Using Bayesian Analysis and Inference to Identify Medical Conditions and Patient Characteristics that Increase the Mortality Risk of COVID-19 Patients

Robin Lee, Ahmed Awadalla

STATS C116

2022-12-01

Abstract

With the onset of COVID-19 in 2020, countless doctors and nurses around the world struggled to accurately determine which patients required priority treatment. Unfortunately, at the peak of the pandemic, there were simply not enough supplies and resources to provide every victim the utmost care. Thus, hospital staff had to make difficult decisions by predicting each victim's mortality risk. This study aims to develop a predictor model of mortality risk from underlying medical condition parameters. In this report, we analyze 4611 patients with underlying medical conditions, who were laboratory-confirmed coronavirus disease 2019 (COVID-19), some of whom survived and some of whom did not. We use Bayesian analysis to analyze patient data from thousands of victims in the United States in order to determine which factors contribute to COVID-19's mortality rate the most. Because the virus is generally not deadly amongst young people with no pre-existing illnesses, we specifically examine patients with various common pre-existing medical conditions to predict which ones tend to have the highest COVID-19 mortality risk. By fitting a series of logistic regression models, we calculate and predict the mortality risk of COVID-19 patients who also possess other medical conditions. Given the large number of variables, we implement a Horseshoe Prior on the regression parameters to determine which of these conditions contain the highest death rates, before computing our model's Expected Log Pointwise Predictive Density (ELPD)—allowing us to measure its accuracy.

1 Introduction

As billions of people across the world prepared to celebrate a brand new year towards the waning days of 2019, the World Health Organization received several incongruous reports from Wuhan, China, regarding an unprecedented illness with pneumonia-like symptoms—the very first glimpses of Coronavirus (COVID-19). In the following months, this newfound virus rapidly spread across the globe, subsequently creating the largest world-wide epidemic in more than a century and affecting millions of lives. Contrary to a majority of other common respiratory viruses, COVID-19 leaves its victims in fluctuating levels of condition, ranging from relatively mild flu-like symptoms to almost certain death. Unfortunately, at the time of this report, more than a million people in the United States have lost their lives due to COVID-related complications.

Countless doctors and nurses around the world struggled to accurately determine which patients required priority treatment; unfortunately, at the peak of the pandemic, there were simply not enough supplies and resources to provide every victim the utmost care. Thus, hospital staff had to make these difficult decisions by predicting each victim's mortality risk. In this report, we use Bayesian Statistics to analyze patient data from thousands of victims in the United States in order to determine which factors contribute to COVID-19's mortality rate the most. By identifying the root cause behind these fatal outcomes, we discern which patients are at a higher risk of severe illness or mortality after contracting COVID, ultimately establishing the appropriate clinical decision-making.

Because the virus is generally not deadly among young people with no pre-existing illnesses, we specifically examine patients with various common pre-existing medical conditions to predict which ones tend to have the highest COVID-19 mortality risk. Thus, in this report, we examine nearly 5,000 patients—who possess a variety of common medical conditions—around the United States, and utilize Bayesian methods to ascertain the specific characteristics that lead to the highest mortality risk. By fitting a series of logistic regression models, we calculate and predict the mortality risk of COVID-19 patients who also possess other medical conditions. Given the large number of variables, we implement a Horseshoe Prior on the regression parameters to determine which of these conditions contain the highest death rates, before computing our model's Expected Log Pointwise Predictive Density (ELPD)—allowing us to measure its accuracy. As such, hospitals are able to predict and preemptively distinguish which victims are in significant danger, allowing doctors and nurses to systematically and effectively arrange necessary resources for those who need them the most.

1.1 Mortality Risk Data

The dataset, created by Harsh Walia on Kaggle (https://www.kaggle.com/datasets/harshwalia/mortality-risk-clinincal-data-of-covid19-patients), contains information regarding the following COVID-19 patient attributes: "demographics, comorbidities, admission laboratory values, admission medications, admission supplemental oxygen orders, discharge, and mortality". Because the data is obtained through a healthcare surveillance software package (Streamline Health: *Clinical Looking Glass*), the information in our dataset regarding COVID-19 patients—who have been admitted to a single healthcare system—is a thorough review of their primary medical records. This data is split over a specific period of time, and separated into the first 3 weeks of the pandemic and the following 3 weeks.

Containing about 50 different medical conditions and individual attributes, the dataframe we have selected contains more than 4700 individuals and 85 variables. Several of these predictors include: length of hospital stay (`LOS`), myocardial infraction (`MI`), peripheral vascular disease (`PVD`), congestive heart failure (`CHF`), cardiovascular disease (`CVD`), dementia (`Dement`), Chronic obstructive pulmonary disease (`COPD`), diabetes mellitus simple (`DM simple`), diabetes mellitus complicated (`DM complicated`), oxygen saturation (`OsSats`), mean arterial pressure, in mmHg (`MAP`), D-dimer, in mg/ml (`Ddimer`), platelets, in k per mm3 (`Plts`), international normalized ratio (`INR`), blood urea nitrogen, in mg/dL (`BUN`), alanine aminotransferase, in U/liter (`AST`), while blood cells, in per mm3 (`WBC`) and interleukin-6, in pg/ml (`IL-6`).

Rather than analyze all 84 predictors, we decided to reduce the number of factors to 60. This is because, given the nature of most healthcare institution questionnaires, patients are typically first asked a categorical question—where the response is a discrete “Yes” or “No”, represented in the dataset by `1` and `0` respectively. Thus, we removed most of these binary variables, as they ultimately proved to be quite repetitive. For example, the patients are required to answer whether or not they ever stayed at a hospital (`LOS_Y`), where they either respond “Yes” or “No”; this initial question is succeeded by another variable delineating how many days they stayed in the hospital for (`LOS`), where they respond with a numeric value. Because patients who did not stay at a hospital were simply assigned a `LOS` value of 0, we ultimately did not need a separate categorical variable to help us answer this question—clearly, `LOS =  0` signifies that the person spent 0 days at a hospital and, as such, definitely did not stay. Because many of the discrete-value predictors followed this pattern, we felt compelled to remove these categorical variables in order to tidy our dataset.

The predictor and response variables are shown below:

```
 [1] "Death"              "LOS"                "Age_Range"
 [4] "Severity"           "Black"              "White"
 [7] "Asian"              "Latino"             "MI"
[10] "PVD"                "CHF"                "CVD"
[13] "DEMENT"             "COPD"               "DM_C"
[16] "DM_S"               "Renal"              "All_CNS"
[19] "Pure_CNS"           "Stroke"             "Seizure"
[22] "OldSyncope"         "OldOtherNeuro"      "OtherBrnLsn"
[25] "Age"                "OsSats"             "OSat_lt_94"
[28] "Temp"               "Temp_gt_38"         "MAP"
[31] "MAP_lt_70"          "Ddimer"             "Ddimer_gt_3"
[34] "Plts"               "INR"                "INR_gt_1.2"
[37] "BUN"                "BUN_gt_30"          "Creatinine"
[40] "Sodium"             "Sodium_bt_139_154" "Glucose"
[43] "Glucose_bt_60_500" "AST"                "AST_gt_40"
[46] "ALT"                "ALT_gt_40"          "WBC"
[49] "WBC_bt_1_4"         "Lympho"             "Lympho_lt_1"
[52] "IL6"                "IL6_gt_150"         "Ferritin"
[55] "Ferritin_gt_300"    "CrctProtein"        "CrctProtein_gt_10"
[58] "Procalcitonin"      "Procalciton_gt_0"   "Troponin"
[61] "Troponin_gt_0"
```

1.2 Bayesian Statistical Methods

In order to identify the pre-existing medical conditions with the highest mortality risk, we strive to construct an accurate predictive model by implementing various common model selection strategies—specifically, fitting a sparse model with a “horseshoe” shaped prior on the logistic regression parameters to discern the most efficient fit between the model and the data. The Horseshoe Prior is utilized when there are many potential predictors for a given outcome, but it is not known which ones are actually relevant. By using this technique, we need to first declare a sparsity prior in order to specify the prior estimate of the number of non-zero variables within the dataset—essentially allowing us to predict the number of factors that negatively influence a patient’s mortality risk following their exposure to COVID-19.

Furthermore, we utilize the expected log pointwise predictive density (ELPD); as such, we are able to calculate the probability of producing a data set from our data generating process. The Expected Log Pointwise Predictive Density (ELPD) is a measure of the quality of a statistical model and how much it accurately fits not only our dataset, but also new data. In other words, if our model has a high probability of producing a dataset similar to our original COVID-19 mortality data and therefore possess a high ELPD value, we are confident it is accurate. Through these methods, we are able to build a predictive model of COVID-19 mortality risk.

2 Analysis and Results

2.1 Horseshoe Prior on Regression Parameters

Because we have a large number of predictors in our dataset, we use a Horseshoe Prior approach on our regression parameters—where $(y|\beta) \sim N(\beta, \sigma^2 I)$, and β is believed to be sparse—to elucidate the unknown sparsity and handle the significant amount of strong signals. The most notable characteristic of the sparsity prior is that we need to specify the prior estimate of the number of non-zero variables out of the selected 44 predictors; in other words, we want to predict the number of factors that contribute to COVID-related mortality before analyzing the data.

We decide to use this method, because the Horseshoe Prior is particularly useful as a shrinkage prior for sparse problems, allowing us to handle the dataset's relatively unknown sparsity and the number of large outlying signals. Because of its flat, Cauchy-like tails and the distribution's spike at the origin, the Prior allows the strong signals—variables that noticeably increase a patient's mortality risk following their exposure to COVID-19—to remain large, while the zero elements of β are severely shrunk. As mentioned earlier, the Horseshoe Prior is utilized when there are many potential predictors for the response (`Death`), but we are not sure which ones are actually relevant. The Prior helps us shrink the estimates of the irrelevant predictors towards zero, while preserving the relevant predictors. By diminishing the factors that are not useful, this method allows us to see which variables are sufficient in predicting COVID-19 mortality risk and which are not.

Applying the Horseshoe Prior

In order to implement the Horseshoe Prior, we standardize all the columns and create a new dataset (`covid1`) featuring all the modified data; this ensures that each value in the dataset is on the same numeric scale. Because we are utilizing a Logistic Regression, we do not need to standardize the response variable (`Death`), since all the values are either `0` (Did not die) or `1` (Died). This is shown as follows:

```r
covid$X <- data.matrix(covid[,2:ncol(covid)])
yf<-covid$Death
Xf<-covid$X
Xf<-t( (t(Xf)-apply(Xf,2,mean))/apply(Xf,2,sd))
n<-length(yf)
i.te<-sample(1:n,100)
i.tr<-(1:n)[-i.te]
y<-yf[i.tr] ; y.te<-yf[i.te]
X<-Xf[i.tr,]; X.te<-Xf[i.te,]
p=dim(X)[2]
covid1 <- as.data.frame(cbind(y,X))
```

With our new dataset, we specify the number of non-zero coefficients (`p_nonzero`). According to the CDC, individuals with obesity, diabetes, chronic lung disease, or sickle cell disease face an increased mortality risk following COVID-19; furthermore, older patients (typically above 65) and those who are immunocompromised are at significant danger as well. Because our dataset contains a large number of predictors, ranging from race to various medical conditions, we believe that there will be about 10 non-zero coefficients—meaning that we estimate that there will be 10 relevant predictors that increase risk.

An example of our dataset, after standardizing the values, is:

```
  y  Age OsSats OSat_lt_94 Temp Temp_gt_38   MAP MAP_lt_70 Ddimer Ddimer_gt_3
1 0 1.53   0.29      -0.81 0.14      -0.47  0.85     -0.28  -0.49       -0.57
2 1 0.22  -0.03       1.24 0.07      -0.47  0.17     -0.28  -0.07       -0.57
3 1 1.83   0.50      -0.81 0.44       2.12 -0.59     -0.28  -0.08       -0.57
```

Coming to this conclusion, we construct our prior distribution for the regression coefficients using the Hierarchical Shrinkage Family (exemplified by the `hs` function). We utilize this distribution, because it implements a half-Cauchy distributed standard deviation with a median of zero and a half-Cauchy scale parameter (these are the descriptors of the distribution); in other words, this function is the same as the Horseshoe Prior we are trying to apply.

After specifying our prior guess for the number of relevant variables that we believe will influence the COVID-19 fatality risk, we create our Bayesian Generalized Linear Model using the `stan_glm` function, allowing us to perform a full Bayesian estimation using Markov Chain Monte Carlo (MCMC). Because we have specified the value above, we are able to add priors onto the coefficients using `prior = hs_prior`, where `hs_prior` is the prior distribution for the regression coefficients. Because we want to find the number of non-zero variables out of the selected 44 predictors, we need to fit all 44 variables into the model—starting with a patient's length of stay at the hospital (`LOS`) and ending with their Troponin levels (`Troponin`).

It is important to note that even though there are many predictors that increase a patient's risk, we will not implement all non-zero coefficients in our final model. For example, while we are confident that the severity of a patient's symptoms ('severity') and their length of stay at the hospital ('LOS') will definitely contribute to an individual's mortality risk, we will not use these variables in our final model; this is because we are focusing specifically on pre-existing conditions and patient attributes. However, in order to fully analyze the dataset, our initial fit, named `fit`, contains all the parameters in our tidied dataset.

Applying these aforementioned concepts, the code to construct the Hierarchical Shrinkage Family and the Horseshoe Prior model appears as the following:

```
p_nonzero <- 10
tau0 <- p_nonzero/(p-p_nonzero) * 1/sqrt(n)
hs_prior <- hs(df=1, global_df=1, global_scale=tau0)
t_prior <- student_t(df = 7, location = 0, scale = 2.5)
fit <- stan_glm(y ~ LOS + Age_Range + Severity + Black + White + Asian + Latino +
                  MI + PVD + CHF + DEMENT + COPD + DM_C + DM_S + Renal + All_CNS +
                  Pure_CNS + Stroke + Seizure + OldSyncope + OldOtherNeuro +
                  OtherBrnLsn + Age + OsSats + OSat_lt_94 + Temp + Temp_gt_38 +
                  MAP + MAP_lt_70 + Ddimer + Ddimer_gt_3 + Plts + INR + INR_gt_1.2 +
                  BUN + BUN_gt_30 + Creatinine + Sodium + Sodium_bt_139_154 +
                  Glucose + Glucose_bt_60_500 + AST + AST_gt_40 + ALT + ALT_gt_40 +
                  WBC + WBC_bt_1_4 + Lympho + Lympho_lt_1 + IL6 + IL6_gt_150 +
                  Ferritin + Ferritin_gt_300 + CrctProtein + CrctProtein_gt_10 +
                  Procalcitonin + Procalciton_gt_0 + Troponin + Troponin_gt_0,
                data = covid1, family=binomial(),
                prior = hs_prior, prior_intercept = t_prior,
                seed = 1, adapt_delta = 0.99, refresh=0)
```

With our Generalized Bayesian model, we are able to plot each variable's signal and identify which predictors have a non-zero coefficient. As such, we locate the factors that prove to increase a patient's mortality risk after contracting COVID-19 by distinguishing which predictors deviate from the line at 0. In order to present this data in a detailed and comprehensive manner, we create two plots that essentially both display the coefficients, with the first graph exemplifying each variable's distribution and the second graph highlighting the actual value of the coefficient.

```
pplot <- plot(fit, "areas", prob = 0.95, prob_outer = 1)
plot1 <- pplot + geom_vline(xintercept = 0)
plot2 <- plot(fit)
grid.arrange(plot1, plot2, ncol=2)
```

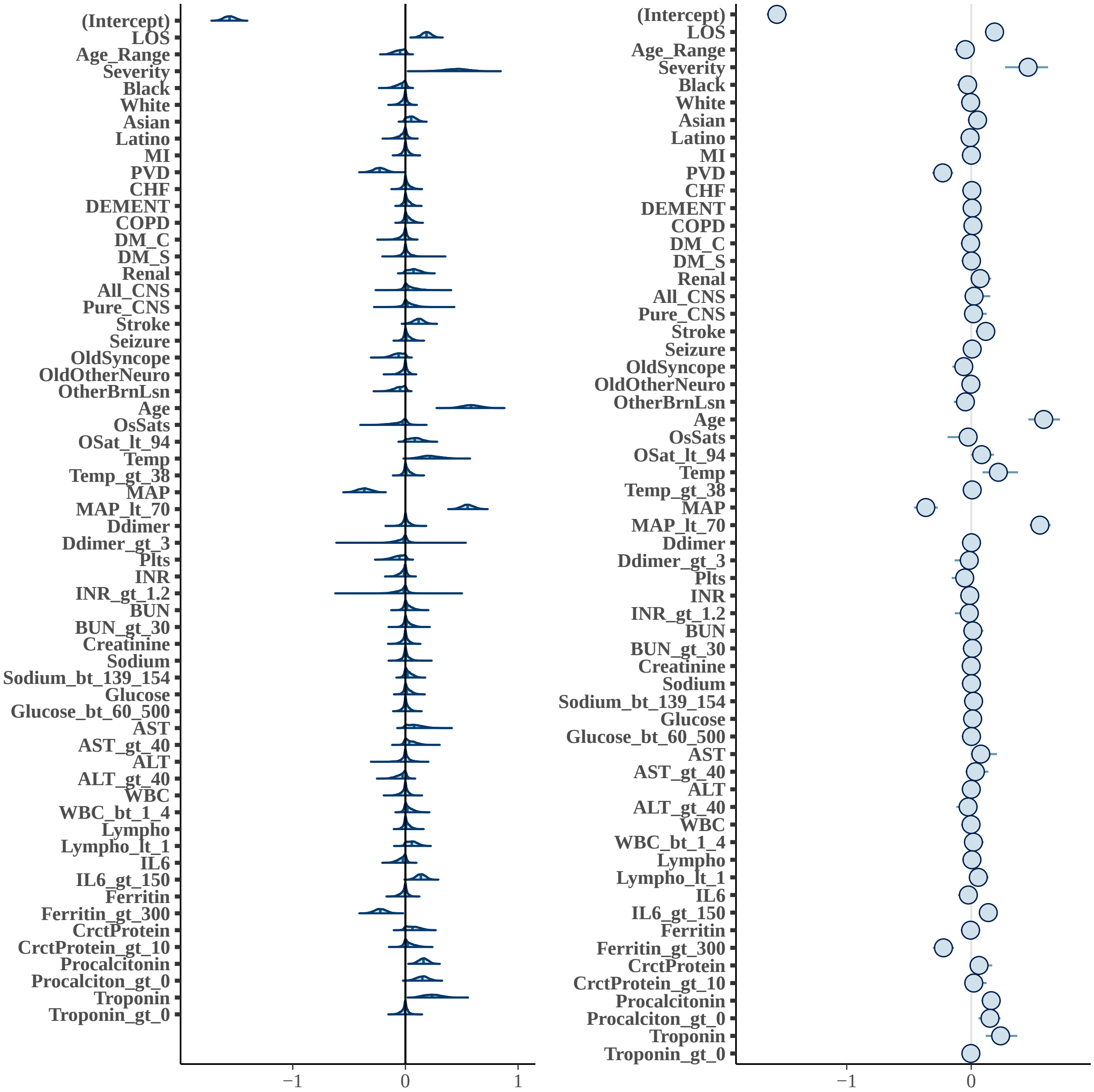

Figure 1: Horseshoe Prior on Regression Parameters in COVID-19 Patient Data

From the Figure 1, we can see that the following parameters significantly increase a patient's mortality risk following their exposure to COVID-19 and are all strong predictors: Peripheral Vascular Disease (`PVD`), End-Stage Renal Disease (`Renal`), Stroke (`Stroke`), Syncope (`OldSyncope`), Age (`Age`), Temperature (`Temp`), Mean Arterial Pressure in mmHg (`MAP`), Alanine Aminotransferase in U/liter (`AST`), Lymphocyte (`Lympho`), Interleukin-6 in pg/ml (`IL-6`), Ferritin (specifically Ferritin $>$ 300: `Ferritin_gt_300`), Procalcitonin (`Procalcitonin`), C-reactive Protein (`CrctProtein`), Troponin (`Troponin`). By implementing the Horseshoe Prior, we identify the parameters with the largest signals in relation to the `Death` parameter. Because we are evaluating the mortality risk of COVID-19, the response variable is whether or not a patient has died; thus, we use these variables in a Logistic Regression model to highlight which factors have the highest mortality rate.

2.2 Constructing the Logistic Regression Model

Logistic regression is a type of regression analysis that is typically utilized to predict the outcome of a categorical dependent variable, based on one or more independent variables. In this type of model, the dependent variable is binary, meaning that it can only take on two values (in our model, `Death` is denoted by `0` or `1`). The independent variables we have chosen are either continuous (such as `Stroke`) or categorical (such as `CrctProtein`). These parameters predict the probability that the dependent variable will take on a certain value: `1` or `0`.

Various studies have reported the clinical features of critical patients with COVID-19. In this study, we intend to analyze different clinical features and risk factors to identify the fatal consequences of the disease. Despite scientists' efforts to better understand the clinical features of the disease, the current understanding of the risk factors for COVID-19 is still ongoing, this study does not include every clinical feature of critical patients that have COVID-19; and therefore the study does have limitations. Hamidreza Kouhpayeh—a distinguished member of the Infectious Disease Department at the Zahedan Uiversity of Medical Sciences—found in 2022 that the overall mortality rate from COVID-19 is between 3.77% and 5.4%; however, we see this rate increase to between 41.1% and 61.5% among severe or critical patients [2]. In our analysis, we identify risk factors associated with disease severity and mortality in patients with COVID-19.

As mentioned earlier, logistic regression is a standard way to model binary outcomes (that is, response values y_i take the values: `0` or `1`). In the previous section, we have illustrated a logistic regression model using Bayesian ideas, specifically with prior knowledge regarding which variables are strong indicators of mortality; this method is the Horseshoe Prior. For each case i, we label $y_i = 1$ if the patient died or $y_i = 0$ if the patient survived. The logistic regression will help us in predicting the likelihood of mortality from COVID-19, given a patient possesses one or more of the variables at question.

A logistic regression model is fitted to the our dataset with dependent variable—specifically whether a patient survived or not—and the aforementioned independent variables that we selected using the Horseshoe Prior—specifically Peripheral Vascular Disease (`PVD`), End-Stage Renal Disease (`Renal`), Stroke (`Stroke`), Syncope (`OldSyncope`), Age (`Age`), Temperature (`Temp`), Mean Arterial Pressure in mmHg (`MAP`), Alanine Aminotransferase in U/liter (`AST`), Lymphocyte (`Lympho`), Interleukin-6 in pg/ml (`IL-6`), Ferritin (specifically Ferritin > 300: `Ferritin_gt_300`), Procalcitonin (`Procalcitonin`), C-reactive Protein (`CrctProtein`), Troponin (`Troponin`).

We model the probability that y = 1:

$$Pr(y_i = 1)\text{logit}^{-1}(X_iB); \text{logit}^{-1}(x) = \frac{e^x}{1+e^x}$$

Notice that: $\text{logit}^{-1}(x)$ transforms continuous values to the range (0,1), which is necessary, since probabilities must be bewteen 0 and 1.

```
fit_bayes <- stan_glm(y ~ Age + PVD + Renal + Stroke + OldSyncope + Temp + MAP + AST +
                        Lympho + Ferritin_gt_300 + Procalcitonin + CrctProtein + Troponin,
                      prior = normal(), prior_intercept = normal(),
                      family=binomial(link="logit"),data= covid1)
```

```
SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000292 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.92 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
```

```
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 1.75482 seconds (Warm-up)
Chain 1:                1.46128 seconds (Sampling)
Chain 1:                3.21611 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 0.000137 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 1.37 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 1.98908 seconds (Warm-up)
Chain 2:                2.2881 seconds (Sampling)
Chain 2:                4.27718 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 0.000289 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 2.89 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 1.93493 seconds (Warm-up)
Chain 3:                2.25129 seconds (Sampling)
Chain 3:                4.18622 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 0.00014 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 1.4 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 2.00068 seconds (Warm-up)
Chain 4:                2.02321 seconds (Sampling)
Chain 4:                4.02389 seconds (Total)
Chain 4:
```

We choose `Age`, `PVD`, `Renal`, `Stroke`, `OldSyncope`, `Temp`, `MAP`, `AST`, `Lympho`, `Ferritin_gt_300`, `Procalcitonin`, `CrctProtein`, and `Troponin` as the covariates we want to focus on. We choose these covariates, because our Horseshoe Prior shows that these parameters are strong predictors that significantly increase mortality risk. Furthermore, they are all relatively independent of each other, which means that they contain a lot of non-intersecting predictive capacity. The results are:

```r
summary(fit_bayes, digits= 3)
```

```
Model Info:
 function:     stan_glm
 family:       binomial [logit]
 formula:      y ~ Age + PVD + Renal + Stroke + OldSyncope + Temp + MAP + AST +
      Lympho + Ferritin_gt_300 + Procalcitonin + CrctProtein +
      Troponin
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help(’prior_summary’)
 observations: 4611
 predictors:   14

Estimates:
                  mean   sd     10%    50%    90%
(Intercept)     -1.501  0.046 -1.561 -1.501 -1.443
Age              0.859  0.049  0.796  0.859  0.921
PVD             -0.276  0.047 -0.338 -0.276 -0.216
Renal            0.141  0.039  0.092  0.141  0.192
Stroke           0.145  0.035  0.101  0.145  0.191
OldSyncope      -0.062  0.041 -0.116 -0.061 -0.010
Temp             0.474  0.055  0.405  0.473  0.545
MAP             -0.816  0.049 -0.879 -0.816 -0.753
AST              0.199  0.063  0.118  0.196  0.282
Lympho           0.002  0.037 -0.045  0.004  0.048
Ferritin_gt_300 -0.116  0.044 -0.173 -0.115 -0.060
Procalcitonin    0.222  0.037  0.174  0.222  0.269
CrctProtein      0.343  0.042  0.288  0.343  0.396
Troponin         0.415  0.075  0.321  0.413  0.512

Fit Diagnostics:
           mean   sd    10%   50%   90%
mean_PPD 0.242  0.008 0.232 0.241 0.252

The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for detai

MCMC diagnostics
                mcse  Rhat  n_eff
(Intercept)     0.001 1.000 4331
Age             0.001 1.000 4676
PVD             0.001 1.000 5284
Renal           0.001 1.000 5714
Stroke          0.000 0.999 6184
OldSyncope      0.001 1.000 5299
Temp            0.001 1.000 4487
MAP             0.001 1.001 3909
AST             0.001 1.000 4920
Lympho          0.000 0.999 5775
Ferritin_gt_300 0.001 1.000 4385
Procalcitonin   0.000 0.999 6062
CrctProtein     0.001 0.999 4529
Troponin        0.001 0.999 6226
mean_PPD        0.000 1.000 4860
log-posterior   0.061 1.001 1903

For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample si
```

Table 1: Coefficients of Logistic Regression Model

Taking a look at Table 1, the parameter estimates no longer have test statistics and p-values as in the Frequentist approach. This is because—unlike a Frequentist approach—Bayesian estimation samples from the posterior distribution, which means that instead of a point estimate and a test statistic, we get a distribution of plausible values for the parameters; the estimates section summarizes those distributions. Specifically, we get the mean, standard deviation, and commonly used percentiles.

The parameters in the estimates section (other than the coefficients we have entered into the model) are sigma—which represents the standard deviation of errors—amd mean_ppd—the mean of the posterior predictive distribution of our outcome variable, `Death`. Finally, log-posterior is analogous to the likelihood; this represents the log of the combined posterior distributions, which will be used for model comparisons.

```
mcmc_dens(fit_bayes)
```

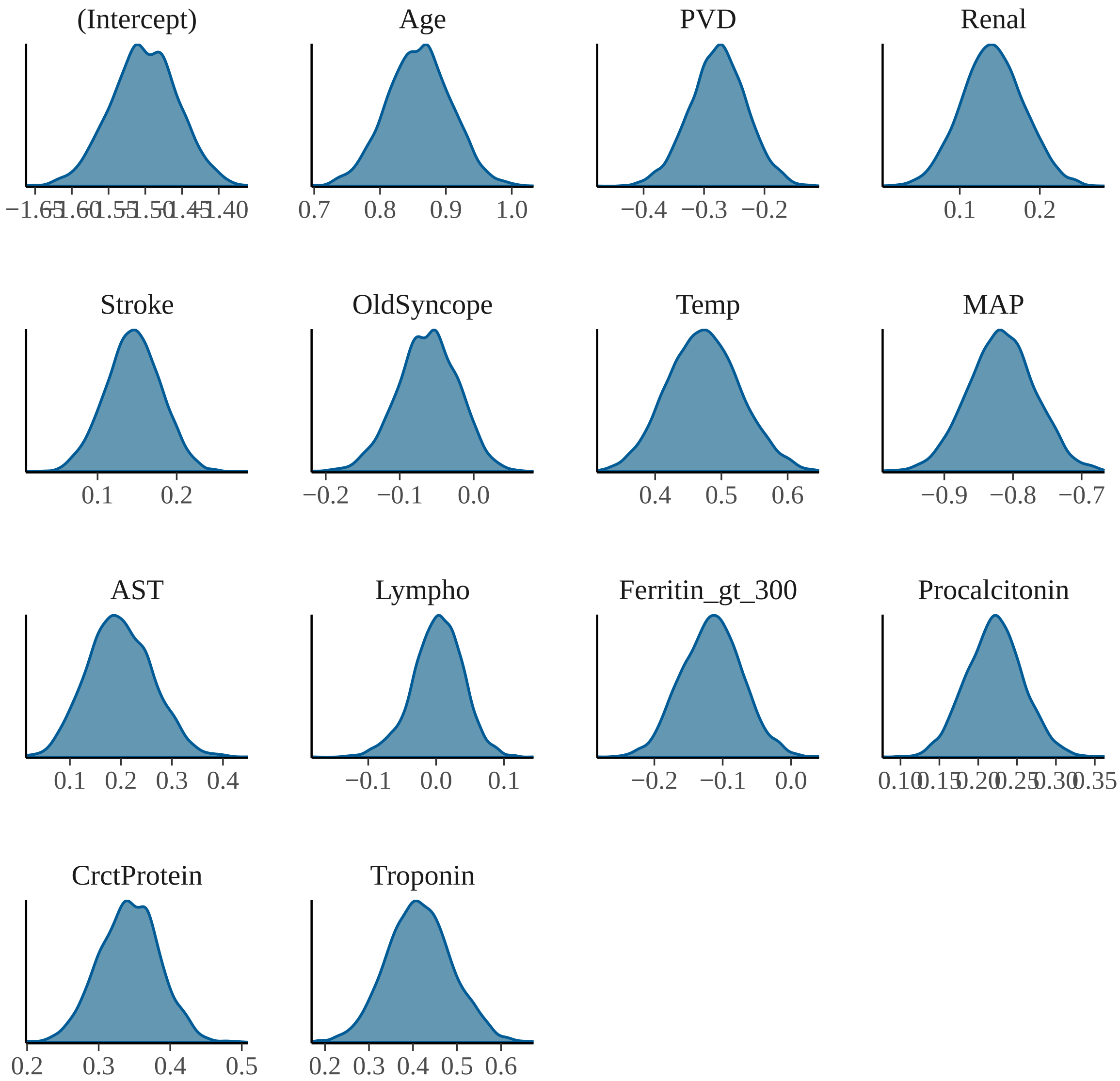

Figure 2: Scrunched Pairs Plot of Logistic Regression for Risk Factors

The above density plots confirm that our estimates are normal, which means that we know the central limit theorem is effective. Our scrunched pairs plot takes a closer look at the distribution of each parameter.

Noticeably, each predictor has a peak that is above or below 0, indicating that they are significant. For example, we can see that `Age` and `MAP` are strong indicators for COVID-related mortality.

```
pp_check(fit_bayes, "dens_overlay")
```

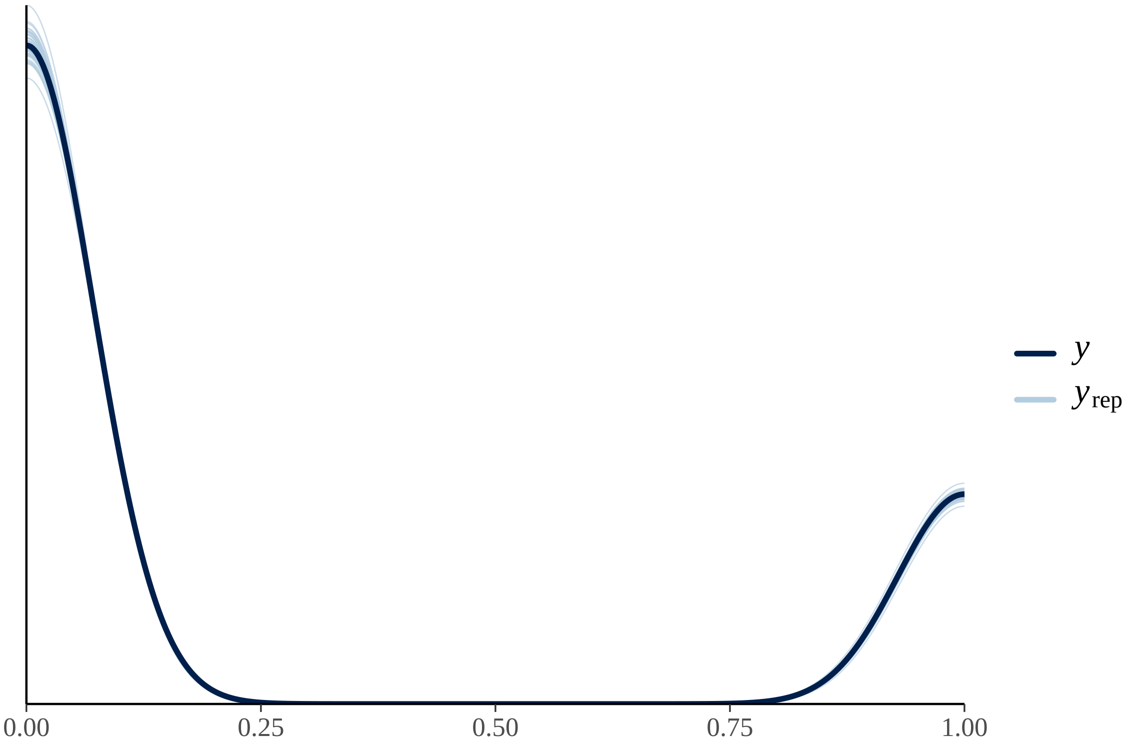

Figure 3: Factor-Analysis-Based Logistic Regression for risk factors associated with COVID-19 mortality

Unlike in the Frequentist regression—where there is always a solution using ordinary least squares, in Bayesian models, we have to check to make sure the model converged. If a model converges, then we are confident that the parameter estimates are stable. Otherwise, our results are unreliable. In Bayesian estimation, posterior distributions are sampled in groups, known as chains. We can measure the stability of our estimates by comparing the variance within chains to the variance across chains, which is denoted by the R-hat statistic. In general, we want all R-hat values to be close to 1 in order to conclude the model has converged, as in this example. Taking a look at Figure 3 (right), we can see our model fits the observed data pretty well, which allows us to make inferences about the data and shows that our model is accurate.

```
p_direction(fit_bayes)
```

```
Probability of Direction

Parameter       |     pd
------------------------
(Intercept)     |   100%
Age             |   100%
PVD             |   100%
Renal           |   100%
Stroke          |   100%
OldSyncope      | 94.03%
```

```
Temp            |    100%
MAP             |    100%
AST             |    100%
Lympho          | 54.47%
Ferritin_gt_300 | 99.60%
Procalcitonin   |    100%
CrctProtein     |    100%
Troponin        |    100%
```

Table 2: Probability of Direction for Logistic Regression Model

Further corroborating the belief that our model fits the data well, we calculate the probability of direction—a measure of the likelihood that a given event will occur in a specific direction. To calculate the probability, we identify the so-called event we are interested in (the predictors) and the specific direction we want to assess the likelihood of (the data). Once we identify these two values, we count the number of times the event occurred in that direction. In order to find the percentage, we divide each value associated with a parameter by the total number of times the event occurred, giving the probability of direction between 0 and 1.

In Table 2, we can see the probability of direction of our variables, this allows us to see which variables are statistically significant. If the value is over 98% we can say it is statistically significant. From Table 2, we can see that all variables—except for `Lympho` and `OldSnycope`—are significant in our analysis.

Our analysis suggests that Peripheral Vascular Disease (`PVD`), End-Stage Renal Disease (`Renal`), Stroke (`Stroke`), Age (`Age`), Temperature (`Temp`), Mean Arterial Pressure in mmHg (`MAP`), Alanine Aminotransferase in U/liter (`AST`), Ferritin (specifically Ferritin > 300: `Ferritin_gt_300`), Procalcitonin (`Procalcitonin`), C-reactive Protein (`CrctProtein`), Troponin (`Troponin`) are important risk factors for this disease. These results have important clinical implications, such as clinical management and specific preventive measures for patients with these underlying medical conditions.

In conclusion, the selection of appropriate clinical indicators for early identification and class treatment of COVID-19 patients is very important and can help save lives.

2.3 Expected Log Pointwise Predictive Density (ELPD)

After selecting the parameters that appear to increase a patient's mortality risk, we want to calculate just how accurate our logistic regression model is and compute its prediction accuracy. The Expected Log Pointwise Predictive Density (ELPD) is a measure of the quality of our statistical model and how much it fits not only the current dataset, but also any new data.

By juxtaposing the fit of our logit model to a probit regression—which determines categorical likelihood rather than odds of success (logit)—using the same data, we employ the theoretical Expected Log Pointwise Predictive Density (ELPD) of a new dataset. As such, we utilize cross-validation to compare how well our model prognosticates any new observations and calculate the predictive density values for each observation, ensuring the model's accuracy.

In order to understand how this process works, we let $y_1^{new}, ...y_n^{new}$ be a new dataset created by the ELPD process, where the covariates are the same as the original dataset:

$$P(y_i^{new}|y_1, ..., y_n) = \int P(y_i^{new}|\theta)P(\theta|y_1, ..., y_n)d\theta, \text{ where } i = 1, ..., n$$

For each prediction, we are calculating the log-likelihood (probability) of each observation given our model. We obtain the ELPD by averaging the log-likelihood of each observations in the dataset, since it shows how well the model predicts the data. Thus, because we want to construct a model that has a high probability of producing a dataset relatively similar to our original data, we focus on the new dataset's ELPD after the generating process:

$$\text{elpd} = \text{E}_{y^{new}}[\log P(y^{new}|y)] = \Sigma_{i=1}^{n} p(y_i^{new}) \log P(y_i^{new}|y_1, ..., y_n) dy_i^{new}$$

If our model has a high probability of generating a dataset similar to the one we are using, then we are confident that it should be a good model. In general, a model with a higher ELPD is considered to be a better fit for the data and more likely to make accurate predictions of new data.

We want to use cross-validation (as mentioned above), because the exact computation of the ELPD is mathematically challenging and estimating it is the most efficient method. This procedure essentially removes a single observation from the data and refits our model with the other patients' information. Thus, by using cross-validation, we predict the point we removed in order to show how accurate the model is. The natural estimator is:

$$\widehat{\text{elpd}} = \Sigma_{i=1}^{n} \log P(y_i|y_{-i})$$

where the posterior probability of the observed y_i is:

$$P(y_i|y_{-1}) = \int P(y_i|\theta)P(\theta|y_{-i})d\theta$$

.

Because we have more than 4700 independent observations, we decide to only sample 200 patients. This optimizes our ELPD, since the function requires us to cross-validate each observation and predictor.

```
covid2 <- sample_n(covid1, 200)
fit1 <- stan_glm(y ~ PVD + Renal + Stroke + OldSyncope + OtherBrnLsn + Age + OSat_lt_94 +
                   Temp + MAP + MAP_lt_70 + AST + Lympho_lt_1 + IL6_gt_150 +
                   Ferritin_gt_300 + Procalcitonin + CrctProtein +
                   Procalciton_gt_0 + Troponin,
                data = covid2, family=binomial(),
                prior = hs_prior, prior_intercept = t_prior,
                seed = 1, adapt_delta = 0.99, refresh=0)
```

Using the predictors we found through the Horseshoe prior, the model we are testing looks like such:

Note: the values are different from the logistic regression model in section 2.2, because we are applying a Bayesian model like in section 2.1. This means that we are applying a Horseshoe Prior using `stan_glm`.

```
summary(fit1)[,1]
```

```
    (Intercept)              PVD            Renal           Stroke 
  -1.492377e+00    -1.234787e-01    -1.171916e-01     1.227878e-01 
     OldSyncope      OtherBrnLsn              Age       OSat_lt_94 
  -4.111225e-03    -2.585580e-01     8.745635e-01     3.332733e-01 
           Temp              MAP        MAP_lt_70              AST 
  -2.470389e-02    -1.894708e-01     8.428758e-01     4.680223e-01 
    Lympho_lt_1       IL6_gt_150  Ferritin_gt_300    Procalcitonin 
   1.912087e-01     9.599600e-02    -1.695625e-01     3.384025e-01 
    CrctProtein Procalciton_gt_0         Troponin         mean_PPD 
   1.103390e-01     4.983202e-01     2.854325e-01     3.057937e-01 
  log-posterior 
  -1.869916e+02 
```

In order to measure how well our chosen relevant variables predict the response variable, we plot the ELPD and the root mean squared error (RMSE) of each predictor:

```
refmodel <- get_refmodel(fit1)
vs <- cv_varsel(refmodel, method='forward', cores=2)
plot(vs, stats = 'elpd')
```

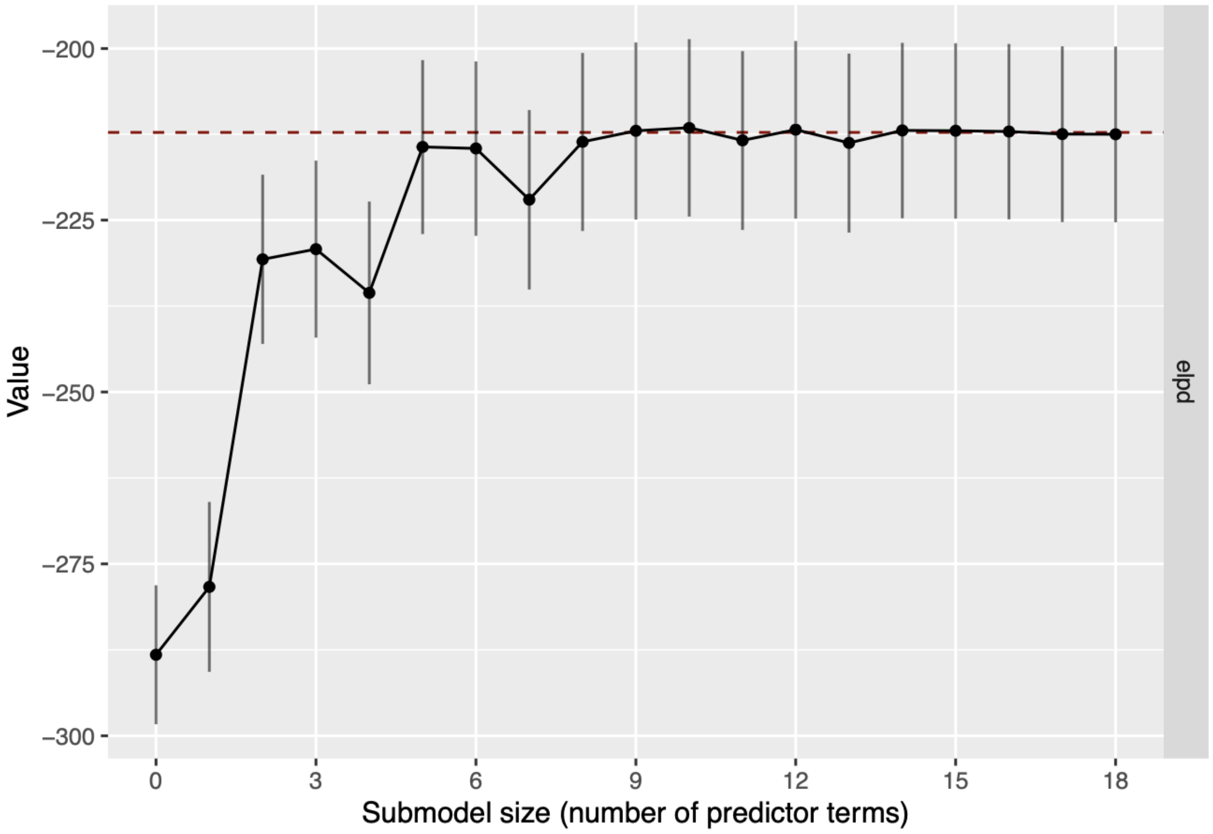

Figure 2: Expected Log Pointwise Predictive Density Validation

Using the `cv_varsel` method, we are able to implement cross-validation to see how many of variables should be included in our final model. Because we want to measure how accurate our model is and compare y_i (the removed observation) to the posterior predictive for y_i, we use the root mean squared error to measure the distance between the actual value and the predicted value. In the plot above (Figure 2), the line measures the overall quality of the fit.

Thus, we need to find the x value that ensures the line is the most optimal; this means that we have to find an ideal number of predictors—which is labeled on the x-axis—that is not only closest to the dotted line, but also does not contain too many variables. We do not want too many variables, because even though the line is extremely close to the dotted line, the quality of the fit actually decreases and is not necessarily more accurate.

Comparing the Expected Log Pointwise Predictive Density relative to the full model:

```
plot(vs, stats = c('elpd', 'rmse'), deltas = TRUE)
```

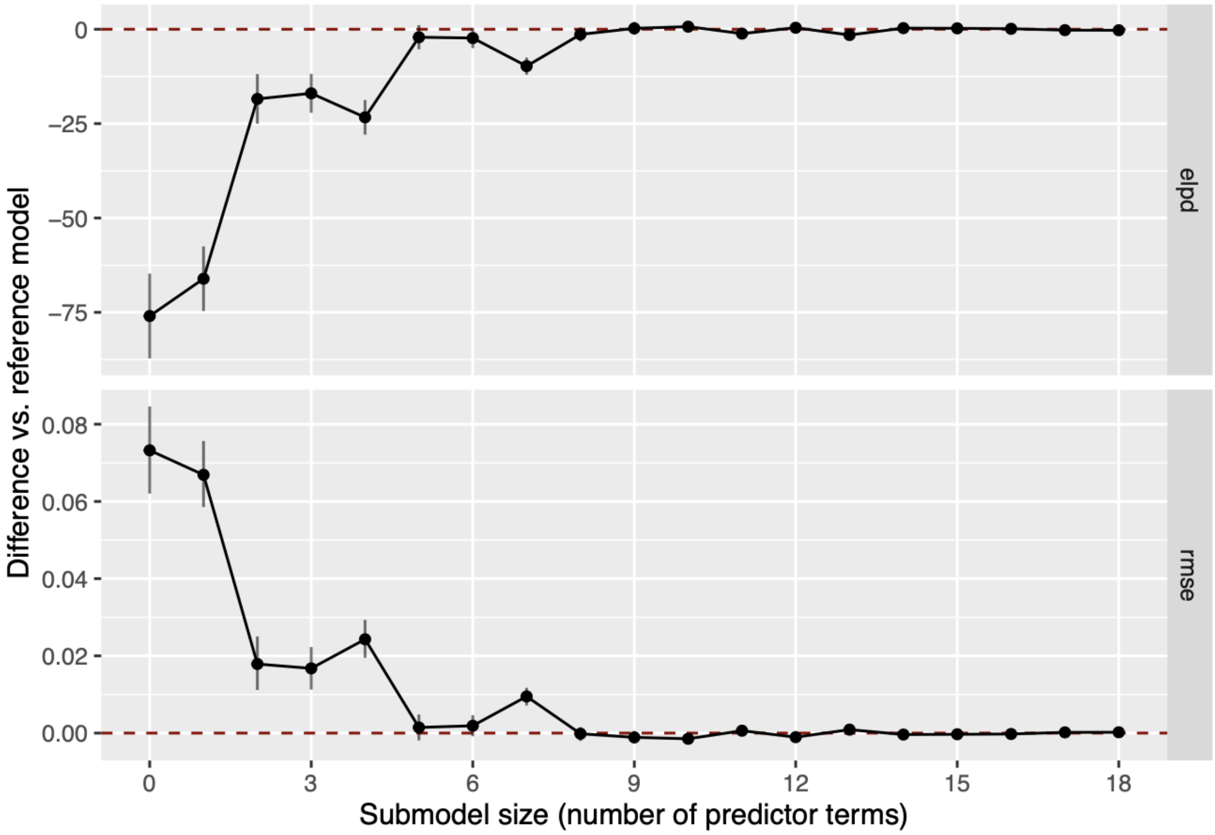

Figure 3: Expected Log Pointwise Predictive Density Validation Relative to the Full Model

Therefore, after plotting the ELPD and the root mean squared error (RMSE) in Figure 2 and 3, we see the model start to over-fit after about 9 predictors. This is because the number of predictors that is closest to the dotted line is around 9, and we do not want to risk decreasing the quality of our fit by including too many parameters. However, while we believe the ideal number of variables is 9, we decide to use the 11 predictors as mentioned in the Section 2.2 Constructing the Logistic Regression Model; this is because we believe that several of the predictors (such as `Age` and `Stroke`) work hand-in-hand to increase COVID-19 mortality risk. Thus, in order to better encapsulate the data, we implement a few more parameters to our model than recommended. Comparing to the full model (Figure 3), the two plots are extremely similar.

3 Discussion

Conclusion

COVID-19 patient's mortality risk could be predicted by developing a Logistic Regression model with Peripheral Vascular Disease (`PVD`), End-Stage Renal Disease (`Renal`), Stroke (`Stroke`), Syncope (`OldSyncope`), Age (`Age`), Temperature (`Temp`), Mean Arterial Pressure in mmHg (`MAP`), Alanine Aminotransferase in U/liter (`AST`), Lymphocyte (`Lympho`), Interleukin-6 in pg/ml (`IL-6`), Ferritin (specifically Ferritin > 300: `Ferritin_gt_300`), Procalcitonin (`Procalcitonin`),C-reactive Protein (`CrctProtein`), Troponin (`Troponin`) as predictors.

Further analysis showed that these 11 predictors in particular model risk the best: Peripheral Vascular Disease (`PVD`), End-Stage Renal Disease (`Renal`), Stroke (`Stroke`), Age (`Age`), Temperature (`Temp`), Mean Arterial Pressure in mmHg (`MAP`), Alanine Aminotransferase in U/liter (`AST`), Ferritin (specifically Ferritin > 300: `Ferritin_gt_300`), Procalcitonin (`Procalcitonin`), C-reactive Protein (`CrctProtein`), Troponin (`Troponin`)

The model developed has shown a good performance based on all metrics. It can help hospitals prioritize patients who are really in need and reduce the mortality rate. However, based on other studies our data does not include all clinical features of critical patients with COVID-19, which may result in a lack of recognizing the pattern. In future studies, gathering more data on different clinical features for training is expected.

Limitations and Further Research

A significant issue that we faced while creating our model involved the Expected Log Pointwise Predictive Density (ELPD). Because we utilized cross-validation to compare our model with any new values, we had to compare each observation and variable, as well as calculate the log-likelihood (probability) of each value given our model. Thus, with 4711 observations and more than 40 parameters, computing the entire dataset creates issues with running the `cv_varsel` function. As such, we had to sample the dataset and utilize only 200 observations. This means that the model might not be the most accurate, since it does not represent the entire data. While the overall data for each variable is normal, this is not necessarily the case for a sample, because we are randomly sampling. This is exemplified by our code below:

```r
covid2 <- sample_n(covid1, 200)
fit1 <- stan_glm(y ~ PVD + Renal + Stroke + OldSyncope + OtherBrnLsn + Age + OSat_lt_94 +
                   Temp + MAP + MAP_lt_70 + AST + Lympho_lt_1 + IL6_gt_150 +
                   Ferritin_gt_300 + Procalcitonin + CrctProtein +
                   Procalciton_gt_0 + Troponin,
                data = covid2, family=binomial(),
                prior = hs_prior, prior_intercept = t_prior,
                seed = 1, adapt_delta = 0.99, refresh=0)
```

Furthermore, there are many factors that contribute to COVID's mortality risk. While we focused specifically on certain medical conditions, there are countless other pre-existing diseases and conditions—such as gender—that work alongside the variables in our dataset to increase fatality. Because there are other significant factors that we failed to include, we are not completely confident that all the predictors we selected in our model actually increase mortality risk, but if there are underlying factors that played a role.

While we do not have missing data, the data only includes patients from a specific healthcare surveillance software package in the state of Georgia. This means that the individuals we have analyzed are relatively financially and economically stable, because they are able to afford healthcare. Thus, our dataset is not only unrepresentative of the United States, but the world as a whole. Even though we have more than 4711 individuals, this is small in comparison to the total number of people who contracted COVID-19.

```r
refmodel <- get_refmodel(fit1)
vs <- cv_varsel(refmodel, method='forward', cores=2)
plot(vs, stats = 'elpd')
```

For further research, we want to run an ELPD for the entire dataset with better resources, since our computers were unable to handle beyond 4 cores in the `cv_varsel` function. Taking a look at the code above, we only utilized 2 cores. We also believe that including more factors—such as gender—would improve our model's accuracy, as well as more data from across the United States and the world. Ultimately, we want our model to be representative of all individuals who contract COVID-19.

4 References

[1] Chowdhury, M. E. H., Rahman, T., Khandakar, A., Al-Madeed, S., Zughaier, S. M., Doi, S. A. R., Hassen, H., & Islam, M. T. (2021, April 21). *An early warning tool for predicting mortality risk of COVID-19 patients using machine learning - cognitive computation.* SpringerLink. Retrieved December 9, 2022, from https://link.springer.com/article/10.1007/s12559-020-09812-7

[2] Kouhpayeh, H. (2022, April 12). *Clinical Features Predicting COVID-19 Mortality Risk.* European journal of translational myology. Retrieved December 9, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9295175/

[3] The Novel Coronavirus Pneumonia Emergency Response Epidemiology Team. (2020, February 1). *The Epidemiological Characteristics of an Outbreak of 2019 Novel Coronavirus Diseases (COVID-19) - China, 2020.* China CDC Weekly. Retrieved December 9, 2022, from https://weekly.chinacdc.cn/en/article/id/e53946e2-c6c4-41e9-9a9b-fea8db1a8f51

[4] *WHO China Joint Mission on COVID-19 Final Report.* (n.d.). Retrieved December 10, 2022, from https://www.who.int/docs/default-source/coronaviruse/who-china-joint-mission-on-covid-19-final-report

[5] Centers for Disease Control and Prevention. (n.d.). *CDC Covid Data Tracker.* Centers for Disease Control and Prevention. Retrieved December 9, 2022, from https://covid.cdc.gov/covid-data-tracker/#demographicsovertime

[6] *Age, Sex, Existing Conditions of COVID-19 Cases and Deaths.* Worldometer. (n.d.). Retrieved December 9, 2022, from https://www.worldometers.info/coronavirus/coronavirus-age-sex-demographics/

[7] Gelman, A., & Hill, J. (2018). *Data Analysis using Regression and Multilevel/Hierarchical Models.* Cambridge Univ. Press.

5 Tables and Figures

```r
pplot <- plot(fit, "areas", prob = 0.95, prob_outer = 1)
plot1 <- pplot + geom_vline(xintercept = 0)
plot2 <- plot(fit)
grid.arrange(plot1, plot2, ncol=2)
```

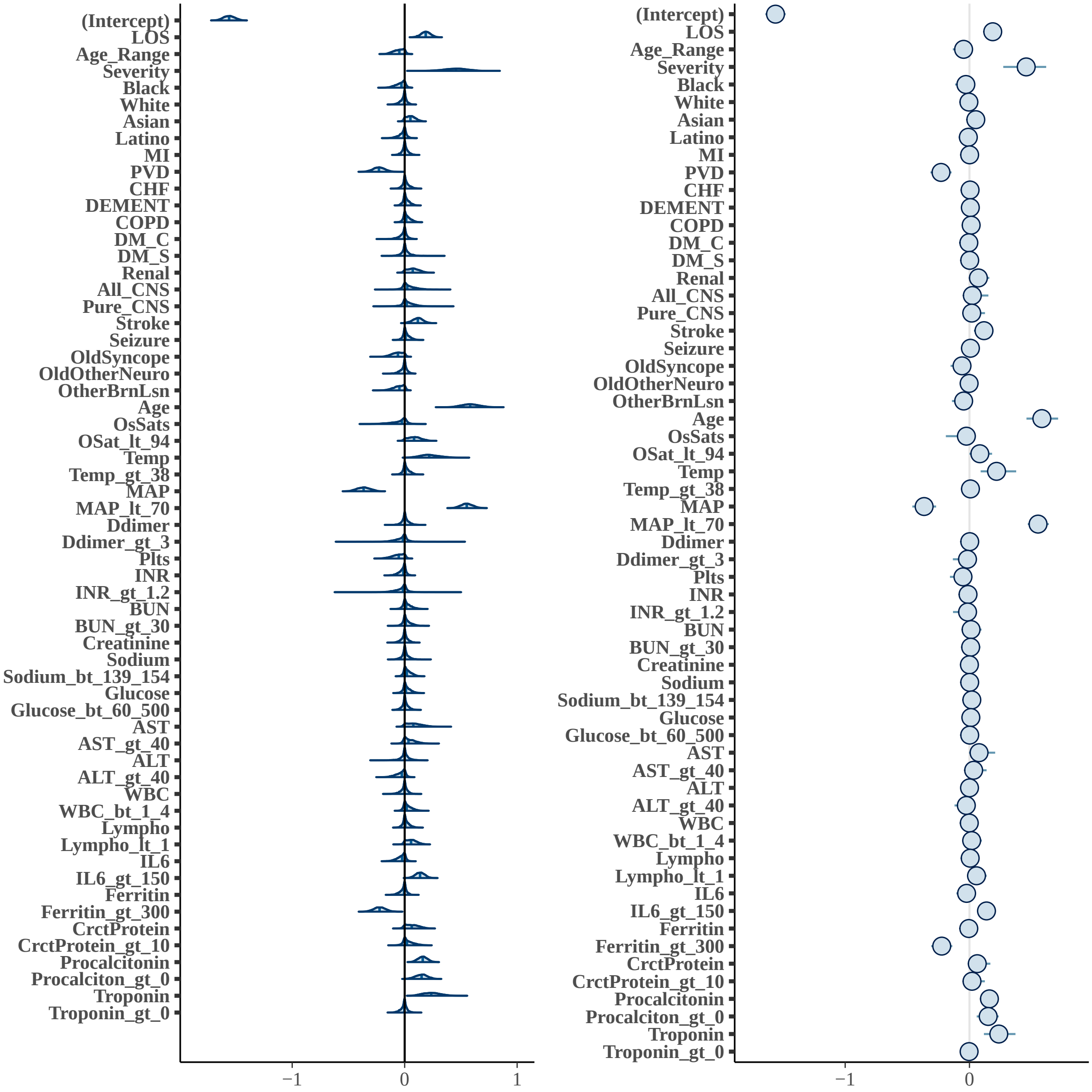

Figure 1: Horseshoe Prior on Regression Parameters in COVID-19 Patient Data

```
summary(fit_bayes, digits= 3)[,1]
```

```
    (Intercept)              Age              PVD            Renal           Stroke
  -1.501137e+00     8.588931e-01    -2.764302e-01     1.413202e-01     1.453521e-01
     OldSyncope             Temp              MAP              AST           Lympho
  -6.235766e-02     4.739175e-01    -8.163017e-01     1.986250e-01     2.423800e-03
Ferritin_gt_300   Procalcitonin      CrctProtein         Troponin         mean_PPD
  -1.156939e-01     2.215270e-01     3.428591e-01     4.150359e-01     2.415425e-01
  log-posterior
  -2.006320e+03
```

Table 1: Coefficients of Logistic Regression Model

```
mcmc_dens(fit_bayes)
```

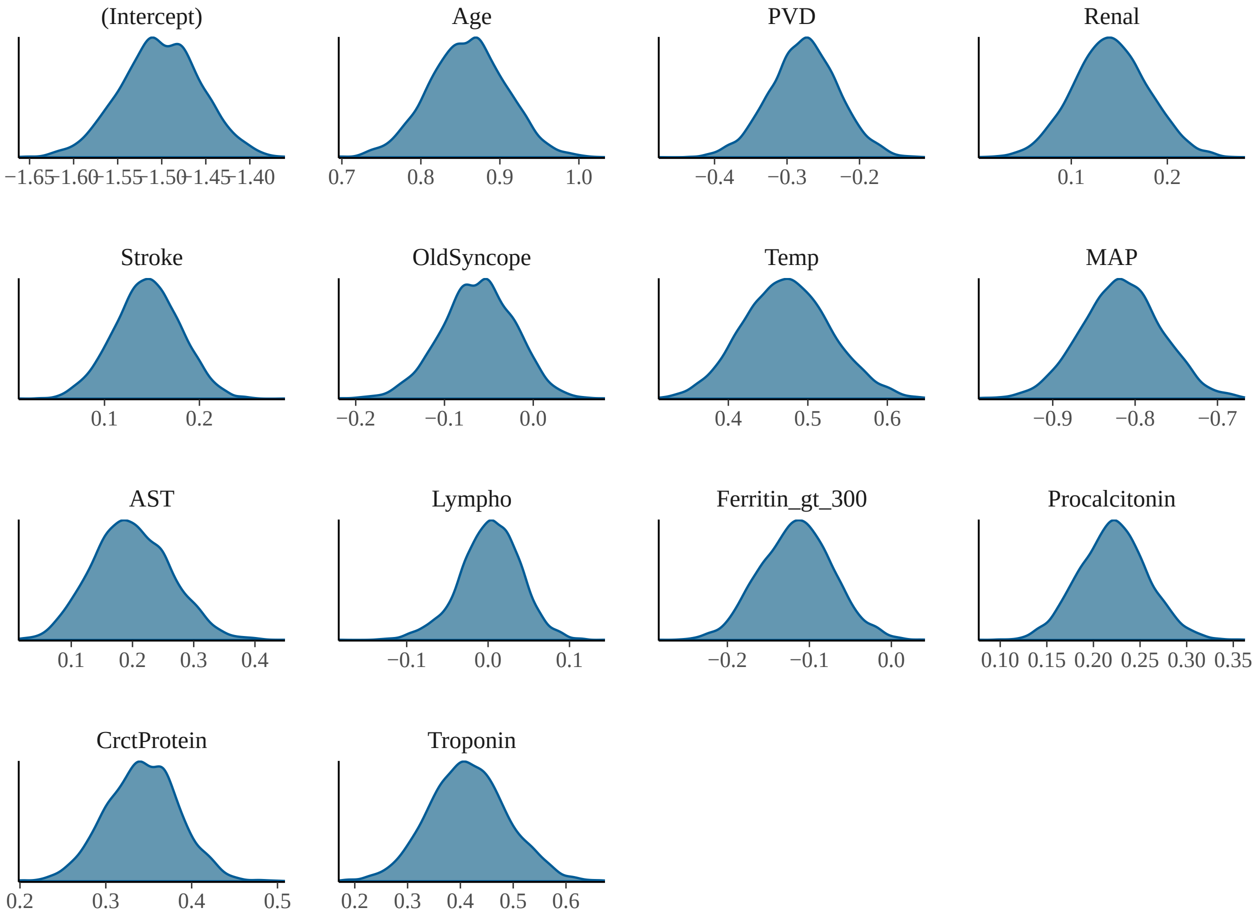

Figure 2: Scrunched Pairs Plot of Logistic Regression for Risk Factors

```
pp_check(fit_bayes, "dens_overlay")
```

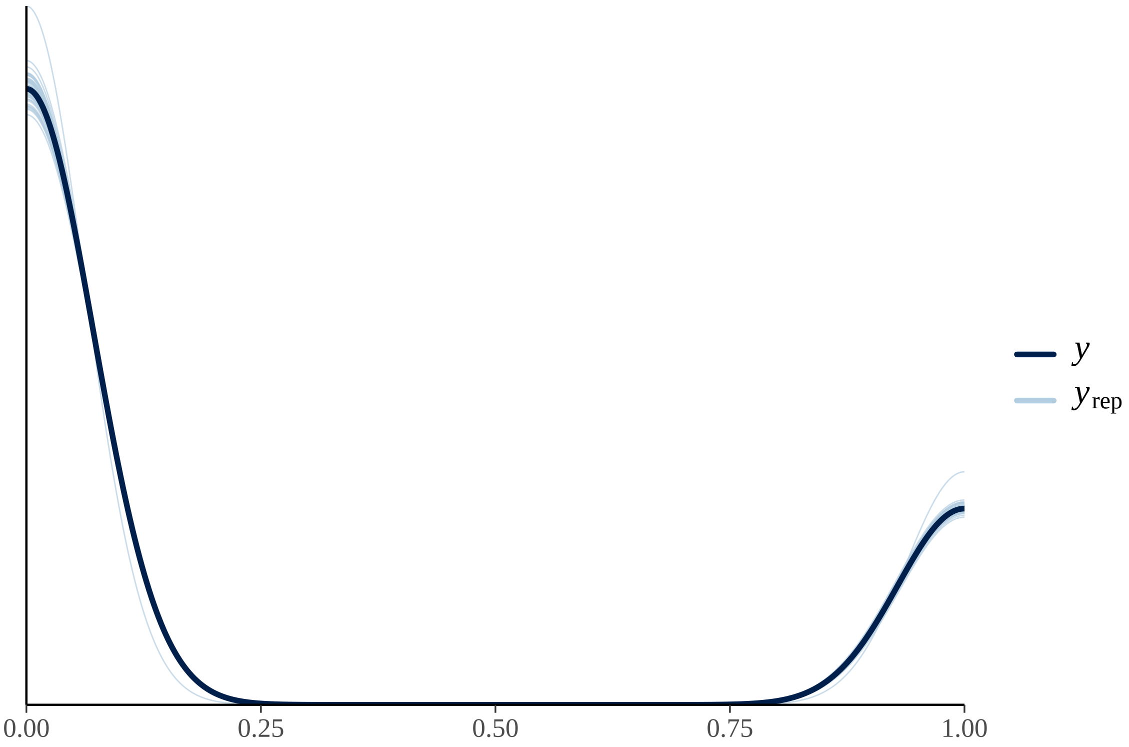

Figure 3: Factor-Analysis-Based Logistic Regression for risk factors associated with COVID-19 mortality

```
p_direction(fit_bayes)
```

```
Probability of Direction

Parameter       |     pd
------------------------
(Intercept)     |   100%
Age             |   100%
PVD             |   100%
Renal           |   100%
Stroke          |   100%
OldSyncope      | 94.03%
Temp            |   100%
MAP             |   100%
AST             |   100%
Lympho          | 54.47%
Ferritin_gt_300 | 99.60%
Procalcitonin   |   100%
CrctProtein     |   100%
Troponin        |   100%
```

Table 2: Probability of Direction for Logistic Regression Model

```
refmodel <- get_refmodel(fit1)
vs <- cv_varsel(refmodel, method='forward', cores=2)
plot(vs, stats = 'elpd')
```

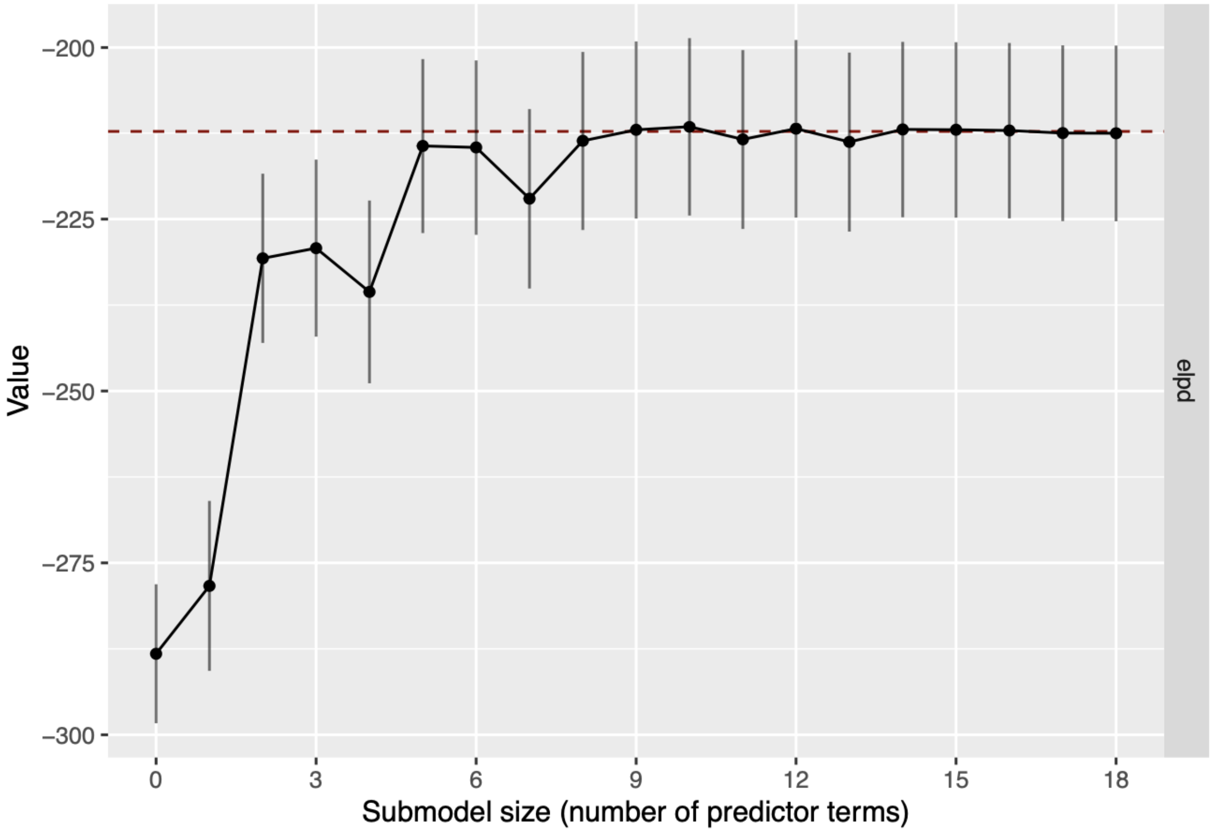

Figure 4: Expected Log Pointwise Predictive Density Validation

```r
plot(vs, stats = c('elpd', 'rmse'), deltas = TRUE)
```

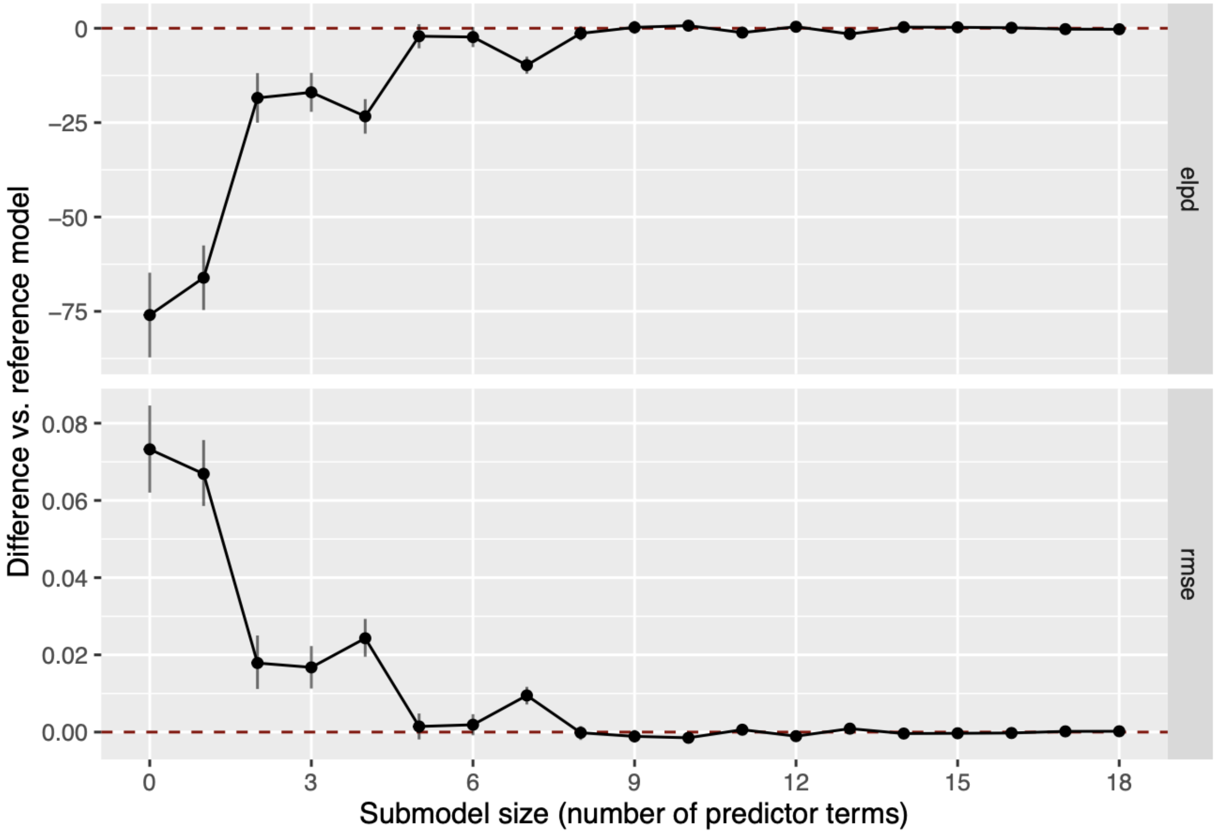

Figure 5: Expected Log Pointwise Predictive Density Validation Relative to the Full Model