# p8131_hw2_ps3194

Pangsibo Shen

2/9/2021

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

**Question 1**

```r
#Create the bioassay dataset
bioassy_df = tibble(dose = 0:4, n_dying = c(2,8,15,23,27))
```

```r
#fit the glm with logit link
fit_logit = glm(cbind(n_dying, 30-n_dying) ~ dose, family = binomial(link = 'logit'), data = bioassy_df
summary(fit_logit)
```

**(a)**

```
##
## Call:
## glm(formula = cbind(n_dying, 30 - n_dying) ~ dose, family = binomial(link = "logit"),
##     data = bioassy_df)
##
## 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
#calculate the lower and upper bounds of CI
logit_ci_lower = summary(fit_logit)$coefficient[2,1] - qnorm(0.975)*summary(fit_logit)$coefficient[2,2]
logit_ci_higher = summary(fit_logit)$coefficient[2,1] + qnorm(0.975)*summary(fit_logit)$coefficient[2,2

#Calculate the Deviance
dev_logit = deviance(fit_logit)

#Calculate the p(dying|x=0.01)
predict(fit_logit, tibble(dose = 0.01), type = 'response')
```

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

The estimate of $\beta_{logit}$ is 1.1618949, the CI for $\beta_{logit}$ is (0.8063266,1.5174633), the deviance is 0.3787483 and $\hat{p}(dying|x = 0.01) = 0.09012$.

```r
#fit the glm with probit link
fit_probit = glm(cbind(n_dying, 30-n_dying) ~ dose, family = binomial(link = 'probit'), data = bioassy_
summary(fit_probit)
```

```
##
## Call:
## glm(formula = cbind(n_dying, 30 - n_dying) ~ dose, family = binomial(link = "probit"),
##     data = bioassy_df)
##
## 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
```

```
#calculate the lower and upper bounds of CI
probit_ci_lower = summary(fit_probit)$coefficient[2,1] - qnorm(0.975)*summary(fit_probit)$coefficient[2
probit_ci_higher = summary(fit_probit)$coefficient[2,1] + qnorm(0.975)*summary(fit_probit)$coefficient[

#Calculate the Deviance
dev_probit = deviance(fit_probit)

#Calculate the p(dying|x=0.01)
predict(fit_probit, tibble(dose = 0.01), type = 'response')
```

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

The estimate of $\beta_{probit}$ is 0.6863805 and the CI for $\beta_{probit}$ is (0.4967217,0.8760393), the deviance is 0.3136684 and $\hat{p}(dying|x = 0.01) = 0.0853078$.

```
#fit the glm with probit link
fit_cloglog = glm(cbind(n_dying, 30-n_dying) ~ dose, family = binomial(link = 'cloglog'), data = bioass
summary(fit_cloglog)
```

```
##
## Call:
## glm(formula = cbind(n_dying, 30 - n_dying) ~ dose, family = binomial(link = "cloglog"),
##     data = bioassy_df)
##
## 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
```

```r
#calculate the lower and upper bounds of CI
cloglog_ci_lower = summary(fit_cloglog)$coefficient[2,1] - qnorm(0.975)*summary(fit_cloglog)$coefficien
cloglog_ci_higher = summary(fit_cloglog)$coefficient[2,1] + qnorm(0.975)*summary(fit_cloglog)$coefficie

#Calculate the Deviance
dev_cloglog = deviance(fit_cloglog)

#Calculate the p(dying|x=0.01)
predict(fit_cloglog, tibble(dose = 0.01), type = 'response')
```

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

The estimate of $\beta_{cloglog}$ is 0.7468193and the CI for $\beta_{cloglog}$ is (0.53232,0.9613187), the deviance is 2.2304792 and $\hat{p}(dying|x = 0.01) = 0.1281601$.

```r
#Generating the table for 1a
table_1a = tibble(Model = c("Estimate of beta", "CI for beta", "Deviance", "p hat"), logit = c(1.1619,"

table_1a %>%
  knitr::kable()
```

| Model | logit | probit | cloglog |
|---|---|---|---|
| Estimate of beta | 1.1619 | 0.6864 | 0.7468 |
| CI for beta | (0.8063,1.517) | (0.4967,0.8760) | (0.5323,0.9613) |
| Deviance | 0.3787 | 0.3137 | 2.2304 |
| p hat | 0.09012 | 0.08531 | 0.1282 |

The logit model has the highest estimate of $\beta$ among three models. The interpretation for $\beta_{logit}$ is that 1 unit increase in dose will increase the log odds of dying by 1.16. Since the probit model has the smallest deviance, hence it may be the best model to fit the data. All three models have the similar predict for $\hat{p}(dying|x = 0.01)$.

### (b)

$$
\begin{aligned}
g(P(dying) = 0.5) = g(0.5) = \beta_0 + \beta_1, \\
point\ estimate = \hat{x_o} = -\frac{\hat{\beta}_0}{\hat{\beta}_1} \\
logit\ model : g(0.5) = log(0.5/(1-0.5)) = 0 \\
probit\ model : g(0.5) = qnorm(0.5) = 0 \\
C - log - log\ model : log(-log(1-0.5)) = -0.3665
\end{aligned}
$$

```r
# LD50 est and CI
beta0_l = fit_logit$coefficients[1]
beta1_l = fit_logit$coefficients[2]
betacov_l = vcov(fit_logit) # inverse fisher information
x0fit_l = -beta0_l/beta1_l

logit_ld50 = exp(x0fit_l)# point estimate of LD50
```

```r
varx0_l = betacov_l[1,1]/(beta1_l^2) + betacov_l[2,2]*(beta0_l^2)/(beta1_l^4) - 2*betacov_l[1,2]*beta0_l

logit_ld50_ci = exp(x0fit_l + c(qnorm(0.05),-qnorm(0.05))*sqrt(varx0_l)) # 90% CI for LD50
```

```r
# LD50 est and CI
beta0_p = fit_probit$coefficients[1]
beta1_p = fit_probit$coefficients[2]
betacov_p = vcov(fit_probit) # inverse fisher information
x0fit_p = -beta0_p/beta1_p

probit_ld50 = exp(x0fit_p)# point estimate of LD50

varx0_p = betacov_p[1,1]/(beta1_p^2) + betacov_p[2,2]*(beta0_p^2)/(beta1_p^4) - 2*betacov_p[1,2]*beta0_p

fit_probit_ld50_ci = exp(x0fit_p + c(qnorm(0.05),-qnorm(0.05))*sqrt(varx0_p)) # 90% CI for LD50
```

```r
# LD50 est and CI
beta0_c = fit_cloglog$coefficients[1]
beta1_c = fit_cloglog$coefficients[2]
betacov_c = vcov(fit_cloglog) # inverse fisher information
x0fit_c = (-0.3665 - beta0_c)/beta1_c

cloglog_ld50 = exp(x0fit_c)# point estimate of LD50

varx0_c = betacov_c[1,1]/(beta1_c^2) + betacov_c[2,2]*((-0.3665-beta0_c)^2)/(beta1_c^4) - 2*betacov_c[1

cloglog_ld50_ci_1 = exp(x0fit_c + c(qnorm(0.05),-qnorm(0.05))*sqrt(varx0_c)) # 90% CI for LD50
```

```r
#Generating the table for 1b
table_1b = tibble(Model = c("Estimate of LD50", "Lower CI", "Upper CI"), logit = c(logit_ld50,logit_ld5

table_1b %>%
  knitr::kable()
```

| Model | logit | probit | cloglog |
|---|---|---|---|
| Estimate of LD50 | 7.389056 | 7.435830 | 8.841402 |
| Lower CI | 5.509632 | 5.582588 | 2.698847 |
| Upper CI | 9.909583 | 9.904289 | 28.964368 |

---

**Question 2**

```r
# create a mph dataframe for question 2
mph_df = tibble(
  amount = seq(from = 10, to = 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)
)
```

```r
fit_2 = glm(cbind(enroll, offer-enroll) ~ amount, family = binomial(link = 'logit'), data = mph_df)
summary(fit_2)
```

**(a) How does the model fit the data?**

```
##
## Call:
## glm(formula = cbind(enroll, offer - enroll) ~ amount, family = binomial(link = "logit"),
##     data = mph_df)
##
## 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
hoslem.test(fit_2$y, fitted(fit_2),g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  fit_2$y, fitted(fit_2)
## X-squared = 1.6111, df = 8, p-value = 0.9907
```

Since from theb hoslem test the p value is greater than 0.05, we reject the null and conclude that the model fits data well.

```r
#coefficient for beta 1
beta_1_mph = fit_2$coefficients[2]

#calculate the lower and upper bounds of CI
beta_estimate_ci_lower = summary(fit_2)$coefficient[2,1] - qnorm(0.975)*summary(fit_2)$coefficient[2,2]
beta_estimate_ci_higher = summary(fit_2)$coefficient[2,1] + qnorm(0.975)*summary(fit_2)$coefficient[2,2
```

**(b) How do you interpret the relationship between the scholarship amount and enrollment rate? What is 95% CI?** The estimate for $\hat{\beta}$ is 0.0309504 b and the 95% CI is (0.0119785, 0.0499224) Per \$1 increase in scholarship amount,there is an 0.03095043 unit increase in log odds ratio of enrollment rate

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

$$g(p) - \hat{\beta}_0 = g(0.4) - \hat{\beta}_0 The\ estimate\ \hat{x_0} = \frac{log\frac{0.4}{1-0.4} - \hat{\beta}_0}{\hat{\beta}_1}$$

```
beta_0_mph = fit_2$coefficients[1]
betacov_mph = vcov(fit_2) # inverse fisher information

x0fit_mph = (log(0.4/0.6)-beta_0_mph)/beta_1_mph

varx0_mph = betacov_mph[1,1]/(beta_1_mph^2) + betacov_mph[2,2]*(log(0.4/0.6)-beta_0_mph)^2/(beta_1_mph^4

mph_ci = x0fit_mph + c(qnorm(0.025),-qnorm(0.025))*sqrt(varx0_mph)
```

```
#Generating the table for 2c
table_2c = tibble(estimate = x0fit_mph, ci_lower_bound = mph_ci[1], ci_upper_bound = mph_ci[2])

table_2c %>%
  knitr::kable()
```

| estimate | ci_lower_bound | ci_upper_bound |
|---|---|---|
| 40.13429 | 30.58304 | 49.68553 |

We need to provide 40.13 thousands of dollars as scholarship to get 40% yield rate and the 95% CI is (30.5830397, 49.6855327)