Práctica 10. Regresión Lineal Simple y Correlación en R.

Juan Melchor, Pedro J. Femia y Grupo BioestadísticaR

UNIVERSIDAD
DE GRANADA

Práctica 10. Regresión Lineal Simple y Correlación en R.

Juan Melchor y Pedro Femia – Grupo BioestadisticaR

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 situa la variable que se representa en el eje de abcisas (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 prentende 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:

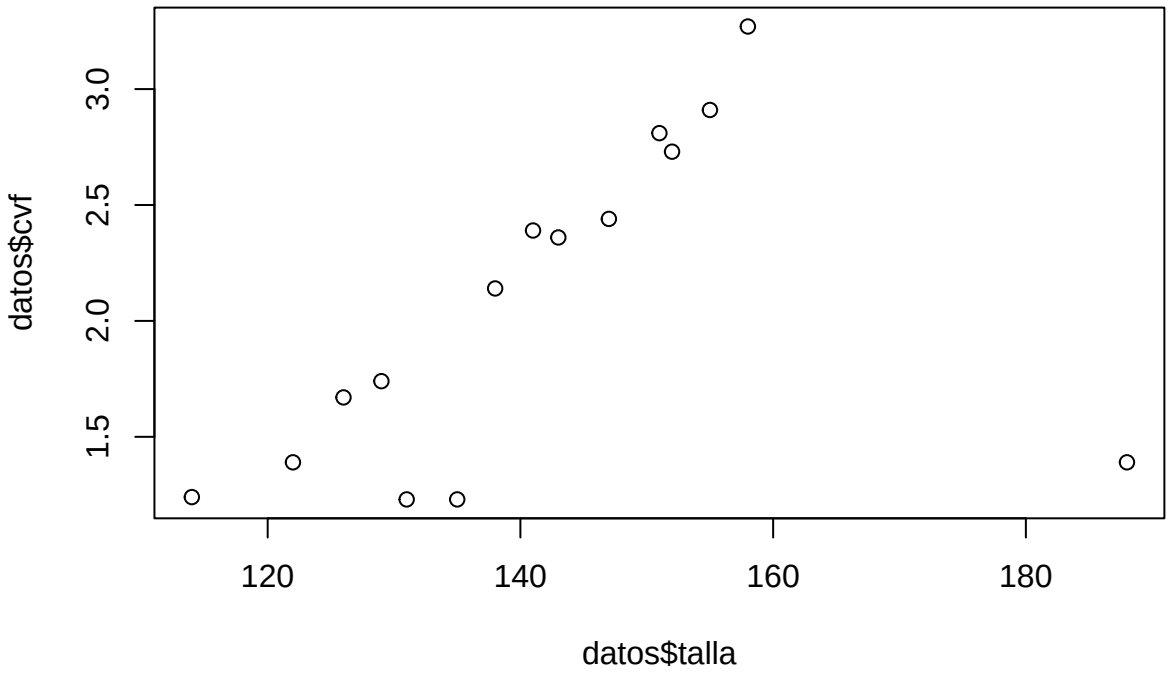
```
caso<-c(1:15)
talla<-c(114,188,122,126,129,131,135,138,141,143,147,151,152,155,158)
cvf<-c(1.24,1.39,1.39,1.67,1.74,1.23,1.23,2.14,2.39,2.36, 2.44,2.81,2.73,2.91,3.27)
datos<-data.frame(caso,talla,cvf)
datos
```

```
##    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
```

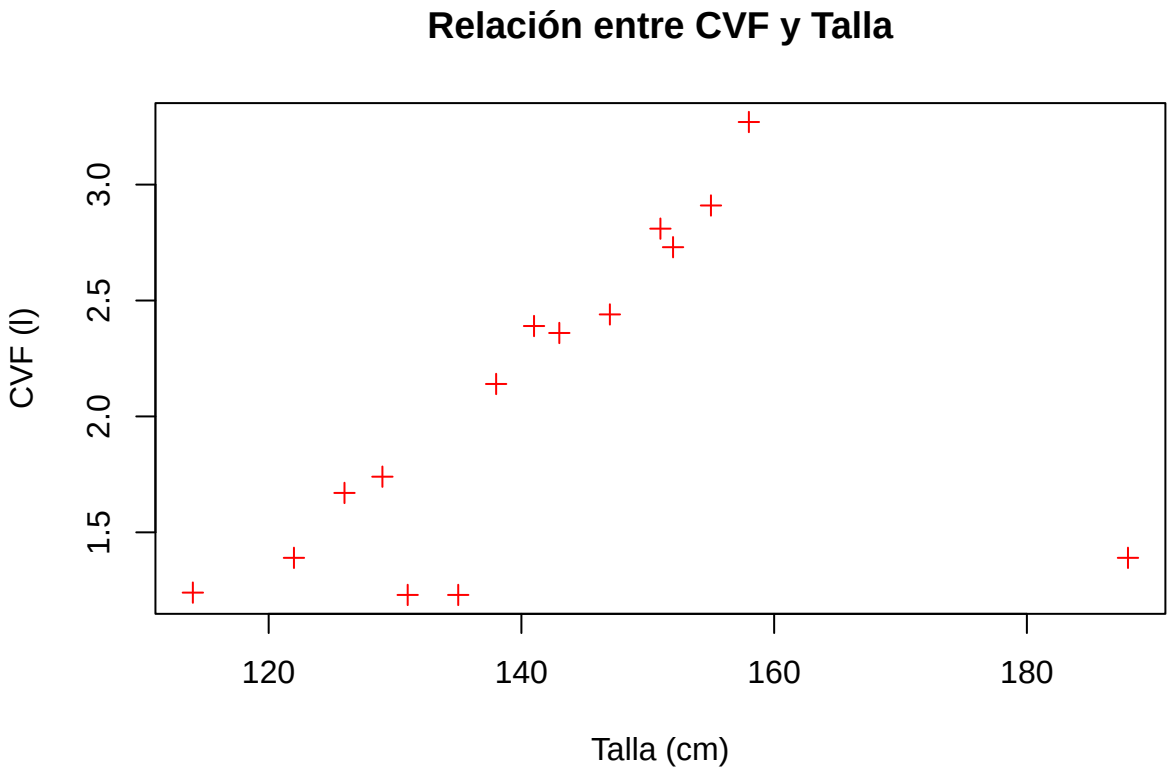
Si queremos visualizar la relación entre ambas variables mediante un diagrama de dispersión haciendo uso de la función **plot()**, la sentencia sería la siguiente:

```
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*.

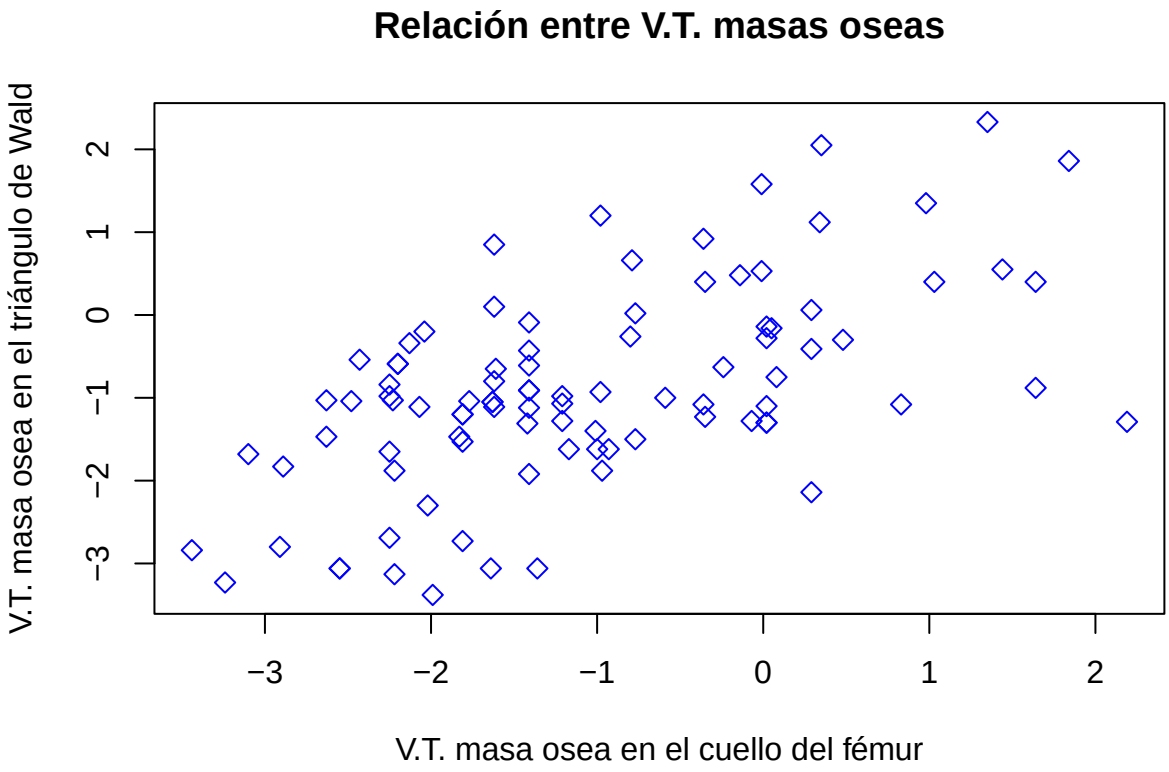
```
plot(x=datos$talla, y=datos$cvf, main = "Relación entre CVF y Talla",
     xlab ="Talla (cm)" ,ylab = "CVF (l)",pch=3,col="red" )
```



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 osea en el cuello del fémur y la variable *sztri* valor tipificado de la masa osea en el triángulo de Wald y representar un diagrama de dispersión como se muestra en el gráfico.

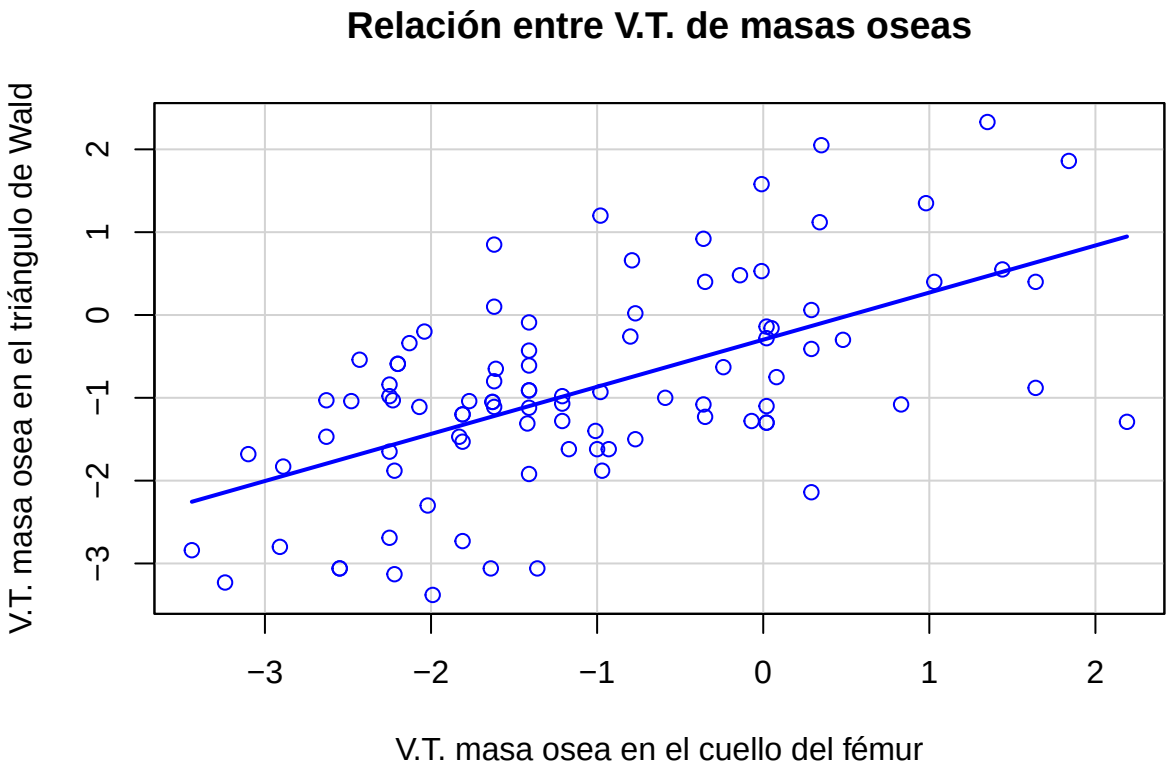
```
library(foreign)
osteo<-read.spss("~/Desktop/osteo.sav",to.data.frame=TRUE)
```

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



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*.

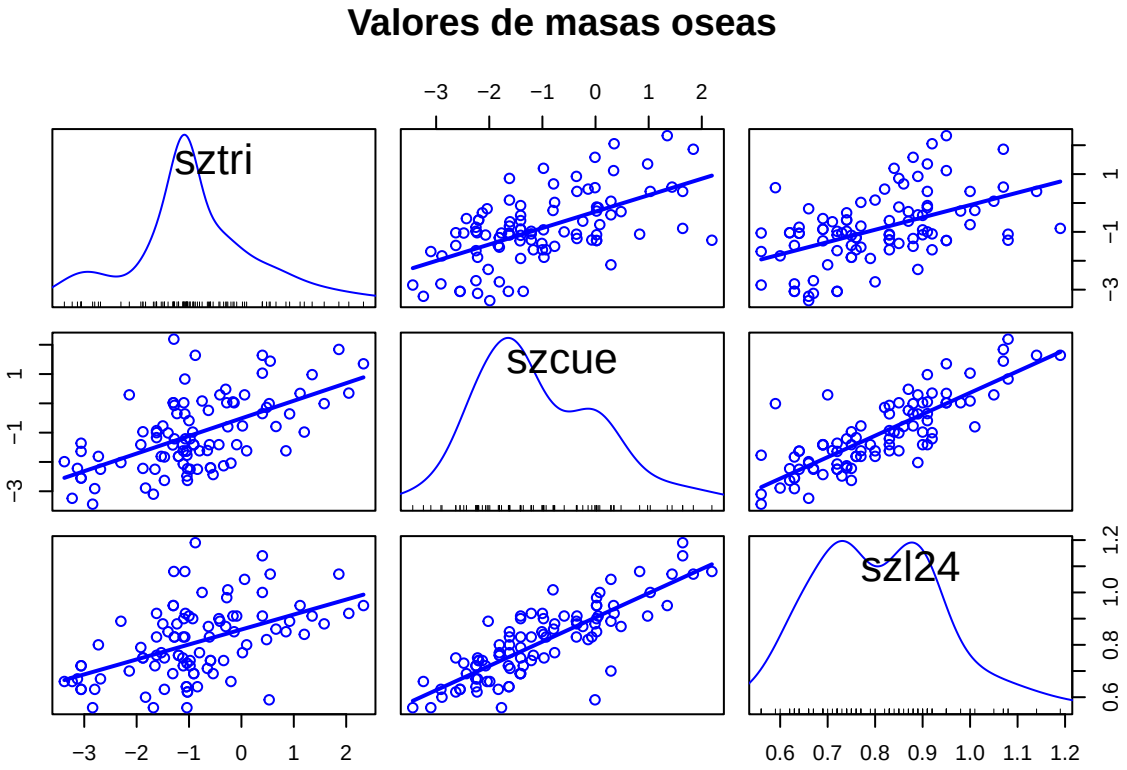
```
library(car)
scatterplot(sztri ~ szcue, data=osteo, xlab="V.T. masa osea en el cuello del fémur",
          ylab="V.T. masa osea en el triángulo de Wald",
          main="Relación entre V.T. de masas oseas",smooth=FALSE,boxplots="",)
```



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=" "*.

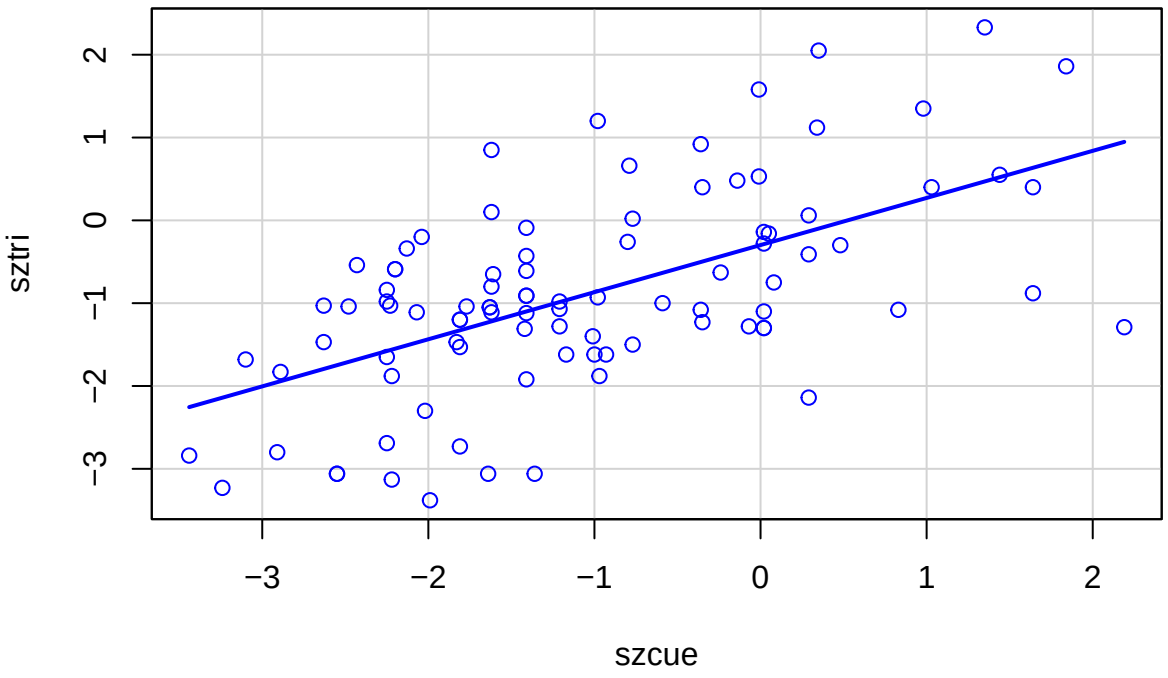
Este paquete también nos permite realizar gráficos matriciales de dispersión a través de la función **scatterplotMatrix()**, añadiendo ~ y después las variables que se pretenden analizar.

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

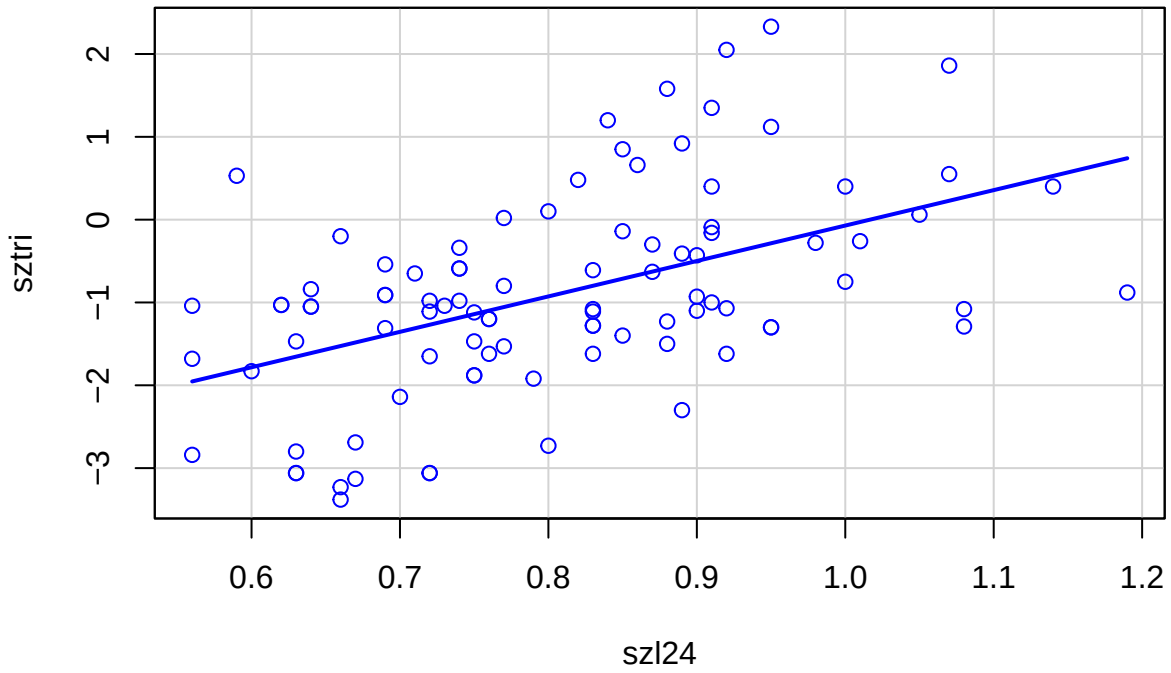


Cómo se puede observar en el gráfico, las diagonales muestran las distribuciones de frecuencias de cada variable y los gráficos son simétricos. Los diagramas representados por separado son:

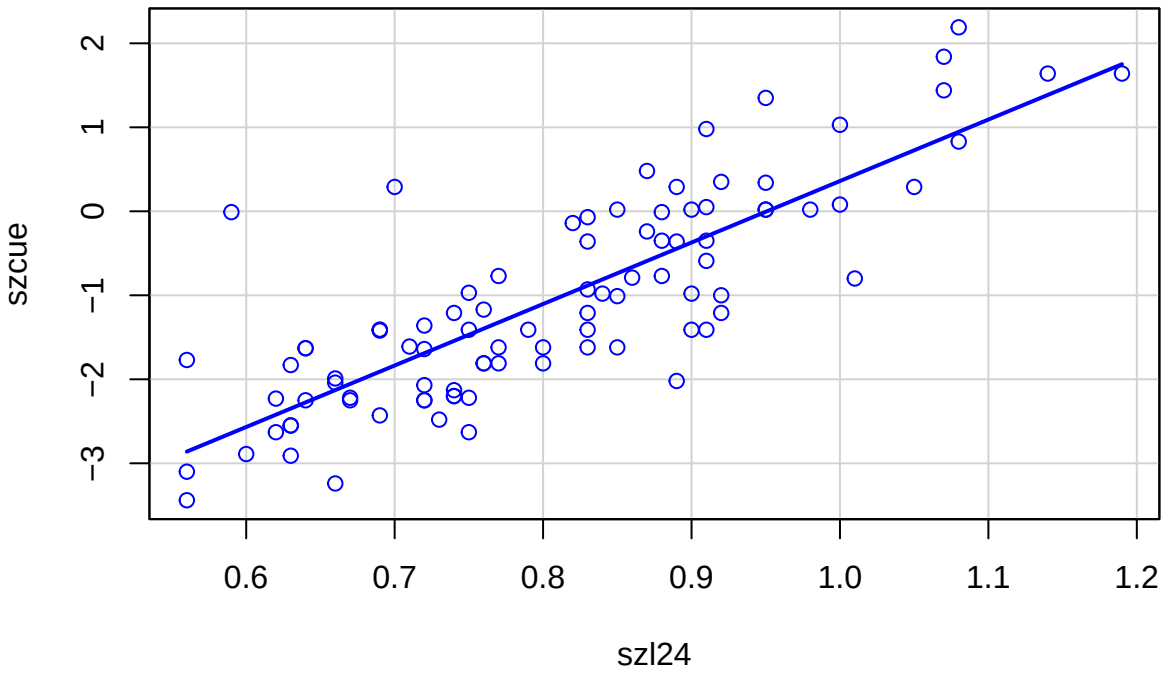
```
scatterplot(sztri ~ szcue, data=osteo,smooth=FALSE,boxplots="")
```




```
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 salida 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 ~. 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 aletaorio y otro determinista. En este contexto, lineal y simple, su forma generalizada será:

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

donde β_0 es el término independiente, β_1 es la pendiente y ε 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 multiplcada 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 descritiva, 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:

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

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 variablidad explicada del modelo con respecto al total de la varibilidad.

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 obervaciones 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 rechanzando la hipótesis nula.

El modelo de regresión lineal simple será:

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

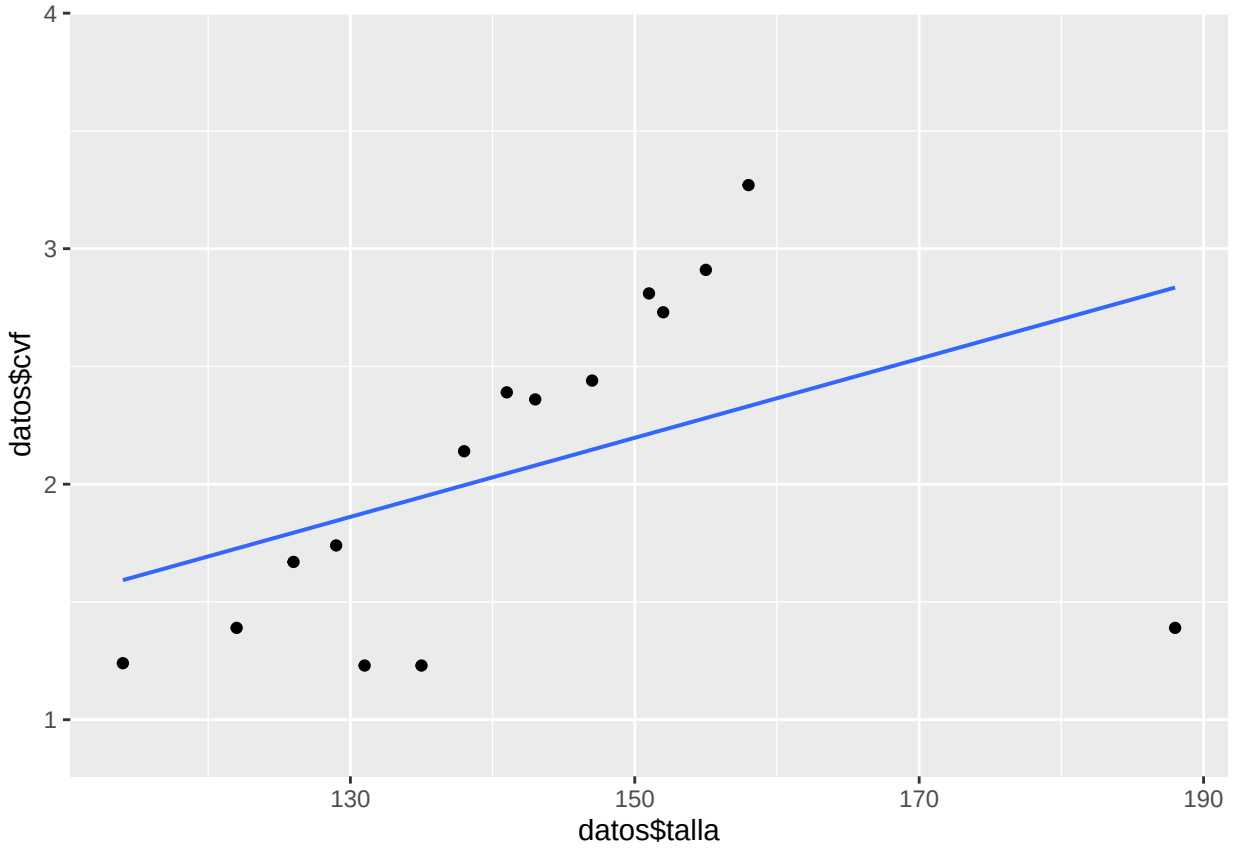
O bien,

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

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

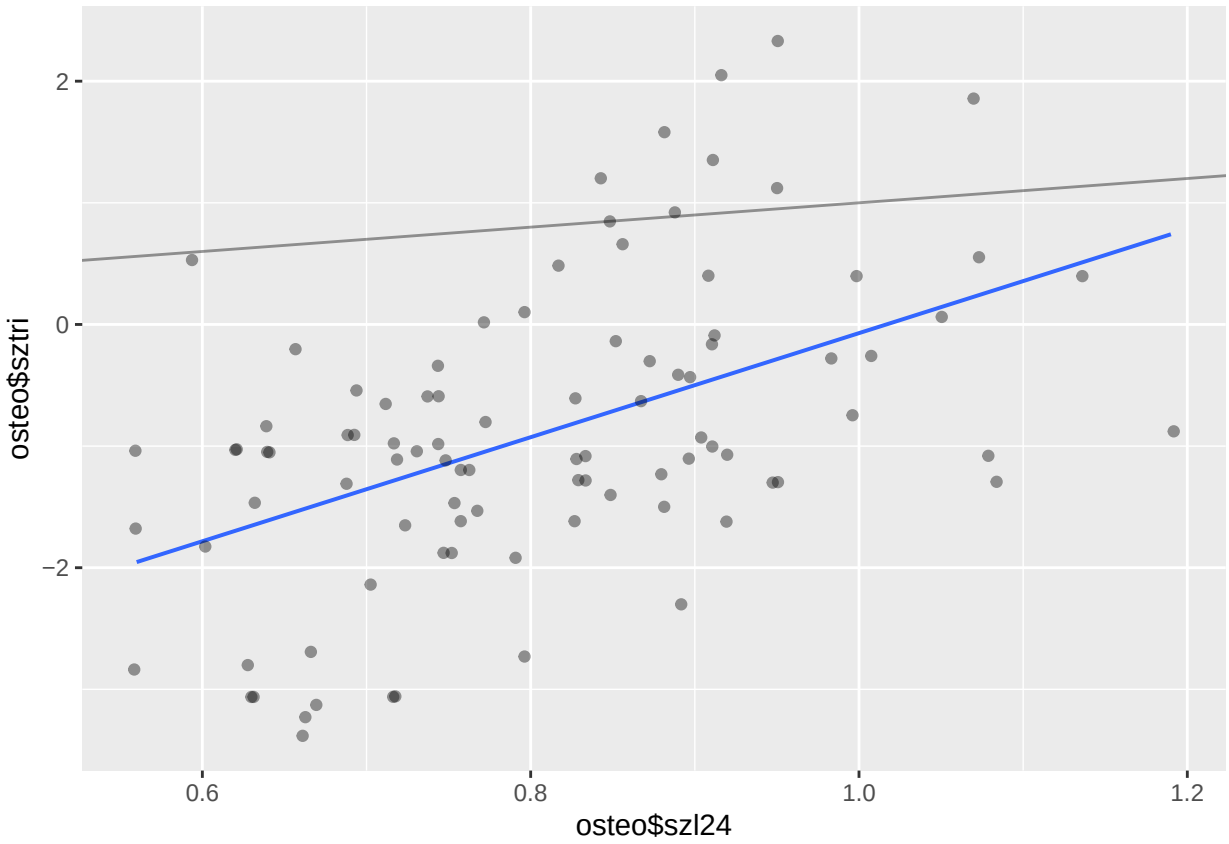
Otra forma para realizar el ajuste de regresión lineal simple puede ser mediante la función **gf_lm()** una vez representados lo datos con **gf_point()**.

```
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_aline()**:

```
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)
```

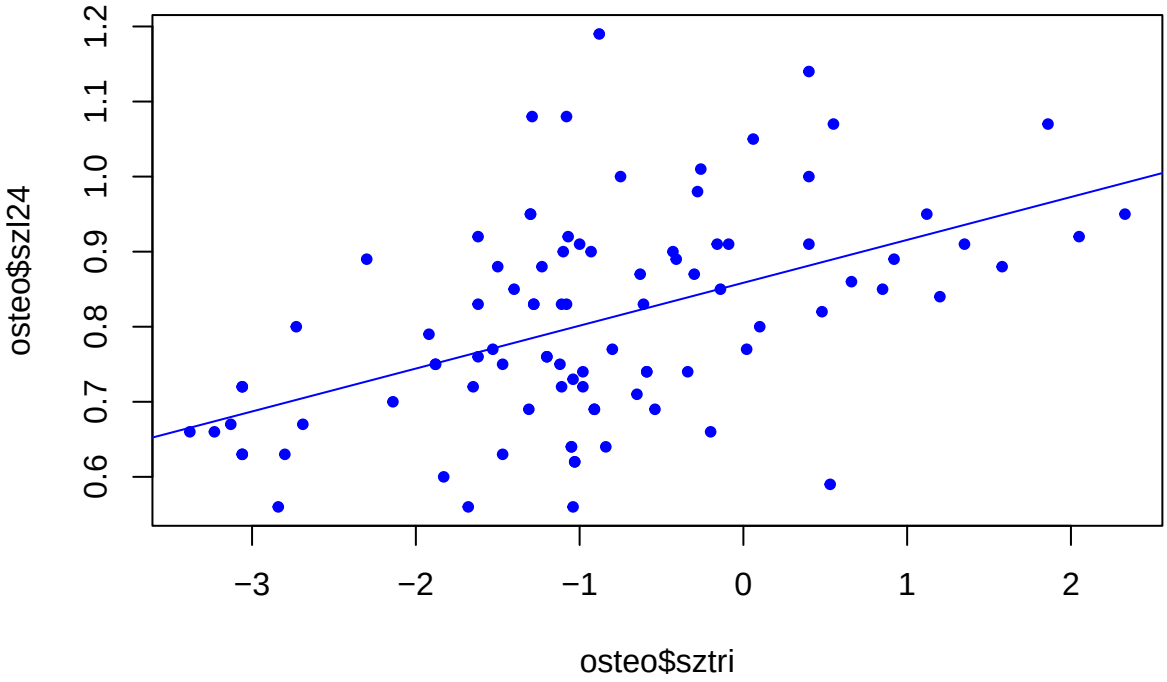
```
##          1          2          3          4          5          6          7
## -0.3525698 -1.4449686 -0.3368832 -0.1240399 -0.1044074 -0.6479858 -0.7151425
##          8          9         10         11         12         13         14
##  0.1444900  0.3441225  0.2805442  0.2933875  0.5962308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

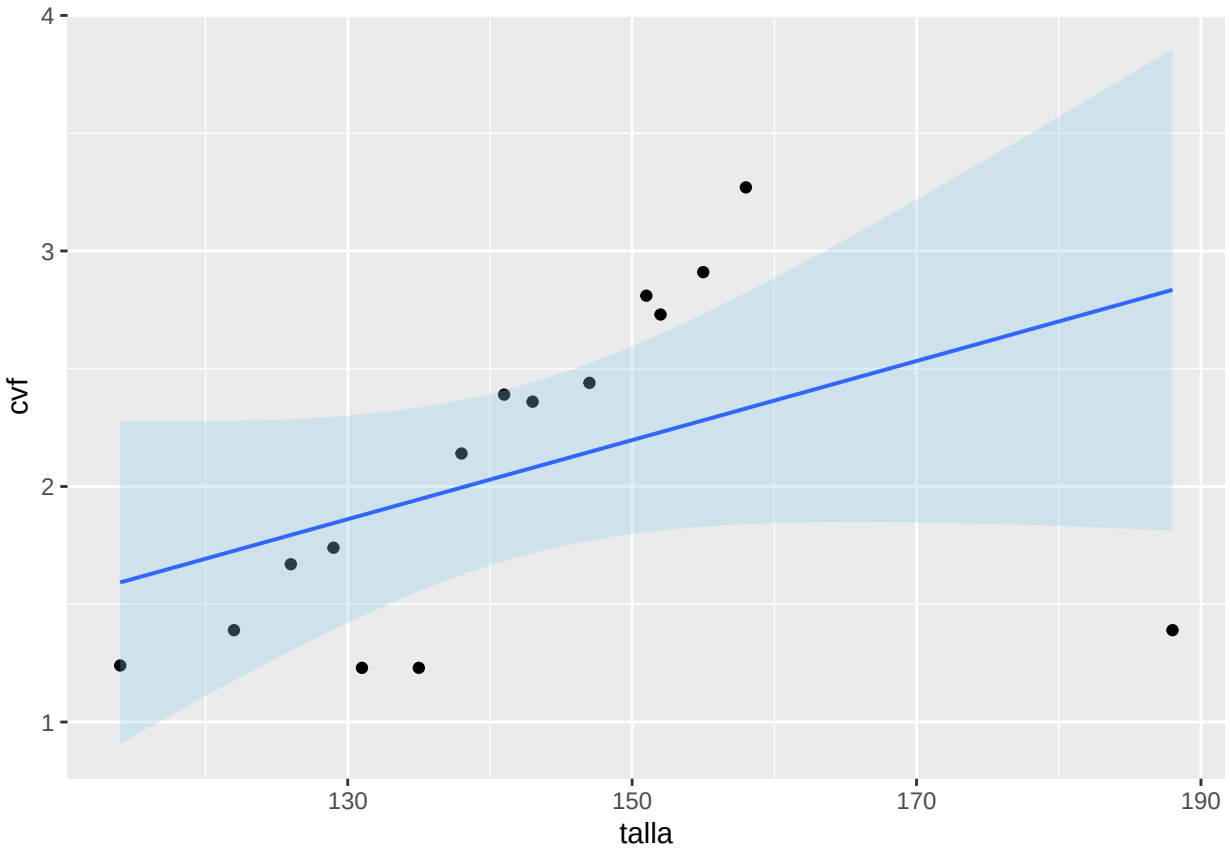
Podríamos indicar la línea de ajuste en el diagrama de dispersión también gracias a la función **abline()** y así observar los resultados:

```
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 la 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 osea, 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 estaadístico 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 $n \leqslant 30$ se compara $|\rho_s|$ con la tabla r_α. Pero en el progama 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
```

10.4. Inferencias sobre el modelo de Regresión Lineal Simple

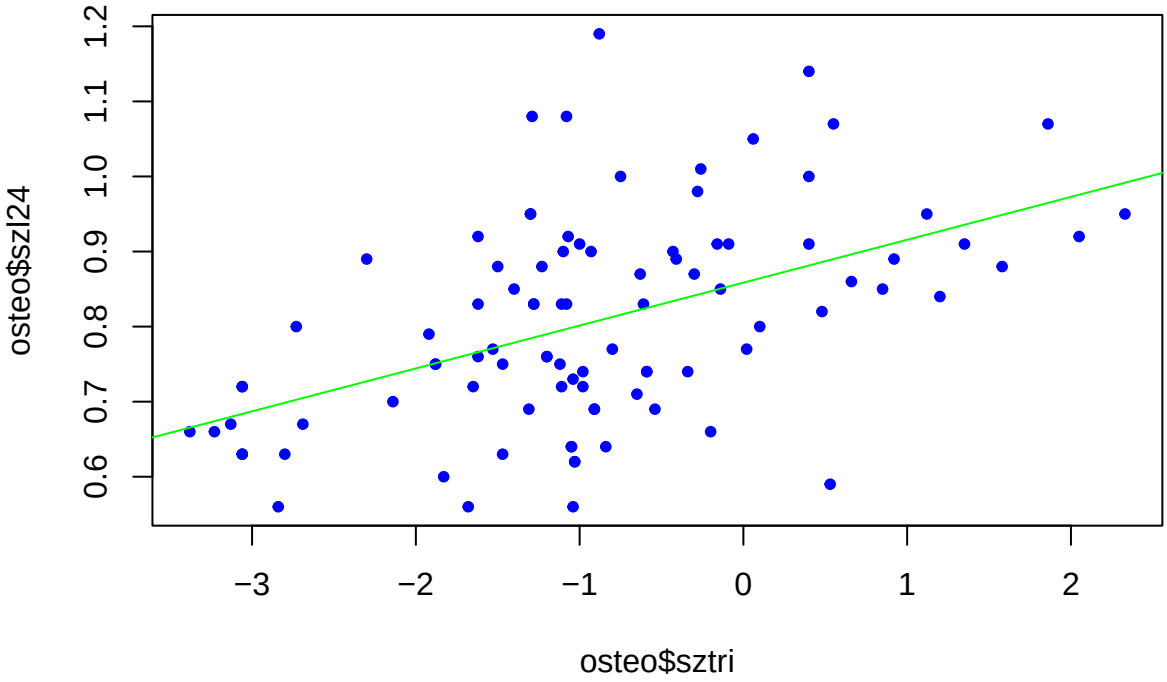
Si queremos predecir el valor de la variable respuesta a partir de la variable explivativa, tenemos que tener en cuenta las hipótesis del modelo de Regresión Lineal Simple (Normalidad, Homocedasticidad y Linealidad) y una vez comprobadas, en el rango de obtención de los datos podremos calcular un valor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

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

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

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:

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

Se podría hacer rápidamente con calculadora pero es muy útil para casos más complejos o múltiples en R eligiendo los coeficentes 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
```

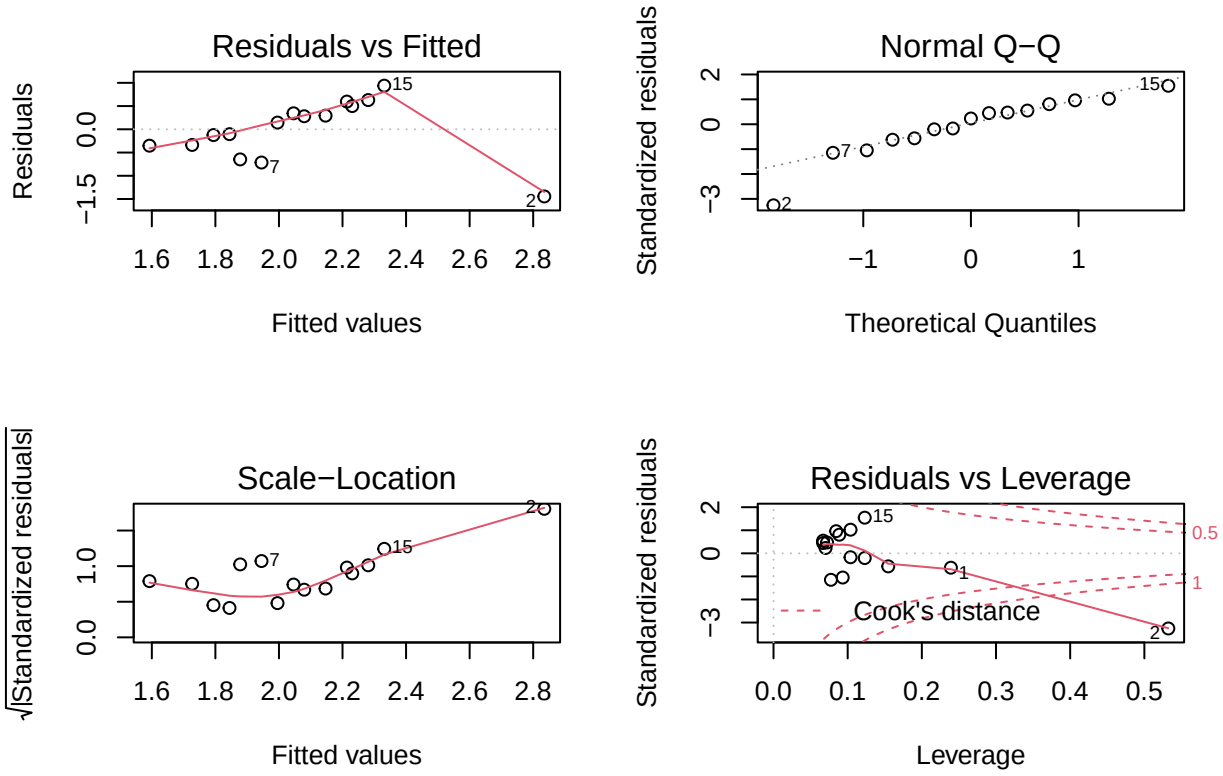
```
## [1] 0.8748322
```

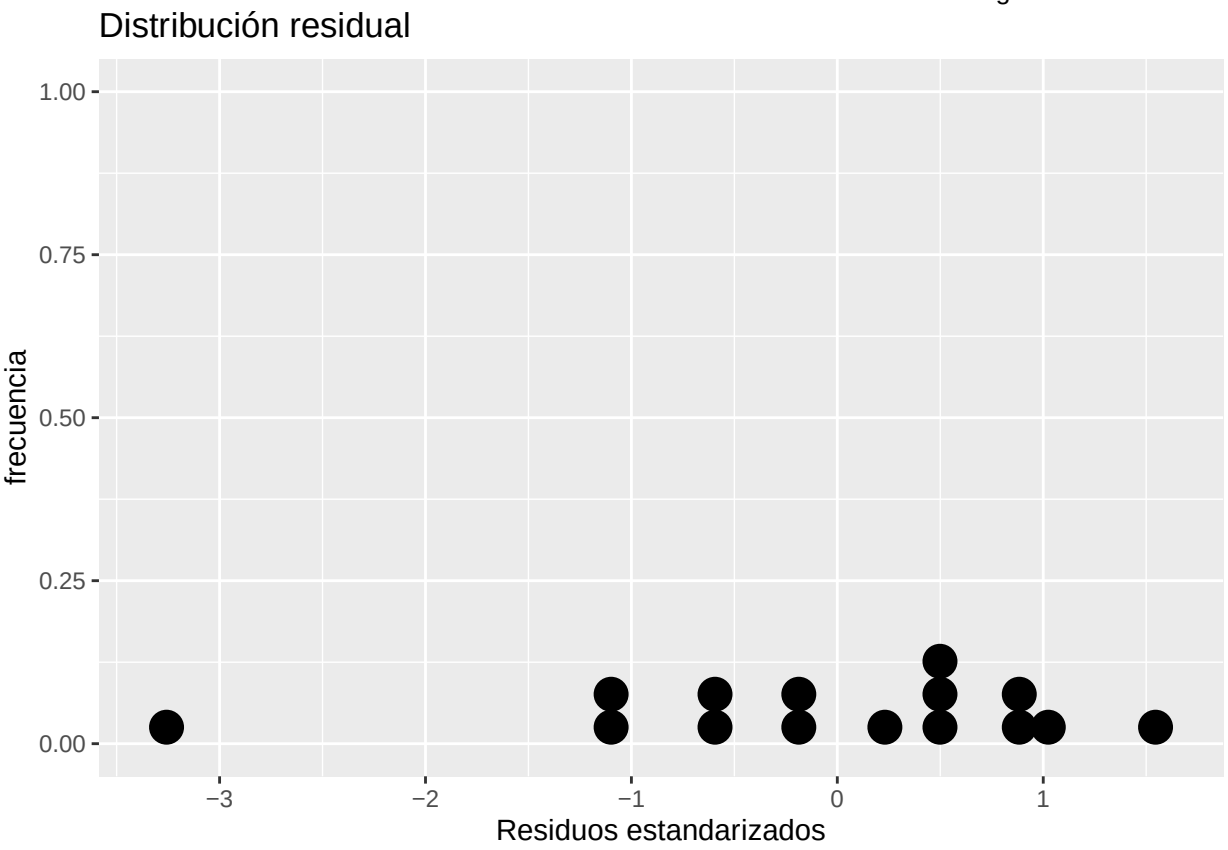
10.5. Regresión Lineal Simple con BioestadísticaR

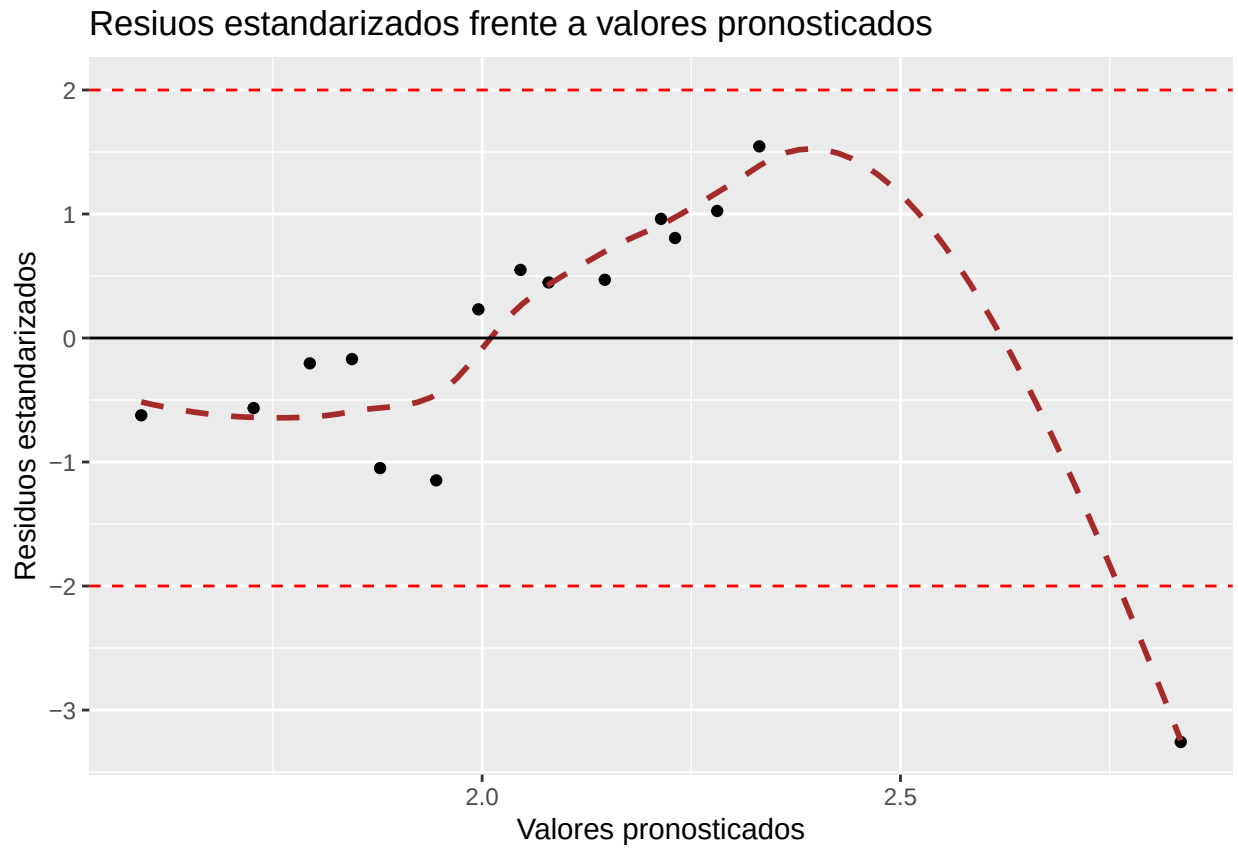
El paquete **BioestadísticaR** ha sido desarrollado con la función *rls()* para calcular el contraste de linealidad. Proponemos realizar el primer ejemplo de este guión y analizar los resultados que se generan a continuación.

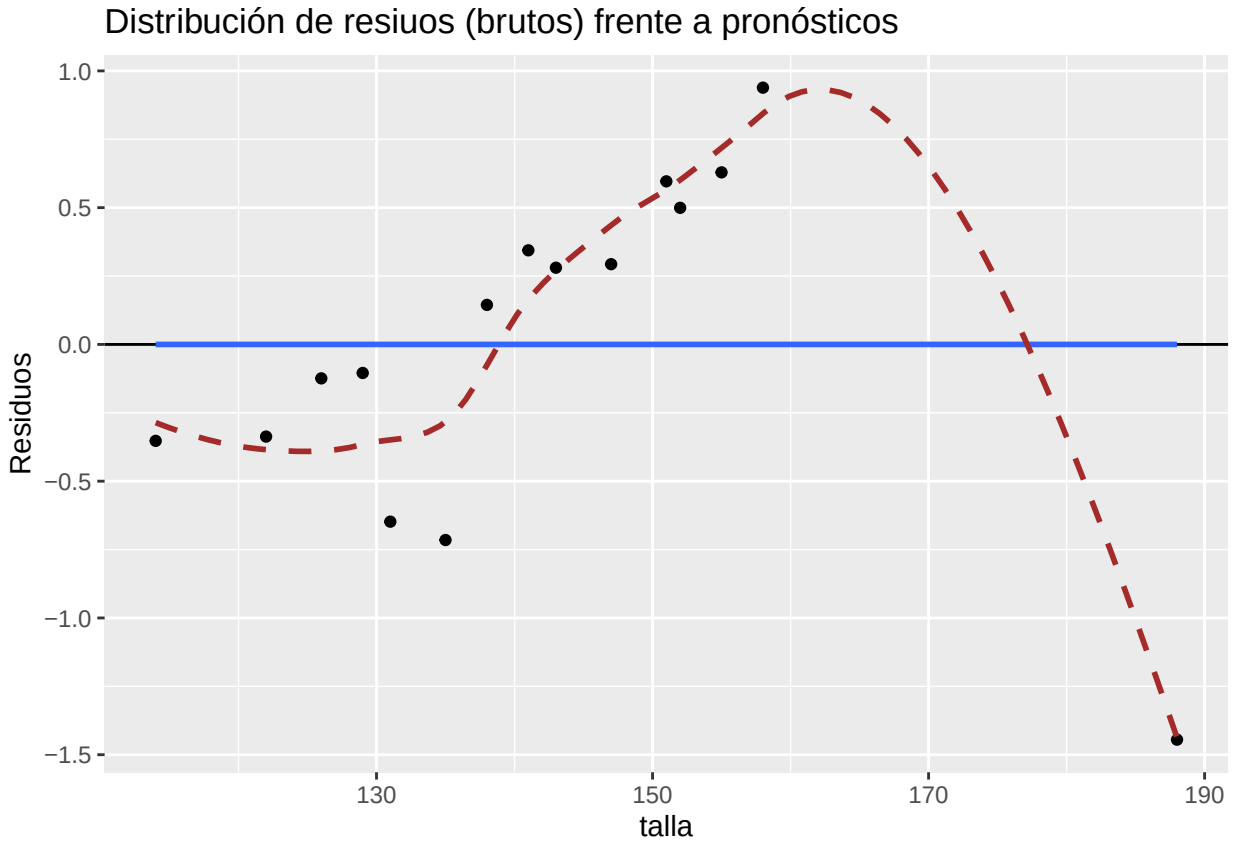
```
library(BioestadisticaR2)
caso<-c(1:15)
talla<-c(114,188,122,126,129,131,135,138,141,143,147,151,152,155,158)
cvf<-c(1.24,1.39,1.39,1.67,1.74,1.23,1.23,2.14,2.39,2.36, 2.44,2.81,2.73,2.91,3.27)
datos<-data.frame(caso,talla,cvf)

rls(cvf~talla, data=datos)
```

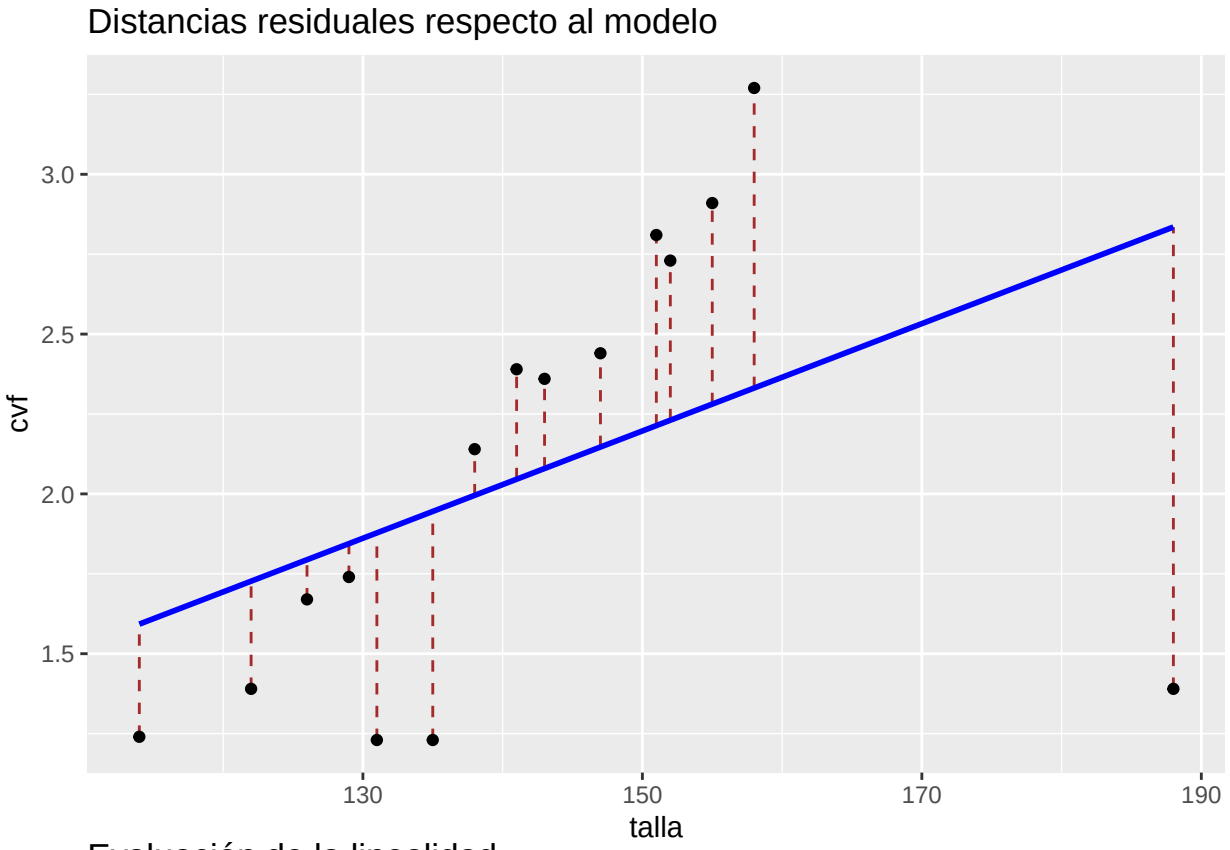
```
##
## Regresión lineal simple
## ---------------------------------------------------------------------
## # Información muestral ---
##
##   variable  n   media     dt    Min    Max Rango
## 1      cvf 15   2.063  0.694   1.23   3.27  2.04
## 2    talla 15 142.000 18.016 114.00 188.00 74.00
##
##   Cov(cvf,talla) = 5.449
##
## # Correlación de Pearson ---
##
##       r IC_inf IC_sup gl  texp     sig
##   0.436 -0.099  0.775 13 1.745 = 0.105
##
## # Modelo lineal ---
##
##   Modelo:  cvf ~ talla
##   R² =  0.19
##   S²residual =  0.421
##   Coeficientes del modelo:
##
##          Coef  estim    se ic_inf ic_sup  texp   sig
## 1 (Constante) -0.321 1.377 -3.295  2.653 0.233 0.819
## 2       talla  0.017 0.010 -0.004  0.038 1.745 0.105
##
## # Distribución residual ---
##   Error estándar residual:  0.649
##        res   zres
## min -1.445 -3.258
## Q1  -0.345 -0.594
## Q2   0.144  0.231
## Q3   0.422  0.678
## max  0.939  1.545
##
##   Test de normalidad residual (Shapiro-Wilk):
##   w =0.957, p= 0.643
```

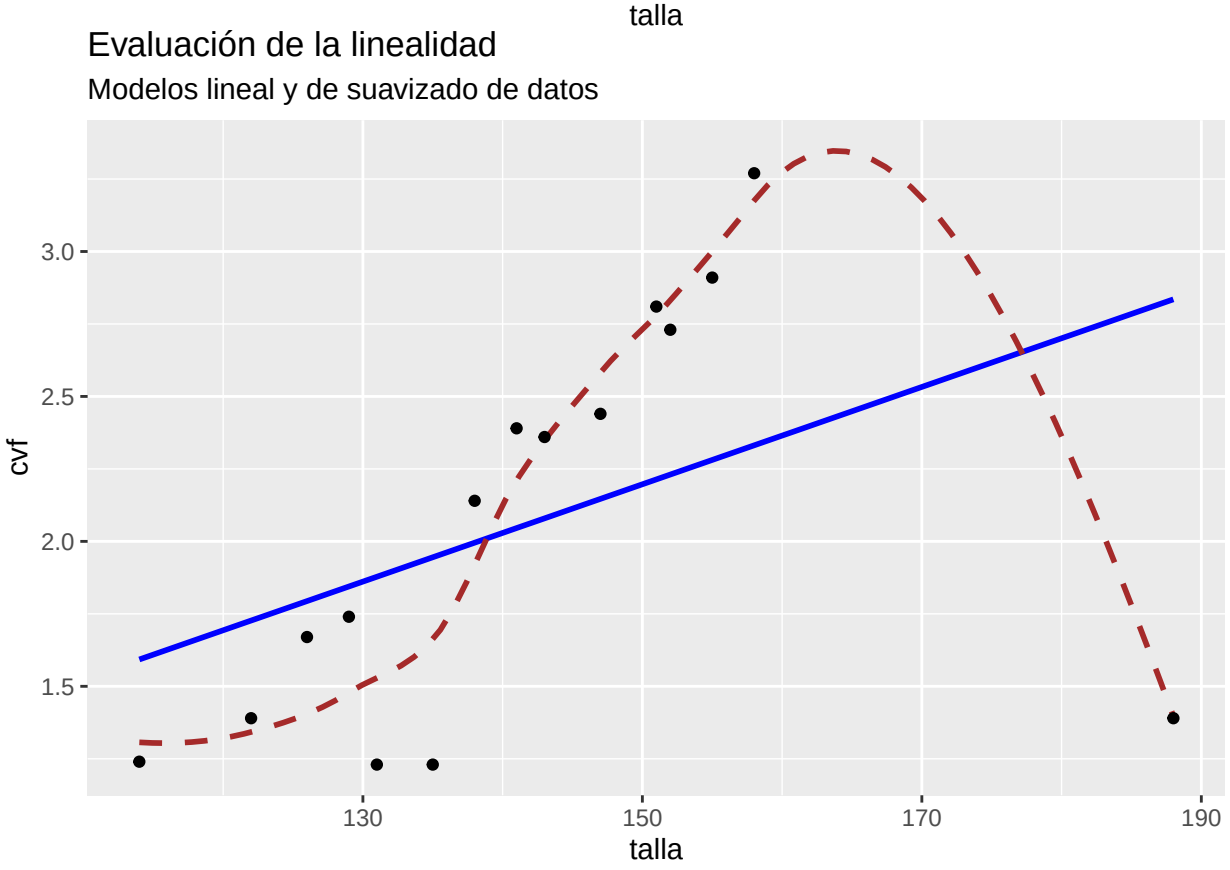

Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

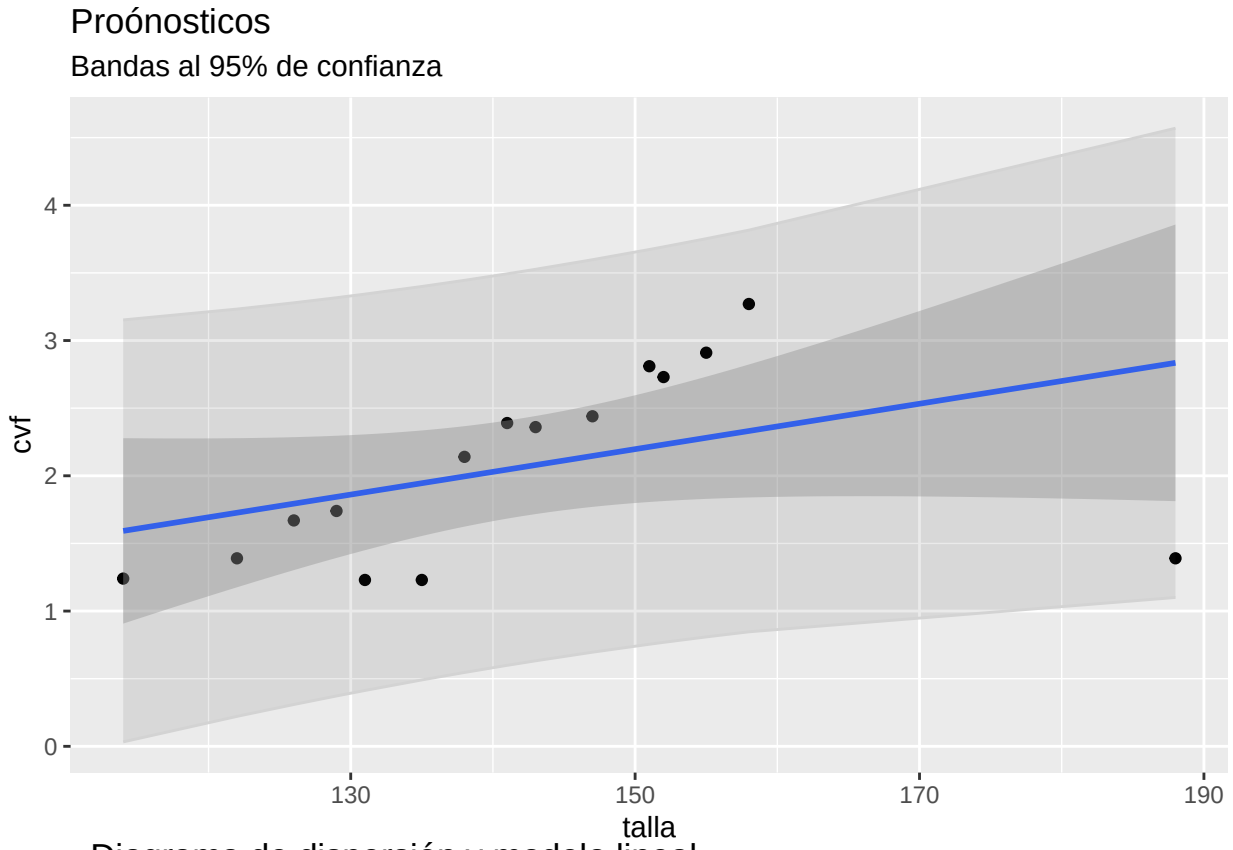

Distribución residual
frecuencia
Residuos estandarizados


Resiuos estandarizados frente a valores pronosticados
Residuos estandarizados
2
1
0
−1
−2
−3
2.0
2.5
Valores pronosticados


Distribución de resiuos (brutos) frente a pronósticos
Residuos
1.0
0.5
0.0
−0.5
−1.0
−1.5
130
150
170
190
talla


Distancias residuales respecto al modelo
3.0
2.5
2.0
1.5
cvf
130
150
170
190
talla


Evaluación de la linealidad
Modelos lineal y de suavizado de datos
3.0
2.5
2.0
1.5
cvf
130
150
170
190
talla


Proónosticos
Bandas al 95% de confianza
cvf
talla
0
1
2
3
4
130
150
170
190

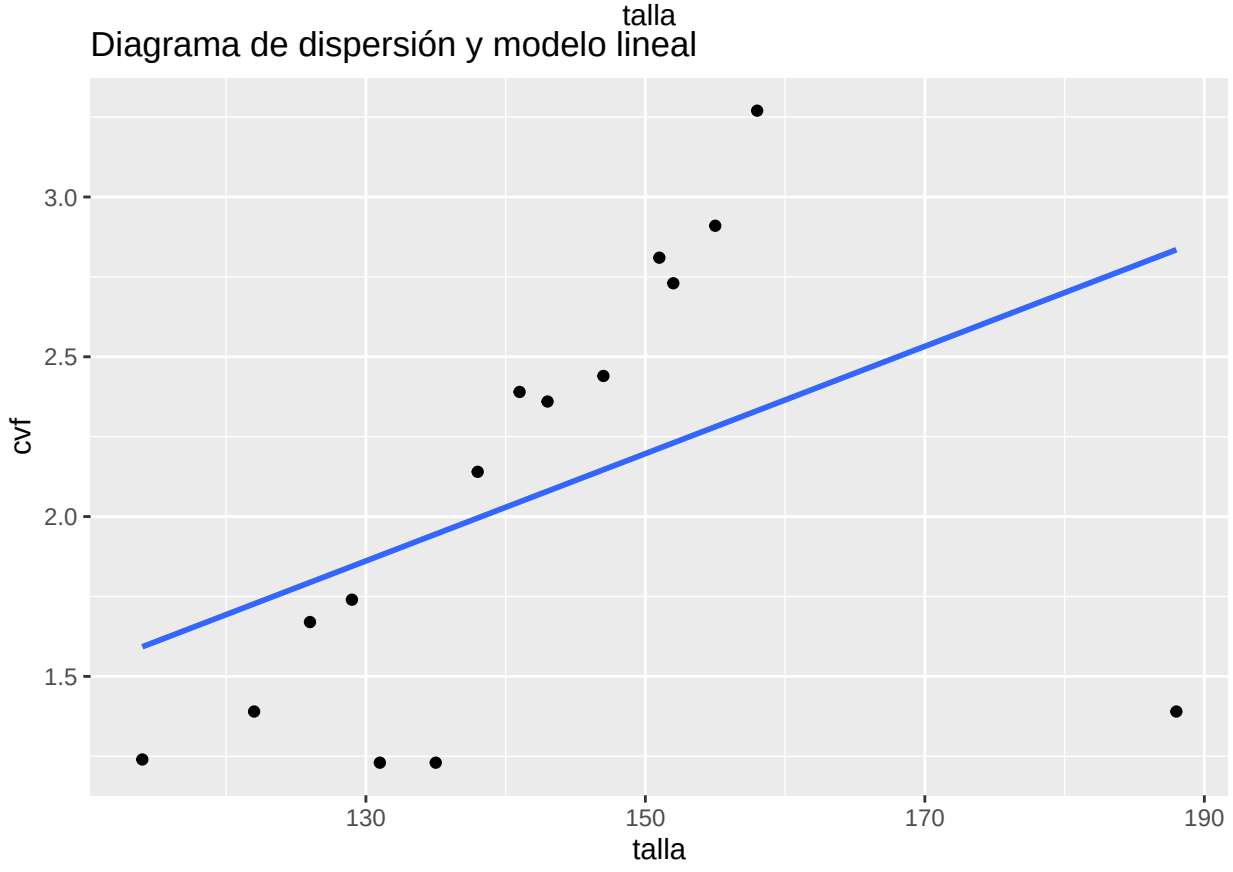
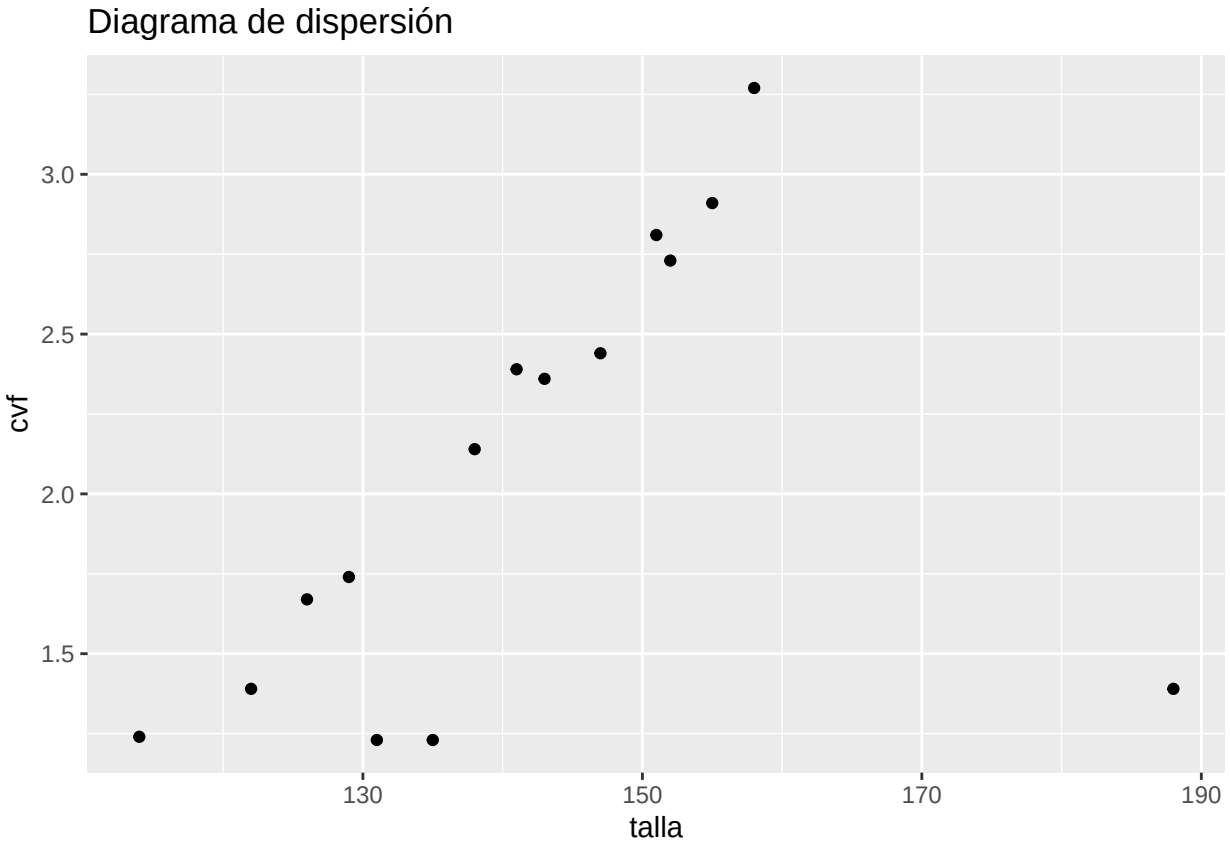


Diagrama de dispersión y modelo lineal
cvf
talla
1.5
2.0
2.5
3.0
130
150
170
190



La entrada de la fórmula puede implementarse a través de la sentencia previa *rls(cvf~talla, data=datos)* o bien, mediante el nombre de las variables consecutivamente: *rls(cvf,talla)*.

El resultado nos ofrece tanto el contraste para la pendiente de la recta de regresión como para el coeficiente de correlación lineal. En cualquier caso, obtenemos tanto los p-valores como los intervalos de confianza y todos los gráficos que explican el ajuste para esos datos a la recta.

Además, podemos calcular un valor predictivo como respuesta del modelo a partir de un valor dado por ejemplo 135cm de talla:

```
rls(cvf~talla, data=datos,pred=135,grf=FALSE)
```

```
##
## Regresión lineal simple
## ---------------------------------------------------------------
## # Información muestral ---
##
##   variable  n   media     dt    Min    Max Rango
## 1      cvf 15   2.063  0.694   1.23   3.27  2.04
## 2    talla 15 142.000 18.016 114.00 188.00 74.00
##
##   Cov(cvf,talla) = 5.449
##
## # Correlación de Pearson ---
##
##       r IC_inf IC_sup gl  texp     sig
##   0.436 -0.099  0.775 13 1.745 = 0.105
##
```

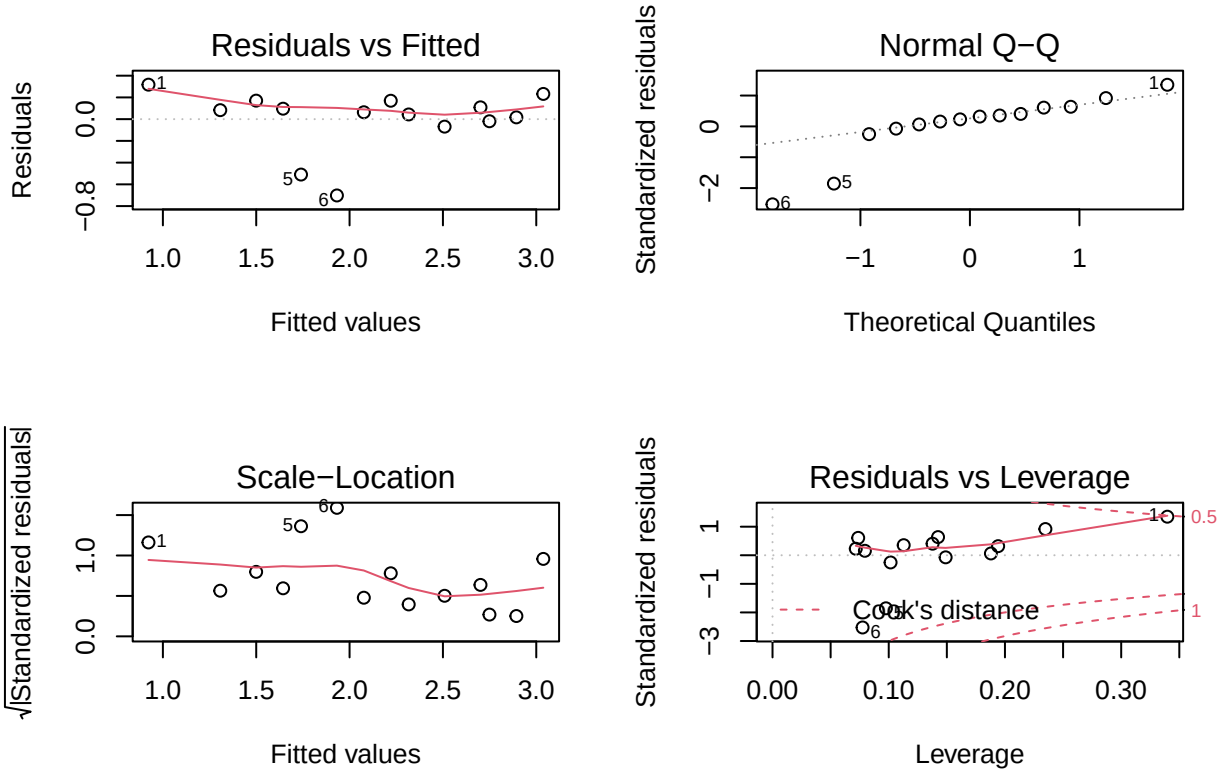
```
## # Modelo lineal ---
##
##   Modelo:  cvf ~ talla
##   R² =  0.19
##   S²residual =  0.421
##   Coeficientes del modelo:
##
##          Coef  estim    se ic_inf ic_sup  texp   sig
## 1 (Constante) -0.321 1.377 -3.295  2.653 0.233 0.819
## 2       talla  0.017 0.010 -0.004  0.038 1.745 0.105
##
## # Pronósticos con el modelo ---
##   Pronosticos puntuales y bandas al 95 % de confianza para
##   promedios IC(m), y para una nueva observación: IC(obs)
##
##   Predictor  Puntual IC(m)_inf IC(m)_sup IC(obs)_inf IC(obs)_sup
## 1       135 1.945142  1.555173  2.335112   0.4906263    3.399659
##
## # Distribución residual ---
##   Error estándar residual:  0.649
##        res   zres
## min -1.445 -3.258
## Q1  -0.345 -0.594
## Q2   0.144  0.231
## Q3   0.422  0.678
## max  0.939  1.545
##
##   Test de normalidad residual (Shapiro-Wilk):
##   w =0.957, p= 0.643
```

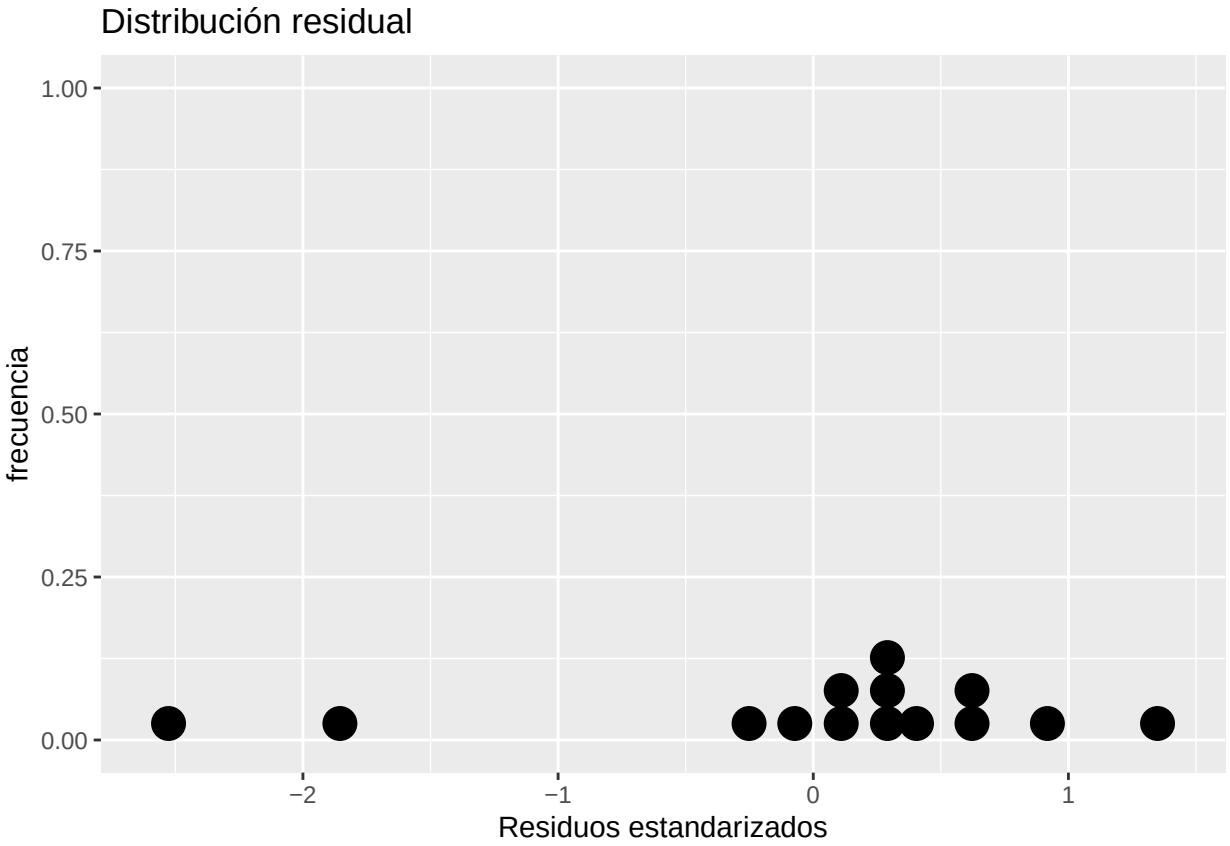
Eliminando la observación el punto discordante en la muestra, podemos recalcular los resultados del contraste para la pendiente de regresión lineal simple como sigue:

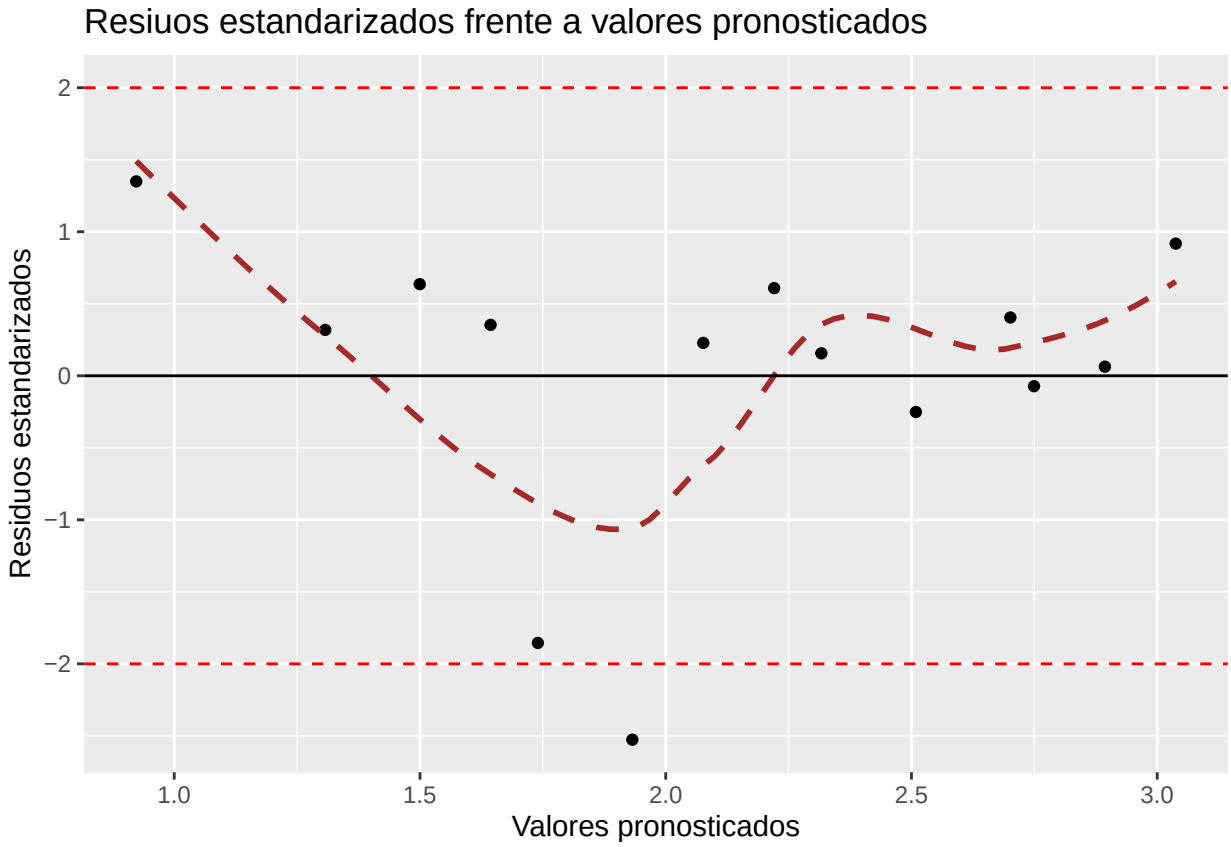
```
talla2= c(114, 122, 126, 129, 131, 135, 138, 141, 143, 147, 151,152, 155, 158)
cvf2= c(1.24, 1.39, 1.67, 1.74, 1.23,1.23,2.14, 2.39, 2.36, 2.44,2.81, 2.73, 2.91, 3.27)
datos2= data.frame(talla2,cvf2)
rls(cvf2 ~ talla2, data=datos2)
```

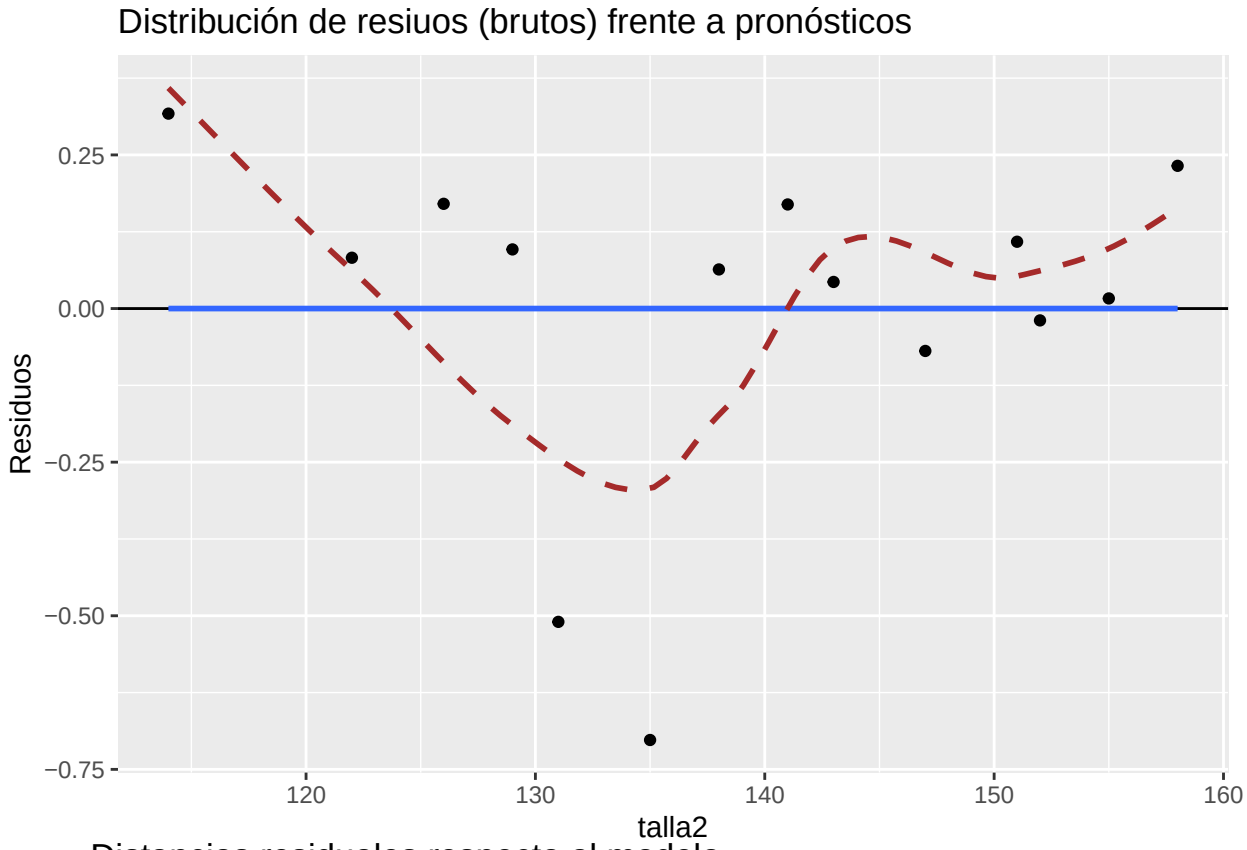
```
##
## Regresión lineal simple
## ---------------------------------------------------------------
## # Información muestral ---
##
##   variable  n   media     dt    Min    Max Rango
## 1     cvf2 14   2.111  0.694   1.23   3.27  2.04
## 2   talla2 14 138.714 13.234 114.00 158.00 44.00
##
##   Cov(cvf2,talla2) = 8.419
##
## # Correlación de Pearson ---
##
##       r IC_inf IC_sup gl  texp     sig
##   0.916  0.751  0.974 12 7.927 < 0.001
```

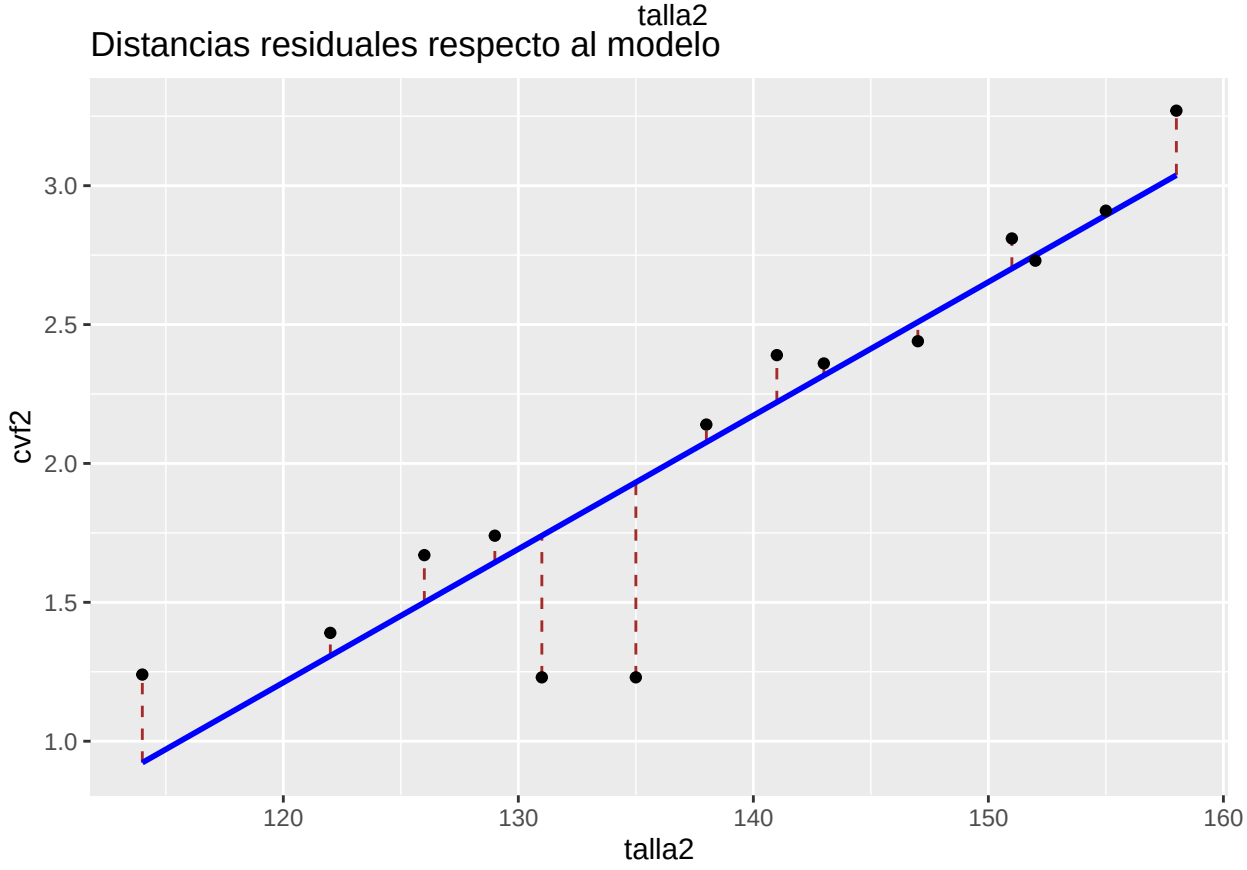
```
##
## # Modelo lineal ---
##
##   Modelo:  cvf2 ~ talla2
##   R² =  0.839
##   S²residual =  0.084
##   Coeficientes del modelo:
##
##          Coef  estim    se ic_inf ic_sup  texp    sig
## 1 (Constante) -4.557 0.845 -6.397 -2.717 5.395 <0.001
## 2      talla2  0.048 0.006  0.035  0.061 7.927 <0.001
##
## # Distribución residual ---
##   Error estándar residual:  0.289
##        res   zres
## min -0.702 -2.527
## Q1  -0.010 -0.039
## Q2   0.073  0.273
## Q3   0.154  0.557
## max  0.317  1.349
##
##   Test de normalidad residual (Shapiro-Wilk):
##   w =0.789, p= 0.004
```

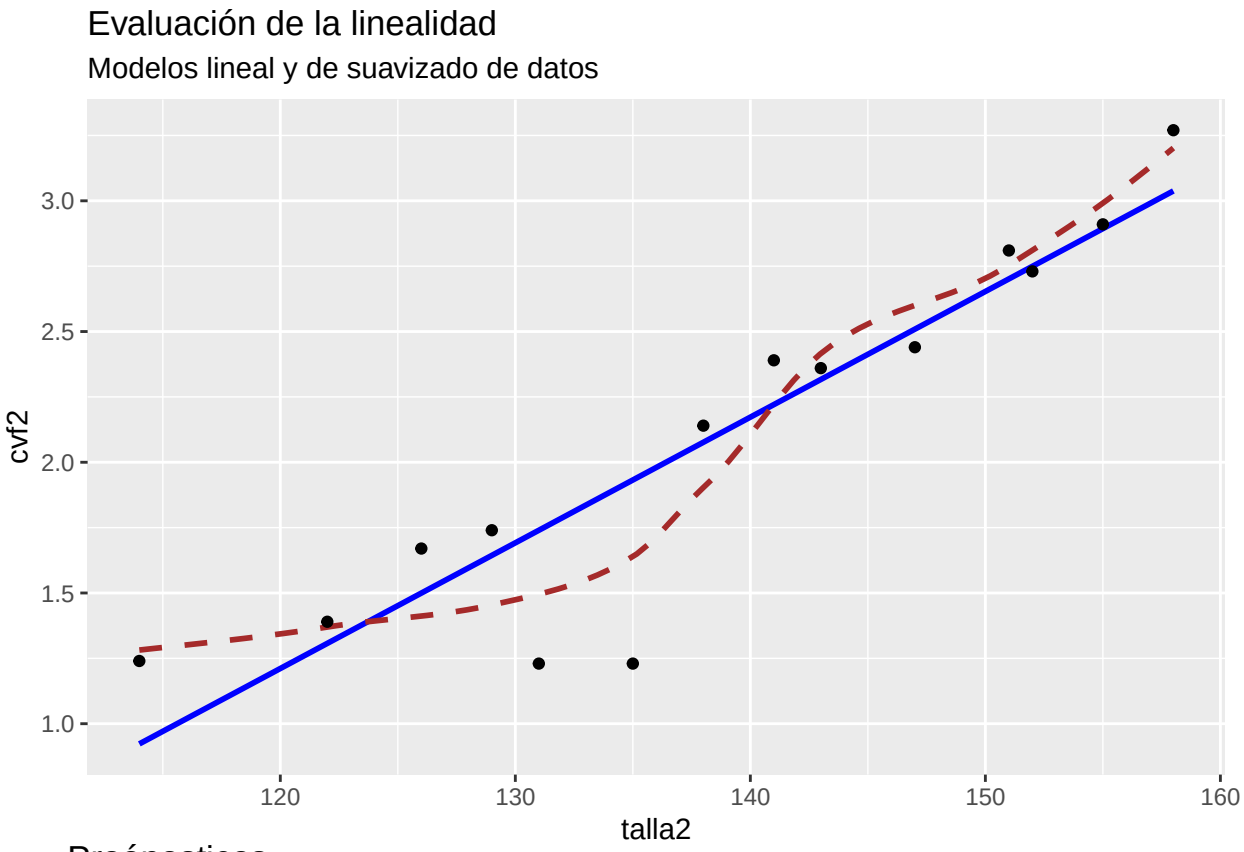


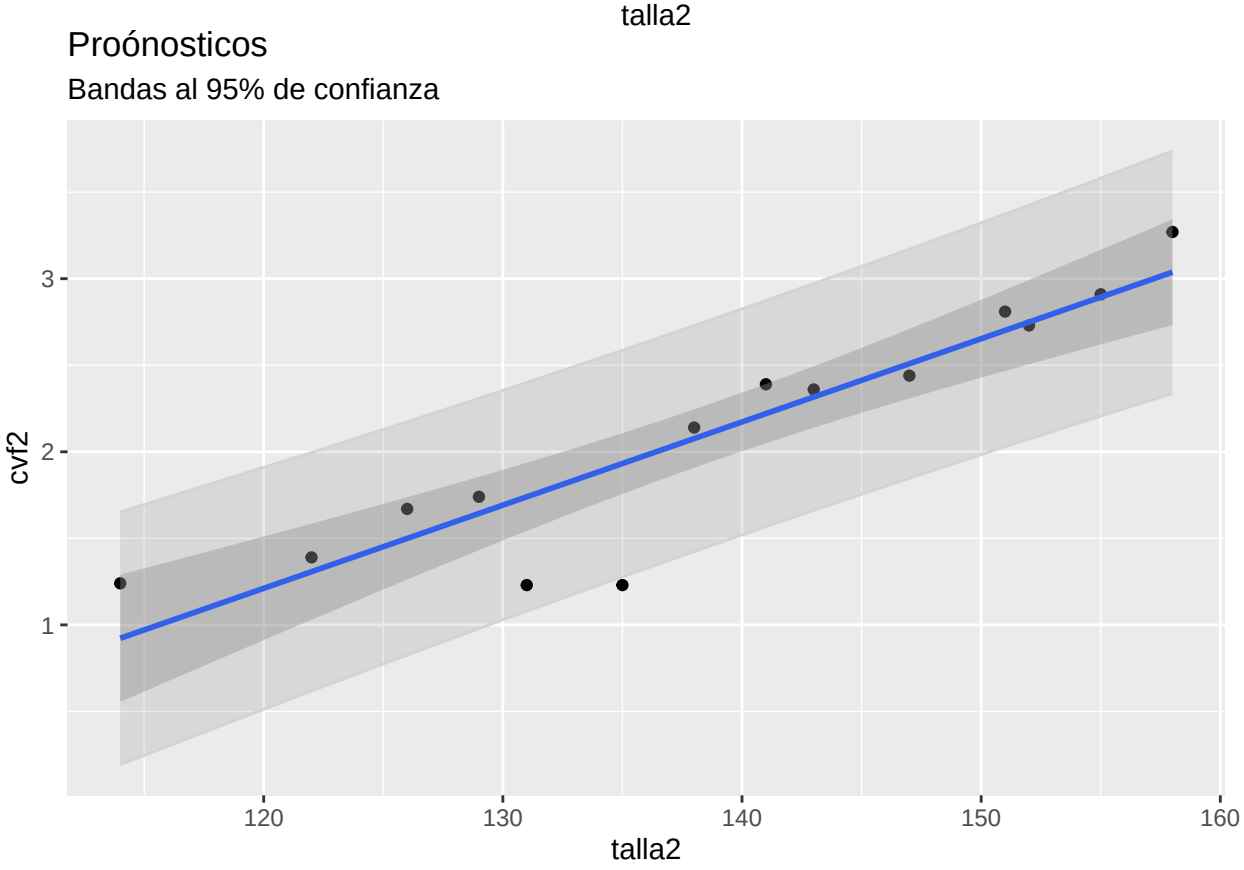

Distribución residual
1.00
0.75
0.50
0.25
0.00
frecuencia
−2
−1
0
1
Residuos estandarizados

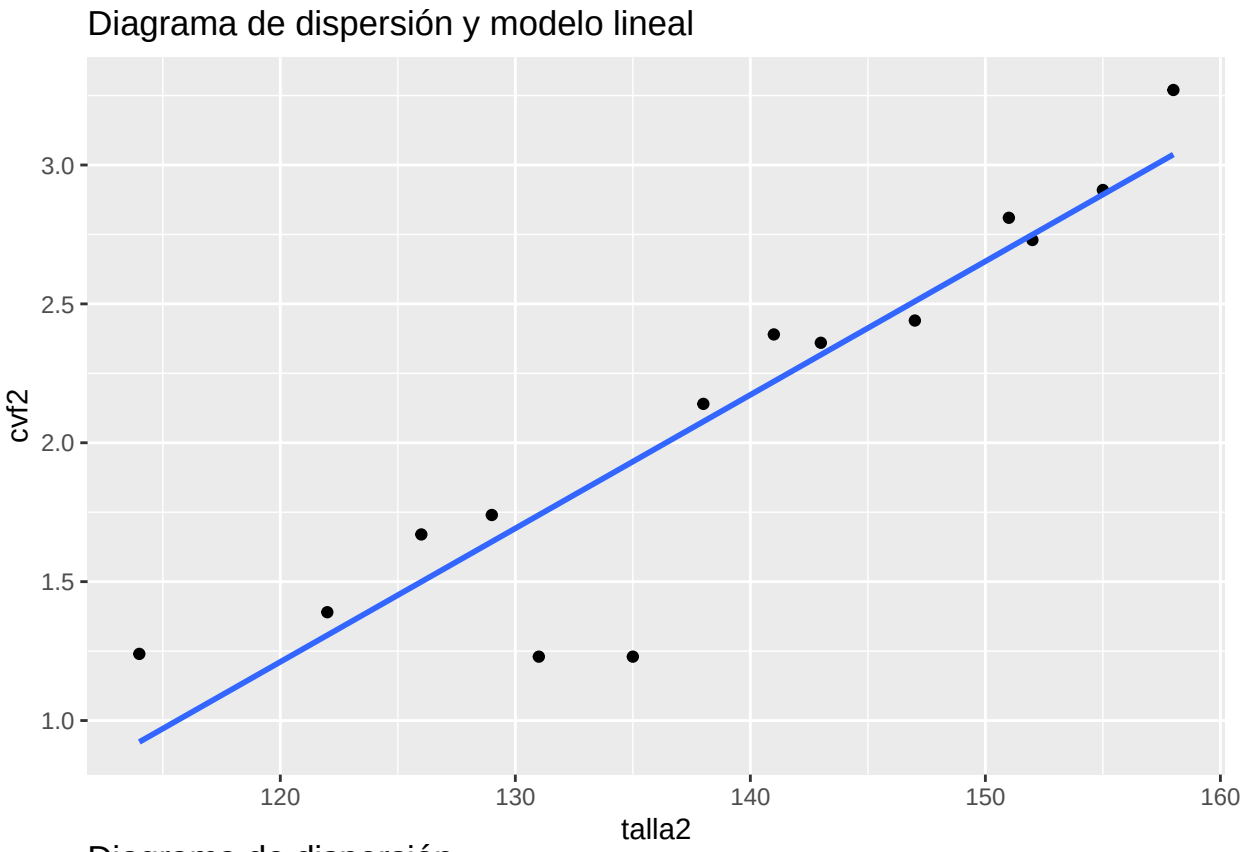

Resiuos estandarizados frente a valores pronosticados
2
1
0
−1
−2
Residuos estandarizados
1.0
1.5
2.0
2.5
3.0
Valores pronosticados

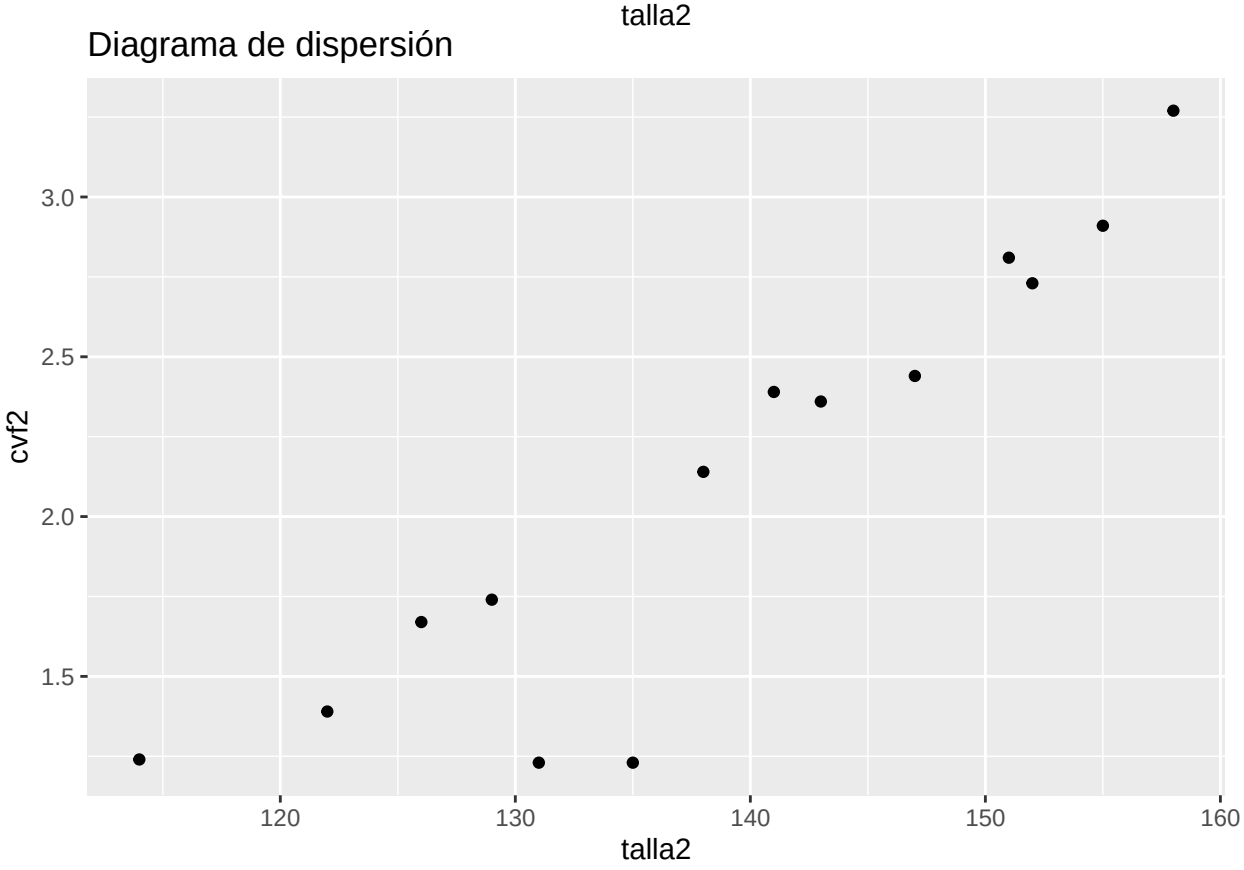

Distribución de resiuos (brutos) frente a pronósticos
Residuos
0.25
0.00
−0.25
−0.50
−0.75
120
130
140
150
160
talla2


Distancias residuales respecto al modelo
cvf2
3.0
2.5
2.0
1.5
1.0
120
130
140
150
160
talla2


Evaluación de la linealidad
Modelos lineal y de suavizado de datos
cvf2
talla2
3.0
2.5
2.0
1.5
1.0
120
130
140
150
160


Proónosticos
Bandas al 95% de confianza
cvf2
talla2
3
2
1
120
130
140
150
160





De igual forma que antes, podemos establecer un valor dado para predecir un valor predicho por modelo:

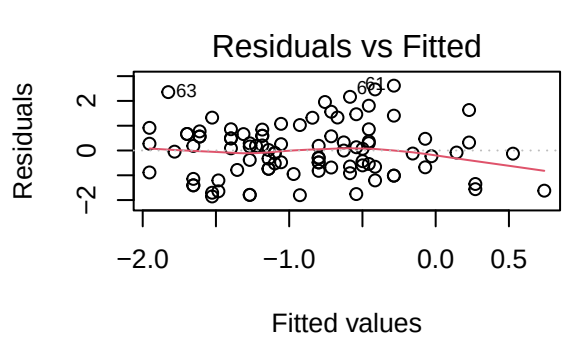
```
rls(cvf2~talla2, data=datos2,pred=135,grf=FALSE)
```

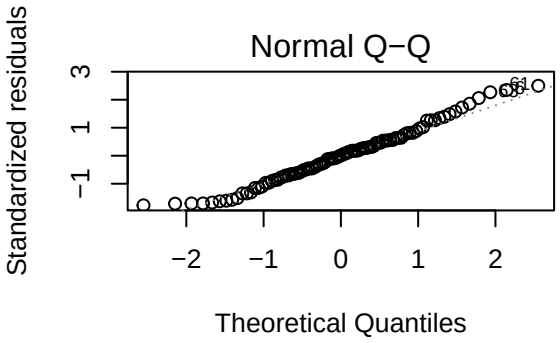
```
##
## Regresión lineal simple
## ---------------------------------------------------------------
## # Información muestral ---
##
##   variable  n   media     dt    Min    Max Rango
## 1     cvf2 14   2.111  0.694   1.23   3.27  2.04
## 2   talla2 14 138.714 13.234 114.00 158.00 44.00
##
##   Cov(cvf2,talla2) = 8.419
##
## # Correlación de Pearson ---
##
##       r IC_inf IC_sup gl  texp     sig
##   0.916  0.751  0.974 12 7.927 < 0.001
##
## # Modelo lineal ---
##
##   Modelo:  cvf2 ~ talla2
##   R² =  0.839
##   S²residual =  0.084
##   Coeficientes del modelo:
##
##          Coef  estim    se ic_inf ic_sup  texp    sig
## 1 (Constante) -4.557 0.845 -6.397 -2.717 5.395 <0.001
## 2      talla2  0.048 0.006  0.035  0.061 7.927 <0.001
##
## # Pronósticos con el modelo ---
##   Pronosticos puntuales y bandas al 95 % de confianza para
##   promedios IC(m), y para una nueva observación: IC(obs)
##
##   Predictor  Puntual IC(m)_inf IC(m)_sup IC(obs)_inf IC(obs)_sup
## 1       135 1.932178  1.756688  2.107668    1.277779    2.586576
##
## # Distribución residual ---
##   Error estándar residual:  0.289
##        res   zres
## min -0.702 -2.527
## Q1  -0.010 -0.039
## Q2   0.073  0.273
## Q3   0.154  0.557
## max  0.317  1.349
##
##   Test de normalidad residual (Shapiro-Wilk):
##   w =0.789, p= 0.004
```

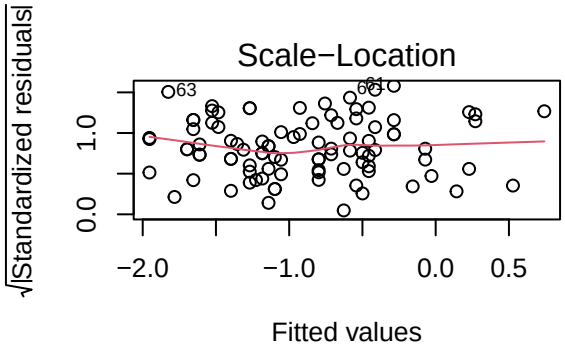
El paquete **BioesatdísticaR** además permite modificar el nivel de significación, la confianza, los decimales como con el resto de funciones, e incluso añadir los datos desde un data frame o una base de datos. Es interesante resaltar también, que el argumento *dfout=TRUE* nos devuelve una matriz de datos y gráficos con los valores residuales y pronósticados, muy útil para evaluar la bondad de ajuste del modelo.

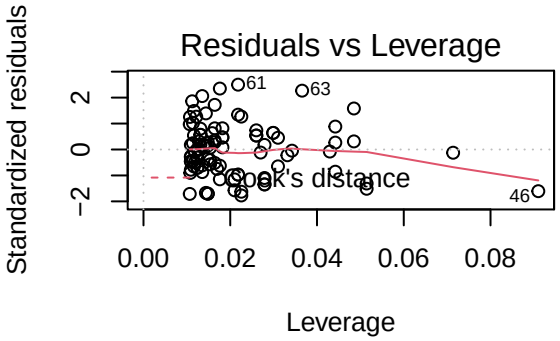
```r
rls(sztri~szl24, data=osteo,dfout=TRUE)
```

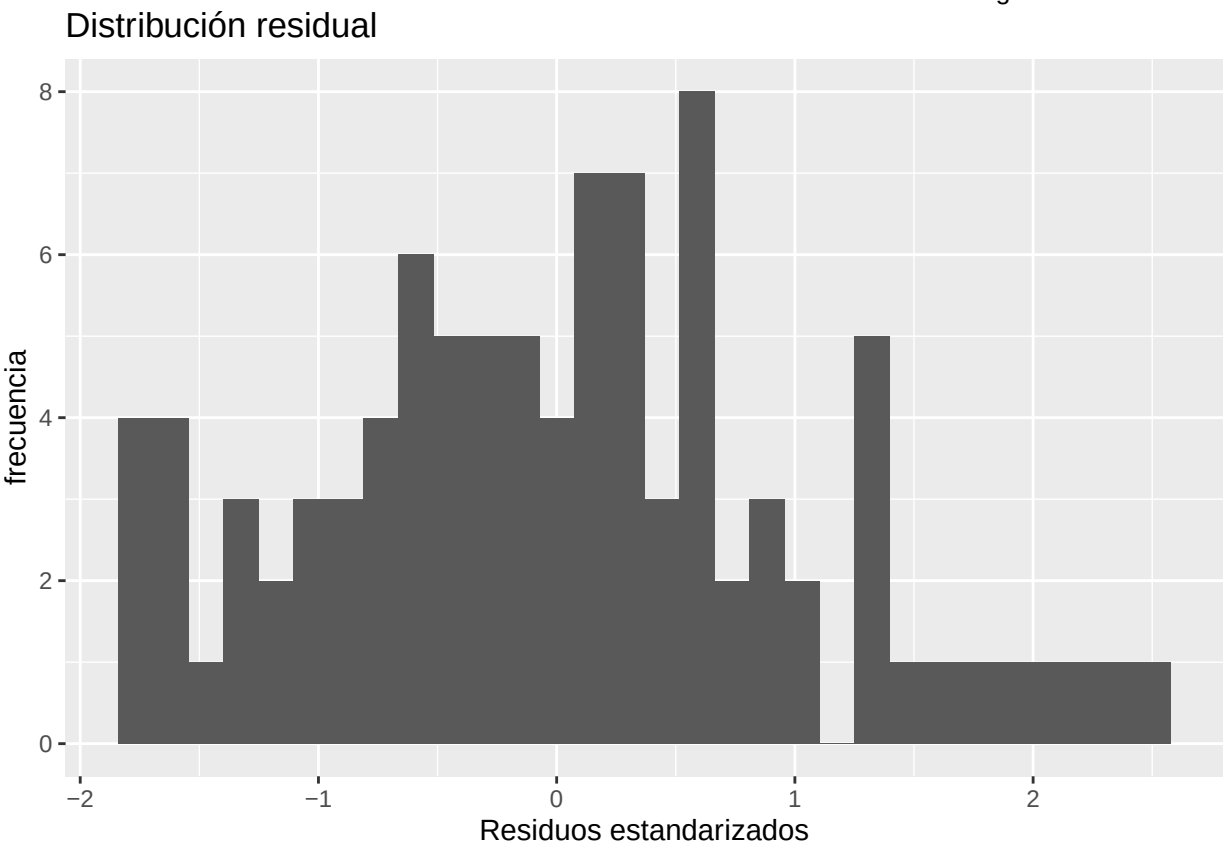
```
##
## Regresión lineal simple
## ---------------------------------------------------------------------
## # Información muestral ---
##
##   variable  n  media    dt   Min  Max Rango
## 1    sztri 94 -0.896 1.211 -3.38 2.33  5.71
## 2    szl24 94  0.807 0.140  0.56 1.19  0.63
##
##   Cov(sztri,szl24) = 0.084
##
## # Correlación de Pearson ---
##
##       r IC_inf IC_sup gl  texp     sig
##   0.494  0.324  0.634 92 5.456 < 0.001
##
## # Modelo lineal ---
##
##   Modelo:  sztri ~ szl24
##   R² =  0.244
##   S²residual =  1.12
##   Coeficientes del modelo:
##
##          Coef  estim    se ic_inf ic_sup  texp    sig
## 1 (Constante) -4.349 0.642 -5.625 -3.074 6.770 <0.001
## 2       szl24  4.278 0.784  2.721  5.835 5.456 <0.001
##
## # Distribución residual ---
##   Error estándar residual:  1.058
##        res   zres
## min -1.854 -1.772
## Q1  -0.726 -0.690
## Q2   0.009  0.009
## Q3   0.594  0.565
## max  2.615  2.498
##
##   Test de normalidad residual (Shapiro-Wilk):
##   w =0.981, p= 0.185
```

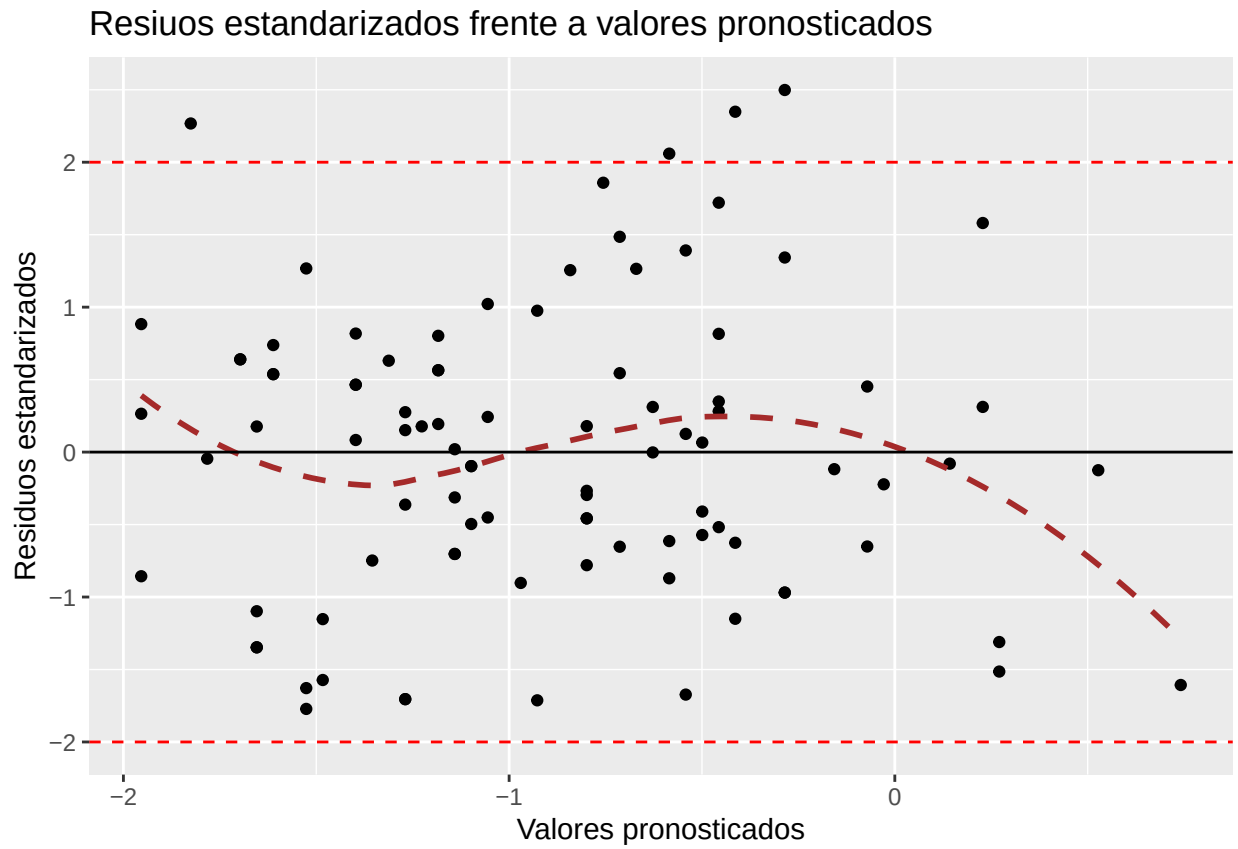

Residuals vs Fitted
Residuals
Fitted values
63
61
66

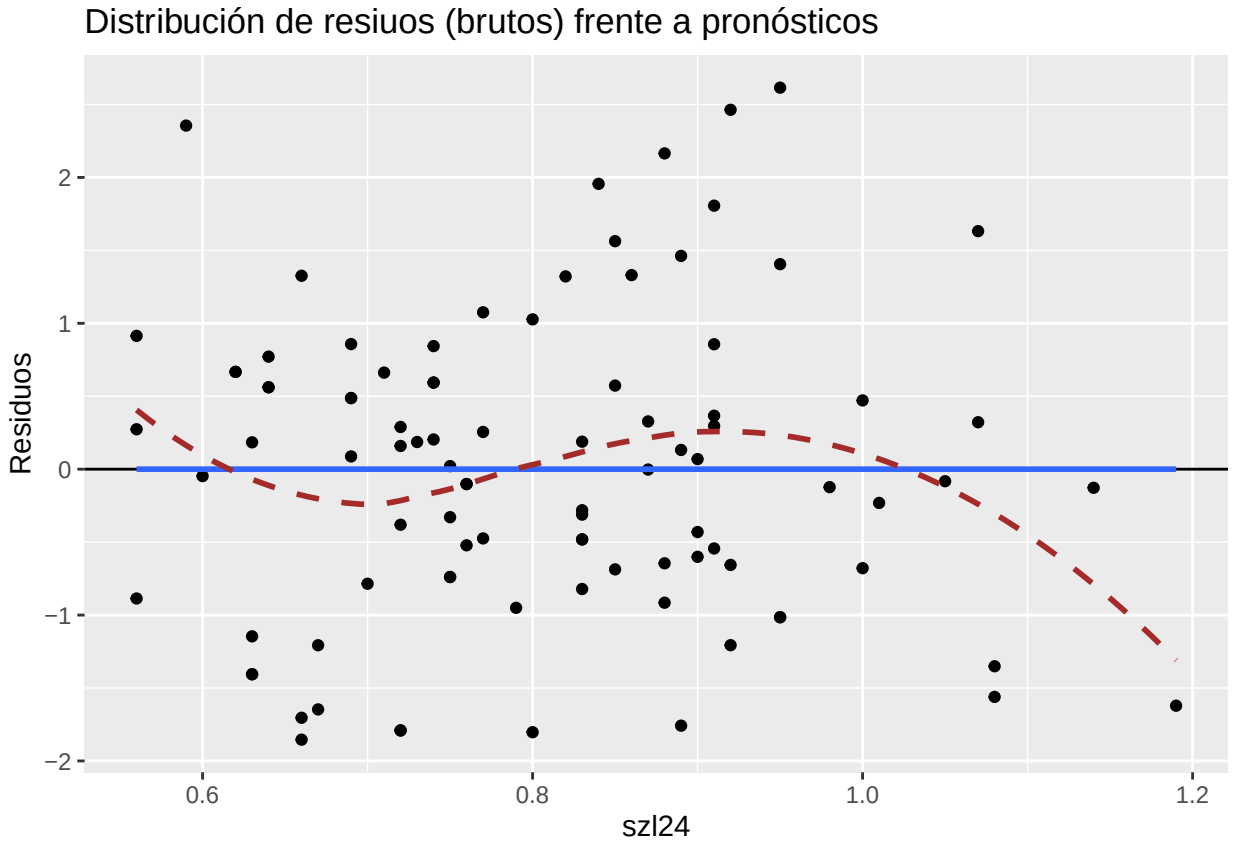

Normal Q−Q
Standardized residuals
Theoretical Quantiles
61
63
66

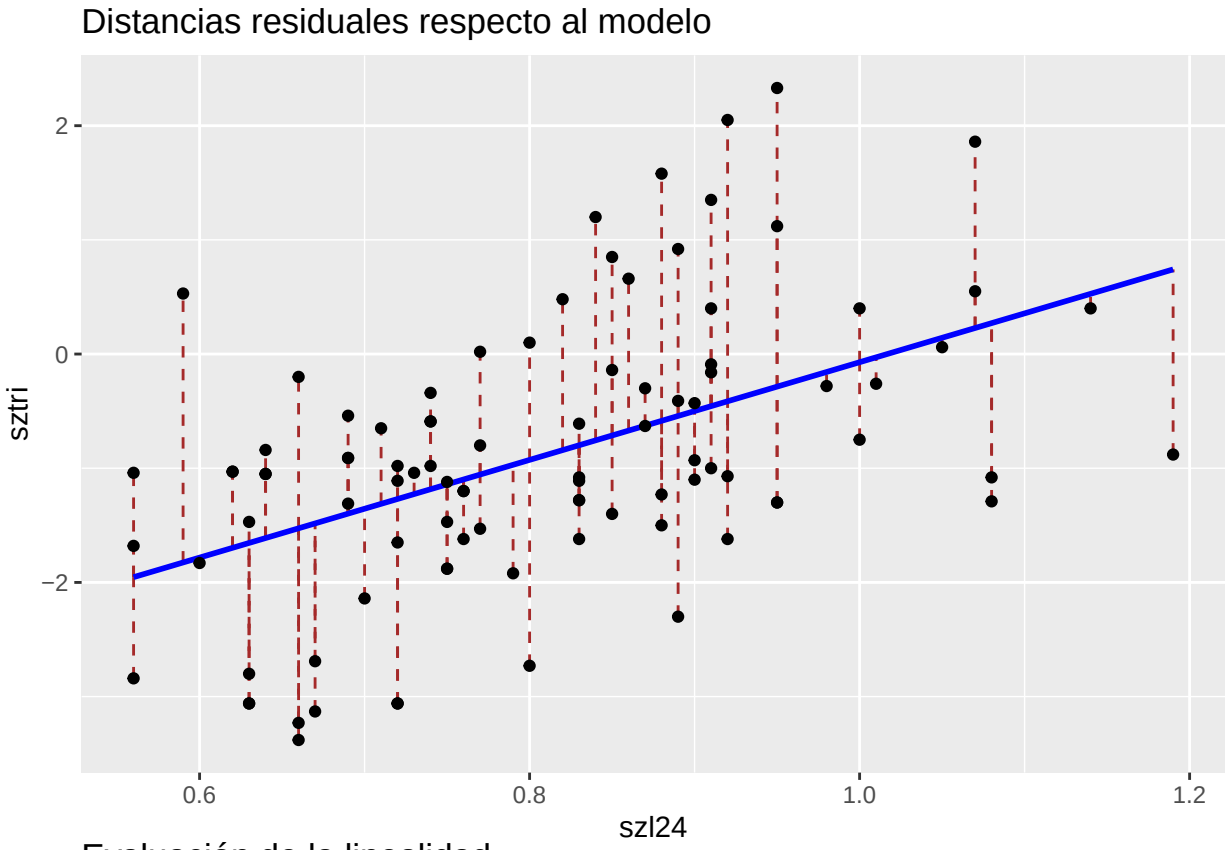

Scale−Location
√|Standardized residuals|
Fitted values
63
61

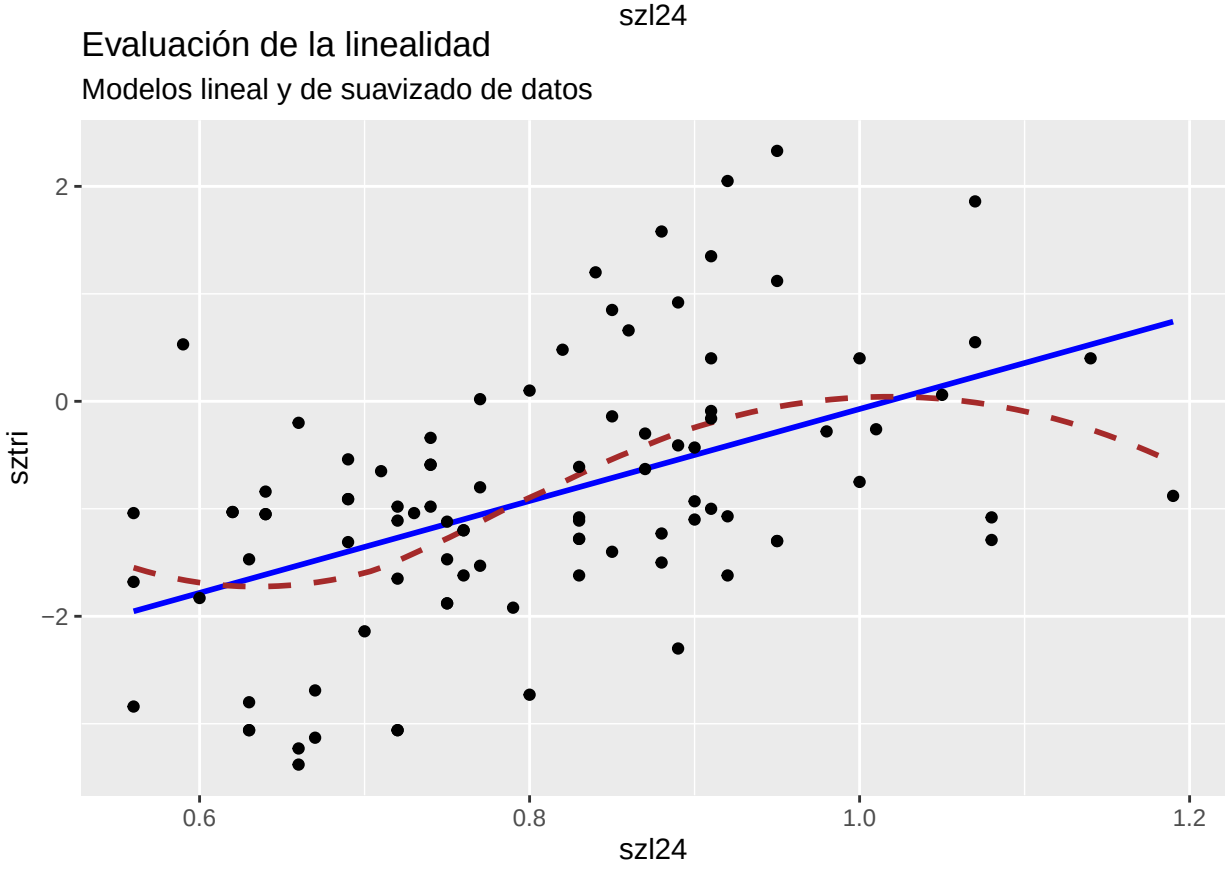

Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
61
63
46

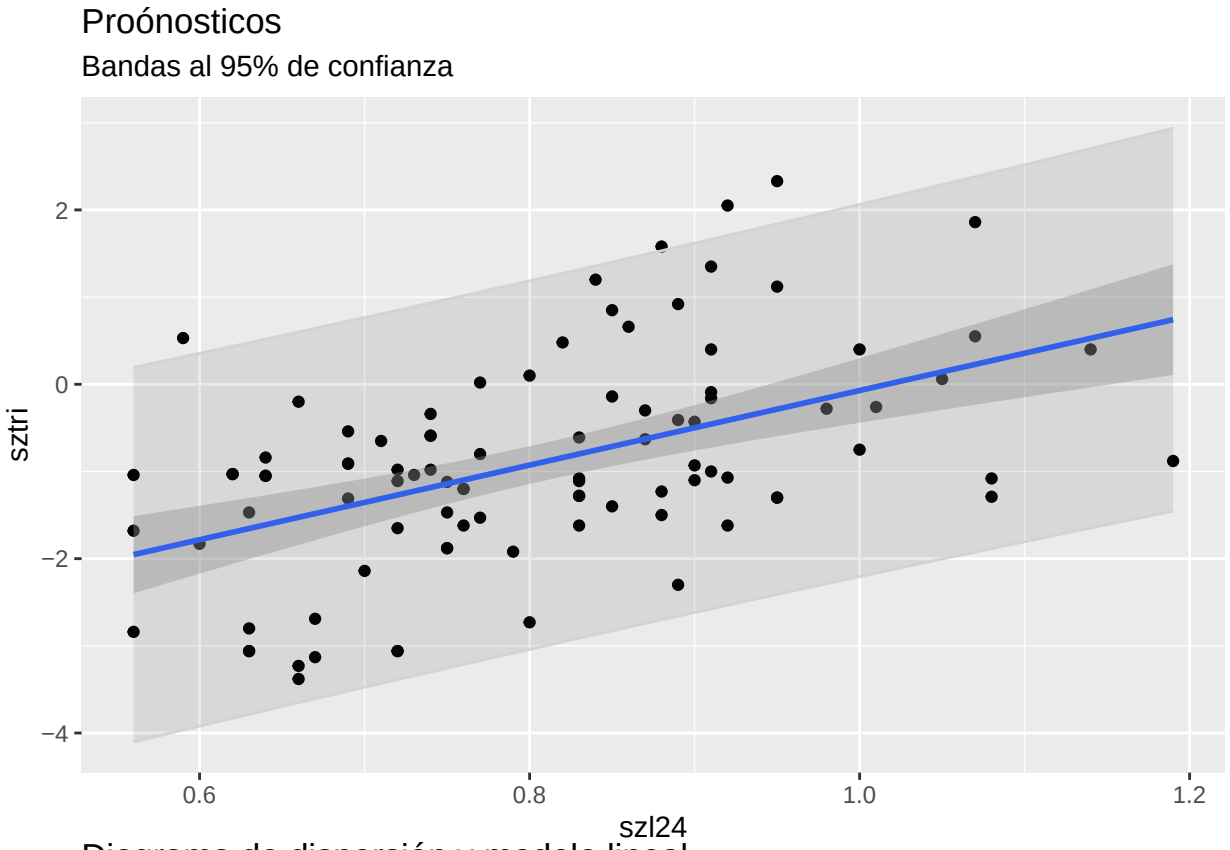

Distribución residual
frecuencia
Residuos estandarizados


Resiuos estandarizados frente a valores pronosticados
Residuos estandarizados
Valores pronosticados
2
1
0
−1
−2
−2
−1
0


Distribución de resiuos (brutos) frente a pronósticos
Residuos
szl24
2
1
0
−1
−2
0.6
0.8
1.0
1.2


Distancias residuales respecto al modelo
sztri
szl24
2
0
−2
0.6
0.8
1.0
1.2


Evaluación de la linealidad
Modelos lineal y de suavizado de datos
sztri
szl24
2
0
−2
0.6
0.8
1.0
1.2


Proónosticos
Bandas al 95% de confianza
sztri
szl24
2
0
−2
−4
0.6
0.8
1.0
1.2

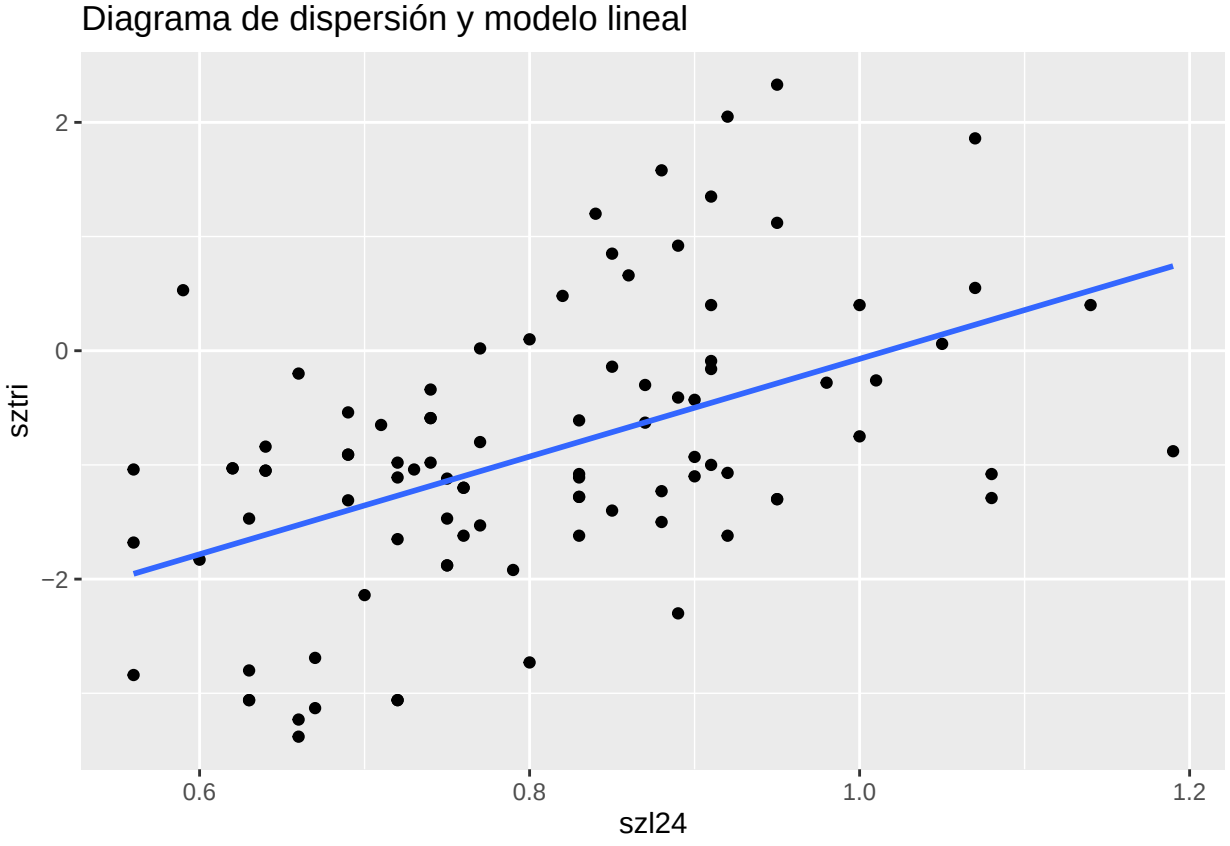
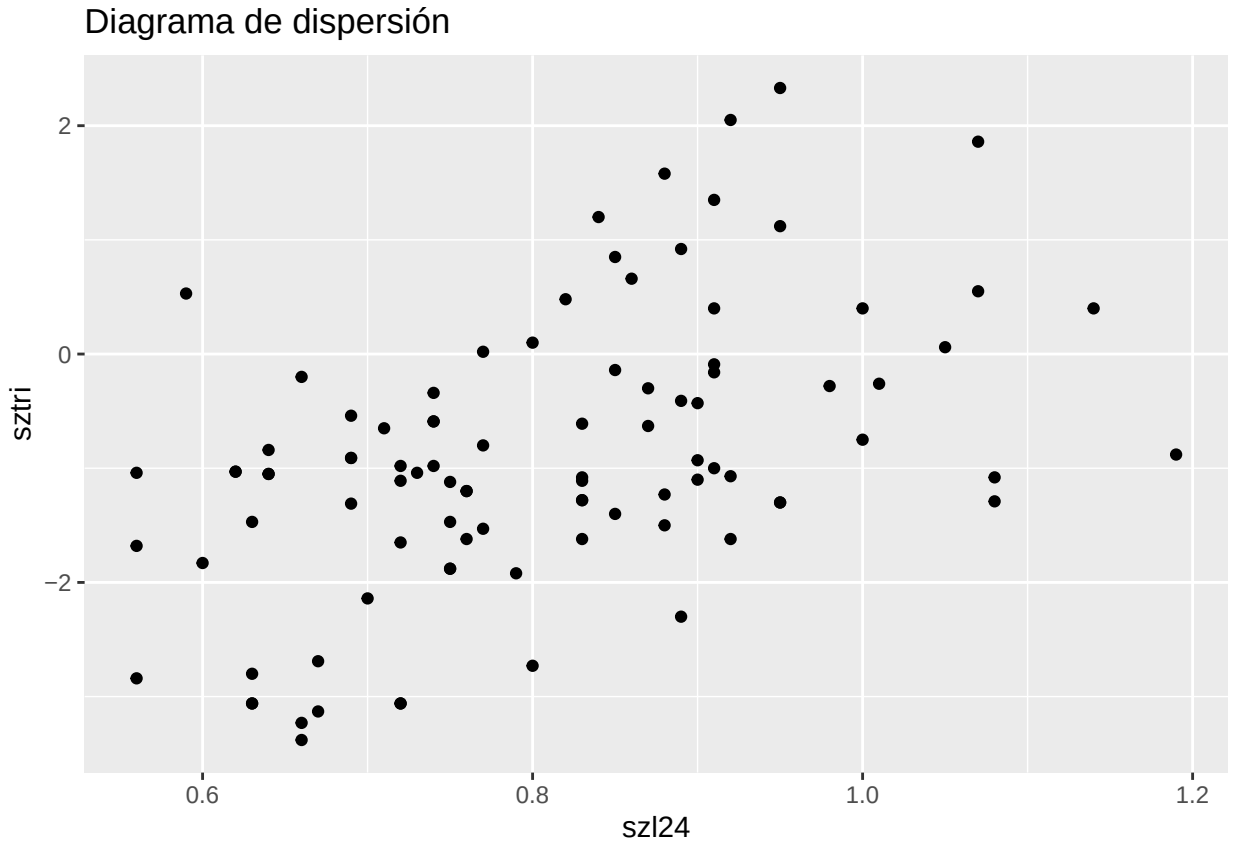


Diagrama de dispersión y modelo lineal
sztri
szl24
2
0
−2
0.6
0.8
1.0
1.2




```
##       x     y         res        zres        pre        lwr        upr
## 1  1.07  0.55  0.321969127  0.311851755  0.22803087 -0.23494923  0.69101098
## 2  0.83 -1.62 -0.821300355 -0.780230907 -0.79869965 -1.01837192 -0.57902737
## 3  0.83 -1.28 -0.481300355 -0.457232741 -0.79869965 -1.01837192 -0.57902737
## 4  0.72 -1.11  0.159284466  0.151619011 -1.26928447 -1.52524066 -1.01332828
## 5  0.56 -1.68  0.273771478  0.264573092 -1.95377148 -2.39581240 -1.51173055
## 6  0.92  2.05  2.463675701  2.348428210 -0.41367570 -0.69259547 -0.13475593
## 7  1.00 -0.75 -0.678567805 -0.651284728 -0.07143220 -0.44162409  0.29875970
## 8  0.83 -1.11 -0.311300355 -0.295733658 -0.79869965 -1.01837192 -0.57902737
## 9  0.72 -0.98  0.289284466  0.275362850 -1.26928447 -1.52524066 -1.01332828
## 10 0.69 -0.91  0.487625781  0.464955155 -1.39762578 -1.68118811 -1.11406345
## 11 0.92 -1.62 -1.206324299 -1.149894044 -0.41367570 -0.69259547 -0.13475593
## 12 0.91 -0.09  0.366456139  0.349103872 -0.45645614 -0.72585087 -0.18706140
## 13 0.98 -0.28 -0.123006928 -0.117817646 -0.15699307 -0.50242020  0.18843406
## 14 0.87 -0.30  0.327577892  0.311492344 -0.62757789 -0.86534628 -0.38980950
## 15 0.69 -0.91  0.487625781  0.464955155 -1.39762578 -1.68118811 -1.11406345
## 16 0.92 -1.07 -0.656324299 -0.625622316 -0.41367570 -0.69259547 -0.13475593
## 17 0.69 -0.54  0.857625781  0.817753169 -1.39762578 -1.68118811 -1.11406345
## 18 0.83 -1.28 -0.481300355 -0.457232741 -0.79869965 -1.01837192 -0.57902737
## 19 0.83 -0.61  0.188699645  0.179263645 -0.79869965 -1.01837192 -0.57902737
## 20 0.67 -3.13 -1.646813343 -1.572490517 -1.48318666 -1.78775604 -1.17861727
## 21 0.77  0.02  1.075382275  1.021857151 -1.05538227 -1.27986496 -0.83089959
## 22 0.63 -2.80 -1.145691590 -1.097865851 -1.65430841 -2.00544045 -1.30317637
## 23 0.76 -1.62 -0.521837287 -0.495980432 -1.09816271 -1.32717465 -0.86915078
## 24 1.01 -0.26 -0.231348243 -0.222295495 -0.02865176 -0.41157573  0.35427222
## 25 0.74 -0.34  0.843723589  0.802428724 -1.18372359 -1.42457581 -0.94287137
## 26 0.91 -0.16  0.296456139  0.282418481 -0.45645614 -0.72585087 -0.18706140
```

```
## 27 0.64 -1.05  0.561527972  0.537564691 -1.61152797 -1.95054602 -1.27250992
## 28 0.72 -3.06 -1.790715534 -1.704538580 -1.26928447 -1.52524066 -1.01332828
## 29 0.90 -0.43  0.069236578  0.065922200 -0.49923658 -0.75968988 -0.23878327
## 30 0.77 -0.80  0.255382275  0.242671104 -1.05538227 -1.27986496 -0.83089959
## 31 0.66 -3.38 -1.854032905 -1.771772528 -1.52596710 -1.84166891 -1.21026528
## 32 0.62 -1.03  0.667088848  0.639901071 -1.69708885 -2.06059853 -1.33357916
## 33 0.64 -0.84  0.771527972  0.738602914 -1.61152797 -1.95054602 -1.27250992
## 34 0.90 -0.93 -0.430763422 -0.410142638 -0.49923658 -0.75968988 -0.23878327
## 35 0.80 -2.73 -1.802959040 -1.712582985 -0.92704096 -1.14416116 -0.70992076
## 36 0.86  0.66  1.330358331  1.264626387 -0.67035833 -0.90217003 -0.43854663
## 37 0.79 -1.92 -0.950178602 -0.902611260 -0.96982140 -1.18831586 -0.75132693
## 38 0.80  0.10  1.027040960  0.975558975 -0.92704096 -1.14416116 -0.70992076
## 39 0.69 -1.31  0.087625781  0.083551896 -1.39762578 -1.68118811 -1.11406345
## 40 1.08 -1.29 -1.560811311 -1.514104972  0.27081131 -0.20598541  0.74760803
## 41 0.83 -1.08 -0.281300355 -0.267233820 -0.79869965 -1.01837192 -0.57902737
## 42 0.87 -0.63 -0.002422108 -0.002303171 -0.62757789 -0.86534628 -0.38980950
## 43 0.85 -1.40 -0.686861231 -0.652751126 -0.71313877 -0.93990876 -0.48636877
## 44 0.89 -2.30 -1.757982984 -1.673010434 -0.54201702 -0.79417458 -0.28985945
## 45 0.95 -1.30 -1.014665614 -0.969277211 -0.28533439 -0.59577771  0.02510894
## 46 1.19 -0.88 -1.621396131 -1.606745056  0.74139613  0.10721755  1.37557471
## 47 0.85 -0.14  0.573138769  0.544676216 -0.71313877 -0.93990876 -0.48636877
## 48 0.88 -1.50 -0.915202546 -0.870589557 -0.58479745 -0.82937069 -0.34022422
## 49 0.88 -1.23 -0.645202546 -0.613751132 -0.58479745 -0.82937069 -0.34022422
## 50 0.75 -1.47 -0.329056849 -0.312843168 -1.14094315 -1.37543318 -0.90645313
## 51 0.72 -3.06 -1.790715534 -1.704538580 -1.26928447 -1.52524066 -1.01332828
## 52 0.62 -1.03  0.667088848  0.639901071 -1.69708885 -2.06059853 -1.33357916
## 53 0.88  1.58  2.164797454  2.059270994 -0.58479745 -0.82937069 -0.34022422
## 54 0.77 -1.53 -0.474617725 -0.450994523 -1.05538227 -1.27986496 -0.83089959
## 55 1.05  0.06 -0.082469996 -0.079646667  0.14247000 -0.29323272  0.57817271
## 56 0.89  0.92  1.462017016  1.391350055 -0.54201702 -0.79417458 -0.28985945
## 57 0.82  0.48  1.321480084  1.255275937 -0.84148008 -1.05919370 -0.62376647
## 58 0.63 -3.06 -1.405691590 -1.347012413 -1.65430841 -2.00544045 -1.30317637
## 59 0.90 -1.10 -0.600763422 -0.572004684 -0.49923658 -0.75968988 -0.23878327
## 60 1.00  0.40  0.471432195  0.452477389 -0.07143220 -0.44162409  0.29875970
## 61 0.95  2.33  2.615334386  2.498344269 -0.28533439 -0.59577771  0.02510894
## 62 0.71 -0.65  0.662064904  0.630527964 -1.31206490 -1.57662705 -1.04750275
## 63 0.59  0.53  2.355430163  2.267233109 -1.82543016 -2.22740867 -1.42345165
## 64 0.95  1.12  1.405334386  1.342470442 -0.28533439 -0.59577771  0.02510894
## 65 0.63 -1.47  0.184308410  0.176614642 -1.65430841 -2.00544045 -1.30317637
## 66 0.76 -1.20 -0.101837287 -0.096791285 -1.09816271 -1.32717465 -0.86915078
## 67 0.66 -3.23 -1.704032905 -1.628427780 -1.52596710 -1.84166891 -1.21026528
## 68 0.75 -1.12  0.020943151  0.019911215 -1.14094315 -1.37543318 -0.90645313
## 69 0.67 -2.69 -1.206813343 -1.152348289 -1.48318666 -1.78775604 -1.17861727
## 70 0.76 -1.20 -0.101837287 -0.096791285 -1.09816271 -1.32717465 -0.86915078
## 71 0.63 -3.06 -1.405691590 -1.347012413 -1.65430841 -2.00544045 -1.30317637
## 72 0.85  0.85  1.563138769  1.485511985 -0.71313877 -0.93990876 -0.48636877
## 73 0.60 -1.83 -0.047350275 -0.045522247 -1.78264972 -2.17160448 -1.39369497
## 74 0.74 -0.59  0.593723589  0.564664623 -1.18372359 -1.42457581 -0.94287137
## 75 0.91  0.40  0.856456139  0.815901612 -0.45645614 -0.72585087 -0.18706140
## 76 0.66 -0.20  1.325967095  1.267136126 -1.52596710 -1.84166891 -1.21026528
## 77 0.75 -1.88 -0.739056849 -0.702641161 -1.14094315 -1.37543318 -0.90645313
## 78 0.64 -1.05  0.561527972  0.537564691 -1.61152797 -1.95054602 -1.27250992
## 79 0.56 -1.04  0.913771478  0.883069877 -1.95377148 -2.39581240 -1.51173055
## 80 0.84  1.20  1.955919207  1.858397981 -0.75591921 -0.97862465 -0.53321377
```

```
## 81 0.75 -1.88 -0.739056849 -0.702641161 -1.14094315 -1.37543318 -0.90645313
## 82 0.91  1.35  1.806456139  1.720917637 -0.45645614 -0.72585087 -0.18706140
## 83 0.74 -0.59  0.593723589  0.564664623 -1.18372359 -1.42457581 -0.94287137
## 84 0.95 -1.30 -1.014665614 -0.969277211 -0.28533439 -0.59577771  0.02510894
## 85 0.73 -1.04  0.186504028  0.177447262 -1.22650403 -1.47453452 -0.97847354
## 86 0.91 -1.00 -0.543543861 -0.517806215 -0.45645614 -0.72585087 -0.18706140
## 87 0.70 -2.14 -0.785154658 -0.748181607 -1.35484534 -1.62862957 -1.08106111
## 88 1.08 -1.08 -1.350811311 -1.310389095  0.27081131 -0.20598541  0.74760803
## 89 1.14  0.40 -0.127493940 -0.124999271  0.52749394 -0.03413734  1.08912522
## 90 0.74 -0.98  0.203723589  0.193752625 -1.18372359 -1.42457581 -0.94287137
## 91 0.72 -1.65 -0.380715534 -0.362393861 -1.26928447 -1.52524066 -1.01332828
## 92 0.89 -0.41  0.132017016  0.125635940 -0.54201702 -0.79417458 -0.28985945
## 93 0.56 -2.84 -0.886228522 -0.856452331 -1.95377148 -2.39581240 -1.51173055
## 94 1.07  1.86  1.631969127  1.580687069  0.22803087 -0.23494923  0.69101098
##        lwr.1     upr.1
## 1  -1.924488 2.3805495
## 2  -2.912285 1.3148854
## 3  -2.912285 1.3148854
## 4  -3.386948 0.8483792
## 5  -4.101884 0.1943407
## 6  -2.534237 1.7068860
## 7  -2.205918 2.0630532
## 8  -2.912285 1.3148854
## 9  -3.386948 0.8483792
## 10 -3.518803 0.7235516
## 11 -2.534237 1.7068860
## 12 -2.575786 1.6628738
## 13 -2.287323 1.9733369
## 14 -2.743120 1.4879645
## 15 -3.518803 0.7235516
## 16 -2.534237 1.7068860
## 17 -3.518803 0.7235516
## 18 -2.912285 1.3148854
## 19 -2.912285 1.3148854
## 20 -3.607274 0.6409010
## 21 -3.169473 1.0587081
## 22 -3.785571 0.4769540
## 23 -3.212739 1.0164134
## 24 -2.165382 2.1080786
## 25 -3.299615 0.9321676
## 26 -2.575786 1.6628738
## 27 -3.740828 0.5177721
## 28 -3.386948 0.8483792
## 29 -2.617448 1.6189753
## 30 -3.169473 1.0587081
## 31 -3.651680 0.5997453
## 32 -3.830425 0.4362477
## 33 -3.740828 0.5177721
## 34 -2.617448 1.6189753
## 35 -3.040362 1.1862803
## 36 -2.785240 1.4445228
## 37 -3.083284 1.1436415
## 38 -3.040362 1.1862803
## 39 -3.518803 0.7235516
```

```
## 40 -1.884721 2.4263439
## 41 -2.912285 1.3148854
## 42 -2.743120 1.4879645
## 43 -2.827473 1.4011957
## 44 -2.659225 1.5751909
## 45 -2.410272 1.8396034
## 46 -1.454320 2.9371120
## 47 -2.827473 1.4011957
## 48 -2.701115 1.5315205
## 49 -2.701115 1.5315205
## 50 -3.256120 0.9742333
## 51 -3.386948 0.8483792
## 52 -3.830425 0.4362477
## 53 -2.701115 1.5315205
## 54 -3.169473 1.0587081
## 55 -2.004347 2.2892868
## 56 -2.659225 1.5751909
## 57 -2.954862 1.2719023
## 58 -3.785571 0.4769540
## 59 -2.617448 1.6189753
## 60 -2.205918 2.0630532
## 61 -2.410272 1.8396034
## 62 -3.430786 0.8066561
## 63 -3.965657 0.3147970
## 64 -2.410272 1.8396034
## 65 -3.785571 0.4769540
## 66 -3.212739 1.0164134
## 67 -3.651680 0.5997453
## 68 -3.256120 0.9742333
## 69 -3.607274 0.6409010
## 70 -3.212739 1.0164134
## 71 -3.785571 0.4769540
## 72 -2.827473 1.4011957
## 73 -3.920469 0.3551696
## 74 -3.299615 0.9321676
## 75 -2.575786 1.6628738
## 76 -3.651680 0.5997453
## 77 -3.256120 0.9742333
## 78 -3.740828 0.5177721
## 79 -4.101884 0.1943407
## 80 -2.869822 1.3579832
## 81 -3.256120 0.9742333
## 82 -2.575786 1.6628738
## 83 -3.299615 0.9321676
## 84 -2.410272 1.8396034
## 85 -3.343224 0.8902163
## 86 -2.575786 1.6628738
## 87 -3.474738 0.7650470
## 88 -1.884721 2.4263439
## 89 -1.648377 2.7033651
## 90 -3.299615 0.9321676
## 91 -3.386948 0.8483792
## 92 -2.659225 1.5751909
## 93 -4.101884 0.1943407
```

```
## 94 -1.924488 2.3805495
```