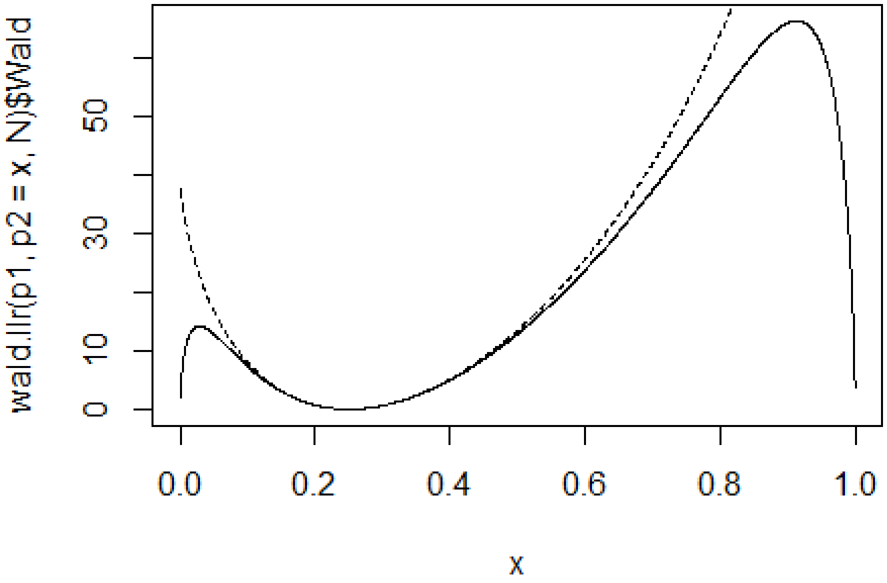
Lecture 4: Wald, Score and Likelihood Ratio Tests

Hauck-Donner effect

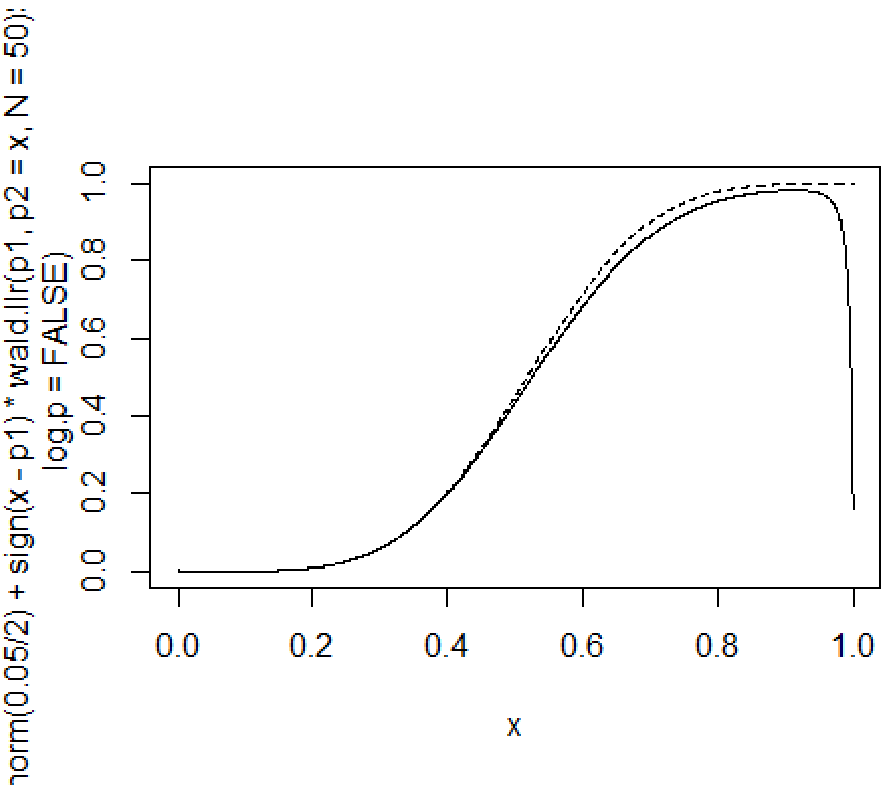
```
N <- 200
p1 <- .25
p2 <- .45

beta1 <- log(p2/(1-p2))-log(p1/(1-p1))
se.beta1 <- sqrt(1/(N/2*p1*(1-p1)) + 1/(N/2*p2*(1-
p2)))#sqrt(1/(p1*p2*N)+1/((1-p1)*p2*N)+1/((1-p2)*(1-p1)*N)+1/((1-p2)*p1*N))

beta1

## [1] 0.8979416

beta1^2/se.beta1^2

## [1] 8.601682

l1 <- N*(p1*log(p1)+(1-p1)*log(1-p1)+p2*log(p2)+(1-p2)*log(1-p2))/2
pbar <- (p1+p2)/2
l0 <- N*(pbar*log(pbar)+(1-pbar)*log(1-pbar))

-2*(l0-l1)

## [1] 8.883864

wald.llr <- function(p1,p2,N) {
  beta1 <- log(p2/(1-p2))-log(p1/(1-p1))
  se.beta1 <- sqrt(1/(N/2*p1*(1-p1)) + 1/(N/2*p2*(1-
p2)))#sqrt(1/(p1*p2*N)+1/((1-p1)*p2*N)+1/((1-p2)*(1-p1)*N)+1/((1-p2)*p1*N))

  beta1
  W <- beta1^2/se.beta1^2

  l1 <- N*(p1*log(p1)+(1-p1)*log(1-p1)+p2*log(p2)+(1-p2)*log(1-p2))/2
  pbar <- (p1+p2)/2
  l0 <- N*(pbar*log(pbar)+(1-pbar)*log(1-pbar))

  LLR <- -2*(l0-l1)

  list(Wald.stat=W,LR.stat=LLR)
}

wald.llr(p1=.5,p2=.6,N=200)

## $Wald.stat
## [1] 2.013085
```

```
##
## $LR.stat
## [1] 2.023756
```

```
# test statistics
curve(wald.llr(p1,p2=x,N)$Wald,n=2e3)
curve(wald.llr(p1,p2=x,N)$LR,add=TRUE,lty=2,n=2e3)
```



```
# power curves
curve(pnorm(qnorm(.05/2)+sign(x-
p1)*wald.llr(p1,p2=x,N=50)$Wald^.5,log.p=FALSE),n=2e3,ylim=c(0,1))
curve(pnorm(qnorm(.05/2)+sign(x-
p1)*wald.llr(p1,p2=x,N=50)$LR^.5,log.p=FALSE),add=TRUE,lty=2,n=2e3)
```



Smoking revisited

This example will again look at the smoking data set from the previous lecture.

We can use our user-defined likelihood function to calculate LR and score test statistics.

```
mod1 <- smoke.logitmod
mod0 <- glm(cbind(y,n-y) ~ 1, # intercept only
            data=smoke,family=binomial)

smoke.ll1 <-
logit.llikfun(coef(mod1),y=smoke$y,n=smoke$n,xmat=as.matrix(smoke$passive,nco
l=1))
smoke.ll0 <-
logit.llikfun(c(coef(mod0),0),y=smoke$y,n=smoke$n,xmat=as.matrix(smoke$passiv
e,ncol=1))

# Wald statistic (from model output)
(coef(mod1)[2]/sqrt(diag(vcov(mod1)))[2])^2

##  passive
## 14.92624

# likelihood ratio statistic
2*(smoke.ll1-smoke.ll0)

## [1] 15.04092
```

```
# score statistic
U <- grad(function(theta)
logit.llikfun(theta,y=smoke$y,n=smoke$n,xmat=as.matrix(smoke$passive,ncol=1))
,
          x=c(mod0$coef,0))
imat <- -hessian(function(theta)
logit.llikfun(theta,y=smoke$y,n=smoke$n,xmat=as.matrix(smoke$passive,ncol=1))
,
          x=c(mod0$coef,0))
U %*% solve(imat) %*% U

##          [,1]
## [1,] 15.00217
```

For the score statistic, we have to evaluate the score function and information matrix of the full model at the parameter estimates under the null model.

AIDS and AZT

This example is from Agresti (page. 184). It comes from a study looking at when to administer AZT and its effect on development of AIDS symptoms in a sample of 338 veterans.

```
azt <- data.frame(race=c(rep('White',2),rep('Black',2)),
                  azt.use=c('Yes','No','Yes','No'),
                  symptoms.yes=c(14,32,11,12),
                  symptoms.no=c(93,81,52,43))

azt$pts <- azt$symptoms.no+azt$symptoms.yes

aids.fit1 <-
glm(cbind(symptoms.yes,symptoms.no)~azt.use+race,data=azt,family=binomial,x=T
RUE)
summary(aids.fit1)

##
## Call:
## glm(formula = cbind(symptoms.yes, symptoms.no) ~ azt.use + race,
##     family = binomial, data = azt, x = TRUE)
##
## Deviance Residuals:
##       1        2        3        4
## -0.5547   0.4253   0.7035  -0.6326
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.07357    0.26294  -4.083 4.45e-05 ***
## azt.useYes  -0.71946    0.27898  -2.579  0.00991 **
## raceWhite    0.05548    0.28861   0.192  0.84755
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8.3499  on 3  degrees of freedom
## Residual deviance: 1.3835  on 1  degrees of freedom
## AIC: 24.86
##
## Number of Fisher Scoring iterations: 4
```

```
cbind(aids.fit1$x,aids.fit1$fitted.values)
```

```
##   (Intercept) azt.useYes raceWhite
## 1           1          1         1 0.1496245
## 2           1          0         1 0.2653998
## 3           1          1         0 0.1427012
## 4           1          0         0 0.2547241
```