 0.50, 0.75, 1.00, and 2.00. Below is a plot of our six models and visually it appears that the best model has a span of .75 because the other models appear to be affected by outlier points or are indifferent to them. The curves with $span = 0.1, 0.2$ overfit the points. The curve with $span = 2$ looks to have high bias. Similarly, the curve with $span = 1$ seems to not capture the relationship. The differences between the curves with $span = 0.5, 0.75$ are considered to not be significant.



2.8 Best Smooth Model

Now, as aforementioned, the primary goal of our model is to better understand the total damage dealt for the average Gold-ranked player on the blue team. As such, we care more about coefficient estimates and their interpretation over predictive power. The Regression Spline smoothing technique does provides a functional model, whereas loess does not. Therefore, the Regression Spline model would be better than the Loess model for our particular research question.



2.9 Conclusion

After building or MLR model, we began to look at our data through Shrinkage and Smoothing methods, as such we built and fitted our data to Ridge Regression, LASSO, Regression Spline, and Loess Models. We found that interestingly in our LASSO and Ridge Regression models our MSE did not change much with a change in our penalty value until large values of λ which interested us because it made us wonder if this was simply due to our data or if it was because we had mutated our variables. When comparing MLR, LASSO, and Ridge Regression we found that there were significant differences in the coefficients estimated by the three models, but when compared visually MLR and LASSO had very similar predictive accuracy while Ridge Regression appeared the least accurate. When looking at our Smoothing models individually, we found that our Ridge Rrgression model using degrees of freedom 6 and Loess model uwing a span of .6 to be the best models for our data. Based on our results we would ask the question: would the results of our Shrinkage models be different if we had not mutated our data but instead used our initial data, and would it have been different if we had used more data from the initial data set?

3 Bayesian Inference for Simple Linear Regression (Bolstad and Curran 2016)

3.1 Bayes' Theorem for the Regression Model

While a frequentist assumes that there are true values of the parameters of the model and computes the point estimates of the parameters, a Bayesian asserts that only data are real, and treats the model parameters as probability distributions which are to be inferred. Bayes' theorem is summarized by

$$posterior \propto prior \times likelihood,$$

so we need to determine the likelihood and decide on our prior for this model.

3.2 The Joint Prior for β and $\alpha_{\bar{x}}$

Using the alternate parameterization we obtain

$$y_i = \alpha_{\bar{x}} + \beta(x_i - \bar{x}) + e_i$$

where $\alpha_{\bar{x}}$ is the mean value for y given $x = \bar{x}$, and β is the slope.Each e_i is normally distributed with mean 0 and known variance σ^2. The e_i are all independent of each other. Therefore $y_i \mid x_i$ is normally distributed with mean $\alpha_{\bar{x}} + \beta(x_i - \bar{x})$ and variance σ^2 and all the $y_i \mid x_i$ are all independent of each other.

The likelihood of observation i is

$$likelihood_i(\alpha_{\bar{x}}, \beta) \propto e^{-\frac{1}{2\sigma^2}[y_i - (\alpha_{\bar{x}} + \beta(x_i - \bar{x}))]^2}$$

The likelihood of a sample of observations $i = 1, \ldots, n$ is

$$\begin{aligned} likelihood_{sample}(\alpha_{\bar{x}}, \beta) &\propto \prod_{i=1}^{n} e^{-\frac{1}{2\sigma^2}[y_i - (\alpha_{\bar{x}} + \beta(x_i - \bar{x}))]^2} \\ &\propto e^{\sum_{i=1}^{n} -\frac{1}{2\sigma^2}[y_i - (\alpha_{\bar{x}} + \beta(x_i - \bar{x}))]^2} \end{aligned}$$

The term in brackets in the exponent equals

$$\left[\sum_{i=1}^{n} [y_i - \bar{y} + \bar{y} - (\alpha_{\bar{x}} + \beta(x_i - \bar{x}))]^2\right].$$

Breaking this into three sums and multiplying it out gives us

$$\begin{aligned} \sum_{i=1}^{n}(y_i - \bar{y})^2 + 2\sum_{i=1}^{n}(y_i - \bar{y})(\bar{y} - (\alpha_{\bar{x}} + \beta(x_i - \bar{x}))) \\ + \sum_{i=1}^{n}(\bar{y} - (\alpha_{\bar{x}} + \beta(x_i - \bar{x})))^2. \end{aligned}$$

This simplifies into

$$SS_y - 2\beta SS_{xy} + \beta^2 SS_x + n(\alpha_{\bar{x}} - \bar{y})^2,$$

where $SS_y = \sum_{i=1}^{n}(y_i - \bar{y})$, $SS_{xy} = \sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})$, and $SS_x = \sum_{i=1}^{n}(x_i - \bar{x})^2$.

The joint likelihood is

$$likelihood_{sample}(\alpha_{\bar{x}}, \beta) \propto \prod_{i=1}^{n} e^{-\frac{1}{2\sigma^2}\left[SS_y - 2\beta SS_{xy} + \beta^2 SS_x + n(\alpha_{\bar{x}} - \bar{y})^2\right]}$$
$$\propto \prod_{i=1}^{n} e^{-\frac{1}{2\sigma^2}\left[SS_y - 2\beta SS_{xy} + \beta^2 SS_x\right]} \times e^{-\frac{1}{2\sigma^2}\left[n(\alpha_{\bar{x}} - \bar{y})^2\right]}$$

We factor out SS_x in the first exponential, complete the square, and absorb the part that does not depend on any parameter into the proportionality constant. This gives us

$$f(y_1, \ldots, y_n \mid \alpha_{\bar{x}}, \beta) \propto e^{-\frac{SS_{\bar{x}}}{2\sigma^2}\left[\beta - \frac{SS_{xy}}{SS_x}\right]^2} \times e^{-\frac{n}{2\sigma^2}\left[(\alpha_{\bar{x}} - \bar{y})^2\right]}$$
$$\propto f(y_1, \ldots, y_n \mid \alpha_{\bar{x}}) \times e^f(y_1, \ldots, y_n \mid \beta)$$

The joint likelihood has been factored into two independent likelihoods. If we multiply the joint likelihood by the joint prior, then the result will be proportional to the joint posterior. The joint prior of the parameter is proportional to:

$$g(\alpha_{\bar{x}}, \beta) = g(\alpha_{\bar{x}}) \times g(\beta).$$

We can either use normal priors, or flat priors. In this project, we use independent flat priors for β and $\alpha_{\bar{x}}$.

3.3 The Joint Posterior for β and $\alpha_{\bar{x}}$

We are not interested in the posterior for $\alpha_{\bar{x}}$, but we are more interested in the posterior for β. We see that the posterior mean for β is the least squares slope

$$m'_\beta = \beta,$$

and that the posterior variance is

$$(s'_\beta)^2 = \frac{\sigma^2}{SS_x}.$$

3.4 Bayesian Credible Interval for Slope

With simple linear regression we end up with point estimates of parameters, but now we have an entire distribution for each parameter, and can use it to determine confidence levels. A $(1-\alpha)100\%$ Bayesian credible interval for slope β is

$$m'_\beta \pm z_{\frac{\alpha}{2}} \times \sqrt{(s'_\beta)^2}.$$

If we do not know σ^2, then we can estimate it from the sample data:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(y_i - (A_{\bar{x}} + B(x_i - \bar{x})))^2}{n-2},$$

resulting in a confidence interval of

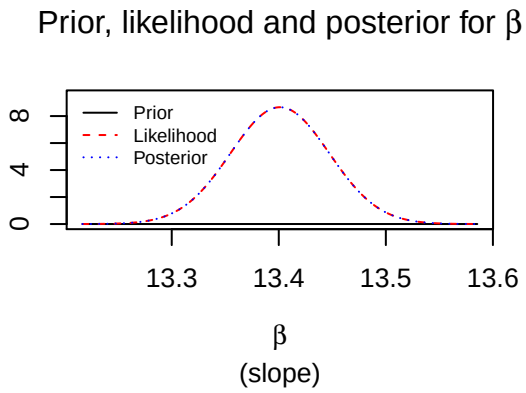
$$m'_\beta \pm t_{\frac{\alpha}{2}} \sqrt{(s'_\beta)^2}.$$

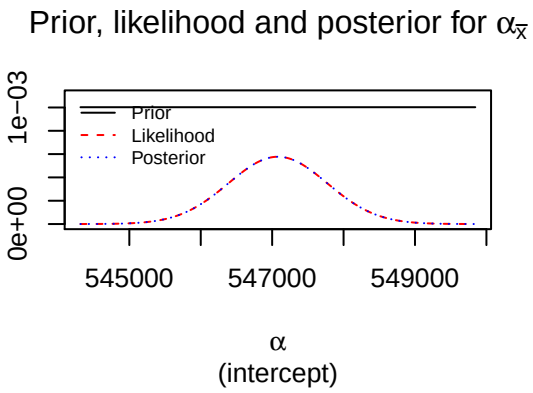
3.5 Testing Two-Sided Hypothesis about Slope

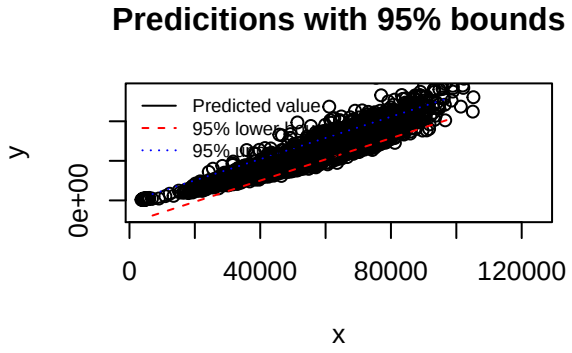
We wish to test whether or not the slope could be zero, i.e. $\beta = 0$. If it could be zero, then we can not be sure that the mean of the response variable depends on the explanatory variable. We would like to test $H_0 : \beta = 0$ versus $H_1 : \beta \neq 0$ at the α level of significance in a Bayesian manner, before we use the regression model to make predictions. To do the test in a Bayesian manner, look where 0 lies in relation to the credible interval. If it lies outside the interval, we reject H_0. Otherwise, we cannot reject the null hypothesis, and we should not use the regression model to help with predictions.

```
## Standard deviation of residuals:  69100

##             Posterior Mean Posterior Std. Deviation
##             -------------- ------------------------
## Intercept:  547100         691.03
## Slope:      13.4           0.046095
```







In this case, the 95% confidence interval for β is $(13.310511, 13.4911997)$; this means that we are 95% confident that β_1 is in this range. Since the confidence interval for β_1 does not contain 0, it can be concluded that there is evidence of a linear relationship between the gold earned and the total damage dealt for the average Gold-ranked player on the blue team, in a Bayesian manner.

4 Summary

In the first part of the project, we found and described 10,000 League of Legends ranked matches from the North American region with 775 variables offered through the Riot Games API. We performed descriptive statistical analyses by considering measures of central tendency and measures of dispersion for numerical

variables. In the second part of the project, we were motivated by the following question: Does the amount of gold earned have an effect on the total damage dealt for the average Gold-ranked player on the blue team? We wanted to describe the relationship between the gold earned and the total damage dealt on the blue team in the `lol10` dataset using a line. We used the gold earned across all summoners on the blue team as the predictor variable, x, to predict the total damage dealt across all summoners on the blue team, y. We observed and handled the nonlinearity of the regression model and the nonconstancy of the error terms. In the third part of the project, we were motivated by the question: Can we better understand the total damage dealt for the average Gold-ranked player on the blue team? A more complex model, containing additional predictor variables `b_gold_earned, b_kills, b_longest_time_spent_living,` and `b_time_c_cing_others`, was employed to provide sufficiently precise predictions of the response variable, $b_total_damage_dealt$. The method is motivated by scenarios where many variables may be simultaneously connected to an output. In the fourth part of the project, we explored applications of ridge regression, LASSO, smoothing splines, kernel smoothers. We also explored Bayesian inference for simple linear regression, which allowed us factor in our prior beliefs and to treat the model parameters as probability distributions.

Bibliography

Bolstad, W.M., and J.M. Curran. 2016. *Introduction to Bayesian Statistics*. Wiley.

"Distribution of League of Legends (Lol) Summoners in North America as of October 2021, by Tier." 2021. *Statista*. https://www.statista.com/statistics/807298/league-of-legends-player-tier/.

Games, Riot. 2021a. "Riot Games Api." *Riot Developer Portal*. https://developer.riotgames.com/apis.

———. 2021b. *Twitter*. Twitter. https://twitter.com/riotgames/status/1455172784938651649?s=20&t=AQmQGrTa1ijf6u3cEDPZcg.

James. 2020. "League of Legends Ranked Match Data from Na." *Kaggle*. https://www.kaggle.com/jamesbting/league-of-legends-ranked-match-data-from-na.

Variable	Description
b_total_damage_dealt	The total damage dealt by the blue side team.
b_gold_earned	The gold obtained by the blue side team.
b_kills	The kills obtained by the blue side team.
b_longest_time_spent_living	The longest time spent living obtained by the blue side team.
b_time_c_cing_others	The time spent crowd controlling others by the blue side team.

Figure 1: Variables and their descriptions