Gotchas in R's Logistic Regression output

Sri Seshadri

10/17/2017

1. Introduction

This document illustrates the discrepancies in output of glm() for logistic regression in R. The output of Minitab (not shown here) is used as a reference.

2. Data

The data used is from page 427 of **"Introduction to Linear Regression"" 5th Edition by Montgomery and Vining**. Datasets used in the book are available at http://www.winzip.com/prod_down.htm. The R code to analyze the data is also provided in page 460 of the book.

The data used in the illustration is **Pneumoconiosis**. The dataset has long column names; for convenience the column names are shortened.

```
df <- readxl::read_xls(path = "data-ex-13-1 (Pneumoconiosis).xls", sheet = 1,
    col_names = T)
# rename columns for convenience
colnames(df) <- c("Years", "Cases", "Miners", "Proportion")
# remove Proportion column
df.cleaned <- df[, -4]
```

3. Logistic regression

The y piece of the *"formula"* argument in *glm()* in R can be supplied in 2 ways for "binomial" family ...

1. 2 column matrix - The first column denoting the number of successes and the second being the number of failures
2. Variable of type *factor*

We'll use both the methods and compare the results.

3.1. Two (2) column matrix

The output of the 2 column matrix as response is supplied to glm(). The results for **eta** i.e.

$$X\beta$$

is identical to the one from Minitab shown in page 427 of Montogomery et.al. The Deviance and Likelyhood Ratio (LR) test output also matches with the book.

```
y <- cbind(df.cleaned$Cases, df.cleaned$Miners - df.cleaned$Cases)
x <- df.cleaned$Years

fit1 <- glm(y ~ x, family = binomial)
summary.glm(fit1)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6625  -0.5746  -0.2802   0.3237   1.4852
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.79648    0.56859  -8.436  < 2e-16 ***
## x            0.09346    0.01543   6.059 1.37e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 56.9028  on 7  degrees of freedom
## Residual deviance:  6.0508  on 6  degrees of freedom
## AIC: 32.877
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fit1)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev
## NULL                     7     56.903
## x     1   50.852         6      6.051
```

3.1.1 Discrepancy

However, the loglikelyhood result does not match the one in the book. The loglikelihood is -109.664 in the book and R reports it as -14.4386095. The deviance is expected to be -2(log likelihood), which is 28.8772189. But the above software output shows deviance as 6.0507748

```
logLik(fit1)
```

```
## 'log Lik.' -14.43861 (df=2)
```

```
deviance(fit1)
```

```
## [1] 6.050775
```

We also know that the loglikelihood function for repeated observations is

$$lnL(y,\beta) = \sum_{i=1}^{n} y_i X_i'\beta - \sum_{i=1}^{n} n_i ln[1 + exp(X_i'\beta)]$$

$$
\begin{aligned}
lnL(y,\beta) &= \sum_{i=1}^{n} y_i ln(\pi_i) + \sum_{i=1}^{n} n_i ln(1-\pi_i)] - \sum_{i=1}^{n} y_i ln(1-\pi_i) \\
&= \sum_{i=1}^{n} y_i ln(\pi_i) + \sum_{i=1}^{n} (n_i - y_i) ln(1-\pi_i)
\end{aligned}
$$

Where y_i is the number of success observed for the ith observation and n_i is the number of trials at each observation. Lets compute the loglikelihood by hand below:

```
loglikelihood <- sum(df.cleaned$Cases * (coef(fit1)[1] + coef(fit1)[2] *
    x)) - sum(df.cleaned$Miners * log(1 + exp(coef(fit1)[1] + coef(fit1)[2] *
    x)))

print(loglikelihood)
```

```
## [1] -109.6637
```

it is seen that the hand calculation and logLik() output differs.

Is the discrepancy due to the logLik() not weighting by the n_i ?

```
loglikelihood_test <- sum((coef(fit1)[1] + coef(fit1)[2] * x)) - sum(log(1 +
    exp(coef(fit1)[1] + coef(fit1)[2] * x)))

print(loglikelihood_test)
```

```
## [1] -17.7582
```

The result above for a doctored likelihood hand calculation does not yield the same as logLik()

Therefore the logLik() function is not yielding the correct results. Use this function with caution.

3.2. Dichotomous (factor) variable as response.

Lets create a version of dataset that makes the response variables 0s (non severe cases) and 1s (severe cases), as below:

```
df2 <- data.frame(y = as.factor(c(rep(1, 8), rep(0, 8))), years = rep(df.cleaned$Years,
    2), counts = c(df.cleaned$Cases, df.cleaned$Miners - df.cleaned$Cases))

knitr::kable(df2, caption = "Factor response")
```

Table 1: Factor response

y	years	counts
1	5.8	0
1	15.0	1
1	21.5	3
1	27.5	8
1	33.5	9
1	39.5	8
1	46.0	10
1	51.5	5
0	5.8	98

y	years	counts
0	15.0	53
0	21.5	40
0	27.5	40
0	33.5	42
0	39.5	30
0	46.0	18
0	51.5	6

Let's run the glm fit with weights as counts…

```
fit2 <- glm(formula = y ~ years, family = binomial, weights = counts, data = df2)
summary.glm(fit2)
```

```
##
## Call:
## glm(formula = y ~ years, family = binomial, data = df2, weights = counts)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.1437  -2.8730  -0.8313   4.2021   6.1041
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.79648    0.56858  -8.436  < 2e-16 ***
## years        0.09346    0.01543   6.059 1.37e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.18  on 14  degrees of freedom
## Residual deviance: 219.33  on 13  degrees of freedom
## AIC: 223.33
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fit2)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##       Df Deviance Resid. Df Resid. Dev
## NULL                     14     270.18
## years  1   50.852        13     219.33
```

3.2.1 Discrepancy in fit statistics.

Though the regression coefficients for η (linear regression)is identical to that of in section 3.1. The fit statistics like Deviance and AIC are different. Now how does the logLik() output look like?

```
logLik(fit2)
```

```
## 'log Lik.' -109.6637 (df=2)
```

```
deviance(fit2)
```

```
## [1] 219.3273
```

It is seen that the log likelihood function is reporting the correct and expected result per the hand calculation in section 3.1.1.

4. Conclusion

When comparing fits that's based on differernt arrangements of the data, one must be careful in interpreting deviance and AIC statistics. However the Likelihood ratio tests (ggodness of fit) and the odds ratio remain unaffected.