8

Nonlinear Models

Nonlinear regression models are used when the relationship between the response variable (dependent variable, *Y*) and the predictor variables (independent variables, *X*) is not linear. In other words, they are employed when a straight line is not an appropriate representation of the relationship between the variables.

As we have seen in Section 4.1, polynomial regressions provide a nonlinear relationship between the response and predictor variables (as seen in the regression line fit to the data, Figure 8.1 A), but they are considered linear models because the parameters are estimated using linear least squares. Another type of nonlinear model is a semi-parametric model where the relationship between the response and predictor variables is described by a function that includes both parametric and non-parametric components. An example of a semi-parametric model is the generalised additive model (GAM) that includes a non-parametric component in the form of a spline function (Chapter 12; Figure 8.1 B).

The type of nonlinear model I cover in this chapter is a parametric model where the relationship between the response and predictor variables is described by a specific nonlinear function (Figure 8.1 C). The model still assumes that the residuals are normally distributed and exhibit homoscedasticity. The model parameters are estimated by minimising the sum of squared differences between the observed and predicted values, a method commonly referred to as nonlinear least squares (NLS) regression. This is the term I will adopt.

The primary purpose of nonlinear regression is to derive a formula (model), analyse data, and predict new values where the phenomenon exhibits a nonlinear causal pattern or behaviour. Nonlinear models include a variety of response forms, such as exponential growth models, logistic growth models, and other mechanistic models derived from physical, chemical, or biological processes. Examples of such models include trigonometric, logarithmic, and user-defined functions like the von Bertalanffy model or seasonal cycle represented by a sine curve (Figure 8.1 C). These models are explicitly nonlinear in both their form and parameters. Unlike polynomial regression, where only the terms of *X* are transformed, nonlinear models involve an entirely nonlinear function relating *X* and *Y*. They are often used when there is a theoretical basis for the specific form of the relationship, providing interpretable parameters that carry specific meanings based on the underlying theory, making them useful for detailed applications where the dynamics of the system

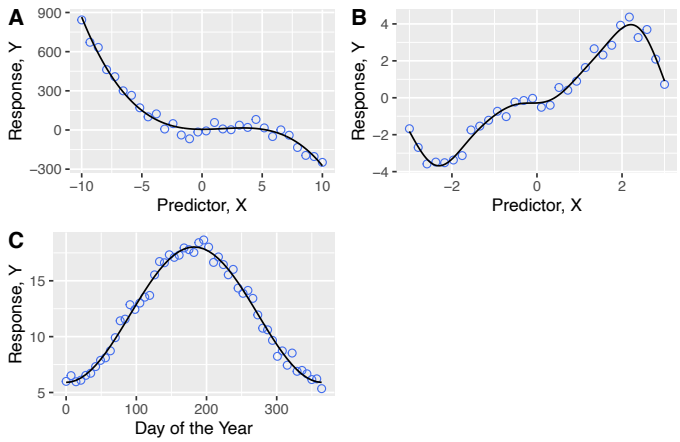


FIGURE 8.1. Nonlinear regression models fitted to simulated data. A) a cubic polynomial model, B) a GAM with a thin plate regression spline, and C) a NLS sine curve as a seasonal cycle.

are well-understood.

A general formula for a nonlinear regression model is:

$$Y_i = f(X_i; \theta) + \epsilon_i \tag{1}$$

Where:

- Y_i is the response variable for the i-th observation,
- X_i is the predictor variable for the i-th observation,
- $f(X_i; \theta)$ is a nonlinear function of X_i parameterised by the vector θ,
- θ is the vector of parameters to be estimated, and
- ϵ_i is the error term for the i-th observation and is assumed to be i.i.d. with a normal distribution.

An example of a specific nonlinear regression model is the exponential growth model:

$$Y_i = \alpha e^{\beta X_i} + \epsilon_i \tag{2}$$

Where:

- α and β are the parameters to be estimated,
- e is the base of the natural logarithm, and
- ϵ_i is the error term for the i-th observation.

This model is nonlinear in the parameters α and β, and it describes an exponential relationship between the predictor X and the response Y.

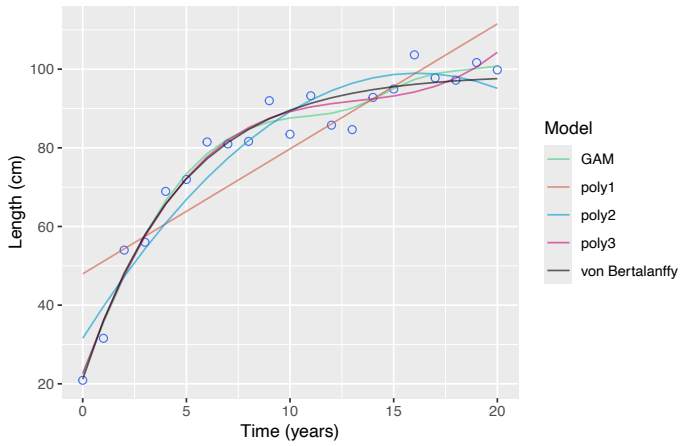


FIGURE 8.2. Plot of growth rate data fitted with a von Bertalanffy model, a first- (straight line), second- and third-order polynomial, and a GAM.

8.1 EXTENSION OF NONLINEAR MODELS

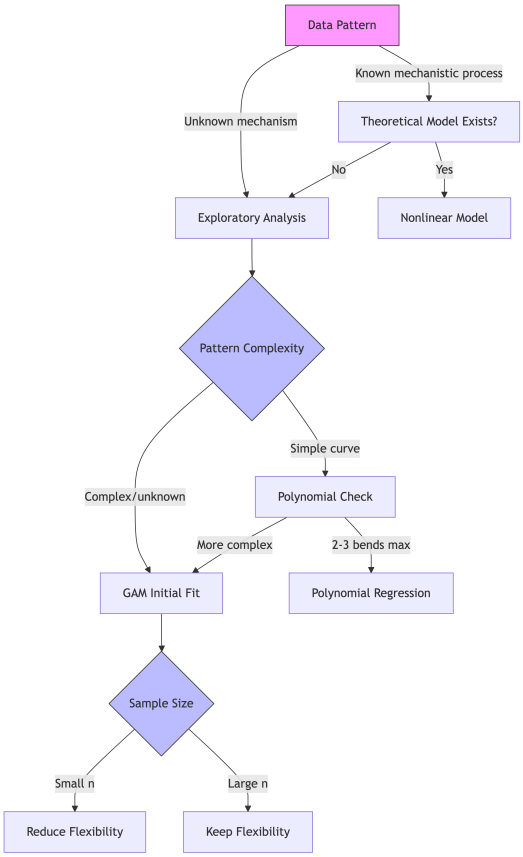
Like linear models, nonlinear models have also been extended to include multiple predictors, interactions, and other terms to capture complex relationships between the variables. The first type of more complex nonlinear models accommodates a wider range of data distributions by generalising to non-normal error distributions through link functions. These models are called generalised nonlinear models (GNLMs). The examples of GLMs in Chapter 7 should prepare you sufficiently to handle nonlinear models too. The other type deals with hierarchical data structures and incorporates fixed and random effects. As such, you can also correctly model repeated measures and longitudinal, and nested (grouped) designs. These hierarchical models are called nonlinear mixed models (NLMMs). Examples of NLMMs are provided in Section 8.5.2.3 and Section 8.5.3.

8.2 CONSIDERATIONS FOR MODEL SELECTION

There are a few practical considerations to keep in mind when choosing a suitable nonlinear (in shape) model. Sometimes different models can provide similar fits to the same data, but they may have different implications for the interpretation of the relationship between the variables. See for example Figure 8.2. The plot shows growth rate data fitted with a first-, second- and third-order polynomial, a GAM, and a NLS von Bertalanffy model. To the untrained eye and inexperienced biologist, all models seem to provide a good fit to the data, but they do differ subtly in the shape of the fitted curve. The von Bertalanffy model is a saturating growth model (it reaches a plateau), while the polynomial models and the GAM are more flexible and can capture a wider range of shapes. The choice of model should be guided

by the underlying biological or physical processes that generated the data and the research question you are trying to answer.

Regression analysis will often require that we decide among polynomial regressions, nonlinear models, GAMs, or some of the more complex hierarchical models, and there are various considerations to keep in mind when decising which model to use. I will cover some of these in the next sections.



8.2.1 *Linearity vs. Nonlinearity*

The first fork in our decision making process involves seeing if the relationship between the variables is linear or can be adequately approximated by a polynomial function, polynomial regression is a suitable choice. Nonlinear models or GAMs may be more appropriate if the relationship is nonlinear and does not follow a specific polynomial form. In Figure 8.2, it is obvious that the straight line model is not a good fit for the data, but the second- and third-order polynomial models, the GAM, and the von Bertalanffy model all provide better fits.

8.2.2 *Complexity of the Relationship*

Polynomial regression is limited in its ability to capture complex nonlinear relationships, especially those with more bends, peaks, or valleys than a polynomial of order <3 (or even 4 at a push) can capture. Another consideration is the process the data represent: if it is inherently nonlinear according to a known function such as exponential growth or decay, seasonal sinusoidal patterns, or logistic growth, then nonlinear models or GAMs are more flexible and can capture a wider range of nonlinear responses. In Figure 8.2, the von Bertalanffy model is a saturating growth model, which is a known biological process that can be captured by a nonlinear model. The 3rd-order polynomial model also seems to capture a saturating growth pattern, but it also somewhat influenced by the dip in the raw data around 12.5 years (in addition to some other nuances), but this is likely due to some random variation and is not part of the growth response.

8.2.3 *Interpretability vs. Flexibility*

Polynomial regression provides coefficients that relate to the powers of the predictor variables, but the interpretation of the β parameters is not as intuitive as in a linear model of order 1. In contrast, nonlinear models and GAMs offer greater flexibility in capturing complex patterns. GAMs may lack direct interpretability of the coefficients, but the nonlinear model offers coefficients that can be interpreted in the context of the model's structure. In Figure 8.2, the von Bertalanffy model has a clear biological interpretation (see Section 8.6), while the 3rd-order polynomial model and the GAM are more flexible and can capture a wider range of shapes (it follows the dips and peaks in the raw data closer). The 2nd-order polynomial does not fit the data as well at very low ages at 20 year, but it is still a better fit than the linear model.

8.2.4 *Overfitting Concerns*

Polynomial regression with high-degree polynomials can lead to overfitting, especially when the model complexity exceeds the underlying data patterns. Nonlinear models and GAMs can also overfit if not properly regularised or constrained. These insights can be seen when we examine the summaries of the regression fits, and can be formally assessed using cross-validation or information criteria. In Figure 8.2, the 3rd-order polynomial model seems to capture some of the random variation in the data, which may be an indication of overfitting. The GAM also seems to capture some of the random variation, but it is less pronounced than in the 3rd-order polynomial model.

8.2.5 *Data Size and Complexity*

For small to moderate-sized datasets with complex nonlinear relationships, GAMs may be more suitable due to their flexibility and ability to capture intricate patterns. For simpler relationships or when interpretability is important, nonlinear regression (with mechanistically-informed parameters) may be preferred. These are not of

concern in Figure 8.2.

8.2.6 *Model Complexity and Assumptions*

Polynomial regression assumes a specific polynomial form for the relationship, which may not hold in practice. Nonlinear models and GAMs are more flexible and do not always impose strict parametric assumptions (see Section 8.3), making them more robust to deviations from the assumed form. A detailed assessment of the model assumptions and the complexity of the relationship can help guide the choice of model. We need to add to this our biologist specialist knowledge to make the best choice.

8.2.7 *Computational Considerations*

Polynomial regression is relatively simple to implement and computationally efficient, especially for low-degree polynomials. Nonlinear models and GAMs may require more computational resources, especially for large datasets or complex models. Not a concern for the models represented in Figure 8.2.

8.3 REQUIREMENTS AND ASSUMPTIONS

Polynomial regression, nonlinear regression, and GAMs are built upon the principles of linear regression; therefore, the fundamental assumptions of normality and homoscedasticity of residuals usually still apply. Specifically, these models assume that the residuals are independent and identically distributed (i.i.d.), which implies that they are normally distributed with a constant variance (homoscedasticity). However, the specifics can vary depending on the model and the distribution of the response variable. Of course, there is also the requirement for the response variable to be continuous and independent. These assumptions help ensure that the error terms (residuals) in the model are well-behaved so that reliable inference and predictions can be obtained.

Nuances:

- **Polynomial Regression:** While a type of nonlinear regression, polynomial models are still linear in their parameters. This means that they are more bound to the classic regression assumptions and can be more sensitive to violations.
- **GAMs:** Offer more flexibility in handling nonlinear relationships. Depending on the distributions used for the outcome variable and the link functions employed, GAMs can potentially relax some of the strict normality assumptions.
- **Nonlinear Models in General:** Some truly nonlinear models (like those based on exponential or logarithmic functions) may have inherently different error structures and may not strictly require the same assumptions of normality and homoscedasticity. However, these models come with their own set of assumptions and considerations.

Important considerations:

- **Diagnostic Checks:** Regardless of the model type, it's *essential* to perform residual diagnostics to assess if assumptions are met. Visualisations (e.g., histograms, Q-Q

plots, residuals vs. fitted plots) are well-known tools.
- **Transformations:** If violations of assumptions are found, data transformation techniques (e.g., Box-Cox, log) could be considered to improve model validity.
- **Generalised Linear Models (GLMs):** An important class of models designed to handle various non-normal responses (e.g., count, binary) while extending the linear modeling framework. GLMs are good alternative to both polynomial regression and GAMs in certain contexts.
- **Mixed models:** Linear Mixed Models (LLMs), Generalised Linear Mixed Models (GLMMs), and Generalised Nonlinear Models (GNLMs) can be used to account for dependencies in the data, such as repeated measures or hierarchical structures. GAMs also accommodate mixed data structures.

The rest of this chapter will focus on the practical aspects of fitting polynomial regression models and nonlinear regressions in R. GAMs will be covered in a separate chapter due to their unique characteristics and implementation details.

8.4 R FUNCTIONS AND PACKAGES

8.4.1 *Polynomial Regression*

To fit a polynomial model in R, use the simple linear regression function `lm()` to fit the model. The purpose of `poly()` is to generate polynomial terms of a specified degree. The basic form is:

```
poly_model <- lm(y ~ poly(x, degree = 2), data = data)
```

GLMs are a generalisation of ordinary linear regression that allows for the response variable to have non-Gaussian error distributions such as one of the exponential family distributions (e.g., binomial, Poisson, gamma). These distributions are accommodated via so-called link functions within the GLM framework. The most common R function for fitting GLMs is `glm()`.

Mixed models that include random and fixed effects (see box 'Fixed and Random Effects') are also available. These are necessary for the analysis of data that have correlations within groups or hierarchies (e.g., repeated measures[1] or the inclusion of grouped variables). Commonly used are `lmer()` for LLMs and `glmer()` for GLMMs. Both functions are in the **lme4** package. Another package that accommodates LLMs is **nlme** and its `lme()` function. It has somewhat different capabilities and syntax compared to **lme4**.

> **Fixed and Random Effects**
>
> Random effects and fixed effects are used in regression models to account for different sources of variation in the data.

1. Repeated measures are multiple observations taken on the same subject or unit over time or under different conditions. Sometimes this is called longitudinal data.

Fixed effects are variables or factors that represent sources of variation that are of primary interest in the study or that have a finite and fixed number of levels or categories. These effects are assumed to have an influence on the mean response. Examples of fixed effects include:
- Treatment groups in an experiment (e.g., fertiliser A, fertiliser B, control)
- Categorical variables (e.g., sex, age group, species)
- Continuous variables (e.g., time, temperature, concentration)

The coefficients associated with fixed effects are estimated and interpreted as the primary effects of interest in the model.
Random effects are variables or factors that represent sources of variation that are not of primary interest but need to be accounted for in the model. These effects are assumed to be randomly sampled from a larger population, and their levels are theoretically infinite or too numerous to be modeled as fixed effects. Examples of random effects include:
- Subjects or individuals in a study (e.g., individual plants or animals)
- Clusters or groups (e.g., plots, aquaria, transects)
- Repeated measures or time points within subjects

Random effects are used to model the correlation or dependence among observations within the same cluster, subject, or time series. They allow for subject-specific or cluster-specific adjustments to the overall model, accounting for the fact that observations within the same group are more similar than observations from different groups.
In LMMs and GLMMs, both fixed and random effects are included. The fixed effects represent the primary effects of interest and the random effects account for the correlation or dependence within clusters or subjects.

8.4.2 *Nonlinear Regression*

In R, nonlinear regressions can be performed using the `nls()` function in the **base** package. It uses iterative algorithms to minimise the residual sum of squares and find the best-fit parameters for the user-specified nonlinear model.

The `nls()` function is most frequently used to fit user-specified nonlinear functions. The basic syntax is:

```
nls_model <- nls(y ~ f(x, theta1, theta2, ...), data = data,
                 start = list(theta1 = value1, theta2 = value2, ...))
```

GNLMs extend nonlinear models by allowing the response variable to follow one of the exponential family distributions, such as binomial, Poisson, or gamma, etc. This is done through a link function that relates the mean of the distribution to the predictors through the nonlinear model. GNLMs are fit using maximum likelihood estimation, which is flexible enough to handle various types of error distribution and link functions. The **gnm** package rovides the `gnm()` function designed for this purpose.

For data with dependencies within groups or hierarchies (such as in longitudinal studies), NLMMs are available within `nlme()`. NLMMs incorporate fixed effects (associated with the nonlinear terms) and random effects (to account for correlation and variation within groups).

8.5 EXAMPLE: ALGAL NUTRIENT UPTAKE KINETCIS

We can measure algal nutrient uptake rates using two types of experiments: multiple flask experiments and perturbation experiments. The fundamental concept underlying both methods is to introduce a known quantity of nutrients (termed the substrate) into a flask or a series of flasks and then measure the rate of nutrient uptake (V) at different substrate concentrations ($[S]$). We calculate the nutrient uptake rate as the change in nutrient concentration in the flask over a predefined time interval ($V = \Delta[S]/\Delta t$). Consequently, both experiments generate data that relate the nutrient uptake rate to the corresponding substrate concentration. The primary difference between the two methods lies in the experimental setup and the data analysis.

In the **multiple flask method**, we prepare a series of flasks, each containing a different initial concentration of the substrate nutrient to span the range typically encountered by the specimen in its natural environment. We then measure the nutrient uptake rate in *each individual flask* over a specific time period, for example by taking measurements at the start ($t = 0$) and end ($t = 30$ minutes) of the incubation. We calculate the change in substrate concentration over this time interval in each flask to determine the corresponding nutrient uptake rate. The resulting data from this method therefore consists of the different initial substrate concentrations used in each flask, paired with their respective measured nutrient uptake rates over the incubation period.

The **perturbation method** uses a single flask to which we add a high initial concentration of the substrate nutrient, set at a level that is ecologically meaningful and relevant to the study system. Instead of using multiple flasks, we measure the change in the remaining substrate concentration at multiple time points within this *same flask*, for example by taking samples every 10 or 20 minutes until all the substrate is depleted, say at 120 minutes. We calculate the change in substrate concentration between each successive time point to determine the corresponding nutrient uptake rate over that time interval. The resulting data, therefore, consist of a time series of substrate concentrations at each measurement time point, paired with the nutrient uptake rates calculated over the periods between those time points.

The important differences between the multiple flask and perturbation experiments are summarised in Table 8.1.

TABLE 8.1. Key differences between multiple flask and perturbation experiments.

Feature	Multiple Flask Experiments	Perturbation Experiments
Experimental Setup	Multiple flasks, each with different $[S]$	Single flask with initial high $[S]$

Feature	Multiple Flask Experiments	Perturbation Experiments
Data Independence	Data points are independent	Data points are correlated (repeated measures)
Analysis	Nonlinear least squares regression (NLS)	Nonlinear mixed model (NLMM)
R Function	`nls()`	`nlme::nlme()`

Our choice between multiple flask and perturbation experiments depends on our research questions and experimental constraints. In both methods, we must consider all sources of error and variability, such as measurement error, the type of nutrient, the physiological state of the alga, the light intensity, the experimental temperature, and other variables that might affect the uptake response.

We apply the Michaelis-Menten model (Equation 3) to data from multiple flask and perturbation experiments to characterise nutrient uptake. Applied to algae, this model assumes an irreversible uptake process that saturates at high substrate concentrations. It effectively quantifies key characteristics of the nutrient uptake system, including the maximum uptake rate and the algae's affinity for the nutrient.

We use the `nls()` function to fit the Michaelis-Menten model to the data from multiple flask experiments. For the perturbation experiment, things are a bit more complicated. This method includes dependent data points because the measurements are taken from the same flask at different times, introducing a correlation between observations. This violates the independence assumption required for standard regression models. To accurately analyse these data, I recommend a *nonlinear mixed-effects model* implemented in the `nlme()` function. Mixed-effects models account for fixed effects (overall trends across all observations) and random effects (variations specific to individual experimental units, in this case, time points within the same flask). This helps handle the correlation between repeated measures and produces reliable estimates of the uptake dynamics within the flask.

The Michaelis-Menten equation is given by:

$$V_i = \frac{V_{max} \cdot [S_i]}{K_m + [S_i]} + \epsilon_i \tag{3}$$

Where:

- V_i is the uptake rate at the i-th observation,
- V_{max} is the maximum nutrient uptake rate achieved,
- $[S_i]$ is the substrate concentration at the i-th observation,
- K_m is the Michaelis constant, which represents the substrate concentration at which the uptake rate is half of V_{max}, and
- ϵ_i is the error term at the i-th observation. and

The two parameters of the Michaelis-Menten model are rooted in theory and have ecophysiological interpretations. K_m is a measure of the alga's affinity for the nutrient and is determined by the kinetic constants governing the formation and dissociation of the enzyme-substrate complex responsible for taking up the nutrient; lower values indicate a higher affinity. V_{max} represents the maximum capacity of the alga to utilise the nutrient.

8.5.1 *Hypothesis Testing and the Michaelis-Menten Model*

8.5.1.1 *Linear vs. Michaelis-Menten Model.* Often, we aim to understand the relationship between two variables but we may not yet know which model best describes this relationship. For instance, in algal nutrient uptake kinetics, both a linear model and a nonlinear Michaelis-Menten model can be used to describe the relationship between nutrient uptake rate and substrate concentration. Both models are valid but they have different interpretations and unique ecophysiological implications. The choice between the two models depends on the biological system.

- **Linear models** indicate that the uptake process is inherently unsaturated, such as with the uptake of ammonium. In this case, the uptake rate continues to increase linearly with substrate concentration.
- The **Michaelis-Menten model** suggests that the uptake rate eventually saturates as the substrate concentration increases, which is often the case with nitrate.

The key question is: How do we decide which model fits our data best?

The simplest way is to visually inspect the scatter of points on a plot of the V vs. $[S]$ data, which would be part of any exploratory data analysis. If the data exhibit a clear saturation pattern, where the uptake rate levels off at high substrate concentrations, the Michaelis-Menten model is likely to provide a better fit. Conversely, if the data show a linear relationship over the observed range of substrate concentrations, the linear model may be more appropriate.

It is also important to consider the biological plausibility of the models. If there is prior knowledge or theoretical reasons to expect a saturating relationship between the uptake rate and substrate concentration, the Michaelis-Menten model may be more appropriate, even if both models provide a similar fit to the data.

Confirmation can be obtained by fitting both models to our data and comparing their performance using statistical measures such as the sum of squared residuals (SSR), Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), or log-likelihood test.

To proceed with the statistical approach, we must first set hypotheses such as these to compare the models:

H_0**:** The Michaelis-Menten model does not provide a better fit to the data than a simple linear model.

In other words, we suggest with the null hypothesis that the relationship between nutrient uptake rate and the substrate concentration is adequately described by a linear model rather than the Michaelis-Menten nonlinear model. The implication is that the uptake rate increases linearly with substrate concentration, without saturation.

H_a**:** The Michaelis-Menten model provides a significantly better fit to the data than a simple linear model.

With the alternative hypothesis we propose that the relationship between the nutrient uptake rate and the substrate concentration is best described by the nonlinear Michaelis-Menten model, so the uptake rate initially increases with substrate concentration but eventually levels off, indicating saturation.

To test these hypotheses, we can:

1. Fit both the Michaelis-Menten model and a linear model to the data.

2. Compare the goodness-of-fit of both models using statistical measures such as the SSR, AIC, or BIC.
3. Perform a model comparison test (such as an F-test or likelihood ratio test) to determine if the improvement in fit provided by the Michaelis-Menten model is statistically significant compared to the linear model.

In the above scenario, which is to decide among the linear and Michaelis-Menten models, hypotheses concerning the parameters of the models are not directly tested as they are not really of interest (except for estimating their magnitude, perhaps). Instead, the focus is on the overall goodness-of-fit of the models to the data.

8.5.1.2 *Comparing Two Michaelis-Menten Models.* Here, we may be interested in testing whether the parameters $V_{\max}$ and K_m differ from some hypothesised values or across different experimental conditions.

In the first instance, we can set up the hypotheses as follows:

$H_0 : V_{\max} = V^*_{\max}$ and $K_m = K^*_m$

where $V^*_{\max}$ and K^*_m are the hypothesised values (or values from a reference condition) for the maximum uptake rate and Michaelis constant, respectively.

$H_a : V_{\max} \neq V^*_{\max}$ or $K_m \neq K^*_m$

This alternative hypothesis states that at least one of the parameters ($V_{\max}$ or K_m) differs from the hypothesised value.

If the experiment involves different experimental conditions or treatments, we can modify the hypotheses accordingly. For example, if we want to test whether the parameters differ between two experimental conditions (A and B), the hypotheses could be:

$H_0 : V^A_{\max} = V^B_{\max}$ and $K^A_m = K^B_m$

$H_a : V^A_{\max} \neq V^B_{\max}$ or $K^A_m \neq K^B_m$

In this case, the null hypothesis states that the maximum uptake rate and Michaelis constant are the same for both experimental conditions, while the alternative hypothesis states that at least one of the parameters differs between the two conditions.

After fitting the Michaelis-Menten model to the data using the `nls()` or `nlme()` functions in R, appropriate statistical tests (e.g., likelihood ratio tests, Wald tests, or other model comparison techniques) can be performed to evaluate the hypotheses and determine whether the parameter estimates significantly differ from the hypothesised values or across experimental conditions.

8.5.2 *Multiple Flask Experiment*

8.5.2.1 *Fitting a single model (NLS).* To demonstrate fitting a nonlinear model to V vs $[S]$ data produced from a multiple flask experiment, I simulate data across a range of substrate concentrations. We then fit the model to the data using the `nls()` function in R. The dataset consists of five replicate flask sets ($n = 5$) for each of 13 substrate concentrations. Each set therefore results in independently estimated uptake rates for the initial nutrient concentrations. The dataset is shown in Table 8.2, and a plot of V as a function of $[S]$ is shown in Figure 8.3.

TABLE 8.2. Simulated data for a multiple flask experiment on an alga (showing only the top and bottom three rows).

Replicate flask	[S]	V
1	0	0.00
2	0	0.00
3	0	0.00
3	30	37.64
4	30	37.97
5	30	35.95

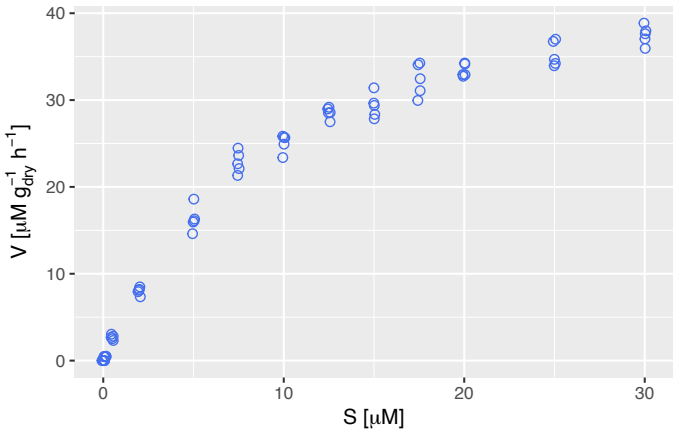


FIGURE 8.3. Plot of V as a function of $[S]$ for a multiple flask experiment involving seven replicate flask sets.

In Figure 8.3, there is a clear indication that the uptake rates plateau at higher substrate concentrations, suggesting that fitting a Michaelis-Menten model is advisable. Later, I will compare this with a linear model for completeness. A central feature of this dataset is that the data were collected independently, with each flask set representing a separate experimental unit. There is no correlation between flasks within a set, and no correlation across the initial substrate concentrations. Consequently, the assumption of independence is fully met, allowing the simplest expression of the `nls()` function to be used to fit the Michaelis-Menten model to the data.

The Michaelis-Menten model is fit to the data using the `nls()` function in R. It is specified as:

```
# Define the model function
mm_fun <- function(S, Vmax, Km) {
  Vmax * S / (Km + S)
}

# Fit the nonlinear model Michaelis-Menten model
nls_mod <- nls(V ~ mm_fun(S, Vmax, Km),                    ①
               data = mf_data,
               start = c(Vmax = 30, Km = 5))               ②
```

① The model formula specifies the Michaelis-Menten equation, with `V` as the dependent variable on the left-hand side and `S` as the independent variable on the right. The model parameters `Vmax` and `Km` will be estimated when fitting the model.
② The `start` argument provides initial values for the model parameters. The `Vmax` and `Km` parameters are estimated by minimising the sum of squared residuals between the observed and predicted values of `V`. The `nls()` function uses an iterative process to find the best-fitting values for these parameters, and the starting values improve the success of model convergence.

Here is the model summary:

```
summary(nls_mod)
>
> Formula: V ~ mm_fun(S, Vmax, Km)
>
> Parameters:
>      Estimate Std. Error t value Pr(>|t|)
> Vmax  49.2444     0.8924   55.18   <2e-16 ***
> Km     9.4953     0.4474   21.22   <2e-16 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> Residual standard error: 1.092 on 63 degrees of freedom
>
> Number of iterations to convergence: 4
> Achieved convergence tolerance: 4.705e-07
```

The above output provides the estimates for $V_{\max}$ and K_m, along with their standard errors, t-values, and p-values:

- The estimated maximum uptake rate ($V_{\max}$) is approximately 49.24 $\mu MNg^{-1}hr^{-1}$ and the small standard error associated with this parameter (0.89) indicates a precise estimate. The t-value (55.18) is very high, and the corresponding p-value is extremely small (<0.0001), indicating that $V_{\max}$ is highly significantly different from zero.
- The estimated Michaelis constant (K_m) is approximately 9.50 μM and its standard error (0.45) is also small, suggesting a precise estimate. The t-value (21.22) and the very small p-value (<0.0001) indicate that K_m is also highly significantly different from zero.

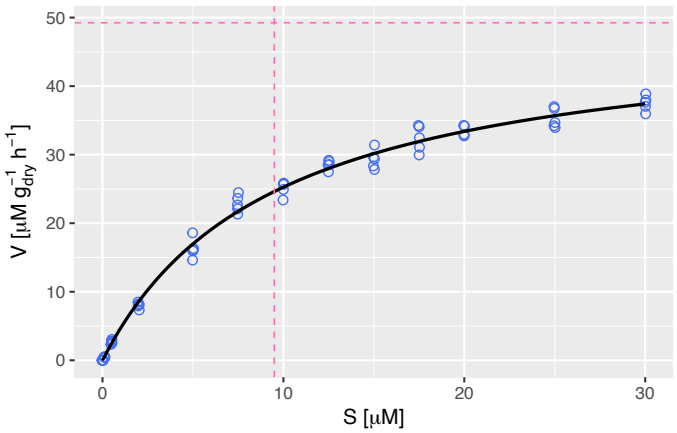


FIGURE 8.4. Plot of the Michaelis-Menten model fitted to the data in Figure 8.3. The vertical and horizontal dashed lines indicate the estimated K_m and V_{max} values, respectively.

- The residual standard error is 1.10 on 63 degrees of freedom, indicating the average deviation of the observed uptake rates from the fitted model values.
- The model converged in 4 iterations with a very small convergence tolerance, indicating a good fit and stability of the model.

Results

The Michaelis-Menten parameters, maximum uptake rate (V_{max}) and half-saturation constant (K_m), were estimated using nonlinear regression (Figure 8.4). The estimated V_{max} was 49.24 μM N g^{-1} hr^{-1} (SE = 0.89, t = 55.18, $p < 0.0001$), and the estimated K_m was 9.50 μM (SE = 0.45, t = 21.22, $p < 0.0001$). Both parameters were significantly different from zero. The model fit was good, converging in 3 iterations with a residual standard error of 1.10 (63 degrees of freedom).

The text is clear and concise, but here are a few minor changes for improved readability and precision:

Assumption tests Since these data are simulated and drawn from a normal distribution with equal variances across the range of substrate concentrations, the assumptions of homoscedasticity and normality of residuals are inherently met. In this example, we fit the model solely to obtain estimates of the Michaelis-Menten parameters, rather than to make predictions, inferences, or calculate confidence intervals. Therefore, assumption tests are not critical at this stage. We will formally test assumptions in Section 8.5.2.3 when comparing the effects of experimental treatments on kinetic parameters.

8.5.2.2 *Is the Michaelis-Menten model a better fit than a linear model?.* In Section 8.5.1.1, we pose a hypothesis that requires comparing a linear model to a Michaelis-Menten model fitted to the same data. Figure 8.4 indicates the nonlinear model indeed provides a very good fit but in some situations this distinction may be less clear and require verification. Let us fit a linear model to the above data and compare it to the Michaelis-Menten model.

```
# Fit the linear model
lm_mod <- lm(V ~ S, data = mf_data)

summary(lm_mod)
>
> Call:
> lm(formula = V ~ S, data = mf_data)
>
> Residuals:
>    Min     1Q Median     3Q    Max
> -9.354 -4.791  0.580  4.948  8.293
>
> Coefficients:
>             Estimate Std. Error t value Pr(>|t|)
> (Intercept)  6.46005    0.98044   6.589 1.03e-08 ***
> S            1.29488    0.06683  19.376  < 2e-16 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> Residual standard error: 5.13 on 63 degrees of freedom
> Multiple R-squared:  0.8563,  Adjusted R-squared:  0.854
> F-statistic: 375.4 on 1 and 63 DF,  p-value: < 2.2e-16
```

The linear model summary shows that the slope and intercept are significantly different from zero, indicating a good fit. The R^2 value is 0.86, which is very high, suggesting that the linear model explains 86% of the variance in the data. The residual standard error is 5.13, which is higher than the Michaelis-Menten model, indicating a worse fit. We can test the difference between the models formally by examining the AIC, BIC, or SSR, and the likelihood ratio test.

```
AIC(lm_mod, nls_mod)
>         df      AIC
> lm_mod   3 400.9933
> nls_mod  3 199.8814
```

```
BIC(lm_mod, nls_mod)
>         df      BIC
> lm_mod   3 407.5164
> nls_mod  3 206.4046
```

```
# Calculate the sum of squared residuals (SSR)
sum(residuals(lm_mod)^2)
> [1] 1657.938
sum(residuals(nls_mod)^2)
> [1] 75.13611
```

```
anova(lm_mod, nls_mod)
> Analysis of Variance Table
>
> Response: V
>           Df Sum Sq Mean Sq F value    Pr(>F)
> S          1 9879.9  9879.9  375.43 < 2.2e-16 ***
> Residuals 63 1657.9    26.3
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The AIC, BIC, and SSR values for the Michaelis-Menten model are lower than those for the linear model. Low is good, and we conclude that the Michaelis-Menten model is a better fit. The likelihood ratio test also shows that the Michaelis-Menten model is significantly better than the linear model (d.f. $= 1$, $F = 375.43$, $p < 0.0001$). Therefore, we can conclude that the Michaelis-Menten model is the most appropriate model for these data and that the rate of nutrient uptake by the seaweed (in this example) is saturated at high nutrient concentrations.

8.5.2.3 *Comparing treatment effects (NLS and NLMM).* Experiments are seldom as simple as the one above. To develop our example further, consider an experiment designed to assess whether an experimental treatment, such as light intensity or seawater temperature, affects the nutrient uptake rate of a seaweed. It is biologically plausible to expect that each treatment will result in unique V_{max} and/or K_m values. For example, we know that the uptake rate of nitrate (NO_3^-) might increase at higher light intensities and higher temperatures. Therefore, our hypothesis for this experiment is that the nutrient uptake kinetics of the seaweed is influenced by the treatment, as more formally stated in Section 8.5.1.2. To test this hypothesis, we fit a Michaelis-Menten model so that it allows estimates of V_{max} and K_m to vary among treatment groups.

The data for a multiple flask experiment with a treatment effect comprised of three levels are provided in Table 8.3. Except for a new variable (treatment), the data are in all other respects identical to those in Section 8.5.2.1.

Option 1 The `nls()` function in R does not handle factor variables directly, which means we cannot include the treatment variable as a factor in the model formula. To address this limitation, we fit the `nls()` model separately for each treatment group. This approach allows each treatment to have its own $V_{\max}$ and K_m values, effectively accommodating the variability in the Michaelis-Menten parameters across treatments.

In addition to fitting separate models for each treatment, we also fit a global

TABLE 8.3. Simulated data with three treatment levels for a multiple flask experiment on a seaweed species.

Treatment	Replicate flask	[S]	V
Treatment 1	1	0	0.00
Treatment 1	2	0	0.00
Treatment 1	3	0	0.00
Treatment 3	3	30	17.19
Treatment 3	4	30	16.66
Treatment 3	5	30	16.00

model (a null model) to all the data. The global model assumes that the effect of the experimental treatment is negligible, meaning that all treatments share the same $V_{\max}$ and K_m. This global fit serves as a baseline for comparison.

To determine whether the Michaelis-Menten parameters significantly differ among the treatment groups, we perform a likelihood ratio test. The likelihood ratio test compares the fit of the global model (where parameters are shared across treatments) to the combined fit of the separate models (where parameters vary by treatment). The test statistic is the difference in the log-likelihoods of the two models, which follows a χ^2 distribution with degrees of freedom equal to the difference in the number of parameters between the two models.

```
# Fit separate models
separate_models <- mf_data2 ▷
  group_by(trt) ▷
  nest() ▷
  mutate(model = map(data, ~nls(V ~ mm_fun(S, Vmax, Km),
                                data = .x,
                                start = list(Vmax = 40, Km = 10))))

# Extract model summaries of separate models
model_summaries <- separate_models▷
  mutate(summary = map(model, broom::tidy))

# Display summaries of separate models
model_summaries ▷
  select(trt, summary) ▷
  unnest(summary)
> # A tibble: 6 x 6
> # Groups:   trt [3]
>   trt         term  estimate std.error statistic  p.value
>   <fct>       <chr>    <dbl>     <dbl>     <dbl>    <dbl>
> 1 Treatment 1 Vmax     49.2      0.958      51.4 3.94e-53
> 2 Treatment 1 Km        9.55     0.482      19.8 9.50e-29
```

```
> 3 Treatment 2 Vmax     39.4      0.865      45.5 6.66e-50
> 4 Treatment 2 Km        7.54     0.481      15.7 2.14e-23
> 5 Treatment 3 Vmax     19.2      0.558      34.5 1.34e-42
> 6 Treatment 3 Km        5.87     0.560      10.5 1.97e-15

# Fit the global model
global_model <- nls(V ~ mm_fun(S, Vmax, Km),
                    data = mf_data2,
                    start = list(Vmax = 45, Km = 9))

# Extract log-likelihoods and degrees of freedom
logLik_global <- logLik(global_model)
df_global <- attr(logLik_global, "df")

# Combined log-likelihoods and degrees of freedom
logLik_separate <- sum(sapply(separate_models$model, logLik))
df_separate <- sum(sapply(separate_models$model,
                          function(m) attr(logLik(m), "df")))

# Perform the likelihood ratio test
lrt_stat <- 2 * (logLik_separate - logLik_global)
p_value <- pchisq(lrt_stat, df = df_separate - df_global,
                  lower.tail = FALSE)

# Display results
cat("Global model log-likelihood:", logLik_global, "\n")
> Global model log-likelihood: -620.5374
cat("Separate models log-likelihood:", logLik_separate, "\n")
> Separate models log-likelihood: -300.2111
cat("Degree of freedom:", df_separate - df_global, "\n")
> Degree of freedom: 6
cat("Likelihood ratio test statistic:", lrt_stat, "\n")
> Likelihood ratio test statistic: 640.6525
cat("p-value:", p_value, "\n")
> p-value: 3.953134e-135
```

The results of the likelihood ratio test indicate whether the variation in $V_{\max}$ and K_m among the treatments is statistically significant. If the test is significant, it suggests that the Michaelis-Menten parameters differ across treatments. We interpret the results as follows:

- The log-likelihood value (-620.7498) for the global model, indicating the fit of the model with shared parameters.
- The combined log-likelihood value (-313.1862) for the separate models, indicating the fit of the models with parameters varying by treatment.
- The calculated test statistic (615.1273) for the likelihood ratio test on 6 degrees of freedom.
- The p-value of the test is less than 0.0001 and provides strong evidence that $V_{\max}$ and K_m differ significantly among the treatment groups.

Results

The analysis aimed to determine if the Michaelis-Menten parameters $V_{\max}$ and K_m significantly differed among the three experimental treatments. This was evaluated by fitting a global model with shared $V_{\max}$ and K_m values across all treatments and comparing it to a model allowing separate $V_{\max}$ and K_m estimates for each treatment. The log-likelihood value for the global model, which assumes shared $V_{\max}$ and K_m values across all treatments, was -620.75, indicating the fit of the model with common parameters. In contrast, the combined log-likelihood value for the separate models, which allow $V_{\max}$ and K_m to vary by treatment, was -313.19, indicating the fit of the models with treatment-specific parameters. The calculated test statistic for the likelihood ratio test was 615.13 (d.f. $= 6$, $p < 0.001$), providing strong evidence that the Michaelis-Menten parameters $V_{\max}$ and K_m differ significantly among the treatment groups. Consequently we estimate a V_{max} of 49.2 ± 0.96, 39.4 ± 0.87 μM N g^{-1} hr^{-1} and 18.9 ± 0.65 and a K_m of 9.55 ± 0.48, 7.54 ± 0.48 and 5.50 ± 0.64 μM for treatments 1, 2 and 3 respectively.

Option 2 If Option 1 seems cumbersome, we can fit a NLMM using the **nlme** package instead. This package allows us to fit a mixed model with random effects for each treatment group. In this model, the fixed effects are the Michaelis-Menten parameters $V_{\max}$ and K_m, which vary by treatment, while the random effects are the replicate-specific intercepts. Thus, the cumbersome `nls()` formulation is replaced by the compact but more fiddly `nlme()` model specification. Pick your poison. The model is specified as follows:

```
# Fit the model with the same parameters for both treatments
# Starting values for Vmax and Km
start_vals <- c(Vmax = 50, Km = 10)
global_model <- nlme(
  V ~ mm_fun(S, Vmax, Km),
  data = mf_data2,
  fixed = Vmax + Km ~ 1,                                    ①
  random = Vmax ~ 1 | trt/rep,                              ②
  start = start_vals
)

# Fit the model with parameters varying by treatment
# Starting values for Vmax and Km for each treatment
start_vals <- c(Vmax1 = 50, Vmax2 = 40, Vmax3 = 30,
                Km1 = 10, Km2 = 10, Km3 = 5)                ③
separate_models <- nlme(
  V ~ mm_fun(S, Vmax, Km),
  data = mf_data2,
  fixed = list(Vmax ~ trt, Km ~ trt),                       ④
  random = Vmax ~ 1 | trt/rep,
```

```
  start = start_vals
)
```

① The fixed effects indicate that both $V_{\max}$ and K_m are fixed (do not vary) across treatments.
② The random effects indicate that the $V_{\max}$ parameter varies by treatment and replicate.
③ The starting values for the $V_{\max}$ and K_m parameters are specified for each treatment group. Because we are now fitting a separate model for each treatment, we need to provide starting values for each treatment.
④ The fixed effects now indicate that both $V_{\max}$ and K_m vary by treatment.

The estimated parameters for the global model and the separate models can be extracted using the `summary()` function:

```
# Extract the estimated parameters (abbreviated output)
# summary(global_model) # for verbose output
summary(global_model)$tTable
>          Value Std.Error  DF   t-value      p-value
> Vmax 36.248519  6.287216 179  5.765432 3.504878e-08
> Km    8.271727  0.304345 179 27.178780 1.952829e-65
```

```
# Extract the estimated parameters (abbreviated output)
# summary(separate_models) # for verbose output
summary(separate_models)$tTable
>                          Value Std.Error  DF    t-value       p-value
> Vmax.(Intercept)     49.199643 0.9498953 175  51.794808 4.546825e-108
> Vmax.trtTreatment 2  -9.879910 1.2312719 175  -8.024150  1.422499e-13
> Vmax.trtTreatment 3 -29.971535 1.1529098 175 -25.996427  5.425758e-62
> Km.(Intercept)        9.542071 0.4707903 175  20.268197  8.782004e-48
> Km.trtTreatment 2    -2.027313 0.6350017 175  -3.192611  1.671900e-03
> Km.trtTreatment 3    -3.689268 0.7898830 175  -4.670651  5.961284e-06
```

The log-likelihood ratio test can then easily be performed using the `anova()` function, which compares the global model with the separate models:

```
anova(global_model, separate_models)
>                Model df      AIC      BIC    logLik   Test  L.Ratio p-value
> global_model       1  5 657.9763 674.3413 -323.9882
> separate_models    2  9 621.5038 650.9608 -301.7519 1 vs 2 44.47252  <.0001
```

Again, the results of the likelihood ratio test indicate that the variation in $V_{\max}$ and K_m among the treatments is statistically significant (log-likelihood = 45.20, $p < 0.0001$). The AIC values can also be used to compare the models, with lower AIC values indicating a better fit. In this case, the separate models have a lower AIC value (644.28), suggesting that they provide a better fit to the data than the global model (681.479). The data fitted with the global and separate models is presented in

Figure 8.6.

Assumption tests To complete our example comparing the Michaelis-Menten parameters among treatments, let's confirm the assumptions by examining the residuals. Residuals in nonlinear regression models have the same interpretation as in linear models, and therefore, the assumption tests available for linear models can be applied here as well. For instance, we can use the `shapiro.test()` function to check the normality of residuals, as shown below, and the `hist()` and `plot()` functions for diagnostic plots. In real-world data, it is advised to verify these assumptions before accepting the analysis and drawing conclusions from the nonlinear regression model. Let's check the normality of residuals for each treatment and plot the residuals to check for normality and homoscedasticity (Figure 8.5).

```
# Add residuals and fitted information to the data frame
mf_data2$residuals_separate <- residuals(separate_models)
mf_data2$fitted_values_separate <- fitted(separate_models)

# Perform the Shapiro-Wilk test for each treatment
shapiro.test(mf_data2$residuals_separate[mf_data2$trt == "Treatment 1"])
>
>   Shapiro-Wilk normality test
>
> data:  mf_data2$residuals_separate[mf_data2$trt == "Treatment 1"]
> W = 0.976, p-value = 0.2374
shapiro.test(mf_data2$residuals_separate[mf_data2$trt == "Treatment 2"])
>
>   Shapiro-Wilk normality test
>
> data:  mf_data2$residuals_separate[mf_data2$trt == "Treatment 2"]
> W = 0.97125, p-value = 0.1344
shapiro.test(mf_data2$residuals_separate[mf_data2$trt == "Treatment 3"])
>
>   Shapiro-Wilk normality test
>
> data:  mf_data2$residuals_separate[mf_data2$trt == "Treatment 3"]
> W = 0.95091, p-value = 0.01177
```

The Shapiro-Wilk test results indicate that the residuals are normally distributed for Treatments 1 and 2 ($p > 0.05$) but not for Treatment 3 ($p < 0.05$). However, the histograms in Figure 8.5 show that the residuals are approximately normally distributed for all treatment groups, with the median roughly in the middle of the distribution in each case. This apparent discrepancy can be explained by the sensitivity of the Shapiro-Wilk test to sample size. With large sample sizes, even minor deviations from normality can be detected as statistically significant. In situations such as this one, I suggest that it is important to consider the sample size and visual inspection of the data when interpreting the results of normality tests. Here, given the relatively large sample size and the visual assessment of the histograms, we can reasonably conclude that the residuals are approximately

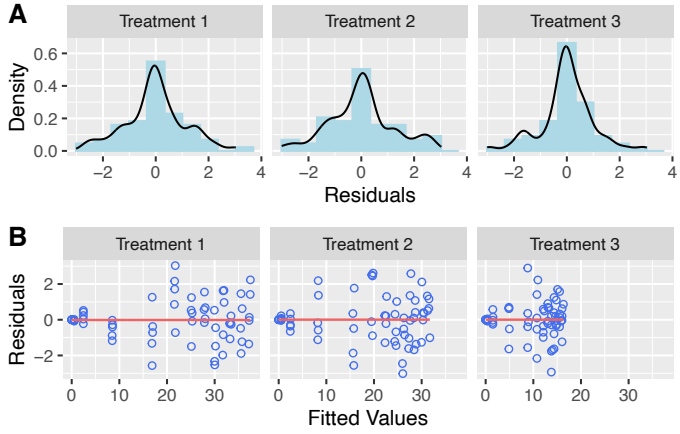


FIGURE 8.5. Histograms (A) of residuals and plots of residuals vs. the fitted values (B) for residuals for the three treatments in the multiple-flask experiment.

normally distributed for all treatment groups.

Another normality tests such as the Kolmogorov-Smirnov (K-S) test might be less sensitive to sample size and could be considered for comparison. The K-S test is a non-parametric statistical test that is used to determine if a sample comes from a specific probability distribution. Here I use it to test if a sample follows a normal distribution (`pnorm`), but it can also be used to test against other theoretical distributions or to compare two empirical distributions. The K-S test can be performed using the `ks.test()`, as shown below.

```
perform_ks_test <- function(data, treatment) {
  ks.test(data$residuals_separate[data$trt == treatment], "pnorm",
          mean = mean(data$residuals_separate[data$trt == treatment]),
          sd = sd(data$residuals_separate[data$trt == treatment]))
}

# Perform the test for each treatment group
perform_ks_test(mf_data2, "Treatment 1")
>
>   Asymptotic one-sample Kolmogorov-Smirnov test
>
> data:  data$residuals_separate[data$trt == treatment]
> D = 0.10658, p-value = 0.4513
> alternative hypothesis: two-sided
perform_ks_test(mf_data2, "Treatment 2")
>
>   Asymptotic one-sample Kolmogorov-Smirnov test
>
```

```
> data:  data$residuals_separate[data$trt == treatment]
> D = 0.1246, p-value = 0.2652
> alternative hypothesis: two-sided
perform_ks_test(mf_data2, "Treatment 3")
>
>   Asymptotic one-sample Kolmogorov-Smirnov test
>
> data:  data$residuals_separate[data$trt == treatment]
> D = 0.14151, p-value = 0.148
> alternative hypothesis: two-sided
```

We see that the K-S test indicates that the residuals are normally distributed for all treatment groups ($p > 0.05$). As already noted, this test is less sensitive to sample size than the Shapiro-Wilk test, and the results are consistent with the visual assessment of the histograms.

We should also check for homoscedasticity (here I use the Levene test) and a plot of residuals versus fitted values.

```
# Perform the Levene test
car::leveneTest(residuals_separate ~ trt, data = mf_data2)
> Levene's Test for Homogeneity of Variance (center = median)
>        Df F value Pr(>F)
> group   2  1.4933 0.2272
>       192
```

The Levene test shows that the variances are the same across the three treatments and this is confirmed by the plot of residuals against the fitted values in Figure 8.5.

Results

Michaelis-Menten models were fitted to nutrient uptake data across three experimental treatments to investigate the effects of the treatments on seaweed nutrient kinetics. A global model, assuming shared kinetic parameters (V_{max} and K_m) across all treatments, was compared to a model with separate parameters for each treatment. The model allowing treatment-specific parameters (AIC = 644.3) provided a significantly better fit to the data than the global model (AIC = 681.5), a finding confirmed by the log-likelihood test (log-likelihood ratio = 45.20, d.f. = 4, $p < 0.0001$). As the assumption tests do not indicate any cause for concern regarding the distribution of residuals, we conclude that the experimental treatments significantly influenced the nutrient uptake kinetics of the seaweed (Figure 8.6).
Specifically, all three treatments exhibited unique combinations of V_{max} and K_m values (Treatment 1: $V_{max} = 49.2$, $K_m = 9.5$; Treatment 2: $V_{max} = 39.3$, $K_m = 7.5$; Treatment 3: $V_{max} = 19.0$, $K_m = 5.5$). These findings support the hypothesis that nutrient uptake kinetics in this seaweed species are sensitive

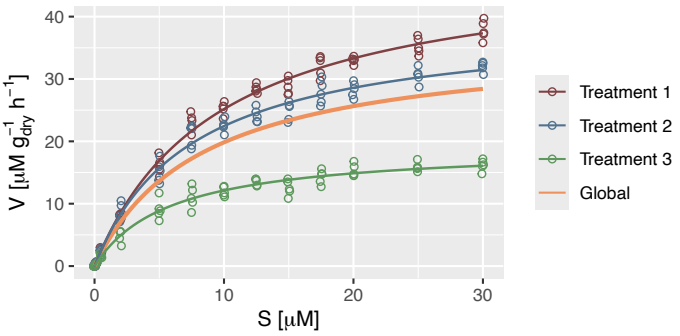


FIGURE 8.6. Plot of the Michaelis-Menten model fitted to the data in Table 8.3. Fits are provided for the separate models and the global model.

TABLE 8.4. Simulated data for a multiple flask experiment on an alga (showing only the top and bottom three rows).

Replicate flask	Treatment	V	[S]
1	low	10.8	60.2
2	low	10.0	61.1
3	low	14.1	60.8
1	high	0.0	0.1
2	high	0.0	0.1
3	high	0.0	0.1

to environmental perturbations.

8.5.3 *The Perturbation Method (NLMM)*

The data for this example is by [5]. A perturbation experiment was conducted to determine the nutrient uptake rate versus nutrient concentration of the red seaweed, *Gracilaria* sp. The experiment involved flasks, initially enriched to approximately 55 µ M nitrate, sampled 16 times over approximately 2.5 hours. The uptake rates were measured under three rates of water movement (treatments): low, medium, and high. Each treatment had three replicate flasks (Table 8.4). The primary objective was to determine if the Michaelis-Menten parameters significantly differ among the three levels of water movement, and we must state a hypothesis similar to those in Section 8.5.1.2.

For the reasons discussed in Section 8.5, we will use a nonlinear mixed effects model, `nlme()`, to analyse these data. Models such as these can be quite challenging

to fit. There are several things we have to deal with. First and most obviously is the fact that the data are repeated measures, and the residuals may be correlated. Second, the flasks are nested within the treatment levels, and we need to account for this in the model. Finally, we need to account for the possibility that the Michaelis-Menten parameters may vary among the treatment levels—in fact, we want to test this! Here is the model:

```
# Determine the number of levels in the factor 'trt'
num_levels <- length(levels(mm_data$trt))

# Starting values for the fixed parameters
# (one set for each level of 'trt')
start_vals <- list(fixed = c(Vmax = rep(max(mm_data$V), num_levels),
                             Km = rep(median(mm_data$S), num_levels)))

nlme_mod2 <- nlme(V ~ mm_fun(S, Vmax, Km),
                  data = mm_data,
                  fixed = Vmax + Km ~ trt,                    ①
                  random = Vmax + Km ~ 1 | flask,             ②
                  start = start_vals,
                  method = "REML")
```

① The `fixed` argument specifies that the Michaelis-Menten parameters `Vmax` and `Km` are fixed effects that vary among the treatment levels, and a grouping variable (`trt`) is used to specify the levels of the treatment factor.
② The `random` argument specifies that the Michaelis-Menten parameters `Vmax` and `Km` are random effects that vary among the replicate flasks.

This model brings us closer to our goal, but there are some notable omissions. The specification allows the Michaelis-Menten parameters to vary among the treatment levels, which is central to our hypothesis. We have also accounted for the replication structure of the data, recognising that random variations may arise not due to the treatment levels but due to the replicate flasks.

However, we have not accounted for the central feature of a perturbation experiment, which is the correlation structure of the residuals. We must deal with the fact that the residuals may be correlated due to the repeated measures nature of the data. Additionally, we have omitted the nesting of the flasks within the treatment levels.

Let's update our model accordingly:

```
nlme_mod3 <- nlme(V ~ mm_fun(S, Vmax, Km),
                  data = mm_data,
                  fixed = list(Vmax ~ trt, Km ~ trt),
                  random = Vmax ~ 1 | trt/flask,              ①
                  groups = ~ trt/flask,                       ②
                  correlation = corAR1(form = ~ 1 | trt/flask), ③
                  start = start_vals,
                  method = "REML")
```

① The `random` argument specifies that the Michaelis-Menten parameter `Vmax` is a random effect that varies among the replicate flasks nested within the treatment levels.
② The `groups` argument specifies that the replicate flasks are nested within the treatment levels.
③ The `correlation` argument specifies that the residuals have a first-order autoregressive correlation structure. This structure assumes that the correlation between residuals decreases exponentially with the time lag between observations. Flask is nested *within* treatment.

If we are not convinced that `nlme_mod3` is the best model, we can compare it to `nlme_mod2` using a likelihood ratio test. It is used to compare the fit of two models, where one model is a special case of the other. The test statistic is the difference in the log-likelihoods of the two models, and the null hypothesis is that the simpler model is the best fit.

```
anova(nlme_mod2, nlme_mod3)
>           Model df      AIC      BIC    logLik
> nlme_mod2     1 10 637.2782 665.6411 -308.6391
> nlme_mod3     2 10 632.1053 660.4681 -306.0527

# Likelihood ratio test
lrt_stat <- -2 * (logLik(nlme_mod2) - logLik(nlme_mod3))

# Determine degrees of freedom and p-value
df_diff <- attr(logLik(nlme_mod3), "df") - attr(logLik(nlme_mod2), "df")
p_value <- pchisq(lrt_stat, df = df_diff, lower.tail = FALSE)

print(paste("LRT statistic:", lrt_stat))
> [1] "LRT statistic: 5.17293584867423"
print(paste("Degrees of freedom:", df_diff))
> [1] "Degrees of freedom: 0"
print(paste("P-value:", p_value))
> [1] "P-value: 0"
```

The likelihood ratio test indicates that `nlme_mod3` is a better fit than `nlme_mod2` ($p < 0.001$). This result suggests that the Michaelis-Menten parameters vary among the treatment levels, and the residuals have a first-order autoregressive correlation structure.

```
summary(nlme_mod3)
> Nonlinear mixed-effects model fit by REML
>   Model: V ~ mm_fun(S, Vmax, Km)
>   Data: mm_data
>        AIC      BIC    logLik
>   632.1053 660.4681 -306.0527
>
> Random effects:
```

```
>  Formula: Vmax ~ 1 | trt
>         Vmax.(Intercept)
> StdDev:       0.00837941
>
>  Formula: Vmax ~ 1 | flask %in% trt
>         Vmax.(Intercept) Residual
> StdDev:     0.0002584018 2.731378
>
> Correlation Structure: AR(1)
>  Formula: ~1 | trt/flask
>  Parameter estimate(s):
>       Phi
> 0.2048944
> Fixed effects:  list(Vmax ~ trt, Km ~ trt)
>                     Value Std.Error  DF   t-value p-value
> Vmax.(Intercept) 15.394469  1.082697 118 14.218627  0.0000
> Vmax.trtlow      -1.660245  2.381505 118 -0.697141  0.4871
> Vmax.trtmed      -3.555246  1.503682 118 -2.364361  0.0197
> Km.(Intercept)    5.381378  1.873000 118  2.873133  0.0048
> Km.trtlow        11.448682  8.044641 118  1.423144  0.1573
> Km.trtmed        -0.381246  3.147606 118 -0.121123  0.9038
>  Correlation:
>               Vm.(I) Vmx.trtl Vmx.trtm Km.(I) Km.trtl
> Vmax.trtlow    -0.455
> Vmax.trtmed    -0.720  0.327
> Km.(Intercept)  0.726 -0.330   -0.523
> Km.trtlow      -0.169  0.876    0.122   -0.233
> Km.trtmed      -0.432  0.196    0.734   -0.595  0.139
>
> Standardized Within-Group Residuals:
>        Min         Q1        Med         Q3        Max
> -2.0222398 -0.7529003 -0.2362146  0.4364407  3.2055101
>
> Number of Observations: 132
> Number of Groups:
>           trt flask %in% trt
>             3             9
```

8.6 EXAMPLE: THE GROWTH RATE OF FISH (NLMM)

The von Bertalanffy model (Equation 4) is used to describe the growth patterns of animals over time. For example, in a fish growth study, we measure the length of individual fish at regular intervals as the fish ages. We can estimate growth parameters specific to the fish species by fitting the von Bertalanffy model to these length-at-age data

The model is given by:

$$L(t) = L_\infty \left(1 - e^{-k(t-t_0)}\right) \tag{4}$$

TABLE 8.5. The Atlantic Cod data set with 30 fish and 15 years of growth data (showing only the top and bottom three rows).

Fish ID	Age (yr)	Length (cm)
1	0.0	5.3
1	0.5	16.8
1	1.0	27.2
30	14.0	115.4
30	14.5	116.0
30	15.0	116.5

Where:

- $L(t)$ is the length of the fish at time t.
- L_∞ is the asymptotic length, representing the theoretical maximum length that the individual would reach if it grew indefinitely.
- k is the growth coefficient, indicating the rate at which the growth of the fish approaches its maximum size. A higher k value means it reaches its asymptotic length more quickly.
- t_0 is the hypothetical age at which the individual's length would be zero according to the model.

L_∞ (the asymptotic length) represents the length towards which the individual grows as time (t) approaches infinity. The concept behind L_∞ is that as the fish ages, its growth rate slows down and eventually approaches zero, with its length nearing the asymptotic value L_∞. k (the growth rate coefficient) determines how quickly the fish reaches its asymptotic length. Physiologically, k reflects the metabolic rates and general fitness of the fish, while ecologically, it can be influenced by environmental factors such as food availability and temperature. Lastly, t_0 (the theoretical age at zero length) is not directly observable in practice but provides a useful way to shift the growth curve along the time axis to provide a better fit to the data, especially in the early developmental stages.

Consider a study where the lengths of 30 Atlantic Cod, *Gadus morua*, in captivity are measured twice a year from hatching to 15 years. This creates a longitudinal dataset with repeated length measurements for each fish over time. In this experiment, we will focus on the growth patterns of individual fish, assuming they were raised under identical conditions. This allows us to attribute any growth differences to inherent biological variation among the fish. Apart from the repeated measures on individual fish, we will assume that the data are independent in all other respects.

The longitudinal nature of the data requires that we use appropriate statistical methods that account for the correlation among the repeated measures. We will use a nonlinear mixed-effects regression for the data in Table 1.

A plot of the data is shown in Figure 8.7; here, each line represents the growth trajectory of an individual fish over time.

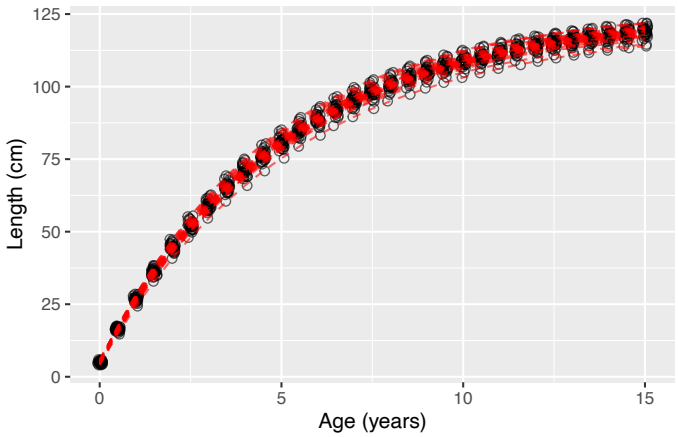


FIGURE 8.7. Plot of growth data measured in 30 Atlantic cod, *Gadus morua*.

```
> List of 1
>  $ legend.position: chr "none"
>  - attr(*, "class")= chr [1:2] "theme" "gg"
>  - attr(*, "complete")= logi FALSE
>  - attr(*, "validate")= logi TRUE
```

We will fit the von Bertalanffy growth model to the data using `nlme::nlme()` as follows, and the output is provided:

```
# von Bertalanffy growth function
vb_growth <- function(age, L_inf, k, t0) {
  L_inf * (1 - exp(-k * (age - t0)))
}

# Define the nonlinear mixed-effects model
nlme_model <- nlme(Length ~ vb_growth(Age, L_inf, k, t0),
                   data = vb_data,
                   fixed = L_inf + k + t0 ~ 1,                ①
                   random = L_inf + k ~ 1 | Fish_ID,          ②
                   groups = ~ Fish_ID,                        ③
                   correlation = corAR1(form = ~ 1),          ④
                   start = c(L_inf = 100, k = 0.2, t0 = -0.5))

# Print the summary of the model
summary(nlme_model)
> Nonlinear mixed-effects model fit by maximum likelihood
>   Model: Length ~ vb_growth(Age, L_inf, k, t0)
>   Data: vb_data
>         AIC       BIC  logLik
```

```
>   -2833.361 -2794.679 1424.68
>
> Random effects:
>  Formula: list(L_inf ~ 1, k ~ 1)
>  Level: Fish_ID
>  Structure: General positive-definite, Log-Cholesky parametrization
>          StdDev      Corr
> L_inf    1.857032547 L_inf
> k        0.008341198 -0.139
> Residual 0.555742464
>
> Correlation Structure: AR(1)
>  Formula: ~1 | Fish_ID
>  Parameter estimate(s):
>       Phi
> 0.9972623
> Fixed effects:  L_inf + k + t0 ~ 1
>           Value Std.Error  DF  t-value p-value
> L_inf 124.80230 0.3551041 898 351.4527       0
> k       0.20042 0.0015299 898 131.0050       0
> t0     -0.20415 0.0040574 898 -50.3164       0
>  Correlation:
>    L_inf  k
> k  -0.137
> t0 -0.260 -0.007
>
> Standardized Within-Group Residuals:
>        Min         Q1        Med         Q3        Max
> -2.2053004 -0.7552048  0.2402975  0.4902483  1.9150860
>
> Number of Observations: 930
> Number of Groups: 30
```

① The fixed effects are the parameters of the von Bertalanffy growth model which are invariant among fish.
② The random effects are the asymptotic length and growth rate to account for the intrinsic differences among fish.
③ The grouping variable is the fish ID.
④ The correlation structure is autoregressive of order 1 to account for the correlation among repeated measures within the same fish, the `~ 1` indicates that the order of the observations in the data must be used along which measurements are serially correlated, and since no grouping variable is provided, all fish will have the same correlation structure.

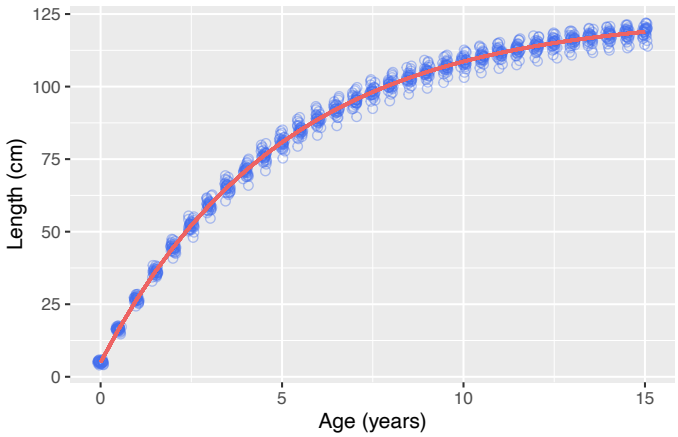


FIGURE 8.8. Fit of the von Bertalanffy model to experimental data obtained from 30 Atlantic Cod individuals.

8.7 SCRATHPAD

8.7.1 *To include in the article*

- **Assumptions:** Not necessary for simply estimating model parameters, but if the model is used for prediction or inference, it is important to state the assumptions of the model (e.g., linearity, homoscedasticity, independence of residuals) and test them.
- **i.i.d:** The residuals are assumed to be independent and identically distributed (i.i.d.), which is a common assumption in linear regression models. For a normal distribution, this is written as $\epsilon_i \sim N(0, \sigma^2)$, where σ^2 is the variance of the residuals.

8.7.2 *Contuinuing the MM model*