

Build Survival Model: Cox Proportional Hazards Model

Mingcheng Hu

Table of contents

Load Data	2
Univariate Cox Proportional Hazards Model	4
Multivariate Cox Proportional Hazards Model	8
PH Assumption Assessment	8
Variable Selection	10
LASSO	10
Stepwise Selection based on BIC	15
Summary of Variable Selection	16

```
library(tidyverse)
library(survival)
library(forestplot)
library(glmnet)
library(ggfortify)
library(kableExtra) # include knitr automatically

source("/work/users/y/u/yuukias/BIOS-Material/BIOS992/utils/csv_utils.r")
# * Don't use setwd() for Quarto documents!
# setwd("/work/users/y/u/yuukias/BIOS-Material/BIOS992/data")

adjust_type <- ifelse(exists("params"), params$adjust_type, "minimal") #
  ↳ options: "minimal", "partial", "full"
```

```

impute_type <- ifelse(exists("params"), params$impute_type, "unimputed") #
  ↪ options: "unimputed", "imputed"
include_statin <- ifelse(exists("params"), params$include_statin, "no") #
  ↪ options: "yes", "no"

```

```

# string of parameters
adjust_type_str <- switch(adjust_type,
  minimal = "minimal",
  partial = "partial",
  full = "full"
)
print(paste0("Model Adjustment Type: ", adjust_type_str))

```

```
[1] "Model Adjustment Type: minimal"
```

```

impute_type_str <- switch(impute_type,
  unimputed = "unimputed",
  imputed = "imputed"
)
print(paste0("Data Imputation Type: ", impute_type_str))

```

```
[1] "Data Imputation Type: unimputed"
```

Load Data

```

if (include_statin == "yes") {
  data_train <-
  ↪ read.csv(paste0("/work/users/y/u/youkias/BIOS-Material/BIOS992/data/train_data_",
  ↪ impute_type_str, "_statin.csv"),
    header = TRUE
  )
} else {
  data_train <-
  ↪ read.csv(paste0("/work/users/y/u/youkias/BIOS-Material/BIOS992/data/train_data_",
  ↪ impute_type_str, ".csv"),
    header = TRUE
  )
}

```

```
}
```

```
data_train <- data_train[, -1] # the first column is the index generated by  
  ↪ sklearn  
(dim(data_train))
```

```
[1] 28127    100
```

```
data <- select_subset(data_train, type = adjust_type)  
(dim(data))
```

```
[1] 28127    48
```

```
colnames(data)
```

[1] "event"	"time"
[3] "HRV_SD1"	"HRV_SD2"
[5] "HRV_SD1SD2"	"HRV_S"
[7] "HRV_CSI"	"HRV_CVI"
[9] "HRV_CSI_Modified"	"HRV_PIP"
[11] "HRV_IALS"	"HRV_PSS"
[13] "HRV_PAS"	"HRV_GI"
[15] "HRV_SI"	"HRV_AI"
[17] "HRV_PI"	"HRV_C1d"
[19] "HRV_C1a"	"HRV_SD1d"
[21] "HRV_SD1a"	"HRV_C2d"
[23] "HRV_C2a"	"HRV_SD2d"
[25] "HRV_SD2a"	"HRV_Cd"
[27] "HRV_Ca"	"HRV_SDNNd"
[29] "HRV_SDNNa"	"HRV_ApEn"
[31] "HRV_ShanEn"	"HRV_FuzzyEn"
[33] "HRV_MSEn"	"HRV_CMSEn"
[35] "HRV_RCMSEn"	"HRV_CD"
[37] "HRV_HFD"	"HRV_KFD"
[39] "HRV_LZC"	"HRV_DFA_alpha1"
[41] "HRV_MFDFA_alpha1_Width"	"HRV_MFDFA_alpha1_Peak"
[43] "HRV_MFDFA_alpha1_Mean"	"HRV_MFDFA_alpha1_Max"
[45] "HRV_MFDFA_alpha1_Delta"	"HRV_MFDFA_alpha1_Asymmetry"
[47] "HRV_MFDFA_alpha1_Fluctuation"	"HRV_MFDFA_alpha1_Increment"

```
data <- tibble::as_tibble(data)
```

```
# * It is very hard to compare the HR as different predictors are on  
  ↳ different magnitudes, so we need to normalize them.  
time_col <- data$time  
event_col <- data$event  
data <- data %>%  
  select(-c(time, event)) %>%  
  mutate(across(where(is.numeric), scale)) %>%  
  mutate(  
    time = time_col,  
    event = event_col  
  )
```

Note now the interpretation of HR is different! For example, if HR=1.16 for the predictor in the univariate model fitted using scaled data, it means that each standard deviation increase is associated with 16% higher risk of event.

```
data_complete <- na.omit(data)
```

Univariate Cox Proportional Hazards Model

```
if (!("time" %in% colnames(data) && "event" %in% colnames(data))) {  
  stop("time and event columns are required")  
}  
predictors <- colnames(data)[!colnames(data) %in% c("time", "event")]  
  
results_univariate <- map_dfr(predictors, function(predictor) {  
  formula <- as.formula(paste("Surv(time, event) ~", predictor))  
  # cox_model_single <- coxph(Surv(time, event) ~ get(predictor), data =  
    ↳ data) # equivalent way  
  cox_model_single <- coxph(formula, data = data)  
  
  coef <- coef(cox_model_single) # log hazard ratio  
  se <- sqrt(diag(vcov(cox_model_single)))  
  
  hr <- exp(coef)  
  lower_ci <- exp(coef - 1.96 * se)
```

```

upper_ci <- exp(coef + 1.96 * se)
p_value <- summary(cox_model_single)$coefficients[5]

return(
  data.frame(
    predictor = predictor,
    hr = hr,
    lower_ci = lower_ci,
    upper_ci = upper_ci,
    p_value = p_value
  )
)
})
results_univariate$hr <- round(results_univariate$hr, 2)
results_univariate$lower_ci <- round(results_univariate$lower_ci, 2)
results_univariate$upper_ci <- round(results_univariate$upper_ci, 2)
results_univariate$ci <- paste0("(", results_univariate$lower_ci, ",",
  ↪ results_univariate$upper_ci, ")")
results_univariate$p_value <- round(results_univariate$p_value, 3)
results_univariate <- results_univariate %>% arrange(desc(hr)) # sort
  ↪ descendingly by HR

```

```

# Create forest plot
results_univariate %>%
  forestplot(
    labeltext = c(predictor, hr, ci, p_value),
    mean = hr,
    lower = lower_ci,
    upper = upper_ci,
    xlab = "Hazard Ratio",
    title = "Univariate Cox Models",
    xlog = TRUE, # * Make sure the CI are not symmetric and need to be
      ↪ transformed
    boxsize = 0.2,
    xticks = c(0.8, 0.9, 1.0, 1.1, 1.2),
    clip = c(0.8, 1.2),
    zero = 1
  ) %>%
  fp_set_style(
    box = "royalblue",
    line = "darkblue",

```

```

    summary = "royalblue"
) %>%
fp_add_header(
  predictor = c("Predictor", ""),
  hr = c("Hazard Ratio", "per SD increase"),
  ci = c("95% CI", ""),
  p_value = c("p-value", "")
) %>%
fp_decorate_graph(
  box = gpar(lty = 2, col = "lightgray"),
  graph.pos = 4
) %>% # change the position of forest plot
fp_set_zebra_style("#f9f9f9")

```



Multivariate Cox Proportional Hazards Model

```
cox_model_full <- coxph(Surv(time, event) ~ ., data = data)
summary(cox_model_full)
```

```
cox_model_full_complete <- coxph(Surv(time, event) ~ ., data = data_complete)
summary(cox_model_full_complete)
```

PH Assumption Assessment

```
cox.zph(cox_model_full)
```

	chisq	df	p
HRV_SD1	5.01e-01	1	0.48
HRV_SD2	3.96e-01	1	0.53
HRV_SD1SD2	5.07e-01	1	0.48
HRV_S	1.07e+00	1	0.30
HRV_CSI	5.14e-01	1	0.47
HRV_CVI	4.07e-03	1	0.95
HRV_CSI_Modified	3.06e-01	1	0.58
HRV_PIP	2.45e-02	1	0.88
HRV_IALS	7.94e-02	1	0.78
HRV_PSS	5.32e-02	1	0.82
HRV_PAS	2.08e+00	1	0.15
HRV_GI	5.67e-02	1	0.81
HRV_SI	5.29e-03	1	0.94
HRV_AI	9.58e-02	1	0.76
HRV_PI	6.22e-01	1	0.43
HRV_C1d	2.73e-02	1	0.87
HRV_SD1d	6.07e-01	1	0.44
HRV_SD1a	4.64e-01	1	0.50
HRV_C2d	1.41e-02	1	0.91
HRV_SD2d	3.69e-01	1	0.54
HRV_SD2a	3.71e-01	1	0.54
HRV_Cd	1.53e-01	1	0.70
HRV_SDNNd	5.35e-01	1	0.46
HRV_SDNNa	4.43e-01	1	0.51

HRV_ApEn	1.21e+00	1	0.27
HRV_ShanEn	1.52e-01	1	0.70
HRV_FuzzyEn	2.74e-01	1	0.60
HRV_MSEn	9.20e-02	1	0.76
HRV_CMSEn	9.87e-01	1	0.32
HRV_RCMSEn	3.57e-01	1	0.55
HRV_CD	9.40e-03	1	0.92
HRV_HFD	1.16e-01	1	0.73
HRV_KFD	9.12e-01	1	0.34
HRV_LZC	3.76e-04	1	0.98
HRV_DFA_alpha1	6.29e-01	1	0.43
HRV_MFDFA_alpha1_Width	4.90e-02	1	0.82
HRV_MFDFA_alpha1_Peak	1.41e-01	1	0.71
HRV_MFDFA_alpha1_Mean	3.71e-02	1	0.85
HRV_MFDFA_alpha1_Max	5.88e-01	1	0.44
HRV_MFDFA_alpha1_Delta	1.75e-01	1	0.68
HRV_MFDFA_alpha1_Asymmetry	5.36e-05	1	0.99
HRV_MFDFA_alpha1_Fluctuation	4.43e-01	1	0.51
HRV_MFDFA_alpha1_Increment	1.79e-01	1	0.67
GLOBAL	2.95e+01	43	0.94

```
cox.zph(cox_model_full_complete)
```

	chisq	df	p
HRV_SD1	5.01e-01	1	0.48
HRV_SD2	3.96e-01	1	0.53
HRV_SD1SD2	5.07e-01	1	0.48
HRV_S	1.07e+00	1	0.30
HRV_CSI	5.14e-01	1	0.47
HRV_CVI	4.07e-03	1	0.95
HRV_CSI_Modified	3.06e-01	1	0.58
HRV_PIP	2.45e-02	1	0.88
HRV_IALS	7.94e-02	1	0.78
HRV_PSS	5.32e-02	1	0.82
HRV_PAS	2.08e+00	1	0.15
HRV_GI	5.67e-02	1	0.81
HRV_SI	5.29e-03	1	0.94
HRV_AI	9.58e-02	1	0.76
HRV_PI	6.22e-01	1	0.43
HRV_C1d	2.73e-02	1	0.87
HRV_SD1d	6.07e-01	1	0.44
HRV_SD1a	4.64e-01	1	0.50

HRV_C2d	1.41e-02	1	0.91
HRV_SD2d	3.69e-01	1	0.54
HRV_SD2a	3.71e-01	1	0.54
HRV_Cd	1.53e-01	1	0.70
HRV_SDNNd	5.35e-01	1	0.46
HRV_SDNNa	4.43e-01	1	0.51
HRV_ApEn	1.21e+00	1	0.27
HRV_ShanEn	1.52e-01	1	0.70
HRV_FuzzyEn	2.74e-01	1	0.60
HRV_MSEn	9.20e-02	1	0.76
HRV_CMSEn	9.87e-01	1	0.32
HRV_RCMSEn	3.57e-01	1	0.55
HRV_CD	9.40e-03	1	0.92
HRV_HFD	1.16e-01	1	0.73
HRV_KFD	9.12e-01	1	0.34
HRV_LZC	3.76e-04	1	0.98
HRV_DFA_alpha1	6.29e-01	1	0.43
HRV_MFDFA_alpha1_Width	4.90e-02	1	0.82
HRV_MFDFA_alpha1_Peak	1.41e-01	1	0.71
HRV_MFDFA_alpha1_Mean	3.71e-02	1	0.85
HRV_MFDFA_alpha1_Max	5.88e-01	1	0.44
HRV_MFDFA_alpha1_Delta	1.75e-01	1	0.68
HRV_MFDFA_alpha1_Asymmetry	5.36e-05	1	0.99
HRV_MFDFA_alpha1_Fluctuation	4.43e-01	1	0.51
HRV_MFDFA_alpha1_Increment	1.79e-01	1	0.67
GLOBAL	2.95e+01	43	0.94

The proportional hazards assumption was tested using Schoenfeld residuals. None of the variables violated the PH assumption (all $p > 0.05$), indicating that the Cox proportional hazards model was appropriate for our analysis.

Variable Selection

LASSO

```
# * LASSO doesn't allow missing values
set.seed(1234)
x <- as.matrix(data_complete %>% select(-c(time, event)))
y <- Surv(data_complete$time, data_complete$event)
cox_model_lasso.cv <- cv.glmnet(
```

```

    x,
    y,
    family = "cox",
    alpha = 1, # 1 for LASSO, 0 for Ridge
    nfolds = 10
)
# plot(cox_model_lasso.cv) # Plot partial likelihood deviance vs log(lambda)
print(cox_model_lasso.cv$lambda.min)

```

```
[1] 0.0002914579
```

```
print(cox_model_lasso.cv$lambda.1se)
```

```
[1] 0.01592024
```

As mentioned in the paper, we will use the value of hyperparameter `lambda.1se` that gave the most shrunk model but still was within one standard error from the value that gave the lowest error. This is shown to produce consistently better performance than `lambda.min`.

```

cox_model_lasso <- glmnet(
  x,
  y,
  family = "cox",
  alpha = 1,
  lambda = cox_model_lasso.cv$lambda.1se
)
cox_model_lasso.coef <- coef(cox_model_lasso)
print(cox_model_lasso.coef)

```

```

46 x 1 sparse Matrix of class "dgCMatrix"
                                s0
HRV_SD1                        .
HRV_SD2                        .
HRV_SD1SD2                     .
HRV_S                          .
HRV_CSI                       -0.0009671365
HRV_CVI                       -0.0344744471
HRV_CSI_Modified              .
HRV_PIP                       0.0158238222

```

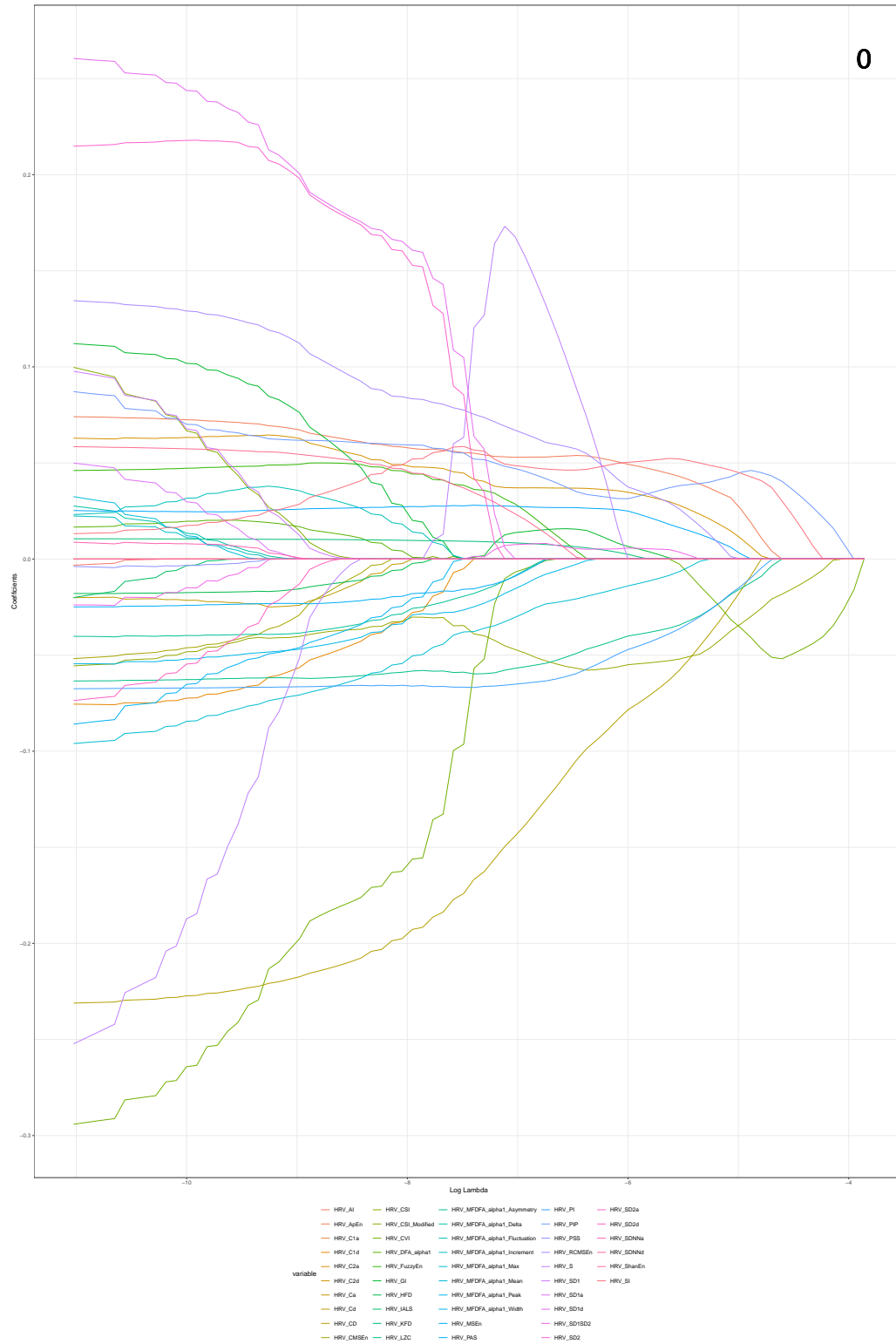
HRV_IALS	.
HRV_PSS	.
HRV_PAS	.
HRV_GI	.
HRV_SI	.
HRV_AI	.
HRV_PI	.
HRV_C1d	.
HRV_C1a	.
HRV_SD1d	.
HRV_SD1a	.
HRV_C2d	.
HRV_C2a	.
HRV_SD2d	.
HRV_SD2a	.
HRV_Cd	.
HRV_Ca	.
HRV_SDNNd	.
HRV_SDNNa	.
HRV_ApEn	.
HRV_ShanEn	.
HRV_FuzzyEn	.
HRV_MSEn	.
HRV_CMSEn	.
HRV_RCMSEn	.
HRV_CD	.
HRV_HFD	.
HRV_KFD	.
HRV_LZC	.
HRV_DFA_alpha1	.
HRV_MFDFA_alpha1_Width	.
HRV_MFDFA_alpha1_Peak	.
HRV_MFDFA_alpha1_Mean	.
HRV_MFDFA_alpha1_Max	.
HRV_MFDFA_alpha1_Delta	.
HRV_MFDFA_alpha1_Asymmetry	.
HRV_MFDFA_alpha1_Fluctuation	.
HRV_MFDFA_alpha1_Increment	.

```
selected_vars <- rownames(cox_model_lasso.coef)[which(cox_model_lasso.coef !=
↪ 0)]
print(selected_vars)
```

```
[1] "HRV_CSI" "HRV_CVI" "HRV_PIP"
```

```
# * To visualize the LASSO path, we should not supply lambda
cox_model_lasso_fullpath <- glmnet(
  x,
  y,
  family = "cox",
  alpha = 1
)
```

```
# plot(cox_model_lasso_fullpath, xvar = "lambda", label = TRUE)
autoplot(cox_model_lasso_fullpath, xvar = "lambda", label = TRUE, label.size
  ↪ = 15) +
  theme_bw() +
  theme(legend.position = "bottom") # better way of visualizing the LASSO
  ↪ path
```



Stepwise Selection based on BIC

```
# * Stepwise selection doesn't allow missing values
cox_model_step <- MASS::stepAIC(cox_model_full_complete,
  direction = "both",
  k = log(nrow(data)), # Use BIC instead of AIC
  trace = FALSE
)
```

```
summary(cox_model_step)
```

Call:

```
coxph(formula = Surv(time, event) ~ HRV_SD1 + HRV_CVI + HRV_PIP +
  HRV_PI + HRV_RCMSEn + HRV_CD, data = data_complete)
```

n= 27198, number of events= 3435

	coef	exp(coef)	se(coef)	z	Pr(> z)	
HRV_SD1	0.37451	1.45428	0.06154	6.086	1.16e-09	***
HRV_CVI	-0.26757	0.76524	0.04621	-5.791	7.02e-09	***
HRV_PIP	0.09227	1.09666	0.02344	3.936	8.28e-05	***
HRV_PI	-0.12237	0.88482	0.02044	-5.986	2.15e-09	***
HRV_RCMSEn	0.13889	1.14900	0.02187	6.350	2.15e-10	***
HRV_CD	-0.16319	0.84943	0.02434	-6.704	2.03e-11	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
HRV_SD1	1.4543	0.6876	1.2890	1.6407
HRV_CVI	0.7652	1.3068	0.6990	0.8378
HRV_PIP	1.0967	0.9119	1.0474	1.1482
HRV_PI	0.8848	1.1302	0.8501	0.9210
HRV_RCMSEn	1.1490	0.8703	1.1008	1.1993
HRV_CD	0.8494	1.1773	0.8099	0.8909

Concordance= 0.574 (se = 0.005)

Likelihood ratio test= 209.3 on 6 df, p=<2e-16

Wald test = 195.1 on 6 df, p=<2e-16

Score (logrank) test = 195.8 on 6 df, p=<2e-16

Summary of Variable Selection

We will compare the selection of variables from all models we have built.

```
# Obtain the selected variables from all models
variable_names_all <- colnames(data) %>%
  setdiff(c("time", "event"))

variable_names_univariate <- results_univariate %>%
  filter(p_value < 0.05) %>%
  pull(predictor)

variable_names_multivariate <- summary(cox_model_full_complete)$coefficients
  ↪ %>%
  as.data.frame() %>%
  rownames_to_column(var = "predictor") %>% # transpose, "predictor" will
  ↪ now be the column name
  filter(`Pr(>|z|)` < 0.05) %>%
  pull(predictor)

variable_names_lasso <-
  ↪ rownames(cox_model_lasso.coef)[which(cox_model_lasso.coef != 0)]

variable_names_step <- summary(cox_model_step)$coefficients %>%
  as.data.frame() %>%
  filter(`Pr(>|z|)` < 0.05) %>%
  rownames()
```

```
variable_selection_matrix <- matrix(
  0,
  nrow = length(variable_names_all),
  ncol = 4 # univariate, multivariate, lasso, stepwise
)
colnames(variable_selection_matrix) <- c("univariate", "multivariate",
  ↪ "lasso", "stepwise")
rownames(variable_selection_matrix) <- variable_names_all

for (variable in variable_names_all) {
  if (variable %in% variable_names_univariate) {
    variable_selection_matrix[variable, "univariate"] <- 1
  }
  if (variable %in% variable_names_multivariate) {
```



```

    variable_selection_matrix[variable, "multivariate"] <- 1
  }
  if (variable %in% variable_names_lasso) {
    variable_selection_matrix[variable, "lasso"] <- 1
  }
  if (variable %in% variable_names_step) {
    variable_selection_matrix[variable, "stepwise"] <- 1
  }
}

```

```

symbol_selected <- "*"

selection_table <- data.frame(
  Variable = variable_names_all,
  Univariate = ifelse(variable_selection_matrix[, "univariate"] == 1,
    ↪ symbol_selected, ""),
  Multivariate = ifelse(variable_selection_matrix[, "multivariate"] == 1,
    ↪ symbol_selected, ""),
  LASSO = ifelse(variable_selection_matrix[, "lasso"] == 1,
    ↪ symbol_selected, ""),
  Stepwise = ifelse(variable_selection_matrix[, "stepwise"] == 1,
    ↪ symbol_selected, "")
) %>%
  mutate(Num_Selected = rowSums(variable_selection_matrix)) %>%
  arrange(desc(Num_Selected), Variable) %>%
  as.data.frame() %>%
  remove_rownames()

variable_categories <- sapply(variable_names_all, determine_category)
category_colors <- c(
  "covariate" = "#FFB6C1", #
  "time" = "#1E90FF", #
  "frequency" = "#32CD32", #
  "poincare" = "#FF4500", #
  "entropy" = "#FF8C00", #
  "fractal" = "#FFD700", #
  "unknown" = "#000000" #
)
category_colors_names <- c(
  "covariate" = "pink", #
  "time" = "blue", #

```

```

    "frequency" = "green", #
    "poincare"  = "red", #
    "entropy"   = "orange", #
    "fractal"   = "gold" #
  )
category_legend <- sapply(names(category_colors_names), function(cat) {
  sprintf(
    "%s: %s",
    tools::toTitleCase(cat),
    tools::toTitleCase(category_colors_names[cat])
  )
}) %>%
  paste(collapse = "; ")

selection_table %>%
  kbl(
    caption = "Variable Selection by Different Models",
    align = c("|l", "c", "c", "c", "c", "c|"),
    col.names = c("Variable", "Univariate", "Multivariate", "LASSO",
      ↪ "Stepwise", "Selected Times"),
    longtable = TRUE
  ) %>%
  kable_styling(
    bootstrap_options = c("striped", "hover", "condensed", "responsive"),
    position = "center",
    font_size = 9,
    latex_options = c("repeat_header", "striped", "HOLD_position")
  ) %>%
  # Add color for different categories of variables
  column_spec(1,
    color =
      ↪ category_colors[variable_categories[selection_table$Variable]],
    bold = TRUE
  ) %>%
  # Add a header colname for four columns: Univariate, Multivariate, LASSO,
  ↪ Stepwise
  add_header_above(c(
    " " = 1,
    "Selection Methods" = 4,
    " " = 1
  )) %>%

```

```

footnote(
  general = sprintf("%s", category_legend),
  general_title = "Note:"
)

```

Warning: 'xfun::attr()' is deprecated.
 Use 'xfun::attr2()' instead.
 See help("Deprecated")

Warning: 'xfun::attr()' is deprecated.
 Use 'xfun::attr2()' instead.
 See help("Deprecated")

Table 1: Variable Selection by Different Models

Variable	Selection Methods				Selected Times
	Univariate	Multivariate	LASSO	Stepwise	
HRV_CVI	*	*	*	*	4
HRV_CD	*	*		*	3
HRV_PI	*	*		*	3
HRV_PIP	*		*	*	3
HRV_RCMSEn	*	*		*	3
HRV_ApEn	*	*			2
HRV_CSI	*		*		2
HRV_LZC	*	*			2
HRV_SD1	*			*	2
HRV_AI	*				1
HRV_C1a	*				1
HRV_C1d	*				1
HRV_C2a	*				1
HRV_C2d	*				1
HRV_CMSEn	*				1
HRV_CSI_Modified	*				1
HRV_Ca	*				1
HRV_Cd	*				1
HRV_DFA_alpha1	*				1
HRV_FuzzyEn	*				1
HRV_GI	*				1
HRV_HFD	*				1
HRV_IALS	*				1
HRV_MFDFA_alpha1_Asymmetry	*				1
HRV_MFDFA_alpha1_Delta	*				1
HRV_MFDFA_alpha1_Max	*				1
HRV_MFDFA_alpha1_Peak	*				1
HRV_PAS	*				1
HRV_PSS	*				1

Table 1: Variable Selection by Different Models (*continued*)

Variable	Univariate	Multivariate	LASSO	Stepwise	Selected Times
HRV_S		*			1
HRV_SD1SD2	*				1
HRV_SD1a	*				1
HRV_SD1d	*				1
HRV_SD2	*				1
HRV_SD2a	*				1
HRV_SD2d	*				1
HRV_SDNNa	*				1
HRV_SDNNd	*				1
HRV_SI	*				1
HRV_KFD					0
HRV_MFDFA_alpha1_Fluctuation					0
HRV_MFDFA_alpha1_Increment					0
HRV_MFDFA_alpha1_Mean					0
HRV_MFDFA_alpha1_Width					0
HRV_MSEn					0
HRV_ShanEn					0

Note:

Covariate: Pink; Time: Blue; Frequency: Green; Poincare: Red; Entropy: Orange; Fractal: Gold