Actividad2_3

Ricardo Kaleb Flores Alfonso

2024-09-27

```
df <- read.csv("datosRes.csv")
y <- df$Resistencia
x1<-df$Longitud
x2<-df$Altura.matriz
x3<-df$Altura.poste
x4<-df$Altura.amarre
```

1) Seleccion de variables

```
modelo <- lm(y ~ x1+x2+x3+x4)
```

AIC

```
step(modelo,direction="both",trace=1)
```

```
## Start:  AIC=32.54
## y ~ x1 + x2 + x3 + x4
##
##        Df Sum of Sq     RSS     AIC
## - x3    1      1.81   63.41  31.269
## <none>                61.60  32.543
## - x4    1     28.92   90.52  40.167
## - x2    1     40.50  102.10  43.176
## - x1    1   1568.75 1630.34 112.442
##
## Step:  AIC=31.27
## y ~ x1 + x2 + x4
##
##        Df Sum of Sq     RSS     AIC
## <none>                63.41  31.269
## + x3    1      1.81   61.60  32.543
## - x2    1     40.21  103.62  41.546
## - x4    1     51.76  115.17  44.189
## - x1    1   2552.49 2615.90 122.262
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4)
##
## Coefficients:
## (Intercept)           x1           x2           x4
##    1.367068     2.534919     0.008522     2.599278
```

BIC

```
n <- nrow(df)
modelo <- lm(y ~ x1+x2+x3+x4)
step(modelo,direction="both",trace=1, k=log(n))
```

```
## Start:  AIC=38.64
## y ~ x1 + x2 + x3 + x4
##
##        Df Sum of Sq     RSS     AIC
## - x3    1      1.81   63.41  36.144
## <none>                61.60  38.638
## - x4    1     28.92   90.52  45.043
## - x2    1     40.50  102.10  48.052
## - x1    1   1568.75 1630.34 117.317
##
## Step:  AIC=36.14
## y ~ x1 + x2 + x4
##
##        Df Sum of Sq     RSS     AIC
## <none>                63.41  36.144
## + x3    1      1.81   61.60  38.638
## - x2    1     40.21  103.62  45.203
## - x4    1     51.76  115.17  47.846
## - x1    1   2552.49 2615.90 125.919
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4)
##
## Coefficients:
## (Intercept)           x1           x2           x4
##    1.367068     2.534919     0.008522     2.599278
```

```
modelo <- lm(y ~ x1+x2+x4)
```

```
summary(modelo)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2828 -1.1230  0.1207  1.0497  3.1978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.367068   0.833704   1.640 0.115952
## x1          2.534919   0.087187  29.074  < 2e-16 ***
## x2          0.008522   0.002335   3.649 0.001498 **
## x4          2.599278   0.627791   4.140 0.000465 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.738 on 21 degrees of freedom
```

```
## Multiple R-squared:  0.9896, Adjusted R-squared:  0.9881
## F-statistic:   667 on 3 and 21 DF,  p-value: < 2.2e-16
```

2) Datos atípicos

Metodo de desviación estandar

```
residuals_values<- rstandard(modelo)
residuals_values
```

```
##           1           2           3           4           5           6
##  0.30995789 -0.44841839 -0.70704561 -0.97259115 -1.03667630  0.62805855
##           7           8           9          10          11          12
##  1.31710547  0.07663099 -2.03067946  0.41597748 -1.96463341 -0.50828735
##          13          14          15          16          17          18
## -1.21298733 -0.27131869  2.06875195  0.22750728  1.07912706 -0.82003000
##          19          20          21          22          23          24
##  0.66091424 -0.08090377  0.28009197  0.03204011  1.27689138  1.28159757
##          25
##  0.66393145
```

Metodo de estandarización

```
rstudents_values<-rstudent(modelo)
rstudents_values
```

```
##           1           2           3           4           5           6
##  0.30318224 -0.43972181 -0.69836840 -0.97127920 -1.03861768  0.61876125
##           7           8           9          10          11          12
##  1.34198733  0.07479465 -2.21063596  0.40763536 -2.12221013 -0.49911740
##          13          14          15          16          17          18
## -1.22753822 -0.26524523  2.26256901  0.22229849  1.08359296 -0.81339633
##          19          20          21          22          23          24
##  0.65180073 -0.07896631  0.27385378  0.03126871  1.29750549  1.30269262
##          25
##  0.65483998
```

3) Datos influyentes

Por grado de leverage

```
hat_values<-hatvalues(modelo)
hat_values
```

```
##          1          2          3          4          5          6          7
## 0.17997067 0.11494030 0.16449688 0.10188969 0.06751594 0.07494729 0.11898344
##          8          9         10         11         12         13         14
## 0.17842872 0.13452586 0.06794985 0.15609899 0.05560348 0.19198635 0.11340056
##         15         16         17         18         19         20         21
## 0.20870226 0.09845327 0.42289094 0.30685227 0.22222579 0.14775554 0.26018831
##         22         23         24         25
## 0.17974960 0.23312954 0.10912925 0.09018521
```

Por distance de Cook

```
cooks_values<-cooks.distance(modelo)
cooks_values
```

```
##            1            2            3            4            5            6
## 5.271300e-03 6.528398e-03 2.460620e-02 2.682880e-02 1.945321e-02 7.989686e-03
##            7            8            9           10           11           12
## 5.857113e-02 3.188368e-04 1.602413e-01 3.153762e-03 1.784891e-01 3.802824e-03
##           13           14           15           16           17           18
## 8.739854e-02 2.353896e-03 2.821916e-01 1.413099e-03 2.133318e-01 7.442229e-02
##           19           20           21           22           23           24
## 3.120119e-02 2.836986e-04 6.897757e-03 5.624053e-05 1.239148e-01 5.030021e-02
##           25
## 1.092368e-02
```

4) Resumen de resultados

```
tabla= data.frame(residuals_values,rstudents_values, hat_values, cooks_values)
tabla
```

```
##    residuals_values rstudents_values hat_values cooks_values
## 1        0.30995789       0.30318224 0.17997067 5.271300e-03
## 2       -0.44841839      -0.43972181 0.11494030 6.528398e-03
## 3       -0.70704561      -0.69836840 0.16449688 2.460620e-02
## 4       -0.97259115      -0.97127920 0.10188969 2.682880e-02
## 5       -1.03667630      -1.03861768 0.06751594 1.945321e-02
## 6        0.62805855       0.61876125 0.07494729 7.989686e-03
## 7        1.31710547       1.34198733 0.11898344 5.857113e-02
## 8        0.07663099       0.07479465 0.17842872 3.188368e-04
## 9       -2.03067946      -2.21063596 0.13452586 1.602413e-01
## 10       0.41597748       0.40763536 0.06794985 3.153762e-03
## 11      -1.96463341      -2.12221013 0.15609899 1.784891e-01
## 12      -0.50828735      -0.49911740 0.05560348 3.802824e-03
## 13      -1.21298733      -1.22753822 0.19198635 8.739854e-02
## 14      -0.27131869      -0.26524523 0.11340056 2.353896e-03
## 15       2.06875195       2.26256901 0.20870226 2.821916e-01
## 16       0.22750728       0.22229849 0.09845327 1.413099e-03
## 17       1.07912706       1.08359296 0.42289094 2.133318e-01
## 18      -0.82003000      -0.81339633 0.30685227 7.442229e-02
## 19       0.66091424       0.65180073 0.22222579 3.120119e-02
## 20      -0.08090377      -0.07896631 0.14775554 2.836986e-04
## 21       0.28009197       0.27385378 0.26018831 6.897757e-03
## 22       0.03204011       0.03126871 0.17974960 5.624053e-05
## 23       1.27689138       1.29750549 0.23312954 1.239148e-01
## 24       1.28159757       1.30269262 0.10912925 5.030021e-02
## 25       0.66393145       0.65483998 0.09018521 1.092368e-02
```

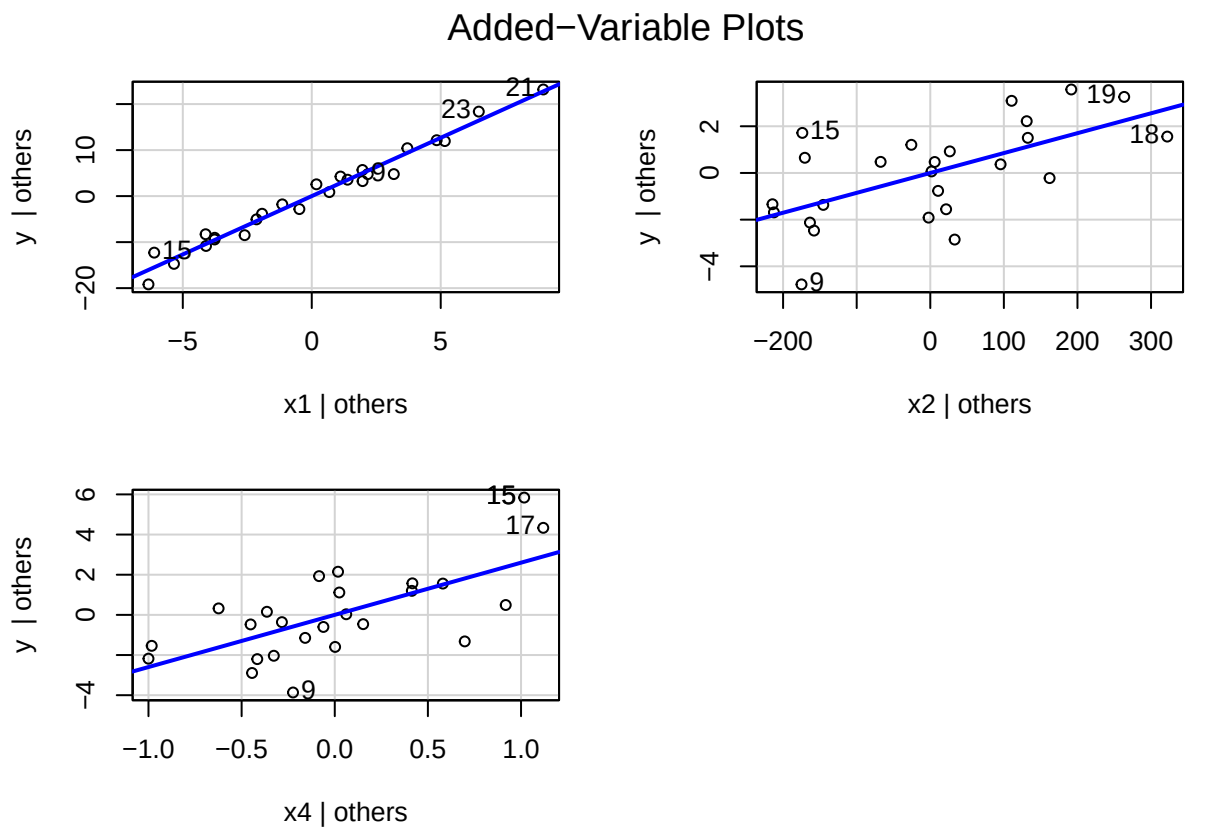
```
library(car)
```

```
## Cargando paquete requerido: carData
```

5) Gráficos complementarios

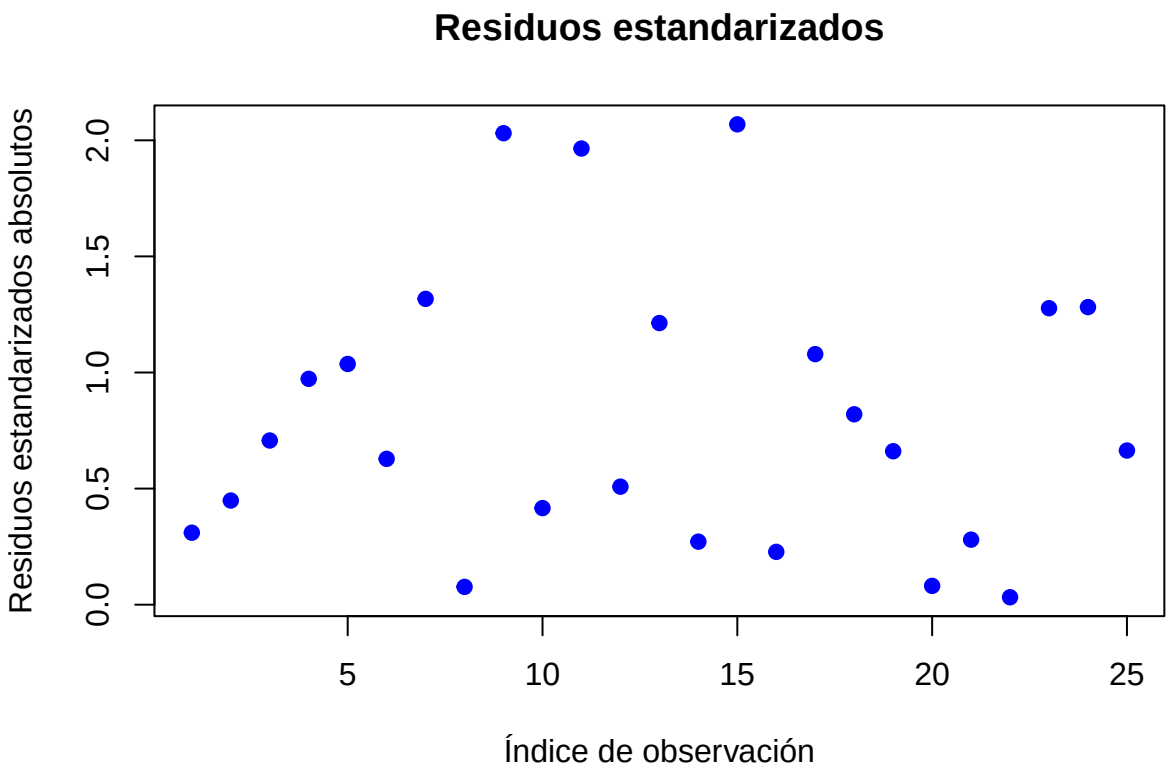
Variable dependiente contra las variables predictoras

```r
avPlots(modelo)
```



Residuos estandarizados absolutos e identifica aquellos cuyo valor absoluto es mayor a 3.

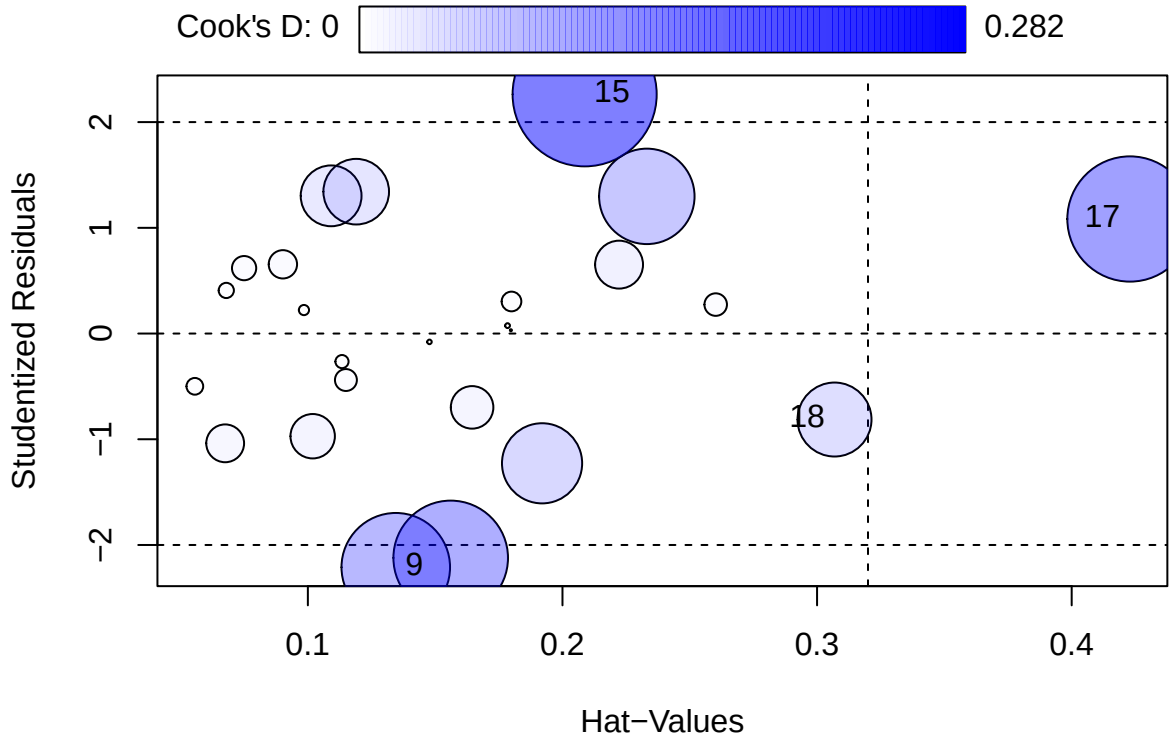
```r
plot(abs(residuals_values), ylab = "Residuos estandarizados absolutos",
     xlab = "Índice de observación", pch = 19, col = "blue", main = "Residuos estandarizados")

# Identificar aquellos cuyo valor absoluto es mayor a 3
abline(h = 3, col = "red", lty = 2)
abline(h = -3, col = "red", lty = 2)
```




```
# Destacar los puntos que superan el valor absoluto de 3
 #No hay datos mayores a 3
```

Gráfico de influencia

```
influencePlot(modelo, id=TRUE)
```




```
##       StudRes       Hat      CookD
## 9  -2.2106360 0.1345259 0.16024130
## 15  2.2625690 0.2087023 0.28219160
## 17  1.0835930 0.4228909 0.21333184
## 18 -0.8133963 0.3068523 0.07442229
```

6) Ajustes del modelo

En caso de que haber datos influyentes, realice el análisis de regresión sin éstos y reporte la comparación en ambos modelos.

```
# Crear un nuevo dataframe sin los puntos influyentes
df_influyente <- df[-c(9,15,17,18), ]

# Ajustar un nuevo modelo sin los puntos influyentes
modelo_ajustado <- lm(y ~ x1 + x2 + x4, data = df_influyente)

# Comparar los dos modelos
summary(modelo_ajustado)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4, data = df_influyente)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.2828 -1.1230  0.1207  1.0497  3.1978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.367068   0.833704   1.640 0.115952
## x1          2.534919   0.087187  29.074  < 2e-16 ***
## x2          0.008522   0.002335   3.649 0.001498 **
## x4          2.599278   0.627791   4.140 0.000465 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.738 on 21 degrees of freedom
## Multiple R-squared:  0.9896, Adjusted R-squared:  0.9881
## F-statistic:   667 on 3 and 21 DF,  p-value: < 2.2e-16
```

BIC

```
n <- nrow(df)
modelo <- lm(y ~ x1+x2+x3+x4)
step(modelo_ajustado,direction="both",trace=1, k=log(n))
```

```
## Start:  AIC=36.14
## y ~ x1 + x2 + x4
##
##        Df Sum of Sq     RSS     AIC
## <none>                63.41  36.144
## - x2    1     40.21  103.62  45.203
## - x4    1     51.76  115.17  47.846
## - x1    1   2552.49 2615.90 125.919
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4, data = df_influyente)
##
## Coefficients:
## (Intercept)           x1           x2           x4
##    1.367068     2.534919     0.008522     2.599278
```

Variabilidad explicada por el modelo

El modelo inicial reportó la siguiente variabilidad y valor de p:

Multiple R-squared: 0.9896, Adjusted R-squared: 0.9881 F-statistic: 667 on 3 and 21 DF, p-value: $<$ 2.2e-16

El modelo nuevo reportó esta variabilidad y valor de p: Multiple R-squared: 0.9896, Adjusted R-squared: 0.9881 F-statistic: 667 on 3 and 21 DF, p-value: $<$ 2.2e-16

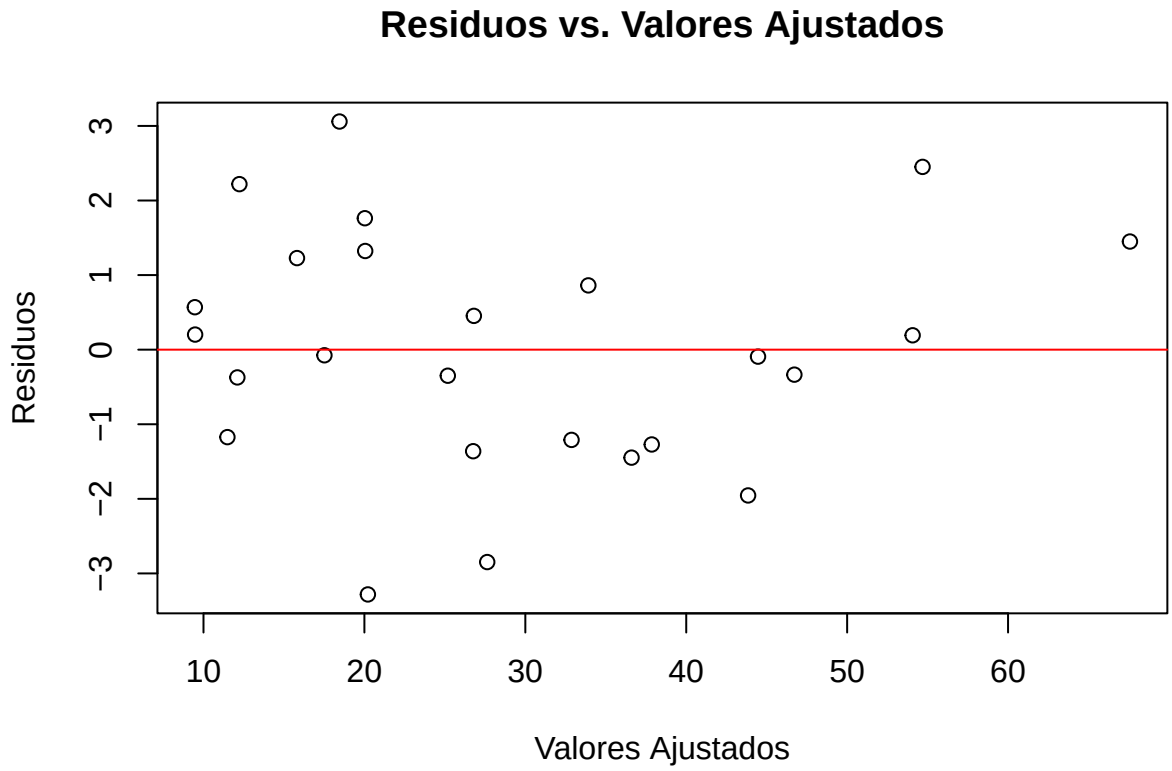
Es por esto que podemos ver que la R ajustada con valor 0.9881, y su F Statistic es de 667, es mejor que el modelo original. Pero el haber eliminado datos, no hizo que el modelo mejorará.

Supuestos de los modelos

Linealidad)

```
# Gráfico de residuos vs. valores ajustados
plot(modelo_ajustado$fitted.values, modelo$residuals,
     xlab = "Valores Ajustados", ylab = "Residuos",
```

```
     main = "Residuos vs. Valores Ajustados")
abline(h = 0, col = "red")
```



Se observa como los residuos tiene promedio alrededor de 0 y no muestran algun tipo de patron

Independencia de errores

```
library(lmtest)
```

```
## Cargando paquete requerido: zoo
```

```
##
## Adjuntando el paquete: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# Prueba de Durbin-Watson para autocorrelación de los errores
dwtest(modelo_ajustado)
```

```
##
##  Durbin-Watson test
##
## data:  modelo_ajustado
## DW = 1.6688, p-value = 0.1814
## alternative hypothesis: true autocorrelation is greater than 0
```

El test de durbin watson presente un valor de 0.1814, lo cual es mayor a 0.05, significa que no hay evidencia

para rechazar que no existe autocorrelación entre los errores.

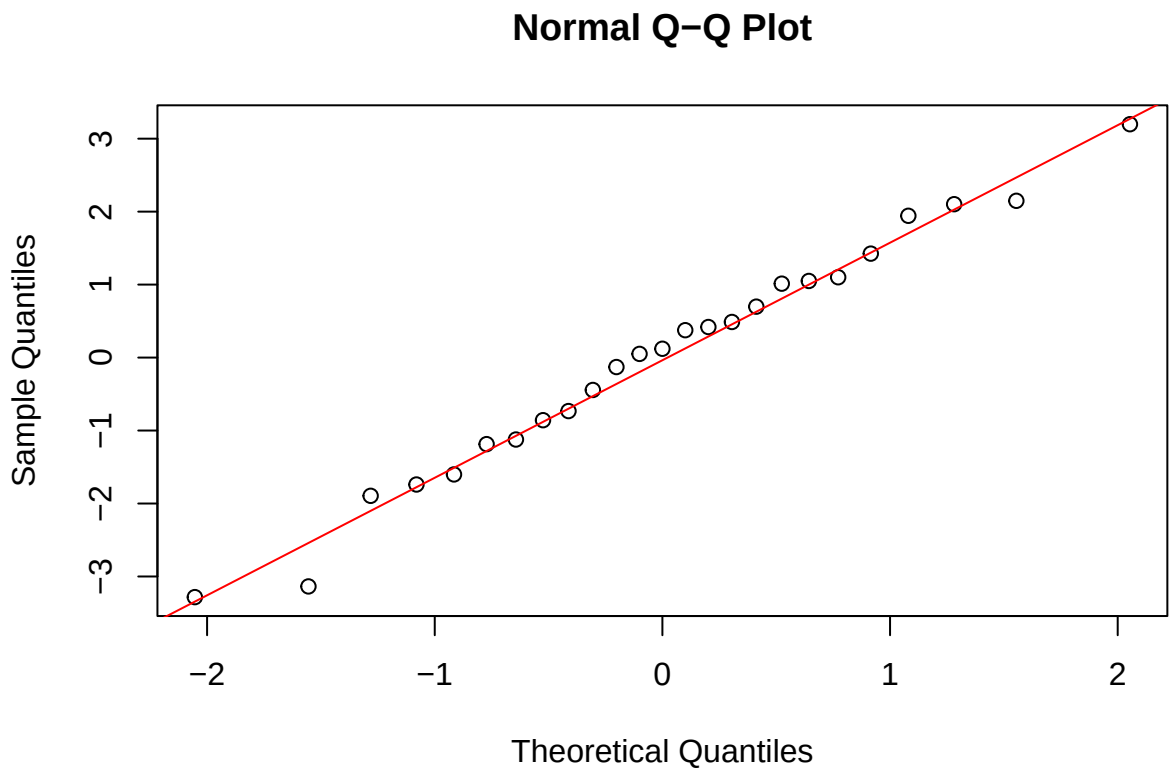
Homocedasticidad

```
library(lmtest)

bptest(modelo_ajustado)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo_ajustado
## BP = 2.8808, df = 3, p-value = 0.4104
```

Con el valor de breusch pagan con valor de $p = 0.4104 > 0.05$, no hay evidencia para rechazar la hipotesis nula, que dice que los errores tienen varianza constante. ### Normalidad de residuos

```
# Gráfico Q-Q plot
qqnorm(modelo_ajustado$residuals)
qqline(modelo_ajustado$residuals, col = "red")
```




```
# Prueba de Shapiro-Wilk para normalidad de los residuos
shapiro.test(modelo_ajustado$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelo_ajustado$residuals
```

```
## W = 0.98394, p-value = 0.9504
```

Con el valor de $p = 0.9504 > 0.05$ no hay evidencia para rechazar la hipotesis nula, de manera que se acepta que existe normalidad en los residuos.

AIC y BIC

AIC : Modelo anterior: 32.54 Modelo nuevo: 31.27, no presentó cambios al eliminar datos BIC: Modelo Anterior 38.64 Modelo Nuevo: 36.14, no presentó cambios al eliminar datos

Conclusión

En este caso el modelo obtenido explica al 98.81% la variabilidad de y, asi como pasa todos los supuestos para ser valido. De igual manera demuestra tener significancia y esta optimizado respecto a las variables que incluye. Se eliminaron los outliers presentes, sin embargo esto no mejoró el modelo, lo cual significa que no estaban teniendo tanta influencia respecto al modelo. Sin embargo en casos donde hayan más valores atipicos, puede influir al modelado de los datos.

Para este caso el modelo obtenido es muy bueno y puede ser usado para predecir datos futuros.