

Modelos Discretos

Regresión Ordinal – Librería "ordinal"

Mtr. Alcides Ramos Calcina



Regresión Ordinal

Librería “ordinal”

Mtr. Alcides Ramos Calcina

Ejemplo 6 – Librería “ordinal”



Sigamos considerando los datos de acidez de vinos.

```
datos <- read.csv("C:/.../wine.txt", sep="")  
head(datos)
```

	response	rating	temp	contact	bottle	judge
1	36	2	cold	no	1	1
2	48	3	cold	no	2	1
3	47	3	cold	yes	3	1
4	67	4	cold	yes	4	1
5	77	4	warm	no	5	1
6	60	4	warm	no	6	1

Importante revisar la estructura de la base de datos, donde las variables independientes deben estar en condición de factor y la variable dependiente como factor ordenado.

Ejemplo 6



```
datos$temp <- factor(datos$temp)
datos$contact <- factor(datos$contact)
datos$rating <- factor(datos$rating, ordered = T)
```

Verificando.

```
str(datos)
```

```
'data.frame': 72 obs. of  6 variables:
 $ response: int  36 48 47 67 77 60 83 90 17 22 ...
 $ rating  : Ord.factor w/ 5 levels "1"<"2"<"3"<"4"<..: 2 3 3 4 4 4 ...
 $ temp    : Factor w/ 2 levels "cold","warm": 1 1 1 1 2 2 2 2 1 1 ...
 $ contact : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 2 2 1 1 ...
 $ bottle  : int  1 2 3 4 5 6 7 8 1 2 ...
 $ judge   : int  1 1 1 1 1 1 1 1 2 2 ...
```

Ejemplo 6



Ahora instalamos y activamos la librería **ordinal**, previo a la estimación del modelo.

```
install(ordinal)  
library(ordinal)
```

Ajustamos el modelo acumulado para los datos **wine**, a través de la función **clm** (Cumulative Link Models).

A diferencia del análisis anterior, esta vez iniciaremos estimando el primer modelo considerando las variables de temperatura y contacto.

Ejemplo 6



Estimación del modelo.

```
mod1 <- clm(rating ~ temp + contact, data = datos)
```

```
mod1
```

```
formula: rating ~ temp + contact
```

```
data:      datos
```

link	threshold	nobs	logLik	AIC	niter	max.grad	cond.H
logit	flexible	72	-86.49	184.98	6(0)	4.02e-12	2.7e+01

Coefficients:

tempwarm contactyes

2.503 1.528

Threshold coefficients:

1 2	2 3	3 4	4 5
-1.344	1.251	3.467	5.006

Ejemplo 6



Resultados más detallados a través de la función **summary**.

```
summary(mod1)
```

```
formula: rating ~ temp + contact
```

```
data:      datos
```

```
link threshold nobs logLik AIC      niter max.grad cond.H  
logit flexible  72    -86.49 184.98 6(0)   4.02e-12 2.7e+01
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
tempwarm	2.5031	0.5287	4.735	2.19e-06	***
contactyes	1.5278	0.4766	3.205	0.00135	**

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Threshold coefficients:

	Estimate	Std. Error	z value
1 2	-1.3444	0.5171	-2.600
2 3	1.2508	0.4379	2.857
3 4	3.4669	0.5978	5.800
4 5	5.0064	0.7309	6.850

Ejemplo 6



El primer resultado es la tabla de coeficientes estimados, errores estándar y los p-valores basados en el método de Wald. Sin embargo, las pruebas de razón de verosimilitud proporcionan pruebas más precisas.

```
mod2 <- clm(rating ~ temp, data = datos)
anova(mod2, mod1)
```

Likelihood ratio tests of cumulative link models:

	formula:	link:	threshold:
mod2	rating ~ temp	logit	flexible
mod1	rating ~ temp + contact	logit	flexible

	no.par	AIC	logLik	LR.stat	df	Pr(>Chisq)
mod2	5	194.03	-92.013			
mod1	6	184.98	-86.492	11.043	1	0.0008902 ***

En este caso produce un *p*-valor ligeramente menor, indicando la significancia de la variable.

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Ejemplo 6



Seguidamente realizaremos la evaluación de la variable temperatura, planteando un tercer modelo.

```
mod3 <- clm(rating ~ contact, data = datos)
anova(mod3, mod1)
```

Likelihood ratio tests of cumulative link models:

	formula:	link:	threshold:
mod3	rating ~ contact	logit	flexible
mod1	rating ~ temp + contact	logit	flexible

	no.par	AIC	logLik	LR.stat	df	Pr(>Chisq)
mod3	5	209.91	-99.956			
mod1	6	184.98	-86.492	26.928	1	2.112e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Del mismo modo, la variable temperatura también es significativa.

Ejemplo 6



podemos usar la función **drop1** para obtener pruebas de razón de verosimilitud de las variables explicativas mientras se controlan el resto variables:

```
drop1(mod1, test = "Chi")
```

Single term deletions

Model:

```
rating ~ temp + contact
```

	Df	AIC	LRT	Pr(>Chi)	
<none>		184.98			
temp	1	209.91	26.928	2.112e-07	***
contact	1	194.03	11.043	0.0008902	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Ejemplo 6



También, podemos usar la función **drop1** para obtener pruebas de razón de verosimilitud de las variables explicativas mientras se controlan el resto variables:

```
drop1(mod1, test = "Chi")
```

Single term deletions

Model:

```
rating ~ temp + contact
```

	Df	AIC	LRT	Pr(>Chi)	
<none>		184.98			
temp	1	209.91	26.928	2.112e-07	***
contact	1	194.03	11.043	0.0008902	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Ejemplo 6



Las pruebas de razón de verosimilitud de las variables explicativas ignorando el resto vienen dadas por el método **add1**:

```
mod0 <- clm(rating ~ 1, data = datos)
add1(mod0, scope = ~temp + contact, test = "Chi")
```

Single term additions

Model:

rating ~ 1

	Df	AIC	LRT	Pr(>Chi)
<none>		215.44		
temp	1	194.03	23.4113	1.308e-06 ***
contact	1	209.91	7.5263	0.00608 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

En este caso, estas últimas pruebas no son tan fuertes como las pruebas que controlan la otra variable.

Ejemplo 6



Finalmente, como parte del método de selección de variables, los intervalos de confianza se obtienen a partir del comando `confint` como sigue:

```
confint(mod1)
```

	2.5 %	97.5 %
tempwarm	1.5097627	3.595225
contactyes	0.6157925	2.492404

Estos se basan en los perfiles de la función de verosimilitud y generalmente son bastante precisos.

Ejemplo 6



Menos preciso, pero más simple y con intervalos de confianza simétricos basados en los errores estándar de los parámetros (también llamados, intervalos de confianza de Wald) se obtienen con:

```
confint(mod1, type = "Wald")
```

	2.5 %	97.5 %
1 2	-2.3578848	-0.330882
2 3	0.3925794	2.109038
3 4	2.2952980	4.638476
4 5	3.5738541	6.438954
tempwarm	1.4669081	3.539296
contactyes	0.5936345	2.461961

Ejemplo 6



Ahora que en el modelo 1 se justifica la inclusión de las variables temperatura y contacto, procedemos a estimar los OR.

```
exp(coef(mod1))
```

1 2	2 3	3 4	4 5	tempwarm	contactyes
0.2607004	3.4931671	32.0368540	149.3666773	12.2203428	4.6080172

Predicciones



Los valores ajustados se obtienen con la función **fitted(mod1)** y producen probabilidades ajustadas, es decir, la *i*-ésima probabilidad ajustada sería la probabilidad de que la *i*-ésima observación caiga sobre la categoría prevista.

```
p_mod1 <- predict(mod1, type = "class")  
p_mod1$fit[1:20]  
[1] 2 2 3 3 3 3 4 4 2 2 3 3 3 3 4 4 2 2 3 3  
Levels: 1 2 3 4 5
```


Predicciones



Digamos que solo queremos las predicciones para las cuatro posibles combinaciones entre temperatura y contacto, entonces sería:

```
comb <- expand.grid(temp = levels(datos$temp), contact =  
levels(datos$contact))
```

```
cbind(comb, predict(mod1, newdata = comb)$fit)
```

	temp	contact	1	2	3	4	5
1	cold	no	0.206790132	0.57064970	0.1922909	0.02361882	0.00665041
2	warm	no	0.020887709	0.20141572	0.5015755	0.20049402	0.07562701
3	cold	yes	0.053546010	0.37764614	0.4430599	0.09582084	0.02992711
4	warm	yes	0.004608274	0.05380128	0.3042099	0.36359581	0.27378469

FINESI

Modelos Discretos

IV Semestre



<https://aulavirtual2.unap.edu.pe/>

GRACIAS

