Breast Exploratory Analysis

Jonathan Ma

2025-11-15

Table of Contents

1 Introduction

This exploratory data analysis (EDA) is the first step in our project pipeline:

1. Bayesian Hierarchical Modeling
2. Posterior Inference & Diagnostics
3. Clustering of Individual Risk Profiles

We use data from the SEER Program focused on breast cancer cases, structured across three levels:

- **Level 1 (Tumor):** Tumor-specific details (e.g., stage, size, grade)
- **Level 2 (Patient):** Demographics and marital status
- **Level 3 (Region):** County-level metro/rural designations

2 Data Structure and Levels

Level	Variable	Description
Tumor	stage	SEER stage: Localized, Regional, Distant
Tumor	grade	Tumor grade: 1–4 or unknown
Tumor	size	EOD Tumor Size (mm)
Tumor	site	Tumor site (ICD code)
Patient	id	Patient identifier
Patient	age	Age group at diagnosis
Patient	sex	Biological sex
Patient	raceth	Race/ethnicity
Patient	year	Year of diagnosis
Patient	marry	Marital status
Region	region	County-level metro/rural code

```
seer <- read.delim("../data/raw/breastcancer.txt", sep = "\t", header = FALSE,
 stringsAsFactors = FALSE)
```

We replace “Unknown” and “Blank(s)” with NA, and then keep only complete cases:

```
seer_prime <- seer %>%
  transmute(
    id     = V1,
    region = V2,
    stage  = V3,
    age    = V4,
    sex    = V5,
    raceth = V6,
    grade  = V7,
    size   = V10,
    year   = V11,
    marry  = V12,
    site = V8
  )

seer_prime <- seer_prime %>%
  mutate(across(where(is.character), ~na_if(., "Unknown"))) %>%
  mutate(across(where(is.character), ~na_if(., "Blank(s)")))

seer_prime <- seer_prime %>%
  filter(complete.cases(.))

total_cases <- nrow(seer)
complete_cases <- nrow(seer_prime)
percent_complete <- round(100 * complete_cases / total_cases, 2)

cat("Complete cases:", complete_cases, "out of", total_cases,
    sprintf("(%s%% complete)\n", percent_complete))

## Complete cases: 209123 out of 5149008 (4.06% complete)
```

we have 209123 cases to work with (4.06% of the whole thing) after removing “Unknown” and “Blank(s)” Values.

Further cleaning is done via tabling all variables.

2.1 Cancer Site

We want to filter for top ten cancer sites (breast, lung, etc). We exclude 998 and 999 because they are unknown. The code is structured so we can switch sites.

```
seer_prime <- seer_prime %>%
  filter(!(site %in% c("998", "999")))

top_sites <- seer_prime %>%
  count(site, sort = TRUE) %>%
  slice_head(n = 10)

seer_prime <- seer_prime %>%
  filter(site %in% top_sites$site)

table(seer_prime$site)

##
##  010  012  015  018  020  025  030  035  040  050
## 4269 3940 5738 3284 5440 4994 6046 3768 4953 4415
```

2.2 Region

REGION: collapse and recode into ordered, interpretable regional factors, removes the Alaska/Hawaii “unknown” entries.

```
seer_prime <- seer_prime %>%
  mutate(
    region = case_when(
      str_detect(region, "1 million") ~ "Large",
      str_detect(region, "lt 250") | str_detect(region, "250,000") ~ "Small",
      str_detect(region, "adjacent") | str_detect(region, "not adjacent") ~ "
Nonmetro",
      TRUE ~ NA_character_
    ),
    region = factor(region, levels = c("Large", "Small", "Nonmetro"))
  )
table(seer_prime$region)

##
##    Large    Small  Nonmetro
##    39835     2410     4497
```

2.3 Tumor Stage

STAGE: restrict to tumors that are invasive (drop In situ, Unknown/unstaged).

```
seer_prime <- seer_prime %>%
  mutate(
    stage = case_when(
      stage %in% c("Localized", "Regional", "Distant") ~ stage,
      TRUE ~ NA_character_
    ),
    stage = factor(stage, levels = c("Localized", "Regional", "Distant"))
  )
table(seer_prime$stage)

##
## Localized  Regional   Distant
##     26446     13382      3918
```

2.4 Tumor Grade

GRADE: extract numeric grade 1–4; drop cell-line entries (“B-cell”, “T-cell” etc.). Convert them to Start (1/2) and End (3/4) since our model is logistic

```
seer_prime <- seer_prime %>%
  mutate(
    grade = case_when(
      str_detect(grade, "I$") ~ "1",
      str_detect(grade, "II$") ~ "2",
      str_detect(grade, "III$") ~ "3",
      str_detect(grade, "IV$") ~ "4",
      TRUE ~ NA_character_
    ),
    grade = case_when(
      grade %in% c("1", "2") ~ "Start",
      grade %in% c("3", "4") ~ "End",
      TRUE ~ NA_character_
    ),
    grade = factor(grade, levels = c("Start", "End"), ordered = FALSE)
  )
table(seer_prime$grade)

## 
## Start   End 
## 42877  3650
```

2.5 Tumor Size

SIZE: Making this numeric and then filtering implausible values, we can bin these (“≤20mm”, “21–50mm”, “51–100mm”, “>100mm”) if needed.

```
seer_prime <- seer_prime %>%
  mutate(
    size = as.numeric(size),
    size = ifelse(size > 0 & size <= 200, size, NA)
  )
summary(seer_prime$size)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   10.00   23.00   30.00   34.66   44.00   99.00    6305
```

2.6 Marital Status

MARRY: Making this binary, only married or unmarried.

```
seer_prime <- seer_prime %>%
  mutate(
    marry = case_when(
      str_detect(marry, "Married") ~ "Married",
      TRUE ~ "Unmarried"
    ),
    marry = factor(marry)
  )
table(seer_prime$marry)

## 
##   Married Unmarried 
##     28032     18815
```

2.7 Race

Recode RACETH to be three levels

```
seer_prime <- seer_prime %>%
  mutate(
    raceth = case_when(
      raceth == "White" ~ "W",
      raceth == "Black" ~ "B",
      TRUE ~ "O"
    ),
    raceth = factor(raceth, levels = c("W", "B", "O"))
  )
table(seer_prime$raceth)

## 
##     W     B     O 
## 36288  3978  6581
```

2.8 Complete Summary of available data

Seeing how many cases are available now:

```
seer_prime <- seer_prime %>% filter(complete.cases(id, region, stage, age, se
x, raceth, grade, size, site, marry, year))
cat("Remaining complete cases:", nrow(seer_prime), "\n")

## Remaining complete cases: 37530
```

3 Hierarchal Encoding

To do hierarchal models in R, identifiers must be encoded as numeric integers (e.g., 1, 2, ..., N) for levels like REGION and ID.

```
seer_prime <- seer_prime %>%
  mutate(
    regionid = as.integer(factor(region)),
    patientid = as.integer(factor(id))
  )
```

Setting factor and checking data structure.

```
seer_prime <- seer_prime %>%
  mutate(
    age = factor(age),
    sex = factor(sex),
    raceth = factor(raceth),
  )
```

Inspect the data:

```
str(seer_prime)

## 'data.frame':    37530 obs. of  13 variables:
##  $ id       : int  1028 1034 1152 2172 4489 5518 9181 9314 9415 10092 ...
##  $ region   : Factor w/ 3 levels "Large","Small",..: 1 1 1 1 1 1 1 1 1 1
 ...
##  $ stage    : Factor w/ 3 levels "Localized","Regional",..: 1 1 1 1 1 1 1
2 1 2 ...
##  $ age      : Factor w/ 20 levels "00 years","01-04 years",..: 14 16 17 19
 15 13 17 18 18 17 ...
##  $ sex      : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 1 1 1 1 ...
##  $ raceth   : Factor w/ 3 levels "W","B","O": 1 2 1 3 1 2 3 1 3 1 ...
##  $ grade    : Factor w/ 2 levels "Start","End": 1 1 1 2 1 1 1 1 2 1 ...
##  $ size     : num  30 30 38 27 22 49 22 50 27 27 ...
##  $ year     : int  2017 2016 2016 2017 2017 2016 2017 2017 2017 2016 ...
##  $ marry    : Factor w/ 2 levels "Married","Unmarried": 1 2 1 2 2 2 1 1 2
2 ...
##  $ site     : chr  "020" "025" "035" "020" ...
##  $ regionid : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ patientid: int  1 2 3 4 5 6 7 8 9 10 ...
```

4 Saving Final Data

Code here so you can save it as CSV: 37530 observations of 13 variables

```
write.csv(seer_prime, "../data/clean/seer_nov13.csv", row.names = FALSE)
```