Factors Influencing the Risk of Liver Cirrhosis: A Statistical Analysis Using Logistic Regression and Cox Proportional Hazards Models

Giorgos Tzimas

March 2025

Abstract

Liver cirrhosis is a progressive condition that is characterized by fibrosis and liver dysfunction, which could result in severe complications and mortality if left untreated. This study uses data from a publicly available RCT dataset to investigate the clinical and biochemical predictors of liver cirrhosis and associated mortality. We apply two statistical models for this analysis: logistic regression and the Cox proportional hazards model. These models were fitted to identify significant risk factors that play a role in death due to liver cirrhosis. Predictors were selected using stepwise regression for both models, and missing values were imputed using the missForest algorithm. Key predictors identified include bilirubin, albumin, prothrombin, and age. The findings are consistent with previously established knowledge about risk factors that determine death due to liver cirrhosis and further reinforce the clinical value of liver function markers.

1 Introduction

Liver cirrhosis is a disease characterized by scarring of the liver, which leads to impairing its function over time (Mayo Clinic,). It results from chronic liver damage brought on from diseases and conditions such as alcohol use, hepatitis B, C, and D, cystic fibrosis, among other causes. In its advanced stages, liver cirrhosis can result to serious health complications such as high blood pressure, swelling of the legs, belly, and spleen, hemorrhagic events, among others (Mayo Clinic,). There are various risk factors that can contribute towards the progression of liver cirrhosis, such as demographic, biochemical markers, and clinical history. The goal of this study is to identify some of these factors for the purposes of early diagnosis and prevention of liver cirrhosis in a clinical setting and to allow for more targeted and personalized treatment approaches based on patient characteristics.

2 Methods

2.1 Data Source

The dataset used for this study is the *Cirrhosis Patient Survival Prediction* dataset from Kaggle. The original source of the dataset was from a Mayo Clinic randomized, placebo-controlled trial that tested the effectiveness of the drug D-penicillamine on primary biliary cirrhosis (PBC) of the liver carried out from 1974 to 1984. Out of the 424 PBC patients that qualified for the trial, 312 patients took part in it and have comprehensive data available. We have a total of 16 predictor variables in the dataset which provide patient demographic, biochemical, and clinical characteristics, a time-to-event and status variable for the outcome of the patient, as we as the treatment arm, totaling 19 variables for this analysis.

2.2 Data Preprocessing

The first preprocessing step was to remove any records that were censored due to liver transplantation, leaving us with only the two outcomes of censoring and death and a total of 393 patients. All categorical variables were converted into factors and the outcome was redefined with 0 for censored and 1 for death. A subset of the predictors contained missing values for some of the records (Table 1). Instead of removing those records, the *missForest* R package was used to fill in the missing values. missForest is a machine learning random forest-based method that imputes missing data using a multiple imputation approach. This approach resulted in an out-of-bag error rate of 0.01, which suggests a high imputation accuracy.

Table 1: Summary of missing values in the dataset by count and percentage.

Feature	Missing Count (%)
Tryglicerides	29 (9.9%)
Cholesterol	27 (9.2%)
Platelets	4 (1.4%)
Copper	2 (0.7%)

2.3 Exploratory Data Analysis

The next step was to perform exploratory data analysis to visualize the relationships among the predictors in the dataset. A correlation heatmap was generated to examine pairwise associations between the predictors. Different statistical methods were utilized based on the data type of the variables that were being compared (Table 6. Variables such as bilirubin, copper and prothrombin showed moderate positive correlations with the outcome variable. Features were further analyzed in univariate and multivariate plots to identify any other interesting relationships among the variables. Figures and plots are available in Appendix B.

2.4 Statistical Models

This analysis used two primary models to assess what factors influence the risk of liver cirrhosis: Logistic regression and Cox proportional hazards model. These models were chosen because the outcome variable was binary, with the additional benefit of modeling time-to-event outcomes offered by the Cox model for comparison.

2.4.1 Logistic Regression

The logistic regression model was built using all of the baseline characteristics (Table 5) available

in the dataset. Logistic regression estimates the probability of an event given a set of predictor variables using the logit function:

$$logit(P) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \quad (1)$$

In order to determine the most parsimonious model, a stepwise procedure was applied to determine the most relevant predictor variables and optimize the model fit. The final predictions of the model come in the form of probabilities ranging from 0 to 1 that indicate the probability that a patient has liver cirrhosis.

2.4.2 Cox Proportional Hazards

The Cox proportional hazards model was used to assess the impact of the predictors on on survival time for liver cirrhosis. The Cox model is a semi-parametric model that estimates the hazard function of an event occuring at a specific time, given a set of baseline characteristics. The Cox model utilizes the hazard function for its predictions:

$$h(t) = h_0(t)e^{\beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n} \quad (2)$$

3 Results

3.1 Treatment Arm and Survival

A Kaplan-Meier analysis was performed to compare survival rates between the treatment arm (D-penicillamine) and placebo. The results showed that there were no statistically significant differences in survival between the two treatment arms (log rank p = 0.79). This result was also corroborated by the subsequent Cox proportional hazards model, where the treatment arm did not appear as a significant variable during the predictor selection process. These results suggest that D-penicillamine had no significant improvement in survival rates for this patient cohort.

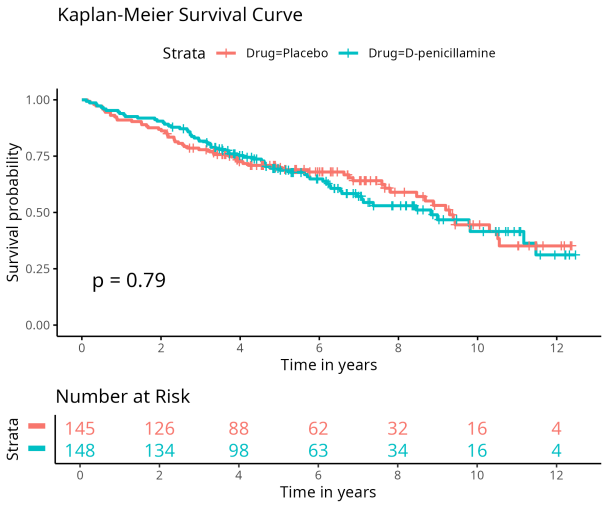

Figure 1: Kaplan-Meier survival curve stratified by treatment arm.

3.2 Logistic Regression Analysis

The final logistic regression model fitted using the stepwise selection process is detailed in Table 2. The model achieved an area under the receiving operating characteristic score of 0.89, A total of 8 predictors were identified as statistically significant for predicting liver cirrhosis. These are the predictors that are strongly associated with death due to liver cirrhosis:

Table 2: Logistic regression coefficients for predicting liver cirrhosis.

Variable	Estimate	Std. Error	Pr(>|z|)
(Intercept)	-12.00	2.202	5.12e-08
AscitesYes	2.282	1.084	0.035367
HepatomegalyYes	0.7161	0.3265	0.028306
Bilirubin	0.1884	0.08011	0.018666
Copper	0.006227	0.002545	0.014419
Alk_Phos	0.0001776	0.00006904	0.010093
SGOT	0.006729	0.003105	0.030222
Prothrombin	0.6153	0.1859	0.000936
Age_years	0.04561	0.01618	0.004804

- The presence of ascites in the peritoneal cavity increases the odds by 9.796.
- Having an enlarged liver (Hepatomegaly) increases the odds by 2.046.
- A 1 mg/dL increase in bilirubin increases the odds by 1.207.
- Each additional µg of copper present in the urine increases the odds by 1.006.
- Each additional U/L of alkaline phosphatase in the blood increases the odds by 1.000.
- Each additional U/L of the SGOT enzyme increases the odds by 1.007.
- Each additional second it takes for blood to clot (Prothrombin time) increases the odds by 1.850.
- Each additional year of age increases the odds by 1.047.

In general, the logistic regression model was able to identify several significant predictors that are helpful in determining death due to liver cirrhosis. The residual deviance of the model (250.03, df=284) indicates a good fit relative to the null deviance (399.85). The AIC of 268.03 supports model parsimony. Residual diagnostics plots show a few influential observations that are present but no major violations of assumptions.

3.3 Cox Regression Analysis

The Cox model was also fitted using a stepwise selection process based on AIC. The model was able to effectively discriminate between patients who experienced events and those who did not with a concordance index was 0.847. All metrics of global significance (Wald, likelihood, logrank) had p-values below 0.05. The Schoenfeld residuals also indicated that the assumption of proportional hazards was met (p = 0.0256), although some of the variables showed stronger individual contributions to the model (Table 4). The final model included a total of eight predictors and eleven model coefficients (multilevel variables):

Table 3: Cox proportional hazards model coefficients for predictors of survival.

Variable	coef	exp(coef)	se(coef)	Pr(>|z|)
EdemaS	0.1165063	1.1235645	0.2875336	0.68534
EdemaY	0.8717875	2.3911814	0.3147658	0.00561
Bilirubin	0.0830488	1.0865948	0.0190743	0.0000134
Albumin	-0.7846299	0.4562885	0.2590880	0.00246
Copper	0.0028880	1.0028922	0.0009353	0.00202
SGOT	0.0041349	1.0041435	0.0016680	0.01318
Prothrombin	0.3068929	1.3591954	0.1039253	0.00315
Stage.L	1.5904523	4.9059674	0.6979376	0.02268
Stage.Q	-0.6360539	0.5293773	0.5525564	0.24969
Stage.C	0.3284702	1.3888418	0.3110739	0.29100
Age_years	0.0289300	1.0293525	0.0095610	0.00248

- Presence of edema without the use of diuretics was associated with an increased hazard of 1.124.
- Presence of edema despite the use of diuretics was associated with an increased hazard of 2.391.

- A 1 mg/dL increase in bilirubin increases the hazard of death by 1.087.
- Each additional g/dL decrease in albumin was associated with an increased hazard of 0.456.
- Each additional µg of copper present in the urine was associated with an increased hazard of 1.003.
- Each additional U/L of the SGOT enzyme was associated with an increased hazard of 1.004.
- Each additional second it takes for blood to clot (Prothrombin time) was associated with an increased hazard of 1.359.
- Patients in stage L had a significantly higher hazard compared to the reference group (HR = 4.906).
- Each additional year of age was associated with an increased hazard of 1.029.

Table 4: Tests of the proportional hazards assumption using Schoenfeld residuals.

Variable	**Chisq**	**df**	**p**
Edema	4.956	2	0.0839
Bilirubin	9.654	1	0.0019
Albumin	0.653	1	0.4192
Copper	0.142	1	0.7062
SGOT	3.257	1	0.0711
Prothrombin	3.888	1	0.0486
Stage	4.808	3	0.1864
Age_years	0.497	1	0.4809
GLOBAL	21.848	11	0.0256

In summary, the Cox model was able to identify several significant predictors (most of them similar to the logistic regression model) for determining patient survival. These results are as expected from a clinical point of view regarding liver function and patient prognosis.

4 Discussion

The findings of this analysis showed evidence that several factors and biochemical markers play a significant role for predicting the outcome of death due to liver cirrhosis. For example, elevated levels of bilirubin is a known indicator of impaired liver function (Guerra Ruiz et al.,), and low albumin levels are a common indicator of liver failure (Spinella, Sawhney, & Jalan,). Our findings align with existing clinical evidence and knowledge as to what influences liver cirrhosis outcomes. Other demographic factors such as age also play a role, with older individuals being at a greater risk. However, treatment assignment was not determined to be statistically significant in predicting the outcome, as it did not show a meaningful association with survival risk.

The results align with existing clinical evidence that liver function markers are essential indicators of disease progression. Other emographic factors such as age also influence cirrhosis outcomes. Older individuals tend to have more advanced liver disease, presumably due to prolonged exposure to risk factors, while males generally have a higher risk of developing cirrhosis due to differences in alcohol consumption and metabolic responses.

5 Limitations

One of the limitations of this analysis was the relatively small sample size (n=293), which

could limit the generalizability of the findings. Although RCTs typically provide high-quality data, they are often difficult to access due to privacy concerns, cost, and logistical constraints. Further research into the role of D-penicillamine for the treatment of liver cirrhosis could help determine its therapeutic efficacy, safety profile, and potential benefits across different stages of disease progression.

6 Conclusion

This analysis utilized two primary statistical modeling approaches to itendify the factors that influence the outcome of death due to liver cirrhosis. The results highlight bilirubin, albumin, age, and sex as significant predictors. Future research should aim to investigate the effects of D-penicillamine on liver cirrhosis using a larger and more diverse patient cohort, along with an extended follow-up period. This would allow for a more accurate assessment of the risk profile of D-penicillamine to determine whether it offers clinically significant advantages.

References

Guerra Ruiz, A. R., Crespo, J., López Martínez, R. M., Iruzubieta, P., Casals Mercadal, G., Lalana Garcés, M., ... Morales Ruiz, M. (2021). Measurement and clinical usefulness of bilirubin in liver disease. *Advances in Laboratory Medicine/Avances en Medicina de Laboratorio*, *2*(3), 352–361.

Mayo Clinic. (2025). *Cirrhosis symptoms and causes.* Retrieved from `https://www.mayoclinic.org/diseases-conditions/cirrhosis/symptoms-causes/syc-20351487` (Accessed: 2025-03-11)

Spinella, R., Sawhney, R., Jalan, R. (2016). Albumin in chronic liver disease: structure, functions and therapeutic implications. *Hepatology international*, *10*, 124–132.

Appendix A: Summary Table of Baseline Variables

Table 5: Summary of patient baseline characteristics, including demographic, clinical, and lifestyle factors, stratified by treatment.

	level	Placebo	D-penicillamine	p
n		145	148	
Bilirubin (mean (SD))		3.69 (5.41)	2.85 (3.73)	0.124
Cholesterol (mean (SD))		376.42 (259.92)	352.77 (178.17)	0.389
Albumin (mean (SD))		3.52 (0.40)	3.51 (0.45)	0.926
Copper (mean (SD))		96.33 (80.19)	95.53 (88.79)	0.936
Alk_Phos (mean (SD))		1988.90 (2155.66)	2033.98 (2241.83)	0.861
SGOT (mean (SD))		125.13 (60.06)	119.06 (55.45)	0.369
Tryglicerides (mean (SD))		123.42 (56.65)	124.75 (72.58)	0.868
Platelets (mean (SD))		263.65 (91.18)	255.45 (101.00)	0.470
Prothrombin (mean (SD))		10.83 (1.16)	10.67 (0.87)	0.196
Age_years (mean (SD))		49.01 (10.03)	52.14 (10.87)	0.011
N_Years (mean (SD))		5.52 (3.22)	5.64 (3.02)	0.741
Sex (%)	Female	130 (89.7)	130 (87.8)	0.759
	Male	15 (10.3)	18 (12.2)	
Ascites (%)	No	135 (93.1)	134 (90.5)	0.557
	Yes	10 (6.9)	14 (9.5)	
Hepatomegaly (%)	No	66 (45.5)	79 (53.4)	0.219
	Yes	79 (54.5)	69 (46.6)	

Table 5: Summary of patient baseline characteristics, including demographic, clinical, and lifestyle factors, stratified by treatment. *(continued)*

	level	Placebo	D-penicillamine	p
Spiders (%)	No	104 (71.7)	104 (70.3)	0.884
	Yes	41 (28.3)	44 (29.7)	
Edema (%)	N	124 (85.5)	122 (82.4)	0.634
	S	11 (7.6)	16 (10.8)	
	Y	10 (6.9)	10 (6.8)	
Stage (%)	1	4 (2.8)	12 (8.1)	0.163
	2	30 (20.7)	34 (23.0)	
	3	61 (42.1)	51 (34.5)	
	4	50 (34.5)	51 (34.5)	
Status (%)	0	85 (58.6)	83 (56.1)	0.748
	1	60 (41.4)	65 (43.9)	

Table 6: Summary of different correlation metrics used based on data type.

Variable Type 1	**Variable Type 2**	**Correlation Method**	**Description**
Numeric	Numeric	Pearson	Linear correlation between two continuous variables
Numeric	Binary	Point-Biserial	Pearson correlation adapted for numeric and binary variables
Binary	Binary	Phi Coefficient	Pearson correlation applied to two binary variables via 2x2 table
Categorical	Categorical	Cramér's V	Association strength based on chi-squared statistic
Binary	Categorical	Cramér's V	Treats binary as categorical; uses chi-squared-based association
Numeric	Categorical	Eta-squared (^{2})	Variance in numeric variable explained by categories (from ANOVA)

Appendix B: EDA Plots

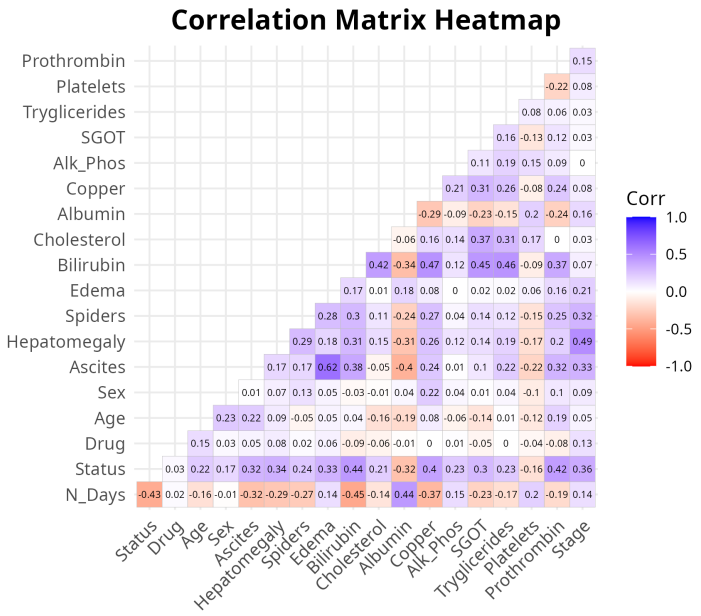

Figure 2: Heatmap showing pairwise correlations among clinical and demographic variables.

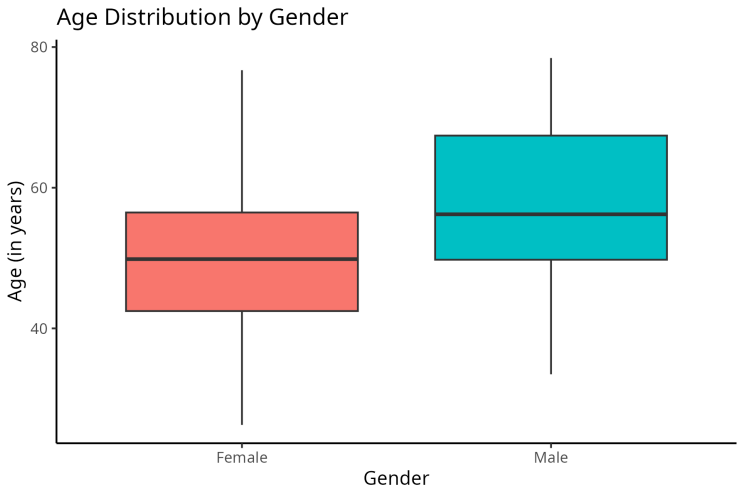

Figure 3: Distribution of patient ages by gender.

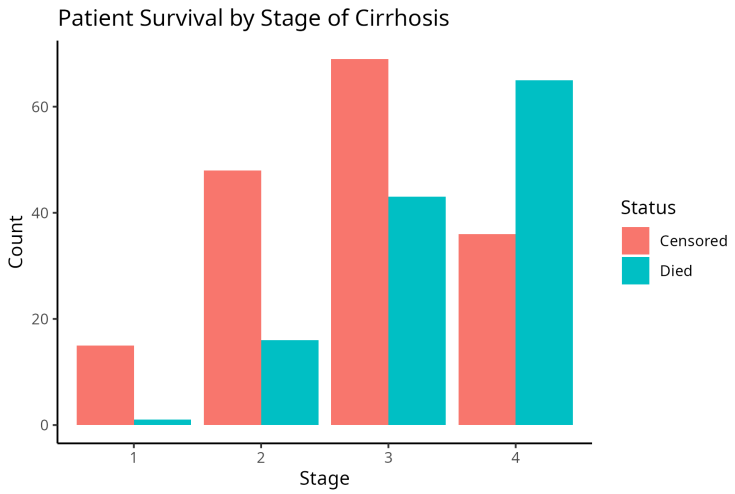

Figure 4: Frequency of censoring and death by stage of liver cirrhosis.

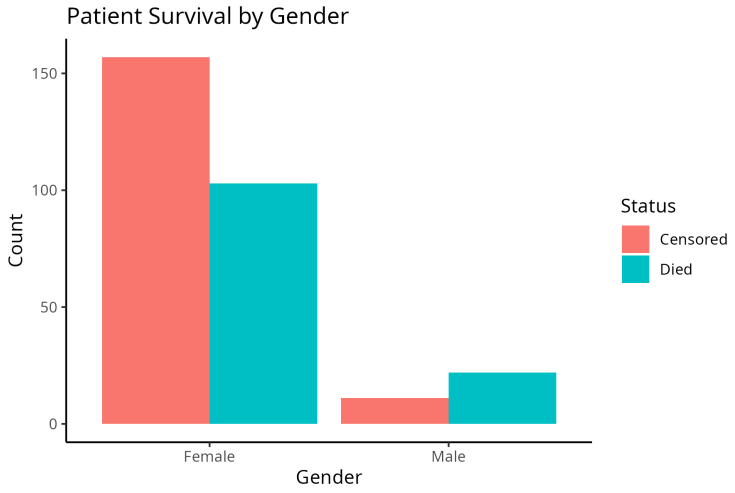

Figure 5: Frequency of censoring and death by gender.

Appendix C: R Code

```
## ----setup,
    include=FALSE-------------------------------------------------------------------------------
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```r
library(ggplot2)
library(tableone)
library(kableExtra)
library(clipr)
library(missForest)
library(ggcorrplot)
library(DescTools)
library(survival)
library(survminer)
library(caret)
library(MASS)
library(rsample)
library(pROC)
library(ResourceSelection)
library(car)
# library(timeROC)

source('mixed_corr_matrix.R')


##
    ------------------------------------------------------------------------------------------------------
show_colors <- function() {
  scales::show_col(scales::hue_pal()(20))
  print(scales::hue_pal()(20))
}


##
    ------------------------------------------------------------------------------------------------------
df <- read.csv('cirrhosis.csv')
df <- df %>% filter(Status %in% c('C', 'D'))
df <- df %>%
  mutate(
    Status = factor(
      case_when(
        Status == 'C' ~ 0,
        Status == 'D' ~ 1
      ), levels=c(0, 1)
    ),
    Drug = factor(Drug, levels=c('Placebo', 'D-penicillamine')),
    Sex = factor(
      case_when(
        Sex == 'F' ~ 'Female',
        Sex == 'M' ~ 'Male'
      ), levels=c('Female', 'Male'),
    ),
    Ascites = factor(
      case_when(
```

```
        Ascites == 'N' ~ 'No',
        Ascites == 'Y' ~ 'Yes',
      ), levels=c('No', 'Yes')
    ),
    Hepatomegaly = factor(
      case_when(
        Hepatomegaly == 'N' ~ 'No',
        Hepatomegaly == 'Y' ~ 'Yes',
      ), levels=c('No', 'Yes')
    ),
    Spiders = factor(
      case_when(
        Spiders == 'N' ~ 'No',
        Spiders == 'Y' ~ 'Yes',
      ), levels=c('No', 'Yes')
    ),
    Edema = factor(Edema, levels=c('N', 'S', 'Y')),
    Stage = factor(Stage, levels=c(1, 2, 3, 4), ordered=T),

    Age_years = Age / 365.25,
    N_Years = N_Days / 365.25
  ) %>%
  filter(!is.na(Drug)) %>%
  dplyr::select((-c(Age, N_Days)))
head(df)


##
    ------------------------------------------------------------------------------------------------------------
categorical_vars <- names(df)[sapply(df, is.factor)][-c(1,2)] # Remove Drug stratification variable
numerical_vars <-names(df)[sapply(df, is.numeric)][-1] # Remove ID variable
# stopifnot(length(categorical_vars) + length(numerical_vars) == ncol(df)-2)

table1 <- CreateTableOne(
  vars = c(numerical_vars, categorical_vars, c('Status')),
  strata = 'Drug',
  data=df,
  factorVars=categorical_vars
)


##
    ------------------------------------------------------------------------------------------------------------
p <- print(table1, printToggle = F, noSpaces = T, showAllLevels = T, missing=F)
# Run in console mode
kbl(p, booktabs = T, longtable=T, format = "latex", caption="Summary of patient baseline
    characteristics, including demographic, clinical, and lifestyle factors, stratified by
    treatment.", label="tableone") %>%
  kable_styling(latex_options = c('striped', 'repeat_header')) %>% write_clip()
```

```
##
    ------------------------------------------------------------------------------------------------------------
missing_counts <- df %>%
  summarise(across(everything(), ~sum(is.na(.)))) %>%
  pivot_longer(cols=everything(), names_to='Feature', values_to='Missing_Count') %>%
  mutate(Missing_Perc = (Missing_Count / nrow(df)) * 100) %>%
  mutate(Missing_Info = paste0(Missing_Count, ' (', round(Missing_Perc, 1), '%)')) %>%
  arrange(desc(Missing_Count)) %>%
  filter(Missing_Count > 0) %>%
  dplyr::select(Feature, Missing_Info)

missing_counts


##
    ------------------------------------------------------------------------------------------------------------
miss_forest <- missForest(df)
print(miss_forest$OOBerror)


##
    ------------------------------------------------------------------------------------------------------------
df_imputed <- miss_forest$ximp
head(df_imputed)


##
    ------------------------------------------------------------------------------------------------------------
p <- df_imputed %>%
  mutate(Status = dplyr::recode(Status, `0` = 'Censored', `1` = 'Died')) %>%
  ggplot(aes(x=Sex, fill=Status)) +
  geom_bar(position='dodge') +
  labs(
    title = 'Patient Survival by Gender',
    x = 'Gender',
    y = 'Count',
    fill = 'Status'
  ) +
  theme_classic()
ggsave('status_count_by_gender.png', plot=p, width=6, height=4, dpi=300)
print(p)


##
    ------------------------------------------------------------------------------------------------------------
p <- df_imputed %>%
  mutate(Status = dplyr::recode(Status, `0` = 'Censored', `1` = 'Died')) %>%
```

```
  ggplot(aes(x=Drug, fill=Status)) +
  geom_bar(position='dodge') +
  labs(
    title = 'Patient Survival by Treatment Arm',
    x = 'Treatment',
    y = 'Count',
    fill = 'Status'
  ) +
  theme_classic()
ggsave('status_count_by_treatment.png', plot=p, width=6, height=4, dpi=300)
print(p)

##
    ------------------------------------------------------------------------------------------------------
p <- df_imputed %>%
  mutate(Status = dplyr::recode(Status, `0` = 'Censored', `1` = 'Died')) %>%
  ggplot(aes(x=Stage, fill=Status)) +
  geom_bar(position='dodge') +
  labs(
    title = 'Patient Survival by Stage of Cirrhosis',
    x = 'Stage',
    y = 'Count',
    fill = 'Status'
  ) +
  theme_classic()
ggsave('status_count_by_stage.png', plot=p, width=6, height=4, dpi=300)
print(p)


##
    ------------------------------------------------------------------------------------------------------
p <- df_imputed %>%
  # mutate(Status = recode(Status, `0` = 'Censored', `1` = 'Died')) %>%
  ggplot(aes(x=Sex, y=Age_years, fill=Sex)) +
  geom_boxplot() +
  labs(
    title = 'Age Distribution by Gender',
    x = 'Gender',
    y = 'Age (in years)',
    fill = 'Gender'
  ) +
  # coord_flip() +
  theme_classic() +
  theme(legend.position='none')
ggsave('age_boxplot_by_gender.png', plot=p, width=6, height=4, dpi=300)
print(p)


##
```

```
    ------------------------------------------------------------------------------------------------------------
corr_matrix <- compute_mixed_corr(df_imputed %>% dplyr::select(-c(ID, Age_years, N_Years)))
corr_matrix


##
    ------------------------------------------------------------------------------------------------------------
p <- ggcorrplot(corr_matrix,
           method = "square",    # Boxed shape for grid
           type = "lower",       # Show only lower triangle
           lab = TRUE,           # Show correlation values inside the boxes
           lab_size = 2,         # Adjust text size
           digits = 2,           # Round correlation values
           tl.cex = 10,          # Increase axis label size
           tl.srt = 45,          # Rotate axis labels for better readability
           colors = c("red", "white", "blue")) +  # Red (-1), White (0), Blue (+1)

  ggtitle("Correlation Matrix Heatmap") +  # Add title
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),  # Center title
    panel.background = element_rect(fill = "white", color = NA),  # Fix transparent issue
    plot.background = element_rect(fill = "white", color = NA)  # Ensure full white background
  )

ggsave('corr_matrix.png', plot=p, width=6, height=5, dpi=300)
print(p)

##
    ------------------------------------------------------------------------------------------------------------
surv_object <- Surv(time=tst$N_Years, event=as.numeric(tst$Status))
km_fit <- survfit(surv_object ~ Drug, data=df_imputed)
logrank_test <- survdiff(surv_object ~ Drug, data=df_imputed)
print(logrank_test)


##
    ------------------------------------------------------------------------------------------------------------
grid.draw.ggsurvplot <- function(x){
  survminer:::print.ggsurvplot(x, newpage = FALSE)
}

p <- ggsurvplot(
  km_fit,
  data = df_imputed,
  pval = TRUE,        # Show Log-Rank p-value
  conf.int = F,    # Show confidence intervals
  risk.table = TRUE,  # Show risk table
  risk.table.y.text.col = TRUE,   # Color risk table text
  risk.table.y.text = FALSE,      # Remove vertical risk table labels
```

```
  risk.table.height = 0.25,       # Adjust risk table size
  risk.table.col = "strata",      # Color risk table by group
  break.time.by = 2,            # Change time intervals (e.g., every 500 days)
  xlab = "Time in years",          # Customize x-axis label
  title = "Kaplan-Meier Survival Curve",  # Add title
  risk.table.title = "Number at Risk",    # Risk table title
  ggtheme = theme_classic()
)

ggsave('km_plot.png', plot=p, width=6, height=5, dpi=300)
print(p)

##
    ------------------------------------------------------------------------------------------------------
set.seed(123)
split <- initial_split(df_imputed, prop=0.5, strata=Status)
train_data <- training(split)
test_data <- testing(split)

cat('dim(train_data):', dim(train_data), '\n')
cat('dim(test_data):', dim(test_data), '\n')


##
    ------------------------------------------------------------------------------------------------------
response_var <- 'Status'
time_var <- 'N_Years'
predictors <- df_imputed %>% dplyr::select(-c(ID, N_Years, Drug, Status)) %>% colnames()
reg_formula <- as.formula(paste(response_var, '~', paste(predictors, collapse='+')))
print(logreg_formula)


##
    ------------------------------------------------------------------------------------------------------
logit_model <- glm(reg_formula, data=train_data, family=binomial(link='logit'))
summary(logit_model)

##
    ------------------------------------------------------------------------------------------------------
logit_train_preds <- predict(logit_model, train_data, type='response')
logit_test_preds <- predict(logit_model, test_data, type='response')

cat('Train AUC:', auc(roc(train_data$Status, logit_train_preds, levels=c(0,1), direction='<')),
    '\n')
cat('Test AUC:', auc(roc(test_data$Status, logit_test_preds, levels=c(0,1), direction='<')), '\n')


##
    ------------------------------------------------------------------------------------------------------
```

```
stepwise_model <- step(glm(reg_formula, data=train_data, family=binomial(link='logit')),
    direction='both')
summary(stepwise_model)


##
    ------------------------------------------------------------------------------------------------------------
stepwise_train_preds <- predict(stepwise_model, train_data, type='response')
stepwise_test_preds <- predict(stepwise_model, test_data, type='response')

cat('Train AUC:', auc(roc(train_data$Status, stepwise_train_preds, levels=c(0,1), direction='<')),
    '\n')
cat('Test AUC:', auc(roc(test_data$Status, stepwise_test_preds, levels=c(0,1), direction='<')),
    '\n')


##
    ------------------------------------------------------------------------------------------------------------
calibration_data <- data.frame(Status=as.numeric(as.character(test_data$Status)),
    RiskScore=stepwise_test_preds) %>%
  mutate(calib_bin = ntile(RiskScore, 10)) %>%
  group_by(calib_bin) %>%
  summarise(
    mean_predicted = mean(RiskScore),
    observed_rate = mean(Status),
    abs_deviation = abs(mean_predicted - observed_rate)
  )

calibration_data


##
    ------------------------------------------------------------------------------------------------------------
p <- ggplot(calibration_data, aes(x = mean_predicted, y = observed_rate, label = calib_bin)) +
  geom_point(size = 3, color = 'black', fill = 'blue', shape = 21, stroke = 2) +  # Blue dots
  geom_text(vjust = -0.8, size = 4, color = 'black') +  # Label points with bin numbers
  xlim(0, max(max(calibration_data$mean_predicted), max(calibration_data$observed_rate))) +
  ylim(0, max(max(calibration_data$mean_predicted), max(calibration_data$observed_rate))) +
  geom_abline(slope = 1, intercept = 0, linetype = 'dashed', color = 'red', linewidth = 1) +  #
    Reference line
  labs(
    title = 'Model Calibration Plot',
    x = 'Predicted Probability',
    y = 'Observed Event Rate'
  ) +
  theme_classic() +
  theme(
    axis.title.x = element_text(size = 16),
    axis.title.y = element_text(size = 16),
```

```
    axis.text.x = element_text(size = 12),
    axis.text.y = element_text(size = 12)
  )

ggsave('calibration_plot.png', plot=p, width=6, height=5, dpi=300)
print(p)

##
    ------------------------------------------------------------------------------------------------------------
hl_test <- hoslem.test(as.numeric(as.character(test_data$Status)), logit_test_preds, g=10)
print(hl_test)


##
    ------------------------------------------------------------------------------------------------------------
cox_formula <- as.formula(paste0('Surv(', time_var, ', ', response_var, ') ~ ', paste(predictors,
    collapse='+')))
print(cox_formula)


##
    ------------------------------------------------------------------------------------------------------------
train_data_cox <- train_data %>% mutate(Status = as.numeric(as.character(Status)))
test_data_cox <- test_data %>% mutate(Status = as.numeric(as.character(Status)))
cox_model <- coxph(cox_formula, data=train_data_cox)
summary(cox_model)


##
    ------------------------------------------------------------------------------------------------------------
cox_train_preds <- predict(cox_model, train_data_cox, type='lp')
cox_test_preds <- predict(cox_model, test_data_cox, type='lp')

cox_train_preds <- predict(cox_model, train_data_cox, type='risk')
cox_test_preds <- predict(cox_model, test_data_cox, type='risk')


roc_train <- timeROC(
  T = train_data_cox$N_Years,
  delta = train_data_cox$Status,
  marker = cox_train_preds,
  cause = 1,
  times = c()

)
print(roc_train$AUC)
hist(predict(cox_model, train_data_cox, type='risk'))
```

```
##
    ------------------------------------------------------------------------------------------------------------
calibration_data <- data.frame(Status=as.numeric(as.character(test_data$Status)),
    RiskScore=cox_test_preds) %>%
  mutate(calib_bin = ntile(RiskScore, 10)) %>%
  group_by(calib_bin) %>%
  summarise(
    mean_predicted = mean(RiskScore),
    observed_rate = mean(Status),
    abs_deviation = abs(mean_predicted - observed_rate)
  )

calibration_data


##
    ------------------------------------------------------------------------------------------------------------
p <- ggplot(calibration_data, aes(x = mean_predicted, y = observed_rate, label = calib_bin)) +
  geom_point(size = 3, color = 'black', fill = 'blue', shape = 21, stroke = 2) +  # Blue dots
  geom_text(vjust = -0.8, size = 4, color = 'black') +  # Label points with bin numbers
  xlim(0, max(max(calibration_data$mean_predicted), max(calibration_data$observed_rate))) +
  ylim(0, max(max(calibration_data$mean_predicted), max(calibration_data$observed_rate))) +
  geom_abline(slope = 1, intercept = 0, linetype = 'dashed', color = 'red', linewidth = 1) +  #
    Reference line
  labs(
    title = 'Model Calibration Plot',
    x = 'Predicted Probability',
    y = 'Observed Event Rate'
  ) +
  theme_classic() +
  theme(
    axis.title.x = element_text(size = 16),
    axis.title.y = element_text(size = 16),
    axis.text.x = element_text(size = 12),
    axis.text.y = element_text(size = 12)
  )

ggsave('calibration_plot_probit.png', plot=p, width=6, height=5, dpi=300)
print(p)


##
    ------------------------------------------------------------------------------------------------------------
hl_test <- hoslem.test(as.numeric(as.character(test_data$Status)), probit_test_preds, g=10)
print(hl_test)


##
    ------------------------------------------------------------------------------------------------------------
```

```
time_var <- 'N_Years'
event_var <- 'Status'
treatment_var <- 'Drug'
risk_group_var <- 'RiskGroup'
n_groups <- 3

results_data <- data.frame(ID=test_data$ID, Drug=test_data$Drug, N_Years=test_data$N_Years,
    Status=as.numeric(as.character(test_data$Status)), RiskScore=cox_test_preds) %>%
  mutate(RiskGroup = as.factor(ntile(RiskScore, n_groups)))
head(results_data)


##
    ------------------------------------------------------------------------------------------------------
fit_treatment_riskgroup_interacted <- survfit(Surv(N_Years, Status) ~ Drug + RiskGroup,
    data=results_data)
print(fit_treatment_riskgroup_interacted)


##
    ------------------------------------------------------------------------------------------------------
# Define survival formulas
model_wo_covariates <- coxph(Surv(N_Years, Status) ~ 1, data = results_data, ties = 'breslow')
model_w_covariates <- coxph(Surv(N_Years, Status) ~ Drug * RiskGroup, data = results_data, ties =
    'breslow')


# Model comparison using log-likelihood, AIC, and BIC
tibble(
  Criterion = c('-2 Log L', 'AIC', 'SBC'),
  `Without Covariates` = c(
    -2 * as.numeric(logLik(model_wo_covariates)),
    AIC(model_wo_covariates),
    BIC(model_wo_covariates)
  ),
  `With Covariates` = c(
    -2 * as.numeric(logLik(model_w_covariates)),
    AIC(model_w_covariates),
    BIC(model_w_covariates)
  )
)


##
    ------------------------------------------------------------------------------------------------------
model_summary <- summary(model_w_covariates)
likelihood_ratio <- model_summary$logtest
score_test <- model_summary$sctest
wald_test <- model_summary$waldtest
```

```
tibble(
  Test = c('Likelihood Ratio', 'Score', 'Wald'),
  `Chi Square` = c(likelihood_ratio['test'], score_test['test'], wald_test['test']),
  DF = c(likelihood_ratio['df'], score_test['df'], wald_test['df']),
  `Pr > ChiSq` = c(likelihood_ratio['pvalue'], score_test['pvalue'], wald_test['pvalue'])
)



##  ---------------------------------------------------------------------------------------------------
anova_results <- Anova(model_w_covariates, test.statistic = 'Wald', type = '3')

joint_test <- tibble(
  Effect = rownames(anova_results),
  DF = anova_results$DF,
  `Wald Chi-Square` = round(anova_results$Chisq, 4),
  `Pr > ChiSq` = round(anova_results$`Pr(>Chisq)`, 6)
)

joint_test


##  ---------------------------------------------------------------------------------------------------
tibble(
  Parameter = rownames(model_summary$coefficients),
  DF = rep(1, length(Parameter)),
  `Parameter Estimate` = round(model_summary$coefficients[, 'coef'], 5),
  `Standard Error` = round(model_summary$coefficients[, 'se(coef)'], 5),
  `Chi-Square` = round((model_summary$coefficients[, 'coef'] / model_summary$coefficients[,
    'se(coef)'])^2, 4),
  `Pr > ChiSq` = round(model_summary$coefficients[, 'Pr(>|z|)'], 4),
  `Hazard Ratio` = round(model_summary$coefficients[, 'exp(coef)'], 3),
  `95% CI` = paste0('(', round(model_summary$conf.int[, 'lower .95'], 3), ', ',
    round(model_summary$conf.int[, 'upper .95'], 3), ')')
)



##  ---------------------------------------------------------------------------------------------------
hr_results <- tibble(
  RiskGroup = character(),
  HazardRatio = numeric(),
  LowerCI = numeric(),
  UpperCI = numeric(),
  stringAsFactors = FALSE
```

```
)

formula <- as.formula(paste0('Surv(', time_var, ', ', event_var, ') ~ ', treatment_var))

for (groupi in levels(results_data[[risk_group_var]])) {
  subset_data <- results_data %>% filter(!!sym(risk_group_var) == groupi)
  group_model <- coxph(formula, data = subset_data)

  hr <- exp(coef(group_model))
  ci <- exp(confint(group_model))

  hr_results <- rbind(hr_results, tibble(
    RiskGroup = groupi,
    HazardRatio = round(hr, 4),
    LowerCI = round(ci[1], 4),
    UpperCI = round(ci[2], 4)
  ))
}

print(hr_results)


##    ------------------------------------------------------------------------------------------------------------
p <- ggplot(hr_results, aes(x = RiskGroup, y = HazardRatio)) +
  geom_errorbar(aes(ymin = LowerCI, ymax = UpperCI), width = 0.2, size = 2, color = 'orange',
    alpha = 1) +
  geom_point(color = 'black', size = 4, fill = 'blue', shape = 21, stroke = 1) +
  geom_hline(yintercept = 1, linetype = 'solid', color = 'black') +
  scale_y_log10() +
  labs(
    # title = "HTE on the Relative Scale (Hazard Ratio)",
    y = "Hazard Ratio (95% CI)",
    x = "Risk Group"
  ) +
  theme_classic()

# ggsave(file.path(IMG_BASE_DIR, IMG_SUBDIR, 'hr_by_risk_group.png'), plot = p, width =
    PLOT_WIDTH, height = PLOT_HEIGHT, dpi = PLOT_DPI)

print(p)
```

```
library(ltm)          # For Point-Biserial correlation
library(lsr)          # For Eta Squared
library(dplyr)        # For data manipulation
library(tidyr)        # For handling missing values
library(DescTools)    # For Cramr's V & Phi coefficient
library(reshape2)
```

```r
# Function to compute correlation matrix for mixed data types
compute_mixed_corr <- function(df, filter_p = FALSE) {
  df <- df %>% drop_na()  # Remove rows with missing values
  cols <- colnames(df)
  n <- length(cols)

  cor_matrix <- matrix(NA, nrow = n, ncol = n, dimnames = list(cols, cols))

  calc_pearson <- function(x, y) cor(x, y, use = "complete.obs", method = "pearson")
  calc_point_biserial <- function(num, bin) biserial.cor(num, as.numeric(bin), level = 2)  #
    Ensure binary is numeric
  calc_phi <- function(bin1, bin2) Phi(table(bin1, bin2))  # Fix: Use Phi() from DescTools
  calc_cramers_v <- function(cat1, cat2) CramerV(table(cat1, cat2))
  calc_eta_squared <- function(num, cat) {
    if (length(unique(cat)) > 1) {
      eta_sq <- etaSquared(aov(num ~ cat), type = 2)
      return(as.numeric(eta_sq[1]))  # Convert to numeric
    } else {
      return(NA)
    }
  }

  # Identify feature types
  numeric_features <- names(df)[sapply(df, is.numeric)]
  binary_features <- names(df)[sapply(df, function(x) is.factor(x) && nlevels(x) == 2)]
  categorical_features <- names(df)[sapply(df, function(x) is.factor(x) && nlevels(x) > 2)]

  # Compute correlation values
  for (i in 1:n) {
    for (j in i:n) {
      var1 <- df[[cols[i]]]
      var2 <- df[[cols[j]]]

      # Self-correlation
      if (i == j) {
        cor_matrix[i, j] <- 1
        next
      }

      # Numeric-Numeric: Pearson
      if (cols[i] %in% numeric_features && cols[j] %in% numeric_features) {
        cor_matrix[i, j] <- cor_matrix[j, i] <- calc_pearson(var1, var2)

        # Binary-Numeric: Point-Biserial
      } else if ((cols[i] %in% binary_features && cols[j] %in% numeric_features) ||
                 (cols[j] %in% binary_features && cols[i] %in% numeric_features)) {
        if (cols[i] %in% binary_features) {
          cor_matrix[i, j] <- cor_matrix[j, i] <- calc_point_biserial(var2, var1)
        } else {
```

```
          cor_matrix[i, j] <- cor_matrix[j, i] <- calc_point_biserial(var1, var2)
        }

        # Binary-Binary: Phi
      } else if (cols[i] %in% binary_features && cols[j] %in% binary_features) {
        cor_matrix[i, j] <- cor_matrix[j, i] <- calc_phi(var1, var2)

        # Categorical-Categorical: Cramrs V
      } else if (cols[i] %in% categorical_features && cols[j] %in% categorical_features) {
        cor_matrix[i, j] <- cor_matrix[j, i] <- calc_cramers_v(var1, var2)

        # Binary-Categorical: Cramrs V
      } else if ((cols[i] %in% binary_features && cols[j] %in% categorical_features) ||
                 (cols[j] %in% binary_features && cols[i] %in% categorical_features)) {
        cor_matrix[i, j] <- cor_matrix[j, i] <- calc_cramers_v(var1, var2)

        # Numeric-Categorical: Eta-Squared
      } else if ((cols[i] %in% numeric_features && cols[j] %in% categorical_features) ||
                 (cols[j] %in% numeric_features && cols[i] %in% categorical_features)) {
        if (cols[i] %in% numeric_features) {
          cor_matrix[i, j] <- cor_matrix[j, i] <- calc_eta_squared(var1, var2)
        } else {
          cor_matrix[i, j] <- cor_matrix[j, i] <- calc_eta_squared(var2, var1)
        }
      }
    }
  }

  # Convert correlation matrix to data frame
  cor_df <- as.data.frame(cor_matrix)
  cor_df <- round(cor_df, 2)

  # Optional: Filter correlations below p < 0.05
  if (filter_p) {
    cor_df[abs(cor_df) < 0.05] <- NA
  }

  return(cor_df)
}
```