Session 3: Regression coefficients and model matrices

Levi Waldron

CUNY SPH Biostatistics 2

Learning objectives and outline

Learning objectives

1. Interpret main coefficients in logistic regression
2. Interpret interaction terms in logistic regression
3. Define and interpret model matrices for (generalized) linear models

Outline

1. Review of GLM
2. Interpretation of logistic regression coefficients
3. Introduction to model matrices

GLM review

Components of GLM

- **Random component** specifies the conditional distribution for the response variable
  - doesn't have to be normal
  - can be any distribution in the "exponential" family of distributions
- **Systematic component** specifies linear function of predictors (linear predictor)
- **Link** [denoted by g(.)] specifies the relationship between the expected value of the random component and the systematic component
  - can be linear or nonlinear

Logistic Regression as GLM

- **The model**:

$$Logit(P(x)) = log\left(\frac{P(x)}{1-P(x)}\right) = \beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}+...+\beta_p x_{pi}$$

- **Random component**: y_i follows a Binomial distribution (outcome is a binary variable)

- **Systematic component**: linear predictor

$$\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_p x_{pi}$$

- **Link function**: *logit* (log of the odds that the event occurs)

$$g(P(x)) = logit(P(x)) = log\left(\frac{P(x)}{1-P(x)}\right)$$

$$P(x) = g^{-1}(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_p x_{pi})$$

Additive vs. Multiplicative models

- Linear regression is an *additive* model
  - *e.g.* for two binary variables $\beta_1 = 1.5$, $\beta_2 = 1.5$.
  - If $x_1 = 1$ and $x_2 = 1$, this adds 3.0 to $E(y|x)$
- Logistic regression is a *multiplicative* model
  - If $x_1 = 1$ and $x_2 = 1$, this adds 3.0 to $log(\frac{P}{1-P})$
  - Odds-ratio $\frac{P}{1-P}$ increases 20-fold: $exp(1.5 + 1.5)$ or $exp(1.5) * exp(1.5)$

Motivating example: contraceptive use data

From http://data.princeton.edu/wws509/datasets/#cuse

```
##       age                education          wantsMore
##  Length:16          Length:16          Length:16
##  Class :character   Class :character   Class :char
##  Mode  :character   Mode  :character   Mode  :char
##
##
##
##       using
##  Min.   : 4.00
##  1st Qu.: 9.50
##  Median :29.00
##  Mean   :31.69
##  3rd Qu.:49.00
##  Max.   :80.00
```

Motivating example: contraceptive use data

Univariate regression to "wants more children" only:

```
fit <- glm(cbind(using, notUsing) ~ wantsMore,
           data=cuse, family=binomial("logit"))
```

Estimate

Std. Error

z value

Pr(>|z|)

(Intercept)

-0.1864

0.0797

-2.34

0.0194

wantsMoreyes

-1.0486

Interpretation of coefficients

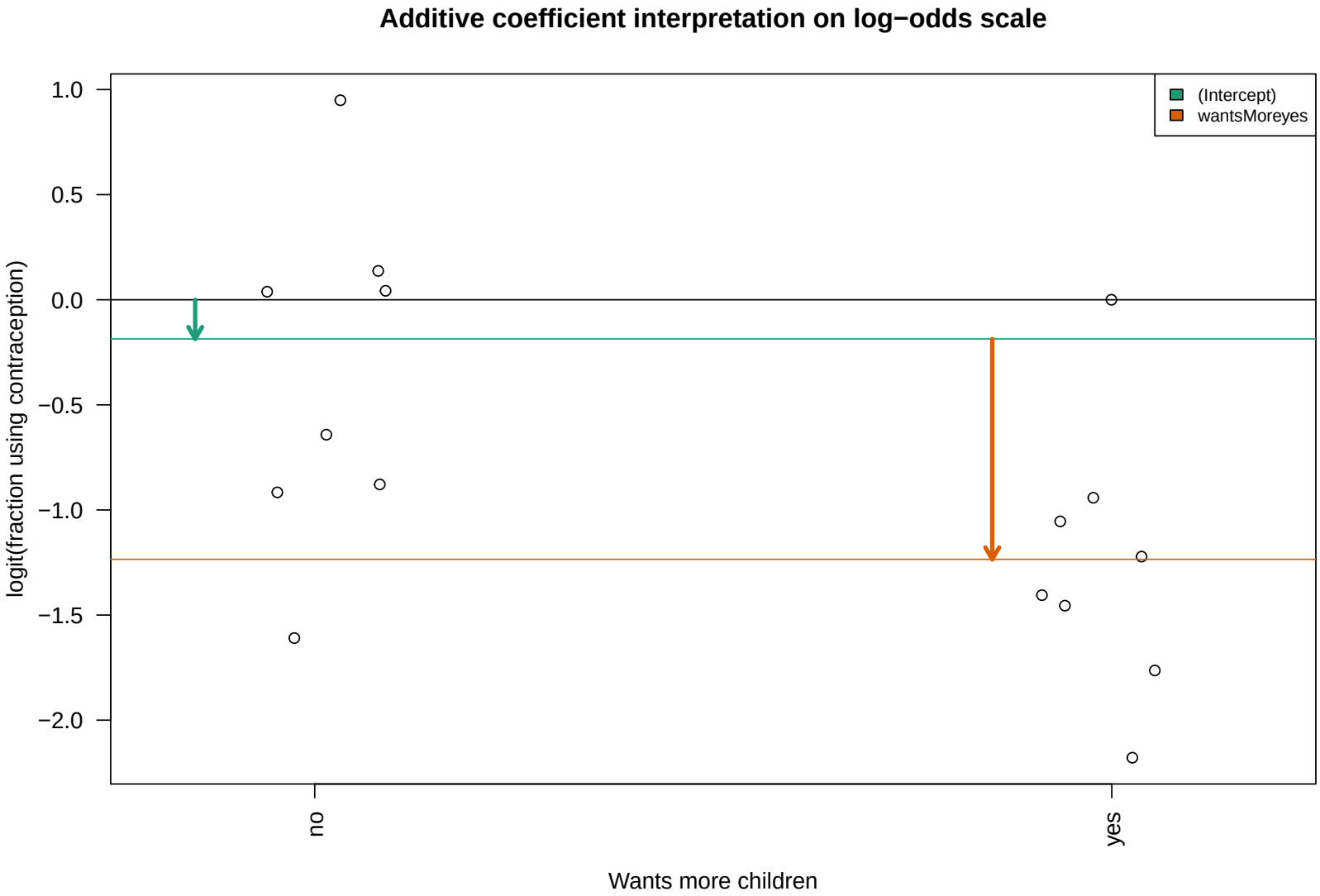

Figure 1: Diagram of the estimated coefficients in the GLM. The green arrow indicates the Intercept term, which goes from zero to the

Regression on age

There are four age groups:

```
fit <- glm(cbind(using, notUsing) ~ age,
           data=cuse, family=binomial("logit"))
```

Estimate

Std. Error

z value

Pr(>|z|)

(Intercept)

-1.5072

0.1303

-11.57

Regression with multiple predictors - model formulae:

symbol	example	meaning
+	+ x	include this variable
-	- x	delete this variable
:	x : z	include the interaction
*	x * z	include these variables and their interactions
^	(u + v + w)^3	include these variables and all interactions up to three way
1	-1	intercept: delete the intercept

Regression on age and wantsMore

```
fit <- glm(cbind(using, notUsing) ~ age + wantsMore,
           data=cuse, family=binomial("logit"))
```

Estimate

Std. Error

z value

Pr(>|z|)

(Intercept)

-0.8698

0.1571

-5.54

Interaction / Effect Modification

- What if we want to know whether the effect of age is modified by whether the woman wants more children or not?

Interaction is modeled as the product of two covariates:

$$E[y|x] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 * x_2$$

Interaction / Effect Modification (cont'd)

```
fit <- glm(cbind(using, notUsing) ~ age * wantsMore,
           data=cuse, family=binomial("logit"))
```

Estimate

Std. Error

z value

Pr(>|z|)

(Intercept)

-1.4553

0.2968

-4.90

The Design Matrix

What is the design matrix, and why?

- **What?** The design matrix is the most generic, flexible way to specify them
- **Why?** There are multiple possible and reasonable regression models for a given study design.

Matrix notation for the multiple linear regression model

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

or simply:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

- The design matrix is $\mathbf{X}$
  - which the computer will take as a given when solving for β by minimizing the sum of squares of residuals ε, or maximizing likelihood.

Choice of design matrix

- The model formula encodes a default model matrix, e.g.:

```
group <- factor( c(1, 1, 2, 2) )
model.matrix(~ group)
```

```
##   (Intercept) group2
## 1           1      0
## 2           1      0
## 3           1      1
## 4           1      1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

Choice of design matrix (cont'd)

What if we forgot to code group as a factor?

```
group <- c(1, 1, 2, 2)
model.matrix(~ group)
```

```
##   (Intercept) group
## 1           1     1
## 2           1     1
## 3           1     2
## 4           1     2
## attr(,"assign")
## [1] 0 1
```

More groups, still one variable

```
group <- factor(c(1,1,2,2,3,3))
model.matrix(~ group)
```

```
##   (Intercept) group2 group3
## 1           1      0      0
## 2           1      0      0
## 3           1      1      0
## 4           1      1      0
## 5           1      0      1
## 6           1      0      1
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

Changing the baseline group

```
group <- factor(c(1,1,2,2,3,3))
group <- relevel(x=group, ref=3)
model.matrix(~ group)
```

```
##   (Intercept) group1 group2
## 1           1      1      0
## 2           1      1      0
## 3           1      0      1
## 4           1      0      1
## 5           1      0      0
## 6           1      0      0
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

More than one variable

```
agegroup <- factor(c(1,1,1,1,2,2,2,2))
wantsMore <- factor(c("y","y","n","n","y","y","n","n"
model.matrix(~ agegroup + wantsMore)
```

```
##   (Intercept) agegroup2 wantsMorey
## 1           1         0          1
## 2           1         0          1
## 3           1         0          0
## 4           1         0          0
## 5           1         1          1
## 6           1         1          1
## 7           1         1          0
## 8           1         1          0
## attr(,"assign")
## [1] 0 1 2
## attr(,"contrasts")
## attr(,"contrasts")$agegroup
## [1] "contr.treatment"
```

With an interaction term

```
model.matrix(~ agegroup + wantsMore + agegroup:wantsM
```

```
##   (Intercept) agegroup2 wantsMorey agegroup2:wants
## 1           1         0          1
## 2           1         0          1
## 3           1         0          0
## 4           1         0          0
## 5           1         1          1
## 6           1         1          1
## 7           1         1          0
## 8           1         1          0
## attr(,"assign")
## [1] 0 1 2 3
## attr(,"contrasts")
## attr(,"contrasts")$agegroup
## [1] "contr.treatment"
##
## attr(,"contrasts")$wantsMore
```

Design matrix to contrast what we want

- Contraceptive use example
  - Is the effect of wanting more children different for 40-49 year-olds than for <25 year-olds is answered by the term `age40-49:wantsMoreyes` in a model with interaction terms:

```
fitX <- glm(cbind(using, notUsing) ~ age * wantsMore,
            data=cuse, family=binomial("logit"))
summary(fitX)
```

```
##
## Call:
## glm(formula = cbind(using, notUsing) ~ age * wants
##     data = cuse)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
```

Design matrix to contrast what we want (cont'd)

- What if we want to ask this question for 40-49 year-olds vs. 30-39 year-olds?

The desired contrast is:

`age40-49:wantsMoreyes - age30-39:wantsMoreyes`

There are many ways to construct this design, one is with `library(multcomp)`:

```
names(coef(fitX))
```

```
## [1] "(Intercept)"                "age25-29"
## [4] "age40-49"                   "wantsMoreyes"
## [7] "age30-39:wantsMoreyes" "age40-49:wantsMoreyes
```

```
contmat <- matrix(c(0,0,0,0,0,0,-1,1), 1)
new.interaction <- multcomp::glht(fitX, linfct=contma
summary(new.interaction)
```