CatData Final

Matthew Vanaman

05-19-2019

Code shown in the appendix.

5.19

(a)

$\text{logit}(\pi) = \alpha + \beta_1 M_1 + \beta_2 M_2 + \beta_3 M_3 + \beta_4 M_4 + \beta_5 M_5 + \beta_6 M_6$, with M = major, once for each level. When you want to assess the effect of a particular level of major, code that major (e.g. M_3) to 1 and the rest to zero.

(b)

The deviance goodness-of-fit test approximates the χ^2 for GLMs with binomial responses and large numbers of successes and failures. Our model meets these criteria. The p-value for the deviance test is 0.001, indicating that other possible predictors excluded from this model may not equal zero (i.e. bad fit).

(c)

The rule of thumb for standardized residuals is that residuals with aboslute values above 2 or 3 are suspect. Most of the standardized residuals given are pretty small, with the exception of the first Major. This indicates that the model in (a) with Major as the only predictor may not have good fit because it excludes another important predictor (gender).

(d)

Because the model in (a) excludes a predictor for gender, it assumes an equal probability of admission for males and females across majors. If this is true, the standardized residuals should be close to zero (they are normally distributed when the model holds). Because gender is a binary predictor in this case, we can expect the standardized residual for males to be identical to female, but with an opposite sign. In other words, whatever deviation from the expected probability there is for females, there is probability of equal magnitude in the opposite direction for males. The residual of -4.15 for males indicates that males have a lower-than-expected probability of admission, likely violating the assumption of equal probability for males and females within that major.

(e)

This illustrates Simpson's paradox: failing to account for differences across majors leads to an effect in the opposite direction. When you condition on major females have 10% lower odds of admission, yet collapsing over major, females have 84% greater odds of admission. This *could* mean that females apply to the more competitive majors relatively more often, so are accepted less often than males after conditioning on major. So when you control for major, females fare worse because the majors they are applying to are more competitive. Males on the other hand, are applying to less competitive majors, meaning that when you average

across majors of varying difficulty, it appears that males are doing worse. When you allow for the conditionality of major, males fare better because they apply in relatively higher numbers to the easier majors. This shows why it's important to keep track of whom applies to where when making group comparisons.

7.9

(a)

Table 1: Model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.6894	0.0563	101.03	0.0000
admit1	0.5821	0.0690	8.44	0.0000
female1	-2.0985	0.1155	-18.17	0.0000
major2	-0.3598	0.0887	-4.06	0.0000
major3	-0.3153	0.0830	-3.80	0.0001
major4	-0.0554	0.0787	-0.70	0.4815
major5	-0.7105	0.0946	-7.51	0.0000
major6	0.1686	0.0769	2.19	0.0284
admit1:major2	-0.0434	0.1098	-0.40	0.6928
admit1:major3	-1.2626	0.1066	-11.84	0.0000
admit1:major4	-1.2946	0.1058	-12.23	0.0000
admit1:major5	-1.7393	0.1261	-13.79	0.0000
admit1:major6	-3.3065	0.1700	-19.45	0.0000
admit1:female1	0.0999	0.0808	1.24	0.2167
female1:major2	-1.0748	0.2286	-4.70	0.0000
female1:major3	2.6651	0.1261	21.14	0.0000
female1:major4	1.9583	0.1273	15.38	0.0000
female1:major5	2.7952	0.1393	20.07	0.0000
female1:major6	2.0023	0.1357	14.75	0.0000

To get the conditional odds ratio for AF (which is AG, except the gender variable is called F in this case), you exponentiate the admit:female coefficient: exp(-0.1084) = 0.9.

Table 2: Fitted Values

Major	Myes	Mno	Fyes	Fno
1	36.27	295.73	71.73	529.27
2	8.64	206.36	16.36	353.64
3	380.25	215.75	212.75	109.25
4	243.21	279.79	131.79	137.21
5	291.68	145.32	101.32	45.68
6	317.96	350.04	23.04	22.96
Total	1278.01	1492.99	556.99	1198.01

To get the marginal odds, you can use use of the model fitted count values. In this case, we're ignoring the Major grouping. Using the totals of each column, with admission indicated by the subscrtipt, the calculation comes out to be

$$\begin{aligned} &\frac{\mathrm{F}_{no} \times \mathrm{M}_{yes}}{\mathrm{F}_{yes} \times \mathrm{M}_{no}}, \\ &= \frac{1198.01 \times 1278.01}{556.99 \times 1492.99}, \\ &= 1.84. \end{aligned}$$

This turns out to be nearly identical to the marginal odds ratio of the raw counts.

(b)

Table 3: Predictors and Standardized Residuals

admit	female	major	Std Res
0	0	1	4.03
0	0	2	0.28
0	0	3	-1.88
0	0	4	-0.14
0	0	5	-1.63
0	0	6	0.30
1	0	1	-4.03
1	0	2	-0.28
1	0	3	1.88
1	0	4	0.14
1	0	5	1.63
1	0	6	-0.30
0	1	1	-4.03
0	1	2	-0.28
0	1	3	1.88
0	1	4	0.14
0	1	5	1.63
0	1	6	-0.30
1	1	1	4.03
1	1	2	0.28
1	1	3	-1.88
1	1	4	-0.14
1	1	5	-1.63
1	1	6	0.30

The deviance for this model is 20.2, $df = 5$, $p = 0.001$. Bad fit.

As can be seen from the table above, the standardized residuals show that there doesn't seem to be much misfit about the admit and female predictors, but major, specifically at level 1, is consistently problematic.

(c)

Table 4: Model Sans Major 1

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.3261	0.0688	77.39	0.0000
admit1	0.5442	0.0858	6.34	0.0000
female1	-3.0897	0.2115	-14.61	0.0000
major3	0.0188	0.0934	0.20	0.8408
major4	0.2870	0.0890	3.22	0.0013
major5	-0.3691	0.1035	-3.57	0.0004
major6	0.5279	0.0867	6.09	0.0000
admit1:major3	-1.1401	0.1219	-9.35	0.0000
admit1:major4	-1.1946	0.1198	-9.97	0.0000
admit1:major5	-1.6131	0.1393	-11.58	0.0000
admit1:major6	-3.2053	0.1788	-17.93	0.0000
admit1:female1	-0.0307	0.0868	-0.35	0.7235
female1:major3	3.7019	0.2170	17.06	0.0000
female1:major4	2.9940	0.2179	13.74	0.0000
female1:major5	3.8190	0.2250	16.98	0.0000
female1:major6	3.0020	0.2231	13.46	0.0000

The deviance for this new model is 2.56, $df = 4$, $p = 0.77$. Fit has improved a lot.

(d)

Table 5: Logit Model Sans Major 1

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.5337	0.0875	6.10	0.0000
female1	0.2200	0.4376	0.50	0.6151
major3	-1.0693	0.1445	-7.40	0.0000
major4	-1.2377	0.1360	-9.10	0.0000
major5	-1.4907	0.1838	-8.11	0.0000
major6	-3.3035	0.2366	-13.96	0.0000
female1:major3	-0.3449	0.4607	-0.75	0.4540
female1:major4	-0.1380	0.4627	-0.30	0.7654
female1:major5	-0.4202	0.4812	-0.87	0.3826
female1:major6	-0.0311	0.5335	-0.06	0.9535

The equivalent logit model yields a conditional odds ratio between gender and admissions is exp(0.2200) = 1.25. Compare this to the OR from the previous model, which is exp(-0.0307) = 0.97. The logit model shows much less of an effect of gender on admissions than the poisson model. Given the negligibale effect of gender on admissions, it might be worth comparing a model with no gender terms. A deviance test between these models shows the the simpler model with no gender variable works just as well as the model with gender ($G^2 = 2.68$, $df = 5$, $p = 0.75$). Thus it is important to carefully consider model assumptions when choosing a test.

4.37

(a)

This is true. To figure this out, find the $\log(\hat{\pi})$ for each coding scheme:

$$\log(\hat{\pi}) = 2.06 + .87d2.40v.$$

Defendent white, victim black:

$$-4.46 = \log(0.01) = 2.06 + .87(0)2.40(1).$$

Defendent black, victim white:

$$-1.19 = \log(0.3) = 2.06 + .87(1)2.40(0).$$

Defendent black, victim black:

$$-2.06 = \log(0.13) = 2.06 + .87(0)2.40(0).$$

Defendent white, victim white:

$$-3.59 = \log(0.03) = 2.06 + .87(0)2.40(0).$$

(b)

False. An OR of 1.15 would indicate that whites have 15% greater odds of the death penalty. There's no reason why changing the coding should flip the effect since the relative relationships among the variables remain the same no matter how they are coded. Only now your coefficient will be 1.15, so to get the odds for whites you take 1/1.15, which is 0.87.

(c)

True. This is the definition of an interaction: there is no multiplicative effect of one variable on the other; the effect of one variable remains constant across all levels of the other variable. You could have also said that the estimated odds ratio between the death penalty out come and victim's race is the same for each category of defendent's race.

(d)

Nope. The intercept is on the log-odds scale.

(e)

True. Victim and defendent variables would both be coded as zero, meaning that the expected count for when v = 0 and d = 0 is just the logit with no other terms in the exponent. In other words:

$$\frac{500e^{\alpha+B_d d+B_v v}}{(1-e^{\alpha+B_d d+B_v v})},$$

$$= \frac{500e^{-2.06+0.87(0)-2.40(0)}}{(1-e^{-2.06+0.87(0)-2.40(0)})},$$

$$= \frac{500e^{-2.06}}{(1-e^{-2.06})}.$$

Appendix

```
# function for printing good-looking p-values
pvalr <- function(pvals, sig.limit = .001, digits = 3, html = FALSE) {

  roundr <- function(x, digits = 1) {
    res <- sprintf(paste0('%.', digits, 'f'), x)
    zzz <- paste0('0.', paste(rep('0', digits), collapse = ''))
    res[res == paste0('-', zzz)] <- zzz
    res
  }

  sapply(pvals, function(x, sig.limit) {
    if (x < sig.limit)
      if (html)
        return(sprintf('&lt; %s', format(sig.limit))) else
          return(sprintf('< %s', format(sig.limit)))
    if (x > .1)
      return(roundr(x, digits = 2)) else
        return(roundr(x, digits = digits))
  }, sig.limit = sig.limit)
}
```

5.19

(b)

```
# get p-value for deviance test
dev <- pvalr(pchisq(21.7, df=6, lower.tail = FALSE))
```

7.9

(a)

```
berk_admin$major <- as.factor(berk_admin$major)
berk_admin$female <- as.factor(berk_admin$female)
berk_admin$admit <- as.factor(berk_admin$admit)

# fit the model
fit <- glm(count ~ admit + female + major +
             admit:major + admit:female + major:female,
           family=poisson,
           data=berk_admin)

print(xtable::xtable(summary(fit),
                     caption = "Model",
                     align = "lrrrr"),
      caption.placement = "top",
      latex.environments = "flushleft")
```

```r
# get fitted values
fitted <- as.data.frame(predict(fit, type = "response"))
berk_admin <- cbind(berk_admin,round(fitted,2))

# make a table of fitted values
Mno <- berk_admin[1:6,5]
Fno <- berk_admin[7:12,5]
Myes <- berk_admin[13:18,5]
Fyes <- berk_admin[19:24,5]
sum_Mno <- round(sum(berk_admin[1:6,5]),2)
sum_Fno <- round(sum(berk_admin[7:12,5]),2)
sum_Myes <- round(sum(berk_admin[13:18,5]),2)
sum_Fyes <- round(sum(berk_admin[19:24,5]),2)

margins <- as.data.frame(matrix(c(1,2,3,4,5,6, "", "Total",
                                  Myes, "", sum_Myes,
                                  Mno, "", sum_Mno,
                                  Fyes, "", sum_Fyes,
                                  Fno, "", sum_Fno),
                                nrow=8, ncol=5))

names(margins) <- c("Major", "Myes", "Mno", "Fyes", "Fno")
print(xtable::xtable(margins,
                     caption = "Fitted Values",
                     align = "lrrrrr"),
      caption.placement = "top",
      include.rownames=FALSE,
      latex.environments = "flushleft")

# get marg odds
marg_OR <- round((sum_Fno * sum_Myes)/(sum_Fyes * sum_Mno),2)
```

(b)

```r
# deviance test
dev <- round(fit$deviance,2) # deviance
df <- fit$df.residual # df
p <- pvalr(pchisq(dev, df=5, lower.tail = FALSE)) # p-value

# standardized residuals
SRs <- as.data.frame(resid(fit, type="pearson")/sqrt(1 - hatvalues(fit)))
colnames(SRs) <- "Std Res"
berk_admin <- cbind(berk_admin, SRs)

print(xtable::xtable(berk_admin[,c(2:4,6)],
                     caption = "Predictors and Standardized Residuals",
                     align = "rcccr"),
      caption.placement = "top",
      include.rownames=FALSE,
      latex.environments = "flushleft")
```

(c)

```
# fit the model with the dropped level
fit_drop <- glm(count ~ admit + female + major +
                  admit:major + admit:female + major:female,
                family=poisson,
                subset = (major != 1),
                data=berk_admin)

print(xtable::xtable(summary(fit_drop),
                     caption = "Model Sans Major 1",
                     align = "lrrrr"),
      caption.placement = "top",
      latex.environments = "flushleft")

# deviance test for drop model
dev_drop <- round(fit_drop$deviance,2) # deviance
df_drop <- fit_drop$df.residual # df
p_drop <- pvalr(pchisq(dev_drop, df=5, lower.tail = FALSE)) # p-value
```

(d)

```
# run logit model
logit <- glm(admit ~ female + major +
               major:female,
             weights = count,
             family=binomial,
             subset = (major != 1),
             data=berk_admin)

print(xtable::xtable(summary(logit),
                     caption = "Logit Model Sans Major 1",
                     align = "lrrrr"),
      caption.placement = "top",
      latex.environments = "flushleft")

# get odds ratio from logit model
OR_logit <- round(exp(0.2200),2)

# fit logit without gender term
logit_nogender <- glm(admit ~ major,
                      weights = count,
                      family=binomial,
                      subset = (major != 1),
                      data=berk_admin)

# deviance test between logit model and logit model without gender
dev_comp <- round(logit_nogender$deviance - logit$deviance,2) # deviance
df_comp <- logit_nogender$df.residual - logit$df.residual # df
p_comp <- pvalr(pchisq(dev_comp, df = df_comp, lower.tail = FALSE)) # p-value
```