HDAT9600 Final Team Assignment

Please see course timetable / 'Announcements' for submission deadline

Team 3: Zhenyu

insert date of completion here

Instructions

- This file (hdat9600assess4_teamX.Rmd) is the R Markdown document in which you need to complete your HDAT9600 final team assignment.
- You should rename the file by replacing 'X' with your team number.
- This assignment is assessed and will count for 30% of the total course marks.
- The assignment comprises two tasks. The first task will focus on building a model to PREDICT an outcome and the second task will focus on using a model to EXPLAIN the relationships between a variable of interest and the outcome.
- There is no word limit, nor limit on the length of your submitted Rmarkdown (i.e. this file) document. However, the **rendered** html document, were it to be printed, should be **no more** than about 10 A4 pages in length.
- Your html rendered document should include sufficient code chunks, output, and visual display as necessary to support your discursive commentary addressing each task. You may therefore wish to **suppress some of the contents** of your Rmarkdown file from appearing within the rendered document to ensure your output focuses on the most relevant results and discussion to address the questions.
- The rendered (html) document will be the submission primarily considered when awarding credit. However, the Rmarkdown file may also be considered if required to understand the analysis.
- **Excessively long reports will be penalised** by cessation of awarding of credit after roughly 10 pages of content.
- Feel free to delete the assignment instructions as you see fit to allow you some more room.
- All tasks are intended to be completed as a team. Have fun discussing the tasks and working together.

Don't hesitate to ask the course convenor for help via OpenLearning. The course instructor is happy to point you in the right direction and to make suggestions, but they won't, of course, complete your assessment for you!

Data for this assessment

The data used for this assessment consist of records from Intensive Care Unit (ICU) hospital stays in the USA. All patients were adults who were admitted for a wide variety of reasons. ICU stays of less than 48 hours have been excluded.

The source data for the assessment are data made freely available for the 2012 MIT PhysioNet/Computing for Cardiology Challenge. Details are provided here (https://physionet.org/challenge/2012/). Training Set A data have been used. The original data has been modified and assembled to suit the purpose of this assessment.

The dataframe consists of 120 variables, which are defined as follows:

Patient Descriptor Variables

- *RecordID:* a unique integer for each ICU stay
- *Age:* years

- *Gender:* male/female
- *Height:* cm
- *ICUType:* Coronary Care Unit; Cardiac Surgery Recovery Unit; Medical ICU; Surgical ICU
- *Length_of_stay:* The number of days between the patient's admission to the ICU and the end of hospitalisation
- *Survival:* The number of days between ICU admission and death for patients who died

Outcome Variables

- *in_hospital_death:* 0:survivor/1:died in-hospital **this is the outcome variable for Task 1: Logistic Regression**
- *Status:* True/False **this is the censoring variable for Task 2: Survival Analysis**
- *Days:* Length of survival (in days) **this is the survival time variable for Task 2: Survival Analysis**

Clinical Variables

Use the hyperlinks below to find out more about the clinical meaning of each variable. The first two clinical variables are summary scores that are used to assess patient condition and risk.

- SAPS-I score - Simplified Acute Physiological Score Le Gall et al., 1984 (http://www.ncbi.nlm.nih.gov/pubmed/6499483)
- SOFA score - Sequential Organ Failure Assessment Ferreira et al., 2001 (http://www.ncbi.nlm.nih.gov/pubmed/11594901)

The following 36 clinical measures were assessed at multiple timepoints during each patient's ICU stay. For each of the 36 clinical measures, you are given 3 summary variables: a) The minimum value during the first 24 hours in ICU (_min), b) The maximum value during the first 24 hours in ICU (_max), and c) The difference between the mean and the most extreme values during the first 24 hours in ICU (_diff). For example, for the clinical measure Cholesterol, these three variables are labelled 'Cholesterol_min', 'Cholesterol_max', and 'Cholesterol_diff'.

- Albumin (http://en.wikipedia.org/wiki/Human_serum_albumin) (g/dL)
- ALP (http://en.wikipedia.org/wiki/Alkaline_phosphatase) [Alkaline phosphatase (IU/L)]
- ALT (http://en.wikipedia.org/wiki/Alanine_transaminase) [Alanine transaminase (IU/L)]
- AST (http://en.wikipedia.org/wiki/Aspartate_transaminase) [Aspartate transaminase (IU/L)]
- Bilirubin (http://en.wikipedia.org/wiki/Bilirubin) (mg/dL)
- BUN (http://en.wikipedia.org/wiki/BUN) [Blood urea nitrogen (mg/dL)]
- Cholesterol (http://en.wikipedia.org/wiki/Cholesterol) (mg/dL)
- Creatinine (http://en.wikipedia.org/wiki/Serum_creatinine#Plasma_creatinine) [Serum creatinine (mg/dL)]
- DiasABP (http://en.wikipedia.org/wiki/Diastolic_blood_pressure) [Invasive diastolic arterial blood pressure (mmHg)]
- FiO2 (http://en.wikipedia.org/wiki/FIO2) [Fractional inspired O_2 (0-1)]
- GCS (http://en.wikipedia.org/wiki/Glasgow_coma_score) [Glasgow Coma Score (3-15)]
- Glucose (http://en.wikipedia.org/wiki/Serum_glucose) [Serum glucose (mg/dL)]
- HCO3 (http://en.wikipedia.org/wiki/Bicarbonate#Diagnostics) [Serum bicarbonate (mmol/L)]
- HCT (http://en.wikipedia.org/wiki/Hematocrit) [Hematocrit (%)]
- HR (http://en.wikipedia.org/wiki/Heart_rate) [Heart rate (bpm)]
- K (http://en.wikipedia.org/wiki/Hypokalemia) [Serum potassium (mEq/L)]
- Lactate (http://en.wikipedia.org/wiki/Lactic_acid) (mmol/L)
- Mg (http://en.wikipedia.org/wiki/Magnesium#Biological_role) [Serum magnesium (mmol/L)]
- MAP (http://en.wikipedia.org/wiki/Mean_arterial_pressure) [Invasive mean arterial blood pressure (mmHg)]
- MechVent (http://en.wikipedia.org/wiki/Mechanical_ventilation) [Mechanical ventilation respiration (0:false, or 1:true)]
- Na (http://en.wikipedia.org/wiki/Serum_sodium) [Serum sodium (mEq/L)]
- NIDiasABP (http://en.wikipedia.org/wiki/Diastolic_blood_pressure) [Non-invasive diastolic arterial blood pressure (mmHg)]

- NIMAP (http://en.wikipedia.org/wiki/Mean_arterial_pressure) [Non-invasive mean arterial blood pressure (mmHg)]
- NISysABP (http://en.wikipedia.org/wiki/Systolic_blood_pressure) [Non-invasive systolic arterial blood pressure (mmHg)]
- PaCO2 (http://en.wikipedia.org/wiki/Arterial_blood_gas) [partial pressure of arterial CO_2 (mmHg)]
- PaO2 (http://en.wikipedia.org/wiki/Arterial_blood_gas) [Partial pressure of arterial O_2 (mmHg)]
- pH (http://en.wikipedia.org/wiki/Arterial_blood_gas) [Arterial pH (0-14)]
- Platelets (http://en.wikipedia.org/wiki/Platelets) (cells/nL)
- RespRate (http://en.wikipedia.org/wiki/Respiratory_physiology) [Respiration rate (bpm)]
- SaO2 (http://en.wikipedia.org/wiki/Arterial_blood_gas) [O_2 saturation in hemoglobin (%)]
- SysABP (http://en.wikipedia.org/wiki/Systolic_blood_pressure) [Invasive systolic arterial blood pressure (mmHg)]
- Temp (http://en.wikipedia.org/wiki/Normal_human_body_temperature) [Temperature (°C)]
- TropI (http://en.wikipedia.org/wiki/Troponin) [Troponin-I (µg/L)]
- TropT (http://en.wikipedia.org/wiki/Troponin) [Troponin-T (µg/L)]
- Urine (http://en.wikipedia.org/wiki/Fluid_balance) [Urine output (mL)]
- WBC (http://en.wikipedia.org/wiki/Reference_ranges_for_blood_tests#Hematology) [White blood cell count (cells/nL)]
- Weight (kg)

Accessing the Data

There are two datasets provided. The data frames can be loaded with the following code:

```
mydat <- file.path("C:/Users/froze/OneDrive/Documents/2.1 UNSW bioinfo/HDAT9600/Assignment/Grou
p/HDAT9600_Assign4_GroupProject")
icu_patients_df0 <- readRDS(file.path(mydat,"icu_patients_df0.rds"))
icu_patients_df1 <- readRDS(file.path(mydat,"icu_patients_df1.rds"))

#summary(icu_patients_df1) #useful for initial checks
#str(icu_patients_df1) # Structure of the dataset
#glimpse(icu_patients_df1) #quickly see the entirety of the data frame's structure and less lik
ely to truncate
length(unique(icu_patients_df1$RecordID)) # Unique patient check
```

```
## [1] 2061
```

```
# Calculate the number of missing values for each column
missing_values <- colSums(is.na(icu_patients_df1))
# Remove the variables that do not have missing values
missing_values <- missing_values[missing_values > 0]
# Print the variables with their count of missing values
print(missing_values)
```

```
##          SAPS1       Survival   DiasABP_diff    DiasABP_max    DiasABP_min
##             96           1288            715            715            715
##         Height NIDiasABP_diff  NIDiasABP_max  NIDiasABP_min     NIMAP_diff
##            992            455            455            455            455
##      NIMAP_max      NIMAP_min  NISysABP_diff  NISysABP_max   NISysABP_min
##            455            455            453            453            453
##    SysABP_diff     SysABP_max     SysABP_min    Weight_diff     Weight_max
##            715            715            715            146            146
##     Weight_min
##            146
```

Note: icu_patients_df1 is an imputed (i.e. missing values are 'derived') version of icu_patients_df0. This assessment does not concern the methods used for imputation. You should use `icu_patients_df1` for all tasks unless it is specifically noted to use `icu_patients_df0`.

From the output and combined with reference papers, here are some variables that correspond to the SAPS-I and SOFA scores and could be considered important for the models: 1. **SAPS-I score**: The dataset includes SAPS1 as a variable, which is directly relevant. 2. **SOFA score**: The dataset includes SOFA as a variable, also directly relevant. 3. **Respiratory function**: PaO2/FiO2 ratio (part of SOFA), and Respiratory rate (part of SAPS-I). In the dataset, PaO2, FiO2 and RespRate variables could be related to respiratory function (not find MechVent variable). 4. **Coagulation**: Platelets count is directly mentioned in the dataset and SOFA. 5. **Liver function**: Bilirubin levels are included in both the dataset and the SOFA score. 6. **Cardiovascular system**: The presence of hypotension and administration of dopamine or dobutamine are factors in the SOFA score. In the dataset, variables like MAP, SysABP, and NIMAP are related to blood pressure and might be used as proxies. 7. **Central nervous system**: Glasgow Coma Score is present in both the SOFA and SAPS-I scores and directly available in the dataset (GCS variables). 8. **Renal function**: Creatinine levels and Urine output, which are in SOFA and indirectly in SAPS-I (blood urea), correspond to Creatinine and Urine variables in the data. 9. **Age**: This is explicitly mentioned in SAPS-I and is a known risk factor in ICU outcomes. The Age variable is available in the dataset.

Considering these variables for inclusion in the predictive models for Task 1 (Logistic Regression for in-hospital death) and Task 2 (Survival Analysis for length of survival) would be reasonable based on their clinical importance and representation in established severity scores.

Variables with Missing Data: - `SAPS1`: 96 missing values (4.66% missing) - This is a relatively small proportion of the dataset. Since SAPS-I is an important score for ICU prognosis, it might be worth imputing these values rather than excluding the variable. - `Survival`: 1,288 missing values (62.48% missing) - This is a significant amount of missing data. However, 'Survival' may represent censored data rather than simply missing information. In survival analysis, this would be the expected setup, where patients who did not die in the hospital are censored at the last follow-up time. - Blood pressure-related variables (`DiasABP_diff`, `DiasABP_max`, `DiasABP_min`, `NIDiasABP_diff`, `NIDiasABP_max`, `NIDiasABP_min`, `NIMAP_diff`, `NIMAP_max`, `NIMAP_min`, `NISysABP_diff`, `NISysABP_max`, `NISysABP_min`, `SysABP_diff`, `SysABP_max`, `SysABP_min`): All these have a high degree of missingness, ranging from 21.98% to 34.54%. Since these are many related variables with substantial missing data, we might consider/test the importance of blood pressure measurement in your analysis and whether it could be captured by fewer variables (maybe only MAP). - `Height`: 992 missing values (48.10% missing) - If height is not central to analysis and considering the high rate of missingness, it could be a candidate for exclusion. - Weight-related variables (`Weight_diff`, `Weight_max`, `Weight_min`): 146 missing values (7.08% missing) - This is a smaller proportion of missing data and might be worth imputing if weight is considered important in analysis.

values outside of plausible clinical ranges or logically incorrect - `SOFA` should not be negative Convert to integers and replace negative values with NA. - `Length_of_stay` should not be negative (they are all -1 and not +1 in data set) Convert to positive - `Weight` absurd weight differences and extremely high values

```
# Initial type conversions and handling of illogical variables
icu_patients_df1_conv <- icu_patients_df1 %>%
  mutate(
    SAPS1 = if_else(SAPS1 < 0, NA_real_, as.integer(SAPS1)),
    SOFA = if_else(SOFA < 0, NA_real_, as.integer(SOFA)),
    Length_of_stay = if_else(Length_of_stay < 0, abs(Length_of_stay), Length_of_stay),
    Age = as.integer(Age),
    GCS_max = as.integer(GCS_max),
    Gender = as.factor(Gender),
    ICUType = as.factor(ICUType),
    Status_Label = factor(Status, levels = c(FALSE, TRUE), labels = c("Survived", "Died")),
    LR_in_hospital_death = factor(in_hospital_death, levels = c(0, 1), labels = c("Survived",
"Died"))
  )


# Set a threshold for maximum acceptable proportion of missing data
missing_threshold <- 0.3
# Calculate the proportion of missing data for each variable
prop_missing <- colSums(is.na(icu_patients_df1_conv)) / nrow(icu_patients_df1_conv)
# Identify variables to exclude based on the threshold
vars_to_check <- setdiff(names(icu_patients_df1_conv), "Survival") # Exclude the 'Survival' var
iable from the evaluation for missing data
vars_to_exclude <- names(prop_missing[vars_to_check][prop_missing[vars_to_check] > missing_thre
shold])
print("Variables to be excluded due to high missing data:")
```

```
## [1] "Variables to be excluded due to high missing data:"
```

```
print(vars_to_exclude)
```

```
## [1] "DiasABP_diff" "DiasABP_max"  "DiasABP_min"  "Height"       "SysABP_diff"
## [6] "SysABP_max"   "SysABP_min"
```

```
# Identify variables to impute
vars_to_impute <- names(prop_missing[prop_missing <= missing_threshold & prop_missing > 0])
# Make a copy of the data frame to clean
icu_patients_df1_cleaned <- icu_patients_df1_conv
# Impute missing values for selected variables using median
for(var in vars_to_impute) {
  icu_patients_df1_cleaned[[var]][is.na(icu_patients_df1_cleaned[[var]])] <- median(icu_patient
s_df1_cleaned[[var]], na.rm = TRUE)
}
# Now exclude the variables with too much missing data from the cleaned dataframe
icu_patients_df1_cleaned <- icu_patients_df1_cleaned[ , !(names(icu_patients_df1_cleaned) %in%
vars_to_exclude)]

# Check if there are any remaining missing values
missing_values <- colSums(is.na(icu_patients_df1_cleaned))
missing_values <- missing_values[missing_values > 0]
print("Remaining missing values in the cleaned data:")
```

```
## [1] "Remaining missing values in the cleaned data:"
```

```
print(missing_values)
```

```
## Survival
##     1288
```

```
# Print the number of observations after cleaning
print(paste("Number of observations after cleaning:", nrow(icu_patients_df1_cleaned)))
```

```
## [1] "Number of observations after cleaning: 2061"
```

Aims of your project

You have two analytic goals in this assignment. 1. The first goal (Task 1) is to build a model to PREDICT in-hospital death. The context for this is that the hospital management wishes to use this model to prioritise the allocation of resources with the overarching aim of improving patient survival.
2. The second goal (Task 2) is to try to understand and EXPLAIN how survival varies by `Age`. How does survival change with increasing Age? Are age-related survival differences explained solely by differing clinical characteristics of patients?

Task 1

Part A: Exploratory data analysis

Conduct a brief EDA for the dataset `icu_patients_df1`. One of your goals for this EDA should be to identify a sub-set of variables (approx 15-20 variables) that *may* be useful predictors of survival and/or relevant to your analytical task. You may wish to firstly undertake a brief literature review to try to identify factors that may be important predictors of survival in an ICU. This doesn't need to be in-depth, but enough to get a basic idea of what the variables are measuring and which variables might, in theory, be most important.

For the 15-20 chosen variables, present enough information to: (i) explain why you selected these variables, based on a combination of your literature review and EDA, and (ii) Provide an overview of the variables and how they are related to `in-hospital death`. Present your summary using tables and/or plots with supporting commentary where relevant.

```
# Overall Demographics Summary
overall_demo <- summary(icu_patients_df1[c("Age", "Height", "Weight_max", "Weight_min", "SOFA", "SAPS1")])
kable(overall_demo, caption = "Overall Demographics Summary", format = "markdown")
```

Overall Demographics Summary

Age	Height	Weight_max	Weight_min	SOFA	SAPS1
Min. :16.00	Min. : 13.0	Min. : 34.60	Min. : 34.60	Min. :-1.000	Min. : 1.00
1st Qu.:52.00	1st Qu.:162.6	1st Qu.: 66.00	1st Qu.: 65.00	1st Qu.: 3.000	1st Qu.:11.00
Median :67.00	Median :170.2	Median : 80.00	Median : 77.70	Median : 6.000	Median :15.00

Age	Height	Weight_max	Weight_min	SOFA	SAPS1
Mean :64.41	Mean :170.0	Mean : 82.66	Mean : 80.86	Mean : 6.441	Mean :14.96
3rd Qu.:78.00	3rd Qu.:177.8	3rd Qu.: 94.55	3rd Qu.: 91.95	3rd Qu.: 9.000	3rd Qu.:19.00
Max. :90.00	Max. :426.7	Max. :230.00	Max. :230.00	Max. :22.000	Max. :34.00
NA	NA's :992	NA's :146	NA's :146	NA	NA's :96

```
# Split the dataset into survivors and non-survivors
survivors_df <- icu_patients_df1[icu_patients_df1$in_hospital_death == 0, ]
deceased_df <- icu_patients_df1[icu_patients_df1$in_hospital_death == 1, ]

# Summary for Survivors
survivors_demo <- summary(survivors_df[c("Age", "Weight_max", "Weight_min", "Length_of_stay",
"SOFA", "SAPS1")])
kable(survivors_demo, caption = "Demographics Summary for Survivors", format = "markdown")
```

Demographics Summary for Survivors

Age	Weight_max	Weight_min	Length_of_stay	SOFA	SAPS1
Min. :16.00	Min. : 34.60	Min. : 34.60	Min. : -1.00	Min. :-1.000	Min. : 1.00
1st Qu.:51.00	1st Qu.: 66.50	1st Qu.: 65.00	1st Qu.: 6.00	1st Qu.: 3.000	1st Qu.:11.00
Median :66.00	Median : 80.35	Median : 78.50	Median : 10.00	Median : 6.000	Median :15.00
Mean :63.29	Mean : 83.14	Mean : 81.24	Mean : 13.55	Mean : 6.074	Mean :14.49
3rd Qu.:78.00	3rd Qu.: 94.83	3rd Qu.: 92.00	3rd Qu.: 17.00	3rd Qu.: 9.000	3rd Qu.:18.00
Max. :90.00	Max. :230.00	Max. :230.00	Max. :154.00	Max. :18.000	Max. :34.00
NA	NA's :132	NA's :132	NA	NA	NA's :84

```
# Summary for Non-Survivors
deceased_demo <- summary(deceased_df[c("Age", "Weight_max", "Weight_min", "Length_of_stay", "Su
rvival", "SOFA", "SAPS1")])
kable(deceased_demo, caption = "Demographics Summary for Non-Survivors", format = "markdown")
```

Demographics Summary for Non-Survivors

Age	Weight_max	Weight_min	Length_of_stay	Survival	SOFA	SAPS1
Min. :22.00	Min. : 35.00	Min. : 35.00	Min. : 2.00	Min. : 1.00	Min. :-1.00	Min. : 3.00
1st Qu.:62.00	1st Qu.: 62.85	1st Qu.: 62.35	1st Qu.: 6.00	1st Qu.: 4.00	1st Qu.: 5.00	1st Qu.:14.00
Median :76.00	Median : 76.00	Median : 74.60	Median : 10.00	Median : 8.00	Median : 8.00	Median :18.00
Mean :71.05	Mean : 79.88	Mean : 78.65	Mean : 14.89	Mean : 11.85	Mean : 8.62	Mean :17.72
3rd Qu.:83.00	3rd Qu.: 93.15	3rd Qu.: 90.65	3rd Qu.: 17.00	3rd Qu.: 15.00	3rd Qu.:12.00	3rd Qu.:21.00

Age	Weight_max	Weight_min	Length_of_stay	Survival	SOFA	SAPS1
Max. :90.00	Max. :165.00	Max. :165.00	Max. :146.00	Max. :145.00	Max. :22.00	Max. :33.00
NA	NA's :14	NA's :14	NA	NA	NA	NA's :12

```
# Create a table for Gender Distribution, Number of in hospital Deaths, and ICU Types
# Calculate the required numbers and percentages
patients_count <- nrow(icu_patients_df1)
gender_counts <- table(icu_patients_df1$Gender)
gender_percentages <- prop.table(gender_counts) * 100
death_counts <- sum(icu_patients_df1$in_hospital_death, na.rm = TRUE)
death_percentage <- (death_counts / patients_count) * 100
icu_type_counts <- table(icu_patients_df1$ICUType)
icu_type_percentages <- prop.table(icu_type_counts) * 100

# Create a data frame with the calculated values
demographics_table <- data.frame(
  Category = c("Number_of_Patients",
               "Gender: Male",
               "Gender: Female",
               "in_hospital_death",
               "ICU Types: Coronary Care Unit",
               "ICU Types: Cardiac Surgery Recovery Unit",
               "ICU Types: Medical ICU",
               "ICU Types: Surgical ICU"),
  Number = c(patients_count,
             gender_counts["Male"],
             gender_counts["Female"],
             death_counts,
             icu_type_counts["Coronary Care Unit"],
             icu_type_counts["Cardiac Surgery Recovery Unit"],
             icu_type_counts["Medical ICU"],
             icu_type_counts["Surgical ICU"]),
  Percentage = c(NA,  # Placeholder for total patients percentage
                 gender_percentages["Male"],
                 gender_percentages["Female"],
                 death_percentage,
                 icu_type_percentages["Coronary Care Unit"],
                 icu_type_percentages["Cardiac Surgery Recovery Unit"],
                 icu_type_percentages["Medical ICU"],
                 icu_type_percentages["Surgical ICU"])
)

# Use kable to print the table
kable(demographics_table, caption = "Demographics Summary 2", align = c('l', 'r', 'r'), format
= "markdown")
```

Demographics Summary 2

Category	Number	Percentage
Number_of_Patients	2061	NA
Gender: Male	1148	55.70112

Category	Number	Percentage
Gender: Female	913	44.29888
in_hospital_death	297	14.41048
ICU Types: Coronary Care Unit	297	14.41048
ICU Types: Cardiac Surgery Recovery Unit	448	21.73702
ICU Types: Medical ICU	788	38.23387
ICU Types: Surgical ICU	528	25.61863

Overall Demographics Summary: - Patients range in age from 16 to 90, with a median age of 67, suggesting a predominantly older patient population. - Height varies widely, with some extreme values (min 13 cm, max 426.7 cm), indicating possible data entry errors, given the maximum height is biologically implausible. - The max weight of patients (Weight_max) and minimum weight (Weight_min) during the stay are similar in range, indicating relatively stable weight during ICU stay or uniformity in recording practices. - SOFA scores, which range from -1 to 22, with a mean of 6.441, indicate varying degrees of organ failure severity among patients. - SAPS1 scores also show a wide range, but there are missing values for SAPS1 that may need addressing for complete analysis.

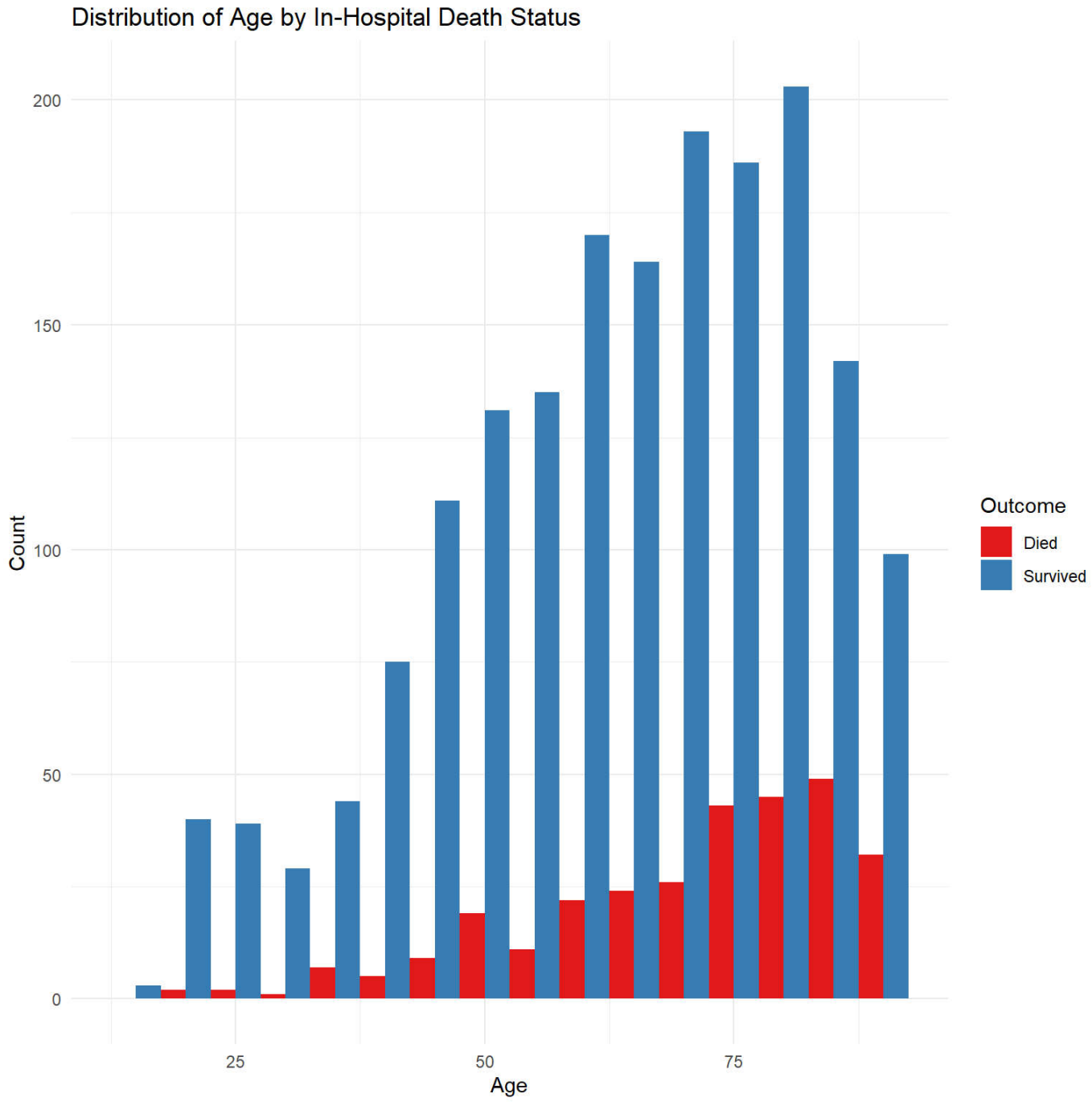
Demographics Summary for Survivors and Non-Survivors: - Survivors have a slightly lower mean age (63.29) compared to the overall (64.41) and non-survivor (71.05) groups, and the length of stay varies from -1 (which could be an error) to 154 days. - A small number of survivors have maximum and minimum weights at the upper limit of 230 kg, which may be outliers or may represent a small subset of critically ill patients with a higher body weight. - The median and mean weights for non-survivors are slightly lower than for survivors, which could reflect severe illness trajectories that may affects body weight. However, in general, the distribution of weight data does not indicate dramatic differences between survivors and non-survivors. - The median and average SOFA and SAPS1 scores among survivors is lower than non-survivors, which aligns with the understanding that higher scores are associated with more severe organ dysfunction and a higher likelihood of mortality.

Demographics Table Summary 2: - There were 2061 patients in total, with a higher proportion of males (approximately 55.70%) compared to females. - The proportion of patients who died in the hospital was about 14.41%. - The distribution of patients across ICU types shows the largest number in the Medical ICU (38.23%) and the least in the Coronary Care Unit (14.41%).

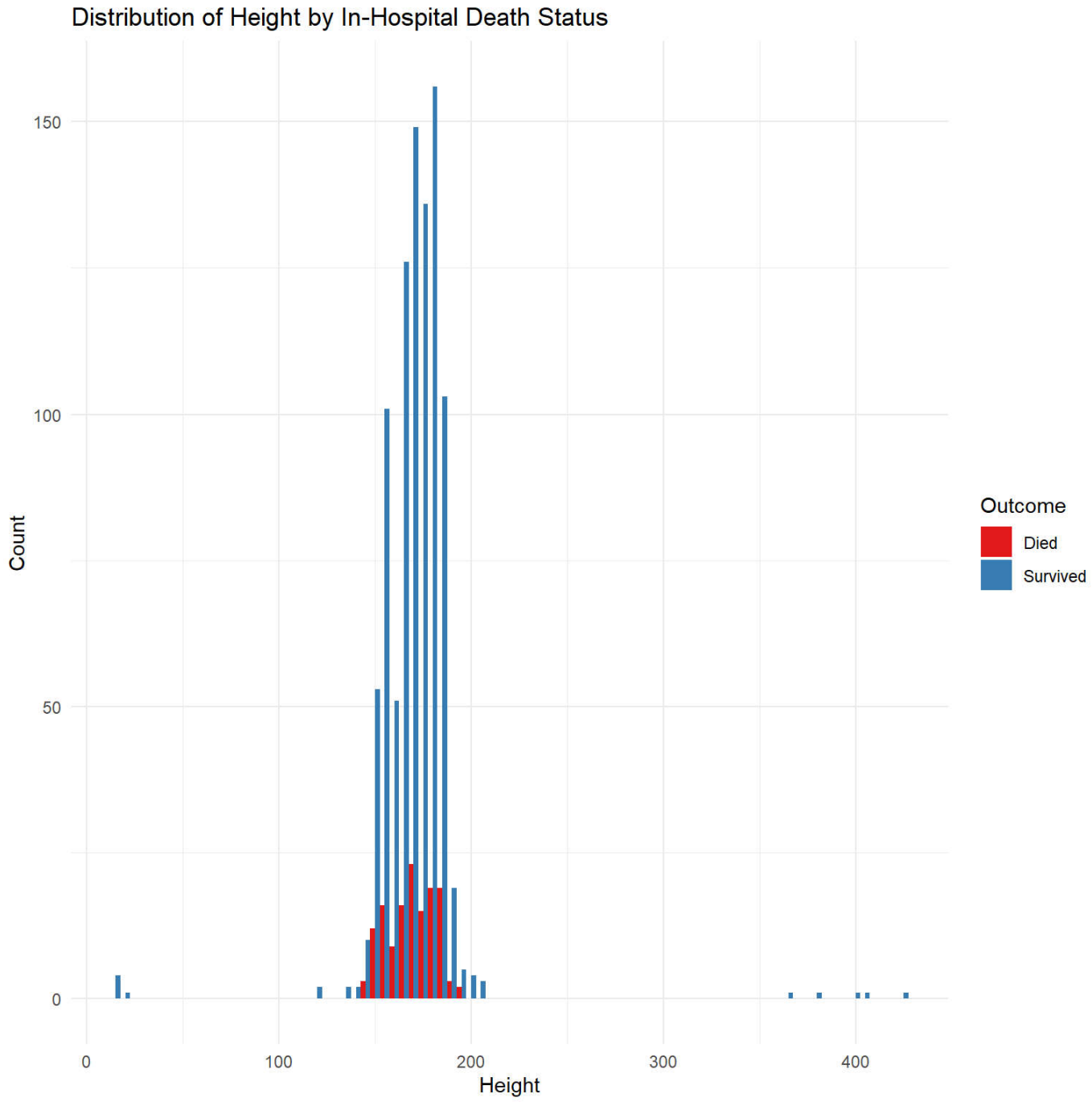
```
# Add a descriptive label for in-hospital death status
icu_patients_df1$Status_Label <- ifelse(icu_patients_df1$in_hospital_death == 0, "Survived", "D
ied")
icu_patients_df1$Status_Label <- as.factor(icu_patients_df1$Status_Label)

# Create a list of demographic variables to plot
demographic_vars <- c("Age", "Height", "Weight_max", "Weight_min", "Length_of_stay", "Surviva
l")

# Create a series of plots for each demographic variable
plots <- lapply(demographic_vars, function(var) {
  ggplot(icu_patients_df1, aes_string(x = var, fill = "Status_Label")) +
    geom_histogram(position = "dodge", binwidth = 5, na.rm = TRUE) +
    labs(x = var, y = "Count") +
    scale_fill_brewer(palette = "Set1", name = "Outcome") +
    theme_minimal() +
    ggtitle(paste("Distribution of", var, "by In-Hospital Death Status"))
})
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

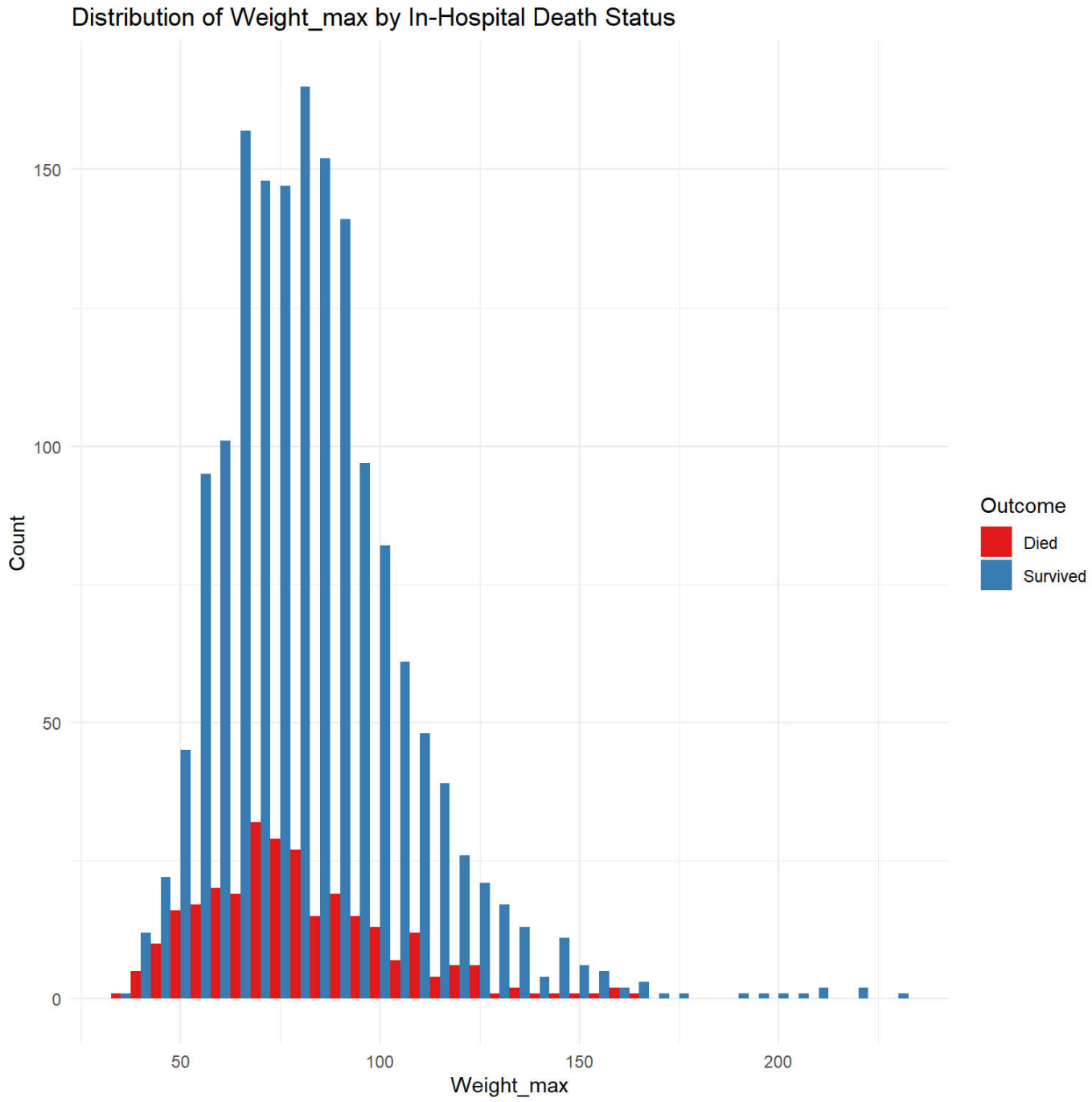
```
# Print the plots
print(plots[[1]]) # For Age
```



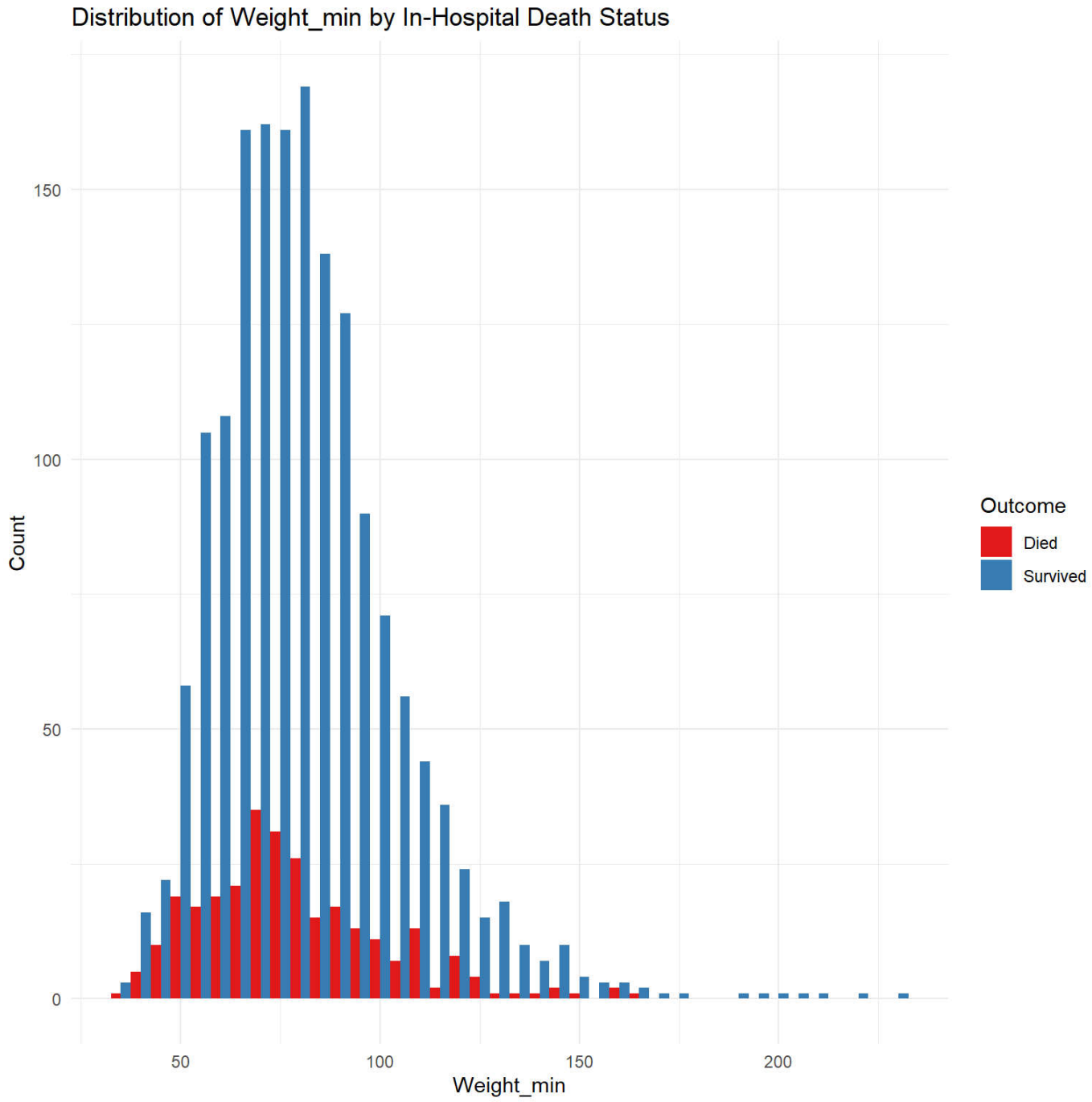

```
print(plots[[2]]) # For Height
```



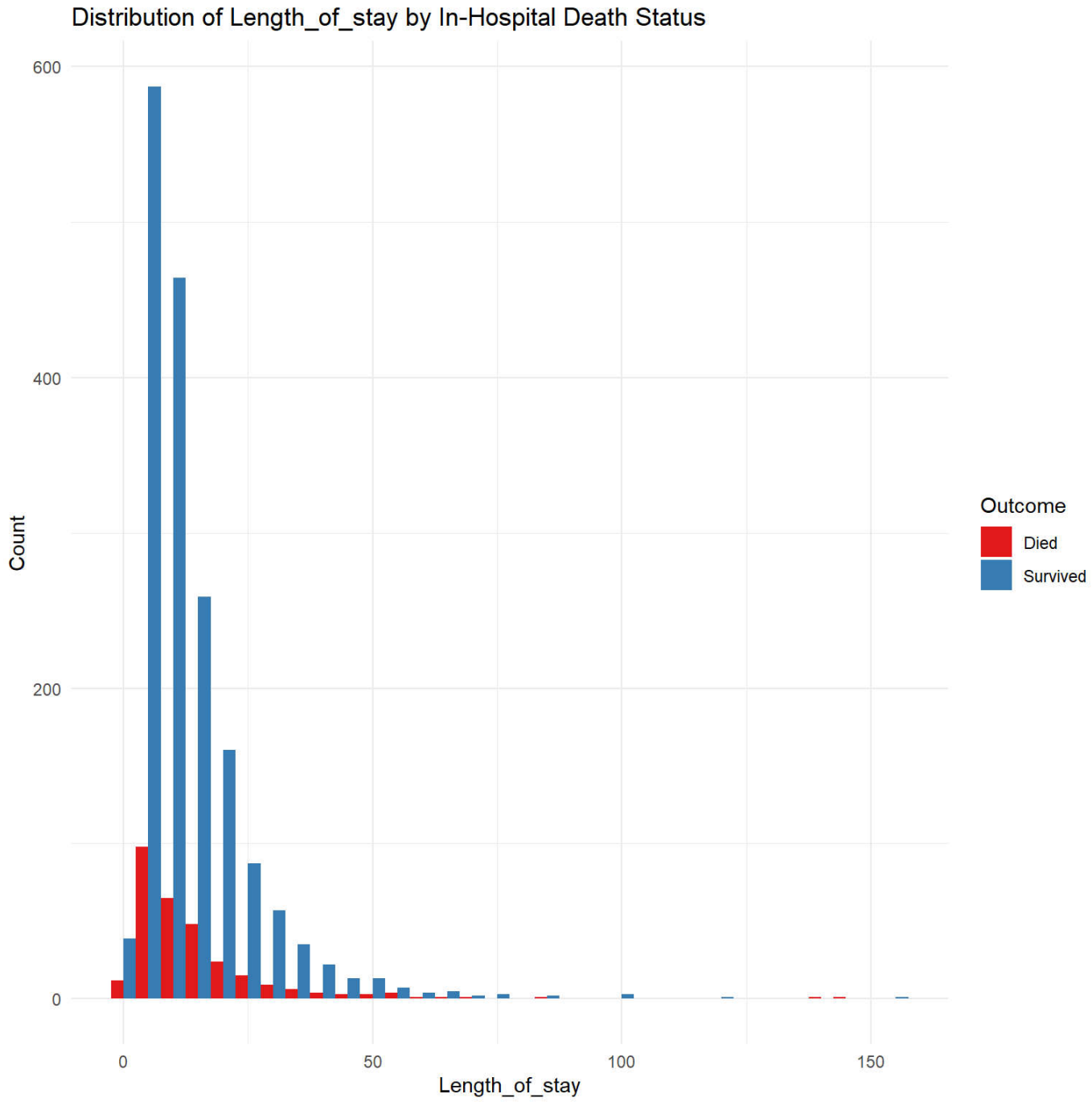

```
print(plots[[3]]) # For Weight_max
```



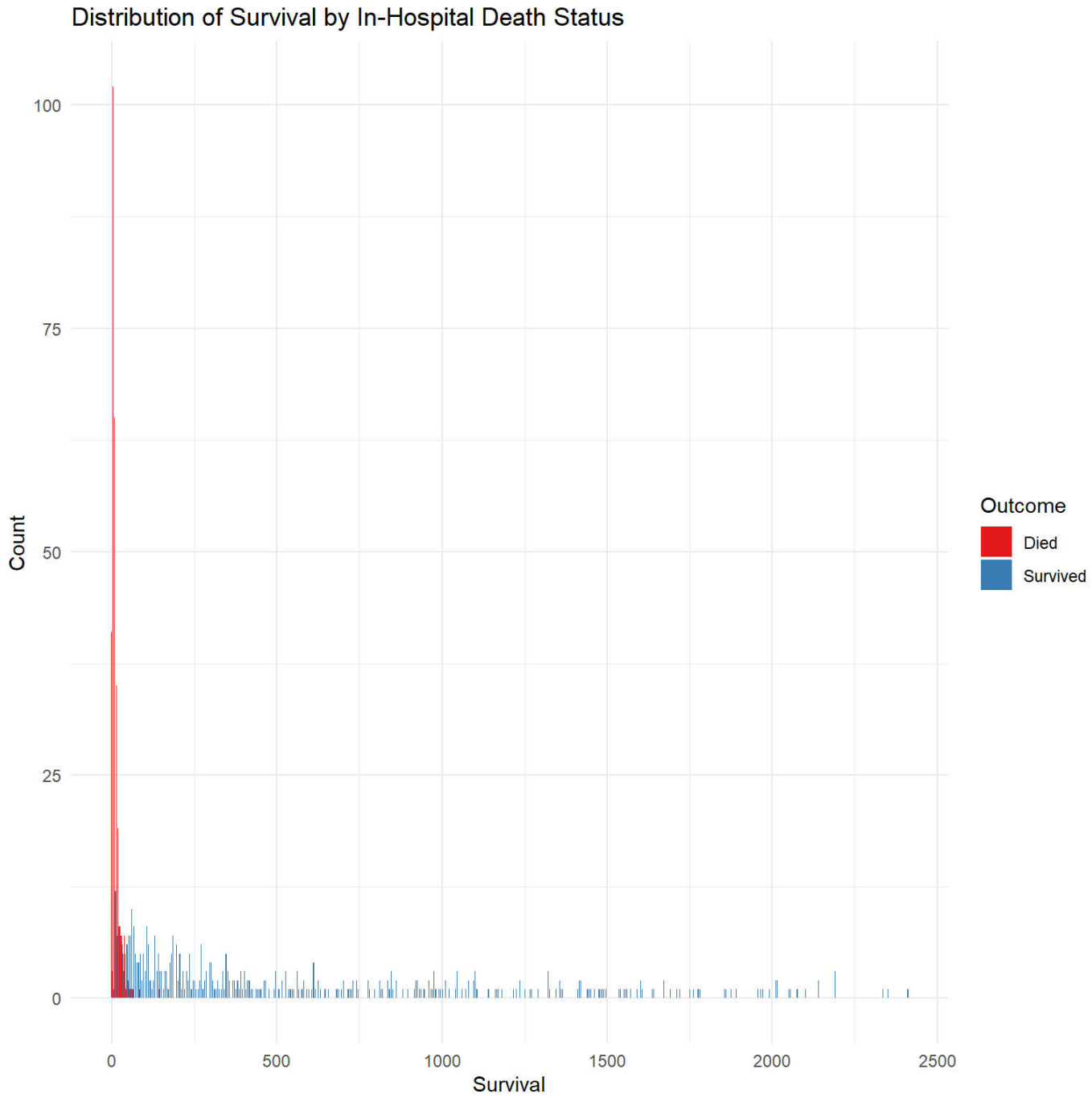

```
print(plots[[4]]) # For Weight_min
```




```
print(plots[[5]]) # For Length_of_stay
```




```
print(plots[[6]]) # For Survival
```

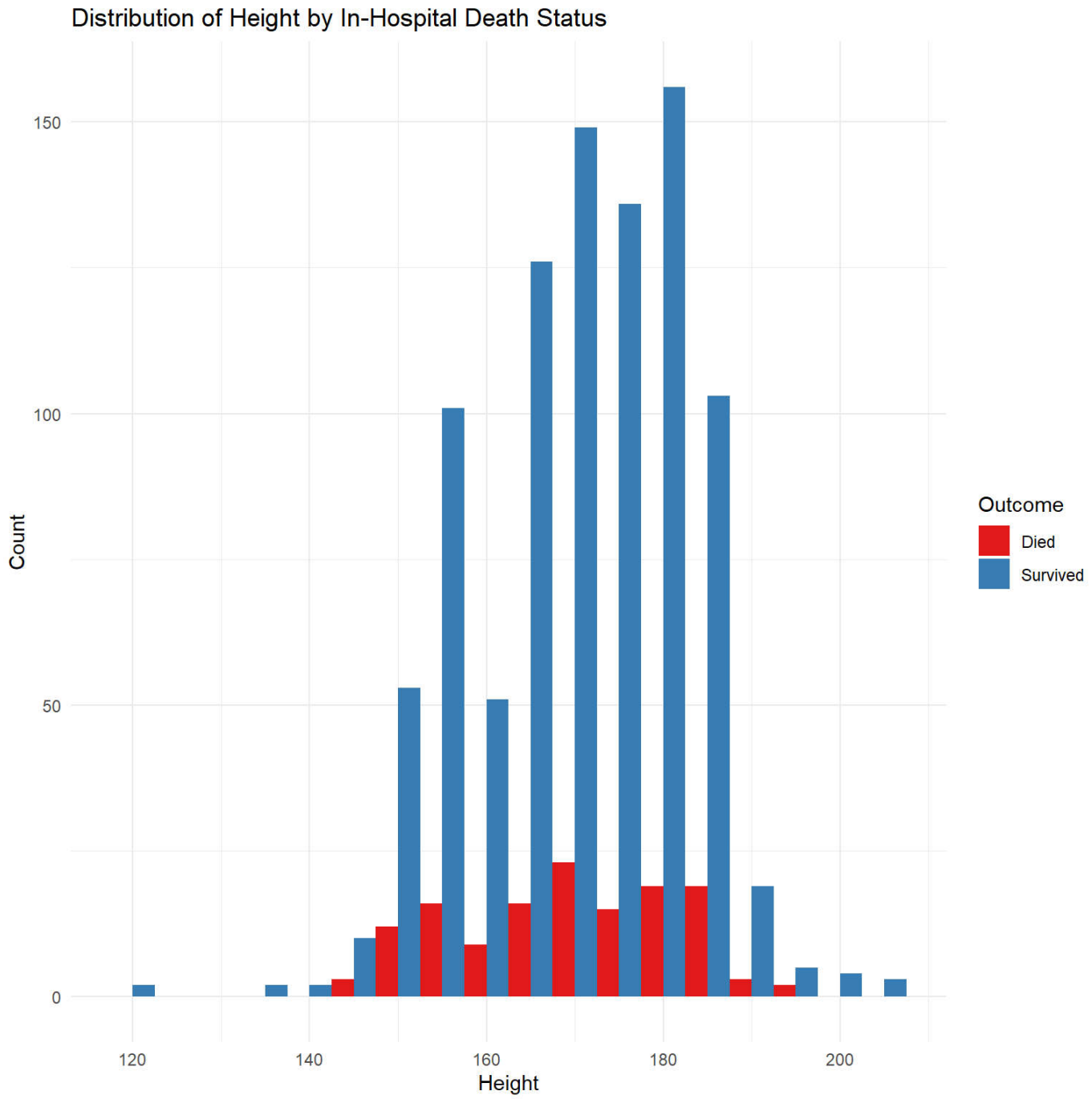


For the `Height` which < 17.8 or > 365 should be error on data collection or recording

```
# Correct the 'Height' variable by setting values outside the range 50 to 250 to NA
icu_patients_df1_h <- icu_patients_df1
icu_patients_df1_h$Height <- ifelse(icu_patients_df1_h$Height < 50 | icu_patients_df1_h$Height
> 250, NA, icu_patients_df1_h$Height)

# Recheck the summary for Height in the new dataframe and reprint the plot
summary(icu_patients_df1_h$Height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   121.9   162.6   170.2   169.6   177.8   205.7    1002
```

```
plots2 <- lapply(demographic_vars, function(var) {
  ggplot(icu_patients_df1_h, aes_string(x = var, fill = "Status_Label")) +
    geom_histogram(position = "dodge", binwidth = 5, na.rm = TRUE) +
    labs(x = var, y = "Count") +
    scale_fill_brewer(palette = "Set1", name = "Outcome") +
    theme_minimal() +
    ggtitle(paste("Distribution of", var, "by In-Hospital Death Status"))
})

print(plots2[[2]])
```



Histograms of the distribution of patient demographics - Age Distribution: There is a noticeable increase in the count of younger patients who survived compared to those who did not, with survival rates decreasing as age increases. Particularly, the survival rate is significantly lower for patients aged above 70. - Height Distribution: The distribution of height shows a fairly normal range with most individuals between 150 and 200 cm. A few outliers below 150 cm may require verification for accuracy. The survival seems independent of height as both survived and died groups are similarly distributed. - Maximum & Minimum Weight Distribution: The weight of patients is concentrated around 50 to 100 kg. This indicates that extreme weights are not

common among patients. The survival seems not too much dependent of the patient's weight. - Length of Stay: The majority of patients had a shorter length of stay in the ICU. The survival seems independent of the number of days between the patient's admission to the ICU and the end of hospitalisation as both survived and died groups are similarly distributed. - Survival Days: The survival days histogram is heavily skewed, with most patients surviving fewer days. The high number of patients with 0 survival days may indicate a significant number of deaths occurring shortly after admission to the ICU.

Part B: Predictive logistic model

Your aim is to build a model to PREDICT in-hospital death. In this task, you are required to develop a logistic regression model using the `icu_patients_df1` data set which adequately **predicts** the `in_hospital_death` variable as the outcome and utilises predictor variables from your chosen subset. You should fit a series of models, evaluating each one, before you present your final model. Your final model should **not** include all the predictor variables, just a small selection of them, which you have selected based on statistical significance and/or background knowledge. Remember, your aim is prediction, but you should also consider generalisability of the model, so we are aiming for parsimony. Aim for between five and ten predictor variables (slightly more or fewer is OK). You should assess each model you consider for goodness of fit and other relevant statistics to help you choose between them. For your final model, present a set of relevant diagnostic statistics and/or charts and comment on them.

Finally, re-fit your final model to the unimputed data frame (`icu_patients_df0.rds`) and comment on any differences you find compared to the same model fitted to the imputed data.

Create your response to task 1 here, as a mixture of embedded (`knitr`) R code and any resulting outputs, and explanatory or commentary text. Add code chunks as you see fit and choose whether you wish for the code and or results to be displayed in the final html document.

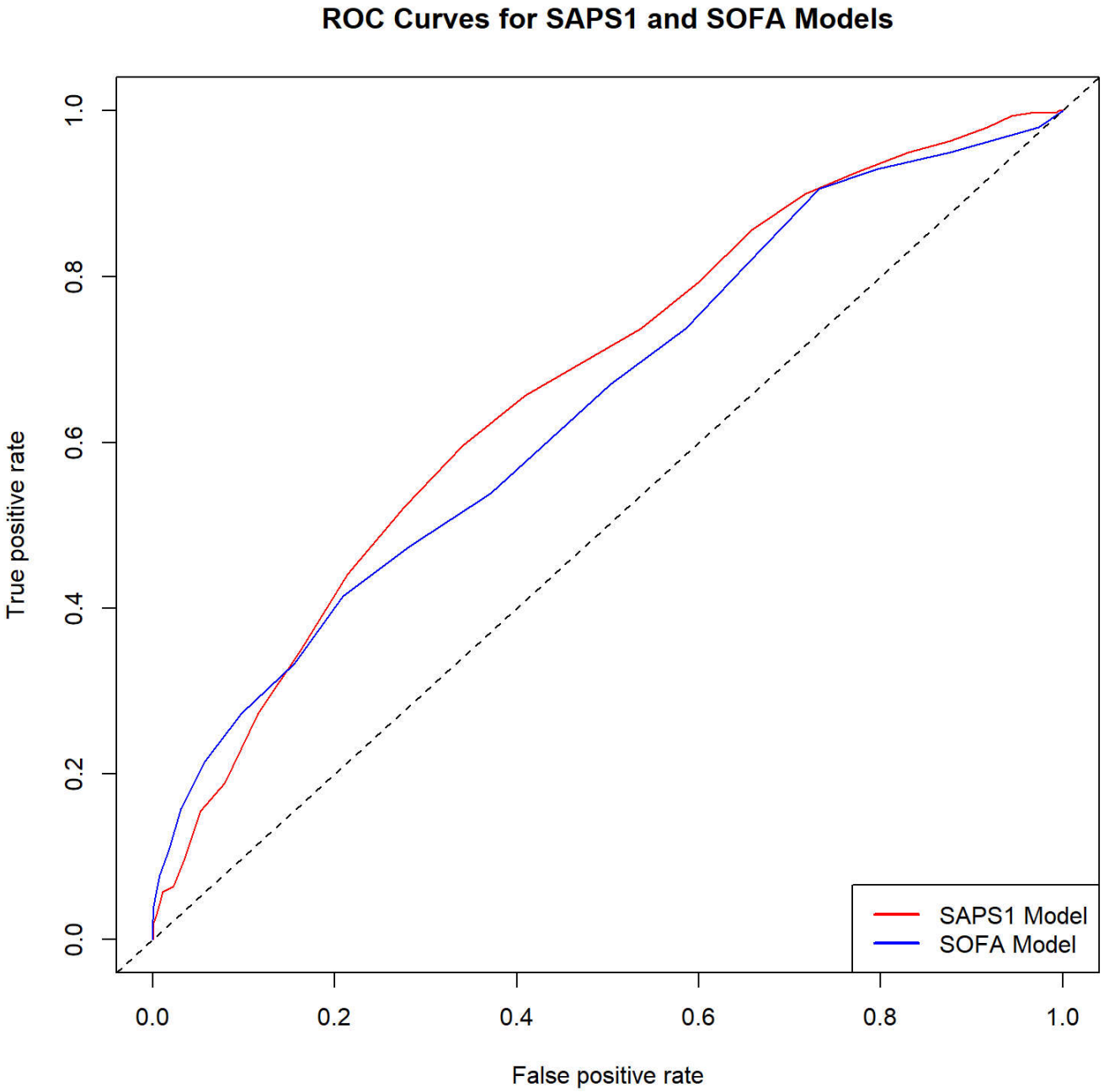
```
# Logistic regression to see the effect of SAPS1 and SOFA scores on in-hospital death
model_saps1 <- glm(in_hospital_death ~ SAPS1, data = icu_patients_df1_cleaned, family = binomia
l)
model_sofa <- glm(in_hospital_death ~ SOFA, data = icu_patients_df1_cleaned, family = binomial)

# Summary of logistic regression models
summary(model_saps1)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ SAPS1, family = binomial, data = icu_patients_df1_cleaned)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.80084    0.23648 -16.073   <2e-16 ***
## SAPS1        0.12570    0.01333   9.428   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1603.2  on 2059  degrees of freedom
## AIC: 1607.2
##
## Number of Fisher Scoring iterations: 5
```

```
summary(model_sofa)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ SOFA, family = binomial, data = icu_patients_df1_cleaned)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.88800    0.14713 -19.628   <2e-16 ***
## SOFA         0.14736    0.01601   9.207   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1611.1  on 2059  degrees of freedom
## AIC: 1615.1
##
## Number of Fisher Scoring iterations: 5
```

```
# Predicted probabilities for the models
pred_saps1 <- predict(model_saps1, type = "response")
pred_sofa <- predict(model_sofa, type = "response")

# Create prediction objects for ROC analysis
pred_obj_saps1 <- prediction(pred_saps1, icu_patients_df1_cleaned$in_hospital_death)
pred_obj_sofa <- prediction(pred_sofa, icu_patients_df1_cleaned$in_hospital_death)

# Create performance objects for ROC analysis
perf_saps1 <- performance(pred_obj_saps1, "tpr", "fpr")
perf_sofa <- performance(pred_obj_sofa, "tpr", "fpr")

# Plot ROC curves
plot(perf_saps1, col = "red", main = "ROC Curves for SAPS1 and SOFA Models")
plot(perf_sofa, col = "blue", add = TRUE)
abline(a = 0, b = 1, lty = 2)

# Add a legend
legend("bottomright", legend = c("SAPS1 Model", "SOFA Model"), col = c("red", "blue"), lwd = 2)
```




```
# Calculate AUC for SAPS1 model
auc_saps1 <- performance(pred_obj_saps1, measure = "auc")
auc_saps1_value <- auc_saps1@y.values[[1]]
cat("AUC for SAPS1 Model:", auc_saps1_value, "\n")
```

```
## AUC for SAPS1 Model: 0.6689533
```

```
# Calculate AUC for SOFA model
auc_sofa <- performance(pred_obj_sofa, measure = "auc")
auc_sofa_value <- auc_sofa@y.values[[1]]
cat("AUC for SOFA Model:", auc_sofa_value, "\n")
```

```
## AUC for SOFA Model: 0.6453499
```

These models provide a baseline for what can be expected when using SAPS1 and SOFA scores to predict in-hospital mortality For the **SAPS1 model**: - The coefficient for SAPS1 is 0.12558 with a p-value < 2e-16, indicating a statistically significant association with in-hospital death. The positive sign indicates that higher SAPS1 scores are associated with an increased probability of in-hospital death. - The model has an AIC of 1534.8, which can be used to compare the relative quality of statistical models for a given dataset.

For the **SOFA model**: - The coefficient for SOFA is 0.13540 with a p-value < 2e-16, which is also statistically significant. Similar to SAPS1, a higher SOFA score is associated with a higher probability of in-hospital death. - The AIC for this model is slightly higher at 1555, suggesting that the SAPS1 model might fit the data slightly better when considering only these scores as predictors.

residual deviance: The smaller the residual deviance, the better the model fits. - SAPS1 model has 1530.8 residual deviance on 1963 degrees of freedom - SOFA model has 1551 residual deviance on 1963 degrees of freedom

ROC analysis: The closer the ROC curve is to the top left corner, the higher the overall accuracy of the test. In this case, the SAPS1 model's ROC curve (in red) is closer to the top left corner than the SOFA model's curve (in blue), showing it performs better in the dataset for predicting the outcome.In addition, the SAPS1 model has a higher area under the curve (AUC), confirming its stronger predictive performance compared to the SOFA model in the dataset. - The SAPS1 model has an AUC of 0.6733918, which suggests it has a fair predictive power. Generally, an AUC close to 0.7 is considered acceptable, but there's room for improvement. - The SOFA model has an AUC of 0.6392481, which is slightly lower than SAPS1, indicating it is less predictive of in-hospital mortality compared to the SAPS1 model in the dataset.

Alternative variables From the previous outputs and reference papers, here are some variables that be considered important for the Regression models: 1. **SAPS-I score**: The dataset includes SAPS1 as a variable, which is directly relevant 2. **SOFA score**: The dataset includes SOFA as a variable, also directly relevant 3. **Respiratory function**: PaO2, FiO2 and RespRate 4. **Coagulation & blood**: Platelets, pH, Glucose 5. **Liver function**: Bilirubin, Albumin, ALP, ALT, AST 6. **Cardiovascular system**: HR, MAP, TroponinI, TroponinT, Lactate 7. **Central nervous system**:GCS 8. **Renal function**: Creatinine, Urine 9. **immune system**: Temp, WBC 10.**Age**: Age

Building separate logistic regression models for each system (like respiratory, coagulation & blood, cardiovascular, etc.) can be a good approach to understand which variables within each system are most predictive of in-hospital death. Here's a structured approach to building these models: 1. Respiratory Function Model: Use variables like PaO2_max, FiO2_max, and RespRate_max to reflect the most extreme states that may have critical implications for patient outcomes. 2. Coagulation & Blood Model: Platelets_max, pH_min, pH_max, and Glucose_max could be used to capture the most critical levels that may reflect acute events or stress responses. 3. Liver Function Model: Bilirubin_max, Albumin_min, ALP_max, ALT_max, and AST_max might be chosen to reflect liver function and injury. 4. Cardiovascular System Model: HR_max, MAP_max, TroponinI_max, and TroponinT_max, Lactate_max, Cholesterol_max can be indicators of cardiovascular stability and myocardial injury. 5. Central Nervous System Model: The GCS_min could be used as it represents the lowest level of consciousness. 6. Renal Function Model: Creatinine_max and Urine_min could be selected to reflect renal impairment or failure. 7. Immune System Model: Temp_max and WBC_max could reflect the highest levels of inflammatory response. 8. General Model: Incorporate variables like Age, Gender, weight, ICUType are broadly representative of patients' status.

Once these models are built 1. Could compare them to determine which system's variables are the strongest predictors. 2. Could build a combined model with the best predictors from each system.

```
# Respiratory Function Model
resp_model <- glm(in_hospital_death ~ PaO2_max + FiO2_max + RespRate_max,
                  family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(resp_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ PaO2_max + FiO2_max + RespRate_max,
##     family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6154825  0.2998626  -8.722  < 2e-16 ***
## PaO2_max     -0.0020090  0.0005988  -3.355 0.000794 ***
## FiO2_max      0.2986832  0.2748118   1.087 0.277097
## RespRate_max  0.0344080  0.0075397   4.564 5.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1666.1  on 2057  degrees of freedom
## AIC: 1674.1
##
## Number of Fisher Scoring iterations: 4
```

```
# Coagulation & Blood Model
coag_model <- glm(in_hospital_death ~ Platelets_max + pH_min + pH_max + Glucose_max,
                  family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(coag_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Platelets_max + pH_min + pH_max +
##     Glucose_max, family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   18.3259500  7.7186996   2.374 0.017586 *
## Platelets_max -0.0012617  0.0006031  -2.092 0.036433 *
## pH_min        -2.5456808  0.7066192  -3.603 0.000315 ***
## pH_max        -0.2083017  0.9912664  -0.210 0.833561
## Glucose_max    0.0020872  0.0005909   3.532 0.000412 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1661.4  on 2056  degrees of freedom
## AIC: 1671.4
##
## Number of Fisher Scoring iterations: 4
```

```
# Liver Function Model
liver_model <- glm(in_hospital_death ~ Bilirubin_max + Albumin_min + ALP_max + ALT_max + AST_ma
x,
                   family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(liver_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Bilirubin_max + Albumin_min +
##     ALP_max + ALT_max + AST_max, family = binomial(link = "logit"),
##     data = icu_patients_df1_cleaned)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.7763010  0.3203548  -2.423  0.01538 *
## Bilirubin_max  0.0466547  0.0118039   3.952 7.73e-05 ***
## Albumin_min   -0.4192390  0.1019020  -4.114 3.89e-05 ***
## ALP_max        0.0007561  0.0006333   1.194  0.23251
## ALT_max       -0.0006480  0.0003221  -2.012  0.04426 *
## AST_max        0.0006037  0.0002016   2.994  0.00275 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1636.7  on 2055  degrees of freedom
## AIC: 1648.7
##
## Number of Fisher Scoring iterations: 4
```

```
# Cardiovascular System Model
cardio_model <- glm(in_hospital_death ~ HR_max + MAP_max + TroponinI_max + TroponinT_max + Lact
ate_max +Cholesterol_max,
                    family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(cardio_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ HR_max + MAP_max + TroponinI_max +
##     TroponinT_max + Lactate_max + Cholesterol_max, family = binomial(link = "logit"),
##     data = icu_patients_df1_cleaned)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.6297909  0.4239103  -3.845 0.000121 ***
## HR_max           0.0049931  0.0028614   1.745 0.080995 .
## MAP_max          0.0004522  0.0019161   0.236 0.813454
## TroponinI_max   -0.0062793  0.0059501  -1.055 0.291271
## TroponinT_max    0.0542017  0.0284773   1.903 0.056998 .
## Lactate_max      0.1403429  0.0272981   5.141 2.73e-07 ***
## Cholesterol_max -0.0076152  0.0016589  -4.591 4.42e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1641.2  on 2054  degrees of freedom
## AIC: 1655.2
##
## Number of Fisher Scoring iterations: 4
```

```
# Central Nervous System Model
cns_model <- glm(in_hospital_death ~ GCS_min,
                 family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(cns_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ GCS_min, family = binomial(link = "logit"),
##     data = icu_patients_df1_cleaned)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.40261    0.12298 -11.405  < 2e-16 ***
## GCS_min     -0.04514    0.01317  -3.426 0.000612 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1687.7  on 2059  degrees of freedom
## AIC: 1691.7
##
## Number of Fisher Scoring iterations: 4
```

```
# Renal Function Model
renal_model <- glm(in_hospital_death ~ Creatinine_max + Urine_min,
                   family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(renal_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Creatinine_max + Urine_min,
##     family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.884295   0.095766 -19.676  < 2e-16 ***
## Creatinine_max  0.157655   0.031742   4.967 6.81e-07 ***
## Urine_min      -0.005531   0.001732  -3.193  0.00141 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1660.0  on 2058  degrees of freedom
## AIC: 1666
##
## Number of Fisher Scoring iterations: 5
```

```
# Immune System Model
immune_model <- glm(in_hospital_death ~ Temp_max + WBC_max,
                    family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(immune_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Temp_max + WBC_max, family = binomial(link = "logit"),
##     data = icu_patients_df1_cleaned)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.881103   3.092064   0.608    0.543
## Temp_max    -0.102805   0.082261  -1.250    0.211
## WBC_max      0.014721   0.006865   2.144    0.032 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1694.1  on 2058  degrees of freedom
## AIC: 1700.1
##
## Number of Fisher Scoring iterations: 4
```

```
# General Model
general_model <- glm(in_hospital_death ~ Age + Gender + Weight_max + ICUType,
                     family = binomial(link = 'logit'),
                     data = icu_patients_df1_h)
summary(general_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Age + Gender + Weight_max +
##     ICUType, family = binomial(link = "logit"), data = icu_patients_df1_h)
##
## Coefficients:
##                                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          -4.0772069  0.4857020  -8.394  < 2e-16 ***
## Age                                   0.0328361  0.0045562   7.207 5.72e-13 ***
## GenderMale                            0.1026777  0.1387050   0.740   0.4591
## Weight_max                            0.0002256  0.0030495   0.074   0.9410
## ICUTypeCardiac Surgery Recovery Unit -1.0938298  0.2655173  -4.120 3.80e-05 ***
## ICUTypeMedical ICU                    0.4044202  0.1931902   2.093   0.0363 *
## ICUTypeSurgical ICU                   0.1510145  0.2104912   0.717   0.4731
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1604.2  on 1914  degrees of freedom
## Residual deviance: 1493.4  on 1908  degrees of freedom
##   (因为不存在，146个观察量被删除了)
## AIC: 1507.4
##
## Number of Fisher Scoring iterations: 5
```

model summaries: 1. In the Respiratory Function Model, PaO2_max and RespRate_max are significant predictors. FiO2_max is not statistically significant, which means it might not be a strong predictor for the outcome in this model. 2. For the Coagulation & Blood Model, pH_min and Glucose_max are significant, while pH_max is not. This suggests that low blood pH (Acidosis) and glucose levels could be important predictors for in-hospital death. 3. For liver function model, except ALP_max all other biochemical markers are significant. Bilirubin_max and Albumin_min being particularly strong predictors. 4. The Cardiovascular System Model shows that Lactate_max and Cholesterol_max have a significant relationship with the outcome. HR_max, MAP_max, TroponinI_max, and TroponinT_max did not reach statistical significance, although TroponinT_max is close and could be considered for inclusion in a more comprehensive model. 5. In the Central Nervous System Model, GCS_min is a significant predictor. This aligns with medical knowledge, as a lower Glasgow Coma Score is associated with higher severity of disease. 6. For the Renal Function Model, both Creatinine_max and Urine_min are significant. This suggests that worse renal function is associated with a higher risk of in-hospital death. 7. The Immune System Model indicates that WBC_max is significant, but Temp_max is not. This could suggest that a higher white blood cell count, which might indicate infection or inflammation, is more predictive of in-hospital death than the maximum temperature. 8. Finally, the Demographics Model with Age, Gender, Weight, and ICUType shows that Age and ICUType (specifically, the Cardiac Surgery Recovery Unit and Medical ICU) are significant predictors. Gender and Weight_max do not appear to contribute significantly to this model.

If a predictor like `PaO2_max` has a very small estimate (-0.0020090) and a high p-value, it might be dropped in favor of keeping the model simpler and more interpretable, unless there is a strong clinical reason to keep it.

Based on observations, a model aiming for parsimony and generalizability could include the following variables: Age Albumin_max Bilirubin_max BUN_max Cholesterol_max Creatinine_max GCS_max GCS_min Glucose_max HR_diff Lactate_max Na_diff NISysABP_diff PaO2_max pH_diff pH_min RespRate_max Temp_diff Urine_max Urine_min WBC_max

Base on `Estimate` values we could try three different style models

Research has demonstrated that elevated levels of Albumin, AST, Bilirubin, BUN, Creatinine, GCS, Glucose, HCO3, Lactate, respiratory rate and urine output are significant indicators of survival outcome in an ICU. Hence, the maximum variables of these clinical measures may be potential predictors. On the other hand, sudden large changes in heart rate, sodium, pH, temperature and non-invasive systolic blood pressure have been found to have a strong association with ICU survival outcomes. Hence, the difference values of these variables may be more suitable as predictors.

```
# Significant Maximum Levels Model (focusing on variables identified as critical)
max_levels_model <- glm(in_hospital_death ~ Albumin_max + AST_max + Bilirubin_max + BUN_max + C
reatinine_max + GCS_max + Glucose_max + HCO3_max + Lactate_max + RespRate_max + Urine_max,
                        family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(max_levels_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Albumin_max + AST_max + Bilirubin_max +
##     BUN_max + Creatinine_max + GCS_max + Glucose_max + HCO3_max +
##     Lactate_max + RespRate_max + Urine_max, family = binomial(link = "logit"),
##     data = icu_patients_df1_cleaned)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     9.477e-01  6.667e-01   1.421  0.15522
## Albumin_max    -3.051e-01  1.134e-01  -2.690  0.00715 **
## AST_max         1.372e-04  7.786e-05   1.762  0.07807 .
## Bilirubin_max   3.919e-02  1.315e-02   2.981  0.00288 **
## BUN_max         2.444e-02  3.555e-03   6.876 6.17e-12 ***
## Creatinine_max -1.417e-01  5.355e-02  -2.645  0.00817 **
## GCS_max        -1.653e-01  2.122e-02  -7.788 6.79e-15 ***
## Glucose_max     9.886e-04  7.002e-04   1.412  0.15800
## HCO3_max       -1.864e-02  1.610e-02  -1.157  0.24712
## Lactate_max     4.770e-02  3.079e-02   1.549  0.12132
## RespRate_max    4.565e-03  8.950e-03   0.510  0.61002
## Urine_max      -1.029e-03  2.099e-04  -4.902 9.50e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1442.0  on 2049  degrees of freedom
## AIC: 1466
##
## Number of Fisher Scoring iterations: 5
```

```
# Significant Difference Levels Model (focusing on variables where changes were noted as import
ant)
diff_levels_model <- glm(in_hospital_death ~ HR_diff + Na_diff + pH_diff + Temp_diff + NISysABP
_diff,
                         family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(diff_levels_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ HR_diff + Na_diff + pH_diff +
##     Temp_diff + NISysABP_diff, family = binomial(link = "logit"),
##     data = icu_patients_df1_cleaned)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.708414   0.233460 -15.885  < 2e-16 ***
## HR_diff        0.012204   0.003714   3.286 0.001017 **
## Na_diff        0.054231   0.015055   3.602 0.000316 ***
## pH_diff        3.856638   0.939342   4.106 4.03e-05 ***
## Temp_diff      0.234241   0.081195   2.885 0.003915 **
## NISysABP_diff  0.014426   0.003351   4.306 1.67e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1608.9  on 2055  degrees of freedom
## AIC: 1620.9
##
## Number of Fisher Scoring iterations: 5
```

```
# Combined Model incorporating both max and difference values of selected indicators
combined_model <- glm(in_hospital_death ~ Albumin_max + AST_max + Bilirubin_max + BUN_max + Cre
atinine_max + GCS_max + Glucose_max + HCO3_max + Lactate_max + RespRate_max + Urine_max + HR_di
ff + Na_diff + pH_diff + Temp_diff + NISysABP_diff + Age + Gender + Weight_max,
                      family = binomial(link = 'logit'), data = icu_patients_df1_cleaned)
summary(combined_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Albumin_max + AST_max + Bilirubin_max +
##     BUN_max + Creatinine_max + GCS_max + Glucose_max + HCO3_max +
##     Lactate_max + RespRate_max + Urine_max + HR_diff + Na_diff +
##     pH_diff + Temp_diff + NISysABP_diff + Age + Gender + Weight_max,
##     family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.665e+00  8.602e-01  -1.935 0.052954 .
## Albumin_max    -3.452e-01  1.174e-01  -2.939 0.003291 **
## AST_max         1.924e-04  8.547e-05   2.251 0.024389 *
## Bilirubin_max   5.021e-02  1.327e-02   3.784 0.000155 ***
## BUN_max         1.886e-02  3.693e-03   5.108 3.25e-07 ***
## Creatinine_max -7.960e-02  5.457e-02  -1.459 0.144619
## GCS_max        -1.594e-01  2.216e-02  -7.193 6.33e-13 ***
## Glucose_max     6.685e-04  7.569e-04   0.883 0.377152
## HCO3_max       -1.789e-02  1.651e-02  -1.083 0.278701
## Lactate_max     1.397e-03  3.433e-02   0.041 0.967545
## RespRate_max    3.515e-03  9.341e-03   0.376 0.706650
## Urine_max      -8.896e-04  2.083e-04  -4.270 1.96e-05 ***
## HR_diff         1.028e-02  4.086e-03   2.517 0.011828 *
## Na_diff         2.214e-02  1.850e-02   1.197 0.231453
## pH_diff         2.504e+00  1.135e+00   2.207 0.027294 *
## Temp_diff       1.425e-01  9.053e-02   1.574 0.115476
## NISysABP_diff   9.901e-03  3.812e-03   2.597 0.009400 **
## Age             2.744e-02  4.935e-03   5.561 2.69e-08 ***
## GenderMale      5.097e-03  1.464e-01   0.035 0.972221
## Weight_max     -3.701e-03  3.424e-03  -1.081 0.279662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1372.7  on 2041  degrees of freedom
## AIC: 1412.7
##
## Number of Fisher Scoring iterations: 6
```

```
#vars_needed <- c("Age", "SAPS1", "SOFA", "Albumin_max", "ALT_max", "AST_max", "Bilirubin_max",
"BUN_max",
#                 "Cholesterol_max", "Creatinine_max", "GCS_max", "GCS_min", "Glucose_max", "HR
_diff",
#                 "Lactate_max", "Na_diff", "NISysABP_diff", "PaO2_max", "pH_diff", "pH_min",
"RespRate_max",
#                 "Temp_diff", "Urine_max", "Urine_min", "WBC_max", "ICUType", "in_hospital_dea
th") # Add all other variables involved in both models

#consistent_data <- icu_patients_df1[complete.cases(icu_patients_df1[, vars_needed]), ]

# Creating intact model with all significant predictors
intact_model <- glm(in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max + ALT_max + AST_max +
Bilirubin_max + BUN_max + Cholesterol_max +
                      Creatinine_max + GCS_max + GCS_min + Glucose_max + HR_diff + Lactate_ma
x + Na_diff +
                      NISysABP_diff + PaO2_max + pH_diff + pH_min + RespRate_max + Temp_diff
+ Urine_max +
                      Urine_min + WBC_max + ICUType,
                    family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
summary(intact_model)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max +
##     ALT_max + AST_max + Bilirubin_max + BUN_max + Cholesterol_max +
##     Creatinine_max + GCS_max + GCS_min + Glucose_max + HR_diff +
##     Lactate_max + Na_diff + NISysABP_diff + PaO2_max + pH_diff +
##     pH_min + RespRate_max + Temp_diff + Urine_max + Urine_min +
##     WBC_max + ICUType, family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         0.0164869  7.9167297   0.002 0.998338
## Age                                 0.0248098  0.0054517   4.551 5.34e-06 ***
## SAPS1                               0.0479217  0.0256290   1.870 0.061508 .
## SOFA                                0.0542964  0.0288910   1.879 0.060197 .
## Albumin_max                        -0.2207325  0.1247121  -1.770 0.076738 .
## ALT_max                            -0.0010265  0.0004560  -2.251 0.024394 *
## AST_max                             0.0007422  0.0002760   2.689 0.007159 **
## Bilirubin_max                       0.0441906  0.0141351   3.126 0.001770 **
## BUN_max                             0.0151427  0.0039018   3.881 0.000104 ***
## Cholesterol_max                    -0.0028892  0.0021848  -1.322 0.186032
## Creatinine_max                     -0.1098293  0.0549164  -2.000 0.045507 *
## GCS_max                            -0.1757922  0.0303920  -5.784 7.29e-09 ***
## GCS_min                             0.0573900  0.0289728   1.981 0.047611 *
## Glucose_max                         0.0002731  0.0008164   0.335 0.737943
## HR_diff                             0.0076404  0.0043728   1.747 0.080590 .
## Lactate_max                         0.0072310  0.0365558   0.198 0.843196
## Na_diff                             0.0173701  0.0193868   0.896 0.370266
## NISysABP_diff                       0.0077138  0.0039997   1.929 0.053780 .
## PaO2_max                           -0.0019202  0.0008011  -2.397 0.016529 *
## pH_diff                             2.5020583  1.4083357   1.777 0.075633 .
## pH_min                             -0.3332980  1.0332017  -0.323 0.747008
## RespRate_max                        0.0037918  0.0098111   0.386 0.699138
## Temp_diff                           0.0820163  0.0983564   0.834 0.404355
## Urine_max                          -0.0007298  0.0002151  -3.393 0.000692 ***
## Urine_min                          -0.0019255  0.0018548  -1.038 0.299206
## WBC_max                            -0.0091427  0.0086483  -1.057 0.290434
## ICUTypeCardiac Surgery Recovery Unit -0.6354553  0.3242303  -1.960 0.050009 .
## ICUTypeMedical ICU                 -0.1043047  0.2136477  -0.488 0.625402
## ICUTypeSurgical ICU                 0.0099559  0.2359503   0.042 0.966343
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1338.2  on 2032  degrees of freedom
## AIC: 1396.2
##
## Number of Fisher Scoring iterations: 6
```

```
# Using stepwise regression to find an optimal set of predictors for intact model
stepwise_result <- stepAIC(intact_model, direction = "both", trace = FALSE) #stepwise_result2 <
- stats::step(intact_model,direction="both", trace=0)
summary(stepwise_result)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max +
##     ALT_max + AST_max + Bilirubin_max + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + HR_diff + NISysABP_diff + PaO2_max +
##     pH_diff + Urine_max + ICUType, family = binomial(link = "logit"),
##     data = icu_patients_df1_cleaned)
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -2.9586700  0.7516403  -3.936 8.28e-05 ***
## Age                               0.0261907  0.0052084   5.029 4.94e-07 ***
## SAPS1                             0.0580310  0.0230064   2.522 0.011656 *
## SOFA                              0.0583371  0.0280534   2.079 0.037572 *
## Albumin_max                      -0.2497119  0.1199924  -2.081 0.037428 *
## ALT_max                          -0.0010521  0.0004431  -2.375 0.017571 *
## AST_max                           0.0007670  0.0002632   2.914 0.003570 **
## Bilirubin_max                     0.0463521  0.0135900   3.411 0.000648 ***
## BUN_max                           0.0154032  0.0037981   4.055 5.00e-05 ***
## Creatinine_max                   -0.1037505  0.0541855  -1.915 0.055527 .
## GCS_max                          -0.1708445  0.0275421  -6.203 5.54e-10 ***
## GCS_min                           0.0546851  0.0282780   1.934 0.053133 .
## HR_diff                           0.0078886  0.0043060   1.832 0.066948 .
## NISysABP_diff                     0.0075894  0.0039740   1.910 0.056160 .
## PaO2_max                         -0.0018209  0.0007904  -2.304 0.021244 *
## pH_diff                           2.5952276  1.1409727   2.275 0.022931 *
## Urine_max                        -0.0007388  0.0002126  -3.475 0.000511 ***
## ICUTypeCardiac Surgery Recovery Unit -0.7094156  0.3114644  -2.278 0.022746 *
## ICUTypeMedical ICU               -0.0874038  0.2103697  -0.415 0.677793
## ICUTypeSurgical ICU               0.0258841  0.2331446   0.111 0.911599
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1344.0  on 2041  degrees of freedom
## AIC: 1384
##
## Number of Fisher Scoring iterations: 6
```

```
# Variance Inflation Factor (VIF) check
vif_values <- vif(intact_model, type = 'terms')
print(vif_values)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## Age                1.424504  1        1.193526
## SAPS1              2.897716  1        1.702268
## SOFA               2.777311  1        1.666527
## Albumin_max        1.154386  1        1.074424
## ALT_max           10.538677  1        3.246333
## AST_max           10.829452  1        3.290813
## Bilirubin_max      1.181173  1        1.086818
## BUN_max            2.211248  1        1.487027
## Cholesterol_max    1.640938  1        1.280991
## Creatinine_max     2.054436  1        1.433330
## GCS_max            2.233864  1        1.494612
## GCS_min            3.591989  1        1.895254
## Glucose_max        1.283392  1        1.132869
## HR_diff            1.184397  1        1.088300
## Lactate_max        1.567846  1        1.252137
## Na_diff            1.190584  1        1.091139
## NISysABP_diff      1.147000  1        1.070981
## PaO2_max           1.554490  1        1.246792
## pH_diff            1.830825  1        1.353080
## pH_min             2.068455  1        1.438212
## RespRate_max       1.322456  1        1.149981
## Temp_diff          1.297283  1        1.138983
## Urine_max          1.169846  1        1.081594
## Urine_min          1.126448  1        1.061343
## WBC_max            1.187742  1        1.089836
## ICUType            1.881888  3        1.111132
```

Steps to Refine the Model:

stepAIC method adds or removes predictors based on the AIC value in a stepwise manner to identify a model that balances model complexity with information loss. StepAIC() helped refine the model by retaining significant predictors and removing those less contributive variables. - The summary of the stepwise regression model presents similar significant variables and AIC as found in the previous model. - Predictors shows a relatively high p-value (>0.05),implies that it may not contribute meaningful explanatory power to the model in predicting in-hospital death.

VIF values 1. The VIF values common thresholds are 5 or 10, below that suggests that there isn't a concerning level of multicollinearity among the main effects in the model. 1. High GVIF values for interaction terms are not uncommon, as these terms are products of their component variables and can inherit their collinearity. 2. High VIF/GVIF values do not imply that a model is invalid; rather, they suggest that the precision of the coefficient estimates for the related variables may be reduced.

When removing the predictor, consider the following: 1. **Drop Non-Significant (especially low abs of Estimate value) Predictors:** Glucose_max, Urine_min, Cholesterol_max, RespRate_max 2. **Address Multicollinearity:** ALT_max and AST_max both have high VIFs and are related liver enzymes, need redundancy. 3. **Clinical Relevance:** `GCS_min` (representing the lowest Glasgow Coma Score), despite its higher p-value, are clinically relevant and known to be associated with patient outcomes, need keep.

```
# Creating refined model
intact_model_rf <- glm(in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max  + ALT_max + AST_ma
x + Bilirubin_max + BUN_max  + Creatinine_max + GCS_max + GCS_min
                         + HR_diff + Lactate_max + Na_diff + NISysABP_diff + PaO2_max + pH_dif
f  + Temp_diff + Urine_max + WBC_max + ICUType,
                     family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
summary(intact_model_rf)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max +
##     ALT_max + AST_max + Bilirubin_max + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + HR_diff + Lactate_max + Na_diff + NISysABP_diff +
##     PaO2_max + pH_diff + Temp_diff + Urine_max + WBC_max + ICUType,
##     family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -2.9816620  0.7637398  -3.904 9.46e-05 ***
## Age                                 0.0268039  0.0052702   5.086 3.66e-07 ***
## SAPS1                               0.0503409  0.0251194   2.004 0.045063 *
## SOFA                                0.0566962  0.0283075   2.003 0.045191 *
## Albumin_max                        -0.2521060  0.1202452  -2.097 0.036029 *
## ALT_max                            -0.0010830  0.0004507  -2.403 0.016261 *
## AST_max                             0.0007894  0.0002722   2.900 0.003727 **
## Bilirubin_max                       0.0461873  0.0137088   3.369 0.000754 ***
## BUN_max                             0.0155127  0.0038377   4.042 5.30e-05 ***
## Creatinine_max                     -0.1032609  0.0543665  -1.899 0.057519 .
## GCS_max                            -0.1683329  0.0276420  -6.090 1.13e-09 ***
## GCS_min                             0.0507482  0.0285071   1.780 0.075044 .
## HR_diff                             0.0078809  0.0043546   1.810 0.070329 .
## Lactate_max                        -0.0023251  0.0350660  -0.066 0.947134
## Na_diff                             0.0149942  0.0188911   0.794 0.427358
## NISysABP_diff                       0.0077357  0.0040043   1.932 0.053377 .
## PaO2_max                           -0.0018319  0.0007946  -2.306 0.021137 *
## pH_diff                             2.6146253  1.1766264   2.222 0.026274 *
## Temp_diff                           0.0943813  0.0976836   0.966 0.333947
## Urine_max                          -0.0007538  0.0002138  -3.526 0.000422 ***
## WBC_max                            -0.0056884  0.0082299  -0.691 0.489447
## ICUTypeCardiac Surgery Recovery Unit -0.6713575  0.3131944  -2.144 0.032067 *
## ICUTypeMedical ICU                 -0.0832946  0.2111772  -0.394 0.693263
## ICUTypeSurgical ICU                 0.0295446  0.2341156   0.126 0.899576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1341.7  on 2037  degrees of freedom
## AIC: 1389.7
##
## Number of Fisher Scoring iterations: 6
```

```
# Using stepwise regression to find an optimal set of predictors
stepwise_result_rf <- stepAIC(intact_model_rf, direction = "both", trace = FALSE)
summary(stepwise_result_rf)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max +
##     ALT_max + AST_max + Bilirubin_max + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + HR_diff + NISysABP_diff + PaO2_max +
##     pH_diff + Urine_max + ICUType, family = binomial(link = "logit"),
##     data = icu_patients_df1_cleaned)
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -2.9586700  0.7516403  -3.936 8.28e-05 ***
## Age                                 0.0261907  0.0052084   5.029 4.94e-07 ***
## SAPS1                               0.0580310  0.0230064   2.522 0.011656 *
## SOFA                                0.0583371  0.0280534   2.079 0.037572 *
## Albumin_max                        -0.2497119  0.1199924  -2.081 0.037428 *
## ALT_max                            -0.0010521  0.0004431  -2.375 0.017571 *
## AST_max                             0.0007670  0.0002632   2.914 0.003570 **
## Bilirubin_max                       0.0463521  0.0135900   3.411 0.000648 ***
## BUN_max                             0.0154032  0.0037981   4.055 5.00e-05 ***
## Creatinine_max                     -0.1037505  0.0541855  -1.915 0.055527 .
## GCS_max                            -0.1708445  0.0275421  -6.203 5.54e-10 ***
## GCS_min                             0.0546851  0.0282780   1.934 0.053133 .
## HR_diff                             0.0078886  0.0043060   1.832 0.066948 .
## NISysABP_diff                       0.0075894  0.0039740   1.910 0.056160 .
## PaO2_max                           -0.0018209  0.0007904  -2.304 0.021244 *
## pH_diff                             2.5952276  1.1409727   2.275 0.022931 *
## Urine_max                          -0.0007388  0.0002126  -3.475 0.000511 ***
## ICUTypeCardiac Surgery Recovery Unit -0.7094156  0.3114644  -2.278 0.022746 *
## ICUTypeMedical ICU                 -0.0874038  0.2103697  -0.415 0.677793
## ICUTypeSurgical ICU                 0.0258841  0.2331446   0.111 0.911599
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1699.7  on 2060  degrees of freedom
## Residual deviance: 1344.0  on 2041  degrees of freedom
## AIC: 1384
##
## Number of Fisher Scoring iterations: 6
```

```
# Variance Inflation Factor (VIF) check
vif_values_rf <- vif(intact_model_rf, type = 'terms')
print(vif_values_rf)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## Age             1.327586  1        1.152209
## SAPS1           2.813377  1        1.677312
## SOFA            2.687065  1        1.639227
## Albumin_max     1.078107  1        1.038319
## ALT_max        10.284792  1        3.206991
## AST_max        10.566579  1        3.250627
## Bilirubin_max   1.127433  1        1.061807
## BUN_max         2.162246  1        1.470458
## Creatinine_max  2.037418  1        1.427381
## GCS_max         1.849188  1        1.359848
## GCS_min         3.489592  1        1.868045
## HR_diff         1.174427  1        1.083710
## Lactate_max     1.464537  1        1.210181
## Na_diff         1.133021  1        1.064435
## NISysABP_diff   1.145875  1        1.070456
## PaO2_max        1.528006  1        1.236125
## pH_diff         1.282878  1        1.132642
## Temp_diff       1.284592  1        1.133398
## Urine_max       1.140818  1        1.068091
## WBC_max         1.100717  1        1.049150
## ICUType         1.694663  3        1.091894
```

```
# Comparing models with ANOVA
anova_results <- anova(intact_model, intact_model_rf, test = "Chisq")
print(anova_results)
```

```
## Analysis of Deviance Table
##
## Model 1: in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max + ALT_max +
##     AST_max + Bilirubin_max + BUN_max + Cholesterol_max + Creatinine_max +
##     GCS_max + GCS_min + Glucose_max + HR_diff + Lactate_max +
##     Na_diff + NISysABP_diff + PaO2_max + pH_diff + pH_min + RespRate_max +
##     Temp_diff + Urine_max + Urine_min + WBC_max + ICUType
## Model 2: in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max + ALT_max +
##     AST_max + Bilirubin_max + BUN_max + Creatinine_max + GCS_max +
##     GCS_min + HR_diff + Lactate_max + Na_diff + NISysABP_diff +
##     PaO2_max + pH_diff + Temp_diff + Urine_max + WBC_max + ICUType
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      2032     1338.2
## 2      2037     1341.7 -5    -3.43    0.634
```

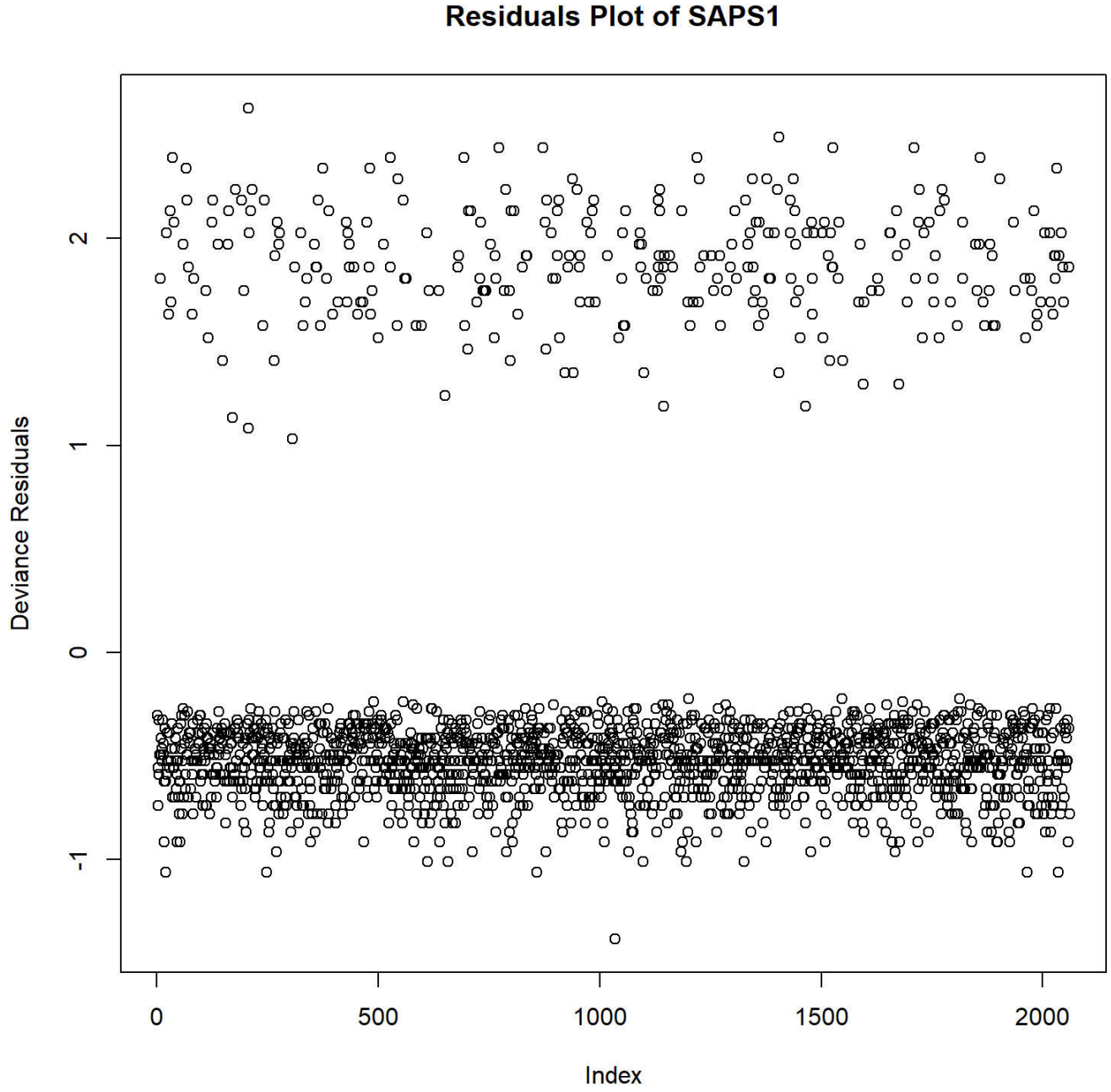
ANOVA result: - **Residual Degrees of Freedom:** The first model (`intact_model`) has 1522 degrees of freedom, while the second model (`intact_model_rf`) has 1526. This suggests that the second model has four fewer predictors than the first. - **Residual Deviance:** The residual deviance for the first model is 1076.9 compared to 1081.5 for the second model. Lower residual deviance generally indicates a better fit to the data. - **Difference in Deviance:** The difference in deviance between the two models is -4.5503, which means that removing the four predictors (going from the first model to the second model) has increased the deviance slightly, indicating a slight loss in fit. - **Pr(>Chi):** The p-value of 0.3366 suggests that the increase in deviance (loss in fit) by removing these four predictors is statistically significant. It will be reasonable to prefer the simpler model (`intact_model_rf`)

```
# Creating two-way interaction models from previous models
#two_way_model <- update(intact_model_rf, . ~ .^2)
# Dropping each two-way interaction term to test significance
#drop_interaction_result <- drop1(two_way_model, test = "Chisq")
#drop_interaction_result
```

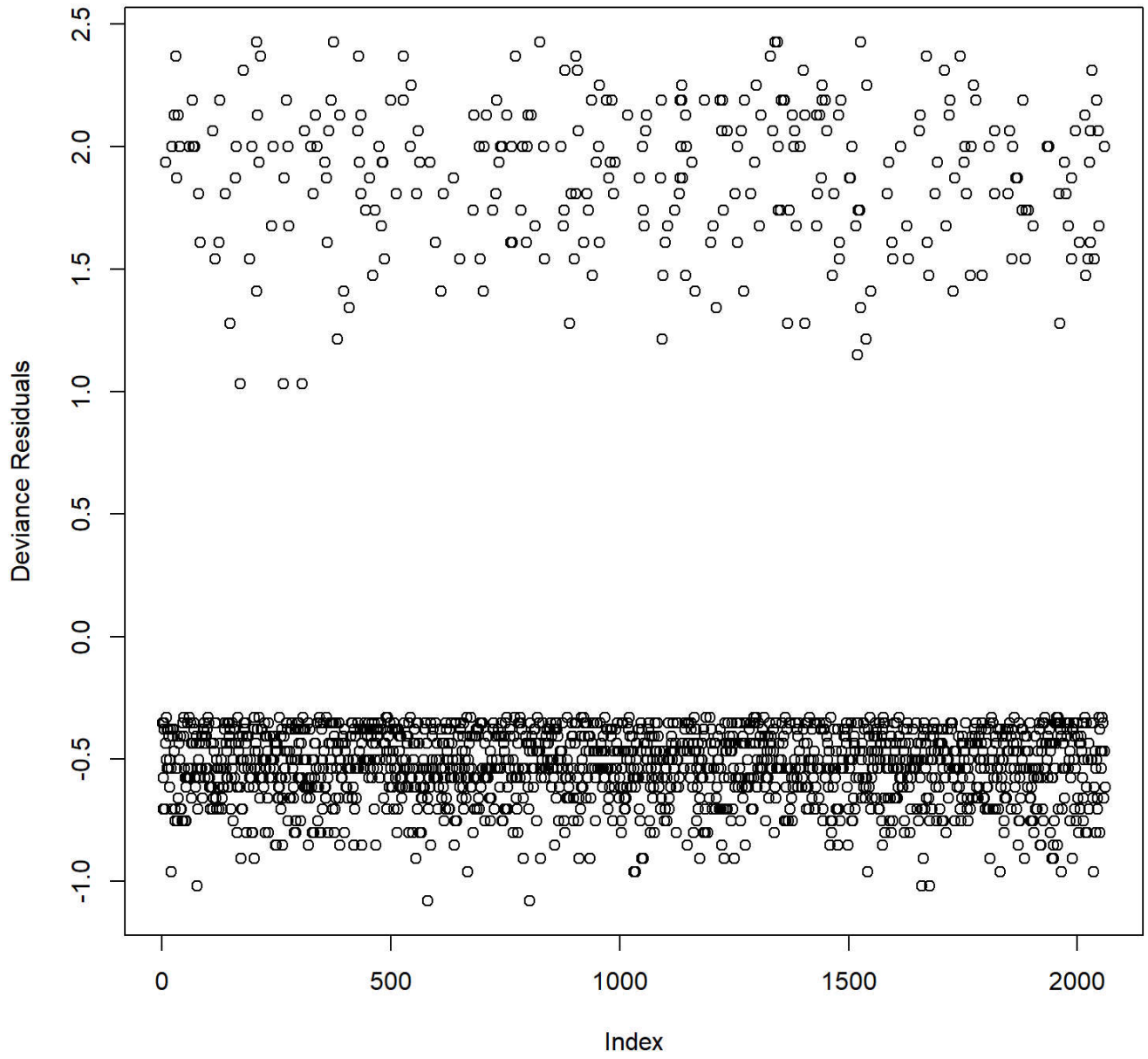
Two-way interactions 1. Most interaction terms do not significantly improve the model (indicated by high p-values), which implies that the main effects of the predictors are sufficient for the prediction. 2. A few interaction terms do show significance (indicated by low p-values), which might suggest that the relationship between predictors and the in_hospital_death is more complex than additive. Notably: 3. considering the inclusion of interaction terms, it is essential to evaluate their clinical relevance and the potential for overfitting. Models with too many interactions, especially in smaller datasets, can fit the noise in the data rather than the underlying relationships, leading to models that may not generalize well to new data.

Inclusion of Main Effects: Typically, when including an interaction term in a regression model, should also include the main effects of the interacting variables. This approach avoids the model misattributing the effects that should be captured by the main effects alone to the interaction term.

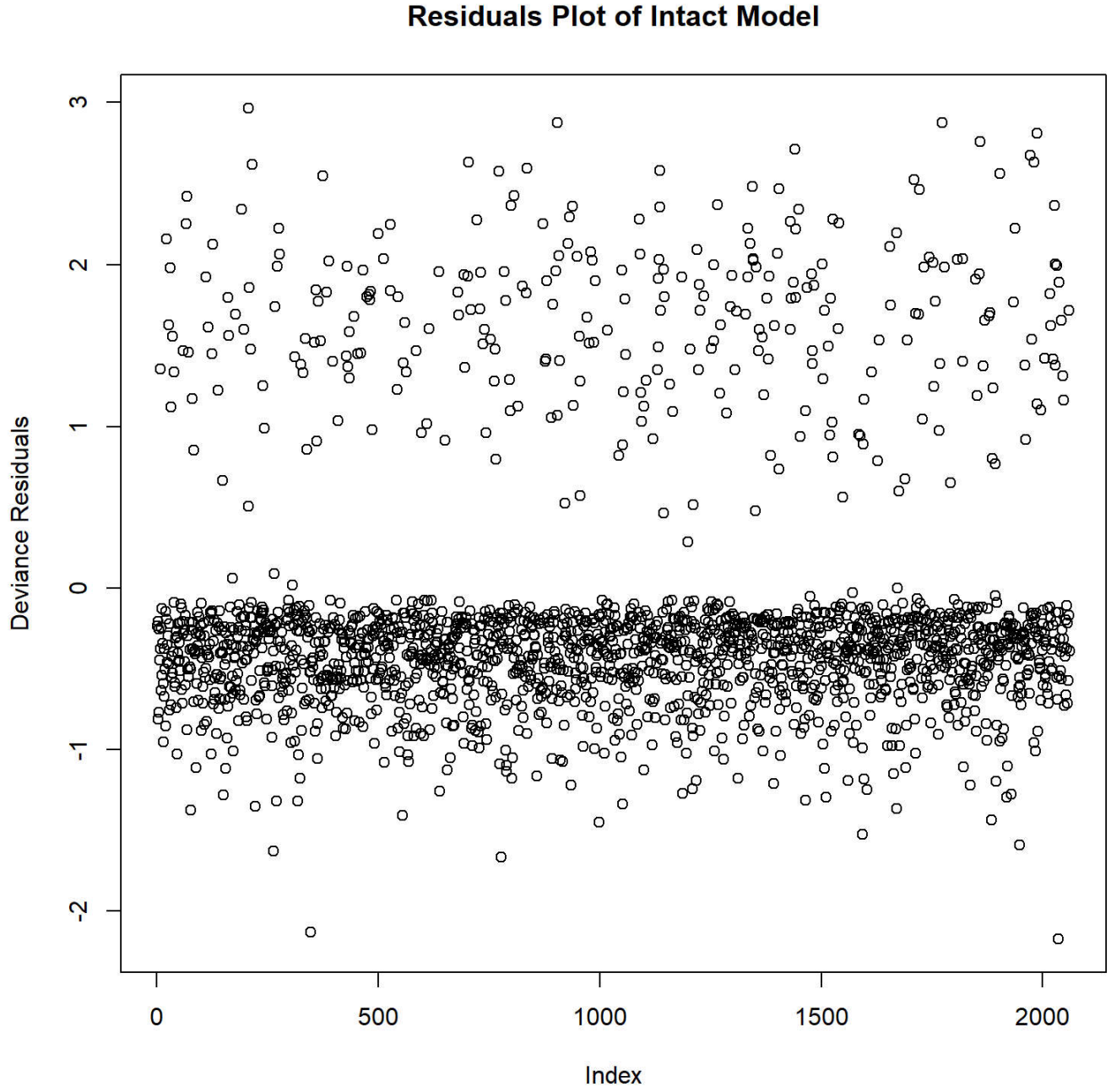
```
# 1. Residuals Plot
plot(residuals(model_saps1, type = "deviance"), main = "Residuals Plot of SAPS1", ylab = "Devia
nce Residuals")
```



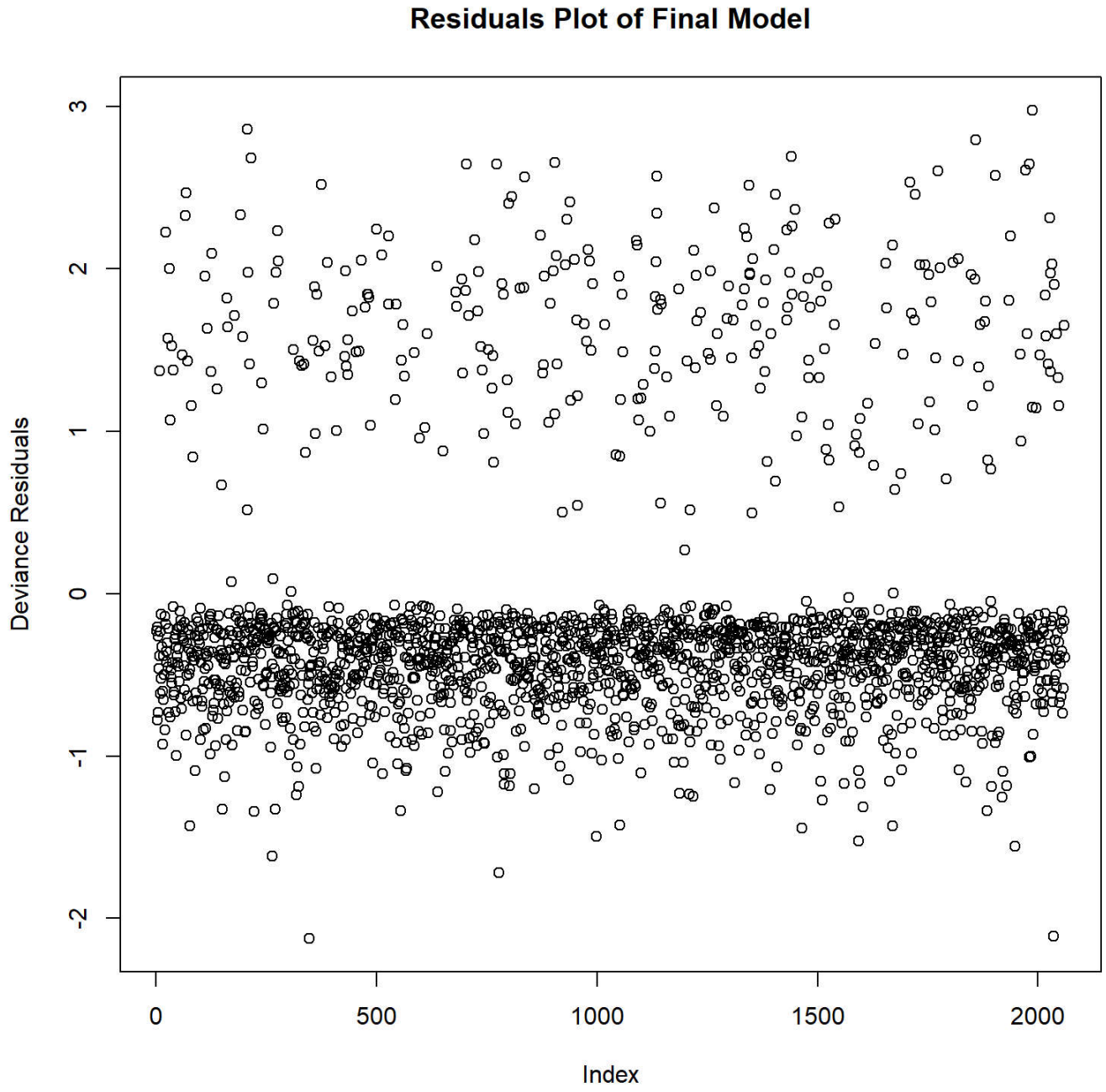

```
plot(residuals(model_sofa, type = "deviance"), main = "Residuals Plot of SOFA", ylab = "Deviance Residuals")
```




```
plot(residuals(intact_model, type = "deviance"), main = "Residuals Plot of Intact Model", ylab
= "Deviance Residuals")
```



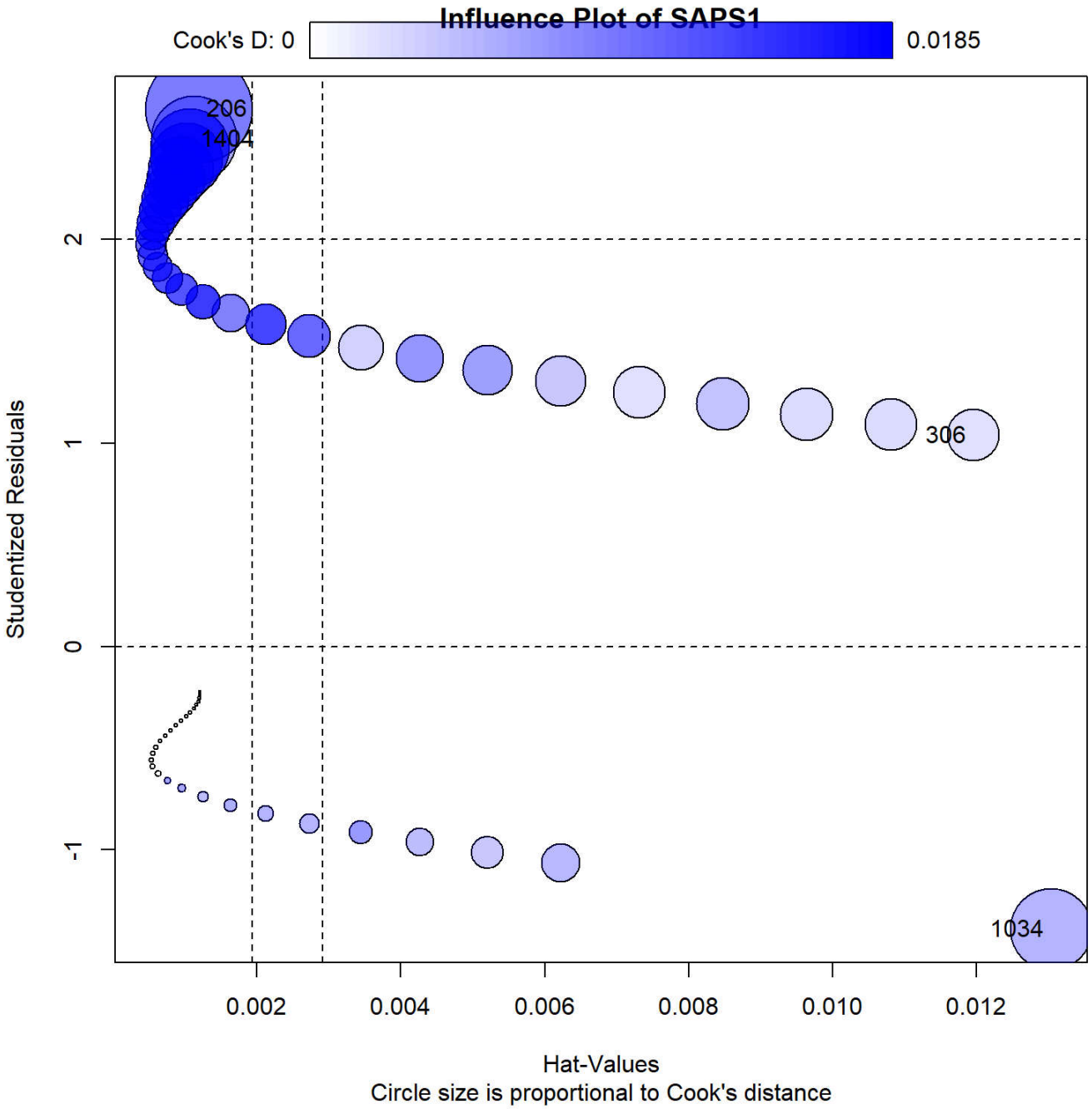

```
plot(residuals(intact_model_rf, type = "deviance"), main = "Residuals Plot of Final Model", yla
b = "Deviance Residuals")
```




```
intact_model
```

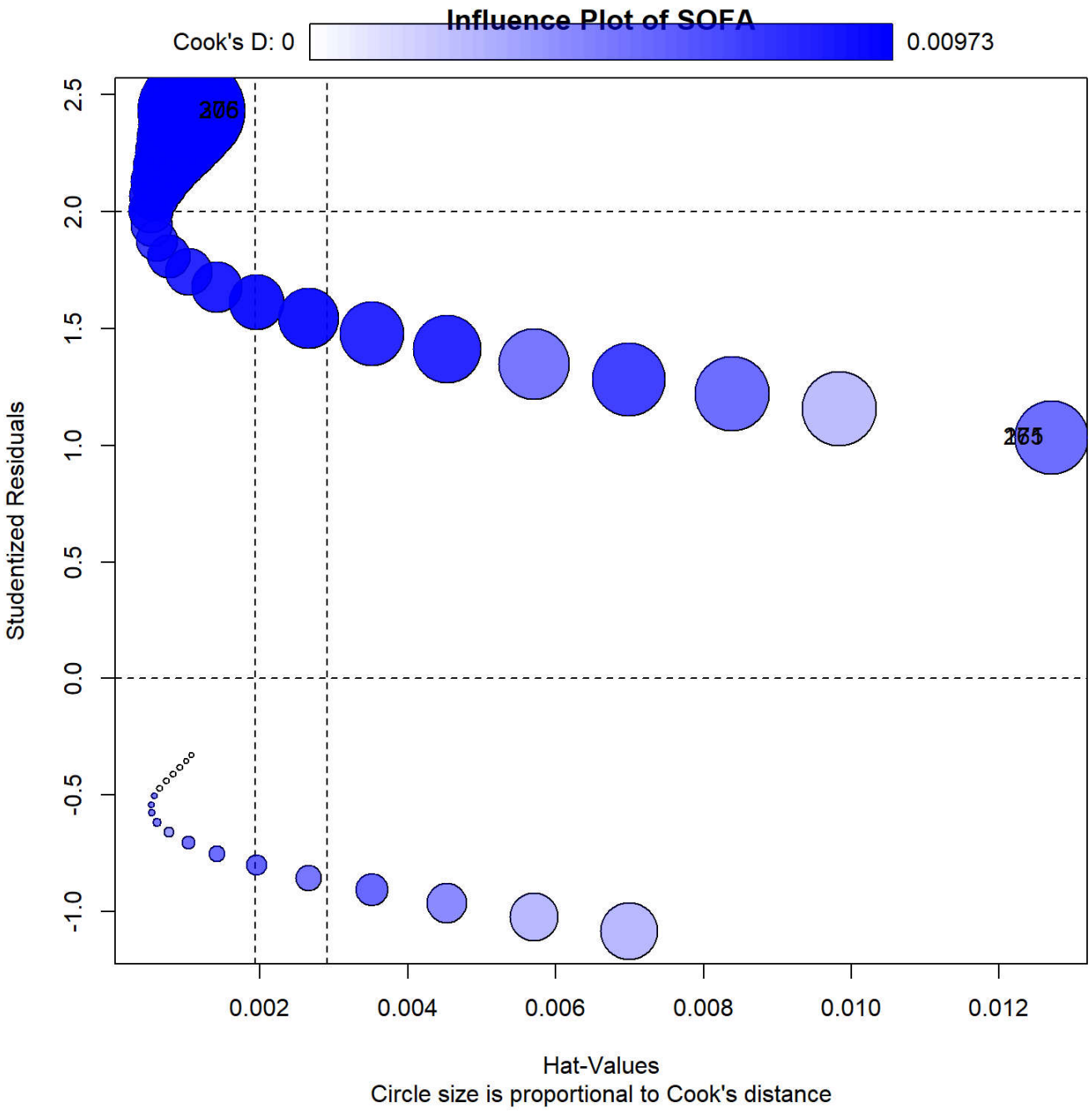
```
##
## Call:  glm(formula = in_hospital_death ~ Age + SAPS1 + SOFA + Albumin_max +
##     ALT_max + AST_max + Bilirubin_max + BUN_max + Cholesterol_max +
##     Creatinine_max + GCS_max + GCS_min + Glucose_max + HR_diff +
##     Lactate_max + Na_diff + NISysABP_diff + PaO2_max + pH_diff +
##     pH_min + RespRate_max + Temp_diff + Urine_max + Urine_min +
##     WBC_max + ICUType, family = binomial(link = "logit"), data = icu_patients_df1_cleaned)
##
## Coefficients:
##                           (Intercept)                                   Age
##                             0.0164869                             0.0248098
##                                 SAPS1                                  SOFA
##                             0.0479217                             0.0542964
##                           Albumin_max                               ALT_max
##                            -0.2207325                            -0.0010265
##                               AST_max                         Bilirubin_max
##                             0.0007422                             0.0441906
##                               BUN_max                       Cholesterol_max
##                             0.0151427                            -0.0028892
##                         Creatinine_max                               GCS_max
##                            -0.1098293                            -0.1757922
##                               GCS_min                          Glucose_max
##                             0.0573900                             0.0002731
##                               HR_diff                          Lactate_max
##                             0.0076404                             0.0072310
##                               Na_diff                       NISysABP_diff
##                             0.0173701                             0.0077138
##                              PaO2_max                               pH_diff
##                            -0.0019202                             2.5020583
##                                 pH_min                        RespRate_max
##                            -0.3332980                             0.0037918
##                              Temp_diff                          Urine_max
##                             0.0820163                            -0.0007298
##                              Urine_min                             WBC_max
##                            -0.0019255                            -0.0091427
## ICUTypeCardiac Surgery Recovery Unit                   ICUTypeMedical ICU
##                            -0.6354553                            -0.1043047
##                   ICUTypeSurgical ICU
##                             0.0099559
##
## Degrees of Freedom: 2060 Total (i.e. Null);  2032 Residual
## Null Deviance:       1700
## Residual Deviance: 1338  AIC: 1396
```

```
# 2. Influence Plot (Cook's Distance)
influencePlot(model_saps1, main="Influence Plot of SAPS1", sub="Circle size is proportional to Cook's distance")
```



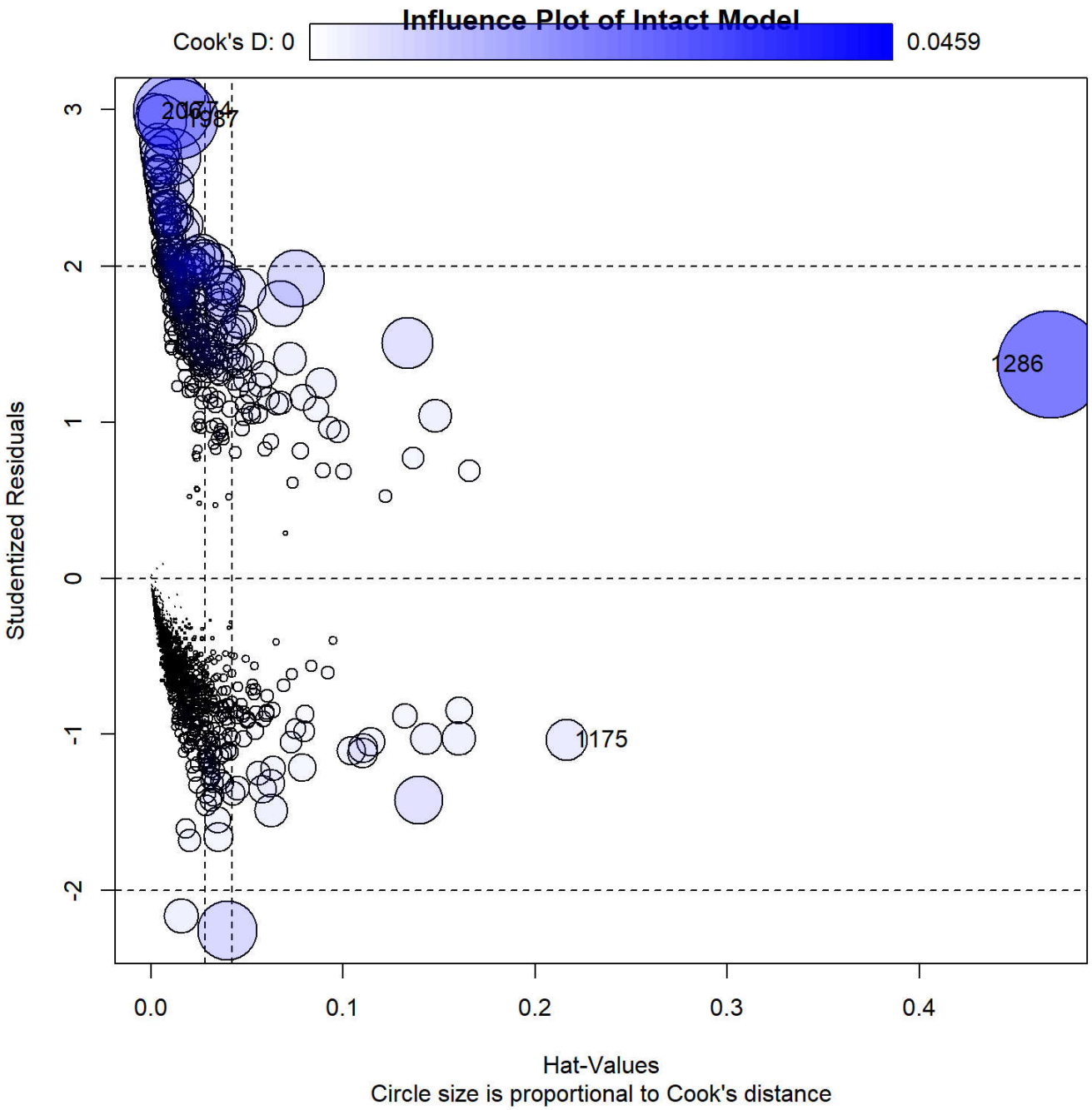

```
##         StudRes         Hat      CookD
## 206    2.636007 0.001201769 0.01848184
## 306    1.038040 0.011953567 0.00432586
## 1034 -1.391381 0.013035538 0.01073811
## 1404   2.491937 0.001122171 0.01183420
```

```
influencePlot(model_sofa, main="Influence Plot of SOFA", sub="Circle size is proportional to Cook's distance")
```



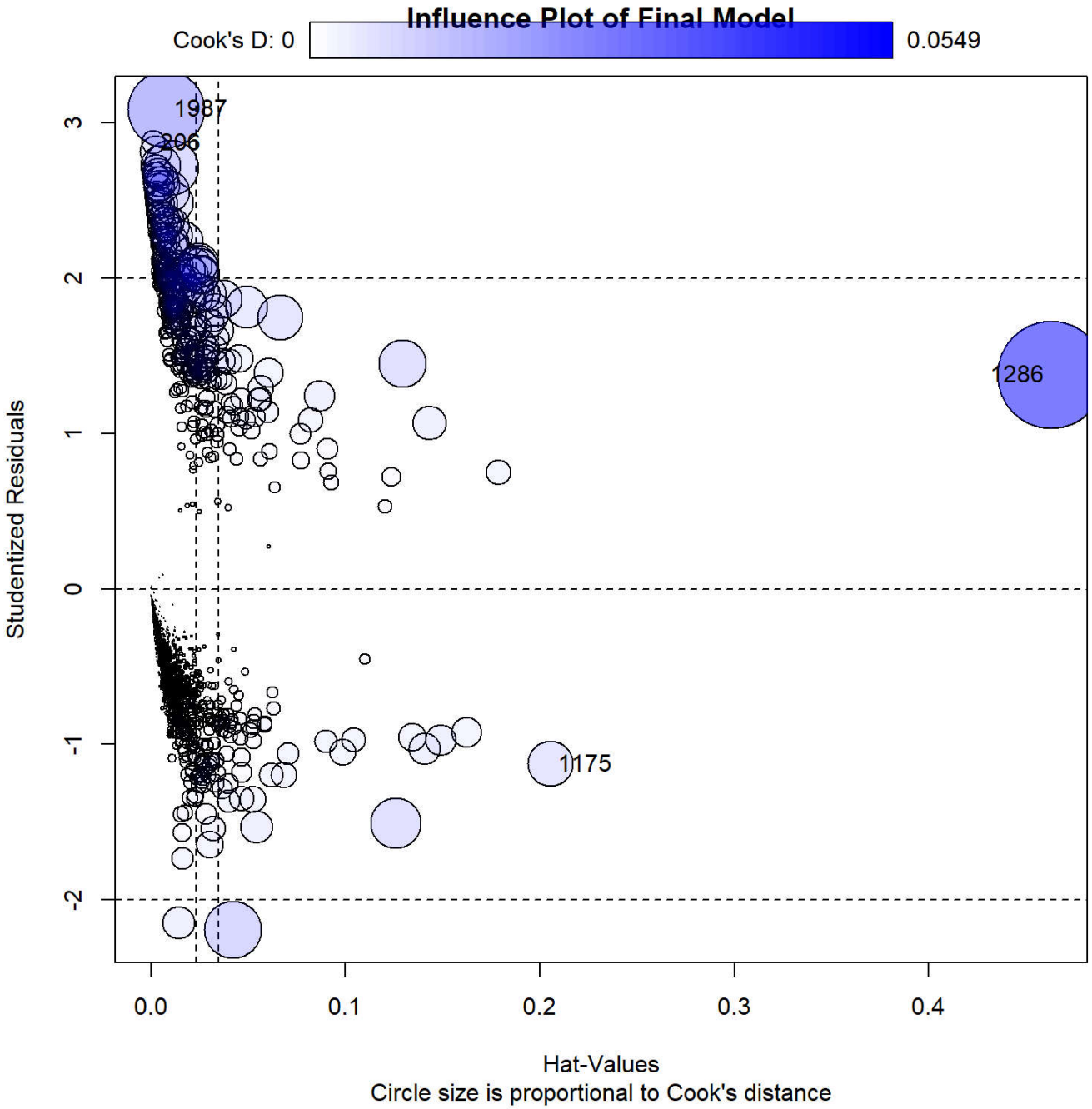

```
##      StudRes         Hat       CookD
## 171 1.035620 0.012702075 0.004573168
## 206 2.429778 0.001081729 0.009733499
## 265 1.035620 0.012702075 0.004573168
## 375 2.429778 0.001081729 0.009733499
```

```
influencePlot(intact_model, main="Influence Plot of Intact Model", sub="Circle size is proportional to Cook's distance")
```



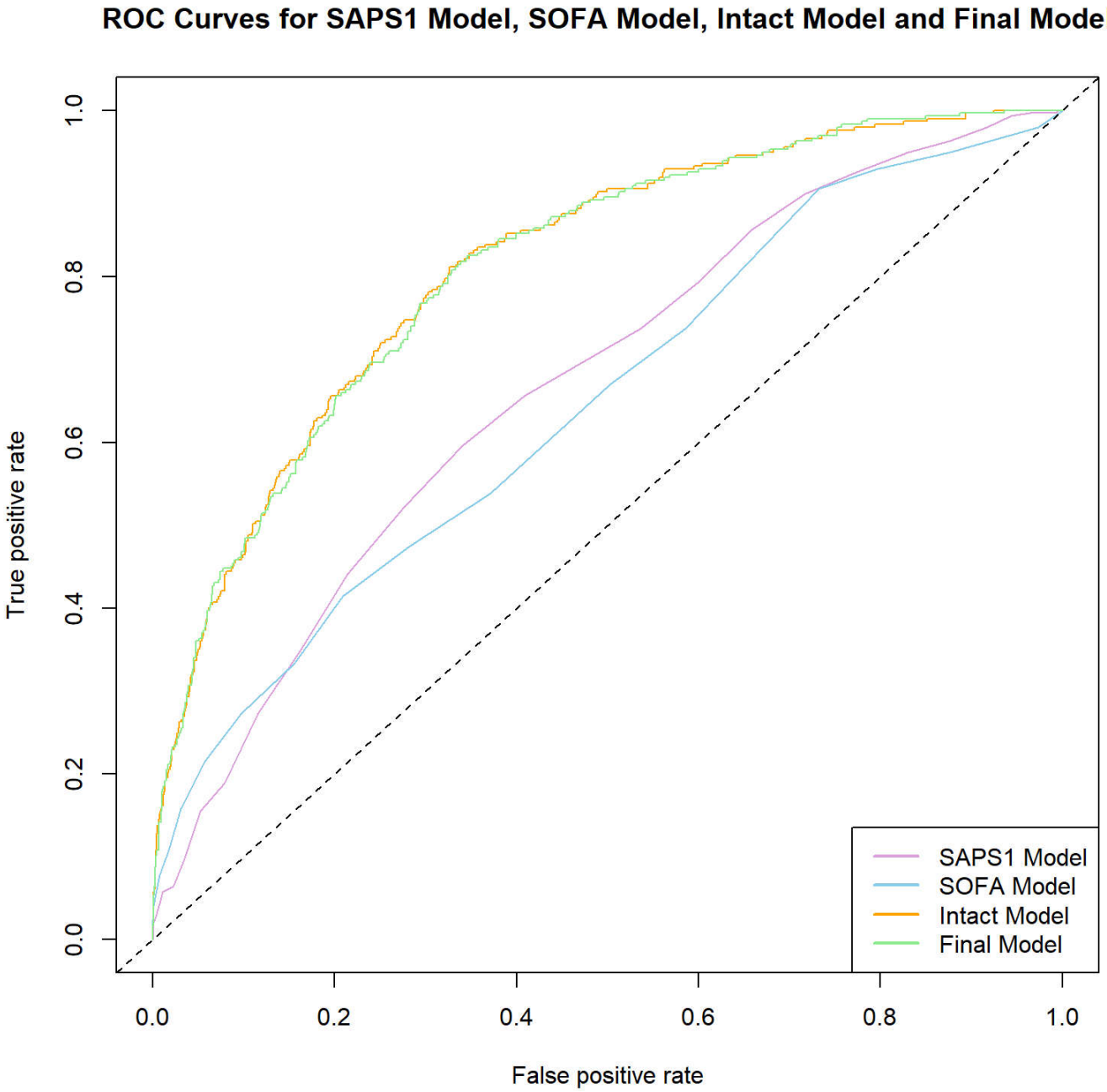

```
##         StudRes         Hat       CookD
## 206    2.990649 0.001826921 0.005078943
## 1175  -1.036685 0.216523989 0.007051454
## 1286   1.372395 0.468695161 0.045865369
## 1774   2.993856 0.010874507 0.023720569
## 1987   2.938952 0.014254571 0.025774350
```

```
influencePlot(intact_model_rf, main="Influence Plot of Final Model", sub="Circle size is proportional to Cook's distance")
```




```
##          StudRes         Hat       CookD
## 206    2.870899 0.001136160 0.002780565
## 1175  -1.123986 0.205437400 0.009691034
## 1286   1.380182 0.462975495 0.054859623
## 1987   3.085632 0.007955292 0.027890648
```

```
# 3. ROC Curve
# Predicted probabilities for the models
pred_saps1 <- predict(model_saps1, type = "response")
pred_sofa <- predict(model_sofa, type = "response")
pred_imodel <- predict(intact_model, type = "response")
pred_fmodel <- predict(intact_model_rf, type = "response")

# Create prediction objects for ROC analysis
pred_obj_saps1 <- prediction(pred_saps1, icu_patients_df1_cleaned$in_hospital_death)
pred_obj_sofa <- prediction(pred_sofa, icu_patients_df1_cleaned$in_hospital_death)
pred_obj_imodel <- prediction(pred_imodel, icu_patients_df1_cleaned$in_hospital_death)
pred_obj_fmodel <- prediction(pred_fmodel, icu_patients_df1_cleaned$in_hospital_death)

# Create performance objects for ROC analysis
perf_saps1 <- performance(pred_obj_saps1, "tpr", "fpr")
perf_sofa <- performance(pred_obj_sofa, "tpr", "fpr")
perf_imodel <- performance(pred_obj_imodel, "tpr", "fpr")
perf_fmodel <- performance(pred_obj_fmodel, "tpr", "fpr")

# Plot ROC curves
plot(perf_saps1, col = "plum", main = "ROC Curves for SAPS1 Model, SOFA Model, Intact Model and
Final Model")
plot(perf_sofa, col = "skyblue", add = TRUE)
plot(perf_imodel, col = "orange", add = TRUE)
plot(perf_fmodel, col = "lightgreen", add = TRUE)
abline(a = 0, b = 1, lty = 2)

# Add a legend
legend("bottomright", legend = c("SAPS1 Model", "SOFA Model", "Intact Model", "Final Model"),
       col = c("plum", "skyblue", "orange", "lightgreen"),
       lwd = 2)
```




```
# Calculate AUC for SAPS1 model
auc_saps1 <- performance(pred_obj_saps1, measure = "auc")
auc_saps1_value <- auc_saps1@y.values[[1]]
cat("AUC for SAPS1 Model:", auc_saps1_value, "\n")
```

```
## AUC for SAPS1 Model: 0.6689533
```

```
# Calculate AUC for SOFA model
auc_sofa <- performance(pred_obj_sofa, measure = "auc")
auc_sofa_value <- auc_sofa@y.values[[1]]
cat("AUC for SOFA Model:", auc_sofa_value, "\n")
```

```
## AUC for SOFA Model: 0.6453499
```

```
# Calculate AUC for Intact models
auc_imodel <- performance(pred_obj_imodel, measure = "auc")
auc_imodel_value <- auc_imodel@y.values[[1]]
cat("AUC for Intact Model:", auc_imodel_value, "\n")
```

```
## AUC for Intact Model: 0.8118887
```

```
# Calculate AUC for Final models
auc_fmodel <- performance(pred_obj_fmodel, measure = "auc")
auc_fmodel_value <- auc_fmodel@y.values[[1]]
cat("AUC for Final Model:", auc_fmodel_value, "\n")
```

```
## AUC for Final Model: 0.8100583
```

Residual Plots For all models, plots are a similar that shows a random scatter of residuals around the zero line without any clear pattern. This suggests non-linearity or other issues with the models fit. Compared to individual predictor plots, Intact Model and Final Model Residuals Plots shows a much tighter clustering of residuals around zero, which is a positive indication of the model's fit.

Residuals Analysis - **SAPS1 and SOFA Residuals**: The scatter is consistent across the range of the index, although some outliers are evident. The presence of outliers suggests these variables have some leverage, but they do not necessarily impact the model adversely unless reflected in the influence plots. - **Final and Intact Model Residuals**: Both models show a good distribution of residuals, though some points lie beyond the 2 standard deviation lines, especially in the intact model. This might suggest potential outliers or influential points which could be affecting the robustness of the model.

Influence Plots In the influence plots provided for each model, we see the standardized residuals plotted against the leverage (hat values) of each observation, with the size of the circles proportional to Cook's Distance. Most observations cluster around low leverage, which suggests that they have low leverage and limited influence on the model. 1. **SAPS1 Model Influence Plot**: There are a few observations with high leverage (206, 306, 1034), particularly one that stands out with a very high Cook's Distance. This suggests that this observation has a significant influence on the model. 2. **SOFA Model Influence Plot**: Similar to the SAPS1 model, there are points with notably higher leverage (206, 265), and one observation, in particular, has a high Cook's Distance, suggesting it has a strong influence on the model's predictions. 3. **Intact Model Influence Plot**: Observations like 206, 1175, and 1286 are notably influential. The high leverage (hat values) and Cook's distance suggest that these points are particularly impactful on the model's predictions. 4. **Final Model Influence Plot**: It similarly shows high influence for certain observations (206, 1175 and 1286). The pattern of influential points is consistent with those observed in the intact model, suggesting that these observations are inherently influential across different model specifications. 5. **Further Investigation**: Given that some indices are consistently appearing across different plots (206, 1175, 1286), it could be useful to investigate these points for data accuracy, potential errors, or unique characteristics.

ROC curves and AUC values An AUC close to 1 indicates a very good model, while an AUC closer to 0.5 suggests a less useful model. - Intact Model has the highest AUC (0.812), and shows the best ability to discriminate between patients who will have in-hospital death and those who will not. - Final Model has similar ROC curve to the Intact Model and maintains a strong discrimination, with an AUC of around 0.807. Suggesting that any refinements or adjustments made from the Intact to the Final model preserve its strong predictive power.

```
analyze_influential_points <- function(data, indices) {
  # Specify the columns of interest
  columns_of_interest <- c("Age", "SAPS1", "SOFA", "Albumin_max", "ALT_max", "AST_max",
                           "Bilirubin_max", "BUN_max", "Creatinine_max", "GCS_max", "GCS_min",
                           "HR_diff", "Lactate_max", "Na_diff", "NISysABP_diff", "PaO2_max",
                           "pH_diff", "Temp_diff", "Urine_max", "WBC_max")

  # Calculate summary statistics for the entire dataset for comparison
  summary_stats <- sapply(data[, columns_of_interest, drop = FALSE], function(x) {
    c(Min = min(x, na.rm = TRUE), Max = max(x, na.rm = TRUE))
  })

  # Define a threshold for "near" max or min (e.g., within 5%)
  threshold <- 0.05

  # Function to check if values are at or near the min/max
  check_near_min_max <- function(val, min, max) {
    list(AtMin = val == min || abs(val - min) <= threshold * abs(min),
         AtMax = val == max || abs(val - max) <= threshold * abs(max))
  }

  # Initialize a list to store results
  results <- list()

  # Analyze each point
  for (index in indices) {
    point_data <- data[index, c(columns_of_interest, "in_hospital_death"), drop = FALSE]
    survival_status <- ifelse(point_data["in_hospital_death"] == 1, "Died in hospital", "Surviv
or")
    individual_results <- c()

    for (var in columns_of_interest) {
      val <- point_data[[var]]
      min_max_check <- check_near_min_max(val, summary_stats["Min", var], summary_stats["Max",
var])
      if (min_max_check$AtMin || min_max_check$AtMax) {
        individual_results <- c(individual_results, sprintf("Index %d, %s: %f %s",
                                                            index, var, val,
                                                            ifelse(min_max_check$AtMin, "near/a
t Min", "near/at Max")))
      }
    }

    if (length(individual_results) > 0) {
      individual_results <- c(individual_results, sprintf("Index %d, %s", index, survival_statu
s))
      results[[length(results) + 1]] <- individual_results
    }
  }

  # Print results
  if (length(results) > 0) {
    cat("Significant Values at or near Min/Max:\n")
    for (result in results) {
      for (line in result) {
```

```
        cat(line, "\n")
      }
    }
  } else {
    cat("No significant values at or near Min/Max found for specified indices.\n")
  }
}

# Example usage with specified indices
indices <- c(206, 1175, 1286)  # indices of influential points
analyze_influential_points(icu_patients_df1_cleaned, indices)
```

```
## Significant Values at or near Min/Max:
## Index 206, SOFA: 0.000000 near/at Min
## Index 206, GCS_max: 15.000000 near/at Max
## Index 206, GCS_min: 15.000000 near/at Max
## Index 206, Died in hospital
## Index 1175, BUN_max: 197.000000 near/at Max
## Index 1175, Creatinine_max: 22.000000 near/at Max
## Index 1175, GCS_min: 3.000000 near/at Min
## Index 1175, Survivor
## Index 1286, AST_max: 16040.000000 near/at Max
## Index 1286, Died in hospital
```

After check influential_point 1175, we find that: - **BUN_max:**: Blood urea nitrogen measures the amount of nitrogen in the blood that comes from urea, a waste product processed by the kidneys, 197 is maximum in whole data - **Creatinine_max:** A maximum creatinine level of 22 is maximum in whole data. - **GCS_min:** The minimum Glasgow Coma Scale (GCS) score of 3 is the lowest possible score, indicating severe brain injury or deep unconsciousness. - **Urine_min:** no volume of urine output. - **In-hospital death (0):** Despite severe kidney dysfunction, deep unconsciousness, and possible respiratory distress, the patient survived the hospital stay.This is kind of counterintuitive by consider the clinical parameters. It's might be essential to review the medical records for such patients to understand the context better, validate the data accuracy, and decide on the inclusion in the analysis.

```
length(unique(icu_patients_df0$RecordID)) # Unique patient check
```

```
## [1] 2061
```

```
# Calculate the number of missing values for each column
missing_values_0 <- colSums(is.na(icu_patients_df0))
# Remove the variables that do not have missing values
missing_values_0 <- missing_values_0[missing_values_0 > 0]
# Print the variables with their count of missing values
print(missing_values_0)
```

```
##            SAPS1         Survival     Albumin_diff      Albumin_max
##               96             1288             1353             1353
##      Albumin_min        ALP_diff         ALP_max         ALP_min
##             1353             1270             1270             1270
##         ALT_diff         ALT_max         ALT_min         AST_diff
##             1256             1256             1256             1255
##         AST_max         AST_min   Bilirubin_diff   Bilirubin_max
##             1255             1255             1273             1273
##    Bilirubin_min        BUN_diff         BUN_max         BUN_min
##             1273               48               48               48
## Cholesterol_diff Cholesterol_max Cholesterol_min Creatinine_diff
##             1930             1930             1930               49
##   Creatinine_max   Creatinine_min     DiasABP_diff     DiasABP_max
##               49               49              702              702
##      DiasABP_min        FiO2_diff        FiO2_max        FiO2_min
##              702              759              759              759
##         GCS_diff         GCS_max         GCS_min           Gender
##               40               40               40                2
##     Glucose_diff      Glucose_max      Glucose_min        HCO3_diff
##              127              127              127               64
##         HCO3_max         HCO3_min         HCT_diff         HCT_max
##               64               64               55               55
##         HCT_min         HR_diff          HR_max          HR_min
##               55               37               37               37
##          K_diff           K_max           K_min    Lactate_diff
##              111              111              111             1071
##      Lactate_max      Lactate_min         MAP_diff         MAP_max
##             1071             1071              711              711
##         MAP_min         Mg_diff          Mg_max          Mg_min
##              711              137              137              137
##         Na_diff          Na_max          Na_min   NIDiasABP_diff
##               74               74               74              453
##    NIDiasABP_max    NIDiasABP_min       NIMAP_diff       NIMAP_max
##              453              453              454              454
##        NIMAP_min    NISysABP_diff    NISysABP_max    NISysABP_min
##              454              448              448              448
##       PaCO2_diff       PaCO2_max       PaCO2_min       PaO2_diff
##              603              603              603              603
##        PaO2_max        PaO2_min         pH_diff          pH_max
##              603              603              597              597
##          pH_min   Platelets_diff    Platelets_max    Platelets_min
##              597               68               68               68
##    RespRate_diff     RespRate_max     RespRate_min        SaO2_diff
##             1506             1506             1506             1299
##        SaO2_max         SaO2_min      SysABP_diff      SysABP_max
##             1299             1299              702              702
##       SysABP_min        Temp_diff        Temp_max        Temp_min
##              702               39               39               39
##   TroponinI_diff    TroponinI_max    TroponinI_min   TroponinT_diff
##             1955             1955             1955             1611
##    TroponinT_max    TroponinT_min       Urine_diff       Urine_max
##             1611             1611               70               70
##        Urine_min         WBC_diff         WBC_max         WBC_min
##               70               76               76               76
```

```
# Initial type conversions and handling of illogical variables
icu_patients_df0_conv <- icu_patients_df0 %>%
  mutate(
    SAPS1 = if_else(SAPS1 < 0, NA_real_, as.integer(SAPS1)),
    SOFA = if_else(SOFA < 0, NA_real_, as.integer(SOFA)),
    Length_of_stay = if_else(Length_of_stay < 0, abs(Length_of_stay), Length_of_stay),
    Age = as.integer(Age),
    GCS_max = as.integer(GCS_max),
    Gender = as.factor(Gender),
    ICUType = as.factor(ICUType),
    Status_Label = factor(Status, levels = c(FALSE, TRUE), labels = c("Survived", "Died")),
    LR_in_hospital_death = factor(in_hospital_death, levels = c(0, 1), labels = c("Survived",
"Died"))
  )


# Set a threshold for maximum acceptable proportion of missing data
missing_threshold <- 0.3
# Calculate the proportion of missing data for each variable
prop_missing_df0 <- colSums(is.na(icu_patients_df0_conv)) / nrow(icu_patients_df0_conv)
# Identify variables to exclude based on the threshold
vars_to_check_df0 <- setdiff(names(icu_patients_df0_conv), "Survival") # Exclude the 'Survival'
variable from the evaluation for missing data
vars_to_exclude_df0 <- names(prop_missing_df0[vars_to_check_df0][prop_missing_df0[vars_to_check
_df0] > missing_threshold])
print("Variables to be excluded due to high missing data:")
```

```
## [1] "Variables to be excluded due to high missing data:"
```

```
print(vars_to_exclude_df0)
```

```
##  [1] "Albumin_diff"     "Albumin_max"      "Albumin_min"      "ALP_diff"
##  [5] "ALP_max"          "ALP_min"          "ALT_diff"         "ALT_max"
##  [9] "ALT_min"          "AST_diff"         "AST_max"          "AST_min"
## [13] "Bilirubin_diff"   "Bilirubin_max"    "Bilirubin_min"    "Cholesterol_diff"
## [17] "Cholesterol_max"  "Cholesterol_min"  "DiasABP_diff"     "DiasABP_max"
## [21] "DiasABP_min"      "FiO2_diff"        "FiO2_max"         "FiO2_min"
## [25] "Lactate_diff"     "Lactate_max"      "Lactate_min"      "MAP_diff"
## [29] "MAP_max"          "MAP_min"          "RespRate_diff"    "RespRate_max"
## [33] "RespRate_min"     "SaO2_diff"        "SaO2_max"         "SaO2_min"
## [37] "SysABP_diff"      "SysABP_max"       "SysABP_min"       "TroponinI_diff"
## [41] "TroponinI_max"    "TroponinI_min"    "TroponinT_diff"   "TroponinT_max"
## [45] "TroponinT_min"
```

```
# Variables intended for the model
model_vars_df0 <- c("Age", "SAPS1", "SOFA", "Albumin_max", "ALT_max", "AST_max",
                        "Bilirubin_max", "BUN_max", "Creatinine_max", "GCS_max", "GCS_min",
                        "HR_diff", "Lactate_max", "Na_diff", "NISysABP_diff", "PaO2_max",
                        "pH_diff", "Temp_diff", "Urine_max", "WBC_max", "ICUType")

# Check if any model variables should be excluded due to high missing data
excluded_model_vars <- intersect(model_vars_df0, vars_to_exclude_df0)
cat("\nVariables intended for the model have high missing data and will be excluded:\n")
```

```
##
## Variables intended for the model have high missing data and will be excluded:
```

```
print(excluded_model_vars)
```

```
## [1] "Albumin_max"   "ALT_max"       "AST_max"       "Bilirubin_max"
## [5] "Lactate_max"
```

missing data When attempting to refit the 'final_model3' to the unimputed data frame "icu_patients_df0" directlly, a significant problem was encountered due to missing data. - The presence of NA values in the model summary for most coefficients suggests that the glm function could not estimate the model parameters. - With predictor variables like `Albumin_max` missing over 1353 values, `ALT_max` missing 1256, and `AST_max` missing 1255, as well as `Lactate_max` missing 1071, there's a substantial loss of information. - Several predictor variables had even more than 30% of their values missing, making it impractical to include them in the model without some form of imputation. - For those variables have more than 30% missing data, might consider dropping it from the model entirely, especially if it is not a key predictor or if there is no easy way to impute its values accurately. **Thus, data clean is necessary before fit model.**

It appears that several key liver function indicators, including `Albumin_max`, `ALT_max`, `AST_max`, and `Bilirubin_max`, are among the variables with a high proportion of missing data in the dataset, and thus are recommended for exclusion from the analysis. Additionally, `Lactate_max`, which is a critical marker of metabolic status indicate of how well oxygen is being used by the body and can be reflective of liver function and other physiological states, is also flagged for exclusion due to high missingness.

```
# Identify variables to impute and ensure they are numeric
vars_to_impute_fd0 <- names(prop_missing_df0[prop_missing_df0 <= missing_threshold & prop_missi
ng_df0 > 0])
vars_to_impute_fd0 <- vars_to_impute_fd0[sapply(icu_patients_df0_conv[vars_to_impute_fd0], is.n
umeric)]  # Ensure variables are numeric
print(vars_to_impute_fd0)
```

```
##  [1] "SAPS1"           "SOFA"            "BUN_diff"        "BUN_max"
##  [5] "BUN_min"         "Creatinine_diff" "Creatinine_max"  "Creatinine_min"
##  [9] "GCS_diff"        "GCS_max"         "GCS_min"         "Glucose_diff"
## [13] "Glucose_max"     "Glucose_min"     "HCO3_diff"       "HCO3_max"
## [17] "HCO3_min"        "HCT_diff"        "HCT_max"         "HCT_min"
## [21] "HR_diff"         "HR_max"          "HR_min"          "K_diff"
## [25] "K_max"           "K_min"           "Mg_diff"         "Mg_max"
## [29] "Mg_min"          "Na_diff"         "Na_max"          "Na_min"
## [33] "NIDiasABP_diff"  "NIDiasABP_max"   "NIDiasABP_min"   "NIMAP_diff"
## [37] "NIMAP_max"       "NIMAP_min"       "NISysABP_diff"   "NISysABP_max"
## [41] "NISysABP_min"    "PaCO2_diff"      "PaCO2_max"       "PaCO2_min"
## [45] "PaO2_diff"       "PaO2_max"        "PaO2_min"        "pH_diff"
## [49] "pH_max"          "pH_min"          "Platelets_diff"  "Platelets_max"
## [53] "Platelets_min"   "Temp_diff"       "Temp_max"        "Temp_min"
## [57] "Urine_diff"      "Urine_max"       "Urine_min"       "WBC_diff"
## [61] "WBC_max"         "WBC_min"
```

```
# Make a copy of the data frame to clean
icu_patients_df0_cleaned <- icu_patients_df0_conv

# Impute missing values for selected variables using median
for(var in vars_to_impute_fd0) {
  icu_patients_df0_cleaned[[var]][is.na(icu_patients_df0_cleaned[[var]])] <- median(icu_patient
s_df0_cleaned[[var]], na.rm = TRUE)
}
# Now exclude the variables with too much missing data from the cleaned dataframe
icu_patients_df0_cleaned <- icu_patients_df0_cleaned[ , !(names(icu_patients_df0_cleaned) %in%
vars_to_exclude_df0)]

# Check if there are any remaining missing values
missing_values <- colSums(is.na(icu_patients_df0_cleaned))
missing_values <- missing_values[missing_values > 0]
print("Remaining missing values in the cleaned data:")
```

```
## [1] "Remaining missing values in the cleaned data:"
```

```
print(missing_values)
```

```
## Survival   Gender
##     1288        2
```

```
# Print the number of observations after cleaning
print(paste("Number of observations after cleaning:", nrow(icu_patients_df0_cleaned)))
```

```
## [1] "Number of observations after cleaning: 2061"
```

```
# Check for NaN, or Infinite values in the dataset for the specified variables
# Calculate the sum of NaN, and infinite values for each specified model variable
nan_inf_counts <- sapply(icu_patients_df0[model_vars_df0], function(x) {
  sum(is.nan(x) | is.infinite(x), na.rm = TRUE)  # Add na.rm = TRUE to ensure NAs are not consi
dered in other calculations
})

# Print the results
print("Sum of NaN, and Infinite values for each variable:")
```

```
## [1] "Sum of NaN, and Infinite values for each variable:"
```

```
print(nan_inf_counts)
```

```
##            Age          SAPS1           SOFA    Albumin_max        ALT_max
##              0              0              0              0              0
##        AST_max  Bilirubin_max        BUN_max Creatinine_max        GCS_max
##              0              0              0              0              0
##        GCS_min        HR_diff    Lactate_max        Na_diff  NISysABP_diff
##              0              0              0              0              5
##       PaO2_max        pH_diff      Temp_diff      Urine_max        WBC_max
##              0              0              0              0              0
##        ICUType
##              0
```

```
print("Unique problematic values in NISysABP_diff:")
```

```
## [1] "Unique problematic values in NISysABP_diff:"
```

```
print(unique(icu_patients_df0_cleaned$NISysABP_diff[is.nan(icu_patients_df0_cleaned$NISysABP_di
ff) | is.infinite(icu_patients_df0_cleaned$NISysABP_diff)]))
```

```
## [1] Inf
```

```
# Remove rows where 'NISysABP_diff' is infinite
icu_patients_df0_cleaned <- icu_patients_df0_cleaned[!is.infinite(icu_patients_df0_cleaned$NISy
sABP_diff), ]
```

```
# Refit the model using modified predictor set  (remove "Albumin_max" "ALT_max" "AST_max" "Bili
rubin_max" "Lactate_max")
fmodel_df0_1 <- glm(in_hospital_death ~ Age + SAPS1 + SOFA +  BUN_max  + Creatinine_max + GCS_m
ax + GCS_min
                         + HR_diff + Na_diff + NISysABP_diff + PaO2_max + pH_diff  + Temp_diff
+ Urine_max + WBC_max + ICUType,
                    family = binomial(link = "logit"), data = icu_patients_df0_cleaned)
summary(fmodel_df0_1)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Age + SAPS1 + SOFA + BUN_max +
##     Creatinine_max + GCS_max + GCS_min + HR_diff + Na_diff +
##     NISysABP_diff + PaO2_max + pH_diff + Temp_diff + Urine_max +
##     WBC_max + ICUType, family = binomial(link = "logit"), data = icu_patients_df0_cleaned)
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -3.7235171  0.6673232  -5.580 2.41e-08 ***
## Age                               0.0226616  0.0050938   4.449 8.63e-06 ***
## SAPS1                             0.0522765  0.0249345   2.097 0.036033 *
## SOFA                              0.0865036  0.0272495   3.175 0.001501 **
## BUN_max                           0.0168784  0.0038235   4.414 1.01e-05 ***
## Creatinine_max                   -0.1159434  0.0543077  -2.135 0.032766 *
## GCS_max                          -0.1695429  0.0275448  -6.155 7.50e-10 ***
## GCS_min                           0.0707941  0.0278368   2.543 0.010985 *
## HR_diff                           0.0069745  0.0042804   1.629 0.103230
## Na_diff                           0.0135620  0.0184180   0.736 0.461521
## NISysABP_diff                     0.0070530  0.0039362   1.792 0.073158 .
## PaO2_max                         -0.0017387  0.0008124  -2.140 0.032346 *
## pH_diff                           3.1154134  1.2335000   2.526 0.011548 *
## Temp_diff                         0.0484354  0.0959280   0.505 0.613619
## Urine_max                        -0.0007199  0.0002105  -3.421 0.000625 ***
## WBC_max                          -0.0071681  0.0083670  -0.857 0.391605
## ICUTypeCardiac Surgery Recovery Unit -0.6483220  0.3096858  -2.093 0.036306 *
## ICUTypeMedical ICU                0.0139443  0.2092073   0.067 0.946858
## ICUTypeSurgical ICU               0.0959875  0.2314028   0.415 0.678283
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1698.1  on 2055  degrees of freedom
## Residual deviance: 1364.6  on 2037  degrees of freedom
## AIC: 1402.6
##
## Number of Fisher Scoring iterations: 6
```

```
# Refit the model using modified predictor set  (add "Glucose_max" "Urine_min" which are candidate predictors and not excluded by high proportion of missing data)
fmodel_df0_2 <- glm(in_hospital_death ~ Age + SAPS1 + SOFA +  BUN_max  + Creatinine_max + GCS_max + GCS_min + Glucose_max
                          + HR_diff + Na_diff + NISysABP_diff + PaO2_max + pH_diff  + Temp_diff + Urine_max + Urine_min + WBC_max + ICUType,
                    family = binomial(link = "logit"), data = icu_patients_df0_cleaned)
summary(fmodel_df0_2)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Age + SAPS1 + SOFA + BUN_max +
##     Creatinine_max + GCS_max + GCS_min + Glucose_max + HR_diff +
##     Na_diff + NISysABP_diff + PaO2_max + pH_diff + Temp_diff +
##     Urine_max + Urine_min + WBC_max + ICUType, family = binomial(link = "logit"),
##     data = icu_patients_df0_cleaned)
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -3.597e+00  6.799e-01  -5.290 1.22e-07 ***
## Age                                2.204e-02  5.131e-03   4.296 1.74e-05 ***
## SAPS1                              5.189e-02  2.543e-02   2.040  0.04135 *
## SOFA                               8.347e-02  2.746e-02   3.040  0.00236 **
## BUN_max                            1.712e-02  3.838e-03   4.462 8.13e-06 ***
## Creatinine_max                    -1.187e-01  5.443e-02  -2.181  0.02916 *
## GCS_max                           -1.698e-01  2.750e-02  -6.176 6.57e-10 ***
## GCS_min                            7.259e-02  2.791e-02   2.601  0.00931 **
## Glucose_max                       -9.494e-05  8.024e-04  -0.118  0.90581
## HR_diff                            6.910e-03  4.288e-03   1.611  0.10712
## Na_diff                            1.451e-02  1.906e-02   0.761  0.44654
## NISysABP_diff                      7.017e-03  3.936e-03   1.783  0.07463 .
## PaO2_max                          -1.742e-03  8.129e-04  -2.143  0.03214 *
## pH_diff                            3.108e+00  1.236e+00   2.514  0.01194 *
## Temp_diff                          4.971e-02  9.603e-02   0.518  0.60474
## Urine_max                         -6.845e-04  2.105e-04  -3.252  0.00115 **
## Urine_min                         -1.999e-03  1.902e-03  -1.051  0.29330
## WBC_max                           -7.481e-03  8.386e-03  -0.892  0.37236
## ICUTypeCardiac Surgery Recovery Unit -6.781e-01  3.156e-01  -2.149  0.03166 *
## ICUTypeMedical ICU                 1.032e-02  2.113e-01   0.049  0.96105
## ICUTypeSurgical ICU                8.245e-02  2.331e-01   0.354  0.72352
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1698.1  on 2055  degrees of freedom
## Residual deviance: 1363.4  on 2035  degrees of freedom
## AIC: 1405.4
##
## Number of Fisher Scoring iterations: 6
```

```
# Comparing models with ANOVA
anova_results_df0 <- anova(fmodel_df0_1, fmodel_df0_2, test = "Chisq")
print(anova_results_df0)
```

```
## Analysis of Deviance Table
##
## Model 1: in_hospital_death ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + HR_diff + Na_diff + NISysABP_diff + PaO2_max +
##     pH_diff + Temp_diff + Urine_max + WBC_max + ICUType
## Model 2: in_hospital_death ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + Glucose_max + HR_diff + Na_diff + NISysABP_diff +
##     PaO2_max + pH_diff + Temp_diff + Urine_max + Urine_min +
##     WBC_max + ICUType
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      2037     1364.6
## 2      2035     1363.4  2   1.2416   0.5375
```

Basicly no different after adding two candidate predictors

```
# Creating two-way interaction models from previous models
two_way_model <- update(fmodel_df0_1, . ~ .^2)
# Dropping each two-way interaction term to test significance
drop_interaction_result <- drop1(two_way_model, test = "Chisq")
drop_interaction_result
```

```
## Single term deletions
##
## Model:
## in_hospital_death ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + HR_diff + Na_diff + NISysABP_diff + PaO2_max +
##     pH_diff + Temp_diff + Urine_max + WBC_max + ICUType + Age:SAPS1 +
##     Age:SOFA + Age:BUN_max + Age:Creatinine_max + Age:GCS_max +
##     Age:GCS_min + Age:HR_diff + Age:Na_diff + Age:NISysABP_diff +
##     Age:PaO2_max + Age:pH_diff + Age:Temp_diff + Age:Urine_max +
##     Age:WBC_max + Age:ICUType + SAPS1:SOFA + SAPS1:BUN_max +
##     SAPS1:Creatinine_max + SAPS1:GCS_max + SAPS1:GCS_min + SAPS1:HR_diff +
##     SAPS1:Na_diff + SAPS1:NISysABP_diff + SAPS1:PaO2_max + SAPS1:pH_diff +
##     SAPS1:Temp_diff + SAPS1:Urine_max + SAPS1:WBC_max + SAPS1:ICUType +
##     SOFA:BUN_max + SOFA:Creatinine_max + SOFA:GCS_max + SOFA:GCS_min +
##     SOFA:HR_diff + SOFA:Na_diff + SOFA:NISysABP_diff + SOFA:PaO2_max +
##     SOFA:pH_diff + SOFA:Temp_diff + SOFA:Urine_max + SOFA:WBC_max +
##     SOFA:ICUType + BUN_max:Creatinine_max + BUN_max:GCS_max +
##     BUN_max:GCS_min + BUN_max:HR_diff + BUN_max:Na_diff + BUN_max:NISysABP_diff +
##     BUN_max:PaO2_max + BUN_max:pH_diff + BUN_max:Temp_diff +
##     BUN_max:Urine_max + BUN_max:WBC_max + BUN_max:ICUType + Creatinine_max:GCS_max +
##     Creatinine_max:GCS_min + Creatinine_max:HR_diff + Creatinine_max:Na_diff +
##     Creatinine_max:NISysABP_diff + Creatinine_max:PaO2_max +
##     Creatinine_max:pH_diff + Creatinine_max:Temp_diff + Creatinine_max:Urine_max +
##     Creatinine_max:WBC_max + Creatinine_max:ICUType + GCS_max:GCS_min +
##     GCS_max:HR_diff + GCS_max:Na_diff + GCS_max:NISysABP_diff +
##     GCS_max:PaO2_max + GCS_max:pH_diff + GCS_max:Temp_diff +
##     GCS_max:Urine_max + GCS_max:WBC_max + GCS_max:ICUType + GCS_min:HR_diff +
##     GCS_min:Na_diff + GCS_min:NISysABP_diff + GCS_min:PaO2_max +
##     GCS_min:pH_diff + GCS_min:Temp_diff + GCS_min:Urine_max +
##     GCS_min:WBC_max + GCS_min:ICUType + HR_diff:Na_diff + HR_diff:NISysABP_diff +
##     HR_diff:PaO2_max + HR_diff:pH_diff + HR_diff:Temp_diff +
##     HR_diff:Urine_max + HR_diff:WBC_max + HR_diff:ICUType + Na_diff:NISysABP_diff +
##     Na_diff:PaO2_max + Na_diff:pH_diff + Na_diff:Temp_diff +
##     Na_diff:Urine_max + Na_diff:WBC_max + Na_diff:ICUType + NISysABP_diff:PaO2_max +
##     NISysABP_diff:pH_diff + NISysABP_diff:Temp_diff + NISysABP_diff:Urine_max +
##     NISysABP_diff:WBC_max + NISysABP_diff:ICUType + PaO2_max:pH_diff +
##     PaO2_max:Temp_diff + PaO2_max:Urine_max + PaO2_max:WBC_max +
##     PaO2_max:ICUType + pH_diff:Temp_diff + pH_diff:Urine_max +
##     pH_diff:WBC_max + pH_diff:ICUType + Temp_diff:Urine_max +
##     Temp_diff:WBC_max + Temp_diff:ICUType + Urine_max:WBC_max +
##     Urine_max:ICUType + WBC_max:ICUType
##                              Df Deviance    AIC     LRT  Pr(>Chi)
## <none>                            1095.6 1433.6
## Age:SAPS1                     1   1095.6 1431.6  0.0514  0.820565
## Age:SOFA                      1   1102.0 1438.0  6.4391  0.011163 *
## Age:BUN_max                   1   1095.8 1431.8  0.2733  0.601121
## Age:Creatinine_max            1   1096.0 1432.0  0.4289  0.512508
## Age:GCS_max                   1   1096.4 1432.4  0.8047  0.369678
## Age:GCS_min                   1   1097.0 1433.0  1.4473  0.228961
## Age:HR_diff                   1   1095.8 1431.8  0.2087  0.647769
## Age:Na_diff                   1   1095.8 1431.8  0.2416  0.623088
## Age:NISysABP_diff             1   1097.0 1433.0  1.4277  0.232138
## Age:PaO2_max                  1   1095.7 1431.7  0.1082  0.742146
## Age:pH_diff                   1   1096.5 1432.5  0.9575  0.327827
## Age:Temp_diff                 1   1095.9 1431.9  0.2878  0.591635
```

```
## Age:Urine_max                 1   1096.4 1432.4  0.8024  0.370388
## Age:WBC_max                   1   1095.7 1431.7  0.1242  0.724501
## Age:ICUType                   3   1099.2 1431.2  3.5807  0.310452
## SAPS1:SOFA                    1   1096.8 1432.8  1.2277  0.267858
## SAPS1:BUN_max                 1   1095.8 1431.8  0.2726  0.601591
## SAPS1:Creatinine_max          1   1096.2 1432.2  0.6441  0.422212
## SAPS1:GCS_max                 1   1096.8 1432.8  1.2433  0.264836
## SAPS1:GCS_min                 1   1097.5 1433.5  1.9717  0.160268
## SAPS1:HR_diff                 1   1099.6 1435.6  4.0053  0.045358 *
## SAPS1:Na_diff                 1   1098.7 1434.7  3.1348  0.076639 .
## SAPS1:NISysABP_diff           1   1100.9 1436.9  5.3281  0.020984 *
## SAPS1:PaO2_max                1   1095.9 1431.9  0.3485  0.554940
## SAPS1:pH_diff                 1   1100.0 1436.0  4.3817  0.036327 *
## SAPS1:Temp_diff               1   1095.6 1431.6  0.0525  0.818765
## SAPS1:Urine_max               1   1099.7 1435.7  4.0892  0.043157 *
## SAPS1:WBC_max                 1   1095.6 1431.6  0.0457  0.830632
## SAPS1:ICUType                 3   1099.5 1431.5  3.8735  0.275448
## SOFA:BUN_max                  1   1095.6 1431.6  0.0421  0.837512
## SOFA:Creatinine_max           1   1097.7 1433.7  2.0995  0.147351
## SOFA:GCS_max                  1   1098.5 1434.5  2.9401  0.086406 .
## SOFA:GCS_min                  1   1095.6 1431.6  0.0157  0.900212
## SOFA:HR_diff                  1   1095.7 1431.7  0.0797  0.777753
## SOFA:Na_diff                  1   1095.7 1431.7  0.1101  0.739974
## SOFA:NISysABP_diff            1   1096.7 1432.7  1.0826  0.298117
## SOFA:PaO2_max                 1   1095.7 1431.7  0.0729  0.787096
## SOFA:pH_diff                  1   1113.0 1449.0 17.4745 2.912e-05 ***
## SOFA:Temp_diff                1   1099.7 1435.7  4.1595  0.041403 *
## SOFA:Urine_max                1   1101.1 1437.1  5.5438  0.018546 *
## SOFA:WBC_max                  1   1108.2 1444.2 12.6377  0.000378 ***
## SOFA:ICUType                  3   1098.1 1430.1  2.5478  0.466721
## BUN_max:Creatinine_max        1   1097.0 1433.0  1.4104  0.234983
## BUN_max:GCS_max               1   1096.7 1432.7  1.0965  0.295032
## BUN_max:GCS_min               1   1098.7 1434.7  3.1350  0.076627 .
## BUN_max:HR_diff               1   1096.9 1432.9  1.3637  0.242896
## BUN_max:Na_diff               1   1095.7 1431.7  0.0701  0.791122
## BUN_max:NISysABP_diff         1   1096.3 1432.3  0.7492  0.386743
## BUN_max:PaO2_max              1   1095.6 1431.6  0.0563  0.812386
## BUN_max:pH_diff               1   1096.9 1432.9  1.3604  0.243462
## BUN_max:Temp_diff             1   1095.7 1431.7  0.0912  0.762701
## BUN_max:Urine_max             1   1098.8 1434.8  3.2318  0.072223 .
## BUN_max:WBC_max               1   1096.0 1432.0  0.4716  0.492255
## BUN_max:ICUType               3   1105.2 1437.2  9.6499  0.021789 *
## Creatinine_max:GCS_max        1   1098.1 1434.1  2.5086  0.113229
## Creatinine_max:GCS_min        1   1099.3 1435.3  3.7301  0.053439 .
## Creatinine_max:HR_diff        1   1096.0 1432.0  0.4729  0.491672
## Creatinine_max:Na_diff        1   1095.8 1431.8  0.2086  0.647882
## Creatinine_max:NISysABP_diff  1   1095.7 1431.7  0.1132  0.736579
## Creatinine_max:PaO2_max       1   1097.6 1433.6  2.0032  0.156969
## Creatinine_max:pH_diff        1   1096.5 1432.5  0.9647  0.326015
## Creatinine_max:Temp_diff      1   1095.7 1431.7  0.1516  0.697020
## Creatinine_max:Urine_max      1   1096.0 1432.0  0.4033  0.525404
## Creatinine_max:WBC_max        1   1099.2 1435.2  3.5976  0.057862 .
## Creatinine_max:ICUType        3   1096.4 1428.4  0.8650  0.833876
## GCS_max:GCS_min               1   1098.2 1434.2  2.6331  0.104658
## GCS_max:HR_diff               1   1097.9 1433.9  2.2911  0.130120
## GCS_max:Na_diff               1   1095.7 1431.7  0.0928  0.760703
```

```
## GCS_max:NISysABP_diff         1   1102.1 1438.1  6.5418  0.010537 *
## GCS_max:PaO2_max              1   1098.6 1434.6  2.9811  0.084241 .
## GCS_max:pH_diff               1   1097.9 1433.9  2.2794  0.131099
## GCS_max:Temp_diff             1   1095.7 1431.7  0.1631  0.686296
## GCS_max:Urine_max             1   1096.2 1432.2  0.5708  0.449934
## GCS_max:WBC_max               1   1095.7 1431.7  0.1457  0.702674
## GCS_max:ICUType               3   1100.5 1432.5  4.9409  0.176173
## GCS_min:HR_diff               1   1097.9 1433.9  2.3485  0.125406
## GCS_min:Na_diff               1   1096.4 1432.4  0.8234  0.364183
## GCS_min:NISysABP_diff         1   1099.5 1435.5  3.8796  0.048875 *
## GCS_min:PaO2_max              1   1095.7 1431.7  0.0910  0.762937
## GCS_min:pH_diff               1   1096.3 1432.3  0.6965  0.403964
## GCS_min:Temp_diff             1   1096.6 1432.6  0.9991  0.317539
## GCS_min:Urine_max             1   1095.6 1431.6  0.0251  0.874003
## GCS_min:WBC_max               1   1103.6 1439.6  7.9944  0.004692 **
## GCS_min:ICUType               3   1100.6 1432.6  4.9833  0.173026
## HR_diff:Na_diff               1   1097.5 1433.5  1.9006  0.168010
## HR_diff:NISysABP_diff         1   1095.6 1431.6  0.0553  0.814145
## HR_diff:PaO2_max              1   1097.4 1433.4  1.8547  0.173236
## HR_diff:pH_diff               1   1095.7 1431.7  0.1130  0.736740
## HR_diff:Temp_diff             1   1095.6 1431.6  0.0014  0.970406
## HR_diff:Urine_max             1   1098.5 1434.5  2.9415  0.086333 .
## HR_diff:WBC_max               1   1099.2 1435.2  3.5852  0.058295 .
## HR_diff:ICUType               3   1101.6 1433.6  5.9942  0.111893
## Na_diff:NISysABP_diff         1   1099.2 1435.2  3.6680  0.055466 .
## Na_diff:PaO2_max              1   1102.8 1438.8  7.2028  0.007279 **
## Na_diff:pH_diff               1   1100.0 1436.0  4.4539  0.034822 *
## Na_diff:Temp_diff             1   1097.0 1433.0  1.4565  0.227483
## Na_diff:Urine_max             1   1096.1 1432.1  0.5279  0.467485
## Na_diff:WBC_max               1   1096.4 1432.4  0.7837  0.376003
## Na_diff:ICUType               3   1103.0 1435.0  7.4642  0.058485 .
## NISysABP_diff:PaO2_max        1   1095.8 1431.8  0.1823  0.669420
## NISysABP_diff:pH_diff         1   1096.6 1432.6  1.0521  0.305013
## NISysABP_diff:Temp_diff       1   1096.6 1432.6  1.0206  0.312382
## NISysABP_diff:Urine_max       1   1096.6 1432.6  1.0137  0.314012
## NISysABP_diff:WBC_max         1   1101.7 1437.7  6.1225  0.013347 *
## NISysABP_diff:ICUType         3   1098.2 1430.2  2.6338  0.451600
## PaO2_max:pH_diff              1   1096.0 1432.0  0.4220  0.515937
## PaO2_max:Temp_diff            1   1095.8 1431.8  0.1886  0.664082
## PaO2_max:Urine_max            1   1095.7 1431.7  0.1469  0.701539
## PaO2_max:WBC_max              1   1099.5 1435.5  3.9199  0.047718 *
## PaO2_max:ICUType              3   1100.0 1432.0  4.3868  0.222614
## pH_diff:Temp_diff             1   1098.5 1434.5  2.9184  0.087576 .
## pH_diff:Urine_max             1   1095.9 1431.9  0.3471  0.555782
## pH_diff:WBC_max               1   1096.6 1432.6  1.0025  0.316714
## pH_diff:ICUType               3   1100.8 1432.8  5.1943  0.158113
## Temp_diff:Urine_max           1   1098.7 1434.7  3.0933  0.078615 .
## Temp_diff:WBC_max             1   1096.6 1432.6  1.0278  0.310672
## Temp_diff:ICUType             3   1101.0 1433.0  5.3896  0.145393
## Urine_max:WBC_max             1   1096.5 1432.5  0.9637  0.326245
## Urine_max:ICUType             3   1097.0 1429.0  1.4521  0.693364
## WBC_max:ICUType               3   1097.5 1429.5  1.9301  0.587035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Refit the model using modified predictor set  (add "SOFA:pH_diff" "SOFA:WBC_max" which are si
gnificant in two-way interaction model)
fmodel_df0_3 <- glm(in_hospital_death ~ Age + SAPS1 + SOFA +  BUN_max  + Creatinine_max + GCS_m
ax + GCS_min
                         + HR_diff + Na_diff + NISysABP_diff + PaO2_max + pH_diff  + Temp_diff
+ Urine_max + SOFA:pH_diff + SOFA:WBC_max + WBC_max + ICUType,
                    family = binomial(link = "logit"), data = icu_patients_df0_cleaned)
summary(fmodel_df0_3)
```

```
##
## Call:
## glm(formula = in_hospital_death ~ Age + SAPS1 + SOFA + BUN_max +
##     Creatinine_max + GCS_max + GCS_min + HR_diff + Na_diff +
##     NISysABP_diff + PaO2_max + pH_diff + Temp_diff + Urine_max +
##     SOFA:pH_diff + SOFA:WBC_max + WBC_max + ICUType, family = binomial(link = "logit"),
##     data = icu_patients_df0_cleaned)
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -3.4799837  0.7684367  -4.529 5.94e-06 ***
## Age                                  0.0236112  0.0051397   4.594 4.35e-06 ***
## SAPS1                                0.0478443  0.0251620   1.901 0.057243 .
## SOFA                                 0.0756475  0.0479496   1.578 0.114647
## BUN_max                              0.0172677  0.0038413   4.495 6.95e-06 ***
## Creatinine_max                      -0.1139042  0.0546732  -2.083 0.037218 *
## GCS_max                             -0.1696093  0.0277451  -6.113 9.77e-10 ***
## GCS_min                              0.0621560  0.0281931   2.205 0.027479 *
## HR_diff                              0.0063438  0.0043282   1.466 0.142729
## Na_diff                              0.0143355  0.0184481   0.777 0.437117
## NISysABP_diff                        0.0068984  0.0039452   1.749 0.080370 .
## PaO2_max                            -0.0016464  0.0008195  -2.009 0.044529 *
## pH_diff                             -3.3102338  3.2559069  -1.017 0.309303
## Temp_diff                            0.0398843  0.0992578   0.402 0.687813
## Urine_max                           -0.0007066  0.0002119  -3.335 0.000852 ***
## WBC_max                              0.0298206  0.0194050   1.537 0.124355
## ICUTypeCardiac Surgery Recovery Unit -0.6933112  0.3111812  -2.228 0.025881 *
## ICUTypeMedical ICU                   0.0026752  0.2092925   0.013 0.989802
## ICUTypeSurgical ICU                  0.0529317  0.2322147   0.228 0.819691
## SOFA:pH_diff                         0.6453096  0.2926166   2.205 0.027433 *
## SOFA:WBC_max                        -0.0043416  0.0020745  -2.093 0.036359 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1698.1  on 2055  degrees of freedom
## Residual deviance: 1356.4  on 2035  degrees of freedom
## AIC: 1398.4
##
## Number of Fisher Scoring iterations: 6
```

```r
# Comparing models with ANOVA
anova_results2_df0 <- anova(fmodel_df0_1, fmodel_df0_3, test = "Chisq")
print(anova_results2_df0)
```

```
## Analysis of Deviance Table
##
## Model 1: in_hospital_death ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + HR_diff + Na_diff + NISysABP_diff + PaO2_max +
##     pH_diff + Temp_diff + Urine_max + WBC_max + ICUType
## Model 2: in_hospital_death ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + HR_diff + Na_diff + NISysABP_diff + PaO2_max +
##     pH_diff + Temp_diff + Urine_max + SOFA:pH_diff + SOFA:WBC_max +
##     WBC_max + ICUType
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      2037     1364.6
## 2      2035     1356.4  2   8.1874  0.01668 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Multicollinearity check by Variance Inflation Factor (VIF)
vif_values_fm <- vif(fmodel_df0_1, type = 'terms')
vif_values_fm3 <- vif(fmodel_df0_3, type = 'terms')
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```r
cat("VIF Results for Final Model:\n")
```

```
## VIF Results for Final Model:
```

```r
print(vif_values_fm)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## Age           1.274372  1        1.128881
## SAPS1         2.865233  1        1.692700
## SOFA          2.496149  1        1.579921
## BUN_max       2.189192  1        1.479592
## Creatinine_max 2.051455  1        1.432290
## GCS_max       1.842209  1        1.357280
## GCS_min       3.308384  1        1.818896
## HR_diff       1.152274  1        1.073440
## Na_diff       1.106316  1        1.051816
## NISysABP_diff  1.131424  1        1.063684
## PaO2_max      1.331500  1        1.153907
## pH_diff       1.238742  1        1.112988
## Temp_diff     1.268614  1        1.126328
## Urine_max     1.129558  1        1.062806
## WBC_max       1.100461  1        1.049029
## ICUType       1.638310  3        1.085757
```

```
cat("\n")
```

```
cat("VIF Results for Final Model 3:\n")
```

```
## VIF Results for Final Model 3:
```

```
print(vif_values_fm3)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## Age              1.279855  1        1.131307
## SAPS1            2.859438  1        1.690987
## SOFA             7.604720  1        2.757666
## BUN_max          2.195848  1        1.481839
## Creatinine_max   2.067651  1        1.437933
## GCS_max          1.851756  1        1.360792
## GCS_min          3.383533  1        1.839438
## HR_diff          1.154096  1        1.074289
## Na_diff          1.109213  1        1.053192
## NISysABP_diff    1.130968  1        1.063470
## PaO2_max         1.346751  1        1.160496
## pH_diff          7.992425  1        2.827088
## Temp_diff        1.265734  1        1.125048
## Urine_max        1.128684  1        1.062395
## WBC_max          5.639475  1        2.374758
## ICUType          1.638797  3        1.085811
## SOFA:pH_diff    14.435688  1        3.799433
## SOFA:WBC_max     9.560577  1        3.092018
```

```
cat("\n")
```

- **Residual Deviance**: Lower residual deviance in `fmodel_df0_3` (1358.2 on 2040 degrees of freedom) compared to `fmodel_df0` (1364.6 on 2037 degrees of freedom) suggests that the model with the interaction terms explains more of the variability in the data.
- **AIC (Akaike Information Criterion)**: The AIC has slightly decreased from 1402.6 in `fmodel_df0` to 1400.2 in `fmodel_df0_3`. This decrease, although modest, indicates a better model fit when accounting for the penalty of additional predictors.
- **Deviance Analysis**: The deviance reduction of 10.412 with 2 degrees of freedom between the models is statistically significant (p = 0.005485), suggesting that the interaction terms contribute significantly to the model.
- **Clinical Relevance**: The significant interactions (`SOFA:pH_diff` and `SOFA:WBC_max`) suggest that the effect of SOFA (Sequential Organ Failure Assessment) scores on in-hospital death risk is modified by changes in pH and WBC counts. This could indicate more complex relationships between organ failure, acid-base balance, and immune response in critically ill patients, which could be considered in clinical assessments and interventions.

VIF values 1. The VIF values common thresholds are 5 or 10, below that suggests that there isn't a concerning level of multicollinearity among the main effects in the model. 1. High GVIF values for interaction terms are not uncommon, as these terms are products of their component variables and can inherit their collinearity. 2. High VIF/GVIF values do not imply that a model is invalid; rather, they suggest that the precision of the coefficient estimates for the related variables may be reduced.

```
# Using dplyr to filter out cases where the specified variables are NA
#valid_data0 <- icu_patients_df0 %>% filter(!is.na(SAPS1) & !is.na(GCS_min) & !is.na(Creatinine_max))


# Logistic regression to see the effect of SAPS1 and SOFA scores on in-hospital death
model0_saps1 <- glm(in_hospital_death ~ SAPS1, data = icu_patients_df0_cleaned, family = binomial)
model0_sofa <- glm(in_hospital_death ~ SOFA, data = icu_patients_df0_cleaned, family = binomial)

# Predicted probabilities for the models
pred0_saps1 <- predict(model0_saps1, type = "response")
pred0_sofa <- predict(model0_sofa, type = "response")
pred0_fmodel <- predict(fmodel_df0_1, type = "response")
pred0_fmodel_TW <- predict(fmodel_df0_3, type = "response")

# Calculate Brier scores
Brier_model0_saps1 <- mean((pred0_saps1 - icu_patients_df0_cleaned$in_hospital_death)^2)
Brier_model0_sofa <- mean((pred0_sofa - icu_patients_df0_cleaned$in_hospital_death)^2)
Brier_fmodel_df0_1 <- mean((pred0_fmodel - icu_patients_df0_cleaned$in_hospital_death)^2)
Brier_fmodel_df0_3 <- mean((pred0_fmodel_TW - icu_patients_df0_cleaned$in_hospital_death)^2)

# Print Brier scores
cat("Brier score for SAPS1 Model: ", Brier_model0_saps1, "\n")
```

```
## Brier score for SAPS1 Model:  0.117486
```

```
cat("Brier score for SOFA Model: ", Brier_model0_sofa, "\n")
```

```
## Brier score for SOFA Model:  0.1168699
```

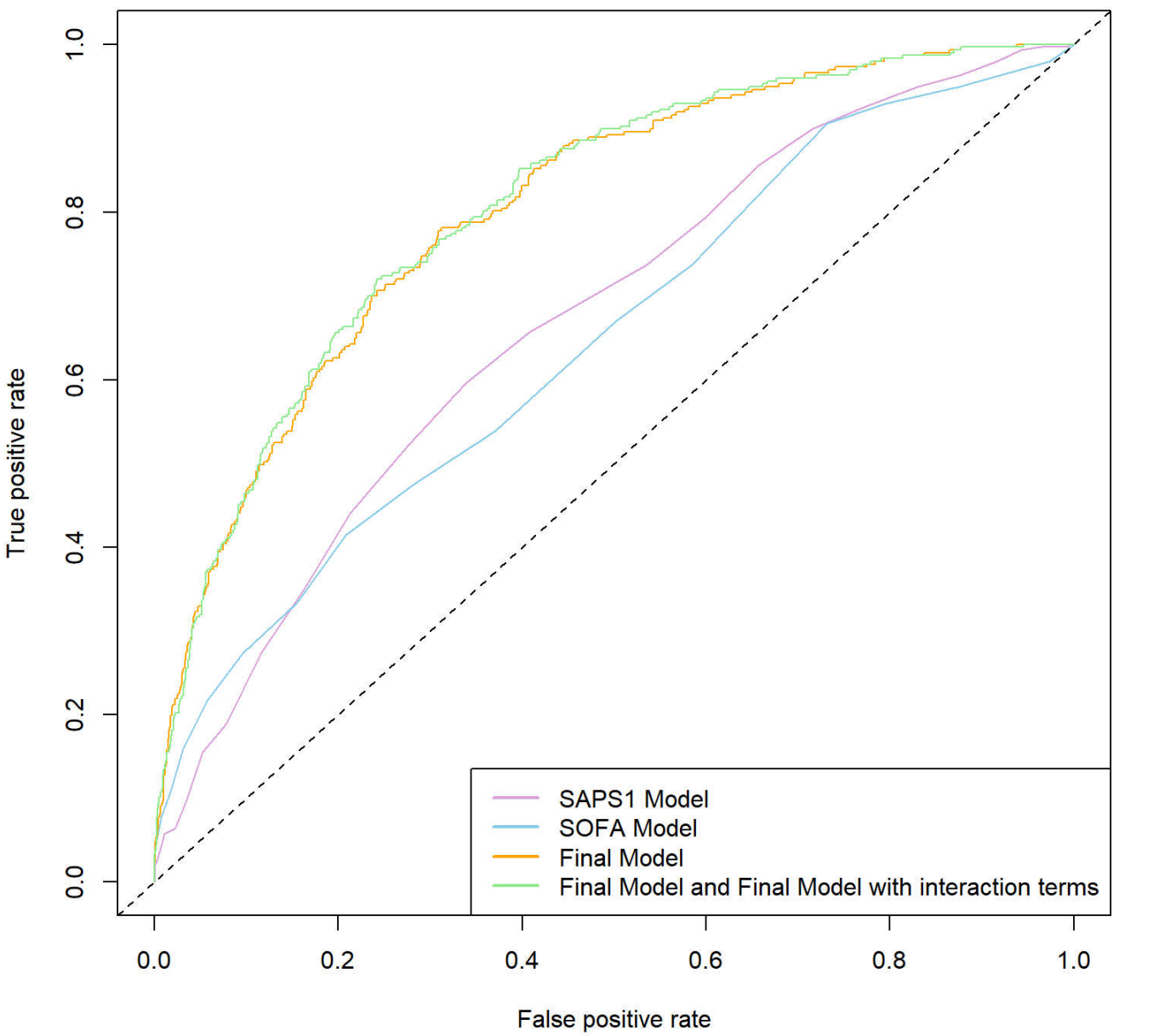
```
cat("Brier score for Final Model: ", Brier_fmodel_df0_1, "\n")
```

```
## Brier score for Final Model:  0.1008051
```

```
cat("Brier score for Final Model with interaction terms: ", Brier_fmodel_df0_3, "\n")
```

```
## Brier score for Final Model with interaction terms:  0.1003641
```

```
# Create prediction objects for ROC analysis
pred0_obj_saps1 <- prediction(pred0_saps1, icu_patients_df0_cleaned$in_hospital_death)
pred0_obj_sofa <- prediction(pred0_sofa, icu_patients_df0_cleaned$in_hospital_death)
pred0_obj_fmodel <- prediction(pred0_fmodel, icu_patients_df0_cleaned$in_hospital_death)
pred0_obj_fmodel_TW <- prediction(pred0_fmodel_TW, icu_patients_df0_cleaned$in_hospital_death)

# Create performance objects for ROC analysis
perf0_saps1 <- performance(pred0_obj_saps1, "tpr", "fpr")
perf0_sofa <- performance(pred0_obj_sofa, "tpr", "fpr")
perf0_fmodel <- performance(pred0_obj_fmodel, "tpr", "fpr")
perf0_fmodel_TW <- performance(pred0_obj_fmodel_TW, "tpr", "fpr")



# Plot ROC curves
plot(perf0_saps1, col = "plum", main = "ROC Curves for SAPS1, SOFA, Final Model and Final Model
with interaction terms")
plot(perf0_sofa, col = "skyblue", add = TRUE)
plot(perf0_fmodel, col = "orange", add = TRUE)
plot(perf0_fmodel_TW, col = "lightgreen", add = TRUE)
abline(a = 0, b = 1, lty = 2)

# Add a legend
legend("bottomright", legend = c("SAPS1 Model", "SOFA Model", "Final Model", "Final Model and F
inal Model with interaction terms"), col = c("plum", "skyblue", "orange", "lightgreen"), lwd =
2)
```




```
# Calculate AUC for SAPS1 model
auc0_saps1 <- performance(pred0_obj_saps1, measure = "auc")
auc0_saps1_value <- auc0_saps1@y.values[[1]]
cat("AUC for SAPS1 Model:", auc0_saps1_value, "\n")
```

```
## AUC for SAPS1 Model: 0.6693982
```

```
# Calculate AUC for SOFA model
auc0_sofa <- performance(pred0_obj_sofa, measure = "auc")
auc0_sofa_value <- auc0_sofa@y.values[[1]]
cat("AUC for SOFA Model:", auc0_sofa_value, "\n")
```

```
## AUC for SOFA Model: 0.6460789
```

```
# Calculate AUC for Final model
auc0_fmodel <- performance(pred0_obj_fmodel, measure = "auc")
auc0_fmodel_value <- auc0_fmodel@y.values[[1]]
cat("AUC for Final Model:", auc0_fmodel_value, "\n")
```

```
## AUC for Final Model: 0.8032456
```

```
# Calculate AUC for Final model with interaction terms
auc0_fmodel_TW <- performance(pred0_obj_fmodel_TW, measure = "auc")
auc0_fmodel_TW_value <- auc0_fmodel_TW@y.values[[1]]
cat("AUC for Final Model with interaction terms:", auc0_fmodel_TW_value, "\n")
```

```
## AUC for Final Model with interaction terms: 0.8074798
```

Interpretations: **Brier Score**: Lower Brier scores indicate better model calibration, where the predicted probabilities are closer to the actual outcomes. Both full models show better performance than the simpler SAPS1 and SOFA models, with the interaction-inclusive model (Full Model 3) slightly outperforming Full Model 1.

AUC: Higher AUC values indicate better discriminative ability, i.e., the model's ability to distinguish between patients who survived and those who did not. Similar to the Brier scores, the full models show superior performance, with the interaction-inclusive model demonstrating the highest AUC.

Conclusions of analysis and ROC Curves: - SAPS1 and SOFA models, while simpler and less computationally intensive, do not perform as well as the full models, which justifies the inclusion of additional predictors and interactions to improve model accuracy. - The improvement from "Final Model" to "Final Model with interaction terms" is marginal but consistent across both metrics, reinforcing the potential utility of these interaction terms in predicting patient outcomes. - "Final Model with interaction terms", which includes interactions, performs the best in terms of both calibration (Brier score) and discrimination (AUC), suggesting that including interaction terms between SOFA and pH_diff and SOFA and WBC_max may capture important complexities in the data that relate to the likelihood of in-hospital death.

Interpretation:

- The **inclusion of interaction terms** in final model provides a meaningful improvement, albeit slight, suggesting that the interactions between SOFA scores and both pH differences and WBC counts are relevant in predicting patient outcomes in the dataset.
- This improvement, while modest, could be clinically significant, especially in a setting where predicting patient outcomes can help tailor interventions more effectively.

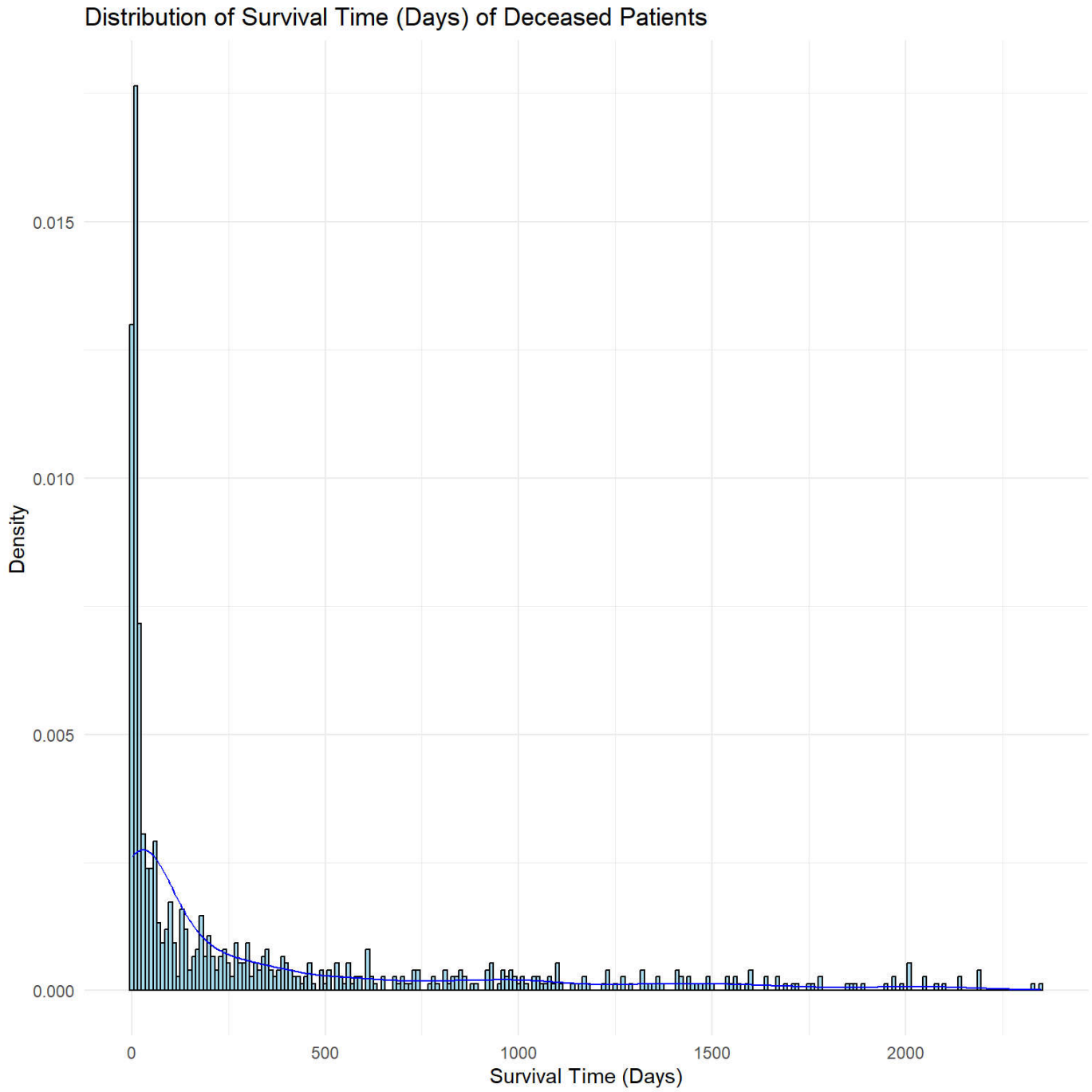
Task 2

Part A: Exploratory Data Analysis

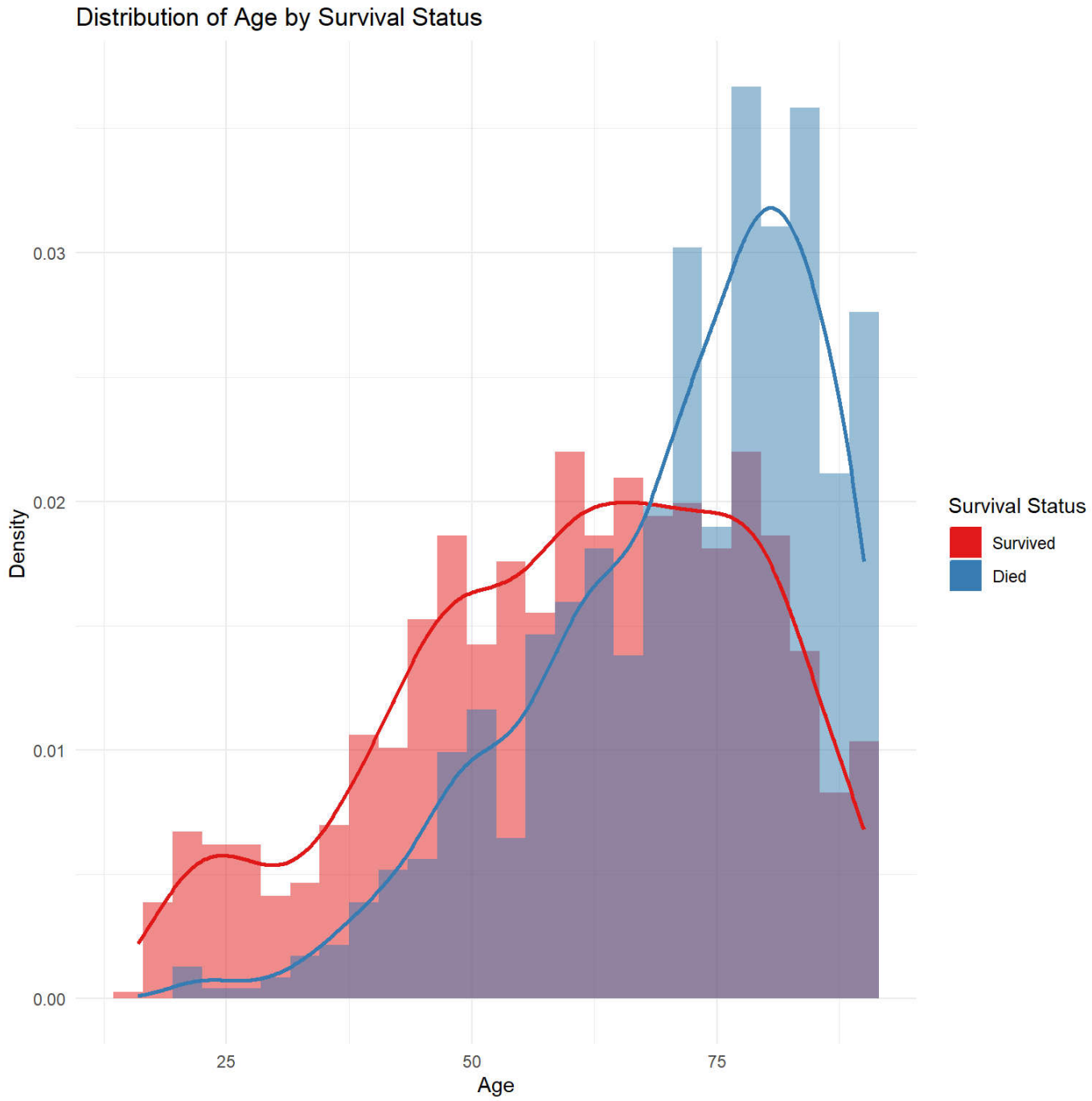
The goal of this EDA is similar to Task 1 and you may choose to use the same sub-set of predictors if they are relevant and appropriate to the analysis for Task 2. You may also choose to select a different sub-set if you wish. Justify whichever approach you choose.

Present enough information to: (i) explain why you selected these variables, and (ii) Provide an overview of the variables and how they are related to survival. Present your summary using tables and/or plots with supporting commentary where relevant.

```
# Convert days to years for the survival analysis
icu_patients_df1_cleaned$Years <- icu_patients_df1_cleaned$Days / 365.25
# Filter out the maximum 'Days' value for a clearer histogram via focus on deceased patients
# **In data instraction "ICU stays of less than 48 hours have been excluded.", But I still see the 1 and 2 days value in `Days` variable.
filtered_2days_data <- icu_patients_df1_cleaned %>% filter(Days >= 2)
filtered_days_data <- icu_patients_df1_cleaned %>% filter(Days >= 2 & Days < 2408)
#all(icu_patients_df1_cleaned$Days[icu_patients_df1_cleaned$Status == 'FALSE'] == 2408) #Return TRUE if all entries in icu_patients_df1 where the Status is FALSE have the Days value of 2408.

# Histogram of survival time with density line
ggplot(filtered_days_data, aes(x = Days)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 10, fill = "skyblue", color = "black", alpha = 0.7) +
  geom_density(color = "blue") +
  labs(title = "Distribution of Survival Time (Days) of Deceased Patients",
       x = "Survival Time (Days)",
       y = "Density") +
  theme_minimal()
```




```
# Histogram of patient ages with density line
# Transform the 'Status' logical variable into a factor with levels 'Survived' and 'Died'
icu_patients_df1_cleaned$Survival_Status <- factor(icu_patients_df1_cleaned$Status, levels = c
(FALSE, TRUE), labels = c("Survived", "Died"))

ggplot(icu_patients_df1_cleaned, aes(x = Age, y = after_stat(density))) +
  geom_histogram(aes(fill = Survival_Status), position = "identity", binwidth = 3, alpha = 0.5,
na.rm = TRUE) +
  scale_fill_brewer(palette = "Set1", name = "Survival Status") +
  geom_density(aes(color = Survival_Status), linewidth = 1, na.rm = TRUE) +
  labs(title = "Distribution of Age by Survival Status",
       x = "Age",
       y = "Density") +
  theme_minimal() +
  scale_color_brewer(palette = "Set1", name = "Survival Status")
```



Distribution of Survival Time (Days) of Deceased Patients: - A heavy concentration of data at the lower end of the survival time spectrum, indicating that a significant number of patients unfortunately passed away shortly after their ICU admission. - If including values at 2408 days, the presence of a sharp peak at the far right end could indicate right-censoring at a study endpoint or a standard follow-up period.

Distribution of All Patient Age: - The distribution of ages among patients who survived (represented by the red bars and density curve) spans a broad range, and there's a notable peak around the middle age range. - The distribution of ages among patients who died (depicted by the blue bars and density curve) also covers a wide age range but with a peak at an older age, indicating older patients possibly having a higher risk of mortality.

```
# Survival status frequency and proportion table
status_table <- table(icu_patients_df1_cleaned$Status)
status_prop <- prop.table(status_table) * 100  # Convert to percentages
status_df <- data.frame(
  Status = c("Survived", "Deceased"),
  Frequency = as.integer(status_table),
  Proportion = sprintf("%.2f%%", status_prop)
)
kable(status_df, caption = "Survival Status of Patients", row.names = FALSE)
```

Survival Status of Patients

Status	Frequency	Proportion
Survived	1288	62.49%
Deceased	773	37.51%

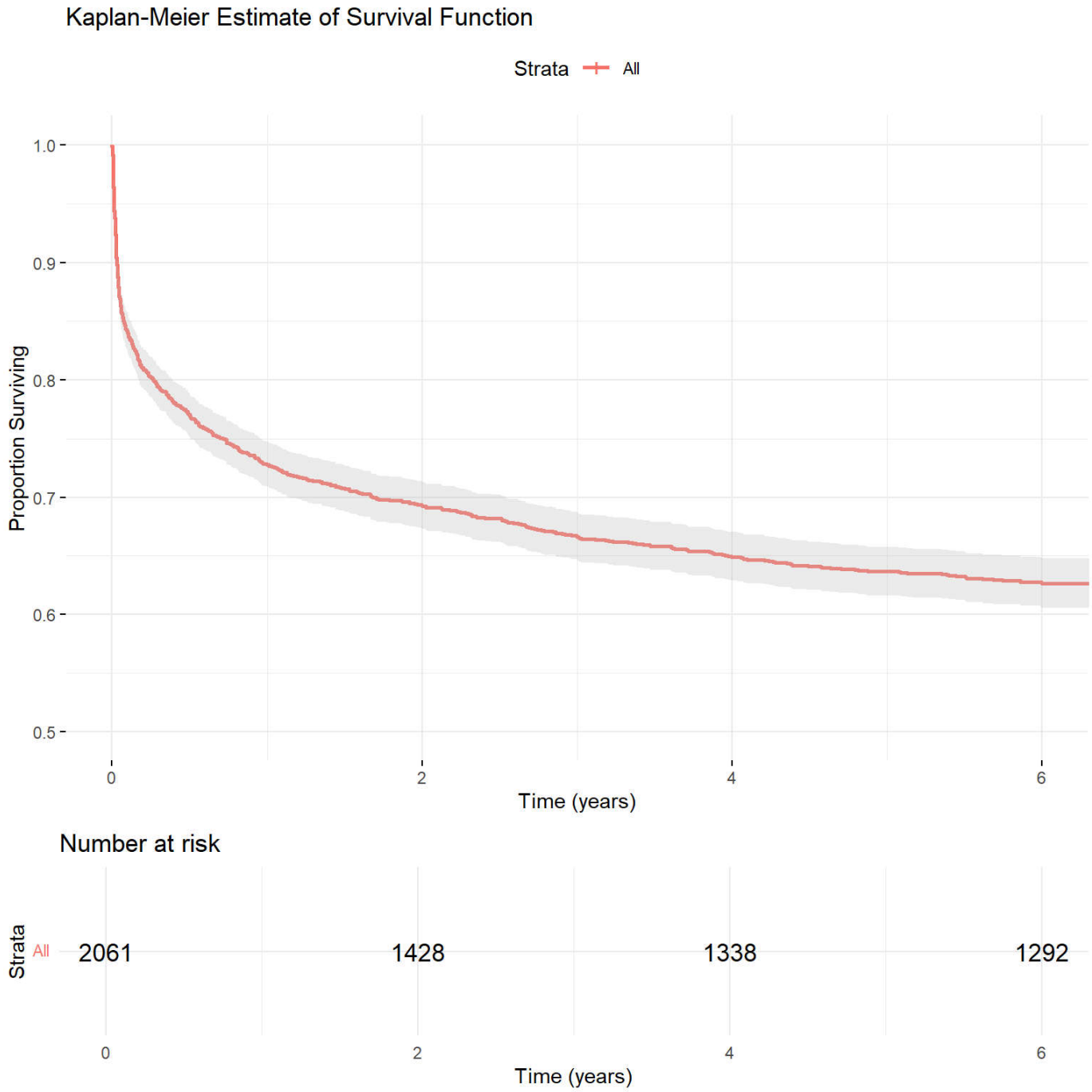
```
# Summaries for continuous variables (Days)
days_summary <- summary(filtered_days_data$Days)
days_summary_df <- data.frame(
  Statistic = c("Minimum", "1st Quartile", "Median", "Mean", "3rd Quartile", "Maximum"),
  Value = c(days_summary[c(1,2,3,4,5,6)])
)
kable(days_summary_df, caption = "Summary of Survival Time (Days) of Deceased Patients")
```

Summary of Survival Time (Days) of Deceased Patients

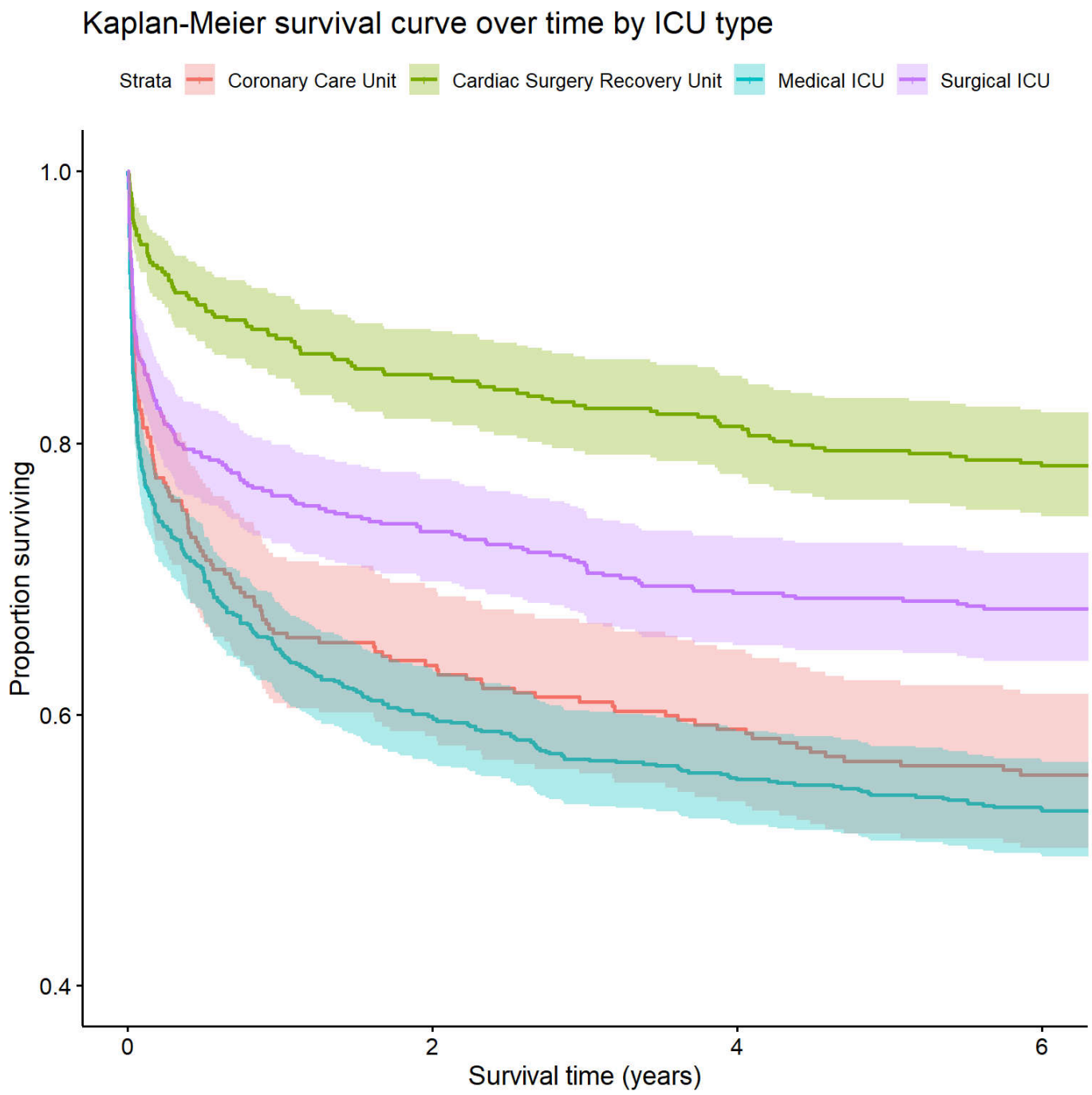
	Statistic	Value
Min.	Minimum	2.0000
1st Qu.	1st Quartile	11.0000
Median	Median	77.0000
Mean	Mean	348.5013
3rd Qu.	3rd Quartile	445.5000
Max.	Maximum	2350.0000

```
# Creating the survival object
surv_obj <- Surv(time = icu_patients_df1_cleaned$Days, event = icu_patients_df1_cleaned$Status)
surv_obj_years <- Surv(time = icu_patients_df1_cleaned$Years, event = icu_patients_df1_cleaned
$Status)

# Fitting the Kaplan-Meier survival model
surv_fit <- survfit(surv_obj ~ 1)  # `~ 1` for an overall curve
surv_fit_years <- survfit(surv_obj_years ~ 1, data = icu_patients_df1_cleaned)
surv_fit_ICUtype <- survfit(surv_obj_years ~ ICUType, data = icu_patients_df1_cleaned)

# Plot the overall Kaplan-Meier survival curve
ggsurvplot(surv_fit_years, data = icu_patients_df1_cleaned,
  conf.int = TRUE,  risk.table = TRUE, # Display the confidence interval and risk table
  xlab = "Time (years)", ylab = "Proportion Surviving", title = "Kaplan-Meier Estimate of Survi
val Function", ylim = c(0.5, 1),
  surv.median.line = "hv",  # Add a horizontal line at the median
  ggtheme = theme_minimal()  # Use a minimal theme for a cleaner look
)
```

```
## Warning in .add_surv_median(p, fit, type = surv.median.line, fun = fun, :
## Median survival not reached.
```




```
#Kaplan Meier curve by ICU types
ggsurvplot(surv_fit_ICUtype, data=icu_patients_df1_cleaned,  conf.int = TRUE,  censor.size=2, risk.table = FALSE, legend.labs = c("Coronary Care Unit", "Cardiac Surgery Recovery Unit", "Medical ICU", "Surgical ICU"), ylim = c(0.4,1.0),) +
   xlab("Survival time (years)") + ylab("Proportion surviving")+ ggtitle("Kaplan-Meier survival curve over time by ICU type")
```



Part B: Explanatory survival model

Your aim is to try to understand and EXPLAIN how survival time varies by `Age`. How does survival change with increasing Age? Can any age-related differences in survival time be explained by differing clinical presentations of older patients?

In this task, you are required to develop a Cox proportional hazards survival model using the `icu_patients_df1` data set which adequately **explains** the length of survival indicated by the `Days` variable, with censoring as indicated by the `Status` variable. You should fit a univariable model examining the relationship between Age and survival. You should then also fit another one or two multivariable models, to examine the extent that survival is explained by other variables in the dataset and whether age is independently related to survival after controlling for other factors. Your final model should **not** include all the predictor variables, just a relevant subset of them, which you have selected based on statistical significance and/or background knowledge. It is perfectly acceptable to include predictor variables in your final model which are not statistically significant, as long as you justify their inclusion on medical or physiological grounds (you will not be marked down if your medical justification is not exactly correct, but do you best). You should assess

each model you consider for goodness of fit and other relevant statistics, and you should assess your final model for violations of assumptions and perform other diagnostics which you think are relevant (and modify the model if indicated, or at least comment on the possible impact of what your diagnostics show).

Finally, re-fit your final model to the unimputed data frame (`icu_patients_df0.rds`) and comment on any differences you find.

Create your response to task 2 here, as a mixture of embedded (`knitr`) R code and any resulting outputs, and explanatory or commentary text. Add code chunks as you see fit and choose whether you wish for the code and or results to be displayed in the final html document.

```
cox_model_age <- coxph(Surv(Days, Status) ~ Age, data = icu_patients_df1_cleaned)
summary(cox_model_age)
```

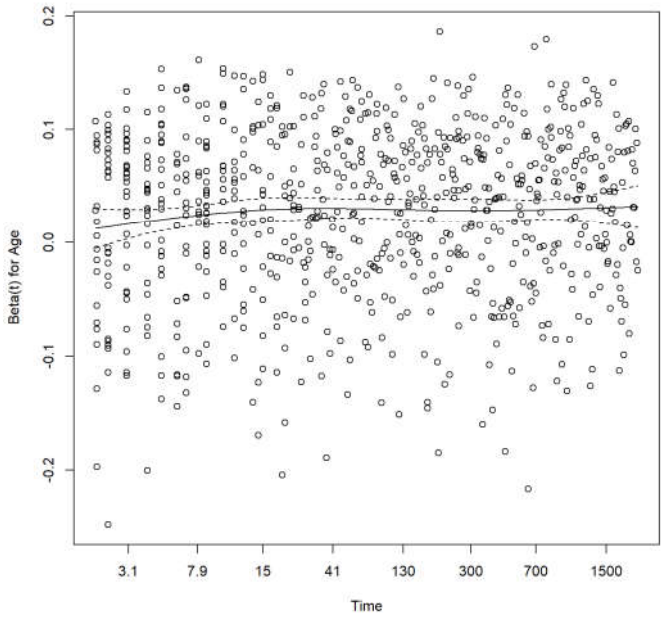
```
## Call:
## coxph(formula = Surv(Days, Status) ~ Age, data = icu_patients_df1_cleaned)
##
##   n= 2061, number of events= 773
##
##        coef exp(coef) se(coef)     z Pr(>|z|)
## Age 0.03355   1.03412  0.00250 13.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##     exp(coef) exp(-coef) lower .95 upper .95
## Age     1.034      0.967     1.029     1.039
##
## Concordance= 0.646  (se = 0.01 )
## Likelihood ratio test= 209.4  on 1 df,   p=<2e-16
## Wald test            = 180.1  on 1 df,   p=<2e-16
## Score (logrank) test = 187  on 1 df,   p=<2e-16
```

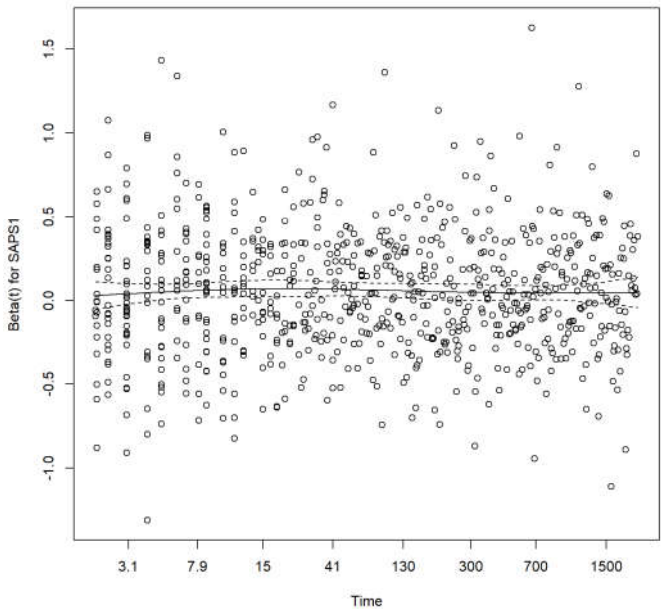
```
cox_model_age2 <- coxph(Surv(Days, Status) ~ Age, data = filtered_2days_data) # not sure if we should exclude < 2 days data as "ICU stays of less than 48 hours have been excluded."
summary(cox_model_age2)
```

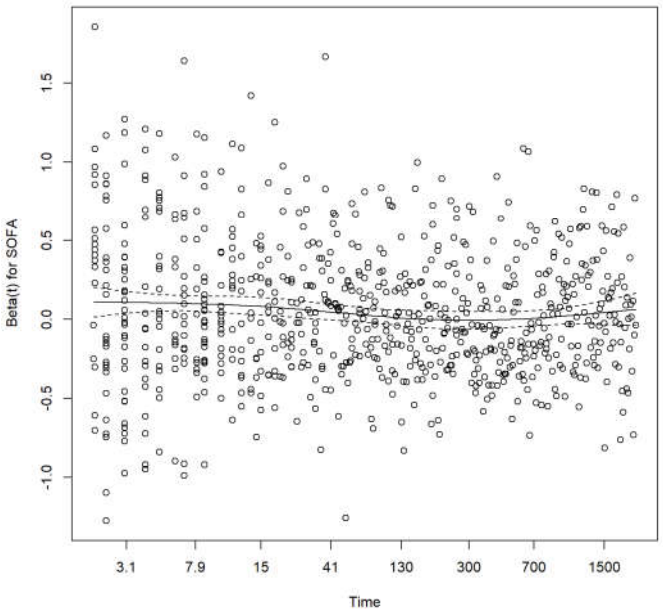
```
## Call:
## coxph(formula = Surv(Days, Status) ~ Age, data = filtered_2days_data)
##
##   n= 2043, number of events= 755
##
##         coef exp(coef) se(coef)     z Pr(>|z|)
## Age 0.034403  1.035002 0.002543 13.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##     exp(coef) exp(-coef) lower .95 upper .95
## Age     1.035     0.9662      1.03      1.04
##
## Concordance= 0.649  (se = 0.01 )
## Likelihood ratio test= 213.7  on 1 df,   p=<2e-16
## Wald test            = 183  on 1 df,   p=<2e-16
## Score (logrank) test = 190.4  on 1 df,   p=<2e-16
```

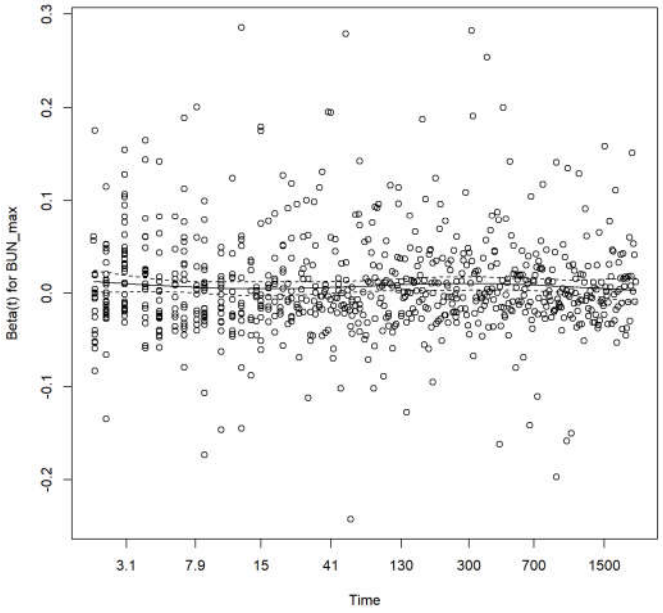
Coefficient (Age): The coefficients in model are positive indicating that with additional year of age, risk of the event (death) increased. **Hazard Ratio (exp(coef)):** Each additional year of age increases the risk of death by approximately 3.5% **Concordance:** It measures the model's predictive accuracy 0.65 suggest a moderate predictive ability **Statistical Tests:** All three statistical tests (Likelihood ratio test, Wald test, and Score (logrank) test) confirm the significant relationship between age and survival in both models.

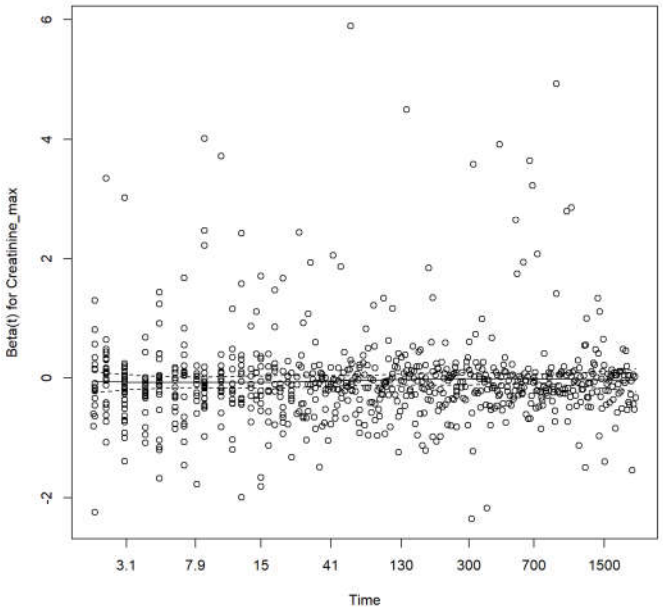
task2.B_2 Multivariable Cox Models Creating Checking and Selection

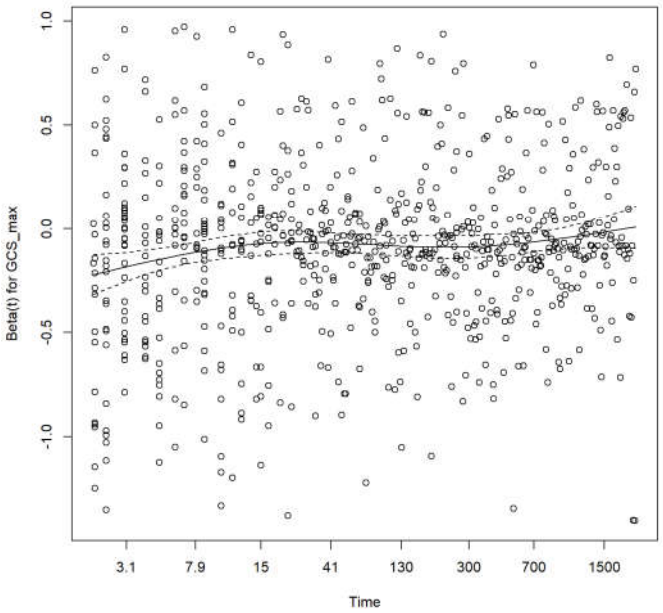


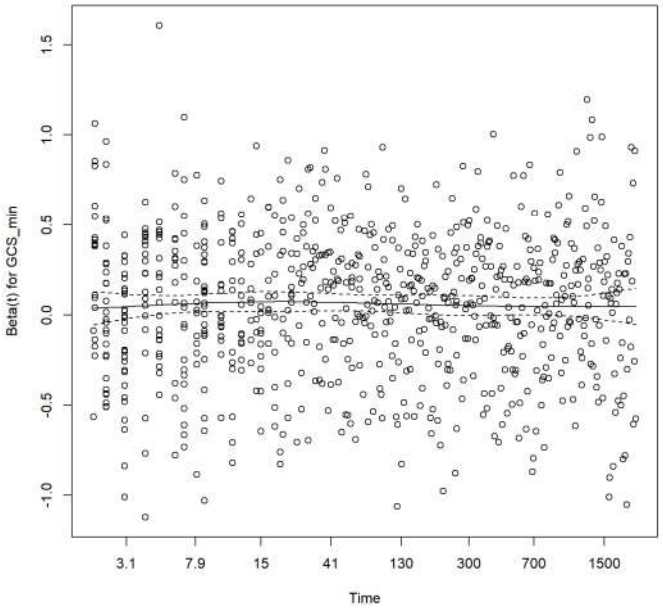


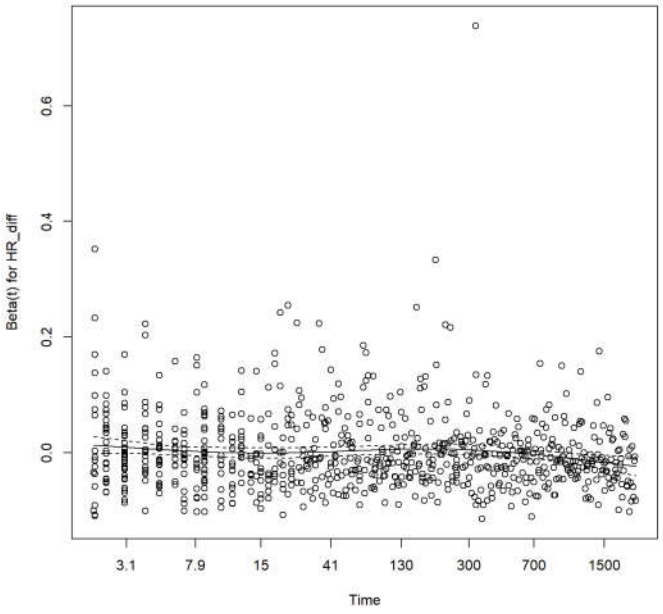

Beta(t) for SOFA
1.5
1.0
0.5
0.0
-0.5
-1.0
3.1
7.9
15
41
130
300
700
1500
Time

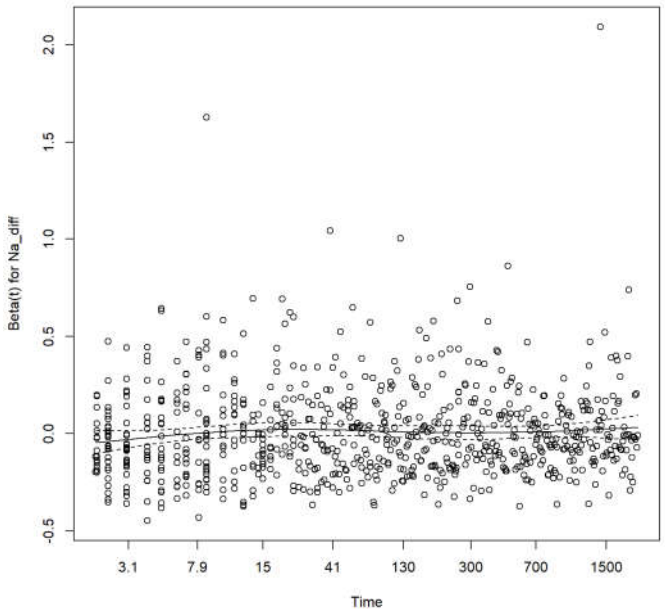

Beta(t) for BUN_max
0.3
0.2
0.1
0.0
-0.1
-0.2
3.1
7.9
15
41
130
300
700
1500
Time

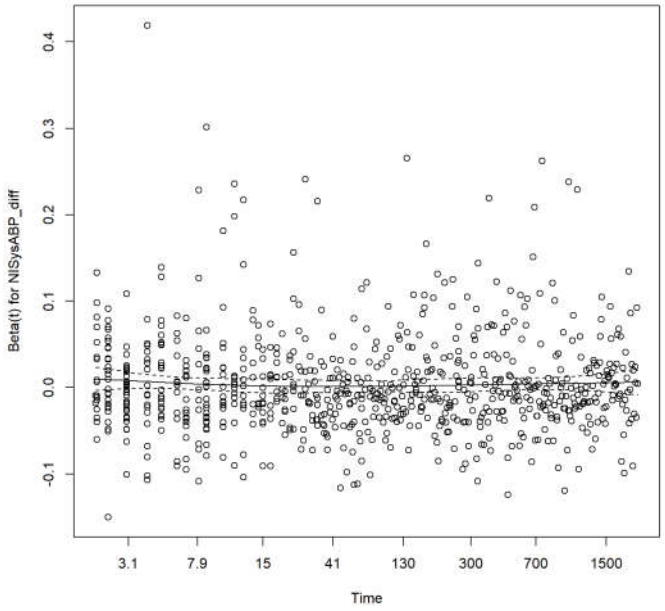

Beta(t) for Creatinine_max
6
4
2
0
-2
3.1
7.9
15
41
130
300
700
1500
Time

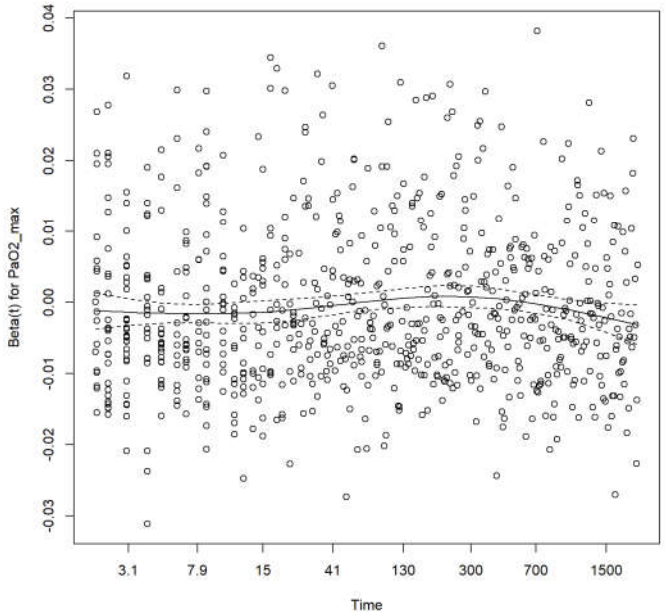

Beta(t) for GCS_max
1.0
0.5
0.0
-0.5
-1.0
3.1
7.9
15
41
130
300
700
1500
Time

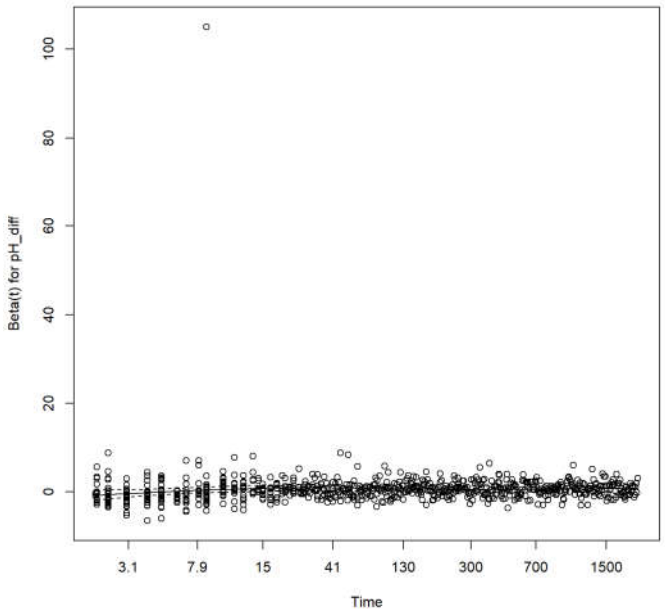

Beta(t) for GCS_min
1.5
1.0
0.5
0.0
-0.5
-1.0
3.1
7.9
15
41
130
300
700
1500
Time

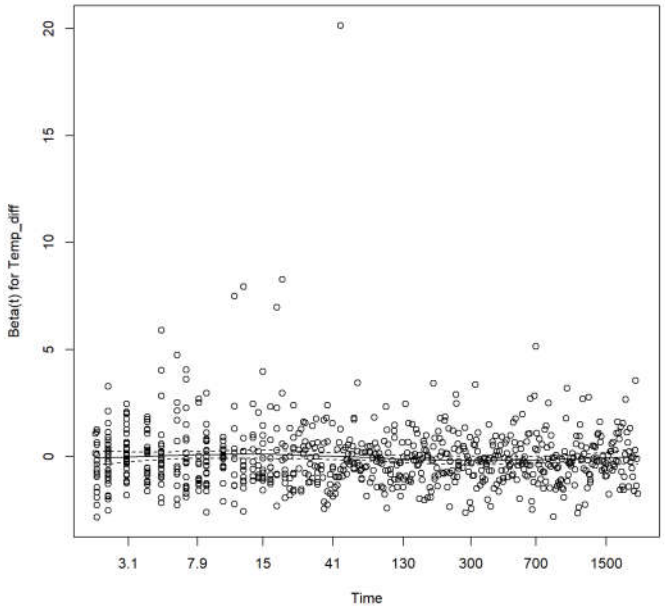

Beta(t) for HR_diff
0.6
0.4
0.2
0.0
3.1
7.9
15
41
130
300
700
1500
Time

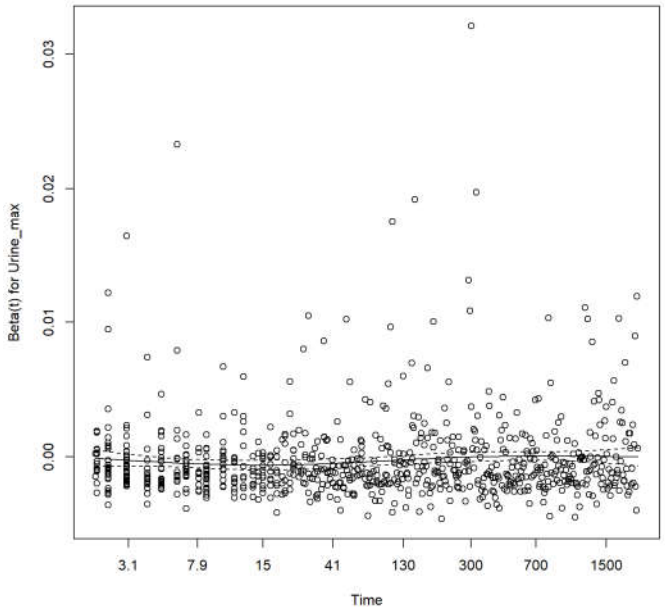

Beta(t) for Na_diff
2.0
1.5
1.0
0.5
0.0
-0.5
3.1
7.9
15
41
130
300
700
1500
Time

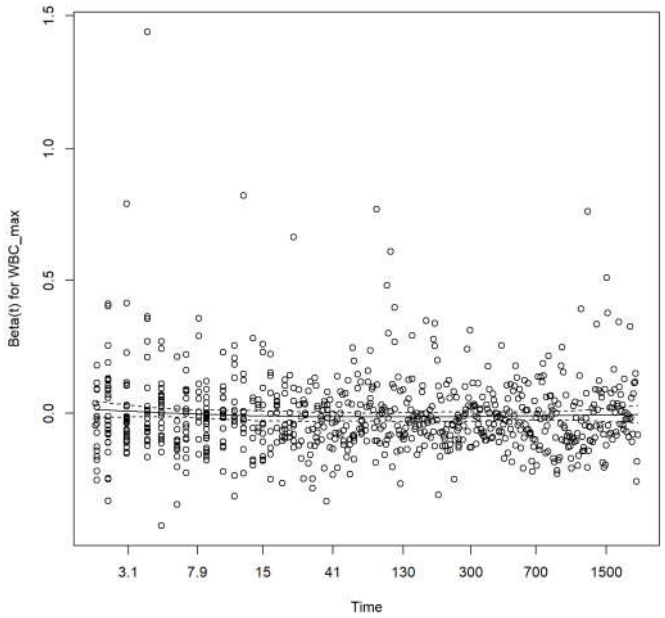

Beta(t) for NISysABP_diff
0.4
0.3
0.2
0.1
0.0
-0.1
3.1
7.9
15
41
130
300
700
1500
Time

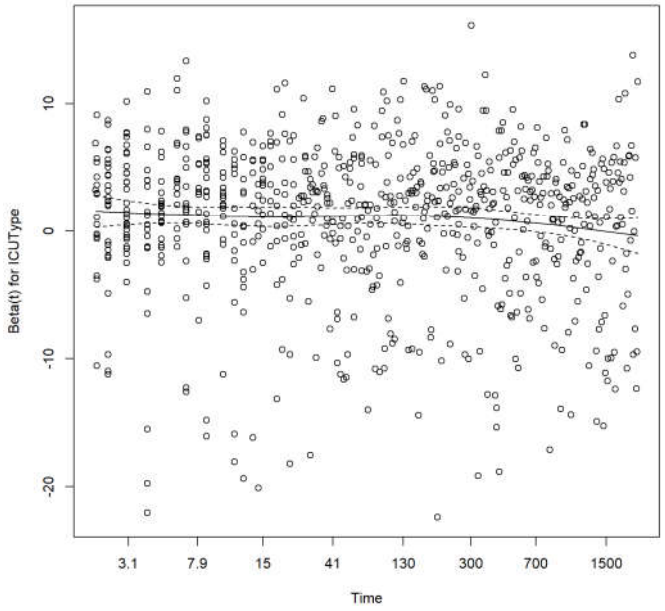

Beta(t) for PaO2_max
0.04
0.03
0.02
0.01
0.00
-0.01
-0.02
-0.03
3.1
7.9
15
41
130
300
700
1500
Time


Beta(t) for pH_diff
100
80
60
40
20
0
3.1
7.9
15
41
130
300
700
1500
Time


Beta(t) for Temp_diff
20
15
10
5
0
3.1
7.9
15
41
130
300
700
1500
Time


Beta(t) for Urine_max
0.03
0.02
0.01
0.00
3.1
7.9
15
41
130
300
700
1500
Time





The `cox.zph` function in R tests the proportional hazards assumption of a Cox regression model. 1. **Plot Interpretation**: - The x-axis represents the transformed time, and the y-axis shows the scaled Schoenfeld residuals. - A horizontal line at zero (sometimes shown with confidence bands) is the reference line indicating no deviation from proportionality. - If the points form a random pattern around the horizontal line, it suggests that the proportional hazards assumption holds for that variable. - Systematic patterns, trends, or a non-random dispersion suggest a violation of the assumption. 2. **General Observation**: - Age, GCS max, SOFAF, HR_diff and ICUType plots show some pattern or deviation from the zero line, particularly for Age and ICUType, which suggests potential violations of the proportional hazards assumption, indicating the effect of these variables on the hazard changes over time. 3. **The proportional hazards assumption test**: performed using the scaled Schoenfeld residuals, which should be approximately independent of time if the proportional hazards assumption holds. - chisq: This column shows the chi-square statistic for the test of the null hypothesis that there is no time dependence of the covariate's effect. A higher value indicates more evidence against the null hypothesis. - df: This column indicates the degrees of freedom associated with the chi-square test for each covariate. Most individual tests will have 1 degree of freedom unless a covariate is categorical with more than two levels (like ICUType). - p: This column provides the p-value associated with the chi-square test. A small p-value (typically less than 0.05) suggests that the effect of the covariate is not proportional over time, indicating a violation of the proportional hazards assumption. For covariates with significant tests (low p-values), consider

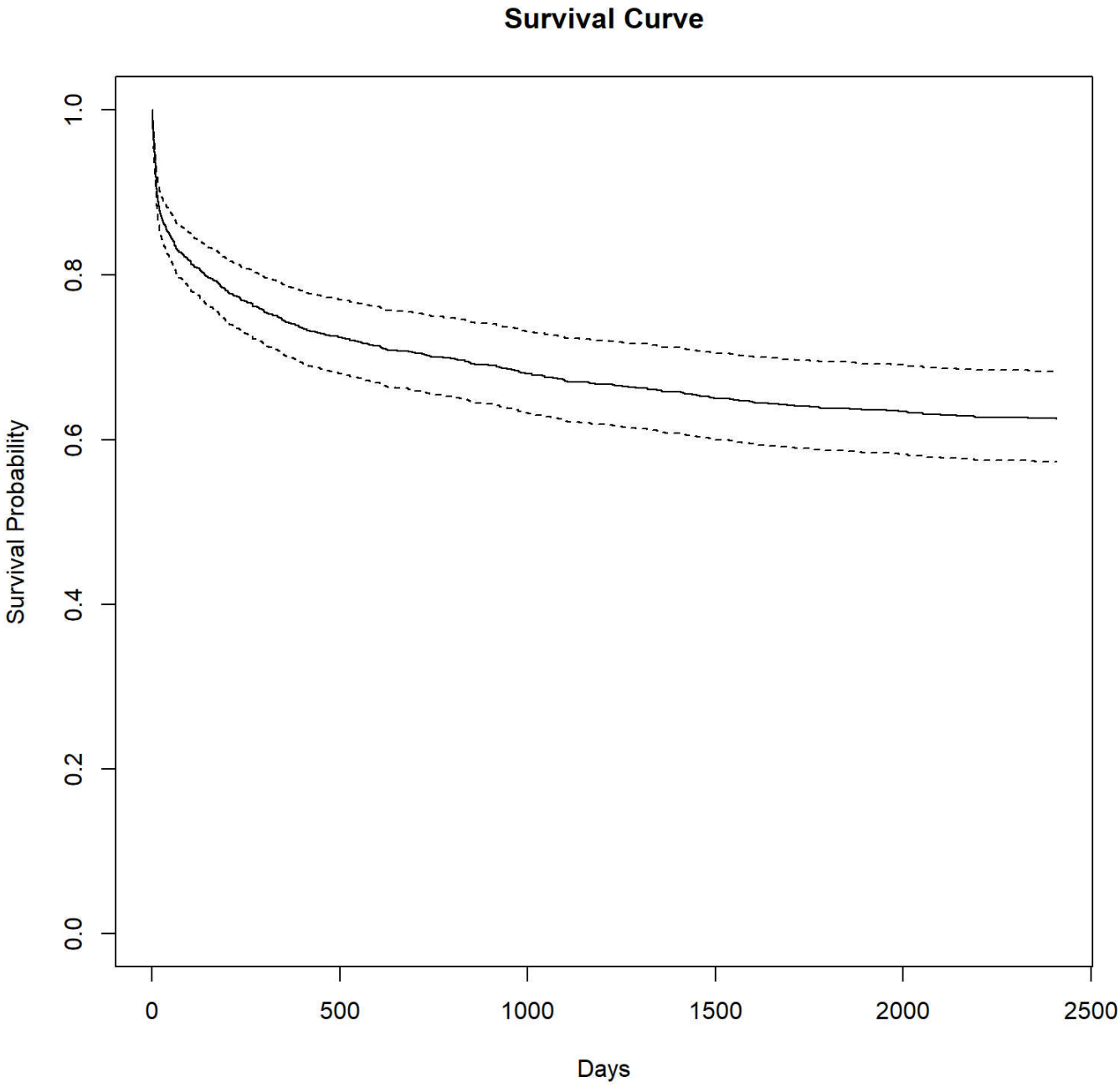
modeling them with time-varying coefficients or including interaction terms with time to account for their changing effects. 4. **Stepwise Cox model summary interpretation**: - coef (Coefficient): This represents the estimated effect of each covariate on the hazard rate, assuming all other covariates are held constant. A positive coefficient increases the hazard rate, a negative coefficient reduces the hazard rate. - exp(coef) (Hazard Ratio): The factor by which the hazard rate is multiplied for a one-unit increase in the covariate.Above 1 indicates increased risk; below 1 indicates decreased risk. - se(coef) (Standard Error): The standard deviation of the estimated coefficient, indicating the precision of the estimate. Smaller values suggest more precise estimates. - Z-score: The ratio of the coefficient to its standard error. It's used to test the null hypothesis that the coefficient is zero (no effect). - Pr(>|z|) (P-value): This indicates the probability of observing the given result, if the null hypothesis (that the coefficient is zero) is true. A small p-value less than 0.05 suggests that the effect is statistically significant. 5.**Stepwise Cox model summary**: - Creatinine_max, NISysABP_diff, PaO2_max, pH_diff, Urine_max, WBC_max**: These variables show varying levels of significance. - Notably, pH_diff is significant and shows a substantial increase in hazard with increasing pH variability. - Different ICU types show different risks compared to the baseline category (not shown here). The Cardiac Surgery Recovery Unit has a significantly lower hazard ratio, suggesting a protective effect compared to the baseline.

```
# Assessing proportional hazards assumption
test_proportional_hazards <- cox.zph(stepwise_cox_model)
print(test_proportional_hazards)
```

```
##                   chisq df       p
## Age              2.2060  1  0.1375
## SAPS1           16.3363  1 5.3e-05
## SOFA            25.9931  1 3.4e-07
## BUN_max          2.1442  1  0.1431
## Creatinine_max   0.7937  1  0.3730
## GCS_max         30.2955  1 3.7e-08
## GCS_min         15.6664  1 7.6e-05
## NISysABP_diff    0.5275  1  0.4677
## PaO2_max         0.0031  1  0.9556
## pH_diff          0.0588  1  0.8084
## Urine_max        6.0373  1  0.0140
## WBC_max          4.4705  1  0.0345
## ICUType         11.8779  3  0.0078
## GLOBAL          63.4238 15 6.4e-08
```

After selection process (dropping non-significant and/or multicollinear predictors through stepwise selection by `stepAIC()`), `HR_diff`, `Na_diff`, and `Temp_diff` are were removed.

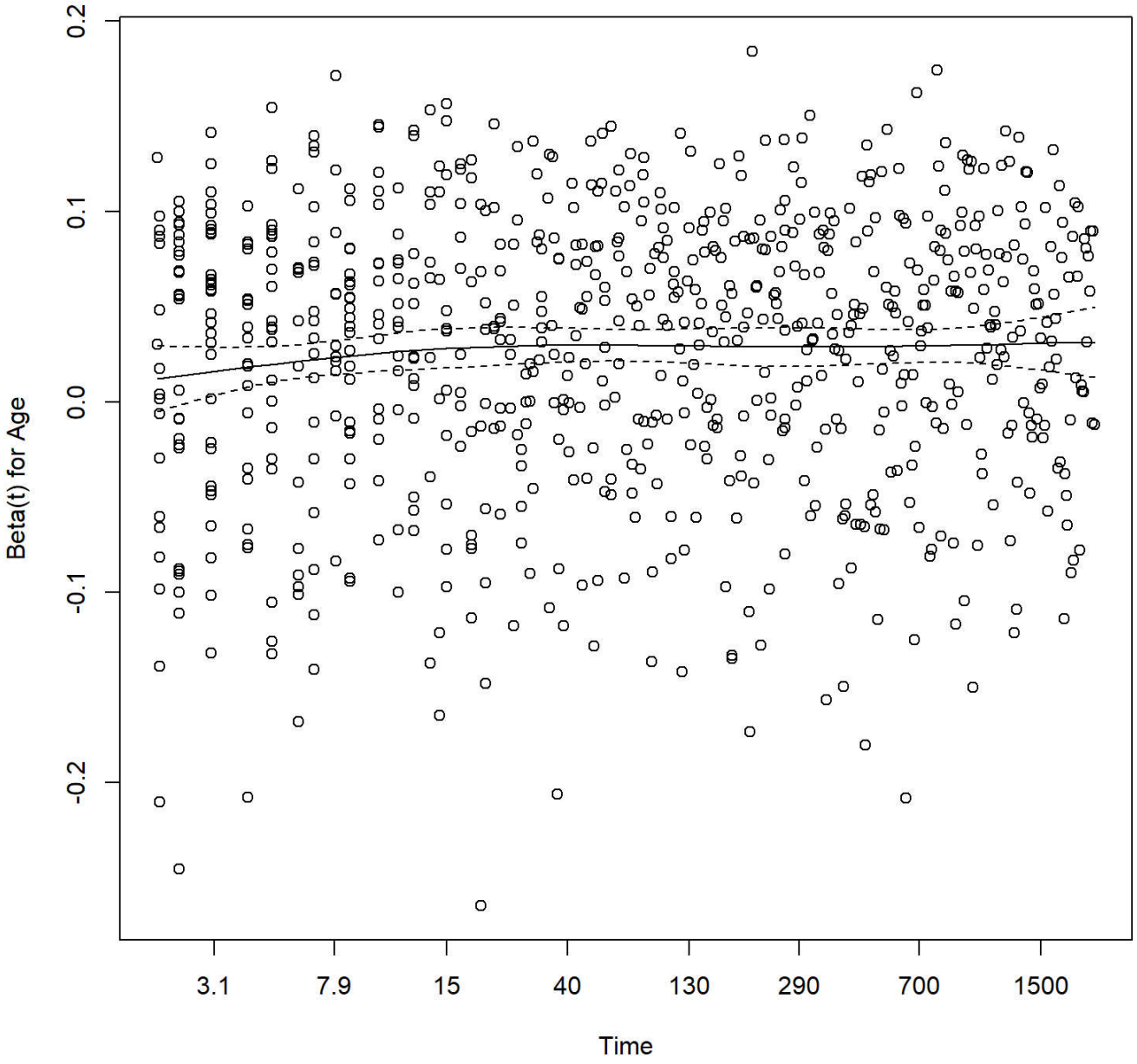
```
# Fit Cox model to df0 data
cox_model <- coxph(Surv(Days, Status) ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max + GCS_max
+ GCS_min + NISysABP_diff + PaO2_max + pH_diff + Urine_max + WBC_max + ICUType,
                           data = icu_patients_df0_cleaned)

plot(survfit(cox_model), xlab = "Days", ylab = "Survival Probability", main = "Survival Curve")
```

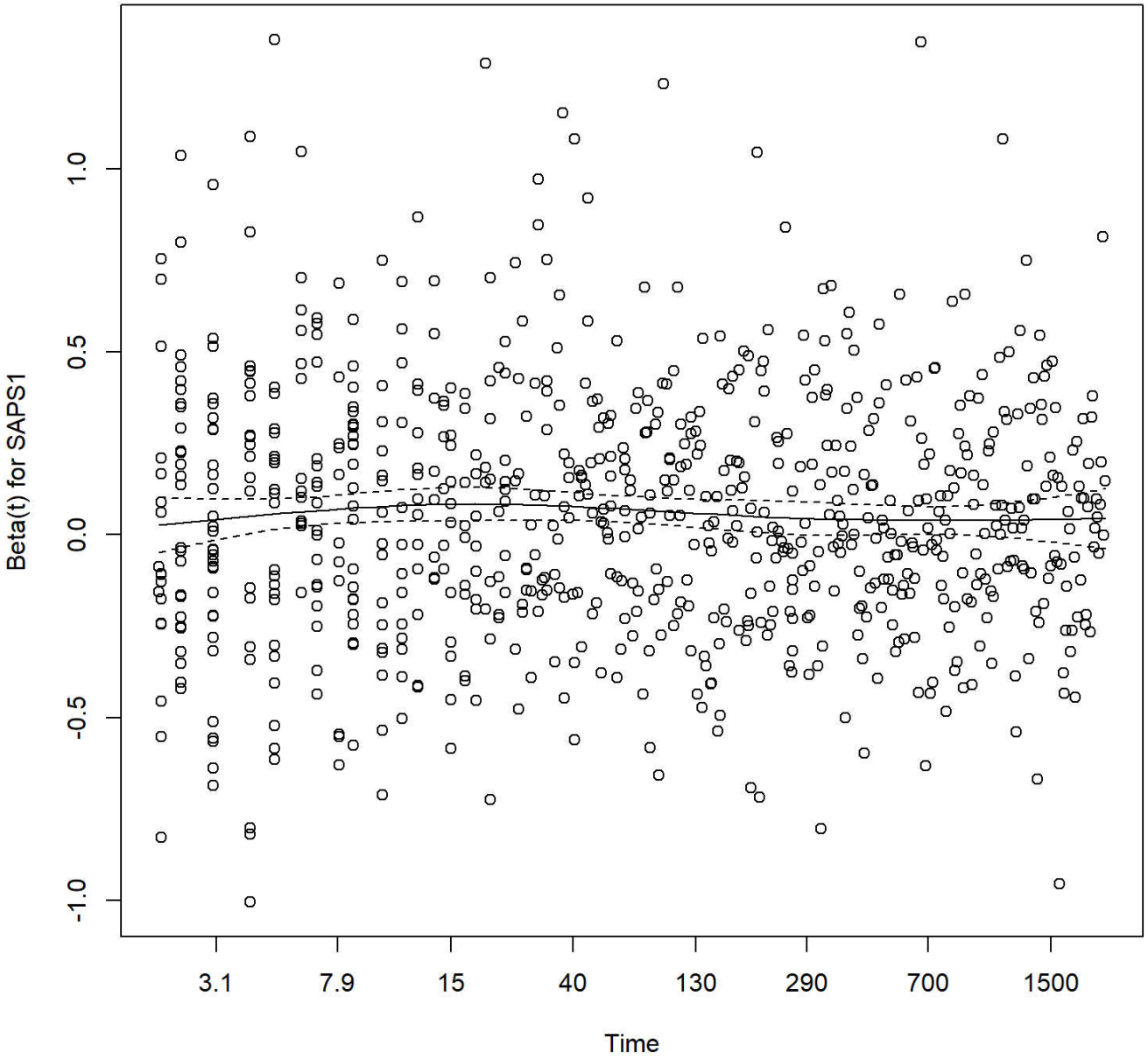


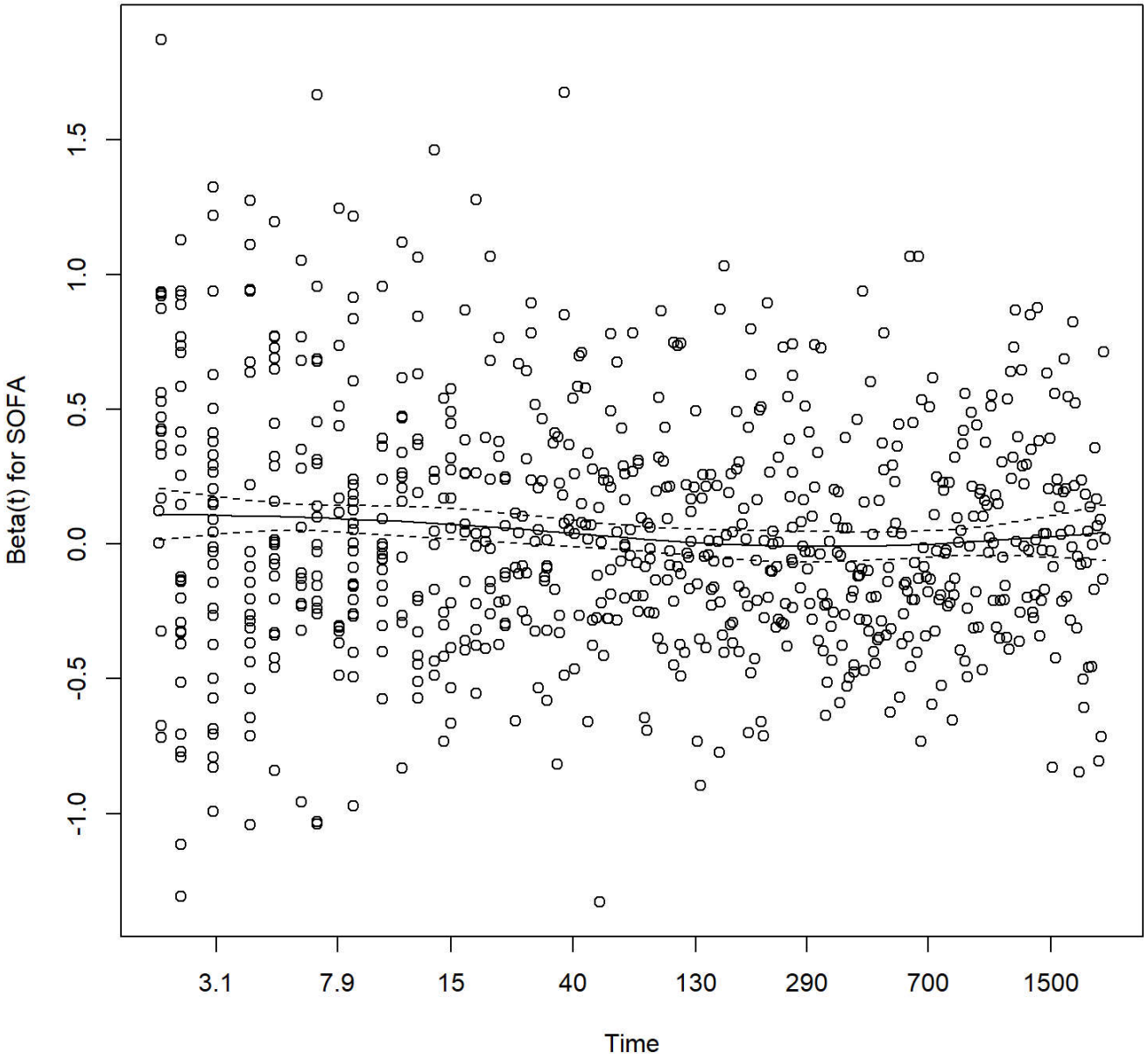

```
# Check proportional hazards assumption
cox.zph_res_df0 <- cox.zph(cox_model)
print(cox.zph_res_df0)
```

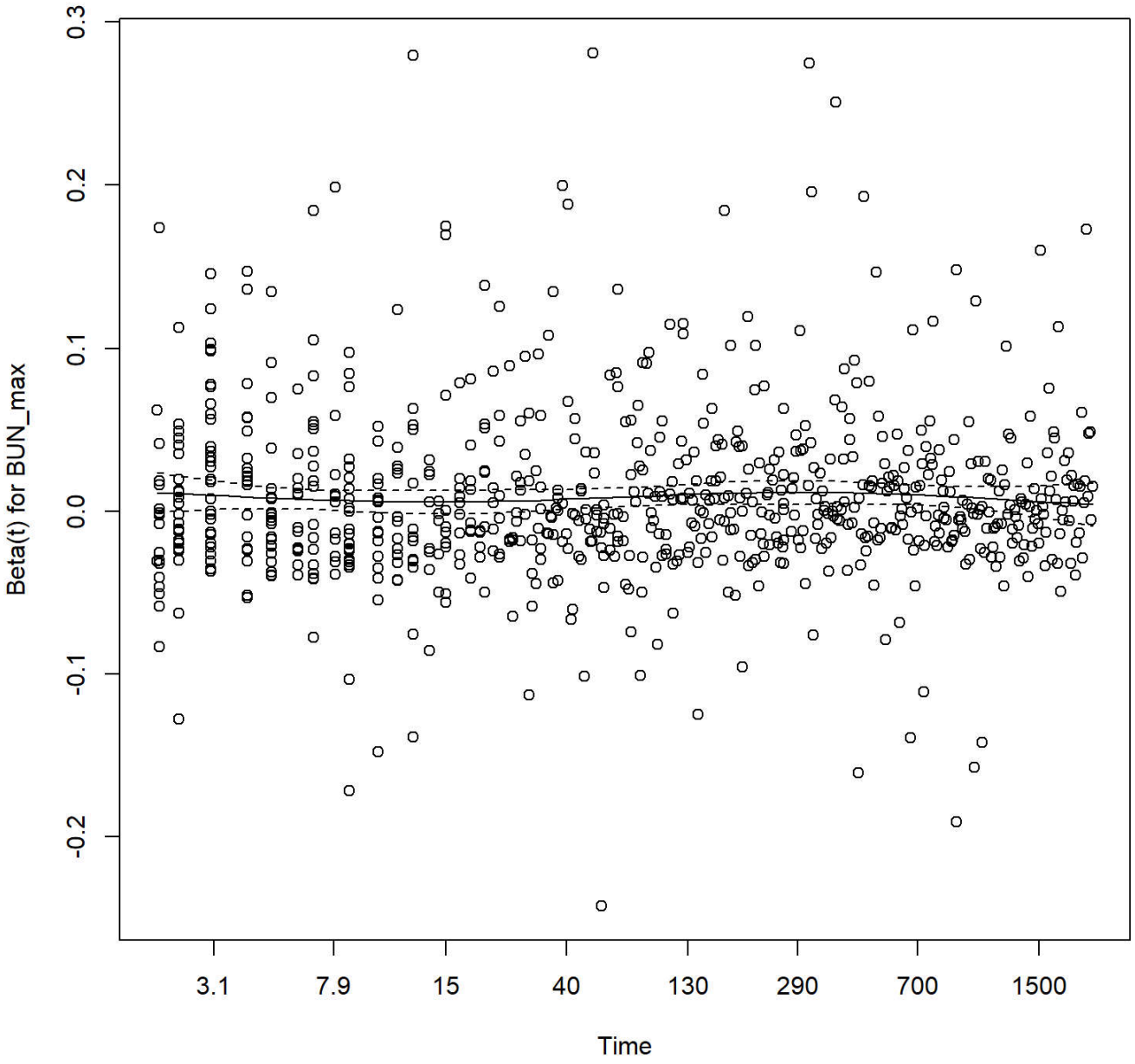
```
##                   chisq df       p
## Age            2.30e+00  1   0.129
## SAPS1          1.57e+01  1 7.5e-05
## SOFA           2.69e+01  1 2.1e-07
## BUN_max        1.82e+00  1   0.178
## Creatinine_max 6.26e-01  1   0.429
## GCS_max        2.55e+01  1 4.5e-07
## GCS_min        1.54e+01  1 8.8e-05
## NISysABP_diff  5.22e-01  1   0.470
## PaO2_max       9.33e-08  1   1.000
## pH_diff        6.37e-02  1   0.801
## Urine_max      5.26e+00  1   0.022
## WBC_max        5.05e+00  1   0.025
## ICUType        1.10e+01  3   0.012
## GLOBAL         5.83e+01 15 4.9e-07
```

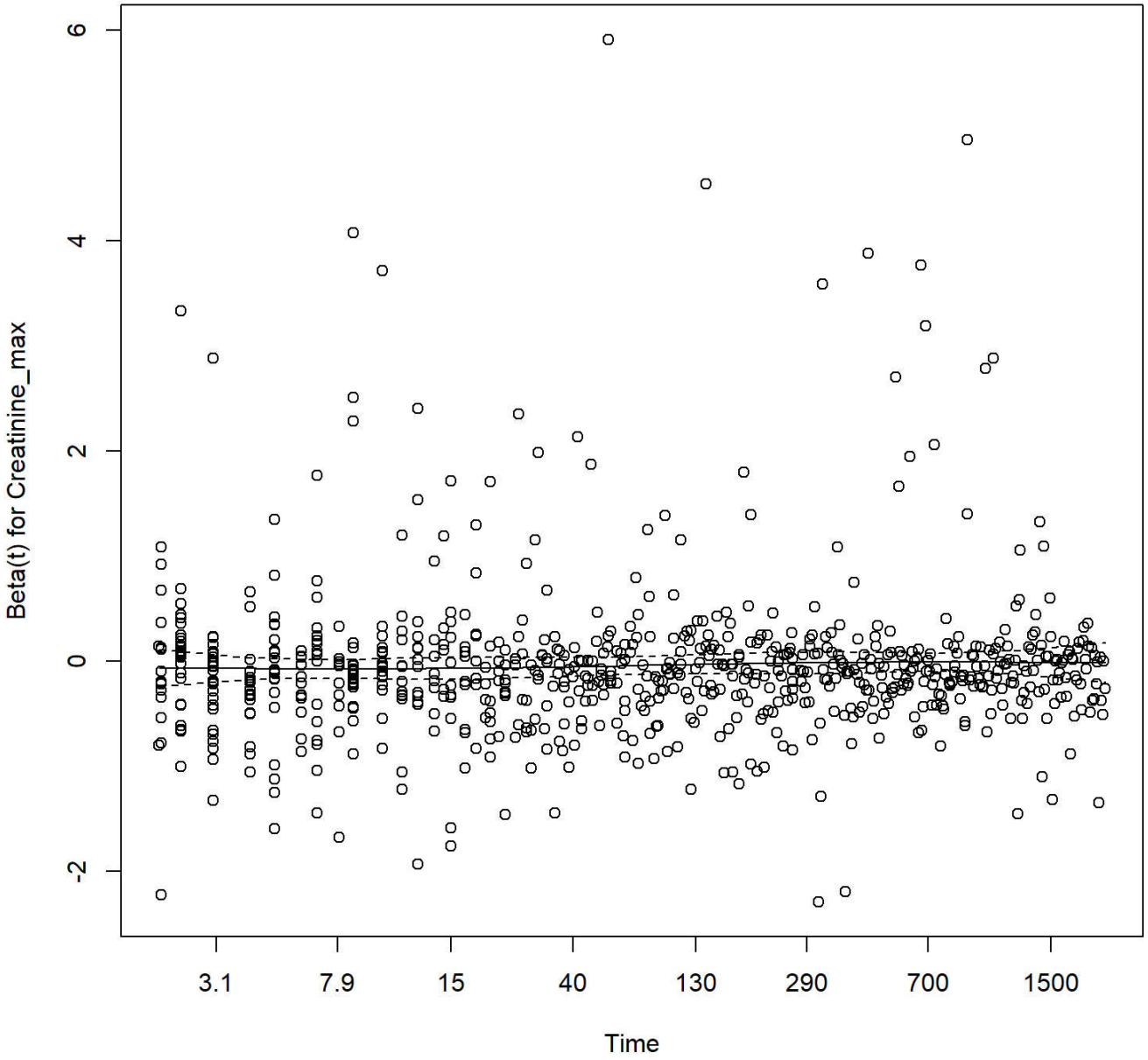
```
plot(cox.zph_res_df0) #diagnostic plots
```

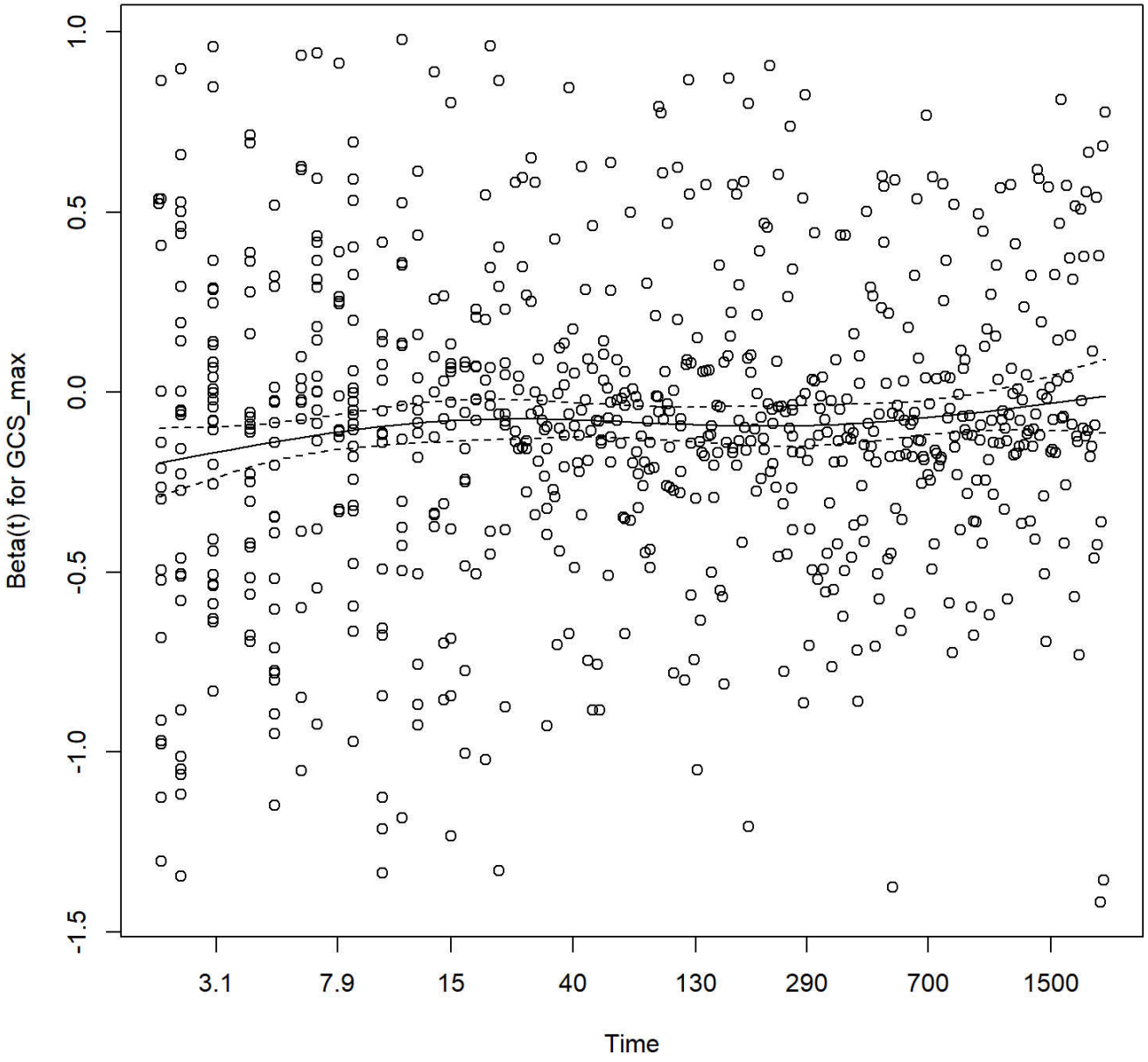

Beta(t) for Age
0.2
0.1
0.0
-0.1
-0.2
3.1
7.9
15
40
130
290
700
1500
Time

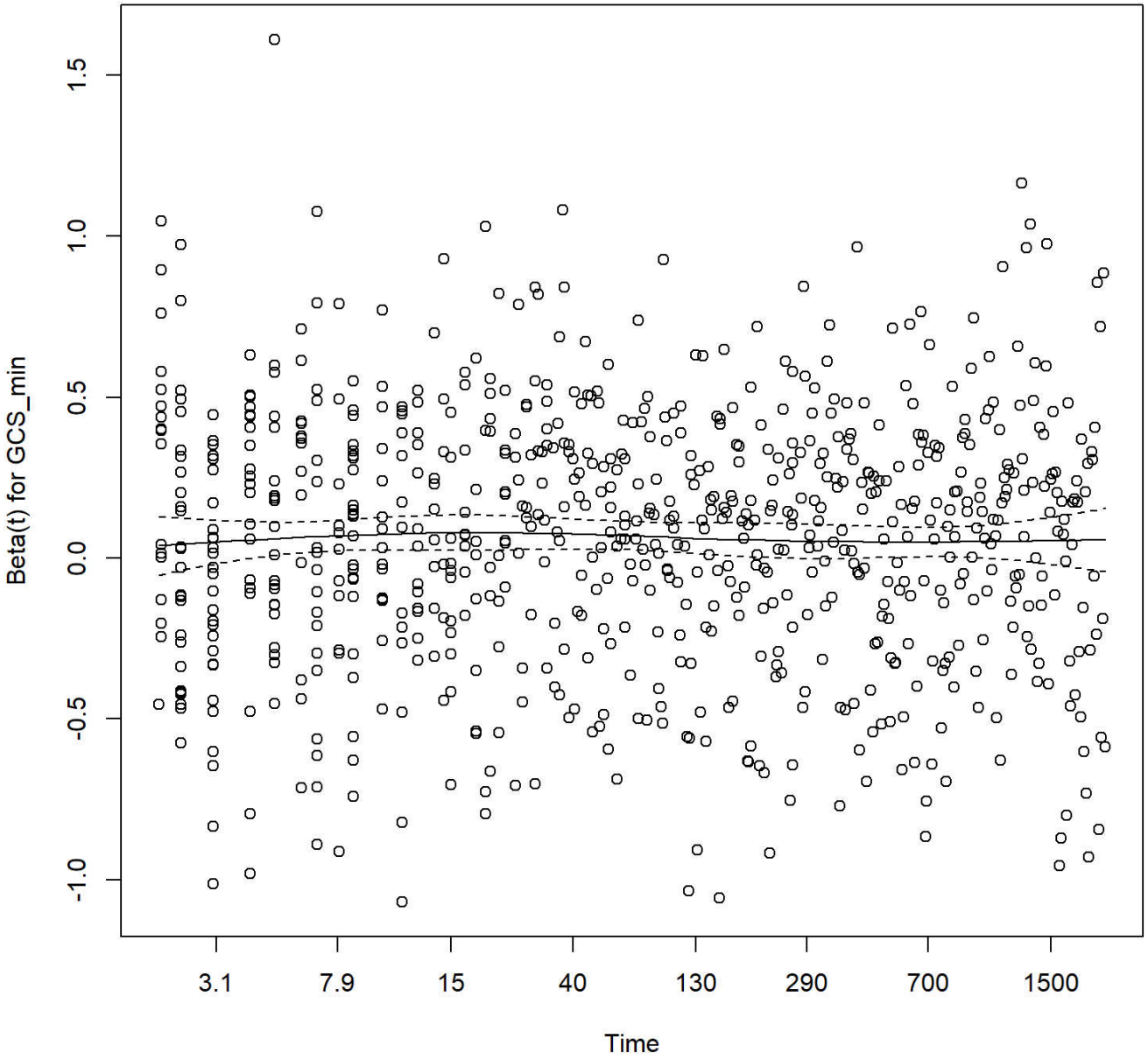

Beta(t) for SAPS1
1.0
0.5
0.0
-0.5
-1.0
3.1
7.9
15
40
130
290
700
1500
Time

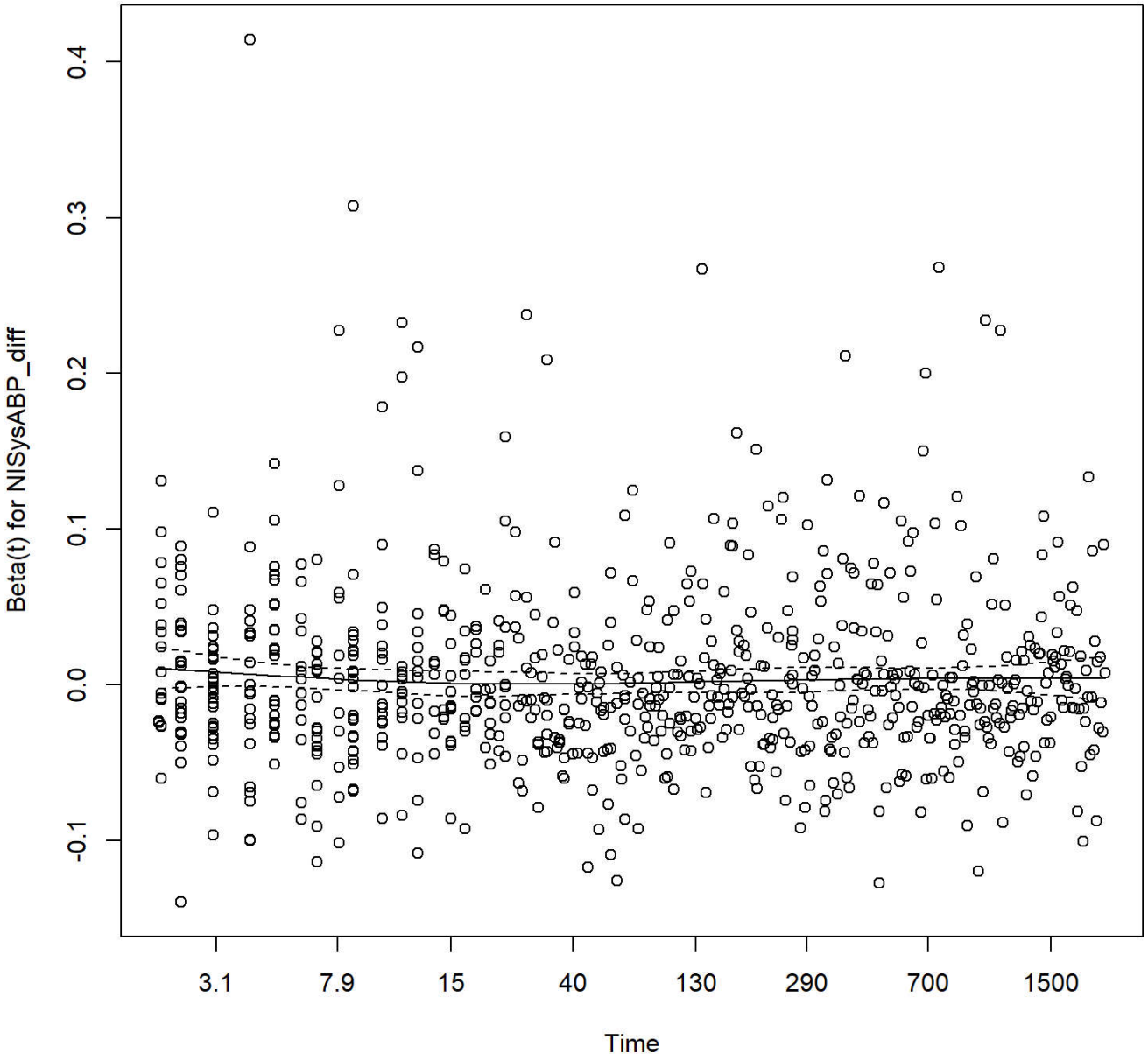

Beta(t) for SOFA
1.5
1.0
0.5
0.0
-0.5
-1.0
3.1
7.9
15
40
130
290
700
1500
Time

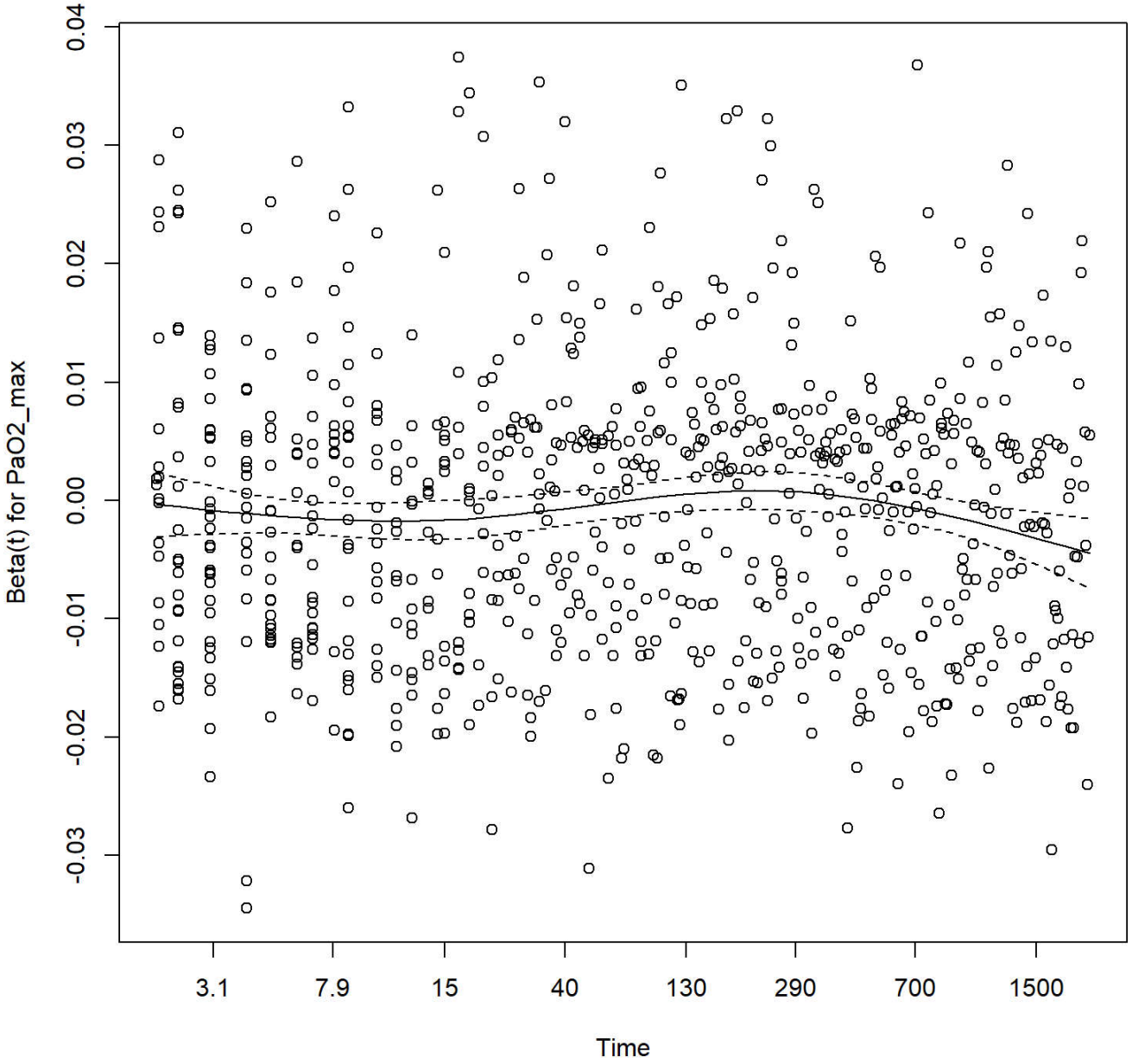

Beta(t) for BUN_max
0.3
0.2
0.1
0.0
-0.1
-0.2
3.1
7.9
15
40
130
290
700
1500
Time

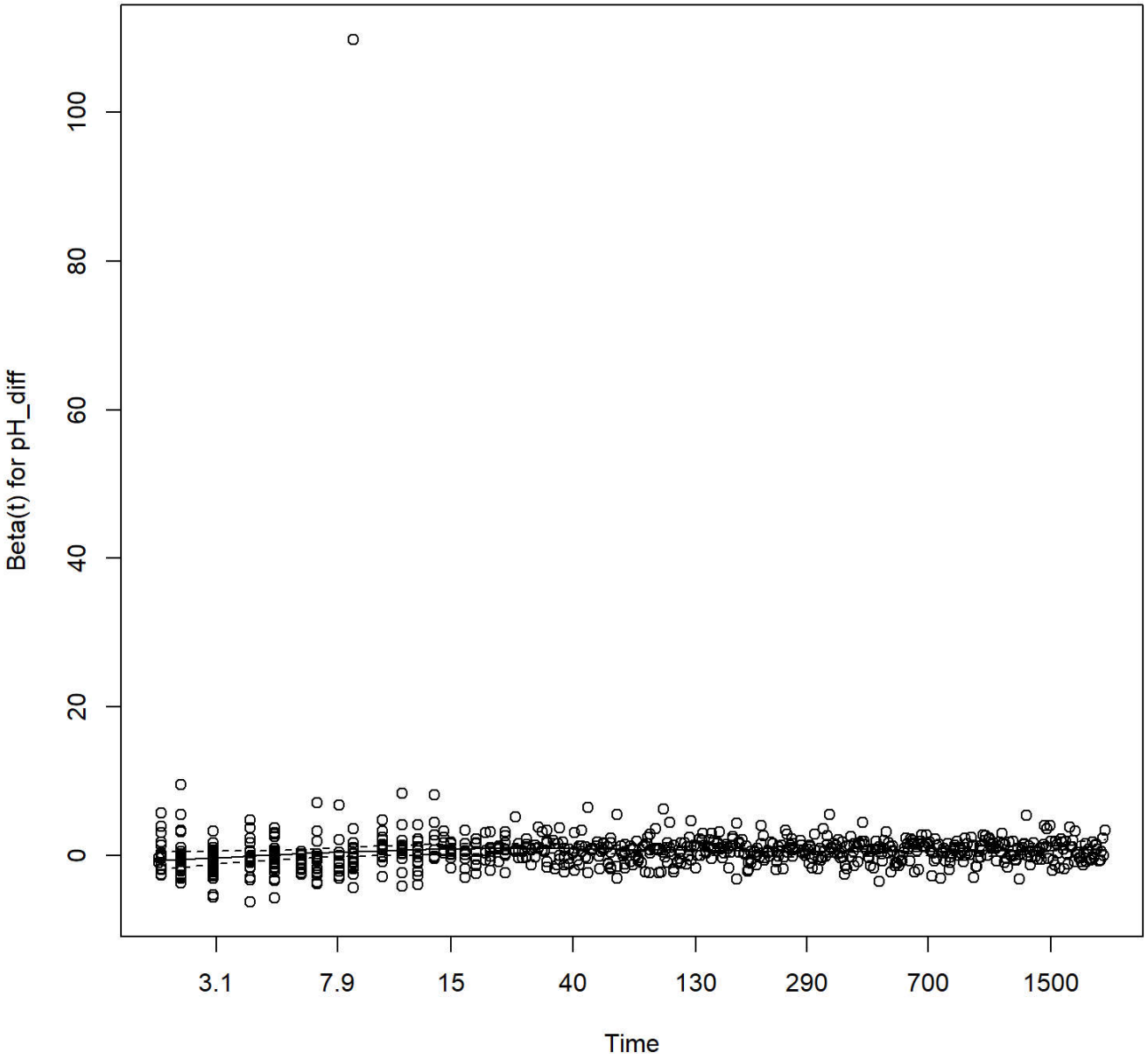

Beta(t) for Creatinine_max
6
4
2
0
-2
3.1
7.9
15
40
130
290
700
1500
Time

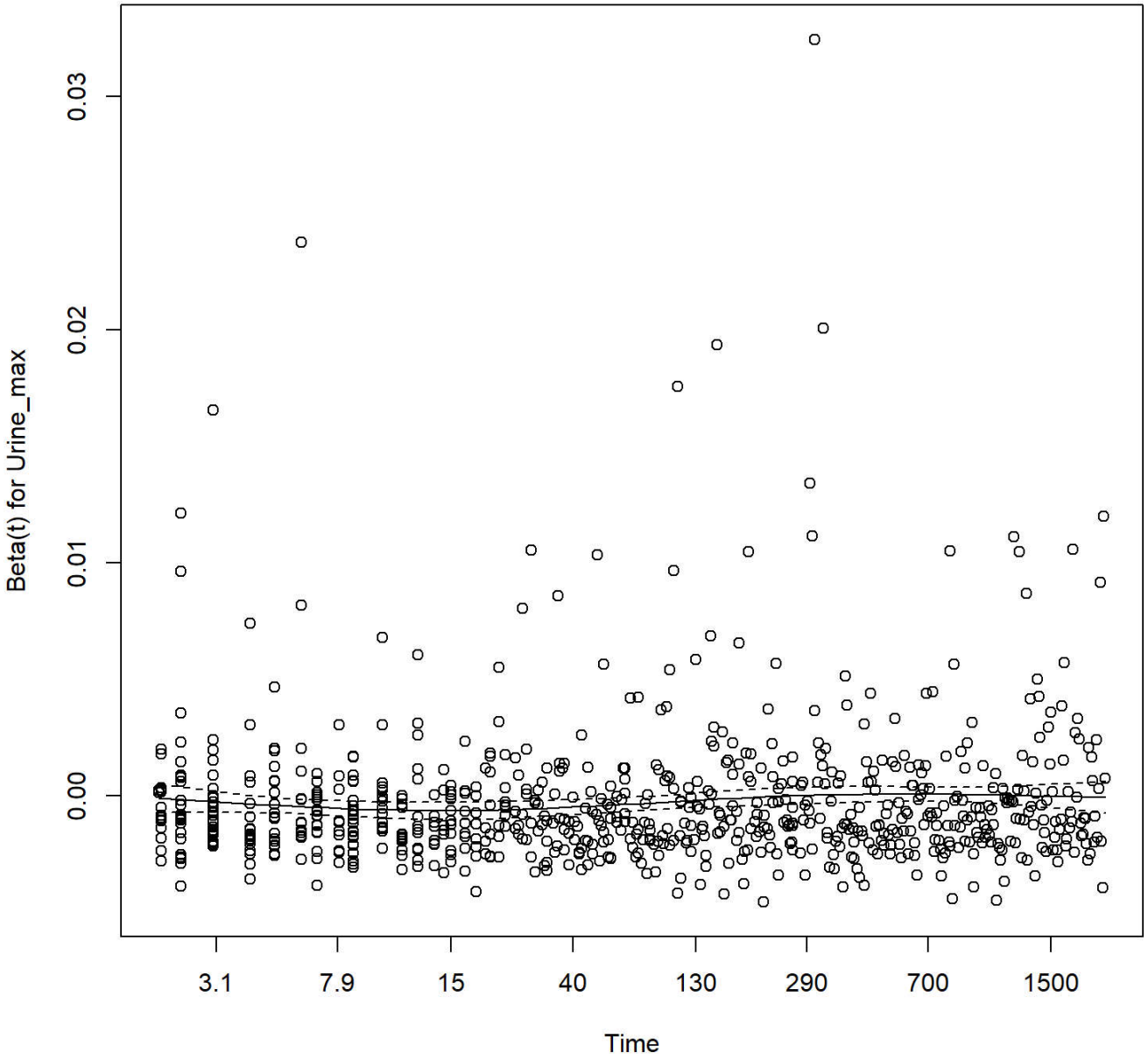

Beta(t) for GCS_max
1.0
0.5
0.0
-0.5
-1.0
-1.5
3.1
7.9
15
40
130
290
700
1500
Time

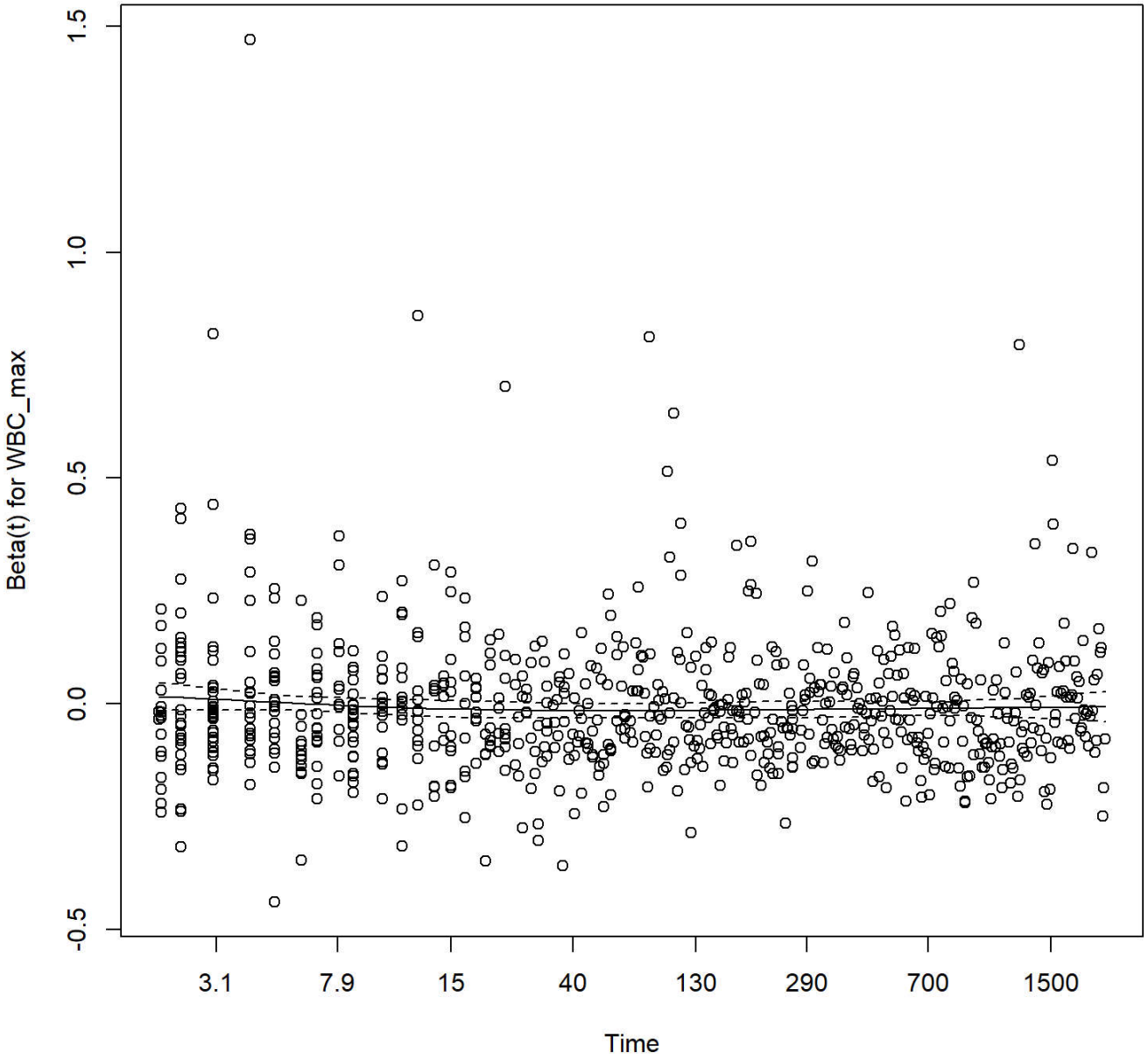

Beta(t) for GCS_min
1.5
1.0
0.5
0.0
-0.5
-1.0
3.1
7.9
15
40
130
290
700
1500
Time

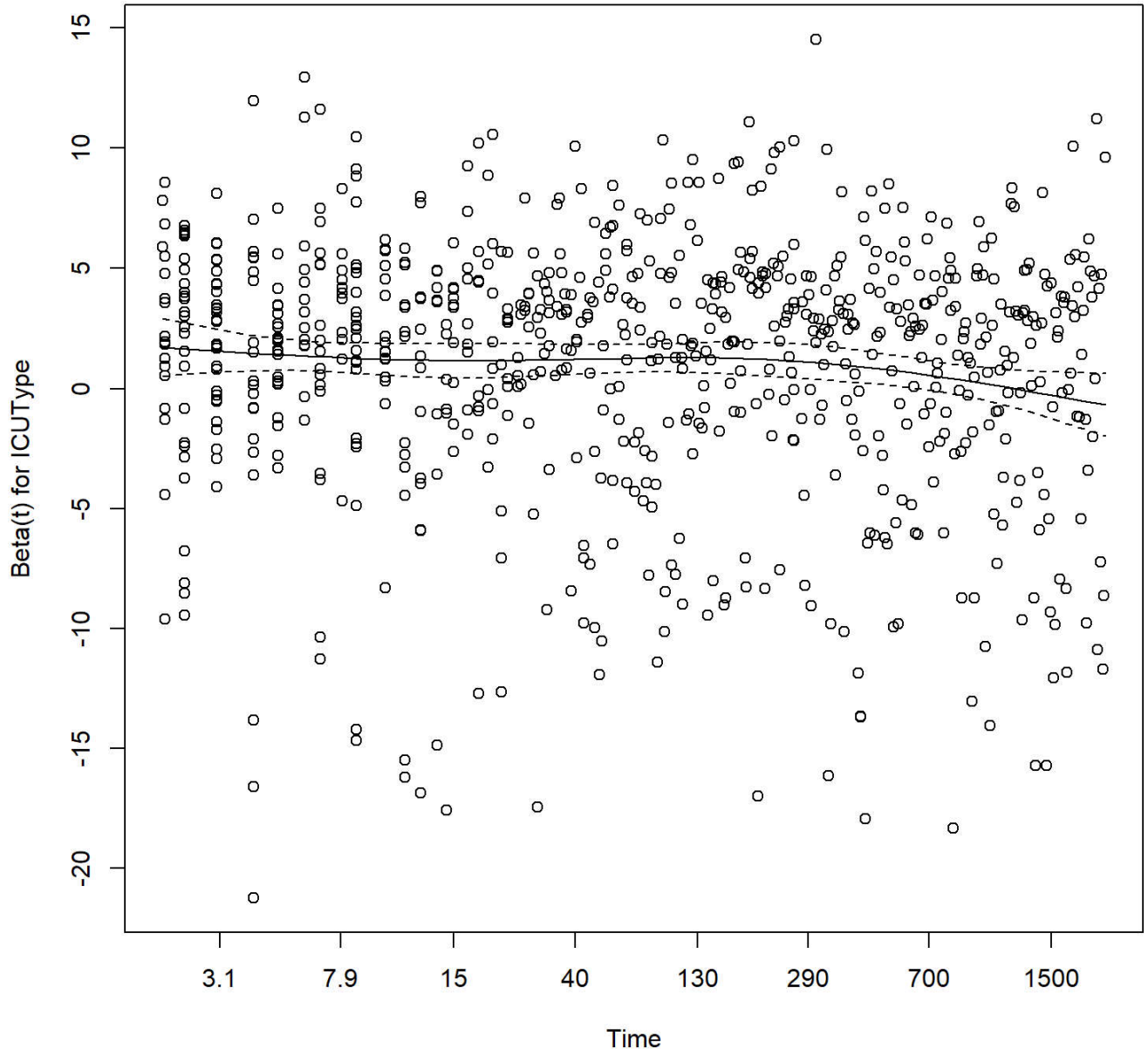

Beta(t) for NISysABP_diff
0.4
0.3
0.2
0.1
0.0
-0.1
3.1
7.9
15
40
130
290
700
1500
Time


Beta(t) for PaO2_max
0.04
0.03
0.02
0.01
0.00
-0.01
-0.02
-0.03
3.1
7.9
15
40
130
290
700
1500
Time


Beta(t) for pH_diff
100
80
60
40
20
0
3.1
7.9
15
40
130
290
700
1500
Time


Beta(t) for Urine_max
0.00
0.01
0.02
0.03
3.1
7.9
15
40
130
290
700
1500
Time


Beta(t) for WBC_max
1.5
1.0
0.5
0.0
-0.5
3.1
7.9
15
40
130
290
700
1500
Time




```
# Perform stepwise selection based on AIC
stepwise_cox_model_df0 <- stepAIC(cox_model, direction = "both")
```

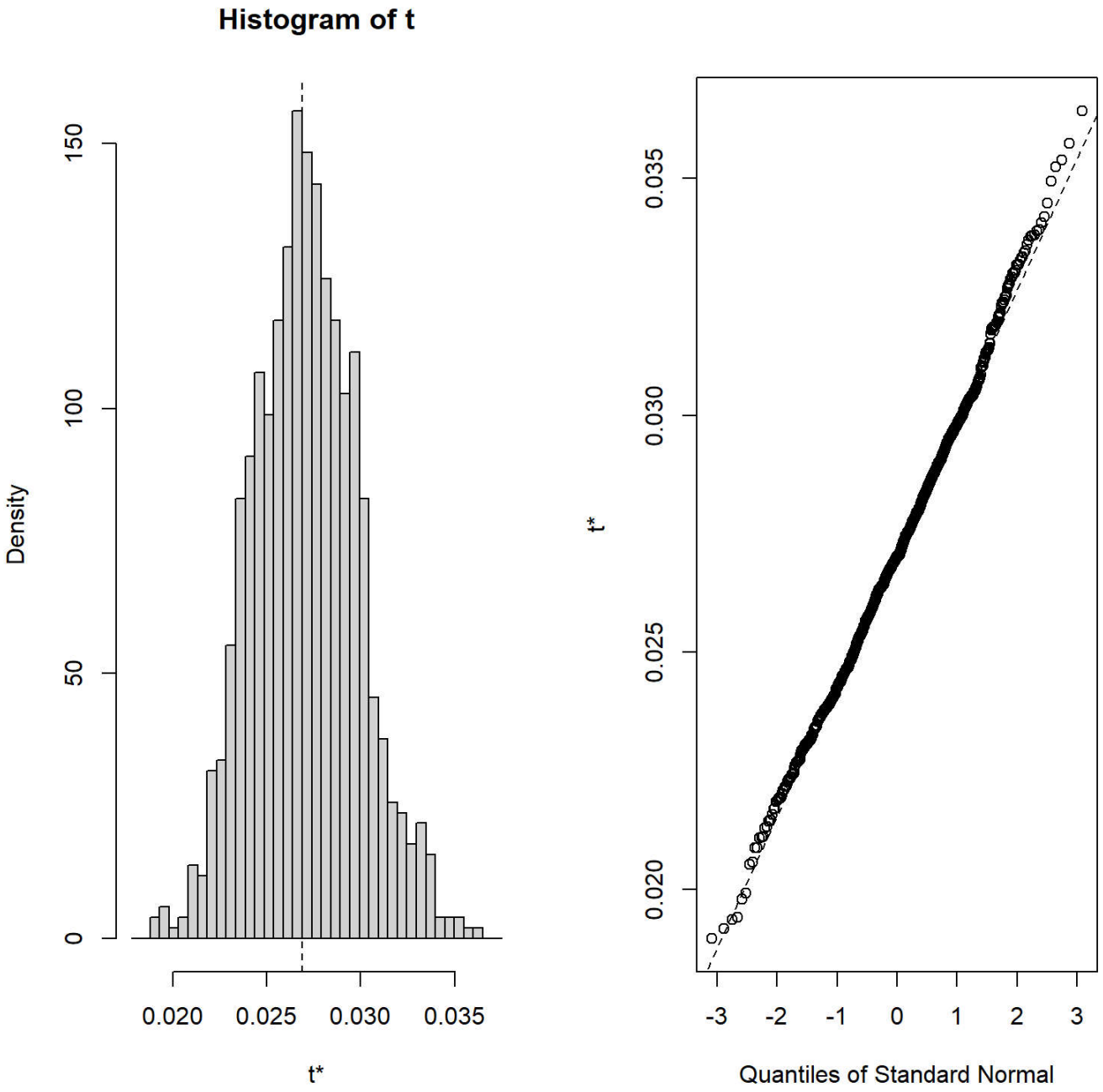
```
## Start:  AIC=10934.26
## Surv(Days, Status) ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max +
##     GCS_max + GCS_min + NISysABP_diff + PaO2_max + pH_diff +
##     Urine_max + WBC_max + ICUType
##
##                   Df   AIC
## <none>               10934
## - Creatinine_max  1 10935
## - WBC_max         1 10935
## - NISysABP_diff   1 10935
## - PaO2_max        1 10937
## - pH_diff         1 10937
## - Urine_max       1 10940
## - SOFA            1 10941
## - GCS_min         1 10950
## - BUN_max         1 10951
## - SAPS1           1 10955
## - ICUType         3 10958
## - GCS_max         1 10969
## - Age             1 11040
```

```
kable(summary(stepwise_cox_model_df0)$coefficients, caption = "Summary of Stepwise Cox Model")
```

Summary of Stepwise Cox Model

	coef	exp(coef)	se(coef)	z	Pr(>|z|)
Age	0.0267929	1.0271550	0.0026999	9.923789	0.0000000
SAPS1	0.0566644	1.0583006	0.0117830	4.808998	0.0000015
SOFA	0.0437166	1.0446863	0.0149756	2.919192	0.0035094
BUN_max	0.0083768	1.0084120	0.0018808	4.453905	0.0000084
Creatinine_max	-0.0426158	0.9582795	0.0271831	-1.567729	0.1169444
GCS_max	-0.0911256	0.9129030	0.0148177	-6.149763	0.0000000
GCS_min	0.0608879	1.0627798	0.0143255	4.250312	0.0000213
NISysABP_diff	0.0035230	1.0035292	0.0020764	1.696680	0.0897572
PaO2_max	-0.0009089	0.9990915	0.0004284	-2.121613	0.0338703
pH_diff	0.5444756	1.7237042	0.1861991	2.924158	0.0034539
Urine_max	-0.0002601	0.9997399	0.0000964	-2.697619	0.0069837
WBC_max	-0.0077808	0.9922494	0.0048667	-1.598775	0.1098707
ICUTypeCardiac Surgery Recovery Unit	-0.6583563	0.5177016	0.1530792	-4.300757	0.0000170
ICUTypeMedical ICU	0.0595781	1.0613886	0.1055182	0.564624	0.5723295
ICUTypeSurgical ICU	-0.1843512	0.8316437	0.1215965	-1.516089	0.1294968

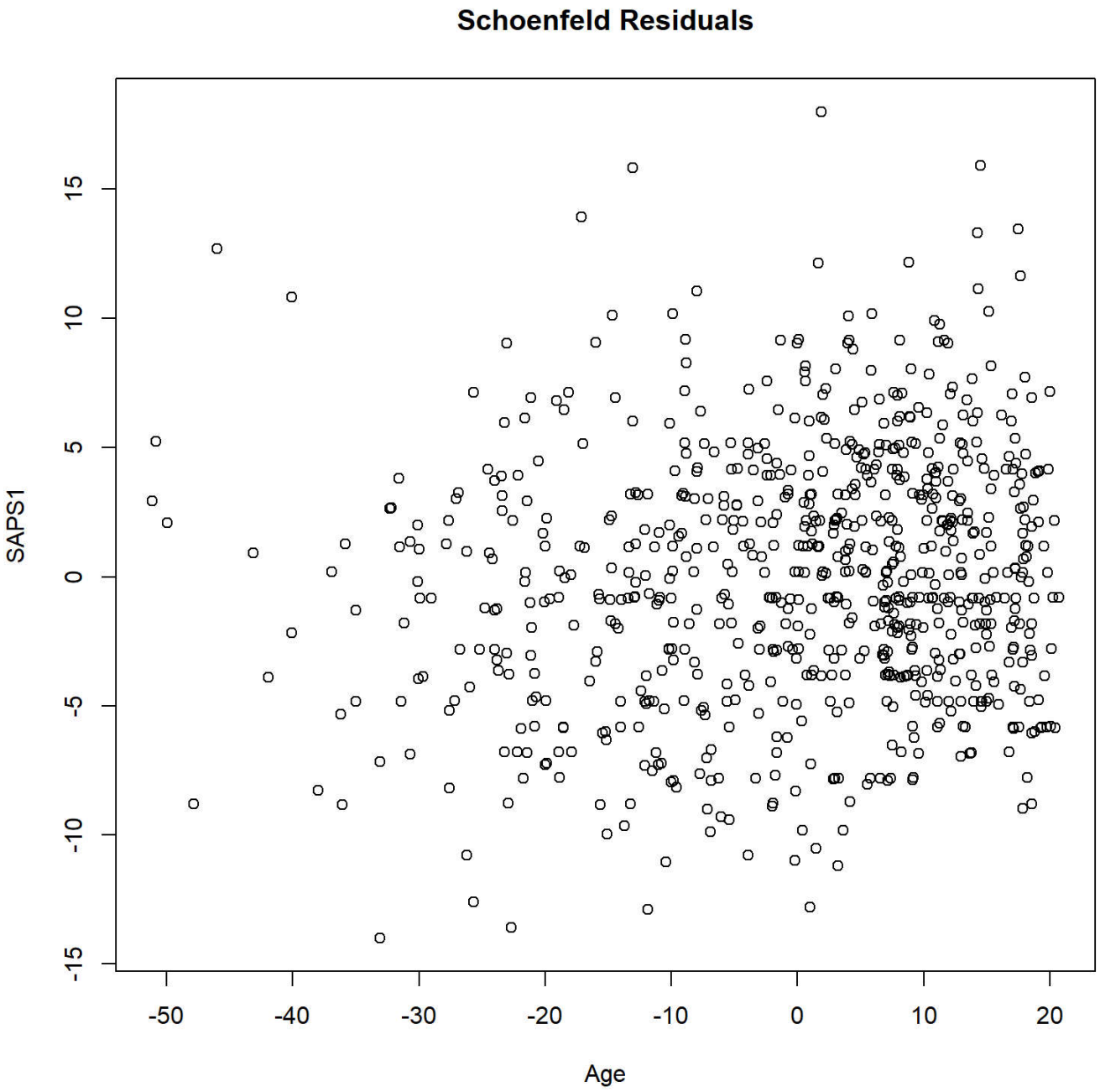
```
# bootstrapping to validate the model (fits a Cox proportional hazards model to a subset of dat
a)
coxph_boot <- function(data, indices) {
  d <- data[indices,] # allows boot to select sample
  fit <- coxph(Surv(Days, Status) ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max + GCS_max + G
CS_min + NISysABP_diff + PaO2_max + pH_diff + Urine_max + WBC_max + ICUType, data = d)
  return(coef(fit)) #returns the coefficients of the fitted model
}

# Bootstrapping with 1000 resamples
results_boot <- boot(icu_patients_df1_cleaned, coxph_boot, R=1000)
plot(results_boot)
```



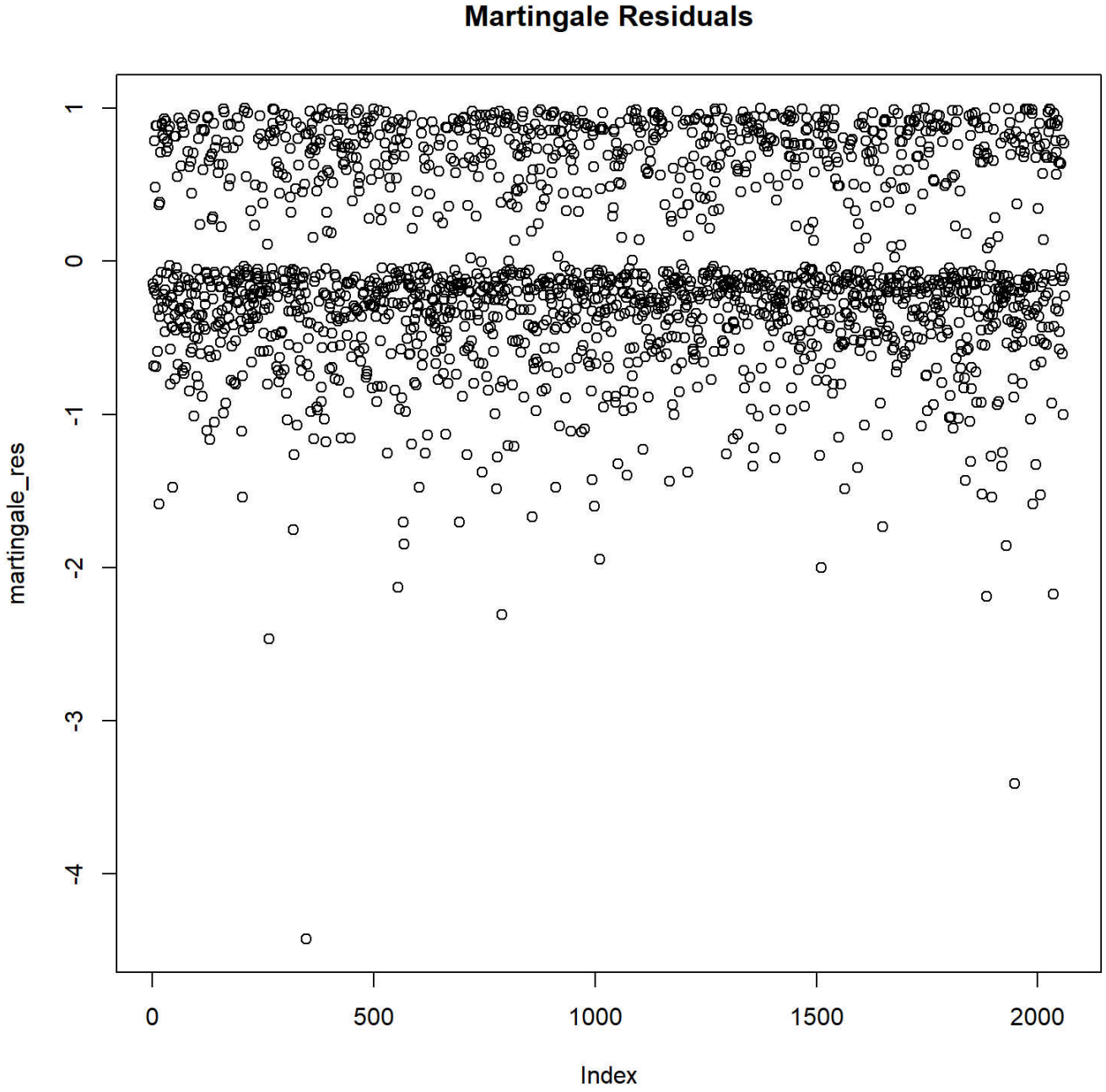
- **Histogram**: Shows the distribution of one of the bootstrap estimates (usually the first coefficient if not specified), helps visualize the variability and bias of the estimate. In here, the distribution looks symmetric about the mean, suggesting that the bootstrap samples do not show much bias. - **Quantile-Quantile (QQ)**

plot: This plot compares the quantiles of the bootstrap estimates against the quantiles of a standard normal distribution. The points lie close to the diagonal line, which implies that the distribution of the bootstrap estimates is approximately normal.

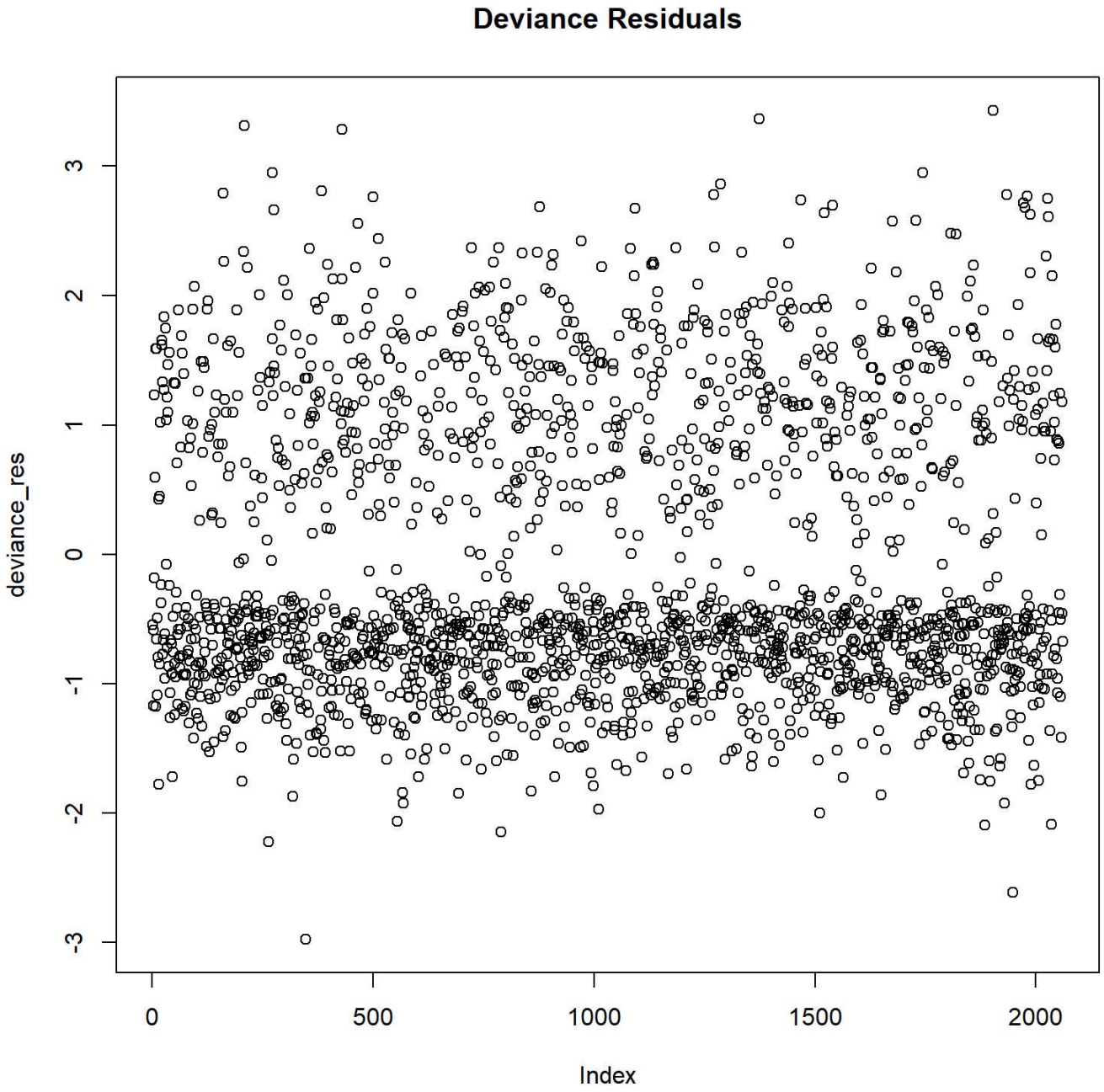
```
# Plot diagnostics
# Compute Schoenfeld residuals
schoenfeld_res <- residuals(stepwise_cox_model, type = "schoenfeld")

# Compute Martingale residuals
martingale_res <- residuals(stepwise_cox_model, type = "martingale")

# Compute deviance residuals
deviance_res <- residuals(stepwise_cox_model, type = "deviance")

# Compute score residuals
score_res <- residuals(stepwise_cox_model, type = "score")

# Plotting Schoenfeld residuals
plot(schoenfeld_res, main = "Schoenfeld Residuals")
```



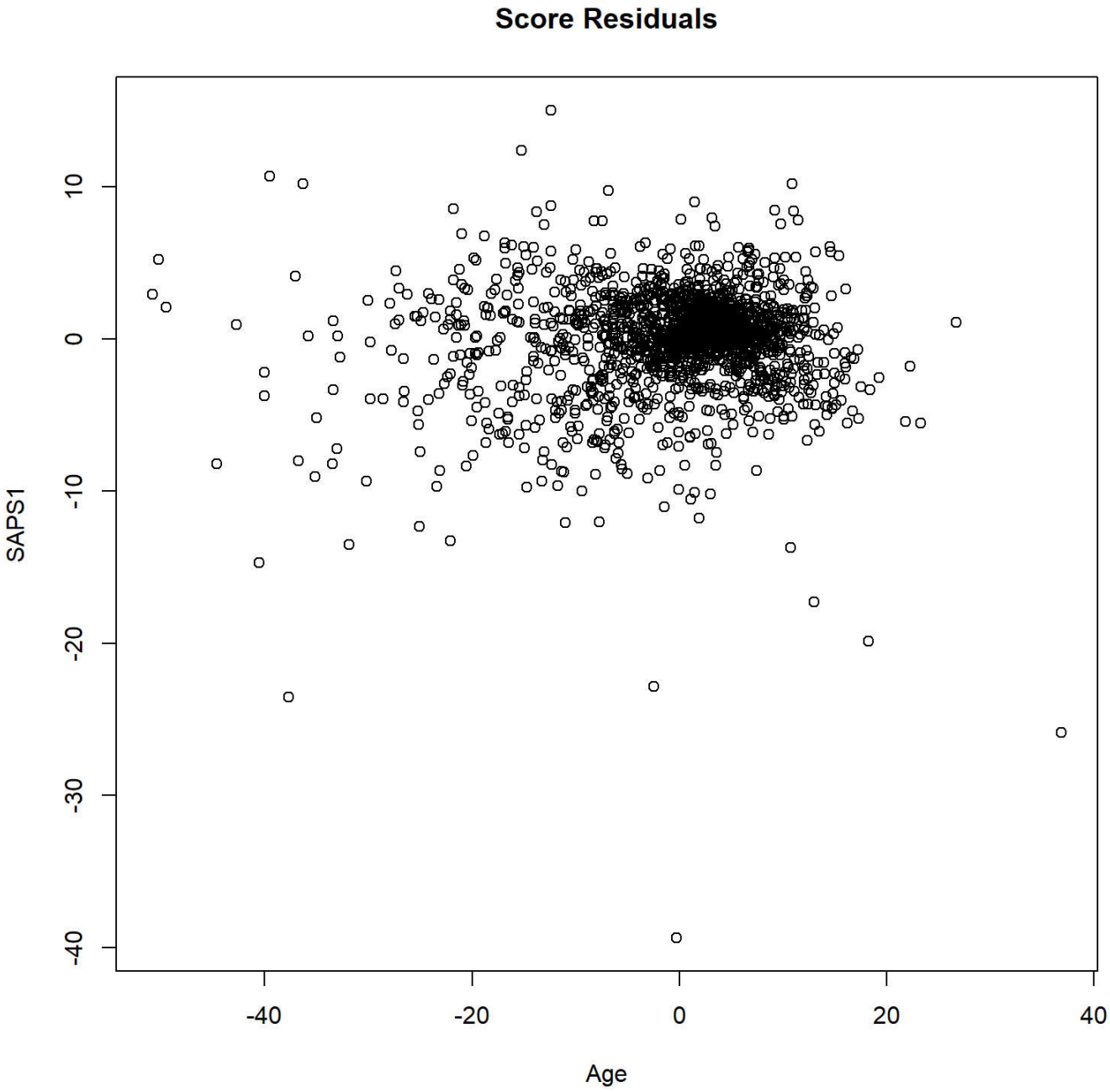

```
# Plotting Martingale residuals
plot(martingale_res, main = "Martingale Residuals")
```



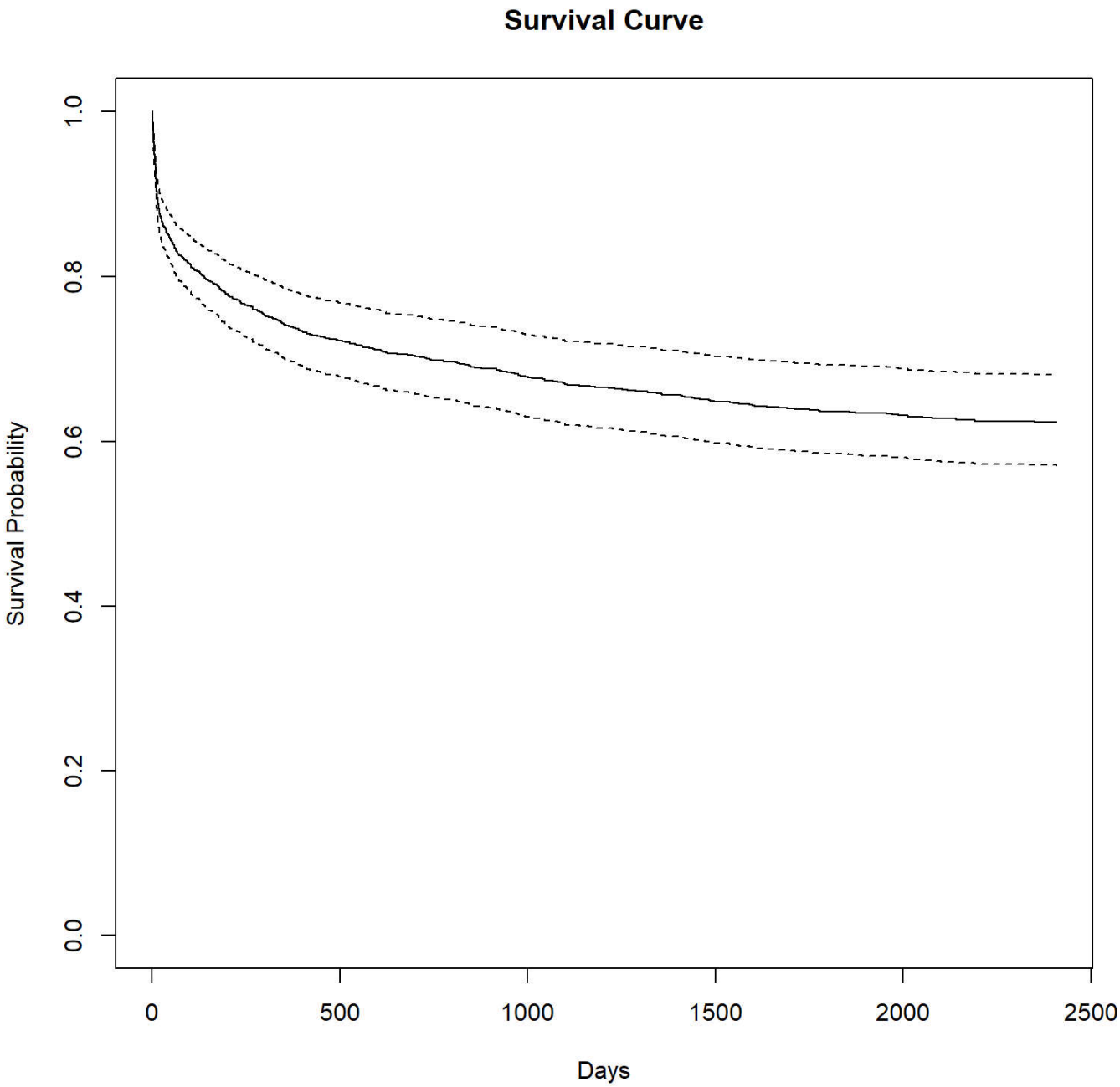

```
# Plotting Deviance residuals
plot(deviance_res, main = "Deviance Residuals")
```




```
# Plotting Score residuals
plot(score_res, main = "Score Residuals")
```



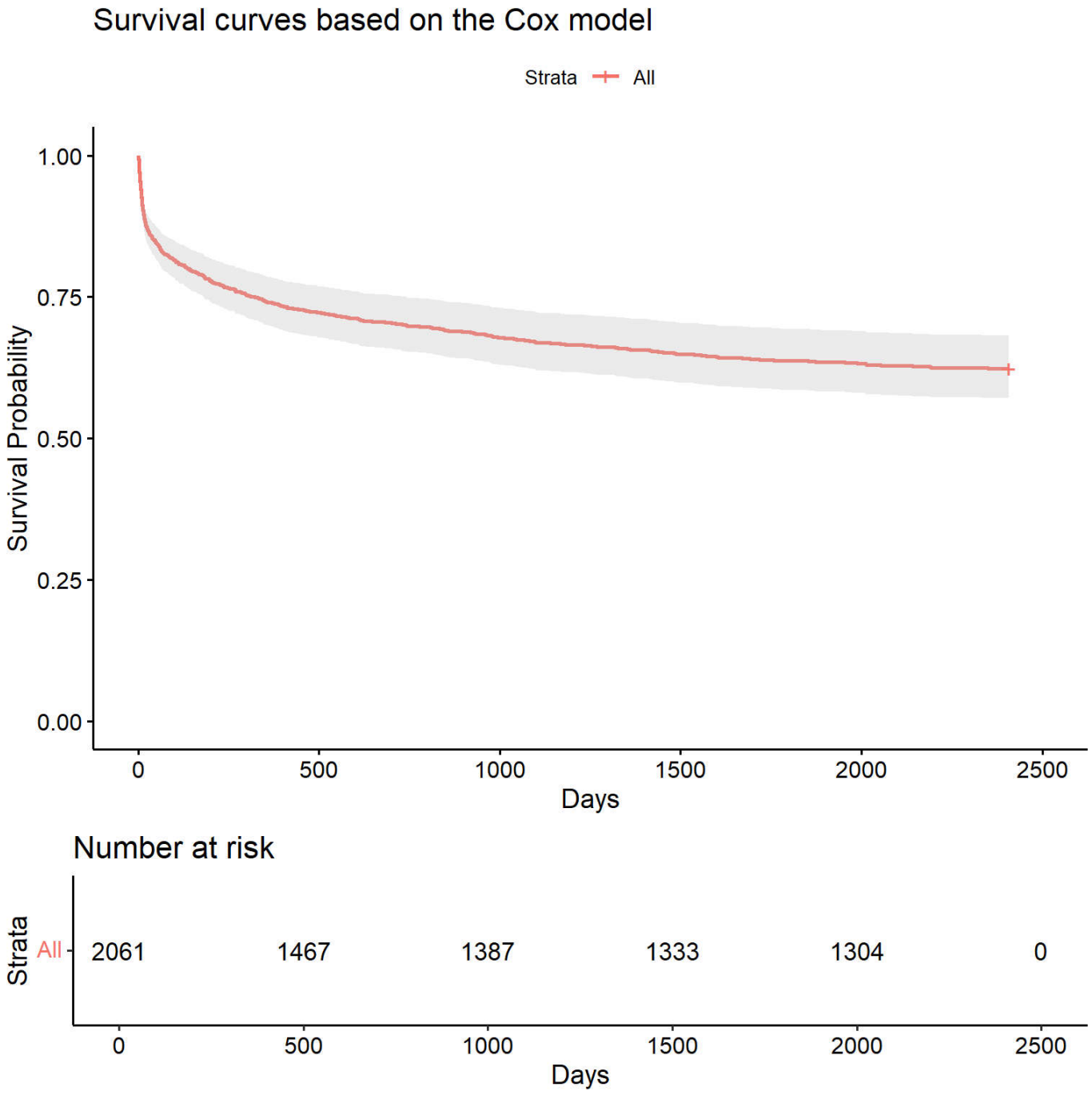

```
# Diagnostic plot
plot(survfit(stepwise_cox_model), xlab = "Days", ylab = "Survival Probability", main = "Survival Curve")
```



1. **Schoenfeld Residuals**: - Calculates Schoenfeld residuals, which are used to check the proportional hazards assumption of a Cox model. Each residual represents the contribution of an individual covariate to the hazard at each event time. - Plots these residuals against time., if the residuals display any systematic pattern over time, it suggests a violation of the proportional hazards assumption. - In this case, residuals lack of clear patterns would suggest that the proportional hazards assumption holds for age. 2. **Martingale Residuals**: - Computes Martingale residuals, which represent the difference between the observed number of events and the expected number under the model, for each individual. - Detecting non-linear effects of covariates and potential outliers. Ideally, residuals should be randomly scattered around zero without clear patterns. - In deviance residuals plot, a random scatter indicates good model fit, while patterns or trends could suggest model inadequacies. 3. **Deviance Residuals**: - Calculates deviance residuals, providing a measure of how well the model predicts each observation. - Like Martingale residuals, these should ideally scatter randomly around zero, aiding in identifying poorly predicted observations. - A random scatter indicates good model fit 4. **Score Residuals**: - Computes score residuals, assessing the contribution of each observation to the overall model fit. - Identify influential cases where an individual observation markedly affects the coefficient estimates. - Dense clustering without extreme outliers is ideal.

```
# Plot survival curves
library(survminer)
ggsurvplot(survfit(initial_cox_model), data = icu_patients_df1_cleaned,
           pval = TRUE, conf.int = TRUE, risk.table = TRUE,
           xlab = "Days", ylab = "Survival Probability",
           title = "Survival curves based on the Cox model")
```

```
## Warning in .pvalue(fit, data = data, method = method, pval = pval, pval.coord = pval.coord,
: There are no survival curves to be compared.
##  This is a null model.
```




```
# Fit a Cox model for baseline hazard comparison
```

```
#cox_model <- coxreg(surv_obj ~ Age + SAPS1 + SOFA + BUN_max + Creatinine_max + GCS_max + GCS_m
in + NISysABP_diff + PaO2_max + pH_diff + Urine_max + WBC_max + ICUType, data = icu_patients_df
1_cleaned)

#check.dist(cox_model, weibull_model)
#check.dist(cox_model, gompertz_model)
#check.dist(cox_model, pch_model)

# outputting coefficients from different models
#cat("Cox Model Coefficients:\n", paste(names(cox_model$coefficients), cox_model$coefficients,
sep = ": ", collapse = "\n "), "\n\n")
#cat("Weibull Model Coefficients:\n", paste(names(weibull_model$coefficients), weibull_model$co
efficients, sep = ": ", collapse = "\n "), "\n\n")
#cat("Gompertz Model Coefficients:\n", paste(names(gompertz_model$coefficients), gompertz_model
$coefficients, sep = ": ", collapse = "\n "), "\n\n")
#cat("PCH Model Coefficients:\n", paste(names(pch_model$coefficients), pch_model$coefficients,
sep = ": ", collapse = "\n "), "\n\n")
```

Hints

1. Make sure you justify your choice of potential predictors. This could be based on a range of different factors, eg. review of the literature, data quality assessment, strong association with the outcome.
2. Present the story of your analysis by including explanation and commentary supported by well-presented plots and/or tables.
3. Make sure all plots and tables are clearly presented with titles and labels as needed. Pay attention to sizing of plots and make sure you check how the rendered result appears.
4. Clearly define your starting population for the analysis and describe the characteristics of your outcome variable.
5. Present summaries of appropriate univariate models examining the unadjusted relationship between each predictor and the outcome.
6. Fit an appropriate series of multivariable regression models, justifying your approach. Assess each model you consider for goodness of fit and other relevant statistics.
7. Present your final model for each task and justify why this model is the most appropriate to answer the analytical question(s). Your final model(s) should **not** include all the predictor variables, just a small subset of them, which you have selected based on statistical significance, background knowledge and relevance to achieving the analytical aim.
8. For your *final model*, present a set of diagnostic statistics and/or charts and comment on them.
9. For each task make sure you write a paragraph or two summarising the most important findings of your analysis. Include reference to the most important results from the statistical output, and a simple clinical interpretation. Make sure your summary of findings addresses the main analytical aim(s) for each task.

Save, knit and submit

Reminder: don't forget to save this file, to knit it to check that everything works and appears in the right format, and then submit TWO documents via the drop box in OpenLearning (.rmd AND .html).

Problems?

If you encounter problems with any part of the process described above, please contact the course convenor via OpenLearning as soon as possible so that the issues can be resolved in good time, and well before the assessment is due.

Additional Information

The instructions are deliberately less prescriptive than the individual assessments to allow you some latitude in what you do and how you go about the task. However, to complete the tasks, you only need to replicate or repeat the steps covered in the course - you do not need to attempt any types of analyses that have not been covered. Refer to the marking rubric for how marks will be awarded.

Note also that with respect to the model fitting, there are no **right** or **wrong** answers when it comes to variable selection and other aspects of model specification. Deep understanding of the underlying medical concepts which govern patient treatment and outcomes in ICUs is not required or assumed, although you should try to gain some understanding of each variable using the links provided. You will not be marked down if your medical justifications are not exactly correct or complete, but do you best, and don’t hesitate to seek help from the course convenor.