

Linear & Generalized Linear Models/ Advanced Regression for Independent Data

Andrew J. Spieker, Ph.D.

Associate Professor of Biostatistics
Vanderbilt University

Set 11: Hypothesis testing for GLMs

Version: 07/03/2026

TABLE OF CONTENTS

- 1 Hypothesis testing
- 2 Examples: Logistic regression
- 3 Examples: Gaussian regression
- 4 Inverting score tests

Generally:

- We seek to test hypotheses $H_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{0}$, where $\boldsymbol{\beta}$ denotes the regression parameters of a GLM (you may be thinking about it through the likelihood, quasi-likelihood, or estimating equations framework) and $\mathbf{C} = \mathbf{C}_{Q \times K}$ encodes Q linear hypotheses.
- The goal in this set of notes is to present and discuss different hypothesis testing methods. You'll already be familiar with many of the ideas. The three major classes of tests we will discuss are:
 - ▶ Likelihood ratio tests.
 - ▶ Score-based tests (also called Rao tests).
 - ▶ Wald-based tests.
- In correctly specified models, and under suitable regularity conditions, the three are asymptotically equivalent.

For GLMs: Wald statistic

- The following is a form for the Wald statistic (we've encountered it before):

$$X_W^2 = (\mathbf{C}\hat{\boldsymbol{\beta}})^\top \left(\mathbf{C}\widehat{\text{Cov}}[\hat{\boldsymbol{\beta}}]\mathbf{C}^\top \right)^{-1} (\mathbf{C}\hat{\boldsymbol{\beta}}).$$

- $\widehat{\text{Cov}}[\hat{\boldsymbol{\beta}}]$ is any reasonable and appropriate estimator (model-based, sandwich, quasi-likelihood, bootstrap-based).
- Under H_0 , we have $X_W^2 \xrightarrow{d} \chi_Q^2$ (I state this without proof; we have already done a rigorous treatment of hypothesis testing for linear regression, so I don't wish to belabor the point).

For GLMs: Score statistic

- Let $\hat{\boldsymbol{\beta}}^0$ denote the estimate under H_0 .
 - ▶ Subject to the constraint $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$.
- The score takes the form $\mathbb{S}_N(\boldsymbol{\beta}, \phi) = \mathbf{D}^\top \mathbf{V}^{-1}(\mathbf{y} - \boldsymbol{\mu})/\phi$.
- The information takes the form $\mathcal{I}_N(\boldsymbol{\beta}, \phi) = \mathbb{A}_N(\boldsymbol{\beta})/\phi$.
- The following is a form for the score statistic:

$$X_S^2 = (\mathbb{S}_N(\hat{\boldsymbol{\beta}}^0, \hat{\phi}))^\top \left(\mathcal{I}_N(\hat{\boldsymbol{\beta}}^0, \hat{\phi}) \right)^{-1} (\mathbb{S}_N(\hat{\boldsymbol{\beta}}^0, \hat{\phi})).$$

- Importantly, $\hat{\phi}$ denotes the estimate **under the full model**.
- This test is model-based, although use of the *observed* information may offer some robustness under certain circumstances.
 - ▶ This is a rabbit hole I'm electing not to go down.
- Under H_0 , we have $X_S^2 \xrightarrow{d} \chi_Q^2$.

For GLMs: Likelihood ratio test statistic

- Let $\hat{\boldsymbol{\beta}}^0$ denote the estimate under H_0 .
 - ▶ Subject to the constraint $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$.
- The following is a form for the likelihood ratio test statistic:

$$X_{LR}^2 = -2 \left(\ell(\hat{\boldsymbol{\beta}}^0, \hat{\phi}) - \ell(\hat{\boldsymbol{\beta}}, \hat{\phi}) \right)$$

- Importantly, $\hat{\phi}$ denotes the estimate **under the full model**.
- This test is model-based.
- Under H_0 , we have $X_{LR}^2 \xrightarrow{d} \chi_Q^2$.

TABLE OF CONTENTS

- 1 Hypothesis testing
- 2 Examples: Logistic regression
- 3 Examples: Gaussian regression
- 4 Inverting score tests

Model setup: For simulation

- Exposure: $X = 0, 1, 2$, each with probability $1/3$.
- Outcome: $Y \sim \text{Bernoulli}(p = \text{expit}(\beta_0 + \beta_1 1(x = 1) + \beta_2 1(x = 2)))$.
- Let $\beta_0 = -1$, $\beta_1 = 0.4$ and $\beta_2 = 0.6$.
- Let $N = 150$.
- Consider the following hypothesis tests:
 - 1 $H_0 : \beta_1 = 0$ (groups 0 and 1 the same).
 - 2 $H_0 : \beta_2 - \beta_1 = 0$ (groups 1 and 2 the same).
 - 3 $H_0 : \beta_1 = \beta_2 = 0$ (all three groups the same).

Example 11.1: Generate data

```
1 ## Set seed and generate data
2 set.seed(7345)
3 n <- 150
4 beta <- matrix(c(-1, 0.4, 0.6), nrow = 3)
5 x <- sample(c(0,1,2), size = n, replace = TRUE)
6 X <- matrix(cbind(1, as.numeric(x==1), as.numeric(x==2)), ncol = 3)
7 y <- rbinom(n, 1, plogis(X %*% beta))
```

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Fit GLM (unrestricted)

```
1 ## Initialize and run GLM
2 betaj <- c(log(sum(y)/(n - sum(y))), 0, 0)
3 tol <- 1
4 iter <- 1
5
6 while(tol > 1e-15 & iter < 50)
7 {
8   betaj.prior <- betaj
9   etaj <- c(X %*% betaj)
10  Gn <- t(X) %*% (y - plogis(etaj))
11  An <- t(X * (plogis(etaj)*(1 - plogis(etaj)))) %*% X
12  betaj <- betaj + solve(An) %*% Gn
13  tol <- sum((betaj - betaj.prior)^2)
14  iter <- iter + 1
15 }
16 bhat <- betaj
17 An <- t(X * (plogis(c(X %*% bhat))*(1 - plogis(c(X %*% bhat))))) %*% X
18 Vhat <- solve(An)
19
20 > c(bhat)
21 [1] -1.0726368  0.2841794  0.7159619
```

Test 1: $H_0 : \beta_1 = 0$ (groups 0 and 1 the same).

- Model: $Y \sim \text{Bernoulli}(p = \text{expit}(\beta_0 + \beta_1 1(x = 1) + \beta_2 1(x = 2)))$.
- Constrained model: $Y \sim \text{Bernoulli}(p = \text{expit}(\beta_0 + \beta_2 1(x = 2)))$.

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Fit GLM (constrained)

```
1 X0 <- cbind(X[,1], X[,3])
2 betaj <- c(log(sum(y)/(n - sum(y))), 0)
3 tol <- 1
4 iter <- 1
5
6 while(tol > 1e-15 & iter < 50)
7 {
8   betaj.prior <- betaj
9   etaj <- c(X0 %*% betaj)
10  Gn <- t(X0) %*% (y - plogis(etaj))
11  An <- t(X0 * (plogis(etaj)*(1 - plogis(etaj)))) %*% X0
12  betaj <- betaj + solve(An) %*% Gn
13  tol <- sum((betaj - betaj.prior)^2)
14  iter <- iter + 1
15 }
16
17 bhat.0 <- cbind(c(betaj[1], 0, betaj[2]))
18
19 > c(bhat.0)
20 [1] -0.9304754  0.0000000  0.5738004
```

Example 11.1: Likelihood ratio test

```
1 loglik <- sum(log(dbinom(y, size = 1, prob = plogis(X %*% bhat))))
2 loglik.0 <- sum(log(dbinom(y, size = 1, prob = plogis(X %*% bhat.0))))
3 LR <- -2*(loglik.0 - loglik)
4 pLR <- 1 - pchisq(LR, df = 1)
```

Example 11.1: Score test

```
1 muhat.0 <- plogis(X %*% bhat.0)
2 Gn0 <- t(X) %*% (y - muhat.0)
3 An0 <- t(X * c(muhat.0*(1 - muhat.0))) %*% X
4 S <- t(Gn0) %*% solve(An0) %*% Gn0
5 pS <- 1 - pchisq(S, df = 1)
```

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Wald test

```
1 C <- matrix(0, nrow = 1, ncol = 3)
2 C[1,2] <- 1
3
4 > C
5      [,1] [,2] [,3]
6 [1,]    0    1    0
7
8
9 W <- t(C %*% bhat) %*% solve(C %*% Vhat %*% t(C)) %*% (C %*%
   bhat)
10 pW <- 1 - pchisq(W, df = 1)
```

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Report and compare to ANOVA function

```
1 > pLR
2 [1] 0.5247948
3
4 > pS
5           [,1]
6 [1,] 0.5248092
7
8 > pW
9           [,1]
10 [1,] 0.5253544
11
12 zz.Full <- glm(y ~ X[,2] + X[,3], family = binomial, control = list(epsilon = 1e-15))
13 zz.Reduced <- glm(y ~ X[,3], family = binomial, control = list(epsilon = 1e-15))
14
15 ## Exact match! :)
16 > anova(zz.Full, zz.Reduced, test = "LRT")$Pr[2]
17 [1] 0.5247948
18
19 ## Exact match! :)
20 > anova(zz.Full, zz.Reduced, test = "Rao")$Pr[2]
21 [1] 0.5248092
22
23 ## Not aware of a Wald-based test from anova() function
```

Test 2: $H_0 : \beta_2 - \beta_1 = 0$ (groups 1 and 2 the same).

- Model: $Y \sim \text{Bernoulli}(p = \text{expit}(\beta_0 + \beta_1 1(x = 1) + \beta_2 1(x = 2)))$.
- Constrained model: $Y \sim \text{Bernoulli}(p = \text{expit}(\beta_0 + \beta_1 1(x > 0)))$.

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Fit GLM (constrained)

```
1 X0 <- cbind(X[,1], X[,2] + X[,3])
2 betaj <- c(log(sum(y)/(n - sum(y))), 0)
3 tol <- 1
4 iter <- 1
5
6 while(tol > 1e-15 & iter < 50)
7 {
8   betaj.prior <- betaj
9   etaj <- c(X0 %*% betaj)
10  Gn <- t(X0) %*% (y - plogis(etaj))
11  An <- t(X0 * plogis(etaj)*(1 - plogis(etaj))) %*% X0
12  betaj <- betaj + solve(An) %*% Gn
13  tol <- sum((betaj - betaj.prior)^2)
14  iter <- iter + 1
15 }
16
17 bhat.0 <- cbind(c(betaj[1], betaj[2], betaj[2]))
18
19 > c(bhat.0)
20 [1] -1.072637  0.513021  0.513021
```

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Wald test

```
1 C <- matrix(0, nrow = 1, ncol = 3)
2 C[1,2] <- -1
3 C[1,3] <- 1
4
5 > C
6      [,1] [,2] [,3]
7 [1,]    0  -1    1
8
9
10 W <- t(C %*% bhat) %*% solve(C %*% Vhat %*% t(C)) %*% (C %*%
    bhat)
11 pW <- 1 - pchisq(W, df = 1)
```

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Report and compare to ANOVA function

```
1 ## The likelihood ratio test has the same code as previously shown
2 > pLR
3 [1] 0.3039759
4
5 ## The score test has the same code as previously shown
6 > pS
7           [,1]
8 [1,] 0.3048367
9
10 > pW
11           [,1]
12 [1,] 0.3060022
13
14 zz.Full <- glm(y ~ X[,2] + X[,3], family = binomial, control = list(epsilon = 1e-15))
15 zz.Reduced <- glm(y ~ I(X[,2] + X[,3]), family = binomial, control = list(epsilon = 1e-15))
16
17 ## Exact match! :)
18 > anova(zz.Full, zz.Reduced, test = "LRT")$Pr[2]
19 [1] 0.3039759
20
21 ## Exact match! :)
22 > anova(zz.Full, zz.Reduced, test = "Rao")$Pr[2]
23 [1] 0.3048367
```

Test 3: $H_0 : \beta_1 = \beta_2 = 0$ (all groups the same).

- Model: $Y \sim \text{Bernoulli}(p = \text{expit}(\beta_0 + \beta_1 1(x = 1) + \beta_2 1(x = 2)))$.
- Constrained model: $Y \sim \text{Bernoulli}(p = \text{expit}(\beta_0))$.

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Fit GLM (constrained)

```
1 X0 <- cbind(X[,1])
2 betaj <- c(log(sum(y)/(n - sum(y))))
3 tol <- 1
4 iter <- 1
5
6 while(tol > 1e-15 & iter < 50)
7 {
8   betaj.prior <- betaj
9   etaj <- c(X0 %*% betaj)
10  Gn <- t(X0) %*% (y - plogis(etaj))
11  An <- t(X0 * plogis(etaj)*(1 - plogis(etaj))) %*% X0
12  betaj <- betaj + solve(An) %*% Gn
13  tol <- sum((betaj - betaj.prior)^2)
14  iter <- iter + 1
15 }
16
17 bhat.0 <- cbind(c(betaj[1], 0, 0))
18
19 > c(bhat.0)
20 [1] -0.7233002  0.0000000  0.0000000
21
22 ## This iterative code was completely unnecessary. Why? :)
```

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Wald test

```
1 C <- matrix(0, nrow = 2, ncol = 3)
2 C[1,2] <- 1
3 C[2,3] <- 1
4
5 > C
6      [,1] [,2] [,3]
7 [1,]    0    1    0
8 [2,]    0    0    1
9
10 W <- t(C %*% bhat) %*% solve(C %*% Vhat %*% t(C)) %*% (C %*%
    bhat)
11 pW <- 1 - pchisq(W, df = 2)
```

EXAMPLES: LOGISTIC REGRESSION

Example 11.1: Report and compare to ANOVA function

```
1 ## The likelihood ratio test has the same code as previously shown (except df=2)
2 > pLR
3 [1] 0.2336163
4
5 ## The score test has the same code as previously shown (except df=2)
6 > pS
7           [,1]
8 [1,] 0.232581
9
10 > pW
11           [,1]
12 [1,] 0.2370055
13
14 zz.Full <- glm(y ~ X[,2] + X[,3], family = binomial, control = list(epsilon = 1e-15))
15 zz.Reduced <- glm(y ~ 1, family = binomial, control = list(epsilon = 1e-15))
16
17 ## Exact match! :)
18 > anova(zz.Full, zz.Reduced, test = "LRT")$Pr[2]
19 [1] 0.2336163
20
21 ## Exact match! :)
22 > anova(zz.Full, zz.Reduced, test = "Rao")$Pr[2]
23 [1] 0.232581
```

EXAMPLES: LOGISTIC REGRESSION

Notes:

- In this example, all tests should have been valid because the model was correctly specified.
- This was exemplified (though certainly not proven) by the fact that the resulting p-values were so similar.
- Let's try an example where the model is *not* fully correct, in which case we might expect to see some differences.

TABLE OF CONTENTS

- 1 Hypothesis testing
- 2 Examples: Logistic regression
- 3 Examples: Gaussian regression
- 4 Inverting score tests

Example 11.2: Model setup for simulation

- Exposure: $X \sim \text{Uniform}(1, 10)$.
- Outcome: $Y_i \sim \mathcal{N}(\beta_0 + \beta_1 x_i + \beta_2 x_i^2, \sigma_i^2 = 0.25 x_i^2)$.
- Let $\beta_0 = 50$, $\beta_1 = 0$ and $\beta_2 = 0$.
- Let $N = 100$.
- Consider the hypothesis test: $H_0 : \beta_1 = \beta_2 = 0$
 - ▶ No quadratic association between X and mean Y .
- OLS: correctly specified mean model, but model-based variance wrong.
- Consider LR test, score test, Wald test (model-based variance), and Wald test (sandwich-based variance).

Example 11.2: Generate data

```
1 set.seed(7345)
2 n <- 1000
3 beta <- matrix(c(50, 0, 0), nrow = 3)
4 x <- runif(n, 1, 10)
5 X <- matrix(cbind(1, x, x^2), ncol = 3)
6 y <- rnorm(n, X %*% beta, x/2)
```

Example 11.2: Gaussian GLM

```
1 ## Gaussian GLM
2 bhat <- solve(t(X) %*% X) %*% t(X) %*% y
3 phi.hat <- t(y - X %*% bhat) %*% (y - X %*% bhat) / (n - 3)
4 An <- t(X) %*% X
5 Bn <- t(X * c(y - X %*% bhat)^2) %*% X
```

Example 11.2: Variance estimators (unrestricted)

```
1 ## Model-based variance
2 V1 <- solve(An) * as.numeric(phi.hat)
3
4 ## Sandwich variance
5 V2 <- solve(An) %*% Bn %*% solve(An)
```

Example 11.2: Variance estimators (unrestricted)

```
1 ## Restricted model
2 X0 <- X[,c(-2,-3)]
3 bhat.0 <- solve(t(X0) %*% X0) %*% t(X0) %*% y
4 bhat.0 <- cbind(c(bhat.0, 0, 0))
```

Example 11.2: Likelihood ratio test

```
1 ## Likelihood ratio test
2 loglik <- sum(log(dnorm(y, X %*% bhat, sqrt(phi.hat))))
3 loglik.0 <- sum(log(dnorm(y, X %*% bhat.0, sqrt(phi.hat))))
4 LR <- -2*(loglik.0 - loglik)
5 pLR <- 1 - pchisq(LR, df = 2)
```

Example 11.2: Score test

```
1 ## Score test
2 Gn0 <- t(X) %*% (y - X %*% bhat.0)/as.numeric(phi.hat)
3 An0 <- t(X) %*% X/as.numeric(phi.hat)
4 S <- t(Gn0) %*% solve(An0) %*% Gn0
5 pS <- 1 - pchisq(S, df = 2)
```

EXAMPLES: GAUSSIAN REGRESSION

Example 11.2: Wald tests

```
1 ## Wald tests
2 C <- matrix(0, nrow = 2, ncol = 3)
3 C[1,2] <- 1
4 C[2,3] <- 1
5
6 ## Model-based Wald test
7 W1 <- t(C %*% bhat) %*% solve(C %*% V1 %*% t(C)) %*% (C %*% bhat)
8 pW1 <- 1 - pchisq(W1, df = 2)
9
10 ## Sandwich-based Wald test
11 W2 <- t(C %*% bhat) %*% solve(C %*% V2 %*% t(C)) %*% (C %*% bhat)
12 pW2 <- 1 - pchisq(W2, df = 2)
```

EXAMPLES: GAUSSIAN REGRESSION

Example 11.2: Report and compare to ANOVA function

```
1 > pLR
2 [1] 0.05750231
3
4 > pS
5           [,1]
6 [1,] 0.05750231
7
8 > pW1
9           [,1]
10 [1,] 0.05750231
11
12 > pW2
13           [,1]
14 [1,] 0.1752994
15
16 zz.Full <- glm(y ~ X[,2] + X[,3], family = gaussian)
17 zz.Reduced <- glm(y ~ 1, family = gaussian)
18
19 ## Exact match! :)
20 > anova(zz.Full, zz.Reduced, test = "LRT")$Pr[2]
21 [1] 0.05750231
22
23 ## Exact match! :)
24 > anova(zz.Full, zz.Reduced, test = "Rao")$Pr[2]
25 [1] 0.05750231
```

Discussion questions:

- Which p-values (if any) are valid for the test of interest?
- Can the likelihood ratio and score tests be modified to be valid?

Robustness of sandwich-based inference:

- Continuing from this setup, we can illustrate by simulation that the model-based and sandwich-based inference produce different results.

Example 11.2: Simulation of Wald-based tests

```
1 ## Set seed for reproducibility
2 set.seed(7345)
3
4 ## Sample size
5 n <- 1000
6
7 ## Coefficient
8 beta <- matrix(c(50, 0, 0), nrow = 3)
9
10 ## Constraint matrix
11 C <- matrix(0, nrow = 2, ncol = 3)
12 C[1,2] <- 1
13 C[2,3] <- 1
14
15 ## Number of simulations
16 nsim <- 50000
17
18 ## Space to store results
19 results <- matrix(0, nrow = nsim, ncol = 2)
```

Example 11.2: Simulation of Wald-based tests

```
1 for (j in 1:nsim) {
2   ## Predictor
3   x <- runif(n, 1, 10)
4
5   ## Design matrix
6   X <- matrix(cbind(1, x, x^2), ncol = 3)
7
8   ## Outcome
9   y <- rnorm(n, X %*% beta, x/2)
10
11  ## Gaussian GLM
12  bhat <- solve(t(X) %*% X) %*% t(X) %*% y
13  phi.hat <- t(y - X %*% bhat) %*% (y - X %*% bhat) / (n - 3)
14
15  ## Bread
16  An <- t(X) %*% X
17
18  ## Meat
19  Bn <- t(X * c(y - X %*% bhat)^2) %*% X
```

EXAMPLES: GAUSSIAN REGRESSION

Example 11.2: Simulation of Wald-based tests

```
1  ## Model-based variance
2  V1 <- solve(An) * as.numeric(phi.hat)
3
4  ## Sandwich variance
5  V2 <- solve(An) %*% Bn %*% solve(An)
6
7  ## Model-based Wald test
8  W1 <- t(C %*% bhat) %*% solve(C %*% V1 %*% t(C)) %*% (C %*% bhat)
9  pW1 <- 1 - pchisq(W1, df = 2)
10
11 ## Sandwich-based Wald test
12 W2 <- t(C %*% bhat) %*% solve(C %*% V2 %*% t(C)) %*% (C %*% bhat)
13 pW2 <- 1 - pchisq(W2, df = 2)
14
15 ## Extract results
16 results[j,1] <- as.numeric(pW1 < 0.05)
17 results[j,2] <- as.numeric(pW2 < 0.05)
18
19 ## Track progress
20 if (round(j/2000) == (j/2000)) {print(paste(j, "iterations complete!"))}
21 }
22
23 ## How do these results compare in terms of size?
24
25 # > colMeans(results)
26 # [1] 0.08236 0.05250
27
28 ## There we have it...this is a case of "wrong model, wrong answer."
```

TABLE OF CONTENTS

- 1 Hypothesis testing
- 2 Examples: Logistic regression
- 3 Examples: Gaussian regression
- 4 Inverting score tests

Background:

- The most widely used approach to form a confidence interval for a regression parameter is to use the observed standard error:

$$\hat{\beta}_k \pm z_{1-\alpha/2} \widehat{SE}[\hat{\beta}_k].$$

(or, similarly, we could use $t_{df, 1-\alpha/2}$)

- This “inverts the Wald test,” in the sense that the confidence interval encompasses all values, c , such that the null hypotheses $H_0 : \beta_k = c$ would not be rejected by a two-sided α -level Wald test.
- What if we wanted to invert, say, the score test instead?
 - ▶ I am able to identify two GLMs in which it is “straightforward enough” to do so: Gaussian and Poisson regression. There may be others.

Example 11.3: Gaussian regression

- Model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$, $E[\epsilon] = 0$ and $\text{Var}[\epsilon] = \sigma^2$.
- Suppose we use OLS to estimate β (i.e., Gaussian regression with identity link) and seek to form a 95% confidence interval for β_1 by inverting the score test rather than the Wald test.
- Under $H_0 : \beta_1 = c$, the null model is given as:

$$\begin{aligned} E[Y|X_1, X_2] &= \beta_0 + cX_1 + \beta_2 X_2 \\ \Rightarrow E[Y - cX_1|X_2] &= \beta_0 + \beta_2 X_2. \end{aligned}$$

- This suggests that we can regress $Y - cX_1$ on X_2 to obtain estimates of β_0 and β_2 under H_0 (call the null estimate $\hat{\beta}^0 = (\hat{\beta}_0, c, \hat{\beta}_2)$).
- Compute p-value, p_c associated with score statistic, and search for all c such that $p_c > 0.05$.
- Nice example to illustrate the concept, but “unnecessary” in that Wald and score tests are equivalent for Gaussian regression.

Example 11.4: Poisson regression

- Model: $Y \sim \text{Poisson}(\lambda = \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2))$.
- Suppose we seek to form a 95% confidence interval for β_1 by inverting the score test rather than the Wald test.
- Under $H_0 : \beta_1 = c$, the estimating equations are given as:

$$\begin{aligned}\mathbf{X}^T(\mathbf{y} - \text{vec}(\exp(\beta_0 + cX_{1i} + \beta_2 X_{2i}))) &= \mathbf{0} \\ \Rightarrow \mathbf{X}^T(\mathbf{y} - \text{diag}(\exp(cX_{1i}))\text{vec}(\exp(\beta_0 + \beta_2 X_{2i}))) &= \mathbf{0}.\end{aligned}$$

- This suggests that we can include cX_1 as an offset in the Poisson model to obtain estimates of β_0 and β_2 under H_0 (call the null estimate $\hat{\boldsymbol{\beta}}^0 = (\hat{\beta}_0, c, \hat{\beta}_2)$).
- Compute p-value, p_c associated with score statistic, and search for all c such that $p_c > 0.05$.

INVERTING SCORE TESTS

Example 11.4: Poisson regression

```
1 ## Set seed for reproducibility
2 set.seed(7345)
3
4 ## Sample size of 100
5 n <- 100
6
7 ## Generate exposure
8 X <- runif(n, 1, 5)
9
10 ## Generate outcome
11 Y <- rpois(n, lambda = exp(1 + 0.1*X))
12
13 ## Fit Poisson model
14 zz <- glm(Y ~ X, family = poisson)
15
16 ## Wald-based confidence interval
17 > confint(zz)[2,]
18      2.5 %      97.5 %
19 0.04481821 0.22128675
```

INVERTING SCORE TESTS

Example 11.4: Poisson regression

```
1 ## Wald-based confidence interval is a good starting point
2 > confint(zz)[2,]
3      2.5 %      97.5 %
4 0.04481821 0.22128675
5
6 ## p-value under offset given by left-hand side
7 zz <- glm(Y ~ X, offset = 0.04481821*X, family = poisson)
8 > summary(zz)$coefficients[2,4]
9 [1] 0.0518257
10
11 ## p-value under offset given by right-hand side
12 zz <- glm(Y ~ X, offset = 0.2212868*X, family = poisson)
13 > summary(zz)$coefficients[2,4]
14 [1] 0.04794482
15
16 ## This is from a manual search (not fun), though one could be programmed
17 zz <- glm(Y ~ X, offset = 0.044126034*X, family = poisson)
18 > summary(zz)$coefficients[2,4]
19 [1] 0.05
20
21 zz <- glm(Y ~ X, offset = 0.220481791*X, family = poisson)
22 > summary(zz)$coefficients[2,4]
23 [1] 0.05
24
25 ## 95% CI obtained by inverting score test
26 ## [0.044126034, 0.220481791]
```

Further commentary:

- You'll sometimes see confidence intervals formed by inverting other kinds of tests (e.g., permutation tests).
- This can be controversial, especially if the emerging interval is not expected to have proper coverage of a target parameter.
- Some say testing and estimation should be thought of separately.
- Perhaps a compromise would be to refer to these special intervals as “inversion” intervals, and then have a conversation about settings under which an inversion interval and a confidence interval are equivalent or share similar properties.
 - ▶ Recall: there is no reason to believe that a “1/20” likelihood support interval will have 95% coverage.

This unit:

- Score tests.
- Wald tests.
- Likelihood ratio tests.

SUMMARY: SO FAR

- Random vectors and matrices; multivariate normal theory.
- Ordinary least squares.
- Hypothesis testing and ANOVA.
- Weighted least squares.
- Confidence regions and prediction.
- Diagnostics.
- Exponential families.
- Generalized linear models.
- Sandwich and bootstrap.
- Quasi-likelihood.
- Hypothesis testing for GLMs.

SUMMARY: COMING UP

- Diagnostics for GLMs.
- Further considerations for binary outcomes.
- Nonlinear least squares.