BIOS 6312: Modern Regression Analysis

Andrew J. Spieker, Ph.D.

Assistant Professor of Biostatistics
Vanderbilt University Medical Center

Set 3 supplementary slides for `R` enthusiasts

Version: 01/25/2021

Examples for Set 3

Examples for R enthusiasts:

- Subgroup effects (Slide 296)
- Examples for three-way interactions (Slide 301)
- Weighted least squares for DSST and age (Slide 349)

Examples for Set 3

Examples for R enthusiasts:

- **Subgroup effects (Slide 296)**
- Examples for three-way interactions (Slide 301)
- Weighted least squares for DSST and age (Slide 349)

Subgroup effects in REACH

Reading in the REACH data:

- Read in data:

```
reach.data <- read.csv("reach.csv",
                       header = TRUE,
                       stringsAsFactors = FALSE)
```

Subgroup effects in REACH

Example: Subgroup effect with a continuous interaction term

- Model: $\mathrm{E}[Y|X = x, Z = z] = \beta_0 + \beta_1 x + \beta_2 z + \beta_3 xz$.
  - X: REACH (0 = control; 1 = REACH).
  - Z: baseline A1c.
  - Y: six-month A1c.
- Goal: learn about REACH effect among subgroup with $Z = z_0$.

Subgroup effects in REACH

Regression fit:

- Fit the regression model and extract sandwich variance.

```
library("sandwich")
regr.a1c <- lm(a1c.6 ~ reach * a1c.0,
               data = reach.data)
robust.var <- vcovHC(regr.a1c, type = "HC1")
```

- R knows to include lower-order interaction term.

Subgroup effects in REACH

Regression fit: Output

```
> summary(regr.a1c)

Call:
lm(formula = a1c.6 ~ reach * a1c.0, data = reach.data)

Residuals:
   Min     1Q Median     3Q    Max
-5.125 -1.108 -0.170  0.719  9.455

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.4990     0.5434    6.44  3.2e-10 ***
reach         0.7069     0.7729    0.91     0.36
a1c.0         0.6058     0.0615    9.85  < 2e-16 ***
reach:a1c.0  -0.1638     0.0869   -1.89     0.06 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.73 on 431 degrees of freedom
  (70 observations deleted due to missingness)
Multiple R-squared:  0.276,Adjusted R-squared:  0.271
F-statistic: 54.8 on 3 and 431 DF,  p-value: <2e-16
```

Subgroup effects in REACH

Regression fit: Robust variance

- Robust variance matrix is a 4×4 matrix.

```
> robust.var
             (Intercept)   reach    a1c.0 reach:a1c.0
(Intercept)       0.4396 -0.4396 -0.05116     0.05116
reach            -0.4396  0.7552  0.05116    -0.08620
a1c.0            -0.0512  0.0512  0.00618    -0.00618
reach:a1c.0       0.0512 -0.0862 -0.00618     0.01021
```

Subgroup effects in REACH

Example: Subgroup effect with a continuous interaction term

- Model: $\mathrm{E}[Y|X = x, Z = z] = \beta_0 + \beta_1 x + \beta_2 z + \beta_3 xz$.
  - X: REACH (0 = control; 1 = REACH).
  - Z: baseline A1c.
  - Y: six-month A1c.
- How do I learn about the REACH effect among the subgroup with $Z = z_0$?
  - $\mathrm{E}[Y|X = x + 1, Z = z_0] - \mathrm{E}[Y|X = x, Z = z_0] = \beta_1 + \beta_3 z_0$.

Subgroup effects in REACH

Linear combinations: Extra work in R

- To the best of my knowledge, R does not have a generalizable analog to Stata's `lincom`.
- To save you the agony, I created one for use in R:

```
lincom.R <- function(par, mults, coefs, vcov, N, alpha = 0.05) {
  R <- matrix(0, nrow = 1, ncol = length(coefs))
  for (q in 1:length(par)) {R[1,par[q]] <- mults[q]}
  w <- sqrt(as.numeric(t(R %*% coefs) %*%
                   solve(R %*% vcov %*% t(R)) %*%
                   (R %*% coefs)))
  p <- 2*(1 - pt(w, df = N - length(coefs)))
  Est <- R %*% coefs
  tol <- qt(1 - alpha/2, df = N - length(coefs))
  CI.Lo <- R %*% coefs - tol*sqrt(R %*% vcov %*% t(R))
  CI.Hi <- R %*% coefs + tol*sqrt(R %*% vcov %*% t(R))
  return(c(EST = Est, CI.LO = CI.Lo, CI.HI = CI.Hi, P = p))
}
```

- Won't catch mistakes, but will work when used correctly.
- Mimics Stata's *t*-statistic formulation.

Subgroup effects in REACH

Parameters for R function: `lincom.R`

- `par`: indices of parameters to combine.
- `mults`: multiples of those parameters noted by `par`.
- `coefs`: vector of model coefficients.
- `vcov`: variance-covariance matrix.
- `N`: number of observations used in analysis.
- `alpha`: confidence level (0.05 by default).

Subgroup effects in REACH

Linear combinations:

- Subgroup effect among those with baseline A1c of 7.5%.

```
lincom.R(par = c(2,4),
         mults = c(1,7.5),
         coefs = regr.a1c$coefficients,
         vcov = robust.var,
         N = dim(regr.a1c$model)[1])

EST    CI.LO    CI.HI         P
-0.52160 -0.89603 -0.14717  0.00322
```

- If you want to test $\beta_1 + 7.5\beta_3$, then the *indices* are 2 and 4 (not 1 and 3). The multiples are 1 and 7.5.
- Agrees with Stata output (Slide 298).

Examples for Set 3

Examples for R enthusiasts:

- *Subgroup effects (Slide 296)*
- **Examples for three-way interactions (Slide 301)**
- Weighted least squares for DSST and age (Slide 349)

Three-way interactions in REACH

Reminder of setup:

- This example also makes use of the REACH data.
- Allow interaction by REACH, gender, baseline A1c.
  - X: REACH.
  - Z: gender.
  - W: baseline A1c.
  - Y: six-month A1c.

$$\begin{aligned} \mathrm{E}[Y|X=x,Z=z,W=w] &= \beta_0+\beta_1 x+\beta_2 z+\beta_3 w \\ &\quad +\beta_4 xz+\beta_5 xw+\beta_6 wz+\beta_7 xzw \end{aligned}$$

Three-way interactions in REACH

Regression fit:

- Fit the regression model and extract sandwich variance.

```
library("sandwich")
regr.a1c <- lm(a1c.6 ~ reach * gender * a1c.0,
               data = reach.data)
robust.var <- vcovHC(regr.a1c, type = "HC1")
```

- R knows to include all lower-order interaction terms.

Three-way interactions in REACH

Regression fit: Output

```
> summary(regr.a1c)

Call:
lm(formula = a1c.6 ~ reach * gender * a1c.0, data = reach.data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.6703 -1.0592 -0.1864  0.7220  9.3823

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         2.70617    0.79717   3.395 0.000751 ***
reach               1.89309    1.18664   1.595 0.111376
gender              1.47667    1.09984   1.343 0.180110
a1c.0               0.69993    0.09327   7.504 3.63e-13 ***
reach:gender       -2.11828    1.57526  -1.345 0.179430
reach:a1c.0        -0.29522    0.13492  -2.188 0.029198 *
gender:a1c.0       -0.17057    0.12521  -1.362 0.173838
reach:gender:a1c.0  0.23074    0.17739   1.301 0.194044
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.733 on 427 degrees of freedom
  (70 observations deleted due to missingness)
Multiple R-squared:  0.2799,Adjusted R-squared:  0.2681
F-statistic: 23.71 on 7 and 427 DF,  p-value: < 2.2e-16
```

Three-way interactions in REACH

Regression fit: Robust variance

- Robust variance matrix is an 8×8 matrix (too big to report, but I'll show you the first five rows and columns below).

```
                  (Intercept)   reach  gender   a1c.0 reach:gender   . . .
(Intercept)            0.8172 -0.8172 -0.8172 -0.0959       0.8172
reach                 -0.8172  1.3806  0.8172  0.0959      -1.3806
gender                -0.8172  0.8172  1.6166  0.0959      -1.6166
a1c.0                 -0.0959  0.0959  0.0959  0.0117      -0.0959
reach:gender           0.8172 -1.3806 -1.6166 -0.0959       2.8024
                         .
                         .
                         .
```

Three-way interactions in REACH

Joint testing: Extra work in R

- To the best of my knowledge, R does not have a generalizable analog to `testparm`. I created one for your convenience:

```
testparm.R <- function(par, coefs, vcov, N = NULL, type = "F") {
  R <- matrix(0, nrow = length(par), ncol = length(coefs))
  for (q in 1:length(par)) {R[q,unlist(par[q])] <- 1}
  if (type == "F") {
    if (is.null(N)) {stop("Please provide a value for N")}
    f <- as.numeric(t(R %*% coefs) %*%
                      solve(R %*% vcov %*% t(R)) %*%
                      (R %*% coefs)/(length(par)))
    p <- 1 - pf(f, df1 = length(par),
                df2 = N - (length(coefs)))
    return(c(F = f, P = p)) }
  if (type == "W") {
    w <- as.numeric(t(R %*% coefs) %*%
                      solve(R %*% vcov %*% t(R)) %*%
                      (R %*% coefs))
    p <- 1 - pchisq(w, df = length(par))
    return(c(W = w, P = p))}
}
```

Three-way interactions in REACH

Parameters for R function: `testparm.R`

- `par`: list of combinations of parameters for joint test.
- `coefs`: vector of model coefficients.
- `vcov`: variance-covariance matrix.
- `N`: number of observations used in analysis.
- `type`: either "F" for F-test or "W" for Wald test.

Three-way interactions in REACH

Example: Testing overall effect of REACH

- $H_0 : \beta_1 = \beta_4 = \beta_5 = \beta_7 = 0$.

```
testparm.R(par = list(2,5,6,8),
           coefs = regr.a1c$coefficients,
           vcov = robust.var,
           N = dim(regr.a1c$model)[1])

## Output
        F         P
5.397209 0.000301
```

- Agrees with Stata output (Slide 304).

Three-way interactions in REACH

Example: Testing effect of REACH among females

- $H_0 : \beta_1 = \beta_5 = 0$.

```
testparm.R(par = list(2,6),
           coefs = regr.a1c$coefficients,
           vcov = robust.var,
           N = dim(regr.a1c$model)[1])

## Output
      F       P
5.25470 0.00557
```

- Agrees with Stata output (Slide 306).

Three-way interactions in REACH

Example: Testing effect of REACH among males

- $H_0 : \beta_1 + \beta_4 = \beta_5 + \beta_7 = 0$.

```
testparm.R(par = list(c(2,5), c(6,8)),
           coefs = regr.a1c$coefficients,
           vcov = robust.var,
           N = dim(regr.a1c$model)[1])

## Output
      F       P
5.53972 0.00422
```

- Agrees with Stata output (Slide 308/309).

Three-way interactions in REACH

Example: Testing interaction between baseline A1c and REACH

- $H_0 : \beta_5 = \beta_7 = 0$.

```
testparm.R(par = list(6,8),
           coefs = regr.a1c$coefficients,
           vcov = robust.var,
           N = dim(regr.a1c$model)[1])

## Output
     F      P
2.3293 0.0986
```

- Agrees with Stata output (Slide 311).

Examples for Set 3

Examples for R enthusiasts:

- *Subgroup effects (Slide 296)*
- *Examples for three-way interactions (Slide 301)*
- **Weighted least squares for DSST and age (Slide 349)**

Weighted least squares in MRI

Example: DSST and age

- X: age.
- Y: DSST.
- Model: $\mathrm{E}[Y|X = x] = \beta_0 + \beta_1 x$.
- Consider unweighted model, and model weighting inversely to age (as an example).

Weighted least squares in MRI

Reading in the MRI data:

- Read in data:

```
mri.data <- read.csv("mri.csv",
                     header = TRUE,
                     stringsAsFactors = FALSE)
```

Weighted least squares in MRI

Generating weights:

- Create and attach weights for weighted model:

```
mri.data$wts <- mri.data$age
```

Weighted least squares in MRI

Model fitting:

- Fit unweighted and weighted regression models.

```
model.u <- lm(dsst ~ age,
              data = mri.data)
model.w <- lm(dsst ~ age, weights = wts,
              data = mri.data)
```

- Make note of option `weights`.

Weighted least squares in MRI

Results:

- Unweighted model: ordinary standard errors.

```
> summary(model.u)

Call:
lm(formula = dsst ~ age, data = mri.data)

Residuals:
   Min     1Q Median     3Q    Max
-41.45  -7.61  -0.14   7.55  44.00

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 105.3395     6.1570    17.1   <2e-16 ***
age          -0.8633     0.0825   -10.5   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.8 on 721 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.132,Adjusted R-squared:  0.131
F-statistic:  110 on 1 and 721 DF,  p-value: <2e-16
```

- Agrees with Stata output (Slide 352).

Weighted least squares in MRI

Results:

- Weighted model: ordinary standard errors.

```
> summary(model.w)

Call:
lm(formula = dsst ~ age, data = mri.data, weights = wts)

Weighted Residuals:
   Min     1Q Median     3Q    Max
-356.7  -64.6   -0.8   65.5  388.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 106.2011     5.9753    17.8   <2e-16 ***
age          -0.8748     0.0796   -11.0   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 102 on 721 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.143,Adjusted R-squared:  0.142
F-statistic:  121 on 1 and 721 DF,  p-value: <2e-16
```

- Agrees with Stata output (Slide 353).

Weighted least squares in MRI

Results:

- Unweighted model: sandwich standard errors.

```
> sqrt(diag(vcovHC(model.u, type = "HC1")))
(Intercept)         age
    5.70871     0.07551
```

- Agrees with Stata output (Slide 355).

Weighted least squares in MRI

Results:

- Weighted model: sandwich standard errors.

```
> sqrt(diag(vcovHC(model.w, type = "HC1")))
(Intercept)         age
    5.61944     0.07428
```

- Agrees with Stata output (Slide 356).