intro to logistic regression

2025-05-24

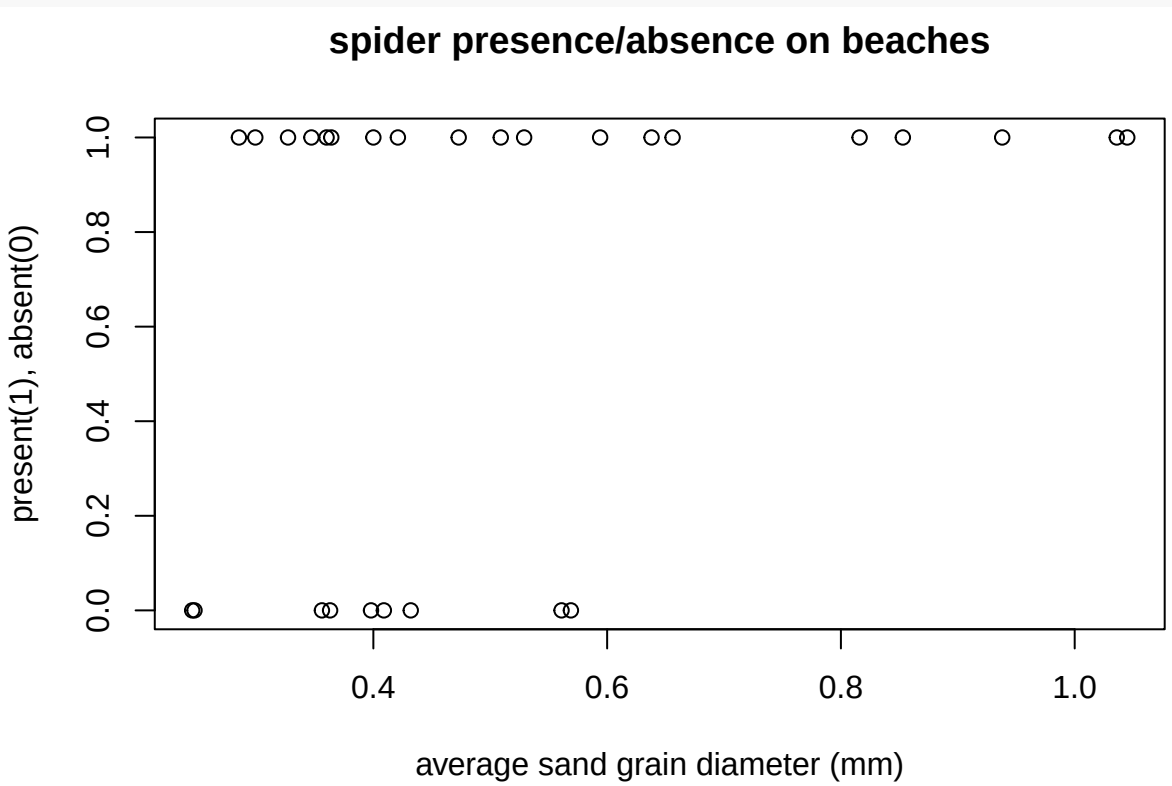
As an example of simple logistic regression, Suzuki et al. (2006) measured sand grain size on 28 beaches in Japan and observed the presence or absence of the burrowing wolf spider Lycosa ishikariana on each beach. Sand grain size is the predictor variable, and spider presence or absence is the dependent variable.

One goal of this study would be to determine whether there was a relationship between sand grain size and the presence or absence of the species, in hopes of understanding more about the biology of the spiders.

Because this species is endangered, another goal was to find an equation to predict the probability of a wolf spider population surviving on a beach with a particular sand grain size, to help determine which beaches to reintroduce the spider to.

A statistical model needs a numerical outcome variable. In logistic regression the outcome variable is 0 or 1, with 1 representing success (in this case present), and 0 representing failure (absence).
We let y be the indicator of success, and x be average grain diameter on the beach. Here is a plot of y vs x.

```
data=read.csv("~/D/spider.csv")
attach(data)
y=ifelse(data$spiders=="present",1,0)
x=data[[1]]
plot(x,y,ylab="present(1), absent(0)",xlab="average sand grain diameter (mm)",main="spider presence/abs
```



I makes no sense to fit a regression model $y = \beta_0 + \beta_1 x + \epsilon$ because y is 0 or 1, so far from normally distributed.
In logistic regression we let $p(x)$ be the probability of success when the predictor variable takes value x, and

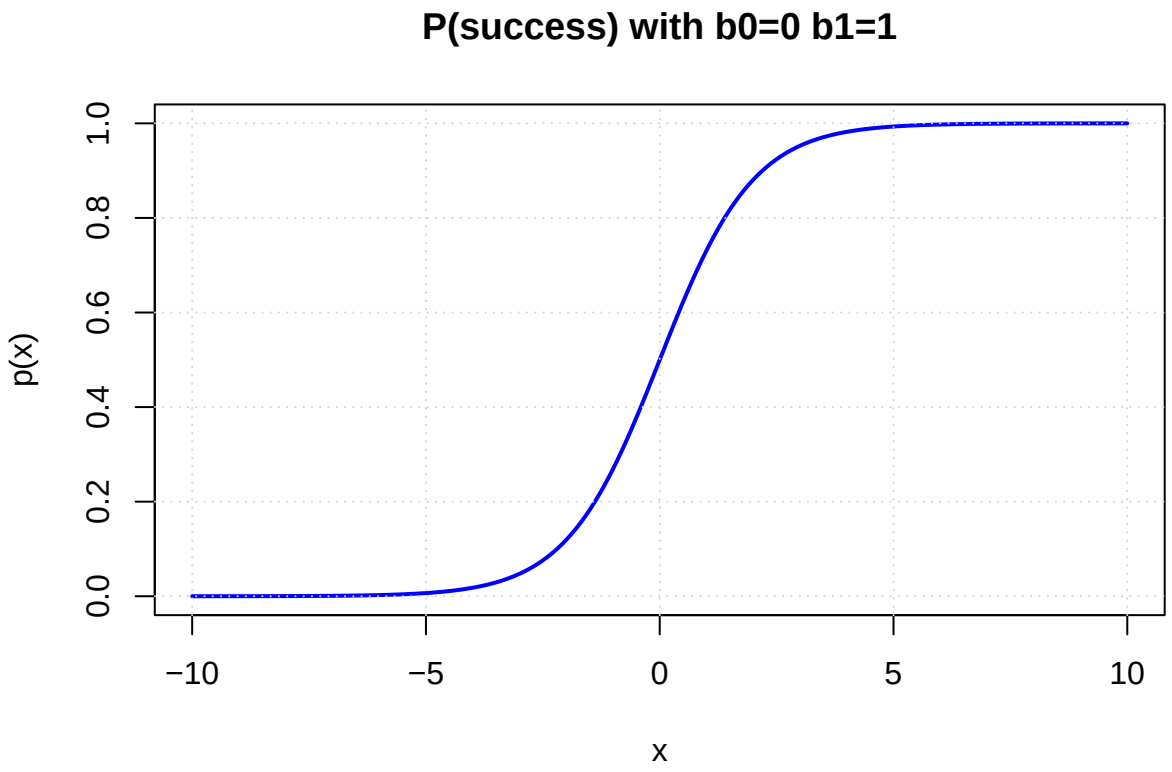
specifically

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

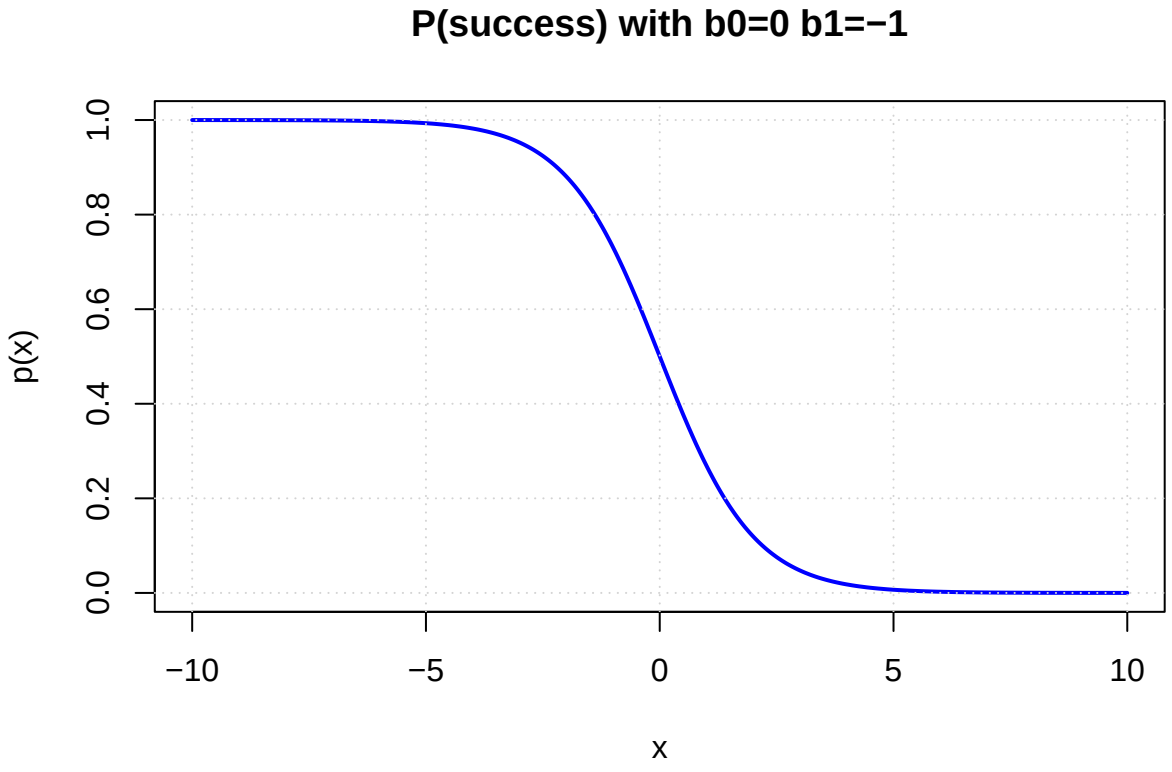
$p(x)$ takes values between 0 and 1. The models says that the outcome variable y will be 1 with probability $p(x)$ and 0 with probability $1-p(x)$, and these probabilities depend on the parameters.

Let's see what it looks like for a few choices for β_0 and β_1, and with x taking values between -10 and 10 for convenience.

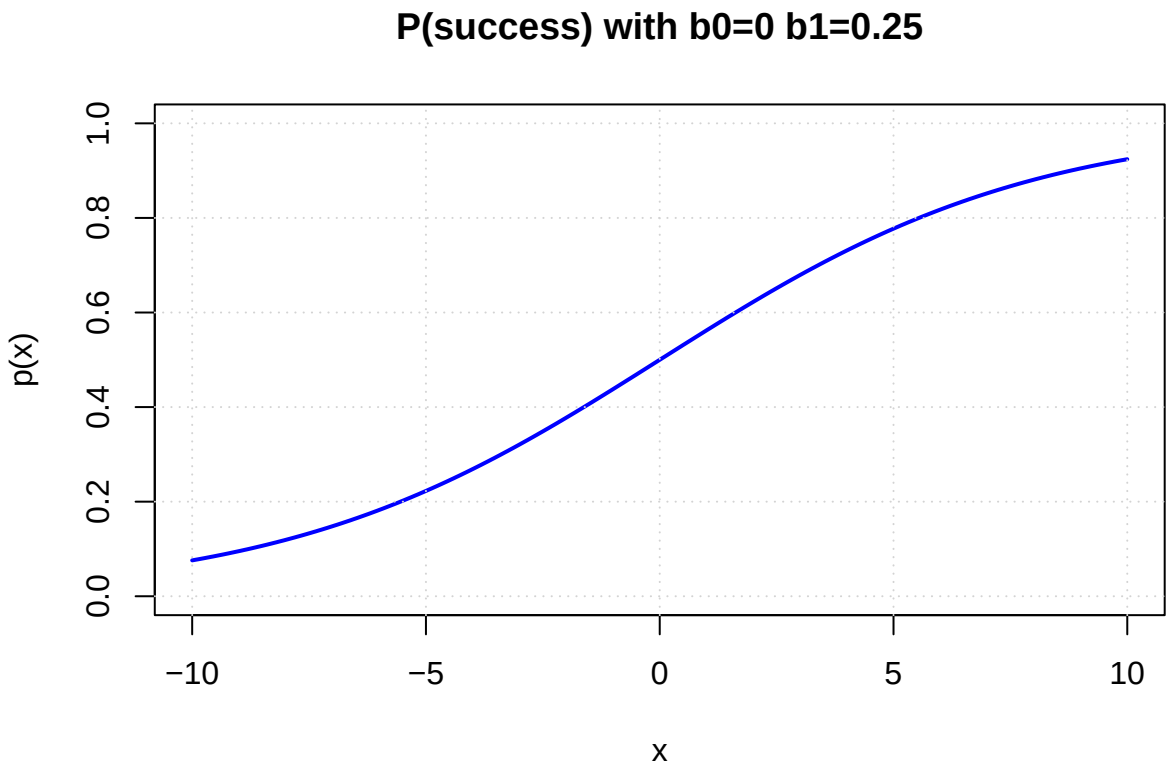
```
plot_logistic <- function(x_range = c(-10, 10), b0=0, b1=1,
                          title = paste("P(success) with"," b0=",as.character(b0
),
              " b1=",as.character(b1), sep=""),
                          xlab = "x", ylab = "p(x)",
                          col = "blue", lwd = 2) {
  # Logistic function definition
  p <- function(x) {
    exp(b0+b1*x)/(1+exp(b0+b1*x))
  }

  # Generate x values
  x <- seq(x_range[1], x_range[2], length.out = 500)

  # Compute logistic function values
  y <- p(x)

  # Plot
  plot(x, y, type = "l", col = col, lwd = lwd,ylim=c(0,1),
       main = title, xlab = xlab, ylab = ylab)

  # Add grid
  grid()
}

plot_logistic(b0=0,b1=1)
```



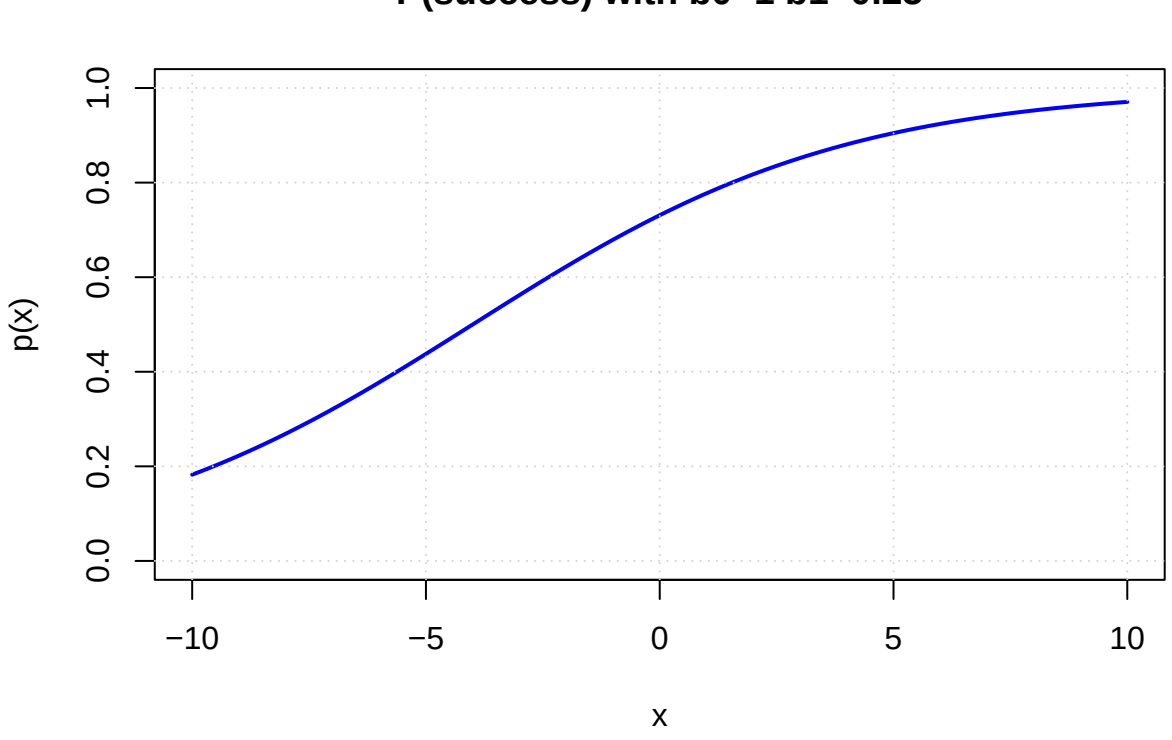

```
plot_logistic(b0=0,b1=-1)
```




```
plot_logistic(b0=0,b1=.25)
```




```
plot_logistic(b0=1,b1=.25)
```



To fit the logistic regression model in R, we use the glm function, specifying a binomial distribution (the y's have binomial distributions with $n = 1$, and with p depending on x and the parameters β_0 and β_1).

```r
glm.out=glm(y~x,family=binomial)
summary(glm.out)
```

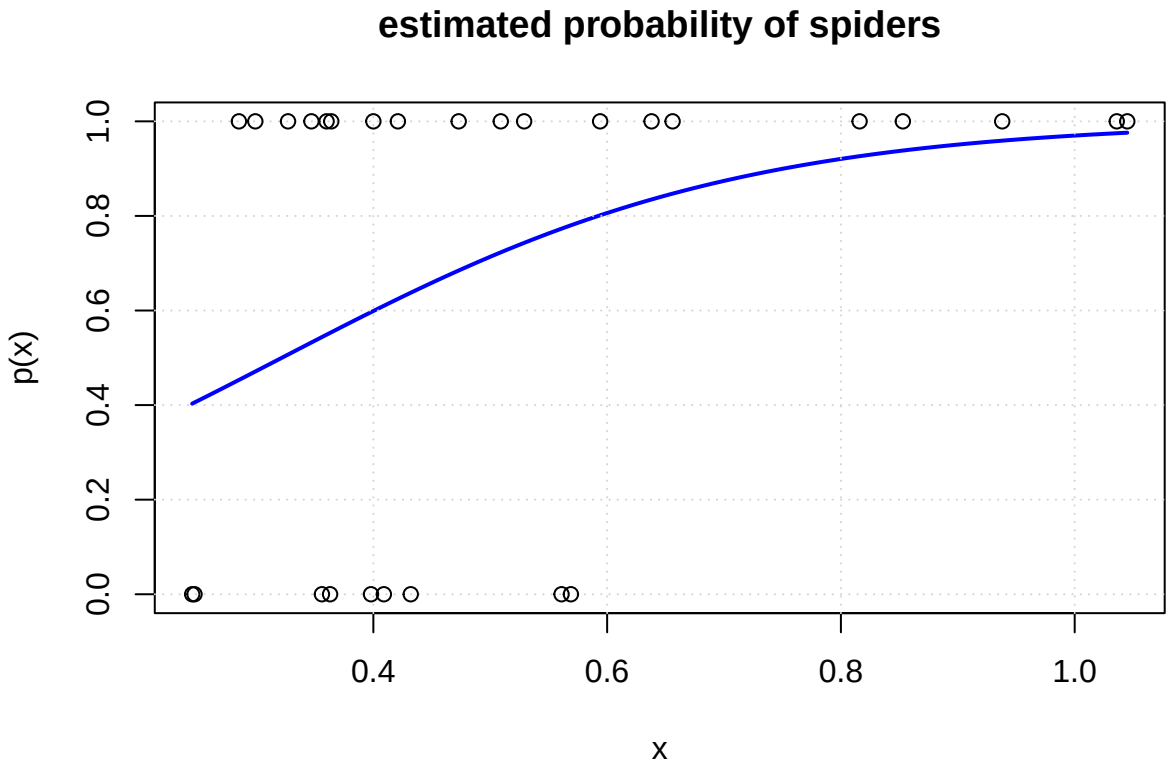
```
##
## Call:
## glm(formula = y ~ x, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7406  -1.0781   0.4837   0.9809   1.2582
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.648      1.354  -1.217   0.2237
## x              5.122      3.006   1.704   0.0884 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 35.165  on 27  degrees of freedom
## Residual deviance: 30.632  on 26  degrees of freedom
## AIC: 34.632
##
## Number of Fisher Scoring iterations: 5
```

```r
anova(glm.out)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev
## NULL                    27     35.165
## x     1   4.5332        26     30.632
```

What can we say about the predicted probability of wolf spiders as a function of average sand grain size? That is summarized in the following plot.

```r
plot_logistic(b0=coef(glm.out)[1],b1=coef(glm.out)[2],
   x_range=c(min(x),max(x)),title="estimated probability of spiders")
points(x,y)
```



This makes sense. The estimated probability is high for those values of x where all beaches had spiders, and moderate for values of x where some beaches had spiders, and some didn't. The important thing is that we can use the estimated function $p(x)$ to predict the probability of spiders based on average grain size, which was goal two of the original study.

As with linear regression, we could put confidence intervals on the estimated model parameters and the estimate of $p(x)$. Details of statistical inference for generalized linear models (glm's), of which logistic regression is a particular case, are discussed at length in Stat4620.

You should note is that the fitting function called in R has changed (from lm for linear models to glm for generalized linear models), but the summary and anova outputs look rougly comparable. In the example x (grain diameter) looks marginally significant (p-value about .09). It's important to know that for generalized linear models the validity of p-values, confidence intervals, etc, requires moderately large sample sizes, while in this example sample size is fairly small.
Also, there is no restriction to a single predictor variable, and $\beta_0 + \beta_1 x$ can be extended to $\beta' \mathbf{x}$ as in linear regression.

Reference:

Suzuki, S., N. Tsurusaki, and Y. Kodama. 2006. Distribution of an endangered burrowing spider Lycosa ishikariana in the San'in Coast of Honshu, Japan (Araneae: Lycosidae). Acta Arachnologica 55: 79-86.