BIOS621 Session 3

Levi Waldron

Learning objectives - session 3

- fit and interpret interaction terms
- define and interpret model matrices for (generalized) linear models

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**:

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

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

- **Systematic component**: linear predictor

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

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

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

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

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    notUsing          using      
##  <25  :4   high:8    no :8     Min.   :  8.00   Min.   : 4.00  
##  25-29:4   low :8    yes:8     1st Qu.: 31.00   1st Qu.: 9.50  
##  30-39:4                       Median : 56.50   Median :29.00  
##  40-49:4                       Mean   : 68.75   Mean   :31.69  
##                                3rd Qu.: 85.75   3rd Qu.:49.00  
##                                Max.   :212.00   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	0.1107	-9.48	0.0000

- Interpretation of this table:
  - *Coefficients for **(Intercept)** and **dummy variables***
  - *Coefficients are normally distributed when assumptions are correct*

Interpretation of coefficients

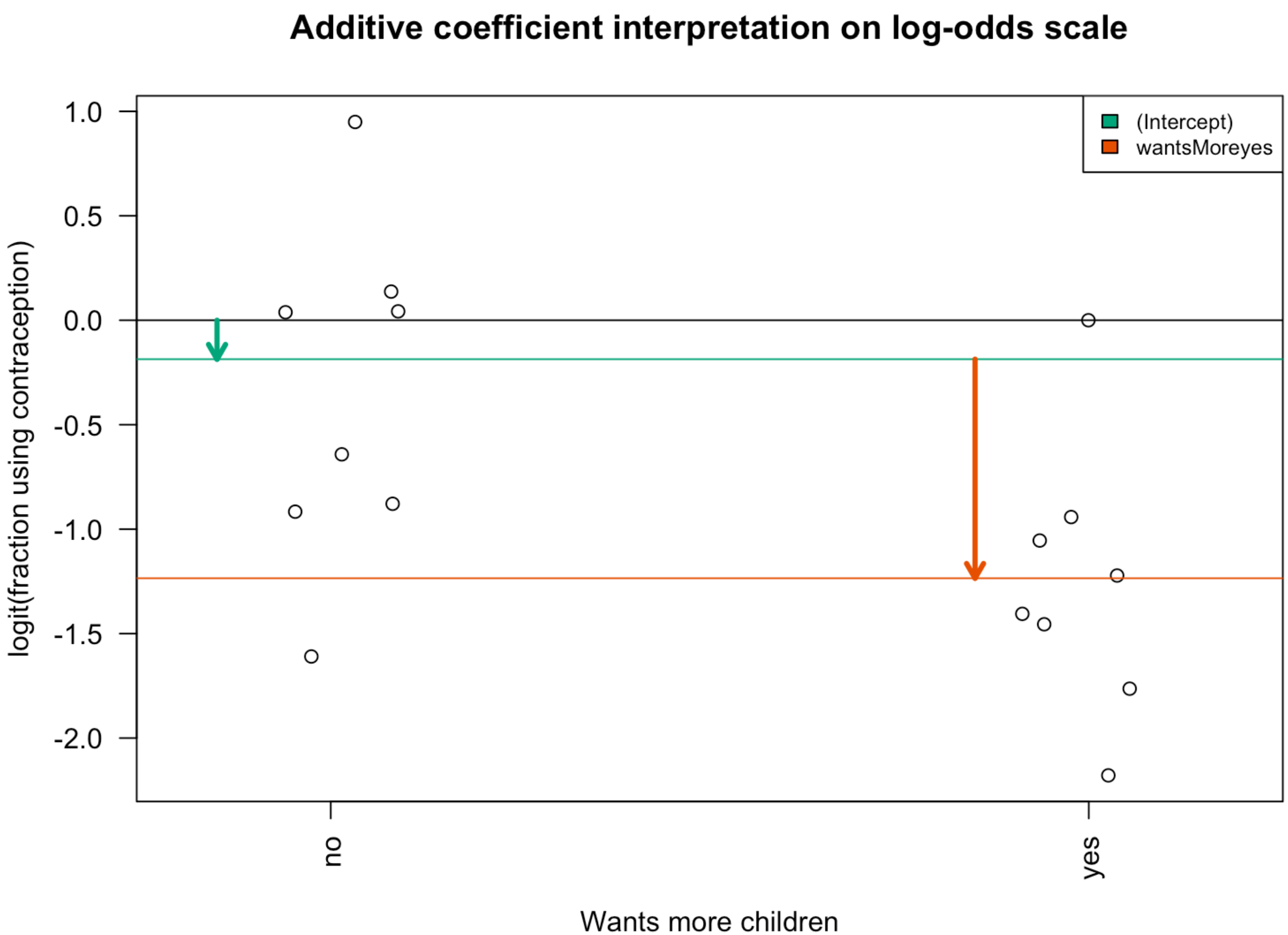

Diagram of the estimated coefficients in the GLM. The green arrow indicates the Intercept term, which goes from zero to the mean of the reference group (here the ‘pull’ samples). The orange arrow indicates the difference between the push group and the pull group, which is negative in this example. The circles show the individual samples, jittered horizontally to avoid overplotting.

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	0.0000
age25-29	0.4607	0.1727	2.67	0.0077
age30-39	1.0483	0.1544	6.79	0.0000
age40-49	1.4246	0.1940	7.35	0.0000

- Interpretation of the dummy variables `age25-29`, `age30-39`, `age40-49`

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	0.0000
age25-29	0.3678	0.1754	2.10	0.0360
age30-39	0.8078	0.1598	5.06	0.0000
age40-49	1.0226	0.2039	5.01	0.0000
wantsMoreyes	-0.8241	0.1171	-7.04	0.0000

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	0.0000
age25-29	0.6354	0.3564	1.78	0.0746
age30-39	1.5411	0.3183	4.84	0.0000
age40-49	1.7643	0.3435	5.14	0.0000
wantsMoreyes	-0.0640	0.3303	-0.19	0.8464
age25-29:wantsMoreyes	-0.2672	0.4091	-0.65	0.5137
age30-39:wantsMoreyes	-1.0905	0.3733	-2.92	0.0035
age40-49:wantsMoreyes	-1.3671	0.4834	-2.83	0.0047

The Design Matrix

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

The Design Matrix

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 $\boldsymbol{\beta}$ by minimizing the sum of squares of residuals $\boldsymbol{\varepsilon}$, 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

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

```r
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

```r
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")$agegroup
## [1] "contr.treatment"
##
## attr(,"contrasts")$wantsMore
## [1] "contr.treatment"
```

With an interaction term

```r
model.matrix(~ agegroup + wantsMore + agegroup:wantsMore)
```

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

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"))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.4553	0.2968	-4.90	0.0000
age25-29	0.6354	0.3564	1.78	0.0746
age30-39	1.5411	0.3183	4.84	0.0000
age40-49	1.7643	0.3435	5.14	0.0000
wantsMoreyes	-0.0640	0.3303	-0.19	0.8464
age25-29:wantsMoreyes	-0.2672	0.4091	-0.65	0.5137
age30-39:wantsMoreyes	-1.0905	0.3733	-2.92	0.0035

age40-49:wantsMoreyes	-1.3671	0.4834	-2.83	0.0047

Design matrix to contrast what we want

- 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"               "age30-39"
## [4] "age40-49"               "wantsMoreyes"           "age25-29: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=contmat)
summary(new.interaction)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
```

```
##
## Fit: glm(formula = cbind(using, notUsing) ~ age * wantsMore, family = binomial("logit"),
##     data = cuse)
##
## Linear Hypotheses:
##        Estimate Std. Error z value Pr(>|z|)
## 1 == 0  -0.2767     0.3935  -0.703    0.482
## (Adjusted p values reported -- single-step method)
```

Lab Exercises

1. What is the mean fraction of women using birth control for each age group? Each education level? For women who do or don't want more children?

    - *Hint: look at the "data wrangling" cheat sheet functions* `mutate`, `group_by`, *and* `summarize`

2. Create a fit to the birth control data using all predictors, called `fit1`. Based on `fit1`, write on paper the model for expected probability of using birth control. Don't forget the logit function.

3. Based on `fit1`, what is the expected probability of an individual 25-29 years old, with high education, who wants more children, using birth control? Calculate it manually, and using `predict(fit1)`

4. Based on `fit1`: Relative to women under 25 who want to have children, what is the predicted increase in odds that a woman 40-49 years old who does *not* want to have children will be taking birth control?

5. Using a likelihood ratio test, is there evidence that a model with interactions improves on `fit1` (no interactions)?
6. Which, if any, variables have the strongest interactions?
7. Create a contrast matrix for a fit on age only, with contrasts between *every pair* of age groups. Between which age groups is the contrast significant?