Categorical Data Analysis - Assignment 4

Hyungyeong Hong (2022KU0104)

2022-06-09

Note 1) Show your code and output
Note 2) Conducting a test includes
a) write the hypotheses, b) calculate the test statistic, c) find the p-value, and d) write the conclusion

```
setwd("/Users/hyungyeonghong/Desktop/2022_Spring/CDA_2022/Assignment_4")
rm(list = ls(all = TRUE))
```

[Question 1]

For a survey data on X = gender (male=1, female=2) and Y = belief in an afterlife (no=1, yes=2), the output below shows results of fitting the independence loglinear model.

```
-------------------------------------------------------
                Estimate   Std. Error
 Intercept        4.5849       0.0752
 genderfemales    0.2192       0.0599
 beliefyes        1.4165       0.0752
-------------------------------------------------------
```

[Question 1-(a)]

The deviance is 0.82 with df = 1. Write the hypothesis, p-value and the corresponding conclusion in terms of the context.

- *(i) Setting the hypothesis*
H_0 : *the independence loglinear model is better* *vs* H_a : *the saturated model is better*

- *(ii) Calculating the p-value*
Since the deviance is 0.82 with df = 1, the p-value can be calculated as,

```
1-pchisq(0.82, df = 1)
```

```
## [1] 0.3651802
```

- *(iii) Conclusion*
Since the p-value is larger than 0.05, we cannot reject H_0, which means that the current independent loglinear model is better.
Here, the saturated model has only one more term, the XY association term.
Therefore, the result of the test indicates that XY association does not exist.
In other words, gender seems to be independent of belief in afterlife.

[Question 1-(b)]

Report the estimates $\{\hat{\lambda}_j^Y\}$ for $j = 1$ and 2. Interpret $\hat{\lambda}_1^Y - \hat{\lambda}_2^Y$.
Since j = 1 (belief : no) is the baseline, $\hat{\lambda}_1^Y = 0$ due to the constraint.

For j = 2 (belief : yes), $\hat{\lambda}_2^Y = 1.4165$.
$logit[P(Y = 1 \,||X = i)] = \log \frac{P(Y=1 \mid X=i)}{P(Y=2 \mid X=i)} = \lambda_1^Y - \lambda_2^Y$ where Y = belief in an afterlife (no = 1, yes = 2)
$\hat{\lambda}_1^Y - \hat{\lambda}_2^Y$ represents the log odds of not believing in afterlife, given gender.
Therefore, the estimated odds of not believing in afterlife is exp(-1.4165) = 0.2425615, given gender.

[Question 1-(c)]

Calculate the 95% Wald confidence interval for $\hat{\lambda}_2^Y$.
The lower bound is,

```
1.4165 - qnorm(1-0.025)*0.0752
```

```
## [1] 1.269111
```

and the upper bound is,

```
1.4165 + qnorm(1-0.025)*0.0752
```

```
## [1] 1.563889
```

[Question 2]

Use the "DeathPenalty.dat" which is downloadable from the BlackBoard. (See Slide#58 in Chapter 2 for the detailed variable names in the data.)

```
rm(list = ls(all = TRUE))
Penalty <- read.table("DeathPenalty.dat", header = TRUE); Penalty
```

```
##       D     V   P count
## 1 white white yes    53
## 2 white white  no   414
## 3 black white yes    11
## 4 black white  no    37
## 5 white black yes     0
## 6 white black  no    16
## 7 black black yes     4
## 8 black black  no   139
```

[Question 2-(a)]

Fit a homogeneous association model (DV, DP, PV) where the baseline levels are Dblack, Vblack, and Pyes. Show the R output for the estimated coefficients.
We first change the baseline level of each categorical variable.

```
Penalty$D <- relevel(as.factor(Penalty$D), ref = "black")
Penalty$V <- relevel(as.factor(Penalty$V), ref = "black")
Penalty$P <- relevel(as.factor(Penalty$P), ref = "yes")
```

After changing the baseline level of each categorical variable, we fit the homogeneous association model.

```
fit.HA <- glm(count ~ D*V + D*P + P*V, family = poisson, data = Penalty)
summary(fit.HA)
```

```
##
## Call:
## glm(formula = count ~ D * V + D * P + P * V, family = poisson,
##     data = Penalty)
```

```
##
## Deviance Residuals:
##        1         2         3         4         5         6         7         8
##  0.02505  -0.00895  -0.05463   0.03000  -0.60362   0.04572   0.09251  -0.01545
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.3397     0.5004   2.677  0.00742 **
## Dwhite         -3.0424     0.4484  -6.785 1.16e-11 ***
## Vwhite          1.0746     0.5750   1.869  0.06163 .
## Pno             3.5961     0.5069   7.094 1.30e-12 ***
## Dwhite:Vwhite   4.5950     0.3135  14.656  < 2e-16 ***
## Dwhite:Pno      0.8678     0.3671   2.364  0.01807 *
## Vwhite:Pno     -2.4044     0.6006  -4.003 6.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1225.07955  on 7  degrees of freedom
## Residual deviance:    0.37984  on 1  degrees of freedom
## AIC: 52.42
##
## Number of Fisher Scoring iterations: 3
```

[Question 2-(b)]

Report the estimated conditional odds ratio between D and P.
Since the conditional log odds ration between D and P is 0.8678,
the estimated conditional odds ratio between D and P is

```
exp(0.8678)
```

```
## [1] 2.381665
```

[Question 2-(c)]

Considering P (=death penalty) as a response variable, fit an equivalent logistic regression model with the homogeneous association model. (You will need to modify the data to fit the logistic model.) Then, write the fitted equation for the logistic regression model.

```
library(tidyverse)
```

`spread()` function helps us to modify the data for fitting logistic regression model.

```
Penalty2 <- Penalty %>% spread(P, count); Penalty2
```

```
##       D     V yes  no
## 1 black black   4 139
## 2 black white  11  37
## 3 white black   0  16
## 4 white white  53 414
```

Then, we fit the logistic regression model.
The fitted logistic regression model is given as $logit[P(Y=1)] = 3.5961 + 0.8678D - 2.4044V$.

```
fit.logit <- glm(no/(no+yes) ~ D + V, family = binomial, weights = no+yes, data = Penalty2)
summary(fit.logit)
```

```
##
## Call:
## glm(formula = no/(no + yes) ~ D + V, family = binomial, data = Penalty2,
##     weights = no + yes)
##
## Deviance Residuals:
##        1         2         3         4
## -0.09379   0.06232   0.60535  -0.02660
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.5961     0.5069   7.094 1.30e-12 ***
## Dwhite        0.8678     0.3671   2.364   0.0181 *
## Vwhite       -2.4044     0.6006  -4.003 6.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22.26591  on 3  degrees of freedom
## Residual deviance:  0.37984  on 1  degrees of freedom
## AIC: 19.3
##
## Number of Fisher Scoring iterations: 4
```

[Question 3]

The following table is based on automobile accident records supplied by the state of Florida Department of Highway Safety and Motor Vehicles. Subjects were classified by whether they were wearing a seat belt (S), whether ejected (E), and whether killed (K). (Fatal indicates a driver was killed.)

Safety Equipment in Use	Whether Ejected	Injury	
		Nonfatal	Fatal
Seat belt	Yes	1,105	14
	No	411,111	483
None	Yes	4,624	497
	No	157,342	1008

Source: Florida Department of Highway Safety and Motor Vehicles.

```
rm(list = ls(all = TRUE))
```

[Question 3-(a)]

First, make a dataset according to the table. Code the variable names as S, E and K. Print the data. (That is, show the data set with the correct variable names.)

```
S <- c(rep("Seat belt", 4), rep("None", 4))
E <- rep(c("Yes", "Yes", "No", "No"), 2)
```

```r
K <- rep(c("Nonfatal", "Fatal"), 4)
count <- c(1105, 14, 411111, 483, 4624, 497, 157342, 1008)
Accident <- data.frame(S, E, K, count); Accident
```

```
##           S   E        K  count
## 1 Seat belt Yes Nonfatal   1105
## 2 Seat belt Yes    Fatal     14
## 3 Seat belt  No Nonfatal 411111
## 4 Seat belt  No    Fatal    483
## 5      None Yes Nonfatal   4624
## 6      None Yes    Fatal    497
## 7      None  No Nonfatal 157342
## 8      None  No    Fatal   1008
```

[Question 3-(b)]

Fit a loglinear model that describes the data well. (Attempt several possible models and choose one. Show the estimates from the final model only.)

The best model was homogeneous association model, as it had the smallest deviance and AIC value.

```r
fit.HA <- glm(count ~ S*E + S*K + E*K, family = poisson, data = Accident)
summary(fit.HA)
```

```
##
## Call:
## glm(formula = count ~ S * E + S * K + E * K, family = poisson,
##     data = Accident)
##
## Deviance Residuals:
##        1         2         3         4         5         6         7         8
##  0.20704  -1.59987  -0.01071   0.31400  -0.10095   0.30951   0.01731  -0.21583
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           6.92251    0.03110  222.56   <2e-16 ***
## SSeat belt           -0.75682    0.05394  -14.03   <2e-16 ***
## EYes                 -0.72784    0.05345  -13.62   <2e-16 ***
## KNonfatal             5.04362    0.03120  161.65   <2e-16 ***
## SSeat belt:EYes      -2.39964    0.03334  -71.97   <2e-16 ***
## SSeat belt:KNonfatal  1.71732    0.05402   31.79   <2e-16 ***
## EYes:KNonfatal       -2.79779    0.05526  -50.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1.6249e+06  on 7  degrees of freedom
## Residual deviance: 2.8540e+00  on 1  degrees of freedom
## AIC: 93.853
##
## Number of Fisher Scoring iterations: 3
```

[Question 3-(c)]

Since the sample size is large, goodness-of-fit statistics are large unless the model fits very well. Calculate the dissimilarity index for the model you found in (b).
The dissimilarity index for the homogeneous association model is,

```
sum(abs(Accident$count - fitted(fit.HA))) / (2*sum(Accident$count))
```

```
## [1] 4.767967e-05
```

[Question 3-(d)]

Manually calculate the fitted (expected) counts for those who used seatbelt, ejected and killed from the homogeneous association model. Do not use the function fitted.
For the given case, the level of categorical variables are S = Seat Belt, E = Yes, K = fatal.
The estimated intercept value is 6.92251.
The estimated coefficient for S = Seat Belt is -0.75682, E = Yes is -0.72784 and K = fatal is 0 since fatal is the baseline level for the variable K.
Moreover, the estimated coefficient for the interaction of S and E when S = Seat Belt and E = Yes is -2.39964.
Therefore the fitted(expected) counts for those who used seat belt, ejected and killed is,

```
exp(6.92251-0.75682-0.72784-2.39964)
```

```
## [1] 20.86786
```

[Question 3-(e)]

Manually calculate the fitted (expected) counts for those who did not use seatbelt, not ejected, and killed from the homogeneous association model. Do not use the function fitted.
For the given case, the level of the categorical variables are S = None, E = No and K = fatal.
The estimated intercept value is 6.92251.
The estimated coefficients for S = None, E = No and K = Fatal are all 0 since they are the baseline level of each categorical variable.
Since all the levels are baselines, there are no interaction term to take into account(they are all set to 0 due to constraints).
Therefore the fitted(expected) counts for those who did not use seatbelt, not ejected, and killed is,

```
exp(6.92251)
```

```
## [1] 1014.864
```

[Question 4]

The following table shows results from a matched case-control study. A study of effects on birthweight matched each case in which the child was underweight with a control in which the child had normal weight. The mothers, who were matched according to their age, were asked whether they were smokers (x= 0, no; x= 1, yes).

Normal Birth Weight (Controls)	Low Birth Weight (Cases)	
	Nonsmokers	Smokers
Nonsmokers	159	22
Smoker	8	14

Source: Partly based on data in B. Mukherjee, I. Liu, and S. Sinha, *Statist. Medic.* **26**: 3240–3257 (2007).

You will conduct a McNemar test to see whether the smoking status and low birth weight are related by following the sequence of questions.

```
rm(list = ls(all = TRUE))
```

[Question 4-(a)]

Write the null hypothesis.
The null hypothesis for the McNemar test is,
$H_0 : P(Normal\ Birth\ Weight = Nonsmokers) = P(Low\ Birth\ Weight = Nonsmokers)$
Or, the null hypothesis can be set as H_0 :the population proportions of nonsmokers are identical for both low birth weight and normal birth weight, equivalently.

[Questuin 4-(b)]

Find the test statistic and p-value.
We first generate the table which is given above in order to conduct the McNemar test.

```
BirthWeight <- matrix(c(159, 8, 22, 14), nrow = 2, ncol = 2); BirthWeight
```

```
##      [,1] [,2]
## [1,]  159   22
## [2,]    8   14
```

Then, we apply the McNemar test on the data by using the `mcnemar.test()` function.
The test statistic is given as 6.5333 and the corresponding p-value is 0.01059, which is less than 0.05.

```
mcnemar.test(BirthWeight, correct = FALSE)
```

```
##
##  McNemar's Chi-squared test
##
## data:  BirthWeight
## McNemar's chi-squared = 6.5333, df = 1, p-value = 0.01059
```

Or we can calculate the test statistic and the p-value without using the function, as the sum of off-diagonal elements in the table exceeds 10.

```
n12 <- BirthWeight[1, 2]; n21 <- BirthWeight[2, 1]
n.off <- n12 + n21

Z.stat <- (n12 - 0.5*n.off) / sqrt(n.off * 0.5 * 0.5); Z.stat
```

```
## [1] 2.556039
```

If we square the z-test statistic value, we can find the chi-square test statistic value.

```
Chi.stat <- Z.stat^2; Chi.stat
```

```
## [1] 6.533333
```

Then, the p-value for the test is,

```
1- pchisq(Chi.stat, df = 1)
```

```
## [1] 0.01058714
```

[Question 4-(c)]

Write the conclusion in terms of the context (under the significance level 0.05).
Since the p-value is less than 0.05, we reject the null hypothesis of

$H_0 : P(Normal\ Birth\ Weight = Nonsmokers) = P(Low\ Birth\ Weight = Nonsmokers)$
Therefore, the smoking status of a mother and the low birth weight of the child are related.

[Question 5]

A recent General Social Survey asked subjects whether they believed in heaven and whether they believed in hell.

Believe in Heaven	Believe in Hell	
	Yes	No
Yes	833	125
No	2	160

```
rm(list = ls(all = TRUE))
```

[Question 5-(a)]

Test the hypothesis that the population proportions answering yes were identical for heaven and hell. Here, perform McNemar test using the standardized normal test statistic. Calculate the test statistic and provide the two-sided p-value only.

- *i) Setting the hypotheses*

The hypotheses are
$H_0 : P(Belive\ in\ Heaven = Yes) = P(Believe\ in\ Hell = Yes)$
$vs \quad H_1 : P(Belive\ in\ Heaven = Yes) \neq P(Believe\ in\ Hell = Yes)$
Or, the hypotheses can be set as
H_0 : the population proportions for answering yes for heaven and hell were identical
$vs\ H_1$: the population proportions answering yes for heaven and hell were not identical, equivalently.

- *ii) Calculating the test statistic*

We first need to generate the table in order to conduct the McNemar test.

```
Belief <- matrix(c(833, 2, 125, 160), nrow = 2, ncol = 2); Belief
```

```
##      [,1] [,2]
## [1,]  833  125
## [2,]    2  160
```

Then, we conduct the McNemar test.
Since the sum of the off-diagonal elements is 127, normal approximation is possible.
Therefore, we use the standardized normal test statistic for the McNemar test, which can be calculated as,

```
n12 <- Belief[1, 2]; n21 <- Belief[2, 1]
n.off <- n12 + n21

Z.stat <- (n12 - 0.5*n.off) / sqrt(n.off * 0.5 * 0.5); Z.stat
```

```
## [1] 10.91449
```

The standardized normal test statistic can be calculated by the formula given below.

```
(n12 - n21) / sqrt(n.off)
```

```
## [1] 10.91449
```

- *iii) finding the p-value*

Since we have calculated the normal test statistic, we now find the p-value.
The two-sided p-value can be calculated as,

```
2*(1- pnorm(Z.stat))
```

```
## [1] 0
```

- *iv) Conclusion* Since the p-value is less than 0.05, we reject the null hypothesis.
Therefore, we can conclude that the population proportions answering yes for heaven and hell were not identical.

[Question 5-(b)]

Find a Wald 95% confidence interval for the difference between the population proportions.
We first get the values from the table that are needed to find a 95% Wald CI.

```
n.row <- rowSums(Belief)[1]
n.col <- colSums(Belief)[1]
n.total <- sum(Belief)
p.row <- n.row/n.total
p.col <- n.col/n.total
std.err <- sqrt(n.off - ((n12-n21)^2)/n.total)/n.total
```

The lower bound is,

```
(p.row - p.col) - qnorm(1-0.025) * std.err
```

```
## [1] 0.09117856
```

The upper bound is,

```
(p.row - p.col) + qnorm(1-0.025) * std.err
```

```
## [1] 0.1284643
```