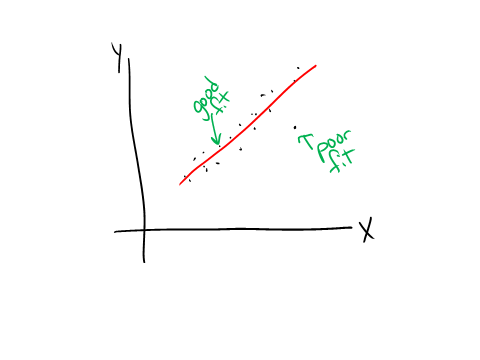
**Section 2.2.4 – Probability of success (continued)**

Plots of the model

Let’s look at a visual assessment of the logistic regression model when there is only one explanatory variable x. For a normal linear regression model, we would examine something like



The closeness of the points to the estimated regression model line tell us something about how well an observation is fit by the model.

Because the response variable in logistic regression is binary, constructing a plot like this is not very informative because all plotted points would be at y = 0 or 1 on the y-axis. However, we can plot the observed proportion of successes at each x instead to obtain a general understanding of how well the model fits the data.

This only works well when the number of observations at each possible x is not small. In the case of a truly continuous x, the number of observations would be 1 and this plot generally would not be very useful. Alternatives for this situation include grouping observations by x and finding the observed proportion of successes for each group; however this leads to potentially different results depending on how the grouping is done

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

Earlier we found the number of observations and number of successes at each unique distance:

> w <- aggregate(x = good ~ distance, data = placekick, FUN = sum)

> n <- aggregate(x = good ~ distance, data = placekick,

FUN = length)

> w.n <- data.frame(distance = w$distance, success = w$good, trials = n$good, proportion = round(w$good/n$good,4))

> head(w.n)

distance success trials proportion

1 18 2 3 0.6667

2 19 7 7 1.0000

3 20 776 789 0.9835

4 21 19 20 0.9500

5 22 12 14 0.8571

6 23 26 27 0.9630

This was used to estimate a logistic regression model using a binomial response form of the data. Instead, we can plot the observed proportion of successes at each distance and overlay the estimated logistic regression model:

> dev.new(width = 7, height = 6, pointsize = 12)

> plot(x = w$distance, y = w$good/n$good, xlab = "Distance

(yards)", ylab = "Estimated probability", panel.first =

grid())

> curve(expr = predict(object = mod.fit, newdata =

data.frame(distance = x), type = "response"), col =

"red", add = TRUE, xlim = c(18, 66))



Previously, I used the curve function with

> curve(expr = exp(mod.fit$coefficients[1] +

mod.fit$coefficients[2]\*x) / (1 +

exp(mod.fit$coefficients[1] +

mod.fit$coefficients[2]\*x)), col = "red", xlim = c(18,

66), ylab = expression(hat(pi)), xlab = "Distance",

main = "Estimated probability of success for a

placekick", panel.first = grid())

to plot the model. Now, I use a simpler way with the predict() function.

Question: If we were to assess the fit of the logistic regression model here in the same manner as for a normal linear regression model, what would you conclude?

To include a measure of how many observations are at each distance, we can use a bubble plot. For this plot, we are going to make the plotting point size proportional to the observed number of observations at each unique distance.

> dev.new(width = 7, height = 6, pointsize = 12)

> symbols(x = w$distance, y = w$good/n$good, circles =

sqrt(n$good), inches = 0.5, xlab = "Distance (yards)",

ylab = "Estimated probability", panel.first = grid())

> curve(expr = predict(object = mod.fit, newdata =

data.frame(distance = x), type = "response"), col =

"red", add = TRUE, xlim = c(18, 66))



The circles argument specifies how to scale the plotting points. I used the sqrt() function within this argument value only to lessen the absolute differences between the values in n$good, and this does not need to be used in general.

There are 789 observations at a distance of 20 yards. The next largest number of observations is 30, which is at a distance of 32 yards. Due to the large absolute difference in the number of observations, this causes the plotting point at 20 yards to be very large and the other plotting points to be very small.

Points that may have caused us concern before, now generally do not because we see they represent a small number of observations. For example,

* 18-yard placekicks: Only 3 observations occurred and there were 2 successes.
* The largest in distance placekicks: Many of these correspond to 1 observation only.

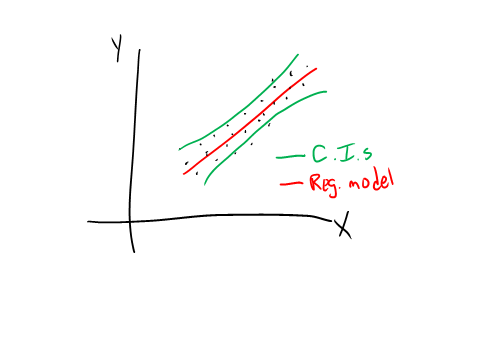
However, there are a few large plotting points, such as at 32 yards ( = 0.89, observed proportion = 23/30 = 0.77) and 51 yards ( = 0.49, observed proportion = 11/15 = 0.73), that may not be fit well by the model. How to more formally assess these observations and others will be an important subject of Chapter 5 when we examine model diagnostic measures.

Question: Suppose the previous plot looked like this:



What do you think about the fit of the model?

Another commonly made plot in normal linear regression with one explanatory variable is a plot of the estimated regression model with confidence interval bands:



We can add these bands to the previous plots for the estimated logistic regression models.

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

I wrote a function called ci.pi() to help automate some of the needed calculations for a Wald interval:

> ci.pi <- function(newdata, mod.fit.obj, alpha){

linear.pred <- predict(object = mod.fit.obj, newdata =

newdata, type = "link", se = TRUE)

CI.lin.pred.lower < -linear.pred$fit - qnorm(p = 1-

alpha/2)\*linear.pred$se

CI.lin.pred.upper <- linear.pred$fit + qnorm(p = 1-

alpha/2)\*linear.pred$se

CI.pi.lower <- exp(CI.lin.pred.lower) / (1 +

exp(CI.lin.pred.lower))

CI.pi.upper <- exp(CI.lin.pred.upper) / (1 +

exp(CI.lin.pred.upper))

list(lower = CI.pi.lower, upper = CI.pi.upper)

}

> #Test case

> ci.pi(newdata = data.frame(distance = 20), mod.fit.obj =

mod.fit, alpha = 0.05)

$lower

1

0.9597647

$upper

1

0.9791871

**You may find this function quite useful for your own calculations whenever you need a confidence interval for π!**

To add the confidence interval bands to the previous plot, we can use the following code:

> curve(expr = ci.pi(newdata = data.frame(distance = x),

mod.fit.obj = mod.fit, alpha = 0.05)$lower, col =

"blue", lty = "dotdash", add = TRUE, xlim = c(18, 66))

> curve(expr = ci.pi(newdata = data.frame(distance = x),

mod.fit.obj = mod.fit, alpha = 0.05)$upper, col =

"blue", lty = "dotdash", add = TRUE, xlim = c(18, 66))

> legend(x = 20, y = 0.4, legend = c("Logistic regression

model", "95% individual C.I."), lty = c("solid",

"dotdash"), col = c("red", "blue"), bty = "n")



Therefore, we have the 95% confidence interval at each possible distance. Note that the familywise error rate though is not controlled.

Code for confidence interval bands for a profile LR interval is included in the program.