How can Multinomial Logistic Regression be applied in R for data analysis?

Authored by

stats writer

June 29, 2024

Multinomial Logistic Regression is a statistical technique used for analyzing data that involves predicting a categorical outcome with more than two possible values. This method is commonly applied in R, a statistical programming language, for data analysis. To use Multinomial Logistic Regression in R, the data must first be structured in a format that includes the outcome variable and the predictor variables. The R package "nnet" is then used to fit the model and obtain the coefficients, which represent the relationship between the predictor variables and the outcome. This information can then be used to make predictions on new data. Additionally, R provides various tools for evaluating the performance of the model, such as confusion matrices and classification tables. Overall, Multinomial Logistic Regression in R is a powerful tool for analyzing categorical data and making predictions based on the relationships between variables.

Multinomial Logistic Regression | R Data Analysis Examples

Multinomial logistic regression is used to model nominal outcome variables,
in which the log odds of the outcomes are modeled as a linear
combination of the predictor variables.

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if you see the version is out of date, run: `update.packages()`.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(nnet)require(ggplot2)require(reshape2)

```
Version info: Code for this page was tested in R version 3.1.0 (2014-04-10)
On: 2014-06-13
With: reshape2 1.2.2; ggplot2 0.9.3.1; nnet 7.3-8; foreign 0.8-61; knitr 1.5
```

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 multinomial logistic regression

Example 1. People's occupational choices might be influenced by their parents' occupations and their own education level. We can study the relationship of one's occupation choice with education level and father's occupation. The occupational choices will be the outcome variable which consists of categories of occupations.

Example 2. A biologist may be interested in food

choices that alligators make.
Adult alligators might have different preferences from young ones.
The outcome variable here will be the types of food, and the predictor
variables might be size of the alligators and other environmental variables.

Example 3. Entering high school students make program choices among
general program, vocational program and academic program.
Their choice might be modeled using their writing score and their social economic status.

Description of the data

For our data analysis example, we will expand the third example using
the `hsbdemo` data set. Let's first read in the data.

```
ml<-
read.dta("https://stats.idre.ucla.edu/stat/data/hsbdemo.dta")
```

The data set contains variables on 200 students. The outcome variable
is `prog`, program type. The predictor variables are social economic status,
`ses`, a three-level categorical variable and writing score, `write`,
a continuous variable. Let's start with getting some descriptive
statistics of the variables of interest.

```
with(ml,table(ses, prog))
```

```
## prog
## ses general academic vocation
## low 16 19 12
## middle 20 44 31
## high 9 42 7
```

```
with(ml,do.call(rbind,tapply(write,
prog,function(x)c(M=mean(x),SD=sd(x)))))
```

```
## M SD
## general 51.33 9.398
## academic 56.26 7.943
```

vocation 46.76 9.319

Analysis methods you might consider

Multinomial logistic regression

Below we use the `multinom` function from the `nnet` package to estimate a multinomial logistic regression model. There are other functions in other R packages capable of multinomial regression. We chose the `multinom` function because it does not require the data to be reshaped (as the `mlogit` package does) and to mirror the example code found in Hilbe's *Logistic Regression Models*.

First, we need to choose the level of our outcome that we wish to use as our baseline and specify this in the `relevel` function. Then, we run our model using `multinom`. The `multinom` package does not include p-value calculation for the regression coefficients, so we calculate p-values using Wald tests (here z-tests).

```
ml$prog2<-relevel(ml$prog,ref="academic")test<-multinom(prog2~ses+write,data= ml)

## # weights: 15 (8 variable)
## initial value 219.722458
## iter 10 value 179.982880
## final value 179.981726
## converged

summary(test)

## Call:
## multinom(formula = prog2 ~ ses + write, data = ml)
##
## Coefficients:
## (Intercept) sesmiddle seshigh write
## general 2.852 -0.5333 -1.1628 -0.05793
## vocation 5.218 0.2914 -0.9827 -0.11360
##
## Std. Errors:
## (Intercept) sesmiddle seshigh write
## general 1.166 0.4437 0.5142 0.02141
## vocation 1.164 0.4764 0.5956 0.02222
##
```

Residual Deviance: 360
AIC: 376

```
z<-
summary(test)$coefficients/summary(test)$standard.errorsz
```

(Intercept) sesmiddle seshigh write
general 2.445 -1.2018 -2.261 -2.706
vocation 4.485 0.6117 -1.650 -5.113

```
# 2-tailed z testp<-(1-pnorm(abs(z),0,1))*2p
```

(Intercept) sesmiddle seshigh write
general 1.448e-02 0.2294 0.02374 6.819e-03
vocation 7.299e-06 0.5408 0.09895 3.176e-07

The ratio of the probability of choosing one outcome category over the
probability of choosing the baseline category is often referred as relative risk
(and it is sometimes referred to as *odds*, described in the regression parameters above). The relative risk is

the right-hand side linear equation exponentiated, leading to the fact that the exponentiated regression coefficients are relative risk ratios for a unit change in the predictor
variable. We can exponentiate the coefficients from our model to see these
risk ratios.

```
## extract the coefficients from the model and exponentiateexp(coef(test))

## (Intercept) sesmiddle seshigh write
## general 17.33 0.5867 0.3126 0.9437
## vocation 184.61 1.3383 0.3743 0.8926
```

You can also use predicted probabilities to help you understand the model.
You can calculate predicted probabilities for each of our outcome levels using the
`fitted` function. We can start by generating the predicted probabilities
for the observations in our dataset and viewing the first few rows

```
head(pp<-fitted(test))

## academic general vocation
## 1 0.1483 0.3382 0.5135
## 2 0.1202 0.1806 0.6992
## 3 0.4187 0.2368 0.3445
## 4 0.1727 0.3508 0.4765
## 5 0.1001 0.1689 0.7309
## 6 0.3534 0.2378 0.4088
```

Next, if we want to examine the changes in predicted probability associated
with one of our two variables, we can create small datasets varying one variable
while holding the other constant. We will first do this holding `write` at
its mean and examining the predicted probabilities for each level of `ses`.

```
dses<-
data.frame(ses=c("low","middle","high"),write=mean(ml
$write))predict(test,newdata= dses,"probs")

## academic general vocation
```

1 0.4397 0.3582 0.2021
2 0.4777 0.2283 0.2939
3 0.7009 0.1785 0.1206

Another way to understand the model using the predicted probabilities is to look at the averaged predicted probabilities for different values of the continuous predictor variable `write` within each level of `ses`.

```
dwrite<-
data.frame(ses=rep(c("low","middle","high"),each=41),
write=rep(c(30:70),3))## store the predicted probabilities for each value of ses and writepp.write<-cbind(dwrite,predict(test,newdata=
dwrite,type="probs",se=TRUE))## calculate the mean probabilities within each level of sesby(pp.write, pp.write$ses, colMeans)
```

pp.write$ses: high
academic general vocation
0.6164 0.1808 0.2028
--

pp.write$ses: low
academic general vocation
0.3973 0.3278 0.2749
--
pp.write$ses: middle
academic general vocation
0.4256 0.2011 0.3733

Sometimes, a couple of plots can convey a good deal amount of information.
Using the predictions we generated for the pp.write object above, we can plot the predicted probabilities against the writing score by the
level of ses for different levels of the outcome variable.

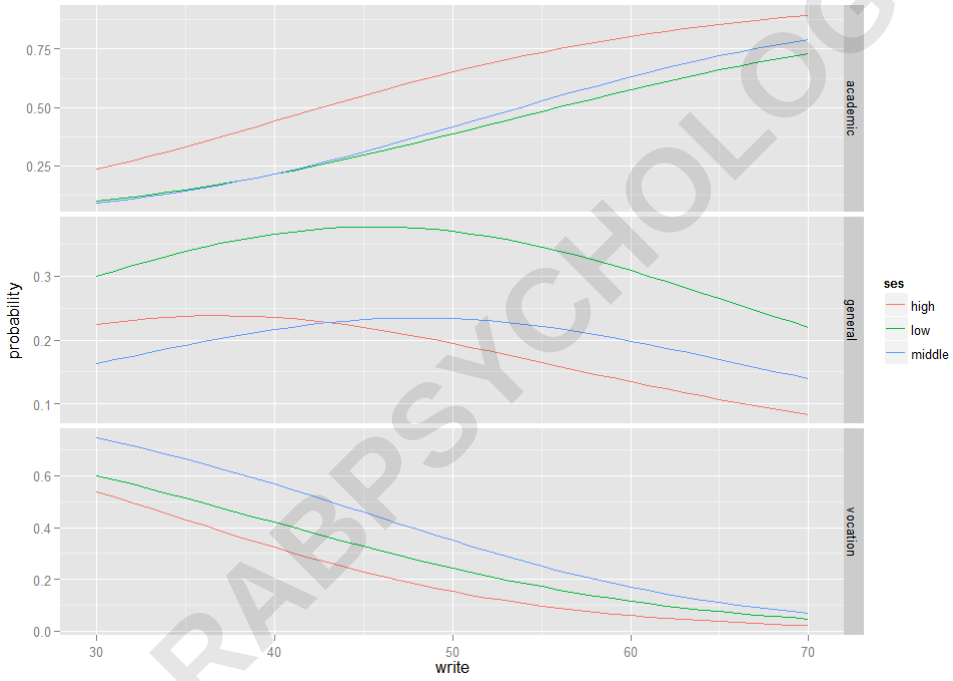
melt data set to long for ggplot2lpp<-melt(pp.write,id.vars=c("ses","write"),value.name="probability")head(lpp)# view first few rows

ses write variable probability
1 low 30 academic 0.09844
2 low 31 academic 0.10717
3 low 32 academic 0.11650
4 low 33 academic 0.12646

5 low 34 academic 0.13705

6 low 35 academic 0.14828

plot predicted probabilities across write values for each level of ses## facetted by program typeggplot(lpp,aes(x= write,y= probability,colour= ses))+geom_line()+facet_grid(variable~.,scales="free")



Things to consider

See also