What is Interval Regression and how can it be applied in SAS data analysis?

Authored by

stats writer

June 29, 2024

Interval Regression is a statistical technique used to analyze data where the dependent variable is measured in intervals or ranges rather than exact values. It is commonly used in situations where the outcome of interest cannot be precisely measured, such as income or time-to-event data.

In SAS data analysis, Interval Regression can be applied to model and predict the relationship between a continuous independent variable and an interval dependent variable. This allows researchers to determine the impact of the independent variable on the outcome within a specific range or interval. Additionally, SAS provides various tools and procedures to estimate parameters and interpret results from Interval Regression models, making it a powerful tool for data analysis and decision-making. Overall, Interval Regression in SAS allows for a more accurate and comprehensive analysis of data with interval outcomes, providing valuable insights for various fields such as economics, finance, and healthcare.

Interval Regression | SAS Data Analysis Examples

Version info: Code for this page was tested in SAS 9.3.

Interval regression is used to model outcomes that have interval censoring.
In other words, you know the ordered category into which each observation falls,
but you do not know the exact value of the observation. Interval
regression is a generalization of censored regression.

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 or
potential follow-up analyses.

Examples of interval regression

Example 1. We wish to model annual income using years of education and marital status. However,
we do not have access to the precise values for income. Rather, we only have data on the income ranges: <$15,000, $15,000-$25,000, $25,000-$50,000, $50,000-$75,000, $75,000-$100,000, and >$100,000. Note
that the extreme values of the categories on either end of the range are either left-censored
or right-censored. The other categories are interval censored, that is, each interval is both
left- and right-censored. Analyses of this type require a generalization of censored regression
known as interval regression.

Example 2. We wish to predict GPA from teacher ratings of effort and from reading and writing test
scores. The measure of GPA is a self-report response to the following item:

Select the category that best represents your overall GPA.
less than 2.0
2.0 to 2.5
2.5 to 3.0
3.0 to 3.4
3.4 to 3.8
3.8 to 3.9
4.0 or greater

Again, we have a situation with both interval censoring and left- and right-censoring. We do not know the exact value of GPA for each student; we only know the interval in which their
GPA falls.

Example 3. We wish to predict GPA from teacher ratings of effort, writing
test scores and the type of program in which the student was enrolled
(vocational, general or academic). The measure of GPA is a self-report response to the following item:

Select the category that best represents your overall

GPA.

0.0 to 2.0

2.0 to 2.5

2.5 to 3.0

3.0 to 3.4

3.4 to 3.8

3.8 to 4.0

This is a slight variation of Example 2. In this example, there is only interval censoring.

Description of the data

Let's pursue Example 3 from above.

We have a hypothetical data file, intregex,
with 30 observations. The GPA score is represented by two values, the lower interval score (lgpa) and the upper interval score (ugpa). The writing test scores, the teacher rating
and the type of program (a nominal variable which has three levels) are write, rating and type, respectively.

Let's look at the data. It is always a good idea to start with

descriptive statistics.

proc print data = mylib.intreg_data;
var lgpa ugpa;
run;
Obs lgpa ugpa

1 2.50000 3.00000
2 3.40000 3.80000
3 2.50000 3.00000
4 0.00000 2.00000
5 3.00000 3.40000
6 3.40000 3.80000
7 3.80000 4.00000
8 2.00000 2.50000
9 3.00000 3.40000
10 3.40000 3.80000
11 2.00000 2.50000
12 2.00000 2.50000
13 2.00000 2.50000
14 2.50000 3.00000
15 2.50000 3.00000
16 2.50000 3.00000
17 3.40000 3.80000

18 2.50000 3.00000
19 2.00000 2.50000
20 3.00000 3.40000
21 3.40000 3.80000
22 3.80000 4.00000
23 2.00000 2.50000
24 3.00000 3.40000
25 3.40000 3.80000
26 2.00000 2.50000
27 2.00000 2.50000
28 2.00000 2.50000
29 2.50000 3.00000
30 2.50000 3.00000

Note that there are two GPA responses for each observation, lgpa for the lower end of the interval and ugpa for the upper end.

```
proc means data = mylib.intreg_data;
var lgpa ugpa write rating;
run;
```

The MEANS Procedure

Variable N Mean Std Dev Minimum Maximum

---

lgpa 30 2.6000000 0.7754865 0 3.8000000

ugpa 30 3.0966666 0.5708332 2.0000000 4.0000000

write 30 113.8333333 49.9427834 50.0000000 205.0000000

rating 30 57.5333333 8.3034406 48.0000000 72.0000000

---

proc sort data = mylib.intreg_data;

by type;

run;

proc means data = mylib.intreg_data;

by type;

var lgpa ugpa;

run;

---------- type=1 ----------

The MEANS Procedure

Variable N Mean Std Dev Minimum Maximum

---

lgpa 8 1.7500000 0.7071068 0 2.0000000

ugpa 8 2.4375000 0.1767767 2.0000000 2.5000000

---------- type=2 ----------

Variable N Mean Std Dev Minimum Maximum

---

lgpa 10 2.7800000 0.3852849 2.5000000 3.4000001
ugpa 10 3.2400000 0.3373096 3.0000000 3.8000000

---------- type=3 ----------

Variable N Mean Std Dev Minimum Maximum

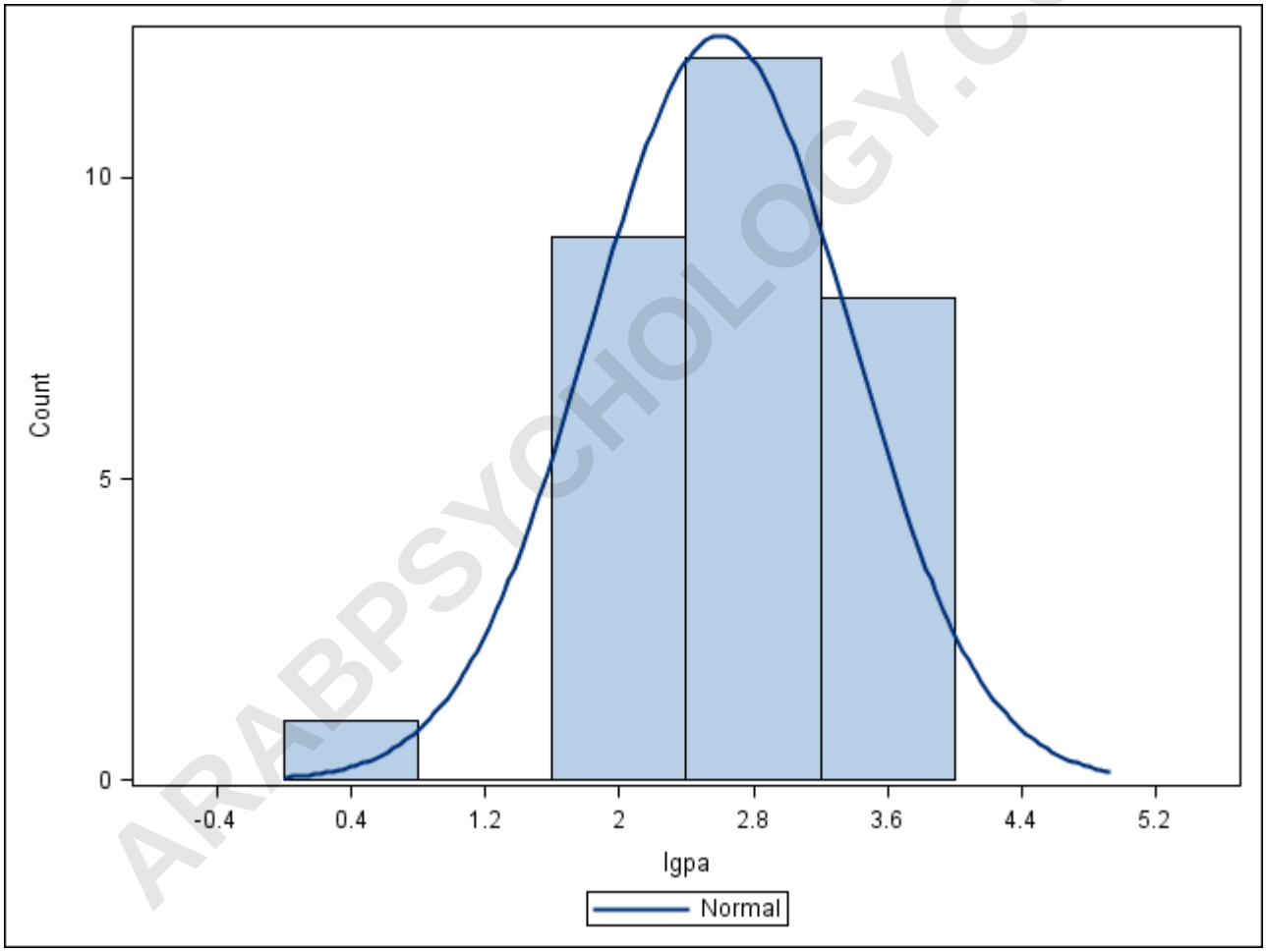
---

lgpa 12 3.0166667 0.6336522 2.0000000 3.8000000
ugpa 12 3.4166666 0.5474458 2.5000000 4.0000000

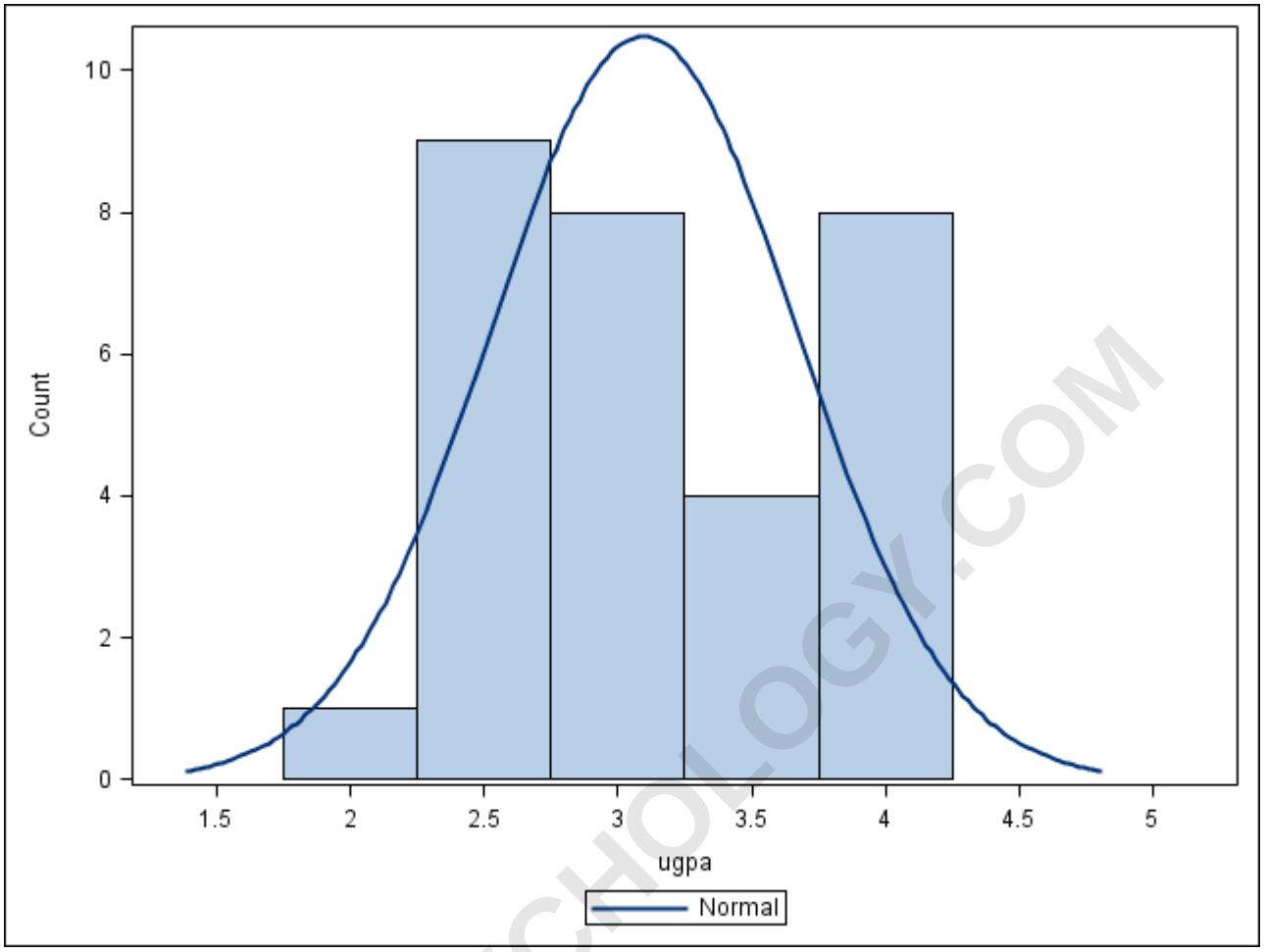
Graphing these data can be rather tricky. So just to get an idea of what the distribution of
GPA is, we will do separate histograms for lgpa and ugpa. We

will also correlate the variables in the dataset.

```
proc sgplot data = mylib.intreg_data;
histogram lgpa / scale = count showbins;
density lgpa;
run;
```




```
proc sgplot data = mylib.intreg_data;
histogram ugpa / scale = count showbins;
density ugpa;
run;
```



proc corr data = mylib.intreg_data;

var lgpa ugpa write rating;

run;

The CORR Procedure

4 Variables: lgpa ugpa write rating

Simple Statistics

Variable N Mean Std Dev Sum Minimum Maximum

lgpa 30 2.60000 0.77549 78.00000 0 3.80000

ugpa 30 3.09667 0.57083 92.90000 2.00000 4.00000
write 30 113.83333 49.94278 3415 50.00000 205.00000
rating 30 57.53333 8.30344 1726 48.00000 72.00000

Pearson Correlation Coefficients, N = 30
Prob > |r| under H0: Rho=0

lgpa ugpa write rating

lgpa 1.00000 0.94878 0.62057 0.53551

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.

Interval regression analysis

We will use proc lifereg to run the interval regression analysis.
We list the variable type on the class statement. We enclose
both lgpa and ugpa in parentheses on the model statement

**before the equals sign to indicate that these variables are the outcome
variables. We list write, rating and type as the predictor variables. We use the d=normal option to specify the
distribution as normal.**

```
proc lifereg data = intreg_data;
class type;
model (lgpa ugpa) = write rating type / d=normal;
run;
```

The LIFEREG Procedure

Model Information

**Data Set MYLIB.INTREG_DATA intreg_data dataset written by Stat/Transfer
Ver. 10.1.1655.0406**

**Dependent Variable lgpa
Dependent Variable ugpa
Number of Observations 30
Noncensored Values 0
Right Censored Values 0
Left Censored Values 0**

Interval Censored Values 30

Number of Parameters 6

Name of Distribution Normal

Log Likelihood -33.12890521

Number of Observations Read 30

Number of Observations Used 30

Class Level Information

Name Levels Values

type 3 1 2 3

Fit Statistics

-2 Log Likelihood 66.258

AIC (smaller is better) 78.258

AICC (smaller is better) 81.910

BIC (smaller is better) 86.665

Algorithm converged.

Type III Analysis of Effects

Wald

Effect DF Chi-Square Pr > ChiSq

write 1 9.7541 0.0018
rating 1 2.1314 0.1443
type 2 18.7076 ChiSq

Intercept 1 1.8136 0.5011 0.8315 2.7957 13.10 0.0003
write 1 0.0053 0.0017 0.0020 0.0086 9.75 0.0018
rating 1 0.0133 0.0091 -0.0046 0.0312 2.13 0.1443
type 1 1 -0.7097 0.1668 -1.0367 -0.3827 18.10

The lifereg procedure does not compute an R2 or pseudo-R2.
You can compute a rough-and-ready measure of fit by calculating the R2
between the predicted and observed values.

```
proc lifereg data = mylib.intreg_data;
class type;
model (lgpa ugpa) = write rating type / d=normal;
output out = mylib.t xbeta=xb;
run;

ods output PearsonCorr=mylib.int_corr;
proc corr data = mylib.t nosimple;
```

```
var xb lgpa ugpa;
run;

data _null_;
set mylib.int_corr;
file print;
if variable = "lgpa" then do;
a = round((xb)**2, .0001);
put "The squared multiple correlation between lgpa and the predicted value is " a;
end;
if variable = "ugpa" then do;
b = round((xb)**2, .0001);
put "The squared multiple correlation between ugpa and the predicted value is " b;
end;
run;
```

The squared multiple correlation between lgpa and the predicted value is 0.6314
The squared multiple correlation between ugpa and the predicted value is 0.7107

Things to consider

See also

References