Inlämningsuppgift 5 - från data till inferens.

Statistiska Metoder med R

Jacob Widaeus

2024-11-09

Bakgrund

Till detta arbete används ett allmänt tillgängligt dataset som heter "jasa" och inkluderas i paketet survival. Den finns även bifogad som .rda fil som en del i inlämningen.

Jag kommer använda tidyverse till de flesta datamanipulationer och ggplot till grafritningar, då jag använt det tidigare och planerar använda det i framtiden.

Datan beskriver överlevnad av patienter på väntelistan till Standford hjärttransplantationsprogram. Den kommer i följande format:

Variable	Description
birth.dt	Birth date
accept.dt	Acceptance into program
tx.date	Transplant date
fu.date	End of followup
fustat	Dead or alive
surgery	Prior bypass surgery
age	Age (in years)
futime	Followup time
wait.time	Time before transplant
transplant	Transplant indicator
mismatch	Mismatch score
hla.a2	Particular type of mismatch
mscore	Another mismatch score
reject	Rejection occurred

Hypotes

Min hypotes är att tidigare kirurgi är en prediktor för död då det kan vara en indikation på att patienten har haft hjärtproblem tidigare.

Dependencies

Detta arbete bygger på följande paket:

```
library(tidyverse)
library(survival)
library(fs)
library(broom)
library(survminer)
library(knitr)
library(kableExtra)
```

```
Warning: package 'kableExtra' was built under R version 4.4.2
```

```
Attaching package: 'kableExtra'
```

```
The following object is masked from 'package:dplyr':

    group_rows
```

Import

Först importeras datan genom att skapa en funktion för att skapa en ny mapp “data” och spara den i en csv fil där datum inkluderas.

```
initiate <- function(output_path) {
  # Ensure the directory exists
  dir_path <- dirname(output_path)
  if (!dir.exists(dir_path)) dir.create(dir_path, recursive = TRUE)

  # Get the current date
  today_date <- Sys.Date()
```

```
  # Extract file name and extension
  file_name <- basename(output_path)
  file_extension <- tools::file_ext(file_name)
  file_base <- tools::file_path_sans_ext(file_name)

  # Remove any existing date from the file_base
  file_base <- sub("_\\d{4}-\\d{2}-\\d{2}$", "", file_base)

  # Construct the new file name with the current date
  new_file_name <- paste0(file_base, "_", today_date, ".", file_extension)
  new_output_path <- file.path(dir_path, new_file_name)

  # Assuming 'jasa' is a data frame that you want to write to CSV
  if (exists("jasa") && is.data.frame(jasa)) {
    write.csv(jasa, new_output_path)
  } else {
    stop("Data frame 'jasa' does not exist or is not a data frame.")
  }
}
```

base_dir definieras som bas-filsökvägen. Denna är datorberoende.

```
# Define base directory
base_dir <- file.path(getwd(), "block5")
```

```
# Define output path
output_path <- file.path(base_dir, "data", paste0("jasa_", Sys.Date(), ".csv"))
```

Därefter läses .csv filen in, och samtidigt sparas som en .rda fil för redundans - även här med datum. Om den som granskar detta ska återskapa, se till att ha rätt path och använd helst .r filen då .qmd inte alltid samarbetar väl med importfunktioner.

```
initiate(output_path)
data <- read.csv(output_path)
save(data, file = file.path(base_dir, "data", paste0("jasa_", Sys.Date(), ".rda")))
```

Översikt och städa upp klasser

Först bildar jag mig en översikt över datan med glimpse()

```r
glimpse(data)
```

```
Rows: 103
Columns: 15
$ X          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ~
$ birth.dt   <chr> "1937-01-10", "1916-03-02", "1913-09-19", "1927-12-23", "19~
$ accept.dt  <chr> "1967-11-15", "1968-01-02", "1968-01-06", "1968-03-28", "19~
$ tx.date    <chr> NA, NA, "1968-01-06", "1968-05-02", NA, NA, "1968-08-31", N~
$ fu.date    <chr> "1968-01-03", "1968-01-07", "1968-01-21", "1968-05-05", "19~
$ fustat     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ surgery    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,~
$ age        <dbl> 30.84463, 51.83573, 54.29706, 40.26283, 20.78576, 54.59548,~
$ futime     <int> 49, 5, 15, 38, 17, 2, 674, 39, 84, 57, 152, 7, 80, 1386, 0,~
$ wait.time  <int> NA, NA, 0, 35, NA, NA, 50, NA, NA, 11, 25, NA, 16, 36, NA, ~
$ transplant <int> 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1,~
$ mismatch   <int> NA, NA, 2, 3, NA, NA, 4, NA, NA, 2, 1, NA, 3, 1, NA, 2, NA,~
$ hla.a2     <int> NA, NA, 0, 0, NA, NA, 0, NA, NA, 0, 0, NA, 0, 0, NA, 0, NA,~
$ mscore     <dbl> NA, NA, 1.11, 1.66, NA, NA, 1.32, NA, NA, 0.61, 0.36, NA, 1~
$ reject     <int> NA, NA, 0, 0, NA, NA, 1, NA, NA, 1, 0, NA, 1, 1, NA, 1, NA,~
```

Man kan redan nu se att flera av klasserna av variablerna är fel. Jag konverterar datumen till datumklasser, och de binära variablerna som representerar TRUE/FALSE till booleans.

```r
# Convert character strings that are dates to Date type
data_cleaned <- data %>%
  mutate(across(where(~ is.character(.) && any(!is.na(as.Date(.)))), as.Date))

# Convert all 0 and 1 integer columns (including those with NAs) to booleans
data_cleaned <- data_cleaned %>%
  mutate(across(where(~ all(. %in% c(0, 1, NA)) && is.numeric(.)), ~ as.logical(.)))
```

Faktorisera alla kolumner som har färre än 10 diskreta variabler, men inte är logical.

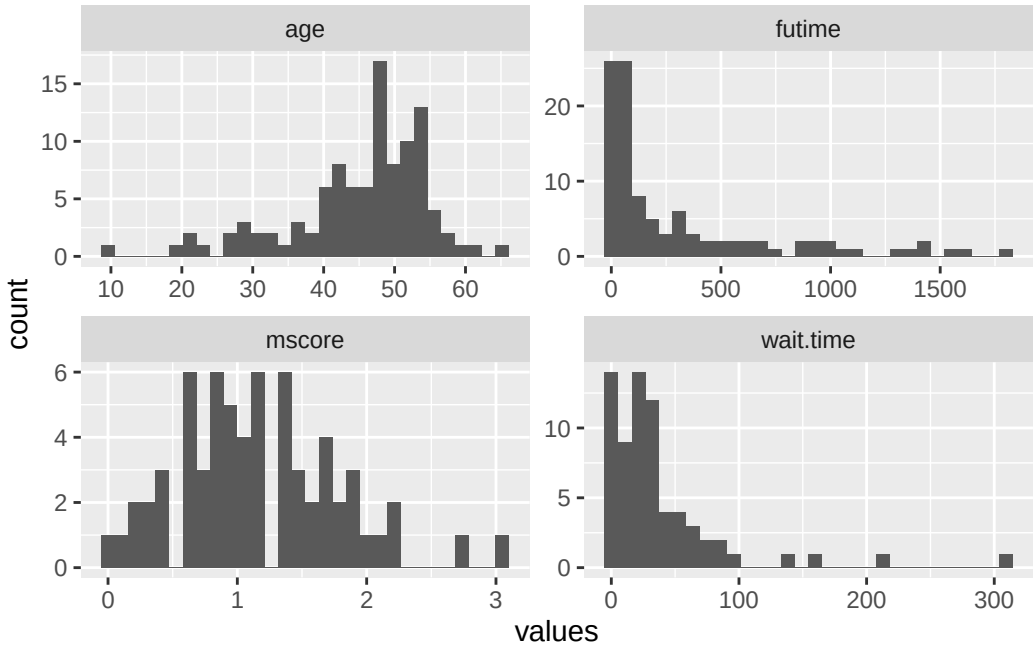
```r
data_cleaned <- data_cleaned %>%
  mutate(across(where(~ n_distinct(.) < 10 && !is.logical(.)), as.factor))
```

EDA - Exploratory data analysis

Barplots för att skildra numeriska variabler.

```r
data_cleaned %>%
  select_if(is.numeric) %>%
  select(-X) %>%  # Remove the variable 'X'
  gather(key = "variables", value = "values") %>%
  ggplot(aes(x = values)) +
  facet_wrap(~variables, scales = "free") +
  geom_histogram(bins = 30)
```

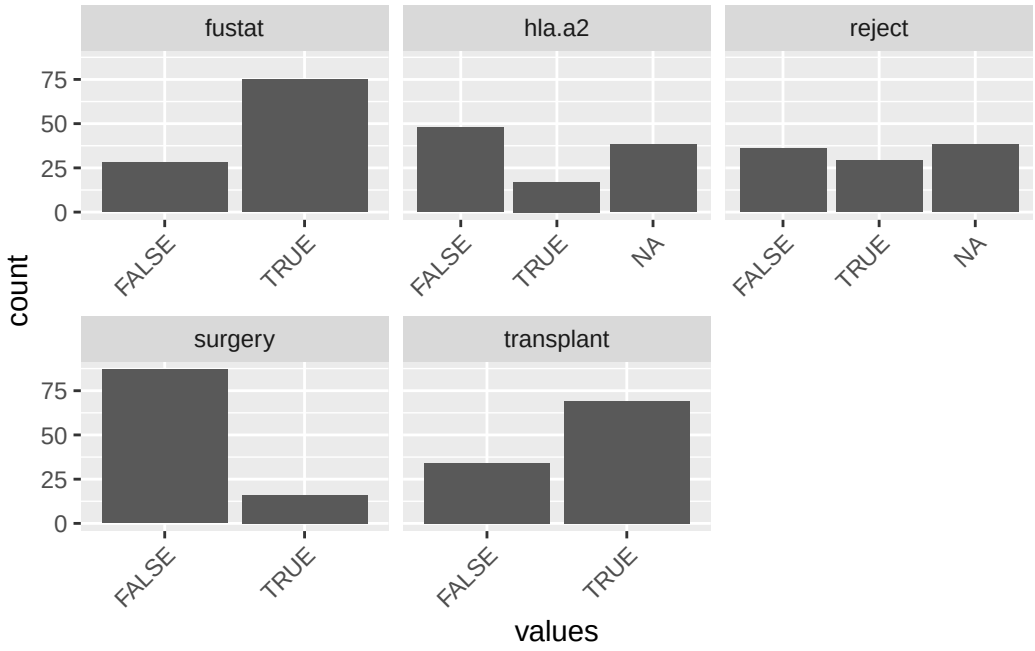
```
Warning: Removed 72 rows containing non-finite outside the scale range
(`stat_bin()`).
```



Man ser att alla numeriska variabler är skewed och ej normalfördelade.

Barplot for logistiska variabler

```r
data_cleaned %>%
  select(where(is.logical)) %>%
  pivot_longer(cols = everything(), names_to = "variables", values_to = "values") %>% #nolint
  ggplot(aes(x = values)) +
  facet_wrap(~variables, scales = "free_x") +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Hur många av de som **inte** är NA på reject är NA på hla.a2?

```
count_na_hla_a2 <- data_cleaned %>%
  filter(!is.na(reject)) %>%
  summarize(count = sum(is.na(hla.a2)))

print(count_na_hla_a2)
```

```
  count
1     0
```

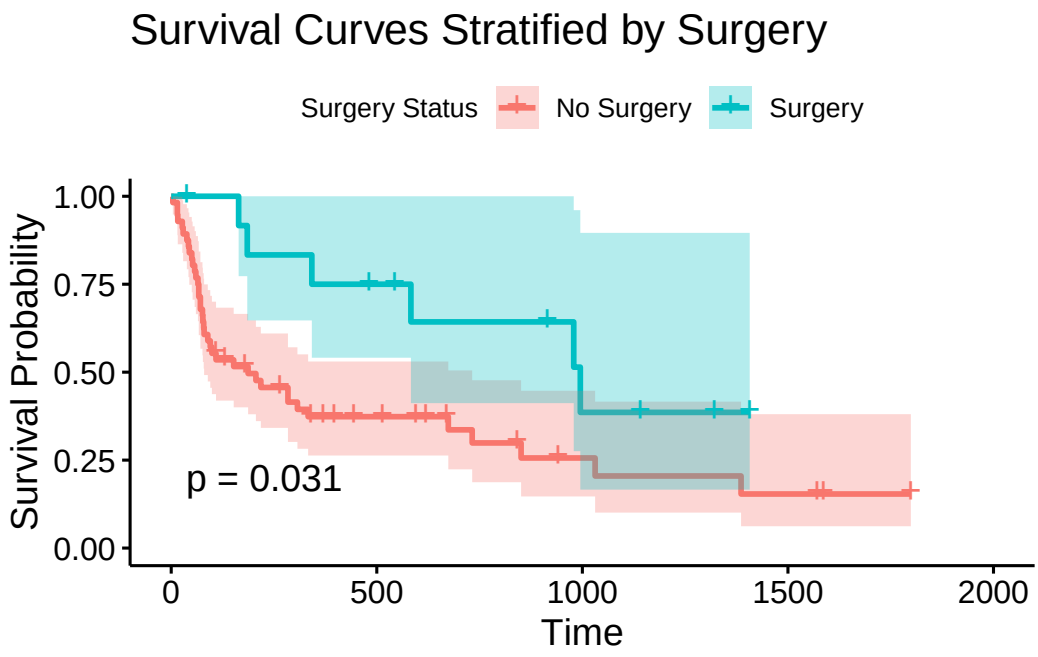
Alltså, de som inte har fått rejection finns inga värden på hla provtagning. Det verkar som att man endast provtagit de som fått en rejection.

Inferentiell analys

Vilka variabler predicerar död?

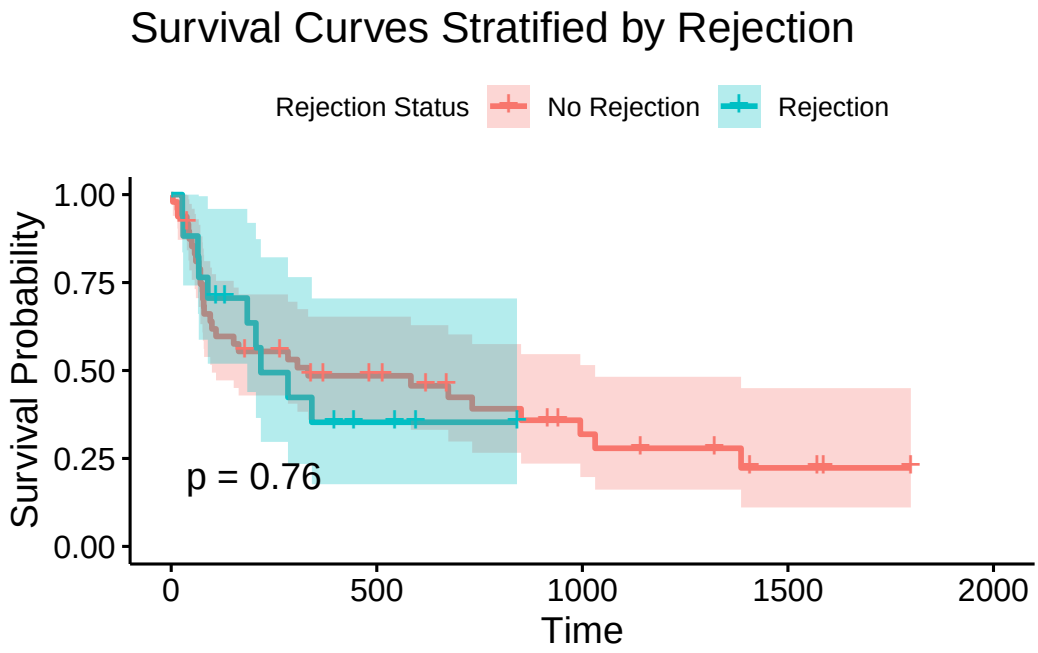
Hur ser surgery or reject ut som ensamma variabler sett över tid för de som genomgått transplantation?

```r
transplanted <- data_cleaned %>%
  filter(transplant == 1)

surv_object <- Surv(time = transplanted$futime, event = transplanted$fustat)

# Fit Kaplan-Meier curve stratified by surgery
km_fit_surgery <- survfit(surv_object ~ surgery, data = transplanted)

# Plot survival curves
ggsurvplot(km_fit_surgery,
           data = transplanted,
           pval = TRUE,
           conf.int = TRUE,
           legend.labs = c("No Surgery", "Surgery"),
           legend.title = "Surgery Status",
           xlab = "Time",
           ylab = "Survival Probability",
           title = "Survival Curves Stratified by Surgery")
```




```r
# Fit Kaplan-Meier curve stratified by reject
km_fit_reject <- survfit(surv_object ~ hla.a2, data = transplanted)
```

```r
# Plot survival curves
ggsurvplot(km_fit_reject,
           data = transplanted,
           pval = TRUE,
           conf.int = TRUE,
           legend.labs = c("No Rejection", "Rejection"),
           legend.title = "Rejection Status",
           xlab = "Time",
           ylab = "Survival Probability",
           title = "Survival Curves Stratified by Rejection")
```



Jag väljer nu att utvärdera hazard ratios för död för age, surgery, wait time och rejection. Jag selekterar först ut de transplanterade patienterna. Jag konvertar variabeln age till en boolean variabel, där jag sätter gränsen vid 45 år. jag konverterar även wait.time till en boolean variabel, där jag sätter gränsen vid 30 dagar.

```r
transplanted <- data_cleaned %>%
  filter(transplant == 1)

# Create binary variables for age and time based on specified cutoffs
transplanted$age_bin <- ifelse(transplanted$age > 45, ">45", "<=45")
transplanted$wait_bin <- ifelse(transplanted$wait.time > 30, ">30", "<=30")
```

```r
# Convert them to factors for the Cox model
transplanted$age_bin <- factor(transplanted$age_bin, levels = c("<=45", ">45"))
transplanted$wait_bin <- factor(transplanted$wait_bin, levels = c("<=30", ">30"))

# Univariable models
cox_age <- coxph(Surv(futime, fustat) ~ age_bin, data = transplanted)
cox_reject <- coxph(Surv(futime, fustat) ~ reject, data = transplanted)
cox_surgery <- coxph(Surv(futime, fustat) ~ surgery, data = transplanted)
cox_wait <- coxph(Surv(futime, fustat) ~ wait_bin, data = transplanted)

# Multivariable model
cox_multivariable <- coxph(Surv(futime, fustat) ~ age_bin + reject + surgery + wait_bin, data
```

Jag testar multivariabelanalysen.

```r
ph_test <- cox.zph(cox_multivariable)
print(ph_test)
```

```
          chisq df      p
age_bin    3.35  1 0.0672
reject     8.90  1 0.0029
surgery    2.60  1 0.1067
wait_bin   2.57  1 0.1090
GLOBAL    15.58  4 0.0036
```

Det verkar som att `reject` inte möter proportional hazards. Jag gör `reject` till en tidberoende variabel.

```r
cox_adjusted_td <- coxph(Surv(futime, fustat) ~ age_bin + surgery + wait_bin + reject + tt(re
                         data = transplanted,
                         tt = function(x, time, ...) x * time)
```

Jag utvärderar HR för variablerna i univariable modellerna samt i den justerade modellen.

```r
transplanted$age_bin <- ifelse(transplanted$age > 45, ">45", "<=45")
transplanted$wait_bin <- ifelse(transplanted$wait.time > 30, ">30", "<=30")
transplanted$age_bin   <- factor(transplanted$age_bin)
transplanted$surgery   <- factor(transplanted$surgery)
transplanted$wait_bin  <- factor(transplanted$wait_bin)
transplanted$reject    <- factor(transplanted$reject, levels = c("FALSE", "TRUE"))
```

```r
# Models
cox_age <- coxph(Surv(futime, fustat) ~ age_bin, data = transplanted)
cox_reject <- coxph(Surv(futime, fustat) ~ reject, data = transplanted)
cox_surgery <- coxph(Surv(futime, fustat) ~ surgery, data = transplanted)
cox_wait <- coxph(Surv(futime, fustat) ~ wait_bin, data = transplanted)

variables <- c("age_bin", "surgery", "wait_bin")

# Function to extract HR, CI, and p-value from Cox models
extract_unadjusted <- function(var) {
  formula <- as.formula(paste("Surv(futime, fustat) ~", var))
  model <- coxph(formula, data = transplanted)
  summary_model <- summary(model)
  coef <- summary_model$coefficients[1, ]
  conf_int <- summary_model$conf.int[1, c("lower .95", "upper .95")]
  hr <- coef["exp(coef)"]
  lower95 <- conf_int["lower .95"]
  upper95 <- conf_int["upper .95"]
  pvalue <- coef["Pr(>|z|)"]
  hr_ci <- paste0(round(hr, 2), " (", round(lower95, 2), ", ", round(upper95, 2), ")")
  data.frame(
    Variable = var,
    `Unadjusted HR (95% CI)` = hr_ci,
    `Unadjusted P-value` = round(pvalue, 2),
    stringsAsFactors = FALSE
  )
}

# Generate unadjusted results
unadjusted_results <- do.call(rbind, lapply(variables, extract_unadjusted))

adjusted_vars <- c("age_bin", "surgery", "wait_bin")
adjusted_formula <- as.formula(paste("Surv(futime, fustat) ~", paste(adjusted_vars, collapse
adjusted_model <- coxph(adjusted_formula, data = transplanted)

# Function to extract adjusted HRs
extract_adjusted <- function(var) {
  formula <- as.formula(paste("Surv(futime, fustat) ~", paste(adjusted_vars, collapse = " + "
  adjusted_model <- coxph(formula, data = transplanted)
  summary_model <- summary(adjusted_model)
  coef_names <- rownames(summary_model$coefficients)
  var_coefs <- coef_names[startsWith(coef_names, var)]
```

```
  result <- do.call(rbind, lapply(var_coefs, function(coef_name){
    coef_info <- summary_model$coefficients[coef_name, ]
    conf_int  <- summary_model$conf.int[coef_name, c("lower .95", "upper .95")]
    hr       <- coef_info["exp(coef)"]
    lower95  <- conf_int["lower .95"]
    upper95  <- conf_int["upper .95"]
    pvalue   <- coef_info["Pr(>|z|)"]
    hr_ci <- paste0(round(hr, 2), " (", round(lower95, 2), ", ", round(upper95, 2), ")")
    var_level <- sub(paste0("^", var), "", coef_name)
    var_display <- ifelse(var_level != "", paste0(var, var_level), var)
    data.frame(
      Variable           = var_display,
      `Adjusted HR (95% CI)` = hr_ci,
      `Adjusted P-value`    = round(pvalue, 2),
      stringsAsFactors       = FALSE
    )
  }))

  return(result)
}

# Generate adjusted results
adjusted_results <- do.call(rbind, lapply(adjusted_vars, extract_adjusted))


transplanted$reject_num <- ifelse(transplanted$reject == "TRUE", 1, 0)
adjusted_td_formula <- as.formula("Surv(futime, fustat) ~ age_bin + surgery + wait_bin + rej

# Fit the Cox model
adjusted_td_model <- coxph(
  adjusted_td_formula,
  data = transplanted,
  tt = function(x, time, ...) x * time
)
coef_td <- coef(adjusted_td_model)
cov_td  <- vcov(adjusted_td_model)
# Extract coefficients
coef_reject      <- coef_td["reject_num"]
coef_tt_reject   <- coef_td["tt(reject_num)"]
# Extract variances and covariance
var_reject      <- cov_td["reject_num", "reject_num"]
var_tt_reject   <- cov_td["tt(reject_num)", "tt(reject_num)"]
```

```r
cov_reject_tt   <- cov_td["reject_num", "tt(reject_num)"]
# Time points of interest
time_points <- c(180, 365, 730)
# Compute HRs at each time point
td_results <- do.call(rbind, lapply(time_points, function(t) {
  lp <- coef_reject + coef_tt_reject * t
  var_lp <- var_reject + (t^2) * var_tt_reject + 2 * t * cov_reject_tt
  se_lp  <- sqrt(var_lp)
  hr        <- exp(lp)
  lower95   <- exp(lp - 1.96 * se_lp)
  upper95   <- exp(lp + 1.96 * se_lp)
  # z-score and p-value
  z        <- lp / se_lp
  pvalue   <- 2 * (1 - pnorm(abs(z)))
  # Format HR and CI
  hr_ci <- paste0(round(hr, 2), " (", round(lower95, 2), ", ", round(upper95, 2), ")")
  data.frame(
    Variable               = paste0("Rejection (", t, " days)"),
    `Adjusted HR (95% CI)`  = hr_ci,
    `Adjusted P-value`     = round(pvalue, 2),
    `Unadjusted HR (95% CI)` = NA,
    `Unadjusted P-value`    = NA,
    stringsAsFactors        = FALSE
  )
}))

# Step 4: Combine All Results
variable_names_unadjusted <- c(
  "age_bin" = "Age",
  "surgery" = "Surgery",
  "wait_bin" = "Wait Time"
)

variable_names_adjusted <- c(
  "age_bin>45" = "Age",
  "surgeryTRUE" = "Surgery",
  "wait_bin>30" = "Wait Time"
)

unadjusted_results$Variable <- variable_names_unadjusted[unadjusted_results$Variable]
adjusted_results$Variable <- variable_names_adjusted[adjusted_results$Variable]
```

```r
combined_results <- merge(
  unadjusted_results,
  adjusted_results,
  by = "Variable",
  all = TRUE
)

td_results <- td_results[, c(
  "Variable",
  "Unadjusted.HR..95..CI.",
  "Unadjusted.P.value",
  "Adjusted.HR..95..CI.",
  "Adjusted.P.value"
)]

final_results <- rbind(combined_results, td_results)

final_results$Variable <- as.character(final_results$Variable)

desired_order <- c(
  "Age",
  "Surgery",
  "Wait Time",
  "Rejection",
  "Rejection (180 days)",
  "Rejection (365 days)",
  "Rejection (730 days)"
)
final_results$Variable <- factor(final_results$Variable, levels = desired_order)
final_results <- final_results[order(final_results$Variable), ]

rownames(final_results) <- NULL

colnames(final_results) <- c("Variable", "Unadjusted HR (95% CI)", "Unadjusted P-value",
                             "Adjusted HR (95% CI)", "Adjusted P-value")

kable(final_results,
      align = "c",
      row.names = FALSE,
      booktabs = TRUE) %>%
  kable_styling(
    font_size = 8,
```

```
    full_width = FALSE,
    position = "center",
    latex_options = c("scale_down")
  ) %>%
  row_spec(0, bold = TRUE) %>%
  column_spec(1, bold = TRUE) %>%
  row_spec(
    1:nrow(final_results),
    extra_css = "padding-top: 4px; padding-bottom: 4px;"
  )
```

```
Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
resized.
```

Variable	Unadjusted HR (95% CI)	Unadjusted P-value	Adjusted HR (95% CI)	Adjusted P-value
Age	1.54 (0.79, 2.99)	0.21	1.88 (0.94, 3.75)	0.07
Surgery	0.4 (0.17, 0.94)	0.04	0.4 (0.16, 0.97)	0.04
Wait Time	0.62 (0.34, 1.13)	0.12	0.66 (0.35, 1.24)	0.20
Rejection (180 days)	NA	NA	3.21 (1.54, 6.72)	0.00
Rejection (365 days)	NA	NA	6.41 (1.84, 22.26)	0.00
Rejection (730 days)	NA	NA	25.01 (1.6, 390.18)	0.02

Reflektion

I detta material har jag utforskat en dataset som beskriver överlevnad av patienter på väntelistan till Standford hjärttransplantationsprogram. Jag har utforskat datan, städat upp den, och utfört en inferentiell analys för att undersöka vilka variabler som predicerar död. Jag har använt mig av Kaplan-Meier kurvor för att undersöka överlevnad för surgery och reject, och sedan utfört en Cox regression för att undersöka hazard ratios för `age`, `surgery`, `wait time` och `reject`. Jag har även konstaterat att `reject` är en tidsberoende variabel.

Resultaten visar att `age` och `wait time` inte predicerar för död, med risk att `wait time` är en confounder för att patienten inte är tillräckligt sjuk att motivera snabb åtgärd. Vidare är `surgery` både ojusterat och justerat en skyddande faktor för död, med HR på 0.4 - detta skulle kunna tala för att de som genomgått en CABG innan har som transplantationsindikation ischemisk kardiomyopati i kontrast till övriga transplantationsindikationer och således har förbättrad mortalitet. Det behöver ej betyda att CABG pre-op är bra i sig.

Vidare är `reject` justerat för övriga variabler ovan en stark prediktor för död postoperativt vid 180, 365 och 720 dagar, med ökad HR över tid.

Avseende min tidigare hypotes att tidigare kirurgi är en prediktor för död, så visar resultaten att detta inte stämmer och det snarast skildrar lägre risk för död.