Session 3.2

Louisa Smith

Why do we need time-varying weights?

In our cloning approach, people get "censored" when they deviate from their assigned treatment strategy.

- This censoring happens at different times for different people

- We have to weight the uncensored observations by the inverse probability of remaining uncensored *up to that time point*

Why time-varying weights?

The probability of remaining uncensored changes over time as more people deviate from the strategy.

Consider our “treat at week 12” strategy:

- Week 8: Nearly everyone still following strategy (high probability of remaining uncensored)
- Week 11: Some people already treated early (lower probability)
- Week 15: Many people have deviated (much lower probability for those remaining)

People contributing data at week 15 must be weighted more heavily because they represent not just themselves, but also those who would have had similar outcomes but were censored earlier.

Data example: two individuals

Let's follow two people assigned to "treat at week 12" strategy:

ID	Maternal age	Bleeding starts	Exposed	Adheres to strategy	Outcome
101	28	10	Never	No	Censored
102	32	10	Week 12	Yes	Yes

- **Person 101**: Never treated → censored at target week 12
- **Person 102**: Treated at week 12 → follows strategy, contributes outcome data

Person 102 must be weighted to represent both themselves and people like Person 101 who would have had similar outcomes

Cox regression approach (exercise 7)

Data structure: Interval survival format

- Each person contributes intervals where censoring could occur
- Variables: `time_in`, `time_out`, `censor` (event indicator)

```
# A tibble: 5 × 7
     ID time_in time_out maternal_age bleeding censor outcome
  <dbl>   <dbl>    <dbl>         <dbl>    <dbl>  <dbl>   <dbl>
1   101       8     10              28        0      0       0
2   101      10     12              28        1      1       0
3   102       8     10              32        0      0       0
4   102      10     12              32        1      0       0
5   102      12     16.8            32        1      0       1
```

Person 101 is censored in interval (10, 12]. Person 102 has the outcome in interval (12, 16.8].

Cox regression model equation

The Cox model estimates the hazard of censoring at time t:

$$h_c(t|X) = h_{c0}(t)\exp(\beta_1 X_1 + \beta_2 X_2 + \ldots)$$

Where:

- $h_c(t|X)$ = hazard of censoring at time t given covariates X
- $h_{c0}(t)$ = baseline hazard (unspecified)
- β = log hazard ratios for predictors of censoring

Survival probability: remaining uncensored

$$S_c(t|X) = \exp\left(-\int_0^t h_c(u|X)du\right)$$

- While the baseline hazard $h_{c0}(t)$ is not specified in the model, it must ultimately be estimated to compute survival probabilities. The `survival` package in R does this automatically using the Breslow estimator.
- With interval data we will actually get interval survival estimates and need to multiply them together to get the cumulative survival

Weight: $w(t) = \frac{1}{S_c(t|X)}$ = inverse probability of remaining uncensored

Weight calculation: cox approach

Step 1: Fit Cox model for censoring in each clone

```
mod_cens <- coxph(Surv(time_in, time_out, censor) ~ maternal_age + bleeding
                  data = clone_long, subset = outcome == 0)
```

Step 2: Extract interval survival probabilities

```
clone_weighted <- clone_long |>
  mutate(.fitted = predict(mod_cens, type = "survival")) |>
  group_by(ID) |>
  arrange(time_in) |>
  mutate(
    # interval survival probabilities
    p_uncens = lag(.fitted, default = 1),
    # cumulative probability of remaining uncensored
    p_uncens_cumulative = cumprod(p_uncens),
    # inverse probability weight
    weight = 1 / p_uncens_cumulative
  )
```

Pooled logistic regression approach (exercise 8)

Data structure: Long format with weekly observations

- Each person contributes one row per week at risk (or whatever time scale)
- Variables: `week` (every one), `censor_now` (0/1 for censored this week)

```
# A tibble: 13 × 6
      ID  week maternal_age bleeding censor_now outcome_now
   <dbl> <dbl>        <dbl>    <dbl>      <dbl>       <dbl>
 1   101     8           28        0          0           0
 2   101     9           28        0          0           0
 3   101    10           28        1          0           0
 4   101    11           28        1          1           0
 5   102     8           32        0          0           0
 6   102     9           32        0          0           0
 7   102    10           32        1          0           0
 8   102    11           32        1          0           0
 9   102    12           32        1          0           0
10   102    13           32        1          0           0
11   102    14           32        1          0           0
12   102    15           32        1          0           0
13   102    16           32        1          0           1
```

Pooled logistic regression model equation

The logistic model estimates the probability of censoring in week t:

$$\text{logit}(P(C_t = 1|C_{t-1} = 0, X_t)) = \alpha_t + \beta_1 X_{1t} + \beta_2 X_{2t} + \ldots$$

Where: - C_t = indicator of censoring in week t - α_t = week-specific intercepts (baseline hazard) - β = log odds ratios for predictors of censoring

Survival probability:

$$S_c(t|X) = \prod_{k=1}^{t}(1 - P(C_k = 1|C_{k-1} = 0, X_k))$$

Weight: $w(t) = \frac{1}{S_c(t|X)}$ = inverse probability of remaining uncensored

Weight calculation: pooled logistic approach

Step 1: Fit logistic model for weekly censoring probability

```
mod_cens <- glm(censor_now ~ factor(week) + maternal_age + bleeding,
                data = clone_long_weekly, family = binomial())
```

Step 2: Calculate cumulative probability of remaining uncensored

```
clone_weighted <- clone_long_weekly |>
  mutate(
    # probability of censoring this week
    p_censor_week = predict(mod_cens, type = "response"),
    # probability of remaining uncensored this week
    p_uncens_week = 1 - p_censor_week
  ) |>
  group_by(ID) |>
  mutate(
    # cumulative probability of remaining uncensored
    p_uncens_cumulative = cumprod(p_uncens_week),
    # inverse probability weight
    weight = 1 / p_uncens_cumulative
  )
```

Comparison of approaches

Aspect	Cox regression	Pooled logistic
Data size	Generally smaller (interval format)	Larger (weekly format)
Baseline hazard	Not specified	Must be modeled
Flexibility	Semi-parametric	Fully parametric
Time modeling	Automatic	Manual (e.g., splines, indicators)

Both produce valid inverse probability weights when models are correctly specified.

Practical considerations

Time scale:

- What time scale makes most sense for your data / won't be overly computationlly intensive?

Model checking:

- Make sure things look reasonable
- Pooled logistic: Check fit of baseline hazard function

Extreme weights:

- Both approaches can produce very large weights, particularly when multiplied for long time periods
- Consider weight truncation or stabilization
- Examine distribution of weights before outcome analysis