Análisis de regresión

Capítulo 9: Modelos polinomiales

Mg. Sc. J. Eduardo Gamboa U.

Introducción

Hemos visto que es posible construir modelos de regresión lineal polinomiales de la siguiente forma:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_k x^k + \epsilon$$

Sin embargo, mientras más términos polinomiales se tengan, aumenta el riesgo de multicolinealidad. Para evitar ese problema, se propone trabajar con el siguiente modelo:

$$y = \alpha_0 + \alpha_1 P_1(x) + \alpha_2 P_2(x) + \cdots + \alpha_k P_k(x) + \epsilon$$

donde $P_u(x)$ es un polinomio ortogonal de u-ésimo orden, los cuales están sujetos a estas restricciones:

$$P_0(x) = 1 \quad \text{y} \quad \sum_{i=1}^{n} P_r(x_i)\, P_s(x_i) = 0, \qquad r \neq s, \qquad r, s = 0, 1, \ldots, k$$

Esta última expresión es conocida como la propiedad de ortogonalidad.

Los valores de estos polinomios se obtienen de la siguiente manera:

$$P_0(x_i) = 1$$

$$P_1(x_i) = \lambda_1 \left(\frac{x_i - \bar{x}}{d} \right)$$

$$P_2(x_i) = \lambda_2 \left[\left(\frac{x_i - \bar{x}}{d} \right)^2 - \left(\frac{n^2 - 1}{12} \right) \right]$$

$$P_3(x_i) = \lambda_3 \left[\left(\frac{x_i - \bar{x}}{d} \right)^3 - \left(\frac{x_i - \bar{x}}{d} \right) \left(\frac{3n^2 - 7}{20} \right) \right]$$

$$P_4(x_i) = \lambda_4 \left[\left(\frac{x_i - \bar{x}}{d} \right)^4 - \left(\frac{x_i - \bar{x}}{d} \right)^2 \left(\frac{3n^2 - 13}{14} \right) + \frac{3(n^2 - 1)(n^2 - 9)}{560} \right]$$

donde n es el tamaño de muestra, d es el espaciamiento entre los valores de x, y los valores de λ_j son constantes ya establecidas para cada P_j y n.

En consecuencia, el modelo estará expresado matricialmente de la siguiente manera:

$$\mathbf{y} = X\alpha + \epsilon$$

donde la matriz X tendrá la siguiente forma:

$$X = \begin{pmatrix} P_0(x_1) & P_1(x_1) & \cdots & P_k(x_1) \\ P_0(x_2) & P_1(x_2) & \cdots & P_k(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ P_0(x_n) & P_1(x_n) & \cdots & P_k(x_n) \end{pmatrix}$$

Luego, dada la ortogonalidad de los polinomios, el valor de $X^\top X$ es:

$$X^\top X = \begin{pmatrix} \sum_{i=1}^n P_0^2(x_i) & 0 & \cdots & 0 \\ 0 & \sum_{i=1}^n P_1^2(x_i) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{i=1}^n P_k^2(x_i) \end{pmatrix}$$

Mientras que $X^\top \mathbf{y}$:

$$X^\top \mathbf{y} = \begin{pmatrix} \sum_{i=1}^n P_0(x_i)\, y_i \\ \sum_{i=1}^n P_1(x_i)\, y_i \\ \vdots \\ \sum_{i=1}^n P_k(x_i)\, y_i \end{pmatrix}$$

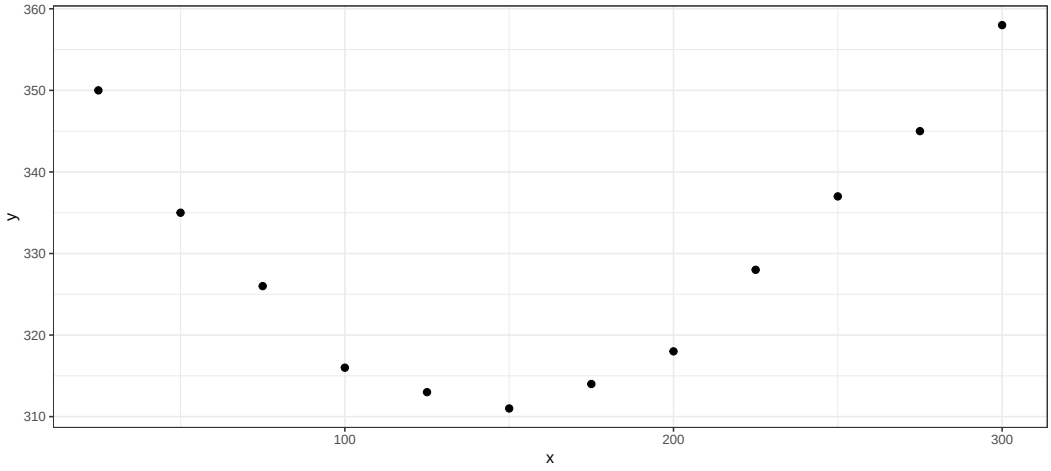
De aquí se puede notar que, al aplicar la expresión para estimar puntualmente α, tenemos:

$$\hat{\alpha}_j = \frac{\sum_{i=1}^{n} P_j(x_i)\, y_i}{\sum_{i=1}^{n} P_j^2(x_i)}, \qquad j = 0, 1, \ldots, k$$

Un analista de investigación de operaciones se encuentra desarrollando un modelo de cómputo que permita tomar en cuenta el efecto de diversas cantidades de pedido sobre el costo anual del inventario, el cual sospecha que tiene un comportamiento cuadrático o cúbico Los datos se muestran a continuación:

```
x = seq(25,300,25)
y = c(350,335,326,316,313,311,314,318,328,337,345,358)
datos = data.frame(x,y)
library(ggplot2)
```

```
datos |> ggplot(aes(x,y)) + geom_point(size=2) + theme_bw()
```



Modelo 1: Algunos análisis a partir de un primer modelo. Comente los siguientes resultados

```
lm(y ~ x, datos) -> modelo_1
modelo_1 |> summary()
```

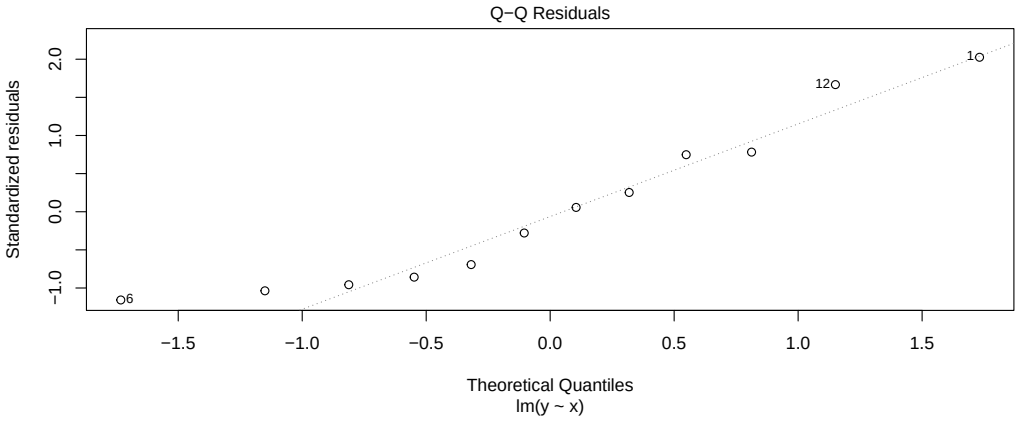
```
Call:
lm(formula = y ~ x, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-17.668 -13.373  -1.668  10.630  27.154

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 321.68182    9.82496  32.741 1.67e-11 ***
x             0.04657    0.05340   0.872    0.404
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.96 on 10 degrees of freedom
Multiple R-squared:  0.07069,   Adjusted R-squared:  -0.02224
F-statistic: 0.7607 on 1 and 10 DF,  p-value: 0.4036
```

```
modelo_1 |> plot(which=2)
```

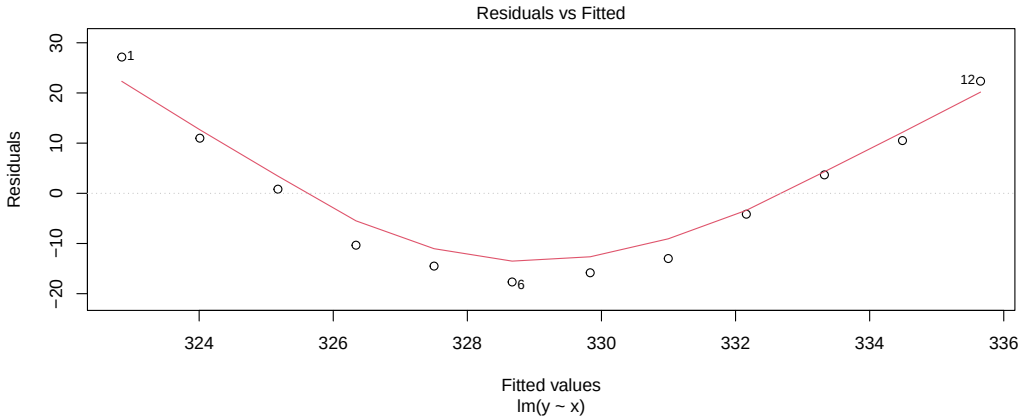


```r
modelo_1 |> residuals() |> shapiro.test()
```

```
    Shapiro-Wilk normality test

data:  residuals(modelo_1)
W = 0.91901, p-value = 0.2778
```

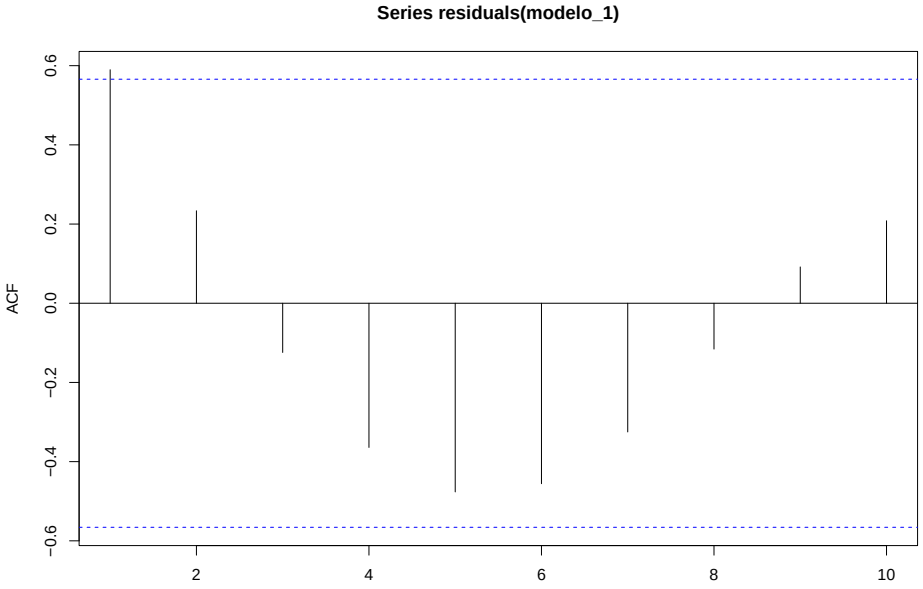
```
modelo_1 |> plot(which=1)
```



```r
modelo_1 |> car::ncvTest()
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.2052885, Df = 1, p = 0.65049
```

```
modelo_1 |> residuals() |> TSA::acf()
```



```
modelo_1 |> lmtest::dwtest()
```

```
    Durbin-Watson test

data:  modelo_1
DW = 0.33522, p-value = 1.298e-07
alternative hypothesis: true autocorrelation is greater than 0
```

```
modelo_1 |> AIC()
```

```
[1] 104.3542
```

Modelo 2: ¿Qué sucede si se añade un componente polinomial de orden 2? Analice los siguientes resultados:

```
lm(y ~ x + I(x**2), datos) -> modelo_2
modelo_2 |> summary()
```

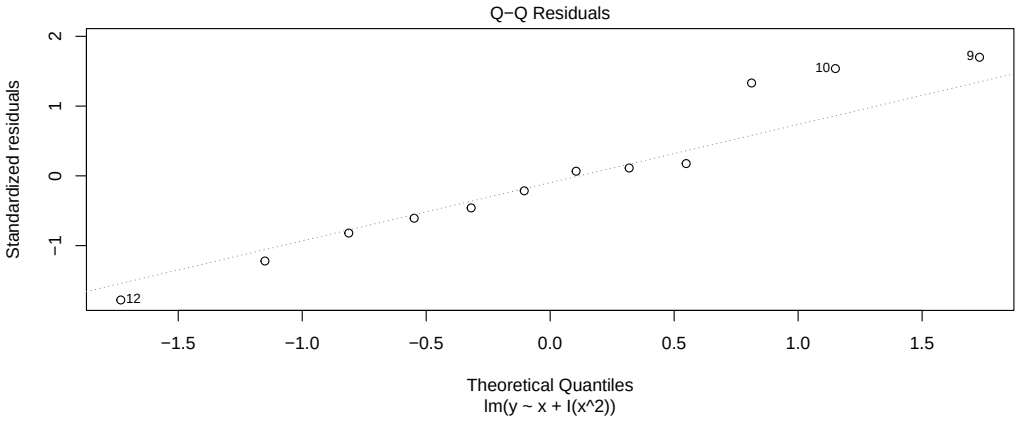
```
Call:
lm(formula = y ~ x + I(x^2), data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7500 -1.3776 -0.1399  0.7911  3.5962

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.632e+02  2.373e+00  153.06  < 2e-16 ***
x           -6.652e-01  3.357e-02  -19.82 9.85e-09 ***
I(x^2)       2.190e-03  1.006e-04   21.78 4.27e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.296 on 9 degrees of freedom
Multiple R-squared:  0.9827,    Adjusted R-squared:  0.9789
F-statistic: 255.6 on 2 and 9 DF,  p-value: 1.178e-08
```

```
modelo_2 |> plot(which=2)
```

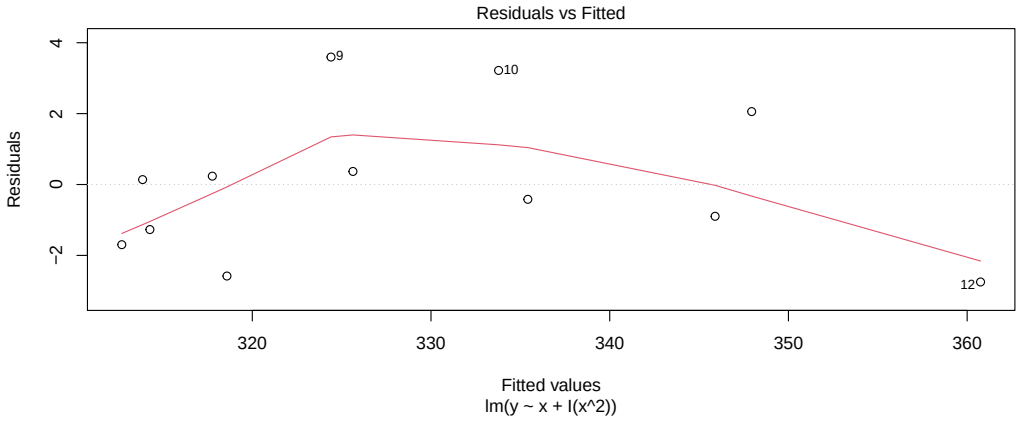


```r
modelo_2 |> residuals() |> shapiro.test()
```

```
    Shapiro-Wilk normality test

data:  residuals(modelo_2)
W = 0.93666, p-value = 0.456
```

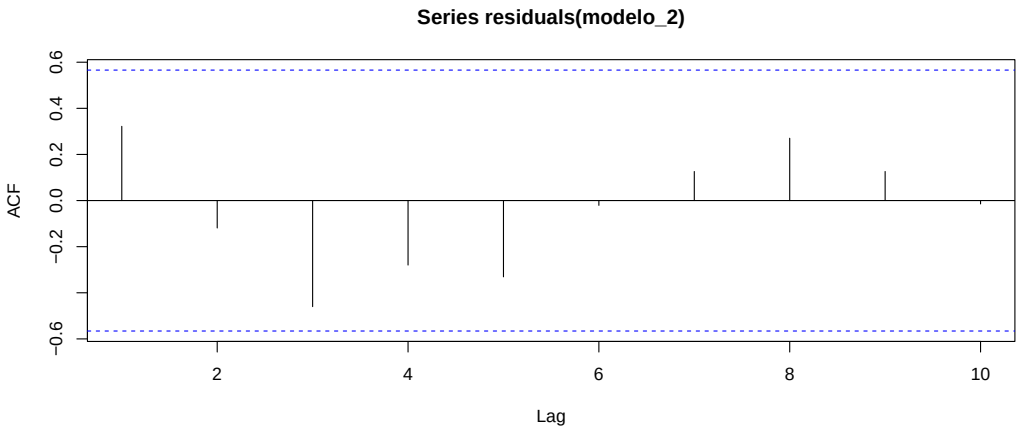
```
modelo_2 |> plot(which=1)
```



```r
modelo_2 |> car::ncvTest()
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.3497495, Df = 1, p = 0.55425
```

```
modelo_2 |> residuals() |> TSA::acf()
```



```
modelo_2 |> lmtest::dwtest()
```

```
    Durbin-Watson test

data:  modelo_2
DW = 1.1074, p-value = 0.003291
alternative hypothesis: true autocorrelation is greater than 0
```

```
modelo_2 |> car::vif()
```

```
       x   I(x^2)
19.10714 19.10714
```

```
modelo_2 |> AIC()
```

```
[1] 58.54982
```

Modelo 3: Se ejecutará el modelo de polinomios ortogonales segundo orden. Note la ortogonalidad de los polinomios

```
(P1 = 2*(x-mean(x))/25)
```

```
 [1] -11  -9  -7  -5  -3  -1   1   3   5   7   9  11
```

```
(P2 = 3*(((x-mean(x))/25)**2 - 143/12))
```

```
 [1]  55  25   1 -17 -29 -35 -35 -29 -17   1  25  55
```

R escala adicionalmente los polinomios de modo que $Pj^2 = 1$

```
P1 = P1/sqrt(sum(P1**2))
P2 = P2/sqrt(sum(P2**2))
(data.frame(P1, P2, y) -> datos)
```

```
           P1           P2   y
1  -0.4599331  0.501828160 350
2  -0.3763089  0.228103709 335
3  -0.2926847  0.009124148 326
4  -0.2090605 -0.155110522 316
5  -0.1254363 -0.264600302 313
6  -0.0418121 -0.319345193 311
7   0.0418121 -0.319345193 314
8   0.1254363 -0.264600302 318
9   0.2090605 -0.155110522 328
10  0.2926847  0.009124148 337
11  0.3763089  0.228103709 345
12  0.4599331  0.501828160 358
```

Estos mismos polinomios se obtienen con la función poly

```
poly(x,2)[1:nrow(datos),]
```

```
                1           2
 [1,] -0.4599331  0.501828160
 [2,] -0.3763089  0.228103709
 [3,] -0.2926847  0.009124148
 [4,] -0.2090605 -0.155110522
 [5,] -0.1254363 -0.264600302
 [6,] -0.0418121 -0.319345193
 [7,]  0.0418121 -0.319345193
 [8,]  0.1254363 -0.264600302
 [9,]  0.2090605 -0.155110522
[10,]  0.2926847  0.009124148
[11,]  0.3763089  0.228103709
[12,]  0.4599331  0.501828160
```

```
sum(poly(x,2)[,1]*poly(x,2)[,2]) # Note la ortogonalidad:
```

```
[1] -7.806256e-18
```

Ejecutando el modelo y analizando los resultados obtenidos

```
lm(y ~ P1 + P2, datos) -> modelo_3
modelo_3 |> summary()
```

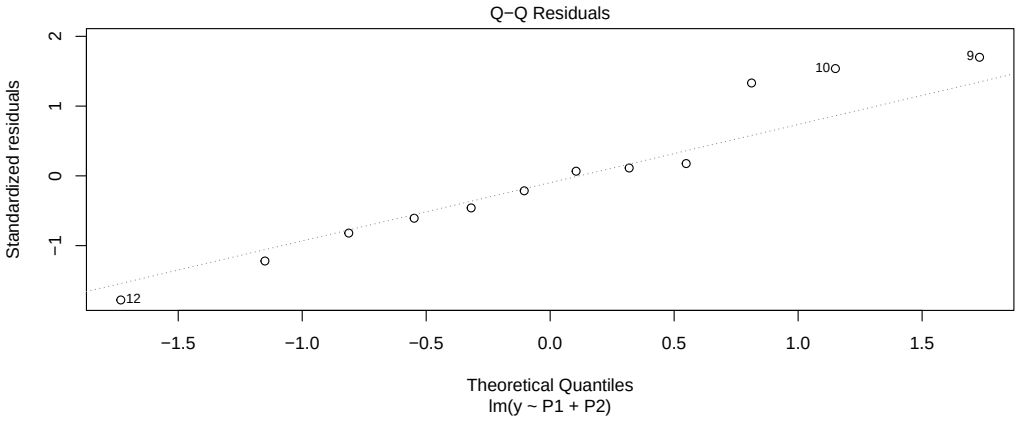
```
Call:
lm(formula = y ~ P1 + P2, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7500 -1.3776 -0.1399  0.7911  3.5962

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 329.2500     0.6628 496.769  < 2e-16 ***
P1           13.9234     2.2959   6.064 0.000187 ***
P2           50.0095     2.2959  21.782 4.27e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.296 on 9 degrees of freedom
Multiple R-squared:  0.9827,    Adjusted R-squared:  0.9789
F-statistic: 255.6 on 2 and 9 DF,  p-value: 1.178e-08
```

```
modelo_3 |> plot(which=2)
```

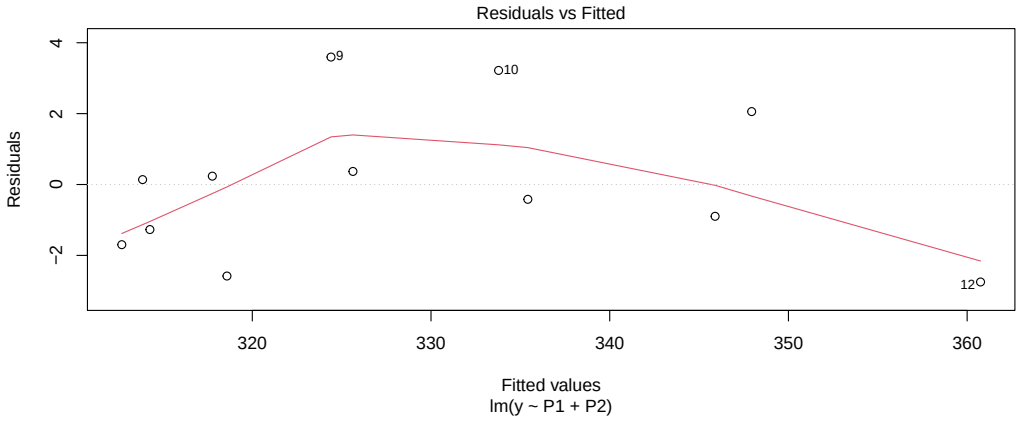


```r
modelo_3 |> residuals() |> shapiro.test()
```

```
    Shapiro-Wilk normality test

data:  residuals(modelo_3)
W = 0.93666, p-value = 0.456
```

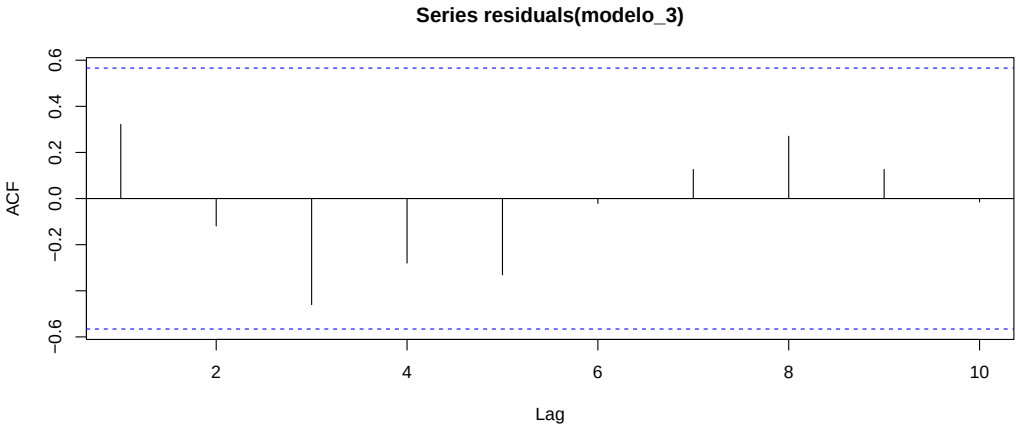
```
modelo_3 |> plot(which=1)
```



```r
modelo_3 |> car::ncvTest()
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.3497495, Df = 1, p = 0.55425
```

```
modelo_3 |> residuals() |> TSA::acf()
```



```
modelo_3 |> lmtest::dwtest()
```

```
    Durbin-Watson test

data:  modelo_3
DW = 1.1074, p-value = 0.003291
alternative hypothesis: true autocorrelation is greater than 0
```

```
modelo_3 |> car::vif()
```

```
P1 P2
 1  1
```

```
modelo_3 |> AIC()
```

```
[1] 58.54982
```

Modelo 4: Se realiza lo mismo para el modelo de polinomios ortogonales de tercer orden:

```
poly(x,3)[1:nrow(datos),]
```

```
                1           2           3
 [1,] -0.4599331  0.501828160 -0.45993311
 [2,] -0.3763089  0.228103709  0.04181210
 [3,] -0.2926847  0.009124148  0.29268470
 [4,] -0.2090605 -0.155110522  0.34843417
 [5,] -0.1254363 -0.264600302  0.26480997
 [6,] -0.0418121 -0.319345193  0.09756157
 [7,]  0.0418121 -0.319345193 -0.09756157
 [8,]  0.1254363 -0.264600302 -0.26480997
 [9,]  0.2090605 -0.155110522 -0.34843417
[10,]  0.2926847  0.009124148 -0.29268470
[11,]  0.3763089  0.228103709 -0.04181210
[12,]  0.4599331  0.501828160  0.45993311
```

```
sum(poly(x,3)[,1]*poly(x,3)[,2])
```

```
[1] -7.806256e-18
```

```
sum(poly(x,3)[,1]*poly(x,3)[,3])
```

```
[1] -2.428613e-17
```

```
sum(poly(x,3)[,2]*poly(x,3)[,3])
```

```
[1] -1.374768e-16
```

```r
lm(y ~ poly(x,3), datos) -> modelo_4
modelo_4 |> summary()
```

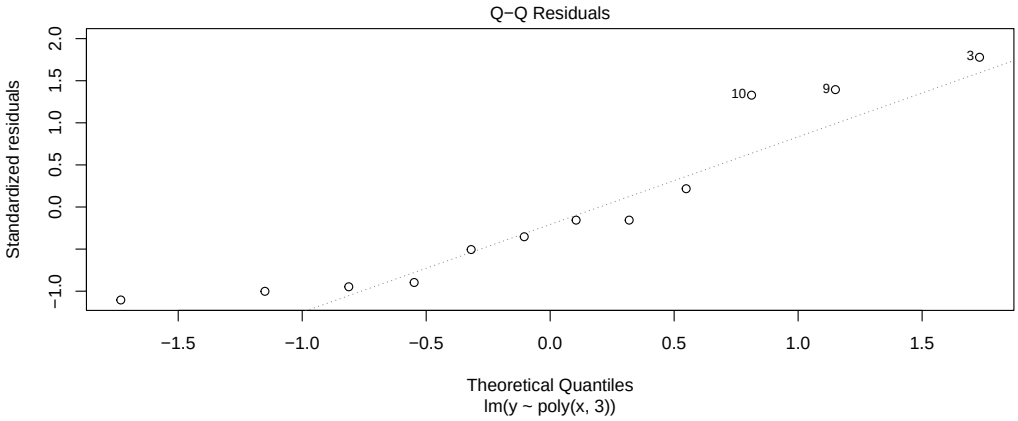
```
Call:
lm(formula = y ~ poly(x, 3), data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2883 -0.7263 -0.3003  0.5734  2.0536 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 329.2500     0.3861 852.811  < 2e-16 ***
poly(x, 3)1  13.9234     1.3374  10.411 6.28e-06 ***
poly(x, 3)2  50.0095     1.3374  37.393 2.87e-10 ***
poly(x, 3)3  -5.7561     1.3374  -4.304   0.0026 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.337 on 8 degrees of freedom
Multiple R-squared:  0.9948,    Adjusted R-squared:  0.9928 
F-statistic: 508.4 on 3 and 8 DF,  p-value: 1.821e-09
```

```
modelo_4 |> plot(which=2)
```

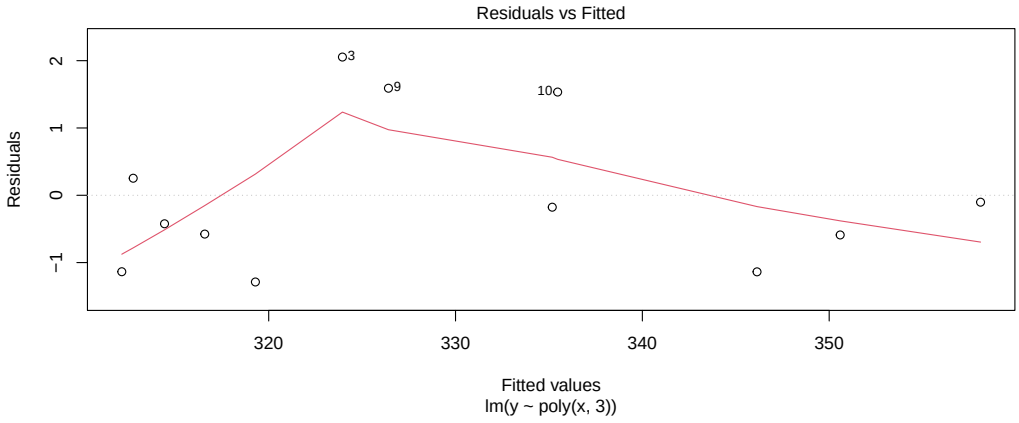


```r
modelo_4 |> residuals() |> shapiro.test()
```

```
    Shapiro-Wilk normality test

data:  residuals(modelo_4)
W = 0.8798, p-value = 0.08712
```

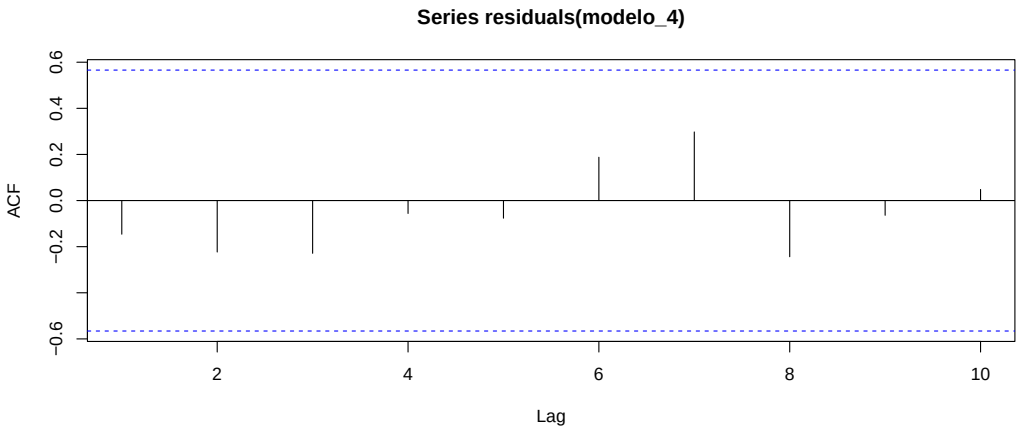
```
modelo_4 |> plot(which=1)
```



```
modelo_4 |> car::ncvTest()
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.1296415, Df = 1, p = 0.7188
```

```
modelo_4 |> residuals() |> TSA::acf()
```



```
modelo_4 |> lmtest::dwtest()
```

```
    Durbin-Watson test

data:  modelo_4
DW = 2.2668, p-value = 0.2355
alternative hypothesis: true autocorrelation is greater than 0
```

```
#modelo_4 |> car::vif()
modelo_4 |> AIC()
```

```
[1] 46.16652
```

Modelo 5: ¿Y si intentamos un modelo de polinomios ortogonales de cuarto orden?

```
poly(x,4)[1:nrow(datos),]
```

```
                1           2           3          4
 [1,] -0.4599331  0.501828160 -0.45993311  0.3687669
 [2,] -0.3763089  0.228103709  0.04181210 -0.3017184
 [3,] -0.2926847  0.009124148  0.29268470 -0.3687669
 [4,] -0.2090605 -0.155110522  0.34843417 -0.1452718
 [5,] -0.1254363 -0.264600302  0.26480997  0.1340970
 [6,] -0.0418121 -0.319345193  0.09756157  0.3128931
 [7,]  0.0418121 -0.319345193 -0.09756157  0.3128931
 [8,]  0.1254363 -0.264600302 -0.26480997  0.1340970
 [9,]  0.2090605 -0.155110522 -0.34843417 -0.1452718
[10,]  0.2926847  0.009124148 -0.29268470 -0.3687669
[11,]  0.3763089  0.228103709 -0.04181210 -0.3017184
[12,]  0.4599331  0.501828160  0.45993311  0.3687669
```

```r
lm(y ~ poly(x,4), datos) -> modelo_5
modelo_5 |> summary()
```

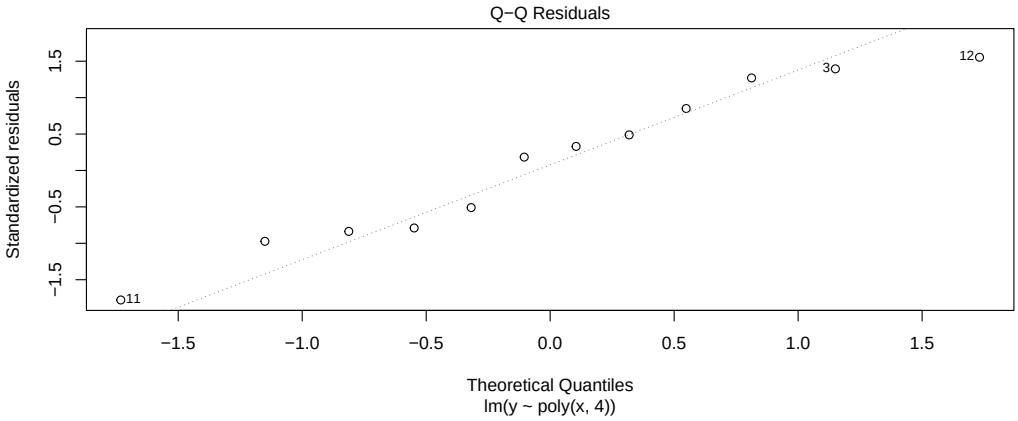
```
Call:
lm(formula = y ~ poly(x, 4), data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7276 -0.7905  0.1599  0.6671  1.3325

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 329.2500     0.3533 931.928  < 2e-16 ***
poly(x, 4)1  13.9234     1.2239  11.377 9.09e-06 ***
poly(x, 4)2  50.0095     1.2239  40.862 1.37e-09 ***
poly(x, 4)3  -5.7561     1.2239  -4.703   0.0022 **
poly(x, 4)4  -1.9556     1.2239  -1.598   0.1541
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.224 on 7 degrees of freedom
Multiple R-squared:  0.9962,    Adjusted R-squared:  0.994
F-statistic: 455.9 on 4 and 7 DF,  p-value: 1.551e-08
```

```r
modelo_5 |> plot(which=2)
```

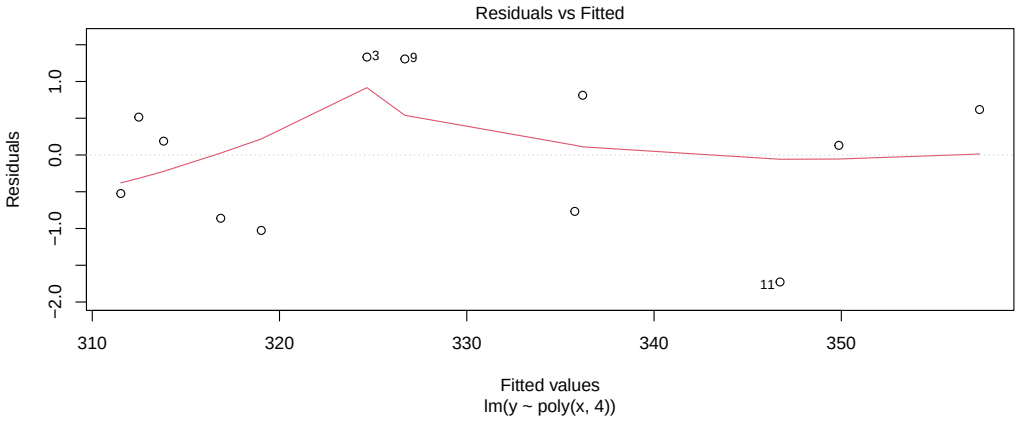


```r
modelo_5 |> residuals() |> shapiro.test()
```

```
    Shapiro-Wilk normality test

data:  residuals(modelo_5)
W = 0.95315, p-value = 0.6834
```

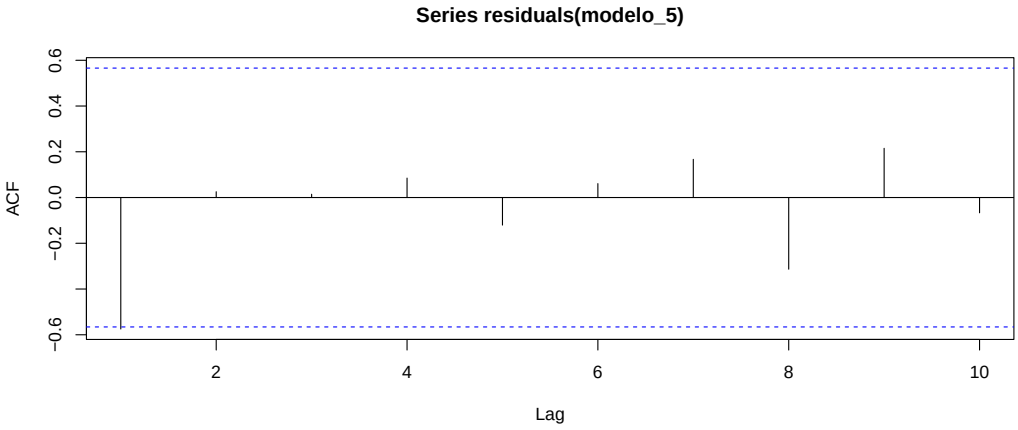
```
modelo_5 |> plot(which=1)
```



```r
modelo_5 |> car::ncvTest()
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.2070563, Df = 1, p = 0.64908
```

```
modelo_5 |> residuals() |> TSA::acf()
```



```
modelo_5 |> lmtest::dwtest()
```

```
    Durbin-Watson test

data:  modelo_5
DW = 3.1115, p-value = 0.7474
alternative hypothesis: true autocorrelation is greater than 0
```

```
modelo_5 |> AIC()
```

```
[1] 44.43493
```

El modelo de polinomios ortogonales de segundo orden presenta buenos indicadores, a excepción del supuesto de independencia. Si solo se desea utilizar el modelo con fines predictivos, podría ser una opción (revisar en testing).

```
lm(y ~ poly(x,2), datos) -> modelo_3
c(60,70,84.1,171.9,206.6,280) -> xpred
modelo_3 |> predict(data.frame('x'=xpred)) -> ypred
data.frame('x'= c(60,70,84.1,171.9,206.6,280),ypred)
```

```
      x    ypred
1  60.0 331.1746
2  70.0 327.3694
3  84.1 322.7484
4 171.9 313.5688
5 206.6 319.2509
6 280.0 348.6485
```

El modelo que cumple con todos los supuestos es el de polinomios ortogonales de tercer orden:

```
lm(y ~ poly(x,3), datos) -> modelo_4
c(60,70,84.1,171.9,206.6,280) -> xpred
modelo_4 |> predict(data.frame('x'=xpred)) -> ypred
data.frame('x'= c(60,70,84.1,171.9,206.6,280),ypred)
```

```
      x    ypred
1  60.0 330.2010
2  70.0 325.8734
3  84.1 320.8337
4 171.9 313.9933
5 206.6 320.9622
6 280.0 348.4373
```