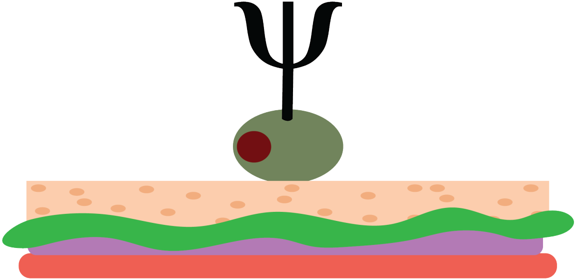
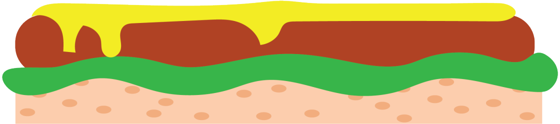


ABC's of M-estimation



Paul Zivich, Rachael Ross, Stephen Cole

University of North Carolina at Chapel Hill

Overview

Section 1: introduction

Section 2: applied examples

Overview: Section 2 - Applied Examples

1. Logistic regression
2. Standardization
   - Outcome model based standardization
   - Inverse probability weighting
3. Data fusion

Each section will include time to work with code

As the goal is to illustrate implementation, we will not dwell on identification assumptions of these examples[1]

[1] Hernan & Robins Causal Inference: What If; Cole et al. *AJE* 2012

Notation – Reminder

O_i: observed data for unit i

- $O_i = (W_i, X_i, Y_i)$
- $O_i = (R_i, W_i, Y_i)$

expit$(a) = 1/(1 + \exp(-a))$

Notation – Reminder

Estimating function

$$\psi(O_i; \theta)$$

Estimating equation

$$\sum_{i=1}^{n} \psi(O_i; \theta)$$

Logistic regression

Example

Data from the Zambia Preterm Birth Prevention Study (ZAPPS)[2]
$O_i = (Y_i, X_i, W_i)$
Y_i, preterm birth
X_i, anemia
W_i, elevated blood pressure

We want to estimate

$$\begin{aligned}\Pr(Y_i = 1) &= \text{expit}(\beta_0 + \beta_1 X_i + \beta_2 W_i) \\ &= \text{expit}(\boldsymbol{Z}_i \boldsymbol{\beta}^T)\end{aligned}$$

where $\boldsymbol{Z}_i = (1, X_i, W_i)$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)$

[2] Castillo et al. *Gates Open Research* 2019

ZAPPS data

$n = 826$ pregnant people

14.2% delivered preterm

14.3% diagnosed with anemia

18.9% diagnosed with elevated blood pressure

Estimating β by MLE

Let $\tilde{\boldsymbol{\beta}}$ be the estimate of $\boldsymbol{\beta}$ by MLE

$\tilde{\boldsymbol{\beta}}$ are the values that maximize the log-likelihood

$$\sum_{i=1}^{n} \ln\big(L(\boldsymbol{O_i}, \tilde{\boldsymbol{\beta}})\big)$$

$$= \sum_{i=1}^{n} Y_i \ln\big(\text{expit}(\boldsymbol{Z}_i \tilde{\boldsymbol{\beta}}^T)\big) + (1 - Y_i) \ln\big(1 - \text{expit}(\boldsymbol{Z}_i \tilde{\boldsymbol{\beta}}^T)\big)$$

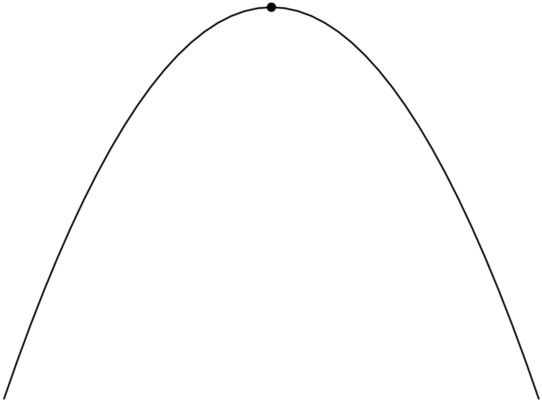
Estimating $\tilde{\beta}$: Code

Under `Regression by MLE`

- SAS - `proc logistic` or `genmod`
- R - `stats::glm`
- Python - `statsmodels.GLM`

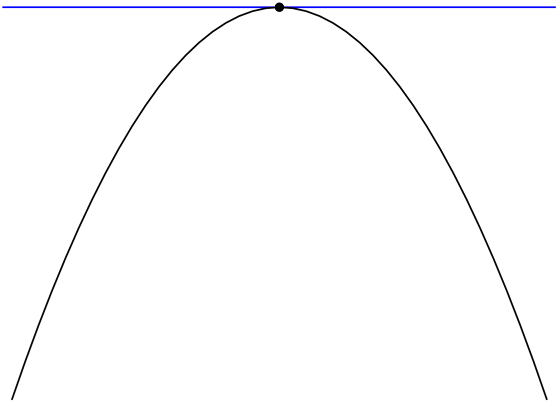
Estimating β by root-finding

Maximum of likelihood is where slope (derivative) is zero ("root")



Estimating β by root-finding

Maximum of likelihood is where slope (derivative) is zero ("root")



Estimating β by root-finding

Our log-likelihood is

$$\ln\big(L(\boldsymbol{O_i}, \boldsymbol{\beta})\big) = Y_i \ln\big(\text{expit}(\boldsymbol{Z}_i \tilde{\boldsymbol{\beta}}^T)\big) + (1 - Y_i)\ln\big(1 - \text{expit}(\boldsymbol{Z}_i \tilde{\boldsymbol{\beta}}^T)\big)$$

Functions for the slope (derivative[3])

$$\boldsymbol{\psi}(\boldsymbol{O_i}, \boldsymbol{\beta}) = \frac{\partial \ln\big(L(\boldsymbol{O_i}, \boldsymbol{\beta})\big)}{\partial \boldsymbol{\beta}} = \begin{bmatrix} Y_i - \text{expit}(\boldsymbol{Z}_i \boldsymbol{\beta}^T) \\ \big(Y_i - \text{expit}(\boldsymbol{Z}_i \boldsymbol{\beta}^T)\big) X_i \\ \big(Y_i - \text{expit}(\boldsymbol{Z}_i \boldsymbol{\beta}^T)\big) W_i \end{bmatrix}$$

[3] first-order partial derivatives, called the "score" functions

Estimating β by root-finding

Let $\hat{\boldsymbol{\beta}}$ be the estimate of $\boldsymbol{\beta}$ by root finding (M-estimation)

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T) \\ \big(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\big) X_i \\ \big(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\big) W_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Estimating $\hat{\beta}$: Code

To implement, we need to define the estimating functions

Under `Defining estimating equation`

- SAS

```
p = 1 / (1 + exp(-(beta[1] + beta[2]*anemia + beta[3]*bp)));

ef_1 = ptb - p;
ef_2 = (ptb - p)#anemia;
ef_3 = (ptb - p)#bp;
```

Estimating $\hat{\beta}$: Code

To implement, we need to define the estimating functions

Under `Defining estimating equation`

- R

```
p <- plogis(beta[1] + beta[2]*dat$anemia + beta[3]*dat$bp)

ef_1 <- (dat$ptb - p)
ef_2 <- (dat$ptb - p)*dat$anemia
ef_3 <- (dat$ptb - p)*dat$bp
```

Estimating $\hat{\beta}$: Code

To implement, we need to define the estimating functions

Under `Defining estimating equation`

- Python

```
yhat = inverse_logit(theta[0]*1 +
                     theta[1]*d['X'] + theta[2]*d['W'])

residual = d['Y'] - yhat

score = [residual*1, residual*d['X'], residual*d['W']]
```

Estimating $\hat{\beta}$: Code

Under `Root-finding`

- SAS – `nlplm`
- R – `rootSolve::multiroot`
- Python – `scipy.optimize.root`

$\tilde{\beta}$ and $\hat{\beta}$ from R

	β_0	β_1	β_2
$\tilde{\boldsymbol{\beta}}$	-1.89450081814**3**	0.1187353484**2**8	0.3605113262**7**6
$\hat{\boldsymbol{\beta}}$	-1.89450081814**8**	0.1187353484**3**2	0.3605113262**8**3

Difference is result of algorithm/tolerance

Variance

Let $\boldsymbol{V}(\boldsymbol{\beta})$ be the 3×3 covariance matrix of $\boldsymbol{\beta}$

The diagonal vector of $\boldsymbol{V}/n$ (i.e., $\text{diag}(\boldsymbol{V}/n)$) are the finite sample variances

For Wald-type confidence intervals, we use $\text{SE}(\boldsymbol{\beta}) = \sqrt{\text{diag}(\boldsymbol{V}/n)}$

Estimating $\boldsymbol{V}(\tilde{\boldsymbol{\beta}})$ by MLE

$$\boldsymbol{V}(\tilde{\boldsymbol{\beta}}) = \boldsymbol{I}(\tilde{\boldsymbol{\beta}})^{-1}$$

where $\boldsymbol{I}(\tilde{\boldsymbol{\beta}})$ is the observed information matrix at $\tilde{\boldsymbol{\beta}}$

Estimating $\boldsymbol{V}(\tilde{\boldsymbol{\beta}})$

There are two ways to estimate $\boldsymbol{I}(\tilde{\boldsymbol{\beta}})$

1. Hessian-based estimator

$$\boldsymbol{I}(\tilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n}\left[-\boldsymbol{\psi}'(\boldsymbol{O_i}, \tilde{\boldsymbol{\beta}})\right] \text{ where } \boldsymbol{\psi}' = \frac{\partial \boldsymbol{\psi}(\boldsymbol{O_i}, \tilde{\boldsymbol{\beta}})}{\partial \tilde{\boldsymbol{\beta}}}$$

   $\boldsymbol{\psi}'$ are first-order partial derivatives of the score

2. Residual-based estimator

$$\boldsymbol{I}(\tilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n}\left[\boldsymbol{\psi}(\boldsymbol{O_i}, \tilde{\boldsymbol{\beta}})\boldsymbol{\psi}(\boldsymbol{O_i}, \tilde{\boldsymbol{\beta}})^T\right]$$

Asymptotically equal when chosen parametric family is correct
Hessian is more efficient and default (`logistic`, `glm`, `GLM`)

Estimating $\boldsymbol{V}(\hat{\boldsymbol{\beta}})$ by sandwich variance estimator

$$\boldsymbol{V}(\hat{\boldsymbol{\beta}}) = \boldsymbol{B}(\hat{\boldsymbol{\beta}})^{-1}\boldsymbol{F}(\hat{\boldsymbol{\beta}})\left[\boldsymbol{B}(\hat{\boldsymbol{\beta}})^{-1}\right]^T$$

Recall

- $\boldsymbol{B}(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\sum_i \left[-\boldsymbol{\psi}'(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}})\right]$
  This corresponds to the Hessian-based info matrix estimator
- $\boldsymbol{F}(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\sum_i \left[\boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}})\boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}})^T\right]$
  This corresponds to the residual-based info matrix estimator

Asymptotic equivalence

When chosen parametric family is correct (here, logistic)

$$\begin{aligned}\boldsymbol{V}(\boldsymbol{\beta}) &= \boldsymbol{B}(\boldsymbol{\beta})^{-1}\boldsymbol{F}(\boldsymbol{\beta})\left[\boldsymbol{B}(\boldsymbol{\beta})^{-1}\right]^{T} \\ &\sim \boldsymbol{I}(\boldsymbol{\beta})^{-1}\boldsymbol{I}(\boldsymbol{\beta})\left[\boldsymbol{I}(\boldsymbol{\beta})^{-1}\right]^{T} \\ &= \boldsymbol{I}(\boldsymbol{\beta})^{-1}\end{aligned}$$

Estimating $V(\hat{\beta})$: code

Under `Baking the bread`

- SAS – `nlpfdd`
- R – `numDeriv::jacobian`
- Python – `scipy.optimize.approx_fprime`

Estimating $V(\hat{\beta})$: code

Under `Cooking the Filling`

- Transpose
  - SAS – `'`
  - R – `base::t`
  - Python – `numpy.transpose`
- Dot product
  - SAS – `*`
  - R – `%*%`
  - Python – `numpy.dot`

Estimating $V(\hat{\beta})$: code

Under `Assembling the Sandwich`

- Inverse
  - SAS – `inv`
  - R – `base::solve()`
  - Python – `numpy.linalg.inv`

Standard errors from R

	β_0	β_1	β_2
MLE, Hessian-based	0.01**4**96046	0.077**6**4837	0.056**6**0453
MLE, Residual-based	0.01**5**08328	0.077**6**1366	0.056**7**0628
Sandwich estimator	0.01**4**84043	0.077**7**2035	0.056**5**2969

Time to work with code

Standardization

Standardization

Using the same data as the prior example
$O_i = (Y_i, X_i, W_i)$
Y_i, preterm birth
X_i, anemia
W_i, elevated blood pressure

We want to estimate the marginal risk difference and risk ratio of anemia on preterm birth, standardized by elevated blood pressure.

We will illustrate two standardization approaches

- Outcome model based standardization (i.e., g-computation)
- Inverse probability weighting

Outcome model based standardization

Following implementation illustrated in Snowden et al. AJE 2011

1. Estimate outcome model (from prior example)

$$\text{Pr}(Y_i = 1) = \text{expit}(\beta_0 + \beta_1 X_i + \beta_2 W_i) = \text{expit}(\boldsymbol{Z}_i \boldsymbol{\beta}^T)$$

Outcome model based standardization

Following implementation illustrated in Snowden et al. AJE 2011

1. Estimate outcome model (from prior example)

$$\Pr(Y_i = 1) = \text{expit}(\beta_0 + \beta_1 X_i + \beta_2 W_i) = \text{expit}(\boldsymbol{Z}_i \boldsymbol{\beta}^T)$$

2. Use $\hat{\boldsymbol{\beta}}$ to predict outcome when $X_i = 1$. Take the mean.

$$\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^{n} \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 1 + \hat{\beta}_2 W_i)$$

Outcome model based standardization

Following implementation illustrated in Snowden et al. AJE 2011

1. Estimate outcome model (from prior example)

$$\Pr(Y_i = 1) = \text{expit}(\beta_0 + \beta_1 X_i + \beta_2 W_i) = \text{expit}(\boldsymbol{Z}_i \boldsymbol{\beta}^T)$$

2. Use $\hat{\boldsymbol{\beta}}$ to predict outcome when $X_i = 1$. Take the mean.

$$\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^{n} \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 1 + \hat{\beta}_2 W_i)$$

3. Use $\hat{\boldsymbol{\beta}}$ to predict outcome when $X_i = 0$. Take the mean.

$$\hat{\mu}_2 = \frac{1}{n} \sum_{i=1}^{n} \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 0 + \hat{\beta}_2 W_i)$$

Outcome model based standardization

Following implementation illustrated in Snowden et al. AJE 2011

1. Estimate outcome model (from prior example)

$$\Pr(Y_i = 1) = \text{expit}(\beta_0 + \beta_1 X_i + \beta_2 W_i) = \text{expit}(\boldsymbol{Z}_i \boldsymbol{\beta}^T)$$

2. Use $\hat{\boldsymbol{\beta}}$ to predict outcome when $X_i = 1$. Take the mean.

$$\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^{n} \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 1 + \hat{\beta}_2 W_i)$$

3. Use $\hat{\boldsymbol{\beta}}$ to predict outcome when $X_i = 0$. Take the mean.

$$\hat{\mu}_2 = \frac{1}{n} \sum_{i=1}^{n} \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 0 + \hat{\beta}_2 W_i)$$

4. Calculate the risk difference $\hat{\delta_1} = \hat{\mu}_1 - \hat{\mu}_2$ & log risk ratio $\hat{\delta_2} = \ln(\hat{\mu}_1 / \hat{\mu}_2)$

Outcome model based standardization

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T) \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) X_i \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) W_i \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 1 + \hat{\beta}_2 W_i) - \hat{\mu}_1 \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 0 + \hat{\beta}_2 W_i) - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Outcome model based standardization

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T) \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) X_i \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) W_i \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 1 + \hat{\beta}_2 W_i) - \hat{\mu}_1 \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 0 + \hat{\beta}_2 W_i) - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Outcome model based standardization

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T) \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) X_i \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) W_i \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 1 + \hat{\beta}_2 W_i) - \hat{\mu}_1 \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 0 + \hat{\beta}_2 W_i) - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Outcome model based standardization

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T) \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) X_i \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) W_i \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 1 + \hat{\beta}_2 W_i) - \hat{\mu}_1 \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 0 + \hat{\beta}_2 W_i) - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Outcome model based standardization

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T) \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) X_i \\ \left(Y_i - \text{expit}(\boldsymbol{Z}_i \hat{\boldsymbol{\beta}}^T)\right) W_i \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 1 + \hat{\beta}_2 W_i) - \hat{\mu}_1 \\ \text{expit}(\hat{\beta}_0 + \hat{\beta}_1 \times 0 + \hat{\beta}_2 W_i) - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Estimating functions: Code

- SAS

```
p = 1/(1+exp(-(theta[1] + theta[2]*anemia + theta[3]*bp)));
ef_1 = ptb - p;
ef_2 = (ptb - p)#anemia;
ef_3 = (ptb - p)#bp;

ef_r1 = 1/(1+exp(-(theta[1] + theta[2]*1 + theta[3]*bp)))
                                              - theta[4];

ef_r0 = 1/(1+exp(-(theta[1] + theta[2]*0 + theta[3]*bp)))
                                              - theta[5];

ef_rd = j(n,1,(theta[4] - theta[5]) - theta[6]);
ef_lnrr = j(n,1,log(theta[4]/theta[5]) - theta[7]);
```

Estimating functions: Code

- R

```
p <- plogis(beta[1] + beta[2]*dat$anemia + beta[3]*dat$bp)

ef_1 <- (dat$ptb - p)
ef_2 <- (dat$ptb - p)*dat$anemia
ef_3 <- (dat$ptb - p)*dat$bp

ef_r1 <- plogis(beta[1] + beta[2]*1 + beta[3]*dat$bp) - mu[1]
ef_r0 <- plogis(beta[1] + beta[2]*0 + beta[3]*dat$bp) - mu[2]

ef_rd <- (mu[1] - mu[2]) - delta[1]
ef_lnrr <- log(mu[1]/mu[2]) - delta[2]
```

Estimating functions: Code

- Python

```
ee_logit = ee_regression(theta=beta,
        y=d['Y'],
        X=d[['intercept', 'X', 'W']],
        model='logistic')

y1_hat = inverse_logit(np.dot(d1[['intercept', 'X', 'W']], beta))
y0_hat = inverse_logit(np.dot(d0[['intercept', 'X', 'W']], beta))

ee_r1 = y1_hat - mu1
ee_r0 = y0_hat - mu0

ee_rd = np.ones(d.shape[0])*((mu1 - mu0) - delta1)
ee_rr = np.ones(d.shape[0])*(np.log(mu1 / mu0) - delta2)
```

Inverse probability weighting

1. Estimate propensity score model

$$\Pr(X_i = 1) = \text{expit}(\alpha_0 + \alpha_1 W_i)$$

Inverse probability weighting

1. Estimate propensity score model

$$\text{Pr}(X_i = 1) = \text{expit}(\alpha_0 + \alpha_1 W_i)$$

2. Calculate the inverse probability weight

$$\hat{\pi}_i = \frac{X_i}{\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} + \frac{1 - X_i}{1 - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)}$$

Inverse probability weighting

1. Estimate propensity score model

$$\mathsf{Pr}(X_i = 1) = \mathsf{expit}(\alpha_0 + \alpha_1 W_i)$$

2. Calculate the inverse probability weight

$$\hat{\pi}_i = \frac{X_i}{\mathsf{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} + \frac{1 - X_i}{1 - \mathsf{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)}$$

3. Take a weighted mean of the outcome among those with $X_i = 1$

$$\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^{n} \{X_i Y_i \hat{\pi}_i\}$$

Inverse probability weighting

1. Estimate propensity score model

$$\Pr(X_i = 1) = \text{expit}(\alpha_0 + \alpha_1 W_i)$$

2. Calculate the inverse probability weight

$$\hat{\pi}_i = \frac{X_i}{\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} + \frac{1 - X_i}{1 - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)}$$

3. Take a weighted mean of the outcome among those with $X_i = 1$

$$\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^{n} \{X_i Y_i \hat{\pi}_i\}$$

4. Take a weighted mean of the outcome among those with $X_i = 0$

$$\hat{\mu}_2 = \frac{1}{n} \sum_{i=1}^{n} \{(1 - X_i) Y_i \hat{\pi}_i\}$$

Inverse probability weighting

1. Estimate propensity score model
$$\text{Pr}(X_i = 1) = \text{expit}(\alpha_0 + \alpha_1 W_i)$$
2. Calculate the inverse probability weight
$$\hat{\pi}_i = \frac{X_i}{\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} + \frac{1 - X_i}{1 - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)}$$
3. Take a weighted mean of the outcome among those with $X_i = 1$
$$\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^{n} \{X_i Y_i \hat{\pi}_i\}$$
4. Take a weighted mean of the outcome among those with $X_i = 0$
$$\hat{\mu}_2 = \frac{1}{n} \sum_{i=1}^{n} \{(1 - X_i) Y_i \hat{\pi}_i\}$$
5. Calculate the risk difference $\hat{\delta}_1 = \hat{\mu}_1 - \hat{\mu}_2$ & log risk ratio $\hat{\delta}_2 = \ln(\hat{\mu}_1 / \hat{\mu}_2)$

Inverse probability weighting

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} X_i - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i) \\ \left(X_i - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)\right) W_i \\ \frac{X_i Y_i}{\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} - \hat{\mu}_1 \\ \frac{(1-X_i) Y_i}{1-\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Inverse probability weighting

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} X_i - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i) \\ (X_i - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i))\, W_i \\ \frac{X_i Y_i}{\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} - \hat{\mu}_1 \\ \frac{(1-X_i) Y_i}{1-\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Inverse probability weighting

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} X_i - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i) \\ (X_i - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)) \, W_i \\ \frac{X_i Y_i}{\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} - \hat{\mu}_1 \\ \frac{(1 - X_i) Y_i}{1 - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Inverse probability weighting

Our estimating equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \begin{bmatrix} X_i - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i) \\ (X_i - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)) \, W_i \\ \frac{X_i Y_i}{\text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} - \hat{\mu}_1 \\ \frac{(1 - X_i) Y_i}{1 - \text{expit}(\hat{\alpha}_0 + \hat{\alpha}_1 W_i)} - \hat{\mu}_2 \\ (\hat{\mu}_1 - \hat{\mu}_2) - \hat{\delta}_1 \\ \ln(\hat{\mu}_1 / \hat{\mu}_2) - \hat{\delta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\delta})$

Estimating functions: Code

- SAS

```
pscore = 1/(1 + exp(-(theta[1] + theta[2]*bp)));
ef_1 = anemia - pscore;
ef_2 = (anemia - pscore)#bp;

wt = anemia/pscore + (1-anemia)/(1-pscore);
ef_r1 = anemia#wt#ptb - theta[3];
ef_r0 = (1-anemia)#wt#ptb - theta[4];

ef_rd = j(n,1,(theta[3] - theta[4]) - theta[5]);
ef_lnrr = j(n,1,log(theta[3]/theta[4]) - theta[6]);
```

Estimating functions: Code

- R

```
pscore <- plogis(alpha[1] + alpha[2]*bp)
ef_1 <- (anemia - pscore)
ef_2 <- (anemia - pscore)*bp

wt <- anemia/pscore + (1-anemia)/(1-pscore)
ef_r1 <- anemia*wt*ptb - mu[1]
ef_r0 <- (1 - anemia)*wt*ptb - mu[2]

ef_rd <- (mu[1] - mu[2]) - delta[1]
ef_lnrr <- log(mu[1]/mu[2]) - delta[2]
```

Estimating functions: Code

- Python

```
ee_logit = ee_regression(theta=alpha,
        y=d['X'],
        X=d[['intercept', 'W']],
        model='logistic')

pscore = inverse_logit(np.dot(d1[['intercept', 'W']], alpha))
wt = d['X']/pscore + (1-d['X'])/(1-pscore)

ee_r1 = d['X']*d['Y']*wt - mu1
ee_r0 = (1-d['X'])*d['Y']*wt - mu0

ee_rd = np.ones(d.shape[0])*((mu1 - mu0) - delta1)
ee_rr = np.ones(d.shape[0])*(np.log(mu1 / mu0) - delta2)
```

Time to work with code

Data Fusion

Data Fusion

"Fusion" study designs combine data from multiple sources[4]

Examples

- Using external validation data to address measurement error
- Transporting a parameter to an external target population
- Bridged treatment comparisons (combine trials)

[4] Cole et al. AJE 2012

Example: Misclassification

We want to estimate the point prevalence of HIV treatment for 950 HIV positive adults in MACS/WIHS cohort study ($R = 1$) in 1995[5]

$$\Pr(Y = 1 | R = 1)$$

In MACS/WIHS, HIV treatment is measured by self-report which may be inaccurate. Let W be self-reported treatment.

In a separate data source ($R = 0$), we have data on both self-reported treatment W and treatment documented in medical and pharmacy records (the gold-standard, Y) for 331 individuals.

$$O_i = (R_i, Y_i(1 - R_i), W_i)$$

[5] Cole et al. AJE 2010;171(1)

Data

Study cohort: 680 reported HIV treatment

$$\widehat{\Pr}(W = 1 | R = 1) = 0.716$$

Validation data

		Medical Records	
		+	-
Reported	+	204	18
	-	38	71
		242	89

Sensitivity: $\widehat{\Pr}(W = 1 | Y = 1) = 0.843$
Specificity: $\widehat{\Pr}(W = 0 | Y = 0) = 0.798$

Let $\text{Se} = \Pr(W = 1|Y = 1)$ (sensitivity)
Let $\text{Sp} = \Pr(W = 0|Y = 0)$ (specificity)

$$\Pr(Y = 1) = \frac{\Pr(W = 1) - (1 - \text{Sp})}{\text{Se} - (1 - \text{Sp})}$$

Addressing misclassification

We will apply this Rogan Gladen equation in our study sample

$$\text{Pr}(Y = 1 | R = 1) = \frac{\text{Pr}(W = 1 | R = 1) - (1 - \text{Sp})}{\text{Se} - (1 - \text{Sp})}$$

leveraging our validation sample to estimate Se and Sp

Addressing misclassification

Four steps

1. Estimate $\nu = \Pr(W = 1 | R = 1)$
2. Estimate $\gamma = \Pr(W = 1 | Y = 1, R = 0)$
3. Estimate $\eta = \Pr(W = 0 | Y = 0, R = 0)$
4. Estimate $\phi = \Pr(Y = 1 | R = 1) = \frac{\nu - (1 - \eta)}{\gamma - (1 - \eta)}$

Addressing misclassification

Our estimating equation is

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \begin{bmatrix} R_i(W_i - \hat{\nu}) \\ (1 - R_i)Y_i(W_i - \hat{\gamma}) \\ (1 - R_i)(1 - Y_i)((1 - W_i) - \hat{\eta}) \\ \hat{\phi}(\hat{\gamma} - (1 - \hat{\eta})) - (\hat{\nu} - (1 - \hat{\eta})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\nu, \gamma, \eta, \phi)$

Addressing misclassification

Our estimating equation is

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \begin{bmatrix} R_i(W_i - \hat{\nu}) \\ (1 - R_i)Y_i(W_i - \hat{\gamma}) \\ (1 - R_i)(1 - Y_i)((1 - W_i) - \hat{\eta}) \\ \hat{\phi}(\hat{\gamma} - (1 - \hat{\eta})) - (\hat{\nu} - (1 - \hat{\eta})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\nu, \gamma, \eta, \phi)$

Deriving our estimating functions

Step 1. $\nu = \text{Pr}(W = 1|R = 1) = \frac{\text{Pr}(W=1,R=1)}{\text{Pr}(R=1)}$

$$\hat{\nu} = \frac{\sum_{i=1}^{n} W_i R_i}{\sum_{i=1}^{n} R_i}$$

$$\hat{\nu} \sum_{i=1}^{n} R_i = \sum_{i=1}^{n} W_i R_i$$

$$0 = \sum_{i=1}^{n} W_i R_i - \sum_{i=1}^{n} \hat{\nu} R_i$$

$$0 = \sum_{i=1}^{n} \{W_i R_i - \hat{\nu} R_i\}$$

Therefore, our estimating function is $W_i R_i - \hat{\nu} R_i = R_i(W_i - \hat{\nu})$

Addressing misclassification

Our estimating equation is

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \begin{bmatrix} R_i(W_i - \hat{\nu}) \\ (1 - R_i)Y_i(W_i - \hat{\gamma}) \\ (1 - R_i)(1 - Y_i)((1 - W_i) - \hat{\eta}) \\ \hat{\phi}(\hat{\gamma} - (1 - \hat{\eta})) - (\hat{\nu} - (1 - \hat{\eta})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\nu, \gamma, \eta, \phi)$

Addressing misclassification

Our estimating equation is

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \begin{bmatrix} R_i(W_i - \hat{\nu}) \\ (1 - R_i)Y_i(W_i - \hat{\gamma}) \\ (1 - R_i)(1 - Y_i)((1 - W_i) - \hat{\eta}) \\ \hat{\phi}(\hat{\gamma} - (1 - \hat{\eta})) - (\hat{\nu} - (1 - \hat{\eta})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\nu, \gamma, \eta, \phi)$

Deriving our estimating functions

Step 2. $\gamma = \Pr(W = 1 | Y = 1, R = 0) = \frac{\Pr(W=1, Y=1, R=0)}{\Pr(Y=1, R=0)}$

$$\hat{\gamma} = \frac{\sum_{i=1}^{n} W_i(1 - R_i)Y_i}{\sum_{i=1}^{n}(1 - R_i)Y_i}$$

$$\hat{\gamma} \sum_{i=1}^{n}(1 - R_i)Y_i = \sum_{i=1}^{n} W_i(1 - R_i)Y_i$$

$$0 = \sum_{i=1}^{n} W_i(1 - R_i)Y_i - \sum_{i=1}^{n} \hat{\gamma}(1 - R_i)Y_i$$

$$0 = \sum_{i=1}^{n} \{W_i(1 - R_i)Y_i - \hat{\gamma}(1 - R_i)Y_i\}$$

Therefore, our estimating function is $(1 - R_i)Y_i(W_i - \hat{\gamma})$

Addressing misclassification

Our estimating equation is

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \begin{bmatrix} R_i(W_i - \hat{\nu}) \\ (1 - R_i)Y_i(W_i - \hat{\gamma}) \\ (1 - R_i)(1 - Y_i)((1 - W_i) - \hat{\eta}) \\ \hat{\phi}(\hat{\gamma} - (1 - \hat{\eta})) - (\hat{\nu} - (1 - \hat{\eta})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\nu, \gamma, \eta, \phi)$

Addressing misclassification

Our estimating equation is

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \begin{bmatrix} R_i(W_i - \hat{\nu}) \\ (1 - R_i)Y_i(W_i - \hat{\gamma}) \\ (1 - R_i)(1 - Y_i)((1 - W_i) - \hat{\eta}) \\ \hat{\phi}(\hat{\gamma} - (1 - \hat{\eta})) - (\hat{\nu} - (1 - \hat{\eta})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\nu, \gamma, \eta, \phi)$

Addressing misclassification

Our estimating equation is

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{O_i}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \begin{bmatrix} R_i(W_i - \hat{\nu}) \\ (1 - R_i)Y_i(W_i - \hat{\gamma}) \\ (1 - R_i)(1 - Y_i)((1 - W_i) - \hat{\eta}) \\ \hat{\phi}(\hat{\gamma} - (1 - \hat{\eta})) - (\hat{\nu} - (1 - \hat{\eta})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\boldsymbol{\theta} = (\nu, \gamma, \eta, \phi)$

Estimating functions: Code

- SAS

```
ef_1 = r#(w - theta[1]);
ef_2 = (1 - r)#y#(w - theta[2]);
ef_3 = (1 - r)#(1 - y)#((1 - w) - theta[3]);
ef_4 = j(n,1,
         (theta[1] - (1 - theta[3]))/(theta[2] - (1 - theta[3]))
                                             - theta[4]);
```

Estimating functions: Code

- R

```
ef_1 <- r*(w - nu)
ef_2 <- (1 - r)*y*(w - gamma)
ef_3 <- (1 - r)*(1 - y)*((1 - w) - eta)
ef_4 <- psi*(gamma - (1 - eta)) - (nu - (1 - eta))
```

Estimating functions: Code

- Python

```
ee_1 = r*(w - nu)
ee_2 = (1-r) * y * (w - gamma)
ee_3 = (1-r) * (1-y) * ((1-w) - eta)
ee_4 = np.ones(y.shape[0])*phi*(gamma + eta - 1) - (nu + eta - 1)
```

Time to work with code