The Biostatistics Book

AJ Smit

2024-05-12

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

```
  1 + 1
```

```
[1] 2
```

Chapter 1

Introduction

1.1 The Scientific Method in Practice

Answering questions about the natural world using a scientific workflow requires that we draw on many years' of accumulated knowledge and experience. The workflow unpacks into roughly the following sequence of steps:

1. Look around you at the world, be curious about it, and ask questions to figure out an explanation for the pattern or phenomenon that tickled your interest.
2. Create an unambiguous statement of the question you want to answer, think about what is causing the pattern or phenomenon you observed, and how you might go about measuring the response (the thing you observed initially).
3. Translate this question into a testable hypothesis. This is the statement that you can test using the data you will collect.
4. Design an experiment or sampling campaign to collect data that will allow you to test this hypothesis. Clearly understand what the data you'll collect will look like, both for the response and the explanatory variables. For example, do you have a categorical or continuous predictor, is the response continuous, binary, ordinal, etc.? For this, you should have a firm grasp of the various kinds of Data Classes and Structures in R.
5. Think deeply about any confounding influences that might affect your data, and specify exactly what additional data you will have to collect to isolate the hypothesised influence in your analysis. You need to fully understand all the ways that factors not considered in your hypothesis might affect your study's outcome. Omissions cannot be rectified after the fact without repeating the entire experiment or sampling work. It requires knowledge and experience to avoid confounding influences ruining your work.
6. Depending on your experiment's design (4) and the nature of the data you'll obtain (4, 5), choose the appropriate statistical methods to analyse them. You should be able to develop a good idea of what statistical methods you'll use—even before the experiment has been done! Decide on the parametric test, or, should the statistical god with the dice not provide an outcome that favours your expectations, you can also decide upfront on a non-parametric equivalent. It is important not to decide on the statistical method after you've collected the data. This is called *p*-hacking, and it is almost a cardinal sin in science.
7. Do the experiment or go out into the world to sample, and collect the data. Have fun—this

is why we do science, afterall!
8. Go have a few drinks after a hard day's work and celebrate your success.
9. Analyse your newly-collected data. This will include explaratory data analyses (see Exploring With Summaries and Descriptions and Exploring With Figures), and then the application of the statistical methods you chose in step 6.
10. Communicate your results in tables and figures.

This textbook deals with many of these steps (except for 1, 5, and 7). This knowledge is codified in the form of the statistical method, which provides a systematic framework for collecting,[1] analysing, and interpreting data. In this chapter, I introduce the fundamental concepts of inferential statistics, which allow us to make inferences about populations based on sample data. I also provide an overview of the types of statistical methods used in inferential statistics, and discuss the importance of understanding the assumptions underlying these methods.

1.2 The Statistical Toolbox

With inferential statistics you can analyse data obtained from representative samples to draw conclusions or test hypotheses about populations or processes. I broadly categorise these methods into four main types, each serving different research applications[2]:

1. **Hypothesis Tests**: These parametric and non-parametric techniques assess whether sample data provide evidence for or against a specific claim (hypothesis) about population parameters such as their means, proportions, variances, or correlations between variables. Common hypothesis tests include:
    - Comparisons of group means or medians for a continuous variable (e.g., *t*-tests, ANOVA, Mann-Whitney *U* test)
    - Comparisons of group proportions for a categorical variable (e.g., χ-square test, Fisher's exact test)
    - Assessments of the relationship between two continuous or ordinal variables (e.g., Pearson's correlation, Spearman's rank correlation)
2. **Regression Analysis**: Regression with its parametric and non-parametric offerings lets us analyse the relationship between a response variable and one or more predictor variables. Regression models estimate coefficients representing the predictor effects, allow for prediction of the response, and enable hypothesis tests on the predictors. Common regression models include:
    - Linear regression for continuous response variables
    - Logistic regression for binary response variables
    - Generalised linear models (GLMs) for non-normal response variables
    - Various non-linear regressions for complex relationships, such as generalised additive models (GAMs)
3. **Survival Analysis**: Methods like the Kaplan-Meier estimator and Cox proportional hazards model analyse time-to-event data, where the interest lies in modelling the waiting times until certain events occur. I do not cover survival analysis in this book or any of my modules.

[1] Yes, statistics also informs us about how to collect data.

[2] This categorisation reflects my teaching approach, based on the order in which I think topics need to be covered, rather than a strict classification by statisticians. It is intended to provide a high-level overview of the types of statistical methods used in inferential statistics.

4. **Multivariate Analysis:** This includes an assortment of methods to analyse multiple response and predictor variables simultaneously. Dimension reduction methods, such as canonical correlation analysis (CCA) and non-metric multidimensional scaling (nMDS), help simplify complex datasets by identifying key patterns and relationships. Classification, including cluster analysis, is used to group similar observations together based on their characteristics. Multivariate approaches make fewer assumptions about the data's distribution, and there are techniques to deal with parametric and non-parametric data types (often without discrimination). Although these methods are not covered in this textbook, they are taught in my Quantitative Ecology module, which will eventually be developed into its own textbook.

The above methods include parametric or non-parametric (sometimes called 'robust') methods. Parametric methods assume that the data follow a specific distribution (e.g., normal, Poisson), while non-parametric methods make fewer assumptions about the data distribution. I will cover the parametric methods first, in Part A, followed by non-parametric methods in Part B. Part C of the book will look at semi-parametric methods, which combine aspects of both parametric and non-parametric approaches.

1.3 I. Parametric Methods (Known Distribution)

Parametric statistics rely on specific assumptions about the underlying probability distribution of the population from which the sample data were drawn. Biologists are taught that our data must be normally distributed, but this an unreasonable expectation considering the widely varying data sources we will encounter. Some biological processes simply do not generate normally distributed data!

Nevertheless, parametric statistics have through convention (rather than best practice) become the starting point for introductory forays into statistics. This is not terrible, because, should we be fortunate enough to have normally distributed data, parametric methods are more powerful than their non-parametric counterparts; however, they are also more sensitive to violations of some assumptions about our data.

The staple parametric statistics, such as the *t*-test, ANOVAs, Pearson correlation, and simple linear regression, require that two key assumptions are met: i) that our data (or sometimes the residuals) are **normality distributed** and ii) that the **variances are homoscedastic**. Section X is devoted to statistical tests that we may use to test the assumptions. However, because of the Central Limit Theorem, parametric methods can withstand moderate violations of the normality assumption when the sample size is large.[3]

A common mistake biologists makes is to think that parametric tests only apply to normal data. This is not true. Generalised linear models (GLMs) extend the statistical framework to accommodate non-normal error distributions, such as Poisson for count data or binomial for binary outcomes.[4] GLMs require that the distribution of the response variable belongs to the exponential family of distributions and that a suitable link function is chosen to connect the mean of the response to the linear predictor. Therefore, the defining characteristic of parametric methods is the assumption of a **known distribution** for the response variable, not necessarily that is is normal.

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

[4] The independent and dependent variables are also called the predictor and response variables, respectively. The predictor is often under the experimenter's control (in which case it is a fixed effects model), while the response is the variable predicted to respond in the manner hypothesised.

Within the parametric statistics framework, we can divide the methods into four groups depending on the type of question we are asking. We can ask questions about i) **difference in means**, ii) **differences in proportions**, iii) **relationships between variables**, or iv) the **effect of one or more predictors on a response variable**.

1.3.1 A. Hypotheses About the Means of Groups

The simplest form of comparison is to test whether the sample **means** of two or more groups differ.[5] Although this seems quite unimaginative, comparisons of the measures of central tendency are very common statistical tests in biology. Because this concept is so simple to understand, it serves as a good starting point for learning about hypothesis testing and the interpretation of the statistics which tell us about the strength of the evidence for or against our hypotheses.

You might have hypotheses that require you to compare the means of the outcomes of different experimental treatments, differences in the number of sea urchins among populations of kelp, or the number of species within replicate samples taken from different vegetation types. Look at some of the following examples to see if any of them resonate with your own research question, and then use this as a guide to find the appropriate statistical test in this book.

One-Sample *t*-Test (Section X.X.X)

Example: Is the mean height of a sample of *Protea* sp. grown in a specific experimental landscape (given below) different from the known (established *a priori*) average height of the same species (163.3 $\pm$ 15.5 cm) in the general population?

```
   Height
1     150
2     152
3     148
8     150
9     149
10    148
```

The example requires that you have *one normally-distributed continuous outcome variable* with *independent observations* and that you want to compare its mean value against a known population mean established *a priori*.

In this case, you'll want to use the R function `t.test()`. Since this function can accommodate data with equal or unequal variances[6] via the `var.equal` argument, you only need to assure the data are normally distributed. The test can be one-sided or two-sided. Alternatively, consider non-parametric alternatives, such as the Wilcoxon signed-rank test.

Two-Sample *t*-Test (Section X.X.X)

Example: Is the average number of leopard cubs born per female leopard in the Overberg region different from that in the Cederberg region? The dataset is:

[5] If *Y* not independent across the range of *X*, use a different type of regression model, such as a linear mixed-effects model.

[6] The dependent variable can also be ordinal, but this is less common. If this is the case, use *ordinal (logistic) regression instead.

```
      Region Cubs_Per_Female
1   Overberg               2
2   Overberg               3
3   Overberg               2
18 Cederberg               3
19 Cederberg               2
20 Cederberg               1
```

This requires that we obtain *two samples of continuous, normally-distributed measurements*. In other words, our experiment or sampling campaign will include two groups (sometimes two treatments, other times a treatment and a control) and we collect a sample of measurements of the response in both of them. This is again catered for by the `t.test()` function, and, as before, we don't have to fuss too much about the variances as equal and unequal variances can be accommodated. If the normality assumption is not met, consider a non-parametric alternative such as the Mann-Whitney U test.

A variant of the two-sample *t*-test is the paired *t*-test, which is used when the two samples are related (not independent); for example, the same individuals are measured before and after applying a treatment.

Analysis of Variance (ANOVA) for >2 Samples (Section X.X.X)

Example: Is the chirp rate of bladder grasshoppers different between the four seasons?

Table 1.1: Chirp Rate Data for Bladder Grasshoppers Across Four Seasons

	Season	Chirp Rate
1	Spring	17.7
2	Spring	13.9
3	Spring	15.7
58	Winter	10.2
59	Winter	4.0
60	Winter	10.6

We have *three or more samples of continuous, normally-distributed observations*. These data must also have more-or-less equal variances, so the homoscedasticity assumption is important. The `aov()` function in R is used to perform the ANOVA, which can be one-way, two-way, a repeated measures ANOVA, or an ANCOVA.[7] If the normality or homoscedasticity assumptions are not met, consider non-parametric alternatives, such as the Kruskal-Wallis test, or try transforming the data.

[7] A repeated measures ANOVA is used when the same subjects are measured at different time points or under different conditions. A two-way ANOVA is used when there are two independent variables (there are also higher-order ANOVAs but they become more of a pain to interpret and require cumbersome experimental designs). An ANCOVA is used when you want to compare the means of groups while controlling for the effect of a continuous covariate. There are many kinds of ANOVA designs and each relates to specific experimental designs well beyond the scope of this book. Tony Underwood provides a pedantic overview of ANOVA designs in his book *Experiments in Ecology* (Underwood 1997) if you really want to go there.

Table 1.2: Foraging time and diving depth of African penguin.

Sex	Foraging time (hr)	Diving depth (m)
Male	1.2	10
Male	1.5	15
Male	1.8	20
Female	2.0	25
Male	2.3	30
Female	2.5	35
Female	2.8	40
Female	3.0	45
Male	3.3	50
Male	3.5	55

Analysis of Covariance (ANCOVA)* (Section X.X.X)

Example: We have a set of data about African penguins and we want to determine if there are differences between male and female penguins in terms of their mean foraging time, and if that difference is influenced by their diving depth. The dataset is as follows:

In this example, we are interested in the mean foraging time of male and female penguins, controlling for their diving depth. An ANCOVA focuses on the differences in means (the categorical variable), and the continuous covariates (diving depth) is specifically controlled for to remove its effect from the dependent variable. This reduces the error variance and so more accurately assesses the comparison of group means. The assumptions of normality and homoscedasticity apply. The functions `aov()` accommodates the categorical and continuous predictors.

Multivariate Analysis of Variance (MANOVA)

MANOVAs are similar to ANOVAs, except here you have *multiple dependent variables*, all *independent*, *continuous*, *and normally-distributed*. This is useful when you want to compare the means of multiple groups across multiple dependent variables. For example, you might want to compare the average foraging time together with diving depth of African penguins in three colonies (two in South Africa and one in Namibia) around the coast. The `manova()` function in R is used to perform a MANOVA and there are similar variants to what we have seen in ANOVA.

1.3.2 B. Hypotheses About the Proportions of Groups

You can compare the proportions of groups using tests for proportions when the outcome variable is binary (e.g., success/failure, presence/absence, up/down, day/night). These tests are used to determine if the proportion of successes differs between groups. Use the following tests to compare group proportions:

One-Sample Test for Proportions

Example: Is the proportion of African penguins foraging in a specific colony different from the known proportion of the same species in the general population? The data might look like this:

- `Sample data: 55 of the 100 penguins observed were foraging in a specific colony`
- `The known proportion of penguins foraging in the general population is 60%`

In this scenario, we are comparing the proportion of a single sample (the proportion of foraging African penguins in a specific colony) to a known population proportion. The data must consist of a *binary outcome variable* (e.g., foraging vs. not foraging) and the observations must be independent. The `prop.test()` function in R is used to perform this test, which can be either one-sided or two-sided. If the requirement of independent observations is not met, consider non-parametric alternatives, such as the sign test.

Two-Sample Test for Proportions

Example: Is the proportion of endangered sea turtles successfully reaching the ocean different between two beaches? Here are data:

Table 1.3: Number of Sea Turtles Reaching the Ocean on Two Beaches

Beach	Successes	Observed
Beach A	75	100
Beach B	65	120

Here we compare the proportions from two independent samples (e.g., the proportion of sea turtles successfully reaching the ocean on Beach A versus Beach B). As before, the data yield *a binary outcome* (e.g., reached the ocean vs. did not reach the ocean) for each group, and the observations within each group are independent. The `prop.test()` function is used it has one-sided or two-sided options. If the sample sizes are small or expected frequencies are low, consider using Fisher's exact test instead of the proportion test. If the assumption of independent observations within groups is violated, you may need to consider methods that account for dependency in the data, such as Generalised Estimating Equations (GEE) or mixed-effects models.

Chi-square Test for Count Data

Example: Is there an association between vegetation type and the presence of leopards in different areas of Kruger National Park? A hypothetical dataset:

Table 1.4: Contingency Table of Plant Species and Insect Occurrence

	Presence	Absence
Grassland	20	30
Woodland	25	40
Shrubland	35	15

Here we examine the relationship between two categorical variables (vegetation type and leopard presence) within Kruger National Park. The data are organised into a contingency table, where each cell represents the count or frequency of observations for a specific combination of categories. The

chi-square test of independence is used to determine if there's a significant association between the variables.

As with other categorical tests, the data yield *discrete outcomes* (e.g., savanna, woodland, or riverine for vegetation type; present or absent for leopard presence). The observations should be independent, meaning the presence of a leopard in one area should not influence its presence in another.

The `chisq.test()` function in R is commonly used for this analysis. This test compares the observed frequencies in each cell of the contingency table to the frequencies that would be expected if there were no association between vegetation type and leopard presence.

If the sample size is large and the expected frequencies in each cell are adequate (typically > 5), the chi-square test is appropriate. However, if the sample size is small or if there are cells with low expected frequencies, consider using Fisher's exact test instead.

If the assumption of independence is violated (e.g., if the data include multiple observations from the same leopard individuals or territories), you may need to consider more advanced methods that account for dependency in the data, such as log-linear models or Generalised Estimating Equations (GEE).

Fisher's Exact Test

Example: Is there a significant association between the presence of certain plant species and the occurrence of rare fynbos endemic insects in the Cape Floristic Region? Here are the data:

Table 1.5: Contingency Table of Plant Species and Insect Occurrence

	Present	Absent
Plant A	2	8
Plant B	3	7

Fisher's Exact Test is used when we have two categorical variables and want to determine if there's a significant association between them, particularly when sample sizes are small or when we have sparse data in some categories. This test is especially useful in ecological studies where rare species or events are being investigated.

In this example we examine the relationship between the presence of specific plant species and the occurrence of rare fynbos endemic insects. The data are organised into a 2x2 contingency table, where each cell represents the count of observations for a combination of presence/absence of the plant species and the insect species.

The test calculates the exact probability of observing the given set of cell frequencies under the null hypothesis of no association. It does not rely on approximations and it more accurate than the chi-square test for small samples. Use the `fisher.test()` function to perform this analysis. Like other categorical tests, the observations should be independent, meaning the presence of an insect in one area should not influence its presence in another.

Fisher's Exact Test is particularly appropriate when:

- The total sample size is less than 1000

Table 1.6: Foraging time and diving depth of African penguin.

Foraging time (hr)	Diving depth (m)
1.2	10
1.5	15
1.8	20
2.0	25
2.3	30
2.5	35
2.8	40
3.0	45
3.3	50
3.5	55

- The expected frequency in any cell of the contingency table is less than 5
- You're dealing with rare events or species

If the sample size becomes very large, Fisher's Exact Test can become computationally intensive, and the chi-square test may be more practical.

If the assumption of independence is violated (e.g., if the data include multiple observations from the same locations over time), you may need to consider more advanced methods that account for dependency in the data, such as mixed-effects models or Generalised Estimating Equations (GEE).

1.3.3 C. Hypotheses About the Strength of Association

Example: Is there a relationship between the foraging time and diving depth of African penguins?

You'll want to use a Pearson's correlation to determine if there is a linear relationship between *two continuous variables*, both of them normally distributed and homoscedastic. A correlation analysis does not presume causation and does not provide a predictive model, both of which are the domain of regression. The strength of the relationship is quantified by the correlation coefficient called Pearson's rho, which ranges from -1 to 1. Use the `cor.test( ... , method = "pearson" )` function in R to perform this analysis.

Non-parametric alternatives such as the Spearman's rank correlation or Kendall's tau correlation (see 'II. Non-Parametric Methods') are available and implemented with the same R function.

1.3.4 D. Modelling and Predicting Causal Relationships

The relationship between one or a few predictors and an outcome can be represented by a function, which is a model that reconstructs part of the 'reality' of the observed phenomenon. Regression analysis helps you understand how changes in the continuous predictor variable(s) drive changes in a continuous outcome variable. The model quantifies the strength of the associations and makes predictions for new data points. You may use regression models for hypothesis testing and for identifying which predictor variables have the most substantial impact on the outcome.

Simple Linear Regression

Example: The same dataset of foraging time and diving depth of African penguins can be used to model the relationship between these two variables. Does diving depth depend on foraging time?

What is different now is that we are interested in *predicting the diving depth* (response) of penguins based on their foraging time (predictor). Assuming there is a linear response, we can use a simple linear regression model to quantify the relationship between these two continuous variables. The model provides an equation that describes how the diving depth changes as the foraging time increases. The assumptions of normality and homoscedasticity apply to the residuals, and are accessed after having fit the model.

This calls for a simple linear regression model and you can fit it using the `lm()` function in R. The model can also be specified as a generalised linear model (GLM) with `glm( ... , family = gaussian)`.

If assumptions fail, apply data transformations (e.g., log, square root), robust regression (`rlm()` in **MASS** package), or consider non-linear models.

Polynomial Regression

I'll not provide an example here. It suffices to say that a polynomial regression is effectively a simple linear regression that allows for a curvilinear relationship between the predictor and the outcome. To accomplish this, the model includes polynomial terms (e.g., quadratic, cubic, which are simply powers of the predictor) to capture the non-linear patterns in the data. The model can be fit using the `lm()` function in R.

Assess the relationship between x vs. y by making a scatterplot of the data and eye balling a best fit curve through the scatter of points. Is the line curvy or bendy? Do you know in advance if a more complicated model describes the response? If the answer is 'yes' to the first and 'no' to the second question, then a polynomial regression might be just the thing for you.

Multiple Linear Regression (MLR)

Example: I've added a second predictor to the dataset of foraging time and diving depth of African penguins. Does diving depth depend on the penguins' body mass index (BMI) and foraging time?

The only difference between this example and the simple linear regression is that we now have two predictors (foraging time and BMI) instead of one. The predictors can be *continuous* (as in the example) *and/or categorical*. If you are more concerned with the means of the categorical variables, consider an ANCOVA as an alternative option. The multiple linear regression model can be extended to include interaction terms between predictors. You can quantify the relationship between both predictors and the outcome simultaneously, and ask which of the two best predicts the response. The same assumptions apply as in the simple linear regression and we hope for a linear relationship between x_1 and x_2 vs. y. Other considerations are provided in the chapter on MLR.

The R functions `lm()` and `glm( ... , family = gaussian)` accommodate situations such as these where we have multiple predictors.

Table 1.7: Foraging time and diving depth of African penguin.

BMI	Foraging time (hr)	Diving depth (m)
1.2	1.2	10
1.5	1.5	15
1.8	1.8	20
2.0	2.0	25
2.3	2.3	30
2.5	2.5	35
2.8	2.8	40
3.0	3.0	45
3.3	3.3	50
3.5	3.5	55

Generalised Linear Models (GLM)

GLMs are a class of regression models that extend the simple linear regression framework to accommodate various types of response distributions. As such, they can accommodate data that violate the assumptions of normality and homoscedasticity, as well as situations where the response variable is not continuous.

Use GLMs to model count data (e.g., number of occurrences), binary outcomes (e.g., success/failure), and other non-continuous response variables that cannot be adequately represented by a normal distribution. Unlike linear models, which assume a normal error distribution, GLMs specify the distribution of the response variable using a probability distribution from the exponential family, such as the Gaussian (normal), binomial, Poisson, or negative binomial distributions.

GLMs incorporate a link function that relates the linear predictor (a linear combination of the predictor variables) to the expected value of the response variable. This link function can take various forms, including the identity (linear), logit (for binary data), probit, or other transformations, depending on the nature of the response variable and the desired relationship between the predictors and the outcome.

The `glm()` function is a staple for fitting GLMs. It is designed to handle the exponential family distributions and will allow you to specify the appropriate distribution and link function for your data and research question. A few common types of GLMs are presented next.

Logistic Regression (Chapter 6)

You'll encounter binomial data in experiments or processes with binary outcomes, such as presence/absence, success/failure, or alive/dead. To model this type of data, you will want to use logistic regression. Logistic regression estimates the log-odds of the outcome as a linear combination of the predictor variables. The logistic function is then used to convert these log-odds into probabilities, which range from 0 to 1, so it is suitable for predicting the likelihood of the binary outcomes.

- **Use When:** You have a binary outcome variable and want to model the relationship between predictors and the probability of the outcome.
- **Data Requirements:** Binary outcome, continuous or categorical predictors.
- **Assumptions:** Linear relationship between the log-odds of the outcome and predictors.

- **Diagnostics:** Check for influential observations, multicollinearity, and overall model fit.
- **If Assumptions Fail:** Consider interactions, alternative link functions (probit, complementary log-log) in `glm()`, or non-linear logistic regression, zero-inflated models when excess zeroes.
- **R Function:** `glm( ... , family = binomial)`
- **Model Selection:** Stepwise regression, regularisation techniques, information criteria (AIC, BIC).

Poisson Regression (Chapter 6)

Typical examples of count data include the number of offspring, parasites, or seeds. Poisson regression is used to model the relationship between predictors and the count outcome. The model assumes that the count data follow a Poisson distribution, where the mean and variance are equal. Poisson regression is suitable for data with a single count outcome.

- **Use When:** You have count data and want to model the relationship between predictors and the count outcome.
- **Data Requirements:** Count outcome, continuous or categorical predictors.
- **Assumptions:** Equidispersion (variance equals the mean).
- **Diagnostics:** Check for overdispersion, excess zeros, and overall model fit.
- **If Assumptions Fail:** Negative binomial regression (`glm.nb()` in the **MASS** package, overdispersion), zero-inflated models (`zeroinfl()` in the **pscl** package, excess zeros).
- **R Function:** `glm( ... , family = poisson)`

Negative Binomial Regression

Negative binomial regression is an extension of Poisson regression that accommodates overdispersion, where the variance exceeds the mean. It is used when the count data exhibit more variability than expected under a Poisson distribution. The model assumes that the count data follow a negative binomial distribution, which has an additional parameter to account for overdispersion. Biological and ecological processes such as species abundance, parasite counts, and gene expression often exhibit overdispersion.

- **Use When:** You have count data with overdispersion and want to model the relationship between predictors and the count outcome.
- **Data Requirements:** Count outcome, continuous or categorical
- **Assumptions:** Overdispersion (variance exceeds the mean).
- **Diagnostics:** Check for overdispersion, excess zeros, and overall model fit.
- **R Function:** `glm.nb()` in **MASS** package

Gamma Regression

Gamma regression is for modelling continuous, positive outcomes that exhibit a right-skewed distribution and possibly also a non-constant variance (heteroscedasticity). The gamma distribution is well suited for continuous measurements where the variability increases as the mean increases. You might encounter this kind of distribution in growth rates, enzyme activity levels, species abundance data, and other phenomena or processes characterised by positive, skewed data.

- **Use When:** You have a continuous, positive outcome and want to model the relationship between predictors and the outcome.
- **Data Requirements:** Continuous, positive outcome, continuous or categorical predictors.
- **Assumptions:** Outcome values are positive, potentially non-constant variance.
- **Diagnostics:** Check for overall model fit, influential observations, and residual
- **R Function:** `glm( ... , family = Gamma)`

Beta Regression

Beta regression is a statistical technique appropriate when the response variable is a continuous proportion or rate bounded between 0 and 1. These types of data might, for example, arise in ecology where one might study the proportions of time animals spend exhibiting different behaviours, the relative abundances of species in a community, or the proportions of habitat patches comprising a landscape. Proportional data inherently exhibit heteroscedasticity (non-constant variance).

- **Use When:** You have a proportional outcome ($0 < y < 1$) and want to model the relationship between predictors and the outcome.
- **Data Requirements:** Proportional outcome ($0 < y < 1$), continuous or categorical predictors.
- **Assumptions:** Outcome values within $(0, 1)$, potentially non-constant variance.
- **Diagnostics:** Check for overall model fit, influential observations, and residual analysis.
- **If Assumptions Fail:** Transformations, consider alternative link functions, or zero/one-inflated beta regression.
- **R Function:** `betareg()` in the **betareg** package

Modelling Non-Linear Relationships

We use non-linear models when the relationship between predictor variables and the outcome variable is not linear. This non-linearity arises from the predictor variables themselves being non-linearly related to the outcome or from the model's parameters (coefficients) appearing non-linearly in the functional form. The visualised response curve is typically curved, rather than a straight line. These models are often derived from theoretical understanding or prior knowledge about the underlying mechanisms governing the relationship between the predictors and the outcome variables.

Non-Linear Least Squares (NLS) Regression (Chapter 7)

- **Use When:** The relationship between the predictors and the outcome is non-linear.
- **Data Requirements:** Continuous outcome, continuous predictors.
- **Assumptions:** Appropriate functional form, normality, and homoscedasticity of residuals.
- **Diagnostics:** Check residual plots, normality of residuals, and leverage/influence points.
- **R Function:** `nls()` (for non-linear regression models with user-specified functions)

Generalised Non-Linear Models (GNLMs)

GNLMs are an extension of generalised linear models (GLMs) that allow for non-linear relationships between the predictors and the outcome variable. GNLMs are used when the relationship between the predictors and the outcome is non-linear, and the outcome variable follows a non-normal distribution. GNLMs are particularly useful for count data, binary outcomes, and other non-continuous response variables that exhibit non-linear relationships with the predictors.

Linear and Non-Linear Hierarchical Models (Mixed-Effects Models)

Hierarchical models are used when data are structured hierarchically, such as when multiple observations are nested within higher-level units (e.g., plants within fields, sheep within rangelands). These models account for the correlation between observations within the same group and allow for the estimation of both fixed effects (population-level parameters) and random effects (group-level parameters). Hierarchical models are also known as multilevel models or mixed-effects models.

Linear Mixed-Effects Models (LMMs) (Section X.X.X)

- **Use When:** You have nested or hierarchical data structures and the relationship between the predictors and the outcome is linear.
- **Data Requirements:** Continuous outcome, continuous predictors, potentially with nested or hierarchical data structures.
- **Assumptions:** Normality, homoscedasticity of residuals, correct specification of random effects structure.
- **If Assumptions Fail:** Consider transformations, robust regression, or non-linear mixed-effects models.
- **Diagnostics:** Check residual plots, normality of residuals, and leverage/influence points, assess random effects structure.
- **R Function:** `lmer()` in the **lme4** package (for linear mixed-effects models with user-specified functions)

Non-Linear Mixed-Effects Models (NLMMs) (Chapter 7)

- **Use When:** You have nested or hierarchical data structures and the relationship between the predictors and the outcome is non-linear.
- **Data Requirements:** Continuous outcome, continuous predictors, potentially with nested or hierarchical data structures.
- **Assumptions:** Appropriate functional form, normality, and homoscedasticity of residuals, correct specification of random effects structure.
- **If Assumptions Fail:** Generalised non-linear mixed models (GNLMMs) and generalised additive mixed models (GAMMs) can be used when the assumptions of non-linear mixed models (NLMMs) are violated. Else, consult a statistician.
- **Diagnostics:** Check residual plots, normality of residuals, and leverage/influence points, assess random effects structure.
- **R Function:** `nlme()` in the **nlme** package (for non-linear mixed-effects models with user-specified functions)

Generalised Linear and Non-Linear Mixed-Effects Models (GLMMs and GNLMMs)

GLMMs and GNLMMs combine the flexibility of regression model generalisation (i.e. by accommodating non-Gaussian distribution families) with the ability to account for nested or hierarchical data structures. GLMMs are used when the outcome variable is not normally distributed (a different, known distribution) and the data are structured hierarchically. GLMMs include both fixed effects (population-level parameters) and random effects (group-level parameters) and can accommodate a wide range of outcome distributions, including binary, count, and continuous outcomes.

- **Use When:** You have non-normally distributed outcome data and nested or hierarchical data structures.
- **Data Requirements:** Binary outcome, continuous or categorical predictors, potentially with nested or hierarchical data structures.
- **Assumptions:** Linear relationship between the log-odds of the outcome and predictors, correct specification of random effects structure.
- **Diagnostics:** Check residual plots, normality of residuals, and leverage/influence points, assess random effects structure.
- **R Function:** `glmer()` in the **lme4** package

Other Regression Models

Zero-Inflated Models

- **Use When:** You have count data with an excess of zeros and want to model the zero-inflation separately from the count process.
- **Data Requirements:** Count outcome, continuous or categorical
- **Assumptions:** Correct specification of zero-inflation and count processes, no omitted variables.
- **Diagnostics:** Check zero-inflation and count process, overall model fit.
- **R Function:** `zeroinfl()` in the **pscl** package

Survival Analysis

- **Data Requirements:** Time-to-event outcome, continuous or categorical predictors.
- **Assumptions:** Proportional hazards, non-informative censoring.
- **Diagnostics:** Check proportional hazards assumption, influential observations, and overall model fit.
- **R Function:** `survival::coxph()`

Time Series Analysis

- **Data Requirements:** Time-ordered data, potentially with autocorrelation.
- **Assumptions:** Stationarity, no autocorrelation in residuals.
- **Diagnostics:** Check autocorrelation, stationarity, and overall model fit.
- **R Function:** `arima()`, `auto.arima()` in the **forecast** package

Structural Equation Modelling (SEM)

- **Data Requirements:** Continuous outcome, continuous
- **Assumptions:** Correct specification of the structural model, no omitted variables, no measurement error.
- **Diagnostics:** Check model fit, parameter estimates, and overall model validity.
- **R Function:** `sem()` in the **lavaan** package

Bayesian Regression

- **Data Requirements:** Continuous outcome, continuous or categorical predictors.
- **Assumptions:** Correct specification of priors, likelihood, and model structure.
- **Diagnostics:** Check for convergence, posterior predictive checks, and overall model fit.
- **R Function:** `brms::brm()`

1.4 II. Non-Parametric Methods (Distribution-Free)

Non-parametric statistics are statistical methods that do not rely on assumptions about the specific form or parameters of the population distribution. They are also referred to as *distribution-free methods*. These methods often use ranks or other order statistics of the data rather than the actual data values themselves.

1.4.1 A. Hypotheses About Groups

One-Sample Tests for Medians

Use a one-sample test to compare the median of a single sample to a known population median. It is as an alternative to one-sample *t*-tests when the data do not meet the assumptions of parametric tests.

- Wilcoxon signed-rank test
- Sign test

Two-Sample Tests for Medians (Section X.X.X)

Use two-sample tests to compare the medians of two independent or related samples. Use it when the assumptions of parametric two-sample tests are violated.

- Mann-Whitney U test (two independent groups)
- Wilcoxon rank-sum test (two independent groups)
- Kruskal-Wallis test (multiple groups)
- Friedman test (related samples)

1.4.2 B. Hypotheses About Proportions

- *Chi-Square Test for Independence:* Comparing proportions of two groups

1.4.3 C. Correlation Analysis for Tests of Association

Use non-parametric correlation to assess the strength and direction of a relationship between two continuous (or ordinal) variables when the assumptions of parametric correlation tests cannot be met.

Spearman's Rank Correlation (Chapter 2)

A non-parametric measure of the strength and direction of association between two variables.

Kendall's Tau Correlation (Chapter 2)

A non-parametric measure of the strength and direction of association between two variables.

1.4.4 D. Regression Analysis

Quantile Regression (Section X.X.X)

Models different quantiles of the response distribution.

Robust Regression (Section X.X.X)

Less sensitive to outliers than ordinary least squares regression.

Kernel Density Estimation

KDE is a non-parametric method for visualising the distribution of a continuous variable. Unlike histograms, which bin data into discrete intervals, KDE creates a smooth curve that represents the estimated probability density function (PDF) of the underlying data. It does this by placing a kernel function (often a symmetric curve like a Gaussian or Epanechnikov) at each data point and summing up the contributions of these kernels across the entire range of the variable. The bandwidth of the kernel controls the smoothness of the resulting density estimate. Wider bandwidths lead to smoother curves but may obscure finer details, while narrower bandwidths reveal more local fluctuations but can be noisy. KDE is useful when the underlying distribution of the data is unknown or non-standard and it offers a convenient way to visualise and understand the shape and spread of the data without being constrained by parametric assumptions.

Local Regression (LOESS)

LOESS (Locally Estimated Scatterplot Smoothing) is a non-parametric regression technique that produces a smooth curve through a set of data points by fitting simple models to localised subsets of the data. It achieves this by weighting the data points in each subset, with higher weights assigned to points closer to the point being estimated. The model used for local fitting is typically a low-degree polynomial, although other choices are possible.

LOESS is primarily used for data exploration and visualisation. It is best known for smoothing scatterplots and revealing underlying trends or patterns in the data. It is advantageous because it doesn't assume any particular functional form for the relationship between the predictors and the response variable, so it to adapts to various data shapes. But LOESS does not provide a single, easily interpretable equation for the entire dataset, making it less suitable for making predictions or drawing global inferences. It can also be computationally demanding with large datasets as it fits separate models in the vicinity of locally-selected points.

Penalised Regression

Penalised regression (also known as regularisation) is used to enhance the performance of regression models. This might be desirable when dealing with high-dimensional data or when the predictor variables are highly collinear. It introduces a penalty to the regression objective function which discourages the model from having overly complex or large coefficients. This effectively prevents overfitting. Common types of penalised regression include Ridge regression (L2 regularisation), which adds the sum of the squared coefficients as a penalty term, and Lasso regression (L1 regularisation), which adds the sum of the absolute values of the coefficients. The penalty terms encourage simpler models by shrinking some coefficients towards zero, with Lasso potentially setting some coefficients exactly to zero, thus performing variable selection. The balance between fitting the data well and maintaining model simplicity helps in improving the model's generalisation to new data. Penalised regression methods can achieve a trade-off between bias and variance and result in more robust and interpretable models.

1.5 III. Semi-Parametric Methods

Semi-parametric methods combine parametric and non-parametric techniques to provide a balance between flexibility and efficiency. These methods are useful when the assumptions of parametric tests are violated, but the data do not meet the requirements for non-parametric tests. Semi-parametric methods are often more powerful than non-parametric tests, as they make fewer assumptions about the data distribution. These methods are particularly useful when the sample size is small or when the data are skewed or have outliers.

Generalised Additive Models (GAMs) (Chapter 10)

- **Use When:** You have non-linear relationships between predictors and outcome.
- **R Function:** `gam()` in the **mgcv** package; also `gamm4()` in the **gamm4** package
- **Data Requirements:** Continuous, binary, or categorical outcome, continuous or categorical predictors, potentially with nested or hierarchical data structures.
- **Advantages:** Flexible modelling of non-linear relationships using smoothing functions, can handle mixed-effects structures.
- **Limitations:** Interpretation can be challenging, potential overfitting.

Generalised Estimating Equations (GEEs)

- **Use When:** You have correlated data and non-normally distributed outcomes.

- **R Function:** `geeglm()` in the **geepack** package; also functions in the **gee** package
- **Data Requirements:** Correlated data, non-normal outcomes, continuous or categorical predictors.
- **Advantages:** Robust to misspecification of the correlation structure, can handle non-normal outcomes, flexible in handling missing data.
- **Limitations:** Assumes correct specification of the correlation structure, may be less efficient than mixed-effects models.

Semi-Parametric Survival Models

- **Use When:** You have time-to-event data and want to model the hazard function.
- **R Function:** `coxph()` in the **survival** package
- **Data Requirements:** Time-to-event data, censoring, continuous or categorical predictors.
- **Assumptions:** Proportional hazards assumption, independence of censoring.
- **Diagnostics:** Check proportional hazards assumption, influential observations, goodness

Spline Regression

- **Use When:** You have non-linear relationships between predictors and outcome.
- **R Function:** `lm()` with splines, `gam()` in the **mgcv** package
- **Data Requirements:** Continuous outcome, continuous predictors.
- **Assumptions:** Linearity within each spline, potentially non-constant variance.
- **Diagnostics:** Check for overall model fit, influential observations, and residual analysis.
- **If Assumptions Fail:** Transformations, consider alternative link functions, or penalised regression.

1.6 IV. Machine Learning Methods

Machine learning methods are a set of algorithms that can learn patterns from data without being explicitly programmed. These methods are particularly useful for prediction, classification, and clustering tasks. Machine learning models can handle complex relationships in the data and are often more flexible than traditional statistical models. However, they can be more computationally intensive and may require more data to train effectively.

Random Forests

A machine learning method that uses an ensemble of decision trees to predict an outcome.

Support Vector Machines

A machine learning method that finds the optimal hyperplane to separate two classes of data.

Ensemble Methods

A machine learning technique that combines the predictions of multiple models to improve accuracy.

Neural Networks

A machine learning method that uses interconnected nodes to model complex relationships in data.

Deep Learning

A subset of machine learning that uses neural networks with multiple layers to model complex relationships in data.

1.7 V. Miscellaneous Methods

Bootstrapping

A resampling method for estimating the sampling distribution of a statistic.

Permutation Tests

A non-parametric method for testing hypotheses by randomly permuting the data.

Monte Carlo Simulation

A method for estimating the distribution of a statistic by generating random samples from a known distribution.

Bayesian Methods

A statistical approach that uses Bayes' theorem to update prior beliefs based on observed data.

Dimensionality Reduction

Also called muitvariate analyses. A set of techniques for reducing the number of variables in a dataset while preserving important information.

Clustering

A set of unsupervised learning techniques for grouping similar data points together.

Feature Selection

A process for identifying the most important variables in a dataset for predicting an outcome.

Regularisation

See penalised regression. A technique for preventing overfitting by adding a penalty term to the model coefficients.

Cross-Validation

A method for estimating the performance of a model by splitting the data into training and test sets.

Hyperparameter Tuning

The process of selecting the optimal values for the parameters of a machine learning model.

Model Evaluation

The process of assessing the performance of a model using metrics such as accuracy, precision, recall, and F1 score.

Model Interpretation

The process of understanding how a model makes predictions by examining the relationship between the input variables and the output.

Model Deployment

The process of putting a trained model into production so that it can be used to make predictions on new data.

Model Monitoring

The process of tracking the performance of a deployed model over time to ensure that it continues to make accurate predictions.

Model Explainability

The process of explaining how a model makes predictions in a way that is understandable to humans.

Model Fairness

The process of ensuring that a model does not discriminate against certain groups of people based on sensitive attributes.

Model Robustness

The process of ensuring that a model performs well on new data that is different from the training data.

Part I

Parametric Methods

If the research question does not involve exploring the relationship between a response variable and predictor variables, then non-regression inferential statistical methods would be more appropriate. These include tests of means/medians, tests of proportions, correlation analysis, and nonparametric tests. These methods are suitable when the goal is to compare groups, assess central tendencies, test for differences, or measure the strength of association between two variables without explicitly modeling the relationship.

Chapter 2

Correlation

Chapter 3

Linear Regression

Linear models are frequently used statistical tools that all biologists should know. They describe and quantify relationships between variables and are widely employed to predict the value of a dependent variable (or response variable, *Y*) based on the values of one or more independent variables (or predictor variables, *X*). A linear model is an equation where the relationship between the dependent variable and the independent variables is linear in the parameters (though not necessarily in the variables themselves), allowing us to predict the dependent variable from the predictors. In statistics, models are mathematical representations or descriptions of real-world processes or systems. They offer idealised and simplified representations of reality and capture the essential features and relationships we find interesting.

Regression analysis is a statistical technique used to estimate the parameters of the model that best describes the relationship between a dependent variable and one or more independent variables. The primary goal of regression analysis is to fit the model to the observed data and offer insights into the strength and nature of the relationships between variables.

One of the simplest forms of linear models is the **simple linear model**, which is the topic of this chapter. A simple linear model estimates model parameters through the process of simple linear regression (SLR). SLR involves a single independent variable and is often applied when the independent variable is hypothesised to causally influence the dependent variable. However, a causal relationship is not a strict requirement. The primary goal of SLR may simply be to derive a formula (model) that predicts the values of the dependent variable based on the independent variable, regardless of whether a causal relationship exists between them.

SLR serves as a foundational regression technique that extends to more complex forms, including **polynomial regression** (Chapter 4), **multiple linear regression (MLR)** (Chapter 5), and **generalised linear models (GLMs)** (Chapter 6). Polynomial regression includes polynomial terms (higher powers of the independent variable, like X^2, X^3, etc.) to model curvilinear relationships, while MLR involves multiple independent variables to describe more complex relationships where the dependent variable is influenced by several predictors simultaneously. GLMs further extend these concepts to handle various types of dependent variables (besides responses drawn from the normal distribution) and relationships (e.g. logistic).

In cases where prediction is not the primary objective, and causation is neither expected nor implied, but one variable exhibits a systematic change with another, **correlation analysis** (Chapter 2)

is a more appropriate technique.

The terminology surrounding linear models and linear regression can sometimes be confusing because we often use terms like 'linear model,' 'linear regression,' and 'least squares regression' interchangeably. But 'linear model' is a broader term that encompasses various types of linear relationships, including simple linear models, multiple linear models, polynomial models, and GLMs. In this section, you will learn about simple linear models and regression analysis, which will provide you with the foundational knowledge to understand more complex linear models and regression techniques.

3.1 Simple Linear Regression

Linear models help us answer questions like:

- How does body mass change with age in a particular species?
- Does the number of offspring depend on the amount of food available?
- How does a species' geographic distribution change with temperature?

By assuming a linear relationship between variables, these models provide a clear and interpretable way to quantify and predict biological outcomes. For example, should a linear model describe the relationship between body mass (g) and age (years), we can predict the body mass of a particular species of fish would increase by 230 g for every additional year of age up to the age of five years (however, please see the von Bertalanffy model in Chapter 7.6).

The simple linear model is given by:

$$Y_i = \beta \cdot X_i + \alpha + \epsilon \tag{3.1}$$

Where:

- Y_i is the i-th measurement of the dependent variable,
- X_i is the i-th measurement of the independent variable,
- α is the intercept (the value of Y when $X = 0$),
- β is the slope (the change in Y for a one-unit change in X), and
- ϵ is the error term (residual; see box 'The residuals, ϵ_i').

The residuals, ϵ_i

In most regression models, such as linear regressions and those discussed in Chapter 7, we assume that the residuals are *independent and identically distributed (i.i.d.)*. This implies that each residual ϵ_i is drawn from the same probability distribution and that they are mutually independent. When the residuals follow a normal distribution, this can be expressed as $\epsilon_i \sim N(0, \sigma^2)$, where:

- ϵ_i represents the residual for the i-th observation,
- $N(0, \sigma^2)$ denotes a normal distribution with a mean of 0 and a variance of σ^2.

The requirement of a zero mean for residuals implies that, on average, the model's predictions neither systematically overestimate nor underestimate the true values. The constant variance assumption ensures that the spread or dispersion of residuals around the mean remains consistent across all levels of the predictor variables. This ensures that the model's accuracy is uniform across the range of data.

The requirement for independence indicates that the residual for any given observation is not influenced by or correlated with the residuals of other observations. It also means that the residual for an observation does not depend on the order in which the observations were collected (i.e. no serial correlation or auto-correlation). Independence ensures that each data point contributes unique information to the model and prevents any systematic patterns from influencing the estimates of the model's parameters.
Violation of any of these assumptions could lead to biased or inefficient parameter estimates.

3.2 Nature of the Data

The experimenter must ensure the following key requirements for a simple linear regression:

1. **Causality:** There should be a theoretical or philosophical basis for expecting a causal relationship, where the independent variable (X) influences or determines the dependent variable (Y).[1] It is assumed that changes in X cause changes in Y.

2. **Independence of Observations:**

    - The observations or measured values of Y must be independent of each other. For each value of X, there should be only one corresponding value of Y, or if there are replicate Y values, they must be statistically independent and not influence each other.
    - The observations of Y must also be independently across the range of X values. This means that the value of Y at one point should not influence the value of Y at another point.[2]

3. **Independent Variable Scale:** The independent variable (X) should be measured on a continuous scale, such as integers, real numbers, intervals, or ratios.

4. **Dependent Variable Scale:** Similarly, the dependent variable (Y) should also be measured on a continuous scale, such as integers, real numbers, intervals, or ratios.[3]

What if my data are not continuous?

- If the independent variable is ordinal, use *ordinal regression*.
- If the dependent variable is ordinal, use *ordinal (logistic) regression*.

What if I have more than one independent variable?

- Use *multiple linear regression*.

Additional assumptions and requirements are discussed next in Section 3.3.

[1] The independent and dependent variables are also called the predictor and response variables, respectively. The predictor is often under the experimenter's control (in which case it is a fixed effects model), while the response is the variable predicted to respond in the manner hypothesised.

[2] If Y not independent across the range of X, use a different type of regression model, such as a linear mixed-effects model.

[3] The dependent variable can also be ordinal, but this is less common. If this is the case, use *ordinal (logistic) regression instead.

3.3 Assumptions

The following assumptions are made when performing a simple linear regression; 1-3 must be tested *after* fitting the linear model:

1. **Normality:** For each value of X, there is a corresponding normal distribution of Y values. Each value of Y is randomly sampled from this normal distribution.
2. **Homoscedasticity:** The variances of the Y distributions corresponding to each X value should be approximately equal.
3. **Linearity:** There exists a linear relationship between the variables Y and X.
4. **Measurement Error:** It is assumed that the measurements of X are obtained without error. However, in practical scenarios, this is rarely the case. Therefore, we assume any measurement error in X to be negligible.

See Section 3.8 for more information about how to proceed when assumptions 1-3 are violated.

3.4 Outliers and Their Impact on Simple Linear Regression

In simple linear regression, outliers can have significant detrimental effects on the analysis and the reliability of the results. Outliers are data points that deviate substantially from the overall pattern or trend observed in the data, and their presence can lead to biased parameter estimates, inflated standard errors, distorted confidence and prediction intervals, violation of assumptions, and masking of underlying patterns.

Specifically, they can greatly impact the estimation of the slope and intercept due to their influence on the process of minimising the sum of squared residuals. Their presence can increase the standard errors of the regression coefficients, making it harder to detect significant relationships between the independent and dependent variables. Furthermore, the inclusion of outliers in the dataset can distort the calculation of confidence and prediction intervals for individual observations, preventing accurate inference and prediction. Their presence may also lead to violations of the assumptions of linear regression, such as the normality of residuals and the constant variance of errors (homoscedasticity). Lastly, extreme outliers can mask underlying patterns or relationships in the data and hinder our ability to discern the true nature of the associations between variables.

3.5 R Function

The `lm()` function in R is used to fit linear models. It can be used to carry out simple linear regression, multiple linear regression, and more.

The general form of the function written in R is:

```
lm(formula, data, ...)
```

where `formula` is a symbolic description of the model to be fitted, and `data` is the data frame containing the variables. The `...` argument is used to pass additional arguments to the function (consult `?lm`). For example:

```
lm(y ~ x, data = df)                                                    ①
```

① You can read the statement `y ~ x` as "y is modelled as a function of x."

The above statement fits a simple linear regression model with `y` as the dependent variable and `x` as the independent variable. The data frame `df` contains the variables named `x` and `y`.

3.6 Example: The Penguin Dataset

The following example workflow uses the `penguin` dataset from the `palmerpenguins` package to demonstrate how to perform a simple linear regression in R. The data are in Table 3.1.

Although we can also do a correlation here, we will use a simple linear regression because we want to develop a predictive model that can be used to estimate the bill length of Adelie penguins based on their body mass—this is a permissible application of a simple linear regression even though the two variables are not assumed to be causally related.

Table 3.1: Size measurements for adult foraging Adelie penguins near Palmer Station, Antarctica.

Bill length (mm)	Body mass (g)
39.1	3750
39.5	3800
40.3	3250
36.7	3450
39.3	3650
38.9	3625

3.6.1 Do an Exploratory Data Analysis (EDA)

```
dim(Adelie)
```

```
[1] 151   8
```

```
summary(Adelie)
```

```
     species          island   bill_length_mm  bill_depth_mm
 Adelie   :151   Biscoe   :44   Min.   :32.10   Min.   :15.50
 Chinstrap:  0   Dream    :56   1st Qu.:36.75   1st Qu.:17.50
 Gentoo   :  0   Torgersen:51   Median :38.80   Median :18.40
                                Mean   :38.79   Mean   :18.35
                                3rd Qu.:40.75   3rd Qu.:19.00
                                Max.   :46.00   Max.   :21.50
 flipper_length_mm  body_mass_g       sex          year
 Min.   :172       Min.   :2850   female:73   Min.   :2007
 1st Qu.:186       1st Qu.:3350   male  :73   1st Qu.:2007
```

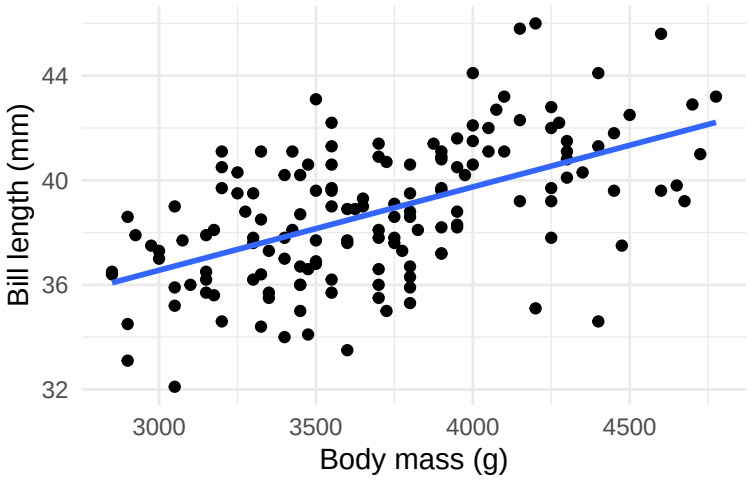

Figure 3.1: Scatter plot of the Palmer Station Adelie penguin data with a best fit line.

```
Median :190       Median :3700   NA's   : 5   Median :2008
Mean   :190       Mean   :3701                Mean   :2008
3rd Qu.:195       3rd Qu.:4000                3rd Qu.:2009
Max.   :210       Max.   :4775                Max.   :2009
```

We see that the dataset contains 344 observations of 8 variables. We shall focus on the `body_mass_g` and `bill_length_mm` variables for this example. Importantly, the two variables are continuous, which seems to satisfy the requirements for a simple linear regression. We will also restrict this analysis to the `Adelie` penguins ($n = 152$). Is the relationship between the body mass and bill length of the penguins linear? Let's find out.

3.6.2 Create a Plot

Construct a scatter plot of the data and include a best fit straight line:

```
ggplot(Adelie,
       aes(x = body_mass_g, y = bill_length_mm)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Body mass (g)", y = "Bill length (mm)") +
  theme_minimal()
```

Although there is some scatter in the data (Figure 3.1), there appears to be a positive relationship between the body mass and bill length of the penguins. This relationship might be amenable for modelling with a linear relationship and we shall continue to explore this.

3.6.3 State the Hypothesis

- Null Hypothesis (H_0): there is no relationship between the body mass of the penguins and their bill length.

- Alternative Hypothesis (H_A): there is a relationship between the two variables.

This can be written as:

$$H_0 : \beta = 0 \tag{3.2}$$

As seen above, this hypothesis concerns the slope of the regression line, β. If the slope is zero, then there is no relationship between the two variables. Regression models also tests an hypothesis about the intercept, α, but this is less commonly reported.

3.6.4 Fit the Model

Since the assumptions of a linear regression can only be tested *after* fitting the model, we first fit the model and then test the assumptions.

```
mod1 <- lm(bill_length_mm ~ body_mass_g,
           data = Adelie)
summary(mod1)
```

```
Call:
lm(formula = bill_length_mm ~ body_mass_g, data = Adelie)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4208 -1.3690  0.1874  1.4825  5.6168

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.699e+01  1.483e+00  18.201  < 2e-16 ***
body_mass_g 3.188e-03  3.977e-04   8.015 2.95e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.234 on 149 degrees of freedom
Multiple R-squared:  0.3013,    Adjusted R-squared:  0.2966
F-statistic: 64.24 on 1 and 149 DF,  p-value: 2.955e-13
```

3.6.5 Test the Assumptions

Assumptions of normality, homoscedasticity, and linearity must be tested (Section 7.3).

We already noted that a linear model will probably be appropriate for the data (see Figure 3.1), so we proceed with the other assumptions.

To facilitate the production of the diagnostic plots, we will use the **broom** package's `augment()` function to add the residuals to the data within the original dataset (now appearing as the tidied dataset, `mod1_data`). This will allow us to create the diagnostic plots more easily, and later we can also use it to look for the presence of outliers (Section 3.6.6).

```
library(broom)

mod1_data <- augment(mod1)
```

Normality

I first check the normality assumption using one of several options (Options 1-3). Here I use the Shapiro-Wilk test, a Residual Q-Q plot, and a histogram of the residuals.

Option 1: Perform the Shapiro-Wilk test on the residuals. The Shapiro-Wilk test is useful for detecting departures from normality in small sample sizes. The hypothesis is:

- H_0: the residuals are normally distributed.
- H_A: the residuals are not normally distributed.

```
shapiro.test(residuals(mod1))
```

```
    Shapiro-Wilk normality test

data:  residuals(mod1)
W = 0.99613, p-value = 0.9637
```

The *p*-value is greater than 0.05, so I reject the alternative hypothesis. I conclude that the residuals are normally distributed.

Option 2: Create a Residual Q-Q plot to visually assess the normality of the residuals:

The residuals are plotted against a theoretical normal distribution. The residuals fall along the line without major deviations, therefore the residuals are normally distributed (Figure 3.2 A).

Option 3: Create a histogram of the residuals to visually assess the normality of the residuals:

The histogram of the residuals appears to be normally distributed (Figure 3.2 B).

Homoscedasticity

I now examine the homoscedasticity assumption. The residuals should be approximately equal across all values of the independent variable. There are several options.

Option 1: I will use the Breusch-Pagan test to test for homoscedasticity.

The Breusch-Pagan test is used to assess the presence of heteroscedasticity (non-constant variance) in the residuals of a regression model.

The hypothesis is:

- H_0: the residuals are homoscedastic.
- H_A: the residuals are heteroscedastic.

```
library(lmtest)
bptest(mod1)
```

```
    studentized Breusch-Pagan test
```

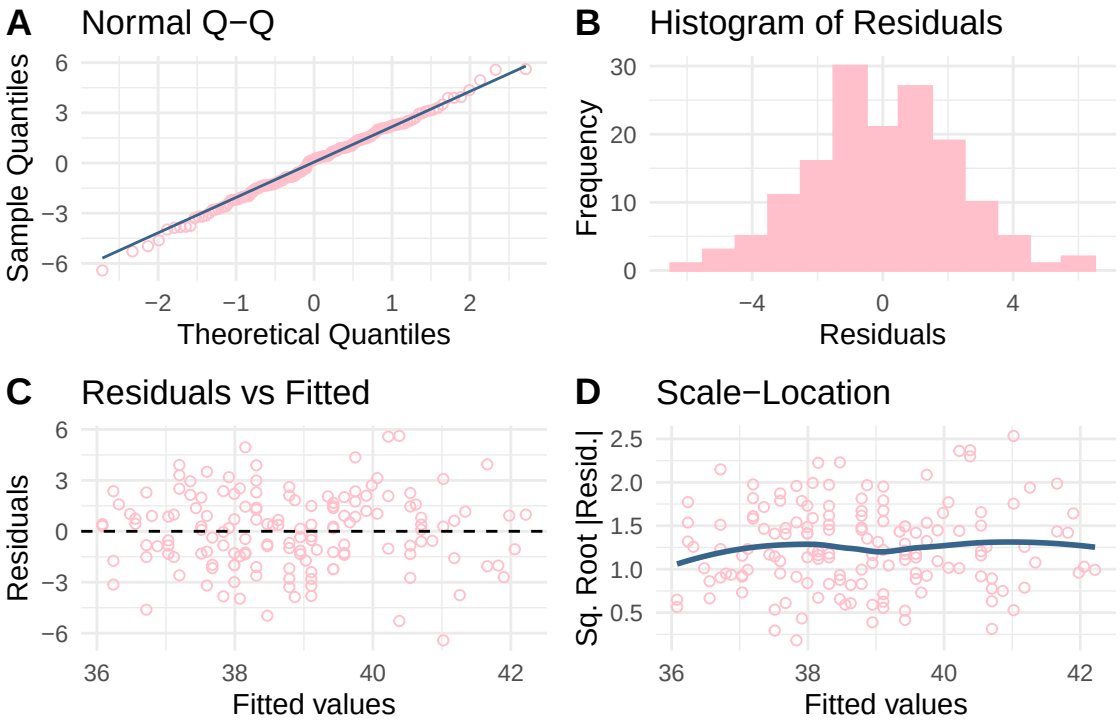

Figure 3.2: Diagnostics plots the linear regression, `mod1`, for assumption testing.

```
data:  mod1
BP = 1.6677, df = 1, p-value = 0.1966
```

The *p*-value is greater than 0.05, so I reject the alternative hypothesis. I conclude that the residuals are homoscedastic.

Option 2: Create a plot of the residuals against the fitted values to visually assess homoscedasticity:

The residuals are scattered evenly around zero from short through to long bill lengths, indicating that the residuals have constant variance (Figure 3.2 C).

Option 3: Create a plot of the standardised residuals against the independent variable to visually assess homoscedasticity:

The residuals are scattered evenly around zero from low through to high bill lenghts, indicating that the residuals have constant variance (Figure 3.2 D).

Other tests for homoscedasticity include the Goldfeld-Quandt (`lmtest::gqtest`) test, Levene's test (`car::leveneTest`), and others.

3.6.6 Check for outliers

How do we identify outliers in linear regression analysis? There are several approaches (see Figure 3.3):

1. **Difference in Fits (DFFITS):** DFFITS is a measure of the impact of each observation on

the predicted values (fitted values) of the model. It quantifies how much the predicted values would change if an observation were removed from the analysis. DFFITS values > Threshold $= 2\sqrt{\frac{p}{n}}$ indicate observations that have a substantial impact on the predicted values and may be influential or outliers. Here, p is the number of parameters in the model (including the intercept, i.e. 2 in a simple linear regression) and n is the number of observations.

2. **Cook's Distance Plot:** Cook's distance is a measure of the influence of each observation on the estimated regression coefficients. The Cook's distance plot shows the Cook's distance values for each observation against the row numbers (or observation numbers). Points with large Cook's distance values (typically greater than $\frac{4}{n}$) indicate observations that are potentially influential and may have a significant impact on the regression results.

3. **Residuals vs Leverage Plot:** This plot displays the standardised residuals against the leverage values (hat values) for each observation. Leverage values measure the influence of an observation on the fitted values (predicted values) of the model. The plot helps identify outliers and influential observations. Points with high leverage (typically greater than 2-3 times the average leverage) and large residuals are considered influential observations that may warrant further investigation or potential removal from the analysis.

4. **Cook's Distance vs Lev./(1-Lev.) Plot:** This plot combines information from Cook's distance and leverage values. The x-axis represents the leverage values divided by (1 minus the leverage values), which is a transformation that spreads out the points for better visualisation. The y-axis shows the Cook's distance values. This plot helps identify influential observations by considering both their impact on the regression coefficients (Cook's distance) and their influence on the fitted values (leverage). Points in the top-right corner of the plot indicate observations that are potentially influential and may require further examination or removal.

```
cooksd_thresh <- 4 / nrow(mod1_data)                                    ①
dffits_threshold <- 2 * sqrt(2 / nrow(Adelie))                          ②

mod1_data <- mod1_data %>%
  mutate(index = row_number(),
         leverage = hatvalues(mod1),
         dffits = dffits(mod1),
         colour = ifelse(.cooksd > cooksd_thresh, "black", "pink"))
```

① Calculate thresholds for Cook's distance.
② Calculate the threshold for DFFITS.

Once we have found them (Figure 3.4), what do we do with outliers? There are a few strategies:

1. **Remove them:** If the outliers are due to data entry errors or other issues, it may be appropriate to remove them from the analysis. However, this should be done with caution, as outliers may be functionally important in the dataset if they represent rare, extreme events.

2. **Robust regression methods:** When there is certainty that the outliers are part of the observed response and represent extreme but rare occurrences, robust regression techniques such as M-estimation or least trimmed squares, which are less sensitive to the presence of outliers, could be used.

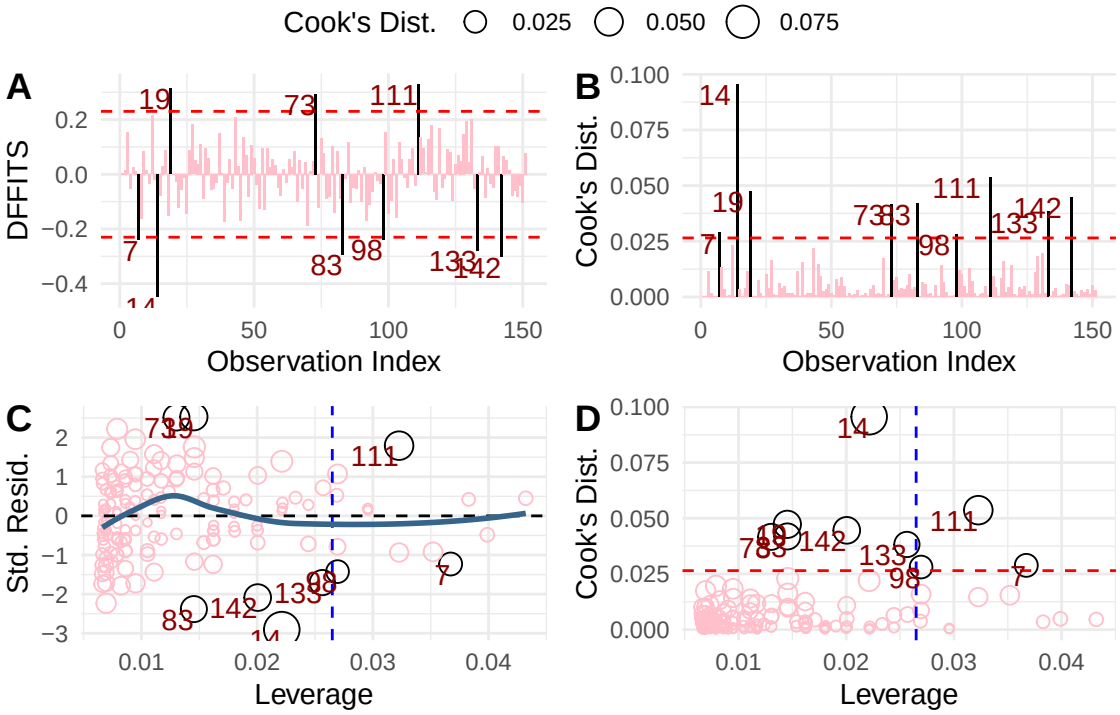

Figure 3.3: Diagnostic plots for visual inspection of outliers in the pernguin data. A) Difference in Fits (DFFITS) for `mod1`. B) Cook's distance. C) Residuals vs. leverage. D) Cook's distance vs. Lev./(1-Lev.). Outliers are identified beyond the Cook's distance threshold ($4/n$) and are plotted in black and their row numbers in dark red. The vertical dashed blue lines in C) and D) are positioned at 2 times the average leverage. The horizontal red dashed lines in B) and D) are located at the Cook's distance threshold. A) to C) are custom **ggplot2** plots corresponding to `plot(mod1, which = c(4, 5, 6))`.

3. **Transformation of variables:** Applying appropriate transformations (e.g., logarithmic, square root) to the variables can sometimes reduce the impact of outliers.

3.6.7 Interpret the Results

Now that we have tested the assumptions, we can interpret the results of the model fitted in Section 3.6.4. The slope of the regression line is 0.003188 mm/g, with a standard error of ± 0.0003977. The p-value is less than 0.001, so we reject the null hypothesis that the slope is zero. We conclude that there is a significant relationship between the body mass of the penguins and their bill length.

The fit of the model is given by the multiple R^2 value, which is 0.3013. This means that 30.13% of the variation in bill length can be explained by body mass. The remaining ~70% is due to other factors not included in the model. The intercept of the model is 26.99 mm, with a standard error of ± 0.0003977. The intercept is the value of the dependent variable when the independent variable is zero. In this case, it is the bill length of a penguin with a body mass of zero grams, which is not a meaningful value.

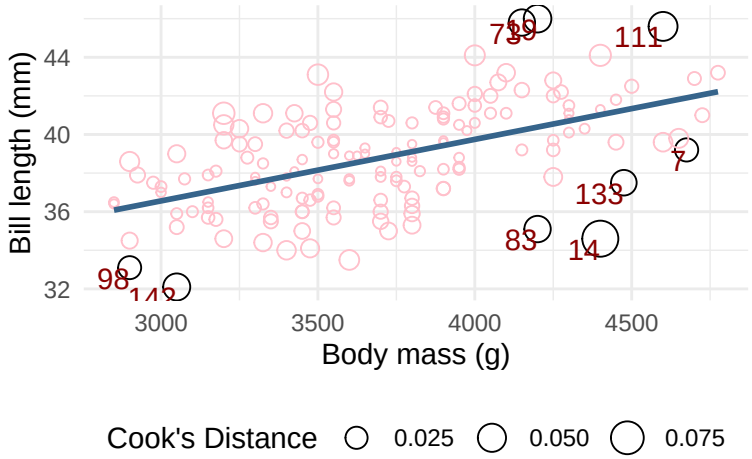

Figure 3.4: Plot of the linear regression resulting from `mod1` with the outliers identified using Cook's distance highlighted.

The significance of the overall fit of the model can be assessed using an analysis of variance (ANOVA) test. The *p*-value is less than 0.001, so we reject the null hypothesis that the model does not explain a significant amount of the variation in the data against an *F*-value of 64.25 on 1 and 149 degrees of freedom. We conclude that the model is a good fit for the data.

3.6.8 Reporting

I provide example Methods, Results, and Discussion sections in a format more-or-less suited for inclusion in a scientific manuscript. Feel free to use it as a template and edit it as necessary to describe your study.

Methods

Study data

The data analysed in this study were derived from the Palmer Penguins dataset, a comprehensive collection of measurements from three penguin species (Adelie, Chinstrap, and Gentoo) collected in the Palmer Archipelago, Antarctica. The dataset includes variables species, island, bill length, bill depth, flipper length, body mass, and sex of the penguins. This dataset has been made publicly available by Dr. Kristen Gorman and the Palmer Station, Antarctica LTER, a member of the Long Term Ecological Research Network.

Statistical analysis

The primary objective of our statistical analysis was to investigate the relationship between the penguins' body mass and bill length. For this purpose, we employed a simple linear regression model to quantify the extent to which the independent variable predicts bill length.

We fitted a simple linear regression model using the `lm()` function in R version 4.4.0 (R Core Team, 2024). The model included bill length as the dependent variable, and body mass as continuous predictor. We ensured all assumptions for linear regression were assessed including linearity, independence, homoscedasticity, and normality of residuals.

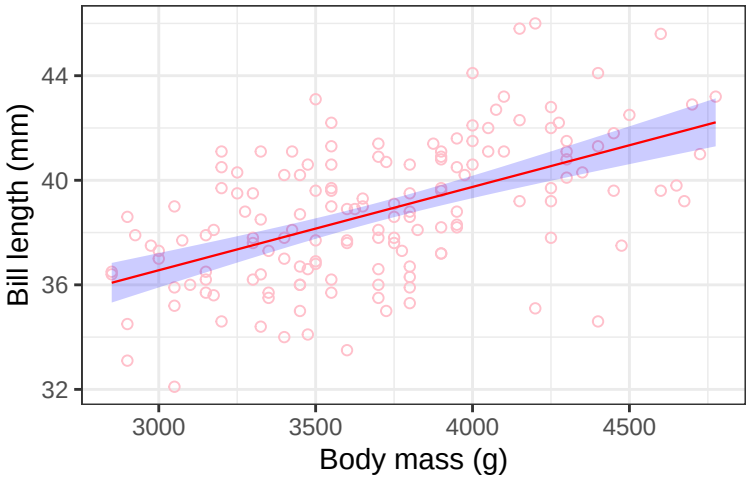

Figure 3.5: Plot of bill length as a function of body mass for Adelie penguins sampled at the Palmer Station. The straight line indicates the best fit regression line and the blue shading is the 95% confidence interval.

After fitting the model, diagnostic plots were generated using the `plot()` function in R to visually assess the residuals for any patterns indicating potential violations of regression assumptions. Additionally, the Shapiro-Wilk test was conducted to confirm the normality of the residuals. The presence of heteroscedasticity was evaluated using the Breusch-Pagan test.

The adequacy of the model fit was judged based on the coefficient of determination (R^2), which provided insight into the variance in body mass explained by the predictors. The significance of the regression coefficients was determined using t-tests, and the overall model fit was evaluated by an F-test.

Results

The regression coefficient for bill length with respect to body mass was estimated to be approximately 3.2×10^{-3} mm/g $\pm$ 3.977×10^{-4} (mean slope $\pm$ SE) ($p < 0.001$, $t = 8.015$), indicating a significant dependence of bill length on body mass (Figure 3.5).

The multiple R^2 value of the model was 0.3013, suggesting that approximately 30.13% of the variability in bill length can be accounted for by changes in body mass. This indicates that while bill length variation is notably influenced by body mass, about 69.87% of the variation is attributable to other factors not included in the model.

The overall fit of the model, assessed by an ANOVA, strongly supported the model's validity ($F = 64.25$, $p < 0.001$, d.f. = $1, 149$) and confirms that a linear model provides adequate support for predicting penguin bill length from body mass.

Discussion

In conclusion, the statistical analysis confirms a significant relationship between body mass and bill length in penguins. Although the model explains a substantial portion of the variation, future studies should consider additional variables that could account for the remaining variability in bill length. This would enhance our understanding of the morphological adaptations of penguins in their natural habitat.

3.7 Confidence and Prediction Intervals

Confidence intervals estimate the range within which the true mean of the dependent variable (*Y*) is likely to fall for a given value of the independent variable (*X*). In other words, if you were to repeat your experiment many times and calculate the mean response at a specific *X* value each time, the confidence interval would contain the true population mean a certain percentage of the time (e.g., 95%). Therefore, a 95% confidence interval means you can be 95% confident that the interval contains the true mean response for the population at that particular *X* value. It's about the average, not individual data points.

Prediction intervals, on the other hand, provide a range of *Y* values that are likely to contain a single new observation of the dependent variable for a given value of the independent variable *X*. These intervals account for the variability around individual observations and are generally wider than confidence intervals because they include both the variability of the estimated mean response and the variability of individual observations around that mean. Continuing with the Adelie penguin data, the confidence and prediction intervals are shown in Figure 3.6.

```
# Predict values with confidence intervals
pred_conf <- as.data.frame(predict(mod1,
                                   newdata = Adelie,
                                   interval = "confidence"))

# Predict values with prediction intervals
pred_pred <- as.data.frame(predict(mod1,
                                   newdata = Adelie,
                                   interval = "prediction"))

# Add body mass to the data frame
results <- cbind(Adelie, pred_conf, pred_pred[,2:3])

# Rename columns for clarity
names(results)[c(9:13)] <- c("fit", "lwr_conf", "upr_conf",
                             "lwr_pred", "upr_pred")

ggplot(data = results, aes(x = body_mass_g, y = fit)) +
  geom_line(linewidth = 0.4, colour = "red") +
  geom_ribbon(aes(ymin = lwr_pred, ymax = upr_pred),
              alpha = 0.2, fill = "red") +
  geom_ribbon(aes(ymin = lwr_conf, ymax = upr_conf),
              alpha = 0.2, fill = "blue") +
  geom_point(aes(y = bill_length_mm), shape = 1) +
  labs(x = "Body mass (g)", y = "Bill length (mm)") +
  theme_bw()
```

Confidence and prediction intervals are relevant for understanding the uncertainty associated with a linear regression model's predictions. While confidence intervals focus on quantifying the uncertainty around the estimated mean response, prediction intervals comprehensively assess the variability that can be expected for individual observations. We can use both when interpreting the results of a linear regression analysis.

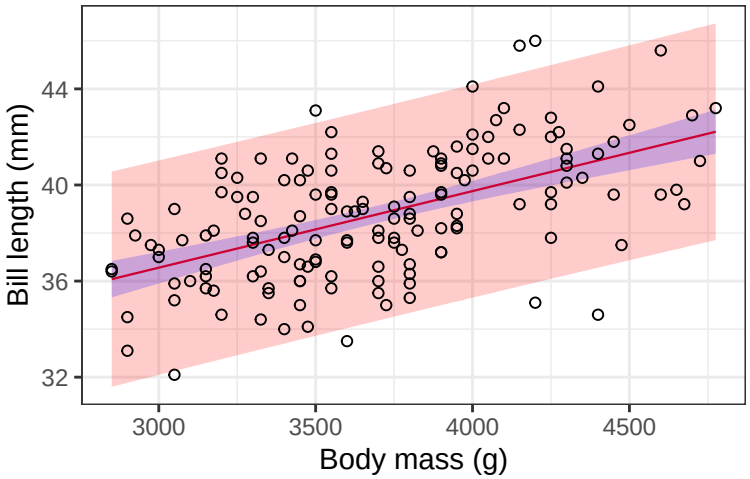

Figure 3.6: Plot of pernguin data with the confidence interval (blue) and prediction interval (pink) around the fitted values.

Confidence intervals are useful when the primary interest lies in making inferences about the mean response at specific values of the independent variable(s). For instance, in a study examining the relationship between soil nutrient levels and plant biomass, confidence intervals can help determine the range of mean biomass that can be expected for a given level of soil nutrients. This information may be valuable for crop management practices, such as designing fertilisation strategies or assessing the impact of nutrient depletion on plant productivity.

Prediction intervals, on the other hand, are more relevant when the goal is to predict the value of an individual observation or to assess the range of values that future observations might take. For example, in a study investigating the relationship between ambient temperature and the growth rate of a species of fish, prediction intervals provide a range of growth rates that an individual fish might exhibit based on the observed temperature. This information is invaluable in aquaculture, for instance, where predicting individual growth patterns can inform decisions about optimal stocking densities or feed management strategies.

The relative widths of confidence and prediction intervals can provide insights into the variability in the data. If the prediction intervals are substantially wider than the confidence intervals, it may indicate a high level of variability in individual observations around the mean response, which could suggest the presence of influential factors or sources of variation that are not accounted for by the current model, such as microhabitat differences or genetic variation within the studied population.

3.8 What Do I Do When Some Assumptions Fail?

3.8.1 Failing Assumptions of Normality and Homoscedasticity

I will use the sparrow data from Zar (1999) to demonstrate what to do when the assumptions of normality and homoscedasticity are violated. I will fit a linear model to the data and then check the assumptions.

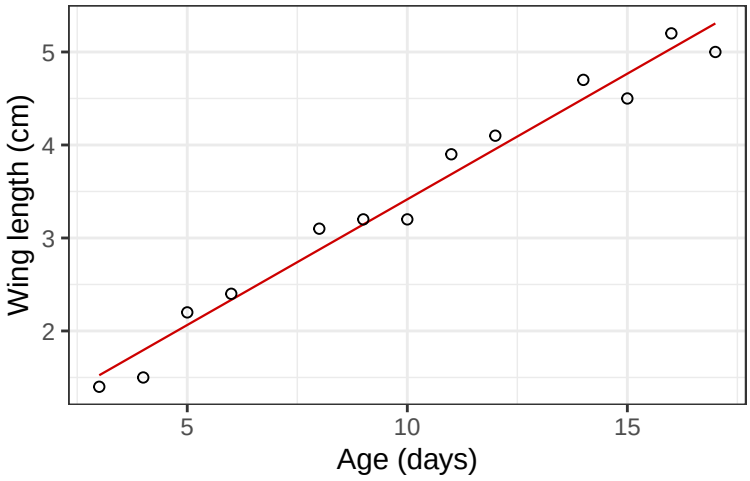

Figure 3.7: Scatter plot of the sparrow dataset with a best fit line.

Figure 3.7 is a scatter plot of the sparrow data with a best fit line. At first glance, the linear model seems to almost perfectly describe the relationship of wing length on age. I will fit a linear model to the data and then check the assumptions.

```
mod2 <- lm(wing ~ age, data = sparrows)
summary(mod2)
```

```
Call:
lm(formula = wing ~ age, data = sparrows)

Residuals:
     Min       1Q   Median       3Q      Max
-0.30699 -0.21538  0.06553  0.16324  0.22507

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.71309    0.14790   4.821 0.000535 ***
age          0.27023    0.01349  20.027 5.27e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2184 on 11 degrees of freedom
Multiple R-squared:  0.9733,    Adjusted R-squared:  0.9709
F-statistic: 401.1 on 1 and 11 DF,  p-value: 5.267e-10
```

Check the assumption of normality of residuals using the Shapiro-Wilk test, a histogram, and a residual Q-Q plot.

```
shapiro.test(residuals(mod2))
```

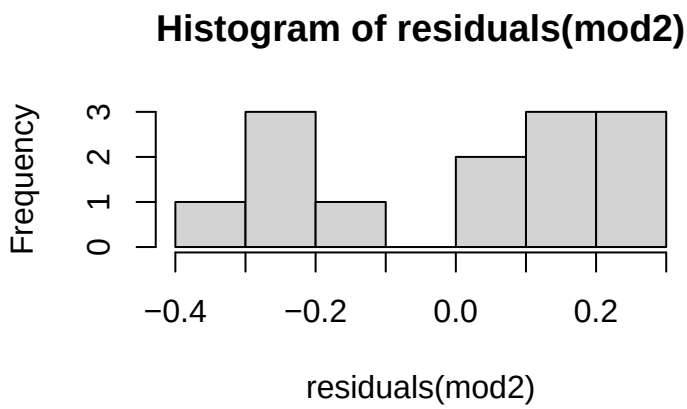

Figure 3.8: A histogram of the residual of the linear regression, `mod2`.

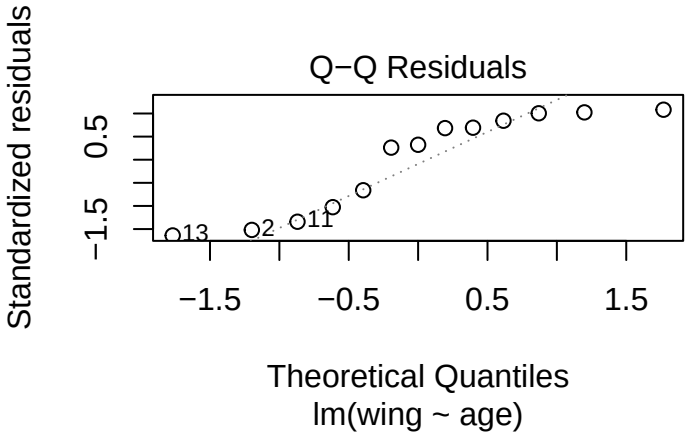

Figure 3.9: A Residual Q-Q plot of the linear regression, `mod2`.

```
    Shapiro-Wilk normality test

data:  residuals(mod2)
W = 0.84542, p-value = 0.02487
```

The *p*-value for the Shapiro-Wilk test is < 0.05, indicating that the residuals are not normally distributed. The histogram and Q-Q plot of the residuals also show that the residuals are not normally distributed (Figure 3.8 and Figure 3.9). In the Residual Q-Q plot, the points deviate from the straight line, indicating non-normality—note the S-shaped curvature to the data.

```
hist(residuals(mod2))
```

```
plot(mod2, which = 2)
```

It is enough to know that the normality assumption is not met – I cannot proceed with a simple linear regression. However, let us for completeness also look at the homoscedasticity assumption.

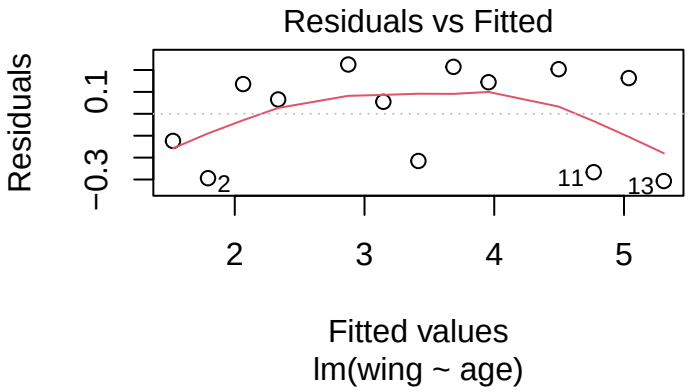

Figure 3.10: A plot of residuals against fitted values for the linear regression, `mod2`.

I will use the Breusch-Pagan test to check for homoscedasticity, followed by a plot of residuals against fitted values.

```
bptest(mod2)
```

```
    studentized Breusch-Pagan test

data:  mod2
BP = 1.6349, df = 1, p-value = 0.201
```

The *p*-value for the Breusch-Pagan test is > 0.05, indicating that the residuals are homoscedastic. The plot of residuals against fitted values shows gives a slightly different impression (Figure 3.10).

```
plot(mod2, which = 1)
```

The assumptions of normality and homoscedasticity are violated (it is sufficient that one or the other fails, not both). As already noted, I cannot proceed with the linear model. I will need to consider alternative models or transformations to address these issues.

When the assumptions of normality and homoscedasticity are violated, I have some options—these broadly group into transforming the data and using a non-parametric test.

Transforming the data can sometimes help attain normality and homoscedasticity. Common transformations include the logarithmic, square root, and inverse transformations. However, be cautious when interpreting the results of transformed data, as the transformed coefficients may not be directly interpretable.

I will show the Theil-Sen estimator (also known as Sen's slope estimator) as a robust non-parametric replacement for a simple linear model. It calculates the median of the slopes of all pairs of sample points to determine the overall slope of the line.

```
library(mblm)

mod3 <- mblm(wing ~ age, data = sparrows)
```

```
summary(mod3)
```

```
Call:
mblm(formula = wing ~ age, dataframe = sparrows)

Residuals:
     Min       1Q   Median       3Q      Max
-0.44524 -0.31190 -0.00714  0.06905  0.14048

Coefficients:
            Estimate     MAD V value Pr(>|V|)
(Intercept)  0.75000 0.18532      91 0.000244 ***
age          0.27619 0.00956      91 0.000244 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.244 on 11 degrees of freedom
```

The interpretation of the Theil-Sen estimator is similar to the simple linear regression. The Theil-Sen estimator provides a robust estimate of the slope of the relationship between age and wing length. The slope of the line is 0.28 ($\pm$ 0.19 mean absolute deviation) (V value = 91, $p < 0.001$), indicating that for each additional day of age, the wing length increases by 0.28 cm. The intercept of the line is 0.75, indicating that the wing length is ~0.8 cm when the age is 0 days.

3.8.2 My Data Do Not Display a Linear Response

In simple linear regression, the dependent variable *Y* is expected to exhibit a straight-line relationship with the independent variable *X*. However, several factors can cause deviations from a linear pattern.

Statistical assumptions underlying linear regression can affect the appearance of a linear response. The normality assumption is important but primarily pertains to the residuals rather than the *Y* vs. *X* plot. A scatterplot of *Y* vs. *X* might deviate from a linear pattern due to the non-normality of the residuals or heteroscedasticity, where the variability of the residuals changes with the level of *X*. Addressing these issues and then reassessing the linearity of the relationship is a logical first step. Refer to Section 3.8 for more details on how to proceed.

Outliers in the data can significantly impact the regression line, leading to misleading results (Section 3.6.6). Measurement errors in the independent variable can also lead to biased and inconsistent estimations, which may require revisiting the data collection process to address systemic problems. Variable bias, where excluding relevant variables distorts the observed relationship, could also explain seemingly nonlinear responses. Considering multiple predictor variables in a regression model (Chapter 5) might be more appropriate in such situations.

It's important to note that simple linear regression might not be suitable for all scenarios. For instance, the dependent variable *Y* might inherently follow a different probability distribution, such as a Poisson or a binomial distribution, rather than a normal distribution. This is particularly relevant in count data or binary outcome scenarios. In such cases, other types of models like Poisson regression or logistic regression, accommodated by generalised linear models (GLM; Chapter 6), would be more appropriate.

Lastly, if the data do not exhibit a linear relationship even after addressing these issues, the relationship between the variables may really be nonlinear. This can occur when the underlying functional relationship between X and Y is better described by exponential, logarithmic, or other more complex mechanistic responses. In such cases, nonlinear regression (Chapter 7) or generalised additive models (GAM; Chapter 10) might be necessary to describe the relationship between the variables accurately.

Chapter 4

Polynomial Regression

Polynomial regressions may resemble non-linear regression in terms of the visual appearance of the regression line (i.e. with bends and curves), but they handle non-linearity by transforming the independent variable X into higher powers (e.g., X^2, X^3), which are then included in the model along with coefficients that are linear in terms of estimation. For instance, a cubic (of *order*, *degree*, or *power* 3; denoted as m) polynomial regression model (Figure 7.1 A) and is expressed as:

$$Y_i = \alpha + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \epsilon_i \tag{4.1}$$

Where:

- Y_i is the response variable for the i-th observation,
- X_i is the predictor variable for the i-th observation,
- α is the intercept,
- β_1, β_2, and β_3 are the coefficients for the linear, quadratic, and cubic terms, respectively, and
- ϵ_i is the error term for the i-th observation (the residuals).

It is worth noting that higher order polynomials can lead to overfitting, where the model captures the noise in the data rather than the inherent pattern. This can result in poor generalisation to new data and poor predictive performance. Overfitting becomes more likely as m increases. The m of a polynomial should not exceed $n - 1$, where n is the number of data points. An m greater than 4 or 5 is rarely justified.[1] If $m = n - 1$, the polynomial will fit the data perfectly (i.e., $R^2 = 1$). For example, a linear regression ($m = 1$) fits two data points exactly, a quadratic regression ($m = 2$) fits three data points perfectly, and so on. Therefore, always consider the trade-off between model complexity and generalisation when using polynomial regression.

Another complication is that the biological interpretation of more complex (higher order) models may be lacking. However, polynomial regression are more often than not used for prediction rather than their interpretability.

<THE REST OF THIS CHAPTER IS TO BE DEVELOPED.>

[1] The appropriate maximum m can be determined using methods such as the backward-elimination or forward-selection multiple-regression procedure.

Chapter 5

Multiple Linear Regression

In Section 3.1 we have seen how to model the relationship between two variables using simple linear regression (SLR). However, in ecosystems, the relationship between the response variable and the explanatory variables is more complex and in many cases cannot be adequately captured by a single driver (i.e. influential or predictor variable). In such cases, multiple linear regression (MLR) can be used to model the relationship between the response variable and multiple explanatory variables.

5.1 Multiple Linear Regression

Multiple linear regression helps us answer questions such as:

- How do various environmental factors influence the population size of a species? Factors like average temperature, precipitation levels, and habitat area can be used to predict the population size of a species in a given region. Which of these factors are most important in determining the population size?
- What are the determinants of plant growth in different ecosystems? Variables such as soil nutrient content, water availability, and light exposure can help predict the growth rate of plants in various ecosystems. How do these factors interact to influence plant growth?
- How do genetic and environmental factors affect the spread of a disease in a population? The incidence of a disease might depend on factors like genetic susceptibility, exposure to pathogens, and environmental conditions (e.g., humidity and temperature). What is the relative importance of these factors in determining the spread of the disease?

Multiple linear regression extends the simple linear regression model to include several independent variables. The model is expressed as:

$$Y_i = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik} + \epsilon_i \tag{5.1}$$

Where:

- Y_i is the response variable for the i-th observation,
- $X_{i1}, X_{i2}, \ldots, X_{ik}$ are the k predictor variables for the i-th observation,
- α is the intercept,

- $\beta_1, \beta_2, \ldots, \beta_k$ are the coefficients for the k predictor variables, and
- ϵ_i is the error term for the i-th observation (the residuals).

When including a categorical variable in a multiple linear regression model, dummy (indicator) variables are used to represent the different levels of the categorical variable. Let's assume we have a categorical variable C with three levels: C_1, C_2, and C_3. We can represent this categorical variable using two dummy variables:

- D_1: Equals 1 if $C = C_2$, 0 otherwise.
- D_2: Equals 1 if $C = C_3$, 0 otherwise.

C_1 is considered the reference category and does not get a dummy variable. This way, we avoid multicollinearity (see Section 5.6.4). R's `lm()` function will automatically convert the categorical variables to dummy variables (sometimes called treatment coding). The first level of the alphabetically sorted categorical variable is taken as the reference level. See Section 8.5 for more information about how to include categorical variables in a multiple linear regression model. At the end of the chapter you'll find alternative ways to assess categorical variables in a multiple linear regression model (Section 5.9).

Assume we also have k continuous predictors $X_1, X_2, \ldots, X_k$. The multiple linear regression model with these predictors and the categorical variable can be expressed as:

$$Y_i = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik} + \gamma_1 D_{i1} + \gamma_2 D_{i2} + \epsilon_i \tag{5.2}$$

Where:

- Y_i is the dependent variable for observation i.
- α is the intercept term.
- $\beta_1, \beta_2, \ldots, \beta_k$ are the coefficients for the continuous independent variables $X_{i1}, X_{i2}, \ldots, X_{ik}$.
- D_{i1} and D_{i2} are the dummy variables for the categorical predictor C.
- γ_1 and γ_2 are the coefficients for the dummy variables, representing the effect of levels C_2 and C_3 relative to the reference level C_1.
- ϵ_i is the error term for observation i.

5.2 Nature of the Data

You are referred to the discussion in simple linear regression (Section 3.1). The only added consideration is that the data should be multivariate, i.e., it should contain more than one predictor variable. The predictor variables are generally continuous, but there may also be categorical variables.

5.3 Assumptions

Basically, this is as already discussed in simple linear regression (Section 3.1)—in multiple linear regression, the same assumptions apply to the response relative to each of the predictor variables. In Section 5.6.7 I will assess the assumptions in an example dataset. An additional consideration is that the predictors must not be highly correlated with each other (multicollinearity) (see Section 5.6.4).

5.4 Outliers

Again, this is as discussed in simple linear regression (Section 3.1). In multiple linear regression, the same considerations apply to the response relative to each of the predictor variables.

5.5 R Function

The `lm()` function in R is used to fit a multiple linear regression model. The syntax is similar to that of the `lm()` function used for simple linear regression, but with multiple predictor variables. The function takes the basic form:

```
lm(formula, data)
```

For a multiple linear regression with only continuous predictor variables (as in Equation 5.1), the formula is:

```
lm(response ~ predictor1 + predictor2 + ... + predictorN,
   data = dataset)
```

Interaction effects are implemented by including the product of two variables in the formula. For example, to include an interaction between `predictor1` and `predictor2`, we can use:

```
lm(response ~ predictor1 * predictor2, data = dataset)
```

When we have both continuous and categorical predictor variables (Equation 5.2), the formula is:

```
lm(response ~ continuous_predictor1 + continuous_predictor2 + ...
   + continuous_predictorN + factor(categorical_predictor1) +
     factor(categorical_predictor2) + ...
   + factor(categorical_predictorM),
   data = dataset)
```

5.6 Example 1: The Seaweed Dataset

Load some data produced in the analysis by Smit et al. (2017). Please refer to the chapter Deep Dive into Gradients on Tangled Bank for the data description.

This dataset is suitable for a multiple linear regression because it has continuous response variables ($\beta_{sør}$, β_{sim}, and β_{sne}, the Sørenesen dissimilarity, the turnover component of β-diversity, and the nestedness-resultant component of β-diversity, respectively), continuous predictor variables (the mean climatological temperature for August, the mean climatological temperature for the year, the temperature range for February and August, and the SD of February and August), and a categorical variable (the bioregional classification of the samples).

```
sw <- read.csv("data/spp_df2.csv")
rbind(head(sw, 3), tail(sw, 3))[,-1]
```

```
      dist  bio    augMean   febRange      febSD     augSD    annMean
1     0.000  BMP 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000
2    51.138  BMP 0.05741369 0.09884404 0.16295271 0.3132800 0.01501846
3   104.443  BMP 0.15043904 0.34887754 0.09934163 0.4188239 0.02602247
968 102.649 ECTZ 0.41496099 0.11330069 0.24304493 0.7538546 0.52278161
969  49.912 ECTZ 0.17194242 0.05756093 0.18196664 0.3604341 0.24445006
970   0.000 ECTZ 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000
              Y        Y1          Y2
1   0.000000000 0.0000000 0.000000000
2   0.003610108 0.0000000 0.003610108
3   0.003610108 0.0000000 0.003610108
968 0.198728140 0.1948882 0.003839961
969 0.069337442 0.0443038 0.025033645
970 0.000000000 0.0000000 0.000000000
```

We will do a multiple linear regression analysis to understand the relationship between some of the environmental variables and the seaweed species. Specifically, we will consider only the variables `augMean`, `febRange`, `febSD`, `augSD`, and `annMean` as predictors of the species composition as measured by $\beta_{sør}$ (`Y` in the data file).

The model, which we will call `full_mod1` below, can be stated formally as Equation 5.3:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \epsilon \tag{5.3}$$

Where:

- Y is the response variable, the mean Sørensen dissimilarity,
- the predictors X_1, X_2, X_3, X_4, and X_5 correspond to `augMean`, `febRange`, `febSD`, `augSD`, and `annMean`, respectively, and
- ϵ is the error term.

But before we jump into multiple linear regression, let's warm up by first fitting some simple linear regressions.

5.6.1 Simple Linear Models

For interest sake, let's fit simple linear models for each of the predictors against the response variable. Let's look at relationships between the continuous predictors and the response in the East Coast Transition Zone (`ECTZ`), ignoring the other bioregions for now. We will first fit the simple linear models and then create scatter plots of the response variable $\beta_{sør}$ against each of the predictor variables. To these plots, we will add a best fit (regression) lines.

```
sw_ectz <- sw ▷ filter(bio == "ECTZ")

predictors <- c("augMean", "febRange", "febSD", "augSD", "annMean")

# Fit models using purrr::map and store in a list
```

```
  models <- map(predictors, ~ lm(as.formula(paste("Y ~", .x)),
                                 data = sw_ectz))

  names(models) <- predictors

  model_summaries <- map(models, summary)
  model_summaries
```

```
$augMean

Call:
lm(formula = as.formula(paste("Y ~", .x)), data = sw_ectz)

Residuals:
      Min        1Q    Median        3Q       Max
-0.180961 -0.059317 -0.008346  0.045695  0.192444

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.060104   0.007359   8.168 1.01e-14 ***
augMean     0.346011   0.010899  31.748  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07721 on 287 degrees of freedom
Multiple R-squared:  0.7784,    Adjusted R-squared:  0.7776
F-statistic:  1008 on 1 and 287 DF,  p-value: < 2.2e-16


$febRange

Call:
lm(formula = as.formula(paste("Y ~", .x)), data = sw_ectz)

Residuals:
     Min       1Q   Median       3Q      Max
-0.21744 -0.08311 -0.01543  0.07536  0.25699

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.092722   0.009638   9.621   <2e-16 ***
febRange    0.181546   0.008897  20.405   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1048 on 287 degrees of freedom
Multiple R-squared:  0.592, Adjusted R-squared:  0.5905
F-statistic: 416.4 on 1 and 287 DF,  p-value: < 2.2e-16
```

```
$febSD

Call:
lm(formula = as.formula(paste("Y ~", .x)), data = sw_ectz)

Residuals:
     Min       1Q   Median       3Q      Max
-0.24267 -0.10709 -0.02587  0.08888  0.39171

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.12018    0.01168   10.29   <2e-16 ***
febSD        0.17166    0.01245   13.79   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1272 on 287 degrees of freedom
Multiple R-squared:  0.3985,    Adjusted R-squared:  0.3964
F-statistic: 190.1 on 1 and 287 DF,  p-value: < 2.2e-16


$augSD

Call:
lm(formula = as.formula(paste("Y ~", .x)), data = sw_ectz)

Residuals:
      Min        1Q    Median        3Q       Max
-0.307683 -0.111051 -0.003922  0.086322  0.308041

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.12781    0.01231   10.38   <2e-16 ***
augSD        0.08793    0.00720   12.21   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.133 on 287 degrees of freedom
Multiple R-squared:  0.3419,    Adjusted R-squared:  0.3396
F-statistic: 149.1 on 1 and 287 DF,  p-value: < 2.2e-16


$annMean

Call:
lm(formula = as.formula(paste("Y ~", .x)), data = sw_ectz)
```

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.144251 -0.051607 -0.005023  0.045095  0.145173

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.053883   0.006309   8.541 7.94e-16 ***
annMean     0.332150   0.008667  38.325  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0663 on 287 degrees of freedom
Multiple R-squared:  0.8365,    Adjusted R-squared:  0.836
F-statistic:  1469 on 1 and 287 DF,  p-value: < 2.2e-16
```

The individual models show that, for each predictor, the estimate of the coefficients (for slope) and the test for the overall hypothesis are both significant ($p < 0.05$ in all cases; refer to the model output). All the predictor variables are therefore good predictors of the structure of seaweed species composition along.

```r
# Create individual plots for each predictor
plts1 <- map(predictors, function(predictor) {
  ggplot(sw_ectz, aes_string(x = predictor, y = "Y")) +
    geom_point(shape = 1, colour = "dodgerblue4") +
    geom_smooth(method = "lm", col = "magenta", fill = "pink") +
    labs(title = paste("Y vs", predictor),
         x = predictor,
         y = "Y") +
    theme_bw()
})

# Name the list elements for easy reference
names(plts1) <- predictors

ggpubr::ggarrange(plotlist = plts1, ncol = 2,
                  nrow = 3, labels = "AUTO")
```

Figure 5.1 is a series of scatter plots showing the relationship between the response variable $\beta_{sør}$ and each of the predictor variables. The blue line represents the linear regression fitted to the data. We see that the relationship between the response variable and each of the predictors is positive and linear. Each of the models are significant, as indicated by the p-values in the model summaries. These simple models do not tell us how some predictors might act together to influence the response variable.

To consider combined effects and interactions between predictor variables, we must explore multiple linear regression models that include all the predictors. Multiple regression will give us a more integrated understanding of how various environmental variables jointly influence species composition along the coast. In doing so, we can control for confounding variables, improve model fit, deal with multicollinearity, test for interaction effects, and enhance predictive power.

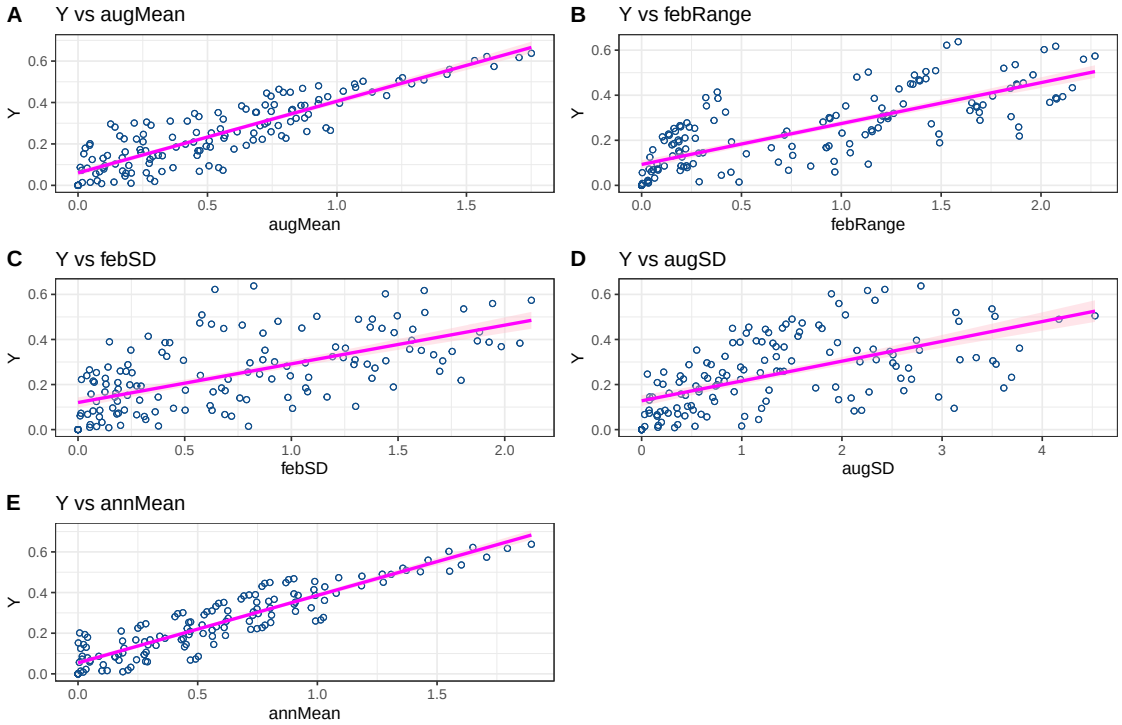

Figure 5.1: Individual simple linear regressions fitted to the variables `augMean`, `febRange`, `febSD`, `augSD`, and `annMean` as predictors of the seaweed species composition as measured by the Sørensen dissimilarity, `Y`.

We will fit this multiple regression model next.

5.6.2 State the Hypotheses for a Multiple Linear Regression

As with all inferential statistics, we need to consider the hypotheses when performing multiple linear regression.

The null hypothesis (H_0) states that there is no significant relationship between the Sørensen diversity index and any of the the climatological variables entered into the model, implying that the coefficients for all predictors are equal to zero. The alternative hypothesis (H_A), on the other hand, states that there is a significant relationship between the Sørensen diversity index and the climatological variables, positing that at least one of the coefficients is not equal to zero.

The hypotheses can be divided into two kinds: those dealing with the main effects and the one assessing the overall model stated in Equation 5.3.

Main effects hypotheses

The main effects hypotheses test, for each predictor, X_i, if the predictor has a significant effect on the response variable Y.

H_0: There is no linear relationship between the environmental variables (`augMean`, `febRange`, `febSD`, `augSD`, and `annMean`) and the community composition as measured by $\beta_{sør}$ (in `Y`). Formally,

for each predictor variable X_i:

- $H_0 : \beta_i = 0$ for $i = 1, 2, 3, 4, 5$

Where β_i are the coefficients of the predictors in the multiple linear regression model.

H_A: There is a linear relationship between the environmental variables (`augMean`, `febRange`, `febSD`, `augSD`, and `annMean`) and the species composition as measured by $\beta_{sør}$:

- $H_A : \beta_i \neq 0$ for $i = 1, 2, 3, 4, 5$

Overall hypothesis

In addition to testing the individual predictors, X_i, we can also test a hypothesis about the overall significance of the model (F-test), which examines whether the model as a whole explains a significant amount of variance in the response variable Y. A significant F-test would suggest that *at least one* predictor (excluding the intercept) in the model is likely to be significantly related to the response, but it requires further investigation of individual predictors and potential multicollinearity to fully understand the relationships. For the overall model hypothesis:

Null Hypothesis (H_0):

- $H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$

Alternative Hypothesis (H_A):

- $H_A : \exists \beta_i \neq 0$ for at least one i

5.6.3 Fit the Model

We fit two models:

- a full model that includes an intercept term and the five environmental variables, and
- a null model that includes only an intercept term.

The reason the null model is included is to compare the full model with a model that has no predictors. This comparison will help us determine which of the predictors are useful in explaining the response variable—we will see this in action in the forward model selection process later on (Section 5.6.5).

```
# Select only the variables that will be used in model building
sw_sub1 <- sw_ectz[, c("Y", "augMean", "febRange",
                       "febSD", "augSD", "annMean")]

# Fit the full and null models
full_mod1 <- lm(Y ~ augMean + febRange + febSD +
                  augSD + annMean, data = sw_sub1)
null_mod1 <- lm(Y ~ 1, data = sw_sub1)

# Add fitted values from the full model to the dataframe
sw_ectz$.fitted <- fitted(full_mod1)
```

5.6.4 Dealing With Multicollinearity

Some of the predictor variables may be correlated with each other and this can lead to multicollinearity. When predictor variables are highly correlated, the model may not be able to distinguish the individual effects of each predictor. Consequently, the model becomes less precise and harder to interpret due to the coefficients' inflated standard errors (Graham (2003)). One can create a plot of pairwise correlations to visually inspect the correlation structure of the predictors. I'll not do this here, but you can try it on your own.

A formal way to detect multicollinearity is to calculate the variance inflation factor (VIF) for each predictor variable. The VIF measures how much the variance of the estimated regression coefficients is increased due to multicollinearity. A VIF value greater than 5 or 10 indicates a problematic amount of multicollinearity.

```
initial_formula <- as.formula("Y ~ .")

threshold <- 10 # Define a threshold for VIF values

# Extract the names of the predictor variables
predictors <- names(vif(full_mod1))

# Iteratively remove collinear variables
while (TRUE) {
  # Calculate VIF values
  vif_values <- vif(full_mod1)
  print(vif_values) # Print VIF values for debugging
  max_vif <- max(vif_values)

  # Check if the maximum VIF is above the threshold
  if (max_vif > threshold) {
    # Find the variable with the highest VIF
    high_vif_var <- names(which.max(vif_values))
    cat("Removing variable:",
        high_vif_var,
        "with VIF:",
        max_vif,
        "\n")

    # Update the formula to exclude the high VIF variable
    updated_formula <- as.formula(paste("Y ~ . -", high_vif_var))

    # Refit the model without the high VIF variable
    full_mod1 <- lm(updated_formula, data = sw_sub1)

    # Update the environment data frame to reflect the removal
    sw_sub1 <- sw_sub1[, !(names(sw_sub1) %in% high_vif_var)]
  } else {
    break
```

```
    }
  }
```

```
  augMean  febRange     febSD     augSD   annMean
27.947767 10.806635  8.765732  2.497739 31.061900
Removing variable: annMean with VIF: 31.0619
  augMean  febRange     febSD     augSD
 2.290171 10.648752  8.637679  1.616390
Removing variable: febRange with VIF: 10.64875
 augMean    febSD    augSD
1.423601 1.674397 1.585055
```

Regularisation techniques such as ridge regression, lasso regression, or elastic net can also be used to deal with multicollinearity. These advanced techniques add a penalty term to the regression model that shrinks the coefficients towards zero, which can help to reduce the impact of multicollinearity. However, these techniques are not covered in this guide. Please refer to Chapter 8 for more information on regularisation techniques.

5.6.5 Perform Forward Selection

It might be that not all of the variables included in the full model are necessary to explain the response variable. We can use a stepwise regression to select the best combination (subset) of predictors that best explains the response variable. To do this, we will use the `stepAIC` function that lives in the `MASS` package.

`stepAIC()` works by starting with the null model and then adding predictors one by one, selecting the one that improves the model the most as seen in the reduction of the AIC values along the way. This process continues until no more predictors can be added to improve the model (i.e. to further reduce the AIC). Progress is tracked as the function runs.

```
  # Perform forward selection
  mod1 <- stepAIC(null_mod1,
                  scope = list(lower = null_mod1, upper = full_mod1),
                  direction = "forward")
```

```
Start:  AIC=-1044.97
Y ~ 1

          Df Sum of Sq    RSS     AIC
+ augMean  1    6.0084 1.7108 -1478.4
+ febSD    1    3.0759 4.6433 -1189.9
+ augSD    1    2.6394 5.0797 -1163.9
<none>                 7.7192 -1045.0

Step:  AIC=-1478.41
Y ~ augMean

        Df Sum of Sq    RSS     AIC
+ febSD  1   0.36036 1.3504 -1544.8
```

```
+ augSD  1   0.31243 1.3984 -1534.7
<none>               1.7108 -1478.4

Step:  AIC=-1544.77
Y ~ augMean + febSD

        Df Sum of Sq    RSS     AIC
+ augSD  1   0.10568 1.2448 -1566.3
<none>               1.3504 -1544.8

Step:  AIC=-1566.32
Y ~ augMean + febSD + augSD
```

The model selection process shows that as we add more variables to the model, the AIC value decreases. We can infer from this that the multiple regression model provides a better fit that simple linear models that use the variables in isolation.

We also see that `stepAIC()` has not removed any variables from the full model. Probably one reason for failing to remove any variables is that the VIF process has already accomplished this by virtue of dealing with multicollinearity. This means that all the variables retained in `mod1` are important in explaining the response variable.

5.6.6 Added-Variable Plots (Partial Regression Plots)

Before looking at the output in more detail, I'll introduce partial regression plots as a means to examine the relationship between the response variable and each predictor variable. Although they can be calculated by hand, the **car** package provides a convenient function, `avPlots()`, to create these plots.

Added variable plots are also sometimes called 'partial regression plots' or 'individual coefficient plots.' They are used to display the relationship between a response variable and an individual predictor variable while accounting for the effect of other predictor variables in a multiple regression model (the marginal effect).

```
# Create partial regression plots
avPlots(mod1, col = "dodgerblue4", col.lines = "magenta")
```

What insights can we draw from the added-variable plots? Although there are better ways to assess the model fit, we can already make some observations about the linearity of the model or the presence of outliers. The slope of the line in an added variable plot corresponds to the regression coefficient for that predictor in the full multiple regression model. Seen in this way, it visually indicates the magnitude and direction of each predictor's effect. In Figure 5.2, the added-variable plot for `augMean` shows a tighter clustering of points around the regression line and a strong linear relationship (steep slope) with the response variable; the plots for `febSD` and `augSD`, on the other hand, show a weaker response and more scatter about the regression line. Importantly, this suggests that `augMean` has a stronger and more unique contribution to the multiple-variable model than the other two variables.

There are also insights to be made about possible multicollinearity using added-variable plots. These plots are not a definitive test for multicollinearity, but they can provide some clues. Notably,

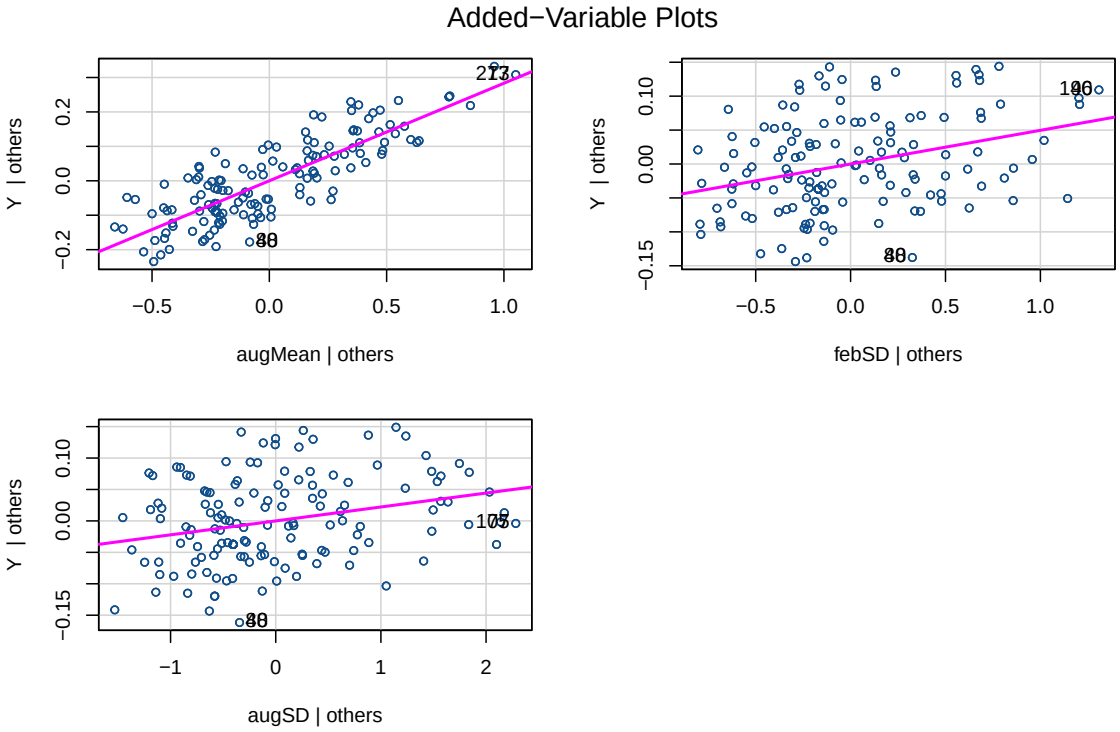

Figure 5.2: Partial regression plots for `mod1` with the selected variables `augMean`, `febSD`, and `augSD`.

if a predictor shows a strong relationship with the response variable in a simple correlation but appears to have little relationship in the added-variable plot, it might indicate collinearity with other predictors. This discrepancy suggests that the predictor's effect on the response is being masked by the presence of other correlated predictors.

5.6.7 Model Diagnostics

We are back in the territory of parametric statistics, so we need to check the assumptions of the multiple linear regression model (similar to those of simple linear regression). We can do this by making the various diagnostic plots. all of them consider various aspects of the residuals, which are simply the differences between the observed and predicted values.

Diagnostic plots of final model

You have been introduced to diagnostic plots in the context of simple linear regression (Section 3.1). They are also useful in multiple linear regression. Although `plot.lm()` can easily do this, here I use `autoplot()` from the **ggfortify** package. When applied to the final model, `mod1`, the plot will in its default setting show four diagnostic plots: residuals vs. fitted values, normal Q-Q plot, scale-location plot, and residuals vs. leverage plot. Note, this is for the full model inclusive of the combined contributions of all the predictors, so we will not see separate plots for each predictor as we have seen in the added-variable plots or component plus residual plots.

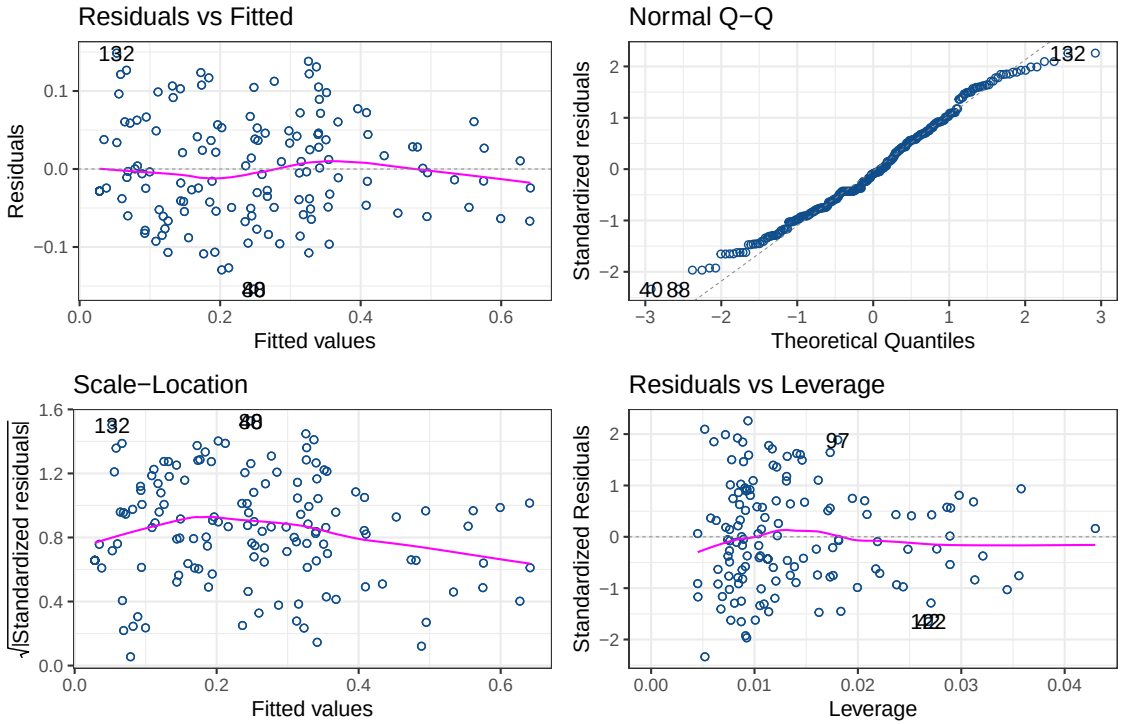

Figure 5.3: Diagnostic plots to assess the fit of the final multiple linear regression model, `mod1`.

```
# Generate diagnostic plots
autoplot(mod1, shape = 21, colour = "dodgerblue4",
         smooth.colour = "magenta") +
  theme_bw()
```

Residuals vs. Fitted Values: In this plot we can assess linearity and homoscedasticity of the residuals. If the seaweed gods were with us, we'd expect the points to be randomly scattered about a horizontal line situation at zero. This would indicate that the relationship between the predictors selected by the forward selection process (`augMean`, `febSD`, and `augSD`) and the response variable (`Y`) is linear, and the variance of the residuals is constant across the range of fitted values. In this plot, there's a very slight curvature which might suggest a potential issue with the linearity assumption—it is minute and I'd suggest not worrying about it. The variance of the residuals seems to decrease slightly at higher fitted values, indicating a mild case of heteroscedasticity.

Q-Q Plot (Quantile-Quantile Plot): This plot is used to check the normality of the residuals. The points should fall approximately along a straight diagonal line if the residuals are normally distributed. Here we see that the points generally follow the line although some deviations may be seen at the tails. These deviations are not that extreme and again I don't think this is not a big concern.

Scale-Location Plot: This plot should reveal potential issues with homoscedasticity. The square root of the standardised residuals is used here to make it easier to spot patterns, so we would like the points to be randomly scattered around the horizontal red line. Here, the line slopes slightly

downward and this indicates that the variance of the residuals might decrease as the fitted values increase. We can also see evidence of this in a plot of the observed values vs. the predictors in Figure 5.3.

Residuals vs. Leverage: This diagnostic highlights influential points (outliers). Points with high leverage (far from the mean of the predictors) can be expected to exert a strong influence on the regression line, tilting it in some direction. Cook's distance (indicated by the yellow line) helps identify such outliers. In our seaweed data a few points could have a high leverage, but since they don't seem to cross the Cook's distance thresholds, I doubt they are overly worrisome.

Considering that no glaring red flags were raised by the diagnostic plots, I doubt that they are severe enough to invalidate the model. However, if you cannot stand these small issues, you could i) consider transforming the predictor or response variables to address your concerns about heteroscedasticity, ii) investigate the outliers (high leverage points) to confirm if they are valid data points or errors, or iii) try robust regression methods that are less sensitive to outliers and heteroscedasticity.

Component plus residual plots

Component plus residual plots offer another way to assess the fit of the model in multiple regression models. Unlike simple linear regression where we only had one predictor variable, here we have several. So, we need to assure ourselves that there is a linear relationship between each predictor variable and the response variable (we could already see this in the added-variable plots in Section 5.6.6). We can make component plus residual plots using the `crPlots()` function in the **car** package. It displays the relationship between the response variable and each predictor variable. If the relationship is linear, the points should be randomly scattered about a best fit line and the spline (in pink in Figure 5.4) should plot nearly on top of the linear regression line.

```
# Generate component plus residual plots
crPlots(mod1, col = "dodgerblue4", col.lines = "magenta")
```

5.6.8 Understanding the Model Fit

The above model selection process has led us to the `mod1` model, which can be stated formally as:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{5.4}$$

Where:

- Y: The response variable, the mean Sørensen dissimilarity.
- X_1, X_2, and X_3: The predictors corresponding to `augMean`, `febSD`, and `augSD`, respectively.
- ϵ: The error term.

We have convinced ourselves that the model is a good fit for the data, and we can proceed to examine the model's output. The fitted model can be explored in two ways: by applying the `summary()` function or by using the `anova()` function. The `summary()` function provides a detailed output of the model, while the `anova()` function provides a table of deviance values that can be used to compare models.

The model summary

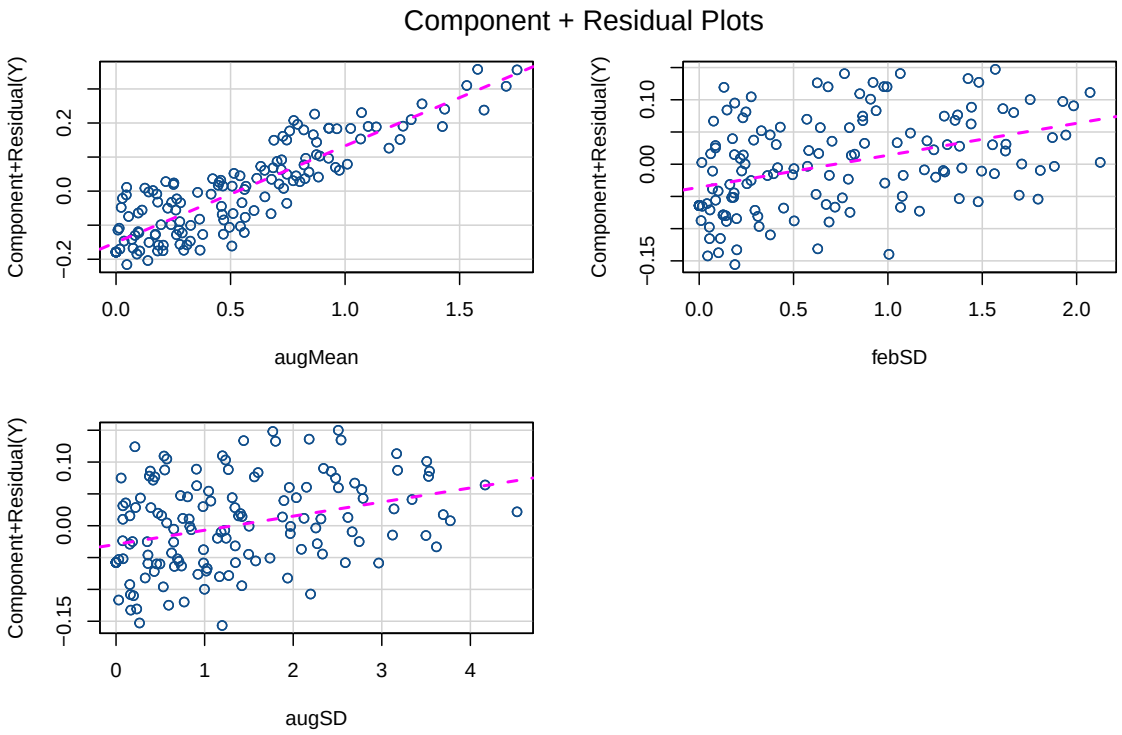

Figure 5.4: Component plus residual diagnostic plots to assess the fit of the final multiple linear regression model, `mod1`.

```
# Summary of the selected model
summary(mod1)
```

```
Call:
lm(formula = Y ~ augMean + febSD + augSD, data = sw_sub1)

Residuals:
      Min        1Q    Median        3Q       Max
-0.153994 -0.049229 -0.006086  0.045947  0.148579

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.028365   0.007020   4.040 6.87e-05 ***
augMean     0.283335   0.011131  25.455  < 2e-16 ***
febSD       0.049639   0.008370   5.930 8.73e-09 ***
augSD       0.022150   0.004503   4.919 1.47e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06609 on 285 degrees of freedom
Multiple R-squared:  0.8387,    Adjusted R-squared:  0.837
```

```
F-statistic: 494.1 on 3 and 285 DF,  p-value: < 2.2e-16
```

The first part of the `summary()` function's output is the `Coefficients` section. This is where the main effects hypotheses are tested (this model does not have interactions—if there were, they'd appear here, too). The important components of the coefficients part of the model summary are:

- `(Intercept)`: This row provides information about where the regression line intersects the *y*-axis.
- Main Effects:
  - `augMean`, `febSD`, and `augSD`: These rows give the model coefficients associated with the slopes of the regression lines fit to those predictor variables. They indicate the rate of change in the response variable for a one-unit change in the predictor variable.
  - `Estimate`, `Std. Error`, `t value`, and `Pr(>|t|)`: These columns contain the statistics used to interpret the hypotheses about the main effects. In the `Estimate` column are the coefficients for the *y*-intercept and the main effects' slopes, and `Std. Error` indicates the variability of the estimate. The `t value` is obtained by dividing the coefficient by its standard error. The *p*-value tests the null hypothesis that the coefficient is equal to zero and significance codes are provided as a quick visual reference (their use is sometimes frowned upon by statistics purists). Using this information, we can quickly see that, for example, `augMean` has a coefficient of 0.2833 ± 0.0111 and the slope of the line is highly significant, i.e. there is a significant effect of `Y` due to the temperature gradient set up by `augMean`.

The intercept and slope coefficients

The interpretation of the coefficients is a bit more complicated in multiple linear regression compared to what we are accustomed to in simple linear regression. Let us look at some greater detail at the intercept and the slope coefficients:
Intercept (α):) The intercept is the expected value of the response variable, Y, when all predictor variables are zero. It is not always meaningful, but it can be useful in some cases.
Slope Coefficients ($\beta_1, \beta_2, \ldots, \beta_k$): Each slope coefficient, β_j, represents the expected change in the response variable, Y, for a one-unit increase in the predictor variable, X_j, holding all other predictor variables constant. This partial effect interpretation implies that β_j accounts for the direct contribution of X_j to Y while removing the confounding effects of other predictors in the model. Figure 5.2 provides a visual representation of this concept and isolates the effect of each predictor variable on the response variable.
Therefore, in the context of our model (Equation 5.4) for this analysis, the partial interpretation is as follows:

- β_1: Represents the change in Y for a one-unit increase in X_1, holding X_2 and X_3 constant.
- β_2: Represents the change in Y for a one-unit increase in X_2, holding X_1 and X_3 constant.
- β_3: Represents the change in Y for a one-unit increase in X_3, holding X_1 and X_2 constant.

There are also several overall model fit statistics—it is here where you'll find the information you need to assess the hypothesis about the overall significance of the model. `Residual standard error` indicates the average distance between observed and fitted values. `Multiple R-squared` and `Adjusted R-squared` values tell us something about the model's goodness of fit. The latter adjusts for the number of predictors in the model, and is the one you must use and report in multiple linear regressions. As you also know, higher numbers approaching 1 are better, with 1 suggesting that the model perfectly captures all of the variability in the data. The `F-statistic`

and its associated *p*-value test the overall significance of the model and examines whether all regression coefficients are simultaneously equal to zero. You can also use the brief overview of the residuals, but I don't find this particularly helpful—best examine the residuals in a histogram.

The ANOVA tables

```
anova(mod1)
```

```
Analysis of Variance Table

Response: Y
           Df Sum Sq Mean Sq  F value    Pr(>F)
augMean     1 6.0084  6.0084 1375.660 < 2.2e-16 ***
febSD       1 0.3604  0.3604   82.507 < 2.2e-16 ***
augSD       1 0.1057  0.1057   24.196 1.473e-06 ***
Residuals 285 1.2448  0.0044
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This function provides a sequential analysis of variance (Type I ANOVA) table for the regression model (see more about Type I ANOVA, below). As such, this function can also be used to compare nested models. Used on a single model, it gives a more interpretable breakdown of the variability in the response variable `Y` and assesses the contribution of each predictor variable in explaining this variability.

The ANOVA table firstly shows the degrees of freedom (`Df`) for each predictor variable added sequentially to the model, as well as the residuals. For each predictor, the degrees of freedom is typically 1. For the residuals, however, it represents the total number of observations minus the number of estimated parameters. The Sum of Squares (`Sum Sq`) indicates the variability in `Y` attributable to each predictor, and the mean sum of squares (`Mean Sq`) is the sum of squares divided by the degrees of freedom.

The `F value` is calculated as the ratio of the predictor's mean square to the residual mean square tests. It is used in testing the null hypothesis that the predictor has no effect on `Y`. Whether or not we accept the alternative hypothesis (reject the null) is given by the *p*-value (`Pr(>F)`) that goes with each *F*-statistic. You know how that works.

Because this is a sequential ANOVA, the amount of variance in `Y` explained by each predictor (or group of predictors) is calculated by adding the predictors to the model in sequence (as specified in the model formula). For example, the Sum of Squares for `augMean` (6.0084) represents the amount of variance explained by adding `augMean` to a model that doesn't include any predictors yet. The Sum of Squares for `febSD` 0.3604) represents the amount of variance explained by adding `febSD` to a model that already includes `augMean`—this improvement indicates that `febSD` explains some of the variance in `Y` that `augMean` doesn't.

> **Order in which predictors are assessed in multiple linear regression**
>
> The interpretation of sequential ANOVA (Type I) is inherently dependent on the order in which predictors are entered. In `mod1` the order is first `augMean`, then `febSD`, and last comes `augSD`. This order might not be the most meaningful for interpreting the sequential sums

of squares and their significance in the ANOVA table. How, then, does one decide on the order of predictors in the model?

- If you have a strong theoretical or causal basis for thinking that certain predictors influence others, you can enter them in that order.
- If you have a hierarchy of predictors based on their importance or general vs. specific nature, you can enter them hierarchically.
- You can manually fit models with different predictor orders and compare the ANOVA tables to see how the results change. This can be time-consuming but might offer insights into the sensitivity of your conclusions to the order of entry.
- You can use automated model selection procedures, such as stepwise regression, to determine the best order of predictors. This is a more objective approach but can be criticised for being data-driven and not theory-driven.
- Use Type II or Type III ANOVAs, which are are not order-dependent and can be used to assess the significance of predictors after accounting for all other predictors in the model. However, they have their own limitations and assumptions that need to be considered.

My advice would be to have sound theoretical reasons for the order of predictors in the model.

Both ways of looking at the model fit of `mod1`—`summary()` and `anova()`—show that forward selection retained the variables `augMean`, `febSD`, and `augSD`. These three predictors should be used together to explain the response, `Y`.

Let's make a plot of the full model with all the initial predictors and the selected model with the predictors chosen by the forward selection process.

```
# Add fitted values from the selected model to the dataframe
sw_ectz$.fitted_selected <- fitted(mod1)

# Create the plot of observed vs fitted values for the selected model
ggplot(sw_ectz, aes(x = .fitted_selected, y = Y)) +
  geom_point(shape = 1, colour = "black", alpha = 1.0) +
  geom_point(aes(x = .fitted), colour = "red",
             shape = 1, alpha = 0.4) +
  geom_abline(intercept = 0, slope = 1,
              color = "blue", linetype = "dashed") +
  labs(x = "Fitted Values",
       y = "Observed Values") +
  theme_bw()
```

5.6.9 Reporting

A Results section should be written in a format sutable for inclusion in your report or publication. Present the results in a clear and concise manner, with tables and figures used to help substantiate your findings. The results should be interpreted in the context of the research question and the study design. The limitations of the analysis should also be discussed, along with any potential sources of bias or confounding. Here is an example.

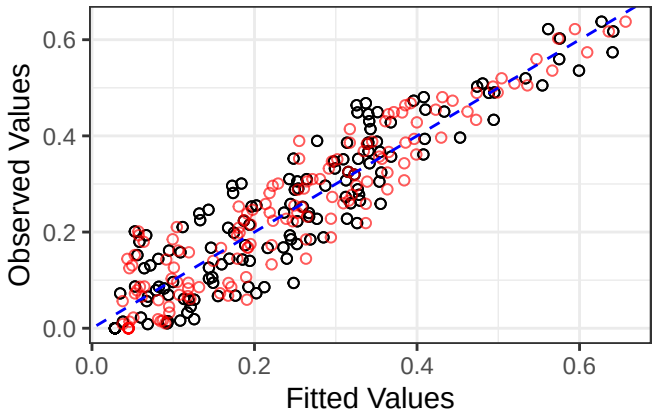

Figure 5.5: Plot of observed vs. predicted value obtained from the final multiple linear regression model (`mod`) with the selected variables `augMean`, `febSD`, and `augSD` as predictors (black points), and the initial model with also `annMean` and `febRange` (red points).

Results

The model demonstrates a strong overall fit, as indicated by the high R^2 value of 0.839 and an adjusted R^2 of 0.837, suggesting that approximately 83.7% of the variance in the mean Sørensen dissimilarity is explained by the predictors `augMean`, `febSD`, and `augSD`. All predictors in the model are statistically significant, with `augMean` showing the strongest effect ($\beta_1 = 0.283$, $p < 0.0001$) (Figure 5.2). The predictors `febSD` and `augSD` also have significant positive relationships with the response variable ($\beta_2 = 0.050$, $p = 0.0001$; $\beta_3 = 0.022$, $p = 0.0001$). A sequential ANOVA further confirms the significance of each predictor variable in the model, with all F-values indicating that the inclusion of each predictor significantly improves the model fit ($p < 0.0001$ in all cases). Our model therefore provides clear support for the mean temperatures in August, the standard deviation of temperatures in February, and the standard deviation of temperatures in August as strong predictors of the mean Sørensen dissimilarity, with each contributing uniquely to the explanation of variability in the response variable.

5.7 Example 2: Interaction of Distance and Bioregion

Our seaweed dataset includes two additional variables that we have not yet considered. These are the continuous variable `dist` which represents the geographic distance between the seaweed samples taken along the coast of South Africa, and the categorical variable `bio` which is the bioregional classification of the seaweed samples.

These two new variables lend themselves to a few interesting questions. For example:

1. Is the geographic distance between samples related to the Sørensens dissimilarity of the seaweed flora?
2. Does the average Sørensens dissimilarity vary among the bioregions to which the samples belong?
3. Is the effect of geographic distance on the Sørensens dissimilarity different for each bioregion?

The most complex model is (3), the one that answers the question about whether the effect of `dist` on the response variable *Y* is different for each bioregion. Questions (1) and (2) are subsets of this more inclusive question. To fully answer these quesitons, let's first consider the full model, which includes an *interaction term* between the continuous predictor `dist` and the categorical predictor `bio`. When we finally test our model, we will also have to consider the simpler models that do not include the interaction term.

'Interaction' means that the effect of one predictor on the response variable is contingent on the value of another predictor. For example, we might have reason to suspect that the relationship of the Sørensens dissimilarity with the geographic distance between samples is different between the west coast compared to, say, the east coast. This is indeed a plausible expectation, but we will test this formally below.

The full multiple linear regression model with the interaction terms can be formally expressed as Equation 5.5:

$$\begin{aligned} Y = \alpha &+ \beta_1 \text{dist} + \beta_2 \text{bio}_{\text{B-ATZ}} + \beta_3 \text{bio}_{\text{BMP}} \\ &+ \beta_4 \text{bio}_{\text{ECTZ}} + \beta_5 (\text{dist} \times \text{bio}_{\text{B-ATZ}}) \\ &+ \beta_6 (\text{dist} \times \text{bio}_{\text{BMP}}) + \beta_7 (\text{dist} \times \text{bio}_{\text{ECTZ}}) + \epsilon \end{aligned} \tag{5.5}$$

Where:

- Y: The response variable, the mean Sørensen dissimilarity.
- α: The intercept term.
- dist: The continuous predictor variable representing distance.
- bio: The categorical predictor variable representing bioregional classification with four levels: `AMP` (reference category), `B-ATZ`, `BMP`, and `ECTZ`.
- $\text{bio}_{\text{B-ATZ}}, \text{bio}_{\text{BMP}}, \text{bio}_{\text{ECTZ}}$: Dummy variables for the bioregional classification, where:
  - $\text{bio}_{\text{B-ATZ}} = 1$ if `bio = B-ATZ`, and 0 otherwise,
  - $\text{bio}_{\text{BMP}} = 1$ if `bio = BMP`, and 0 otherwise, and
  - $\text{bio}_{\text{ECTZ}} = 1$ if `bio = ECTZ`, and 0 otherwise.
- $\text{dist} \times \text{bio}_{\text{B-ATZ}}, \text{dist} \times \text{bio}_{\text{BMP}}, \text{dist} \times \text{bio}_{\text{ECTZ}}$: Interaction terms between distance and the bioregional classification dummy variables.
- $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7$: The coefficients to be estimated for the main effects and interactions.
- ϵ: The error term.

If this seems tricky, it is because of the dummy variable coding used to represent interactions in multiple linear regression. The `bio` variable is a categorical variable with four levels, so we need to create three dummy variables to represent the bioregional classification. The `dist` variable is then interacted with each of these dummy variables to create the interaction terms. The `lm()` function in R takes care of this for us in a far less complicated model statement. I'll explain the details around the interpretation of dummy variable coding when we look at the output of the model with the `summary()` function.

5.7.1 State the Hypotheses for a Multiple Linear Regression with Interaction Terms

Equation 5.5 expands into the following series of hypotheses that concern the main effects, the interactions between the main effects, and the overall hypothesis:

Main effects hypotheses

In the main effects hypotheses we are concerned with the effect of each predictor variable on the response variable. For the main effect of distance we have the null:

- $H_0 : \beta_1 = 0$

vs. the alternative:

- $H_A : \beta_1 \neq 0$

For the main effect of bioregional classification, the nulls are:

- $H_0 : \beta_2 = 0 \quad (\text{bio}_{\text{B-ATZ}})$
- $H_0 : \beta_3 = 0 \quad (\text{bio}_{\text{BMP}})$
- $H_0 : \beta_4 = 0 \quad (\text{bio}_{\text{ECTZ}})$

vs. the alternatives:

- $H_A : \beta_2 \neq 0 \quad (\text{bio}_{\text{B-ATZ}})$
- $H_A : \beta_3 \neq 0 \quad (\text{bio}_{\text{BMP}})$
- $H_A : \beta_4 \neq 0 \quad (\text{bio}_{\text{ECTZ}})$

Hypotheses about interactions

This is where the hypothesis tests whether the effect of distance on the response variable is different for each bioregional classification. The null hypotheses are:

- $H_0 : \beta_5 = 0 \quad (\text{dist} \times \text{bio}_{\text{B-ATZ}})$
- $H_0 : \beta_6 = 0 \quad (\text{dist} \times \text{bio}_{\text{BMP}})$
- $H_0 : \beta_7 = 0 \quad (\text{dist} \times \text{bio}_{\text{ECTZ}})$

vs. the alternatives:

- $H_A : \beta_5 \neq 0 \quad (\text{dist} \times \text{bio}_{\text{B-ATZ}})$
- $H_A : \beta_6 \neq 0 \quad (\text{dist} \times \text{bio}_{\text{BMP}})$
- $H_A : \beta_7 \neq 0 \quad (\text{dist} \times \text{bio}_{\text{ECTZ}})$

Overall hypothesis

The overall hypothesis states that all coefficients associated with the predictors (distance, bioregional categories, and their interactions) are equal to zero, therefore indicating no relationship between these predictors and the response variable, the Sørensen index. The null hypothesis is:

- $H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = \beta_7 = 0$

vs. the alternative:

- $H_A : \exists\, \beta_i \neq 0$ for at least one i

5.7.2 Visualise the Main Effects

To facilitate the interpretation of the main effects hypotheses and make an argument for why an interaction term might be necessary, I've visualised the main effects (Figure 5.6). I see this as part of my exploratory data analysis ensemble of tests. We see that fitting a straight line to the Y vs. distance relationship seems unsatisfactory as there is too much scatter around that single line to adequately capture all the structure in the variability of the points. Colouring the points

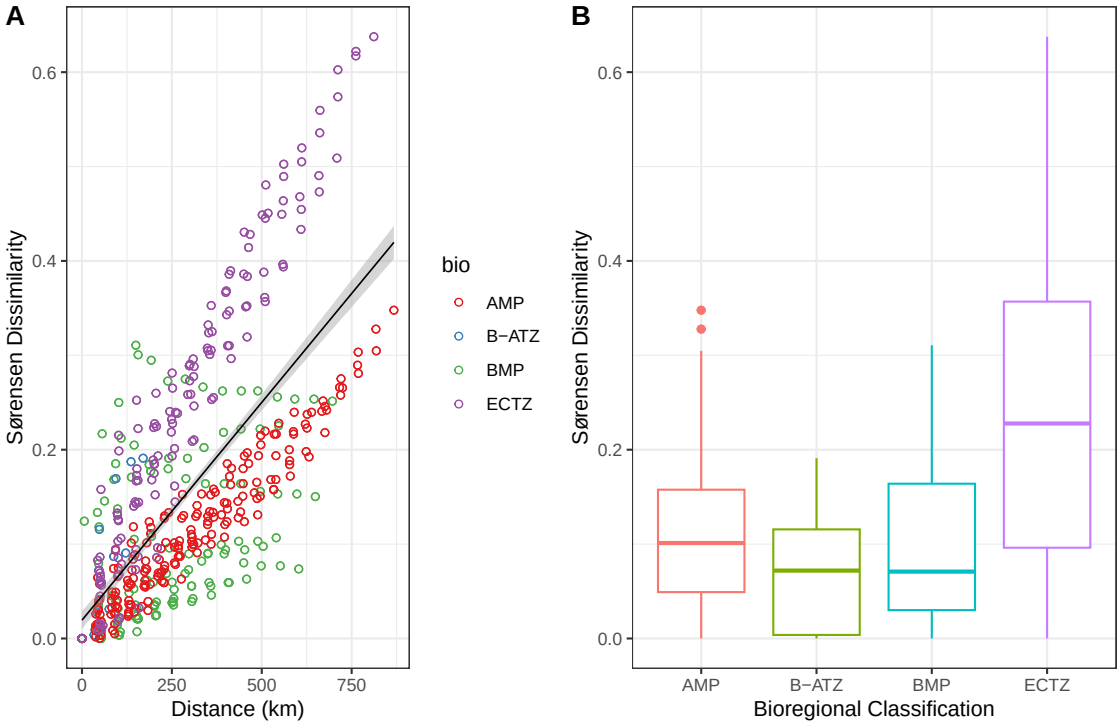

Figure 5.6: Plot of main effects of A) distance along the coast and B) bioregional classification on the Sørensen dissimilarity index.

by bioregion reveals the hidden structure. The model could benefit from including an additional level of complexity: see how points in the same bioregion show less scatter compared to points in different bioregions.

Now look at the boxplots of the Sørensen dissimilarity index for each bioregional classification. It shows that the median values of the Sørensen dissimilarity index are different for each bioregion. Taken together, Figure 5.6 (A, B) provide a good indication that adding the bioregional classification might be an important predictor of the Sørensen dissimilarity index as a function of distance between pairs of sites along the coast.

Next, we will move ahead and fit the model inclusive of the distance along the coast and bioregion as per Equation (5.5).

5.7.3 Fit and Assess Nested Models

I have a suspicion that the full model (`mod2`; see below) with the interaction terms will be a better fit than reduced models with only the effect due to distance (seen independently). How can we have greater certainty that we should indeed favour a slightly more complex model (with two predictors) over a simpler one with only (distance only)?

One way to do this is to use a nested model comparison. We will fit a reduced model (one slope for all bioregions) and compare this model to the full model (slopes are allowed to vary among bioregions).

```r
# Fit the linear regression model with only distance
mod2a <- lm(Y ~ dist, data = sw)

# Fit the multiple linear regression model with interaction terms
mod2 <- lm(Y ~ dist * bio, data = sw)
```

This is a nested model where `mod2a` is nested within `mod2`. 'Nested' means that the reduced model is a subset of the full model. Nested models can be used to test hypotheses about the significance of the predictors in the full model—does adding more predictors to the model improve the fit? Comparing a nested model with a full model can be done with a sequential ANOVA, which is what the `anova()` function also does (in addition to its use in Section 5.6.8).

So, comparing `mod2a` to `mod2` with an *F*-test tests the significance of adding the `bio` and using it together with `dist`. The interaction is built into `mod2` but we are not yet testing the significance of the interaction terms. We will do that later.

```r
anova(mod2a, mod2, test = "F")
```

```
Analysis of Variance Table

Model 1: Y ~ dist
Model 2: Y ~ dist * bio
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    968 7.7388
2    962 2.2507  6    5.4881 390.95 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The sequential ANOVA shows that there is significant merit to consider an interaction term in the model. This model would then allow us to have a separate slope for the Sørensen index as function of distance for each bioregion. The residual sum of squares (RSS) decreases from 7.7388 in Model 1 to 2.2507 in Model 2, which indicates that Model 2 explains a significantly larger proportion of the variance in the response variable. The *F*-test for comparing the two models yields an *F*-value of 390.95 with a highly significant *p*-value (< 0.0001). The improvement in model fit due to the inclusion of the interaction term is therefore statistically significant.

The above analyses skirted around the questions stated in the beginning of Section 5.7. I've provided statistical evidence that full model is a better fit than the reduced model (the sequential *F*-test tested this), so we should use both `dist` and `bio` in the model. I have not looked explicitly at the main effects of the predictors. However, we can easily address questions (1) and (2):

- Question 1: looking at the summary of `mod2a` tells us that the main effect of `dist` is a significant ($p < 0.0001$) predictor of the Sørensen dissimilarity index.
- Question 2: the main effect of `bio` is also significant ($p < 0.0001$), which is what we'd see if we fit the model `mod2b ← lm(Y ~ bio, data = sw)`.

Question 3 warrants deeper investigation. Next, we will look at the interaction terms in the full model `mod2` to see if the effect of `dist` on `Y` is different for each level of `bio`.

5.7.4 Interpret the Full Model

The model summary

```
# Summary of the model
summary(mod2)
```

```
Call:
lm(formula = Y ~ dist * bio, data = sw)

Residuals:
      Min        1Q    Median        3Q       Max
-0.112117 -0.030176 -0.004195  0.023698  0.233520

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.341e-03  4.177e-03   1.279   0.2013
dist           3.530e-04  1.140e-05  30.958  < 2e-16 ***
bioB-ATZ      -6.140e-03  1.659e-02  -0.370   0.7114
bioBMP         3.820e-02  6.659e-03   5.737 1.29e-08 ***
bioECTZ        1.629e-02  6.447e-03   2.527   0.0117 *
dist:bioB-ATZ  7.976e-04  1.875e-04   4.255 2.30e-05 ***
dist:bioBMP   -1.285e-04  2.065e-05  -6.222 7.31e-10 ***
dist:bioECTZ   4.213e-04  1.801e-05  23.392  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04837 on 962 degrees of freedom
Multiple R-squared:  0.8607,    Adjusted R-squared:  0.8597
F-statistic: 849.2 on 7 and 962 DF,  p-value: < 2.2e-16
```

In the output returned by `summary(mod2)`, we need to pay special attention to the use of dummy variable encoding for the categorical predictor. The `Coefficients` section is similar to that of `mod1` (see Section 5.6.8), but now it includes the categorical predictor `bio*` and the interaction terms `dist:bio*` (`*` indicating the levels of the categorical variable). The `bio` variable has four levels, `BMP`, `B-ATZ`, `AMP`, and `ECTZ`, and `AMP` is selected as reference level. This decision to selected `AMP` as reference is entirely arbitrary, and alphabetical sorting offers a convenient approach to selecting the reference. The coefficients for the other levels of `bio` are interpreted as the sum of the response variable and the reference level.

The following are the key coefficients in the model summary:

- `(Intercept)`: This is the estimated average value of `Y` when `dist` is zero and `bio` is the reference category (`AMP`). Its *p*-value (> 0.05) suggests it's not significantly different from zero.
- Main Effects:
  - `dist`: This represents the estimated change in `Y` for a one-unit increase in `dist` when the bioregion is the reference category, `AMP`. The highly significant *p*-value (< 0.0001) indicates a strong effect of distance in the `AMP`.
  - `bioB-ATZ`, `bioBMP`, `bioECTZ`: These are dummy variables representing different biore-

gions. Their coefficients indicate the difference in the average value of `Y` between each of these bioregions and the reference bioregion when `dist` is zero. Only `bioBMP` and `bioECTZ` are significantly different from the reference bioregion, `AMP`.

- Interaction Effects:
  - `dist:bioB-ATZ`, `dist:bioBMP`, `dist:bioECTZ`: These interaction terms capture how the effect of `dist` on `Y` varies across different bioregions. For instance, `dist:bioB-ATZ` indicates the additional change in the effect of `dist` in the `B-ATZ` bioregion compared to the reference bioregion, `AMP`. All interaction terms are highly significant, suggesting the effect of distance is different across bioregions.

Given this explanation, we can now interpret the coefficients of, for example, the `bioB-ATZ` main effect and `dist:bioB-ATZ` interaction. Since `AMP` is the reference bioregion, its effect is absorbed into the intercept term. Therefore, the coefficient for `bioB-ATZ` directly reflects the difference we are interested in. The coefficient for `bioB-ATZ` is -0.0061 $\pm$ 0.0166 lower than that of the reference, but the associated *p*-value (> 0.05) indicates that the average value of `Y` in the `B-ATZ` bioregion is not significantly different from the reference bioregion, `AMP`.

If we'd want to report the actual coefficient for `B-ATZ`, we'd calculate the sum of the coefficients for (`Intercept`) and `bioB-ATZ`. This would give us the estimated average value of `Y` in the `B-ATZ` bioregion when `dist` is zero. The associated SE is calculated as the square root of the sum of the squared SEs of the two coefficients. Therefore, the coefficient for `B-ATZ` is $-8 \times 10^{-4} \pm 0.0171$.

The coefficient of 8×10^{-4} for `dist:bioB-ATZ` indicates that the effect of distance on `Y` is 8×10^{-4} units greater in the `B-ATZ` bioregion compared to the `AMP` bioregion. The SE of 2×10^{-4} suggests a high level of precision in this estimate, and the *p*-value (< 0.0001) indicates that this difference is statistically significant.

As before, to calculate the actual coefficient for `dist` in the `B-ATZ` bioregion, we'd sum the coefficients for `dist` and `dist:bioB-ATZ`. The associated SE of this sum is calculated as the square root of the sum of the squared SEs of the two coefficients. Therefore, the coefficient for `dist` in the `B-ATZ` bioregion is $0.0012 \pm 2 \times 10^{-4}$.

Concerning the overall hypothesis, the `Adjusted R-squared` value of 0.8597 indicates that the model explains 85.97% of the variance in the response variable `Y`. The `F-statistic` and associated `p-value` (< 0.0001) indicate that the model as a whole is highly significant, meaning at least one of the predictors (including interactions) has a significant effect on `Y`.

The ANOVA table

```
# The ANOVA table
anova(mod2)
```

```
Analysis of Variance Table

Response: Y
           Df Sum Sq Mean Sq F value    Pr(>F)
dist        1 8.4199  8.4199 3598.79 < 2.2e-16 ***
bio         3 3.6232  1.2077  516.21 < 2.2e-16 ***
dist:bio    3 1.8648  0.6216  265.69 < 2.2e-16 ***
Residuals 962 2.2507  0.0023
---
```

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

The ANOVA table's interpretation is intuitive and simple: the `Pr(>F)` column shows the *p*-value for each predictor in the model. The `dist` predictor has a highly significant effect on `Y` (< 0.0001), as do all the bioregions and their interactions with `dist`. This confirms the results we obtained from the coefficients. We don't need to overthink this result.

5.8 Example 3: The Final Model

I'll now expand `mod1` to include `bio` as a predictor alongside `augMean`, `febSD`, and `augSD` (`mod1` was applied only to data pertaining to `ECTZ`, one of the four levels in `bio`).

$$
\begin{aligned}
Y = {} & \alpha + \beta_1 \text{augMean} + \beta_2 \text{febSD} + \beta_3 \text{augSD} \\
& + \beta_4 \text{bio}_{\text{B-ATZ}} + \beta_5 \text{bio}_{\text{BMP}} + \beta_6 \text{bio}_{\text{ECTZ}} \\
& + \beta_7 (\text{augMean} \times \text{bio}_{\text{B-ATZ}}) + \beta_8 (\text{augMean} \times \text{bio}_{\text{BMP}}) \\
& + \beta_9 (\text{augMean} \times \text{bio}_{\text{ECTZ}}) + \beta_{10} (\text{febSD} \times \text{bio}_{\text{B-ATZ}}) \\
& + \beta_{11} (\text{febSD} \times \text{bio}_{\text{BMP}}) + \beta_{12} (\text{febSD} \times \text{bio}_{\text{ECTZ}}) \\
& + \beta_{13} (\text{augSD} \times \text{bio}_{\text{B-ATZ}}) + \beta_{14} (\text{augSD} \times \text{bio}_{\text{BMP}}) \\
& + \beta_{15} (\text{augSD} \times \text{bio}_{\text{ECTZ}}) + \epsilon
\end{aligned} \tag{5.6}
$$

Where:

- Y: The response variable (mean Sørensen dissimilarity).
- α: The intercept term, representing the expected value of `Y` when all predictors are zero and `bio` is at the reference level `AMP`).
- β_1: The coefficient for the main effect of `augMean`.
- β_2: The coefficient for the main effect of `febSD`.
- β_3: The coefficient for the main effect of `augSD`.
- $\beta_4, \beta_5, \beta_6$: The coefficients for the main effects of the categorical predictor `bio` (for levels `B-ATZ`, `BMP`, and `ECTZ` respectively, with `AMP` as the reference category).
- $\beta_7, \beta_8, \beta_9$: The coefficients for the interaction effects between `augMean` and `bio` (for levels `B-ATZ`, `BMP`, and `ECTZ` respectively).
- $\beta_{10}, \beta_{11}, \beta_{12}$: The coefficients for the interaction effects between `febSD` and `bio` (for levels `B-ATZ`, `BMP`, and `ECTZ` respectively).
- $\beta_{13}, \beta_{14}, \beta_{15}$: The coefficients for the interaction effects between `augSD` and `bio` (for levels `B-ATZ`, `BMP`, and `ECTZ` respectively).
- ϵ: The error term, representing the unexplained variability in the response variable.

In this multiple regression model, we aim to understand the complex and interacting relationships between the response variables and the set of predictors. It allows us to investigate not only the individual effects of the continuous predictors on `Y`, but also how these effects might vary across the different bioregions.

The model therefore incorporates interaction terms between each continuous predictor (`augMean`, `febSD`, and `augSD`) and the categorical variable `bio`. This allows us to assess whether the relationships between `augMean`, `febSD`, or `augSD` and `Y` change depending on the specific bioregion.

Essentially, we are testing whether the slopes of these relationships are different in different bioregions.

Additionally, the model examines the main effects of the bioregions themselves on `Y`. This means we're testing whether the average value of `Y` differs significantly across bioregions, after accounting for the influence of the continuous predictors.

This is how these different insights pertain to the model components:

- Main Effects: The coefficients for the main effects of `augMean`, `febSD`, and `augSD` represent the effect of each predictor when `bio` is at its reference level.
- Coefficients for `bio`: The coefficients for `bio` (e.g., $\beta_4 \text{bio}_{\text{B-ATZ}}$) represent the difference in the intercept for the corresponding level of `bio` compared to the reference level.
- Interaction Terms: The interaction terms allow the slopes of `augMean`, `febSD`, and `augSD` to vary across the different levels of `bio`. For example, $\beta_7(\text{augMean} \times \text{bio}_{\text{B-ATZ}})$ represents how the effect of `augMean` on `Y` changes when `bio` is `B–ATZ` compared to `AMP`.

5.8.1 State the Hypotheses

Overall hypothesis

I'll only state the overall hypothesis for this model as the expansion of the individual hypotheses for each predictor and interactions (all the β-coefficients in Equation 5.6) is quite voluminous.

The null is that there is no relationship between the response variable `Y` and the predictors (including their interactions):

- H_0 : $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = \beta_7 = \beta_8 = \beta_9 = \beta_{10} = \beta_{11} = \beta_{12} = \beta_{13} = \beta_{14} = \beta_{15} = 0$

The alternative is that at least one predictor or interaction term has a significant relationship with the response variable `Y`:

- H_A : At least one $\beta_i \neq 0$ for $i \in \{1, 2, ..., 15\}$

5.8.2 Fit the Model

In Section 5.6 I included the ECTZ seaweed flora in my analysis, but here I expand it to the full dataset. To assure myself that there is not a high degree of multicollinearity between the predictors, I have calculated the variance inflation factors (VIFs) for the full model (not shown). This allowed me to retain the same three predictors used in `mod1`, i.e. `augMean`, `febSD`, and `augSD`. This is the point of departure for `mod3`.

Now I fit the model with those three continuous predictors and their interactions with the categorical variable `bio`.

```
# Make a dataframe with only the relevant columns
sw_sub2 <- sw |>
  dplyr::select(Y, augMean, febSD, augSD, bio)

# Fit the multiple linear regression model with interaction terms
full_mod3 <- lm(Y ~ (augMean + febSD + augSD) * bio, data = sw_sub2)
full_mod3a <- lm(Y ~ augMean + febSD + augSD, data = sw_sub2)
```

```
null_mod3 <- lm(Y ~ 1, data = sw_sub2)
```

Model `full_mod3a` is similar to `full_mod3` but without the interaction terms. This will allow me to compare the two models and assess the importance of the interactions.

```
# Compare the models
anova(full_mod3, full_mod3a)
```

```
Analysis of Variance Table

Model 1: Y ~ (augMean + febSD + augSD) * bio
Model 2: Y ~ augMean + febSD + augSD
  Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
1    954 3.5603
2    966 5.6890 -12   -2.1288 47.535 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(full_mod3, full_mod3a)
```

```
           df       AIC
full_mod3  17 -2652.498
full_mod3a  5 -2221.852
```

The AIC value for `full_mod3` is lower than that of `full_mod3a`, indicating that including the interaction with `bio` is necessary. Likewise, the ANOVA test also shows that the full model (lower residual sum of squares) is significantly better than the reduced model.

I therefore use `full_mod3` going forward. This is a complex model so I have used the stepwise selection function, `stepAIC()`, to identify the most important predictors and interactions (code and output not shown). I hoped that this might have simplified the model somewhat, but the simplification I had hoped for did not materialise.

5.8.3 Interpret the Model

The model summary

The model summary provides a detailed look at the individual predictors and their interactions in the model.

```
# Summary of the model
summary(mod3) # full_mod3 renamed to mod3 during stepAIC()
```

```
Call:
lm(formula = Y ~ augMean + bio + augSD + febSD + augMean:bio +
    bio:augSD + bio:febSD, data = sw_sub2)

Residuals:
     Min       1Q   Median       3Q      Max
```

```
-0.15399 -0.03841 -0.01475  0.03464  0.24051

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.0299094  0.0062756   4.766 2.17e-06 ***
augMean           0.3441099  0.0158575  21.700  < 2e-16 ***
bioB-ATZ         -0.0459611  0.0242519  -1.895 0.058374 .
bioBMP            0.0160756  0.0100749   1.596 0.110906
bioECTZ          -0.0015444  0.0090275  -0.171 0.864197
augSD            -0.0059012  0.0034011  -1.735 0.083044 .
febSD            -0.0006481  0.0027954  -0.232 0.816706
augMean:bioB-ATZ -0.0461775  0.0874044  -0.528 0.597400
augMean:bioBMP   -0.2406297  0.0211404 -11.382  < 2e-16 ***
augMean:bioECTZ  -0.0607745  0.0189030  -3.215 0.001348 **
bioB-ATZ:augSD    0.0655983  0.0371033   1.768 0.077382 .
bioBMP:augSD      0.0410220  0.0114706   3.576 0.000366 ***
bioECTZ:augSD     0.0280513  0.0053752   5.219 2.21e-07 ***
bioB-ATZ:febSD    0.0409425  0.0818927   0.500 0.617223
bioBMP:febSD      0.0056433  0.0150126   0.376 0.707070
bioECTZ:febSD     0.0502867  0.0082266   6.113 1.43e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06109 on 954 degrees of freedom
Multiple R-squared:  0.7797,    Adjusted R-squared:  0.7762
F-statistic: 225.1 on 15 and 954 DF,  p-value: < 2.2e-16
```

The first thing to notice is that the model function has been rewritten in the forward selection process (but none of the variables were deemed insignificant and removed):

- Initial specification: `Y ~ (augMean + febSD + augSD) * bio`
- Specification after `stepAIC()`: `Y ~ augMean + bio + augSD + febSD + augMean:bio + bio:augSD + bio:febSD`

Functionally, these two are identical, but the order in which the terms are presented differs. Although this has affected the order in which the coefficients are presented in the summary output, the coefficients are the same. The coefficients are:

- `(Intercept)`: This is the estimated average value of `Y` when all predictor variables are zero and the observation is in the reference bioregion (`AMP`).
- Main Effects:
  - `augMean`: For every one-unit increase in `augMean`, `Y` increases by 0.3441, on average, assuming all other predictors are held constant. This effect is highly significant.
  - `augSD` and `febSD`: The main effects of these variables are not statistically significant, suggesting they might not have a direct impact on `Y` when averaged across all bioregions.
  - `bioB-ATZ`, `bioBMP`, `bioECTZ`: These coefficients represent the average difference in `Y` between each of these bioregions and the reference bioregion, when the continuous predictors are held at zero.
- Interaction Effects:
  - `augMean` interactions: The significant interactions of `augMean` with bioregion indicate

that the effect of `augMean` on `Y` varies across bioregions. Notably, the interaction with `bioBMP` has a strong, significant negative effect, suggesting that the positive effect of `augMean` is much weaker in this bioregion compared to the reference.
- `augSD` and `febSD` interactions: These interactions with bioregions are sometimes significant, providing good support for the alternative hypothesis that the effects of `augSD` and `febSD` on `Y` depend on the specific bioregion.

Since dummy coding returns differences with respect to reference levels, how would we calculate the actual coefficients for, say, `augMean`? Since there are significant interaction effects, we must consider the main effect of `augMean` in conjunction with bioregion.

For `bio = B-ATZ`:

- $\beta_{\text{augMean}} + \beta_{\text{augMean:bioB-ATZ}} = 0.3441099 + (-0.0461775) = 0.2979324$

For `bio = BMP`:

- $\beta_{\text{augMean}} + \beta_{\text{augMean:bioBMP}} = 0.3441099 + (-0.2406297) = 0.1034802$

For `bio = ECTZ`:

$\beta_{\text{augMean}} + \beta_{\text{augMean:bioECTZ}} = 0.3441099 + (-0.0607745) = 0.2833354$

The respective SEs for these coefficients can be calculated using the formula for the standard error of the sum of two variables. For example:

- $SE_{\text{augMean}} = \sqrt{SE^2_{\text{augMean}} + SE^2_{\text{augMean:bio}}}$

The ANOVA table

The ANOVA table assesses the overall significance of groups of predictors or the sequential addition of predictors to the model.

```
anova(mod3)
```

```
Analysis of Variance Table

Response: Y
             Df Sum Sq Mean Sq  F value    Pr(>F)
augMean       1 9.9900  9.9900 2676.902 < 2.2e-16 ***
bio           3 1.1901  0.3967  106.296 < 2.2e-16 ***
augSD         1 0.1393  0.1393   37.331 1.451e-09 ***
febSD         1 0.0053  0.0053    1.422    0.2334
augMean:bio   3 0.7910  0.2637   70.647 < 2.2e-16 ***
bio:augSD     3 0.3426  0.1142   30.602 < 2.2e-16 ***
bio:febSD     3 0.1401  0.0467   12.517 4.953e-08 ***
Residuals   954 3.5603  0.0037
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA table shows that the model is highly significant, with very low p-values throughout (< 0.0001). This indicates that the model as a whole is a good fit for the data.

5.8.4 Reporting

Here is what the reporting of the findings could look like in the Results section in your favourite journal.

Results

A multiple linear regression model examining the effects of the August climatological mean temperature (`augMean`), the August and February climatological SD of temperature (`augSD` and `febSD`, respectively), and the bioregion classification (`bio`) on the response variable, the Sørensen dissimilarity (`Y`), including their interaction terms, revealed several significant findings (Table 5.1). This model allows a separate regression slope for each predictor within the bioregions (Figure 5.7). The model explains a substantial portion of the variance in `Y` ($R^2 = 0.780$, adjusted $R^2 = 0.776$), and the overall model fit is highly significant ($F(15, 954) = 225.1$, $p < 0.0001$).

Table 5.1: Summary of the multiple linear regression model examining the effects of `augMean`, `augSD`, `febSD`, and `bio` on `Y`.

Coefficient	Estimate	Std. Error	t value	P-value
`(Intercept)`	0.0299	0.0063	4.766	< 0.0001 ***
`augMean`	0.3441	0.0159	21.700	< 0.0001 ***
`bioB-ATZ`	-0.0460	0.0243	-1.895	> 0.05
`bioBMP`	0.0161	0.0101	1.596	> 0.05
`bioECTZ`	-0.0015	0.0090	-0.171	> 0.05
`augSD`	-0.0059	0.0034	-1.735	> 0.05
`febSD`	-0.0006	0.0028	-0.232	> 0.05
`augMean:bioB-ATZ`	-0.0462	0.0874	-0.528	> 0.05
`augMean:bioBMP`	-0.2406	0.0211	-11.382	< 0.0005 ***
`augMean:bioECTZ`	-0.0608	0.0189	-3.215	< 0.005 **
`bioB-ATZ:augSD`	0.0656	0.0371	1.768	> 0.05
`bioBMP:augSD`	0.0410	0.0115	3.576	< 0.0005 ***
`bioECTZ:augSD`	0.0281	0.0054	5.219	< 0.0005 ***
`bioB-ATZ:febSD`	0.0409	0.0819	0.500	> 0.05
`bioBMP:febSD`	0.0056	0.0150	0.376	> 0.05
`bioECTZ:febSD`	0.0503	0.0082	6.113	< 0.0005 ***

The main effect of `augMean` was highly significant (Estimate = 0.3441, $p < 0.0001$), indicating a strong positive relationship with `Y`. The interaction term `augMean:bioBMP` (Estimate = -0.2406, $p < 0.0001$) and `augMean:bioECTZ` (Estimate = -0.0608, $p < 0.005$) were also significant, suggesting that the effect of `augMean` on `Y` varies significantly for `BMP` and `ECTZ` bioregions compared to the reference category (`AMP`). The `bioBMP` (Estimate = 0.0161, $p > 0.05$) and `bioECTZ` (Estimate = -0.0015, $p > 0.05$) terms were not significant, indicating no significant difference from `AMP`.

For `augSD`, the main effect was not significant (Estimate = -0.0059, $p > 0.05$). Significant interaction terms for `bioBMP:augSD` (Estimate = 0.0410, $p < 0.001$) and `bioECTZ:augSD` (Estimate = 0.0281, $p < 0.0001$) indicate that the effect of `augSD` on `Y` varies by bioregion.

The main effect of `febSD` was not significant (Estimate = -0.0006, $p > 0.05$), suggesting no direct relationship with `Y`. However, the interaction term `bioECTZ:febSD` (Estimate = 0.0503, $p = 0.0001$) was significant, indicating that the effect of `febSD` on `Y` differs for the `ECTZ` bioregion.

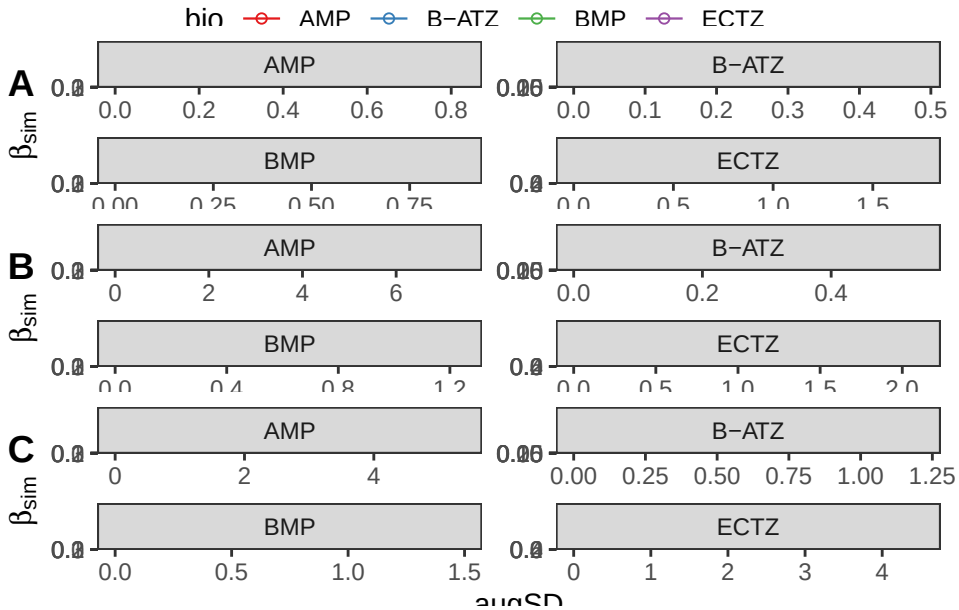

Figure 5.7: Individual linear regression fit to the variables `augMean`, `febSD`, and `augSD` for each bioregion as predictors of the seaweed species composition.

The ANOVA further highlights the overall significance of each predictor. `augMean` had a highly significant contribution to the model ($F = 2676.902$, $p < 0.0001$), as did `bio` ($F = 106.296$, $p < 0.0001$), and their interactions (`augMean:bio`, $F = 70.647$, $p < 0.0001$; `bio:augSD`, $F = 30.602$, $p < 0.0001$; `bio:febSD`, $F = 12.517$, $p = 4.953 \times 10^{-8}$). The main effect of `augSD` was also significant ($F = 37.331$, $p = 1.451 \times 10^{-9}$), while `febSD` did not significantly contribute to the model on its own ($F = 1.422$, $p = 0.2334$).

These findings suggest that the effects of `augMean`, `augSD`, and `febSD` on `Y` are influenced by the bioregional classification, with significant variations in the relationships depending on the specific bioregion.

5.9 Alternative Categorical Variable Coding Schemes (Contrasts)

Throughout the book, we have used dummy variable coding the specify the categorical variables in the multiple linear regression models. But, should dummy variable coding not be to your liking, there are other coding schemes that can be used to represent the categorical variables. These alternative coding schemes are known as contrasts. The choice of contrast coding can affect the interpretation of the regression coefficients.

I'll provide some synthetic data to illustrate a few different contrasts. The data consist of a continuous variable `x`, a categorical variable `cat_var` with four levels, and a response variable `y` that has some relationship with `x` and `cat_var`. I'll use dummy variable coding as the reference (haha!).

```
head(data)
```

```
           y           x cat_var
1  0.6667876 -0.56047565       B
2  1.3086873 -0.23017749       B
3  0.4496192  1.55870831       D
4  2.1326402  0.07050839       A
5 -2.8608771  0.12928774       D
6  0.1497346  1.71506499       D
```

Categorical variable coding (any scheme) only affects the interpretation of the categorical variable main effects and their interactions, so I'll not discuss the coefficient associated with the continuous variable x (the slope) in the model throughout the explanations offered below.

Dummy Variable Coding (Treatment Contrasts)

This is the most commonly used coding scheme, and `lm()`'s default. One level is the reference category (A) and the other levels are compared against it. Contrast matrices can be assigned and/or inspected using the `contrasts()` function. For the dummy coding, the reference level A will remain 0 and the other levels will be independently coded as 1 in three columns. You'll now understand why, when we have four levels within a categorical variable, we only need three dummy variables to represent them.

```
# Dummy coding (treatment coding) ... default
contrasts(data$cat_var)
```

```
  B C D
A 0 0 0
B 1 0 0
C 0 1 0
D 0 0 1
```

When we have four levels in a categorical variable, there are three dummy variable columns in the contrast matrix. The first row, consisting of all zeros (0, 0, 0), represents the reference level, which in this case is A. The other rows represent the different levels of the categorical variable, with a 1 in the respective column indicating that level. For example, level A is represented by (0, 0, 0), B by (1, 0, 0), C by (0, 1, 0), and D by (0, 0, 1). In the regression model, these contrasts are used to estimate the differences between each level and the reference level. Specifically, the first contrast column indicates that the coefficient for this column will represent the difference between the mean of the response variable for level B and the mean for the reference level A, holding all other variables constant. Similarly, the second and third columns represent the differences between levels C and A, and D and A, respectively. This coding allows for a straightforward interpretation of how each level of the categorical variable affects the response variable relative to the reference level.

```
model_dummy <- lm(y ~ x + cat_var, data = data)
summary(model_dummy)
```

```
Call:
lm(formula = y ~ x + cat_var, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-1.6615 -0.6297 -0.1494  0.4978  2.9305

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.8176     0.1635  17.232  < 2e-16 ***
x             1.8274     0.1040  17.572  < 2e-16 ***
cat_varB     -1.7201     0.2499  -6.883 6.24e-10 ***
cat_varC     -3.9056     0.2678 -14.586  < 2e-16 ***
cat_varD     -5.4880     0.2512 -21.850  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9246 on 95 degrees of freedom
Multiple R-squared:  0.887, Adjusted R-squared:  0.8822
F-statistic: 186.4 on 4 and 95 DF,  p-value: < 2.2e-16
```

The model summary shows that the coefficients for `cat_varB`, `cat_varC`, and `cat_varD` represent the differences in the mean of the response variable `y` between the reference category `A` and categories `B`, `C`, and `D`, respectively, while controlling for the effect of the continuous variable `x`.

Interpretation:

- `(Intercept)` (2.8176): The intercept represents the estimated mean value of the response (`y`) when `x` is zero and the categorical variable is at the reference level `A`. This is the baseline from which other categories are compared.
- `x` (1.8274): For each one-unit increase in `x`, `y` is expected to increase by 1.8274 units, holding the categorical variable constant. This effect is consistent across all levels of the categorical variable because the model does not have an interaction effect present.
- `cat_varB` (-1.7201): On average, the value of `y` for level `B` is 1.7201 units lower than that for the reference level `A`, when `x` is held constant. This corresponds to the (1, 0, 0) row in the contrast matrix.
- `cat_varC` (-3.9056): Similarly, on average, the value of `y` for level `C` is 3.9056 units lower than that for the reference level, when `x` is held constant. This corresponds to the (0, 1, 0) row in the contrast matrix.
- `cat_varD` (-5.4880): Lastly, on average, the value of `y` for level `D` is 5.4880 units lower compared to the reference , when `x` is held constant. This is row (0, 0, 1) row in the contrast matrix.

All these coefficients are highly significant ($p < 0.0001$), indicating strong evidence for differences between each category and the reference category `A`.

The model explains a large proportion of the variance in `y` (Adjusted R-squared: 0.8822), suggesting a good fit. The F-statistic (186.4) with a very low p-value (< 0.0001) indicates that the model as a whole is statistically significant.

If you want to change the reference level, you can use the `relevel()` function. For example, to change the reference level of `cat_var` variable to `C_2`, you can use:

```r
# Set "C" as the reference level for cat_var
data$cat_var <- relevel(data$cat_var, ref = "C")
contrasts(data$cat_var)
```

```
  A B D
C 0 0 0
A 1 0 0
B 0 1 0
D 0 0 1
```

This may be useful when you want to compare the other levels to a different reference level.

Effect Coding (Sum Contrasts)

This coding method compares the levels of a categorical variable to the overall mean of the dependent variable. The coefficients represent the difference between each level and the grand mean. Instead of using 0 and 1 as we did with dummy variable coding, effect coding uses -1, 0, and 1 to represent the different levels of the categorical variable.

```
# Reset the reference level to "A"
data <- data.frame(y, x, cat_var)

# Effect coding
contrasts(data$cat_var) <- contr.sum(4)
contrasts(data$cat_var)
```

```
  [,1] [,2] [,3]
A    1    0    0
B    0    1    0
C    0    0    1
D   -1   -1   -1
```

In effect coding (sum contrasts), each level of the categorical variable is compared to the overall mean rather than a specific reference category. This contrast matrix with four levels (A, B, C, D) and three columns can be interpreted as follows:

- Level `A` (1, 0, 0): The first row indicates that level `A` is included in the first contrast (`cat_var1`), which means the mean of level `A` is being compared to the overall mean. Since the other columns are zero, level `A` does not contribute to the other contrasts.
- Level `B` (0, 1, 0): The second row indicates that level `B` is included in the second contrast (`cat_var2`). The mean of level `B` is being compared to the overall mean, and it does not contribute to the other contrasts.
- Level `C` (0, 0, 1): The third row indicates that level `C` is included in the third contrast (`cat_var3`). The mean of level C is being compared to the overall mean, and it does not contribute to the other contrasts.
- Level `D` (-1, -1, -1): The fourth row is a balancing row, ensuring that the sum of the contrasts for each level equals zero. This indicates that level D is being compared to the overall mean indirectly by balancing the contributions of levels `A`, `B`, and `C`.

```
model_effect <- lm(y ~ x + cat_var, data = data)
summary(model_effect)
```

```
Call:
lm(formula = y ~ x + cat_var, data = data)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-1.6615 -0.6297 -0.1494  0.4978  2.9305

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03921    0.09452   0.415    0.679
x            1.82741    0.10400  17.572  < 2e-16 ***
cat_var1     2.77844    0.14968  18.563  < 2e-16 ***
cat_var2     1.05832    0.16329   6.481 4.04e-09 ***
cat_var3    -1.12720    0.17765  -6.345 7.53e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9246 on 95 degrees of freedom
Multiple R-squared:  0.887, Adjusted R-squared:  0.8822
F-statistic: 186.4 on 4 and 95 DF,  p-value: < 2.2e-16
```

Interpretation:

- `(Intercept)` 0.03921: The intercept represents the grand mean of the response variable (y). Since the intercept is not statistically significant ($p > 0.05$), it indicates that the overall mean is not significantly different from zero when considering the average effect of all levels of the categorical variable.
- x (1.82741): For each one-unit increase in (x), the response (y) increases by approximately 1.82741 units. This effect is highly significant ($p < 0.0001$).
- `cat_var1` (2.77844): Level `A` has a mean (y) that is 2.77844 units higher than the grand mean. This effect is highly significant ($p < 0.0001$).
- `cat_var2` (1.05832): Level `B` has a mean (y) that is 1.05832 units higher than the grand mean. This effect is also highly significant ($p < 0.0001$).
- `cat_var3` (-1.12720): Level `C` has a mean (y) that is 1.12720 units lower than the grand mean. This effect is highly significant (p < 0.0001).

All these coefficients are highly significant ($p < 0.0001$), indicating strong evidence for differences between each category and the overall mean of all levels.

The model explains a large proportion of the variance in y (Adjusted R-squared: 0.8822), suggesting a good fit. The F-statistic (186.4) with a very low p-value (< 0.0001) indicates that the model as a whole is statistically significant.

Helmert Coding

Helmert coding compares each level of a categorical variable to the mean of the subsequent levels. It is useful for testing ordered differences.

```r
# Helmert coding
contrasts(data$cat_var) <- contr.helmert(4)
contrasts(data$cat_var)
```

```
  [,1] [,2] [,3]
A   -1   -1   -1
```

```
B    1   -1   -1
C    0    2   -1
D    0    0    3
```

The contrast matrix for a categorical variable with four levels (A, B, C, D) and three columns can be interpreted as follows:

- Level A (-1, -1, -1): Level A is compared to the mean of levels B, C, and D. The negative values indicate that level A is being subtracted in these comparisons.
- Level B (1, -1, -1): Level B is compared to the mean of levels C and D. The positive value in the first column indicates that level B is being added in this comparison.
- Level C (0, 2, -1): Level C is compared to the mean of level D. The positive value in the second column indicates that level C is being added in this comparison, while the negative value in the third column is part of the comparison for subsequent levels.
- Level D (0, 0, 3): Level D is compared on its own in the final contrast. The positive value in the third column indicates that level D is being added in this comparison.

```r
model_helmert <- lm(y ~ x + cat_var, data = data)
summary(model_helmert)
```

```
Call:
lm(formula = y ~ x + cat_var, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6615 -0.6297 -0.1494  0.4978  2.9305

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03921    0.09452   0.415    0.679
x            1.82741    0.10400  17.572  < 2e-16 ***
cat_var1    -0.86006    0.12495  -6.883 6.24e-10 ***
cat_var2    -1.01519    0.08206 -12.371  < 2e-16 ***
cat_var3    -0.90319    0.05477 -16.491  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9246 on 95 degrees of freedom
Multiple R-squared:  0.887, Adjusted R-squared:  0.8822
F-statistic: 186.4 on 4 and 95 DF,  p-value: < 2.2e-16
```

Interpretation:

- (Intercept) (0.03921): The grand mean of y when x is zero.
- x (1.82741): For each unit increase in x , y increases by 1.82741 units.
- cat_var1 (-0.86006): The mean of level A is 0.86006 units lower than the combined mean of levels B, C, and D.
- cat_var2 (-1.01519): The mean of level B is 1.01519 units lower than the combined mean of levels C and D.
- cat_var3 (-0.90319): The mean of level C is 0.90319 units lower than the mean of level D.

The interpretation of the overall model remains more-or-less similar to before:

All these coefficients are highly significant ($p < 0.0001$), indicating strong evidence for differences between each level and the overall mean of all subsequent levels.

The model explains a large proportion of the variance in y (Adjusted R-squared: 0.8822), suggesting a good fit. The F-statistic (186.4) with a very low p-value (< 0.0001) indicates that the model as a whole is statistically significant.

5.10 Exercises

! Task G

Use the data loaded at the start of this chapter for this task.
In this task you will develop data analysis, undertake model building, and provide an interpretation of the findings. Your goal is to explore the species composition and assembly processes of the seaweed flora around the coast of South Africa. See Smit et al. (2017) for more information about the data and the analysis.

a. **Analysis:** Please develop multiple linear regression models for the seaweed species composition (β_{sim} and β_{sne}, i.e. columns called `Y1` and `Y2`, respectively) using the all the predictors in this dataset. At the end, the final model(s) that best describe(s) the species assembly processes operating along the South African coast should be presented. The final model may/may not contain all the predictors in the dataset, and it is your goal to justify the variable and model selection.
   - Accomplishing a) will require that you work through the whole model-building process as outlined in the chapter. This includes the following steps:
     - Data exploration and visualisation (EDA)
     - Model building (providing hypothesis statements, variable selection using VIF and forward selection, comparisons of nested models, justifications for model selection)
     - Model diagnostics
     - Explanation of `summary()` and `anova()` outputs
     - Producing the Results section
     - **[60%]**

b. **Interpretation:** Once you have arrived at the best model, discuss your findings in the light of the appropriate ecological hypotheses that explain the relationships between the predictors and the seaweed species composition. Include insights drawn from the analysis of $\beta_{sør}$ that I developed in this chapter, and also rely on the theory you have developed for the lecture material the class presented in Task A2.
   - Accomplishing b) is thus all about model interpretation and discussing the ecological relevance of the results.
   - **[40%]**

The format of this task is a Quarto file that will be converted to an HTML file. The HTML file will contain the graphs, all calculations, and the text sections. The task should be written up as a publication (i.e. use appropriate headings) using a journal style of your choice. Aside from this, there are no limitations.

Chapter 6

Generalised Linear Models (GLM)

6.1 Logistic Regression

A logistic regression model is used when the dependent variable is binary (e.g., 0 or 1, yes or no). The logistic regression model is expressed as:

$$\log\left(\frac{p}{1-p}\right) = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik} \tag{6.1}$$

Where:

- p is the probability of the dependent variable being 1,
- $X_{i1}, X_{i2}, \ldots, X_{ik}$ are the k predictor variables for the i-th observation,
- α is the intercept,
- $\beta_1, \beta_2, \ldots, \beta_k$ are the coefficients for the k predictor variables.

Chapter 7

Nonlinear Models

In This Chapter

- Nonlinear Regression

Elsewhere in the Book

- Simple Linear Regression
- Polynomial Regression
- Multiple Linear Regression
- Generalised Linear Models
- Generalised Additive 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 3.1, polynomial regressions provide a nonlinear relationship between the response and predictor variables (as seen in the regression line fit to the data, Figure 7.1 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 10; Figure 7.1 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 7.1 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 7.1 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 are well-understood.

A general formula for a nonlinear regression model is:

$$Y_i = f(X_i; \theta) + \epsilon_i \tag{7.1}$$

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:

$$Y_i = \alpha e^{\beta X_i} + \epsilon_i \tag{7.2}$$

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.

7.1 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 6 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 7.5.2 and Section 7.5.3.

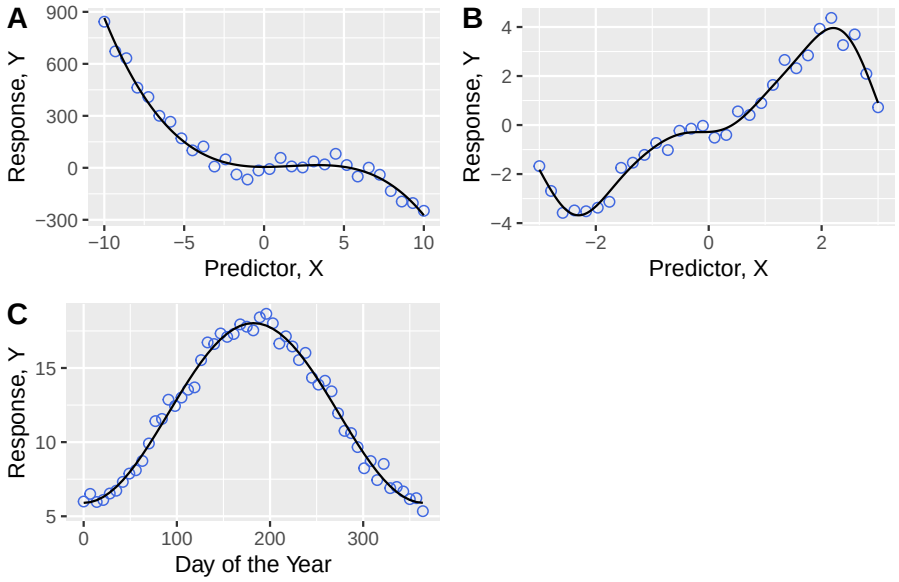

Figure 7.1: Nonlinear regression models fitted to simulated data. A) a cubic polynomial model, B) a GAM with a thin plate regression spline, and C) a NLS sine curve as a seasonal cycle.

7.2 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 7.2. 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.

Since you will often have to decide among polynomial regressions, nonlinear models, and GAMs, I'll outline some general guidelines to help you make an informed decision.

- **Linearity vs. Nonlinearity:** 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 7.2, 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.

- **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

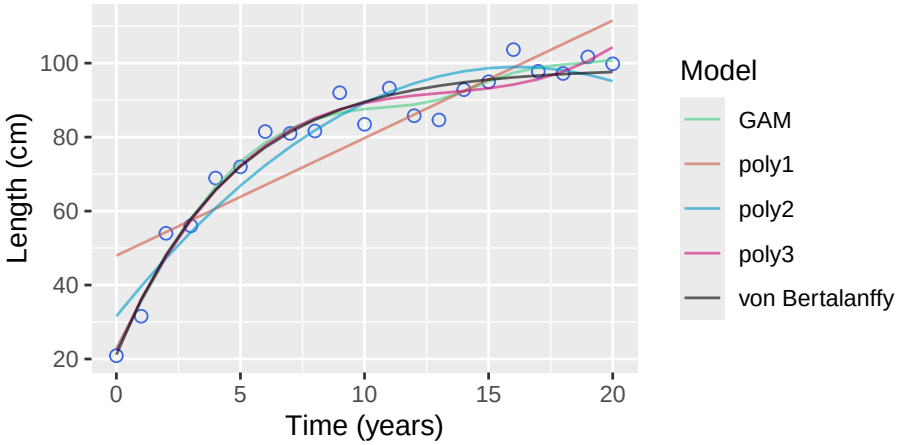

Figure 7.2: Plot of growth rate data fitted with a von Bertalanffy model, a first- (straight line), second- and third-order polynomial, and a GAM.

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 7.2, 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.

- **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 7.2, the von Bertalanffy model has a clear biological interpretation (see Section 7.6), 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.
- **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 7.2, 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.
- **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 7.2.

- **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 7.3), 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.
- **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 7.2.

7.3 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.

7.4 R Functions and Packages

7.4.1 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.
> *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)

[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.

- 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.

7.4.2 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).

7.5 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 7.1.

Table 7.1: 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 $[S]$
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 7.3) 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:

$$V_i = \frac{V_{max} \cdot [S_i]}{K_m + [S_i]} + \epsilon_i \tag{7.3}$$

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.

7.5.1 Hypothesis Testing and the Michaelis-Menten Model

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.

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:

$H_0 : V_{\text{max}} = V^*_{\text{max}}$ 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.

$H_a : V_{\text{max}} \neq V^*_{\text{max}}$ 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:

$H_0 : V_{\max}^A = V_{\max}^B$ and $K_m^A = K_m^B$

$H_a : V_{\max}^A \neq V_{\max}^B$ or $K_m^A \neq K_m^B$

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.

7.5.2 Multiple Flask Experiment

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 7.2, and a plot of V as a function of $[S]$ is shown in Figure 7.3.

Table 7.2: 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

In Figure 7.3, 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:

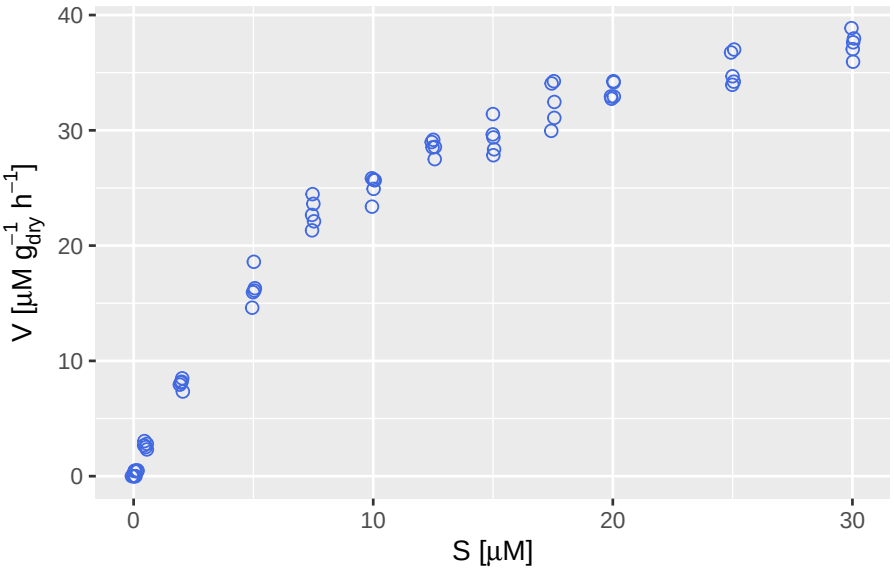

Figure 7.3: Plot of V as a function of $[S]$ for a multiple flask experiment involving seven replicate flask sets.

```
# 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.
- 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 7.4). 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 7.5.2 when comparing the effects of experimental treatments on kinetic parameters.

Is the Michaelis-Menten model a better fit than a linear model?

In Section 7.5.1, we pose a hypothesis that requires comparing a linear model to a Michaelis-Menten model fitted to the same data. Figure 7.4 indicates the nonlinear model indeed provides

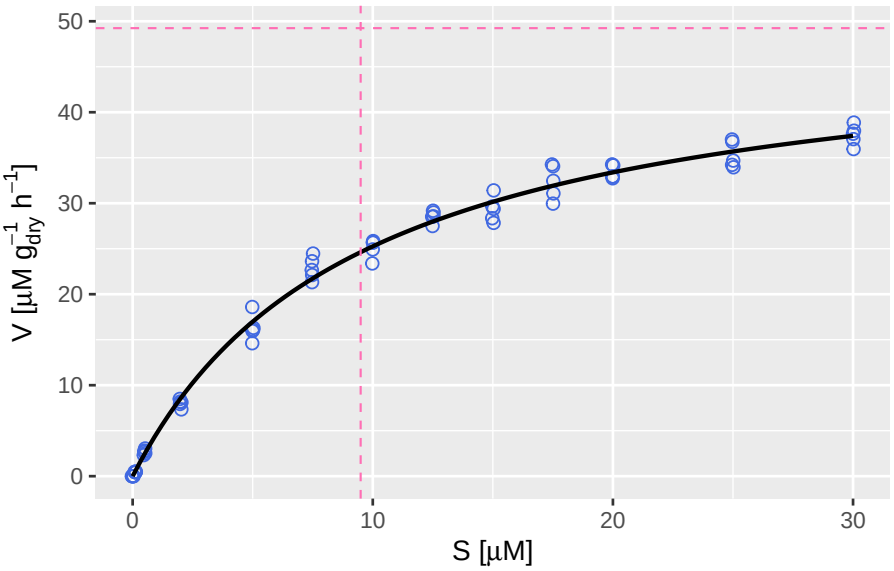

Figure 7.4: Plot of the Michaelis-Menten model fitted to the data in Figure 7.3. The vertical and horizontal dashed lines indicate the estimated K_m and V_{max} values, respectively.

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.

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 7.5.1. 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 7.3. Except for a new variable (treatment), the data are in all other respects identical to those in Section 7.5.2.

Table 7.3: 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

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 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:

```r
# 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:

```r
# 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
```

```r
# 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 7.6.

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 7.5).

```
# 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"]
```

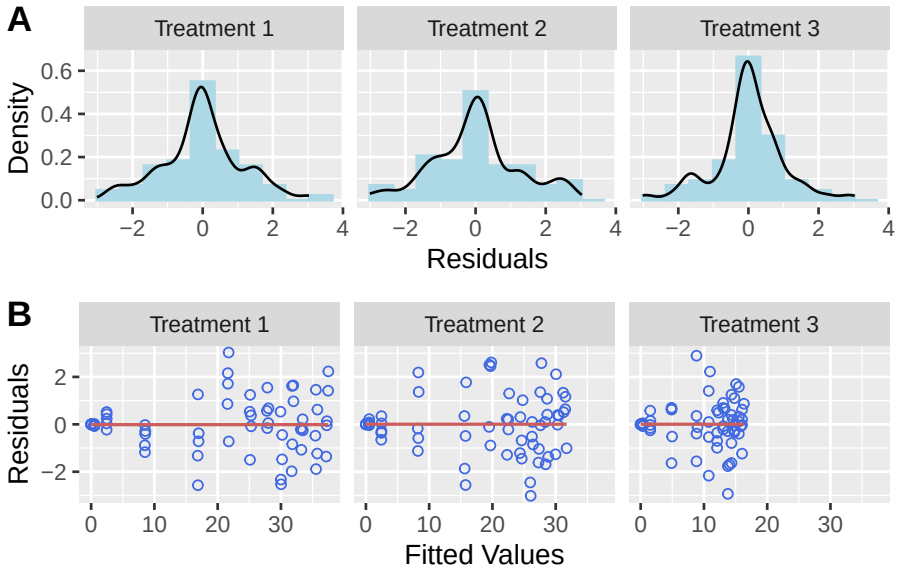

Figure 7.5: Histograms (A) of residuals and plots of residuals vs. the fitted values (B) for residuals for the three treatments in the multiple-flask experiment.

```
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 7.5 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 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 7.5.

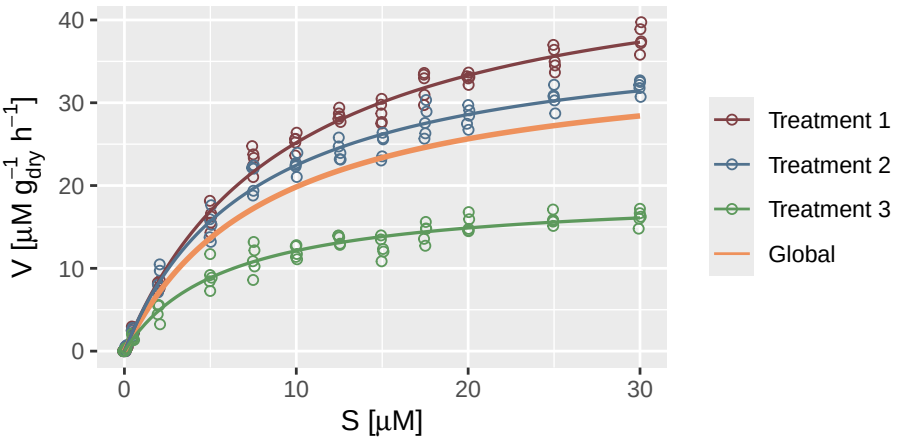

Figure 7.6: Plot of the Michaelis-Menten model fitted to the data in Table 7.3. Fits are provided for the separate models and the global model.

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 7.6). 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 to environmental perturbations.

7.5.3 The Perturbation Method (NLMM)

The data for this example is by Smit (2002). 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 µM 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 7.4). 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 7.5.1.

Table 7.4: 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

For the reasons discussed in Section 7.5, 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.

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
```

7.6 Example: The Growth Rate of Fish (NLMM)

The von Bertalanffy model (Equation 7.4) 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 age. By fitting the von Bertalanffy model to these length-at-age data, we can estimate growth parameters specific to the fish species.

The model is given by:

$$L(t) = L_\infty \left(1 - e^{-k(t-t_0)}\right) \tag{7.4}$$

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.

Table 7.5: The Atlantic Cod data set with 30 fish and 15 years of growth data (showing only the top and bottom three rows).

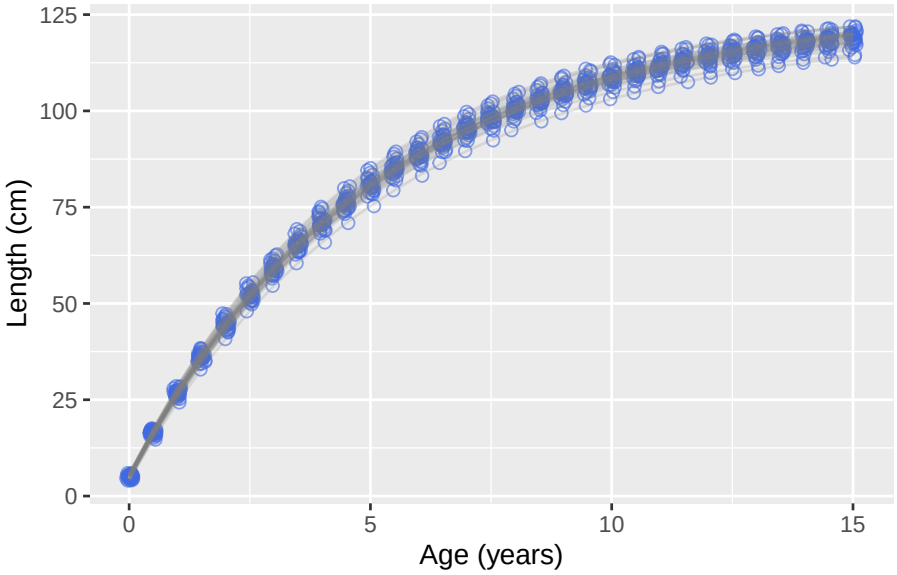

Figure 7.7: Plot of growth data measured in 30 Atlantic cod, *Gadus morua*.

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

A plot of the data is shown in Figure 7.7; here, each line represents the growth trajectory of an individual fish over time.

```
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)
```

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.

```
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:
```

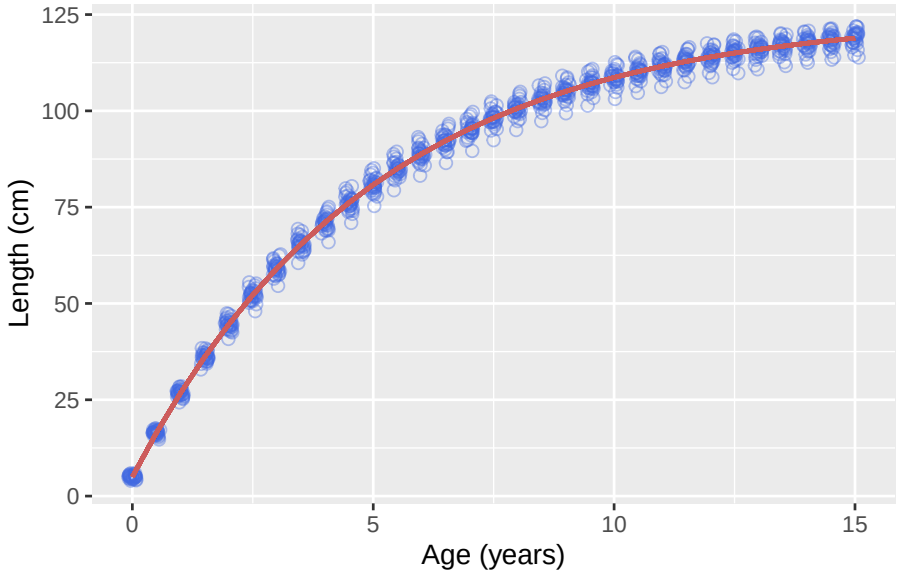

Figure 7.8: Fit of the von Bertalanffy model to experimental data obtained from 30 Atlantic Cod individuals.

```
   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
```

7.7 Scrathpad

7.7.1 To include in the article

- **Use Cases:** Mentioning use cases like trend analysis in time-series data or other applications where the relationship between variables might be expected to follow a polynomial form (e.g., growth rates, trajectories) can provide practical context.
- **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.
- **Homoscedasticity:**
- **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.

7.7.2 Contuinuing the MM model

Chapter 8

Regularisation Techniques

Regularisation techniques are invaluable when dealing with complex datasets or situations where traditional methods may fall short. They are used to enhance model stability, improve predictive performance, and increase interpretability, especially when working with multi-dimensional data in multiple linear regression models and multivariate analyses. Regularisation addresses several common challenges in statistical modelling: i) multicollinearity, ii) variable selection, iii) overfitting, and iv) model simplification.

Environmental datasets often contain many independent variables, and it is likely that only some of them are necessary to explain the phenomenon of interest. **Variable selection** is the process of identifying the most important predictors to include in a model. This can be achieved through the application of specialist, domain-specific knowledge, or through statistical or data-driven approaches. Regularisation is an example of the latter, as it can automatically identify the most relevant predictors on statistical grounds, serving as an alternative to traditional variable selection methods such as Variance Inflation Factor (VIF) and stepwise selection (see Section 5.6.4 and Section 5.6.5).

Overfitting occurs when a model 'explains' the noise in data together with the underlying pattern, which might happen when the model has too many predictors relative to the number of observations. This may also result when variable selection has not been sufficiently addressed. An overfit model performs exceptionally well on training data but fails to generalise to new, unseen data. Additionally, having too many predictors can lead to **multicollinearity** (see Section 5.6.4). This is a common issue in multiple linear regression when some of the many predictors included in the model are correlated. Multicollinearity can lead to inflated standard errors, unstable coefficients, and difficulty interpreting the model. Regularisation help manage multicollinearity by shrinking coefficient estimates or setting some to zero.

Effectiveness in variable selection, reducing multicollinearity, and mitigating overfitting all contribute to **model simplification**. Regularisation achieves similar outcomes by shrinking coefficient estimates or setting some to zero, making the model easier to understand, explain, and interpret.

In this chapter, we will discuss three common regularisation techniques: Lasso, Ridge, and Elastic Net Regression.

8.1 Ridge Regression (L2 Regularisation)

Ridge regression mathematically 'tames' the wildness of linear regression when faced with multicollinearity. It achieves this by adding a penalty term to the linear regression loss function—a term proportional to the square of the coefficients (the L2 norm). This penalty nudges the coefficients towards zero, effectively shrinking them without forcing them to be exactly zero.

In linear regression, the loss function is typically the Mean Squared Error (MSE), which is the average of the squared residuals (also known as the residual sum of squares, RSS). The optimisation objective is to minimise this loss function. In other words, the linear regression model aims to find the coefficients that minimise the average squared difference between the observed values and the predicted values. The RSS is expressed in Equation 8.1:

$$RSS(\beta) = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \tag{8.1}$$

And the MSE, which is the loss function to be minimised, is in Equation 8.2:

$$MSE(\beta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \tag{8.2}$$

Where:

- y_i is the observed value for the i-th observation.
- β_0 is the intercept.
- β_j are the coefficients for the predictors.
- x_{ij} is the value of the j-th predictor variable for the i-th observation.
- n is the number of observations.
- p is the number of predictors.

The notation $RSS(\beta)$ and $MSE(\beta)$ indicates that these are functions of the coefficients β. The optimisation objective for linear regression is to find the coefficients β_0 and β_1 to β_p that minimise the MSE. This can be expressed in Equation Equation 8.3:

$$\min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \right\} \tag{8.3}$$

Ridge regression extends the optimisation of the least squares regression by introducing a penalty term to the loss function. This penalty term is proportional to the square of the L2 norm of the coefficient vector, penalising large coefficient values. Ridge regression is specifically designed to handle multicollinearity and mitigate issues caused by correlated predictors. It also helps prevent overfitting when there are many predictors relative to the sample size, providing a more stable estimation process.

The penalty term is controlled by a hyperparameter[1] called lambda (λ) that determines the strength of the penalty. Larger values of λ lead to more shrinkage of the coefficients. When $\lambda = 0$, Ridge Regression is equivalent to ordinary least squares regression. As λ approaches infinity, all coefficients

[1] Hyperparameters are configuration settings that are external to your model and not learned from the data itself.

(except the intercept) approach zero. To find the optimal λ, you might have to use techniques like cross-validation. Cross-validation will be discussed later in Section Section 8.4.

The loss function in Ridge Regression is given by Equation 8.4:

$$L_{ridge}(\beta) = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\beta_j^2 \tag{8.4}$$

Where λ is the regularisation parameter controlling the penalty's strength. Note that typically, the intercept β_0 is not included in the penalty term.

In Equation Equation 8.4, $L_{ridge}(\beta)$ is the Ridge Regression loss function. This loss function includes the residual sum of squares (RSS) plus a penalty term $\lambda \sum_{j=1}^{p}\beta_j^2$. The optimisation objective in Ridge Regression is to find the values of the coefficients β_1 through β_p that minimise this penalised loss function, while also finding the optimal value for the intercept β_0.

Ridge regression introduces a bias-variance trade-off. By shrinking the coefficients, it introduces a slight bias, as the model's predictions may not perfectly match the training data. However, this bias is often offset by a significant reduction in variance. The reduced variance means the model's predictions are more stable and less sensitive to small changes in the input data. This trade-off often results in improved overall predictive performance, especially on new, unseen data.

So, Ridge Regression sacrifices a bit of bias (accuracy on the sample data) to gain a lot in terms of reduced variance (generalisation to new data). This is a typical example of the bias-variance trade-off in statistical modelling and machine learning, where we often find that a bit of bias can lead to a much more robust and reliable model.

Unlike some other regularisation methods, such as principal component regression, Ridge Regression maintains the interpretability of the coefficients in terms of their relationship with the outcome. It is also versatile and can be applied to various types of regression models, including linear and logistic regression.

8.2 Lasso Regression (L1 Regularisation)

Lasso (Least Absolute Shrinkage and Selection Operator) regression employs a different penalty term compared to Ridge Regression. Instead of squaring the coefficients, Lasso Regression takes their absolute values. The cost function in Lasso Regression is given in Equation Equation 8.5:

$$L_{lasso}(\beta) = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j| \tag{8.5}$$

In Equation Equation 8.5, $L_{lasso}(\beta)$ is the Lasso Regression loss function. It includes the residual sum of squares (RSS) plus a penalty term $\lambda \sum_{j=1}^{p}|\beta_j|$ (L1 norm). This penalty term is the sum of the absolute values of the coefficients, scaled by the regularisation parameter λ (similar to Ridge Regression). Lasso regression seeks the values of β_0 through β_p that minimise $L_{lasso}(\beta)$. As with Ridge Regression, the intercept β_0 is typically not included in the penalty term.

The strength of Lasso Regression lies in its ability to shrink some coefficients all the way to zero, effectively eliminating those variables from the model. This automatic variable selection makes

Lasso Regression well-suited for creating sparse models where only the most influential variables are retained. This simplification aids in interpretation and can enhance model performance by reducing noise and overfitting.

Lasso Regression still applies a degree of shrinkage for the coefficients that are not shrunk to zero. Shrinkage reduces their variance and provide more stable models that are less sensitive to fluctuations in the data. Similar to Ridge Regression, Lasso involves a trade-off between bias and variance. The shrinkage introduces a small bias but can greatly reduce variance and result in better overall predictions.

Lasso regression is useful when dealing with datasets that have a large number of potential predictor variables. It helps identify the most relevant predictors. The end results is a simpler and more interpretable model. If you suspect redundancy among your predictor variables, Lasso can prune them and retain only those that provide the best predictive value. As always, the optimal value for λ should be determined through techniques like cross-validation.

8.3 Elastic Net Regression

Elastic net regression is a hybrid regularisation technique that combines the penalties of Ridge and Lasso Regression. It tries to provide the advantages of both methods and mitigate their drawbacks.

Here, the penalty term is the weighted average of the L1 (Lasso) and L2 (Ridge) penalties. A mixing parameter called alpha (α) controls the weighting between the two penalties. When $\alpha = 0$, Elastic Net is equivalent to Ridge Regression and when $\alpha = 1$ it is equivalent to Lasso Regression. For values of α between 0 and 1, Elastic Net blends the properties of both methods and provides some flexibility to regularisation.

The cost function in Elastic Net Regression is given in Equation 8.6:

$$L_{e_net}(\beta) = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\left(\alpha\sum_{j=1}^{p}|\beta_j| + (1-\alpha)\sum_{j=1}^{p}\beta_j^2\right) \tag{8.6}$$

Where α is the mixing parameter, with $0 \leq \alpha \leq 1$.

In Equation 8.6 there is the familiar RSS plus the combined penalty term that is a weighted sum of the L1 and L2 norms. The objective of Elastic Net Regression is again to minimise $L_{e_net}(\beta)$ by seeking optimal values of β_1 through β_p.

Like the other regularisation techniques, Elastic Net is also used when you have highly correlated predictors. While Lasso Regression might arbitrarily select one variable from a group and ignore the rest, Elastic Net tends to select groups of correlated features together and so provide a more comprehensive understanding of variable importance. The flexibility of adjusting the α parameter allows you to fine-tune the regularisation to best suit your specific dataset and modelling goals. It balances variable selection (Lasso) and shrinkage (Ridge). Also, Elastic Net can outperform Lasso and Ridge Regression in terms of prediction accuracy when dealing with high-dimensional datasets where the number of predictors exceeds the number of observations.

Elastic net is a good option if you have a dataset with many potential predictor variables and suspect strong correlations among them. Use it when you are uncertain whether pure variable selection (Lasso) or pure shrinkage (Ridge) is the best approach. The challenge is that now we

also have to tune the α parameter in addition to the regularisation parameter λ. A caveat is that Elastic Net retains the interpretability of individual coefficients but the interpretation becomes slightly more nuanced due to the mixed penalty term. This requires a thoughtful approach to understanding the model outputs.

8.4 Cross-Validation

The values of the hyperparameters (λ or α) significantly affect the model's performance and generalisation ability and so it necessitates careful optimisation. The `cv.glmnet()` function (see Section 8.5) automates this process by performing both hyperparameter tuning[2] and cross-validation. It systematically evaluates different combinations of λ or α values across multiple subsets of our data, using cross-validation to estimate their out-of-sample performance. This allows for the selection of the hyperparameter combination that yields the best performance and thus avoids the risk of overfitting and improves model generalisation.

The most widely used cross-validation method is k-fold cross-validation. The dataset is divided into k equally sized subsets (specified by the user). The subsets are called 'folds'. The model is then trained k times, each time using $k - 1$ folds for training and the remaining fold for validation. It provides a robust estimate of model performance by utilising all data points for both training and validation. It balances computational cost and bias reduction. But, the choice of k can influence results, and there's a trade-off between bias and variance: lower k values may lead to higher bias but lower variance, whilst higher k values do the opposite.

The general approach taken in k-fold cross validation is that, for each combination of hyperparameter values, we:

1. Perform k-fold cross-validation on the training data.
2. Calculate the average performance metric (e.g., mean squared error) across all folds.
3. Select the hyperparameter values that produced the best average performance.

This ensures that the hyperparameters we select are robust and generalissable to unseen data, rather than being overly influenced by the peculiarities of a single training set.

K-fold cross-validation is the most frequently-used form of cross-validation, but several other types exist. Some of them are:

Leave-one-out cross-validation (LOOCV) is an extreme case of k-fold cross-validation where k equals the number of data points. This method trains the model on all but one data point and validates on the left-out point, repeating this process for each data point. LOOCV provides an nearly unbiased estimate of model performance but can be computationally expensive for large datasets. It's most often used for small datasets where maximising training data is important. The downside is that LOOCV can suffer from high variance, especially for noisy datasets.

Stratified cross-validation ensures each fold maintains the same proportion of samples for each class as in the complete dataset. It useful for imbalanced datasets or when dealing with categorical outcomes. By preserving the class distribution in each fold, stratified cross-validation provides a more representative evaluation of model performance across all classes. Implementing stratification can be complex for multi-class problems or continuous outcomes.

[2]The goal of hyperparameter tuning is to find the optimal combination of hyperparameters that leads to the best model performance on your specific dataset. This is done by systematically evaluating different hyperparameter values and selecting the combination that yields the best results.

Holdout validation is the simplest form of cross-validation. The dataset is split into a training set and a test set. Typically, about 70-80% of the data is used for training and the balance is reserved for testing. The model is trained on the training set and then evaluated on the held-out test set. It is computationally efficient and provides a quick estimate of model performance but it has several limitations. Firstly, because it doesn't make full use of the available data for training, it can be an issue for smaller datasets. Secondly, the results can be highly dependent on the particular split chosen, leading to high variance in performance estimates. This is especially true for smaller datasets or when the split doesn't represent the overall data distribution well. But holdout validation remains useful for large datasets or as a quick initial assessment before applying more complex cross-validation techniques.

The examples will show k-fold cross validation, but you can easily adapt the code to use other cross-validation methods.

8.5 R Function

In R, the **glmnet** package provides functions for fitting regularised linear models. The `cv.glmnet()` function performs cross-validated regularisation path selection for the Elastic Net, Lasso, and Ridge Regression models.

```
cv.glmnet(x, y, alpha = 1, lambda = NULL, nfolds = 10,
          standardize = TRUE)
```

The function takes the following arguments:

- `x`: A matrix of predictors.
- `y`: A matrix of response variables (but read the help file as this varies depending on the data type).
- `alpha`: The mixing parameter for the Elastic Net penalty. When `alpha = 0`, the model is a Ridge Regression. When `alpha = 1`, the model is a Lasso Regression. The default value is `alpha = 1`.
- `lambda`: A vector of regularisation parameters. The function fits a model for each value of `lambda` and selects the best one based on cross-validation. The default is `lambda = NULL`, which means the function will generate a sequence of `100` values between `10^-2` and `10^2`.
- `nfolds`: The number of folds in the cross-validation. The default is `nfolds = 10`.
- `standardize`: A logical value indicating whether the predictors should be standardised. The default is `standardize = TRUE`.

It is not clearly documented in the function's help file, but the 'glm' in the function name indicates that the function fits a generalised linear model. This implies 'gaussian,' 'binomial,' 'poisson,' 'multinomial,' 'cox,' and 'mgaussian' families are supported, which can be supplied via the `family` argument to the function. The 'net' part of the name indicates that the function fits an Elastic Net, thus allowing choose between Lasso and Ridge by setting `alpha` to 1 or 0 (or something in-between). The 'cv' part of the name indicates that the function performs cross-validation.

8.6 Example 1: Ridge Regression

The data I use here should be well-known by now. They are the same seaweed dataset used throughout Chapter 5. I will use Ridge Regression to predict the response variable `Y` using the predictors `annMean`, `augMean`, `augSD`, `febSD`, and `febRange`.

First, I will read in the data and prepare them in the format required by `cv.glmnet()`. This involves standardising the response variable and predictors and converting them to matrices. I specify the range of λ values to try and set up 10-fold cross-validation. I then fit the model and plot the results of the cross-validation.

```
# Ridge Regression with Cross-Validation

# Set seed for reproducibility
set.seed(123)

# Load necessary libraries
library(glmnet)
library(tidyverse)

# Read the data
sw <- read.csv("data/spp_df2.csv")

# Standardise the response variable and present as a matrix
y <- sw ▷
  select(Y) ▷
  scale(center = TRUE, scale = FALSE) ▷
  as.matrix()

# Provide the predictors as a matrix
X <- sw ▷
  select(-X, -dist, -bio, -Y, -Y1, -Y2) ▷
  as.matrix()

# Set up lambda sequence
lambdas_to_try <- 10 ^ seq(-3, 5, length.out = 100)

# Perform 10-fold cross-validation
ridge_cv <- cv.glmnet(X, y, alpha = 0, lambda = lambdas_to_try,
  standardize = TRUE, nfolds = 10)
```

```
# Plot cross-validation results (ggplot shown)
plot(ridge_cv)
```

Figure 8.1, generated from the `cv.glmnet()` object, illustrates the relationship between the regularisation parameter λ and the model's cross-validation performance. The y-axis represents the mean squared error (MSE) from cross-validation, whilst the x-axis shows the $log(\lambda)$ values tested. Red dots indicate the mean MSE for each λ, with error bars showing ±1 standard er-

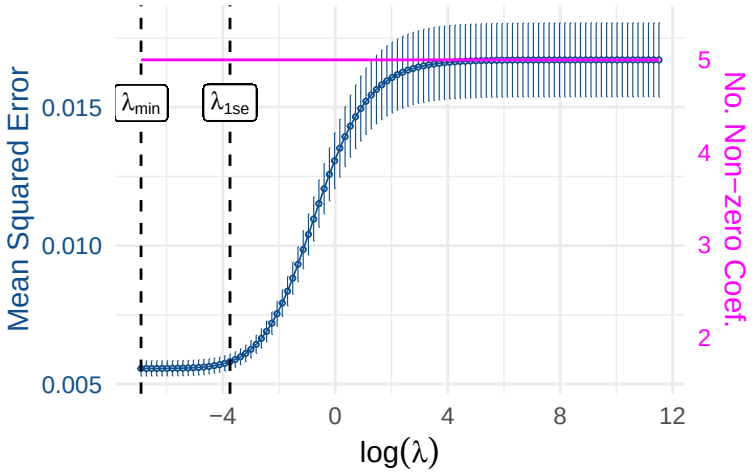

Figure 8.1: Cross-validation statistics for the Ridge Regression approach applied to the seaweed data.

ror. Two vertical dashed lines highlight important λ values: λ_{min}, which minimises the mean MSE, and λ_{1se}, the largest λ within one standard error of the minimum MSE. One may select the optimal λ using either the λ_{min} or the λ_{1se} rule, accessible via `cv.glmnet_object$lambda.min` and `cv.glmnet_object$lambda.1se`, respectively. To utilise the chosen λ, one refits the model using `glmnet()` and extract the coefficients.

For performance evaluation, one can calculate the sum of squared residuals (SSR) as the sum of squared differences between observed and predicted values, and the R-squared value as the square of the correlation between observed and predicted values, representing the proportion of variance in the dependent variable that is predictable from the independent variable(s).

The results show that the model explains 67.07% of the variance in the response variable:

```
# Fit models and calculate performance metrics
fit_model_and_calculate_metrics <- function(X, y, lambda) {
  model <- glmnet(X, y, alpha = 0, lambda = lambda,
                  standardize = TRUE)
  y_hat <- predict(model, X)
  ssr <- sum((y - y_hat) ^ 2)
  rsq <- cor(y, y_hat) ^ 2
  list(model = model, ssr = ssr, rsq = rsq)
}

# Best cross-validated lambda
lambda_cv <- ridge_cv$lambda.min
mod_cv <- fit_model_and_calculate_metrics(X, y, lambda_cv)

# Print results
mod_cv
```

```
$model

Call:  glmnet(x = X, y = y, alpha = 0, lambda = lambda, standardize = TRUE)

  Df  %Dev Lambda
1  5 67.06  0.001

$ssr
[1] 5.321994

$rsq
         s0
Y 0.6706681
```

As already indicated, an alternative to using `lambda.min` for selecting the optimal λ value is to use the 1 SE rule, which is contained in the attribute `lambda.1se`. This reduces the risk of overfitting as it tends to select a simpler model. We can use this value to refit the model and extract the coefficients, as before.

AIC and BIC can also be used to select suitable models. These information criteria penalise the model for the number of parameters used, providing a balance between model complexity and goodness of fit. The `calculate_ic()` function below calculates the AIC and BIC for a given model and returns the results in a list. We can then use this function to calculate the AIC and BIC for each model fit with each λ in `lambdas_to_try`:

```
# Calculate AIC and BIC
calculate_ic <- function(X, y, lambda) {
  model <- glmnet(X, y, alpha = 0, lambda = lambda,
                  standardize = TRUE)
  betas <- as.vector(coef(model)[-1])
  resid <- y - (scale(X) %*% betas)
  H <- scale(X) %*%
    solve(t(scale(X)) %*% scale(X) + lambda *
            diag(ncol(X))) %*% t(scale(X))
  df <- sum(diag(H))
  log_resid_ss <- log(sum(resid ^ 2))
  aic <- nrow(X) * log_resid_ss + 2 * df
  bic <- nrow(X) * log_resid_ss + log(nrow(X)) * df
  list(aic = aic, bic = bic)
}

ic_results <- map(lambdas_to_try, ~ calculate_ic(X, y, .x)) ▷
  transpose()
```

A plot of the change in the information criteria with $log(\lambda)$ is shown in Figure 8.2. The optimal λ values according to both AIC and BIC can then be used to refit the model and arrive at the coefficients of interest.

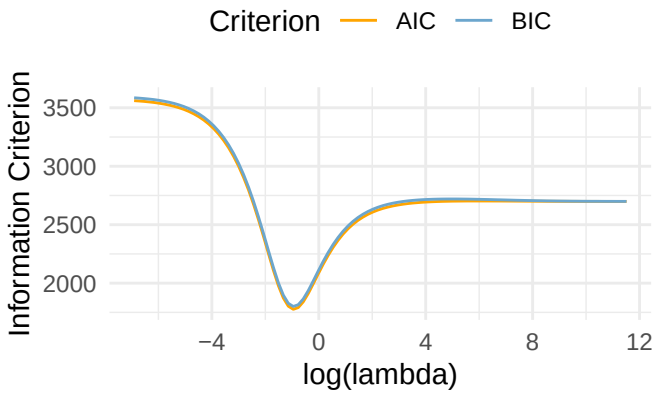

Figure 8.2: Plot of information criteria for best model fit selected through Ridge Regression.

```r
# Plot information criteria
plot_ic <- function(lambdas, ic_results) {
  df <- data.frame(lambda = log(lambdas),
                   aic = unlist(ic_results$aic),
                   bic = unlist(ic_results$bic))

  df_long <- pivot_longer(df, cols = c(aic, bic),
                          names_to = "criterion",
                          values_to = "value")

  ggplot(df_long, aes(x = lambda, y = value, color = criterion)) +
    geom_line() +
    scale_color_manual(values = c("aic" = "orange", "bic" = "skyblue3"),
                       labels = c("aic" = "AIC", "bic" = "BIC")) +
    labs(x = "log(lambda)",
         y = "Information Criterion", color = "Criterion") +
    theme_minimal() +
    theme(legend.position = "top",
          legend.direction = "horizontal",
          legend.box = "horizontal")
}

plot_ic(lambdas_to_try, ic_results)
```

Now we find the λ values that minimise the AIC and BIC, and refit the models using these values. It so happens that both AIC and BIC selects the same λ values:

```r
# Optimal lambdas according to both criteria
lambda_aic <- lambdas_to_try[which.min(ic_results$aic)]
lambda_bic <- lambdas_to_try[which.min(ic_results$bic)]
```

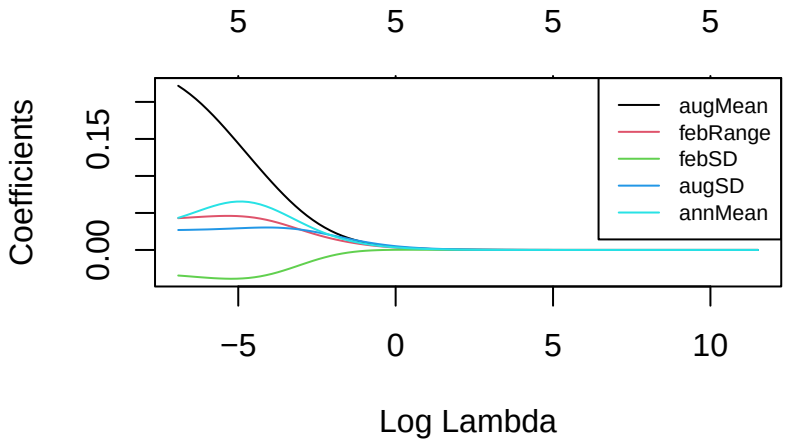

Figure 8.3: Plot of the Ridge Regression coefficients paths.

```
# Fit final models using the optimal lambdas
mod_aic <- fit_model_and_calculate_metrics(X, y, lambda_aic)
mod_bic <- fit_model_and_calculate_metrics(X, y, lambda_bic)
```

For interest sake, we may also produce a plot that traces the coefficients of the model as λ changes. This can help us understand how the coefficients shrink as λ increases, and which variables are most important in the model. The plot below shows the Ridge Regression coefficients path for each variable in the model (Figure 8.3).

```
# Plot the Ridge Regression coefficients path
res <- glmnet(X, y, alpha = 0, lambda = lambdas_to_try,
              standardize = FALSE)
plot(res, xvar = "lambda")
legend("topright", lwd = 1, col = 1:6,
       legend = colnames(X), cex = 0.7)
```

So, after having demonstrated the different methods for selecting the optimal λ value, we can now summarise the results:

```
[1] "CV Lambda: 0.001"
```

```
[1] "AIC Lambda: 0.3854"
```

```
[1] "BIC Lambda: 0.3854"
```

```
[1] "CV R-squared: 0.6707"
```

```
[1] "AIC R-squared: 0.6025"
```

```
[1] "BIC R-squared: 0.6025"
```

Now we can extract the coefficient produced from models selected via the AIC and CV methods.

```
  res_aic <- glmnet(X, y, alpha = 0, lambda = lambda_aic,
                    standardize = FALSE)
  res_aic
```

```
Call:  glmnet(x = X, y = y, alpha = 0, lambda = lambda_aic, standardize = FALSE)

  Df  %Dev Lambda
1  5 13.46 0.3854
```

```
  coef(res_aic)
```

```
6 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) -0.021327121
augMean      0.009856026
febRange     0.007118466
febSD       -0.001074341
augSD        0.010696102
annMean      0.008114467
```

```
  res_cv <- glmnet(X, y, alpha = 0, lambda = lambda_cv,
                   standardize = FALSE)
  res_cv
```

```
Call:  glmnet(x = X, y = y, alpha = 0, lambda = lambda_cv, standardize = FALSE)

  Df  %Dev Lambda
1  5 66.77  0.001
```

```
  coef(res_cv)
```

```
6 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) -0.12384440
augMean      0.22200994
febRange     0.04287655
febSD       -0.03446642
augSD        0.02699458
annMean      0.04324177
```

Ridge regression adds a penalty to the size of the coefficients, resulting in their shrinkage towards zero. This penalty affects all coefficients simultaneously. Notably, there is a difference in the model fit obtained using λ_{AIC} (which is larger) and λ_{min} (which is smaller). The former model explains 55.69% of the variance, compared to λ_{min}, which explains 63.37% of the variance.

Although shrinkage affects the absolute magnitude of the coefficients (they are biased estimates of the true relationships between the predictors and the response variable), the coefficients in Ridge Regression retain their general meaning—they still represent the change in the response

variable associated with a one-unit change in the predictor variable, holding other predictors constant. While the absolute values of the coefficients may be biased due to regularisation, the relative importance of the predictors can still be interpreted. The magnitude of the coefficients can indicate the relative influence of each predictor on the response variable, even if their exact values are reduced.

Importantly, the predictive ability of the model can improve with shrunk coefficients because Ridge Regression reduces overfitting and enhances the model's generalisability to new, unseen data. By stabilising the coefficient estimates, the model often achieves better performance on validation and test datasets, which is important should robust predictive analytics be the goal.

8.7 Example 2: Lasso Regression

Doing a Lasso Regression is easy. Simply change the `alpha` parameter to 1 in the `glmnet` function. The rest of the code remains the same. I'll show only the final output of this analysis to avoid repetition.

```
  # Print results
  mod_cv
```

```
$model

Call:  glmnet(x = X, y = y, alpha = 1, lambda = lambda, standardize = TRUE)

  Df %Dev Lambda
1  5   67  0.001

$ssr
[1] 5.332835

$rsq
         s0
Y 0.6701255
```

```
  coef(mod_cv$model)
```

```
6 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept) -0.12886019
augMean      0.26097296
febRange     0.03431981
febSD       -0.02497532
augSD        0.02441380
annMean      0.02021480
```

```
  # Print results
  print(paste("CV Lambda:", lambda_cv))
```

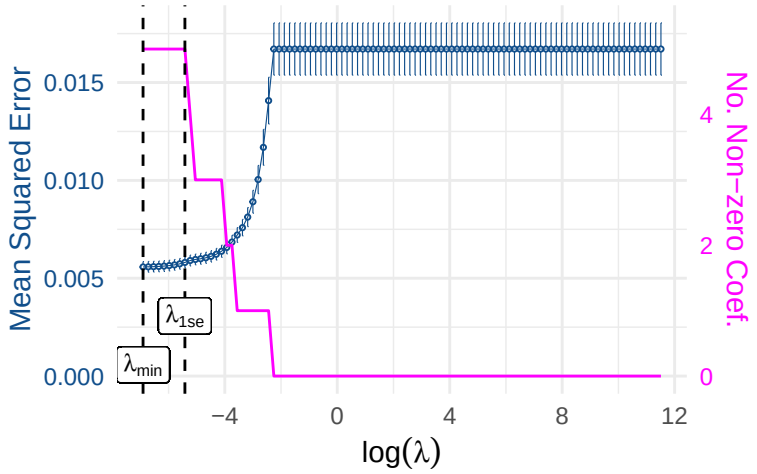

Figure 8.4: Cross-validation statistics for Lasso Regression applied to the seaweed data.

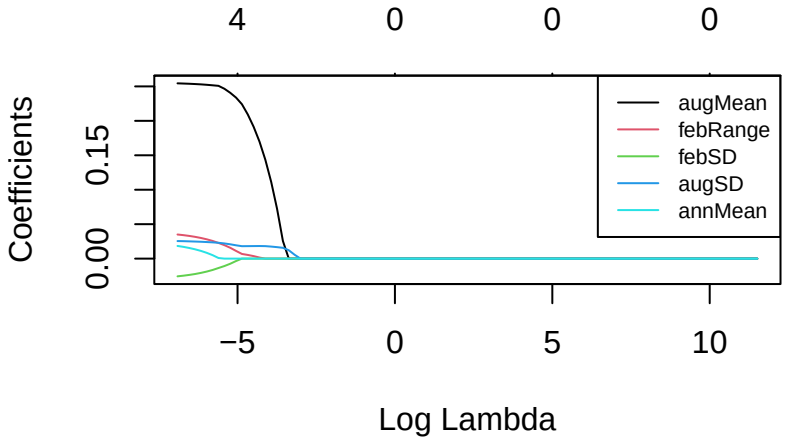

Figure 8.5: Plot of the Lasso Regression coefficients paths.

```
[1] "CV Lambda: 0.001"
```

```
print(paste("CV R-squared:", round(mod_cv$rsq, 4)))
```

```
[1] "CV R-squared: 0.6701"
```

Lasso regression incorporates an L1 penalty term in its cost function, which shrinks some coefficient estimates to exactly zero. By reducing certain coefficients to zero, Lasso effectively eliminates those predictors from the model, which achieves automatic variable selection:

- When λ is small, the penalty is minimal, and Lasso behaves similarly to ordinary least squares regression, retaining most coefficients.
- When λ is large, the penalty increases, causing more coefficients to shrink to zero. This results in a sparser model where only the most significant predictors have non-zero coefficients.

In our example (Figure 8.4), we see at λ_{min}, the number of non-zero coefficients is minimised—all five coefficients remain. At λ_{1se}, the number of non-zero coefficients decreases to four. Consequently, for higher values of λ, more predictors will have coefficients exactly equal to zero. This is also seen in Figure 8.4. In Figure 8.5 we can see that the first predictor to reach zero is `annMean`, then `febSD`, `febRange`, and so forth. The implication is that they are excluded from the model and the model is simplified. This leads to several benefits: reduced multicollinearity, improved interpretability, and better generalisation to new data.

Coefficients that remain non-zero after Lasso regularisation are considered more important predictors. Those remaining coefficients can be interpreted similarly to standard linear regression: as the expected change in the response variable for a one-unit change in the predictor, holding other predictors constant.

The λ parameter controls the amount of bias introduced. While Lasso can produce biased estimates, it reduces variance, often resulting in a model that performs better on new, unseen data. This trade-off enhances predictive accuracy but means that the exact coefficient values may not represent the true underlying relationships as closely as those in an unregularised model.

Despite regularisation, the relative magnitudes of the non-zero coefficients provide a glimpse into predictor importance. Larger absolute values of coefficients indicate stronger relationships with the response variable. The exact numerical values are biased, but ranking predictors by their coefficients still offers useful insight into their relative importance.

8.8 Example 3: Elastic Net Regression

In this last example we'll look at Elastic Net Regression, which combines the L1 and L2 penalties of Lasso and Ridge Regression. There are now two parameters to optimise: α and λ. The α parameter controls the mix between the L1 and L2 penalties, with $\alpha = 0$ behaving like Ridge Regression and $\alpha = 1$ behaving like Lasso Regression. For α values between 0 and 1, Elastic Net combines the strengths of both Lasso and Ridge Regression. Optimisation of α and λ is also done using cross-validation. In practise, the steps are:

1. Set up a grid of α values (from 0 to 1) and λ values to try.
2. Performs cross-validation for each combination of α and λ using `cv.glmnet()`.
3. Select the best α and λ combination based on the minimum mean cross-validated error.
4. Fit the final model using the best α and λ.
5. Calculate the performance metrics.
6. For the Elastic Net model with the best alphaCreate plots similar to those in the Ridge and Lasso examples.

```
# Define the range of alpha values to try
alphas_to_try <- seq(0, 1, by = 0.1)

# Define the range of lambda values to try
lambdas_to_try <- 10^seq(-3, 3, length.out = 100)

# Perform grid search with cross-validation
cv_results <- lapply(alphas_to_try, function(a) {
  cv.glmnet(X, y, alpha = a, lambda = lambdas_to_try,
```

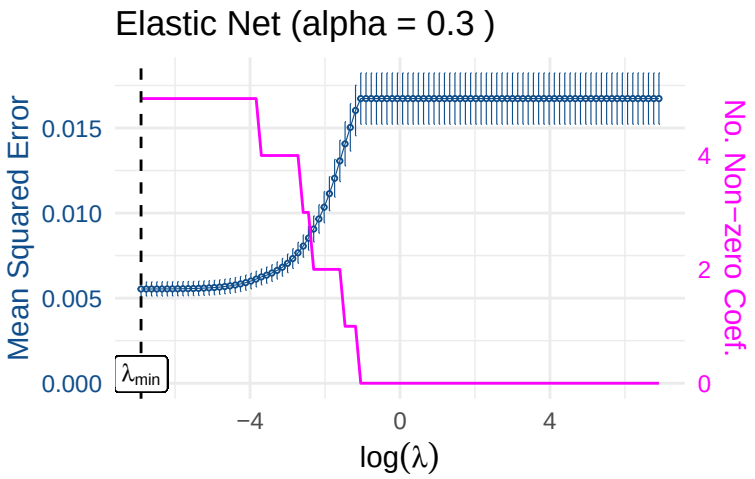

Figure 8.6: Cross-validation statistics for Elastic Net Regression applied to the seaweed data.

```
            standardize = TRUE, nfolds = 10)
})

# Find the best alpha and lambda
best_result <- which.min(sapply(cv_results, function(x) min(x$cvm)))
best_alpha <- alphas_to_try[best_result]
best_lambda <- cv_results[[best_result]]$lambda.min

# Fit the final model with the best parameters
final_model <- glmnet(X, y, alpha = best_alpha,
                      lambda = best_lambda,
                      standardize = TRUE)

# Calculate performance metrics
y_hat <- predict(final_model, X)
ssr <- sum((y - y_hat) ^ 2)
rsq <- cor(y, y_hat) ^ 2
```

```
[1] "Best Alpha: 0.3"
```

```
[1] "Best Lambda: 0.001"
```

```
[1] "R-squared: 0.6706"
```

The model coefficients are:

```
coef(cv_results[[best_result]])
```

```
6 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) -0.117063010
```

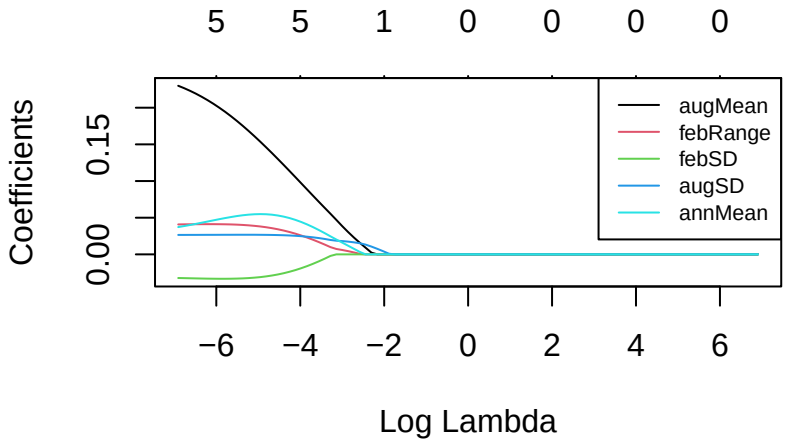

Figure 8.7: Plot of the Elastic Net Regression coefficients paths.

```
augMean      0.240447330
febRange     0.015404286
febSD       -0.007224065
augSD        0.014882111
annMean      0.026504736
```

The interpretation of coefficients in Elastic Net is a blend of Ridge and Lasso. Some coefficients may be shrunk to zero (feature selection), while others are shrunk but remain non-zero (magnitude reduction). The non-zero coefficients retain their general meaning with an emphasis on their relative importance.

8.9 Theory-Driven and Data-Driven Variable Selection

The choice between theory-driven and data- or statistics-driven variable selection represents an important consideration that can greatly influence model interpretation, its predictive power, and your value as an ecologist. This decision reflects a broader tension in scientific methodology between deductive and inductive reasoning. Each offers advantages and limitations that you should be aware of as an ecologist.

Theory-driven variable selection is core to the scientific method. It relies on *a priori* knowledge and established ecological theories (as far as they exist in ecology!) to guide your choice of predictors in a model. This aligns closely with the hypothetico-deductive method, where we formulate hypotheses based on existing knowledge and subsequently test these against the data we collect. The strength of this method lies in its interpretability. Models built on theoretical foundations often contribute directly to testing and refining ecological hypotheses. By focusing on variables with known or hypothesised relationships (with mechanisms often rooted in ecophysiological or ecological inquiries), the theory-driven hypothetico-deductive method should lead to more parsimonious models that are less prone to overfitting and more reflecting of reality.

Theory-driven selection is not without its drawbacks. It requires that we have a good grasp of the mechanism underlying our favourite ecological system. This is not always the case in complex systems where the underlying mechanisms are not well understood. Theory-driven selection can

then lead to the exclusion of important variables that were not initially hypothesised and it can limit the scope of the analysis and potentially overlook significant relationships in the data.

A naive young ecologist might place undue value on the notion that their hard work collecting diverse data and developing hypotheses should all be reflected in their final model. This can lead to confirmation bias, where one is more likely to select variables that support our hypotheses and ignore those that do not. This bias can compromise the objectivity of the model and lead to skewed results that do not accurately represent the underlying ecological processes.

Moreover, the insistence on including all variables that were initially considered important can result in overly complex models. Such models can be difficult to interpret and may suffer from overfitting, where the model captures noise rather than the true signal in the data. Overfitted models perform well on the data we collected but poorly on new, unseen data. The consequence is a loss of predictive power and generalisability.

Another weakness of theory-driven variable selection is that the reliance on existing theories or the novel, promising hypothsis of the day may lead us to overlook important but unexpected relationships in the data. In complex ecological systems, where our theoretical understanding may be incomplete, some variables could be missed entirely—these might in fact hold the key to the real cause of the ecological patterns we observe. This limitation becomes concerning when studying ecosystems or phenomena that are not well understood or are undergoing rapid changes, such as those affected by climate change or novel anthropogenic pressures.

On the other hand, data-driven approaches, including regularisation techniques, VIF, and forward model variable selection (Chapter 5), allow the data itself to guide variable selection. These methods are increasingly used in today's era of high-dimensional datasets common in modern ecological research. The primary advantage of data-driven selection lies in its potential for discovery—it can uncover unexpected relationships and generate new hypotheses, which is valuable in complex ecological systems where interactions may not be immediately apparent.

Data-driven methods are well-suited for handling the complexity often encountered in environmental and ecological datasets, where numerous potential predictors may co-occur and interact. They offer a degree of objectivity, reducing the potential for our personal biases in variable selection. But these approaches are not without risks. There's a danger of identifying relationships that are statistically significant but ecologically meaningless—we refer to this as spurious correlations (e.g. the belief that consuming carrots significantly improves our night vision). Moreover, models with many variables can present significant interpretability challenges, especially when complex interactions are present. This can make it difficult to extract meaningful (plausible) insights from the model and to communicate results to a broader audience.

In practice, the most robust approach to selecting which of the multitude of variables to include in our model often involves a thoughtful combination of theory-driven and data-driven methods. Well-trained ecologists should start with theory-driven variable selection to identify the core predictors based on established ecological principles. We could then employ regularisation techniques to explore additional variables and potential interactions, and use the results to refine our models and generate new hypotheses for future research.

This hybrid approach combines the strengths of both methods. It allows for rigorous hypothesis testing while remaining open to unanticipated and new insights from the data. In ecology, where systems are often characterised by complex, non-linear relationships and interactions that may vary across spatial and temporal scales, this two-pronged approach offers distinct benefits.

Consider how these methods complement theoretical knowledge. Use variable selection methods

as tools for prediction, and to assit generating new insights and hypotheses about ecosystems. The choice between theory-driven and data-driven variable selection is not a binary one, but rather a spectrum of approaches.

Part II

Non-Parametric Methods

Chapter 9

Testing Assumptions

Assumption tests are a fundamental component of any statistical analysis workflow. They ensure the validity, reliability, and robustness of statistical analyses. Although these tests test assumptions about parametric statistical methods, they are generally non-parametric, meaning they do not assume a specific probability distribution for the data.

9.1 Tests for Normality

In biological research, it is often important to determine whether a normal distribution adequately represents the underlying population distribution. This assessment is relevant when applying statistical procedures that rely on the assumption of normality, such as many of those discussed in earlier chapters. However, note that not all biological data conform to a normal distribution. In fact, many natural processes will result in non-normal data.

Assessing normality allows us to make informed decisions about appropriate statistical methods. If the data reasonably approximates a normal distribution, we can confidently apply parametric tests using probability calculations based on the normal curve. Conversely, if the data significantly deviates from normality, alternative non-parametric approaches may be more suitable.

Beyond simply validating statistical assumptions, examining the distribution of biological data can offer valuable insights into the underlying mechanisms and processes shaping the population. Identifying deviations from normality can challenge existing hypotheses, reveal hidden patterns, or suggest the influence of unanticipated factors. Therefore, normality tests are not only a technical requirement but they may also offer a tool for understanding the biological phenomena under investigation.

In this section, we will explore a range of graphical methods (e.g., histograms, Q-Q plots) and statistical tests (e.g., Shapiro-Wilk test, Kolmogorov-Smirnov test) to assess the goodness-of-fit of a normal distribution to our data.

Part III

Semi-Parametric Methods

Chapter 10

Generalised Additive Models

GAMs utilise a sum of smooth functions, each of which may depend on different subsets of the predictors. This additive structure, with smooth functions modelling the nonlinear effects of different predictor variables, allows GAMs to capture complex, nonlinear relationships without the need for a single, global parametric form. GAMs are useful in areas where the data exhibit complex patterns that are not easily described by traditional parametric or even non-parametric models.

Unlike polynomial regressions or specific nonlinear models that capture functional relationships with parameters directly linked to the system's mechanics, GAMs do not necessarily provide parameters that correspond to a mechanistic understanding. Instead, they offer flexibility and robustness in modelling, making them suitable for a wide range of applications where the relationship dynamics are complex and not well-defined by simpler models.

In GAMs, the smooth functions are typically represented using regression splines (Figure A). Splines are piecewise polynomial functions that are flexible and can approximate complex nonlinear relationships. In GAMs, the smooth functions are estimated using various types of regression splines, such as Thin Plate Regression Splines, Cubic Regression Splines, and P-Splines (B-Splines). These spline functions are used to model the nonlinear effects of the predictor variables in a flexible and data-driven manner, without assuming any specific parametric form. A GAM can be expressed as:

$$Y_i = \alpha + f_1(X_{i1}) + f_2(X_{i2}) + \ldots + f_p(X_{ip}) + \epsilon_i \tag{10.1}$$

Where:

- Y_i is the response variable for the i-th observation,
- α is the intercept,
- $f_j(X_{ij})$ are smooth functions of the predictor variables X_{ij} (for $j = 1, 2, \ldots, p$), and
- ϵ_i is the error term for the i-th observation.

The degree of smoothness of the smooth functions f_j is typically chosen based on the data and the modelling objectives, often using cross-validation or other model selection techniques.

Chapter 11

Summary

In summary, this book has no content whatsoever.

```
  1 + 1
```

```
[1] 2
```

References

Graham MH (2003) Confronting multicollinearity in ecological multiple regression. Ecology 84:2809–2815.

Smit A (2002) Nitrogen uptake by *Gracilaria gracilis* (Rhodophyta): Adaptations to a temporally variable nitrogen environment. Bot Mar 45:196–209.

Smit AJ, Bolton JJ, Anderson RJ (2017) Seaweeds in two oceans: Beta-diversity. Frontiers in Marine Science 4:404.

Underwood AJ (1997) Experiments in ecology: Their logical design and interpretation using analysis of variance. Cambridge university press

Appendix A

Appendix A

Index