

# Práctica 10. Regresión Lineal Simple y Correlación en R.

Juan Melchor y Pedro J. Femia Marzo



**UNIVERSIDAD  
DE GRANADA**

Todo el material para el conjunto de actividades de este curso ha sido elaborado y es propiedad intelectual del grupo **BioestadísticaR** formado por:

Antonio Martín Andrés  
Juan de Dios Luna del Castillo,  
Pedro Femia Marzo,  
Miguel Ángel Montero Alonso,  
Christian José Acal González,  
Pedro María Carmona Sáez,  
Juan Manuel Melchor Rodríguez,  
José Luis Romero Béjar,  
Manuela Expósito Ruíz,  
Juan Antonio Villatoro García.

Todos los integrantes del grupo han participado en todas las actividades, en su elección, construcción, correcciones o en su edición final, no obstante, en cada una de ellas, aparecerán uno o más nombres correspondientes a las personas que han tenido la máxima responsabilidad de su elaboración junto al grupo de **BioestadísticaR**.

Todos los materiales están protegidos por la Licencia Creative Commons **CC BY-NC-ND** que permite "descargar las obras y compartirlas con otras personas, siempre que se reconozca su autoría, pero no se pueden cambiar de ninguna manera ni se pueden utilizar comercialmente".

# Práctica 10. Regresión Lineal Simple y Correlación en R.

Juan Melchor y Pedro Femia – Grupo BioestadísticaR

Para introducirnos en resolver problemas de predicción donde la naturaleza de las variables es cuantitativa se recurre a la Teoría de Regresión y Correlación que nos sirve para establecer un modelización lineal adecuada. En el desarrollo de este guión utilizamos el lenguaje R y sus funciones relativas a los procedimientos de Regresión lineal Simple y Correlaciones bivariadas. Para llevar a cabo esta metodología, se requiere del cumplimiento de los supuestos de normalidad, homocedasticidad y linealidad (que conocemos como modelo de regresión lineal). En este sentido, para la verificación de dos de dichos supuestos (linealidad e igualdad de varianzas), son muy útiles los paquetes gráficos de dispersión. Por lo tanto, el desarrollo de este documento comienza explorando este tipo de diagramas.

## 10.1. Diagramas de Dispersión

Para realizar un diagrama de dispersión, es muy útil observar la nube de puntos generada por las coordenadas de dos variables cuantitativas. En R, podemos recurrir como primera opción, siendo quizás la más sencilla a la función `plot()`. En la primera posición se sitúa la variable que se representa en el eje de abscisas (horizontal), que será la explicativa, que corresponde a la  $X$ , y en segundo lugar la que corresponde al eje de ordenadas (vertical), que es la que se pretende explicar es decir la  $Y$ .

Ejemplo 1. A continuación, figuran las tallas medidas en cm (Talla) y la Capacidad Vital Forzada medida en litros (CVF), en una muestra de 15 niños entre 6 y 12 años de edad:

Caso	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Talla (cm)	114	188	122	126	129	131	135	138	141	143	147	151	152	155	158
CFV (l)	1.24	1.39	1.39	1.67	1.74	1.23	1.23	2.14	2.39	2.36	2.44	2.81	2.73	2.91	3.27

El objetivo es visualizar la relación entre las variables *talla* y *cvf*.

Podríamos representar la variable *talla* como explicativa  $X$  y la capacidad vital forzada *cvf* como la variable que se quiere explicar  $Y$ . Pero primero creemos un *data frame* con los datos:

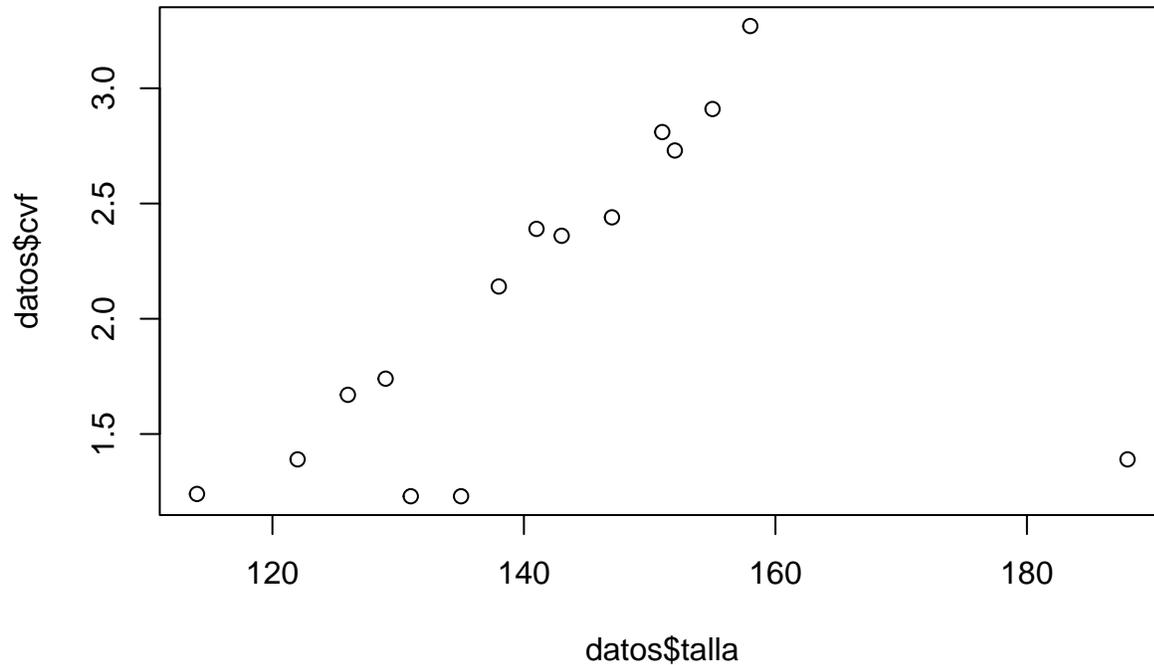
```
caso<-c(1:15)
talla<-c(114,188,122,126,129,131,135,138,141,143,147,151,152,155,158)
cvf<-c(1.24,1.39,1.39,1.67,1.74,1.23,1.23,2.14,2.39,2.36, 2.44,2.81,2.73,2.91,3.27)
datos<-data.frame(caso,talla,cvf)
datos
```

```
##      caso talla  cvf
## 1         1  114 1.24
## 2         2  188 1.39
## 3         3  122 1.39
## 4         4  126 1.67
## 5         5  129 1.74
## 6         6  131 1.23
## 7         7  135 1.23
## 8         8  138 2.14
## 9         9  141 2.39
```

```
## 10  10  143 2.36
## 11  11  147 2.44
## 12  12  151 2.81
## 13  13  152 2.73
## 14  14  155 2.91
## 15  15  158 3.27
```

Si queremos visualizar la relación entre ambas variables mediante un diagrama de dispersión haciendo uso de la función `plot()`, la sentencia sería la siguiente:

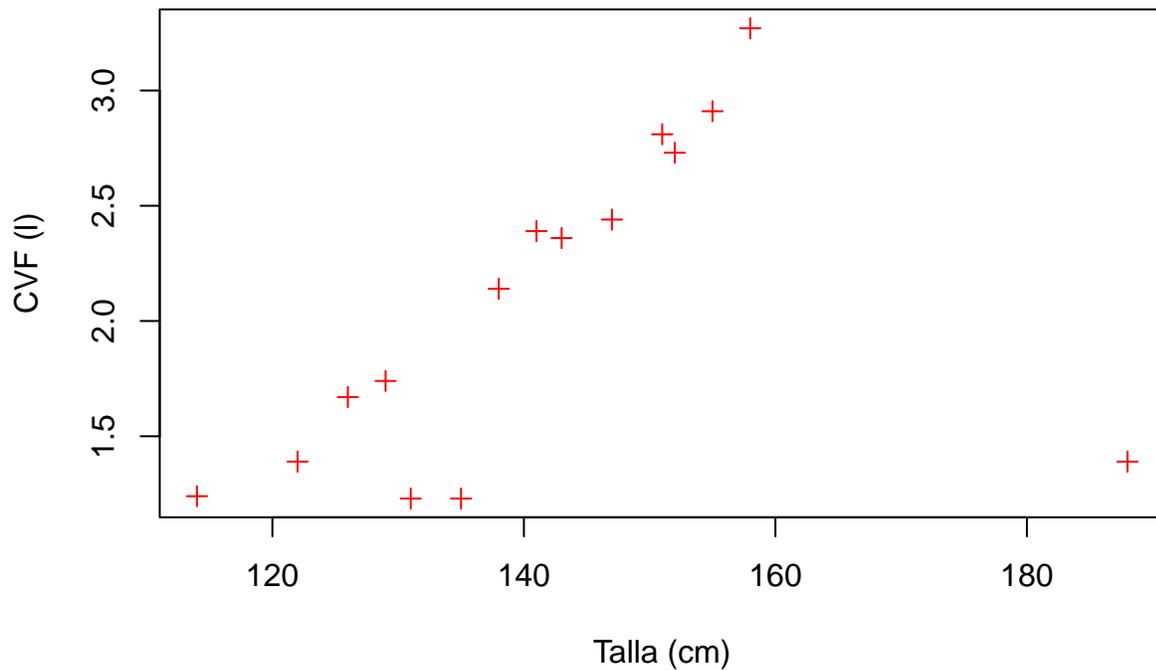
```
plot(datos$talla,datos$cvf)
```



Podemos además, customizar la apariencia del diagrama mediante el etiquetado de los ejes *xlab* e *ylab*, dándole nombre al gráfico *main* y cambiando el símbolo con el que se representan los puntos *pch* (desde 0 a 25 valores distintos) o su color *col*.

```
plot(x=datos$talla, y=datos$cvf, main = "Relación entre CVF y Talla",
     xlab = "Talla (cm)" ,ylab = "CVF (l)",pch=3,col="red" )
```

## Relación entre CVF y Talla

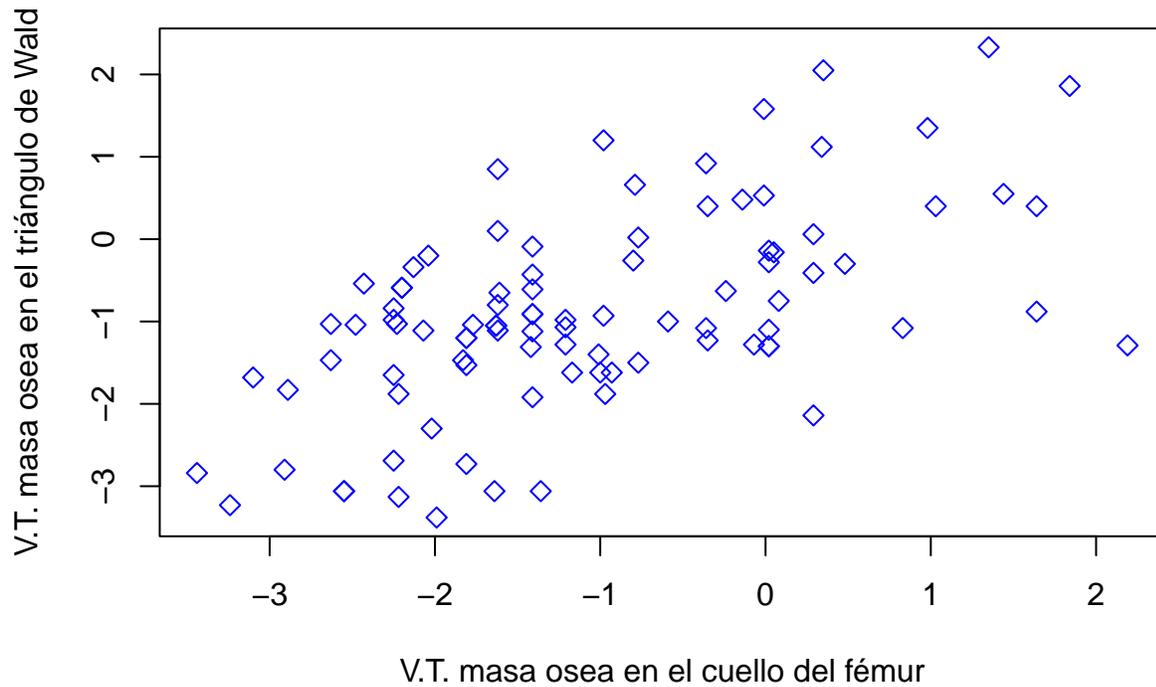


La función `plot()` tiene buena compatibilidad con la importación de variables desde bases de datos sin necesidad de instalar o cargar otras librerías. Por ejemplo, si recurrimos a la base de datos `osteo.sav` podemos elegir las variables `szcue` valor tipificado de la masa ósea en el cuello del fémur y la variable `sztri` valor tipificado de la masa ósea en el triángulo de Wald y representar un diagrama de dispersión como se muestra en el gráfico.

```
library(foreign)
osteo<-read.spss("~/Desktop/osteo.sav",to.data.frame=TRUE)
```

```
plot(x=osteo$szcue, y=osteo$sztri, main = "Relación entre V.T. masas oseas",
     xlab = "V.T. masa ósea en el cuello del fémur" ,
     ylab = "V.T. masa ósea en el triángulo de Wald",pch=5,col="blue")
```

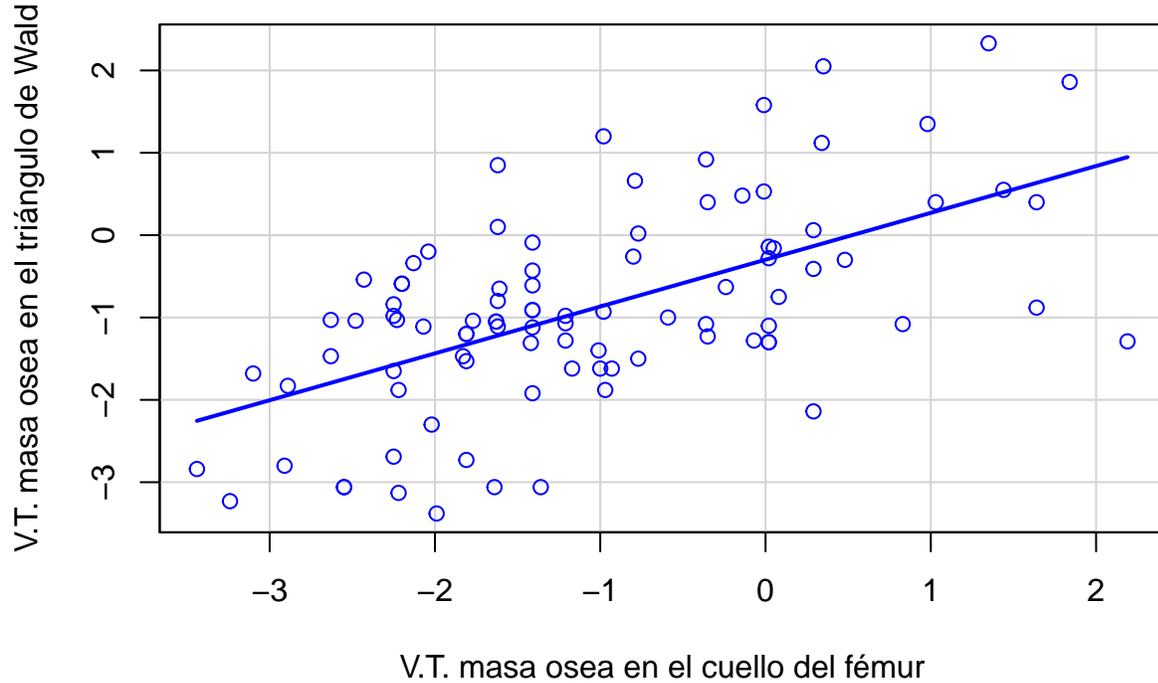
## Relación entre V.T. masas oseas



Otra función muy útil para representar un diagrama de dispersión es la función `scatterplot()`, esta función nos permite añadir las variables separadas por `~` en formato fórmula y la base de datos a la que pertenecen. Para poder utilizarla es necesario instalar y cargar el paquete `car`.

```
library(car)
scatterplot(sztri ~ szcue, data=osteo, xlab="V.T. masa osea en el cuello del fémur",
           ylab="V.T. masa osea en el triángulo de Wald",
           main="Relación entre V.T. de masas oseas", smooth=FALSE, boxplots="",)
```

## Relación entre V.T. de masas oseas

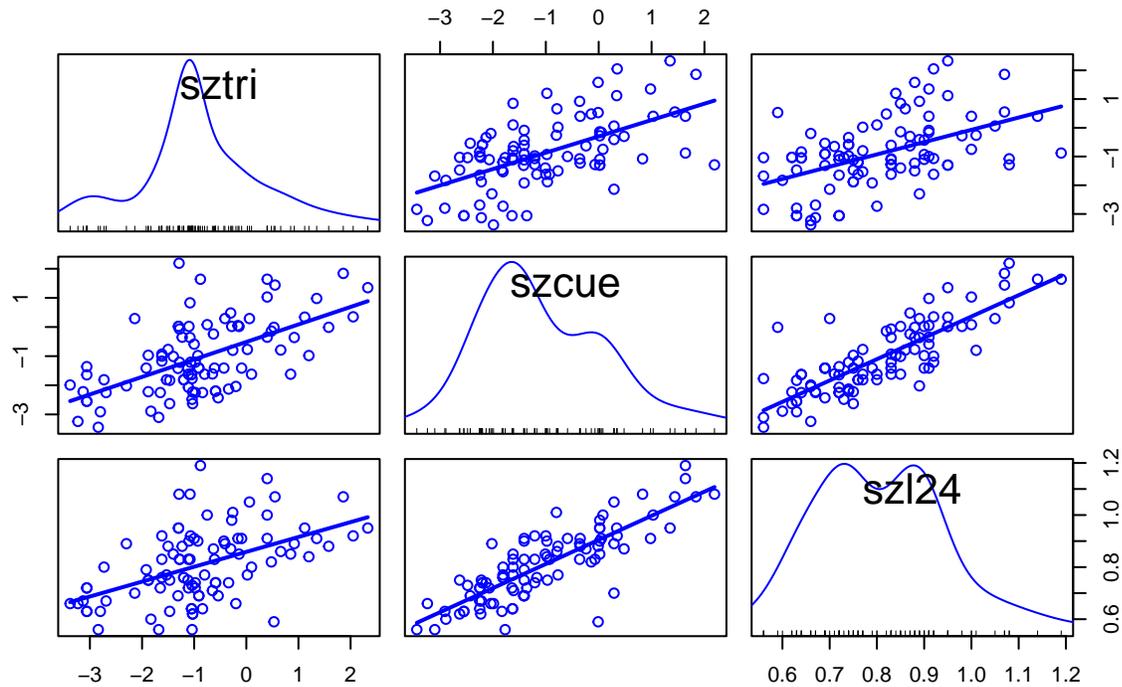


La función nos introduce el ajuste del modelo en el gráfico mediante la recta de regresión, esta podría desactivarse con el comando `regLine=FALSE`. Además, esta función permite ajustar la relación entre las variables con un suavizado de la regresión mediante un método no lineal, que en este caso se desactiva con `smooth=FALSE`, hay que destacar que se puede introducir (en los ejes) un diagrama de caja para las variables que también se desactiva con `boxplots=""`.

Este paquete también nos permite realizar gráficos matriciales de dispersión a través de la función `scatterplotMatrix()`, añadiendo `~` y después las variables que se pretenden analizar.

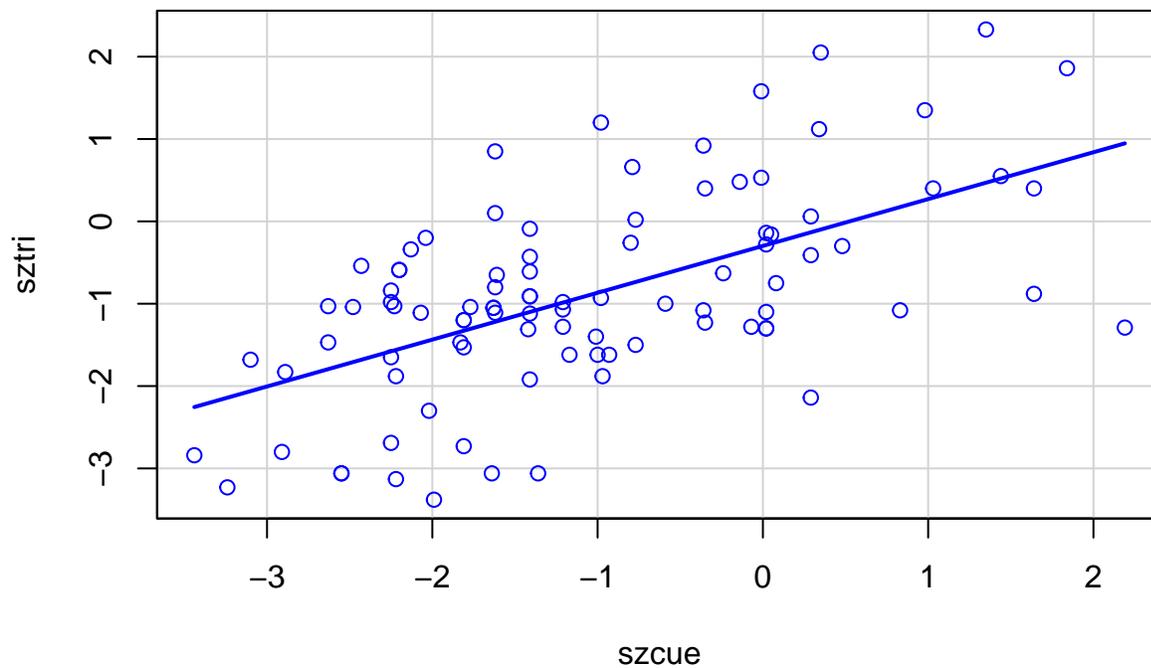
```
library(car)
scatterplotMatrix(~sztri+szcue+szl24,data=osteo,main="Valores de masas oseas",smooth=FALSE)
```

## Valores de masas oseas

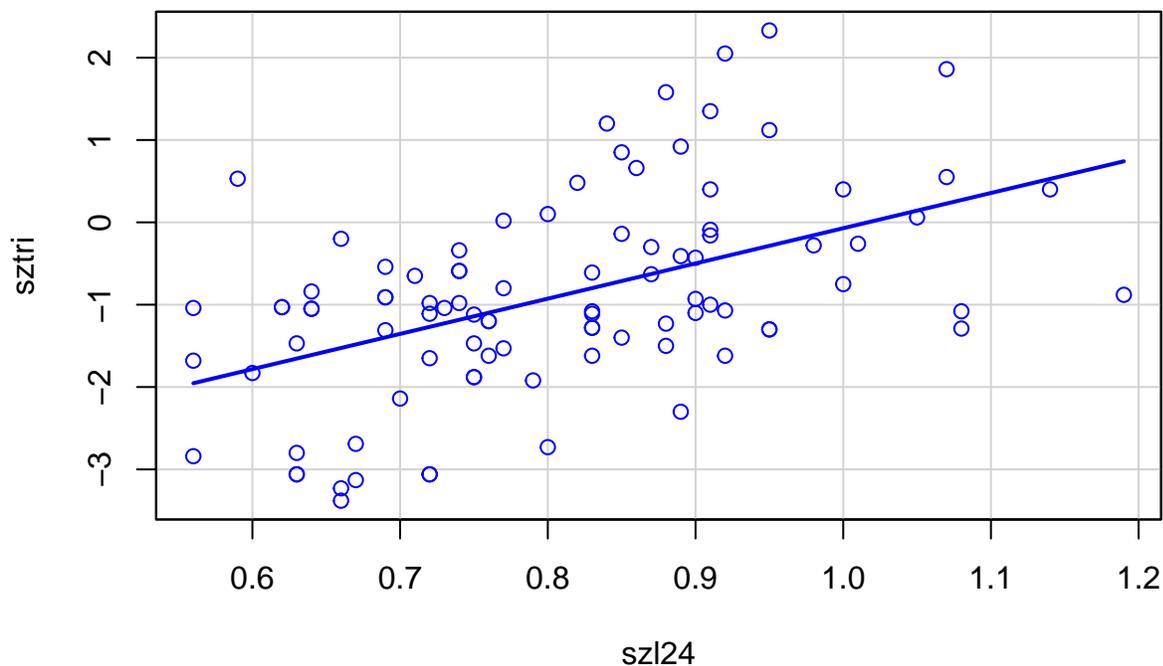


Cómo se puede observar en el gráfico, las diagonales muestran las distribuciones de frecuencias de cada variable y los gráficos son simétricos. Los diagramas representados por separado son:

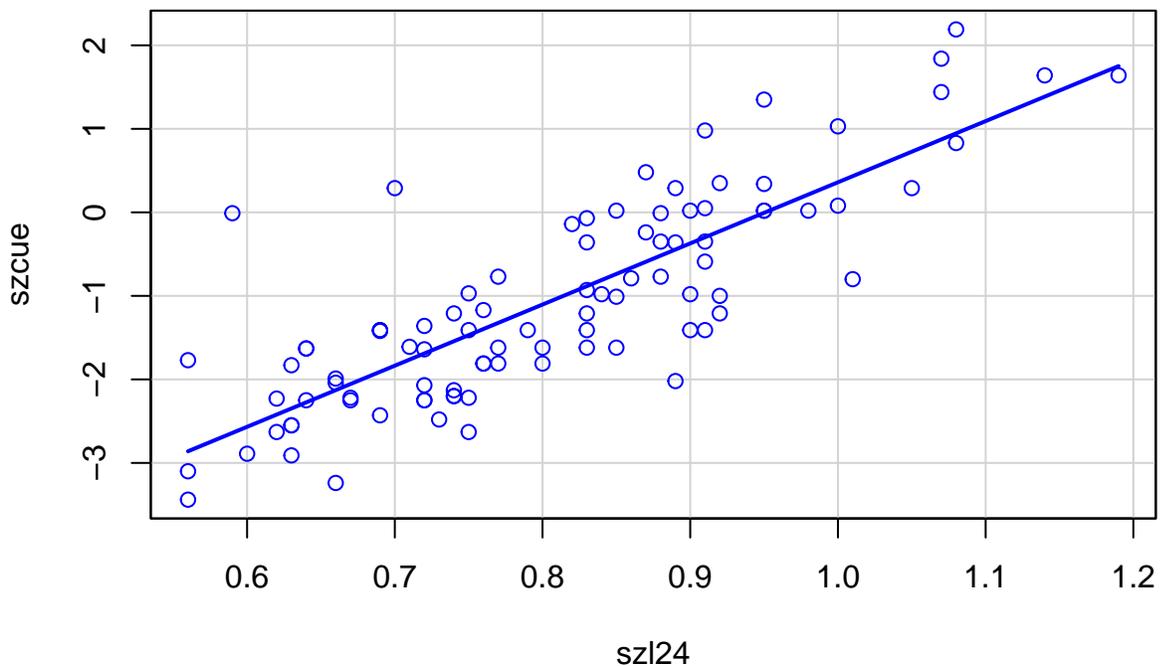
```
scatterplot(sztri ~ szcue, data=osteo, smooth=FALSE, boxplots="")
```



```
scatterplot(sztri ~ szl24, data=osteo, smooth=FALSE, boxplots="")
```



```
scatterplot(szcue ~ szl24, data=osteo, smooth=FALSE, boxplots="")
```



La función con más interés para los diagramas de dispersión puede ser **gf\_point()**, que nos muestra como salda un diagrama de dispersión en el que es posible identificar por color y forma diferentes categorías de una variable cualitativa. Para ello, es necesario instalar el paquete **ggformula**, sin embargo, su uso es similar a los descritos anteriormente, introduciendo la fórmula con las variables continuas separadas por  $\sim$ . Los datos se añaden con el comando *data=*, el color con *color=*, añadiendo la variable por la que se quiere cambiar el color, la forma con *shape=* donde se llama a la variable mediante la cual se quiere señalar en la nube de puntos con una forma atendiendo a una modalidad y la intensidad del color con *alpha=* (que será un número entre 0 y 1, dónde 1 es la máxima intensidad y 0 la mínima donde se difumina el punto).

```
library(ggformula)
gf_point(sztri~szcue,data=osteo,color=~osteo$alcohol,shape=osteo$alcohol,alpha = 0.7)
```



## 10.2. Regresión Lineal Simple

### 10.2.1. Modelo de Regresión Lineal Simple

Para explicar la relación de las variables cuantitativas a estudio, en un caso genérico  $Y$  con  $X$  recurrimos a expresar de forma general como  $Y = f(X)$ . En nuestro caso, este tipo formulación da lugar a un modelo bioestadístico, cuyas componentes de  $f(X)$  están compuestas por un sumando aleatorio y otro determinista. En este contexto, lineal y simple, su forma generalizada será:

$$Y = f(X) = \beta_0 + \beta_1 X + \varepsilon$$

donde  $\beta_0$  es el término independiente,  $\beta_1$  es la pendiente y  $\varepsilon$  el término de error.

Si queremos ajustar un modelo de regresión lineal, una función que podemos utilizar en R para llevar a cabo este procedimiento es `lm()`. La formulación que se introduce en la función corresponde a utilizar la variable dependiente seguida del símbolo tilde `~` y la variable independiente.

En el ejemplo 1, podríamos intentar explicar la relación de la variable CVF a partir de la variable talla y el modelo sería:

$$\text{CVF} = \beta_0 + \beta_1 \text{Talla} + \varepsilon$$

En R podríamos calcularlo como:

```
Modelo.RLS<-lm(datos$cvf~datos$talla)
Modelo.RLS
```

```
##
## Call:
## lm(formula = datos$cvf ~ datos$talla)
##
## Coefficients:
## (Intercept)  datos$talla
##   -0.32140      0.01679
```

La salida que nos muestra el programa se interpreta como la estimación de  $\hat{\beta}_0$  con el primer número -0.32140 como la constante (intercept) y  $\hat{\beta}_1$  con el segundo número 0.01679 como la pendiente (datos\$talla). Es decir, tras el resultado estimado el modelo toma la siguiente forma:

$$\widehat{\text{CVF}} = \hat{\beta}_0 + \hat{\beta}_1 \text{Talla} = -0.32140 + 0.01679 \cdot \text{Talla}$$

Es decir, el valor pronóstico del CVF puede obtenerse a partir de la Talla multiplicada por la pendiente estimada y sumando el valor constante estimado. Sin embargo, como la salida de R no indica nada sobre la significación de esta estimación o sus intervalos de confianza, recurrimos al test de regresión.

### 10.2.2. Test de Regresión Lineal Simple

A cada coeficiente del modelo de Regresión Lineal Simple se le puede asociar un test. No tiene mucho sentido el test de regresión para la constante, por lo tanto el test de regresión simple que vamos a tener en cuenta se calcula para la pendiente, que viene definido como:

$$\begin{cases} \mathcal{H}_0 : \beta_1 = 0 \\ \mathcal{H}_1 : \beta_1 \neq 0 \end{cases}$$

La interpretación nos permite concluir que  $Y$  cambia significativamente con  $X$ , que en el caso del ejemplo 1 sería lo mismo que indicar que la CVF cambia significativamente con la talla. La manera de explorar y calcular la significación del modelo para la pendiente a través de R es mediante la función **summary()**, ampliamente utilizada en el ámbito de estadística descriptiva, para calcular las medias usuales de posición. En este caso, si introducimos la función del modelo **summary(Modelo.RLS)** la salida de R nos permite calcular el valor p entre otros resultados:

```
summary(Modelo.RLS)
```

```
##
## Call:
## lm(formula = datos$cvf ~ datos$talla)
##
## Residuals:
##   Min     1Q  Median     3Q    Max
## -1.4450 -0.3447  0.1445  0.4218  0.9387
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.321396   1.376573  -0.233   0.819
## datos$talla  0.016789   0.009622   1.745   0.105
##
## Residual standard error: 0.6486 on 13 degrees of freedom
## Multiple R-squared:  0.1898, Adjusted R-squared:  0.1274
## F-statistic: 3.044 on 1 and 13 DF,  p-value: 0.1046
```

El resumen que se obtiene comienza mostrando el modelo, los cuartiles, máximos y mínimos de los residuos, las estimaciones de los coeficientes del modelo, los errores standard de los coeficientes, los estadísticos de contraste  $t_{exp}$  y los valores p y el grado de asociación entre las variables.

Una forma adecuada de presentar este resultado sería:

$$\hat{\beta}_{\text{Talla}}(\text{S.E.}) = 0.017 \cdot 0.010; t_{\text{exp}}(13\text{gl}) = 1.745; p = 0.105$$

Se interpreta como que la estimación del coeficiente de la pendiente de la regresión en términos de su error standard no es significativo estadísticamente. Por lo tanto, no podemos aceptar la hipótesis alternativa. Y no habría una relación predictiva significativa (seguramente debido a la presencia de algún valor atípico que se podría observar en el diagrama de dispersión).

Además de este test, tendríamos que asegurarnos de la normalidad de las variables a través de un test como por ejemplo el de Shapiro-Wilk u otros métodos como por ejemplo los gráficos Q-Q, y de la hipótesis de homocedasticidad, por ejemplo analizando el diagrama de dispersión.

La línea donde se menciona el coeficiente  $R^2$  de determinación corresponde al porcentaje de la variabilidad explicada del modelo con respecto al total de la variabilidad.

Este coeficiente aparece de dos formas, la primera es para el caso simple y la segunda en su forma ajustada para el caso en el que tenemos varias variables predictoras (que no se verá en este curso). Por lo tanto, en este caso la cantidad de la variabilidad que explica el modelo es aproximadamente del 19%.

Si queremos extraer la información de los intervalos de confianza para los coeficientes de regresión, utilizaremos la función `confint()`, pudiendo regular el nivel de confianza con la orden `level=0.95` (95% por defecto).

```
confint(Modelo.RLS, level=0.95)

##                2.5 %    97.5 %
## (Intercept) -3.295301584 2.6525099
## datos$talla -0.003998258 0.0375766
```

O bien, ser más precisos y utilizar la pendiente que es la variable de interés para calcular el intervalo, introduciendo entre comillas el nombre de la variable dentro de la función:

```
confint(Modelo.RLS, "datos$talla", level=0.95)

##                2.5 %    97.5 %
## datos$talla -0.003998258 0.0375766
```

En cualquier caso, el resultado para la pendiente es el intervalo de confianza a un 95% (-0.04,0.038), que cómo contiene al cero es coherente con el valor  $p$  obtenido y no podría rechazarse la hipótesis nula.

Para que el modelo sea predictivo y se pueda pronósticar un valor a partir de un dato, los valores deben de estar en el rango de observación y no exceder ni en el mínimo, ni en el máximo de las observaciones de las variables.

En el siguiente supuesto podemos utilizar el ejemplo de la base *osteo*, para las variables Valor tipificado de la densidad de masa ósea del L24 (*szl24*), es decir  $Y$ , la variable dependiente y Valor tipificado de la densidad de masa ósea del triángulo (*sztri*), es decir,  $X$  la variable independiente. En este ejemplo, se puede observar que el diagrama de dispersión cumple los supuestos de linealidad e igualdad de varianzas y suponemos comprobado el supuesto de normalidad. Podemos por tanto escribir el modelo como:

```
Modelo.osteo<-lm(osteo$szl24~osteo$sztri)
Modelo.osteo

##
## Call:
## lm(formula = osteo$szl24 ~ osteo$sztri)
##
## Coefficients:
## (Intercept)  osteo$sztri
##      0.85851      0.05714
```

Ahora calculando su valor p e intervalos de confianza:

```
summary(Modelo.osteo)

##
## Call:
## lm(formula = osteo$szl24 ~ osteo$sztri)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29880 -0.07359 -0.01182  0.06555  0.38177
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.85851    0.01572  54.608 < 2e-16 ***
## osteo$sztri  0.05714    0.01047   5.456 4.09e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1223 on 92 degrees of freedom
## Multiple R-squared:  0.2444, Adjusted R-squared:  0.2362
## F-statistic: 29.76 on 1 and 92 DF,  p-value: 4.093e-07
confint(Modelo.osteo, "osteo$sztri")

##              2.5 %      97.5 %
## osteo$sztri 0.03633592 0.07793704
```

En este caso al ser un valor p muy inferior al nivel de error podríamos inferir que la relación de regresión es estadísticamente significativa rechazando la hipótesis nula.

El modelo de regresión lineal simple será:

$$Y = 0.859 + 0.057 \cdot X$$

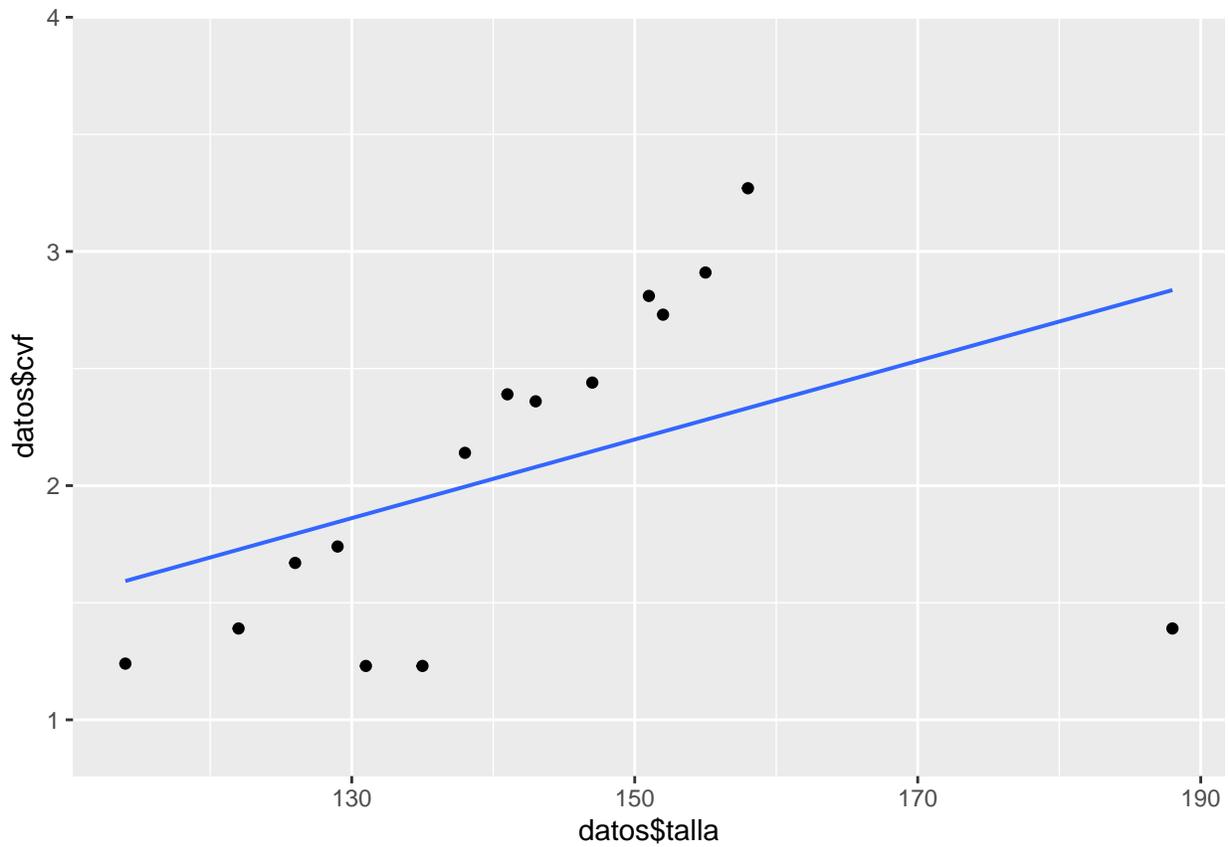
O bien,

$$\widehat{szl24} = 0.859 + 0.057 \cdot sztri$$

Es fácil comprobar en los resultados que el intervalo de confianza no contiene al cero lo que corrobora la aceptación de la hipótesis alternativa.

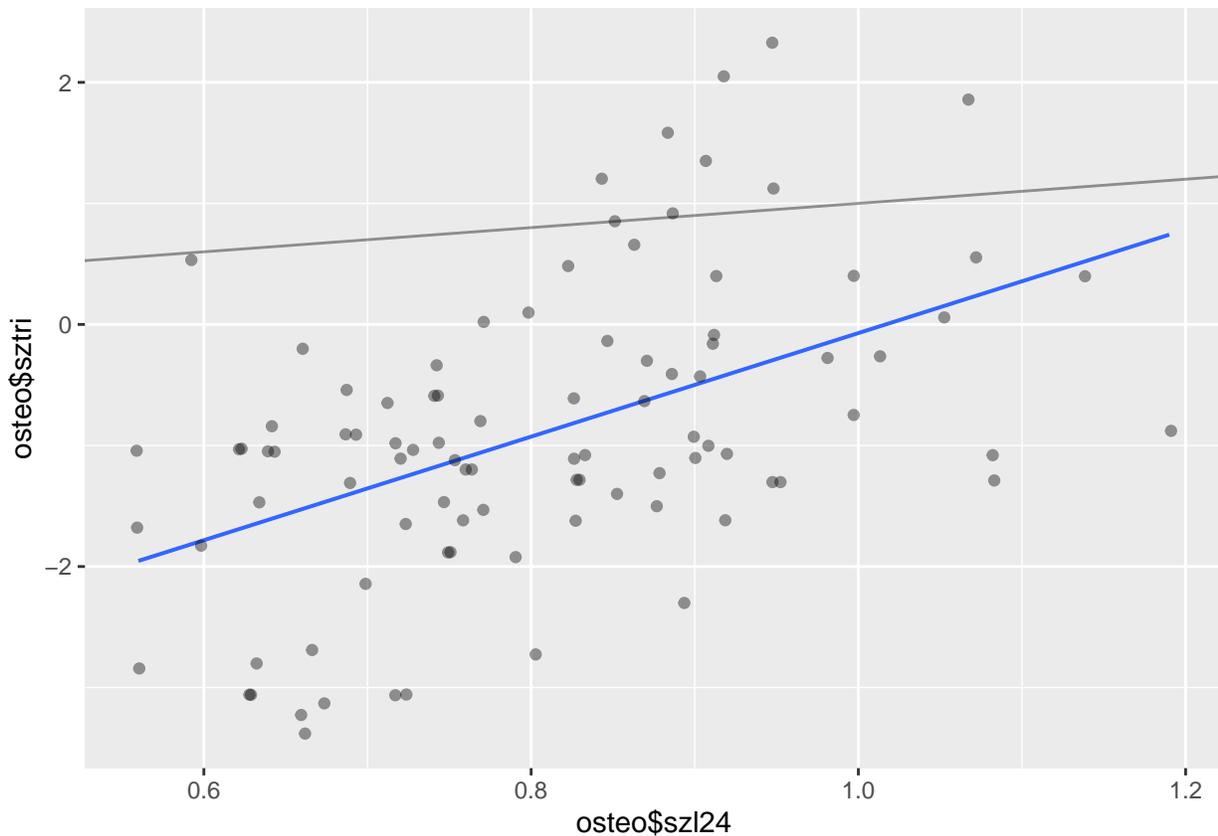
Otra forma para realizar el ajuste de regresión lineal simple puede ser mediante la función `gf_lm()` una vez representados los datos con `gf_point()`.

```
Modelo.RLS<-lm(datos$cvf~datos$talla)
gf_lm(datos$cvf~datos$talla, data=datos)%>%
gf_point()
```



E incluso, para comparar con otras pendientes como la pendiente 1 de referencia se usa la función `gf_abline()`:

```
gf_lm(osteo$sztri~osteo$szl24, data=osteo)%>%  
gf_jitter(alpha=0.4)%>%  
gf_abline(slope=1,intercept=0, alpha=0.4)
```



En adelante, se detallan con ejemplos una serie de funciones de interés para regresión lineal simple, `summary()` se ha visto previamente, ahora vamos a considerar: `coef()` para visualizar los coeficientes del modelo, `resid()` para calcular los residuos y `fitted()` para los valores ajustados a partir de los datos del modelo (en este caso  $\widehat{CVF}$ ).

```
coef(Modelo.RLS)
```

```
## (Intercept) datos$talla
## -0.32139583  0.01678917
```

```
fitted(Modelo.RLS)
```

```
##      1      2      3      4      5      6      7      8
## 1.592570 2.834969 1.726883 1.794040 1.844407 1.877986 1.945142 1.995510
##      9     10     11     12     13     14     15
## 2.045877 2.079456 2.146613 2.213769 2.230558 2.280926 2.331293
```

```
resid(Modelo.RLS)
```

```
##      1      2      3      4      5      6      7
## -0.3525698 -1.4449686 -0.3368832 -0.1240399 -0.1044074 -0.6479858 -0.7151425
##      8      9     10     11     12     13     14
##  0.1444900  0.3441225  0.2805442  0.2933875  0.5962308  0.4994416  0.6290741
##      15
##  0.9387066
```

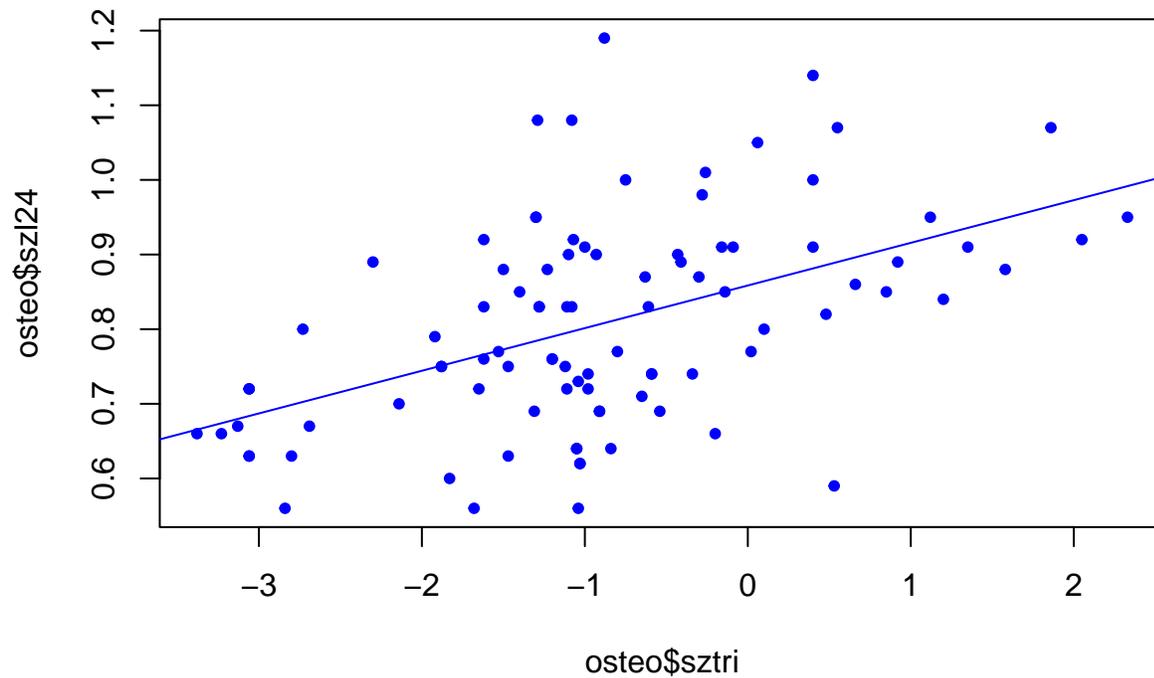
```
datos$cvf-fitted(Modelo.RLS)
```

```
##      1      2      3      4      5      6      7
## -0.3525698 -1.4449686 -0.3368832 -0.1240399 -0.1044074 -0.6479858 -0.7151425
```

```
##           8           9           10           11           12           13           14
## 0.1444900 0.3441225 0.2805442 0.2933875 0.5962308 0.4994416 0.6290741
##          15
## 0.9387066
```

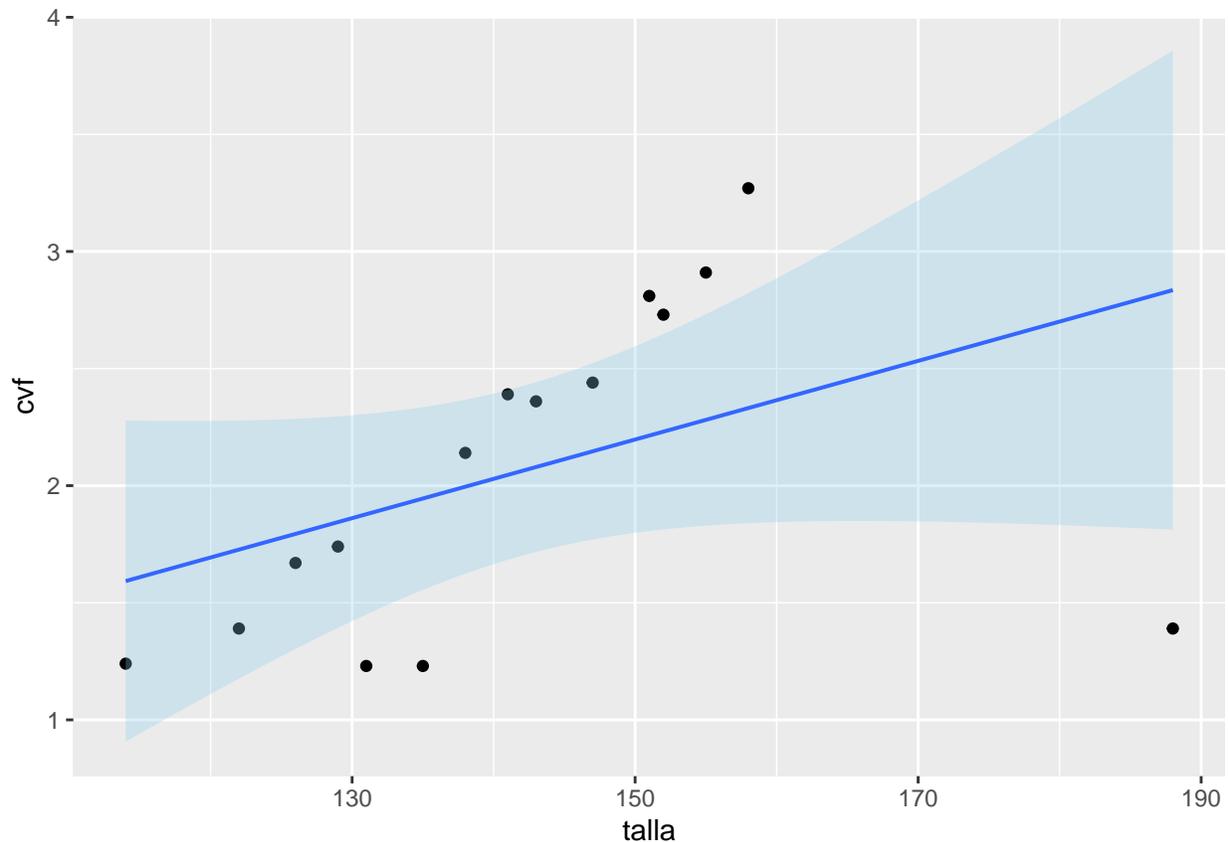
Podríamos indicar la línea de ajuste en el diagrama de dispersión también gracias a la función `abline()` y así observar los resultados:

```
plot(x=osteo$sztri,y=osteo$szl24,pch=20,col="blue")
abline(Modelo.osteo,col="blue")
```



También es interesante explorar los intervalos de confianza en el ajuste del modelo, con esto se puede interpretar el rango estimado de manera gráfica. Con la función `gf_lm()` es posible realizar esta representación, indicando el tipo de intervalo, en este caso `interval="confidence"`:

```
gf_point(cvf~talla,data=datos)%>%
gf_lm(interval="confidence", fill="skyblue")
```



### 10.3. Correlación Lineal

Si queremos correlación entre las variables continuas, entendida como el grado de asociación entre las mismas podemos utilizar la función `cor()`. Para ello, simplemente introducimos como parámetros las dos variables separadas por una coma:

```
cor(datos$talla,datos$cvf)
```

```
## [1] 0.4356057
```

#### 10.3.1. Correlación Lineal (Coeficiente de correlación de Pearson)

Por defecto, el coeficiente de correlación calculado es el coeficiente de correlación de Pearson  $R^2$ . En regresión lineal simple se cumple que el coeficiente de determinación es el de correlación.

Veamos el ejemplo introduciendo los datos de los valores tipificados de la densidad de masa ósea, desde la base de datos `osteo`:

```
cor(osteo$sztri,osteo$szl24)
```

```
## [1] 0.494401
```

#### 10.3.2. Correlación Lineal (Coeficiente de correlación de Spearman)

Además, podemos cambiar el método para calcular la asociación entre las variables. El método de Spearman nos permite incluso averiguar el sentido de la asociación, a través del parámetro `method`:

```
cor(datos$talla,datos$cvf, method="spearman")
```

```
## [1] 0.6726308
```

```
cor(osteo$sztri,osteo$szl24, method="spearman")
```

```
## [1] 0.4735113
```

Si queremos realizar el test de correlación cuya formulación es:

$$\begin{cases} \mathcal{H}_0 : \rho = 0 \\ \mathcal{H}_1 : \rho \neq 0 \end{cases}$$

y cuyo estadístico de contraste es  $t_{exp} = \sqrt{\frac{(n-2)r^2}{1-r^2}}$  siendo  $n$  el tamaño muestral y  $r$  el estimador muestral del coeficiente de correlación lineal. Podemos entonces, recurrir a la función `cor.test()`, sus parámetros son los mismos que se calculan en la función `cor()`, pero en este caso obtendremos la estimación del intervalo de confianza para la correlación poblacional y el valor  $p$ . También es posible cambiar el método para obtener el valor de la asociación a través del parámetro `method`.

```
cor.test(osteo$sztri,osteo$szl24)
```

```
##
## Pearson's product-moment correlation
##
## data: osteo$sztri and osteo$szl24
## t = 5.4555, df = 92, p-value = 4.093e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3242672 0.6335525
## sample estimates:
##      cor
## 0.494401
```

Si  $n > 30$  el estadístico de contraste es  $z_{exp} = |\rho_s| \sqrt{n-1}$ , en caso en el que  $n \leq 30$  se compara  $|\rho_s|$  con la tabla  $r_\alpha$ . Pero en el programa R studio bastaría con cambiar el método de correlación con el comando `method=spearman` dentro de la función `cor.test()`:

```
cor.test(osteo$sztri,osteo$szl24, method="spearman")
```

```
##
## Spearman's rank correlation rho
##
## data: osteo$sztri and osteo$szl24
## S = 72874, p-value = 1.435e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.4735113
```

## 10.5. Inferencias sobre el modelo de Regresión Lineal Simple

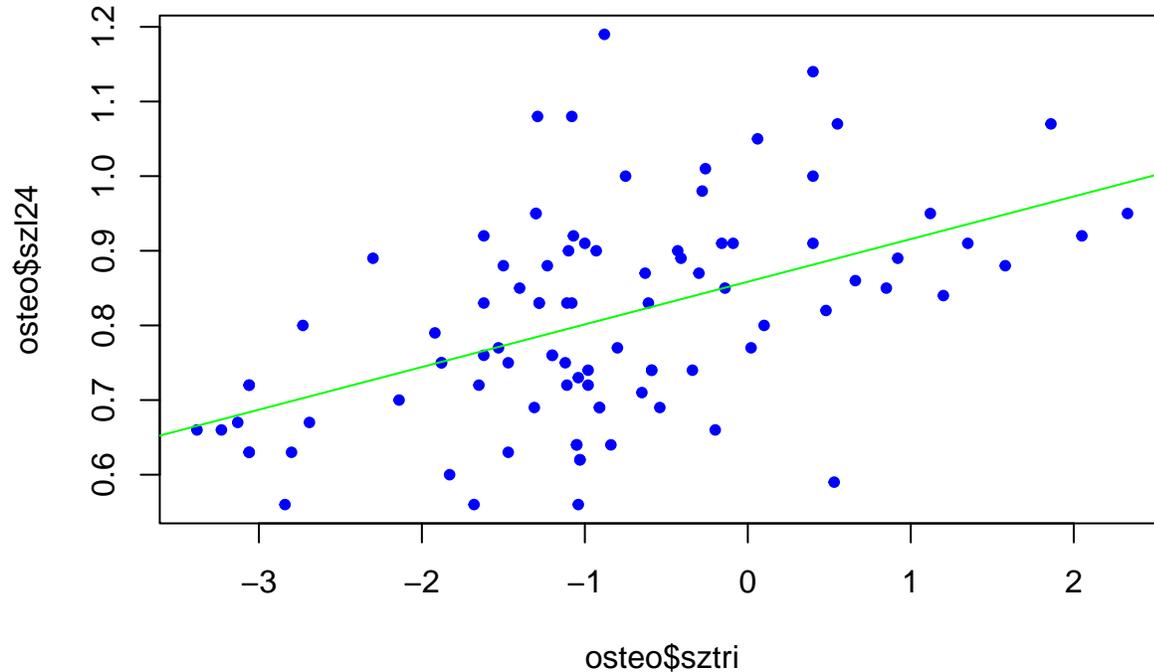
Si queremos predecir el valor de la variable respuesta a partir de la variable explicativa, tenemos que tener en cuenta las hipótesis del modelo de Regresión Lineal Simple (Normalidad, Homocedasticidad y Linealidad) y una vez comprobadas, en el rango de obtención de los datos podremos calcular un valor pronóstico.

Si vamos al ejemplo de la base de datos *osteo*, donde tenemos el valor tipificado de la densidad de la masa ósea del L24 *szl24* a partir del valor en el triángulo de Wald *sztri*, el modelo se formularía de la siguiente forma:

$$\widehat{szl24} = 0.859 + 0.057 \cdot sztri$$

si lo queremos representar gráficamente como se vio anteriormente, por ejemplo a través de las funciones `plot()` y `abline()` se vería así:

```
Modelo.osteo<-lm(osteo$szl24~osteo$sztri)
plot(x=osteo$sztri,y=osteo$szl24,pch=20,col="blue")
abline(Modelo.osteo,col="green")
```



Si queremos ahora estimar el valor de `szl24` a partir de una observación de `sztri` por ejemplo 0.2856. La fórmula y notación adecuada sería:

$$\widehat{szl24}|(sztri = 0.2856) = 0.859 + 0.057 \cdot 0.2856$$

Se podría hacer rápidamente con calculadora pero es muy útil para casos más complejos o múltiples en R eligiendo los coeficientes del modelo desde `coefficients[[]]` numerando `[[1]]` como el término independiente o intercept y `[[2]]` como la pendiente o slope (aparece con el nombre de la variable predictora en la salida de R):

```
Modelo.osteo$coefficients[[1]]+Modelo.osteo$coefficients[[2]]*0.2856
```

```
## [1] 0.8748322
```