```
# ============================================================
#  WSIC-RDC: Publication-Ready Analysis Script
# ------------------------------------------------------------
# Author: Sunnia Gupta (Clinical Research Fellow) & collaborators
# Maintainer: Sunnia Gupta
# Script: rdc_analysis_clean.R
# Created: 2026-03-02
# Layout : Repo layout = scripts/, figures/, metadata/
#
# ============================================================
# Description:
#   Final integrated analysis script including:
#     - Cohort derivation
#     - Demographics, lifestyle, frailty, LTCs
#     - Symptom extraction
#     - Monthly pre-referral activity panel (0–24m)
#     - Logistic regression
#     - Poisson / Negative Binomial models
#     - event_by_month() + Figure 4 panels
#     - Kaplan–Meier (referral → diagnosis)

# ------------------------------
# 0) Setup
# ------------------------------

suppressPackageStartupMessages({
  library(tidyverse)
  library(readr)
  library(readxl)
  library(janitor)
  library(lubridate)
  library(stringr)
  library(broom)
  library(forcats)
  library(ggplot2)
  library(survival)
})


# Project paths ------------------------------------------------------------

data_dir <- "/mnt/workspaceblobstore/RDC/"  # WSIC SDE export location used in your
drafts

# Small helpers ------------------------------------------------------------

read_csv_here <- function(filename, ...) {
  path <- file.path(data_dir, filename)
  message("→ Reading ", path)
  readr::read_csv(path, show_col_types = FALSE, ...)
```

```
}

rename_patient_key <- function(df) {
  candidate <- names(df)[grepl("^PATIENT_KEY", names(df))][1]
  if (!is.na(candidate) && candidate != "PATIENT_KEY") {
    df <- dplyr::rename(df, PATIENT_KEY = dplyr::all_of(candidate))
  }
  df
}

# Figures: a clean theme (colorblind-friendly) ----------------------

fig_theme_a <- function() {
  theme_minimal(base_size = 11) +
    theme(
      panel.grid.major = element_line(colour = "#e6e6e6", linewidth = 0.4),
      panel.grid.minor = element_blank(),
      plot.title.position = "plot",
      plot.title = element_text(face = "bold"),
      axis.title = element_text(face = "bold"),
      legend.position = "right"
    )
}

scale_colour_a <- scale_colour_brewer(palette = "Set2", na.translate = FALSE)
scale_fill_a   <- scale_fill_brewer(palette = "Set2", na.translate = FALSE)

save_plot <- function(p, name, w = 7, h = 5) {
  # Save both PNG and PDF for journals & GitHub
  ggsave(filename = file.path(fig_dir, paste0(name, ".png")), plot = p, width = w,
height = h, dpi = 300)
  ggsave(filename = file.path(fig_dir, paste0(name, ".pdf")), plot = p, width = w,
height = h, dpi = 300, device = cairo_pdf)
}

moving_average3 <- function(x) stats::filter(x, rep(1/3, 3), sides = 2)

# ------------------------------
# 1) Load ALL datasets (kept)
# ------------------------------
# (Supplementary B inputs: demographics, EFI, LTC, GP visits, meds, labs, OBS)
#   - Aligns to Tables B.2–B.13 in Supplementary B
#   - Cohort flow links back to main manuscript Methods

original_cohort   <- read_csv_here("RDC_original_cohort.csv")
RDC_REFERRALS     <- read_csv_here("RDC_REFERRALS.csv")
LTC               <- read_csv_here("RDC_LTC.csv")
GPvisits          <- read_csv_here("RDC_GPvisits.csv")
demographics_raw  <- read_csv_here("RDC_demographics.csv")
cancer_cohort     <- read_csv_here("cancer_cohort.csv")
```

```
EFI               <- read_csv_here("RDC_EFI.csv")
ALCOHOL           <- read_csv_here("RDC_ALCOHOL.csv")
SMOKING           <- read_csv_here("RDC_SMOKING.csv")
LTC_unpivotted    <- read_csv_here("RDC_LTC_unpivotted.csv")
specific_meds     <- read_csv_here("RDC_specific_meds.csv")
RDC_BLOOD_RESULTS <- read_csv_here("RDC_BLOOD_RESULTS.csv")
RDC_OBS           <- read_csv_here("RDC_OBS.csv")
DISORDER          <- read_csv_here("Disorder_table.csv")   # for Non-cancer
outcomes (Table B.8)

# ------------------------------
# 2) Core harmonisation & key cohorts
# ------------------------------

# Demographics ------------------------------------------------------------
demographics <- demographics_raw %>% rename_patient_key() %>% clean_names()
reg_col <- names(demographics)[grepl("register_status", names(demographics),
ignore.case = TRUE)][1]
non_registered <- demographics %>% filter(.data[[reg_col]] %in% c("X", "O")) %>%
distinct(PATIENT_KEY)

# ================================================
# (2a) Cancer diagnoses (WSIC; 0–3 months window)
# ================================================
# Used throughout: case vs non-case splits; maps to Supplementary B.7 and Results
text
cancer_cohort_3 <- cancer_cohort %>%
  mutate(
    earliest_diagnosis_date = floor_date(EARLIEST_DIAGNOSIS_DATE, unit = "day"),
    ref_date = floor_date(DOR, unit = "day"),
    rel_month = interval(ref_date, earliest_diagnosis_date) %/% months(1)
  ) %>%
  filter(rel_month >= 0, rel_month <= 3) %>%
  group_by(PATIENT_KEY, earliest_diagnosis_date, DOR) %>%
  summarise(diagnosis = paste(WD_SNOMED_TERM, collapse = "\n"), .groups = "drop")

cancer_diagnosis_path <- file.path(data_dir, "cancer_diagnosis_grouped.csv")
if (file.exists(cancer_diagnosis_path)) {
  cancer_diagnosis <- read_csv_here("cancer_diagnosis_grouped.csv") %>%
distinct(PATIENT_KEY, .keep_all = TRUE)
} else {
  cancer_diagnosis <- cancer_cohort_3 %>% transmute(PATIENT_KEY, diagnosis_grouped
= "Cancer (unspecified)")
}

cases     <- cancer_diagnosis %>% anti_join(non_registered, by = "PATIENT_KEY")
case_keys <- cases$PATIENT_KEY

# Optional: future cancer among non-cases in +12m (sensitivity; Results narrative)
future_cancer_12 <- cancer_cohort %>%
```

```
  filter(!PATIENT_KEY %in% case_keys) %>%
  mutate(earliest_diagnosis_date = EARLIEST_DIAGNOSIS_DATE, ref_date = DOR) %>%
  filter(ref_date >= earliest_diagnosis_date %m-% months(12) &
earliest_diagnosis_date >= ref_date) %>%
  group_by(PATIENT_KEY) %>% summarise(diagnosis = paste(WD_SNOMED_TERM, collapse =
"\n"), .groups = "drop")

# ============================================================
#(2b) Non-cancer diagnoses (SNOMED ‘disorder’; 0–3 months)
# ============================================================
# **Table B.8 (Supplementary B):** most common benign diagnoses within 3 months
post-referral,
# captured from SNOMED entries labelled 'disorder', excluding cancer cases.

DISORDER <- DISORDER %>%
  rename_patient_key() %>%
  clean_names() %>%
  rename(
    diag_date    = clinical_effective_date,
    snomed_term  = wd_snomed_term,
    snomed_code  = wd_snomed_code,
    semantic_tag = core_concept_description
  ) %>%
  mutate(diag_date = ymd(diag_date)) %>%
  left_join(demographics %>% select(PATIENT_KEY, DOR), by = "PATIENT_KEY") %>%
  mutate(DOR = ymd(DOR))

noncancer_dx <- DISORDER %>%
  filter(!PATIENT_KEY %in% case_keys) %>%
  filter(str_detect(tolower(semantic_tag %||% ""), "disorder") |
         str_detect(tolower(snomed_term  %||% ""), "disorder")) %>%
  filter(!is.na(DOR), !is.na(diag_date),
         diag_date >= DOR, diag_date <= DOR %m+% months(3)) %>%
  anti_join(non_registered, by = "PATIENT_KEY") %>%
  distinct(PATIENT_KEY, diag_date, snomed_term, snomed_code)

noncancer_top            <- noncancer_dx %>% count(snomed_term, name = "n") %>%
arrange(desc(n))
noncancer_patient_counts <- noncancer_dx %>% count(PATIENT_KEY, name =
"n_disorders_0to3m")

# Exports supporting Figure 3
readr::write_csv(noncancer_top,            file.path(out_dir,
"noncancer_top_0to3m.csv"))
readr::write_csv(noncancer_patient_counts, file.path(out_dir,
"noncancer_patient_counts_0to3m.csv"))

# ============================================================
# 3) Demographics & basics (Age, Gender, Ethnicity, Language, IMD)
# ============================================================
```

```
# **Supplementary B.2–B.3**: Ethnicity/Language distributions; IMD & EFI summaries

age_table <- demographics %>%
  transmute(
    PATIENT_KEY,
    DOR = DOR,
    yob = WD_YEAR_MONTH_OF_BIRTH,
    referral_age = round(as.numeric(interval(yob, DOR) / years(1)))
  ) %>%
  mutate(
    age_band = case_when(
      referral_age >= 18 & referral_age < 35 ~ "18-34",
      referral_age >= 35 & referral_age < 55 ~ "35-54",
      referral_age >= 55 & referral_age < 65 ~ "55-64",
      referral_age >= 65 & referral_age < 80 ~ "65-79",
      referral_age >= 80 ~ "80+",
      TRUE ~ NA_character_
    )
  )

gender_tab    <- demographics %>% transmute(PATIENT_KEY, gender    =
coalesce(GENDER_CODE, GENDER_DESCRIPTION)) %>% distinct()
ethnicity_tab <- demographics %>% transmute(PATIENT_KEY, ethnicity =
na_if(ETHNIC_CATEGORY_DESCRIPTION, "Not stated")) %>% distinct()
language_tab  <- demographics %>% transmute(PATIENT_KEY, language  =
na_if(WD_FIRST_LANGUAGE_DESCRIPTION, "Refusal by patient")) %>% distinct()

imd_tab <- demographics %>%
  transmute(PATIENT_KEY, imd_decile = WD_DEPRIVATION_DASHBOARD) %>%
  mutate(
    IMD_quintile = case_when(
      imd_decile > 0 & imd_decile <= 2 ~ "1st (most deprived)",
      imd_decile >= 3 & imd_decile <= 4 ~ "2nd",
      imd_decile >= 5 & imd_decile <= 6 ~ "3rd",
      imd_decile >= 7 & imd_decile <= 8 ~ "4th",
      imd_decile >= 9 & imd_decile <=10 ~ "5th (least deprived)",
      TRUE ~ "Unknown"
    )
  ) %>% distinct()

# ============================================================
# 4) Lifestyle: Alcohol & Smoking (latest status pre-referral)
# ============================================================
# **Supplementary B.5**: Alcohol & smoking status and intensity distributions

ALCOHOL <- ALCOHOL %>% rename_patient_key() %>% mutate(WD_SNOMED_CODE =
format(WD_SNOMED_CODE, scientific = FALSE) |> str_trim())
SMOKING <- SMOKING %>% rename_patient_key() %>% mutate(WD_SNOMED_CODE =
format(WD_SNOMED_CODE, scientific = FALSE) |> str_trim())
```

```
alcohol_status <- ALCOHOL %>%
  select(PATIENT_KEY, WD_REGISTER_STATUS, CORE_CONCEPT_DESCRIPTION,
CLINICAL_EFFECTIVE_DATE, WD_SNOMED_TERM, WD_SNOMED_CODE) %>%
  mutate(
    alc_cat = case_when(
      str_detect(WD_SNOMED_CODE, "NA") ~ NA_character_,
      str_detect(WD_SNOMED_CODE, "^(228274009|137953007|783261004)$") ~ "never
drinker",
      str_detect(WD_SNOMED_CODE,
"^(160579004|82581004|160583004|160585006|160587003|160582009|286857004|160584005|1
05542008)$") ~ "ex-drinker",
      TRUE ~ "current drinker"
    )
  ) %>%
  arrange(PATIENT_KEY, desc(CLINICAL_EFFECTIVE_DATE)) %>% group_by(PATIENT_KEY) %>%
slice_head(n = 1) %>% ungroup()

alcohol_quant <- ALCOHOL %>%
  filter(!str_detect(WD_SNOMED_CODE, "^(228274009|137953007|783261004)$")) %>%
  filter(str_detect(RESULT_VALUE_UNITS, regex("wk|week", ignore_case = TRUE))) %>%
  group_by(PATIENT_KEY) %>%
  summarise(average_units = round(mean(RESULT_VALUE, na.rm = TRUE)), .groups =
"drop") %>%
  mutate(
    drinker_group = case_when(
      is.na(average_units) ~ NA_character_,
      average_units < 1    ~ "trivial",
      average_units <= 2   ~ "light",
      average_units <= 6   ~ "moderate",
      average_units >= 7 & average_units <= 9 ~ "heavy",
      average_units >  9   ~ "very_heavy"
    )
  )

smoking_status <- SMOKING %>%
  select(PATIENT_KEY, WD_REGISTER_STATUS, CORE_CONCEPT_DESCRIPTION,
CLINICAL_EFFECTIVE_DATE, WD_SNOMED_TERM, WD_SNOMED_CODE) %>%
  mutate(
    smoking_cat = case_when(
      str_detect(WD_SNOMED_CODE, "^(266919005|221000119102)$") ~ "never-smoked",
      str_detect(WD_SNOMED_CODE,
"(281018007|160621008|266928006|266922007|266924008|266923002|360900008|

1092071000000105|1092111000000104|1092031000000108|160624000|138004007|36089004|839
2000|405746006|53896009|160620009|

492191000000103|8517006|735128000|137992000|137988007|48031000119106|266921000|1092
131000000107|160617001|137989004|

707381000000107|266925009|1092041000000104|517211000000106|138005008|137991007|8773
```

```
9003|160607006|160608001|138009002|

160609009|160610004|160611000|160618006|1092131000000100|137990008|485971000000109|
487391000000101|492201000000101|492211000000104)") ~ "current non-smoker",
      str_detect(WD_SNOMED_CODE, "NA") ~ NA_character_,
      TRUE ~ "current smoker"
    )
  ) %>%
  arrange(PATIENT_KEY, desc(CLINICAL_EFFECTIVE_DATE)) %>% group_by(PATIENT_KEY) %>%
slice_head(n = 1) %>% ungroup()

smoking_quant <- SMOKING %>%
  filter(!str_detect(WD_SNOMED_CODE, "^(266919005|221000119102)$")) %>%
  filter(str_detect(RESULT_VALUE_UNITS, regex("day", ignore_case = TRUE))) %>%
  group_by(PATIENT_KEY) %>%
  summarise(average_cigs = round(mean(RESULT_VALUE, na.rm = TRUE)), .groups =
"drop") %>%
  mutate(
    smoker_group = case_when(
      is.na(average_cigs) ~ NA_character_,
      average_cigs < 1    ~ "trivial",
      average_cigs <= 9   ~ "light",
      average_cigs <= 19  ~ "moderate",
      average_cigs >= 20  ~ "heavy",
      TRUE ~ NA_character_
    )
  )

# ================================================
# 5) Comorbidity burden (LTC) & EFI (frailty)
# ================================================
# **Supplementary B.4**: Common LTC combos & counts; **Supplementary B.3**: EFI

LTC_counts <- LTC_unpivotted %>%
  rename_patient_key() %>%
  mutate(
    LTC = if_else(LTC %in% c('SMOKING','PALLIATIVE_CARE'), NA_character_, LTC),
    LTC_clean = LTC |> str_remove("_\\d+$") |> str_replace_all(regex("^HF\\d*$",
ignore_case = TRUE), "HEART_FAILURE")
  ) %>%
  distinct(PATIENT_KEY, WD_REGISTER_STATUS, DIAGNOSIS_DATE, LTC_clean) %>%
  anti_join(non_registered, by = "PATIENT_KEY") %>%
  group_by(PATIENT_KEY) %>% summarise(LTC_count = sum(!is.na(LTC_clean)), .groups =
"drop") %>%
  mutate(
    LTC_count_cat = case_when(
      LTC_count == 0 ~ "0",
      LTC_count == 1 ~ "1",
      LTC_count == 2 ~ "2",
      LTC_count == 3 ~ "3",
```

```
      LTC_count == 4 ~ "4",
      LTC_count >=5  ~ "5+",
      TRUE ~ NA_character_
    )
  )

EFI_filtered <- EFI %>%
  rename_patient_key() %>%
  select(PATIENT_KEY, DOR, WD_REGISTER_STATUS, REPORTING_DATE, EFI_CATEGORY,
EFI_SCORE) %>%
  arrange(PATIENT_KEY, desc(REPORTING_DATE)) %>% group_by(PATIENT_KEY) %>%
slice_head(n = 1) %>% ungroup()

# ================================================
# 6) Primary care contacts (24-month window)
# ================================================
# **Supplementary B.9**: GP visit rates per patient–month (6month windows)

GPvisits_f <- GPvisits %>%
  rename_patient_key() %>%
  filter(!PATIENT_KEY %in% non_registered$PATIENT_KEY) %>%
  filter(!is.na(CORE_CONCEPT_DESCRIPTION)) %>%
  filter(str_detect(CORE_CONCEPT_CODE,
"(_OnPremiseEncounter|_TelephoneConsultation|_GPSurgeryConsultation|_HomeVisit|_Vid
eoConsultation|_Consultation)")) %>%
  arrange(GP_VISIT_DATE) %>%
  distinct(PATIENT_KEY, DOR, GP_VISIT_DATE)

monthly_visits <- GPvisits_f %>%
  mutate(
    ref_month   = floor_date(DOR, unit = "month"),
    visit_month = floor_date(GP_VISIT_DATE, unit = "month"),
    rel_month   = interval(visit_month, ref_month) %/% months(1)
  ) %>%
  filter(rel_month >= 0, rel_month <= 24)

visits_24 <- monthly_visits %>%
  distinct(PATIENT_KEY, GP_VISIT_DATE) %>%
  count(PATIENT_KEY, name = "Visit_Count")

# ================================================
# 7) Selected medications (BNF-grouped; 24 months)
# ================================================
# **Supplementary B.11–B.12**: Incident prescribing across 12 classes; polypharmacy
& Rx rates

specific_meds_f <- specific_meds %>%
  rename_patient_key() %>%
  filter(!PATIENT_KEY %in% non_registered$PATIENT_KEY) %>%
  filter(!is.na(WD_SNOMED_TERM)) %>%
```

```
  mutate(
    med_cat = case_when(
      str_detect(BNF_REFERENCE, "^0407")   ~ "pain killers (opioids, paracetamol)",
      str_detect(BNF_REFERENCE, "^0401")   ~ "anxiolytics",
      str_detect(BNF_REFERENCE, "^0402")   ~ "antipsychotics",
      str_detect(BNF_REFERENCE, "^0501")   ~ "antibiotics",
      str_detect(BNF_REFERENCE, "^0106")   ~ "laxatives",
      str_detect(BNF_REFERENCE, "^0101")   ~ "antacids",
      str_detect(BNF_REFERENCE, "^0103")   ~ "antacids (PPI+H2)",
      str_detect(BNF_REFERENCE, "^1001")   ~ "pain killers (NSAIDs)",
      str_detect(BNF_REFERENCE, "^0603")   ~ "steroids",
      str_detect(BNF_REFERENCE, "^0902")   ~ "supplements (potassium)",
      str_detect(BNF_REFERENCE, "^070401") ~ "alpha-adrenergic blockers",
      str_detect(BNF_REFERENCE, "^0202")   ~ "diuretics",
      TRUE ~ BNF_REFERENCE
    )
  ) %>%
  select(PATIENT_KEY, DOR, WD_SNOMED_TERM, med_cat, CLINICAL_EFFECTIVE_DATE)

poly_meds <- specific_meds_f %>%
  distinct(PATIENT_KEY, med_cat, CLINICAL_EFFECTIVE_DATE) %>%
  count(PATIENT_KEY, name = "meds_count") %>%
  mutate(
    meds_count_cat = case_when(
      meds_count == 1 ~ "1",
      meds_count == 2 ~ "2",
      meds_count == 3 ~ "3",
      meds_count == 4 ~ "4",
      meds_count >=5  ~ "5+",
      TRUE ~ "Unknown"
    ) |> factor(levels = c("1","2","3","4","5+","Unknown"))
  )

# ================================================
# 8) Blood tests (selected SNOMED groups; 24 months)
# ================================================
# **Supplementary B.10**: Test request & abnormal-result rates per patient–month

RDC_BLOOD_RESULTS <- RDC_BLOOD_RESULTS %>% rename_patient_key() %>%
  mutate(WD_SNOMED_CODE = format(WD_SNOMED_CODE, scientific = FALSE) |> str_trim())

map_test <- function(code) {
  case_when(
    str_detect(code,
"^(909321000000105|1000381000000105|1030791000000100|1006591000000104)$") ~ "PSA",
    str_detect(code, "^(391558003|767002|1022541000000102|1110441000000100)$") ~
"WCC",
    str_detect(code, "^(104485008|390955003|1000821000000103|1153477009)$") ~
"albumin",
    str_detect(code,
```

```
"^(135842001|1001371000000100|999651000000107|270980008|55235003|119971000119104|71
1357009)$") ~ "CRP",
    str_detect(code, "^(416838001|1022511000000103|165468009|365649001)$") ~ "ESR",
    str_detect(code, "^(275806002|1022431000000105|271026005)$" ) ~ "Hb",
    str_detect(code,
"^(166610007|313840000|997591000000109|997561000000103|176271000119108|313951004|16
6612004|26165005|365787000|365786009|1021751000000102|999691000000104|1107231000000
106|1107221000000109)$") ~ "bilirubin",
    str_detect(code, "^(61928009|1022651000000100|365632008)$") ~ "Platelet",
    str_detect(code, "^(271234008|390962007|1000621000000104|88810008)$") ~ "ALP",
    str_detect(code,
"^(34608000|389586005|390318006|250637003|201321000000108|1018251000000107|39096100
0|219651000000107|219661000000105|889391000000100|104481004|104482006|1013211000000
103)$") ~ "ALT",
    str_detect(code,
"^(250641004|45896001|889441000000104|1000881000000102|313852007|1000891000000100|3
73679006|1106071000000102|1106681000000105|166669000|1031101000000102)$") ~ "AST",
    str_detect(code,
"^(269821003|401152007|489004|993381000000106|992911000000107)$") ~ "ferritin",
    str_detect(code,
"^(909121000000107|1000461000000109|1027511000000100|80529009|432519008)$") ~
"CA125",
    str_detect(code,
"^(271240001|443796007|390963002|1027931000000101|390966005|1000691000000101|252148
000|312472004|1017391000000108|166702002)$") ~ "calcium",
    str_detect(code,
"^(43396009|371981000000106|144176003|166902009|313835008|365845005|491841000000105
|733830002|567641000000108|999791000000106|1003671000000109|259689004|1013511000000
100)$") ~ "HbA1c",
    str_detect(code,
"^(313936008|113075003|1000731000000107|365756002|365757006)$")~ "creatinine",
    TRUE ~ NA_character_
  )
}

blood_tests <- RDC_BLOOD_RESULTS %>%
  mutate(blood_test = map_test(WD_SNOMED_CODE)) %>%
  filter(!is.na(blood_test)) %>%
  mutate(
    blood_test_grouped = case_when(
      blood_test %in% c("WCC","Hb","Platelet")                    ~ "FBC",
      blood_test %in% c("albumin","bilirubin","ALT","AST","ALP")  ~ "LFT",
      blood_test == "creatinine"                                  ~ "Renal",
      TRUE ~ blood_test
    )
  ) %>%
  anti_join(non_registered, by = "PATIENT_KEY")

sex_tab <- demographics %>% select(PATIENT_KEY, gender =
coalesce(GENDER_DESCRIPTION, GENDER_CODE)) %>% distinct()
```

```
ref_range <- tribble(
  ~blood_test, ~gender, ~min, ~max, ~flag_dir,
  "PSA", "Male", 0, 4, "high",
  "Platelet", "Male", 150, 400, "both",
  "Platelet", "Female", 150, 400, "both",
  "WCC", "Male", 4, 11, "both",
  "WCC", "Female", 4, 11, "both",
  "Hb", "Male", 135, 175, "low",
  "Hb", "Female", 120, 155, "low",
  "HbA1c", "Male", 20, 38, "high",
  "HbA1c", "Female", 20, 38, "high",
  "calcium", "Male", 2.1, 2.6, "high",
  "calcium", "Female", 2.1, 2.6, "high",
  "AST", "Male", 10, 40, "high",
  "AST", "Female", 10, 40, "high",
  "ALT", "Male", 10, 41, "high",
  "ALT", "Female", 10, 41, "high",
  "ALP", "Male", 44, 147, "high",
  "ALP", "Female", 44, 147, "high",
  "bilirubin", "Male", 3, 21, "high",
  "bilirubin", "Female", 3, 21, "high",
  "ferritin", "Male", 30, 300, "both",
  "ferritin", "Female", 13, 150, "both",
  "CA125", "Female", 0, 35, "high",
  "ESR", "Male", 0, 15, "high",
  "ESR", "Female", 0, 20, "high",
  "albumin", "Male", 35, 50, "low",
  "albumin", "Female", 35, 50, "low"
)

blood_flagged <- blood_tests %>%
  left_join(sex_tab, by = "PATIENT_KEY") %>%
  left_join(ref_range, by = c("blood_test" = "blood_test", "gender" = "gender"))
%>%
  filter(!is.na(min)) %>%
  mutate(
    flag = case_when(
      flag_dir == "high" & RESULT_VALUE > max ~ "high",
      flag_dir == "low"  & RESULT_VALUE < min ~ "low",
      flag_dir == "both" & RESULT_VALUE < min ~ "low",
      flag_dir == "both" & RESULT_VALUE > max ~ "high",
      TRUE ~ "normal"
    )
  )

abnormal_results <- blood_flagged %>% filter(flag != "normal")

# ========================================
# 9) Symptoms (OBS; selected groups)
```

```
# ========================================
# **Main Figure 3 (panels a–c)**: symptom trajectories; abnormal labs & weight-loss
associations

OBS_FINDING <- RDC_OBS %>%
  mutate(across(where(is.character), ~str_trim(.))) %>%
  mutate(WD_SNOMED_CODE = format(WD_SNOMED_CODE, scientific = FALSE) |> str_trim())
%>%
  filter(str_detect(CORE_CONCEPT_DESCRIPTION, "finding")) %>%
  rename_patient_key() %>%
  select(-matches("PATIENT_KEY\\.\\.\\d+|CORE_CONCEPT_CODE"))

# Weight loss (SNOMED + measured trend)
symp_weight_loss_code <- RDC_OBS %>%
  filter(str_detect(WD_SNOMED_CODE,
"^(262285001|89362005|267024001|448765001|422868009|198511000000103|426977000|30925
7005|60861008|
139091004|168134000|161832001|267158006|401003006|426977000|699205002|816160009|363
806002|75071000000109|79311000000104|213791000000109|
213801000000108|240261000000105|511461000000103|768571000000103|768581000000101)$")
) %>%
  mutate(weight_change = RESULT_VALUE) %>%
  distinct(PATIENT_KEY, OB_DATE, .keep_all = TRUE)

weight_trend <- RDC_OBS %>%
  filter(RESULT_VALUE_UNITS == "Kg" & WD_SNOMED_TERM == "Body weight" &
RESULT_VALUE > 20) %>%
  arrange(PATIENT_KEY, OB_DATE) %>%
  group_by(PATIENT_KEY) %>%
  mutate(
    Earliest_date = min(OB_DATE),
    Latest_date   = max(OB_DATE),
    weight_change_month = interval(Earliest_date, Latest_date) %/% months(1),
    weight_change_days  = interval(Earliest_date, Latest_date)/days(1),
    first_weight  = first(RESULT_VALUE),
    last_weight   = last(RESULT_VALUE),
    weight_change = abs(last_weight-first_weight),
    weight_trend  = case_when(weight_change > 1 ~ "weight loss", TRUE ~ "no weight
loss")
  ) %>%
  ungroup()

symp_weight_loss_trend <- weight_trend %>%
  filter(weight_trend == "weight loss") %>%
  select(-weight_change_month, -weight_change_days, -first_weight, -last_weight,
-weight_trend, -Earliest_date, -Latest_date) %>%
  distinct(PATIENT_KEY, OB_DATE, .keep_all = TRUE)

symp_weight_loss <- bind_rows(symp_weight_loss_code, symp_weight_loss_trend) %>%
  distinct(PATIENT_KEY, OB_DATE, .keep_all = TRUE) %>%
```

```
  mutate(symptom = "weight loss")

# Other symptom groups (fatigue, abdominal, chest, pain, appetite loss; plus
anaemia/jaundice from labs)
symp_fatigue <- RDC_OBS %>%
  filter(str_detect(WD_SNOMED_CODE , "^(13791008|22496004|84229001|271795006|
                    367391008|248278004|272036004|52663000|60113006|
                    123087006|139124002|139126000|139127009|
                    139512001|139131003|158169005|158170006|
                    158171005|158173008|161868006|161870002|
                    161875007|206767001|206765009|206768006|
                    206769003|206771003|206773000|214264003|224960004|
                    248268002|248270006|248269005|248271005|
                    248274002|248275001|248278004|267031002|
                    267032009|267033004|267160008|271795006|271797003|
                    272036004|272060000|272062008|274236006|
                    274638001|279123003|302758002|314109004|
                    366947008|367172001|367392001|713568000|
                    414631000000100|439201000000104|463761000000103|
                    495751000000103|495771000000107|495781000000109|
                    495801000000105|502201000000109|580521000000101)$")) %>%
  mutate(symptom = "fatigue_malaise")

symp_appetite_loss <- RDC_OBS %>%
  filter(str_detect(WD_SNOMED_CODE, "^(249469002|79890006|64379006|289163007|
                    158268008|158269000|158272007|161826006|192017004|
                    206917003|206916007|206915006|249469002|249468005|

249271002|267024001|267158006|269813009|271816004|39161000000101|
                    41421000000102|46641000000101|78901000000106|43251000000106|
                    451151000000109|496871000000107|496881000000109|)$")) %>%
  mutate(symptom ="loss of appetite")

symp_abdomen <- RDC_OBS %>%
  filter(str_detect(WD_SNOMED_CODE,
                    "^(21522001|248490000|422587007|398032003|236070005|301790009|

9991008|304542004|300359004|300306001|285388000|285387005|371102005|

275297005|271864008|271860004|271681002|271352005|314212008|439469002|271681002|

268941000|267060006|249519007|249497008|249517009|102614006|162038003|119416008|

162068007|162059005|162046002|116289008|83132003|71820002|62315008|60728008|5458600
4|

51197009|43364001|37389005|79922009|225587003|268941000|308903002|405729008|
                    37109400|136571000119109)$" )) %>%
  mutate(
    symptom = case_when(
```

```
      grepl("pain|colic|ache", WD_SNOMED_TERM,  ignore.case = TRUE) ~ "abdominal
pain",
      grepl("diarrhoea|constipation|loose stool", WD_SNOMED_TERM,  ignore.case =
TRUE ) ~ "altered bowel habits",
      grepl("discomfort", WD_SNOMED_TERM,  ignore.case = TRUE ) ~ "abdominal
discomfort",
      grepl("bloating|distension", WD_SNOMED_TERM,  ignore.case = TRUE ) ~
"bloating",
      grepl("nausea|vomit", WD_SNOMED_TERM,  ignore.case = TRUE ) ~ "N & V",
      TRUE ~ "other GI symptom"
    )
  )

symp_chest <- RDC_OBS %>%
  filter(str_detect(WD_SNOMED_CODE,

"^(49727002|267036007|28743005|68154008|11833005|60845006|102589003|161923004|29857
009|

102588006|161924005|161925006|161929000|161939006|161940008|161941007|161947006|

248596009|248604008|272039006|274668005|277907006|277908001|279084009|

281245003|284523002|390871002|391120009|391123006|391124000|391125004|391126003|

810511000000109|297216006|1089971000000104|1089981000000102|1089991000000100|

1090001000000101|1090021000000105|1090011000000104|17216000|72365000|39950000|11833
005|

81953000|73322006|870535009|7142008|315246003|297217002|102587001)$")) %>%
  mutate(
    symptom = case_when(
      grepl("pain", WD_SNOMED_TERM,  ignore.case = TRUE) ~ "chest pain",
      grepl("dyspnoea|breathless", WD_SNOMED_TERM,  ignore.case = TRUE ) ~
"breathlessness",
      grepl("cough",     WD_SNOMED_TERM,  ignore.case = TRUE ) ~ "cough",
      grepl("discomfort",WD_SNOMED_TERM,  ignore.case = TRUE ) ~ "chest
discomfort",
      TRUE ~ "other chest symptoms"
    )
  )

pain_codes <- OBS_FINDING %>% filter(str_detect(WD_SNOMED_TERM,
"(?i)(pain|ache|discomfort)")) %>% distinct(WD_SNOMED_CODE)

symp_pain <- RDC_OBS %>%
  filter(WD_SNOMED_CODE %in% pain_codes$WD_SNOMED_CODE) %>%
  filter(!WD_SNOMED_CODE %in% symp_abdomen$WD_SNOMED_CODE) %>%
  filter(!WD_SNOMED_CODE %in% symp_chest$WD_SNOMED_CODE) %>%
```

```
  filter(!str_detect(WD_SNOMED_CODE,
                     "^(301379001|163729003|287495009|162934003|308927000|
                       305799001|185248009|315014005|305469009|277286006|
                       315526007|279041008)$")) %>%
  mutate(symptom = "pain other than abdominal/chest")

symp_jaundice <- blood_flagged %>% filter(blood_test == "bilirubin", flag ==
"high") %>%
  transmute(PATIENT_KEY, OB_DATE = CLINICAL_EFFECTIVE_DATE, symptom = "jaundice")

symp_anaemia  <- blood_flagged %>% filter(blood_test == "Hb",        flag ==
"low")  %>%
  transmute(PATIENT_KEY, OB_DATE = CLINICAL_EFFECTIVE_DATE, symptom = "anaemia")

symp_abnormal_lab <- blood_flagged %>% filter(flag != "normal") %>%
  transmute(PATIENT_KEY, OB_DATE = CLINICAL_EFFECTIVE_DATE, symptom = "abnormal
lab")

combined_sympt <- bind_rows(
  symp_weight_loss %>%    transmute(PATIENT_KEY, DOR, OB_DATE, symptom),
  symp_fatigue %>%        transmute(PATIENT_KEY, DOR, OB_DATE =
CLINICAL_EFFECTIVE_DATE, symptom),
  symp_abdomen %>%        transmute(PATIENT_KEY, DOR, OB_DATE =
CLINICAL_EFFECTIVE_DATE, symptom),
  symp_chest %>%          transmute(PATIENT_KEY, DOR, OB_DATE =
CLINICAL_EFFECTIVE_DATE, symptom),
  symp_appetite_loss %>%  transmute(PATIENT_KEY, DOR, OB_DATE, symptom),
  symp_abnormal_lab
) %>%
  mutate(
    ref_month = floor_date(DOR, unit = "month"),
    test_month = floor_date(OB_DATE, unit = "month"),
    rel_month = interval(test_month, ref_month) %/% months(1)
  ) %>%
  filter(rel_month >= 0 & rel_month <= 24)

# ================================
# 10) Example figures
# ================================
# **Main Figure 4a** – monthly GP visit rate (case vs control) with 3month moving
average

case_monthly <- monthly_visits %>%
  mutate(diagnosis = if_else(PATIENT_KEY %in% case_keys, "cancer", "non-cancer"))
%>%
  group_by(diagnosis, rel_month) %>%
  summarise(events= n(), exposure_pm=sum(person_months, na.rm=TRUE), rate=
events/exposure_pm, .groups = "drop") %>%
  arrange(diagnosis, rel_month) %>%
  group_by(diagnosis) %>% mutate(MA = as.numeric(moving_average3(rate)) %>%
```

```
ungroup()

p_1 <- ggplot(case_monthly, aes(rel_month, rate, colour = diagnosis)) +
  geom_point(alpha = 0.5) +
  geom_line(aes(y = MA), linewidth = 1.1) +
  scale_x_reverse(breaks = seq(0, 24, by = 3)) +
  labs(title = "Monthly Primary Care Visit Rate", x = "Months before referral", y =
"Rate per person-month") +
  scale_colour_a + fig_theme_a()

suppressWarnings(save_plot(p_1, "fig_visits_monthly_optionA"))

#**Main Figure 3c** — association of weight loss symptom+ abnormal blood test with
diagnostic outcome

combined_common_sympt <- bind_rows(
  symp_weight_loss,     # carries symptom == "weight loss"
  symp_abnormal_lab     # carries symptom == "abnormal lab"
) %>%
  mutate(
    cancer   = PATIENT_KEY %in% case_keys,
    # For weight-loss rows, blood_test/flag are NA; for abnormal-lab they are
present
    test_flag = dplyr::case_when(
      !is.na(blood_test) & !is.na(flag) ~ paste(blood_test, flag, sep = "_"),
      TRUE                              ~ "weight_loss"  # label for the
weight-loss symptom rows before we filter them away
    )
  ) %>%
  group_by(cancer, PATIENT_KEY) %>%
  mutate(
    has_weight_loss = any(symptom == "weight loss"),
    # interaction() returns factors with levels like FALSE.FALSE, etc.
    interaction_dg_weight_loss = interaction(has_weight_loss, cancer, drop = TRUE)
  ) %>%
  ungroup() %>%
  # EXCLUDE the weight-loss rows before counting by test_flag
  filter(symptom != "weight loss") %>%
  group_by(test_flag, has_weight_loss, cancer, interaction_dg_weight_loss) %>%
  summarise(n = n_distinct(PATIENT_KEY), .groups = "drop_last") %>%
  mutate(
    percent = n / sum(n) * 100,
    combo   = interaction_dg_weight_loss,
    combo   = recode(
      combo,
      "TRUE.TRUE"  = "weight loss + cancer",
      "TRUE.FALSE" = "weight loss only",
      "FALSE.TRUE" = "cancer without weight loss",
      "FALSE.FALSE"= "Neither cancer nor weight loss"
    )
```

```
  ) %>%
  ungroup()

# order combos for facet readability
combined_common_sympt$combo <- factor(
  combined_common_sympt$combo,
  levels = c("weight loss + cancer",
             "cancer without weight loss",
             "weight loss only",
             "Neither cancer nor weight loss")
)


)

# Plot (dodged bars)
p_2 <- ggplot(combined_common_sympt, aes(x = test_flag, y = n, fill = combo)) +
  geom_col(position = "dodge") +
  coord_flip() +
  labs(
    y   = "No. of patients",
    x   = "Abnormal blood test",
    fill = "Symptom (Weight Loss)\n& Diagnosis (Cancer)"
  ) +
  facet_wrap(~ combo, ncol = 2, scales = "free_y") +
  theme_bw(base_size = 8) +
  theme(
    strip.text      = element_text(face = "bold"),
    text            = element_text(size = 8, lineheight = 0.9, colour = "black"),
    legend.position = "none"
  )



suppressWarnings(save_plot(p_2, "fig_abnormal_tests_optionA"))

# ================================================================
# 11) Minimal outputs (tables) saved as CSV for manuscript reporting
# ================================================================
# **Supplementary B outputs** — ready-to-paste tables (B.2–B.14)

# Example: cancer subtype (top-5) — supports Results & Supplementary B.7
case_top5 <- cases %>% count(diagnosis_grouped, sort = TRUE) %>% slice_head(n = 5)
readr::write_csv(case_top5, file.path(out_dir, "case_top5.csv"))


# ================================================================================
# 12) LOGISTIC REGRESSION
# ================================================================================
```

```
# Age (decades) from DOR and YOB (robust to absence of earlier age_tab)
age_tab2 <- demographics %>%
  transmute(PATIENT_KEY,
            DOR = ymd(DOR),
            yob = WD_YEAR_MONTH_OF_BIRTH,
            referral_age = round(as.numeric(interval(yob, DOR) / years(1))),
            age_decade = referral_age/10) %>%
  select(PATIENT_KEY, age_decade)

# Gender (prefer description, fall back to code; lower-case)
gender_tab2 <- demographics %>%
  transmute(PATIENT_KEY, gender = tolower(coalesce(GENDER_DESCRIPTION,
GENDER_CODE))) %>%
  distinct()

# Ethnicity (White vs Non-White)
ethnicity_tab2 <- demographics %>%
  transmute(PATIENT_KEY,
            Ethnicity_q = if_else(ETHNIC_CATEGORY_DESCRIPTION == "White",
                                  "White", "Non-White")) %>%
  distinct()

# Symptom burden in the 0–5 months pre-referral
sympt_0to5 <- combined_sympt %>%
  filter(rel_month >= 0, rel_month <= 5) %>%
  count(PATIENT_KEY, name = "sympt_count_total")



# Assemble modelling table
lg_combined_table <- demographics %>%
  select(PATIENT_KEY) %>%
  left_join(age_tab2,      by = "PATIENT_KEY") %>%
  left_join(gender_tab2,   by = "PATIENT_KEY") %>%
  left_join(ethnicity_tab2,by = "PATIENT_KEY") %>%
  left_join(IMD_tab %>% select(PATIENT_KEY, IMD_q), by = "PATIENT_KEY") %>%
  left_join(LTC_counts,    by = "PATIENT_KEY") %>%
  left_join(sympt_0to5,    by = "PATIENT_KEY") %>%
  mutate(
    cancer        = as.integer(PATIENT_KEY %in% case_keys),
    gender_q      = fct_relevel(factor(gender), "female"),
    Ethnicity_q   = fct_relevel(factor(Ethnicity_q), "White"),
    sympt_count_total = replace_na(sympt_count_total, 0L),
    LTC_count         = replace_na(LTC_count, 0L)
  )

# Fit model
model1 <- glm(
  cancer ~ age_decade + IMD_q + gender_q + Ethnicity_q +
           sympt_count_total + LTC_count,
```

```
  data   = lg_combined_table,
  family = binomial
)

# Tidy + export
model1_results <- broom::tidy(model1) %>%
  mutate(OR = exp(estimate),
         CI_low = exp(estimate - 1.96 * std.error),
         CI_high = exp(estimate + 1.96 * std.error))

readr::write_csv(model1_results,
                 file.path(out_dir, "logistic_model1_results.csv"))

# Optional forest-style plot
p_logit <- model1_results %>%
  filter(!grepl("(Intercept)", term, ignore.case = TRUE)) %>%
  ggplot(aes(x = OR, y = reorder(term, OR))) +
  geom_vline(xintercept = 1, linetype = "dashed", colour = "grey60") +
  geom_errorbarh(aes(xmin = CI_low, xmax = CI_high), height = 0.2, colour =
"grey40") +
  geom_point(size = 2.5, colour = "steelblue") +
  scale_x_log10() +
  labs(x = "Odds ratio (log scale)", y = NULL,
       title = "Logistic model (Model 1 – Version A)") +
  theme_minimal(base_size = 11)

ggsave(file.path(fig_dir, "logistic_model1_forest.pdf"), p_logit, width = 6, height
= 4, device = cairo_pdf)

# ============================================================================
# 13) POISSON / NEGATIVE BINOMIAL MODELS
    determinants of pre-referral health care activity
# ============================================================================

# Ensure covariates/codings present for the models
monthly_panel <- monthly_panel %>%
  mutate(
    gender_q    = fct_relevel(factor(tolower(coalesce(gender, GENDER_DESCRIPTION,
GENDER_CODE))), "female"),
    Ethnicity_q = fct_relevel(factor(if_else(coalesce(ethnicity,
WD_ETHNIC_CATEGORY) == "White", "White", "Non-White")), "White")
  )

fit_poisson_nb <- function(df, outcome, overdisp_threshold = 1.2) {
  fml <- as.formula(paste0(
    outcome, " ~ cancer_3m + gender_q + Ethnicity_q + age_decade + EFI_q + IMD_q"
  ))

  pois <- glm(fml, offset = log(person_months), family = poisson(link = "log"),
data = df)
```

```
  over <- sum(residuals(pois, type = "pearson")^2) / pois$df.residual
  if (is.na(over)) over <- 1

  if (over > overdisp_threshold) {
    nb <- MASS::glm.nb(update(fml, . ~ . + offset(log(person_months))), data = df)
    model <- nb; type <- "NB"
  } else {
    model <- pois; type <- "Poisson"
  }

  list(model = model, type = type, overdisp = over)
}

# Fit models across outcomes
poisson_outcomes <- c( "gp_contacts", "meds_count", "test_count", "abnormal_tests")

poisson_tbl <- purrr::map_dfr(poisson_outcomes, function(out) {
  ft <- fit_poisson_nb(monthly_panel, out)
  broom::tidy(ft$model) %>%
    mutate(
      outcome    = out,
      IRR        = exp(estimate),
      CI_low     = exp(estimate - 1.96 * std.error),
      CI_high    = exp(estimate + 1.96 * std.error),
      model_type = ft$type,
      overdisp   = ft$overdisp
    )
})

readr::write_csv(poisson_tbl, file.path(out_dir, "poisson_nb_results.csv"))

# ============================================================================
# 14) event_by_month() + FIGURE 4 PANELS
#    Produces per-month rates, 3-month MA, and saves 4 panels
# ============================================================================

event_by_month <- function(df, outcome) {
  df %>%
    group_by(rel_month, cancer_3m) %>%
    summarise(
      events   = sum(.data[[outcome]], na.rm = TRUE),
      exposure = sum(person_months,     na.rm = TRUE),
      .groups  = "drop"
    ) %>%
    mutate(
      diagnosis = if_else(cancer_3m, "cancer", "non-cancer"),
      rate      = events / exposure
    ) %>%
    arrange(diagnosis, rel_month) %>%
    group_by(diagnosis) %>%
```

```
    mutate(MA = zoo::rollmean(rate, k = 3, fill = NA, align = "right")) %>%
    ungroup()
}

# Build monthly series for each outcome
vis_m   <- event_by_month(monthly_panel, "gp_contacts")
tests_m <- event_by_month(monthly_panel, "test_count")
abn_m   <- event_by_month(monthly_panel, "abnormal_tests")
meds_m  <- event_by_month(monthly_panel, "meds_count")

# A compact plotting helper
fig_rate <- function(df, title, ylab = "Rate (per person-month)") {
  ggplot(df, aes(x = rel_month, y = rate, color = diagnosis)) +
    geom_line(aes(y = MA), linewidth = 1) +
    geom_line(alpha = 0.3) +
    scale_x_reverse(breaks = seq(0, 24, 3)) +
    labs(x = "Months before referral", y = ylab, title = title) +
    theme_minimal(base_size = 11) +
    theme(legend.position = "none",
          panel.grid.minor = element_blank())
}

p_vis  <- fig_rate(vis_m,   "GP visits")
p_test <- fig_rate(tests_m, "Blood test requests")
p_abn  <- fig_rate(abn_m,   "Abnormal tests")
p_meds <- fig_rate(meds_m,  "Prescriptions")


ggsave(file.path(fig_dir, "fig4_visits.pdf"),    p_vis,  width = 6, height = 4,
device = cairo_pdf)
ggsave(file.path(fig_dir, "fig4_tests.pdf"),     p_test, width = 6, height = 4,
device = cairo_pdf)
ggsave(file.path(fig_dir, "fig4_abnormal.pdf"),  p_abn,  width = 6, height = 4,
device = cairo_pdf)
ggsave(file.path(fig_dir, "fig4_meds.pdf"),      p_meds, width = 6, height = 4,
device = cairo_pdf)

# ============================================================================
# 15) KAPLAN–MEIER: REFERRAL → DIAGNOSIS
#    Uses linked cancer cohort; event = diagnosis within an analysis window
# ============================================================================

analysis_window_days <- 90  # adjust if needed

km_df <- noncancer_dx%>%
select(diag_date, PATIENT_KEY)%>%
rename(dx_date=diag_date)%>%
full_join(cancer_cohort, by = c("PATIENT_KEY", "dx_date"),) %>%
  left_join(demographics %>% select(PATIENT_KEY, DOR), by = "PATIENT_KEY") %>%
  mutate(
```

```
    dx=ifelse(PATIENT_KEY%in%cancer_cohort3$PATIENT_KEY, "cancer", "no cancer"),
    DOR = ymd(DOR),
    time_to_dx = as.numeric(pmin(EARLIEST_DIAGNOSIS_DATE, DOR +
analysis_window_days) - DOR, units = "days"),
    event = if_else(!is.na(EARLIEST_DIAGNOSIS_DATE) & EARLIEST_DIAGNOSIS_DATE <=
DOR + analysis_window_days, 1L, 0L)
  ) %>%
  filter(time_to_dx >= 0)

km_fit <- survival::survfit(survival::Surv(time_to_dx, event) ~ 1, data = km_df)

# Save KM plot (base graphics for robustness)

pdf(file.path(fig_dir, "KM_referral_to_diagnosis.pdf"), width = 7, height = 5)
plot(km_fit, xlab = "Days since referral", ylab = "Probability of remaining
undiagnosed",
     main = "Kaplan–Meier: Referral → Diagnosis", lwd = 2, col = "black", mark.time
= TRUE)
abline(h = 0.5, lty = 2, col = "grey60")
dev.off()

# ------------------------------ Done --------------------------
```