Statistical Analyses with Missing Data

Day 2

Cindy Chen and Bryan Shepherd

Vanderbilt-Nigeria Biostatistics Workshop

June 2–6, 2025

Review from Day 1

- Defined missing data mechanisms
  - MCAR, MAR, NMAR
- Discussed simple (typically naive) analysis approaches
  - Complete case analysis
  - Single imputation
    - ⋆ Mean imputation, conditional expectation imputation, single imputation from fitted distribution, hot deck imputation
  - Missing data indicators
- Simulated data with different missing data mechanisms and performed some of these simple analysis approaches
- Began exploring the Nigerian renal function dataset

Outline for Today

Inverse probability weighted estimators

- Motivation
- Theory
- Estimation
- Standard Errors
- Bootstrap

Simulation exercises with IPW estimators

HIV data analysis with IPW estimators

Weighted means in Sample Surveys

Consider a survey where the goal is to estimate the population mean income.

- Target population
  - 30% of population is rural
  - 70% of population is urban
- Random sample: draw 10 of every 1000 people
  - Each person in sample represents 100 others
  - Everyone sampled with same probability has equal weight
- Computing sample mean

$$\overline{Y} = \frac{1}{n} \sum_{i=1} Y_i$$

Weighted means in Sample Surveys

- Probability sample
  - Draw 33 for every 1000 in rural areas
  - Draw 14 for every 1000 in urban areas
- Unequal representation
  - Each rural person represents 1000/33=30 others
  - Each urban person represents 1000/14=70 others
- Computing sample mean

$$\overline{Y} = \frac{\sum_i W_i Y_i}{\sum W_i},$$

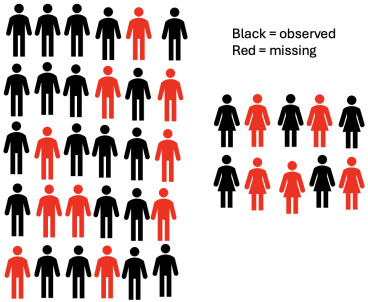
For rural, $W = 1/0.33 = 30$.

For urban, $W = 1/0.14 = 70$

Weights are inversely proportional to sampling probability

How does this apply to missing data?

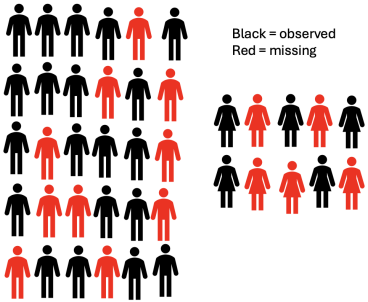
Suppose we are interested in measuring height in a sample with 30 males and 10 females



- Height is missing for 10 men and 5 women.
- For each observed woman, there is one missing woman. Each observed woman represents 2 women.
- Each woman is weighted by the inverse probability of being observed (2=1/0.5).
- $20/30 = 2/3$ of men have an observation. So each observed man represents 1.5 men.
- Each man is weighted by the inverse probability of being observed (1.5=1/(2/3))

How does this apply to missing data?

Suppose we are interested in measuring height in a sample with 30 males and 10 females



$$\overline{Y} = \frac{\sum_{i=1}^{N} W_i R_i Y_i}{\sum_{i=1}^{N} W_i R_i},$$

where R_i is the indicator an observation is observed and $W_i = \pi_i^{-1}$, with $\pi_i = P(R_i = 1|X_i) = 0.5$ for women and $2/3$ for men.

Notation

- The full data are $Y_1, Y_2, \ldots, Y_n$
- Each data element has
  - a response indicator R_i
  - a probability of being observed, $\pi_i = P(R_i = 1)$
- Key idea:
  - *The observed Y's are a nonrandom sample from the population*
- What would we do if we knew π_i? Compute weighted average

$$\overline{Y} = \frac{\sum_{i=1}^{N} W_i R_i Y_i}{\sum_{i=1}^{N} W_i R_i},$$

where $W_i = 1/\pi_i$ is the inverse probability that Y_i is observed

Examine the weighted mean

The weighted mean is

$$\overline{Y} = \frac{\sum_{i=1}^{N} W_i R_i Y_i}{\sum_{i=1}^{N} W_i R_i},$$

where $W_i = 1/\pi_i$ is the inverse probability that Y_i is observed

- Only those with $R_i = 1$ contribute, so it is a weighted average of respondents
- Each respondent counts for $1/\pi_i$ other individuals
  - If $\pi_i = 0.9$, then $W_i = 1/\pi_i = 1.11$
  - If $\pi_i = 0.2$, then $W_i = 1/\pi_i = 5$
- W_i is the inverse probability weight (IPW)

How do we know the weights?

If we knew the true values of the weights, estimation would be easy

In practice, we do not know this, so we must estimate the weights from data

How we do this depends on missing data assumptions being made

Using IPW to estimate a mean under MAR

- The setting:
  - Full data sample is $Y_1, \ldots, Y_n$
  - Indicators are $R_1, \ldots, R_n$
  - Also have covariate for each person, $X_1, \ldots, X_n$
- Recall MAR: $Y \perp R|X$
- MAR implies

$$P(R = 1|Y, X) = P(R = 1|X)$$

  or that the sampling probabilities only depend on X.
- Suppose first that we *knew* the sampling probabilities as a function of X

IPW theory

- Define $\pi(X_i) = P(R_i = 1|X_i)$
- Assume MAR
- Assume $\pi(X_i) > 0$ for all i
- The weighted estimator

$$\hat{E}_{IPW}(Y) = \frac{\sum_{i=1} R_i Y_i / \pi(X_i)}{\sum_{i=1} R_i / \pi(X_i)}$$

  is a *consistent* estimator of $E(Y)$.
- i.e., $\hat{E}_{IPW}(Y) \to E(Y)$.

IPW under MAR

$\hat{E}_{IPW}(Y)$ is still a consistent estimator of $E(Y)$ when $\pi(X_i)$ is replaced by a consistent estimator $\hat{\pi}(X_i)$.

To estimate $\pi(X_i)$, one must specify a model such as

$$\text{logit } \pi(X_i) = \alpha + \beta X_i.$$

In this case,

$$\hat{\pi}(X_i) = \frac{\exp(\hat{\alpha} + \hat{\beta} X_i)}{1 + \exp(\hat{\alpha} + \hat{\beta} X_i)}$$

Other models can be used – but the model must yield consistent estimates of $\pi(X_i)$ for IPW to result in a consistent estimator of $E(Y)$.

Example

```
library("mice")
summary(airquality)
```

```
##      Ozone           Solar.R          Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

Estimate the mean Ozone

First, we create the complete data indicator, R_i

```
r<-with(airquality, ifelse(is.na(Ozone), 0, 1))
table(r)
```

```
## r
##   0   1
##  37 116
```

Estimate the mean Ozone; Creating model for $\pi_i = P(R_i = 1|X_i)$

```
pi.model1<-glm(r~Wind + Temp*Month + Day, family="binomial", data=airquality)
summary(pi.model1)
```

```
##
## Call:
## glm(formula = r ~ Wind + Temp * Month + Day, family = "binomial",
##     data = airquality)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 13.93971    9.00833   1.547   0.1218
## Wind        -0.03533    0.06284  -0.562   0.5740
## Temp        -0.22006    0.12226  -1.800   0.0719 .
## Month       -1.35866    1.44077  -0.943   0.3457
## Day         -0.01679    0.02300  -0.730   0.4654
## Temp:Month   0.02661    0.01908   1.395   0.1631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 169.27  on 152  degrees of freedom
## Residual deviance: 151.92  on 147  degrees of freedom
## AIC: 163.92
##
## Number of Fisher Scoring iterations: 4
```

Estimate the mean Ozone; Creating model for $\pi_i = P(R_i = 1|X_i)$

```
pi.model2<-glm(r~Wind + Temp + Month, family="binomial", data=airquality)
summary(pi.model2)
```

```
##
## Call:
## glm(formula = r ~ Wind + Temp + Month, family = "binomial", data = airquality)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.33541    2.09508   0.637 0.523864
## Wind        -0.04235    0.06145  -0.689 0.490721
## Temp        -0.05385    0.02698  -1.996 0.045934 *
## Month        0.65774    0.19895   3.306 0.000946 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 169.27  on 152  degrees of freedom
## Residual deviance: 154.66  on 149  degrees of freedom
## AIC: 162.66
##
## Number of Fisher Scoring iterations: 5
```

Estimate the mean Ozone; Estimating π_i

```
pi.est<-predict(pi.model2, type="response")
summary(pi.est)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4088  0.6776  0.7385  0.7582  0.8738  0.9636
```

```
#
#### Equivalently

lp <- pi.model2$coeff[1] + pi.model2$coeff[2]*airquality$Wind +
          pi.model2$coeff[3]*airquality$Temp + pi.model2$coeff[4]*airquality$Month

pi.est2 <- exp(lp)/(1+exp(lp))
summary(pi.est2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4088  0.6776  0.7385  0.7582  0.8738  0.9636
```

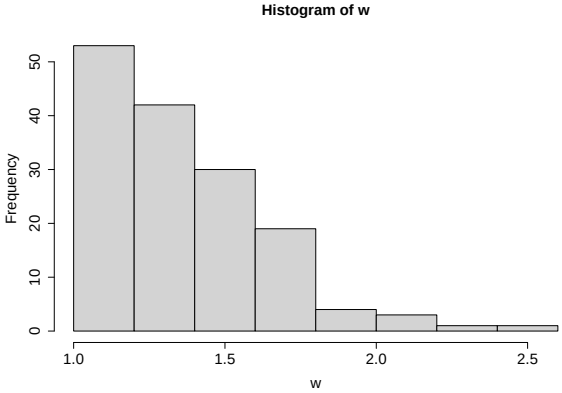
Estimate the mean Ozone; Estimating W_i

Creating inverse probability weights

```
w<-1/pi.est
```

Some measurements represent up to 2.5 other measurements, because measurements like these were often missing

If $\pi_i = 0$ for some i, then some of the weights might be huge. There is no evidence of this in the data.



Estimate the mean Ozone

Estimated mean of Ozone using IPW

```
sum(w*r*airquality$Ozone, na.rm=TRUE)/sum(w*r)
```

```
## [1] 42.15591
```

Estimated mean of Ozone using only those with available data

```
mean(airquality$Ozone, na.rm=TRUE)
```

```
## [1] 42.12931
```

Simulation Exercise 2A

Fit an IPW estimator of the mean of X for the following data you generated on Day 1:

1. Missing data that are MCAR
2. Missing data where $R_x \perp (X, Y)|V$
3. Missing data where $R_x \perp (V, Y)|X$
4. Missing data where $R_x \perp (V, X)|Y$

How do these estimates compare to the mean of X based on the full data?

IPW and regression

- Many of the same principles apply when using IPW in the regression context
- The key is the MAR assumption and specifying the model for the probability of response, $\pi(X_i)$.

IPW for regression models

- Full data for each individual: Y, R, V, X
  - R is observed data indicator for Y
  - X is a set of covariates (may have multiple elements)
  - V is a set of auxiliary covariates
- Recall that $\mu(X) = E(Y|X)$
- We are interested in the regression model

$$g[\mu(X)] = X^T \beta$$

- The selection probability for person i is

$$\pi_i = P(R_i = 1|X_i, V_i)$$

What if we knew the selection probability?

- The IPW method could be carried out by fitting a weighted regression model to those with observed Y data.
- The weights would be $W_i = 1/\pi_i$
- In practice we rarely know the weights, so we have to estimate them from available information
- Informally, MAR says that the available information is sufficient for estimating the weights.
  - But we still need to specify a model for this

Implications of MAR for estimation by IPW

MAR provides a justification for using inverse probability weighting.

- IPW requires us to weight inversely by the selection probability

$$\pi_i = P(R = 1|X_i, V_i, Y_i)$$

- Under MAR, this simplifies to

$$\pi_i = P(R = 1|X_i, V_i)$$

- If we accept MAR, then we have to specify a model for $P(R = 1|X, V)$

Algorithm for IPW estimation

1. Specify the regression model you are interested in. This is the model you would fit if all the data were observed (i.e., the *full data* model)

$$g[\mu(X)] = X^T\beta$$

By 'specify', we mean to write down how the mean of Y is related to the covariates X

- Functional form of the covariates
- Which covariates are included
- Interactions?
- Splines / non-linear terms?

Algorithm for IPW estimation

2. Specify the selection model in terms of X and V; for example

$$\text{logit } \pi(X_i, V_i) = \alpha_0 + \alpha_1 X_i + \alpha_2 V_i$$

- As before, need to decide what variables to include, whether there are interaction, etc.
- The objective is to get good predictions of π

3. Obtain predicted values of π from the fitted selection model.

4. Fit a weighted regression to the observed data, where the weights are $W_i = 1/\hat{\pi}_i$

Example: Estimating Assocation between Ozone and Solar.R

```
library("mice")
summary(airquality)
```

```
##      Ozone           Solar.R          Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

Example: Estimating Assocation between Ozone and Solar.R

```
rm(r,w,pi.est)
r.O<-with(airquality, ifelse(is.na(Ozone), 0, 1))
r.S<-with(airquality, ifelse(is.na(Solar.R),0,1))
table(r.O,r.S)
```

```
##    r.S
## r.O   0   1
##   0   2  35
##   1   5 111
```

```
chisq.test(table(r.O,r.S))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(r.O, r.S)
## X-squared = 1.799e-29, df = 1, p-value = 1
```

```
r<-r.O*r.S
table(r)
```

```
## r
##   0   1
##  42 111
```

Example: Estimating Assocation between Ozone and Solar.R

We already fit a model for the probability that Ozone is missing:

```
pi.model.O<-glm(r.O~Wind + Temp + Month, family="binomial", data=airquality)
pi.est.O<-predict(pi.model.O, type="response")
```

Example: Estimating Assocation between Ozone and Solar.R

Now let's consider a model for the Probability that Solar.R is missing

```
pi.model.S1<-glm(r.S~Wind, family="binomial", data=airquality)
pi.model.S2<-glm(r.S~Temp, family="binomial", data=airquality)
pi.model.S3<-glm(r.S~Month, family="binomial", data=airquality)
pi.model.S4<-glm(r.S~Day, family="binomial", data=airquality)
AIC(pi.model.S1)
```

```
## [1] 60.29451
```

```
AIC(pi.model.S2)
```

```
## [1] 58.9733
```

```
AIC(pi.model.S3)
```

```
## [1] 58.9606
```

```
AIC(pi.model.S4)
```

```
## [1] 56.3999
```

Example: Estimating Assocation between Ozone and Solar.R

It looks like the fourth model is the best

```
summary(pi.model.S4)
```

```
##
## Call:
## glm(formula = r.S ~ Day, family = "binomial", data = airquality)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.76346    0.64274   2.744  0.00608 **
## Day          0.10274    0.05395   1.904  0.05687 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 56.858  on 152  degrees of freedom
## Residual deviance: 52.400  on 151  degrees of freedom
## AIC: 56.4
##
## Number of Fisher Scoring iterations: 6
```

Example: Estimating Assocation between Ozone and Solar.R

The predicted probability of Solar.R being observed:

```
pi.est.S<-predict(pi.model.S4, type="response")
summary(pi.est.S)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8660  0.9299  0.9679  0.9542  0.9841  0.9930
```

The predicted probability of both Ozone and Solar.R being observed can be estimated as the product of the predicted probabilities of each individually

$$\begin{aligned} P(R = 1|X) &= P(R_O = 1, R_S = 1|X) \\ &= P(R_O = 1|R_S = 1, X)P(R_S = 1|X) \\ &\approx P(R_O = 1|X)P(R_S = 1|X) \end{aligned}$$

```
pi.est<-pi.est.O*pi.est.S
summary(pi.est)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4053  0.6287  0.7213  0.7237  0.8242  0.9497
```

Example: Estimating Assocation between Ozone and Solar.R

Alternatively, directly model probability that both Ozone and Solar.R are Observed:

```
pi.model2<-glm(r~Wind + Temp + Month + Day, family="binomial", data=airquality)
pi.est2<-predict(pi.model2, type="response")
summary(pi.est2)
```

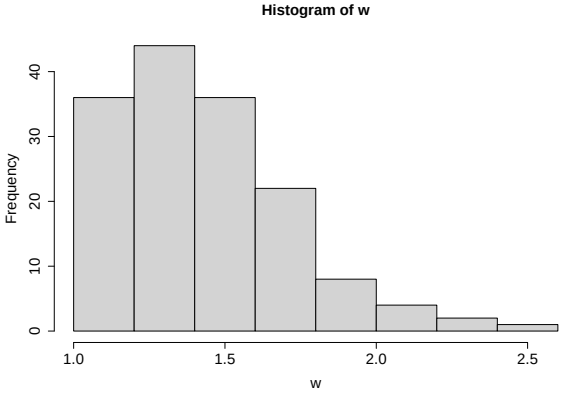
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4233  0.6338  0.7097  0.7255  0.8425  0.9562
```

Example: Estimating Assocation between Ozone and Solar.R

Creating inverse probability weights

```
w <- 1/pi.est
```

Some measurements represent up to 2.5 other measurements, because measurements like these were often missing

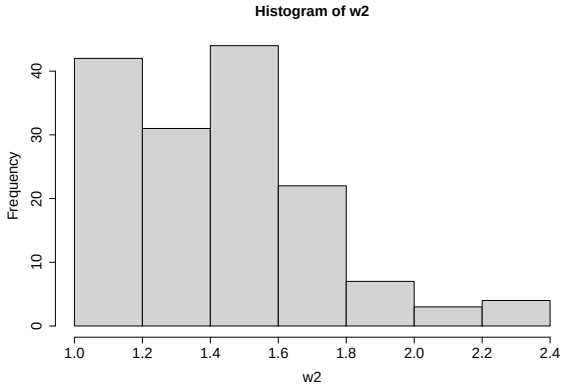


Example: Estimating Assocation between Ozone and Solar.R

Creating inverse probability weights based on the other model for $P(R = 1|X)$

```
w2 <- 1/pi.est2
```

Slight differences, but fairly similar distribution of weights.



Example: Estimating Assocation between Ozone and Solar.R

Now let's fit the inverse probability weighted regression model using the first set of weights

```
fit<-lm(Ozone ~ Solar.R, data=airquality, weights=w, subset= r==1)
summary(fit)
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R, data = airquality, subset = r ==
##     1, weights = w)
##
## Weighted Residuals:
##    Min     1Q Median     3Q    Max
## -67.64 -25.01 -10.19  22.46 127.67
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.46185    6.83298   2.848 0.005256 **
## Solar.R      0.12133    0.03251   3.732 0.000304 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.85 on 109 degrees of freedom
## Multiple R-squared:  0.1133, Adjusted R-squared:  0.1052
## F-statistic: 13.93 on 1 and 109 DF,  p-value: 0.0003036
```

Example: Estimating Assocation between Ozone and Solar.R

Repeating with the second set of weights. Very similar results.

```
fit2<-lm(Ozone ~ Solar.R, data=airquality, weights=w2, subset= r==1)
summary(fit2)
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R, data = airquality, subset = r ==
##     1, weights = w2)
##
## Weighted Residuals:
##    Min     1Q Median     3Q    Max
## -66.77 -24.95 -10.20  21.78 129.61
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.48948    6.88976   2.829 0.005563 **
## Solar.R      0.12261    0.03273   3.746 0.000289 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.03 on 109 degrees of freedom
## Multiple R-squared:  0.1141, Adjusted R-squared:  0.1059
## F-statistic: 14.03 on 1 and 109 DF,  p-value: 0.000289
```

Example: Estimating Assocation between Ozone and Solar.R

Let's compare this with the complete case analysis:

```
fit0<-lm(Ozone ~ Solar.R, data=airquality)
summary(fit0)
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R, data = airquality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.292 -21.361  -8.864  16.373 119.136
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.59873    6.74790   2.756 0.006856 **
## Solar.R      0.12717    0.03278   3.880 0.000179 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.33 on 109 degrees of freedom
##   (42 observations deleted due to missingness)
## Multiple R-squared:  0.1213, Adjusted R-squared:  0.1133
## F-statistic: 15.05 on 1 and 109 DF,  p-value: 0.0001793
```

Simulation Exercise 2B

Fit an IPW estimator for the regression coefficients of the model

$$E(Y|X) = \beta_0 + \beta_1 X + \beta_2 V$$

using the data you generated on Day 1:

1. Missing data that are MCAR
2. Missing data where $R_x \perp (X, Y)|V$
3. Missing data where $R_x \perp (V, Y)|X$
4. Missing data where $R_x \perp (V, X)|Y$

How do these estimates compare to the regression estimates based on the full data?

Standard Errors for IPW Estimators

- Standard error (SE) calculations may need to be adjusted because of the variability introduced by using estimated weights
- Formulas are based on sandwich variance estimators and large sample approximations
- An alternative is to use the bootstrap
- Interestingly, the SE that accounts for model fitting uncertainty in estimating weights is usually lower than the SE that does not

What is the bootstrap?

What is the standard error (SE)?

- Consider a sample $Y_1, \ldots, Y_n$
- The sample mean is $\overline{Y} = \sum_i Y_i / n$
- The standard error is $S/\sqrt{n}$, where S is the sample standard deviation
- The SE measures the average distance between a sample mean and the true mean *in repeated sampling from the population*
- Often overlooked: the SE of the sample mean is the *standard deviation of its sampling distribution*
- The problem in understanding SE: we do not have the luxury of repeated sampling.

The bootstrap

- The bootstrap is designed to simulate replicated sampling
- The procedure is easy; the theory is complicated
- The idea:
  1. Draw a sample of size n of your observed (Y, X, R, V) with replacement
  2. Calculate the sample mean (or any other statistic you are interested in)
  3. Repeat this lots and lots of times. The repeated calculations of the statistics resemble its sampling distribution
  4. The estimated SE of your statistic is the standard deviation of the replicates.
  5. Confidence intervals (CIs) can be calculated using the estimated SE or using percentiles of the bootstrapped statistics. e.g., 95% CIs:
     - ⋆ estimate $\pm 1.96 \times \widehat{\text{SE}}$
     - ⋆ or 2.5th and 97.5th percentiles of the replicate bootstrap estimates

Why the bootstrap is helpful

- For some statistics, the sampling distribution is well understood
  - Sample mean
  - Regression estimates

  In these and other cases, we know the sampling distribution is normal in large samples and we know how to estimate its variance

- For example, we know that

$$\overline{Y} \sim N(\mu, \sigma^2/n)$$

  From this, we can calculate SE, confidence intervals, p-values, etc.

- For other estimators, the sampling distribution is more difficult to work out
  - Medians
  - Estimators from multi-step procedures like IPW and imputation

Example: Using Bootstrap to Compute SE for Assocation between Ozone and Solar.R

I am going to start by creating a dataset with everything I need.

```
d<-airquality
d$r.O<-r.O
d$r.S<-r.S
d$r<-r
```

Example: Using Bootstrap to Compute SE for Ozone ∼ Solar.R

Randomly sample rows with replacement from the dataset and repeat all IPW estimation steps nboot times, saving coefficients for each replication

```
nboot<-1000
set.seed(100)
beta0s<-beta1s<-NULL
for (i in 1:nboot){
  samp<-sample(1:length(d$Ozone), length(d$Ozone), replace=TRUE)
  dboot<-d[samp,]
  pi.model.O<-glm(r.O~Wind + Temp + Month, family="binomial", data=dboot)
  pi.est.O<-predict(pi.model.O, type="response")
  pi.model.S4<-glm(r.S~Day, family="binomial", data=dboot)
  pi.est.S<-predict(pi.model.S4, type="response")
  pi.est<-pi.est.O*pi.est.S
  w<-1/pi.est
  fit<-lm(Ozone ~ Solar.R, data=dboot, weights=w, subset= r==1)
  beta0s[i]<-fit$coeff[1]
  beta1s[i]<-fit$coeff[2]
}
```

Example: Using Bootstrap to Compute SE for Ozone ∼ Solar.R

Estimating SE and 95% CI

```
sd(beta1s)
```

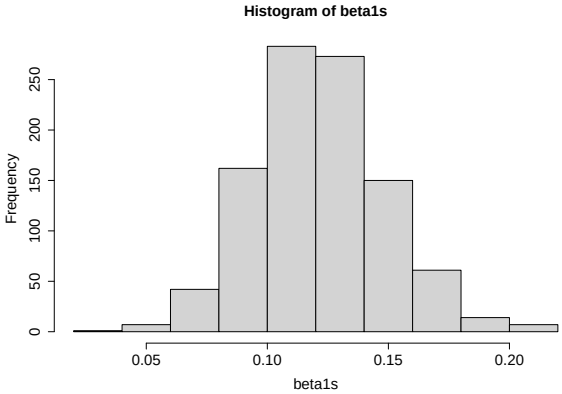
```
## [1] 0.0269234
```

```
fit$coeff[2] + c(-1,1)*1.96*sd(beta1s)
```

```
## [1] 0.05917842 0.16471816
```

```
quantile(beta1s, c(.025,.975))
```

```
##       2.5%      97.5%
## 0.07001759 0.17743170
```



Compare with the standard deviation without bootstrapping (ignoring the fact that the weights were estimated):

```
summary(fit)$coeff
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 20.9591509 5.74758847 3.646599 4.006126e-04
## Solar.R      0.1119483 0.02754957 4.063522 8.870435e-05
```

Simulation Exercise 2C

Compute standard errors and 95% confidence intervals using the bootstrap for the regression estimates of the model

$$E(Y|X) = \beta_0 + \beta_1 X + \beta_2 V$$

using the data you generated on Day 1:

1. Missing data that are MCAR
2. Missing data where $R_x \perp (X, Y)|V$
3. Missing data where $R_x \perp (V, Y)|X$
4. Missing data where $R_x \perp (V, X)|Y$
5. The full data (no missing data)

How do these standard errors and 95% confidence intervals compare to those that do not use the bootstrap?

Assumptions needed for IPW

- Missing at random: $Y \perp R|(X, V)$
  - This allows us to specify a selection model in terms of X and V only, because selection does not depend on Y.
- Selection model correctly specified
  - Needed to obtain consistent estimates of π
- Positivity condition:
  - There is a nonzero probability of observing Y within each (X, V) covariate profile.

$$P(R = 1|X, V) > 0$$

Advanced Material: Doubly Robust Estimators

Suppose we have data that are MAR: $Y \perp R|(X, V)$

Suppose we are interested in estimating $E(Y)$

We just talked about how to estimate $E(Y)$ using IPW estimators:

- Fit model of $P(R = 1|X, V)$
- Fitted weighted mean of $E(Y)$ based on the inverse probability weights
- This estimator will be consistent if the model for $P(R = 1|X, V)$ is correctly specified
- But what if it is not correctly specified?

Simulation Example

Generate $V \sim N(0,1)$; $X \sim N(V,1)$; and $Y \sim N(X+V^2,1)$.

$R \sim$ Bernoulli(π) where $\pi = \exp(X+V^2)/\exp(X+V^2)$

Note that there are quadratic terms for the associations between V and both Y and R.

I use a big sample size so that estimates are approximately the truth.

```
rm(list=ls())
n<-100000
v<-rnorm(n,0,1)
x<-rnorm(n,v,1)
y1<-rnorm(n,x+v^2,1)
p<-exp(x+v^2)/(1+exp(x+v^2))
r<-rbinom(n,1,p)
y<-ifelse(r==1,y1,NA)
d<-data.frame(y,x,v,r)
```

Simulation Example

The true $E(Y)$ is 1.

The estimated mean based on the full data:

```
mean(y1)
```

```
## [1] 1.000374
```

If we estimate the mean only among those with observed data, then we get biased estimates:

```
mean(y, na.rm=TRUE)
```

```
## [1] 1.689746
```

Simulation Example: Fitting IPW estimator

Obtaining IPW estimator using the correctly specified model for π:

```
modr<-glm(r~x+I(v^2), family="binomial", data=d)
modr$coeff
```

```
## (Intercept)           x      I(v^2)
## 0.008833634 0.990480268 0.977772055
```

```
pi.hat<-predict(modr, type="response")
summary(pi.hat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01697 0.43452 0.65171 0.63261 0.85406 1.00000
```

```
sum(r*y/pi.hat,na.rm=TRUE)/sum(r/pi.hat,na.rm=TRUE)
```

```
## [1] 1.003671
```

```
mod.ipw<-lm(y~1, weight=1/pi.hat)
mod.ipw$coeff
```

```
## (Intercept)
##    1.003671
```

Simulation Example: Fitting Misspecified IPW estimator

But suppose we fit an IPW estimator that did not include the quadratic term in the model for π:

```
modr2<-glm(r~x+v, family="binomial", data=d)
modr2$coeff
```

```
## (Intercept)           x           v
##   0.6556807   0.8590575  -0.3396454
```

```
pi.hat2<-predict(modr2, type="response")
summary(pi.hat2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02298 0.49569 0.65822 0.63261 0.79144 0.99553
```

```
mod.ipw2<-lm(y~1, weight=1/pi.hat2)
mod.ipw2$coeff
```

```
## (Intercept)
##    1.369007
```

IPW estimation

- This simulation demonstrates the importance of correctly specifying the model for π
- But in practice, this can be very hard to do, even if you have all of the relevant covariates
- Real data do not always follow nicely behaved linear models

Outcome Regression estimator:

- Since data are MAR: $E(Y|X, V, R = 1) = E(Y|X, V)$
- So a consistent estimator of $E(Y|X)$ could be motivated by:

$$E(Y) = E[E(Y|X, V)] = E[E(Y|X, V, R = 1)]$$

- In words, fit a regression model of Y given X and V among those with non-missing data;
- Then compute the fitted (predicted) value of Y based on observed X and V for everyone, including those with missing data; we will call this $\hat{Y}$;
- Finally, take the mean of $\hat{Y}$ using everyone.
- The estimated mean will be consistent if model of $E(Y|X, V)$ is correctly specified.
- The standard error would need to be estimated by the bootstrap or some other approach.

Simulation Example: Fitting Outcome Regression estimator

Obtaining outcome regression estimator using the correctly specified model for $E(Y|X, V)$:

```
mody<-lm(y~x+I(v^2), subset=r==1, data=d)
mody$coeff
```

```
## (Intercept)           x      I(v^2)
##   0.0072570   0.9985562   0.9976918
```

```
yhat<-predict(mody, newdata=d)
mean(yhat)
```

```
## [1] 1.003024
```

```
mod.or<-lm(yhat~1)
mod.or$coeff
```

```
## (Intercept)
##    1.003024
```

Estimates of $E(Y)$ is unbiased.

Simulation Example: Fitting Misspecified Outcome Regression estimator

Obtaining outcome regression estimator using the misspecified model for $E(Y|X, V)$ that does not include the quadratic term:

```
mody2<-lm(y~x+v, subset=r==1, data=d)
mody2$coeff
```

```
## (Intercept)           x           v
##   1.3134860   0.8134300   0.2083227
```

```
yhat2<-predict(mody2, newdata=d)
mean(yhat2)
```

```
## [1] 1.316363
```

Estimate of $E(Y)$ is now biased.

Doubly Robust Estimators

Even when data are MAR: $Y \perp R|X, V$

- The IPW estimator is biased if $P(R|X, V)$ is misspecified
- The outcome regression estimator is biased if $E(Y|X, V)$ is misspecified

What if we give ourselves two chances to correctly specify the model?

- This is the idea behind a doubly robust estimator

A doubly robust estimator for $E(Y)$ is consistent if $P(R|X, V)$ *or* $E(Y|X, V)$ is correctly specified

Doubly Robust Estimators

There are lots of doubly robust estimators. Here is one of them:

$$\hat{E}_{DR}(Y) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{R_i Y_i}{\pi(X_i, V_i; \hat{\alpha})} - \frac{R_i - \pi(X_i, V_i; \hat{\alpha})}{\pi(X_i, V_i; \hat{\alpha})} m(X_i, V_i; \hat{\beta})\right]$$

This is sometimes referred to as an Augmented Inverse Probability Weighted (AIPW) estimator.

$\hat{E}_{DR}(Y)$ is consistent estimator of $E(Y)$ if at least one of these two conditions holds:

- $\pi(X_i, V_i; \alpha)$ is correctly specified (i.e., consistent for $P(R = 1|X, V)$)
- $m(X_i, V_i; \beta)$ is correctly specified (i.e., consistent for $E(Y|X, V)$)

Simulated Doubly Robust Estimator

$P(R = 1|X, V)$ misspecified, $E(Y|X, V)$ properly specified:

```
sum(r*y/pi.hat2,na.rm=TRUE)/n - sum(((r-pi.hat2)/pi.hat2)*yhat)/n
```

```
## [1] 1.001733
```

$P(R = 1|X, V)$ properly specified, $E(Y|X, V)$ misspecified:

```
sum(r*y/pi.hat,na.rm=TRUE)/n - sum(((r-pi.hat)/pi.hat)*yhat2)/n
```

```
## [1] 1.005081
```

Simulated Doubly Robust Estimator

Both models correctly included the quadratic term:

```
sum(r*y/pi.hat,na.rm=TRUE)/n - sum(((r-pi.hat)/pi.hat)*yhat)/n
```

```
## [1] 1.006115
```

Both $P(R = 1|X, V)$ and $E(Y|X, V)$ misspecified:

```
sum(r*y/pi.hat2,na.rm=TRUE)/n - sum(((r-pi.hat2)/pi.hat2)*yhat2)/n
```

```
## [1] 1.496396
```

Summary

Today we have

- Presented inverse probability weighted (IPW) estimators
- Motivated IPW estimators
- Described assumptions needed for these to result in consistent estimators
- Described estimation procedures
- Discussed SEs and how to calculate with the bootstrap
- Introduced doubly robust estimators