

# Time Series Lab 2

AUTHOR

Nhu Vo

PUBLISHED

November 17, 2024

```
# Load the necessary packages
library(haven)
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

```
library(stringr)

# Function to dynamically get the latest biweekly file based on the week number in the filename
get_latest_biweekly_file <- function(directory) {
  # List all files in the specified directory
  files <- list.files(directory, pattern = "stata_widedata_week_\\d+_\\d{2}[a-zA-Z]+\\d{4}.dta",
    full.names = TRUE)

  # Extract the week number from each filename
  week_numbers <- as.numeric(str_extract(files, "(?<=week_)\\d+"))

  # Find the file with the highest week number
  latest_file <- files[which.max(week_numbers)]
  return(latest_file)
}

# Step 1: Define paths and load the baseline and latest biweekly datasets
baseline_data <- read_dta("P:/Dynamics of Contra in Eswatini/1. Baseline code and data/final data.dta")
latest_biweekly_file <- get_latest_biweekly_file("P:/Dynamics of Contra in Eswatini/2. Biweekly/5")
biweekly_data <- read_dta(latest_biweekly_file)

# Step 2: Select relevant columns in biweekly data (all columns that start with specific patterns)
biweekly_subset <- biweekly_data %>%
  select(caseid,
    starts_with("relupdate_w"),
    starts_with("cursexrel_w"),
    starts_with("curromrel_w"),
    starts_with("final_outcome_w"),
    starts_with("contmethuse_w"), # Include contraception method use
    starts_with("cmethouse_2wk")) # Include contraception method use in the past 2 weeks
```

```

# Step 3: Merge the baseline and biweekly data on 'id' from baseline and 'caseid' from biweekly
merged_data <- baseline_data %>%
  inner_join(biweekly_subset, by = c("id" = "caseid"))

# Step 4: Remove rows where relstartyr or relstartmonth is missing
merged_data <- merged_data %>%
  filter(!is.na(relstartyr) & !is.na(relstartmonth))

# Step 5: Remove rows where relstartmonth indicates an unknown month (assuming -8 indicates "Don't know")
if (!is.numeric(merged_data$relstartmonth)) {
  merged_data$relstartmonth <- as.numeric(as.character(merged_data$relstartmonth))
}
merged_data <- merged_data %>%
  filter(relstartmonth != -8)

# Step 6: Calculate the baseline relationship duration in months and weeks
# Define the end date as April 2024
end_month <- 4
end_year <- 2024

# Calculate the duration in months and then convert to weeks
merged_data <- merged_data %>%
  mutate(
    rls_baseline_duration_months = (end_year - relstartyr) * 12 + (end_month - relstartmonth),
    rls_baseline_duration_weeks = rls_baseline_duration_months * 4.345 # Approximate weeks per month
  )

# Step 7: Filter the dataset to include only the necessary columns, ensuring 'id' is retained
output_data <- merged_data %>%
  select(id, relstartyr, relstartmonth, rls_baseline_duration_months, rls_baseline_duration_weeks,
         starts_with("relupdate_w"), starts_with("cursexrel_w"), starts_with("curromrel_w"),
         starts_with("final_outcome_w"), starts_with("contmethuse_w"), starts_with("cmethouse_2w"))

# Display the output to verify the result
print("Final output (first 6 rows):")

```

```
[1] "Final output (first 6 rows):"
```

```
print(head(output_data))
```

```
# A tibble: 6 × 101
```

	id	relstartyr	relstartmonth	rls_baseline_duration... <sup>1</sup>	rls_baseline_duration... <sup>2</sup>
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	235	2019	2 [2. Februar...	62	269.
2	287	2023	12 [12. Decemb...	4	17.4
3	37	2018	6 [6. June]	70	304.
4	196	2023	3 [3. March]	13	56.5

```

5  237      2023  6 [6. June]          10          43.4
6  194      2018  3 [3. March]        73          317.
# i abbreviated names: 1rls_baseline_duration_months,
#   2rls_baseline_duration_weeks
# i 96 more variables: relupdate_w1 <dbl+lbl>, relupdate_w2 <dbl+lbl>,
#   relupdate_w3 <dbl+lbl>, relupdate_w4 <dbl+lbl>, relupdate_w5 <dbl+lbl>,
#   relupdate_w6 <dbl+lbl>, relupdate_w7 <dbl+lbl>, relupdate_w8 <dbl+lbl>,
#   relupdate_w9 <dbl+lbl>, relupdate_w10 <dbl+lbl>, relupdate_w11 <dbl+lbl>,
#   relupdate_w12 <dbl+lbl>, relupdate_w13 <dbl+lbl>, ...

```

```

# Step 8: Initialize new columns in output_data for each week
weekCount <- max(as.numeric(str_extract(names(output_data), "\\d+$")), na.rm = TRUE) # Automatic

# Initialize the columns for each week and set up baseline columns (including `incremented_w0`)
output_data$incremented_w0 <- 1 # Initialize incremented_w0 for baseline
output_data$relationship_duration_w0 <- output_data$rls_baseline_duration_weeks # Set baseline relationship duration

for (week in 1:weekCount) {
  output_data[[paste0("relationship_duration_w", week)]] <- NA
  output_data[[paste0("incremented_w", week)]] <- 0
}

# Step 9: Iterate over each row in output_data
for (i in 1:nrow(output_data)) {
  # Initialize baseline week duration
  prevWeekDur <- output_data$relationship_duration_w0[i]

  for (week in 1:weekCount) {
    # Set default incremented_w for the current week to 0
    output_data[[paste0("incremented_w", week)]] [i] <- 0

    # Ensure `relupdate_w` is not missing before evaluating the conditions
    relupdate_val <- output_data[[paste0("relupdate_w", week)]] [i]
    cursexrel_val <- output_data[[paste0("cursexrel_w", week)]] [i]
    curromrel_val <- output_data[[paste0("curromrel_w", week)]] [i]

    # Condition 1: If relupdate_w equals 0, reset relationship_duration_w to 0
    if (!is.na(relupdate_val) && relupdate_val == 0) {
      output_data[[paste0("relationship_duration_w", week)]] [i] <- 0
      prevWeekDur <- 0
    }

    # Condition 2: If relupdate_w, cursexrel_w, or curromrel_w equals 1, increment relationship_duration_w by 2
    } else if (!is.na(relupdate_val) && relupdate_val == 1 ||
              (!is.na(cursexrel_val) && cursexrel_val == 1) ||
              (!is.na(curromrel_val) && curromrel_val == 1)) {
      output_data[[paste0("relationship_duration_w", week)]] [i] <- prevWeekDur + 2
      prevWeekDur <- output_data[[paste0("relationship_duration_w", week)]] [i]
      output_data[[paste0("incremented_w", week)]] [i] <- 1
    }

    # Condition 3: If relupdate_w is NA, check future weeks and previous increment status

```

```

} else if (is.na(relupdate_val)) {
  prevWeekIncremented <- output_data[[paste0("incremented_w", week - 1)]] [i]

  # Restrict future columns to avoid out-of-bounds error
  futureRelUpdates <- output_data[i, grep("^relupdate_w", names(output_data))] %>%
    unlist() %>% .[(week + 1):min(length(.), weekCount)]
  futureSexRel <- output_data[i, grep("^cursexrel_w", names(output_data))] %>%
    unlist() %>% .[(week + 1):min(length(.), weekCount)]
  futureRomRel <- output_data[i, grep("^curromrel_w", names(output_data))] %>%
    unlist() %>% .[(week + 1):min(length(.), weekCount)]

  # Find the next non-NA relupdate_w value
  nextNonNAUpdate <- tryCatch({
    nextNonNA <- na.omit(futureRelUpdates)[1]
    ifelse(!is.na(nextNonNA), nextNonNA, NA)
  }, error = function(e) NA)

  # Sub-condition: if nextNonNAUpdate is 0, do nothing (leave relationship_duration_w as NA)
  if (!is.na(nextNonNAUpdate) && nextNonNAUpdate == 0) {
    next

  # Sub-condition: if nextNonNAUpdate is 1 and previous week was incremented, increment duration
  } else if (!is.na(nextNonNAUpdate) && nextNonNAUpdate == 1 && prevWeekIncremented == 1) {
    output_data[[paste0("relationship_duration_w", week)]] [i] <- prevWeekDur + 2
    prevWeekDur <- output_data[[paste0("relationship_duration_w", week)]] [i]
    output_data[[paste0("incremented_w", week)]] [i] <- 1

  # Handle cases where all future relupdate_w values are NA
  } else if (is.na(nextNonNAUpdate)) {
    if ((1 %in% futureSexRel || 1 %in% futureRomRel) && prevWeekIncremented == 1) {
      output_data[[paste0("relationship_duration_w", week)]] [i] <- prevWeekDur + 2
      prevWeekDur <- output_data[[paste0("relationship_duration_w", week)]] [i]
      output_data[[paste0("incremented_w", week)]] [i] <- 1
    }
  }
}
}
}
}

# Display the modified output_data to verify the transformations
print("Final output (first 6 rows):")

```

```
[1] "Final output (first 6 rows):"
```

```
print(head(output_data))
```

```
# A tibble: 6 × 135
```

```
id relstartyr relstartmonth rls_baseline_duratio...1 rls_baseline_duratio...2
```

	<dbl>	<dbl>	<dbl+lbl>	<dbl>	<dbl>
1	235	2019	2 [2. Februar...	62	269.
2	287	2023	12 [12. Decemb...	4	17.4
3	37	2018	6 [6. June]	70	304.
4	196	2023	3 [3. March]	13	56.5
5	237	2023	6 [6. June]	10	43.4
6	194	2018	3 [3. March]	73	317.

```
# i abbreviated names: ^rls_baseline_duration_months,
#   ^rls_baseline_duration_weeks
# i 130 more variables: relupdate_w1 <dbl+lbl>, relupdate_w2 <dbl+lbl>,
#   relupdate_w3 <dbl+lbl>, relupdate_w4 <dbl+lbl>, relupdate_w5 <dbl+lbl>,
#   relupdate_w6 <dbl+lbl>, relupdate_w7 <dbl+lbl>, relupdate_w8 <dbl+lbl>,
#   relupdate_w9 <dbl+lbl>, relupdate_w10 <dbl+lbl>, relupdate_w11 <dbl+lbl>,
#   relupdate_w12 <dbl+lbl>, relupdate_w13 <dbl+lbl>, ...
```

```
# Check if the code works
```

```
output_data %>%
  filter(id == 15) %>%
  select(id, starts_with("relationship_duration"), starts_with("relupdate"), starts_with("cursexn
```

```
# A tibble: 1 × 82
```

	id	relationship_duration_w0	relationship_duration_w1	relationship_duratio...
	<dbl>	<dbl>	<dbl>	<dbl>
1	15	361.	363.	365.

```
# i abbreviated name: ^relationship_duration_w2
# i 78 more variables: relationship_duration_w3 <dbl>,
#   relationship_duration_w4 <dbl>, relationship_duration_w5 <dbl>,
#   relationship_duration_w6 <dbl>, relationship_duration_w7 <dbl>,
#   relationship_duration_w8 <dbl>, relationship_duration_w9 <dbl>,
#   relationship_duration_w10 <dbl>, relationship_duration_w11 <dbl>,
#   relationship_duration_w12 <dbl>, relationship_duration_w13 <dbl>, ...
```

```
output_data %>%
  filter(id == 172) %>%
  select(id, starts_with("relationship_duration"), starts_with("relupdate"), starts_with("cursexn
```

```
# A tibble: 1 × 82
```

	id	relationship_duration_w0	relationship_duration_w1	relationship_duratio...
	<dbl>	<dbl>	<dbl>	<dbl>
1	172	374.	376.	378.

```
# i abbreviated name: ^relationship_duration_w2
# i 78 more variables: relationship_duration_w3 <dbl>,
#   relationship_duration_w4 <dbl>, relationship_duration_w5 <dbl>,
#   relationship_duration_w6 <dbl>, relationship_duration_w7 <dbl>,
#   relationship_duration_w8 <dbl>, relationship_duration_w9 <dbl>, ...
```

```
# relationship_duration_w10 <dbl>, relationship_duration_w11 <dbl>,
# relationship_duration_w12 <dbl>, relationship_duration_w13 <dbl>, ...
```

## Survival Analysis

### A) State what your “failure” variable is and how you expect your independent variables to affect it.

For survival analysis, the **event variable** usually focuses on a single type of event/event of interest. In survival analysis, “failure” is used more broadly to indicate the occurrence of the event being studied, regardless of whether it’s a positive or negative event.

My “failure” variable/event of interest is **contraceptive use** (Yes/No). Specifically, the “failure” occurs when an individual transitions from not using contraception (0) to using contraception (1). For each time period (my 2 weeks intervals, I’ll have a binary indicator:

0 = No contraception use (event hasn’t occurred yet)

1 = Contraception use started (event has occurred)

#### Failure Variable:

- For starting contraception: `contraception_start = 1` (e.g., `contmethuse` transitions from 0 to 1).

#### Independent Variables:

- Predictors could include:
  - Relationship status (`curromrel` or `cursexrel`).
  - Relationship duration (`relationship_duration`).
  - Other covariates you’ve included, such as age or education.

#### My hypothesis:

- Longer relationship duration increases the likelihood of contraception use.
- Being in a current romantic or sexual relationship (`curromrel`, `cursexrel`) increases contraception use.
- Changes in relationship status (`relupdate`) might affect contraception use, for example: Stopping contraception may be influenced by relationship stability or transitions (e.g., relationships ending).

- 
- The `contraception_start_w#` variable is `1` for the week when contraception starts/restarts and `0` otherwise.

- This is now a binary variable suitable for survival analysis.

```
for (week in 1:weekCount) {
  # Define current and previous week column names
  current_col <- paste0("contmethuse_w", week)
  previous_col <- paste0("contmethuse_w", week - 1)
  start_col <- paste0("contraception_start_w", week) # Renamed variable

  # Initialize the new variable
  output_data[[start_col]] <- 0 # Default to 0 (no event)

  for (i in 1:nrow(output_data)) {
    current_val <- output_data[[current_col]][i]
    previous_val <- if (week == 1) NA else output_data[[previous_col]][i]

    # Detect starting/restarting contraception
    if (!is.na(current_val) && current_val == 1 && (is.na(previous_val) || previous_val == 0)) {
      output_data[[start_col]][i] <- 1 # Event occurred
    }
  }
}

# Display the updated output_data with the new variables
print(head(output_data))
```

# A tibble: 6 × 151

	id	relstartyr	relstartmonth	rls_baseline_duration... <sup>1</sup>	rls_baseline_duratio... <sup>2</sup>
	<dbl>	<dbl>	<dbl+lbl>	<dbl>	<dbl>
1	235	2019	2 [2. Februar...	62	269.
2	287	2023	12 [12. Decemb...	4	17.4
3	37	2018	6 [6. June]	70	304.
4	196	2023	3 [3. March]	13	56.5
5	237	2023	6 [6. June]	10	43.4
6	194	2018	3 [3. March]	73	317.

# i abbreviated names: <sup>1</sup>rls\_baseline\_duration\_months,

# <sup>2</sup>rls\_baseline\_duration\_weeks

# i 146 more variables: relupdate\_w1 <dbl+lbl>, relupdate\_w2 <dbl+lbl>,

# relupdate\_w3 <dbl+lbl>, relupdate\_w4 <dbl+lbl>, relupdate\_w5 <dbl+lbl>,

# relupdate\_w6 <dbl+lbl>, relupdate\_w7 <dbl+lbl>, relupdate\_w8 <dbl+lbl>,

# relupdate\_w9 <dbl+lbl>, relupdate\_w10 <dbl+lbl>, relupdate\_w11 <dbl+lbl>,

# relupdate\_w12 <dbl+lbl>, relupdate\_w13 <dbl+lbl>, ...

## Data Preparation for Analysis

My current dataset is in wide format. For survival analysis, apparently it is better to use a long format.

- Each row represents a **person-week** (or person-time period).
- Each column represents:

- The **time period** (week or biweekly interval).
- The **event variable** (e.g., `contraception_change` for starting/restarting).
- Time-varying covariates like `relationship_duration`, `curromrel`, `cursexrel`.

### Converting from wide to long data

```
library(tidyr)

# Convert wide-format data to long-format and handle relationship_duration
long_data <- output_data %>%
  # Pivot all `_w#` columns into long format
  pivot_longer(
    cols = matches("_w\\d+$"), # Match all columns with `_w#` suffix
    names_to = c(".value", "week"), # Extract variable name and week number
    names_pattern = "(.*)_w(\\d+)" # Split column names into "variable" and "week"
  ) %>%
  # Convert week to numeric
  mutate(week = as.numeric(week)) %>%
  # Select only the required columns
  select(id, rls_baseline_duration = rls_baseline_duration_weeks, week, relupdate, cursexrel, curromrel, curmethuse, cmethuse_2wk)
  arrange(id, week) # Order data by id and week
```

Warning: `relupdate\_w1` and `relupdate\_w12` have conflicting value labels.

! Labels for these values will be taken from `relupdate\_w1`.

✗ Values: -9, -8, -6, 0, and 1

Warning: `cursexrel\_w1` and `cursexrel\_w12` have conflicting value labels.

! Labels for these values will be taken from `cursexrel\_w1`.

✗ Values: -9, -8, -6, 0, and 1

Warning: `curromrel\_w1` and `curromrel\_w12` have conflicting value labels.

! Labels for these values will be taken from `curromrel\_w1`.

✗ Values: -9, -8, -6, 0, and 1

Warning: `final\_outcome\_w1` and `final\_outcome\_w12` have conflicting value labels.

! Labels for these values will be taken from `final\_outcome\_w1`.

✗ Values: 1, 2, 3, 5, and 6

Warning: `contmethuse\_w1` and `contmethuse\_w12` have conflicting value labels.

! Labels for these values will be taken from `contmethuse\_w1`.

✗ Values: -9, -8, -6, 0, and 1

Warning: `cmethuse\_2wk\_w1` and `cmethuse\_2wk\_w12` have conflicting value labels.

! Labels for these values will be taken from `cmethuse\_2wk\_w1`.

✗ Values: -9, -8, -6, 0, and 1





```

.. ..- attr(*, "names")= chr [1:5] "REFUSED" "DON'T KNOW" "HUNG UP" "No" ...
..@ label : chr "Last time we spoke you were in a relationship with ${rel_init_old}. Are you
sti"
$ cursexrel          : dbl+lbl [1:4675] NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
N...
..@ labels: Named num [1:5] -9 -8 -6 0 1
.. ..- attr(*, "names")= chr [1:5] "REFUSED" "DON'T KNOW" "HUNG UP" "No" ...
..@ label : chr "Are you currently in any type of relationship that involves physical or
emotiona"
$ curromrel          : dbl+lbl [1:4675] NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
N...
..@ labels: Named num [1:5] -9 -8 -6 0 1
.. ..- attr(*, "names")= chr [1:5] "REFUSED" "DON'T KNOW" "HUNG UP" "No" ...
..@ label : chr "Are you currently in a special romantic relationship with anyone?"
$ contmethuse        : dbl+lbl [1:4675] NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
...
..@ labels: Named num [1:5] -9 -8 -6 0 1
.. ..- attr(*, "names")= chr [1:5] "REFUSED" "DON'T KNOW" "HUNG UP" "No" ...
..@ label : chr "Are you or your partner currently doing something or using any method to
delay o"
$ relationship_duration: num [1:4675] 52.1 54.1 56.1 58.1 60.1 ...
$ contraception_start : num [1:4675] NA 0 0 0 0 0 0 0 0 0 ...

```

```

# Check for missing values
colSums(is.na(long_data))

```

id	rls_baseline_duration	week
0	0	0
relupdate	cursexrel	curromrel
1453	4361	4307
contmethuse	relationship_duration	contraception_start
2584	836	275

```
table(long_data$contraception_start)
```

```

0    1
4239 161

```

## B) Determining Risk Window

For survival analysis, the risk window is the period where a participant is “at risk” for the event of interest: starting/restarting contraception.

- **Right Truncation:**

- Right truncation occurs when participants are observed only after a certain point in time, meaning they are included in the study only if they have already experienced certain characteristics (e.g.,

relationship initiation).

- Cases where contraception use doesn't start by the end of the study period will be naturally right-censored.

- **Left-Censoring:**

- Left-censoring occurs when the event of interest has already occurred before the study begins or before data collection starts.
- By including baseline relationship duration in my data, it accounts for relationships that started before the study began.
- Given my data, my risk window is likely defined by individuals who were already using contraception when the study began
- Therefore, participants who had already started using contraception before their inclusion in the study would be left-censored for the "contraception start" analysis.

## Defining the Risk Window

For participants to be included in the model:

- **Starting Contraception:** Filter participants who were at risk (i.e., not using contraception at the start of the study). Ensure only those eligible for failure are included.

```
# Filter for starting contraception
start_data <- long_data %>%
  filter(contmethuse == 0) # Include only participants not using contraception
```

Ensure Binary Events Variables:

```
# Confirm that contraception_start is binary
table(long_data$contraception_start)
```

```
0    1
4239 161
```

```
#How many participants experienced the event (contraception_start == 1)?
table(start_data$contraception_start)
```

```
0
1924
```

## Discrete-Time Logistic Regression

### Model 1: Starting Contraception

```
library(stats)

# Fit the logistic regression model
start_model <- glm(
  contraception_start ~ relationship_duration + curromrel + cursexrel,
  family = binomial,
  data = start_data
)

# Summarize results
summary(start_model)
```

Call:

```
glm(formula = contraception_start ~ relationship_duration + curromrel +
    cursexrel, family = binomial, data = start_data)
```

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-2.657e+01	5.924e+04	0	1
relationship_duration	-8.630e-18	1.317e+03	0	1
curromrel	NA	NA	NA	NA
cursexrel	1.968e-14	2.587e+05	0	1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 0.0000e+00 on 46 degrees of freedom  
 Residual deviance: 2.7267e-10 on 44 degrees of freedom  
 (1877 observations deleted due to missingness)  
 AIC: 6

Number of Fisher Scoring iterations: 25

## Check Model Fit:

Evaluate the model using metrics like AIC or deviance:

```
# Compare null deviance and residual deviance
anova(start_model, test = "Chisq")
```

Analysis of Deviance Table

Model: binomial, link: logit

Response: contraception\_start

Terms added sequentially (first to last)

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
--	----	----------	-----------	------------	----------

NULL			46	0.0000e+00	
relationship_duration	1	0	45	2.7267e-10	1
curromrel	0	0	45	2.7267e-10	
cursexrel	1	0	44	2.7267e-10	1

## C) Explain whether the results were consistent with your expectations, and do that by interpreting the coefficients from the models, model fit, and so on.

The current model indicates that my dataset has too few events or too much multicollinearity to yield meaningful results.

I'm going rerun the model again, with a different approach. The revised code based on the clarified definitions for my event variables. It accounts for:

- **Restarts:** Tracks any transitions from 0 to 1 in `contmethuse` (indicates restarting contraception).
- **Stopping Contraception:** Tracks transitions from 1 to 0 in `contmethuse` (indicates stopping contraception).
- **Switching Methods:** Transition from 1 to 0 in `cmethuse_2wk` (1: Participant did not switch methods; 0: Participant did switch methods.)

```
for (week in 1:weekCount) {
  # Define current and previous week column names
  contmethuse_current <- paste0("contmethuse_w", week)
  contmethuse_previous <- paste0("contmethuse_w", week - 1)
  cmethuse_current <- paste0("cmethuse_2wk_w", week)
  cmethuse_previous <- paste0("cmethuse_2wk_w", week - 1)

  # Define new event columns
  restart_col <- paste0("contraception_restart_w", week)
  stop_col <- paste0("contraception_stop_w", week)
  switch_col <- paste0("contraception_switch_w", week)

  # Initialize new variables
  output_data[[restart_col]] <- 0 # Default to 0 (no restart event)
  output_data[[stop_col]] <- 0 # Default to 0 (no stop event)
  output_data[[switch_col]] <- NA # NA by default (only 1 or 0 for valid rows)

  for (i in 1:nrow(output_data)) {
    # Current and previous values for contraception use
    current_contmethuse <- output_data[[contmethuse_current]][i]
    previous_contmethuse <- if (week == 1) NA else output_data[[contmethuse_previous]][i]

    # Current and previous values for method use status
    current_cmethuse <- output_data[[cmethuse_current]][i]
```

```

# Detect restarting contraception (0 to 1 in contmethuse)
if (!is.na(current_contmethuse) && current_contmethuse == 1 && (is.na(previous_contmethuse) |
  output_data[[restart_col]][i] <- 1 # Restart event occurred
})

# Detect stopping contraception (1 to 0 in contmethuse)
if (!is.na(current_contmethuse) && current_contmethuse == 0 && !is.na(previous_contmethuse) &&
  output_data[[stop_col]][i] <- 1 # Stop event occurred
})

# Detect switching methods (1 to 0 in cmethoduse_2wk)
if (!is.na(current_cmethoduse)) {
  if (current_cmethoduse == 1) {
    output_data[[switch_col]][i] <- 0 # Did not switch methods
  } else if (current_cmethoduse == 0) {
    output_data[[switch_col]][i] <- 1 # Switched methods
  }
}
}
}

# Display the updated output_data with the new variables
print(head(output_data))

```

# A tibble: 6 × 199

	id	relstartyr	relstartmonth	rls_baseline_duration... <sup>1</sup>	rls_baseline_duration... <sup>2</sup>
	<dbl>	<dbl>	<dbl+lbl>	<dbl>	<dbl>
1	235	2019	2 [2. Februar...	62	269.
2	287	2023	12 [12. Decemb...	4	17.4
3	37	2018	6 [6. June]	70	304.
4	196	2023	3 [3. March]	13	56.5
5	237	2023	6 [6. June]	10	43.4
6	194	2018	3 [3. March]	73	317.

# i abbreviated names: <sup>1</sup>rls\_baseline\_duration\_months,

# <sup>2</sup>rls\_baseline\_duration\_weeks

# i 194 more variables: relupdate\_w1 <dbl+lbl>, relupdate\_w2 <dbl+lbl>,

# relupdate\_w3 <dbl+lbl>, relupdate\_w4 <dbl+lbl>, relupdate\_w5 <dbl+lbl>,

# relupdate\_w6 <dbl+lbl>, relupdate\_w7 <dbl+lbl>, relupdate\_w8 <dbl+lbl>,

# relupdate\_w9 <dbl+lbl>, relupdate\_w10 <dbl+lbl>, relupdate\_w11 <dbl+lbl>,

# relupdate\_w12 <dbl+lbl>, relupdate\_w13 <dbl+lbl>, ...

## Switch Wide to Long Data

```

# Convert wide-format data to long-format and handle specified variables
long_data <- output_data %>%
  # Pivot all `_w#` columns into long format
  pivot_longer(
    cols = matches("_w\\d+$"), # Match all columns with `_w#` suffix
    names_to = c(".value", "week"), # Extract variable name and week number
    names_pattern = "(.*)_w(\\d+)" # Split column names into "variable" and "week"
  )

```

```

) %>%
# Convert week to numeric
mutate(week = as.numeric(week)) %>%
# Select only the requested columns
select(
  id,
  week,
  cmethoduse_2wk,
  relupdate,
  cursexrel,
  curromrel,
  contmethuse,
  relationship_duration,
  contraception_restart,
  contraception_stop,
  contraception_switch
) %>%
arrange(id, week) # Order data by id and week

```

Warning: `relupdate\_w1` and `relupdate\_w12` have conflicting value labels.

i Labels for these values will be taken from `relupdate\_w1`.

✗ Values: -9, -8, -6, 0, and 1

Warning: `cursexrel\_w1` and `cursexrel\_w12` have conflicting value labels.

i Labels for these values will be taken from `cursexrel\_w1`.

✗ Values: -9, -8, -6, 0, and 1

Warning: `curromrel\_w1` and `curromrel\_w12` have conflicting value labels.

i Labels for these values will be taken from `curromrel\_w1`.

✗ Values: -9, -8, -6, 0, and 1

Warning: `final\_outcome\_w1` and `final\_outcome\_w12` have conflicting value labels.

i Labels for these values will be taken from `final\_outcome\_w1`.

✗ Values: 1, 2, 3, 5, and 6

Warning: `contmethuse\_w1` and `contmethuse\_w12` have conflicting value labels.

i Labels for these values will be taken from `contmethuse\_w1`.

✗ Values: -9, -8, -6, 0, and 1

Warning: `cmethoduse\_2wk\_w1` and `cmethoduse\_2wk\_w12` have conflicting value labels.

i Labels for these values will be taken from `cmethoduse\_2wk\_w1`.

✗ Values: -9, -8, -6, 0, and 1

```

# Display the first few rows of the filtered long-format dataset
print(head(long_data))

```

# A tibble: 6 × 11

	id	week	cmethoduse_2wk	relupdate	cursexrel	curromrel	contmethuse
	<dbl>	<dbl>	<dbl+lbl>	<dbl+lbl>	<dbl+lbl>	<dbl+lbl>	<dbl+lbl>
1	1	0	NA	NA	NA	NA	NA

```

2      1      1 NA          1 [Yes] NA      NA      0 [No]
3      1      2 NA          1 [Yes] NA      NA      0 [No]
4      1      3 NA          1 [Yes] NA      NA      0 [No]
5      1      4 NA          1 [Yes] NA      NA      0 [No]
6      1      5 NA          1 [Yes] NA      NA      0 [No]
# i 4 more variables: relationship_duration <dbl>, contraception_restart <dbl>,
#   contraception_stop <dbl>, contraception_switch <dbl>

```

```
table(long_data$contraception_restart)
```

```

  0    1
4239 161

```

```
table(long_data$contraception_stop)
```

```

  0    1
4325  75

```

```
table(long_data$contraception_switch)
```

```

  0    1
1297  58

```

**Inspecting the distribution:** Check how these events are distributed across participants and time periods to ensure meaningful patterns:

```

long_data %>%
  group_by(id) %>%
  summarize(
    restarts = sum(contraception_restart, na.rm = TRUE),
    stops = sum(contraception_stop, na.rm = TRUE),
    switches = sum(contraception_switch, na.rm = TRUE)
  ) %>%
  arrange(desc(restarts), desc(stops), desc(switches)) # Arrange by multiple columns

```

```
# A tibble: 275 × 4
```

```

      id restarts stops switches
  <dbl>   <dbl> <dbl>   <dbl>
1     59         3     1         2
2    270         3     0         3
3     64         2     1         1
4    175         2     1         1
5     56         2     1         0
6     95         2     1         0
7    148         2     1         0

```



```

8    255      2    1      0
9     69      2    0      2
10   100      2    0      2

```

```
# i 265 more rows
```

**Checking Event Timing:** Verify whether these events occur early or late in the study to evaluate censoring effects

```

long_data %>%
  filter(contraception_restart == 1 | contraception_stop == 1 | contraception_switch == 1) %>%
  group_by(week) %>%
  summarize(
    restarts = sum(contraception_restart, na.rm = TRUE),
    stops = sum(contraception_stop, na.rm = TRUE),
    switches = sum(contraception_switch, na.rm = TRUE)
  )

```

```
# A tibble: 16 × 4
```

	week	restarts	stops	switches
	<dbl>	<dbl>	<dbl>	<dbl>
1	1	105	0	12
2	2	5	74	5
3	3	7	0	3
4	4	1	0	1
5	5	4	0	3
6	6	6	1	1
7	7	5	0	4
8	8	6	0	1
9	9	2	0	2
10	10	1	0	5
11	11	4	0	3
12	12	6	0	5
13	13	3	0	3
14	14	2	0	2
15	15	2	0	4
16	16	2	0	4

## Discrete-Time Logistic Regression Trial 2

### Model 1: Contraception Restart

```

# Filter data for restart analysis
restart_data <- long_data %>%
  filter(!is.na(contraception_restart)) # Exclude rows with missing restart info

# Fit the logistic regression model for restart
restart_model <- glm(
  contraception_restart ~ relationship_duration + curromrel + cursexrel,

```

```
family = binomial,
data = restart_data
)

# Summarize the model results
summary(restart_model)
```

Call:

```
glm(formula = contraception_restart ~ relationship_duration +
    curromrel + cursexrel, family = binomial, data = restart_data)
```

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-3.301e+00	6.805e-01	-4.851	1.23e-06 ***
relationship_duration	1.279e-02	6.434e-03	1.988	0.0468 *
curromrel	NA	NA	NA	NA
cursexrel	-1.427e+01	2.284e+03	-0.006	0.9950

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 30.546 on 68 degrees of freedom  
 Residual deviance: 26.504 on 66 degrees of freedom  
 (4331 observations deleted due to missingness)  
 AIC: 32.504

Number of Fisher Scoring iterations: 16

```
anova(restart_model, test = "Chisq")
```

Analysis of Deviance Table

Model: binomial, link: logit

Response: contraception\_restart

Terms added sequentially (first to last)

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL			68	30.546	
relationship_duration	1	3.8285	67	26.718	0.05039 .
curromrel	0	0.0000	67	26.718	
cursexrel	1	0.2139	66	26.504	0.64376

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Model 2: Contraception Stop

```
# Filter data for stop analysis
stop_data <- long_data %>%
  filter(!is.na(contraception_stop)) # Exclude rows with missing stop info

# Fit the logistic regression model for stop
stop_model <- glm(
  contraception_stop ~ relationship_duration + curromrel + cursexrel,
  family = binomial,
  data = stop_data
)

# Summarize the model results
summary(stop_model)
```

Call:

```
glm(formula = contraception_stop ~ relationship_duration + curromrel +
     cursexrel, family = binomial, data = stop_data)
```

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-2.657e+01	4.821e+04	-0.001	1
relationship_duration	-4.662e-17	8.462e+02	0.000	1
curromrel	NA	NA	NA	NA
cursexrel	-5.673e-15	2.111e+05	0.000	1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 0.0000e+00 on 68 degrees of freedom  
 Residual deviance: 4.0031e-10 on 66 degrees of freedom  
 (4331 observations deleted due to missingness)  
 AIC: 6

Number of Fisher Scoring iterations: 25

## Model 3: Switching Contraception Methods

```
# Filter data for switch analysis
switch_data <- long_data %>%
  filter(!is.na(contraception_switch)) # Exclude rows with missing switch info

# Fit the logistic regression model for switching methods
switch_model <- glm(
  contraception_switch ~ relationship_duration + curromrel + cursexrel,
  family = binomial,
  data = switch_data
```

```
)

# Summarize the model results
summary(switch_model)
```

Call:

```
glm(formula = contraception_switch ~ relationship_duration +
     curromrel + cursexrel, family = binomial, data = switch_data)
```

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-2.3026	1.0488	-2.195	0.0281 *
relationship_duration	-0.1833	74.4382	-0.002	0.9980
curromrel	NA	NA	NA	NA
cursexrel	-16.8969	10755.0435	-0.002	0.9987

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 7.2049 on 13 degrees of freedom  
 Residual deviance: 6.7020 on 11 degrees of freedom  
 (1341 observations deleted due to missingness)  
 AIC: 12.702

Number of Fisher Scoring iterations: 18

## Final Results

### 1. Restarting Contraception

**Coefficients:**

#### 1. Intercept (-3.301):

- The baseline log-odds of restarting contraception when all predictors are at their reference level.
- A significant negative coefficient ( $p < 0.001$ ) suggests a low baseline probability of restarting contraception.

#### 2. Relationship Duration (0.01279):

- A positive coefficient ( $p = 0.0468$ ) indicates that longer relationship durations slightly increase the odds of restarting contraception.
- Odds Ratio =  $\exp(0.01279) \approx 1.013$ : Each additional unit (e.g., two weeks) of relationship duration increases the odds of restarting contraception by 1.3%.

### 3. **curromrel**:

- Excluded from the model due to multicollinearity (likely perfectly correlated with another variable or constant within the dataset).

### 4. **cursexrel** ( **-14.27** ):

- Very large negative coefficient with a high standard error and an insignificant p-value ( $p = 0.9950$ ), suggesting no meaningful relationship between being in a sexual relationship and restarting contraception.

### Model Fit:

- **Null deviance** (30.546) vs. **Residual deviance** (26.504): Modest improvement in fit, indicating predictors explain some variation.
- **AIC (32.504)**: Indicates the overall model fit but is not interpretable in isolation.

### Conclusion:

- Relationship duration is the only significant predictor of restarting contraception.
- Being in a sexual relationship (cursexrel) does not significantly influence restarting contraception, and the model is limited by excluded variables and potential collinearity.

## 2. Stopping Contraception

### Coefficients:

#### 1. **Intercept** ( **-26.57** ):

- Extremely large negative coefficient with an insignificant p-value ( $p = 1$ ), suggesting no meaningful baseline odds.

#### 2. **Relationship Duration** ( **-4.66e-17** ):

- Essentially zero with an insignificant p-value ( $p = 1$ ), indicating no effect of relationship duration on stopping contraception.

#### 3. **curromrel**:

- Excluded due to multicollinearity.

#### 4. **cursexrel** ( **-5.67e-15** ):

- Essentially zero with an insignificant p-value ( $p = 1$ ), suggesting no relationship between being in a sexual relationship and stopping contraception.

### Model Fit:

- **Null deviance** (0) vs. **Residual deviance** ( $\sim 0$ ): The model does not explain any variation in the data.

- **AIC (6):** Very low, but this is likely due to the lack of meaningful variation in the outcome.

**Conclusion:**

- None of the predictors significantly explain stopping contraception.
- The lack of variability in the dependent variable or multicollinearity issues may be impacting the model.

### 3. Switching Contraceptive Methods

**Coefficients:****1. Intercept ( -2.3026 ):**

- Significant negative coefficient ( $p = 0.0281$ ) indicates a low baseline log-odds of switching contraception methods.

**2. Relationship Duration ( -0.1833 ):**

- A negative coefficient with an insignificant p-value ( $p = 0.9980$ ), suggesting no meaningful effect of relationship duration on switching methods.

**3. curromrel:**

- Excluded due to multicollinearity.

**4. cursexrel ( -16.90 ):**

- Extremely large negative coefficient with a high standard error and insignificant p-value ( $p = 0.9987$ ), indicating no meaningful relationship between being in a sexual relationship and switching contraception methods.

**Model Fit:**

- **Null deviance** (7.205) vs. **Residual deviance** (6.702): Minimal improvement in fit.
- **AIC (12.702):** Indicates the overall model fit but is not interpretable in isolation.

**Conclusion:**

- The intercept suggests a low likelihood of switching methods, but none of the predictors significantly explain the outcome.
- The small number of events (14 switches) and potential multicollinearity issues are likely limiting the model.