Project 2

Tabib Chowdhury

2023-11-14

Abstract

In this collaborative project with Dr. Chris Schmid, we attempt to address the challenges of determining indication criteria and timing for tracheostomy placement in neonates with severe bronchopulmonary dysplasia (sBPD). Leveraging a national dataset comprising demographic, diagnostic, and respiratory parameters of infants from collaborative NICUs, we conducted an extensive analysis that involved Exploratory Data Analysis (EDA), missing data analysis, leading to the implementation of multiple imputation techniques. Imputation was performed for both the complete dataset and a targeted subset at 44 weeks, because we wanted to address the non-proportionate missing data between 36 and 44 weeks.

To develop a robust predictive model for the composite outcome of tracheostomy/death, various model selection approaches were explored. We performed Lasso regularization based on Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), and backward stepwise selection based on AIC. We consider 4 different logistic mixed effects models and their performance was rigorously compared using prediction, discrimination, and calibration metrics. Additionally, to find the optimal timeframe for tracheostomy, predictions were stratified across diverse subsets of the data.

Introduction

Although many studies suggests that early tracheostomy placement for babies with severe bronchopulmonary dysplasia (sBPD) has benefits, the work around the criteria and timing of the precedure is still incomplete. In this research project, we will attempt to address this issue by developing a regression model to predict the composite outcome of tracheostomy/death to guide the indication criteria and timing of tracheostomy placement. In our model we will include birth variables, respiratory support variables, and infant data at 36 and 44 weeks corrected gestational age(CGA). We estimate an ideal time frame to refer a patient for tracheostomy by providing predictions across different postmenstrual ages. This report is split up into 7 main sections: Introduction, Methods, Exploratory Data Analysis, Regression Analysis, Results, Discussion and Conclusion.

Methods

We first perform multiple data imputation(m = 5) using the MICE package in R and split the data into training and testing sets with a 70-30 split. From our exploratory data analysis we know that the composite outcome of tracheostomy and death of the neonatal infants are not evenly distributed throughout the centers. This is due to the fact that the patient's severity with sBPD is correlated with which center they are placed in. Another point to consider with the centers is that once we fit a model, it will be difficult to get accurate predictions from said models since we cannot extrapolate outside the centers given in the data. Because of this we were either left with the option of leaving out the center variable and fit a logistic generalized linear

model or fit a generalized mixed effects model with the center variable as a random intercept. We chose the latter as we believe that the random intercept will capture the variability between centers for each patient based on their differing levels of severity. In our fixed effect model we consider the composite outcome of tracheostomy and death as the binary outcome variable and the patients birth variables, respiratory support variables, and infant data at 36 and 44 weeks corrected gestational age(CGA) as predictors.

Exploratory Data Analysis

Like we mentioned before, based on previous work done on the project we know that the data has been collected from multiple medical centers and the patients who go to the centers have differing levels of severity when it comes to bronchopulmonary dysplasia. To visualize this we can look at the table below that displays the proporiton of the composite death/trach outcomes by centers. Note that this table does not show all cases as there is one occurance of center 21. We remove this as it can be biased due to the low sample size. We also remove cases when both death and trach are missing since we cannot predict the outcome if both are missing.

```
##
##   1   2   3   4   5   7  12  16  20  21
##  55 633  57  60  40  32  69  38   4   1
```

```
## Warning in df$record_id == df$record_id[duplicated(df$record_id)]: longer
## object length is not a multiple of shorter object length
```

center	Proportion	N
1	0.4181818	55
2	0.1019108	628
3	0.0175439	57
4	0.1864407	59
5	0.1250000	40
7	0.0312500	32
12	0.5072464	69
16	0.0263158	38
20	0.0000000	4

From the table above we can see that the majority of death or tracheostomy occur in centers 1 and 12, even if the sample size of the center 1 and 12 are relatively low compared to the others. The summary table shows huge discrepencies between death/trach outcomes between centers For example center 12 has a much higher chance of taking in more serious patients.

From the summary table stratified by the death and tracheostomy we find that the variables birthweight, delivery method, prenatal corticosteroids, small size for gestational age, weight at 36 weeks, ventilation support level at 36 and 44 weeks, Fraction of Inspired Oxygen needed at 36 and 44 weeks, Peak Inspiratory Pressure (cmH2O) needed at 36 and 44 weeks, Positive end exploratory pressure (cm H2O) needed at 44 weeks and Medication for Pulmonary Hypertension at 36 and 44 weeks.

The summary table also shows that the number of missing values for the variables recorded at 44 weeks is significantly higher than the ones at 36 weeks. We will explore this more in the next section.

Missing data analysis:

From the table of missing value proportions we can see that approximately 45% of the missing values recorded in week 44 is missing while only about 10% of the recorded values at 36 weeks is missing. Because of this I decided to fit the mixed effects model on two different datasets:

Table 1: **Data Summary stratified by Death/Trach Composite outcome**

Variable	N	Overall, N = 992	Death/Trach Composite outcome		p-value
			0, N = 847	**1**, N = 145	
mat_ethn	935				0.22
1		73 / 935 (7.8%)	66 / 800 (8.3%)	7 / 135 (5.2%)	
2		862 / 935 (92%)	734 / 800 (92%)	128 / 135 (95%)	
Unknown		57	47	10	
bw	992	806 (297)	814 (295)	762 (303)	0.005
ga	992	26 (2)	26 (2)	26 (2)	0.68
blength	914	32 (4)	33 (4)	32 (4)	0.18
Unknown		78	48	30	
birth_hc	915	23.19 (2.76)	23.22 (2.72)	23.00 (3.07)	0.14
Unknown		77	46	31	
del_method	989				0.039
1		284 / 989 (29%)	253 / 845 (30%)	31 / 144 (22%)	
2		705 / 989 (71%)	592 / 845 (70%)	113 / 144 (78%)	
Unknown		3	2	1	
prenat_ster	957	831 / 957 (87%)	709 / 827 (86%)	122 / 130 (94%)	0.011
Unknown		35	20	15	
com_prenat_ster	799	607 / 799 (76%)	521 / 687 (76%)	86 / 112 (77%)	0.83
Unknown		193	160	33	
mat_chorio	930	160 / 930 (17%)	138 / 797 (17%)	22 / 133 (17%)	0.83
Unknown		62	50	12	
gender	988				0.91
Female		406 / 988 (41%)	347 / 843 (41%)	59 / 145 (41%)	
Male		582 / 988	496 / 843	86 / 145 (59%)	

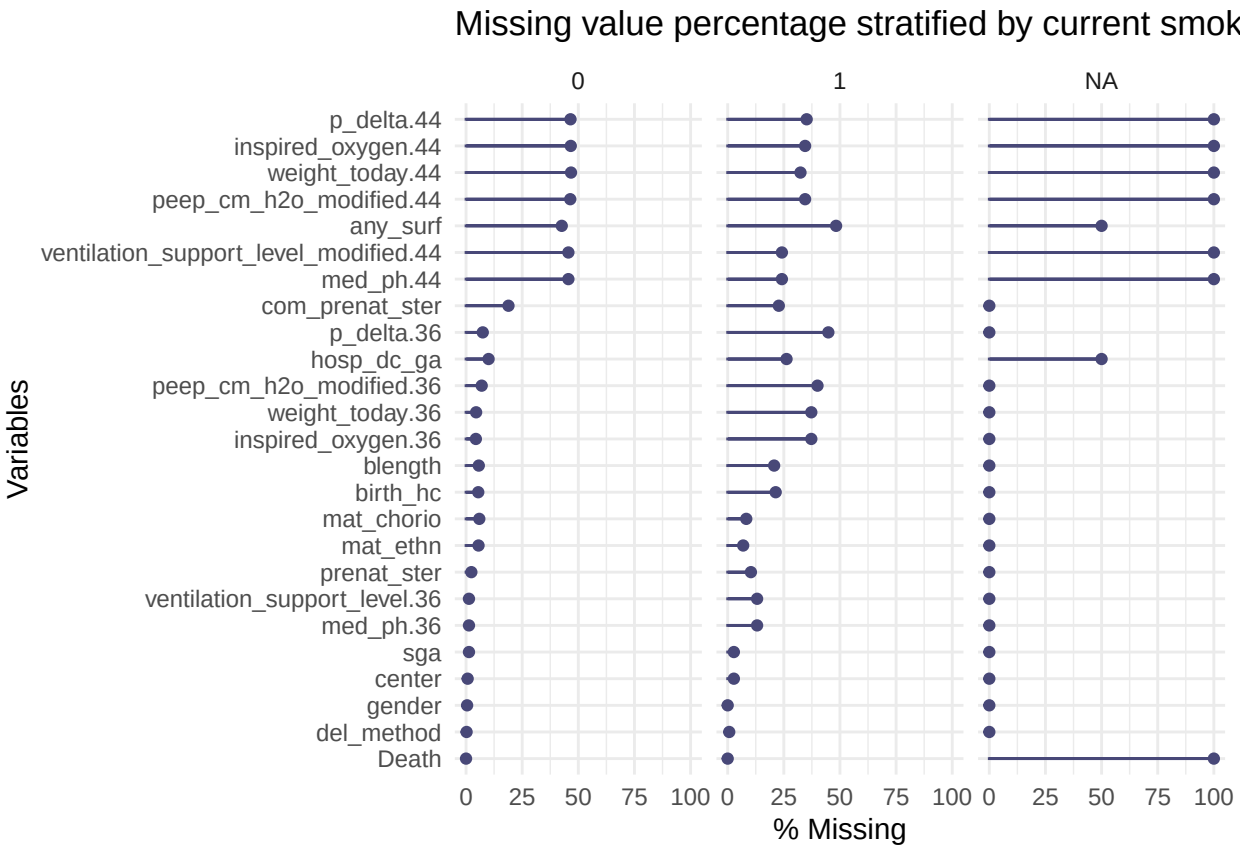
This points to some pattern and structure in the missing data. Another way to visualize the missing data is to stratify the proportion of missing data by the composite outcomes.

1) The first data set contains data on both 36 and 44 week data and the multiple imputation, train/test split, and analysis was performed on both 36 and 44 week data.
2) The second data only includes complete(~55%) 44 week data. The multiple imputation, train/test split, and analysis was performed on only 44 week data.

This allows us to to not only compare the effects of recent recorded history of the patient vs all history, but also allows us to see if including a large amount of missing values from the 44 week data in the multiple imputation model will lead to bias and inaccuracies in our prediction.

Table 2: Missing Data Proportion for Each Variable

Variable	Observation Missing	Proportion Missing
inspired_oxygen.44	447	44.9698189
p_delta.44	447	44.9698189
weight_today.44	445	44.7686117
peep_cm_h2o_modified.44	445	44.7686117
any_surf	432	43.4607646
ventilation_support_level_modified.44	423	42.5553320
med_ph.44	423	42.5553320
com_prenat_ster	193	19.4164990
p_delta.36	128	12.8772636
hosp_dc_ga	124	12.4748491
peep_cm_h2o_modified.36	117	11.7706237
weight_today.36	92	9.2555332
inspired_oxygen.36	91	9.1549296
blength	78	7.8470825
birth_hc	77	7.7464789
mat_chorio	62	6.2374245
mat_ethn	57	5.7344064
prenat_ster	35	3.5211268
ventilation_support_level.36	30	3.0181087
med_ph.36	30	3.0181087
sga	15	1.5090543
center	10	1.0060362
gender	4	0.4024145
del_method	3	0.3018109
Death	2	0.2012072
Composite_Outcome	2	0.2012072



From the table and plot above we see that the data is missing unproportinately between 36 and 44 weeks. Additionally there is some pattern to the missing data when we stratify by the composite outcome. Because of this we can say that the pattern is Missing not at random. This allows us to go forward with multiple imputation.

Regression analysis: Mixed effects model

Because center is has high variance in determining the outcome, we decided to fit a mixed effects model using center as a random intercept.

We will fit 2 main models. One will include data from both 36 and 44 weeks. The second will only include 44 week data. This will allow us to see if including the 36 week data to our model is useful. We will not include race, id and discharge gestational age in any of the models as they are not logically useful in determining the outcome.

We will perform 2 main model selection with lasso and backward model selection. In total we will consider 4 main models:

1) Lasso selected full model with 36 amnd 44 week data.
2) Lasso selected 44 week data model.
3) Backward selected full model with 36 and 44 week data.
4) Backward selected 44 week data model.

Since our outcome is binary, we will fit a mixed effects model with the binomial(logit) family or a logistic mixed effects model. The generalized structure of a logistic mixed effects model is as follows:

$$logit(p_{ij}) = \beta_0 + \beta_1 X_{1ij} + \beta_2 X_{2ij} + \ldots + \beta_p X_{pij} + u_{0j} + \epsilon_{ij}$$

where p_{ij} is the probability of success(a patient passing away or getting tracheostomy) for the ith observation in the jth center, $logit(p_{ij}) = log(\frac{p_{ij}}{1-p_{ij}})$ is the log odds of and β_0 is the fixed intercept. & $\beta_1, \beta_2, \ldots, \beta_p$ are the fixed effects associated with the predictors $X_{1ij}, X_{2ij}, \ldots, X_{pij}$ respectively. & u_{0j} is the random intercept for the jth group, assumed to be $N(0, \sigma_u^2)$. & ϵ_{ij} is the residual error term.

Additionally the following is assumed for any mixed effects model:

1) Linearity: The relationship between the fixed effect predictors $X_1, X_2, ..., X_p$ and the logodds of a success is linear.
2) Independence of residuals: The residuals ϵ_{ij} are independent of each other.
3) Normality of residuals: The residuals ϵ_{ij} are normally distributed.
4) Homoscedasticity: The variance of the residuals is assumed to be constant across all levels of the predictors.
5) Distribution of the random intercept: The random intercepts u_{0j} are normally distributed around mean 0 and variance σ_j^2, i.e $u_{0j} \sim \mathcal{N}(0, \sigma_j^2)$

Note: We don't consider interaction terms as this model is meant to be used in a clinical setting. So to keep interpretations simple we don't consider any interaction terms. During the EDA process I found multiple outliers that in birth variables which I removed before my analysis.

From our multiple imputation model, we have 5 sets of training data along with 5 sets of test data for the full dataset and 5 sets of training data along with 5 sets of test data for the 44 week data. As a reminder we will fit and test 4 different models on all the training data and validate our models using the test set.

After doing imputation on the data I decieded on doing lasso and backward model selection because I wanted to perform both shrinkage regularization and a strict model selection technique and compare between the two. I decided to perform lasso and backward stepwise model selection.

The set of models were selected using the lasso shrinkage technique on the the both the 44 and 36 week data and also only the 44 week data. Lasso is a regularization technique that prevents overfitting of the model. It is a hybrid between a strict stewise model selection and a regularization model like ridge regression. It performs both model selection and also shrinkage. The full data and the 44 week data was fit to the lasso model and coefficients shrinkage and deletion of the model coefficients were completed. To find the optimal lambda parameter, I fit the models to many different lasso functions with differing lambda's. Usually lambda is chosen based on either the AIC or BIC criterions. I chose the lambda by combining these 2 criterions: I found the optimal λ_{AIC} based on minimizing the AIC and another optimal lambda λ_{BIC} to minimize the BIC. After doing so I took the average of the two and chose that parameter as the final shrinkage parameter. We know that AIC penalizes less for more complex models, meaning it prefers more complex models compared to BIC. Taking the average between λ_{BIC} and λ_{AIC} allows us to choose a sweet spot between the two criterions. After finding the optimal lambda parameter, I found the coefficients fitting all 5 training sets and averaged them out. Like the lasso regression process, I did backward stepwise which selects the final model based on the AIC on all 5 training sets and averaged out the coefficients.

Fixed Effects table:

Coefficient	Full Backward	44 week Backstep	44 week Lasso	Full Lasso
(Intercept)	-8.7077569	-13.4321743	-3.6618639	-4.8989563
birth_hc	0.1068152	0.2602966	0.0000000	0.0000000
com_prenat_sterYes	0.6247767	0.5128735	0.0000000	0.0000000
blength	0.0397892	0.1466820	0.0000000	0.0000000
prenat_sterYes	0.6217963	1.1175496	0.0000000	0.0000000
ventilation_support_level.36	0.4287413	0.0000000	0.0000000	0.5073282
inspired_oxygen.36	4.2715417	0.0000000	0.0000000	4.4033305
peep_cm_h2o_modified.36	0.0225185	0.0000000	0.0000000	0.0117749
weight_today.36	-0.0005388	0.0000000	0.0000000	0.0000000
p_delta.36	0.0265671	0.0000000	0.0000000	0.0124459
med_ph.36	-0.5085609	0.0000000	0.0000000	-0.4893202
weight_today.44	0.0001183	-0.0002035	0.0000000	0.0000000
ventilation_support_level_modified.441	-0.7953863	-0.0213046	0.0000000	-0.3448088
ventilation_support_level_modified.442	1.7765203	1.6776673	1.6149050	1.8209683
inspired_oxygen.44	-1.8064123	-0.4016768	0.1915731	-1.2696813
p_delta.44	-0.0266837	0.0324956	0.0166146	-0.0236806
peep_cm_h2o_modified.44	0.0887352	0.1847439	0.1914362	0.0606250
med_ph.44	0.8683420	0.4511735	0.4037232	0.8390917
bw	0.0000000	-0.0039995	0.0000545	0.0001273
ga	0.0000000	0.0527309	0.0000000	0.0000000
mat_chorioYes	0.0000000	0.3765575	0.0000000	0.0000000
genderMale	0.0000000	0.4591140	0.0000000	0.0000000
sgaSGA	0.0000000	0.0081330	0.0000000	0.0000000
any_surfYes	0.0000000	-0.1717863	0.0000000	0.0000000

Model Selection Analysis:

From the table of ceofficient above, we can see that the variables Obstetrical gestational age, Maternal Chorioamnionitis, gender, sga and surfactant are selected only the 44 week backward stepwise selected model. Another important observation when it comes to model selection is the fact that the 44 week lasso model only selects 6 variables which is significantly lower than the rest of the models. We also see that the lasso models are very strict when it comes to model selection. They don't select any variables outside the ones that are time specific(i.e the variables end with 36 or 44). The only exception to this is birth weight.

Intepretation:

The value of each coefficient represents the increase or decrease in the log odds of death or tracheostomy. The variables describing ventilation supportnat 44 weeks, weight, Medication for Pulmonary Hypertension at 36 week are less than zero across all models. This means that those who non-invasive positive pressure ventalation support at 44 weeks have lower log odds of death or tracheostomy compared to those who do invasive positive press conditioned that we keep other covariates constant. A negative coefficeint for weight at 36 weeks indicates that as the the weight of the patient at 36 increases, the log odds of death or trach decreases. These two interpretations makes sense.

Birth Variables: The variable sga is a discrete variable describing whether the baby is small for its gestational age. We find that the the sga variable is not selected by any of the models except the 44 week backward stepwise selected model. Since the corresponding coefficient is positive we can say that according this model, babies that are small for their age have a higher likelihood(log odds) of death or recieving tracheostomy.

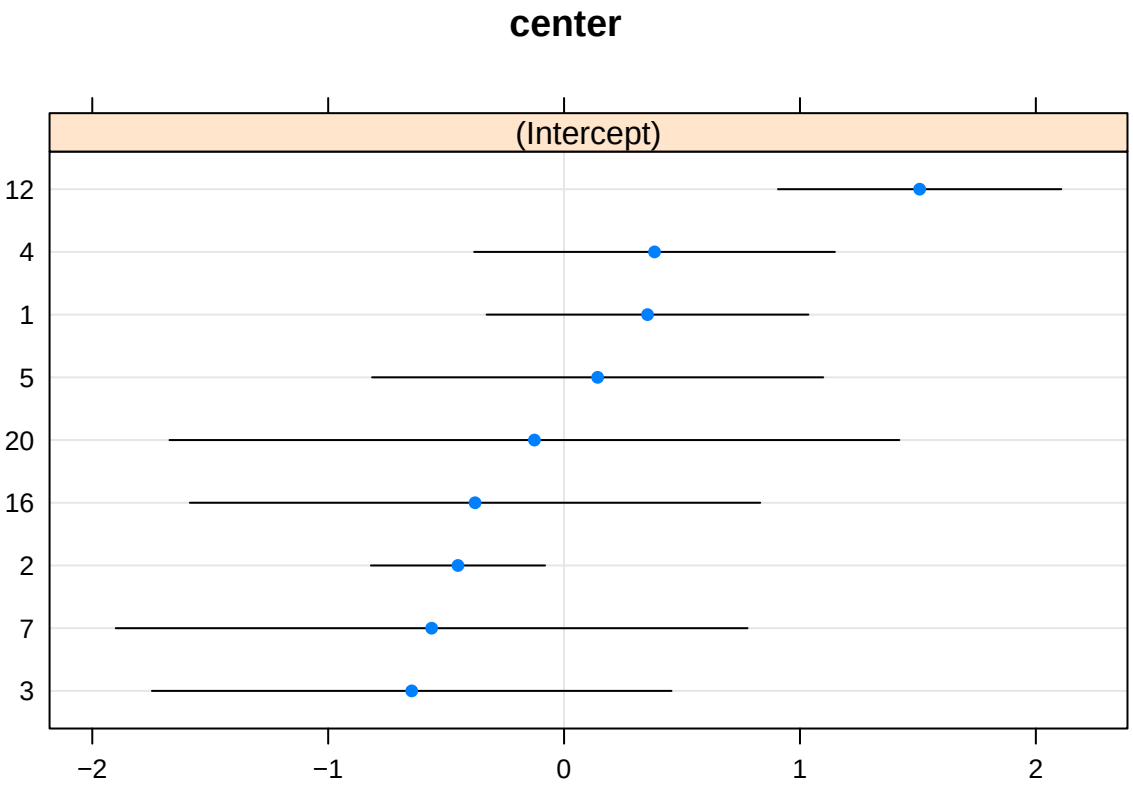
Comparing variables at 36 vs 44 weeks: To compare variables at 36 and 44 weeks we can look at the full lasso and backward selected models. The variable insired_oxygen.36 and inspired_oxygen.44 represent the fraction of inspired Oxygen at 36 and 44 weeks. The insired_oxygen.36 coeffcient is positive while the inspired_oxygen.44 is negative. This means that higher levels of inspired oxygen at 36 weeks is associated with higher log odds of death or tracheostomy while higher levels of inspired oxygen at 44 weeks is associated with lower log odds of death or trach. Another variable that behaves similary is med_ph.36 and med_ph.44. We see that the coefficient that represents medication for Pulmonary Hypertension(PH) at 36 and 44 weeks are of opposite signs. This indicates that patients who take medication for PH at 36 weeks have a lower log-odds of the death/trach compared to those who don't. Where was those who take medication for PH at 44 weeks have a higher log-odds of death/trach compared to those who don't. Finally the variable representing Peak Inspiratory Pressure (cmH2O) at 36 and 44 weeks are also of opposite signs, meaning that cmH2O levels have differing associations with death/trach at 36 and 44 weeks.

Next, we can study how the random effect of centers influence the outcome of death/trach for each center. ## Random effects visualized:

To visualize the random effects of the model we can first create a table that takes calculates the mean of each center's random effect accross each of the 4 models.

	Full Backwards	44 week backwards	Full Lasso	44 Week Lasso
center1	0.4409021	-0.0698791	0.4762948	0.1553881
center2	-0.4969382	-0.7458748	-0.4203236	-0.7260333
center3	-0.7357395	-0.0125981	-0.7808741	-0.2542637
center4	0.3737255	0.0000000	0.2396889	0.0000000
center5	0.1306986	-0.1258309	0.1318771	-0.0927979
center7	-0.5314760	-0.4255765	-0.6254847	-0.5330685
center12	1.4219031	1.6520966	1.4911689	1.7186271
center16	-0.2387709	-0.0704499	-0.2650720	-0.1218541
center20	-0.1355625	-0.0424631	-0.2472753	-0.1459978
variance	0.6955867	0.9929598	0.7439503	1.2440674

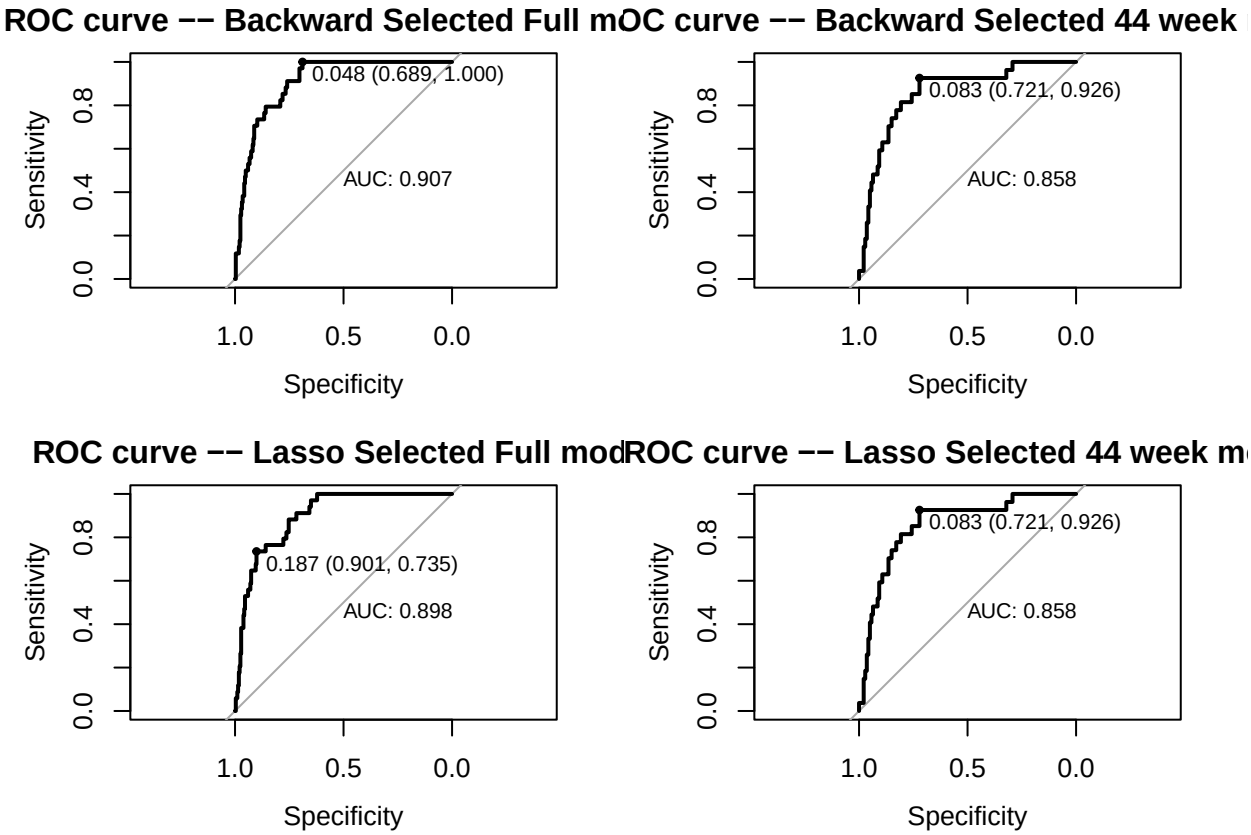
The table shows the average random effect of each center for each model along with average variance. We can see that patients in center 12 have the highest random effect that is positively associated with the log odds of death/tracheostomy, while centers 2, 3, and 7 have the most negative random effects. This indicates that those random effect of centers 2, 3, and 7 have the most negative influence towards the log odds of death/trach.



The plots above illustrate the random effects for the full backward selected model in a dot plot and its variance. We see a similar trend as the table.

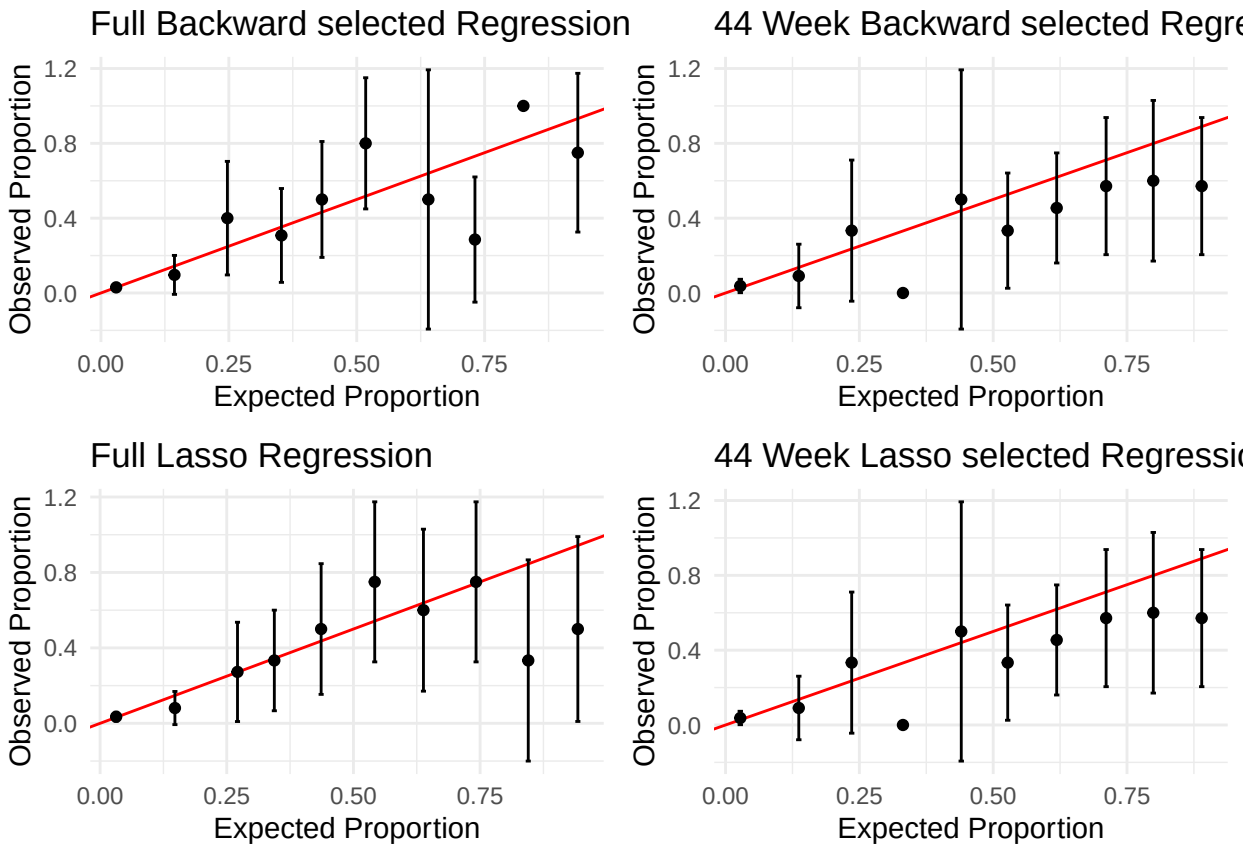
ROC and calibration curves for Backward stepwise and Lasso model

To evaluate the discriminiation and calibration of our models we will look at ROC curves along with calibration plots. When we test for calibration and discrimination we will use the test data.



The ROC curves for all 4 models show good discrimination among all 4 models as the ROC curves are above the specificy = sensitivity line. Additionally we see that the AUC scores are all above 89% which is good.

Plot Calibration curves for all models:



The calibration plots groups the data by the estimated probabilities and compares the mean probability with the observed proportion of observations of death/trach. The plots show that that the estimated distribution from the models are very close to the true distribution. The plots also include standard errors from a 95% confidence interval. The standard errors for the logistic regression model intersect with the red line(i.e the perfect fit where our estimated and true distributions match). It is worrisome that the standard error lines around 0.6 is so large for the full backward selected model. Additionally the confidence interval for the expected proportion around 0.75 does not intersect the red line. For the lasso model we see that the standard errors increase as the expected proportions increase.

A measure of calibration is the brier score. We summarize that in the table in the following section, along with the measures of accuracy and discrimination.

Prediction metrics

Model	AUC	Brier_Score	Accuracy	PPV	NPV	Sensitivity	Specifi
Backward Selected Full model	0.9071919	0.0713602	0.7223975	0.2786885	1.0000000	1.0000000	0.6890
Backward Selected 44 week model	0.8584656	0.1082728	0.7544910	0.3906250	0.9805825	0.9259259	0.7214
Lasso Selected Full mode	0.8975265	0.0729268	0.8832808	0.4716981	0.9659091	0.7352941	0.9010
Lasso Selected 44 week model	0.8841270	0.1082728	0.8383234	0.5000000	0.9593496	0.8148148	0.8428

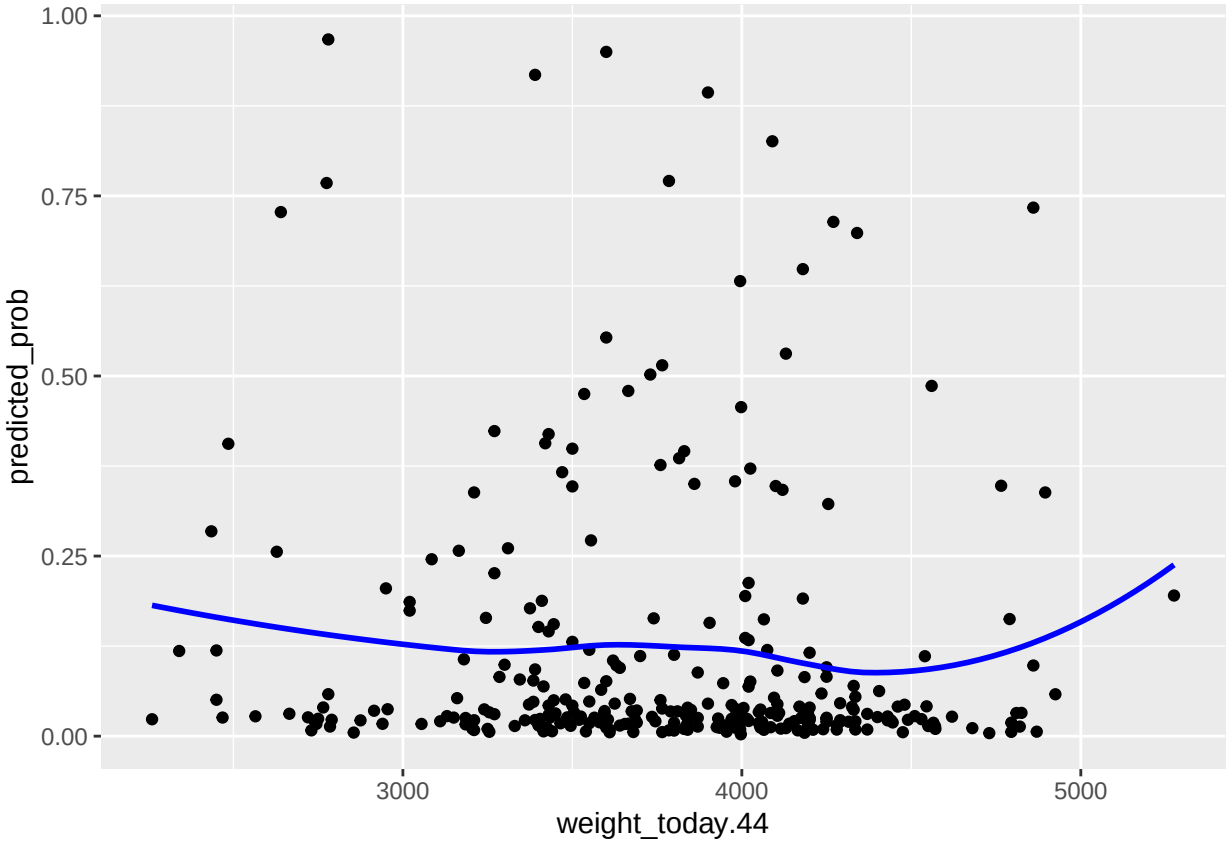
From the table above we find that the highest AUC score, accuracy and specificiy between the 4 models come from the backward selected model. The lowest Brier score also comes from the backward selected model. One worrying factor from the table above is the significanly low PPV(Positive Predictive value). This indicates a high false positive rate. This may be due to the fact that our data is not proportional between the outcomes

of death/trach(i.e the proportion of those who have died or recieved trach is significantly lower than those who did not). However because the prediction, discrimination and calibration scores are so close between the full model and the 44 week model, it may imply that the 36 week data is not needed to accurately predict the outcomes

Prediction on different ages

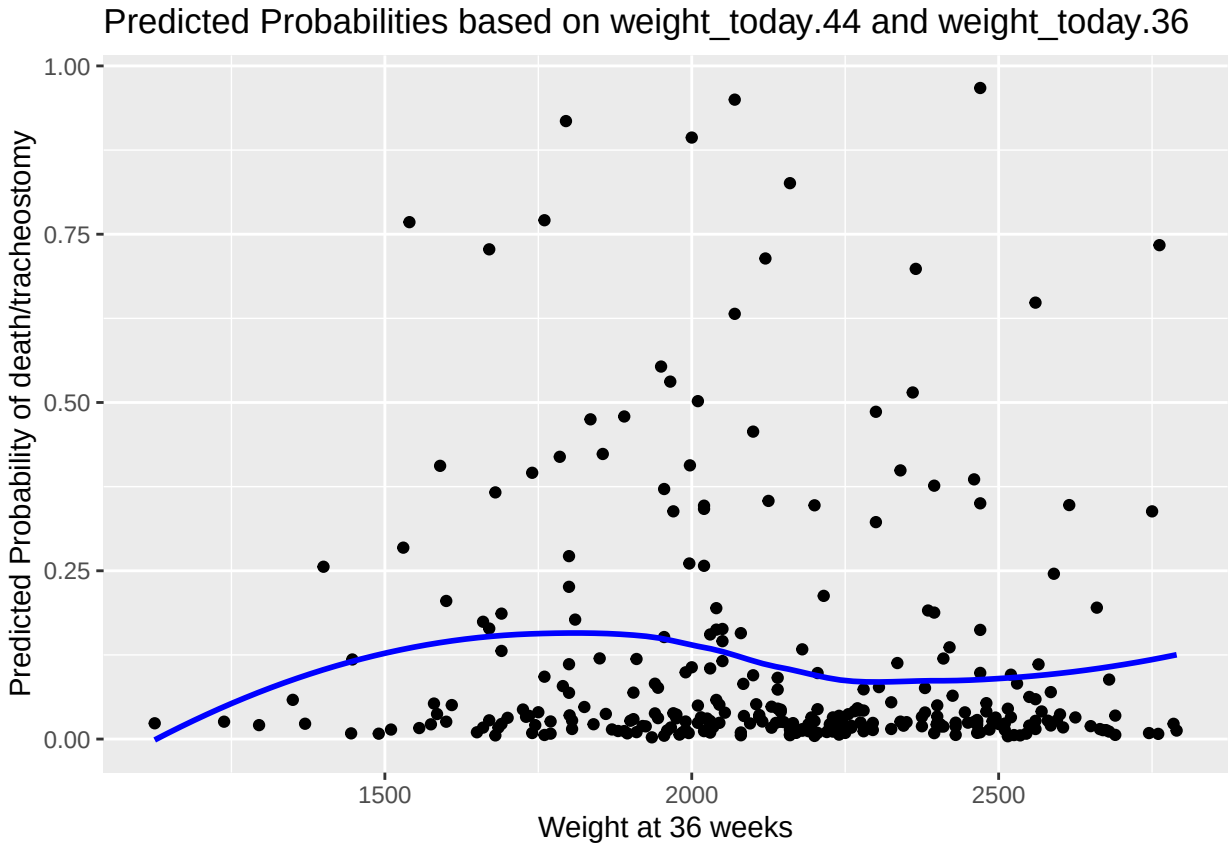
To find the timeframe at which death or trach is most likely to occur, we can plot weight at 36 and 44 weeks against the predicted probability of the outcome(trach or death).

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
## $x
## [1] "Weight at 44 weeks"
##
## $y
## [1] "Predicted Probability of death/tracheostomy"
##
## $title
## [1] "Predicted Probabilities based on weight_today.44 and weight_today.36"
##
## attr(,"class")
## [1] "labels"

## `geom_smooth()` using formula = 'y ~ x'
```



From the plot above we can see that the for the 36 week data, those around the weight of 1500 are more likely to get death/trach while for those in 44 week data, those below 2700 are the most likely to get death/trach.

Discussion and Conclusion

In conclusion, we found that a full mixed effect model is best for prediction for the outcome of death and trachoestomy. Additionally, implementing a full model with both the 36 week and 44 week data allowed us to compare the coefficeints between 36 and 44 week data. We found that the variables describing the fraction of inspired Oxygen, medication for Pulmonary Hypertension(PH) and Peak Inspiratory Pressure (cmH2O) has opposite effects on the log odds of death/tracheostomy. We also found that the influence andom effects of the centers on the log odds of the outcome accurately depict what we discussed in the EDA section.

Some of the setbacks of our work here is that we don't take into account some of the outliers in the data, although we did remove outliers heavily affecting the data we predicted on. There are many missing values that are indiscriminate between 36 weeks and 44 weeks which can sway the bias of the models. The model is difficult to generalize outside of centers included in the data. We don't use interaction terms.

Code Appendix

```
library(dplyr)
library(ggplot2)
library(HDSinRdata)
library(tidyverse)
library(egg)

library(tableone)
library(mice)
library(naniar)
library(gt)
library(gtsummary)
library(kableExtra)
library(lme4)
library(reshape2)
library(StatisticalModels)
library(glmmLasso)
library(pROC)

df <- read.csv("~/Downloads/project2.csv")
df.three =  read.csv("~/Downloads/project2.csv")
table(df$center) #Only one occurance of center 21-> delete it?
df = df[-810,]

df = df[-which(df$record_id == df$record_id[duplicated(df$record_id)]),] #remove repeated id's
df = subset(df, select = -c(mat_race, record_id) )
df_center = df$center

# Create a table to summarize the proportions of patients with trach and/or deaths
prop_outcome = xtabs(~Death + Trach, data=df) %>%
  prop.table %>%
  addmargins

#Table of proportion of deaths and trach.
df$Composite_Outcome <- ifelse(df$Death == 1 | df$Trach == 1, 1, 0) #prevalence of death is low.
df.loc.comp <- df %>% #include center as mixed
  dplyr::select(center, Composite_Outcome) %>%
  group_by(center) %>%
 summarise(Proportion =mean(Composite_Outcome, na.rm = TRUE), N = sum(!is.na(Composite_Outcome)))

#Create stratified table with important variables:



knitr::kable(df.loc.comp[complete.cases(df.loc.comp),])

summary_strat = df[,-1] %>%
  tbl_summary( by = Composite_Outcome,   statistic = list(
      all_continuous() ~ "{mean} ({sd})",
      all_categorical() ~ "{n} / {N} ({p}%)"
    )) %>%
  add_p(pvalue_fun = ~ style_pvalue(.x, digits = 2)) %>%
```

```
  add_overall() %>%
  add_n() %>%
  modify_header(label ~ "**Variable**") %>%
  modify_spanning_header(c("stat_1", "stat_2") ~ "**Death/Trach Composite outcome**") %>%
  modify_caption("**Data Summary stratified by Death/Trach Composite outcome**") %>%
  bold_labels() %>%
  as_kable_extra(booktabs = TRUE) %>%
  kableExtra::kable_styling(latex_options = "scale_down")%>%
   column_spec(1,width = "1in") %>%
  column_spec(2,width = "0.4in") %>%
  column_spec(3,width = "0.4in") %>%
  column_spec(4,width = "0.4in")

summary_strat

varMissingProp = miss_var_summary(df)
varMissingProp %>%
  filter(n_miss > 0) %>%
  kableExtra::kbl(caption = 'Missing Data Proportion for Each Variable'
                  , booktabs = T
                  , escape = T
                  , align = 'c'
                  , col.names = c('Variable','Observation Missing','Proportion
                                  Missing')) %>%
  kableExtra::kable_classic(full_width = F
                            , html_font = 'Cambria'
                            , latex_options = 'HOLD_position')

drop.cols = colnames(df[,colSums(is.na(df))==0])

df %>%
  dplyr::select(-one_of(drop.cols)) %>%
  gg_miss_var(show_pct = TRUE, facet = Composite_Outcome)  + ggtitle("Missing value percentage stratifi

# Check missing values and impute using mice package
apply(df, 2, function(x){return(sum(!is.na(x))/length(x))})
df.two = df
df.two = subset(df.two, select = -c(Death, Trach, hosp_dc_ga) )

#drop center and mat_ethn
char_columns <- sapply(df.two, is.character)
char_columns = c(colnames(df.two)[char_columns], "Trach", "ventilation_support_level_modified.44", "ven
#df.two[char_columns] <- lapply(df.two[char_columns], factor)
df.two[colnames(df.two) %in% char_columns] <- lapply(df.two[colnames(df.two) %in% char_columns], factor

df.two[c("blength", "ga", "weight_today.36", "weight_today.44", "peep_cm_h2o_modified.36", "peep_cm_h2o

set.seed(2550)

#use 70% of dataset as training set and 30% as test set
ignore <- sample(c(TRUE, FALSE), size =  dim(df.two)[1], replace = TRUE, prob = c(0.3, 0.7))
```

```
df_mice_out <- mice(df.two, 5, pri=F, seed = 2550, ignore = ignore)
#df_mice_out <- mice(df.two, 5, pri=F, seed = 2550)
imp.test1 <- filter(df_mice_out, ignore) #test set
df_test <- vector("list",5)

for (i in 1:5){
  df_test[[i]] <- mice::complete(imp.test1,i)
  df_test[[i]]$Composite_Outcome = as.integer(df_test[[i]]$Composite_Outcome)
  df_test[[i]]$center = as.factor(df_test[[i]]$center)

}

# Store each imputed train set
df_train <- vector("list",5)

imp.train <- filter(df_mice_out, !ignore) #test set
df_train <- vector("list",5)
for (i in 1:5){
  df_train[[i]] <- mice::complete(imp.train,i)
  df_train[[i]]$Composite_Outcome = as.integer(df_train[[i]]$Composite_Outcome)
  df_train[[i]]$center = as.factor(df_train[[i]]$center)

}


df.44.cc = df.two[complete.cases(df.two$inspired_oxygen.44), ]
apply(df.44.cc, 2, function(x){return(sum(!is.na(x))/length(x))})
ignore <- sample(c(TRUE, FALSE), size =  dim(df.44.cc)[1], replace = TRUE, prob = c(0.3, 0.7))



df_mice_out.44 <- mice(df.44.cc, 5, pri=F, seed = 2550, ignore = ignore)
imp.test.44 <- filter(df_mice_out.44, ignore) #test set
df_test.44 <- vector("list",5)

for (i in 1:5){
  df_test.44[[i]] <- mice::complete(imp.test.44,i)
  df_test.44[[i]]$Composite_Outcome = as.integer(df_test.44[[i]]$Composite_Outcome)
  df_test.44[[i]]$center = as.factor(df_test.44[[i]]$center)
}

# Store each imputed train set
df_train.44 <- vector("list",5)

imp.train.44 <- filter(df_mice_out.44, !ignore) #test set
df_train.44 <- vector("list",5)
for (i in 1:5){
  df_train.44[[i]] <- mice::complete(imp.train.44,i)
  df_train.44[[i]]$Composite_Outcome = as.integer(df_train.44[[i]]$Composite_Outcome)
  df_train.44[[i]]$center = as.factor(df_train.44[[i]]$center)

}
########Lasso selection##########
```

```
lasso = function(data_df, week_all){
#set lambdas... go from 0 to 10^5, in 10 log steps
lambda <- 10^seq(-3,5, length=10)


#dummy vectors of model fit values for each lambda: BIC, AIC, prediction error


BIC_vec <- rep(Inf, length(lambda))
AIC_vec <- rep(Inf, length(lambda))
Devianz_ma<-NULL
Coeff_ma<-NULL


family = binomial(link = "logit")


for (j in 1:length(BIC_vec)){
  if(week_all == TRUE){
 glm1 <-
 glmmLasso(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chorio+ gender+wei
 data= data_df,
 rnd = list(center=~1),
 family =  binomial(link = "logit"),
 lambda = lambda[j],
 final.re = TRUE)} else if(week_all == FALSE){
    glm1 <-
 glmmLasso(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chorio+ gender+sga
 data= data_df,
 rnd = list(center=~1),
 family =  binomial(link = "logit"),
 lambda = lambda[j],
 final.re = TRUE)
 }

# code to make it continue anyway if an error occurs
 # if(class(glm1)!="try-error")
# {

 #save BIC, AIC
 BIC_vec[j]<-glm1$bic
 AIC_vec[j]<-glm1$aic

 #save coefficient outputs
 Coeff_ma<-cbind(Coeff_ma,glm1$coefficients)

 #save error (deviance) values
 y.hat<-predict(glm1,data_df)
 Devianz_ma[j]<-sum(family$dev.resids(data_df$Composite_Outcome,y.hat,wt=rep(1,length(y.hat))))

# }
}
```

```
#these are the possible different optimized lambda values, based on different criteria
lambda.final = mean(lambda[which.min(BIC_vec)], lambda[which.min(AIC_vec)])
if(week_all == TRUE){
glm_lasso =  glmmLasso(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chori
 data= data_df,
 rnd = list(center=~1),
 family =  binomial(link = "logit"),
 lambda = lambda.final,
 final.re = TRUE)}else if(week_all ==FALSE){
   glm_lasso =  glmmLasso(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_ch
 data= data_df,
 rnd = list(center=~1),
 family =  binomial(link = "logit"),
 lambda = lambda.final,
 final.re = TRUE)
 }

  return(glm_lasso)

}

## lasso coefficients with variables of 36 and 44 :
lasso_coef1 <- lasso(df_train[[1]], TRUE)
lasso_coef2 <- lasso(df_train[[2]], TRUE)
lasso_coef3 <- lasso(df_train[[3]], TRUE)
lasso_coef4 <- lasso(df_train[[4]], TRUE)
lasso_coef5 <- lasso(df_train[[5]], TRUE)
lasso_coef <- cbind(lasso_coef1$coef, lasso_coef2$coef, lasso_coef3$coef, lasso_coef4$coef, lasso_coef5
avg_coefs_lasso.full <- apply(lasso_coef, 1, mean)

#lasso for 44 week data only:
lasso_coef.44.1 <- lasso(df_train.44[[1]], FALSE)
lasso_coef.44.2 <- lasso(df_train.44[[2]], FALSE)
lasso_coef.44.3 <- lasso(df_train.44[[3]], FALSE)
lasso_coef.44.4 <- lasso(df_train.44[[4]], FALSE)
lasso_coef.44.5 <- lasso(df_train.44[[5]], FALSE)
lasso_coef <- cbind(lasso_coef.44.1$coef, lasso_coef.44.2$coef, lasso_coef.44.3$coef,
                    lasso_coef.44.4$coef, lasso_coef.44.5$coef)
avg_coefs_lasso.44 <- apply(lasso_coef, 1, mean)


############Backward Stepwise Model selection############
full_mod = glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chorio+ gende
 data= df_train[[1]])

#back on full model:
back.full.1 = step(full_mod, direction = "backward")

back.full.2 = step(glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chori
 data= df_train[[2]]), direction = "backward")

back.full.3 = step(glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chori
 data= df_train[[3]]),  direction = "backward")
```

```
back.full.4 = step(glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chori
 data= df_train[[4]]), direction = "backward")

back.full.5 = step(glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chori
 data= df_train[[5]]), direction = "backward")

all_coefs <- unique(unlist(lapply(list(back.full.1$coef, back.full.2$coef, back.full.3$coef, back.full.

avg_coefs_back.full <- sapply(all_coefs, function(coef_name) {
  coef_values <- sapply(list(back.full.1$coef, back.full.2$coef, back.full.3$coef, back.full.4$coef, ba
    if (coef_name %in% names(model_coef)) {
      return(model_coef[coef_name])
    } else {
      return(0)
    }
  })
  return(mean(coef_values))
})

me.full.1 = glmer(Composite_Outcome ~ birth_hc+com_prenat_ster+blength+  prenat_ster+ventilation_suppor

me.full.2 = glmer(Composite_Outcome ~ birth_hc+com_prenat_ster+blength+  prenat_ster+ventilation_suppor

me.full.3 = glmer(Composite_Outcome ~ birth_hc+com_prenat_ster+blength+  prenat_ster+ventilation_suppor


me.full.4 = glmer(Composite_Outcome ~ birth_hc+com_prenat_ster+blength+  prenat_ster+ventilation_suppor


me.full.5 = glmer(Composite_Outcome ~ birth_hc+com_prenat_ster+blength+  prenat_ster+ventilation_suppor

step_coef <- cbind(fixef(me.full.1), fixef(me.full.2),fixef(me.full.3),fixef(me.full.4), fixef(me.full.
avg_coefs_backstep.full <- apply(step_coef, 1, mean)


#backward stepwise on only 44 weeks.
four_mod.1 = glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chorio+ gen
               data= df_train.44[[1]])
four_mod.2= glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chorio+ gend
               data= df_train.44[[2]])
four_mod.3 = glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chorio+ gen
               data= df_train.44[[3]])
four_mod.4 = glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chorio+ gen
               data= df_train.44[[4]])
four_mod.5 = glm(Composite_Outcome ~ bw+ga+blength+birth_hc+prenat_ster+com_prenat_ster+mat_chorio+ gen
               data= df_train.44[[5]])

all_coefs <- unique(unlist(lapply(list(four_mod.1$coef, four_mod.2$coef, four_mod.3$coef, four_mod.4$co

avg_coefs_back.44 <- sapply(all_coefs, function(coef_name) {
  coef_values <- sapply(list(four_mod.1$coef, four_mod.2$coef, four_mod.3$coef, four_mod.4$coef, four_m
    if (coef_name %in% names(model_coef)) {
      return(model_coef[coef_name])
```

```
    } else {
      return(0)
    }
  })
  return(mean(coef_values))
})

me.44.1 = glmer(Composite_Outcome ~ bw+ga+blength+birth_hc+com_prenat_ster+  prenat_ster+mat_chorio + g

me.44.2 = glmer(Composite_Outcome ~ bw+ga+blength+birth_hc+com_prenat_ster+  prenat_ster+mat_chorio + g

me.44.3 = glmer(Composite_Outcome ~ bw+ga+blength+birth_hc+com_prenat_ster+  prenat_ster+mat_chorio + g

me.44.4 = glmer(Composite_Outcome ~ bw+ga+blength+birth_hc+com_prenat_ster+  prenat_ster+mat_chorio + g

me.44.5 = glmer(Composite_Outcome ~ bw+ga+blength+birth_hc+com_prenat_ster+  prenat_ster+mat_chorio + g

step_coef.44 <- cbind(fixef(me.44.1), fixef(me.44.2),fixef(me.44.3),fixef(me.44.4), fixef(me.44.5))
avg_coefs_backstep.44 <- apply(step_coef.44, 1, mean)

#make dataframe to compare coefficients
all_coefficients <- unique(c(names(avg_coefs_backstep.full), names(avg_coefs_backstep.44), names(avg_co

# Create a data frame with coefficients from all models
coefficients_table <- data.frame(
  Coefficient = all_coefficients,
  avg_coefs_backstep.full = ifelse(all_coefficients %in% names(avg_coefs_backstep.full), avg_coefs_backs
  avg_coefs_backstep.44 = ifelse(all_coefficients %in% names(avg_coefs_backstep.44), avg_coefs_backstep
  avg_coefs_lasso.44 = ifelse(all_coefficients %in% names(avg_coefs_lasso.44), avg_coefs_lasso.44[all_c
  avg_coefs_lasso.full = ifelse(all_coefficients %in% names(avg_coefs_lasso.full), avg_coefs_lasso.full
)
#coefficients_table$avg_coefs_backstep.full <- exp(coefficients_table$avg_coefs_backstep.full)
#coefficients_table$avg_coefs_backstep.44 <- exp(coefficients_table$avg_coefs_backstep.44)
#coefficients_table$avg_coefs_lasso.44 <- exp(coefficients_table$avg_coefs_lasso.44)
#coefficients_table$avg_coefs_lasso.full <- exp(coefficients_table$avg_coefs_lasso.full)

# Print the coefficients table
colnames(coefficients_table) = c("Coefficient", "Full Backward", "44 week Backstep", "44 week Lasso", "
knitr::kable(coefficients_table)

var.full.back = mean(as.numeric(VarCorr(me.full.1)), as.numeric(VarCorr(me.full.2)), as.numeric(VarCorr

var.44.back = mean(as.numeric(VarCorr(me.44.1)), as.numeric(VarCorr(me.44.2)), as.numeric(VarCorr(me.44



ran_ef.full.back = rowMeans(data.frame(ranef(me.full.1, which = "center", condVar = TRUE)$center, ranef
ran_ef.full.back["variance"] = var.full.back

ran_ef.44.back = rowMeans(data.frame(ranef(me.44.1, which = "center", condVar = TRUE)$center, ranef(me.
ran_ef.44.back["variance"] = var.44.back

ran_ef.full.lasso = rowMeans(data.frame(lasso_coef1$ranef, lasso_coef2$ranef, lasso_coef3$ranef,lasso_c
```

```r
ran_ef.full.lasso["variance"] = mean(as.numeric(lasso_coef1$StdDev^2), as.numeric(lasso_coef2$StdDev^2)

ran_ef.44.lasso = rowMeans(data.frame(lasso_coef.44.1$ranef, lasso_coef.44.2$ranef, lasso_coef.44.3$ran

ran_ef.44.lasso["variance"] = mean(as.numeric(lasso_coef.44.1$StdDev^2), as.numeric(lasso_coef.44.2$Std


ran_ef.full.back = data.frame(ran_ef.full.back)
ran_ef.44.back <- c(ran_ef.44.back[1:3], 0, ran_ef.44.back[4:length(ran_ef.44.back)])
ran_ef.44.back = data.frame(ran_ef.44.back)
ran_ef.full.lasso = data.frame(ran_ef.full.lasso)
ran_ef.44.lasso <- c(ran_ef.44.lasso[1:3], 0, ran_ef.44.lasso[4:length(ran_ef.44.lasso)])
ran_ef.44.lasso = data.frame(ran_ef.44.lasso)

row.names(ran_ef.44.back) = row.names(ran_ef.44.lasso)
row.names(ran_ef.full.back) = row.names(ran_ef.full.lasso)


ran_ef_df = data.frame(ran_ef.full.back, ran_ef.44.back, ran_ef.full.lasso, ran_ef.44.lasso)
colnames(ran_ef_df) = c("Full Backwards", "44 week backwards", "Full Lasso", "44 Week Lasso")
knitr::kable(ran_ef_df)

# Print the merged data frame
lattice::dotplot(ranef(me.full.1, which = "center", condVar = TRUE), title = "First Random Effect of ful


#full backwards model
models = list(me.full.1, me.full.2, me.full.3, me.full.4, me.full.5)
predictions_list <- lapply(1:5, function(i) {
  predict(models[[i]], newdata = df_test[[1]], type = "response")
})
combined_predictions.back.full <- rowMeans(do.call(cbind, predictions_list))
roc_curve.back.full <- roc(response = df_test[[1]]$Composite_Outcome, predictor = combined_predictions.
auc_value.back.full <- auc(roc_curve.back.full)


plot(roc_curve.back.full ,main ="Backward Selected Full model", print.auc=TRUE, print.thres = TRUE)


#calibration
brier.back.full = mean((combined_predictions.back.full - (as.numeric(df_test[[1]]$Composite_Outcome)))^

#calibration: make tables of Brier Scores
num_cuts <- 10
calib_data.back.full <-  data.frame(prob = combined_predictions.back.full,
                          bin = cut(combined_predictions.back.full, breaks = num_cuts),
                          class = df_test[[1]]$Composite_Outcome)
calib_data.back.full <- calib_data.back.full %>%
             group_by(bin) %>%
             dplyr::summarize(observed = sum(class)/n(),
                       expected = sum(prob)/n(),
                       se = sqrt(observed*(1-observed)/n()))
#calib_data
```

```r
calib.back.full = ggplot(calib_data.back.full) +
  geom_abline(intercept = 0, slope = 1, color="red") +
  geom_errorbar(aes(x = expected, ymin=observed-1.96*se,
                    ymax=observed+1.96*se),
                colour="black", width=.01)+
  geom_point(aes(x = expected, y = observed)) +
  labs(x="Expected Proportion", y="Observed Proportion") +
  ggtitle("Full Backward selected Regression")+

  theme_minimal()

# 44 backwards model
models = list(me.44.1, me.44.2, me.44.5, me.44.4, me.44.5)
predictions_list <- lapply(1:5, function(i) {
  predict(models[[i]], newdata = df_test.44[[1]], type = "response")
})
combined_predictions.back.44 <- rowMeans(do.call(cbind, predictions_list))
roc_curve.back.44 <- roc(response = df_test.44[[1]]$Composite_Outcome, predictor = combined_predictions
auc_value.back.44 <- auc(roc_curve.back.44)


plot(roc_curve.back.44, main ="Backward Selected 44 week model", print.auc=TRUE, print.thres = TRUE)


#calibration
brier.back.44 = mean((combined_predictions.back.44 - (as.numeric(df_test.44[[1]]$Composite_Outcome)))^2

#calibration: make tables of Brier Scores
num_cuts <- 10
calib_data.back.44 <-  data.frame(prob = combined_predictions.back.44,
                          bin = cut(combined_predictions.back.44, breaks = num_cuts),
                          class = df_test.44[[1]]$Composite_Outcome)
calib_data.back.44 <- calib_data.back.44 %>%
             group_by(bin) %>%
             dplyr::summarize(observed = sum(class)/n(),
                       expected = sum(prob)/n(),
                       se = sqrt(observed*(1-observed)/n()))
#calib_data
calib_data.back.44 = ggplot(calib_data.back.44) +
  geom_abline(intercept = 0, slope = 1, color="red") +
  geom_errorbar(aes(x = expected, ymin=observed-1.96*se,
                    ymax=observed+1.96*se),
                colour="black", width=.01)+
  geom_point(aes(x = expected, y = observed)) +
  labs(x="Expected Proportion", y="Observed Proportion") +
  ggtitle("44 Week Backward selected Regression")+

  theme_minimal()

#full backwards model
models = list(lasso_coef1, lasso_coef2, lasso_coef3, lasso_coef4, lasso_coef5)
predictions_list <- lapply(1:5, function(i) {
  predict(models[[i]], newdata = df_test[[1]], type = "response")
```

```
})
#######START HERE##########
combined_predictions.lasso.full <- rowMeans(do.call(cbind, predictions_list))
roc_curve.lasso.full <- roc(response = df_test[[1]]$Composite_Outcome, predictor = combined_predictions
auc_value.lasso.full <- auc(roc_curve.lasso.full)


plot(roc_curve.lasso.full ,main ="ROC curve -- Lasso Selected Full model", print.auc=TRUE, print.thres =


#calibration
brier.lasso.full = mean((combined_predictions.lasso.full - (as.numeric(df_test[[1]]$Composite_Outcome))

#calibration: make tables of Brier Scores
num_cuts <- 10
calib_data.lasso.full <-  data.frame(prob = combined_predictions.lasso.full,
                          bin = cut(combined_predictions.lasso.full, breaks = num_cuts),
                          class = df_test[[1]]$Composite_Outcome)
calib_data.lasso.full <- calib_data.lasso.full %>%
             group_by(bin) %>%
             dplyr::summarize(observed = sum(class)/n(),
                       expected = sum(prob)/n(),
                       se = sqrt(observed*(1-observed)/n()))
#calib_data
calib_data.lasso.full = ggplot(calib_data.lasso.full) +
  geom_abline(intercept = 0, slope = 1, color="red") +
  geom_errorbar(aes(x = expected, ymin=observed-1.96*se,
                    ymax=observed+1.96*se),
                colour="black", width=.01)+
  geom_point(aes(x = expected, y = observed)) +
  labs(x="Expected Proportion", y="Observed Proportion") +
  ggtitle("Full Lasso Regression")+

  theme_minimal()

# 44 lasso model
models = list(lasso_coef.44.1, lasso_coef.44.2, lasso_coef.44.3, lasso_coef.44.4, lasso_coef.44.5)
predictions_list <- lapply(1:5, function(i) {
  predict(models[[i]], newdata = df_test.44[[1]], type = "response")
})
combined_predictions.lasso.44 <- rowMeans(do.call(cbind, predictions_list))
roc_curve.lasso.44 <- roc(response = df_test.44[[1]]$Composite_Outcome, predictor = combined_predictions
auc_value.lasso.44 <- auc(roc_curve.lasso.44)


plot(roc_curve.lasso.44, main ="ROC curve -- Lasso Selected 44 week model", print.auc=TRUE, print.thres


#calibration
brier.lasso.44 = mean((combined_predictions.back.44 - (as.numeric(df_test.44[[1]]$Composite_Outcome)))^

#calibration: make tables of Brier Scores
num_cuts <- 10
```

```
calib_data.lasso.44 <-  data.frame(prob = combined_predictions.back.44,
                          bin = cut(combined_predictions.back.44, breaks = num_cuts),
                          class = df_test.44[[1]]$Composite_Outcome)
calib_data.lasso.44 <- calib_data.lasso.44 %>%
             group_by(bin) %>%
             dplyr::summarize(observed = sum(class)/n(),
                       expected = sum(prob)/n(),
                       se = sqrt(observed*(1-observed)/n()))
#calib_data
calib_data.lasso.44 = ggplot(calib_data.lasso.44) +
  geom_abline(intercept = 0, slope = 1, color="red") +
  geom_errorbar(aes(x = expected, ymin=observed-1.96*se,
                    ymax=observed+1.96*se),
                colour="black", width=.01)+
  geom_point(aes(x = expected, y = observed)) +
  labs(x="Expected Proportion", y="Observed Proportion") +
  ggtitle("44 Week Lasso selected Regression")+

  theme_minimal()

par(mfrow=c(2,2))
plot(roc_curve.back.full ,main ="ROC curve -- Backward Selected Full model", print.auc=TRUE, print.thres
plot(roc_curve.back.44, main ="ROC curve -- Backward Selected 44 week model", print.auc=TRUE, print.thr
plot(roc_curve.lasso.full ,main ="ROC curve -- Lasso Selected Full model", print.auc=TRUE, print.thres =
plot(roc_curve.back.44, main ="ROC curve -- Lasso Selected 44 week model", print.auc=TRUE, print.thres =
library(egg)
ggarrange(calib.back.full,
calib_data.back.44,
calib_data.lasso.full,
calib_data.lasso.44, ncol = 2, nrow =2)
model_names <- c("Backward Selected Full model", "Backward Selected 44 week model", "Lasso Selected Full
auc_scores <- c(auc_value.back.full, auc_value.back.44, auc_value.lasso.full, auc_value.lasso.44)
brier_scores <- c(brier.back.full, brier.back.44, brier.lasso.full, brier.lasso.44)

# Create a data frame
model_data <- data.frame(Model = model_names, AUC = auc_scores, Brier_Score = brier_scores)

pred_ys <- ifelse(combined_predictions.back.full > as.numeric(coords(roc_curve.back.full, "best", ret =
tab_outcome <- table(df_test[[1]]$Composite_Outcome, pred_ys)
sens.back.full <- tab_outcome[2,2]/(tab_outcome[2,1]+tab_outcome[2,2])
spec.back.full <- tab_outcome[1,1]/(tab_outcome[1,1]+tab_outcome[1,2])
ppv.back.full <- tab_outcome[2,2]/(tab_outcome[1,2]+tab_outcome[2,2])
npv.back.full  <- tab_outcome[1,1]/(tab_outcome[1,1]+tab_outcome[2,1])
acc.back.full  <- (tab_outcome[1,1]+tab_outcome[2,2])/sum(tab_outcome)

pred_ys <- ifelse(combined_predictions.back.44 > as.numeric(coords(roc_curve.back.44, "best", ret = "thr
tab_outcome <- table(df_test.44[[1]]$Composite_Outcome, pred_ys)
sens.back.44 <- tab_outcome[2,2]/(tab_outcome[2,1]+tab_outcome[2,2])
spec.back.44 <- tab_outcome[1,1]/(tab_outcome[1,1]+tab_outcome[1,2])
ppv.back.44 <- tab_outcome[2,2]/(tab_outcome[1,2]+tab_outcome[2,2])
npv.back.44  <- tab_outcome[1,1]/(tab_outcome[1,1]+tab_outcome[2,1])
acc.back.44 <- (tab_outcome[1,1]+tab_outcome[2,2])/sum(tab_outcome)
```

```r
pred_ys <- ifelse(combined_predictions.lasso.full > as.numeric(coords(roc_curve.lasso.full, "best", ret
, 1, 0)
tab_outcome <- table(df_test[[1]]$Composite_Outcome, pred_ys)
sens.lasso.full <- tab_outcome[2,2]/(tab_outcome[2,1]+tab_outcome[2,2])
spec.lasso.full <- tab_outcome[1,1]/(tab_outcome[1,1]+tab_outcome[1,2])
ppv.lasso.full <- tab_outcome[2,2]/(tab_outcome[1,2]+tab_outcome[2,2])
npv.lasso.full <- tab_outcome[1,1]/(tab_outcome[1,1]+tab_outcome[2,1])
acc.lasso.full<- (tab_outcome[1,1]+tab_outcome[2,2])/sum(tab_outcome)

pred_ys <- ifelse(combined_predictions.lasso.44 > as.numeric(coords(roc_curve.lasso.44, "best", ret = "
, 1, 0)
tab_outcome <- table(df_test.44[[1]]$Composite_Outcome, pred_ys)
sens.lasso.44 <- tab_outcome[2,2]/(tab_outcome[2,1]+tab_outcome[2,2])
spec.lasso.44 <- tab_outcome[1,1]/(tab_outcome[1,1]+tab_outcome[1,2])
ppv.lasso.44 <- tab_outcome[2,2]/(tab_outcome[1,2]+tab_outcome[2,2])
npv.lasso.44 <- tab_outcome[1,1]/(tab_outcome[1,1]+tab_outcome[2,1])
acc.lasso.44<- (tab_outcome[1,1]+tab_outcome[2,2])/sum(tab_outcome)

model_data$Accuracy = c(acc.back.full, acc.back.44, acc.lasso.full, acc.lasso.44)
model_data$PPV = c(ppv.back.full, ppv.back.44, ppv.lasso.full, ppv.lasso.44)
model_data$NPV = c(npv.back.full, npv.back.44, npv.lasso.full, npv.lasso.44)
model_data$Sensitivity = c(sens.back.full, sens.back.44, sens.lasso.full, sens.lasso.44)
model_data$Specificity = c(spec.back.full, spec.back.44, spec.lasso.full, spec.lasso.44)

knitr::kable(model_data)
# Assuming logistic_model is your logistic regression model
# Assuming your_data is your dataset

# Create a grid of values for weight_today.44 and weight_today.36

weight_today.44_values <- seq(min(df.two$weight_today.44[-3], na.rm = TRUE), max(df.two$weight_today.44
weight_today.36_values <- seq(min(df.two$weight_today.36, na.rm = TRUE), max(df.two$weight_today.36, na

# Create a data frame with all combinations of weight_today.44 and weight_today.36
grid <- data.frame(expand.grid(weight_today.44 = weight_today.44_values, weight_today.36 = weight_today
grid = df_test[[1]][-which(df_test[[1]]$weight_today.44 < 2200 | df_test[[1]]$weight_today.36 < 1000 | d
#grid = grid[-which(df_test[[1]]$weight_today.36 < 1000 ),]
# Make predictions for each combination
#full backwards model
models = list(me.full.1, me.full.2, me.full.3, me.full.4, me.full.5)
predictions_list <- lapply(1:5, function(i) {
  predict(models[[i]], newdata = grid, type = "response")
})
grid$predicted_prob  <- rowMeans(do.call(cbind, predictions_list))
# Plot the predicted probabilities

ggplot(grid, aes(x = weight_today.44, y = predicted_prob)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "blue")
  labs(title = "Predicted Probabilities based on weight_today.44 and weight_today.36",
       x = "Weight at 44 weeks",
       y = "Predicted Probability of death/tracheostomy")
```

```r
ggplot(grid, aes(x = weight_today.36, y = predicted_prob)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  labs(title = "Predicted Probabilities based on weight_today.44 and weight_today.36",
       x = "Weight at 36 weeks",
       y = "Predicted Probability of death/tracheostomy")
```