Módulo 4 - Aprendizaje Supervisado: Técnicas de Regresión.

4.2 Métodos clásicos de regresión.

- ### La regresión Lineal Simple y Multiple

Autores:

Por Rafael Alcalá

Catedrático de la Universidad de Granada

Instituto Andaluz Interuniversitario en Data Science and Computational Intelligence (DaSCI)

Y Augusto Anguita-Ruiz

Investigador postdoctoral en Instituto de Salud Global de Barcelona- ISGlobal.

Recordatorio: Introducción a NoteBook

Dentro de este cuaderno (*NoteBook*), se le guiará paso a paso desde la carga de un conjunto de datos hasta el análisis descriptivo de su contenido.

El cuaderno de *Jupyter* (Python) es un enfoque que combina bloques de texto (como éste) junto con bloques o celdas de código. La gran ventaja de este tipo de celdas, es su interactividad, ya que pueden ser ejecutadas para comprobar los resultados directamente sobre las mismas. *Muy importante*: el orden las instrucciones es fundamental, por lo que cada celda de este cuaderno debe ser ejecutada secuencialmente. En caso de omitir alguna, puede que el programa lance un error, así que se deberá comenzar desde el principio en caso de duda.

Antes de nada:

Es muy muy importante que al comienzo se seleccione "*Abrir en modo de ensayo*" (draft mode), arriba a la izquierda. En caso contrario, no permitirá ejecutar ningún bloque de código, por cuestiones de seguridad. Cuando se ejecute el primero de los bloques, aparecerá el siguiente mensaje: "*Advertencia: Este cuaderno no lo ha creado Google.*". No se preocupe, deberá confiar en el contenido del cuaderno (*NoteBook*) y pulsar en "Ejecutar de todos modos".

¡Ánimo!

Haga clic en el botón "play" en la parte izquierda de cada celda de código. Las líneas que comienzan con un hashtag (#) son comentarios y no afectan a la ejecución del programa.

También puede pinchar sobre cada celda y hacer "*ctrl+enter*" (*cmd+enter* en Mac).

Cada vez que ejecute un bloque, verá la salida justo debajo del mismo. La información suele ser siempre la relativa a la última instrucción, junto con todos los `print()` (orden para imprimir) que haya en el código.

ÍNDICE

En este *notebook*:

1. Aprenderemos los conceptos generales de la técnica de regresión Lineal Simple y Multiple.
2. Aplicaremos la regresión Lineal Multiple como primera herramienta de estudio sobre cualquier problema de regresión real (con ejemplos sobre nuestro problema de estimación de la insulino-resistencia).

Contenidos:

1. REGRESIÓN LINEAL

El primer **método de regresión** que abordaremos en este módulo es la **regresión lineal**, considerada como un enfoque simple dentro del aprendizaje supervisado. En la **regresión lineal**, se asume que la **dependencia de la variable de salida** Y (en nuestro caso de estudio sobre obesidad infantil representada por la variable *HOMA-IR*) **sobre las variables de entrada** $X_1, X_2, \ldots, X_p$ (resto de variables de nuestro problema) **es lineal**. Aunque pueda parecer demasiado simplista, la **regresión lineal** es extremadamente útil tanto conceptualmente como en la práctica. En esta cápsula veremos cómo hacer uso de la misma para estudiar un conjunto de datos y poder sacar conclusiones últiles sobre el comportamiento de los mismos, sin perjuicio de que luego se utilicen otras técnicas (supuestamente o en teoría más potentes) para obtener modelos mejor ajustados. En un primer lugar, haremos una distinción entre **regresión lineal simple o múltiple**.

1.1 Conceptos básicos sobre la regresión lineal simple

En una **regresión lineal simple**, usando **una sola variable de entrada** X asumimos el siguiente modelo,

$$Y = \beta_0 + \beta_1 X + \epsilon,$$

donde β_0 y β_1 son dos constantes desconocidas que representan el término independiente y la pendiente de una función lineal (una recta) respectivamente. A estas constantes se las conoce como **coeficientes o parámetros**. Por otro lado, ϵ hace referencia al **término de error de la estimación**.

Dada una estimación $\hat{\beta}_0$ y $\hat{\beta}_1$ para los **coeficientes** del modelo, podemos predecir valores futuros de la variable de salida Y utilizando,

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x,$$

donde $\hat{y}$ representa una predicción de Y sobre la base de $X = x$. El símbolo del sombrero denota que nos estamos refiriendo a **valores estimados** (en lugar de a valores reales u observados). Los valores de $\hat{\beta}_0$ y $\hat{\beta}_1$ se obtienen por la **técnica matemática de mínimos cuadrados** que obtiene los coeficientes que minimizan el error cometido para cada instancia disponible en un conjunto de datos de entrenamiento. Dicho de otra forma, por **mínimos cuadrados** se obtienen siempre los valores óptimos de los **coeficientes** que minimizan el valor de **RECM**.

1.2 Conceptos básicos sobre la regresión lineal múltiple

En el caso de la **regresión lineal múltiple**, usando **más de una variable de entrada** X asumimos el siguiente modelo,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon,$$

interpretando cada coeficiente β_j como el efecto promedio en Y de una unidad de incremento en X_j, manteniendo el resto de variables de entrada fijas. El caso ideal es cuando las variables de entrada $X's$ no están correlados entre sí (no existe colinealidad), ya que esto implicaría ciertos problemas de interpretación. No obstante, si el proceso de aprendizaje se realiza paso a paso, atendiendo a los valores estadísticos obtenidos sobre dichos coeficientes, las variables que están correladas entre sí terminan por ser eliminadas. En la **regresión lineal múltiple**, la estimación de los coeficientes también se realiza mediante la **técnica de mínimos cuadrados**, minimizando el error de las predicciones sobre el conjunto de datos de entrenamiento.

1.3 Evaluación de la bondad de la estimación de los coeficientes

Esta técnica genera ciertos **valores estadísticos** que nos permitirán contestar a preguntas del tipo; *¿Existe al menos una variable de entrada (X_j) que tenga relación lineal con la variable de salida (Y)?*, *En caso afirmativo, ¿Cuál/es es/son?*. Los **valores estadísticos** de los que hablamos son dos:

- **Estadístico F**: El estadístico F es un valor que se obtiene para el modelo de regresión de manera completa, en lugar de para cada una de las variables de entrada. En la medida en la que su valor se aleje de 1 (tanto hacia arriba como hacia abajo) indica que al menos hay una variable de entrada que presenta relación lineal con la variable de salida. Si éste valor es cercano a 1 podemos directamente descartar la regresión lineal modelada, ya que no habrá ninguna variable de entrada (X_j) con relación lineal con la salida (Y). Este parámetro resulta de especial importancia en problemas con un elevado número de variables de entrada, entre las que puede haber variables que en un principio se muestren como relevantes sin realmente serlo. Por este motivo, el **estadístico F** es el primer parámetro que debemos de estudiar al realizar una regresión. Tras ello, **solo se continuará con el proceso** si su valor no está cercano a 1.
- **P-valor de los estadísticos t**: El p-valor es un parámetro que está asociado a cada uno de los coeficientes (un p-valor para cada coeficiente), aunque tambien se obtiene un p-valor global para el modelo de regresión completo. Los p-valores indicarán si el coeficiente β_j de cada variable de entrada X_j es relevante, o si por el contrario hay una alta probabilidad de que realmente pudiese ser cero ($\beta_j \approx 0$ implica que es una variable sin relación lineal con la variable de salida, y por lo tanto a eliminar del modelo). Esto pasa cuando tenemos un p-valor para dicho coeficiente por encima de $0,1$ o $0,15$, que indicaría que la variable podría no ser relevante ($\beta_j \approx 0$).

1.4 Evaluación de la bondad del modelo

Como ya se mencionó en la cápsula anterior, el valor del **Coeficiente de Determinación** R^2 es un valor que puede estar entre 0 y 1, indicando el valor 1 un ajuste perfecto del modelo (error cero), y el 0 un ajuste con el peor error posible. Para **comparar varios modelos de regresión lineal entre sí**, se suele atender al valor de R^2 obtenido en cada uno de ellos. No obstante, como se debe de tener en cuenta el número de coeficientes β_j de los distintos modelos, para poder comparar modelos con distinto número de variables de entrada, en realidad, tendríamos que mirar el conocido como R^2 **ajustado** (que ya los tiene en cuenta durante su cálculo).

Por otra parte, si queremos comparar un **modelo de regresión lineal** con modelos obtenidos por **otras técnicas de regresión** (Knn, Redes neuronales, M5, ...) **se utilizará el valor de RECM**. Recordemos que, por el contrario, R^2 es un parámetro relativo y que además no todas las técnicas permiten su cálculo.

1.5 Elección de las variables de entrada relevantes

La utilidad de la regresión lineal depende mucho de las variables que se escojan para aprender sus **coeficientes**. En principio, no hay otra forma que seleccionarlas de manera manual, ya que no se pueden examinar todas las posibles combinaciones de variables. A modo de ejemplo, para un problema **con 40 variables de entrada**, $p = 40$, tendríamos 2^p combinaciones de modelos distintos **(por encima de un billón de combinaciones)**. **Para decidir qué variables deberían ser consideradas en el modelo final** existen **dos enfoques** comúnmente utilizados:

1. **Selección ascendente**: Este procedimiento consiste en comenzar construyendo el **modelo nulo** - un modelo que contiene el término independiente pero ninguna variable de entrada - y realizar en paralelo regresiones lineales simples entre la variable de salida/dependiente Y y cada uno de las variables de entrada X_j. A continuación, se agregará al modelo nulo la variable de entrada que resulte en **el error más bajo entre todos los modelos testeados (aquel con el R^2 ajustado más alto)**. Continuaremos con este procedimiento hasta que se cumpla alguna regla de detención, por ejemplo, cuando al añadir cualquiera de las variables restantes **se obtenga un p-valor por encima de algún umbral determinado para dicha variable ($> 0,1$ o $0,15$ por ejemplo)**. **¡¡¡IMPORTANTE!!!:** *Las variables deben de ser añadidas de una en una y NUNCA varias de golpe. Cada vez que se añade una variable, ésta contribuye a explicar una parte de los datos, por lo que los p-valores de las demás podrían cambiar y dejar de ser relevantes al ya no haber necesidad de explicar dicha parte de los datos*.
2. **Selección descendente**: Empezar con todas las variables en el modelo. Eliminar la variable de entrada con el p-valor más alto, es decir, la variable estadísticamente menos significativa y ajustar el nuevo modelo con las $(p - 1)$ variables restantes. En la siguiente vuelta, de nuevo, la variable con el mayor p-valor se elimina, continuando con el proceso hasta que se alcance una regla de parada. Por ejemplo, podemos detenernos cuando todas las variables de entrada que componen el modelo tengan un valor significativo (por ejemplo p-valor por debajo de 0.1 o 0.15). **¡¡¡IMPORTANTE!!!:** *Las variables deben de ser eliminadas de una en una y NUNCA varias de golpe. Cada vez que se elimina una variable estamos facilitando que alguna de las que siguen quedando pueda explicar de manera adecuada una parte de los datos, pasando de ser una variable poco relevante o incluso molesta (con un p-valor muy alto) a una variable imprescindible. Por lo tanto, perderíamos esas variables si quitamos varias de golpe. Si ésto se sigue a rajatabla, se solucionará buena parte de los problemas de correlación (colinealidad) entre variables de entrada.*

Es importante considerar que, en cualquiera de estos dos enfoques, el **término independiente** de la regresión no entra en juego, y por lo tanto nunca se eliminará del modelo.

Existen otras posibilidades además de estos dos enfoques, como por ejemplo calcular las correlaciones que todas las variables de entrada tienen con la variable de salida, y quedarse con un conjunto pequeño de las mejores; o por ejemplo, aplicar alguno de los algoritmos de selección de variables existentes. **Por desgracia, ninguna de ellas asegura que se llegue a la mejor combinación existente**.

En este curso, **recomendamos utilizar la selección descendente cuando el número de variables no sea excesivamente elevado**, y **la ascendente cuando si lo sea** (automatizando la elección de la variable que debe incluirse en cada paso).

El uso de cualquiera de estos dos enfoques nos permitirá justamente lo que vamos buscando con la regresión: **Estudiar un conjunto de datos y poder sacar conclusiones últiles** sobre el comportamiento de los mismos, incluso aunque finalmente se apliquen otro tipo de técnicas más versátiles. En nuestro **conjunto de datos sobre la obesidad e insulino-resistencia en niños**, aplicaremos **el enfoque descendente**.

1.6 Extensiones del modelo lineal. Eliminando la suposición aditiva: interacciones y no linealidad.

Para atender a la **no linealidad** de los datos existente en la mayoría de conjuntos de datos de la vida real, se pueden considerar nuevos **términos de interacción** (variables con sinergia positiva que se potencian la una a la otra) o de **no linealidad** (variables con crecimiento cuadrático, logarítmico, etc.). Para poder estudiar estos comportamientos en una **regresión lineal** haremos uso de dos figuras:

- **Interacciones** (términos que no presentan un comportamiento aditivo entre sí): Estos términos de interacción se presentan en los modelos de regresión como $X_1 * X_2$, y representan como el cambio en dos o más variables de manera conjunta provoca cambios en la variable de salida Y mayores de lo que lo harían por separado. Por ejemplo: es conocido que la inversión de 5000 euros de publicidad en $radio$ y otros 5000 en $televisión$ provoca ventas mucho mayores de un producto que si se invierte directamente 10000 euros en cualquiera de los dos medios de manera unilateral. Incluir en el modelo de regresión un nuevo término multiplicativo del tipo $radio * televisión$ nos permitiría explicar adecuadamente dicho tipo de fenómeno no lineal.
- **Otros términos no lineales**: Muchas veces la relación entre una variable de entrada y la variable de salida no es lineal sino cuadrática, cúbica, logarítmica, exponencial, etc. Incluir un término que coincida con dicho tipo de relación puede ayudar a explicar adecuadamente estos fenómenos no lineales.

No obstante, en todos los casos se debe de respetar el **principio de jerarquía**. Es decir, si se incluye una variable nueva X_5^3 a partir de X_5 y su p-valor indica que dicho término cúbico es relevante, entonces se deben incluir también X_5^2 y la propia X_5, incluso si sus p-valores son altos. **En caso contrario, estaremos cometiendo errores graves**. Es igual en el caso de las interacciones. Si se incluye y se mantiene una variable nueva $(X_1 * X_2 * X_6)$ porque las tres se complementan y, por lo tanto, el p-valor para dicho término indica que es relevante, entonces se deben incluir también (independientemente de sus p-valores): $(X_1 * X_2)$, $(X_1 * X_6)$, $(X_2 * X_6)$, X_1, X_2, y X_6.

2. INSTALACIÓN DE R, BIBLIOTECAS Y LECTURA DE LOS DATOS DE OBESIDAD INFANTIL

Como se explicó en la cápsula anterior, necesitamos ejecutar las siguientes 3 celdas antes de empezar con el algoritmo de regresión lineal.

In [1]:

```
# Tiempo estimado de ejecución: 20 segundos aprox.

### Instalación de R en notebooks de Google Colab ###
!apt-get update
!apt-get install r-base
!pip install rpy2==3.5.1
%load_ext rpy2.ipython
print ("Instalación de R en Google Colab terminada")
```

```
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Ign:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_6
4  InRelease
Get:3 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease [3,
626 B]
Hit:4 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:5 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease [1
5.9 kB]
Ign:6 https://developer.download.nvidia.com/compute/machine-learning/repos/ubun
tu1804/x86_64  InRelease
Get:7 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_6
4  Release [697 B]
Get:8 https://developer.download.nvidia.com/compute/machine-learning/repos/ubun
tu1804/x86_64  Release [564 B]
Get:9 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_6
4  Release.gpg [836 B]
Get:10 https://developer.download.nvidia.com/compute/machine-learning/repos/ubu
ntu1804/x86_64  Release.gpg [833 B]
Get:11 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Hit:12 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Get:13 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packa
ges [24.5 kB]
Get:14 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Package
s [1,396 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:16 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease [15.9 k
B]
Get:17 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages
[1,963 kB]
Get:18 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packa
ges [324 kB]
Get:19 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease [2
1.3 kB]
Ign:20 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_
64  Packages
Get:20 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_
64  Packages [577 kB]
Get:21 https://developer.download.nvidia.com/compute/machine-learning/repos/ubu
ntu1804/x86_64  Packages [73.8 kB]
Get:22 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main Sources
[1,745 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2,3
94 kB]
Get:24 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main amd64 P
ackages [893 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages
[2,163 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Package
s [31.4 kB]
Get:27 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Package
s [353 kB]
Get:28 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic/main amd64 Package
s [39.5 kB]
Get:29 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic/main amd64 P
ackages [49.4 kB]
Fetched 12.3 MB in 4s (3,142 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
r-base is already the newest version (4.0.4-1.1804.0).
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
Requirement already satisfied: rpy2 in /usr/local/lib/python3.7/dist-packages
(3.4.2)
```

```
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages
(from rpy2) (2018.9)
Requirement already satisfied: cffi>=1.10.0 in /usr/local/lib/python3.7/dist-pa
ckages (from rpy2) (1.14.5)
Requirement already satisfied: tzlocal in /usr/local/lib/python3.7/dist-package
s (from rpy2) (1.5.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages
(from rpy2) (2.11.3)
Requirement already satisfied: pycparser in /usr/local/lib/python3.7/dist-packa
ges (from cffi>=1.10.0->rpy2) (2.20)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dis
t-packages (from jinja2->rpy2) (1.1.1)
Instalación de R en Google Colab terminada
```

In [2]:
```
# Tiempo estimado de ejecución: 4 segundos aprox (al no necesitar importar kknn
 y Cubist).
# Bibliotecas necesarias:
# ISLR para regresión lineal multivariable
# kknn para k-vecinos más cercanos de regresión
# Cubist para modelos de regresión basados en M5

%%R
### Instalación de las bibliotecas necesarias
#install.packages(c("ISLR", "kknn", "Cubist"))
install.packages(c("ISLR")) #kknn y Cubist se utilizarán en la siguiente cápsula
print ("Instalación de las bibliotecas de R para este módulo terminada")

### Importación de las bibliotecas necesarias ###
require(ISLR)
#require(kknn)
#require(Cubist)
print ("Importación de las bibliotecas de R para este módulo terminada")
```

```
R[write to console]: Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

R[write to console]: trying URL 'https://cran.rstudio.com/src/contrib/ISLR_1.2.
tar.gz'

R[write to console]: Content type 'application/x-gzip'
R[write to console]:  length 1757568 bytes (1.7 MB)

R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: 

R[write to console]: downloaded 1.7 MB
```

```
R[write to console]:

R[write to console]:
R[write to console]: The downloaded source packages are in
        '/tmp/RtmpOz0gic/downloaded_packages'
R[write to console]:
R[write to console]:


[1] "Instalación de las bibliotecas de R para este módulo terminada"

R[write to console]: Loading required package: ISLR


[1] "Importación de las bibliotecas de R para este módulo terminada"
```

In [3]:
```
# Tiempo estimado de ejecución: 2 segundos aprox.

%%R
### Lectura
data <- read.csv(url("https://drive.google.com/uc?id=1GO2NBxYw54K6HkN-YgXbNadrLo
5O6-0u"))

### Visualización de una pequeña parte de los datos
head(data)
```

```
  Sex  Age Tanner Height   BMI   WC TAGmgDL HDLCmgDL LDLCmgDL SBP DBP Sedentary
1   1  9.5      0   1.55 11.34 60.0      55       51       93  97  60  411.0893
2   1  8.0      0   1.15 12.40 46.3      51       70       59  90  55  435.6071
3   0 10.5      0   1.42 12.99 67.5      65       60       96  96  54  483.9048
4   0  8.1      0   1.27 13.43 53.1      41       78      100 108  46  429.2976
5   1 10.4      0   1.32 13.72 51.9      39      100      120 107  69  512.0714
6   0 10.4      0   1.29 14.02 54.9      57       76       73  87  59  451.2321
     Light Moderate  Vigorous HOMA
1 321.5804 22.13393  3.982143 1.98
2 316.9762 48.05952 14.273810 0.87
3 337.7857 33.30952  7.988095 1.46
4 241.9762 39.67857 11.821429 1.07
5 216.0357  9.75000  2.410714 0.80
6 257.6429 36.40179  9.767857 1.35
```

Como podemos observar, el comando *head* nos ofrece una visualización de los datos disponibles para los primeros 6 individuos/instancias del conjunto. En esta visualización, podemos identificar variables como el *sexo* de los individuos (el cual viene codificado como 0 si el individuo es un varón, o 1 si es una niña), el *estadío puberal* (representado por la variable "Tanner", y codificado como 0 para el estadío pre-puberal y 1 para el estadío puberal), o la *presión sanguínea* (representada por las variables "DBP" para la tensión diastólica, y "SBP" para la tensión sistólica). Como se puede observar, también se cuenta con variables de sedentarismo, y actividad física ligera, moderada y vigorosa. El resto de variables abreviadas se refieren a: *BMI*, Body Mass Index (o Índice de Masa Corporal traducido al Español); *WC*, Waist Circumference (o Circunferencia de cintura traducido al Español); *TAG*, tliglicéridos; *HDL*, high-density lipoprotein (Colesterol "Bueno"); *LDL*, Low-density lipoprotein (Colesterol "Malo"); estos tres últimos expresados en miligramos/decilitro en sangre.

3. PRIMERA TOMA DE CONTACTO CON EL PROBLEMA Y ESTUDIO DE LAS VARIABLES DE MAYOR INTERÉS

Una vez importadas las librerías y leídos los datos estamos en disposición de ver **qué variables son las más prometedoras** para **aplicar la regresión lineal** en este **problema**. Para estudiar qué variables explican mejor el comportamiento de la variable de salida *HOMA-IR*, se podrían **cálcular las correlaciones** existentes entre ésta y cada una de las variables de entrada (lo cual se lleva a cabo mediante el comando: *cor(data)*). De esta manera, se podrían escoger aquellas que se encuentren más correladas.

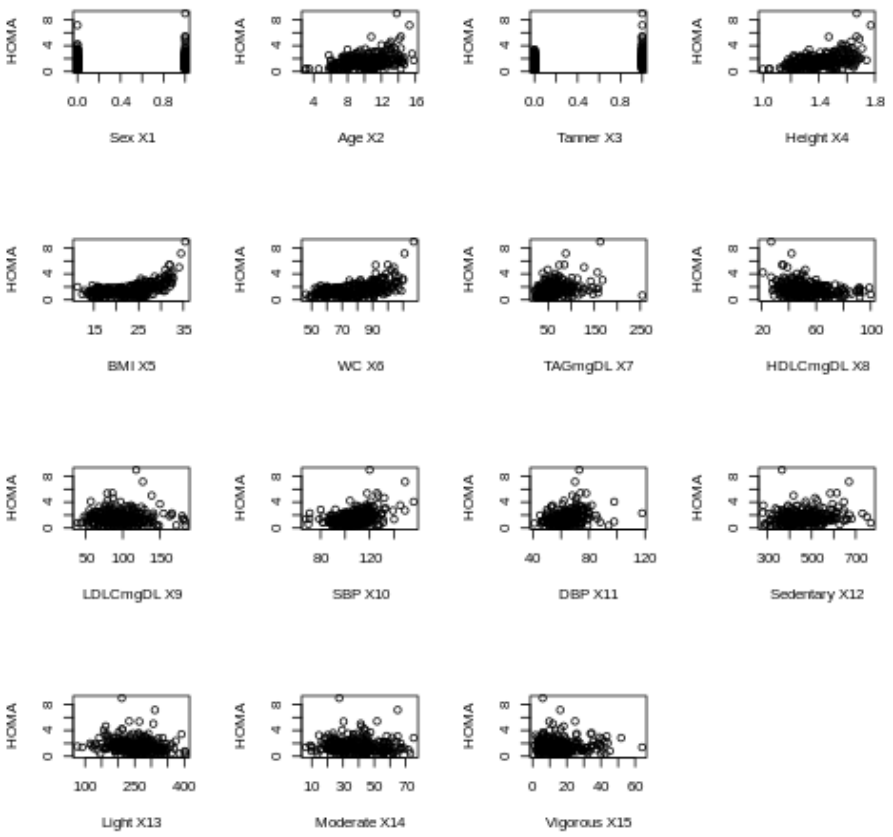
Si llevamos a cabo este procedimiento sobre nuestro conjunto de datos, observaremos cómo SBP tiene una correlación más alta con el *HOMA-IR* que otras variables de entrada (por ejemplo Sex). Sin embargo, al final de nuestro estudio veremos que SBP no terminará formando parte del modelo final, al contrario que Sex que si entrará como variable seleccionada. La explicación para esto es que la construcción del modelo final no sólo depende de la correlación individual de cara variable de entrada con la variable de salida, sino más bien de lo que aporte cada variable con respecto al resto de variables seleccionadas. Si lo que puede explicar una variable de entrada ya está mejor explicado por otra, ésta no debe formar parte del modelo final.

Una alternativa a lo anterior es **mostrar gráficamente la relación de cada variable de entrada con respecto a la variable de salida** *HOMA-IR*. De esta manera, podemos observar visualmente no sólo si presentan una relación más o menos lineal, sino qué forma tiene la nube de puntos. Por ejemplo, podríamos ver si una variable presenta un comportamiento cuadrático o logarítmico, y por lo tanto sería más apropiado incluir estos términos en el modelo.

En nuestro caso de estudio, optaremos por esta **segunda aproximación**, por considerarla tanto o más informativa. El siguiente bloque de código *R* grafica iterativamente, y por orden, todas las variables de entrada con respecto a la variable de salida (*HOMA-IR*).

NOTA: *A partir de aquí es importante que también lea los comentarios incluidos junto con el código para una mejor comprensión*.

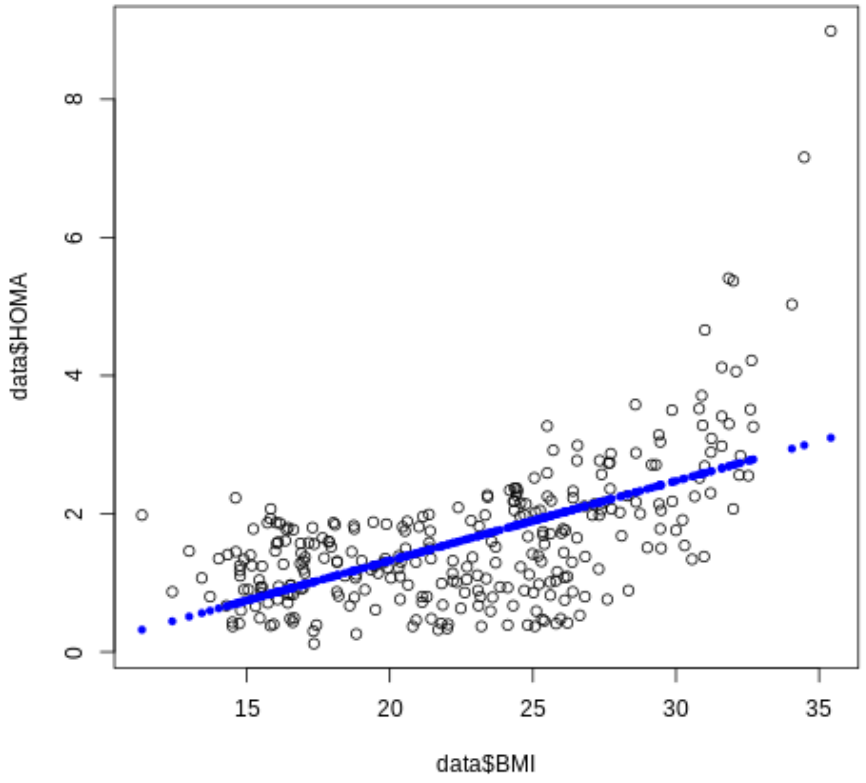
In [4]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Visualización de las variable respecto a HOMA
temp <- data
plotY <- function (x,y) {
        plot(temp[,y]~temp[,x], xlab=paste(names(temp)[x]," X",x,sep=""), ylab=n
ames(temp)[y])
}
par(mfrow=c(4,4)) #Si margin too large => (5,3)
x <- sapply(1:(dim(temp)[2]-1), plotY, dim(temp)[2])
par(mfrow=c(1,1))

#cor(data) # Descomentar si queremos ver los valores concretos de correlación
```



Como resultado, podemos ver como las variables BMI, WC y $Height$ parecen ser las más prometedoras (ya que muestran una relación relativamente lineal con el $HOMA$), a pesar de mostrar cierta dispersión en los datos. Esto último es una señal de que no hay un único factor explicativo de valor de la insulino-resistencia ($HOMA$). Además, se puede ver que las tres presentan cierta no linealidad, siendo este comportamiento más notorio para el BMI, el cual parece mostrar cierta relación cuadrática. En los siguientes bloques de código, nos centraremos en estas tres variables y aplicaremos una regresión lineal simple con cada una de ellas como una primera toma de contacto. Los siguientes bloques de código lanzan una regresión lineal simple entre el $HOMA$ y BMI, $Height$ y WC respectivamente.

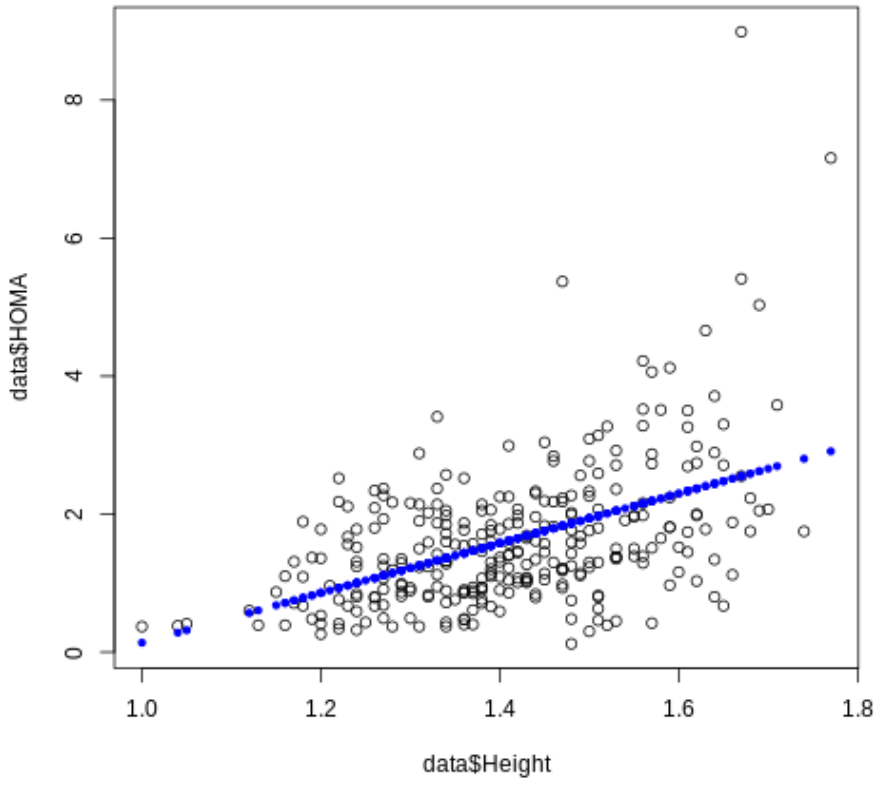
In [5]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Obteción del modelo. Función lm() del paquete ISLR.
### Y=HOMA, X's=BMI (índice de masa corporal) -> formula: HOMA ~ BMI
fitLM <- lm(HOMA ~ BMI, data=data)

### Visualización de la línea (azul, valores estimados) vs valores reales (negr
o, valores observados).
yprime = predict(fitLM,data)
plot(data$HOMA~data$BMI)
points(data$BMI,yprime,col="blue",pch=20)

### Coeficientes (Estimate), p-valores (Pr(>|t|)), R2 ajustado (Adjusted R-squar
ed),
### estadístico F y p-valor (F-statistic y p-value)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ BMI, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6176 -0.5495 -0.0203  0.5005  5.8905

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.987863   0.216053  -4.572 7.15e-06 ***
BMI          0.115430   0.009277  12.442  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8538 on 290 degrees of freedom
Multiple R-squared:  0.348,     Adjusted R-squared:  0.3458
F-statistic: 154.8 on 1 and 290 DF,  p-value: < 2.2e-16
```



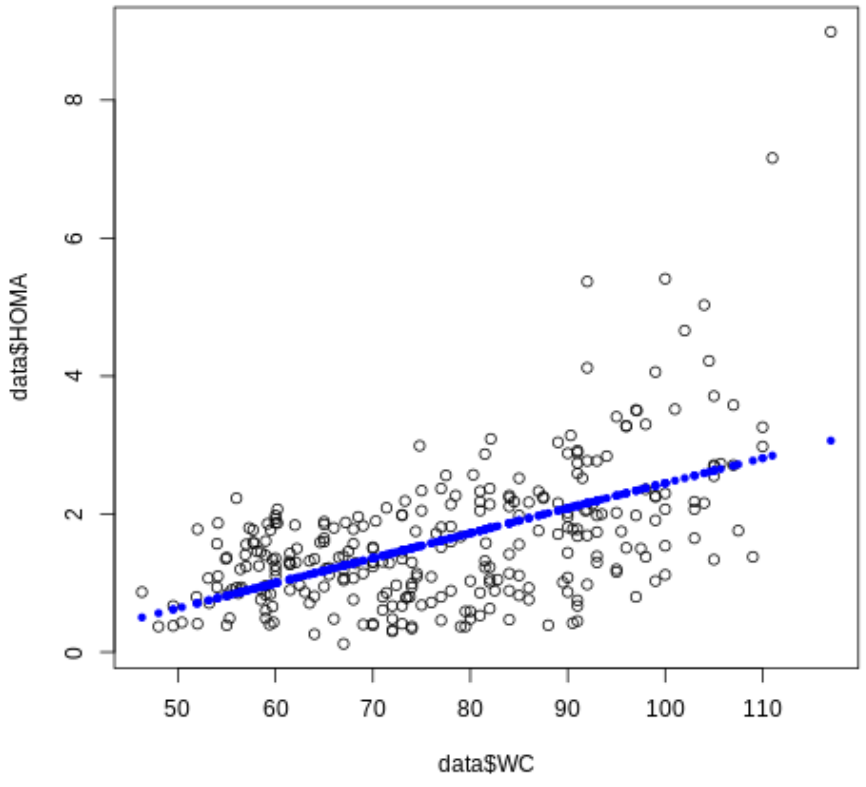
In [6]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Idem para la variable Height (altura)
fitLM <- lm(HOMA ~ Height, data=data)
yprime = predict(fitLM,data)
plot(data$HOMA~data$Height)
points(data$Height,yprime,col="blue",pch=20)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ Height, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.8080 -0.6062 -0.1728  0.4393  6.4400 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -3.4639     0.5467  -6.336    9e-10 ***
Height        3.6012     0.3848   9.359   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9267 on 290 degrees of freedom
Multiple R-squared:  0.232,     Adjusted R-squared:  0.2293 
F-statistic: 87.58 on 1 and 290 DF,  p-value: < 2.2e-16
```



In [7]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Idem para la variable WC (circunferencia de la cintura)
fitLM <- lm(HOMA ~ WC, data=data)
yprime = predict(fitLM,data)
plot(data$HOMA~data$WC)
points(data$WC,yprime,col="blue",pch=20)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ WC, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6840 -0.5542 -0.0354  0.4949  5.9256

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.175876   0.260223  -4.519 9.07e-06 ***
WC           0.036241   0.003296  10.995  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8883 on 290 degrees of freedom
Multiple R-squared:  0.2942,    Adjusted R-squared:  0.2918
F-statistic: 120.9 on 1 and 290 DF,  p-value: < 2.2e-16
```



Como resultado, podemos observar que los **p-valores** asociados a los **coeficientes** de las tres variables (columna con nombre "$Pr(>|t|)$") indican claramente que las **tres están relacionadas con la insulino-resistencia** (ya que adquieren p-valores muy por debajo de 0,1). Si bien es el modelo de regresión basado en (BMI) el que explica más variabilidad del $HOMA$ (de acuerdo a su valor de R^2 **ajustado**), es cierto que no presenta un valor muy alto (0,3458).

Alternativamente, repetiremos el proceso de selección de variables pero esta vez llevando a cabo el mencionado **enfoque descendente**, el cual es aplicable ya que sólo contamos con 15 variables de entrada.

4. SELECCIÓN ADITIVA DE VARIABLES: ENFOQUE DESCENDENTE

A continuación, dejamos atrás las **regresiones lineales simples** y pasamos a considerar el modelo de **regresión múltiple**. Como ya se indicó, seguiremos un enfoque de **selección de variables descendente**. En los siguientes bloques de código se muestran uno a uno los pasos realizados para que se pueda hacer un seguimiento de las decisiones tomadas en cada momento.

Tal y como hemos explicado en las secciones anteriores, la selección de variables mediante **enfoque descendente** se lleva a cabo incluyendo todas las variables en el modelo. Esto se consigue en *R* mediante el comando: (*Y* ~ .), donde el punto es la forma de indicar "*todas las variables de entrada disponibles en el conjunto de datos*".

In [8]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Obteción del modelo. Y=HOMA, X's=Todas -> formula: HOMA ~ .
fitLM <- lm(HOMA ~ ., data=data)

### Recordatorio:
### Coeficientes (Estimate), p-valores (Pr(>|t|)), R2 ajustado (Adjusted R-squar
ed),
### estadístico F y p-valor (F-statistic y p-value)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ ., data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4294 -0.4619 -0.0636  0.4089  5.1679

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.9193063  0.9463797  -6.255 1.51e-09 ***
Sex          0.3046482  0.0961801   3.167  0.00171 **
Age          0.0056148  0.0431482   0.130  0.89656
Tanner       0.1345579  0.1352788   0.995  0.32077
Height       2.4505541  0.7621021   3.216  0.00146 **
BMI          0.1455896  0.0228539   6.370 7.86e-10 ***
WC          -0.0247314  0.0086550  -2.857  0.00460 **
TAGmgDL      0.0072443  0.0016646   4.352 1.90e-05 ***
HDLCmgDL     0.0082428  0.0040629   2.029  0.04344 *
LDLCmgDL    -0.0031440  0.0017745  -1.772  0.07753 .
SBP          0.0041084  0.0043852   0.937  0.34964
DBP          0.0086374  0.0054648   1.581  0.11513
Sedentary    0.0010000  0.0005885   1.699  0.09038 .
Light        0.0013119  0.0010782   1.217  0.22473
Moderate     0.0038260  0.0047826   0.800  0.42441
Vigorous    -0.0056749  0.0057629  -0.985  0.32562
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7556 on 276 degrees of freedom
Multiple R-squared:  0.514,     Adjusted R-squared:  0.4876
F-statistic: 19.46 on 15 and 276 DF,  p-value: < 2.2e-16
```

Una vez obtenido el modelo de **regresión lineal múltiple** con todas las variables de entrada, hay un punto muy importante que no debemos pasar por alto. Este es el valor obtenido para el **estadístico F**, el cual debe ser lo primero que comprobemos. Como se explicó en los conceptos básicos, si su valor es cercano a 1 y/o paralelamente su ***p-valor*** está por encima de $0,1$ o $0,15$ no existiría ninguna variable que presente relación lineal con la variable de salida *HOMA-IR*. Esta interpretación **siempre será independientemente del p-valor individual** obtenido para cada coeficiente, el cual podría engañarnos hasta que no vayamos eliminando variables de entrada redundantes o no informativas. De darse esa situación, se detendría el análisis de regresión lineal y buscaríamos otra técnica de regresión alternativa. Como ya sabíamos por los valores obtenidos en el apartado anterior, este no es el caso de nuestro conjunto de datos.

Chequeando los **p-valores** obtenidos tras este primer paso, podemos ver que el siguiente paso sería eliminar la variable Age por presentar el p-valor más alto, 0,89656. Un detalle interesante es que el R^2 **ajustado** del modelo completo mejora con respecto a los R^2 obtenidos para los modelos de regresión lineal simple del apartado anterior (alcanzando un valor de 0,4876). Para eliminar la variable Age del modelo completo lo hacemos utilizando el comando *R* de signo de sustracción "-" en la fórmula, obteniendo así el nuevo modelo.

In [9]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Obteción del modelo. Y=HOMA, X's=Todas-Age -> formula: HOMA ~ .-Age
fitLM <- lm(HOMA ~ .-Age, data=data)

summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ . - Age, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.4365 -0.4646 -0.0683  0.4115  5.1690 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5.9584990  0.8955812  -6.653 1.53e-10 ***
Sex          0.3034817  0.0955914   3.175  0.00167 ** 
Tanner       0.1396497  0.1292663   1.080  0.28094    
Height       2.5180619  0.5572805   4.518 9.23e-06 ***
BMI          0.1452026  0.0226194   6.419 5.92e-10 ***
WC          -0.0245827  0.0085641  -2.870  0.00441 ** 
TAGmgDL      0.0072474  0.0016615   4.362 1.82e-05 ***
HDLCmgDL     0.0083213  0.0040108   2.075  0.03894 *  
LDLCmgDL    -0.0031315  0.0017687  -1.771  0.07774 .  
SBP          0.0040640  0.0043641   0.931  0.35255    
DBP          0.0086128  0.0054518   1.580  0.11529    
Sedentary    0.0010094  0.0005831   1.731  0.08455 .  
Light        0.0012833  0.0010537   1.218  0.22431    
Moderate     0.0038748  0.0047594   0.814  0.41627    
Vigorous    -0.0057566  0.0057184  -1.007  0.31496    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7543 on 277 degrees of freedom
Multiple R-squared:  0.514,     Adjusted R-squared:  0.4894 
F-statistic: 20.92 on 14 and 277 DF,  p-value: < 2.2e-16
```

Tras eliminar Age, se puede ver cómo el nuevo R^2 **ajustado** del modelo incluso mejora. Esto es el resultado de eliminar una variable que no aportaba nada al modelo. El siguiente paso, en vista de los resultados obtenidos, será eliminar la variable de actividad física $Moderate$.

In [10]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Idem al anterior -Moderate
fitLM <- lm(HOMA ~ .-Age-Moderate, data=data)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ . - Age - Moderate, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.3679 -0.4614 -0.0789  0.4085  5.1735 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5.8816129  0.8900476  -6.608 1.98e-10 ***
Sex          0.2850872  0.0928265   3.071  0.00234 ** 
Tanner       0.1457291  0.1289721   1.130  0.25948    
Height       2.5165450  0.5569393   4.519 9.22e-06 ***
BMI          0.1475379  0.0224231   6.580 2.34e-10 ***
WC          -0.0252296  0.0085219  -2.961  0.00334 ** 
TAGmgDL      0.0072720  0.0016602   4.380 1.68e-05 ***
HDLCmgDL     0.0084659  0.0040044   2.114  0.03539 *  
LDLCmgDL    -0.0032874  0.0017572  -1.871  0.06243 .  
SBP          0.0041578  0.0043599   0.954  0.34109    
DBP          0.0079398  0.0053855   1.474  0.14154    
Sedentary    0.0009849  0.0005820   1.692  0.09168 .  
Light        0.0016399  0.0009578   1.712  0.08799 .  
Vigorous    -0.0031646  0.0047472  -0.667  0.50556    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7538 on 278 degrees of freedom
Multiple R-squared:  0.5128,    Adjusted R-squared:   0.49 
F-statistic: 22.51 on 13 and 278 DF,  p-value: < 2.2e-16
```

Conforme se van eliminando variables no informativas, vemos como el valor R^2 **ajustado** del modelo resultante sigue mejorando poco a poco. La siguiente variable a eliminar es $Vigorous$ (referida a los minutos diarios de actividad física vigorosa de los sujetos).

En este punto queda perfectamente claro como debe acometerse el proceso de ir eliminando variables no informativas de una en una. Siempre debemos hacerlo así aunque resulte tedioso. En lo que sigue se mostrarán los siguientes pasos comentados hasta llegar al último paso (modelo final), el cual si que ejecutaremos para ver a qué se llega finalmente. Por favor, recuerde leer también los comentarios en línea con detenimiento.

In [11]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
#fitLM <- lm(HOMA ~ .-Age-Moderate-Vigorous, data=data)
#summary(fitLM)
#fitLM <- lm(HOMA ~ .-Age-Moderate-Vigorous-SBP, data=data)
#summary(fitLM)
#fitLM <- lm(HOMA ~ .-Age-Moderate-Vigorous-SBP-Tanner, data=data)
#summary(fitLM)

### En el modelo anterior ya todos los p-valores se podrían considerar correcto
s.
### Por simplicidad hemos seguido quitando mientras el R2 ajustado apenas se ha
 visto afectado.
#fitLM <- lm(HOMA ~ .-Age-Moderate-Vigorous-SBP-Tanner-Light, data=data)
#summary(fitLM)
#fitLM <- lm(HOMA ~ .-Age-Moderate-Vigorous-SBP-Tanner-Light-Sedentary, data=dat
a)
#summary(fitLM)

### A partir de aquí R2 empezaría a empeorar significativamente.
### Paramos y reformulamos por legibilidad indicando las variables de entrada se
leccionadas de manera aditiva
### Este modelo es equivalente al inmediatamente anterior pero muestra con clari
dad lo seleccionado
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+DBP, data=data) #
Vea que ya no se incluye el punto
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ BMI + Height + TAGmgDL + Sex + WC + LDLCmgDL +
    HDLCmgDL + DBP, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4022 -0.4525 -0.0408  0.3880  5.0570

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.133154   0.656353  -7.821 1.05e-13 ***
BMI          0.156909   0.021985   7.137 8.04e-12 ***
Height       2.764817   0.413419   6.688 1.21e-10 ***
TAGmgDL      0.007288   0.001646   4.428 1.36e-05 ***
Sex          0.307461   0.090695   3.390 0.000798 ***
WC          -0.027091   0.008387  -3.230 0.001383 **
LDLCmgDL    -0.003229   0.001751  -1.844 0.066178 .
HDLCmgDL     0.008966   0.003986   2.250 0.025241 *
DBP          0.009670   0.005138   1.882 0.060843 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7567 on 283 degrees of freedom
Multiple R-squared:  0.5002,    Adjusted R-squared:  0.4861
F-statistic:  35.4 on 8 and 283 DF,  p-value: < 2.2e-16
```

Como resultado, obtenemos un modelo con 8 variables de entrada y un R^2 **ajustado** de 0,4861.

5. INTERACCIONES Y NO LINEALIDAD

Una vez hemos seleccionado las variables de entrada que deben incorporarse a nuestro **modelo lineal**, intentaremos poder explicar la parte **no lineal** de los datos mediante la adición de **interacciones y otros términos no lineales**. Para probar interacciones nos basamos en el conocimiento previo que tengamos del problema (por ejemplo, en un caso de interacción genética conocida entre dos variantes genéticas, sería apropiado introducir un término de interacción entre ambas para modelar su efecto sobre la variable de salida). En caso de que no haya información previa sobre fenómenos de interacción, también podremos guiarnos por la lógica o intuición según el significado de las variables de entrada. Si aún así no caemos en ninguna posible interacción, podemos hacer pruebas aleatorias entre las variables que se han mostrado más significativas (ensayo-error). Este procedimiento no es un proceso trivial y depende de nuestra propia habilidad y experiencia.

En nuestro caso de estudio sobre obesidad infantil, vamos a comprobar si existe sinergia positiva (factores multiplicativos, $*$) entre la variable de triglicéridos y las dos medidas de colesterol (por pertenecer todas ellas al perfil lipídico).

In [12]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Interacciones entre triglicéridos y colesteroles
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+DBP+TAGmgDL*HDLCm
gDL*LDLCmgDL, data=data)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ BMI + Height + TAGmgDL + Sex + WC + LDLCmgDL +
    HDLCmgDL + DBP + TAGmgDL * HDLCmgDL * LDLCmgDL, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5603 -0.4625 -0.0600  0.3862  4.9757

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                -4.756e+00  1.512e+00  -3.144 0.001844 **
BMI                         1.580e-01  2.221e-02   7.113 9.59e-12 ***
Height                      2.799e+00  4.170e-01   6.712 1.07e-10 ***
TAGmgDL                     1.097e-02  1.748e-02   0.627 0.531002
Sex                         3.130e-01  9.187e-02   3.408 0.000752 ***
WC                         -2.775e-02  8.486e-03  -3.270 0.001209 **
LDLCmgDL                   -9.731e-03  1.388e-02  -0.701 0.483962
HDLCmgDL                    1.049e-03  2.518e-02   0.042 0.966806
DBP                         9.692e-03  5.168e-03   1.876 0.061766 .
TAGmgDL:HDLCmgDL           -7.809e-05  3.563e-04  -0.219 0.826677
TAGmgDL:LDLCmgDL            2.741e-06  1.649e-04   0.017 0.986754
LDLCmgDL:HDLCmgDL           1.366e-04  2.554e-04   0.535 0.593230
TAGmgDL:LDLCmgDL:HDLCmgDL -1.602e-07  3.308e-06  -0.048 0.961424
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.76 on 279 degrees of freedom
Multiple R-squared:  0.503,     Adjusted R-squared:  0.4816
F-statistic: 23.53 on 12 and 279 DF,  p-value: < 2.2e-16
```

Véase que el uso de operadores ya incluye todos los términos de jerarquía. Si no lo hiciese, tendríamos que añadirlos a mano antes de mirar ningún p-valor, ni de tomar decisión alguna. Podemos ver cómo el término *TAGmgDL:LDLCmgDL:HDLCmgDL* presenta un p-valor realmente malo (0,961424), indicando que no existe dicha interacción hipotetizada.

Probaremos de nuevo con la altura y la circunferencia de la cintura.

In [13]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Interacciones entre altura y la circunferencia de la cintura
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+DBP+Height*WC, da
ta=data)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ BMI + Height + TAGmgDL + Sex + WC + LDLCmgDL +
    HDLCmgDL + DBP + Height * WC, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5403 -0.4401 -0.0268  0.4154  4.3876

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.916865   2.172910   2.263  0.02441 *
BMI          0.159749   0.021173   7.545 6.27e-13 ***
Height      -4.246495   1.504020  -2.823  0.00509 **
TAGmgDL      0.007283   0.001584   4.597 6.49e-06 ***
Sex          0.275412   0.087561   3.145  0.00184 **
WC          -0.159204   0.028498  -5.587 5.45e-08 ***
LDLCmgDL    -0.003340   0.001685  -1.982  0.04844 *
HDLCmgDL     0.008494   0.003838   2.213  0.02769 *
DBP          0.010512   0.004949   2.124  0.03453 *
Height:WC    0.090497   0.018721   4.834 2.20e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7285 on 282 degrees of freedom
Multiple R-squared:  0.5384,    Adjusted R-squared:  0.5237
F-statistic: 36.55 on 9 and 282 DF,  p-value: < 2.2e-16
```

En este caso si podemos ver cómo la interacción explica parte de esa **no linealidad** (al obtener un p-valor inferior a 0.1). En principio nos la quedamos como parte del modelo.

Por último, probaremos otros términos de **no linealidad**. En este caso, teniendo en cuenta que inicialmente graficamos todas las variables de entrada respecto a la variable de salida *HOMA-IR*, resulta bastante más sencillo determinar ciertos tipos de comportamientos no lineales de manera visual. En concreto, vimos que la variable de salida $HOMA-IR$ parecía tener una relación cuadrática con BMI. Probaremos, por lo tanto, a incluir dicho término. La forma de hacerlo es mediante la función $I(.)$ de R. La potencia se indica de la siguiente manera: $I(X_j\hat{}exponente)$. En nuestro caso $I(BMI\hat{}2)$. La función $I(.)$ no genera automáticamente los términos de jerarquía, por lo que antes de mirar siquiera el modelo debemos asegurarnos de que todos los términos de jerarquía están en la fórmula. Para el caso que nos aplica, los terminos de jerarquía serían:
$BMI + BMI^2 + Height + WC + Height * WC$

In [14]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
### Interacciones entre altura y la circunferencia de la cintura, más BMI^2
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+DBP+Height*WC+I(BMI^2), data=data)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ BMI + Height + TAGmgDL + Sex + WC + LDLCmgDL + 
    HDLCmgDL + DBP + Height * WC + I(BMI^2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.9267 -0.3865  0.0019  0.3615  3.4774 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.308654   1.964056   2.194  0.02907 *  
BMI         -0.443050   0.077356  -5.727 2.62e-08 ***
Height       0.682096   1.490290   0.458  0.64753    
TAGmgDL      0.006089   0.001439   4.232 3.14e-05 ***
Sex          0.237497   0.079227   2.998  0.00296 ** 
WC          -0.041319   0.029621  -1.395  0.16414    
LDLCmgDL    -0.002949   0.001523  -1.936  0.05383 .  
HDLCmgDL     0.005186   0.003491   1.486  0.13852    
DBP          0.005995   0.004505   1.331  0.18434    
I(BMI^2)     0.012615   0.001569   8.042 2.48e-14 ***
Height:WC    0.016942   0.019224   0.881  0.37890    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.658 on 281 degrees of freedom
Multiple R-squared:  0.6248,    Adjusted R-squared:  0.6115 
F-statistic: 46.79 on 10 and 281 DF,  p-value: < 2.2e-16
```

De nuevo, aparecen términos con p-valores altos, como es el caso de $Height * WC$. La interacción estaba intentando explicar lo que ahora el término BMI cuadrático explica mejor. Quitaremos por lo tanto la interacción ahora no significativa. Recordemos que la variable BMI ya mostraba claramente un aspecto no lineal cuadrático en el gráfico.

In [15]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+DBP+I(BMI^2), dat
a=data)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ BMI + Height + TAGmgDL + Sex + WC + LDLCmgDL +
    HDLCmgDL + DBP + I(BMI^2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8747 -0.3846 -0.0021  0.3674  3.5269

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.821314   1.004299   2.809  0.00531 **
BMI         -0.474890   0.068374  -6.945 2.61e-11 ***
Height       1.954571   0.369059   5.296 2.39e-07 ***
TAGmgDL      0.006028   0.001437   4.196 3.64e-05 ***
Sex          0.240162   0.079138   3.035  0.00263 **
WC          -0.016037   0.007380  -2.173  0.03060 *
LDLCmgDL    -0.002912   0.001522  -1.914  0.05666 .
HDLCmgDL     0.005082   0.003488   1.457  0.14619
DBP          0.005638   0.004485   1.257  0.20978
I(BMI^2)     0.013273   0.001379   9.624  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6577 on 282 degrees of freedom
Multiple R-squared:  0.6238,    Adjusted R-squared:  0.6118
F-statistic: 51.95 on 9 and 282 DF,  p-value: < 2.2e-16
```

De acuerdo al nuevo modelo, ahora habría que eliminar DBP.

In [16]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+I(BMI^2), data=da
ta)
summary(fitLM)
```

```
Call:
lm(formula = HOMA ~ BMI + Height + TAGmgDL + Sex + WC + LDLCmgDL +
    HDLCmgDL + I(BMI^2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9051 -0.3843 -0.0047  0.3459  3.5083

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.176416   0.964737   3.293  0.00112 **
BMI         -0.480043   0.068321  -7.026 1.58e-11 ***
Height       1.930855   0.368954   5.233 3.25e-07 ***
TAGmgDL      0.006063   0.001438   4.217 3.33e-05 ***
Sex          0.241655   0.079210   3.051  0.00250 **
WC          -0.015478   0.007374  -2.099  0.03669 *
LDLCmgDL    -0.002774   0.001519  -1.826  0.06898 .
HDLCmgDL     0.005268   0.003488   1.510  0.13208
I(BMI^2)     0.013435   0.001375   9.774  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6584 on 283 degrees of freedom
Multiple R-squared:  0.6217,    Adjusted R-squared:  0.611
F-statistic: 58.13 on 8 and 283 DF,  p-value: < 2.2e-16
```

Y por último $HDLCmgDL$.

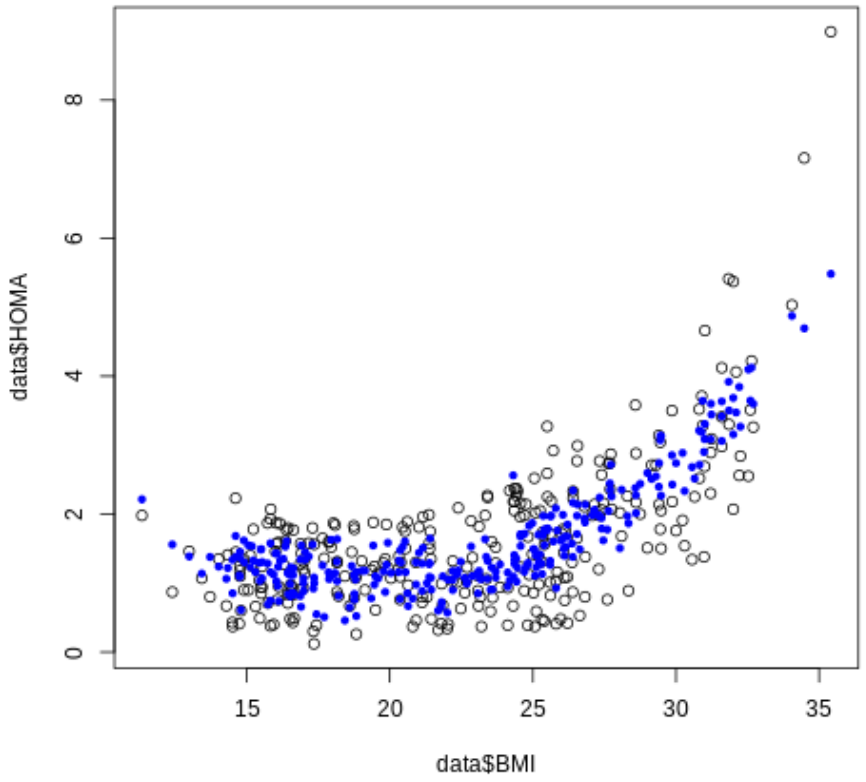
En una línea de código adicional, incluimos también cómo calcular la métrica **RECM** para un modelo de regresión lineal (la cual no aparecía en el *output* de resultados ofrecido por el comando *summary*).

In [17]:
```
# Tiempo estimado de ejecución: 3 segundos aprox.

%%R
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+I(BMI^2), data=data)

### Cálculo de RECM
yprime = predict(fitLM,data)
cat('\nRMSE:', sqrt(sum((data$HOMA-yprime)^2)/length(yprime)), "\n") #RECM->en i
nglés RMSE

summary(fitLM)
```

```
RMSE: 0.6507716

Call:
lm(formula = HOMA ~ BMI + Height + TAGmgDL + Sex + WC + LDLCmgDL +
    I(BMI^2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8923 -0.3932  0.0004  0.3582  3.5091

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.797671   0.874612   4.342 1.97e-05 ***
BMI         -0.495014   0.067751  -7.306 2.79e-12 ***
Height       1.952522   0.369506   5.284 2.52e-07 ***
TAGmgDL      0.005433   0.001379   3.940 0.000103 ***
Sex          0.224973   0.078613   2.862 0.004526 **
WC          -0.016985   0.007322  -2.320 0.021069 *
LDLCmgDL    -0.002803   0.001523  -1.841 0.066719 .
I(BMI^2)     0.013684   0.001368  10.006  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6599 on 284 degrees of freedom
Multiple R-squared:  0.6186,    Adjusted R-squared:  0.6092
F-statistic: 65.81 on 7 and 284 DF,  p-value: < 2.2e-16
```

Visualización:

In [18]:
```
#yprime = predict(fitLM,data)

%%R
plot(data$HOMA~data$BMI)
points(data$BMI,yprime,col="blue",pch=20)
```



Finalmente, hemos llegado hasta un R^2 **ajustado** de 0,6092 cuando partíamos de 0,3458. Pero lo más importante no es dicho valor en sí, sino lo que hemos podido aprender sobre los datos y de nuestro problema basándonos en valores estadísticos. Como principal conclusión, podemos extraer que un alto índice de masa corporal en niños es uno de los principales factores de riesgo para padecer insulino-resistencia. Pensemos que incluso aunque la conclusión fuese que tenemos que replantear el problema con nuevas variables y mediciones, ya es un gran paso el poder darse cuenta.

6. VALIDACIÓN CRUZADA

Una vez que ya tenemos planteada la mejor fórmula para aplicar el ajuste paramétrico (modelo de regresión), si quisieramos estimar nuevos valores de la variable de salida y comparar su habilidad predictiva con otros modelos deberíamos aplicar una validación cruzada por los motivos que se explicaron en el Módulo 3 de este $MOOC$. A continuación, se muestra como poder hacerlo.

In [19]:

```
%%R
set.seed(123456)
k <- 5
data$kfold <- sample(1:k, nrow(data), replace = T)

performances <- c()

# One iteration per fold
for (fold in 1:k){
  # Se crea el conjunto de entrenamiento para la iteración
  training_set <- data[data$kfold != fold,]
  nombres <- names(training_set)
  tam <- length(nombres)-1
  training_set <- training_set[,nombres[1: tam]]

  # Create test set for this iteration
  # Subset all the datapoints where .folds matches the current fold
  testing_set <- data[data$kfold == fold,]
  nombres <- names(testing_set)
  tam <- length(nombres)-1
  testing_set <- testing_set[,nombres[1: tam]]

  ## Entrenando el modelo para la iteración
  model <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+I(BMI^2), data=training_
set)

  ## Calculando el error de test
  yprime <- predict(model, testing_set)
  RMSE <- sqrt(sum((testing_set$HOMA-yprime)^2)/length(yprime))

  # Add the RMSE to the performance list
  performances[fold] <- RMSE
}

#Eliminamos la columna artificial añadida para kfold
#(para que no acumule columnas si se ejecuta varias veces)
nombres <- names(data)
tam <- length(nombres)-1
data <- data[,nombres[1: tam]]

cat("RECM medio en test para 5-fcv:", mean(performances))
```

```
RECM medio en test para 5-fcv: 0.6956061
```

REFERENCIAS BIBLIOGRÁFICAS

- Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. An Introduction to Statistical Learning with Applications in R Springer, 2013 (**Chapter 03**)
- McDonald, J.H. Handbook of Biological Statistics (3rd ed.). Sparky House Publishing, Baltimore, Maryland, 2014. Pages 190-208 in the printed version
- Usando rpy2 en notebooks: https://rpy2.github.io/doc/latest/html/notebooks.html (https://rpy2.github.io/doc/latest/html/notebooks.html)
- Usando read.csv de R: https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/read.table (https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/read.table)
- Usando ISLR: https://cran.r-project.org/web/packages/ISLR/index.html (https://cran.r-project.org/web/packages/ISLR/index.html)

REFERENCIAS ADICIONALES

- M.J. Gacto, J.M. Soto-Hidalgo, J. Alcalá-Fdez, and R. Alcalá (2019). Experimental Study on 164 Algorithms Available in Software Tools for Solving Standard Non-Linear Regression Problems. IEEE Access 7, 2019, pp. 108916-108939; https://doi.org/10.1109/ACCESS.2019.2933261 (https://doi.org/10.1109/ACCESS.2019.2933261)