

MC6104 - Diseño de Experimentos

I Semestre 2023

Apuntes de la clase del jueves 13 de junio

Apuntador:
Luis Diego Hidalgo Blanco

I. DISEÑOS MULTIFACTORIALES 2^k

Para diseños que se realizan cuando tenemos una serie de factores que únicamente tienen dos niveles. Se pueden condensar en un diseño de las diferentes combinaciones para analizarlo de una manera más conveniente. Se utilizan para los k factores (relevantes) que únicamente tienen dos niveles. Ver ejemplo en sección IV.

- K = número de factores.
- Ejemplo factores con 2 niveles:
 - Inclinación.
 - * 30 - A
 - * 60 - B
 - Altura.
 - * 10 - X
 - * 20 - Y
 - Factores condensados en 2^k :
 - * Factor: I-A
 - A - X
 - A - Y
 - B - X
 - B - Y
- ¿Casos cuándo se pueden utilizar?
 - En promedios cuando un factor está o no presente.

II. TRANSFORMACIÓN DE DATOS

- Se realiza cuando no se pueden cumplir los supuestos, se obligan a las variables a tener distribuciones similares, también sirve para reducir sesgos (a los lados).
- Típicamente se transforma la variable dependiente, pero en modelos complejos y de regresión múltiple se puede transformar tanto la dependiente como la independiente.
- La transformación de datos minimiza la distancia entre los valores, por ejemplo si se le calcula la raíz cuadrada a 50 y 100, los resultados serían 7.07 y 10, estos están más cercanos comparado a los valores originales.
- Si no hubo forma de analizar los datos con Anova (agotar las oportunidades de transformar los residuos a normal), se puede utilizar la herramienta de las pruebas no paramétricas.
- No hay nada prohibido, solo hay que ser cuidadoso, lo importante es cómo se reportan los resultados, ser explícitos de que se realizó transformación.

- "Se identificó una diferencia significativa en la transformación logarítmica del rendimiento"
- Para estadísticas resumen, podemos presentar los promedios de los valores transformados
- Se pueden destransformar esos promedios a unidades originales.
 - Si hicimos el análisis con los datos transformados, no se vale presentar los resultados con los datos originales, lo que si se vale ejecutar es todo el análisis con los datos transformados y al final destransformar los resultados para volver a las unidades originales.
- Con logaritmo, se suele sumar una constante para evitar el caso $\log(0)$.
- Camino de la luz:
 - Raíz cuadrada.
 - Raíz cúbica.
 - Logaritmo
- Existen otros métodos de transformaciones:
 - Tukey's Ladder of Powers.
 - Box-Cox.

III. EXPERIMENTO CUALITATIVO HECHO EN CLASE

Instrucciones:

- 1) Abrir Google en la pestaña de imágenes.
- 2) Buscar imágenes sobre experimentos en animales.
- 3) Ver con detalle y detenimiento cada una de las imágenes mostradas.
- 4) Desplace el cursor para ver más resultados.
- 5) Repetir paso 3.



Fig. 1. Ratón de laboratorio

Preguntas a responder:

- ¿Cómo se sintió con lo que vio?

- ¿Es eso correcto o ético?
- ¿Cuáles cuestionamientos vienen a su mente?
- ¿Es algo que apoyaría?
- ¿Qué se puede hacer?

Posturas:

- 1) A favor de la experimentación en animales.
 - Si se minimiza el sufrimiento.
 - Se obtienen beneficios a los humanos que no podrían conseguirse de ninguna otra forma.
 - Brinda un aporte a la ciencia.
- 2) En contra de la experimentación en animales.
 - Inaceptable porque causa sufrimiento.
 - Los beneficios para los humanos no están probados.
 - Cualquier otro beneficio se puede obtener de otras maneras.

Comentarios hechos:

- En la medicina parece ser necesario, puede ser que muchos de nosotros no estemos aquí si no hubiera sido por el resultado de un experimento realizado en un animal.
- Se debe minimizar el sufrimiento.
- Es un mal necesario.
- Beneficios a los humanos aplica en salud y calidad/longevidad de vida.

Daño vs beneficio:

- A favor
 - Los experimentos en animales dan tantos beneficios que es moralmente aceptable lastimar a unos cuantos animales.
- En contra
 - El sufrimiento causado y el número de animales involucrados es tan alto que los beneficios para la humanidad no dan una justificación moral.

Beneficios para el ser humano:

- Fármacos seguros en el ser humano:
 - Los experimentos en animales no son utilizados para probar que fármacos son seguros para los seres humanos.
 - Se utiliza de manera preliminar para decidir si va a ser probado en pocas personas a bajas dosis. Un filtro preliminar.
- Análisis de caso:
 - Se tienen cuatro fármacos para curar el sida:
 - * Fármaco A mató todas las ratas, ratones y perros.
 - * Fármaco B mató todos los perros y ratas.
 - * Fármaco C mató a todos los ratones y ratas.
 - * Fármaco D se le dio a todos los animales hasta dosis muy altas y no causó efectos negativos.
 - ¿Cuál probaríamos en voluntarios jóvenes?
 - * Fármaco D
 - * Ninguno
 - Fármaco D debe darse en condiciones controladas.

Guillotina:

Se conversó sobre la posibilidad de mantener la conciencia después de ser ejecutado por la guillotina. Se realizaron experimentos en ratas para evaluar lo anteriormente mencionado.



Fig. 2. Guillotina

Uso de la guillotina en el siglo XX:

- La guillotina siguió siendo el método oficial de ejecución en Francia hasta que se abolió la pena de muerte en 1981.
- Alemania Nazi usó la guillotina entre 1933 y 1945 para ejecutar a 16 500 prisioneros, 10 000 de ellos solo en 1944 y 1945.
- En el Caribe, se usó muy raramente en Guadalupe y Martinica, la última vez en Fort-de-France en 1965.
- En América del Sur, la guillotina solo se usó en la Guayana Francesa, donde unas 150 personas fueron decapitadas entre 1850 y 1945.
- En Suecia, la decapitación se convirtió en el método obligatorio de ejecución en 1866, reemplazó a la decapitación manual en 1903 y solo se usó una vez.

IV. EJEMPLO ANOVA MULTIFACTORIAL 2^k

Contexto: experimento para evaluar algoritmos en la red neuronal. Se quiere evaluar el desempeño al entrenar con dos conjuntos de datos distintos y el hecho de utilizar un acelerador de software o de hardware. Entre más pequeño el conjunto de datos para entrenar más barato. SW = \$10,000, HW = eBay ASICS.

Nomenclatura:

- NA: acelerador no aplicado.
- SW: acelerador por software = factor 2 niveles.
- HW: acelerador por hardware = factor 2 niveles.

Tabla ANOVA:

		Factor	
	Método de entrenamiento	Conjunto de datos	Acelerador
Niveles	Algoritmo A	MT500	NA-NA
	Algoritmo B	MT1000	NA-SW
	Algoritmo C	MT5000	HW-NA
		MT50000	HW-SW

Notas:

- Se prefiere escoger el MT500.
- Es indiferente si hay acelerador de software o no.
- ¿Por qué es válido hacer el análisis con datos transformados o por qué es válido reportar el análisis con datos transformados?
 - ¿Se introduce error?
 - * Se reduce.
 - Al ejecutar la transformación, se hace la prueba más difícil.

A. ANOVA Multifactorial 2^k

1) Carga inicial de datos

1. Carga inicial de datos.

```
if(!require(psych)){install.packages("psych")}
if(!require(FSA)){install.packages("FSA")}
if(!require(ggplot2)){install.packages("ggplot2")}
if(!require(car)){install.packages("car")}
if(!require(multcompView)){install.packages("multcompView")}
if(!require(lsmmeans)){install.packages("lsmmeans")}
if(!require(rcompanion)){install.packages("rcompanion")}
```

```
In <- ("
Algoritmo      Entrenamiento      Rendimiento      Acelerador
'Algoritmo A'   MT500             12000             NA-NA
'Algoritmo A'   MT500             14005             NA-NA
'Algoritmo A'   MT500             13508             NA-NA
'Algoritmo A'   MT500             9503              NA-NA
'Algoritmo A'   MT500             14004             NA-NA
'Algoritmo A'   MT1000            11502             NA-NA
'Algoritmo A'   MT1000            13006             NA-NA
'Algoritmo A'   MT1000            13252             NA-NA
'Algoritmo A'   MT1000            14253             NA-NA
'Algoritmo A'   MT1000            15003             NA-NA
'Algoritmo A'   MT5000            12504             NA-NA
'Algoritmo A'   MT5000            11504             NA-NA
'Algoritmo A'   MT5000            9500              NA-NA
'Algoritmo A'   MT5000            11506             NA-NA
'Algoritmo A'   MT5000            16000             NA-NA
'Algoritmo A'   MT50000           13008             NA-NA
'Algoritmo A'   MT50000           10506             NA-NA
'Algoritmo A'   MT50000           13005             NA-NA
'Algoritmo A'   MT50000           17002             NA-NA
'Algoritmo A'   MT50000           13008             NA-NA
'Algoritmo B'   MT500             11005             NA-NA
'Algoritmo B'   MT500             12007             NA-NA
'Algoritmo B'   MT500             12509             NA-NA
'Algoritmo B'   MT500             10504             NA-NA
'Algoritmo B'   MT500             12002             NA-NA
'Algoritmo B'   MT1000            12504             NA-NA
'Algoritmo B'   MT1000            13501             NA-NA
```

' Algoritmo B'	MT1000	13501	NA-NA
' Algoritmo B'	MT1000	13252	NA-NA
' Algoritmo B'	MT1000	15256	NA-NA
' Algoritmo B'	MT5000	12253	NA-NA
' Algoritmo B'	MT5000	11255	NA-NA
' Algoritmo B'	MT5000	10006	NA-NA
' Algoritmo B'	MT5000	11252	NA-NA
' Algoritmo B'	MT5000	14004	NA-NA
' Algoritmo B'	MT50000	12007	NA-NA
' Algoritmo B'	MT50000	11505	NA-NA
' Algoritmo B'	MT50000	14009	NA-NA
' Algoritmo B'	MT50000	15000	NA-NA
' Algoritmo B'	MT50000	12009	NA-NA
' Algoritmo C'	MT500	9000	NA-NA
' Algoritmo C'	MT500	11003	NA-NA
' Algoritmo C'	MT500	11505	NA-NA
' Algoritmo C'	MT500	9509	NA-NA
' Algoritmo C'	MT500	11003	NA-NA
' Algoritmo C'	MT1000	11508	NA-NA
' Algoritmo C'	MT1000	12508	NA-NA
' Algoritmo C'	MT1000	12506	NA-NA
' Algoritmo C'	MT1000	12254	NA-NA
' Algoritmo C'	MT1000	13253	NA-NA
' Algoritmo C'	MT5000	11255	NA-NA
' Algoritmo C'	MT5000	10257	NA-NA
' Algoritmo C'	MT5000	9500	NA-NA
' Algoritmo C'	MT5000	9255	NA-NA
' Algoritmo C'	MT5000	12009	NA-NA
' Algoritmo C'	MT50000	11000	NA-NA
' Algoritmo C'	MT50000	9509	NA-NA
' Algoritmo C'	MT50000	13009	NA-NA
' Algoritmo C'	MT50000	14005	NA-NA
' Algoritmo C'	MT50000	11001	NA-NA
' Algoritmo A'	MT500	12046	NA-SW
' Algoritmo A'	MT500	14589	NA-SW
' Algoritmo A'	MT500	13723	NA-SW
' Algoritmo A'	MT500	9799	NA-SW
' Algoritmo A'	MT500	14715	NA-SW
' Algoritmo A'	MT1000	11144	NA-SW
' Algoritmo A'	MT1000	13920	NA-SW
' Algoritmo A'	MT1000	13226	NA-SW
' Algoritmo A'	MT1000	14845	NA-SW
' Algoritmo A'	MT1000	15142	NA-SW
' Algoritmo A'	MT5000	12352	NA-SW
' Algoritmo A'	MT5000	11296	NA-SW
' Algoritmo A'	MT5000	9737	NA-SW
' Algoritmo A'	MT5000	11129	NA-SW
' Algoritmo A'	MT5000	16409	NA-SW
' Algoritmo A'	MT50000	13872	NA-SW
' Algoritmo A'	MT50000	10100	NA-SW
' Algoritmo A'	MT50000	13419	NA-SW
' Algoritmo A'	MT50000	17398	NA-SW
' Algoritmo A'	MT50000	13164	NA-SW
' Algoritmo B'	MT500	11047	NA-SW
' Algoritmo B'	MT500	12226	NA-SW
' Algoritmo B'	MT500	12105	NA-SW
' Algoritmo B'	MT500	10418	NA-SW
' Algoritmo B'	MT500	12446	NA-SW
' Algoritmo B'	MT1000	12156	NA-SW
' Algoritmo B'	MT1000	13968	NA-SW
' Algoritmo B'	MT1000	13891	NA-SW
' Algoritmo B'	MT1000	13778	NA-SW
' Algoritmo B'	MT1000	15448	NA-SW
' Algoritmo B'	MT5000	12441	NA-SW
' Algoritmo B'	MT5000	11767	NA-SW
' Algoritmo B'	MT5000	10340	NA-SW
' Algoritmo B'	MT5000	11306	NA-SW
' Algoritmo B'	MT5000	14565	NA-SW
' Algoritmo B'	MT50000	12725	NA-SW
' Algoritmo B'	MT50000	11169	NA-SW
' Algoritmo B'	MT50000	14749	NA-SW
' Algoritmo B'	MT50000	15566	NA-SW
' Algoritmo B'	MT50000	12239	NA-SW
' Algoritmo C'	MT500	9082	NA-SW
' Algoritmo C'	MT500	11887	NA-SW
' Algoritmo C'	MT500	11799	NA-SW
' Algoritmo C'	MT500	9300	NA-SW
' Algoritmo C'	MT500	11049	NA-SW
' Algoritmo C'	MT1000	11378	NA-SW
' Algoritmo C'	MT1000	12659	NA-SW
' Algoritmo C'	MT1000	12905	NA-SW
' Algoritmo C'	MT1000	12782	NA-SW
' Algoritmo C'	MT1000	13196	NA-SW

' Algoritmo C'	MT5000	11795	NA-SW
' Algoritmo C'	MT5000	10316	NA-SW
' Algoritmo C'	MT5000	9947	NA-SW
' Algoritmo C'	MT5000	9420	NA-SW
' Algoritmo C'	MT5000	12699	NA-SW
' Algoritmo C'	MT50000	11024	NA-SW
' Algoritmo C'	MT50000	9556	NA-SW
' Algoritmo C'	MT50000	13900	NA-SW
' Algoritmo C'	MT50000	14006	NA-SW
' Algoritmo C'	MT50000	11738	NA-SW
' Algoritmo A'	MT500	126572	HW-NA
' Algoritmo A'	MT500	140058	HW-NA
' Algoritmo A'	MT500	139580	HW-NA
' Algoritmo A'	MT500	92583	HW-NA
' Algoritmo A'	MT500	148057	HW-NA
' Algoritmo A'	MT1000	110078	HW-NA
' Algoritmo A'	MT1000	131942	HW-NA
' Algoritmo A'	MT1000	133797	HW-NA
' Algoritmo A'	MT1000	140026	HW-NA
' Algoritmo A'	MT1000	155479	HW-NA
' Algoritmo A'	MT5000	125809	HW-NA
' Algoritmo A'	MT5000	114264	HW-NA
' Algoritmo A'	MT5000	98797	HW-NA
' Algoritmo A'	MT5000	113400	HW-NA
' Algoritmo A'	MT5000	168898	HW-NA
' Algoritmo A'	MT50000	133452	HW-NA
' Algoritmo A'	MT50000	101641	HW-NA
' Algoritmo A'	MT50000	133155	HW-NA
' Algoritmo A'	MT50000	175156	HW-NA
' Algoritmo A'	MT50000	131945	HW-NA
' Algoritmo B'	MT500	110317	HW-NA
' Algoritmo B'	MT500	129244	HW-NA
' Algoritmo B'	MT500	127966	HW-NA
' Algoritmo B'	MT500	109783	HW-NA
' Algoritmo B'	MT500	122936	HW-NA
' Algoritmo B'	MT1000	128830	HW-NA
' Algoritmo B'	MT1000	134437	HW-NA
' Algoritmo B'	MT1000	138321	HW-NA
' Algoritmo B'	MT1000	132000	HW-NA
' Algoritmo B'	MT1000	157693	HW-NA
' Algoritmo B'	MT5000	121964	HW-NA
' Algoritmo B'	MT5000	119872	HW-NA
' Algoritmo B'	MT5000	106654	HW-NA
' Algoritmo B'	MT5000	112666	HW-NA
' Algoritmo B'	MT5000	145535	HW-NA
' Algoritmo B'	MT50000	127938	HW-NA
' Algoritmo B'	MT50000	115179	HW-NA
' Algoritmo B'	MT50000	143021	HW-NA
' Algoritmo B'	MT50000	150357	HW-NA
' Algoritmo B'	MT50000	121216	HW-NA
' Algoritmo C'	MT500	95474	HW-NA
' Algoritmo C'	MT500	113776	HW-NA
' Algoritmo C'	MT500	117473	HW-NA
' Algoritmo C'	MT500	92900	HW-NA
' Algoritmo C'	MT500	115582	HW-NA
' Algoritmo C'	MT1000	115279	HW-NA
' Algoritmo C'	MT1000	122184	HW-NA
' Algoritmo C'	MT1000	124770	HW-NA
' Algoritmo C'	MT1000	128403	HW-NA
' Algoritmo C'	MT1000	135219	HW-NA
' Algoritmo C'	MT5000	112562	HW-NA
' Algoritmo C'	MT5000	108736	HW-NA
' Algoritmo C'	MT5000	91064	HW-NA
' Algoritmo C'	MT5000	98171	HW-NA
' Algoritmo C'	MT5000	120277	HW-NA
' Algoritmo C'	MT50000	111299	HW-NA
' Algoritmo C'	MT50000	90193	HW-NA
' Algoritmo C'	MT50000	135178	HW-NA
' Algoritmo C'	MT50000	146158	HW-NA
' Algoritmo C'	MT50000	113845	HW-NA
' Algoritmo A'	MT500	124252	HW-SW
' Algoritmo A'	MT500	143833	HW-SW
' Algoritmo A'	MT500	138907	HW-SW
' Algoritmo A'	MT500	91010	HW-SW
' Algoritmo A'	MT500	143901	HW-SW
' Algoritmo A'	MT1000	116563	HW-SW
' Algoritmo A'	MT1000	136455	HW-SW
' Algoritmo A'	MT1000	130411	HW-SW
' Algoritmo A'	MT1000	140060	HW-SW
' Algoritmo A'	MT1000	154308	HW-SW
' Algoritmo A'	MT5000	124480	HW-SW
' Algoritmo A'	MT5000	111552	HW-SW
' Algoritmo A'	MT5000	99135	HW-SW

```

' Algoritmo A'      MT5000      110208      HW-SW
' Algoritmo A'      MT5000      167228      HW-SW
' Algoritmo A'      MT50000     134267      HW-SW
' Algoritmo A'      MT50000     102119      HW-SW
' Algoritmo A'      MT50000     138036      HW-SW
' Algoritmo A'      MT50000     171632      HW-SW
' Algoritmo A'      MT50000     130666      HW-SW
' Algoritmo B'      MT500      116942      HW-SW
' Algoritmo B'      MT500      129721      HW-SW
' Algoritmo B'      MT500      128834      HW-SW
' Algoritmo B'      MT500      100390      HW-SW
' Algoritmo B'      MT500      127771      HW-SW
' Algoritmo B'      MT1000     121789      HW-SW
' Algoritmo B'      MT1000     135311      HW-SW
' Algoritmo B'      MT1000     136587      HW-SW
' Algoritmo B'      MT1000     139664      HW-SW
' Algoritmo B'      MT1000     151543      HW-SW
' Algoritmo B'      MT5000     128962      HW-SW
' Algoritmo B'      MT5000     110157      HW-SW
' Algoritmo B'      MT5000     106129      HW-SW
' Algoritmo B'      MT5000     114634      HW-SW
' Algoritmo B'      MT5000     143337      HW-SW
' Algoritmo B'      MT50000     129292      HW-SW
' Algoritmo B'      MT50000     117502      HW-SW
' Algoritmo B'      MT50000     143687      HW-SW
' Algoritmo B'      MT50000     153488      HW-SW
' Algoritmo B'      MT50000     129773      HW-SW
' Algoritmo C'      MT500      99920      HW-SW
' Algoritmo C'      MT500      110833      HW-SW
' Algoritmo C'      MT500      117879      HW-SW
' Algoritmo C'      MT500      96441      HW-SW
' Algoritmo C'      MT500      119688      HW-SW
' Algoritmo C'      MT1000     117995      HW-SW
' Algoritmo C'      MT1000     122984      HW-SW
' Algoritmo C'      MT1000     120317      HW-SW
' Algoritmo C'      MT1000     120213      HW-SW
' Algoritmo C'      MT1000     137806      HW-SW
' Algoritmo C'      MT5000     117014      HW-SW
' Algoritmo C'      MT5000     105529      HW-SW
' Algoritmo C'      MT5000     98755      HW-SW
' Algoritmo C'      MT5000     96010      HW-SW
' Algoritmo C'      MT5000     126548      HW-SW
' Algoritmo C'      MT50000     113527      HW-SW
' Algoritmo C'      MT50000     99385      HW-SW
' Algoritmo C'      MT50000     136573      HW-SW
' Algoritmo C'      MT50000     141965      HW-SW
' Algoritmo C'      MT50000     111994      HW-SW
")

```

```

# Se introduce la tabla.
Data <- read.table(textConnection(ln), header=TRUE)

# Se ordenan los datos segun los ingresamos. (Evitar orden alfabetico por R).
Data$Entrenamiento <- factor(Data$Entrenamiento, levels = unique(Data$Entrenamiento))

```

2) Visualización preliminar

```

library(psych)
headTail(Data)
str(Data)
summary(Data)
rm(ln)

```

```

headTail(Data)
  Algoritmo Entrenamiento Rendimiento Acelerador
1 Algoritmo A      MT500      12000      NA-NA
2 Algoritmo A      MT500      14005      NA-NA
3 Algoritmo A      MT500      13508      NA-NA
4 Algoritmo A      MT500      9503      NA-NA
...      <NA>      <NA>      ...      <NA>
237 Algoritmo C      MT50000     99385      HW-SW
238 Algoritmo C      MT50000     136573     HW-SW
239 Algoritmo C      MT50000     141965     HW-SW
240 Algoritmo C      MT50000     111994     HW-SW
> str(Data)
'data.frame':   240 obs. of  4 variables:

```

```

$ Algoritmo      : chr  "Algoritmo A" "Algoritmo A" "Algoritmo A" "Algoritmo A" ...
$ Entrenamiento : Factor w/ 4 levels "MT500","MT1000",...: 1 1 1 1 2 2 2 2 ...
$ Rendimiento   : int   12000 14005 13508 9503 14004 11502 13006 13252 14253 15003 ...
$ Acelerador     : chr   "NA-NA" "NA-NA" "NA-NA" "NA-NA" ...
> summary(Data)
  Algoritmo      Entrenamiento  Rendimiento      Acelerador
Length:240      MT500 :60      Min.       : 9000      Length:240
Class :character MT1000 :60      1st Qu.: 12236      Class :character
Mode  :character MT5000 :60      Median    : 53796     Mode  :character
                MT50000:60      Mean      : 68432
                                3rd Qu.: 124309
                                Max.       : 175156

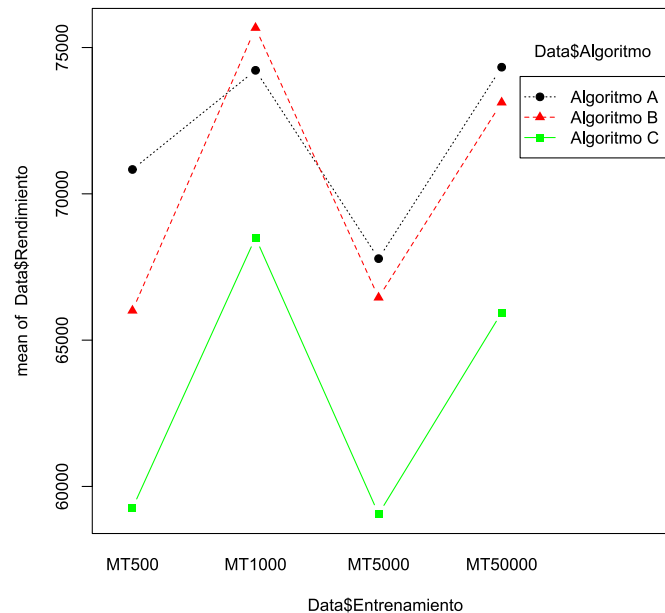
```

3) Gráfico simple de interacción

```

# Variable dependiente: Rendimiento
# Variables independientes: Algoritmo y Metodo de Entrenamiento.
interaction.plot(x.factor = Data$Entrenamiento,
  trace.factor = Data$Algoritmo,
  response = Data$Rendimiento,
  fun = mean,
  type = "b",
  col = c("black", "red", "green"),
  pch = c(19, 17, 15),
  fixed = TRUE,
  leg.bty = "o")

```

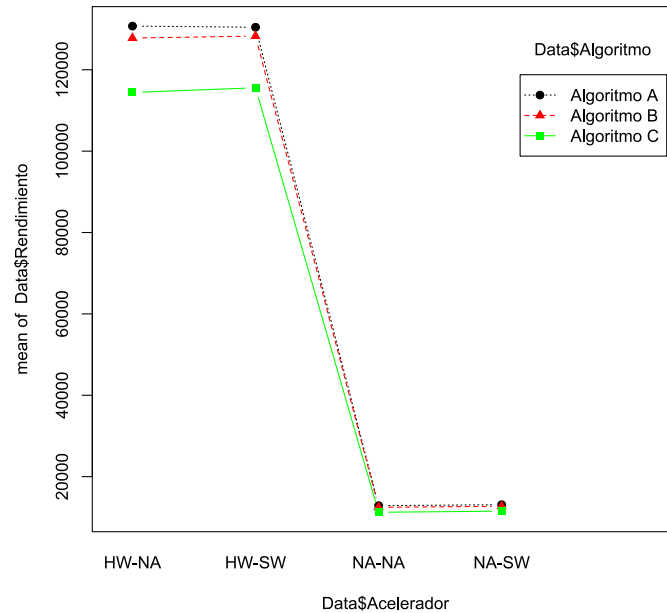


4) Cambio al gráfico para agregar el acelerador

```

interaction.plot(x.factor = Data$Acelerador,
  trace.factor = Data$Algoritmo,
  response = Data$Rendimiento,
  fun = mean,
  type = "b",
  col = c("black", "red", "green"),
  pch = c(19, 17, 15),
  fixed = TRUE,
  leg.bty = "o")

```



5) Modelo lineal y anova

```
model <- lm(Rendimiento ~ Entrenamiento * Algoritmo * Acelerador, data = Data)
library(car)
Anova(model, type = "II")
```

Anova Table (Type II tests)

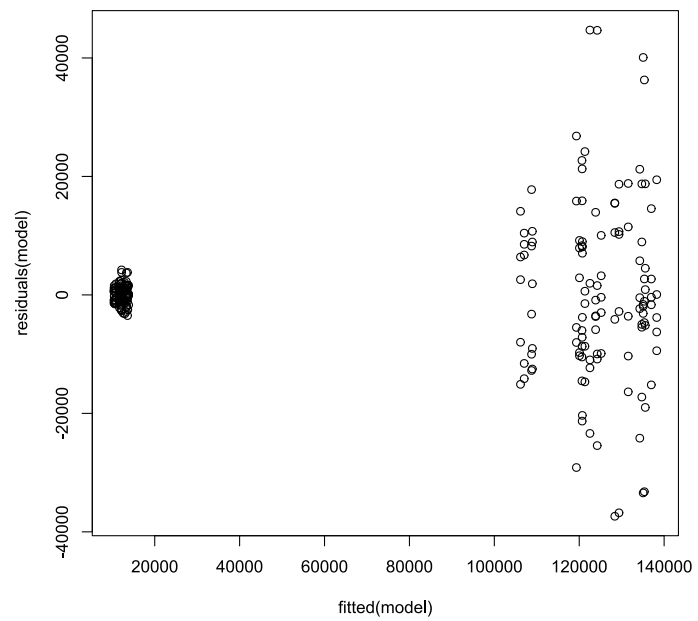
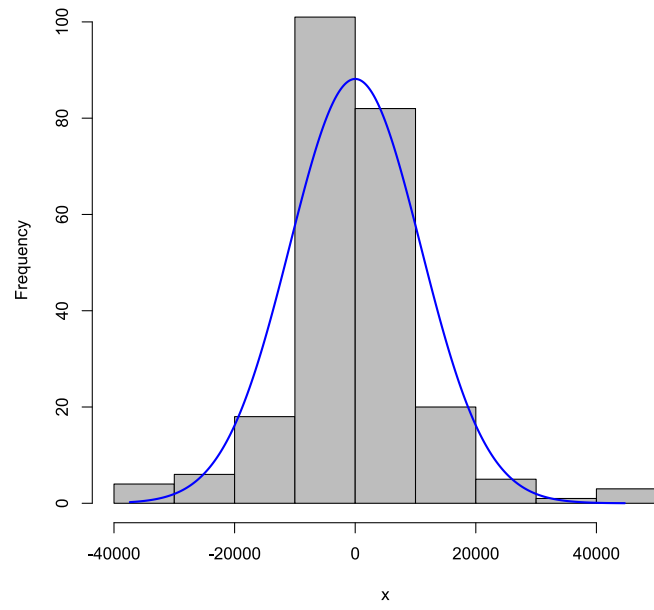
Response: Rendimiento

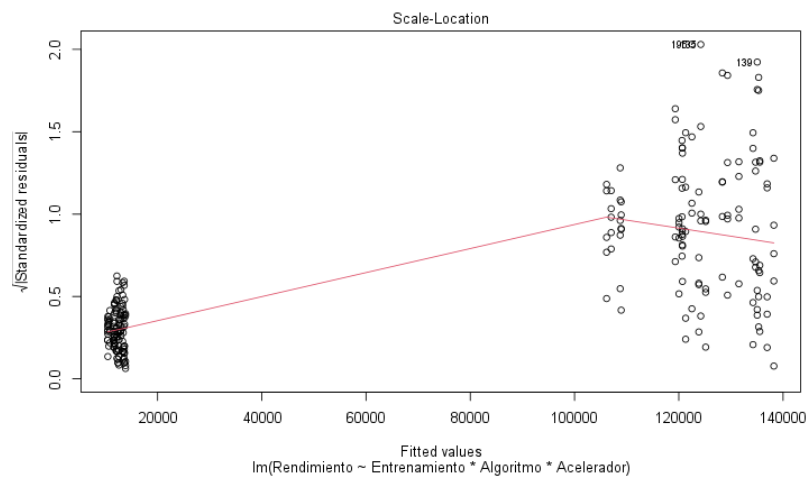
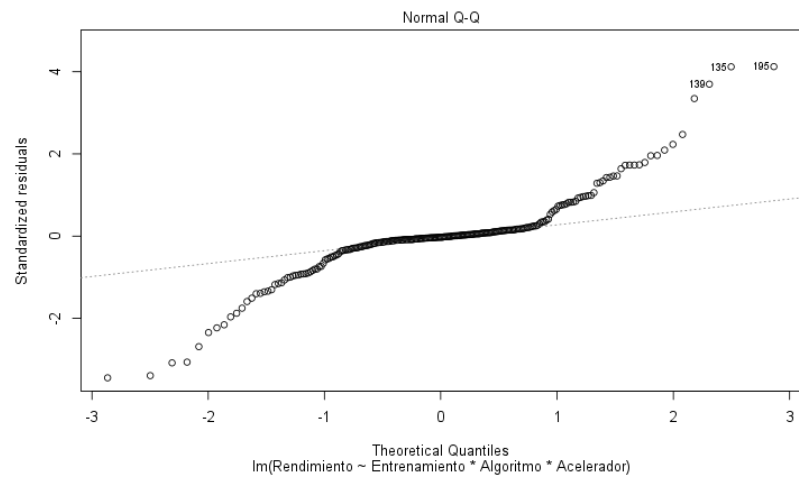
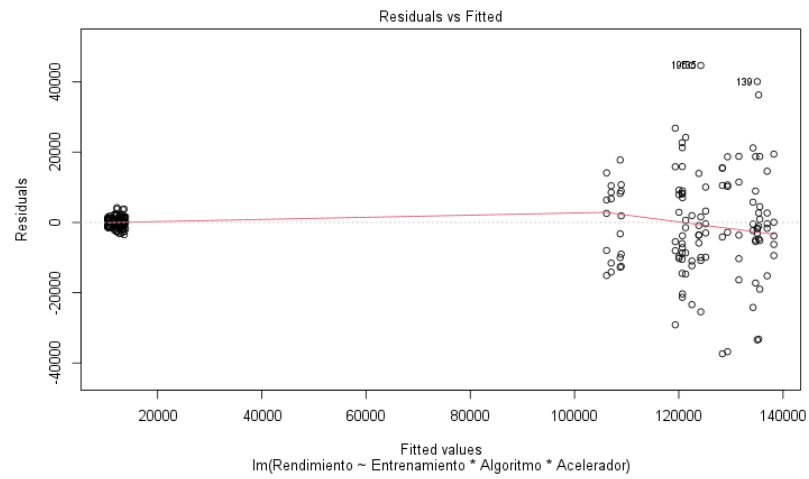
	Sum Sq	Df	F value	Pr(>F)
Entrenamiento	3.1056e+09	3	7.0491	0.0001608 ***
Algoritmo	3.3845e+09	2	11.5231	1.879e-05 ***
Acelerador	7.5562e+11	3	1715.0914	< 2.2e-16 ***
Entrenamiento: Algoritmo	2.4891e+08	6	0.2825	0.9447359
Entrenamiento: Acelerador	2.0023e+09	9	1.5149	0.1449261
Algoritmo: Acelerador	2.2782e+09	6	2.5855	0.0197057 *
Entrenamiento: Algoritmo: Acelerador	2.0204e+08	18	0.0764	0.9999999
Residuals	2.8196e+10	192		

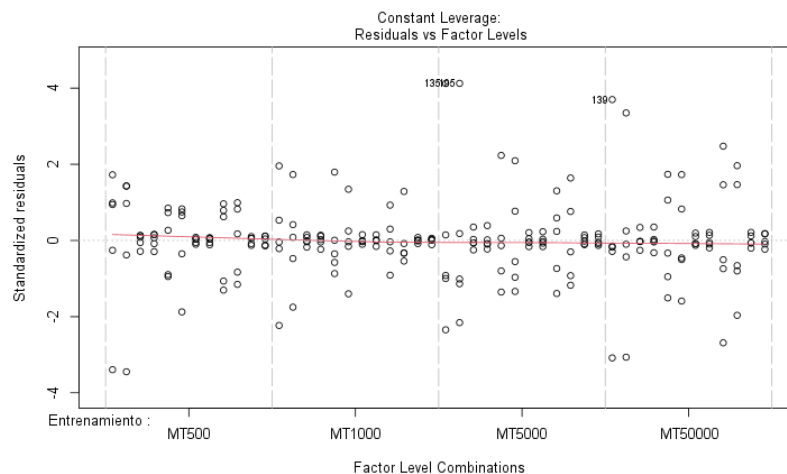
Signif. cod s: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

6) Evaluación de los supuestos

```
x <- residuals(model)
library(rcompanion)
plotNormalHistogram(x)
plot(fitted(model), residuals(model))
plot(model)
```





7) Transformación raíz cuadrada

Se procede a hacer transformación iniciando desde la forma menos agresiva a la mas agresiva, hasta cumplir los supuestos.

```
library(rcompanion)
T_sqrt <- sqrt(Data$Rendimiento) # Ingresar variable dependiente.
model <- lm(T_sqrt ~ Entrenamiento * Algoritmo * Acelerador, data = Data)

library(car)
Anova(model, type = "II")

x <- residuals(model)
library(rcompanion)
```

Anova Table (Type II tests)

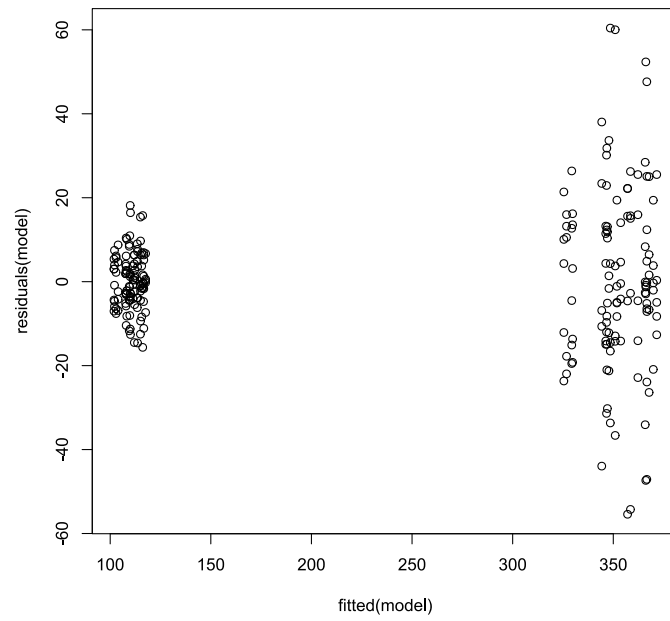
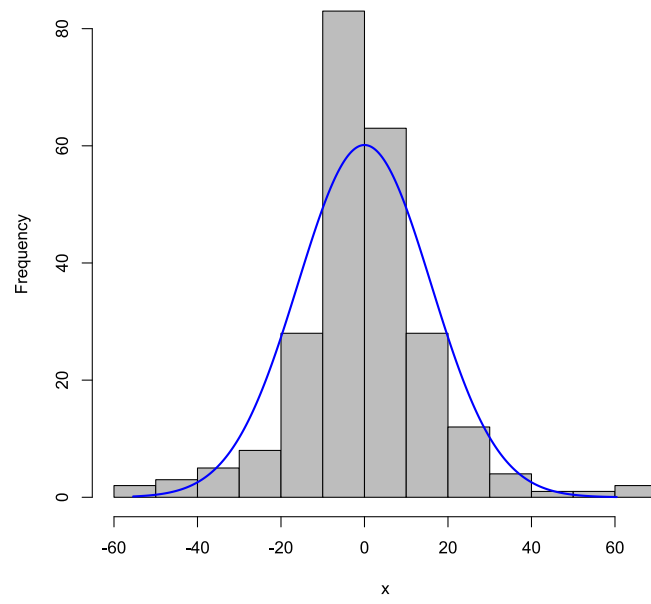
Response: T_sqrt

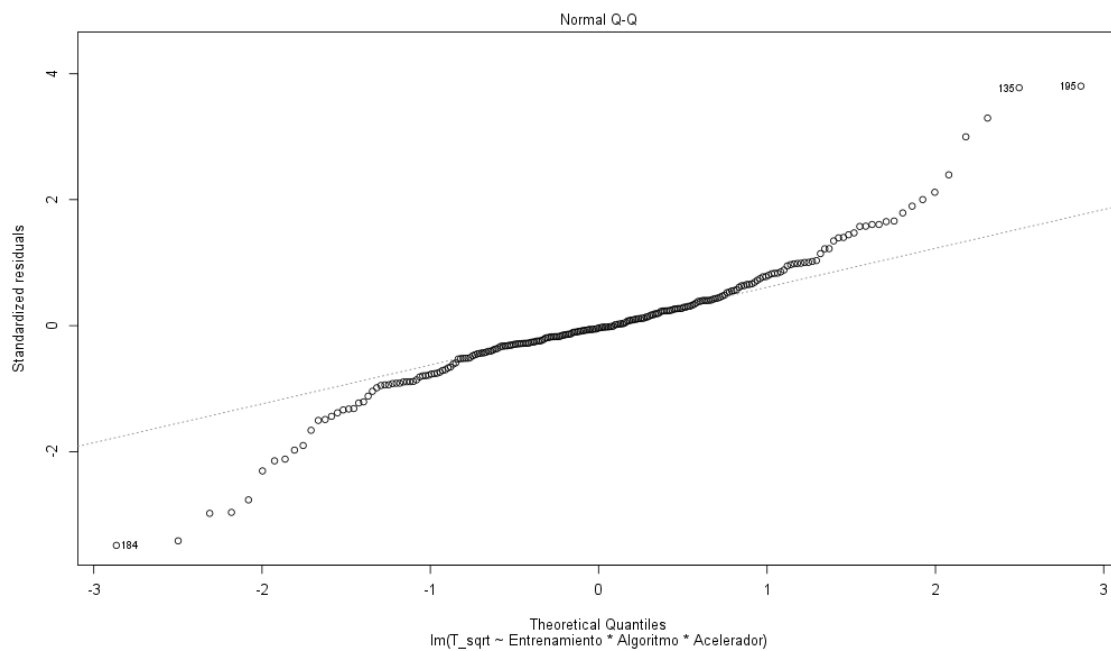
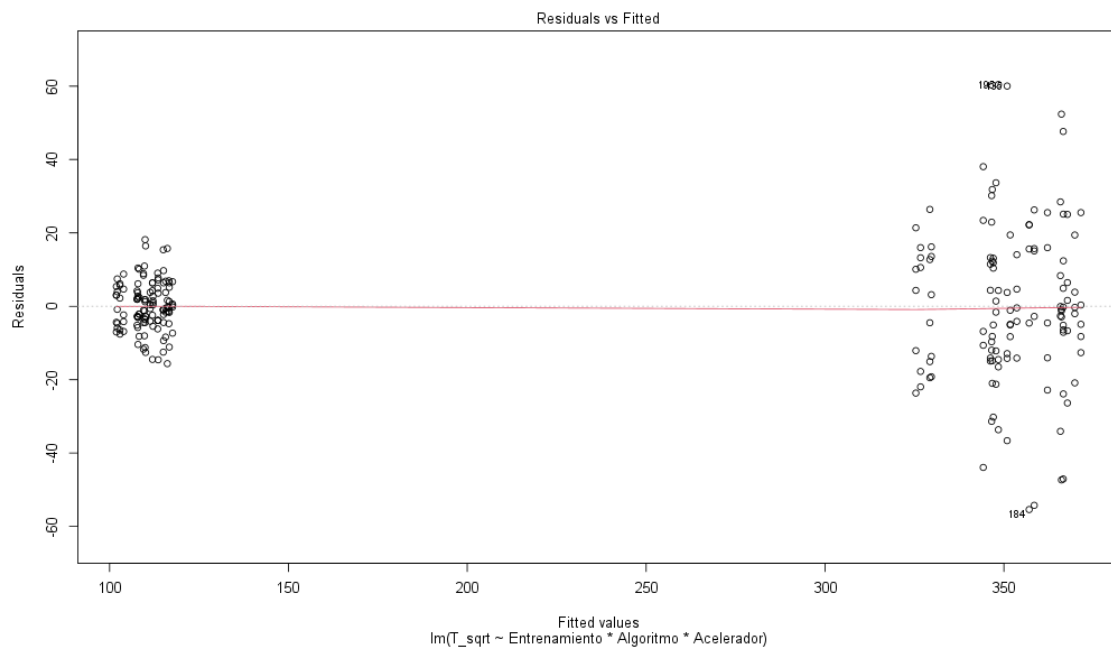
	Sum Sq	Df	F value	Pr(>F)
Entrenamiento	9545	3	10.0836	3.356e-06 ***
Algoritmo	9671	2	15.3253	6.684e-07 ***
Acelerador	3492852	3	3689.9304	< 2.2e-16 ***
Entrenamiento: Algoritmo	767	6	0.4052	0.8750
Entrenamiento: Acelerador	2223	9	0.7827	0.6326
Algoritmo: Acelerador	2626	6	1.3871	0.2216
Entrenamiento: Algoritmo: Acelerador	255	18	0.0448	1.0000
Residuals	60582	192		

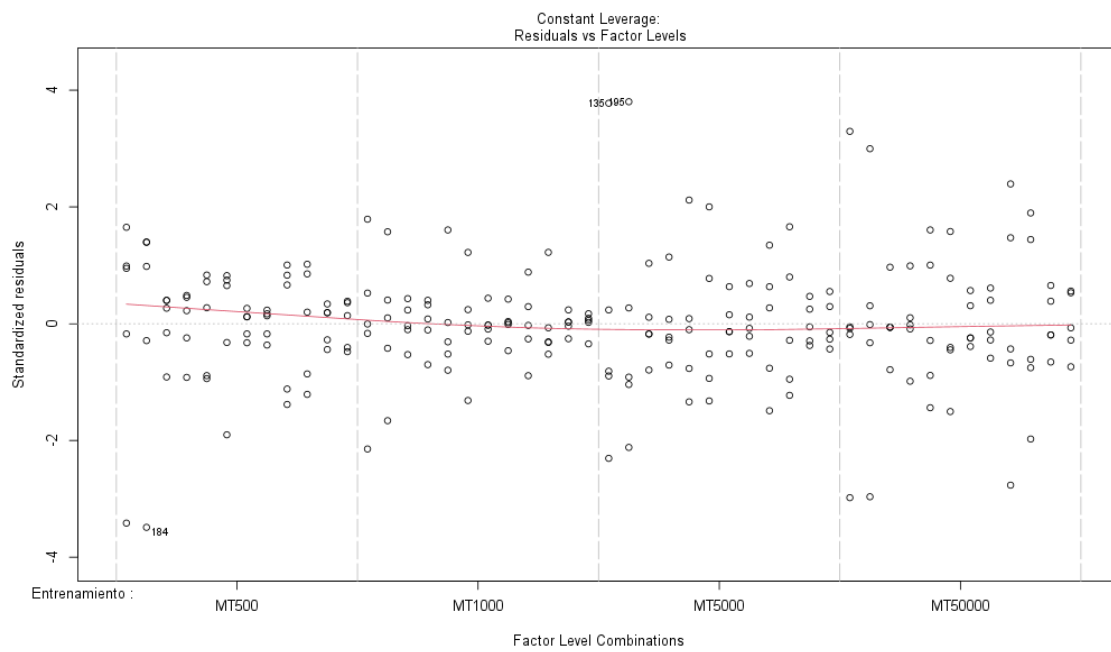
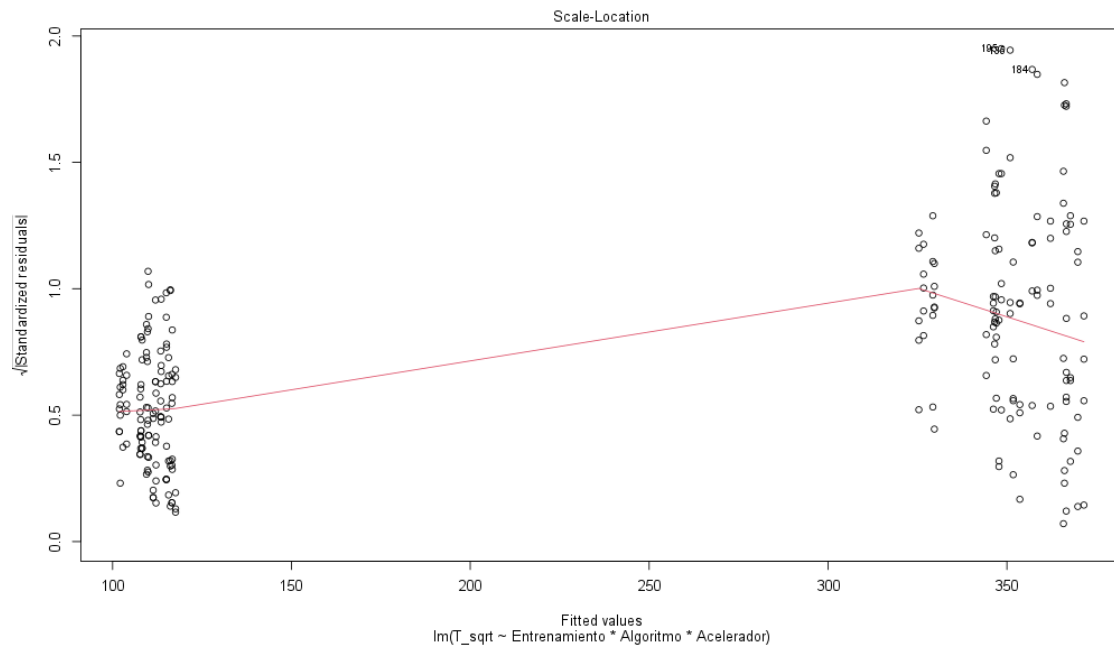
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

8) Evaluación de los supuestos

```
x <- residuals(model)
library(rcompanion)
plotNormalHistogram(x)
plot(fitted(model), residuals(model))
plot(model)
```







9) Transformación por raíz cúbica

```
library(rcompanion)
T_cub <- sign(Data$Rendimiento) * abs(Data$Rendimiento)^(1/3) # Ingresar variable dependiente.
model <- lm(T_cub ~ Entrenamiento * Algoritmo * Acelerador, data = Data)
```

```
library(car)
Anova(model, type = "II")
```

Anova Table (Type II tests)

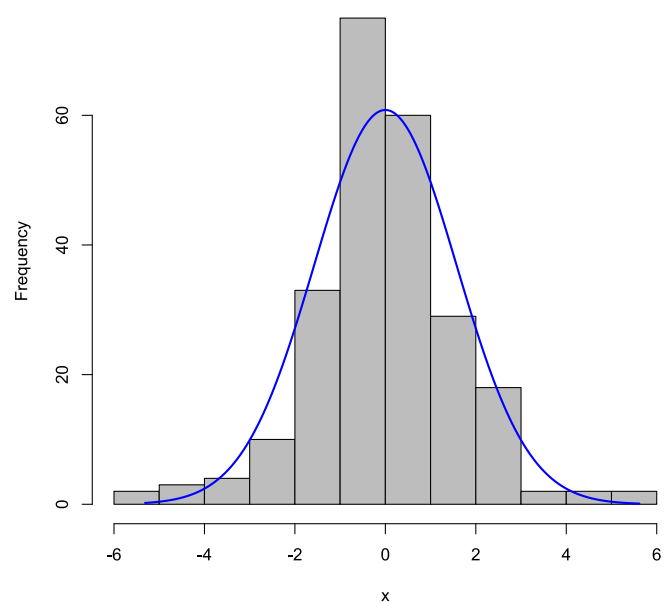
Response: T_cub

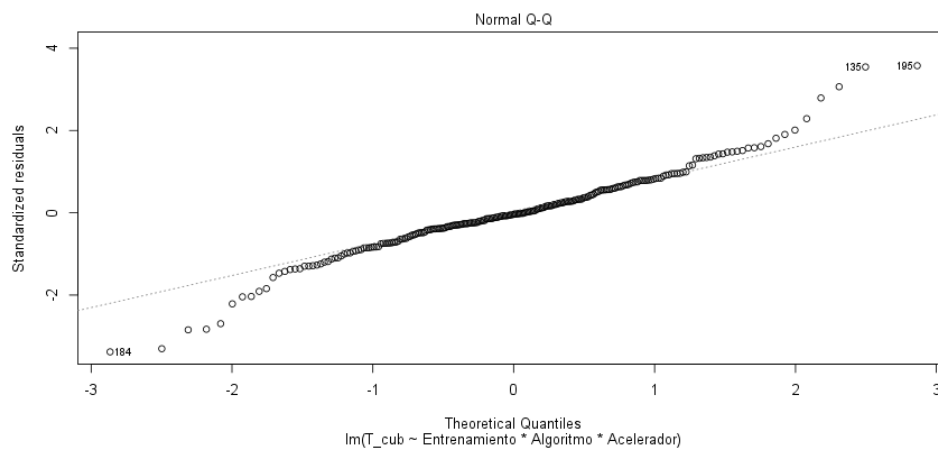
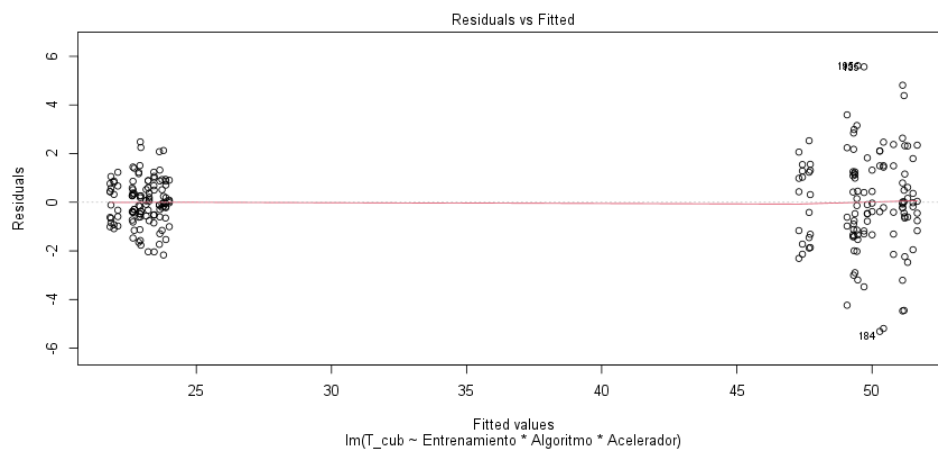
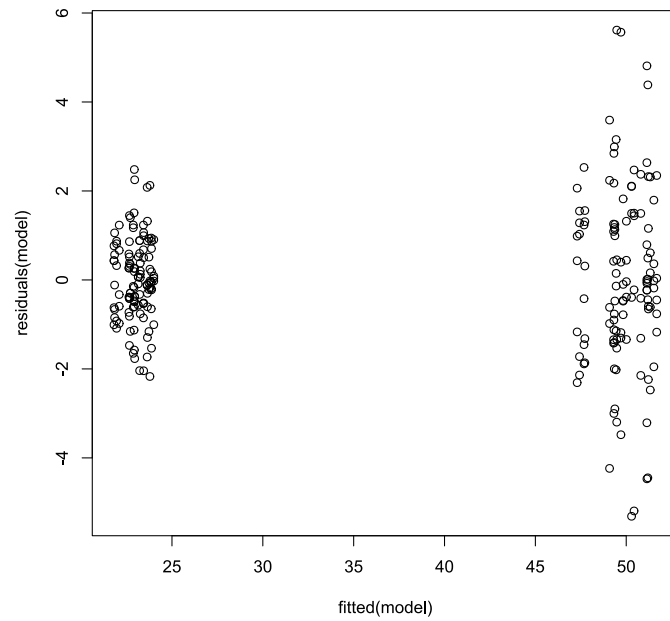
	Sum Sq	Df	F value	Pr(>F)	
Entrenamiento	108	3	11.7101	4.428e-07	***
Algoritmo	107	2	17.2949	1.241e-07	***
Acelerador	43031	3	4649.8989	< 2.2e-16	***
Entrenamiento: Algoritmo	9	6	0.4766	0.8252	
Entrenamiento: Acelerador	11	9	0.4078	0.9300	
Algoritmo: Acelerador	14	6	0.7802	0.5864	
Entrenamiento: Algoritmo: Acelerador	2	18	0.0303	1.0000	
Residuals	592	192			

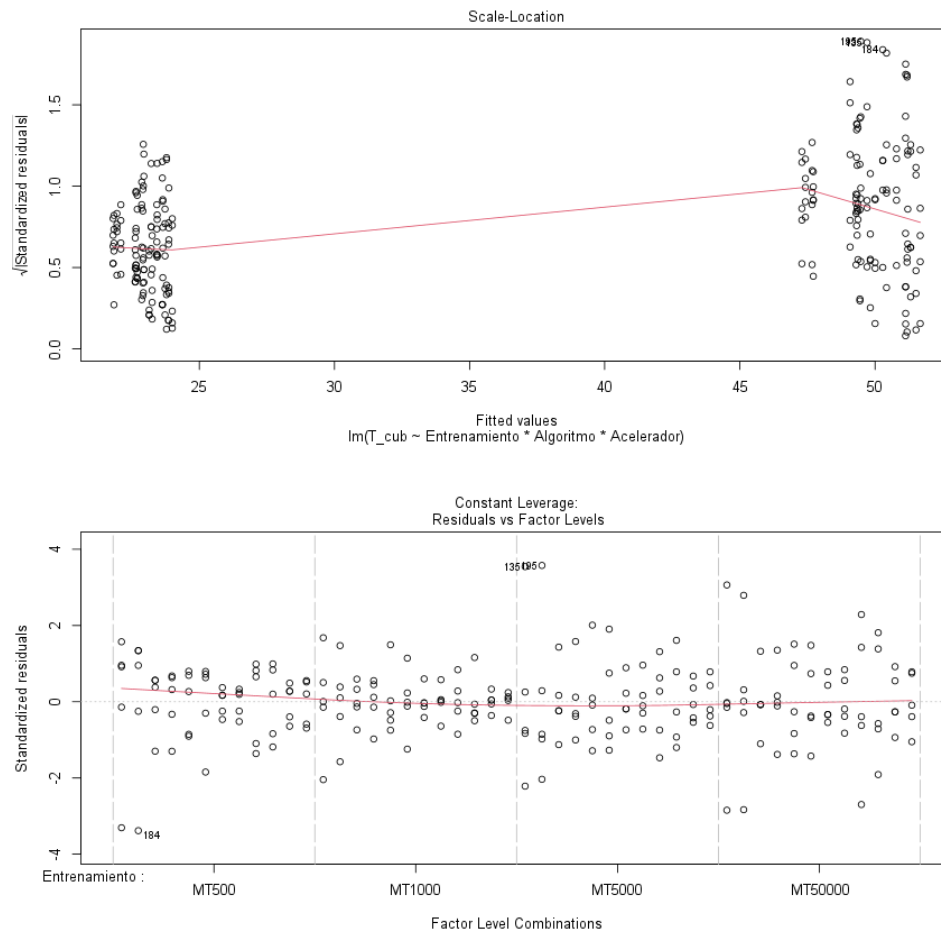
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

10) Evaluación de los supuestos

```
x <- residuals(model)
library(rcompanion)
plotNormalHistogram(x)
plot(fitted(model), residuals(model))
plot(model)
```







11) Transformación por logaritmo

```
library(rcompanion)
T_log <- log(Data$Rendimiento)

model <- lm(T_log ~ Entrenamiento * Algoritmo * Acelerador, data = Data)
library(car)
Anova(model, type = "II")
```

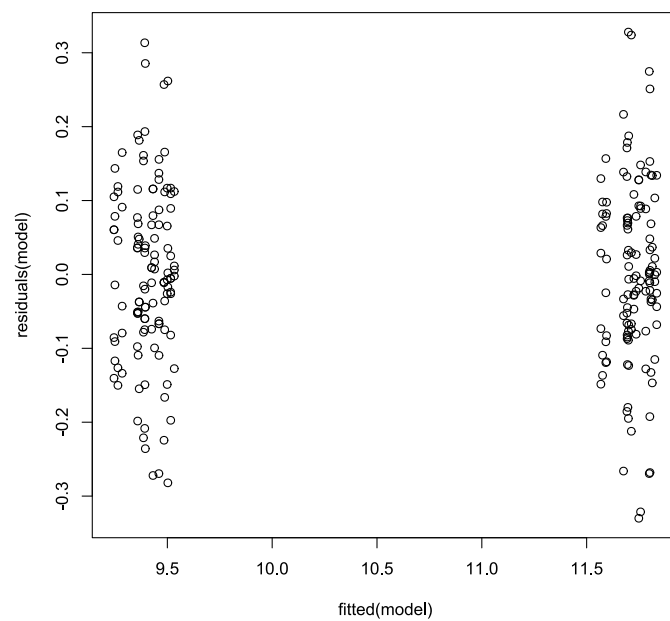
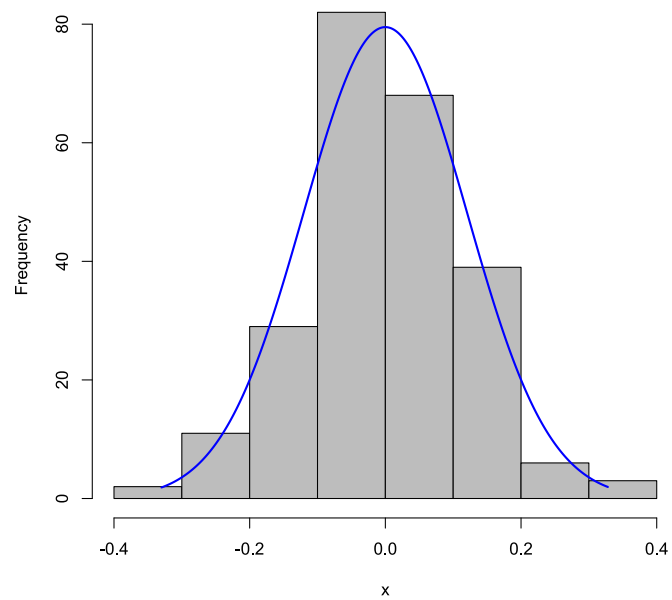
Anova Table (Type II tests)

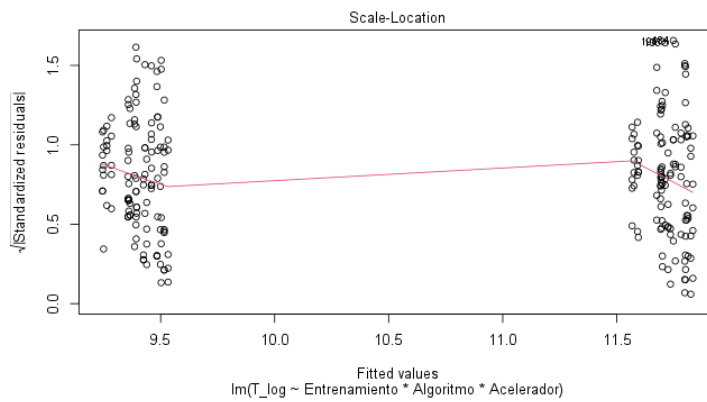
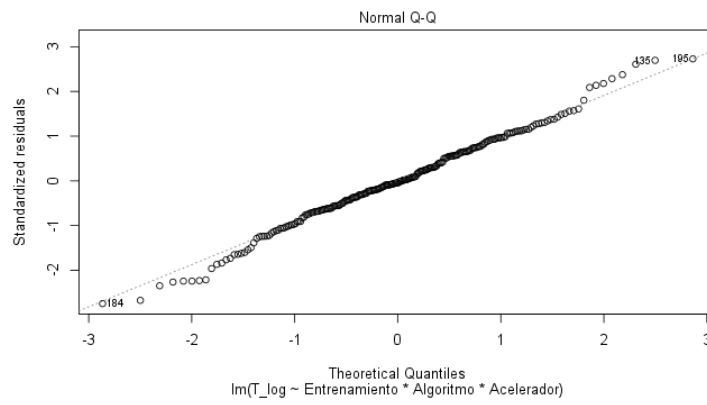
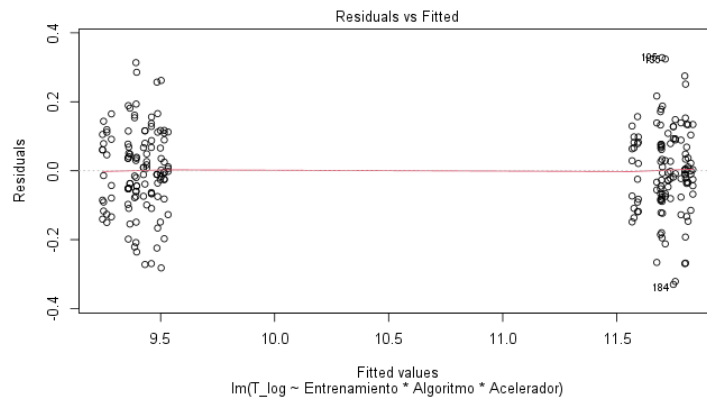
Response: T_log	Sum Sq	Df	F value	Pr(>F)
Entrenamiento	0.78	3	14.4599	1.566e-08 ***
Algoritmo	0.73	2	20.1075	1.179e-08 ***
Acelerador	321.12	3	5928.9761	< 2.2e-16 ***
Entrenamiento: Algoritmo	0.07	6	0.6144	0.7187
Entrenamiento: Acelerador	0.01	9	0.0355	1.0000
Algoritmo: Acelerador	0.00	6	0.0291	0.9999
Entrenamiento: Algoritmo: Acelerador	0.01	18	0.0216	1.0000
Residuals	3.47	192		

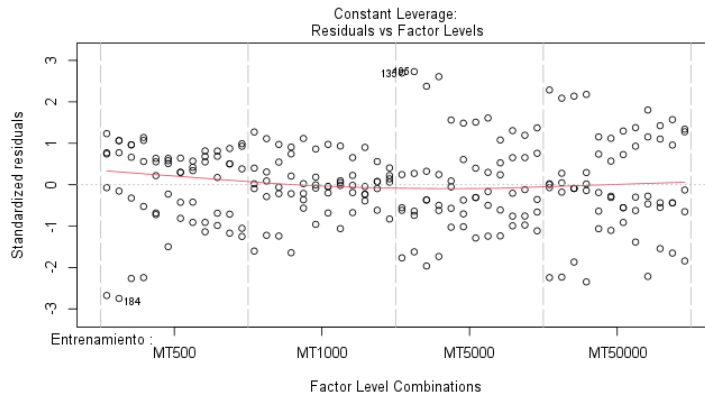
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

12) Evaluación de los supuestos

```
x <- residuals(model)
library(rcompanion)
plotNormalHistogram(x)
plot(fitted(model), residuals(model))
plot(model)
```







13) Prueba Levene

```
leveneTest(T_log ~ Entrenamiento * Algoritmo * Acelerador, data = Data)
```

Levene's Test for Homogeneity of Variance (center = median)

group	Df	F value	Pr(>F)
47	0.4057	0.9998	
192			

14) Análisis post-hoc por algoritmo

```
library(lsmmeans)
marginal <- lsmeans(model, pairwise ~ Algoritmo, adjust="tukey")
```

```
library(multcomp)
CLD <- cld(marginal, alpha = 0.05, Letters = letters, adjunst = "tukey")
CLD
```

Algoritmo	lsmean	SE	df	lower.CL	upper.CL	.group
Algoritmo C	10.49	0.01502	192	10.46	10.52	a
Algoritmo B	10.59	0.01502	192	10.56	10.62	b
Algoritmo A	10.61	0.01502	192	10.58	10.64	b

Results are averaged over the levels of: Entrenamiento, Acelerador
 Confidence level used: 0.95
 P value adjustment: tukey method for comparing a family of 3 estimates
 significance level used: alpha = 0.05
 NOTE: If two or more means share the same grouping symbol,
 then we cannot show them to be different.
 But we also did not show them to be the same.

15) Análisis post-hoc por entrenamiento

```
library(lsmmeans)
marginal <- lsmeans(model, pairwise ~ Entrenamiento, adjust="tukey")
```

```
library(multcomp)
CLD <- cld(marginal, alpha = 0.05, Letters = letters, adjunst = "tukey")
CLD
```

Entrenamiento	lsmean	SE	df	lower.CL	upper.CL	.group
MT5000	10.50	0.01735	192	10.47	10.54	a
MT500	10.52	0.01735	192	10.48	10.55	a
MT50000	10.60	0.01735	192	10.57	10.64	b
MT1000	10.64	0.01735	192	10.60	10.67	b

Results are averaged over the levels of: Algoritmo, Acelerador
 Confidence level used: **0.95**
 P value adjustment: tukey method for comparing a family of **4** estimates
 significance level used: alpha = **0.05**
 NOTE: If two or more means share the same grouping symbol,
 then we cannot show them to be different.
 But we also did not show them to be the same.

16) Análisis post-hoc para el acelerador 2^k

```
library(lsmmeans)
marginal <- lsmeans(model, pairwise ~ Acelerador, adjust="tukey")

library(multcomp)
CLD <- cld(marginal, alpha = 0.05, Letters = letters, adjunst = "tukey")
CLD
```

Acelerador	lsmean	SE	df	lower.CL	upper.CL	.group
NA-NA	9.399	0.01735	192	9.365	9.433	a
NA-SW	9.418	0.01735	192	9.384	9.452	a
HW-NA	11.720	0.01735	192	11.685	11.754	b
HW-SW	11.724	0.01735	192	11.690	11.758	b

Results are averaged over the levels of: Entrenamiento, Algoritmo
 Confidence level used: **0.95**
 P value adjustment: tukey method for comparing a family of **4** estimates
 significance level used: alpha = **0.05**
 NOTE: If two or more means share the same grouping symbol,
 then we cannot show them to be different.
 But we also did not show them to be the same.

17) Gráficos finales

```
library(FSA)
Sum <- Summarize(T_log ~ Entrenamiento + Algoritmo, data = Data, digits = 3)

# Se agrega el se
Sum$se <- Sum$sd / sqrt(Sum$n)
Sum$se <- signif(Sum$se, digits = 3)
Sum

### Ordenamos

Sum$Entrenamiento <- factor(Sum$Entrenamiento, levels = unique(Sum$Entrenamiento))

### Graficamos

library(ggplot2)
pd <- position_dodge(.2)

ggplot(Sum, aes(x = Entrenamiento,
  y = mean, color = Algoritmo)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .2, size = 0.7, position = pd) +
  geom_point(shape = 15, size = 4, position = pd) +
  theme_bw() +
  theme(axis.title = element_text(face = "bold")) +
  scale_colour_manual(values = c("black", "red", "green")) +
  ylab("Logaritmo de rendimiento")

# Para acelerador

Sum <- Summarize(T_log ~ Acelerador + Algoritmo, data = Data, digits = 3)

# Se agrega el se
Sum$se <- Sum$sd / sqrt(Sum$n)
Sum$se <- signif(Sum$se, digits = 3)
Sum

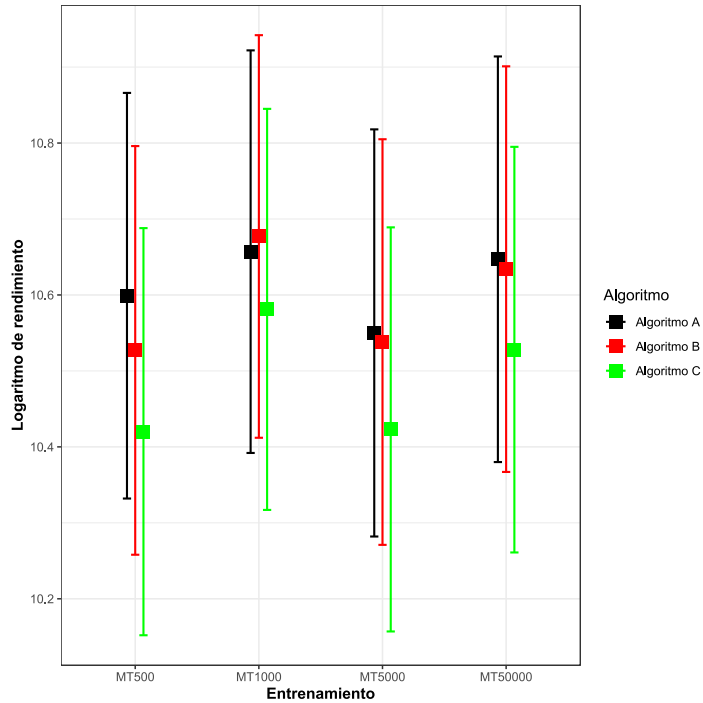
### Ordenamos | Correccion: se cambia entrenamiento por acelerador
Sum$Acelerador <- factor(Sum$Acelerador, levels = unique(Sum$Acelerador))
```

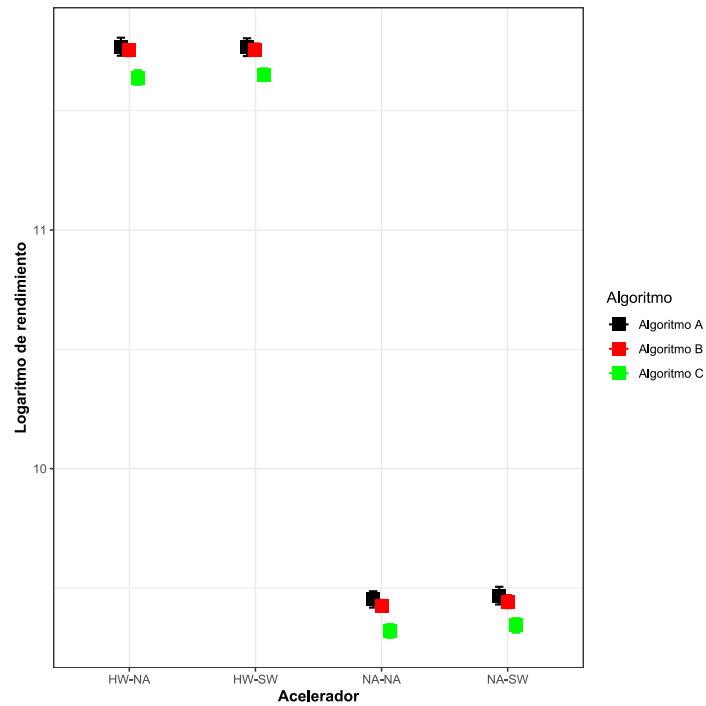
Graficamos

```
ggplot(Sum, aes(x = Acelerador ,
                y = mean, color = Algoritmo)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .2, size = 0.7, position = pd) +
  geom_point(shape = 15, size = 4, position = pd) +
  theme_bw() +
  theme(axis.title = element_text(face = "bold")) +
  scale_colour_manual(values = c("black", "red", "green")) +
  ylab("Logaritmo de rendimiento")
```

Entrenamiento	Algoritmo	n	mean	sd	min	Q1	median	Q3	max	se
1	MT500 Algoritmo A	20	10.599	1.195	9.159	9.523	10.508	11.843	11.905	0.267
2	MT1000 Algoritmo A	20	10.657	1.184	9.319	9.529	10.617	11.809	11.954	0.265
3	MT5000 Algoritmo A	20	10.550	1.200	9.159	9.351	10.603	11.641	12.037	0.268
4	MT50000 Algoritmo A	20	10.647	1.196	9.220	9.482	10.647	11.800	12.073	0.267
5	MT500 Algoritmo B	20	10.527	1.201	9.251	9.393	10.476	11.729	11.773	0.269
6	MT1000 Algoritmo B	20	10.677	1.185	9.406	9.526	10.678	11.818	11.968	0.265
7	MT5000 Algoritmo B	20	10.538	1.195	9.211	9.363	10.579	11.661	11.888	0.267
8	MT50000 Algoritmo B	20	10.634	1.195	9.321	9.442	10.654	11.771	11.941	0.267
9	MT500 Algoritmo C	20	10.420	1.200	9.105	9.306	10.411	11.622	11.693	0.268
10	MT1000 Algoritmo C	20	10.581	1.180	9.339	9.443	10.574	11.715	11.834	0.264
11	MT5000 Algoritmo C	20	10.423	1.191	9.133	9.240	10.434	11.574	11.748	0.266
12	MT50000 Algoritmo C	20	10.528	1.195	9.160	9.355	10.478	11.640	11.892	0.267

Acelerador	Algoritmo	n	mean	sd	min	Q1	median	Q3	max	se
1	HW-NA Algoritmo A	20	11.768	0.169	11.436	11.644	11.795	11.850	12.073	0.0378
2	HW-SW Algoritmo A	20	11.766	0.167	11.419	11.655	11.794	11.856	12.053	0.0373
3	NA-NA Algoritmo A	20	9.452	0.153	9.159	9.351	9.473	9.547	9.741	0.0342
4	NA-SW Algoritmo A	20	9.468	0.166	9.184	9.329	9.497	9.590	9.764	0.0371
5	HW-NA Algoritmo B	20	11.753	0.109	11.577	11.684	11.759	11.816	11.968	0.0244
6	HW-SW Algoritmo B	20	11.756	0.115	11.517	11.673	11.769	11.830	11.941	0.0257
7	NA-NA Algoritmo B	20	9.425	0.113	9.211	9.345	9.403	9.511	9.633	0.0253
8	NA-SW Algoritmo B	20	9.443	0.123	9.244	9.363	9.421	9.540	9.653	0.0275
9	HW-NA Algoritmo C	20	11.639	0.137	11.410	11.571	11.649	11.719	11.892	0.0306
10	HW-SW Algoritmo C	20	11.651	0.118	11.472	11.553	11.674	11.703	11.863	0.0264
11	NA-NA Algoritmo C	20	9.320	0.128	9.105	9.217	9.317	9.419	9.547	0.0286
12	NA-SW Algoritmo C	20	9.343	0.136	9.114	9.232	9.373	9.451	9.547	0.0304





18) Gráfico de promedios transformados

```
### Creamos un dato llamado sum con promedios y se
library(FSA)
Sum <- Summarize(T_log ~ Algoritmo, data = Data, digits = 3)

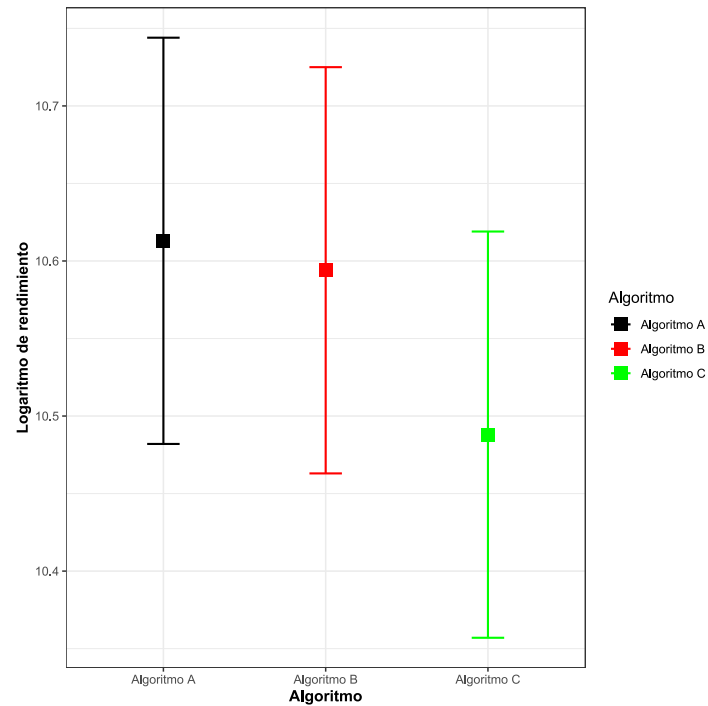
### Agregamos el se
Sum$se <- Sum$sd / sqrt(Sum$n)
Sum$se <- signif(Sum$se, digits = 3)
Sum

### Ordenamos | Correccion: se cambia entrenamiento por algoritmo
Sum$Algoritmo <- factor(Sum$Algoritmo, levels = unique(Sum$Algoritmo))

### Graficamos
library(ggplot2)
pd <- position_dodge(.2)

# Correccion: se cambia entrenamiento por algoritmo
ggplot(Sum, aes(x = Algoritmo,
  y = mean, color = Algoritmo)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .2, size = 0.7, position = pd) +
  geom_point(shape = 15, size = 4, position = pd) +
  theme_bw() +
  theme(axis.title = element_text(face = "bold")) +
  scale_colour_manual(values = c("black", "red", "green")) +
  ylab("Logaritmo de rendimiento")
```

	Algoritmo	n	mean	sd	min	Q1	median	Q3	max	se
1	Algoritmo A	80	10.613	1.172	9.159	9.482	10.591	11.792	12.073	0.131
2	Algoritmo B	80	10.594	1.173	9.211	9.413	10.585	11.766	11.968	0.131
3	Algoritmo C	80	10.488	1.171	9.105	9.348	10.478	11.656	11.892	0.131



19) Des-transformando promedios

```
library(FSA)
# Corrección, no se agrega Entrenamiento. El grafico final no seria el mismo en la presentacion.
Sum <- Summarize(T_log ~ Algoritmo, data = Data, digits = 3)

Sum$mean <- exp(Sum$mean)
Sum$sd <- exp(Sum$sd)

### Agregamos el se
Sum$se <- Sum$sd / sqrt(Sum$n)
Sum$se <- signif(Sum$se, digits = 3)
Sum

ggplot(Sum, aes(x = Algoritmo,
  y = mean, color = Algoritmo)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .2, size = 0.7, position = pd) +
  geom_point(shape = 15, size = 4, position = pd) +
  theme_bw() +
  theme(axis.title = element_text(face = "bold")) +
  scale_colour_manual(values = c("black", "red", "green")) +
  ylab("Logaritmo de rendimiento")
```

	Algoritmo	n	mean	sd	min	Q1	median	Q3	max	se
1	Algoritmo A	80	40660.00	3.228443	9.159	9.482	10.591	11.792	12.073	0.361
2	Algoritmo B	80	39894.75	3.231673	9.211	9.413	10.585	11.766	11.968	0.361
3	Algoritmo C	80	35882.32	3.225216	9.105	9.348	10.478	11.656	11.892	0.361

