Module 4 - Supervised Learning: Regression Techniques.

4.2 Standard regression methods

Authors:

By Rafael Alcalá

Professor at the University of Granada, , Andalusian Inter-University Institute in Data Science and Computational Intelligence (DaSCI).

by Augusto Anguita-Ruiz

Postdoctoral Research Fellow at Barcelona Institute for Global Health- ISGlobal.

Reminder: Introduction to NoteBook.

In this *NoteBook* you will be guided, step-by-step, through loading a dataset to the descriptive analysis of its contents. The Jupyter NoteBook (*Python*) is an approach that combines text blocks (like this one) and code blocks or cells. The great advantage of this system is its interactivity because cells can be executed to directly check the results they contain.

Very important: the order of the instructions is fundamental and so each cell in this NoteBook must be executed sequentially. If any are omitted, the program may throw an error and so if there is any doubt, you will have to start from the beginning again.

First, it is very important to select "*Open in draft mode*" (draft mode) at the top left at the beginning. Otherwise, for security reasons, you will not be allowed to execute any code blocks. When the first of the blocks is executed, the following message will appear: "*Warning: This NoteBook was not created by Google*". Don't worry, you can trust the contents of the NoteBook and click on "*Run anyway*".

Let's start!

Click on the "*play*" button on the left side of each code cell. Remember that lines beginning with a hashtag (#) are comments and do not affect the execution of the script. You can also click on each cell and press "Ctrl+enter" ("Cmd+Enter" on Mac). Each time you execute a block, you will see the output just below it. The information is usually always the last statement, along with any *print()* commands present in the code.

INDEX

In this *Notebook*:

1. We learn the general concepts of Simple and Multiple Linear regression technique.
2. We will apply multiple linear regression as the first tool used to study any regression problem, following examples with our insulin resistance estimation problem.

Contents:

1. LINEAR REGRESSION

The first **regression method** we will address in this module is **linear regression**, which is considered as a simple approach in the field of supervised learning. In **linear regression**, it is assumed that **dependence of the output variable** Y (in our case study of childhood obesity represented by the variable *HOMA-IR*) **on the input variables** $X_1, X_2, \ldots, X_p$ (all other variables in our problem) **is linear**. Although it may seem too simplistic, **linear regression** is extremely useful both conceptually and in practice. In this capsule we will see how to use it to study a data set and draw useful conclusions about the behaviour of the data, without prejudice to the use of other (supposedly or theoretically more powerful) techniques to obtain better-fitting models. First of all, we will distinguish between **simple or multiple linear regression**.

1.1 Simple linear regression basics

In a **simple linear regression**, using **a single input variable** X we assume the following model,

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

where β_0 and β_1 are two unknown constants that represent the independent term and the slope of a linear function (a straight line) respectively. These constants are known as **coefficients or parameters**. In this context, $$epsilon$ refers to the **error term of the estimate**.

Given an estimate of $\hat{\beta}_0$ and $\hat{\beta}_1$ for the model **coefficients**, we can predict future values of the output variable Y using,

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

where $\hat{y}$ represents a prediction of Y based on $X = x$. The hat symbol denotes that we are referring to **estimated values** (rather than actual or observed values). The values of $\hat{\beta}_0$ and $\hat{\beta}_1$ are obtained through the **least squares technique**, which obtains the coefficients that minimise the error made for each available instance in a training data set. In other words, with **least squares**, the optimal values of the **coefficients** that minimise the value of **RMSE** are always obtained.

1.2 Basic concepts of multiple linear regression

In the case of **multiple linear regression**, using **more than one input variable** X we assume the following model,

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

interpreting each coefficient $beta_j$ as the average effect on Y of a unit increase in X_j, keeping the rest of the input variables fixed. The ideal case is when the input variables $X's$ are not correlated with each other (no collinearity), as this would imply certain interpretation problems. However, if the learning process is carried out step by step, taking into account the statistical values obtained on these coefficients, the variables that are correlated with each other are eventually eliminated. In **multiple linear regression**, the estimation of the coefficients is also performed using the **least squares technique**, minimising the error of the predictions on the training data set.

1.3 Evaluation of the goodness of fit of the estimation coefficient

This technique generates certain **statistical values** that will allow us to answer questions such as whether there is at least one input variable (X_j) that has a linear relationship with the output variable (Y). If this is the case, which ones? Here we will refer to two **statistical values**:

- The *F-statistic*: This is a value obtained for the regression model as a whole, rather than for each of the input variables. The extent that its value departs from 1 (either upwards or downwards) indicates that at least one input variable has a linear relationship with the output variable. If this value is close to 1, we can directly discard the modelled linear regression since there will be no input variable (X_j) with a linear relationship with the output (Y). This parameter is particularly important in problems with many input variables which may include variables that initially appear to be relevant but are eventually found not to be. Thus, the **F statistic is** the first parameter we must study when performing a regression. After that, **we only continue with the process** if its value is not close to 1.
- **The p-values of the t-statistics**: This value is a parameter associated with each of the coefficients, although a global p-value is also obtained for the complete regression model. The p-values will indicate whether the coefficient $beta_j$ of each input variable (X_j) is relevant or if, on the contrary there is a high probability that it could actually be zero ($beta_j \approx 0$ implies that the variable in question has no linear relationship with the output variable and therefore, should be eliminated from the model). This happens when the p-value for the coefficient exceeds 0.1 or 0.15, indicating that the variable might be irrelevant ($beta_j \approx 0$).

1.4 Assessing the goodness of fit of the model

As mentioned in the previous capsule, the value of the **Coefficient of Determination** R^2 can fall between 0 and 1, with a value of 1 indicating a perfect model fit (zero error) and 0 indicating a fit with the worst possible error. To **compare several linear regression models with each other**, the value of R^2 obtained in each of them is usually considered. However, because the number of coefficients β_j of the different models must be taken into account, in order to compare models with different numbers of input variables, actually, we would have to look at the so-called adjusted R^2**$** (which already takes them into account during its calculation).

If we want to compare a **linear regression model** with models obtained by **other regression techniques** (e.g., KNN, neural networks, M5, ...) we must use **the value of RMSE**. Let us remember that, on the other hand, R^2 is a relative parameter and that not all techniques allow its calculation.

1.5 Choosing the relevant input variables

The usefulness of linear regression depends very much on the variables chosen to learn its **coefficients**. In principle, these must be manually selected because not all possible combinations of variables can be examined. For example, for a problem **with 40 input variables**, $p = 40$, we would have 2^p combinations of different models **(over a trillion combinations)**. **To decide which variables should be considered in the final model, there are** two approaches** commonly used:

1. **Forward selection**: This procedure consists of starting by building the **null model** which contains the independent term but no input variables, and then running simple linear regressions in parallel between the output/dependent variable Y and each of the input variables X_j. We will then add the input variable that results in **the lowest error among all models tested (the one with the highest adjusted R^2)** to the null model. The procedure is then continued until some stopping rule is satisfied. For example, adding any of the remaining variables **gives a p-value above some threshold for that variable (>0.1 or 0.15 for example)**. **IMPORTANT!!!:** the variables must be added one at a time and NEVER several at once. Each time a variable is added, it contributes to explaining a part of the data and, so the p-values of the other variables may change and no longer be relevant as there is no longer a need to explain that part of the data.
2. **Backward selection**: Start with all the variables in the model and then eliminate the input variable with the highest p-value (i.e., the least statistically significant variable), and fit the new model with the remaining $(p - 1)$ variables. In the next round, we again remove the variable with the highest p-value, and so on, until a stopping rule is reached. For example, we can stop when all the input variables that make up the model have a significant value (e.g., a p-value below 0.1 or 0.15). **IMPORTANT!!!:** the variables must be eliminated one at a time and NEVER several at once. Each time a variable is eliminated, we make it easier for one of the remaining variables to adequately explain part of the data so that they can go from being a variable of little relevance (or even an annoying variable with a very high p-value) to an essential one. Therefore, these variables would be lost if we were to remove several at once. If this process is followed strictly, it will solve most of the correlation problems (collinearity) between the input variables.

It is important to consider that the **independent term** of the regression does not come into play in either of these two approaches, and so it will never be removed from the model.

There are other possibilities besides these two approaches, such as calculating the correlations of all the input variables with the output variable, and keeping a small set of the best ones; or, for example, applying one of the existing variable selection algorithms. **Unfortunately, none of these ensures that the best existing combination is reached**.

In this course, **we recommend using backward selection when the number of variables is not too large**, and **forward selection when there are many variables** (automating the choice of the variable to be included in each step).

Using either of these two approaches will allow us to do exactly what we set out to do with regression: **study a data set and be able to draw useful conclusions** about the behaviour of the data, even if other, more versatile types of techniques are eventually applied. Thus, in our **dataset on obesity and insulin resistance in children**, we will apply **the backward selection approach**.

1.6 Extensions of the linear model.Eliminating the additive assumption: interactions and non-linearity.

To deal with the **non-linearity** of the data existing in most real-life data sets, new **interaction terms** (variables with positive synergy that enhance each other) or **non-linearity** (variables with quadratic, logarithmic growth, etc.) can be considered. In order to study these behaviours in a **linear regression** we must use one of two different terms, as outlined below.

- **Interactions** (terms whose behavior is not additive): these interaction terms are presented in regression models as $X_1 * X_2$, and represent how the change in two or more variables jointly causes changes in the output variable Y greater than they would do separately. For example: it is known that investing 5000 euros of advertising in $radio$ and another 5000 in $television$ leads to much higher sales of a product than if 10000 euros are invested directly in either of the two media unilaterally. Including a new multiplicative term of the type $radio * television$ in the regression model would allow us to adequately explain such a non-linear phenomenon.
- **Other non-linear terms**: Often the relationship between an input variable and the output variable is not linear but quadratic, cubic, logarithmic, exponential, etc. Including a term that matches this type of relationship can help to adequately explain these non-linear phenomena.

However, the **principle of hierarchy** must be respected in every case. That is, if a new variable $(X_5)^3$ is included for X_5 and its p-value indicates that this cubic term is relevant, then $(X_5)^2$ and X_5 must also be included, even if their p-values are not significant; not including them would represent a serious mistake. The same rule follows in the case of interactions: if a new variable $(X_1 * X_2 * X_6)$ is included and retained because the three variables complement each other (meaning that the p-value for that interaction term is significant), then $(X_1 * X_2)$, $(X_1 * X_6)$, $(X_2 * X_6)$, X_1, X_2, and X_6 must also be included (irrespective of their p-values).

2. INSTALLATION OF R AND LIBRARIES , AND READING THE CHILDHOOD OBESITY DATA

As explained in the previous capsule, we must run the following three code cells before starting the linear regression algorithm.

In [1]:

```
# Estimated execution time: approx. 20 seconds.

### Installing R on Google Colab notebooks ###
!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 k
B]
Ign:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/
x86_64  InRelease
Get:3 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InReleas
e [3,626 B]
Hit:4 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:5 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InReleas
e [15.9 kB]
Ign:6 https://developer.download.nvidia.com/compute/machine-learning/repo
s/ubuntu1804/x86_64  InRelease
Get:7 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/
x86_64  Release [696 B]
Hit:8 https://developer.download.nvidia.com/compute/machine-learning/repo
s/ubuntu1804/x86_64  Release
Get:9 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/
x86_64  Release.gpg [836 B]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:11 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ Package
s [76.8 kB]
Hit:12 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Get:13 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 k
B]
Hit:14 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Get:15 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64
Packages [806 kB]
Get:16 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelea
se [21.3 kB]
Get:17 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Pa
ckages [1,474 kB]
Get:18 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packag
es [2,596 kB]
Get:20 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu180
4/x86_64  Packages [930 kB]
Get:21 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main So
urces [1,827 kB]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Pa
ckages [840 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages
[3,035 kB]
Get:24 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main am
d64 Packages [936 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Pack
ages [2,252 kB]
Get:26 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic/main am
d64 Packages [42.8 kB]
Fetched 15.1 MB in 4s (3,879 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
r-base is already the newest version (4.1.2-1.1804.0).
The following package was automatically installed and is no longer require
d:
  libnvidia-common-470
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 69 not upgraded.
Requirement already satisfied: rpy2 in /usr/local/lib/python3.7/dist-packa
ges (3.4.5)
Requirement already satisfied: cffi>=1.10.0 in /usr/local/lib/python3.7/di
st-packages (from rpy2) (1.15.0)
```

```
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-pac
kages (from rpy2) (2.11.3)
Requirement already satisfied: tzlocal in /usr/local/lib/python3.7/dist-pa
ckages (from rpy2) (1.5.1)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packa
ges (from rpy2) (2018.9)
Requirement already satisfied: pycparser in /usr/local/lib/python3.7/dist-
packages (from cffi>=1.10.0->rpy2) (2.21)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.
7/dist-packages (from jinja2->rpy2) (2.0.1)
Instalación de R en Google Colab terminada
```

In [2]:

```
# Estimated execution time: 4 seconds approx (not needing to import kknn and Cubist).
# Libraries needed:
# ISLR for multivariate linear regression
# kknn for k-nearest neighbours regression
# Cubist for M5-based regression models

%%R
### Installation of required libraries
#install.packages(c("ISLR", "kknn", "Cubist"))
install.packages(c("ISLR")) #kknn and Cubist will be used in the following capsule
print ("Installation of R libraries for this module completed")

### Import the required libraries
require(ISLR)
###require(kknn)
##require(Cubist)
print ("Import of the R libraries for this module is finished")
```

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

R[write to console]: trying URL 'https://cran.rstudio.com/src/contrib/ISLR
_1.4.tar.gz'

R[write to console]: Content type 'application/x-gzip'
R[write to console]:  length 1756715 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/RtmptZOy43/downloaded_packages'
R[write to console]:
R[write to console]:


[1] "Installation of R libraries for this module completed"

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


[1] "Import of the R libraries for this module is finished"
```

In [3]:

```
# Estimated execution time: 2 seconds approx.

%%R
### Loading
data <- read.csv(url("https://drive.google.com/uc?id=1GO2NBxYw54K6HkN-YgXbNadrLo5O6-0
u"))

### Visualisation of a small part of the data
head(data)
```

```
  Sex  Age Tanner Height   BMI   WC TAGmgDL HDLCmgDL LDLCmgDL SBP DBP Sede
ntary
1   1  9.5      0   1.55 11.34 60.0      55       51       93  97  60  41
1.0893
2   1  8.0      0   1.15 12.40 46.3      51       70       59  90  55  43
5.6071
3   0 10.5      0   1.42 12.99 67.5      65       60       96  96  54  48
3.9048
4   0  8.1      0   1.27 13.43 53.1      41       78      100 108  46  42
9.2976
5   1 10.4      0   1.32 13.72 51.9      39      100      120 107  69  51
2.0714
6   0 10.4      0   1.29 14.02 54.9      57       76       73  87  59  45
1.2321
     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
```

As we can see, the head command offers us a visualization of the available data for the first 6 individuals/instances of the set. In this visualization, we can identify variables such as the sex of the individuals (coded as 0 for boys or 1 for girls), the pubertal stage (represented by the Tanner variable, and coded as 0 for the pre-pubertal and 1 for pubertal), or the blood pressure (represented by the variables DBP for diastolic blood pressure and SBP for systolic blood pressure). As shown, there are also variables for a sedentary lifestyle, and light, moderate, or vigorous physical activity. The remaining abbreviated variables refer to: BMI (body mass index); WC (waist circumference); TAG (triglycerides); HDL (high-density lipoprotein or 'good' cholesterol); and LDL (low-density lipoprotein or 'bad' cholesterol), with the latter three being expressed in milligrams/deciliter of blood.

3. INITIAL ASSESSMENT OF THE PROBLEM AND STUDY OF THE VARIABLES OF GREATEST INTEREST

Once we have imported the libraries and read the data, we are ready to see **which variables are the most promising for applying linear regression to this problem**. To study which variables best explain the behavior of HOMA-IR as the output variable, we could **calculate the correlations** between it and each of the input variables (using the command: "*cor(data)*"), allowing us to choose those that are most correlated.

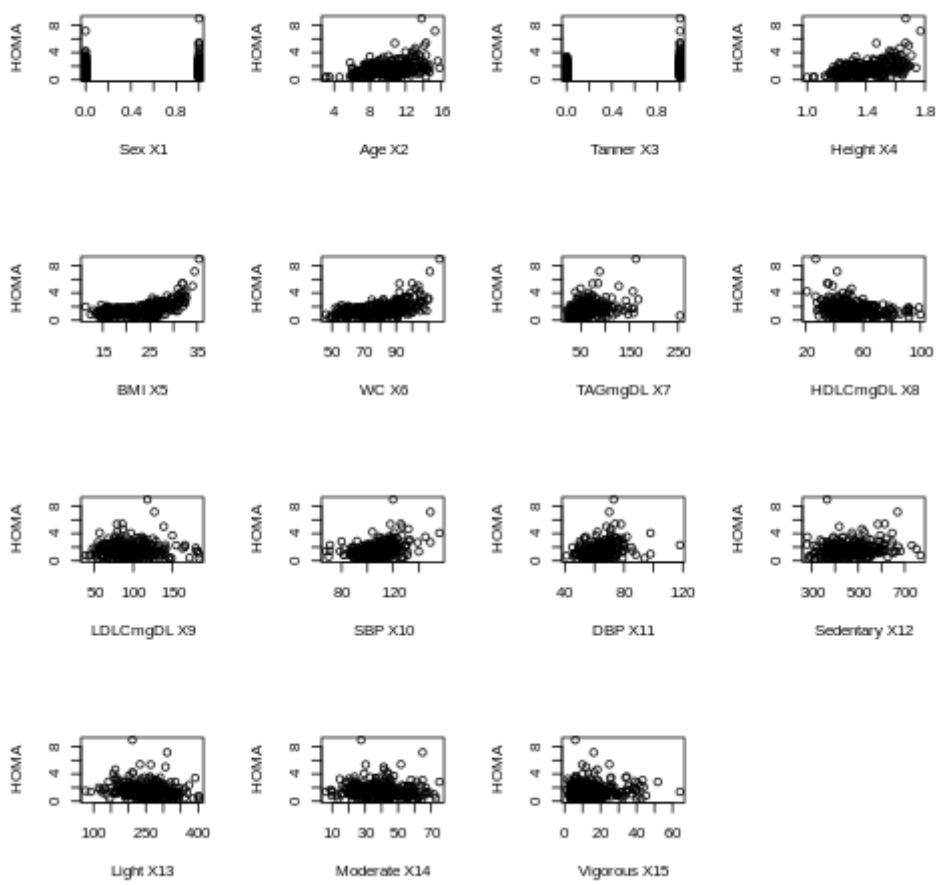
In our case, implementing this procedure on our data set shows that the correlation of SBP with HOMA-IR was higher than for other input variables (e.g., Sex). Nonetheless, at the end of this capsule we will see that, unlike Sex, SBP will eventually be eliminated from the final model. This can be explained by the fact that construction of the final model does not only depend on the individual correlation of each input variable with the output variable but rather, on the contribution of each variable with respect to the rest of the selected variables. If what can be explained by one input variable is already better explained by another, the former should not become part of the final model.

An alternative to the above is to **graphically show the relationship of each input variable with respect to the output variable**, HOMA-IR. Thus, we can visually observe not only whether their relationship is approximately linear, but also the shape of the point cloud. For example, we could see whether a variable exhibits quadratic or logarithmic behavior, and therefore it would be more appropriate to include these terms in the model. In our case study, we will opt for the **second approach** because it is considered more informative. The following R code block iteratively plots, in order, all the input variables with respect to the output variable (HOMA-IR).

NOTE: From here on it is important that you also read the comments included within the code for a better understanding of the process.

In [4]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Display of the variable with respect to HOMA
temp <- data
plotY <- function (x,y) {
        plot(temp[,y]~temp[,x], xlab=paste(names(temp)[x]," X",x,sep=""), ylab=names(te
mp)[y])
}
par(mfrow=c(4,4)) #If 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
```

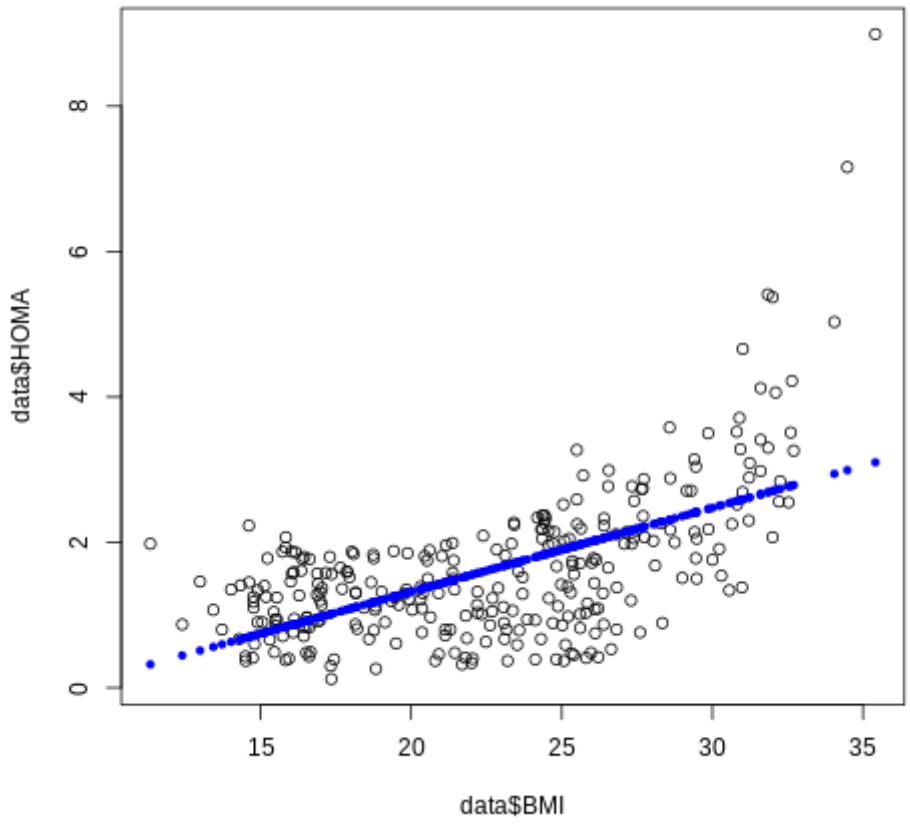


As a result, we can see how, despite showing some scatter in the data, the variables BMI, WC, and $Height$ appear to be the most promising given that they show a relatively linear relationship with $HOMA$. This scatter is a sign that there is no single explanatory factor for the value of insulin resistance ($HOMA - IR$). In addition, all three variables show some non-linearity, a behavior that was most noticeable for BMI, which seems to show a quadratic relationship to some extent. In the following code blocks we will focus on these three variables and apply a simple linear regression to each of them as our first line of analysis. These code blocks launch a simple linear regression between $HOMA$ and BMI, $Height$, or WC, respectively.

In [5]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Obtaining the model. Function lm() from ISLR package.
### Y=HOMA, X's=BMI (body mass index) -> formula: HOMA ~ BMI
fitLM <- lm(HOMA ~ BMI, data=data)

### Line display (blue, estimated values) vs actual values (black, observed values).
yprime = predict(fitLM,data)
plot(data$HOMA~data$BMI)
points(data$BMI,yprime,col="blue",pch=20)

### Coefficients (Estimate), p-values (Pr(>|t|)), Adjusted R2 (Adjusted R-squared),
### F-statistic and p-value (F-statistic and p-value)
summary(fitLM)
```

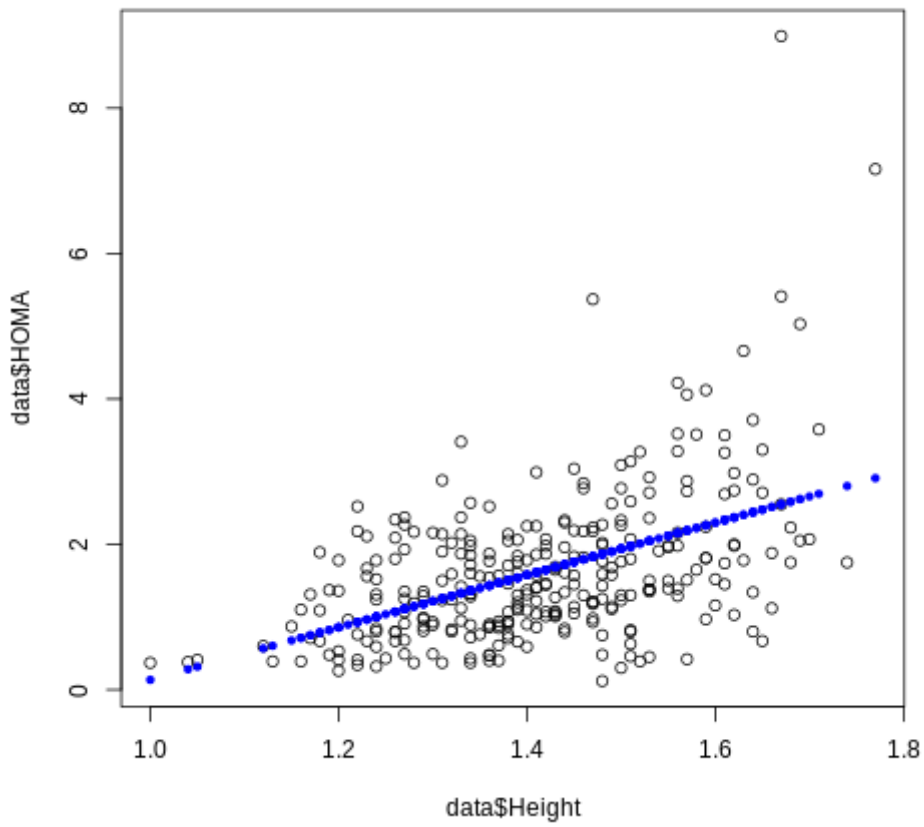
```
Call:
lm(formula = HOMA ~ BMI, data = data)

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]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Idem for the Height variable
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)
```

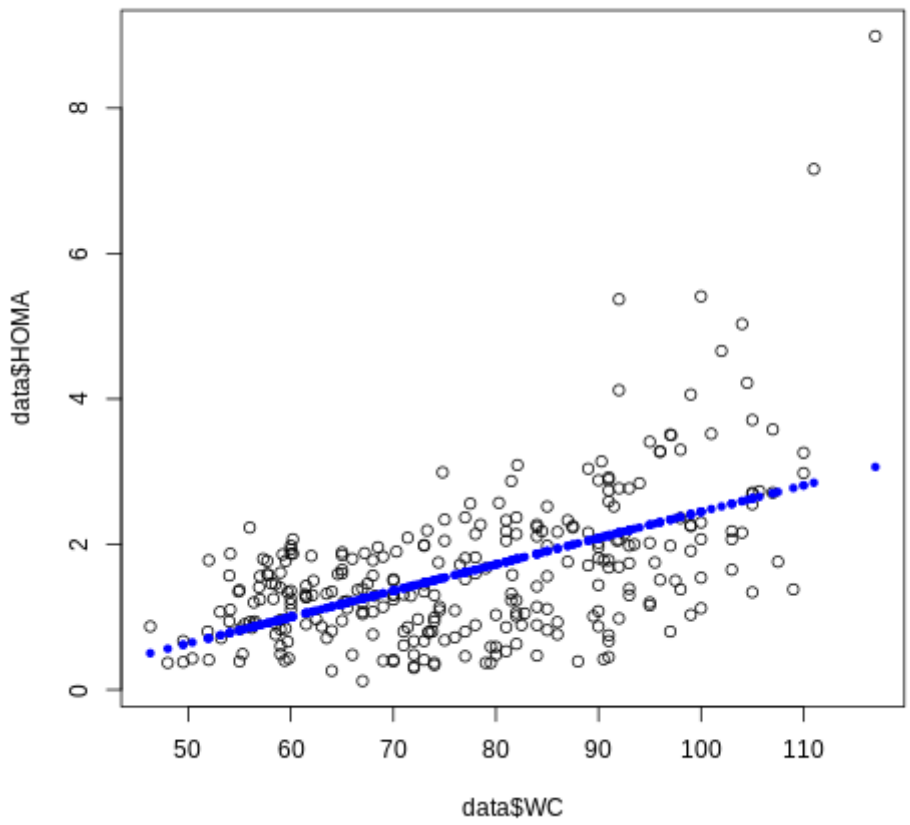
```
Call:
lm(formula = HOMA ~ Height, data = data)

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]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Idem for the variable WC (waist circumference)
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
```



As a result, we can observe that the **p-values** associated with the **coefficients** of the three variables (column with name "$Pr(>|t|)$") clearly indicate that all **three are related to insulin resistance** (as they acquire p-values well below 0.1). Although it is the regression model based on (BMI) that explains more variability in $HOMA$ (according to its adjusted R^2 **value**), it is true that the value is not very high (0.3458).

Next, we will repeat the variable selection process but this time implementing the aforementioned **top-down approach**, which is applicable here because we only have 15 input variables.

4. STEPWISE VARIABLE SELECTION: BACKWARD STEPWISE REGRESSION

In this section, we are going to leave behind **simple linear regressions** and move on to consider **multiple regression** models. As already indicated, we will follow a **descending variable selection** approach. The steps are shown one-by-one in the following code blocks so that the decisions made at each point in time can be tracked.

As explained in the previous sections, variables selected by including all them inthe model by applying a **backward approach**. This is achieved in *R* by means of the (*Y* ~ .) command, where the dot indicates "*all the available input variables in the dataset*".

In [8]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Obtaining the model. Y=HOMA, X's=All -> formula: HOMA ~ .
fitLM <- lm(HOMA ~ ., data=data)

### Reminder:
### Coefficients (Estimate), p-values (Pr(>|t|)), Adjusted R2 (Adjusted R-squared),
### F-statistic and p-value (F-statistic and 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
```

Once we have obtained the **multiple linear regression** model with all the input variables, it is essential we not overlook the value obtained for the **F statistic**; this should be the first thing we check. As explained at beginning of this capsule, if the value of F is close to 1 and/or its accompanying p-value exceeds 0.1 or 0.15, none of the variables used present a linear relationship with the output variable (HOMA-IR in our case). This interpretation **will always be independent of the individual p-value** obtained for each coefficient, which could mislead us until we eliminate redundant or uninformative input variables; in this situation, we would stop the linear regression analysis and look for an alternative regression technique. However, as we already knew from the initial values obtained in the previous section, this is not the case for our data set.

Checking the **p-values** obtained after this first step, we can see that the next step would be to eliminate the Age variable because it has the highest p-value, equal to 0.89656. An interesting detail is that the adjusted R^2 of the complete model improves with respect to the R^2 obtained for the simple linear regression models of the previous section (reaching a value of 0.4876). To eliminate the Age variable from the complete model in R, we use the command with the subtraction sign "-" in the formula, thus obtaining the new model.

In [9]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Obtaining the model. Y=HOMA, X's=All-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
```

After removing Age, we can see how the new **adjusted** R^2 of the model improves because we removed a variable that did not contribute anything to the model. In view of these results, the next step will be to eliminate the $Moderate$ physical activity variable.

In [10]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Same as above -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
```

As non-informative variables are removed, we can see how the **adjusted** R^2 **value** of the resulting model continues to incrementally improve. Next, we will eliminate the $Vigorous$ variable which refers to the daily number of minutes of $vigorous physical activity the children had engaged in.

At this point it is clear how to go about the procedure of eliminating non-informative variables one-by-one. This process should always be done in this way, even if it is tedious. In the following section we will show the steps we must take next to reach the last step (final model), which we will then run to see the final result. Please also remember to carefully read the comments in the following code

In [11]:

```
# Estimated execution time: approx. 3 seconds.

%%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)

### In the above model already all p-values could be considered correct.
### For simplicity we have continued to remove while the adjusted R2 has hardly been af
fected.
#fitLM <- lm(HOMA ~ .-Age-Moderate-Vigorous-SBP-Tanner-Light, data=data)
#summary(fitLM)
#fitLM <- lm(HOMA ~ .-Age-Moderate-Vigorous-SBP-Tanner-Light-Sedentary, data=data)
#summary(fitLM)

### From here R2 would start to get significantly worse.
### We stop and reformulate for readability by indicating the selected input variables
 in an additive way.
### This model is equivalent to the one immediately above but shows clearly what is sel
ected
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+DBP, data=data) #See tha
t the dot is no longer included
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
```

As a result, we obtain a model with 8 input variables and an **adjusted** R^2 of 0.4861.

5. INTERACTIONS AND NON-LINEARITY

Once we have selected the input variables to be incorporated into our **linear model**, we will try to explain the **non-linear** part of the data by adding **interactions and other non-linear terms**. To assess these interactions, we rely on our prior knowledge of the problem. For example, in a case of a known genetic interaction between two genetic variants, it would be appropriate to introduce an interaction term between the two to model their effect on the output variable. Where there is no prior information on any interaction phenomena, we can also be guided by logic or intuition, depending on the meaning of the input variables. If we still cannot find any possible interactions, we can randomly test the variables shown to be most significant (trial-and-error). However, this procedure is not a trivial and depends on our own skill and experience.

In our case study on childhood obesity, we will evaluate whether there is positive synergy (multiplicative factors, $*$) between the triglyceride variable and the two cholesterol metrics (because they all belong to the lipid profile).

In [12]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Interactions between triglycerides and cholesterols
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+DBP+TAGmgDL*HDLCmgDL*LDL
CmgDL, 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
```

Note that the use of operators already includes all the hierarchy terms. If it did not, we would have to add them by hand before looking at any p-values or making any decisions. We can see that the term *TAGmgDL:LDLCmgDL:HDLCmgDL* has an exceptionally bad p-value (0.961424), indicating that this hypothesized interaction was not pertinent.

We will now retest the model with the height and waist circumference values.

In [13]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Interactions between height and waist circumference
fitLM <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+HDLCmgDL+DBP+Height*WC, data=dat
a)
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
```

In this case we can see how the interaction explains part of the **non-linearity** given that a p-value of less than 0.1 was obtained. Therefore, in principle we will retain this as part of the model.

Finally, we must check for other **non-linear** terms. In this case, since we initially plotted all the input variables with respect to the output variable $HOMA-IR$, it is much easier to determine certain types of non-linear behavior visually. Of note, we saw that $HOMA-IR$ appeared to have a quadratic relationship with BMI. Therefore, we will also try to include this term in the model. This can be achieved using the $I(.)$ function in R, denoting the power as follows $I(X_j \hat{} exponent)$, in our case $I(BMI \hat{} 2)$. The $I(.)$ function does not automatically generate the hierarchy terms and so, before we can even look at the model, we must make sure that the formula contains all the hierarchy terms. In our case, the hierarchy terms would be $BMI + BMI^2 + Height + WC + Height * WC$.

In [14]:

```
# Estimated execution time: approx. 3 seconds.

%%R
### Interactions between height and waist circumference, plus 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
```

Again, terms with high p-values appear, as is the case for $Height * WC$. The interaction was trying to explain what the quadratic BMI term now explains better. We will therefore remove the now non-significant interaction. Recall that the BMI variable was already clearly showing a non-linear quadratic appearance in the graph.

In [15]:

```
# Estimated execution time: approx. 3 seconds.

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

According to the new model, DBP would also now have to be eliminated.

In [16]:

```
# Estimated execution time: approx. 3 seconds.

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

And finally $HDLCmgDL$ must also be removed.

In an additional line of code, we have also included how to calculate the **RMSE** metric for a linear regression model (which did not appear in the *output* of the results offered by the *summary* command).

In [17]:

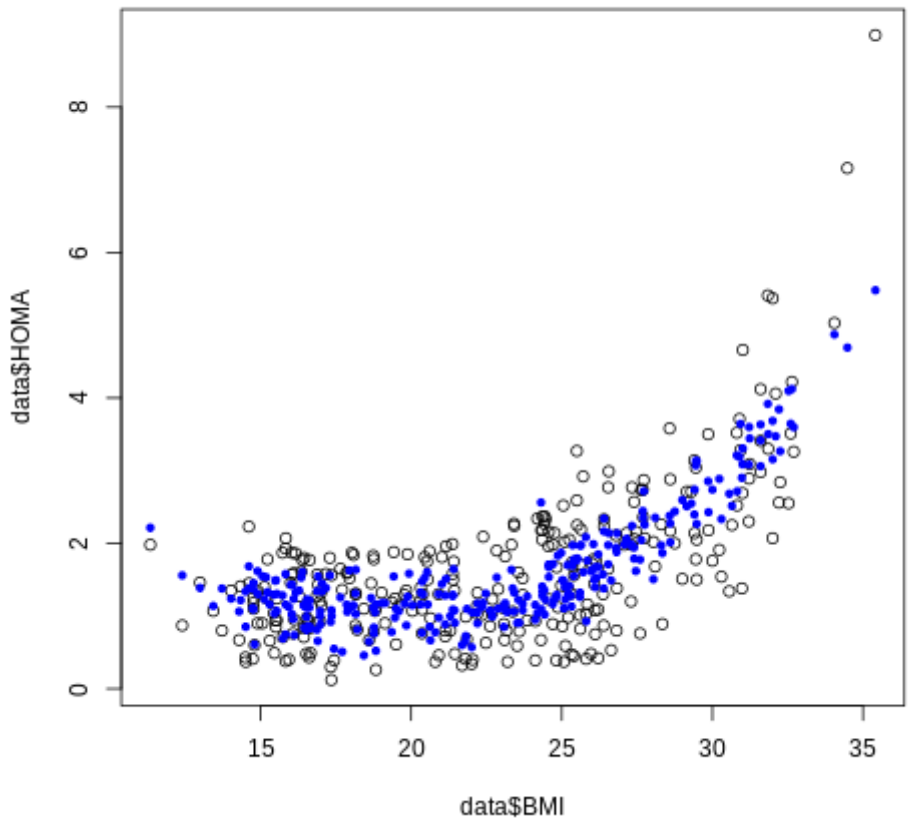
```
# Estimated execution time: approx. 3 seconds.

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

### RECM calculation
yprime = predict(fitLM,data)
cat('\nRMSE:', sqrt(sum((data$HOMA-yprime)^2)/length(yprime)), "\n") #RECM->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
```

Visualization:

In [18]:

```
#yprime = predict(fitLM,data)

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



Finally, we have arrived at **adjusted** R^2 of 0.6092 when we had started at 0.3458. But the most important thing is not the value itself but rather, what we were able to learn about the data and our problem based on statistical values. The main conclusion we can draw is that a high body mass index in children is one of the main risk factors for insulin resistance. Even if the conclusion is that we must rethink the problem with new variables and measurements, coming to this realization would still represent a big step.

6. CROSS- VALIDATION

Once we have obtained the best formula for applying the parametric fit (regression model), if we want to estimate new values of the output variable and compare its predictive ability with other models, as explained in Module 3 (*Data science and machine learning*), we must apply a cross-validation. In the following code, we show how this can be achieved.

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){
  # Training set is created for iteration
  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]]

  ## Training the model for iteration
  model <- lm(HOMA ~ BMI+Height+TAGmgDL+Sex+WC+LDLCmgDL+I(BMI^2), data=training_set)

  ## Calculating test error
  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
}

# Remove the artificial column added for kfold
#(so that it doesn't accumulate columns if it is executed several times)
nombres <- names(data)
tam <- length(nombres)-1
data <- data[,nombres[1: tam]]

cat("mean RMSE  in test for 5-fcv:", mean(performances))
```

```
mean RMSE  in test for 5-fcv: 0.6956061
```

REFERENCES

- 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)

ADDITIONAL REFERENCES

- 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)