From the output, 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, FiO2 variables could be related to respiratory function.

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.

## 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

<li> <em>RecordID:</em> a unique integer for each ICU stay</li>

<li> <em>Age:</em> years</li>

<li> <em>Gender:</em> male/female</li>

<li> <em>Height:</em> cm</li>

<li> <em>ICUType:</em> Coronary Care Unit; Cardiac Surgery Recovery Unit; Medical ICU; Surgical ICU</li>

<li> <em>Length\_of\_stay:</em> The number of days between the patient’s admission to the ICU and the end of hospitalisation</li>

<li> <em>Survival:</em> The number of days between ICU admission and death for patients who died</li>

#### Outcome Variables

<li> <em>in\_hospital\_death:</em> 0:survivor/1:died in-hospital \*\*this is the outcome variable for Task 1: Logistic Regression\*\*</li>

<li> <em>Status:</em> True/False \*\*this is the censoring variable for Task 2: Survival Analysis\*\*</li>

<li> <em>Days:</em> Length of survival (in days) \*\*this is the survival time variable for Task 2: Survival Analysis\*\*</li>

## 本次评估的数据

用于此评估的数据包括美国重症监护病房 (ICU) 住院记录。 所有患者都是因各种原因入院的成年人。 ICU 停留时间少于 48 小时的情况已被排除。

评估的源数据是 2012 年 MIT PhysioNet/心脏病学挑战赛免费提供的数据。 详细信息请参见[此处](https://physionet.org/challenge/2012/)。 已使用训练集 A 数据。 原始数据已被修改和组合以适应本次评估的目的。

数据帧由 120 个变量组成，定义如下：

#### 患者描述符变量

RecordID：每次 ICU 住院的唯一整数

Age年龄：岁

Gender性别：男/女

Height身高： 厘米

ICUType ICU类型：冠心病监护室； 心脏手术康复中心； 医疗重症监护室； 外科重症监护室

Length\_of\_stay：患者入住 ICU 到住院结束之间的天数</li>

Survival生存率：死亡患者入住 ICU 与死亡之间的天数</li>

#### 结果变量

in\_hospital\_death：0：幸存者 / 1：在医院死亡 这是任务 1：Logistic 回归的结果变量\*\*

Status： True/False \*\*这是任务 2：生存分析的审查变量\*\*

Days：生存时间（以天为单位）\*\*这是任务 2：生存分析的生存时间变量\*\*

# Exclude observations where SAPS1 is NA as SAPS1 has 96 missing values

valid\_data <- icu\_patients\_df1[!is.na(icu\_patients\_df1$SAPS1), ]

intact\_model\_rf <- glm(in\_hospital\_death ~ Age + SAPS1 + SOFA + Albumin\_max + AST\_max + Bilirubin\_max + BUN\_max +

Creatinine\_max + GCS\_max + GCS\_min + HR\_diff + Lactate\_max + Na\_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\_rf)

cox\_model <- coxreg(Surv(Days, Status) ~ 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, data = icu\_patients\_df1\_cleaned)

weibull\_model <- phreg(Surv(Days, Status) ~ 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, data = icu\_patients\_df1\_cleaned, dist = "weibull")

gompertz\_model <- phreg(Surv(Days, Status) ~ 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, data = icu\_patients\_df1\_cleaned, dist = "gompertz")

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$coefficients), weibull\_model$coefficients, 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")

To manage the length of your HTML document produced by R Markdown and ensure it fits within the 10-page limit, you can selectively hide less crucial code chunks, tables, and figures. Here are several strategies you can use within your R Markdown document to control what gets included in the final output:

**1. Hide Code Chunks**

You can hide the R code for specific chunks while still displaying the outputs (like figures or tables) by using the **echo = FALSE** chunk option. Here's how to apply it:

```{r results='asis', echo=FALSE}

# Your R code here

### 2. Collapse Code Chunks

For code that is important but not necessary to show by default, you can use the `collapse` option to allow readers to expand it if desired:

```markdown

```{r collapse=TRUE}

# Your R code here

### 3. Hide Messages and Warnings

To make the output cleaner, you can suppress messages and warnings from appearing in the output:

```markdown

```{r warning=FALSE, message=FALSE}

# Your R code that generates warnings or messages

### 4. Use Smaller Plots

When you have many plots, you can arrange them in a grid to save space:

```markdown

```{r fig.show='hold'}

# Multiple plot commands here

plot1()

plot2()

plot3()

plot4()

### 5. Selective Inclusion of Outputs

For data summaries or tables that are too detailed and not necessary for the final document, consider summarizing the results or showing only key parts of the tables. You can use `kable()` from the `knitr` package to create neat tables and optionally use the `kableExtra` package to style them minimally.

### 6. External Links to Detailed Outputs

If detailed data or extensive analyses are crucial but too voluminous, consider hosting the detailed tables or figures online (e.g., GitHub, Dropbox) and include links in your document.

### 7. Section Folding

For longer sections that are not essential for a direct read but you prefer to keep them accessible within the document, you can use HTML folding:

```html

<details>

<summary>Click to expand!</summary>

```{r}

# Your R code here

**8. Adjust Plot Sizes**

Control the size of the plots directly in the chunk options to reduce space:

```{r fig.width=4, fig.height=4}

# Your plot code

**eval=FALSE 显示代码 不运行**

echo = FALSE：此选项可防止在编织文档时评估块内的代码。 该代码存在于文档中用于演示或文档目的，但不会运行以产生输出或结果。

include = FALSE：此选项确保块中的代码及其输出（如果已执行）不会出现在最终文档中。 这对于设置不需要在最终文档中显示的数据或加载库非常有用。

1. **<em></em>**: Emphasizes text, typically rendered as italicized text. Equivalent to Markdown's **\*text\*** or **\_text\_**.
2. **<li></li>**: Represents a list item within ordered (**<ol>**) or unordered (**<ul>**) lists.
3. **<strong></strong>**: Makes text bold, similar to Markdown's **\*\*text\*\*** or **\_\_text\_\_**.
4. **<br>**: Inserts a line break. Useful for spacing out content without starting a new paragraph.
5. **<hr>**: Creates a horizontal rule or line break across the page, often used for thematic breaks in the content.
6. **<a href=""></a>**: Defines a hyperlink. You can embed links to other websites or resources.
7. **<table></table>, <tr></tr>, <th></th>, and <td></td>**: These tags are used for constructing tables manually, though in R Markdown you might often use functions from the **knitr** and **kableExtra** packages to create tables programmatically.
8. **<div></div>**: A block-level container that can be used to group large blocks of HTML elements together and apply CSS styles.
9. **<span></span>**: An inline container used to mark up a part of a text, or a part of a document. It can be used to apply styles to text without causing a line break or a structural change.
10. **<img src="" alt="">**: Embeds an image. The **src** attribute specifies the URL of the image, and the **alt** attribute provides alternative text.
11. **<h1></h1> to <h6></h6>**: Header tags, where **<h1>** is the highest (and usually largest) level and **<h6>** is the lowest.
12. **<p></p>**: Paragraph tag, used to define a block of text.