

Ordered logistic regression R

Below we use the `polr` command from the MASS package to estimate an ordered logistic regression model. The command name comes from proportional odds logistic regression, highlighting the proportional odds assumption in our model.

Ordered logistic regression R

`polr` uses the standard formula interface in R for specifying a regression model with outcome followed by predictors. We also specify `Hess=TRUE` to have the model return the observed information matrix from optimization (called the Hessian) which is used to get standard errors.

Ordered logistic regression R

```
## fit ordered logit model and store results
m <- polr(apply ~ pared +
          public + gpa, data = dat, Hess=TRUE)

## view a summary of the model
summary(m)
## Call:
## polr(formula = apply ~ pared +
        public + gpa, data = dat, Hess = TRUE)
```

Ordered logistic regression R

```
## Coefficients:
```

	Value	Std. Error	t value
pared	1.0477	0.266	3.942
public	-0.0588	0.298	-0.197
gpa	0.6159	0.261	2.363

```
##
```

```
## Intercepts:
```

	Value	Std. E
unlikely somewhat likely	2.204	0.780
somewhat likely very likely	4.299	0.804

```
##
```

Ordered logistic regression R

In the output above, we see

Call, this is R reminding us what type of model we ran, what options we specified, etc. Next we see the usual regression output coefficient table including the value of each coefficient, standard errors, and t value, which is simply the ratio of the coefficient to its standard error. There is no significance test by default.

Ordered logistic regression R

Next we see the estimates for the two intercepts, which are sometimes called cutpoints. The intercepts indicate where the latent variable is cut to make the three groups that we observe in our data. Note that this latent variable is continuous. In general, these are not used in the interpretation of the results. The cutpoints are closely related to thresholds, which are reported by other statistical packages.

Ordered logistic regression R

- ▶ Finally, we see the residual deviance, $-2 * \text{Log Likelihood}$ of the model as well as the AIC. Both the deviance and AIC are useful for model comparison.
- ▶ Some people are not satisfied without a p value. One way to calculate a p-value in this case is by comparing the t-value against the standard normal distribution, like a z test.

Ordered logistic regression R

Of course this is only true with infinite degrees of freedom, but is reasonably approximated by large samples, becoming increasingly biased as sample size decreases. This approach is used in other software packages such as Stata and is trivial to do. First we store the coefficient table, then calculate the pvalues and combine back with the table.

Ordered logistic regression R

```
## store table
(ctable <- coef(summary(m)))
##
```

	Value	Std. Error	t
## pared	1.04769	0.2658	3
## public	-0.05879	0.2979	-0
## gpa	0.61594	0.2606	2
## unlikely somewhat likely	2.20391	0.7795	2
## somewhat likely very likely	4.29936	0.8043	5

```
## calculate and store p values
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE)
```

Ordered logistic regression R

```
## combined table
(ctable <- cbind(ctable, "p value" = p))
##
```

	Value	Std. Error	t
## pared	1.04769	0.2658	3
## public	-0.05879	0.2979	-0
## gpa	0.61594	0.2606	2
## unlikely somewhat likely	2.20391	0.7795	2
## somewhat likely very likely	4.29936	0.8043	5