Exercise for Lecture 5 and 6

Rei Monden

2024-12-19

Exercise 5-1

- The purpose of the link function of a GLM is to determine the function of the mean that is predicted by the linear predictors.
- The identity link function models the binomial probability as a linear function of the predictors. The identity link function is not often used with the binomial parameter because binomial probability values should be between 0 and 1, but predictions from the linear component can fall outside the unit interval.

Exercise 5-2

(a)

```
HD.df <- data.frame(Yes = c(24, 35, 21, 30),
No = c(1355, 603, 192, 224),
x1 = c(0, 2, 4, 6), # 'continuous' Snoring
x2 = c(0, 1, 2, 3),
x3 = c(1, 2, 3, 4),
x4 = c(0, 2, 6, 7),
x5 = c(0, 2, 4, 5)
)
rownames(HD.df) <- c("Never", "Occasional",
"Nearly every night", "Every night")

# Fit the logistic regression (i):
logis1 <- glm(cbind(Yes, No) ~ x1,
          family = binomial(link = "logit"),
          data = HD.df)

# Fit the logistic regression (ii):
logis2 <- glm(cbind(Yes, No) ~ x2,
          family = binomial(link = "logit"),
          data = HD.df)

# Fit the logistic regression (iii):
logis3 <- glm(cbind(Yes, No) ~ x3,
          family = binomial(link = "logit"),
          data = HD.df)
```

From the above analyses, the estimated coefficients are the followings:

```
(logis1$coefficients)
```

```
## (Intercept)          x1
##  -3.7773756   0.3272648
```

```
(logis2$coefficients)
```

```
## (Intercept)          x2
##  -3.7773756   0.6545296
```

```
(logis3$coefficients)
```

```
## (Intercept)          x3
##  -4.4319052   0.6545296
```

From here, we can see that the slope (i.e., the effect of *snoring* on *heart disease*) depends on the distance between scores. Because the spacing between the x-values for codings (ii) and (iii) is the same (1 point), the estimated slopes for *snoring* in both models coincide (= 0.6545). For coding (i), the spacing between the code values doubled (2 points), hence the estimated slope for *snoring* halfed (i.e., 0.6545/2 = 0.3272).

(b)

```
# Fit the logistic regression b-(i):
logisb1 <- glm(cbind(Yes, No) ~ x4,
          family = binomial(link = "logit"),
          data = HD.df)

# Fit the logistic regression b-(ii):
logisb2 <- glm(cbind(Yes, No) ~ x5,
          family = binomial(link = "logit"),
          data = HD.df)

# Check fitted values
(logisb1$fitted.values)
```

```
##              Never         Occasional Nearly every night        Every night
##         0.02288749         0.03807029         0.10151539         0.12805716
```

```
(logisb2$fitted.values)
```

```
##              Never         Occasional Nearly every night        Every night
##         0.02050742         0.04429511         0.09305411         0.13243885
```

No, the fitted values do not seem to change much between both models.

Exercise 5-3

(a) From the model $log\mu = \alpha + \beta x$, where $x = 1$ for treatment B and $x = 0$ for treatment A, we can find one equation per treatment:

- Treatment A: $log\mu_A = \alpha + \beta(0) = \alpha$
- Treatment B: $log\mu_B = \alpha + \beta(1) = \alpha + \beta$.

From the second equation, we have that $\beta = log\mu_B - \alpha$. Now using the first equation we conclude that $\beta = log\mu_B - \log \mu_A$.

Because the difference of two logarithm is equal to the logarithm of the quocient, we get that $\beta = log\mu_B - \log \mu_A = log(\frac{\mu_B}{\mu_A})$.

Finally, exponentiating both members of the previous equality leads to $e^{\beta} = \frac{\mu_B}{\mu_A}$.

(b)

```
imp.df <- data.frame(
  imperfections = c(8, 7, 6,  6, 3,  4,  7, 2, 3, 4,
                    9, 9, 8, 14, 8, 13, 11, 5, 7, 6),
  treatment     = rep(c("A", "B"), each = 10))
reg.pos <- glm(imperfections ~ treatment,
               family = "poisson",
               data   = imp.df)
# Regression table:
round(summary(reg.pos)$coefficients, 4)
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.6094     0.1414 11.3805   0.0000
## treatmentB    0.5878     0.1764  3.3324   0.0009
```

As shown in the output above, the prediction equation is given by $log(\hat{\mu}) = 1.6094 + 0.5878x$.

Interpretation of $\hat{\beta}$*:*
Given that $exp(\hat{\beta}) = 1.8$, the mean number of imperfections is estimated to be 80% higher for treatment B.

(c) As shown on page 34, Wald's test statistic is defined as $z_W = \frac{\hat{\beta}}{SE(\hat{\beta})} = \frac{0.5878}{0.1764} = 3.3324$. This is correct, as it matches the output in question (b) (under `z value`).
Interpretation: At 5% significance level, we decide to reject the null hypothesis.

Exercise 5-4

(a)

```
reg.nopred <- glm(imperfections ~ 1,
               family = "poisson",
               data   = imp.df)
# Regression table:
round(summary(reg.nopred)$coefficients, 4)
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.9459     0.0845 23.0244        0
```

(b)

```
drop1(reg.pos, test = "LRT")
```

```
## Single term deletions
##
## Model:
## imperfections ~ treatment
##           Df Deviance     AIC    LRT  Pr(>Chi)
## <none>         16.268  94.349
## treatment  1   27.857 103.938 11.589 0.0006633 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output shows the following result:
test statistic = 11.589, df = 1, p < .001. At 5% significance level, we conclude that the intercept-only model fits significantly worse than the model including the *treatment* predictor. We therefore reject the intercept-only model.