**Section 2.2.5 – Interactions and transformations for explanatory variables**

Interactions and transformations of explanatory variables can be added to a logistic regression model in the same way as for normal linear regression models.

Interactions

Interactions between explanatory variables are needed when the effect of one explanatory variable on the probability of success depends on the value for a second explanatory variable.

Consider the model of



The effect that x1 has on logit(π) is dependent on the specific value of x2. Thus, we no longer can only look at β1 when trying to understand the effect x1 has on the response. Similarly, the effect that x2 has on logit(π) is dependent on the specific value of x1.

While we can use odds ratios to interpret these effects, it is now a little more difficult. This interpretation still is based on the ratio of two odds. In particular, the odds ratio for x2 holding x1 constant is



Notice the role that x1 plays in OR. Thus, you will always need to include x1 when interpreting x2’s corresponding odds ratio:

The odds of success are  times as large for a c-unit increase in x2 when x1 is fixed at a value of \_\_\_.

Another correct interpretation is

The odds of success change by  times for every c-unit increase in x2 when x1 is fixed at a value of \_\_\_.

Compare this to the odds ratio interpretation for x2 corresponding to the model of . The “when x1 is fixed at a value of \_\_\_” of the previous interpretation would be replaced with “when holding x1 constant”. If there was an additional x3 explanatory variable in the model (not included in the interaction), one would need to include “when holding x3 constant”.

Wald and profile likelihood ratio intervals again can be found for OR. With respect to the Wald interval, we use the same basic form as before, but now with a more complicated variance expression. For example, the interval for the x2 odds ratio in the  model is



where



Of course, the variances and covariances can be found from the estimated covariance matrix. Note that the  expression again makes use of the general result



for random variables Y1 and Y2 and constants a and b.

Profile LR intervals are generally preferred. However, they can be more difficult to calculate due to the additional parameters included in the odds ratio similar to what we saw in previous notes.

Below are a few general comments about these models before we get into an example:

* The individual explanatory variables are often referred to as main effect terms in a model. Thus,  represents a main effect term and  is an interaction term.
* There are a few different ways the  can be coded into the formula argument of glm() when there are two variables called x1 and x2 in a data frame:
* formula = y ~ x1 + x2 + x1:x2
* formula = y ~ x1\*x2
* formula = y ~ (x1 + x2)^2

The colon indicates an interaction term between variables. The asterisk indicates all interactions AND lower order effects (only x1 and x2). The ^2 indicates all two-way interactions and lower order effects.

* I will use the convention of including all lower order interactions and main effects whenever an interaction is included. For example, if a three-way interaction is included like x1x2x3, all two-way interactions and main effects among the explanatory variables would also be included. This is referred to as the principle of marginality.

Example: Placekicking (Placekick.R, Placekick.csv)

Generally, one would conjecture that the more time that a football is in the air (more time equates to a longer distance), the more it is susceptible to the effect of wind.

For example, a 50-yard placekick will have a longer time period that the wind can affect it than a 20-yard placekick.

Thus, a distance and wind interaction would be of interest to examine. The wind explanatory variable in the data set is a binary variable for placekicks attempted in windy conditions (1) vs. non-windy conditions (0) placekicks, where windy conditions are defined as a wind stronger than 15 miles per hour at kickoff in an outdoor stadium.

Below is how a model including distance, wind, and the distance×wind interaction is estimated:

> mod.fit.Ha <- glm(formula = good ~ distance + wind +

 distance:wind, family = binomial(link = logit), data =

 placekick)

> summary(mod.fit.Ha)

Call: glm(formula = good ~ distance + wind + distance:wind, family = binomial(link = logit), data = placekick)

Deviance Residuals:

 Min 1Q Median 3Q Max

-2.7291 0.2465 0.2465 0.3791 1.8647

Coefficients:

 Estimate Std. Error z value Pr(>|z|)

(Intercept) 5.684181 0.335962 16.919 <2e-16 \*\*\*

distance -0.110253 0.008603 -12.816 <2e-16 \*\*\*

wind 2.469975 1.662144 1.486 0.1373

distance:wind -0.083735 0.043301 -1.934 0.0531 .

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1013.43 on 1424 degrees of freedom

Residual deviance: 767.42 on 1421 degrees of freedom

AIC: 775.42

Number of Fisher Scoring iterations: 6

The estimated logistic regression model is



Wald test information for the test of H0: β3 = 0 vs. Ha: β3 ≠ 0 is given in the output. The test statistic is ZW = -1.934, and the p-value is 0.0531. Thus, there is marginal evidence of a distance and wind interaction.

To perform a LRT, we can fit the model under the null hypothesis and then use the anova() function:

> mod.fit.Ho <- glm(formula = good ~ distance + wind, family = binomial(link = logit), data = placekick)

> anova(mod.fit.Ho, mod.fit.Ha, test = "Chisq")

Analysis of Deviance Table

Model 1: good ~ distance + wind

Model 2: good ~ distance + wind + distance:wind

 Resid. Df Resid. Dev Df Deviance P(>|Chi|)

1 1422 772.53

2 1421 767.42 1 5.1097 0.02379 \*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The test statistic is -2log(Λ) = 5.1097, and the p-value is 0.0238. Again, there is marginal evidence of a distance and wind interaction.

Notes:

* The negative coefficient on the distance main effect indicates that the probability of success decreases with increasing distance when wind is 0. The negative coefficient on the interaction term indicates that this effect is exacerbated under windy conditions for the longer placekicks as the distance increases.
* Another way to obtain the LRT information would be to use the Anova() function from the car package:

> Anova(mod.fit.Ha, test = "LR")

Analysis of Deviance Table (Type II tests)

Response: good

 LR Chisq Df Pr(>Chisq)

distance 238.053 1 < 2e-16 \*\*\*

wind 3.212 1 0.07312 .

distance:wind 5.110 1 0.02379 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

The p-value for the interaction is the same as calculated by anova(). One needs to be careful though with interpreting the p-values for distance and wind. The Anova() function uses what is known as a type II test format. This involves testing a particular term in the model, given all of the other terms in the model except for higher order interactions that contain the term. For example, the wind test is for

H0: 

Ha: 

Notice there is no interaction in either model.

* A plot of the estimated logistic regression model is quite useful to see how windy conditions affect the probability of success. Using the curve() function in a similar manner as before, below are plots of the models with and without the interaction:



To begin using odds ratios, we need to find the odds ratio for wind comparing windy (1) vs. non-windy (0) holding distance constant. Below is the expression from our earlier discussion:



For this equation, x1 is distance and x2 is wind. Of course, c = 1 for this setting due to wind being binary. Because distance could be anywhere from 18 to 66 yards, I will choose distances of 20, 30, 40, 50, and 60 yards to interpret the odds ratio for wind. Below is the code:

> beta.hat <- mod.fit.Ha$coefficients[2:4]

> c <- 1

> distance <- seq(from = 20, to = 60, by = 10)

> OR.wind <- exp(c\*(beta.hat[2] + beta.hat[3]\*distance))

> cov.mat <- vcov(mod.fit.Ha)[2:4,2:4]

> #Var(beta^\_2 + distance\*beta^\_3)

> var.log.OR <- cov.mat[2,2] + distance^2\*cov.mat[3,3] +

 2\*distance\*cov.mat[2,3]

> ci.log.OR.low <- c\*(beta.hat[2] + beta.hat[3]\*distance) –

 c\*qnorm(p = 0.975)\*sqrt(var.log.OR)

> ci.log.OR.up <- c\*(beta.hat[2] + beta.hat[3]\*distance) +

 c\*qnorm(p = 0.975)\*sqrt(var.log.OR)

> round(data.frame(distance = distance, OR.hat = 1/OR.wind,

 OR.low = 1/exp(ci.log.OR.up), OR.up =

 1/exp(ci.log.OR.low)),2)

 distance OR.hat OR.low OR.up

1 20 0.45 0.09 2.34

2 30 1.04 0.40 2.71

3 40 2.41 1.14 5.08

4 50 5.57 1.54 20.06

1. 60 12.86 1.67 99.13

Notes:

* I put the estimated ’s into one object so that I could use the same [index] values as would correspond to , , and  in my model (remember that  is the [1] element of mod.fit.Ha$coefficients).
* I inverted the odds ratio so that the estimate and interval for 1/OR is given. I still use column names in the data frame like “OR.hat” because this is still an odds ratio.
* Because the odds ratio was inverted, we now have the odds of success for non-windy conditions divided by the odds of success for windy conditions.
* As originally conjectured, the longer the distance, the more it is susceptible to the wind. Notice that from 40 yards on, the confidence interval does not contain 1. Thus, there is sufficient evidence at those distances to indicate the wind has an effect on the success or failure of the placekick.

Example interpretation: With 95% confidence, the odds of a success are between 1.14 to 5.08 times as large for non-windy (wind = 0) than for windy (wind = 1) placekicks at a distance of 40 yards.

The previous code relied on programming into R the corresponding expressions for the Wald interval. The next set of code shows how this can be simplified by using the emmeans package.

> library(package = emmeans)

> calc.est <- emmeans(object = mod.fit.Ha, specs = ~ distance + wind, at = list(distance = c(20,30,40,50,60)), type = "response")

> summary(calc.est)

 distance wind prob SE df asymp.LCL asymp.UCL

 20 0 0.9701 0.00518 Inf 0.95808 0.979

 30 0 0.9150 0.00907 Inf 0.89549 0.931

 40 0 0.7814 0.01717 Inf 0.74593 0.813

 50 0 0.5428 0.03632 Inf 0.47121 0.613

 60 0 0.2827 0.04421 Inf 0.20452 0.377

 20 1 0.9863 0.01110 Inf 0.93510 0.997

 30 1 0.9117 0.03803 Inf 0.80354 0.963

 40 1 0.5974 0.08827 Inf 0.41954 0.753

 50 1 0.1758 0.09236 Inf 0.05761 0.427

 60 1 0.0297 0.02940 Inf 0.00414 0.184

Confidence level used: 0.95

Intervals are back-transformed from the logit scale

The summary() function is presented here ONLY to show what emmeans() does. It creates a grid of calculations for all possible values as given in the at argument. Normally, one would not view these calculations when the goal is to obtain the odds ratios.

Next, we use a combination of the method functions for the contrast() and confint() functions in the emmeans package to obtain the Wald intervals. Note that the confint() method function in the stats package cannot be used for Wald or profile LR intervals when there is an interaction.

> test.info <- contrast(object = calc.est, method = "pairwise", simple = list("wind"), combine = TRUE)

> confint(object = test.info, adjust = "none", level = 0.95)

 distance contrast odds.ratio SE df asymp.LCL

 20 wind0 / wind1 0.451 0.379 Inf 0.0872

 30 wind0 / wind1 1.043 0.508 Inf 0.4019

 40 wind0 / wind1 2.410 0.917 Inf 1.1430

 50 wind0 / wind1 5.567 3.641 Inf 1.5447

 60 wind0 / wind1 12.860 13.401 Inf 1.6683

 asymp.UCL

 2.34

 2.71

 5.08

 20.06

 99.13

Confidence level used: 0.95

Intervals are back-transformed from the log odds ratio scale

Notes:

* The estimated odds ratios and the Wald intervals match what we had before.
* The odds ratios given by default were for non-windy conditions divided by windy conditions. The function defaulted to this because 0 (non-windy) < 1 (windy). We will examine this behavior more later in the course to understand what would happen if we had non-numerical levels for a variable like wind. Also, If we wanted to invert the odds ratios, we can use reverse = TRUE in contrast().
* If control of the familywise confidence level is desired, this can be specified with the adjust argument. For example, adjust = "bonferroni" uses the Bonferroni adjustment for all distances because combine = TRUE is provided in contrast() to combine all odds ratios into one family.

For the distance odds ratio, we need to hold wind constant at 0 or 1. Also, we need to choose a value for c with respect to distance. The OR equation is



with x1 representing distance and x2 representing wind. Below is my code:

> calc.est.dist <- emmeans(object = mod.fit.Ha, specs = ~ distance + wind, at = list(distance = c(20,30)), type = "response")

> confint(object = contrast(object = calc.est.dist, method = "pairwise", simple = list("distance"), combine = TRUE), adjust = "none", level = 0.95)

 wind contrast odds.ratio SE df

 0 distance20 / distance30 3.01 0.259 Inf

 1 distance20 / distance30 6.96 2.953 Inf

 asymp.LCL asymp.UCL

 2.54 3.56

 3.03 15.98

Confidence level used: 0.95

Intervals are back-transformed from the log odds ratio scale

Why were distances of 20 and 30 yards specified?

We needed a 10-yard decrease in the distance. The same odds ratio will result no matter which 10-yard decrease is chosen (x1 is not in the expression for OR, only c). Thus, we could choose distance = c(20,30)) for the at argument.

The program shows how the corresponding expressions for the Wald interval can be programmed.

Below are the interpretations:

* With 95% confidence, the odds of a success are between 2.54 to 3.56 times as large for a 10-yard decrease in distance under non-windy conditions.
* With 95% confidence, the odds of a success are between 3.03 to 15.98 times as large for a 10-yard decrease in distance under windy conditions.

Equivalently, we can say

* With 95% confidence, the odds of a success change by an amount between 2.54 to 3.56 times for every 10-yard decrease in distance under non-windy conditions.
* With 95% confidence, the odds of a success change by an amount between 3.03 to 15.98 times for every 10-yard decrease in distance under windy conditions.

I think football fans would be more interested in the odds ratios when we conditioned on the distance value.

Next is the code for the profile LR intervals corresponding to the wind odds ratio.

> library(package = mcprofile)

> K.names <- list(c("Wind, distance = 20", "Wind, distance = 30", "Wind, distance = 40", "Wind, distance = 50", "Wind, distance = 60"), c("beta0", "beta1", "beta2", "beta3"))

> K <- matrix(data = c(0, 0, 1, 20,

 0, 0, 1, 30,

 0, 0, 1, 40,

 0, 0, 1, 50,

 0, 0, 1, 60),

 nrow = 5, ncol = 4, byrow = TRUE, dimnames = K.names)

> K

 beta0 beta1 beta2 beta3

Wind, distance = 20 0 0 1 20

Wind, distance = 30 0 0 1 30

Wind, distance = 40 0 0 1 40

Wind, distance = 50 0 0 1 50

Wind, distance = 60 0 0 1 60

> # A little quicker way to form K

> # distance <- seq(from = 20, to = 60, by = 10)

> # K <- cbind(0, 0, 1, distance)

Why are using this value for K? It is because the odds ratio is

 

The estimates and intervals for  are computed next using the method function for confint() in the mcprofile package. The exp() function is applied for the odds ratio calculation.

> linear.combo <- mcprofile(object = mod.fit.Ha, CM = K)

> ci.log.OR <- confint(object = linear.combo, level = 0.95,

 adjust = "none")

> ci.log.OR

 mcprofile - Confidence Intervals

level: 0.95

adjustment: none

 Estimate lower upper

Wind, distance = 20 0.7953 -0.605 2.776

Wind, distance = 30 -0.0421 -0.882 1.075

Wind, distance = 40 -0.8794 -1.638 -0.133

Wind, distance = 50 -1.7168 -3.126 -0.521

Wind, distance = 60 -2.5541 -4.874 -0.700

> exp(ci.log.OR$confint)

 lower upper

1 0.546269219 16.0612507

2 0.413986292 2.9302033

3 0.194295117 0.8757076

4 0.043873290 0.5936966

5 0.007642597 0.4967926

> exp(ci.log.OR)

 mcprofile - Confidence Intervals

level: 0.95

adjustment: none

 Estimate lower upper

Wind, distance = 20 2.2150 0.54627 16.061

Wind, distance = 30 0.9588 0.41399 2.930

Wind, distance = 40 0.4150 0.19430 0.876

Wind, distance = 50 0.1796 0.04387 0.594

Wind, distance = 60 0.0778 0.00764 0.497

> round(data.frame(distance, OR.hat = 1/exp(ci.log.OR$estimate), OR.low = 1/exp(ci.log.OR$confint$upper), OR.up = 1/exp(ci.log.OR$confint$lower)), digits = 2)

 Estimate OR.low OR.up

Wind, distance = 20 0.45 0.06 1.83

Wind, distance = 30 1.04 0.34 2.42

Wind, distance = 40 2.41 1.14 5.15

Wind, distance = 50 5.57 1.68 22.79

Wind, distance = 60 12.86 2.01 130.85

Compare these results with those for the Wald intervals. Also, note that one could use CM = -K in mcprofile() to get the inverted confidence intervals too.

With the matrix formulation given in K, you can use the wald() function in the mcprofile package to find multiple Wald intervals as well.

> save.wald <- wald(object = linear.combo)

> ci.log.OR.wald <- confint(object = save.wald, level = 0.95, adjust = "none")

> round(data.frame(distance, OR.hat = 1/exp(ci.log.OR.wald$estimate), OR.low = 1/exp(ci.log.OR.wald$confint$upper), OR.up = 1/exp(ci.log.OR.wald$confint$lower)), digits = 2)

 distance Estimate OR.low OR.up

Wind, distance = 20 20 0.45 0.09 2.34

Wind, distance = 30 30 1.04 0.40 2.71

Wind, distance = 40 40 2.41 1.14 5.08

Wind, distance = 50 50 5.57 1.54 20.06

Wind, distance = 60 60 12.86 1.67 99.13

My program contains the profile LR interval code for an odds ratio with a 10-yard decrease in distance holding wind constant at 0 or 1.