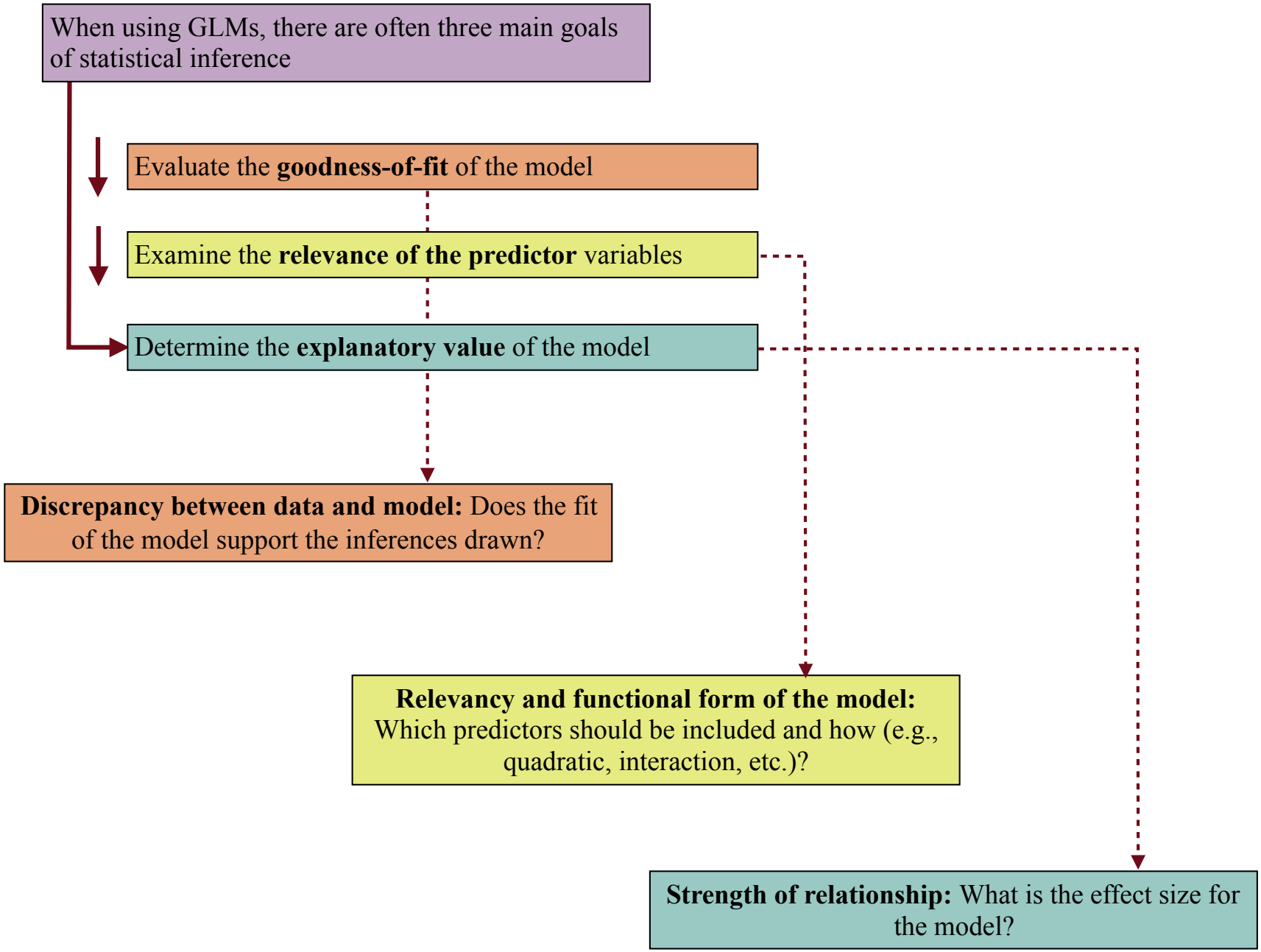
Inference for Logistic Regression

Andrew Zieffler
Department of Educational Psychology


When using GLMs, there are often three main goals of statistical inference
Evaluate the **goodness-of-fit** of the model
Examine the **relevance of the predictor** variables
Determine the **explanatory value** of the model
Discrepancy between data and model: Does the fit of the model support the inferences drawn?
Relevancy and functional form of the model: Which predictors should be included and how (e.g., quadratic, interaction, etc.)?
Strength of relationship: What is the effect size for the model?

Reading and Examining the Data

```
# Read in the data
glow = read.table(file = "http://www.tc.umn.edu/~zief0002/Data/GLOW.txt", header = TRUE)


# Examine data
head(glow)
```

```
  id site phys prior age weight height      bmi premeno momfrac armassist smoke selfrisk fracscore fracture
1  1    1   14     0  62   70.3    158 28.16055       0       0          0     0        2         1        0
2  2    4  284     0  65   87.1    160 34.02344       0       0          0     0        2         2        0
3  3    6  305     1  88   50.8    157 20.60936       0       1          1     0        1        11        0
4  4    6  309     0  82   62.1    160 24.25781       0       0          0     0        1         5        0
5  5    1   37     0  61   68.0    152 29.43213       0       0          0     0        2         1        0
6  6    5  299     1  67   68.0    161 26.23356       0       0          0     1        2         4        0
```

momfrac
Mother had a hip fracture
(1=yes; 0=no)

fracture
(Response)

```
summary(glow$momfrac)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    0.00    0.13    0.00    1.00
```

Range is between 0 and 1

In the sample of 500 subjects...

- 13% had a mother who fractured her hip
- 87% had a mother who never fractured her hip

Examining the Relationship with Fractures

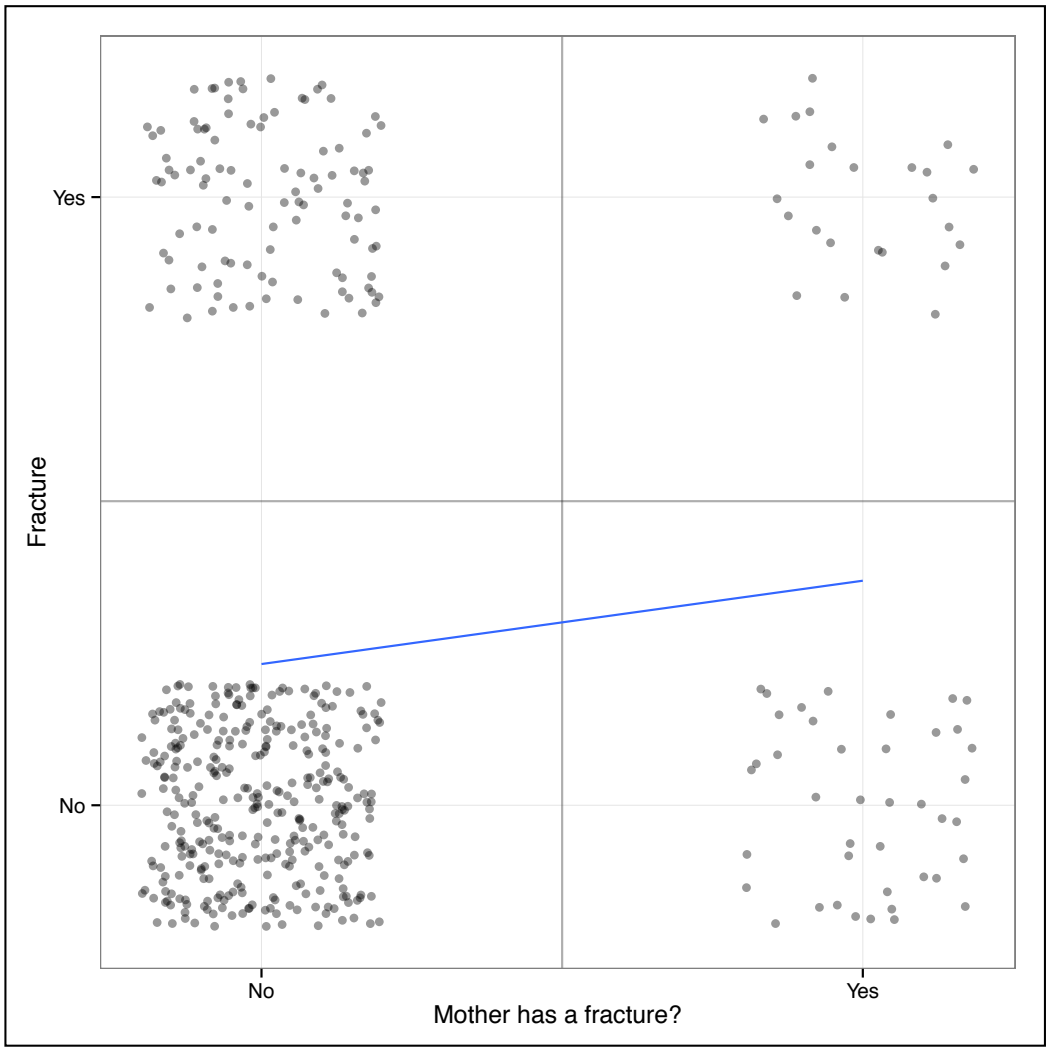
```
CrossTable(glow$fracture, glow$momfrac, format = "SPSS")
```

```
   Cell Contents
|-------------------------|
|                   Count |
| Chi-square contribution |
|             Row Percent |
|          Column Percent |
|            Total Percent |
|-------------------------|
```

```
Total Observations in Table:  500

                | glow$momfrac
glow$fracture |         0 |         1 | Row Total |
--------------|-----------|-----------|-----------|
            0 |       334 |        41 |       375 |
              |     0.184 |     1.232 |           |
              |    89.067% |    10.933% |    75.000% |
              |    76.782% |    63.077% |           |
              |    66.800% |     8.200% |           |
--------------|-----------|-----------|-----------|
            1 |       101 |        24 |       125 |
              |     0.552 |     3.696 |           |
              |    80.800% |    19.200% |    25.000% |
              |    23.218% |    36.923% |           |
              |    20.200% |     4.800% |           |
--------------|-----------|-----------|-----------|
 Column Total |       435 |        65 |       500 |
              |    87.000% |    13.000% |           |
--------------|-----------|-----------|-----------|
```

- **Of the subjects who had a mother who fractured her hip**
  - 36.9% had a fracture
  - 63.1% did not have a fracture
- **Of the subjects who had a mother who never fractured her hip**
  - 23.2% had a fracture
  - 76.8% did not have a fracture

We are conditioning on `momfrac` (in the table columns) so we look at the column percentages



Scatterplot for categorical predictor and categorical outcome

- Only four possible ordered pairs (0, 0), (0, 1), (1, 0), and (1,1)
- Jitter artificially adds variation so that the pattern is easier to see
- Positive relationship would show density in the (0, 0) and (1, 1) quadrants
- Negative relationship would show density in the (0, 1) and (1, 0) quadrants

The regression suggests a **positive relationship**...having a mother who had a fracture is associated with subjects having a fracture

(Because so many subjects did not experience a fracture, it is weighted toward the bottom of the plot.)

The relationship between two categorical variables is measured through the phi (Φ) coefficient

Phi Coefficient: Measure of Association between Two Dichotomous Variables

	Fracture		
Mother fracture	No	Yes	
No	334	41	375
Yes	101	24	125
	435	65	500

	y		
x	0	1	
0	A	B	$(A+B)$
1	C	D	$(C+D)$
	$(A+C)$	$(B+D)$	N

The numerator looks at the product of the sample sizes for the (0, 0) and (1, 1) ordered pairs and compares it to the product of the sample sizes for the (0, 1) and (1,0) ordered pairs.

If the difference is positive it means there is more weight attributed to the ordered pairs that have like *x* and *y* values.

If the difference is negative it means there is more weight attributed to the ordered pairs that have *x* and *y* values that are different.

$$\Phi = \frac{AD - BC}{\sqrt{(A+B)(C+D)(A+C)(B+D)}}$$

Note: The phi coefficient is Pearson's *r* for dummy coded *x* and *y*

Phi Coefficient: Measure of Association between Two Dichotomous Variables

	Fracture		
Mother fracture	No	Yes	
No	334	41	375
Yes	101	24	125
	435	65	500

$$\Phi = \frac{(334 \times 24) - (101 \times 41)}{\sqrt{(375)(125)(435)(65)}}$$

$$\Phi = \frac{8016 - 4141}{36405.91}$$

$$\Phi = 0.1064388$$

The phi coefficient shows positive association between the two variables.

```
cor(glow$fracture, glow$momfrac)
```

```
[1] 0.1064387
```

There are several other measures of association between dichotomous variables that researchers use (see Agresti)

- Pearson's contingency coefficent (C)
- Cramer's V coefficient
- Goodman and Kruskal's lambda coefficient (λ)
  - Symmetric
  - Asymmetric

Inference for Model Parameters

```
> glm.a <- glm(fracture ~ momfrac, data = glow, family = binomial(link = "logit"))
> summary(glm.a)


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 *
```

Parameter estimates from the logistic regression (odds)

- The intercept, $e^{-1.20} = 0.302$, is the **predicted odds** of getting a fracture for subjects whose mother did not have a hip fracture.
  - ✓ Recall odds values < 1 correspond to odds with higher numbers in the denominator than in the numerator. (higher probability of no fracture for these subjects than fracture)
  - ✓ The reciprocal, 1/0.302 = 3.31, gives the **odds of not getting a fracture** for subjects whose mother did not have a hip fracture.
  - ✓ The odds of getting a fracture for subjects whose mother did not have a hip fracture are ~3:1 against

- The slope, $e^{-1.20 + 0.66} = 0.582$, is the **predicted odds** of getting a fracture between subjects whose mother did and did not have a hip fracture.
  - ✓ The predicted odds of getting a fracture for subjects whose mother had a hip fracture are $e^{-0.66} = 1.94$ times higher than for subjects whose mother did not have a hip fracture

Parameter estimates from the logistic regression (logits/log-odds)

- The intercept, –1.20, is the predicted log-odds of getting a fracture for subjects whose mother did not have a hip fracture

- The slope, 0.66, is the **predicted difference (change)** in the log-odds of getting a fracture between subjects whose mother did and did not have a hip fracture.
  - –1.20 + 0.66 = –0.54 is the predicted log-odds of getting a fracture for subjects whose mother had a hip fracture

Parameter estimates from the logistic regression (probabilities)

- The intercept, $e^{-1.20}/ (1 + e^{-1.20}) = 0.232$, is the **predicted probability** of getting a fracture for subjects whose mother did not have a hip fracture.

- $e^{-0.54}/ (1 + e^{-0.54}) = 0.369$, is the **predicted probability** of getting a fracture for subjects whose mother did not have a hip fracture

Testing Model Parameters

Three primary methods used to test model parameters in the GLM

- Wald test
- Likelihood ratio test
- Score test

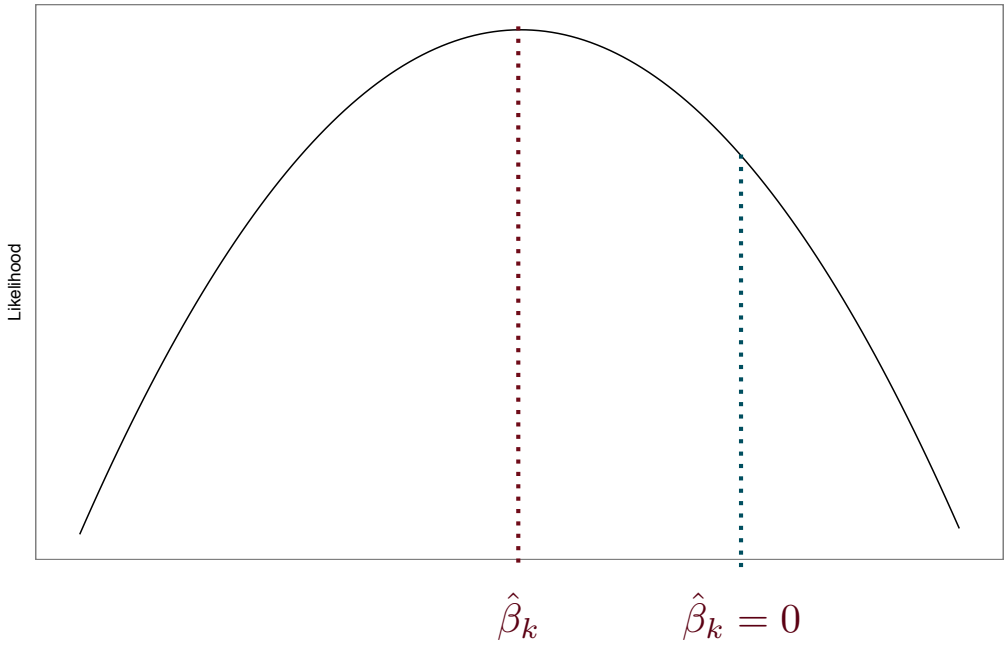
$$H_0 : \beta_k = 0$$

All three methods utilize the likelihood function

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

and the large-sample normality property of the ML estimators

Conceptually, the SE evaluated at a some value, say $\hat{\beta}_k$ is linked to the curvature of the likelihood function at $\hat{\beta}_k$



The SE is found by substituting the ML estimates into the inverse of the information function.

This implies that in order to estimate the SE, we need to make the assumption that H_0 is not true

The information matrix is minus the expected value of the second derivatives of the likelihood (or log-likelihood) function

$$I(\boldsymbol{\beta}) = -E\left[\frac{\partial^2 \mathcal{L}(\boldsymbol{\beta}; \mathbf{y})}{\partial \beta_k}\right]$$

Wald test

The Wald statistic, denoted z, is

$$z = \frac{\hat{\beta}_k - 0}{SE_{\hat{\beta}_k}} \qquad SE_{\hat{\beta}_k} = \frac{1}{\sqrt{I(\boldsymbol{\beta})}}$$

$SE_{\hat{\beta}_k}$ is obtained from the inverse of the information matrix evaluated at $\hat{\beta}_k$

when β = 0,

$z \sim \mathcal{N}(0,1)$ two-tailed

$z^2 \sim \chi^2(1)$ one-tailed

The `summary()` function for a GLM in R produces a Wald test for each parameter estimate in the model.

```
> glm.a <- glm(fracture ~ momfrac, data = glow, family = binomial(link = "logit"))
> summary(glm.a)


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 *
```

$\hat{\beta}_k$ $SE_{\hat{\beta}_k}$ z

Wald, A. (1943). Tests of statistical hypothoses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society, 54,* 426–482.

Likelihood ratio test

The likelihood ratio test compares the likelihoods of two models:

1. A reduced model composed under the null hypothesis where $\beta_k = 0$
2. A fuller model where $\beta_k \neq 0$

$$Y_i = \beta_0 + \epsilon_i \qquad \text{Reduced model}$$

$$Y_i = \beta_0 + \beta_k(X_{ik}) + \epsilon_i \qquad \text{Fuller model}$$

$$G^2 = -2 \times \ln\left(\frac{\mathcal{L}_{\mathrm{R}}}{\mathcal{L}_{\mathrm{F}}}\right)$$

The measure–2 times the **likelihood ratio** is distributed as

$$G^2 \sim \chi^2(1)$$

Reduced Model

$$\mathcal{L} = 7.770963 \times 10^{-123}$$
$$\ell = -281.1676$$

Full Model

$$\mathcal{L} = 1.083407 \times 10^{-121}$$
$$\ell = -278.5327$$

```
# 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 G
> LR = exp(logLik(glm.r)[1]) / exp(logLik(glm.f)[1])
> G = -2 * log(LR)
> G
[1] 5.269774

# Compute p-value
> pchisq(G, df = 1, lower.tail = FALSE)
[1] 0.02169883
```

It is also possible to carry out the likelihood ratio test using the `anova()` function with the argument `test="LRT"` or `test="Chisq"`

```
# Fit models
> glm.f = glm(fracture ~ momfrac, data = glow, family = binomial(link = "logit"))
> glm.r = glm(fracture ~ 1, data = glow, family = binomial(link = "logit"))

# Likelihood ratio test
> 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
```

Score test

The score test, sometimes called the Lagrange multiplier test, uses the slope of the log-likelihood (i.e., the score function) for a fitted model to examine the statistical importance of the predictor.

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

# Obtain score test
> anova(glm.f, test = "Rao")

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev    Rao Pr(>Chi)
NULL                      499     562.34
momfrac  1   5.2698       498     557.07 5.6646  0.01731 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Comparison of the Three Tests

Wald test, Likelihood ratio test, Score test

- All three tests answer the question, does a fuller model (i.e., adding predictors) improve the fit of the model?
- Each test has some advantages and disadvantages
- The three tests are asymptotically equivalent as N approaches infinity
- For finite sample sizes, the three tests will give slightly different results, but generally lead to the same practical conclusion about the predictor importance.

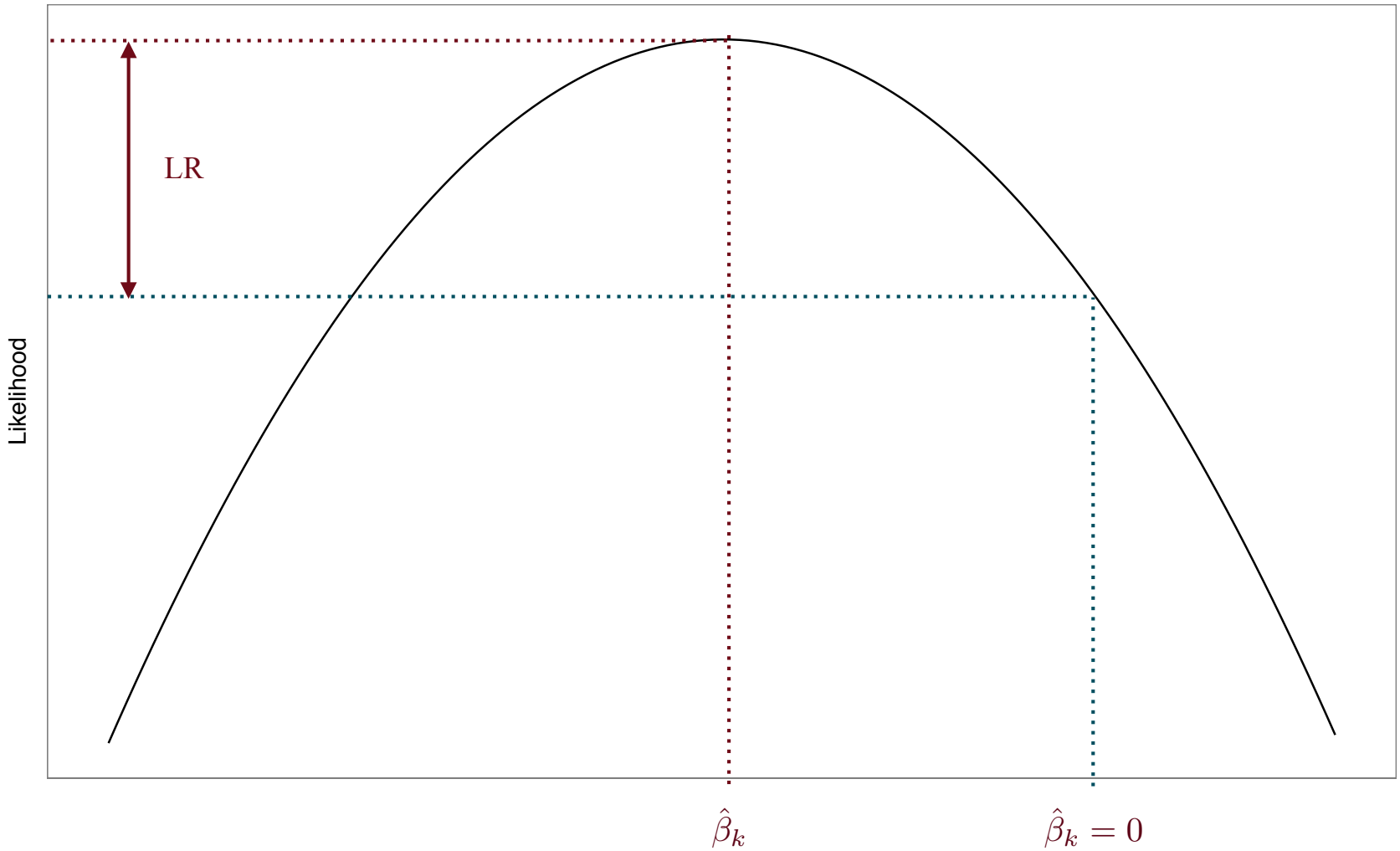
For the Wald and Score tests, computation of the statistic requires the estimation of only one model...this was a computational advantage in the early days of computing

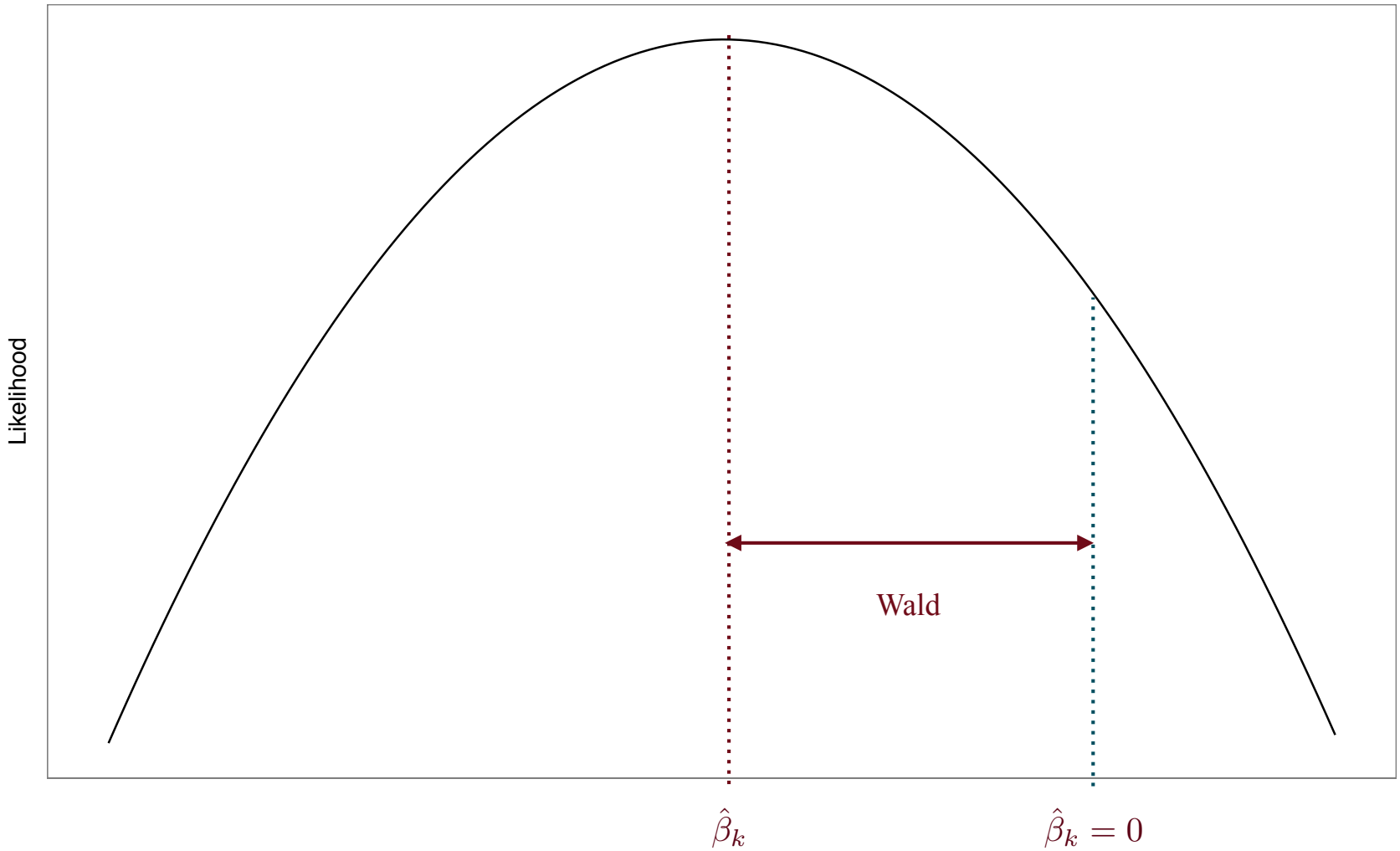
When there is only one parameter that is different between a full and the reduced model, for large n the likelihood ratio (LR) = Z^2_{Wald}

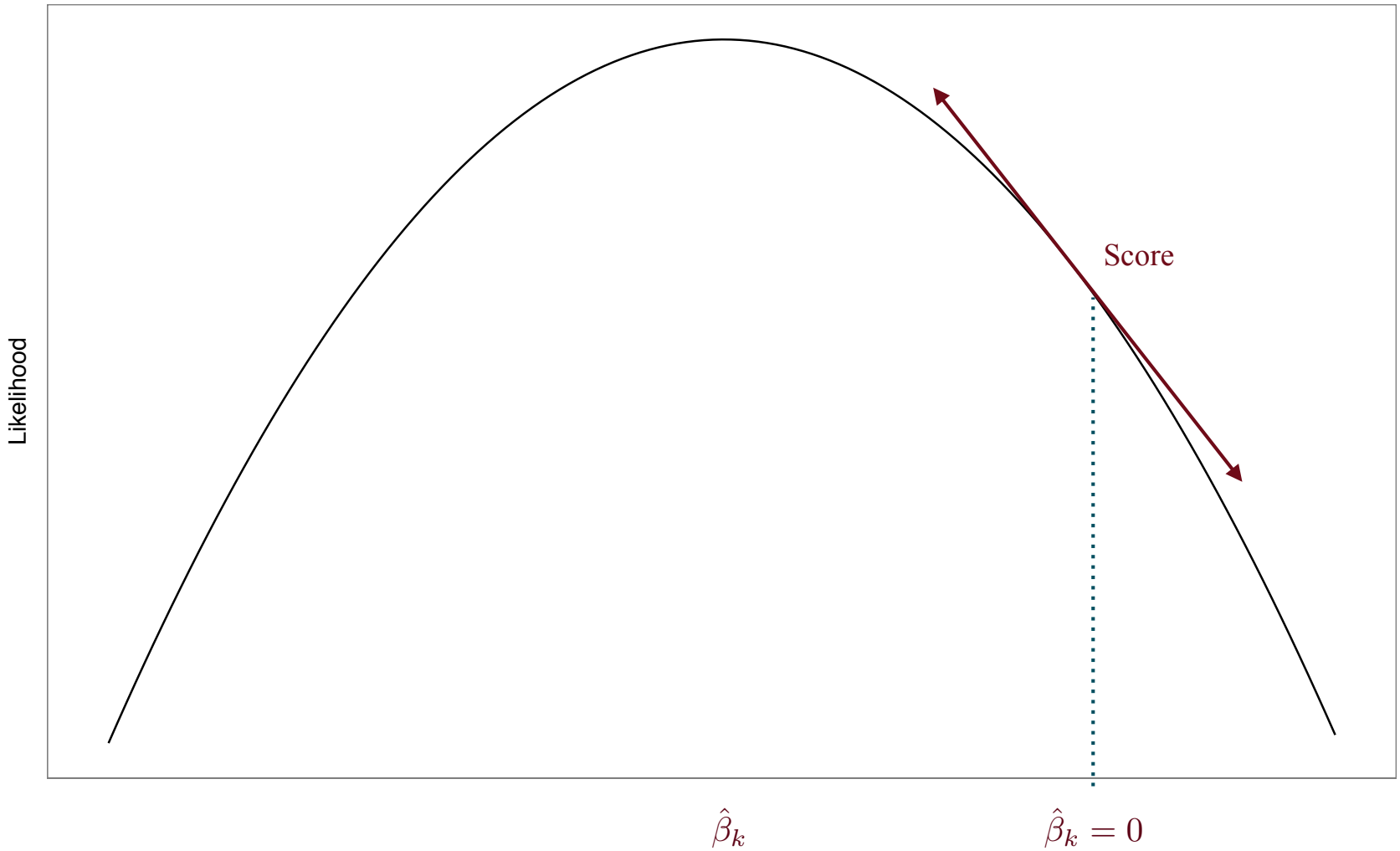
Test	Statistic	p-value
Score	5.665	0.0173
Wald	2.351	0.0187
LR	5.2698	0.0217

$2.35^2 = 5.52 \approx 5.27$

LR test is recommended.


LR
Likelihood
$\hat{\beta}_k$
$\hat{\beta}_k = 0$


Likelihood
Wald
$\hat{\beta}_k$
$\hat{\beta}_k = 0$


Likelihood
Score
$\hat{\beta}_k$
$\hat{\beta}_k = 0$

Confidence Intervals and Envelopes

Wald intervals

Confidence intervals can also be computed based on the Wald statistic

$$\hat{\beta}_j \pm 2 \times SE_{\hat{\beta}_j}$$

```
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 *
```

$$0.6605 \pm 2 \times 0.2810$$

$$[0.0985, 1.2225]$$ Confidence limits for the effect on the logit

With 95% confidence the log-odds of having a fracture if your mother had a hip fracture is between 0.10 and 1.22.

Confidence intervals based on the Wald statistic are formed by **inverting** the Wald tests

- Inversion determines the parameter values for which the null hypothesis is not rejected

$$H_0 : \beta_1 = 1.10$$
$$H_0 : \beta_1 = 1.20$$
$$\vdots$$
$$H_0 : \beta_1 = 3.40$$

Fail to reject

For small sample sizes the Wald test can be terribly wrong , which means the CI based on the inversion will be wrong as well

Profile likelihood intervals

Better CIs can be produced by inverting the LR test. This is done by **profiling the likelihood**.

- Every parameter *except* the parameter of interest is fixed to its ML estimate
- The likelihood (or log-likelihood) is then maximized over the parameter of interest
- The result is plotted in a profile plot
- Any reduced model that has a log-likelihood greater than the log-likelihood of the fitted model – half the 0.95 quantile of the chi-squared distribution is included in the interval

Fixing a parameter produces a reduced model from the initial fitted model.

When would we fail to reject H_0?

$$-2 \times \ln\left(\frac{\mathcal{L}_{\mathrm{R}}}{\mathcal{L}_{\mathrm{F}}}\right) < \chi^2_{0.95}$$

$$-2 \times \ln\left(\mathcal{L}_{\mathrm{R}}\right) + 2 \times \ln\left(\mathcal{L}_{\mathrm{F}}\right) < \chi^2_{0.95}$$

$$-2 \times \ln\left(\mathcal{L}_{\mathrm{R}}\right) < \chi^2_{0.95} - 2 \times \ln\left(\mathcal{L}_{\mathrm{F}}\right)$$

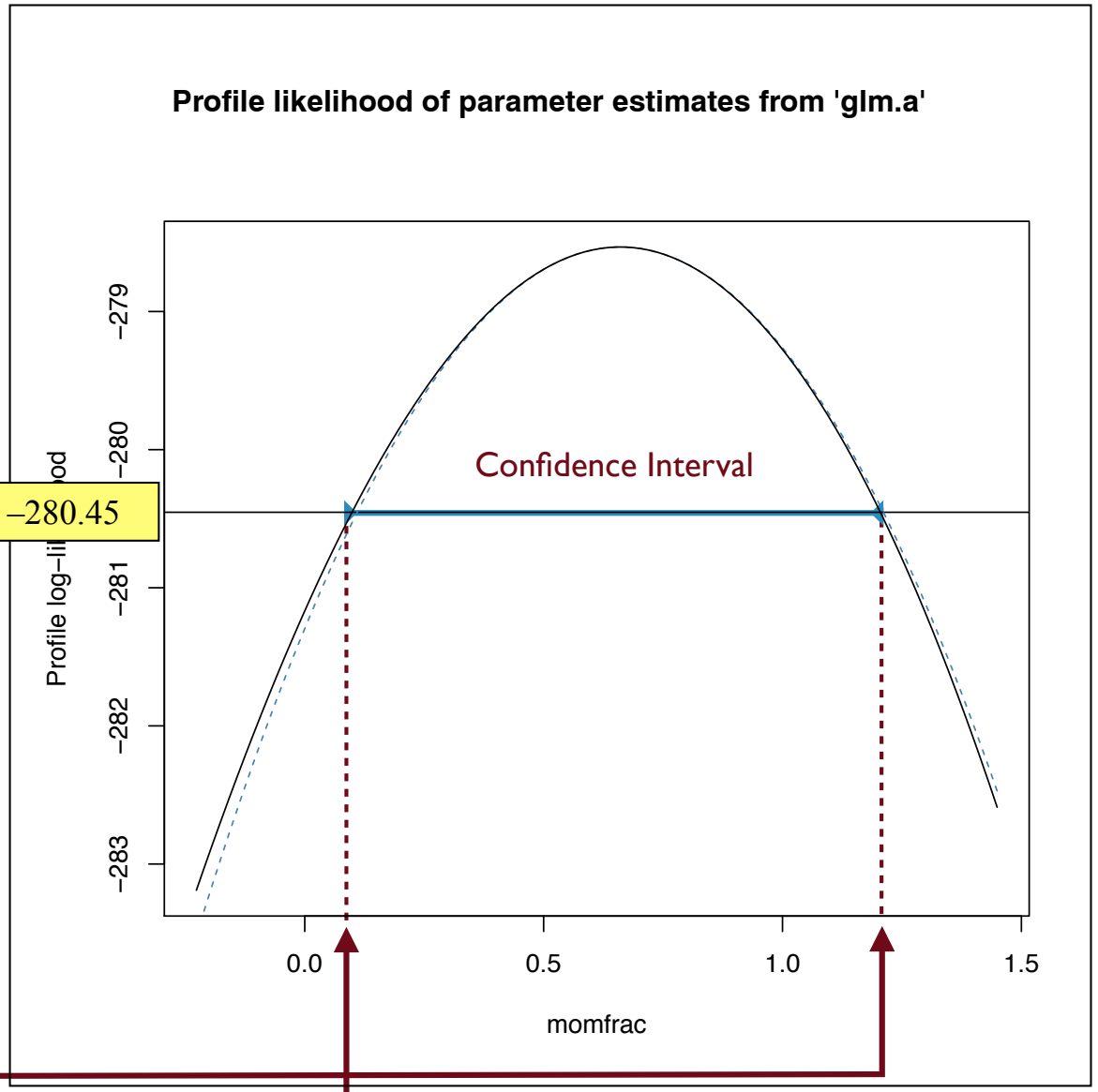
$$\ln\left(\mathcal{L}_{\mathrm{R}}\right) > \ln\left(\mathcal{L}_{\mathrm{F}}\right) - \frac{\chi^2_{0.95}}{2}$$

Parameter values that produce log-likelihood values greater than –280.45 will be included in the interval

```
> logLik(glm.a)
'log Lik.' -278.5327 (df=2)


> qchisq(0.95, df = 1) / 2
[1] 1.920729

> -278.5327 - 1.920729
[1] -280.4534
```

```
> library(binomTools)

> profile.a = profile(glm.a)

> plot(profile.a)
```

```
> confint(glm.a)
Waiting for profiling to be done...
                 2.5 %     97.5 %
(Intercept) -1.42327258 -0.9776359
momfrac      0.09890293  1.2048136
```



Score intervals

Wilson score intervals and psuedo score intervals can also be calculated

- http://www.stat.ufl.edu/~aa/articles/agresti_2011.pdf

Confidence Intervals for Odds

Most statisticians prefer to give confidence limits for the odds ratio rather than the logit. Two methods to produce these interval limits

1. Transform the limits for the logit interval
2. Estimate the SE for the odds directly

$$[e^{0.0985}, e^{1.2225}]$$

$$[1.10, 3.40]$$

Wald confidence limits for the effect on the odds ratio

With 95% confidence the odds of having a fracture if your mother had a hip fracture is between 1.10 and 3.40.

```
> exp(confint(glm.a))

Waiting for profiling to be done...
                2.5 %    97.5 %
(Intercept) 0.2409243 0.3761994
momfrac     1.1039591 3.3361373
```

Profile likelihood confidence limits for the effect on the odds ratio

With 95% confidence the odds of having a fracture if your mother had a hip fracture is between 1.10 and 3.34.

Estimating the SE

Transformations work well for point estimates, but not SE

- SE can be approximated using the delta method

Calculate the variance of the Taylor series of a function based on the first two terms of the expansion

1. Find the Taylor expansion based on two terms

$$f(\mathbf{X}) \approx f(\mathbf{U}) + \nabla f(\mathbf{U})^T \cdot (\mathbf{X} - \mathbf{U})$$

where **U** is the mean vector of **X** = {$x_1, x_2, \ldots, x_n$}

2. Calculate the variance

$$\mathrm{Var}\left(f(\mathbf{X})\right) \approx \nabla f(\mathbf{X})^T \cdot \mathrm{Cov}(\mathbf{X}) \cdot \nabla f(\mathbf{X})$$

where Cov(**X**) is the variance–covariance matrix of **X**

Consider the transformation function *f*(**B**) which transforms the coefficients to logits

$$f(\mathbf{B}) = \exp\left(b_k\right)$$

To calculate the variance we need

1. The derivative of *f*(**B**)
2. The apprpriate term from the variance–covariance matrix of the fitted model

The derivative is relatively easy

$$\frac{\partial f(\mathbf{B})}{\partial b_1} = \exp\left(b_1\right)$$

The variance–covariance matrix can be computed using the `vcov()` function

```
> vcov(glm.a)

             (Intercept)     momfrac
(Intercept)  0.01289498 -0.01289498
momfrac     -0.01289498  0.07895189
```

Now we can compute the variance

$$\text{Var}\left(f(\mathbf{X})\right) \approx \nabla f(\mathbf{X})^T \cdot \text{Cov}(\mathbf{X}) \cdot \nabla f(\mathbf{X})$$

```
> grad = exp(coef(glm.a))[2]

> vc = vcov(glm.a)[2, 2]

> var.b1 = t(grad) %*% vc %*% grad

> se.b1 = sqrt(0.2958472)
> se.b1
[1] 0.5439184

> grad - 2 * se.b1
[1] 0.8479275

> grad + 2 * se.b1
[1] 3.023601
```

We can use the `deltamethod()` function from the **msm** package

```
> library(msm)

> b1 = coef(glm.a)[2]
> vc = vcov(glm.a)[2, 2]

> deltamethod(~exp(x1), b1, vc)
[1] 0.5439184
```

The `deltamethod()` function takes three arguments:

1. A formula representing the transformation expressed using x1, x2, etc.
2. The untransformed coefficient(s)
3. The estimated variance–covariance matrix of **X**

With 95% confidence the odds of having a fracture if your mother had a hip fracture is between 0.85 and 3.02.

CI for $\mu_Y|X$

The CI for μ_Y can be computed at particular values of X

- We use the `predict()` function to compute actual values
- We use `geom_smooth()` to add the intervals to the graphical model

These can be based on the logit scale, the odds scale, or in the probability scale.

Create a data frame with the values of X that you want to find CIs for. Be sure that the variable has the same name as the X's in your fitted model.

```
> ci.data = data.frame(momfrac = c(0, 1))
```

```
> ci.data = data.frame(momfrac = c(0, 1))

> predict(glm.a, newdata = ci.data, type = "link", se.fit = TRUE)

$fit
         1          2
-1.1960205 -0.5355182

$se.fit
        1         2
0.1135561 0.2570154

> -0.5355182 - 2* 0.2570154
-1.049549

> -0.5355182 + 2* 0.2570154
[1] -0.0214874
```

logit scale

With 95% confidence the log-odds of having a fracture if your mother had a hip fracture is between –1.05 and –0.02.

```
> predict(glm.a, newdata = ci.data, type = "response", se.fit = TRUE)

$fit
        1         2
0.2321839 0.3692308

$se.fit
         1          2
0.02024416 0.05985873


> 0.3692308 - 2* 0.05985873
0.2495133

> 0.3692308 + 2* 0.05985873
[1] 0.4889483
```

probability scale

With 95% confidence the probability of having a fracture if your mother had a hip fracture is between 0.25 and 0.49.

Adding a Confidence Envelope to the Graphical Model

This is more appropriate for continuous predictors. We will provide example with the model regressing fracture on age.

Fit a model

```
> glm.b <- glm(fracture ~ age, data = glow,
     family = binomial(link = "logit"))
```

Create a data frame for a range of *X* values

```
> data1 <- data.frame(
     age = seq(from = 55, to = 90, by = 1)
     )
```

Use the `se.fit = TRUE` aand the `type="response"` arguments in the `predict()` function

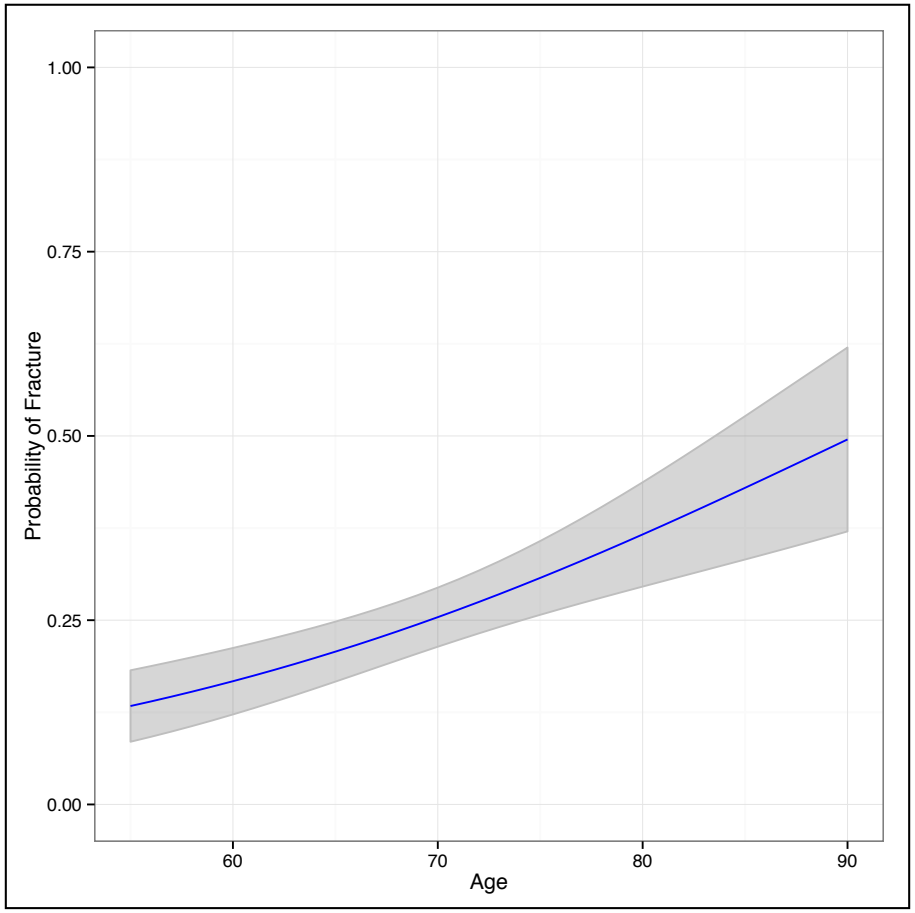
```
> fitted <- predict(glm.b, newdata = data1,
     type = "response", se.fit = TRUE)
```

Add the fitted values and the SEs to the data frame with the *X* values

```
> data1$fit = fitted$fit
> data1$se = fitted$se.fit
```

Compute the lower and upper bounds for the confidence envelope

```
> data1$lowerLimit <- data1$fit - 2 * data1$se
> data1$upperLimit <- data1$fit + 2 * data1$se
```



Use `geom_ribbon()` to draw the confidence envelope

```
> ggplot(data = data1, aes(x = age, y = fit)) +
    geom_ribbon(aes(ymin = lowerLimit, ymax = upperLimit),
          color = "grey", alpha = 0.3) +
    geom_line(color = "blue") +
    xlab("Age") +
    ylab("Probability of Fracture") +
    ylim(c(0, 1)) +
    theme_bw()
```

A shortcut using the original data

```
> ggplot(data = glow, aes(x = age, y = fracture)) +
    geom_smooth(method = "glm", family = binomial(link = "logit")) +
    xlab("Age") +
    ylab("Probability of Fracture") +
    ylim(c(0, 1)) +
    theme_bw()
```

Prediction Intervals

What about **prediction intervals**...this is a somewhat open problem for GLMs. Most people use simulation to produce these intervals, if they produce them at all.

Prediction intervals (i.e., intervals with 95% probability of catching a new observation) are somewhat tricky even to define for GLMs

For normally distributed responses,

$$\hat{y} \pm 2 \times \sqrt{s^2 + SE_{\hat{y}}^2}$$

This "error" combines the model uncertainty in the prediction of the mean with the variability in the observation.

One major problem is that the interval, which is supposed to cover 95% of the potential responses should be included in the interval. But, by adding and subtracting some amount from the fitted value would actually include 0% of responses that are either 0 or 1.

In order to cover responses our interval would have to go from [0, 1] which would have 100% coverage!.

Because of this, people tend to use the CI for making predictions