Project Basic EDA: ICU Catheter Analysis

Hillel, Yael and Matan

2025-05-03

Research Question & Notations

Question:

Among adult ICU stays for mechanically ventilated, hemodynamically stable patients with respiratory failure, what is the causal effect of receiving an arterial catheter within the first 24 hours of ICU admission (treatment) versus not receiving one on the risk of in-hospital death within 28 days (outcome)?

Notations:

Population: Adult ICU stays for patients with respiratory failure who were mechanically ventilated on day 1 and hemodynamically stable at admission.

Unit of analysis: Each ICU admission (icustay_id).

Time-zero: the first 24 hours of ICU admission.

Treatment: Arterial catheter placed within 24 hours of ICU admission (aline_flg = 1 means catheter in first day; 0 means none).

Outcome: In-hospital mortality within 28 days of ICU admission (day_28_flg: 1 = died, 0 = survived).

Dataset:

Name & Source: Subset of the MIMIC-II database, assembled for Chapter 16 of Secondary Analysis of EHR (Link)

Key variables available:

- Demographics & stay details: age, gender, service unit (FICU/MICU/SICU), admission day/hour.
- Severity scores: SAPS I (sapsi_first), SOFA (sofa_first).
- Vital signs & labs (first measurements): mean arterial pressure (map_1st), heart rate (hr_1st), SpO2 (spo2_1st), WBC, creatinine, etc.
- Comorbidity flags: sepsis, CHF, COPD, renal disease, liver disease, CAD, stroke, malignancy, respiratory disease.
- Lengths of stay: ICU LOS (icu_los_day), hospital LOS (hospital_los_day).
- Treatment flag: aline_flg (arterial line in first 24 h).
- Outcome flags: hosp_exp_flg, icu_exp_flg, day_28_flg, censoring indicators.

How treatment was allocated in practice:
Clinician decision based on monitoring needs and perceived severity.
To identify variables that may have influenced treatment allocation, we fitted a logistic regression model. The following covariates were selected by that model:

- renal_flg: Chronic renal disease (binary: 0 = no, 1 = yes)
- abg_count: Arterial blood gas count (number of tests, numeric)
- resp_flg: Respiratory disease (non-COPD) (binary: 0 = no, 1 = yes)
- stroke_flg: Stroke (binary: 0 = no, 1 = yes)
- afib_flg: Atrial fibrillation (binary: 0 = no, 1 = yes)
- liver_flg: Liver disease (binary: 0 = no, 1 = yes)
- copd_flg: Chronic obstructive pulmonary disease (binary: 0 = no, 1 = yes)
- mal_flg: Malignancy (binary: 0 = no, 1 = yes)
- sofa_first: First SOFA score (numeric)
- chloride_first: First chloride level (mEq/L, numeric)
- service_unit: Type of service unit (character: FICU, MICU, SICU)

Key potential confounders:
We identified confounders by taking the intersection of covariates included in both the treatment–allocation and outcome models:

- renal_flg: Chronic renal disease (binary: 0 = no, 1 = yes)
- resp_flg: Respiratory disease (non-COPD) (binary: 0 = no, 1 = yes)
- stroke_flg: Stroke (binary: 0 = no, 1 = yes)
- afib_flg: Atrial fibrillation (binary: 0 = no, 1 = yes)
- liver_flg: Liver disease (binary: 0 = no, 1 = yes)
- copd_flg: Chronic obstructive pulmonary disease (binary: 0 = no, 1 = yes)
- mal_flg: Malignancy (binary: 0 = no, 1 = yes)
- sofa_first: First SOFA score (numeric)
- service_unit: Type of service unit (character: FICU, MICU, SICU)

Potential effect-modifiers:
To detect variables that modify the treatment effect, we fit outcome models including interaction terms between each covariate and the treatment indicator:

- cad_flg: Coronary artery disease (binary: 0 = no, 1 = yes)
- stroke_flg: Stroke (binary: 0 = no, 1 = yes)
- copd_flg: Chronic obstructive pulmonary disease (binary: 0 = no, 1 = yes)
- hospital_los_day: Length of hospital stay (days, numeric)
- afib_flg: Atrial fibrillation (binary: 0 = no, 1 = yes)
- icu_los_day: Length of stay in ICU (days, numeric)
- gender_num: Patient gender (1 = male, 0 = female)
- resp_flg: Respiratory disease (non-COPD) (binary: 0 = no, 1 = yes)
- mal_flg: Malignancy (binary: 0 = no, 1 = yes)
- service_unit: Type of service unit (character: FICU, MICU, SICU)

EDA for Detection of Confounders & Effect Modifiers

```
df <- read.csv("../data/full_cohort_data.csv")
skim(df)
```

Table 1: Data summary

Name	df
Number of rows	1776
Number of columns	46
Column type frequency:	
character	2
numeric	44
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
service_unit	0	1	4	4	0	3	0
day_icu_intime	0	1	9	9	0	7	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
aline_flg	0	1.00	0.55	0.50	0.00	0.00	1.00	1.00	1.00	
icu_los_day	0	1.00	3.35	3.36	0.50	1.37	2.18	4.00	28.24	
hospital_los_day	0	1.00	8.11	8.16	1.00	3.00	6.00	10.00	112.00	
age	0	1.00	54.38	21.06	15.18	38.25	53.68	72.76	99.11	
gender_num	1	1.00	0.58	0.49	0.00	0.00	1.00	1.00	1.00	
weight_first	110	0.94	80.08	22.49	30.00	65.40	77.00	90.00	257.60	
bmi	466	0.74	27.83	8.21	12.78	22.62	26.32	30.80	98.80	
sapsi_first	85	0.95	14.14	4.11	3.00	11.00	14.00	17.00	32.00	
sofa_first	6	1.00	5.82	2.33	0.00	4.00	6.00	7.00	17.00	
service_num	0	1.00	0.55	0.50	0.00	0.00	1.00	1.00	1.00	
day_icu_intime_num	0	1.00	4.05	1.99	1.00	2.00	4.00	6.00	7.00	
hour_icu_intime	0	1.00	10.59	7.92	0.00	3.00	9.00	19.00	23.00	
hosp_exp_flg	0	1.00	0.14	0.34	0.00	0.00	0.00	0.00	1.00	
icu_exp_flg	0	1.00	0.10	0.29	0.00	0.00	0.00	0.00	1.00	
day_28_flg	0	1.00	0.16	0.37	0.00	0.00	0.00	0.00	1.00	
mort_day_censored	0	1.00	614.33	403.11	0.00	434.32	731.00	731.00	3094.08	
censor_flg	0	1.00	0.72	0.45	0.00	0.00	1.00	1.00	1.00	
sepsis_flg	0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
chf_flg	0	1.00	0.12	0.32	0.00	0.00	0.00	0.00	1.00	
afib_flg	0	1.00	0.12	0.32	0.00	0.00	0.00	0.00	1.00	
renal_flg	0	1.00	0.03	0.18	0.00	0.00	0.00	0.00	1.00	
liver_flg	0	1.00	0.06	0.23	0.00	0.00	0.00	0.00	1.00	
copd_flg	0	1.00	0.09	0.28	0.00	0.00	0.00	0.00	1.00	
cad_flg	0	1.00	0.07	0.25	0.00	0.00	0.00	0.00	1.00	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
stroke_flg	0	1.00	0.12	0.33	0.00	0.00	0.00	0.00	1.00	
mal_flg	0	1.00	0.14	0.35	0.00	0.00	0.00	0.00	1.00	
resp_flg	0	1.00	0.32	0.47	0.00	0.00	0.00	1.00	1.00	
map_1st	0	1.00	88.25	17.60	5.00	76.67	87.00	99.00	195.00	
hr_1st	0	1.00	87.91	18.76	30.00	74.75	87.00	100.00	158.00	
temp_1st	3	1.00	97.79	4.54	32.00	96.90	98.10	99.30	104.80	
spo2_1st	0	1.00	98.43	5.51	4.00	98.00	100.00	100.00	100.00	
abg_count	0	1.00	5.98	8.68	0.00	1.00	3.00	7.00	115.00	
wbc_first	8	1.00	12.32	6.60	0.17	8.20	11.30	15.00	109.80	
hgb_first	8	1.00	12.55	2.20	2.00	11.10	12.70	14.12	19.00	
platelet_first	8	1.00	246.08	99.87	7.00	182.00	239.00	297.00	988.00	
sodium_first	5	1.00	139.56	4.73	105.00	137.00	140.00	142.00	165.00	
potassium_first	5	1.00	4.11	0.79	1.90	3.60	4.00	4.40	9.80	
tco2_first	5	1.00	24.42	4.99	2.00	22.00	24.00	27.00	62.00	
chloride_first	5	1.00	103.84	5.73	78.00	101.00	104.00	107.00	133.00	
bun_first	5	1.00	19.28	14.37	2.00	11.00	15.00	22.00	139.00	
creatinine_first	6	1.00	1.10	1.08	0.00	0.70	0.90	1.10	18.30	
po2_first	186	0.90	227.62	144.86	22.00	108.00	195.00	323.00	634.00	
pco2_first	186	0.90	43.41	13.98	8.00	36.00	41.00	47.00	158.00	
iv_day_1	143	0.92	1622.91	1677.13	0.00	329.75	1081.53	2493.90	13910.00	

Treatment & Outcome Distribution

```
df %>%
  count(aline_flg) %>%
  mutate(pct = n / sum(n) * 100) %>%
  rename(Catheter = aline_flg)
```

```
##   Catheter   n      pct
## 1        0 792 44.59459
## 2        1 984 55.40541
```

```
df %>%
  count(day_28_flg) %>%
  mutate(pct = n / sum(n) * 100) %>%
  rename(Death28d = day_28_flg)
```

```
##   Death28d    n      pct
## 1        0 1493 84.06532
## 2        1  283 15.93468
```

Missing Values Analysis & Handling

```
# Percentage of rows with at least one missing value
total_rows        <- nrow(df)
rows_with_missing <- sum(!complete.cases(df))
pct_rows_missing  <- rows_with_missing / total_rows * 100
cat(sprintf("Rows with  1 missing: %d / %d (%.2f%%)\n",
            rows_with_missing, total_rows, pct_rows_missing))
```

```
## Rows with  1 missing: 712 / 1776 (40.09%)
```

```r
# Percentage of missing values per column
col_pct_missing <- sapply(df, function(col) mean(is.na(col)) * 100)

missing_summary <- data.frame(
  variable    = names(col_pct_missing),
  pct_missing = col_pct_missing,
  row.names   = NULL
)

print(missing_summary)
```

```
##              variable pct_missing
## 1           aline_flg  0.00000000
## 2         icu_los_day  0.00000000
## 3    hospital_los_day  0.00000000
## 4                 age  0.00000000
## 5          gender_num  0.05630631
## 6        weight_first  6.19369369
## 7                 bmi 26.23873874
## 8         sapsi_first  4.78603604
## 9          sofa_first  0.33783784
## 10       service_unit  0.00000000
## 11        service_num  0.00000000
## 12     day_icu_intime  0.00000000
## 13 day_icu_intime_num  0.00000000
## 14    hour_icu_intime  0.00000000
## 15       hosp_exp_flg  0.00000000
## 16        icu_exp_flg  0.00000000
## 17         day_28_flg  0.00000000
## 18  mort_day_censored  0.00000000
## 19         censor_flg  0.00000000
## 20         sepsis_flg  0.00000000
## 21            chf_flg  0.00000000
## 22           afib_flg  0.00000000
## 23          renal_flg  0.00000000
## 24          liver_flg  0.00000000
## 25           copd_flg  0.00000000
## 26            cad_flg  0.00000000
## 27         stroke_flg  0.00000000
## 28            mal_flg  0.00000000
## 29           resp_flg  0.00000000
## 30            map_1st  0.00000000
## 31             hr_1st  0.00000000
## 32           temp_1st  0.16891892
## 33           spo2_1st  0.00000000
## 34          abg_count  0.00000000
## 35          wbc_first  0.45045045
## 36          hgb_first  0.45045045
## 37     platelet_first  0.45045045
## 38       sodium_first  0.28153153
## 39    potassium_first  0.28153153
## 40         tco2_first  0.28153153
## 41     chloride_first  0.28153153
## 42          bun_first  0.28153153
## 43   creatinine_first  0.33783784
## 44          po2_first 10.47297297
## 45         pco2_first 10.47297297
## 46           iv_day_1  8.05180180
```

```
# We remove the bmi variable from the df
df <- df %>% select(-bmi)
```

Fitting Models with Lasso to Find Key Covariates

Fitting a logistic regression model to identify covariates that predict treatment

```
# Prepare data: convert character columns to factors, drop missing
df_lasso <- df %>%
  mutate(across(where(is.character), as.factor)) %>%
  drop_na()

# Define response vector y and design matrix X,
# Excluding outcome and day-of-week/time columns
y <- df_lasso$aline_flg
X <- model.matrix(
  aline_flg ~ .
    - day_28_flg
    - day_icu_intime
    - day_icu_intime_num
    - icu_exp_flg
    - icu_los_day
    - hour_icu_intime,
  data = df_lasso
)[, -1]

# Cross-validated LASSO logistic regression
set.seed(123)
cvfit <- cv.glmnet(
  x           = X,
  y           = y,
  family      = "binomial",
  alpha       = 1,
  standardize = TRUE
)

# Fit final model at the optimal
lambda_min <- cvfit$lambda.min
lasso_mod   <- glmnet(
  x           = X,
  y           = y,
  family      = "binomial",
  alpha       = 1,
  lambda      = lambda_min,
  standardize = TRUE
)

# Extract coefficients into a data frame
coef_df <- as.matrix(coef(lasso_mod)) %>%
  as.data.frame() %>%
  rownames_to_column("term")

# Rename the second column to "estimate"
names(coef_df)[2] <- "estimate"

# Filter on absolute value > 0.01 and sort by magnitude
coef_df <- coef_df %>%
  filter(term != "(Intercept)", abs(estimate) > 0.05) %>%
  arrange(desc(abs(estimate)))

treatment_predictors <- coef_df$term

# Print nicely
kable(coef_df, digits = 3, caption = "LASSO Coefficients (|estimate| > 0.05) for Treatment Model")
```

Table 4: LASSO Coefficients (|estimate| > 0.05) for Treatment Model

term	estimate
service_unitSICU	0.984
renal_flg	0.930
abg_count	0.803
service_unitMICU	-0.772
resp_flg	-0.756
stroke_flg	0.308
afib_flg	-0.218
liver_flg	-0.166
copd_flg	-0.164
mal_flg	0.157
sofa_first	0.080
chloride_first	0.062

Fitting a logistic regression model to identify covariates that predict outcome using interactions, in order to find potential effect modifiers.

```
# Prepare data: convert character columns to factors, drop missing
df_mod <- df %>%
  mutate(across(where(is.character), as.factor)) %>%
  drop_na()

# Build formula including main effects and treatment interactions
# Exclude post-treatment/time variables
excluded <- c(
  "hosp_exp_flg", "day_icu_intime", "day_icu_intime_num",
  "icu_exp_flg", "mort_day_censored", "censor_flg",
  "hour_icu_intime"
)
# All covariates except outcome and excluded
covs <- setdiff(names(df_mod), c("day_28_flg", excluded))
# Remove treatment from covariates list to add it separately
other_covs <- setdiff(covs, "aline_flg")

# Construct formula: main effects + interactions with treatment
```

```r
form_str <- paste0(
  "day_28_flg ~ aline_flg + ",
  paste(other_covs, collapse = " + "), " + ",
  paste0("aline_flg:", other_covs, collapse = " + ")
)
formula_inter <- as.formula(form_str)

# Define response vector y and design matrix X
y <- df_mod$day_28_flg
X <- model.matrix(formula_inter, data = df_mod)[, -1]

# Cross-validated LASSO logistic regression
set.seed(123)
cvfit_outcome <- cv.glmnet(
  x           = X,
  y           = y,
  family      = "binomial",
  alpha       = 1,
  standardize = TRUE
)

# Fit final model at the optimal
lambda_min_outcome <- cvfit_outcome$lambda.min
lasso_outcome <- glmnet(
  x           = X,
  y           = y,
  family      = "binomial",
  alpha       = 1,
  lambda      = lambda_min_outcome,
  standardize = TRUE
)

# Extract coefficients into a data frame
coef_df_outcome <- as.matrix(coef(lasso_outcome)) %>%
  as.data.frame() %>%
  rownames_to_column("term")
names(coef_df_outcome)[2] <- "estimate"

# Filter on |estimate| > 0.05 and sort by magnitude
coef_df_outcome <- coef_df_outcome %>%
  filter(term != "(Intercept)", abs(estimate) > 0.05) %>%
  arrange(desc(abs(estimate)))

# Save selected terms as potential effect modifiers
outcome_predictors <- coef_df_outcome$term

potential_effect_modifiers <- coef_df_outcome %>%
  filter(grepl(":", term)) %>%
  pull(term)

# Print nicely
kable(
  potential_effect_modifiers,
  digits  = 3,
  caption = "LASSO Coefficients (|estimate| > 0.05) for 28-Day Mortality Model with Treatment Interacti
)
```

Table 5: LASSO Coefficients (|estimate| > 0.05) for 28-Day Mortality Model with Treatment Interactions

x
aline_flg:cad_flg
aline_flg:stroke_flg
aline_flg:service_unitMICU
aline_flg:copd_flg
aline_flg:hospital_los_day
aline_flg:afib_flg
aline_flg:icu_los_day
aline_flg:gender_num
aline_flg:resp_flg
aline_flg:mal_flg

```
potential_effect_modifiers <- sub("^.*:", "", potential_effect_modifiers)
```

Finding potential con-founders based on the intersection between the variables that "survived" the lasso fit of the treatment prediction and predicting the outcomes.

```
# 1. Compute intersection
outcome_predictors <- union(outcome_predictors,potential_effect_modifiers)
potential_confounders <- intersect(treatment_predictors, outcome_predictors)

# 2. Create a tidy data frame
shared_df <- tibble(
  term = potential_confounders
)

# 3. Print nicely with a caption
kable(
  shared_df,
)
```

term
renal_flg
service_unitMICU
resp_flg
stroke_flg
afib_flg
liver_flg
copd_flg
mal_flg
sofa_first

```
potential_confounders <- potential_confounders[!potential_confounders %in% c("service_unitMICU", "servi
potential_confounders <- c(potential_confounders, "service_unit")
```

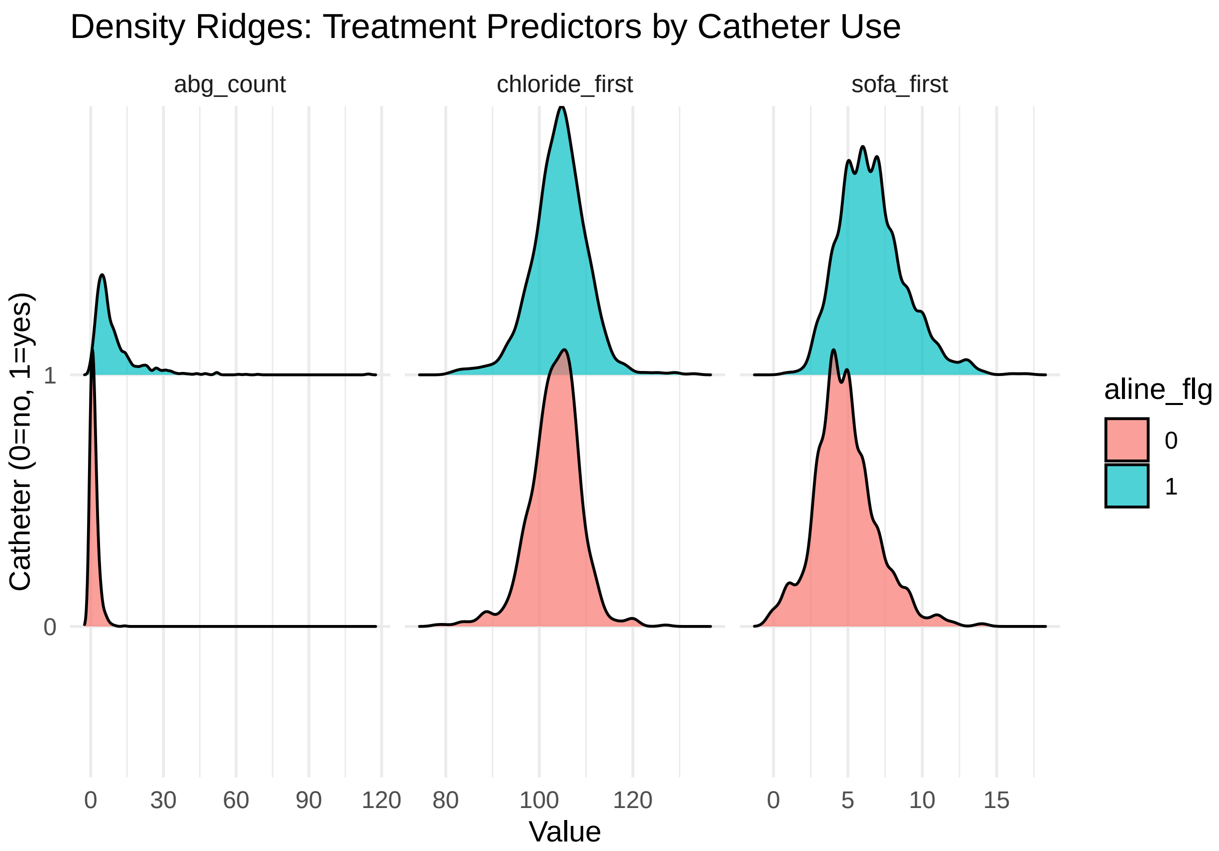
```
potential_effect_modifiers <- potential_effect_modifiers[!potential_effect_modifiers %in% c("service_uni
potential_effect_modifiers <- c(potential_effect_modifiers, "service_unit")

treatment_predictors <- treatment_predictors[!treatment_predictors %in% c("service_unitMICU", "service_
treatment_predictors <- c(treatment_predictors, "service_unit")
```

Continuous Covariates Distributions

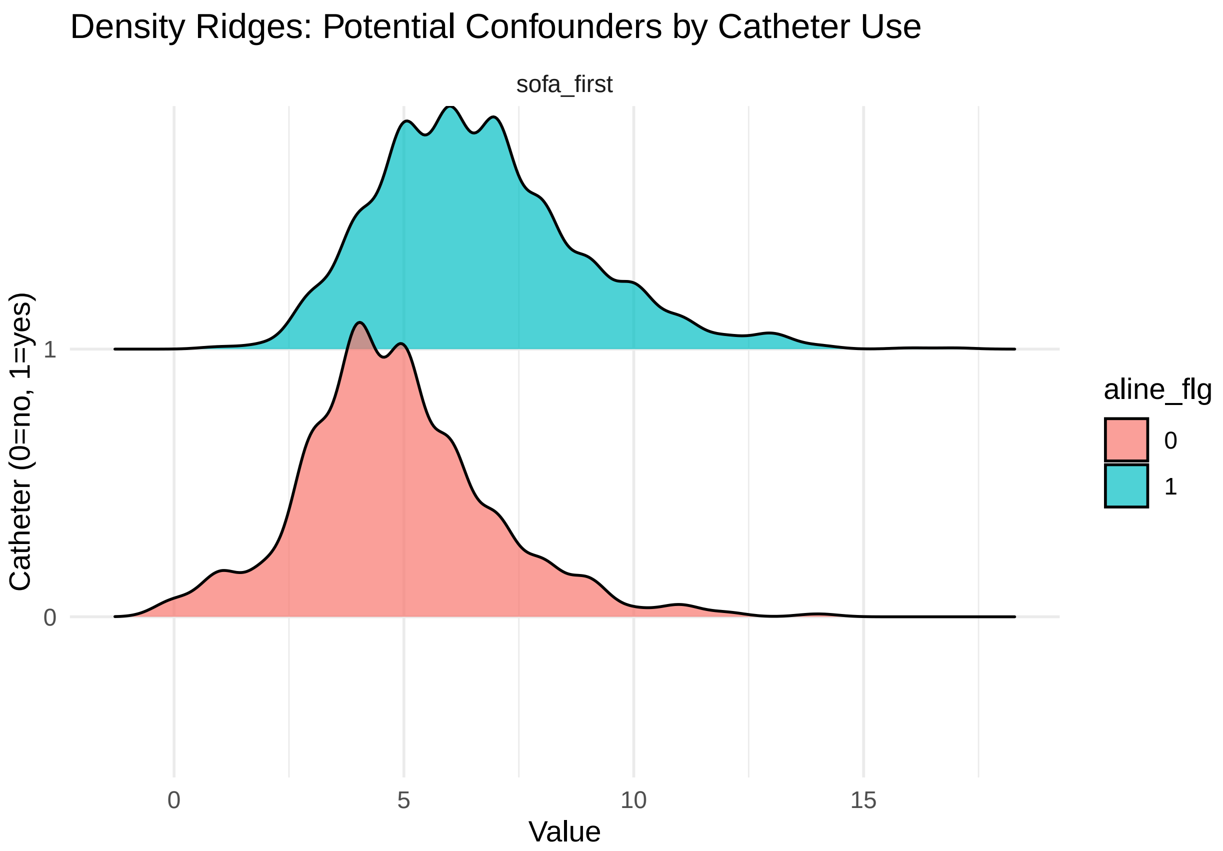
```
# Split a vector of variable names into continuous vs. categorical/binary
split_covs <- function(var_list, df) {
  continuous <- var_list[sapply(var_list, function(v) {
    col <- df[[v]]
    is.numeric(col) && length(unique(na.omit(col))) > 2
  })]
  categorical <- var_list[sapply(var_list, function(v) {
    col <- df[[v]]
    is.factor(col) ||
    is.character(col) ||
    (is.numeric(col) && length(unique(na.omit(col))) <= 2)
  })]
  list(
    continuous = continuous,
    categorical = categorical
  )
}

# Apply to treatment predictors
tp_split <- split_covs(treatment_predictors, df)
treatment_predictors_continuous  <- tp_split$continuous
treatment_predictors_categorical <- tp_split$categorical

# Apply to potential effect modifiers
em_split <- split_covs(potential_effect_modifiers, df)
potential_effect_modifiers_continuous  <- em_split$continuous
potential_effect_modifiers_categorical <- em_split$categorical

# Apply split_covs to potential_confounders
pc_split <- split_covs(potential_confounders, df)

# Extract continuous vs. categorical lists
potential_confounders_continuous  <- pc_split$continuous
potential_confounders_categorical <- pc_split$categorical
```

```
# Helper function to plot density ridges if there are continuous vars
plot_density_ridges <- function(data, group_var, cont_vars, title, y_label) {
  if (length(cont_vars) == 0) {
    message("No continuous variables to plot for: ", title)
    return(invisible(NULL))
  }
  # Select the grouping column and continuous vars
  df_sub <- data %>% select(all_of(c(group_var, cont_vars)))
  # Only proceed if there is at least one continuous column
  if (ncol(df_sub) < 2) {
```

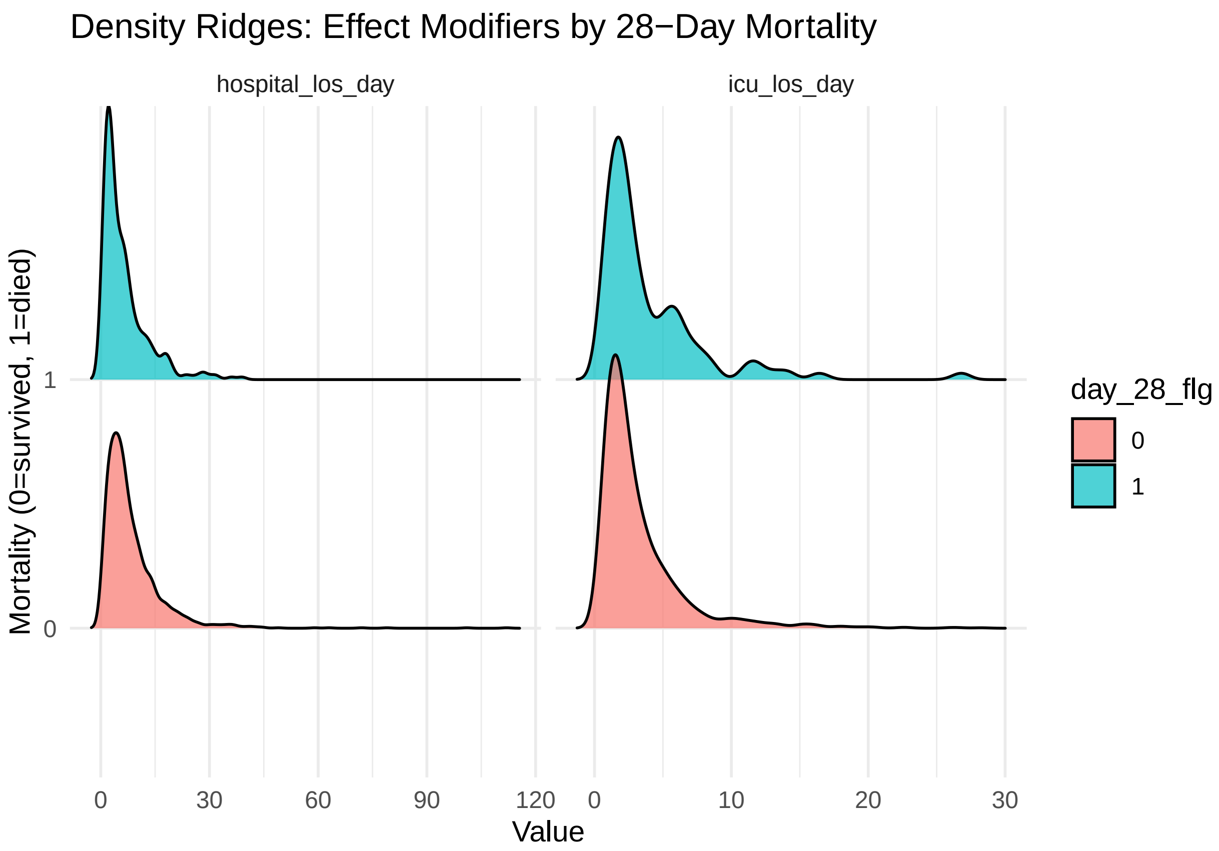
```r
    message("Not enough columns to pivot for: ", title)
    return(invisible(NULL))
  }
  df_sub %>%
    pivot_longer(
      cols      = -all_of(group_var),
      names_to  = "Variable",
      values_to = "Value"
    ) %>%
    ggplot(aes(x = Value, y = factor(.data[[group_var]]), fill = factor(.data[[group_var]]))) +
      geom_density_ridges(alpha = 0.7, scale = 1.1) +
      facet_wrap(~ Variable, scales = "free_x", ncol = 3) +
      labs(
        title = title,
        y     = y_label,
        fill  = group_var
      ) +
      theme_minimal()
}

# 1) Treatment predictors by catheter use
plot_density_ridges(
  data       = df,
  group_var  = "aline_flg",
  cont_vars  = treatment_predictors_continuous,
  title      = "Density Ridges: Treatment Predictors by Catheter Use",
  y_label    = "Catheter (0=no, 1=yes)"
)
```



```r
# 2) Potential confounders by catheter use
plot_density_ridges(
  data       = df,
  group_var  = "aline_flg",
  cont_vars  = potential_confounders_continuous,
  title      = "Density Ridges: Potential Confounders by Catheter Use",
  y_label    = "Catheter (0=no, 1=yes)"
)
```

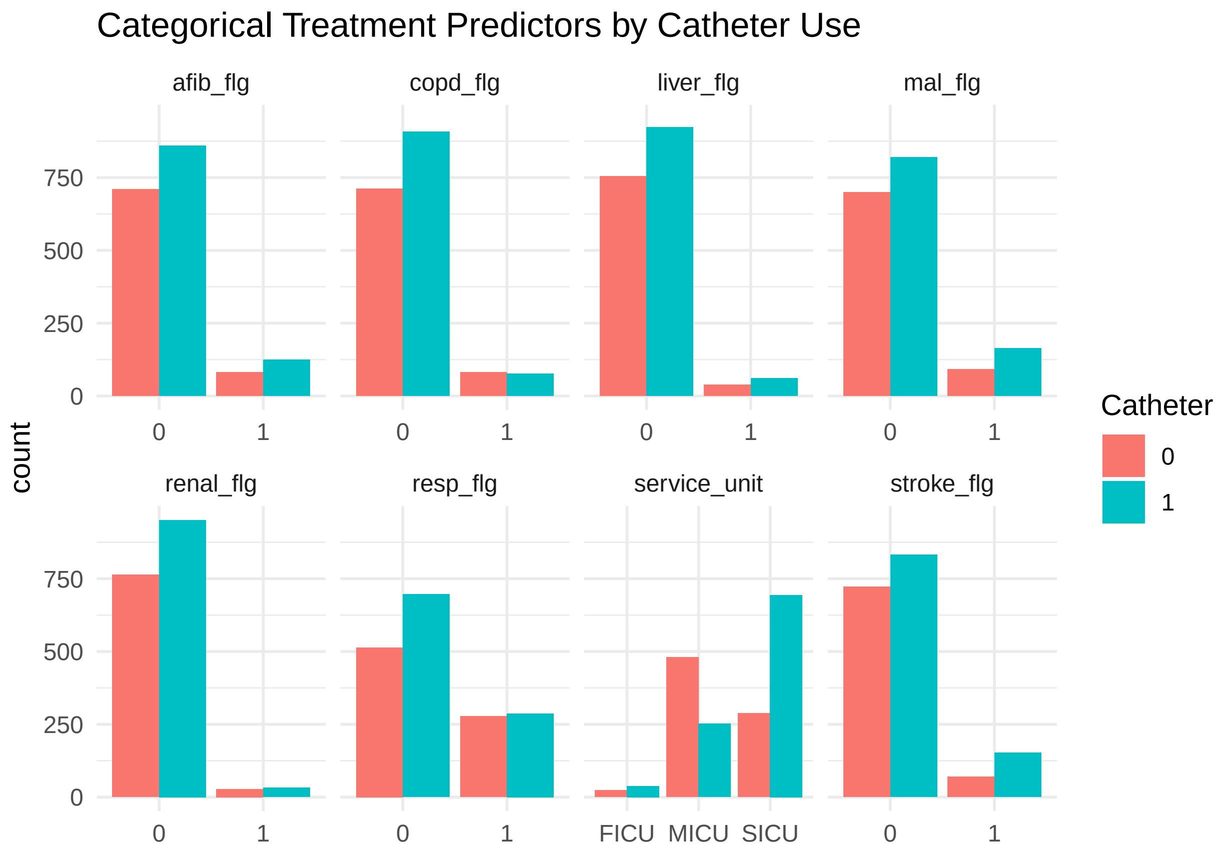


```
# 3) Potential effect modifiers by 28-day mortality
plot_density_ridges(
  data       = df,
  group_var  = "day_28_flg",
  cont_vars  = potential_effect_modifiers_continuous,
  title      = "Density Ridges: Effect Modifiers by 28-Day Mortality",
  y_label    = "Mortality (0=survived, 1=died)"
)
```



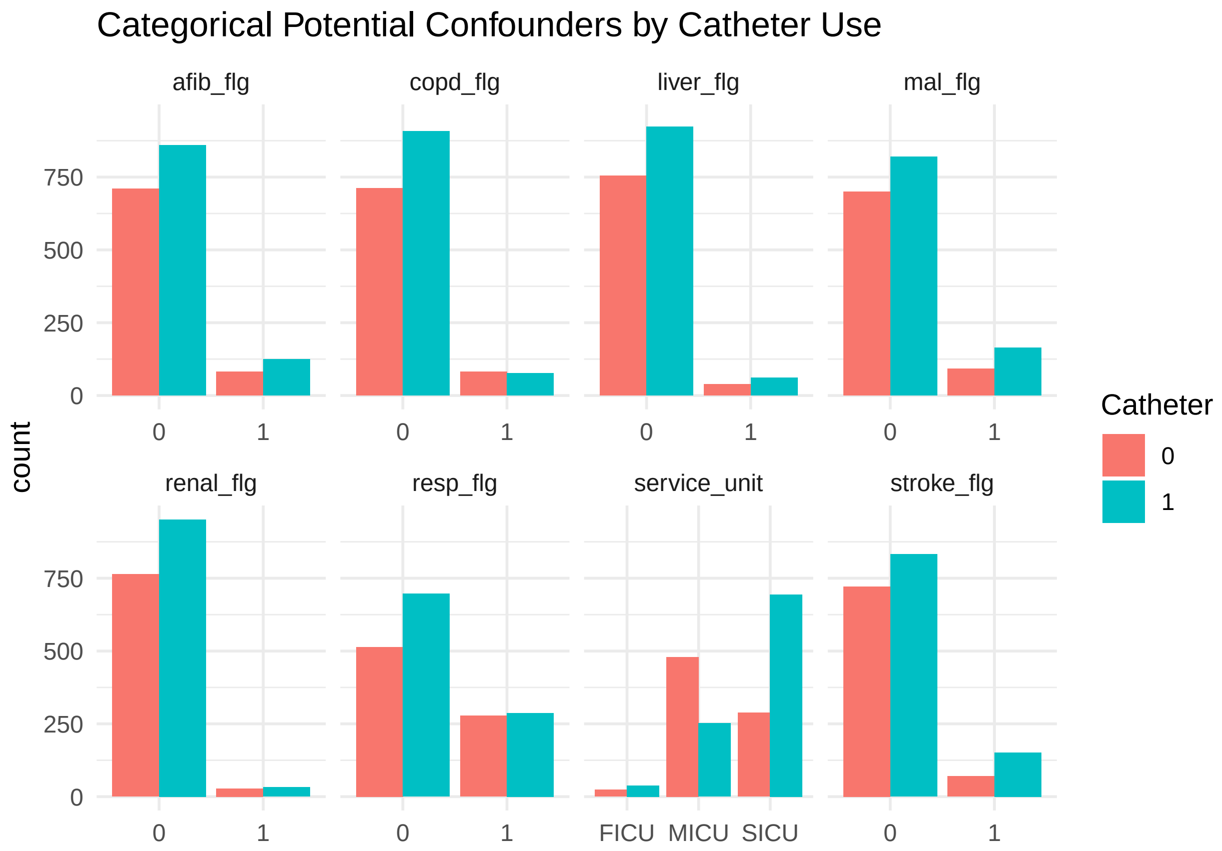
Categorical Covariates Histograms

```
# Helper to plot bar charts for categorical/binary vars,
# coercing all selected vars to character so pivot_longer works
plot_cat_histograms <- function(data, group_var, cat_vars, title, legend_title) {
  if (length(cat_vars) == 0) {
    message("No categorical variables to plot for: ", title)
    return(invisible(NULL))
  }
  df_sub <- data %>%
    # coerce all categorical vars to character to avoid type conflicts
    mutate(across(all_of(cat_vars), as.character)) %>%
    select(all_of(c(group_var, cat_vars)))

  df_sub %>%
    pivot_longer(
      cols      = -all_of(group_var),
      names_to  = "Variable",
      values_to = "Value"
    ) %>%
    mutate(
      Group = factor(.data[[group_var]]),
      Value = factor(Value)
    ) %>%
    ggplot(aes(x = Value, fill = Group)) +
      geom_bar(position = "dodge") +
      facet_wrap(~ Variable, scales = "free_x", ncol = 4) +
      labs(
        title = title,
        x     = NULL,
        fill  = legend_title
      ) +
      theme_minimal()
}

# 1) Categorical Treatment Predictors by Catheter Use
plot_cat_histograms(
  data         = df,
  group_var    = "aline_flg",
  cat_vars     = treatment_predictors_categorical,
  title        = "Categorical Treatment Predictors by Catheter Use",
  legend_title= "Catheter"
)
```

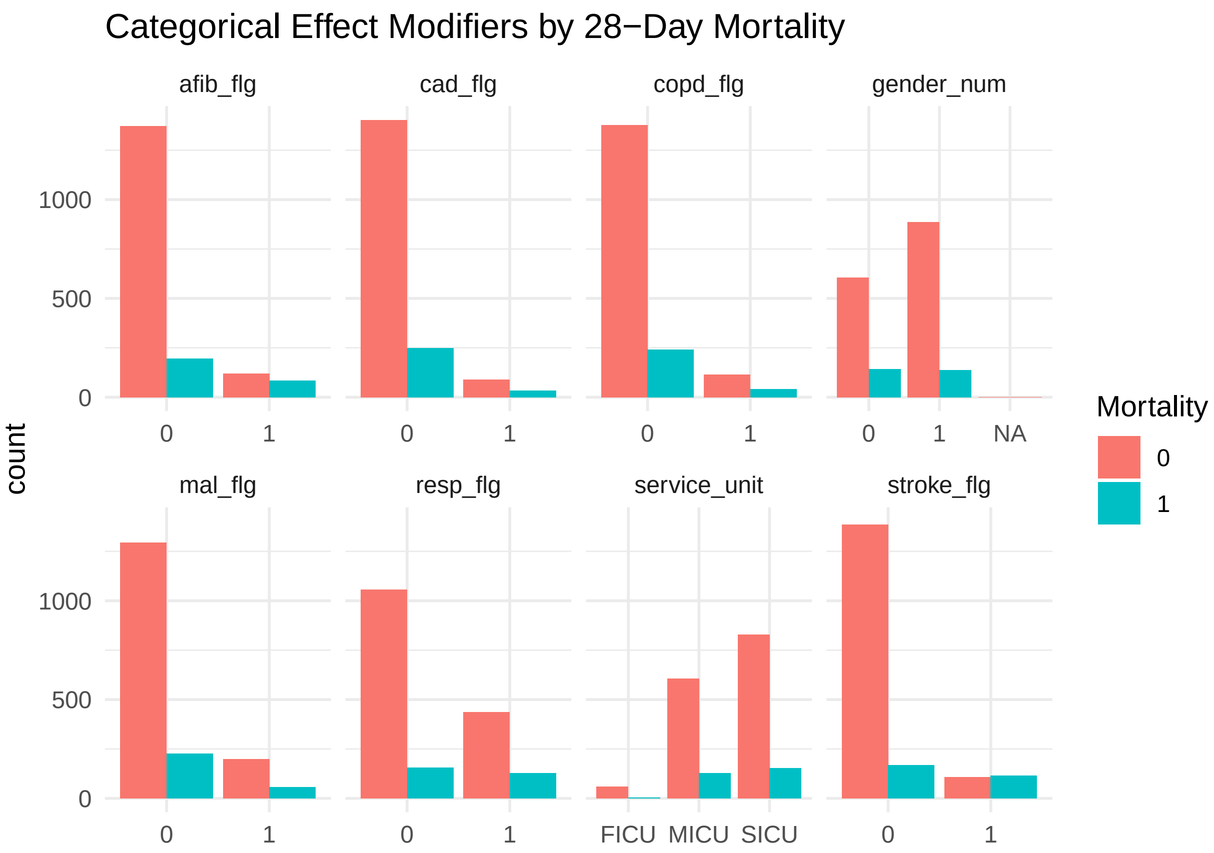


```
# 2) Categorical Potential Confounders by Catheter Use
plot_cat_histograms(
  data         = df,
  group_var    = "aline_flg",
  cat_vars     = potential_confounders_categorical,
  title        = "Categorical Potential Confounders by Catheter Use",
  legend_title= "Catheter"
```

```
)
```



```
# 3) Categorical Potential Effect Modifiers by 28-Day Mortality
plot_cat_histograms(
  data         = df,
  group_var    = "day_28_flg",
  cat_vars     = potential_effect_modifiers_categorical,
  title        = "Categorical Effect Modifiers by 28-Day Mortality",
  legend_title= "Mortality"
)
```


Categorical Effect Modifiers by 28−Day Mortality
afib_flg
cad_flg
copd_flg
gender_num
mal_flg
resp_flg
service_unit
stroke_flg
count
1000
500
0
0
1
NA
FICU
MICU
SICU
Mortality
0
1