CatData HW 4

Matthew Vanaman

03/18/19

3.15 (do not verify for 3.15d)

3.9

(a)

Answer: $\text{logit}[\hat{P}(Y = 1)] = 0.0532x - 3.5561$. In this example, x is income, Y is possession of travel card (0 = does not possess, 1 = does possess).

(b)

Answer: $\hat{\beta}$ means, in this case, that as income increases, the probability of possessing a travel card increases (because $\hat{\beta}$ is positive).

(c)

Answer: First, we know that $\hat{\pi}$ denotes $P(Y = 1)$, and that the logit function is $\log\left[\frac{P(Y=1)}{1-P(Y=1)}\right]$. When you substitute 0.50 in for $P(Y = 1)$, you get 0:

```
log(0.50 / (1 - 0.50))
```

```
[1] 0
```

To show when the etsimated π is 0.5, we just need to solve for x:

```
solveset(Eq(-3.556 + 0.0532*x)), 0.50
```

```
({66.8421052631579}, 0.5)
```

Therefore, when x = 66.84, $\hat{\pi} = 0.50$. According to the model, there is a 50/50 shot of owning a travel card for those who make around 66.86 million lira.

```
italianData <- as.data.frame(matrix(c(24, 27, 28, 29, 30, 31, 32, 33,
                                      34, 35, 38, 39, 40, 41, 42, 45,
                                      48, 49, 50, 52, 59, 60, 65, 68,
                                      70, 79, 80, 84, 94, 120, 130,

                                      1, 1, 5, 3, 9, 5, 8, 1,
                                      7, 1, 3, 2, 5, 2, 2, 1,
                                      1, 1, 10, 1, 1, 5, 6, 3,
                                      5, 1, 1, 1, 1, 6, 1,

                                      0, 0, 2, 0, 1, 1, 0, 0,
                                      1, 1, 1, 0, 0, 0, 0, 1,
                                      0, 0, 2, 0, 0, 2, 6, 3,
```

```r
                                   3, 0, 0, 0, 0, 6, 1), ncol = 3))
colnames(italianData) <- c("Inc", 'NumCases', 'CreditCards')
knitr::kable(italianData)
```

Inc	NumCases	CreditCards
24	1	0
27	1	0
28	5	2
29	3	0
30	9	1
31	5	1
32	8	0
33	1	0
34	7	1
35	1	1
38	3	1
39	2	0
40	5	0
41	2	0
42	2	0
45	1	1
48	1	0
49	1	0
50	10	2
52	1	0
59	1	0
60	5	2
65	6	6
68	3	3
70	5	3
79	1	0
80	1	0
84	1	0
94	1	0
120	6	6
130	1	1

```r
q9_fit <- glm(cbind(CreditCards,NumCases - CreditCards) ~ Inc, italianData, family=binomia
"glm((CreditCards,NumCases - CreditCards) ~ Inc,  family=binomial(link='logit'))"
```

```
[1] "glm((CreditCards,NumCases - CreditCards) ~ Inc,  family=binomial(link='logit'))"
```

```r
summary(q9_fit)$coefficients
```

```
               Estimate Std. Error   z value         Pr(>|z|)
(Intercept) -3.55611161 0.71689896 -4.960408 0.0000007034514
Inc          0.05317877 0.01314222  4.046408 0.0000520095124
```

3.15

(a)

Answer: The intercept in this case is whites (white = 0). To calculate the estimate population mean, just exponentiate the y-intercept by itself (or more accurately, the effect of white when black = 0):

```
exp(-2.38 + (1.733 * 0))
```

```
[1] 0.09255058
```

To get the estimated population means for black, add black to the model:

```
exp(-2.38 + (1.733 * 1))
```

```
[1] 0.5236143
```

(b)

The wald confidence interval for poisson data is $\hat{\beta} \pm z_{\alpha/2}(SE)$. $\hat{\beta}$ is the log of the ratio between the two means. Therefore:

```
# CI Lower:
log(0.522/0.092) - 1.96 * 0.147
```

```
[1] 1.447759
```

```
# CI Upper:
log(0.522/0.092) + 1.96 * 0.147
```

```
[1] 2.023999
```

(c)

The negative binomial is model is more believable because overdispersion is likely biasing our Poisson coefficients. We know we have overdispersion because in Poisson models, the mean should equal the variance, but in our case the variances for both blacks (1.150) and whites (0.155) are much, much greater taen their group means (0.522 and 0.092, respectively).

(d)

$\hat{D}$, the dispersion index of the negative binomial model, is 4.94, which indicates a huge discrepancy between the observed dispersion and the dispersion assumed by the Poisson model (i.e. $D = 0$). If $\hat{D}$ had turned out to be close to zero, then Poisson would be appropriate.

Question 3, AD Data

Answer:

```
q3Data <- as.data.frame(matrix(c(730,130,860,
                      100, 40, 140,
                      830,170,1000), ncol = 3))
```

```r
colnames(q3Data) <- c("NLD", 'LD', 'Margin')
rownames(q3Data) <- c('NAD', 'AD','Margin')
knitr::kable(q3Data)
```

	NLD	LD	Margin
NAD	730	100	830
AD	130	40	170
Margin	860	140	1000

AR is 0.11, CI around AR is (0.04, 0.18)
RR is 1.95, log RR is 0.17, CI around log RR is (1.63, 2.28)
OR is 2.25, log OR is 0.81, CIs around log OR is (0.40, 1.22)

Check with GLM

```r
q3_mod <- glm(cbind(c(40, 130),c(100, 730)) ~ as.factor(c(1, 0)), q3Data, family = binomia
q3_mod$coefficients
```

```
        (Intercept) as.factor(c(1, 0))1
         -1.7255101           0.8092194
```

The coefficient is the logit, so exponentiate the coeficient to get the odds ratio:

```r
exp(q3_mod$coefficients)
```

```
        (Intercept) as.factor(c(1, 0))1
          0.1780822           2.2461538
```

Double check the confidence interval as well:

```r
confint(q3_mod)
```

```
Waiting for profiling to be done...

                         2.5 %    97.5 %
(Intercept)          -1.9163041 -1.542830
as.factor(c(1, 0))1   0.3900145  1.214725
```

Work

Attributable Risk and CIs for AR

Attributable risk $= P(LD|AD) - P(LD|NAD)$. In other words, the additional risk for having a learning disability one faces if one has anxiety vs not:

```r
(40/170)
```

```
[1] 0.2352941
```

```r
(100/830)
```

```
[1] 0.1204819
```

```r
0.2352941 - 0.1204819
```

```
[1] 0.1148122
```

The standard error is $\sqrt{\left(\frac{P_1(1-P_1)}{n_1}+\frac{P_2(1-P_2)}{n_2}\right)}$

```r
# get SE
((0.2352941 * (1 - 0.2352941) / 170) +
 (0.1204819 * (1 - 0.1204819) / 830))^0.5
```

```
[1] 0.0344396
```

```r
# CI Lower
0.11 - (1.96 * 0.0344396)
```

```
[1] 0.04249838
```

```r
# CI Upper
0.11 + (1.96 * 0.0344396)
```

```
[1] 0.1775016
```

Risk Ratio and CIs for log RR

Risk ratio $= \dfrac{P(LD|AD)}{P(LD|NAD)}$

```r
0.2352941 / 0.1204819
```

```
[1] 1.952941
```

But risk ratios are not central-limit-theorem friendly, so in order to make inferences we have to take the natural log of the risk ratio and get confidence intervals around *that*:

```r
log(0.2352941 / 0.1204819)
```

```
[1] 0.6693367
```

The SE is given as $\sqrt{\left(\frac{1-P_1}{n_1P_1}+\frac{1-P_2}{n_2P_2}\right)}$:

```r
# Get SE
sqrt((1 - 0.2352941) / (170 * 0.2352941) + (1 - 0.1204819) / (830 * 0.1204819))
```

```
[1] 0.1670713
```

```r
# CI lower
1.952941 - (1.96 * 0.1670713)
```

```
[1] 1.625481
```

```r
# CI upper
1.952941 + (1.96 * 0.1670713)
```

```
[1] 2.280401
```

Odds Ratio and CIs for log OR

Odds ratio $= \frac{P(\text{LD}|\text{AD})}{P(\text{LD}|\text{NAD})} \times \frac{1 - P(\text{LD}|\text{NAD})}{1 - P(\text{LD}|\text{AD})}$:

```
(0.2352941 / 0.1204819) * ((1 - 0.1204819) / (1 - 0.2352941))
```

```
[1] 2.246154
```

Odds ratio is not so great for inferences due to long tail; get CIs for log odds instead.

The SE is $\sqrt{\frac{1}{n_{11}} + \frac{1}{n_{12}} + \frac{1}{n_{21}} + \frac{1}{n_{22}}}$.

```
# get SE
sqrt((1/730) + (1/100) + (1/130) + (1/40))
```

```
[1] 0.2099099
```

```
# log odds
log(2.246154)
```

```
[1] 0.8092194
```

```
# CI lower
log(2.246154) - (1.96 * 0.2099099)
```

```
[1] 0.397796
```

```
# CI upper
log(2.246154) + (1.96 * 0.2099099)
```

```
[1] 1.220643
```