p8105\_hw2\_mp3745

Matthew Parker

2020-02-17

## PROBLEM 1

### i)

Enter data

# Covariate  
dose = 0:4  
  
# Response  
resp = tibble(  
 num\_dead = c(2, 8, 15, 23, 27),  
 num\_total = 30 - num\_dead  
) %>%   
 as.matrix()

Logistic models with grouped data

# Logit  
fit\_logit =   
 glm(resp ~ dose,   
 family = binomial(link = 'logit'))  
  
  
# Probit  
fit\_probit =   
 glm(resp ~ dose,   
 family = binomial(link = 'probit'))  
  
  
# cloglog  
fit\_cloglog =   
 glm(resp ~ dose,   
 family = binomial(link = 'cloglog'))

95% confidence intervals

# logit  
ci\_logit = fit\_logit %>%   
 broom::tidy() %>%   
 mutate(  
 lower = estimate - 1.96 \* std.error,  
 upper = estimate + 1.96 \* std.error  
 )  
  
# probit  
ci\_probit = fit\_probit %>%   
 broom::tidy() %>%   
 mutate(  
 lower = estimate - 1.96 \* std.error,  
 upper = estimate + 1.96 \* std.error  
 )  
  
# cloglog  
# probit  
ci\_cloglog = fit\_cloglog %>%   
 broom::tidy() %>%   
 mutate(  
 lower = estimate - 1.96 \* std.error,  
 upper = estimate + 1.96 \* std.error  
 )

Prob of dying given x = 0.01

# value to predict  
newdata = data.frame(  
 dose = 0.01  
)  
  
# logit  
pred\_logit = predict(fit\_logit, newdata = newdata, type = "response")  
  
# probit  
pred\_probit = predict(fit\_probit, newdata = newdata, type = "response")  
  
# cloglog  
pred\_cloglog = predict(fit\_cloglog, newdata = newdata, type = "response")

tibble(  
 model = c("Logit", "Probit", "c-log-log"),  
 Estimate\_of\_β = c(fit\_logit$coefficients[2], fit\_probit$coefficients[2], fit\_cloglog$coefficients[2]),  
 CI\_for\_β = c(str\_c("(", round(ci\_logit$lower[2], 3), ", ", round(ci\_logit$upper[2], 3), ")"),   
 str\_c("(", round(ci\_probit$lower[2], 3), ", ", round(ci\_probit$upper[2], 3), ")"),  
 str\_c("(", round(ci\_cloglog$lower[2], 3), ", ", round(ci\_cloglog$upper[2], 3), ")")),  
 Deviance = c(round(fit\_logit$deviance, 3), round(fit\_probit$deviance, 3), round(fit\_cloglog$deviance, 3)),  
 p\_dying\_x\_0.01 = c(pred\_logit, pred\_probit, pred\_cloglog)  
) %>% knitr::kable()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| model | Estimate\_of\_β | CI\_for\_β | Deviance | p\_dying\_x\_0.01 |
| Logit | 1.1618949 | (0.806, 1.517) | 0.379 | 0.0901200 |
| Probit | 0.6863805 | (0.497, 0.876) | 0.314 | 0.0853078 |
| c-log-log | 0.7468193 | (0.532, 0.961) | 2.230 | 0.1281601 |

The logit model has the highest estimate for β, with the probit model having the lowest. All estimates of β are positive, which means the risk of dying increases with dose. For the logit model, the interpretation is a 1 unit increase in dose causes the log odds of dying to increase by 1.1618949. All three 95% confidence intervals do not contain zero, which further suggests the risk of dying increases with dose. The probit model had the smallest deviance and the c-log-log model had the highest deviance. This suggests that the logit and probit models may fit the data better. All three models provide similar estimates for the probability of dying given the dose is 0.01, but the c-log-log model has the highest estimate at 0.1281601.

### ii)

#### logit

# coefficients  
logit\_beta0 = fit\_logit$coefficients[1]  
logit\_beta1 = fit\_logit$coefficients[2]  
  
# fisher information inverse  
logit\_betacov = vcov(fit\_logit)   
  
# point estimate of ln(LD50)  
logit\_x0fit = -logit\_beta0/logit\_beta1  
  
# asymptotic variance  
logit\_varx0 = logit\_betacov[1,1]/(logit\_beta1^2) + logit\_betacov[2,2]\*(logit\_beta0^2)/(logit\_beta1^4) -  
 2\*logit\_betacov[1,2]\*logit\_beta0/(logit\_beta1^3)

The point estimate for LD50 (median lethal dose) is 7.3890561 with a 90% CI of (5.5096315, 9.909583)

#### Probit

# coefficients  
probit\_beta0 = fit\_probit$coefficients[1]  
probit\_beta1 = fit\_probit$coefficients[2]  
  
# fisher information inverse  
probit\_betacov = vcov(fit\_probit)   
  
# point estimate of ln(LD50)  
probit\_x0fit = -probit\_beta0/probit\_beta1  
  
# asymptotic variance  
probit\_varx0 = probit\_betacov[1,1]/(probit\_beta1^2) + probit\_betacov[2,2]\*(probit\_beta0^2)/(probit\_beta1^4) -  
 2\*probit\_betacov[1,2]\*probit\_beta0/(probit\_beta1^3)

The point estimate for LD50 is 7.4358301 with a 90% CI of (5.5825882, 9.9042894)

#### Cloglog

# coefficients  
cloglog\_beta0 = fit\_cloglog$coefficients[1]  
cloglog\_beta1 = fit\_cloglog$coefficients[2]  
  
# fisher information inverse  
cloglog\_betacov = vcov(fit\_cloglog)   
  
# point estimate of ln(LD50)  
cloglog\_x0fit = (log(-log(0.5)) - cloglog\_beta0)/cloglog\_beta1  
  
# asymptotic variance  
cloglog\_varx0 = cloglog\_betacov[1,1]/(cloglog\_beta1^2) +   
 cloglog\_betacov[2,2]\*((cloglog\_beta0 - log(-log(0.5)))^2)/(cloglog\_beta1^4) -  
 2\*cloglog\_betacov[1,2]\*(cloglog\_beta0 - log(-log(0.5)))/(cloglog\_beta1^3)

The point estimate for LD50 is 8.8412489 with a 90% CI of (6.5262606, 11.9774075)

Comparing the three estimates of the median lethal dose, it appears that the logit and probit estimates and 90% confidence intervals are similar. The estimate of the median lethal dose for cloglog is higher and the confidence interval is wider than logit and probit.

## Problem 2

Enter in the data

prob2\_data = tibble(  
 amount = seq(from = 10000, to = 90000, by = 5000),  
 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),  
 non\_enrolls = offers - enrolls  
)  
  
# Predictor  
amount = prob2\_data %>%   
 select(amount) %>%   
 as.matrix()  
  
# Response  
enrolls = prob2\_data %>%   
 select(enrolls, non\_enrolls) %>%   
 as.matrix()

## i)

Logistic model

prob2\_fit =   
 glm(enrolls ~ amount,   
 family = binomial(link = 'logit'))  
  
summary(prob2\_fit)

##   
## Call:  
## glm(formula = enrolls ~ 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) -1.648e+00 4.214e-01 -3.910 9.25e-05 \*\*\*  
## amount 3.095e-05 9.680e-06 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

Check goodness of fit

# Deviance  
prob2\_dev = prob2\_fit$deviance  
  
# compare with chisq(17-2)  
fit\_pval = 1 - pchisq(prob2\_dev, 15) # fit is not good, later will see why (over dispersion; lack of covariate)

It looks like this model may do a good job of fitting the data. The p-value from the deviance test for goodness of fit is 0.7795345 > 0.05. This means we fail to reject the null hypothesis of the model being a good fit.

## ii)

Get 95% confidence interval

# Tidy output  
prob2\_fit\_tidy =   
 prob2\_fit %>%   
 broom::tidy()  
  
# 95% CI  
ci\_amount = prob2\_fit\_tidy %>%   
 mutate(  
 lower = estimate - 1.96 \* std.error,  
 upper = estimate + 1.96 \* std.error  
 ) %>%   
 filter(term == "amount") %>%   
 select(estimate, lower, upper)

3.095042610^{-5} is the change in log odds of enrollment per increase of $5,000 in scholarship amount. The 95% CI is (1.197810510^{-5}, 4.992274710^{-5}).

## iii)

# coefficients  
prob2\_beta0 = prob2\_fit$coefficients[1]  
prob2\_beta1 = prob2\_fit$coefficients[2]  
  
# fisher information inverse  
prob2\_betacov = vcov(prob2\_fit)   
  
# point estimate  
prob2\_x0fit = (log(2/3) - prob2\_beta0)/prob2\_beta1  
  
# asymptotic variance  
prob2\_varx0 = prob2\_betacov[1,1]/(prob2\_beta1^2) + prob2\_betacov[2,2]\*((prob2\_beta0 - log(2/3))^2)/(prob2\_beta1^4) -  
 2\*prob2\_betacov[1,2]\*(prob2\_beta0 - log(2/3))/(prob2\_beta1^3)

The point estimate for scholarship amount to provide to get 40% yield rate is $40134.29 with a 95% CI of ($30583.04, $49685.53)