

Regresión Normal Heteroscedástica

Carlos Mario Castaño Suaza

Juan Pablo Lara Chaves

María Paula Camargo Rincón

Modelo de regresión lineal

Sean Y_1, Y_2, \dots, Y_n variables aleatorias independientes tales que para cada $i = 1, \dots, n$,

$$Y_i = \mu_i + \varepsilon_i,$$

donde

$$\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2),$$

independientes entre sí.

Además,

$$\mu_i = \mathbf{x}_i^\top \beta, \quad \ln \sigma_i^2 = \mathbf{z}_i^\top \gamma,$$

donde \mathbf{x}_i y \mathbf{z}_i son vectores de covariables correspondientes al individuo i , asociados a la media y a la varianza respectivamente, y

$$\beta \in \mathbb{R}^p, \quad \gamma \in \mathbb{R}^q$$

son vectores columna de parámetros.

El uso del logaritmo en la varianza asegura que $\sigma_i^2 > 0$ para todo i .

Los parámetros β representan el efecto de los predictores sobre la media condicional de Y_i dado \mathbf{x}_i , mientras que γ representan el efecto de los predictores sobre el logaritmo de la varianza condicional de Y_i dado \mathbf{z}_i .

Si $\gamma = \mathbf{0}$, el modelo se reduce al modelo clásico de regresión lineal homocedástico.

Estimación

Sea $\theta = (\beta, \gamma)$ el vector de parámetros.

La función de densidad para una observación y_i es:

$$f_\theta(y_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_i - \mathbf{x}_i^\top \beta)^2}{2\sigma_i^2}\right),$$

donde $\sigma_i^2 = \exp(\mathbf{z}_i^\top \gamma)$.

Por el supuesto de independencia, la función de verosimilitud para la muestra completa es:

$$L(\beta, \gamma) = \prod_{i=1}^n f_\theta(y_i) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_i - \mathbf{x}_i^\top \beta)^2}{2\sigma_i^2}\right).$$

La log-verosimilitud es:

$$\begin{aligned} \ell(\beta, \gamma) &= \log L(\beta, \gamma) \\ &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^n \log \sigma_i^2 - \frac{1}{2} \sum_{i=1}^n \frac{(y_i - \mathbf{x}_i^\top \beta)^2}{\sigma_i^2} \\ &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^n \mathbf{z}_i^\top \gamma - \frac{1}{2} \sum_{i=1}^n \frac{(y_i - \mathbf{x}_i^\top \beta)^2}{\exp(\mathbf{z}_i^\top \gamma)} \end{aligned}$$

$$\ell(\beta, \gamma) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^n \log \sigma_i^2 - \frac{1}{2} \sum_{i=1}^n \frac{(y_i - \mu_i)^2}{\sigma_i^2}$$

$$\ell(\beta, \gamma) \propto -\frac{1}{2} \sum_{i=1}^n \left[\log \sigma_i^2 + \frac{(y_i - \mu_i)^2}{\sigma_i^2} \right].$$

Las derivadas parciales de la log-verosimilitud son:

$$\begin{aligned}\frac{\partial \ell}{\partial \beta} &= \sum_{i=1}^n \frac{y_i - \mu_i}{\sigma_i^2} \mathbf{x}_i, \\ \frac{\partial \ell}{\partial \gamma} &= -\frac{1}{2} \sum_{i=1}^n \mathbf{z}_i \left(1 - \frac{(y_i - \mu_i)^2}{\sigma_i^2} \right), \\ \frac{\partial^2 \ell}{\partial \beta \partial \beta^\top} &= -\sum_{i=1}^n \frac{1}{\sigma_i^2} \mathbf{x}_i \mathbf{x}_i^\top, \\ \frac{\partial^2 \ell}{\partial \gamma \partial \gamma^\top} &= -\frac{1}{2} \sum_{i=1}^n \mathbf{z}_i \mathbf{z}_i^\top \frac{(y_i - \mu_i)^2}{\sigma_i^2}, \\ \frac{\partial^2 \ell}{\partial \beta \partial \gamma^\top} &= -\sum_{i=1}^n \frac{(y_i - \mu_i)}{\sigma_i^2} \mathbf{x}_i \mathbf{z}_i^\top.\end{aligned}$$

Para construir la matriz de Información de Fisher, tomamos valores esperados de las derivadas parciales de la logverosimilitud

$$\begin{aligned}\mathbb{E} \left[\frac{\partial^2 \ell}{\partial \beta \partial \beta^\top} \right] &= -\sum_{i=1}^n \frac{1}{\sigma_i^2} \mathbf{x}_i \mathbf{x}_i^\top, \\ \mathbb{E} \left[\frac{\partial^2 \ell}{\partial \gamma \partial \gamma^\top} \right] &= -\frac{1}{2} \sum_{i=1}^n \mathbf{z}_i \mathbf{z}_i^\top, \\ \mathbb{E} \left[\frac{\partial^2 \ell}{\partial \beta \partial \gamma^\top} \right] &= -\sum_{i=1}^n \mathbb{E} \left[\frac{y_i - \mu_i}{\sigma_i^2} \right] \mathbf{x}_i \mathbf{z}_i^\top = \mathbf{0}.\end{aligned}$$

$$\begin{aligned}\mathcal{J}(\theta) &= -\mathbb{E} \begin{bmatrix} \frac{\partial^2 \ell}{\partial \beta \partial \beta^\top} & \frac{\partial^2 \ell}{\partial \beta \partial \gamma^\top} \\ \frac{\partial^2 \ell}{\partial \gamma \partial \beta^\top} & \frac{\partial^2 \ell}{\partial \gamma \partial \gamma^\top} \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^n \frac{1}{\sigma_i^2} \mathbf{x}_i \mathbf{x}_i^\top & \mathbf{0} \\ \mathbf{0} & \frac{1}{2} \sum_{i=1}^n \mathbf{z}_i \mathbf{z}_i^\top \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{X}^\top \text{diag} \left(\frac{1}{\sigma_1^2}, \dots, \frac{1}{\sigma_n^2} \right) \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}^\top \text{diag} \left(\frac{1}{2}, \dots, \frac{1}{2} \right) \mathbf{Z} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{J}_\beta & \mathbf{0} \\ \mathbf{0} & \mathcal{J}_\gamma \end{bmatrix}\end{aligned}$$

Algoritmo de Fisher Scoring

Ecuación principal del algoritmo Fisher Scoring

$$I^{(k)}(\theta) \theta^{(k+1)} = I^{(k)}(\theta) \theta^{(k)} + q^{(k)},$$

donde

$$\theta = \begin{pmatrix} \beta \\ \gamma \end{pmatrix}, \quad I^{(k)}(\theta) = \begin{pmatrix} I_\beta^{(k)} & \mathbf{0} \\ \mathbf{0} & I_\gamma^{(k)} \end{pmatrix}, \quad q^{(k)} = \begin{pmatrix} q_\beta^{(k)} \\ q_\gamma^{(k)} \end{pmatrix}.$$

Separando en bloques:

$$\begin{aligned} I_{\beta}^{(k)} \beta^{(k+1)} &= I_{\beta}^{(k)} \beta^{(k)} + q_{\beta}^{(k)}, \\ I_{\gamma}^{(k)} \gamma^{(k+1)} &= I_{\gamma}^{(k)} \gamma^{(k)} + q_{\gamma}^{(k)}. \end{aligned}$$

Multiplicando por las inversas:

$$\begin{aligned} \beta^{(k+1)} &= \beta^{(k)} + \left(I_{\beta}^{(k)}\right)^{-1} q_{\beta}^{(k)}, \\ \gamma^{(k+1)} &= \gamma^{(k)} + \left(I_{\gamma}^{(k)}\right)^{-1} q_{\gamma}^{(k)}. \end{aligned}$$

1. Inicializar:

- Escoge un valor inicial $\gamma^{(0)}$
- Calcula las varianzas: $\sigma_i^{2(0)} = \exp(\mathbf{z}_i^{\top} \gamma^{(0)})$
- Define la matriz de pesos: $\mathbf{W}_{\mu}^{(0)} = \text{diag}\left(\frac{1}{\sigma_i^{2(0)}}\right)$

2. Actualizar β :

$$\beta^{(k+1)} = \left(\mathbf{X}^{\top} \mathbf{W}_{\mu}^{(k)} \mathbf{X}\right)^{-1} \mathbf{X}^{\top} \mathbf{W}_{\mu}^{(k)} \mathbf{y}$$

3. Construir pseudorespuesta para γ :

$$\tilde{z}_i^{(k)} = \log\left((y_i - \mathbf{x}_i^{\top} \beta^{(k+1)})^2\right)$$

4. Actualizar γ : (usando pesos constantes $\frac{1}{2}$)

$$\gamma^{(k+1)} = (\mathbf{Z}^{\top} \mathbf{W}_{\sigma} \mathbf{Z})^{-1} \mathbf{Z}^{\top} \mathbf{W}_{\sigma} \tilde{\mathbf{z}}^{(k)}, \quad \text{con } \mathbf{W}_{\sigma} = \frac{1}{2} \mathbf{I}_n$$

5. Actualizar varianzas y pesos:

$$\sigma_i^{2(k+1)} = \exp(\mathbf{z}_i^{\top} \gamma^{(k+1)}), \quad \mathbf{W}_{\mu}^{(k+1)} = \text{diag}\left(\frac{1}{\sigma_i^{2(k+1)}}\right)$$

6. Repetir pasos 2 a 5 hasta convergencia.

1. Inicialización

- Escoge un valor inicial $\gamma^{(0)}$.
- Calcula las varianzas iniciales:

$$\sigma_i^{2(0)} = \exp(\mathbf{z}_i^{\top} \gamma^{(0)}), \quad i = 1, \dots, n$$

- Define la matriz de pesos inicial para la media:

$$\mathbf{W}^{(0)} = \text{diag}\left(\frac{1}{\sigma_i^{2(0)}}\right)$$

2. Pasos iterativos (para $k = 0, 1, 2, \dots$)

a) Actualiza $\beta^{(k+1)}$:

$$\beta^{(k+1)} = \left(\mathbf{X}^{\top} \mathbf{W}^{(k)} \mathbf{X}\right)^{-1} \mathbf{X}^{\top} \mathbf{W}^{(k)} \mathbf{y}$$

donde

$$\mathbf{W}^{(k)} = \text{diag}\left(\frac{1}{\sigma_i^{2(k)}}\right), \quad \sigma_i^{2(k)} = \exp(\mathbf{z}_i^{\top} \gamma^{(k)}).$$

b) Construye la respuesta de trabajo $\tilde{y}_i^{(k)}$:

$$\tilde{y}_i^{(k)} = \eta_i^{(k)} + \frac{1}{\sigma_i^{2(k)}} \left(y_i - \mathbf{x}_i^{\top} \beta^{(k+1)}\right)^2 - 1, \quad \text{donde } \eta_i^{(k)} = \mathbf{z}_i^{\top} \gamma^{(k)}.$$

c) Actualiza $\gamma^{(k+1)}$:

$$\gamma^{(k+1)} = (\mathbf{Z}^\top \mathbf{W}_\sigma \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{W}_\sigma \tilde{\mathbf{y}}^{(k)}, \quad \text{con } \mathbf{W}_\sigma = \frac{1}{2} \mathbf{I}_n.$$

d) Actualiza las varianzas y la matriz de pesos para la siguiente iteración:

$$\sigma_i^{2(k+1)} = \exp(\mathbf{z}_i^\top \gamma^{(k+1)}), \quad \mathbf{W}^{(k+1)} = \text{diag} \left(\frac{1}{\sigma_i^{2(k+1)}} \right).$$

3. **Repetir** los pasos 2a) a 2d) hasta que se cumpla el criterio de convergencia

$$\|\theta^{(k+1)} - \theta^{(k)}\|_2 < \varepsilon,$$

donde $\theta = (\beta^\top, \gamma^\top)^\top$ y $\varepsilon > 0$ es un umbral pequeño.

Implementación del algoritmo de Fisher Scoring

```
# Algoritmo de Fisher Scoring para Modelo Normal con Heteroscedasticidad

# y: vector de variable respuesta (tamaño n)
# X: matriz de covariables para la media (n x p)
# Z: matriz de covariables para la varianza (n x q)
# beta inicial: valor inicial para beta
# gamma inicial: valor inicial para beta
# tolerancia: valor epsilon para determinar la convergencia 1e-6
# max_iteraciones: número máximo de iteraciones del algoritmo
# progreso: mostrar iteraciones

fisher_scoring <- function(y, X, Z,
                           beta_inicial = NULL,
                           gamma_inicial = NULL,
                           tolerancia = 1e-6,
                           max_iteraciones = 100) {

  n <- length(y)      # número de observaciones
  p <- ncol(X)         # número de parámetros beta
  q <- ncol(Z)         # número de parámetros gamma

  # PASO 1: INICIALIZACIÓN

  # Valores iniciales
  if (is.null(beta_inicial)) {
    beta <- rep(0, p) # vector de ceros si no se proporcionan valores iniciales
  } else {
    beta <- beta_inicial
  }

  if (is.null(gamma_inicial)) {
    gamma <- rep(0, q)
  } else {
    gamma <- gamma_inicial
  }

  # Calcular varianzas iniciales: sigma_i^2 = exp(Z_i' * gamma)
  eta <- as.vector(Z %*% gamma)
  sigma2 <- exp(eta)

  # Matriz de pesos inicial: W = diagonal(1/sigma_i^2)
  W <- diag(1 / sigma2)
```

```

# iteramos
iteracion <- 0
convergio <- FALSE

# PASO 2: BUCLE PRINCIPAL DE ITERACIONES

while (iteracion < max_iteraciones && !convergio) {

  iteracion <- iteracion + 1

  # Guardar valores anteriores para verificar convergencia
  beta_anterior <- beta
  gamma_anterior <- gamma

  # a) ACTUALIZAR BETA
  #  $\beta_{\text{nuevo}} = (X' * W * X)^{-1} * X' * W * y$ 

  cov_beta <- solve(t(X) %*% W %*% X)
  XtWy <- t(X) %*% W %*% y
  beta <- cov_beta %*% XtWy

  # b) CONSTRUIR RESPUESTA DE TRABAJO Y_TILDE
  #  $y_{\text{tilde}_i} = \eta_i + (1/\sigma_i^2) * (y - X\beta)^2 - 1$ 

  eta <- as.vector(Z %*% gamma)
  residuos <- y - X %*% beta
  y_tilde <- eta + (residuos^2) / sigma2 - 1

  # c) ACTUALIZAR GAMMA
  #  $W_{\text{sigma}} = (1/2) * \text{Identidad}$ 
  W_sigma <- diag(rep(0.5, n))

  #  $\gamma_{\text{nuevo}} = (Z' * W_{\text{sigma}} * Z)^{-1} * Z' * W_{\text{sigma}} * y_{\text{tilde}}$ 
  cov_gamma <- solve(t(Z) %*% W_sigma %*% Z)
  ZtWy_tilde <- t(Z) %*% W_sigma %*% y_tilde
  gamma <- cov_gamma %*% ZtWy_tilde

  # d) ACTUALIZAR VARIANZAS Y PESOS PARA LA SIGUIENTE ITERACIÓN

  eta <- as.vector(Z %*% gamma)
  sigma2 <- exp(eta)
  W <- diag(1 / sigma2)

  # VERIFICAR CONVERGENCIA
  # Criterio:  $\| \text{parametros\_nuevos} - \text{parametros\_anteriores} \|_2 < \text{tolerancia}$ 

  parametros_anteriores <- c(beta_anterior, gamma_anterior)
  parametros_nuevos <- c(beta, gamma)
  diferencia <- sqrt(sum((parametros_nuevos - parametros_anteriores)^2))

  if (diferencia < tolerancia) {
    convergio <- TRUE
  }
}

# Advertencia si no convergió
if (!convergio) {
  warning(sprintf("No convergió en %d iteraciones", max_iteraciones))
}

# CALCULAR RESULTADOS FINALES

```

```

# Valores predichos y residuales finales
valores_ajustados <- as.vector(X %*% beta)
residuales_finales <- y - valores_ajustados
residuales_estandarizados <- residuales_finales / sqrt(sigma2)

# RETORNAR RESULTADOS

resultado <- list(
  # parámetros estimados
  beta = as.vector(beta),
  gamma = as.vector(gamma),

  # varianzas estimadas
  sigma2 = as.vector(sigma2),

  # errores estándar
  se_beta = sqrt(diag(cov_beta)),
  se_gamma = sqrt(diag(cov_gamma)),

  # valores ajustados y residuales
  valores_ajustados = valores_ajustados,
  residuales = residuales_finales,
  residuales_estandarizados = residuales_estandarizados
)

return(resultado)
}

```

Ejemplo

```

data(mtcars)

y <- mtcars$mpg # millas por galón
X <- cbind(1, mtcars$wt) # intercepto + peso
Z <- cbind(1, mtcars$hp) # intercepto + caballos de fuerza

# Fisher Scoring
resultado <- fisher_scoring(y, X, Z)

```

```

library(knitr)

# Tabla para beta con std error
beta_df <- data.frame(
  Parámetro = paste0("Beta_", 0:(length(resultado$beta)-1)),
  Estimación = resultado$beta,
  `Error Std` = resultado$se_beta
)

kable(beta_df, caption = "Estimación del vector Beta con Error Estándar", digits = 4)

# Tabla para gamma con std error
gamma_df <- data.frame(
  Parámetro = paste0("Gamma_", 0:(length(resultado$gamma)-1)),
  Estimación = resultado$gamma,
  `Error Std` = resultado$se_gamma
)

kable(gamma_df, caption = "Estimación del vector Gamma con Error Estándar", digits = 4)

```

Table 1: Estimación del vector Beta con Error Estándar

Parámetro	Estimación	Error.Std
Beta_0	37.5141	1.8060
Beta_1	-5.3992	0.5423

Table 2: Estimación del vector Gamma con Error Estándar

Parámetro	Estimación	Error.Std
Gamma_0	2.0638	0.5982
Gamma_1	0.0007	0.0037

```
# Usando la función gls de nlme

library(nlme)

gls_model <- gls(mpg ~ wt, data = mtcars,
                weights = varExp(form = ~ hp))
coef_summary <- summary(gls_model)$tTable

coef(gls_model)
```

```
(Intercept)          wt
   37.523052    -5.401375
```

```
(std_errors <- coef_summary[, "Std.Error"])
```

```
(Intercept)          wt
   1.8647283    0.5601475
```

Ahora usaremos los datos de los Cherry Tree, en los que se tiene como variable respuesta el volumen de madera obtenida de un árbol(V), y como variables predictoras tenemos La altura del árbol (H) y el diámetro del tronco (D).

```
D <- c(8.3, 8.6, 8.8, 10.2, 10.7, 10.8, 11.0, 11.0, 11.1, 11.2,
      11.3, 11.4, 11.4, 11.7, 12.0, 12.2, 12.9, 13.3, 13.7, 13.8,
      14.0, 14.2, 14.5, 16.0, 16.3, 17.3, 17.5, 17.9, 18.0, 18.0,
      20.6)

H <- c(70, 65, 63, 72, 81, 83, 66, 75, 80, 75,
      79, 76, 76, 69, 75, 74, 85, 86, 71, 64,
      78, 80, 74, 72, 77, 81, 82, 80, 80, 74,
      87)

V <- c(10.3, 10.3, 10.2, 16.4, 18.8, 19.7, 15.6, 18.2, 22.6, 19.9,
      24.2, 21.0, 21.4, 21.3, 19.1, 22.2, 33.8, 27.4, 25.7, 24.9,
      34.5, 31.7, 36.3, 38.3, 42.6, 55.4, 55.7, 58.3, 51.5, 51.0,
      77.0)

datos <- data.frame(D, H, V)
```

```
library(GGally)
```

Cargando paquete requerido: ggplot2

```
Registered S3 method overwritten by 'GGally':
  method from
+.gg      ggplot2
```

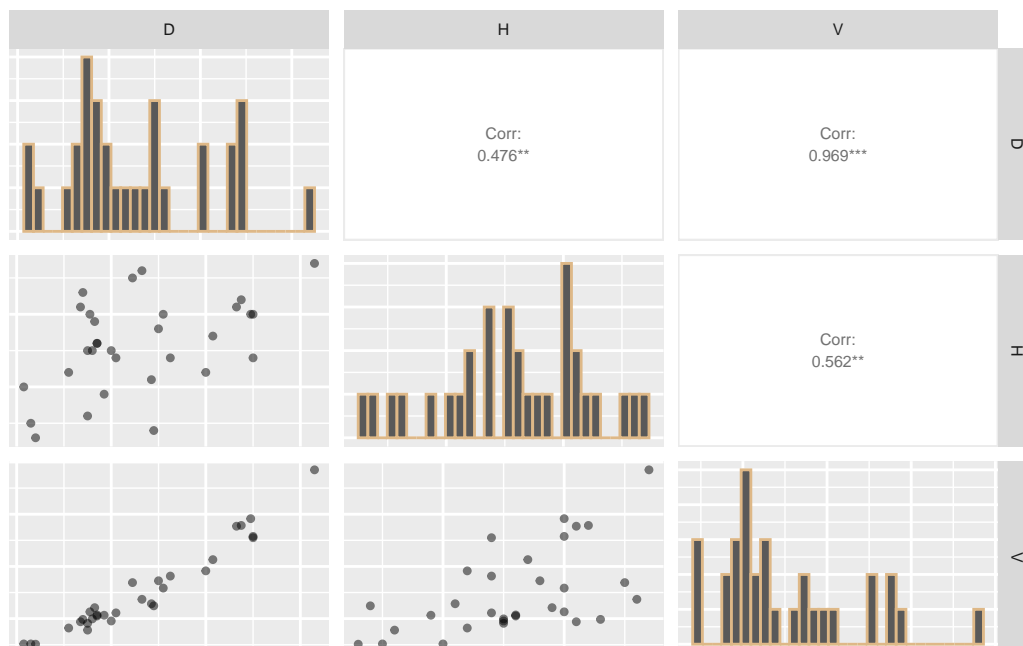
```
datos %>%
  ggpairs(
    upper = list(continuous = wrap("cor", size = 2)),
    diag = list(continuous = wrap("barDiag", colour = "burlywood")),
    lower = list(continuous = wrap("points", alpha = 0.5, shape = 20,
                                   fill = "lightblue")),

    axisLabels = "none") +
  theme(strip.text = element_text(size = 6))
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Se observa que las variables predictoras presentan una correlación moderadamente alta entre sí, pero también vemos que presentan una correlación moderadamente alta entre el volumen y la altura, y una correlación alta entre el volumen y el diámetro, recordando que el volumen es la variable respuesta, este resultado es bueno.

Se realizará el algoritmo con $X = Z$ mientras observamos como se comportan los residuales de este modelo.

```
X <- model.matrix(~., data = datos[, -3])
fisher_tree <- fisher_scoring(y = V, X = X, Z = X)
```

	$\hat{\beta}$	Desviación estandar β	$\hat{\gamma}$	Desviación estandar γ
Intercepto	-31.6758447	2.50389857	-13.33879116	3.13029162
Diámetro (D)	3.0145332	0.16278569	0.64004059	0.09314556
Altura (H)	0.2423712	0.04435291	0.09058667	0.04633345

Vamos a revisar los residuales:

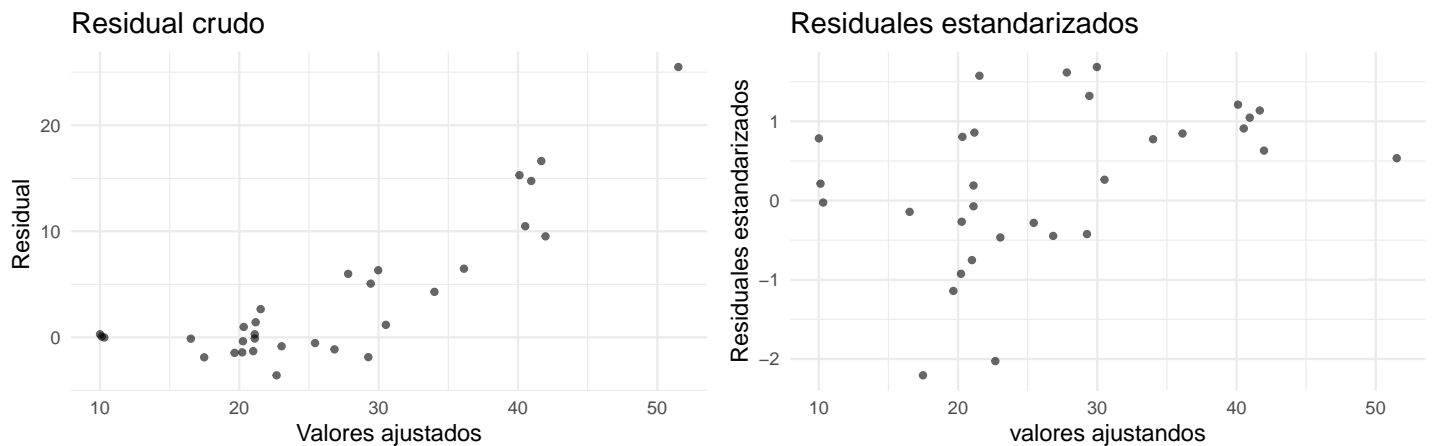
```
df_cherry_tree <- data.frame(
  valores_ajustados=fisher_tree$valores_ajustados,
  residuales=fisher_tree$residuales,
  residuales_estandarizados=fisher_tree$residuales_estandarizados,
  H=datos$H,
  D=datos$D)
```

```
res <- ggplot(df_cherry_tree, aes(x=valores_ajustados, y=residuales)) +
  geom_point(color='black', alpha=0.6) +
  labs(title='Residual crudo', x='Valores ajustados', y='Residual') +
  theme_minimal(13)

res_est <- ggplot(df_cherry_tree, aes(x=valores_ajustados, y=residuales_estandarizados)) +
  geom_point(color='black', alpha=0.6) +
  labs(title='Residuales estandarizados', x='valores ajustados', y='Residuales estandarizados') +
  theme_minimal(13)
```



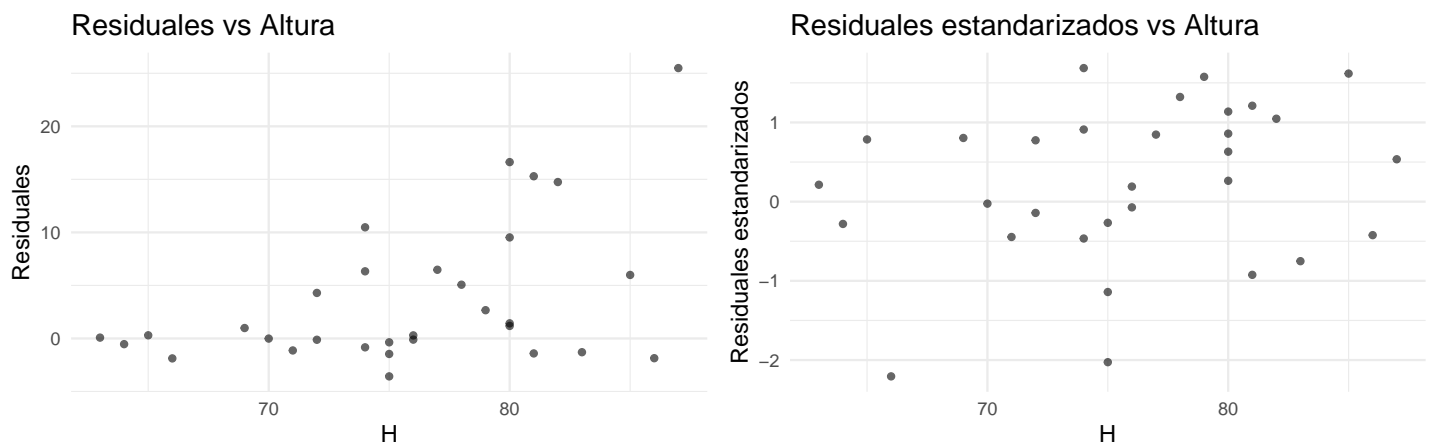
```
res; res_est
```



Aunque en los residuales crudos, podemos ver que hay una relación creciente, como denotando la falta de algún término cuadrático o de algún problema de heteroscedasticidad, los residuales estandarizados son los mejores para interpretar en este caso, ya que esos son más robustos, en estos residuales estandarizados no se refleja un patrón no explicado, y están casi centrados en 0, lo que indica que el modelo parece estar reflejando bien la variabilidad.

```
res_H <- ggplot(df_cherry_tree, aes(x = H, y = residuales)) +  
  geom_point(color = 'black', alpha = 0.6) +  
  labs(title = 'Residuales vs Altura', x = 'H', y = 'Residuales') +  
  theme_minimal(13)  
res_Hest <- ggplot(df_cherry_tree, aes(x = H, y = residuales_estandarizados)) +  
  geom_point(color = 'black', alpha = 0.6) +  
  labs(title = 'Residuales estandarizados vs Altura', x = 'H', y = 'Residuales estandarizados') +  
  theme_minimal(13)
```

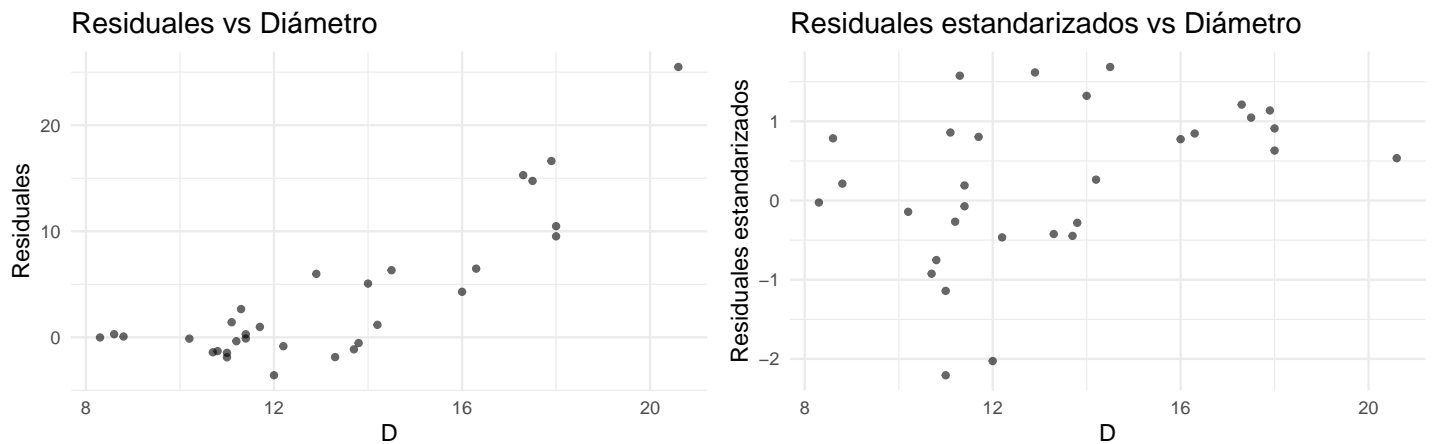
```
res_H; res_Hest
```



Nuevamente nos enfocaremos en los residuales estandarizados, se observa que los residuales parecen estar un poco más centrados en 0 y nuevamente no presentan un patrón no explicado, lo que indica que el modelo está captando bien la variable Altura (H).

```
res_D <- ggplot(df_cherry_tree, aes(x = D, y = residuales)) +  
  geom_point(color = 'black', alpha = 0.6) +  
  labs(title = 'Residuales vs Diámetro', x = 'D', y = 'Residuales') +  
  theme_minimal(13)  
res_Dest <- ggplot(df_cherry_tree, aes(x = D, y = residuales_estandarizados)) +  
  geom_point(color = 'black', alpha = 0.6) +  
  labs(title = 'Residuales estandarizados vs Diámetro', x = 'D', y = 'Residuales estandarizados') +  
  theme_minimal(13)
```

```
res_D; res_Dest
```



Nuevamente se puede ver que los residuales estan todavía más centrados en 0 que con H, y no se ve un patrón no explicado. Se decidió realizar un modelo en el que las covariables para la regresión de la varianza será unicamente el diámetro, esto debido a su efecto o estimación, a su desviación estándar, su alta correlación con la variable respuesta y a los gráficos de los residuales.

```
Z <- model.matrix(~D,data = datos)
fisher_tree_D <- fisher_scoring(y = V,X = X,Z = Z)
fisher_tree_D$se_gamma
```

```
(Intercept)      D
1.11188675  0.08190644
```

	$\hat{\beta}$	Desviación estandar β	$\hat{\gamma}$	Desviación estandar γ
Intercepto	-32.1888152	2.98286409	-6.4667733	1.11188675
Diámetro (D)	3.2353858	0.21520718	0.6469465	0.08190644
Altura (H)	0.2191107	0.05443738		

```
library(nlme)
gls_model_tree <- gls(V~.,data = datos,weights = varExp(form = ~ D))
summary(gls_model_tree)
```

Generalized least squares fit by REML

Model: V ~ .

Data: datos

```
AIC      BIC      logLik
168.7432 175.4042 -79.37159
```

Variance function:

Structure: Exponential of variance covariate

Formula: ~D

Parameter estimates:

```
expon
0.2885111
```

Coefficients:

```
Value Std.Error t-value p-value
(Intercept) -32.80477 3.493947 -9.389028 0.000
D             3.34181 0.236866 14.108471 0.000
H             0.21285 0.062307 3.416135 0.002
```

Correlation:
 (Intr) D
 D 0.119
 H -0.843 -0.629

Standardized residuals:
 Min Q1 Med Q3 Max
 -1.9731417 -0.6263599 0.2237038 0.8395404 1.9764716

Residual standard error: 0.06613217
 Degrees of freedom: 31 total; 28 residual

	$\hat{\beta}$	Desviación estandar β	p_value
Intercepto	-32.80477	3.493947	-0.000
Diámetro (D)	3.34181	0.236866	0.000
Altura (H)	0.21285	0.062307	0.002

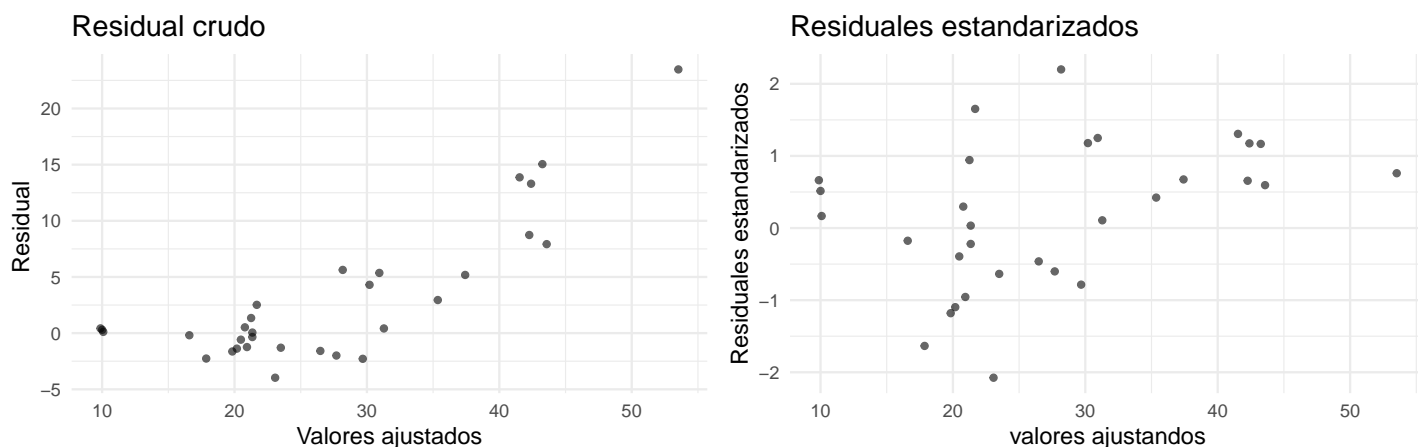
Todas las variables aparecen como significativas, además de esto los resultados son bastante similares con la función de “fisher_scoring”.

```
df_cherry_tree_D <- data.frame(
  valores_ajustados=fisher_tree_D$valores_ajustados,
  residuales=fisher_tree_D$residuales,
  residuales_estandarizados=fisher_tree_D$residuales_estandarizados,
  H=datos$H,
  D=datos$D)
```

```
res <- ggplot(df_cherry_tree_D, aes(x =valores_ajustados , y = residuales)) +
  geom_point(color = 'black', alpha = 0.6) +
  labs(title = 'Residual crudo', x = 'Valores ajustados', y = 'Residual ') +
  theme_minimal(13)

res_est <- ggplot(df_cherry_tree_D, aes(x = valores_ajustados, y = residuales_estandarizados)) +
  geom_point(color = 'black', alpha = 0.6) +
  labs(title = 'Residuales estandarizados', x = 'valores ajustandos', y = 'Residuales estandarizados') +
  theme_minimal(13)
```

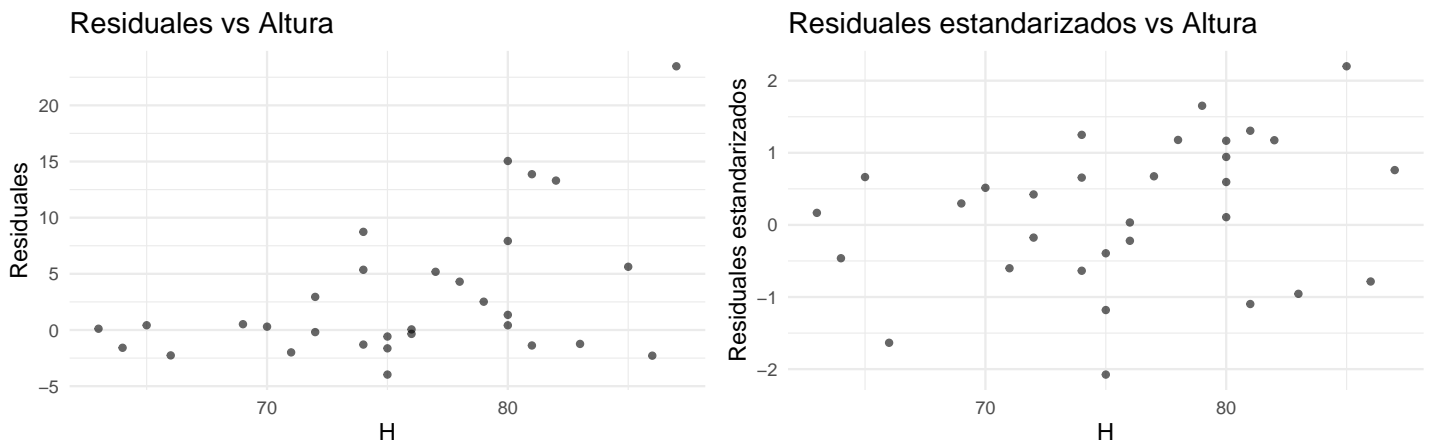
```
res; res_est
```



A comparación con el modelo anterior los esiduales estandarizados vs los valores ajustados se ven más centrados en 0, nuevamente no se ve un patrón no explicado.

```
res_H <- ggplot(df_cherry_tree_D, aes(x = H, y = residuales)) +
  geom_point(color = 'black', alpha = 0.6) +
  labs(title = 'Residuales vs Altura', x = 'H', y = 'Residuales') +
  theme_minimal(13)
res_Hest <- ggplot(df_cherry_tree_D, aes(x = H, y = residuales_estandarizados)) +
  geom_point(color = 'black', alpha = 0.6) +
  labs(title = 'Residuales estandarizados vs Altura', x = 'H', y = 'Residuales estandarizados') +
  theme_minimal(13)
```

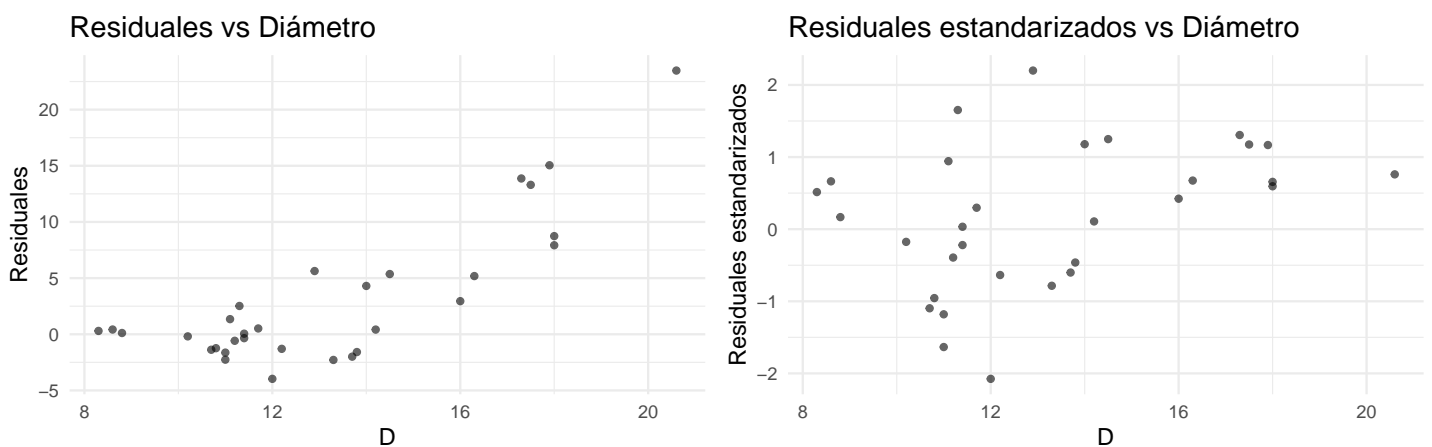
```
res_H; res_Hest
```



Nuevamente no se ve un patrón no explicado, y los residuales se ven más centrados en 0 a comparación con el modelo anterior.

```
res_D <- ggplot(df_cherry_tree_D, aes(x = D, y = residuales)) +
  geom_point(color = 'black', alpha = 0.6) +
  labs(title = 'Residuales vs Diámetro', x = 'D', y = 'Residuales') +
  theme_minimal(13)
res_Dest <- ggplot(df_cherry_tree_D, aes(x = D, y = residuales_estandarizados)) +
  geom_point(color = 'black', alpha = 0.6) +
  labs(title = 'Residuales estandarizados vs Diámetro', x = 'D', y = 'Residuales estandarizados') +
  theme_minimal(13)
```

```
res_D; res_Dest
```



Nuevamente los residuales se ven mucho más centrados en 0 a comparación con el modelo anterior, no se ve un patrón no explicado.

Aunque la función de r no acepte dentro de su formula más de una covariable para estimar la varianza, se tomaron los resultados de los criterios, que se muestran a continuación

Modelo	AIC	BIC	$\sum \hat{e}_i^2$
Z=D+H	177.0761	183.7371	657.308
Z=H	178.7115	185.3726	404.3394
Z=D	168.7432	175.4042	

1302.844 | Se puede ver que el modelo que obtuvo valores más bajos en los criterios de predictividad fue el último modelo realizado donde la única covariable para estimar la varianza es Diámetro (D), como estos criterios penalizan la complejidad quiere decir que este modelo es un buen balance entre simplicidad y capacidad de modelar la varianza, a pesar de que fue el que obtuvo la suma de cuadrados de los residuales más alta.