Métodos estadísticos para la computación **Curso 2023/24**

Trabajo de modelización estadística.

Hecho por: Ana Martín Conejo, Informática A

Descripción del Data Frame del trabajo:

	Nombre:	Descripción:
Variables para construir la variable dependiente:		
1	peso	Peso del individuo en Kg
2	altura	Altura del individuo en metros
Datos básicos:		
2	sexo	Sexo
3	edad	Edad
Hábitos de vida:		
4	tabaco	Media de consumo de cigarros por semana
5	ubes	Media de consumo de unidades de bebida estándar (UBE) de alcohol por semana
6	carneRoja	Media de veces que come carne roja por semana
7	verduras	Media de veces que consume verduras a la semana
8	deporte	Media veces práctica deporte por semana
9	drogas	Media consumo de otras drogas por mes
10	dietaEsp	Dieta especial por enfermedad (S/N)
Nivel socioeconómico:		
11	nivEstPad	Nivel estudio más alto padres(4=doct o master 3=grado, 2=Bach, 1=Secundaria, 0=menos)
12	nivEstudios	Nivel de estudios (4=doct o master 3=grado, 2=Bach, 1=Secundaria, 0=menos)
13	nivIngresos	Ingresos anuales en el hogar (0:<20k, 1:20k<i<=30k, 2:30k<i<40k, 3:40k<i<60k, 4:>60k

Ejercicio 1.- **Carga en memoria el fichero CSV como tibble, asegurándote de que las variables cualitativas sean leídas como factores.**

Primero observamos cuales son las variables cualitativas: sexo y dietaEsp
A continuación ejecutamos el siguiente código:

```
csv <- read_csv("D:\\1Estadistica\\17206.csv",
                col_types = cols(.default = col_double(),
                                 sexo = col_factor(),
                                 dietaEsp = col_factor()))
```

Vemos a continuación que se cargaron los datos correctamente:

```
# ℹ Use `print(n = ...)` to see more rows
> print(csv)
# A tibble: 5,000 × 14
    peso altura sexo   edad tabaco  ubes carneRoja verduras deporte drogas dietaEsp nivEstPad nivEstudios
   <dbl>  <dbl> <fct> <dbl>  <dbl> <dbl>     <dbl>    <dbl>   <dbl>  <dbl> <fct>        <dbl>       <dbl>
 1  55.4   1.66 M        38      0     0         4        0       0      0 N                0           2
 2  89.8   1.63 M        38      0     6         2        6       6      0 N                1           2
 3  65.4   1.69 M        49      0     1         3        0       1      0 N                0           1
 4  72.1   1.77 V        39      0     0         0        0       0      0 N                1           2
 5  68.3   1.77 V        45    130     0         1        3       0      2 N                1           2
 6  60.8   1.73 V        55      0     0         4        1       0      0 N                0           1
 7 125.    1.78 V        18      0     1         0       23      14      0 N                2           2
 8  74.0   1.7  V        18    220     0         0       10       5      0 N                1           4
 9  64.9   1.81 V        30      0     0         0        0       0      0 N                3           4
10  78.7   1.74 V        32      0     2         3        6       9      0 N                1           3
# ℹ 4,990 more rows
# ℹ 1 more variable: nivIngresos <dbl>
# ℹ Use `print(n = ...)` to see more rows
```

Ejercicio 2.- **Construye una nueva columna llamada IMC que sea igual al peso dividido por la altura al cuadrado. La variable explicada será IMC, las variables explicatorias serán el resto de 12 variables exceptuando peso y altura.**

IMC va a ser igual a IMC= peso/altura^2.
Para acceder a esas variables simplemente hacemos csv$peso y csv$altura.

Luego IMC = csv$peso/(csv$altura^2)

Para añadir la columna IMC a csv usamos add_column()

Luego nuestro código quedará así:

```
#Vamos entonces a añadir a csv la columna IMC con la especificación dada:

csv <- add_column(csv, IMC = csv$peso/(csv$altura^2))
csv
```

El csv final siendo ejecutado para ver el resultado en la línea de comandos:

```
# A tibble: 5,000 × 15
    peso altura sexo   edad tabaco  ubes carneRoja verduras deporte drogas dietaEsp nivEstPad nivEstudios nivIngresos   IMC
   <dbl>  <dbl> <fct> <dbl>  <dbl> <dbl>     <dbl>    <dbl>   <dbl>  <dbl> <fct>        <dbl>       <dbl>       <dbl> <dbl>
 1  55.4   1.66 M        38      0     0         4        0       0      0 N                0           2           2  20.1
 2  89.8   1.63 M        38      0     6         2        6       6      0 N                1           2           1  33.8
 3  65.4   1.69 M        49      0     1         3        0       1      0 N                0           1           1  22.9
 4  72.1   1.77 V        39      0     0         0        0       0      0 N                1           2           2  23.0
 5  68.3   1.77 V        45    130     0         1        3       0      2 N                1           2           3  21.8
 6  60.8   1.73 V        55      0     0         4        1       0      0 N                0           1           0  20.3
 7 125.    1.78 V        18      0     1         0       23      14      0 N                2           2           1  39.4
 8  74.0   1.7  V        18    220     0         0       10       5      0 N                1           4           4  25.6
 9  64.9   1.81 V        30      0     0         0        0       0      0 N                3           4           4  19.8
10  78.7   1.74 V        32      0     2         3        6       9      0 N                1           3           4  26.0
# ℹ 4,990 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

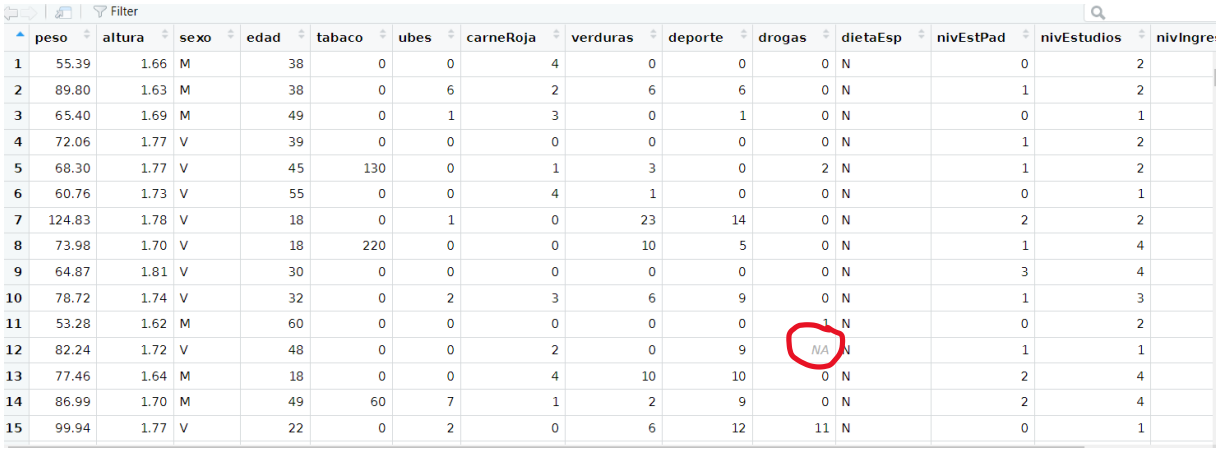
Vemos que se creó la columna IMC correctamente.

Ejercicio 3.- **Elimina completamente las filas que tengan algún valor NA en una de sus columnas.**

Podemos simplemente usar la función na.omit() que nos borrará las filas que contengan algún valor NA de la siguiente forma:

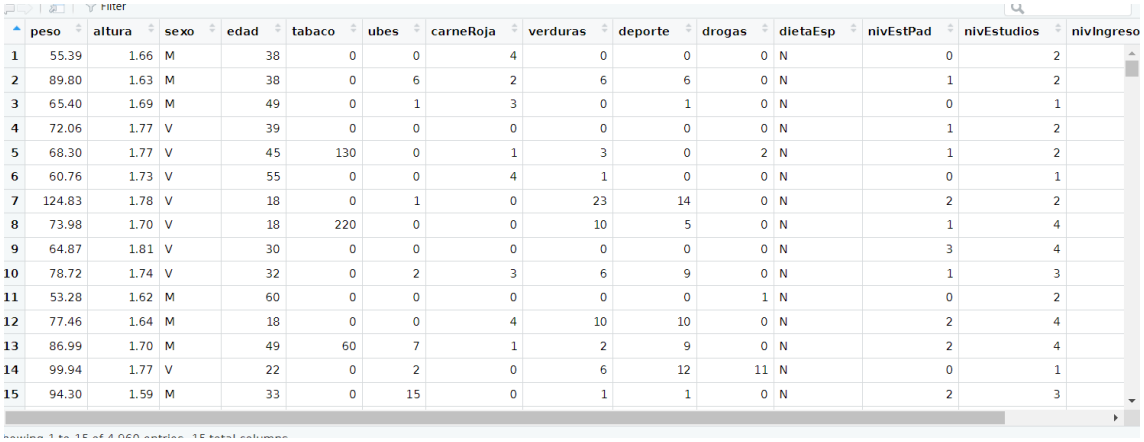
```
csv<- na.omit(csv)
```

Antes de la ejecución:



	peso	altura	sexo	edad	tabaco	ubes	carneRoja	verduras	deporte	drogas	dietaEsp	nivEstPad	nivEstudios	nivIngre
1	55.39	1.66	M	38	0	0	4	0	0	0	N	0	2	
2	89.80	1.63	M	38	0	6	2	6	6	0	N	1	2	
3	65.40	1.69	M	49	0	1	3	0	1	0	N	0	1	
4	72.06	1.77	V	39	0	0	0	0	0	0	N	1	2	
5	68.30	1.77	V	45	130	0	1	3	0	2	N	1	2	
6	60.76	1.73	V	55	0	0	4	1	0	0	N	0	1	
7	124.83	1.78	V	18	0	1	0	23	14	0	N	2	2	
8	73.98	1.70	V	18	220	0	0	10	5	0	N	1	4	
9	64.87	1.81	V	30	0	0	0	0	0	0	N	3	4	
10	78.72	1.74	V	32	0	2	3	6	9	0	N	1	3	
11	53.28	1.62	M	60	0	0	0	0	0	1	N	0	2	
12	82.24	1.72	V	48	0	0	2	0	9	NA	N	1	1	
13	77.46	1.64	M	18	0	0	4	10	10	0	N	2	4	
14	86.99	1.70	M	49	60	7	1	2	9	0	N	2	4	
15	99.94	1.77	V	22	0	2	0	6	12	11	N	0	1	

Después:



	peso	altura	sexo	edad	tabaco	ubes	carneRoja	verduras	deporte	drogas	dietaEsp	nivEstPad	nivEstudios	nivIngreso
1	55.39	1.66	M	38	0	0	4	0	0	0	N	0	2	
2	89.80	1.63	M	38	0	6	2	6	6	0	N	1	2	
3	65.40	1.69	M	49	0	1	3	0	1	0	N	0	1	
4	72.06	1.77	V	39	0	0	0	0	0	0	N	1	2	
5	68.30	1.77	V	45	130	0	1	3	0	2	N	1	2	
6	60.76	1.73	V	55	0	0	4	1	0	0	N	0	1	
7	124.83	1.78	V	18	0	1	0	23	14	0	N	2	2	
8	73.98	1.70	V	18	220	0	0	10	5	0	N	1	4	
9	64.87	1.81	V	30	0	0	0	0	0	0	N	3	4	
10	78.72	1.74	V	32	0	2	3	6	9	0	N	1	3	
11	53.28	1.62	M	60	0	0	0	0	0	1	N	0	2	
12	77.46	1.64	M	18	0	0	4	10	10	0	N	2	4	
13	86.99	1.70	M	49	60	7	1	2	9	0	N	2	4	
14	99.94	1.77	V	22	0	2	0	6	12	11	N	0	1	
15	94.30	1.59	M	33	0	15	0	1	1	0	N	2	3	

Ejercicio 4.- **Calcula las medias y desviaciones típicas (no cuasidesviación) de todas las variables numéricas.**

Para la media creamos la siguiente función:

```
medias<- sapply(csv, function(x) if(is.numeric(x)) mean(x) else NA)
medias
```

Explicación:

sapply() es una función en R que aplica una función a cada elemento de una lista o vector, y devuelve un vector o matriz.
function(x) define una función anónima que se aplicará a cada columna del dataframe csv. x es el nombre del argumento de esta función, que representa cada columna del dataframe.
if(is.numeric(x)) comprueba si la columna x es de tipo numérico usando la función is.numeric(). Si la columna es numérica, se ejecuta la siguiente parte del código, de lo contrario, se devuelve NA.
mean(x) , si la columna es numérica, se calcula la media de los valores en esa columna else NA : Si la columna no es numérica, se devuelve NA.

Nos dará de resultado:

```
> medias<- sapply(csv, function(x) if(is.numeric(x)) mean(x) else NA)
> medias
      peso     altura       sexo       edad     tabaco       ubes   carneRoja    verduras     deporte      drogas
 80.2753286  1.6998407         NA 40.4917339 20.1875000  4.0659274  1.7034274  5.8183468  4.1201613  0.4836694
   dietaEsp  nivEstPad nivEstudios nivIngresos        IMC
         NA  1.2558468   2.1993952   2.1699597 27.7423309
>
```

Para la desviación típica, creamos una función propia ya que la que hay en R nos calcula la cuasidesviación:

```
##Funcion de desviacion típica:
desviacion_tipica <- function(x) {
  return(sd(x) * sqrt((length(x)-1) / length(x)))
}
```

Una vez creada, hacemos lo mismo que con la media:

```
desviaciones_tipicas <- sapply(csv, function(x) if(is.numeric(x)) desviacion_tipica(x) else NA)
desviaciones_tipicas
```

Que nos dará el siguiente resultado:

```
> desviaciones_tipicas <- sapply(csv, function(x) if(is.numeric(x)) desviacion_tipica(x) else NA)
> desviaciones_tipicas
      peso     altura       sexo       edad     tabaco       ubes   carneRoja    verduras     deporte      drogas
19.63726869 0.07066272         NA 14.34118857 41.33050703 5.93141992 2.07212401 6.87312042 4.60317176 1.39546191
   dietaEsp  nivEstPad nivEstudios nivIngresos        IMC
         NA 0.97202093  1.24221756  1.34836550 6.40654523
>
```

Ejercicio 5.- **Calcula los coeficientes de regresión y el coeficiente de determinación para las 12 regresiones lineales unidimensionales de la variable IMC a partir de cada una de las 12 variables separadamente.**

Para que nos quede más sencillo de leer, creamos un vector llamado coeficientes para almacenar los coeficientes:

```
# Creamos coeficientes para almacenar nuestros coeficientes:
coeficientes <- data.frame(Variable = character(),
                           Coeficiente_Regresion = numeric(),
                           Coeficiente_Determinacion = numeric(),
                           stringsAsFactors = FALSE)
```

Como nos pide calcularlos sobre las 12 regresiones lineales unidimensionales de la variable IMC a partir de cada una de las 12 variables separadamente,vamos a iterar por estas sacando el modelo y guardando los resultados en el vector creado:

```
# Iteramos sobre cada variable independiente (nos pide usar las 12)
for (variable in names(csv)[!names(csv) %in% "IMC"]) {

  # Ajustamos el modelo de regresión lineal
  modelo <- lm(IMC ~ csv[[variable]], data = csv)

  # Obtenemos los coeficientes de regresión y R-cuadrado(determinacion)
  coef_regresion <- coef(summary(modelo))[2, 1]
  r_cuadrado <- summary(modelo)$r.squared

  # Guardar los resultados en el vector que creamos
  coeficientes[nrow(coeficientes) + 1, ] <- c(variable, coef_regresion, r_cuadrado)
}
```

Vemos nuestros resultados:

```
+   coeficientes[nrow(coeficientes) + 1, ] <- c(variable, coef_regresion, r_cuadrado)
+ }
> # Mostramos los resultados
> coeficientes
      Variable Coeficiente_Regresion Coeficiente_Determinacion
1         peso     0.305990193582416         0.879689459764508
2       altura     -1.34375210844754      0.000219670266922272
3         sexo    -0.294953737738614      0.000529904346171804
4         edad   -0.0670830612642286        0.0225500882396848
5       tabaco   0.00238118841269095      0.000235983258753677
6         ubes     0.829446816577902         0.589721062198317
7    carneRoja    -0.571745057671797        0.0341970412948339
8     verduras     0.397487858453704         0.181847714893966
9      deporte     0.657830572205458         0.223405918549338
10      drogas    -0.150834565982927       0.00107941870342805
11    dietaEsp     0.437689367735599      0.000188387535315347
12   nivEstPad    -0.755281881724884        0.0131317187319073
13 nivEstudios    -0.851538541956598        0.0272619022405418
14 nivIngresos    -0.779338113921123        0.0269041515838755
>
```

Ejercicio 6.- **Representa los gráficos de dispersión en el caso de variables numéricas y los boxplots en el caso de variables cualitativas. En el caso de las variables numéricas (y sólo en ese caso) el gráfico debe tener sobreimpresa la recta de regresión simple correspondiente.**

Vamos a utilizar la librería ggplot2 para generar gráficos ya que está muy bien. Cargamos entonces la librería:

```
library(ggplot2)
```

Para automatizar la generación de gráficos lo que vamos a hacer es iterar por nombres y guardar los gráficos generados en un directorio nuevo:

```
# Creamos un directorio para almacenar los gráficos si no existe
dir.create("D:/1Estadistica/graficos", showWarnings = FALSE)
```

```
#gráficos de dispersión con recta de regresión para las variables numéricas:
for (variable in names(csv)[!names(csv) %in% c("IMC", "sexo", "dietaEsp")]) {
  modelo <- lm(IMC ~ ., data = csv[, c(variable, "IMC")])
  grafico <- ggplot(data = csv, aes_string(x = variable, y = "IMC")) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    labs(x = variable, y = "IMC") +
    ggtitle(paste("Gráfico de dispersión para", variable))

  # Guardamos el gráfico en el directorio
  ggsave(filename = paste("D:/1Estadistica/graficos/", variable, "_vs_IMC.png", sep = ""), plot = grafico)
}
```

```
# Boxplots para las Variables cualitativas
for (variable in c("sexo", "dietaEsp")) {
  grafico <- ggplot(data = csv, aes_string(x = variable, y = "IMC")) +
    geom_boxplot() +
    labs(x = variable, y = "IMC") +
    ggtitle(paste("Boxplot para", variable))

  # Guardamos el gráfico en el directorio
  ggsave(filename = paste("D:/1Estadistica/graficos/", variable, "_vs_IMC.png", sep = ""), plot = grafico)
}
```

Nos generará los siguientes gráficos:

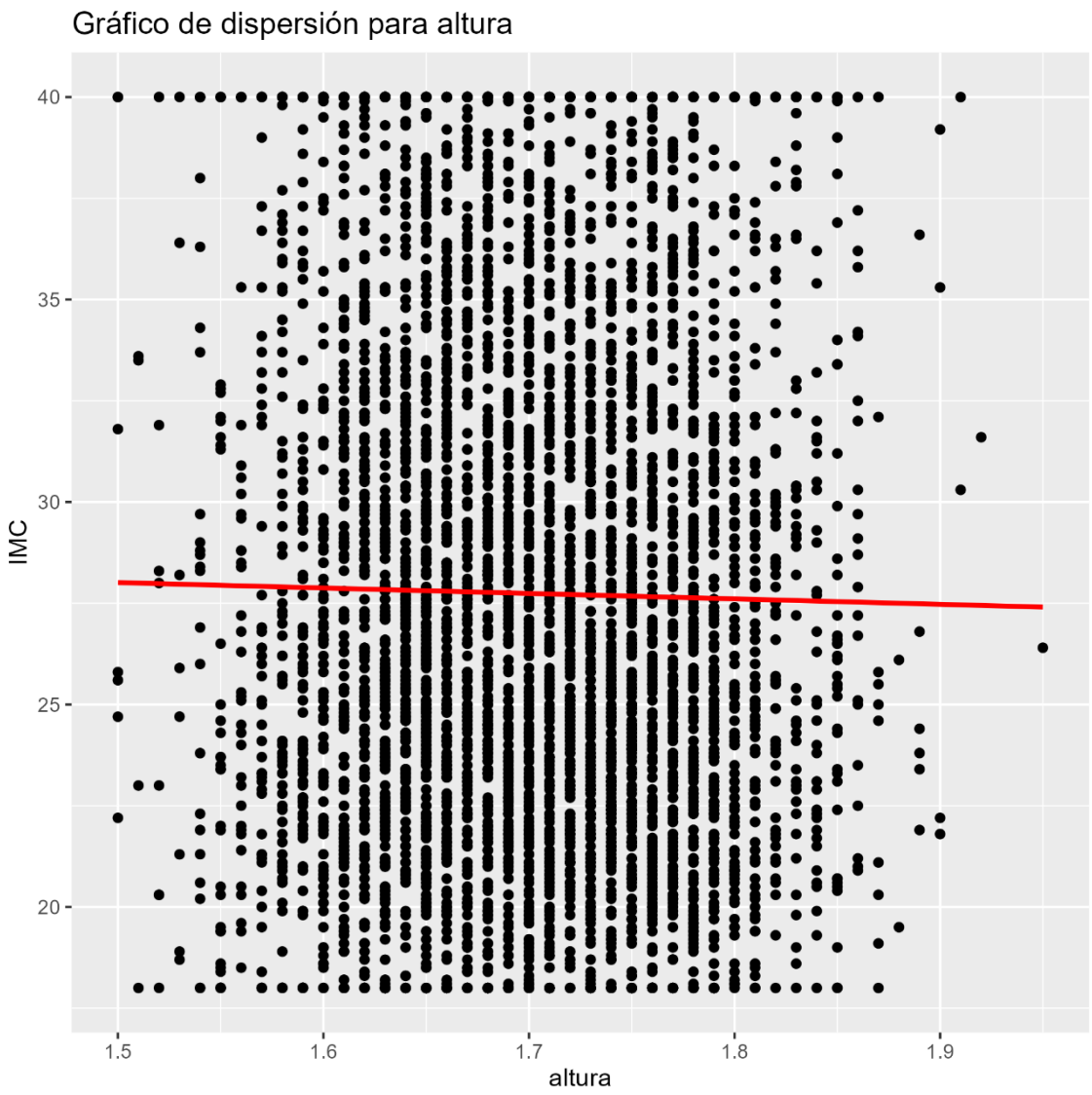


Gráfico de dispersión para altura
IMC
40
35
30
25
20
1.5
1.6
1.7
1.8
1.9
altura

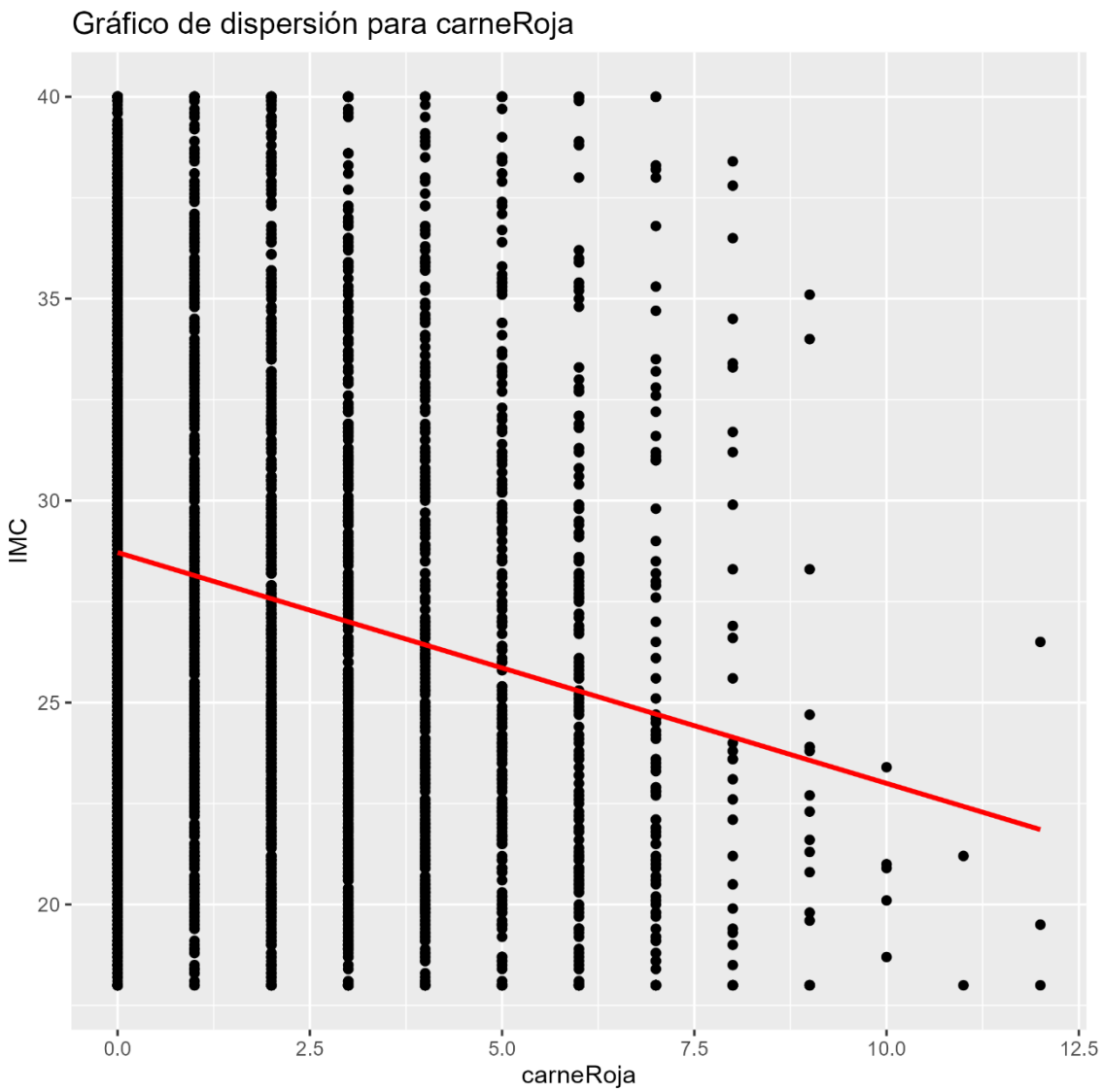


Gráfico de dispersión para carneRoja
IMC
40
35
30
25
20
0.0
2.5
5.0
7.5
10.0
12.5
carneRoja

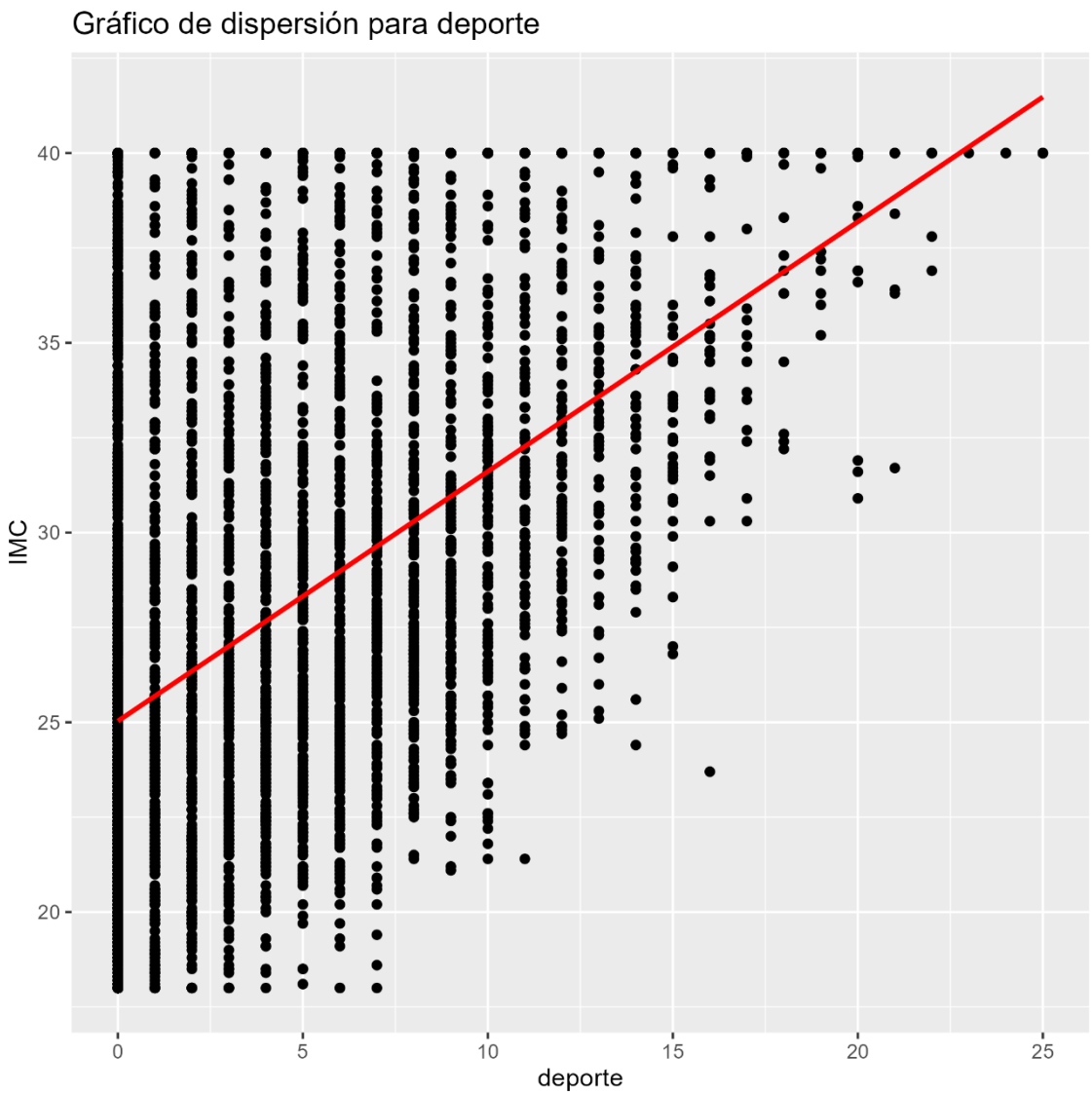


Gráfico de dispersión para deporte
IMC
40
35
30
25
20
0
5
10
15
20
25
deporte

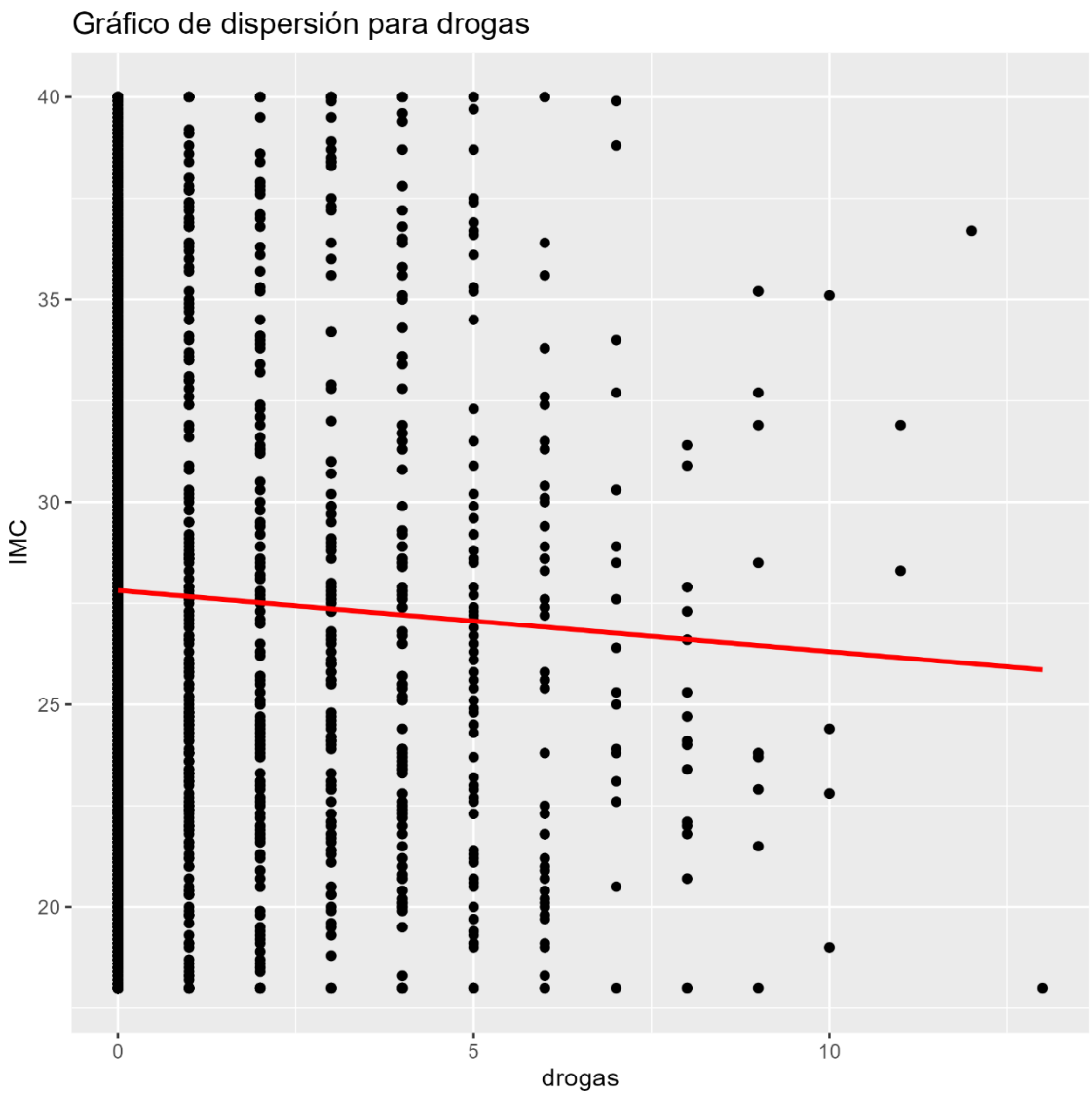


Gráfico de dispersión para drogas
IMC
40
35
30
25
20
0
5
10
drogas

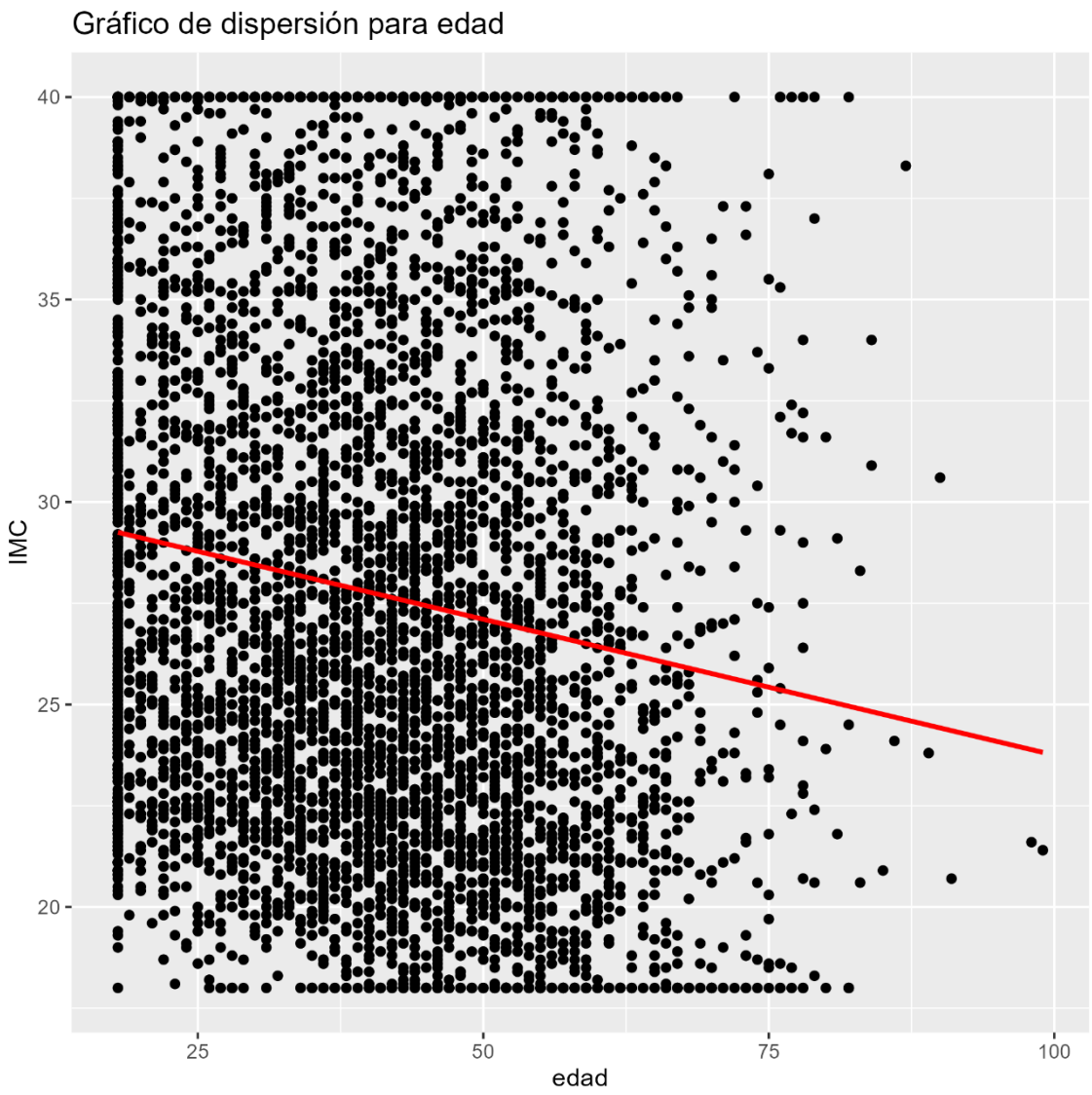


Gráfico de dispersión para edad
IMC
40
35
30
25
20
25
50
75
100
edad

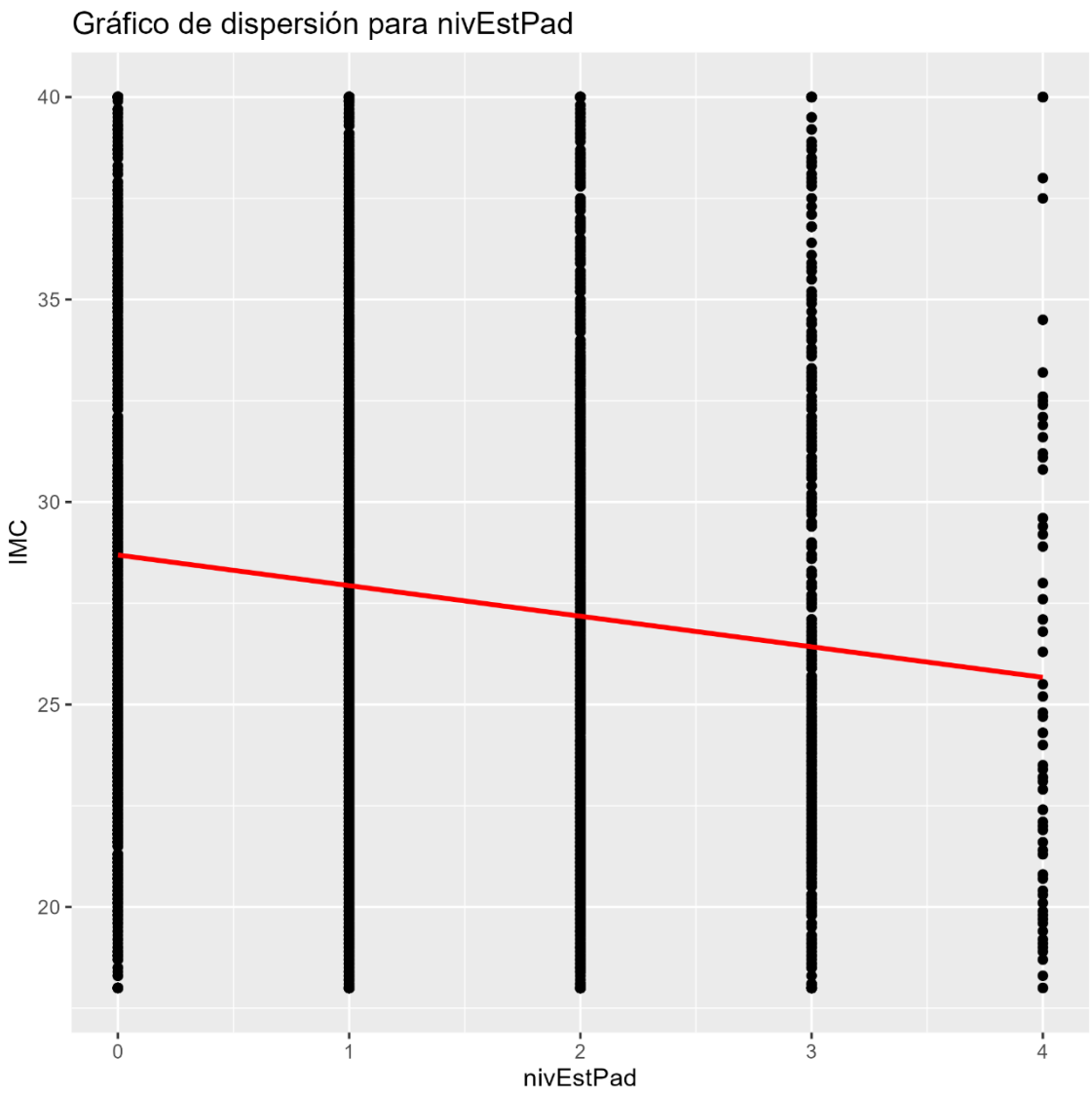


Gráfico de dispersión para nivEstPad
IMC
40
35
30
25
20
0
1
2
3
4
nivEstPad

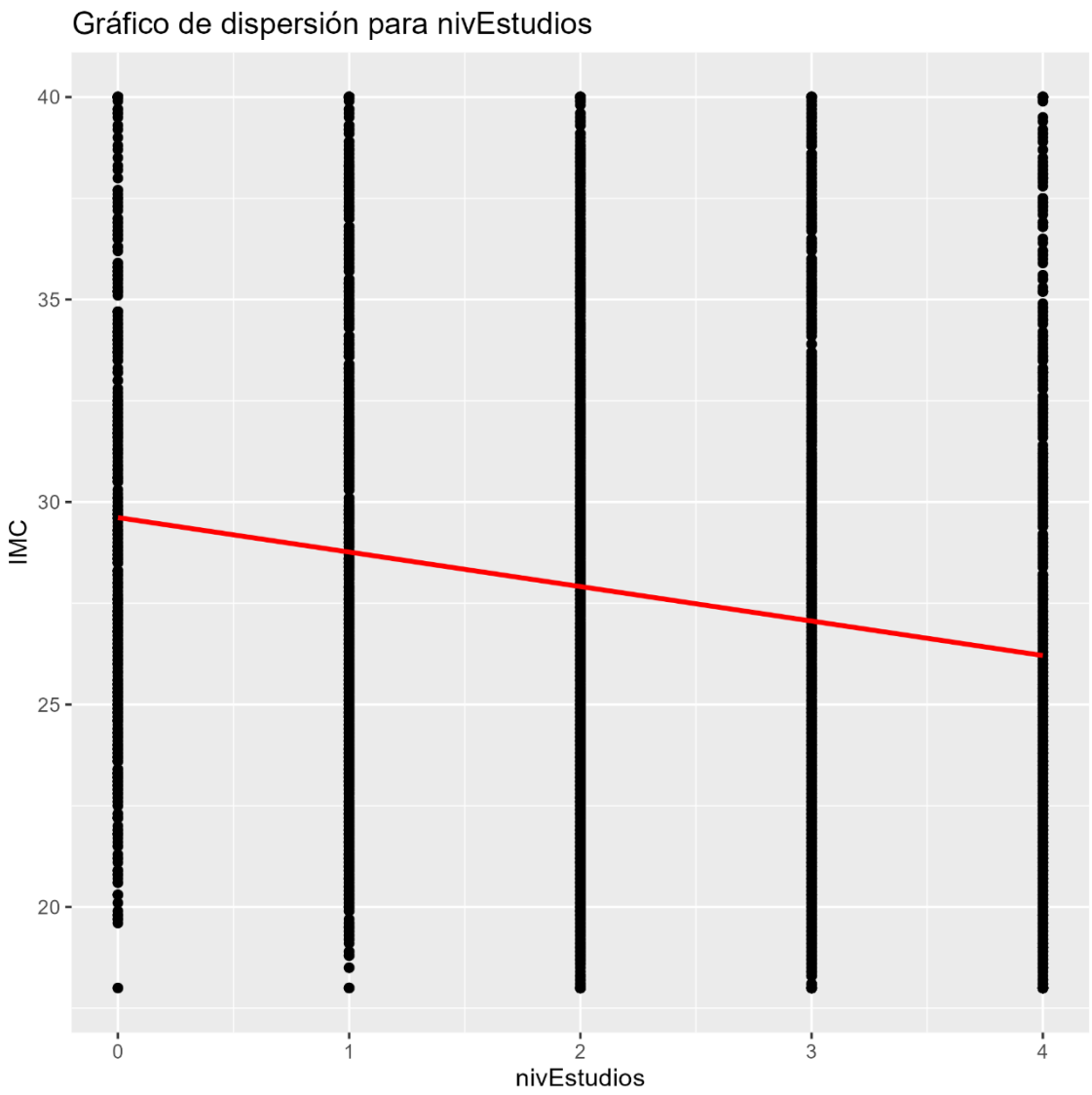


Gráfico de dispersión para nivEstudios
IMC
40
35
30
25
20
0
1
2
3
4
nivEstudios

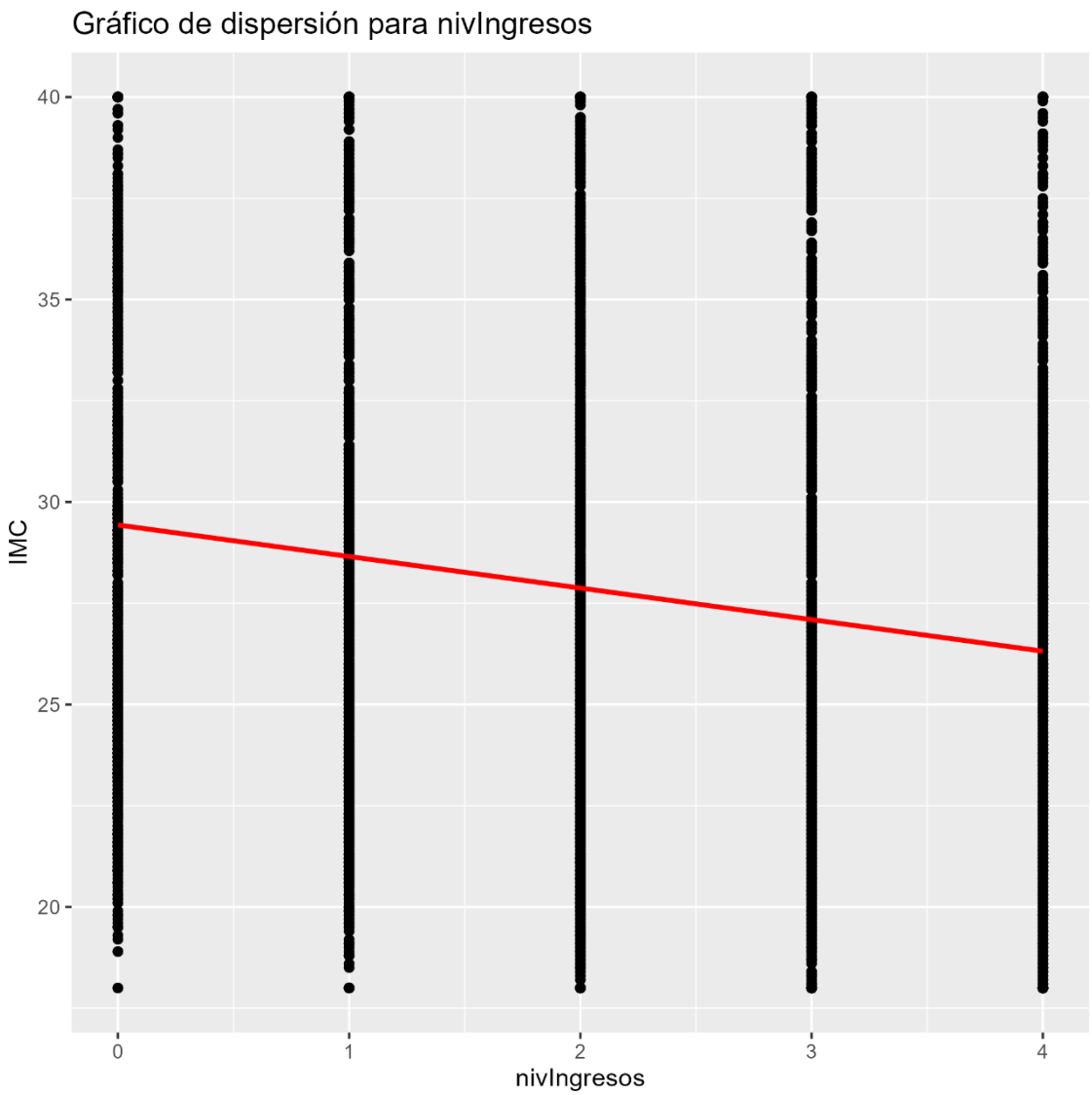


Gráfico de dispersión para nivIngresos
IMC
40
35
30
25
20
0
1
2
3
4
nivIngresos


Gráfico de dispersión para peso
IMC
40
30
20
60
90
120
150
peso

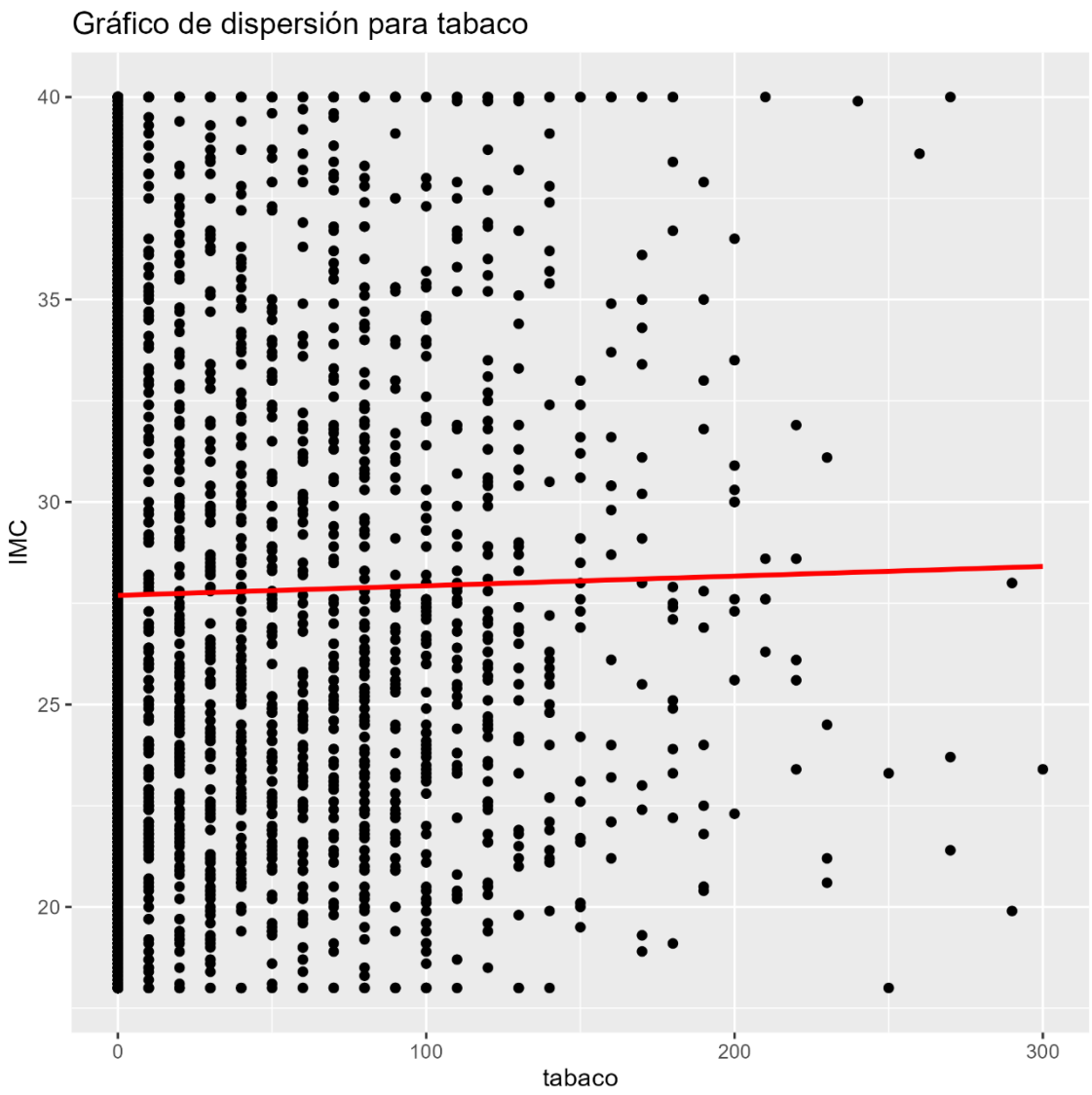


Gráfico de dispersión para tabaco
IMC
40
35
30
25
20
0
100
200
300
tabaco

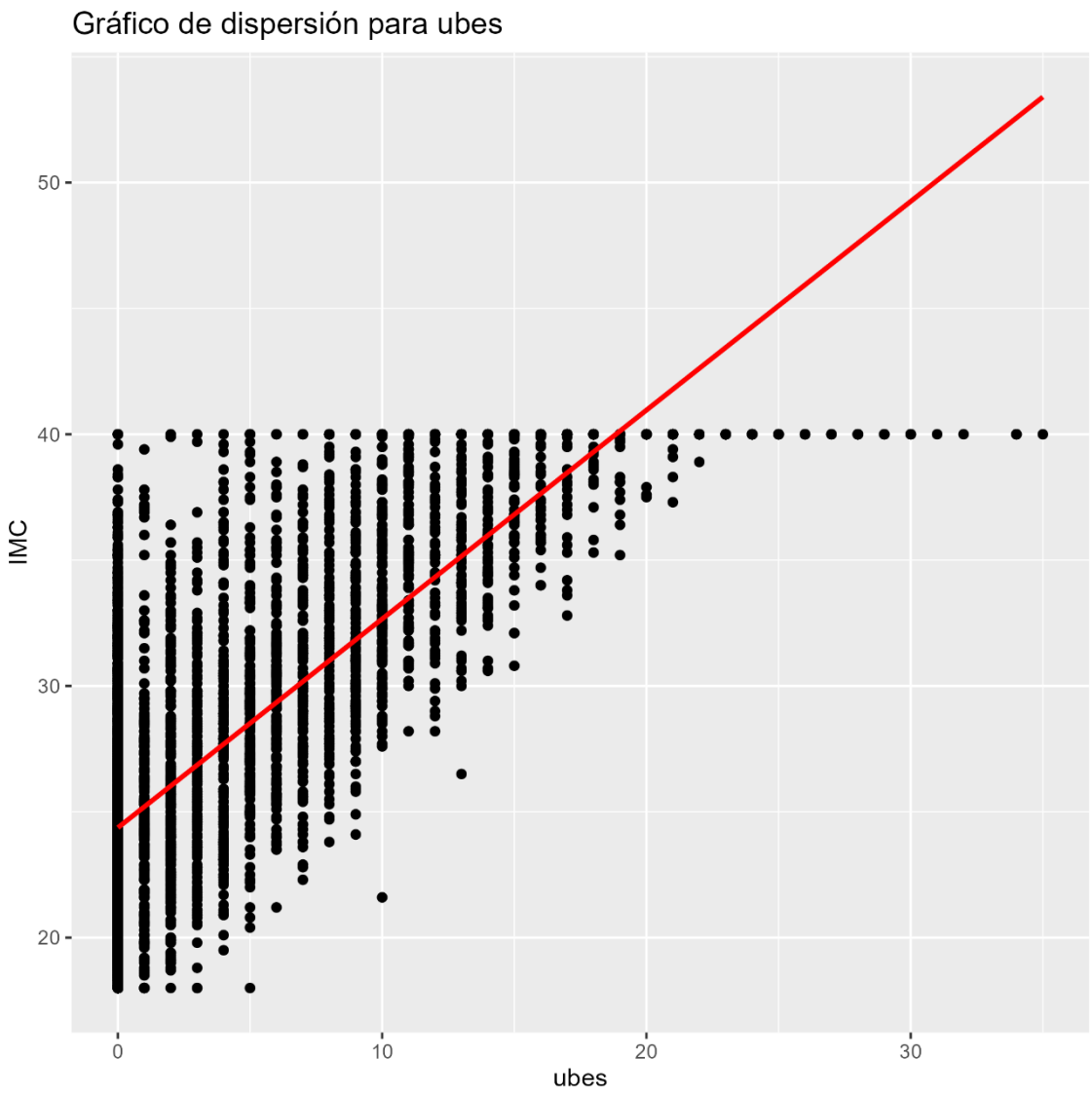


Gráfico de dispersión para ubes
IMC
50
40
30
20
0
10
20
30
ubes

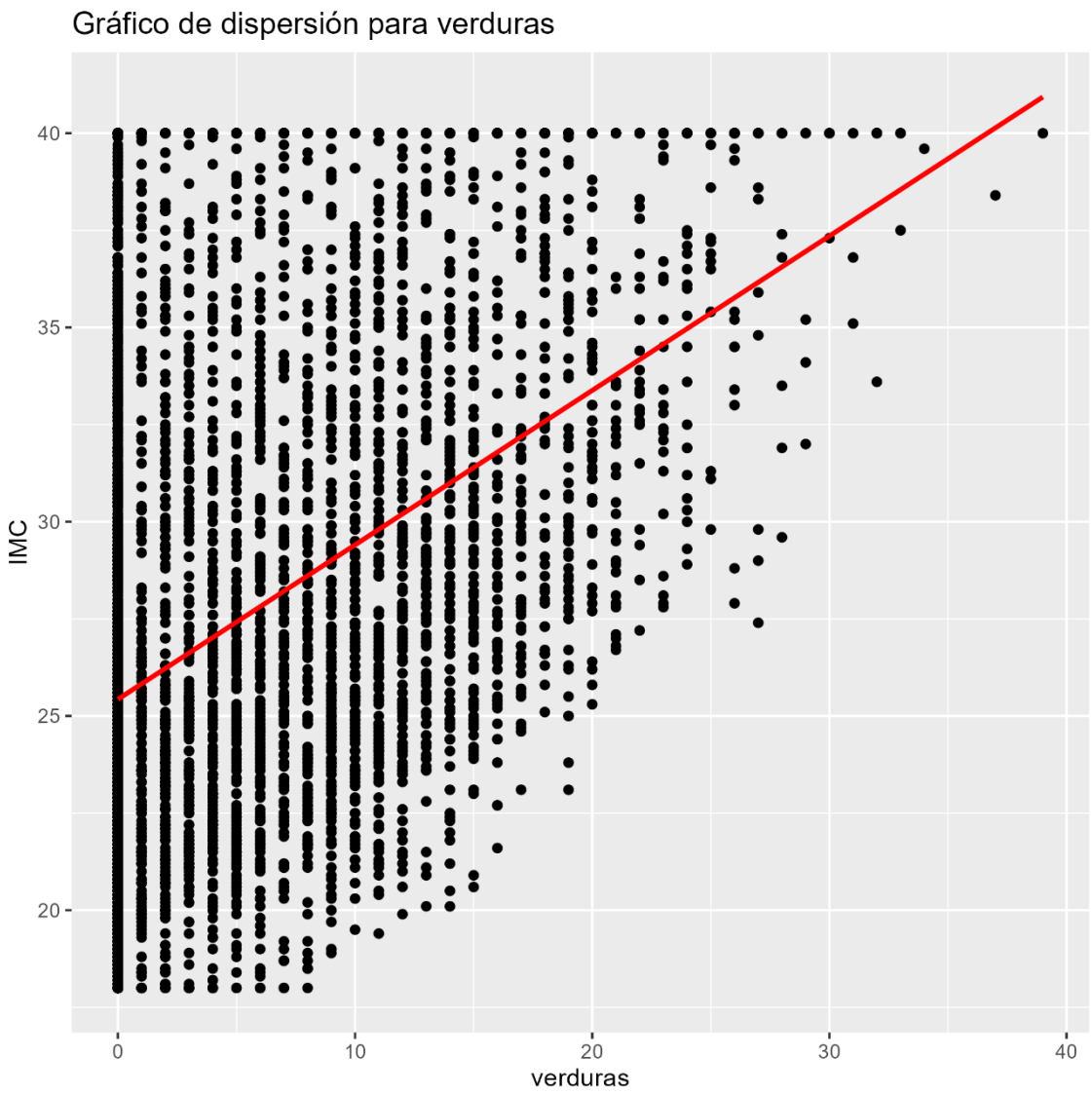
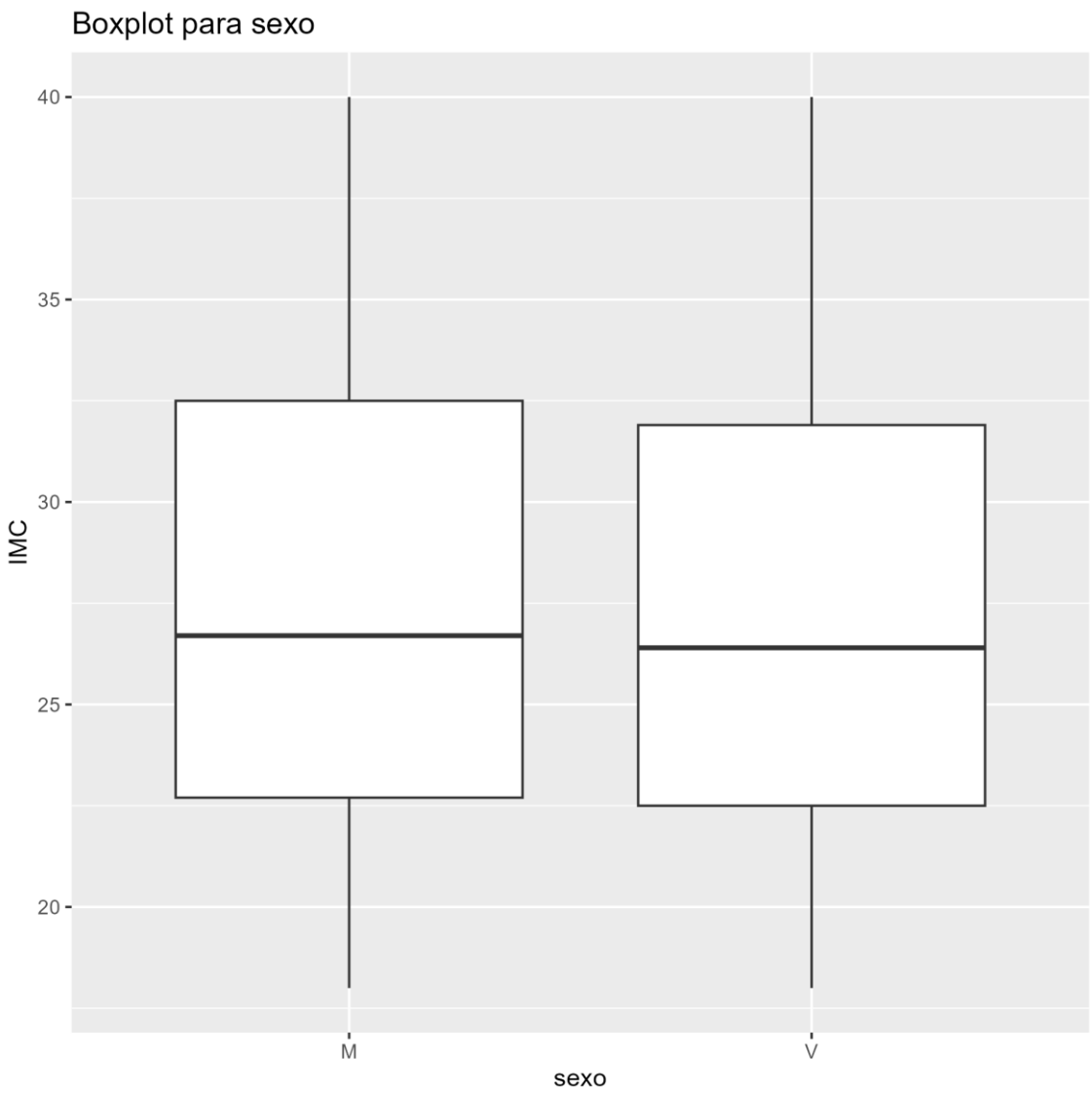
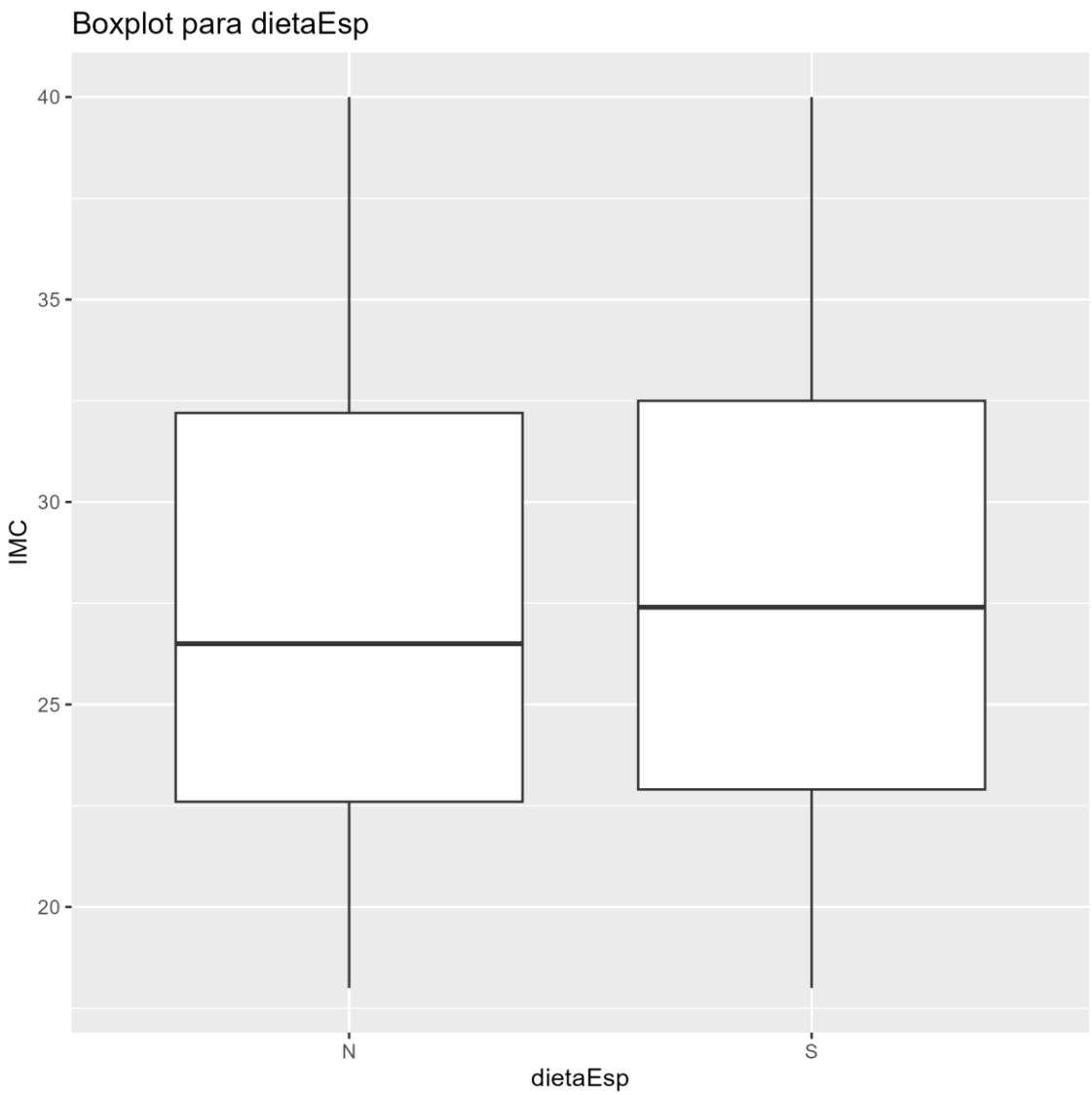


Gráfico de dispersión para verduras
IMC
verduras
40
35
30
25
20
0
10
20
30
40


Boxplot para sexo
40
35
30
25
20
IMC
M
V
sexo


Boxplot para dietaEsp
40
35
30
25
20
IMC
N
S
dietaEsp

Ejercicio 7.- **Separa el conjunto original de datos en tres conjuntos de entrenamiento, test y validación en las proporciones 60%, 20% y 20%.**

Vamos a utilizar la librería caret usada en aprendizaje computacional para entrenar, la cual es muy cómoda también.

Exportamos la librería caret:

```
#Exportamos la librería:
library(caret)
```

Le establecemos una semilla, en este caso me pareció correcto ponerle la siguiente:

```
#Establecemos una semilla:
set.seed(123)
```

A continuación, vamos a dividir el conjunto de entrenamiento y el de prueba y validación en índices:

```
#Creamos las particiones para entrenamiento (60%), prueba (20%) y validación (20%)
indices_entrenamiento <- createDataPartition(csv$IMC, p = 0.6, list = FALSE) #60%
indices_prueba_validacion <- createDataPartition(csv[-indices_entrenamiento, ]$IMC, p = 0.5, list = FALSE)#lo restante se divide 50/50
```

Para el de entrenamiento le damos un porcentaje de 0.6 (60%), y para los demás le damos el resto en porcentaje de 50/50, es decir pruebas se llevará el 20% de csv y validación el otro 20% como nos pide el enunciado.

Una vez divididos, vamos a sacar los conjuntos:

```
#Sacamos el Conjunto de entrenamiento
conjunto_entrenamiento <- csv[indices_entrenamiento, ]
conjunto_entrenamiento
```

```
> conjunto_entrenamiento
# A tibble: 2,977 × 15
    peso altura sexo   edad tabaco  ubes carneRoja verduras deporte drogas dietaEsp nivEstPad nivEstudios nivIngresos   IMC
   <db1>  <db1> <fct> <db1>  <db1> <db1>     <db1>    <db1>   <db1>  <db1> <fct>        <db1>       <db1>       <db1> <db1>
 1  89.8   1.63 M        38      0     6         2        6       6      0 N                1           2           1  33.8
 2  65.4   1.69 M        49      0     1         3        0       1      0 N                0           1           1  22.9
 3  72.1   1.77 V        39      0     0         0        0       0      0 N                1           2           2  23.0
 4  68.3   1.77 V        45    130     0         1        3       0      2 N                1           2           3  21.8
 5 125.    1.78 V        18      0     1         0       23      14      0 N                2           2           1  39.4
 6  78.7   1.74 V        32      0     2         3        6       9      0 N                1           3           4  26.0
 7  53.3   1.62 M        60      0     0         0        0       0      1 N                0           2           1  20.3
 8  77.5   1.64 M        18      0     0         4       10      10      0 N                2           4           3  28.8
 9  87.0   1.7  M        49     60     7         1        2       9      0 N                2           4           3  30.1
10  59.5   1.63 M        60     40     0         0       14       4      0 N                2           4           4  22.4
# ℹ 2,967 more rows
# ℹ Use `print(n = ...)` to see more rows
```

```r
#Sacamos el conjunto de pruebas
conjunto_prueba <- csv[-indices_entrenamiento, ][indices_prueba_validacion, ]
conjunto_prueba
```

```
> conjunto_prueba
# A tibble: 992 × 15
    peso altura sexo   edad tabaco  ubes carneRoja verduras deporte drogas dietaEsp nivEstPad nivEstudios nivIngresos   IMC
   <dbl>  <dbl> <fct> <dbl>  <dbl> <dbl>     <dbl>    <dbl>   <dbl>  <dbl> <fct>        <dbl>       <dbl>       <dbl> <dbl>
 1  55.4   1.66 M        38      0     0         4        0       0      0 N                0           2           2  20.1
 2  60.8   1.73 V        55      0     0         4        1       0      0 N                0           1           0  20.3
 3  74.0   1.7  V        18    220     0         0       10       5      0 N                1           4           4  25.6
 4  64.9   1.81 V        30      0     0         0        0       0      0 N                3           4           4  19.8
 5  99.9   1.77 V        22      0     2         0        6      12     11 N                0           1           0  31.9
 6  55.6   1.59 M        51      0     0         4        6       0      0 N                0           0           0  22.0
 7  66.0   1.68 M        32      0     0         1        4      10      0 N                4           4           4  23.4
 8  98.5   1.73 V        18     80     4         0       12       5      0 N                2           3           3  32.9
 9 112.    1.79 V        42      0     0         0       19      17      0 N                0           1           2  34.9
10  75.1   1.74 V        34      0     0         3        8       3      1 N                1           2           0  24.8
# ℹ 982 more rows
```

```r
#Sacamos el conjunto de validacion
conjunto_validacion <- csv[-c(indices_entrenamiento, indices_prueba_validacion), ]
conjunto_validacion
```

```
> conjunto_validacion
# A tibble: 1,608 × 15
    peso altura sexo   edad tabaco  ubes carneRoja verduras deporte drogas dietaEsp nivEstPad nivEstudios nivIngresos   IMC
   <dbl>  <dbl> <fct> <dbl>  <dbl> <dbl>     <dbl>    <dbl>   <dbl>  <dbl> <fct>        <dbl>       <dbl>       <dbl> <dbl>
 1  60.8   1.73 V        55      0     0         4        1       0      0 N                0           1           0  20.3
 2  99.9   1.77 V        22      0     2         0        6      12     11 N                0           1           0  31.9
 3  94.3   1.59 M        33      0    15         0        1       1      0 N                2           3           3  37.3
 4  56.4   1.62 M        46      0     0         0        4       4      0 N                1           3           2  21.5
 5  55.6   1.59 M        51      0     0         4        6       0      0 N                0           0           0  22.0
 6  98.5   1.73 V        18     80     4         0       12       5      0 N                2           3           3  32.9
 7  50.8   1.68 V        63     80     0         5        0       0      0 N                2           4           4  18.0
 8  75.1   1.74 V        34      0     0         3        8       3      1 N                1           2           0  24.8
 9  53.2   1.64 M        41      0     2         1        0       0      1 N                3           4           4  19.8
10 101.    1.75 M        22      0     2         2       10      13      0 S                2           3           2  32.9
# ℹ 1,598 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Ejercicio 8.- **Selecciona cuál de las 12 variables sería la que mejor explica la variable IMC de manera individual, entrenando con el conjunto de entrenamiento y testeando con el conjunto de test.**

Primero vamos a automatizar el proceso mediante la creación de una función que nos entrena un modelo de regresión lineal para una variable dada:

```
entrenar_y_evaluar_modelo <- function(variable, conjunto_entrenamiento, conjunto_prueba) {
  #Ajustamos el modelo de regresión lineal:
  modelo <- lm(IMC ~ ., data = conjunto_entrenamiento[, c(variable, "IMC")])

  #Predecimos los valores de IMC para el conjunto de prueba:
  predicciones <- predict(modelo, newdata = conjunto_prueba[, variable])
  #Calculamos el coeficiente de determinación (R-cuadrado) para evaluar el desempeño del modelo
  r_cuadrado <- summary(modelo)$r.squared
  #Devolvemos el R-cuadrado del modelo
  return(r_cuadrado)
}
```

Una vez creada esta función, vamos a crear un vector para almacenar los r-cuadrados de los diferentes modelos que calculemos

```
#Creamos un vector para almacenar los R-
r_cuadrados <- numeric(length = 12)
```

A continuación, vamos a iterar por todas las variables independientes y evaluamos su modelo con la función que creamos:

```
#Iteramos sobre las variables independientes y entrenamos y evaluamos un modelo para cada una
for (variable in names(conjunto_entrenamiento)[!names(conjunto_entrenamiento) %in% c("IMC")]) {
  r_cuadrados[variable] <- entrenar_y_evaluar_modelo(variable, conjunto_entrenamiento, conjunto_prueba)
}
```

Ahora seleccionamos la variable con el mayor r-cuadrado, que será la que mejor explique la variable IMC:

```
#Seleccionamos la variable con el mayor R-cuadrado
mejor_variable <- names(r_cuadrados)[which.max(r_cuadrados)]
mejor_variable
```

Ejecutándolo todo tenemos:

```
+ }
> # Creamos un vector para almacenar los R-cuadrados de los modelos
> r_cuadrados <- numeric(length = 12)
> # Iteramos sobre las variables independientes y entrenamos y evaluamos un modelo para cada una
> for (variable in names(conjunto_entrenamiento)[!names(conjunto_entrenamiento) %in% c("IMC")]) {
+   r_cuadrados[variable] <- entrenar_y_evaluar_modelo(variable, conjunto_entrenamiento, conjunto_prueba)
+ }
> # Seleccionamos la variable con el mayor R-cuadrado
> mejor_variable <- names(r_cuadrados)[which.max(r_cuadrados)]
> mejor_variable
[1] "peso"
>
```

Nos da como mejor variable peso, lo cual tiene sentido si nos fijamos en los gráficos, ya que es menos disperso.

Ejercicio 9.- **Selecciona un modelo óptimo lineal de regresión, entrenando en el conjunto de entrenamiento, testeando en el conjunto de test el coeficiente de determinación ajustado y utilizando una técnica progresiva de ir añadiendo la mejor variable.**

Vamos a crearnos otra función para entrenar el modelo ajustado para automatizar más las cosas:

```
#Función para entrenar y evaluar un modelo con una variable dada de forma ajustada:
entrenar_y_evaluar_modelo_ajustado <- function(variables, conjunto_entrenamiento, conjunto_prueba) {
  formula <- as.formula(paste("IMC ~", paste(variables, collapse = "+")))
  modelo <- lm(formula, data = conjunto_entrenamiento)
  r_cuadrado_ajustado <- summary(modelo)$adj.r.squared
  return(r_cuadrado_ajustado)
}
```

Creamos un conjunto de variables disponibles para usar:

```
#Conjunto de variables disponibles para usar
variables_disponibles <- setdiff(names(conjunto_entrenamiento), "IMC")
```

```
> variables_disponibles <- setdiff(names(conjunto_entrenamiento), "IMC")
> variables_disponibles
 [1] "peso"        "altura"      "sexo"        "edad"        "tabaco"      "ubes"        "carneRoja"   "verduras"    "deporte"     "drogas"
[11] "dietaEsp"    "nivEstPad"   "nivEstudios" "nivIngresos"
>
```

A continuación inicializamos el conjunto de variables_seleccionadas y el r_cuadrado_maximo a vacío:

```
#Inicializamos la lista de variables seleccionadas y el coeficiente de determinación ajustado máximo
variables_seleccionadas <- character(0)
r_cuadrado_maximo <- 0
```

Ahora vamos a ir iterando hasta que no haya variables disponibles, y vamos a ir seleccionando la mejor variable en cada caso y añadiéndola a variables disponibles, y cogiendo el r_cuadrado_maximo de estas:

```
#Iteramos para añadir la mejor variable en cada paso
while (length(variables_disponibles) > 0) {
  r_cuadrados <- sapply(variables_disponibles, function(variable) {
    entrenar_y_evaluar_modelo_ajustado(c(variables_seleccionadas, variable), conjunto_entrenamiento, conjunto_prueba)
  })

  mejor_variable <- names(r_cuadrados)[which.max(r_cuadrados)]
  mejor_r_cuadrado <- max(r_cuadrados)

  if (mejor_r_cuadrado > r_cuadrado_maximo) {
    variables_seleccionadas <- c(variables_seleccionadas, mejor_variable)
    r_cuadrado_maximo <- mejor_r_cuadrado
    variables_disponibles <- setdiff(variables_disponibles, mejor_variable)
  } else {
    break
  }
}
```

Entrenamos el modelo final utilizando las variables seleccionadas:

```
#Entrenamos el modelo final utilizando todas las variables seleccionadas
formula_final <- as.formula(paste("IMC ~", paste(variables_seleccionadas, collapse = "+")))
modelo_final <- lm(formula_final, data = conjunto_entrenamiento)
```

Y lo evaluamos en el conjunto de prueba:

```
#Evaluamos el modelo final en el conjunto de prueba
r_cuadrado_ajustado_final <- summary(modelo_final)$adj.r.squared
```

Al final tendremos que:

Las variables seleccionadas por orden de mejor a peor:

```
> variables_seleccionadas
 [1] "peso"       "altura"     "ubes"       "deporte"    "verduras"   "nivEstudios" "edad"       "carneRoja"  "nivIngresos" "drogas"
[11] "nivEstPad"
```

El r_cuadrado_ajustado final:

```
> r_cuadrado_ajustado_final
[1] 0.9935706
```

Ejercicio 10.-.**Evalúa el resultado en el conjunto de validación.**

Primero preparamos los datos de validación, predecimos los valores de IMC en el conjunto de validación:

```
#Vamos a preparar los datos de validación, primero predecimos los valores de IMC
predicciones_validacion <- predict(modelo_final, newdata = conjunto_validacion)
predicciones_validacion
```

```
> predicciones_validacion
       1        2        3        4        5        6        7        8        9       10       11       12       13       14       15       16
20.09999 32.17237 36.08351 22.23591 22.79035 32.83585 18.02757 24.66517 20.44963 33.05628 18.12118 21.89268 23.06507 27.98609 34.71761 21.94903
      17       18       19       20       21       22       23       24       25       26       27       28       29       30       31       32
28.47017 29.49677 17.52118 25.08959 25.44925 28.58456 23.85045 19.11572 35.90888 36.15316 33.21503 24.62270 22.31129 29.11326 20.92725 23.86825
      33       34       35       36       37       38       39       40       41       42       43       44       45       46       47       48
19.29592 25.78076 25.99715 17.37857 40.45714 22.85180 37.32049 39.95084 35.77948 36.04292 22.13170 23.38025 20.93165 26.98940 22.17056 29.10859
      49       50       51       52       53       54       55       56       57       58       59       60       61       62       63       64
28.55973 20.48652 20.28460 25.65029 22.20917 25.74335 24.21103 36.37236 22.92316 30.19771 33.11249 34.44781 22.32343 40.64909 27.10912 41.71145
      65       66       67       68       69       70       71       72       73       74       75       76       77       78       79       80
36.11827 30.38688 34.31006 23.16644 30.83770 30.65222 25.82397 24.50705 41.13549 22.77037 25.11602 39.16386 20.21600 35.26265 18.81032 28.32467
      81       82       83       84       85       86       87       88       89       90       91       92       93       94       95       96
27.91582 28.36866 20.86315 30.28872 22.34027 30.98206 27.56299 31.90019 26.57132 31.13830 29.19640 29.82741 35.16798 27.54722 22.32790 33.32705
      97       98       99      100      101      102      103      104      105      106      107      108      109      110      111      112
18.08725 25.74480 34.48658 25.82561 22.36358 22.73274 24.06263 26.20110 22.53025 24.50533 22.88314 21.65579 22.61453 29.28036 20.56179 26.19391
     113      114      115      116      117      118      119      120      121      122      123      124      125      126      127      128
```

Y ahora ajustamos sus r_cuadrados:

```
241 #Luego calculamos los r_cuadrado_ajustado
242 r_cuadrado_ajustado_validacion <- 1 - (sum((conjunto_validacion$IMC - predicciones_validacion)^2) / sum((conjunto_validacion$IMC - mean(conjunto_validacion$IMC))^2))
243
244 r_cuadrado_ajustado_validacion
245
```

```
> r_cuadrado_ajustado_validacion
[1] 0.9940032
```

Este coeficiente nos indica que el modelo explica aproximadamente el 99.4% de la variabilidad en la variable de respuesta (IMC) en el conjunto de datos de validación. Lo que sugiere que el modelo tiene un buen ajuste y capacidad predictiva en datos no vistos.

Un valor tan alto de coeficiente de determinación ajustado en el conjunto de validación es prometedor, y sugiere que el modelo es capaz de generalizar bien nuevos datos, lo cual es una buena señal de que el modelo es robusto y fiable.

Ejercicio 11.- **Lee el dataframe de evaluación que te habrá llegado (eval.csv) y utiliza el modelo creado para añadirle una nueva columna con el valor de la variable IMC y, a continuación, otra columna con el valor de la variable Peso. Salva el resultado como evalX.csv para enviarlo como parte de la solución al trabajo**

Leemos el dataframe de evaluación:

```
# Leer el dataframe de evaluación
evaluacion <- read_csv("D:\\1Estadistica\\eval.csv",col_types = cols(.default = col_double(),
                                                                     sexo = col_factor(),
                                                                     dietaEsp = col_factor()))
evaluacion
```

Modificamos la fórmula del modelo final para excluir la variable de peso, ya que el conjunto de evaluación no contiene esta columna. Utilizamos la función update para hacer esta modificación en la fórmula original del modelo.

Ajustamos un nuevo modelo utilizando la fórmula actualizada, pero esta vez sin la variable de peso. Esto lo hacemos utilizando la función lm.

```
#Ajustamos el modelo y la fórmula ya que evaluacion no contiene la columna peso
formula_final_sin_peso <- update(formula_final, . ~ . - peso)
modelo_final_sin_peso <- lm(formula_final_sin_peso, data = conjunto_entrenamiento)
```

Utilizamos el modelo final sin la variable de peso para predecir el IMC en el conjunto de evaluación. Esto lo hacemos utilizando la función predict.

```
#Predecimos el IMC en el conjunto de evaluación utilizando el modelo final sin la variable de peso
evaluacion$IMC_predicho <- predict(modelo_final_sin_peso, newdata = evaluacion)
```

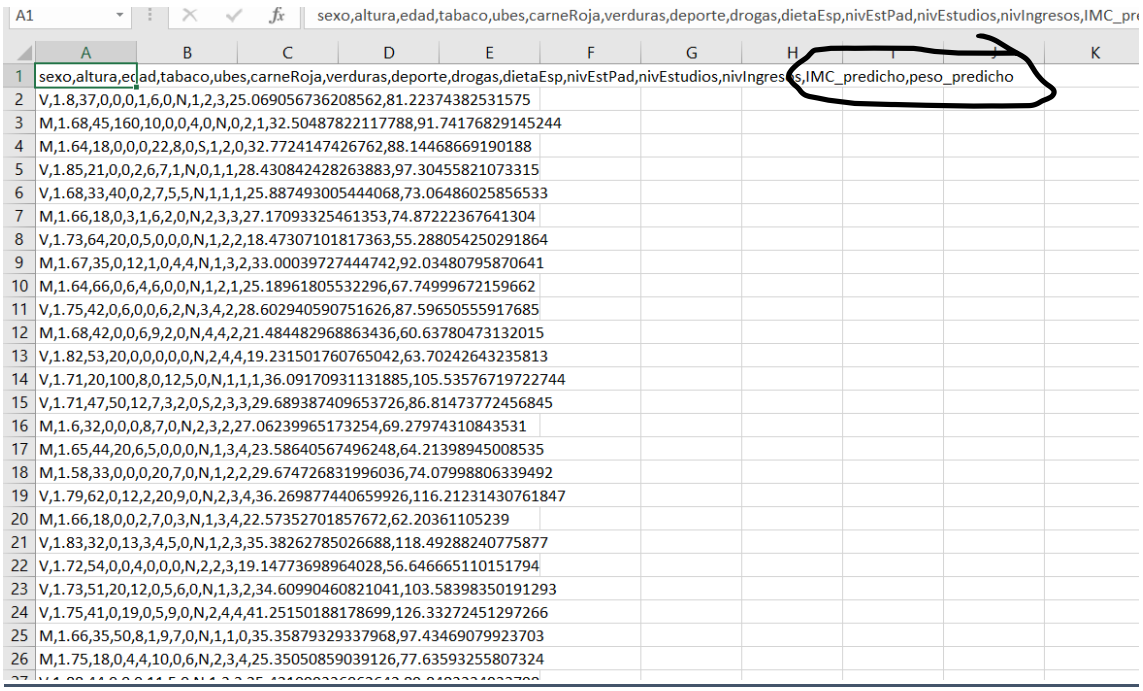
Calculamos el peso basado en el IMC predicho utilizando la fórmula del IMC, esto lo hacemos multiplicando el IMC predicho por la altura al cuadrado para cada fila del conjunto de evaluación.

```
#Calculamos el peso basado en IMC predicho
evaluacion$peso_predicho <- evaluacion$IMC_predicho * (evaluacion$altura^2)
```

```
> evaluacion
# A tibble: 1,000 × 15
   sexo  altura  edad tabaco  ubes carneRoja verduras deporte drogas dietaEsp nivEstPad nivEstudios nivIngresos IMC_predicho peso_predicho
   <fct>  <db1> <db1>  <db1> <db1>     <db1>    <db1>   <db1>  <db1> <fct>        <db1>       <db1>       <db1>       <db1>        <db1>
 1 V        1.8    37      0     0         0        1       6      0 N                1           2           3        25.1         81.2
 2 M        1.68   45    160    10         0        0       4      0 N                0           2           1        32.5         91.7
 3 M        1.64   18      0     0         0       22       8      0 S                1           2           0        32.8         88.1
 4 V        1.85   21      0     0         2        6       7      1 N                0           1           1        28.4         97.3
 5 V        1.68   33     40     0         2        7       5      5 N                1           1           1        25.9         73.1
 6 M        1.66   18      0     3         1        6       2      0 N                2           3           3        27.2         74.9
 7 V        1.73   64     20     0         5        0       0      0 N                1           2           2        18.5         55.3
 8 M        1.67   35      0    12         1        0       4      4 N                1           3           2        33.0         92.0
 9 M        1.64   66      0     6         4        6       0      0 N                1           2           1        25.2         67.7
10 V        1.75   42      0     6         0        0       6      2 N                3           4           2        28.6         87.6
# ℹ 990 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

Y por último guardamos los resultados en el fichero.

```
#Guardamos el resultado como evalX.csv
write_csv(evaluacion, "D:\\1Estadistica\\eval.csv")
```



	A
1	sexo,altura,edad,tabaco,ubes,carneRoja,verduras,deporte,drogas,dietaEsp,nivEstPad,nivEstudios,nivIngresos,IMC_predicho,peso_predicho
2	V,1.8,37,0,0,0,1,6,0,N,1,2,3,25.069056736208562,81.22374382531575
3	M,1.68,45,160,10,0,0,4,0,N,0,2,1,32.50487822117788,91.74176829145244
4	M,1.64,18,0,0,0,22,8,0,S,1,2,0,32.7724147426762,88.14468669190188
5	V,1.85,21,0,0,2,6,7,1,N,0,1,1,28.430842428263883,97.30455821073315
6	V,1.68,33,40,0,2,7,5,5,N,1,1,1,25.887493005444068,73.06486025856533
7	M,1.66,18,0,3,1,6,2,0,N,2,3,3,27.17093325461353,74.87222367641304
8	V,1.73,64,20,0,5,0,0,0,N,1,2,2,18.47307101817363,55.288054250291864
9	M,1.67,35,0,12,1,0,4,4,N,1,3,2,33.00039727444742,92.03480795870641
10	M,1.64,66,0,6,4,6,0,0,N,1,2,1,25.18961805532296,67.74999672159662
11	V,1.75,42,0,6,0,0,6,2,N,3,4,2,28.602940590751626,87.59650555917685
12	M,1.68,42,0,0,6,9,2,0,N,4,4,2,21.484482968863436,60.63780473132015
13	V,1.82,53,20,0,0,0,0,0,N,2,4,4,19.231501760765042,63.70242643235813
14	V,1.71,20,100,8,0,12,5,0,N,1,1,1,36.09170931131885,105.53576719722744
15	V,1.71,47,50,12,7,3,2,0,S,2,3,3,29.689387409653726,86.81473772456845
16	M,1.6,32,0,0,0,8,7,0,N,2,3,2,27.06239965173254,69.27974310843531
17	M,1.65,44,20,6,5,0,0,0,N,1,3,4,23.58640567496248,64.21398945008535
18	M,1.58,33,0,0,0,20,7,0,N,1,2,2,29.674726831996036,74.07998806339492
19	V,1.79,62,0,12,2,20,9,0,N,2,3,4,36.269877440659926,116.21231430761847
20	M,1.66,18,0,0,2,7,0,3,N,1,3,4,22.57352701857672,62.20361105239
21	V,1.83,32,0,13,3,4,5,0,N,1,2,3,35.38262785026688,118.49288240775877
22	V,1.72,54,0,0,4,0,0,0,N,2,2,3,19.14773698964028,56.646665110151794
23	V,1.73,51,20,12,0,5,6,0,N,1,3,2,34.60990460821041,103.58398350191293
24	V,1.75,41,0,19,0,5,9,0,N,2,4,4,41.25150188178699,126.33272451297266
25	M,1.66,35,50,8,1,9,7,0,N,1,1,0,35.35879329337968,97.43469079923703
26	M,1.75,18,0,4,4,10,0,6,N,2,3,4,25.35050859039126,77.63593255807324

Ejercicio 12.- **Expresa tus conclusiones sobre el modelo creado. Incluyendo, al menos, respuestas a las siguientes cuestiones:**

- Qué utilidad podría tener el modelo matemático que has obtenido.
- Qué se puede deducir a partir del modelo sobre la relación entre las variables.
- Problemas que has encontrado en el desarrollo.
- Qué te ha llamado la atención en el proceso.
- Qué más podría hacerse y cómo plantearlo

El modelo matemático que hemos obtenido proporciona una herramienta para predecir el Índice de Masa Corporal (IMC) de un individuo utilizando variables como el sexo, la edad, la altura,...
Esta predicción puede ser útil en entornos médicos, de salud pública o de análisis de datos relacionados con la salud.

A través del modelo, podemos observar la relación entre las variables utilizadas y el IMC. Por ejemplo, podemos determinar qué tan significativas son el sexo, la edad y la dieta en la predicción del IMC.
Además, podemos entender cómo estas variables interactúan entre sí para influir en el IMC de una persona.

Durante el desarrollo, enfrenté algunos desafíos, como la falta de ciertos datos o la presencia de valores nulos en el conjunto de datos. Además, la selección de variables puede ser un proceso complejo y subjetivo que requiere cierta exploración y análisis.

Me llamó la atención la importancia de la selección de variables y cómo esto puede afectar significativamente la calidad y precisión del modelo.
También me sorprendió la complejidad de las relaciones entre las variables y cómo pueden variar según el contexto y la población estudiada.

Para mejorar el modelo, podríamos considerar la inclusión de más variables o la exploración de técnicas de modelado más avanzadas, como el aprendizaje automático. Además, podríamos realizar un análisis más detallado de la influencia de cada variable en el IMC y su interacción con otras variables.
También podríamos explorar cómo el modelo se comporta en diferentes subpoblaciones o contextos específicos.

En resumen, este modelo es una herramienta útil para las predicciones sobre el Índice de masa corporal, pero se tienen que tener en cuenta las variables usadas y las relaciones entre estas para poder determinar un modelo apropiado.
Se podría mejorar este modelo entrando en más detalle en aplicaciones vistas en Aprendizaje Computacional.