Homework 2

Yuki Joyama

2024-02-16

1

```r
# load data
dose = c(0, 1, 2, 3, 4)
num = c(30, 30, 30, 30, 30)
dead = c(2, 8, 15, 23, 27)
data_1 = data.frame(dose, num, dead)

# visualization
# plot(data_1$dose, data_1$dead/data_1$num, xlab = 'dose', ylab = 'Proportion dying', cex = 1.5, pch =

# data prep
x = data_1$dose
y = data_1$dead
m = data_1$num
resp = cbind(y, m-y)
```

Now, I will fit the model $g(P(dying)) = \alpha + \beta X$ with logit, probit, and complementary log-log links.

```r
# fit logistic model, logit
glm_logit = glm(resp ~ x, family = binomial(link = 'logit'))
summary(glm_logit)
```

```
##
## Call:
## glm(formula = resp ~ x, family = binomial(link = "logit"))
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.3238     0.4179  -5.561 2.69e-08 ***
## x             1.1619     0.1814   6.405 1.51e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 64.76327  on 4  degrees of freedom
## Residual deviance:  0.37875  on 3  degrees of freedom
## AIC: 20.854
##
## Number of Fisher Scoring iterations: 4
```

```r
# fit logistic model, probit
glm_probit = glm(resp ~ x, family = binomial(link = 'probit'))
summary(glm_probit)
```

```
##
## Call:
## glm(formula = resp ~ x, family = binomial(link = "probit"))
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.37709    0.22781  -6.045 1.49e-09 ***
## x            0.68638    0.09677   7.093 1.31e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 64.76327  on 4  degrees of freedom
## Residual deviance:  0.31367  on 3  degrees of freedom
## AIC: 20.789
##
## Number of Fisher Scoring iterations: 4
```

```r
# fit logistic model, cloglog
glm_cloglog = glm(resp ~ x, family = binomial(link = 'cloglog'))
summary(glm_cloglog)
```

```
##
## Call:
## glm(formula = resp ~ x, family = binomial(link = "cloglog"))
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9942     0.3126  -6.378 1.79e-10 ***
## x             0.7468     0.1094   6.824 8.86e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 64.7633  on 4  degrees of freedom
## Residual deviance:  2.2305  on 3  degrees of freedom
## AIC: 22.706
##
## Number of Fisher Scoring iterations: 5
```

(a)

```r
# 95% CI for beta, logit
beta = glm_logit$coefficients[2]
```

```r
se = sqrt(vcov(glm_logit)[2, 2])
round(beta + c(qnorm(0.025), -qnorm(0.025)) * se, 3)
```

```
## [1] 0.806 1.517
```

```r
# 95% CI for beta, probit
beta = glm_probit$coefficients[2]
se = sqrt(vcov(glm_probit)[2, 2])
round(beta + c(qnorm(0.025), -qnorm(0.025)) * se, 3)
```

```
## [1] 0.497 0.876
```

```r
# 95% CI for beta, cloglog
beta = glm_cloglog$coefficients[2]
se = sqrt(vcov(glm_cloglog)[2, 2])
round(beta + c(qnorm(0.025), -qnorm(0.025)) * se, 3)
```

```
## [1] 0.532 0.961
```

```r
# p_hat(dying|x = 0.01), logit
predict(glm_logit, newdata = data.frame(x = 0.01), type = 'response')
```

```
##          1
## 0.09011997
```

```r
# calculate by hand
# or = exp(coef(glm_logit)[1] + 0.01 * coef(glm_logit)[2])
# or / (1 + or)

# p_hat(dying|x = 0.01), probit
predict(glm_probit, newdata = data.frame(x = 0.01), type = 'response')
```

```
##         1
## 0.0853078
```

```r
# p_hat(dying|x = 0.01), cloglog
predict(glm_cloglog, newdata = data.frame(x = 0.01), type = 'response')
```

```
##         1
## 0.1281601
```

Model	Estimate of beta	CI for beta	Deviance	p_hat(dying|x=0.01)
logit	1.162	(0.806-1.517)	0.379	0.0901
probit	0.686	(0.497-0.876)	0.314	0.0853
c-log-log	0.747	(0.532-0.961)	2.231	0.128

The estimate of beta in the logit model represents the change in the log odds of the response variable for a one-unit change in the predictor variable (dose). The 95% CI for the estimate of beta provides a range within which we can be 95% confident that the true value of beta lies. The deviance can be used to check the goodness of fit of the models and 0.314 in probit model indicates a better fit. $\hat{p}(dying|x = 0.01)$ gives a probability estimate given that the predictor variable x takes the value of 0.01. In logit and probit model, the probabilities are similar. However, in the c-log-log model, which employs an asymmetric link function, the estimated probabilities appears to be larger than the other two models.

(b)

Three models can be expressed as below:
$log(\frac{p}{1-p}) = \alpha + \beta X$
$\Phi^{-1}(p) = \alpha + \beta X$
$log(-log(1-p)) = \alpha + \beta X$
We want to estimate x_0 such that $\alpha + \beta X = g(p = 0.5)$

Given p=0.5,
Logit: $0 = \alpha + \beta x_0$ -> $x_0 = -\frac{\alpha}{\beta}$

$var(\hat{x_0}) = (\frac{\partial x_0}{\partial \alpha})^2 var(\hat{\alpha}) + (\frac{\partial x_0}{\partial \beta})^2 var(\hat{\beta}) + 2(\frac{\partial x_0}{\partial \alpha})(\frac{\partial x_0}{\partial \beta}) cov(\hat{\alpha}, \hat{\beta})$
-> $var(\hat{x_0}) = \frac{1}{\beta^2} var(\hat{\alpha}) + \frac{\alpha^2}{\beta^4} var(\hat{\beta}) - 2\frac{\alpha}{\beta^3} cov(\hat{\alpha}, \hat{\beta})$

Probit:

$0 = \alpha + \beta x_0$ -> $x_0 = -\frac{\alpha}{\beta}$
$var(\hat{x_0}) = \frac{1}{\beta^2} var(\hat{\alpha}) + \frac{\alpha^2}{\beta^4} var(\hat{\beta}) - 2\frac{\alpha}{\beta^3} cov(\hat{\alpha}, \hat{\beta})$

C-loglog:

$-0.367 = \alpha + \beta x_0$ -> $x_0 = -\frac{0.367+\alpha}{\beta}$
$var(\hat{x_0}) = (\frac{\partial x_0}{\partial \alpha})^2 var(\hat{\alpha}) + (\frac{\partial x_0}{\partial \beta})^2 var(\hat{\beta}) + 2(\frac{\partial x_0}{\partial \alpha})(\frac{\partial x_0}{\partial \beta}) cov(\hat{\alpha}, \hat{\beta})$
-> $var(\hat{x_0}) = \frac{1}{\beta^2} var(\hat{\alpha}) + \frac{(\alpha - log(-log(1-0.5)))^2}{\beta^4} var(\hat{\beta}) + 2\frac{log(-log(1-0.5)) - \alpha}{\beta^3} cov(\hat{\alpha}, \hat{\beta})$

The asymptotic $(1-\alpha)100\%$ CI of x_0 is $[\hat{x_0} - z_{\alpha/2}\sqrt{var(\hat{x_0})}, \hat{x_0} + z_{\alpha/2}\sqrt{var(\hat{x_0})}]$

Now, I will calculate these values using the following codes.

```
# LD50 point est, logit
x0 = - glm_logit$coefficients[1]/glm_logit$coefficients[2]
round(exp(x0), 3)
```

```
## (Intercept)
##       7.389
```

```
# 95% CI
beta0 = glm_logit$coefficients[1]
beta1 = glm_logit$coefficients[2]
betacov = vcov(glm_logit) # inverse fischer information
varx0 = betacov[1, 1]/(beta1^2) + betacov[2, 2]*(beta0^2)/(beta1^4) - 2*betacov[1,2]*beta0/(beta1^3)
se = sqrt(varx0)
round(exp(x0 + c(qnorm(0.05), -qnorm(0.05)) * sqrt(varx0)), 3)
```

```
## [1] 5.51 9.91
```

```r
# LD50 point est, probit
x0 = - glm_probit$coefficients[1]/glm_probit$coefficients[2]
round(exp(x0), 3)
```

```
## (Intercept)
##       7.436
```

```r
# 95% CI
beta0 = glm_probit$coefficients[1]
beta1 = glm_probit$coefficients[2]
betacov = vcov(glm_probit) # inverse fischer information
varx0 = betacov[1, 1]/(beta1^2) + betacov[2, 2]*(beta0^2)/(beta1^4) - 2*betacov[1,2]*beta0/(beta1^3)
se = sqrt(varx0)
round(exp(x0 + c(qnorm(0.05), -qnorm(0.05)) * sqrt(varx0)), 3)
```

```
## [1] 5.583 9.904
```

```r
# LD50 point est, cloglog
x0 = (log(-log(1 - 0.5)) - glm_cloglog$coefficients[1])/(glm_cloglog$coefficients[2])
round(exp(x0), 3)
```

```
## (Intercept)
##       8.841
```

```r
# 95% CI
beta0 = glm_cloglog$coefficients[1]
beta1 = glm_cloglog$coefficients[2]
betacov = vcov(glm_cloglog) # inverse fischer information
varx0 = betacov[1, 1]/(beta1^2) + betacov[2, 2]*(beta0 - (log(-log(1 - 0.5))))^2/(beta1^4) + 2*betacov[
se = sqrt(varx0)
round(exp(x0 + c(qnorm(0.05), -qnorm(0.05)) * sqrt(varx0)), 3)
```

```
## [1]  6.526 11.977
```

The results are as follows.

Model	Estimate LD50	90% CI
logit	7.389	(5.510-9.910)
probit	7.436	(5.583-9.904)
c-log-log	8.841	(6.526-11.977)

2

- Amount: one-time two-year scholarship
- Offer: the number of offers made with the corresponding scholarship
- Enrolls: the number of offer accepted

```r
# load data
amount = seq(10, 90, 5)
offers = c(4, 6, 10, 12, 39, 36, 22, 14, 10, 12, 8, 9, 3, 1, 5, 2, 1)
enrolls = c(0, 2, 4, 2, 12, 14, 10, 7, 5, 5, 3, 5, 2, 0, 4, 2, 1)

data_2 = data.frame(amount, offers, enrolls)

# visualization
# plot(data_2$amount, data_2$enrolls/data_2$offers, xlab = 'amount', ylab = 'Proportion enrollment', ce

# data prep
x = data_2$amount
y = data_2$enrolls
m = data_2$offers + data_2$enrolls
resp = cbind(y, m-y)
```

(a) How does the model fit the data?

```r
# fit logistic model, logit
glm_logit = glm(resp ~ x, family = binomial(link = 'logit'))
summary(glm_logit)
```

```
##
## Call:
## glm(formula = resp ~ x, family = binomial(link = "logit"))
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.597627   0.365184  -4.375 1.22e-05 ***
## x            0.016290   0.007893   2.064    0.039 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9.0736  on 16  degrees of freedom
## Residual deviance: 4.8285  on 15  degrees of freedom
## AIC: 52.086
##
## Number of Fisher Scoring iterations: 4
```

The residual deviance is 4.8285, with 15 degrees of freedom. This indicates that the model explains some of the variability in the data, as the residual deviance is lower than the null deviance.

(b) How do you interpret the relationship between the scholarship amount and enrollment rate? What is 95% CI?

```r
# 95% CI for beta, logit
beta = glm_logit$coefficients[2]
```

```r
se = sqrt(vcov(glm_logit)[2, 2])
round(beta + c(qnorm(0.025), -qnorm(0.025)) * se, 3)
```

```
## [1] 0.001 0.032
```

(c) How much scholarship should we provide to get 40% yield rate (the percentage of admitted students who enroll?) What is the 95% CI?