PRÁCTICA 3B: REGRESIÓN MULTINOMIAL

ANÁLISIS ESTADÍSTICO MULTIVARIANTE

GRADO EN CIENCIA E INGENIERÍA DE DATOS

Sumario: En esta práctica mostramos cómo llevar a cabo un análisis de Regresión Multinomial (RMultinomial) usando R. La estimación de los parámetros del modelo se realizará usando la función *multinom()* del paquete `nnet` de R, que utiliza también para redes neuronales. Recordemos que el modelo RMultinomial es una generalización del modelo de Regresión Logística, donde la variable respuesta, en lugar de ser binaria, es de tipo categórico con más de dos niveles. Además, mostramos cómo convertir el análisis RMultinomial en un problema de clasificación en varios grupos y calculamos la matriz de confusión de la clasificación.

1. Conjunto de datos y análisis descriptivo previo

Los datos que usaremos en esta práctica han sido descargados de la web de la Universidad de UCLA, cuya práctica sobre Regresión Multinomial ha servido de base para el desarrollo de la nuestra. El conjunto de datos **RegMultinomial_example.csv** contiene datos simulados de 200 estudiantes universitarios. Las variables del fichero son: `write` (puntuación obtenida en una prueba escrita), `ses` (estatus económico del estudiante, con niveles: bajo, medio y alto) y `prog` (tipo de programa a cursar por el estudiante, con niveles: académico, general y vocacional).

Objetivo del estudio: analizar cómo afectan las variables `write` y `ses` en el programa a cursar por el estudiante. Obsérvese que disponemos de dos predictores, uno de ellos de tipo continuo (`write`) y otro categórico (`ses`), y que la variable respuesta (`prog`) es categórica con 3 niveles, de manera que plantearemos un análisis de Regresión Multinomial.

Comenzamos cargando los datos y haciendo un resumen de los mismos.

```
library("tidyverse")
mydata <- read.csv2("../data/RegMultinomial_example.csv")
summary(mydata)
```

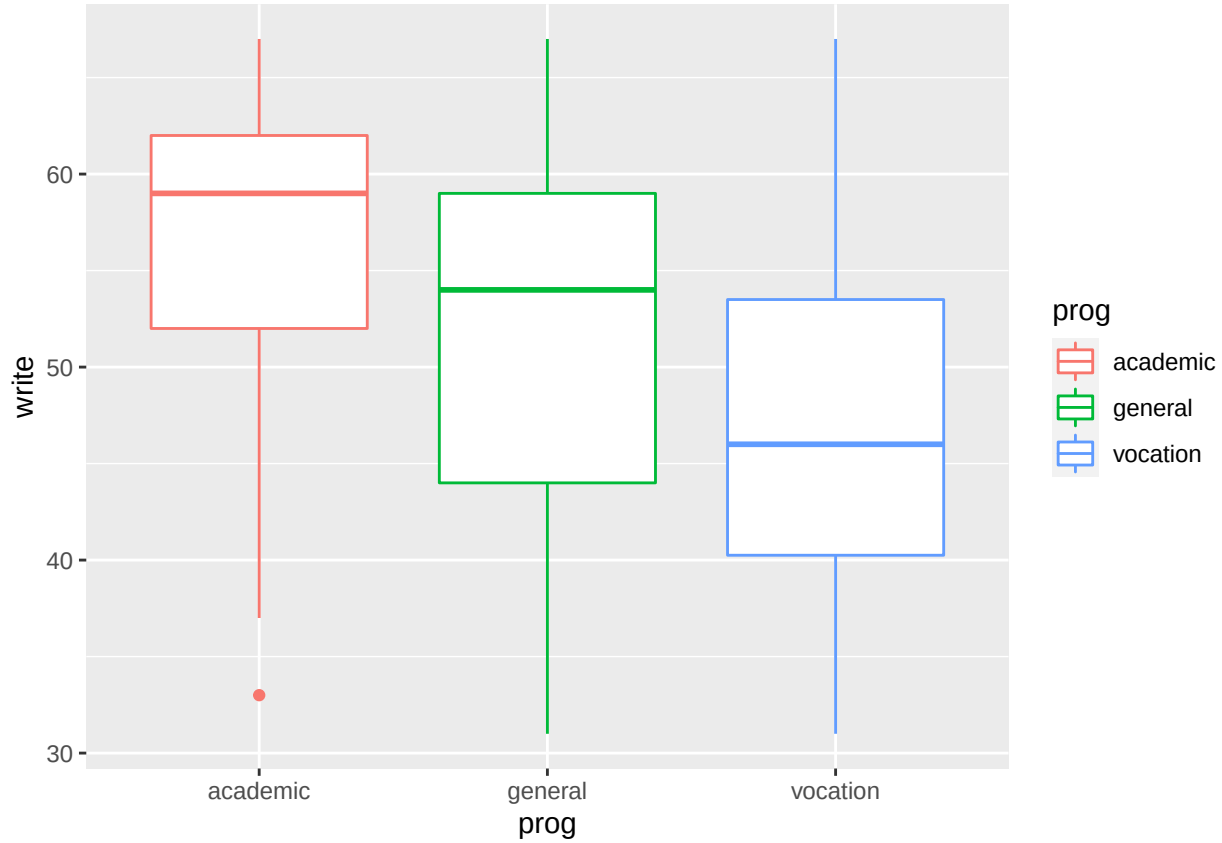
```
##      ses                prog               write
##  Length:200         Length:200         Min.   :31.00
##  Class :character   Class :character   1st Qu.:45.75
##  Mode  :character   Mode  :character   Median :54.00
##                                        Mean   :52.77
##                                        3rd Qu.:60.00
##                                        Max.   :67.00
```

Pasamos las variables `prog` y `ses` a tipo factor, aunque no sería necesario para algunos análisis.

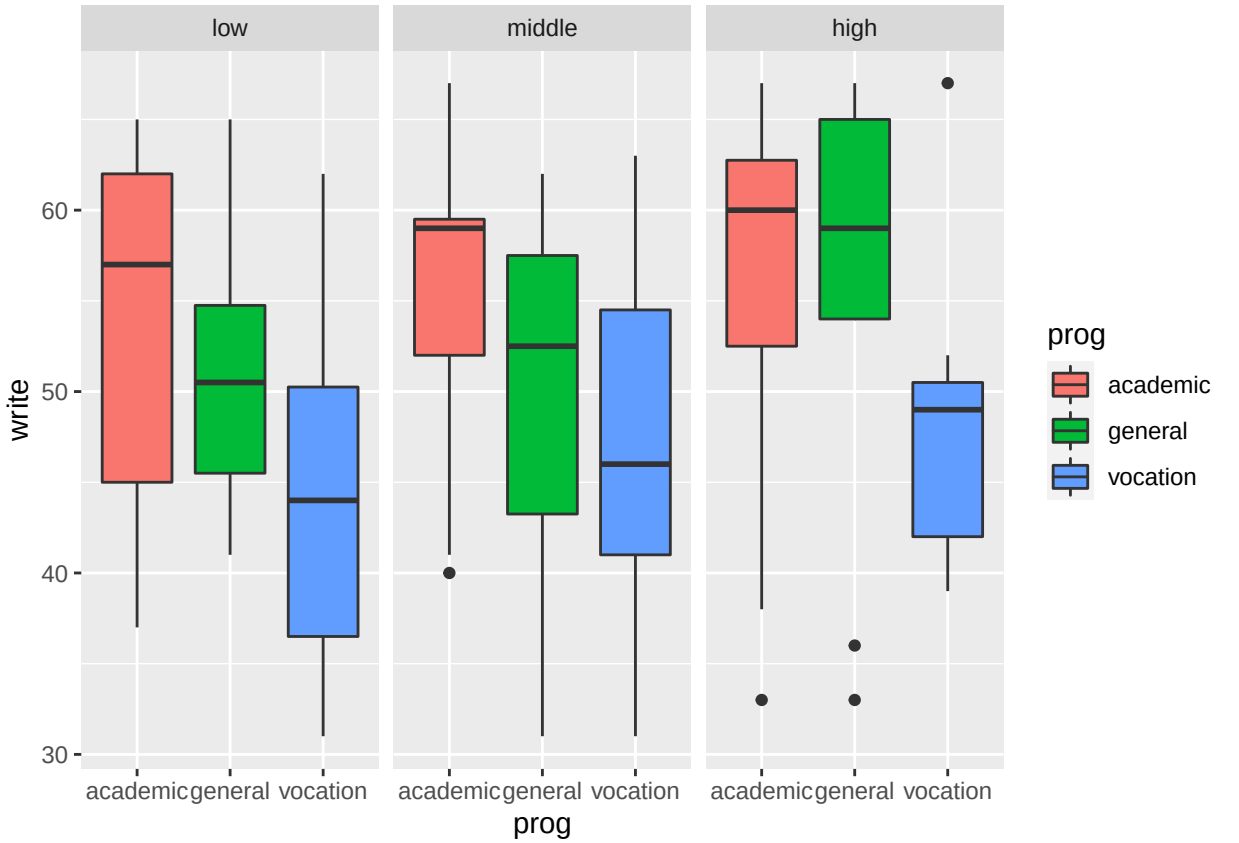
```
mydata$prog <- factor(mydata$prog)
mydata$ses <- factor(mydata$ses, levels = c("low", "middle", "high"))
#ponemos el argumento levels para ordenar niveles

#summary(mydata)
```

Realizamos un análisis descriptivo inicial.

```
#Diagramas de caja comparativos de "write" para cada nivel de la variable respuesta
mydata %>%
  ggplot(aes(x = prog, y = write)) +
  geom_boxplot(aes(color = prog))
```




```
#Ahora separamos los boxplot para cada nivel del predictor categorico "ses"
mydata %>%
  ggplot(aes(x = prog, y = write)) +
  geom_boxplot(aes(fill=prog))+
  facet_grid (.~ ses)
```



Comprobamos que no hay celdas vacías (o frecuencias muy bajas) en las tablas de contingencia de los predictores categóricos y la variable respuesta. Si hay celdas vacías o con pocos casos, el modelo RMultinomial puede ser inestable y poco fiable.

```
addmargins(table(mydata$ses, mydata$prog,
                 dnn = c("status", "program")))
```

```
##         program
## status   academic general vocation Sum
##   low          19      16       12  47
##   middle       44      20       31  95
##   high         42       9        7  58
##   Sum         105      45       50 200
```

Tras el estudio descriptivo previo, podemos observar influencia de los predictores `write` y `ses` en la variable respuesta.

2. Estimación de los parámetros del modelo RMultinomial y métodos de selección de regresores

Recordemos que el modelo teórico de Regresión Multinomial supone que el logaritmo de los *odds* de pertenecer a una clase de la variable respuesta frente a pertenecer a la clase de referencia viene explicado por un modelo lineal de los predictores. En nuestro caso, si tomamos el programa académico como clase de referencia, el modelo sería:

$$\log\left(\frac{p_j}{p_{ref}}\right) = \theta_0^{(j)} + \theta_1^{(j)} \cdot write + \theta_2^{(j)} \cdot sesmiddle + \theta_3^{(j)} \cdot seshigh\,,$$

donde p_j denota la probabilidad de pertenecer a la clase j-ésima de la variable respuesta ($j =$ "*general*", "*vocation*") y p_{ref} la probabilidad de pertenecer a la clase de referencia (`academic`).

Hay que entender las variables `sesmiddle` y `seshigh` como variables binarias (dummy) que toman el valor 1 si el valor de `ses` se corresponde con su nivel y valen cero en caso contrario. Por ejemplo, si $ses = high$, entonces $sesmiddle = 0$ y $seshigh = 1$.

Comenzamos estimando el modelo multinomial usando todos los predictores (en este caso `write` y `ses`). Para ello, usamos la función *multinom()* del paquete `nnet` de R. Primero tenemos que seleccionar la clase de la variable respuesta que actuará como referencia, usando la función 'relevel´.

Tomaremos el **programa académico como clase de referencia**. De esta forma, tendremos un modelo lineal ajustado para explicar el logaritmo de los +odds* de participar en el programa general frente al académico y otro modelo lineal ajustado para explicar el logaritmo de los *odds* de participar en el programa vocacional frente al académico.

```
library("nnet")
```

```
## Warning: package 'nnet' was built under R version 4.1.3
```

```
mydata$prog <- relevel(mydata$prog, ref = "academic")
mymultinom <- multinom(prog ~ ses + write, data = mydata)
```

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

```
summary(mymultinom)
```

```
## Call:
## multinom(formula = prog ~ ses + write, data = mydata)
##
## Coefficients:
##          (Intercept)  sesmiddle    seshigh      write
## general     2.852198 -0.5332810 -1.1628226 -0.0579287
## vocation    5.218260  0.2913859 -0.9826649 -0.1136037
##
## Std. Errors:
##          (Intercept) sesmiddle   seshigh      write
## general     1.166441 0.4437323 0.5142196 0.02141097
## vocation    1.163552 0.4763739 0.5955665 0.02221996
##
## Residual Deviance: 359.9635
## AIC: 375.9635
```

De los resultados anteriores, obtenemos que el modelo lineal ajustado que explica el logaritmo de los odds de participar en el programa general frente al académico viene dado por:

$$2.8522 - 0.0579 \cdot write - 0.5333 \cdot sesmiddle - 1.1628 \cdot seshigh$$

Y el modelo lineal ajustado que explica el logaritmo de los *odds* de participar en el programa vocacional frente al académico viene dado por:

$$5.2183 - 0.1136 \cdot write + 0.2914 \cdot sesmiddle - 0.9827 \cdot seshigh$$

La **interpretación de los coefientes** del modelo RMultinomial sería la siguiente:

Por cada unidad adicional en la variable `write`, el logaritmo de los *odds* de participar en el programa general frente al académico disminuye en 0.0579.

Por cada unidad adicional en la variable `write`, el logaritmo de los *odds* de participar en el programa vocacional frente al académico disminuye en 0.1136.

El logaritmo de los *odds* de participar en el programa general frente al académico decrecerá en 1.1628 si pasamos de estatus bajo a estatus alto.

De forma similar tendríamos las interpretaciones de todos los coeficientes.

Como en el caso de RLM y RL, podemos aplicar los **métodos de selección de regresores** *backward*, *fordward* y *stepwise*, para ver si el modelo completo es reducible a otro más sencillo (con menos predictores).

```
modelo_backward <- step(mymultinom, direction = "backward")
```

```
## Start:  AIC=375.96
## prog ~ ses + write
##
## trying - ses
## # weights:  9 (4 variable)
## initial  value 219.722458
## final  value 185.510837
## converged
## trying - write
## # weights:  12 (6 variable)
## initial  value 219.722458
## iter  10 value 195.705189
## iter  10 value 195.705188
## iter  10 value 195.705188
## final  value 195.705188
## converged
##         Df      AIC
## <none>   8 375.9635
## - ses    4 379.0217
## - write  6 403.4104
```

```
modelo_nulo <- multinom(prog ~ 1, data = mydata)
```

```
## # weights:  6 (2 variable)
## initial  value 219.722458
## final  value 204.096674
## converged
```

```
modelo_forward <- step(modelo_nulo, scope = formula(mymultinom),
                       direction = "forward")
```

```
## Start:  AIC=412.19
## prog ~ 1
##
## trying + ses
## # weights:  12 (6 variable)
## initial  value 219.722458
## iter  10 value 195.705189
## iter  10 value 195.705188
## iter  10 value 195.705188
## final  value 195.705188
```

```
## converged
## trying + write
## # weights:  9 (4 variable)
## initial  value 219.722458
## final  value 185.510837
## converged
##          Df      AIC
## + +write  4 379.0217
## + +ses    6 403.4104
## <none>    2 412.1933
## # weights:  9 (4 variable)
## initial  value 219.722458
## final  value 185.510837
## converged
##
## Step:  AIC=379.02
## prog ~ write
##
## trying + ses
## # weights:  15 (8 variable)
## initial  value 219.722458
## iter  10 value 179.982880
## final  value 179.981726
## converged
##        Df      AIC
## + +ses  8 375.9635
## <none>  4 379.0217
## # weights:  15 (8 variable)
## initial  value 219.722458
## iter  10 value 179.982880
## final  value 179.981726
## converged
##
## Step:  AIC=375.96
## prog ~ write + ses
```

```
modelo_stepwise <- step(modelo_nulo, scope = formula(mymultinom),
                        direction = "both")
```

```
## Start:  AIC=412.19
## prog ~ 1
##
## trying + ses
## # weights:  12 (6 variable)
## initial  value 219.722458
## iter  10 value 195.705189
## iter  10 value 195.705188
## iter  10 value 195.705188
## final  value 195.705188
## converged
## trying + write
## # weights:  9 (4 variable)
## initial  value 219.722458
## final  value 185.510837
## converged
```

```
##          Df      AIC
## + +write  4 379.0217
## + +ses    6 403.4104
## <none>    2 412.1933
## # weights:  9 (4 variable)
## initial  value 219.722458
## final  value 185.510837
## converged
##
## Step:  AIC=379.02
## prog ~ write
##
## trying - write
## # weights:  6 (2 variable)
## initial  value 219.722458
## final  value 204.096674
## converged
## trying + ses
## # weights:  15 (8 variable)
## initial  value 219.722458
## iter  10 value 179.982880
## final  value 179.981726
## converged
##         Df      AIC
## + +ses   8 375.9635
## <none>   4 379.0217
## - write  2 412.1933
## # weights:  15 (8 variable)
## initial  value 219.722458
## iter  10 value 179.982880
## final  value 179.981726
## converged
##
## Step:  AIC=375.96
## prog ~ write + ses
##
## trying - write
## # weights:  12 (6 variable)
## initial  value 219.722458
## iter  10 value 195.705189
## iter  10 value 195.705188
## iter  10 value 195.705188
## final  value 195.705188
## converged
## trying - ses
## # weights:  9 (4 variable)
## initial  value 219.722458
## final  value 185.510837
## converged
##         Df      AIC
## <none>   8 375.9635
## - ses    4 379.0217
## - write  6 403.4104
```

Observamos que los tres métodos conducen al modelo completo con los 2 predictores, así que ese será nuestro modelo de referencia.

La **bondad del ajuste** del modelo se puede medir a través del `AIC` (criterio de Akaike), siendo mejor el ajuste cuanto menor sea el AIC. También se puede usar la `devianza residual`, siendo mejor el ajuste cuanto menor sea la devianza.

```
mymultinom$AIC #Valor AIC del modelo
```

```
## [1] 375.9635
```

```
mymultinom$deviance #Valor de la devianza
```

```
## [1] 359.9635
```

```
#Comprobamos que la devianza del modelo coincide con -2*log(likelihood)
-2*logLik(mymultinom)
```

```
## 'log Lik.' 359.9635 (df=8)
```

3. Inferencias en el modelo RMultinomial. Predicciones

Una vez que se valide el modelo RMultinomial (véase la última sección), tendremos garantías para realizar inferencias con dicho modelo. Estas inferencias incluyen la obtención de intervalos de confianza para los parámetros de regresión, la prueba de significación del modelo, pruebas individuales de significación de los predictores y también para un conjunto de predictores (test de Wald).

En primer lugar calculamos los **intervalos de confianza** para los parámetros de regresión.

```
confint(mymultinom, level = 0.95)
```

```
## , , general
##
##                   2.5 %      97.5 %
## (Intercept)  0.56601580  5.13838015
## sesmiddle   -1.40298023  0.33641823
## seshigh     -2.17067452 -0.15497060
## write       -0.09989343 -0.01596397
##
## , , vocation
##
##                   2.5 %      97.5 %
## (Intercept)  2.9377406  7.49877877
## sesmiddle   -0.6422898  1.22506155
## seshigh     -2.1499538  0.18462402
## write       -0.1571540 -0.07005342
```

Recordemos que los coeficientes del modelo miden la variación del logaritmo de los *odds* por unidad de cambio en el correspondiente predictor. Tomando exponenciales sobre los coeficientes, medimos las variaciones producidas sobre los *odds* directamente (sin tomar logaritmo).

```
exp(coef(mymultinom))
```

```
##          (Intercept) sesmiddle   seshigh     write
## general     17.32582 0.5866769 0.3126026 0.9437172
## vocation   184.61262 1.3382809 0.3743123 0.8926116
```

Medimos la **significación del modelo** comparando la devianza del modelo completo frente a la devianza del modelo nulo. Esta diferencia sigue una Chi-cuadrado con los grados de libertad dados por la diferencia

de grados de ambos modelos. Recordar que los grados de libertad para el modelo nulo se corresponden con $n-1$ (tamaño de la muestra menos 1) y los grados de libertad para el modelo completo se corresponden con $n-k-1$ (tamaño de la muestra menos número de coeficientes de regresión)

```
diferencia_devianzas <- modelo_nulo$deviance - mymultinom$deviance
grados_libertad <- 199 - 196
p_valor <- pchisq(diferencia_devianzas, df = grados_libertad, lower.tail = FALSE)
p_valor
```

```
## [1] 1.902608e-10
```

Obsérvese que el p-valor resultante permite concluir que el modelo completo es significativo.

La signficación individual de cada predictor se obtiene con el test de Wald y el estadístico Z.

```
valores_z <- summary(mymultinom)$coefficients/summary(mymultinom)$standard.errors
valores_z
```

```
##          (Intercept)  sesmiddle   seshigh     write
## general     2.445214 -1.2018081 -2.261334 -2.705562
## vocation    4.484769  0.6116747 -1.649967 -5.112689
```

```
p_valores <- 2*(1 - pnorm(abs(valores_z), 0, 1))
p_valores
```

```
##           (Intercept) sesmiddle    seshigh        write
## general  0.0144766100 0.2294379 0.02373856 6.818902e-03
## vocation 0.0000072993 0.5407530 0.09894976 3.176045e-07
```

Observamos en este caso que el estatus económico medio (`sesmiddle`) no es significativo (p-valores > 0.10).

Para realizar **predicciones** del modelo RMultinomial, usamos la función *predict()*, que en este caso permite hacer predicciones de las probabilidades asociadas a cada clase (argumento `type = "probs"`) y de la clase de la variable respuesta (argumento por defecto, `type = "class"`).

Por ejemplo, hagamos la predicción para un estudiante con `write=58` y `ses=high`, veremos que lo clasificaría como que cursará el programa `academic`.

```
nuevos_1 <- data.frame(write = 58, ses = "high")
predict(mymultinom, newdata = nuevos_1, type = "probs")
```

```
##   academic    general   vocation
## 0.77930272 0.14662970 0.07406758
```

```
predict(mymultinom, newdata = nuevos_1)
```

```
## [1] academic
## Levels: academic general vocation
```

Los valores ajustados nos proporcionan, para cada fila de inputs (predictores), la probabilidad de pertenecer a cada uno de los 3 niveles de la variable respuesta

```
# fitted(mymultinom)
```

Podemos mantener un predictor constante y variar el otro con el fin de ver cómo cambian las probabilidades predichas.

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

```
##    academic   general  vocation
## 1 0.4396845 0.3581917 0.2021238
```

```
## 2 0.4777488 0.2283353 0.2939159
## 3 0.7009007 0.1784939 0.1206054
```

```
predict(mymultinom, newdata = dses, type = "class")
```

```
## [1] academic academic academic
## Levels: academic general vocation
```

Interpretación: para un alumno con nota de la prueba escrita igual a la media, lo más probable es que seleccione el programa académico sea cual sea su estatus social, aumentando dicha probabilidad conforme aumenta el estatus.

Ahora variamos los resultados de la prueba escrita, es decir, la variable `write`

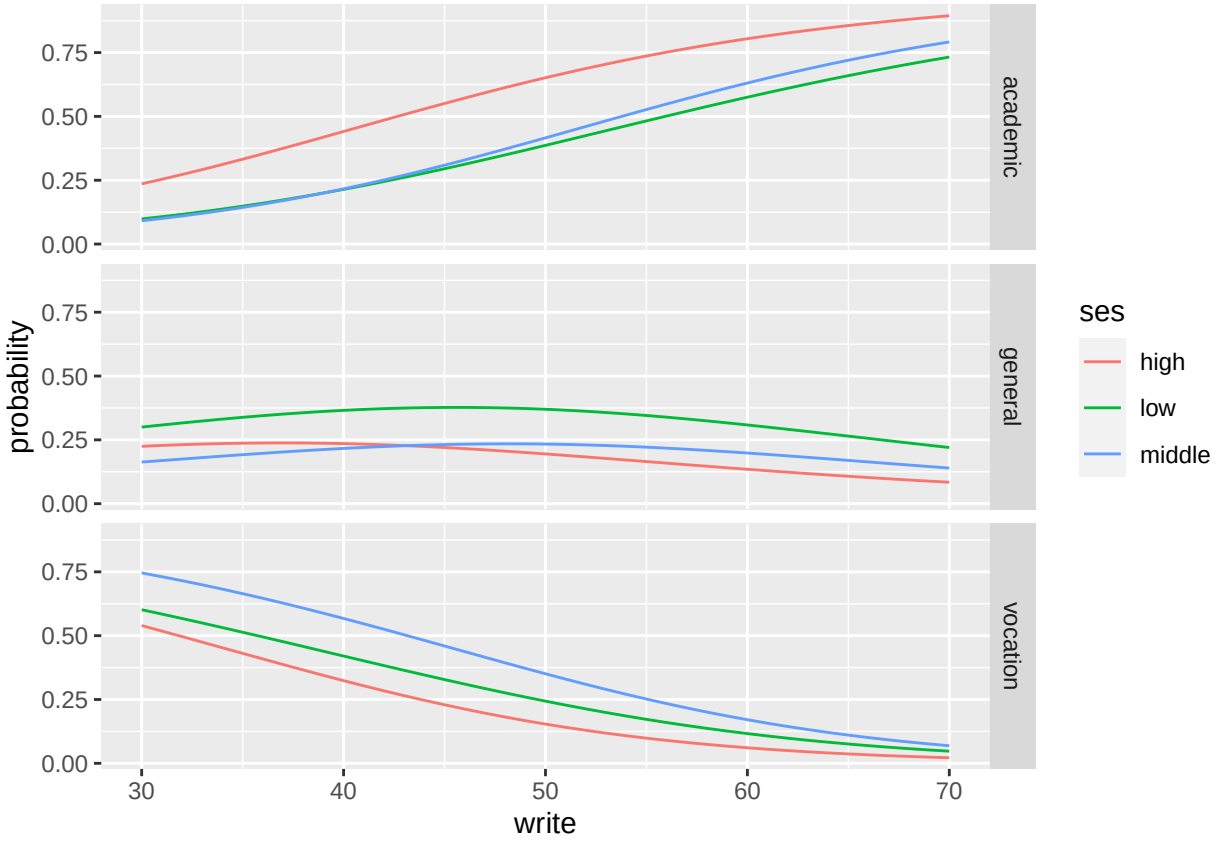
```
dwrite <- data.frame(ses = rep(c("low", "middle", "high"), each = 41),
                     write = rep(c(30:70), 3))

## Guardamos las probabilidades predichas para cada valor de ses y write
pp.write <- cbind(dwrite, predict(mymultinom, newdata = dwrite, type = "probs",
                                  se = TRUE))
# pp.write
```

Modificamos el `dataframe` para usar `ggplot2` y representar las probabilidades predichas.

```
pp.write_longer <- pivot_longer(pp.write, cols = c(3:5),
                                names_to = "program", values_to = "probability")
head(pp.write_longer)
```

```
## # A tibble: 6 x 4
##   ses   write program  probability
##   <chr> <int> <chr>          <dbl>
## 1 low      30 academic      0.0984
## 2 low      30 general       0.300
## 3 low      30 vocation      0.602
## 4 low      31 academic      0.107
## 5 low      31 general       0.308
## 6 low      31 vocation      0.585
```

```
pp.write_longer %>%
  ggplot(aes(x = write, y = probability, colour = ses)) +
  geom_line() +
  facet_grid (program ~.)
```



Obsérvese que cuando aumenta la nota en `write`, los aumnos se decantan más por la vertiente académica mientras que ocurre lo contrario para las vocacionales. Esta nota tiene poca relación con la clase general.

4. La Regresión Multinomial como un problema de clasificación

Como en RMultinomial la variable respuesta es categórica, podemos entenderlo como un problema de clasificación con varios grupos (tantos como niveles de la variable respuesta). De hecho, al realizar las predicciones usando el argumento `type="class"`, estamos definiendo un clasificador.

Vamos a calcular la predicción del grupo (o clase) para cada fila de inputs (predictores), y luego creamos un dataframe con datos originales y predicciones.

```
prog_predict <- predict(mymultinom, newdata = mydata, "class")
mydata_predict <- cbind(mydata, prog_predict)
```

Calculamos la matriz de confusión de la clasificación, que se construye comparando los valores observados de la muestra con las predicciones del modelo.

```
matriz_confusion <- table(mydata_predict$prog, mydata_predict$prog_predict,
                          dnn = c("real", "predicho"))
matriz_confusion
```

```
##           predicho
## real       academic general vocation
##   academic       92       4        9
##   general        27       7       11
##   vocation       23       4       23
```

Obtenemos una medida de bondad del ajuste (*accuracy*), dividiendo aciertos entre total de datos.

```
accuracy <- sum(diag(matriz_confusion)) / sum(matriz_confusion)
accuracy
```

```
## [1] 0.61
```

IMPORTANTE: En este ejemplo no hemos dividido el conjunto de datos en `entrenamiento` y `test`. Por tanto, se puede dar sobreajuste (*overfitting*) al calcular la matriz de confusión y el *accuracy* (tasa de aciertos).

5. Consideraciones adicionales

1) Como en el caso de la Regresión Logística, hay que comprobar que no tenemos celdas vacías o muy pequeñas al hacer las tablas de contingencia.
2) Como en el caso de la Regresión Logística, hay que llevar cuidado con el problema de separación completa o casi completa.
3) En Regresión Multinomial también se usa el estimador máximo verosímil, así que conviene tener tamaños muestrales grandes.
4) La validación del modelo es más complicada por no disponer de herramientas directas, sino que requiere realizar modelos logísticos por separado. Debe cumplirse la linealidad, la independencia, no debe existir multicolinealidad entre los predictores y no debe haber observaciones influyentes. Aunque no se cumplan algunas hipótesis, este método de clasificación podría dar buenos resultados. Recíprocamente, si se cumplen las hipótesis pero el método de clasificación no es eficiente, el método carece de utilidad.
5) Existen otras funciones y paquetes para realizar Regresion Multinomial, por ejemplo, la función *mlogit()* del paquete `mlogit`.
6) En el caso de que los niveles de la variables respuesta esten ordenados, conviene usar el modelo multinomial ordinal (función *clm()* `cumulative link models` del paquete `ordinal`)