What is the process of truncated regression and how is it used in R for data analysis?

Authored by

stats writer

June 29, 2024

Truncated regression is a statistical technique used to analyze data that is censored or truncated, meaning that some values are missing or limited. This method is commonly used when the data follows a specific pattern or distribution, and the values outside of that pattern are not included in the analysis. In R, truncated regression is implemented through the use of the "truncreg" function, which allows for the estimation of regression coefficients and other statistical measures for truncated data. This process involves identifying the truncation point, fitting a truncated regression model, and evaluating the results to understand the relationship between the variables. Truncated regression is often used in various fields such as economics, finance, and social sciences to handle data with limitations and provide more accurate and reliable results.

Truncated Regression | R Data Analysis Examples

Truncated regression is used to model dependent variables for which some of the observations are not included in the analysis because of the value of the dependent variable.

This page uses the following packages. Make sure that you can load them before trying to run the examples on this page. If you do not have a package installed, run: `install.packages("packagename")`, or if you see the version is out of date, run: `update.packages()`.

require(foreign)require(ggplot2)require(truncreg)require(boot)

```
Version info: Code for this page was tested in R version 3.1.1 (2014-07-10)
On: 2014-08-21
With: boot 1.3-11; truncreg 0.2-1; maxLik 1.2-0; miscTools 0.6-16; ggplot2
1.0.0; foreign 0.8-61; knitr 1.6
```

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 truncated regression

Example 1. A study of students in a special GATE (gifted and talented education) program wishes to model achievement as a function of language skills and the type of program in which the student is currently enrolled. A major concern is that students are required to have a minimum achievement score of 40 to enter the special program. Thus, the sample is truncated at an achievement score

of 40.

Example 2. A researcher has data for a sample of Americans whose income is
above the poverty line. Hence, the lower part of the distribution of
income is truncated. If the researcher had a sample of Americans whose
income was at or below the poverty line, then the upper part of the income
distribution would be truncated. In other words, truncation is a result of
sampling only part of the distribution of the outcome variable.

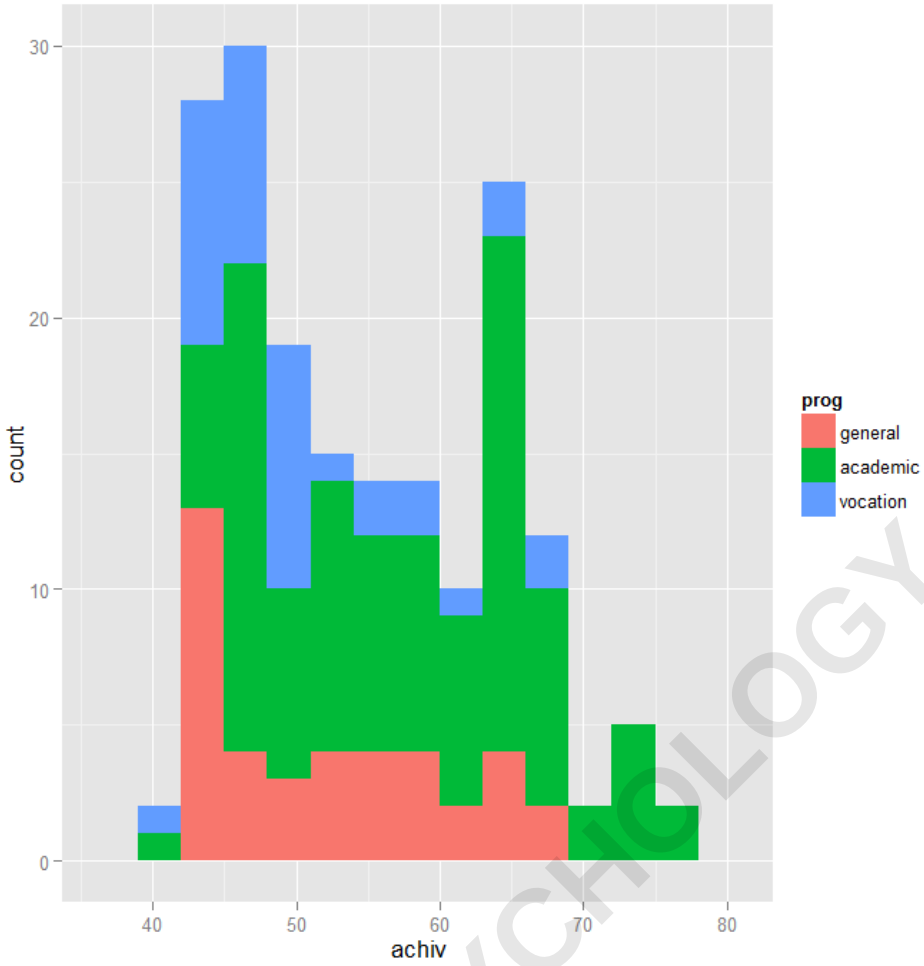
Description of the data

Let's pursue Example 1 from above. We have a hypothetical data file,
`truncreg.dta`, with 178 observations. The
outcome variable is called `achiv`, and the language test score
variable is called `langscore`. The variable `prog` is a
categorical predictor variable with three levels indicating the type

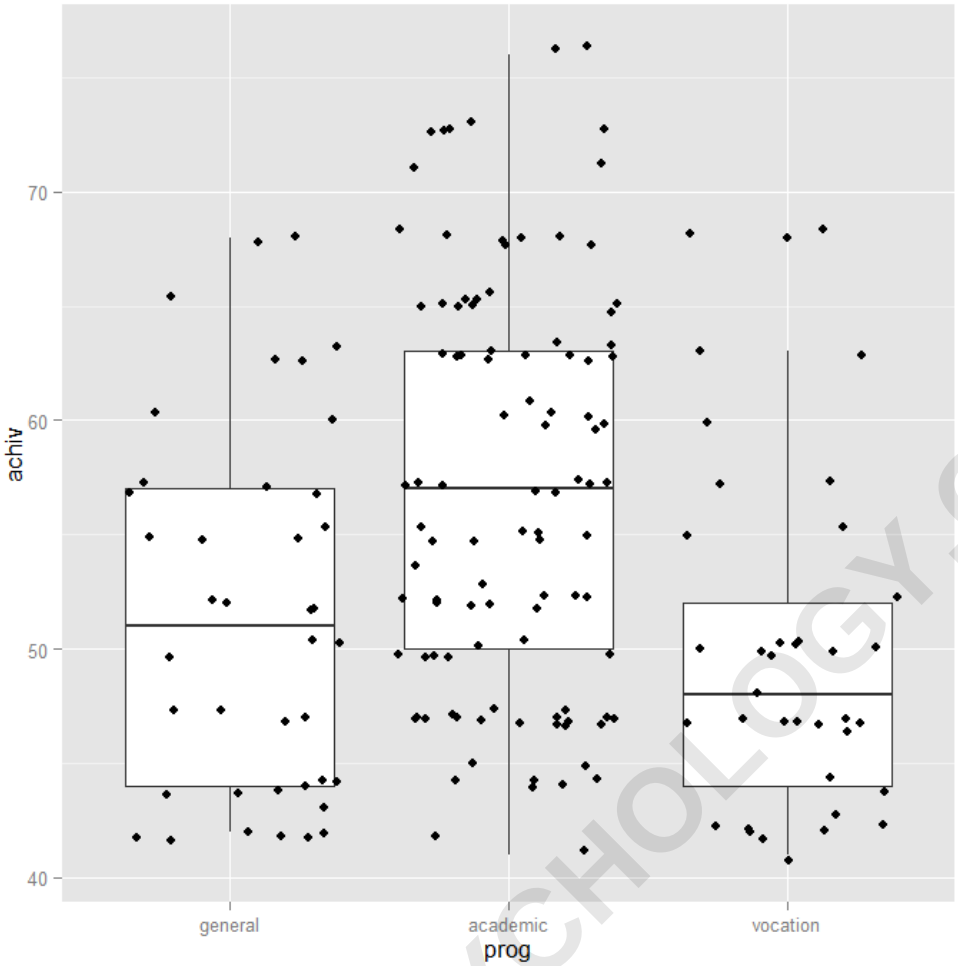
of program in which the students were enrolled.

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

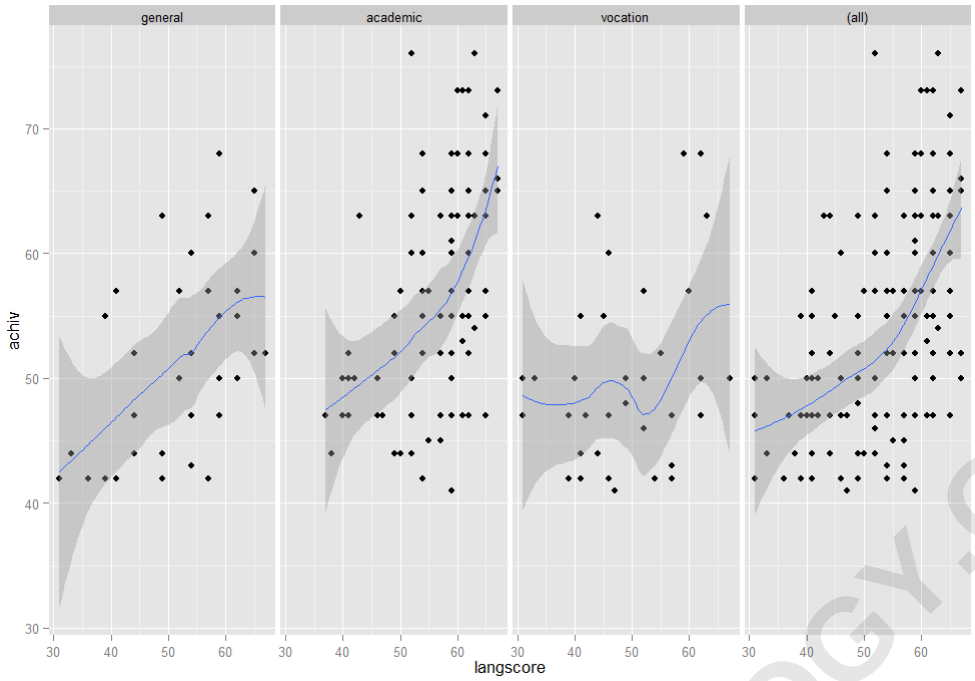
```
dat<-
read.dta("https://stats.idre.ucla.edu/stat/data/truncreg.dta")summary(dat)

## id achiv langscore prog
## Min. : 3.0 Min. :41.0 Min. :31.0 general : 40
## 1st Qu.: 55.2 1st Qu.:47.0 1st Qu.:47.5 academic:101
## Median :102.5 Median :52.0 Median :56.0 vocation: 37
## Mean :103.6 Mean :54.2 Mean :54.0
## 3rd Qu.:151.8 3rd Qu.:63.0 3rd Qu.:61.8
## Max. :200.0 Max. :76.0 Max. :67.0

# histogram of achiv coloured by program typeggplot(dat,aes(achiv,fill=
prog))+geom_histogram(binwidth=3)
```



 # boxplot of achiv by program typeggplot(dat,aes(prog, achiv))+geom_boxplot()+geom_jitter()




```
ggplot(dat,aes(x= langscore,y= achiv))+geom_point()+stat_smooth(method="loess")+facet_grid(.~prog,margins=TRUE)
```



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.

Truncated regression

Below we use the `truncreg` function in the `truncreg` package to estimate a truncated regression model. The `point` argument indicates where the data are truncated, and the direction indicates whether it is

left or right truncated.

```
m<-truncreg(achiv~langscore+prog,data=
dat,point=40,direction="left")summary(m)

##
## Call:
## truncreg(formula = achiv ~ langscore + prog, data =
dat, point = 40,
## direction = "left")
##
##
## Coefficients :
## Estimate Std. Error t-value Pr(>|t|)
## (Intercept) 11.299 6.772 1.67 0.095 .
## langscore 0.713 0.114 6.23 4.8e-10 ***
## progacademic 4.063 2.054 1.98 0.048 *
## progvocation -1.144 2.670 -0.43 0.668
## sigma 8.754 0.666 13.13

# update old model dropping progm2<-update(m, .~.-
prog)pchisq(-2*(logLik(m2)-
logLik(m)),df=2,lower.tail=FALSE)
```

'log Lik.' 0.0252 (df=3)

The two degree-of-freedom chi-square test indicates that `prog` is a
statistically significant predictor of `achiv`. We can get the expected means for each program at the mean of `langscore` by
reparameterizing the model.

```
# create mean centered langscore to use laterdat<-within(dat, {mlangscore<-langscore-mean(langscore)})malt<-truncreg(achiv~0+mlangscore+prog,data= dat,point=40)summary(malt)


##
## Call:
## truncreg(formula = achiv ~ 0 + mlangscore + prog, data = dat,
## point = 40)
##
##
## Coefficients :
## Estimate Std. Error t-value Pr(>|t|)
```

mlangscore 0.713 0.114 6.22 4.8e-10 ***
proggeneral 49.789 1.897 26.24

Notice all that has changed is the intercept is gone and the program scores
are now the expected values when `langscore` is at its mean for each
type of program.

We could also calculate the bootstrapped confidence intervals if
we wanted to. First, we define a function that returns the parameters
of interest, and then use the `boot` function to run the bootstrap.

```
f<-function(data,i) {require(truncreg)m<-truncreg(formula= achiv~langscore+prog,data= data,point=40)as.vector(t(summary(m)$coefficients))}set.seed(10)(res<-boot(dat, f,R=1200,parallel="snow",ncpus=4))
```

##
ORDINARY NONPARAMETRIC BOOTSTRAP

```
##
##
## Call:
## boot(data = dat, statistic = f, R = 1200, parallel =
"snow",
## ncpus = 4)
##
##
## Bootstrap Statistics :
## original bias std. error
## t1* 11.299 0.303557 5.9327
## t2* 6.772 -0.055936 0.8620
## t3* 0.713 -0.004048 0.0965
## t4* 0.114 -0.000685 0.0137
## t5* 4.063 -0.053784 2.0333
## t6* 2.054 -0.001430 0.2411
## t7* -1.144 0.021373 2.8721
## t8* 2.670 0.012132 0.2944
## t9* 8.754 -0.109523 0.5500
## t10* 0.666 -0.010924 0.0754
```

We could use the bootstrapped standard error to get a normal approximation
for a significance test and confidence intervals for

every parameter. However,
instead we will get the percentile and bias adjusted 95 percent confidence
intervals, using the `boot.ci` function.

```
# basic parameter estimates with percentile and bias adjusted CIsparms<-t(sapply(c(1,3,5,7,9),function(i) {out<-boot.ci(res,index=c(i, i+1),type=c("perc","bca"))with(out,c(Est= t0,pLL= percent,pUL= percent,bcaLL= bca,bcaLL= bca))}))# add row namesrow.names(parms)<-names(coef(m))# print resultsparms

## Est pLL pUL bcaLL bcaLL
## (Intercept) 11.299 -1.258 22.297 -3.7231 21.320
## langscore 0.713 0.539 0.916 0.5508 0.944
## progacademic 4.063 0.058 8.011 0.0842 8.043
## progvocation -1.144 -6.805 4.277 -6.8436 4.250
## sigma 8.754 7.674 9.792 7.8896 10.110
```

The conclusions are the same as from the default model tests. You can compute
a rough estimate of the degree of association for the overall model,

by correlating `achiv` with the predicted value and squaring the result.

```
dat$yhat<-fitted(m)#  correlation(r<-with(dat,cor(achiv, yhat)))
```

```
## 0.552
```

```
# rough variance accounted forr^2
```

```
## 0.305
```

The calculated value of .31 is rough estimate of the R2 you would find in an OLS regression. The squared correlation between the observed and predicted academic aptitude values is about 0.31, indicating that these predictors accounted for over 30% of the variability in the outcome variable.

Things to consider

References