Regresión logística múltiple

Contenido

1. Formulación

Considerando k variables cuantitativas $X_1, X_2, \dots, X_k$, entonces, para cada combinación de dichas variables, se tiene que la variable de respuesta Y sigue una distribución de Bernoulli.

$Y/(X_1 = x_1, \dots, X_k = x_k)$ tiende $\beta(1, p(x_1, \dots, x_k))$

al igual que en el caso del modelo simple, nos interesa modelar la esperanza condicionada

$$E[Y/(X_1 = x_1, \dots, X_k = x_k)] = P[Y = 1/X_1 = x_1, \dots, X_k = x_k] = p(x_1, \dots, x_k)$$

El modelo de regresión logística múltiple para Y en términos de los valores de las variables X, se puede modelizar como:

$$p(x_1, \dots, x_R) = \frac{\exp\left(\sum_{r=1}^{R} \beta_r x_r\right)}{1 + \exp\left(\alpha + \left(\sum_{r=1}^{R} \beta_r x_r\right)\right)}$$

si notamos $\alpha = 0$ y $x_0 = 1$ la expresión quedaría cómo

$$p(x_1, \dots, x_R) = \frac{\exp\left(\sum_{r=0}^{R} \beta_r x_r\right)}{1 + \exp\left(\alpha + \left(\sum_{r=0}^{R} \beta_r x_r\right)\right)}$$

que en términos matriciales sería

$$p(x) = \frac{\exp\ \beta^t \mathrm{x}}{1 + \exp\ \beta^t \mathrm{x}}$$

Con *x el vector 1,* $x_1, \dots, x_R$ y $\beta = \beta_0, \dots, \beta_R$

Al igual que en el caso de una sola variable explicativa, podemos considerar un modelo lineal para la transformación logit de p(x) como sigue

$$\ln\left[\frac{\mathrm{p(x)}}{1 + p(x)}\right] = \sum_{r=0}^{R} \beta_r x_r$$

con lo que tenemos un modelo lineal generalizado cuya función link es la transformación logit. En la figura (2.3) vemos la curva logística con dos variables explicativas en el intervalo (−10, 10) y con todos los $\beta_r = 1$

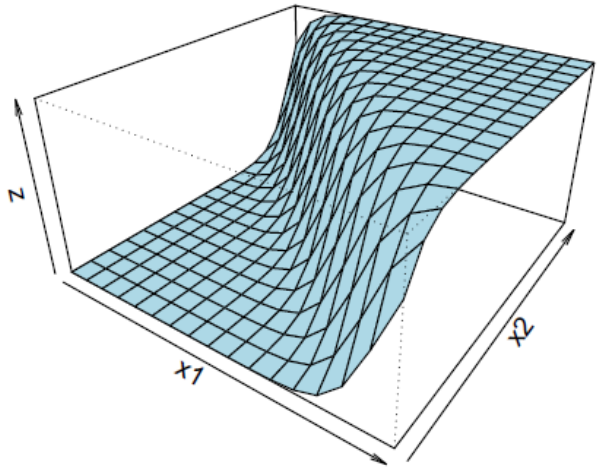


Figura 2.3 Función logit con dos variables explicativas continuas x_1 y x_2 y con parámetros $\beta_0 = 1, \beta_2 = 1$. z es la probabilidad estimada

2. Interpretación

- Si todos los β_r son iguales a cero salvo β_0 entonces $p(x) = \frac{e^{\beta_o}}{1+e^{\beta_o}}$, con lo que en este caso la variable Y es independiente de las explicativas.
- β_0 es el valor del logaritmo de la ventaja de respuesta Y = 1 frente a Y = 0 cuando $\forall_r$ = 1 . . .r o también es el valor del logaritmo de la ventaja para un caso donde $X_1 = X_2 = \ldots = X_r = 0$
- El cociente de ventajas entre dos configuraciones de los valores de las variables explicativas $x_1 = (1, x_{11}, \ldots, x_{1R})$ y $x_2 = (1, x_{21}, \ldots, x_{2R})$ seria:

$$\emptyset(x_1, x_2) = \frac{\frac{p(x_1)}{1-p(x_1)}}{\frac{p(x_2)}{1-p(x_2)}} = \frac{exp(\sum_{r=0}^{R} \beta_r x_{1r})}{exp(\sum_{r=0}^{R} \beta_r x_{2r})} = exp\left(\sum_{r=1}^{R} \beta_r (x_{1r} - x_{2r})\right)$$

Si la diferencia entre x_1 y x_2 en cada valor de $X_{1,\ldots,}X_R$ es de 1. Entonces

$$\emptyset(x_1, x_2) = exp\left(\sum_{r=1}^{R} \beta_r\right) = \prod_{r=1}^{R} e^{\beta_r}$$

Si la diferencia entre x_1 y x_2 es de 1, pero sólo en una de las variables explicativas, digamos en X_l mientras que sus valores son los mismos en el resto de variables, entonces:

$$\emptyset(x_1, x_2) = e^{\beta_t}$$

Es decir, el exponencial del parámetro asociado a la variable X_l es la cantidad por la que queda multiplicada la ventaja de respuesta Y=1 cuando el valor en X_l aumenta en una unidad, sin que cambien los valores en el resto de variables explicativas.

3. Para realizar una regresión logística tener en cuenta:

Tener claro qué se pretende en el estudio. Esto es especialmente importante cuando se llevan a cabo análisis multivariante, en los que se introducen muchas (> 2) variables simultáneamente para evaluar sus relaciones o asociaciones, por lo que las posibilidades de encontrar dependencias espúreas (cuando no absurdas) es elevada; y, por otra parte, las probabilidades de no encontrar relaciones importantes por no saber cómo explorarlas o por la imprecisión de los datos (error aleatorio) también es alta.

4. La Regresión Logística Multivariante tiene tres objetivos básicos:

1. Obtener una estimación no sesgada o ajustada de la relación entre la variable dependiente y una variable independiente que es la que el investigador quiere conocer.

   "Efecto del tabaquismo materno sobre el bajo peso al nacer: un estudio caso-control"

2. Evaluar varios factores simultáneamente que estén presumiblemente relacionados de alguna manera (o no) con la variable dependiente, y conocer su papel (predictor, confundente, modificador de efecto) y su efecto de forma ajustada.

   "Factores que influyen en el bajo peso al nacer"

   En este caso no hay una variable independiente principal sino varias, que habrán sido seleccionadas por el investigador tras un profundo conocimiento del tema en cuestión y una rigurosa búsqueda bibliográfica. El análisis de RLM permitirá obtener medidas de asociación (OR) para cada variable ajustadas por las demás y detectar posibles interacciones entre ellas y el efecto estudiado (BAJO PESO).

3. Construir un modelo y obtener una ecuación con fines de predicción o cálculo del riesgo, de manera que éste pueda estimarse para un nuevo individuo con una cierta validez y precisión.

   "Predicción del bajo peso al nacer: una fórmula para calcular el riesgo"
   El investigador debe conocer muy bien el tema en cuestión, tener información fidedigna de aquellos factores que ya se conocen de riesgo o de protección, y disponer de una amplia muestra de individuos donde medir con el menor error posible estas variables.

5. Idea intuitiva

La regresión logística múltiple es una extensión de la regresión logística simple, se basa en los mismos principios que la regresión logística simple, pero ampliando el número de predictores. Los predictores pueden ser tanto continuos como categóricos.

$$ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_2 + \beta_2 X_3 + \cdots + \beta_i X_i$$

$$logit(Y) = \beta_0 + \beta_1 X_2 + \beta_2 X_3 + \cdots + \beta_i X_i$$

El valor de la probabilidad de Y se puede obtener con la inversa del logaritmo natural:

$$p(Y) = \frac{e^{\beta_0 + \beta_1 X_2 + \beta_2 X_3 + \cdots + \beta_i X_i}}{1 + e^{(\beta_0 + \beta_1 X_2 + \beta_2 X_3 + \cdots + \beta_i X_i)}}$$

A la hora de evaluar la validez y calidad de un modelo de regresión logística múltiple se analiza tanto el modelo en su conjunto como los predictores que lo forman. Se considera que el modelo es útil si es capaz de mostrar una mejora respecto al modelo nulo, el modelo sin predictores. Existen 3 test estadísticos que cuantifican esta mejora mediante la comparación de los residuos: *likelihood ratio*, *score* y *Wald test*. No hay garantías de que los 3 lleguen a la misma conclusión, cuando esto ocurre parece ser recomendable basarse en el *likelihood ratio*.

6. Condiciones del modelo logístico

La regresión logística no requiere de ciertas condiciones como linealidad, normalidad y homocedasticidad de los residuos que sí lo son para la regresión lineal. Las principales condiciones que este modelo requiere son:

- **Respuesta binaria:** La variable dependiente ha de ser binaria.
- **Independencia:** las observaciones han de ser independientes.
- **Multicolinealidad:** se requiere de muy poca a ninguna multicolinealidad entre los predictores (para regresión logística múltiple).
- **Linealidad** entre la variable independiente y el logaritmo natural de odds.
- **Tamaño muestral:** como regla general, se requiere un mínimo de 10 casos con el resultado menos frecuente para cada variable independiente del modelo.

7. Explorar las asociaciones bi-variantes

Un posible primer paso es explorar es la posible asociación entre la variable dependiente (Y) y las diferentes variables independientes (X_i), esto tiene como objetivo tener una primera aproximación a la estimación de la medida de asociación. Si es dicotómica se usa la OR y si es politómica la Chi cuadrado. Aquí reconoceremos si pudieran tratarse de estimaciones sesgadas, si existiese confusión, de estimaciones poco informativas o si existiese interacción con una tercera variable.

Un aspecto previo muy importante a tener en cuenta en el análisis de variables categóricas es el tema de la codificación numérica de las categorías. Tras comprobar que efectivamente nuestras variables están medidas en una escala NOMINAL (también conocida por cualitativa o categórica), conviene fijarse en los números que identifican cada categoría, pues en los procedimientos automáticos de análisis, el programa va a considerar siempre la categoría de referencia (1) aquella que tiene menor valor numérico.

Ejemplo: obteniendo ji cuadrado de las variables bajo peso y hábito tabáquico materno. Pearson Chi cuadrado = 4.924 con $p(0.026) < \alpha(0.05)$, indicaría que las

variables se asocian. Obteniendo Risk estimate (Odds ratio) = 2.022 con IC Lower(1.081) y Upper(3.783, indicaría que la fuerza de esta asociación es 2.022 que representa el riesgo que tienen las madres fumadoras frente a las que no fuman (categoría de referencia en este contraste, al tener el valor "0") de tener un RN de bajo peso. Dicho de otra manera, el hábito materno de ser fumadora hace que se incremente por dos (se duplique) el riesgo de tener un RN de bajo peso. El OR es significativo debido a que el intervalo no contiene al valor "cero".

8. Recomendaciones:

A) Si se trata de variables categóricas lo haremos a través del procedimiento Tablas de contingencia (Crosstabs).

   1) Lo mejor es trabajar con variables categóricas dicotómicas, pues en ellas se establece una categoría "de referencia" y se calcula la OR para la categoría "expuesta" en relación a dicha categoría "de referencia".

   2) Si tenemos variables politómicas un procedimiento aconsejable es colapsar o agrupar categorías para transformarlas en dicotómicas.

   3) Si se trata de variables ordinales, podemos explorar si hay asociación lineal con la variable dependiente e introducirlas en el modelo logístico como variables continuas, ofreciéndonos entonces la OR calculada un valor medio del riesgo de cada categoría frente a la inmediatamente anterior en orden decreciente.

B) Si se trata de variables continuas podemos optar por dos soluciones:

   1) Evaluar si hay diferencias en las medias de la dicha variable contínua comparando los dos grupos que se establecen por las dos categorías de la variable dependiente, a través de un test T de Student o de un ANOVA de una vía.

      Finalmente podría trabajarse con variables tipo DUMMY

   2) Intentar transformaciones de la variable contínua en categórica, preferiblemente dicotómica. El punto de corte puede establecerse arbitrariamente, aunque debe tenerse en cuenta…

Si existe una hipótesis teórica que pueda operativizarse en el estudio y que tenga cierto sentido explorar; así, por ejemplo, si se sospecha que el seguimiento médico, operativizado en el número de visitas durante el embarazo (VISITAS) puede ser un factor predictor de BAJO PESO, una categorización posible de la variable independiente sería "ninguna visita ni control médico" (VISITAS = 0) versus "al menos una visita médica durante la gestación" (VISITAS ≥ 1).

Si no hay una hipótesis previa, un buen punto de corte es la mediana, que permite agrupar los individuos en dos grupos de igual tamaño; o los quantiles en general.

Cuando la variable predictora es…	Asociación con una variable dependiente dicotómica		Recomendación
	Análisis bivariante simple	Análisis de regresión logística binaria simple	
Dicotómica nominal	Chi cuadrado … OR	Test de Wald .. OR	Dejar tal cual
Politómica nominal	Chi cuadrado	Test de Wald, Se crean variables dummy como categorías menos uno… OR	Intente agrupar o colapsar categorías para transformarla en dicotómica
Ordinal	Chi cuadrado u otras pruebas. No se calcula OR si hay más de 2 categorias		Intente agrupar o colapsar categorías para transformarla en dicotómica o prueba introducirla como continua si se detecta asociación lineal
Continua	T test o ANOVA para diferencia de medias	Test de Wald. OR en relaciona valores -1	Intente categorizarla o dicotomizarla

Ejemplo. *Veamos cómo se relaciona la variable RAZA* (independiente) *con la variable BAJOPESO (dependiente)*. La variable independiente tiene tres categorías (1: blanca; 2: negra; 3: otras). Una evaluación de asociación arroja los siguientes resultados.

Raza	Bajo peso al nacer		Total
	>=2500 gr	< 2500 gr	
Blanca	73	23	96
Negra	15	11	26
Otras	42	25	67
Total	130	59	189

Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	5,005[a]	2	,082
Razón de verosimilitud	5,010	2	,082
Asociación lineal por lineal	3,570	1	,059
N de casos válidos	189		

a. 0 casillas (,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 8,12.

No se encuentra asociación debido a que el test Chi cuadrado no es significativo $p(0.082) > \alpha(0.05)$, aunque si evaluamos el gráfico de barras agrupadas podemos ver mejor lo que está pasando.

Transformando la variable RAZA para convertirla en una variable dicotómica; tendría sentido colapsar la categoría "Blanca" con "Otras razas", o juntar "Negras" con "Otras razas", pero no agrupar en una sola categoría la raza "Blanca" con la raza "Negra", puesto que tienen los valores extremos de proporción de RN de bajo peso.

Raza	Bajo peso al nacer		Total
	>=2500 gr	< 2500 gr	
Blanca	73	23	96
Negra y otras	57	36	93
Total	130	59	189

Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	4,787[b]	1	,029		
Corrección por continuidad[a]	4,125	1	,042		
Razón de verosimilitud	4,815	1	,028		
Estadístico exacto de Fisher				,041	,021
Asociación lineal por lineal	4,762	1	,029		
N de casos válidos	189				

a. Calculado sólo para una tabla de 2x2.

b. 0 casillas (,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 29,03.

Se encuentra asociación debido a que el test Chi cuadrado es significativa $p(0.029) < \alpha(0.05)$.

Como la tabla es de 2x2. Podemos estimar el riesgo.

Estimación de riesgo

	Valor	Intervalo de confianza al 95%	
		Inferior	Superior
Razón de las ventajas para Raza materna (blanca vs otras) (blanca / otras)	2,005	1,070	3,754
Para la cohorte Bajo peso al nacer = >=2500 gr	1,241	1,019	1,510
Para la cohorte Bajo peso al nacer = <2500 gr	,619	,399	,960
N de casos válidos	189		

Observando la primera fila: el intervalo [1.070 ; 3.754] no contiene al cero, por lo que concluimos que existe asociación significativa entre RAZAREC y BAJOPESO. Las madres de raza “blanca” tienen el doble de riesgo (OR = 2.005) de tener un RN de bajo peso que las madres de raza “Negra y otras razas”.

Si recurrimos a la Regresión Logística e introducimos la variable RAZA sin colapsar (con tres categorías), el programa convertirá automáticamente en dos variables dicotómicas dummys, para poder así calcular la OR de cada categoría frente a una de referencia. Veámoslo:

Codificaciones de variables categóricas

		Frecuencia	Codificación de (1)	Codificación de (2)
Raza de la madre	Blanca	96	,000	,000
	Negra	26	1,000	,000
	Otras	67	,000	1,000

Se crean dos variables nuevas: raza (1) y raza (2). La raza "Blanca" ha sido tomada como categoría de referencia (tiene valores ceros en ambas), ya que era la que tenía una codificación absoluta más baja en la variable original, por lo que raza (1) es una dicotómica en la que el valor "1" es "Negra" y raza (2) es una dicotómica en la que el valor "1" es "Otra raza". Y en la ecuación de Regresión Logística:

Variables en la ecuación

		B	E.T.	Wald	gl	Sig.	Exp(B)	I.C. 95,0% para EXP(B) Inferior	Superior
Paso 1[a]	raza			4,922	2	,085			
	raza(1)	,845	,463	3,323	1	,068	2,328	,939	5,772
	raza(2)	,636	,348	3,345	1	,067	1,889	,955	3,736
	Constante	-1,155	,239	23,330	1	,000	,315		

a. Variable(s) introducida(s) en el paso 1: raza.

Observamos que se introducen dos variables nuevas que llevan información desagregada de la variable original RAZA. De hecho, la variable original no tiene interpretación en la ecuación, está solo para indicarnos que de ella se han generado las dos dummys (aunque puede comprobarse que el estadístico de Wald tiene dos grados de libertad -hay tres categorías, aunque todas no significativas.

Variables explicativas nominales y ordinales

Cuando la variable explicativa es categórica, el modelo se construye considerando variables numéricas asociadas a la categórica, son las llamadas variables de diseño o auxiliares.

Cuando se tienen variables categóricas con más de dos categorías de respuesta, se construyen K – 1 variables de diseño. Existen diferentes formas de codificar esas variables de diseño, destacando los métodos parcial y marginal o, para

variables ordinales, utilizar una codificación que considere distancias equidistantes entre las categorías de respuesta.

Codificación de la variable dependiente.

1 : a la ocurrencia (éxito)

0 : a la ausencia *(fracaso)*

Codificación de las variables independientes.

Codificación parcial

En la codificación parcial se elige una categoría de referencia, de modo que todas las variables de diseño toman el valor 0 para dicha categoría. Para cada una de las categorías restantes, su variable de diseño toma el valor 1 para la categoría asociada y 0 para el resto. Esta forma de codificación suele venir implementada en los diversos programas estadísticos, aunque según el que se use, se toma como referencia la primera categoría o la última. Suponiendo que se tienen I categorías en una variable explicativa categórica A y que se ha utilizado el método de codificación parcial asignando el valor 0 para la categoría l, el valor para las variables de diseño m-ésima asociada a la categoría A_m sería

$$X_{im}^{A} = X_{m}^{A}/(A = A_i) = \begin{cases} 1 & i = m \\ 0 & i \neq m \end{cases} \forall m = 2, \dots, I; i = 1 \dots . I$$

Caso dicotómico. Se codifica como 1, si en caso se cree que favorece la ocurrencia del evento. Se codifica como 0, en caso contrario. Ejemplo:

Riesgo	Z_1
Expuesto	1
No expuesto	0

Codificación marginal

En este método, las variables de diseño toman el valor 1 para su categoría asociada y el valor 0 para las restantes, excepto para la categoría de referencia que toma el valor -1. La codificación sería, suponiendo la primera categoría como la de referencia.

$$X_{im}^{A} = X_{m}^{A} \mid (A = A_i) = \begin{cases} 1 & i = m \\ -1 & i = 1 \\ 0 & i \neq m, 1 \end{cases} \quad \forall m = 2 \dots, I \quad i = 1 \dots I$$

En la regresión logística se utiliza mayoritariamente el método de codificación parcial, debido a que facilita la interpretación en términos de cocientes de ventajas. Otro motivo por el que usar este tipo de codificación, se debe al uso de la regresión logística en epidemiología y en diseño de experimentos, dónde es usual tener un grupo de control no expuesto al tratamiento y con el cuál se quieren comparar los otros grupos.

Una vez que se han codificado las variables categóricas, el modelo se reduce al caso de regresión logística simple si lo que se tiene es una sola variable explicativa que tenga sólo dos categorías, o al modelo de regresión logística si se tienen más variables explicativas o que se tenga sólo una, pero con tres o más categorías.

Caso categórico. Cuando la variable tomas más de dos posibles categorías, puede usarse variables indicadoras dummy. La solución al problema es crear tantas variables dicotómicas como número de respuestas - 1.

Ejemplo: Si la variable en cuestión recoge datos de tabaquismo con las siguientes respuestas:

- Nunca fumó
- Ex-fumador
- Actualmente fuma menos de 10 cigarrillos diarios
- Actualmente fuma 10 o más cigarrillos diarios

Tenemos 4 posibles respuestas por lo que construiremos 3 variables internas dicotómicas, existiendo diferentes posibilidades de codificación, que conducen a diferentes interpretaciones, y siendo la más habitual la siguiente:

Tabaquismo	Z1	Z2	Z3
Nunca fumo	0	0	0
Ex fumador	1	0	0
Menos de 10 cigarrillos diarios	0	1	0
10 a más cigarrillos diarios	0	0	1

En este tipo de codificación el coeficiente de la ecuación de regresión para cada variable diseño (siempre transformado con la función exponencial), se corresponde al odds ratio de esa categoría con respecto al nivel de referencia (la

primera respuesta), en nuestro ejemplo cuantifica cómo cambia el riesgo respecto a no haber fumado nunca.

Existen otras posibilidades entre las que se destaca con un ejemplo para una variable cualitativa de tres respuestas:

	I1	I2
Respuesta 1	0	0
Respuesta 2	1	0
Respuesta 3	1	1

Con esta codificación cada coeficiente se interpreta como una media del cambio del riesgo al pasar de una categoría a la siguiente.

En el caso una categoría que NO pueda ser considerada de forma natural como nivel de referencia, como por ejemplo el grupo sanguíneo, un posible sistema de clasificación es:

	I1	I2
Respuesta 1	-1	-1
Respuesta 2	1	0
Respuesta 3	0	1

donde cada coeficiente de las variables indicadoras tiene una interpretación directa como cambio en el riesgo con respecto a la media de las tres respuestas.

Codificación de variables ordinales

Cuando se tienen variables explicativas ordinales, se pueden tratar como si fueran nominales y codificarlas por alguno de los métodos anteriores.

Otra forma de codificarlas es asignar puntuaciones monótonas a cada categoría, de forma que conserven el orden. Normalmente se consideran puntuaciones equidistantes entre categorías. Si se codifican de esta forma, las variables se incluyen en el modelo como variables cuantitativas cuyos valores serán los códigos asignados.

En el capítulo 4 de (Fox and Weisberg, 2011) sección 4.6, se realiza un análisis exhaustivo de otras formas de codificación, con un apartado específico sobre las diferentes formas

de codificar variables ordinales, incluyendo el uso de polinomios ortogonales, utilizado sobre todo en análisis de la varianza.

Caso de variable numérica: pueden darse dos situaciones:
Si creemos que por cada unidad que aumente la variable, la OR aumenta en un factor multiplicativo constante, podemos usar la variable tal cual en el modelo. Si tenemos dudas de que esto sea así, o no sepamos ni siquiera lo que significa la frase anterior, mejor olvidamos esta posibilidad y consideramos la siguiente; Si creemos que la variable numérica puede afectar a la respuesta, pero no tenemos muy claro de qué manera, podemos —categorizar la variable. Esto consiste por ejemplo en estratificar la variable en valores pequeños, medianos y grandes. Los puntos de corte los podemos elegir nosotros manualmente, o usar cortes automáticos basados en que cada categoría tenga el mismo número de observaciones (usando percentiles).

Ejemplo: el responsable de marketing de la entidad financiera considera que la predisposición de sus clientes a adquirir el nuevo producto depende no solo de sus ingresos, sino de otras variables como la edad, grado de confianza en el sistema público de pensiones y del hecho de ser propietario o no de su vivienda.

Medidas originales:

INGRESOS Miles de soles
EDAD: menos de 40 años
40 – 60 años
Más de 60 años
CONFIANZA: En una escala de 1(ninguna confianza) a 9 (plena confianza).
VIVIENDA: propietario
no propietario

Existen dos variables cualitativas: EDAD Y VIVIENDA.

Vivienda	VIVIENDA1
Propietario	1
No propietario	0

Para la variable EDAD, es necesario utilizar variables ficticias (se crea k-1 variables: 3-1=2).

EDAD1: 1: cuando el individuo tenga de 40 a 60 años

0: en caso contrario

EDAD2: 1: cuando el individuo tenga mayor de 60 años

0: cualquier otra situación

Edad	EDAD1	EDAD2
Menos de 40 años	0	0
De 40 a 60 años	1	0
Mas de 60 años	0	1

Ejemplo:

Se realiza un estudio para considerar si existe relación en que un estudiante asista a clases de repaso de lectura (asista = 1, no asista = 0), tomando como variables predictoras la nota que obtiene en un examen de lectura estándar (realizado antes de iniciar las clases de repaso), un examen de letras (antes de iniciar las clases de repaso) y el sexo (hombre = 1, mujer = 0). Obtenga un modelo para predecir la probabilidad de que un estudiante tenga que asistir a clases de repaso.

Data: clases de repaso.

```
#### REGRESION LOGISTICA MULTIPLE ####
# Ejemplo clases de repaso
datos <- read.csv("clases de repaso.csv", head=T,sep=",")
str(datos)
```

```
## 'data.frame':    189 obs. of  4 variables:
##  $ clases_repaso   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sexo            : int  1 1 0 0 0 1 0 1 0 0 ...
##  $ examen_lectura: int  91 60 70 54 50 62 59 60 45 60 ...
##  $ examen_letras   : int  87 78 60 54 60 58 55 49 65 60 ...
```

La data consta de 189 observaciones y 4 variables, todas reconocidas como cuantitativas, es necesario convertir las variables clases de repaso y sexo en cualitativas.

```
# configurando como factor las variables cualitativas clase de repaso y sexo
datos$clases_repaso <- as.factor(datos$clases_repaso)
datos$sexo <- as.factor(datos$sexo)
str(datos)
```

```
## 'data.frame':    189 obs. of  4 variables:
##  $ clases_repaso    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sexo             : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 1 2 1 1 ...
##  $ examen_lectura : int  91 60 70 54 50 62 59 60 45 60 ...
##  $ examen_letras   : int  87 78 60 54 60 58 55 49 65 60 ...
```

La salida muestra que la base de datos tiene 189 observaciones y 4 variables, de las cuales, sexo y clases repaso están reconocidas como factor (cualitativas) y examen_lectura y examen_letras como variables numéricas discretas.

a. Análisis de las observaciones

Etiquetando las variables cualitativas

```
# Análisis de las observaciones
# etiquetando las categorías para efectos de obtener tablas
datos$clases_repaso <- factor(datos$clases_repaso,
                              levels = c("0","1"),labels=c("no asista", "asista"))
datos$sexo <- factor(datos$sexo,levels = c("0","1"),labels=c("Mujer", "Hombre"))
str(datos)
```

```
         ## 'data.frame':    189 obs. of  4 variables:
##  $ clases_repaso    : Factor w/ 2 levels "no asista","asista": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sexo             : Factor w/ 2 levels "Mujer","Hombre": 2 2 1 1 1 2 1 2 1 1 ...
##  $ examen_lectura : int  91 60 70 54 50 62 59 60 45 60 ...
##  $ examen_letras   : int  87 78 60 54 60 58 55 49 65 60 ...
```

Las tablas de frecuencias, así como representaciones gráficas de las observaciones son útiles para intuir si las variables independientes escogidas están relacionadas con la variable respuesta y por lo tanto ser buenos predictores.

Analizando la variable de clasificación o dependiente con tablas y figuras.

```
# Analizando la variable de clasificación

require(ggplot2)

require(gridExtra)

require(tidyverse)
```

```
datos %>%

  group_by(datos$clases_repaso) %>%

  summarise( numero_casos = n(),
```

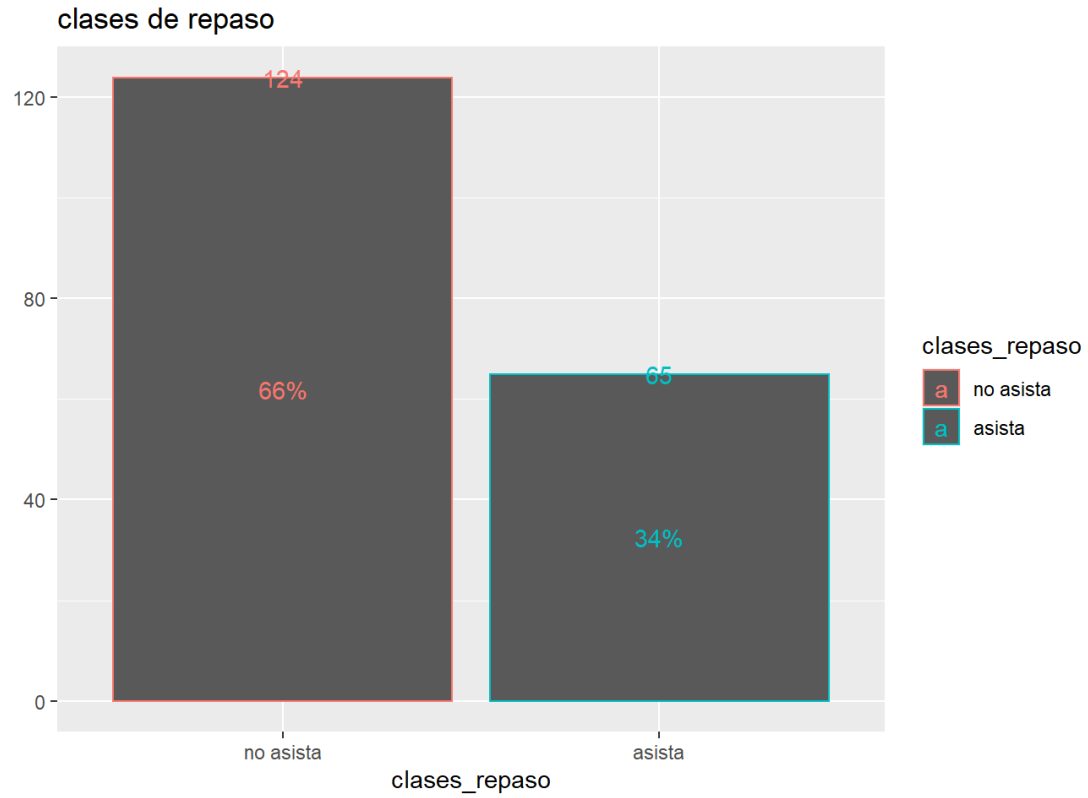
```
              porcentaje = numero_casos / nrow(datos)
  )
```

```
## # A tibble: 2 x 3
##   `datos$clases_repaso` numero_casos porcentaje
##   <fct>                        <int>      <dbl>
## 1 no asista                      124      0.656
## 2 asista                          65      0.344
```

El 65.6% no asiste a clases de repaso de lectura y el 34% si asisten a clases de repaso de lectura. No es necesario balancear los datos las categorías.

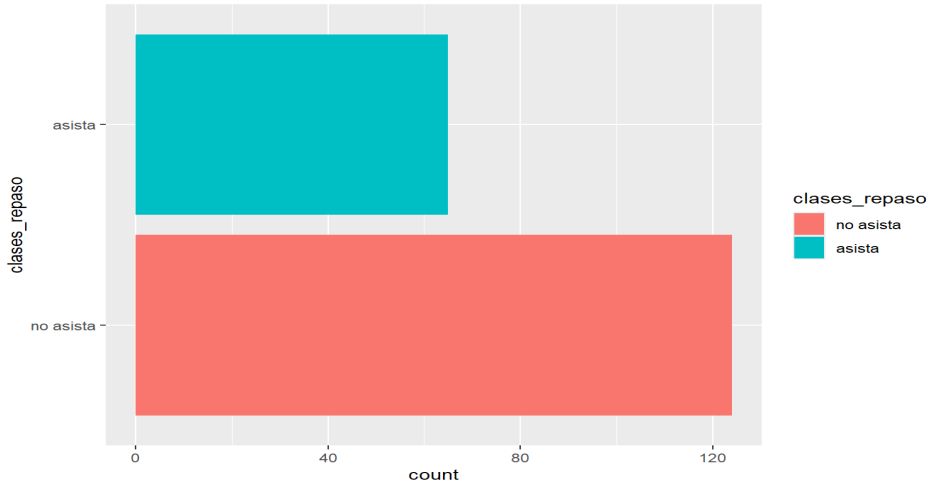
Graficamante,

```
qplot(clases_repaso, data = datos, col = clases_repaso,
      main = "clases de repaso",geom = "bar") +
  geom_text(aes(label=scales::percent(..count../sum(..count..))),
            stat='count',position=position_stack(0.5))+
  geom_text(aes(label=..count..),
            stat="count",position=position_stack())
```



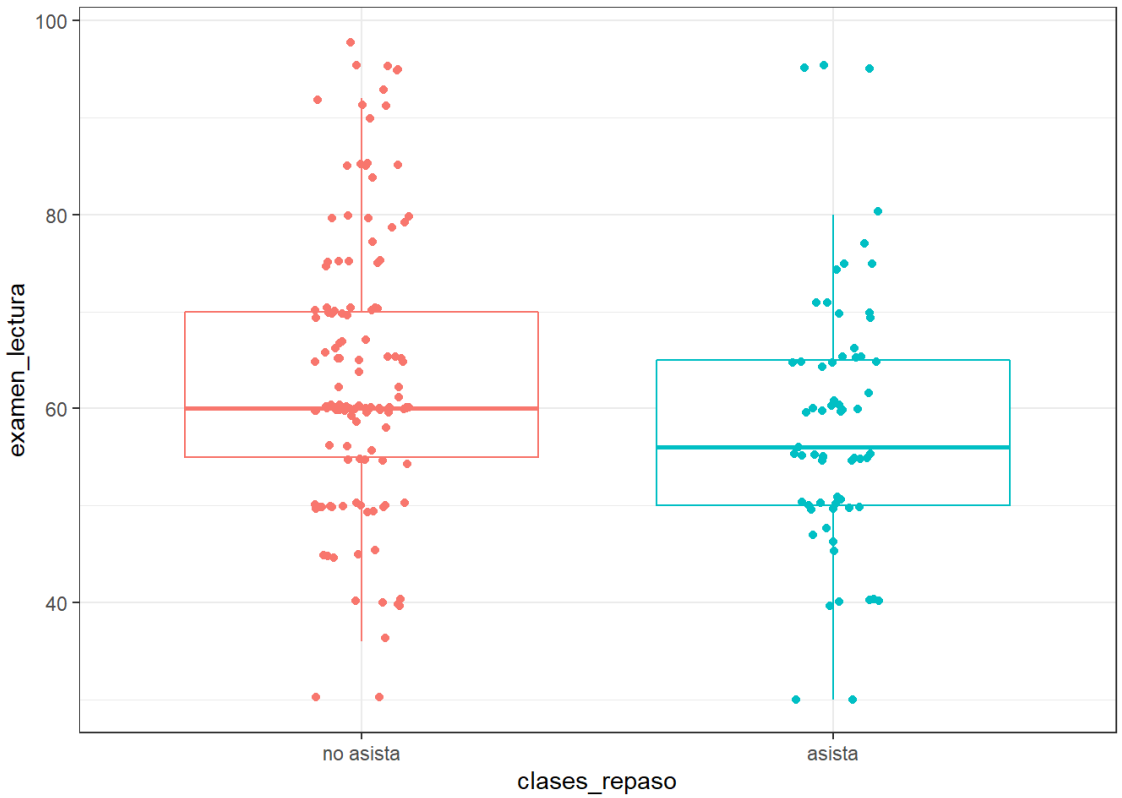
Vemos otro tipo de figura,

```
# otro Plot
ggplot(datos, aes(clases_repaso, fill = clases_repaso)) +
  geom_bar() +
  coord_flip()
```



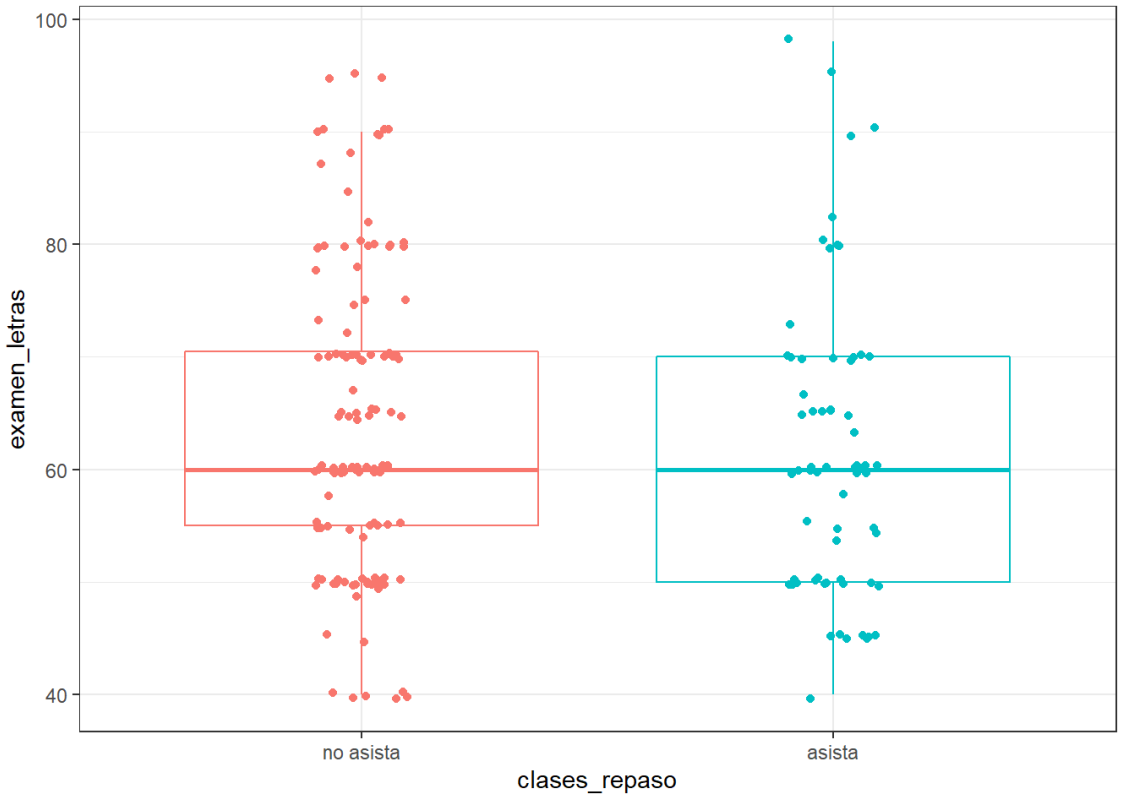
Veamos el comportamiento de la variable independiente examen de lectura frente a la dependiente clases de repaso

```
# observando las variables independientes frente a la dependiente
ggplot(data = datos, aes(x = clases_repaso, y =examen_lectura, color=clases_repaso)) +
  geom_boxplot(outlier.shape=NA) +
  geom_jitter (width=0.1) +
  theme_bw() +
  theme(legend.position = "null")
```



El promedio de los estudiantes que necesitan asistir a las clases de repaso según el examen de lectura es menor al promedio de los que no necesitan asistir, las varianzas son casi iguales, existen estudiantes con puntajes bajos que necesitan asistir a las clases y estudiantes con puntajes altos que no necesitarían asistir a las clases de repaso.

```
ggplot(data = datos, aes(x = clases_repaso, y =examen_letras, color=clases_repaso)) +
  geom_boxplot(outlier.shape=NA) +
  geom_jitter (width=0.1) +
  theme_bw() +
  theme(legend.position = "null")
```

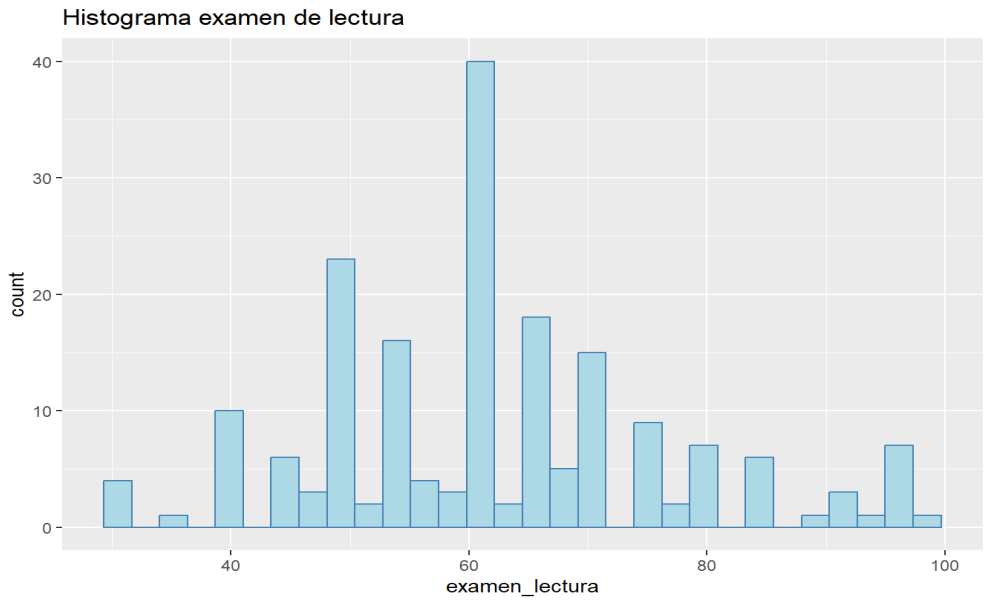


El promedio de los estudiantes que necesitan asistir a las clases de repaso según el examen de letras es similar al promedio de los que no necesitan asistir, la varianza de los que necesitan asistir es mayor a la varianza de los que no necesitan asistir, existen estudiantes con puntajes bajos que necesitan asistir a las clases y estudiantes con puntajes altos que no necesitarían asistir a las clases de repaso.

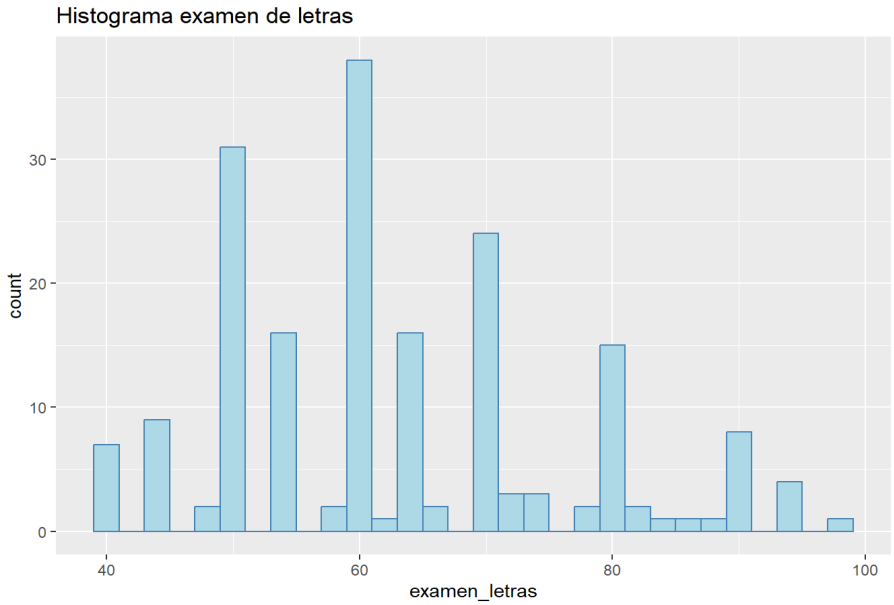
Existe mayor simetría en los que asisten a clases y se observa asimetría en los que no asisten según la variable examen de letras.

Podemos representar también los datos utilizando histogramas de frecuencia por variables.

```
# observando las variables independientes
# Histograma de variable cuantitativa
ggplot(datos, aes(x=examen_lectura))+
  geom_histogram(color="steelblue",
  fill="lightblue")+
  ggtitle("Histograma examen de lestras")
```




```
ggplot(datos, aes(x=examen_letras))+
  geom_histogram(color="steelblue",
                 fill="lightblue")+
  ggtitle("Histograma examen de letras")
```




```
# tablas bidimensionales
# clases de repaso y sexo
tabla <- table(datos$clases_repaso, datos$sexo,
```

```
                dnn=c("clases de repaso", "sexo"))
addmargins(tabla)
```

```
##                         sexo
## clases de repaso    Mujer Hombre Sum
##         no asista      71     53  124
##         asista         25     40   65
##         Sum            96     93  189
```

```
library(sjPlot)
```

```
tab_xtab(var.row = datos$clases_repaso,datos$sexo,show.cell.prc = T,sh
ow.summary = F)
```

clases_repaso	*sexo*		***Total***
	Mujer	Hombre	
no asista	71 37.6 %	53 28 %	124 65.6 %
asista	25 13.2 %	40 21.2 %	65 34.4 %
Total	96 50.8 %	93 49.2 %	189 100 %

```
# clases de repaso y examen_lectura
with(datos, prop.table(table(cut(examen_lectura, 5),
                             clases_repaso), 1))
```

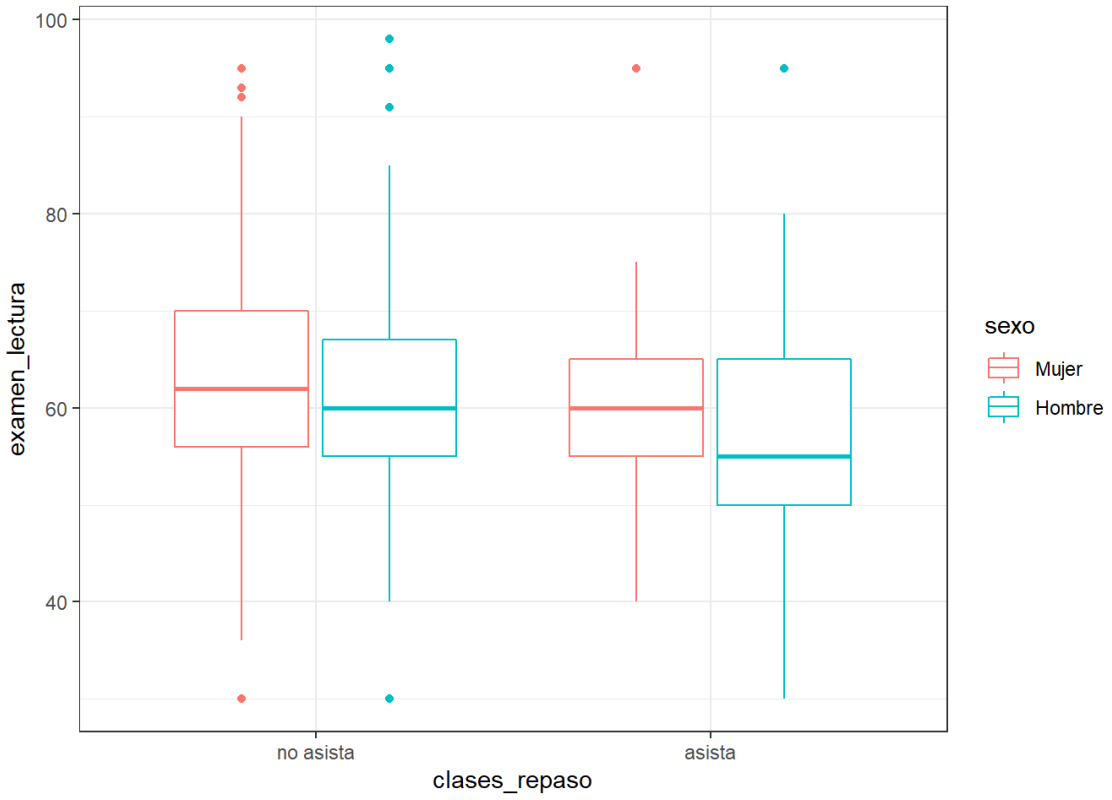
```
##                clases_repaso
##                 no asista    asista
##    (29.9,43.6] 0.5333333 0.4666667
##    (43.6,57.2] 0.5185185 0.4814815
##    (57.2,70.8] 0.7283951 0.2716049
```

```
##    (70.8,84.4] 0.6666667 0.3333333
##    (84.4,98.1] 0.8333333 0.1666667
```

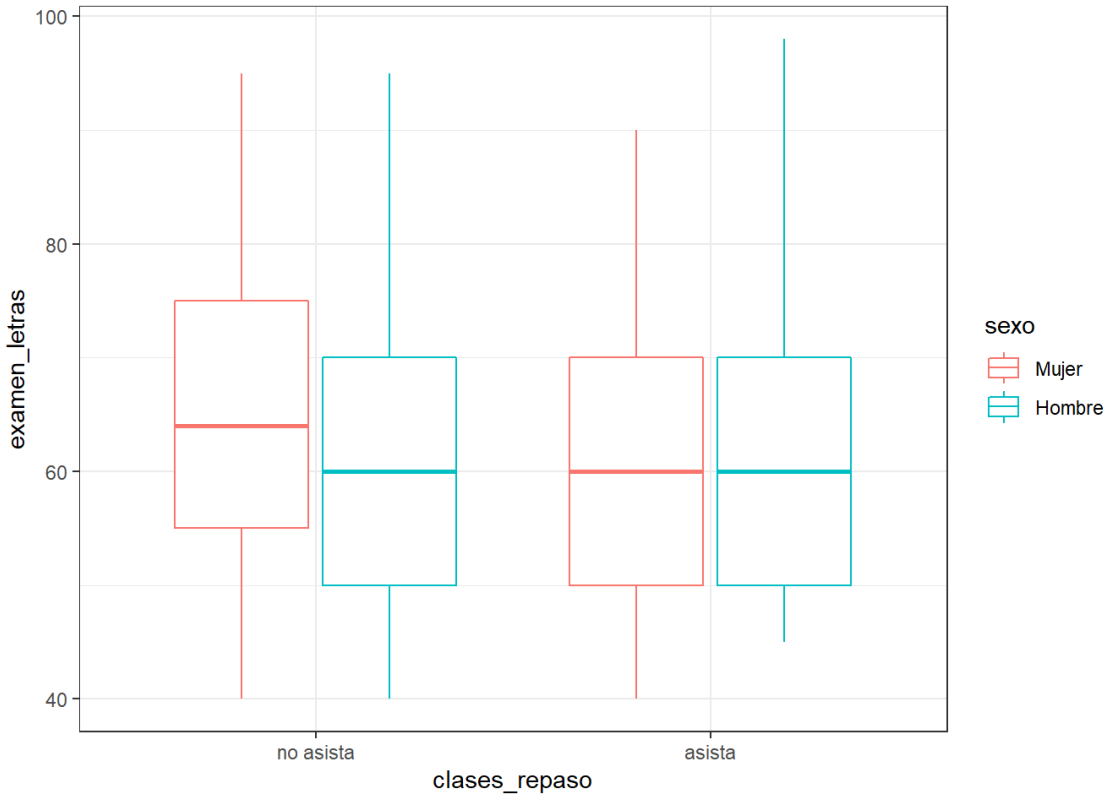
```
# clases de repaso y examen_letras
with(datos, prop.table(table(cut(examen_letras, 5),
                             clases_repaso), 1))
```

```
##              clases_repaso
##               no asista    asista
##    (39.9,51.6] 0.5918367 0.4081633
##    (51.6,63.2] 0.6491228 0.3508772
##    (63.2,74.8] 0.6444444 0.3555556
##    (74.8,86.4] 0.7826087 0.2173913
##    (86.4,98.1] 0.7333333 0.2666667
```

```
# al ser sexo una variable cualitativa, puede observarse como control
# observando la variable repaso, segun sexo y examen lectura
require(ggplot2)
ggplot(data = datos, mapping = aes(x = clases_repaso,
                                   y = examen_lectura,
                                   colour = sexo)) +
  geom_boxplot() +
  theme_bw()
```




```
# observando la variable repaso, segun sexo y examen letras
require(ggplot2)
ggplot(data = datos, mapping = aes(x = clases_repaso,
                                   y = examen_letras,
                                   colour = sexo)) +
  geom_boxplot() +
  theme_bw()
```



b. Generar el modelo de regresión logística mediante glm

```
# Generar el modelo de regresion logistica
modelo <- glm(clases_repaso ~ examen_lectura + sexo + examen_letras, d
ata = datos,
              family = "binomial")
summary(modelo)
```

```
## Call:
## glm(formula = clases_repaso ~ examen_lectura + sexo +examen_letras,
##     family = "binomial", data = datos)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3727  -0.9261  -0.7278   1.2390   1.9671
## Coefficients:
##                 Estimate  Std.Error z value Pr(>|z|)
## (Intercept)      0.07707    0.81788   0.094   0.9249
## examen_lectura -0.04017    0.01997  -2.012   0.0443 *
## sexoHombre       0.70495    0.31824   2.215   0.0268 *
```

```
## examen_letras    0.02178    0.02111   1.032    0.3021
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 243.28  on 188  degrees of freedom
## Residual deviance: 231.81  on 185  degrees of freedom
## AIC: 239.81
## Number of Fisher Scoring iterations: 4
```

$$Ln\left[\frac{P(exito}{P(fracaso)}\right] = logit$$
$$= 0.07707 - 0.04017\text{examen\_lectura} + 0.70495\text{sexohombre} + 0.02178\text{examen\_letras}$$

Acorde al modelo, el *logaritmo de odds* de que un estudiante necesite clases de repaso esta negativamente relacionado con la puntuación obtenida en el examen de lectura (coeficiente parcial = -0.04017), siendo significativa esta relación (*p-value* = 0.0443). 0.04017 es el cambio esperado en el logit al aumentar una unidad en el examen de lectura manteniendo estables el resto de las variables en el modelo. (El coeficiente de la variable examen_lectura (-0.04017) nos está indicando que el logaritmo del odds ratio de que asista a las clases de repaso se asocia a que en promedio **disminuya** en 0.04017 unidades por cada unidad que aumenta la puntuación en el examen de lectura.)

El logaritmo de odss de que un estudiante necesite clases de repaso esta positivamente relacionado con la puntuación obtenida en el examen de letras (coeficiente parcial = 0.02178), siendo no significativa esta relación (*p-value* = 0.3021). 0.02178 es el cambio esperado en el logit al aumentar una unidad el examen de letras manteniendo estables el resto de las variables en el modelo. (La variable examen_letras indica que el logaritmo del odds ratio de que asista a las clases de repaso aumenta en 0.02178 unidades por cada unidad que **aumenta** la puntuación en el examen de letras).

Existe una relación significativa positiva entre el *logaritmo de odds* de necesitar clases de repaso y el género del estudiante (*p-value* = 0.0268). (La variable sexo (0.70495) nos indica que el logaritmo del odds ratio de que asista a las clases de repaso siendo varón **aumenta** en 0.70495 o en concreto los odds de que un hombre requiera clases de repaso es 0.04017 veces mayores que los de las mujeres.)

$$Odds = \frac{P(exito)}{P(fracaso)} = e^{logit}$$
$$= e^{0.07707 - 0.04017\text{examen\_lectura} + 0.70495\text{sexohombre} + 0.02178\text{examen\_letras}}$$

$$P(exito) = \frac{e^{logit}}{1 + e^{logit}}$$
$$= \frac{e^{0.07707 - 0.04017\text{examen\_lectura} + 0.70495\text{sexohombre} + 0.02178\text{examen\_letra}}}{1 + e^{0.07707 - 0.04017\text{examen\_lectura} + 0.70495\text{sexohombre} + 0.02178\text{examen\_letra}}}$$

```
#variables significativas
sig.var <- summary(modelo)$coeff[-1,4] < 0.05
sig.var
```

```
## examen_lectura     sexoHombre  examen_letras
##           TRUE           TRUE          FALSE
```

Interpretación de los coeficientes exponenciando:

```
# exponenciando los coeficientes
round(exp(coefficients(modelo)),6)
```

```
##    (Intercept) examen_lectura     sexoHombre  examen_letras
##       1.080120       0.960624       2.023742       1.022021
```

Lo que nos viene a decir que aumentar el examen de lectura un punto, aumenta un 0.96 las posibilidades de **no necesitar** clases de repaso, mientras que aumentar un punto en el examen _letras aumenta la posibilidad de **necesitar** examen de repaso en casi un 2.024, finalmente, la posibilidad de que un hombre requiera clases de repaso es 2.014 veces más que las mujeres.

Además del valor estimado de los coeficientes parciales de correlación calculados por el modelo, es conveniente generar sus correspondientes intervalos de confianza. En el caso de regresión logística, estos intervalos suelen calcularse basados en profile likelihood (en R es el método por defecto si se tiene instalado el paquete MASS).

c. Comparación de modelos mediante anova

La función anova permite comparar modelos anidados. Cuando se usa un solo modelo se determina la significatividad de cada término añadido.

```
# mediante anova
```

```
anova(modelo,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: clases_repaso
## Terms added sequentially (first to last)
##
##                Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                             188     243.28
## examen_lectura  1   5.3988       187     237.88  0.02015 *
## sexo            1   4.9959       186     232.89  0.02541 *
## examen_letras   1   1.0745       185     231.81  0.29993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Se determina la significatividad de cada estimador $p < \alpha$ al 5% de significancia.

```
# modelo excluyendo la variable no significativa (examen letras)
modelo_final<- glm(clases_repaso ~ examen_lectura + sexo, data = datos
,
                   family = "binomial")
summary(modelo_final)
```

```
## Call:
## glm(formula = clases_repaso ~ examen_lectura + sexo, family = "bino
mial",
##     data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3607  -0.9225  -0.7416   1.2484   1.9795
##
## Coefficients:
```

```
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.42158    0.74834   0.563   0.5732
## examen_lectura -0.02346    0.01157  -2.028   0.0426 *
## sexoHombre       0.70337    0.31727   2.217   0.0266 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 243.28  on 188  degrees of freedom
## Residual deviance: 232.89  on 186  degrees of freedom
## AIC: 238.89
##
## Number of Fisher Scoring iterations: 4
```

```
# exponenciando
round(exp(coefficients(modelo_final)),6)
```

```
##    (Intercept) examen_lectura     sexoHombre
##       1.524367       0.976812       2.020542
```

```
confint(modelo_final) # intervalo de confianza para los estimadores
```

```
#                      2.5 %       97.5 %
## (Intercept)    -1.03459248  1.914801524
## examen_lectura -0.04693827 -0.001331319
## sexoHombre       0.08621040  1.333398832
```

```
# Tasa de ventajas e IC 95%
round(exp(cbind(OR = coef(modelo_final), confint(modelo_final, level=0
.95))),5)
```

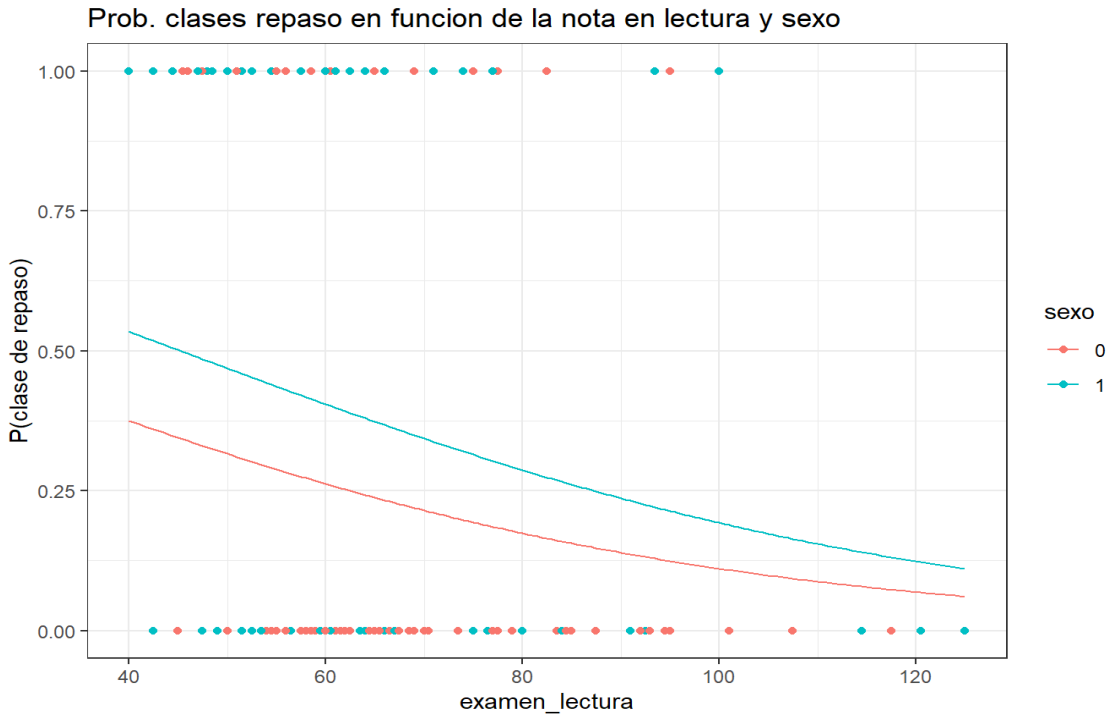
```
##                       OR   2.5 %  97.5 %
## (Intercept)     1.52437 0.35537 6.78559
## examen_lectura 0.97681 0.95415 0.99867
## sexoHombre      2.02054 1.09004 3.79392
```

d. Representación gráfica del modelo

Al tratarse de un modelo con 2 predictores, no se puede obtener una representación en 2D en la que se incluyan ambos predictores a la vez. Sí es posible representar la curva del modelo logístico cuando se mantiene constante uno de los dos predictores. Por ejemplo, al representar las predicciones del modelo diferenciando entre hombres y mujeres (fijando el valor del predictor sexo) se aprecia que la curva de los hombres (sexo=1) siempre está por encima. Esto se debe a que, como indica el coeficiente parcial de correlación del predictor sexo, para una misma nota en el examen de lectura el *logaritmo de ODDs* de necesitar clases de repaso es 0.64749 veces mayor en hombres.

```
# Representacion grafica del modelo
require(ggplot2)
# Para graficar los valores en ggplot junto con la curva, la variable
# respuesta tiene que ser numerica en lugar de factor.
datos$clases_repaso <- as.numeric(as.character(datos$clases_repaso))

# Se crea un dataframe que contenga la probabilidad de que se necesiten
# clases de repaso dada una determinada nota en el examen de lectura y
# siendo hombre (sex=1).

# Vector con nuevos valores interpolados en el rango de observaciones
nuevos_valores_examen <- seq(from = min(datos$examen_lectura),
                             to = max(datos$examen_lectura), by = 0.5)
sexo <- as.factor(rep(x = 1, length(nuevos_valores_examen)))
# Predicciones de los nuevos puntos segun el modelo. type = 'response'
# devuelve las predicciones en forma de probabilidad en lugar de en log_ODDs
predicciones <- predict(object = modelo, newdata = data.frame(examen_lectura =
                                                    nuevos_valores_examen, sexo = sexo), type =
"response")
# Se crea un data frame con los nuevos puntos y sus predicciones para
# graficar la curva
```

```
datos_curva_hombre <- data.frame(examen_lectura = nuevos_valores_examen,
                                 sexo = sexo, clases_repaso = predicciones)
# Mismo proceso para mujeres (sexo = 0)
nuevos_valores_examen <- seq(from = min(datos$examen_lectura),
                             to = max(datos$examen_lectura), by = 0.5)
sexo <- as.factor(rep(x = 0, length(nuevos_valores_examen)))
predicciones <- predict(object = modelo, newdata = data.frame(examen_lectura =
                                                    nuevos_valores_examen, sexo = sexo), type =
"response")
datos_curva_mujer <- data.frame(examen_lectura = nuevos_valores_examen,
                                sexo = sexo,clases_repaso = predicciones)

# Se unifican los dos dataframe
datos_curva <- rbind(datos_curva_hombre, datos_curva_mujer)
ggplot(data = datos, mapping = aes(x = examen_lectura, y =
                                     as.numeric(clases_repaso),color = sexo)) +
  geom_point() +
  geom_line(data = datos_curva, aes(y = clases_repaso)) +
  geom_line(data = datos_curva, aes(y = clases_repaso)) +
  theme_bw() +
  labs(title = "Prob. clases repaso en funcion de la nota en lectura y sexo",
       y = "P(clase de repaso)")
```



La probabilidad de que los hombres asistan a clases de repaso es mayor que la probabilidad de que mujeres asistan a las clases de repaso.

```
# Otra opcion para graficarlo es: qplot(x = modelo$data$examen_lectura, y =
# modelo$fitted.values, geom = c('point','line'), colour = modelo$data$sexo,
# ylim = c(0,1))
```

Interacciones:

En algunas investigaciones pueda que sea necesario ver las interacciones

```
# modelo con interaccion
modelo_con_interaccion <- glm(clases_repaso ~ examen_lectura + sexo +
                                examen_lectura*sexo,
                              data = datos, family = binomial)
summary(modelo_con_interaccion )
```

```
## Call:
## glm(formula = clases_repaso ~ examen_lectura + sexo + examen_lectur
a *
##     sexo, family = binomial, data = datos)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3824  -0.9127  -0.7376   1.2452   1.9470
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.284600   1.112457   0.256    0.798
## examen_lectura             -0.021234   0.017652  -1.203    0.229
## sexoHombre                  0.938846   1.453931   0.646    0.518
## examen_lectura:sexoHombre -0.003883   0.023384  -0.166    0.868
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 243.28  on 188  degrees of freedom
## Residual deviance: 232.86  on 185  degrees of freedom
## AIC: 240.86
##
## Number of Fisher Scoring iterations: 4
```

Podemos observar que la interacción no es importante $p(0.787) > \alpha(0.05)$ y que además hace que todos los estimadores sean no significativos, **nos quedamos con el modelo sin interacción.**

e. Evaluación del modelo

```
# EVALUACION DEL MODELO (Al modelo final)
dif_residuos <- modelo_final$null.deviance - modelo_final$deviance
# Grados libertad
df <- modelo_final$df.null - modelo_final$df.residual
p_value <- pchisq(q = dif_residuos, df = df, lower.tail = FALSE)
paste("Diferencia de residuos: (chi)", round(dif_residuos, 4))
```

```
## [1] "Diferencia de residuos: (chi) 10.3946"
```

```
paste("p-value:", p_value)
```

```
## [1] "p-value: 0.00553136092344261"
```

```
# Bondad de ajuste del modelo (Hosmer-Lemeshow)
library(ResourceSelection)
```

```
hoslem.test(datos$clases_repaso, fitted(modelo_final))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  datos$clases_repaso, fitted(modelo_final)
## X-squared = 189, df = 8, p-value < 2.2e-16
```

```
# Pseudo R2 de McFadden
(RsqrMcFadden <- 1 - modelo_final$deviance/modelo_final$null.deviance)
```

```
## [1] 0.04272704
```

```
# Pseudo R2 de Cox y Snell
LR <- modelo_final$null.deviance - modelo_final$deviance
N <- sum(weights(modelo_final))
(RsqrCN <- 1 - exp(-LR/N))
```

```
## [1] 0.05351306
```

```
# Pseudo R2de Nagelkerke
```

```
L0.adj <- exp(-modelo_final$null.deviance/N)
(RsqrNal <- RsqrCN/(1 - L0.adj))
```

```
## [1] 0.07391751
```

La bondad de ajuste global del modelo se evalúa mediante la devianza (-2 veces el logaritmo de la verosimilitud) y tenemos que valores grandes indican que los modelos estadísticos son pobres.

En el resumen del modelo R calcula la devianza nula (solo con la constante) y la devianza residual (todo el modelo). Para que el modelo sea bueno la devianza residual debe ser menor que la devianza nula ya que valores más bajo de -2LL indican que el modelo predice la variable respuesta con mayor precisión

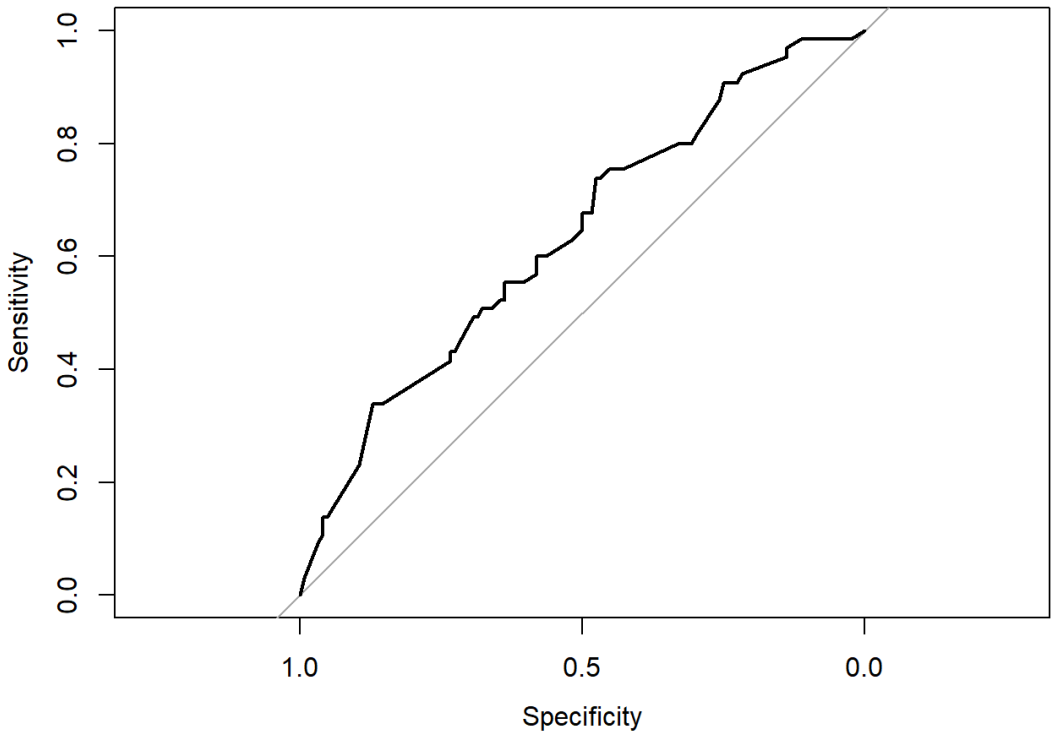
En este caso la devianza del modelo nulo es −2LL = 234.67, pero cuando añadimos Tratamientos este valor se reduce a 224.57, lo que nos dice que con esta variable el modelo mejora.

Curva ROC

```
# Curva ROC y Area bajo la curva
library(ROCR)
library(stats)
library(pROC)
```

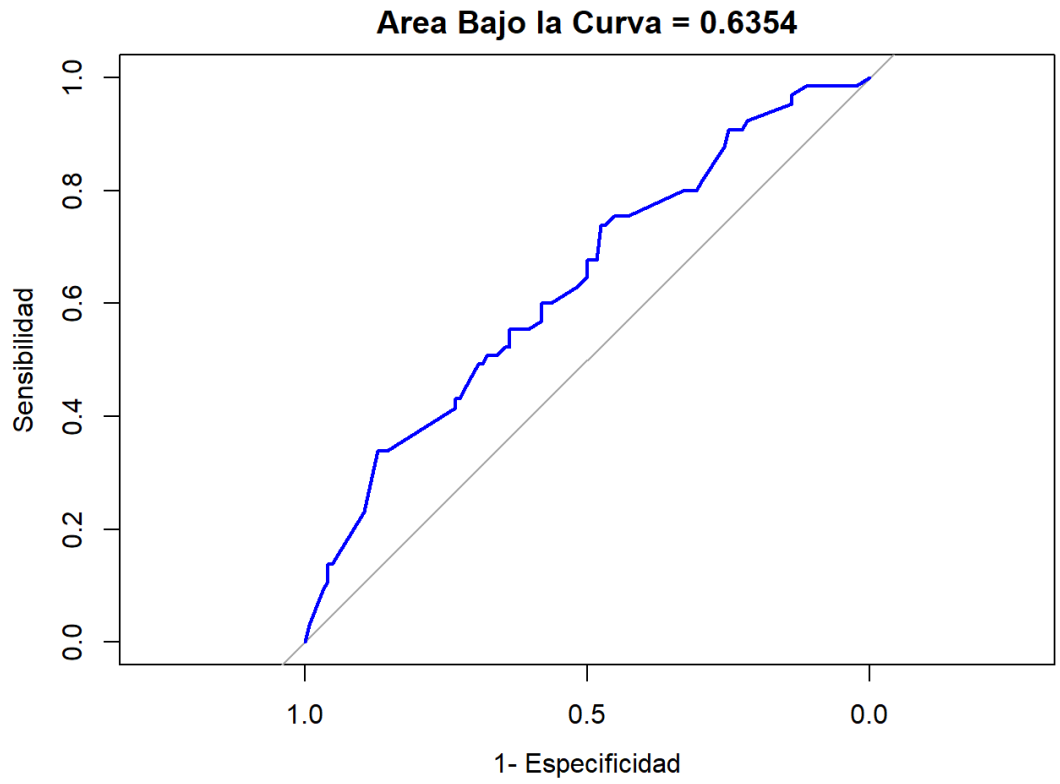
```
prob=predict(modelo_final,type="response")
g <- roc(clases_repaso ~ prob, datos)
```

```
plot(g)
```

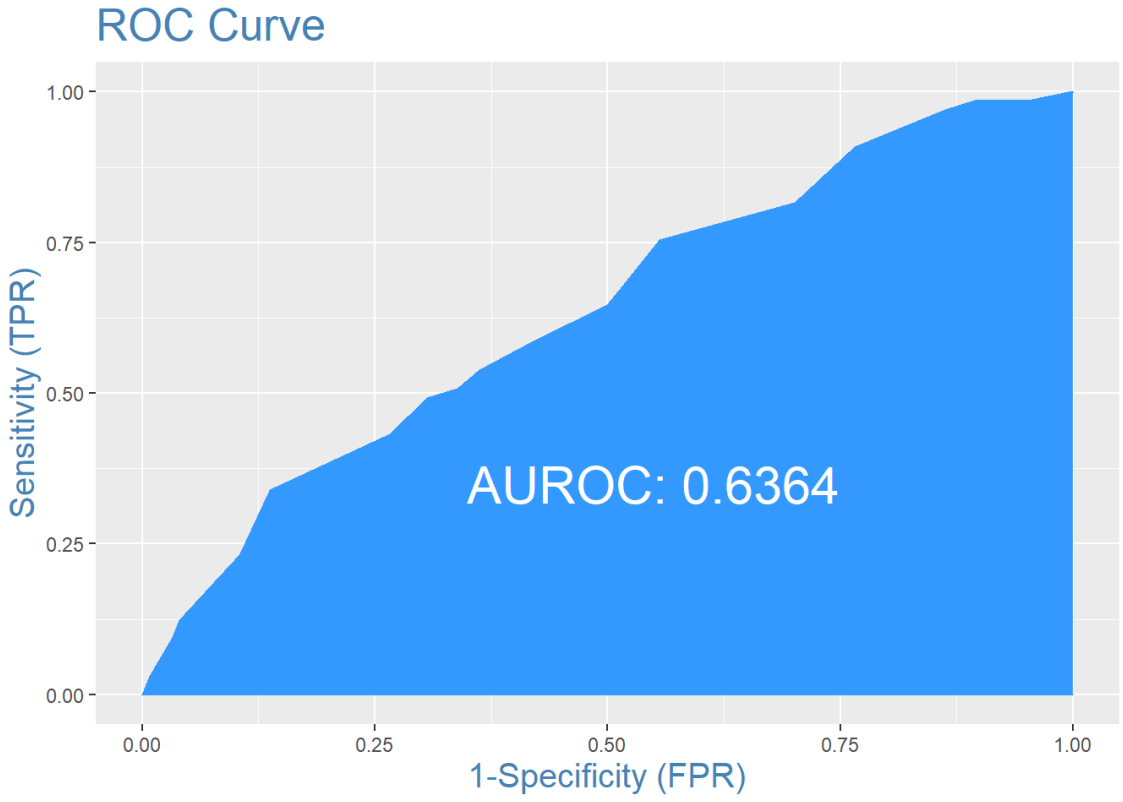



```
#Calculando area bajo la curva

areaROC<-auc(roc(datos$clases_repaso,prob))
```

```
ROC<-plot.roc(datos$clases_repaso,prob, xlab="1- Especificidad", ylab=
"Sensibilidad",

              main = paste('Area Bajo la Curva =',round(areaROC,4)), c
ol="blue")
```




```
library(InformationValue)

plotROC(sjlabelled::as_numeric(datos$clases_repaso,start.at = 0), prob
)
```



f. Comparación de las predicciones con las observaciones

Para este estudio se va a emplear un *threshold* de 0.5. Si la probabilidad predicha de asistir a clases de repaso es superior a 0.5 se asigna al nivel 1 (sí asiste), si es menor se asigna al nivel 0 (no clases de repaso).

```
#probabilidades y grupos estimados
```

```
prob=predict(modelo,type="response")
head(prob,14)
```

```
##         1         2         3         4         5         6         7
## 0.23183844 0.30055758 0.30198921 0.29377898 0.29650123 0.39198263 0.26737966
##         8         9        10        11        12        13        14
## 0.45904566 0.25475955 0.28038973 0.48512959 0.31448902 0.48512959 0.44607846
```

```
library(vcd)
```

```
predicciones <- ifelse(test = modelo$fitted.values > 0.5, yes = 1, no = 0)
head(predicciones,20)
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
matriz_confusion <- table(modelo$model$clases_repaso, predicciones,
                          dnn = c("observaciones", "predicciones"))
matriz_confusion
```

```
##                  predicciones
## observaciones    0    1
```

```
##         0       129  1
##         1        56  3
```

```
error1 <- sum(matriz_confusion[1,2], matriz_confusion[2,1])/sum(matriz
_confusion)
error1
```

```
## [1] 0.3280423
```

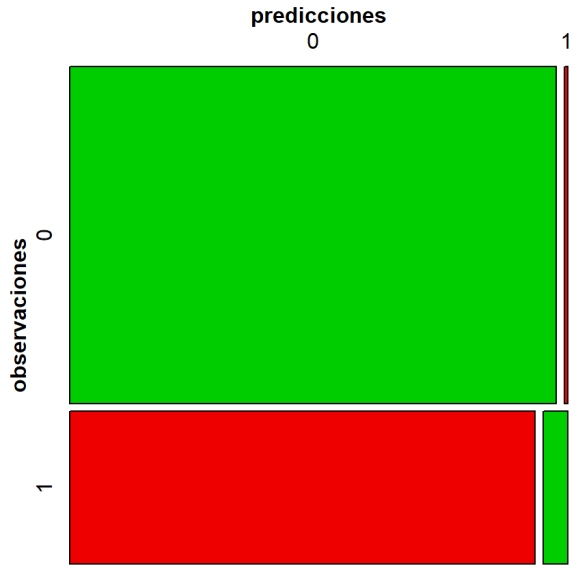
```
finaldata = cbind(datos, prob,predicciones)
head(finaldata,20)
```

```
##     sexo examen_lectura clases_repaso  prob     predicciones
## 1     1         91.0          0          0.23183844          0
## 2     1         77.5          0          0.30055758          0
## 3     0         52.5          0          0.30198921          0
## 4     0         54.0          0          0.29377898          0
## 5     0         53.5          0          0.29650123          0
## 6     1         62.0          0          0.39198263          0
## 7     0         59.0          0          0.26737966          0
## 8     1         51.5          0          0.45904566          0
## 9     0         61.5          0          0.25475955          0
## 10    0         56.5          0          0.28038973          0
## 11    1         47.5          0          0.48512959          0
## 12    1         75.0          0           0.31448902          0
## 13    1         47.5          0          0.48512959          0
## 14    1         53.5          0          0.44607846          0
## 15    0         50.0          0          0.31595703          0
## 16    0         50.0          0          0.31595703          0
## 17    1         49.0          0          0.47533094          0
## 18    0         59.0          0          0.26737966          0
## 19    1         60.0          0          0.40452571          0
```

```
## 20    0         60.0          0          0.26228430          0
```

```
write.csv(finaldata,"clases_repaso_predict.csv")
```

```
mosaic(matriz_confusion, shade = T, colorize = T,
       gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"), 2, 2)))
```



El modelo es capaz de clasificar correctamente (129+3)/(129+3+56+1)=0.698 (69.8%) de las observaciones cuando se emplea el *trainig data set*. Si se analiza en detalle cómo se distribuye el error, se aprecia que el modelo solo ha sido capaz de identificar correctamente a 3 de los 59 alumnos que realmente asisten a clases de repaso. El porcentaje de falsos negativos es muy alto. Seleccionar otro *threshold* puede mejorar la exactitud del modelo.

g. Conclusión

los resultados necesarios para interpretar son:

```
## [1] "p-value: 0.0066"
```

```
## Coefficients:
##                 Estimate Std. Error    z value   Pr(>|z|)
## (Intercept)      0.53616    0.81088   0.661     0.5085
```

```
## examen_lectura -0.02617     0.01223  -2.139     0.0324 *
## sexo1                0.64749     0.32484   1.993     0.0462 *
```

El modelo logístico creado para predecir la probabilidad de que un alumno tenga que asistir a clases de repaso a partir de la nota obtenida en un examen de lectura y el sexo del alumno es en conjunto significativo acorde al *Likelihood ratio* (*p-value* = 0.0066). El *p-value* de ambos predictores es significativo (examen_lectura = 0.0324, sexo1 = 0.0462). El ratio de error obtenido empleando las observaciones con las que se ha entrenado el modelo muestra un porcentaje de falsos negativos muy alto.

modelo logit:

$$ln\left(\frac{p(exito)}{p(fracaso)}\right) = logit = 0.53616 - 0.02617(examen\ lectura) + 0.64749(sexo1)$$

modelo Odds:

$$Odss = \frac{p(exito)}{p(fracaso)} = e^{logit} = e^{0.53616-0.02617(examen\ lectura)+0.64749(sexo1)}$$

modelo de probabilidad:

$$p(exito) = \frac{e^{logit}}{1+e^{logit}} = \frac{e^{0.53616-0.02617(examen\ lectura)+0.64749(sexo1)}}{1+\ e^{0.53616-0.02617(examen\ lectura)+0.64749(sexo1)}}$$

éxito = clases de repaso

h. **Interpretación** de los coeficientes del modelo:

coeficiente examen lectura:

Acorde al modelo, el *logaritmo de odds* de que un estudiante necesite clases de repaso esta negativamente relacionado con la puntuación obtenida en el examen de lectura (coeficiente parcial = -0.02617), siendo significativa esta relación (*p-value*(0.0324) < α(0.05)).

1. -0.02617 es el cambio esperado en el logit al disminuir una unidad la nota que obtiene el estudiante en un examen de lectura estándar.
2. exp(-0.02617) es la razón de Odds al disminuir una unidad la nota que obtiene el estudiante en un examen de lectura estándar, supuestas estables el resto de las variables en el modelo.

   $$e^{-0.02617} = 0.97416947$$

   Es decir, al aumentar una unidad la nota, la Odds o ventaja se vuelve positivo es decir aumenta la posibilidad de que entre a clases de repaso (podríamos decir que el riesgo que necesite clases de repaso frente a que no necesite es 0.97)

También existe una relación significativa positiva entre el *logaritmo de odds* de necesitar clases de repaso y el género del estudiante (p-*value*(0.0462) < α(0.05)), siendo, para un mismo resultado en el examen de lectura, mayor si el estudiante es hombre.

Coeficiente de sexo:

1. 0.64749 es el cambio esperado en el logit al pasar de un hombre a una mujer.
2. La razón de odds que *compara* mujeres con hombres es igual a exp(0.64749)= 1.91073885. Es decir, es 1,9 veces superior la odds o ventaja que necesiten clases de repaso los hombres que las mujeres.

   En concreto los *odds* de que un hombre requiera clases de repaso es $e^{0.64749}$=1.91073885 mayores que los de las mujeres.

4. predicción

```
#  Prediccion para nuevos individuos
# sexo    examen lectura
#  Hombre       60
nuevo1<-data.frame(sexo= "Hombre", examen_lectura=60)
predict(modelo_final, newdata=nuevo1, type="response")
```

```
##         1
## 0.4297831
```

```
# sexo    examen lectura
#  Mujer       60
nuevo1<-data.frame(sexo="Mujer", examen_lectura=10)
predict(modelo_final,newdata=nuevo1,type="response")
```

```
##         1
## 0.5466057
```

a. selección automática (forward)

```
# seleccion del mejor modelo
# modelo completo
```

```
datos <- read.csv("clases de repaso.csv", head=T,sep=",")
str(datos
```

```
## 'data.frame':    189 obs. of  4 variables:
##  $ clases_repaso : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sexo          : int  1 1 0 0 0 1 0 1 0 0 ...
##  $ examen_lectura: int  91 60 70 54 50 62 59 60 45 60 ...
##  $ examen_letras : int  87 78 60 54 60 58 55 49 65 60 ...
```

```
# como factor las variables cualitativas
datos$clases_repaso <- as.factor(datos$clases_repaso)
datos$sexo <- as.factor(datos$sexo)
str(datos)
```

```
## 'data.frame':    189 obs. of  4 variables:
##  $ clases_repaso : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ sexo          : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 1 2 1 1
...
##  $ examen_lectura: int  91 60 70 54 50 62 59 60 45 60 ...
##  $ examen_letras : int  87 78 60 54 60 58 55 49 65 60 ...
```

```
# 1. Modelo sin predictores
null_model <- glm(clases_repaso ~ 1, data = datos,
                  family = binomial())
summary(null_model)
```

```
## Call:
## glm(formula = clases_repaso ~ 1, family = binomial(), data = datos)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.6459     0.1531  -4.218 2.47e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 243.28  on 188  degrees of freedom
## Residual deviance: 243.28  on 188  degrees of freedom
## AIC: 245.28
##
## Number of Fisher Scoring iterations: 4
```

```
# 2. modelo completo
full_model <- glm(clases_repaso ~ ., data = datos,
                  family = binomial())
summary(full_model)
```

```
## Call:
## glm(formula = clases_repaso ~ ., family = binomial(), data = datos)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.07707    0.81788   0.094   0.9249
## sexo1           0.70495    0.31824   2.215   0.0268 *
## examen_lectura -0.04017    0.01997  -2.012   0.0443 *
## examen_letras   0.02178    0.02111   1.032   0.3021
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 243.28  on 188  degrees of freedom
## Residual deviance: 231.81  on 185  degrees of freedom
## AIC: 239.81
##
## Number of Fisher Scoring iterations: 4
```

Aplicamos el **método paso a paso** hacia atrás

```r
# Aplicamos el metodo paso a paso hacia atras
step1 <- step(null_model, scope = list(lower = null_model,
                                       upper = full_model),
              direction = "backward")
```

```
## Start:  AIC=245.28
## clases_repaso ~ 1
```

```r
summary(step1)
```

```
## Call:
## glm(formula = clases_repaso ~ 1, family = binomial(), data = datos)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.6459     0.1531  -4.218 2.47e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 243.28  on 188  degrees of freedom
## Residual deviance: 243.28  on 188  degrees of freedom
## AIC: 245.28
##
## Number of Fisher Scoring iterations: 4
```

```r
# forward
step_model <- step(null_model,scope = list(lower = null_model,
                                           upper = full_model),
              direction = "forward")
```

```
## Start:  AIC=245.28
## clases_repaso ~ 1
```

```
##
##                   Df Deviance    AIC
## + sexo             1   237.21 241.21
## + examen_lectura   1   237.88 241.88
## <none>                 243.28 245.28
## + examen_letras    1   241.58 245.58
##
## Step:  AIC=241.21
## clases_repaso ~ sexo
##
##                   Df Deviance    AIC
## + examen_lectura   1   232.89 238.89
## <none>                 237.21 241.21
## + examen_letras    1   235.97 241.97
##
## Step:  AIC=238.89
## clases_repaso ~ sexo + examen_lectura
##
##                  Df Deviance    AIC
## <none>                232.89 238.89
## + examen_letras  1   231.81 239.81
```

```
summary(step_model)
```

```
## Call:
## glm(formula = clases_repaso ~ sexo + examen_lectura, family = binom
ial(),
##     data = datos)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.42158    0.74834   0.563   0.5732
## sexo1           0.70337    0.31727   2.217   0.0266 *
## examen_lectura -0.02346    0.01157  -2.028   0.0426 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 243.28  on 188  degrees of freedom
## Residual deviance: 232.89  on 186  degrees of freedom
## AIC: 238.89
##
## Number of Fisher Scoring iterations: 4
```

```
step_model2 <- step(null_model,scope = list(lower = null_model,
                                            upper = full_model),
                    direction = "both")
```

```
## Start:  AIC=245.28
## clases_repaso ~ 1
##
##                   Df Deviance    AIC
## + sexo             1   237.21 241.21
## + examen_lectura   1   237.88 241.88
## <none>                 243.28 245.28
## + examen_letras    1   241.58 245.58
##
## Step:  AIC=241.21
## clases_repaso ~ sexo
##
##                   Df Deviance    AIC
## + examen_lectura   1   232.89 238.89
## <none>                 237.21 241.21
## + examen_letras    1   235.97 241.97
## - sexo             1   243.28 245.28
##
## Step:  AIC=238.89
## clases_repaso ~ sexo + examen_lectura
##
##                   Df Deviance    AIC
## <none>                 232.89 238.89
```

```
## + examen_letras   1   231.81 239.81
## - examen_lectura  1   237.21 241.21
## - sexo            1   237.88 241.88
```

```
summary(step_model2)
```

```
##
## Call:
## glm(formula = clases_repaso ~ sexo + examen_lectura, family = binom
ial(),
##     data = datos)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.42158    0.74834   0.563   0.5732
## sexo1           0.70337    0.31727   2.217   0.0266 *
## examen_lectura -0.02346    0.01157  -2.028   0.0426 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 243.28  on 188  degrees of freedom
## Residual deviance: 232.89  on 186  degrees of freedom
## AIC: 238.89
##
## Number of Fisher Scoring iterations: 4
```

Como Machine learning

```
#########################################################
# como machine learning
datos <- read.csv("clases de repaso.csv", head=T,sep=",")
str(datos)
```

```
## 'data.frame':    189 obs. of  4 variables:
##  $ clases_repaso : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sexo          : int  1 1 0 0 0 1 0 1 0 0 ...
##  $ examen_lectura: int  91 60 70 54 50 62 59 60 45 60 ...
##  $ examen_letras : int  87 78 60 54 60 58 55 49 65 60 ...
```

```
# como factor las variables cualitativas
datos$clases_repaso <- as.factor(datos$clases_repaso)
datos$sexo <- as.factor(datos$sexo)
str(datos)
```

```
## 'data.frame':    189 obs. of  4 variables:
##  $ clases_repaso : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ sexo          : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 1 2 1 1
...
##  $ examen_lectura: int  91 60 70 54 50 62 59 60 45 60 ...
##  $ examen_letras : int  87 78 60 54 60 58 55 49 65 60 ...
```

```
# etiquetando las categorias par efectos de obtener tablas
datos$clases_repaso <- factor(datos$clases_repaso,levels = c("0","1"),
labels=c("no asista", "asista"))
datos$sexo <- factor(datos$sexo,levels = c("0","1"),labels=c("Mujer",
"Hombre"))
str(datos)
```

```
## 'data.frame':    189 obs. of  4 variables:
##  $ clases_repaso : Factor w/ 2 levels "no asista","asista": 1 1 1 1
1 1 1 1 1 1 ...
##  $ sexo          : Factor w/ 2 levels "Mujer","Hombre": 2 2 1 1 1 2
1 2 1 1 ...
##  $ examen_lectura: int  91 60 70 54 50 62 59 60 45 60 ...
##  $ examen_letras : int  87 78 60 54 60 58 55 49 65 60 ...
```

```
# Divide set en Train y Test (machine learning)
# particion de datos
set.seed(1)
muestra <- sample(nrow(datos),nrow(datos)*.7)
Train   <- datos[muestra,]
dim(Train)
```

```
## [1] 132   4
```

```
Test    <- datos[-muestra,]
dim (Test)
```

```
## [1] 57  4
```

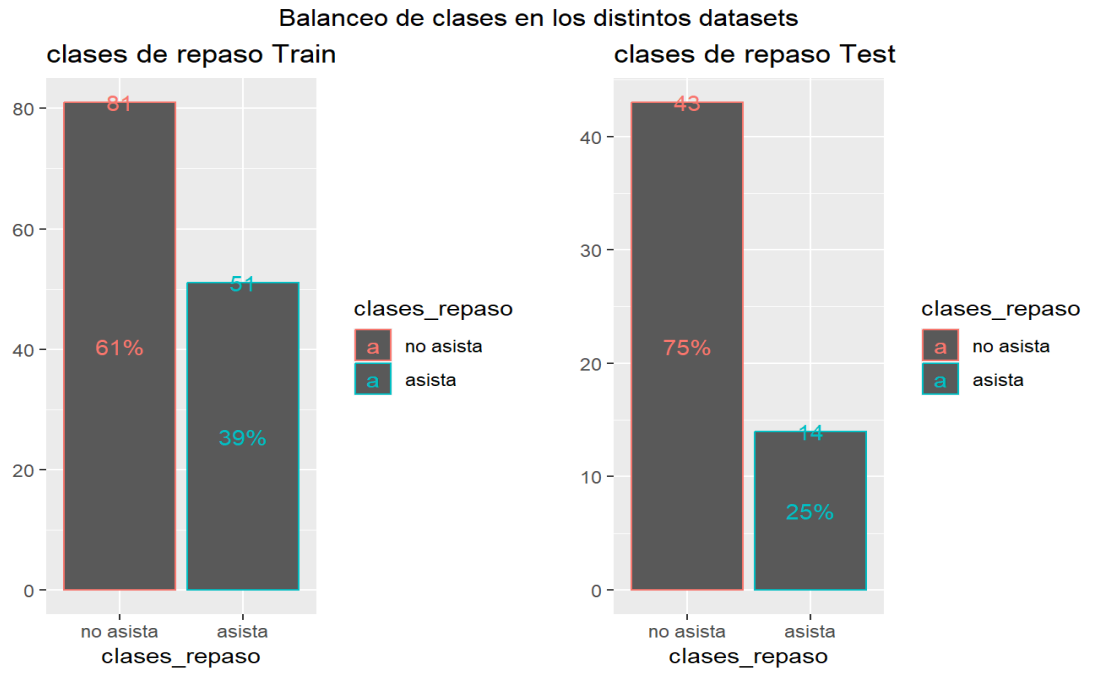
```
# Chequeo la distribucion de las clases
library(ggplot2)
library(gridExtra)
library(grid)
table(Train$clases_repaso)
```

```
##
## no asista    asista
##        81        51
```

```
table(Test$clases_repaso)
```

```
##
## no asista    asista
##        43        14
```

```
g1 <- qplot(clases_repaso, data = Train, col = clases_repaso,
      main = "clases de repaso Train",geom = "bar") +
  geom_text(aes(label=scales::percent(..count../sum(..count..))),
            stat='count',position=position_stack(0.5))+
  geom_text(aes(label=..count..),
            stat="count",position=position_stack())
g2<-qplot(clases_repaso, data = Test, col = clases_repaso,
      main = "clases de repaso Test",geom = "bar") +
  geom_text(aes(label=scales::percent(..count../sum(..count..))),
            stat='count',position=position_stack(0.5))+
  geom_text(aes(label=..count..),
            stat="count",position=position_stack())
grid.arrange(g1,g2,
             nrow = 1,
             top = "Balanceo de clases en los distintos datasets",
             bottom = textGrob(
               "1 asista a clases de repaso",
               gp = gpar(fontface = 3, fontsize = 9),
               hjust = 1,
               x = 1
             ))
```



1 asista a clases de repaso

```r
# Generar modelo inicial modelo train
modelo1 <- glm(clases_repaso ~ examen_lectura + sexo + examen_letras,
               data = Train,family = "binomial")
summary(modelo1)
```

```
##
## Call:
## glm(formula = clases_repaso ~ examen_lectura + sexo + examen_letras
,
##     family = "binomial", data = Train)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.01012    0.86472  -0.012    0.991
## examen_lectura -0.04357    0.02481  -1.757    0.079 .
## sexoHombre      0.54744    0.36736   1.490    0.136
## examen_letras   0.03070    0.02664   1.152    0.249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 176.11  on 131  degrees of freedom
## Residual deviance: 169.65  on 128  degrees of freedom
## AIC: 177.65
##
## Number of Fisher Scoring iterations: 4
```

```r
# variables significativas
sig.var <- summary(modelo1)$coeff[-1.4] < 0.05
names (sig.var)
```

```
## NULL
```

```
# no se puede seguir con machine learning (veamos con otro ejemplo)
```

3. Ejemplo 2 (varias variables independientes numéricas)

Se dispone de un registro que contiene cientos de emails con información de cada uno de ellos. El objetivo de estudio es intentar crear un modelo que permita filtrar que emails son "spam" y cuáles no, en función de determinadas características. Ejemplo extraído del libro OpenIntro Statistics.

```
library(openintro)
library(airports)
library(caret)
library(lattice)
data(email)
str(email)
```

```
## 'data.frame':    3921 obs. of  21 variables:
##  $ spam        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ to_multiple : num  0 0 0 0 0 0 1 1 0 0 ...
##  $ from        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ cc          : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ sent_email  : num  0 0 0 0 0 0 1 1 0 0 ...
##  $ time        : POSIXct, format: "2012-01-01 01:16:41" "2012-01-01 02:03:59" ...
##  $ image       : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ attach      : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ dollar      : num  0 0 4 0 0 0 0 0 0 0 ...
##  $ winner      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ inherit     : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ viagra      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ password    : num  0 0 0 0 2 2 0 0 0 0 ...
##  $ num_char    : num  11.37 10.5 7.77 13.26 1.23 ...
##  $ line_breaks : int  202 202 192 255 29 25 193 237 69 68 ...
##  $ format      : num  1 1 1 1 0 0 1 1 0 1 ...
```

```
##  $ re_subj     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ exclaim_subj: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ urgent_subj : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ exclaim_mess: num  0 1 6 48 1 1 1 18 1 0 ...
##  $ number      : Factor w/ 3 levels "none","small",..: 3 2 2 2 1 1 3 2 2 2 ...
```

En este caso se van a emplear únicamente como posibles predictores variables categóricas. Esto se debe a que los *outliers* complican bastante la creación de estos modelos y en el data set que se emplea como ejemplo las variables cuantitativas son muy asimétricas. En particular, las variables que se van a estudiar como posibles predictores son:

- spam: si el email es spam (1) si no lo es (0)
- to_multiple: si hay más de una persona en la lista de distribución.
- format: si está en formato HTLM
- cc: si hay otras direcciones en copia.
- attach: si hay archivos adjuntos
- dollar: si el email contiene la palabra dollar o el símbolo $.
- inherit: si contiene la palabra inherit
- winner: si el email contiene la palabra winner.
- password: si el email contiene la palabra password.
- re_subj: si la palabra "Re:" está escrita en el asunto del email.
- exclaim_subj: si se incluye algún signo de exclamación en el email.

En primer lugar, se genera el modelo completo introduciendo todas las variables como predictores.

```
email$spam <- as.factor(email$spam)
str(email)
```

```
## tibble [3,921 × 21] (S3: tbl_df/tbl/data.frame)
##  $ spam        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ to_multiple : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 1 ..
.
##  $ from        : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ..
.
```

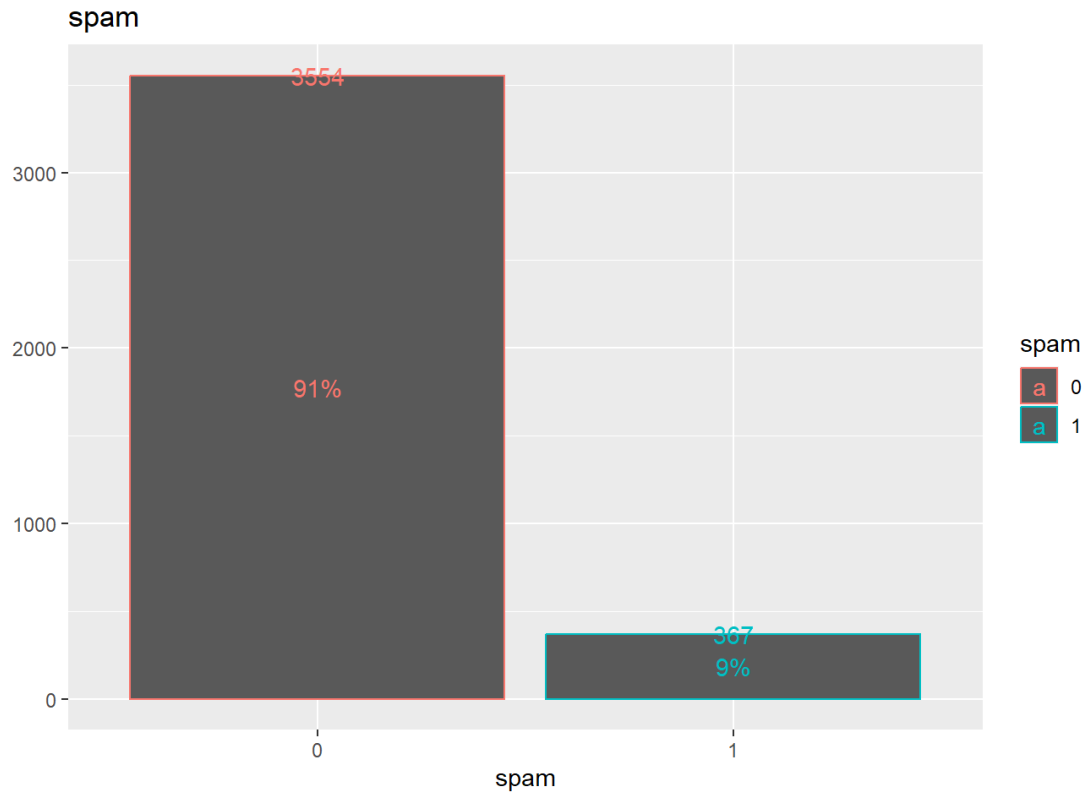
```
##  $ cc           : int [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ sent_email   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 1 ..
.
##  $ time         : POSIXct[1:3921], format: "2012-01-01 01:16:41" "20
12-01-01 02:03:59" ...
##  $ image        : num [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ attach       : num [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ dollar       : num [1:3921] 0 0 4 0 0 0 0 0 0 0 ...
##  $ winner       : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ inherit      : num [1:3921] 0 0 1 0 0 0 0 0 0 0 ...
##  $ viagra       : num [1:3921] 0 0 0 0 0 0 0 0 0 0 ...
##  $ password     : num [1:3921] 0 0 0 0 2 2 0 0 0 0 ...
##  $ num_char     : num [1:3921] 11.37 10.5 7.77 13.26 1.23 ...
##  $ line_breaks  : int [1:3921] 202 202 192 255 29 25 193 237 69 68 .
..
##  $ format       : Factor w/ 2 levels "0","1": 2 2 2 2 1 1 2 2 1 2 ..
.
##  $ re_subj      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ exclaim_subj: num [1:3921] 0 0 0 0 0 0 0 0 0 0 ...
##  $ urgent_subj  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ exclaim_mess: num [1:3921] 0 1 6 48 1 1 1 18 1 0 ...
##  $ number       : Factor w/ 3 levels "none","small",..: 3 2 2 2 1 1
3 2 2 2 ...
```

```
# Analisis de las observaciones
table(email$spam)
```

```
##
##    0    1
## 3554  367
```

```
library(ggplot2)
qplot(spam, data = email, col = spam,
      main = "spam",geom = "bar") +
  geom_text(aes(label=scales::percent(..count../sum(..count..))),
```

```
            stat='count',position=position_stack(0.5))+
  geom_text(aes(label=..count..),
            stat="count",position=position_stack())
```




```
# Generar el modelo de regresion logistica
modelo_completo <- glm(spam ~ to_multiple + format + cc + attach + dol
lar +
                         winner + inherit + password + re_subj + excla
im_subj, data = email,
                       family = binomial())
summary(modelo_completo)
```

```
##
## Call:
## glm(formula = spam ~ to_multiple + format + cc + attach + dollar +
##     winner + inherit + password + re_subj + exclaim_subj, family =
binomial(),
```

```
##     data = email)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.79976    0.08935  -8.950  < 2e-16 ***
## to_multiple1 -2.84097    0.31158  -9.118  < 2e-16 ***
## format1      -1.52284    0.12270 -12.411  < 2e-16 ***
## cc            0.03134    0.01895   1.654 0.098058 .
## attach        0.20351    0.05851   3.478 0.000505 ***
## dollar       -0.07304    0.02306  -3.168 0.001535 **
## winneryes     1.83103    0.33641   5.443 5.24e-08 ***
## inherit       0.32999    0.15223   2.168 0.030184 *
## password     -0.75953    0.29597  -2.566 0.010280 *
## re_subj1     -3.11857    0.36522  -8.539  < 2e-16 ***
## exclaim_subj  0.24399    0.22502   1.084 0.278221
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2437.2  on 3920  degrees of freedom
## Residual deviance: 1936.2  on 3910  degrees of freedom
## AIC: 1958.2
##
## Number of Fisher Scoring iterations: 7
```

```
#variables significativas
sig.var <- summary(modelo_completo)$coeff[-1,4] < 0.05
sig.var
```

```
## to_multiple1      format1           cc       attach       dollar
winneryes
##         TRUE         TRUE        FALSE         TRUE         TRUE
TRUE
##      inherit     password     re_subj1 exclaim_subj
##         TRUE         TRUE         TRUE        FALSE
```

Se mejora el modelo mediante el proceso basado en *p-values*. El resultado es el siguiente modelo.

```
# sin cc y exclaim_subj por no ser significativo
modelo_final <- glm(spam ~ to_multiple + format + attach + dollar + wi
nner +
                      inherit + password + re_subj, data = email, fami
ly = binomial())
summary(modelo_final)
```

```
##
## Call:
## glm(formula = spam ~ to_multiple + format + attach + dollar +
##     winner + inherit + password + re_subj, family = binomial(),
##     data = email)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.78138    0.08860  -8.820  < 2e-16 ***
## to_multiple1 -2.77682    0.30752  -9.030  < 2e-16 ***
## format1      -1.51770    0.12226 -12.414  < 2e-16 ***
## attach        0.20419    0.05789   3.527  0.00042 ***
## dollar       -0.06970    0.02239  -3.113  0.00185 **
## winneryes     1.86675    0.33652   5.547  2.9e-08 ***
## inherit       0.33614    0.15073   2.230  0.02575 *
## password     -0.76035    0.29680  -2.562  0.01041 *
## re_subj1     -3.11329    0.36519  -8.525  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2437.2  on 3920  degrees of freedom
## Residual deviance: 1939.6  on 3912  degrees of freedom
## AIC: 1957.6
##
```

```
## Number of Fisher Scoring iterations: 7
```

```
# prediccion del modelo
pred1=predict(modelo_final,newdata = email,type="response")
head(pred1)
```

```
##          1          2          3          4          5          6
## 0.09119874 0.09119874 0.09606451 0.09119874 0.09095046 0.09095046
```

```
library(vcd)
predicciones <- ifelse(test = modelo_final$fitted.values > 0.5, yes = 1, no = 0)
head(predicciones)
```

```
## 1 2 3 4 5 6
## 0 0 0 0 0 0
```

Con este modelo podemos saber la probabilidad, dadas unas determinadas características, de que el email sea *spam* (valor 1 de la variable).

Para evaluar el modelo, se puede comparar el valor real (si realmente es spam) con el predicho por el modelo.

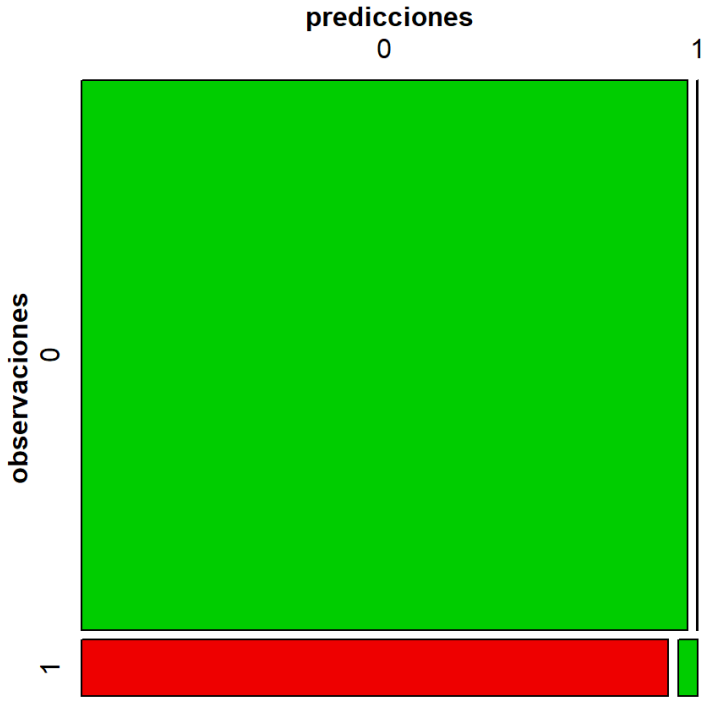
```
# Evaluacion del modelo
matriz_confusion <- table(email$spam, floor(pred1+0.5),
                          dnn = c("observaciones", "predicciones"))
matriz_confusion
```

```
##              predicciones
## observaciones    0    1
##             0 3551    3
##             1  355   12
```

```
error1 <- sum(matriz_confusion[1,2], matriz_confusion[2,1])/sum(matriz
_confusion)

error1
```

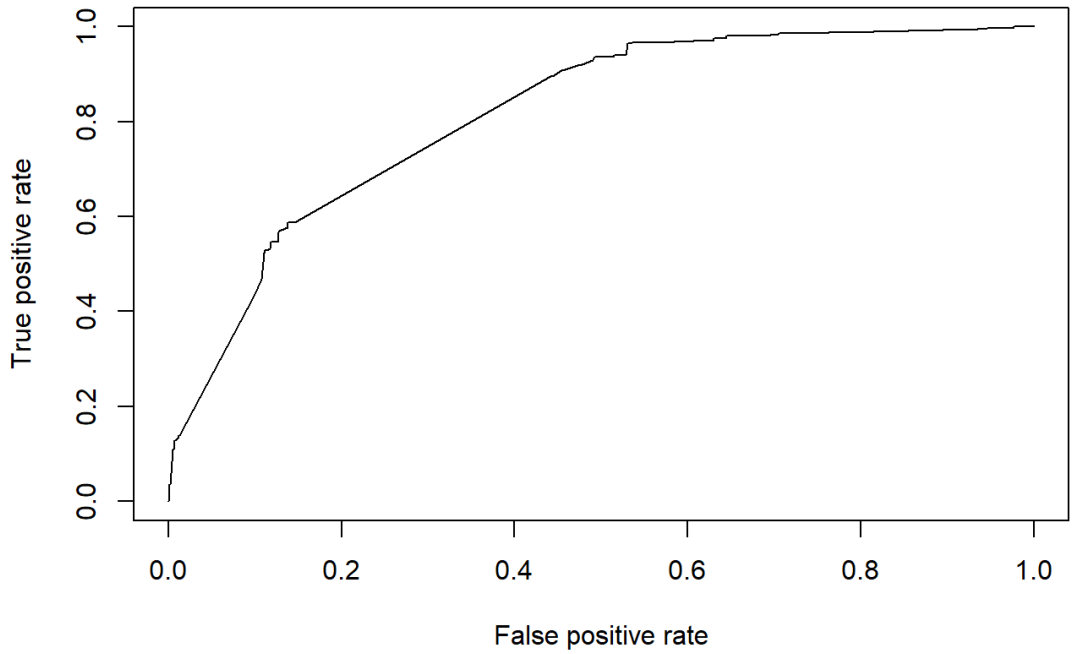
```
## [1] 0.09130324
```

```
mosaic(matriz_confusion, shade = T, colorize = T,

       gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"),
2, 2)))
```




```
#####################

# Curva ROC y Area bajo la curva

# library(InformationValue)

# plotROC(sjlabelled::as_numeric(email$spam,start.at = 0), pred1)


library(ROCR)

pred = ROCR::prediction(pred1,email$spam)
```

```
perf <- performance(pred, "tpr", "fpr")
plot(perf)
```




```
AUCLog1=performance(pred, measure = "auc")@y.values[[1]]
cat("AUC: ",AUCLog1,"n")
```

```
## AUC:  0.8155534 n
```

```
#################################
#  Prediccion para nuevos individuos
nuevo1<- data.frame(attach =0, dollar=4,inherit=0, password=2,
                    to_multiple=c("0", "1"),format=c("0","1"),winner=c
("no","yes"),re_subj=c("0","1"))
predict(modelo_final,newdata=nuevo1,type="response")
```

```
##            1            2
## 0.0703796336 0.0002968675
```

```
# spam 0=0.07 si attach = 0, dollar=4,inherit=0, password=2,
# to_multiple=0,format=0,winner=no y re_subj=0)

# spam 0=0.00002 si attach = 0, dollar=4,inherit=0, password=2,
# to_multiple=1,format=1,winner=si y re_subj=1)

# metodos de seleccion de variables
# Stepwise
library(stats)
library(dplyr)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following objects are masked from 'package:openintro':
##
##     housing, mammals
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
step <- stepAIC(modelo_final, direction="both",trace=1)
```

```
## Start:  AIC=1957.56
## spam ~ to_multiple + format + attach + dollar + winner + inherit +
##     password + re_subj
##
##               Df Deviance    AIC
## <none>              1939.6 1957.6
## - inherit      1   1943.9 1959.9
```

```
## - attach       1   1947.8 1963.8
## - dollar       1   1953.7 1969.7
## - password     1   1956.1 1972.1
## - winner       1   1967.0 1983.0
## - format       1   2096.2 2112.2
## - to_multiple  1   2105.8 2121.8
## - re_subj      1   2133.1 2149.1
```

```
# con el resultado del modelo se obtiene un nuevo modelo
modelo_final <- glm(spam ~ to_multiple + format + attach + dollar + wi
nner +
                      inherit + password + re_subj, data = email, fami
ly = binomial())
summary(modelo_final)
```

```
##
## Call:
## glm(formula = spam ~ to_multiple + format + attach + dollar +
##     winner + inherit + password + re_subj, family = binomial(),
##     data = email)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.78138    0.08860  -8.820  < 2e-16 ***
## to_multiple1 -2.77682    0.30752  -9.030  < 2e-16 ***
## format1      -1.51770    0.12226 -12.414  < 2e-16 ***
## attach        0.20419    0.05789   3.527  0.00042 ***
## dollar       -0.06970    0.02239  -3.113  0.00185 **
## winneryes     1.86675    0.33652   5.547  2.9e-08 ***
## inherit       0.33614    0.15073   2.230  0.02575 *
## password     -0.76035    0.29680  -2.562  0.01041 *
## re_subj1     -3.11329    0.36519  -8.525  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 2437.2  on 3920  degrees of freedom
## Residual deviance: 1939.6  on 3912  degrees of freedom
## AIC: 1957.6
##
## Number of Fisher Scoring iterations: 7
```

```
# prediccion del modelo
probStep=predict(modelo_final,newdata = email,type="response")
head(probStep)
```

```
##         1         2         3         4         5         6
## 0.09119874 0.09119874 0.09606451 0.09119874 0.09095046 0.09095046
```

```
library(vcd)
predicciones <- ifelse(test = modelo_final$fitted.values > 0.5, yes =
1, no = 0)
head(predicciones)
```

```
## 1 2 3 4 5 6
## 0 0 0 0 0 0
```

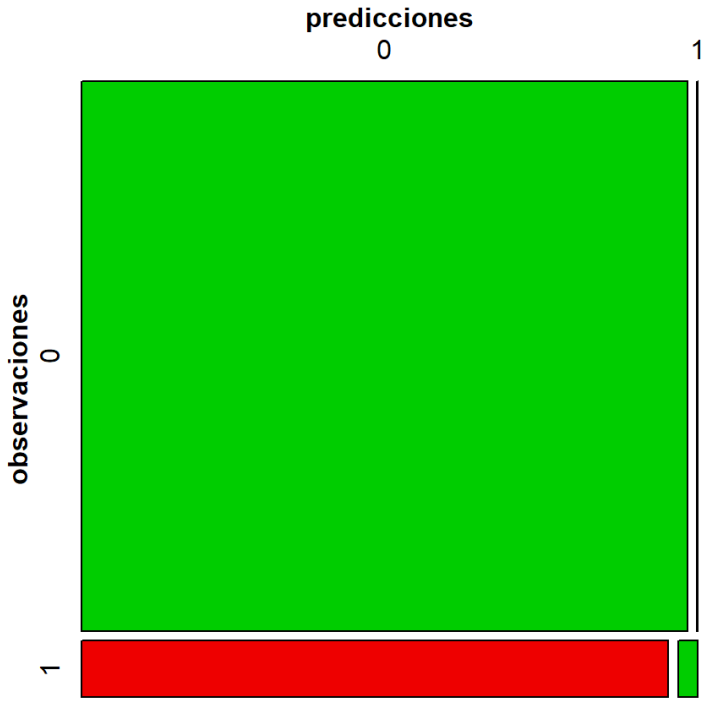
```
##################
# Evaluacion del modelo
matriz_confusion <- table(email$spam, floor(probStep+0.5),
                          dnn = c("observaciones", "predicciones"))
matriz_confusion
```

```
##              predicciones
## observaciones    0    1
##             0 3551    3
##             1  355   12
```

```
sion);error1
```

```
## [1] 0.09130324
```

```
mosaic(matriz_confusion, shade = T, colorize = T,
       gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"),
2, 2)))
```




```
############################
# Curva ROC y Area bajo la curva
# library(InformationValue)
# plotROC(sjlabelled::as_numeric(email$spam,start.at = 0), probStep)


# 1. Modelo sin predictores
null_model <- glm(spam ~ 1, data = email,
```

```
                  family = binomial())
summary(null_model)
```

```
##
## Call:
## glm(formula = spam ~ 1, family = binomial(), data = email)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.27047    0.05483  -41.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2437.2  on 3920  degrees of freedom
## Residual deviance: 2437.2  on 3920  degrees of freedom
## AIC: 2439.2
##
## Number of Fisher Scoring iterations: 5
```

```
# 2. modelo completo
full_model <- glm(spam ~ ., data = email,
                  family = binomial())
```

```
summary(full_model)
```

```
##
## Call:
## glm(formula = spam ~ ., family = binomial(), data = email)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -9.090e+01  9.800e+03  -0.009  0.99260
```

```
## to_multiple1 -2.680e+00  3.265e-01  -8.208 2.25e-16 ***
## from1        -2.195e+01  9.800e+03  -0.002  0.99821
## cc            1.877e-02  2.197e-02   0.855  0.39282
## sent_email1  -2.074e+01  3.870e+02  -0.054  0.95725
## time          8.482e-08  2.850e-08   2.976  0.00292 **
## image        -1.782e+00  5.946e-01  -2.996  0.00273 **
## attach        7.345e-01  1.443e-01   5.089 3.61e-07 ***
## dollar       -6.846e-02  2.645e-02  -2.588  0.00964 **
## winneryes     2.071e+00  3.652e-01   5.672 1.41e-08 ***
## inherit       3.146e-01  1.556e-01   2.022  0.04316 *
## viagra        2.843e+00  2.216e+03   0.001  0.99898
## password     -8.545e-01  2.971e-01  -2.876  0.00403 **
## num_char      5.061e-02  2.380e-02   2.127  0.03346 *
## line_breaks  -5.490e-03  1.352e-03  -4.060 4.91e-05 ***
## format1      -6.142e-01  1.485e-01  -4.136 3.53e-05 ***
## re_subj1     -1.642e+00  3.865e-01  -4.248 2.16e-05 ***
## exclaim_subj  1.420e-01  2.427e-01   0.585  0.55843
## urgent_subj1  3.885e+00  1.317e+00   2.950  0.00318 **
## exclaim_mess  1.083e-02  1.811e-03   5.980 2.23e-09 ***
## numbersmall  -1.192e+00  1.540e-01  -7.744 9.62e-15 ***
## numberbig    -2.953e-01  2.196e-01  -1.345  0.17867
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2437.2  on 3920  degrees of freedom
## Residual deviance: 1635.7  on 3899  degrees of freedom
## AIC: 1679.7
##
## Number of Fisher Scoring iterations: 19
```

```
# Aplicamos el método paso a paso hacia atrás
step1 <- step(null_model, scope = list(lower = null_model,
                                       upper = full_model),
              direction = "backward")
```

```
## Start:  AIC=2439.18
## spam ~ 1
```

```
summary(step1)
## Call:
## glm(formula = spam ~ 1, family = binomial(), data = email)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.27047    0.05483  -41.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2437.2  on 3920  degrees of freedom
## Residual deviance: 2437.2  on 3920  degrees of freedom
## AIC: 2439.2
##
## Number of Fisher Scoring iterations: 5
```

```
# forward
step_model <- step(null_model,
                   scope = list(lower = null_model,
                                upper = full_model),
                   direction = "forward")
```

```
## Start:  AIC=2439.18
## spam ~ 1
```

```
##                Df Deviance    AIC
```

```
## + sent_email    1   2183.8 2187.8
## + number        2   2250.4 2256.4
## + re_subj       1   2264.1 2268.1
## + format        1   2314.6 2318.6
## + line_breaks   1   2322.4 2326.4
## + num_char      1   2346.4 2350.4
## + to_multiple   1   2372.0 2376.0
## + winner        1   2412.7 2416.7
## + from          1   2422.9 2426.9
## + dollar        1   2424.8 2428.8
## + image         1   2425.0 2429.0
## + urgent_subj   1   2427.2 2431.2
## + password      1   2427.7 2431.7
## + time          1   2428.7 2432.7
## + viagra        1   2432.4 2436.4
## + attach        1   2433.2 2437.2
## + inherit       1   2435.2 2439.2
## <none>              2437.2 2439.2
## + exclaim_mess  1   2437.1 2441.1
## + cc            1   2437.1 2441.1
## + exclaim_subj  1   2437.2 2441.2
##
## Step:  AIC=2187.8
## spam ~ sent_email
```

```
##                Df Deviance    AIC
## + number        2   1977.1 1985.1
## + line_breaks   1   2048.9 2054.9
## + num_char      1   2060.3 2066.3
## + format        1   2093.4 2099.4
## + to_multiple   1   2108.5 2114.5
## + re_subj       1   2155.3 2161.3
## + winner        1   2161.2 2167.2
## + dollar        1   2163.2 2169.2
## + from          1   2171.5 2177.5
## + password      1   2172.0 2178.0
```

```
## + urgent_subj   1   2172.2 2178.2
## + attach        1   2174.4 2180.4
## + image         1   2174.5 2180.5
## + viagra        1   2179.7 2185.7
## + time          1   2180.8 2186.8
## <none>              2183.8 2187.8
## + exclaim_subj  1   2183.3 2189.3
## + inherit       1   2183.4 2189.4
## + cc            1   2183.4 2189.4
## + exclaim_mess  1   2183.8 2189.8
##
## Step:  AIC=1985.12
## spam ~ sent_email + number
```

```
##                Df Deviance    AIC
## + to_multiple   1   1903.3 1913.3
## + line_breaks   1   1923.3 1933.3
## + num_char      1   1929.9 1939.9
## + format        1   1949.8 1959.8
## + winner        1   1950.8 1960.8
## + re_subj       1   1957.8 1967.8
## + password      1   1960.0 1970.0
## + from          1   1963.9 1973.9
## + urgent_subj   1   1969.5 1979.5
## + image         1   1970.0 1980.0
## + time          1   1971.5 1981.5
## + dollar        1   1971.6 1981.6
## + viagra        1   1972.1 1982.1
## + attach        1   1972.8 1982.8
## <none>              1977.1 1985.1
## + inherit       1   1975.2 1985.2
## + cc            1   1976.9 1986.9
## + exclaim_mess  1   1977.0 1987.0
## + exclaim_subj  1   1977.0 1987.0
##
## Step:  AIC=1913.29
```

```
## spam ~ sent_email + number + to_multiple
```

```
##                Df Deviance    AIC
## + line_breaks   1   1835.4 1847.4
## + format        1   1839.8 1851.8
## + num_char      1   1846.7 1858.7
## + winner        1   1879.8 1891.8
## + password      1   1882.8 1894.8
## + re_subj       1   1885.2 1897.2
## + from          1   1891.5 1903.5
## + attach        1   1892.0 1904.0
## + dollar        1   1892.8 1904.8
## + urgent_subj   1   1894.2 1906.2
## + time          1   1897.1 1909.1
## + viagra        1   1898.5 1910.5
## + image         1   1900.5 1912.5
## <none>              1903.3 1913.3
## + cc            1   1902.0 1914.0
## + inherit       1   1902.3 1914.3
## + exclaim_subj  1   1902.9 1914.9
## + exclaim_mess  1   1903.3 1915.3
##
## Step:  AIC=1847.4
## spam ~ sent_email + number + to_multiple + line_breaks
```

```
##                Df Deviance    AIC
## + exclaim_mess  1   1803.2 1817.2
## + winner        1   1808.1 1822.1
## + format        1   1809.0 1823.0
## + re_subj       1   1812.8 1826.8
## + password      1   1815.4 1829.4
## + attach        1   1820.9 1834.9
## + from          1   1826.1 1840.1
## + urgent_subj   1   1827.0 1841.0
```

```
## + time          1   1828.8 1842.8
## + viagra        1   1831.6 1845.6
## <none>              1835.4 1847.4
## + dollar        1   1833.5 1847.5
## + image         1   1833.7 1847.7
## + inherit       1   1833.8 1847.8
## + num_char      1   1833.8 1847.8
## + cc            1   1834.6 1848.6
## + exclaim_subj  1   1835.1 1849.1
##
## Step:  AIC=1817.16
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss
```

```
##                Df Deviance    AIC
## + winner        1   1776.0 1792.0
## + re_subj       1   1780.7 1796.7
## + password      1   1783.4 1799.4
## + format        1   1783.5 1799.5
## + attach        1   1788.8 1804.8
## + from          1   1794.3 1810.3
## + urgent_subj   1   1794.9 1810.9
## + time          1   1796.3 1812.3
## + viagra        1   1799.7 1815.7
## + dollar        1   1799.8 1815.8
## + num_char      1   1800.6 1816.6
## + inherit       1   1801.1 1817.1
## <none>              1803.2 1817.2
## + image         1   1801.7 1817.7
## + cc            1   1802.4 1818.4
## + exclaim_subj  1   1802.9 1818.9
##
## Step:  AIC=1791.95
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner
```

```
##                Df Deviance    AIC
## + re_subj       1   1752.5 1770.5
## + password      1   1754.1 1772.1
## + format        1   1758.8 1776.8
## + attach        1   1760.8 1778.8
## + from          1   1766.8 1784.8
## + urgent_subj   1   1767.6 1785.6
## + time          1   1769.0 1787.0
## + dollar        1   1771.4 1789.4
## + viagra        1   1772.4 1790.4
## + num_char      1   1773.8 1791.8
## <none>              1776.0 1792.0
## + inherit       1   1774.2 1792.2
## + image         1   1774.7 1792.7
## + cc            1   1775.2 1793.2
## + exclaim_subj  1   1775.9 1793.9
```

```
## Step:  AIC=1770.54
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj
```

```
##                Df Deviance    AIC
## + password      1   1728.6 1748.6
## + format        1   1732.0 1752.0
## + attach        1   1738.3 1758.3
## + urgent_subj   1   1742.7 1762.7
## + from          1   1743.9 1763.9
## + time          1   1745.3 1765.3
## + dollar        1   1746.6 1766.6
## + viagra        1   1749.1 1769.1
## <none>              1752.5 1770.5
## + inherit       1   1751.2 1771.2
## + cc            1   1751.2 1771.2
```

```
## + num_char      1   1751.3 1771.3
## + image         1   1751.3 1771.3
## + exclaim_subj  1   1752.5 1772.5
```

```
##
## Step:  AIC=1748.61
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password
```

```
##                Df Deviance    AIC
## + format        1   1708.1 1730.1
## + attach        1   1715.6 1737.6
## + urgent_subj   1   1719.0 1741.0
## + from          1   1720.3 1742.3
## + dollar        1   1721.8 1743.8
## + time          1   1721.8 1743.8
## + viagra        1   1725.2 1747.2
## <none>              1728.6 1748.6
## + num_char      1   1727.2 1749.2
## + image         1   1727.2 1749.2
## + cc            1   1727.4 1749.4
## + inherit       1   1727.5 1749.5
## + exclaim_subj  1   1728.6 1750.6
##
## Step:  AIC=1730.12
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format
```

```
##                Df Deviance    AIC
## + urgent_subj   1   1696.8 1720.8
## + attach        1   1697.4 1721.4
## + from          1   1700.2 1724.2
## + time          1   1701.7 1725.7
```

```
## + dollar        1   1702.0 1726.0
## + viagra        1   1704.2 1728.2
## <none>              1708.1 1730.1
## + inherit       1   1706.1 1730.1
## + num_char      1   1706.3 1730.3
## + image         1   1706.8 1730.8
## + cc            1   1707.0 1731.0
## + exclaim_subj  1   1708.0 1732.0
##
## Step:  AIC=1720.76
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj
```

```
#                Df Deviance    AIC
## + attach        1   1686.1 1712.1
## + from          1   1688.9 1714.9
## + dollar        1   1690.3 1716.3
## + time          1   1690.7 1716.7
## + viagra        1   1692.8 1718.8
## <none>              1696.8 1720.8
## + num_char      1   1695.0 1721.0
## + inherit       1   1695.0 1721.0
## + cc            1   1695.5 1721.5
## + image         1   1695.5 1721.5
## + exclaim_subj  1   1696.6 1722.6
```

```
##
## Step:  AIC=1712.09
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj + attach
```

```
##                Df Deviance    AIC
## + image         1   1671.1 1699.1
```

```
## + from          1   1678.2 1706.2
## + time          1   1678.9 1706.9
## + dollar        1   1680.6 1708.6
## + viagra        1   1682.1 1710.1
## + num_char      1   1683.6 1711.6
## <none>              1686.1 1712.1
## + inherit       1   1684.1 1712.1
## + cc            1   1685.2 1713.2
## + exclaim_subj  1   1685.9 1713.9
```

```
## Step:  AIC=1699.14
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj + attach +
##     image
```

```
##                Df Deviance    AIC
## + from          1   1663.1 1693.1
## + time          1   1663.4 1693.4
## + dollar        1   1665.3 1695.3
## + viagra        1   1667.2 1697.2
## + num_char      1   1668.8 1698.8
## <none>              1671.1 1699.1
## + inherit       1   1669.3 1699.3
## + cc            1   1670.5 1700.5
## + exclaim_subj  1   1671.1 1701.1
```

```
##
## Step:  AIC=1693.12
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj + attach +
##     image + from
```

```
##                Df Deviance    AIC
## + time          1   1654.3 1686.3
## + dollar        1   1657.2 1689.2
## + viagra        1   1659.1 1691.1
## + num_char      1   1660.8 1692.8
## <none>              1663.1 1693.1
## + inherit       1   1661.2 1693.2
## + cc            1   1662.5 1694.5
## + exclaim_subj  1   1663.0 1695.0
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##
## Step:  AIC=1686.35
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj + attach +
##     image + from + time
```

```
##                Df Deviance    AIC
## + dollar        1   1649.2 1683.2
## + viagra        1   1649.8 1683.8
## + num_char      1   1651.9 1685.9
## + inherit       1   1652.3 1686.3
## <none>              1654.3 1686.3
## + cc            1   1653.6 1687.6
## + exclaim_subj  1   1654.2 1688.2
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##
## Step:  AIC=1683.15
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj + attach +
##     image + from + time + dollar
```

```
##                Df Deviance    AIC
## + viagra        1   1644.5 1680.5
## + num_char      1   1645.0 1681.0
```

```
## + inherit       1   1645.8 1681.8
## <none>              1649.2 1683.2
## + exclaim_subj  1   1648.4 1684.4
## + cc            1   1648.5 1684.5
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##
## Step:  AIC=1680.49
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj + attach +
##     image + from + time + dollar + viagra
```

```
##                Df Deviance    AIC
## + num_char      1   1640.4 1678.4
## + inherit       1   1641.0 1679.0
## <none>              1644.5 1680.5
## + cc            1   1643.8 1681.8
## + exclaim_subj  1   1644.0 1682.0
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##
## Step:  AIC=1678.36
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj + attach +
##     image + from + time + dollar + viagra + num_char
```

```
##                Df Deviance    AIC
## + inherit       1   1636.7 1676.7
## <none>              1640.4 1678.4
## + cc            1   1639.6 1679.6
## + exclaim_subj  1   1640.0 1680.0
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##
## Step:  AIC=1676.7
## spam ~ sent_email + number + to_multiple + line_breaks + exclaim_me
ss +
##     winner + re_subj + password + format + urgent_subj + attach +
```

```
##     image + from + time + dollar + viagra + num_char + inherit
```

```
##                Df Deviance    AIC
## <none>              1636.7 1676.7
## + cc            1   1636.0 1678.0
## + exclaim_subj  1   1636.4 1678.4
```

```
summary(step_model)
```

```
Call:
## glm(formula = spam ~ sent_email + number + to_multiple + line_break
s +
##     exclaim_mess + winner + re_subj + password + format + urgent_su
bj +
##     attach + image + from + time + dollar + viagra + num_char +
##     inherit, family = binomial(), data = email)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -9.050e+01  9.805e+03  -0.009  0.99264
## sent_email1  -2.089e+01  3.849e+02  -0.054  0.95671
## numbersmall  -1.190e+00  1.536e-01  -7.748 9.36e-15 ***
## numberbig    -3.014e-01  2.193e-01  -1.374  0.16934
## to_multiple1 -2.660e+00  3.260e-01  -8.158 3.40e-16 ***
## line_breaks  -5.546e-03  1.349e-03  -4.113 3.91e-05 ***
## exclaim_mess  1.085e-02  1.809e-03   5.997 2.01e-09 ***
## winneryes     2.095e+00  3.642e-01   5.751 8.85e-09 ***
## re_subj1     -1.633e+00  3.864e-01  -4.226 2.38e-05 ***
## password     -8.515e-01  2.972e-01  -2.865  0.00417 **
## format1      -6.045e-01  1.474e-01  -4.100 4.13e-05 ***
## urgent_subj1  3.853e+00  1.315e+00   2.929  0.00340 **
## attach        7.434e-01  1.438e-01   5.168 2.36e-07 ***
## image        -1.804e+00  5.940e-01  -3.036  0.00239 **
## from1        -2.193e+01  9.805e+03  -0.002  0.99822
## time          8.451e-08  2.848e-08   2.967  0.00301 **
```

```
## dollar       -6.588e-02  2.580e-02  -2.553  0.01068 *
## viagra        2.857e+00  2.216e+03   0.001  0.99897
## num_char      5.118e-02  2.373e-02   2.157  0.03104 *
## inherit       3.192e-01  1.545e-01   2.067  0.03876 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2437.2  on 3920  degrees of freedom
## Residual deviance: 1636.7  on 3901  degrees of freedom
## AIC: 1676.7
##
## Number of Fisher Scoring iterations: 19
```

```
#######################################
# como machine learning
library(openintro)
data(email)
str(email)
```

```
## tibble [3,921 × 21] (S3: tbl_df/tbl/data.frame)
##  $ spam        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ to_multiple : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 1 ..
.
##  $ from        : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ..
.
##  $ cc          : int [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ sent_email  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 1 ..
.
##  $ time        : POSIXct[1:3921], format: "2012-01-01 01:16:41" "20
12-01-01 02:03:59" ...
##  $ image       : num [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ attach      : num [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ dollar      : num [1:3921] 0 0 4 0 0 0 0 0 0 0 ...
##  $ winner      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1
...
```

```
##  $ inherit     : num [1:3921] 0 0 1 0 0 0 0 0 0 0 ...
##  $ viagra      : num [1:3921] 0 0 0 0 0 0 0 0 0 0 ...
##  $ password    : num [1:3921] 0 0 0 0 2 2 0 0 0 0 ...
##  $ num_char    : num [1:3921] 11.37 10.5 7.77 13.26 1.23 ...
##  $ line_breaks : int [1:3921] 202 202 192 255 29 25 193 237 69 68 .
..
##  $ format      : Factor w/ 2 levels "0","1": 2 2 2 2 1 1 2 2 1 2 ..
.
##  $ re_subj     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ exclaim_subj: num [1:3921] 0 0 0 0 0 0 0 0 0 0 ...
##  $ urgent_subj : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ exclaim_mess: num [1:3921] 0 1 6 48 1 1 1 18 1 0 ...
##  $ number      : Factor w/ 3 levels "none","small",..: 3 2 2 2 1 1
3 2 2
```

```
email$spam <- as.factor(email$spam)
str(email)
```

```
## tibble [3,921 × 21] (S3: tbl_df/tbl/data.frame)
##  $ spam        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ to_multiple : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 1 ..
.
##  $ from        : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ..
.
##  $ cc          : int [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ sent_email  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 1 ..
.
##  $ time        : POSIXct[1:3921], format: "2012-01-01 01:16:41" "20
12-01-01 02:03:59" ...
##  $ image       : num [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ attach      : num [1:3921] 0 0 0 0 0 0 0 1 0 0 ...
##  $ dollar      : num [1:3921] 0 0 4 0 0 0 0 0 0 0 ...
##  $ winner      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ inherit     : num [1:3921] 0 0 1 0 0 0 0 0 0 0 ...
##  $ viagra      : num [1:3921] 0 0 0 0 0 0 0 0 0 0 ...
##  $ password    : num [1:3921] 0 0 0 0 2 2 0 0 0 0 ...
```

```
##  $ num_char    : num [1:3921] 11.37 10.5 7.77 13.26 1.23 ...
##  $ line_breaks : int [1:3921] 202 202 192 255 29 25 193 237 69 68 .
..
##  $ format      : Factor w/ 2 levels "0","1": 2 2 2 2 1 1 2 2 1 2 ..
.
##  $ re_subj     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ exclaim_subj: num [1:3921] 0 0 0 0 0 0 0 0 0 0 ...
##  $ urgent_subj : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ exclaim_mess: num [1:3921] 0 1 6 48 1 1 1 18 1 0 ...
##  $ number      : Factor w/ 3 levels "none","small",..: 3 2 2 2 1 1
3 2 2 2 ...
```

```
# particion de datos
set.seed(1)
muestra1 <- sample(nrow(email),nrow(email)*.7)
Train   <- email[muestra1,]
dim(Train)
```

```
## [1] 2744   21
```

```
Test    <- email[-muestra1,]
dim(Test)
```

```
## [1] 1177   21
```

```
# escalando variables
# Generar modelo inicial modelo train
modelo1 <- glm(spam ~ to_multiple + format + attach + dollar + winner
+
                      inherit + password + re_subj, data = Train, fami
ly = binomial())
summary(modelo1)
```

```
# Call:
## glm(formula = spam ~ to_multiple + format + attach + dollar +
##     winner + inherit + password + re_subj, family = binomial(),
##     data = Train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.77155    0.10685  -7.221 5.16e-13 ***
## to_multiple1 -2.96216    0.39861  -7.431 1.08e-13 ***
## format1      -1.48841    0.14622 -10.179  < 2e-16 ***
## attach        0.15736    0.07448   2.113 0.034621 *
## dollar       -0.07626    0.02935  -2.598 0.009375 **
## winneryes     1.73002    0.46086   3.754 0.000174 ***
## inherit       0.32208    0.26854   1.199 0.230380
## password     -0.71970    0.31288  -2.300 0.021434 *
## re_subj1     -3.02551    0.42066  -7.192 6.37e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1688.1  on 2743  degrees of freedom
## Residual deviance: 1355.6  on 2735  degrees of freedom
## AIC: 1373.6
##
## Number of Fisher Scoring iterations: 7
```

```
modelo2 <- glm(spam ~ to_multiple + format + attach + dollar + winner
+
        password + re_subj, data = Train, family = binomial())
summary(modelo2)
```

```
## Call:
## glm(formula = spam ~ to_multiple + format + attach + dollar +
##     winner + password + re_subj, family = binomial(), data = Train)
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.77120    0.10683  -7.219 5.23e-13 ***
## to_multiple1 -2.95986    0.39835  -7.430 1.08e-13 ***
## format1      -1.47804    0.14571 -10.144  < 2e-16 ***
## attach        0.15691    0.07448   2.107 0.035147 *
## dollar       -0.06946    0.02826  -2.458 0.013966 *
## winneryes     1.75825    0.46221   3.804 0.000142 ***
## password     -0.72345    0.31313  -2.310 0.020865 *
## re_subj1     -3.03260    0.42065  -7.209 5.62e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1688.1  on 2743  degrees of freedom
## Residual deviance: 1356.9  on 2736  degrees of freedom
## AIC: 1372.9
##
## Number of Fisher Scoring iterations: 7
```

```
# prediccion del modelo
library(dplyr)
pred1<- predict(modelo2,newdata = Test, type="response")
head (pred1)
```

```
##           1           2           3           4           5
6
## 0.095414758 0.095414758 0.006354667 0.005057352 0.095414758 0.09541
4758
```

```
library(vcd)
predicciones <- ifelse(pred1 >= 0.5, 1, 0)
head(predicciones)
```

```
## 1 2 3 4 5 6
## 0 0 0 0 0 0
```

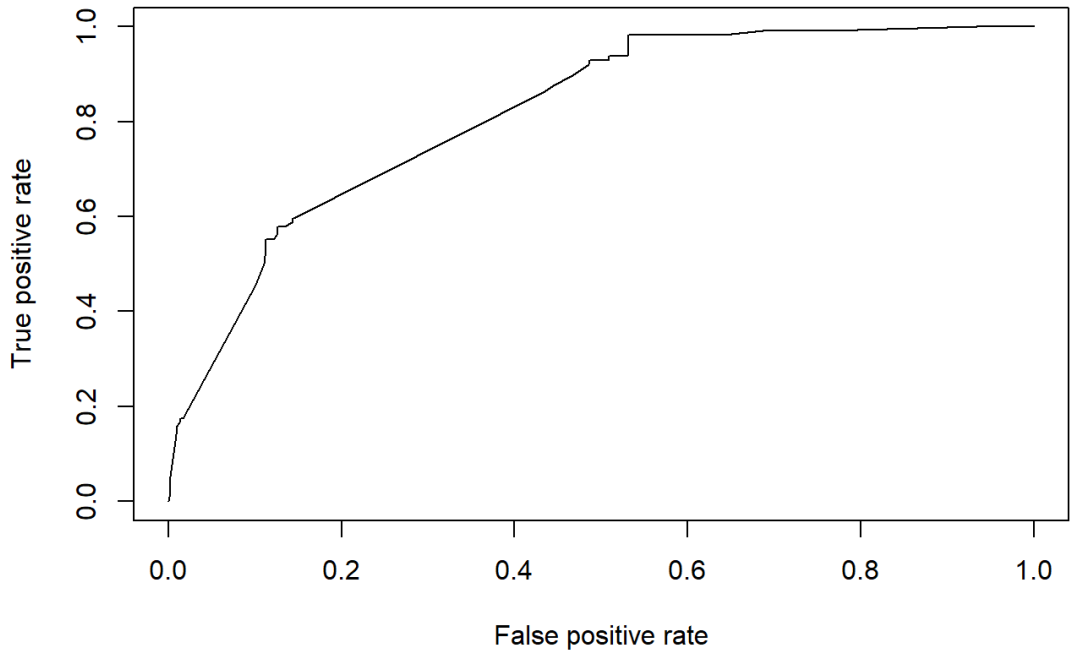
```
# Evaluacion del modelo2
# Matriz de Confusion
matriz_confusion <- table(Test$spam, predicciones)
matriz_confusion
```

```
##    predicciones
##        0    1
##   0 1061    2
##   1  109    5
```

```
error2 <- sum(matriz_confusion[1,2], matriz_confusion[2,1])/sum(matriz
_confusion)
error2
```

```
## [1] 0.09430756
```

```
# mosaic(matriz_confusion, shade = T, colorize = T,
#       gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"),
2, 2)))


# Curva ROC
# library(InformationValue)
# plotROC(sjlabelled::as_numeric(Test$spam, start.at = 0), pred1)


library(ROCR)
pred = ROCR::prediction(pred1, Test$spam)
perf <- performance(pred, "tpr", "fpr")
plot(perf)
```




```
AUCLog1=performance(pred, measure = "auc")@y.values[[1]]
cat("AUC: ",AUCLog1,"n")
```

```
## AUC:  0.817085 n
```

```
#### analisis residuos



###################################
# https://bookdown.org/josefortou/lab-book/logit.html#regresi%C3%B3n-log%C3%ADstica


# otro ejemplo


library(tidyverse)


library(ggeffects) # efectos en modelos de regresion
library(sjPlot) # tablas de regresion
library(haven) # datos en formato .dta
```

```
dat <- read_csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
```

```
dat
```

```
## # A tibble: 400 × 4
##    admit   gre   gpa  rank
##    <dbl> <dbl> <dbl> <dbl>
##  1     0   380  3.61     3
##  2     1   660  3.67     3
##  3     1   800  4        1
##  4     1   640  3.19     4
##  5     0   520  2.93     4
##  6     1   760  3        2
##  7     1   560  2.98     1
##  8     0   400  3.08     2
##  9     1   540  3.39     3
## 10     0   700  3.92     2
## # ⓘ 390 more rows
```

```
# admit: dummy de admisión (0=no admitio; 1=admitido).
# gre: nota en el examen GRE. Puede tomar valores de 200 a 800.
# gpa: promedio crédito acumulado de pregrado. Puede tomar valores de
0 a 4.
# rank: ránquin de la universidad de pregrado. Hay 4 categorías, de ma
yor (1) a menor (4) calidad.


# para recodificar números como categorías
dat <- dat %>%
  mutate(
    admit_fct = case_when(
      admit == 0 ~ "No", # usar los valores de la variable admit
      admit == 1 ~ "Sí"
    ),
    admit_fct = as.factor(admit_fct), # dependiente
```

```
    rank_fct = as.factor(rank) # rank como factor
  )


dat
```

```
## # A tibble: 400 × 6
##    admit   gre   gpa  rank admit_fct rank_fct
##    <dbl> <dbl> <dbl> <dbl> <fct>     <fct>
##  1     0   380  3.61     3 No        3
##  2     1   660  3.67     3 Sí        3
##  3     1   800  4        1 Sí        1
##  4     1   640  3.19     4 Sí        4
##  5     0   520  2.93     4 No        4
##  6     1   760  3        2 Sí        2
##  7     1   560  2.98     1 Sí        1
##  8     0   400  3.08     2 No        2
##  9     1   540  3.39     3 Sí        3
## 10     0   700  3.92     2 No        2
## # ⓘ 390 more rows
```

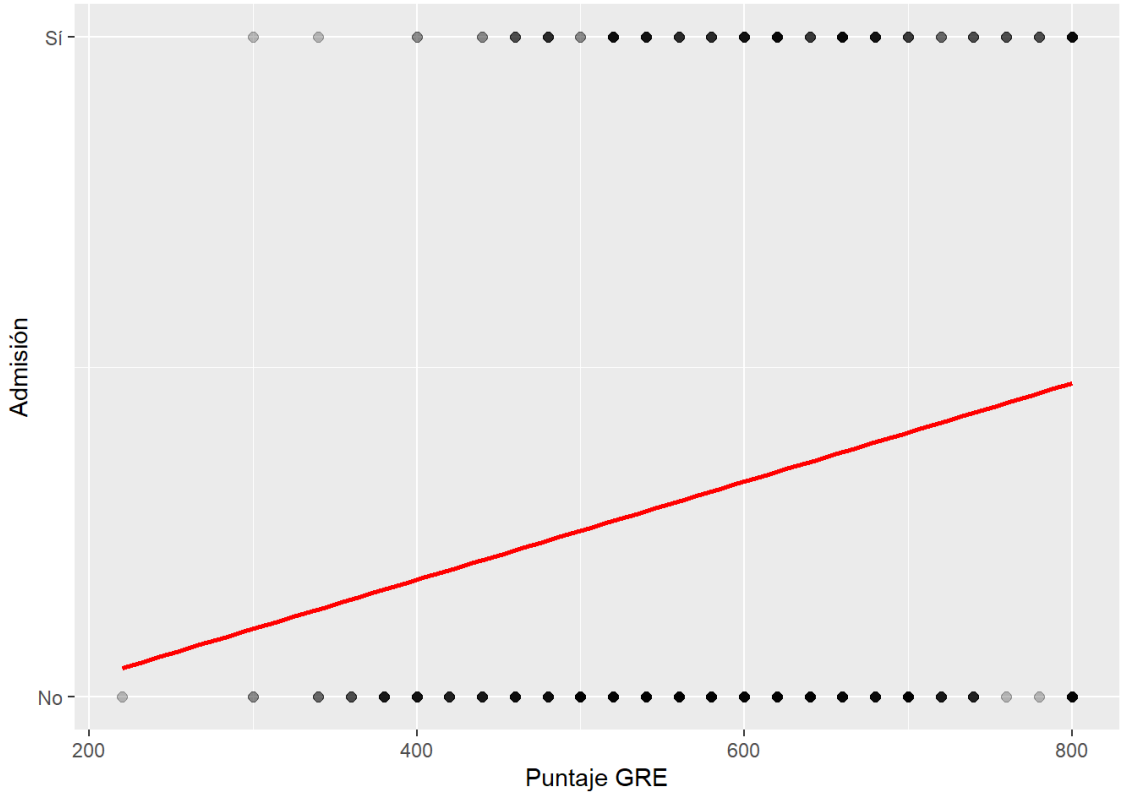
```
# distribucion de la variable dependiente
dat %>%
  ggplot(aes(x = admit_fct)) +
  geom_bar() +
  labs(x = "Admitido",
       y = "Número de observaciones")
```

```
# relacines entre la variable numerica y las dependiente
dat %>%
  group_by(admit_fct) %>%
  summarize(
    obs = n(),
    media_gre = mean(gre),
    media_gpa = mean(gpa)
```

```
  )
```

```
## # A tibble: 2 × 4
##   admit_fct   obs media_gre media_gpa
##   <fct>     <int>     <dbl>     <dbl>
## 1 No          273      573.      3.34
## 2 Sí          127      619.      3.49
```


```
## # A tibble: 2 × 4
##   admit_fct   obs media_gre media_gpa
##   <fct>     <int>     <dbl>     <dbl>
## 1 No          273      573.      3.34
## 2 Sí          127      619.      3.49
```


```
# El model de probabilidad lineal (MPL)
mpl <- lm(
  admit ~ gre + gpa + rank_fct,
  data = dat
```

```
)
summary(mpl)
```

```
##
## Call:
## lm(formula = admit ~ gre + gpa + rank_fct, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7022 -0.3288 -0.1922  0.4952  0.9093
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.2589102  0.2159904  -1.199   0.2314
## gre          0.0004296  0.0002107   2.038   0.0422 *
## gpa          0.1555350  0.0639618   2.432   0.0155 *
## rank_fct2   -0.1623653  0.0677145  -2.398   0.0170 *
## rank_fct3   -0.2905705  0.0702453  -4.137 4.31e-05 ***
## rank_fct4   -0.3230264  0.0793164  -4.073 5.62e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4449 on 394 degrees of freedom
## Multiple R-squared:  0.1004, Adjusted R-squared:  0.08898
## F-statistic: 8.795 on 5 and 394 DF,  p-value: 6.333e-08
```

```
coef(mpl)[[1]] +
  coef(mpl)[[2]]*min(dat$gre) +
  coef(mpl)[[3]]*min(dat$gpa) +
  coef(mpl)[[4]]*0 +
  coef(mpl)[[5]]*0 +
  coef(mpl)[[6]]*1
```

```
## [1] -0.1359216
```

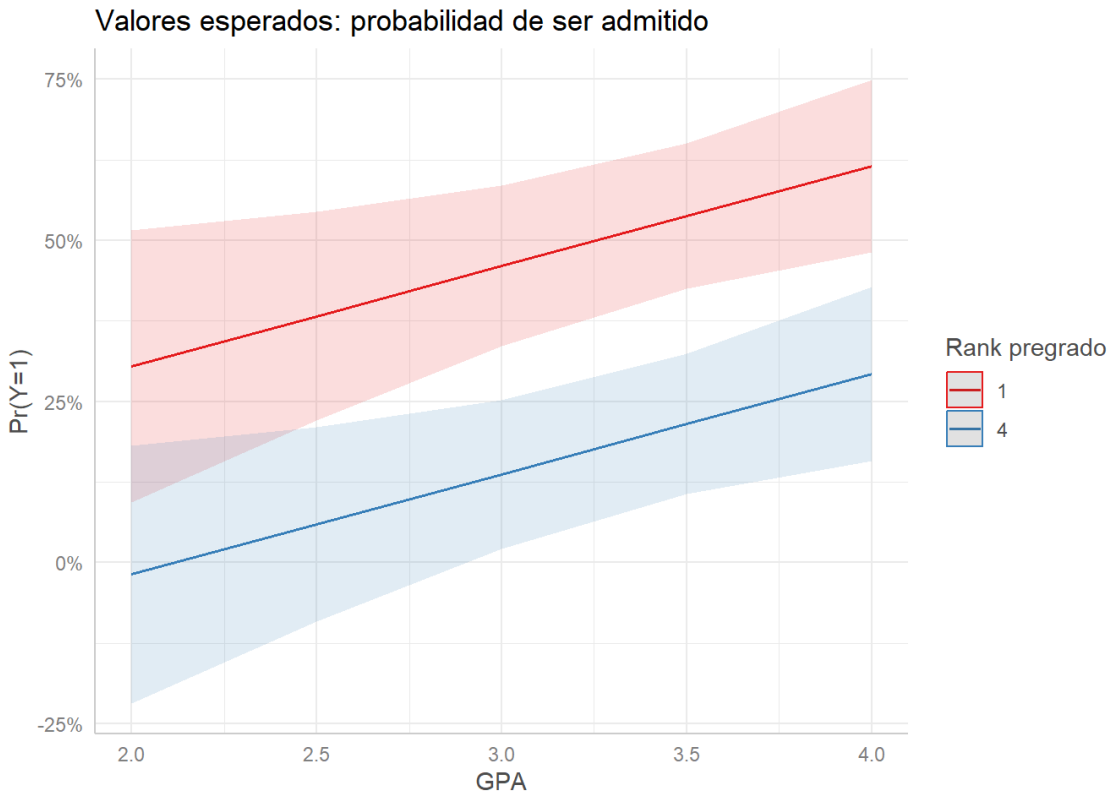
```r
mpl %>%
  predict() %>% # extrae las predicciones o "fitted values" del modelo
  as_tibble() %>%
  ggplot(aes(x = value)) +
  geom_histogram(binwidth = 0.01) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "red") +
  scale_x_continuous(labels = scales::percent) +
  labs(x = expression(hat(Y)), y = "Número de observaciones")
```

```r
mpl %>%
  ggpredict()
```

```
##
## $gpa
## # Predicted values of admit
##
##  gpa | Predicted |       95% CI
## --------------------------------
## 2.00 |      0.30 | [0.09, 0.52]
## 2.50 |      0.38 | [0.22, 0.54]
## 3.00 |      0.46 | [0.34, 0.58]
## 3.50 |      0.54 | [0.43, 0.65]
## 4.00 |      0.62 | [0.48, 0.75]
##
## Adjusted for:
## *      gre = 587.70
## * rank_fct =      1
##
## $rank_fct
## # Predicted values of admit
##
## rank_fct | Predicted |       95% CI
## -----------------------------------
## 1        |      0.52 | [0.41, 0.63]
```

```
## 2         |      0.36 | [0.29, 0.43]
## 3         |      0.23 | [0.15, 0.31]
## 4         |      0.20 | [0.09, 0.31]
##
## Adjusted for:
## * gre = 587.70
## * gpa =   3.39
##
## attr(,"class")
## [1] "ggalleffects" "list"
## attr(,"model.name")
## [1] "."
```

```
mpl %>%
  ggpredict(
    terms = c("gpa", "rank_fct[1, 4]"), # mantiene gre constante en la
media
  ) %>%
  plot() +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Valores esperados: probabilidad de ser admitido",
       x = "GPA", y = "Pr(Y=1)", color = "Rank pregrado")
```




```
# Regresión logística
logit <- glm( # modelos lineales generalizados estimados por MLE
  admit_fct ~ gre + gpa + rank_fct, # formula
  data = dat, # datos
  family = binomial() # tipo de modelo/distribucion
)

glm( # modelos lineales generalizados estimados por MLE
  admit ~ gre + gpa + rank, # formula
  data = dat, # datos
  family = "gaussian" # tipo de modelo/distribucion
) %>%
  summary() # ver coeficientes
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa + rank, family = "gaussian",
##     data = dat)
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.1824127  0.2169695  -0.841   0.4010
## gre          0.0004424  0.0002101   2.106   0.0358 *
## gpa          0.1510402  0.0633854   2.383   0.0176 *
## rank        -0.1095019  0.0237617  -4.608 5.48e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1978669)
##
##     Null deviance: 86.677  on 399  degrees of freedom
## Residual deviance: 78.355  on 396  degrees of freedom
## AIC: 493.07
##
## Number of Fisher Scoring iterations: 2
```

```
summary(logit)
```

```
##
## Call:
## glm(formula = admit_fct ~ gre + gpa + rank_fct, family = binomial()
,
##     data = dat)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.989979   1.139951  -3.500 0.000465 ***
## gre          0.002264   0.001094   2.070 0.038465 *
## gpa          0.804038   0.331819   2.423 0.015388 *
## rank_fct2   -0.675443   0.316490  -2.134 0.032829 *
## rank_fct3   -1.340204   0.345306  -3.881 0.000104 ***
## rank_fct4   -1.551464   0.417832  -3.713 0.000205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.52
##
## Number of Fisher Scoring iterations: 4
```

```
coef(logit) %>%
  as_tibble() %>%
  mutate(odds = exp(value))
```

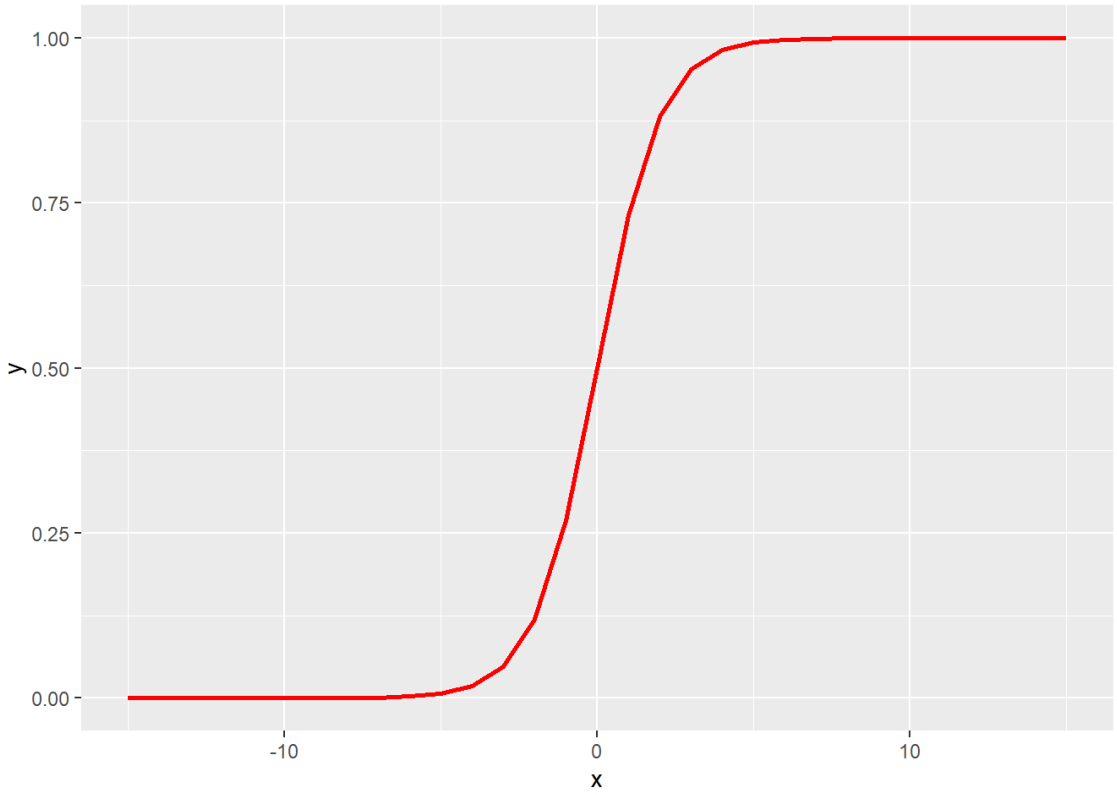
```
## # A tibble: 6 × 2
##      value   odds
##      <dbl>  <dbl>
## 1 -3.99    0.0185
## 2  0.00226 1.00
## 3  0.804   2.23
## 4 -0.675   0.509
## 5 -1.34    0.262
## 6 -1.55    0.212
```

```
tab_model(
  mpl, logit,
  dv.labels = c("MPL", "Logit")
)
```

	MPL			Logit		
Predictors	*Estimates*	*CI*	*p*	*Odds Ratios*	*CI*	*p*
(Intercept)	-0.26	-0.68 – 0.17	0.231	0.02	0.00 – 0.17	**<0.001**
gre	0.00	0.00 – 0.00	**0.042**	1.00	1.00 – 1.00	**0.038**
gpa	0.16	0.03 – 0.28	**0.015**	2.23	1.17 – 4.32	**0.015**

rank fct [2]	-0.16	-0.30 – -0.03	**0.017**	0.51	0.27 – 0.94	**0.033**
rank fct [3]	-0.29	-0.43 – -0.15	**<0.001**	0.26	0.13 – 0.51	**<0.001**
rank fct [4]	-0.32	-0.48 – -0.17	**<0.001**	0.21	0.09 – 0.47	**<0.001**
Observations	400			400		
R^2 / R^2 adjusted	0.100 / 0.089			0.102		

```
# Coeficientes como probabilidades

# exp(coeficiente) / (1 + exp(coeficiente))

tibble(x = -15:15, y = plogis(x)) %>%
  ggplot(aes(x, y)) +
  geom_line(size = 1, color = "red")
```

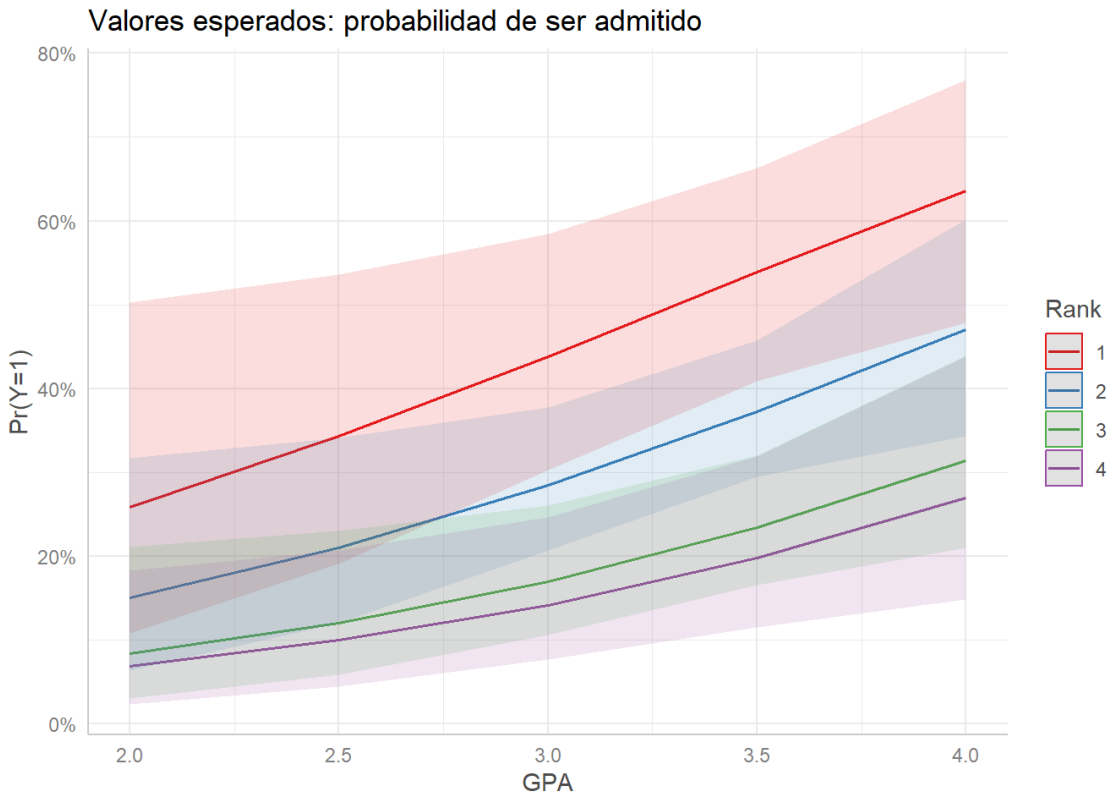



```
coef(logit) %>%
  as_tibble() %>%
  mutate(
    odds = exp(value),
    prob = plogis(value)
```

```
  )
```

```
## # A tibble: 6 × 3
##       value   odds   prob
##       <dbl>  <dbl>  <dbl>
## 1 -3.99     0.0185 0.0182
## 2  0.00226 1.00   0.501
## 3  0.804    2.23   0.691
## 4 -0.675    0.509  0.337
## 5 -1.34     0.262  0.207
## 6 -1.55     0.212  0.175
```

```
logit %>%
  ggpredict(terms = c("gpa", "rank_fct")) %>%
  plot() +
  labs(title = "Valores esperados: probabilidad de ser admitido",
       color = "Rank", x = "GPA", y = "Pr(Y=1)")
```



```
# evaluar el modelo
modelo_aug <- as_tibble(logit$model)
modelo_aug <- modelo_aug %>%
  mutate(fitted = logit$fitted.values,
         residuals = logit$residuals,
         std_residuals = rstandard(logit),
         cooks_d = cooks.distance(logit))
preds <- modelo_aug %>%
  mutate(
    model_prob = plogis(fitted), # probabilidad de ser admitido
    model_pred = if_else(
      model_prob > 0.5, "Sí", "No" # convertir a binario (clasificar)
    )
  )
preds
```

```
## # A tibble: 400 × 10
##    admit_fct   gre   gpa rank_fct fitted residuals std_residuals  c
ooks_d
##    <fct>     <dbl> <dbl> <fct>     <dbl>     <dbl>         <dbl>
<dbl>
##  1 No          380  3.61 3         0.173     -1.21        -0.621 0.
000579
##  2 Sí          660  3.67 3         0.292      3.42         1.58  0.
00450
##  3 Sí          800  4    1         0.738      1.35         0.788 0.
00145
##  4 Sí          640  3.19 4         0.178      5.61         1.87  0.
0132
##  5 No          520  2.93 4         0.118     -1.13        -0.505 0.
000302
##  6 Sí          760  3    2         0.370      2.70         1.43  0.
00631
##  7 Sí          560  2.98 1         0.419      2.39         1.33  0.
00535
##  8 No          400  3.08 2         0.217     -1.28        -0.704 0.
000610
##  9 Sí          540  3.39 3         0.201      4.98         1.80  0.
00573
## 10 No          700  3.92 2         0.518     -2.07        -1.22  0.
00268
```

```
## # ℹ 390 more rows
## # ℹ 2 more variables: model_prob <dbl>, model_pred <chr>
```

```
preds %>%
  group_by(admit_fct, model_pred) %>% # agrupar
  summarize(casos = n()) %>%
  ungroup() %>%
  mutate(prop = casos/sum(casos)) %>%
  pivot_wider(admit_fct, names_from = model_pred, values_from = prop)
# organizar
```

```
## # A tibble: 2 × 2
##   admit_fct    Sí
##   <fct>     <dbl>
## 1 No        0.682
## 2 Sí        0.318
```