TRABAJO R

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

Para cargar el archivo lo primero que haré será cargar la librería readr para cargar el archivo como tibble, en esta importación además asignare a cada variable su tipo de variable.

```
#Primero empiezo importando el archivo pero usando la libreria readr y con barra baja en lugar
#de punto para que devuelva un tibble en vez de un data frame, asignare tipos a las columnas ya que cada una
#representa un tipo de datos distinto, por ejemplo las de ingresos, sexo o estudios que representan variables categóricas
library(readr)
library(tidyverse)
datos <- read_csv("18608.csv", col_types=cols(.default=col_double(), sexo=col_factor(),
                                              dietaEsp=col_factor(), nivEstPad=col_factor(),
                                              nivEstudios=col_factor(), nivIngresos=col_factor()))
```

Resultado:

```
> library(readr)
> datos <- read_csv("18608.csv")
Rows: 5000 Columns: 14
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (2): sexo, dietaEsp
dbl (12): peso, altura, edad, tabaco, ubes, carneRoja, verduras, deporte, drogas, nivEstPad, nivEstudios, nivIngresos

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Se han añadido las variables correctamente

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.

Para crear la columna de IMC creo una función con la formula del IMC que es peso en kg dividido entre altura al cuadrado y realizo un mutate para añadir la columna a la tabla.

```
#Ahora cargare la operacion para calcular el IMC en la variable, la formula del imc es peso en kg dividido entre la altura en metros al cuadrado,
#el peso y la altura en este caso seran variables dependientes mientras que el resto seran independientes
calc_imc<- function(peso, altura){
  imc<- peso/altura^2
}
datos <- datos %>% mutate(IMC=calc_imc(datos$peso, datos$altura))
```

Resultado:

```
> datos
# 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>    <fct>     <fct>       <fct>       <dbl>
 1  76.6   1.71 M        56      0     0         0       10       7      0 N        1         1           3            26.2
 2  46.7   1.61 M        41      0    15         0        0       7      0 N        2         2           2            18.0
 3  45.5   1.59 M        37    120     0         0        0       6      0 N        0         0           0            18.0
 4  62.2   1.8  V        66      0     9         1        0       3      0 N        1         3           4            19.2
 5  55.8   1.76 V        45     40     8         0        0       0      0 N        1         3           4            18.0
 6  86.1   1.83 V        43      0     0         1        0       2      0 N        2         1           1            25.7
 7  66.6   1.67 M        18      0     0         1        0       0      0 N        3         4           4            23.9
 8  75.4   1.72 V        52      0     0         1        0       0      0 N        2         4           4            25.5
 9  79.3   1.74 V        42      0     0         3        1       0      0 N        0         2           1            26.2
10  52.6   1.71 V        52     30     7         0        0       0      2 N        3         4           4            18.0
# i 4,990 more rows
# i Use `print(n = ...)` to see more rows
>
```

Puedo ver que la columna IMC se ha añadido correctamente.

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

Para eliminar las filas con datos no disponibles uso na.omit

```
#Ahora tratare de eliminar las filas en las que nos falte algun dato
datos <- na.omit(datos)
```

Resultado:

```
> datos
# A tibble: 4,964 × 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>    <fct>     <fct>       <fct>       <dbl>
 1  76.6   1.71 M        56      0     0         0       10       7      0 N        1         1           3            26.2
 2  46.7   1.61 M        41      0    15         0        0       7      0 N        2         2           2            18.0
 3  45.5   1.59 M        37    120     0         0        0       6      0 N        0         0           0            18.0
 4  62.2   1.8  V        66      0     9         1        0       3      0 N        1         3           4            19.2
 5  55.8   1.76 V        45     40     8         0        0       0      0 N        1         3           4            18.0
 6  86.1   1.83 V        43      0     0         1        0       2      0 N        2         1           1            25.7
 7  66.6   1.67 M        18      0     0         1        0       0      0 N        3         4           4            23.9
 8  75.4   1.72 V        52      0     0         1        0       0      0 N        2         4           4            25.5
 9  79.3   1.74 V        42      0     0         3        1       0      0 N        0         2           1            26.2
10  52.6   1.71 V        52     30     7         0        0       0      2 N        3         4           4            18.0
# ℹ 4,954 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

Puedo ver que la salida no tiene valores na.

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

Primero para quedarme con las variables numéricas aprovechare la función is.numeric con la función keep, luego aplico map con la función mean para la media, en el caso de la desviación típica aplico con el map una función sin nombre que calcula la desviación típica directamente con la corrección de Bessel aplicada

```
#Primero para quedarme con las variables numéricas aprovechare la función is.numeric con la función keep
##lo guardo en la variable datos numericos y con un map le aplico la funcion mean a todo el dataframe
datosNumericos <- keep(datos, is.numeric)
datosMedia<- map_dbl(datosNumericos, mean)

#Para calcular la desviacion tipica usare la funcion sd que que calcula la desviacion estandar con un map
#Debajo de la desviacion tipica tambien le he aplicado la correcion de bessel debido a que la desviacion estandar por si sola
#tiende a subestimar la desviacion estandar de la poblacion de la que se extrajo la muestra.

desvTipica <- map_dbl(datosNumericos, sd)
desvTipicasBessel <- map_dbl(datosNumericos, function(x) sqrt(sum((x-mean(x))^2) / (length(x)-1)))
```

Resultado:

```
> datosMedia
      peso     altura       edad     tabaco       ubes  carneRoja   verduras    deporte     drogas        IMC
64.0389988  1.7017002 40.6865431 20.5157131  3.9625302  1.7502015  5.8932313  4.2435536  0.4981869 22.0785265
> desvTipicasBessel
      peso     altura       edad     tabaco       ubes  carneRoja   verduras    deporte     drogas        IMC
12.0326622  0.0710788 14.2481410 42.3690924  5.7714729  2.0727378  6.9591858  4.6755404  1.4776857  3.7112451
>
```

Los resultados obtenidos me parecen bastante acordes por lo que puedo deducir que mi calculo está bien, en la media podemos apreciar valores que cuadran perfectamente y en la desviación apreciamos una dispersión coherente respecto a la media para cada variable.

5. Calcula los coeficientes de regresión y el coeficiente de determinación para las 12 regresiones lineales unidimensionales.

Para hallar los coeficientes he tenido que primero aislar las variables independientes, luego crear una función que me calculase cada coeficiente con la entrada del dataset, la variable dependiente y la variable independiente, y finalmente con un map he llamado a estas funciones creadas por mí.

```r
#Lo primero que hare sera extraer las variables independientes
VariablesInd <- names(datos[3:14])

#Una vez los tengo usare lm y summary para obtener los coeficientes de regresion que cuantifican la relacion entre la variable dependiente e independiente
coeficienteReg<-function(df, y, x) {
  modelo<- lm(y ~ x, df)
  summary(modelo)$coefficients[2]
}

#El siguiente paso sera calcular el coeficiente de determinación que nos indica cuanto depende la variable dependiente de la variable independiente
coeficienteDet <- function(df, y, x) {
  modelo<- lm(y ~ x, df)
  summary(modelo)$r.squared
}

#Ahora calculare el coeficiente de regresión entre la variable de respuesta IMC
#y cada una de las variables predictoras
coeficientesRegIMC <- map_dbl(VariablesInd,~ coeficienteReg(datos, datos$IMC, datos[[.x]]))

#Y hare lo mismo con el coeficiente de determinacion
coeficientesDetIMC <- map_dbl(VariablesInd,~ coeficienteDet(datos, datos$IMC, datos[[.x]]))
```

Resultados:

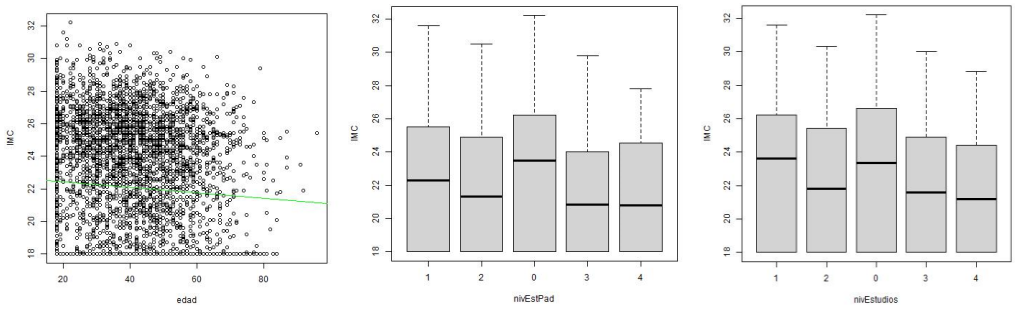
```
> coeficientesRegIMC
 [1] -0.07105066 -0.01644825 -0.04650589 -0.28278407  0.14986019  0.02062958  0.12289048 -0.65029904 -0.29632692 -0.53622308 -0.82965180  0.31207674
> coeficientesDetIMC
 [1] 9.164111e-05 3.987647e-03 2.818870e-01 1.933944e-01 7.005224e-03 1.496436e-03 2.396960e-02 6.704264e-02 3.038141e-04 1.879217e-02 2.188543e-02
[12] 1.895011e-02
>
```

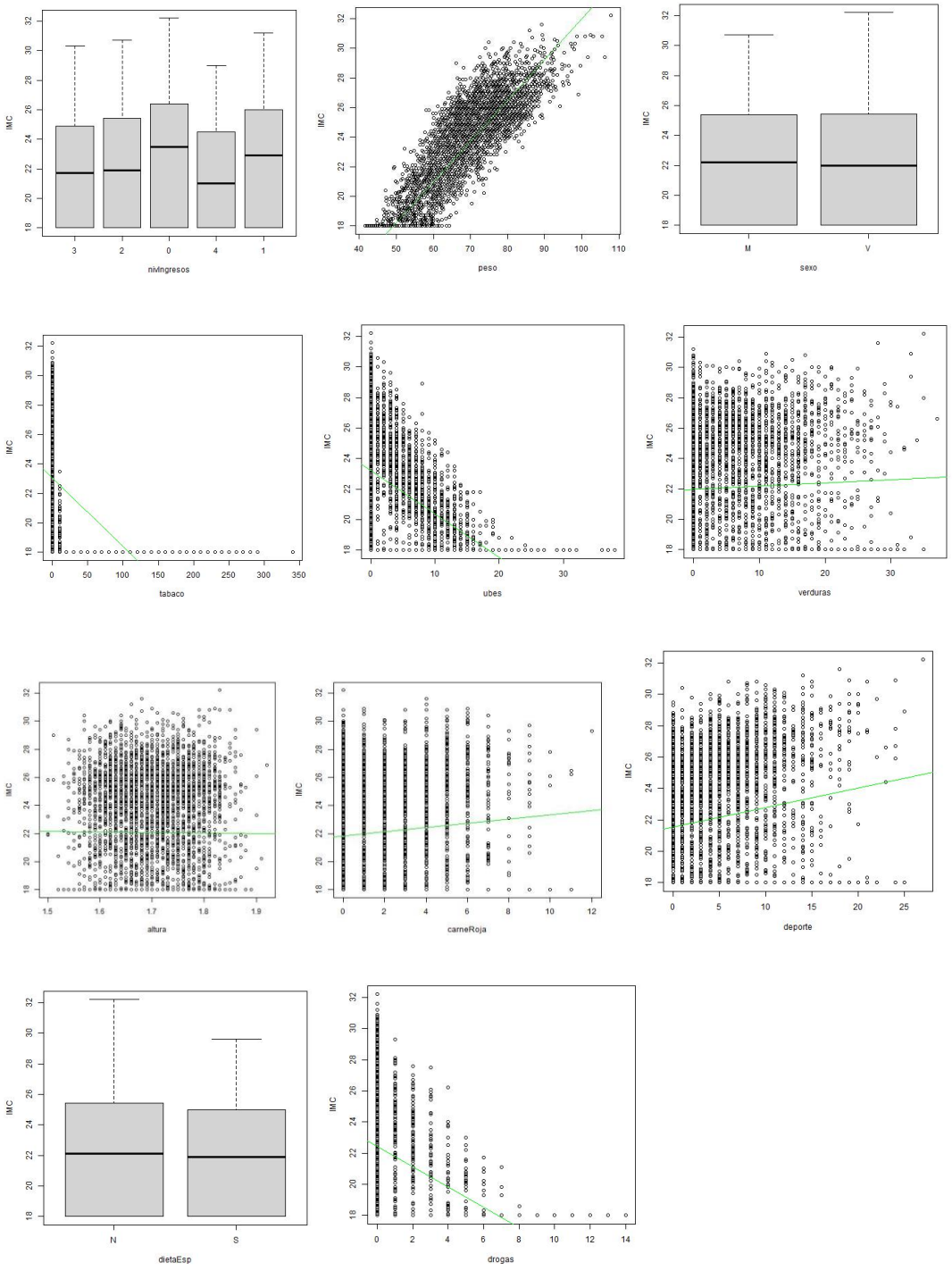
La salida aparenta ser correcta , hay valores negativos que indican un decrecimiento de la regresión y viceversa.

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.

Para llevar a cabo la tarea creo una función que hace el ajuste linear entre las variables, otra que dibuja los modelos, finalmente solo tengo que aplicarlas al dataset y R genera en la carpeta que le he especificado las gráficas abajo mostradas.

```r
# Para empezar creare una funcion que calcule el ajuste lineal entre la variable dependiente y la independiente
ajusteLinear <- function(df, y, x) {
  list(x = x, y = y, mod = lm(str_c(y, "~", x), df))
}

# Dibujaremos los modelos teniendo en cuenta que si son datos numericos usare un plot que unira los datos con una linea verde
# y en caso contrario un boxplot
dir.create("Graficos")
dibujarModelos <- function(mod) {
  jpeg(str_c("./Graficos/", mod$x, ".jpeg"))
  varx <- mod$x
  vary <- mod$y

  if (is.numeric(datos[[varx]])) {
    plot(datos[[varx]], datos[[vary]], xlab = varx, ylab = vary)
    abline(mod$mod, col = "green")
  } else {
    boxplot(formula = datos[[vary]] ~ datos[[varx]], xlab = varx, ylab = vary)
  }
  dev.off()
}

# ahora generare los modelos de regresion lineal ayudandome de las funciones que he creado
modelos <- map(names(datos), ~ ajusteLinear(datos, "IMC", .x))

# Ya solo me queda generar los graficos usando walk
walk(modelos, dibujarModelos)
```





Los resultados obtenidos representan la variable IMC en función del resto de variables, en estas imágenes ya podemos ir haciéndonos a la idea de que manera variables como las drogas, el tabaco, el deporte pueden afectar al IMC de una persona.

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

Para separar en tres conjuntos creo una función cuya entrada es el dataset, y los tamaños de los conjuntos.

```r
#Usare la forma que vemos en los apuntes para separar los sets creando una funcion que tiene como parametros los tamaños de los sets
separarSets <- function(df, p1, p2) {
  rDf <- 1:nrow(df)
  rTrain <- sample(rDf, p1 * length(rDf))
  rResto  <- setdiff(rDf, rTrain)
  rTest <- sample(rResto, p2*length(rDf))
  rValido <- setdiff(rResto, rTest)

  list(train=df[rTrain,], test=df[rTest,], valido=df[rValido,])
}

setsSeparados <- separarSets(datos,.6,.2)
```

Resultado:

```
> setsSeparados
$train
# A tibble: 2,978 × 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>    <fct>     <fct>       <fct>       <dbl>
 1  80.7   1.8  V        44      0     4         0        0       5      0 N        1         2           3            24.9
 2  50.8   1.68 M        43    120     4         0        4       0      0 N        2         1           0            18.0
 3  72.2   1.65 V        46      0     0         2        9       0      0 N        0         1           0            26.5
 4  69.1   1.78 V        71      0     6         0        5       0      0 N        1         1           1            21.8
 5  47.8   1.63 M        31     90     4         2        0       6      0 N        3         4           4            18.0
 6  66.3   1.69 V        52      0     0         0        9       1      0 N        1         3           4            23.2
 7  71.3   1.72 V        54      0     0         0        0       0      0 N        2         3           2            24.1
 8  67.5   1.74 V        38      0     0         1        0       1      0 N        2         4           4            22.3
 9  79.9   1.76 V        38      0     0         0       11       4      0 N        0         1           0            25.8
10  74.3   1.71 M        49      0     0         0        0       1      0 N        2         3           3            25.4
# i 2,968 more rows
# i Use `print(n = ...)` to see more rows

$test
# 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>    <fct>     <fct>       <fct>       <dbl>
 1  70.6   1.69 M        56      0     0         0       32      10      0 N        2         2           3            24.7
 2  44.9   1.58 M        59     30     0         2       12       7      0 N        0         2           3            18.0
 3  56.1   1.7  M        48      0     3         1        3       0      0 N        2         3           4            19.4
 4  67.4   1.8  V        34      0     0         2        0      10      5 N        2         2           4            20.8
 5  47.8   1.63 M        28      0    29         2        1       8      0 N        1         0           0            18.0
 6  73.7   1.79 V        36      0     8         0        4       8      0 N        1         2           3            23.0
 7  87.4   1.76 V        28      0     0         2        0       2      0 N        0         2           1            28.2
 8  73     1.77 V        57      0     0         0        9       4      0 N        2         4           3            23.3
 9  70.2   1.65 V        37      0     0         4        9       3      0 N        2         2           3            25.8
10  67.5   1.64 M        53      0     0         3        0       4      0 N        2         3           3            25.1
# i 982 more rows
# i Use `print(n = ...)` to see more rows

$valido
# A tibble: 994 × 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>    <fct>     <fct>       <fct>       <dbl>
 1  70.7   1.64 M        34      0     0         0       14      10      0 N        1         2           2            26.3
 2  80.8   1.76 V        32      0     0         0        0       3      0 N        2         2           1            26.1
 3  60.7   1.54 M        39      0     0         0        0       0      0 N        0         1           0            25.6
 4  63.9   1.66 M        54      0     0         0        0       0      0 S        2         3           3            23.2
 5  80.7   1.79 V        33      0     0         1        0       0      0 N        1         3           4            25.2
 6  59.5   1.76 V        46     10     0         0       12       5      0 N        1         2           2            19.2
 7  73.8   1.75 V        62      0     0         2        0       0      0 N        0         2           2            24.1
 8  79.0   1.77 M        41      0     2         0       16      15      0 N        2         4           4            25.2
 9  79.3   1.64 M        46      0     0         5        0       5      0 N        1         1           2            29.5
10  86.9   1.74 M        52      0     0         7        0       0      0 N        1         0           0            28.7
# i 984 more rows
# i Use `print(n = ...)` to see more rows
```

Se aprecian que los sets se han creado con éxito.

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.

Tengo que buscar cual de las 12 variables es un mejor reflejo del IMC. Para ello, defino una función que calcula el coeficiente de determinación ajustado R2 para un modelo de regresión lineal y otra función que evalúa el modelo de regresión lineal con el conjunto de entrenamiento y prueba. Luego, aplico la función a cada variable predictora y selecciono la variable que genera el coeficiente de determinación ajustado R2 más alto.

```r
#Para seleccionar la variable que mejor explica el IMC necesitaré ayudarme de unas cuantas funciones
# Esta función calcula el coeficiente de determinación ajustado R2 para un modelo de regresión lineal.
calcR2 <- function(df, mod, y) {
  MSE <- mean((df[[y]] - predict.lm(mod, df)) ^ 2)
  varY <- mean(df[[y]] ^ 2) - mean(df[[y]]) ^ 2
  R2 <- 1 - MSE / varY
  ajR2 <- 1 - (1- R2) * (nrow(df) - 1) / (nrow(df) - mod$rank)
  ajR2
}
#Con esta funcion evaluare el modelo de regresión lineal utilizando el conjunto de datos de entrenamiento y el otro de prueba.
calcModR2 <- function(dfTrain, dfTest, y, x) {
  mod <- ajusteLinear(dfTrain, y, x)
  calcR2(dfTest, mod$mod, y)
}
#Pasaré a utilizar la función map_dbl para aplicar la función calcModR2 que acabo de crear a cada variable predictora del vector VariablesInd.
AjusteR2 <- VariablesInd %>%
  map_dbl(calcModR2,dfTrain=setsSeparados$train,dfTest=setsSeparados$test,y="IMC")

#Ahora calcularé la variable predictora que genera el coeficiente de determinación ajustado R2 más alto.
x <- which.max(AjusteR2)
mejorVar <- VariablesInd[x]
```

Resultado:

```
> mejorVar
[1] "tabaco"
```

El resultado recibido es que la variable tabaco es la que mejor explica el IMC de una persona, esto quiere decir que es la que mas probabilidades tiene de predecir la realidad de la variable independiente.

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.

Para encontrar un modelo optimo lineal de regresión uso la función “encontrarMejorAjuste” para encontrar la combinación de variables predictoras que mejor explican el IMC en el conjunto de datos de entrenamiento y devuelve un modelo ajustado utilizando esas variables.

```r
encontrarMejorAjuste <- function(dfTrain, dfTest, varPos) {
  bestVars <- character(0)
  aR2      <- 0

  repeat {
    aR2v <- map_dbl(varPos, ~calcModR2(dfTrain, dfTest, "IMC", c(bestVars, .)))
    i    <- which.max(aR2v)
    aR2M <- aR2v[i]
    if (aR2M <= aR2) break

    cat(sprintf("%1.4f %s\n", aR2M, varPos[i]))
    aR2 <- aR2M
    bestVars <- c(bestVars, varPos[i])
    varPos   <- varPos[-i]
  }

  mod <- ajusteLinear(dfTrain, "IMC", bestVars)

  list(vars=bestVars, mod=mod)
}
# Llamo a la función encontrarMejorAjuste para obtener el modelo que mejor explica el IMC
# mejorAjuste es una lista que contiene las variables predictoras seleccionadas y el modelo ajustado
mejorAjuste <- encontrarMejorAjuste(setsSeparados$train, setsSeparados$test, VariablesInd)
```

Resultado:

```
> # Llamo a la función encontrarMejorAjuste para obtener el modelo que mejor explica el IMC
> # mejorAjuste es una lista que contiene las variables predictoras seleccionadas y el modelo ajustado
> mejorAjuste <- encontrarMejorAjuste(setsSeparados$train, setsSeparados$test, VariablesInd)
0.2950 tabaco
There were 11 warnings (use warnings() to see them)
>
```

El mejor ajuste es 0.2950 en tabaco

10.Evalúa el resultado en el conjunto de validación.

Para evaluarlo se extrae el modelo ajustado de la lista devuelta y se utiliza para calcular el coeficiente de determinación ajustado R2 en el conjunto de datos de validación utilizando la función "calcR2". Finalmente, el valor del coeficiente de determinación ajustado R2 se utiliza para evaluar la calidad del modelo ajustado.

```
# Se extrae el modelo ajustado de la lista mejorAjuste
mejorMod <- mejorAjuste$mod$mod
# Calculo el valor de R2 ajustado para el conjunto de validación utilizando el modelo ajustado
calcR2(setsSeparados$valid, mejorMod, "IMC")
```

Resultado:

```
> #EJERCICIO 10
> # Se extrae el modelo ajustado de la lista mejorAjuste
> mejorMod <- mejorAjuste$mod$mod
> # Calculo el valor de R2 ajustado para el conjunto de validación utilizando el modelo ajustado
> calcR2(setsSeparados$valid, mejorMod, "IMC")
[1] 0.2571852
>
```

Como el dato es coherente respecto al del apartado A comprobamos que el resultado es válido.

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.

Para llevar a cabo esta tarea solo tengo que leer el dataframe, usar predict.lm con el mejor modelo sacado en el ejercicio anterior y finalmente usar la formula del IMC para con la altura y esta columna deducida sacar el valor de peso.

```
#Empezare importando el archivo que contiene el dataframe eval
dfEval <- read_csv("eval.csv")
#Ahora tendre que deducir el IMC
dfEval["IMC"] <- predict.lm(mejorMod, dfEval)
#Finalmente dedicire la variable de peso con la de IMC y altura
dfEval["Peso"] <- dfEval$IMC*dfEval$altura^2

#Por ultimo ya solo tengo que guardar este archivo
write.csv(dfEval, "evalX.csv", row.names = FALSE)
```

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

• Que utilidad podría tener el modelo matemático que has obtenido.

Este modelo podría ser muy útil para saber la predisposición a la obesidad o a la anorexia de una persona según sus hábitos y estilo de vida.

• Que se puede deducir a partir del modelo sobre la relación entre las variables.

Se puede deducir que existe una relación entre los hábitos y las características de una persona y su IMC.

• Problemas que has encontrado en el desarrollo.

Los problemas que he encontrado ha sido al manejar vectores y dataframes en R ya que cambia la manera de aplicar algunas funciones y eso me ha producido muchos errores que me ha costado solucionar.

• Qué te ha llamado la atención en el proceso.

Me llama mucho la atención el poder predecir desde una base algo que a primera vista parece imposible de saber, pero que con un análisis de los datos se descubren relaciones entre variables que tienen mucho poder y como esto puede ser aplicado a ámbitos desde marketing, pasando por afluencia de personas o estimaciones de ventas, hasta a la salud.

• Qué más podría hacerse y cómo plantearlo.

Se me ocurre que se podría tener una lista con las variables que mas influyen y tratar de controlar a las personas que reúnen estas debido a que sufren una probabilidad mucho mas alta de tener problemas de salud relacionados con su peso salvando así incluso vidas. Fuera del ámbito de salud hay muchas otras relaciones que también podrían ser interesantes como la relación entre las personas que consumen drogas y su nivel de estudios, la relación entre la gente que hace deporte y el nivel de ingresos o el nivel de ingresos y los estudios de los padres.