BM2_HW2

Yangyang Chen

2024-02-16

Problem_1

(a) Fill out the table and give comments.

```
# load data
dose=c(0:4)## x_i
num=rep(30, 5) ## m_i
killed=c(2, 8, 15, 23, 27) ## y_i
data=data.frame(dose,num,killed) ## (x_i, m_i, y_i)

# data preparation
x=data$dose
y=data$killed
m=data$num
resp=cbind(y,m-y)   #### counts of success (death=1), failure (surv=0)

# Model fitting
# logit link
glm_logit=glm(resp~x, family=binomial(link='logit'))
# probit link
glm_probit=glm(resp~x, family=binomial(link='probit'))
# complementary log-log link
glm_clog=glm(resp~x, family=binomial(link='cloglog')) # asymmetric

results_func = function(fit,model){
  fit_summary = fit |> summary()
  beta = fit_summary$coefficients[2]
  CI = fit$coefficients +
    kronecker(t(c(0,qnorm(0.025),-qnorm(0.025))), t(t(sqrt(diag(vcov(fit))))))
  CI_lower = CI[2,2]
  CI_upper = CI[2,3]
  dev = fit_summary$deviance
  p_new = predict.glm(fit, newdata = tibble(x = 0.01), type = "response")
  return(tibble(model, beta, CI_lower, CI_upper, dev, p_new))
}

results_func(glm_logit, "logit") |>
  rbind(results_func(glm_probit, "probit")) |>
  rbind(results_func(glm_clog, "c-log-log")) |>
  knitr::kable(digits = 4)
```

model	beta	CI_lower	CI_upper	dev	p_new
logit	1.1619	0.8063	1.5175	0.3787	0.0901
probit	0.6864	0.4967	0.8760	0.3137	0.0853
c-log-log	0.7468	0.5323	0.9613	2.2305	0.1282

(a) All models showed a significant relationship between dose and death with significant p-values less than 0.001.

(b) As the dose increases, the probability of death decreases because all CIs for the dose estimates are positive and do not include zero.

(c) The probit link model fits the data best since it minimizes residual deviance.

(d) The probability of death estimated using the probit link model at a dose level of 0.01 was 0.0853.

(b) Suppose that the dose level is in natural logarithm scale, estimate LD50 with 90% confidence interval based on the three models.

```
LD50_func = function(fit, model){
  alpha = fit$coefficients[1]
  beta = fit$coefficients[2]
  betacov = vcov(fit)
  if(model == "c-log-log"){
    x0fit = (log(-log(0.5)) - alpha) /beta
    varx0 = betacov[1,1]/(beta^2)+
      betacov[2,2]*(log(-log(0.5)) - alpha) ^ 2 / (beta ^ 4) + 2 * betacov[1,2] * (log(-log(0.5)) - alp
  }
  else
  {
    x0fit = -alpha / beta
    varx0 = betacov[1,1] / (beta ^ 2) +
      betacov[2,2] * (alpha ^ 2) / (beta ^ 4) -
      2 * betacov[1,2] *alpha / (beta ^ 3)
  }
  estimate = exp(x0fit)
  CI_lower = exp(x0fit - c(qnorm(0.95)) * sqrt(varx0))
  CI_upper = exp(x0fit + c(qnorm(0.95))*sqrt(varx0))
  return(tibble(model, estimate, CI_lower, CI_upper))
}

LD50_func(glm_logit, "logit") |>
  rbind(LD50_func(glm_probit, "probit")) |>
  rbind(LD50_func(glm_clog, "c-log-log")) |>
  knitr::kable(digits = 2)
```

model	estimate	CI_lower	CI_upper
logit	7.39	5.51	9.91
probit	7.44	5.58	9.90
c-log-log	8.84	6.53	11.98

Problem_2

```r
amount=seq(10, 90, by = 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=data.frame(amount, offers, enrolls)
```

```r
glm_logit_2 = glm(cbind(enrolls, offers - enrolls) ~ amount,
                  family = binomial(link = "logit"))
glm_logit_2 |> summary()
```

```
##
## Call:
## glm(formula = cbind(enrolls, offers - enrolls) ~ amount, family = binomial(link = "logit"))
##
## 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
```

(a) How does the model fit the data?

Since most of the offers are less than 30, we can assume that the data is sparse. Therefore, we should use the Hosmer-Lemeshow statistic to evaluate the goodness of fit.

```r
library(ResourceSelection)
```

```
## ResourceSelection 0.3-6   2023-06-27
```

```r
hoslem_stat = hoslem.test(glm_logit_2$y, fitted(glm_logit_2), g = 10)
hoslem_stat
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  glm_logit_2$y, fitted(glm_logit_2)
## X-squared = 1.6111, df = 8, p-value = 0.9907
```

Because $p = 0.9907 > 0.05$, we fail to reject the null and conclude that the model fits the data well at 95% significant level.

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