What is Tobit Analysis and how can it be applied in SAS Data Analysis?

Authored by

stats writer

June 29, 2024

Tobit Analysis is a statistical method used for analyzing data that contains both continuous and censored (limited or truncated) values. It is commonly used in econometrics and other disciplines to analyze data that has a significant number of observations at either the upper or lower limit of the measurement scale. In SAS Data Analysis, Tobit Analysis can be applied to model and estimate the relationship between variables, taking into account the censoring of data. This allows for a more accurate and comprehensive analysis of the data, providing insights into the underlying relationship between variables. It is particularly useful in situations where traditional regression methods may not be appropriate due to the presence of censored data. Overall, Tobit Analysis in SAS enables researchers and analysts to make more informed decisions by accounting for the limitations in the data.

Tobit Analysis | SAS Data Analysis Examples

Note: This page uses SAS 9.2.

The tobit model, also called a censored regression model, is designed to estimate linear relationships between variables when there is either left- or right-censoring in the dependent variable (also known as censoring from below and above, respectively). Censoring from above takes place when cases with a value at or above some threshold, all take on the value of that threshold, so that the true value might be equal to the threshold, but it might also be higher. In the case of censoring from below, values those that

fall at or below some threshold are censored.

Please note: The purpose of this page is to show how to use various data analysis commands. It does not cover all aspects of the research process which researchers are expected to do. In particular, it does not cover data cleaning and checking, verification of assumptions, model diagnostics and potential follow-up analyses.

Examples of tobit analysis

Example 1.

In the 1980s there was a federal law restricting speedometer readings to no more than 85 mph. So if you wanted to try and predict a vehicle's top-speed from a combination of horse-power and engine size, you would get a reading no higher than 85, regardless of how fast the vehicle was really traveling. This is a classic case of right-censoring (censoring from above) of the data. The only thing we are certain of is that those vehicles were traveling at least 85 mph.

Example 2. A research project is studying the level of lead in home drinking water as a function of the age of a house and family income. The water testing kit cannot detect lead concentrations below 5 parts per billion (ppb). The EPA considers levels above 15 ppb to be dangerous. These data are an example of left-censoring (censoring from below).

Example 3. Consider the situation in which we have a measure of academic aptitude (scaled 200-800) which we want to model using reading and math test scores, as well as, the type of program the student is enrolled in (academic, general, or vocational). The problem here is that students who answer all questions on the academic aptitude test correctly receive a score of 800, even though it is likely that these students are not "truly" equal in aptitude. The same is true of students who answer all of the questions incorrectly. All such

students would have a score of
200, although they may not all be of equal aptitude.

Description of the data

Let's pursue Example 3 from above. We have a hypothetical data file,
tobit.sas7bdat
with 200 observations with format
defined below. The academic aptitude variable is apt, the reading and math test scores are read and math respectively.
The variable prog is the type of program the student is in, it is a
categorical (nominal) variable that takes on three values, academic (prog
= 1), general (prog = 2), and vocational (prog = 3). Variable
prog comes with a format provided below.

```
proc format;
value proga 1="academic"
2="general"
3="vocational";
```

```
run;

data tobit;
set tobit;
format prog proga.;
run;
```

Let's look at the data. Note that in this dataset, the lowest value of apt
is 352. Note that no students received a score of 200 (i.e., the lowest score possible),
meaning that even though censoring from below was possible, it does not occur in
the dataset.

```
options nolabel nocenter nodate formchar = '|----|+|---+=|-/*';
proc means data = tobit maxdec=2 nonobs;
class prog;
vars apt read math;
run;
prog Variable N Mean Std Dev Minimum Maximum
----------
```

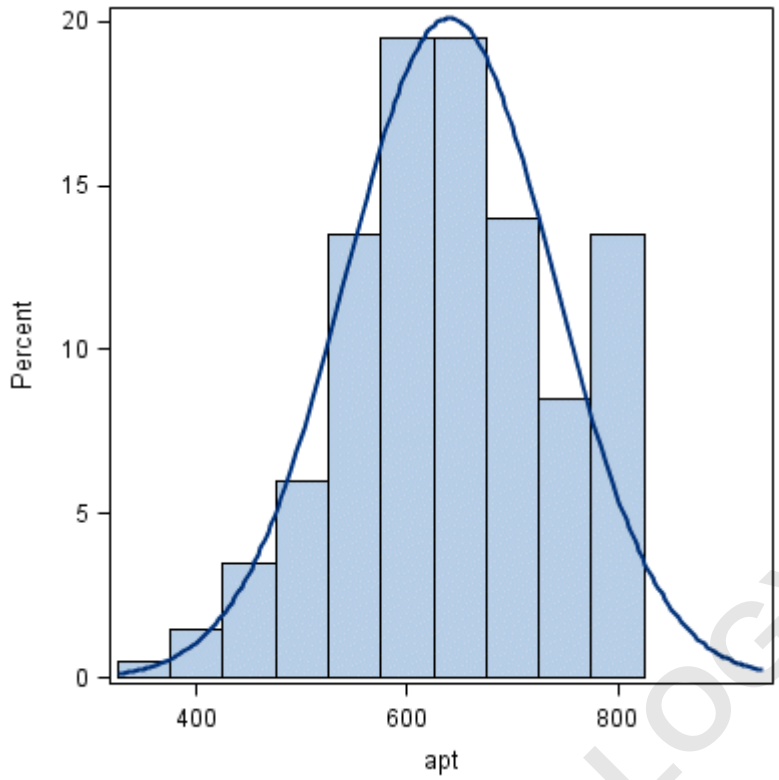
---

academic apt 45 639.02 78.63 454.00 800.00
read 45 49.76 9.23 28.00 68.00
math 45 50.02 7.44 35.00 63.00

general apt 105 677.76 88.21 462.00 800.00
read 105 56.16 9.59 34.00 76.00
math 105 56.73 8.73 38.00 75.00

vocational apt 50 561.72 92.76 352.00 800.00
read 50 46.20 8.91 31.00 68.00
math 50 46.42 7.95 33.00 75.00

```
ods graphics / reset=all imagename='dens'
imagefmt=png
width=4in height=4in border=off;
proc sgplot data = tobit noautolegend;
histogram apt;
density apt /type = normal lineattrs=(color=blue);
```

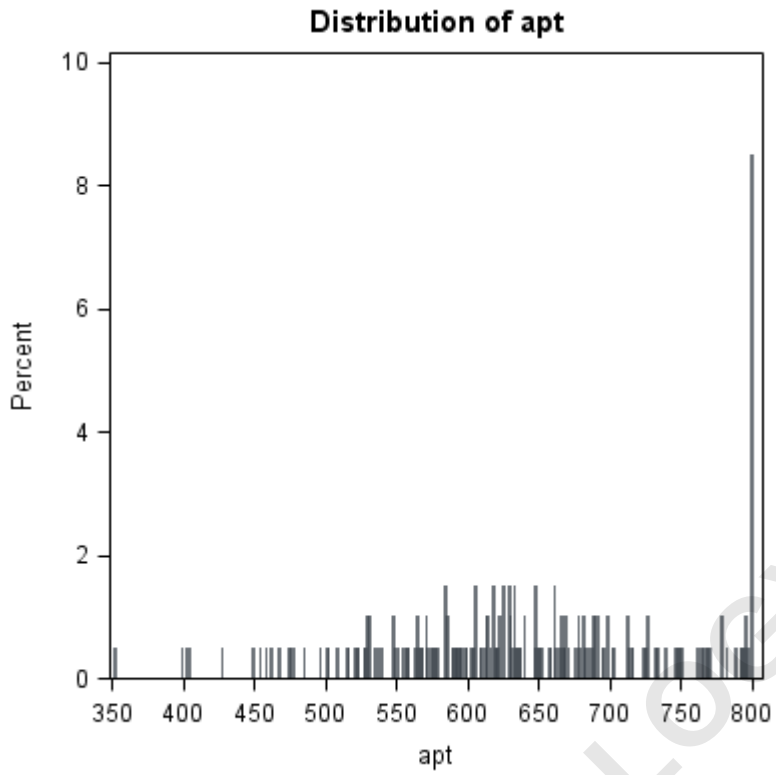


run;

Looking at the above histogram showing the distribution of apt, we can see the censoring in the data, that is, there are far more cases with scores of 775 to 800 (i.e., the final bin) than one would expect looking at the rest of the distribution. Below is an alternative histogram that further highlights the excess of cases where apt=800. In the histogram below, midpoints option is used to produce a histogram where each unique value of apt has its own bar by specifying that there should be bins from

350 (the minimum of apt is 352) and a max of 800 in units of 1. Because apt is continuous, most values of apt are
unique in the dataset, although close to the center of the distribution there are a few
values of apt that have two or three cases. The spike on the far right of the histogram is
the bar for cases where apt=800, the height of this bar relative to all the others clearly
shows the excess number of cases with this value.

```
ods graphics / reset=all imagename='hist' imagefmt=png
width=4in height=4in border=off;
proc univariate data=tobit noprint;
histogram apt / midpoints=350 to 800 by 1 normal ;
```



run;

Next we'll explore the bivariate relationships in our dataset. We make use of the scatter matrix plot created by
proc corr via ods graphics on option.

```
ods graphics / reset=all imagename='mat' imagefmt=png
width=4in height=4in border=off;
ods graphics on;
proc corr data = tobit nosimple;
var read math apt;
```

run;
ods graphics off;
Pearson Correlation Coefficients, N = 200
Prob > |r| under H0: Rho=0

read math apt

read 1.00000 0.66228 0.64512

Note the collection of cases at the top of the bottom row of the scatter plots are due to the censoring in the distribution of
apt.

Analysis methods you might consider

Below is a list of some analysis methods you may have encountered.
Some of the methods listed are quite reasonable while others have either
fallen out of favor or have limitations.

Tobit analysis

Below we use proc qlim to fit a tobit regression model. Note that proc qlim is part of the ETS module for SAS. It

is also possible to fit a
tobit model using proc lifereg (part of the STAT module), although the syntax to do so is somewhat different from the example shown below.
The class statement identifies prog as a categorical variable, and the
model statement specifies that apt should be modeled using read,
math,
and prog. The endogenous statement specifies that the outcome variable
apt is censored, with an upper bound of 800 (i.e., ub=800).

proc qlim data = tobit ;
class prog;
model apt = read math prog;
endogenous apt ~ censored (ub=800);
run;

Parameter Estimates

Standard Approx

Parameter DF Estimate Error t Value Pr > |t|

Intercept 1 163.422155 30.408580 5.37

Under the heading Parameter Estimates we see the coefficients, their standard errors, the t-statistics, and associated p-values. The coefficients for read and math are statistically significant, as are the terms for prog="academic" and prog="general" (with prog="vocational" as the reference category). Tobit regression coefficients are interpreted in the similiar manner to OLS regression coefficients; however, the linear effect is on the uncensored latent variable, not the observed outcome. See McDonald and Moffitt, (1980) for more details.

We can include a test of the overall effect of prog, by testing whether the coefficients for prog="academic" and prog="general" are simultaneously equal to 0. To do this we add a test statement to the proc qlim code. To figure out how SAS names the dummy variables for a class variable, it

is usually a good idea to output the parameter estimates as a data set (in this example, we named it as t) and print it out to see how internally SAS names these variables. In our example, we see that SAS has appended the value label
to prog in naming the dummy variables for prog.

```
proc qlim data = tobit outest=t;
class prog;
model apt = read math prog;
endogenous apt ~ censored (ub=800);
run;
proc print data = t noobs;
run;
```

```
prog_ prog_ prog_
_NAME_ _TYPE_ _STATUS_ Intercept read math academic general vocational _Sigma

PARM 0 Converged 163.422 2.69794 5.91448 46.1439 33.4292 . 65.6767
STD 0 Converged 30.409 0.61881 0.70982 13.7242 12.9556 . 3.4814
```

```
proc qlim data = tobit ;
class prog;
```

```
model apt = read math prog;
endogenous apt ~ censored (ub=800);
test 'prog' prog_academic=0,
prog_general =0;
```

Test Results				
Test	Type	Statistic	Pr > ChiSq	Label
'prog'	Wald	11.96	0.0025	prog_academic = 0 , prog_general = 0

```
run;
```

Because the model is the same, the output for this syntax is the same as
before, except for the addition section shown showing the results of the
test statement. Under Test Results, we see
that the overall effect of prog is statistically significant.

We can also test additional hypotheses about the differences in the coefficients
for different levels of prog. Below we test that the coefficient for prog="academic" is equal
to the coefficient for prog="general".

```
proc qlim data = tobit;
class prog;
```

```
model apt = read math prog;
endogenous apt ~ censored (ub=800);
test 'academic vs. general' prog_academic -
prog_general = 0;
```

Test Results				
Test	Type	Statistic	Pr > ChiSq	Label
'academic vs. general'	Wald	1.05	0.3054	prog_academic - prog_general = 0

```
run;
```

We may also wish to evaluate how well our model fits. This can be particularly useful when comparing competing models. One method of assessing model fit is to compare the predicted values based on the tobit model to the observed values in the dataset. Below we use proc qlim to generate predicted values along with the data via the output statement. Then proc corr is used to estimate the correlation between the predicted and observed values of apt. The output from proc corr gives the correlation between the predicted and observed values of apt, which is 0.78094. If we square this value, we get the squared multiple correlation, this indicates that the predicted values

share about 61% (0.78094^2 = .6099) of their variance with the observed values
of apt.

```
proc qlim data=tobit ;
model apt = read math prog;
endogenous apt ~ censored (ub=800);
output out = temp1 predicted;
run;

proc corr data = temp1 nosimple;
var apt p_apt;
run;
```

Pearson Correlation Coefficients, N = 200
Prob > |r| under H0: Rho=0

	apt	P_apt
apt	1.00000	0.78094

See also

McDonald, J. F. and Moffitt, R. A. 1980. The Uses of Tobit Analysis. The Review of Economics and Statistics

Vol 62(2): 318-321.