SPPH604: Statistical Analysis Plan Analytic Code

Teal Green (Alexi, Alex, Ezra)

2025-09-28

Statistical Analysis Plan

This is an R Markdown document of the reproducible code for the statistical analysis plan from Teal Green for SPPH604.

Our research question is: Among U.S. adults aged 45 years or above in NHANES 2003–2018, are levels of polypharmacy associated with all-cause mortality?

PICOT Elements

Population (P): U.S. adults aged 45 years and up from the 2003–2018 NHANES cycles.

Intervention/Exposure (I): Levels of polypharmacy categorized by number of medications reported at their interview: low polypharmacy (1 to 4 medications reported), polypharmacy (5 to 9), hyperpolypharmacy (≥10)

Comparator (C): No polypharmacy (0)

Outcome (O): Risk of all-cause mortality time from NHANES interview to all-cause death or censoring at the end of follow-up

Timeframe (T): Up to 15 years of follow-up through NHANES-linked mortality files.

Confounders: sex RIAGENDR, race/ethnicity RIDRETH1/ RIDRETH3, age RIDAGEYR, survey cycle indicator SDDSRVYR, education DMDEDUC2, income-to-poverty ratio INDFMPIR, MCQ (comorbidity count/index), insurance HIQ011

Load packages and define NHANES cycles

```
# Load packages
library(pacman)
pacman::p_load(
  tidyverse, nhanesA, janitor, haven, readr, stringr,
  survival, survey, gtsummary, gt, broom, splines
)

options(survey.lonely.psu = "adjust")  # reasonable default for single-PSU strata

# Define 2-year NHANES cycles used
cycle_meta <- tibble::tribble(
```

```
  ~cycle,        ~suffix,
  "2003-2004",   "_C",
  "2005-2006",   "_D",
  "2007-2008",   "_E",
  "2009-2010",   "_F",
  "2011-2012",   "_G",
  "2013-2014",   "_H",
  "2015-2016",   "_I",
  "2017-2018",   "_J"
)
```

Import NHANES linked mortality dataset from 2003-2018

```
# List all mortality .dat files (already downloaded to data/mortality/)
files <- list.files("data/mortality", full.names = TRUE, pattern = "\\.dat$",
ignore.case = TRUE)

# Fixed-width spec for public-use mortality files
spec <- fwf_cols(
  SEQN         = c(1, 6),
  eligstat     = c(15, 15),
  mortstat     = c(16, 16),
  ucod_leading = c(17, 19),
  diabetes     = c(20, 20),
  hyperten     = c(21, 21),
  permth_int   = c(43, 45),
  permth_exm   = c(46, 48)
)

# Extract cycle label from filename
get_cycle <- function(path) {
  fn <- basename(path)
  m  <- stringr::str_match(fn, "NHANES_(\\d{4})_(\\d{4})")
  if (is.na(m[1,2])) NA_character_ else paste0(m[1,2], "/", m[1,3])
}

# Reader for one mortality file
read_one <- function(path) {
  cy <- get_cycle(path)
  readr::read_fwf(
    file          = path,
    col_positions = spec,
    col_types     = "iiiiiiii",
    na            = c("", ".")
  ) %>%
    dplyr::mutate(cycle = cy, .before = 1)
}

# Combine all mortality files and keep linkage-eligible participants
nhanes_mortality <- purrr::map_dfr(files, read_one) %>%
```

```
  dplyr::filter(eligstat == 1)

# Quick peek
glimpse(nhanes_mortality)

## Rows: 47,632
## Columns: 9
## $ cycle        <chr> "2003/2004", "2003/2004", "2003/2004", "2003/2004",
"2003…
## $ SEQN         <int> 21005, 21009, 21010, 21012, 21015, 21017, 21018,
21019, 2…
## $ eligstat     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, …
## $ mortstat     <int> 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
1, …
## $ ucod_leading <int> NA, NA, NA, 1, 1, NA, NA, NA, NA, 1, NA, NA, NA, NA,
NA, …
## $ diabetes     <int> NA, NA, NA, 0, 0, NA, NA, NA, NA, 1, NA, NA, NA, NA,
NA, …
## $ hyperten     <int> NA, NA, NA, 0, 0, NA, NA, NA, NA, 0, NA, NA, NA, NA,
NA, …
## $ permth_int   <int> 201, 196, 182, 127, 25, 202, 183, 180, 183, 20, 196,
203,…
## $ permth_exm   <int> 201, 195, 181, 126, 24, 201, 182, 179, 182, NA, 195,
201,…
```

Helper functions for recoding NHANES data

```
# Clean numeric from char-like NHANES fields
to_num <- function(x) {
  x <- trimws(as.character(x))
  x[x %in% c("", ".", "..", "...")] <- NA_character_
  as.numeric(x)
}

# Sex (RIAGENDR)
recode_sex <- function(x) {
  xn <- suppressWarnings(as.numeric(x))
  out <- dplyr::case_when(xn == 1 ~ "Male", xn == 2 ~ "Female", TRUE ~
NA_character_)
  factor(out, levels = c("Male","Female"))
}

# Race/ethnicity; use RIDRETH3 when present (2011–18)
derive_race_eth <- function(df) {
  ridreth1 <- suppressWarnings(as.numeric(df$ridreth1))
  ridreth3 <- suppressWarnings(as.numeric(df$ridreth3))
  map_r1 <- c(`1`="Mexican American", `2`="Other Hispanic",
              `3`="Non-Hispanic White", `4`="Non-Hispanic Black",
`5`="Other/Multi")
```

```r
  map_r3 <- c(`1`="Non-Hispanic White", `2`="Non-Hispanic Black",
              `3`="Non-Hispanic Asian", `4`="Other/Multi",
              `6`="Mexican American", `7`="Other Hispanic")
  use3 <- !is.na(ridreth3)
  out <- ifelse(use3, unname(map_r3[as.character(ridreth3)]),
                unname(map_r1[as.character(ridreth1)]))
  factor(out, levels = c("Non-Hispanic White","Non-Hispanic Black",
                         "Non-Hispanic Asian","Mexican American",
                         "Other Hispanic","Other/Multi"))
}

# Education (DMDEDUC2) — 4 levels + "Unknown"
derive_ed_cat <- function(x) {
  v <- suppressWarnings(as.numeric(x))
  out <- dplyr::case_when(
    v %in% c(1, 2) ~ "Less than HS",
    v == 3         ~ "HS/GED",
    v == 4         ~ "Some college",
    v == 5         ~ "College+",
    v %in% c(7, 9) ~ "Unknown",
    TRUE           ~ "Unknown"
  )
  factor(out, levels = c("Less than HS","HS/GED","Some
college","College+","Unknown"))
}

# Insurance HIQ011 robust recode -> 1/0/NA, then map to Yes/No/Unknown later
recode_hiq011_yesno <- function(x) {
  xs <- trimws(as.character(x))
  xs[xs %in% c("", ".", "..", "...")] <- NA_character_
  lead_num <- suppressWarnings(as.numeric(stringr::str_extract(xs, "^[0-
9]+")))
  out_num <- dplyr::case_when(
    !is.na(lead_num) & lead_num == 1 ~ 1L,
    !is.na(lead_num) & lead_num == 2 ~ 0L,
    TRUE ~ NA_integer_
  )
  need_fallback <- is.na(out_num) & !is.na(xs)
  xs_up <- toupper(xs)
  out_fallback <- dplyr::case_when(
    stringr::str_detect(xs_up, "\\bYES\\b|\\bCOVERED\\b")       ~ 1L,
    stringr::str_detect(xs_up, "\\bNO\\b|\\bNOT\\s+COVERED\\b") ~ 0L,
    TRUE ~ NA_integer_
  )
  ifelse(need_fallback, out_fallback, out_num)
}

# MCQ recode helper for comorbidity: 1/Yes -> 1; everything else
(No/Refused/DK/blank) -> 0
```

```r
mcq_yes1_else0 <- function(x) {
  xs <- trimws(as.character(x))
  # numeric first
  xn <- suppressWarnings(as.numeric(xs))
  out <- ifelse(!is.na(xn), ifelse(xn == 1, 1L, 0L), NA_integer_)
  # text fallback if needed
  need_fb <- is.na(out) & !is.na(xs)
  xs_up <- toupper(xs)
  fb <- ifelse(grepl("\\bYES\\b", xs_up), 1L, 0L)
  out[need_fb] <- fb[need_fb]
  out[is.na(out)] <- 0L  # missing counts as 0 per spec
  as.integer(out)
}

# Missingness audit for a data frame
missingness_summary <- function(d) {
  tibble::tibble(
    variable = names(d),
    n = nrow(d),
    missing_n = vapply(d, function(x) sum(is.na(x)), integer(1)),
    non_missing_n = n - missing_n,
    missing_pct = round(100 * missing_n / n, 2)
  ) %>% arrange(desc(missing_pct), variable)
}

# Basic nhanesA getter
nhanes_get <- function(table_base, suffix, quiet = FALSE) {
  nm <- paste0(table_base, suffix)
  if (!quiet) message("Downloading ", nm, " ...")
  out <- tryCatch(
    suppressWarnings(nhanesA::nhanes(nm)),
    error = function(e) NULL
  )
  if (is.null(out)) return(NULL)
  out %>% janitor::clean_names() %>% haven::zap_labels()
}

# Safe pull: always returns at least SEQN, lowercased
safe_get <- function(base, suf) {
  df <- nhanes_get(base, suf, quiet = TRUE)
  if (is.null(df)) return(tibble(seqn = integer()))
  df <- janitor::clean_names(df)
  if (!"seqn" %in% names(df)) df$seqn <- NA
  dplyr::mutate(df, seqn = as.integer(seqn))
}

# Yes = 1; everything else (No/Refused/DK/Missing or not-asked) = 0
yes1_else0 <- function(x) {
  xs <- trimws(as.character(x))
```

```
  xs[xs %in% c("", ".", "..", "...")] <- NA_character_
  xn <- suppressWarnings(as.numeric(xs))
  out <- ifelse(!is.na(xn), ifelse(xn == 1, 1L, 0L),
                ifelse(grepl("^YES$", toupper(xs)), 1L, 0L))
  out[is.na(out)] <- 0L
  as.integer(out)
}
```

Query NHANES components (DEMO, RXQ_RX, MCQ, HIQ)

```
# DEMO (age, sex, race/eth, education, income-to-poverty, design vars)
demo_all <- purrr::map_dfr(cycle_meta$suffix, ~{
  df <- nhanes_get("DEMO", .x, quiet = TRUE); if (is.null(df))
return(tibble())
  df %>%
    dplyr::select(dplyr::any_of(c(
      "seqn","sddsrvyr","ridageyr","riagendr","ridreth1","ridreth3",
      "dmdeduc2","indfmpir","wtint2yr","sdmvpsu","sdmvstra"
    )))
})
stopifnot(nrow(demo_all) > 0)

# Roster = one row per participant
demo_roster <- demo_all %>% dplyr::transmute(seqn = as.integer(seqn)) %>%
dplyr::distinct()

# RX: Polypharmacy counts with true zeros
# Official RXDCOUNT tables by cycle
rx_tables <- tryCatch(
  nhanesA::nhanesSearchVarName(varname = "RXDCOUNT", ystart = 2003, ystop =
2018, namesonly = TRUE),
  error = function(e) character(0)
)
stopifnot(length(rx_tables) > 0)

rx_official <- purrr::map_dfr(rx_tables, function(tbl) {
  df <- tryCatch(suppressWarnings(nhanesA::nhanes(tbl)), error = function(e)
NULL)
  if (is.null(df)) return(tibble())
  df %>%
    janitor::clean_names() %>%
    dplyr::transmute(seqn = as.integer(seqn),
                     rxdcount = suppressWarnings(as.numeric(rxdcount))) %>%
    dplyr::group_by(seqn) %>%
    dplyr::summarise(rxdcount =
dplyr::coalesce(dplyr::first(na.omit(rxdcount)), NA_real_), .groups = "drop")
})

# Fallback: count only valid medication rows in RXQ_RX*
rx_alt <- purrr::map_dfr(cycle_meta$suffix, function(suf) {
```

```r
  tbl <- paste0("RXQ_RX", suf)
  df <- tryCatch(suppressWarnings(nhanesA::nhanes(tbl)), error = function(e)
NULL)
  if (is.null(df)) return(tibble())
  df <- janitor::clean_names(df)
  df %>%
    dplyr::mutate(
      seqn         = as.integer(seqn),
      rxduse_n     = suppressWarnings(as.numeric(rxduse)),
      rxddrgid_n   = suppressWarnings(as.numeric(rxddrgid)),
      has_drugcode = !is.na(rxddrgid_n),
      has_drugname = !is.na(rxddrug) & nzchar(trimws(rxddrug)),
      med_row      = (rxduse_n == 1) & (has_drugcode | has_drugname)
    ) %>%
    dplyr::group_by(seqn) %>%
    dplyr::summarise(rxdcount_alt = sum(med_row, na.rm = TRUE), .groups =
"drop")
})

# Final RX dataset (prefer official; else fallback; else 0)
rx_all <- demo_roster %>%
  dplyr::left_join(rx_official, by = "seqn") %>%
  dplyr::left_join(rx_alt,      by = "seqn") %>%
  dplyr::mutate(rxdcount = dplyr::coalesce(as.numeric(rxdcount),
as.numeric(rxdcount_alt), 0)) %>%
  dplyr::select(seqn, rxdcount)

# Comorbidities: MCQ + DIQ010 + BPQ020 + KIQ022
# Target item names after clean_names() (all lower case)
comorb_vars <- c(
  # MCQ (ever told …)
  "mcq010","mcq080","mcq160a","mcq160n","mcq160b","mcq160c","mcq160d",
  "mcq160e","mcq160f","mcq160m","mcq160g","mcq160k","mcq160l","mcq220",
  # Other modules
  "diq010",  # diabetes
  "bpq020",  # hypertension
  "kiq022"   # kidney disease
)
# Include 16 conditions: angina pectoris, arthritis, asthma, chronic
bronchitis, congestive heart failure, coronary heart disease, diabetes,
emphysema, gout, hypertension, liver condition, myocardial infarction,
obesity, stroke, thyroid problems, and kidney disease

# Pull each module across cycles
mcq_wide <- purrr::map_dfr(cycle_meta$suffix, ~{
  df <- safe_get("MCQ", .x)
  keep <- intersect(names(df), c("seqn", comorb_vars))
  dplyr::select(df, dplyr::all_of(keep))
})
```

```r
diq_wide <- purrr::map_dfr(cycle_meta$suffix, ~{
  df <- safe_get("DIQ", .x)
  if ("diq010" %in% names(df)) dplyr::select(df, seqn, diq010) else
tibble(seqn = df$seqn)
})

bpq_wide <- purrr::map_dfr(cycle_meta$suffix, ~{
  df <- safe_get("BPQ", .x)
  if ("bpq020" %in% names(df)) dplyr::select(df, seqn, bpq020) else
tibble(seqn = df$seqn)
})

kiq_wide <- purrr::map_dfr(cycle_meta$suffix, ~{
  df <- safe_get("KIQ_U", .x)
  if ("kiq022" %in% names(df)) dplyr::select(df, seqn, kiq022) else
tibble(seqn = df$seqn)
})

# Start from roster to keep everyone; left-join modules
health_wide <- demo_roster %>%
  dplyr::left_join(mcq_wide, by = "seqn") %>%
  dplyr::left_join(diq_wide, by = "seqn") %>%
  dplyr::left_join(bpq_wide, by = "seqn") %>%
  dplyr::left_join(kiq_wide, by = "seqn")

# Ensure all comorbidity columns exist; create if missing (as NA -> will be
coded 0)
for (v in comorb_vars) {
  if (!v %in% names(health_wide)) health_wide[[v]] <- NA
}

# Recode to 1/0 and sum across items
mcq_clean <- health_wide %>%
  dplyr::mutate(dplyr::across(dplyr::all_of(comorb_vars), yes1_else0)) %>%
  dplyr::mutate(comorbidity_n =
rowSums(dplyr::across(dplyr::all_of(comorb_vars)), na.rm = TRUE)) %>%
  dplyr::select(seqn, comorbidity_n)

# Insurance: HIQ011 -> Yes/No/Unknown
hiq_all <- purrr::map_dfr(cycle_meta$suffix, ~{
  df <- nhanes_get("HIQ", .x, quiet = TRUE); if (is.null(df))
return(tibble())
  df <- janitor::clean_names(df)
  nm <- names(df)
  cand <- nm[stringr::str_detect(nm, "^hiq011")]
  tibble(
    seqn       = as.integer(df$seqn),
    hiq011_raw = if (length(cand)) as.character(df[[cand[1]]]) else
```

```
NA_character_
  )
}) %>%
  dplyr::mutate(
    yn = recode_hiq011_yesno(hiq011_raw),
    insured = dplyr::case_when(
      yn == 1L ~ "Yes",
      yn == 0L ~ "No",
      TRUE     ~ "Unknown"
    ),
    insured = factor(insured, levels = c("No","Yes","Unknown"))
  ) %>%
  dplyr::select(seqn, insured)
```

Merge, inclusion criteria (age ≥ 45), construct exposure + covariates

```
# mortality vars needed
mort_keep <- nhanes_mortality %>%
  rename_with(tolower) %>%
  select(seqn, mortstat, permth_int, permth_exm, ucod_leading)

# Merge all components
raw <- demo_all %>%
  mutate(seqn = as.integer(seqn)) %>%
  left_join(rx_all,     by = "seqn") %>%
  left_join(mcq_clean,  by = "seqn") %>%
  left_join(hiq_all,    by = "seqn") %>%
  inner_join(mort_keep, by = "seqn")

# Keep ≥45 (middle-age); build exposure, covariates, and time/event
df45 <- raw %>%
  filter(!is.na(ridageyr) & ridageyr >= 45) %>%
  mutate(
    # exposure (polypharmacy categories)
    rxdcount = suppressWarnings(as.numeric(rxdcount)),
    poly_cat = case_when(
      !is.na(rxdcount) & rxdcount == 0        ~ "0",
      !is.na(rxdcount) & rxdcount >= 1 & rxdcount <= 4 ~ "1–4",
      !is.na(rxdcount) & rxdcount >= 5 & rxdcount <= 9 ~ "5–9",
      !is.na(rxdcount) & rxdcount >= 10       ~ "≥10",
      TRUE ~ NA_character_
    ) %>% factor(levels = c("0","1–4","5–9","≥10")),

    # core demo + SES
    age_cat = case_when(
      ridageyr >= 45 & ridageyr < 65 ~ "45-64",
      ridageyr >= 65 & ridageyr < 80 ~ "65-79",
      ridageyr >= 80 ~ "80 or above"
    ) %>% factor(levels = c("45-64", "65-79", "80 or above")),
```

```
    sex       = recode_sex(riagendr),
    race_eth  = derive_race_eth(cur_data()),
    ed_cat    = derive_ed_cat(dmdeduc2),

    # income-to-poverty: keep numeric + categorical with "Unknown"
    indfmpir  = to_num(indfmpir),
    indfmpir_cat = case_when(
      is.na(indfmpir)        ~ "Unknown",
      indfmpir < 1           ~ "<1.00 (below poverty)",
      indfmpir >= 1 & indfmpir < 2 ~ "1.00–1.99",
      indfmpir >= 2 & indfmpir < 4 ~ "2.00–3.99",
      indfmpir >= 4          ~ "≥4.00"
    ) %>% factor(levels = c("<1.00 (below poverty)","1.00–1.99","2.00–
3.99","≥4.00","Unknown")),

    # insurance already built as factor(No/Yes/Unknown)
    insured   = factor(insured, levels = c("No","Yes","Unknown")),

    # survival time/event
    time_m    = as.numeric(permth_int),
    time_y    = time_m / 12,
    event     = as.integer(mortstat)
  )

# Assemble analytic dataset
analytic <- df45 %>%
  filter(!is.na(time_y), event %in% c(0,1)) %>%
  transmute(
    seqn, sddsrvyr,
    age_cat, sex, race_eth, ed_cat,
    indfmpir, indfmpir_cat,
    insured,
    rxdcount, poly_cat,
    comorbidity_n,
    event, time_y,
    wtint2yr, sdmvpsu, sdmvstra,
    ucod_leading
  )

# Create complete-case set for models
model_vars <- c("age_cat","sex","race_eth","ed_cat","indfmpir_cat",

"insured","poly_cat","comorbidity_n","time_y","event","sddsrvyr")

analytic_final <- analytic %>%
  filter(if_all(all_of(model_vars), ~ !is.na(.))) %>%
  mutate(
    event_f = factor(event, levels = c(0,1), labels =
c("Alive/Censored","Deceased")),
```

```
    wtint_pool = wtint2yr / 8  # pooled weight across 8 cycles
  )

# Survey designs (for models)
dsgn <- svydesign(
  ids = ~sdmvpsu, strata = ~sdmvstra, weights = ~wtint_pool,
  nest = TRUE, data = analytic_final
)

# Quick audit
print(missingness_summary(analytic %>% select(age_cat, sex, race_eth, ed_cat,
indfmpir, indfmpir_cat,
                                               insured, rxdcount, poly_cat,
comorbidity_n, time_y, event)))
```

```
## # A tibble: 12 × 5
##    variable          n missing_n non_missing_n missing_pct
##    <chr>         <int>     <int>         <int>       <dbl>
##  1 indfmpir      25741      2570         23171        9.98
##  2 race_eth      25741      1534         24207        5.96
##  3 age_cat       25741         0         25741        0
##  4 comorbidity_n 25741         0         25741        0
##  5 ed_cat        25741         0         25741        0
##  6 event         25741         0         25741        0
##  7 indfmpir_cat  25741         0         25741        0
##  8 insured       25741         0         25741        0
##  9 poly_cat      25741         0         25741        0
## 10 rxdcount      25741         0         25741        0
## 11 sex           25741         0         25741        0
## 12 time_y        25741         0         25741        0
```

Table 1: Descriptive statistics

```
tbl1_by_outcome <- gtsummary::tbl_summary(
  data = analytic_final,
  by = event_f,
  include = c(
    poly_cat,
    rxdcount, age_cat, sex, race_eth, ed_cat, indfmpir_cat, insured,
    comorbidity_n
  ),
  label = list(
    age_cat     ~ "Age group",
    sex         ~ "Sex",
    race_eth    ~ "Race/ethnicity",
    ed_cat      ~ "Education",
    indfmpir_cat ~ "Income-to-poverty ratio",
    insured     ~ "Insurance coverage",
    rxdcount    ~ "Medication count",
    comorbidity_n ~ "Number of comorbidities",
```

```
    poly_cat      ~ "Polypharmacy"
  ),
  statistic = list(
    all_continuous()  ~ "{mean} ({sd})",
    all_categorical() ~ "{n} ({p}%)"
  ),
  missing = "ifany"
) |>
  gtsummary::bold_labels()

tbl1_by_outcome
```

Characteristic	Alive/Censored N = 18,688[1]	Deceased N = 5,519[1]
Polypharmarcy		
0	5,161 (28%)	705 (13%)
1–4	9,005 (48%)	2,308 (42%)
5–9	3,823 (20%)	1,958 (35%)
≥10	699 (3.7%)	548 (9.9%)
Medication count	2.9 (3.0)	4.6 (3.6)
Age group		
45-64	12,242 (66%)	1,267 (23%)
65-79	5,267 (28%)	2,226 (40%)
80 or above	1,179 (6.3%)	2,026 (37%)
Sex		
Male	8,833 (47%)	3,051 (55%)
Female	9,855 (53%)	2,468 (45%)
Race/ethnicity		
Non-Hispanic White	5,580 (30%)	2,789 (51%)
Non-Hispanic Black	3,033 (16%)	855 (15%)
Non-Hispanic Asian	4,397 (24%)	788 (14%)
Mexican American	1,900 (10%)	491 (8.9%)
Other Hispanic	683 (3.7%)	165 (3.0%)
Other/Multi	3,095 (17%)	431 (7.8%)
Education		
Less than HS	2,492 (13%)	1,640 (30%)
HS/GED	1,983 (11%)	1,054 (19%)
Some college	2,186 (12%)	906 (16%)
College+	1,762 (9.4%)	555 (10%)
Unknown	10,265 (55%)	1,364 (25%)
Income-to-poverty ratio		
<1.00 (below poverty)	2,996 (16%)	1,028 (19%)
1.00–1.99	4,335 (23%)	1,826 (33%)
2.00–3.99	4,488 (24%)	1,379 (25%)
≥4.00	4,974 (27%)	806 (15%)
Unknown	1,895 (10%)	480 (8.7%)
Insurance coverage		
No	2,657 (14%)	271 (4.9%)
Yes	14,472 (77%)	3,900 (71%)
Unknown	1,559 (8.3%)	1,348 (24%)
Number of comorbidities	2.31 (1.93)	3.25 (2.25)

[1]n (%); Mean (SD)

```
tbl1_by_exposure <- gtsummary::tbl_summary(
  data = analytic_final,
```

```r
  by = poly_cat,
  include = c(
    age_cat, sex, race_eth, ed_cat, indfmpir_cat, insured,
    rxdcount, comorbidity_n, event_f
  ),
  label = list(
    age_cat     ~ "Age group",
    sex         ~ "Sex",
    race_eth    ~ "Race/ethnicity",
    ed_cat      ~ "Education",
    indfmpir_cat ~ "Income-to-poverty ratio",
    insured     ~ "Insurance coverage",
    rxdcount    ~ "Medication count",
    comorbidity_n ~ "Number of comorbidities",
    event_f     ~ "Mortality prevalence"
  ),
  statistic = list(
    all_continuous()  ~ "{mean} ({sd})",
    all_categorical() ~ "{n} ({p}%)"
  ),
  missing = "ifany"
) |>
  gtsummary::bold_labels()

tbl1_by_exposure
```

Characteristic	**0** N = 5,866[1]	**1–4** N = 11,313[1]	**5–9** N = 5,781[1]	**≥10** N = 1,247[1]
Age group				
45-64	4,632 (79%)	6,182 (55%)	2,185 (38%)	510 (41%)
65-79	952 (16%)	3,657 (32%)	2,379 (41%)	505 (40%)
80 or above	282 (4.8%)	1,474 (13%)	1,217 (21%)	232 (19%)
Sex				
Male	3,305 (56%)	5,334 (47%)	2,656 (46%)	589 (47%)
Female	2,561 (44%)	5,979 (53%)	3,125 (54%)	658 (53%)
Race/ethnicity				
Non-Hispanic White	1,886 (32%)	4,032 (36%)	2,045 (35%)	406 (33%)
Non-Hispanic Black	1,039 (18%)	1,851 (16%)	809 (14%)	189 (15%)
Non-Hispanic Asian	956 (16%)	2,438 (22%)	1,436 (25%)	355 (28%)
Mexican American	803 (14%)	1,040 (9.2%)	463 (8.0%)	85 (6.8%)
Other Hispanic	302 (5.1%)	352 (3.1%)	157 (2.7%)	37 (3.0%)
Other/Multi	880 (15%)	1,600 (14%)	871 (15%)	175 (14%)
Education				
Less than HS	1,120 (19%)	1,836 (16%)	966 (17%)	210 (17%)
HS/GED	694 (12%)	1,437 (13%)	769 (13%)	137 (11%)
Some college	756 (13%)	1,455 (13%)	716 (12%)	165 (13%)
College+	574 (9.8%)	1,249 (11%)	427 (7.4%)	67 (5.4%)
Unknown	2,722 (46%)	5,336 (47%)	2,903 (50%)	668 (54%)
Income-to-poverty ratio				
<1.00 (below poverty)	1,049 (18%)	1,619 (14%)	1,043 (18%)	313 (25%)
1.00–1.99	1,469 (25%)	2,682 (24%)	1,619 (28%)	391 (31%)
2.00–3.99	1,377 (23%)	2,783 (25%)	1,434 (25%)	273 (22%)
≥4.00	1,312 (22%)	3,157 (28%)	1,137 (20%)	174 (14%)
Unknown	659 (11%)	1,072 (9.5%)	548 (9.5%)	96 (7.7%)
Insurance coverage				
No	1,616 (28%)	1,005 (8.9%)	272 (4.7%)	35 (2.8%)
Yes	3,575 (61%)	8,852 (78%)	4,841 (84%)	1,104 (89%)
Unknown	675 (12%)	1,456 (13%)	668 (12%)	108 (8.7%)
Medication count	0.0 (0.0)	2.4 (1.1)	6.4 (1.3)	12.0 (2.2)
Number of comorbidities	0.96 (1.19)	2.27 (1.54)	3.91 (1.91)	5.68 (2.32)
Mortality prevalence				
Alive/Censored	5,161 (88%)	9,005 (80%)	3,823 (66%)	699 (56%)
Deceased	705 (12%)	2,308 (20%)	1,958 (34%)	548 (44%)

[1]n (%); Mean (SD)

Cox PH models (survey-weighted): crude → basic → full, with a merged table

```r
# Crude (exposure only)
m_crude_svy <- svycoxph(Surv(time_y, event) ~ poly_cat, design = dsgn)

# Basic (add demographics)
m_basic_svy <- svycoxph(
  Surv(time_y, event) ~ poly_cat + age_cat + sex + race_eth,
  design = dsgn
)

# Full planned model
m_full_svy <- svycoxph(
  Surv(time_y, event) ~ poly_cat + age_cat + sex + race_eth +
    ed_cat + indfmpir_cat + insured + comorbidity_n,
  design = dsgn
)

# Labels, per model (only include terms that appear in that model)
labels_crude <- list(
  poly_cat ~ "Polypharmacy"
)

labels_basic <- list(
  poly_cat  ~ "Polypharmacy",
  age_cat ~ "Age group",
  sex       ~ "Sex",
  race_eth  ~ "Race/ethnicity"
)

labels_full <- list(
  poly_cat      ~ "Polypharmacy",
  age_cat     ~ "Age group",
  sex           ~ "Sex",
  race_eth      ~ "Race/ethnicity",
  ed_cat        ~ "Education",
  indfmpir_cat  ~ "Income-to-poverty ratio",
  insured       ~ "Insurance coverage",
  comorbidity_n ~ "Number of comorbidities"
)

# Build tables with intuitive labels
tbl_crude <- gtsummary::tbl_regression(
  m_crude_svy, exponentiate = TRUE, label = labels_crude
)

## Stratified 1 - level Cluster Sampling design (with replacement)
## With (244) clusters.
## svydesign(ids = ~sdmvpsu, strata = ~sdmvstra, weights = ~wtint_pool,
##     nest = TRUE, data = analytic_final)
```

```r
tbl_basic <- gtsummary::tbl_regression(
  m_basic_svy, exponentiate = TRUE, label = labels_basic
)

## Stratified 1 - level Cluster Sampling design (with replacement)
## With (244) clusters.
## svydesign(ids = ~sdmvpsu, strata = ~sdmvstra, weights = ~wtint_pool,
##     nest = TRUE, data = analytic_final)

tbl_full  <- gtsummary::tbl_regression(
  m_full_svy,  exponentiate = TRUE, label = labels_full
)

## Stratified 1 - level Cluster Sampling design (with replacement)
## With (244) clusters.
## svydesign(ids = ~sdmvpsu, strata = ~sdmvstra, weights = ~wtint_pool,
##     nest = TRUE, data = analytic_final)

# Create a side-by-side table
models_merged <- gtsummary::tbl_merge(
  tbls = list(tbl_crude, tbl_basic, tbl_full),
  tab_spanner = c("**Crude**", "**Basic (Age/Sex/Race)**", "**Full**")
) |>
  gtsummary::bold_labels()

models_merged
```

Characteristic	Crude			Basic (Age/Sex/Race)			Full		
	HR	95% CI	p-value	HR	95% CI	p-value	HR	95% CI	p-value
Polypharmacy									
0	—	—		—	—		—	—	
1–4	1.85	1.63, 2.09	<0.001	1.24	1.10, 1.40	<0.001	1.16	1.02, 1.32	0.022
5–9	4.29	3.75, 4.90	<0.001	2.19	1.91, 2.50	<0.001	1.68	1.44, 1.96	<0.001
≥10	7.33	6.15, 8.74	<0.001	4.08	3.44, 4.84	<0.001	2.42	1.97, 2.98	<0.001
Age group									
45-64				—	—		—	—	
65-79				3.59	3.28, 3.93	<0.001	3.12	2.82, 3.45	<0.001
80 or above				13.5	12.3, 14.9	<0.001	11.1	9.94, 12.4	<0.001
Sex									
Male				—	—		—	—	
Female				0.66	0.62, 0.71	<0.001	0.61	0.57, 0.66	<0.001
Race/ethnicity									
Non-Hispanic White				—	—		—	—	
Non-Hispanic Black				1.19	1.07, 1.31	<0.001	0.99	0.89, 1.10	0.8
Non-Hispanic Asian				0.93	0.82, 1.05	0.2	0.98	0.83, 1.16	0.8
Mexican American				0.93	0.80, 1.07	0.3	0.71	0.62, 0.83	<0.001
Other Hispanic				0.80	0.62, 1.02	0.073	0.65	0.51, 0.84	<0.001
Other/Multi				1.00	0.86, 1.16	>0.9	0.91	0.78, 1.08	0.3
Education									
Less than HS							—	—	
HS/GED							0.83	0.74, 0.94	0.002
Some college							0.84	0.73, 0.96	0.009
College+							0.71	0.62, 0.81	<0.001
Unknown							0.82	0.70, 0.96	0.012
Income-to-poverty ratio									
<1.00 (below poverty)							—	—	
1.00–1.99							0.82	0.74, 0.90	<0.001
2.00–3.99							0.62	0.55, 0.69	<0.001
≥4.00							0.44	0.38, 0.51	<0.001
Unknown							0.70	0.59, 0.82	<0.001
Insurance coverage									
No							—	—	
Yes							0.94	0.79, 1.12	0.5

Characteristic	Crude			Basic (Age/Sex/Race)			Full		
	HR	95% CI	p-value	HR	95% CI	p-value	HR	95% CI	p-value
Unknown							1.09	0.90, 1.32	0.4
Number of comorbidities							1.12	1.09, 1.14	<0.001

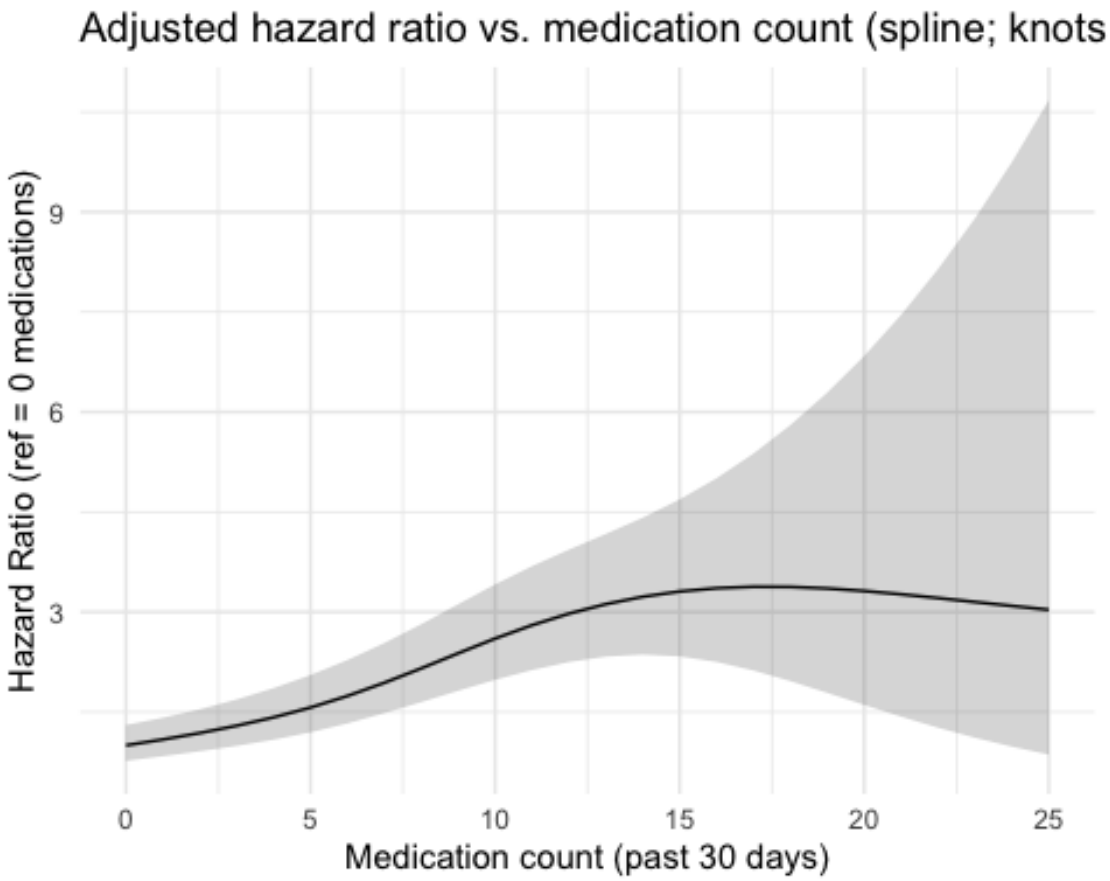
Abbreviations: CI = Confidence Interval, HR = Hazard Ratio

Sensitivity: treat polypharmacy as continuous with natural cubic spline in the full model

```
# Literature often places knots around 5 and 10 medications.
# We fit the full covariate set, replacing poly_cat with ns(rxdcount,
knots=c(5,10))
m_spline_svy <- svycoxph(
  Surv(time_y, event) ~ splines::ns(rxdcount, knots = c(5, 10)) +
    age_cat + sex + race_eth + ed_cat + indfmpir_cat + insured +
comorbidity_n,
  design = dsgn
)

# Global design-based Wald test for the overall (possibly non-linear) effect
of rxdcount
glob_test <- survey::regTermTest(m_spline_svy, ~ splines::ns(rxdcount, knots
= c(5, 10)))
cat("\nGlobal test for spline(rxdcount) (design-based Wald):\n")

##
## Global test for spline(rxdcount) (design-based Wald):

print(glob_test)

## Wald test for splines::ns(rxdcount, knots = c(5, 10))
##  in svycoxph(formula = Surv(time_y, event) ~ splines::ns(rxdcount,
##     knots = c(5, 10)) + age_cat + sex + race_eth + ed_cat + indfmpir_cat +
##     insured + comorbidity_n, design = dsgn)
## F =  71.46976  on  3  and  103  df: p= < 2.22e-16

# Make the spline interpretable: adjusted HR curve vs rxdcount = 0
#    Build a simple reference profile from typical values (median/mode)
Mode <- function(x) names(sort(table(x), decreasing = TRUE))[1]

ref_row <- tibble::tibble(
  age_cat     = factor(Mode(analytic_final$age_cat), levels =
levels(analytic_final$age_cat)),
  sex         = factor(Mode(analytic_final$sex), levels =
levels(analytic_final$sex)),
  race_eth    = factor(Mode(analytic_final$race_eth), levels =
levels(analytic_final$race_eth)),
  ed_cat      = factor(Mode(analytic_final$ed_cat), levels =
levels(analytic_final$ed_cat)),
```

```r
  indfmpir_cat  = factor(Mode(analytic_final$indfmpir_cat), levels =
levels(analytic_final$indfmpir_cat)),
  insured       = factor(Mode(analytic_final$insured), levels =
levels(analytic_final$insured)),
  comorbidity_n = median(analytic_final$comorbidity_n, na.rm = TRUE)
)

# Prediction grid across medication counts
grid <- tidyr::crossing(rxdcount = 0:25, ref_row)

# Predict linear predictors
p_out <- tryCatch(
  predict(m_spline_svy, newdata = grid, type = "lp", se.fit = TRUE),
  error = function(e) NULL
)

if (is.null(p_out)) {
  lp_vec <- as.numeric(predict(m_spline_svy, newdata = grid, type = "lp"))
  se_lp  <- rep(NA_real_, length(lp_vec))
} else {
  if (is.list(p_out)) {
    lp_vec <- as.numeric(p_out$fit)
    se_lp  <- as.numeric(p_out$se.fit)
  } else {
    lp_vec <- as.numeric(p_out)
    se_lp  <- rep(NA_real_, length(lp_vec))
  }
}

grid <- grid %>%
  mutate(
    lp    = lp_vec,
    se_lp = se_lp,
    lp0   = lp[rxdcount == 0],                # reference at 0 medications
    HR      = exp(lp - lp0),                  # hazard ratio vs 0
    HR_low  = ifelse(is.na(se_lp), NA_real_, exp((lp - 1.96*se_lp) - lp0)),
    HR_high = ifelse(is.na(se_lp), NA_real_, exp((lp + 1.96*se_lp) - lp0))
  )

# Plot: adjusted HR vs medication count (ribbon shown only if SEs available)
p <- ggplot(grid, aes(x = rxdcount, y = HR)) +
  geom_line() +
  { if (all(is.na(grid$HR_low))) NULL else geom_ribbon(aes(ymin = HR_low,
ymax = HR_high), alpha = 0.20) } +
  labs(
    title = "Adjusted hazard ratio vs. medication count (spline; knots at 5 &
10)",
    x = "Medication count (past 30 days)",
    y = "Hazard Ratio (ref = 0 medications)"
```

```
  ) +
  theme_minimal()

print(p)
```




```
# HR table at useful counts (relative to 0 meds)
at_counts <- tibble::tibble(rxdcount = c(0, 2, 5, 7, 10, 15, 20)) %>%
tidyr::crossing(ref_row)

p_out2 <- tryCatch(
  predict(m_spline_svy, newdata = at_counts, type = "lp", se.fit = TRUE),
  error = function(e) NULL
)

if (is.null(p_out2)) {
  lp2 <- as.numeric(predict(m_spline_svy, newdata = at_counts, type = "lp"))
  se2 <- rep(NA_real_, length(lp2))
} else {
  if (is.list(p_out2)) {
    lp2 <- as.numeric(p_out2$fit)
    se2 <- as.numeric(p_out2$se.fit)
  } else {
    lp2 <- as.numeric(p_out2)
```

```
    se2 <- rep(NA_real_, length(lp2))
  }
}

hr_table <- at_counts %>%
  mutate(
    lp  = lp2,
    se  = se2,
    lp0 = lp[rxdcount == 0],
    HR     = exp(lp - lp0),
    HR_lo  = ifelse(is.na(se), NA_real_, exp((lp - 1.96*se) - lp0)),
    HR_hi  = ifelse(is.na(se), NA_real_, exp((lp + 1.96*se) - lp0))
  ) %>%
  select(rxdcount, HR, HR_lo, HR_hi)

cat("\nAdjusted hazard ratios at selected medication counts (vs 0 meds):\n")

##
## Adjusted hazard ratios at selected medication counts (vs 0 meds):

print(hr_table)

## # A tibble: 7 × 4
##   rxdcount    HR HR_lo HR_hi
##      <dbl> <dbl> <dbl> <dbl>
## 1        0  1    0.767  1.30
## 2        2  1.18 0.911  1.54
## 3        5  1.57 1.20   2.05
## 4        7  1.94 1.48   2.53
## 5       10  2.60 1.98   3.41
## 6       15  3.31 2.33   4.69
## 7       20  3.32 1.61   6.84

# gtsummary model table with clear labels and a global p-value for the spline
term
nice_labels <- list(
  `splines::ns(rxdcount, knots = c(5, 10))` ~ "Medication count (spline;
knots 5 & 10)",
  age_cat     ~ "Age group",
  sex         ~ "Sex",
  race_eth    ~ "Race/ethnicity",
  ed_cat      ~ "Education",
  indfmpir_cat ~ "Income-to-poverty ratio",
  insured     ~ "Insurance coverage",
  comorbidity_n ~ "Number of comorbidities"
)

spline_tbl <- gtsummary::tbl_regression(
  m_spline_svy, exponentiate = TRUE, label = nice_labels
) |>
```

```
  gtsummary::add_global_p(terms = "splines::ns(rxdcount, knots = c(5, 10))")
|>
  gtsummary::bold_labels() |>
  gtsummary::modify_header(label ~ "**Full model with rxdcount spline**")
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (244) clusters.
## svydesign(ids = ~sdmvpsu, strata = ~sdmvstra, weights = ~wtint_pool,
##     nest = TRUE, data = analytic_final)
```

```
spline_tbl
```

Full model with rxdcount spline	HR	95% CI	p-value
Medication count (spline; knots 5 & 10)			<0.001
splines::ns(rxdcount, knots = c(5, 10))1	3.29	2.48, 4.37	
splines::ns(rxdcount, knots = c(5, 10))2	4.89	2.73, 8.75	
splines::ns(rxdcount, knots = c(5, 10))3	2.56	0.98, 6.65	
Age group			<0.001
45-64	—	—	
65-79	3.09	2.79, 3.42	
80 or above	10.9	9.81, 12.2	
Sex			<0.001
Male	—	—	
Female	0.61	0.57, 0.65	
Race/ethnicity			<0.001
Non-Hispanic White	—	—	
Non-Hispanic Black	1.00	0.90, 1.10	
Non-Hispanic Asian	0.99	0.83, 1.18	
Mexican American	0.72	0.62, 0.83	
Other Hispanic	0.66	0.51, 0.84	
Other/Multi	0.91	0.78, 1.08	
Education			<0.001
Less than HS	—	—	
HS/GED	0.83	0.74, 0.94	
Some college	0.84	0.74, 0.96	
College+	0.72	0.63, 0.82	
Unknown	0.82	0.70, 0.95	
Income-to-poverty ratio			<0.001
<1.00 (below poverty)	—	—	
1.00–1.99	0.82	0.74, 0.91	
2.00–3.99	0.62	0.55, 0.70	
≥4.00	0.44	0.38, 0.52	
Unknown	0.70	0.60, 0.83	
Insurance coverage			0.010
No	—	—	
Yes	0.92	0.78, 1.09	
Unknown	1.07	0.89, 1.28	
Number of comorbidities	1.09	1.07, 1.12	<0.001

Full model with rxdcount spline	HR	95% CI	p-value

Abbreviations: CI = Confidence Interval, HR = Hazard Ratio