Evaluation of Systolic Blood Pressure for Pregnant Women

Victoria Fenwick, Helen Jacobson, and Jack Roberts

MTH 245 Statistical Methods I with R

December 12th, 2023

Contents

1 Abstract

According to the American Heart Association, hypertension during pregnancy is the second-leading cause of maternal death worldwide (News (2021)). In the United States alone, almost 15% of maternal deaths are related to hypertension (Abera and Mekonnen (2019)). High systolic blood pressure is a form of hypertension, which is the focus of our project. The purpose of this study is to determine which variables are most predictive of systolic blood pressure, which predictive variable interaction has the most important influence on systolic blood pressure, and what variables are least indicative of systolic blood pressure. We used linear regression as the key strategy to identify the best fit model that highlights the relationship between variables such as age, blood sugar level, risk level, heart rate, and diastolic blood pressure to systolic blood pressure. We created multiple models including a first-order, two-way interaction, and two-way interaction with backwards selection using Team (2023).

To determine our best model, we ranked each of them by R^2, R^2_{adj}, LL, AIC, BIC, RMSE, MAE, CV R^2, CV RMSE, and CV MAE (CV = Leave-One-Out Cross-Validation). The two-way backward model was the best fit model according to our ranking system. Overall, our research concluded that the best fit model was the two-way backwards model. We found that the most predictive variables of systolic blood pressure were diastolic blood pressure (t-value = 13.2154, p-value < 0.0001) and low risk level(t-value = -5.4471, p-value < 0.0001). The most significant interaction was between diastolic blood pressure and body temperature (t-value = 5.6983, p-value < 0.0001). The least indicative variables were interaction between age and risk level low (t-value = -0.4559, $p - value = 0.6486$), blood sugar (t-value = -0.7550, p-value = 0.4504), and age (t-value = 0.9706, p-value = 0.3320).

2 Introduction

The Maternal Health Risk data set was sourced from Kaggle (Ahmed (2023)). The data set includes data that was collected from different hospitals, community clinics, and maternal health cares from the rural Bangladesh. The data was collected through an IoT based risk monitoring system. We were interested in how various variables such as a pregnant woman's age, heart rate, blood sugar, and how her project risk during pregnancy impacts their systolic blood pressure. Additionally, there are seven variables that we could work with and 1,014 observations which made the data set robust for our intended research. The variables include age (Age), systolic blood pressure (SystolicBP), diastolic blood pressure (DiastolicBP), blood sugar levels (BS), body temperatures (BodyTemp), heart rate (HeartRate), and risk level (RiskLevel). All of the variables are continuous quantitative, with the exception of risk level which is categorical. The measurements for each variable are as follows: age in years, systolic and diastolic blood pressure in mmHg, blood sugar level in mmol/L, body temperature in Fahrenheit, heart rate in beats per minute, and risk level as low, medium, and high.

This analysis has potential to be extremely beneficial to pregnant women in the "real" world because each of these variables are responsible and significant risk factors for maternal mortality which is a major concern. High systolic blood pressure is a form of hypertension which is the second-leading cause of maternal death worldwide (Abera and Mekonnen (2019)). Severe hypertension increases a pregnant woman's risk of cardiac failure, heart attack, renal failure, and cerebral vascular accidents (News (2021)). Concurrently, there is increased risk for the fetus as well, such as poor placental transfer of oxygen and neonatal death (News (2021)). By analyzing this data from pregnant

women, we could find potential correlations between certain variables leading to higher systolic blood pressures, and with that knowledge, take further steps to evaluate how to decrease pregnant women's risk of mortality. We utilized the following packages for our research: GGally (Schloerke et al. (2021)), car (Fox and Weisberg (2019)), ggplot2 (Wickham (2016)), MASS (Venables and Ripley (2002)), Stat2Data (Cannon et al. (2019)), rms (E and Jr (2023)), FSA (Ogle et al. (2023)), RVAideMemoire (HERVE (2023)), xtable (Dahl et al. (2019)), tidyverse (Wickham et al. (2019)), caret (Kuhn and Max (2008)), RColorBrewer (Neuwirth (2022)), bestglm (McLeod et al. (2020)), and patchwork (Pedersen (2023)).

3 Exploratory Data Analysis

Now we will be transitioning into the exploratory data analysis portion of our research project. Here we wanted to evaluate our data numerically by glimpsing our data and evaluating the summary statistics. Additionally, we evaluated the correlation coefficients between each predictor variable and systolic blood pressure. Lastly, we inspected the risk level, our only categorical variable, to ensure our data had significant proportions of low, medium, and high risk patient cases.

```
library(tidyverse)
library(xtable)
library(patchwork)
library(RVAideMemoire)
library(FSA)
library(rms)
library(Stat2Data)
library(MASS)
library(ggplot2)
library(car)
library(GGally)
library(caret)
library(RColorBrewer)
library(glmnet)
library(bestglm)

set.seed(0)
```

```
maternal.df <- read_csv("/home/cosmos/Documents/School/MTH 245/Project/Maternal Health Risk
```

```
glimpse(maternal.df)
```

```
## Rows: 1,014
## Columns: 7
## $ Age         <dbl> 25, 35, 29, 30, 35, 23, 23, 35, 32, 42, 23, 19, 25, 20, 48~
## $ SystolicBP  <dbl> 130, 140, 90, 140, 120, 140, 130, 85, 120, 130, 90, 120, 1~
## $ DiastolicBP <dbl> 80, 90, 70, 85, 60, 80, 70, 60, 90, 80, 60, 80, 89, 75, 80~
## $ BS          <dbl> 15.00, 13.00, 8.00, 7.00, 6.10, 7.01, 7.01, 11.00, 6.90, 1~
## $ BodyTemp    <dbl> 98, 98, 100, 98, 98, 98, 98, 102, 98, 98, 98, 98, 98, 100,~
## $ HeartRate   <dbl> 86, 70, 80, 70, 76, 70, 78, 86, 70, 70, 76, 70, 77, 70, 88~
## $ RiskLevel   <chr> "high risk", "high risk", "high risk", "high risk", "low r~
```

```
summary_stats <- maternal.df %>%
  reframe(mean = c(mean(SystolicBP), mean(Age),
                   mean(DiastolicBP), mean(BS),
                   mean(BodyTemp), mean(HeartRate)),
          median = c(median(SystolicBP), median(Age),
                     median(DiastolicBP), median(BS),
                     median(BodyTemp), median(HeartRate)),
          min = c(min(SystolicBP), min(Age),
                  min(DiastolicBP), min(BS),
                  min(BodyTemp), min(HeartRate)),
         max = c(max(SystolicBP), max(Age),
                 max(DiastolicBP), max(BS),
                 max(BodyTemp), max(HeartRate)))
```

Parameter	$\bar{x}$	Median	Min	Max
SystolicBP	113.18	120	70	160
Age	29.90	26	10	70
DiastolicBP	76.46	80	49	100
BS	8.73	7.50	6	19
BodyTemp	98.67	98	98	103
HeartRate	74.43	76	60	90

Table 1: Summary statistics of maternal.df.

There are 1,014 pregnant women within the data set, and they have 7 different factors listed for each of them. According to Table 1, the ages of the participants ranged from 10 years old all the way to 70, with the median age being 26 years old. The median systolic blood pressure for the data set was 120. The median diastolic blood pressure was 80, and median blood sugar was 7.5, with a range from 6 to 19. Body Temperature was consistently around 98, with a maximum of 103. Finally, the median and mean heart rate was 76 and 74.3, respectively. However, there were two values that displayed '7' as the heart rate, which could not have been true because the normal range for humans is between 60 and 100. So, we decided to take the rows that contained those values out of the dataset entirely.

```
maternal.df <- maternal.df %>% filter(HeartRate != 7)

summary_stats <- maternal.df %>%
  reframe(mean = c(mean(SystolicBP), mean(Age),
                   mean(DiastolicBP), mean(BS),
                   mean(BodyTemp), mean(HeartRate)),
          median = c(median(SystolicBP), median(Age),
                     median(DiastolicBP), median(BS),
                     median(BodyTemp), median(HeartRate)),
          min = c(min(SystolicBP), min(Age),
                  min(DiastolicBP), min(BS),
                  min(BodyTemp), min(HeartRate)),
         max = c(max(SystolicBP), max(Age),
```

```
                max(DiastolicBP), max(BS),
                max(BodyTemp), max(HeartRate)))
```

Parameter	$\bar{x}$	Median	Min	Max
SystolicBP	113.18	120	70	160
Age	29.90	26	10	70
DiastolicBP	76.46	80	49	100
BS	8.73	7.50	6	19
BodyTemp	98.67	98	98	103
HeartRate	74.43	76	60	90

Table 2: Summary statistics after data cleaning.

After our minimal data cleaning, our median heart rate stayed the same, but the mean heart rate increased to 74.43, as shown in Table 2, with now a minimum rate of 60, which is a lot more likely. Because we removed the entire row, this also caused some other values from Table 1 to change. The mean systolic blood pressure decreased from 113.20 to 113.18. The mean age also increased from 29.87 to 29.9. All other values within the summary statistics table stayed the same.

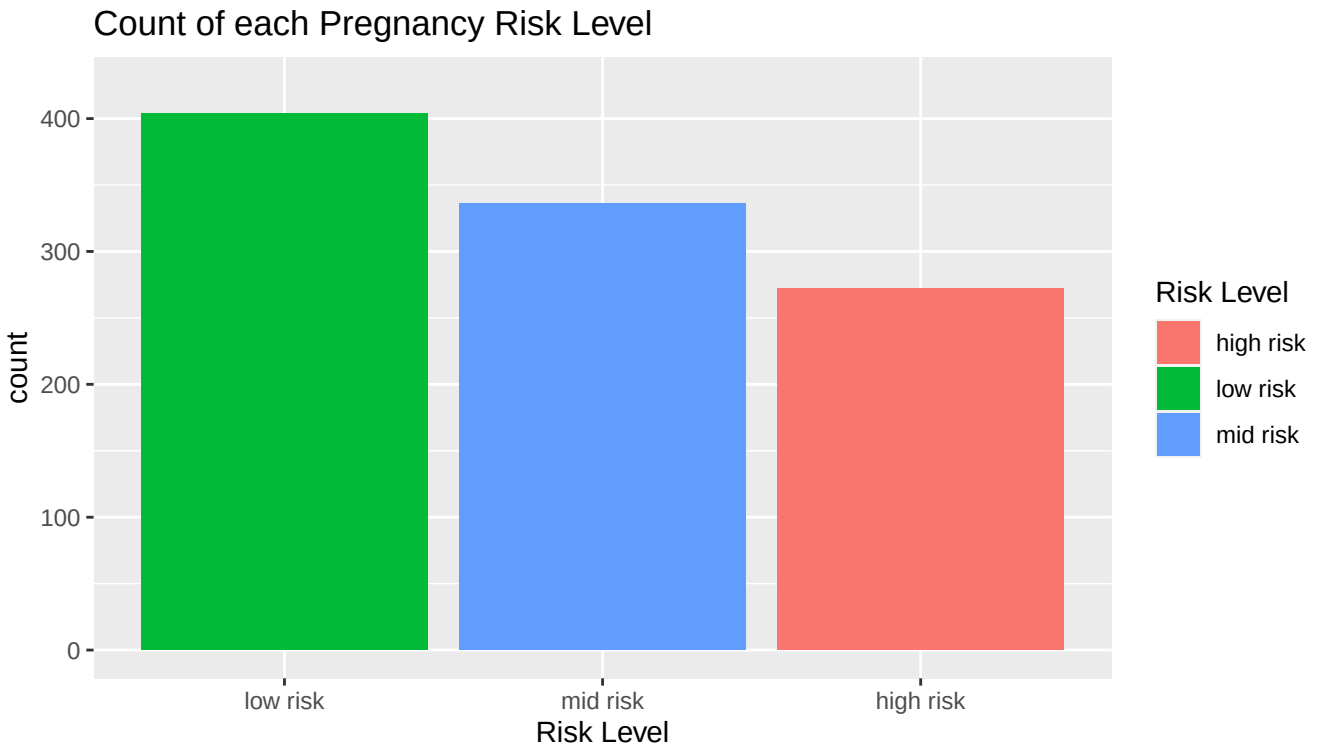

Figure 1: Bar chart of each pregnancy Risk Level's count.

According to the bar chart shown in Figure 1, a low risk pregnancy was found to be the most common within the data, with just over 400 participants being placed into this category. Next was

a mid level risk, which had under 350 participants. Finally, the smallest pregnancy risk level were those who had a high risk pregnancy. There were around 275 participants in this category.

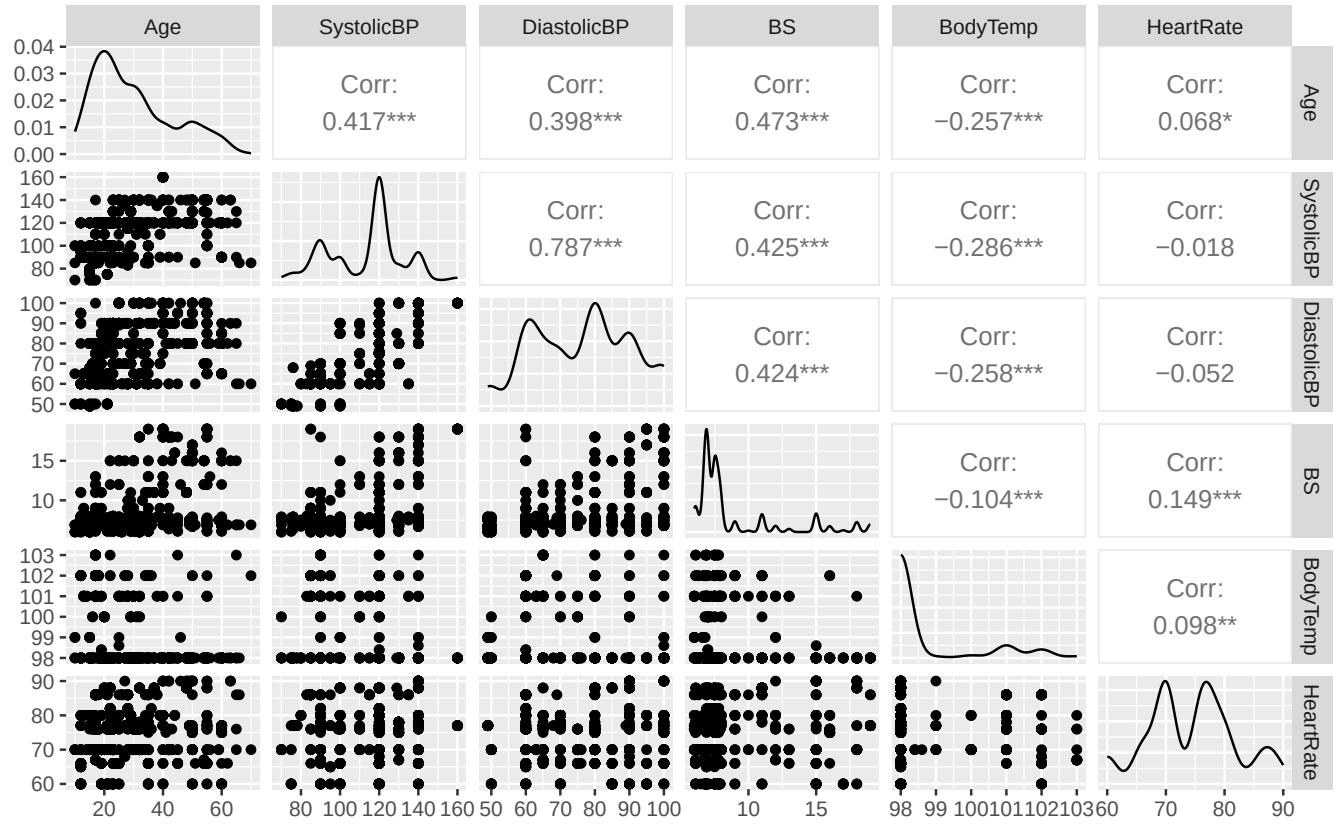

Figure 2: A correlogram comparing the relationships between the quatitative predictor variables by the risk levels (low, medium, and high) from the maternal health risk dataset.

```
cor(maternal.df$Age, maternal.df$SystolicBP)
```

```
## [1] 0.4172921
```

Age and systolic blood pressure have a correlation coefficient of approximately 0.4163. This indicates a moderate positive linear relationship, where as age increases, there tends to be a moderate increase in systolic blood pressure on average. It is important to note that correlation does not imply causation.

```
cor(maternal.df$DiastolicBP, maternal.df$SystolicBP)
```

```
## [1] 0.7871984
```

Diastolic blood pressure and systolic blood pressure have a correlation coefficient of approximately 0.7872. This indicates a strong positive linear relationship, where as diastolic blood pressure increases, there tends to be a strong increase in systolic blood pressure on average. It is important to note that correlation does not imply causation.

```
cor(maternal.df$BS, maternal.df$SystolicBP)
```

```
## [1] 0.425439
```

Blood sugar level and systolic blood pressure have a correlation coefficient of approximately 0.4254. This indicates a moderate positive linear relationship, where as blood sugar level increases, there tends to be a moderate increase in systolic blood pressure on average. It is important to note that correlation does not imply causation.

```
cor(maternal.df$BodyTemp, maternal.df$SystolicBP)
```

```
## [1] -0.2863663
```

Body temperature and systolic blood pressure have a correlation coefficient of approximately -0.2863. This indicates a small negative linear relationship, where as body temperature increases, there tends to be a small decrease in systolic blood pressure on average. It is important to note that correlation does not imply causation.

```
cor(maternal.df$HeartRate, maternal.df$SystolicBP)
```

```
## [1] -0.01832823
```

Heart rate and systolic blood pressure have a correlation coefficient of approximately -0.0183. This indicates a very small negative linear relationship, where as heart rate increases, there tends to be a very small decrease in systolic blood pressure on average. It is important to note that correlation does not imply causation.

```
maternal.z <- maternal.df %>%
  mutate(
    SystolicBP.z = scale(SystolicBP, center = TRUE, scale = TRUE),
    Age.z = scale(Age, center = TRUE, scale = TRUE),
    DiastolicBP.z = scale(DiastolicBP, center = TRUE, scale = TRUE),
    BS.z = scale(BS, center = TRUE, scale = TRUE),
    BodyTemp.z = scale(BodyTemp, center = TRUE, scale = TRUE),
    HeartRate.z = scale(HeartRate, center = TRUE, scale = TRUE),
    RiskLevellow_risk = ifelse(RiskLevel == "low risk", 1, 0),
    RiskLevelmid_risk = ifelse(RiskLevel == "mid risk", 1, 0)
  )
```

We then decided to standardize our quantitative variables to make it easier to compare our results to one another when examining the test statistic and p-values later on in our analyses.

4 Methods

After our exploratory data analysis we began to work through the methods portion of our project. We begin by creating a first order model and then evaluate the assumptions for regression. Then, we test various model selection techniques to find the best fit model for our research questions.

4.1 Fitting an ANOVA

We decided to then fit an ANOVA regression using our first order linear model to examine which variables would most likely be significant when predicting systolic blood pressure. Below is our ANOVA for regression hypotheses:

$$H_0: \ \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 0$$
$$H_a: \ \text{at least one } \beta_j \neq 0, \ j = 1, 2, 3, ..., 6$$

```
anova.model <-
  anova(ols(SystolicBP.z ~ Age.z + DiastolicBP.z + BS.z +
              BodyTemp.z + HeartRate.z + RiskLevel, data=maternal.z))
```

Parameter	d.f.	Partial SS	MS	F	p-value
Age.z	1	3.7488	3.7488	11.0724	< 0.0001
DiastolicBP.z	1	317.8381	317.8381	938.7668	< 0.0001
BS.z	1	0.5222	0.5222	1.5425	0.2145
BodyTemp.z	1	11.6088	11.6088	34.2879	< 0.0001
HeartRate.z	1	0.2629	0.2629	0.7766	0.3784
RiskLevel	2	22.5218	11.2609	33.2602	< 0.0001
TOTAL	7	672.3988	96.0570	283.7140	< 0.0001
ERROR	1006	340.6012	0.3386		

Table 3: ANOVA table for regression.

From the ANOVA Regression Table (Table 3), there is sufficient evidence that Age, Diastolic Blood Pressure, Body Temperature, and Pregnancy Risk Level (p-value < 0.0001 for all listed variables) are strong predictors for Systolic Blood Pressure. There is also sufficient evidence that Blood Sugar (p-value $= 0.2145$) and Heart Rate (p-value $= 0.3784$) are not strong predictors for Systolic Blood Pressure. Since this an ANOVA, the results only give us indicators of prediction and not specifically what these indicators mean within the linear model and how much weight each factor has on the model.

4.2 First-Order Model and Model Selection

```
one.way <- lm(SystolicBP.z ~ Age.z + DiastolicBP.z + BS.z + BodyTemp.z +
                HeartRate.z + RiskLevel, data=maternal.z)

summary(one.way)$adj.r.squared
```

```
## [1] 0.6619938
```

Parameter	Estimate	Std. Error	t-value	p-value
(Intercept)	0.1353	0.0450	3.0046	0.0027
Age.z	0.0736	0.0219	3.3683	< 0.0001
DiastolicBP.z	0.6754	0.0220	30.6702	< 0.0001
BS.z	0.0306	0.0260	1.1765	0.2397
BodyTemp.z	-0.1192	0.0203	-5.8581	< 0.0001
HeartRate.z	-0.0057	0.0189	-0.2989	0.7651
RiskLevellow risk	-0.3412	0.0619	-5.5128	< 0.0001
RiskLevelmid risk	0.0029	0.0585	0.0494	0.9606

Table 4: Regression summary with standardized predictors.

From Table 4 the linear regression formula for this model gives a clear view for how important each variable is when predicting systolic blood pressure. A larger t-value, whether it be positive or negative indicates a stronger predictor variable than a smaller number that is closer to zero. In this way, Diastolic Blood Pressure has the highest predictive value (t-value = 30.6702, p-value $<$ 0.0001) and a medium pregnancy risk level has the lowest predictive value (t-value = 0.0494, p-value = 0.9696). For example, for every one standard deviation that diastolic blood pressure increases, systolic blood pressure increases by 0.6754 standard deviations on average. On the other other hand, if someone is dictated to have a medium risk level to their pregnancy, their systolic blood pressure is expected to increase by 0.0029 standard deviations on average. Within this model, the variables that are expected to increase systolic blood pressure include, age, diastolic blood pressure, body sugar, and a medium risk level versus a high risk level. The variables expected to decrease systolic blood pressure include an increase in body temperature, heart rate, and having a low risk level as compared to a high pregnancy risk level. The adjusted R^2 value for this model is 0.6620. This means that about 66.20% of the variability of systolic blood pressure in pregnant women is explained by this linear regression model.

```
source("https://cipolli.com/students/code/plotResiduals.R")
plotResiduals(one.way)
```

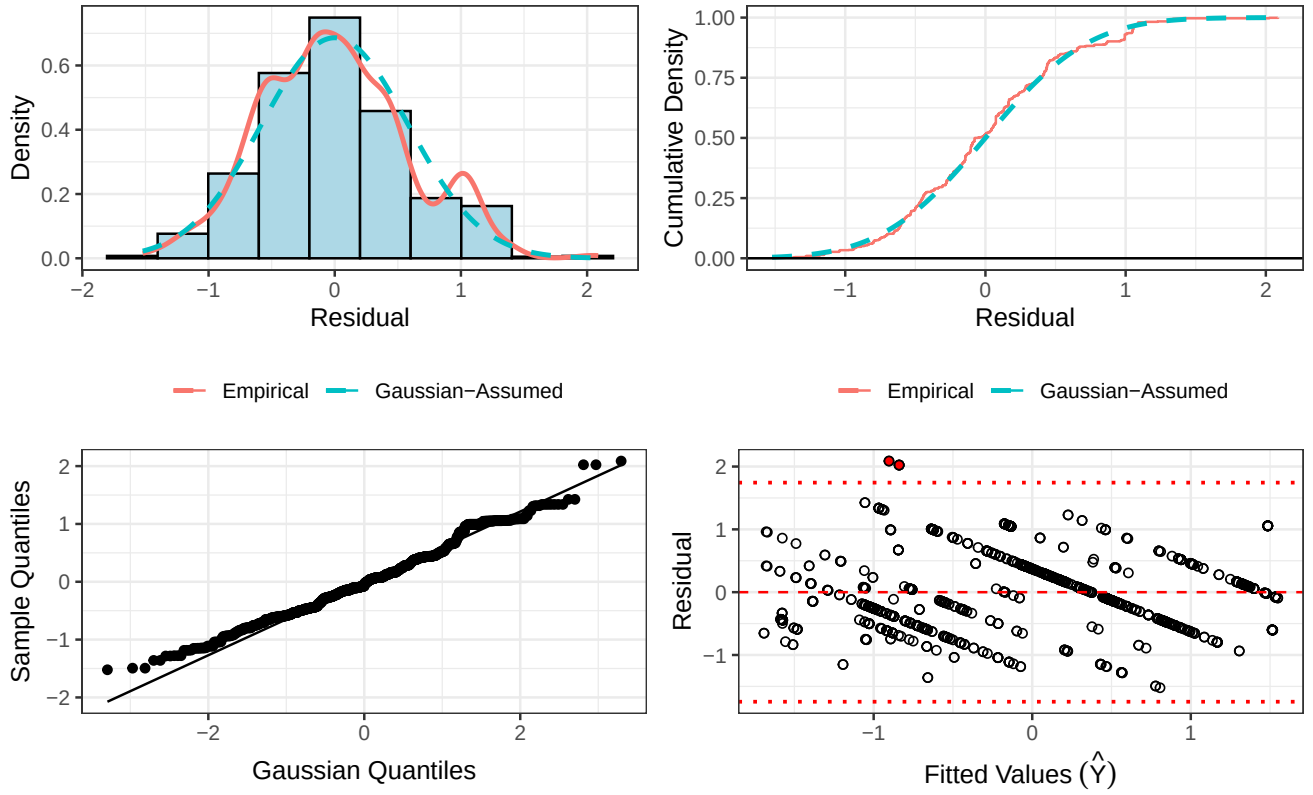

Figure 3: Diagnostic plots using age, diastolic blood pressure, blood sugar levels, body temperature, heart rate, and pregnancy risk level to predict systolic blood pressure.

Assumptions for regression appear to be met. Firstly, the response variable, systolic blood pressure is continuous quantitative. Meaning, a person's systolic blood pressure can be a wide range of numerical values within a range of plausible blood pressures. Systolic blood pressure is measured in millimeters of mercury (mmHg) and vary continuously within a range. Regarding the sample being representative of the population, we are assuming it is fairly representative, however, we would like to contact the researchers who collected the data to ask them about their collection process. It is unclear how random the sample is, so we may want to assume this was a convenience sample. However, the data was collected from various hospitals, community clinic, and maternal health cares from rural areas in Bangladesh so we can fairly confidently say the sample is representative to pregnant women in rural Bangladesh. But, we would not further generalize to all pregnant women.

Additionally, we can assume that the sample observations are independent. One pregnant woman's systolic blood pressure should not impact another pregnant woman's systolic blood pressure and vise versa. Each pregnant woman is representative of her individual pregnancy case because her systolic blood pressure that was measured at a certain point in time, would not be impacted by another patient's.

According to the plot residuals in Figure 3, the model predicting systolic blood pressure with all other variables is approximately normal. Although there are slight deviations from perfect

normality, we decided not to make any transformations to the data so that the we can have the best interpretability for our models. There also appears to only be 2 outliers on our fitted values plot, which shows that our model is most likely acceptable to use. Overall, the residuals appear to have relatively constant variance, are approximately normal distributed, and the predictor variables do not appear to be heavily correlated.

```
intercept.model <- lm(SystolicBP.z ~ 1, data = maternal.z)

one.way.forward <- stepAIC(intercept.model,
        direction = "forward",
        scope = list(upper = one.way),
        trace = FALSE)

one.way.backward <- stepAIC(one.way,
        direction = 'backward',
        trace=FALSE)

one.way.both <- stepAIC(intercept.model,
        direction = "both",
        scope = list(upper = one.way),
        trace = FALSE)
```

```
metrics<-data.frame(Model=c("1", "2", "3", "4"),
                    R.squared = c(summary(one.way)$r.squared,
                                  summary(one.way.backward)$r.squared,
                                  summary(one.way.forward)$r.squared,
                                  summary(one.way.both)$r.squared),
                    R.adj.squared = c(summary(one.way)$adj.r.squared,
                                      summary(one.way.backward)$adj.r.squared,
                                      summary(one.way.forward)$adj.r.squared,
                                      summary(one.way.both)$adj.r.squared),
                    LL=c(logLik(one.way),
                         logLik(one.way.backward),
                         logLik(one.way.forward),
                         logLik(one.way.both)),
                    AIC=c(AIC(one.way), AIC(one.way.backward),
                          AIC(one.way.forward),
                          AIC(one.way.both)),
                    BIC=c(BIC(one.way), BIC(one.way.backward),
                          BIC(one.way.forward),
                          BIC(one.way.both)),
                    RMSE=c(sqrt(mean(sum(residuals(one.way)^2))),
                           sqrt(mean(sum(residuals(one.way.backward)^2))),
                           sqrt(mean(sum(residuals(one.way.forward)^2))),
                           sqrt(mean(sum(residuals(one.way.both)^2)))),
                    MAE=c(mean(sum(abs(residuals(one.way)))),
                          mean(sum(abs(residuals(one.way.backward)))),
                          mean(sum(abs(residuals(one.way.forward)))),
```

```
                    mean(sum(abs(residuals(one.way.both)))),
                Parameters=c("8","6", "6", "6"))
```

Model	R^2	R^2_{adj}	LL	AIC	BIC	RMSE	MAE	Parameters
One-Way	0.6643	0.6620	-883.0963	1784.1926	1828.4697	18.4217	467.4012	8
One-Way Forward	0.6639	0.6622	-883.8189	1781.6378	1816.0756	18.4348	467.5434	6
One-Way Backward	0.6639	0.6622	-883.8189	1781.6378	1816.0756	18.4348	467.5434	6
One-Way Step-wise	0.6639	0.6622	-883.8189	1781.6378	1816.0756	18.4348	467.5434	6

Table 5: A table of model metrics for our first order models.

Both the backwards, forward, and step-wise model selections on the first order model select the same subset of predictors, which ends up making each factor the same numbers. To simplify our analysis, we will just look at the linear regression model using backwards selection.

Parameter	Estimate	Std. Error	t-value	p-value
(Intercept)	0.1594	0.0388	4.1140	< 0.0001
Age.z	0.0813	0.0207	3.9199	< 0.0001
DiastolicBP.z	0.6794	0.0216	31.3999	< 0.0001
BodyTemp.z	-0.1217	0.0202	-6.0167	< 0.0001
RiskLevellow risk	-0.3761	0.0521	-7.2186	< 0.0001
RiskLevelmid risk	-0.0279	0.0509	-0.5489	0.5832

Table 6: Regression summary using model selection.

The model selection chose Age.z, DiastolicBP.z, BodyTemp.z, and Risk Level as the highest predictors of Systolic Blood Pressure. The adjusted R^2 value is now 0.6622, which means that 66.22%of the variability of Systolic Blood Pressure is explained by this model. The adjusted R^2 value increased very slightly from the regular first order model. This means a bit more of Systolic Blood Pressure's variability is explained through the subset of predictors chosen through model selection. Since the number of predictor variables within the linear regression model has also decreased, this can also be an indicator that the model is stronger overall. In this linear regression model, an increase in age and diastolic blood pressure causes the prediction of systolic blood pressure to increase, while an increase in body temperature and having a low or medium risk level as compared to a high pregnancy risk level causes the prediction of systolic blood pressure to decrease. Diastolic blood pressure is still the highest predictor of systolic blood pressure (t-value = 31.40, p-value = $<$ 0.0001).

```
plotResiduals(one.way.backward)
```

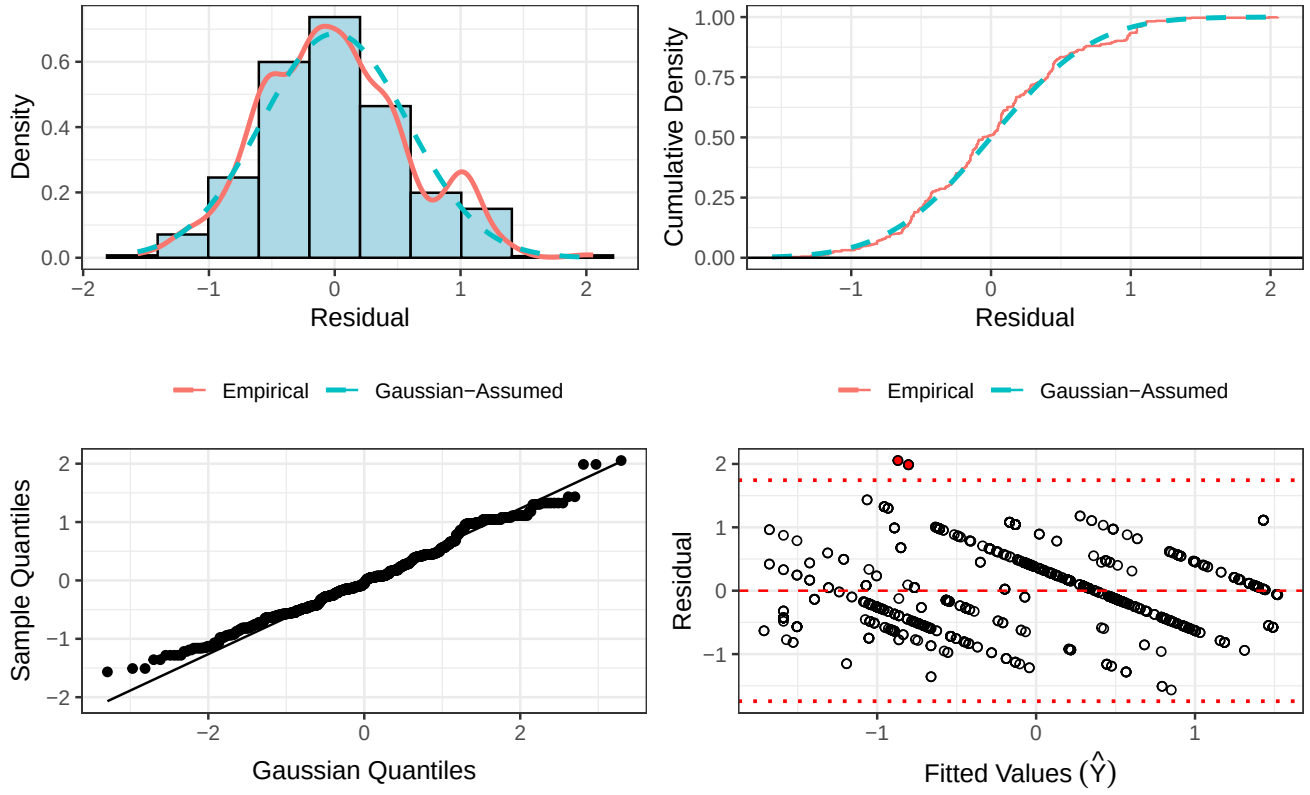

Figure 4: Diagnostic plots model created through model selection.

According to Figure 4 the data regression model after model selection is still approximately normal. The plots still have the same divots as before and look very similar to one another. The vast majority of values are within the residuals, besides the two outliers from before. The residuals also fit pretty nicely on the cumulative density line.

4.3 Interaction Analysis

To get a better picture of the relationship between our parameters and systolic blood pressure, we also want to consider all interactions between variables. We do this in case, for example, diastolic blood pressure is not an important factor for low-risk individuals, but it is for individuals of moderate risk. In which case, it would be beneficial to know the impact of that relation on observed systolic blood pressure. We call this model the two-way interaction all variables model, or two-way for short. This is achieved with the following lines:

```
two.way <- lm(SystolicBP.z ~ (Age.z + DiastolicBP.z + BS.z + BodyTemp.z +
                              HeartRate.z + RiskLevel)^2, data=maternal.z)
```

And now we conduct an ANOVA test to evaluate significance of interactions:

```
anova(two.way)
```

Parameter	Df	Sum Sq	Mean Sq	F value	p-value
Age.z	1	175.34	175.34	581.23	< 0.0001
DiastolicBP.z	1	464.81	464.81	1540.74	< 0.0001
BS.z	1	4.22	4.22	14.00	0.0002
BodyTemp.z	1	5.50	5.50	18.23	< 0.0001
HeartRate.z	1	0.00	0.00	0.01	0.9425
RiskLevel	2	22.52	11.26	37.33	< 0.0001
Age.z:DiastolicBP.z	1	3.08	3.08	10.21	0.0014
Age.z:BS.z	1	0.65	0.65	2.14	0.1434
Age.z:BodyTemp.z	1	3.12	3.12	10.34	0.0013
Age.z:HeartRate.z	1	2.41	2.41	7.99	0.0048
Age.z:RiskLevel	2	0.49	0.25	0.81	0.4441
DiastolicBP.z:BS.z	1	12.44	12.44	41.24	< 0.0001
DiastolicBP.z:BodyTemp.z	1	4.57	4.57	15.15	0.0001
DiastolicBP.z:HeartRate.z	1	0.07	0.07	0.22	0.6361
DiastolicBP.z:RiskLevel	2	11.66	5.83	19.33	< 0.0001
BS.z:BodyTemp.z	1	1.10	1.10	3.66	0.0560
BS.z:HeartRate.z	1	0.49	0.49	1.63	0.2020
BS.z:RiskLevel	2	0.59	0.29	0.98	0.3772
BodyTemp.z:HeartRate.z	1	0.02	0.02	0.06	0.8032
BodyTemp.z:RiskLevel	2	2.20	1.10	3.65	0.0262
HeartRate.z:RiskLevel	2	0.25	0.13	0.42	0.6604
Residuals	986	297.45	0.30		

Table 7: ANOVA table for regression of two-way interaction parameters.

From the ANOVA test in Table 7, we find that there are twelve out of twenty-one statistically significant parameters. Most importantly, Age and DiastolicBP, with F values of 175.34 and 464.81, respectively. Additionally, we find that there are several interactions with little importance. Three that stand out are HeartRate (F value = 0.01, p-value = 0.9425), the interaction between DiastolicBP and HeartRate (F value = 0.22, p-value = 0.6361), and the interaction between BodyTemp and HeartRate (F value = 0.06, p-value = 0.8032). These findings corroborate earlier indications that DiastolicBP and Age are strong predictors of SystolicBP, while HeartRate is not.

Next, a look at model's summary statistics:

From the summary of the two-way interaction model, we see that in fact diastolic blood pressure is a statistically significant factor to consider for individuals of moderate risk (t-value = -3.2794; p-value = 0.0011), whereas for low-risk individuals diastolic blood pressure is not statistically significant (t-value = 0.9841; p-value = 0.3253). The reverse is true for body temperature. For individuals of low risk, body temperature is statistically significant (t-value = 2.412; p-value = 0.01605), whereas it is not for moderate risk (t-value = 0.892; p-value = 0.37237).

Interestingly, low risk on its own is an important factor but not moderate risk. Low risk has a t-value of -2.409 and p-value of 0.01617 while moderate risk has a t-value of -0.697 and a p-value of 0.48593. It is only in the presence of other parameters that moderate risk is relevant.

It is also important to note that there are nine statistically significant parameters. They are: intercept, diastolic blood pressure, body temperature, low risk level, the interaction between age

Parameter	Estimate	Std. Error	t-value	p-value
(Intercept)	0.2247	0.0671	3.3503	< 0.001
Age.z	0.0278	0.0631	0.4409	0.6594
DiastolicBP.z	0.6876	0.0536	12.8336	< 0.001
BS.z	-0.0292	0.0478	-0.6117	0.5409
BodyTemp.z	-0.1186	0.0481	-2.4677	0.0138
HeartRate.z	-0.0777	0.0450	-1.7261	0.0846
RiskLevellow risk	-0.4769	0.0947	-5.0361	< 0.001
RiskLevelmid risk	-0.1204	0.0788	-1.5271	0.1271
Age.z:DiastolicBP.z	-0.1155	0.0258	-4.4783	< 0.001
Age.z:BS.z	0.0270	0.0346	0.7798	0.4357
Age.z:BodyTemp.z	0.0355	0.0233	1.5226	0.1282
Age.z:HeartRate.z	0.0348	0.0237	1.4652	0.1432
Age.z:RiskLevellow risk	0.0120	0.0806	0.1492	0.8814
Age.z:RiskLevelmid risk	0.1292	0.0809	1.5975	0.1105
DiastolicBP.z:BS.z	0.1091	0.0330	3.3033	0.0010
DiastolicBP.z:BodyTemp.z	0.1158	0.0248	4.6675	< 0.001
DiastolicBP.z:HeartRate.z	0.0017	0.0234	0.0719	0.9427
DiastolicBP.z:RiskLevellow risk	0.0716	0.0727	0.9841	0.3253
DiastolicBP.z:RiskLevelmid risk	-0.2452	0.0748	-3.2794	0.0011
BS.z:BodyTemp.z	-0.0332	0.0355	-0.9370	0.3490
BS.z:HeartRate.z	0.0380	0.0275	1.3783	0.1684
BS.z:RiskLevellow risk	-0.1891	0.1545	-1.2240	0.2212
BS.z:RiskLevelmid risk	-0.0165	0.0697	-0.2367	0.8129
BodyTemp.z:HeartRate.z	0.0090	0.0231	0.3893	0.6971
BodyTemp.z:RiskLevellow risk	0.1577	0.0654	2.4120	0.0160
BodyTemp.z:RiskLevelmid risk	0.0552	0.0618	0.8925	0.3724
HeartRate.z:RiskLevellow risk	0.0572	0.0627	0.9125	0.3617
HeartRate.z:RiskLevelmid risk	0.0361	0.0582	0.6205	0.5351

Table 8: Two way interaction model results.

and diastolic blood pressure, the interaction between diastolic blood pressure and blood sugar levels, the interaction between diastolic blood pressure and body temperature, the interaction between diastolic blood pressure and moderate risk level, and the interaction between body temperature and low risk level. Overall, the most important interaction is the interaction between diastolic blood pressure and age (t-value $= -4.478$; p-value < 0.0001).

4.4 Model Selection

As seen in the previous sections, most (19 of 28) two-way interaction parameters are not statistically significant. Leaving all of these parameters in the model is fine for maximizing R^2, but in general it is considered a pretty poor model. This belief is due to the fact that R^2 always improves with additional parameters, even if those parameters poorly explain the response variable. In addition to R^2, we consider nine other measurements that can be used to evaluate and interpret the performance of a linear model. Going forward, these measurements will simply be known as 'metrics'. The nine additional metrics are as follows: R^2_{adj}, log likelihood (LL), AIC, BIC, RMSE, MAE, LOOCV R^2,

LOOCV RMSE, and LOOCV MAE. Note that we excluded the number of parameters because that is already captured by R^2_{adj}, AIC, BIC, and LOOCV R^2. Including the number of parameters along with those factors would be redundant and place a larger emphasis on smaller models than we believe is necessary. The reason we include these additional nine metrics is to give us a holistic understanding of each model. Each model performs similarly across most metrics, so including more metrics gives us more confidence in which model is the "best", rather than being a randomly good at one or two metrics.

We choose to conduct Leave-One-Out Cross-Validation (LOOCV) on R^2, RMSE, and MAE as well to get an idea of how each model would perform in a real world setting. For the other seven metrics, all of the data is used to calculate the metric value. However, using the same data to both train and test a model can lead to overfitting and an inaccurate picture of accuracy on new data. Cross-Validation helps to remedy this but splitting the data into two parts: one part for training the model, which we can then calculate the metrics from, and a second part to test the model, which gives us an idea of real world performance. LOOCV takes this a step further and sets aside only one data point at a time to test the model. This means all of the data except one is used to train the model, and it gets one chance to predict an unseen data point. This is repeated such that every single data point has its turn to be left out, and the performance metrics are averaged across every test. This method offers a robust way to evaluate a model's performance on unseen data. For this reason, we value these three metrics highly. We do not feel confident in assigning a specific weight to each of these metrics (i.e. CV R^2 is twice as important as R^2), so we do not. However, this hightened importance is something we will keep in mind going forward.

In addition to these extra metrics, we want to consider more models. Like with the one-way interaction models, we want to include forward, backward, and step-wise regression models for the two-way interactions. These models are simple to implement, fast to run, and often competitive with more complex models. Also, the assumptions are the same for two-way interactions as one-way ones, so we can proceed. We implement that code here:

```
two.way.backward <- stepAIC(two.way,
                            direction = 'backward',
                            trace=FALSE)

two.way.forward <- stepAIC(intercept.model,
                           direction = 'forward',
                           scope = list(upper = two.way),
                           trace=FALSE)

two.way.both <- stepAIC(intercept.model,
                        direction = 'both',
                        scope = list(upper = two.way),
                        trace=FALSE)
```

This brings our total number of models to eight: one- and two-way interactions for all parameters as well as forward, backward, and step-wise regressions. Like before, however, some of these two-way models are redundant. The metric values for each can be seen in Table 8.

Checking the metrics, we find again that backward, forward, and step-wise regression produce the same models. Therefore, we will be excluding forward and step-wise regression. Because we did the

Model	R^2	R^2_{adj}	LL	AIC	BIC	RMSE
Two-Way	0.7064	0.6983	-817.0209	1692.0418	1834.7699	0.5416
Two-Way Backward	0.7043	0.6992	-820.6264	1679.2528	1772.7643	0.5435
Two-Way Forward	0.7043	0.6992	-820.6264	1679.2528	1772.7643	0.5435
Two-Way Both	0.7043	0.6992	-820.6264	1679.2528	1772.7643	0.5435

Model	MAE	CV R^2	CV RMSE	CV MAE	Parameters
Two-Way	0.4135	0.6883	0.5582	0.4254	28
Two-Way Backward	0.4160	0.6935	0.5534	0.4232	18
Two-Way Forward	0.4160	0.6935	0.5534	0.4232	18
Two-Way Both	0.4160	0.6935	0.5534	0.4232	18

Table 9: Metrics for two-way interaction models.

same for one-way interactions, we now only have four models: all variables and backwards regression for one- and two-way interactions.

Having just four models is okay, but we would really like to have more to compare to. Just off of our intuition and previous work, it is likely one-way all variables, one-way backward regressions, and two-way all variables are not great. They can be good, but they are often too basic to perform well on all the metrics we chose. Two-way backward regression could be quite good, but there are lots of other parameter selection techniques out there. Relying on just two-way backward regression to be the best model for predicting systolic blood pressure does not leave us with a lot of confidence. To remedy this, we chose five more parameter selection techniques: LASSO and Ridge regression on one- and two-way interactions, and best subset for one-way interactions. These additional models will give us a much better understanding of what a "good" and "bad" model look like for this data. We chose to omit two-way best subset due to the computation time required to evaluate all 134 million subsets of 28 parameters. Lastly, it is important to note that while these models have no additional assumptions, LASSO and Ridge regression can introduce additional bias.

LASSO (Least Absolute Shrinkage and Selection Operator) regression and Ridge regression are similar. Both parameter selection techniques introduce regularization to help with multicollinearity, overfitting, and interpretability. This is done through the use of a λ tuning parameter, which the optimal value is found via cross-validation. While Ridge regression looks to minimize the sum of the squares of the regression coefficients, LASSO minimizes the sum of the absolute values of the regression coefficients. This difference means LASSO regression can set some parameters to have a coefficient of zero, while Ridge regression can only make the coefficient very small. For this reason, Ridge regression is less of a parameter selection technique and more of a coefficient optimization strategy. LASSO regression on the other hand performs both - optimizing parameter selection and the coefficients of each of those parameters. The code for both is similar, and we implement each for one- and two-way interactions here:

```
# -- one-way ridge --
one.x <- model.matrix(one.way)[, -1]
one.y <- maternal.z$SystolicBP.z

one.way.ridge <- glmnet(one.x, one.y, alpha = 0)
```

```r
# -- one-way LASSO --
one.way.lasso <- glmnet(one.x, one.y, alpha = 1)

# -- two-way ridge --
two.x <- model.matrix(two.way)[, -1]
two.y <- maternal.z$SystolicBP.z

two.way.ridge <- glmnet(two.x, two.y, alpha = 0)

# -- two-way LASSO --
two.way.lasso <- glmnet(two.x, two.y, alpha = 1)
```

Before we continue to best subsets, it can be useful to compare the values of λ for each LASSO model as the number of parameters changes. In doing so, we will have a better understanding of the complexities of each model, and can be alerted of potential impacts on bias by large λ values. We implement that as so:

```r
# Function to create a tidy data frame from a glmnet object
extract_coefs <- function(model, model_name) {
  lambda <- model$lambda
  coefs <- coef(model, s = lambda)

  # Initialize a vector to store the number of non-zero coefficients
  num_nonzero <- numeric(length(lambda))

  for (i in seq_along(lambda)) {
    # Count non-zero coefficients for each lambda
    num_nonzero[i] <- sum(coefs[, i] != 0)
  }

  # Subtract 1 to account for the intercept
  data.frame(lambda = lambda, num_nonzero = num_nonzero - 1, model = model_name)
}

# Extracting data
one_way_lasso_data <- extract_coefs(one.way.lasso, "One-Way LASSO")
two_way_lasso_data <- extract_coefs(two.way.lasso, "Two-Way LASSO")

all_data <- rbind(one_way_lasso_data, two_way_lasso_data)
```

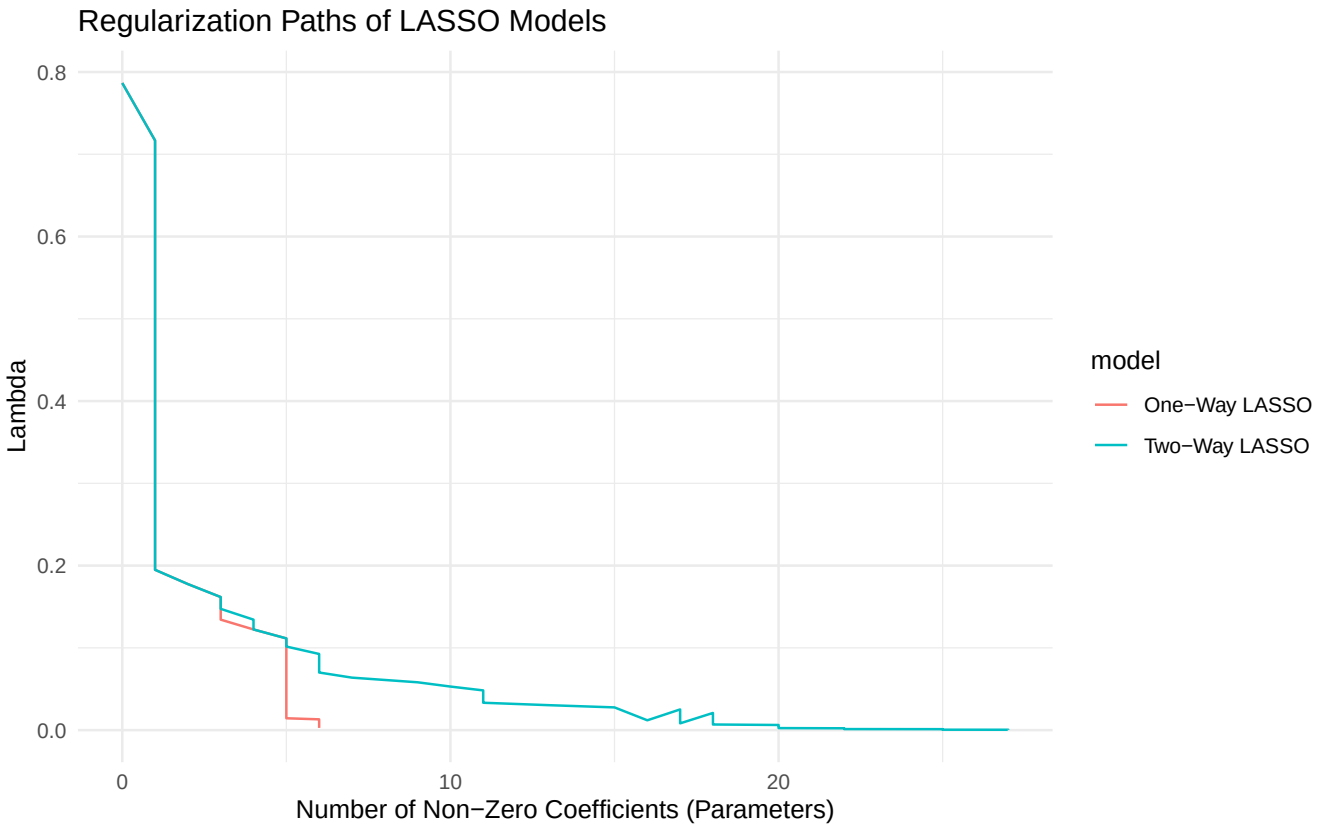

Figure 5: λ versus number of parameters for one- and two-way LASSO regressions.

The comparison of λ's is interesting, but hard to see the differences for larger numbers of parameters. We fix this by plotting the logarithm of λ instead:

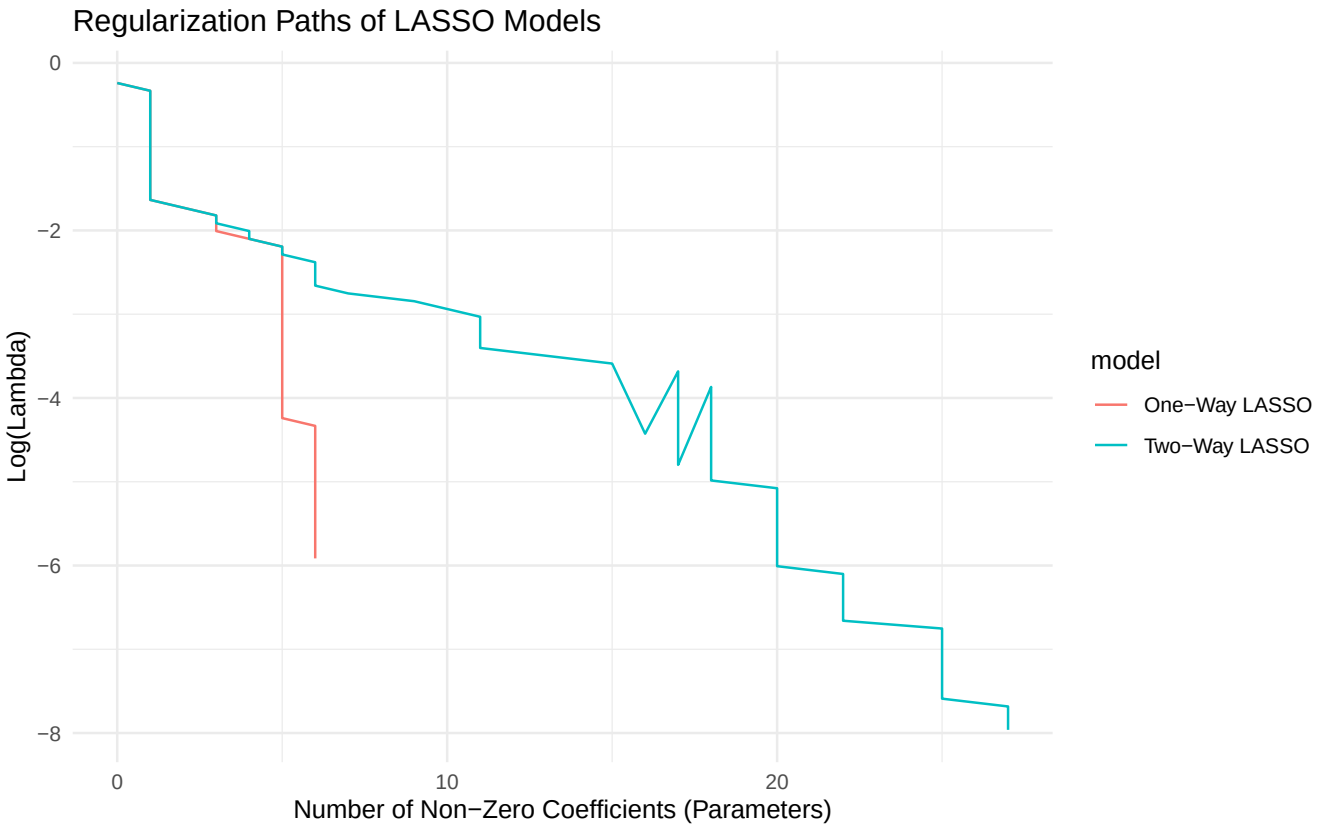

Figure 6: λ versus number of parameters for one- and two-way LASSO regressions.

This graph is much better as it more clearly shows the impact on λ as the number of parameters increases. Very large values for λ do not appear to occur, so there are no worries of extra bias being added to the model. We can now proceed to best subsets.

For best subsets, rather than using statistical strategies to find a subset of parameters that are close to the best subset for explaining the response variable, an exhaustive search is used to find the truly optimal subset. This means the number of subsets that need to be checked grows exponentially. Specifically, the computation time of best subsets is $O(2^n)$, where n is the number of parameters available. The reason for a $O(2^n)$ computation time is because best subset is a binary selection. Each parameters is either chosen (1), or not (0), which can be represented as a binary number. Each digit in this binary number represents whether a parameter is included in the model. Thus, the binary number has a length of n. There are 2^n numbers in a binary number of length n, so 2^n subsets of parameters must be checked. For small sets of parameters, such as one-way all parameters, this is feasible. With six parameters, only 64 different subsets need to be checked. This can be accomplished in less than a second. However, for larger sets, such as the 28 parameters in the two-way interaction model, exhaustive search is simply not feasible. A rough estimate of computation time for finding such a subset would require our computers to run continuously for multiple days straight, even when making use of parallel processing. For this reason, we include best subsets for the one-way interaction model but not the two-way interaction model. The code for best subset is slightly more complicated than Ridge or LASSO regression:

```
# -- one-way best subset --
maternal.z <- maternal.z %>%
  mutate(RiskLevel_low_risk = as.numeric(RiskLevel == "low risk"))

one.xy <- as.data.frame(cbind(one.x, one.y))

one.bs.glm <- bestglm(one.xy, IC = "AIC", TopModels = 1)

one.bs.glm$BestModel

# manually transform BestModel to lm
one.way.best <- lm(SystolicBP.z ~ Age.z + DiastolicBP.z + BodyTemp.z +
                     RiskLevel_low_risk, data=maternal.z)
```

This specific implementation of best subsets aims to minimize AIC. We can see how it accomplishes that visually by comparing the best subset for each n parameters versus the AIC. From there, we find the subset that has the lowest AIC and pick that as our best. The code to implement that is as follows:

```
# table 9
regsubsets.out <- regsubsets(SystolicBP.z ~ Age.z + DiastolicBP.z + BS.z +
                               BodyTemp.z + HeartRate.z + RiskLevellow_risk +
                               RiskLevelmid_risk, data = maternal.z, nbest = 1)

as.data.frame(summary(regsubsets.out)$outmat)

# chart 7
subset1 <- lm(SystolicBP.z ~ DiastolicBP.z, data = maternal.z)
subset2 <- lm(SystolicBP.z ~ DiastolicBP.z + RiskLevellow_risk,
              data = maternal.z)
subset3 <- lm(SystolicBP.z ~ DiastolicBP.z + RiskLevellow_risk +
                BodyTemp.z, data = maternal.z)
subset4 <- lm(SystolicBP.z ~ DiastolicBP.z + RiskLevellow_risk +
                BodyTemp.z + Age.z, data = maternal.z)
subset5 <- lm(SystolicBP.z ~ DiastolicBP.z + RiskLevellow_risk +
                BodyTemp.z + Age.z + BS.z, data = maternal.z)
subset6 <- lm(SystolicBP.z ~ DiastolicBP.z + RiskLevellow_risk +
                BodyTemp.z + Age.z + BS.z + HeartRate.z, data = maternal.z)
subset7 <- lm(SystolicBP.z ~ DiastolicBP.z + RiskLevellow_risk +
                BodyTemp.z + Age.z + BS.z + HeartRate.z + RiskLevelmid_risk,
              data = maternal.z)

# Calculate AIC for each subset model
aic_values <- c(AIC(subset1), AIC(subset2), AIC(subset3), AIC(subset4),
                AIC(subset5), AIC(subset6), AIC(subset7))

# Number of parameters in each model (including the intercept)
```

```r
num_parameters <- c(1,2,3,4,5,6,7)

# Create a data frame for the AIC and number of parameters
aic_data <- data.frame(
  Model = paste("Subset", 1:7),
  NumParameters = num_parameters,
  AIC = aic_values
)

# Find the subset with the lowest AIC
lowest_aic_subset <- which.min(aic_data$AIC)

# Define a size multiplier
size_multiplier <- ifelse(1:7 == lowest_aic_subset, 5, 1)
```

Figure 7: Best n parameters versus AIC.

Because the subset of four parameters produces the lowest AIC in Figure 7, it is the best subset. We call this model "One-Way Best", because no combination of one-way parameters is better than it at minimizing AIC.

We now have nine total models, which makes us far more confident that one of these models will perform very well at predicting systolic blood pressure. Next, we need to calculate the metrics

Parameter	1	2	3	4	5	6	7
Age				*	*	*	*
DiastolicBP	*	*	*	*	*	*	*
BS					*	*	*
BodyTemp			*	*	*	*	*
HeartRate						*	*
LowRisk		*	*	*	*	*	*
MidRisk							*

Table 10: Best n parameters to minimize AIC.

for each model. Non-Ridge and Non-LASSO models are easy, and we accomplish that with the following code. It works by first creating a list of the models, and then calculating each metric for each model:

```
# Storing models in a list
models <- list(
  one.way = one.way,
  one.way.backward = one.way.backward,
  one.way.best = one.way.best,
  two.way = two.way,
  two.way.backward = two.way.backward)

# Leave Out One Cross Validation
specs <- trainControl(method = "LOOCV")

# Creating the data frame
metrics <- data.frame(
  Model = names(models),
  R.sq = sapply(models, function(model) summary(model)$r.squared),
  R.adj.sq = sapply(models, function(model) summary(model)$adj.r.squared),
  LL = sapply(models, logLik),
  AIC = sapply(models, AIC),
  BIC = sapply(models, BIC),
  RMSE = sapply(models, function(model) sqrt(mean(residuals(model)^2))),
  MAE = sapply(models, function(model) mean(abs(residuals(model)))),
  Cross.R.sq = sapply(models, function(model) train(
    formula(model),
    data = maternal.z,
    method = "lm",
    trControl = specs,
    na.action = na.omit)$results$Rsquared),
  Cross.RMSE = sapply(models, function(model) train(
    formula(model),
    data = maternal.z,
    method = "lm",
    trControl = specs,
```

```
    na.action = na.omit)$results$RMSE),
  Cross.MAE = sapply(models, function(model) train(
    formula(model),
    data = maternal.z,
    method = "lm",
    trControl = specs,
    na.action = na.omit)$results$MAE),
  Parameters = sapply(models, function(model) length(coef(model))))
```

LASSO and Ridge models are harder to get the metrics for because the same functions do not work. These parameter selection techniques focus on optimizing coefficients and lambda values rather than selecting parameters to optimize a metric. Therefore, we need to calculate each metric manually. To identify which parameters are used in each model, we check for coefficients that are not equal to 0. As an example, we calculate the metrics for one-way Ridge regression with the following code:

```
owr.cv_fit <- cv.glmnet(one.x, one.y, alpha = 0)
owr.predicted_values <- predict(one.way.ridge,
                                s = owr.cv_fit$lambda.min, newx = one.x)

owr.R2 <- cor(one.y, owr.predicted_values)^2
owr.n <- length(one.y)  # Number of observations
owr.p <- ncol(one.x)  # Number of predictors
owr.R2_adj <- 1 - ((1 - owr.R2) * (owr.n - 1) / (owr.n - owr.p - 1))

owr.k <- sum(coef(one.way.ridge, s = owr.cv_fit$lambda.min) != 0)

owr.model_gaussian <- lm(one.y ~ one.x)
owr.ll <- logLik(owr.model_gaussian)
owr.AIC <- -2 * owr.ll + 2 * owr.k
owr.BIC <- -2 * owr.ll + log(owr.n) * owr.k

owr.RMSE <- RMSE(owr.predicted_values, one.y)
owr.MAE <- MAE(owr.predicted_values, one.y)

owr.predictions <- rep(NA, owr.n)
owr.actuals <- one.y

# Perform LOOCV
for (i in 1:owr.n) {
  train_index <- setdiff(1:owr.n, i)

  # Fit the model without the i-th observation
  owr.model_loocv <- glmnet(one.x[train_index, ], one.y[train_index],
                            alpha = 0)

  # Predict for the i-th observation
  owr.predictions[i] <- predict(owr.model_loocv, s = owr.cv_fit$lambda.min,
```

```
                                    newx = one.x[i, , drop = FALSE])
}

# Calculate R^2
owr.R2_loocv <- cor(owr.actuals, owr.predictions)^2

# Calculate RMSE
owr.RMSE_loocv <- sqrt(mean((owr.actuals - owr.predictions)^2))

# Calculate MAE
owr.MAE_loocv <- mean(abs(owr.actuals - owr.predictions))

# Store for later use
one.way.ridge.values <- list(
  Model = "one.way.ridge",
  R.sq = owr.R2,
  R.adj.sq = owr.R2_adj,
  LL = owr.ll,
  AIC = owr.AIC,
  BIC = owr.BIC,
  RMSE = owr.RMSE,
  MAE = owr.MAE,
  Cross.R.sq = owr.R2_loocv,
  Cross.RMSE = owr.RMSE_loocv,
  Cross.MAE = owr.MAE_loocv,
  Parameters = owr.k
)
```

The code required for the other LASSO and Ridge models is nearly identical, so it will be omitted. We combine the metrics values for each of these models with the other models as follows:

```
one.way.ridge.df <- as.data.frame(t(one.way.ridge.values))
one.way.lasso.df <- as.data.frame(t(one.way.lasso.values))
two.way.ridge.df <- as.data.frame(t(two.way.ridge.values))
two.way.lasso.df <- as.data.frame(t(two.way.lasso.values))

metrics <- rbind(metrics, one.way.ridge.df, one.way.lasso.df,
                 two.way.ridge.df, two.way.lasso.df)

for (variable in names(metrics)) {
  metrics[[variable]] <- unlist(metrics[[variable]])
}
```

Now, we can look at all of the metrics for all of the models:

```
print(xtable(metrics, digits = 4), include.rownames = FALSE)
```

Table 11 is clean, but 90 data points is a lot to evaluate and decide which model is the best. So, we

Model	R^2	R^2_{adj}	LL	AIC	BIC	RMSE
One-Way	0.6638	0.6614	-885.6941	1789.3882	1833.6831	0.5796
One-Way Backward	0.6630	0.6614	-886.8094	1787.6188	1822.0704	0.5802
One-Way Ridge	0.6629	0.6605	-885.6941	1787.3882	1826.7615	0.5820
One-Way LASSO	0.6637	0.6614	-885.6941	1785.3882	1819.8398	0.5796
One-Way Best	0.6629	0.6616	-886.9664	1785.9327	1815.4627	0.5803
Two-Way	0.7064	0.6983	-817.0209	1692.0418	1834.7699	0.5416
Two-Way Backward	0.7043	0.6992	-820.6264	1679.2528	1772.7643	0.5435
Two-Way Ridge	0.7012	0.6605	-817.0209	1690.0418	1827.8483	0.5478
Two-Way LASSO	0.7048	0.6967	-817.0209	1676.0418	1779.3966	0.5431

Model	MAE	CV R^2	CV RMSE	CV MAE	Parameters
One-Way	0.4621	0.6585	0.5841	0.4658	8
One-Way Backward	0.4627	0.6591	0.5836	0.4654	6
One-Way Ridge	0.4701	0.6578	0.5861	0.4735	8
One-Way LASSO	0.4627	0.6591	0.5835	0.4659	7
One-Way Best	0.4630	0.6597	0.5831	0.4652	5
Two-Way	0.4135	0.6883	0.5582	0.4254	28
Two-Way Backward	0.4160	0.6935	0.5534	0.4232	18
Two-Way Ridge	0.4272	0.6858	0.5611	0.4371	28
Two-Way LASSO	0.4181	0.6915	0.5552	0.4270	21

Table 11: Metrics for all models.

need some way of objectively determining which model is truly "best". It is very possible (and does occur in this case) that the best model according to one metric is not the best model according to other metrics. We are unable to just add or average all the metrics because they are of different scales, and for some metrics a higher value is better whereas others a lower value is better. To overcome this, we rank each model from one to nine in each metric against the others and average the ranks (one is best, nine is worst). The model with the lowest rank is then considered the "best". To account for some metrics preferring higher values (R^2, R^2_{adj}, Log Likelihood, LOOCV R^2_{adj}) and the rest preferring lower, we negate the metrics that prefer high values.

First, let's compare the models by scaling and negating the metrics. In this case for clarity, higher is better:

```
# Higher is better, so we negate
metrics_ordered <- metrics %>%
  mutate(
    AIC = -AIC,
    BIC = -BIC,
    RMSE = -RMSE,
    MAE = -MAE,
    Cross.RMSE = -Cross.RMSE,
    Cross.MAE = -Cross.MAE,
    Parameters = -Parameters
  )
```

```
metrics_long <- gather(metrics_ordered, Metric, Value, -Model)

metrics_long$Metric <-
  factor(metrics_long$Metric,
         levels = c("R.sq", "R.adj.sq", "LL", "AIC",
                    "BIC", "RMSE", "MAE", "Cross.R.sq",
                    "Cross.RMSE", "Cross.MAE", "Parameters"),
         labels = c("R^2", "R_adj^2", "LL", "AIC", "BIC",
                    "RMSE", "MAE", "CV R^2", "CV RMSE",
                    "CV MAE", "Parameters"))

# Adding the Top 3 column
metrics_long <- metrics_long %>%
  group_by(Metric) %>%
  mutate(Rank = rank(-Value)) %>%
  mutate(Top_3 = ifelse(Rank <= 3, 1, 0)) %>%
  ungroup()

# Filter out 'Parameters' metric
metrics_long <- metrics_long %>% filter(Metric != "Parameters")

# Define the colors for the top 3 models and "Other"
top_model_colors <- setNames(
  c(rep("grey", nrow(metrics_long) - sum(metrics_long$Top_3 == 1)),
    brewer.pal(sum(metrics_long$Top_3 == 1), "Set1")),
  c(rep("Other", nrow(metrics_long) - sum(metrics_long$Top_3 == 1)),
    unique(metrics_long$Model[metrics_long$Top_3 == 1])))
```

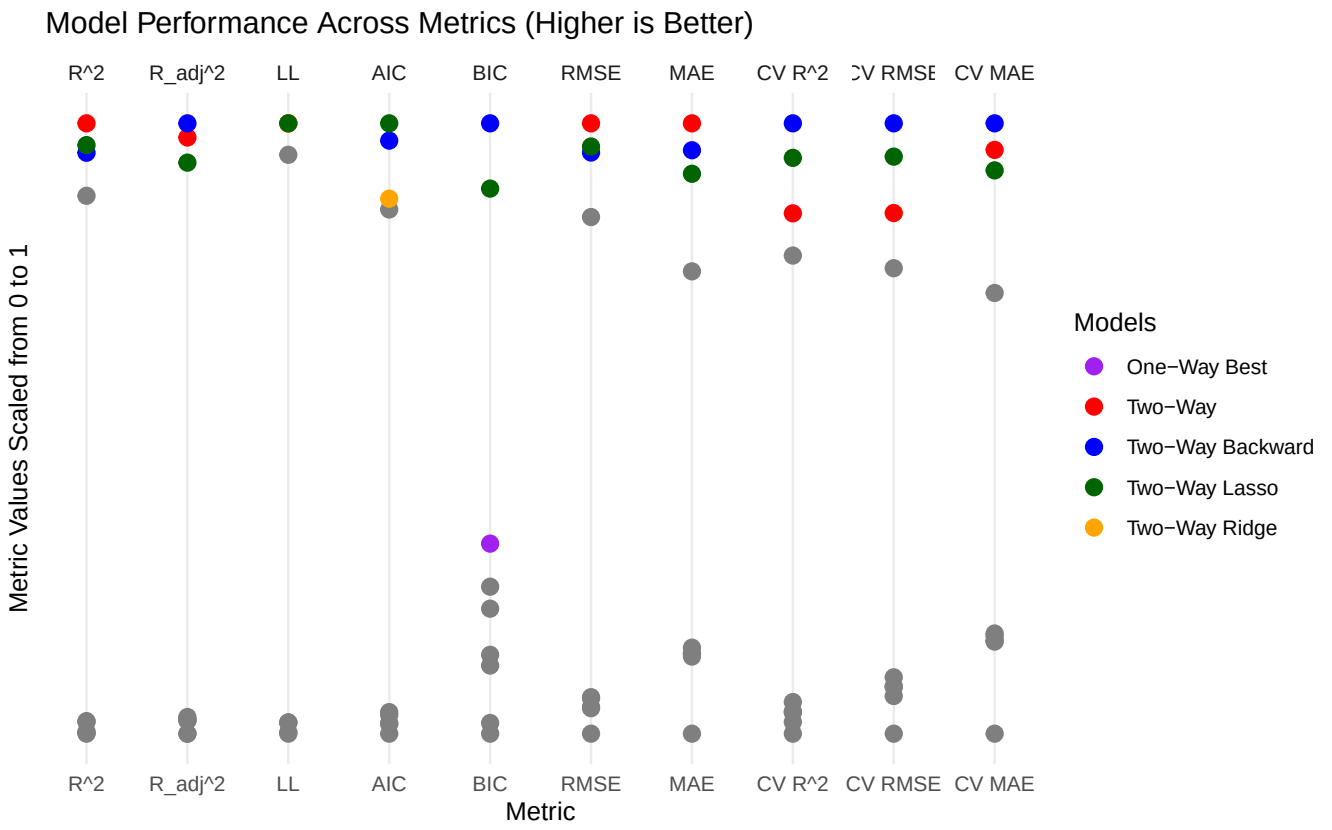

Figure 8: Comparison of models across all metrics.

This chart displays the top three models in each metric, scaled from 0 (worst) to 1 (best). We see that three models stand out as potential "best" models: two-way (red), two-way backward regression (blue), and two-way LASSO (green). Those three seem to trade back and forth across the metrics, each being the best in a few categories. One-way best subset (purple) and two-way ridge (yellow) also make appearances, but only once each and in third place both times. It is clear which models are our front runners.

Finally, we are ready to narrow down to a final model. We implement the code to rank the models on each metric below. The "best" model will be the one with the lowest average rank. If two models tie in a category, the average rank between the two is given. Metrics are negated again so that best values correspond to first place:

```
# Lower is better, so we negate
metrics_ranked <- metrics_ordered %>%
  mutate(
    Rank_R.sq = rank(-R.sq),
    Rank_R.adj.sq = rank(-R.adj.sq),
    Rank_LL = rank(-LL),
    Rank_AIC = rank(-AIC),
    Rank_BIC = rank(-BIC),
    Rank_RMSE = rank(-RMSE),
    Rank_MAE = rank(-MAE),
```

```
    Rank_Cross.R.sq = rank(-Cross.R.sq),
    Rank_Cross.RMSE = rank(-Cross.RMSE),
    Rank_Cross.MAE = rank(-Cross.MAE)
  )

metrics_ranked <- metrics_ranked %>%
  mutate(
    Rank_avg = rowMeans(cbind(Rank_R.sq, Rank_R.adj.sq, Rank_LL,
                              Rank_AIC, Rank_BIC, Rank_RMSE, Rank_MAE,
                              Rank_Cross.R.sq, Rank_Cross.RMSE,
                              Rank_Cross.MAE))
  )

# Specify the columns to be printed
columns_to_print <- c("Model", "Rank_R.sq", "Rank_R.adj.sq", "Rank_LL",
                      "Rank_AIC","Rank_BIC", "Rank_RMSE", "Rank_MAE",
                      "Rank_Cross.R.sq", "Rank_Cross.RMSE", "Rank_Cross.MAE",
                      "Rank_avg")
```

Which results in this table:

Model	R^2	R^2_{adj}	LL	AIC	BIC	RMSE	MAE	CV R^2	CV RMSE	CV MAE	Avg Rank
One-Way	5	5	6	9	8	5	5	8	8	7	6.6
One-Way Backward	7	7	8	8	5	7	6	7	7	6	6.8
One-Way Ridge	9	8.5	6	7	6	9	9	9	9	9	8.2
One-Way LASSO	6	6	6	5	4	6	7	6	6	8	6.0
One-Way Best	8	4	9	6	3	8	8	5	5	5	6.1
Two-Way	1	2	2	4	9	1	1	3	3	2	2.8
Two-Way Backward	3	1	4	2	1	3	2	1	1	1	1.9
Two-Way Ridge	4	8.5	2	3	7	4	4	4	4	4	4.5
Two-Way LASSO	2	3	2	1	2	2	3	2	2	3	2.2

Table 12: Models ranked from 1 (best) to 9 (worst).

Based on these findings, the two-way backward regression model is the best, with an average rank of 1.9. It was first place across five metrics, second in two, third place in two, and fourth place in one. Receiving third or fourth place is expected for these three metrics (R^2, Log Likelihood, and RMSE) because they do not account for the number of parameters. The winner in those categories (the two-way model - average rank of 2.8) has 10 additional parameters than the two-way backward regression model. This fact benefits the two-way model greatly because R^2, Log Likelihood, and RMSE do not account for number of parameters. But, as indicated by the two-way model's ranks in the other six metrics, they are worse when number of parameters is accounted for. This process gives us the confidence that the two-way interaction backward selection model is the best, and is thus our final model.

A notable runner-up, however, is the two-way LASSO regression model. It performed similarly well to the two-way backward selection model with an average rank of 2.2. It was only first place in one category (AIC), but had six second place scores and three third place scores. This is commendable, but we chose the two-way backward regression model as our best for a few reasons. First, the

two-way backward model had the lowest average rank. Average rank is the objective criteria we established earlier, and we are sticking with it now. If it says two-way backward regression is the best, it is the best. Secondly, the two-way backward regression model had the best scores across all cross-validation metrics. This is really important, because it means in a real world test, it would likely perform the best. Thirdly, LASSO is a complicated and difficult to interpret model. Not all of the metrics we use work especially well with LASSO, so some of them may appear better than reality. This leads us to conclude that LASSO is not a good enough option to upset the first place model, two-way backward regression.

4.5 Final Model

Our final model is the two-way interaction backward selection model (hereafter known as the best model). Let's get a better understanding of it. First, a look at the residuals:

```
source("https://cipolli.com/students/code/plotResiduals.R")
plotResiduals(two.way.backward)
```

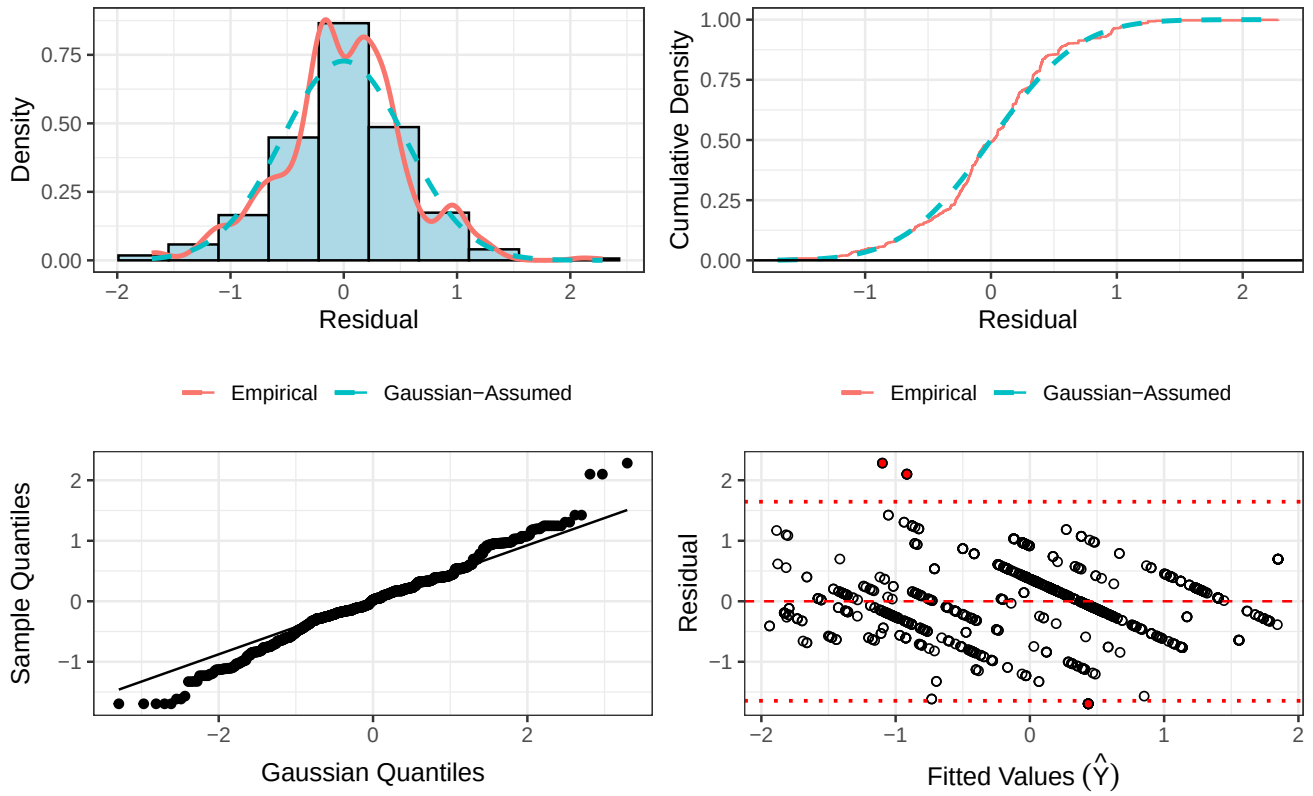

Figure 9: Residuals to assess assumptions of normality and constant variance.

The residuals appear to be approximately normal. The residual density is bell-curved and follows closely to the Gaussian-assumed cumulative density function. The q-q plot is okay, but the tails are not the best. Specifically, the tails indicate a rightward skew of residuals due to the overabundance at higher quantiles and lack at lower quantiles. The fitted residuals look good, save for three data points out of 1206.

Next, conducting an ANOVA test of the best can reveal the most significant interactions. We can do that with the following lines:

```
anova(two.way.backward)
```

Parameter	Df	Sum Sq	Mean Sq	F value	p-value
Age.z	1	175.34	175.34	582.96	< 0.0001
DiastolicBP.z	1	464.81	464.81	1545.33	< 0.0001
BS.z	1	4.22	4.22	14.04	0.0002
BodyTemp.z	1	5.50	5.50	18.29	< 0.0001
HeartRate.z	1	0.00	0.00	0.01	0.9424
RiskLevel	2	22.52	11.26	37.44	< 0.0001
Age.z:DiastolicBP.z	1	3.08	3.08	10.24	0.0014
Age.z:HeartRate.z	1	2.48	2.48	8.24	0.0042
Age.z:RiskLevel	2	1.71	0.85	2.84	0.0590
DiastolicBP.z:BS.z	1	12.43	12.43	41.34	< 0.0001
DiastolicBP.z:BodyTemp.z	1	6.36	6.36	21.15	< 0.0001
DiastolicBP.z:RiskLevel	2	11.81	5.91	19.64	< 0.0001
BodyTemp.z:RiskLevel	2	3.15	1.58	5.24	0.0055
Residuals	996	299.58	0.30		

Table 13: ANOVA table for regression of two-way interaction parameters with backward regression.

From the ANOVA in Table 13, we see that there are 7 statistically significant parameters out of 13. Similar to the two-way ANOVA, Age and DiastolicBP are the most statistically significant. Age has a F value of 582.96 and p-value of < 0.0001, while DiastolicBP has a F value of 1454.33 and p-value of < 0.0001. The least stastically significant is again HeartRate, with an F value of 0.01 and p-value of 0.9424. The most significant interaction is that between DiastolicBP and Blood Sugar (F value $= 41.34$, p-value < 0.0001), which is not surprising because that was also true for the two-way ANOVA.

Finally, a look at the summary statistics of the best model:

Parameter	Estimate	Std. Error	t-value	p-value
(Intercept)	0.2333	0.0600	3.8896	< 0.001
Age.z	0.0468	0.0477	0.9799	0.3274
DiastolicBP.z	0.6763	0.0512	13.2055	< 0.001
BS.z	-0.0259	0.0350	-0.7385	0.4604
BodyTemp.z	-0.1335	0.0455	-2.9326	0.0034
HeartRate.z	-0.0383	0.0185	-2.0737	0.0384
RiskLevellow risk	-0.4004	0.0736	-5.4413	< 0.001
RiskLevelmid risk	-0.1206	0.0718	-1.6805	0.0932
Age.z:DiastolicBP.z	-0.1175	0.0244	-4.8127	< 0.001
Age.z:HeartRate.z	0.0434	0.0168	2.5799	0.0100
Age.z:RiskLevellow risk	-0.0260	0.0572	-0.4543	0.6497
Age.z:RiskLevelmid risk	0.0898	0.0577	1.5550	0.1203
DiastolicBP.z:BS.z	0.1340	0.0281	4.7640	< 0.001
DiastolicBP.z:BodyTemp.z	0.1296	0.0228	5.6951	< 0.001
DiastolicBP.z:RiskLevellow risk	0.0967	0.0686	1.4096	0.1590
DiastolicBP.z:RiskLevelmid risk	-0.2158	0.0702	-3.0729	0.0022
BodyTemp.z:RiskLevellow risk	0.1772	0.0587	3.0205	0.0026
BodyTemp.z:RiskLevelmid risk	0.0705	0.0569	1.2389	0.2157

Table 14: Summary Statistics of Two-Way Backward Regression Model.

Compared to the full two-way interaction model, the backward selection model has 3 more statistically significant parameters (including intercept) for a total of 11. The most important of these factors, by several orders of magnitude, is diastolic blood pressure (t-value = 13.215; p-value < 0.0001). Specifically, the model indicates that there is a 1 in 5 quadrillion chance for the relation between diastolic and systolic blood pressures seen in the data to be due to random chance.

Importantly, this best model has a very high LOOCV R^2 (0.6935), low AIC (1679.25), and low LOOCV RMSE (0.5534). These metrics indicate that the model is very accurate at explaining variance in systolic blood pressure. The cross-validation reinforces this belief, as it is often a better reflection of real-life performance.

4.6 Conclusions

Our approach of ranking models offers a robust and objective way to determine the best model. In total, we looked at fourteen different models and measured their performances with a diverse group of ten metrics. We did this to have a variety of models to chose from and compare, and with ten measurements of performance we could get a holistic understanding of each. From this process of model creation, ranking, and determining a best model, we have found that Diastolic Blood Pressure strongest positive predictor of Systolic Blood Pressure (t-value = 13.2154, p-value < 0.0001). At the same time, we have found that a low risk level is the strongest negative predictor of Systolic Blood Pressure (t-value = -5.4471, p-value < 0.0001). The most significant interaction happened to be between Diastolic Blood Pressure and body temperature (t-value = 5.6951, p-value < 0.0001), while the least significant interaction (while still being in the best model) is between age and low risk level (t-value = -0.4543, p-value = 0.6497). The least significant parameters overall were blood sugar (p-value 0.4504) and age (p-value 0.3320).

Lastly, we want a final way to compare all of the models we evaluated in a separate way from lowest average rank. This time, we compare each model's performance to the one-way model's performance. The way we do this is by calculating the percentage improvement in each metric of a model over the one-way model's performance. Then, we average these percentage improvements and plot them for all nine models. What this shows is relative performance versus are original one-way model. This calculations provides an indication of how much better (or worse) are models got compared to the most simple case: one-way all parameters. We implement the code for this as follows:

```
base_values <- as.list(metrics_ordered[1, ])

metrics_improvement <- metrics_ordered %>%
  mutate(
    Imprv_R.sq = R.sq / base_values$R.sq,
    Imprv_R.adj.sq = R.adj.sq / base_values$R.adj.sq,
    Imprv_LL = LL / base_values$LL,
    Imprv_AIC = base_values$AIC / AIC,
    Imprv_BIC = base_values$BIC / BIC,
    Imprv_RMSE = base_values$RMSE / RMSE,
    Imprv_MAE = base_values$MAE /MAE,
    Imprv_Cross.R.sq = Cross.R.sq / base_values$Cross.R.sq,
    Imprv_Cross.RMSE = base_values$Cross.RMSE / Cross.RMSE,
    Imprv_Cross.MAE = base_values$Cross.MAE / Cross.MAE
  )

# Manually specify the improvement columns
improvement_columns <- c("Imprv_R.sq", "Imprv_R.adj.sq", "Imprv_LL",
                         "Imprv_AIC", "Imprv_BIC", "Imprv_RMSE",
                         "Imprv_MAE", "Imprv_Cross.R.sq",
                         "Imprv_Cross.RMSE", "Imprv_Cross.MAE")

# Adding a new column to the metrics_improvement dataframe
# This column will contain the average of
# the specified improvement values for each model
metrics_improvement$average_improvement <- 100 *
  (rowMeans(metrics_improvement[improvement_columns], na.rm = TRUE) - 1)

# Reorder the models based on average improvement
metrics_improvement <- metrics_improvement %>%
  arrange(average_improvement) %>%
  mutate(Model = factor(Model, levels = Model))
```

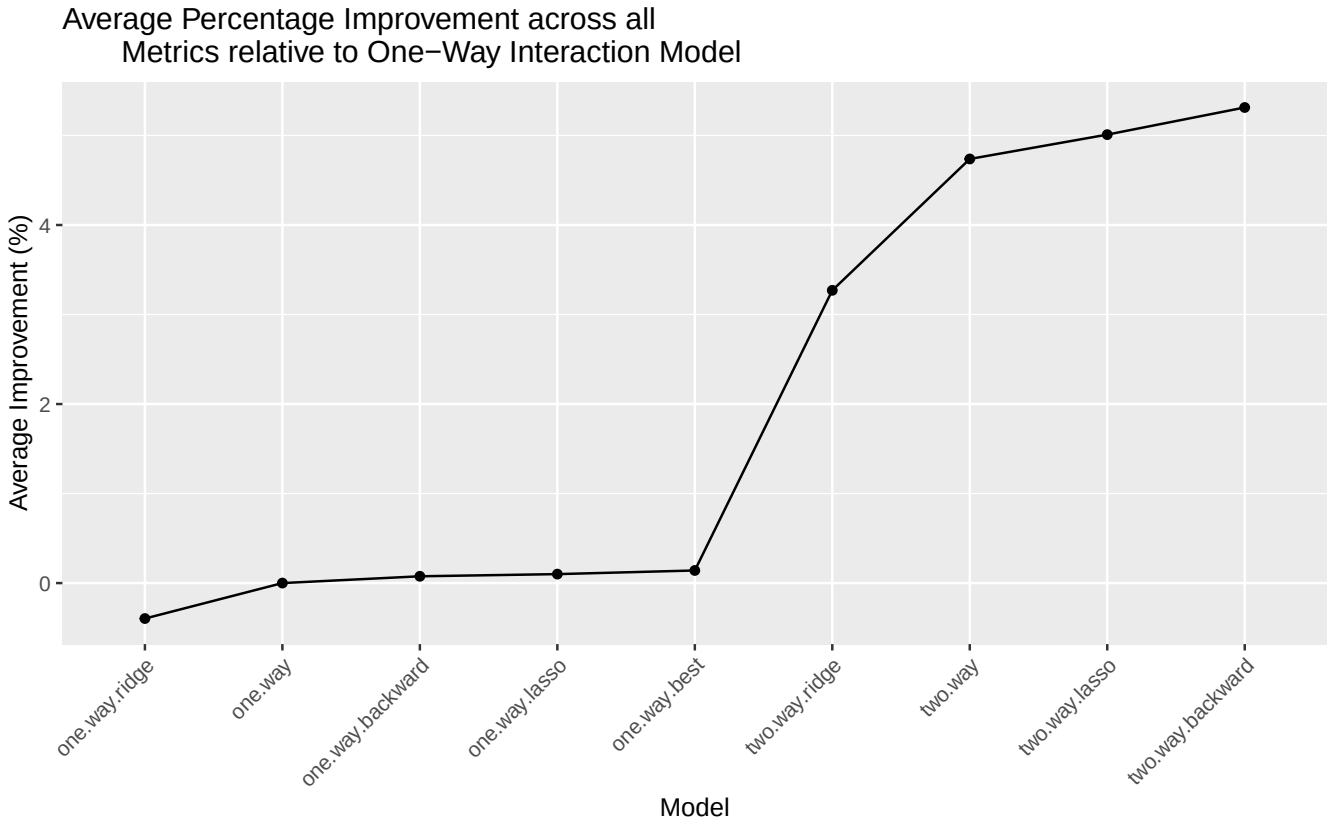

Figure 10: Line graph demonstrating average line graph improvement.

In Figure 10, we display the average relative performance of each model, across all metrics, versus the one-way model. We notice several interesting things from this. First, only one model performed worse, on average, than the one-way model. That was the one-way Ridge regression, which is a bit surprising and unfortunate, considering its complexity. Then, there are a group of one-way regression models just slightly better than the one-way all parameters model. As expected, one-way best subsets is the best of all first-order models. Then, there is a sizable average improvement of about 3.2% to two-way Ridge regression. Again, ridge regression performs the worst of the two-way interaction models. Next, we have the two-way all parameters model, followed by two-way LASSO regression, and finally two-way backward regression, with an average improvement of 5.3% versus the one-way model. This is a sizable improvement, and corroborates the findings from are ranking system which had two-way backward regression as the best.

4.7 Next Steps

As mentioned earlier, best subsets regression for 28 parameters is not feasible given the current computation power of our group. With a very powerful computer, or lots of computation time, it is feasible for us to evaluate all 134 million linear models to find the best, but given monetary and time constraints, it is not something we could complete during this project. However, if we were to continue the project and seek the truly best linear model, we could implement code similar to the one-way best subsets model. We could do that as follows:

```r
# -- two-way best subset --
maternal.z <- maternal.z %>%
  mutate(RiskLevel_low_risk = as.numeric(RiskLevel == "low risk"))

two.x <- model.matrix(two.way)[, -1]
two.y <- maternal.z$SystolicBP.z

two.xy <- as.data.frame(cbind(two.x, two.y))

two.bs.glm <- bestglm(two.xy, IC = "AIC", TopModels = 1)

two.bs.glm$BestModel
```

This method would successfully find the best subset, however there would be some challenges. First and foremost, this would take a long time to compute. Even if we could check 1000 models a second, that would still take 37 continuous hours of compute time. Even if we were willing to have our computers run for that long, we might run into memory issues long before then. 134 million models is a lot of data, and even with 32 or 64 gigabytes or RAM, that likely would not be enough to store all of them. There are potentially some ways around both of these issues, such as parallel processing and storing model data in csv files, but the process gets complicated fast.

Unfortunately, pursuing this area further seems out of reach at the current moment. However, it is interesting to think about what the best subset of two-way interactions would be, and how it would compare against our current best model, the two-way backward regression model. Is there a large improvement over it? A small one? Only by checking all 134 million models will we know.

5 Conclusion

The two-way backward model was the best overall model which we deduced through a ranking system that measured R^2, R^2_{adj}, LL, AIC, BIC, RMSE, MAE, CV R^2, CV RMSE, and CV MAE. The overall average rank was 1.4 which was calculated by averaging the model's overall scores in each category listed above. The $R^2 = 2.0$, $R^2_{adj} = 1.0$, LL = 2.0, AIC = 1.0, BIC = 1.0, RMSE = 2.0, MAE = 2.0, CV $R^2 = 1.0$, CV RMSE = 1.0, and CV MAE = 1.0. The top two most significant predictors for systolic blood pressure include diastolic blood pressure, with a t-value of 13.2154, and low risk level, with a t-value of 5.6983. Meaning, these two predictive variables are most effective when predicting different systolic blood pressure levels which could help prevent severe hypertension risk. Additionally, the most significant interaction was between diastolic blood pressure and body temperature, with a t-value of 5.6983. This means that as diastolic blood pressure impacts systolic blood pressure, there are also variances based on the body temperature of the patient. Doctors may want to see if there are ways to control diastolic blood pressure and body temperature in pregnant women in Bangladesh to reduce their chance of high systolic blood pressure, and in turn, hypertension.

On the other hand, the least indicative variables were the interaction between age and low risk level (p-value = 0.6486), blood sugar level (p-value = 0.4504), and age (p-value = 0.3320). Our model highlighted results that we had expected after our exploratory analysis with ANOVA. Both blood sugar level and heart rate were expected to be weak predictors and was consistently so in the

two-way backwards model. Blood sugar only appears as itself and in one interaction with diastolic blood pressure. Similarly, heart rate only appears as itself and in an interaction with age. Overall, our final model highlighted key predictive variables that could be researched further so that the medical field can take more preventative action against hypertension. Additionally, our data can only be applied to pregnant women in rural Bangladesh, but provides important insight for studies that could be done on larger samples across the globe. With increased emphasis on what variables impact systolic blood pressure, doctors can begin to control the risk their patients are at suffering from hypertension and prevent maternal deaths worldwide.

After completing our project there were a few things that we would have done differently. Firstly, while we were highly interested in our data set topic, in the future, we would like to have had access to a more robust data set. Specifically, it would be interesting to explore additional parameters such as a patient's weight, number of times they had given birth, and race. According to the American Heart Association, there weight is a significant factor in a patient's risk for hypertension, so finding a data set that included that parameter would be ideal for future research. Additionally, a data set with more observations spanning across other countries and locations would help future research be applicable to a larger population. Lastly, for our project we selected our parameters based on AIC, but if we were to go through the project again, we would take time to select those parameters based on an average rank.

References

Abera, T. and Mekonnen, T. (2019). Pregnancy induced hypertension and associated factors among women attending delivery service at mizan-tepi university teaching hospital, tepi general hospital and gebretsadik shawo hospital, southwest, ethiopia. *Ethiopian Journal of Health Sciences*, 29:831–840.

Ahmed, M. (2023). Maternal Health Risk.

Cannon, A., Cobb, G., Hartlaub, B., Legler, J., Lock, R., Moore, T., Rossman, A., and Witmer, J. (2019). *Stat2Data: Datasets for Stat2*. R package version 2.0.0.

Dahl, D. B., Scott, D., Roosen, C., Magnusson, A., and Swinton, J. (2019). *xtable: Export Tables to LaTeX or HTML*. R package version 1.8-4.

E, F. and Jr, H. (2023). *rms: Regression Modeling Strategies*. R package version 6.7-1.

Fox, J. and Weisberg, S. (2019). *An R Companion to Applied Regression*. Sage, Thousand Oaks CA, third edition.

HERVE, M. (2023). *RVAideMemoire: Testing and Plotting Procedures for Biostatistics*. R package version 0.9-83-2.

Kuhn and Max (2008). Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(5):1–26.

McLeod, A., Xu, C., and Lai, Y. (2020). *bestglm: Best Subset GLM and Regression Utilities*. R package version 0.37.3.

Neuwirth, E. (2022). *RColorBrewer: ColorBrewer Palettes*. R package version 1.1-3.

News, A. H. A. (2021). High blood pressure treatment in pregnancy is safe and could reduce mother's risks.

Ogle, D. H., Doll, J. C., Wheeler, A. P., and Dinno, A. (2023). *FSA: Simple Fisheries Stock Assessment Methods*. R package version 0.9.5.

Pedersen, T. L. (2023). *patchwork: The Composer of Plots*. R package version 1.1.3.

Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., Elberg, A., and Crowley, J. (2021). *GGally: Extension to 'ggplot2'*. R package version 2.1.2.

Team, R. C. (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.