Project 2: Predicting Tracheostomy/Death in Neonates with Severe BPD: a comprehensive mix-effect logistic regression model accounting for patients with different discharge age

Mavis (Xinwen) Liang

1 Abstract

Accurate prediction of the need for tracheostomy at early postmenstrual ages (PMA) can have significant implications for families and clinical decision-making. In this analysis, we utilized a national dataset containing demographic, diagnostic, and respiratory parameters of infants with sBPD to develop a logistic regression model to predict the composite outcome of tracheostomy or death in neonates with severe bronchopulmonary dysplasia (sBPD), with a focus on assessing the impact of missing data and accounting for patients discharged before or after 44 week. The model has an overall good performance when we validate it using the test se, and has the potential implications for clinical practice in predicting Tracheostomy/Death.

The code for this analysis can be found in Github: https://github.com/Mavis-Liang/pda_project2.

2 Introduction

This collaborative project, undertaken in partnership with Dr. Chris Schmid from the Biostatistics Department at Brown University, aims to address a critical gap in understanding the indication criteria and timing of tracheostomy placement in neonates with severe bronchopulmonary dysplasia (sBPD). The project is motivated by the lack of clarity surrounding the optimal timing for tracheostomy placement in this population and the potential benefits of early intervention for growth.

Data for this study were sourced from the BPD Collaborative Registry, a multi-center consortium comprising interdisciplinary BPD programs in the United States and Sweden. The

registry includes infants with a gestational age of less than 32 weeks and diagnosed with severe bronchopulmonary dysplasia (sBPD) based on 2001 NHLBI criteria. The study specifically considers infants requiring FiO2 0.3 or positive pressure ventilation (invasive or non-invasive) at 36 weeks PMA. Standard demographic and clinical data are collected at four key time points: birth, 36 weeks PMA, 44 weeks PMA, and discharge.

The inclusion criteria for this study involve patients with BPD and complete growth data, and the data query was conducted for patients meeting these criteria between January 1 and July 19, 2021. The analysis incorporates data contributed by 10 BPD Collaborative centers, reflecting a diverse and representative cohort.

Throughout the statistical analysis report, we built a lasso logistic regression model that can account for patients discharge early or later altogether, after missing value imputations. This model has the potential implications for clinical practice and patient outcomes.

3 Methods

3.1 Data Preprocessing

Variable selection and transformation

The two outcomes (Tracheotomy and Death) are combined into one singel outcome of `trach_or_death`, since the proportion of death is too low in the data, and, on the other hand, both tracheotomy and death can indicate a severe outcome of sBPD in newborns.

The covariates and how they were processed are summarized as below:

Categorical variables: (1) Medical center: the medical center of number 21 has only one data point, thus we dropped this data point. (2) Maternal race. (3) Maternal Ethnicity. (4) Delivery Method. (5) Prenatal Corticosteroids. (6) Complete Prenatal Steroids: it's highly incomplete and closely associated with Prenatal Corticosteroids, thus we dropped this variable in our model. (7) Gender. (8) Was the infant small for gestational age. (9) Did the infant receive surfactant at any point in the first 72 hours? : We dropped this cavariates due to its high incompleteness. (10) Ventilation support level at 36 weeks. (11) Medication for Pulmonary Hypertension at 36 weeks. (12) Ventilation support level at 44 weeks. (13) Medication for Pulmonary Hypertension at 44 weeks.

Contiuous variables: (1) Birth weight (g): transformed into linear spline terms at the knot of 1610. (2) Obstetrical gestational age. (3) Birth length (cm): transformed into linear spline terms at the knot of 40. (4) Birth head circumference (cm). (5) Weight at 36 weeks: : transformed into linear spline terms at the knots of 1510 and 2710. (6) Fraction of Inspired Oxygen at 36 weeks: : transformed into linear spline terms at the knot of 0.789. (7) Peak Inspiratory Pressure (cmH2O) at 36 weeks. (8) Positive and exploratory pressure (cm H2O) at 36 weeks. (9) Weight at 44 weeks: transformed into linear spline terms at the knot of 2110.

(10) Fraction of Inspired Oxygen needed at 44 weeks: transformed into linear spline terms at the knot of 0.737. (11) Peak Inspiratory Pressure (cmH2O) needed at 44 weeks (12) Positive end exploratory pressure (cm H2O) needed at 44 weeks.

Multiple imputation

To deal with the missing patterns using imputation algorithms, we first divided the population into two cohorts, `Cohort 36` and `Cohort 44`, based on if their hospital discharge age is before or after 44 week. Those with missing values in hospital discharge age are categorized into `Cohort 36` or `Cohort 44` based on if they have at least one valid 44 week's measurement. Then, separately for the two cohorts, we employed Multiple Imputation by Chained Equations (MICE) with 5 imputations to handle missing data in both cohorts. We then combined the two cohorts, and formed 5 imputation datasets, for all individuals. In this way, instead of imputaing altogether, we don't have to impute the 44 week's measurements for `Cohort 36`, and we are only using the `Cohort 44` to impute those who should have a record at the 44th week.

3.2 Model Building

Separately for each of the 5 imputation dataset but with the same seed, we randomly sample 70% of the individuals to form a training set. We first fitted a multivariate logistic regression in one of the training set to find the preliminary significant levels of the variables.

To enhance the robustness of our coefficient estimates and account for missing data, on each training set, we fitted LASSO logistic regressions and employed 10-fold cross-validation to determine the optimal penalty factor for each LASSO model. The penalty factor was selected to minimize the Mean Square Error across the 10 folds. Then we pooled the coefficients obtained from the 5 best models corresponding to the respective imputed training sets by averaging coefficients across the 5 best models. The pooled coefficients work as our final model.

In the Lasso logistic regressions, we modeled the log odds of experiencing tracheotomy or death as a linear combination of the specified covariates from the Data Processing session. Instead of directly adding the 44 week's measurements themselves, we created an indicator variable of "whether a patient has been discharged at 44 week" and include this and its interaction term with the measurements at the 44th week in the model. In this way, the coefficients we get for most of the covariates are estimated with the whole data, while coefficients pertaining to measurements at the 44th week are specifically derived from patients who have not been discharged at that time.

$$E(logit(P(Y_j))) = \beta_0 + \beta_j + \sum(\beta_k \cdot X_k) + \beta_{disc44} \cdot disc44 + \sum(\beta_m \cdot disc44 \cdot X_m^{44})$$

Where:

- Y_j is whether a patient has experienced tracheostomy or death.
- Y_j is the outcome variable `trach_or_death`.
- X_k and X_m^{44} are the variables measured before 44 week or at 44 week.
- `disc_44` indicates whether a patient is discharged after the 44 week.
- β_0 is the average intercept for all centers and β_j is the center-specific random effect.
- β_k are the coefficients for the variables measured before 44 week.
- β_m are the coefficients for the variables measured at 44 week.

Moreover, we treat the center effect as random effects on the intercept of the logistic regression model.

3.3 Model evaluation

The 30% left-out samples from the 5 imputation sets were combined into a long dataset to form the test set (each individuals have 5 entries). We then used the pooled coefficients and the test set to evaluate the model performance. We constructed the Receiver Operating Characteristic (ROC) curve depicting the trade-off between sensitivity and specificity across different probability thresholds, along with the AUC which quantify the model's discriminatory power. We chose the best cutoff defined by the frequently used summary index of marker accuracy Youden's Statistics $J = Sensitivity + Specificity - 1$ and the cutoff that maximizes J is considered the optimal cutoff (Youden 1950). Sensitivity and specificity under this cutoff is presented. We also assess the agreement between the predicted probabilities against observed outcomes using calibration plot.

4 Results

4.1 Characteristics of the data

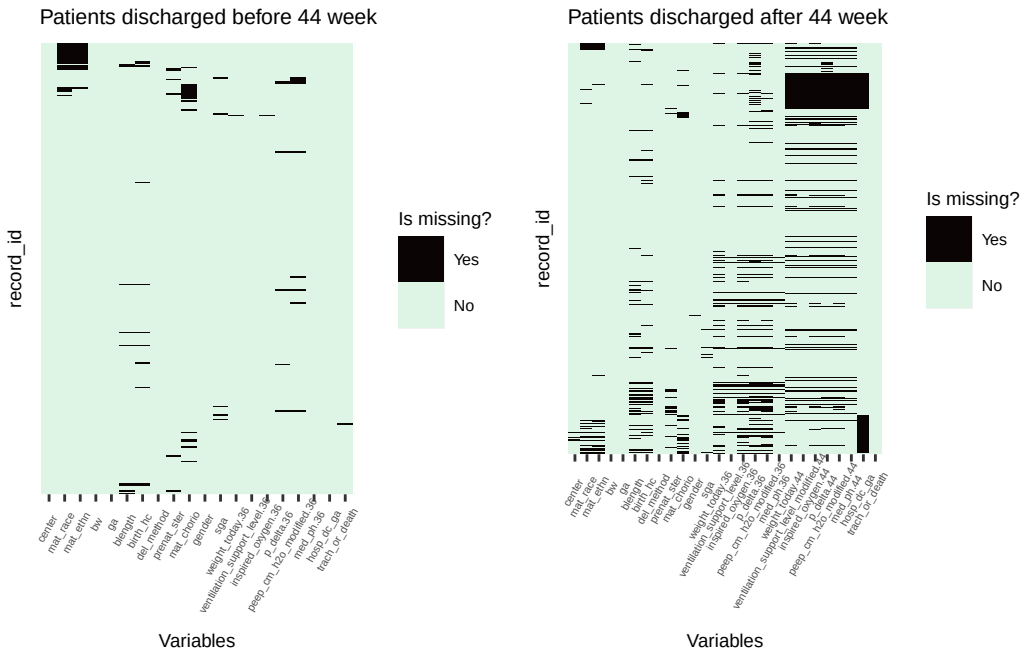

Figure 1: Missing patterns in original dataset

The data contains 993 individuals of 30 variables (after duplicated entries is removed for the individual with the recod id of 2000824, and center 21 is removed). For each individual, we have mother's demographic record, the infants' physical measurements, medical treatments and measurements at 36 week and at 44 week, and the outcome of Tracheotomy or Death. Figure 1 visualized the individual missingness. The data contains a high degree of missingness, which needs imputation, but will still have a large noise in our model and prediction. Specifically, measurements at the 44th week are highly incomplete due to the early discharge of the patients. Within patients discharged after 44 week, we still have ~90 over 685 missingness. Other variables of Complete Prenatal Steroids and Did the infant receive surfactant at any point in the first 72 hours? also have large proportions of missing values.

Table 1 provides a comprehensive summary of the variables stratified by the outcome of Tracheostomy or Death in a cohort of 993 individuals, along with the Pearson's Chi-squared test or Wilcoxon rank sum test p-values within each variables. Specifically, Medical Center 2 has a large proportion of patitient in our study, therefore, the effect of Medical Center 2 will have a high effect in model building. This missingness in the variables of Complete Prenatal Steroids

and Surfactant Received is notable. Patitiens only have the outcome of either Tracheostomy or Death.

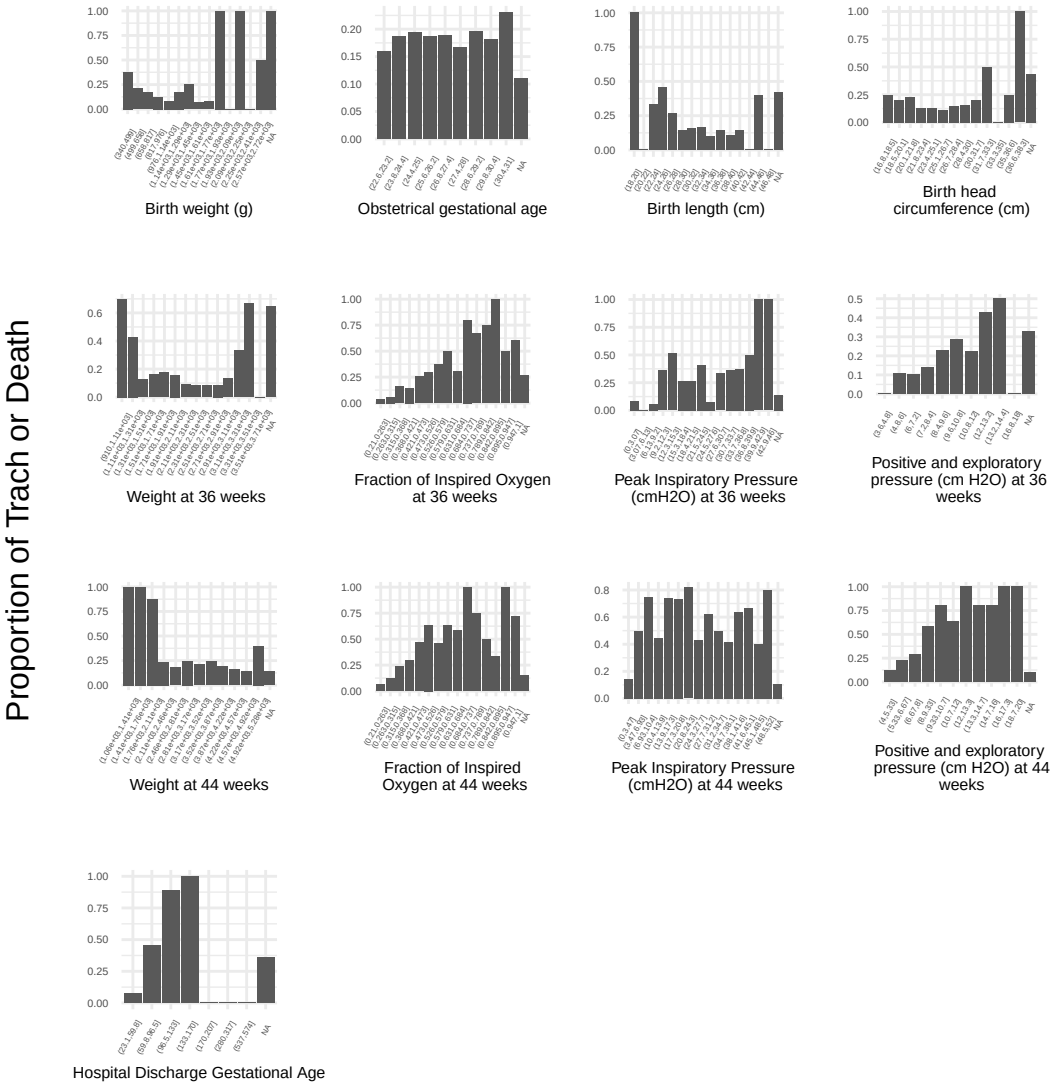

Figure 2: Proportion of Tracheotomy/Death in Continuous Variables

Figure 2 illustrates how the proportion of Tracheotomy/Death has been changing in response to variations in continuous variables. The continuous variables are binned, and within each bin, the proportion of Tracheotomy/Death is computed. As we see, the observed relationships are not linear. Rather, the propotions along the x-axis are often in a concave shape, i.e. high proportions of Tracheotomy/Death often occur at extreme values. Specifically, for the variables

measuring the Fraction of Inspired Oxygen at 36 weeks and Fraction of Inspired Oxygen at 44 weeks, the proportion of Tracheotomy/Death first ascends then declined, indicating a linear spline patterns of the outcomes to the variables.

4.2 Model Interpretation

Table 2 shows the pooled coeficients estimated with cross-validation lasso logistic regression in 5 imputated training set. The lasso coefficients being positive indicates a higher odds of getting Tracheotomy/Death comparing to the reference group or as the continous variable grow by one unit. In this sense, patients in center 12 with maternal race 2, having prenatal steroids, and being small, having a Ventilation support level 2 at 36 or 44, weeks, and having any Medication for Pulmonary Hypertension at 44 weeks, would have a higher odds of getting Tracheotomy/Death. For patients with a birth weight higher than 1610g, the odds of getting Tracheotomy/Death will increase by 12.835 times as the birth weight increased by 1 unit. Same interpretations can be drawn for the coefficients that have a negative value.

From the last 10 variables which include the indicator of whether the patient has been discharged at the week of 44 and its interaction with other 44 week's measurements, we see that the measurements at the 44 week are important predictors for predicting Tracheotomy/Death. Specifically, the variable of Ventilation support level 2 at 44 weeks and Medication for Pulmonary Hypertension at 44 weeks have coefficients of 1.807 and 1.006, which can change our predictions in a relatively large scale.

On the other hand, variables of Obstetrical gestational age, Birth head circumference (cm), Gender, Peak Inspiratory Pressure (cmH2O) at 36 weeks, Positive and exploratory pressure (cm H2O) at 36 weeks, Peak Inspiratory Pressure (cmH2O) needed at 44 weeks and Weight at 44 weeks are not useful in predicting Tracheotomy/Death, since their coefficients shrinked to very close to 0s in LASSO regression.

To use this model for prediction, we first identify whether a patient is discharged before 44 week. For patients discharged before 44 week, we only use the coefficients and variables above `disc_44` to estimate the probibility. For patients discharged after 44 week, we utilize their measurements at the 44 week. Thus for these patients we use all the variables in Table 2 for estimation.

4.3 Assessment of Model Fit

Figure 3 illustrate the discrimination and calibration of the model in the validation set. The ROC is above the diagonal line and very close to a rectangle shape, with an AUC very close to 1. This result suggest that under appropriate sensitivity-specificity trade-off, the model can discriminate positive/negative outcome successfully. When the threshhold is set to be 0.137 (while p>0.137, the patients are predicted to have tracheotomy or death; p<=0.137, the

Table 1: Estimated coefficients

covariates	Lasso_coefficient	covariates	Lasso_coefficient
(Intercept)	-2.510	blength_2	0.000
mat_race1	-0.005	bw_1	0.143
mat_race2	0.031	bw_2	0.000
mat_ethn2	0.000	inspired_oxygen.36_1	2.155
ga	0.000	inspired_oxygen.36_2	2.016
birth_hc	0.000	weight_today.36_1	0.000
del_method2	0.000	weight_today.36_2	-1.138
prenat_sterYes	0.000	weight_today.36_3	0.000
mat_chorioYes	0.000	disc_44	-1.657
genderMale	0.000	disc_44:vent_spt_lev.441	0.000
sgaSGA	0.145	disc_44:vent_spt_lev.442	1.918
ventilation_support_level.361	-0.351	disc_44:p_delta.44	-0.011
ventilation_support_level.362	0.788	disc_44:peep_cm_h2o_modified.44	0.062
p_delta.36	-0.003	disc_44:med_ph.441	1.087
peep_cm_h2o_modified.36	0.012	disc_44:inspired_oxygen.44_1	1.379
med_ph.361	-0.083	disc_44:inspired_oxygen.44_2	-1.161
blength_1	-0.205	disc_44:weight_today.44_1	0.982
		disc_44:weight_today.44_2	0.000

patients are predicted to have no tracheotomy or death), the sensitivity and specificity of this model are 0.816 and 0.917.

The calibration plot shows that most patients (60%) are predicted to be at a low risk (P<10%) of tracheotomy and death. When the predicted probabilities are high, the the predicted values tend to underestimate the original outcome.

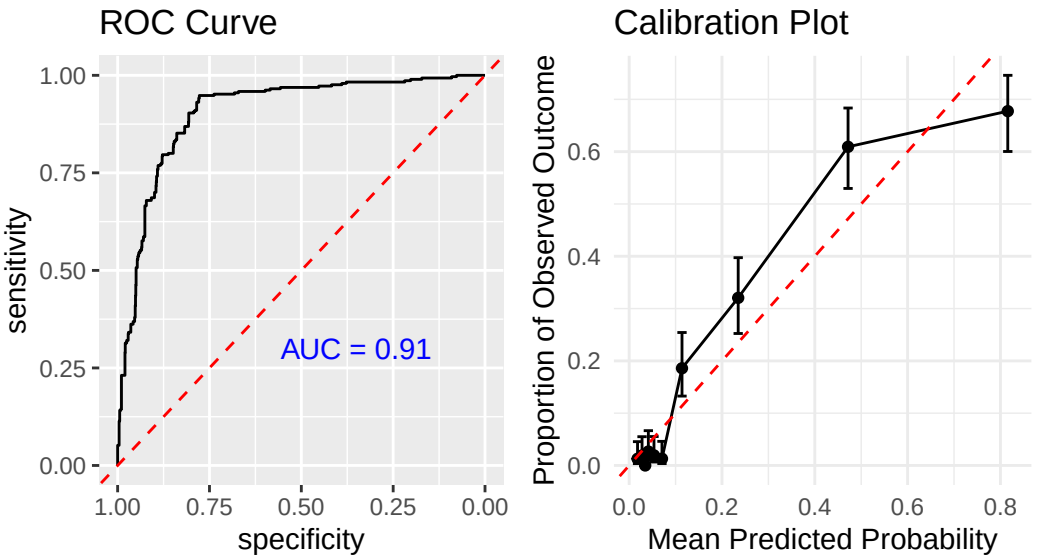

Figure 3: Model evaluation

5 Disscusion

In this analysis, we performed multiple imputations and lasso logistic regressions with cross-validation to build linear models to predict the single outcome of Tracheotomy and Death. Specifically, we conducted the imputation separately for individuals discharged before or after 44 week. We combine the imputation datasets and splited them into training set and validation set. We added an indicator term for those discharge before and after 44 week and form interaction terms for selected variables. We pooled the coeficients from the models with different imputation datasets for validation.

Our analysis have certain strength. The seperate-imputation and the model with the specific interaction terms can treat the patients discharged before or after 44 week differently, but can still incorporate both indiciduals in estimating coeficients not related to the 44 week. Thus, this method ensures a comprehensive evaluation of the impact of various covariates on the outcome, with due consideration for the discharge status at the 44th week.

On the other hand, pooling coefficients from the model estimated with different imputation datasets ensures stability and reliability in variable selection.

However, in this analysis, we didn't explore the multilinearity patterns in the covariates and thus didn't consider to include other interaction terms. Also, we didn't handle the outliers at certain covariates and the multip-centers effect.

6 Reference

Youden, William J. 1950. "Index for Rating Diagnostic Tests." *Cancer* 3 (1): 32–35.

7 Code Appendix

```
# Check and install packages
packages_to_check <- c("gtsummary", "gt", "tidyverse", "kableExtra", "mice",
                       "viridis", "knitr", "gridExtra",
                       "GGally", "pROC", "glmnet", "splines", "glmmLasso", "binom")
# Check if each package is installed and load it if available;
# otherwise, install and load it
for (pkg in packages_to_check) {
  if (!require(pkg, character.only = TRUE, quietly = TRUE)) {
    # If the package is not installed, install it
    install.packages(pkg)

    # Now, load the package
    library(pkg, character.only = TRUE)
  }
}

knitr::opts_chunk$set(warning = FALSE, message = FALSE,
                      echo = FALSE, fig.align = "center")
df <- read.csv("project2.csv")
# Data pre-processing
## Remove duplicated entries
df <- df[!duplicated(df$record_id), ]
df <- df[!df$center==21 | is.na(df$center),]
#df$com_prenat_ster[df$prenat_ster=="Yes"] <- "Yes"
#df <- select(df, -com_prenat_ster)
# Convert character variables to factor variables (for mice)
for (col in names(df)) {
  if (is.character(df[[col]])) {
    df[[col]] <- as.factor(df[[col]])
  }
}
#convert factor variables coded as numeric variables to the right format
fac_vars <- c("center","mat_race", "mat_ethn", "del_method",
              "ventilation_support_level.36", "med_ph.36",
              "ventilation_support_level_modified.44","med_ph.44")
for (var in fac_vars) {
  df[[var]] <- as.factor(df[[var]])
}
```

```r
# EDA for any_surf (which contain so many missing values)
# df %>%
#   mutate(missing_surf = ifelse(is.na(any_surf), 1, 0),
#          center = as.factor(center)) %>%
#   select(-c(record_id, any_surf)) %>%
#   tbl_summary(by = missing_surf,
#               percent = "row") %>%
#   add_p() %>%
#   modify_spanning_header(all_stat_cols() ~ "**If missing any_surf**")


## Separate into two dataset by discharge time
variables.44 <- names(df)[grep("\\.44$", names(df))]
df_clean <-
  df %>%
  mutate(trach_or_death = ifelse(Trach == 1 | Death == "Yes", 1, 0),
         if_any_record_44 = !is.na(variables.44[1]) |
              !is.na(variables.44[2])|
              !is.na(variables.44[3]) | !is.na(variables.44[4])|
              !is.na(variables.44[5])| !is.na(variables.44[6]),
         cohort = case_when(!if_any_record_44 & is.na(hosp_dc_ga) ~ "cohort 36",
                            hosp_dc_ga>=0 & hosp_dc_ga < 44 ~ "cohort 36",
                            hosp_dc_ga>= 44 ~ "cohort 44",
                            if_any_record_44 & is.na(hosp_dc_ga) ~"cohort 44")) %>%
  select(-c(any_surf, Trach, Death, if_any_record_44, com_prenat_ster)) %>%
  filter(center!=21 | is.na(center))

df_44 <-
df_clean %>%
  filter(cohort == "cohort 44") %>%
  select(-cohort)

df_36 <-
  df_clean %>%
  filter(cohort == "cohort 36") %>%
  select(-c(cohort, variables.44))
set.seed(56)
test_num_36 <- sample(c(TRUE, FALSE), size = nrow(df_36),
                      replace = TRUE, prob = c(0.3, 0.7))
test_num_44 <- sample(c(TRUE, FALSE), size = nrow(df_44),
                      replace = TRUE, prob = c(0.3, 0.7))
```

```r
### Caution!! It would take > 0.5h to run mice
# ## Mice
# df_36_mice_out <- mice(df_36, m = 5, ignore = test_num_36,
#                        printFlag = FALSE, seed = 222)
#
#
# df_44_mice_out <- mice(df_44, m = 5, ignore = test_num_44,
#                        printFlag = FALSE, seed = 222,
#                        nnet.MaxNWts = 4300) # nnet.MaxNWts: allow it to run longer
# saveRDS(df_36_mice_out, file = "df_36_mice_out.RDS")
# saveRDS(df_44_mice_out, file = "df_44_mice_out.RDS")
#Load in the mice objects
df_36_mice_out <- readRDS("df_36_mice_out.RDS")
df_44_mice_out <- readRDS("df_44_mice_out.RDS")
# Storing complete training set and testing sets
df_36_train <- vector("list",5)
for (i in 1:5){
df_36_train[[i]] <- mice::complete(filter(df_36_mice_out, !test_num_36),i)
}
df_36_test <- mice::complete(filter(df_36_mice_out, test_num_36), action="stacked")
df_44_train <- vector("list",5)
for (i in 1:5){
df_44_train[[i]] <- mice::complete(filter(df_44_mice_out, !test_num_44),i)
}
df_44_test <- mice::complete(filter(df_44_mice_out, test_num_44), action="stacked")
## Combine the imputations from the two cohorts
df_train <- vector("list", 5)
for (i in 1:5) {
#filling zeros for NAs in Cohort 36 (these values is not used in the glm)
for (j in variables.44) {
df_36_train[[i]][[j]] <- 0
}
# add indicator variable for the two cohorts
df_36_train[[i]]$disc_44 <- 0
df_44_train[[i]]$disc_44 <- 1
df_train[[i]] <- rbind(df_36_train[[i]], df_44_train[[i]]) %>%
  select(-c(record_id))
}
for (j in variables.44) {
df_36_test[[j]] <- 0
}
```

```r
df_36_test$disc_44 <- 0
df_44_test$disc_44 <- 1
df_test <- rbind(df_36_test, df_44_test)
## Missing value heatmap
missing_heatmap <- function(df, main=""){
  df_missing <-
  apply(df, 2, function(x) ifelse(is.na(x), "Yes", "No")) %>%
  as.data.frame()

  plot <-
df_missing %>%
  mutate(record_id = as.factor(df$record_id)) %>%
  pivot_longer(cols = names(df)[2:length(names(df))],
               names_to = "Variables",
               values_to = "is_missing") %>%
  mutate(is_missing = factor(is_missing, levels= c("Yes", "No")))%>%
  mutate(Variables = factor(Variables, levels = names(df)[2:length(names(df))])) %>%
  ggplot(aes(Variables, record_id, fill=is_missing, )) +
  geom_tile() +
  scale_fill_viridis(discrete = TRUE, option = "G") +
  theme(text = element_text(size = 7),
        axis.text.x = element_text(size = 4,
                                   angle = 60, hjust = 0.8, vjust = .9),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank())+
  labs(fill = "Is missing?", title = main)

  return(plot)
}
#missing_heatmap(df)
missing_36 <- missing_heatmap(df_36, main = "Patients discharged before 44 week")
missing_44 <- missing_heatmap(df_44, main = "Patients discharged after 44 week")
grid.arrange(missing_36, missing_44, ncol=2)
## Summarized table
theme_gtsummary_compact(font_size = 4)
df %>%
  mutate(center = as.factor(center),
         trach_or_death = ifelse(Trach == 1 | Death == "Yes", "Yes", "No")) %>%
  select(-record_id, Trach, Death) %>%
  mutate(mat_ethn = case_when(mat_ethn== 1 ~ "Hispanic or Latino",
                              mat_ethn == 2 ~ "Not Hispanic or Latino",
```

```
                              is.na(mat_ethn) ~ NA),
         del_method = case_when(del_method == 1 ~ "Vaginal delivery",
                                del_method == 2 ~ "Cesarean section",
                                is.na(del_method) ~ NA),
         ventilation_support_level.36 = case_when(ventilation_support_level.36
== 0 ~ "No respiratory support or supplemental oxygen",
ventilation_support_level.36 == 1 ~ "Non-invasive positive pressure",
ventilation_support_level.36 == 2 ~ "Invasive positive pressure",
is.na(ventilation_support_level.36) ~ NA),
          ventilation_support_level_modified.44 = case_when(ventilation_support_level_modi
== 0 ~ "No respiratory support or supplemental oxygen",
ventilation_support_level_modified.44 == 1 ~ "Non-invasive positive pressure",
ventilation_support_level_modified.44 == 2 ~ "Invasive positive pressure",
is.na(ventilation_support_level_modified.44) ~ NA),
         Trach = case_when(Trach == 0 ~ "No",
                           Trach == 1 ~ "Yes"),
          med_ph.36 = case_when(med_ph.36 == 0 ~ "No",
                                med_ph.36 == 1 ~ "Yes",
                                is.na(med_ph.36) ~ NA),
         med_ph.44 = case_when(med_ph.44 == 0 ~ "No",
                               med_ph.44 == 1 ~ "Yes",
                               is.na(med_ph.44) ~ NA)) %>%

  tbl_summary(type = list(prenat_ster ~ 'categorical',
                          com_prenat_ster ~ 'categorical',
                          mat_chorio ~ 'categorical',
                          any_surf ~ 'categorical',
                          med_ph.36 ~ 'categorical',
                          med_ph.44 ~ 'categorical',
                          Trach ~ 'categorical',
                          Death ~ 'categorical'),
              by = trach_or_death,
              percent = "row",
              missing_text = "missing",
              statistic = list(all_continuous() ~ "{mean} ({sd})"),
              label = list(`center` = "Medical Center",
                           `mat_race` = "Maternal Race",
                           `mat_ethn` = "Maternal Ethnicity",
                           `bw` = "Birth Weight (g)",
                           `ga` = "Obstetrical Gestational Age",
                           `blength` = "Birth Length (cm)",
```

```
                          `birth_hc` = "Birth Head Circumference (cm)",
                          `del_method` = "Delivery Method",
                          `prenat_ster` = "Prenatal Corticosteroids",
                          `com_prenat_ster` = "Complete Prenatal Steroids",
                          `mat_chorio` = "Maternal Chorioamnionitis",
                          `gender` = "Gender",
                          `sga` = "Small for Gestational Age",
                          `any_surf` = "Surfactant Received",
                          `weight_today.36` = "Weight at 36 Weeks",
                          `ventilation_support_level_modified.36` = "Ventilation Support
                          `inspired_oxygen.36` = "Inspired Oxygen at 36 Weeks",
                          `p_delta.36` = "Peak Inspiratory Pressure at 36 Weeks",
                          `peep_cm_h2o_modified.36` = "PEEP* at 36 Weeks",
                          `med_ph.36` = "Medication for PH* at 36 Weeks",
                          `weight_today.44` = "Weight at 44 Weeks",
                          `ventilation_support_level_modified.44` = "Ventilation Support
                          `inspired_oxygen.44` = "Inspired Oxygen at 44 Weeks",
                          `p_delta.44` = "Peak Pressure at 44 Weeks",
                          `peep_cm_h2o_modified.44` = "PEEP at 44 Weeks",
                          `med_ph.44` = "Medication for PH at 44 Weeks",
                          `hosp_dc_ga` = "Hospital Discharge Gestational Age",
                          `Trach` = "Tracheostomy")) %>%
  bold_labels() %>%
  add_p() %>%
  add_overall() %>%
  modify_header(label = "**Variables**") %>%
  modify_spanning_header(all_stat_cols() ~ "**Tracheostomy or Death**") %>%
  modify_footnote(all_stat_cols() ~ "PEEP: Positive end exploratory pressure;
                  PH: Pulmonary Hypertension")%>%
  as_kable_extra(booktabs = TRUE,
                 format = "latex",
                longtable = TRUE,
                  caption = "Summary of variables by outcome")  %>%
  kable_styling(latex_options = "hold_position", font_size = 4)
linearty_check <- function(var, label){
  min <- min(df_clean[[var]], na.rm = TRUE)
  max <- max(df_clean[[var]],na.rm = TRUE)
  bin <- (max - min)/15
  plot <-
  ggplot(df_clean, aes(x = cut(df_clean[[var]], breaks = seq(min, max, by = bin)),
                       y = trach_or_death)) +
```

```
  stat_summary(fun = "mean", geom = "bar") +
  labs(title = "",
       x = label,
       y = "") +
  theme_minimal()+
  theme(axis.text.x = element_text(size = 3,
                                   angle = 60, hjust = 0.8, vjust = .9),
        axis.text.y = element_text(size = 4),
        axis.title.x = element_text(size = 6))
  return(plot)
}
bw <- linearty_check(var = "bw", "Birth weight (g)")
ga <- linearty_check(var = "ga", "Obstetrical gestational age")
bl <- linearty_check(var = "blength", "Birth length (cm)")
bhc <- linearty_check(var = "birth_hc", "Birth head\n circumference (cm)")
w_36 <- linearty_check(var = "weight_today.36", "Weight at 36 weeks")
iox_36 <- linearty_check(var = "inspired_oxygen.36",
                "Fraction of Inspired Oxygen\n at 36 weeks")
pd_36 <- linearty_check(var = "p_delta.36",
                "Peak Inspiratory Pressure\n (cmH2O) at 36 weeks")
peep_36 <- linearty_check(var = "peep_cm_h2o_modified.36",
                "Positive and exploratory\n pressure (cm H2O) at 36\n weeks")

w_44 <- linearty_check(var = "weight_today.44", "Weight at 44 weeks")
iox_44 <- linearty_check(var = "inspired_oxygen.44",
                "Fraction of Inspired\n Oxygen at 44 weeks")
pd_44 <- linearty_check(var = "p_delta.44",
                "Peak Inspiratory Pressure\n (cmH2O) at 44 weeks")
peep_44 <- linearty_check(var = "peep_cm_h2o_modified.44",
                "Positive and exploratory\n pressure (cm H2O) at 44\n weeks")

hosp <- linearty_check(var = "hosp_dc_ga", "Hospital Discharge Gestational Age")

print(grid.arrange(bw, ga, bl, bhc, w_36, iox_36, pd_36,
                   peep_36,w_44, iox_44, pd_44, peep_44, hosp,
                   ncol=4,
                   left = "Proportion of Trach or Death"))

create_and_replace_spline <- function(imputed_data) {
  # Create spline term
  bw_spline <- bs(imputed_data$bw, knots = 1610,
```

```
                                degree = 1, intercept = FALSE)
  blength_spline <- bs(imputed_data$blength, knots = 40,
                                degree = 1, intercept = FALSE)
  inspired_oxygen_36_spline <- bs(imputed_data$inspired_oxygen.36, knots = .789,
                                degree = 1, intercept = FALSE)
  inspired_oxygen_44_spline <- bs(imputed_data$inspired_oxygen.44, knots = .737,
                                degree = 1, intercept = FALSE)
  weight_today_36_spline <- bs(imputed_data$weight_today.36, knots = c(1510,2710),
                                degree = 1, intercept = FALSE)
    weight_today_44_spline <- bs(imputed_data$weight_today.44, knots = 2110,
                                degree = 1, intercept = FALSE)

  # Replace original variable
  imputed_data <- imputed_data %>% select(-c(bw,blength,
                                             inspired_oxygen.36,
                                             inspired_oxygen.44,
                                             weight_today.36,
                                             weight_today.44))
  imputed_data$blength_1 <- blength_spline[,1]
  imputed_data$blength_2 <- blength_spline[,2]

  imputed_data$bw_1 <- bw_spline[,1]
  imputed_data$bw_2 <- bw_spline[,2]

  imputed_data$inspired_oxygen.36_1 <- inspired_oxygen_36_spline[,1]
  imputed_data$inspired_oxygen.36_2 <- inspired_oxygen_36_spline[,2]

  imputed_data$inspired_oxygen.44_1 <- inspired_oxygen_44_spline[,1]
  imputed_data$inspired_oxygen.44_2 <- inspired_oxygen_44_spline[,2]

  imputed_data$weight_today.36_1 <- weight_today_36_spline[,1]
  imputed_data$weight_today.36_2 <- weight_today_36_spline[,2]
  imputed_data$weight_today.36_3 <- weight_today_36_spline[,3]

  imputed_data$weight_today.44_1 <- weight_today_44_spline[,1]
  imputed_data$weight_today.44_2 <- weight_today_44_spline[,2]
  return(imputed_data)
}

df_train_with_spline <- lapply(df_train, create_and_replace_spline)
df_test_with_spline <- create_and_replace_spline(df_test)
```

```r
variables_other <- names(df_train_with_spline[[1]])[c(1:15, 22:27, 30:32)]
variables.44 <- names(df_train_with_spline[[1]])[c(17:20, 28:29, 33:34)]

#########################################################
#### Lasso ####
#########################################################
lasso <- function(df) {
  #' Runs 10-fold CV for lasso and returns corresponding coefficients
  #' @param df, data set
  #' @return coef, coefficients for minimum cv error

  # Matrix form for ordered variables
  x.ord <- model.matrix(model_formula, data = df)[,-1]#Dropping intercept term since lasso
  y.ord <- df$trach_or_death

  # Generate folds
  k <- 10
  set.seed(1) # consistent seeds between imputed data sets
  folds <- sample(1:k, nrow(df), replace=TRUE)

  # Lasso model
  lasso_mod_cv <- cv.glmnet(x.ord, y.ord, nfolds = 10, foldid = folds,
                            alpha = 1, family = "binomial")
  # lasso_mod <- cv.glmnet(x.ord, y.ord, nfolds = 10,
  #                        lambda =  lasso_mod_cv$lambda.min,
  #                        alpha = 1, family = "binomial")

  # Get coefficients
  coef <- coef(lasso_mod_cv, s = "lambda.min")
  return(coef)
}
# Find average lasso coefficients over imputed datasets
pooling_lasso <- function(traing_data){
  lasso_coef1 <- lasso(traing_data[[1]])
  lasso_coef2 <- lasso(traing_data[[2]])
  lasso_coef3 <- lasso(traing_data[[3]])
  lasso_coef4 <- lasso(traing_data[[4]])
  lasso_coef5 <- lasso(traing_data[[5]])
  lasso_coef <- cbind(lasso_coef1, lasso_coef2, lasso_coef3,
                      lasso_coef4, lasso_coef5)
  avg_coefs_lasso <- apply(lasso_coef, 1, mean)
```

```r
  return(avg_coefs_lasso)
}
set.seed(6)
coef_pred <- pooling_lasso(df_train_with_spline)
#########################################################
#### GLMM Lasso ####
#########################################################
glmm_lasso <- function(df, num_lambda = 10) {
  #' Runs 10-fold CV for GLMM lasso and returns corresponding coefficients
  #' @param df, data set
  #' @param random_effect, the name of the random effect variable (e.g., center)
  #' @return coef, coefficients for minimum cv error

  # Prepare the formula
  formula <- as.formula(paste(
"trach_or_death ~ ",
paste0(variables_other[!variables_other %in% c("inspired_oxygen.36", "inspired_oxygen.44",
" + disc_44 + disc_44 : (",
paste0(variables.44, collapse = " + "), ")"
))

  # Generate folds
  k <- 10
  set.seed(1) # consistent seeds between imputed data sets
  folds <- sample(1:k, nrow(df), replace=TRUE)

  # Initialize matrix for coefficients and vector for deviances
  Coeff_ma <- NULL
  Devianz_ma <- numeric(num_lambda)

  # Define a sequence of lambda values
  lambda <- 10^seq(-3, 5, length = num_lambda)

  # Loop over lambda values
  for (j in seq_along(lambda)) {
    # Initialize a vector to store deviance for each fold
    deviance_folds <- numeric(k)

    # Cross-validation loop
    for (i in 1:k) {
      # Split the data
```

```
      train_data <- df[folds != i, ]
      test_data <- df[folds == i, ]


      # Fit the model on training data
      glm1 <- glmmLasso(formula, data = train_data, rnd = list(center=~1), family = binomi

      # Initialize Coeff_ma in the first iteration
      if (is.null(Coeff_ma)) {
        Coeff_ma <- matrix(nrow = num_lambda, ncol = length(coef(glm1)))
      }

      # Store coefficients
      Coeff_ma[j, ] <- coef(glm1)

      # Predict on test data and calculate deviance
      y.hat <- predict(glm1, test_data)
      deviance_folds[i] <-
        sum(with(test_data,
                 binomial()$dev.resids(trach_or_death,
                                       y.hat,
                                       wt = rep(1, length(y.hat)))))
    }

    # Store average deviance for this lambda
    Devianz_ma[j] <- mean(deviance_folds)
  }

  # Find the lambda with the minimum average deviance
  best_lambda <- lambda[which.min(Devianz_ma)]
  best_model <- glmmLasso(formula, data = df, rnd = list(center=~1),
                          family = binomial(),lambda = best_lambda,
                          switch.NR = TRUE, final.re = TRUE)


  # Return the best model and lambda
  return(coefficients(best_model))
}
# Find average GLMM lasso coefficients over imputed datasets
pooling_glmm_lasso <- function(training_data){
  coefs <- lapply(training_data, function(df) glmm_lasso(df))
```

```
  avg_coefs_glmm_lasso <- rowMeans(do.call(cbind, coefs))
  return(avg_coefs_glmm_lasso)
}

#set.seed(6)
#coef_pred_glmm <- pooling_glmm_lasso(df_train_with_spline)
#saveRDS(coef_pred_glmm, file = "coef_final.RDS")
coef_pred_glmm <- read_rds("coef_final.RDS")
# Create a data frame for odds ratios and their CIs
# Extract odds ratios and their confidence intervals
result_table <-
data.frame(
  covariates = names(coef_pred_glmm),
  Lasso_coefficient = round(coef_pred_glmm, 3)
  ) %>%
  mutate(covariates = ifelse(covariates == "disc_44:ventilation_support_level_modified.441
                             "disc_44:vent_spt_lev.441",
                             covariates),
         covariates = ifelse(covariates == "disc_44:ventilation_support_level_modified.442
                             "disc_44:vent_spt_lev.442",
                             covariates)) %>% `rownames<-`(NULL)
table1 <- result_table[1:17, ] %>%
  knitr::kable(format = "latex", booktabs = TRUE) %>%
  kable_styling(bootstrap_options = "condensed", font_size = 10, full_width = F) %>%
  column_spec(1, width = "3cm")

table2 <- result_table[18:35, ] %>% `rownames<-`(NULL) %>%
  knitr::kable(format = "latex", booktabs = TRUE) %>%
  kable_styling(bootstrap_options = "condensed", font_size = 10, full_width = F) %>%
  column_spec(1, width = "5.5cm")

list(result_table[1:17, ], result_table[18:35, ] %>% `rownames<-`(NULL)) %>%
  knitr::kable(format = "latex", booktabs = TRUE,
               caption = "Estimated coefficients") %>%
  kable_styling(bootstrap_options = "condensed", font_size = 6, full_width = F)
# Get predicted values with pooled coefs
predict_lasso <- function(test_data, coef){
  formula <- as.formula(paste(
"trach_or_death ~ ",
paste0(variables_other[!variables_other %in% c("inspired_oxygen.36", "inspired_oxygen.44",
" + disc_44 + disc_44 : (",
```

```
paste0(variables.44, collapse = " + "), ")"
))

  x_vars <- model.matrix(formula, test_data)
  #test_data$lasso <- x_vars %*% coef
  #mod_lasso <- glm(trach_or_death~lasso, data = test_data, family = "binomial")
  # predict_probs_lasso <- predict(mod_lasso, type="response")
  return(x_vars)
}

df_test_with_spline$predict_probs_lasso <-
  predict(glm(df_test_with_spline$trach_or_death ~ predict_lasso(df_test_with_spline, coef


#ROC-AUC
roc_curve_lasso <- roc(response = df_test_with_spline$trach_or_death,
                       predictor = df_test_with_spline$predict_probs_lasso,
                       levels = c(0,1), direction= "<")
auc_value <- auc(roc_curve_lasso)

roc_plot <- ggroc(roc_curve_lasso) +
  ggtitle("ROC Curve")+
  geom_abline(intercept = 1, slope = 1, linetype = "dashed", color = "red") +
  annotate("text", x = 0.35, y = 0.3,
           label = paste("AUC =", round(auc_value, 2)), color = "blue")
  theme_minimal()

# Spec and sens at the optimal cutoff
optim_threshold = coords(roc_curve_lasso, "best", ret = "threshold")$threshold
predicted_class <- ifelse(df_test_with_spline$predict_probs_lasso >= optim_threshold, 1, 0
confusion_matrix <- table(Actual = df_test_with_spline$trach_or_death,
                          Predicted = predicted_class)

sens  <- confusion_matrix[1, 1] / sum(confusion_matrix[1, ])
spec  <- confusion_matrix[2, 2] / sum(confusion_matrix[2, ])

# Calibration
num_bins <- 10
df_binned <- df_test_with_spline %>%
  mutate(predicted_prob_bin = cut(predict_probs_lasso,
                                  breaks = quantile(predict_probs_lasso,
```

```
                                                    probs = seq(0, 1, length.out = num_bin
  group_by(predicted_prob_bin) %>%
  summarise(
    mean_predicted_prob = mean(predict_probs_lasso),
    observed_outcome = mean(trach_or_death),
    n = n(),
    conf_low = binom.confint(sum(trach_or_death), n, methods = "wilson")$lower,
    conf_high = binom.confint(sum(trach_or_death), n, methods = "wilson")$upper
  )


cal_plot <- ggplot(df_binned, aes(x = mean_predicted_prob, y = observed_outcome)) +
  geom_point() +
  geom_line() +
  geom_errorbar(aes(ymin = conf_low, ymax = conf_high), width = 0.02) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Calibration Plot",
       x = "Mean Predicted Probability",
       y = "Proportion of Observed Outcome") +
  theme_minimal()

print(grid.arrange(roc_plot, cal_plot, ncol=2))
```