

INTERMEDIATE SOCIAL STATISTICS

Workshop 1

Duch and Neuner

Topics: Cronbach's Alpha, factor analysis, binary logistic regression, predicted probabilities.

Data sets: 'ESS_measurement.dta' and 'ESS_binary' from the European Social Survey 2008 (4th round); 'iss_wave_2_week2.dta.' from the Comparative Campaign Analysis Project

Readings: Long, J.S. and Freese, J. (2006) *Regression models for categorical dependent variables using Stata*. Stata Press. Chapter 4 & Chapter 9.

INTRODUCTION

This week, we will first use exploratory factor analysis to reduce a set of variables to a smaller number of underlying constructs. Then we will move on to interpreting simple probability models (binary logit and probit).

DATASETS

Copy the following datasets from the shared drive (P:\Public\Labs_shared\Intermediate Social Statistics 2012) and save them in your own account: **ESS_measurement.dta**; **ESS_binary.dta**; **iss_wave_2_week2.dta**.

MEASUREMENT: RELIABILITY

Open **ESS_measurement.dta** and produce a list of all variables in the dataset:

describe

There are seven variables measuring trust in political institutions on a scale from 0 (no trust at all) to 10 (complete trust): *trstprl trstlgl trstplc trstplt trstprt trstep trstun*. We can give this list of variables a shorthand name using either a 'local' or a 'global' macro. Global macros are denoted with a \$ sign in front, while local macros are included in ``macroname'`. Whenever Stata encounters the macro, it substitutes its contents. We'll store a list of variables in a global macro:

```
global trust trstprl trstlgl trstplc trstplt trstprt trstep trstun  
describe $trust
```

To study the association between these seven variables in the pooled dataset (i.e. all 28 countries combined), first produce a correlation matrix with p-values reported. Use the product of the population and the design weight (*weight*) in this analysis, which is used whenever two or more country samples are combined. The design weight (*dweight*) corrects for the unequal selection probability of persons within households and is used for within country analyses. The population weight (*pweight*) corrects for unequal population sizes of each country, given that the sample sizes are quite similar in the ESS and, therefore, individuals from smaller countries are over-represented.

```
pwcorr $trust [aw=weight], sig
```

To check whether these seven variables can produce a reliable scale, compute the Cronbach's alpha:

```
alpha $trust
```

Note that weights cannot be used with this command. The 7-item scale has a very high reliability of approximately 0.90. It is also possible to check what the reliability coefficient would be if each of the variables were removed in turn, using the option *item*:

```
alpha $trust, item
```

The output shows that the current 7-item scale has the highest reliability coefficient, but only marginally so. The lowest reliability would be achieved if the variable *trstplt* 'trust in politicians' were removed: 0.8735 compared with the current 0.8979.

The command *alpha* can also construct a scale from the list of items and store it as a new variable. The option is *generate*. The resulting scale is simply the mean of the items for each observation. If *std* is used in conjunction with *generate*, the mean of standardized items (mean 0, variance 1) is created.

```
alpha $trust, gen(trust)
```

```
alpha $trust, gen(trust_std) std
```

EXERCISE 1

Calculate the Cronbach's Alpha for nine variables measuring attitudes towards social benefits/services. Which variable contributes the most to the scale? Create an unstandardised and a standardised scale as new variables.

MEASUREMENT: FACTOR ANALYSIS

The seven variables measuring trust in the political system may be capturing one or more underlying (latent) concepts. Exploratory factor analysis can help discover these latent constructs. There are several methods of carrying out factor analysis in STATA (principal factor; principle component factor; iterated principal factor; maximum likelihood factor), but they often produce similar results for data reduction purposes. To find out more about factor analysis in Stata:

```
help factor
```

To carry out principal-component factor analysis:

```
factor $trust [aw=weight], pcf
```

It appears that we have a two-factor solution, since only two factors have eigenvalues greater than one. The eigenvalue is the variance of the factor. 'Difference' is the difference between two successive eigenvalues. 'Proportion' is the proportion of total variance explained by the factor. Factor loadings show the strength of the relationship between each variable and each factor, while uniqueness is the proportion of the variance of a variable not related to any factor.

To get a better idea of which variables are more strongly related to which factor, the factors can be rotated until they are orthogonal (independent):

```
rotate
```

According to the rotated factor loadings, the variables measuring trust in the European Parliament (*trstep*) and trust in the United Nations (*trstun*) are associated with Factor 2 to a greater degree than with Factor 1, while the rest of the items have higher loadings on Factor 1.

Thus, Factor 1 may be described as trust in national institutions, while Factor 2 as trust in international institutions.

Since each variable can be modeled as a linear function of latent factors, it is possible to produce factor scores as regression coefficients, using the command `predict`:

```
predict trust1 trust2
```

Trust1 is the score for factor 1 and trust2 for factor 2. Since these scores have been produced after orthogonal rotation, the factor scores are not correlated:

```
pwcorr trust1 trust2, sig
```

Factor scores can be used as predictors in other models. For example, let's build a simple linear model of attitudes towards EU integration (0-10 scale) in Great Britain, using factor scores from the political trust variables as predictors, along with respondent's age (*agea*) and placement on the left/right (*lrscale*).

```
tab euftf
```

```
factor $trust if centry=="GB" [aw=dweight], pcf
```

There is only one factor:

```
predict trustGB
```

```
regress euftf trustGB agea lrscale if centry=="GB" [pw=dweight]
```

The factor score has a positive and significant effect on the opinion about EU integration: for a one-unit increase in the factor score, controlling for age and the left-right placement, the opinion about EU integration goes up by 0.76 points, on average.

However, if instead of the factor score the original seven trust variables are included, only two of them have a significant effect on the dependent variable:

```
regress euftf $trust agea lrscale if centry=="GB" [pw=dweight]
```

EXERCISE 2

Use factor analysis to study the underlying concepts behind nine variables measuring attitudes towards social benefits/services in the pooled dataset. How many factors are these variables related to? Which variables are more highly associated with which factors? How would you label these factors?

BINARY LOGIT AND PROBIT MODELS

Open **iss_wave_2_week2.dta** and familiarize yourself with the variables in the dataset. Use binary logistic regression to predict vote preference for the incumbent (Labour party) prior to the 2009 British election, controlling for the following predictors: evaluation of the retrospective economic situation (*retnat_2*), trade union membership (*union_1*), placement on the left-right scale (*LR_Self_2*), education (*educate*) and income (*income_2*), age (*age*) and gender (*female*).

```
codebook voteinc_2 retnat_2 LR_Self_2 educate age female income_2
union_1
```

```
logit voteinc_2 retnat_2 LR_Self_2 educate age female income_2 union_1
```

Recode the variable *income_2* into a dummy “rich and poor”:

```
recode income_2 (1/5=0 "poor") (5/15=1 "rich"), into (rich_2)
```

```
logit voteinc_2 retnat_2 LR_Self_2 educate age female income_2 union_1
```

Now generate the predicted probability of incumbent vote for each respondent in the dataset using the post-estimation command `predict`, as a new variable *prob_logit*:

```
predict prob_logit
```

To study the predicted probabilities for different combinations of respondents’ characteristics, use a post-estimation command `margins`. To obtain the average predicted probability for the whole estimation sample:

```
margins
```

This is exactly the mean of the newly created variable *prob_logit*. You can check this with

```
summarize prob
```

Now estimate the same model with probit and store the predicted probabilities in a new variable *prob_probit*. Compare the probabilities estimated with logit and probit:

```
probit voteinc_2 retnat_2 LR_Self_2 educate age female income_2  
union_1
```

```
predict prob_probit
```

```
corr prob_probit prob_logit
```

```
list prob_probit prob_logit in 1/20
```

The estimated probabilities from logit and probit are very similar, so the rest of the examples will use logit to predict probabilities, but the same post-estimation commands apply to probit.

To predict the probability of incumbent vote at different levels of left-right self identification:

```
logit voteinc_2 retnat_2 LR_Self_2 educate age female income_2 union_1  
margins, atmeans at(LR_Self_2 =(1(1)10))
```

To predict the probability of incumbent vote at different levels of economic perceptions, holding other predictors at their means:

```
margins, atmeans at(retnat_2=(1(1)5))
```

There are several ways to create and plot predicted probabilities, one of which is with the `prgen` post-estimation command. `Prgen` generates predicted probabilities over the range of one variable, holding other predictors constant. For example, to produce the probabilities for the three categories of economic evaluation over 10 values of left-right self placement, holding other predictors at their means:

```
logit voteinc_2 retnat_2b LR_Self_2 educate age female rich_2 union_1  
  
prgen LR_Self_2, from(1) to (10) gen(try) x(retnat_2b=0) n(10) ci  
prgen LR_Self_2, from(1) to (10) gen(retnat1) x(retnat_2b=1) n(10) ci  
prgen LR_Self_2, from(1) to (10) gen(retnat2) x(retnat_2b=2) n(10) ci  
prgen LR_Self_2, from(1) to (10) gen(retnat3) x(retnat_2b=3) n(10) ci
```

A number of variables have been generated automatically. `Retnat1p1` contains the estimated probabilities of voting Labour over 10 levels of the left-right scale if `retnat=1`. Similarly, `retnat2p1` refers to `retnat=2` and `retnat3p1` refers to `retnat=3`. `Retnat1x` contains 10 values of the left-right self placement scale. Give meaningful labels to the probability variables and tabulate them:

```
lab var retnat1p1 "economy will get better"  
lab var retnat2p1 "economy will stay the same"  
lab var retnat3p1 "economy will get worse"  
tab1 retnat1p1 retnat2p1 retnat3p1 retnat1x
```

There are several ways to produce graphs in Stata, `graph twoway connect` is only one of them:

```
graph twoway connect retnat1p1 retnat2p1 retnat3p1 retnat1x
```

The option `ci` adds confidence intervals to the graphs.

To graph the predicted probabilities with confidence intervals, use a combination of graphing commands `twoway rarea` (to produce shaded confidence intervals) and `twoway line` (to produce a line for the point estimates). The command will take up more than one line of code, so it needs to be written in the do-file.

```
graph twoway (rarea retnat1plub retnat1p1lb retnat1x) ///  
(line retnat1p1 retnat1x) ///  
(rarea retnat2plub retnat2p1lb retnat1x) ///  
(line retnat2p1 retnat1x) ///  
(rarea retnat3plub retnat3p1lb retnat1x) ///  
(line retnat3p1 retnat1x), scheme(s1mono)
```

Unfortunately, `prgen` does not work when the model includes interaction terms. If your model has polynomial or interaction terms, you would need to use `margins` or `predict` to generate the probabilities and the confidence intervals around them.

Create a square of age and add it to the earlier logit model using a `c.` operator to denote a continuous variable and `#` to denote an interaction:

```
logit voteinc_2 retnat_2b LR_Self_2 educate c.age#c.age female rich_2
union_1
```

Consider a hypothetical respondent who has an optimistic outlook on the economy (*retnat*=1), belongs to lower-middle class (*class_rec*=1), not a union member (*union*=0) and lives in a rural area (*urban*=0) in the north-east (*southwest*=0) and owns their home. What would be the predicted probabilities of voting for the Labour party for such respondents if they have different placements on the left-right scale?

```
logit voteinc_2 retnat_2b LR_Self_2 educate c.age#c.age female rich_2
union_1
```

```
margins,at(retnat_2b=1 union_1=0 female=1 rich_2=1 LR_Self_2=(1(1)10))
```

Copy the predicted probabilities and the 95% confidence bounds into Excel or another Stata window and plot them against 10 values of the left-right scale.

EXERCISE 3

Open a small file called **ESS_binary.dta**. Familiarize yourself with the variables. Build a simple logistic regression model of 0/1 employment status (*work*) as a function of a continuous variable age (*agea*) and a binary 0/1 variable education (*edu*). Run the model for a country of your choice (but NOT Britain), using *dweight* as the probability weight [*pw=dweight*]. You do not need to use weights with the post-estimation commands following the regression.

Using *margins*, calculate the predicted probability of being in paid work for the two categories of education, first holding age at its mean and then at 40 years old. Finally, calculate the predicted probabilities separately by education when age increases from 20 to 70 in increments of 10 years.

Using *prgen*, plot the predicted probabilities of working separately by education over 10 year increments of age, with 95% confidence intervals.