Challenger disaster

Heejung Shim

install.package faraway

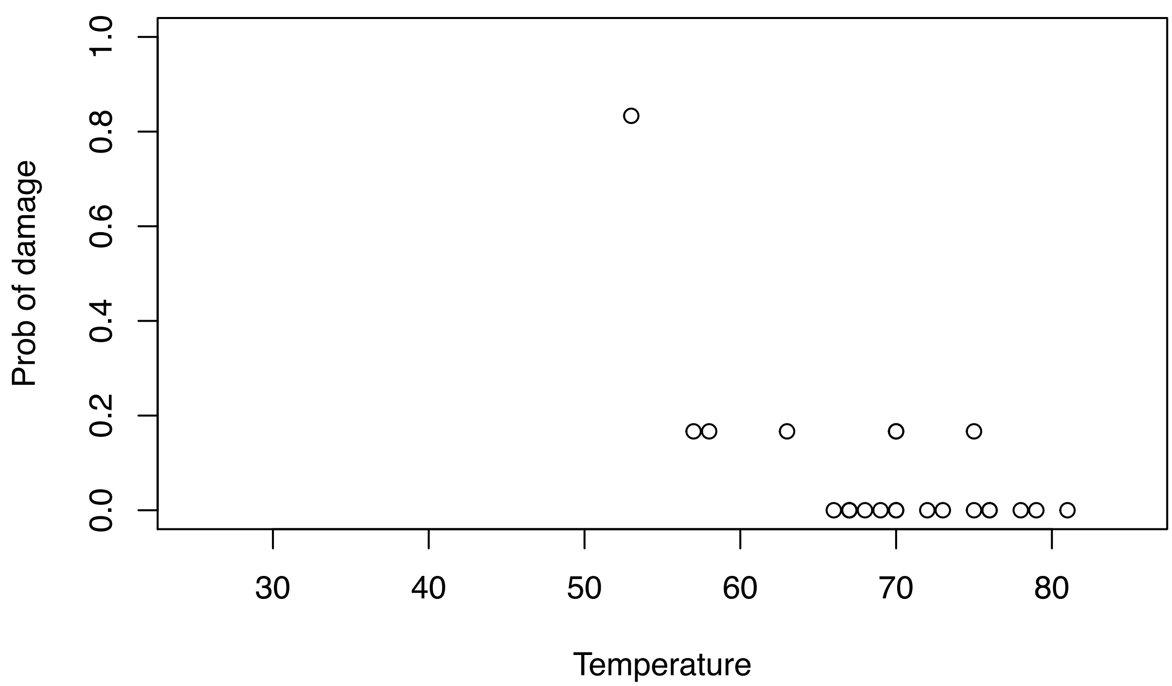
```
#install.packages("faraway")
```

load and plot data

```
library(faraway)
data(orings)
str(orings)
```

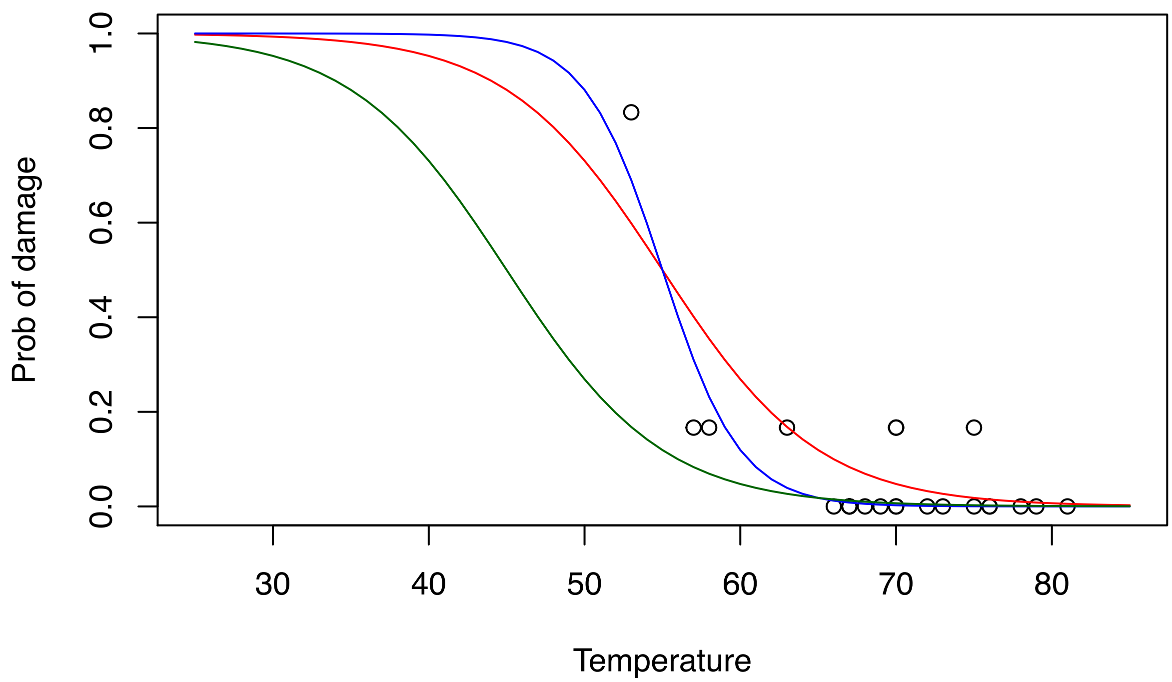
```
## 'data.frame':    23 obs. of  2 variables:
##  $ temp  : num  53 57 58 63 66 67 67 67 68 69 ...
##  $ damage: num  5 1 1 1 0 0 0 0 0 0 ...
```

```
plot(damage/6 ~ temp, orings, xlim=c(25,85), ylim=c(0,1),
     xlab="Temperature", ylab="Prob of damage")
```



logistic function with different values for beta0 and beta1

```
try <- function(a, b, col = "red") {
  t <- seq(25, 85, 1)
  p <- 1/(1 + exp(-a - b*t))
  lines(t, p, col = col)
}
plot(damage/6 ~ temp, orings, xlim=c(25,85), ylim=c(0,1),
     xlab="Temperature", ylab="Prob of damage")
try(11, -0.2, col="red")
## Compared to red curve: same location, stronger steepness
try(22, -0.4, col="blue")
## Compared to red curve: shifted location, same steepness
try(9, -0.2, col="darkgreen")
```



maximum likelihood fitting

$$n\left\{\begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix}\right. \times \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \end{bmatrix}$$

Define the log likelihood

```
logL <- function(beta, orings) {
  eta <- cbind(1, orings$temp) %*% beta
  return( sum( orings$damage*eta - 6*log(1 + exp(eta)) ) )
}
```

⇒ log-likelihood function

Find MLE using optim function ⇒ by default, find argmin

```
(betahat <- optim(c(10, -.1), logL, orings=orings, control=list(fnscale=-1))$par)
```

to find argmax, need to use this control

```
## [1] 11.6671414 -0.2162982
```

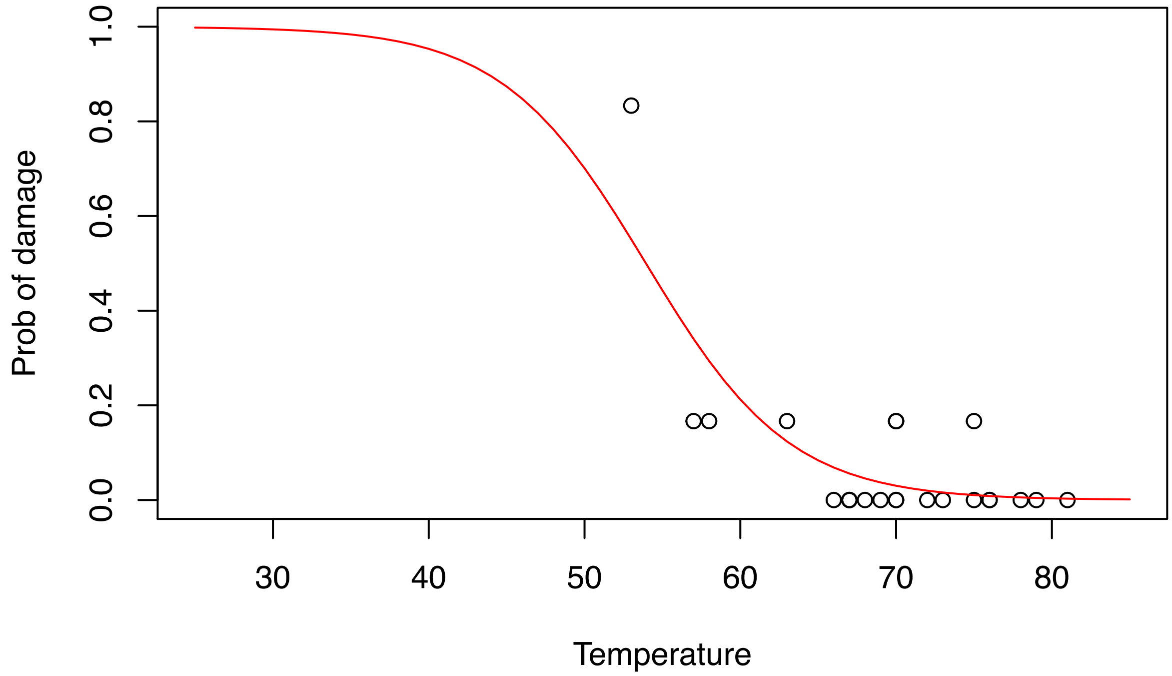
plot fitted model

```
plot(damage/6 ~ temp, orings, xlim=c(25,85), ylim=c(0,1),
     xlab="Temperature", ylab="Prob of damage")
```

$$\hat{p} = \frac{e^{\beta_0 + \beta_1 t}}{1 + e^{\beta_0 + \beta_1 t}}$$

fitted line

```
x <- seq(25,85,1)
ilogit <- function(x) exp(x)/(1+exp(x))
lines(x, ilogit(betahat[1] + betahat[2]*x), col="red")
```



prediction for temp of 29

```
ilogit (betahat[1] + betahat[2]*29)
```

$\hat{p}$

```
## [1] 0.995479
```

Using the glm command instead

estimate MLE

```
logitmod <- glm(cbind(damage,6-damage) ~ temp, family=binomial, orings)
summary(logitmod)
```

by default link function g is logit

if use probit use family=binomial(link=probit)

\# of observation

$$\begin{bmatrix} y_1 & m_1-y_1 \\ y_2 & m_2-y_2 \\ \vdots & \vdots \\ y_n & m_n-y_n \end{bmatrix}$$

```
##
## Call:
## glm(formula = cbind(damage, 6 - damage) ~ temp, family = binomial,
##     data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9529  -0.7345  -0.4393  -0.2079   1.9565
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.66299    3.29626   3.538 0.000403 ***
## temp        -0.21623    0.05318  -4.066 4.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 38.898  on 22  degrees of freedom
## Residual deviance: 16.912  on 21  degrees of freedom
## AIC: 33.675
##
## Number of Fisher Scoring iterations: 6
```

```r
predict(logitmod, newdata=list(temp=29), type="response")
```

```
##         1
## 0.9954687
```

CI. $\eta = \beta_0 + \beta_1 29 = [1\ \ 29]\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}$

CI: $\eta \pm 2\sqrt{[1\ \ 29]\, I(\hat\beta)^{-1}\begin{bmatrix}1\\ 29\end{bmatrix}}$

Confidence Interval for p

Compute standard errors

```r
phat <- ilogit(betahat[1] + orings$temp*betahat[2])
I11 <- sum(6*phat*(1 - phat))
I12 <- sum(6*orings$temp*phat*(1 - phat))
I22 <- sum(6*orings$temp^2*phat*(1 - phat))
Iinv <- solve(matrix(c(I11, I12, I12, I22), 2, 2))
sqrt(Iinv[1,1])
```

se($\hat\beta_1$)

$I(\hat\beta) = \begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22}\end{bmatrix}$

```
## [1] 3.296634
```

binomial regression with a logit link

```r
sqrt(Iinv[2,2])
```

se($\hat\beta_2$)

$I(\hat\beta) = \begin{pmatrix} \sum_i m_i \hat p_i(1-\hat p_i) & \sum_i m_i x_i \hat p_i(1-\hat p_i) \\ \sum_i m_i x_i \hat p_i(1-\hat p_i) & \sum_i m_i x_i^2 \hat p_i(1-\hat p_i)\end{pmatrix}$

where $\hat p_i = \dfrac{e^{\hat\beta_0+\hat\beta_1 t_i}}{1+e^{\hat\beta_0+\hat\beta_1 t_i}}$

```
## [1] 0.05318407
```

Compute CI for eta = beta0 + beta1 29 $[1\ \ 29]\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}$

```r
si2 <- matrix(c(1, 29), 1, 2) %*% Iinv %*% matrix(c(1, 29), 2, 1)
etahat = betahat[1] + betahat[2]*29
eta_l = etahat - 2*sqrt(si2)
eta_r = etahat + 2*sqrt(si2)
etahat
```

$[1\ \ 29]\, I(\hat\beta)^{-1}\begin{bmatrix}1\\ 29\end{bmatrix}$

```
## [1] 5.394493
```

```r
c(eta_l, eta_r)
```

```
## [1] 1.851533 8.937452
```

Compute CI for p

```r
ilogit (etahat)
```

```
## [1] 0.995479
```

```r
c(ilogit (eta_l), ilogit (eta_r))
```

```
## [1] 0.8643070 0.9998686
```

$z^* = \dfrac{\hat\beta_1}{se(\hat\beta_1)}$ → MLE for β_1

Wald Test

Compute MLE

```r
library(faraway)
data(orings)
logL <- function(beta, orings) {
```

```r
  eta <- cbind(1, orings$temp) %*% beta
  return( sum( orings$damage*eta - 6*log(1 + exp(eta)) ) )
}
(betahat <- optim(c(10, -.1), logL, orings=orings, control=list(fnscale=-1))$par)
```

```
## [1] 11.6671414 -0.2162982
```

Compute standard errors of MLE

```r
ilogit <- function(x) exp(x)/(1+exp(x))
phat <- ilogit(betahat[1] + orings$temp*betahat[2])
I11 <- sum(6*phat*(1 - phat))
I12 <- sum(6*orings$temp*phat*(1 - phat))
I22 <- sum(6*orings$temp^2*phat*(1 - phat))
Iinv <- solve(matrix(c(I11, I12, I12, I22), 2, 2))
sqrt(Iinv[1,1])
```

```
## [1] 3.296634
```

```r
sqrt(Iinv[2,2])
```

```
## [1] 0.05318407
```

Wald test statistic

$se(\hat{\beta}_1)$

$\hat{\beta}_1$

```r
betahat[2]/sqrt(Iinv[2,2])
```

```
## [1] -4.066974
```

p-value from Wald test statistic

mean ↑ ↗ var ⇒ consider standard normal

```r
2*pnorm(abs(betahat[2]/sqrt(Iinv[2,2])), 0, 1, lower=FALSE)
```

check p-value

```
## [1] 4.762755e-05
```

Likelihood Ratio test

Full: $y_i \sim Bin(6, p_i)$

$p_i = g^{-1}(\eta_i) \quad \eta_i = \beta_0 + \beta_1 t_i$

Compute maximum log likelihood from the full model

```r
(MaxlogL.F = logL(betahat,orings))
```

reduced $y_i \sim Bin(6, p)$

```
## [1] -27.37971
```

$p = g^{-1}(\eta) \quad \eta = \beta_0$

Compute maximum log likelihood from the reduced model

$\log L(p) = C + \sum_{i=1}^{23} [y_i \log p + (6 - y_i) \log (1-p)]$

```r
y <- orings$damage
n <- rep(6, length(y))
phatN <- sum(y)/sum(n)
(MaxlogL.R = sum(orings$damage)*log(phatN) + sum(6-orings$damage)*log(1-phatN))
```

$n = (\overbrace{6, 6, \cdots, 6}^{23})$

```
## [1] -38.3724
```

LR test statistic

closed form solution for MLE exist

```r
(LR = -2*(MaxlogL.R - MaxlogL.F))
```

✿?

$\boxed{\hat{p}^{R}_{MLE} = \frac{\sum_{i=1}^{23} y_i}{23 \times 6}}$ → reduced form

```
## [1] 21.98538
```

p-value from LR test statistic

```r
pchisq(LR, df=1,lower=FALSE)
```

p value

Max Log Likelihood

$= \sum_{i=1}^{23} [y_i \log \hat{p} + (6-i) \log(1-\hat{p})]$

```
## [1] 2.747354e-06
```

< 0.05

Wald Test vs Likelihood Ratio test

Square of Wald test statistic

```
(betahat[2]/sqrt(Iinv[2,2]))^2
```

```
## [1] 16.54028
```

not the same since the sample size is small

LR test statistic

```
(LR = -2*(MaxlogL.R - MaxlogL.F))
```

```
## [1] 21.98538
```

Deviance

$y_i = \beta_0 + \beta_1 t_i$

Deviance and df for the fitted model

```
y <- orings$damage
n <- rep(6, length(y))
ylogxy <- function(x, y) ifelse(y == 0, 0, y*log(x/y))
(D <- -2*sum(ylogxy(n*phat, y) + ylogxy(n*(1-phat), n - y)))
```

$n = 6$

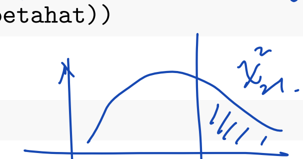
```
## [1] 16.91228
```

$\sum_{i=1}^{n} y_i \log \frac{6\hat{p}}{y_i} + (6-y_i) \log \frac{6(1-\hat{p})}{6-y_i}$

```
(df <- length(y) - length(betahat))
```

$n - k$

```
## [1] 21
```




```
pchisq(D, df,lower=FALSE)
```

```
## [1] 0.7164098
```

> 0.5 fitted model is good enough

Deviance and df for the fitted model using the glm command

```
logitmod <- glm(cbind(damage,6-damage) ~ temp, family=binomial, orings)
deviance(logitmod)
```

```
## [1] 16.91228
```

```
df.residual(logitmod)
```

```
## [1] 21
```

Deviance and df for the null model $y_i = \beta_0$

```
(phatN <- sum(y)/sum(n))
```

```
## [1] 0.07971014
```

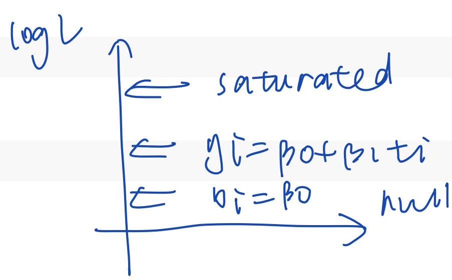
```
(DN <- -2*sum(ylogxy(n*phatN, y) + ylogxy(n*(1-phatN), n - y)))
```

```
## [1] 38.89766
```

```
(dfN <- length(y) - 1)
```

```
## [1] 22
```

```
pchisq(DN, dfN,lower=FALSE)
```



the deviance not explained is too much

→ the null model is not good enough

```
## [1] 0.0144977
```

< 0.05

Deviance and df for the null model using the glm command

```
logitnull <- glm(cbind(y, n - y) ~ 1, family=binomial)
summary(logitnull)
```

```
##
## Call:
## glm(formula = cbind(y, n - y) ~ 1, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9984  -0.9984  -0.9984   0.6947   4.4781
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.4463     0.3143  -7.783 7.06e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 38.898  on 22  degrees of freedom
## Residual deviance: 38.898  on 22  degrees of freedom
## AIC: 53.66
##
## Number of Fisher Scoring iterations: 4
```

β_0

→ scaled deviance for fitted model in this case, it is null models

```
ilogit(-2.4463)
```

$\hat{p}_{null}$.

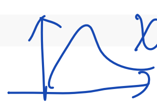
fitted $\eta_i = \beta_0 + \beta_1 t_i$

null $\eta_i = \beta_0$

```
## [1] 0.07970954
```

LRT using deviance

```
DN-D
```



χ^2_1 → # of parameter difference is 1.

```
## [1] 21.98538
```

```
pchisq(DN - D, dfN - df, lower=FALSE)
```

```
## [1] 2.747354e-06
```

< 0.05 → prefer full model