The sandwich formula in R

M-estimators

- We are interested in a parameter (vector) θ
- Data: variables measured on n 'units' (individuals, families etc)
- We obtain an estimate $\hat{\theta}$ by solving an equation (system) on the form

$$\sum_{i=1}^{n} U_i(\theta) = 0$$

  where U is some function of θ and data, such that

$$U_i(\theta) \perp U_{i'}(\theta)$$

  and

$$E\{U(\theta)\} = 0$$

- $\hat{\theta}$ is called an 'M-estimator'

Asymptotic distribution of M-estimators

- $\hat{\theta}$ has an asymptotic normal distribution, with mean equal to the true value:

$$E(\hat{\theta}) = \theta$$

and variance given by the sandwich formula:

$$\mathrm{var}(\hat{\theta}) = \underbrace{E^{-1}\left\{\frac{\partial U(\theta)}{\partial \theta}\right\}}_{\text{Bread}} \underbrace{\mathrm{var}\{U(\theta)\}}_{\text{meat}} \underbrace{\left[E^{-1}\left\{\frac{\partial U(\theta)}{\partial \theta}\right\}\right]^T}_{\text{Bread}} / n$$

- If we can calculate $\mathrm{var}(\hat{\theta})$, then we can use the wald confidence interval

$$\hat{\theta} \pm 1.96\sqrt{\mathrm{var}(\hat{\theta})}$$

Calculation of the sandwich formula

$$\mathrm{var}(\hat{\theta}) = \underbrace{E^{-1}\left\{\frac{\partial U(\theta)}{\partial \theta}\right\}}_{\text{Bread}} \underbrace{\mathrm{var}\{U(\theta)\}}_{\text{meat}} \underbrace{\left[E^{-1}\left\{\frac{\partial U(\theta)}{\partial \theta}\right\}\right]^T}_{\text{Bread}} / n$$

- Problems:
  - The sandwich formula should be evaluated at the true θ
  - $E(\cdot)$ and $\mathrm{var}(\cdot)$ not known
- Solutions:
  - Replace θ by $\hat{\theta}$
  - Replace $E(\cdot)$ and $\mathrm{var}(\cdot)$ by sample counterparts

Example 1: GLM with canoncial link function

- Model: $E(Y|X;\theta)$ + fully parametric distribution
- $U(\theta)$ is the ML score function

$$
\begin{aligned}
U(\theta) &= \frac{\partial \log\{p(Y|X;\theta)\}}{\partial\theta} = X\{Y - E(Y|X;\theta)\} \\
E\{U(\theta)\} &= E[E\{U(\theta)|X\}] = E[X\{E(Y|X) - E(Y|X;\theta)\}] \\
&= 0 \\
E\left\{\frac{\partial U(\theta)}{\partial\theta}\right\} &= \mathrm{var}\{U(\theta)\}
\end{aligned}
$$

so

$$
\begin{aligned}
\mathrm{var}(\hat{\theta}) &= \underbrace{E^{-1}\left\{\frac{\partial U(\theta)}{\partial\theta}\right\}}_{\text{Bread}} \underbrace{\mathrm{var}\{U(\theta)\}}_{\text{meat}} \underbrace{\left[E^{-1}\left\{\frac{\partial U(\theta)}{\partial\theta}\right\}\right]^T}_{\text{Bread}} / n \\
&= \underbrace{E^{-1}\left\{\frac{\partial U(\theta)}{\partial\theta}\right\}}_{\text{Bread}} / n
\end{aligned}
$$

In R

```
> set.seed(8)
> n <- 1000
> x <- rnorm(n)
> y <- rnorm(n,mean=x)
> data <- data.frame(x,y)

> formula <- y~x
> fit <- lm(formula,data)
> diag(vcov(fit)) #Fisher info
 (Intercept)            x
0.0010234268 0.0009789521
```

In R, cont'd

```
> bread <- vcov(fit)*n
> U <- model.matrix(formula,data)*residuals(fit)
> meat <- var(U)
> diag(bread%*%meat%*%t(bread)/n) #sandwich
(Intercept)           x
0.001070403 0.001149813
```

Example 2: GEE with independent working correlation matrix

- Model: $E(Y|X;\theta)$
- $U(\theta)$ has same form as in GLM with canonical link function, but is not ML score function

$$U(\theta) = X\{Y - E(Y|X;\theta)\} \neq \frac{\partial \log\{p(Y|X;\theta)\}}{\partial\theta}$$

$$E\left\{\frac{\partial U(\theta)}{\partial\theta}\right\} \neq \mathrm{var}\{U(\theta)\}$$

so

$$\mathrm{var}(\hat{\theta}) = \underbrace{E^{-1}\left\{\frac{\partial U(\theta)}{\partial\theta}\right\}}_{\text{Bread}} \underbrace{\mathrm{var}\{U(\theta)\}}_{\text{meat}} \underbrace{\left[E^{-1}\left\{\frac{\partial U(\theta)}{\partial\theta}\right\}\right]^T}_{\text{Bread}} / n$$

$$\neq \underbrace{E^{-1}\left\{\frac{\partial U(\theta)}{\partial\theta}\right\}}_{\text{Bread}} / n$$

In R

```
> set.seed(8)
> n <- 1000
> x <- rnorm(n)
> y <- rnorm(n,mean=x,sd=abs(x)) #heteroscedastic
> data <- data.frame(x,y)

> formula <- y~x
> fit <- lm(formula,data)
> diag(vcov(fit)) #naive Fisher info; too small
(Intercept)            x
0.001202841 0.001150569
```

In R, cont'd

```
> bread <- vcov(fit)*n
> U <- model.matrix(formula,data)*residuals(fit)
> meat <- var(U)
> diag(bread%*%meat%*%t(bread)/n) #sandwich
(Intercept)           x
0.001734042 0.005096120
```

Example 3: GEE with clustered data and independent working correlation matrix

- Let $U_{ij}(\theta)$ be the estimating function for subject j in cluster i
- Define $U_i(\theta) = \sum_j U_{ij}(\theta)$
- If the clusters are independent, then

$$U_i(\theta) \perp U_{i'}(\theta)$$

- If $E\left\{U_{ij}(\theta)\right\} = 0$, then

$$\begin{aligned} E\{U_i(\theta)\} &= E\left\{\sum_j U_{ij}(\theta)\right\} \\ &= \sum_j E\left\{U_{ij}(\theta)\right\} \\ &= 0 \end{aligned}$$

In R

```
> set.seed(8)
> n <- 1000
> x <- rnorm(n)
> y <- rnorm(n,mean=x)
> id <- 1:n
> data <- data.frame(x,y,id)
> data <- data[rep(1:n,each=2),]

> formula <- y~x
> fit <- lm(formula,data)
> diag(vcov(fit)) #naive Fisher info; too small
 (Intercept)            x
0.0005112012 0.0004889861
```

In R, cont'd

```
> bread <- vcov(fit)*n
> U <- model.matrix(formula,data)*residuals(fit)
> U <- aggregate(U,by=list(data$id),FUN=sum)[,-1]
> meat <- var(U)
> diag(bread%*%meat%*%t(bread)/n) #sandwich
(Intercept)           x
0.001068261 0.001147513
```

Example 4: Regression standardization

- X = exposure, Z = confounders
- Nuisance parameter θ_1 defined by model: $E(Y|X,Z;\theta_1)$

$$U(\theta_1) = \begin{pmatrix} X \\ Z \end{pmatrix} \{Y - E(Y|X,Z;\theta_1)\}$$

- Target parameter $\theta_2 = E_Z\{E(Y|X=0,Z;\theta_1)\}$
- We estimate θ_2 by averaging over the sample distribution for Z:

$$\begin{aligned} \theta_2 &= \sum_{i=1}^{n} E(Y|X=0,Z_i;\hat{\theta}_1)/n \\ U(\theta_2,\theta_1) &= E(Y|X=0,Z;\theta_1) - \theta_2 \end{aligned}$$

Example 4: standardization, cont'd

$$
\begin{aligned}
U(\theta_1) &= \begin{pmatrix} X \\ Z \end{pmatrix} \{Y - E(Y|X, Z; \theta_1)\} \\
U(\theta_2, \theta_1) &= E(Y|X = 0, Z; \theta_1) - \theta_2 \\
\frac{\partial U(\theta)}{\partial \theta} &= \left\{ \begin{array}{cc} \frac{\partial U(\theta_1)}{\partial \theta_1} & \frac{\partial U(\theta_1)}{\partial \theta_2} \\ \frac{\partial U(\theta_2, \theta_1)}{\partial \theta_1} & \frac{\partial U(\theta_2, \theta_1)}{\partial \theta_2} \end{array} \right\} \\
\frac{\partial U(\theta_1)}{\partial \theta_1} & \quad \text{from inverse Fisher info} \\
\frac{\partial U(\theta_1)}{\partial \theta_2} &= 0 \\
\frac{\partial U(\theta_2, \theta_1)}{\partial \theta_1} &= -\frac{\partial}{\partial \theta_1} E(Y|X = x, Z; \theta_1) \\
\frac{\partial U(\theta_2, \theta_1)}{\partial \theta_2} &= -1
\end{aligned}
$$

In R

```
> set.seed(8)
> n <- 1000
> z <- rnorm(n)
> x <- rnorm(n,mean=z)
> y <- rnorm(n,mean=x+z)
> data <- data.frame(z,x,y)

> formula <- y~x+z
> fit <- lm(formula,data)
> data0 <- data
> data0$x <- 0
> pred <- predict(fit,newdata=data0)
> theta2 <- mean(pred)
```

In R, cont'd

```
> U1 <- model.matrix(formula,data)*residuals(fit)
> U2 <- pred-theta2
> U <- cbind(U1,U2)
> meat <- var(U)
> dU1.dtheta1 <- -solve(vcov(fit))/n
> dU1.dtheta2 <- rep(0,3)
> dU1.dtheta <- cbind(dU1.dtheta1,dU1.dtheta2)
> dU2.dtheta1 <- colMeans(model.matrix(formula,data
> dU2.dtheta2 <- -1
> dU2.dtheta <- c(dU2.dtheta1,dU2.dtheta2)
> dU.dtheta <- rbind(dU1.dtheta,dU2.dtheta)
> bread <- solve(dU.dtheta)
> diag(bread%*%meat%*%t(bread)/n) #sandwich
(Intercept)           x           z dU1.dtheta2
0.001048495 0.001030311 0.004023933 0.002170192
```

Other applications

- Delta method
- Weighting with estimated weights
- Doubly robust estimation
- Propensity scores
- Instrumental variables/Mendelian randomization
- **Most estimators are M-estimators**