

# Análisis de regresión

Capítulo 4: Transformaciones para corregir inadecuaciones

Mg. Sc. J. Eduardo Gamboa U.



## Transformaciones para estabilizar la varianza

La naturaleza de la variable respuesta es una primera señal para elegir una transformación, por ejemplo: transformación logarítmica para datos de conteo y transformación arcsin para datos que se encuentran entre 0 y 1. La relación entre la media y la varianza también es clave (Montgomery et. al, 2012) y se observa en la siguiente tabla.

Relación varianza-media	Transformación
$\sigma^2 \propto \mu$	$y' = \sqrt{y}$
$\sigma^2 \propto \mu(1 - \mu)$	$y' = \sin^{-1}(\sqrt{y})$
$\sigma^2 \propto \mu^2$	$y' = \log(y)$
$\sigma^2 \propto \mu^3$	$y' = y^{-1/2}$
$\sigma^2 \propto \mu^4$	$y' = \frac{1}{y}$

Mendenhall (2012) propone que para valores pequeños de conteos, se considere  
 $y' = \sqrt{y} + \sqrt{y + 1}$

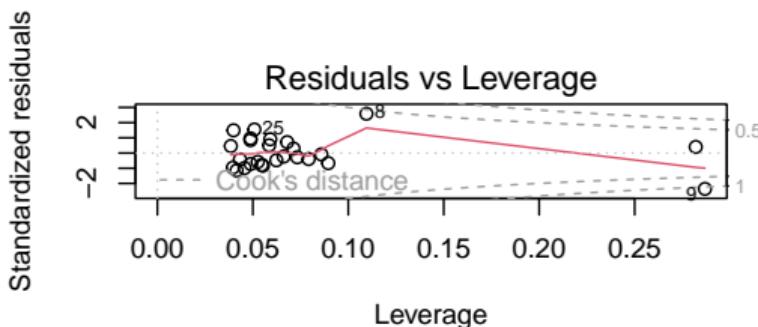
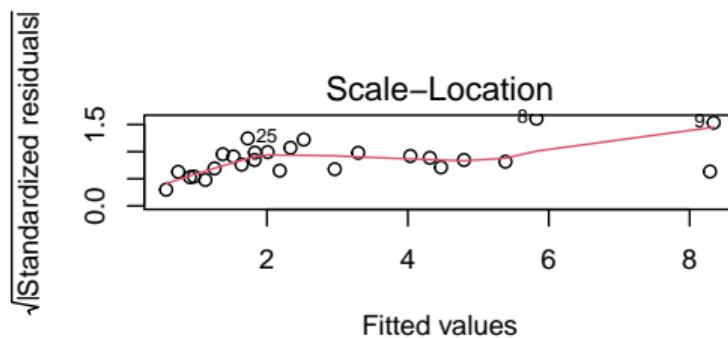
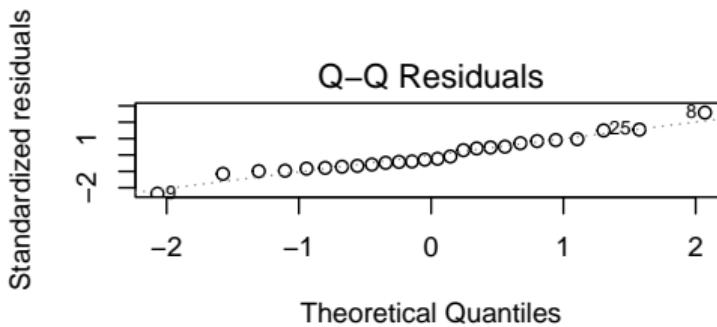
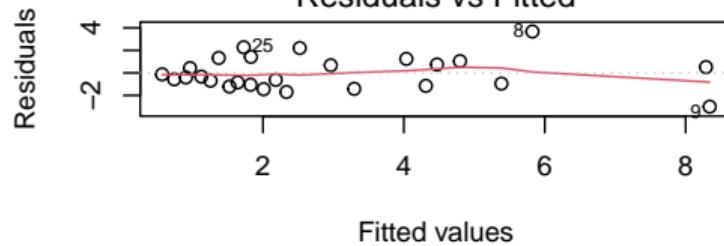
## Ejemplo 1

Una empresa de electricidad está interesada en desarrollar un modelo que relacione la demanda en las horas pico (y) en Kilovatios por hora (Kwh) con el uso total de energía durante el mes (x) en cantidad de Kilovatios (Kw). Los datos se encuentran en el archivo U4\_datos\_1.xlsx y han sido tomados de Montgomery et al (2012).

```
library(readxl)
library(broom)
library(dplyr)
datos2 = read_xlsx('U4_datos_1.xlsx')
lm(y~x, datos2) |> tidy()

# A tibble: 2 x 5
  term      estimate std.error statistic    p.value
  <chr>      <dbl>     <dbl>      <dbl>      <dbl>
1 (Intercept) -0.240     0.530     -0.453    0.654
2 x           0.00277   0.000385     7.21    0.000000191
```

```
par(mfrow=c(2,2))  
lm(y~x, datos2) |> plot()
```



```
lm(y~x, datos2) |> olsrr::ols_test_breusch_pagan()
```

Breusch Pagan Test for Heteroskedasticity

---

Ho: the variance is constant  
Ha: the variance is not constant

Data

---

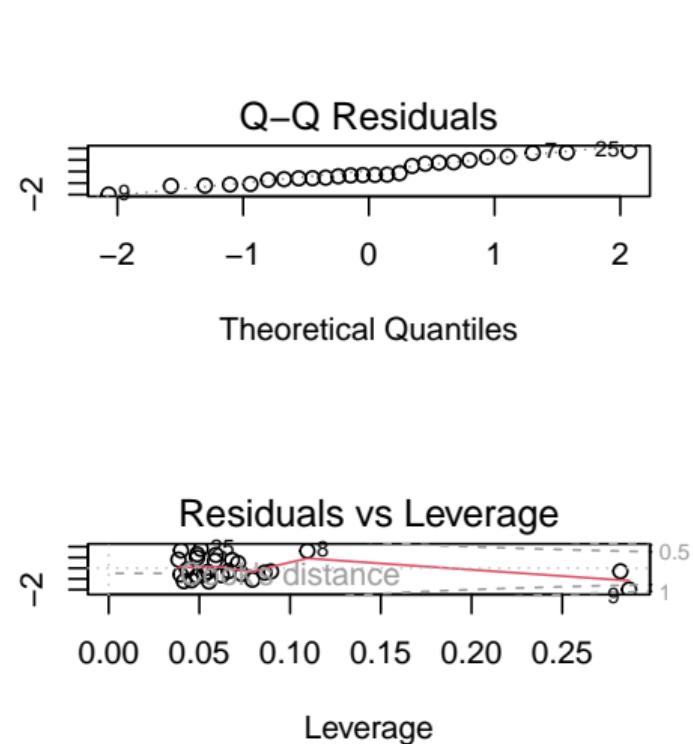
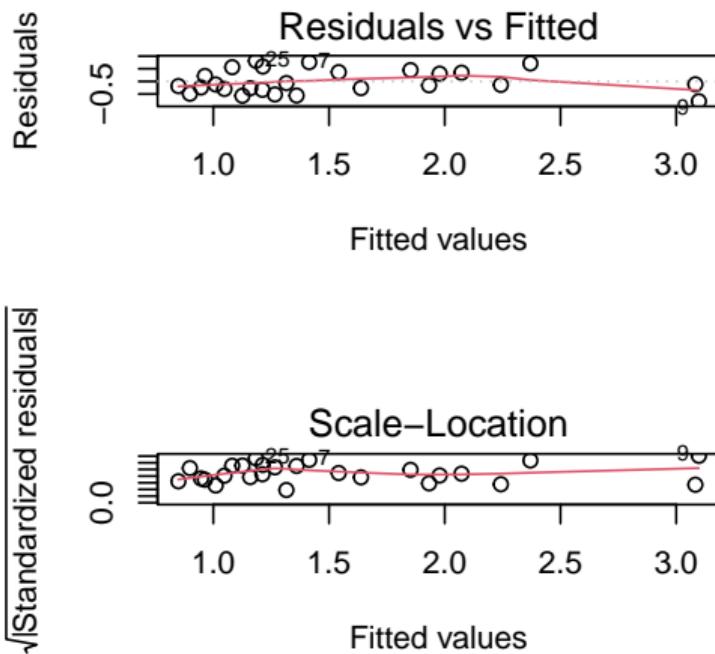
Response : y  
Variables: fitted values of y

Test Summary

---

DF = 1  
Chi2 = 5.407324  
Prob > Chi2 = 0.02005243

```
datos2 |> mutate(y1 = sqrt(y)) -> datos2 # Transformación raíz cuadrada  
par(mfrow=c(2,2))  
lm(y1~x, datos2) |> plot()
```



```
lm(y1~x, datos2) |> olsrr::ols_test_breusch_pagan()
```

Breusch Pagan Test for Heteroskedasticity

---

Ho: the variance is constant

Ha: the variance is not constant

Data

---

Response : y1

Variables: fitted values of y1

Test Summary

---

DF = 1

Chi2 = 0.2540729

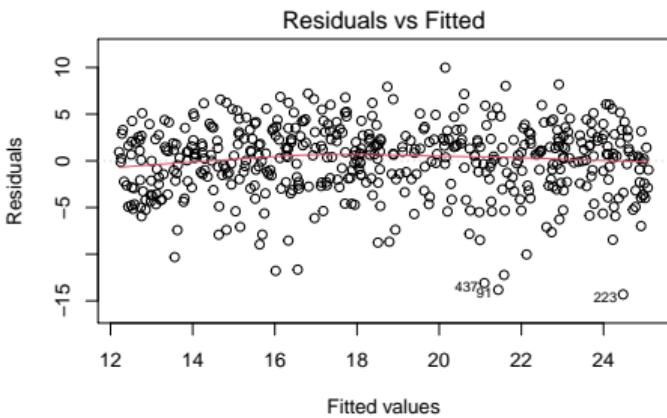
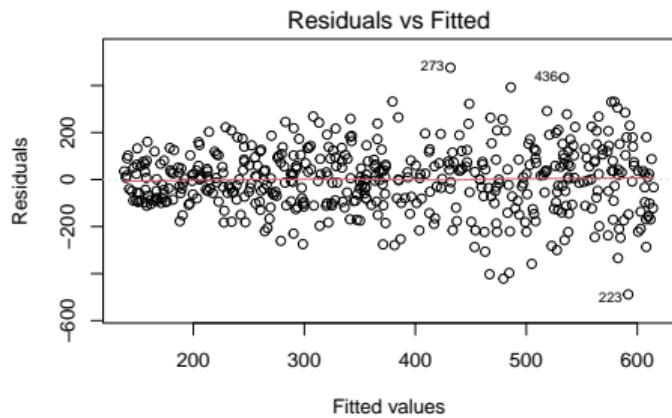
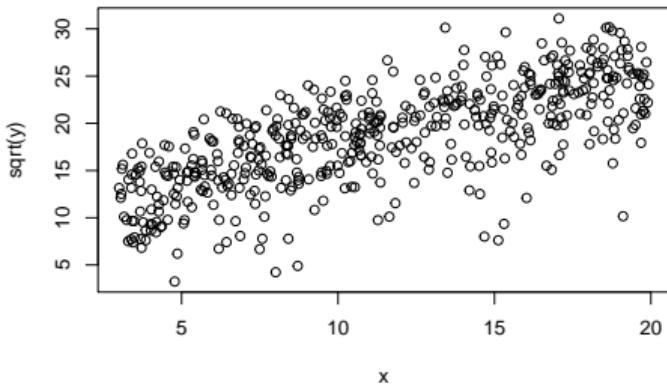
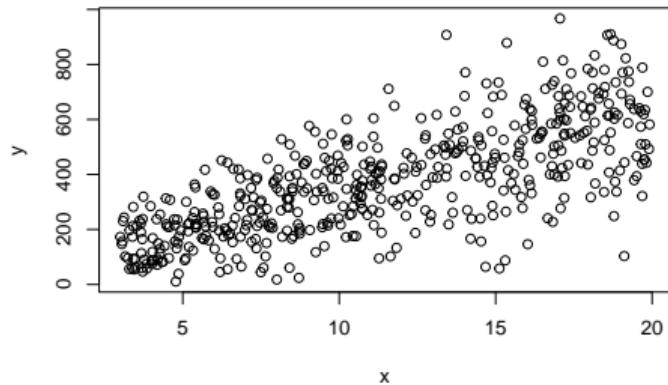
Prob > Chi2 = 0.6142217

Veamos algunos ejemplos a partir de simulaciones de casos:

## Ejemplo 2

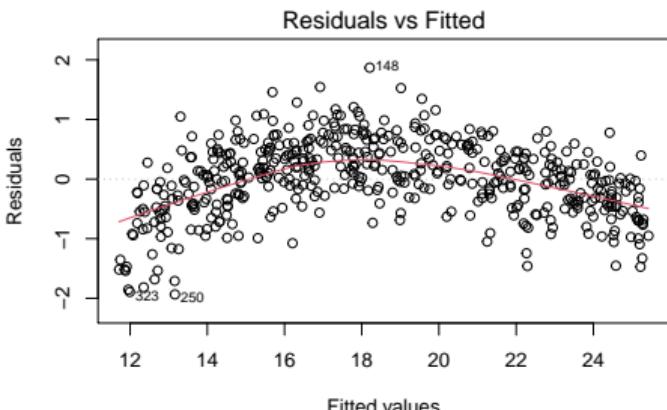
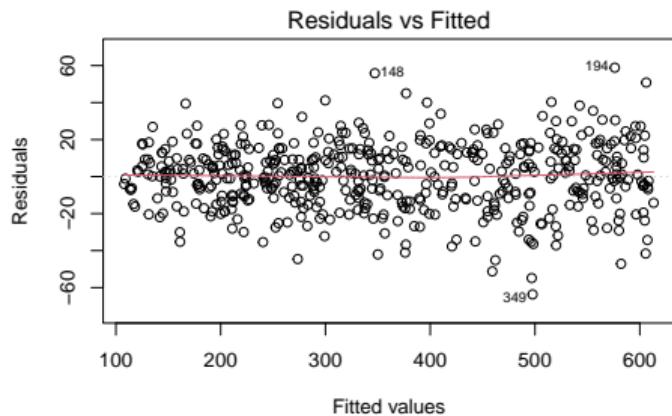
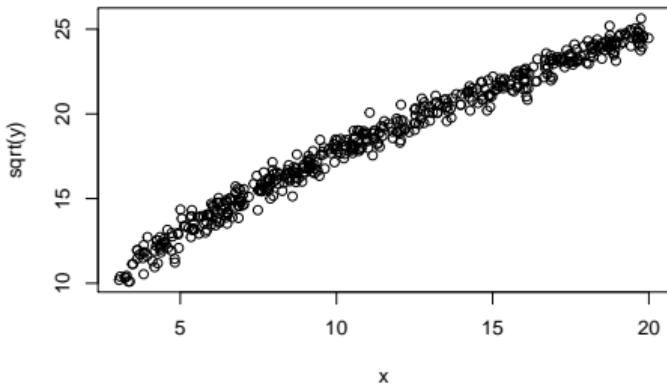
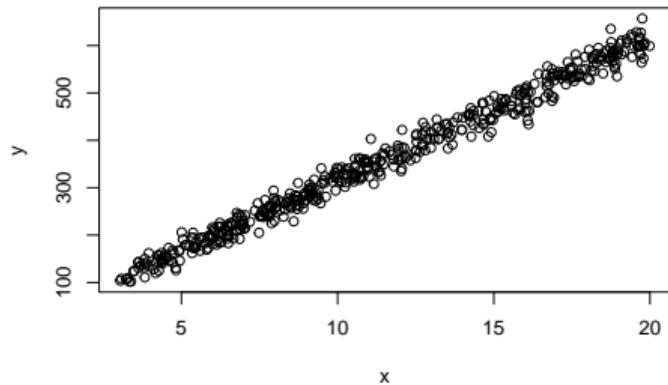
Transformación raíz cuadrada

```
set.seed(5)
x = runif(500,3,20)
b0 = 30;b1 = 30
mu = b0+b1*x;e = rnorm(500,0,sqrt(45*mu))
y = abs(mu + e)
par(mfrow=c(2,2))
plot(x,y)
plot(x,sqrt(y))
plot(lm(y~x),which=1)
plot(lm(sqrt(y)~x),which=1)
```



## Ejemplo 3

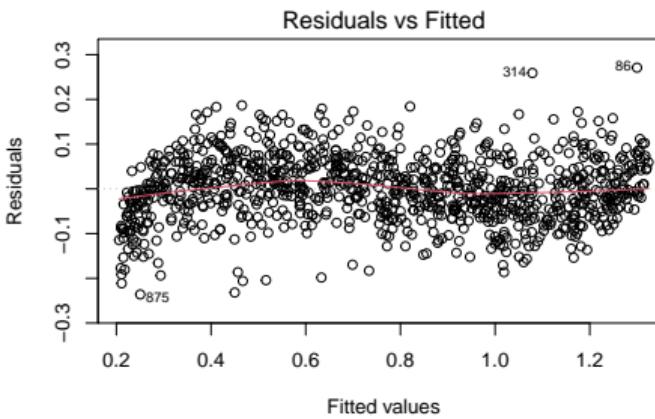
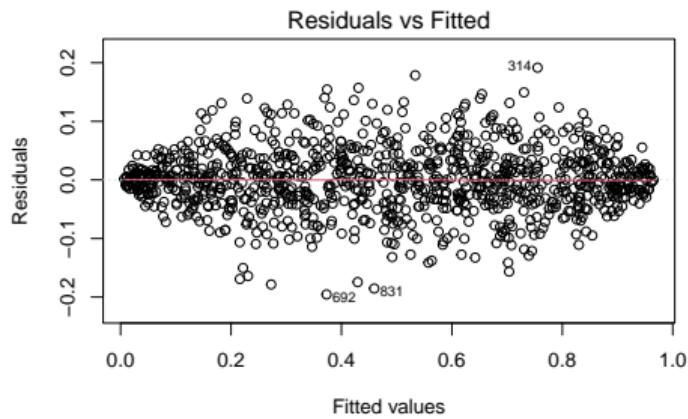
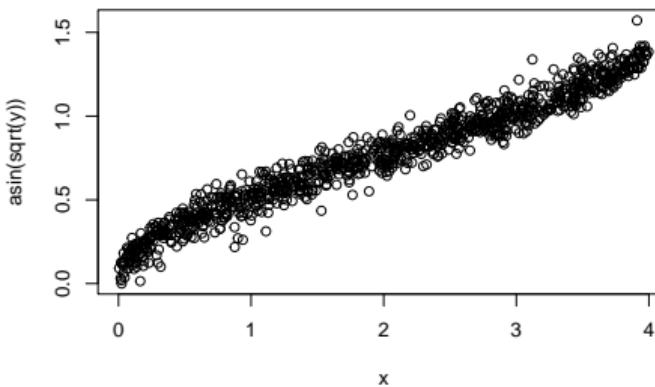
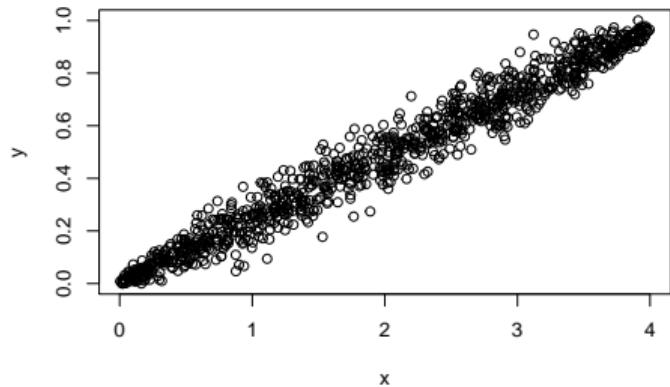
```
set.seed(444)
x = runif(500,3,20)
b0 = 10;b1 = 20
mu = b0+b1*x
y = rpois(500,mu*1.5)
par(mfrow=c(2,2))
plot(x,y)
plot(x,sqrt(y))
plot(lm(y~x),which=1)
plot(lm(sqrt(y)~x),which=1)
```



## Transformación arcsen + raíz cuadrada

### Ejemplo 4

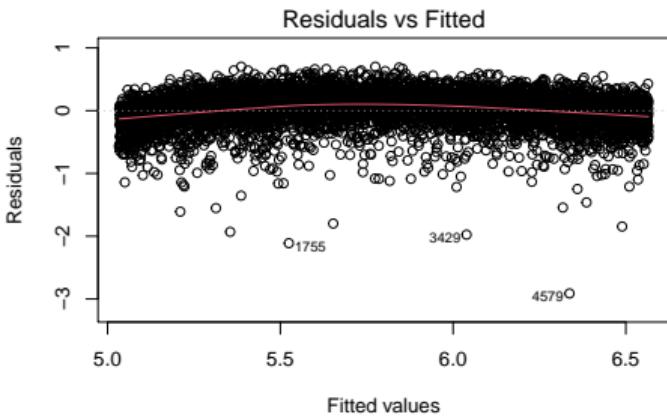
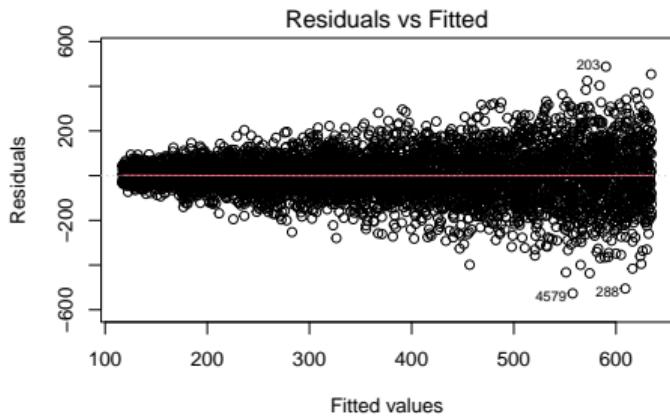
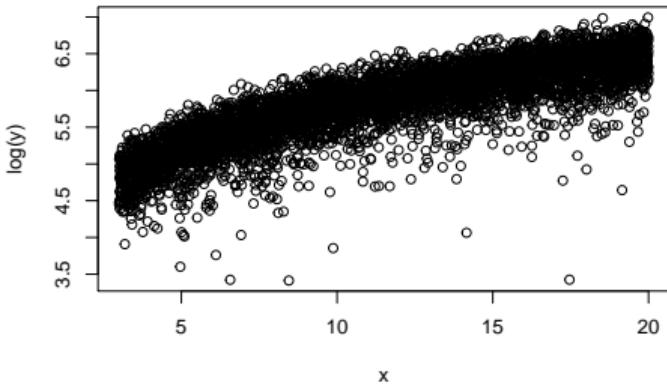
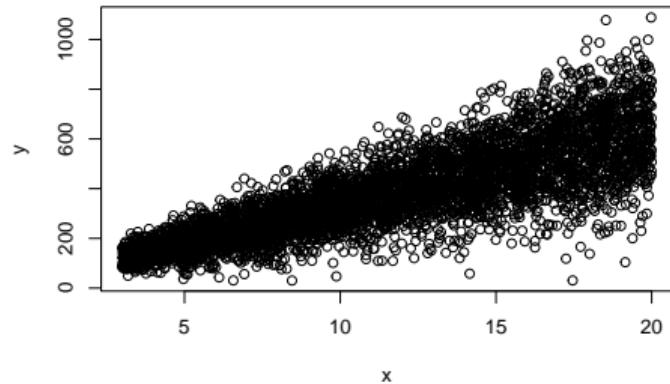
```
set.seed(2026)
x = runif(1000,0,4)
b0 = 1
b1 = 1
mu = b0+b1*x
mu = (mu-min(mu))/(max(mu)-min(mu))
e = rnorm(1000,0,sqrt(1/50*mu*(1-mu)))
y = mu + e
y = (y-min(y))/(max(y)-min(y))
par(mfrow=c(2,2))
plot(x,y)
plot(x,asin(sqrt(y)))
plot(lm(y~x),which=1)
plot(lm(asin(sqrt(y))~x),which=1)
```



## Transformación logaritmo

### Ejemplo 5

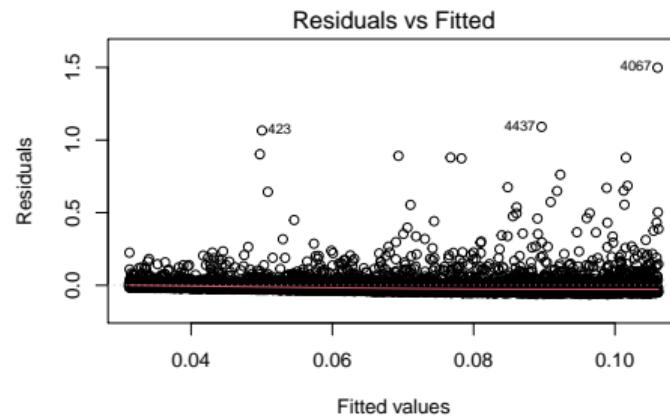
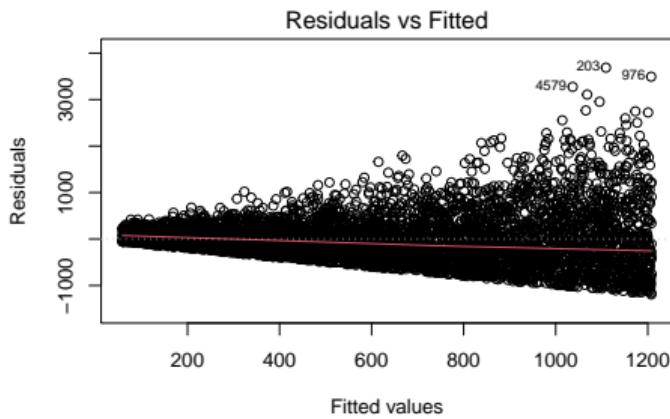
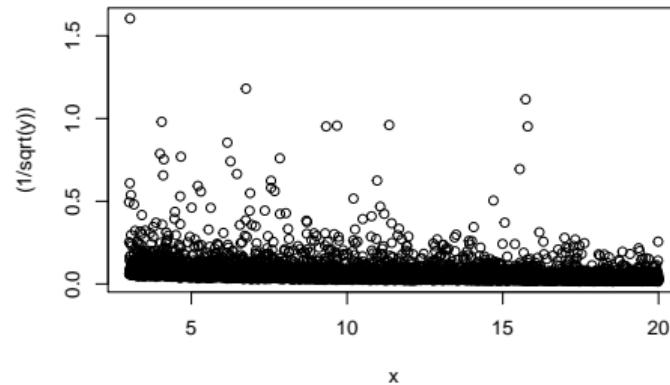
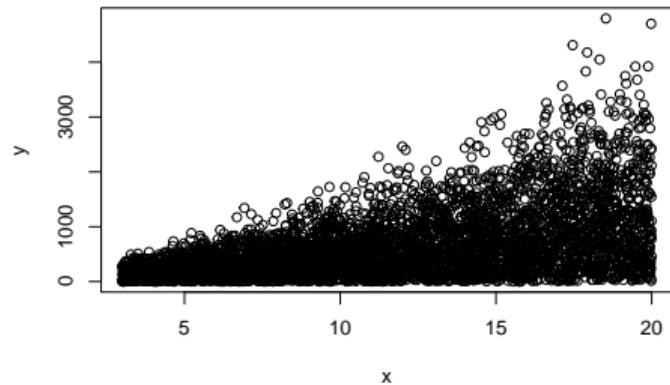
```
set.seed(100)
x = runif(5000,3,20)
b0 = 30
b1 = 30
mu = b0+b1*x
e = rnorm(5000,0,sqrt(1/15*mu**2))
y = abs(mu + e)
plot(x,y)
par(mfrow=c(2,2))
plot(x,y)
plot(x,log(y))
plot(lm(y~x),which=1)
plot(lm(log(y)~x),which=1)
```



## Transformación inversa + raíz cuadrada

### Ejemplo 6

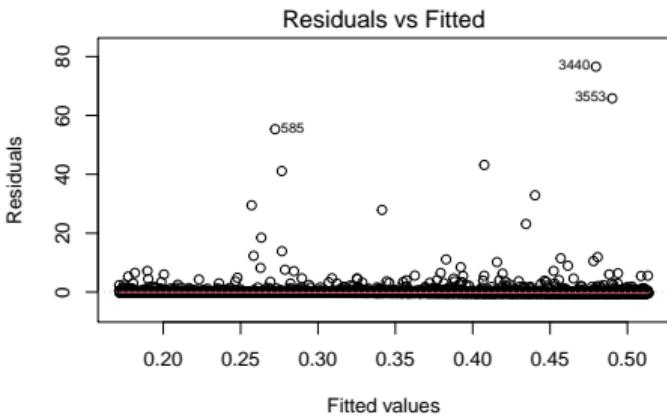
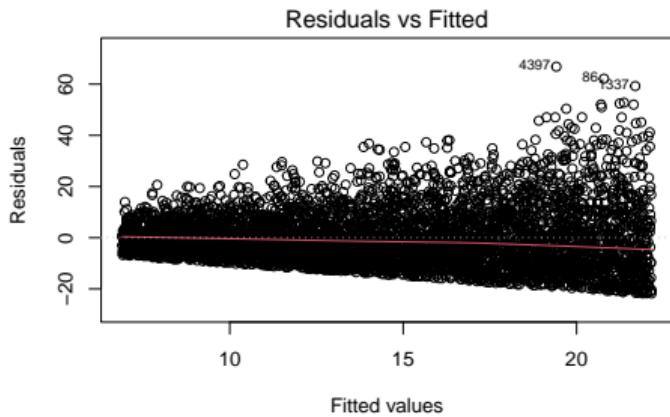
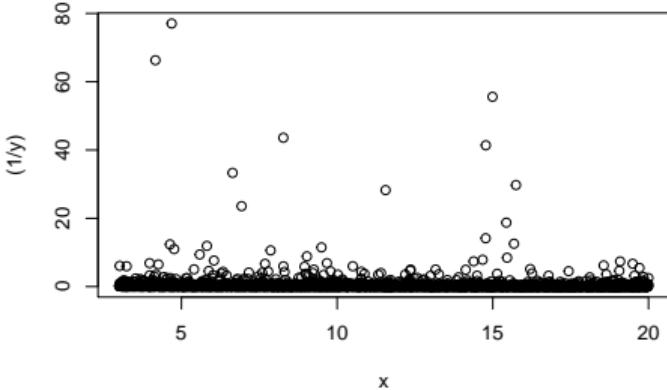
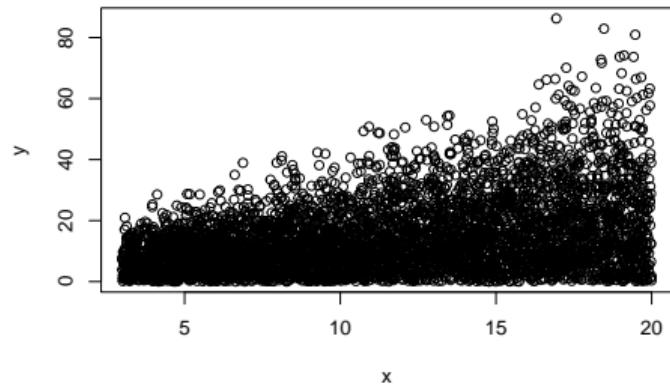
```
set.seed(100)
x = runif(5000,3,20)
b0 = 30
b1 = 30
mu = b0+b1*x
e = rnorm(5000,0,sqrt(1/120*mu**3))
y = abs(mu + e)
plot(x,y)
par(mfrow=c(2,2))
plot(x,y)
plot(x,(1/sqrt(y)))
plot(lm(y~x),which=1)
plot(lm((1/sqrt(y))~x),which=1)
```



## Transformación inversa

### Ejemplo 7

```
set.seed(15)
x = runif(5000,3,20)
b0 = 5
b1 = 0.4
mu = b0+b1*x
e = rnorm(5000,0,sqrt(1/40*mu**4))
y = abs(mu + e)
plot(x,y)
par(mfrow=c(2,2))
plot(x,y)
plot(x,(1/y))
plot(lm(y~x),which=1)
plot(lm((1/y)~x),which=1)
```



Si no se logra estabilizar la varianza, los estimadores seguirán siendo insesgados pero no tendrán la propiedad de mínima varianza, lo que significa que los errores estándar serán ¿más pequeños o más grandes? ¿Cómo afecta eso en sus intervalos de confianza y pruebas de hipótesis asociadas?

Cuando la variable respuesta es transformada, los valores predichos se encontrarán en la escala de la nueva variable. Es común convertir dichos valores a la escala original, sin embargo esa transformación inversa genera la estimación de la mediana de la distribución de la variable endógena en vez de la media. En cuanto a los límites de los intervalos de confianza para estimar o predecir un valor individual, estos pueden ser directamente transformados ya que se tratan de percentiles, los cuales no se ven afectados por transformaciones. Sin embargo, no es posible asegurar que los intervalos obtenidos son los más cortos posibles.

## Métodos analíticos para encontrar una transformación

En principio, se sugiere transformar  $y' = y^\lambda$ ; sin embargo, la variable respuesta se tornaría constante para  $\lambda = 0$ . Ante ello, se propone utilizar  $y' = \frac{y^\lambda - 1}{\lambda}$  de modo que  $y' = \log(y)$  cuando  $\lambda = 0$ .

Para verificar esto, resuelva:

$$\lim_{\lambda \rightarrow 0} \frac{y^\lambda - 1}{\lambda}$$

## Ejemplo 8

Veamos cómo se obtienen los valores transformados  $y'$  dados valores conocidos de  $\lambda$ , utilizando los datos del Ejemplo 7. Luego, veremos cómo estimar este valor.

```
lm((y**0.25-1)/0.25~x) |> aov() |> tidy()
```

```
# A tibble: 2 x 6
```

term	df	sumsq	meansq	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 x	1	1508.	1508.	541.	9.35e-114
2 Residuals	4998	13927.	2.79	NA	NA

```
library(forecast)
```

```
lm(BoxCox(y, 0.25)~x) |> aov() |> tidy()
```

```
# A tibble: 2 x 6
```

term	df	sumsq	meansq	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 x	1	1508.	1508.	541.	9.35e-114
2 Residuals	4998	13927.	2.79	NA	NA

```
lm(BoxCox(y, 0.50)~x) |> aov() |> tidy()
```

# A tibble: 2 x 6

term	df	sumsq	meansq	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 x	1	5595.	5595.	657.	3.17e-136
2 Residuals	4998	42568.	8.52	NA	NA

```
lm(BoxCox(y, 0.73)~x) |> aov() |> tidy()
```

# A tibble: 2 x 6

term	df	sumsq	meansq	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 x	1	20063.	20063.	727.	1.53e-149
2 Residuals	4998	138002.	27.6	NA	NA

Para elegir el valor óptimo de  $\lambda$  se busca aquel que maximiza la (log) verosimilitud al aplicar la transformación:

$$y' = \frac{y^\lambda - 1}{\lambda \hat{Y}^{\lambda-1}} \quad \hat{Y} = \exp\left(\frac{1}{n} \sum_{i=1}^n \log(y_i)\right)$$

Esto último permite que la variabilidad (a través de las sumas de cuadrados) de los modelos con diferentes  $\lambda$  sean comparables, lo cual se puede apreciar en estos ejemplos:

```
lambda = 0.25
exp(1/length(y)*sum(log(y))) -> yp
(y**lambda-1)/(lambda*yp**(lambda-1)) -> ynueva1
lm(ynueva1~x) |> aov() |> tidy()
```

```
# A tibble: 2 x 6
  term      df    sumsq   meansq statistic    p.value
  <chr>    <dbl>  <dbl>    <dbl>     <dbl>      <dbl>
1 x          1  44160.  44160.     541.  9.35e-114
2 Residuals 4998 407801.    81.6       NA      NA
```

```
lambda = 0.50
exp(1/length(y)*sum(log(y))) -> yp
(y**lambda-1)/(lambda*yp**(lambda-1)) -> ynueva2
lm(ynueva2~x) |> aov() |> tidy()
```

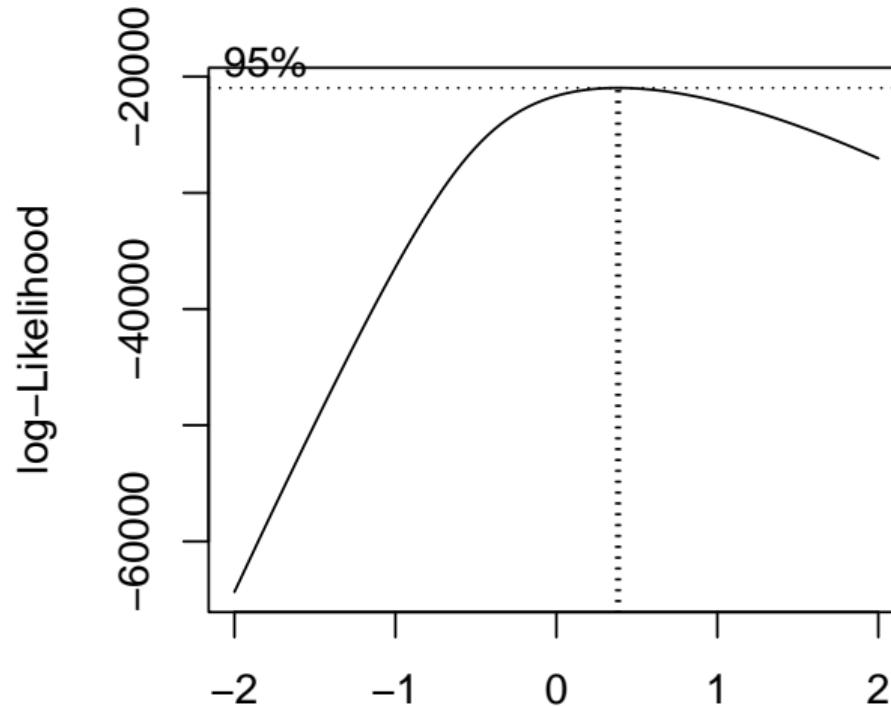
```
# A tibble: 2 x 6
  term      df    sumsq   meansq statistic    p.value
  <chr>    <dbl>  <dbl>    <dbl>     <dbl>    <dbl>
1 x         1  53152.  53152.     657.  3.17e-136
2 Residuals 4998 404407.     80.9      NA     NA
```

```
lambda = 0.73
exp(1/length(y)*sum(log(y))) -> yp
(y**lambda-1)/(lambda*yp**(lambda-1)) -> ynueva2
lm(ynueva2~x) |> aov() |> tidy()
```

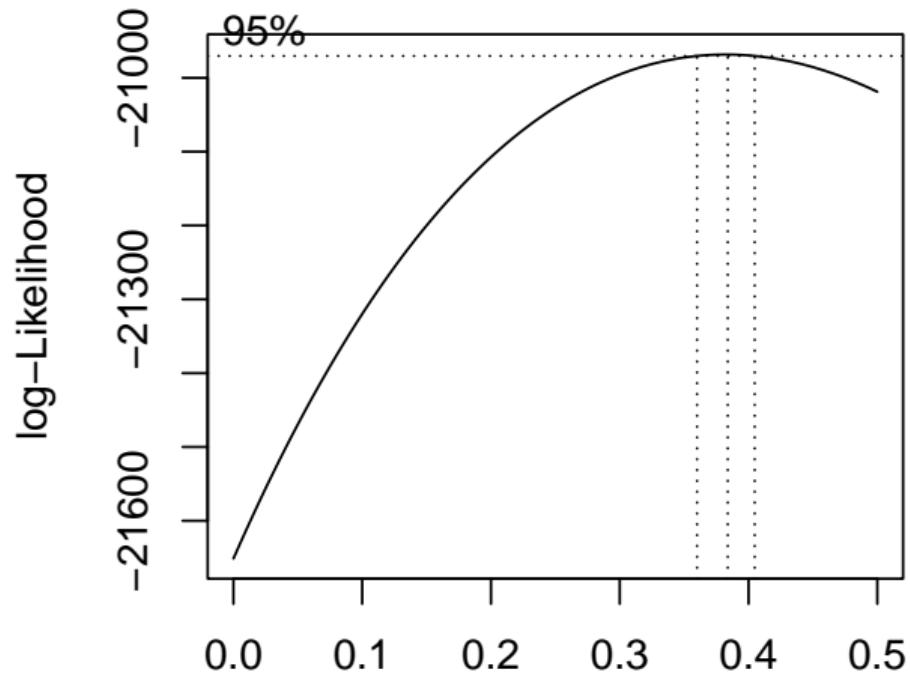
```
# A tibble: 2 x 6
  term      df    sumsq   meansq statistic    p.value
  <chr>    <dbl>  <dbl>    <dbl>     <dbl>    <dbl>
1 x         1  67665.  67665.     727.  1.53e-149
2 Residuals 4998 465438.     93.1      NA     NA
```

Note que este tipo de transformación es necesaria principalmente para elegir el valor de  $\lambda$  ya que para corregir la falta de normalidad y/o la falta de homogeneidad de varianzas, basta aplicar  $(y^\lambda - 1)/\lambda$  o incluso  $y^\lambda$  si sabemos que  $\lambda \neq 0$  y más aún si el valor de  $\lambda$  lleva a una interpretación sencilla como  $\lambda = 0.5$ , para el cual  $y^{0.5} = \sqrt{y}$  ¿Qué sucede al aplicar la transformación?

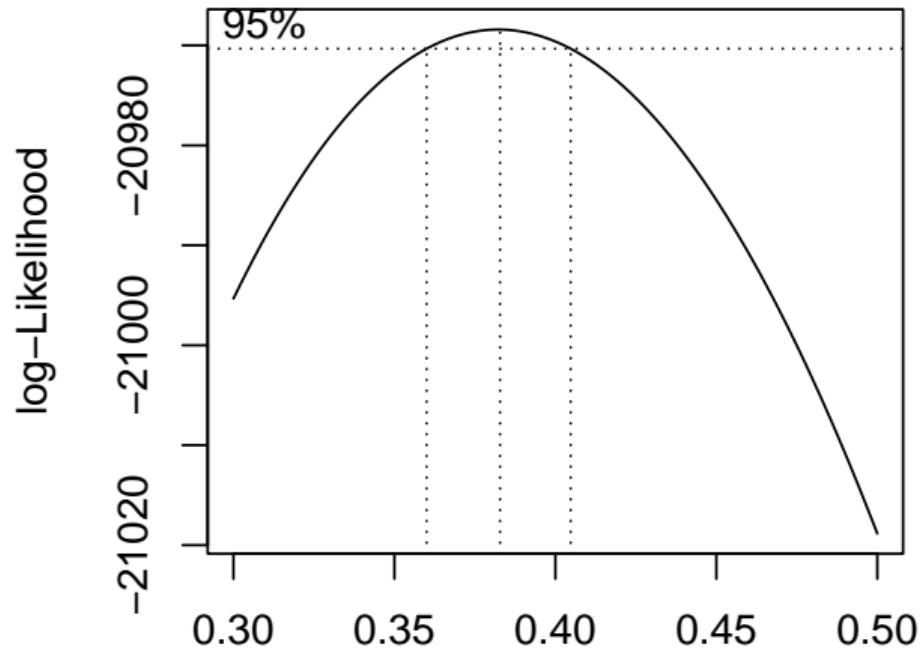
```
library(MASS)  
boxcox(lm(y~x))
```



```
boxcox(lm(y~x),lambda=seq(0,0.5,0.05))
```



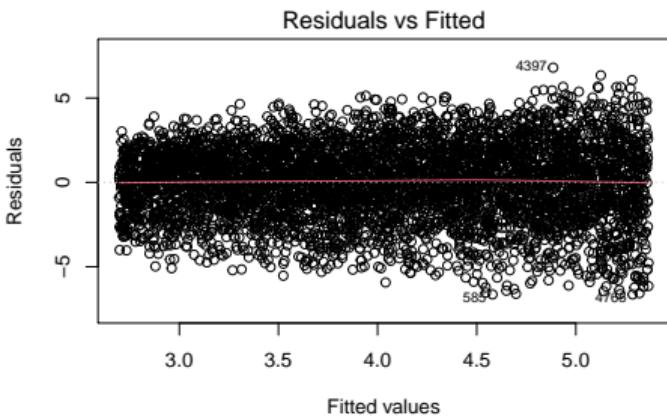
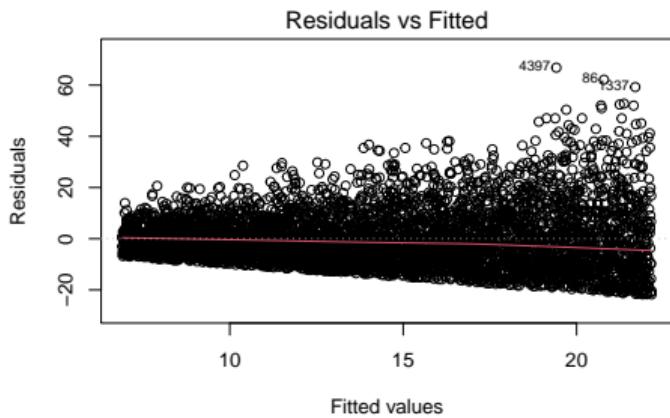
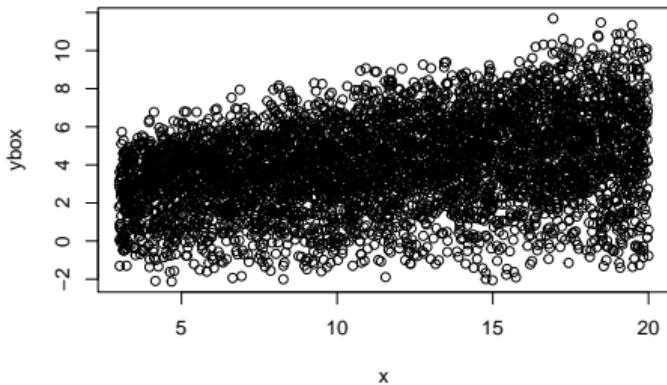
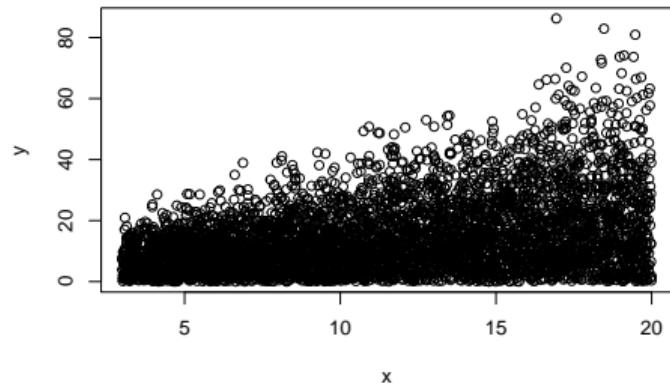
```
boxcox(lm(y~x),lambda=seq(0.3,0.5,0.01))
```



```
boxcox(lm(y~x),lambda=seq(0.35,0.45,0.01),plotit = FALSE) |>  
  rbind.data.frame()
```

	x	y
1	0.35	-20972.45
2	0.36	-20970.32
3	0.37	-20968.98
4	0.38	-20968.43
5	0.39	-20968.63
6	0.40	-20969.60
7	0.41	-20971.32
8	0.42	-20973.77
9	0.43	-20976.95
10	0.44	-20980.85
11	0.45	-20985.46

```
BoxCox(y,lambda = 0.39) -> ybox
lm(ybox ~ x) -> modbox
par(mfrow=c(2,2))
plot(x,y)
plot(x,ybox)
plot(lm(y~x),which=1)
plot(lm(ybox~x),which=1)
```



# Mínimos cuadrados generalizados y ponderados

## Mínimos cuadrados generalizados

Se ha asumido que:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon, \quad E(\varepsilon) = 0, \quad V(\varepsilon) = \sigma^2 \mathbf{I}$$

Modificaremos la última parte del supuesto, de tal modo que  $V(\varepsilon) = \sigma^2 \mathbf{V}$ , donde  $\mathbf{V}$  es una matriz de dimensión  $n \times n$  no singular y positiva definida; es decir, existe una matriz  $\mathbf{K}$  simétrica no singular  $n \times n$  tal que  $\mathbf{K}'\mathbf{K} = \mathbf{K}\mathbf{K}' = \mathbf{V}$ . Esta matriz también es conocida como la raíz cuadrada de  $\mathbf{V}$ . Sea:

$$\mathbf{z} = \mathbf{K}^{-1}\mathbf{y}, \quad \mathbf{B} = \mathbf{K}^{-1}\mathbf{X}, \quad \mathbf{g} = \mathbf{K}^{-1}\varepsilon$$

Entonces, multiplicando por  $K^{-1}$  al modelo de regresión, obtenemos:

$$\mathbf{z} = \mathbf{B}\beta + \mathbf{g}$$

En este modelo, es posible demostrar que  $E(\mathbf{g}) = 0$  y  $V(\mathbf{g}) = \sigma^2 \mathbf{I}$ , por lo que:

$$\hat{\beta} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{z} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$$

es el estimador de mínimos cuadrados generalizados de  $\beta$ , cuya varianza es:

$$V(\hat{\beta}) = \sigma^2(\mathbf{B}'\mathbf{B})^{-1} = \sigma^2(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}$$

## Mínimos cuadrados ponderados

Se asume que los errores no están correlacionados, pero tienen varianzas distintas. Así, se define la matriz  $\mathbf{V} = \mathbf{W}^{-1}$

$$\begin{pmatrix} \frac{1}{\omega_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\omega_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\omega_m} \end{pmatrix}$$

¿Qué valor tomaría la matriz  $\mathbf{K}$  de los mínimos cuadrados generalizados en este caso particular? ¿Qué valores tomarían  $\mathbf{z}$ ,  $\mathbf{B}$  y  $\mathbf{g}$  (errores ponderados)?

Al considerar la matriz de pesos  $\mathbf{W}$ , la suma de cuadrados del error también se pondera:

$$WSSE = SCPE = \sum_{i=1}^n w_i(y_i - \hat{y}_i)^2$$

de modo que las observaciones con mayor variabilidad tendrán menos peso, ya que  $v_i = \frac{1}{w_i}$ . Así, el estimador de mínimos cuadrados ponderados del vector de coeficientes de regresión está dado por:

$$\hat{\beta} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$$

La matriz de pesos debe ser conocida. Se tienen algunos criterios para elegir los valores de  $\mathbf{W}$ :

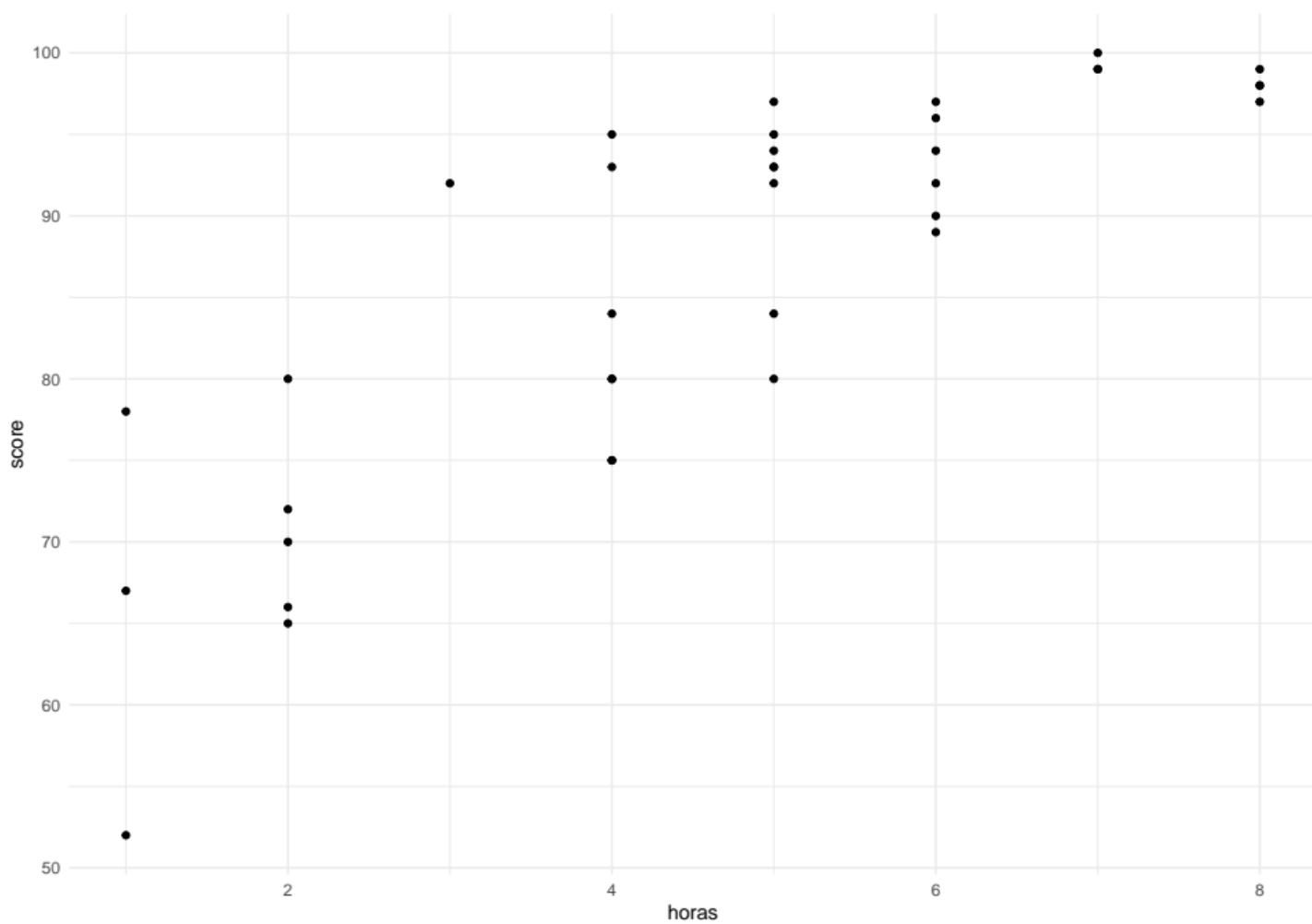
- ▶ Si la variable  $X_j$  está asociada a la variabilidad de los errores, de tal modo que  $V(\epsilon_i) = \sigma^2 x_{ij}$ , entonces  $\omega_i = \frac{1}{x_{ij}}$
- ▶ Si la variable  $x_j$  está asociada a la variabilidad de los errores, de tal modo que  $V(\epsilon_i) = \frac{\sigma^2}{x_{ij}}$  entonces  $\omega_i = x_{ij}$
- ▶ Si el valor de  $y_i$  ha sido obtenido a partir de una muestra de tamaño  $n_i$ , entonces  $\omega_i = n_i$
- ▶ Ejecutar una regresión de  $|\epsilon|$  en función de  $\hat{y}$  y extraer los valores ajustados de esta regresión, luego invertirlos y elevarlos al cuadrado. Cada valor obtenido será el peso  $\omega_i$ .

Si se utiliza OLS cuando debería emplearse WLS, el estimador  $\hat{\beta}$  sigue siendo insesgado pero no será de mínima varianza.

## Ejemplo 9

En este ejemplo se busca obtener una línea de regresión que permita explicar la influencia lineal de las horas de sueño de un estudiante sobre el score que obtiene en una prueba. Los datos se encuentran en U4\_datos\_2.xlsx

```
datos2 = read_xlsx('U4_datos_2.xlsx')
library(ggplot2)
datos2 |>
  ggplot(aes(x=horas,y=score))+
  geom_point(size=1.5)+
  theme_minimal()
```



```
modelo0 = lm(score ~ horas, data = datos2)
modelo0 |> residuals() |> shapiro.test()
```

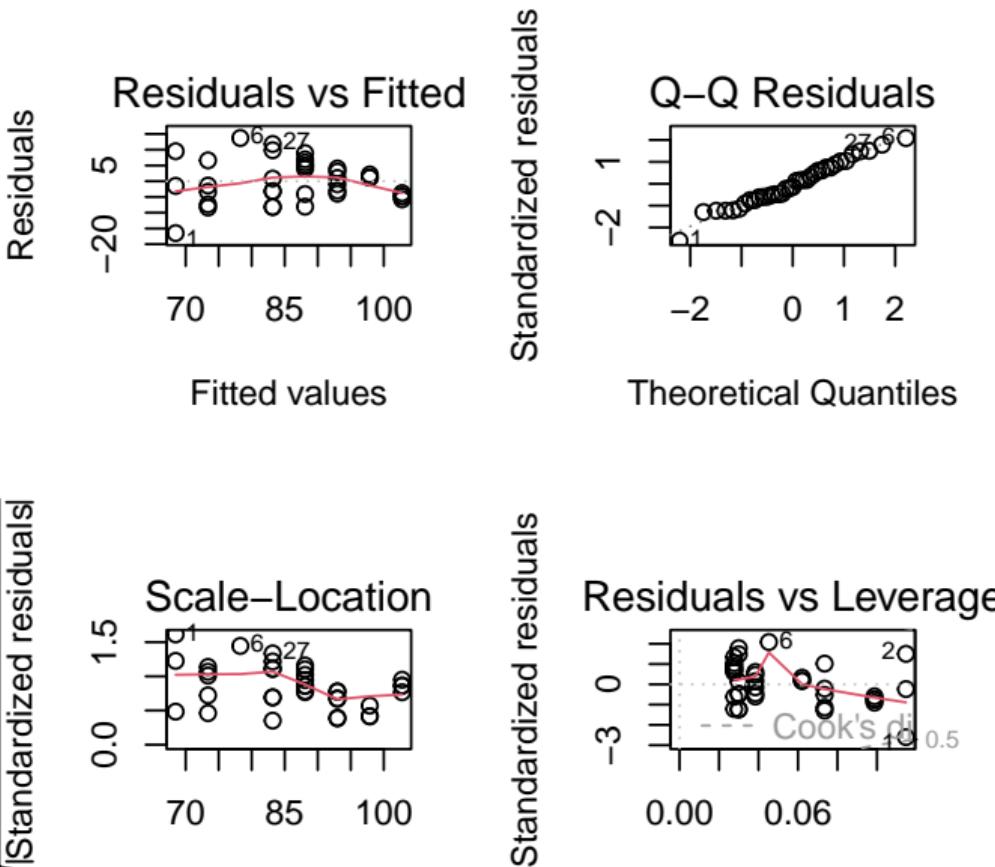
Shapiro-Wilk normality test

```
data: residuals(modelo0)
W = 0.98089, p-value = 0.7621
```

```
modelo0 |> car::ncvTest()
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 6.450996, Df = 1, p = 0.011089
```

```
par(mfrow=c(2,2));plot(modelo0)
```



Primera propuesta: A mayor peso, menor variabilidad de los errores, es decir  $V(\epsilon_i) = \frac{\sigma^2}{x_i}$ . De ahí que  $V_i = \frac{1}{x_i}$  y  $\omega_i = x_i$ .

```
datos2$horas -> peso1  
lm(score ~ horas, data = datos2, weights=peso1) -> modelo1  
modelo1 |> residuals() |> shapiro.test()
```

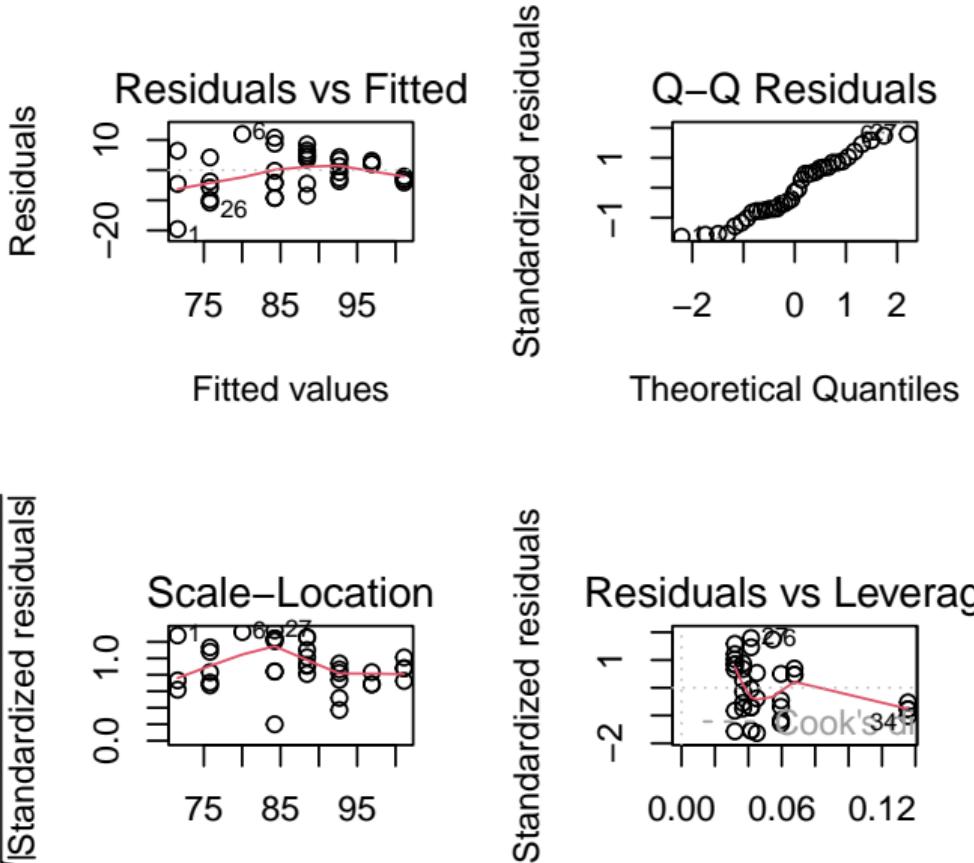
Shapiro-Wilk normality test

```
data: residuals(modelo1)  
W = 0.973, p-value = 0.4957
```

```
modelo1 |> car::ncvTest()
```

Non-constant Variance Score Test  
Variance formula: ~ fitted.values  
Chisquare = 1.12577, Df = 1, p = 0.28868

```
par(mfrow=c(2,2));plot(modelo1)
```



¿Qué pasaría si colocamos  $\omega_i = \frac{1}{x_i}$  en vez de  $\omega_i = x_i$

```
1/datos2$horas -> peso2  
lm(score ~ horas, data = datos2, weights=peso2) -> modelo2  
modelo2 |> residuals() |> shapiro.test()
```

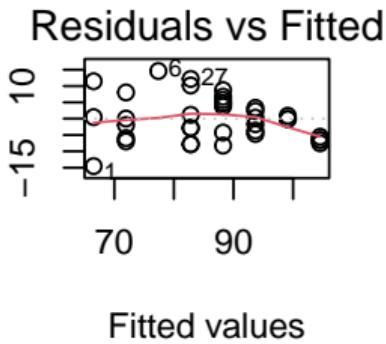
Shapiro-Wilk normality test

```
data: residuals(modelo2)  
W = 0.97621, p-value = 0.6003  
modelo2 |> car::ncvTest()
```

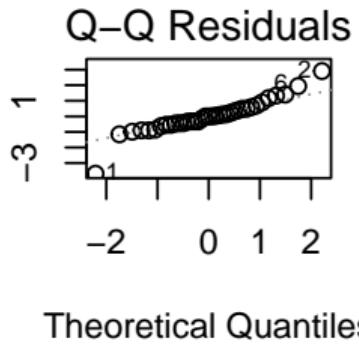
Non-constant Variance Score Test  
Variance formula: ~ fitted.values  
Chisquare = 23.02006, Df = 1, p = 1.6032e-06

```
par(mfrow=c(2,2));plot(modelo2)
```

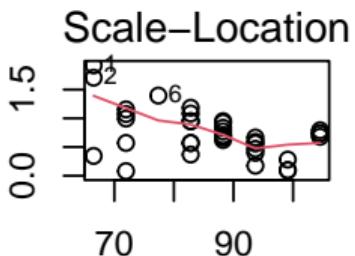
Residuals



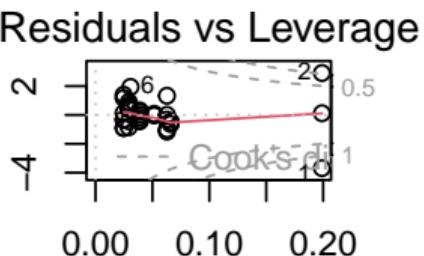
Standardized residuals



$\sqrt{|\text{Standardized residuals}|}$



Standardized residuals



Segunda propuesta: La inversa al cuadrado de los valores ajustados resultantes de la regresión de los residuales absolutos en función de los valores ajustados de la regresión por mínimos cuadrados ordinarios.

```
1/lm(abs(modelo0$residuals) ~ modelo0$fitted.values)$fitted.values**2 -> peso3  
lm(score ~ horas, data = datos2, weights=peso3) -> modelo3  
modelo3 |> residuals() |> shapiro.test()
```

Shapiro-Wilk normality test

```
data: residuals(modelo3)  
W = 0.97271, p-value = 0.4865  
modelo3 |> car::ncvTest()
```

```
Non-constant Variance Score Test  
Variance formula: ~ fitted.values  
Chisquare = 1.592739, Df = 1, p = 0.20694
```

```
par(mfrow=c(2,2));plot(modelo3)
```

