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

1 Introduction

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:

$$\text{Model A: } \hat{Y}_i \sim b_0 + b_1X_{i1} + b_2X_{i2} + b_3X_{i3} + b_4X_{i4} + b_5X_{i5}$$

$$\text{Model B: } \hat{Y}_i \sim b_0 + b_1X_{i1} + b_2X_{i2} + b_3X_{i3} + b_4X_{i4} + b_5X_{i5} + +b_6X_{i6}$$

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 AICs 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. So, 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 it's simplicity.

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 1, 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 statistical tests), see Figures 2 and 3. 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. Centered variables were used for these power terms. This was be repeated until all terms are relevant. 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 data major outliers 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 4 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:

- $\text{GRE}_{new} = 320$
- $\text{TOEFL}_{new} = 110$
- $\text{LOR}_{new} = 4.0$
- $\text{CGPA}_{new} = 8.6$
- Research = 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 ($\hat{Y}^5_{new}$)	95% family PIs ($\hat{Y}^5_{new}$)
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 5 and Figure 6 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 speaks to high precision in this 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

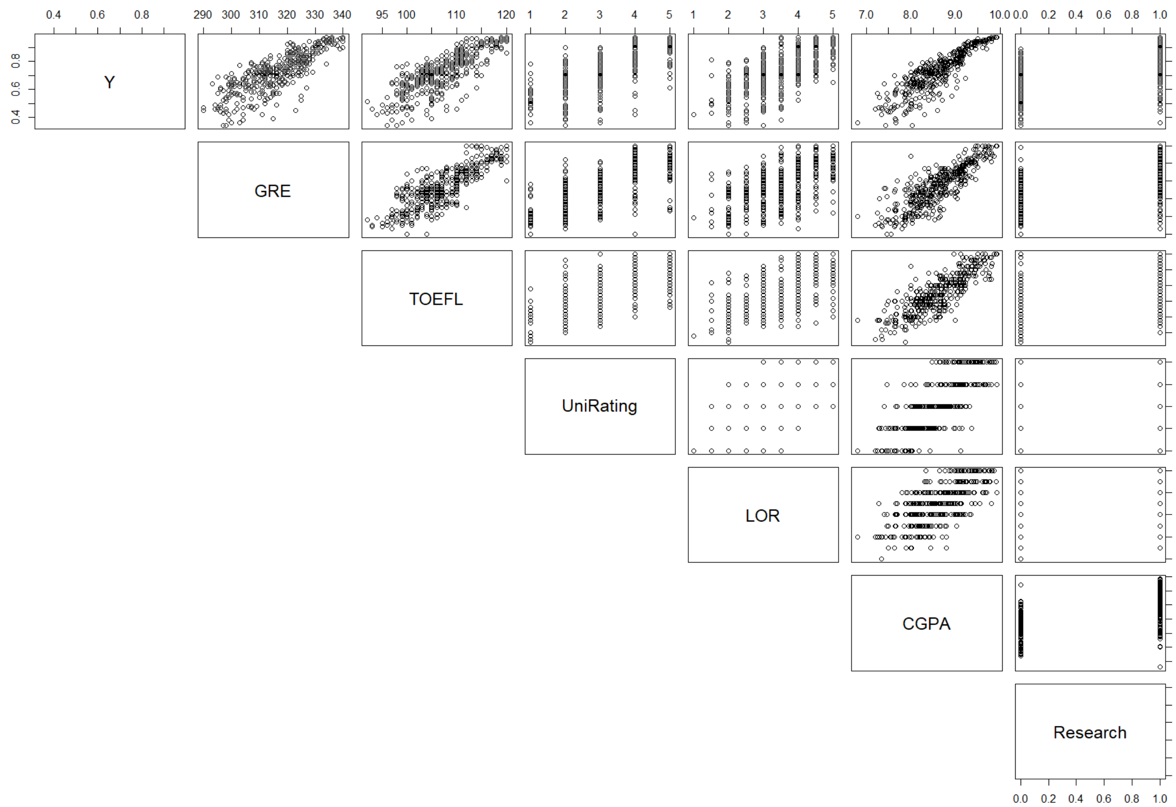

Figure 1: Plots of various predictor variables versus response and against each other. There is linear relationship with Y and all of the X's. There is also significant multicollinearity present.

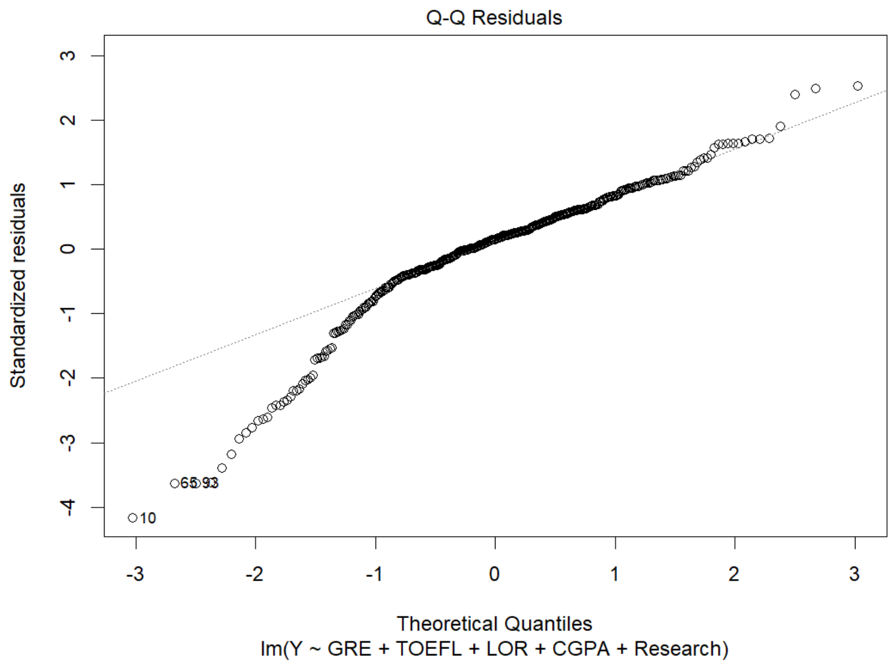

Figure 2: Normal probability plot of basic 5-variable model. Assumption of normality has been violated.

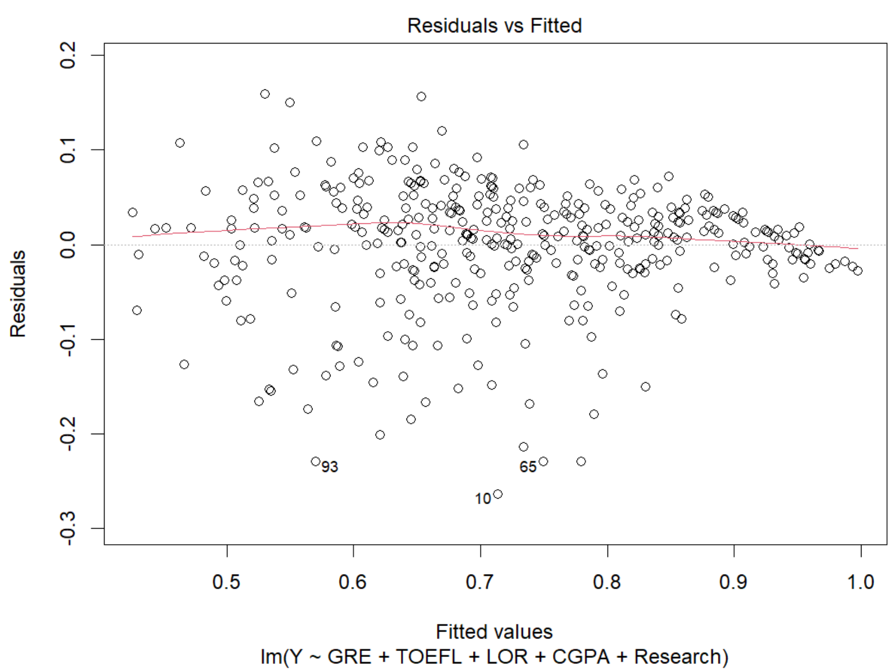

Figure 3: Error variance of basic 5-variable model, heteroskedasticity present.

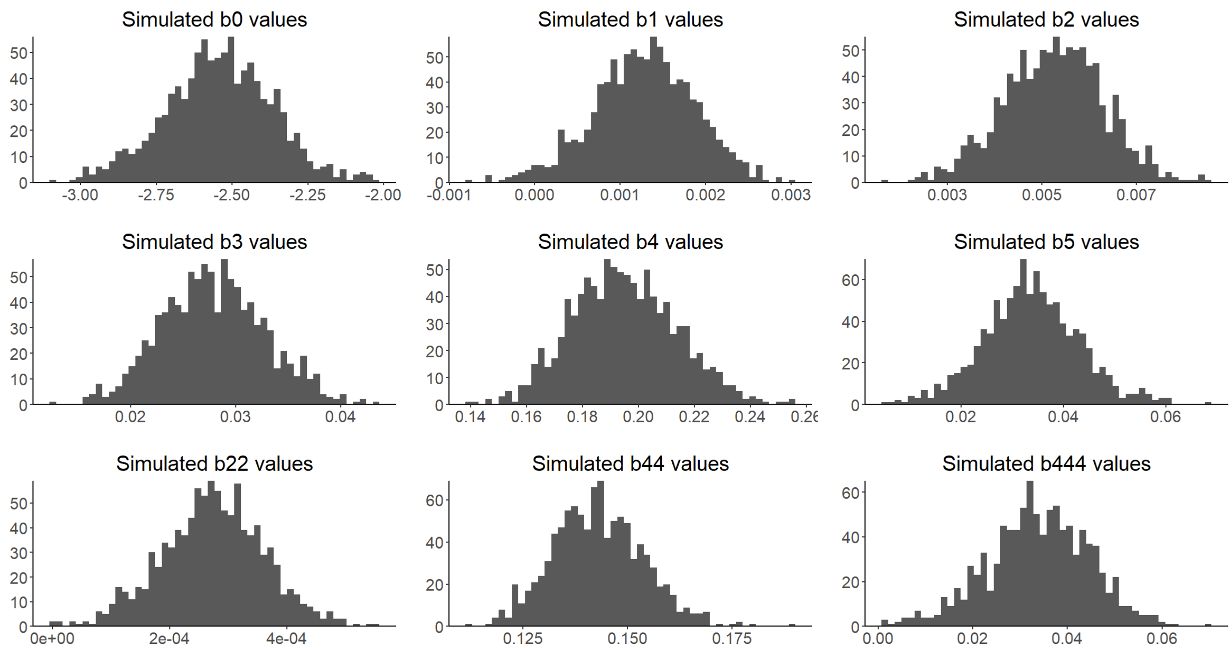

Figure 4: Plots of bootstrapped weighted coefficients over 1000 trials

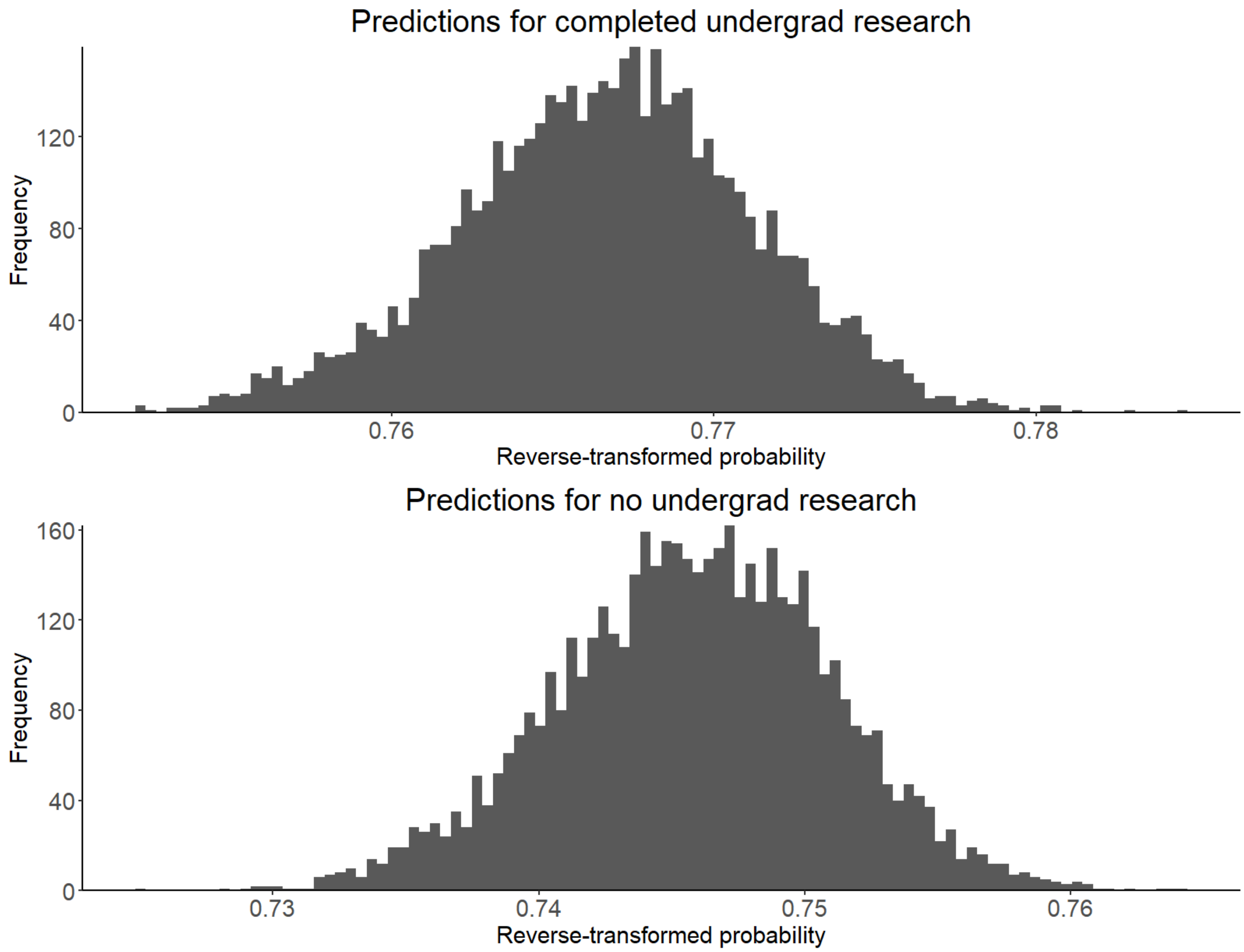

Figure 5: Plots of bootstrapped (5000 iterations) predicted probability of acceptance for an average student with or without having done undergraduate research. Values have already been transformed back into terms of the original data

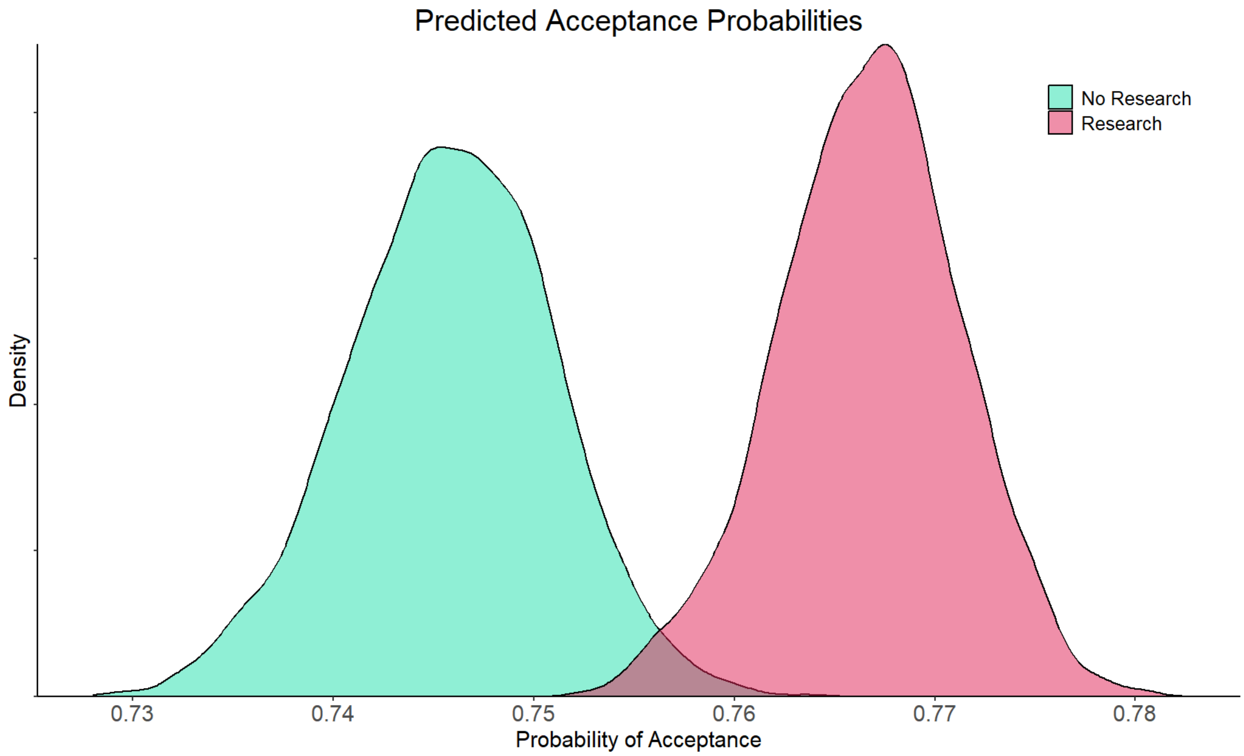

Figure 6: A simultaneous display of the two frequency plots of figure 5, smoothed to show density.

3 Discussion

3.1 Limitations

When this model is able to make predictions, it can do so fairly precisely, as shown 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.

Given the multicollinearity present in the data, the power transform on Y, and the 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.

4 Conclusion

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 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

```
Admission <- read.table("./Admission_Predict.csv", header = TRUE, sep = ",")

Admission = Admission[,which(names(Admission) != "Serial.No.")] # remove
    serial.no.
library(dplyr) # for rename
Admission = rename(Admission,
                   c(GRE= "GRE.Score", TOEFL = "TOEFL.Score", UniRating = "
                      University.Rating", Y = "Chance.of.Admit"))
Admission |> head()
```

6.2 Model Selection

```
library(ExhaustiveSearch)

# Exhaustive search via AIC
es_AIC = ExhaustiveSearch(formula = Y ~ ., data = Admission, family = '
    gaussian', performanceMeasure = "AIC")
print(es_AIC)
# Top models are:
  # GRE + TOEFL + LOR + CGPA + Research
    # AIC: -1059.225
  # GRE + TOEFL + UniRating + LOR + CGPA + Research
    # AIC: -1058.386

model_5var = lm(data = Admission, formula = Y ~ GRE + TOEFL + LOR + CGPA +
    Research)
model_6var = lm(data = Admission, formula = Y ~ GRE + TOEFL + UniRating + LOR
    + CGPA + Research)
summary(model_5var)
summary(model_6var)
# Adj R^2 of 5 var : 0.8002
# Adj R^2 of 6 var : 0.8003

library(lme4) # for BIC

BIC(model_5var) # -1031.284
BIC(model_6var) # -1026.454

library(qpcR) # for PRESS

PRESS(model_5var, verbose = FALSE) # 0.796567
PRESS(model_6var, verbose = FALSE) # 0.7962553

# PRESS for 6 var model is slightly higher than PRESS of 5 var model

library(olsrr) # for mallow's CP
```

```

full_model = lm(data = Admission, formula = Y ~  GRE + TOEFL + LOR + CGPA +
    Research + UniRating + SOP)

ols_mallows_cp(model_5var, full_model) # 5.494153
ols_mallows_cp(model_6var, full_model) # 6.353168

# 6 variable model slightly less biased

# So:
# AIC and BIC favor 5 variable model
# Adj R2, PRESS, and CP favor 6 variable model
# Cross validation necessary
```

6.2.1 Cross Validation

```
# CV will be conducted using leave-one-out method

TotalPE_5 = 0
TotalPE_6 = 0

N = 400

for(i in 1:N){
  mod_5 = lm(data = Admission[-c(i),], formula = Y ~GRE + TOEFL + LOR + CGPA +
       Research)
  mod_6 = lm(data = Admission[-c(i),], formula = Y ~GRE + TOEFL + LOR + CGPA +
       Research + UniRating)

  PE_5 = Admission[i,"Y"] - predict(mod_5,  Admission[i,])
  PE_6 = Admission[i,"Y"] - predict(mod_6,  Admission[i,])

  TotalPE_5 = TotalPE_5 + PE_5^2
  TotalPE_6 = TotalPE_6 + PE_6^2
}

MSPE_5 = TotalPE_5/N
MSPE_6 = TotalPE_6/N

round(MSPE_5, 5) # 0.00413
round(MSPE_6, 5) # 0.00413

# So, the 5 and 6 variable models have almost the exact same predictive power
# Given this and given that the extra variable (UniRating) in 6 variable model
  # is not statistically significant (P-value = 0.29), 5 variable model should
       be selected
  # for simplicity
```

6.3 Model Assumptions

```
library(car) # for ncvTest (Breusch-pagan)

base_model = lm(data = Admission, formula = Y ~ GRE + TOEFL + LOR + CGPA +
    Research)
plot(base_model)
# residuals look nonnormal and heteroskedastic, no large residuals via Cook's
    Distance however

shapiro.test(base_model$residuals) # not normal
ncvTest(base_model) # heteroskedasticity present


# Plots on predictors vs Y to detect violations of linearity between Y and
    each predictor
pairs(Admission[c("Y","GRE", "TOEFL", "UniRating", "LOR", "CGPA","Research")],
     lower.panel = NULL)
# Y appears to be linear wrt. each predictor, no violation there
 # multicolinearity present, but the desire is to use model to make
     predictions in scope
 # of data, nothing will be done about it

# So, linearity and outlier assumptions seem fine, need to fix error
    distrubtion
# Done using Boxcox transformation

library(EnvStats) # for Boxcox

bc = boxcox(base_model, lambda = seq(-6,10, 0.1))
plot(bc) # optimal value appears to be around 5 or 6
bc$lambda[which.max(bc$objective)] # 5.6
# Use Y^6 transform model

model_Y6 = lm(data = Admission, formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA +
    Research)
plot(model_Y6)
shapiro.test(model_Y6$residuals) # not normal
ncvTest(model_Y6) # heteroskedasticity present

# Normality did get better, but heteroskedasticity is still bad, introduce
    polynomial
 # terms to model to try to fix this
```

6.3.1 Polynomial Model Fitting

```
# A hierarchical approach is taken, wherein a polynomial model featuring
    quadratic and
 # cubic terms for GRE, TOEFL, and CGPA are introduced. These terms are
     iteratively deleted
```

```
# by taking the one with the highest p-value > 0.05 and removing it. If a lower-order term
# is removed this way, the higher power terms will also be removed. This will be repeated until
# all terms are relevant.
# Additionally, variables are centered for higher-order terms

poly_mod1 = lm(data = Admission,
               formula = Y ~ GRE + TOEFL + LOR + CGPA + Research +
                 I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) +
                 I((GRE - mean(GRE))^3) + I((TOEFL - mean(TOEFL))^3) + I((CGPA - mean(CGPA))^3))
summary(poly_mod1)
# Least relevant term is (centered) TOEFL ^ 2, remove that and it's cubic term

poly_mod2 = lm(data = Admission,
               formula = Y ~ GRE + TOEFL + LOR + CGPA + Research +
                 I((GRE - mean(GRE))^2)  + I((CGPA - mean(CGPA))^2) +
                 I((GRE - mean(GRE))^3)  + I((CGPA - mean(CGPA))^3))
summary(poly_mod2)
# Least relevant term is (centered) CGPA ^ 2, remove that and it's cubic term

poly_mod3 = lm(data = Admission,
               formula = Y ~ GRE + TOEFL + LOR + CGPA + Research +
                 I((GRE - mean(GRE))^2) + I((GRE - mean(GRE))^3))
summary(poly_mod3)
# Cubic term not relevant, remove that

poly_mod4 = lm(data = Admission,
               formula = Y ~ GRE + TOEFL + LOR + CGPA + Research +
                 I((GRE - mean(GRE))^2))
summary(poly_mod4)
# And last polynomial term is irrelevant
# i.e. we've collapsed back into original model


# This same approach is taken except starting with Y^6 instead of Y
poly_mod2.1 = lm(data = Admission,
               formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
                 I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) +
                 I((GRE - mean(GRE))^3) + I((TOEFL - mean(TOEFL))^3) + I((CGPA - mean(CGPA))^3))
summary(poly_mod2.1)
# Least relevant is (centered) TOEFL^3, remove that

poly_mod2.2 = lm(data = Admission,
                 formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) +
                   I((GRE - mean(GRE))^3)  + I((CGPA - mean(CGPA))^3))
summary(poly_mod2.2)
# Least relevant is GRE^3, remove that
```

```
poly_mod2.3 = lm(data = Admission,
                 formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
                      CGPA - mean(CGPA))^2) +
                   I((CGPA - mean(CGPA))^3))
summary(poly_mod2.3)
# Least relevant is GRE^2, remove that

poly_mod2.4 = lm(data = Admission,
                 formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) +
                   I((CGPA - mean(CGPA))^3))
summary(poly_mod2.4)
# All terms relevant; stop here

cubic_model = lm(data = Admission,
                 formula = I(Y^6) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((
                      CGPA - mean(CGPA))^3))
plot(cubic_model)

ncvTest(cubic_model)
shapiro.test(cubic_model$residuals)

boxcox(cubic_model, lambda = seq(-6,6,0.1)) |> plot()
bc2 = boxcox(cubic_model, lambda = seq(-3,3,0.01))
bc2$lambda[which.max(bc2$objective)] # 0.89
# .89 * 6 = 5.34
# looks like we 'overshot' with Y^6 instead of Y^5
# use 5th power transform instead,

# Redo hierarchical approach, starting at Y^5 instead of Y^6

poly_mod3.1 = lm(data = Admission,
                 formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
                      CGPA - mean(CGPA))^2) +
                   I((GRE - mean(GRE))^3) + I((TOEFL - mean(TOEFL))^3) + I((
                      CGPA - mean(CGPA))^3))
summary(poly_mod3.1)
# Drop TOEFL^3
poly_mod3.2 = lm(data = Admission,
                 formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
                      CGPA - mean(CGPA))^2) +
                   I((GRE - mean(GRE))^3) +  I((CGPA - mean(CGPA))^3))
summary(poly_mod3.2)
# drop GRE^3
poly_mod3.3 = lm(data = Admission,
                 formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((GRE - mean(GRE))^2) + I((TOEFL - mean(TOEFL))^2) + I((
                      CGPA - mean(CGPA))^2) +
                   I((CGPA - mean(CGPA))^3))
```

```
summary(poly_mod3.3)
# Drop GRE^2
poly_mod3.4 = lm(data = Admission,
                 formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) +
                   I((CGPA - mean(CGPA))^3))
summary(poly_mod3.4)

shapiro.test(poly_mod3.4$residuals)
ncvTest(poly_mod3.4)
# Even still, nonnormal, heteroskaditic errors

# Further investiagtion done into outliers to see if omitting strong outliers
  # can significantly improve model fit
# As plot show, points which weren't notable outliers in the original model may become
 # strong outliers in new model
```

6.3.2 Identifying Outliers

```
p = 8
n = 400

# Outlying X obser.

H = hatvalues(poly_mod3.4)
(H > (2 * p / n)) |> which() |> unique() |> sort()
H |> sort(decreasing = TRUE) |> head(20)
#25  29  30  35  39  48  51  53  57  59  72  79  80  98 118 119 131 144 149 169 177 203 204 214 252 258
#27 285 298 345 346 348 349 369 385 386
# ^ strongly influential X
# 59 is by far strongest point here, 4x bigger h value than next highest

dff = dffits(poly_mod3.4)
(dff > (2*sqrt(p / n))) |> which() |> unique() |> sort()
# 83 287 359 360

dff[c(83, 287, 359, 360)]
# 360 is very influential to fitted values

cd = cooks.distance(poly_mod3.4)
(qf(cd, p, n-p) > 0.5) |> which() # none :)

dfb = dfbeta(poly_mod3.4)
(dfb > (2/sqrt(n))) |> which() # none here :)

# Per the plots, it looks like 66,67,69 are things throwing off residual vs. fitted and normal
  # qq plot, look at Y outliers instead

```

```
library(MASS) # for studentized residuals

sr = studres(poly_mod3.4)
rejection = qt(1 - 0.05/(2*n), n - p - 1)
(abs(sr) > rejection) |> which() |> unique()
# 69 is outlier wrt. y

(abs(sr) > 3) |> which() |> unique() # using slightly lower rejection criteria
# 66  67  69 116 360
# try omitting these 5 points

cubic_mod_omitOutliers = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                              I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((CGPA - mean(CGPA))^3),
                              data = Admission[-c( 66,  67  ,69, 116, 360), ])

shapiro.test(cubic_mod_omitOutliers$residuals) # 0.09864
ncvTest(cubic_mod_omitOutliers) # 0.00068006
# Heteroskedasticity still present, but residuals are now reasonably normal
 # weighting can now be applied to model
```

6.4 Weighted Model

```
res = cubic_mod_omitOutliers$residuals
fitted = cubic_mod_omitOutliers$fitted.values
mod = lm(formula = abs(res) ~ fitted)

var = mod$fitted.values^2
w.i = 1/var

cubic.wls = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                 I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((CGPA - mean(CGPA))^3),
               data = Admission[-c( 66,  67  ,69, 116, 360), ],
               weights = w.i)
summary(cubic.wls)
plot(cubic.wls)

shapiro.test(cubic.wls$residuals) # 0.2
ncvTest(cubic.wls) # 0.41676
# So, this weighted cubic model does meet assumptions of regression :)
```

6.5 Bootstrapping

6.5.1 Confidence Intervals

```
    # since the family wise confidence level will be 90%, each CI will use
# alpha = 0.1/9 = 0.0111... , i.e. each CI is at 98.889% level

set.seed(123)

sample_from = setdiff(1:400, c( 66,  67  ,69, 116, 360))
# only want to sample from the data used to construct model

trials = 1000
coeff_values = data.frame(matrix(nrow = 1000, ncol = 9))
colnames(coeff_values) = c("b0","b1","b2","b3","b4","b5","b22","b44","b444")

for(i in 1:trials){
  # Generate sample
  indices = sample(sample_from, size = 395,replace = T)
  bt_samp = Admission[indices,]

  # First, fit unweighted model
  mod1 = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((CGPA
                  - mean(CGPA))^3),
             data = bt_samp)

  # Use that to generate weights
  res = mod1$residuals
  fitted = mod1$fitted.values
  mod = lm(formula = abs(res) ~ fitted)

  var = mod$fitted.values^2
  sim_weights = 1/var

  # Get coefficient estimates of weighted model
  sim_mod = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                   I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((
                     CGPA - mean(CGPA))^3),
               data = bt_samp, weights = sim_weights)

  # Record these coefficients
  coeff_values[i,] = coefficients(sim_mod)
  # and repeat :)
}

write.csv(coeff_values,"./bootstrap_coefficients",row.names = F)

library(ggplot2)
# Generate histograms for each coefficient

plot_b0 = ggplot(data = coeff_values, aes(x = b0)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b0 values") +
  xlab("") +
```

```
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))

plot_b1 = ggplot(data = coeff_values, aes(x = b1)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b1 values") +
  xlab("") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))


plot_b2 = ggplot(data = coeff_values, aes(x = b2)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b2 values") +
  xlab("") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))


plot_b3 = ggplot(data = coeff_values, aes(x = b3)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b3 values") +
  xlab("") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))


plot_b4 = ggplot(data = coeff_values, aes(x = b4)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b4 values") +
```

```
  xlab("") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))


plot_b5 = ggplot(data = coeff_values, aes(x = b5)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b5 values") +
  xlab("") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))


plot_b22 = ggplot(data = coeff_values, aes(x = b22)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b22 values") +
  xlab("") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))

plot_b44 = ggplot(data = coeff_values, aes(x = b44)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b44 values") +
  xlab("") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))

plot_b444 = ggplot(data = coeff_values, aes(x = b444)) +
  theme_classic() +
  geom_histogram(bins = 50) +
  ylab("") +
  labs(title = "Simulated b444 values") +
```

```r
  xlab("") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))


library(ggpubr) # for ggarrange

ggarrange(plot_b0, plot_b1, plot_b2, plot_b3, plot_b4, plot_b5, plot_b22, plot_b44, plot_b444)

# Get quanitle values
family_alpha = 0.1
alpha = family_alpha/9

quantile(coeff_values["b0"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
quantile(coeff_values["b1"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
quantile(coeff_values["b2"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
quantile(coeff_values["b3"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
quantile(coeff_values["b4"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
quantile(coeff_values["b5"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
quantile(coeff_values["b22"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
quantile(coeff_values["b44"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
quantile(coeff_values["b444"][,1], probs = c(alpha/2, 1 - alpha/2)) |> round(4)
```

6.5.2 Prediction Intervals

```r
    h = hatvalues(cubic.wls)
min(h) # 0.006827814
max(h) # 0.4296308
# ^ range of acceptable values for making predictions and ensure no extrapolation

# See how presence of research affects admission chance at different levels of other variables
# Average Grades/test scores
to_predict1.1 = data.frame("GRE" = 320, "TOEFL" = 110, "LOR" = 4.0, "CGPA" = 8.6, "Research" = 0)
to_predict1.2 = data.frame("GRE" = 320, "TOEFL" = 110, "LOR" = 4.0, "CGPA" = 8.6, "Research" = 1)
# Poor Grades/test scores
to_predict2.1 = data.frame("GRE" = 310, "TOEFL" = 105, "LOR" = 3.0, "CGPA" = 7.5, "Research" = 0)
to_predict2.2 = data.frame("GRE" = 310, "TOEFL" = 105, "LOR" = 3.0, "CGPA" = 7.5, "Research" = 1)
# Excellent Grades/test scores
```

```
to_predict3.1 = data.frame("GRE" = 330, "TOEFL" = 113, "LOR" = 5.0, "CGPA" =
    9.8, "Research" = 0)
to_predict3.2 = data.frame("GRE" = 330, "TOEFL" = 113, "LOR" = 5.0, "CGPA" =
    9.8, "Research" = 1)

# Check to make sure any predictions aren't extrapolation

X = model.matrix(cubic.wls) |> unname()

obs1 = matrix(data = c(1,320, 110, 4.0,8.61, 0, (110 - 107.3823)^2,
    (8.6-8.594759)^2, (8.6-8.594759)^3),
              ncol = 1)
obs2 = matrix(data = c(1,320, 110, 4.0,8.61, 1, (110 - 107.3823)^2,
    (8.6-8.594759)^2, (8.6-8.594759)^3),
              ncol = 1)
obs3 = matrix(data = c(1,310, 105, 3.0,7.5, 0, (105 - 107.3823)^2,
    (7.5-8.594759)^2, (7.5-8.594759)^3),
              ncol = 1)
obs4 = matrix(data = c(1,310, 105, 3.0,7.5, 1, (105 - 107.3823)^2,
    (7.5-8.594759)^2, (7.5-8.594759)^3),
              ncol = 1)
obs5 = matrix(data = c(1,330, 113, 5.0,9.8, 0, (113 - 107.3823)^2,
    (9.8-8.594759)^2, (9.8-8.594759)^3),
              ncol = 1)
obs6 = matrix(data = c(1,330, 113, 5.0,9.8, 1, (113 - 107.3823)^2,
    (9.8-8.594759)^2, (9.8-8.594759)^3),
              ncol = 1)

t(obs1) %*% (t(X) %*% X)^-1 %*% obs1  # 0.07276908
t(obs2) %*% (t(X) %*% X)^-1 %*% obs2  # 0.1252422
t(obs3) %*% (t(X) %*% X)^-1 %*% obs3  # -1.318934
t(obs4) %*% (t(X) %*% X)^-1 %*% obs4  # -1.300921
t(obs5) %*% (t(X) %*% X)^-1 %*% obs5  # 3.418855
t(obs6) %*% (t(X) %*% X)^-1 %*% obs6  # 3.596004

# only first 2 points not extrapolation, just look at those

to_predict1.1 = data.frame("GRE" = 320, "TOEFL" = 110, "LOR" = 4.0, "CGPA" =
    8.6, "Research" = 0)
to_predict1.2 = data.frame("GRE" = 320, "TOEFL" = 110, "LOR" = 4.0, "CGPA" =
    8.6, "Research" = 1)


set.seed(777) # for reproducibility

trials = 5000

predicted_values = data.frame(matrix(nrow = 5000, ncol = 2))
colnames(predicted_values) = c("Research", "No Research")

for(i in 1:trials){
  # Generate sample
  indices = sample(sample_from, size = 395,replace = T)
  bt_samp = Admission[indices,]
```

```r

  # First, fit unweighted model
  mod1 = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
              I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((CGPA
                 - mean(CGPA))^3),
            data = bt_samp)

  # Use that to generate weights
  res = mod1$residuals
  fitted = mod1$fitted.values
  mod = lm(formula = abs(res) ~ fitted)

  var = mod$fitted.values^2
  sim_weights = 1/var

  sim_mod = lm(formula = I(Y^5) ~ GRE + TOEFL + LOR + CGPA + Research +
                 I((TOEFL - mean(TOEFL))^2) + I((CGPA - mean(CGPA))^2) + I((
                    CGPA - mean(CGPA))^3),
               data = bt_samp, weights = sim_weights)

  # make predictions with simulated model & record them
  predicted_values[i,1] = predict(sim_mod, to_predict1.2) # with research
  predicted_values[i,2] = predict(sim_mod, to_predict1.1) # w/o research
  # repeat :)
}

write.csv(predicted_values, "./bt_prediction", row.names = F)


library(ggplot2)

# Generate histograms for each prediction
plot_no_research = ggplot(data = predicted_values, aes(x = `No Research`^(1/5)
    )) +
  theme_classic() +
  geom_histogram(bins = 100) +
  ylab("Frequency") +
  labs(title = "Predictions for no undergrad research") +
  xlab("Reverse-transformed probability") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))

plot_research = ggplot(data = predicted_values, aes(x =  Research^(1/5))) +
  theme_classic() +
  geom_histogram(bins = 100) +
  ylab("Frequency") +
  labs(title = "Predictions for completed undergrad research") +
  xlab("Reverse-transformed probability") +
  scale_y_continuous(expand = expansion(mult = 0)) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
```

```r
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))



# Want the 2 PI's at 95% level, family wide
# PIs are calculated on transformed data and endpoints are then translated back into
# original terms
family_alpha = 0.05
alpha = family_alpha/2 #0.025

quantile(predicted_values$Research, probs = c(alpha/2, 1 - alpha / 2))
# 0.2471215 0.2819776
quantile(predicted_values$`No Research`, probs = c(alpha/2, 1 - alpha / 2))
# 0.2128977 0.2486968

# in terms of original data
(c(0.2471215, 0.2819776 )^(1/5)) |> round(4) #  0.7561 0.7763
(c(0.2128977, 0.2486968  )^(1/5)) |> round(4) #  0.7339 0.7571

ggarrange(plot_research, plot_no_research, nrow = 2)

library(tidyr) # for gather

double_plot2 = ggplot(data = gather(predicted_values^(1/5)), aes(x = value, fill = key)) +
  theme_classic() +
  geom_density(alpha = 0.5) +
  scale_y_continuous(expand = c(0,0), labels = c()) +
  scale_x_continuous(limits = c(0.728, 0.7825),breaks = c(0.73,0.74,0.75,.76,0.77,.78)) +
  labs(title = "Predicted Acceptance Probabilities",
       y = "Density",
       x = "Probability of Acceptance",
       fill = "") +
  scale_fill_manual(values = c("#20DFAB","#DF2054")) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15),
        legend.title = element_text(size=13),
        legend.text = element_text(size=10),
        legend.position = c(0.9,0.92))
```