Tutorial 11: Generalised Linear Models I

For this lab, you need the package **AER** ‘Applied Econometrics with R’. You will find that most of the data sets for this tutorial can be found in this package.

I’m using Kubuntu and I found that the easiest way to install **AER**, in such a way that all the dependencies were installed at the same time, was to go to ‘Synaptic Package Manager’ and tell it to install `r-cran-AER`. This puts **AER** in the ‘System Library’, with all the dependencies.

1. The data set `SwissLabor` from **AER** considers the female labour force from a sample of 872 women from Swizerland. The response is a binary variable, `participation` has two possible values, indicating either yes or no.

```
> data(SwissLabor)
```

Try a probit regression `participation` against `income`, `education`, `age`, `age^2`. The following code gives `participation` against *all* the other variables, together with the square of the age variable.

```
> swiss_probit <- glm(participation~.+I(age^2), data = SwissLabor,
+ family=binomial(link="probit"))
> summary(swiss_probit)
```

Note the syntax and how the `glm` call works. What are your conclusions?

Now try to plot participation versus age.

```
> plot(participation~age, data=SwissLabor, ylevels=2:1)
```

You should see that it is not linear; it peaks at 40 years (4 units) and then decreases, hence the need for a quadratic term in age.

For the *probit* model, let Φ denote the $N(0,1)$ c.d.f. and ϕ the $N(0,1)$ density.

For a *continuous* regressor variable, the *effect* of x_{ij} is defined as:

$$\frac{\partial}{\partial x_{ij}}\mathbb{E}\left[Y_i|x_{i.}\right]$$

(namely the rate of increase in the expected response as the value of the regressor is increased). For a probit regression, we can exploit the fact that the link function is the c.d.f. of a normal to compute the effects for the various regressors.

$$\frac{\partial}{\partial x_{ij}}\Phi(x_{i.}^t\beta) = \phi(x_{i.}^t\beta) = \phi(x_{i.}^t\beta).\beta_j.$$

The *average effect* for the jth regressor variable is obtained by averaging this: $\frac{1}{n}\left(\sum_{i=1}^n \phi(x_{i.}^t\widehat{\beta})\right)\widehat{\beta}_j$.

```
> fav  <- mean(dnorm(predict(swiss_probit,type="link")))
> fav*coef(swiss_probit)
```

Interpret the output.

Another version of measuring effects considers $\phi(\overline{x}^t\beta)\beta_j$. This is straightforward as long as the regressors are continuous, but the model usually includes factors. It is then preferable to report average effects for all levels of the factors, averaging over only continuous variables. For SwissLabor, the only factor is foreign, which has two levels. Try

```
> av<-colMeans(SwissLabor[,-c(1,7)])
> av<-data.frame(rbind(swiss=av,foreign=av),
+ foreign=factor(c("no","yes")))
> av<-predict(swiss_probit,newdata=av,type="link")
> av<-dnorm(av)
> av["swiss"]*coef(swiss_probit)[-7]
```

This creates a data frame with two new sets of explanatory variables, where for both sets, we're using the average over all the continuous variables of the data set and we consider each level of the factor 'foreign' separately.

We then use the 'predict' command to compute the estimated probability for the participation variable for each level of the 'foreign' variable.

Understand the commands and interpret the output.

In contrast to linear regression, there is no commonly accepted version of R^2 for generalised linear models. Let $l(\widehat{\beta})$ and $l(\overline{y})$ denote the log-likelihoods for the fitted model and model only containing a constant term respectively. The *McFadden pseudo-R^2* is:

$$R^2 = 1 - \frac{l(\widehat{\beta})}{l(\overline{y})}.$$

Note that, for a Gaussian,

$$\log L(y_1, \ldots, y_n) = -\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(y_j - \mu_j)^2$$

so that even if the model fits perfectly, we get

$$R^2 = 1 - \frac{\frac{1}{2}\log(2\pi) + \log s}{\frac{1}{2}\log(2\pi) + \log s + \frac{1}{2ns^2}\sum_{j=1}^{n}(y_j - \overline{y})^2}$$

where s is the estimate of σ, so that the maximum possible value will be substantially smaller than 1.

Try:

```
> swiss_probit0<-update(swiss_probit,formula=.~1)
> 1 - as.vector(logLik(swiss_probit)/logLik(swiss_probit0))
```

`swiss_probit0` is simply the model that does not include any of the explanatory variables. We are therefore computing a McFadden pseudo R^2 value.

This value does indicate that the modelling is beneficial, although the interpretation of a McFadden pseudo R^2 is not precise.

Understand the commands.

Now consider predicting the class of `participation` from the regression. This may be done simply by:

```
> table(true=SwissLabor$participation,
+ pred=round(fitted(swiss_probit)))
```

'round' takes a value to its nearest integer value. The cut-off value here is 0.5.

The *deviance* is calculated by:

```
> deviance(swiss_probit)
```

For residual-based diagnostics, a `residuals()` method for `glm` objects is available. It provides various types of residuals, the most prominent of which are the deviance and the Pearson residuals. The deviance residuals are defined as the signed contributions to the overall deviance of the model and are computed by default in R. The latter are the raw residuals $y_i - \widehat{\mu}_i$ scaled by the standard error (often called standardised residuals in econometrics) and are available by setting `type = "Pearson"`

```
> sum(residuals(swiss_probit,type="pearson")^2)
[1] 866.5145
> sum(residuals(swiss_probit,type="deviance")^2)
[1] 1017.155
```

2. For the `SwissLabor` data, plot `participation` versus `education`. Does this suggest a non-linear effect of `education`? Fit a model which uses `education^2` in addition to the other regressors. Does the new model result in an improvement?

3. Consider the `PSID1976` data in **AER**.

   (a) Find a probit model for labour force participation using the regressors age, age squared, family income, education, and a factor indicating the presence of children.

   (b) Re-estimate the model assuming that different equations apply to women without children.

   (c) Perform a likelihood ratio test to check whether the more general model is needed.

   Notes You can add a yes / no column indicating whether or not there are children as follows:

```
PSID1976$kids <- with(PSID1976, factor((youngkids + oldkids) > 0,
+                                         levels = c(FALSE, TRUE),
labels = c("no", "yes")))
```

   For part (b), the analysis assuming that dierent equations apply to women with and without children, we can do it as follows:

```
new <-glm(participation~kids/(age + I(age^2) + fincome+
education),data=PSID1976, family=binomial(link="probit"))
```

4. Consider the beetle mortality data, and try to fit logit, probit and log-log models, of the form $\eta = \alpha + z\beta$, where z is the log-dose of the pesticide. Which model fits best?

log dose	no. beetles	no. killed
1.691	59	6
1.724	60	13
1.755	62	18
1.784	56	28
1.811	63	52
1.837	59	53
1.861	62	61
1.884	60	60

5. We first consider a situation where the maximum likelihood estimator for the logistic model does not exist. Consider the data set `MurderRates` from **AER**.

```
> data("MurderRates")
> murder_logit  <- glm(I(executions > 0)~ time+income+
+ noncauc+lfp+southern, data=MurderRates,
+ family=binomial)
```

Note the warning message.

```
> coeftest(murder_logit)
```

Note the suspiciously large standard error for `southernyes`.

We therefore suspect that numerical problems were encountered. It is therefore advisable to modify the default settings of the IWLS algorithm. The relevant argument for `glm()` is `control` which takes a list consisting of the entries `epsilon`, the convergence tolerance epsilon, `maxit`, the maximum number of IWLS iterations, and `trace`, the latter indicating if intermediate output is required for each iteration. Simultaneously decreasing the epsilon and increasing the maximum number of iterations yields:

```
> murder_logit2  <- glm(I(executions > 0)~time+income+
+ noncauc+lfp+southern, data=MurderRates,
+ family=binomial, control=list(epsilon=1e-15,maxit=50,trace=FALSE))
```

Note that the warning does not go away.

```
> coeftest (murder_logit2)
```

Note that the standard error for `southernyes` is even worse. The problem here is that the max. likelihood does not have a maximum in the interior.

```
> table(I(MurderRates$executions>0),MurderRates$southern)
```

Note that all 15 southern states executed convicted murderers during the period under consideration as well as 20 of the remaining states. The variable `southern` alone contains quite a lot of information on the dependent variable.

If *southern* were excluded in the analysis, the warning messages would go away, but the predictions would be worse.