Handling Missing Data in Health Science Research

Day 2 - Part I

2022-06-23

Contents

Missing responses in longitudinal studies

- Except in highly controlled settings, missing data in longitudinal studies are inevitable
- What are the implications for missing data?
  - Create complications for methods that require **balanced** data
  - Reduce the **precision** with which changes in the mean response over time can be estimated
  - Can introduce **bias** and lead to misleading inferences about changes in the mean response
- Statistical methods to account for missing data in correlated (longitudinal) data is still a rapidly developing field
- Usually, the missing data mechanism is not under the control of the investigators
- We make **assumptions** about the missing data mechanism
- Validity of the analysis depends on whether these assumptions hold
- We need to be **explicit** about the assumptions made regarding the reasons for missing data

Example: Longitudinal outcomes in skin cancer study

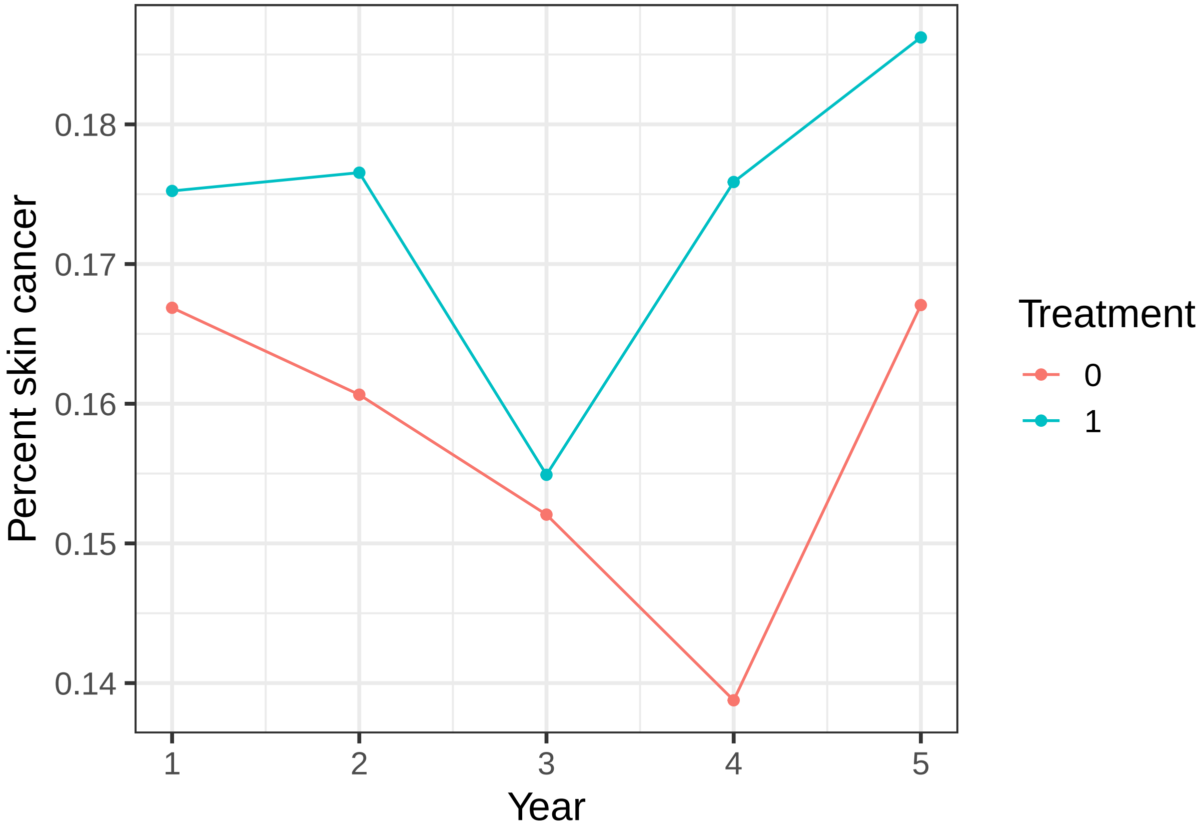
- In the previous session, we focused on the baseline data and skin cancer count after the first year
- In this session, we will analyze the full data including follow-up skin cancer status from year 1 through year 5
- We will focus on the missing longitudinal **outcomes** rather than missing covariates
- However, the implementation of Poisson data is not well developed in the joint modeling framework
- Therefore, we will **dichotomize** the outcome to implement logistic regression
  - If $Y_{\text{orig}} = 0$ then $Y_{\text{new}} = 0$ and if $Y_{\text{orig}} > 0$ then $Y_{\text{new}} = 1$

```r
skin_data <- read.table("skin_data.txt", header = TRUE)
head(skin_data)
```

```
##       ID center age skin gender exposure Y treatment year
## 1 100034      1  51    1      1        4 0         0    1
## 2 100034      1  51    1      1        4 1         0    2
## 3 100034      1  51    1      1        4 1         0    3
## 4 100034      1  51    1      1        4 1         0    4
## 5 100034      1  51    1      1        4 0         0    5
## 6 100045      1  68    1      0        2 0         0    1
```

```r
skin_data <- skin_data %>%
  mutate(Y_bin = ifelse(Y == 0, 0, 1)) %>%
  dplyr::select(-Y)

## plot the data
skin_data %>%
  group_by(treatment, year) %>%
  summarise(pskin = mean(Y_bin)) %>%
  ggplot(aes(y = pskin, x = year, color = as.factor(treatment))) +
  geom_point() +
  geom_line() +
  labs(y = "Percent skin cancer", x = "Year", color = "Treatment")
```



- We are interested in modeling the population-averaged inference of the change in risk of new skin cancer by treatment group

$$\text{logit}(\mu_{ij}) = \beta_0 + \beta_1\text{Year}_{ij} + \beta_2\text{Year}^2_{ij} + \beta_4\text{Treatment}_i + \beta_5(\text{Year}_{ij} \times \text{Treatment}_i) + \beta_6(\text{Year}^2_{ij} \times \text{Treatment}_i)$$

Missing data notation revisted

Suppose we have n repeated measurements of the same individual. Then, the ith subject's set of responses can be represented as a $n \times 1$ vector denoted by

$$Y_i = (Y_{i1}, Y_{i2}, ..., Y_{in})^T.$$

and the response vector Y_i is coupled with a $n \times 1$ vector of **response indicators**

$$R_i = (R_{i1}, R_{i2}, ..., R_{in})^T,$$

where $R_{ij} = 1$ if Y_{ij} is observed and $R_{ij} = 0$ is Y_{ij} is missing.

Given R_i, we can **partition** $Y_i = (Y_{i1}, Y_{i2}, ..., Y_{in})^T$ into two components Y_i^O and Y_i^M where

- Y_i^O denotes the vector of **observed** responses for subject i
- Y_i^M denotes the vector of **missing** responses for subject i

For example,

```
##   id trt  y0  y1  y2  y3
## 1  1   0 2.1 2.6 3.0 3.3
## 2  2   0 2.7  NA  NA 2.9
## 3  3   0 1.9 2.5 2.7  NA
## 4  4   1 3.4 3.5 3.7 3.9
## 5  5   1 1.8 2.7  NA  NA
## 6  6   1 4.0 4.2 4.6 5.0
```

then,

$$Y_1 = \begin{pmatrix} 2.1 \\ 2.6 \\ 3.0 \\ 3.3 \end{pmatrix} \quad R_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad Y_1^O = \begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \end{pmatrix} \quad Y_1^M = ()$$

and

$$Y_2 = \begin{pmatrix} 2.7 \\ \\ \\ 2.9 \end{pmatrix} \quad R_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad Y_2^O = \begin{pmatrix} y_{21} \\ y_{24} \end{pmatrix} \quad Y_2^M = \begin{pmatrix} y_{22} \\ y_{23} \end{pmatrix}$$

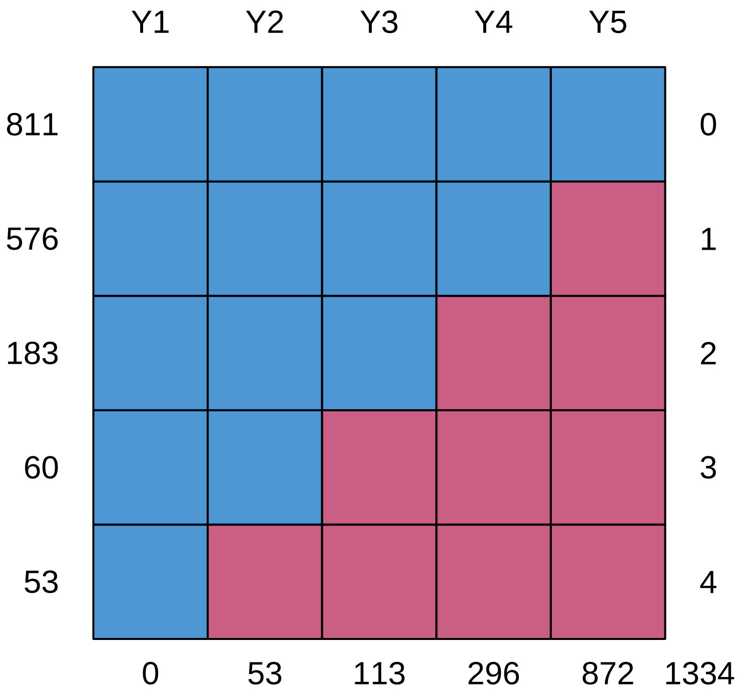
Missing data pattern

Largely, two types of missing data pattern exist in longitudinal studies:

Monotone missing data pattern

- Arises from **dropout**
- The term dropout refers to the special case where if Y_{ik} is missing, then $Y_{ik+1}, ..., Y_{in}$ are also missing
- Key question: Do individuals that dropout and those that remain in the study differ in any further relevant way?

```
mice::md.pattern()
```

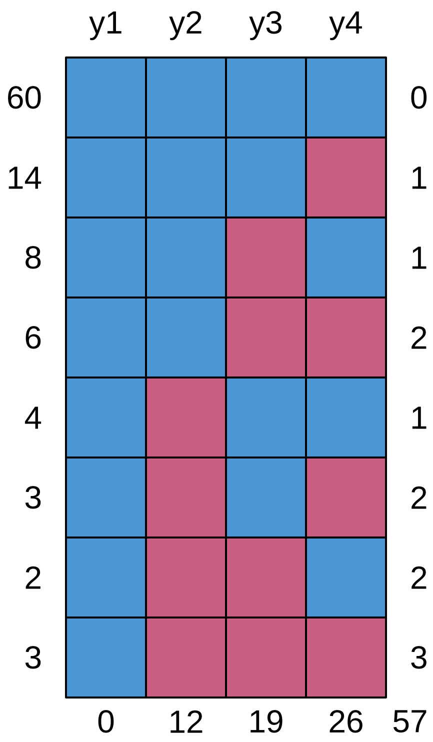


```
##      Y1 Y2  Y3  Y4  Y5
## 811   1  1   1   1   1    0
## 576   1  1   1   1   0    1
## 183   1  1   1   0   0    2
## 60    1  1   0   0   0    3
## 53    1  0   0   0   0    4
##       0 53 113 296 872 1334
```

- Monotone missing data pattern
- 811 people have complete responses
- 53 people dropped out after the first visit
- 60 people dropped out after the second visit
- 183 people dropped out after the third visit
- 576 people dropped out after the fourth visit

Intermittent (non-monotone) missing data pattern

- Missing data pattern that is not monotone



```
##    y1 y2 y3 y4
## 60  1  1  1  1  0
## 14  1  1  1  0  1
## 8   1  1  0  1  1
## 6   1  1  0  0  2
## 4   1  0  1  1  1
## 3   1  0  1  0  2
## 2   1  0  0  1  2
## 3   1  0  0  0  3
##     0 12 19 26 57
```

- Non-monotone missing data pattern
- 60 subjects have complete responses
- 3 subjects have only the baseline response

Approaches for missing data in longitudinal studies

Ignorable missingness (MCAR and MAR)

- **Complete-case analysis**
  - Restrict analysis to individuals with no missing data
  - Valid if data are MCAR
- **Available data analysis**

  - Use all available data (include individuals with some missing data)
  - Valid if data are MCAR

- **Last value carried forward**
  - Applies to monotone missing data
  - Use last observed observation for subsequent missing observations
  - Still popular despite many disadvantages
  - Produces biased estimates and small standard errors
- **Maximum likelihood methods**
  - Maximum likelihood methods (linear mixed effects models, generalized linear mixed effects models) are valid if data are MCAR/MAR
  - Requires the correct specification of the mean and variance model
- **Inverse probability weighting**
  - Weighting method that attempts to "even out" the contribution by individuals
  - Appropriate for monotone missing data pattern
  - Works well with marginal models (generalized estimating equations or GEE)
  - Valid if data are MCAR/MAR
- **Multiple imputation**
  - Appropriate for monotone and intermittent missing patterns
- Other methods include
  - Combination of IPW and MI
  - EM algorithm
  - Bayesian methods
  - Many R packages are available

Inverse probability weighting

Overview

- Basic idea is to estimate the probability of individuals remaining (or dropping out) in the study and weigh each observation according to that probability
  - Individuals with low probability of remaining in the study (high probability of dropping out) are given larger weights
  - Individuals with high probability of remaining in the study (low probability of dropping out) are given smaller weights
- IPW methods are more straightforward to implement with **monotone** missing data pattern
- IPW methods are more appealing when a full likelihood-based analysis is not possible
  - i.e. marginal analysis with discrete responses
  - IPW is often incorporated into **GEE**
- Requires the correct specification of the dropout model ($\Pr(R_{ij} = 1 | R_{i1} = \cdots R_{i,j-1}, X_i, Y_{i1}, ..., Y_{i,j-1})$)

IPW-GEE

The IPW-GEE estimator is obtained as the solution to the following **weighted** estimating equations:

$$\sum_{i=1}^{N} D_i^T V_i^{-1} W_i (Y_i - \mu) = 0,$$

where

- D_i is the $n \times p$ derivative matrix
- V_i is a $n \times n$ working covariance matrix for Y_i
- W_i is a $n \times n$ **diagonal** matrix of the occasion-specific weights, w_{ij}, for $j = 1, ..., n$,

$$W_i = \begin{pmatrix} R_{i1} \times w_{i1} & 0 & ... & 0 \\ 0 & R_{i2} \times w_{i2} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & R_{in} \times w_{in} \end{pmatrix}$$

The weight, w_{ij} is the inverse of the **unconditional** probability of being observed at the jth occasion.

To calculate these weights, let π_{ij} denote the **conditional** probability of the ith individual being observed (or not dropping out) at the jth occasion, given that this individual was observed at the prior occasions.

For the first occasion we usually assume $R_{i1} = 1$ for all individuals, and then $\pi_{i1} = 1$.

The MAR assumption implies that

$$\pi_{ij} = \Pr(R_{ij} = 1 | R_{i1} = \cdots = R_{i,j-1} = 1, Y_{i1} = \cdots = Y_{i,j-1}, X_i).$$

The **unconditional** probability of being observed at the jth occasion can be expressed as the **cumulative product** of the **conditional** probabilities,

$$\pi_{i1} \times \pi_{i2} \times \cdots \times \pi_{ij}.$$

The required weight is then given by the **inverse** of the cumulative product of conditional probabilities,

$$w_{ij} = (\pi_{i1} \times \pi_{i2} \times \cdots \times \pi_{ij})^{-1}.$$

Estimation of weights

We can estimate π_{ij} by constructing a logistic regression model for π_{ij}:

$$\begin{aligned} \text{logit}(\pi_{ij}) &= \text{logit}\left\{\Pr(R_{ij} = 1 | R_{i1} = \cdots = R_{i,j-1} = 1, Z_{ij})\right\} \\ &= Z_{ij}^T \theta \end{aligned}$$

where Z_{ij} is a $q \times 1$ design vector that incorporates:

- certain components of X_{ij}
- past responses $(Y_{i1}, ..., Y_{i,j-1})$
- possibly additional covariates that may be predictive of dropout but are not of subject-matter interest in the marginal model for the mean response

Assumptions

- The missing data mechanism depends only on **variables fully observed** in the sample
- The probability of being observed (π_{ij}) is **positive** (not close to zero)
    - If π_{ij} is very small, w_{ij} will be extremely large
    - Extremely large weights on small subset of observations may yield regression parameter estimates that are unstable and have poor precision
- Safest to assume working **independence** correlation
    - Need robust standard errors using the sandwich variance estimator

Detailed approach with data from skin cancer study

We are interested in modeling the population-averaged inference of the change in risk of new skin cancer by treatment group

$$\text{logit}(\mu_{ij}) = \beta_0 + \beta_1 \text{Year}_{ij} + \beta_2 \text{Year}_{ij}^2 + \beta_4 \text{Treatment}_i + \beta_5(\text{Year}_{ij} \times \text{Treatment}_i) + \beta_6(\text{Year}_{ij}^2 \times \text{Treatment}_i)$$

- Because we are interested in modeling the marginal probability of a **discrete** (or more specifically, binary) outcome, we cannot employ ML methods
- We need to fit a marginal model using GEE
  - Estimates will be biased if data are MAR
- Incorporate IPW

Model for dropout process

- First, we will fit a model for the dropout process
- The outcome is $\text{logit}(\pi_{ij}) = \text{logit}\left\{\Pr(R_{ij} = 1)\right\}$
  - Although it is called the "dropout model", we are modeling the probability of **not** dropping out
  - We don't include baseline data (everybody is observed)
- The predictors include fully observed covariates and previously observed responses

$$\begin{aligned}\text{logit}(\pi_{ij}) = \theta_1 + \theta_2 I(t=3) + \theta_3 I(t=4) + \theta_4 I(t=5) + \theta_5 \text{age}_i + \theta_6 \text{skin}_i + \theta_7 \text{treatment}_i \\ + \theta_8 \text{gender}_i + \theta_9 \text{exposure}_i + \theta_{10} Y_{i,j-1} + \theta_{11}(\text{treatment}_i \times Y_{i,j-1}), \quad j = 2,3,4,5\end{aligned}$$

where $\pi_{ij} = \Pr(R_{ij} = 1 | R_{i1} = \cdots = R_{i,j-1} = 1, Y_{i,j-1}, X_i)$.

```
# change to wide format first to fill in missing years with NA
skin_wide <- skin_data %>%
  pivot_wider(names_from = year,
              names_prefix = "Y",
              values_from = Y_bin)
head(skin_wide)
```

```
## # A tibble: 6 x 12
##        ID center   age  skin gender exposure treatment    Y1    Y2    Y3    Y4
##     <int>  <int> <int> <int>  <int>    <int>     <int> <dbl> <dbl> <dbl> <dbl>
## 1 100034       1    51     1      1        4         0     0     1     1     1
## 2 100045       1    68     1      0        2         0     0     0     0     0
## 3 100056       1    58     1      0        7         0     1     1     0     1
## 4 100067       1    53     1      1        3         0     0     0     0     0
## 5 100102       1    55     0      0        2         0     0     0     0     0
## 6 100113       1    59     1      1       10         0     0     0     1     0
## # ... with 1 more variable: Y5 <dbl>
```

```
skin_long <- skin_wide %>%
  pivot_longer(cols = starts_with("Y"),
               values_to = "Y",
               names_to = "Year",
               names_prefix = "Y") %>%
  mutate(Year = as.numeric(as.factor(Year)),
         Year2 = Year^2,
         trtYear = treatment * Year,
         trtYear2 = treatment * Year2)
head(skin_long)
```

```
## # A tibble: 6 x 12
##       ID center   age  skin gender exposure treatment  Year     Y Year2 trtYear
##    <int>  <int> <int> <int>  <int>    <int>     <int> <dbl> <dbl> <dbl>   <dbl>
## 1 100034      1    51     1      1        4         0     1     0     1       0
## 2 100034      1    51     1      1        4         0     2     1     4       0
## 3 100034      1    51     1      1        4         0     3     1     9       0
## 4 100034      1    51     1      1        4         0     4     1    16       0
## 5 100034      1    51     1      1        4         0     5     0    25       0
## 6 100045      1    68     1      0        2         0     1     0     1       0
## # ... with 1 more variable: trtYear2 <dbl>
```

```
ipwdat <- skin_long %>%
  group_by(ID) %>%
  mutate(prevy = dplyr::lag(Y)) %>%
  ungroup() %>%
  mutate(r = ifelse(is.na(Y), 0, 1),
         t2 = ifelse(Year == 2, 1, 0),
         t3 = ifelse(Year == 3, 1, 0),
         t4 = ifelse(Year == 4, 1, 0),
         t5 = ifelse(Year == 5, 1, 0),
         ## ‘*’ not meaningful for factors
         trt.prevy =  as.numeric(as.character(treatment)) * as.numeric(as.character(prevy))) %>%
  filter(!is.na(Y)|!is.na(prevy))
head(ipwdat)
```

```
## # A tibble: 6 x 19
##       ID center   age  skin gender exposure treatment  Year     Y Year2 trtYear
##    <int>  <int> <int> <int>  <int>    <int>     <int> <dbl> <dbl> <dbl>   <dbl>
## 1 100034      1    51     1      1        4         0     1     0     1       0
## 2 100034      1    51     1      1        4         0     2     1     4       0
## 3 100034      1    51     1      1        4         0     3     1     9       0
## 4 100034      1    51     1      1        4         0     4     1    16       0
## 5 100034      1    51     1      1        4         0     5     0    25       0
## 6 100045      1    68     1      0        2         0     1     0     1       0
## # ... with 8 more variables: trtYear2 <dbl>, prevy <dbl>, r <dbl>, t2 <dbl>,
## #   t3 <dbl>, t4 <dbl>, t5 <dbl>, trt.prevy <dbl>
```

```
# fit drop-out model
rmod <- glm(r ~ t3 + t4 + t5 + age + skin + treatment + gender + exposure + prevy + trt.prevy,
            data = ipwdat, family = binomial("logit"))
round(summary(rmod)$coef,2)
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.77       0.31   12.13     0.00
## t3             -0.16       0.19   -0.85     0.39
## t4             -1.41       0.16   -8.82     0.00
## t5              -3.11       0.15  -20.68     0.00
## age              0.00       0.00    0.06     0.95
## skin            -0.15       0.08   -1.84     0.07
## treatment       -0.14       0.09   -1.59     0.11
## gender          -0.22       0.09   -2.41     0.02
## exposure         0.00       0.01   -0.13     0.90
## prevy           -0.16       0.16   -0.97     0.33
## trt.prevy       -0.20       0.21   -0.95     0.34
```

Compute IPW

- First, compute the predicted logit($\hat{\pi}_{ij}$) from the dropout model
- Then, compute the predicted $\hat{\pi}_{ij}$
- Because the first response was fully observed, with $R_{ij} = 1$ for all individuals, $\hat{\pi}_{i1} = 1$ by definition

$$\hat{\pi}_{ij} = \frac{\exp(Z_{ij}^T \hat{\theta})}{1 + \exp(Z_{ij}^T \hat{\theta})}$$

```r
dropcoef <- summary(rmod)$coef[,1]
## create the dataset for predicting the weight
xmat <- model.matrix(~ t3 + t4 + t5 + age + skin + treatment + gender + exposure + prevy + trt.prevy,
                     model.frame(~., data = ipwdat, na.action = na.pass))
dim(xmat)
```

```
## [1] 7953   11
```

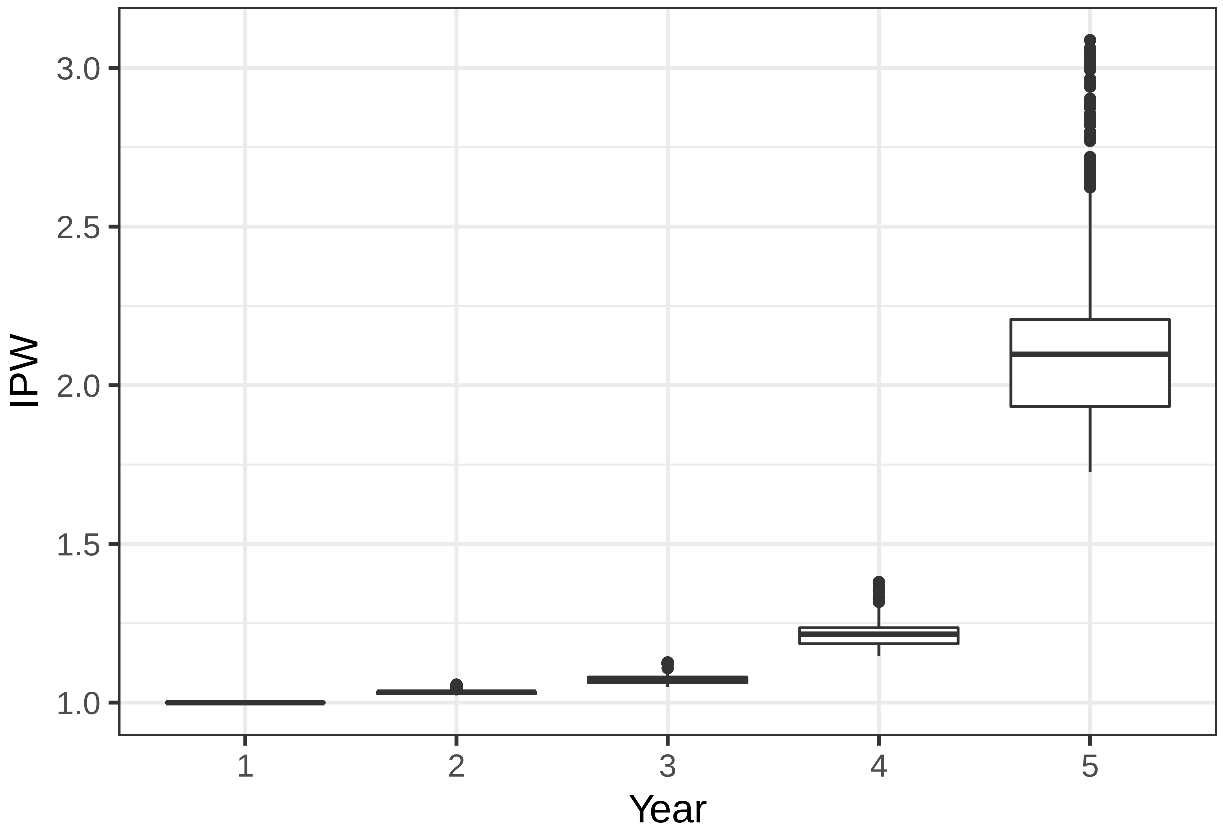
```r
ipwdat <- ipwdat %>%
          mutate(logitp = as.numeric(xmat %*% dropcoef),
                 phat = ifelse(Year == 1, 1, exp(logitp)/(1 + exp(logitp)))) %>%
          group_by(ID) %>%
          mutate(cumprob = cumprod(phat),
          ipw = 1/cumprob) %>%
          ungroup()

ipwdat %>%
  filter(ID %in% c(100034, 100067, 103059, 416964)) %>%
  dplyr::select(ID, treatment, Year, Y, prevy, r, logitp, phat, cumprob, ipw) %>%
  print()
```

```
## # A tibble: 20 x 10
##        ID treatment  Year     Y prevy     r logitp  phat cumprob   ipw
##     <int>     <int> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl>
##  1 100034         0     1     0    NA     1 NA     1       1        1
##  2 100034         0     2     1     0     1   3.41  0.968   0.968    1.03
##  3 100034         0     3     1     1     1   3.09  0.956   0.926    1.08
##  4 100034         0     4     1     1     1   1.83  0.862   0.798    1.25
##  5 100034         0     5     0     1     1 0.141  0.535   0.427    2.34
##  6 100067         0     1     0    NA     1 NA     1       1        1
##  7 100067         0     2     0     0     1   3.41  0.968   0.968    1.03
##  8 100067         0     3     0     0     1   3.25  0.963   0.932    1.07
##  9 100067         0     4     0     0     1   2.00  0.880   0.820    1.22
## 10 100067         0     5     0     0     1  0.303 0.575   0.472  2.12
## 11 103059         0     1     0    NA     1 NA     1       1      1
## 12 103059         0     2     0     0     1  3.42  0.968   0.968  1.03
## 13 103059         0     3     0     0     1  3.25  0.963   0.932  1.07
## 14 103059         0     4     0     0     1  2.00  0.881   0.821  1.22
## 15 103059         0     5     0     0     1  0.309 0.577   0.474  2.11
## 16 416964         0     1     0    NA     1 NA     1       1      1
## 17 416964         0     2     0     0     1  3.57  0.973   0.973  1.03
## 18 416964         0     3     0     0     1  3.41  0.968   0.941  1.06
## 19 416964         0     4     0     0     1  2.16  0.896   0.844  1.19
## 20 416964         0     5    NA     0     0  0.462 0.614   0.518  1.93
```

- Prior to conducting an IPW-GEE analysis, we should examine the **distribution** of the estimated weights for any presence of discernibly large weights

```
# examine the weights by time point
ipwdat %>%
    ggplot(aes(y = ipw, x = as.factor(Year))) +
    geom_boxplot() +
    labs(y = "IPW", x = "Year")
```



Observations

- $\hat{w}_{i1} = 1$ for all individuals, as should be
- Estimated weights are increasing over time
- Estimated weights range from 1.0 to 3.1 → no concern that a small subset of the observations might have undue influence on the analysis

Model for response with IPW

Finally, we will fit a logistic regression model for the marginal probability of developing skin cancer:

- Use `wights=` option in `geeglm` from R package `geepack`
- To ensure that the weights are appropriately incorporated, we need to make the **“working independence”** assumption for the within-subject association among the responses
- Because a “working independence” assumption is made, standard errors are based on the **sandwich variance** estimator
  - Default for `geeglm`

```
# ipw-gee
library(geepack)
ipwgee <- geeglm(Y ~ treatment + Year + Year2 + trtYear + trtYear2,
                 family=binomial("logit"),
                 id = ID, scale.fix = TRUE,
                 corstr = "independence",
                 weights = ipw,
                 data = ipwdat)
summary(ipwgee)
```

```
##
## Call:
## geeglm(formula = Y ~ treatment + Year + Year2 + trtYear + trtYear2,
##     family = binomial("logit"), data = ipwdat, weights = ipw,
##     id = ID, corstr = "independence", scale.fix = TRUE)
##
##  Coefficients:
##             Estimate   Std.err   Wald Pr(>|W|)
## (Intercept) -1.374470  0.186246 54.462 1.58e-13 ***
## treatment    0.003619  0.260712  0.000   0.9889
## Year        -0.244271  0.146087  2.796   0.0945 .
## Year2        0.040117  0.025367  2.501   0.1138
## trtYear      0.051900  0.206521  0.063   0.8016
## trtYear2    -0.002670  0.036047  0.005   0.9410
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation structure = independence
## Scale is fixed.
##
## Number of clusters:   1683  Maximum cluster size: 5
```

- We will compare **four** different analyses
  - Complete-case analysis
  - Available data analysis
  - Last observation carried forward
  - IPW

```
# complete cases (remove all subjects who dropout)
ccdat <- skin_long %>%
```

Table 1: Estimated regression coefficients (standard errors) from logistic regression analysis

Variable	Complete-case	Available data	Last value carried fwd	IPW
Year	-0.312 (0.369)	0.016 (0.256)	0.006 (0.229)	0.004 (0.261)
Year^2	-0.502 (0.191)	-0.208 (0.140)	-0.198 (0.116)	-0.244 (0.146)
Year x Trt	0.082 (0.031)	0.033 (0.024)	0.032 (0.018)	0.040 (0.025)
Year^2 x Trt	0.336 (0.275)	0.046 (0.199)	0.062 (0.160)	0.052 (0.207)

```
  group_by(ID) %>%
  mutate(dropout = ifelse(is.na(mean(Y)), 1, 0)) %>%
  ungroup() %>%
  filter(dropout == 0)

ccgee <- geeglm(Y ~ treatment + Year + Year2 + trtYear + trtYear2,
         family = binomial("logit"),
         id = ID, scale.fix = TRUE,
         corstr = "unstructured",
         data = ccdat)


# available data
avdat <- skin_long %>%
  drop_na()
avgee <- geeglm(Y ~ treatment + Year + Year2 + trtYear + trtYear2,
                family = binomial("logit"),
                id = ID, scale.fix = TRUE,
                corstr = "unstructured",
                data = avdat)


# last value carried forward
lvcfdat <- skin_long %>%
  group_by(ID) %>%
  fill(Y) %>%
  ungroup()
lvcfgee <- geeglm(Y ~ treatment + Year + Year2 + trtYear + trtYear2,
                  family = binomial("logit"),
                  id = ID, scale.fix = TRUE,
                  corstr = "unstructured",
                  data = lvcfdat)
```

Summary

- IPW is useful if only the response variables are missing due to dropout
- IPW requires correct specification of the dropout model for valid estimation of β
- In the presence of discernibly large weights,
    - Check the sensitivity of results to the inclusion of observations that receive large weights
    - If the analysis results are sensitive to a small number of large weights, then
        - apply weight truncation
        - or consider an alternative methods of adjusting for missingness