R Applications for Binary Response Models

Introductory Remarks

This file builds on the previous R Activities. If you have not yet gone through those files, please make sure to do so before tackling this R Activity.

New packages

Here we will discuss how to estimate and run binary response models, including the linear probability model (LPM) and non-linear models such as probit and logit. For this purpose, we will use the *lm* function and the *glm2* function of the packages *GLM2* and *STATS*. To start: make sure you have correctly installed the GLM2 and STATS package and that the R documentation page loads correctly. Read the R *GLM2* documentation for details on how the function is specified.

```r
library(stats)

install.packages("GLM2", dependencies = TRUE)
library(glm2)
view(glm2)
```

We will use three additional packages *sandwich*, that computes heteroskedasticity robust standard errors (HAC), *lmtest*, that allows diagnostic checking in linear and parametric regression models and *mfx* that will allow us to compute marginal effects.

```r
install.packages("lmtest", dependencies = TRUE)
install.packages("sandwich", dependencies = TRUE)
install.packages("mfx", dependencies = TRUE)

library(lmtest)
library(sandwich)
library(mfx)
```

Make sure you have the following packages, that we used in previous R activities, loaded in your current R session.

```r
library(dplyr)
library(wooldridge)
library(ggplot2)
```

Loading the Relevant Dataset

Before loading the relevant dataset, make sure to clear your work environment in R, install and/or load the relevant packages, and set your working directory. The accompanying R script does this step by step and a detailed description of how to do this was provided in the R introduction in Block 1.

The dataset we will work with today is called “recid”. Instead of loading it into R, we will call upon this dataset “within R.” In our case, we are working with a dataset that is used in the coursebook by Wooldridge. All of these datasets are available within a package called “wooldridge” in R. To load the dataset for today make sure that you have installed and loaded the “wooldridge” package and then type:

```
data('recid')
data <-recid
  # loading data on recidivism from the Wooldridge package and saving it as 'data'
```

To check the full labels of the variable, you can go the R documentation on the package and find the *recid* description. This dataset contains data on 1400 individuals that were imprisoned alongside their individual characteristics and criminal history.

```
help(wooldrige)
```

Explore the data

First, explore the summary statistics with particular attention to whether the variables are binary, categorical or continuous.

```
summary(data)
```

We observe that the following variables are binary as they only contain two categories (Yes and No): Black, Alcohol, Drugs, Super, Married, Felon, Workprogram, Property, Person, Cens. The main outcome of interest here will be *Super*, whether a prisoner gets released early from prison or not. We can see that the probability that a prisoner gets released early is 0.69, or 69%.

Linear Probability Model (LPM)

A policy maker is interested in studying whether a recently launched work program within the prison system is effective in improving the behaviour of prisoner and allowing them to be released earlier. To answer this, we will run a linear probability model (LPM) on the outcome *super* on the independent variable *workprg* and several controls that are know to affect release: *married, priors, alcohol, age.*

```
model1 <-lm(super~ workprg+married+priors+alcohol+age, data)
```

From the lectures we know that the standard errors in a linear probability model are not homoskedastic, so we compute the heteroskedasticity-robust (HAC) standard errors and conduct asymptotic hypothesis testing using them.

```
coeftest(model1, vcov = vcovHC(model1))
```

We find that, ceteris paribus, a prisoner that takes part in a work program is 14 percentage points likelier to get released from prison, and this effect is significant as $p<0.05$.

We will save the fitted values from the LPM model so we can later compare them against the fitted values of the non-linear models.

```
data<-cbind(data, fitted = fitted(model1))
```

Non-linear model: Probit/Logit

Would a model that doesn't assume a linear probability be better at understanding a binary outcome? To ensure that predicted values always lie inside the [0,1] interval, we introduced the probit and logit models. To estimate these models, we use the Maximum Likelihood Estimator (MLE), for which we can use the *glm2* function. We will use the *family=binomial* for the binary response model, and select *link=probit* or *link=logit* depending on whether we are estimating the logit or probit model (for more details on the families, please see the R documentation). In the example that follows, we estimate the probit on the same outcome and controls as in the LPM model. We also save the fitted values.

```
probit1 <- glm2(super~ workprg+married+priors+alcohol+age, family = binomial(link = "probit"),
                data = data)
```

```
## model summary
summary(probit1)
data<-cbind(data, fitted_probit = fitted(probit1))
```

Average partial effects (APE) and Partial effects at the average (PEA)

How can we interpret the coefficients in the estimated probit or logit models? As we discussed in the lecture, the parameters estimated by probit/logit are not the partial effects we are interested in. The only thing the estimates are informative about is the direction of the partial effect (given by the sign of the estimate). We can see that the coefficient on *work program* is positive, hence participating in a work program increases the probability of early release, ceteris paribus.

The partial effects are not constant for the probit/logit model. It is common to report either the PEA (partial effect at the average) or the APE (average partial effect). We can obtain them using the command *probitmfx* for a probit model or the command *logitmfx* for a logit model:

```
mod_APE <- probitmfx(super~ workprg+married+priors+alcohol+age, data = data, atmean = FALSE)
# for APE
mod_APE


mod_PEA <- probitmfx(super~ workprg+married+priors+alcohol+age, data = data) # for PEA
mod_PEA
```

On average, ceteris paribus, we find that participating in a work program increases the probability of release by 14.7 percentage points.

In order to evaluate the impact of participating in a work program, we should compare the predicted probability of being released earlier depending on whether the individual participates in a *work program* or not. We may want to evaluate that for an individual with particular characteristics, for example for a specific individual who is married, without priors, doesn't drink and is 40 years old. To do so we find the predicted probabilities at these predictors, with and without having attended the work program.

```
person1 = data.frame(married=1, priors=0, alcohol=0, age=40, workprg=0)
person1_work = data.frame(married=1, priors=0, alcohol=0, age=40, workprg=1)

prob_person1=predict(probit1, person1, type="response")
prob_person1_work=predict(probit1, person1_work, type="response")
margins_workprg_person1=prob_person1_work-prob_person1
margins_workprg_person1
```

We find that for an individual who is married, without priors, doesn't drink and is 40 years old, taking part in a work program increases the probability of getting released from prison by 11.8 percentage points.

Joint hypothesis testing and probit/logit

Are alcohol and age also jointly important in explaining the probability of being released early? Should we include them as controls? To answer this, we conduct the Likelihood Ratio Test. We need to compare the log-likelihood (logL) of the unrestricted model with the log-likelihood of the restricted model where these variables are excluded. In the example below we estimate the restricted model and we then conduct the Likelihood Ratio test using the package we loaded earier *lmtest*.

```
probit2 <- glm2(super~ workprg+married+priors, family = binomial(link = "probit"),
                data = data)

lrtest (probit1, probit2)
```

From the results we can read that LR test statistic is equal to 1.8061 and the p-value is equal to 0.4053, so we cannot reject the null hypothesis. As these predictors are not jointly siginificant in determining the outcome, we can exclude them from our model.

Finally, we plot the fitted values of the LPM and Probit against the observations of *super* using the package *ggplot2* we covered in the previous R exercises.

```r
#scatterplot LPM
data %>%
  ggplot(aes(fitted, super)) +
  geom_point()+  geom_smooth(aes())
ggsave("ScatterLPM.png", width=9, height=6)

#scatterplot probit
data %>%
  ggplot(aes(fitted_probit, super)) +
  geom_point()+  geom_smooth(aes())
ggsave("ScatterProbit.png", width=9, height=6)
```