

Ejercicios sobre Non-linear models

Jose Calatayud Mateu

2025-05-24

```
### Librerias
library(knitr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
##      collapse
```

```
library(msm)
```

Exercise 1

In biochemistry, the kinetic model of **Michaelis-Menten** is used to analyze enzyme kinetics. This model relates the **rate reaction** v with the **substrate concentration** S by means of the equation:

$$v = \frac{\phi_1 S}{\phi_2 + S}$$

where: - ϕ_1 corresponds to the **maximum reaction rate** achieved by the system (saturating value), - ϕ_2 (known as the **Michaelis constant**) corresponds to the concentration where the reaction rate is half of ϕ_1 .

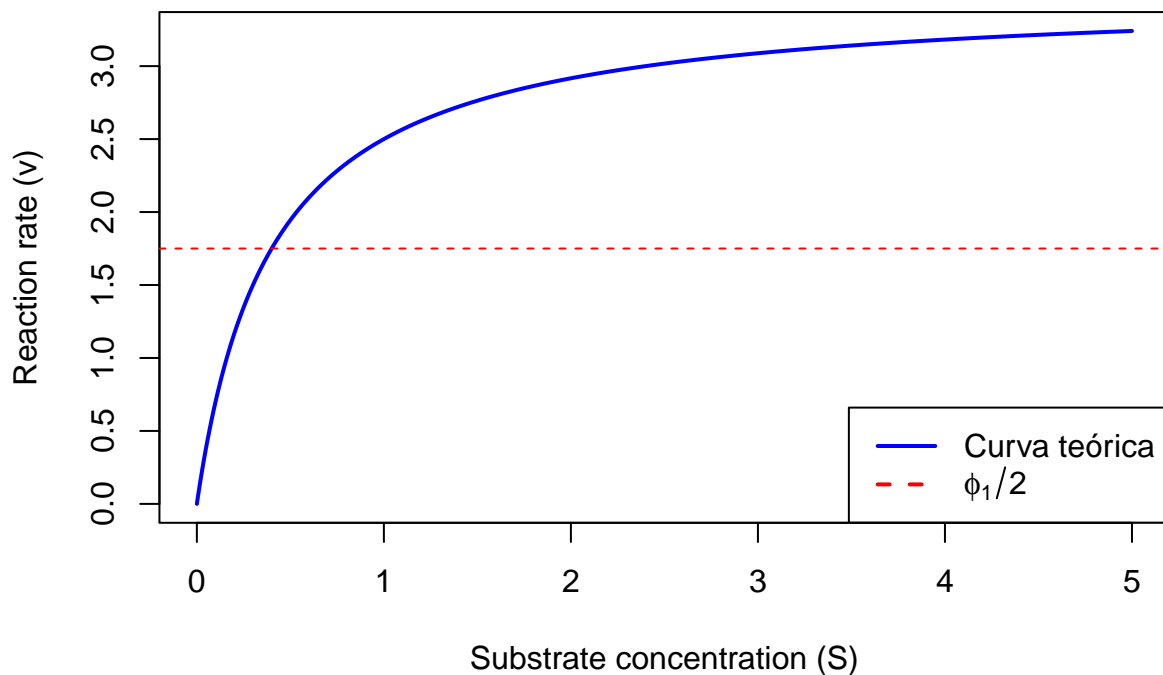
This parameter ϕ_2 is highly relevant to biologists.

- Draw the theoretical curve for $\phi_1 = 3.5$ and $\phi_2 = 0.4$ over a range of concentration values between 0 and 5.

```
phi1 <- 3.5
phi2 <- 0.4
S <- seq(0, 5, by = 0.01)
v <- (phi1 * S) / (phi2 + S)

plot(S, v, type = "l", col = "blue", lwd = 2,
      xlab = "Substrate concentration (S)",
      ylab = "Reaction rate (v)",
      main = "Michaelis-Menten Theoretical Curve")
abline(h = phi1/2, lty = 2, col = "red")
legend("bottomright", legend = c(expression("Curva teórica"), expression(phi[1]/2)),
      col = c("blue", "red"), lty = c(1, 2), lwd = 2)
```

Michaelis-Menten Theoretical Curve



- The file `kinetics.txt` contains information from an experiment to estimate the concentration at which the reaction rate is half of its maximum. Use the Michaelis-Menten model to estimate this parameter and its 95% **confidence interval (CI)** (NOTE: Investigate whether there is any generic R function to automatically compute the CI).

Del enunciado podemos deducir dos interpretaciones:

- Que en el experimento han considerado que la reaction rate es un medio del máximo y, por tanto, como que ϕ_1 es el máximo entonces ya es conocido y solo hace falta estimar ϕ_2 que en esta situación debería de representar la concentración.
- Por otra parte, podemos suponer que desconocemos de los valores de ϕ_1 y ϕ_2 y, en consecuencia, hace falta estimarlos

Suposición 1 De acuerdo con el enunciado, vemos que la variable ϕ_2 , también llamada Michaelis constant, corresponde a la concentración cuando la reaction rate es un medio del máximo, ϕ_1 . Como los datos del experimento `kinetics.txt` contiene información de la reaction rate y de la saturación, tal y como se muestra a continuación:

```
data <- read.table("kinetics.txt", header = TRUE)
print(data)
```

```
##   S    v
## 1 1 2.80
## 2 2 4.05
## 3 3 4.90
## 4 4 5.50
## 5 5 6.00
## 6 6 6.30
```

Y como que lo que se pide es estimar la concentración, que es la Michaelis constant

```
data.modif <- data
data.modif$phi1 <- 2*data$v
head(data.modif)
```

```
##   S    v phi1
## 1 1 2.80  5.6
## 2 2 4.05  8.1
## 3 3 4.90  9.8
## 4 4 5.50 11.0
## 5 5 6.00 12.0
## 6 6 6.30 12.6
```

A partir del modelo **Michaelis-Menten** y del model no lineal, determinamos la estimación de la concentración ϕ_2 y su IC(95%) como:

```
model1 <- nls(v ~ (phi1 * S) / (phi2 + S),
              data = data.modif,
              start = list(phi2 = 0.5))

summary(model1)
```

```
##
## Formula: v ~ (phi1 * S)/(phi2 + S)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## phi2    4.0211     0.8103   4.962  0.00424 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.197 on 5 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 3.108e-06
```

```
ic1 <- confint(model1) # 95% CI for the parameters
```

```
## Waiting for profiling to be done...
```

Así, en este caso el intervalo de confianza para ϕ_2 es $[2.3087199, 6.966174]$.

Suposición 2 En cambio, ahora suponemos que no conocemos nada, es decir, que no interpretamos la reaction rate como un medio del máximo y no podemos suponer conocido ϕ_1 . Entonces:

```
model2 <- nls(v ~ (phi1 * S) / (phi2 + S),
              data = data,
              start = list(phi1 = 3, phi2 = 0.5))

summary(model2)
```

```
##
## Formula: v ~ (phi1 * S)/(phi2 + S)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## phi1    8.5153     0.1730   49.22 1.02e-06 ***
## phi2    2.1503     0.1159   18.55 4.97e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07138 on 4 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 1.924e-06
```

```
ic2 <- confint(model2) # 95% CI for the parameters
```

```
## Waiting for profiling to be done...
```

Así, en este caso el intervalo de confianza para ϕ_2 es $[1.8477994, 2.5038813]$.

Exercise 2

The file `ic50.txt` contains information about **cellular growth across time** (variable `tiempo`) for three different **exposures** (low, medium, high).

- Create a plot comparing the three growth curves.

Primero cargamos los datos y los disponemos en el formato adecuado para trabajar:

```
ic50 <- read.table("ic50.txt", header = TRUE)

ic50_long <- pivot_longer(ic50, cols = c("low", "medium", "high"),
                          names_to = "exposure", values_to = "growth") %>%
  mutate("exposure" = factor(exposure, levels = c("low", "medium", "high"), labels=c("low", "medium", "h

ic50_long <- ic50_long %>% arrange(exposure, time)

head(ic50_long,6)
```

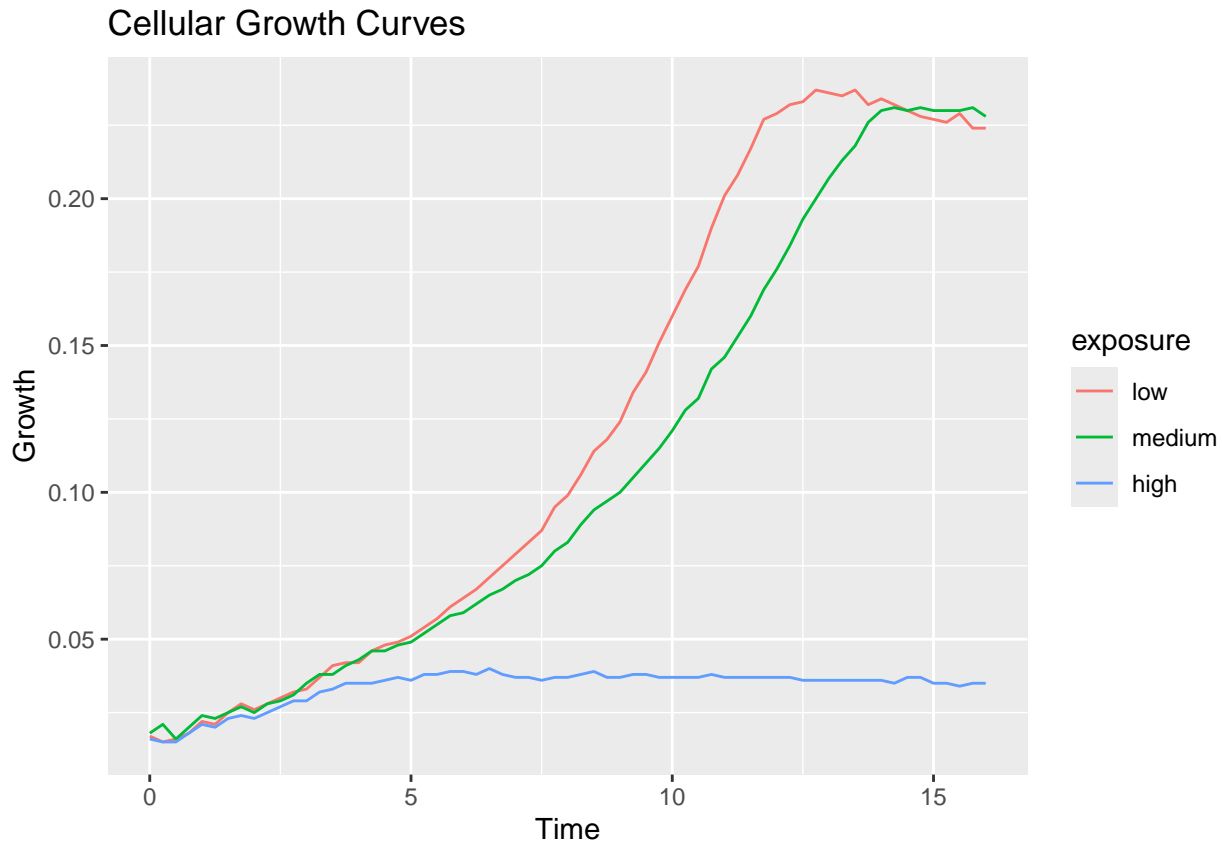
```
## # A tibble: 6 x 3
##   time exposure growth
##   <dbl> <fct>   <dbl>
## 1  0    low      0.017
## 2 0.25 low      0.015
## 3  0.5 low      0.016
## 4 0.75 low      0.018
## 5  1    low      0.022
## 6 1.25 low      0.021
```

```
tail(ic50_long,6)
```

```
## # A tibble: 6 x 3
##   time exposure growth
##   <dbl> <fct>   <dbl>
## 1 14.8 high      0.037
## 2 15    high      0.035
## 3 15.2 high      0.035
## 4 15.5 high      0.034
## 5 15.8 high      0.035
## 6 16    high      0.035
```

Hacemos la representación conjunta de las tres curvas de crecimiento para cada tipo de exposición:

```
ggplot(ic50_long, aes(x = time, y = growth, color = exposure)) +
  geom_line() +
  labs(title = "Cellular Growth Curves", y = "Growth", x = "Time")
```



- Calculate the **IC50** (time at which half of the maximum cellular growth is achieved) for each exposure using the model you consider appropriate. Assess whether there are statistically significant differences.

Se pide que estimemos la mediana IC50 que con la estimación parametrizada por la función SSlogis suponiendo que vamos a trabajar con el modelo logístico de crecimiento es $\hat{\phi}_2 = \frac{\hat{\theta}_2}{\hat{\theta}_3}$. En consecuencia, primero estimamos un modelo para cada exposición:

```
# logistic model fit per exposure group
mod.list <- nlsList(growth ~ SSlogis(time, phi1, phi2, phi3) | exposure,
                   data=ic50_long, pool=FALSE)

summary(mod.list)
```

```
## Call:
##   Model: growth ~ SSlogis(time, phi1, phi2, phi3) | exposure
##   Data: ic50_long
##
## Coefficients:
##   phi1
##       Estimate Std. Error  t value    Pr(>|t|)
## low    0.25852927 0.0066267831  39.01279 2.421817e-45
## medium 0.32550677 0.0151176024  21.53164 1.794223e-30
## high   0.03709327 0.0002377973 155.98691 3.655477e-82
##   phi2
##       Estimate Std. Error  t value    Pr(>|t|)
```

```
## low      8.6261602 0.19684342 43.82245 2.300595e-48
## medium 11.5600535 0.39744571 29.08587 7.629883e-38
## high    0.8805718 0.07501095 11.73924 2.121676e-17
##      phi3
##      Estimate Std. Error  t value    Pr(>|t|)
## low      2.455265  0.1340977 18.30952 1.079309e-26
## medium  3.697322  0.1569336 23.55979 1.230686e-32
## high    1.321006  0.0848310 15.57220 4.216337e-23
```

Ahora obtenemos los coeficientes del modelo:

```
phis <- unlist(lapply(mod.list, coef))
phis
```

```
##      low.phi1      low.phi2      low.phi3 medium.phi1 medium.phi2 medium.phi3
## 0.25852927  8.62616016  2.45526499  0.32550677 11.56005353  3.69732239
##      high.phi1      high.phi2      high.phi3
## 0.03709327  0.88057181  1.32100564
```

Y su matriz de varianzas y covarianzas:

```
vars <- lapply(mod.list, vcov)
vars
```

```
## $low
##           phi1           phi2           phi3
## phi1 4.391425e-05 0.001157851 0.0006964587
## phi2 1.157851e-03 0.038747331 0.0184467702
## phi3 6.964587e-04 0.018446770 0.0179822001
##
## $medium
##           phi1           phi2           phi3
## phi1 0.0002285419 0.005933993 0.002139724
## phi2 0.0059339933 0.157963089 0.056739953
## phi3 0.0021397242 0.056739953 0.024628158
##
## $high
##           phi1           phi2           phi3
## phi1 5.654757e-08 3.332126e-06 8.300599e-06
## phi2 3.332126e-06 5.626643e-03 -2.251783e-03
## phi3 8.300599e-06 -2.251783e-03 7.196299e-03
```

Y además, crear la matriz de varianzas y covarianzas completa:

```
zero <- matrix(0, nrow=3, ncol=3)
var <- rbind( cbind(vars[[1]], zero, zero), cbind(zero, vars[[2]], zero), cbind(zero, zero, vars[[3]]))
var
```

```
##           phi1           phi2           phi3
## phi1 4.391425e-05 0.001157851 0.0006964587 0.0000000000 0.0000000000 0.0000000000
## phi2 1.157851e-03 0.038747331 0.0184467702 0.0000000000 0.0000000000 0.0000000000
## phi3 6.964587e-04 0.018446770 0.0179822001 0.0000000000 0.0000000000 0.0000000000
```

```
## phi1 0.000000e+00 0.000000000 0.0000000000 0.0002285419 0.005933993 0.002139724
## phi2 0.000000e+00 0.000000000 0.0000000000 0.0059339933 0.157963089 0.056739953
## phi3 0.000000e+00 0.000000000 0.0000000000 0.0021397242 0.056739953 0.024628158
## phi1 0.000000e+00 0.000000000 0.0000000000 0.0000000000 0.000000000 0.000000000
## phi2 0.000000e+00 0.000000000 0.0000000000 0.0000000000 0.000000000 0.000000000
## phi3 0.000000e+00 0.000000000 0.0000000000 0.0000000000 0.000000000 0.000000000
##
## phi1 0.000000e+00 0.000000e+00 0.000000e+00
## phi2 0.000000e+00 0.000000e+00 0.000000e+00
## phi3 0.000000e+00 0.000000e+00 0.000000e+00
## phi1 0.000000e+00 0.000000e+00 0.000000e+00
## phi2 0.000000e+00 0.000000e+00 0.000000e+00
## phi3 0.000000e+00 0.000000e+00 0.000000e+00
## phi1 5.654757e-08 3.332126e-06 8.300599e-06
## phi2 3.332126e-06 5.626643e-03 -2.251783e-03
## phi3 8.300599e-06 -2.251783e-03 7.196299e-03
```

Utilizamos el método delta para obtener la estimación y el error estandar del IC50, para cada uno de los modelos:

```
library(car)
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

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

```
##
```

```
## recode
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
## some
```

```
# Modelo amb exposició HIGH
```

```
d.high <- deltaMethod(phis, "high.phi2", vcov=var)
```

```
# Modelo amb exposició LOW
```

```
d.low <- deltaMethod(phis, "low.phi2", vcov=var)
```

```
# Modelo amb exposició MEDIUM
```

```
d.medium <- deltaMethod(phis, "medium.phi2", vcov=var)
```

```
# Crear la tabla
```

```
tabla_ic50 <- data.frame(
  Exposure = c("LOW", "MEDIUM", "HIGH"),
  Estimate = c(d.low$Estimate, d.medium$Estimate, d.high$Estimate),
  SE = c(d.low$SE, d.medium$SE, d.high$SE),
  IC.lower = c(d.low$`2.5 %`, d.medium$`2.5 %`, d.high$`2.5 %`),
  IC.upper = c(d.low$`97.5 %`, d.medium$`97.5 %`, d.high$`97.5 %`)
)
```

```
# Mostrar la tabla formateada
```

```
kable(tabla_ic50, digits = 3, caption = "Estimació de la IC50 per exposició amb intervals de confiança")
```



```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

Table 1: Estimació de la IC50 per exposició amb intervals de confiança al 95%

Exposure	Estimate	SE	IC.lower	IC.upper
LOW	8.626	0.197	8.240	9.012
MEDIUM	11.560	0.397	10.781	12.339
HIGH	0.881	0.075	0.734	1.028

Para ver si hay diferencias significativas entre ellos solo tendremos que calcular las estimaciones de las diferencias a través del método delta y ver que realmente el valor 0 no está incluido en el intervalo de confianza:

```
# Comparación high-low
deltaMethod(phis, "low.phi2 - high.phi2", vcov=var)
```

```
##              Estimate      SE  2.5 % 97.5 %
## low.phi2 - high.phi2  7.74559 0.21065 7.33272 8.1585
```

```
deltaMethod(phis, "low.phi2 - medium.phi2", vcov=var)
```

```
##              Estimate      SE  2.5 % 97.5 %
## low.phi2 - medium.phi2 -2.93389 0.44352 -3.80318 -2.0646
```

```
deltaMethod(phis, "medium.phi2 - high.phi2", vcov=var)
```

```
##              Estimate      SE  2.5 % 97.5 %
## medium.phi2 - high.phi2 10.67948 0.40446 9.88675 11.472
```

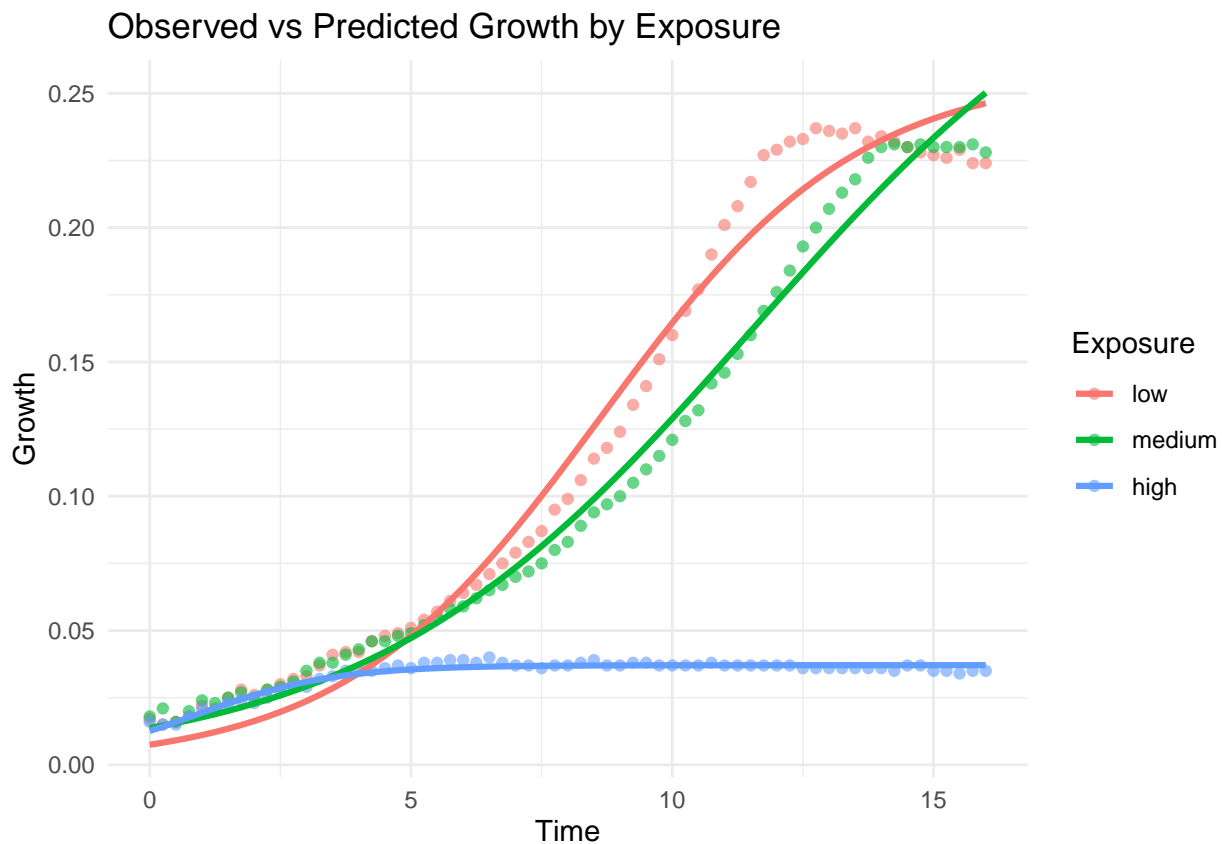
Com en ninguno de los casos, el valor 0 no pertenece dentro de los intervalos de confianza, no hay evidencia estadística suficiente para un nivel de confianza del 95%. Así que los diferentes grupos tienen estimaciones del IC50 estadísticamente diferentes.

- Create a plot with **observed vs. predicted values** to verify the model fit.

```
# Añadir predicciones al data frame original
ic50_long$pred <- predict(mod.list)

# Graficar observados vs predichos
library(ggplot2)
```

```
ggplot(ic50_long, aes(x = time)) +
  geom_point(aes(y = growth, color = exposure), alpha = 0.6) +
  geom_line(aes(y = pred, color = exposure), linewidth = 1.1, linetype = "solid") +
  labs(title = "Observed vs Predicted Growth by Exposure",
       x = "Time",
       y = "Growth",
       color = "Exposure") +
  theme_minimal()
```



- Answer: Which type of exposure has the larger growth rate?

En el modelo:

$$y(x) = \frac{\phi_1}{1 + \exp\left(\frac{\phi_2 - x}{\phi_3}\right)}$$

- ϕ_3 controla la pendiente de la curva logística.
- Una pendiente más pronunciada (crecimiento más rápido) implica valores más pequeño de ϕ_3 .
- Por tanto: menor $\phi_3 \rightarrow$ mayor tasa de crecimiento

Por tanto, extraemos los coeficientes de ϕ_3 :

```
# Extraer phi3 por grupo
phi3 <- sapply(mod.list, function(mod) coef(mod)["phi3"])
phi3
```

```
##      low.phi3 medium.phi3   high.phi3
##      2.455265   3.697322   1.321006
```

Esto devuelve un vector con los valores de `phi3` para "low", "medium" y "high".

Así, el grupo con mayor tasa de crecimiento celular (pendientes más inclinado, es decir, valor más bajo para ϕ_3) es high.phi3

```
# Crear tabla con phi3
data.frame(
  Exposure = names(phi3),
  GrowthRateParameter_phi3 = phi3
) %>%
  arrange(GrowthRateParameter_phi3) %>%
  kable(digits = 3, caption = "Comparació del paràmetre de pendent (phi3) per exposició")
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

Table 2: Comparació del paràmetre de pendent (ϕ_3) per exposició

	Exposure	GrowthRateParameter_phi3
high.phi3	high.phi3	1.321
low.phi3	low.phi3	2.455
medium.phi3	medium.phi3	3.697