p8131_hw2_jsg2145

Jared Garfinkel

2/12/2020

Homework 2

This homework requires the use of logistic regression to practice fitting logistic models with given datasets.

Problem 1

This problem explores the similarities and differences between link functions in linear regression. While the logit and probit link functions return similar results, it appears that the complementary log-log link function has a wider confidence interval and larger deviance, indicating a logistical regression model may be appropriate in this case.

Problem 1i

Create a table of descriptive statistics for beta, a proxy for the effect size.

```
# Create a dataframe

dos_df <- tibble(
  dose = c(0:4),
  dead = c(2, 8, 15, 23, 27),
  alive = 30 - dead
)
```

```
# create a logistic regression

dos_logit <- dos_df %>%
  glm(data = ., cbind(dead, alive) ~ dose, family = binomial(link = "logit"))

summary(dos_logit)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ dose, family = binomial(link = "logit"),
##     data = .)
##
## 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
```

```
# Create a normal regression

dos_probit <- dos_df %>%
  glm(data = ., cbind(dead, alive) ~ dose, family = binomial(link = "probit"))

summary(dos_probit)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ dose, family = binomial(link = "probit"),
##     data = .)
##
## 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
```

```
# Create a complementary log-log regression
dos_cloglog <- dos_df %>%
  glm(data = ., cbind(dead, alive) ~ dose, family = binomial(link = "cloglog"))

summary(dos_cloglog)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ dose, family = binomial(link = "cloglog"),
##     data = .)
##
## 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
# Return confidence intervals for each regression

dos_logit_ci = dos_logit %>%
  broom::tidy() %>%
  mutate(
    lower = (estimate - qnorm(.975) * std.error),
    upper = (estimate + qnorm(.975) * std.error)) %>%
  select(term, estimate, lower, upper) %>%
  filter(term == "dose") %>%
  mutate(term = recode(term, dose = "logit"))

dos_probit_ci = dos_probit %>%
  broom::tidy() %>%
  mutate(
    lower = (estimate - qnorm(.975) * std.error),
    upper = (estimate + qnorm(.975) * std.error)) %>%
  select(term, estimate, lower, upper) %>%
  filter(term == "dose") %>%
  mutate(term = recode(term, dose = "probit"))

dos_cloglog_ci = dos_cloglog %>%
  broom::tidy() %>%
  mutate(
    lower = (estimate - qnorm(.975) * std.error),
    upper = (estimate + qnorm(.975) * std.error)) %>%
  select(term, estimate, lower, upper) %>%
  filter(term == "dose") %>%
  mutate(term = recode(term, dose = "cloglog"))

table = rbind(dos_logit_ci, dos_probit_ci, dos_cloglog_ci)
```

```r
# Return deviances for each regression

dev_logit = deviance(dos_logit)

dev_probit = deviance(dos_probit)
```

```r
dev_cloglog = deviance(dos_cloglog)

deviance = c(dev_logit, dev_probit, dev_cloglog)

table = table %>%
  mutate(deviance = c(dev_logit, dev_probit, dev_cloglog))
```

```r
# Return probability of event for dose = 0.01 on the dataframe

p_logit = predict(dos_logit, data.frame(dose=.01), se.fit=TRUE, type='response')

p_probit = predict(dos_probit, data.frame(dose=.01), se.fit=TRUE, type='response')

p_cloglog = predict(dos_cloglog, data.frame(dose=.01), se.fit=TRUE, type='response')

table = table %>%
  mutate(p_estimate = c(round(p_logit[[1]], digits = 2), round(p_probit[[1]], digits = 2), round(p_clog

table %>%
  knitr::kable(digits = 2)
```

Problem 1ii

The purpose of Problem 1ii is to use the variance-covariance matrix to calculate a 95% confidence interval of the effect size for the exposure in this dataset, dosage.

```r
# Return 95% confidence intervals for the betas and beta_0s for each regression

# logit

beta0_l=dos_logit$coefficients[1]
beta1_l=dos_logit$coefficients[2]
betacov_l=vcov(dos_logit)
x0fit_l=-beta0_l/beta1_l
est_l = exp(x0fit_l)

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/(beta
est_sd_l = c(x0fit_l,sqrt(varx0_l))
ci_l = exp(x0fit_l+c(qnorm(0.05),-qnorm(0.05))*sqrt(varx0_l))

#probit

beta0_p=dos_probit$coefficients[1]
beta1_p=dos_probit$coefficients[2]
betacov_p=vcov(dos_probit)
x0fit_p=-beta0_p/beta1_p
est_p = exp(x0fit_p)

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/(beta
est_sd_p = c(x0fit_p,sqrt(varx0_p)) # point est and se
ci_p = exp(x0fit_p+c(qnorm(0.05),-qnorm(0.05))*sqrt(varx0_p))
```

```r
#cloglog

beta0_c=dos_cloglog$coefficients[1]
beta1_c=dos_cloglog$coefficients[2]
betacov_c=vcov(dos_cloglog)
x0fit_c=(log(log(2))-beta0_c)/beta1_c
est_c = exp(x0fit_c)

varx0_c=betacov_c[1,1]/(beta1_c^2)+betacov_c[2,2]*(beta0_c^2)/(beta1_c^4)-2*betacov_c[1,2]*beta0_c/(beta
est_sd_c = c(x0fit_c,sqrt(varx0_c)) # point est and se
ci_c = exp(x0fit_c+c(qnorm(0.05),-qnorm(0.05))*sqrt(varx0_c))

dos_ci = tibble(
  term = c("logit", "probit", "cloglog"),
  estimate = c(est_l, est_p, est_c),
  lower = c(ci_l[1], ci_p[1], ci_c[1]),
  upper = c(ci_l[2], ci_p[2], ci_c[2])
)

dos_ci %>%
  knitr::kable(digits = 2)
```

term	estimate	lower	upper
logit	7.39	5.51	9.91
probit	7.44	5.58	9.90
cloglog	8.84	6.64	11.78

Problem 2

This problem focuses on how to use linear regression to determine the goodness of fit of a model with a dataset.

```r
# Create a dataframe

data = 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),
  enrolls = c(0,2,4,2,12,14,10,7,5,5,3,5,2,0,4,2,1),
  passes = offer - enrolls
)
```

Problem 2i

This question uses our knowledge of logistic regression to study the goodness of fit of the given data. A Hosmer-Lemeshow test is used for a dataset with small sample size. The Hosmer-Lemeshow and Pearson Chi-Square Residual tests show the model is a good fit.

```r
# Fit a logistic regression
```

```r
dat_glm = data %>% glm(data = ., cbind(enrolls, passes) ~ amount, family = binomial(link = 'logit'))
summary(dat_glm)
```

```
##
## Call:
## glm(formula = cbind(enrolls, passes) ~ amount, family = binomial(link = "logit"),
##     data = .)
##
## 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
# Run diagnostics for small sample sizes (Hos-Lem)

dev_glm = deviance(dat_glm)
hl = hoslem.test(dat_glm$y, fitted(dat_glm), g = 10)

hl
```

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

```r
pval = 1-pchisq(dev_glm,15)

pval
```

```
## [1] 0.7795345
```

The p-values are greater than 0.05, indicating that the model is a good fit.

Problem 2ii

```r
# Return 95% confidence interval for the estimate of beta

dat_tbl = dat_glm %>%
  broom::tidy() %>%
  mutate(
    lower = estimate - qnorm(.975) * std.error,
    upper = estimate + qnorm(.975) * std.error,
  ) %>%
  filter(term == "amount") %>%
  select(term, estimate, lower, upper)

dat_tbl %>%
  knitr::kable(digits = 2)
```

term	estimate	lower	upper
amount	0.03	0.01	0.05

The odds ratio of enrolling in this MPH program increases by 3.14, CI: (1.21, 5.12) percent per $1000 offered.

Problem 2iii

```r
beta0_mph=dat_glm$coefficients[1]
beta1_mph=dat_glm$coefficients[2]
betacov_mph=vcov(dat_glm)
x0_mph=(log(0.4/0.6) - beta0_mph) / beta1_mph
est_mph = x0_mph*1000

varx0_mph=betacov_mph[1,1]/(beta1_mph^2)+betacov_mph[2,2]*(beta0_mph^2)/(beta1_mph^4)-2*betacov_mph[1,2
est_sd_mph = c(x0_mph,sqrt(varx0_mph))
ci_mph = (x0_mph+c(qnorm(0.025),-qnorm(0.025))*sqrt(varx0_mph))*1000

ci_tbl = tibble(
  term = "logit",
  estimate = est_mph,
  lower = ci_mph[1],
  upper = ci_mph[2]
)

ci_tbl %>%
  knitr::kable(digits = 0)
```

term	estimate	lower	upper
logit	40134	27864	52404

On average, an offer of 40134 dollars would result in a 40% yield rate. With 95% confidence, an offer of between 27864 dollars and 52404 dollars would result in a 40% yield rate.