# ali_sim_params

Sarah Lotspeich

2024-02-26

```
## Read in summary data (one row per patient)
summ_data = read.csv("~/Documents/Allostatic_load_audits/summary_data.csv") |>
  dplyr::mutate(ANY_ENCOUNTERS = as.numeric(NUM_ENCOUNTERS > 0)) ### Create indicator of >= 1 healthcar

## Subset to only the variables needed for the analysis
summ_data = summ_data |>
  select(PAT_MRN_ID, ANY_ENCOUNTERS, AGE_AT_ENCOUNTER, SEX,
         CREAT_C, ALB, BMI, BP_SYSTOLIC, BP_DIASTOLIC, A1C, CHOL, TRIG, CRP, HCST)

## Create a long version with one row per variable per patient
summ_data_long = summ_data |>
  gather(key = "COMP",
         value = "VAL",
         -c(1:4))

## Merge in the stratification values
strat_vals = read.csv("~/Documents/Allostatic_load_audits/Audit_Protocol/strat_vals.csv")
summ_data_long = summ_data_long |>
  left_join(strat_vals) |>
  mutate(POINT = case_when(
    INEQ == ">" ~ as.numeric(VAL > STRAT),
    INEQ == ">=" ~ as.numeric(VAL >= STRAT),
    INEQ == "<" ~ as.numeric(VAL < STRAT)
  ))
```

```
## Joining with ‘by = join_by(SEX, COMP)‘
```

```
## Aggregate the stratified components by patient
summ_data = summ_data_long |>
  group_by(PAT_MRN_ID, ANY_ENCOUNTERS, AGE_AT_ENCOUNTER, SEX) |>
  summarize(ALI_STAR_NUM = sum(POINT, na.rm = TRUE), ### Number of components equal to 1
            ALI_STAR_DENOM = sum(!is.na(POINT)) ### Number of non-missing components
            ) |>
  mutate(ALI_STAR = ALI_STAR_NUM / ALI_STAR_DENOM,
         ALI_STAR_RESCALED = ALI_STAR / 0.1 ### Create ALI_STAR_RESCALED = ALI_STAR / 0.1
         )
```

```
## ‘summarise()‘ has grouped output by ’PAT_MRN_ID’, ’ANY_ENCOUNTERS’,
## ’AGE_AT_ENCOUNTER’. You can override using the ‘.groups‘ argument.
```

### Distribution of age at encounter

```
## Model age at first encounter (in 1-year increments) with a Poisson distribution
glm(formula = AGE_AT_ENCOUNTER ~ 1,
    family = poisson,
    data = summ_data) |>
  coefficients() |>
  exp()
```

```
## (Intercept)
##      45.662
```

```
### Rescale age at first encounter to be in 5-year increments
summ_data = summ_data |>
  ### Create AGE_AT_ENCOUNTER5 = AGE_AT_ENCOUNTER / 5
  dplyr::mutate(AGE_AT_ENCOUNTER5 = AGE_AT_ENCOUNTER / 5)

## Model age at first encounter (in 5-year increments) with a Poisson distribution
glm(formula = AGE_AT_ENCOUNTER5 ~ 1,
    family = poisson,
    data = summ_data) |>
  coefficients() |>
  exp()
```

```
## (Intercept)
##      9.1324
```

```
### Rescale age at first encounter to be in 10-year increments
summ_data = summ_data |>
  ### Create AGE_AT_ENCOUNTER10 = AGE_AT_ENCOUNTER / 10
  dplyr::mutate(AGE_AT_ENCOUNTER10 = AGE_AT_ENCOUNTER / 10)

## Model age at first encounter (in 10-year increments) with a Poisson distribution
glm(formula = AGE_AT_ENCOUNTER10 ~ 1,
    family = poisson,
    data = summ_data) |>
  coefficients() |>
  exp()
```

```
## (Intercept)
##      4.5662
```

### Naive model of ALI* and healthcare utilization

```
## Naive model parameters (age in 10-year increments)
naive_mod = glm(formula = ANY_ENCOUNTERS ~ ALI_STAR + AGE_AT_ENCOUNTER10,
                family = "binomial",
                data = summ_data)
summary(naive_mod)
```

```
##
## Call:
## glm(formula = ANY_ENCOUNTERS ~ ALI_STAR + AGE_AT_ENCOUNTER10,
##     family = "binomial", data = summ_data)
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.56762    0.25149  -6.233 4.57e-10 ***
## ALI_STAR            0.94523    0.34183   2.765  0.00569 **
## AGE_AT_ENCOUNTER10  0.10266    0.05222   1.966  0.04928 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1250.7  on 999  degrees of freedom
## Residual deviance: 1234.7  on 997  degrees of freedom
## AIC: 1240.7
##
## Number of Fisher Scoring iterations: 4
```

```
## Naive model parameters (age in 10-year increments)
naive_mod = glm(formula = ANY_ENCOUNTERS ~ ALI_STAR_RESCALED + AGE_AT_ENCOUNTER10,
                family = "binomial",
                data = summ_data)
summary(naive_mod)
```

```
##
## Call:
## glm(formula = ANY_ENCOUNTERS ~ ALI_STAR_RESCALED + AGE_AT_ENCOUNTER10,
##     family = "binomial", data = summ_data)
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.56762    0.25149  -6.233 4.57e-10 ***
## ALI_STAR_RESCALED   0.09452    0.03418   2.765  0.00569 **
## AGE_AT_ENCOUNTER10  0.10266    0.05222   1.966  0.04928 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1250.7  on 999  degrees of freedom
## Residual deviance: 1234.7  on 997  degrees of freedom
## AIC: 1240.7
##
## Number of Fisher Scoring iterations: 4
```

### Naive proportions "yes" and missing data per ALI component

```
## probability of stressor = YES
summ_data_long |>
  group_by(COMP) |>
  summarize(PROP_YES = mean(POINT, na.rm = TRUE),
            PROP_MISS = mean(is.na(POINT)))
```

```
## # A tibble: 10 x 3
##    COMP          PROP_YES PROP_MISS
##    <chr>            <dbl>     <dbl>
##  1 A1C              0.249     0.494
##  2 ALB              0.987     0.153
##  3 BMI              0.455     0.002
##  4 BP_DIASTOLIC     0.058     0
##  5 BP_SYSTOLIC      0.145     0
##  6 CHOL             0.314     0.213
##  7 CREAT_C          0.25      0.996
##  8 CRP              0.311     0.955
##  9 HCST             0         0.983
## 10 TRIG             0.332     0.213
```

## SMLE model of ALI and healthcare utilization (pilot + wave 1)

```
## Read in long audit data (one row per patient per audited variable)
audit_dat = read.csv("~/Documents/ali-data/summary_data_audit1.csv")

## Create a long version with one row per variable per patient
audit_dat_long = audit_dat |>
  gather(key = "COMP",
         value = "VAL",
         -c(1))

## Merge in the stratification values
strat_vals = strat_vals |>
  filter(SEX == "Female") |>
  select(-SEX)
audit_dat_long = audit_dat_long |>
  left_join(strat_vals) |>
  mutate(POINT = case_when(
    INEQ == ">" ~ as.numeric(VAL > STRAT),
    INEQ == ">=" ~ as.numeric(VAL >= STRAT),
    INEQ == "<" ~ as.numeric(VAL < STRAT)
  ))
```

```
## Joining with `by = join_by(COMP)`
```

```
## Aggregate the stratified components by patient
audit_dat = audit_dat_long |>
  group_by(PAT_MRN_ID) |>
  summarize(ALI_NUM = sum(POINT, na.rm = TRUE), ### Number of components equal to 1
            ALI_DENOM = sum(!is.na(POINT)) ### Number of non-missing components
            ) |>
  mutate(ALI = ALI_NUM / ALI_DENOM,
         ALI_RESCALED = ALI / 0.1 ### Create ALI_RESCALED = ALI / 0.1
  )

## Merge with summ_data
summ_data = summ_data |>
  left_join(audit_dat)
```

```
## Joining with `by = join_by(PAT_MRN_ID)`
```

```
# Estimate parameters using Phase IIa audits + the rest of Phase I ---------
## Setup B-splines
B = splines::bs(x = summ_data$ALI_STAR,
                df = 8,
                Boundary.knots = range(summ_data$ALI_STAR),
                intercept = TRUE,
                degree = 3)
colnames(B) = paste0("bs", seq(1, 8))
summ_data = summ_data |>
  bind_cols(B) |>
  data.frame()
```

```r
### Fit SMLE ------------------------------------------------------------
library(sleev)
suppressMessages(fit <- logistic2ph(
  Y_unval = NULL,
  Y = "ANY_ENCOUNTERS",
  X_unval = "ALI_STAR",
  X = "ALI",
  Z = "AGE_AT_ENCOUNTER10",
  Bspline = colnames(B),
  data = summ_data,
  hn_scale = 1,
  noSE = FALSE,
  TOL = 1e-04,
  MAX_ITER = 1000
))
fit$coefficients
```

```
##                       Estimate         SE Statistic      p-value
## Intercept          -1.9309188 0.32092231 -6.016779 1.779213e-09
## ALI                 1.8779397 0.27712564  6.776492 1.231282e-11
## AGE_AT_ENCOUNTER10  0.0991235 0.04488015  2.208627 2.720060e-02
```

```r
# Estimate parameters using Phase IIa audits + the rest of Phase I ---------
## Setup B-splines
summ_data = summ_data[, -grep(pattern = "bs", x = colnames(summ_data))]
B = splines::bs(x = summ_data$ALI_STAR_RESCALED,
                df = 8,
                Boundary.knots = range(summ_data$ALI_STAR_RESCALED),
                intercept = TRUE,
                degree = 3)
colnames(B) = paste0("bs", seq(1, 8))
summ_data = summ_data |>
  bind_cols(B)

summ_data = data.frame(summ_data)

### Fit SMLE ------------------------------------------------------------
suppressMessages(fit <- logistic2ph(
  Y_unval = NULL,
  Y = "ANY_ENCOUNTERS",
  X_unval = "ALI_STAR_RESCALED",
  X = "ALI_RESCALED",
  Z = "AGE_AT_ENCOUNTER10",
  Bspline = colnames(B),
  data = summ_data,
  hn_scale = 1 / 2, #### since the parameter estimates are smaller, need smaller perturbation for SEs
  noSE = FALSE,
  TOL = 1e-04,
  MAX_ITER = 1000
))
fit$coefficients
```

```
##                       Estimate         SE  Statistic      p-value
```

```
## Intercept          -1.9309188 0.02259760 -85.447951 0.000000000
## ALI_RESCALED        0.1877940 0.05999031   3.130405 0.001745653
## AGE_AT_ENCOUNTER10  0.0991235 0.04150285   2.388354 0.016924040
```