

Logistic Regression with Multiple Predictors

Andrew Zieffler
Department of Educational Psychology

Model Formulae in R

Model type	Formula	Notes
Single term	$Y \sim X$	Intercept is implicitly included
	$Y \sim 1 + X$	Intercept is explicitly included
	$Y \sim X - 1$	Force intercept to be 0
Additive	$Y \sim X + Z$	
Interaction (Multiplicative)	$Y \sim X + Z + X:Z$	
	$Y \sim X * Z$	
	$Y \sim X + I(X^2)$	Quadratic model

The model notation used in R is based on the notation introduced by Wilkinson and Rogers (1973).

```
# Fit additive model  
> glm.c = glm(fracture ~ age + momfrac, data = glow, family = binomial(link = "logit"))
```

Coefficients:

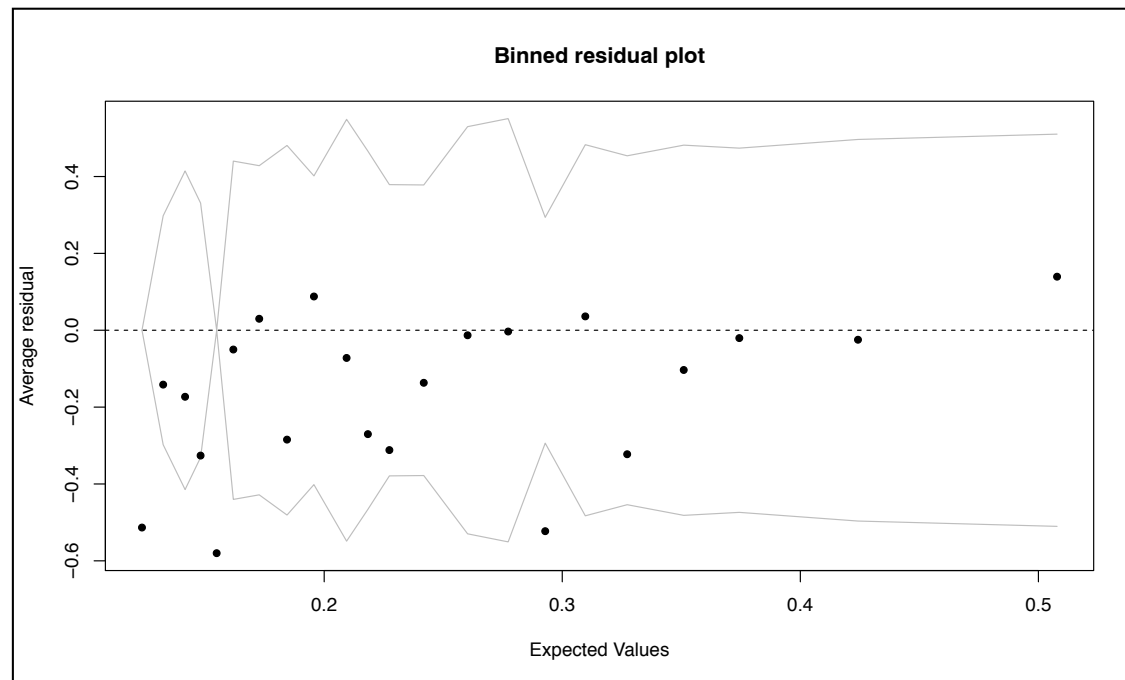
	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-4.85094	0.83383	-5.818	5.97e-09	***
age	0.05256	0.01170	4.492	7.04e-06	***
momfrac	0.64250	0.28848	2.227	0.0259	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 562.34 on 499 degrees of freedom
Residual deviance: 536.30 on 497 degrees of freedom
AIC: 542.3

Number of Fisher Scoring iterations: 4



Binned residual plot
doesn't look good, but
let's ignore that for a
minute

Obtain the LRT tests rather than the Wald tests.

```
# LRT for coefficients
> drop1(glm.c, test = "LRT")

Single term deletions

Model:
fracture ~ age + momfrac
      Df Deviance   AIC    LRT  Pr(>Chi)
<none>     536.30 542.30
age      1    557.07 561.07 20.7617 5.201e-06 ***
momfrac  1    541.06 545.06  4.7577  0.02917 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Age is a statistically significant predictor of fracture, controlling for momfrac.

momfrac is a statistically significant predictor of fracture, controlling for age.

```
# Odds
> exp(coef(glm.c))

(Intercept)      age      momfrac
0.007821009 1.053969181 1.901221667
```

The odds of a fracture increase by a factor of 1.05, on average, for each additional year of age, controlling for momfrac.

The odds of a fracture increase by a factor of 1.90, on average, if a subject's mother had a hip fracture, controlling for age.

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1(X_1) + \hat{\beta}_2(X_2)$$

$$X_2 \in \{0, 1\}$$



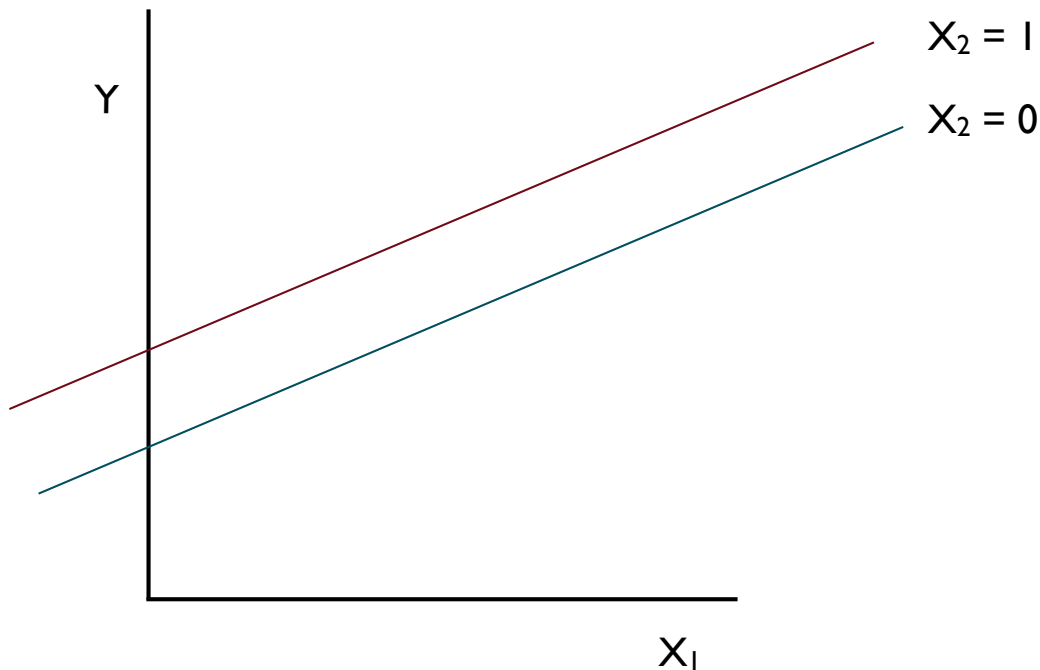
$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1(X_1) + \hat{\beta}_2(0)$$

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1(X_1)$$

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1(X_1) + \hat{\beta}_2(1)$$

$$\hat{Y} = (\hat{\beta}_0 + \hat{\beta}_2) + \hat{\beta}_1(X_1)$$

The β_2 coefficient measures the difference in intercept between the two groups when X_2 is dummy coded



Since the two lines are parallel (same slope) the difference in intercept measures the difference between the two groups for any value of X_1 . It is the *controlled* difference between the groups.

This implies that the effect of X_2 is exactly the same for any value of X_1 .

Since the two lines are parallel (same slope) the the effect of X_1 is exactly the same for the two groups (X_2).

$$\hat{\text{fracture}} = -4.85 + 0.05(\text{age}) + 0.64(\text{momfrac})$$

For $\text{momfrac} = 0$ $\hat{\text{fracture}} = -4.85 + 0.05(\text{age})$

For $\text{momfrac} = 1$ $\hat{\text{fracture}} = -4.21 + 0.05(\text{age})$

```
# Create a data frame of X and the predicted logits for each group
> new = data.frame(
  age = 55:90
)

> new$logit0 = -4.85 + 0.05 * new$age
> new$logit1 = -4.21 + 0.05 * new$age

# Plot the two models
> ggplot(data = new, aes(x = age, y = logit0)) +
  geom_line() +
  geom_line(aes(x = age, y = logit1), lty = "dashed") +
  theme_bw()
```

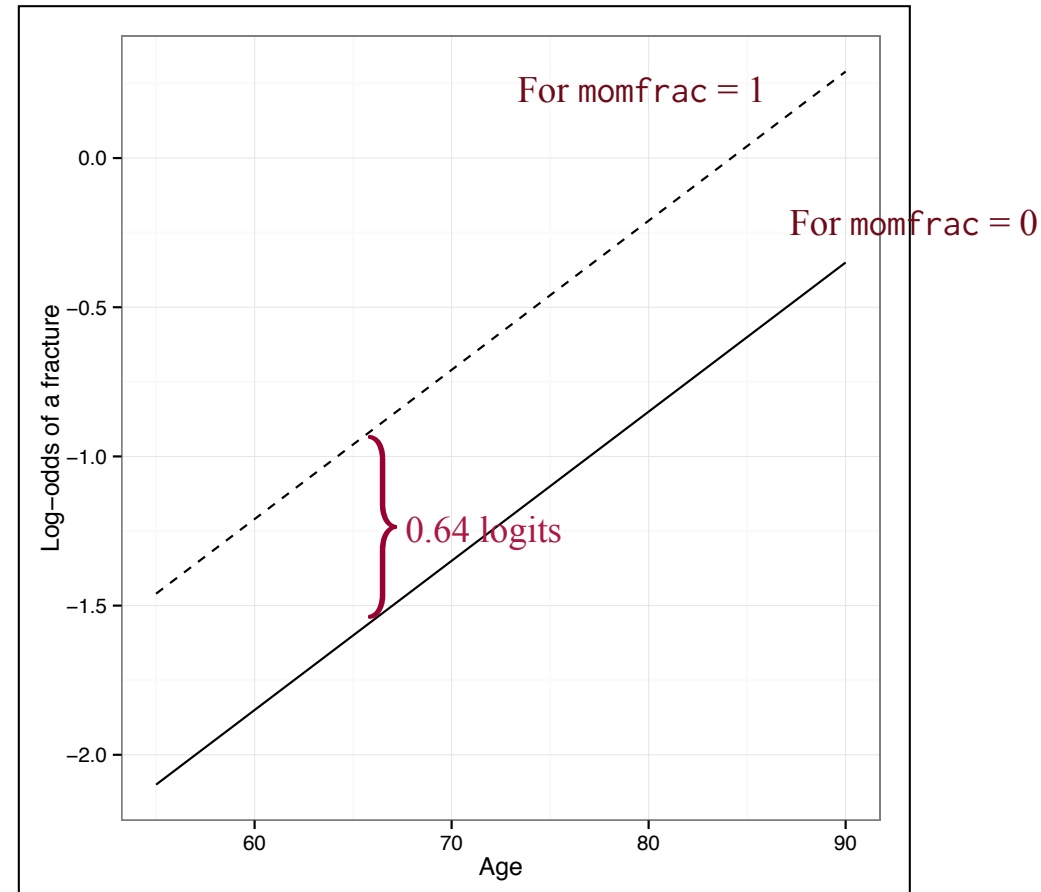
$$\hat{\text{fracture}} = -4.85 + 0.05(\text{age}) + 0.64(\text{momfrac})$$

The effect of momfrac

The predicted difference in log-odds of a fracture between subjects whose mom had a hip fracture and those who didn't is 0.64, controlling for age.

The effect of age

The predicted difference in log-odds of a fracture for a one-year difference in age is 0.05, controlling for whether the subject's mother had a hip fracture.



This implies that the effect of X_2 is exactly the same for any value of X_1 .

Since the two lines are parallel (same slope) the the effect of X_1 is exactly the same for the two groups (X_2).

Since the two lines are parallel (same slope) the difference in intercept measures the difference between the two groups for any value of X_1 . It is the *controlled* difference between the groups.

How about the odds?

$$\text{For momfrac} = 0 \quad \exp[\hat{\text{fracture}}] = \exp[-4.85 + 0.05(\text{age})]$$

$$\text{For momfrac} = 1 \quad \exp[\hat{\text{fracture}}] = \exp[-4.21 + 0.05(\text{age})]$$

Choose an age value and compute the odds for both groups.

$$\text{For momfrac} = 0 \quad \exp[\hat{\text{fracture}}] = \exp[-4.85 + 0.05(60)] = 0.1832051$$

$$\text{For momfrac} = 1 \quad \exp[\hat{\text{fracture}}] = \exp[-4.21 + 0.05(60)] = 0.3483134$$

age = 60

$$\left. \begin{array}{l} 0.1832051 \\ 0.3483134 \end{array} \right\} = 0.1651084$$

Compute the odds for both groups for (age + 1).

$$\text{For momfrac} = 0 \quad \exp[\hat{\text{fracture}}] = \exp[-4.85 + 0.05(61)] = 0.1930925$$

$$\text{For momfrac} = 1 \quad \exp[\hat{\text{fracture}}] = \exp[-4.21 + 0.05(61)] = 0.3671116$$

age = 61

$$\left. \begin{array}{l} 0.1930925 \\ 0.3671116 \end{array} \right\} = 0.1740191$$

$$\left. \begin{array}{l} 0.174 \\ 0.165 \end{array} \right\} \frac{0.174}{0.165} = 1.054$$

Changes by
a factor of
 β_2

Choose an age value and compute the odds for both groups.

age = 60

$$\begin{array}{l} \text{For momfrac} = 0 \quad \exp[\hat{\text{fracture}}] = \exp[-4.85 + 0.05(60)] = 0.1832051 \\ \text{For momfrac} = 1 \quad \exp[\hat{\text{fracture}}] = \exp[-4.21 + 0.05(60)] = 0.3483134 \end{array} \left. \vphantom{\begin{array}{l} \text{For momfrac} = 0 \\ \text{For momfrac} = 1 \end{array}} \right\} \frac{0.348}{0.183} = 1.90$$

Compute the odds for both groups for (age + 1).

age = 61

$$\begin{array}{l} \text{For momfrac} = 0 \quad \exp[\hat{\text{fracture}}] = \exp[-4.85 + 0.05(61)] = 0.1930925 \\ \text{For momfrac} = 1 \quad \exp[\hat{\text{fracture}}] = \exp[-4.21 + 0.05(61)] = 0.3671116 \end{array} \left. \vphantom{\begin{array}{l} \text{For momfrac} = 0 \\ \text{For momfrac} = 1 \end{array}} \right\} \frac{0.367}{0.193} = 1.90$$

Changes by
a factor of
 β_1

To better understand this, we plot the relationship for the two groups

```
# Create a data frame of X and the predicted odds for each group
> new = data.frame(
  age = 55:90
)

> new$odds0 = exp(-4.85 + 0.05 * new$age)
> new$odds1 = exp(-4.21 + 0.05 * new$age)

# Plot the two models
> ggplot(data = new, aes(x = age, y = odds0)) +
  geom_line() +
  geom_line(aes(x = age, y = odds1), lty = "dashed") +
  theme_bw()
```

```
> exp(coef(glm.c))
```

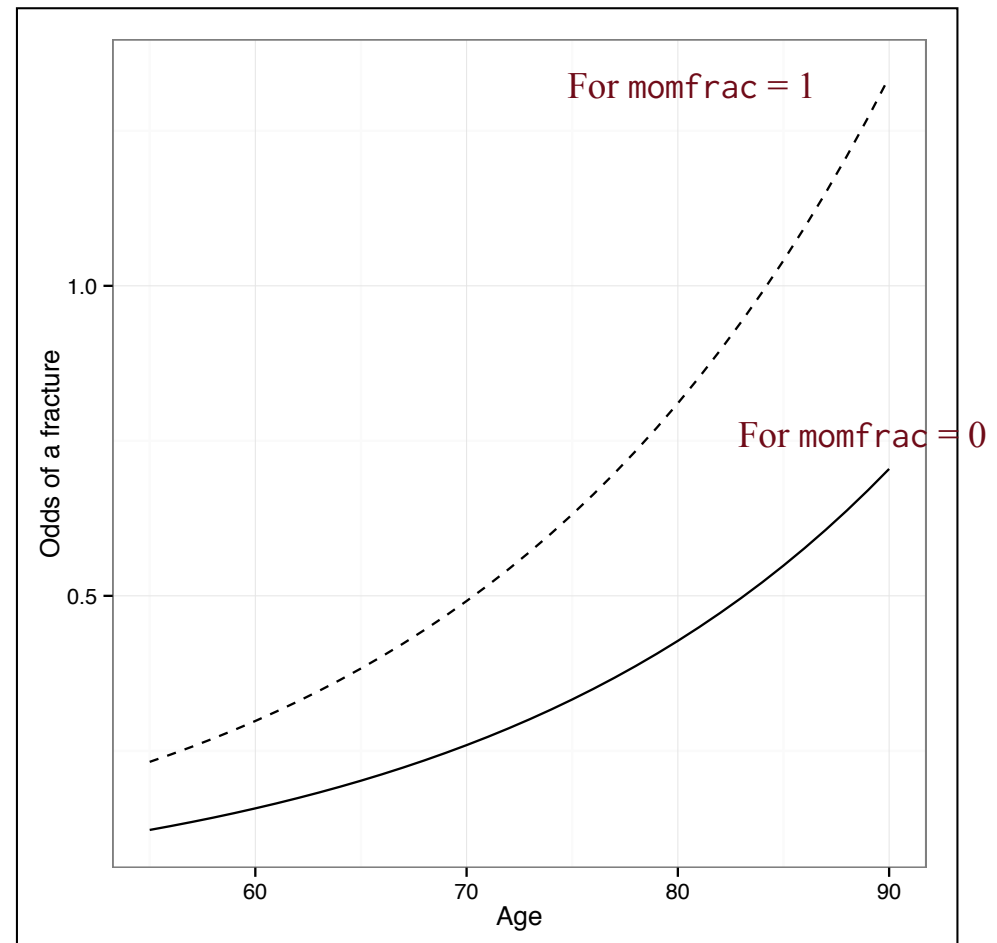
```
(Intercept)      age      momfrac  
0.007821009 1.053969181 1.901221667
```

The effect of momfrac

The odds of a fracture for subjects whose mom had a hip fracture are 1.90 times higher than for subjects whose mom did not have a hip fracture, controlling for age.

The effect of age

The odds of a fracture are 1.05 times higher, on average, for each year a subject ages, controlling for whether the subject's mother had a hip fracture.

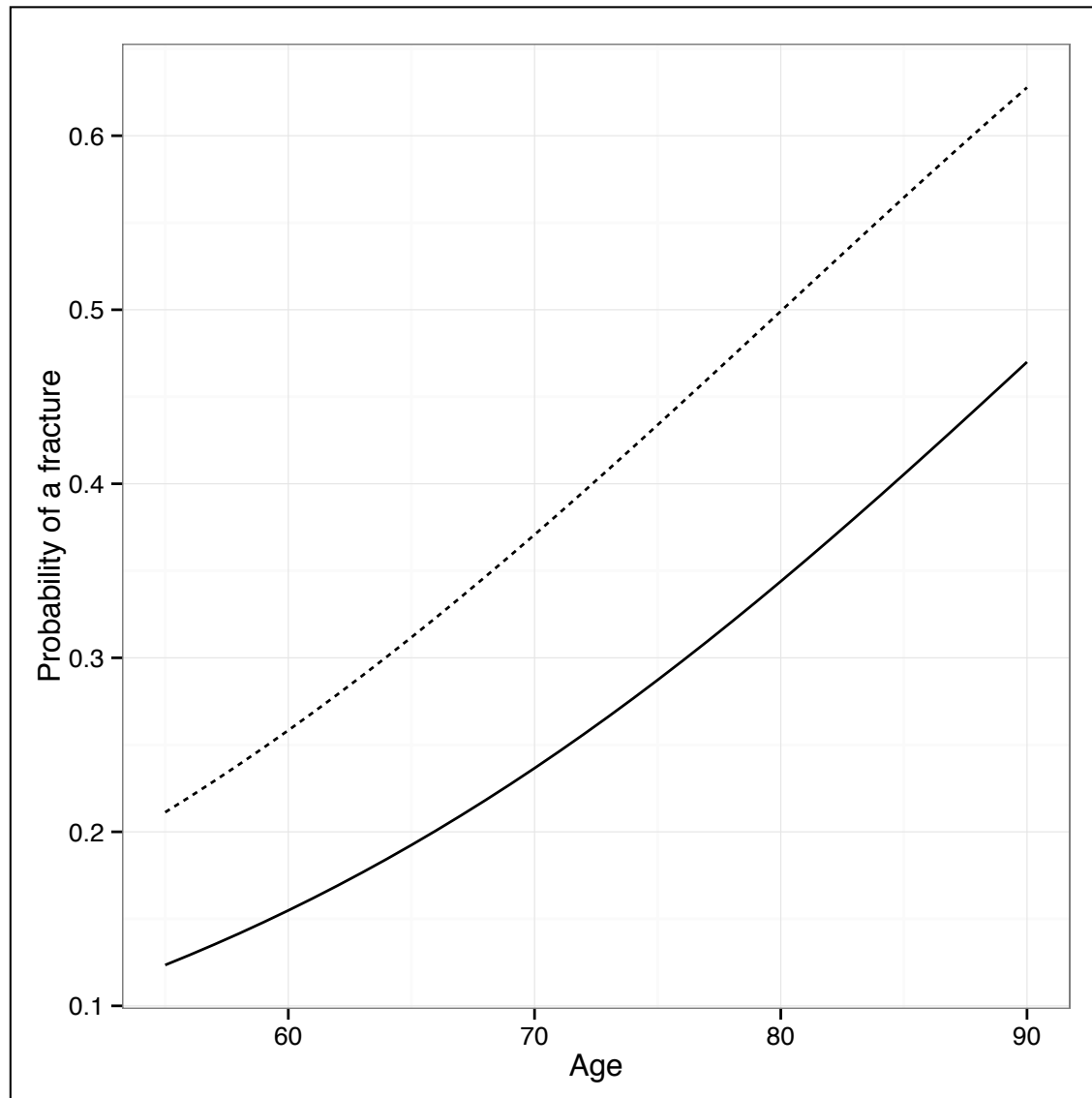


To plot probabilities...

```
> new = expand.grid(  
  age = 55:80,  
  momfrac = c(0, 1)  
)  
  
> new$probs = predict(glm.c, newdata = new, type = "response")  
  
> ggplot(data = new, aes(x = age, y = probs, group = factor(momfrac))) +  
  geom_line(aes(lty = factor(momfrac))) +  
  theme_bw() +  
  xlab("Age") +  
  ylab("Probability of a Fracture") +  
  guides(lty = FALSE)
```

Figure 1.

Fitted probability of a fracture conditioned on age for subjects whose mother had (dashed) and didn't have (solid) a hip fracture.



Interaction Models

Explore the Relationship

We should examine the relationship between age and fracture for each level of momfrac.

1. Create the proportion of fractures for several levels of age (cut) conditioning on momfrac
2. Plot these proportions
3. Are the relationships the same (at least within sampling variation) or different?

```
# Split age into 6 categories  
> glow$age2 = cut(glow$age, breaks = 6, include.lowest = TRUE)
```

```
# Get frequencies  
> tab = table(glow$age2, glow$fracture, glow$momfrac)
```

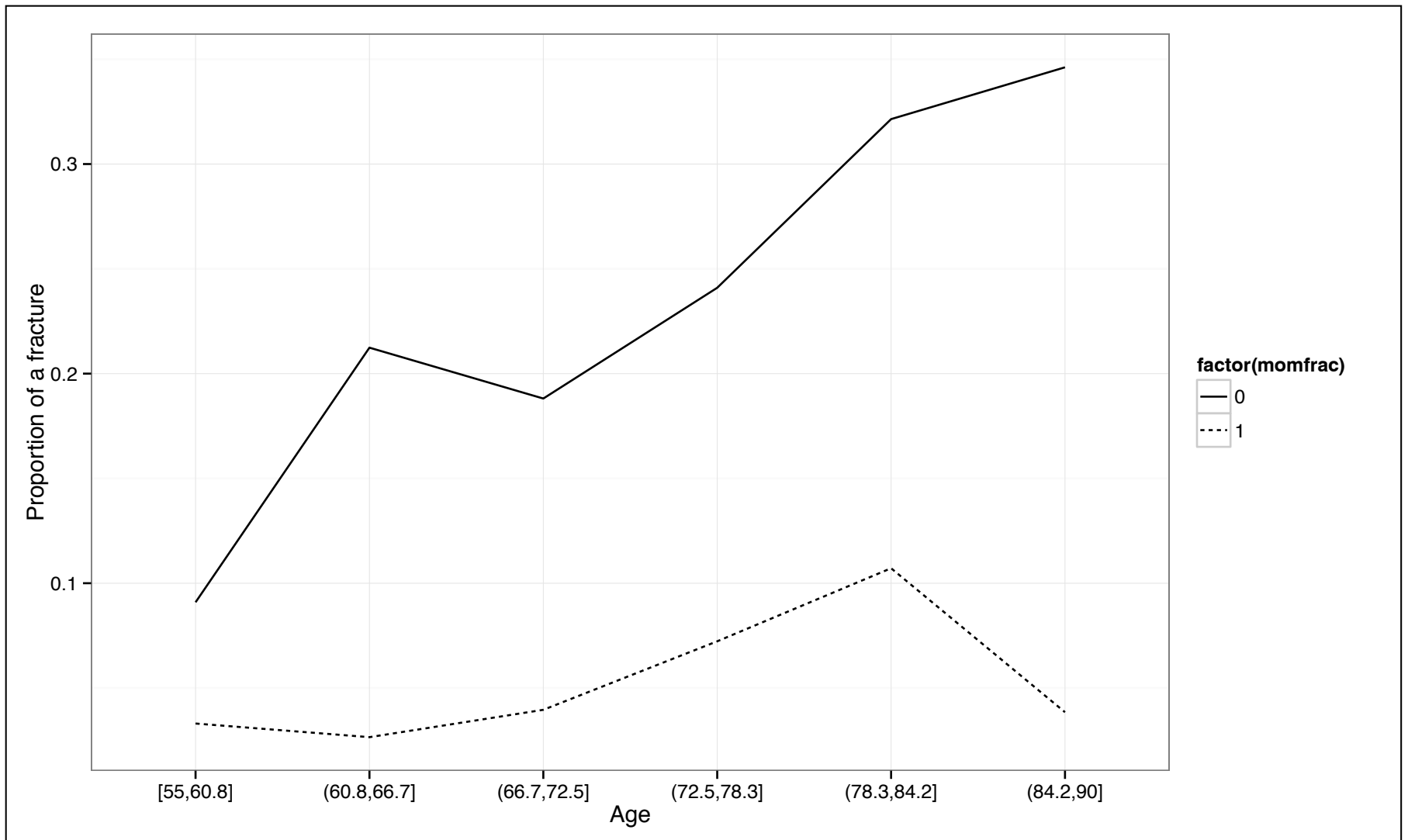
```
# Convert to proportions based on rows (margin = 1)  
> prop = prop.table(tab, 1)
```

```
# Turn into data frame for plotting  
> new = data.frame(prop)
```

```
# Re-name the variables  
> names(new) = c("age", "fracture", "momfrac", "prop")
```

```
# Plot the two lines  
> ggplot(data = new[new$fracture == 1, ],  
  aes(x = age, y = prop)  
  ) +  
  geom_line(  
    aes(group = factor(momfrac), lty = factor(momfrac))  
  ) +  
  theme_bw()
```

	Var1	Var2	Var3	Freq
1	[55,58.5]	0	0	0.78571429
2	(58.5,62]	0	0	0.78666667
3	(62,65.5]	0	0	0.59459459
4	(65.5,69]	0	0	0.72916667
⋮	⋮	⋮	⋮	⋮
37	(76,79.5]	1	1	0.09523810
38	(79.5,83]	1	1	0.11111111
39	(83,86.5]	1	1	0.00000000
40	(86.5,90]	1	1	0.06250000



The sample data show evidence of an interaction between age and momfrac

```
# Fit interaction model
> glm.d = glm(fracture ~ age + momfrac + age:momfrac, data = glow,
  family = binomial(link = "logit"))
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.78521    0.90896  -5.264 1.41e-07 ***
age          0.05163    0.01278   4.040 5.35e-05 ***
momfrac      0.23925    2.25950   0.106  0.916
age:momfrac  0.00573    0.03183   0.180  0.857
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 562.34  on 499  degrees of freedom
Residual deviance: 536.27  on 496  degrees of freedom
AIC: 544.27

Number of Fisher Scoring iterations: 4
```

Nope!

Table 1

Results of fitting a series of logistic regression models predicting fractures for $n = 500$ subjects. Coefficient-level inference includes the p -value based on the likelihood ratio statistic for the one-degree-of-freedom χ^2 .

Predictor	Model A		Model B		Model C		Model D	
	B	SE	B	SE	B	SE	B	SE
Mother hip fracture	0.66	0.28			0.64	0.29	0.24	2.26
	$(p = 0.022)$				$(p = 0.029)$		$(p = 0.029)$	
Age			0.05	0.01	0.05	0.01	0.05	0.01
			$(p < 0.001)$		$(p < 0.001)$		$(p < 0.001)$	
Mother had a hip fracture x Age							0.006	0.03
							$(p = 0.857)$	
(Intercept)	-1.20	0.11	-4.78	0.83	-4.85	0.83	-4.79	0.91
Model evaluation								
Deviance	557.1		541.1		536.3		536.3	
AIC	561.1		545.1		542.3		544.3	
BIC	569.5		553.5		554.9		561.1	

More Models

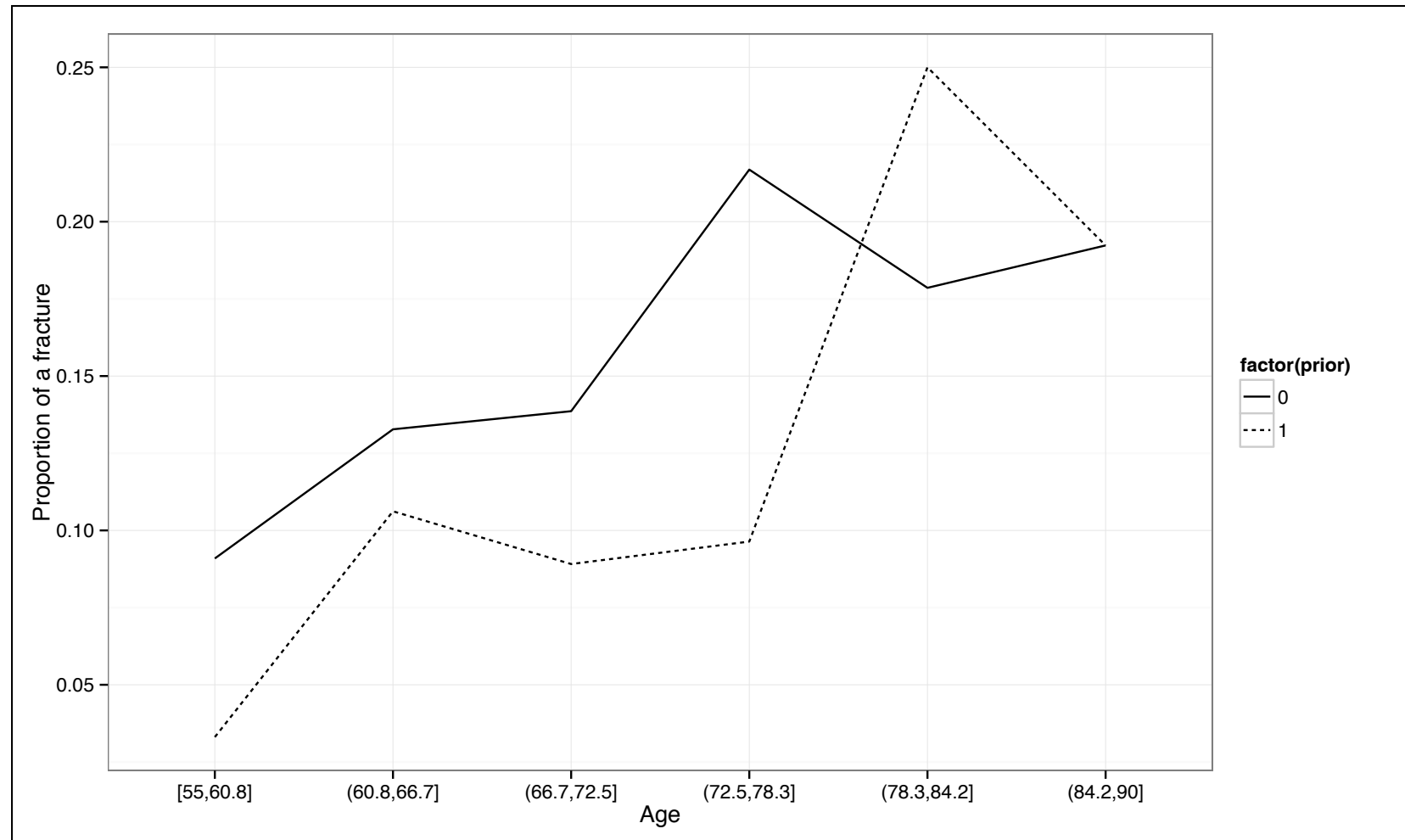
Table 1 (cntd.)

Results of fitting a series of logistic regression models predicting fractures for $n = 500$ subjects. Coefficient-level inference includes the p -value based on the likelihood ratio statistic for the one-degree-of-freedom χ^2 .

Predictor	Model E		Model F		Model G	
	B	SE	B	SE	B	SE
Mother hip fracture					0.66	0.29
					$(p = 0.025)$	
Age			0.04	0.01		
			$(p < 0.001)$			
Prior fracture	1.06	0.22	0.83	0.23	1.06	0.22
	$(p < 0.001)$		$(p < 0.001)$		$(p < 0.001)$	
(Intercept)	-1.42	0.13	-4.21	0.85	-1.51	0.14
Model evaluation						
Deviance	540.1		528.5		535.0	
AIC	544.1		534.5		541.0	
BIC	552.5		547.2		553.7	

Prior fracture seems to be an important predictor to include.

Age by prior fracture interaction?



momfrac by prior fracture interaction?

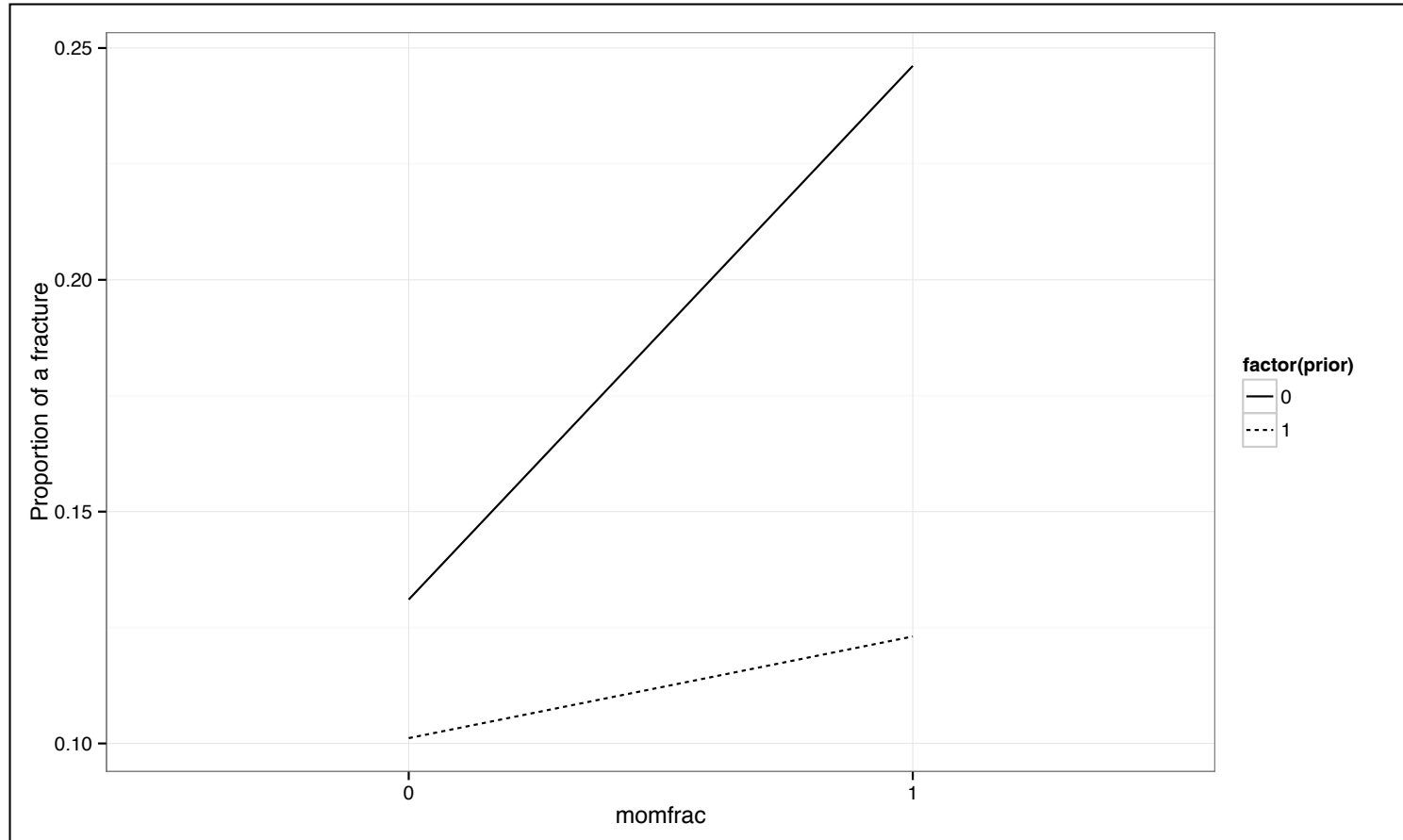


Table 1 (cntd.)

Results of fitting a series of logistic regression models predicting fractures for $n = 500$ subjects. Coefficient-level inference includes the p -value based on the likelihood ratio statistic for the one-degree-of-freedom χ^2 .

Predictor	Model H		Model I		Model J		Model K	
	B	SE	B	SE	B	SE	B	SE
Mother hip fracture			0.89	0.34	0.64	0.29	0.71	0.30
					($p = 0.032$)		($p = 0.019$)	
Age	0.06	0.02			0.04	0.01	0.06	0.02
					($p < 0.001$)			
Prior fracture	4.96	1.81	1.18	0.24	0.84	0.24	5.35	1.83
					($p < 0.001$)			
Age x Mother hip fracture								
Age x Prior fracture	-0.06	0.03					-0.06	0.03
	($p = 0.022$)						($p = 0.013$)	
Mother hip fracture x Prior fracture			-0.74	0.62				
			($p = 0.228$)					
(Intercept)	-5.69	1.08	-0.74	0.62	0.04	0.01	-5.90	1.10
Model evaluation								
Deviance	523.3		533.6		523.9		517.8	
AIC	531.3		541.6		531.9		527.8	
BIC	548.1		558.4		548.8		548.9	

Table 2

Best-fitting models.

Measure	Model	Terms
Deviance	Model K (517.8)	MF + A + P + A:P
AIC	Model K (527.8)	MF + A + P + A:P
BIC	Model F (547.1)	A + P

Presenting Results from "Final" Model

Table of Odds Ratios

We will compute the odds ratio of PF = 1 vs PF = 0 for several ages

$$\exp \left[-5.90 + 0.71(\text{MF}) + 0.06(\text{A}) + 5.35(\text{PF}) - 0.06(\text{PF})(\text{A}) \right]$$

MF = 0

Age	Odds (PF = 1)	Odds (PF = 0)	OR
60	0.623	0.128	4.87

Highlight the interaction

What should we do with the other predictors in the model?

- Set them to a fixed value(s) (e.g., mean).
- Since momfrac is dummy coded, we can compute results for momfrac = 0 or momfrac = 1 (they are the same)

Example for age = 60

$$\text{PF} = 1: \exp \left[-5.90 + 0.71(0) + 0.06(60) + 5.35(1) - 0.06(1)(60) \right] = 0.623$$

$$\text{PF} = 0: \exp \left[-5.90 + 0.71(0) + 0.06(60) + 5.35(0) - 0.06(0)(60) \right] = 0.128$$

$$\text{OR: } \frac{0.623}{0.128} = 4.87$$

```
> exp(-5.89630809 + 0.70771427*(0) + 0.06400330 * (60) + 5.34762635*1 - 0.06274619*60*1)
[1] 0.6229711

> exp(-5.89630809 + 0.70771427*(0) + 0.06400330 * (60) + 5.34762635*0 - 0.06274619*60*0)
[1] 0.1279507

> 0.6229711 / 0.1279507
[1] 4.868837
```

```
> new = expand.grid(
  age = c(60, 70, 80, 90),
  momfrac = c(0, 1),
  prior = c(0, 1)
)
```

```
> new$logits = predict(glm.k, newdata = new)
> new$odds = exp(new$logits)
```

```
> new
  age momfrac prior    logits    odds
1  60        0     0 -2.0561102 0.1279507
2  70        0     0 -1.4160772 0.2426641
3  80        0     0 -0.7760443 0.4602229
4  90        0     0 -0.1360113 0.8728328
5  60        1     0 -1.3483960 0.2596564
6  70        1     0 -0.7083630 0.4924497
7  80        1     0 -0.0683300 0.9339522
8  90        1     0  0.5717030 1.7712809
9  60        0     1 -0.4732550 0.6229712
10 70        0     1 -0.4606839 0.6308520
11 80        0     1 -0.4481128 0.6388326
12 90        0     1 -0.4355417 0.6469142
13 60        1     1  0.2344592 1.2642249
14 70        1     1  0.2470304 1.2802180
15 80        1     1  0.2596015 1.2964133
16 90        1     1  0.2721726 1.3128136
```

MF = 0

Age	Odds (PF = 1)	Odds (PF = 0)	OR
60	0.623	0.128	4.87
70	0.631	0.243	2.6
80	0.639	0.46	1.39
90	0.647	0.873	0.74

same as...

MF = 1

Age	Odds (PF = 1)	Odds (PF = 0)	OR
60	1.264	0.26	4.87
70	1.28	0.492	2.6
80	1.296	0.934	1.39
90	1.313	1.771	0.74

These odds ratios are called **relative risks**.

- They give the odds of getting a fracture for subjects who have had a prior fracture *relative* to those who haven't had a prior fracture.

In order to do this, we need to compute the variances for the predicted relative risks.

We would also like to obtain **prediction intervals** for the relative risks.

In general, the variance for predicted values is

$$\sigma^2\{\hat{Y}\} = \mathbf{x}'\boldsymbol{\sigma}_{\beta}^2\mathbf{x}$$

where $\boldsymbol{\sigma}_{\beta}^2$ is the variance-covariance matrix for the regression model, and $\mathbf{x}' = [1 \ x_1 \ x_2 \ x_3 \ \dots]$ is a row vector corresponding to an observation in the data

Since we are interested in the relative risk between subjects with prior = 1 and prior = 0 at a particular age and momfrac we need two row vectors

```
> covariates = expand.grid(  
  fracture = 1,  
  momfrac = 0,  
  prior = c(0, 1),  
  age = 60  
)
```

```
> new  
  fracture momfrac prior age  
1         1         0     0  60  
2         1         0     1  60
```

We set up the covariate values using `expand.grid()` and then use `model.matrix()` to obtain the two row vectors for the relative risks of interest

```
> m = model.matrix(glm.k, data = new)  
1  0  0  60  0  
1  0  1  60  60
```

```
> x.prime = c(-1, 1) %*% m
0 0 1 0 60
```

The relative risk is then calculated by subtracting the two vectors *or* multiplying by a contrast vector. (Note that the second row in *m* corresponds to the numerator of the relative risk, i.e., has the contrast value of 1)

The variance-covariance matrix is obtained using the `vcov()` function.

```
> v = vcov(glm.k)
              (Intercept)      momfrac      prior      age      prior:age
(Intercept)  1.20714580 -0.0355295227 -1.21616374 -0.0170024344  0.0171973666
momfrac      -0.03552952  0.0880551389  0.05787927  0.0002991445 -0.0007822577
prior        -1.21616374  0.0578792668  3.36074218  0.0170783618 -0.0460750511
age          -0.01700243  0.0002991445  0.01707836  0.0002436153 -0.0002452566
prior:age     0.01719737 -0.0007822577 -0.04607505 -0.0002452566  0.0006418425
```

We can now compute the standard deviation of the predicted values.

```
> se = sqrt(x.prime %*% v %*% t(x.prime))
0.3773181
```

Finally, we compute the CI and exponentiate to get the relative risk and its prediction limits.

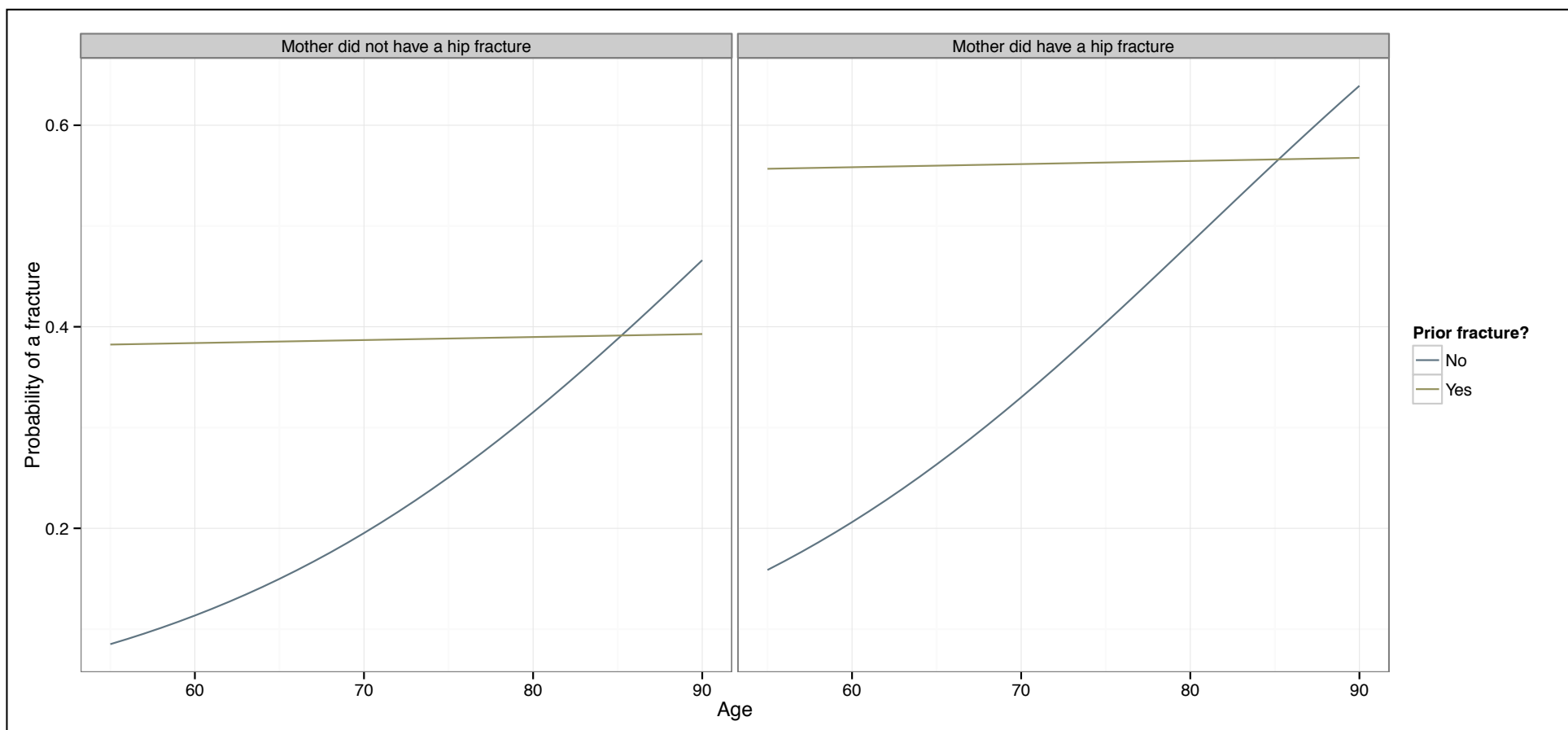
```
> exp(x.prime %*% coef(glm.k) + qnorm(c(0.50, 0.025, 0.975)) * se)
[1] 4.868837 2.324082 10.199974
```

Now, we put this into a function for easier use

```
> predict.or = function(a, m){  
  covariates = expand.grid(age = a, momfrac = m, prior = c(0, 1), fracture = 1)  
  m = model.matrix(glm.k, data = new)  
  x.prime = c(-1, 1) %*% m  
  v = vcov(glm.k)  
  se = sqrt(x.prime %*% v %*% t(x.prime))  
  print(exp(x.prime %*% coef(glm.k) + qnorm(c(0.50, 0.025, 0.975)) * se))  
}
```

We use the function by adding the arguments $a=age$ and $m=momfrac$

```
> predict.or(a = 60, m = 0)  
[1] 4.868837 2.324082 10.199974
```



To compare models, we can examine the **difference in their deviance** measures.

$$\mathcal{D}_{\text{Model Reduced}} - \mathcal{D}_{\text{Model Full}}$$

A reduced model will always have a larger deviance (more misfit) than a fuller model.

$$(-2 \times \ln \mathcal{L}_R) - (-2 \times \ln \mathcal{L}_F)$$

$$-2 \times (\ln \mathcal{L}_R - \ln \mathcal{L}_F)$$

$$-2 \times \ln \left(\frac{\mathcal{L}_R}{\mathcal{L}_F} \right)$$

We can substitute -2 times the log-likelihood for both models in to this difference

$$\mathcal{D}(\boldsymbol{\beta}; \mathbf{y}) = -2 \times \ln [\mathcal{L}(\boldsymbol{\beta}; \mathbf{y})]$$

The difference in deviances is -2 times a **likelihood ratio**. This is again called G .

$G \sim \chi^2$ with df = difference in the estimated number of parameters between the two models

$$G = -2 \times \ln \left(\frac{\mathcal{L}_R}{\mathcal{L}_F} \right)$$

To evaluate a predictor, we compute G based on the likelihood for the model when that predictor is included (full model) and when it is not (reduced model)

Under the null-hypothesis

$$H_0 : \beta_1 = 0$$

$$G \sim \chi^2_{df_R - df_F}$$

$$\ln \left(\frac{\pi_1}{1 - \pi_i} \right) = \beta_0 + \beta_1(\text{momfrac}) + \epsilon_i \quad \text{full model}$$

$$\ln \left(\frac{\pi_1}{1 - \pi_i} \right) = \beta_0 + \epsilon_i \quad \text{reduced model}$$

```
# Fit full model
> glm.f = glm(fracture ~ momfrac, data = glow, family = binomial(link = "logit"))

# Fit reduced model
> glm.r = glm(fracture ~ 1, data = glow, family = binomial(link = "logit"))

# Compute analysis of deviance
> anova(glm.r, glm.f, test = "LRT")
```

Analysis of Deviance Table

```
Model 1: fracture ~ 1
Model 2: fracture ~ momfrac
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      499      562.34
2      498      557.07  1    5.2698   0.0217 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This is exactly the results we obtained from the LRT test of goodness-of-fit.

The deviance for a model can be computed using the `deviance()` function.

```
# Compute deviance for the full model
> deviance(glm.f)

[1] 557.0654
```

The deviance is also given in the `summary()` results

```
> summary(glm.f)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.1960     0.1136 -10.532  <2e-16 ***
momfrac       0.6605     0.2810   2.351   0.0187 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 562.34  on 499  degrees of freedom
Residual deviance: 557.07  on 498  degrees of freedom
AIC: 561.07

Number of Fisher Scoring iterations: 4
```

In certain cases, the deviance from a single model can be used to evaluate that model's fit

Saturated Model

A baseline for model fit

Saturated model is a model that includes as many parameters as there are data points.

- $n = 2$; a model with two parameters (e.g., a model with a slope and intercept) would be a saturated model
- $n = 3$; a model with three parameters (e.g., a quadratic model) would be a saturated model

Saturated model will **always have perfect fit** to the observed data (think about $n = 2$ and a line).

- Since the number of model parameters is the same as the sample size, there is **no gain in parsimony** when we use the model as a proxy for the data
- The saturated model also shifts as we obtain new data, so there is **no generalizability** for the model (the saturated model *overfits* the data)

The saturated model in the GLOW example has 500 parameters

In GLMs, the saturated model has the same distribution and link function as the model of interest, but $g(\mu_i) = \psi_i$ for $i = 1, 2, 3, \dots, N$

In other words, the saturated model allows a different mean response for each subject and thus has N parameters.

Inference for the Logistic Regression Model

Saturated Model

Law of Likelihood

“Within the framework of a statistical model, a particular set of data supports one statistical hypothesis better than another if the likelihood of the first hypothesis on the data exceeds the likelihood of the second hypothesis”

If the fitted model is adequate, then we would expect that

$$\mathcal{L}(\text{Fitted Model}) \sim \mathcal{L}(\text{Saturated Model})$$

and

$$\frac{\mathcal{L}(\text{Fitted Model})}{\mathcal{L}(\text{Saturated Model})} \sim 1$$

Since, the deviance measures how closely the predicted values from the fitted model match the observed values from the raw data, one way to assess the adequacy of a particular fitted model is to compare its deviance with the deviance from the saturated model.

Saturated Model

Recall the likelihood is the joint probability of observing the data that were collected.

$$\mathcal{L}(\beta; \mathbf{y}) = 1$$

$$\ell(\beta; \mathbf{y}) = 0$$

$$\mathcal{D}(\beta; \mathbf{y}) = 0$$

$$G = \mathcal{D}(\text{Fitted Model}) - \mathcal{D}(\text{Saturated Model})$$

$$G = \mathcal{D}(\text{Fitted Model}) - 0$$

$$G = \mathcal{D}(\text{Fitted Model})$$

The difference in deviance has a known distribution and therefore can be used for hypothesis testing.

If the fitted model describes the data as well (or nearly as well) as the saturated model,

$$H_0 : \mathcal{D}(\text{Fitted Model}) - \mathcal{D}(\text{Saturated Model}) = 0$$

or

$$H_0 : \Gamma = 0$$

then

$$G \sim \chi^2_{N-p}$$

This suggests we expect a G or deviance near zero if the fitted model adequately fits the data.

The above deviance test compares the proposed model to the most general (saturated) model, and hence asks the question, can we use a more parsimonious model to describe the data as well as the most general model does?

The χ^2 distribution is correct for fixed df only when n is large.

The df for the saturated model is based on the sample size, n . And as $n \rightarrow \infty$ the df is not fixed ($df \rightarrow \infty$)

For categorical predictors—**grouped data**—we can assume a fixed df (since it is based on the number of values the predictor can take on), even as $n \rightarrow \infty$. No matter how large the sample, there are only a finite number of values the predictor can take.

For most GLMS, there is no benchmark distribution we can use to evaluate the deviance and get a p -value.

The best we can do is point out that a large deviance suggests misfit and a small deviance suggests better fit.

```
> deviance(glm.f)
```

```
[1] 557.0654
```

Although the deviance may suggest misfit, it is hard to evaluate (is this large misfit?) and also does not suggest *why* the misfit occurs.

To evaluate the degree of misfit, we fit the "worst" fitting model to see how bad the deviance can get.

To evaluate why there is misfit, we examine the model residuals.

Evaluating the Degree of Misfit

"Worst" fitting model

The "worst" fitting model is an intercept-only model

```
> deviance(glm.r)

[1] 562.3351
```

The deviance for the model that uses momfrac as a predictor improves the deviance from 562.3 to 557.1, a difference of 5.26.

This difference is statistically significant as already shown by the LRT.

```
> summary(glm.f)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.1960     0.1136 -10.532  <2e-16 ***
momfrac        0.6605     0.2810   2.351   0.0187 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 562.34  on 499  degrees of freedom
Residual deviance: 557.07  on 498  degrees of freedom
AIC: 561.07

Number of Fisher Scoring iterations: 4
```

The summary() function's output also includes the deviance for the "worst" fitting model.

Table 3

Results of fitting a series of logistic regression models predicting fractures for $n = 500$ subjects. Coefficient-level inference includes the Wald statistic and p -value for the one-degree-of-freedom χ^2 .

	Model A		Model B	
Predictor	B	SE	B	SE
Mother had a hip fracture (1 = Yes, 0 = No)	0.66	0.28		
	2.35 (<i>p</i> = .019)			
Age			0.05	0.01
			4.55 (<i>p</i> < .001)	
(Intercept)	−1.20	0.11	−4.78	0.83
	−10.53 (<i>p</i> < .001)		−5.78 (<i>p</i> < .001)	
Model evaluation				
Deviance	557.07		541.06	

The deviance for Model B is smaller than the deviance for Model A, indicating better fit.

Hosmer–Lemeshow Test

Hosmer and Lemeshow (1980) proposed a goodness-of-fit test for logistic models

- Order the fitted values from smallest to largest
- Form g equally sized groups with group 1 composed of the N/g observations with the smallest fitted values, group 2 composed of the observations having the next smallest fitted values, etc.
- Compute

$$\chi^2_{\text{HL}} = \sum_{i=1}^N n_i \frac{(\bar{y}_i - \hat{\pi}_i)^2}{\hat{\pi}_i(1 - \hat{\pi}_i)}$$

The diagram shows the formula $\chi^2_{\text{HL}} = \sum_{i=1}^N n_i \frac{(\bar{y}_i - \hat{\pi}_i)^2}{\hat{\pi}_i(1 - \hat{\pi}_i)}$ with red arrows pointing to specific parts and text labels:

- An arrow points from the summation index i to the label "Average observation in group i ".
- An arrow points from the term n_i to the label "Number of observations in group i ".
- An arrow points from the term \bar{y}_i to the label "Average observation in group i ".
- An arrow points from the term $\hat{\pi}_i$ to the label "Average fitted value in group i ".

The Hosmer–Lemeshow test statistic has a complicated distribution, but it can asymptotically be approximated by a χ^2 -distribution with $g-2$ degrees of freedom.

Large values of the statistic indicate lack of fit, but the test does not indicate why (e.g., model misspecification).

The test has only moderate power since the grouping is based on a model which is assumed to be true.

The test is highly influenced by the number of groups and Kuss (2002) suggested that the test may be highly unstable.


```
> library(MKmisc)
> HLgof.test(fit = fitted(glm.b), obs = glow$fracture)
```

```
$C
```

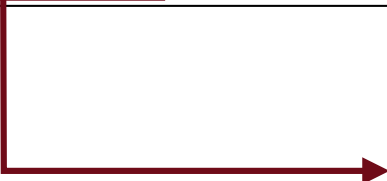
```
Hosmer-Lemeshow C statistic
```

```
data: fitted(glm.b) and glow$fracture
X-squared = 8.8399, df = 8, p-value = 0.356
```

```
$H
```

```
Hosmer-Lemeshow H statistic
```

```
data: fitted(glm.b) and glow$fracture
X-squared = 3.5434, df = 8, p-value = 0.8958
```



The large p -value does not indicate lack-of-fit.

```
> HLgof.test(fit = fitted(glm.f), obs = glow$fracture)

Warning messages:
1: In HLgof.test(fit = fitted(glm.f), obs = glow$fracture)
:
  Found only 1 different groups for Hosmer-Lemesho C
statistic.
2: In pchisq(chisq, param) : NaNs produced
```

Problems computing the HL test often indicate that there are not enough unique predictor values. Need to add other predictors to the model.

Examining Residuals

Assumptions for the Logistic Regression Model

There are very few assumptions for the logistic regression model.

- The model has been correctly specified (all meaningful predictors have been included)
- Errors are independent
- Linearity between the predictors and the logits
- Large sample size for tests and CIs

GLM Residuals

Residuals from a GLM are **not always** defined the same as OLS residuals

$$e_i = \cancel{Y_i} - \cancel{\hat{Y}_i}$$

R produces *five* different types of residuals for GLMs.

- Deviance residuals (default)
- Pearson residuals
- Working residuals
- Response residuals
- Partial residuals

The deviance can be expressed as the sum of N terms—one for each observation

$$\mathcal{D}(\beta; \mathbf{y}) = \sum D_i$$

Deviance residual is then defined as $\sqrt{D_i}$

The sign of the residual is the same as the sign of

$$Y_i - \hat{\mu}_i$$

The **Working residual** is the OLS defined residual based on the last iteration of the IRLS estimation process

The **Partial residual** is a matrix of the working residuals computed from omitting each predictor in the model

The **Response residual** is the OLS defined residual

$$e_i = Y_i - \hat{\mu}_i$$

Response residuals ignore the non-constant variance that is innate to the GLM models and are not recommended for diagnostic purposes

Standardizing the response residual according to the variance function $V(\mu)$ gives the

Pearson residual

$$e_i = \frac{Y_i - \hat{\mu}_i}{\sqrt{V(\hat{\mu}_i)}}$$

Since the Pearson residuals are often skewed, they are not recommended for diagnostic purposes

Examine Residuals

```
> head(resid(glm.b))
```

1	2	3	4	5	6
-0.6347136	-0.6816407	-1.1248673	-0.9962940	-0.6196586	-0.7143982

```
> head(resid(glm.b, type = "pearson"))
```

1	2	3	4	5	6
-0.4723891	-0.5113899	-0.9394774	-0.8016444	-0.4600613	-0.5391633

```
> out.b = fortify(glm.b)
```

```
> head(out.b)
```

fracture	age	.hat	.sigma	.cooksd	.fitted	.resid	.stdresid	
1	0	62	0.003139312	1.042996	0.0003524806	-1.4999047	-0.6347136	-0.6357122
2	0	65	0.002518194	1.042937	0.0003309431	-1.3412460	-0.6816407	-0.6825006
3	0	88	0.013026975	1.042149	0.0059016803	-0.1248631	-1.1248673	-1.1322665
4	0	82	0.006875734	1.042422	0.0022399864	-0.4421804	-0.9962940	-0.9997369
5	0	61	0.003387384	1.043015	0.0003609218	-1.5527909	-0.6196586	-0.6207108
6	0	67	0.002244193	1.042893	0.0003276592	-1.2354736	-0.7143982	-0.7152012

Predicted logit

Standardized
deviance residuals

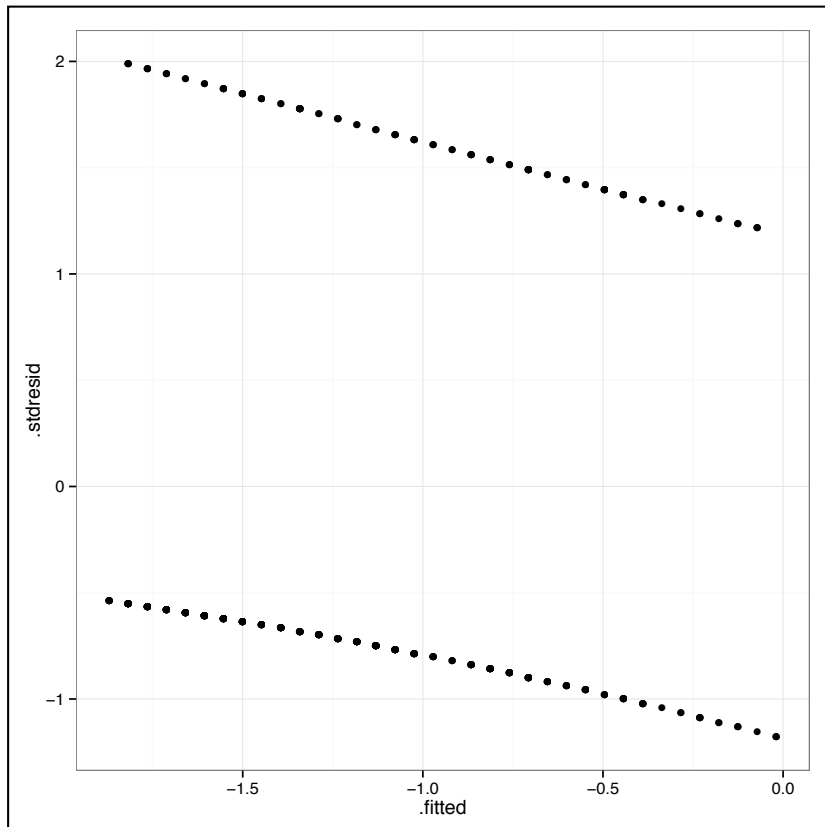
Leverage values

Cook's distance

Deviance residuals

Examine Residuals

```
> ggplot(data = out.b, aes(x = .fitted, y = .stdresid)) +  
  geom_point() +  
  theme_bw()
```



GLM: fracture ~ age

Each value of X produce a unique \hat{y} value. $Y - \hat{y}$ has only two possible values for the residual at each X

Gelman and Hill (2011) suggest examining **binned residuals** from a logistic regression. To compute binned residuals,

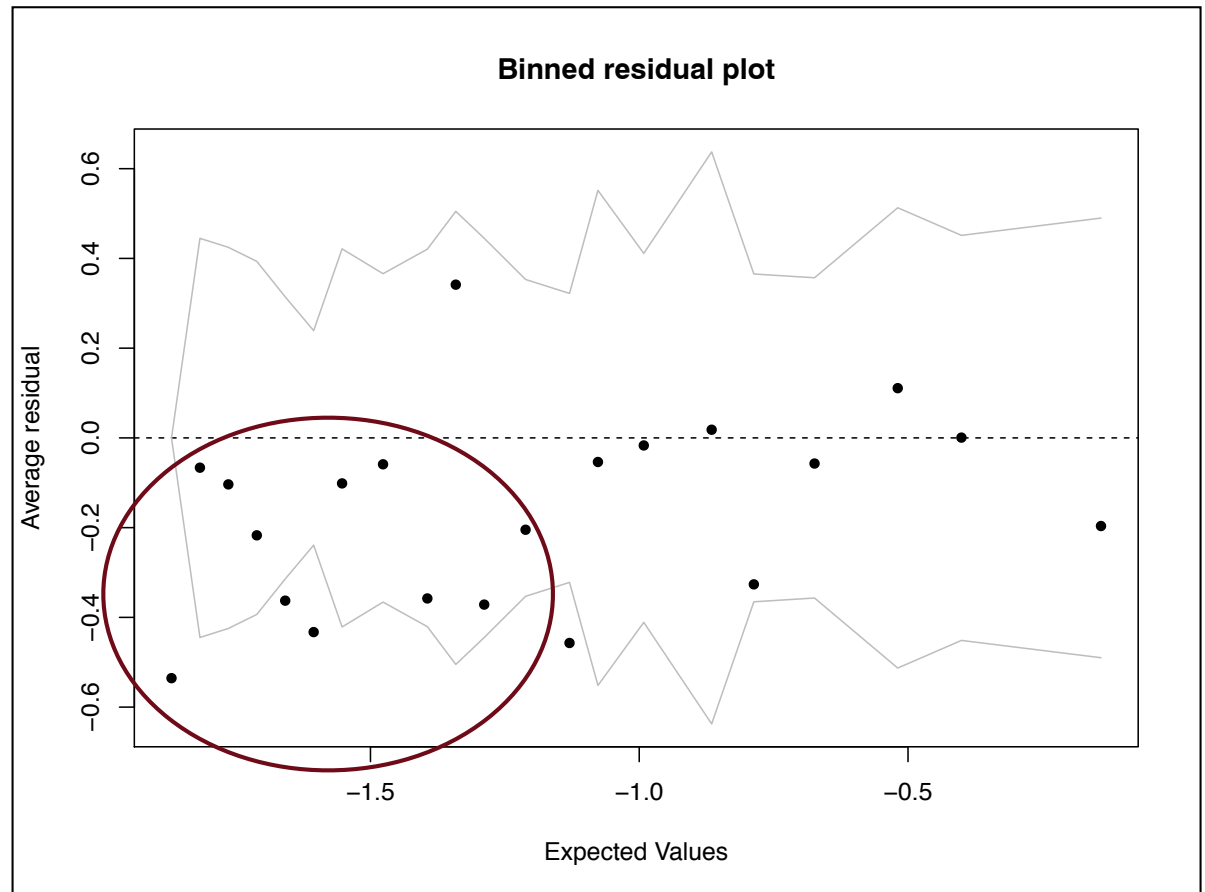
- In the residual plot form g bins, each having equal width (but not necessarily with equal number of observations)
- Compute the average residual within each bin
- Compute the average fitted value for each bin
- Plot the average residuals versus the average fitted value

```
> library(arm)
> binnedplot(x = out.b$fitted, y = out.b$resid)
```

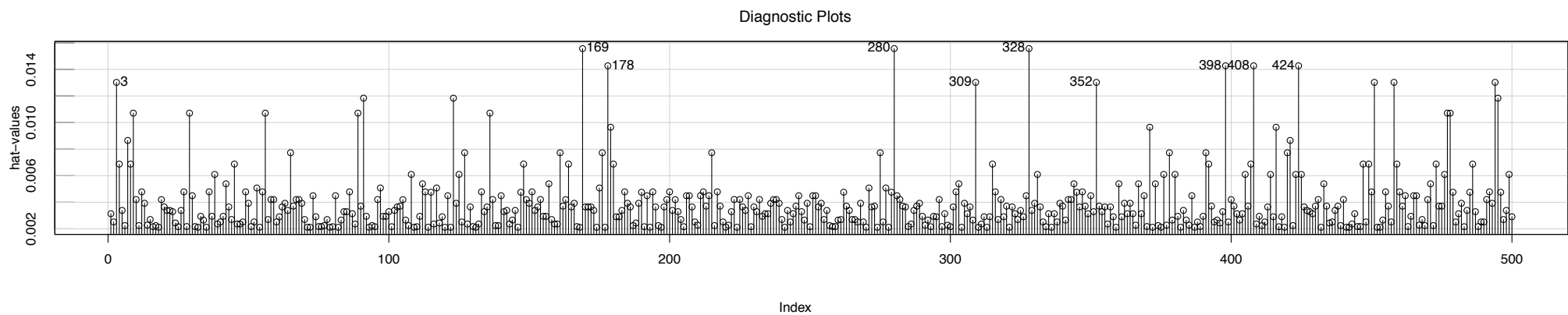
The number of bins is roughly

$$\sqrt{n}$$

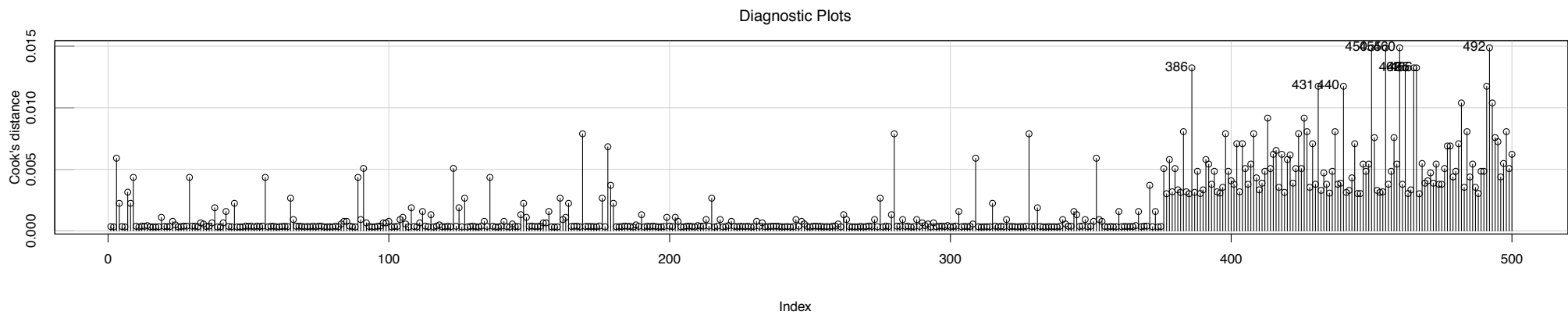
This can be changed by setting the argument `nclass=`



```
> library(car)
> influenceIndexPlot(glm.b, vars = "hat", id.n = 10)
```

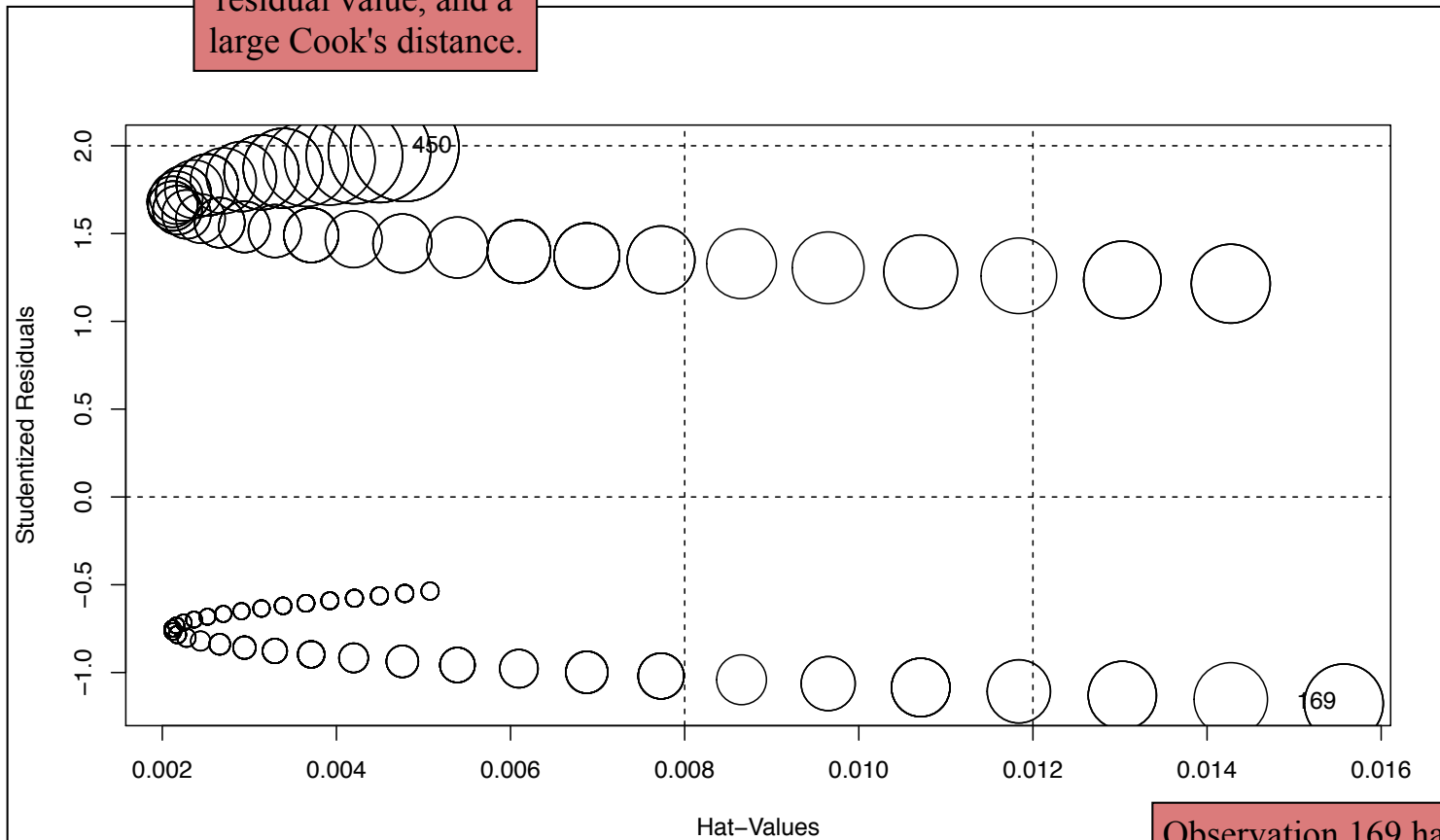


```
> influenceIndexPlot(glm.b, vars = "Cook", id.n = 10)
```




```
> influencePlot(glm.b, vars = "hat", id.n = 10)
```

Observation 460 has a high studentized residual value, and a large Cook's distance.



What happens if we remove these two observations?

Observation 169 has a high leverage value, and a large Cook's distance.

```
> glm.c = update(glm.b, subset = - c(169, 460))
```

Table 3

Results of fitting a series of logistic regression models predicting fractures for $n = 500$ subjects. Coefficient-level inference includes the Wald statistic and p -value for the one-degree-of-freedom χ^2 .

Predictor	Model B		Model C	
	B	SE	B	SE
Age	0.05	0.01	0.06	0.01
	4.55		4.74	
	($p < .001$)		($p < .001$)	
(Intercept)	-4.78	0.83	-4.99	0.84
	-5.78		-5.95	
	($p < .001$)		($p < .001$)	
Model evaluation				
Deviance	541.06		535.69	

There doesn't seem to be much difference in the size of the coefficients (nor their statistical reliability). The fit seems slightly improved.

```
> AIC(glm.b)
```

```
[1] 545.0613
```

One better measure of fit is **Akaike's Information Criteria (AIC)**.

- Basis in framework known as Kullback–Leibler information theory (Kullback & Leibler, 1951)
- AIC is also estimate of predictive accuracy
- Ability of model to predict new data
- This is often of interest to applied researchers

Penalty guards against
improved fit just by
adding predictors

$$AIC = \text{deviance} + 2 \cdot K$$

where K is the number of estimated parameters (number of fixed effects in model)

- Smaller deviance = better fit (same is true for AIC)
- If worthless predictors are added (i.e., deviance does not improve much), AIC will increase because $2K$ will go up
- If non-worthless predictors are added, $2K$ will still go up, but the decrease in deviance will outweigh this

Some details about the AIC to bear in mind

- AIC values are not standardized
- AIC values may be positive or negative
- AIC values may be very large or very small
- AIC does not offer a method of statistical testing (i.e., the term statistically significant should never be used with AIC)
- AIC offers a method of rank ordering the models in a given set of candidate models (nested or not)
- AIC is affected by sample size and should not be used to compare models across studies
- AIC is further changed when the response variable changes so it should not be used to compare models where the response is transformed

Another measure of fit is **Schwarz's (Bayesian) Information Criteria (BIC)**.

$$BIC = \text{deviance} + \ln(N) \cdot K$$

where K is the number of estimated parameters (number of fixed effects in model) and N is the sample size.

Penalty guards against improved fit just by adding predictors and increasing sample size

```
> BIC(glm.b)
```

```
[1] 553.4906
```

Some details about the AIC to bear in mind

- AIC values are not standardized
- AIC values may be positive or negative
- AIC values may be very large or very small
- AIC does not offer a method of statistical testing (i.e., the term statistically significant should never be used with AIC)
- AIC offers a method of rank ordering the models in a given set of candidate models (nested or not)
- AIC is affected by sample size and should not be used to compare models across studies
- AIC is further changed when the response variable changes so it should not be used to compare models where the response is transformed

Comparing Non-Nested Multilevel Models

AIC and BIC

You can (supposedly) compare non-nested multilevel models using information criteria

Information Criteria: AIC and BIC

Each information criterion “penalizes” the log-likelihood statistic for “excesses” in the structure of the current model

- The AIC penalty accounts for the number of parameters in the model.
- The BIC penalty goes further and also accounts for sample size.

Smaller values of AIC & BIC indicate better fit

Models need not be nested, but datasets must be the same.

Goodness-of-fit	Model A	Model B	Model C
Deviance	626.5	571.7	563.8
AIC	632.5	583.7	579.8
BIC	639.7	598	598.8

Disputing evidence...two candidate models

Interpreting differences in BIC across models

(Raftery, 1995):

- 0–2: Weak evidence
- 2–6: Positive evidence
- 6–10: Strong evidence
- >10: Very strong evidence

Careful: Gelman & Rubin (1995) declare these statistics and criteria to be “off-target and only by serendipity manage to hit the target”

Table 3

Results of fitting a series of logistic regression models predicting fractures for $n = 500$ subjects. Coefficient-level inference includes the likelihood ratio statistic and p -value for the one-degree-of-freedom χ^2 .

Predictor	Model A		Model B		Model C	
	B	SE	B	SE	B	SE
Mother had a hip fracture (1 = Yes, 0 = No)	0.66	0.28			0.64	0.29
	$\chi^2 = 5.27$				$\chi^2 = 4.76$	
	$(p = 0.022)$				$(p = 0.029)$	
Age			0.05	0.01	0.05	0.01
			$\chi^2 = 21.27$		$\chi^2 = 20.76$	
			$(p < 0.001)$		$(p < 0.001)$	
(Intercept)	-1.20	0.11	-4.78	0.83	-4.85	0.83
Model evaluation						
Deviance	557.07		541.06		536.30	
HL			3.54 ($df = 8$)		10.63 ($df = 8$)	
			$(p = 0.896)$		$(p = 0.224)$	