Lecture 10: Qualitative and Limited Dependent Variable Models

Thomas Chadefaux

Contents

1 Introduction

We have already encountered the case of binary variables in the context of independent variables. We called these dummy variables, and they could be for example gender (male/female), whether someone went to college or not, etc.

So far we have considered the case of continuous dependent variables. Income, % of votes, etc., are continuous variables. But often your *dependent* variable will be binary or at least limited to a small number of options. For example, whether:

- A country enters (or wins) a war or not
- A politician wins an election
- A citizen votes
- A bill is adopted
- Someone gets a BA/MA/PhD

2 Why Not Use the Linear Probability Model?

Suppose y_i denotes whether a country goes to war or not. I.e., $y_i = 0$ if it does not enter the war, and $y_i = 1$ if it does. Suppose that we are interested in the probability that a country goes to war, i.e., $P(y_i = 1)$. We think the level of trade dependence of that country might explain the decision to go to war. So we estimate the following model:

$$y_i = \beta_0 + \beta_1 trade_i + \varepsilon_i$$

```
n <- 100
x <- rnorm(n, mean = 0)
y <- round(1/(1 + exp(-(x + rnorm(n, sd = 0.5)))))

# --- First use the linear model
lm1 <- lm(y ~ x)
#
```

```
plot(x, y, lty = 2)
abline(lm1)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.5761	0.0322	17.87	0.0000
x	0.3719	0.0309	12.05	0.0000

But the problem is threefold here:

- First, we get nonsensical predictions. For example, it makes no sense to predict a probability of war > 1.

```
predict(lm1, newdata= data.frame(x = 2))

       1
1.319951
```

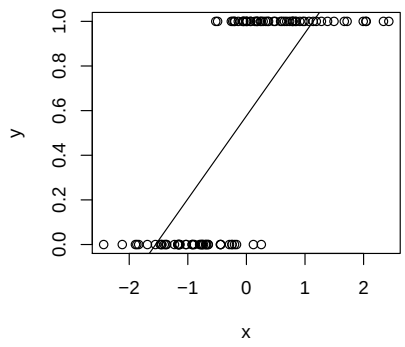


Figure 1: Nonsensical predictions using OLS for a binary response

- Second, the distribution of our error terms is clearly heteroskedastic and not normal (fig. 2).
- Finally, R^2 is highly questionable.

The second problem is addressed by using maximum likelihood estimation (more on this later). The first problem is addressed by choosing a different link from the Xs to Y. In the OLS case, the link was linear. But what if, instead, we chose a sigmoid function?

I.e., let Z be a linear function of the Xs. For example,

$$Z = \beta_0 + \beta_1, X$$

and then we can define $p(Y_i = 1)$ as a sigmoid function of Z.

There are two popular versions of this curve: the logistic function, used for the logit, and the cumulative normal distribution, used in probit estimation. Neither has any particular advantage, and it does not really matter which you use. Here I will only cover the logit, but the probit is pretty much the same.

```
yhat <- predict(lm1)
plot(yhat, resid(lm1))
```

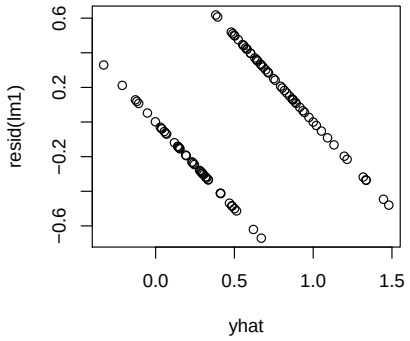


Figure 2: Heteroskedastic distribution of error term (not even mentioning non-normality.)

3 Logit

In logit estimation the probability of the occurrence of the event is determined by the function

$$p(Y_i = 1|X_i) = f(Z) = \frac{1}{1+e^{-Z}} = \frac{1}{1+e^{-(X\beta)}} \tag{1}$$

Note that this is simply choosing a different link from X to p. In the OLS case, the link was linear, i.e., $p(Y_i = 1|X_i) = f(Z) = Z = X\beta$. With the logit (and similarly with the probit), the link is a sigmoid function, such that as Z increases (to infinity), e^{-Z} tends to 0, and hence $p(Y_i = 1|X_i)$ tends to 1. Similarly, as Z decreases (to minus infinity), e^{-Z} tends to infinity, and hence $p(Y_i = 1|X_i)$ tends to 0. As a result, our predictions are bounded by 0 and 1.[1]

Clearly, we cannot estimate this directly by OLS, as $p(Y_i = 1|X_i)$ is nonlinear not only in X, but also in the βs.

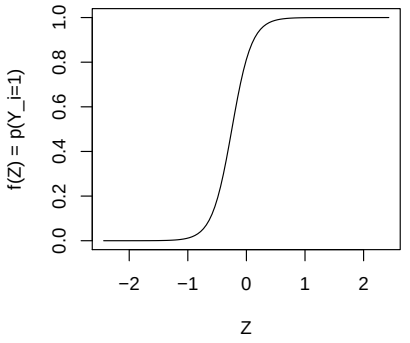


Figure 3: Example of a sigmoid function: the logistic function

[1] The probit model assumes that the transformation function F is the cumulative density function (cdf) of the standard normal distribution. The response probabilities are then

$$\begin{aligned} p(Y_i = 1|X_i) &= \Phi(X\beta) \\ &= \int_{-\infty}^{X\beta} \phi(t)dt \\ &= \int_{-\infty}^{X\beta} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{7}2t^2} dt, \end{aligned}$$

where $\phi(\cdot)$ is the pdf and $\Phi(\cdot)$ is the cdf of the standard normal distribution.

4 Estimation Using R

```
glm.logit <- glm(y ~ x, family = binomial(link = 'logit'))
plot(x, y, type='p', xlab='Z', ylab='f(Z) = p(Y_i=1)')
curve(predict(glm.logit ,data.frame(x=x),type="resp"), add=TRUE)
```

5 Interpretation of Logit Coefficients

VERY IMPORTANT: you can no longer interpret β_k as the marginal effect of variable x_k on y! To interpret the effect of a change of x_k by one unit, we need to

calculate the derivative of $p(Y_i = 1|x_i)$ with respect to x_k:

$$\frac{\partial p(Y_i = 1|x_i)}{\partial x_{ik}} = \frac{e^{x_i\beta}}{(1 + e^{x_i\beta})^2}\beta_k$$

Note that the marginal effect of β_k depends on all the x_is. I.e., you can no longer say "an increase in x leads to an increase in y by such amount. Rather, the effect depends on the values of the other covariates. Look back at the figure of the logit curve, and note that the effect is much strong around $x = 0$ than for $x = -10$ or $x = 10$.

To see this, consider first the case in which you only have one IV. Suppose you obtain the following results:

```
glm.logit <- glm(y ~ x, family = binomial(link = 'logit'))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.4675	0.5144	2.85	0.0043
x	5.9121	1.3918	4.25	0.0000

Now let's calculate the effect of an increase of x by 1. We will do this by calculating $E[y|x = a]$ - $E[y|x = b]$, where the expected value is given in (1). I.e.,

$$p(Y_i = 1|X_i = 0) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \times 0)}}$$

```
z0 <- coef(glm.logit)%*%c(1,0)
z1 <- coef(glm.logit)%*%c(1,1)
z2 <- coef(glm.logit)%*%c(1,2)
Eyx0 <- 1/(1+exp(-z0)); Eyx0

          [,1]
[1,] 0.8126723

Eyx1 <- 1/(1+exp(-z1)); Eyx1

          [,1]
[1,] 0.9993765

Eyx2 <- 1/(1+exp(-z2)); Eyx2

          [,1]
[1,] 0.9999983
```

Notice how the effect of an increase in x is dramatic when x goes from 0 to 1, but much smaller when it goes from 1 to 2. Again, you CANNOT interpret β alone. β only makes sense for a given value of all the xs.

Interpretation of logit in terms of odds ratio We can transform (1) in a way in which the relationship becomes linear. First, rewrite eqn 1 as:

$$p(Y_i = 1|X_i) = f(Z) = \frac{1}{1 + e^{-Z}} = \frac{1}{1 + \frac{1}{e^Z}}$$
$$= \frac{1}{e^Z + \frac{1}{e^Z}} = \frac{e^Z}{1 + e^Z}$$

Now consider the *odds ratio* of the outcome being one. The odds ratio are simply the probability that $y = 1$ divided by the probability that $y = 0$. So the odds ratio are:

$$\frac{p(Y_i = 1|X_i)}{1 - p(Y_i = 1|X_i)} = \frac{\frac{e^Z}{1+e^Z}}{1 - \frac{e^Z}{1+e^Z}} = \frac{\frac{e^Z}{1+e^Z}}{\frac{1}{1+e^Z}} = e^Z$$

Now we simply take the log of (2):

$$L_i = ln(\frac{p(Y_i = 1|X_i)}{1 - p(Y_i = 1|X_i)}) = Z_i = \beta_0 + \beta_1 X$$

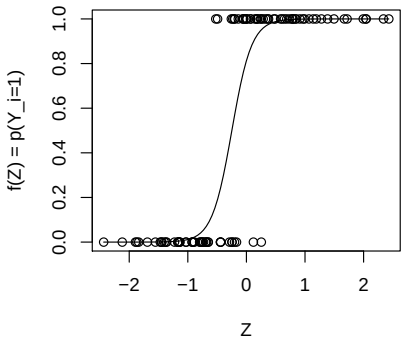


Figure 4: Estimation of a logit model with R's glm function.

Another way to interpret β: interpret it: β_1 measures the change in L for a unit change in X, that is, it tells us how the log-odds of $y = 1$ change as X increases by one unit. The intercept β_0 is the value of the log-odds of $y = 1$ if $X = 0$.

```
glm.logit <- glm(y ~ x, family = binomial(link = 'logit'))
beta <- coef(glm.logit)
newx <- 0
ez <- beta[1] + beta[2]*newx
exp(ez)/(1+exp(ez))
```

```
(Intercept)
  0.8126723
```

```
# note that this is equivalent to R's canned function:
predict(glm.logit, newdata=data.frame(x = newx), type='response')
```

```
        1
0.8126723
```

```
newx = 1
predict(glm.logit, newdata=data.frame(x = newx), type='response')
```

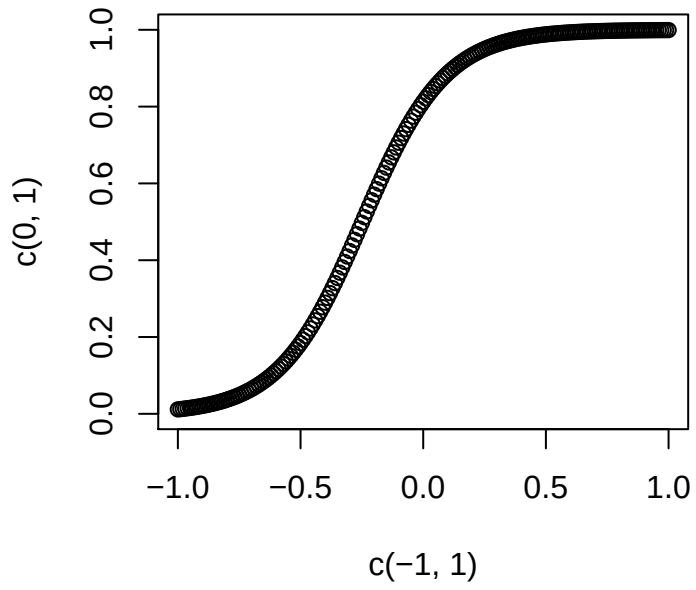
```
        1
0.9993765
```

```
newx = 2
predict(glm.logit, newdata=data.frame(x = newx), type='response')
```

```
        1
0.9999983
```

We can plot the effect of X on Y:

```
plot(c(-1,1), c(0,1), type='n')
for(i in seq(-1,1, 0.01)){
points(i, predict(glm.logit, newdata=data.frame(x = i), type='response'), xlab='x', ylab=' predicted probabiliti
}
```



The best way to present your results is to plot or report predicted probabilities. However, you always need to be aware that you need to hold all other variables at a given level. The level you choose will affect the effect!! Typically what is done is to hold variables at the median, vary x, and report predicted probability. Or hold variables at the median and set x to its 25 percentile, 50 percentiles, and 75 percentile.

6 Latent Variable Interpretation

There is another way to understand the logit. The idea is that while we are observing y as 0 or 1, there is an underlying variable that is continuous. For example, we may observe only whether a voter votes for candidate A or B. But maybe one citizen has a very strong utility for candidate A, whether another has only a slightly higher utility for A than for B. Both will vote for A, and that is all we observe, but their underlying y is different. We call this underlying variable a "latent" variable and denote it $y*$. In short,

$$y_i = \begin{cases} 1 & if\ y_i* > 0 \\ 0 & if\ y_i* \leq 0 \end{cases} \tag{2}$$

Furthermore, assume that the individual observations are iid, that the xs are exogenous, and that the error term is normally distributed and homoskedastic:

$$\varepsilon | x_i \sim N(0, \sigma^2)$$

Then the probability that individual i chooses $y_i = 1$ can be derived from the latent variable:

$$P(Y_i = 1|x_i) = P(y*_i > 0|x_i) = P(x'\beta + \varepsilon_i > 0|x_i) = P(\varepsilon_i > -x'\beta|x_i) = \Phi(x\beta) \tag{3}$$

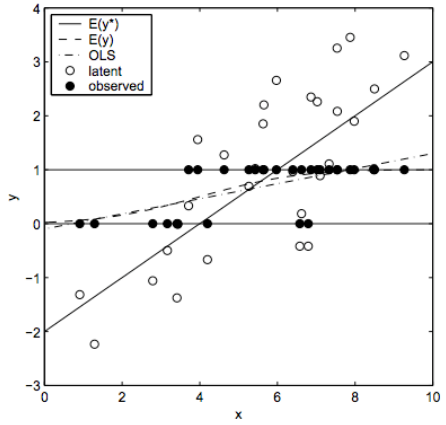


Figure 5: Latent variable model

7 Estimation: An Introduction to Maximum Likelihood Estimation (MLE)

7.1 Finding the mean

Suppose that we observe one draw from a normal distribution X with standard deviation σ, but for which we do not know the mean μ. The one observation we do have is $x_1 = 4$. Remember that the probability density function of the normal distribution is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{X-\mu}{\sigma})^2}$$

In our case we do not know μ, but we know everything else in the equation. So we can try different values of μ, and given that value of μ, see how probable it would be to observe 4. Let's start by assuming that $\mu = 0$:

$$f(4|\mu = 0, \sigma = 1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{4-0}{\sigma})^2}$$

```
(1 / (1*sqrt(2*pi)))*exp(-(0.5*((4-0)/1)^2))

[1] 0.0001338302
```

I.e., the probability to observe 4 given that $\mu = 0$ is very very low. What this tells us is that our observation, 4, is unlikely to have come from a standard normal distribution with mean 0. What about a mean of 3.5?

$$f(4|\mu = 3.5, \sigma = 1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{4-3.5}{\sigma})^2}$$

```
(1 / (1*sqrt(2*pi)))*exp(-(0.5*((4-3.5)/1)^2))

[1] 0.3520653
```

Now the probability to observe 4 would be 0.35. I.e., a much more probable outcome, and so the likelihood that 3.5 is the true population mean is much higher. In fact, we can repeat this process for all possible values of μ and find that (surprise surprise!) $\mu = 4$ was the most likely value for μ, given that we observed $x_1 = 4$ (fig. 6)

```
options(continue=" ")
df <- NULL
for(mu in seq(-5, 10, 0.1)){ #0.1
    p <- (1 / (1*sqrt(2*pi)))*
      exp(-(0.5*((4-mu)/1)^2))
    df.tmp <- data.frame(mu = mu, p = p)
    df <- rbind(df, df.tmp)
}

plot(df, type='l')
library(fields)
xline(df$mu[which(df$p == max(df$p))],
      col=2, lty=2)
```

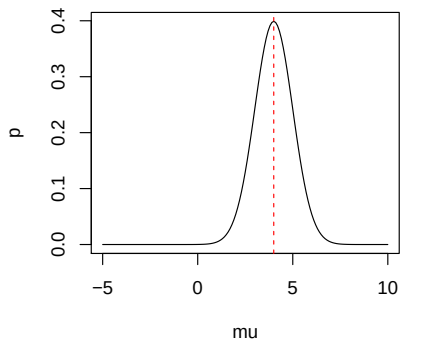


Figure 6: We calculate $p(4|\mu)$ for all possible values of μ and find that $\mu = 4$ was the most likely value for μ, given that we observed $x_1 = 4$

What if we had two observations, say $x_1 = 1$ and $x_2 = 37$. What would then be the most likely value for μ? Again, let us start with a random value of μ, say 13. What is the probability to observe 4 and 37 given that $\mu = 13$? It is simply the product of the two probabilities, i.e.:

$$f(4|\mu = 13, \sigma = 1) \times f(37|\mu = 13, \sigma = 1) = (\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{4-13}{\sigma})^2}) \times (\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{37-13}{\sigma})^2})$$

```
(1 / (1*sqrt(2*pi)))*exp(-(0.5*((4-13)/1)^2)) *
(1 / (1*sqrt(2*pi)))*exp(-(0.5*((37-13)/1)^2))

[1] 3.436235e-144
```

That's a very low probability. What if $\mu = 20$? Then

```
(1 / (1*sqrt(2*pi)))*exp(-(0.5*((4-20)/1)^2)) *
  (1 / (1*sqrt(2*pi)))*exp(-(0.5*((37-20)/1)^2))

[1] 7.187433e-120
```

That's still low, but higher. Again, let's calculate the joint probability for various values of μ. Not surprisingly, we find that the most likely value for μ is 21 (fig. 7.1).

7.2 Finding the mean AND the sd

Suppose now that we want to find two parameters, for example we also want to find the most likely value of σ. We proceed essentially in the same way, except that this time we are evaluating pairs of parameters:

```
df <- NULL
for(mu in seq(15, 25, 1)){ #1
  for(sd in seq(0, 50, 1)){#1
    p <- (1 / (sd*sqrt(2*pi)))*exp(-(0.5*((4-mu)/sd)^2)) *
      (1 / (sd*sqrt(2*pi)))*exp(-(0.5*((37-mu)/sd)^2))
    df.tmp <- data.frame(mu = mu, sd = sd, p = p)
    df <- rbind(df, df.tmp)
}
}
library(lattice)
wireframe(df$p ~ df$sd * df$mu, scales=list(arrows=F))
```

```
df <- NULL
for(mu in seq(15, 25, 0.1)){ #0.1
    p <- (1 / (1*sqrt(2*pi)))*exp(-(0.5*((4-mu)/
      (1 / (1*sqrt(2*pi)))*exp(-(0.5*((37-mu)/1)
    df.tmp <- data.frame(mu = mu, p = p)
    df <- rbind(df, df.tmp)
}
plot(df, type='l')
library(fields)
xline(df$mu[which(df$p == max(df$p))], col=2, lt
```

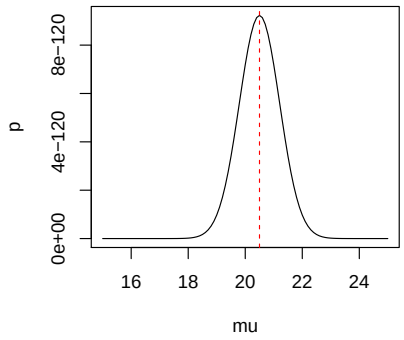
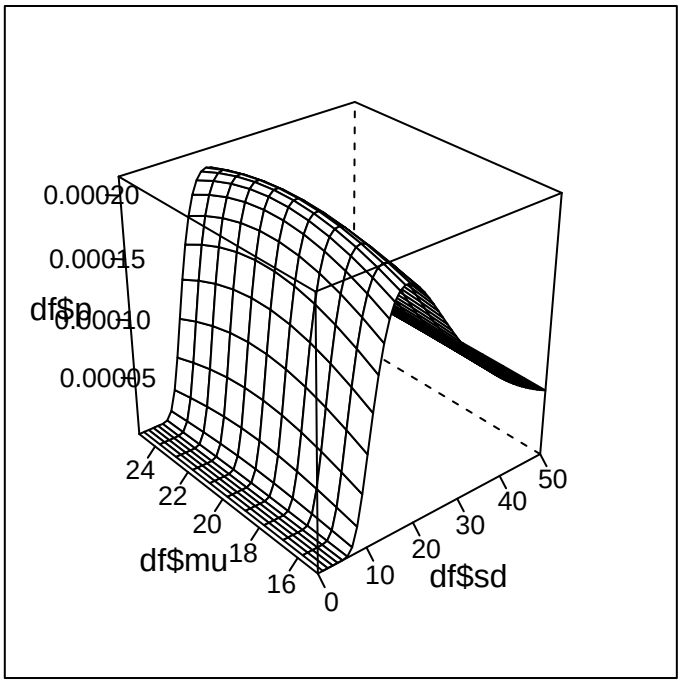


Figure 7: We calculate $p(4, 37|\mu)$ for all possible values of μ and find that $\mu = 21$ was the most likely value for μ, given that we observed $x_1 = 4$ and $x_2 = 37$



7.3 Application to the Simple Regression Model

Consider our usual model:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

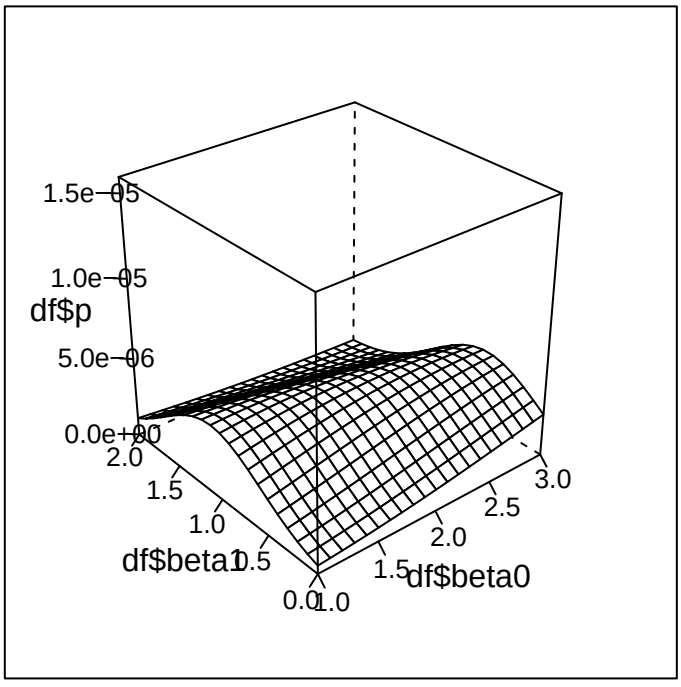
We assume that the error term is normally distributed with mean 0 and sd σ. Suppose that we observe $X = \{1, 2, 3\}$ and $Y = \{2, 3, 7\}$

This time we want to find β_0 and β_1 using maximum likelihood estimation (MLE). Again, we can search for them iteratively.[2]. This time the criteria will be how probable our error terms are given the parameters we've chosen.

[2] This method works, but is computationally very inefficient. I use it here only for intuition purposes. In practice, there are much more sophisticated algorithms to reach the solution)

```
df <- NULL
x <- c(1,2,3,4,5)
y <- c(2,3,7,8,3)
for(beta0 in seq(1, 3, 0.1)){ #0.1
  for(beta1 in seq(0, 2, 0.1)){ #0.1
    for(sd in seq(0, 4, 0.2)){
    e <- y - (beta0 + beta1*x)
    #how probable are these e?
    p <- (1 / (sd*sqrt(2*pi)))*exp(-(0.5*((e-0)/sd)^2))
    df.tmp <- data.frame(beta0 = beta0, beta1 = beta1, sd = sd, p=prod(p))
    df <- rbind(df, df.tmp)
}
```

```
  }
  }
library(lattice)
wireframe(df$p ~ df$beta0 * df$beta1, scales=list(arrows=F))
df[which(df$p ==max(df$p, na.rm=T)),]

     beta0 beta1  sd            p
6774   2.5   0.7 2.2 1.592994e-05

lm1 <- lm(y ~ x); lm1


Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
        2.5          0.7

sigma.ols <- sqrt(sum(resid(lm1)^2)/3)
sigma.mle <- sqrt(sum(resid(lm1)^2)/5) # Note that this is the MLE estimate.
#It is slightly biased, because it divides by n and not n-k
```



7.4 Application to the Logit

In the case of a binary DV, there are two possible outcomes, and we can calculate the probability of each:

$$\begin{aligned} P(Y = 1|X) &= F(X\beta) \\ P(Y = 0|X) &= 1 - F(X\beta), \end{aligned} \tag{4}$$

where $F(\cdot)$ is the logistic function, i.e., $F(X\beta) = \frac{e^{X\beta}}{1+e^{X\beta}}$. Now we just need to realize that our dependent variable is the same as bernouilli variable, since it can take on 2 values, 1 or 0. So we can write:[3]

[3] Remember that the PDF (technically, the pmf) of the bernoulli distribution is $p^x(1-p^x)$

$$L_i = P(Y = y_i|X) = P(Y = 1|X)^{y_i} + P(Y = 0|X)^{1-y_i}$$

so our likelihood function is simply

$$L = \prod_1^n P(Y = y_i|X)$$