Heart Failure Survival Study

Mailman School of Public Health at Columbia University

Huanyu Chen, Runze Cui, Jiahe Deng, Yi Huang, Xuesen Zhao
P8108 Final Project Report: Group 6

Dec 7, 2023

Abstract

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Contents

1 Introduction **1**

1.1 Background 1

1.2 Objective 1

2 Exploratory Data Analysis (EDA) **1**

3 Methods **4**

3.1 Nonparametric Estimate 4

3.2 Hypothesis Testing 4

3.3 Semi-parametric Estimate 4

3.4 Cox Model Selection 4

3.5 Parametric Estimate 4

3.6 Model Validation 5

3.6.1 ROC Curves and AUC 5

3.6.2 C-Index 5

3.6.3 Calibration Slope 5

4 Results **5**

4.1 Model Validation 5

4.1.1 ROC Curves and AUC 5

4.1.2 C-Index 5

4.1.3 Calibration Slope 6

5 Discussion **6**

6 Conclusions **6**

7 References **7**

8 Appendix **8**

8.1 Code 8

1 Introduction

1.1 Background

Heart failure (HF) occurs when the muscles in the heart wall weaken and enlarge, impairing the heart's ability to pump blood effectively. This condition can cause the heart's ventricles to become stiff, hindering their ability to fill properly between beats. Over time, the heart becomes less capable of meeting the body's demand for blood, leading to symptoms like difficulty in breathing as the heart struggles to function efficiently.(Ahmad et al., 2017).

According to the statistics, heart failure affects 1-2% of adults in the general population and is more common in older individuals, with over 10% of those aged over 70 years being diagnosed. The actual prevalence might be as high as 4%, as heart failure is often undiagnosed or misdiagnosed, especially in the elderly. Since 2002, the prevalence of heart failure has increased by nearly 25%, driven by factors such as an aging population, better survival rates post-coronary events, and a rise in risk factors like hypertension and atrial fibrillation. (Jones et al., 2019).

1.2 Objective

Our project aims to assess the influence of key physiological and clinical factors on the outcomes of heart failure patients at the Institute of Cardiology and Allied Hospital, Faisalabad, Pakistan, during April-December 2015. We will examine variables such as creatinine levels, gender, age, ejection fraction, blood pressure, anemia, and serum sodium to determine their impact on patient prognosis. Utilizing a range of analytical techniques including exploratory data analysis, nonparametric methods, Kaplan-Meier curves, Cox proportional hazards modeling, parametric survival models, and validation procedures, our study is designed to identify crucial predictors of patient outcomes and their interrelationships. The insights gained from this analysis are expected to contribute significantly to the development of tailored treatment strategies and improved risk stratification models, thereby enhancing clinical decision-making and patient care for heart failure management.

2 Exploratory Data Analysis (EDA)

The present study focuses on 299 heart failure patients, including 105 women and 194 men. All participants were over 40 years old and diagnosed with left ventricular systolic dysfunction, classified under NYHA classes III and IV. The follow-up duration ranged from 4 to 285 days, with an average of 130 days. Diagnosis of the disease was confirmed through cardiac echocardiogram reports or physician's notes. A brief description of variables in the dataset is shown below:

- **age**: Age in years
- **time**: Survival time in days
- **event**: Event binary indicator (0 = Censored, 1 = Event)
- **gender**: Sex binary indicator (0 = Female, 1 = Male)
- **smoking**: Smoking status (0 = No smoking, 1 = Smoking)
- **diabetes**: Diabetes status (0 = No diabetes, 1 = Diabetes)
- **bp**: Blood pressure status (0 = Normal, 1 = Hypertension)
- **anemia**: Anemia status (0 = No anemia, 1 = Anemia: patients with haematocrit < 36)
- **EF_cat**: Ejection fraction (Low: EF ≤ 30, Medium: $30 <$ EF ≤ 45 and High: EF > 45)
- **sodium**: Sodium in mEq/L
- **creatinine**: Serum creatinine in mg/dL
- **platelets**: Platelets in mcL
- **cpk**: Creatinine phosphokinase in U/L

This study falls into the category of Overall Survival (OS), where event indicator equals 1 indicates the death of the subject and is the endpoint of survival. Specifically, 203 subjects were right-censored and 96 subjects have event. Detailed descriptive statistics table stratified by survival status are presented below.

Table 1: Descriptive Statistics Table

	Censored	Event	Overall	P-value
	(N=203)	(N=96)	(N=299)	
Survival time (days)				
Mean (SD)	158 (67.7)	70.9 (62.4)	130 (77.6)	<0.001
Median [Min, Max]	172 [12.0, 285]	44.5 [4.00, 241]	115 [4.00, 285]	
Age (years)				
Mean (SD)	58.8 (10.6)	65.2 (13.2)	60.8 (11.9)	<0.001
Median [Min, Max]	60.0 [40.0, 90.0]	65.0 [42.0, 95.0]	60.0 [40.0, 95.0]	
Gender				
0	71 (35.0%)	34 (35.4%)	105 (35.1%)	1
1	132 (65.0%)	62 (64.6%)	194 (64.9%)	
Smoking status				
0	137 (67.5%)	66 (68.8%)	203 (67.9%)	0.932
1	66 (32.5%)	30 (31.3%)	96 (32.1%)	
Diabetes				
0	118 (58.1%)	56 (58.3%)	174 (58.2%)	1
1	85 (41.9%)	40 (41.7%)	125 (41.8%)	
Blood Pressure				
0	137 (67.5%)	57 (59.4%)	194 (64.9%)	0.214
1	66 (32.5%)	39 (40.6%)	105 (35.1%)	
Ejection Fraction (EF_cat)				
Low	42 (20.7%)	51 (53.1%)	93 (31.1%)	<0.001
Medium	115 (56.7%)	31 (32.3%)	146 (48.8%)	
High	46 (22.7%)	14 (14.6%)	60 (20.1%)	
Anemia				
0	120 (59.1%)	50 (52.1%)	170 (56.9%)	0.307
1	83 (40.9%)	46 (47.9%)	129 (43.1%)	
Serum Sodium (mEq/L)				
Mean (SD)	137 (3.98)	135 (5.00)	137 (4.41)	0.002
Median [Min, Max]	137 [113, 148]	136 [116, 146]	137 [113, 148]	
Serum creatinine (mg/dL)				
Mean (SD)	1.18 (0.654)	1.84 (1.47)	1.39 (1.03)	<0.001
Median [Min, Max]	1.00 [0.500, 6.10]	1.30 [0.600, 9.40]	1.10 [0.500, 9.40]	
Creatinine phosphokinase (U/L)				
Mean (SD)	540 (754)	670 (1320)	582 (970)	0.369
Median [Min, Max]	245 [30.0, 5210]	259 [23.0, 7860]	250 [23.0, 7860]	
Plateletes (mcL)				
Mean (SD)	267000 (97500)	256000 (98500)	263000 (97800)	0.399
Median [Min, Max]	263000 [25100, 850000]	259000 [47000, 621000]	262000 [25100, 850000]	

Based on the descriptive table, we can observe that the mean survival time for deletions and events is 158 days and 70.9 days, respectively. Since we have the complete dataset, there is no need to worry about missingness issue. The table also lists the p-values for each variable. Some variables have relatively large p-values. However, we still need to check the distribution of each variable [1] and determine which variables need to be transformed.

[1]Histograms for continuous variables and bar charts for categorical variables

Figure 1.1: Histograms of Continuous Variables

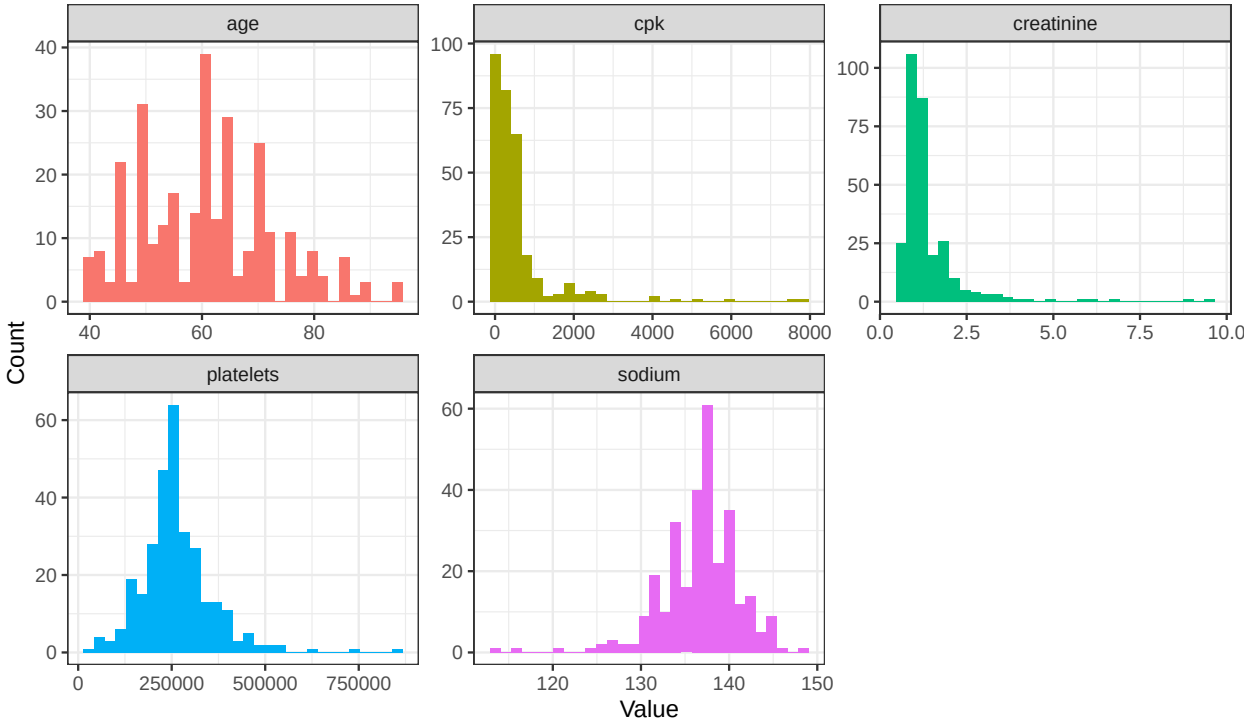
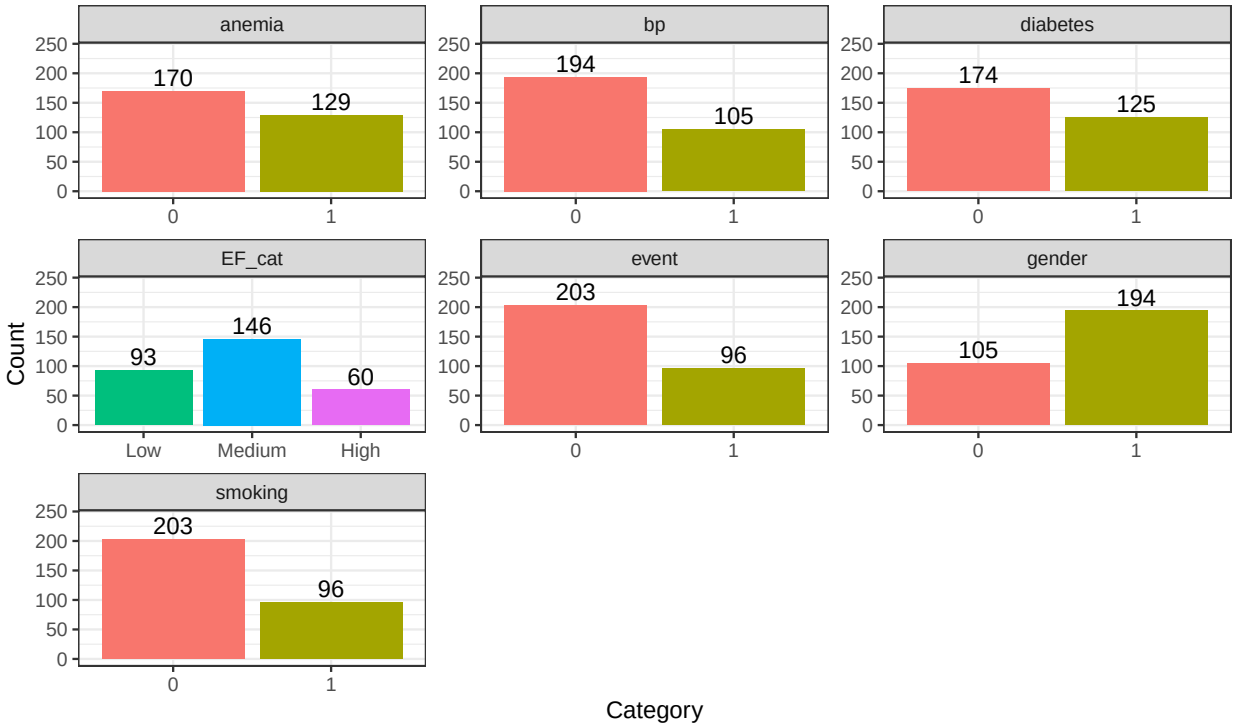


Figure 1.2: Bar Charts of Categorical Variables



After checking the histograms, two continuous variables creatinine phosphokinase (cpk) and serum creatinine are right-skewed. We decide to log-transformed both of them for further model fitting.

3 Methods

3.1 Nonparametric Estimate

(Life table separated by gender, KM and FH comparing)

3.2 Hypothesis Testing

(Log-Rank test and Wilcoxon test)

3.3 Semi-parametric Estimate

Suppose that there is a true survival time, T, as well as a true censoring time, C. The survival time represents the time at which the event of interest occurs: in this dataset, the time at which the patient die. The censoring time is the time at which the patient drop out of the study or survived until the last day of the study.

We observed the Survival Time T and Censoring Time C. Suppose there is a random variable Y

$$Y = min(T, C)$$

In other words, if the event occurs before the censoring such that T < C, then we observed the true survival time T. If censoring occurs before the event such as T > C, then we observe the censoring time. The status indicator as,

$$\delta = \begin{cases} 1 & T \leq C \\ 0 & T > C \end{cases}$$

Thus, $\delta = 1$ if we observe the true survival time, and $\delta = 0$ if we observe the censoring.

We use the Cox-propositional hazard model to evaluate the effect of several factors on survival time. It allows us to examine how specified factors influence the rate of the event that we are interested in at a particular point in time. This rate is the hazard rate. The final model obtained from Stepwise Selection using AIC contains creatinine, age, ejection fraction, blood pressure status, sodium covariates.

The Cox model is expressed by the hazard function denoted by h(t). Briefly, the hazard function can be interpreted as the risk of death at time t. The final model as follows:

$$h(t) = h_0(t) \exp[\beta_0 + X_1\beta_{age} + X_2\beta_{EF_{medium}} + X_3\beta_{EF_{high}} + X_4\beta_{bp} + X_5\beta_{sodium} + \log(x_6 + 1)\beta_{creatinine}]$$

where,

3.4 Cox Model Selection

(model selection (survival tree - optional), and model checking???, fitting the model)

3.5 Parametric Estimate

(Model checking, fitting the model)

3.6 Model Validation

3.6.1 ROC Curves and AUC

In our analysis, we employed time-dependent Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) values to evaluate the discriminative ability of our Cox proportional hazards model over time. Specifically, we focused on two clinically relevant time points: 50 days and 250 days (Ahmad et al., 2017). The ROC curve is a graphical representation that illustrates the diagnostic ability of a binary classifier system by plotting the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various threshold settings. AUC, a key summary measure of the ROC curve, quantifies the overall ability of the model to discriminate between individuals who will experience the event and those who will not, irrespective of the chosen probability threshold (Heagerty & Zheng, 2005). Higher AUC values indicate better discriminative ability.

3.6.2 C-Index

The concordance index (C-index) was calculated to assess the predictive accuracy of our model. This metric is a measure of the model's ability to correctly rank the survival times of pairs of individuals, considering censored data (Steyerberg & Vergouwe, 2014). The C-index is calculated through pairwise comparisons, where a pair is concordant if the individual predicted to have a shorter survival time indeed experiences the event earlier than the other individual in the pair (Steyerberg & Vergouwe, 2014). A C-index of 0.5 suggests no better predictive accuracy than random chance, while a value of 1 indicates perfect prediction.

3.6.3 Calibration Slope

To evaluate the calibration of our model, we focused on the calibration slope. Calibration reflects the agreement between observed outcomes and predicted probabilities and the calibration slope assesses whether the predicted risks are of the correct magnitude (Steyerberg & Vergouwe, 2014). A slope of 1 indicates perfect calibration, meaning the model's predicted probabilities are accurately scaled. We calculated the calibration slope using logistic regression within a bootstrap framework, which allowed us to robustly assess the scale of the predicted risks relative to the actual event occurrences. The bootstrap approach, involving resampling the dataset 400 times, provided a more comprehensive understanding of the model's calibration under varying sample conditions.

4 Results

4.1 Model Validation

4.1.1 ROC Curves and AUC

Our time-dependent ROC analysis at 50 days yielded an AUC of approximately 0.738, while at 250 days, the AUC was 0.935. These values indicate that the model's ability to discriminate between those who will experience the event and those who will not improves over time. The ROC curves further visually demonstrate this improvement, with the curve for 250 days being closer to the top left corner, indicating better performance.

4.1.2 C-Index

The average C-index calculated through bootstrapping (n = 400) was 0.746. This suggests that in about 75% of pairwise comparisons, our model correctly ranks the survival times. A C-index of around 0.75 is

generally indicative of good predictive ability, especially in clinical settings where accurate risk stratification is crucial for treatment planning.

4.1.3 Calibration Slope

Our calculated mean calibration slope came to approximately 1.1529. This value, slightly above the ideal of 1, is significant in understanding the model's performance. The calibration slope measures the extent to which the model's predicted risks are proportionate to the observed risks. A value of 1 would indicate perfect calibration, meaning the model's predictions are perfectly aligned with the actual observed risks. Our finding of a calibration slope above 1 suggests that our model may be mildly overfitting the data, predicting slightly higher risks than what is observed.

5 Discussion

The nuanced findings from our analysis provide a comprehensive view of our Cox model's performance. While the model exhibits strong discriminative ability, as indicated by the AUC values and C-index, our calibration assessment, particularly the calibration slope, suggests areas where improvement is needed. Notably, the calibration slope, slightly over the ideal value of 1, implies a mild overestimation in risk predictions. This indicates a complex calibration scenario where the model might be overfitting to some extent.

Such overestimation, although modest, is critical in clinical settings. Accurate risk prediction is vital for informed decision-making and effective patient management. Overestimated risks might lead to more aggressive interventions than necessary, affecting patient care and resource allocation. Conversely, underestimating risks could result in missed opportunities for timely intervention. This highlights the importance of achieving a balance in predictive accuracy, ensuring that the model neither overestimates nor underestimates risks.

The observed improvement in the model's discriminative ability over time, with increasing AUC values from 50 to 250 days, underscores the dynamic nature of risk factors and their evolving impact on patient outcomes. However, the calibration results emphasize the need to focus not just on the model's ability to discriminate but also on the accuracy of its probability predictions.

Future work should, therefore, focus on refining the model's complexity and variable selection. Re-evaluating the model's components and considering alternative modeling approaches might help in aligning the predicted probabilities more closely with actual outcomes. Applying more advanced calibration techniques could also address the observed overfitting, enhancing the model's reliability. Additionally, external validation on an independent dataset is essential to confirm the model's effectiveness and applicability in different clinical contexts. Such efforts will be crucial in enhancing the model's utility and ensuring its robustness in real-world clinical applications, where precise risk assessment directly informs patient care strategies.

6 Conclusions

7 References

1. Ahmad, T., Munir, A., Bhatti, S. H., Aftab, M., & Raza, M. A. (2017). Survival analysis of heart failure patients: A case study. PLOS ONE, 12(7), e0181001. https://doi.org/10.1371/journal.pone.0181001

2. Jones, N., Ak, R., Adoki, I., Fdr, H., & Cj, T. (2019). Survival of patients with chronic heart failure in the community: a systematic review and meta-analysis. European Journal of Heart Failure, 21(11), 1306–1325. https://doi.org/10.1002/ejhf.1594

Steyerberg, E. W., & Vergouwe, Y. (2014). Towards better clinical prediction models: Seven steps for development and an ABCD for validation. European Heart Journal, 35(29), 1925–1931. https://doi.org/10.1093/eurheartj/ehu207

Pavlou, M., Ambler, G., Seaman, S. R., Guttmann, O., Elliott, P., King, M., & Omar, R. Z. (2015). How to develop a more accurate risk prediction model when there are few events. BMJ, h3868. https://doi.org/10.1136/bmj.h3868

Heagerty, P. J., & Zheng, Y. (2005). Survival Model Predictive Accuracy and ROC Curves. Biometrics, 61(1), 92–105. https://doi.org/10.1111/j.0006-341X.2005.030814.x

8 Appendix

8.1 Code

```
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)
library(tidyverse)
library(knitr)
library(kableExtra)
library(survival)
library(survminer)
library(writexl)
library(readxl)
library(table1)
library(rmarkdown)
library(KMsurv)
library(StepReg)
library(ggplot2)
library(timeROC)
library(boot)
library(rms)
library(survivalROC)
data = read_csv("./data/heart_failure.csv")
dat <- data %>%
  arrange(TIME) %>% janitor::clean_names() %>%
  mutate(ejection_fraction_cat = case_when(ejection_fraction <= 30 ~ "Low",
                                           ejection_fraction > 30 & ejection_fraction <= 45 ~ "Medium",
                                           ejection_fraction > 45 ~ "High")) %>%
  mutate(gender = factor(gender),
         smoking = factor(smoking),
         diabetes = factor(diabetes),
         bp = factor(bp),
         event = factor(event),
         anaemia = factor(anaemia),
         ejection_fraction_cat = factor(ejection_fraction_cat, levels = c("Low", "Medium", "High"))) %>%
  rename(platelets = pletelets,
         anemia = anaemia,
         EF = ejection_fraction,
         EF_cat = ejection_fraction_cat)

# Calculate the number of right-censored:
number_censored <- sum(dat$event == 0)
# Calculate the number of event:
number_event <- sum(dat$event == 1)
dat_table = dat
label(dat_table$time) = "Survival time (days)"
label(dat_table$gender) = "Gender"
label(dat_table$smoking) = "Smoking status"
label(dat_table$diabetes) = "Diabetes"
label(dat_table$bp) = "Blood Pressure"
label(dat_table$anemia) = "Anemia"
label(dat_table$age) = "Age (years)"
label(dat_table$EF_cat) = "Ejection Fraction (EF_cat)"
```

```r
label(dat_table$sodium) = "Serum Sodium (mEq/L)"
label(dat_table$creatinine) = "Serum creatinine (mg/dL)"
label(dat_table$platelets) = "Plateletes (mcL)"
label(dat_table$cpk) = "Creatinine phosphokinase (U/L)"
dat_table$event <- factor(dat$event, levels = c(0, 1), labels = c("Censored", "Event"))

pvalue <- function(x, ...) {
  # Remove the "overall" column
    x <- x[names(x) != "overall"]
    # Construct vectors of data y, and groups (strata) g
    y <- unlist(x)
    g <- factor(rep(1:length(x), times = sapply(x, length)))
    if (is.numeric(y)) {
        # For numeric variables, perform a standard 2-sample t-test
        p <- t.test(y ~ g)$p.value
    } else {
        # For categorical variables, perform a chi-squared test of independence
        p <- chisq.test(table(y, g))$p.value
    }
    # Format the p-value, using an HTML entity for the less-than sign.
    # The initial empty string places the output on the line below the variable label.
    c("", sub("<", "<", format.pval(p, digits = 3, eps = 0.001)))
}
caption = "Table 1: Descriptive Statistics Table"

table1 = table1(~ time + age + gender + smoking + diabetes + bp + EF_cat +
                  anemia + sodium + creatinine + cpk + platelets | event,
                  data = dat_table, extra.col = list(`P-value` = pvalue), caption = caption)
t1kable(table1) %>% kable_styling(font_size = 8, latex_options = "HOLD_position")
# Data contains the continuous vars only
cont_dat = dat %>%
  select(age, sodium, creatinine, platelets, cpk)
# Long format
cont_dat.long = cont_dat %>%
  pivot_longer(cols = c(age, sodium, creatinine, platelets, cpk))
# Plot the continuous variable histograms
cont_hist = ggplot(data = cont_dat.long, aes(x = value)) +
  geom_histogram(aes(fill = name), bins = 30) +
  facet_wrap(~name, scales = "free") +
  labs(x = "Value", y = "Count", title = "Figure 1.1: Histograms of Continuous Variables") +
  theme_bw() +
  theme(legend.position = "none")
cont_hist

# Data contains the categorical vars only
cate_data = dat %>%
  select(event, gender, smoking, diabetes, bp, anemia, EF_cat)
# Long format
cate_dat.long = cate_data %>%
  pivot_longer(cols = c(event, gender, smoking, diabetes, bp, anemia, EF_cat))
# Plot the categorical variable barplots
cate_barplot = ggplot(cate_dat.long, aes(x = value, fill = value)) +
  geom_bar() +
```

```r
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.3) +
  facet_wrap(~name, scales = "free") +
  labs(x = "Category", y = "Count", fill = "Category", title = "Figure 1.2: Bar Charts of Categorical Va
  theme_bw() +
  theme(legend.position = "none") +
  ylim(0, 240)
cate_barplot
dat_log = dat %>%
  mutate(cpk_log = log(cpk + 1),
         creatinine_log = log(creatinine + 1))
# # Calculate predicted risks
# predicted_risks <- predict(step_model, newdata = model_data, type = "risk")
#
# # Time points for ROC analysis
# time_points <- c(50, 250)
#
# # Calculate ROC curves at specified times
# roc_50 <- timeROC(T = model_data$time, delta = model_data$event, marker = predicted_risks, times = 50
# roc_250 <- timeROC(T = model_data$time, delta = model_data$event, marker = predicted_risks, times = 2
#
# # Extract AUC values
# auc_50 <- roc_50$AUC
# auc_250 <- roc_250$AUC
#
# # Print AUC values
# print(paste("AUC at 50 days:", auc_50))
# print(paste("AUC at 250 days:", auc_250))
#
# # Plot ROC curves
# plot(roc_50$FP, roc_50$TP, type = "l", col = "red", xlab = "1 - Specificity", ylab = "Sensitivity", m
# lines(roc_250$FP, roc_250$TP, type = "l", col = "blue")
# legend("bottomright", legend = c("50 days", "250 days"), col = c("red", "blue"), lty = 1)
# abline(0, 1, col = "black", lty = 2)
# Define the function for calculating C-statistic
# boot_c_statistic <- function(original_data, indices) {
#   # Creating a bootstrap sample
#   boot_data <- original_data[indices, ]
#
#   # Fit the Cox model to the bootstrap sample
#   fit <-   coxph(Surv(time, event) ~ log(creatinine+1) + age + ef_cat + bp +
#                        sodium, data = boot_data)
#
#   # Calculate the concordance statistic using the updated function
#   concordance <- concordance(fit)$concordance
#   return(concordance)
# }
#
# # Perform bootstrapping for C-statistic
# set.seed(123) # for reproducibility
# boot_results_c_stat <- boot(data = model_data, statistic = boot_c_statistic, R = 400)
#
# # Calculate the average C-statistic
# mean_c_stat <- mean(boot_results_c_stat$t)
```

```
# print(mean_c_stat)
# # Define the bootstrap function for calibration metrics using logistic regression
# boot_calibration_logistic <- function(original_data, indices) {
#   boot_data <- original_data[indices, ]
#   fit <-  coxph(Surv(time, event) ~ log(creatinine+1) + age + ef_cat + bp +
#                       sodium, data = boot_data)
#
#
#   # Predicted risks for the original dataset
#   predicted_risks <- predict(fit, newdata = original_data, type = "risk")
#
#   # Fit a logistic model for calibration
#   calibration_model_logistic <- glm(event ~ predicted_risks, data = original_data, family = "binomial
#
#   # Calibration slope (coefficient of predicted_risks)
#   calibration_slope_logistic <- coef(calibration_model_logistic)["predicted_risks"]
#
#   return(calibration_slope_logistic)
# }
#
# # Perform bootstrap
# set.seed(123)
# boot_results_logistic <- boot(data = model_data, statistic = boot_calibration_logistic, R = 400)
#
# # Calculate the average calibration slope
# mean_calibration_slope_logistic <- mean(boot_results_logistic$t)
# print(mean_calibration_slope_logistic)
```