

# **Lista de ejercicios 2 - Análisis de regresión**

**Ciclo nivelación 2025-2**

Mg. Sc. J. Eduardo Gamboa U.

En un curso universitario de Estadística General, se busca analizar qué factores influyen en el tiempo que un estudiante tarda en resolver un examen.

Antes del examen, los estudiantes rindieron una prueba diagnóstica para medir conocimientos previos. Además, se registraron las horas de estudio realizadas la semana anterior al examen, y se aplicó un breve cuestionario de ansiedad ante evaluaciones.

Durante el examen final se registró el tiempo de resolución en minutos. Los datos se encuentran en el archivo **Lista2\_datos.csv**

Las variables son:

- time\_minutes: tiempo de resolución del examen final (minutos)
- diagnostic\_score: puntaje en la prueba diagnóstica
- prior\_study\_hours: tiempo de estudio (horas)
- anxiety\_level: nivel de ansiedad (escala ordinal de 1 a 10)

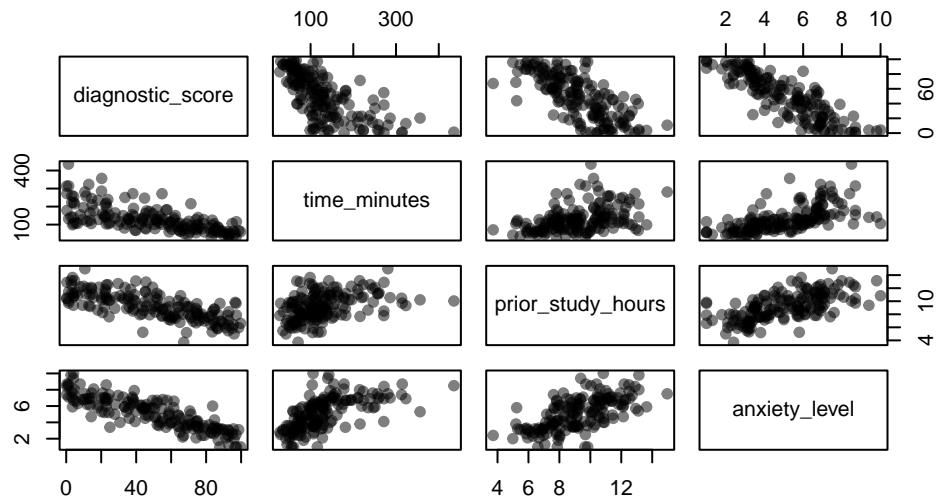
1. Presentar y comentar un gráfico que permite ver las asociaciones entre las variables en estudio.

```
datos = read.csv('Lista2_datos.csv')
datos |> head()
```

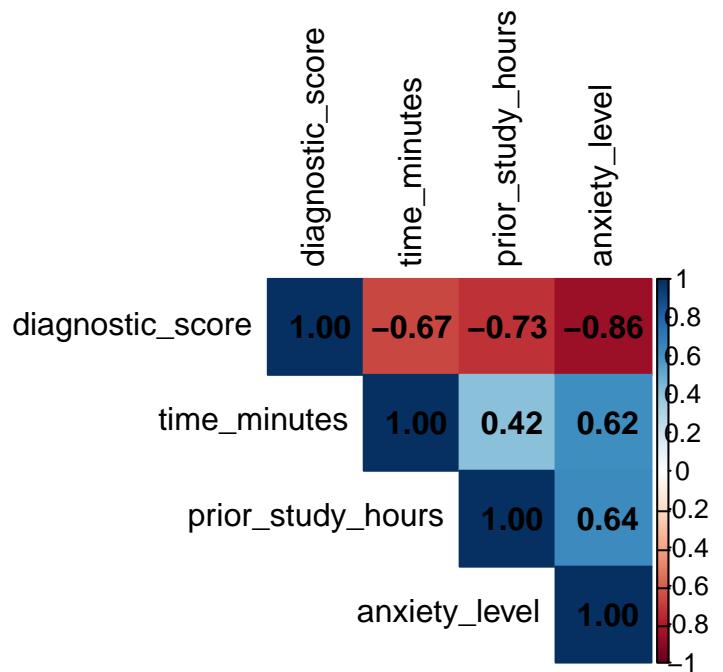
	diagnostic_score	time_minutes	prior_study_hours	anxiety_level
1	65.0	108.44	6.76	2.8
2	29.7	97.58	12.70	7.3
3	33.3	80.79	10.36	5.9
4	69.6	43.97	9.06	3.6
5	14.5	245.14	11.93	7.6
6	28.0	165.07	10.62	6.2

```
pairs(datos, main = "Matriz de dispersión entre variables", pch = 19,  
       col = rgb(0, 0, 0.5))
```

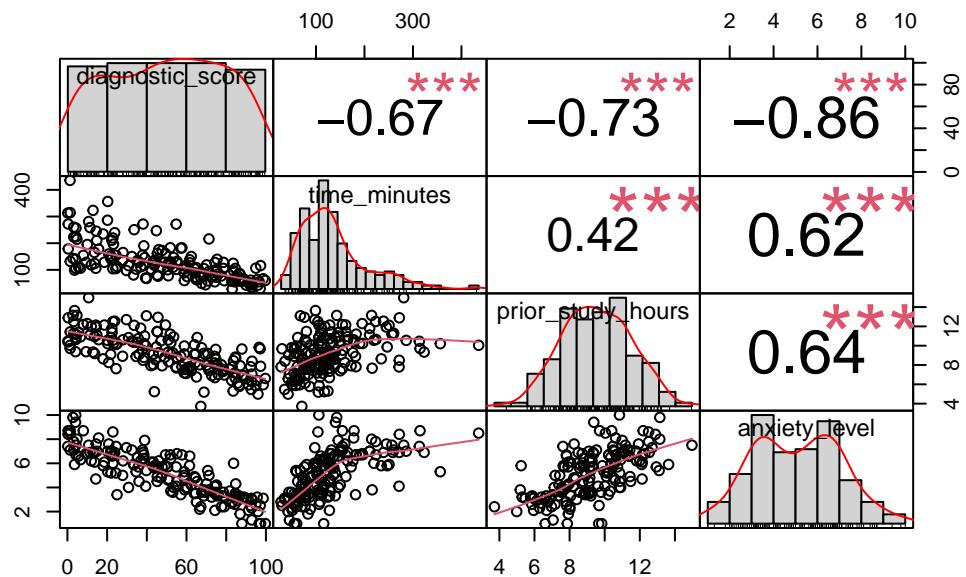
## Matriz de dispersión entre variables



```
library(corrplot)  
corrplot(cor(datos), method = "color",  
        type = "upper", addCoef.col = "black", tl.col = "black")
```



```
library(PerformanceAnalytics)
chart.Correlation(datos)
```



La asociación entre el tiempo de resolución del examen y el puntaje de la prueba diagnóstica es directa o positiva, pero no está tan marcada. Por otro lado, el tiempo de resolución del examen se asocia de manera negativa o inversa y fuerte con el puntaje de la prueba diagnóstica. Finalmente, la asociación entre el nivel de ansiedad y el tiempo de resolución de la prueba, es positiva pero parece no ser lineal.

2. Formular el modelo de regresión lineal.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 x_3 + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

donde:

- $Y$ : Tiempo de resolución del examen (variable respuesta)
- $X_1$ : puntaje en la prueba diagnóstica (variable independiente)
- $X_2$ : Horas de estudio (variable independiente)
- $X_3$ : Nivel de ansiedad (variable independiente)
- $\beta_0$ : Intercepto del modelo de regresión lineal
- $\beta_1, \beta_2, \beta_3$ : coeficientes del modelo de regresión lineal
- $\epsilon$ : error aleatorio del modelo de regresión lineal

3. Presentar el modelo de regresión lineal estimado.

```
modelo1 <- lm(time_minutes ~ diagnostic_score + prior_study_hours + anxiety_level,
                data = datos)

coef(modelo1)
```

	(Intercept)	diagnostic_score	prior_study_hours	anxiety_level
	220.661967	-1.521025	-5.196894	6.478634

$$\hat{Y} = 220.66 - 1.52X_1 - 5.2X_2 + 6.48X_3$$

4. Analizar el cumplimiento del supuesto de normalidad de errores.

```
modelo1 |> residuals() -> res1

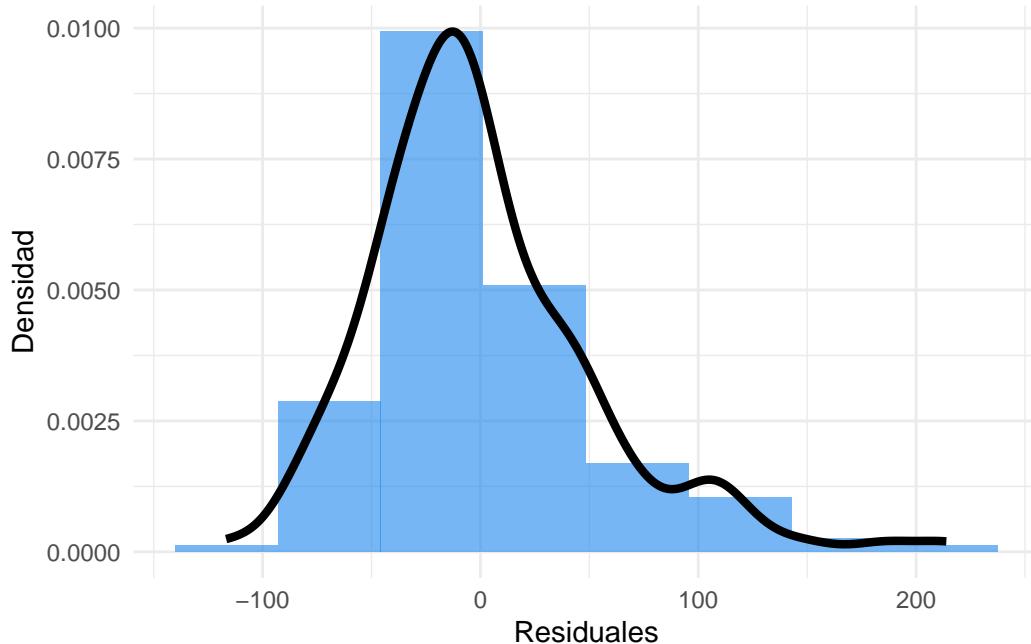
library(ggplot2)
```

Warning: package 'ggplot2' was built under R version 4.4.3

```
data.frame(res1) |>
  ggplot(aes(x=res1))+
  geom_histogram(aes(y =..density..),
                 bins = round(1+3.3*log10(nrow(datos))),
                 fill = "dodgerblue2",
                 alpha = 0.6)+
  geom_density(size = 1.5)+
  labs(x="Residuales",y="Densidad")+
  theme_minimal()
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
i Please use `linewidth` instead.

Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.  
i Please use `after\_stat(density)` instead.

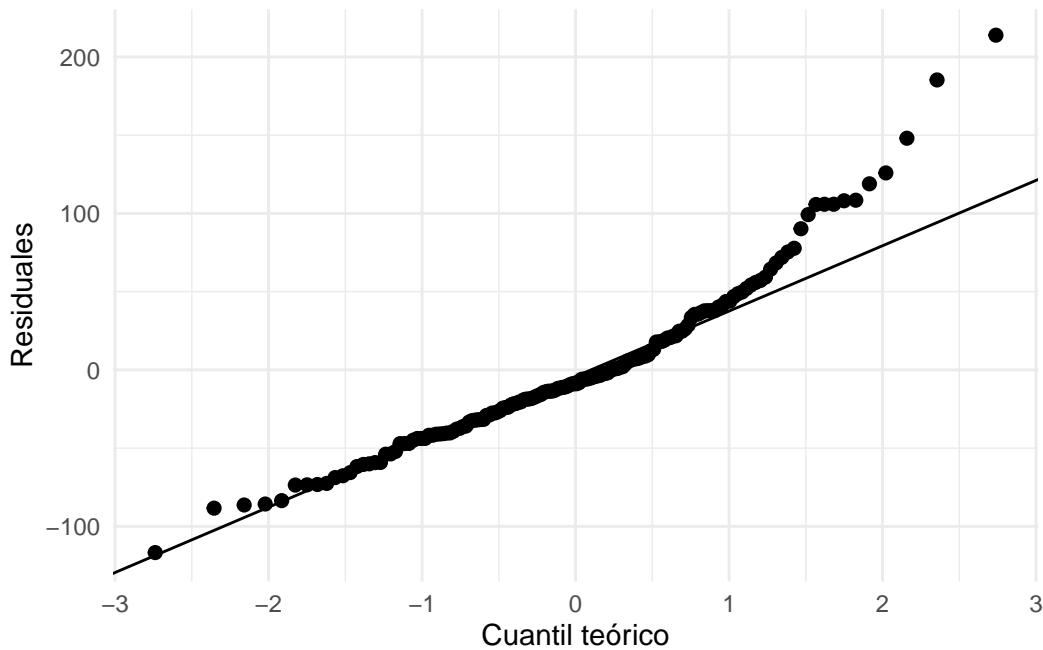


Se aprecia asimetría en la distribución de los residuales, y una posible leptocurtosis, debido al apuntalamiento en el centro.

```

data.frame(res1) |>
  ggplot(aes(sample=res1))+
  stat_qq(size = 2) +
  stat_qq_line(distribution = stats::qnorm)+
  labs(x = "Cuantil teórico", y = "Residuales")+
  theme_minimal()

```



Los puntos se alejan de la recta hacia el lado superior o derecho. Eso es señal de asimetría, lo que complicaría el cumplimiento del supuesto de normalidad de errores.

$$H_0 : As = 0 \quad H_1 : As \neq 0 \quad \alpha = 0.05$$

```

library(moments)
res1 |> agostino.test()

```

D'Agostino skewness test

```

data: res1
skew = 1.0859, z = 4.9322, p-value = 8.129e-07
alternative hypothesis: data have a skewness

```

Se rechaza  $H_0$ , entonces existe evidencia de que el coeficiente de asimetría es distinto de cero (existe asimetría).

$$H_0 : K = 3 \quad H_1 : K \neq 3 \quad \alpha = 0.05$$

```
res1 |> anscombe.test()
```

Anscombe-Glynn kurtosis test

```
data: res1
kurt = 5.0661, z = 3.3685, p-value = 0.0007559
alternative hypothesis: kurtosis is not equal to 3
```

Se rechaza  $H_0$ , entonces existe evidencia de que el coeficiente de kurtosis es distinto de tres (los errores no son mesocúrticos).

Por lo tanto, al no ser simétricos ni mesocúrticos, seguimos acumulando evidencias que nos alejan de la normalidad.

$H_0$ : los errores siguen una distribución normal

$H_1$ : los errores no siguen una distribución normal

$$\alpha = 0.05$$

```
res1 |> shapiro.test()
```

Shapiro-Wilk normality test

```
data: res1
W = 0.93908, p-value = 2.029e-06
```

```
library(nortest)
res1 |> ad.test()
```

Anderson-Darling normality test

```
data: res1
A = 2.5582, p-value = 1.748e-06
```

```
res1 |> lillie.test()
```

Lilliefors (Kolmogorov-Smirnov) normality test

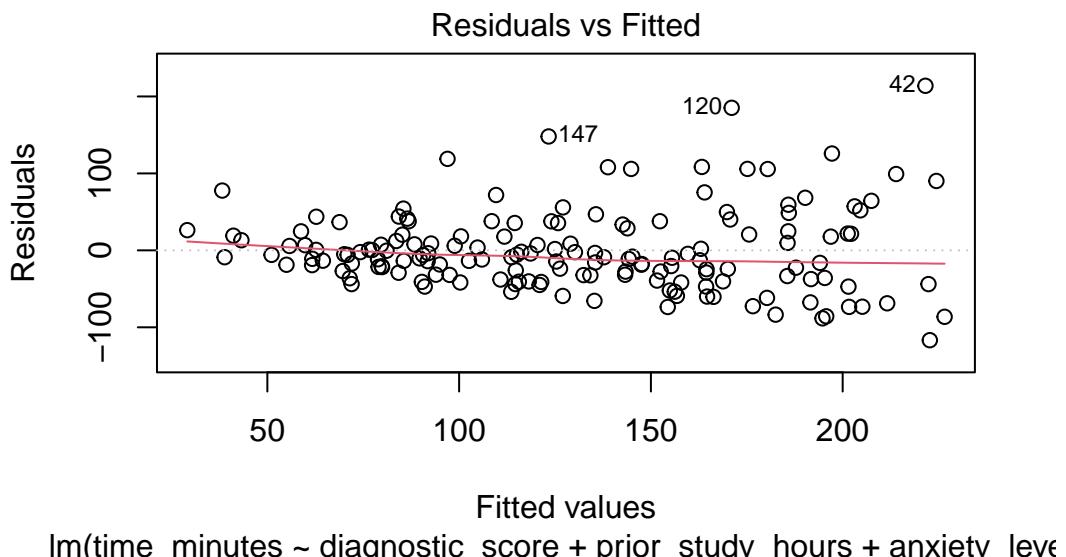
```
data: res1  
D = 0.11159, p-value = 4.048e-05
```

Las pruebas de normalidad dan evidencia de que los errores no siguen una distribución normal.

**En conclusión, no se cumple el supuesto de normalidad de errores.**

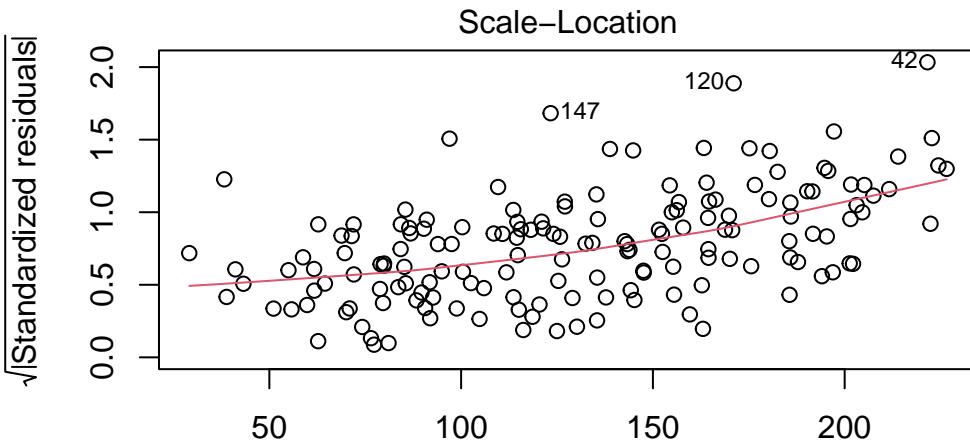
5. Analizar el cumplimiento del supuesto de homocedasticidad de errores.

```
modelo1 |> plot(which=1)
```



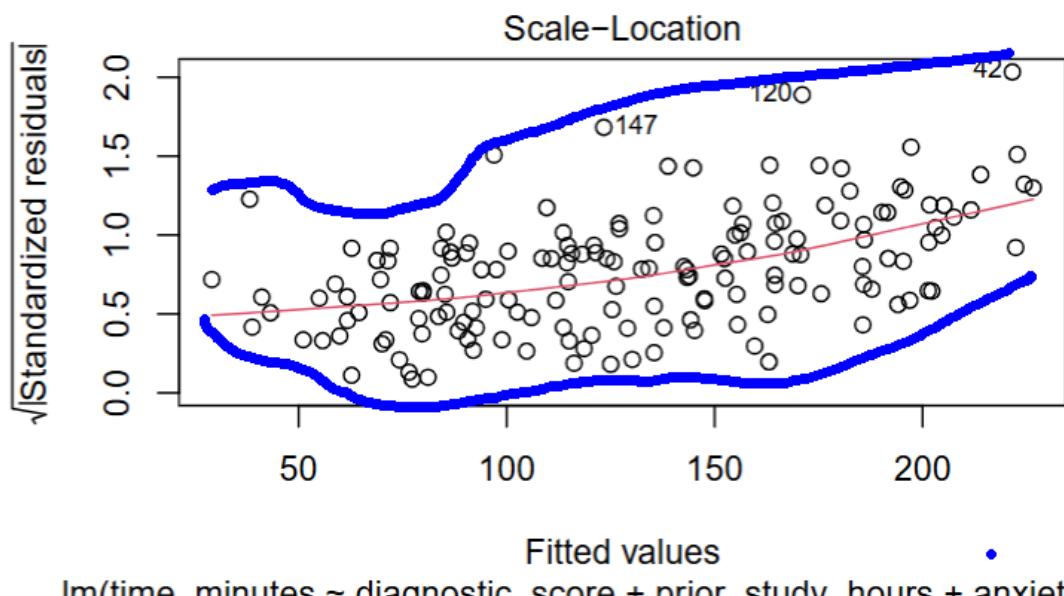
```
lm(time_minutes ~ diagnostic_score + prior_study_hours + anxiety_level)
```

```
modelo1 |> plot(which=3)
```



Fitted values  
 $lm(\text{time\_minutes} \sim \text{diagnostic\_score} + \text{prior\_study\_hours} + \text{anxiety\_level})$

En la primera gráfica (residuals vs fitted), las líneas que deberían “envolver” los puntos divergen cuando los valores ajustados crecen, de modo que los puntos no están homogéneamente distribuidos. Esto es una señal de heterocedasticidad.



$lm(\text{time\_minutes} \sim \text{diagnostic\_score} + \text{prior\_study\_hours} + \text{anxiety\_level})$

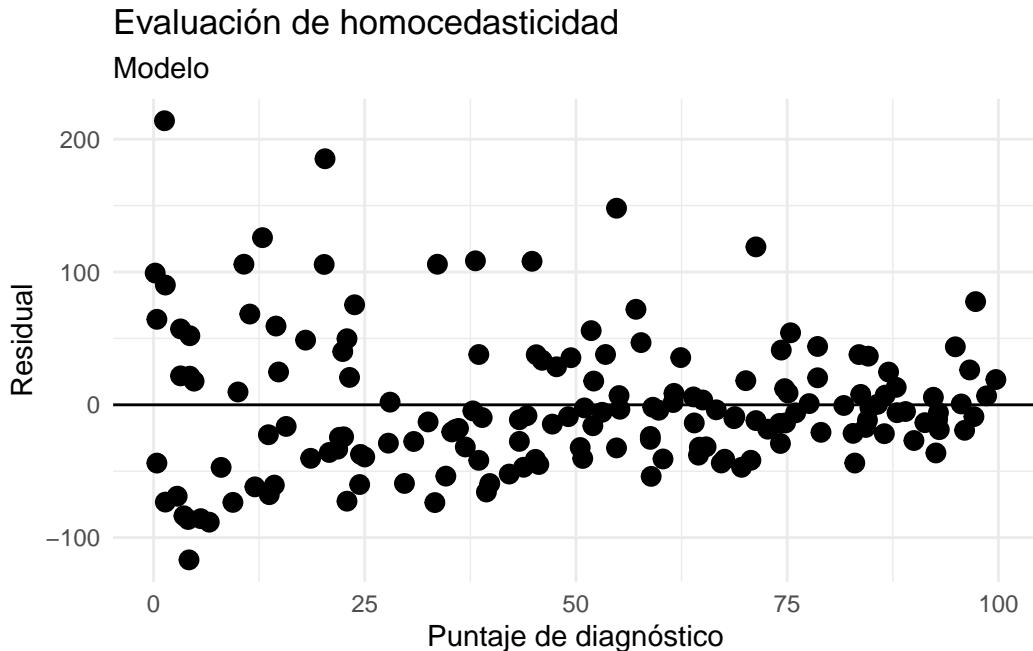
A diferencia de la primera gráfica, en Scale - Location no se aprecia una gran diferencia en la

variabilidad de los puntos.

```
library(broom)
```

```
Warning: package 'broom' was built under R version 4.4.3
```

```
modelo1 |> augment() |>
  ggplot(aes(x=diagnostic_score,y=.resid))+  
  geom_point(size = 3) +  
  geom_hline(yintercept=0)+  
  labs(x = "Puntaje de diagnóstico",  
       y = "Residual",  
       title = "Evaluación de homocedasticidad",  
       subtitle = "Modelo") +  
  theme_minimal()
```



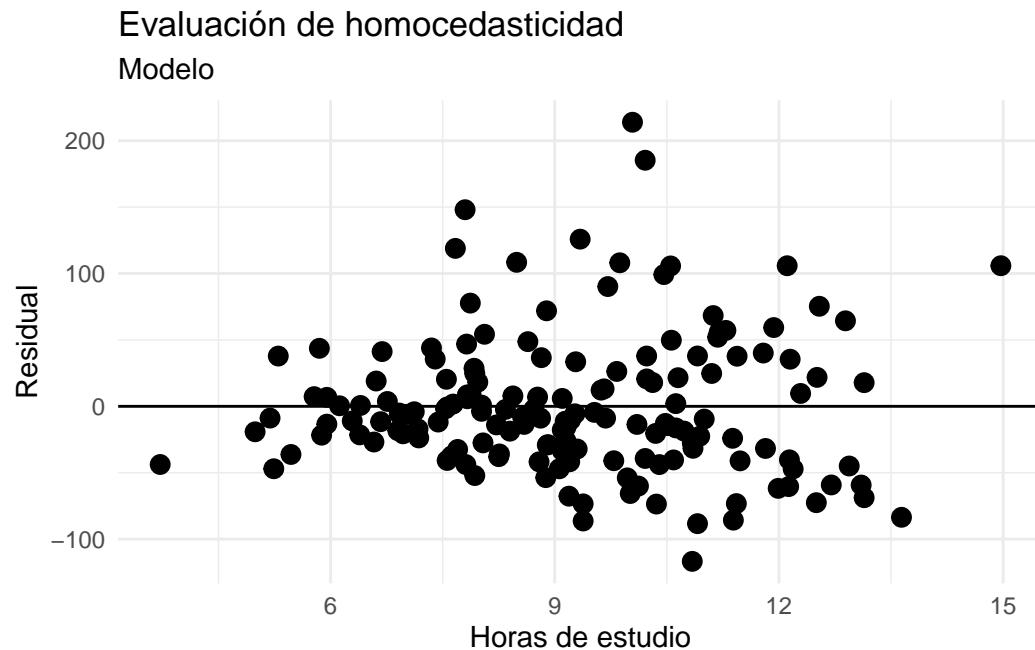
Los residuales tienden a ser más dispersos cuando el puntaje de diagnóstico es bajo.

```
modelo1 |> augment() |>
  ggplot(aes(x=prior_study_hours,y=.resid))+  
  geom_point(size = 3) +  
  geom_hline(yintercept=0)+
```

```

  labs(x = "Horas de estudio",
       y = "Residual",
       title = "Evaluación de homocedasticidad",
       subtitle = "Modelo")+
  theme_minimal()

```

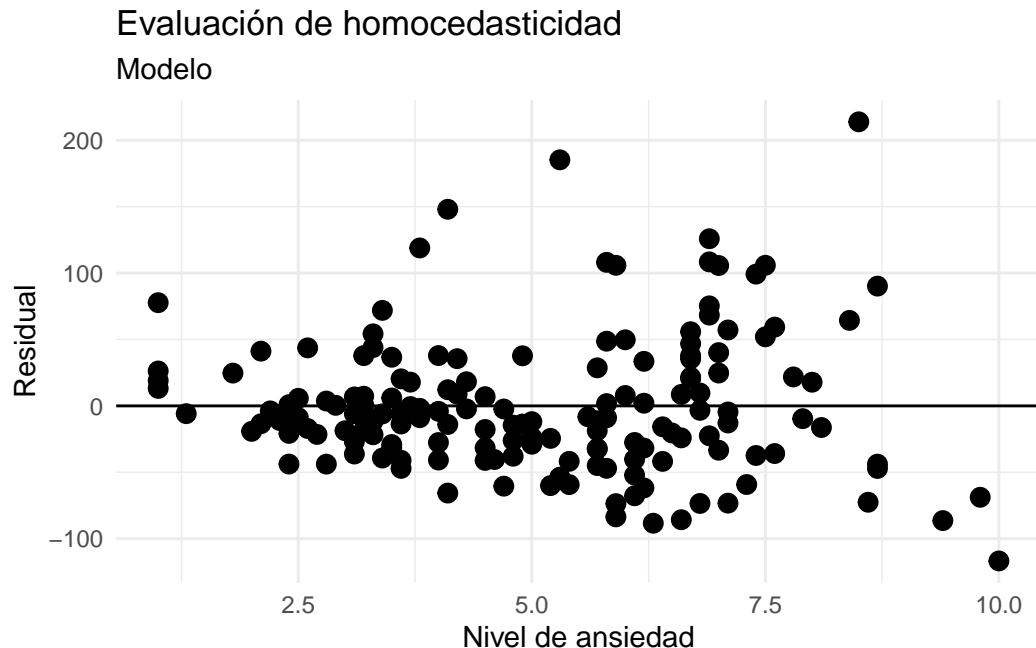


Los residuales presentan baja dispersión cuando las horas de estudio son inferiores.

```

modelo1 |> augment() |>
ggplot(aes(x=anxiety_level,y=.resid))+ 
  geom_point(size = 3) +
  geom_hline(yintercept=0) +
  labs(x = "Nivel de ansiedad",
       y = "Residual",
       title = "Evaluación de homocedasticidad",
       subtitle = "Modelo")+
  theme_minimal()

```



Los residuales presentan mayor dispersión cuando los niveles de ansiedad son altos.

Entonces, al formarse patrones de residuales según los valores de la variable respuesta, es una evidencia más de que existe heterocedasticidad.

$H_0$ : la varianza de los errores es constante

$H_1$ : la varianza de los errores no es constante

$\alpha = 0.05$

```
library(car)
```

Cargando paquete requerido: carData

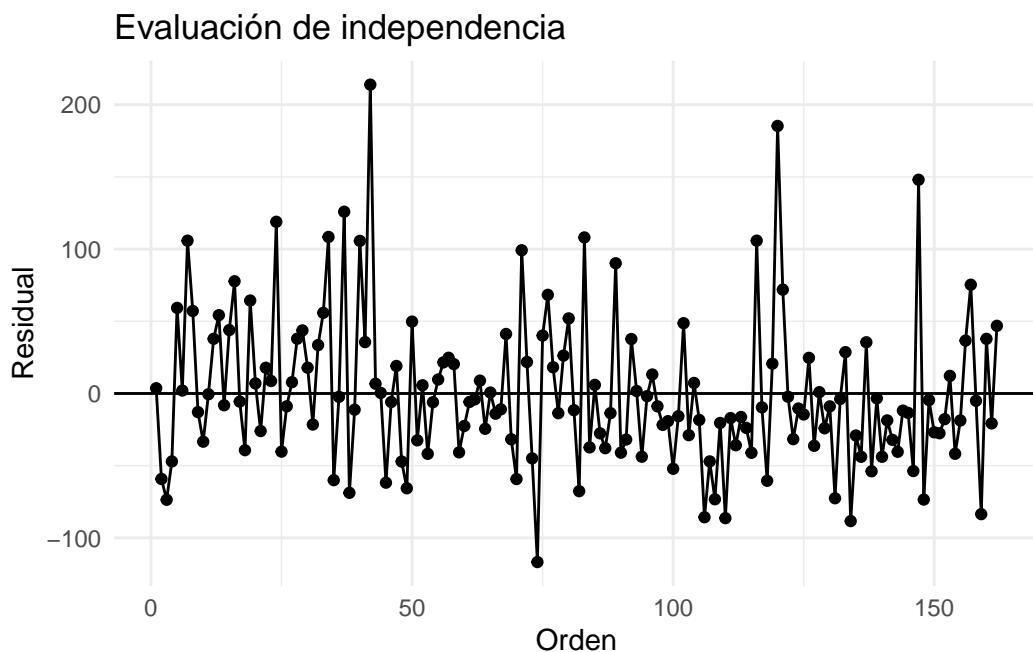
```
modelo1 |> ncvTest()
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 43.15078, Df = 1, p = 5.068e-11
```

Se rechaza la hipótesis nula, por lo tanto **no se verifica el supuesto de homogeneidad de varianzas de los errores.**

6. Analizar el cumplimiento del supuesto de independencia de errores.

```
data.frame(res1) |>
  ggplot(aes(x=1:nrow(datos),y=res1))+
  geom_point(size = 1.5) +
  geom_line()+
  geom_hline(yintercept=0)+
  labs(x = "Orden", y = "Residual", title = "Evaluación de independencia") +
  theme_minimal()
```



Visualizamos una primera evidencia de independencia de errores por la aleatoriedad de los puntos

```
library(ggfortify)
```

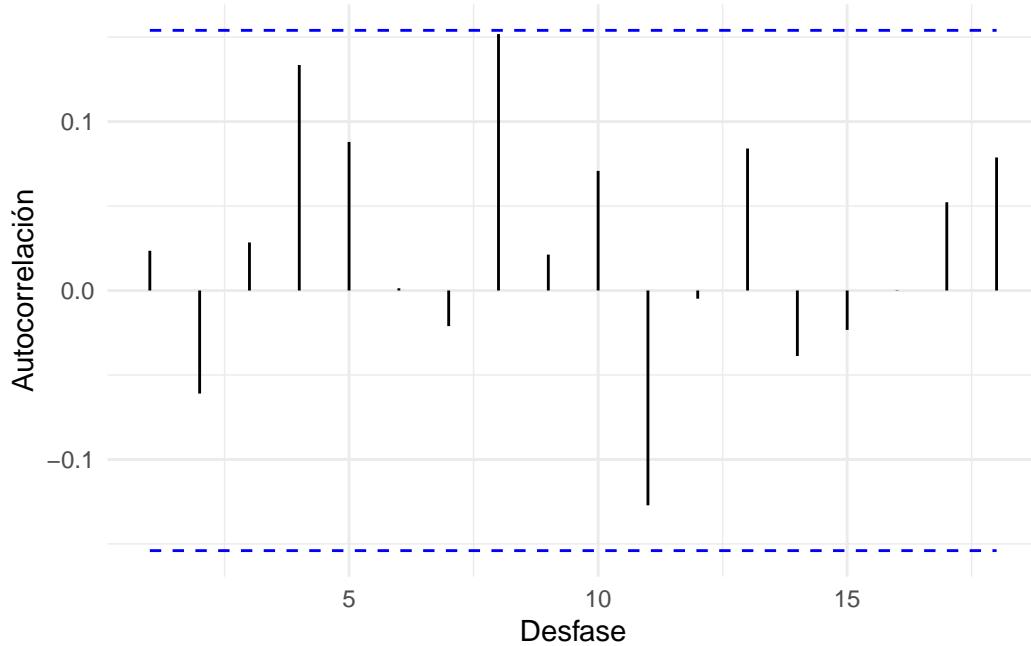
Warning: package 'ggfortify' was built under R version 4.4.3

```
res1 |>
  TSA::acf(lag = 18, plot=F) |>
  autoplot() +
  labs(x = "Desfase",y = "Autocorrelación") +
  theme_minimal()
```

```

Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
  i Please use tidy evaluation idioms with `aes()``.
  i See also `vignette("ggplot2-in-packages")` for more information.
  i The deprecated feature was likely used in the ggfortify package.
Please report the issue at <https://github.com/sinhrks/ggfortify/issues>.

```



Todas las autocorrelaciones son estadísticamente iguales a cero (están dentro de las límites azules de confianza), lo que evidenciaría independencia de errores.

$H_0$ : Los errores son independientes

$H_1$ : Los errores no son independientes

$\alpha = 0.05$

```

library(car)
modelo1 |>
  durbinWatsonTest(alternative = "two.sided",
                     max.lag = 10,
                     reps = 1e5)

```

lag	Autocorrelation	D-W Statistic	p-value
1	0.02352775	1.947898	0.72680
2	-0.06098949	2.107944	0.45368

```

3      0.02844076    1.913435 0.67970
4      0.13346970    1.682359 0.07820
5      0.08793277    1.765341 0.23436
6      0.00130936    1.925615 0.92214
7     -0.02107587    1.941718 0.91036
8      0.15179522    1.587714 0.04278
9      0.02125529    1.844428 0.72778
10     0.07080258    1.742446 0.36128
Alternative hypothesis: rho[lag] != 0

```

Se recomienda evaluar qué sucede a cada 8 filas, porque se ha encontrado una ligera autocorrelación en ese desfase. Por lo demás, no se evidencia falta de independencia.

7. En caso de incumplimiento de modelo, aplicar las siguientes transformaciones y comparar sus efectos: Raíz cuadrada, logaritmo, Box Cox, Mínimos cuadrados ponderados.

El principal problema detectado es la falta de normalidad y homocedasticidad de errores

- a. Transformación raíz cuadrada (modelo2)

```
library(dplyr)
```

```
#####
##### Warning from 'xts' package #####
#
# The dplyr lag() function breaks how base R's lag() function is supposed to #
# work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
# source() into this session won't work correctly.                           #
#
# Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
# conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
# dplyr from breaking base R's lag() function.                                #
#
# Code in packages is not affected. It's protected by R's namespace mechanism #
# Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning. #
#
#####
```

Adjuntando el paquete: 'dplyr'

```
The following object is masked from 'package:car':
```

```
recode
```

```
The following objects are masked from 'package:xts':
```

```
first, last
```

```
The following objects are masked from 'package:stats':
```

```
filter, lag
```

```
The following objects are masked from 'package:base':
```

```
intersect, setdiff, setequal, union
```

```
datos |>
  mutate(raiz_tiempo = sqrt(time_minutes)) -> datos
modelo2 <- lm(raiz_tiempo ~ diagnostic_score + prior_study_hours + anxiety_level, datos)
modelo2 |> residuals() -> res2
```

b. Transformación logaritmo (modelo3)

```
library(dplyr)
datos |>
  mutate(log_tiempo = log(time_minutes)) -> datos
modelo3 <- lm(log_tiempo ~ diagnostic_score + prior_study_hours + anxiety_level, datos)
modelo3 |> residuals() -> res3
```

c. Transformación Box Cox (modelo3)

```
library(MASS)
```

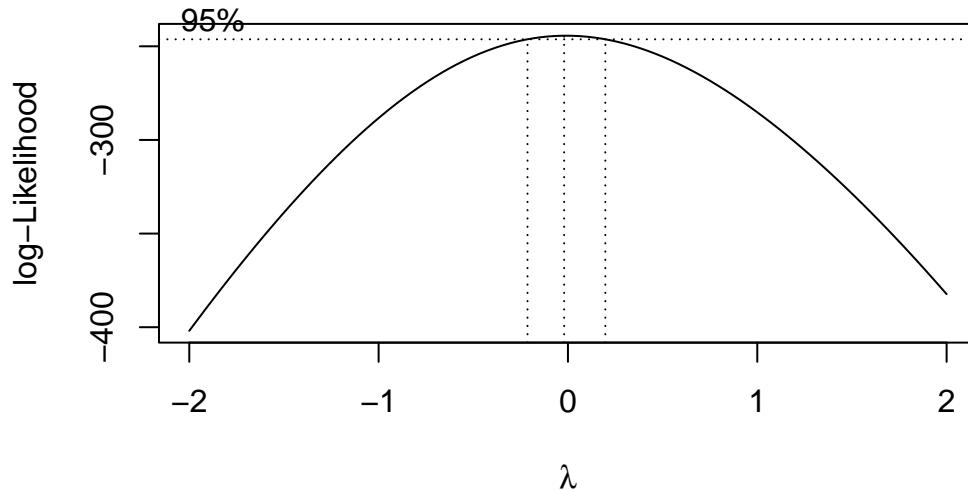
```
Warning: package 'MASS' was built under R version 4.4.3
```

```
Adjuntando el paquete: 'MASS'
```

```
The following object is masked from 'package:dplyr':
```

```
select
```

```
modelo1 |> boxcox()
```



Box Cox sugiere usar  $\lambda = 0$ , lo que significa la transformación logaritmo, la cual ya se ejecutó en el caso b.

d. Mínimos cuadrados ponderados (modelo4)

```
pesos = datos$diagnostic_score
modelo4 <- lm(time_minutes ~ diagnostic_score + prior_study_hours + anxiety_level,
               datos,
               weights = pesos)
modelo4 |> residuals() -> res4
```

8. Presentar un cuadro comparativo y elegir el mejor modelo. Resolver las preguntas que siguen utilizando dicho mejor modelo.

**Analizando el modelo 2:**

$H_0$  : Los errores siguen una distribución Normal

$H_1$  : Los errores no siguen una distribución Normal

```
res2 |> shapiro.test()
```

```
Shapiro-Wilk normality test
```

```
data: res2  
W = 0.97483, p-value = 0.004706
```

No se verifica el supuesto de normalidad de errores.

$H_0$  : Los errores tienen varianza constante

$H_1$  : Los errores no tienen varianza constante

```
modelo2 |> ncvTest()
```

```
Non-constant Variance Score Test  
Variance formula: ~ fitted.values  
Chisquare = 14.51645, Df = 1, p = 0.00013894
```

No se verifica el supuesto de homocedasticidad de errores.

$H_0$  : Los errores son independientes

$H_1$  : Los errores no son independientes

```
modelo2 |> durbinWatsonTest(alternative ="two.sided",  
                           max.lag =10,reps =1e5)
```

```
lag Autocorrelation D-W Statistic p-value  
 1      0.045024847    1.902364 0.52850  
 2     -0.053794422    2.089951 0.52562  
 3      0.028832099    1.904275 0.64334  
 4      0.188419952    1.557898 0.00982  
 5      0.077001274    1.773309 0.25148  
 6      0.018232946    1.877428 0.69824  
 7     -0.002154245    1.891370 0.83228  
 8      0.154778553    1.567250 0.02744  
 9      0.054505065    1.760998 0.38236  
10     0.073129866    1.720994 0.28782  
Alternative hypothesis: rho[lag] != 0
```

No se verifica la independencia de errores, se debe analizar qué sucede a cada 4 y 8 observaciones.

Por tanto, el modelo 2 queda descartado.

### Pasemos al modelo 3

$H_0$  : Los errores siguen una distribución Normal

$H_1$  : Los errores no siguen una distribución Normal

```
res3 |> shapiro.test()
```

```
Shapiro-Wilk normality test

data: res3
W = 0.99473, p-value = 0.8319
```

No se rechaza  $H_0$ , por lo que se cumple el supuesto de normalidad de errores.

$H_0$  : Los errores tienen varianza constante

$H_1$  : Los errores no tienen varianza constante

```
modelo3 |> ncvTest()
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.04791717, Df = 1, p = 0.82673
```

No se rechaza  $H_0$ , por lo que se cumple el supuesto de homocedasticidad de errores.

$H_0$  : Los errores son independientes

$H_1$  : Los errores no son independientes

```
modelo3 |> durbinWatsonTest(alternative ="two.sided",
                               max.lag =10,reps =1e5)
```

```

lag Autocorrelation D-W Statistic p-value
 1      0.05869923   1.872924 0.41730
 2     -0.02091292   2.022674 0.82802
 3      0.02188845   1.913849 0.68672
 4      0.22678188   1.470508 0.00120
 5      0.07461921   1.769475 0.24070
 6      0.03050955   1.845951 0.55914
 7      0.01046771   1.862486 0.69584
 8      0.14257847   1.581114 0.03300
 9      0.09805036   1.661026 0.12334
10      0.07459294   1.704683 0.24148
Alternative hypothesis: rho[lag] != 0

```

No se verifica la independencia de errores, se debe analizar qué sucede a cada 4 y 8 observaciones.

Sugerencia: usar un modelo autorregresivo de errores:

$$\epsilon_t = \alpha_4 \epsilon_{t-4} + \alpha_8 \epsilon_{t-8}$$

**Finalmente, el análisis del modelo 4:**

$H_0$  : Los errores siguen una distribución Normal

$H_1$  : Los errores no siguen una distribución Normal

```
res4 |> shapiro.test()
```

```

Shapiro-Wilk normality test

data: res4
W = 0.92603, p-value = 2.208e-07

```

No se cumple el supuesto de normalidad de errores.

$H_0$  : Los errores tienen varianza constante

$H_1$  : Los errores no tienen varianza constante

```
modelo4 |> ncvTest()
```

```

Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.6684119, Df = 1, p = 0.41361

```

Sí se cumple el supuesto de homogeneidad de varianzas de los errores.

$H_0$  : Los errores son independientes

$H_1$  : Los errores no son independientes

```
modelo4 |> durbinWatsonTest(alternative = "two.sided",
                                max.lag = 10, reps = 1e5)
```

```
lag Autocorrelation D-W Statistic p-value
 1    0.026839345   1.940179 0.68410
 2   -0.069580460   2.124465 0.39310
 3    0.032068349   1.906309 0.64588
 4    0.127831539   1.693911 0.09294
 5    0.090711518   1.758759 0.22468
 6    0.001173407   1.924122 0.92168
 7   -0.029612646   1.957139 0.84302
 8    0.156178634   1.575426 0.03754
 9    0.030687072   1.822355 0.62224
10    0.067236864   1.747425 0.37570
Alternative hypothesis: rho[lag] != 0
```

No se verifica la independencia de errores, se debe analizar qué sucede a cada 8 observaciones.

**EN CONCLUSIÓN**, se elige el modelo 3:

```
modelo3 <- lm(log_tiempo ~ diagnostic_score + prior_study_hours + anxiety_level, datos)
```

9. Escribir la ecuación estimada del mejor modelo e interpretar sus coeficientes.

```
modelo3 |> coef()
```

```
(Intercept)  diagnostic_score prior_study_hours      anxiety_level
 5.42052828       -0.01186468        -0.04394217        0.06001856
```

- $Y$ : Tiempo de resolución del examen (variable respuesta)
- $X_1$ : puntaje en la prueba diagnóstica (variable independiente)
- $X_2$ : Horas de estudio (variable independiente)
- $X_3$ : Nivel de ansiedad (variable independiente)

$$\hat{log(y)} = 5.42 - 0.02X_1 - 0.04X_2 + 0.06X_3$$

Para la interpretación de los coeficientes, ver [aquí](#).

10. Ejecutar la prueba de hipótesis global del modelo, presentando el cuadro ANVA.

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0$$

$$H_1 : \text{Al menos un } \beta_j \text{ es distinto de cero}$$

$$\alpha = 0.05$$

```
y <- datos$log_tiempo
X <- model.matrix(log_tiempo ~ diagnostic_score + prior_study_hours +
                    anxiety_level, datos)
aov(y ~ X) |> summary()
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
X	3	24.78	8.260	63.92	<2e-16 ***						
Residuals	158	20.42	0.129								
---											
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

$$F_{calc} = 63.92$$

$$F_{crit} = F_{0.95, 3, 158} = 2.66$$

```
qf(0.95, 3, 158)
```

[1] 2.661829

Dado que  $F_{calc} > F_{crit}$  se rechaza  $H_0$

Equivalentemente,  $pvalor < \alpha$ . Se rechaza  $H_0$

Al menos una de las variables se relaciona linealmente con el logaritmo del tiempo.

11. Ejecutar las pruebas de hipótesis individuales

```
library(broom)
modelo3 |> tidy()
```

```
# A tibble: 4 x 5
  term          estimate std.error statistic p.value
  <chr>        <dbl>     <dbl>      <dbl>    <dbl>
1 (Intercept)   5.42      0.323     16.8  6.96e-37
2 diagnostic_score -0.0119   0.00213    -5.58 1.01e- 7
3 prior_study_hours -0.0439   0.0200    -2.20 2.94e- 2
4 anxiety_level    0.0600   0.0281     2.14 3.40e- 2
```

$$H_0 : \beta_1 = 0 \quad H_1 : \beta_1 \neq 0 \quad \alpha = 0.05 \quad pvalor = 10^{-7}$$

Se rechaza la hipótesis nula, por lo tanto, el puntaje diagnóstico influye linealmente en el logaritmo del tiempo.

$$H_0 : \beta_2 = 0 \quad H_1 : \beta_2 \neq 0 \quad \alpha = 0.05 \quad pvalor = 0.0294$$

Se rechaza la hipótesis nula, por lo tanto, las horas de estudio influyen linealmente en el logaritmo del tiempo.

$$H_0 : \beta_3 = 0 \quad H_1 : \beta_3 \neq 0 \quad \alpha = 0.05 \quad pvalor = 0.034$$

Se rechaza la hipótesis nula, por lo tanto, el nivel de ansiedad influye linealmente en el logaritmo del tiempo.

## 12. Interpretar el coeficiente de determinación ajustado

```
modelo3 |> summary()
```

Call:

```
lm(formula = log_tiempo ~ diagnostic_score + prior_study_hours +
  anxiety_level, data = datos)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.92379	-0.27009	0.00175	0.24166	0.92978

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	5.420528	0.323406	16.761	< 2e-16 ***
diagnostic_score	-0.011865	0.002125	-5.583	1.01e-07 ***
prior_study_hours	-0.043942	0.019992	-2.198	0.0294 *
anxiety_level	0.060019	0.028062	2.139	0.0340 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 0.3595 on 158 degrees of freedom
Multiple R-squared:  0.5483,    Adjusted R-squared:  0.5397
F-statistic: 63.92 on 3 and 158 DF,  p-value: < 2.2e-16

```

El modelo de regresión explica el 53.97% de la variabilidad del logaritmo del tiempo.

13. Estimar el **tiempo mediano** del examen final si el puntaje diagnóstico fue de 65 puntos, estudió 10 horas y el nivel de ansiedad es 5.5, e indicar si coincide con el tiempo medio. Además, reportar el intervalo de confianza.

```

xnuevo <- data.frame(diagnostic_score = 65,
                      prior_study_hours = 10,
                      anxiety_level = 5.5)
modelo3 |> predict(newdata = xnuevo,
                     interval = "confidence") |> exp() # confidence = media(na)

```

	fit	lwr	upr
1	93.69121	83.59135	105.0114

Cuando el puntaje diagnóstico fue de 65 puntos, estudió 10 horas y el nivel de ansiedad es 5.5, se estima que el tiempo mediano de resolución del examen es de 93.69 minutos, con un intervalo del 95% de confianza que va de 83.59 a 105.01 minutos.

14. Predecir el tiempo que tardará en resolver el examen final **un estudiante** con puntaje diagnóstico de 50 puntos, que estudió 4.5 horas y con nivel de ansiedad 8. Compararlo con uno que tuvo 85 puntos diagnósticos, 12 horas de estudio y nivel de ansiedad 2.3

```

xnuevo <- data.frame(diagnostic_score = c(50,85),
                      prior_study_hours = c(4.5,12),
                      anxiety_level = c(8,2.3))
modelo3 |> predict(newdata = xnuevo,
                     interval = "prediction") |> exp() # prediction = predicción individual

```

	fit	lwr	upr
1	165.62114	77.83408	352.4210
2	55.85544	26.65651	117.0382