
MACHINE LEARNING EN RSTUDIO REGRESION LINEAL

EDUARD LARA

1. INDICE

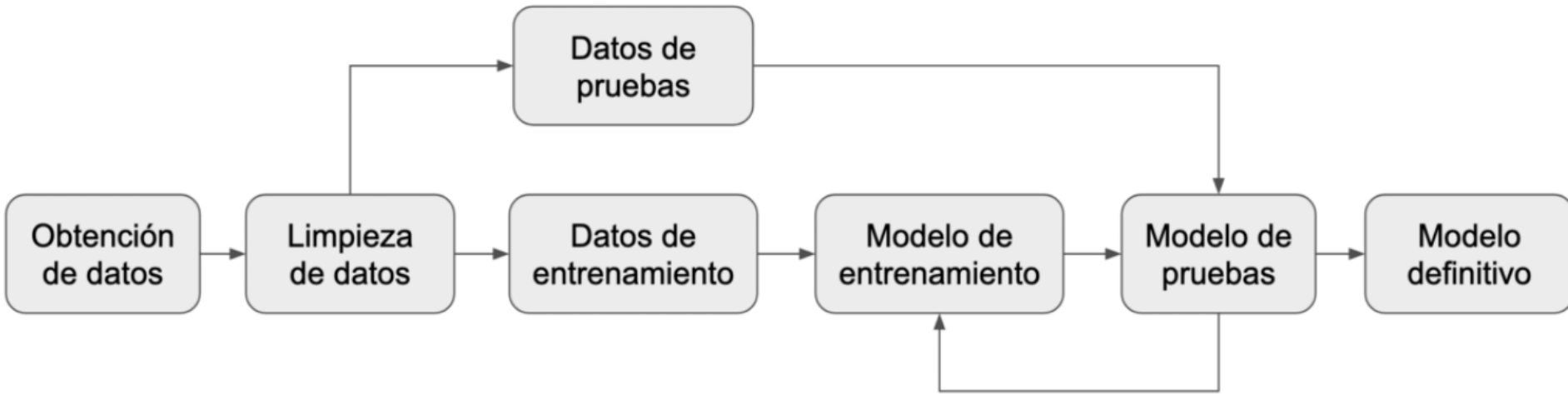
1. Introducción a machine learning
2. Tipos de algoritmos de machine learning
3. Ejemplo regresión lineal parte I
4. Ejemplo regresión lineal parte II

1. INTRODUCCION A MACHINE LEARNING

- El Machine Learning o aprendizaje de máquinas o aprendizaje automático es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial
- El objetivo del Machine Learning es desarrollar técnicas que permitan que las computadoras aprendan
- Machine Learning es un método de análisis de datos que automatiza la construcción de un modelo analítico.
- Permite a los ordenadores encontrar soluciones a problemas, sin ser explícitamente programados para ello, gracias al uso de algoritmos, que aprenden de los datos.

1. INTRODUCCION A MACHINE LEARNING

Diagrama típico de machine learning



1. INTRODUCCION A MACHINE LEARNING

- Primero hay una fase de obtención de datos
- Luego la fase de limpieza de los datos.
- A su vez se dividen en datos de entrenamiento y datos de pruebas.
- Estos datos de entrenamiento se utilizan para entrenar el modelo
- Una vez que están entrenados, se utilizan los datos de prueba para validarlo.
- En el caso de que sea correcto, se pasa al modelo definitivo
- Si no, se vuelve para atrás el modelo de entrenamiento

2. TIPOS ALGORITMOS MACHINE LEARNING

Tenemos tres tipos de algoritmos

1. **Aprendizaje supervisado:** Este algoritmo necesita datos previamente etiquetados (solucionados), para aprender a realizar el trabajo (información de lo que es correcto y lo que no). En base a esos datos, el algoritmo es capaz de aprender a resolver problemas futuros similares, identificando si es correcto o si es de un tipo o de otro. El aprendizaje supervisado se diferencia entre clasificación y regresión



2. TIPOS ALGORITMOS MACHINE LEARNING

- 2) **Aprendizaje no supervisado:** Este algoritmo necesita indicaciones previas que le enseñen a comprender y analizar la información para resolver problemas futuros similares. No necesita datos previamente etiquetados que indiquen que es lo correcto y lo que no es correcto. Simplemente les damos información para que comprenda y analice por su cuenta la información. Dentro del aprendizaje no supervisado estaría el clustering y estarían las reglas de asociación.

Aprendizaje No
Supervisado

Clustering

Reglas de
Asociación

El aprendizaje no supervisado nos ayuda a encontrar **patrones ocultos o relaciones ocultas en los datos.**

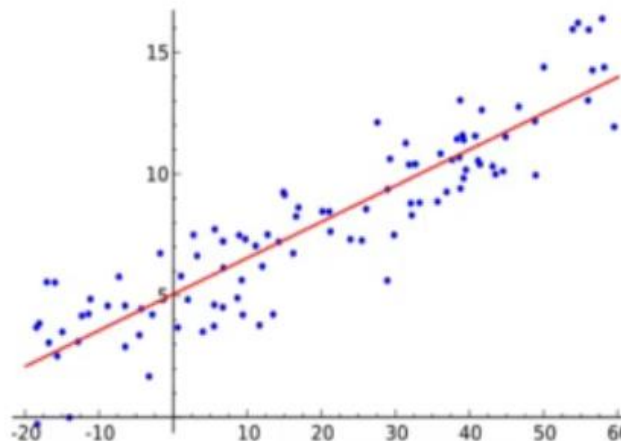
Se aplica sobre conjuntos de **datos no etiquetados** y no requiere conjunto de entrenamiento.

2. TIPOS ALGORITMOS MACHINE LEARNING

- 3) **Aprendizaje de refuerzo:** Este algoritmo aprende por su cuenta, en base a unos conocimientos previos introducidos y a la práctica que realiza sobre los problemas, aprendiendo en función del éxito o fracaso que obtiene al resolver los problemas

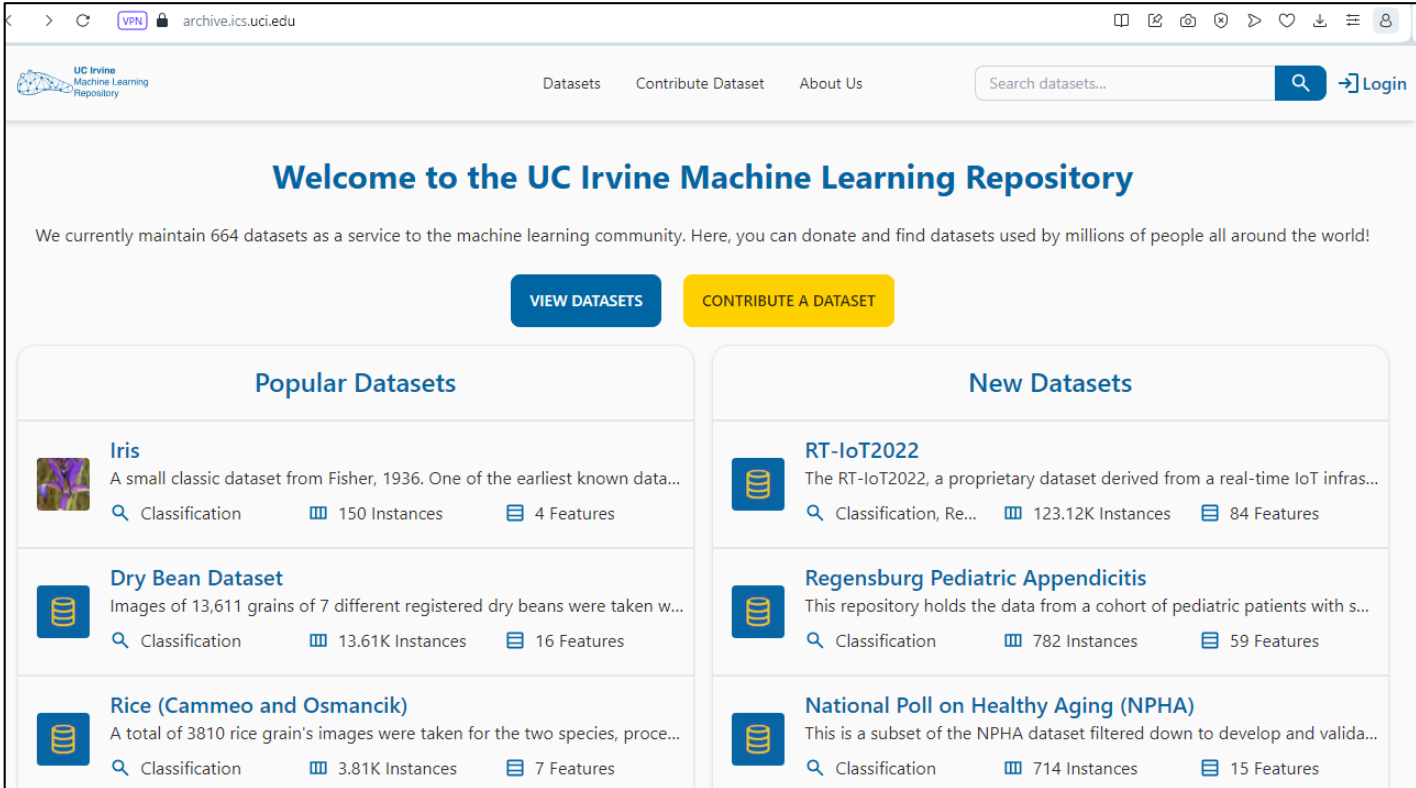
3. INTRODUCCION REGRESION LINEAL

- La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza en M. Learning y estadística.
- La regresión lineal es una aproximación para modelar la relación entre una variable escalar dependiente "y" y una o más variables explicativas x.
- La idea es dibujar una recta (de color rojo en el diagrama), que indica la tendencia del conjunto de datos, para predecir en función de un valor X el valor de Y.



3. EJEMPLO REGRESION LINEAL

Paso 1. Vamos a realizar un ejemplo de regresión lineal. Buscaremos un dataset para poder hacer este ejemplo, en la pagina web <https://archive.ics.uci.edu>, la cual contiene un montón de repositorios para Machine Learning.



The screenshot shows the UC Irvine Machine Learning Repository website. The header includes the site logo, navigation links for 'Datasets', 'Contribute Dataset', and 'About Us', a search bar, and a 'Login' button. The main content area welcomes visitors and states that 664 datasets are available. Two prominent buttons, 'VIEW DATASETS' and 'CONTRIBUTE A DATASET', are displayed. Below these, the page is divided into two columns: 'Popular Datasets' and 'New Datasets'. The 'Popular Datasets' column lists 'Iris', 'Dry Bean Dataset', and 'Rice (Cammeo and Osmancik)'. The 'New Datasets' column lists 'RT-IoT2022', 'Regensburg Pediatric Appendicitis', and 'National Poll on Healthy Aging (NPHA)'. Each dataset entry includes a small icon, the dataset name, a brief description, and metadata such as the number of instances and features.

Dataset Name	Description	Instances	Features
Iris	A small classic dataset from Fisher, 1936. One of the earliest known data...	150	4
Dry Bean Dataset	Images of 13,611 grains of 7 different registered dry beans were taken w...	13.61K	16
Rice (Cammeo and Osmancik)	A total of 3810 rice grain's images were taken for the two species, proce...	3.81K	7
RT-IoT2022	The RT-IoT2022, a proprietary dataset derived from a real-time IoT infras...	123.12K	84
Regensburg Pediatric Appendicitis	This repository holds the data from a cohort of pediatric patients with s...	782	59
National Poll on Healthy Aging (NPHA)	This is a subset of the NPHA dataset filtered down to develop and valida...	714	15

3. EJEMPLO REGRESION LINEAL

Paso 2. Buscaremos un dataset llamado Student Performance y lo descargamos

Datasets

Contribute Dataset

About Us

student performance

Search

Login

Browse Datasets

SORT BY # VIEWS, DESC

EXPAND ALL

Student Performance

Predict student performance in secondary education (high school).

Classification, Regress... Multivariate

649 Instances

33 Features

Higher Education Students Performance Evaluation

The data was collected from the Faculty of Engineering and Facul

Classification Multivariate

Student Performance on an entrance examination

This dataset contains data of the candidates who qualified the m

Classification Multivariate

6

student+performance.zip

Descarga terminada

Student Performance

Donated on 11/26/2014

Predict student performance in secondary education (high school).

Dataset Characteristics

Multivariate

Subject Area

Social Science

Associated Tasks

Classification, Regression

Feature Type

Integer

Instances

649

Features

30

Dataset Information

Additional Information

This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). I...

SHOW MORE

Has Missing Values?

No

DOWNLOAD

IMPORT IN PYTHON

CITE

6 citations

117600 views

Creators

Paulo Cortez

DOI

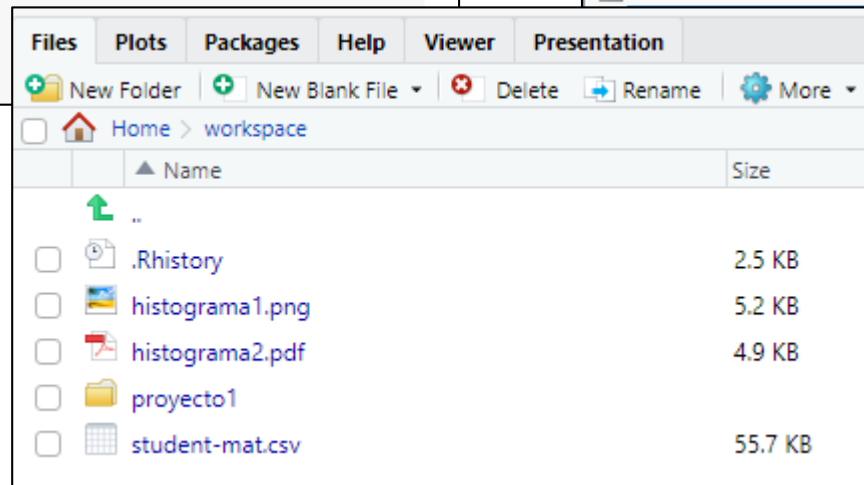
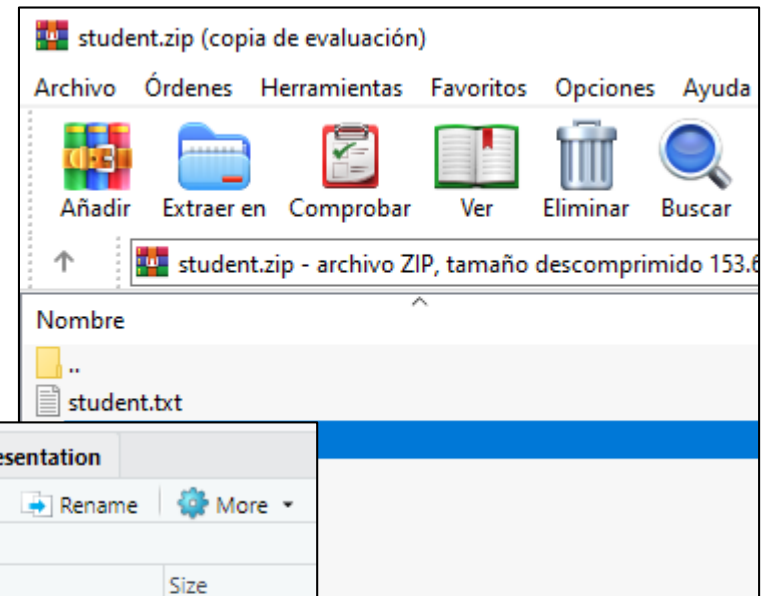
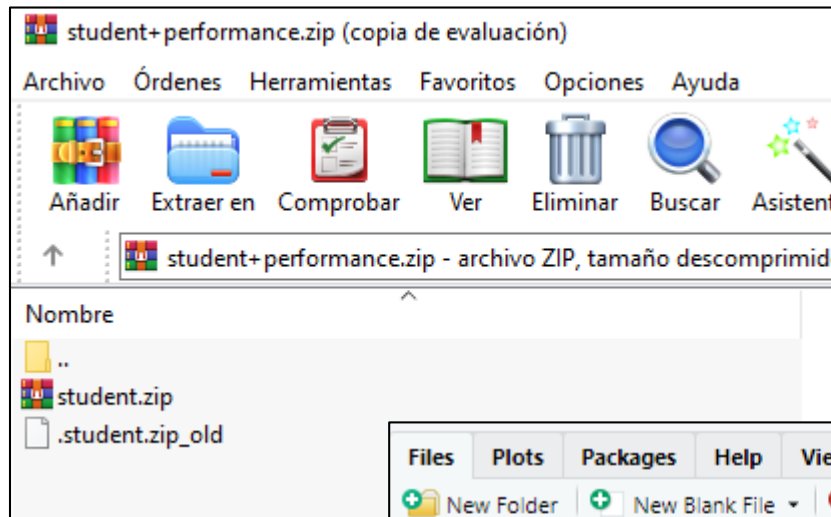
10.24432/CSTG7T

License

This dataset is licensed under a [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](#) license.

3. EJEMPLO REGRESION LINEAL

Paso 3. Descomprimos student+performance.zip, sacamos student.zip y de aquí sacamos student-mat.zip. Copiamos este ultimo archivo en su directorio workspace



3. EJEMPLO REGRESION LINEAL

Paso 4. Cargamos este fichero csv desde la consola de Rstudio, cuyos campos están separados por un punto y coma. Este dataset tiene 395 observaciones y 33 variables o apéndices columnas

```
> datos = read.csv('student-mat.csv', sep=';')
```

Data	
• datos	395 obs. of 33 variables

3. EJEMPLO REGRESION LINEAL

Paso 5. Mostramos las 6 primeras filas. Las columnas muestran el colegio, el sexo, la edad, el trabajo de la madre, el trabajo del padre, las ausencias a clase, y las tres notas G1, G2, G3, que son las que queremos evaluar.

```
> datos = read.csv('student-mat.csv',sep=';')
> head(datos)
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
1	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother
2	GP	F	17	U	GT3	T	1	1	at_home	other	course	father
3	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother
4	GP	F	15	U	GT3	T	4	2	health	services	home	mother
5	GP	F	16	U	GT3	T	3	3	other	other	home	father
6	GP	M	16	U	LE3	T	4	3	services	other	reputation	mother

	traveltime	studytime	failures	schoolsup	famsup	paid	activities	nursery	higher	internet
1	2	2	0	yes	no	no	no	yes	yes	no
2	1	2	0	no	yes	no	no	no	yes	yes
3	1	2	3	yes	no	yes	no	yes	yes	yes
4	1	3	0	no	yes	yes	yes	yes	yes	yes
5	1	2	0	no	yes	yes	no	yes	yes	no
6	1	2	0	no	yes	yes	yes	yes	yes	yes

	romantic	famrel	freetime	goout	dalc	walc	health	absences	G1	G2	G3
1	no	4	3	4	1	1	3	6	5	6	6
2	no	5	3	3	1	1	3	4	5	5	6
3	no	4	3	2	2	3	3	10	7	8	10
4	yes	3	2	2	1	1	5	2	15	14	15
5	no	4	3	2	1	2	5	4	6	10	10
6	no	5	4	2	1	2	5	10	15	15	15

```
> |
```

3. EJEMPLO REGRESION LINEAL

Paso 6. Vamos a hacer una predicción del valor de $G3$ en función de las características de las demás columnas, mediante el modelo de regresión lineal. Primero revisamos que el dataset no tenga algún valor nulo (not available) para que nos funcionen bien las fórmulas, mediante el método `any`

```
> any(is.na(datos))  
[1] FALSE  
> |
```

`is.na(x)`
'Not Available' / Missing Values

NA is a logical constant of length 1 which contains a missing value indicator. NA can be coerced to any other vector type except raw. There are also constants `NA_integer_`, `NA_real_`, `NA_complex_` and `NA_character_` of the other atomic vector types which support missing values: all of these are reserved words in the R language.

The generic function `is.na` indicates which elements are missing.

The generic function `is.na<-` sets elements to NA.

The generic function `anyNA` implements `any(is.na(x))` in a possibly faster way (especially for atomic vectors).

Si es falso, quiere decir que no hay ningún valor que tenga un valor na en ningún sitio, con lo cual es perfecto para hacer los cálculos.

3. EJEMPLO REGRESION LINEAL

Paso 7. Vamos a cargar las librería que vamos a necesitar: ggplot2, ggthemes y dplyr.

install.packages('ggplot2')

install.packages('ggthemes')

install.packages('dplyr')

library(ggplot2)

library(ggthemes)

library(dplyr)

```
> library(ggplot2)
Learn more about the underlying theory at
https://ggplot2-book.org/
> library(ggthemes)
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':
  filter, lag

The following objects are masked from 'package:base':
  intersect, setdiff, setequal, union

>
```


3. EJEMPLO REGRESION LINEAL

Paso 8. Crearemos un dataframe con la correlación entre las variables o las columnas que son numéricas. Del dataset seleccionaremos las columnas que tengan valores numéricas, para ver qué relación guardan entre ellas.

Primero veremos que columnas de los datos del dataset del fichero csv, que son numéricas.

Hacemos un head de la variable columnas.numericas. Las que son FALSE no son numéricas

```
> columnas.numericas= sapply(datos,is.numeric)
>
> head(columnas.numericas)
  school      sex    age address famsize Pstatus
  FALSE    FALSE   TRUE   FALSE    FALSE    FALSE
> |
```

3. EJEMPLO REGRESION LINEAL

Paso 9. En la variable `datos.correlacion` vamos a seleccionar las columnas numéricas.

```
> datos.correlacion = cor(datos[,columnas numericas])  
> print(datos.correlacion)
```

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout
age	1.000000000	-0.163658419	-0.163438069	0.070640721	-0.004140037	0.24366538	0.053940096	0.01643439	0.126963880
Medu	-0.163658419	1.000000000	0.623455112	-0.171639305	0.064944137	-0.23667996	-0.003914458	0.03089087	0.064094438
Fedu	-0.163438069	0.623455112	1.000000000	-0.158194054	-0.009174639	-0.25040844	-0.001369727	-0.01284553	0.043104668
traveltime	0.070640721	-0.171639305	-0.158194054	1.000000000	-0.100909119	0.09223875	-0.016807986	-0.01702494	0.028539674
studytime	-0.004140037	0.064944137	-0.009174639	-0.100909119	1.000000000	-0.17356303	0.039730704	-0.14319841	-0.063903675
failures	0.243665377	-0.236679963	-0.250408444	0.092238746	-0.173563031	1.000000000	-0.044336626	0.09198747	0.124560922
famrel	0.053940096	-0.003914458	-0.001369727	-0.016807986	0.039730704	-0.04433663	1.000000000	0.15070144	0.064568411
freetime	0.016434389	0.030890867	-0.012845528	-0.017024944	-0.143198407	0.09198747	0.150701444	1.000000000	0.285018715
goout	0.126963880	0.064094438	0.043104668	0.028539674	-0.063903675	0.12456092	0.064568411	0.28501871	1.000000000
Dalc	0.131124605	0.019834099	0.002386429	0.138325309	-0.196019263	0.13604693	-0.077594357	0.20900085	0.266993848
walc	0.117276052	-0.047123460	-0.012631018	0.134115752	-0.253784731	0.14196203	-0.113397308	0.14782181	0.420385745
health	-0.062187369	-0.046877829	0.014741537	0.007500606	-0.075615863	0.06582728	0.094055728	0.07573336	-0.009577254
absences	0.175230079	0.100284818	0.024472887	-0.012943775	-0.062700175	0.06372583	-0.044354095	-0.05807792	0.044302220
G1	-0.064081497	0.205340997	0.190269936	-0.093039992	0.160611915	-0.35471761	0.022168316	0.01261293	-0.149103967
G2	-0.143474049	0.215527168	0.164893393	-0.153197963	0.135879999	-0.35589563	-0.018281347	-0.01377714	-0.162250034
G3	-0.161579438	0.217147496	0.152456939	-0.117142053	0.097819690	-0.36041494	0.051363429	0.01130724	-0.132791474
	Dalc	walc	health	absences	G1	G2	G3		
age	0.131124605	0.11727605	-0.062187369	0.17523008	-0.06408150	-0.14347405	-0.16157944		
Medu	0.019834099	-0.04712346	-0.046877829	0.10028482	0.20534100	0.21552717	0.21714750		
Fedu	0.002386429	-0.01263102	0.014741537	0.02447289	0.19026994	0.16489339	0.15245694		
traveltime	0.138325309	0.13411575	0.007500606	-0.01294378	-0.09303999	-0.15319796	-0.11714205		
studytime	-0.196019263	-0.25378473	-0.075615863	-0.06270018	0.16061192	0.13588000	0.09781969		

3. EJEMPLO REGRESION LINEAL

Comentarios:

En datos.correlacion vemos los valores de las columnas numéricas, y la relación de unas con otras.

Por ejemplo, la relación de la edad con la edad es 1 porque la relación es completa, es la misma columna.

En cambio la relación entre la educación de la madre con la edad o la relación entre la educación del padre y la edad, como son negativos, es muy pequeña.

Los valores numéricos son difíciles de interpretar, así que haremos un gráfico para que se vea mejor.

3. EJEMPLO REGRESION LINEAL

Paso 10. Para hacer el grafico necesitamos instalar los paquetes **corrgram** y **corrplot**. Una vez instalados, los cargaremos con library.

```
> install.packages(corrgram)
Error in install.packages : objeto 'corrgram' no encontrado
> install.packages('corrgram')
Installing package into 'C:/Users/eduar/AppData/Local/R/win-library/4.3'
(as 'lib' is unspecified)
probando la URL 'https://cran.rstudio.com/bin/windows/contrib/4.3/corrgram_1.14.zip'
Content type 'application/zip' length 403290 bytes (393 KB)
downloaded 393 KB

package 'corrgram' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\eduar\AppData\Local\Temp\RtmpwMqiq5\downloaded_packages
> install.packages('corrplot')
Installing package into 'C:/Users/eduar/AppData/Local/R/win-library/4.3'
(as 'lib' is unspecified)
probando la URL 'https://cran.rstudio.com/bin/windows/contrib/4.3/corrplot_0.92.zip'
Content type 'application/zip' length 3844924 bytes (3.7 MB)
downloaded 3.7 MB

package 'corrplot' successfully unpacked and MD5 sums checked

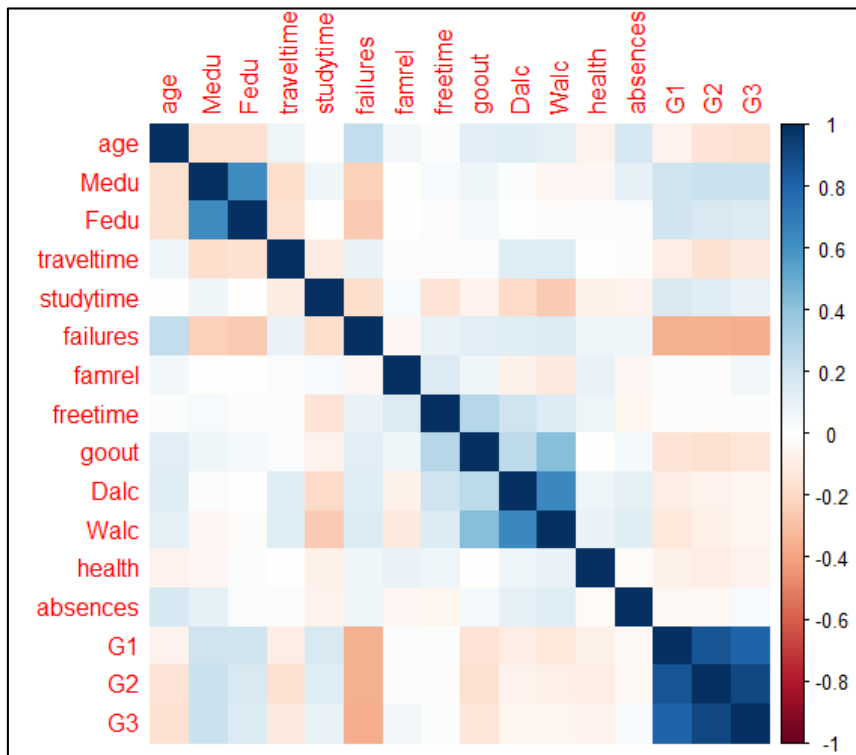
The downloaded binary packages are in
  C:\Users\eduar\AppData\Local\Temp\RtmpwMqiq5\downloaded_packages
> library(corrplot)
corrplot 0.92 loaded
> library(corrgram)
> |
```

3. EJEMPLO REGRESION LINEAL

Paso 11. Para el gráfico, utilizaremos la función `corrplot`:

```
> grafico = corrplot(datos.correlacion, method='color')  
>
```

Le pasamos los datos de correlación (con las columnas numéricas) y el `method` para que sea un gráfico de color.



En la pestaña Plots, se crea un gráfico de la relación entre las variables que son numéricas

3. EJEMPLO REGRESION LINEAL

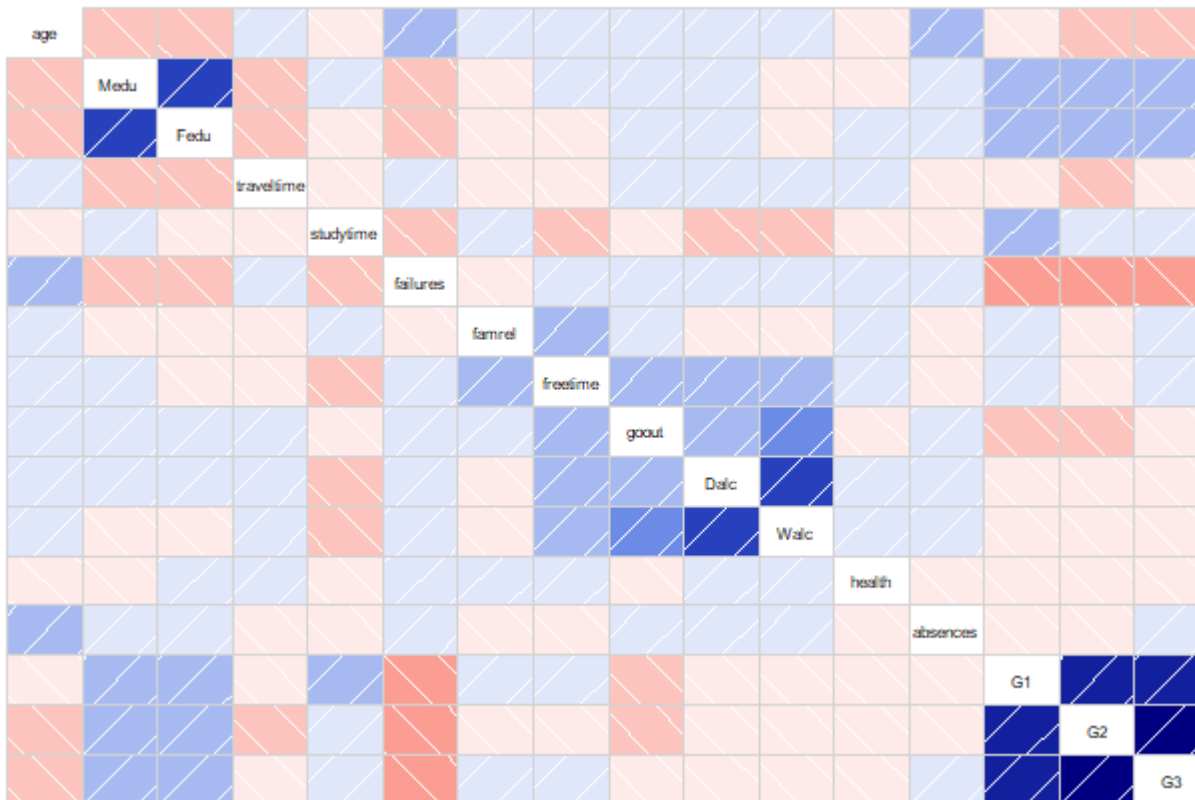
Vemos las diferentes relaciones entre las variables numéricas:

- 1) El color oscuro significa que la relación es muy alta. El 1 significa que es la misma columna, son columnas iguales
- 2) La correlación entre las notas es alta. Hay una dependencia grande entre las notas de $G1$, $G2$ y $G3$.
- 3) La educación de la madre también tiene bastante relación con la del padre madre.

3. EJEMPLO REGRESION LINEAL

Paso 12. Para hacer una vista de las relaciones entre todas las columnas, numéricas o no, usaremos el gráfico corrgram.

```
> corrgram(datos)  
>
```



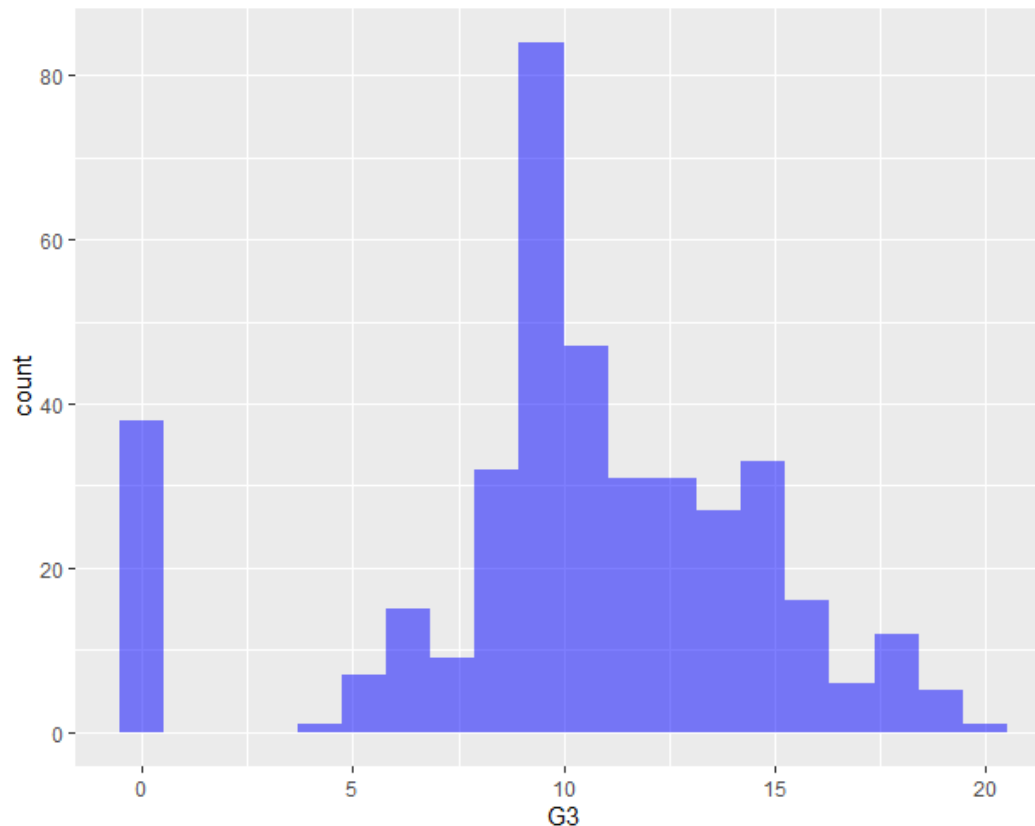
Se genera otro gráfico muy similar al anterior, solo que en este caso están todas las columnas

Sigue habiendo la relación de las notas y también hay una relación bastante fuerte entre las variables en la parte central.

3. EJEMPLO REGRESION LINEAL

Paso 13. Haremos un histograma de la nota G3 para hacernos una idea de cómo va a ir. Usaremos ggplot.

```
> ggplot(datos, aes(x=G3)) + geom_histogram(bins=20, alpha=0.5, fill='blue')  
> |
```



La nota va del 0 al 20. Si 10 fuera el aprobado, hay una mayor cantidad de aprobados que de suspensos. La nota cerca del aprobado es donde hay la mayor frecuencia. La gente que no se ha presentado tiene un cero

4. EJEMPLO REGRESION LINEAL II

Paso 1. Vamos a dividir los datos en entrenamiento y pruebas: un 70% para entrenar el modelo y otro 30% para hacer las pruebas o para hacer las predicciones.

Primero de todo necesitamos instalar el paquete **caTools** y cargarlo en memoria con library

```
> install.packages('caTools')
Installing package into 'C:/Users/eduar/AppData/Local/R/win-library/4.3'
(as 'lib' is unspecified)
also installing the dependency 'bitops'

probando la URL 'https://cran.rstudio.com/bin/windows/contrib/4.3/bitops_1.0-7.zip'
Content type 'application/zip' length 31813 bytes (31 KB)
downloaded 31 KB

probando la URL 'https://cran.rstudio.com/bin/windows/contrib/4.3/caTools_1.18.2.zip'
Content type 'application/zip' length 245776 bytes (240 KB)
downloaded 240 KB

package 'bitops' successfully unpacked and MD5 sums checked
package 'caTools' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
      C:\Users\eduar\AppData\Local\Temp\RtmpWMqiq5\downloaded_packages
> library(caTools)
> |
```

4. EJEMPLO REGRESION LINEAL II

Paso 2. Primero fijamos una semilla de aleatoriedad a 80. Ayuda a crear los mismos números aleatorios (generar la misma respuesta) cada vez que se llama a una función pseudo-aleatoria. De esta manera los valores aleatorios generados a la hora de descomponer los datos de entrenamiento y de pruebas serán siempre los mismos.

```
> set.seed(80)  
> ejemplo = sample.split(datos$G3, SplitRatio = 0.7)
```

En la variable ejemplo indicamos que vamos a dividir los datos de la columna G3 y va a coger el 70% para entrenamiento y el resto para test

4. EJEMPLO REGRESION LINEAL II

Paso 3. La variable entrenamiento representan los datos de entrenamiento, un subconjunto de datos donde la variable ejemplo de antes es igual a TRUE, indicando que contienen el 70% de los datos para entrenamiento

En la variable pruebas están los datos de pruebas, que será también un subconjunto de los datos donde la variable ejemplo, es igual a FALSE.

```
> entrenamiento = subset(datos, ejemplo==TRUE)
> pruebas = subset(datos, ejemplo==FALSE)
> |
```

4. EJEMPLO REGRESION LINEAL II

Paso 4. Ahora construiremos el modelo de regresión lineal para hacer las predicciones.

```
> modelo = lm(G3 ~. , entrenamiento)
>
> print(summary(modelo))
```

Si hacemos un print del summary podemos revisar el modelo que se ha creado.

4. EJEMPLO REGRESION LINEAL II

Paso 5. Revisamos el modelo creado.

```
call:
```

```
lm(formula = G3 ~ ., data = entrenamiento)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-7.3169	-0.4283	0.2962	1.0233	4.2819

Vemos que se ha utilizado la columna *G3* para hacer la estimación y el punto indica que se han cogido todas las columnas del dataset para hacer la estimación de *G3*. Si sólo queremos poner algunas columnas, se pondrían aquí

Ha cogido los datos de entrenamiento

Los residuals son la diferencia entre los valores actuales de las notas y la línea de regresión que vamos a calcular, que son las predicciones. Indica valores mínimo y máximo²⁹

4. EJEMPLO REGRESION LINEAL II

Paso 6. Vemos también los coeficientes. Aquí están todas las columnas de nuestro dataset

Coefficients:				
(Intercept)	Estimate	Std. Error	t value	Pr(> t)
schoolMS	-2.02105	2.61550	-0.773	0.44046
sexM	0.87187	0.48346	1.803	0.07261
age	0.13046	0.28993	0.450	0.65315
addressU	-0.10444	0.13075	-0.799	0.42524
famsizeLE3	0.14813	0.33742	0.439	0.66106
PstatusT	-0.11979	0.29004	-0.413	0.67998
Medu	-0.48749	0.41538	-1.174	0.24175
Fedu	0.15320	0.18461	0.830	0.40748
	-0.22236	0.15602	-1.425	0.15542

romanticyes	-0.32316	0.26290	-1.229	0.22022
famrel	0.42908	0.13868	3.094	0.00221 **
freetime	0.10566	0.13359	0.791	0.42978
goout	-0.18591	0.13105	-1.419	0.15734
Dalc	-0.34192	0.18199	-1.879	0.06151 .
walc	0.29994	0.13488	2.224	0.02712 *
health	0.04568	0.09273	0.493	0.62277
absences	0.03854	0.01642	2.347	0.01975 *
G1	0.22333	0.07522	2.969	0.00330 **
G2	0.92529	0.06317	14.648	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				
Residual standard error: 1.873 on 235 degrees of freedom				
Multiple R-squared: 0.8593, Adjusted R-squared: 0.8348				
F-statistic: 35.02 on 41 and 235 DF, p-value: < 2.2e-16				

- La columna Estimate es el coeficiente de cada variable según la estimación que le ha dado el modelo.
- La columna error sería la medida de la variabilidad en la estimación del coeficiente en la posibilidad de un error.

4. EJEMPLO REGRESION LINEAL II

- La columna `t_value` representa la puntuación para saber si un coeficiente es significativo o no para el modelo
- La columna probabilidad, indica la probabilidad de que una variable no sea relevante para el modelo. Las columnas que tienen un valor mas pequeño son en las que más se va a basar el modelo para calcular la predicción sobre la columna `G3` → Son columnas que tienen varias estrellas: las dos notas `G1` y `G2`, las ausencias, etc
- El Adjusted square representa una métrica que evalúa lo bien que se ajusta el modelo a estos datos. Si se acerca a 1 es que el modelo es bueno. → 0.83 indica que el modelo se ajusta bastante bien a los datos, por tanto es un modelo bueno para hacer la predicción de esta nota

4. EJEMPLO REGRESION LINEAL II

Paso 7. Vamos a visualizar en una grafica los residuos, que es la diferencia entre el valor real y estimado, o el error cometido con nuestras estimaciones futuras.

```
> residuos = residuals(modelo)
> class(residuos)
[1] "numeric"
> |
```

Creamos la variable residuos mediante la función residuals y le pasamos el modelo que hemos creado antes.

La clase de residuos es numérica y tenemos que pasarla a un dataframe

4. EJEMPLO REGRESION LINEAL II

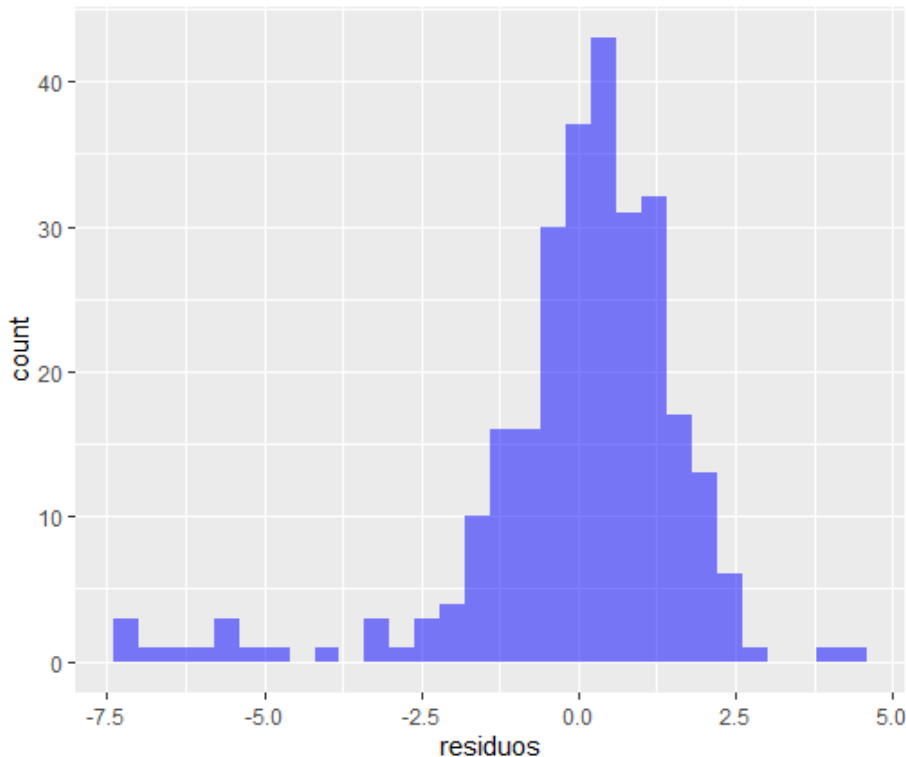
Paso 8. Convertimos residuos en dataframe y visualizamos los primeros elementos. Vemos que esta sería la diferencia entre el valor real y el estimado

```
> residuos = as.data.frame(residuos)
> head(residuos)
      residuos
1  0.9349204
2  1.5308295
3  1.2716170
4  1.7883341
5  1.0233777
6 -1.4216529
> |
```

4. EJEMPLO REGRESION LINEAL II

Paso 9. Con estos residuos generamos un histograma donde veremos mejor todos los datos.

```
> library(ggplot2)
> ggplot(residuos,aes(residuos)) + geom_histogram(fill='blue',alpha=0.5)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> |
```



Vemos que la mayor parte de los valores de los residuos (diferencia entre valores reales y los estimados) se concentran entorno al 0, donde no hay ninguna discrepancia entre lo estimado y el valor real.

Anteriormente ya vimos que el valor del modelo se acercaba a 1 (0.83) y que se ajustaba para calcular las estimaciones del valor de $G3$.

Por tanto el modelo es bueno