Hw2 Logistic regression practice

Cary Ni

2023-02-14

Problem 1

```r
dose = c(0, 1, 2, 3, 4)
all = c(rep(30, 5))
dying = c(2, 8, 15, 23, 27)
no_dying = all - dying
new_data = data.frame(dose=0.01)
# fit model with logit link
model_1 = glm(cbind(dying, no_dying)~dose,family=binomial(link='logit'))
summary(model_1)
```

```
##
## Call:
## glm(formula = cbind(dying, no_dying) ~ dose, family = binomial(link = "logit"))
##
## Deviance Residuals:
##       1        2        3        4        5
## -0.4510   0.3597   0.0000   0.0643  -0.2045
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.3238     0.4179  -5.561 2.69e-08 ***
## dose          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
confint.default(model_1, parm = "dose")
```

```
##          2.5 %   97.5 %
## dose 0.8063266 1.517463
```

```r
predict(model_1, newdata=new_data, type = "response")
```

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

```r
# fit model with probit link
model_2 = glm(cbind(dying, no_dying)~dose,family=binomial(link='probit'))
summary(model_2)
```

```
##
## Call:
## glm(formula = cbind(dying, no_dying) ~ dose, family = binomial(link = "probit"))
##
## Deviance Residuals:
##        1         2         3         4         5
## -0.35863   0.27493   0.01893   0.18230  -0.27545
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.37709    0.22781  -6.045 1.49e-09 ***
## dose         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
confint.default(model_2, parm = "dose")
```

```
##          2.5 %    97.5 %
## dose 0.4967217 0.8760393
```

```r
predict(model_2, newdata=new_data, type = "response")
```

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

```r
# fit model with c-log-log link
model_3 = glm(cbind(dying, no_dying)~dose,family=binomial(link='cloglog'))
summary(model_3)
```

```
##
## Call:
## glm(formula = cbind(dying, no_dying) ~ dose, family = binomial(link = "cloglog"))
##
```

```
## Deviance Residuals:
##       1        2        3        4        5
## -1.0831   0.2132   0.4985   0.5588  -0.6716
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9942     0.3126  -6.378 1.79e-10 ***
## dose          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
```

```
confint.default(model_3, parm = "dose")
```

```
##        2.5 %    97.5 %
## dose 0.53232 0.9613187
```

```
predict(model_3, newdata=new_data, type = "response")
```

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

```
# reference for the approximation for variance
knitr::include_graphics("approx.png")
```

$$Var(x/y) \approx \left(\frac{E(x)}{E(y)}\right)^2 \left(\frac{Var(x)}{E(x)^2} + \frac{Var(y)}{E(y)^2} - 2\frac{Cov(x,y)}{E(x)E(y)}\right)$$

```
# create a function finding x and CI for models, mu is the expected value
# when p = 0.5
get_x_ci = function(input_model, mu=0, alpha = 0.05) {
  cof_zero = input_model$coefficients[1]
  cof_one = input_model$coefficients[2]
  x_hat = (-cof_zero+mu)/cof_one
  cov_matrix = vcov(input_model)
  x_variance = (((mu-cof_zero)/cof_one)^2)*(cov_matrix[1, 1]/((mu-cof_zero)^2)+cov_matrix[2, 2]/(cof_on
  x_ci = c(x_hat + sqrt(x_variance)*qnorm(alpha), x_hat - sqrt(x_variance)*qnorm(alpha))
  return(unname(x_ci))
}
# get 90% CI for dose in logit model
get_x_ci(model_1)
```

```
## [1] 1.706498 2.293502
```

```r
# reverse log transformation
get_x_ci(model_1) %>% exp()
```

```
## [1] 5.509631 9.909583
```

```r
# get 90% CI for probit model
get_x_ci(model_2)
```

```
## [1] 1.719653 2.292968
```

```r
# reverse log transformation
get_x_ci(model_2) %>% exp()
```

```
## [1] 5.582588 9.904289
```

```r
# get 90% CI for cloglog model
get_x_ci(model_3, mu=clogloglink(0.5))
```

```
## [1] 1.875834 2.483022
```

```r
# reverse log transformation
get_x_ci(model_3, mu=clogloglink(0.5)) %>% exp()
```

```
## [1]  6.526261 11.977407
```

Problem 2

```r
amount = c(seq(10, 90, by=5))
offer = c(4, 6, 10, 12, 39, 36, 22, 14, 10, 12, 8, 9, 3, 1, 5, 2, 1)
enroll = c(0, 2, 4, 2, 12, 14, 10, 7, 5, 5, 3, 5, 2, 0, 4, 2, 1)
model_4 = glm(cbind(enroll, offer-enroll)~amount,family=binomial(link='logit'))
summary(model_4)
```

```
##
## Call:
## glm(formula = cbind(enroll, offer - enroll) ~ amount, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4735  -0.6731   0.1583   0.5285   1.1275
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.64764    0.42144  -3.910 9.25e-05 ***
## amount       0.03095    0.00968   3.197  0.00139 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21.617  on 16  degrees of freedom
## Residual deviance: 10.613  on 15  degrees of freedom
## AIC: 51.078
##
## Number of Fisher Scoring iterations: 4
```

```r
# pearson chisq
sum(residuals(model_4,type='pearson')^2)
```

```
## [1] 8.814299
```

```r
# compare with chisq(17-2)
1-pchisq(10.613,15) # larger than 0.05, fail to reject the null that the fit is good
```

```
## [1] 0.7795148
```

```r
# for small mi use Hosmer-Lemeshow
hoslem.test(model_4$y, fitted(model_4), g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model_4$y, fitted(model_4)
## X-squared = 1.6111, df = 8, p-value = 0.9907
```

```r
confint.default(model_4, parm = "amount")
```

```
##              2.5 %    97.5 %
## amount 0.01197845 0.0499224
```

```r
exp(confint.default(model_4, parm = "amount"))
```

```
##           2.5 %  97.5 %
## amount 1.01205 1.05119
```

```r
# get x to have of 40% enrollment rate
cof_zero_2 = model_4$coefficients[1]
cof_one_2 = model_4$coefficients[2]
unname((log(2/3)-cof_zero_2)/cof_one_2)
```

```
## [1] 40.13429
```

```r
# use the function in problem 1 to get 95% CI
get_x_ci(model_4, mu =log(2/3), alpha = 0.025 )
```

```
## [1] 30.58304 49.68553
```