HW2_yq2378

Qi Yumeng

2024-02-17

Q1

a Fill out the table and give comments.

```
Dose = 0:4
Dying = c(2,8, 15, 23, 27)
resp = cbind(Dying,Survive = 30 - Dying) # counts of success (dying=1), failure (dyong=0)

df_results_Q1 =
  tibble(
    link_options = c('logit', 'probit', 'cloglog')
  )  |>
  mutate(
    models = map(link_options, ~glm(resp ~ Dose, family = binomial(link = .x))),
    summaries = map(models, summary),
    deviance =round(map_dbl(summaries, ~.x$deviance),2),
    coefficients = map(models, ~coef(summary(.x))),
    confint = map(models, ~confint(.x, level = 0.95)),
    covbeta = map(models, vcov),
    predict_prob = map(models, ~predict(.x, newdata = data.frame(Dose = 0.01), type = "response"))
  ) |>
  mutate(
    alpha = map_dbl(coefficients, ~.x["(Intercept)", "Estimate"]),
    beta = round(map_dbl(coefficients, ~.x["Dose", "Estimate"]),2),
    CI_lower = round(map2_dbl(coefficients, confint, ~.y["Dose", 1]),2),
    CI_upper = round(map2_dbl(coefficients, confint, ~.y["Dose", 2]),2),
    predicted_probability = round(map_dbl(predict_prob, ~.x),2),

    covbeta0 = map_dbl(covbeta, ~.x[1,1]),
    covbeta1 = map_dbl(covbeta, ~.x[2,2]),
    covbeta01 =  map_dbl(covbeta, ~.x[1,2])
  )
```

```
## Waiting for profiling to be done...
## Waiting for profiling to be done...
## Waiting for profiling to be done...
```

```
df_results_Q1_a = df_results_Q1 |>
  mutate(CI = paste("(", CI_lower,", ", CI_upper,")",sep = "")) |>
  select(link_options, beta,
         CI,  deviance, predicted_probability)

# Print the results
```

```
# stargazer(df_results_Q1_a,  summary = FALSE)
```

Different link functions yield significantly varied estimates of beta, with the logit link function producing the largest value. Despite these variations, the 95% confidence intervals (CIs) for all models do not include 0, indicating that the parameter beta is statistically significantly different from 0. This result leads to the rejection of the null hypothesis. The cloglog link exhibits the highest deviance, which could be attributed to its sensitivity to extreme values in the data. (Not quite sure about this one) The predicted probabilities given does level at 0.01 are approximately 0.1

Table 1:

	link_options	beta	CI	deviance	predicted_probability
1	logit	1.16	(0.83, 1.55)	0.38	0.09
2	probit	0.69	(0.5, 0.88)	0.31	0.09
3	cloglog	0.75	(0.55, 0.96)	2.23	0.13

b LD50

```
# LD50 est and CI
df_results_Q1_b = df_results_Q1|>
  mutate(x0fit = round(if_else(link_options == "cloglog", (log(-log(0.5))- alpha)/beta,- alpha/ beta),2
         nature_x0fit = round(exp(x0fit),2),
         varx0 = round(covbeta0/(beta^2)+covbeta1*(alpha^2)/(beta^4)-2*covbeta01*alpha/(beta^3),2),
         CI_lower = exp(x0fit - qnorm(0.95)*sqrt(varx0)),
         CI_upper = exp(x0fit + qnorm(0.95)*sqrt(varx0)),
         CI = paste("(", round(CI_lower,2),", ",round(CI_upper,2),")",sep = "")) |>
  select(link_options, x0fit,nature_x0fit,
         varx0, CI)
#stargazer(df_results_Q1_b,  summary = FALSE)
```

Notice the estimate of LD50 needs a little modification on the constant, but in general the estimates of the three link functions are quite similar. And naturally the $\hat{X}_0$ are basically the same.

Table 2:

	link_options	x0fit	nature_x0fit	varx0	CI
1	logit	2	7.39	0.03	(5.56, 9.82)
2	probit	2	7.39	0.03	(5.56, 9.82)
3	cloglog	2.17	8.76	0.03	(6.59, 11.65)

Q2

a How does the model fit the data?

```
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)

resp = cbind(Enrolls,Non_enrolls = Offers - Enrolls)
```

```r
fit=glm(resp~Amount,family=binomial(link='logit'))
summary(fit)
```

```
##
## Call:
## glm(formula = resp ~ 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
```

```r
#sum(residuals(fit,type='pearson')^2)
#sum(residuals(fit,type='deviance')^2)
# p-value of amount is 0.00139, significant under 0.05
# Residual deviance = 10.613
# AIC = 51.078
```

In general the model fits the data pretty well.

- A significant p-value of 0.0014 for `Amount` indicates that the predictor is significantly associated with the likelihood of enrolling.
- The residual deviance is 10.613 on 15 degrees of freedom.Compared to the null model, this significant reduction in deviance, when compared to the null model, indicates an improved fit of the model to the data.
- The AIC is 51.07, which is not too high, albeit this is a subjective comment.

b interpret the relationship

The p-value for`Amount` is 0.0014, indicating a statistically significant association between the predictor and the likelihood of enrolling under the significance value of 0.05. The estimate is positive, which means the higher of scholarship amount leads to the higher chance of someone get enrolled in the program. More specifically, per unit change of `Amount` will lead to 0.0014 increase on the log odds of enrolling.

The 95% CIs for β and odds ratio(e^{β}) are listed as

```r
confint(fit) [2,]
```

```
## Waiting for profiling to be done...
```

```
##      2.5 %     97.5 %
## 0.01245819 0.05060401
```

```r
exp(confint(fit)) [2,]
```

```
## Waiting for profiling to be done...
```

```
##    2.5 %   97.5 %
## 1.012536 1.051906
```

c

To get a 40% yield rate, the x0fit should be 40.13. And the 95% CI is (27.86, 52.40)

```
beta0= fit$coefficients[[1]]
beta1= fit$coefficients[[2]]
betacov=vcov(fit) # inverse fisher information

x0fit=(log(0.4/(1-0.4))-beta0)/beta1

varx0=betacov[1,1]/(beta1^2)+betacov[2,2]*(beta0^2)/(beta1^4)-2*betacov[1,2]*beta0/(beta1^3)

round(c(x0fit,sqrt(varx0)),2) # point est and se
```

```
## [1] 40.13  6.26
```

```
round(x0fit+c(qnorm(0.025),-qnorm(0.025))*sqrt(varx0),2) # 95% CI for LD50
```

```
## [1] 27.86 52.40
```