

# Analysis of Graduate School Acceptance Chance

Final Report

Stat 4511

Group A

Group Members: Dagan Larson, Cole Rehbein, Shane Pratt

Due: April 29, 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Data Description . . . . .	2
1.1.1	Variable Overview . . . . .	2
1.2	Exploratory Analysis . . . . .	3
1.3	Figures . . . . .	4
<b>2</b>	<b>Regression Analysis</b>	<b>5</b>
2.1	Model Fitting . . . . .	5
2.2	Cross Validation . . . . .	6
2.3	Assumptions and Remedial Measures . . . . .	6
2.4	Final Model . . . . .	7
2.5	Predictions from Final Model . . . . .	8
2.6	Figures . . . . .	9
<b>3</b>	<b>Discussion</b>	<b>13</b>
3.1	Limitations . . . . .	13
3.2	Appropriateness of Regression & Further Work . . . . .	14
<b>4</b>	<b>Conclusion</b>	<b>14</b>
<b>5</b>	<b>Additional Work</b>	<b>14</b>
5.1	Other Transformed and Polynomial Models . . . . .	14
5.2	Interaction Term Model . . . . .	15
5.3	More Predictions . . . . .	15
<b>6</b>	<b>Appendix</b>	<b>16</b>
6.1	Importing Data & Exploratory Analysis . . . . .	16
6.2	Model Selection . . . . .	17
6.2.1	Cross Validation . . . . .	18
6.3	Model Assumptions . . . . .	18
6.3.1	Polynomial Model Fitting . . . . .	19
6.3.2	Identifying Outliers . . . . .	22
6.4	Weighted Model . . . . .	23
6.5	Bootstrapping . . . . .	23
6.5.1	Confidence Intervals . . . . .	23
6.5.2	Prediction Intervals . . . . .	27

# 1 Introduction

As some of the members of our group are interested in pursuing graduate education, we selected a dataset pertaining to it for our project. Particularly we wanted to examine what factors improve one's chances of acceptance to graduate school. Further, we wanted to see to what extent, if any, that chance decreases as the quality of the school increases. In this paper, we will develop a regression model to predict a person's chances of getting into graduate school based on various academic factors.

## 1.1 Data Description

The dataset used for this project was sourced from Kaggle. It is called Graduate Admission 2 and is sourced at [this link](#). The dataset was created for the sake of helping students shortlist their options for graduate school. The dataset was created from the perspective of Indian graduate schools, hence the emphasis on TOEFL scores and why GPA ranges from 0 to 10. The ultimate goal of this analysis is to make predictions on admission chance, the response variable, based on the various predictors present in the dataset.

For each observation in the dataset, there is a serial number, seven predictor variables associated with that observation, and the response variable. There are 400 rows of the dataset, making its dimensions  $400 \times 9$ , or  $400 \times 8$  excluding serial number.

### 1.1.1 Variable Overview

The variables in the dataset are as follows:

- Response Variable:
  - Admission chance, continuous variable ranging between (0,1). Describes probability of a given student being admitted.
- Predictor Variables:
  1. GRE scores, continuous variable, ranges from 0 to 340. Describes a student's score on the Graduate Records Examination
  2. TOEFL scores, continuous variable, ranges from 0 to 120. Describes a student's score on the Test of English as a Foreign Language exam.
  3. University Rating, categorical variable taking on 1, 2, 3, 4, or 5. Describes the quality of the university being applied to, a higher number representing a more prestigious university.
  4. Statement of Purpose Strength [SOP], categorical variable taking on a value in  $\{1.0, 1.5, 2, \dots, 4.5, 5\}$ . Describes the strength of a student's statement of purpose a higher number representing a stronger statement.
  5. Letter of Recommendation Strength [LOR], categorical variable taking on a value in  $\{1.0, 1.5, 2, \dots, 4.5, 5\}$ . Describes the relative strength of a student's letter of recommendation where a higher number represents a stronger letter.

6. Cumulative GPA [CGPA], continuous variable, ranges from 0 to 10. Gives a student's undergraduate GPA.
7. Research Experience: binary categorical variable, taking on 0 or 1. Whether or not a student did research in their undergraduate, 0 for no, 1 for yes.

## 1.2 Exploratory Analysis

It is intuitive to believe that many of these predictors are correlated with one another, e.g. a student with high GPA probably also does well on the GRE. To check this, a correlation matrix was created with the variables in the dataset. See [Figure 1](#) for this correlation matrix. Interestingly, there are no negative values for correlation between variables, indicating all of the data tends to move in the same direction. Additionally, many of the values for correlation between predictors are fairly close to 1. This suggests that strong multicollinearity is present.

To get a further sense of this multicollinearity, a full model was fitted to the data, that is, a model including all of the predictor variables. This model was used to calculate the variance inflation of each variable. The results are as follows:

1.  $VIF_{GRE} = 4.616$
2.  $VIF_{TOEFL} = 4.289$
3.  $VIF_{LOR} = 2.431$
4.  $VIF_{CGPA} = 5.207$
5.  $VIF_{Research} = 1.543$
6.  $VIF_{University\ Rating} = 2.920$
7.  $VIF_{SOP} = 3.076$

It appears that CGPA is particularly collinear with the rest of the predictors, which is fairly unsurprising, see [Figure 2](#). While we are aware of the multicollinearity present between features, we will not do anything about it, because we are only interested in making predictions in the scope of our data. This does have the unfortunate consequence that any regression coefficients cannot be meaningfully interpreted on their own, however.

### 1.3 Figures

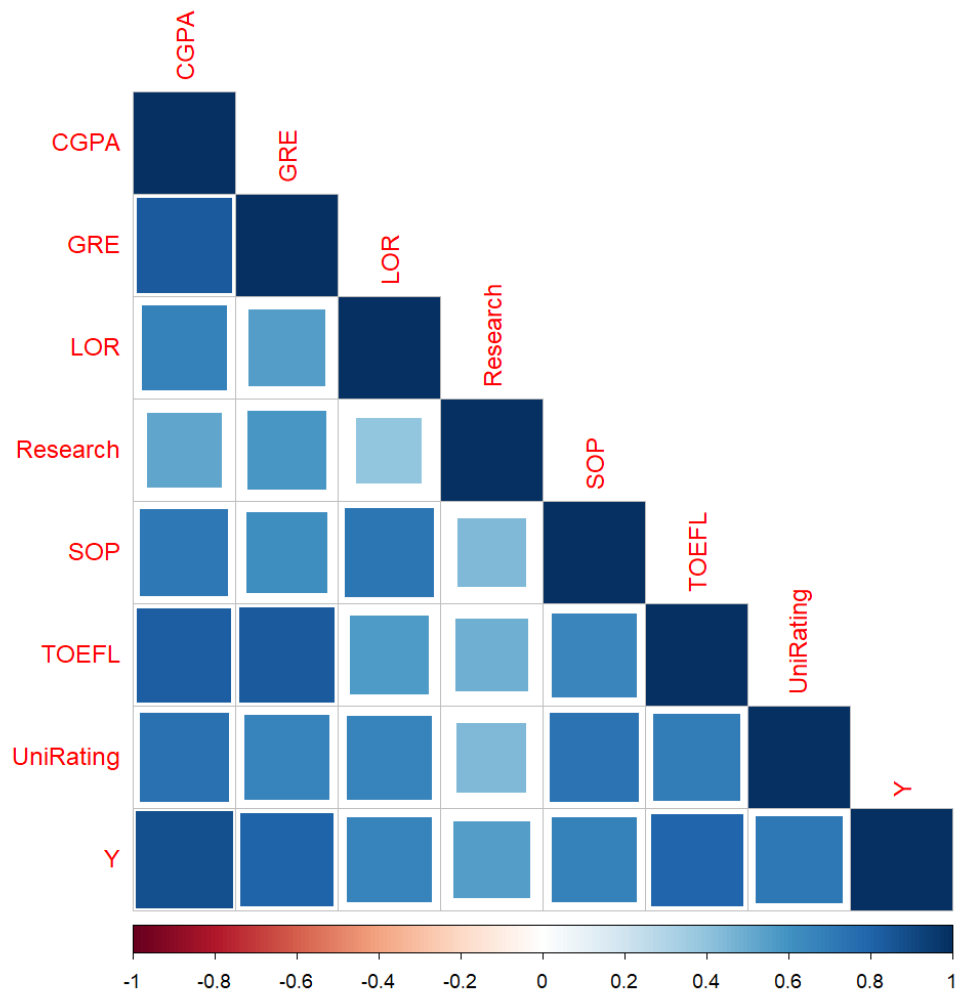


Figure 1: Correlation matrix of dataset. The sizes of the squares and deepness of color indicate magnitude of correlation. All of the correlation values here are positive, indicating that all of the predictors tend to move in the same direction.

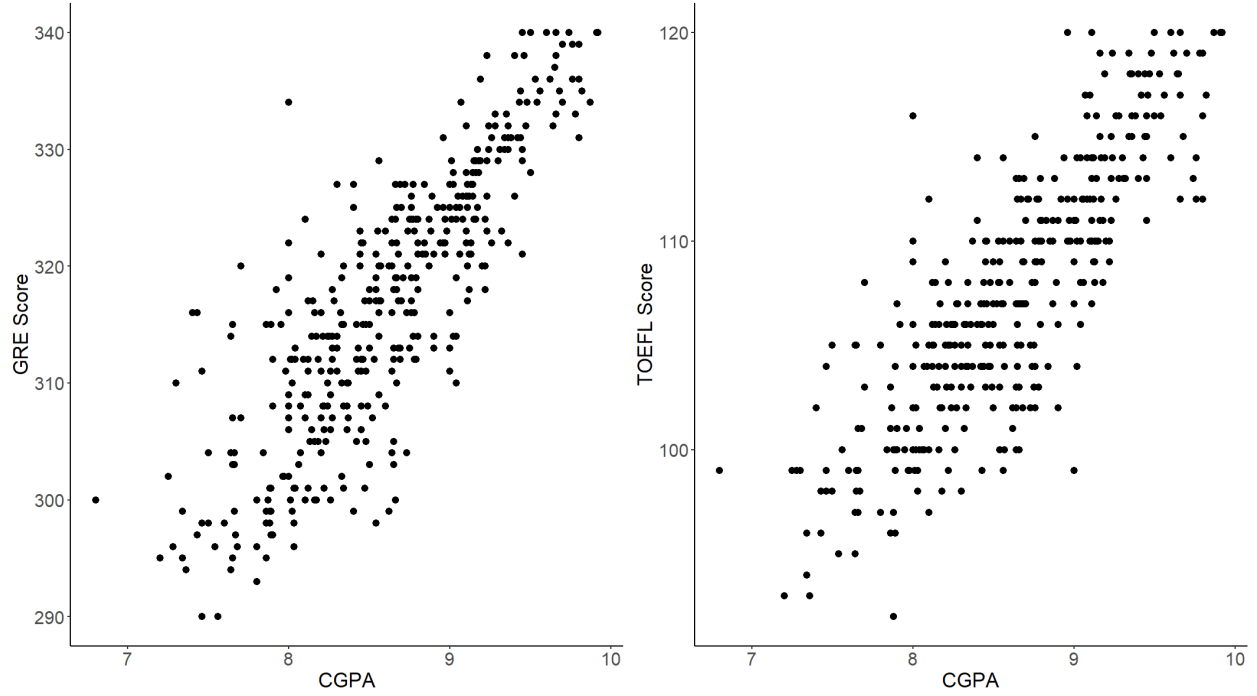


Figure 2: Bivariate plots of CGPA versus GRE or TOEFL score. As shown, CGPA is has a fairly strong positive linear correlation with both of those predictors.

## 2 Regression Analysis

### 2.1 Model Fitting

As there are only 7 variables in the dataset, model fitting was done exhaustively. Comparing these models via *AIC*, the best two models are as follows:

$$\text{Model A: } \hat{Y}_i \sim b_0 + b_1 X_{i1} + b_2 X_{i2} + b_3 X_{i3} + b_4 X_{i4} + b_5 X_{i5}$$

$$\text{Model B: } \hat{Y}_i \sim b_0 + b_1 X_{i1} + b_2 X_{i2} + b_3 X_{i3} + b_4 X_{i4} + b_5 X_{i5} + b_6 X_{i6}$$

Where:

- $\hat{Y}_i$  is estimated probability of acceptance
- $X_{i1}$  is GRE score
- $X_{i2}$  is TOEFL score
- $X_{i3}$  is letter of recommendation strength
- $X_{i4}$  is cumulative GPA
- $X_{i5}$  is whether or not student did undergraduate research

- $X_{i6}$  is university rating

They have  $AIC$ s of -1059.225 and -1058.386, respectively. These two models were also compared with  $BIC$ , adjusted  $R^2$ ,  $PRESS$ , and  $C_p$ . The results are summarized below:

Statistic	Model A	Model B
$AIC$	-1059.225	-1058.386
$BIC$	-1031.284	-1026.454
$R_a^2$	0.8002	0.8003
$PRESS$	0.7966	0.7963
$C_p$	5.494	6.353

As shown, Model A is preferred by  $AIC$  and  $BIC$ , whereas  $R_a^2$ ,  $PRESS$ , and  $C_p$  prefer Model B. Therefore, cross validation was conducted to see which of the two models should be used.

## 2.2 Cross Validation

As there are only 400 observations in the dataset, leave-one-out CV was employed. We find that:

$$MSPE_A = .00413$$

$$MSPE_B = .00413$$

So, the two models have almost identical predictive ability. Additionally, the extra variable in Model B as compared to Model A, namely university rating, was found to not be statistically relevant ( $p = 0.29$ ). Given these facts, we selected Model A for its simplicity. The fact that university rating can be omitted from the model is surprising and different than what we had hypothesized would be the case.

## 2.3 Assumptions and Remedial Measures

To begin, the five predictor variables in Model A were examined for their linear relationship with  $Y$ . As shown in [Figure 3](#), there is linear relationship between  $Y$  and all of the predictors. Also present is significant multicollinearity between the predictors, however, nothing was done to remedy this as this model only seeks to make predictions in the scope of the data.

On the other hand, assumptions of error normalcy and error homoscedasticity are violated in this model (as confirmed by Shapiro-Wilk and Brown-Forsythe tests), see [Figure 4](#) and [Figure 5](#). Transformations on  $Y$  were attempted to fix this, however, this alone was not sufficient. Examination of residual vs. predictor plots showed that polynomial terms for each of the continuous predictors may be necessary.

A hierarchical approach was taken to fitting these polynomial models, wherein a polynomial model featuring quadratic and cubic terms for GRE, TOEFL, and CGPA were introduced. These terms were iteratively deleted by taking the one with the highest p-value  $> 0.05$  and removing it. If a lower-order term was removed this way, the higher power

terms would also be removed. This was repeated until all remaining terms were statistically significant. Note that centered variables were used for these power terms. Applying this approach also did not yield with acceptable error normalcy or distribution, so outliers were investigated for their impact on the model.

It was found that there are five major outliers in the data with respect to  $Y$ . When these five outliers were omitted from the dataset and the procedure outlined above was repeated on this new dataset. When this was done, a model with normal but heteroskedastic errors was constructed. A weighted least squares model was constructed to fix this heteroskedasticity, forming our final model which meets all of the assumptions of regression.

## 2.4 Final Model

The final model is:

$$(\hat{Y}_i)^5 \sim b_0 + b_{w1}X_{i1} + b_{w2}X_{i2} + b_{w3}X_{i3} + b_{w4}X_{i4} + b_{w5}X_{i5} + b_{w22}x_{i2}^2 + b_{w44}x_{i4}^2 + b_{w444}x_{i4}^3$$

Where:

- $\hat{Y}_i$  is estimated probability of acceptance
- $X_{i1}$  is GRE score
- $X_{i2}$  is TOEFL score
- $x_{i2}$  is centered TOEFL score ( $x_{i2} = X_{i2} - \overline{X_2}$ )
- $X_{i3}$  is letter of recommendation strength
- $X_{i4}$  is cumulative GPA
- $x_{i4}$  is centered CGPA
- $X_{i5}$  is whether or not student did undergraduate research
- $b_{wi}$ 's are weighted regression coefficients

This model was constructed from the dataset omitting major  $Y$  outliers, namely points 66, 67, 69, 116, and 360. All of the coefficients are significant to a  $\alpha = 0.05$  level.

The values of these coefficients are listed below alongside bootstrapped confidence intervals with a family wide confidence level of 90% (individual confidence level of  $\alpha = 0.1/9$ ). See [Figure 6](#) for plots showing the estimated distributions of these coefficients.



Coefficient	Estimated Value	90% Family Confidence Limits
$b_{w0}$	-2.5315	(-2.9874 , -2.0727)
$b_{w1}$	0.0013	(-0.0003 , 0.0026)
$b_{w2}$	0.0053	(0.0025 , 0.008)
$b_{w3}$	0.0276	(0.0162 , 0.0401)
$b_{w4}$	0.1928	(0.1516 , 0.243)
$b_{w5}$	0.0337	(0.0102 , 0.0588)
$b_{w22}$	0.0003	(0 , 0.0005)
$b_{w44}$	0.1423	(0.119 , 0.1705)
$b_{w444}$	0.0357	(0.0048 , 0.0591)

## 2.5 Predictions from Final Model

Finally, we used the final model to generate two simultaneous predictions at a family-wide 95% level. We wanted to see how much an average (in the scope of the data) student's acceptance chance changes if they did or did not do research in their undergrad. We let:

- $\text{GRE}_{new} = 320$
- $\text{TOEFL}_{new} = 110$
- $\text{LOR}_{new} = 4.0$
- $\text{CGPA}_{new} = 8.6$
- $\text{Research}_{new} = 0$  for no research and 1 for research

These two new points were confirmed to be well within the range of our data and not extrapolation. We find that:

	Predicted Value ( $\hat{Y}_{new}^5$ )	95% family PIs ( $\hat{Y}_{new}^5$ )
Did Research	0.2651	(0.2471 , 0.2820)
No Research	0.2313	(0.2129 , 0.2487)

	Reverse-Transformed Prediction ( $\hat{Y}_{new}$ )	Reverse-Transformed PIs ( $\hat{Y}_{new}$ )
Did Research	0.7668	(0.7561 , 0.7763)
No Research	0.7462	(0.7339 , 0.7571)

See [Figure 7](#) and [Figure 8](#) for distribution of predicted values. Note that the endpoints for the PIs were calculated before the reverse transformation was applied.

The fact that the endpoints of the PIs are within about  $\pm 1\%$  of the predicted value indicates the high precision of the model's predictions.

As shown in the prediction intervals, whether or not a student with average academics did research seems to only slightly increases their chance of acceptance into graduate school, assuming all else is held constant.

## 2.6 Figures

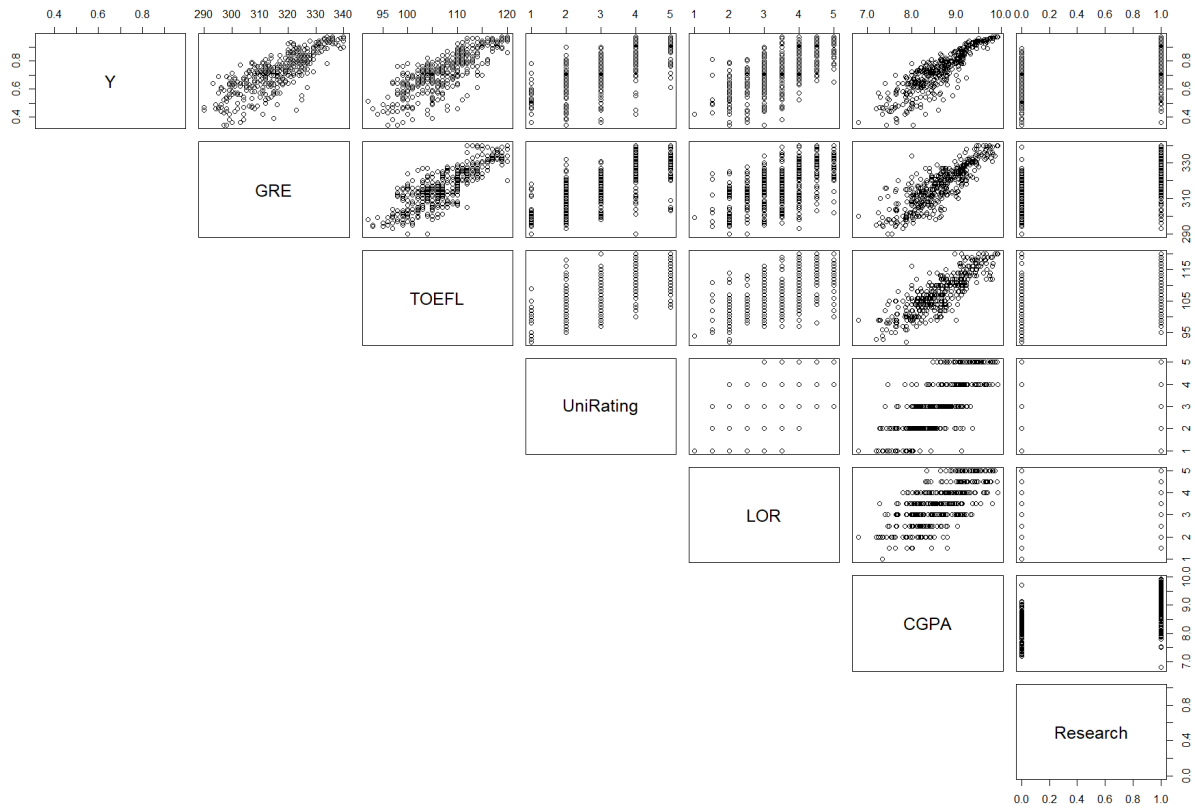


Figure 3: Plots of various predictor variables versus response and against each other. There is linear relationship with  $Y$  and all of the  $X$ 's. There is also significant multicollinearity present.

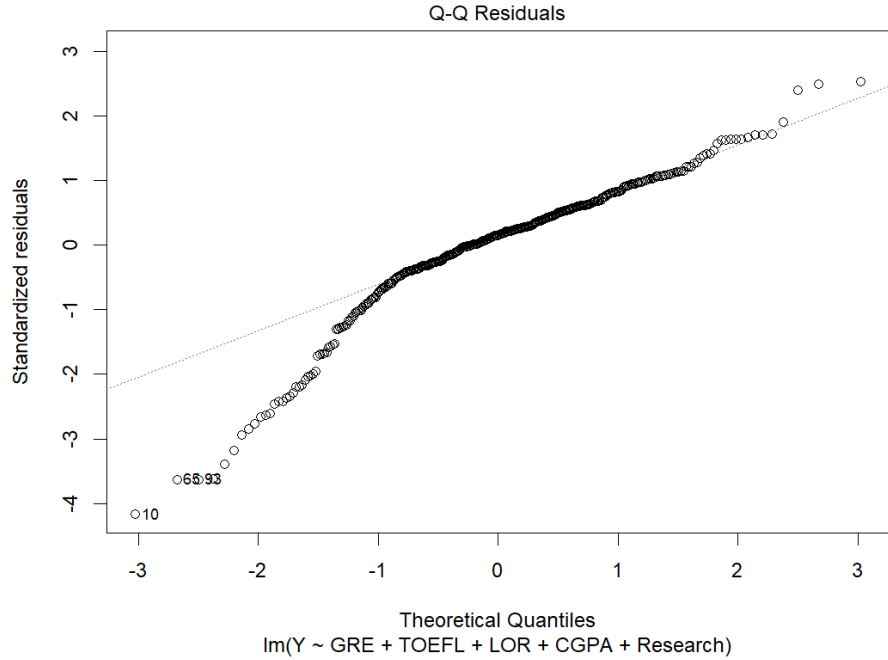


Figure 4: Normal probability plot of basic 5-variable model. Assumption of normality has been violated.

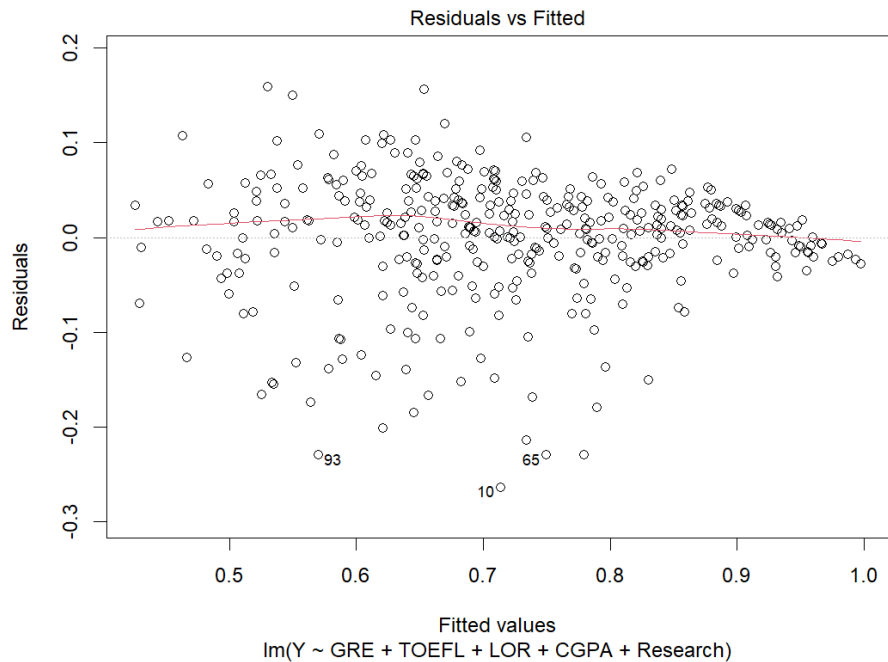


Figure 5: Error variance of basic 5-variable model, heteroskedasticity present.

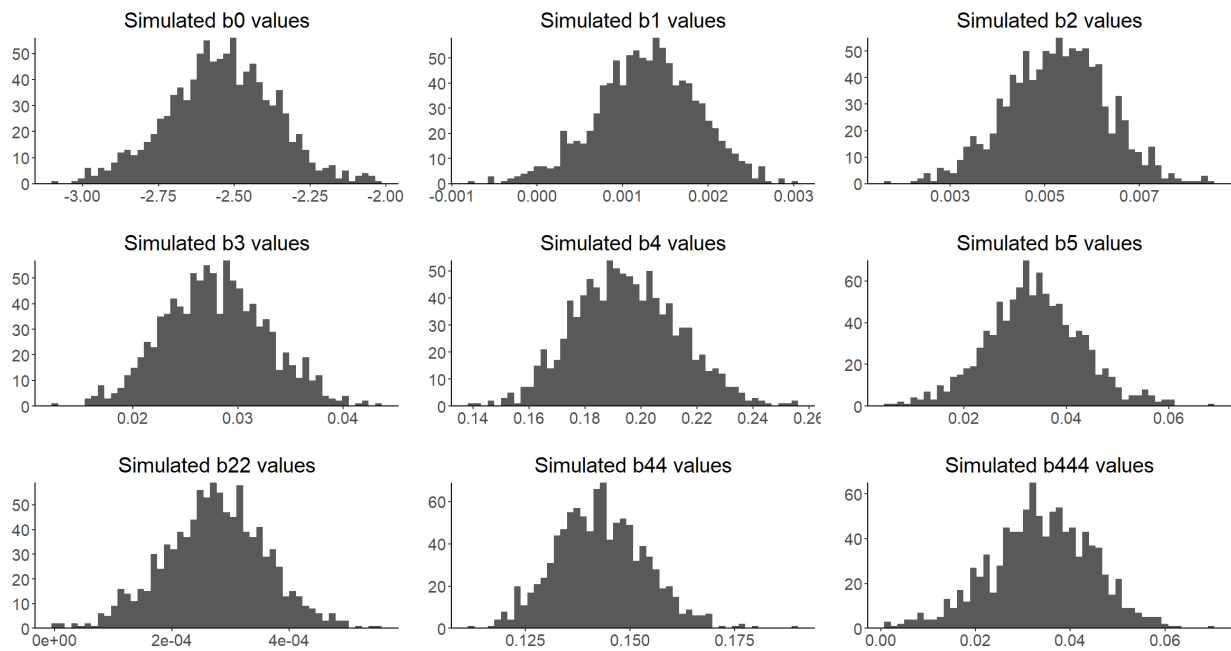


Figure 6: Plots of bootstrapped weighted coefficients over 1000 trials

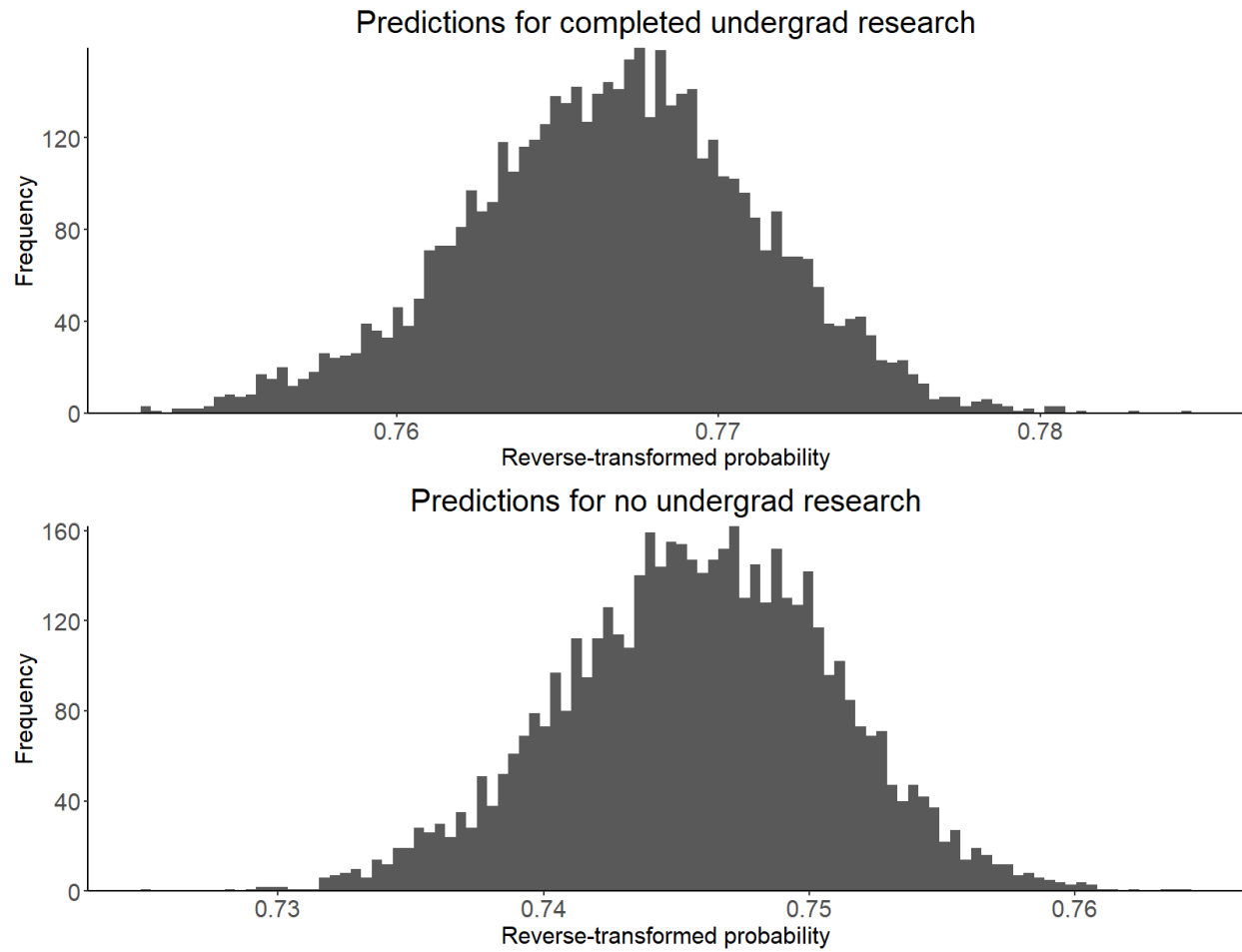


Figure 7: Plots of bootstrapped (5000 iterations) predicted probability of acceptance for an average student with or without having done undergraduate research. Values have already been transformed back into terms of the original data

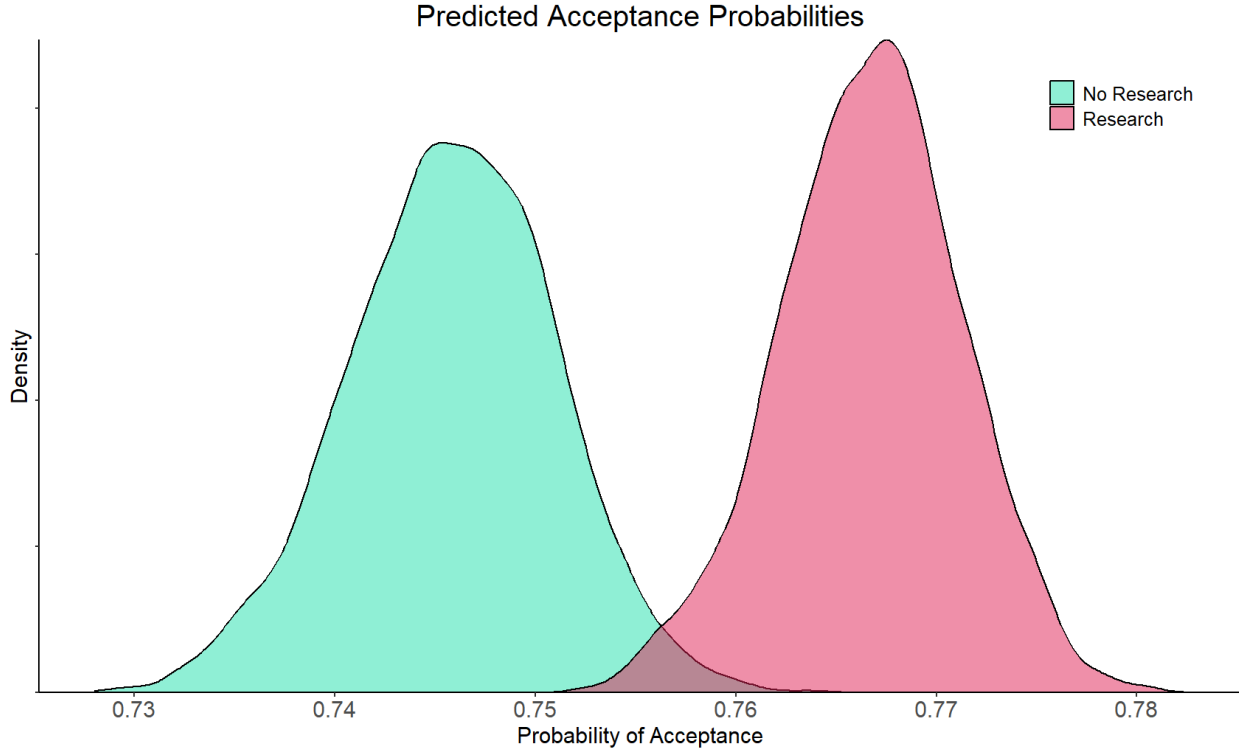


Figure 8: A simultaneous display of the two frequency plots of figure 5, smoothed to show density.

### 3 Discussion

From the predictions we were able to make with this model, we find that completing undergraduate research slightly increases their chance of getting accepted into a graduate school, but only by a few percentage points. We note however, that the variability between predictions is lower in those where research has been completed, compared to those where research wasn't. In any case, though, the prediction intervals are fairly tight, indicating that this model can make precise predictions.

#### 3.1 Limitations

Our method of model selection quickly ruled out university rating and SOP as relevant predictors and proceeded to fit models based upon the five remaining variables. As such, those two variables were never considered in any polynomial or transformed models. It could be the case we missed key insights regarding those two variables by omitting them so quickly from our analysis. We don't believe that this is the case, however, it is still worth mentioning.

Given the multicollinearity present in the data, the transformation of  $Y$ , and the use of weighted regression coefficients, there isn't any meaningful interpretation of the coefficients. This makes it difficult to identify any factor in particular that is important to graduate school acceptance, however, the predictions of the model are still meaningful.

When this model is able to make predictions, it can do so fairly precisely, as stated earlier. However, when we tried to predict acceptance chance for a student in relatively strong or poor academic standing, we could not trust those predictions because they were extrapolatory. Whether or not this a flaw of the model or a consequence of dataset being relatively small is difficult to say without deep analysis. If the latter is not the case, we recognize that another modeling method could improve the range of acceptable predictions.

## 3.2 Appropriateness of Regression & Further Work

While we were ultimately able to create a regression model which meets the assumptions of regression, doing so required significant effort and results in a somewhat exotic model with coefficients with no practical interpretation. Additionally, as WLS regression had to be employed, generating interval estimates or predictions requires the use of bootstrapping. Since the dataset is relatively small, this is not particularly concerning, however, if the dataset were larger this could pose significant computational challenges. Given all of this, we believe that, while regression is a valid tool for the job, there is almost certainly a method of statistical analysis better suited to this dataset.

Were we going to do more work in line with this project, we would likely turn our attention to data from the US regarding graduate admission. Instead of a dataset which describes probability of acceptance, we could look at one which looks at acceptance or rejection as a categorical response variable. Analysis of this data could be done through logistic regression, which may yield better results than the linear regression done here.

## 4 Conclusion

We developed a predictive model for graduate school admission chances using a dataset of academic and application features. Initial model fitting revealed issues with non-normality and heteroskedasticity, which were addressed through transformation, polynomial terms, omission of outliers, and weighted regression. Our results show that factors such as GPA, TOEFL scores, and undergraduate research experience are relevant predictors, while university rating and statement of purpose strength were not found to be statistically significant. Although the transformed response and weighted estimates limit direct interpretation of coefficients, the model generates reliable and precise predictions for cases within the scope of the data.

## 5 Additional Work

### 5.1 Other Transformed and Polynomial Models

Instead of using a fifth-power transform, other powers were tried, however they were outperformed by  $Y^5$ . When the polynomial model was fitted, we tried starting at  $Y$ ,  $Y^2$ ,  $Y^5$ , and  $Y^6$  before removing terms, however, a  $Y^5$  polynomial model was most successful.

## 5.2 Interaction Term Model

We tried to add interaction terms to the model to see if it would fix the error distribution of our model. However, the interaction terms in those models were not statistically significant, did not notably increase model precision, and did not fix the error distribution. Because of this, we abandoned this analysis strategy.

## 5.3 More Predictions

As mentioned earlier, we tried to make further predictions with this model, however, they exceeded the maximum or minimum values of the diagonals of our hat matrix. Perhaps a different modeling approach than the one we took could make reliable predictions over a wider range of values.



## 6 Appendix

### 6.1 Importing Data & Exploratory Analysis

```
1 Admission <- read.table("./Admission_Predict.csv", header = TRUE, sep = ",")
2
3 Admission = Admission[, which(names(Admission) != "Serial.No.")] # remove
  serial.no.
4 library(dplyr) # for rename
5 Admission = rename(Admission,
6                     c(GRE= "GRE.Score", TOEFL = "TOEFL.Score", UniRating = "
  University.Rating", Y = "Chance.of.Admit"))
7 Admission |> head()
8
9 library(corrplot) # for corrplot
10 library(regclass) # for VIF
11
12 M = cor(Admission)
13 corrplot(M, method = 'square', order = 'alphabet', type = 'lower')
14
15 full_model = lm(data = Admission, formula = Y ~ GRE + TOEFL + LOR + CGPA +
  Research + UniRating + SOP)
16 VIF(full_model) |> round(3)
17
18 library(ggplot2)
19 library(ggpubr)
20
21 scatter_gre = ggplot(data = Admission, aes(x = CGPA, y = GRE)) +
22   geom_point(size = 2.5) +
23   theme_classic() +
24   labs(x = "CGPA",
25        y = "GRE Score") +
26   theme(plot.title = element_text(size = 20, hjust = 0.5),
27         axis.text.x = element_text(size = 15),
28         axis.text.y = element_text(size = 15),
29         axis.title.x = element_text(size = 17),
30         axis.title.y = element_text(size = 17))
31 scatter_TOEFL = ggplot(data = Admission, aes(x = CGPA, y = TOEFL)) +
32   geom_point(size = 2.5) +
33   theme_classic() +
34   labs(x = "CGPA",
35        y = "TOEFL Score") +
36   theme(plot.title = element_text(size = 20, hjust = 0.5),
37         axis.text.x = element_text(size = 15),
38         axis.text.y = element_text(size = 15),
39         axis.title.x = element_text(size = 17),
40         axis.title.y = element_text(size = 17))
41 scatter_LOR = ggplot(data = Admission, aes(x = CGPA, y = LOR)) +
42   geom_point(size = 2.5) +
43   theme_classic() +
44   labs(x = "CGPA",
45        y = "Letter of Recommendation Strength") +
46   theme(plot.title = element_text(size = 20, hjust = 0.5),
```

```

47     axis.text.x = element_text(size = 15),
48     axis.text.y = element_text(size = 15),
49     axis.title.x = element_text(size = 17),
50     axis.title.y = element_text(size = 17))
51
52 ggarrange(scatter_gre, scatter_TOEFL, nrow = 1)

```

## 6.2 Model Selection

```

1 library(ExhaustiveSearch)
2
3 # Exhaustive search via AIC
4 es_AIC = ExhaustiveSearch(formula = Y ~ ., data = Admission, family = '
5   gaussian', performanceMeasure = "AIC")
6 print(es_AIC)
7 # Top models are:
8 # GRE + TOEFL + LOR + CGPA + Research
9 # AIC: -1059.225
10 # GRE + TOEFL + UniRating + LOR + CGPA + Research
11 # AIC: -1058.386
12
13 model_5var = lm(data = Admission, formula = Y ~ GRE + TOEFL + LOR + CGPA +
14   Research)
15 model_6var = lm(data = Admission, formula = Y ~ GRE + TOEFL + UniRating + LOR
16   + CGPA + Research)
17 summary(model_5var)
18 summary(model_6var)
19 # Adj R^2 of 5 var : 0.8002
20 # Adj R^2 of 6 var : 0.8003
21
22 library(lme4) # for BIC
23
24 BIC(model_5var) # -1031.284
25 BIC(model_6var) # -1026.454
26
27 library(qpcR) # for PRESS
28
29 PRESS(model_5var, verbose = FALSE) # 0.796567
30 PRESS(model_6var, verbose = FALSE) # 0.7962553
31
32 # PRESS for 6 var model is slightly higher than PRESS of 5 var model
33
34 library(olsrr) # for mallow's CP
35
36 full_model = lm(data = Admission, formula = Y ~ GRE + TOEFL + LOR + CGPA +
37   Research + UniRating + SOP)
38
39 ols_mallows_cp(model_5var, full_model) # 5.494153
40 ols_mallows_cp(model_6var, full_model) # 6.353168

```

```

38 # 6 variable model slightly less biased
39
40 # So:
41 # AIC and BIC favor 5 variable model
42 # Adj R2, PRESS, and CP favor 6 variable model
43 # Cross validation necessary

```

### 6.2.1 Cross Validation

```

1 # CV will be conducted using leave-one-out method
2
3 TotalPE_5 = 0
4 TotalPE_6 = 0
5
6 N = 400
7
8 for(i in 1:N){
9   mod_5 = lm(data = Admission[-c(i),], formula = Y ~ GRE + TOEFL + LOR + CGPA +
   Research)
10  mod_6 = lm(data = Admission[-c(i),], formula = Y ~ GRE + TOEFL + LOR + CGPA +
   Research + UniRating)
11
12  PE_5 = Admission[i,"Y"] - predict(mod_5, Admission[i,])
13  PE_6 = Admission[i,"Y"] - predict(mod_6, Admission[i,])
14
15  TotalPE_5 = TotalPE_5 + PE_5^2
16  TotalPE_6 = TotalPE_6 + PE_6^2
17 }
18
19 MSPE_5 = TotalPE_5/N
20 MSPE_6 = TotalPE_6/N
21
22 round(MSPE_5, 5) # 0.00413
23 round(MSPE_6, 5) # 0.00413
24
25 # So, the 5 and 6 variable models have almost the exact same predictive power
26 # Given this and given that the extra variable (UniRating) in 6 variable model
27 # is not statistically significant (P-value = 0.29), 5 variable model should
   be selected
28 # for simplicity

```

## 6.3 Model Assumptions

```

1 library(car) # for ncvTest (Breusch-pagan)
2
3 base_model = lm(data = Admission, formula = Y ~ GRE + TOEFL + LOR + CGPA +
   Research)

```

```

4 plot(base_model)
5 # residuals look nonnormal and heteroskedastic, no large residuals via Cook's
  Distance however
6
7 shapiro.test(base_model$residuals) # not normal
8 ncvTest(base_model) # heteroskedasticity present
9
10
11 # Plots on predictors vs Y to detect violations of linearity between Y and
    each predictor
12 pairs(Admission[c("Y", "GRE", "TOEFL", "UniRating", "LOR", "CGPA", "Research")],
    lower.panel = NULL)
13 # Y appears to be linear wrt. each predictor, no violation there
14 # multicollinearity present, but the desire is to use model to make
    predictions in scope
15 # of data, nothing will be done about it
16
17 # So, linearity and outlier assumptions seem fine, need to fix error
    distrubtion
18 # Done using Boxcox transformation
19
20 library(EnvStats) # for Boxcox
21
22 bc = boxcox(base_model, lambda = seq(-6,10, 0.1))
23 plot(bc) # optimal value appears to be around 5 or 6
24 bc$lambda[which.max(bc$objective)] # 5.6
25 # Use Y^6 transform model
26
27 model_Y6 = lm(data = Admission, formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA +
    Research)
28 plot(model_Y6)
29 shapiro.test(model_Y6$residuals) # not normal
30 ncvTest(model_Y6) # heteroskedasticity present
31
32 # Normality did get better, but heteroskedasticity is still bad, introduce
    polynomial
33 # terms to model to try to fix this

```

### 6.3.1 Polynomial Model Fitting

```

1 # A hierarchical approach is taken, wherein a polynomial model featuring
    quadratic and
2 # cubic terms for GRE, TOEFL, and CGPA are introduced. These terms are
    iteratively deleted
3 # by taking the one with the highest p-value > 0.05 and removing it. If a
    lower-order term
4 # is removed this way, the higher power terms will also be removed. This will
    be repeated until
5 # all terms are relevant.
6 # Additionally, variables are centered for higher-order terms
7

```

```

8 poly_mod1 = lm(data = Admission ,
9               formula = Y ~ GRE + TOEFL + LOR + CGPA + Research +
10                  I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((CGPA
11                  - mean(CGPA))^2) +
12                  I((GRE - mean(GRE))^3) + I((TOEFL - mean(TOEFL))^3) + I((CGPA
13                  - mean(CGPA))^3))
14 summary(poly_mod1)
15 # Least relevant term is (centered) TOEFL ^ 2, remove that and it's cubic term
16 poly_mod2 = lm(data = Admission ,
17               formula = Y ~ GRE + TOEFL + LOR + CGPA + Research +
18                  I((GRE - mean(GRE))^2) + I((CGPA - mean(CGPA))^2) +
19                  I((GRE - mean(GRE))^3) + I((CGPA - mean(CGPA))^3))
20 summary(poly_mod2)
21 # Least relevant term is (centered) CGPA ^ 2, remove that and it's cubic term
22 poly_mod3 = lm(data = Admission ,
23               formula = Y ~ GRE + TOEFL + LOR + CGPA + Research +
24                  I((GRE - mean(GRE))^2) + I((GRE - mean(GRE))^3))
25 summary(poly_mod3)
26 # Cubic term not relevant , remove that
27 poly_mod4 = lm(data = Admission ,
28               formula = Y ~ GRE + TOEFL + LOR + CGPA + Research +
29                  I((GRE - mean(GRE))^2))
30 summary(poly_mod4)
31 # And last polynomial term is irrelevant
32 # i.e. we've collapsed back into original model
33
34
35
36 # This same approach is taken except starting with Y^6 instead of Y
37 poly_mod2.1 = lm(data = Admission ,
38                formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
39                  I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((CGPA
40                  - mean(CGPA))^2) +
41                  I((GRE - mean(GRE))^3) + I((TOEFL - mean(TOEFL))^3) + I((CGPA
42                  - mean(CGPA))^3))
43 summary(poly_mod2.1)
44 # Least relevant is (centered) TOEFL^3, remove that
45 poly_mod2.2 = lm(data = Admission ,
46                formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
47                  I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
48                  CGPA - mean(CGPA))^2) +
49                  I((GRE - mean(GRE))^3) + I((CGPA - mean(CGPA))^3))
50 summary(poly_mod2.2)
51 # Least relevant is GRE^3, remove that
52 poly_mod2.3 = lm(data = Admission ,
53                formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
54                  I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
55                  CGPA - mean(CGPA))^2) +
56                  I((CGPA - mean(CGPA))^3))
57 summary(poly_mod2.3)

```

```

56 # Least relevant is GRE^2, remove that
57
58 poly_mod2.4 = lm(data = Admission,
59                  formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
60                        I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) +
61                        I((CGPA - mean(CGPA))^3))
62 summary(poly_mod2.4)
63 # All terms relevant; stop here
64
65 cubic_model = lm(data = Admission,
66                  formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
67                        I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((
68                        CGPA - mean(CGPA))^3))
69
70 plot(cubic_model)
71
72 ncvTest(cubic_model)
73 shapiro.test(cubic_model$residuals)
74
75 boxcox(cubic_model, lambda = seq(-6,6,0.1)) |> plot()
76 bc2 = boxcox(cubic_model, lambda = seq(-3,3,0.01))
77 bc2$lambda[which.max(bc2$objective)] # 0.89
78 # .89 * 6 = 5.34
79 # looks like we 'overshot' with Y^6 instead of Y^5
80 # use 5th power transform instead,
81
82 # Redo hierarchical approach, starting at Y^5 instead of Y^6
83
84 poly_mod3.1 = lm(data = Admission,
85                  formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
86                        I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
87                        CGPA - mean(CGPA))^2) +
88                        I((GRE - mean(GRE))^3) + I((TOEFL - mean(TOEFL))^3) + I((
89                        CGPA - mean(CGPA))^3))
90 summary(poly_mod3.1)
91 # Drop TOEFL^3
92 poly_mod3.2 = lm(data = Admission,
93                  formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
94                        I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
95                        CGPA - mean(CGPA))^2) +
96                        I((GRE - mean(GRE))^3) + I((CGPA - mean(CGPA))^3))
97 summary(poly_mod3.2)
98 # drop GRE^3
99 poly_mod3.3 = lm(data = Admission,
100                  formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
101                        I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
102                        CGPA - mean(CGPA))^2) +
103                        I((CGPA - mean(CGPA))^3))
104 summary(poly_mod3.3)
105 # Drop GRE^2
106 poly_mod3.4 = lm(data = Admission,
107                  formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
108                        I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) +
109                        I((CGPA - mean(CGPA))^3))
110 summary(poly_mod3.4)

```

```

105
106 shapiro.test(poly_mod3.4$residuals)
107 ncvTest(poly_mod3.4)
108 # Even still, nonnormal, heteroskedastic errors
109
110 # Further investigation done into outliers to see if omitting strong outliers
111 # can significantly improve model fit
112 # As plot show, points which weren't notable outliers in the original model
113 # may become
114 # strong outliers in new model

```

### 6.3.2 Identifying Outliers

```

1 p = 8
2 n = 400
3
4 # Outlying X obser.
5
6 H = hatvalues(poly_mod3.4)
7 (H > (2 * p / n)) |> which() |> unique() |> sort()
8 H |> sort(decreasing = TRUE) |> head(20)
9 #25 29 30 35 39 48 51 53 57 59 72 79 80 98 118 119 131 144 149
10 #169 177 203 204 214 252 258
11 #27 285 298 345 346 348 349 369 385 386
12 # ^ strongly influential X
13 # 59 is by far strongest point here, 4x bigger h value than next highest
14
15 dff = dffits(poly_mod3.4)
16 (dff > (2*sqrt(p / n))) |> which() |> unique() |> sort()
17 # 83 287 359 360
18
19 dff[c(83, 287, 359, 360)]
20 # 360 is very influential to fitted values
21
22 cd = cooks.distance(poly_mod3.4)
23 (qf(cd, p, n-p) > 0.5) |> which() # none :)
24
25 dfb = dfbeta(poly_mod3.4)
26 (dfb > (2/sqrt(n))) |> which() # none here :)
27
28 # Per the plots, it looks like 66,67,69 are things throwing off residual vs.
29 # fitted and normal
30 # qq plot, look at Y outliers instead
31
32 library(MASS) # for studentized residuals
33
34 sr = studres(poly_mod3.4)
35 rejection = qt(1 - 0.05/(2*n), n - p - 1)
36 (abs(sr) > rejection) |> which() |> unique()
37 # 69 is outlier wrt. y

```

```

37 (abs(sr) > 3) |> which() |> unique() # using slightly lower rejection criteria
38 # 66 67 69 116 360
39 # try omitting these 5 points
40
41 cubic_mod_omitOutliers = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA +
    Research +
42                               I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))
    ^2) + I((CGPA - mean(CGPA))^3),
43                               data = Admission[-c( 66, 67 ,69, 116, 360), ])
44
45 shapiro.test(cubic_mod_omitOutliers$residuals) # 0.09864
46 ncvTest(cubic_mod_omitOutliers) # 0.00068006
47 # Heteroskedasticity still present, but residuals are now reasonably normal
48 # weighting can now be applied to model

```

## 6.4 Weighted Model

```

1 res = cubic_mod_omitOutliers$residuals
2 fitted = cubic_mod_omitOutliers$fitted.values
3 mod = lm(formula = abs(res) ~ fitted)
4
5 var = mod$fitted.values^2
6 w.i = 1/var
7
8 cubic.wls = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
9                I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((
    CGPA - mean(CGPA))^3),
10                data = Admission[-c( 66, 67 ,69, 116, 360), ],
11                weights = w.i)
12 summary(cubic.wls)
13 plot(cubic.wls)
14
15 shapiro.test(cubic.wls$residuals) # 0.2
16 ncvTest(cubic.wls) # 0.41676
17 # So, this weighted cubic model does meet assumptions of regression :)

```

## 6.5 Bootstrapping

### 6.5.1 Confidence Intervals

```

1 # since the family wise confidence level will be 90%, each CI will use
2 # alpha = 0.1/9 = 0.0111... , i.e. each CI is at 98.889% level
3
4 set.seed(123)
5
6 sample_from = setdiff(1:400, c( 66, 67 ,69, 116, 360))
7 # only want to sample from the data used to construct model

```



```

8
9 trials = 1000
10 coeff_values = data.frame(matrix(nrow = 1000, ncol = 9))
11 colnames(coeff_values) = c("b0", "b1", "b2", "b3", "b4", "b5", "b22", "b44", "b444")
12
13 for(i in 1:trials){
14   # Generate sample
15   indices = sample(sample_from, size = 395, replace = T)
16   bt_samp = Admission[indices,]
17
18   # First, fit unweighted model
19   mod1 = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
20             I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((CGPA
21               - mean(CGPA))^3),
22             data = bt_samp)
23
24   # Use that to generate weights
25   res = mod1$residuals
26   fitted = mod1$fitted.values
27   mod = lm(formula = abs(res) ~ fitted)
28
29   var = mod$fitted.values^2
30   sim_weights = 1/var
31
32   # Get coefficient estimates of weighted model
33   sim_mod = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
34               I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((
35                 CGPA - mean(CGPA))^3),
36               data = bt_samp, weights = sim_weights)
37
38   # Record these coefficients
39   coeff_values[i,] = coefficients(sim_mod)
40   # and repeat :)
41 }
42
43 write.csv(coeff_values, "./bootstrap_coefficients", row.names = F)
44
45 library(ggplot2)
46 # Generate histograms for each coefficient
47
48 plot_b0 = ggplot(data = coeff_values, aes(x = b0)) +
49   theme_classic() +
50   geom_histogram(bins = 50) +
51   ylab("") +
52   labs(title = "Simulated b0 values") +
53   xlab("") +
54   scale_y_continuous(expand = expansion(mult = 0)) +
55   theme(plot.title = element_text(size = 20, hjust = 0.5),
56         axis.text.x = element_text(size = 15),
57         axis.text.y = element_text(size = 15),
58         axis.title.x = element_text(size = 15),
59         axis.title.y = element_text(size = 15))
60
61 plot_b1 = ggplot(data = coeff_values, aes(x = b1)) +

```

```

60 theme_classic() +
61 geom_histogram(bins = 50) +
62 ylab("") +
63 labs(title = "Simulated b1 values") +
64 xlab("") +
65 scale_y_continuous(expand = expansion(mult = 0)) +
66 theme(plot.title = element_text(size = 20, hjust = 0.5),
67       axis.text.x = element_text(size = 15),
68       axis.text.y = element_text(size = 15),
69       axis.title.x = element_text(size = 15),
70       axis.title.y = element_text(size = 15))
71
72
73 plot_b2 = ggplot(data = coeff_values, aes(x = b2)) +
74   theme_classic() +
75   geom_histogram(bins = 50) +
76   ylab("") +
77   labs(title = "Simulated b2 values") +
78   xlab("") +
79   scale_y_continuous(expand = expansion(mult = 0)) +
80   theme(plot.title = element_text(size = 20, hjust = 0.5),
81         axis.text.x = element_text(size = 15),
82         axis.text.y = element_text(size = 15),
83         axis.title.x = element_text(size = 15),
84         axis.title.y = element_text(size = 15))
85
86
87 plot_b3 = ggplot(data = coeff_values, aes(x = b3)) +
88   theme_classic() +
89   geom_histogram(bins = 50) +
90   ylab("") +
91   labs(title = "Simulated b3 values") +
92   xlab("") +
93   scale_y_continuous(expand = expansion(mult = 0)) +
94   theme(plot.title = element_text(size = 20, hjust = 0.5),
95         axis.text.x = element_text(size = 15),
96         axis.text.y = element_text(size = 15),
97         axis.title.x = element_text(size = 15),
98         axis.title.y = element_text(size = 15))
99
100
101 plot_b4 = ggplot(data = coeff_values, aes(x = b4)) +
102   theme_classic() +
103   geom_histogram(bins = 50) +
104   ylab("") +
105   labs(title = "Simulated b4 values") +
106   xlab("") +
107   scale_y_continuous(expand = expansion(mult = 0)) +
108   theme(plot.title = element_text(size = 20, hjust = 0.5),
109         axis.text.x = element_text(size = 15),
110         axis.text.y = element_text(size = 15),
111         axis.title.x = element_text(size = 15),
112         axis.title.y = element_text(size = 15))
113

```

```

114
115 plot_b5 = ggplot(data = coeff_values, aes(x = b5)) +
116   theme_classic() +
117   geom_histogram(bins = 50) +
118   ylab("") +
119   labs(title = "Simulated b5 values") +
120   xlab("") +
121   scale_y_continuous(expand = expansion(mult = 0)) +
122   theme(plot.title = element_text(size = 20, hjust = 0.5),
123         axis.text.x = element_text(size = 15),
124         axis.text.y = element_text(size = 15),
125         axis.title.x = element_text(size = 15),
126         axis.title.y = element_text(size = 15))
127
128
129 plot_b22 = ggplot(data = coeff_values, aes(x = b22)) +
130   theme_classic() +
131   geom_histogram(bins = 50) +
132   ylab("") +
133   labs(title = "Simulated b22 values") +
134   xlab("") +
135   scale_y_continuous(expand = expansion(mult = 0)) +
136   theme(plot.title = element_text(size = 20, hjust = 0.5),
137         axis.text.x = element_text(size = 15),
138         axis.text.y = element_text(size = 15),
139         axis.title.x = element_text(size = 15),
140         axis.title.y = element_text(size = 15))
141
142 plot_b44 = ggplot(data = coeff_values, aes(x = b44)) +
143   theme_classic() +
144   geom_histogram(bins = 50) +
145   ylab("") +
146   labs(title = "Simulated b44 values") +
147   xlab("") +
148   scale_y_continuous(expand = expansion(mult = 0)) +
149   theme(plot.title = element_text(size = 20, hjust = 0.5),
150         axis.text.x = element_text(size = 15),
151         axis.text.y = element_text(size = 15),
152         axis.title.x = element_text(size = 15),
153         axis.title.y = element_text(size = 15))
154
155 plot_b444 = ggplot(data = coeff_values, aes(x = b444)) +
156   theme_classic() +
157   geom_histogram(bins = 50) +
158   ylab("") +
159   labs(title = "Simulated b444 values") +
160   xlab("") +
161   scale_y_continuous(expand = expansion(mult = 0)) +
162   theme(plot.title = element_text(size = 20, hjust = 0.5),
163         axis.text.x = element_text(size = 15),
164         axis.text.y = element_text(size = 15),
165         axis.title.x = element_text(size = 15),
166         axis.title.y = element_text(size = 15))
167

```

```

168
169 library(ggpubr) # for ggarrange
170
171 ggarrange(plot_b0, plot_b1, plot_b2, plot_b3, plot_b4, plot_b5, plot_b22, plot_b44,
172           plot_b444)
173 # Get quantile values
174 family_alpha = 0.1
175 alpha = family_alpha/9
176
177 quantile(coeff_values["b0"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
178 quantile(coeff_values["b1"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
179 quantile(coeff_values["b2"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
180 quantile(coeff_values["b3"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
181 quantile(coeff_values["b4"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
182 quantile(coeff_values["b5"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
183 quantile(coeff_values["b22"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
184 quantile(coeff_values["b44"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
185 quantile(coeff_values["b444"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round
  (4)

```

## 6.5.2 Prediction Intervals

```

1      h = hatvalues(cubic.wls)
2 min(h) # 0.006827814
3 max(h) # 0.4296308
4 # ^ range of acceptable values for making predictions and ensure no
   extrapolation
5
6 # See how presence of research affects admission chance at different levels of
   other variables
7 # Average Grades/test scores
8 to_predict1.1 = data.frame("GRE" = 320, "TOEFL" = 110, "LOR" = 4.0, "CGPA" =
   8.6, "Research" = 0)
9 to_predict1.2 = data.frame("GRE" = 320, "TOEFL" = 110, "LOR" = 4.0, "CGPA" =
   8.6, "Research" = 1)
10 # Poor Grades/test scores
11 to_predict2.1 = data.frame("GRE" = 310, "TOEFL" = 105, "LOR" = 3.0, "CGPA" =
   7.5, "Research" = 0)
12 to_predict2.2 = data.frame("GRE" = 310, "TOEFL" = 105, "LOR" = 3.0, "CGPA" =
   7.5, "Research" = 1)
13 # Excellent Grades/test scores
14 to_predict3.1 = data.frame("GRE" = 330, "TOEFL" = 113, "LOR" = 5.0, "CGPA" =
   9.8, "Research" = 0)
15 to_predict3.2 = data.frame("GRE" = 330, "TOEFL" = 113, "LOR" = 5.0, "CGPA" =
   9.8, "Research" = 1)
16
17 # Check to make sure any predictions aren't extrapolation
18
19 X = model.matrix(cubic.wls) |> unname()
20

```

```

21 obs1 = matrix(data = c(1,320, 110, 4.0,8.61, 0, (110 - 107.3823)^2,
22   (8.6-8.594759)^2, (8.6-8.594759)^3),
23   ncol = 1)
24 obs2 = matrix(data = c(1,320, 110, 4.0,8.61, 1, (110 - 107.3823)^2,
25   (8.6-8.594759)^2, (8.6-8.594759)^3),
26   ncol = 1)
27 obs3 = matrix(data = c(1,310, 105, 3.0,7.5, 0, (105 - 107.3823)^2,
28   (7.5-8.594759)^2, (7.5-8.594759)^3),
29   ncol = 1)
30 obs4 = matrix(data = c(1,310, 105, 3.0,7.5, 1, (105 - 107.3823)^2,
31   (7.5-8.594759)^2, (7.5-8.594759)^3),
32   ncol = 1)
33 obs5 = matrix(data = c(1,330, 113, 5.0,9.8, 0, (113 - 107.3823)^2,
34   (9.8-8.594759)^2, (9.8-8.594759)^3),
35   ncol = 1)
36 obs6 = matrix(data = c(1,330, 113, 5.0,9.8, 1, (113 - 107.3823)^2,
37   (9.8-8.594759)^2, (9.8-8.594759)^3),
38   ncol = 1)
39 t(obs1) %*% (t(X) %*% X)^-1 %*% obs1 # 0.07276908
40 t(obs2) %*% (t(X) %*% X)^-1 %*% obs2 # 0.1252422
41 t(obs3) %*% (t(X) %*% X)^-1 %*% obs3 # -1.318934
42 t(obs4) %*% (t(X) %*% X)^-1 %*% obs4 # -1.300921
43 t(obs5) %*% (t(X) %*% X)^-1 %*% obs5 # 3.418855
44 t(obs6) %*% (t(X) %*% X)^-1 %*% obs6 # 3.596004
45
46 # only first 2 points not extrapolation, just look at those
47
48 to_predict1.1 = data.frame("GRE" = 320, "TOEFL" = 110, "LOR" = 4.0, "CGPA" =
49   8.6, "Research" = 0)
50 to_predict1.2 = data.frame("GRE" = 320, "TOEFL" = 110, "LOR" = 4.0, "CGPA" =
51   8.6, "Research" = 1)
52
53 set.seed(777) # for reproducibility
54
55 trials = 5000
56
57 predicted_values = data.frame(matrix(nrow = 5000, ncol = 2))
58 colnames(predicted_values) = c("Research", "No Research")
59
60 for(i in 1:trials){
61   # Generate sample
62   indices = sample(sample_from, size = 395, replace = T)
63   bt_samp = Admission[indices,]
64
65   # First, fit unweighted model
66   mod1 = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
67     I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((CGPA
68     - mean(CGPA))^3),
69     data = bt_samp)
70
71   # Use that to generate weights
72   res = mod1$residuals

```

```

66 fitted = mod1$fitted.values
67 mod = lm(formula = abs(res) ~ fitted)
68
69 var = mod$fitted.values^2
70 sim_weights = 1/var
71
72 sim_mod = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
73               I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((
74                 CGPA - mean(CGPA))^3),
75               data = bt_samp, weights = sim_weights)
76
77 # make predictions with simulated model & record them
78 predicted_values[i,1] = predict(sim_mod, to_predict1.2) # with research
79 predicted_values[i,2] = predict(sim_mod, to_predict1.1) # w/o research
80 # repeat :)
81 }
82 write.csv(predicted_values, "./bt_prediction", row.names = F)
83
84
85 library(ggplot2)
86
87 # Generate histograms for each prediction
88 plot_no_research = ggplot(data = predicted_values, aes(x = 'No Research'^(1/5)
89 )) +
90   theme_classic() +
91   geom_histogram(bins = 100) +
92   ylab("Frequency") +
93   labs(title = "Predictions for no undergrad research") +
94   xlab("Reverse-transformed probability") +
95   scale_y_continuous(expand = expansion(mult = 0)) +
96   theme(plot.title = element_text(size = 20, hjust = 0.5),
97         axis.text.x = element_text(size = 15),
98         axis.text.y = element_text(size = 15),
99         axis.title.x = element_text(size = 15),
100        axis.title.y = element_text(size = 15))
101
102 plot_research = ggplot(data = predicted_values, aes(x = Research^(1/5))) +
103   theme_classic() +
104   geom_histogram(bins = 100) +
105   ylab("Frequency") +
106   labs(title = "Predictions for completed undergrad research") +
107   xlab("Reverse-transformed probability") +
108   scale_y_continuous(expand = expansion(mult = 0)) +
109   theme(plot.title = element_text(size = 20, hjust = 0.5),
110         axis.text.x = element_text(size = 15),
111         axis.text.y = element_text(size = 15),
112         axis.title.x = element_text(size = 15),
113        axis.title.y = element_text(size = 15))
114
115
116 # Want the 2 PI's at 95% level, family wide

```

```

117 # PIs are calculated on transformed data and endpoints are then translated
    back into
118 # original terms
119 family_alpha = 0.05
120 alpha = family_alpha/2 #0.025
121
122 quantile(predicted_values$Research, probs = c(alpha/2, 1 - alpha / 2))
123 # 0.2471215 0.2819776
124 quantile(predicted_values$'No Research', probs = c(alpha/2, 1 - alpha / 2))
125 # 0.2128977 0.2486968
126
127 # in terms of original data
128 (c(0.2471215, 0.2819776 )^(1/5)) |> round(4) # 0.7561 0.7763
129 (c(0.2128977, 0.2486968 )^(1/5)) |> round(4) # 0.7339 0.7571
130
131 ggarrange(plot_research, plot_no_research, nrow = 2)
132
133 library(tidyr) # for gather
134
135 double_plot2 = ggplot(data = gather(predicted_values^(1/5)), aes(x = value,
    fill = key)) +
136   theme_classic() +
137   geom_density(alpha = 0.5) +
138   scale_y_continuous(expand = c(0,0), labels = c()) +
139   scale_x_continuous(limits = c(0.728, 0.7825), breaks = c
    (0.73,0.74,0.75,.76,0.77,.78)) +
140   labs(title = "Predicted Acceptance Probabilities",
141     y = "Density",
142     x = "Probability of Acceptance",
143     fill = "") +
144   scale_fill_manual(values = c("#20DFAB", "#DF2054")) +
145   theme(plot.title = element_text(size = 20, hjust = 0.5),
146     axis.text.x = element_text(size = 15),
147     axis.text.y = element_text(size = 15),
148     axis.title.x = element_text(size = 15),
149     axis.title.y = element_text(size = 15),
150     legend.title = element_text(size=13),
151     legend.text = element_text(size=10),
152     legend.position = c(0.9,0.92))

```