Econométrie 3

Binary Choice & Logistic Regression

Iragaël Joly

Grenoble-Inp, Génie Industriel - Gael

iragael.joly@grenoble-inp.fr

October 2018

Table of contents

Introduction (1)

Context

- Assume we observe the output from a binary choice
- We have to analyse a binary variable (0 or 1)
- Are the classic linear regression a suitable tool to analyse binary choice ?

Linear Probability Model

Linear Probability Model (1)

Simple Linear Regression Model

- $Y_i = \alpha + \beta X_i + u_i$
- assuming $E(u_i) = 0$
- $E(Y_i | X_i) = \alpha + \beta X_i$
- The hypothetical model is simple linear regression (Fig. 1)

Linear Probability Model (2)

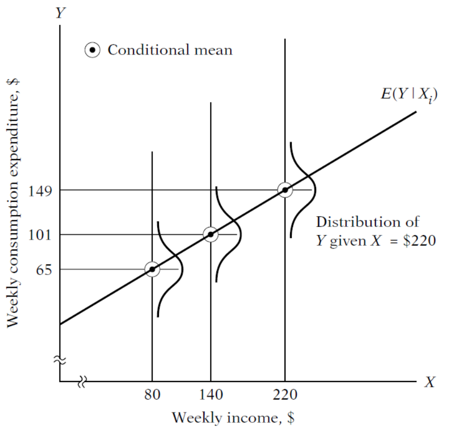


FIGURE 1 – Simple Linear Regression (Gujarati, 2003)

Linear Probability Model (1)

Simple Linear Regression Model

Hypothetical Data on Home Ownership and Income

FAMILY : 40 families

Y : Home Ownership where

1 :Owns a House

0 : Does Not Own a House

X :Family Income, Thousands of $

- Estimate Y over X
- Interpret this regression (intercept, slope, predicted probabilities, residuals)

Linear Probability Model (2)

```
tab151 <- read.table("Rscripts\\Tab151.txt", header = T)
head(tab151)
lm(data=tab151, Y~X)
```

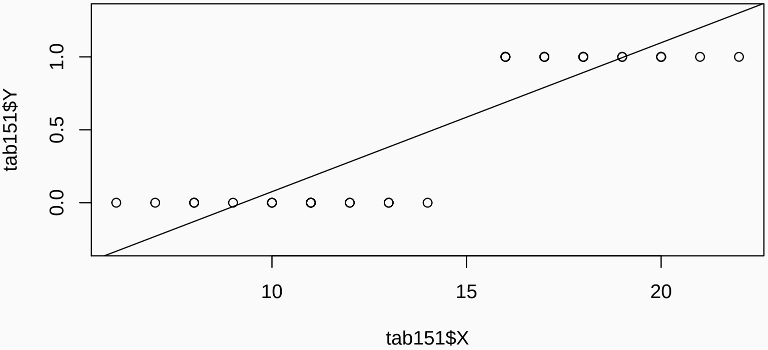
```
##   FAMILY Y  X
## 1      1 0  8
## 2      2 1 16
## 3      3 1 18
## 4      4 0 11
## 5      5 0 12
## 6      6 1 19
```

Linear Probability Model (3)

```
##
## Call:
## lm(formula = Y ~ X, data = tab151)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4842 -0.1777  0.0052  0.2095  0.3329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.94569    0.12284  -7.698 2.85e-09 ***
## X            0.10213    0.00816  12.515 4.73e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
##
## Residual standard error: 0.2264 on 38 degrees of freedom
## Multiple R-squared:  0.8048,Adjusted R-squared:  0.7996
```

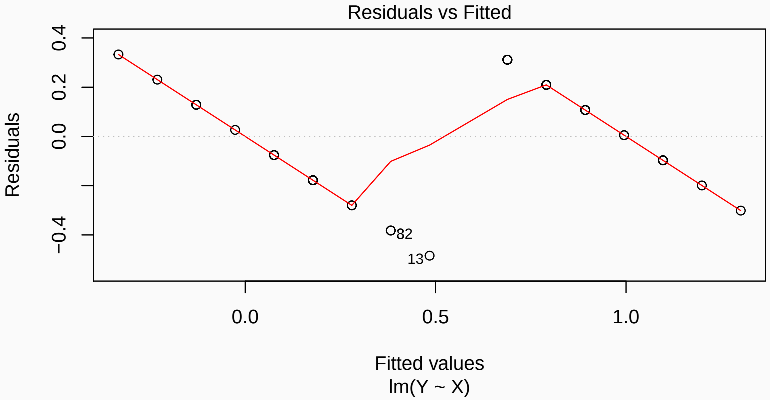
Linear Probability Model Problems (1)

Interpretation ?



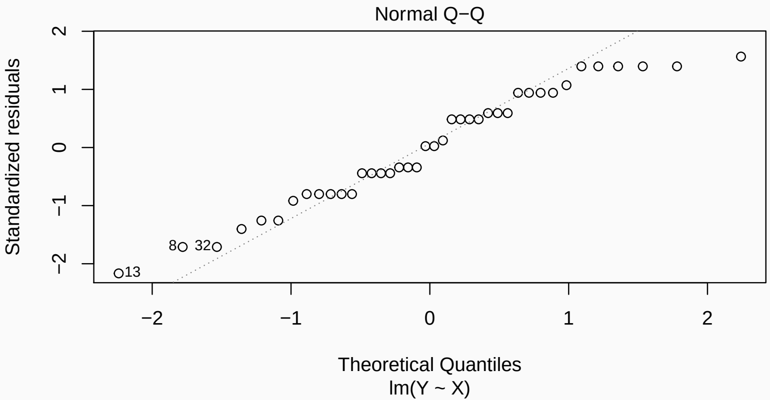
Linear Probability Model Problems (2)

Interpretation ?



Linear Probability Model Problems (3)

Interpretation?



Linear Probability Model Problems (4)

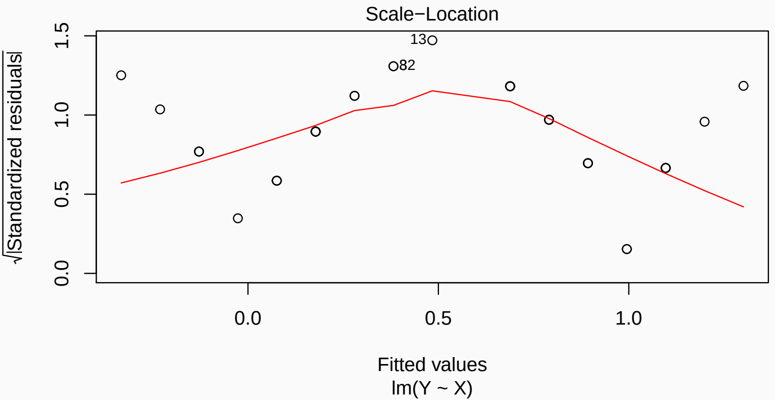
Non normality of the Disturbances u_i

The probability distribution of u_i is given by $u_i = Y_i - \alpha - \beta X_i$

	u_i	Prob.
When $Y_i = 1$	$1 - \alpha - \beta X_i$	P_i
When $Y_i = 0$	$\alpha - \beta X_i$	$(1 - P_i)$

Linear Probability Model Problems (5)

Interpretation ?



Linear Probability Model Problems (6)

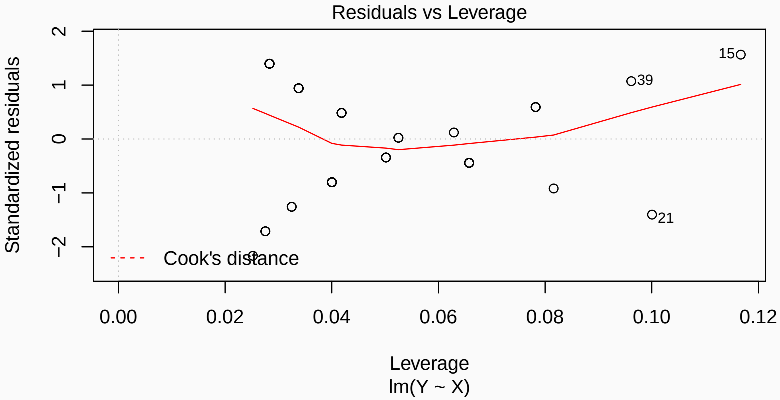
Heteroscedastic Variances of the Disturbances

Even if $E(u_i) = 0$ and $cov(u_i, u_j) = 0$

The variance of the u_i is

$$var(u_i) = P_i(1 - P_i)$$

Linear Probability Model Problems (7)



Problems with LPM (1)

Problems with LPM

LPM is plagued with several problems

- non-normality of u_i
- heteroscedasticity of u_i
- possibility of $\hat{Y}_i$ lying outside $[0, 1]$
- generally lower R^2

That can be surmountable...

- assumes $P_i = E(Y = 1|X)$ increases linearly with X : constant marginal effect of X

Problems with LPM (2)

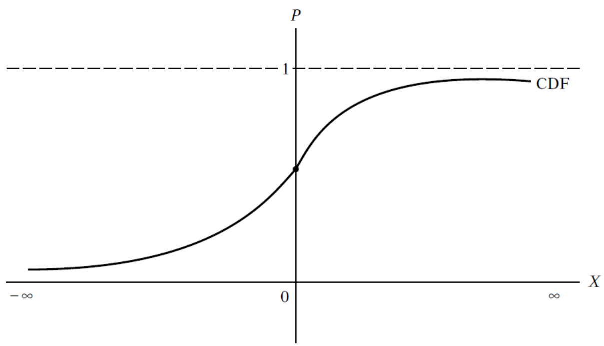


Figure 2 – A cumulative distribution function (cdf) *(Gujarati, 2003)*

Logistic Regression Model

Logit Model - v1

The LPM focuses on :

$$P_i = E(Y = 1|X_i) = \alpha + \beta X_i$$

Logit Model - v1

This linear form can be replaced by the logistic cdf :

$$P_i = \frac{1}{1 + e^{-Z_i}} = \frac{e^{Z_i}}{1 + e^{Z_i}}$$

where $Z_i = \alpha + \beta X_i$

- ranges between 0 and 1
- P_i is non linearly related to Z_i
- can be estimated after simple transformation : see *v2*

Logit Model - v2

The logistic model is a regression of the logit (log odds) (in favor of succes at X_i) :

$$L_i = \text{logit}(P_i) = ln\left(\frac{P_i}{1-P_i}\right) = \alpha + \beta X_i$$

We can write

$$\frac{P_i}{1-P_i} = \frac{1+e^{Z_i}}{1+e^{-Z_i}} = e^{Z_i}$$

Logit Model Interpretation

How can we interpret β ?

- The sign of L_i gives the relative position of P_i vs $1 - P_i$
- β measures the change in L_i for a unit change of X_i : the log-odds change !
- β sign determines whether $L_i(x)$ is increasing or decreasing as x increases.
- α is the value of the log-odds when $X_i = 0$...
- Logit v2 : odds are an exponential function of x.
  The odds multiply by e^{β} for every 1-unit increase in x.
  In other words, e^{β} is an odds ratio, the odds at $X = x + 1$ divided by the odds at $X = x$
- For dummy variables d : e^{β} is the odds ratio for $d = 1$ vs $d = 0$.

Logit Model Interpretation

Logistic regression function has curved appearance (rather than linear) then the rate of change in $P_i(x)$ per unit change in x varies

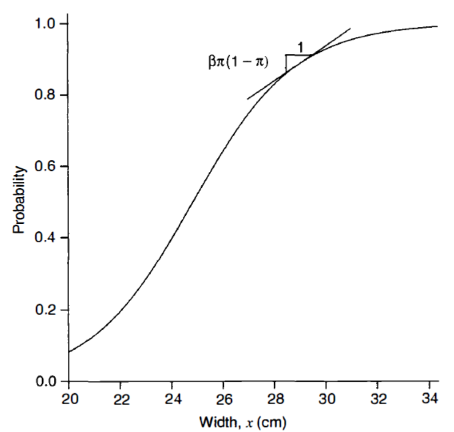


FIGURE 3 – Linear approximation to logistic regression curve *(Agresti, 2013)*

Logit Model Interpretation

- Calculating $\partial P_i(x)/\partial x$: Marginal effect (m.e.) :

$$\frac{\partial P_i(x)}{\partial x_i} = \beta \times P_i(x) \times (1 - P_i(x))$$

  (fairly complex function)
- m.e. is large when $P_i(x) \sim 0.5$ (steepest slope)
  that x value is $x = -\alpha/\beta$
- Near x where $P_i(x) = 1/2$, a change in x of $1/\beta$ corresponds to a change in $P_i(x)$ of roughly $(1/\beta)(\beta/4) = 1/\beta$
  that is $1/\beta$ approximates the distance between x values where $P_i(x) = 0.50$ and where $P_i(x) = 0.25$ or 0.75.
- Since the rate of change varies with x, summary of them is average partial effects : average of $\beta P_i(x)[1 - P_i(x)]$ for the subjects in the sample

Logit Model Interpretation

Derivative of the probability

$$\begin{aligned}\frac{\partial P_i(x)}{\partial x} &= \frac{\partial(e^{Z_i}/1+e^{Z_i})}{\partial x}\\ &= \frac{e^{Z_i}(1+e^{Z_i})-e^{Z_i}e^{Z_i}}{(1+e^{Z_i})^2}\frac{\partial Z_i}{\partial x}\\ &= \frac{e^{Z_i}}{(1+e^{Z_i})^2}\frac{\partial Z_i}{\partial x}\\ &= \frac{\partial Z_i}{\partial x}P_i(1-P_i)\end{aligned}$$

With $P_i(x) = \frac{e^{Z_i}}{1+e^{Z_i}}$ and $1 - P_i(x) = \frac{1}{1+e^{Z_i}}$

Logit Model Interpretation

Computing marginal effects

- Quantitative x

  1. Evaluate the marginal effects at the sample means of the data ($\bar{x}$)
  2. Evaluate the average partial effects : $APE = E[\frac{\partial P_i(x)}{\partial x}]$
     use the sample average of the individual marginal effects ($\bar{m.e.}$)
     $\frac{\partial P_i(x)}{\partial x} = \frac{\partial F(t)}{\partial t}\frac{\partial t}{\partial x} = f(\alpha + \beta x)\beta$
     $A\hat{P}E = \frac{1}{n}\sum_{i=1}^{n} f(\hat{\alpha} + \hat{\beta} X_i)\hat{\beta}$
     with the logistic c.d.f. : $F(x) = \frac{e^x}{1+e^x}$ and $f(x) = \frac{e^x}{(1+e^x)^2}$ (or the probit density function for probit)
  3. Equivalent results in large samples
  4. Current practice favors averaging the individual marginal effects

Logit Model Interpretation

Computing marginal effects

- Marginal effect for dummy variables d :

$$m.e. = P(Y = 1|\bar{x}_{(d)}, d = 1) - P(Y = 1|\bar{x}_{(d)}, d = 0)$$

where $\bar{x}_{(d)}$ denotes the means of all other variables.

Logit Model Interpretation

Elasticity of the probability

$$
\begin{aligned}
E_x &= \frac{\partial P_i(x)}{\partial x}\frac{x}{P_i} \\
&= \frac{\partial Z_i}{\partial x} P_i(1-P_i)\frac{x}{P_i} \\
&= \frac{\partial Z_i}{\partial x} x(1-P_i)
\end{aligned}
$$

If Z_i is linear in x with β, then $E_x = \beta x(1-P_i)$

Estimation and Inference in Binary Logistic Regression

Estimation in Binary Logit

Estimation in Binary Logit

- Estimation of binary models is usually based on ML.
- Each observation is a single draw from a Bernoulli distribution (binomial with one draw).
- The model (resp. Logit) with success probability $P_i = F(\alpha + \beta X_i)$ (resp. $F(\alpha + \beta X_i) = \frac{1}{1+e^{\alpha+\beta X_i}}$)
  and n iid observations
- leads to the joint probability (likelihood function) :

$$P(Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n | X) = \prod_{i=1}^{n} F_i(Y_i)$$

$$= \prod_{y_i=1} F(\alpha + \beta X_i) \prod_{y_i=0} [1 - F(\alpha + \beta X_i)]$$

Estimation in Binary Logit

- The likelihood function for a sample of n observations

$$\mathcal{L} = f(Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n | X) = \prod_{i=1}^{n} P_i^{y_i} (1 - P_i)^{1-y_i}$$

$$= \prod_{i=1}^{n} [F(\alpha + \beta X_i)]^{y_i} [1 - F(\alpha + \beta X_i)]^{1-y_i}$$

- the log likelihood function :

$$ln\mathcal{L} = lnf(Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n | X) =$$

$$\sum_{i=1}^{n} y_i ln[F(\alpha + \beta X_i)] + (1 - y_i) ln[1 - F(\alpha + \beta X_i)]$$

- with the logit function : $F(\alpha + \beta X_i) = \frac{1}{1+e^{\alpha+\beta X_i}}$ for logit model

Estimation in Binary Probit

Estimation in Binary Probit

- For the normal distribution, the log-likelihood is

$$lnL = \sum_{y_i=1} \Phi(\alpha+\beta X_i) \sum_{y_i=0} ln[1-\Phi(\alpha+\beta X_i)]$$

Likelihood and Properties of estimates

Likelihood equations

Likelihood equations

$$\frac{\partial ln\mathcal{L}}{\partial\beta} = \sum_i^n \left(\frac{y_i f_i}{F_i} + (1 - y_i)\frac{-fi}{1 - F_i} \right) x_i = 0$$

- where $fi = dF_i/d(\beta x_i)$
- choice of F_i leads to the empirical model

Likelihood equations (1)

Likelihood equations

- Unless for LPM, these equation are non-linear and require an iterative solution
- if x_i contains a constant term, the first-order conditions imply that the average of the predicted probabilities must equal the proportion of ones in the sample

Optimisation method (1)

Likelihood equations

- Second derivatives for the logit model :

$$H = \frac{\partial^2 ln\mathcal{L}}{\partial\beta\partial\beta'}$$

  do not involve the random variable y_i , so Newton's method is also the method of scoring
- Hessian is always negative definite, so the log-likelihood is globally concave (less obvious for probit)
- Newton's method will usually converge to the maximum of the log-likelihood in just a few iterations

Asymptotic covariance matrix (1)

Likelihood equations

- Asymptotic covariance matrix for the maximum likelihood estimator can be estimated by using the inverse of the Hessian evaluated at the maximum likelihood estimates
- Hessian for the logit model does not involve y_i, so $H = E[H]$, not true for probit.

Asymptotic covariance matrix (1)

For the logit model :

$$\mathcal{L} = \prod_i^n p_i^{y_i}(1 - p_i^{1-y_i}) = \prod_i^n \left(\frac{p_i}{1-p_i}\right)^{y_i}(1-p_i)$$

$$\begin{aligned} ln(\mathcal{L}) &= \sum_i y_i ln\left(\frac{p_i}{1-p_i}\right) + \sum_i ln(1-p_i) \\ &= \sum_i \beta x_i y_i - \sum_i ln(1+e^{\beta x_i}) \end{aligned}$$

Asymptotic covariance matrix (2)

For the logit model :

$$\frac{\partial ln\mathcal{L}}{\partial \beta} = \sum_i x_i y_i - \sum_i x_i (1 + e^{-\beta x_i})^{-1}$$

$$= \sum_i x_i y_i - \sum_i x_i \hat{y}_i$$

$$\frac{\partial^2 ln\mathcal{L}}{\partial \beta \partial \beta} = -\sum_i^n x_i x_i' \hat{y}_i (1 - \hat{y}_i)$$

where $\hat{y}_i = \frac{1}{1+e^{-\beta x_i}}$

Exercise : Binary Logit and Probit

Estimating a Binary Logit

Exercise : Effect of Personalized system of instruction

- Research question [a] : Does a new method of teaching economics (Personalized System of Instructions, PSI) influence performance in later economics courses
- Data available
- GRADE : Indicator of improving grade between basic and intermediate economics courses (binary)
- GPA : Grade point average (number in range)
- TUCE : Score on pretest : entering knowledge (number in range)
- PSI : Exposure to new teaching method (binary)

---

a. Example *(Greene, 2008)* p.694 : Study of *(Spector and Mazzeo, 1980)*
Data : http ://people.stern.nyu.edu/wgreene/Text/Edition6/tablelist6.htm

Estimating a Binary Logit

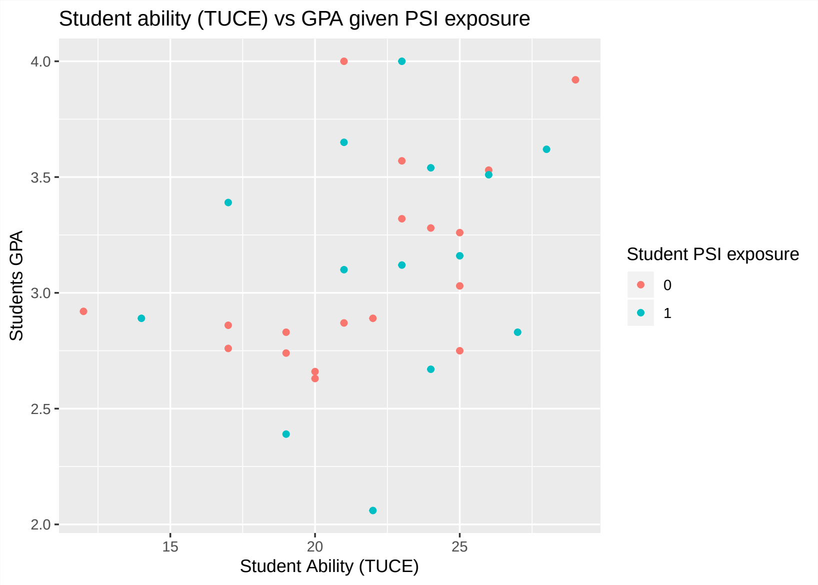
Exercise : Effect of Personalized system of instruction

```r
library(faraway)
head(spector)
spector$psi <- factor(spector$psi)
spector$grade <- factor(spector$grade)
```

```
##   grade psi tuce  gpa
## 1     0   0   20 2.66
## 2     0   0   22 2.89
## 3     0   0   24 3.28
## 4     0   0   12 2.92
## 5     1   0   21 4.00
## 6     0   0   17 2.86
```

Estimating a Binary Logit

```
ggplot(data = spector, aes(x=tuce, y=gpa))
  + geom_point(aes(colour=psi))
  + ggtitle("Student Ability (TUCE) vs GPA given PSI exposure")
  + xlab("Student Ability (TUCE)") + ylab("Students GPA")
  + labs(colour="Student PSI exposure")
```


Student ability (TUCE) vs GPA given PSI exposure
4.0
3.5
3.0
2.5
2.0
Students GPA
15
20
25
Student Ability (TUCE)
Student PSI exposure
0
1

```r
ggplot(data = spector, aes(x=tuce, y=gpa, colour=psi))
+ geom_point() + geom_smooth(method = 'lm', formula = y~x, se=FALSE)
+ ggtitle("Student Ability (TUCE) vs GPA given PSI exposure")
+ xlab("Student Ability (TUCE)") + ylab("Students GPA")
+ labs(colour="Student PSI exposure")
```


Student ability (TUCE) vs GPA given PSI exposure
Students GPA
4.0
3.5
3.0
2.5
2.0
15
20
25
Student Ability (TUCE)
Student PSI exposure
0
1

Estimating the corresponding Binary Logit

Exercise

- Let estimate the logit model :

$$L_i = ln\left(\frac{P_i}{1-P_i}\right) = \alpha + \beta_2 GPA_i + \beta_3 TUCE_i + \beta_3 PSI_i + u_i$$

- Calculate effects of variables on odds-ratio
- Calculate marginal effects on probabilities at the mean point

```r
Logit1 <- glm(grade ~ gpa + tuce + psi, x = TRUE,
    data = spector, family = binomial(link = "logit"))
summary(Logit1)
exp(coefficients(Logit1))
```

```
##
## Call:
## glm(formula = grade ~ gpa + tuce + psi, family = binomial(link = "logi
##     data = spector, x = TRUE)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9551  -0.6453  -0.2570   0.5888   2.0966
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.02135    4.93127  -2.641  0.00828 **
## gpa           2.82611    1.26293   2.238  0.02524 *
## tuce          0.09516    0.14155   0.672  0.50143
## psi1          2.37869    1.06456   2.234  0.02545 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 41.183  on 31  degrees of freedom
## Residual deviance: 25.779  on 28  degrees of freedom
```

Odds Ratios

```
## [1] "exp(beta)"
##  (Intercept)          gpa         tuce         psi1
## 2.212590e-06 1.687971e+01 1.099832e+00 1.079073e+01
```

```r
# Logit  # xb*:
betas<-t(data.frame(coef(Logit1)))  ;  betas
xmean <- c(1, mean(spector$gpa), mean(spector$tuce),
           mean(as.numeric(spector$psi))-1)
xmean
print("XBetas:")
xb_logit <- sum(xmean*betas)  ;  xb_logit
# Slopes (at mean): Lambda(mean(xb))*(b)
print("Slopes:")
logit_slopes <- dlogis(xb_logit)*betas  ;  logit_slopes
```

```
##               (Intercept)      gpa       tuce     psi1
## coef.Logit1.   -13.02135 2.826113 0.09515766 2.378688
## [1]  1.000000  3.117188 21.937500  0.437500
## [1] "XBetas:"
## [1] -1.083627
## [1] "Slopes:"
##               (Intercept)       gpa       tuce      psi1
## coef.Logit1.   -2.459761 0.5338588 0.01797549 0.4493393
```

Estimating the corresponding Binary Probit

Exercise

- Let estimate the probit model
- Calculate effects of variables on odds-ratio
- Calculate marginal effects on probabilities at the mean point

```r
probit1 <- glm(grade ~ gpa + tuce + psi, x = TRUE,
    data = spector, family = binomial(link = "probit"))
summary(probit1)
```

```
##
## Call:
## glm(formula = grade ~ gpa + tuce + psi, family = binomial(link = "prob
##     data = spector, x = TRUE)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9392  -0.6508  -0.2229   0.5934   2.0451
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.45231    2.57152  -2.898  0.00376 **
## gpa          1.62581    0.68973   2.357  0.01841 *
## tuce         0.05173    0.08119   0.637  0.52406
## psi1         1.42633    0.58695   2.430  0.01510 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 41.183  on 31  degrees of freedom
## Residual deviance: 25.638  on 28  degrees of freedom
```

Optional : we could calculate the CI of the estimates

```
confint(probit1)

## Waiting for profiling to be done...

##                   2.5 %     97.5 %
## (Intercept) -13.1148789 -2.9737780
## gpa           0.3740450  3.1314162
## tuce         -0.1044397  0.2279414
## psi1          0.3286434  2.7031621
```

```r
# Probit  # xb*:
betas<-t(data.frame(coef(probit1))) ; betas
xmean <- c(1, mean(spector$gpa), mean(spector$tuce),
          mean(as.numeric(spector$psi))-1)
print("XBetas")
xb_probit <- sum(xmean*betas) ; xb_probit
# Slopes (at mean): Lambda(mean(xb))*(b)
print("Slopes")
probit_slopes <- dnorm(xb_probit)*betas  ;  probit_slopes
```

```
## [1] "Betas"
##               (Intercept)      gpa       tuce     psi1
## coef.probit1.   -7.452313 1.625812 0.05172846 1.426331
## [1] "XBetas:"
## [1] -0.625539
## [1] "Slopes:"
##               (Intercept)       gpa       tuce      psi1
## coef.probit1.   -2.444734 0.5333484 0.01696954 0.4679083
```

Comparison of marginal effects at mean :

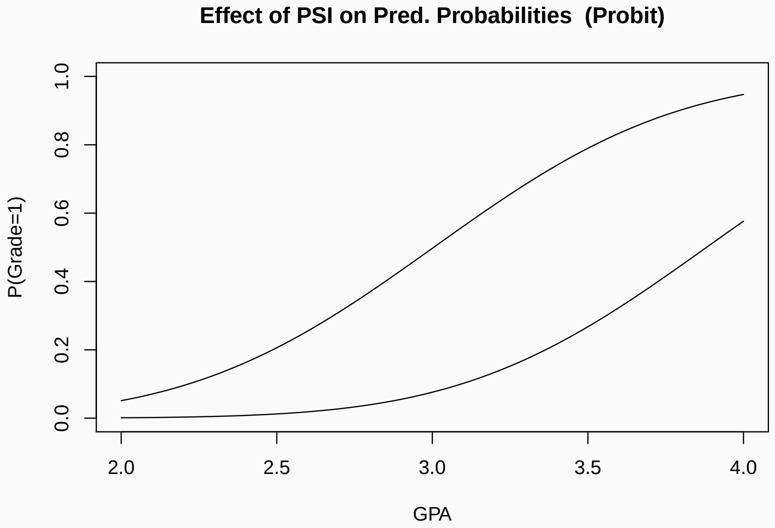
```
## [1] "PROBIT MEM"
##                (Intercept)       gpa       tuce      psi1
## coef.probit1.   -2.444734 0.5333484 0.01696954 0.4679083
## [1] "LOGIT MEM"
##               (Intercept)       gpa       tuce      psi1
## coef.Logit1.   -2.459761 0.5338588 0.01797549 0.4493393
```

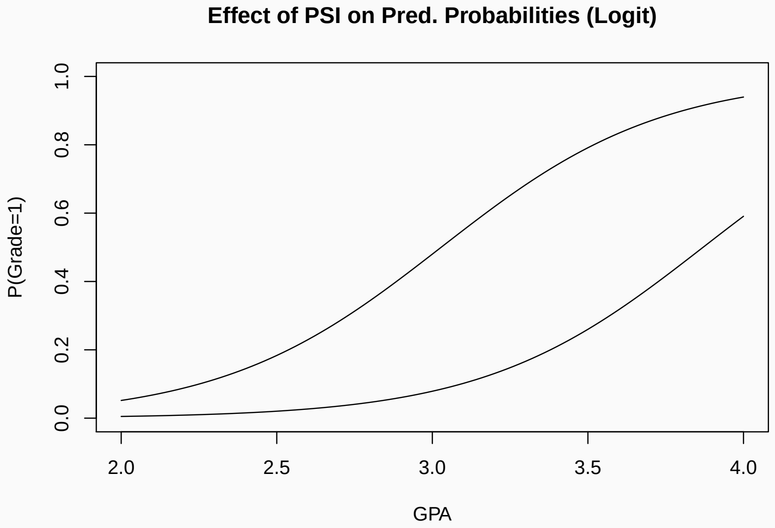
Estimating the corresponding Binary Logit

Exercise

- Draw Predicted probabilities with varying GPA, fixed TUCE (at its average), and for PSI $= 0$ or PSI $= 1$, for logit and probit

```r
# Probability Curves for the Probit model
psi0 <- function(x) pnorm(-7.452 + 1.626*x  + 0.052*21.938)
psi1 <- function(x) pnorm(-7.452 + 1.626*x  + 0.052*21.938 + 1.426)
curve(psi0, xlim=c(2,4), ylim=c(0,1), main="Effect of PSI on Pred.
      Probabilities (Probit)", ylab="P(Grade=1)", xlab="GPA")
curve(psi1, add=T)
# Probability Curves for the Logit model
psi0 <- function(x) plogis(-13.021 + 2.826*x  + 0.095*21.938)
psi1 <- function(x) plogis(-13.021 + 2.826*x  + 0.095*21.938 + 2.379)
curve(psi0, xlim=c(2,4), ylim=c(0,1), main="Effect of PSI on Pred.
      Probabilities (Logit)", ylab="P(Grade=1)", xlab="GPA")
curve(psi1, add=T)
```


Effect of PSI on Pred. Probabilities (Probit)
P(Grade=1)
0.0
0.2
0.4
0.6
0.8
1.0
2.0
2.5
3.0
3.5
4.0
GPA


Effect of PSI on Pred. Probabilities (Logit)
P(Grade=1)
0.0
0.2
0.4
0.6
0.8
1.0
2.0
2.5
3.0
3.5
4.0
GPA

Estimating the corresponding Binary Logit

Exercise

- Calculate Average Partial Effect, for logit and probit

Average Marginal effects (1)

`predict()` returns $X\beta$

```r
# Probit
fav_probit <- mean(dnorm(predict(probit1, type = "link")))
fav_probit * coef(probit1)
# Logit
fav_Logit <- mean(dlogis(predict(Logit1, type = "link")))
fav_Logit * coef(Logit1)
```

```
## [1] "Probit:"
## (Intercept)         gpa        tuce        psi1
## -1.65375669  0.36078701  0.01147916  0.31651968
## [1] "Logit:"
## (Intercept)         gpa        tuce        psi1
## -1.67059543  0.36258083  0.01220841  0.30517770
```

Average Marginal effects (2)

- LPM, logit, and probit give qualitatively similar results
- Between logit and probit, which model is preferable ?
- In most applications the models are quite similar
- Main difference being slightly fatter tails of the logistic distribution
- Conditional probability P_i approaches 0 or 1 at a slower rate in logit than in probit
- There is no compelling reason to choose one over the other
- In practice, logit model because of its comparative mathematical simplicity

Inference in Binary Logit Model

Computing Standard Errors

Asymptotic Covariance Matrix

- The asymptotic covariance matrix for the maximum likelihood estimator can be estimated by using the inverse of the Hessian evaluated at the maximum likelihood

$$H = \frac{\partial^2 lnL}{\partial\beta\partial\beta'}$$

$$H = \begin{bmatrix} \frac{\partial^2 lnL}{\partial\beta_1^2} & \frac{\partial^2 lnL}{\partial\beta_1\partial\beta_2} & \cdots & \frac{\partial^2 lnL}{\partial\beta_1\beta_k} \\ \frac{\partial^2 lnL}{\partial\beta_2\partial\beta_1} & \frac{\partial^2 lnL}{\partial\beta_2^2} & \cdots & \frac{\partial^2 lnL}{\partial\beta_2\beta_k} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 lnL}{\partial\beta_k\partial\beta_1\beta_k} & \frac{\partial^2 lnL}{\partial\beta_k\beta_2} & \cdots & \frac{\partial^2 lnL}{\partial\beta_k^2} \end{bmatrix}$$

$-\nabla 2L(\beta^*)^{-1}$ is a consistent estimator of the variance-covariance matrix of the estimates.

Computing Standard Errors (1)

Computing Standard Errors of partial effects

- Standard errors for partial effects are usually computed using the delta method.
- An alternative approach is the method of *(Krinsky and Robb, 1986)* .

Hypothesis tests (1)

Three familiar procedures for conventional hypothesis tests about restrictions on the model coefficients

Likelihood Ratio test

- The likelihood ratio statistic is : $\lambda_{LR} = 2[lnL_1 - lnL_0](ddl)$
- where lnL_1 indicates the log-likelihood computed at the unrestricted (alternative) estimator
- lnL_0 at the restricted (null) estimator
  item And *dll* are the degree of freedom = number of restrictions

Hypothesis tests (2)

Wald test

- Hypothesis : $r(\theta, c) = 0$
  $r(\theta, c)$ is a vector of J functionally independent restrictions on θ and c is a vector of constants.
- The Wald statistic uses the delta method to obtain an asymptotic covariance matrix for $r(\theta, c)$.
  The statistic is : $W = r(\theta, c)'[Var(r(\theta, c))]^{-1} r(\theta, c)$

Hypothesis tests (3)

Lagrange multiplier (LM) statistic

- $LM = g' V g$,
- where g is the first derivatives of the unrestricted model evaluated at the restricted parameter vector
- V is the estimator of the asymptotic covariance matrix of the maximum likelihood estimator.

Example : Wald test

```r
# For Probit
# a) joint significance of all regressors
#install.packages("aod")
library(aod)
wald.test(b = coef(probit1), Sigma = vcov(probit1),
          Terms =2:length(coef(probit1)))
# b) linear combination of coefficients
# (e.g. are the coefficients signif. different?)
restr <- cbind(0, -1, 1, 0)
wald.test(b = coef(probit1), Sigma = vcov(probit1), L = restr)
# LR test
library(lmtest)
probit0 <- glm(grade ~ 1 , x = TRUE,
    data = spector, family = binomial(link = "probit"))
summary(probit0)
lrtest(probit1, probit0)
```

Example : Wald test

a) joint significance of all regressors

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 10.6, df = 3, P(> X2) = 0.014
```

b) linear combination of coefficients

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 4.8, df = 1, P(> X2) = 0.028
```

Example :LR test

```
## Likelihood ratio test
##
## Model 1: grade ~ gpa + tuce + psi
## Model 2: grade ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -12.819
## 2   1 -20.592 -3 15.546   0.001405 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fit Measures for Binary Choice (1)

ρ^2 or LRI

In Logit and Probit models :

- There is no R squared
- There are no residuals or sums of squares
- The model is not computed to optimize the fit of the model to the data
- Null Deviance : $= -2ln(L_0)$, with L_0 likelihood of the model with only intercept.

Fit Measures for Binary Choice (2)

```
sum(as.numeric(spector$grade)-1 )

## [1] 11

length(spector$grade)

## [1] 32

sum(as.numeric(spector$grade)-1 )/length(spector$grade)

## [1] 0.34375

-2*(11*log(0.34375)+21*log(1-0.34375))

## [1] 41.18346
```

```r
anova(probit0,test="Chisq")

## Analysis of Deviance Table
##
## Model: binomial, link: probit
##
## Response: grade
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                    31     41.183
```

```r
anova(probit1,test="Chisq")

## Analysis of Deviance Table
##
## Model: binomial, link: probit
##
## Response: grade
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                    31     41.183
## gpa   1   8.3744        30     32.809 0.003805 **
## tuce  1   0.5048        29     32.304 0.477420
## psi   1   6.6667        28     25.638 0.009823 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The full model or satured model is : $E(Y_i) = y_i$: mean of the variable is defined by the observation.
- in our case $p = 1$ if $y_i = 1$ and $p = 0$ if $y = 0$. The likelihood is 1 and $lnL = 0$
- Null deviance is : $D_0 = -2ln\frac{L_0}{L_f} = 2(LL_f - LL_0)$
- Residual deviance is $D_x = -2ln\frac{L_x}{L_f} = 2(LL_f - LL_x)$
- Variable effet is measured by $D_x - D_0 = -2ln\frac{L_x}{L_0} = 2(LL_0 - LL_x)$

- Generalized R-squared : $R^2 = 1 - exp\{-\frac{L^2}{n}\}$
- Likelihood ratio index (LRI) ou Pseudo-R^2 :

$$\rho^2 = 1 - \frac{lnL}{lnL_0}$$

  Where $L(\beta^*)$ is the Likelihood of the estimated model (full model) and $L(0)$ the Likelihood of the model with no regressors (only intercept)
- McFadden R^2 : $R^2_{MCF} = 1 - \frac{ln(L)}{ln(L_0)}$
- Cox and Snell R^2 is : $R^2_{C\&S} = 1 - (\frac{L_0}{L})^{2/n}$

Likelihood Ratio Index

- $LRI = 1 - (lnL/lnL_0)$
- Bounded by 0 and 1
- If all the slope coefficients are zero, then LRI equals zero.
- There is no way to make LRI equal 1, although one can come close
- LRI rises when the model is expanded

To Compare Models :

You can also use :

- logL
- Use information criteria to compare nonnested models (AIC, BIC,...)
  - $AIC = -2 \times lnL + 2 \times k$, where k represents the number of parameters in the fitted model
  - It should be used when the number of fitted parameters is large compared to sample size, i.e., when $n/k < 40$ (Hurvich and Tsai, 1995).
  - $BIC = -2 \times lnL + k \times log(n)$,

Adjusted Likelihood ratio index (ALRI)

- ρ^2 is increasing with the number of parameters.
- A higher fit (that is a higher ρ^2) does not mean a better model.
- An adjustment is needed.

$$\bar{\rho}^2 = 1 - \frac{\mathcal{L}(\hat{\beta}) - K}{\mathcal{L}(0)}$$

Example : Log-Likelihood calculation with R

- $(-2) \times$ log-likelihood
- Likelihood Ratio Index : $LRI = 1 - (lnL/lnL_0)$

```r
logL_probit<-logLik(probit1)  ;  -2*logL_probit

## 'log Lik.' 25.63761 (df=4)

logL_logit<-logLik(Logit1)  ;  -2*logL_logit

## 'log Lik.' 25.77927 (df=4)

LRI_probit<-with(probit1,deviance/null.deviance) # -2 cancels out
LRI_probit<-1-LRI_probit
LRI_logit<-with(Logit1,deviance/null.deviance)
LRI_logit<-1-LRI_logit
LRI<-data.frame(LRI_probit,LRI_logit)
LRI

##   LRI_probit LRI_logit
## 1   0.377478 0.3740383
```

```r
AIC_logit <- -2*logL_logit+2*length(coef(Logit1)) ; AIC_logit

## 'log Lik.' 33.77927 (df=4)

AIC_probit <- -2*logL_probit+2*length(coef(probit1)) ; AIC_probit

## 'log Lik.' 33.63761 (df=4)
```

Fit Measures Based on Predictions (1)

Predicting the Outcome

- Predicted probabilities and Predicting outcomes :
  Predict $Y = 1$ if $\hat{P}$ is "large"
  Use 0.5 for "large" (more likely than not)
- Count successes and failures
- For example, suppose $n = 200$, we observe $n_{Y_i=0} = 180$, we predict $n_{\hat{Y}_i=1} = 150$: we predict at least 75% of the outcomes correctly.
  Because of this, report the percent correctly predicted for each of the two outcomes.

A Test of Structural Stability (1)

Structural Change Over Groups

- Counterpart to the Chow test for linear models.
- Fit the same model in each subsample
- Unrestricted log likelihood is the sum of the subsample log likelihoods : lnL_1
- Pool the subsamples, fit the model to the pooled sample
- Restricted log likelihood is that from the pooled sample : lnL_0

$$\chi^2(dll) = 2 \times (lnL_1 - lnL_0)$$

with $dll = (K-1)\times$ model size.

A Test of Structural Stability (1)

Exercise : PSI

- Test the structural change over PSI groups

$$\chi^2(dll) = 2 \times (lnL_1 - lnL_0)$$

with $dll = (\text{nb restriction} - 1) \times \text{nb parameters in model}$.

```r
library(faraway)
head(spector)
```

```
##   grade psi tuce  gpa
## 1     0   0   20 2.66
## 2     0   0   22 2.89
## 3     0   0   24 3.28
## 4     0   0   12 2.92
## 5     1   0   21 4.00
## 6     0   0   17 2.86
```

```r
spector$psi <- factor(spector$psi)
spector$grade <- factor(spector$grade)
```

```r
Logit1 <- glm(grade ~ gpa + tuce + psi, x = TRUE,
    data = spector, family = binomial(link = "logit"))
LogitPSI0 <- glm(grade ~ gpa + tuce, x = TRUE,
    data = spector[spector$psi==0,], family = binomial(link = "logit"))
LogitPSI1 <- glm(grade ~ gpa + tuce , x = TRUE,
    data = spector[spector$psi==1,], family = binomial(link = "logit"))
summary(Logit1) ; summary(LogitPSI1) ; summary(LogitPSI0) ;
LL1 <- logLik(Logit1)
LLPSI1 <- logLik(LogitPSI1)
LLPSI0 <- logLik(LogitPSI0)
ChiT <- 2* ( LLPSI0+LLPSI1) - (LL1)  ;  ChiT
ChiTddl <- (2-1)*length(coef(Logit1)); ChiTddl
qchisq(0.95,ChiTddl)
```

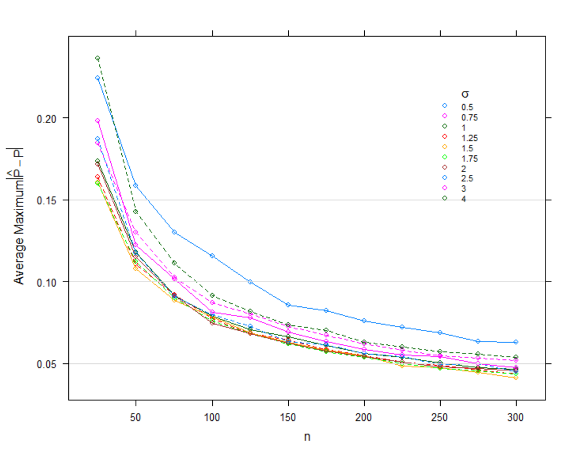
```
## 'log Lik.' -10.98064 (df=3)
## [1] 4
## [1] 9.487729
```

Minimum Sample Size Requirement (1)

- Suppose no covariates, only the intercept.
  What is the sample size required to a precision of 0.1 of the predicted probability with 0.95 confidence, when the true intercept near from zero ?
- The answer is $n = 96$
- The general formula for the sample size required to achieve a margin of error of δ in estimating a **true probability** of θ at the 0.95 confidence level is $n = \left(\frac{1.96}{\delta}\right)^2 \times \theta(1-\theta)$.
  Set $\theta = 1/2$ (intercept=0) for the worst case.

Minimum Sample Size Requirement (2)

FIGURE 4 – Simulated expected maximum error in estimating probabilities for $x \in [-1.5, 1.5]$ with a single normally distributed X with mean zero



Exercise (1)

Exercise MROZ - Woolridge

- Reproduce results from the textbook

Références

Agresti, A. (2013). *Categorical Data Analysis*. Wiley edition.

Greene, W. (2008). *Econometric Analysis, 6th*. Prentice-HallOxford : Clarendon Press edition.

Gujarati, D. (2003). *Basic Econometrics*. McGraw Hill edition.

Krinsky, I. and Robb, A. (1986). On approximating the statistical properties of elasticities. *The Review of Economics and Statistics*, 64 :715–719.

Spector, L. C. and Mazzeo, M. (1980). Probit analysis and economic education. *Journal of Economic Education*, 11 :37–44.