

Biometría II

TP N° 3

Modelado de varianza

Problema 1. Trabajo de hormiga

Durante los últimos años, se ha desarrollado una creciente conciencia social sobre el efecto adverso que tienen los pesticidas sobre la salud ambiental y humana. Es por ello que en lugares sensibles donde se desaconseja liberar insecticidas (en forma de polvos, sprays o líquidos en el piso), se recomienda la implementación de métodos de control de insectos minimizando el uso de insecticidas y evitando su dispersión. Particularmente para el control de hormigas se recomienda la utilización de cebos alimentarios adicionados con un compuesto activo letal. Luego de varios estudios de control de hormigas hospitalarias observaron diferencias en la eficacia de los cebos: en muchos casos los cebos comerciales no resultan efectivos, pero si lo eran cebos azucarados diluidos con la misma dosis de tóxico. En base a estas observaciones, en el Laboratorio de Insectos Sociales se propusieron estudiar aspectos comportamentales de la ingestión de cebos que expliquen la menor efectividad de los cebos comerciales en el control de hormigas con respecto a cebos alternativos preparados en laboratorio.

Para cumplir este objetivo, se analizó comparativamente el comportamiento alimenticio de hormigas ante estos cebos alternativos y un cebo comercial. La experiencia fue realizada con colonias de hormigas *Camponotus mus* mantenidas en el laboratorio en nidos artificiales. En una arena experimental, se suministró una gota de cebo azucarado *ad libitum*.



Figure 1: Esquema del dispositivo experimental.

Se trabajó con 20 hormigas para cada uno de los tres tratamientos aplicados:

- (1) Cebo alternativo Sacarosa 30 % p/p + tóxico.
- (2) Cebo alternativo Sacarosa 68 % p/p + tóxico.
- (3) Cebo comercial con concentración de sacarosa similar a 68 % p/p + tóxico.

El experimento se llevó a cabo durante 5 días, midiendo 4 hormigas diferentes de cada tratamiento por día. Se las pesó antes de ingerir y luego de la ingesta, de dónde se obtuvo el peso ingerido. Sabiendo la concentración del tóxico, se obtuvo la masa ingerida de tóxico y, sabiendo las densidades de las soluciones, el volumen ingerido de solución. También se registró el tiempo de ingestión, tiempo de pausa y tiempo total (como la sumatoria de ambas variables). Estas dos mediciones resultan de interés ya que se plantea que soluciones más viscosas (aquellas con mayor concentración de azúcar) requieren mayores tiempos de ingesta, dado que la hormiga bombea la solución más lentamente y además genera que ésta haga pausas para poder llenar el buche con solución que luego repartirá en la colonia. Mayores tiempos en la fuente de alimentación provocaría un menor ingreso de tóxico en el nido por unidad de tiempo, haciendo menos eficiente a ese cebo para el control de la plaga. Teniendo en cuenta el volumen ingerido y el tiempo de ingestión resultaría de mucha utilidad estudiar la tasa de ingestión.

Parte A

- Realice un análisis exploratorio de los datos. Caracterice a los tratamientos aplicados en relación a la tendencia central y dispersión de cada variable estudiada. Defina el tipo de estudio, la unidad experimental y la cantidad de réplicas. ¿Qué concluye en relación al tiempo de ingesta para cada concentración de azúcar?

```
library(ggplot2)
source("/home/jose/Documents/funciones2.R")
horm = read.delim("/home/jose/Documents/materias/biome2/2018/tps/tp3/Hormigas.txt",
  header = T)
attach(horm)
# estructura de la base de datos, se observa el tipo de variable
str(horm)
```

```
## 'data.frame': 60 obs. of 8 variables:
## $ Tratamiento: Factor w/ 3 levels "CC","SAC30","SAC68": 2 2 2 2 2 2 2 2 2 2 ...
## $ Dia : int 1 1 1 1 2 2 2 2 3 3 ...
## $ MTox_ing : int 38 58 30 50 40 46 34 46 44 48 ...
## $ VSol_ing : num 1.65 2.52 1.3 2.17 1.74 2 1.48 2 1.91 2.08 ...
## $ T_ing : num 63.3 81.8 61.8 70 81.6 65.1 56.7 55.9 76.8 69.3 ...
## $ T_pau : num 0 0 51.8 5 9.5 0 4.5 0 10.9 0 ...
## $ T_tot : num 63.3 81.8 113.6 75 91.1 ...
## $ Nro_pau : int 0 0 1 1 1 0 2 0 1 0 ...
```

```
# Si queremos ver que tipo de objeto son las variables de
# nuestro data frame de manera individual
class(horm$Tratamiento)
```

```
## [1] "factor"
```

```
class(horm$T_ing)
```

```
## [1] "numeric"
```

```
class(horm$MTox_ing)
```

```
## [1] "integer"
```

Defina el tipo de estudio: *Experimental*.

Unidad experimental: *Hormiga?, Nido?*.

Cantidad de réplicas: *20*.

```
# Descriptiva analiticamente summary de toda nuestra base de
# datos
summary(horm)
```

```
## Tratamiento Dia MTox_ing VSol_ing T_ing
## CC :20 Min. :1 Min. :22.00 Min. :0.790 Min. : 36.20
## SAC30:20 1st Qu.:2 1st Qu.:34.00 1st Qu.:1.288 1st Qu.: 69.83
## SAC68:20 Median :3 Median :39.00 Median :1.580 Median :280.15
## Mean :3 Mean :39.97 Mean :1.567 Mean :285.41
## 3rd Qu.:4 3rd Qu.:46.00 3rd Qu.:1.837 3rd Qu.:438.27
## Max. :5 Max. :62.00 Max. :2.520 Max. :782.00
## T_pau T_tot Nro_pau
## Min. : 0.00 Min. : 42.80 Min. : 0.00
## 1st Qu.: 3.60 1st Qu.: 80.85 1st Qu.: 1.00
## Median : 30.55 Median :337.30 Median : 3.00
## Mean : 59.37 Mean :344.78 Mean : 5.45
## 3rd Qu.: 88.08 3rd Qu.:507.82 3rd Qu.: 8.25
## Max. :433.40 Max. :999.90 Max. :24.00
```

```
library(pastecs)
round(stat.desc(horm[, -1]), 2)
```

```
## Dia MTox_ing VSol_ing T_ing T_pau T_tot Nro_pau
```

```
## nbr.val      60.00    60.00    60.00    60.00    60.00    60.00    60.00
## nbr.null     0.00     0.00     0.00     0.00    14.00     0.00    14.00
## nbr.na       0.00     0.00     0.00     0.00     0.00     0.00     0.00
## min          1.00    22.00     0.79    36.20     0.00    42.80     0.00
## max          5.00    62.00     2.52   782.00   433.40   999.90    24.00
## range        4.00    40.00     1.73   745.80   433.40   957.10    24.00
## sum         180.00  2398.00    94.02 17124.40 3562.20 20686.60 327.00
## median       3.00    39.00     1.58   280.15    30.55   337.30     3.00
## mean         3.00    39.97     1.57   285.41    59.37   344.78     5.45
## SE.mean      0.18     1.17     0.05    25.86    10.22    31.82     0.83
## CI.mean.0.95 0.37     2.34     0.10    51.74    20.45    63.67     1.65
## var          2.03    81.83     0.16 40120.66 6265.59 60745.21 40.93
## std.dev       1.43     9.05     0.41   200.30    79.16   246.47     6.40
## coef.var      0.48     0.23     0.26     0.70     1.33     0.71     1.17
```

```
# Podemos armar subset de datos para cada nivel de la variable
# toxico (cc, sac68 y sac30) para pedir medidas descriptivas por
# separado
```

```
setCC <- subset(horm, Tratamiento == "CC")
setSAC68 <- subset(horm, Tratamiento == "SAC68")
setSAC30 <- subset(horm, Tratamiento == "SAC30")
round(stat.desc(setCC[, -1]), 2)
```

```
##          Dia MTox_ing VSol_ing      T_ing      T_pau      T_tot Nro_pau
## nbr.val    20.00    20.00    20.00    20.00    20.00    20.00    20.00
## nbr.null    0.00     0.00     0.00     0.00     0.00     0.00     0.00
## nbr.na      0.00     0.00     0.00     0.00     0.00     0.00     0.00
## min         1.00    22.00     0.79   231.30    26.00   271.30     3.00
## max         5.00    50.00     1.79   782.00   433.40   999.90    24.00
## range       4.00    28.00     1.00   550.70   407.40   728.60    21.00
## sum        60.00   668.00    23.91 8437.70 2437.70 10875.40 245.00
## median      3.00    33.00     1.19   424.55   106.70   501.35     9.00
## mean        3.00    33.40     1.20   421.88   121.89   543.77    12.25
## SE.mean     0.32     1.51     0.05    31.78    22.21    43.48     1.44
## CI.mean.0.95 0.68     3.16     0.11    66.52    46.49    91.01     3.01
## var         2.11    45.52     0.06 20200.26 9868.86 37816.17 41.46
## std.dev      1.45     6.75     0.24   142.13    99.34   194.46     6.44
## coef.var     0.48     0.20     0.20     0.34     0.82     0.36     0.53
```

```
round(stat.desc(setSAC68[, -1]), 2)
```

```
##          Dia MTox_ing VSol_ing      T_ing      T_pau      T_tot Nro_pau
## nbr.val    20.00    20.00    20.00    20.00    20.00    20.00    20.00
## nbr.null    0.00     0.00     0.00     0.00     4.00     0.00     4.00
## nbr.na      0.00     0.00     0.00     0.00     0.00     0.00     0.00
## min         1.00    28.00     1.06   136.90     0.00   179.10     0.00
## max         5.00    62.00     2.34   729.50   149.00   783.00    10.00
## range       4.00    34.00     1.28   592.60   149.00   603.90    10.00
## sum        60.00   886.00    33.47 7481.60 905.20 8386.80 65.00
## median      3.00    45.00     1.70   345.55    30.05   402.85     2.50
## mean        3.00    44.30     1.67   374.08    45.26   419.34     3.25
## SE.mean     0.32     2.07     0.08    33.90    10.26    34.56     0.65
## CI.mean.0.95 0.68     4.33     0.16    70.95    21.48    72.34     1.36
## var         2.11    85.59     0.12 22981.83 2106.74 23890.67 8.41
## std.dev      1.45     9.25     0.35   151.60    45.90   154.57     2.90
## coef.var     0.48     0.21     0.21     0.41     1.01     0.37     0.89
```

```
round(stat.desc(setSAC30[, -1]), 2)
```

```
##          Dia MTox_ing VSol_ing      T_ing      T_pau      T_tot Nro_pau
## nbr.val    20.00    20.00    20.00    20.00    20.00    20.00    20.00
```

```
## nbr.null      0.00      0.00      0.00      0.00 10.00      0.00 10.00
## nbr.na        0.00      0.00      0.00      0.00  0.00      0.00  0.00
## min           1.00     30.00      1.30     36.20   0.00     42.80   0.00
## max           5.00     58.00      2.52     81.80 110.90    147.10   4.00
## range         4.00     28.00      1.22     45.60 110.90    104.30   4.00
## sum           60.00    844.00     36.64 1205.10 219.30   1424.40  17.00
## median        3.00     42.00      1.82     59.25   0.60     64.20   0.50
## mean          3.00     42.20      1.83     60.25  10.96     71.22   0.85
## SE.mean       0.32      1.62      0.07      2.81   5.88      5.55   0.24
## CI.mean.0.95  0.68      3.39      0.15      5.89  12.30     11.62   0.51
## var           2.11     52.59      0.10    158.31 690.96    616.40   1.19
## std.dev       1.45      7.25      0.31     12.58  26.29     24.83   1.09
## coef.var      0.48      0.17      0.17      0.21   2.40      0.35   1.28
```

```
# una libreria muy utilizada para estas cosas es dplyr
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   first, last

## The following objects are masked from 'package:base':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
setCC1 <- horm %>%
  filter(Tratamiento == "CC")
# probemos a ver si hace lo mismo
setCC == setCC1
```

```
##   Tratamiento Dia MTox_ing VSol_ing T_ing T_pau T_tot Nro_pau
## 41      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 42      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 43      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 44      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 45      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 46      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 47      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 48      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 49      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 50      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 51      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 52      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 53      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 54      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 55      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 56      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 57      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 58      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 59      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
## 60      TRUE TRUE      TRUE      TRUE TRUE TRUE TRUE TRUE
```

```
# Si lo vemos graficamente
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
```

```
## method from
## +.gg ggplot2

##
## Attaching package: 'GGally'

## The following object is masked from 'package:emmeans':
##
## pigs

ggpairs(horm[, c(1, 3, 4, 5, 6, 7)], mapping = ggplot2::aes(color = horm$Tratamiento),
  diag = list(discrete = "barDiag", continuous = wrap("densityDiag",
    alpha = 0.5)), upper = list(combo = wrap("box_no_facet", alpha = 0.5),
    continuous = wrap("cor", size = 2, alignPercent = 0.8)), columnLabels = c("Tratamientos",
    "Masa ingerida", "Volumen ingerido", "Tiempo ingesta", "Tiempo pausas",
    "Tiempo total"))

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

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

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

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

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

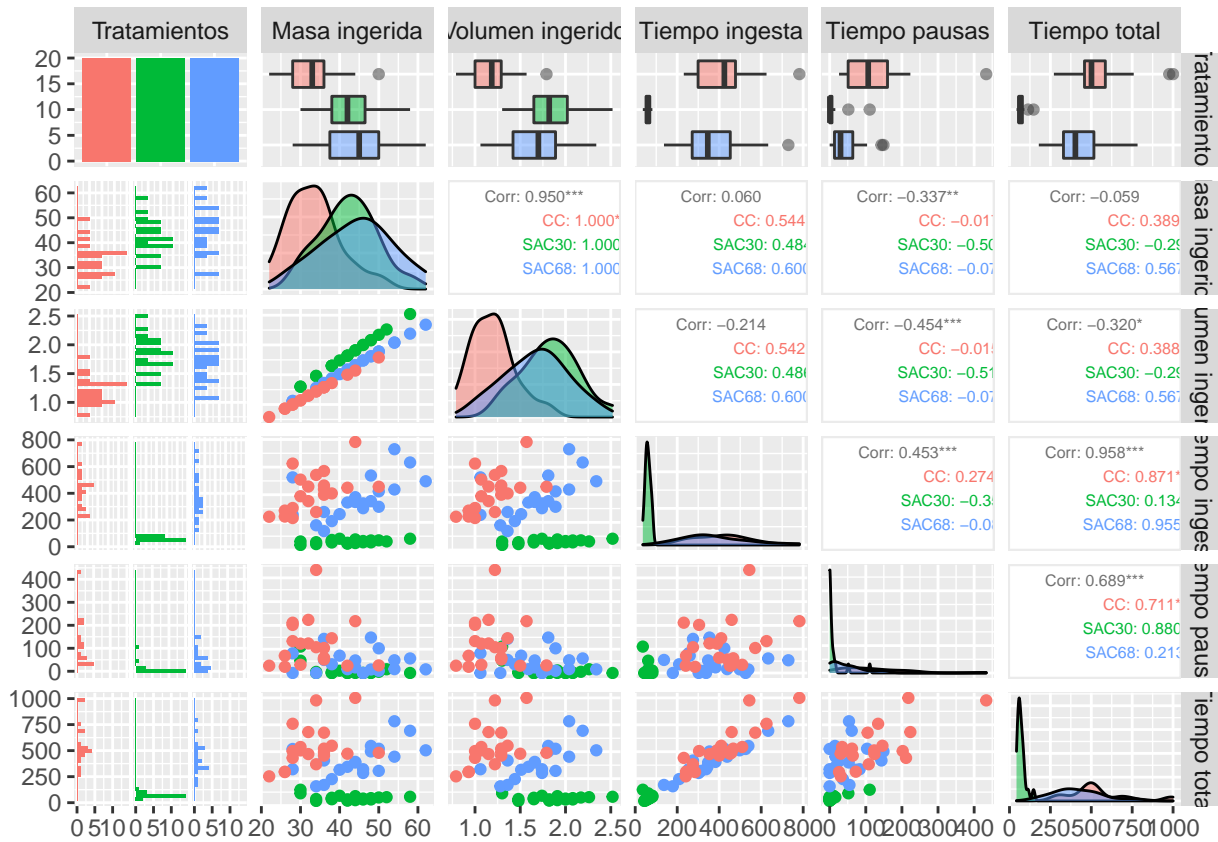
## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

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

## Warning in (function (data, mapping, ..., stars = TRUE, method = "pearson", :
## `alignPercent` is deprecated. Please use `align_percent` if alignment still
## needs to be adjusted

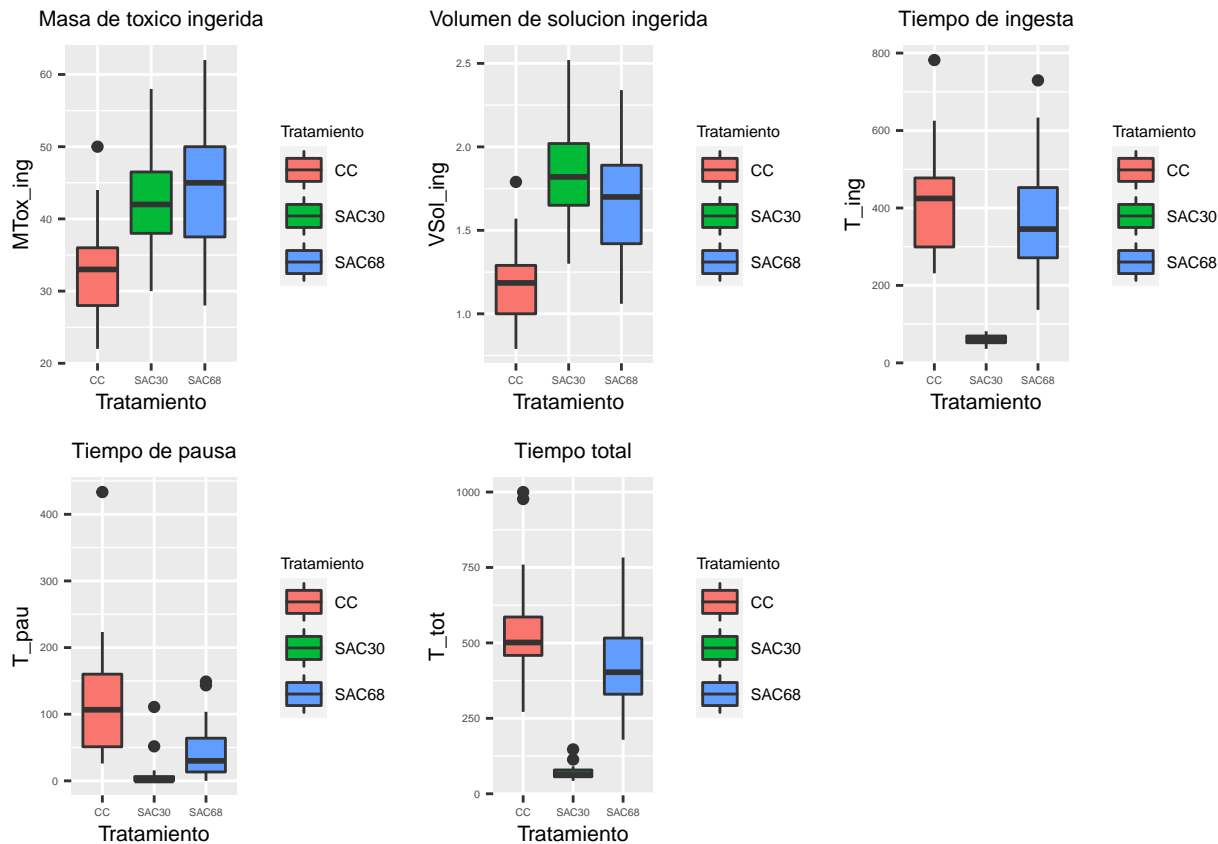
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Estudie la simetría de las variables analizadas.

```
library(gridExtra)
```

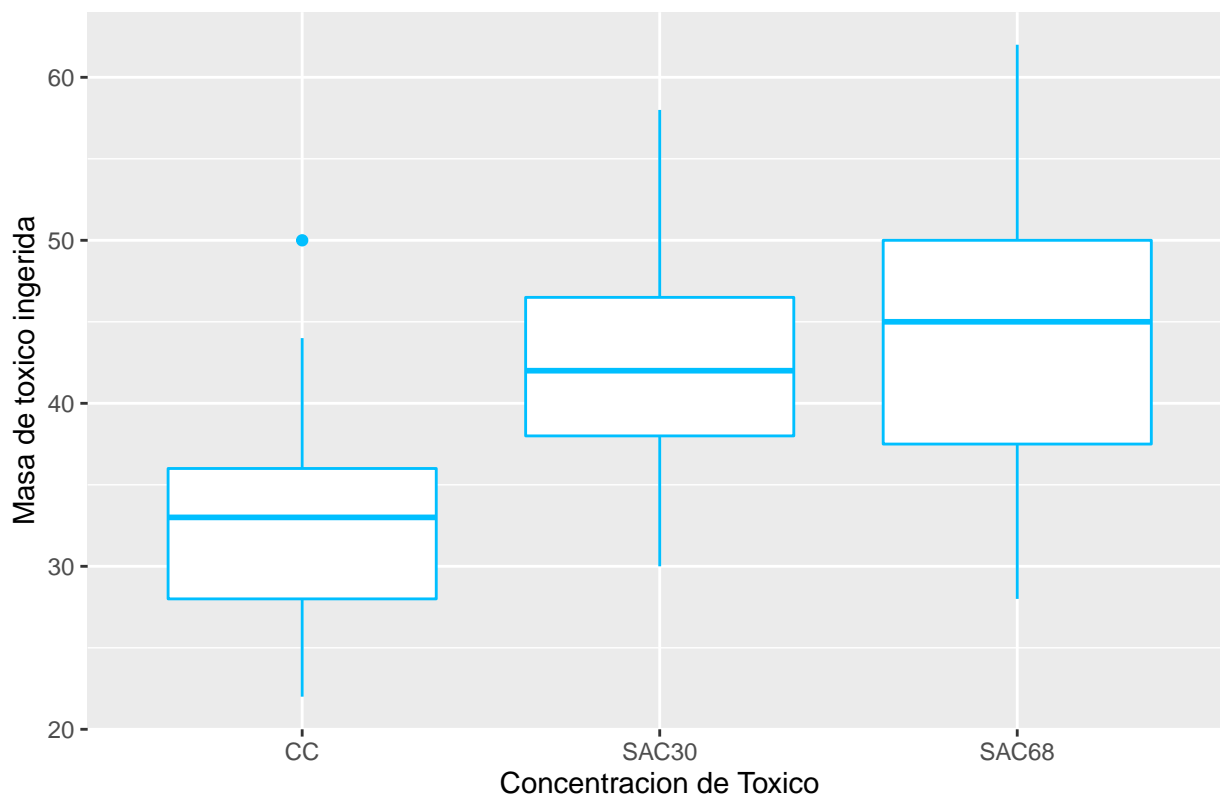
```
##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##      combine
grid.arrange(a, b, c, d, e, ncol = 3)
```



- ¿Con cuál de las soluciones se ingiere mayor cantidad de tóxico? Analícelo a partir de un modelo estadístico lineal. Analice los supuestos. Concluya realizando recomendaciones de uso para el control de hormigas.

```
# Para evaluar los supuestos primero tenemos que generar el
# modelo. Esto permite calcular los residuos del modelo.
modelo0 <- lm(MTox_ing ~ Tratamiento)
```

```
p <- ggplot(horm, aes(Tratamiento, MTox_ing)) + geom_boxplot(aes(),
  colour = "deepskyblue")
p + xlab("Concentracion de Toxico") + ylab("Masa de toxico ingerida") +
  ggtitle("")
```



```
# Supuestos Calculamos los residuos y los predichos
e <- residuals(modelo0) # residuos
re <- rstandard(modelo0) #residuos estandarizados
pre <- predict(modelo0) #predichos
res <- data.frame(horm$Tratamiento, horm$MTox_ing, pre, e, round(re,
2))
colnames(res) <- c("Tratamiento", "MTox_ing", "Predichos", "Residuos",
"residuos std")
head(res)
```

```
## Tratamiento MTox_ing Predichos Residuos residuos std
## 1 SAC30 38 42.2 -4.2 -0.55
## 2 SAC30 58 42.2 15.8 2.07
## 3 SAC30 30 42.2 -12.2 -1.60
## 4 SAC30 50 42.2 7.8 1.02
## 5 SAC30 40 42.2 -2.2 -0.29
## 6 SAC30 46 42.2 3.8 0.50
```

Los residuos estandarizados se calculan como:

$$RE = e_i / \sqrt{S_e^2}$$

A su vez,

$$\sqrt{S_e^2} = \sum_{i=1}^n e_i^2 / (n - 2)$$

Hagamos el primero:

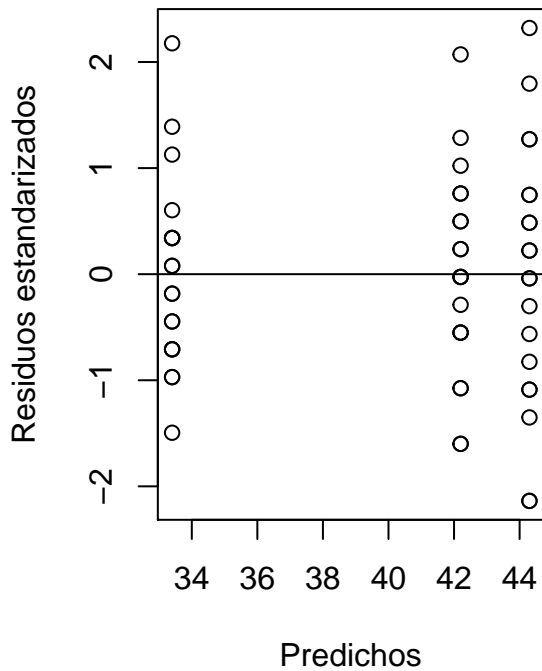
```
residuo1 = 38 - 42.2
sumaRes2 = sum(e^2)
(resStandard1 = residuo1/sqrt(sumaRes2/nrow(horm) - 3))
```

```
## [1] -0.5654547
```

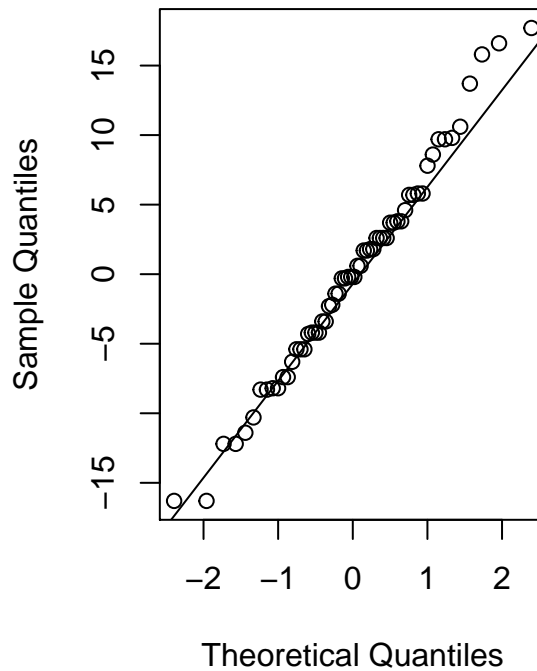


```
# Supuestos modelo
par(mfrow = c(1, 2))
plot(pre, re, xlab = "Predichos", ylab = "Residuos estandarizados",
     main = "Grafico de dispersion de RE vs PRED")
abline(0, 0)
qqnorm(e)
qqline(e)
```

Grafico de dispersion de RE vs PR



Normal Q-Q Plot



```
# prueba analitica para la homogeneidad de varianzas
car::leveneTest(horm$MTox_ing, horm$Tratamiento)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 2  1.1159 0.3347
##      57
```

Que hipotesis nula pone a prueba este supuesto?

```
# prueba analitica
shapiro.test(modelo0$residuals)
```

```
##
## Shapiro-Wilk normality test
##
## data:  modelo0$residuals
## W = 0.98786, p-value = 0.8148
```

```
# o
shapiro.test(e)
```

```
##
## Shapiro-Wilk normality test
##
```

```
## data: e
## W = 0.98786, p-value = 0.8148
# Que hipotesis nula pone a prueba este supuesto?

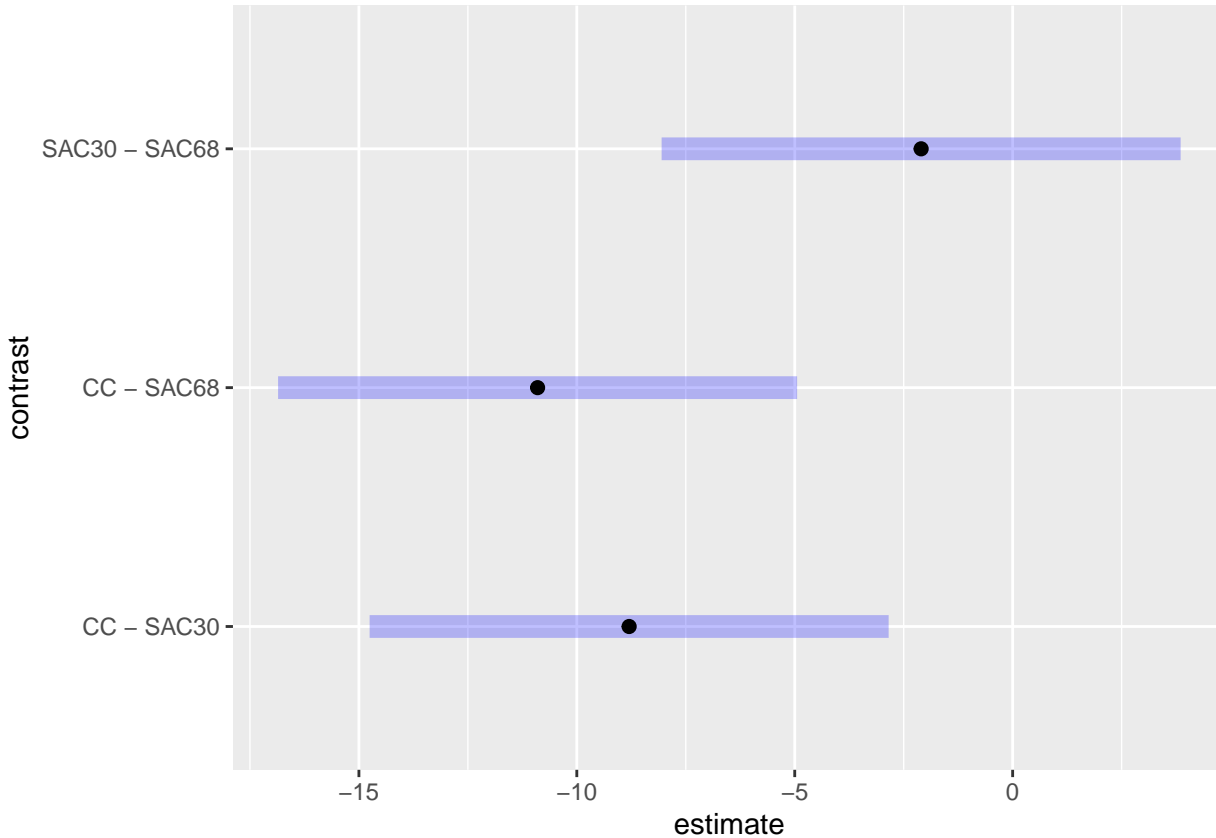
## Resultados generales. Ver resultados en formato regresion y
## en formato anova.
summary(modelo0)

##
## Call:
## lm(formula = MTox_ing ~ Tratamiento)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.3    -5.4    -0.2     4.0    17.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.400      1.750  19.089 < 2e-16 ***
## TratamientoSAC30   8.800      2.475   3.556 0.000765 ***
## TratamientoSAC68  10.900      2.475   4.405 4.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.825 on 57 degrees of freedom
## Multiple R-squared:  0.2771, Adjusted R-squared:  0.2517
## F-statistic: 10.92 on 2 and 57 DF, p-value: 9.639e-05
anova(modelo0)

## Analysis of Variance Table
##
## Response: MTox_ing
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Tratamiento  2 1337.7   668.87  10.924 9.639e-05 ***
## Residuals    57 3490.2    61.23
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# comparaciones multiples
library(emmeans)
TukeyEmmeans <- emmeans(modelo0, pairwise ~ Tratamiento)
TukeyEmmeans

## $emmeans
##      Tratamiento emmean    SE df lower.CL upper.CL t.ratio p.value
## CC              33.4 1.75 57     29.9     36.9  19.089 <.0001
## SAC30            42.2 1.75 57     38.7     45.7  24.118 <.0001
## SAC68            44.3 1.75 57     40.8     47.8  25.318 <.0001
##
## Confidence level used: 0.95
##
## $contrasts
##      contrast      estimate    SE df lower.CL upper.CL t.ratio p.value
## CC - SAC30      -8.8 2.47 57    -14.75    -2.85  -3.556 0.0022
## CC - SAC68     -10.9 2.47 57    -16.85    -4.95  -4.405 0.0001
## SAC30 - SAC68   -2.1 2.47 57     -8.05     3.85  -0.849 0.6745
##
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 3 estimates
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
plot(TukeyEmmeans$contrasts)
```



```
hCI <- round(abs(as.data.frame(TukeyEmmeans$contrasts)[[5]][1]), 2)
lCI <- round(abs(as.data.frame(TukeyEmmeans$contrasts)[[6]][1]), 2)
```

Se concluye que el tratamiento SAC30 es el mas recomendado ya que genera un incremento en la masa de toxico ingerido de entre 2.85 y 14.75(supongamos miligramos) con respecto al cebo comercial. Asimismo, el tratamiento con SAC68 arroja valores similares de ingesta de masa de toxico que el SAC30 por lo que recomendamos este ultimo.

Parte B

Se desea conocer si el tiempo de ingesta de las hormigas difiere entre los tratamientos. Para ello:

- Plantee el modelo en parámetros y en términos del problema. ¿Qué representan los residuos?

Formato de regresión por dummies

$$E[Y_i] = \beta_0 + \beta_1 * SAC30_i + \beta_2 * SAC68_i$$

$$Y_i \sim N(0, \sigma^2)$$

$$i = 1 : 60$$

Formato de comparación de medias

$$E[Y_i] = \mu + \alpha_1$$

$$Y_i \sim N(\mu_i, \sigma^2)$$

$$i = 1 : 3$$

- ¿Cuál es la variable respuesta? ¿Cuál es la variable explicatoria? ¿Cuál podría ser la potencial distribución de probabilidades de la variable respuesta?

VR: Tiempo de ingesta (seg).

VE: Cebo.

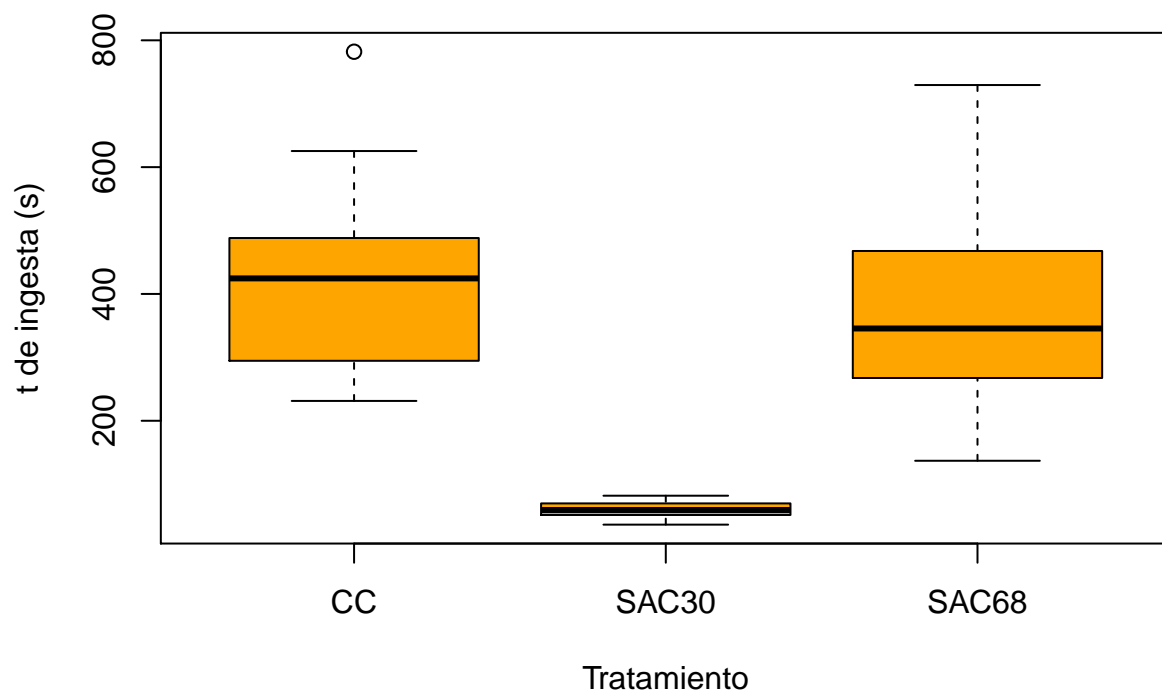
Potencial distribucion, Normal.

- Caracterice las muestras estadísticamente, con estadísticos de posición, dispersión y gráficos. ¿Cómo es la variabilidad en el tiempo de ingesta para cada tratamiento? ¿Qué tratamiento/s presentan mayor variabilidad? ¿Cuál/es menor?

```
library(psych)

##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##      %+%, alpha
describeBy(horm$T_ing, horm$Tratamiento)

##
## Descriptive statistics by group
## group: CC
##      vars  n   mean      sd median trimmed   mad   min  max range skew kurtosis
## X1      1 20 421.88 142.13 424.55  409.89 151.08 231.3 782 550.7 0.61    -0.08
##      se
## X1 31.78
## -----
## group: SAC30
##      vars  n mean      sd median trimmed   mad   min  max range skew kurtosis   se
## X1      1 20 60.25 12.58  59.25  60.17 13.49 36.2 81.8  45.6 0.07    -0.93 2.81
## -----
## group: SAC68
##      vars  n   mean      sd median trimmed   mad   min  max range skew kurtosis
## X1      1 20 374.08 151.6 345.55  362.66 131.28 136.9 729.5 592.6 0.61    -0.36
##      se
## X1 33.9
boxplot(horm$T_ing ~ horm$Tratamiento, xlab = "Tratamiento", ylab = "t de ingesta (s)",
        col = "orange")
```



*# Observe el grafico boxplot... que piensa que ocurrira con el
supuesto de homogeneidad de varianzas?*

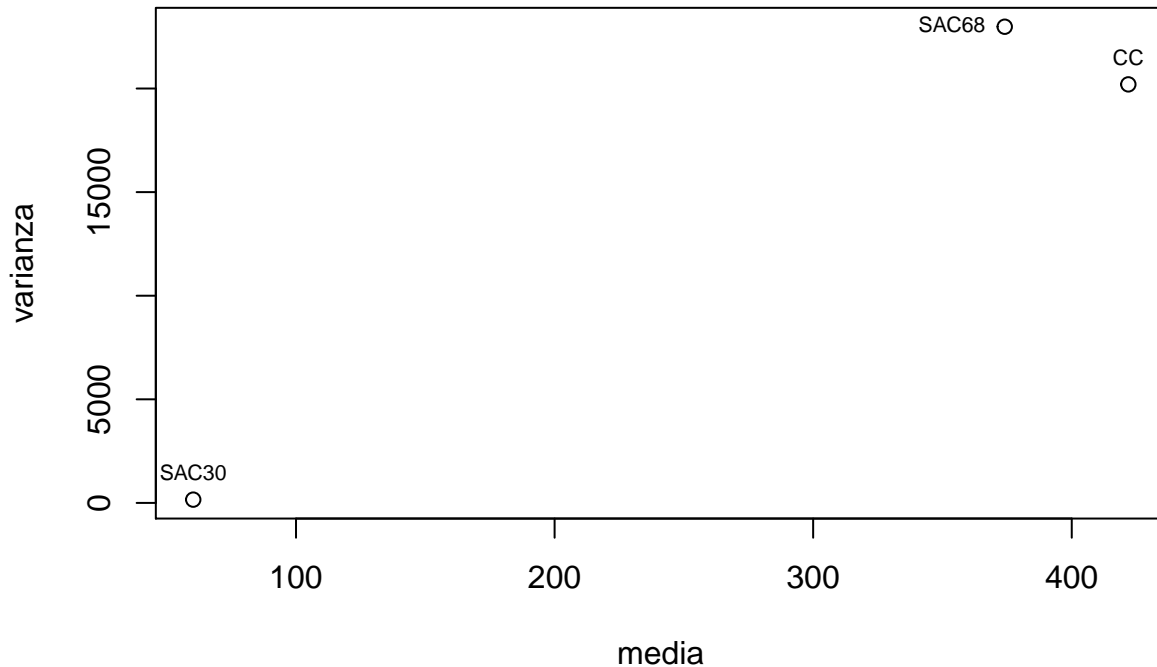
*# Relacion entre varianzas y valores esperados calculo medias y
varianzas, y las grafico*

```
media <- matrix(tapply(horm$T_ing, horm$Tratamiento, mean))
var <- matrix(tapply(horm$T_ing, horm$Tratamiento, sd))
(var <- var^2)
```

```
##           [,1]
## [1,] 20200.2592
## [2,]  158.3131
## [3,] 22981.8291
```

```
Trat <- c("CC", "SAC30", "SAC68")
plot(media, var, main = "Relacion varianza - media", ylab = "varianza")
text(media, var, labels = Trat, cex = 0.7, pos = c(3, 3, 2))
```

Relacion varianza – media



Segun los resultados obtenidos, en el grafico de relacion varianza-media para cada tratamiento se puede observar que las variabilidades entre los mismos son diferentes. Ademas viendo el boxplot podemos concluir que los tratamientos que presentan mayor variabilidad son CC y SAC68, mientras que, el que presenta menor variabilidad es SAC30.

- Plantee las hipótesis correspondientes.

H_0 : El tiempo de ingesta del toxico es igual para todos los tratamientos

H_a : El tiempo de ingesta del toxico no es el mismo para todos los tratamientos

- Evalúe los supuestos.

```
modelo1 <- lm(T_ing ~ Tratamiento, data = horm)
summary(modelo1) # ¿Tiene sentido interpretar esta salida YA?
```

```
##
## Call:
## lm(formula = T_ing ~ Tratamiento, data = horm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -237.18  -42.81   -4.60   30.69  360.11
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      421.89      26.88  15.697 < 2e-16 ***
## TratamientoSAC30 -361.63      38.01  -9.514 2.26e-13 ***
## TratamientoSAC68  -47.80      38.01  -1.258  0.214
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 120.2 on 57 degrees of freedom
## Multiple R-squared:  0.6521, Adjusted R-squared:  0.6399
## F-statistic: 53.43 on 2 and 57 DF, p-value: 8.524e-14
```

```

anova(modelo1)

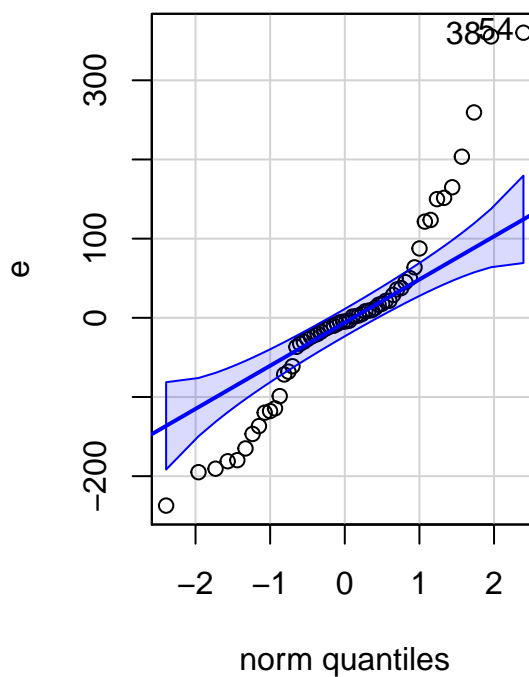
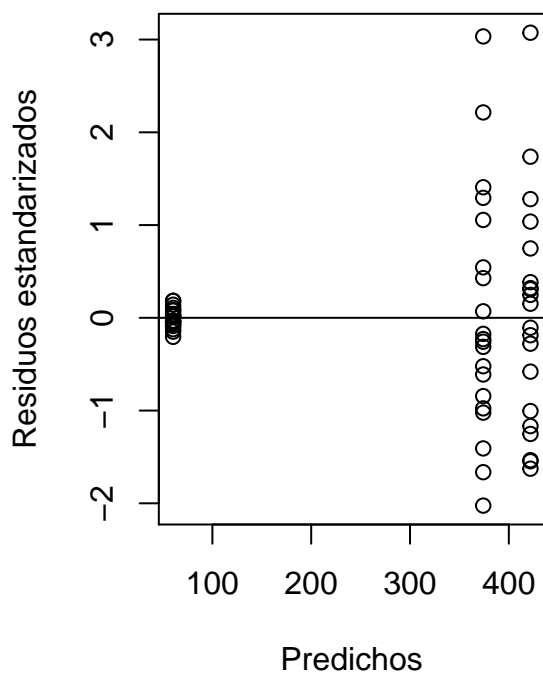
## Analysis of Variance Table
##
## Response: T_ing
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Tratamiento  2 1543651  771826  53.425 8.524e-14 ***
## Residuals    57  823468   14447
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

e <- residuals(modelo1) # residuos
re <- rstandard(modelo1) #residuos estandarizados
pre <- predict(modelo1) #predichos

# Graficamente
par(mfrow = c(1, 2)) # (Que hace este comando?)
plot(pre, re, xlab = "Predichos", ylab = "Residuos estandarizados",
      main = "Grafico de dispersion de RE vs PRED")
abline(0, 0)
car::qqPlot(e)

```

Grafico de dispersion de RE vs PR



```
## [1] 54 38
```

```

# Analíticamente
e = modelo1$residuals
shapiro.test(e)

```

```

##
## Shapiro-Wilk normality test
##
## data:  e
## W = 0.9215, p-value = 0.0008822

```

```

library(car)

## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:psych':
##
##      logit
## The following object is masked from 'package:dplyr':
##
##      recode
leveneTest(horm$T_ing, horm$Tratamiento, center = "median")

## Levene's Test for Homogeneity of Variance (center = "median")
##      Df F value   Pr(>F)
## group  2  11.149 8.19e-05 ***
##      57
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

sum(e)

## [1] -9.814372e-14

```

- Resuelva el modelo lineal utilizando la función `gls()`.

```

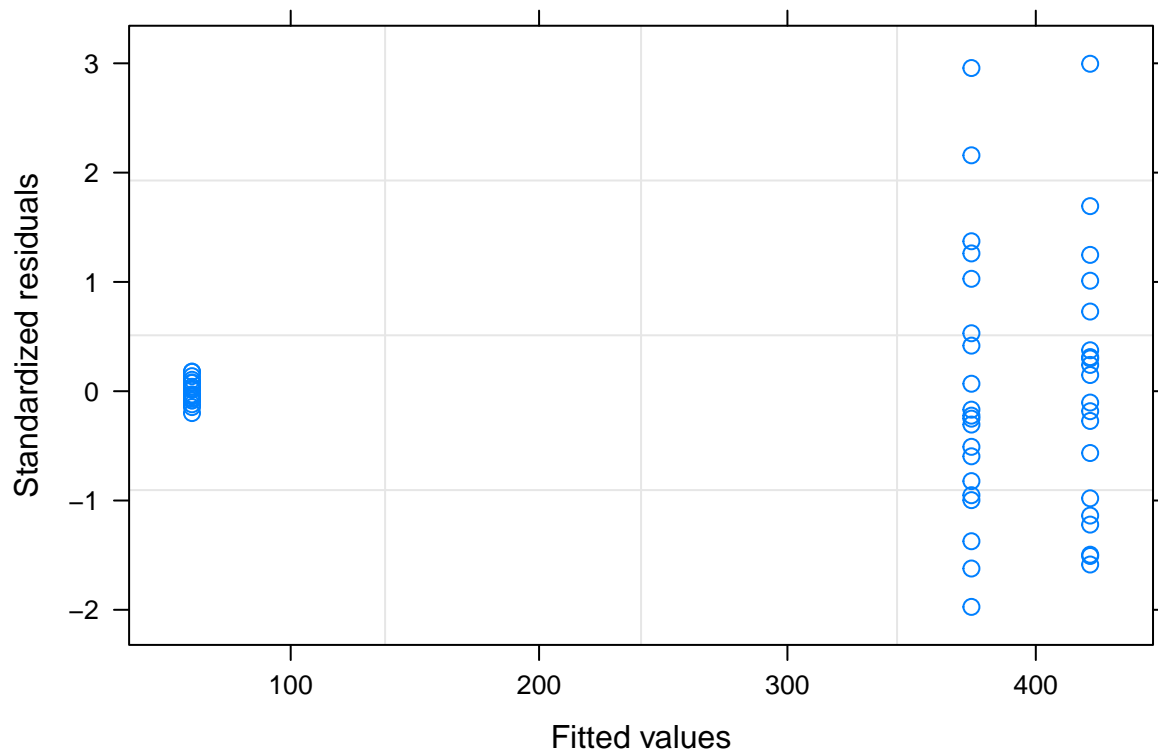
library("nlme") # para la función gls() necesitamos esta libreria

##
## Attaching package: 'nlme'
## The following object is masked from 'package:dplyr':
##
##      collapse
modelo2 <- gls(T_ing ~ Tratamiento, data = horm)
anova(modelo2)

## Denom. DF: 57
##      numDF F-value p-value
## (Intercept)    1 338.3045 <.0001
## Tratamiento    2  53.4254 <.0001

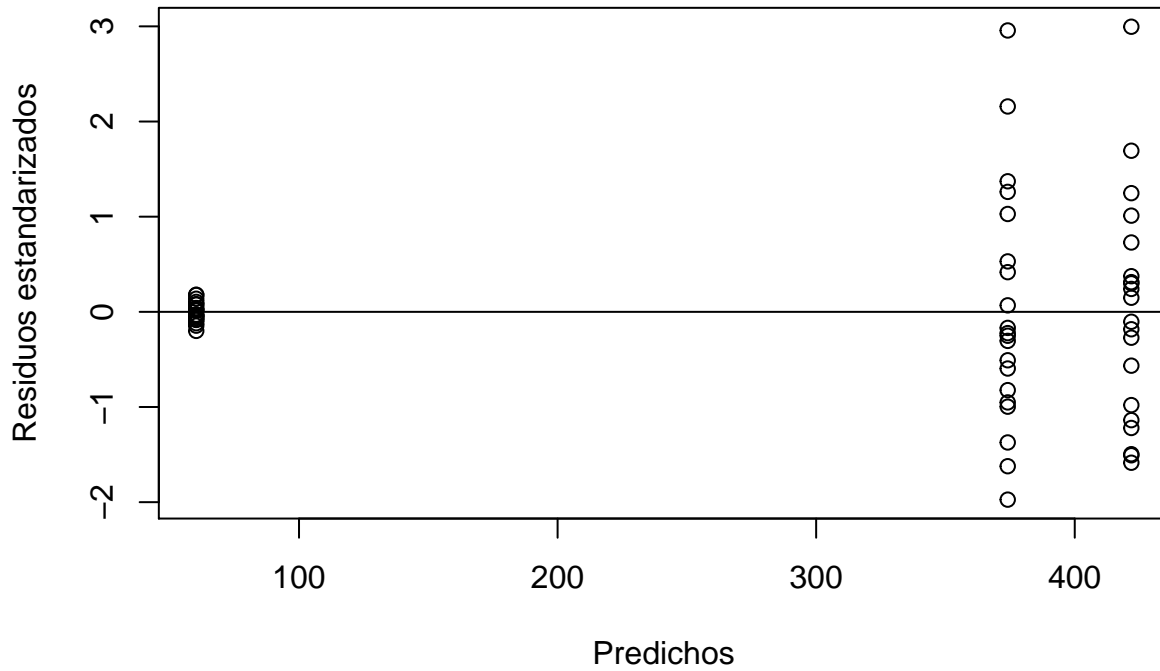
plot(modelo2)

```

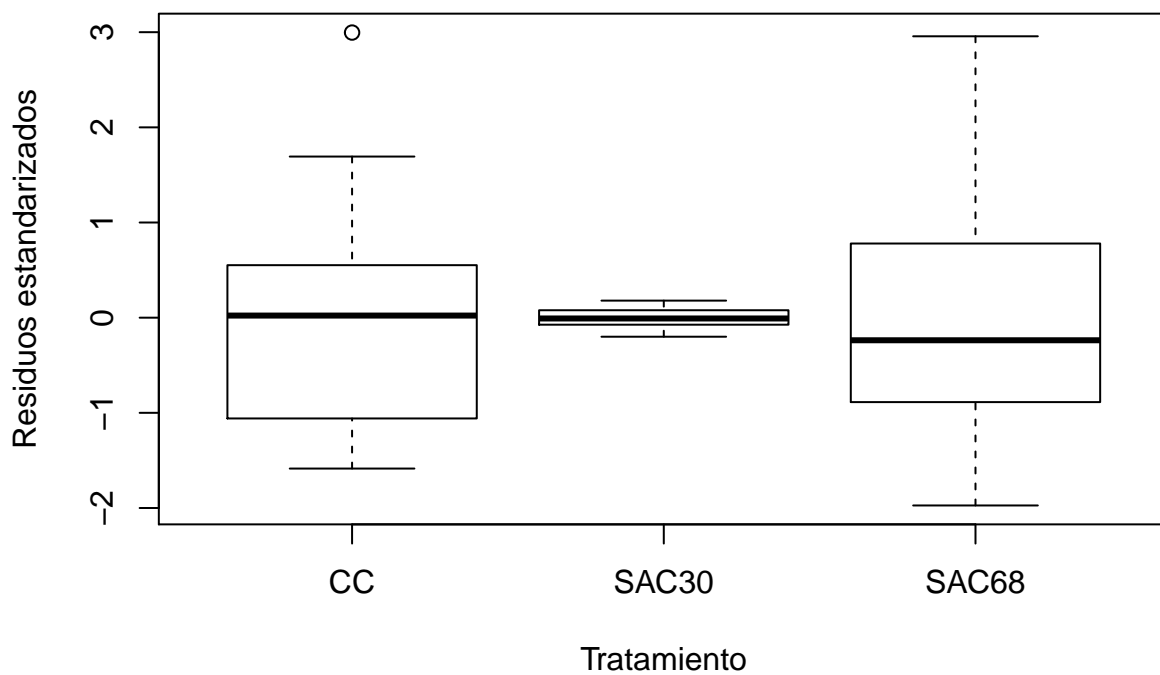



```
# Calculo de los residuos de pearson:
r2 <- residuals(modelo2, type = "pearson") #guardamos los residuos de Pearson = estandarizados
# Calculo de los valores predichos por el modelo:
pred2 <- fitted(modelo2)
plot(pred2, r2, xlab = "Predichos", ylab = "Residuos estandarizados",
      main = "Gráfico de dispersión de RE vs PRED")
abline(0, 0)
```

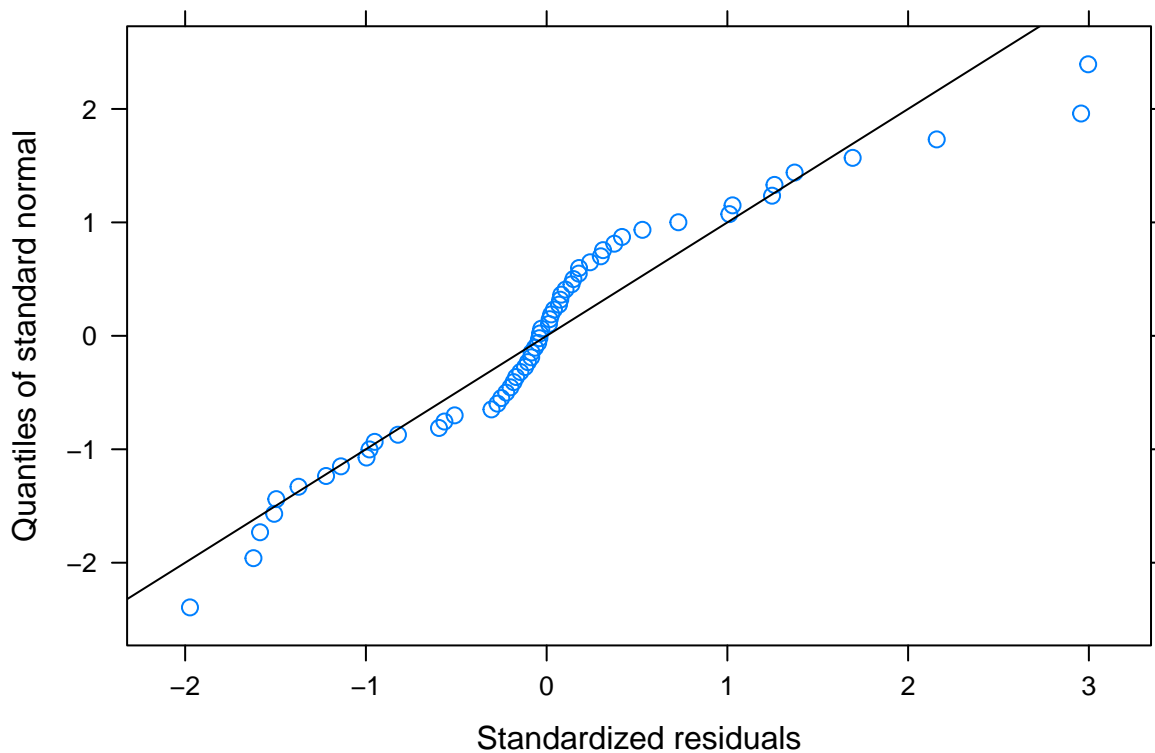
Gráfico de dispersión de RE vs PRED



```
boxplot(r2 ~ horm$Tratamiento, xlab = "Tratamiento", ylab = "Residuos estandarizados")
```



```
qqnorm(modelo2, abline = c(0, 1)) #QQ plot
```



```
leveneTest(r2, horm$Tratamiento, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##      Df F value  Pr(>F)
## group 2  11.149 8.19e-05 ***
##      57
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Antes de continuar: ¿Qué diferencia/s existe entre las funciones `lm()` y `gls()`?

`gls` realiza estimaciones utilizando máxima verosimilitud, mientras que `lm` utiliza mínimos cuadrados.

- Modele la varianza ¿Qué funciones podría utilizar para modelar la varianza?. Elija las funciones de varianza que considere adecuadas, corra los modelos con todas ellas y seleccione el mejor modelo.

```
modelo_varIdent <- gls(T_ing ~ Tratamiento, weights = varIdent(form = ~1 |
  Tratamiento), data = horm)
anova(modelo_varIdent)
```

```
## Denom. DF: 57
##      numDF F-value p-value
## (Intercept)    1 544.5897 <.0001
## Tratamiento    2 106.0402 <.0001
```

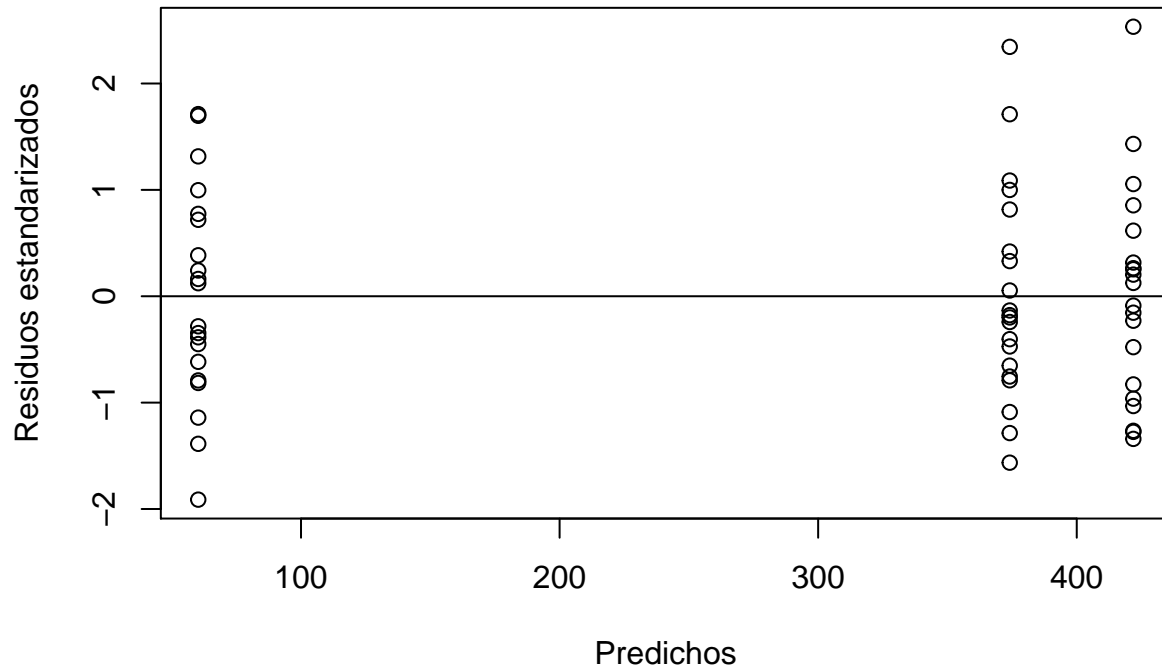
```
summary(modelo_varIdent)
```

```
## Generalized least squares fit by REML
## Model: T_ing ~ Tratamiento
## Data: horm
##      AIC      BIC    logLik
## 658.1354 670.3937 -323.0677
```

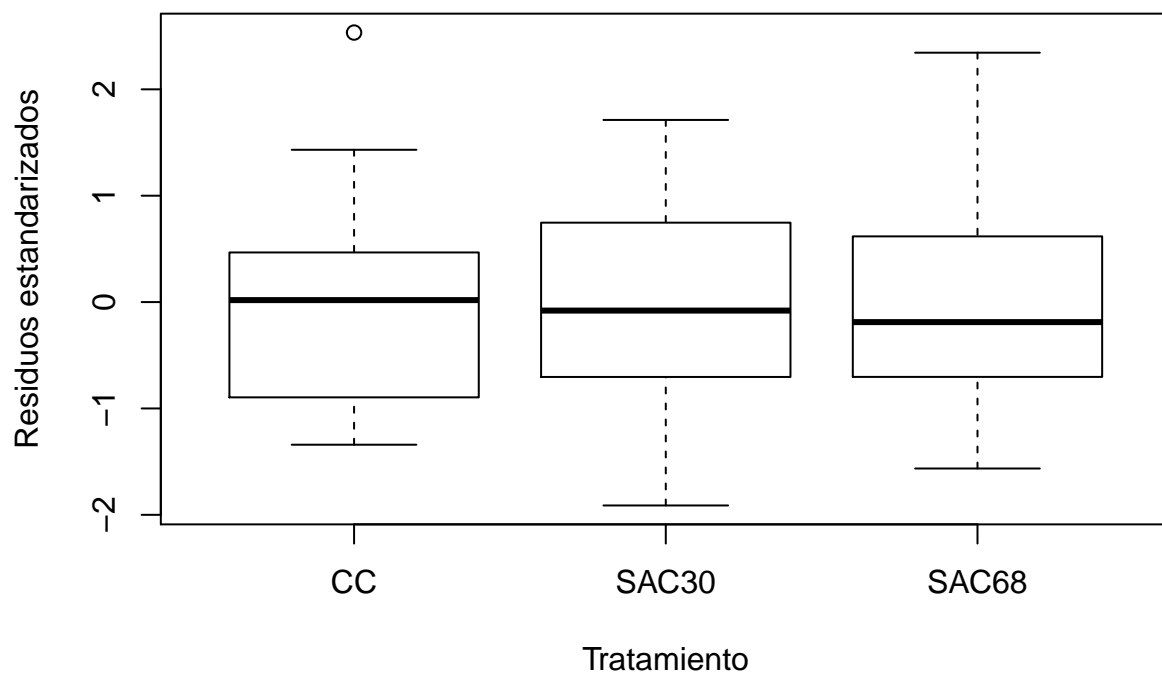
```
##
## Variance function:
## Structure: Different standard deviations per stratum
## Formula: ~1 | Tratamiento
## Parameter estimates:
##   SAC30   SAC68   CC
## 1.00000 12.04851 11.29585
##
## Coefficients:
##               Value Std.Error   t-value p-value
## (Intercept)    421.885   31.78065   13.274901  0.0000
## TratamientoSAC30 -361.630   31.90494  -11.334607  0.0000
## TratamientoSAC68  -47.805   46.46614   -1.028814  0.3079
##
## Correlation:
##               (Intr) TSAC30
## TratamientoSAC30 -0.996
## TratamientoSAC68 -0.684  0.681
##
## Standardized residuals:
##           Min           Q1           Med           Q3           Max
## -1.9118172 -0.7634024 -0.1442302  0.6420577  2.5337480
##
## Residual standard error: 12.58227
## Degrees of freedom: 60 total; 57 residual

r3 <- residuals(modelo_varIdent, type = "p") #guardamos los residuos de Pearson = estandarizados
# Calculo de los valores predichos por el modelo:
pred3 <- fitted(modelo_varIdent)
plot(pred3, r3, xlab = "Predichos", ylab = "Residuos estandarizados",
     main = "Gráfico de dispersión de RE vs PRED")
abline(0, 0)
```

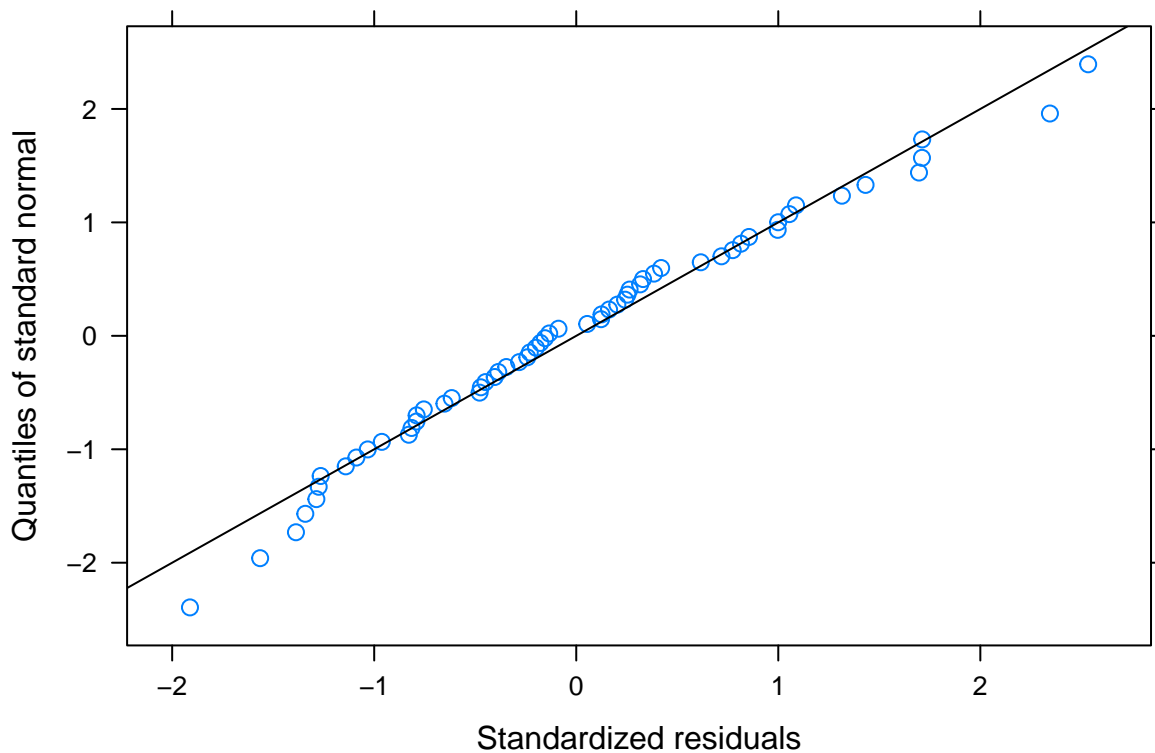
Gráfico de dispersión de RE vs PRED



```
boxplot(r3 ~ horm$Tratamiento, xlab = "Tratamiento", ylab = "Residuos estandarizados")
```



```
qqnorm(modelo_varIdent, abline = c(0, 1)) #QQ plot
```



```
leveneTest(r3, horm$Tratamiento, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##      Df F value Pr(>F)
## group 2  0.0625 0.9395
##      57
```

```
AIC(modelo2, modelo_varIdent)
```

```
##           df      AIC
## modelo2      4 724.7052
## modelo_varIdent 6 658.1354
```

```
modelo_varPower <- gls(T_ing ~ Tratamiento, weights = varPower(),
  data = horm)
anova(modelo_varPower)
```

```
## Denom. DF: 57
##           numDF  F-value p-value
## (Intercept)      1 536.3213 <.0001
## Tratamiento      2 102.0421 <.0001
```

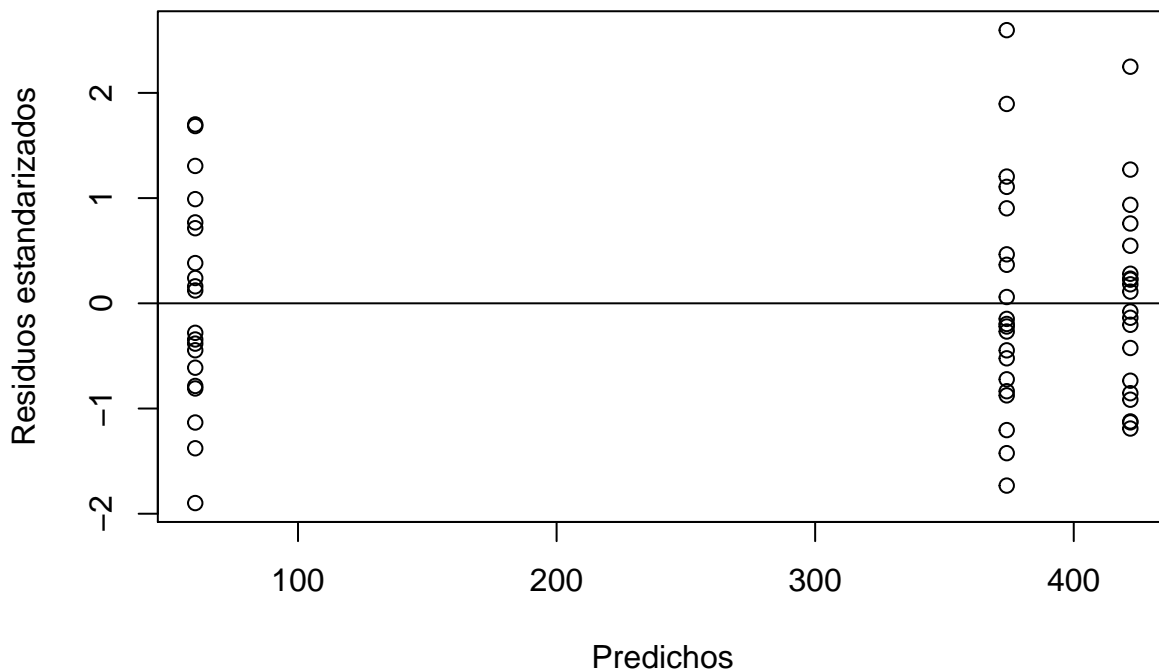
```
summary(modelo_varPower)
```

```
## Generalized least squares fit by REML
## Model: T_ing ~ Tratamiento
## Data: horm
##      AIC      BIC    logLik
## 657.0609 667.2761 -323.5304
##
## Variance function:
```

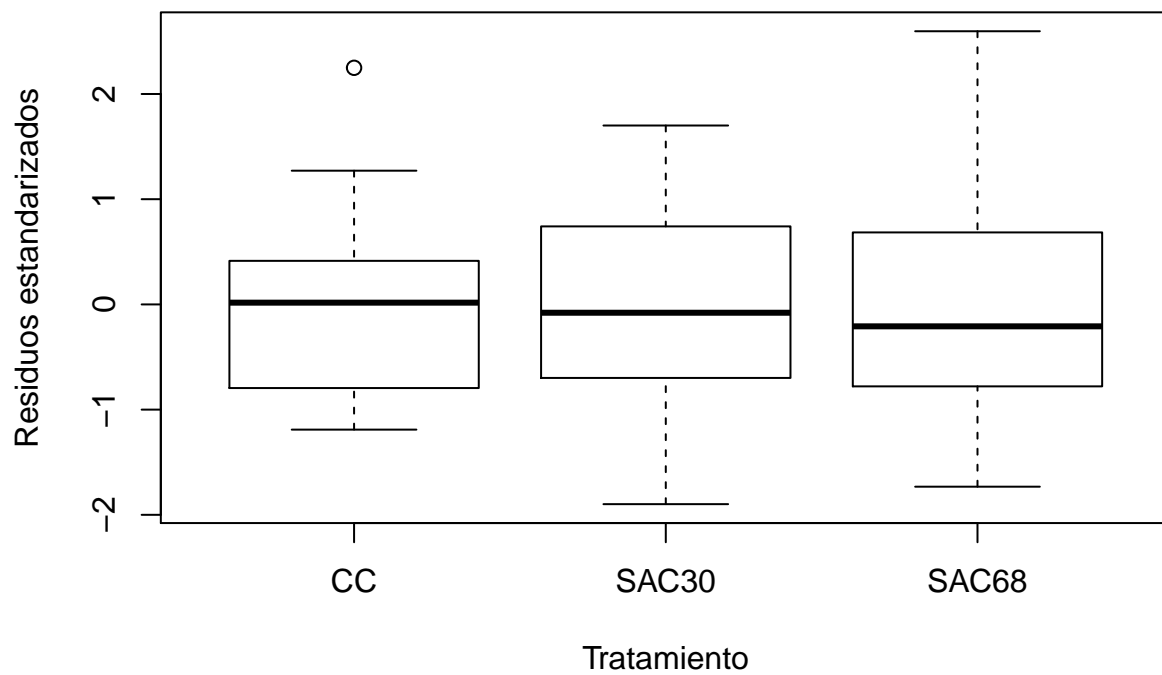
```
## Structure: Power of variance covariate
## Formula: ~fitted(.)
## Parameter estimates:
## power
## 1.30344
##
## Coefficients:
##               Value Std.Error   t-value p-value
## (Intercept)    421.885   35.80712   11.782154  0.0000
## TratamientoSAC30 -361.630   35.91904  -10.067918  0.0000
## TratamientoSAC68  -47.805   47.10884   -1.014778  0.3145
##
## Correlation:
##               (Intr) TSAC30
## TratamientoSAC30 -0.997
## TratamientoSAC68 -0.760  0.758
##
## Standardized residuals:
##               Min         Q1         Med         Q3         Max
## -1.8984089 -0.7480650 -0.1427137  0.5888076  2.5961849
##
## Residual standard error: 0.06063193
## Degrees of freedom: 60 total; 57 residual
```

```
# Estudiamos los residuos
r4 <- residuals(modelo_varPower, type = "p") #guardamos los residuos de Pearson = estandarizados
pred4 <- fitted(modelo_varPower)
plot(pred4, r4, xlab = "Predichos", ylab = "Residuos estandarizados",
      main = "Gráfico de dispersión de RE vs PRED")
abline(0, 0)
```

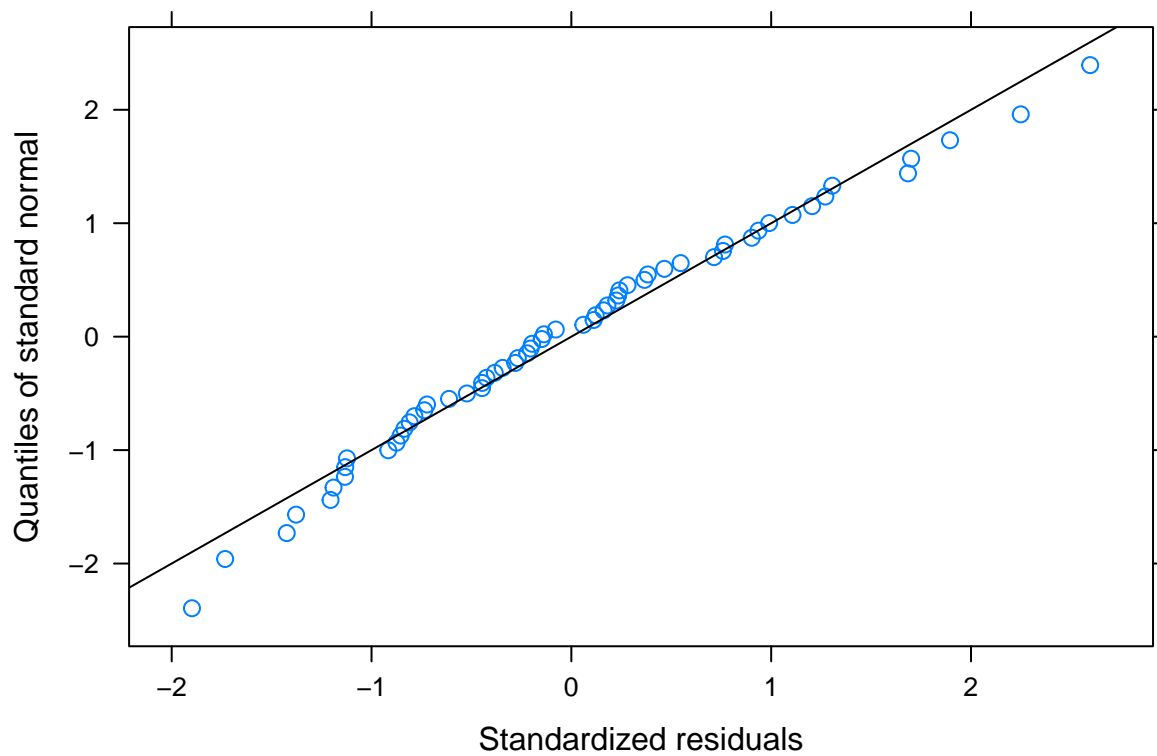
Gráfico de dispersión de RE vs PRED



```
boxplot(r4 ~ horm$Tratamiento, xlab = "Tratamiento", ylab = "Residuos estandarizados")
```



```
qqnorm(modelo_varPower, abline = c(0, 1)) #QQ plot
```




```

leveneTest(r4, horm$Tratamiento, center = "median")

## Levene's Test for Homogeneity of Variance (center = "median")
##      Df F value Pr(>F)
## group  2  0.3253 0.7236
##      57

AIC(modelo2, modelo_varIdent, modelo_varPower)

##           df      AIC
## modelo2      4 724.7052
## modelo_varIdent 6 658.1354
## modelo_varPower 5 657.0609

# modelo 5 modelado varianzas con la fc 'varExp'
modelo_varExp <- gls(T_ing ~ Tratamiento, weights = varExp(), data = horm)
anova(modelo_varExp)

## Denom. DF: 57
##           numDF  F-value p-value
## (Intercept)      1 516.6296 <.0001
## Tratamiento      2 100.3053 <.0001

summary(modelo_varExp)

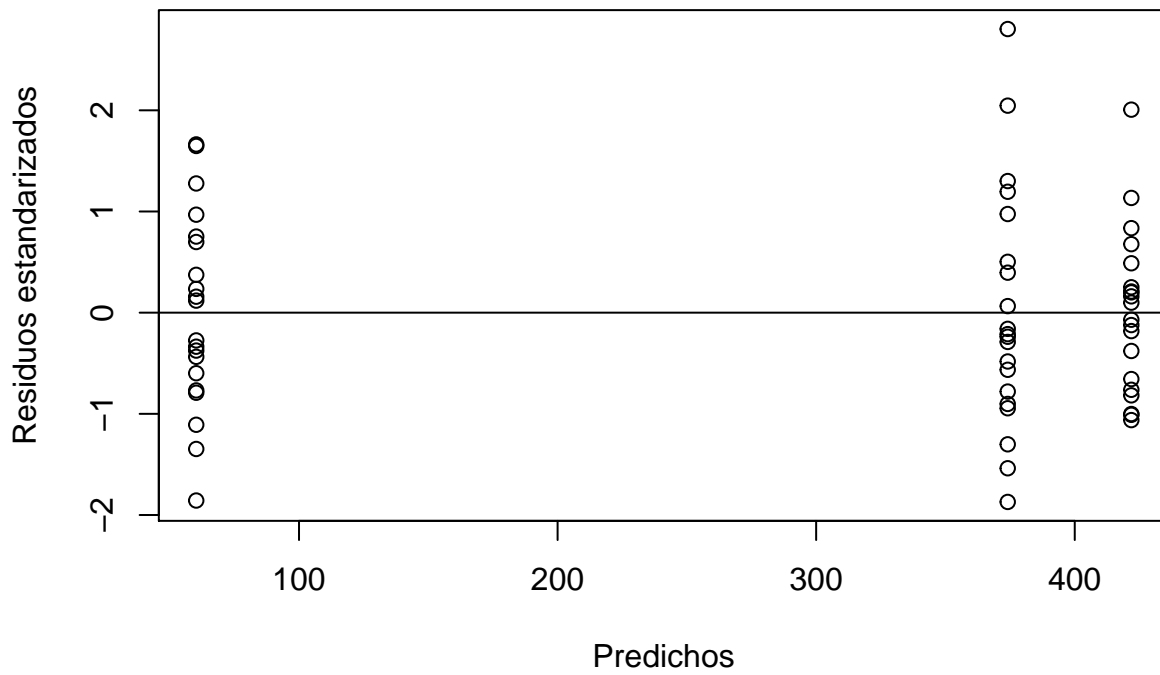
## Generalized least squares fit by REML
##   Model: T_ing ~ Tratamiento
##   Data: horm
##           AIC      BIC    logLik
##   659.3197 669.5349 -324.6598
##
## Variance function:
##   Structure: Exponential of variance covariate
##   Formula: ~fitted(.)
##   Parameter estimates:
##           expon
## 0.007268394
##
## Coefficients:
##           Value Std.Error   t-value p-value
## (Intercept)   421.885  40.13003 10.512949  0.0000
## TratamientoSAC30 -361.630  40.23446 -8.988066  0.0000
## TratamientoSAC68  -47.805  49.13445 -0.972943  0.3347
##
## Correlation:
##           (Intr) TSAC30
## TratamientoSAC30 -0.997
## TratamientoSAC68 -0.817  0.815
##
## Standardized residuals:
##           Min      Q1      Med      Q3      Max
## -1.8706645 -0.7637298 -0.1412261  0.5461970  2.8032363
##
## Residual standard error: 8.360891
## Degrees of freedom: 60 total; 57 residual

# Estudiamos los residuos
r5 <- residuals(modelo_varExp, type = "p") #guardamos los residuos de Pearson = estandarizados
pred5 <- fitted(modelo_varExp)
plot(pred5, r5, xlab = "Predichos", ylab = "Residuos estandarizados",
     main = "Gráfico de dispersión de RE vs PRED")

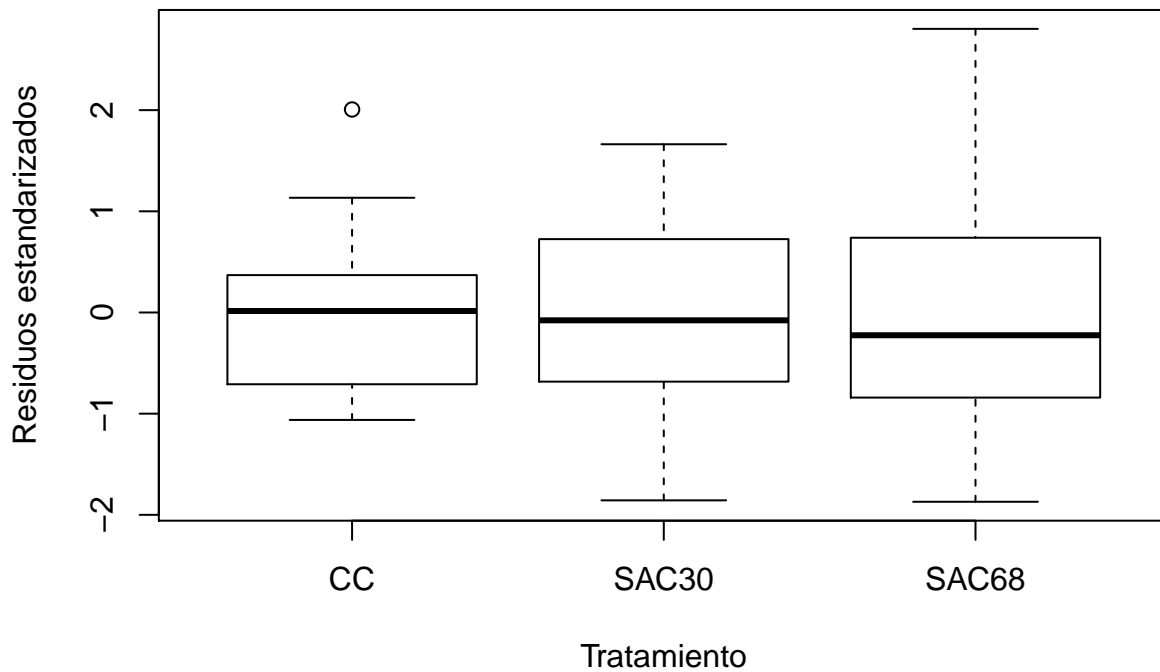
```

```
abline(0, 0)
```

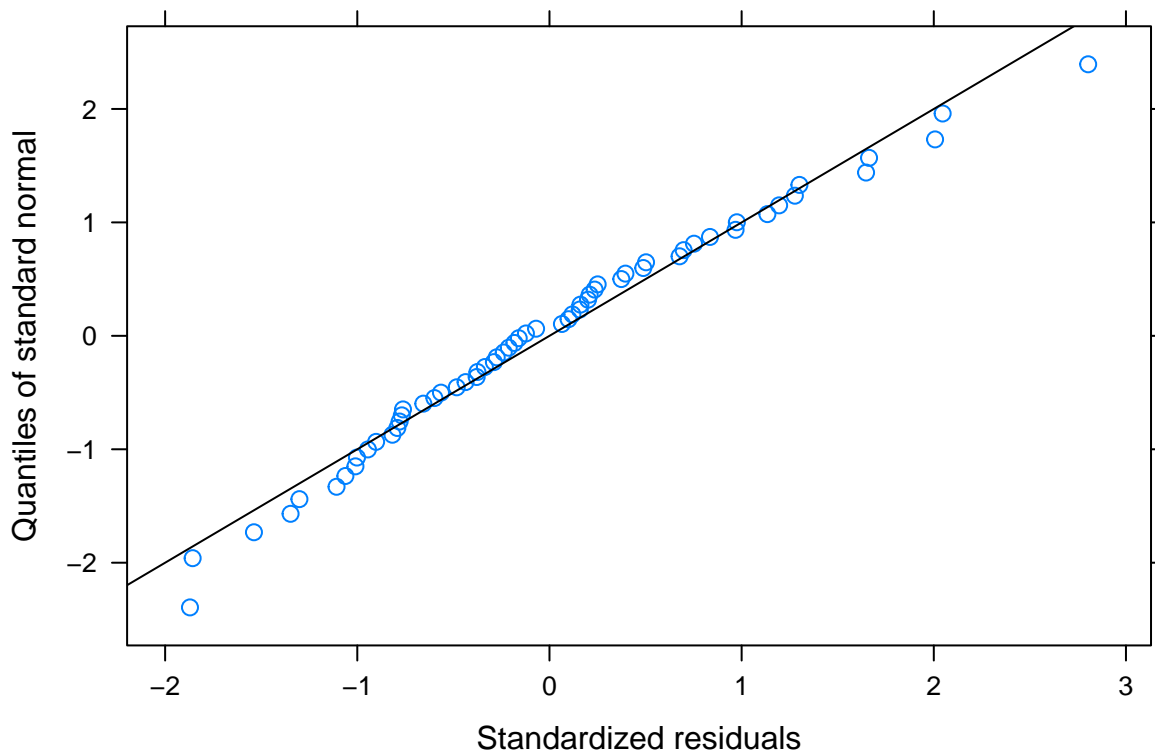
Gráfico de dispersión de RE vs PRED



```
boxplot(r5 ~ horm$Tratamiento, xlab = "Tratamiento", ylab = "Residuos estandarizados")
```



```
qqnorm(modelo_varExp, abline = c(0, 1)) #QQ plot
```



```
leveneTest(r5, horm$Tratamiento, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##      Df F value Pr(>F)
## group 2  1.0582 0.3538
##      57
```

```
AIC(modelo2, modelo_varIdent, modelo_varPower, modelo_varExp)
```

```
##           df      AIC
## modelo2      4 724.7052
## modelo_varIdent 6 658.1354
## modelo_varPower 5 657.0609
## modelo_varExp   5 659.3197
```

```
# Seleccionamos modelo
summary(modelo_varPower)
```

```
## Generalized least squares fit by REML
## Model: T_ing ~ Tratamiento
## Data: horm
##      AIC      BIC    logLik
## 657.0609 667.2761 -323.5304
##
## Variance function:
## Structure: Power of variance covariate
## Formula: ~fitted(.)
## Parameter estimates:
## power
## 1.30344
##
```

```
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)    421.885   35.80712   11.782154  0.0000
## TratamientoSAC30 -361.630   35.91904  -10.067918  0.0000
## TratamientoSAC68  -47.805   47.10884   -1.014778  0.3145
##
## Correlation:
##              (Intr) TSAC30
## TratamientoSAC30 -0.997
## TratamientoSAC68 -0.760  0.758
##
## Standardized residuals:
##              Min          Q1          Med          Q3          Max
## -1.8984089 -0.7480650 -0.1427137  0.5888076  2.5961849
##
## Residual standard error: 0.06063193
## Degrees of freedom: 60 total; 57 residual
```

```
anova(modelo_varPower)
```

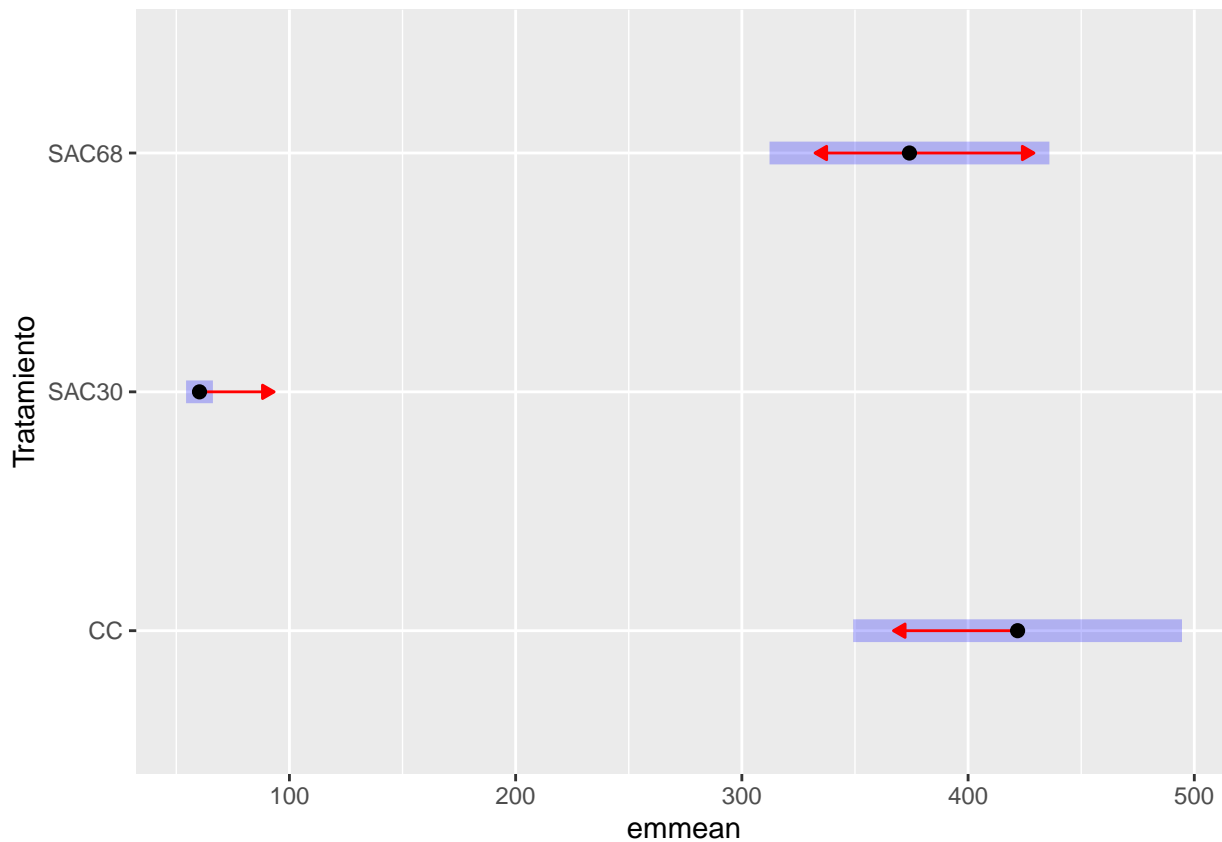
```
## Denom. DF: 57
##              numDF  F-value p-value
## (Intercept)      1 536.3213 <.0001
## Tratamiento      2 102.0421 <.0001
```

- Si corresponde, realice comparaciones.

```
## $emmeans
## Tratamiento emmean    SE    df lower.CL upper.CL t.ratio p.value
## CC              421.9 35.81 35.4    349.2    494.5  11.782 <.0001
## SAC30            60.3  2.83 18.8     54.3     66.2  21.266 <.0001
## SAC68           374.1 30.61 40.3    312.2    435.9  12.220 <.0001
##
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
##
## $contrasts
## contrast      estimate    SE    df lower.CL upper.CL t.ratio p.value
## CC - SAC30      361.6 35.9 35.9    273.8     449  10.068 <.0001
## CC - SAC68      47.8 47.1 37.4    -67.2     163   1.015  0.5723
## SAC30 - SAC68  -313.8 30.7 40.9   -388.6    -239 -10.208 <.0001
##
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 3 estimates
## P value adjustment: tukey method for comparing a family of 3 estimates

## $emmeans
## Tratamiento emmean    SE    df lower.CL upper.CL
## CC              421.9 35.81 35.4    349.2    494.5
## SAC30            60.3  2.83 18.8     54.3     66.2
## SAC68           374.1 30.61 40.3    312.2    435.9
##
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
##
## $contrasts
## contrast      estimate    SE    df lower.CL upper.CL
## CC - SAC30      361.6 35.9 35.9    273.8     449
## CC - SAC68      47.8 47.1 37.4    -67.2     163
## SAC30 - SAC68  -313.8 30.7 40.9   -388.6    -239
```

```
##
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 3 estimates
```

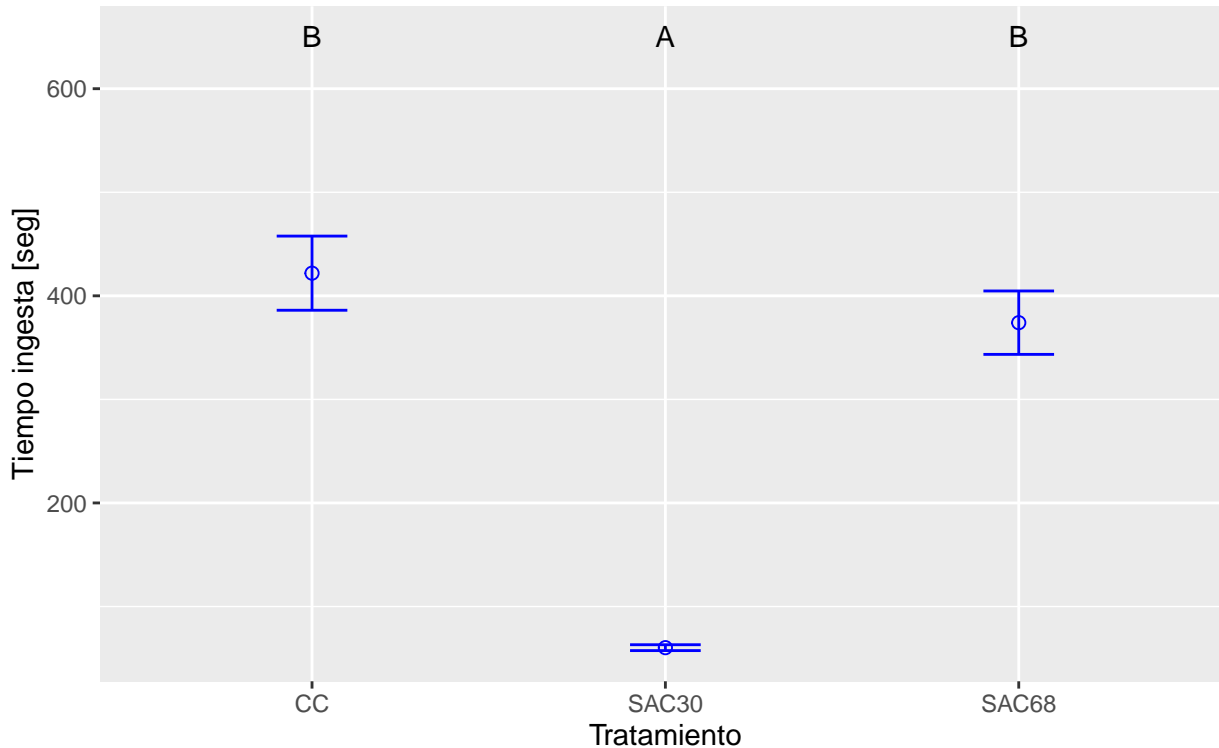


- Concluya e Informe sus resultados presentado un gráfico con epígrafe.

```
resumen_modelo <- as.data.frame(multCompTukey$emmeans)
df_niveles <- data.frame(x = rep(c("CC", "SAC30", "SAC68"), each = 1),
  y = 650, label = c("B", "A", "B"))
ggplot(resumen_modelo, aes(x = Tratamiento, y = emmean)) + labs(x = "Tratamiento") +
  labs(y = "Tiempo ingesta [seg]") + geom_errorbar(aes(ymin = emmean -
    SE, ymax = emmean + SE), color = "blue", width = 0.2) + geom_point(shape = 1,
    size = 2, color = "blue") + ggtitle("Comparaciones", "Media ± Error estándar") +
  geom_text(data = df_niveles, aes(x = x, y = y, label = label),
    position = position_dodge(width = 0.9))
```

Comparaciones

Media \pm Error estándar



Tiempo de ingesta observado en función del tratamiento.

Un análisis de la varianza a partir de un modelo lineal general modelando heterocedasticidad, evidencia diferencias significativas entre los tratamientos ($F_{2,57}=102$, $p<0.05$). Letras indican diferencias entre grupos detectadas a través de comparaciones a posteriori (Tukey, $p < 0.05$).

Problema 2. Efecto de la fracción lipoproteica HDL sobre la progresión de las lesiones ateroscleróticas en conejos alimentados con dieta rica en colesterol

Badimon y cols (J.Clin.Invest 85: 1234, 1990) llevaron a cabo un estudio a fin de determinar el efecto de la fracción lipoproteica HDL-VHDL sobre lesiones ateroscleróticas en conejos Nueva Zelanda. La aterosclerosis fue inducida por la alimentación de los animales con una dieta rica en colesterol 0,5% durante 60 días (Dieta 1). Otro grupo de animales fueron mantenidos con la misma dieta durante 90 días (Dieta 2). Un tercer grupo se alimentó con la misma dieta durante 90 días, pero recibió 50 mg de proteína HDL-VHDL por semana durante los últimos 30 días (Dieta 3). Al finalizar el ensayo los animales fueron sacrificados. En todos los casos se comprobaron lesiones aterógenicas en aorta. Se midió el contenido de colesterol en aorta (en mg/g). Los datos se encuentran en el archivo colesterol.txt 1. Identifique la variable respuesta y explicatoria. ¿Cuál es el diseño experimental? ¿Cuál es la cantidad de réplicas? 2. Escriba el modelo en parámetros y en términos del problema. 3. Concluya en relación al efecto de la dieta rica en colesterol y la suministración adicional de la proteína HDL-VHDL. Igual que antes, antes de efectuar las pruebas analice los supuestos del modelo. 4. Resuelva el problema estimando las SC por mínimos cuadrados (enfoque “clásico”, función `lm()` ó `aov()`) y compare los resultados obtenidos.

Problema 3. Exposición prenatal a mercurio y actividad de la bomba de calcio

Si bien los efectos neurotóxicos de la exposición prenatal a altas dosis de mercurio están bien establecidos, poco se sabe sobre los efectos de la exposición prenatal a niveles por debajo de 10 ppm. Algunos autores han reportado deficiencias neuroconductuales sutiles, especialmente con respecto al lenguaje, atención y memoria. Se sospecha que estos efectos pueden estar mediados por la desregulación de la homeostasis del Ca a través de la inhibición de la bomba de Ca-ATPasa de la membrana plasmática. Se llevó a cabo un estudio en una maternidad a fin de determinar si la exposición materna al mercurio medida por los depósitos de mercurio en el cabello (en $\mu\text{g/g}$) se relacionaba con la actividad de la bomba de Ca del recién nacido (nmol/mg /hr). Para ello se seleccionaron 75 recién nacidos y sus madres. Se tomaron muestras de pelo de 3 cm desde la raíz. El cabello promedio crece aproximadamente 1 cm por mes, por lo que las muestras de cabello

recogidas reflejaron las exposiciones durante los 3-4 meses previos al parto. La actividad de la bomba de Ca se midió en glóbulos rojos de sangre de cordón umbilical de los recién nacidos. Los datos en archivo pelo.txt.

1- Represente los datos en un diagrama de dispersión. ¿Parece que existe una tendencia lineal en los datos? ¿Directa o inversa? ¿Débil o fuerte?

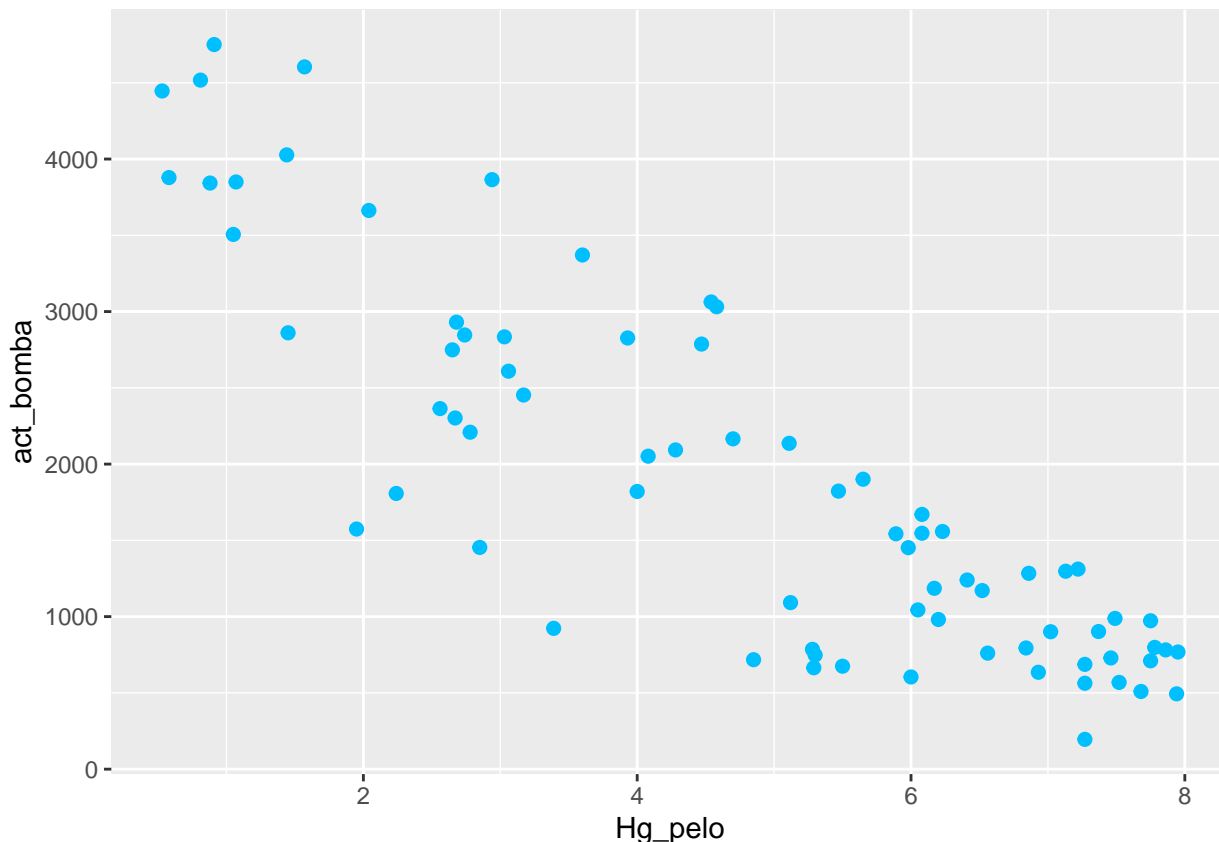
```
library(ggplot2)
library(nlme)
datos = read.delim("/home/jose/Documents/materias/biome2/2018/tps/tp3/pelo.txt",
  header = T, sep = "\t")
colnames(datos) = c("id", "Hg_pelo", "act_bomba")
attach(datos)
head(datos)
```

```
##   id Hg_pelo act_bomba
## 1  1    5.11  2136.09
## 2  2    0.53  4446.25
## 3  3    1.95  1574.30
## 4  4    6.00   604.38
## 5  5    5.65  1900.99
## 6  6    3.93  2826.85
```

```
str(datos)
```

```
## 'data.frame':   75 obs. of  3 variables:
## $ id          : int  1 2 3 4 5 6 7 8 9 10 ...
## $ Hg_pelo     : num  5.11 0.53 1.95 6 5.65 3.93 3.06 6.41 0.88 1.57 ...
## $ act_bomba   : num  2136 4446 1574 604 1901 ...
```

```
ggplot(datos, aes(x = Hg_pelo, y = act_bomba)) + geom_point(aes(),
  colour = "deepskyblue", size = 2)
```

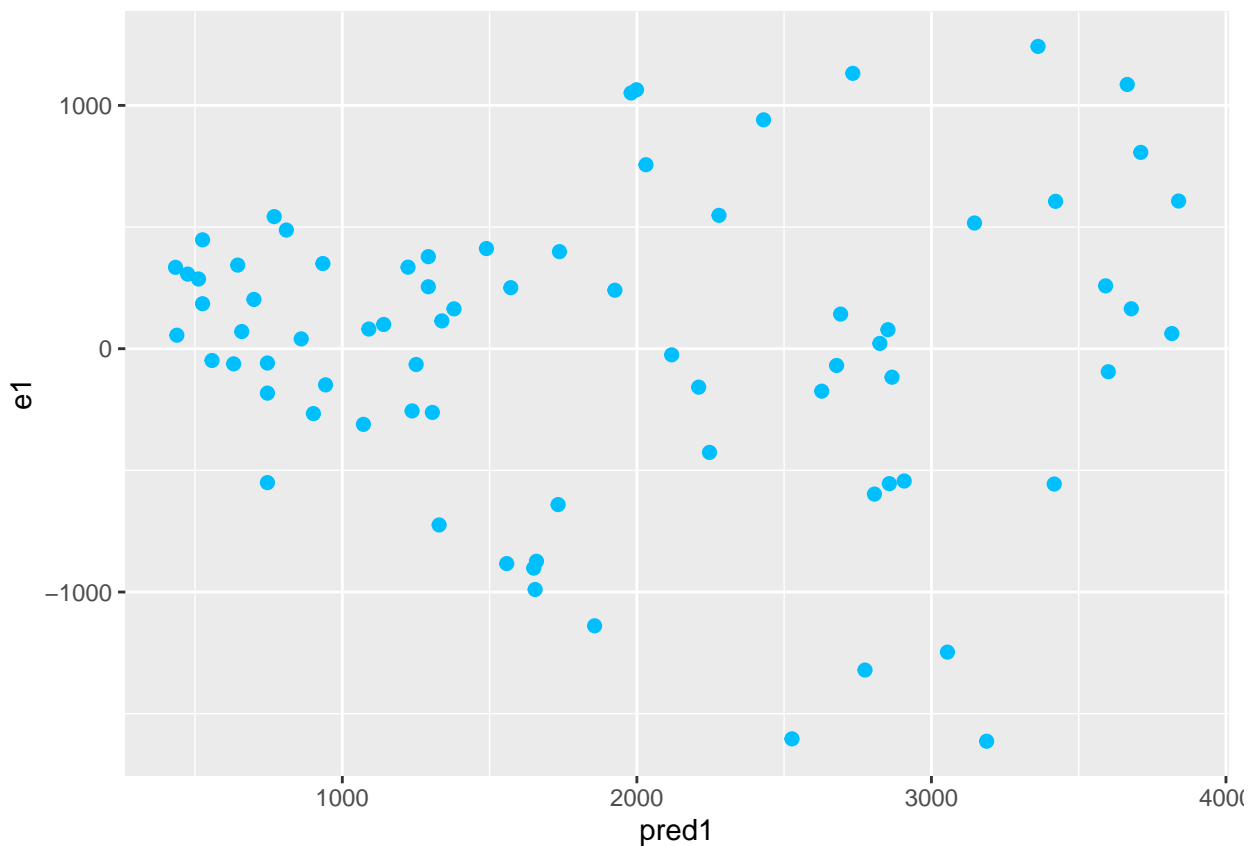


2- Estime la relación funcional. Interprete la pendiente en contexto. ¿Cuáles son sus unidades?

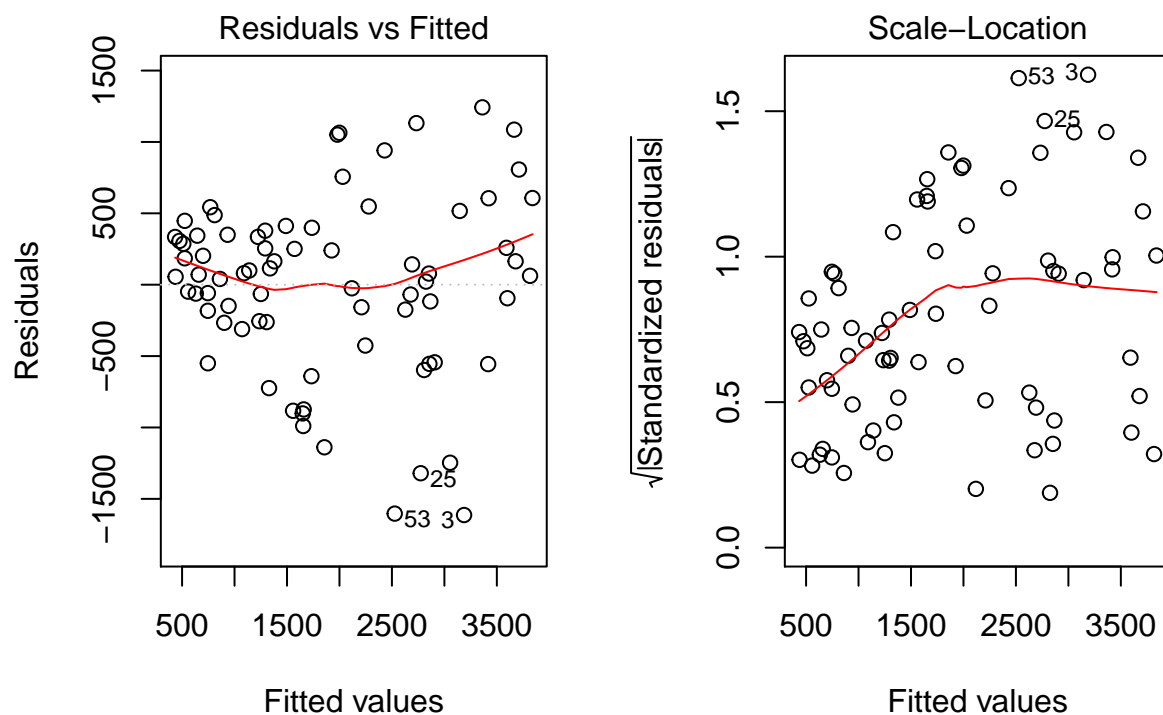
```
modelo1 <- lm(act_bomba ~ Hg_pelo, datos)
# supuestos
e1 = modelo1$residuals
shapiro.test(e1)
```

```
##
## Shapiro-Wilk normality test
##
## data: e1
## W = 0.96938, p-value = 0.06602
```

```
pred1 <- fitted(modelo1)
sup.mod1 = as.data.frame(cbind(e1, pred1))
ggplot(sup.mod1, aes(x = pred1, y = e1)) + geom_point(aes(), colour = "deepskyblue",
  size = 2)
```



```
par(mfrow = c(1, 2))
plot(modelo1, which = c(1, 3))
```

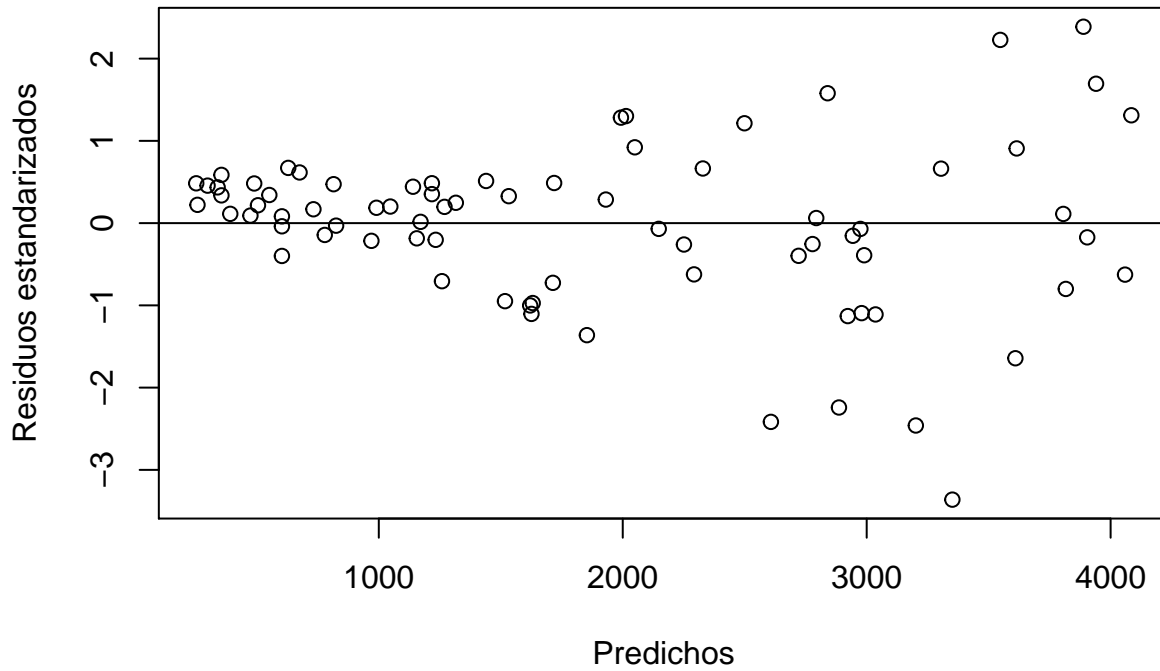
```
sum(e1)
```

```
## [1] -2.515321e-12
```

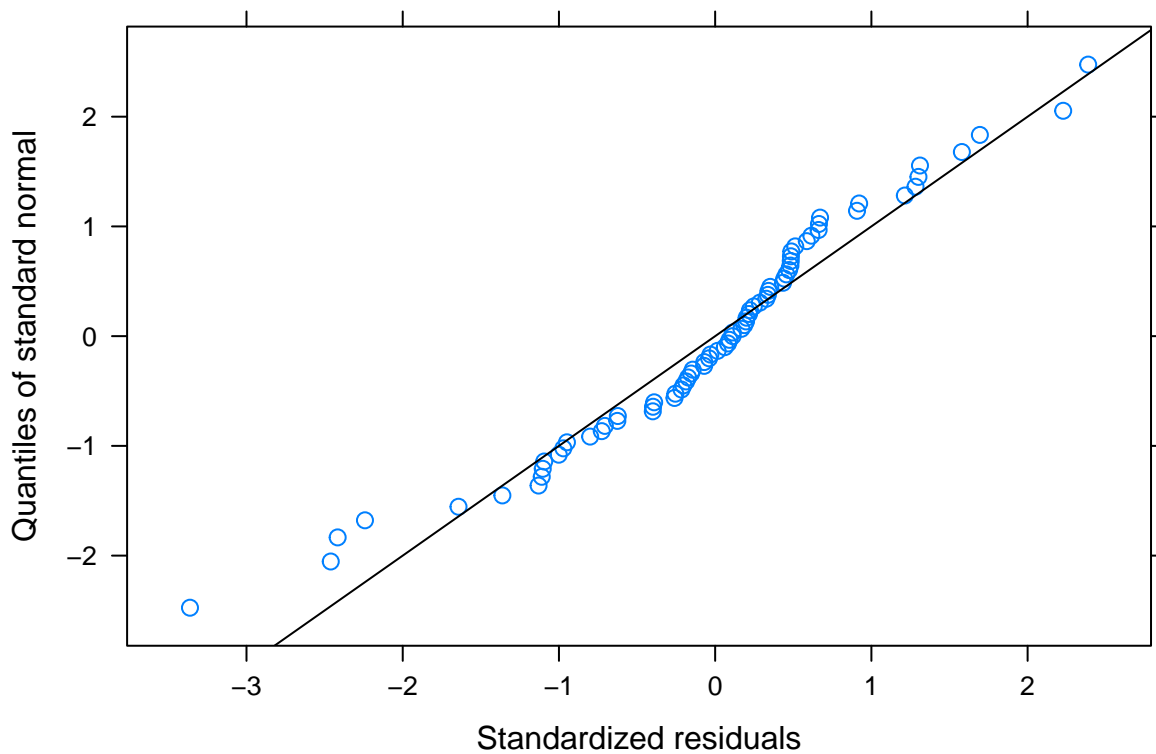
3- ¿Existe evidencia significativa de que la actividad basal de la bomba de Ca en el recién nacido disminuye linealmente con el nivel de mercurio en pelo materno?

```
##### Vamos a modelar
modelo_varFixed1 <- gls(act_bomba ~ Hg_pelo, weights = varFixed(~Hg_pelo),
  data = datos)
# Calculo de los residuos de pearson:
r2 <- residuals(modelo_varFixed1, type = "p") #guardamos los residuos de Pearson = estandarizados
# Calculo de los valores predichos por el modelo:
pred2 <- fitted(modelo_varFixed1)
plot(pred2, r2, xlab = "Predichos", ylab = "Residuos estandarizados",
  main = "Gráfico de dispersión de RE vs PRED")
abline(0, 0)
```

Gráfico de dispersión de RE vs PRED



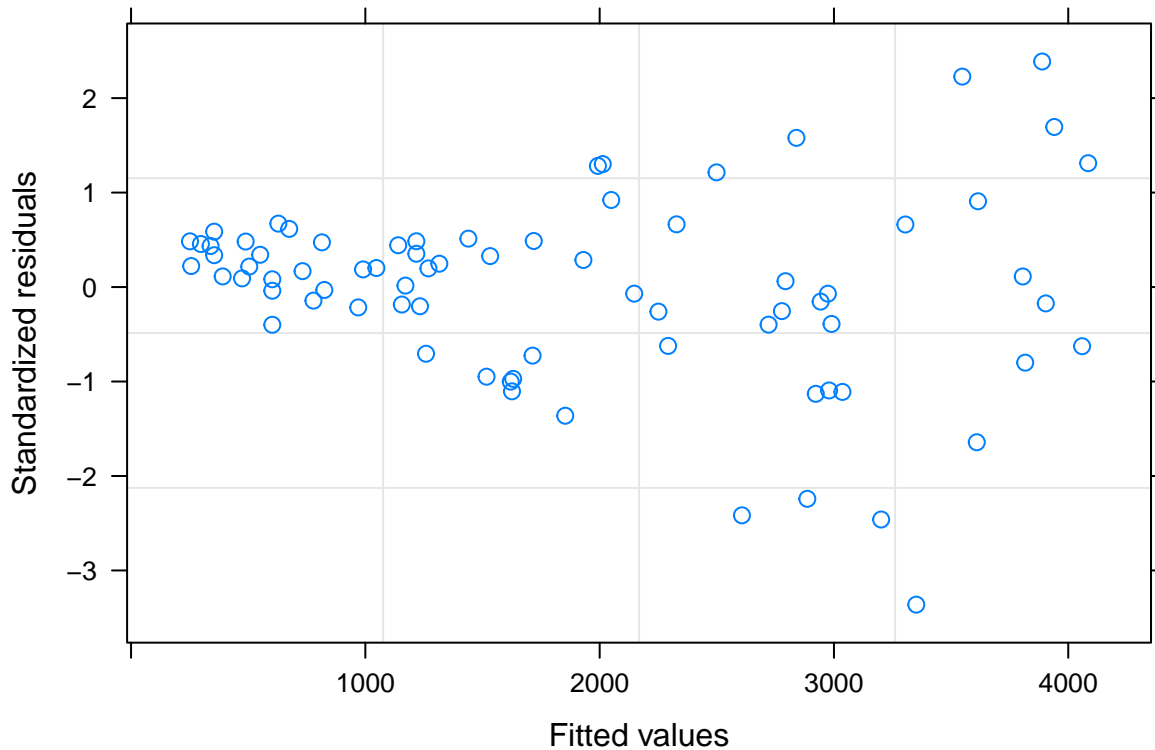
```
qqnorm(modelo_varFixed1, abline = c(0, 1)) #QQ plot
```



```
anova(modelo_varFixed1)
```

```
## Denom. DF: 73
##          numDF    F-value p-value
## (Intercept)      1 1390.9707 <.0001
## Hg_pelo          1  252.2731 <.0001
```

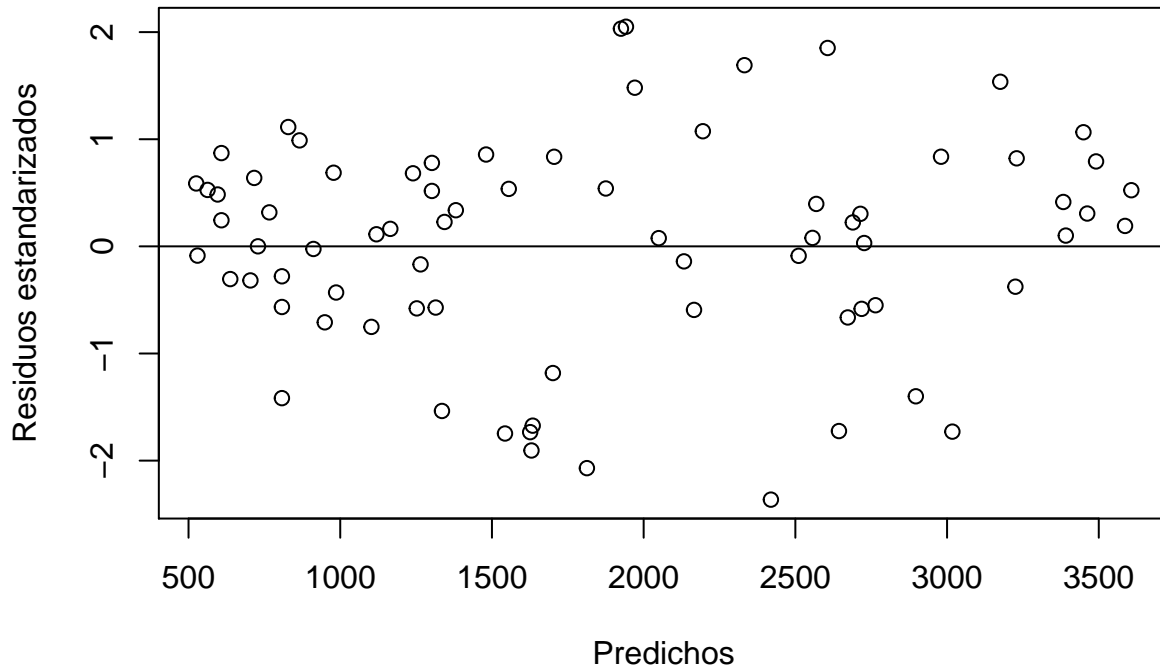
```
plot(modelo_varFixed1)
```



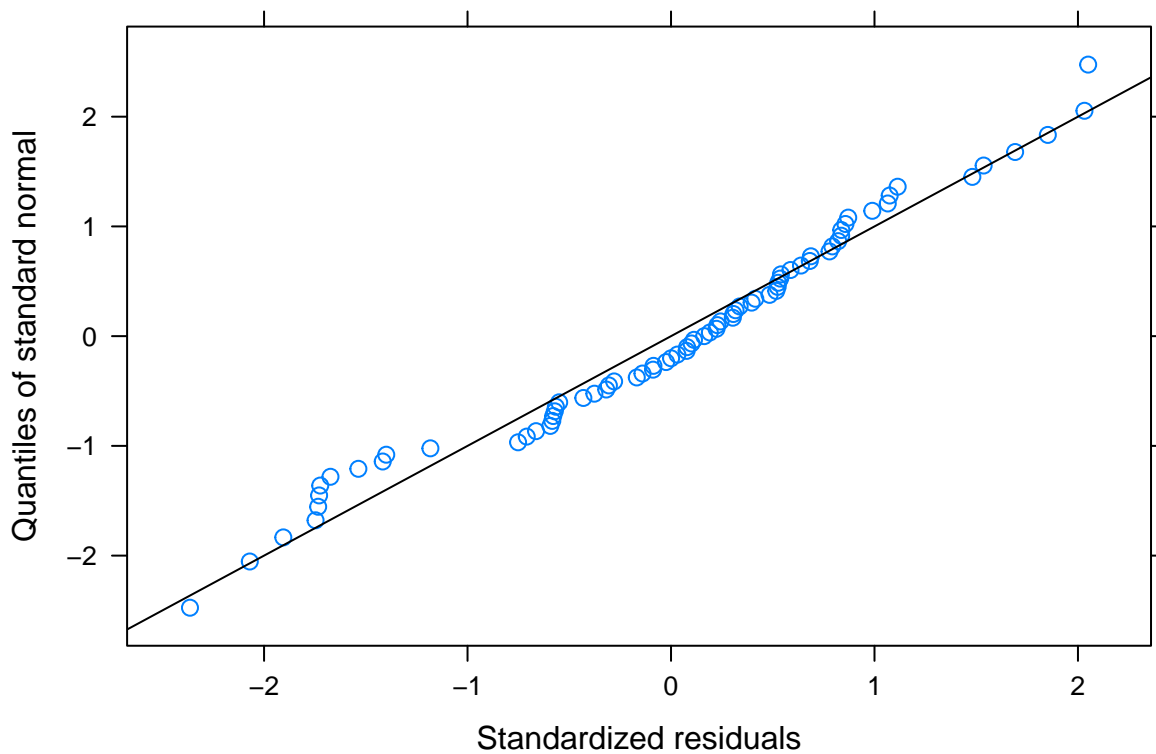
```
shapiro.test(r2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  r2
## W = 0.95116, p-value = 0.005779
#####
modelo_varFixed2 <- gls(act_bomba ~ Hg_pelo, weights = varFixed(~1/Hg_pelo),
  data = datos)
# Calculo de los residuos de pearson:
r3 <- residuals(modelo_varFixed2, type = "p") #guardamos los residuos de Pearson = estandarizados
# Calculo de los valores predichos por el modelo:
pred3 <- fitted(modelo_varFixed2)
plot(pred3, r3, xlab = "Predichos", ylab = "Residuos estandarizados",
  main = "Gráfico de dispersión de RE vs PRED")
abline(0, 0)
```

Gráfico de dispersión de RE vs PRED



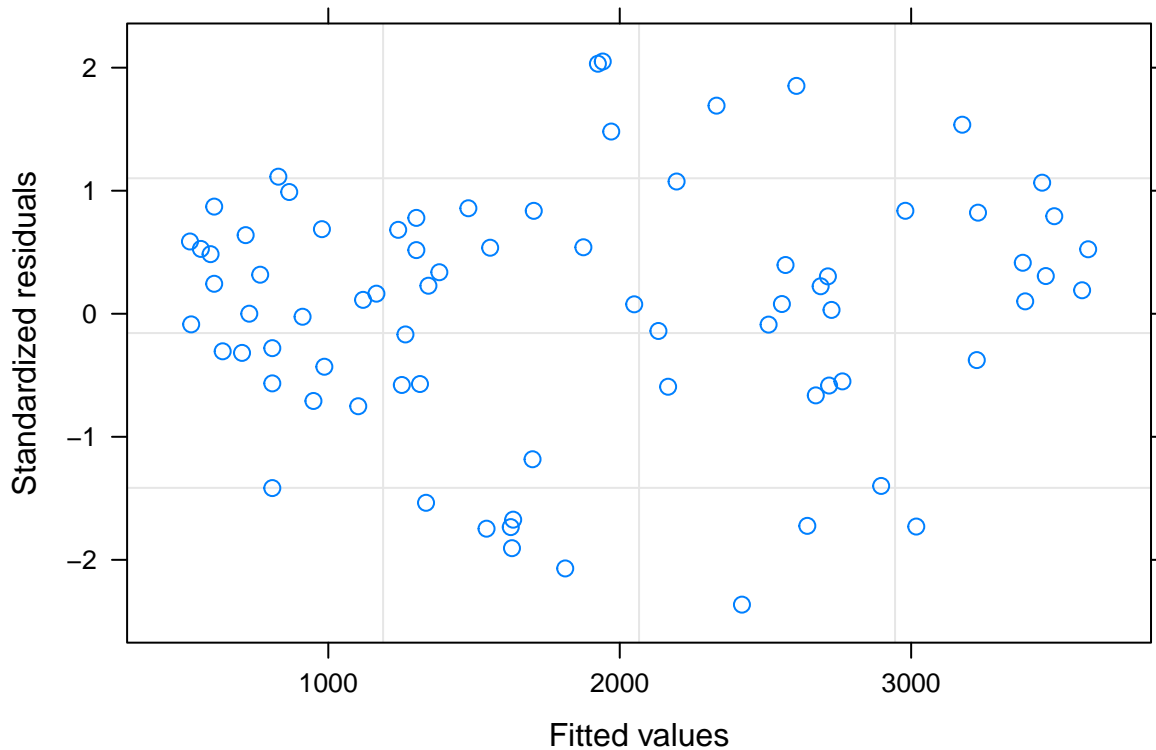
```
qqnorm(modelo_varFixed2, abline = c(0, 1)) #QQ plot
```



```
anova(modelo_varFixed2)
```

```
## Denom. DF: 73
##          numDF  F-value p-value
## (Intercept)    1 517.2723 <.0001
## Hg_pelo        1 144.7556 <.0001
```

```
plot(modelo_varFixed2)
```



```
shapiro.test(r3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  r3
## W = 0.96917, p-value = 0.06411
```

```
#####
```

```
modelo_varPower <- gls(act_bomba ~ Hg_pelo, weights = varPower(),
  data = datos)
```

```
# Calculo de los residuos de pearson:
```

```
r4 <- residuals(modelo_varPower, type = "p") #guardamos los residuos de Pearson = estandarizados
```

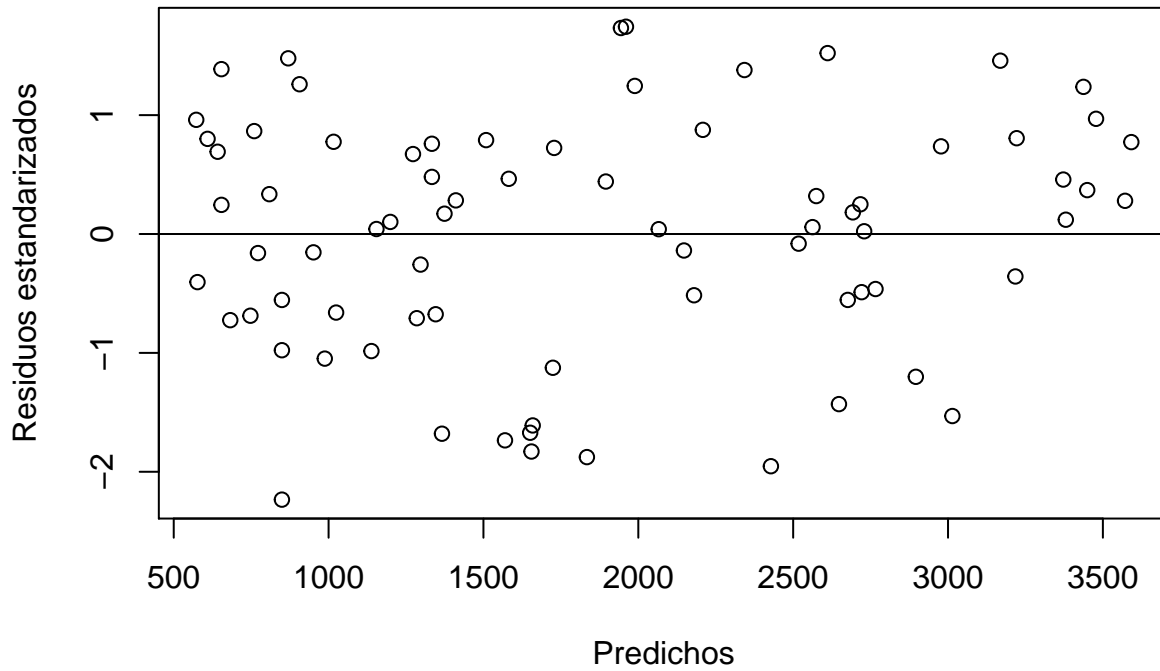
```
# Calculo de los valores predichos por el modelo:
```

```
pred4 <- fitted(modelo_varPower)
```

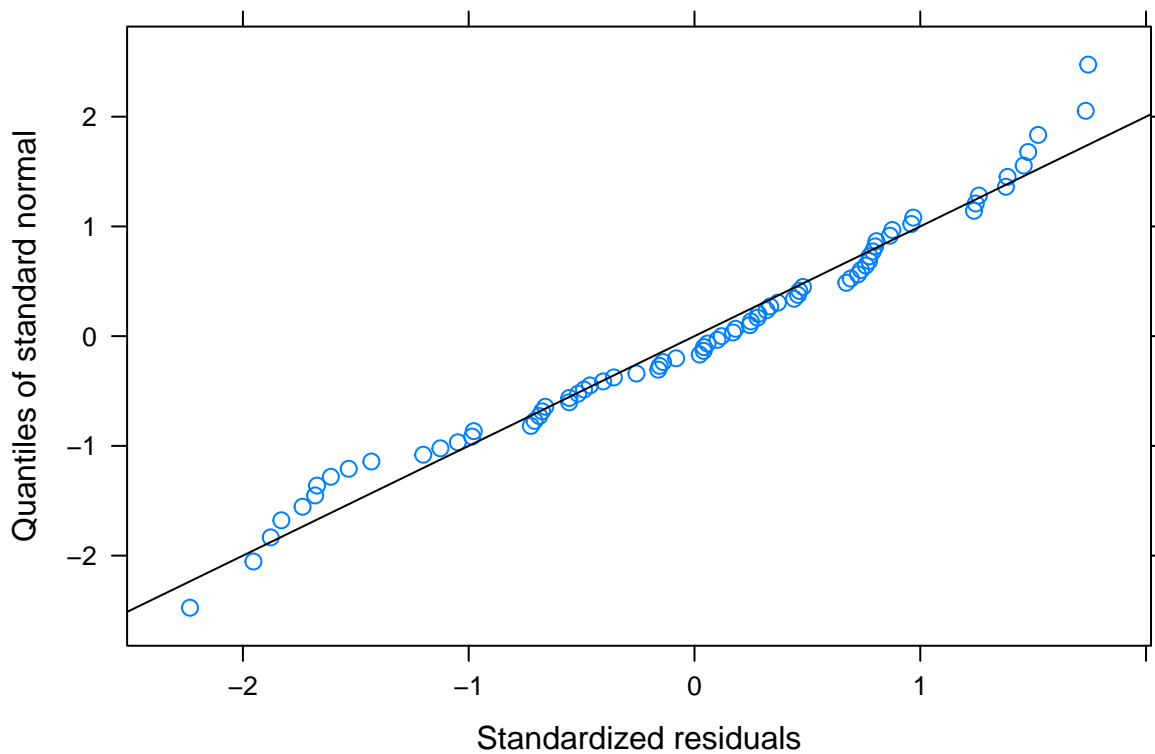
```
plot(pred4, r4, xlab = "Predichos", ylab = "Residuos estandarizados",
  main = "Gráfico de dispersión de RE vs PRED")
```

```
abline(0, 0)
```

Gráfico de dispersión de RE vs PRED



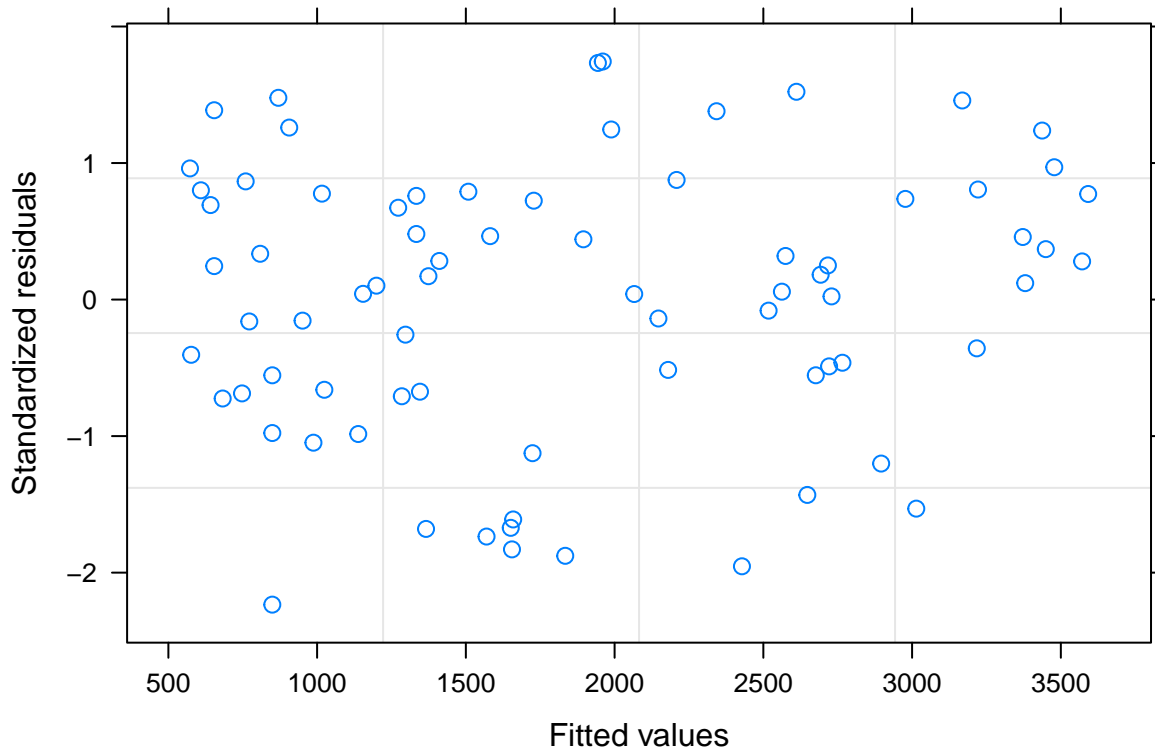
```
qqnorm(modelo_varPower, abline = c(0, 1)) #QQ plot
```



```
anova(modelo_varPower)
```

```
## Denom. DF: 73
##          numDF  F-value p-value
## (Intercept)    1 526.6071 <.0001
## Hg_pelo        1 175.1409 <.0001
```

```
plot(modelo_varPower)
```

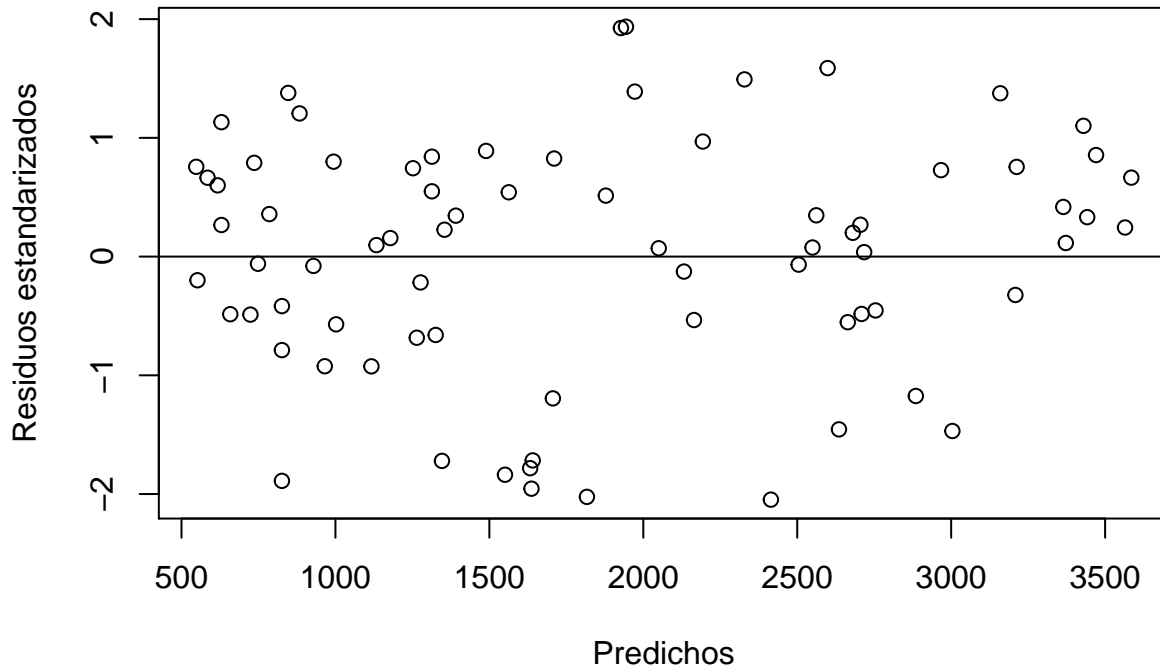


```
shapiro.test(r4)
```

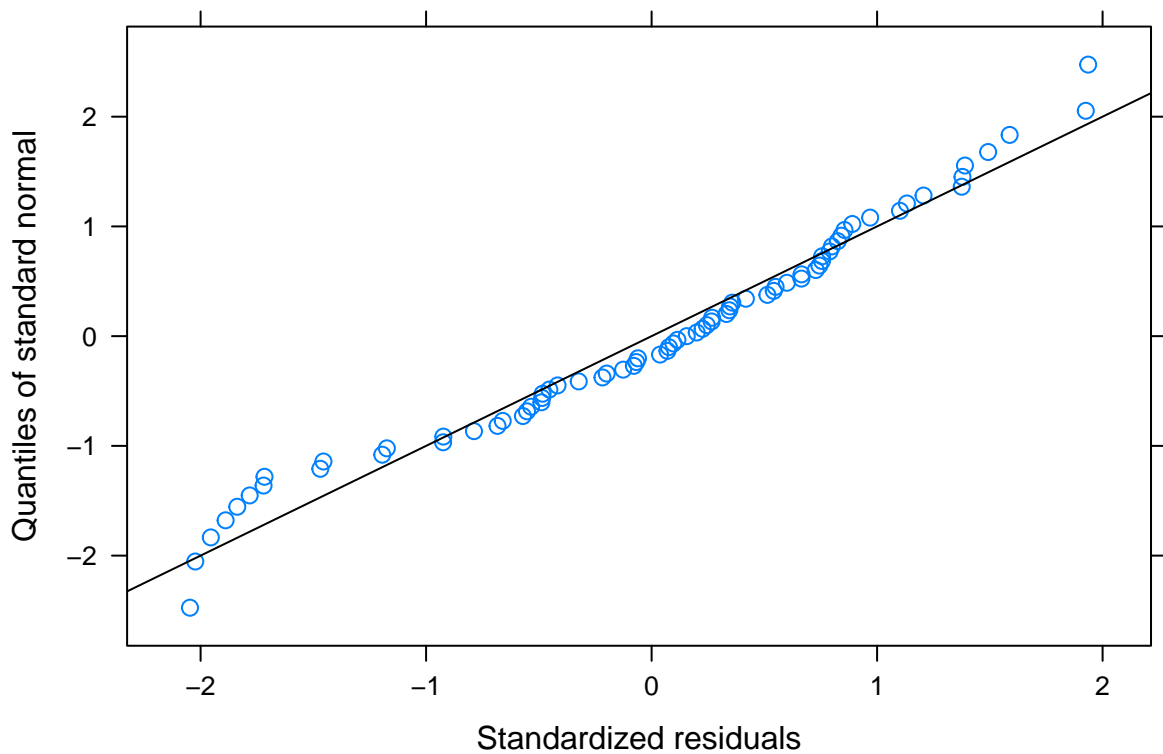
```
##
##  Shapiro-Wilk normality test
##
## data:  r4
## W = 0.96976, p-value = 0.06956
```

```
#####
modelo_varExp <- gls(act_bomba ~ Hg_pelo, weights = varExp(), data = datos)
# Calculo de los residuos de pearson:
r5 <- residuals(modelo_varExp, type = "p") #guardamos los residuos de Pearson = estandarizados
# Calculo de los valores predichos por el modelo:
pred5 <- fitted(modelo_varExp)
plot(pred5, r5, xlab = "Predichos", ylab = "Residuos estandarizados",
     main = "Gráfico de dispersión de RE vs PRED")
abline(0, 0)
```

Gráfico de dispersión de RE vs PRED



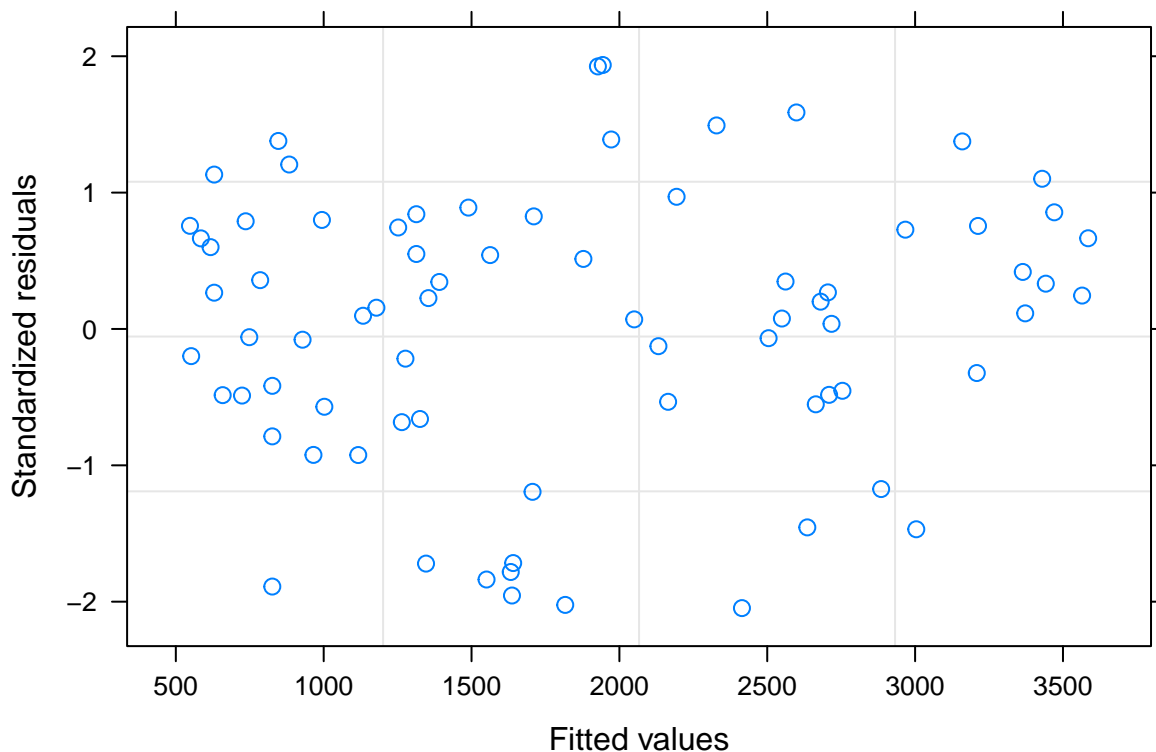
```
qqnorm(modelo_varExp, abline = c(0, 1)) #QQ plot
```




```
anova(modelo_varExp)
```

```
## Denom. DF: 73
##          numDF  F-value p-value
## (Intercept)    1 509.5035 <.0001
## Hg_pelo        1 151.0107 <.0001
```

```
plot(modelo_varExp)
```



```
shapiro.test(r5)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  r5
## W = 0.96643, p-value = 0.04389
```

```
AIC(modelo_varFixed1, modelo_varFixed2, modelo_varPower, modelo_varExp)
```

```
##          df      AIC
## modelo_varFixed1  3 1192.376
## modelo_varFixed2  3 1152.577
## modelo_varPower   4 1141.664
## modelo_varExp     4 1146.596
```

```
summary(modelo_varPower)
```

```
## Generalized least squares fit by REML
## Model: act_bomba ~ Hg_pelo
## Data:  dados
##      AIC      BIC   logLik
## 1141.664 1150.826 -566.832
##
```

```
## Variance function:
## Structure: Power of variance covariate
## Formula: ~fitted(.)
## Parameter estimates:
##      power
## 0.9215103
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept) 3807.536 210.39432  18.09714      0
## Hg_pelo     -406.904  30.74666 -13.23408      0
##
## Correlation:
##      (Intr)
## Hg_pelo -0.974
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -2.2344472 -0.6678858  0.1201694  0.7661635  1.7435899
##
## Residual standard error: 0.5849745
## Degrees of freedom: 75 total; 73 residual
```

```
# prueba de los residuos corregidos
e <- residuals(modelo_varPower, type = "p") # residuos
pre <- fitted(modelo_varPower) #predichos
res <- cbind(datos$act_bomba, datos$Hg_pelo, pre, e)
colnames(res) <- c("bomba", "hg", "Predichos", "Residuos")
head(res)
```

```
##      bomba  hg Predichos  Residuos
## 1 2136.09 5.11 1728.257 0.7241959
## 2 4446.25 0.53 3591.877 0.7731176
## 3 1574.30 1.95 3014.073 -1.5313736
## 4  604.38 6.00 1366.113 -1.6798969
## 5 1900.99 5.65 1508.529 0.7899321
## 6 2826.85 3.93 2208.404 0.8761178
```

Recordar que los residuos en varPower se calculan como:

$$e_{ij} = y_{ij} - \hat{y}_{ij} / \sqrt{(RSE * |\hat{y}_i|^{2*power})}$$

Entonces:

```
head(res)
```

```
##      bomba  hg Predichos  Residuos
## 1 2136.09 5.11 1728.257 0.7241959
## 2 4446.25 0.53 3591.877 0.7731176
## 3 1574.30 1.95 3014.073 -1.5313736
## 4  604.38 6.00 1366.113 -1.6798969
## 5 1900.99 5.65 1508.529 0.7899321
## 6 2826.85 3.93 2208.404 0.8761178
```

```
a = 2136.09 - 1728.257
b = (0.5849745^2) * (1728.257^(2 * 0.9215103))
a/sqrt(b)
```

```
## [1] 0.7241958
```

```
res[1, ]
```

```
##      bomba      hg  Predichos  Residuos
```

2136.0900000 5.1100000 1728.2570529 0.7241959

4- Sobre la base de su respuesta en (3) y el diseño de este estudio, ¿qué puede decir con respecto a la pregunta de investigación sobre si la exposición materna al mercurio medida por depósitos de mercurio en cabello afecta la actividad de la bomba de Ca en recién nacidos?