A Time-to-Event Model Analysis of the German Comprehensive Cohort Study Data on Recurrence-Free Survival in Primary Node-Positive Breast Cancer

student id - 2331122

2023-04-26

Abstract

This study aimed to develop a robust prognostic time-to-event model that assesses the impact of various factors, including hormonal therapy, on recurrence-free survival in primary node-positive breast cancer patients. Our analysis utilised data from the German Comprehensive Cohort Study, which included 686 participants. Addressing missing data was imperative, and we employed multiple imputation techniques after determining that the missingness mechanism was missing at random (MAR). Using Kaplan-Meier curves and log-rank tests for exploratory data analysis helped select the covariates. Each imputed dataset was fitted with a Cox proportional hazards model, and Rubin's principles were used to combine the results. We performed model comparisons using the likelihood ratio test and evaluated the proportionality assumption using Schoenfeld residuals. We implemented stratification to address violations of the proportionality hazard assumption. Finally, we assessed the goodness-of-fit and predictive performance of the final model using the concordance index (c-index) and time-dependent ROC curves. Our findings provide a reliable prognostic model that can guide clinical decision-making and enhance patient outcomes in node-positive primary breast cancer patients.

Introduction

Breast cancer, a prevalent global health issue, particularly affects women and strains healthcare systems. Primary node-positive breast cancer, which has spread to nearby lymph nodes, is often associated with higher relapse rates and poorer prognoses than node-negative cases. Therefore, understanding factors influencing recurrence-free survival is crucial for developing effective treatments and improving patient outcomes.

"Recurrence-free survival" describes the time between breast cancer diagnosis or treatment and the first recurrence or death from any cause. Factors impacting this include age, tumour size, grade, number of positive lymph nodes, progesterone and estrogen receptor statuses, and menopausal status. The role of hormonal therapy in improving recurrence-free survival has been explored in various studies with differing results.

Seven hundred twenty individuals with primary node-positive breast cancer participated in the 1996 German Comprehensive Cohort Study (Schmoor, Olschewski, and Schumacher 1996), which sheds light on the variables influencing recurrence-free survival. There were no discernible changes between the treatment groups (Schumacher et al. 1994). This dataset makes it possible to investigate the relationships between several prognostic variables and recurrence-free survival.

The study aims to build a robust prognostic model for time-to-event analysis, assessing factors such as hormonal therapy on recurrence-free survival. We will address missing data, justify covariate selection, and explore time-to-event models. The chosen model will be fitted and evaluated, providing clinicians with a reliable tool for informed decision-making and personalized treatments. This study contributes to ongoing efforts to improve breast cancer management and patients' quality of life.

Methods

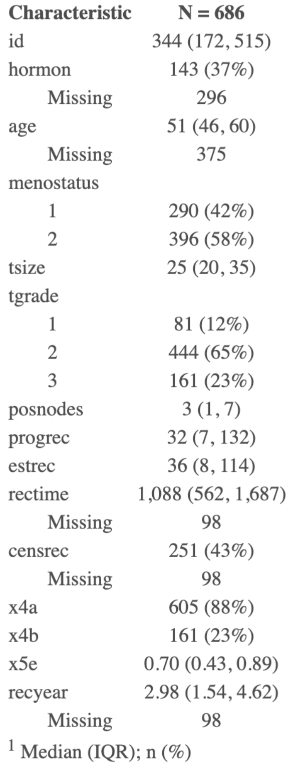
Using R, we implemented the following methods to create a prognostic time-to-event model assessing the impact of various factors on recurrence-free survival time:

1. Data Inspection and management of missing data
2. Covariate selection
3. Model selection
4. Evaluating the model's performance .

1. Data Inspection and management of missing data:

The examination of data and the handling of missing data are essential components in the development of a reliable prognostic time-to-event model. The process of Data Inspection is crucial in ensuring the completeness and consistency of a given dataset. On the other hand, managing missing data involves making informed decisions on the most appropriate method to address the issue of missing values. The missing data can result in biased estimates and decreased statistical power, underscoring the importance of effectively addressing missing data (Little and Rubin 2002). Effective management of missing data can enhance the precision and dependability of our estimations, culminating in a resilient prognostic model that can assess the influence of various factors on the duration of recurrence-free survival.

Table-1: Summary of Breast Cancer Data



Characteristic	N = 686
id	344 (172, 515)
hormon	143 (37%)
Missing	296
age	51 (46, 60)
Missing	375
menostatus	
1	290 (42%)
2	396 (58%)
tsize	25 (20, 35)
tgrade	
1	81 (12%)
2	444 (65%)
3	161 (23%)
posnodes	3 (1, 7)
progrec	32 (7, 132)
estrec	36 (8, 114)
rectime	1,088 (562, 1,687)
Missing	98
censrec	251 (43%)
Missing	98
x4a	605 (88%)
x4b	161 (23%)
x5e	0.70 (0.43, 0.89)
recyear	2.98 (1.54, 4.62)
Missing	98

[1] Median (IQR); n (%)

Table-1 represents the extent of missing values in the given data set. This table provides information about the number and percentage of missing values in the dataset, which is essential for assessing data quality and deciding on appropriate data cleaning and imputation strategies.

Figure-1 is a visual representation of the missing proportions that help illustrate each variable's missing values. This information is useful for understanding the completeness of the dataset and identifying variables that may need to be handled or imputed to avoid bias or loss of information in any analysis or

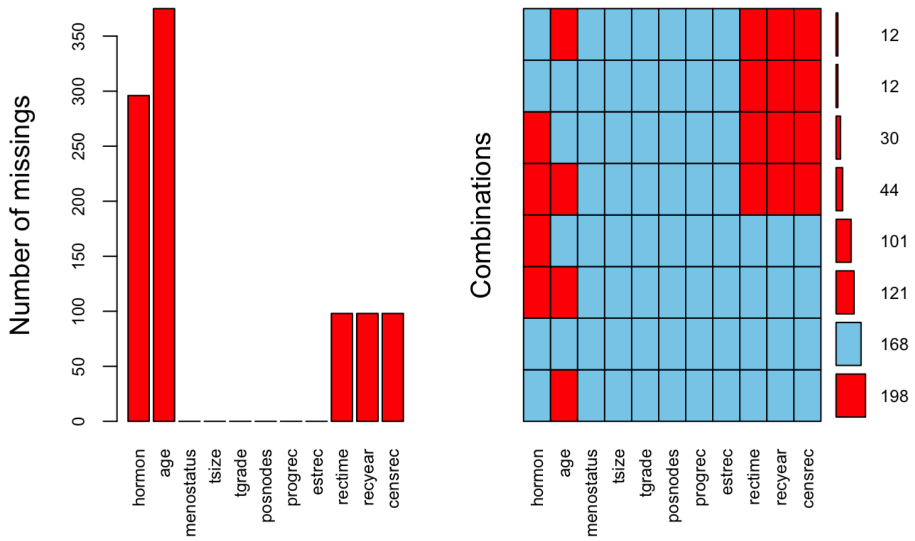


Figure 1: Missingness Plot for Breast Cancer Data

modelling. In addition, the missing proportions plot is more helpful in finding the missing mechanism of the data. Missing Completely At Random (MCAR) occurs when all observations have the same missingness fraction independent of other factors.

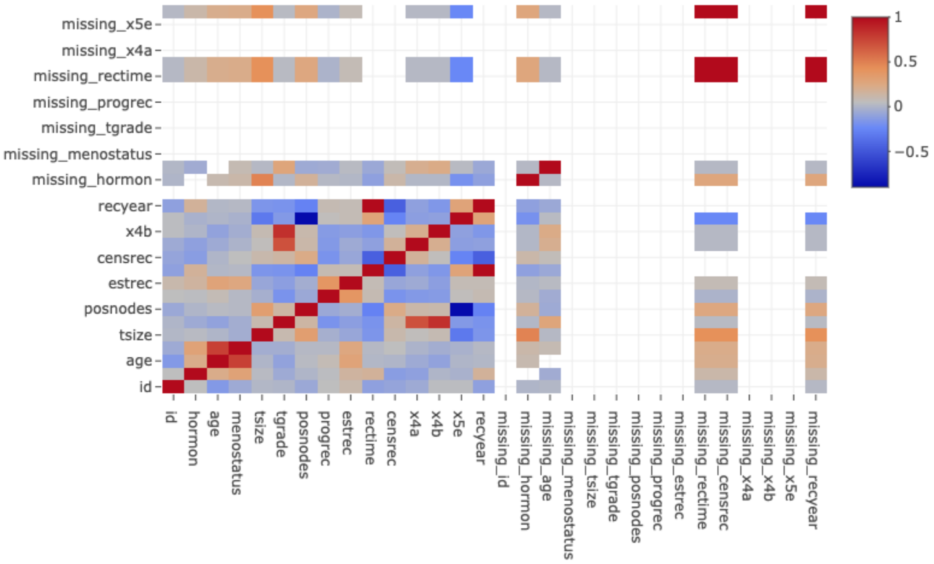


Figure 2: Correlation Matrix Heatmap

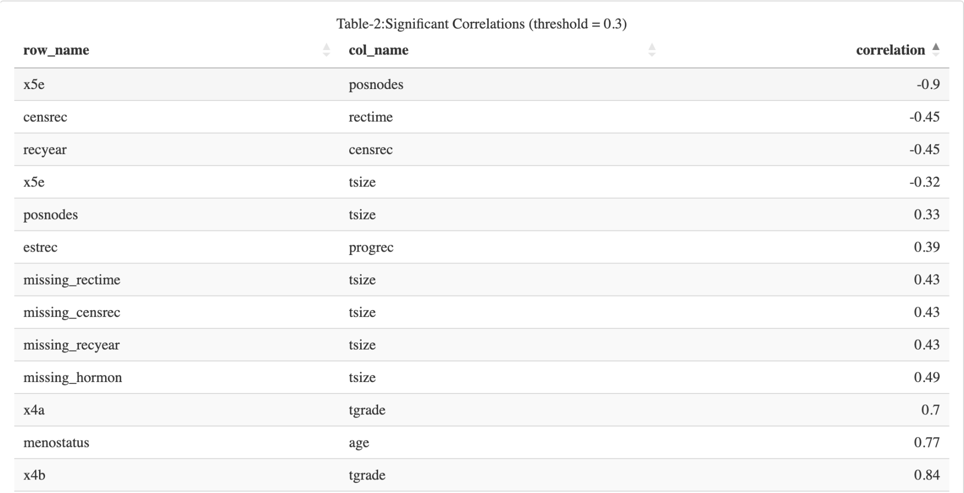


Table-2:Significant Correlations (threshold = 0.3)

row_name	col_name	correlation
x5e	posnodes	-0.9
censrec	rectime	-0.45
recyear	censrec	-0.45
x5e	tsize	-0.32
posnodes	tsize	0.33
estrec	progrec	0.39
missing_rectime	tsize	0.43
missing_censrec	tsize	0.43
missing_recyear	tsize	0.43
missing_hormon	tsize	0.49
x4a	tgrade	0.7
menostatus	age	0.77
x4b	tgrade	0.84

The notable correlations help to identify the dataset relationships and may be helpful in further studies and models. Multiple imputation is a common approach for missing at random (MAR) data. The likelihood of missingness in MAR data relies only on the observed data. Thus, missingness is consistently connected to seen data but not missing data. Multiple imputations may estimate reasonable values for missing data by using observable data (Carpenter 2013).

Multiple imputations accommodate missing data uncertainty, a significant benefit. Numerous imputation creates numerous datasets with plausible missing data values. The same statistical model is used to analyse both datasets individually, and the results are pooled using methods that account for imputed dataset variability. This method yields statistical inference that accounts for missing data uncertainty.

Unlike listwise deletion or single imputation, multiple imputations provide unbiased population parameter estimates under MAR without losing information or statistical power. Multiple imputations may be used with regression, survival analysis, and clustering to address missing data in categorical and continuous variables.

2. Covariate selection

Covariate selection is the process of identifying and choosing relevant variables that may affect the outcome of a study or analysis (Heinze, Wallisch, and Dunkler 2018).

The Kaplan-Meier curve of the interest variable illustrates the impact of said variable on the survival rate free from recurrence. The plot contains:

- a table that presents information on the number of patients at risk,
- the number of events (recurrence or death) at each time point, and
- the survival probability estimate and confidence interval for each group.

The statistical significance of the differences in recurrence-free survival between the two groups is indicated by the plot's p-value. There is a statistically significant difference in recurrence-free survival among groups, with a p-value below 0.05.

The Kaplan-Meier curve is a valuable tool for visualising and analysing recurrence-free survival. Examining the survival curve, p-value, risk table, and pertinent confounding variables allows one to draw meaningful conclusions regarding the association and directs future research or clinical interventions.

The log-rank test is a statistical technique utilised to assess the equivalence of survival distributions among two or more groups. The study evaluates the correlation between a covariate and recurrence-free survival by ascertaining the statistical significance of the differences in survival curves among the covariate categories.

The p-value is a statistical measure that indicates the likelihood of observing the test statistic, or a more extreme value, under the assumption that there is no significant difference in survival rates between the groups being compared. A reduced p-value denotes heightened evidence against the null hypothesis that posits the absence of variance in survival.

3. Model Selection

Choosing an appropriate statistical model is referred to as model selection.

Fitting a Cox proportional hazards regression model on each imputed dataset addresses the uncertainty that arises due to the multiple imputation process while evaluating the correlation between covariates and the outcome of time-to-event.

Upon conducting Cox model fittings on all imputed datasets, the estimates and standard errors are aggregated through Rubin's rules, resulting in a particular set of outcomes. The pooling process integrates the within-imputation variability, which reflects the model fit in each imputed dataset, with the between-imputation variability, which denotes the ambiguity in the imputed values.

A more resilient and precise estimation of the hazard ratios and their corresponding confidence intervals is achieved by utilising the Cox proportional hazards regression model on every imputed dataset. This ultimately leads to a more comprehensive comprehension of the correlation between the covariates and the time-to-event outcome while appropriately addressing the issue of missing data.

The Cox proportional hazards model is mathematically represented as

$$h(t|X) = h_0(t) \times \exp(\beta_1 \times \mathrm{X}_1 + \beta_2 \times \mathrm{X}_2 + \beta_3 \times \mathrm{X}_3 + ...)$$

where, h(t|X) is the hazard at time t given covariates X, h_0(t) is the baseline hazard, X_i represents the selected covariates, $\beta_1 \; to \; \beta_7$ are the regression coefficients.

After the model fitting, the model is evaluated for the proportionality assumption using Schoenfeld residuals or log-log survival plots (Hess 1995). The violations are addressed with different possible mechanisms, including stratification or time-dependent covariates into the Cox model. We have considered the stratification mechanism for addressing the violation of the proportionality assumption in this study.

The stratified Cox model is mathematically represented as

$$\lambda(t|X) = \lambda_0(t) \times \exp(\beta_1 \times \mathrm{X}_1 + \beta_2 \times \mathrm{X}_2 + \beta_3 \times \mathrm{X}_3) + \beta_4 \times \mathrm{strata}(\mathrm{var}_1) + \beta_5 \times \mathrm{strata}(\mathrm{var}_2)$$

where, $\lambda(t|X)$ is the hazard rate at time t for an individual with covariate values X, $\lambda_0(t)$ is the baseline hazard function, β values are the log hazard ratio associated with the chosen variables strata(var_1) and strata(var_2) are the variables addressed using the stratification.

The rationale behind utilising the Cox proportional hazards model in this study was predicated on its relative superiority over the other models for analysing time-to-event data (Schober and Vetter 2018). This approach offers several benefits, including its semi-parametric characteristics, interpretive capacity, and extensive suitability for time-to-event analyses. The Cox model is semi-parametric since survival time distribution assumptions are not needed. Parametric survival models need an exponential or Weibull baseline hazard function. This disadvantages the procedure under evaluation. Due to its baseline hazard flexibility, the semi-parametric Cox model can better represent complicated covariate-survival time interactions. Every covariate's hazard ratio interprets the Cox model. Simple ratios may show how different variables impact event probability. Hazard ratios compare event risk across categories or continuous variable values. Medical practitioners and patients may grasp research results. It is often used to analyse time-to-event data because of its versatility and reliability.

4. Model Assessment

Evaluating a model's goodness-of-fit and predictive performance is essential to ensure accuracy and reliability when predicting outcomes. The concordance index (c-index) and time-dependent receiver operating characteristic (ROC) curves are frequently used to evaluate the predictive performance of survival models.

The c-index in Table-5 quantifies a model's ability to predict the relative order of survival times for pairs of individuals. It ranges from 0.5 (randomly generated predictions) to 1.0 (perfectly accurate predictions). Generally, a c-index of 0.7 or higher is acceptable for predicting survival outcomes.

Utilising time-dependent ROC curves, the sensitivity and specificity of a survival model are evaluated over time. The curve compares the true positive rate (sensitivity) to the false positive rate (1-specificity) at various time intervals. Good predictive performance is indicated by a model with a time-dependent ROC curve that consistently lies above the diagonal (representing chance).

Evaluating a model's goodness-of-fit and predictive performance using measures such as the c-index and time-dependent ROC curves can help identify areas in which the model requires refinement and guide modifications to increase the model's accuracy and reliability in predicting outcomes.

The quantification of predictor variable effects on hazard rate is facilitated by the computation of hazard ratios (HRs) and their corresponding 95% confidence intervals. Hazard ratios (HRs) measure the change in the hazard rate for a unit change in the predictor variable. Confidence intervals, on the other hand, indicate the level of precision associated with the estimate. This data can facilitate the interpretation of the relative significance of each predictor in forecasting event incidences.

Results:

From Table-1, it is statistically significant that the variables other than hormonal therapy, age, recurrence-free survival in days, recurrence-free survival in years and the Censoring indicator are complete.

Figure-1, the missing proportions plot, illustrates the MCAR missingness. The data is likely MCAR if the missingness pattern is random and there are no significant changes in missingness between subgroups. However, consistent variances in the missingness pattern across data subsets show that the data are not MCAR.

Table-2, the Significant correlations table, shows favourable relationships between specific variables. For example, according to the statistical values of the table, Menopausal status and Age, Indicators of tumour grade and tumour grade show similar connections. Their correlation coefficients exceed 0.3, indicating a statistically significant relation. This means that an increase in one variable usually increases the other. Figure-2 is a heatmap, a graphical representation of data in which colours indicate the value of a variable. They help identify patterns and relationships between variables in large datasets and are frequently employed in exploratory data analysis and visualisation.

The significant correlation between the missingness indicators and certain observed variables implies that the missing data mechanism may not conform to the missing completely at random (MCAR) assumption. Instead, the data are probably missing at random (MAR), indicating that the likelihood of missing data is associated with the observed data rather than the missing data. The circumstance mentioned above bears consequences for the methodologies that can be employed to address missing data in subsequent analyses.

The prognostic time-to-event model should employ recurrence-free survival in days as the outcome variable instead of Recurrence-free survival in years as it estimates the time to recurrence. In contrast, Recurrence-free survival in days offers a more exact metric. The more exact time to recurrence would be better for fitting a prognostic time-to-event model. Therefore, it would be better to use recurrence-free survival in days as the outcome variable, which allows for greater flexibility in time scale and time-dependent variables.

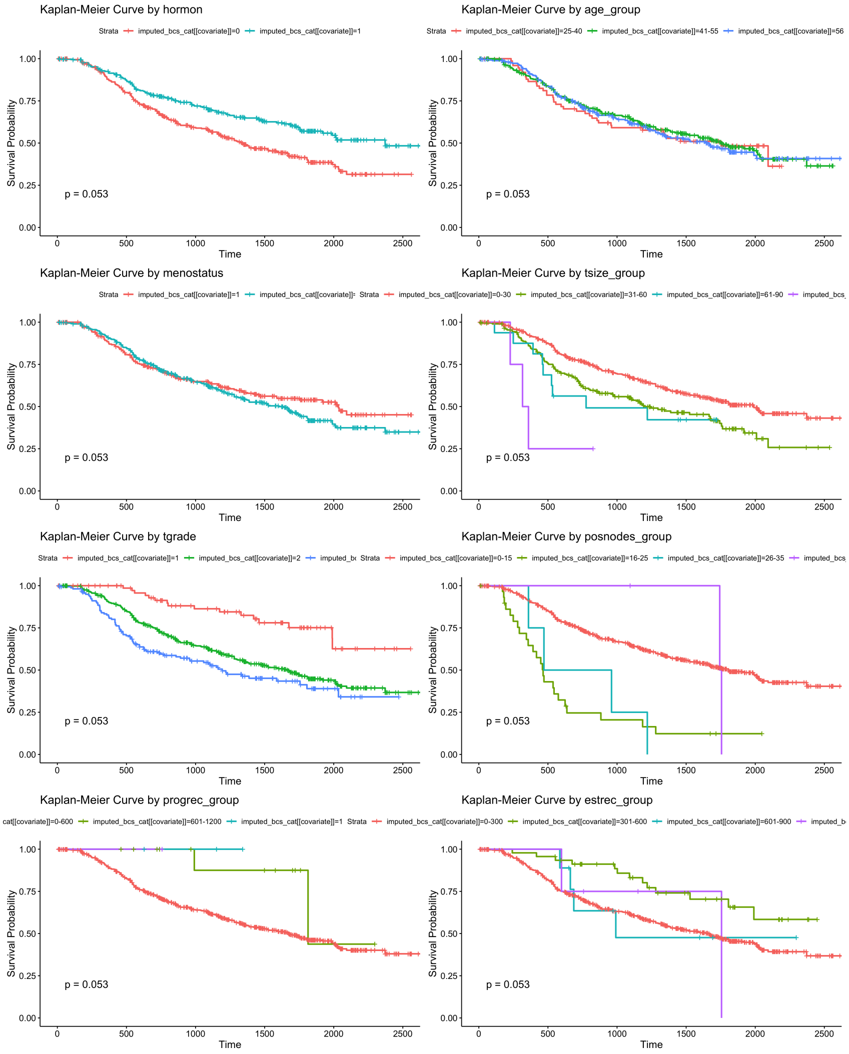


Figure 3: Kaplan-Meier Curve by Covariate

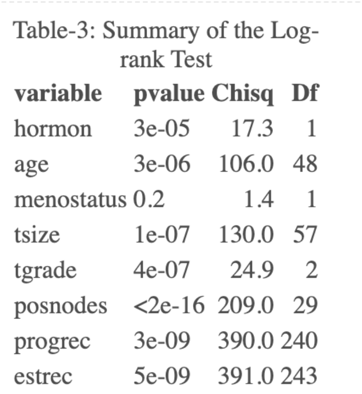


Table-3: Summary of the Log-rank Test

variable	pvalue	Chisq	Df
hormon	3e-05	17.3	1
age	3e-06	106.0	48
menostatus	0.2	1.4	1
tsize	1e-07	130.0	57
tgrade	4e-07	24.9	2
posnodes	<2e-16	209.0	29
progrec	3e-09	390.0	240
estrec	5e-09	391.0	243

The results from Figure-3 and Table-3 suggest that hormonal therapy, age, tumour size, tumour grade, number of positive lymph nodes, progesterone receptor, and estrogen receptor exhibit significantly low p-values, which signifies compelling evidence against the null hypothesis of no variation in survival. The previous observation implies that the variables hold substantial predictive value in determining the likelihood of survival within the imputed dataset.

In contrast, the variable menopausal status has a p-value of 0.2, which indicates less statistical significance against the null hypothesis. The findings suggest that the menopausal status variable may not hold substantial predictive value for survival outcomes within the imputed dataset.

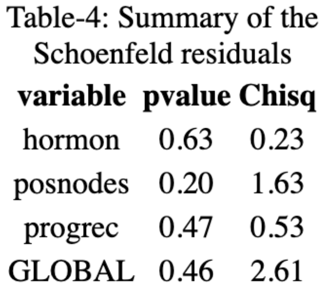


Table-4: Summary of the Schoenfeld residuals

variable	pvalue	Chisq
hormon	0.63	0.23
posnodes	0.20	1.63
progrec	0.47	0.53
GLOBAL	0.46	2.61

The Schoenfeld residuals test for each covariate and the global proportional hazards assumption test are shown in Table-4. Each test assumes the covariate has a consistent influence throughout time. The null hypothesis is rejected if the p-value is less than 0.05 and the covariate violates proportional hazards.

All p-values are more significant than 0.05; thus, we cannot reject the null hypothesis for all variables and infer that they fulfil the proportional hazards assumption. However, the global test gives a p-value of 0.46, showing no model-wide violation of the proportional hazards assumption.

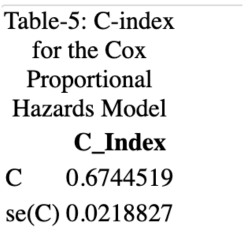


Table-5: C-index for the Cox Proportional Hazards Model

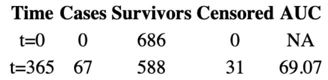
	C_Index
C	0.6744519
se(C)	0.0218827

The c_index from Table-5 indicates that the estimated C-index for the entire dataset is 0.674 with a standard deviation of 0.022. In this instance, the C-index value indicates that the model's predictive accuracy for the entire dataset is moderate to excellent.

Note that interpreting the C-index depends on the study's context and the addressed clinical query. For example, depending on the application and the expected level of predictive accuracy, a C-index of 0.674

may be regarded as either high or low. Therefore, it is essential to interpret the C-index in light of the study's design, patient population, and clinical context.

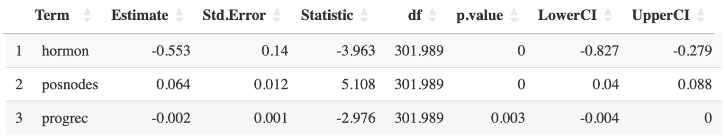
Table-6: Time-dependent ROC Curve Estimated



Time	Cases	Survivors	Censored	AUC
t=0	0	686	0	NA
t=365	67	588	31	69.07

The discriminative ability of the Cox model in predicting event occurrences at a specific time point is evaluated by the time-dependent ROC curve estimated using IPCW, with a sample size of 686 and without competing risks. From the table-6, we can interpret that At time zero, the Area under the curve cannot be applied as no events have occurred. At time point t=365, the dataset recorded 67 cases, 588 survivors, and 31 censored observations. Currently, the Area Under the Curve (AUC) is calculated to be 69.07%. This suggests the model exhibits moderate discriminatory ability in distinguishing between high and low-risk subjects after one year. The estimation process utilised the IPCW: marginal method, and the computational duration was 0 seconds.

Table-7: Summary of Pooled Cox Model with Confidence Intervals

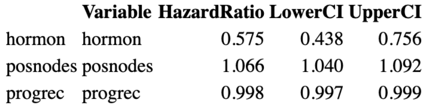


	Term	Estimate	Std.Error	Statistic	df	p.value	LowerCI	UpperCI
1	hormon	-0.553	0.14	-3.963	301.989	0	-0.827	-0.279
2	posnodes	0.064	0.012	5.108	301.989	0	0.04	0.088
3	progrec	-0.002	0.001	-2.976	301.989	0.003	-0.004	0

The summary of the pooled Cox model and stratified variables with confidence intervals is presented in Table 7. Here is a comprehensive explanation of the results:

The estimate for hormone treatment is -0.553, with a standard error of 0.14. The minus indicator indicates that hormone therapy is associated with a decreased risk rate. With 301.989 degrees of freedom, the z-statistic is -3.963, and the p-value is less than 0.001. This indicates that hormone treatment has a statistically significant effect. The 95% confidence interval for the estimate ranges from -0.827 to -0.279, indicating that the actual effect of hormone therapy on the hazard rate is negative and statistically significant. The estimated number of positive lymph nodes is 0.064, with a standard deviation of 0.012. The plus sign indicates that an increase in positive lymph nodes is associated with a higher risk rate. The p-value is less than 0.001, and the z-statistic is 5.108 with 301.989 degrees of freedom, indicating that the number of positive lymph nodes has a statistically significant effect. The estimate's 95% confidence interval ranges from 0.04 to 0.088, confirming a significant positive correlation between positive lymph nodes and hazard rate. The estimate for the progesterone receptor status is -0.002, with a standard error of 0.001. Higher progesterone receptor concentrations are associated with a lower incidence of risk, as indicated by the minus sign. The p-value of 0.003 and the z-statistic of -2,976 with 301,989 degrees of freedom indicate that the effect of progesterone receptor status is statistically significant. The estimate's 95% confidence interval ranges from -0.004 to 0, indicating a significant negative association between progesterone receptor levels and the risk rate. In conclusion, hormone therapy has a significant protective effect, the number of positive lymph nodes has a significant positive association with the risk rate, and the progesterone receptor status has a significant, albeit modest, negative association with the risk rate. These findings offer important insights for clinical decision-making and future research on the factors influencing the event of interest.

Table-8: Hazard Ratios and 95% Confidence Intervals



	Variable	HazardRatio	LowerCI	UpperCI
hormon	hormon	0.575	0.438	0.756
posnodes	posnodes	1.066	1.040	1.092
progrec	progrec	0.998	0.997	0.999

The hazard ratios in Table-8 provide insight into the effect of each variable on the hazard rate, taking into account all other model variables. The risk ratio for hormone therapy is 0.575% (95% confidence interval: 0.438-0.756). Adjusting for the other variables indicates that patients receiving hormone treatment have a 42.5% lower hazard rate than those not. The 95% confidence interval indicates that, with 95% certainty, the true hazard ratio for hormone therapy lies between 0.43 and 0.75, indicating a statistically significant protective effect. The hazard ratio for the number of positive lymph nodes is 1.066 (95% confidence interval: 1.040 - 1.092). This indicates that the risk rate increases by 6.6% for each additional positive lymph node after adjusting for other variables. The 95% confidence interval for the hazard ratio ranges from 1.040 to 1.092, indicating a statistically significant positive correlation between the number of positive lymph nodes and the hazard rate. The risk ratio for progesterone receptor status is 0.999 (95% confidence interval: 0.997 - 0.999). Considering the other variables, this suggests that the hazard rate decreases by 0.2% for each unit increase in progesterone receptor levels. The 95% confidence interval for the hazard ratio lies between 0.997 and 0.999, indicating a statistically significant, albeit modest, inverse relationship between progesterone receptor levels and the hazard rate. In conclusion, the results demonstrate that hormone therapy has a significant protective effect, whereas the number of positive lymph nodes correlates significantly with the risk rate. Moreover, the status of progesterone receptors has a small but statistically significant negative association with the hazard rate. These findings can aid clinical decision-making and guide future investigations into the factors influencing the event of interest.

Discussion

Using clinically relevant and outcome-associated factors is preferable when creating a prognostic model. Survival analysis variables should predict survival or time to event. The log-rank tests show that age, tumour size, grade, progesterone receptor, estrogen receptor, and the number of positive lymph nodes have extremely significant p-values, showing that they are powerful predictors of survival in this dataset and menopausal status does not have any statistical significance for predicting survival in this sample. Variable selection requires more than statistical significance. Consider the factors' clinical significance and influence on the result. For example, for breast cancer patients, age and tumour size may predict survival; tumour grade measures tumour aggressiveness and affects prognosis, whereas several positive lymph nodes, progesterone receptor, and estrogen receptor indicate cancer's extent, responsiveness to therapy, and survival. Breast cancer's hormone receptor status is also therapeutically important. Breast cancer tumours may be hormone receptor-positive, hormone receptor-negative, or HER2-positive. Hormone treatment improves hormone receptor-positive tumour outcomes. Thus, incorporating the hormone variable in the prognostic model may help with therapy and prognosis. Based on statistical significance and clinical relevance, we added age, tumour size, grade, number of positive lymph nodes, progesterone receptor, and estrogen receptor as initial factors in the breast cancer survival predictive model. However, comprehensive data analysis, assessment of possible confounding variables, and model validation in separate datasets should determine the final covariate selection. Possible biases due to the missing data mechanism assumptions and the generalizability of the results to other populations are two limitations of this study. Future work could include validating the prognostic model using external datasets, examining alternative methods for resolving missing data, and incorporating additional variables, such as genomic or lifestyle variables, to improve the model's predictive performance. In addition, constructing machine learning-based models and comparing their performance to the Cox model could provide additional insight into predicting recurrence-free survival in patients with node-positive primary breast cancer.

References

Carpenter, James R. 2013. *Multiple Imputation and Its Application.* Illustrated. Statistics in Practice. Wiley. https://www.wiley.com/en-us/Multiple+Imputation+and+its+Application-p-9780470740521.

Heinze, Georg, Christine Wallisch, and Daniela Dunkler. 2018. "Variable Selection–a Review and Recommendations for the Practicing Statistician." *Biometrical Journal. Biometrische Zeitschrift* 60 (3): 431–49. https://doi.org/10.1002/bimj.201700067.

Hess, K. R. 1995. "Graphical Methods for Assessing Violations of the Proportional Hazards Assumption in Cox Regression." *Statistics in Medicine* 14 (15): 1707–23. https://doi.org/10.1002/sim.4780141510.

Little, Roderick J. A., and Donald B. Rubin. 2002. *Statistical Analysis with Missing Data.* Wiley Series in Probability and Statistics. Wiley. https://www.wiley.com/en-us/Statistical+Analysis+with+Missing+Data%2C+2nd+Edition-p-9780471183860.

Schmoor, Claudia, Mark Olschewski, and Martin Schumacher. 1996. “Randomized and Non-Randomized Patients in Clinical Trials: Experiences with Comprehensive Cohort Studies.” *Stat Med* 15 (3): 263–71. https://doi.org/10.1002/(SICI)1097-0258(19960215)15:3%3C263::AID-SIM165%3E3.0.CO;2-K.

Schober, Patrick, and Thomas R. Vetter. 2018. “Survival Analysis and Interpretation of Time-to-Event Data: The Tortoise and the Hare.” *Anesthesia & Analgesia* 127 (3): 792–98. https://doi.org/10.1213/ANE.0000000000003653.

Schumacher, Martin, Gunther Bastert, Hans Bojar, Klaus Hübner, Mark Olschewski, Willi Sauerbrei, Claudia Schmoor, Christian Beyerle, Robert L. Neumann, and Hans F. Rauschecker. 1994. “Randomized 2 x 2 Trial Evaluating Hormonal Treatment and the Duration of Chemotherapy in Node-Positive Breast Cancer Patients. German Breast Cancer Study Group.” *J Clin Oncol* 12 (10): 2086–93. https://doi.org/10.1200/JCO.1994.12.10.2086.

Appendix

```r
#Section 1: Data inspection and handling missing data:
#loading required libraries
library(gtsummary)
library(VIM)
library(mice)
library(ggplot2)
library(naniar)
library(survival)
library(survminer)
library(gplots) #loading the gplots package for heatmap.2 function
library(dplyr)
library(tidyverse)
library(knitr)
library(gridExtra)
library(timeROC)
library(kableExtra)
```

```r
#loading the data
bcs <- readRDS("assessment.rds")
head(bcs)
```

```r
#summarizing the data and inspecting the data for missing values
summary(bcs)
```

```r
#representing missing values into more understandable format

# create summary table
bcs_tbl <- bcs %>%
  tbl_summary(missing = "ifany", missing_text = "Missing")

# modify caption
bcs_tbl <- bcs_tbl %>% modify_caption("Table-1: Summary of Breast Cancer Data")

# convert to kable format
bcs_tbl_kable <- bcs_tbl %>%
  as_kable_extra()

# print table using knitr::knit_print()
bcs_table <- knit_print(bcs_tbl_kable,
                        caption = "Table-1: Summary of Breast Cancer Data")
bcs_table
```

```
if (!require(VIM)) {
  install.packages("VIM")
  library(VIM)
}
#exploring the patterns of missing data
missingnessplot <- aggr(bcs[, c("hormon", "age", "menostatus", "tsize", "tgrade",
                                "posnodes","progrec", "estrec", "rectime", "recyear", "censrec")],
                    prop = FALSE, numbers = TRUE, sortCombs = TRUE,
                    cex.axis = 0.75, cex.numbers = 0.75)
summary(missingnessplot)
```

```
#perform a correlation analysis between missing indicators and observed variables

# Create a dataframe with missingness indicators
missing_indicators <- as.data.frame(sapply(bcs, function(x) ifelse(is.na(x), 1, 0)))
names(missing_indicators) <- paste0("missing_", names(missing_indicators))

# Combine the original dataset with missingness indicators
combined_data <- cbind(bcs, missing_indicators)

# Calculate the correlation matrix for the combined dataset
combined_correlations <- cor(combined_data, use = "pairwise.complete.obs")

# Round the correlation values to 2 decimals
combined_correlations <- round(combined_correlations, 2)

# Set a threshold for significant correlations
threshold <- 0.3

# Identify significant correlations and store them in a table
significant_correlations <- which(abs(combined_correlations) >
                                    threshold & !is.na(combined_correlations), arr.ind = TRUE)

# Filter out upper triangle of significant correlation matrix
significant_correlations <- significant_correlations[significant_correlations[,1]
                                                     >= significant_correlations[,2],]

# Create data frame of significant correlations
table_rows <- data.frame(row_name =
                           rownames(combined_correlations)[significant_correlations[,1]],
                          col_name =
                           colnames(combined_correlations)[significant_correlations[,2]],
                          correlation =
                           combined_correlations[significant_correlations])

# Remove redundant values
table_rows <- table_rows[!duplicated(paste(pmin(table_rows$row_name,
                                                table_rows$col_name),
                                             pmax(table_rows$row_name,
                                                  table_rows$col_name))),]

# Render the interactive table of significant correlations
datatable(table_rows,
          options = list(dom = 't', pageLength = 40),
          caption = "Table-2:Significant Correlations (threshold = 0.3)",
          rownames = FALSE)
```

```
# Generate the interactive heatmap
heatmap_plot <- plot_ly(x = colnames(combined_correlations),
                        y = rownames(combined_correlations),
                        z = combined_correlations,
                        type = "heatmap",
                        colorscale = "RdBu")

heatmap_plot
```

```
#finding the correlation between rectime and recyear
cor(bcs$rectime, bcs$recyear, use = "complete.obs")
```

```
#using multiple imputation for handling missing data
imputed_bcs <- mice(bcs, m = 5, maxit = 50)
#since recyear and rectime are strongly correlated,
#we divide rectime by 365.25 to convert it from days to years and impute into the dataset
imputed_bcs$imp$recyear <- imputed_bcs$imp$rectime / 365.25
imputed_bcs <- complete(imputed_bcs)
#checking if there are any missing values in the imputed data set
sum(is.na(imputed_bcs))
```

```
#Section-2: Covariate Selection
#Performing exploratory data analysis (e.g., Kaplan-Meier curves, log-rank tests)
#to identify potential associations between the covariates and recurrence-free survival.
#For each covariate, we can create a Kaplan-Meier curve to
# visualize the association between that variable and recurrence-free survival.
imputed_bcs_cat <- imputed_bcs
# Categorize continuous variables into quartiles
#for more readable format of the curves
imputed_bcs_cat$age_group <- cut(imputed_bcs_cat$age,
                               breaks = c(25, 40, 55, Inf),
                               include.lowest = TRUE,
                               labels = c( "25-40", "41-55",
                                           "56 and older"))
imputed_bcs_cat$tsize_group <- cut(imputed_bcs_cat$tsize,
                                 breaks = c(0, 30, 60, 90, 120),
                                 include.lowest = TRUE,
                                 labels = c("0-30", "31-60", "61-90",
                                            "91 and larger"))
imputed_bcs_cat$posnodes_group <- cut(imputed_bcs_cat$posnodes,
                                   breaks = c(0, 15, 25, 35, Inf),
                                   include.lowest = TRUE,
                                   labels = c("0-15", "16-25", "26-35",
                                              "36 and more"))
imputed_bcs_cat$progrec_group <- cut(imputed_bcs_cat$progrec,
                                  breaks = c(0, 600, 1200, 1800, Inf),
                                  include.lowest = TRUE,
                                  labels = c("0-600", "601-1200", "1201-1800",
                                             "1800 and larger"))
imputed_bcs_cat$estrec_group <- cut(imputed_bcs_cat$estrec,
                                  breaks = c(0, 300, 600, 900, Inf),
                                  include.lowest = TRUE,
                                  labels = c("0-300", "301-600", "601-900",
                                             "900 and larger"))

# List of categorized covariate names
covariate_names <- c("hormon", "age_group", "menostatus", "tsize_group", "tgrade",
```

```r
                     "posnodes_group", "progrec_group", "estrec_group")

# Function to create a Kaplan-Meier plot for a specific covariate
create_km_plot <- function(covariate) {
  km_fit <- survfit(Surv(rectime, censrec) ~ imputed_bcs_cat[[covariate]],
                    data = imputed_bcs_cat)
  g <- ggsurvplot(
    km_fit,
    data = imputed_bcs_cat,
    risk.table = TRUE,
    risk.table.height = 0.5,
    pval = TRUE,
    xlab = "Time",
    ylab = "Survival Probability",
    title = paste("Kaplan-Meier Curve by", covariate)
  )
  return(g$plot) # Return the ggplot object
}
# Collect the plots in a list
plots <- lapply(covariate_names, create_km_plot)

# Convert the ggplot objects to grobs
plots_grob <- lapply(plots, ggplotGrob)

# Arrange the grobs into a grid using arrangeGrob()
grid_plots <- do.call(gridExtra::arrangeGrob, c(plots_grob, ncol = 2))

# Save the arranged plots to a single png file
png("km_plots.png", width = 16, height = 20, units = "in", res = 300)
grid::grid.newpage()
grid::grid.draw(grid_plots)
dev.off()
```

```r
#To statistically test the association between a covariate and
# recurrence-free survival, you can perform a log-rank test.
#Log-rank test for the hormon variable
logrank_test_hormon <- survdiff(Surv(rectime, censrec) ~ hormon,
                                data = imputed_bcs)
logrank_test_hormon
#Log-rank test for the age variable
logrank_test_age <- survdiff(Surv(rectime, censrec) ~ age,
                             data = imputed_bcs)
logrank_test_age
#Log-rank test for the menostatus variable
logrank_test_menostatus <- survdiff(Surv(rectime, censrec) ~ menostatus,
                                    data = imputed_bcs)
logrank_test_menostatus
#Log-rank test for the tsize variable
logrank_test_tsize <- survdiff(Surv(rectime, censrec) ~ tsize,
                               data = imputed_bcs)
logrank_test_tsize
#Log-rank test for the tgrade variable
logrank_test_tgrade <- survdiff(Surv(rectime, censrec) ~ tgrade,
                                data = imputed_bcs)
logrank_test_tgrade
#Log-rank test for the posnodes variable
logrank_test_posnodes <- survdiff(Surv(rectime, censrec) ~ posnodes,
                                  data = imputed_bcs)
logrank_test_posnodes
#Log-rank test for the progrec variable
logrank_test_progrec <- survdiff(Surv(rectime, censrec) ~ progrec,
                                 data = imputed_bcs)
logrank_test_progrec
#Log-rank test for the estrec variable
logrank_test_estrec <- survdiff(Surv(rectime, censrec) ~ estrec,
                                data = imputed_bcs)
logrank_test_estrec
```

```
# Create a data frame with variable names and p-values
pvalue_table <- data.frame(variable = c("hormon", "age", "menostatus", "tsize", "tgrade",
                                        "posnodes", "progrec", "estrec"),
                           pvalue = c('3e-05', '3e-06', 0.2, '1e-07',
                                      '4e-07', '<2e-16', '3e-09', '5e-09'),
                           Chisq = c(17.3, 106, 1.4, 130,
                                     24.9, 209, 390, 391),
                           Df = c(1, 48, 1, 57, 2, 29, 240, 243))

# Print the p-value table using knitr::kable()
pvalue_table <- kable(pvalue_table,
                      caption = "Table-3: Summary of the Log-rank Test")

pvalue_table
```

```
#Section-3: Model Selection:
#fitting a prognostic time-to-event mode
#Fit the Cox model on each imputed dataset and store the results in a list.
cox_model_list <- lapply(imputed_bcs, function(x) {
  coxph(Surv(rectime, censrec) ~ hormon + age + tsize + tgrade + posnodes + progrec + estrec, data =
})

pooled_cox_model <- pool(cox_model_list)
summary(pooled_cox_model)
```

```
#Fit the Cox model on each imputed dataset and store the results in a list.
cox_model_list1 <- lapply(imputed_bcs, function(x) {
  coxph(Surv(rectime, censrec) ~ hormon + age + tsize +
          tgrade + posnodes + progrec, data = imputed_bcs)
})

pooled_cox_model1 <- pool(cox_model_list1)
summary(pooled_cox_model1)
```

```
# Calculate the likelihood ratio test statistic

# Define a function to extract log-likelihood values
#and calculate the likelihood ratio test statistic
lr_statistic <- function(model1, model2) {
  loglik1 <- model1$loglik[2]
  loglik2 <- model2$loglik[2]
  return(-2 * (loglik2 - loglik1))
}

# Calculate the likelihood ratio test statistic for each imputed dataset
lr_stats_list <- mapply(lr_statistic, cox_model_list,
                        cox_model_list1, SIMPLIFY = TRUE)
# Pool the likelihood ratio test statistics using Rubin's rules
pooled_lr_statistic <- mean(lr_stats_list)
pooled_lr_statistic

# Calculate the p-value
p_value <- pchisq(pooled_lr_statistic, df = 1, lower.tail = FALSE)
p_value
```

```
#Fit the Cox model on each imputed dataset and store the results in a list.
cox_model_list2 <- lapply(imputed_bcs, function(x) {
  coxph(Surv(rectime, censrec) ~ hormon + tsize + tgrade +
          posnodes + progrec, data = imputed_bcs)
})

pooled_cox_model2 <- pool(cox_model_list2)
summary(pooled_cox_model2)
```

```
# Calculate the likelihood ratio test statistic

# Define a function to extract log-likelihood values and
# calculate the likelihood ratio test statistic
lr_statistic <- function(model1, model2) {
  loglik1 <- model1$loglik[2]
  loglik2 <- model2$loglik[2]
  return(-2 * (loglik2 - loglik1))
}

# Calculate the likelihood ratio test statistic for each imputed dataset
lr_stats_list1 <- mapply(lr_statistic, cox_model_list1,
                         cox_model_list2, SIMPLIFY = TRUE)
# Pool the likelihood ratio test statistics using Rubin's rules
pooled_lr_statistic1 <- mean(lr_stats_list1)
pooled_lr_statistic1

# Calculate the p-value
p_value1 <- pchisq(pooled_lr_statistic1, df = 1, lower.tail = FALSE)
p_value1
```

```
# Calculate the likelihood ratio test statistic

# Define a function to extract log-likelihood values and
# calculate the likelihood ratio test statistic
lr_statistic <- function(model1, model2) {
  loglik1 <- model1$loglik[2]
  loglik2 <- model2$loglik[2]
  return(-2 * (loglik2 - loglik1))
}

# Calculate the likelihood ratio test statistic for each imputed dataset
lr_stats_list2 <- mapply(lr_statistic, cox_model_list,
                         cox_model_list2, SIMPLIFY = TRUE)
# Pool the likelihood ratio test statistics using Rubin's rules
pooled_lr_statistic2 <- mean(lr_stats_list2)
pooled_lr_statistic2

# Calculate the p-value
```

```
p_value2 <- pchisq(pooled_lr_statistic2, df = 1, lower.tail = FALSE)
p_value2
```

```
#To evaluate the proportionality assumption using Schoenfeld residuals,
# you can perform a global test on each imputed dataset and then pool the test results.

#To evaluate the proportionality assumption using Schoenfeld residuals
#for the cox_model_list2, you can use the cox.zph() function
# Create a function to extract the Schoenfeld residuals from a Cox model
get_schoenfeld_res <- function(cox_model) {
  schoenfeld_res <- cox.zph(cox_model)
  schoenfeld_res_df <- data.frame(schoenfeld_res$z, schoenfeld_res$y)
  colnames(schoenfeld_res_df) <- c("Schoenfeld_Residual", "Time")
  return(schoenfeld_res_df)
}
cox_model_schoenfeld_list <- lapply(cox_model_list2, function(cox_model) {
  return(cox.zph(cox_model))
})
cox_model_schoenfeld_list
```

```
# Fit the Cox model on each imputed dataset and store the results in a list.
cox_model_list_strat <- lapply(imputed_bcs, function(x) {
  coxph(Surv(rectime, censrec) ~ hormon + strata(tsize) + strata(tgrade)
        + posnodes + progrec, data = imputed_bcs)
})


pooled_cox_model_strat <- pool(cox_model_list_strat)
summary(pooled_cox_model_strat)
```

```
summary_df <- data.frame(
  Term = c("hormon", "posnodes", "progrec"),
  Estimate = c(-0.553020766, 0.063729157, -0.001577577),
  Std.Error = c(0.1395428452, 0.0124773888, 0.0005300532),
  Statistic = c(-3.963089, 5.107572, -2.976261),
  df = c(301.9893, 301.9893, 301.9893),
  p.value = c(9.239590e-05, 5.791714e-07, 3.153788e-03)
)

summary_datatable <- datatable(summary_df, options = list(pageLength = 10,
                                                          autoWidth = TRUE), caption = "Table")

print(summary_datatable)

estimates <- c(-0.553020766, 0.063729157, -0.001577577)
se_estimates <- c(0.1395428452, 0.0124773888, 0.0005300532)
alpha <- 0.05
z <- qnorm(1 - alpha / 2)
lower_limits <- estimates - z * se_estimates
upper_limits <- estimates + z * se_estimates

summary_df$LowerCI <- lower_limits
summary_df$UpperCI <- upper_limits

summary_datatable <- datatable(
  summary_df,
  options = list(pageLength = 10, autoWidth = TRUE),
  caption = "Table-8: Summary of Pooled Cox Model with Confidence Intervals"
)

print(summary_datatable)
```

```
#To evaluate the proportionality assumption using Schoenfeld residuals,
#you can perform a global test on each imputed dataset and then pool the test results.

#To evaluate the proportionality assumption using Schoenfeld residuals
#for the cox_model_list2, you can use the cox.zph() function
# Create a function to extract the Schoenfeld residuals from a Cox model
get_schoenfeld_res <- function(cox_model) {
  schoenfeld_res <- cox.zph(cox_model)
  schoenfeld_res_df <- data.frame(schoenfeld_res$z, schoenfeld_res$y)
  colnames(schoenfeld_res_df) <- c("Schoenfeld_Residual", "Time")
  return(schoenfeld_res_df)
}
cox_model_schoenfeld_list_strat <- lapply(cox_model_list_strat,
                                          function(cox_model) {
  return(cox.zph(cox_model))
})
cox_model_schoenfeld_list_strat
```

```
# Create a data frame with variable names and p-values
Schoenfeldres_table <- data.frame(variable = c("hormon","posnodes",
                                               "progrec", "GLOBAL"),
                                  pvalue = c(0.63, 0.20, 0.47, 0.46),
                                  Chisq = c(0.23, 1.63, 0.53, 2.61))

# Print the p-value table using knitr::kable()
Schoenfeldres_table <- kable(Schoenfeldres_table,
                             caption = "Table-4: Summary of the Schoenfeld residuals", align = "c")

Schoenfeldres_table
```

```
# Fit Cox proportional hazards model to the complete dataset
cox_model_complete <- with(imputed_bcs, coxph(Surv(rectime, censrec) ~ hormon + strata(tsize)
                                              + strata(tgrade) + posnodes + progrec))

# Calculate C-index for the complete dataset
cox_model_complete_summary <- summary(cox_model_complete)
c_index <- cox_model_complete_summary$concordance
c_index

c_index_table <- data.frame(C_Index = c_index)

# Print the C-index table using knitr::kable()
c_index_table <- kable(c_index_table,
                       caption = "Table-5: C-index for the Cox Proportional Hazards Model", align =
c_index_table
```

```
# Compute the time-dependent ROC curve and AUC
predicted_risk <- predict(cox_model_complete, type = "lp", newdata = imputed_bcs)
time_point <- 365 # Time point of interest (e.g., 1 year)
event_time <- imputed_bcs$rectime
event_status <- imputed_bcs$censrec
```

```
time_roc <- timeROC(event_time, event_status,
                    marker = predicted_risk, cause = 1, times = time_point)
auc <- time_roc$AUC[1]
time_roc
```

```
# Create a data frame with the output values
time_roc_output <- data.frame(
  Time = c("t=0", "t=365"),
  Cases = c(0, 67),
  Survivors = c(686, 588),
  Censored = c(0, 31),
  AUC = c(NA, 69.07)
)
# Print the output table using knitr::kable()
time_roc <- kable(time_roc_output,
                  caption = "Table-6: Time-dependent ROC Curve Estimated", align = "c")
time_roc
```

```
#Calculate hazard ratios (HRs) and their 95% confidence intervals
#to quantify the effect of each predictor variable on the hazard rate.
# Extract hazard ratios (HRs) and 95% confidence intervals from the summary
cox_summary <- summary(cox_model_complete)
hr_table <- data.frame(
  Variable = rownames(cox_summary$conf.int),
  HazardRatio = cox_summary$conf.int[, "exp(coef)"],
  Lower95CI = cox_summary$conf.int[, "lower .95"],
  Upper95CI = cox_summary$conf.int[, "upper .95"]
)

# Print the HR table
#print(hr_table)

hr_table <- kable(hr_table,
                  caption = "Table-7: Hazard Ratios and 95% Confidence Intervals")
hr_table
```