library(corrplot)

library(glmnet)

library(caret)

library(CBCgrps)

library(nortest)

library(tidyverse)

library(ggpubr)

library(rms)

library(pROC)

library(viridis)

library(mice)

summary(data)

imputed\_data <- mice(data, m = 5, method = 'pmm', maxit = 5, seed = 500)

summary(imputed\_data)

completed\_data1 <- complete(imputed\_data, 1)

completed\_data2 <- complete(imputed\_data, 2)

completed\_data3 <- complete(imputed\_data, 3)

completed\_data4 <- complete(imputed\_data, 4)

completed\_data5 <- complete(imputed\_data, 5)

min\_max\_scale = function(x){

(x-min(x))/(max(x)-min(x))

}

library(dplyr)

library(glmnet)

data2 = completed\_data1%>%

mutate\_if(.predicate = is.numeric,

.funs = min\_max\_scale)%>%

as.data.frame()

set.seed(123) #random number generator

x <- data.matrix(data2[, -1])

y <- data2[, 1]

y<-as.numeric(unlist(y))

lasso <- glmnet(x, y, family = "binomial", nlambda = 1000, alpha = 1)

print(lasso)

lasso.cv = cv.glmnet(x, y,alpha = 1,nfolds =20,family="binomial")

plot(lasso.cv)

lasso.cv$lambda.min #minimum

lasso.cv$lambda.1se #one standard error away

coef(lasso.cv, s = "lambda.1se")

lassov <- c("diagnosis","singleton\_status","assisted\_ventilation","birth\_weight","gestational\_weeks","initial\_feeding\_method","birth\_length","apgar\_1min")

mydata<-completed\_data1

attach(mydata)

dd <- datadist(mydata)

options(datadist = 'dd')

fit0 <- lrm(diagnosis ~ singleton\_status + assisted\_ventilation + birth\_weight + gestational\_weeks + initial\_feeding\_method + birth\_length + apgar\_1min, data = mydata, x = T, y = T)

gd<-predict(fit0,newdata = completed\_data1,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

library(pROC)

roc.list<-roc(completed\_data1$diagnosis,gd)#low

roc.list

g.list<- ggroc(roc.list,alpha = 1,size = 0.8,legacy.axes = TRUE,color="red")

g.list+theme\_classic()+ggtitle("data")+annotate(geom="segment",x = 0,y = 0,xend =1,yend = 1)

gd <- predict(fit0, newdata = completed\_data1, type = "fitted")

roc.list <- roc(completed\_data1$diagnosis, gd)

best\_coords <- coords(roc.list,

x = "best",

best.method = "youden",

ret = c("threshold", "sensitivity", "specificity", "accuracy", "ppv", "npv"))

print(round(best\_coords, 3))

min\_max\_scale = function(x){

(x-min(x))/(max(x)-min(x))

}

library(dplyr)

library(glmnet)

data2 = completed\_data2%>%

mutate\_if(.predicate = is.numeric,

.funs = min\_max\_scale)%>%

as.data.frame()

set.seed(123) #random number generator

x <- data.matrix(data2[, -1])

y <- data2[, 1]

y<-as.numeric(unlist(y))

lasso <- glmnet(x, y, family = "binomial", nlambda = 1000, alpha = 1)

lasso.cv = cv.glmnet(x, y,alpha = 1,nfolds =20,family="binomial")

plot(lasso.cv)

lasso.cv$lambda.min #minimum

lasso.cv$lambda.1se #one standard error away

coef(lasso.cv, s = "lambda.1se")

lassov <- c("diagnosis","singleton\_status","assisted\_ventilation","birth\_weight","gestational\_weeks","initial\_feeding\_method","birth\_length","apgar\_1min","admission\_age\_days")

mydata<-completed\_data2

attach(mydata)

dd <- datadist(mydata)

options(datadist = 'dd')

fit0 <- lrm(diagnosis ~ singleton\_status + assisted\_ventilation + birth\_weight + gestational\_weeks + initial\_feeding\_method + birth\_length + apgar\_1min + admission\_age\_days, data = mydata, x = T, y = T)

gd<-predict(fit0,newdata = completed\_data2,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

roc.list<-roc(completed\_data2$diagnosis,gd)#low

roc.list

g.list<- ggroc(roc.list,alpha = 1,size = 0.8,legacy.axes = TRUE,color="red")

g.list+theme\_classic()+ggtitle("data")+annotate(geom="segment",x = 0,y = 0,xend =1,yend = 1)

gd <- predict(fit0, newdata = completed\_data2, type = "fitted")

roc.list <- roc(completed\_data2$diagnosis, gd)

best\_coords <- coords(roc.list,

x = "best",

best.method = "youden",

ret = c("threshold", "sensitivity", "specificity", "accuracy", "ppv", "npv"))

print(round(best\_coords, 3))

min\_max\_scale = function(x){

(x-min(x))/(max(x)-min(x))

}

library(dplyr)

library(glmnet)

data2 = completed\_data3%>%

mutate\_if(.predicate = is.numeric,

.funs = min\_max\_scale)%>%

as.data.frame()

set.seed(123) #random number generator

x <- data.matrix(data2[, -1])

y <- data2[, 1]

y<-as.numeric(unlist(y))

lasso <- glmnet(x, y, family = "binomial", nlambda = 1000, alpha = 1)

lasso.cv = cv.glmnet(x, y,alpha = 1,nfolds =20,family="binomial")

plot(lasso.cv)

lasso.cv$lambda.min #minimum

lasso.cv$lambda.1se #one standard error away

coef(lasso.cv, s = "lambda.1se")

lassov <- c("diagnosis","singleton\_status","assisted\_ventilation","birth\_weight","gestational\_weeks","initial\_feeding\_method","birth\_length","apgar\_1min","admission\_age\_days")

mydata<-completed\_data3

attach(mydata)

dd <- datadist(mydata)

options(datadist = 'dd')

fit0 <- lrm(diagnosis ~ singleton\_status + assisted\_ventilation + birth\_weight + gestational\_weeks + initial\_feeding\_method + birth\_length + apgar\_1min + admission\_age\_days, data = mydata, x = T, y = T)

gd<-predict(fit0,newdata = completed\_data3,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

roc.list<-roc(completed\_data3$diagnosis,gd)#low

roc.list

gd <- predict(fit0, newdata = completed\_data3, type = "fitted")

roc.list <- roc(completed\_data3$diagnosis, gd)

best\_coords <- coords(roc.list,

x = "best",

best.method = "youden",

ret = c("threshold", "sensitivity", "specificity", "accuracy", "ppv", "npv"))

print(round(best\_coords, 3))

min\_max\_scale = function(x){

(x-min(x))/(max(x)-min(x))

}

library(dplyr)

library(glmnet)

data2 = completed\_data4%>%

mutate\_if(.predicate = is.numeric,

.funs = min\_max\_scale)%>%

as.data.frame()

set.seed(123) #random number generator

x <- data.matrix(data2[, -1])

y <- data2[, 1]

y<-as.numeric(unlist(y))

lasso <- glmnet(x, y, family = "binomial", nlambda = 1000, alpha = 1)

lasso.cv = cv.glmnet(x, y,alpha = 1,nfolds =20,family="binomial")

plot(lasso.cv)

lasso.cv$lambda.min #minimum

lasso.cv$lambda.1se #one standard error away

coef(lasso.cv, s = "lambda.1se")

lassov <- c("diagnosis","singleton\_status","assisted\_ventilation","birth\_weight","gestational\_weeks","initial\_feeding\_method","birth\_length","apgar\_1min")

mydata<-completed\_data4

attach(mydata)

dd <- datadist(mydata)

options(datadist = 'dd')

fit0 <- lrm(diagnosis ~ singleton\_status + assisted\_ventilation + birth\_weight + gestational\_weeks + initial\_feeding\_method + birth\_length + apgar\_1min, data = mydata, x = T, y = T)

gd<-predict(fit0,newdata = completed\_data4,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

roc.list<-roc(completed\_data4$diagnosis,gd)#low

roc.list

gd <- predict(fit0, newdata = completed\_data4, type = "fitted")

roc.list <- roc(completed\_data4$diagnosis, gd)

best\_coords <- coords(roc.list,

x = "best",

best.method = "youden",

ret = c("threshold", "sensitivity", "specificity", "accuracy", "ppv", "npv"))

print(round(best\_coords, 3))

min\_max\_scale = function(x){

(x-min(x))/(max(x)-min(x))

}

library(dplyr)

library(glmnet)

data2 = completed\_data5%>%

mutate\_if(.predicate = is.numeric,

.funs = min\_max\_scale)%>%

as.data.frame()

set.seed(123) #random number generator

x <- data.matrix(data2[, -1])

y <- data2[, 1]

y<-as.numeric(unlist(y))

lasso <- glmnet(x, y, family = "binomial", nlambda = 1000, alpha = 1)

lasso.cv = cv.glmnet(x, y,alpha = 1,nfolds =20,family="binomial")

plot(lasso.cv)

lasso.cv$lambda.min #minimum

lasso.cv$lambda.1se #one standard error away

coef(lasso.cv, s = "lambda.1se")

lassov <- c("diagnosis","singleton\_status","assisted\_ventilation","birth\_weight","gestational\_weeks","initial\_feeding\_method","birth\_length","apgar\_1min")

mydata<-completed\_data5

attach(mydata)

dd <- datadist(mydata)

options(datadist = 'dd')

fit0 <- lrm(diagnosis ~ singleton\_status + assisted\_ventilation + birth\_weight + gestational\_weeks + initial\_feeding\_method + birth\_length + apgar\_1min, data = mydata, x = T, y = T)

gd<-predict(fit0,newdata = completed\_data5,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

roc.list<-roc(completed\_data5$diagnosis,gd)#low

roc.list

gd <- predict(fit0, newdata = completed\_data5, type = "fitted")

roc.list <- roc(completed\_data5$diagnosis, gd)

best\_coords <- coords(roc.list,

x = "best",

best.method = "youden",

ret = c("threshold", "sensitivity", "specificity", "accuracy", "ppv", "npv"))

print(round(best\_coords, 3))

data <- na.omit(data)

head(data)

ordata <- data

colnames(data)

data[,1:19] <- lapply(data[,1:19],as.factor)

set.seed(3464)

trainIndex <- createDataPartition(data$diagnosis, p = .7, list = FALSE)

train <- data[trainIndex, ]

test <- data[-trainIndex, ]

write.csv(train, file = "train.csv", row.names = FALSE)

write.csv(test, file = "test.csv", row.names = FALSE)

min\_max\_scale = function(x){

(x-min(x))/(max(x)-min(x))

}

library(dplyr)

library(glmnet)

data2 = train%>%

mutate\_if(.predicate = is.numeric,

.funs = min\_max\_scale)%>%

as.data.frame()

set.seed(123) #random number generator

x <- data.matrix(data2[, -1])

y <- data2[, 1]

y<-as.numeric(unlist(y))

lasso <- glmnet(x, y, family = "binomial", nlambda = 1000, alpha = 1)

print(lasso)

plot(lasso, xvar = "lambda", label = FALSE)

lasso.cv = cv.glmnet(x, y,alpha = 1,nfolds =20,family="binomial")

plot(lasso.cv)

lasso.cv$lambda.min #minimum

lasso.cv$lambda.1se #one standard error away

coef(lasso.cv, s = "lambda.1se")

lassov <- c("diagnosis","umbilical\_cord\_status","assisted\_ventilation","birth\_weight","gestational\_weeks")

install.packages("rms")

library(rms)

mydata<-train

attach(mydata)

dd <- datadist(mydata)

options(datadist = 'dd')

fit0 <- lrm(diagnosis ~ umbilical\_cord\_status + assisted\_ventilation + birth\_weight + gestational\_weeks, data = mydata, x = T, y = T)

fit0

nom0 <- nomogram(fit0,fun = plogis,fun.at = c(.001,.01,.05,.5,.95,.99,.999),

lp = T,funlabel = "diagnosis rate")

plot(nom0)

install.packages("regplot")

library(regplot)

regplot(reg = fit0,

plots = c("density", "boxes"),

center = TRUE,

observation = mydata[1,],

title = "",

points = TRUE,

droplines = TRUE,

clickable = FALSE,

showP = TRUE,

rank = "sd",

failtime = c(1, 2, 3),

prfail = FALSE)

regplot(reg = fit0,

plots = c("density", "boxes"),

center = TRUE,

observation = mydata[1,],

title = "Diagnosis Probability",

points = TRUE,

droplines = TRUE,

clickable = FALSE,

showP = TRUE,

rank = "sd",

boxcol = "#808080",

dencol = "#4F94CD",

obscol = "red",

spkcol = "#696969",

cexscales = 0.6,

cexvars = 0.7,

cexcats = 0.7,

leftlabel = FALSE)

gd<-predict(fit0,newdata = train,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

roc.list<-roc(train$diagnosis,gd)#low

roc.list

g.list<- ggroc(roc.list,alpha = 1,size = 0.8,legacy.axes = TRUE,color="red")

g.list+theme\_classic()+ggtitle("data")+annotate(geom="segment",x = 0,y = 0,xend =1,yend = 1)

library(pROC)

library(ggplot2)

roc.list <- roc(train$diagnosis, gd)

g.list <- ggroc(roc.list, alpha = 1, size = 1, legacy.axes = TRUE, color = "#377EB8")

g.list <- g.list +

theme\_minimal(base\_size = 14) +

ggtitle("ROC Curve of Diagnostic Test") +

theme(plot.title = element\_text(hjust = 0.5, face = "bold", color = "#377EB8"),

axis.title = element\_text(face = "bold", color = "#4D4D4D"),

axis.text = element\_text(color = "#4D4D4D")) +

geom\_segment(aes(x = 0, y = 0, xend = 1, yend = 1), linetype = "dashed", color = "#E41A1C") +

annotate("text", x = 0.2, y = 0.9, label = sprintf("AUC = %.3f", auc(roc.list)), size = 5, color = "#4D4D4D")

print(g.list)

best\_coords <- coords(roc.list, "best", ret = c("threshold", "sensitivity", "specificity"), best.method = "youden")

print(best\_coords)

gd<-predict(fit0,newdata = test,

se.fit = FALSE,dispersion = NULL,trems= NULL,

na.action = na.pass)

#roc

library(pROC)

roc.list<-roc(test$diagnosis,gd)#low

roc.list

g.list<- ggroc(roc.list,alpha = 1,size = 0.8,legacy.axes = TRUE,color="red")

g.list+theme\_classic()+ggtitle("data")+annotate(geom="segment",x = 0,y = 0,xend =1,yend = 1)

library(pROC)

library(ggplot2)

roc.list <- roc(test$diagnosis, gd)

g.list <- ggroc(roc.list, alpha = 1, size = 1, legacy.axes = TRUE, color = "#377EB8")

g.list <- g.list +

theme\_minimal(base\_size = 14) +

ggtitle("ROC Curve of Diagnostic Test") +

theme(plot.title = element\_text(hjust = 0.5, face = "bold", color = "#377EB8"),

axis.title = element\_text(face = "bold", color = "#4D4D4D"),

axis.text = element\_text(color = "#4D4D4D")) +

geom\_segment(aes(x = 0, y = 0, xend = 1, yend = 1), linetype = "dashed", color = "#E41A1C") +

annotate("text", x = 0.2, y = 0.9, label = sprintf("AUC = %.3f", auc(roc.list)), size = 5, color = "#4D4D4D")

print(g.list)

best\_coords <- coords(roc.list, "best", ret = c("threshold", "sensitivity", "specificity"), best.method = "youden")

print(best\_coords)

train <- train[complete.cases(train[, c("diagnosis", "umbilical\_cord\_status", "assisted\_ventilation", "birth\_weight", "gestational\_weeks")]), ]

fit0 <- lrm(diagnosis ~ umbilical\_cord\_status + assisted\_ventilation + birth\_weight + gestational\_weeks, data = train, x = TRUE, y = TRUE)

cal <- calibrate(fit0, cmethod = "hare", method = "boot", B = 1000)

par(mar = c(5.1, 4.1, 4.1, 2.1))

plot(cal, xlim = c(0, 1), ylim = c(0, 1),

xlab = "Nomogram Predicted Survival", ylab = "Actual Survival")

test <- test[complete.cases(test[, c("diagnosis", "umbilical\_cord\_status", "assisted\_ventilation", "birth\_weight", "gestational\_weeks")]), ]

fit0 <- lrm(diagnosis ~ umbilical\_cord\_status + assisted\_ventilation + birth\_weight + gestational\_weeks, data = test, x = TRUE, y = TRUE)

cal <- calibrate(fit0, cmethod = "hare", method = "boot", B = 1000)

par(mar = c(5.1, 4.1, 4.1, 2.1))

plot(cal, xlim = c(0, 1), ylim = c(0, 1),

xlab = "Nomogram Predicted Survival", ylab = "Actual Survival")

library(rmda)

library(ggplot2)

library(rmda)

train$diagnosis <- ifelse(train$diagnosis == 2, 1, 0)

train <- train[complete.cases(train[, c("diagnosis", "umbilical\_cord\_status", "assisted\_ventilation", "birth\_weight", "gestational\_weeks")]), ]

dca\_model <- decision\_curve(diagnosis ~ umbilical\_cord\_status + assisted\_ventilation + birth\_weight + gestational\_weeks,

data = train,

study.design = "cohort",

bootstraps = 100)

library(rmda)

train$diagnosis <- ifelse(train$diagnosis == 2, 1, 0)

train <- train[complete.cases(train[, c("diagnosis", "umbilical\_cord\_status", "assisted\_ventilation", "birth\_weight", "gestational\_weeks")]), ]

dca\_model <- decision\_curve(diagnosis ~ umbilical\_cord\_status + assisted\_ventilation + birth\_weight + gestational\_weeks,

data = train,

study.design = "cohort",

bootstraps = 100)

plot\_decision\_curve(dca\_model,

curve.names = "Logistic Regression Model",

xlim = c(0, 1),

cost.benefit.axis = TRUE,

confidence.intervals = TRUE) +

theme(

legend.title = element\_blank(),

legend.text = element\_text(size = 8),

legend.key.size = unit(0.6, "lines"),

legend.position = c(1, 1),

legend.justification = c(1, 1)

)

library(rmda)

library(ggplot2)

test$diagnosis <- ifelse(test$diagnosis == 2, 1, 0)

test <- test[complete.cases(test[, c("diagnosis", "umbilical\_cord\_status", "assisted\_ventilation", "birth\_weight", "gestational\_weeks")]), ]

dca\_model\_test <- decision\_curve(diagnosis ~ umbilical\_cord\_status + assisted\_ventilation + birth\_weight + gestational\_weeks,

data = test,

study.design = "cohort",

bootstraps = 100)

plot\_decision\_curve(

dca\_model\_test,

curve.names = "Logistic Regression Model",

xlim = c(0, 1),

legend.position = "topright",

cost.benefit.axis = TRUE,

confidence.intervals = TRUE

) +

theme(

legend.title = element\_blank(),

legend.text = element\_text(size = 8),

legend.key.size = unit(0.6, "lines")

)

data$diagnosis\_binary <- ifelse(data$diagnosis == 2, 1, 0)

data$predicted\_prob <- predict(fit0, newdata = data, type = "fitted")

data$risk\_group <- cut(data$predicted\_prob,

breaks = c(0, 0.3, 0.7, 1),

labels = c("Low", "Moderate", "High"),

include.lowest = TRUE)

table(data$risk\_group, data$diagnosis\_binary)

chisq.test(table(data$risk\_group, data$diagnosis\_binary))

df <- data.frame(

risk\_group = factor(c("Low", "Moderate", "High"), levels = c("Low", "Moderate", "High")),

non\_sepsis = c(599, 39, 14),

sepsis = c(232, 77, 98)

)

df$total <- df$non\_sepsis + df$sepsis

df$sepsis\_rate <- df$sepsis / df$total

ci <- lapply(1:nrow(df), function(i) {

prop.test(df$sepsis[i], df$total[i])$conf.int

})

df$lower <- sapply(ci, function(x) x[1])

df$upper <- sapply(ci, function(x) x[2])

library(tidyverse)

df\_long <- df %>%

pivot\_longer(cols = c(non\_sepsis, sepsis),

names\_to = "diagnosis",

values\_to = "count") %>%

mutate(diagnosis = factor(diagnosis, levels = c("non\_sepsis", "sepsis"),

labels = c("No Sepsis", "Sepsis")))

ggplot(df\_long, aes(x = risk\_group, y = count, fill = diagnosis)) +

geom\_bar(stat = "identity") +

labs(title = "Figure 4A. Distribution of Sepsis and Non-Sepsis Across Risk Groups",

x = "Risk Group", y = "Number of Patients", fill = "Diagnosis") +

theme\_minimal(base\_size = 14)

ggplot(df, aes(x = risk\_group, y = sepsis\_rate)) +

geom\_point(size = 3, color = "firebrick") +

geom\_errorbar(aes(ymin = lower, ymax = upper), width = 0.15, color = "firebrick", size = 1) +

labs(title = "Figure 4B. Sepsis Rate with 95% CI by Risk Group",

x = "Risk Group", y = "Sepsis Rate") +

scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1)) +

theme\_minimal(base\_size = 14)

ggplot(df, aes(x = risk\_group, y = sepsis\_rate, group = 1)) +

geom\_line(color = "steelblue", size = 1.2) +

geom\_point(color = "steelblue", size = 3) +

labs(title = "Figure 4C. Trend of Sepsis Rate Across Risk Levels",

x = "Risk Group", y = "Sepsis Rate") +

scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1)) +

theme\_classic(base\_size = 14)

ggsave("figure4a\_sepsis\_barplot.pdf", width = 6, height = 4)

ggsave("figure4b\_sepsis\_rate\_ci.pdf", width = 6, height = 4)

ggsave("figure4c\_sepsis\_trend.pdf", width = 6, height = 4)

library(tidyverse)

library(patchwork)

p1 <- ggplot(df\_long, aes(x = risk\_group, y = count, fill = diagnosis)) +

geom\_bar(stat = "identity") +

labs(title = "Distribution of Sepsis and Non-Sepsis",

x = "Risk Group", y = "Number of Patients", fill = "Diagnosis") +

theme\_minimal(base\_size = 14)

p2 <- ggplot(df, aes(x = risk\_group, y = sepsis\_rate)) +

geom\_point(size = 3, color = "firebrick") +

geom\_errorbar(aes(ymin = lower, ymax = upper), width = 0.15, color = "firebrick", size = 1) +

labs(title = "Sepsis Rate with 95% CI",

x = "Risk Group", y = "Sepsis Rate") +

scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1)) +

theme\_minimal(base\_size = 14)

p3 <- ggplot(df, aes(x = risk\_group, y = sepsis\_rate, group = 1)) +

geom\_line(color = "steelblue", size = 1.2) +

geom\_point(color = "steelblue", size = 3) +

labs(title = "Trend of Sepsis Rate Across Risk Levels",

x = "Risk Group", y = "Sepsis Rate") +

scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1)) +

theme\_classic(base\_size = 14)

library(patchwork)

combo\_plot <- p1 + p2 + p3 +

plot\_layout(ncol = 3) +

plot\_annotation(tag\_levels = "A")

ggsave("Figure4\_ABC\_horizontal.pdf", plot = combo\_plot, width = 13, height = 5)

interaction\_model <- glm(diagnosis\_binary ~ umbilical\_cord\_status \* gestational\_weeks +

assisted\_ventilation + birth\_weight,

data = data,

family = binomial())

summary(interaction\_model)

install.packages("interactions")

library(interactions)

interact\_plot(interaction\_model,

pred = gestational\_weeks,

modx = umbilical\_cord\_status,

plot.points = TRUE,

interval = TRUE,

int.width = 0.95,

x.label = "Gestational Weeks",

y.label = "Predicted Probability of Sepsis")

interaction\_model2 <- glm(diagnosis\_binary ~ assisted\_ventilation \* birth\_weight +

umbilical\_cord\_status + gestational\_weeks,

data = data,

family = binomial())

summary(interaction\_model2)

library(interactions)

interact\_plot(interaction\_model2,

pred = birth\_weight,

modx = assisted\_ventilation,

plot.points = TRUE,

interval = TRUE,

int.width = 0.95,

x.label = "Birth Weight (g)",

y.label = "Predicted Probability of Sepsis")