Análisis a los datos del laboratorio - Nanobiotech labs

Kaled Corona-Romero

09/Nov/2021

```
# Import libraries
library(readr)
library(tibble)
library(dplyr)
```

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

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

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

```
library(ggplot2) # For graphics
library(moments) # For normal test anaylisis
library(psych) # For plot histograms
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(corrplot) # For generate the heatmap
```

```
## corrplot 0.90 loaded
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

Introducción

A principios del semestre Ago-Dic 2021 se realizaró la síntesis de nuevas nanopartículas de ZnO dopado con elementos lantanoides. Se utilizó un modelo de perceptrón multicapa para encontrar un modelo que ajuste a los datos y nos permita predecir las capacidades antimicrobianas para nuevas configuraciones de los parámetros del compuesto. Los resultados del perceptrón multicapa (PMC) fueron inconcluyentes, y se

encontraron errores de coherencia en el mismo modelo. El modelo convergia a una solución demasiado rápido (overfitting). Es debido a esto que se realizará un análisis a detalle de los datos para ver a qué se debió ese problema.

Objetivos

En las conversaciones que tuve que Gil, llegamos a la conclusión que un buen aproximamiento a mejorar el modelo es mediante la normalización de la base de datos. En corridas anteriores, había normalizado solo la variable objetivo y declarado las variables predictoras como etiquetas y se obtuvo un buen resultado. Sin embargo, parece que esa aproximación está mal. Por lo que el objetivo principal de este análisis es normalizar la base de datos usando z-score (estandarización) y sacar una columna con la Min-max para comparar con cual normalización funciona mejor el modelo. Seguido de la normalización, se realizarán pruebas descriptivas de los datos, una prueba de normalidad y ver si se requiere una transformación.

Código y demás talacha

```
# Leemos las bases de datos para las bacterias Straphilococcus aureus y Escherichia coli
database.EC <- read_csv("/media/veracrypt2/bacterias_materiales/bacterias_nanomateriales_2021/data/csv/
```

```
## New names:
## * `` -> ...7
## * `` -> ...8
## * `` -> ...9
## * `` -> ...10
```

```
## Rows: 312 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): Time, Dope, a, c, Size, Abs
## lgl (4): ...7, ...8, ...9, ...10
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
database.SA <- read_csv("/media/veracrypt2/bacterias_materiales/bacterias_nanomateriales_2021/data/csv/
```

```
## New names:
## * `` -> ...7
## * `` -> ...8
## * `` -> ...9
## * `` -> ...10
```

```
## Rows: 312 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): Time, Dope, a, c, Size, Abs
## lgl (4): ...7, ...8, ...9, ...10
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Limpiamos la base de datos
# Esto es: Eliminar columnas sin información y asegurarnos que las columnas si pertenecen al tipo de da
```

```r
database.EC <- database.EC[,-c(7,8,9,10)]
database.SA <- database.SA[,-c(7,8,9,10)]
```

Realizamos una descripción de los datos

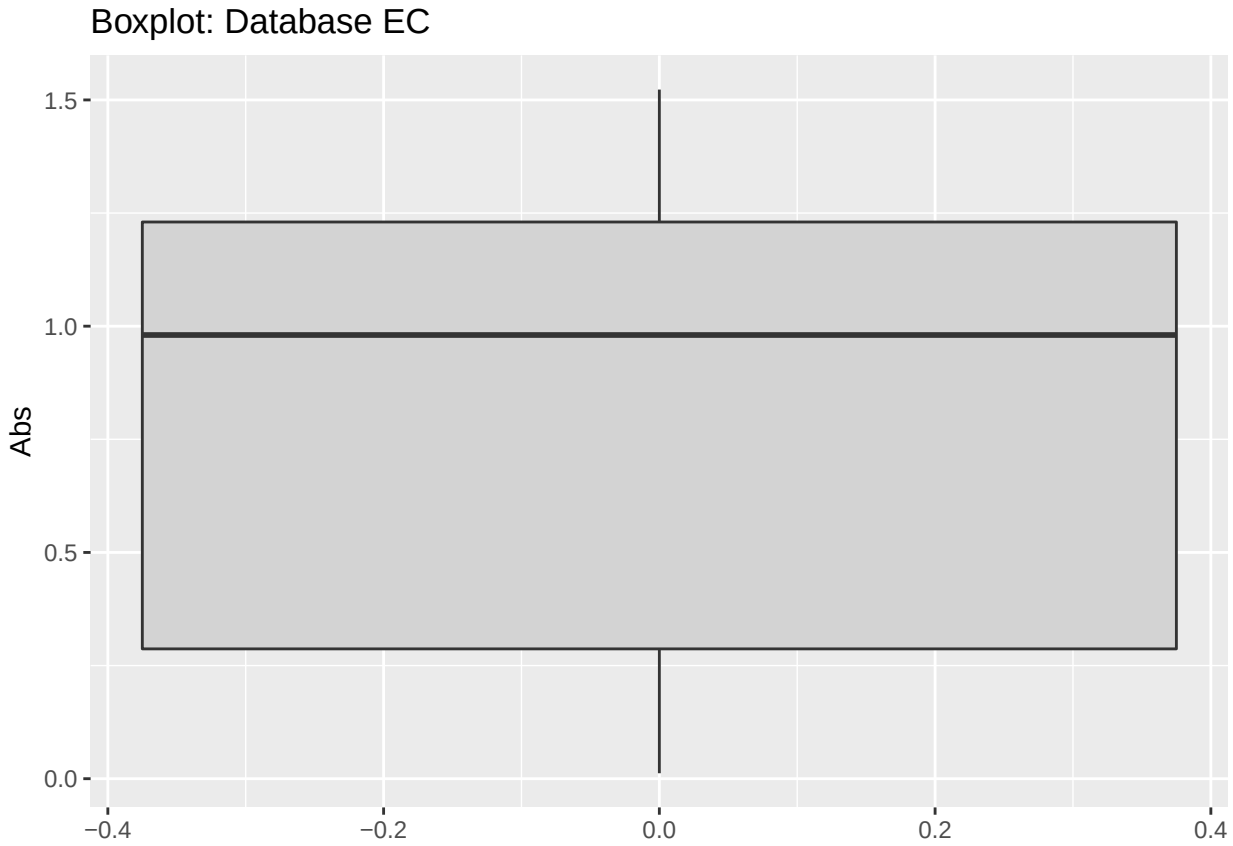
```r
# Primero la descripción básica para EC
summary(database.EC)
```

```
##       Time           Dope              a               c        
##  Min.   : 0.0   Min.   : 0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.: 3.5   1st Qu.: 2.500   1st Qu.:3.242   1st Qu.:5.197  
##  Median : 7.0   Median : 5.000   Median :3.245   Median :5.199  
##  Mean   : 7.0   Mean   : 5.385   Mean   :2.998   Mean   :4.803  
##  3rd Qu.:10.5   3rd Qu.:10.000   3rd Qu.:3.250   3rd Qu.:5.205  
##  Max.   :14.0   Max.   :10.000   Max.   :3.254   Max.   :5.210  
##       Size            Abs        
##  Min.   :0.000   Min.   :0.0121  
##  1st Qu.:6.820   1st Qu.:0.2869  
##  Median :7.180   Median :0.9805  
##  Mean   :7.025   Mean   :0.8301  
##  3rd Qu.:8.000   3rd Qu.:1.2303  
##  Max.   :8.440   Max.   :1.5228
```

Me parece que de los predictores no se puede sacar información relevante, salvo los máximos y mínimos de cada categoría, y la frecuencia de valores (la cual es homogenea en cada columna). Para la columna Abs, me parece que podría existir una gran varianza entre los datos, y a primera vista la distribución de los datos no parece ser homogenea.

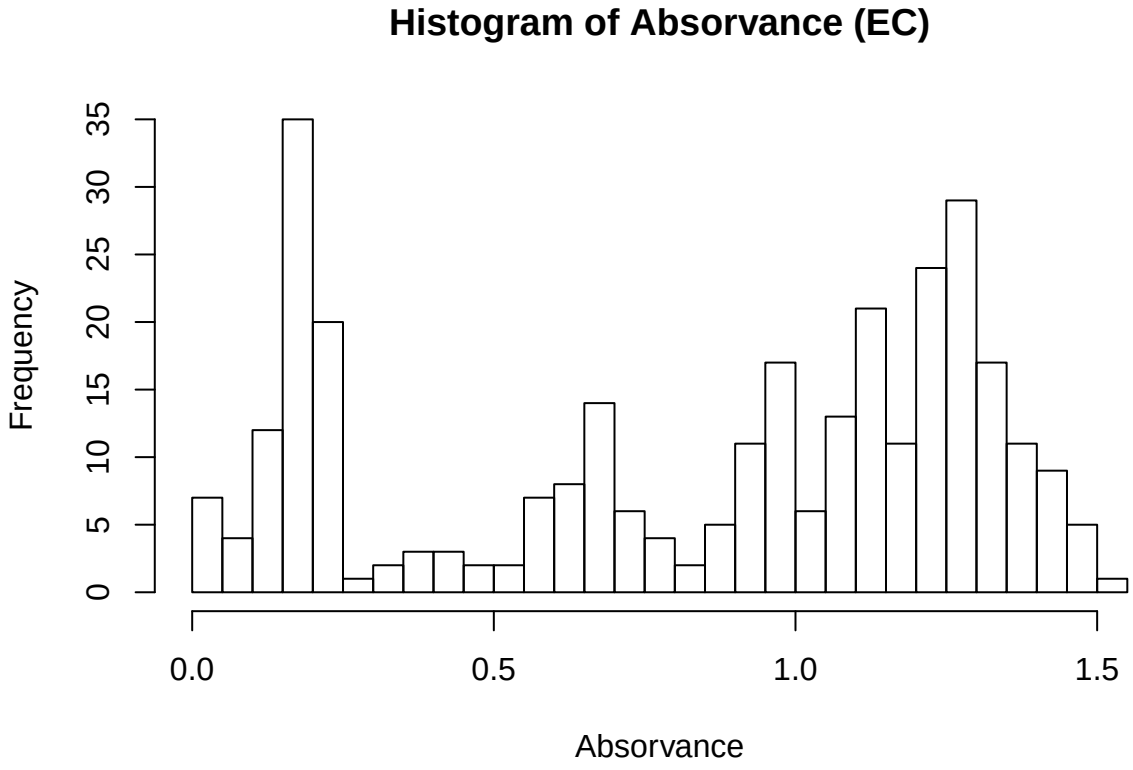
Revisemos una gráfica de cajas y bigotes para ver la localización de los puntos de interés

```r
ggplot(data = database.EC, aes(y = Abs)) +
  geom_boxplot(fill = "light gray") + ggtitle("Boxplot: Database EC")
```



Como se apreció con la descripción la mediana se encuentra muy cercana al tercer cuartil, existe una distancia grande entre el primer y el tercer cuartil, aún así, no parecen existir valores que se consideren como outliers. Ya por último para terminar con la descripción, analizaremos la distribución de los datos.

```
hist(database.EC[[6]], main = "Histogram of Absorvance (EC)", xlab = "Absorvance", breaks = 30)
```



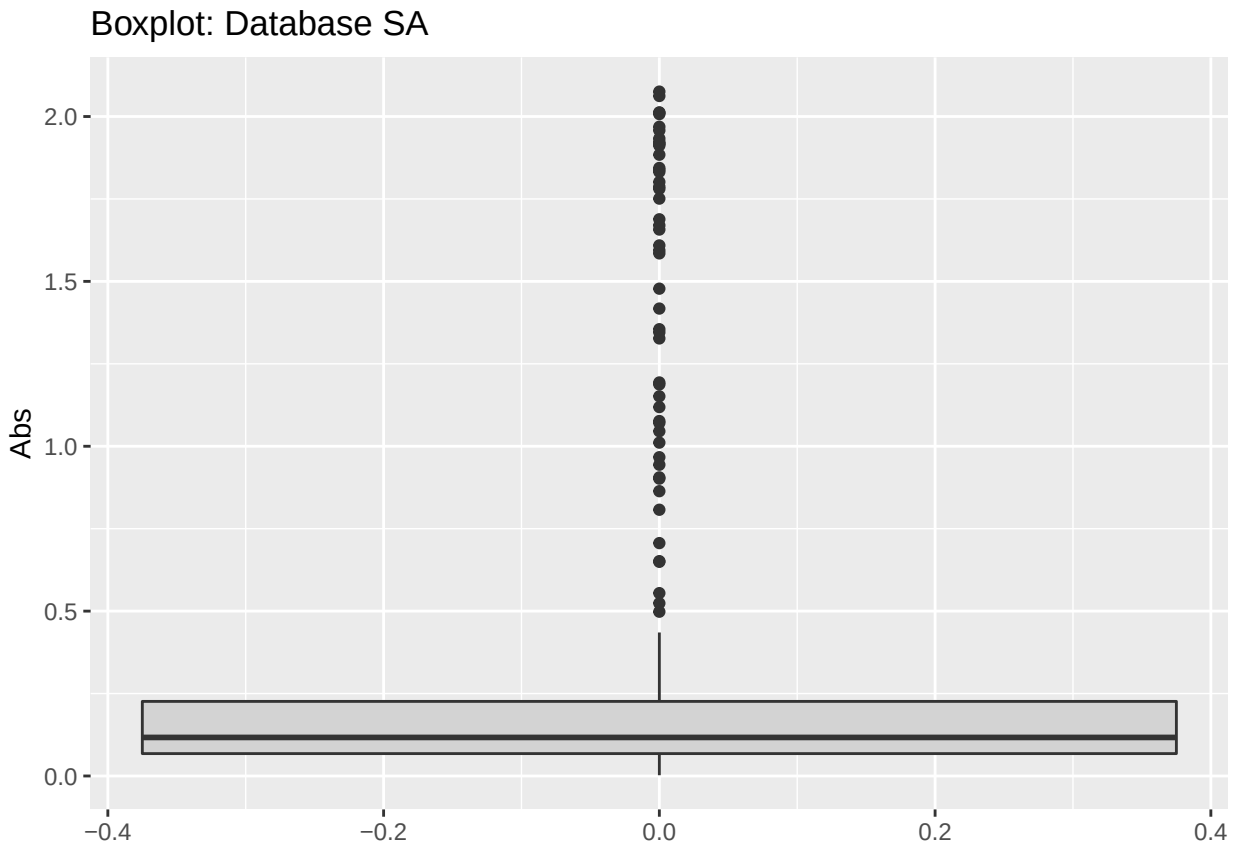
Con esto último, nos podemos dar una idea que los datos no parecen seguir una distribución normal, parecerían ser más una logarítmica. Ahora analizamos el otro conjunto de datos.

```
# Descripción para SA
summary(database.SA)
```

```
##       Time           Dope              a               c         
##  Min.   : 0.0   Min.   : 0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.: 3.5   1st Qu.: 2.500   1st Qu.:3.242   1st Qu.:5.197  
##  Median : 7.0   Median : 5.000   Median :3.245   Median :5.199  
##  Mean   : 7.0   Mean   : 5.385   Mean   :2.998   Mean   :4.803  
##  3rd Qu.:10.5   3rd Qu.:10.000   3rd Qu.:3.250   3rd Qu.:5.205  
##  Max.   :14.0   Max.   :10.000   Max.   :3.254   Max.   :5.210  
##       Size            Abs          
##  Min.   :0.000   Min.   :0.002267  
##  1st Qu.:6.820   1st Qu.:0.067825  
##  Median :7.180   Median :0.116850  
##  Mean   :7.025   Mean   :0.333790  
##  3rd Qu.:8.000   3rd Qu.:0.226192  
##  Max.   :8.440   Max.   :2.075633
```

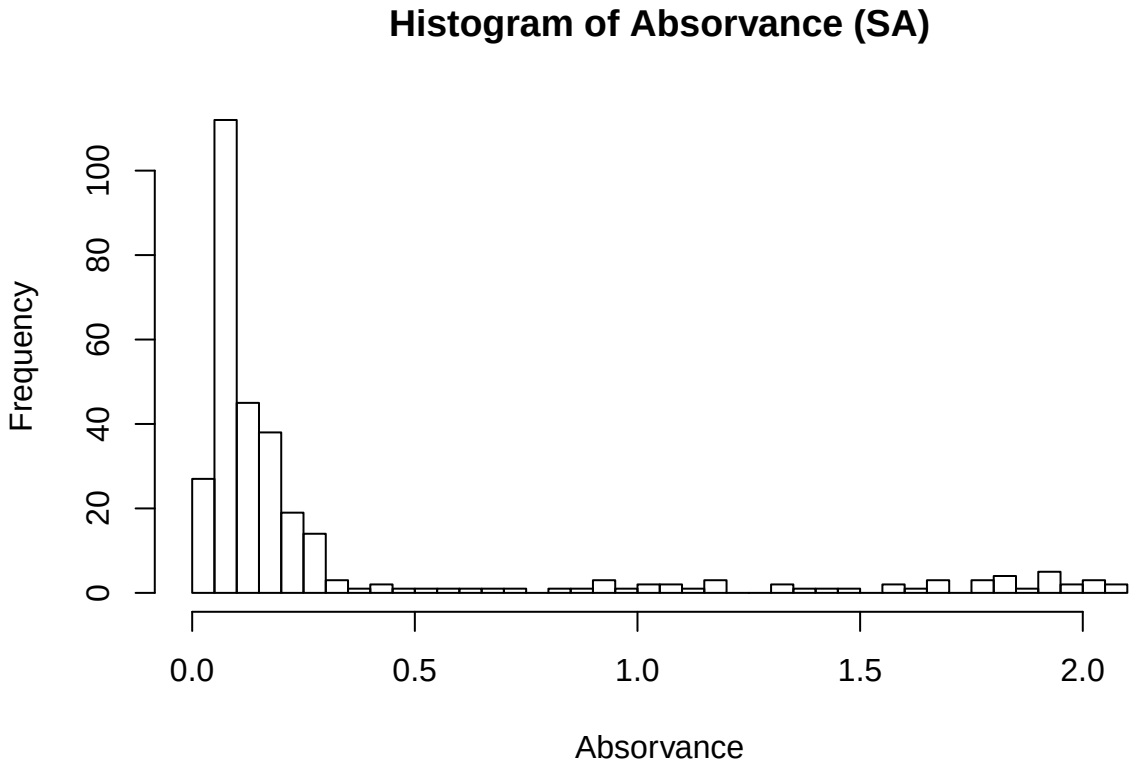
Aquí encontramos algo extraño en la columna de la absorvancia. Se escala muy rápido del 3rd cuartil al valor máximo. Procedemos a ver el gráfico de caja y bigotes.

```
ggplot(data = database.SA, aes(y = Abs)) +
  geom_boxplot(fill = "light gray") + ggtitle("Boxplot: Database SA")
```



Parece ser que existen varios valores que se consideran outliers. Falta analisar cómo es el comportamiento de los datos analizando un solo tipo de material. Antes de eso, pasaremos a revisar el histograma.

```
hist(database.SA[[6]], main = "Histogram of Absorvance (SA)", xlab = "Absorvance", breaks = 30)
```



El his-

tograma verifica la existencia de valores poco frecuentes. Aún así, me parece que podrían estar bien. Tendremos que analizar esos valores para los cuales la absorvancia es mayor a 0.5.

Hipótesis

Los outliers son generados porque se está tomando encuenta todos los materiales. Si se toman por separado, no deberían generar outliers.

```
# Convertimos la base de datos a una estructura tibble (como un dataframe en pandas)
database.SA <- tibble(database.SA)
database.EC <- tibble(database.EC)
```

```
# Separamos los datos por sus respectivos nanomateriales SA
separado.materiales <- database.SA %>% group_by(a)
```

```
separado.materiales %>% summarise(
  Media = mean(Abs),
  Varianza = var(Abs),
  SD = sd(Abs),
  MAX = max(Abs),
  Min = min(Abs)
)
```

```
## # A tibble: 5 x 6
##       a Media Varianza     SD   MAX     Min
##   <dbl> <dbl>    <dbl>  <dbl> <dbl>   <dbl>
## 1  0    1.39   0.475   0.689  2.08  0.0450
## 2  3.24 0.120  0.00521 0.0722 0.364 0.0121
## 3  3.24 0.474  0.458   0.677  1.97  0.00903
## 4  3.25 0.276  0.0987  0.314  1.19  0.0180
## 5  3.25 0.113  0.00406 0.0638 0.328 0.00227
```

La codificación la "a" un parámetro de celda para las nanopartículas es la siguiente: * 3.25 -> ZnEr * 3.242 -> ZnNd * 3.254 -> ZnSm * 3.245 -> ZnCe * 0.0 -> Control

Basado en los datos anteriores, el ZnEr y el ZnCe muestran una desviación estandar más grande que su media, además de valores máximos bastante grandes.

Lo que me hace pensar que se debe a que no estoy tomando encuenta que estos datos se encuentran juntos los 3 tratamientos. Y en las gráficas de tiempo y ratio de supervivencia de las basterias (gráfico proporcionado por el lab) se muestra una diferencia significativa del tratamiento C1 a los tratamientos C2 y C3.

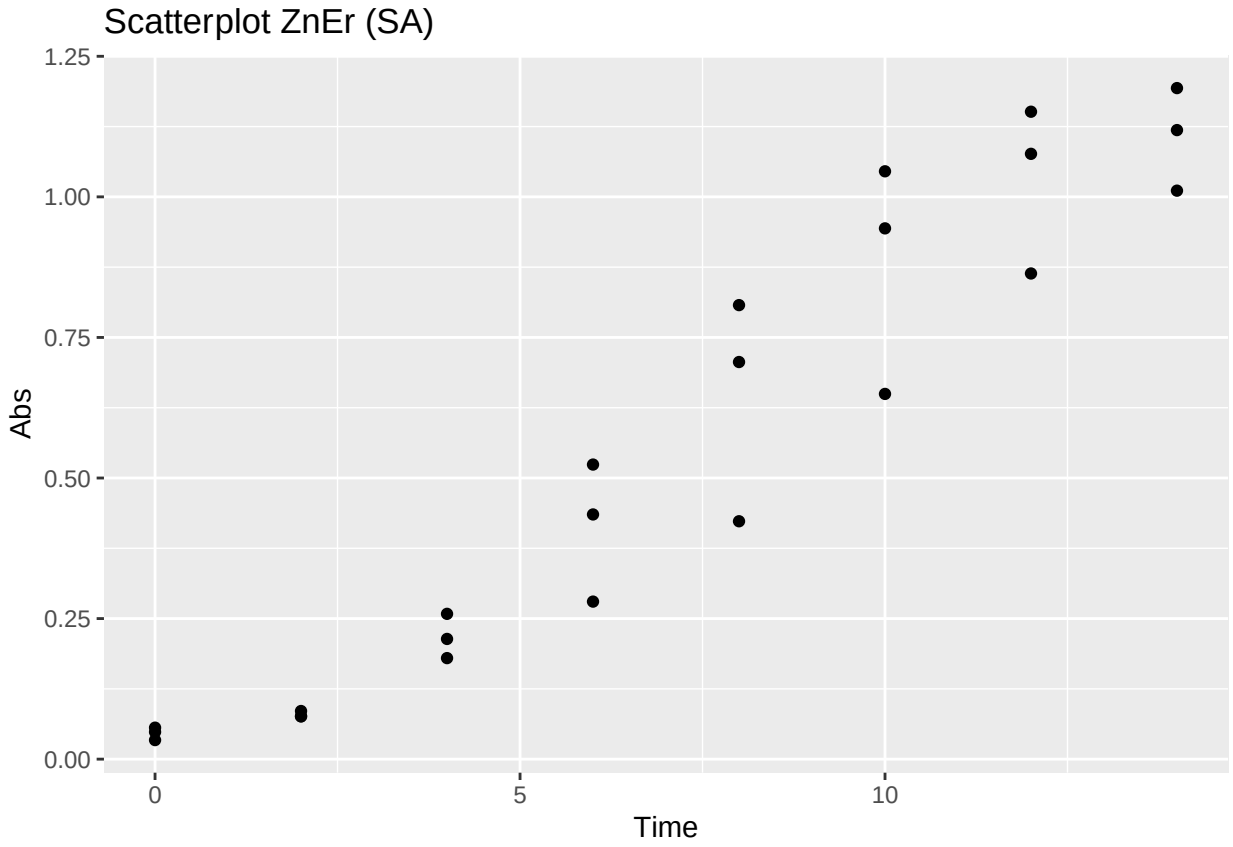
Codificación del tratamiento (Dope): * C1 -> 2.5 * C2 -> 5 * C3 -> 10

```
# Calculamos los siguientes estadísticcos para la configuración ZnEr C1
separado.materiales %>% filter(Dope == 2.5, a == 3.250) %>% summarise(
  Media = mean(Abs),
  Varianza = var(Abs),
  SD = sd(Abs),
  MAX = max(Abs),
  Min = min(Abs)
)
```

```
## # A tibble: 1 x 6
##       a Media Varianza    SD   MAX    Min
##   <dbl> <dbl>    <dbl> <dbl> <dbl>  <dbl>
## 1  3.25 0.552    0.176 0.420  1.19 0.0339
```

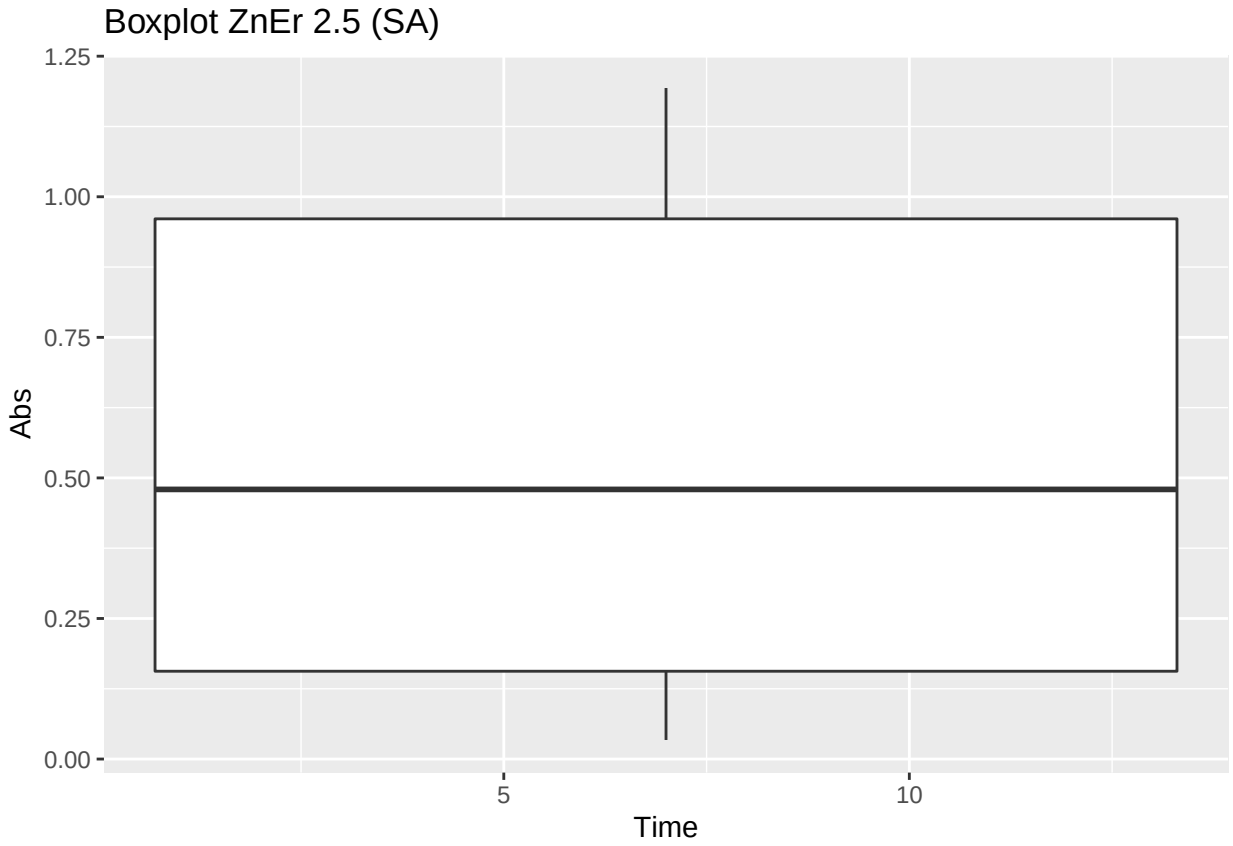
```r
# Guardamos el filtro en una variable y luego la graficamos
var.temp <- separado.materiales %>% filter(Dope == 2.5, a == 3.250)
```

```r
ggplot(data = var.temp, aes(x=Time, y = Abs)) + geom_point() + ggtitle("Scatterplot ZnEr (SA)")
```




```r
ggplot(data = var.temp, aes(x=Time, y = Abs)) + geom_boxplot() + ggtitle("Boxplot ZnEr 2.5 (SA)")
```

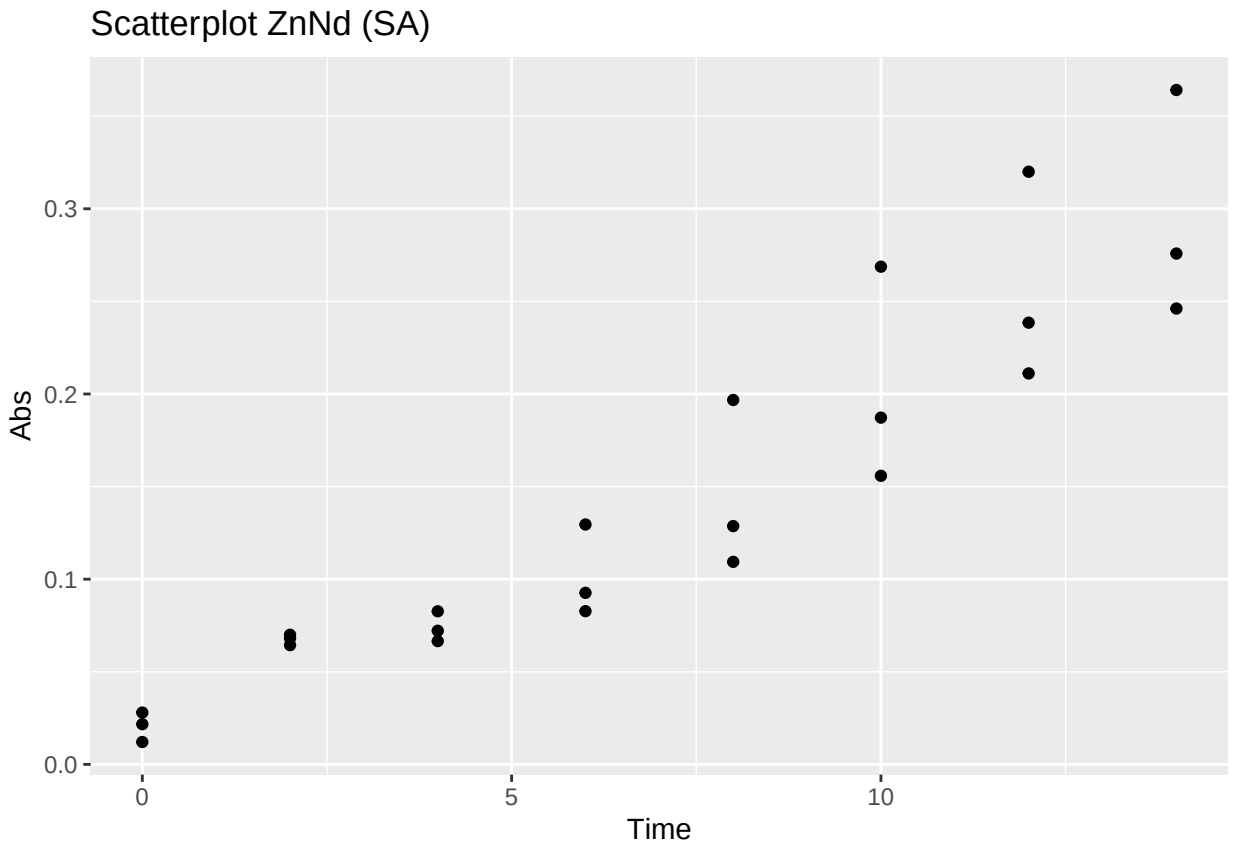
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Aquí evidentemente existen outliers. Parece se que mientras el tiempo aumenta, los valores para la absorvancia se dispersan. También pasará para los demás compuestos con C1?

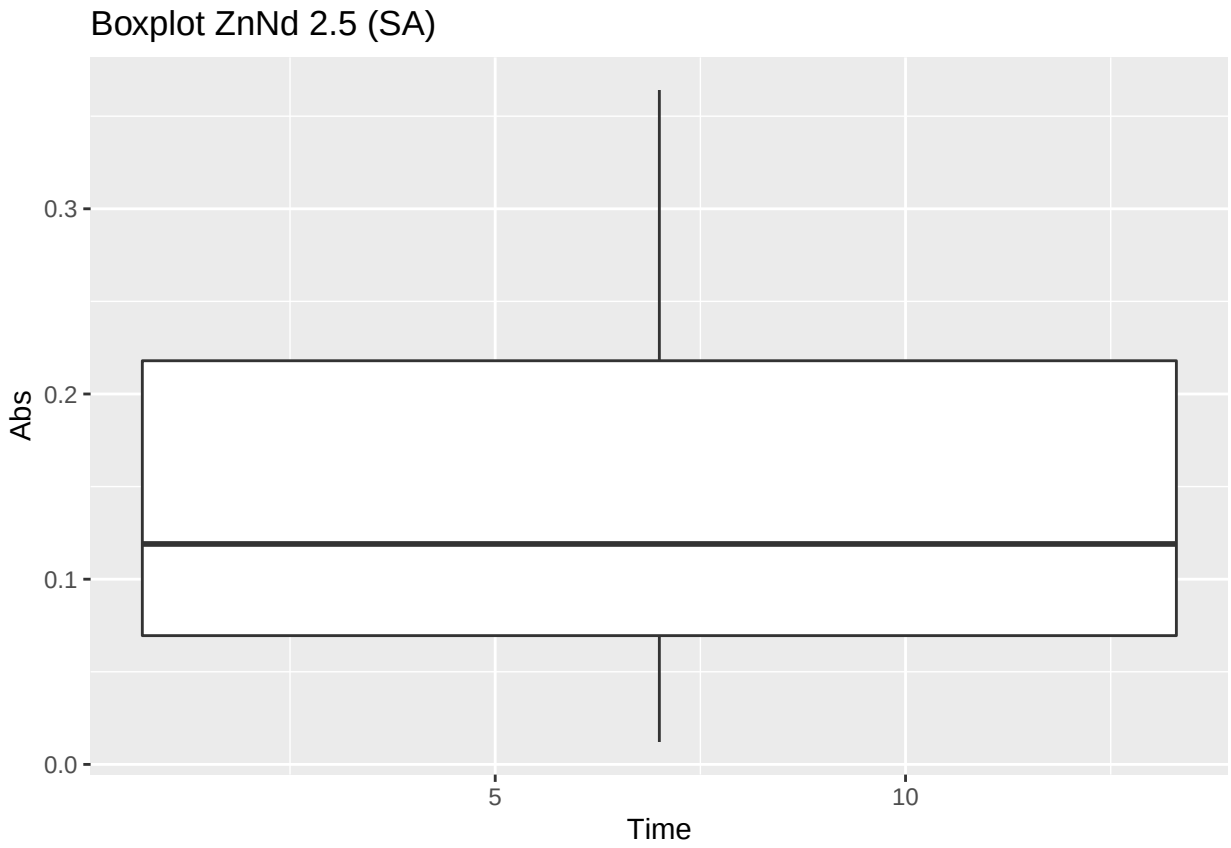
```
# Guardamos el filtro en una variable y luego la graficamos
var.temp <- separado.materiales %>% filter(Dope == 2.5, a == 3.242)
```

```
ggplot(data = var.temp, aes(x=Time, y = Abs))  + geom_point() + ggtitle("Scatterplot ZnNd (SA)")
```




```
ggplot(data = var.temp, aes(x=Time, y = Abs)) + geom_boxplot() + ggtitle("Boxplot ZnNd 2.5 (SA)")
```

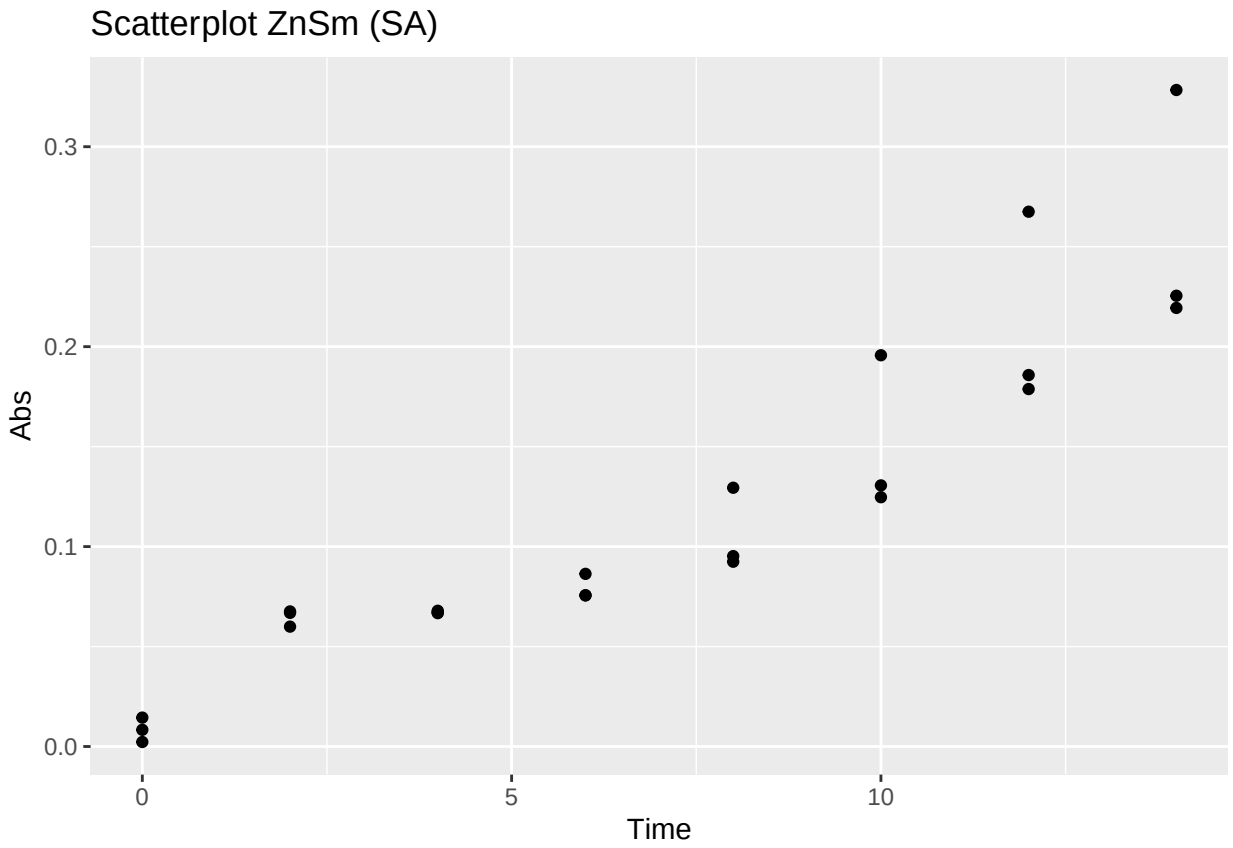
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Parace que también existe mucha dispersión. Seguiremos analizando los demás.

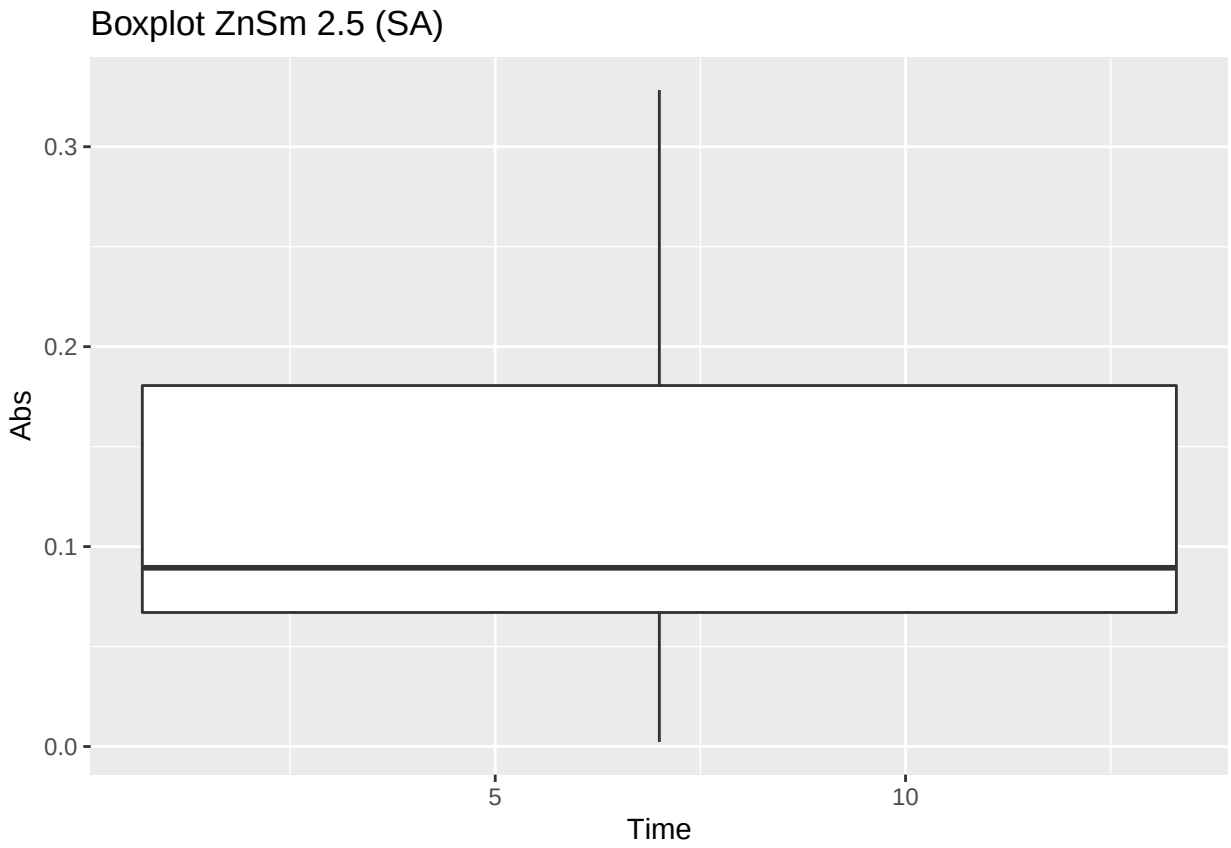
```
# Guardamos el filtro en una variable y luego la graficamos
var.temp <- separado.materiales %>% filter(Dope == 2.5, a == 3.254)
```

```
ggplot(data = var.temp, aes(x=Time, y = Abs)) + geom_point() + ggtitle("Scatterplot ZnSm (SA)")
```




```
ggplot(data = var.temp, aes(x=Time, y = Abs)) + geom_boxplot() + ggtitle("Boxplot ZnSm 2.5 (SA)")
```

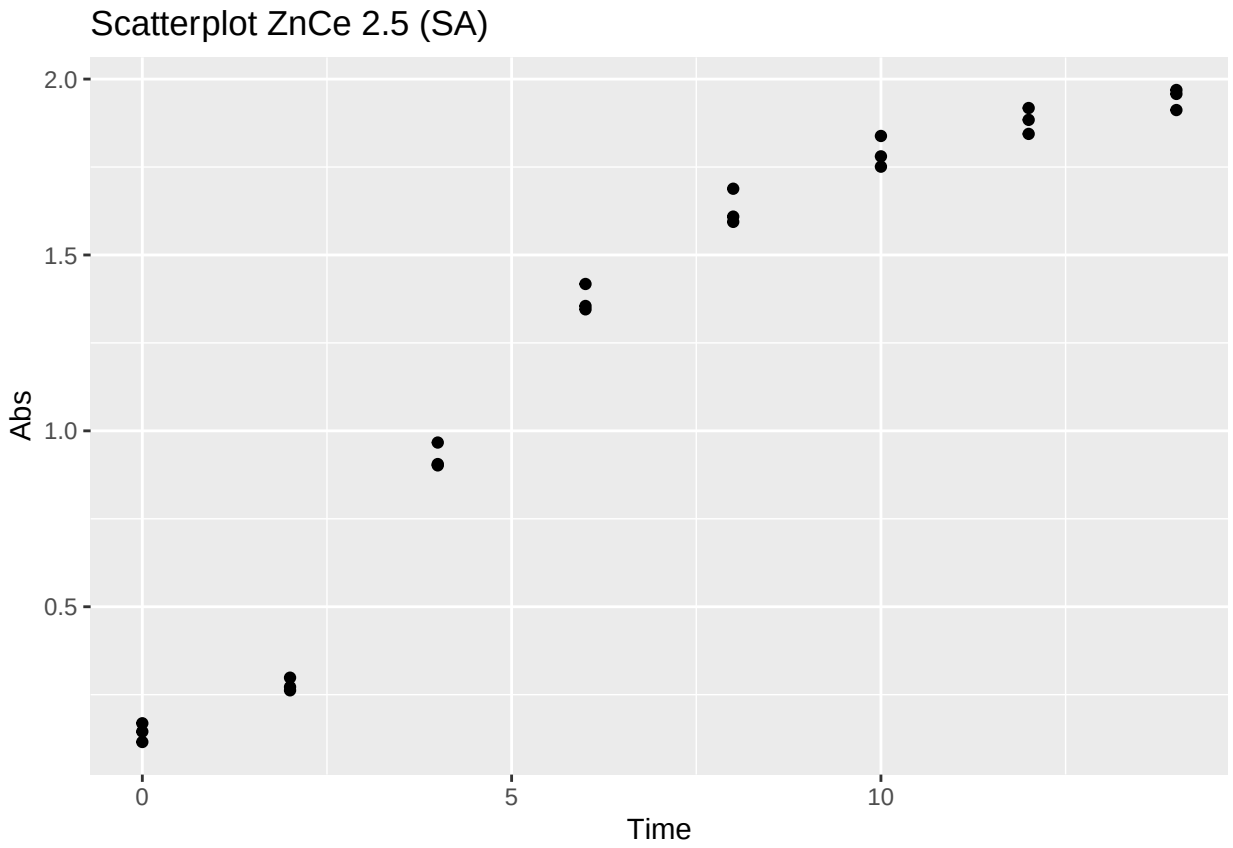
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Aquí los datos ya se muestran más inclinados hacia un valor.

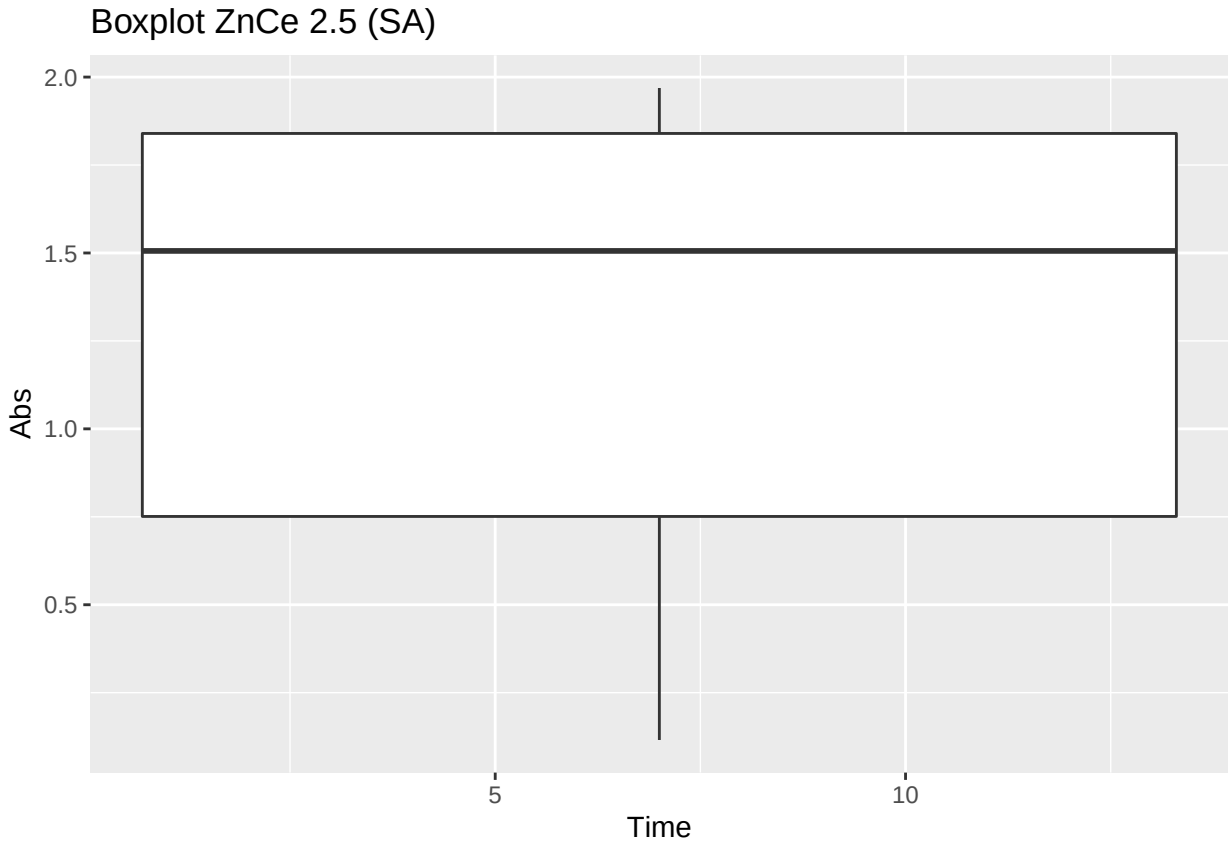
```
# Guardamos el filtro en una variable y luego la graficamos
var.temp <- separado.materiales %>% filter(Dope == 2.5, a == 3.245)
```

```
ggplot(data = var.temp, aes(x=Time, y = Abs)) + geom_point() + ggtitle("Scatterplot ZnCe 2.5 (SA)")
```




```
ggplot(data = var.temp, aes(x=Time, y = Abs)) + geom_boxplot() + ggtitle("Boxplot ZnCe 2.5 (SA)")
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Resultados

Se analizaron los datos para el tratamiento C1, que en las gráficas dadas por el laboratorio eran los datos más alejados de los demás tratamientos. Se descubrió que en efecto los datos de C1 eran los que estaban generando ruido a una escala global (Si se tomaban todos los tratamientos, ya que al tener medias muy diferentes, se les consideraba outliers), pero que si se analizaban en específico, no mostraban rastro de outliers.

Antes de continuar con lo demás, me quedó una inquietud en saber cuáles son las configuraciones de parámatros para los cuales la absorvancia es mayor a 0.5 en SA.

```
database.SA %>% filter(Abs >=  0.5)
```

```
## # A tibble: 50 x 6
##     Time  Dope     a     c  Size   Abs
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1     2   0    0     0     0    0.554
##  2     2   0    0     0     0    0.652
##  3     4   2.5  3.24  5.20  8    0.967
##  4     4   2.5  3.24  5.20  8    0.906
##  5     4   2.5  3.24  5.20  8    0.902
##  6     4   0    0     0     0    1.19
##  7     4   0    0     0     0    1.07
##  8     4   0    0     0     0    1.33
##  9     6   2.5  3.24  5.20  8    1.42
## 10     6   2.5  3.25  5.20  7.18 0.524
## # ... with 40 more rows
```

De esta información, podemos extraer que el conjunto se conforma solo de los tratamientos 0 (control) y C1 (2.5). Y también, viendo las gráficas de ratio de supervivencia puedo deducir que a mayor absorvancia mayor

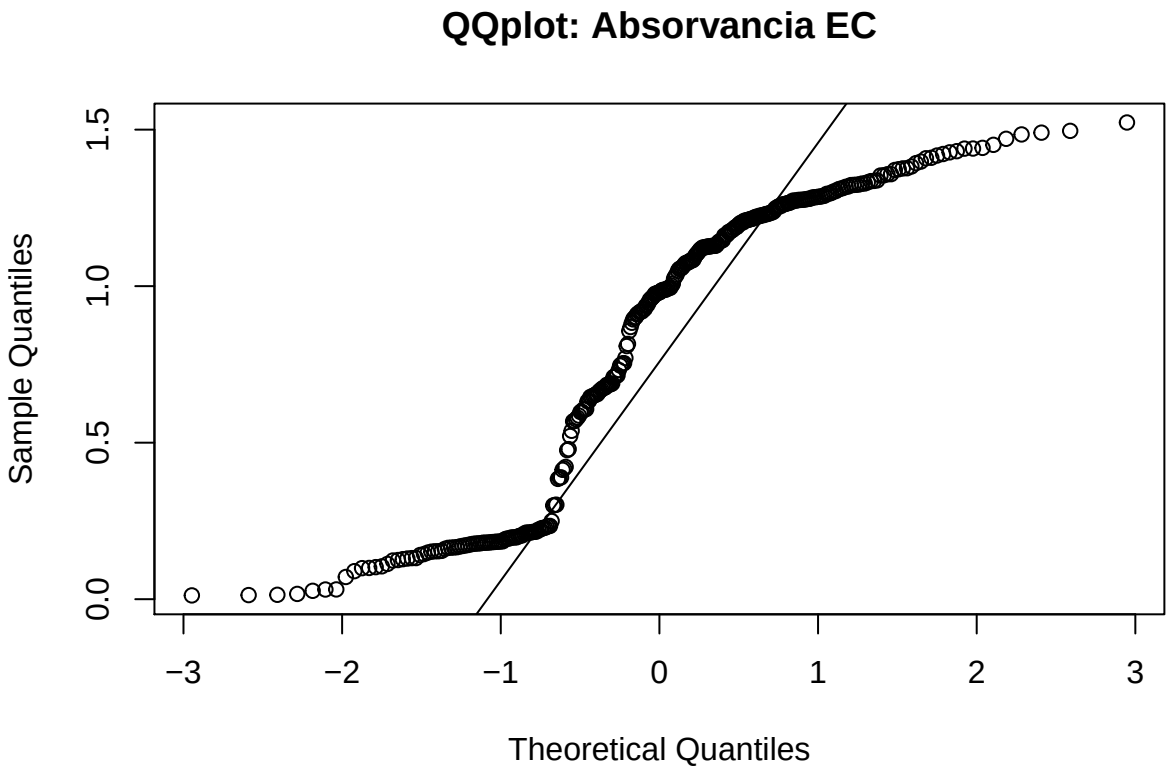
es el grado de supervivencia de las bacterias (SA). Por lo que se podría señalar que el tratamiento C1 es ineficaz contra la bacteria SA.

Normalización

Ahora pasamos a la otra sección del análisis. Esto es normalizar los datos para obtener un mejor rendimiento del modelo.

Primero verificamos que los datos que tenemos no son normales. Esto lo podemos hacer mediante una gráfica qqnorm

```
qqnorm(database.EC[[6]], main = "QQplot: Absorvancia EC")
qqline(database.EC[[6]])
```



En lo que consierne a mi opinión, la gráfica no muestra claramente una tendencia normal. Creo que los datos podrían ser no normales. Para ello vamos a realizar un test de normalidad.

```
# Test de normalidad para la absorvancia sin normalizar
shapiro.test(database.EC$Abs)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  database.EC$Abs
## W = 0.88679, p-value = 1.901e-14
```

Dado que el p-value es muy cercano a cero, aceptamos la hipótesis alternativa de que los datos no son normales.

Probaremos entonces a normaizarlos con los métodos de estandarización y de Min-max y ver cuál da mejores resultados.

Estandaarización

```
# Formula a seguir: (valor - media) / desviación estandar
# Sacar medias
mean.vector.SA <- c()
for (index in 1:6){
  mean.value <- mean(database.SA[[index]])
  mean.vector.SA <- append(mean.vector.SA, mean.value)
}

mean.vector.EC <- c()
for (index in 1:6){
  mean.value <- mean(database.EC[[index]])
  mean.vector.EC <- append(mean.vector.EC, mean.value)
}

# Calcular desviación estandar
sd.vector.SA <- c()
for (index in 1:6){
  sd.value <- sd(database.SA[[index]])
  sd.vector.SA <- append(sd.vector.SA, sd.value)
}

sd.vector.EC <- c()
for (index in 1:6){
  sd.value <- sd(database.EC[[index]])
  sd.vector.EC <- append(sd.vector.EC, sd.value)
}
```

```
# Me aseguro que ambos datasets sean estructuras de tibble
is_tibble(database.SA)
```

```
## [1] TRUE
```

```
is_tibble(database.EC)
```

```
## [1] TRUE
```

```
# Normalizamos toda la base de datos
for (index in 1:6){
  # Normalizar
  database.EC[,index] <- (database.EC[,index] - mean.vector.EC[index]) / sd.vector.EC[index]
}
```

Realizamos el mismo proceso para la otra base de datos.

```
# Realizamos una copia
database.SA.norm <- database.SA
```

```
# Normalizamos toda la base de datos
for (index in 1:6){
  # Normalizar
  database.SA.norm[,index] <- (database.SA.norm[,index] - mean.vector.SA[index]) / sd.vector.SA[index]
}
```

Prueba de normalidad

Volvemos a realizar una prueba de normalidad a las bases de datos ya estandarizadas

```
# Test de normalidad para la absorvancia estandarizada
shapiro.test(database.SA.norm$Abs)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  database.SA.norm$Abs
## W = 0.57149, p-value < 2.2e-16
```

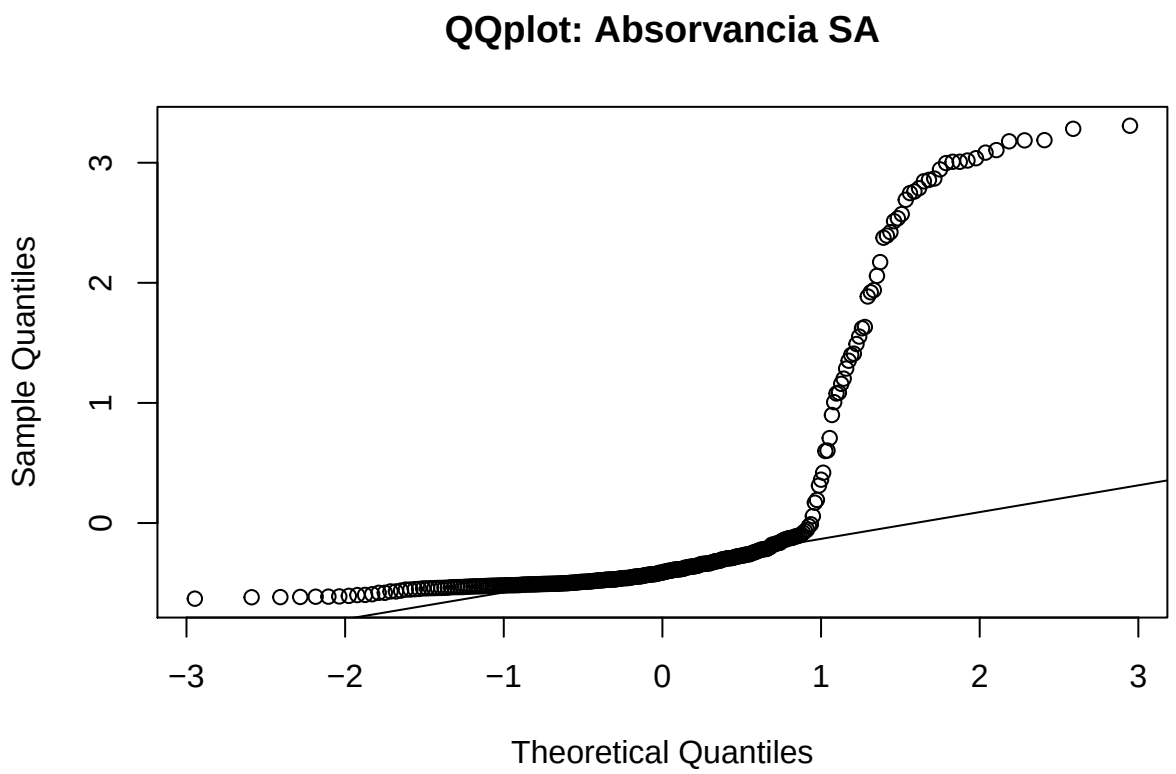
```
# Test de normalidad para la absorvancia sin normalizar
skewness(database.SA.norm$Abs)
```

```
## [1] 2.205208
```

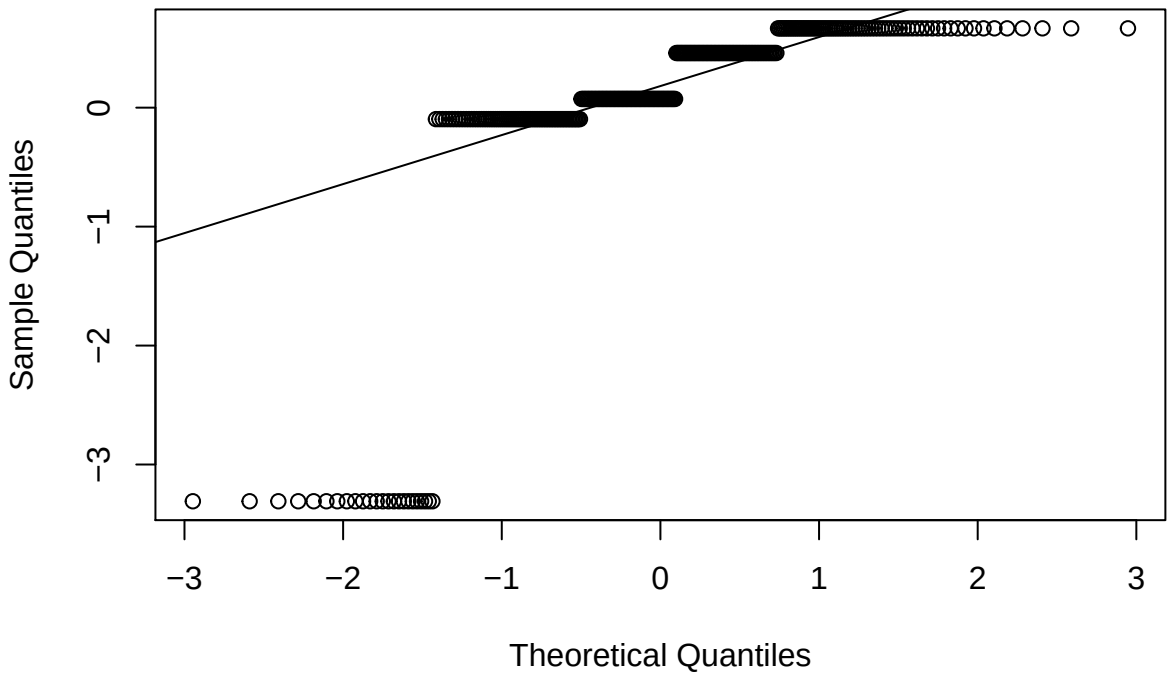
```
agostino.test(database.SA.norm$Abs)
```

```
##
##  D'Agostino skewness test
##
## data:  database.SA.norm$Abs
## skew = 2.2052, z = 10.5190, p-value < 2.2e-16
## alternative hypothesis: data have a skewness
```

```
qqnorm(database.SA.norm[[6]], main = "QQplot: Absorvancia SA")
qqline(database.SA.norm[[6]])
```




```
qqnorm(database.SA.norm[[5]], main = "QQplot: a SA")
qqline(database.SA.norm[[5]])
```




```
# Test de normalidad para la absorvancia estandarizada
shapiro.test(database.EC$Abs)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  database.EC$Abs
## W = 0.88679, p-value = 1.901e-14
```

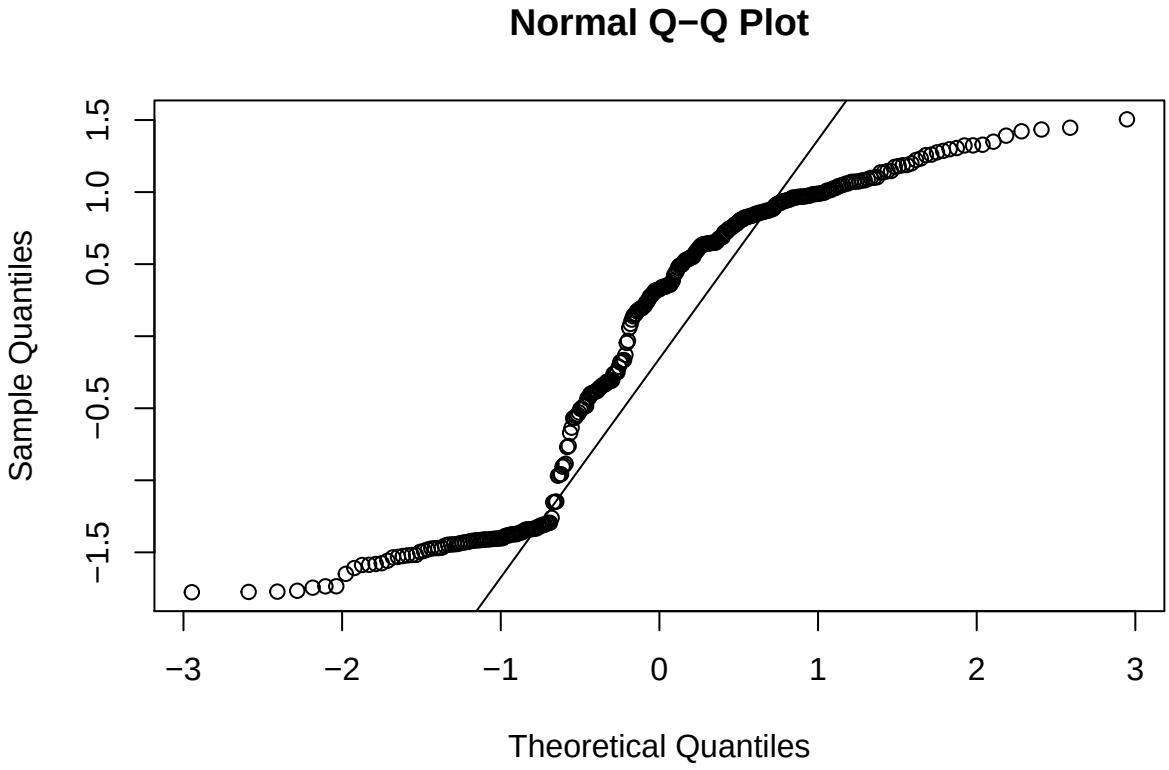
```
# Test de normalidad para la absorvancia sin normalizar
skewness(database.EC$Abs)
```

```
## [1] -0.409983
```

```
agostino.test(database.EC$Abs)
```

```
##
##  D'Agostino skewness test
##
## data:  database.EC$Abs
## skew = -0.40998, z = -2.91630, p-value = 0.003542
## alternative hypothesis: data have a skewness
```

```
qqnorm(database.EC[[6]])
qqline(database.EC[[6]])
```



La base de datos SA mejora un poco al estandarizarla. Sin emabrgo, esta sigue sin ser normal. La base de datos EC no parece verse afectada por la normalización. probaremos hacer una transformación de variable para buscar un espacio donde pueda ser normal.

Transformación de variable

Volvemos a cargar los datos

```
# Leemos las bases de datos para las bacterias Straphilococcus aureus y Escherichia coli
database.EC.transform <- read_csv("/media/veracrypt2/bacterias_materiales/bacterias_nanomateriales_2021
```

```
## New names:
## * `` -> ...7
## * `` -> ...8
## * `` -> ...9
## * `` -> ...10
```

```
## Rows: 312 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): Time, Dope, a, c, Size, Abs
## lgl (4): ...7, ...8, ...9, ...10
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
database.SA.transform <- read_csv("/media/veracrypt2/bacterias_materiales/bacterias_nanomateriales_2021
```

```
## New names:
## * `` -> ...7
## * `` -> ...8
```

```
## * `` -> ...9
## * `` -> ...10
```

```
## Rows: 312 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): Time, Dope, a, c, Size, Abs
## lgl (4): ...7, ...8, ...9, ...10
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#eliminamos las columnas vacias
database.EC.transform <- database.EC.transform[,-c(7,8,9,10)]
database.SA.transform <- database.SA.transform[,-c(7,8,9,10)]
```

```
# Convertimos a tibble
database.EC.transform <- tibble(database.EC.transform)
database.SA.transform <- tibble(database.SA.transform)
```

Solo vamos a transformar la absorvancia y luego estandarizamos toda la base de datos

```
# Transformar datos
database.EC.transform$Abs <- log(database.EC.transform$Abs)
database.SA.transform$Abs <- log(database.SA.transform$Abs)
```

Estandarizamos

```
# Formula a seguir: (valor - media) / desviación estandar
# Sacar medias
mean.vector.SA <- c()
for (index in 1:6){
  mean.value <- mean(database.SA.transform[[index]])
  mean.vector.SA <- append(mean.vector.SA, mean.value)
}

mean.vector.EC <- c()
for (index in 1:6){
  mean.value <- mean(database.EC.transform[[index]])
  mean.vector.EC <- append(mean.vector.EC, mean.value)
}

# Calcular desviación estandar
sd.vector.SA <- c()
for (index in 1:6){
  sd.value <- sd(database.SA.transform[[index]])
  sd.vector.SA <- append(sd.vector.SA, sd.value)
}

sd.vector.EC <- c()
for (index in 1:6){
  sd.value <- sd(database.EC.transform[[index]])
  sd.vector.EC <- append(sd.vector.EC, sd.value)
}
```

```
database.EC.transform.norm <- database.EC.transform
database.SA.transform.norm <- database.SA.transform
```

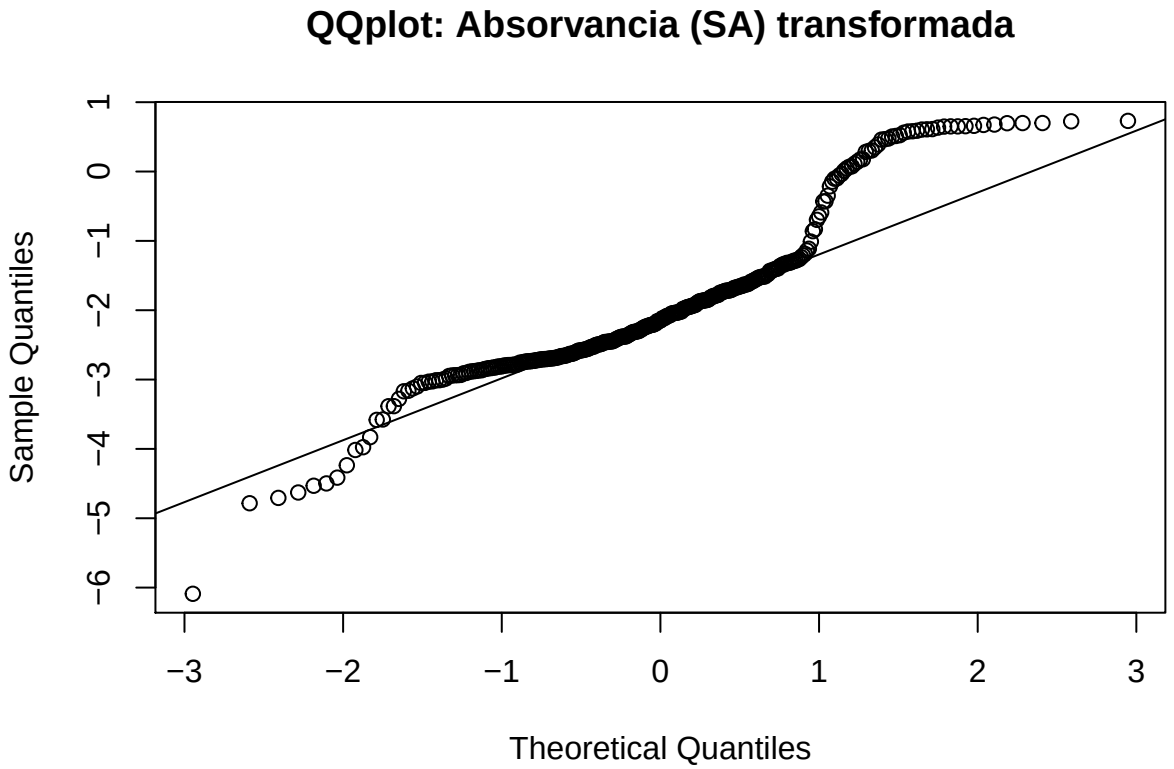
```
# Normalizamos toda la base de datos
for (index in 1:6){
  # Normalizar
  database.SA.transform.norm[,index] <- (database.SA.transform.norm[,index] - mean.vector.SA[index]) / s
}
```

```
# Normalizamos toda la base de datos
for (index in 1:6){
  # Normalizar
  database.EC.transform.norm[,index] <- (database.EC.transform.norm[,index] - mean.vector.EC[index]) / s
}
```

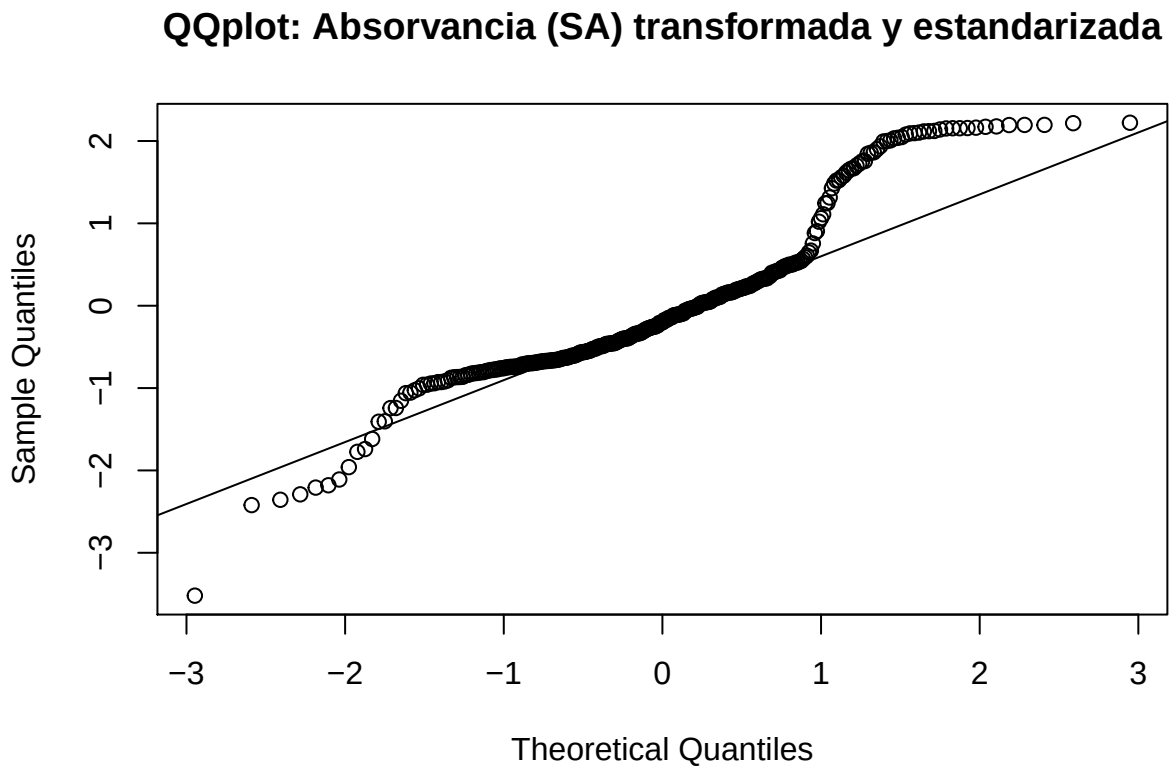
```
head(database.EC.transform.norm)
```

```
## # A tibble: 6 x 6
##    Time   Dope     a     c    Size   Abs
##   <dbl>  <dbl> <dbl> <dbl>   <dbl> <dbl>
## 1 -1.53 -0.853 0.291 0.286  0.0732 -1.36
## 2 -1.53 -0.853 0.282 0.293  0.667  -3.92
## 3 -1.53 -0.853 0.295 0.284 -0.0964 -3.09
## 4 -1.53 -0.853 0.285 0.290  0.459  -1.15
## 5 -1.53 -0.114 0.291 0.286  0.0732 -1.17
## 6 -1.53 -0.114 0.282 0.293  0.667  -1.27
```

```
qqnorm(database.SA.transform[[6]], main = "QQplot: Absorvancia (SA) transformada")
qqline(database.SA.transform[[6]])
```



```
qqnorm(database.SA.transform.norm[[6]], main = "QQplot: Absorvancia (SA) transformada y estandarizada")
qqline(database.SA.transform.norm[[6]])
```



Se puede observar que tanto los datos transformados como los datos transformados y estandarizados parecen ya dar un mejor resultado para la normalidad.

```
# Test de normalidad para la absorvancia
skewness(database.SA.transform.norm$Abs)
```

```
## [1] 0.5431305
```

```
agostino.test(database.SA.transform.norm$Abs)
```

```
##
##  D'Agostino skewness test
##
## data:  database.SA.transform.norm$Abs
## skew = 0.54313, z = 3.77164, p-value = 0.0001622
## alternative hypothesis: data have a skewness
```

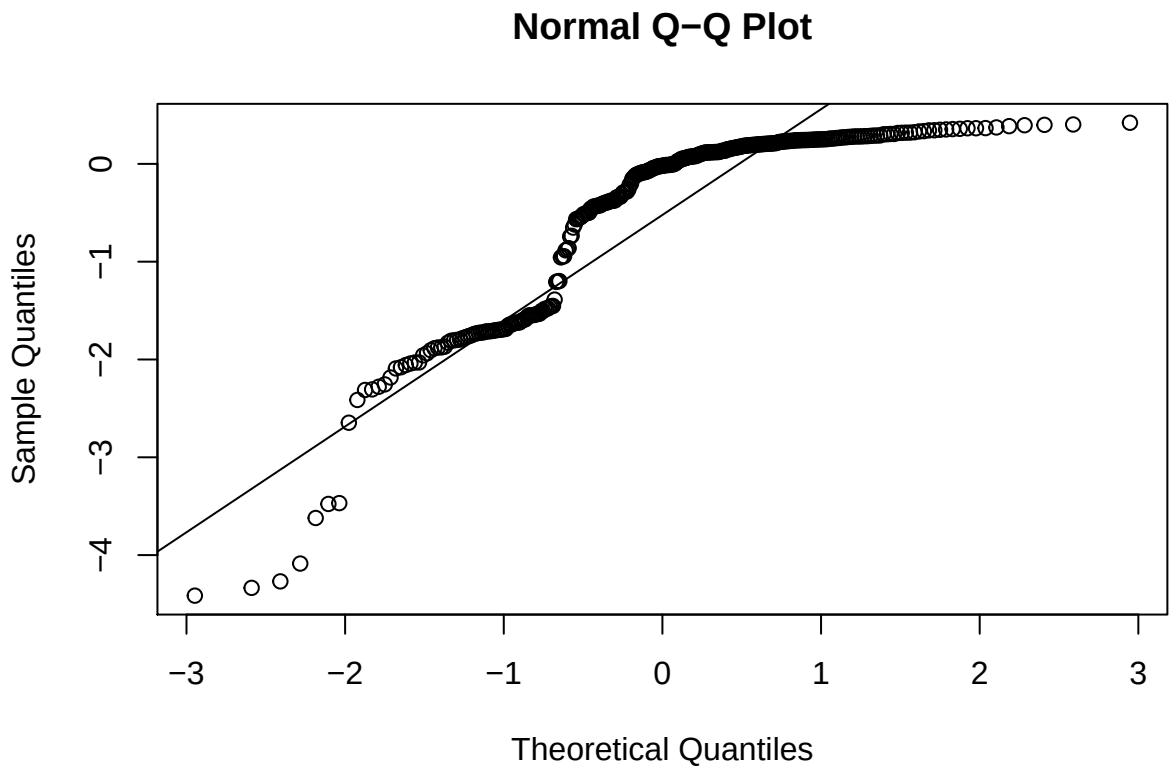
Según indica este test, los datos no son normales.

```
# Test de normalidad para la absorvancia estandarizada
shapiro.test(database.SA.transform.norm$Abs)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  database.SA.transform.norm$Abs
## W = 0.91857, p-value = 5.503e-12
```

El otro test también indica que los datos no son normales.

```
qqnorm(database.EC.transform[[6]])
qqline(database.EC.transform[[6]])
```




```
# Test de normalidad para la absorvancia estandarizada
shapiro.test(database.EC.transform.norm$Abs)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  database.EC.transform.norm$Abs
## W = 0.78867, p-value < 2.2e-16
```

```
# Test de normalidad para la absorvancia
skewness(database.EC.transform.norm$Abs)
```

```
## [1] -1.494205
```

```
agostino.test(database.EC.transform.norm$Abs)
```

```
##
##  D'Agostino skewness test
##
## data:  database.EC.transform.norm$Abs
## skew = -1.4942, z = -8.3219, p-value < 2.2e-16
## alternative hypothesis: data have a skewness
```

Resultados

No logramos hacer que los datos fueran normales. Esto creo se debe a la gran cantidad de outliers dados por los tratamientos C1 y C0. Entrenaremos una perceptrón multicapa para ver si puede aprender de los datos.

Exportamos los databases

```
write_csv(database.SA.transform.norm, "/media/veracrypt2/bacterias_materiales/bacterias_nanomateriales_1
write_csv(database.EC.transform.norm, "/media/veracrypt2/bacterias_materiales/bacterias_nanomateriales_1
```