# Load libraries

library(survminer)

library(survival)

library(dplyr)

library(viridis)

# Set working directory

setwd("working\_directory")

# Read data files

data <- read.table("data\_file.txt", header = TRUE)

metadata <- read.table("metadata\_file.txt", header = TRUE)

# Function to perform survival analysis

perform\_survival\_analysis <- function(data, variables, outcome\_variable, event\_variable, subset\_variable = NULL, subset\_value = NULL) {

if (!is.null(subset\_variable) && !is.null(subset\_value)) {

data <- filter(data, !!sym(subset\_variable) == subset\_value)

}

fit <- coxph(Surv(data[[outcome\_variable]], data[[event\_variable]]) ~ ridge(!!!rlang::syms(variables), theta = 10), data = data)

ggforest(fit)

res.cut <- surv\_cutpoint(data, time = outcome\_variable, event = event\_variable, variables = variables)

summary(res.cut)

plot(res.cut, variables[1], palette = "jco")

res.cat <- surv\_categorize(res.cut)

fit <- survfit(Surv(data[[outcome\_variable]], data[[event\_variable]]) ~ ., data = res.cat)

ggsurvplot(fit, data = res.cat, risk.table = TRUE, cumevents = TRUE, tables.height = 0.15, pval = TRUE, pval.method = TRUE, palette = "jco", legend.title = "subset", legend.labs = c("high", "low"), conf.int = TRUE, surv.median.line = "hv")

}

# Perform survival analysis

perform\_survival\_analysis(data,

variables = c("variable1", "variable2", ...),

outcome\_variable = "outcome\_variable\_name",

event\_variable = "event\_variable\_name",

subset\_variable = "optional\_subset\_variable",

subset\_value = "optional\_subset\_value")

# Generate heatmaps

pheatmap(data, show\_rownames = F, show\_colnames = T, clustering\_method = "ward.D2", color = magma(100), border\_color = NA, fontsize\_col = 6, annotation\_row = coldata\_data)