Chapter 1

Truncated Regression

1.1 Truncated Linear Regression

The linear regression model we consider here has the form

$$Y_i = \sum_{j=1}^{m} a_j x_{ij} + \epsilon_i$$

where the Y_i for $i = 1, \ldots, n$ are the n observations and the a_j are m parameters to be estimated. The ϵ_i are assumed to be normally distributed random variables with mean 0 and variance v

Let $r_i = Y_i - \sum_{j=1}^{m} a_j x_{ij}$. The log-likelihood function for the standard regression model is give by

$$-.5n \log(v) - \sum_{i=1}^{n} \frac{r_i^2}{2v}$$

Now assume that we only consider the Y_i for $Y_i/ge0$ i.e. the left truncated situation. The probability that $Y_i \geq 0$ is equal to the probability that $\epsilon_I > -\sum_{j=1}^{m} a_j x_{ij}$. This is equal to $1 - \Phi(-\sum_{j=1}^{m} a_j x_{ij}/v)$ where

$$\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} \exp(-t^2/2)\, dt$$

For this truncated regression the log-likelihood function has the logarithm of this quantity subtracted from it so it becomes

$$-.5n \log(v) - \sum_{i=1}^{n} \frac{r_i^2}{2v} - \log(1 - \Phi(-\sum_{j=1}^{m} a_j x_{ij}/v))$$

If instead we consider the right truncated case where only $Y_i < 0$ are considered the log-likelihood function becomes

$$-.5n \log(v) - \sum_{i=1}^{n} \frac{r_i^2}{2v} - \log(\Phi(-\sum_{j=1}^{m} a_j x_{ij}/v))$$

To parameterize v we introduce a new parameter a satisfying the condition $v = a\hat{v}$ where $\hat{v} = \frac{1}{n}\sum_{i=1}^{n} r_i^2$ is the usual maximum liklihood estimate for v. This leads to more numerically stable behaviour. In terms of a the expression for the log-likelihood simplifies to

$$-.5n\log(a) - .5n\log(\hat{v}) - \frac{n}{2a} - \log(1 - \Phi(-\sum_{j=1}^{m} a_j x_{ij}/(a\hat{v})))$$

1.2 The AD Model Builder Truncated Regression Program

Here are the contents of the file `truncreg.tpl`.

```
DATA_SECTION
  init_int nobs
  init_int m
  init_int trunc_flag
  init_matrix data(1,nobs,1,m+1)
  vector Y(1,nobs)
  matrix X(1,nobs,1,m)
 LOC_CALCS
  Y=column(data,1);
  for (int i=1;i<=nobs;i++)
  {
    X(i)=data(i)(2,m+1).shift(1);
  }
PARAMETER_SECTION
  sdreport_number sigma
  number vhat
  init_bounded_number log_a(-5.0,5.0);
  sdreport_number a
  init_vector u(1,m)
  objective_function_value f
PROCEDURE_SECTION
  a=exp(log_a);
  dvar_vector pred=X*u;
  dvar_vector res=Y-pred;
  dvariable r2=norm2(res);
  vhat=r2/nobs;
  dvariable v=a*vhat;
  sigma=sqrt(v);

  dvar_vector spred=pred/sigma;
  f=0.0;
  switch (trunc_flag)
  {
  case -1:  // left_truncated
    {
      for (int i=1;i<=nobs;i++)
      {
        f+=log(1.00001-cumd_norm(-spred(i)));
      }
    }
    break;
  case 1:   // right truncated
```

```
    {
      for (int i=1;i<=nobs;i++)
      {
        f+=log(0.99999*cumd_norm(-spred(i)));
      }
    }
    break;
  case 0:   // no truncation
    break;
  default:
    cerr << "Illegal value for truncation flag" << endl;
    ad_exit(1);
  }
  f+=0.5*nobs*log(v)+0.5*r2/v;


REPORT_SECTION
  report << "#u " << endl << u << endl;
  report << "#sigma " << endl << sigma << endl;
  report << "#a " << endl << a << endl;
  report << "#vhat " << endl << vhat << endl;
  report << "#shat " << endl << sqrt(vhat) << endl;
}
```