The statistical analysis of non-linear longitudinal data in biomedical research using generalized additive models

Beyond repeated measures ANOVA and Linear Mixed Models

Ariel I. Mundo[1], John R. Tipton[2], and Timothy J. Muldoon[1]

[1]Department of Biomedical Engineering, University of Arkansas, Fayetteville, AR, USA
[2]Department of Mathematical Sciences, University of Arkansas, Fayetteville, AR, USA

Contents

1 Abstract

In biomedical research, the outcome of longitudinal studies has been traditionally analyzed using the *repeated measures analysis of variance* (rm-ANOVA) or more recently, a *linear mixed model* (LMEM). Although LMEMs are less restrictive that rm-ANOVA in terms of correlation and missing observations, both methodologies share an assumption of linearity in the measured response, which results in biased estimates and unreliable inference when they are used to analyze data where the trends are non-linear. In contrast, generalized additive models (GAMs), are a class of models that relax the linearity assumption and allow the data to determine the fit of the model while permitting missing observations and different correlation structures, thereby being an excellent choice to analyze non-linear longitudinal data. This paper summarizes the limitations of LMEMs and rm-ANOVA, presents the basic theory of GAMs, and demonstrates their implementation in R via the package *mgcv* using simulated data that follows longitudinal trends reported in biomedical literature. To promote reproducibility in biomedical research, the code and data used to generate this paper are available at:____________.

2 Background

Longitudinal studies are designed to repeatedly measure a variable of interest in a group (or groups) of subjects, with the intention of observing the evolution of effect across time rather than analyzing a single time point (e.g., a cross-sectional study). Biomedical research frequently uses longitudinal studies to analyze the evolution of a “treatment” effect across multiple time points; and in such studies the subjects of analysis range from animals (mice, rats, rabbits), to human patients, cells, or blood samples, among many others. Tumor response [1–4], antibody expression [5,6], and cell metabolism [7,8] are examples of the different situations where researchers have used longitudinal designs to study some physiological response. Because the frequency of the measurements in a longitudinal study is dependent on the biological phenomena of interest and the experimental design of the study, the frequency of such measurements can range from minute intervals to study a short-term response such as anesthesia effects in animals[9], to weekly measurements to analyze a mid-term response like the evolution of dermatitis symptoms in breast cancer patients [10], to monthly measurements to study a long-term response such as mouth opening following radiotherapy (RT) in neck cancer patients [11].

Traditionally, a “frequentist” or “classical” statistical paradigm is used in biomedical research to derive inferences from a longitudinal study. The frequentist paradigm regards probability as the limit of the expected outcome when an experiment is repeated a large number of times [12], and such view is applied to the analysis of longitudinal data by assuming a null hypothesis under a statistical model that is often an *analysis of variance over repeated measures* (repeated measures ANOVA or rm-ANOVA). The rm-ANOVA model makes three key assumptions regarding longitudinal data: 1) linearity of the response across time, 2) constant correlation across same-subject measurements, and 3) observations from each subject are obtained at all time points through the study (a condition also known as *complete observations*) [13,14].

The expected linear behavior of the response through time is a key requisite in rm-ANOVA [15]. This “linearity assumption” in rm-ANOVA implies that the model is misspecified when the data does not follow a linear trend, which results in unreliable inference. In biomedical research, non-linear trends are the norm rather than the exception in longitudinal studies. A particular example of this non-linear behavior in longitudinal data arises in measurements of tumor response to chemo and/or radiotherapy in preclinical and clinical settings [1,8,16]. These studies have shown that the collected signal does not follow a linear trend over time, and presents extreme variability at different time points, making the fit of rm-ANOVA model inconsistent with the observed variation. Therefore, when rm-ANOVA is used to draw inference of such highly-variable data the estimates are inevitably biased, because the model is only able to accommodate linear trends that are far from adequately representing the biological phenomenon of interest.

A *post hoc* analysis is the statistical test used in conjunction with rm-ANOVA to perform repeated comparisons to estimate a *p-value*, which in turn is used as a measure of significance. Although it is possible that a *post hoc* analysis of rm-ANOVA is able to find "significant" *p-values*($p<0.05$) from non-linear data, the validity of such metric is dependent on how adequate the model fits the data. In other words, *p-values* are valid only if the model and the data have good agreement; if that is not the case, a "Type III" error (known as "model misspecification") occurs[17]. For example, model misspecification will occur when a model that is only able to explain linear responses (such as rm-ANOVA) is fitted to data that follows a quadratic trend, thereby causing the resulting *p-values* and parameter estimates to be invalid [18].

Additionally, the *p-value* itself is highly variable, and multiple comparisons can inflate the false positivity rate (Type I error or α) [19,20], consequently biasing the conclusions of the study. Corrections exist to address the Type I error issue of multiple comparisons (such as Bonferroni [21]), but they in turn reduce statistical power (1-β)[22], and lead to increased Type II error (failing to reject the null hypothesis when the null hypothesis is false) [23,24]. Therefore, the tradeoff of *post hoc* comparisons in rm-ANOVA between Type I, II and III errors might be difficult to resolve in a biomedical longitudinal study where a delicate balance exists between statistical power and sample size.

On the other hand, the assumption of constant correlation in rm-ANOVA (often known as the *compound symmetry assumption*) is typically unreasonable because correlation between the measured responses often diminishes as the time interval between the observation increases [25]. Corrections can be made in rm-ANOVA in the absence of compound symmetry [26,27], but the effectiveness of the correction is limited by the size of the sample, the number of measurements[28], and group sizes [29]. In the case of biomedical research, where living subjects are frequently used, sample sizes are often not "large" due to ethical and budgetary reasons [30] which might cause the corrections for lack of compound symmetry to be ineffective.

Due to a variety of causes, the number of observations during a study can vary between all subjects. For example, in a clinical trial patients may voluntarily withdraw, whereas attrition due to injury or weight loss in preclinical animal studies is possible. It is even plausible that unexpected complications with equipment or supplies arise that prevent the researcher from collecting measurements at certain time points. In each of these missing data scenarios, the *complete observations* assumption of classical rm-ANOVA is violated. When incomplete observations occur, a rm-ANOVA model is fit by excluding all subjects with missing observations from the analysis [13]. This elimination of partially missing data from the analysis can result in increased costs if the desired statistical power is not met with the remaining observations, because it would be necessary to enroll more subjects. At the same time, if the excluded observations contain insightful information that is not used, their elimination from the analysis may limit the demonstration of significant differences between groups.

During the last decade, the biomedical community has started to recognize the limitations of rm-ANOVA in the analysis of longitudinal information. The recognition on the shortcomings of rm-ANOVA is exemplified by the use of linear mixed effects models (LMEMs) by certain groups to analyze longitudinal tumor response data [8,16]. Briefly, LMEMs incorporate *fixed effects*, which correspond to the levels of experimental factors in the study (e.g., the different drug regimens in a clinical trial), and *random effects*, which account for random variation within the population (e.g., the individual-level differences not due to treatment such as weight or age). When compared to the traditional rm-ANOVA, LMEMs are more flexible as they can accommodate missing observations for multiple subjects and allow different modeling strategies for the variability within each measure in every subject [15]. However, LMEMs impose restrictions in the distribution of the errors of the random effects, which need to be normally distributed and independent [13,31]. And even more importantly, LMEMs also expect a linear relationship between the response and time [15], making them unsuitable to analyze non-linear data.

As the rm-ANOVA and the more flexible LMEM approaches make overly restrictive assumptions regarding the linearity of the response, there is a need for biomedical researchers to explore the use of additional statistical tools that allow the data (and not an assumption in trend) to determine the trend of the fitted model, to enable appropriate inference.

In this regard, generalized additive models (GAMs) present an alternative approach to analyze longitudinal data. Although not frequently used by the biomedical community, these semi-parametric models are customarily used in other fields to analyze longitudinal data. Examples of the use of GAMs include the analysis of temporal variations in geochemical and palaeoecological data [32–34], health-environment interactions [35] and the dynamics of government in political science [36] . There are several advantages of GAMs over LMEMs and rm-ANOVA models: 1) GAMs can fit a more flexible class of smooth responses that enable the data to dictate the trend in the fit of the model, 2) they can model non-constant correlation between repeated measurements [37] and 3) can easily accommodate missing observations. Therefore, GAMs can provide a more flexible statistical approach to analyze non-linear biomedical longitudinal data than LMEMs and rm-ANOVA.

The current advances in programming languages designed for statistical analysis (specifically R), have eased the computational implementation of traditional models such as rm-ANOVA and more complex approaches such as LMEMs and GAMs. In particular, R[38] has an extensive collection of documentation and functions to fit GAMs in the package *mgcv* [37,39] that not only speed up the initial stages of the analysis but also enable the use of advanced modeling structures (e.g. hierarchical models, confidence interval comparisons) without requiring advanced programming skills from the user. At the same time, R has many tools that simplify data simulation, an emerging strategy used to test statistical models [28]. Data simulation methods allow the researcher to create and explore different alternatives for analysis without collecting information in the field, reducing the time window between experiment design and its implementation, and simulation can be also used for power calculations and study design questions.

This work provides biomedical researchers with a clear understanding of the theory and the practice of using GAMs to analyze longitudinal data using by focusing on four areas. First, the limitations of LMEMs and rm-ANOVA regarding linearity of response, constant correlation structures and missing observations is explained in detail. Second, the key theoretical elements of GAMs are presented using clear and simple mathematical notation while explaining the context and interpretation of the equations. Third, using simulated data that reproduces patterns in previously reported studies [16] we illustrate the type of non-linear longitudinal data that often occurs in biomedical research. The simulated data experiments highlight the differences in inference between rm-ANOVA, LMEMs and GAMs on data similar to what is commonly observed in biomedical studies. Finally, reproducibility is emphasized by providing the code to generate the simulated data and the implementation of different models in R, in conjunction with a step-by-step guide demonstrating how to fit models of increasing complexity.

In summary, this work will allow biomedical researchers to identify when the use of GAMs instead of rm-ANOVA or LMEMs is appropriate to analyze longitudinal data, and provide guidance on the implementation of these models by improving the standards for reproducibility in biomedical research.

3 Challenges presented by longitudinal studies

3.1 The repeated measures ANOVA

The *repeated measures analysis of variance* (rm-ANOVA) is the standard statistical analysis for longitudinal data in biomedical research. This statistical methodology requires certain assumptions for the model to be valid. From a practical view, the assumptions can be divided in three areas: 1) linear relationship between covariates and response, 2) a constant correlation between measurements, and, 3) complete observations for all subjects. Each one of these assumptions is discussed below.

3.2 Linear relationship

3.2.1 The repeated measures ANOVA case

In a longitudinal biomedical study, two or more groups of subjects (e.g., human subject, mice, samples) are subject to different treatments (e.g., a "treatment" group receives a novel drug or intervention vs. a "control" group that receives a placebo), and measurements from each subject within each group are collected at specific time points. The collected response is modeled with *fixed* components. The *fixed* component can be understood as a constant value in the response which the researcher is interested in measuring, i.e., the average effect of the novel drug/intervention in the "treatment" group.

Mathematically speaking, a rm-ANOVA model with an interaction can be written as:

$$y_{ijt} = \beta_0 + \beta_1 \times time_t + \beta_2 \times treatment_j + \beta_3 \times time_t \times treatment_j + \varepsilon_{ijt} \tag{1}$$

In this model y_{ijt} is the response for subject i, in treatment group j at time t, which can be decomposed in a mean value β_0, *fixed effects* of time ($time_t$), treatment ($treatment_j$) and their interaction $time_t * treatment_j$ which have linear slopes given by β_1, β_2 and β_3, respectively. Independent errors ε_{tij} represent random variation not explained by the *fixed* effects, and are assumed to be $\sim N(0, \sigma^2)$ (independently normally distributed with mean zero and variance σ^2_μ). In a biomedical research context, suppose two treatments groups are used in a study (e.g., "placebo" vs. "novel drug" or "saline" vs. "chemotherapy"). Then, the group terms in Equation (1) can be written as below with $treatment_j = 0$ representing the first treatment group (Group A) and $treatment_j = 1$ representing the second treatment group (Group B). The linear models then can be expressed as

$$y_{ijt} = \begin{cases} \beta_0 + \beta_1 \times time_t + \varepsilon_{ijt} & \text{if Group A} \\ \beta_0 + \beta_2 + \beta_1 \times time_t + \beta_3 \times time_t + \varepsilon_{ijt} & \text{if Group B} \end{cases} \tag{2}$$

To further simplify the expression, substitute $\widetilde{\beta_0} = \beta_0 + \beta_2$ and $\widetilde{\beta_1} = \beta_1 + \beta_3$ in the equation for Group B. This substitution allows for a different intercept and slope for Groups A and B. The model is then written as

$$y_{ijt} = \begin{cases} \beta_0 + \beta_1 \times time_t + \varepsilon_{ijt} & \text{if Group A} \\ \widetilde{\beta_0} + \widetilde{\beta_1} \times time_t + \varepsilon_{ijt} & \text{if Group B} \end{cases} \tag{3}$$

Presenting the model in this manner makes clear that when treating different groups, an rm-ANOVA model is able to accommodate non-parallel lines in each case (different intercepts and slopes per group). In other words, the rm-ANOVA model "expects" a linear relationship between the covariates and the response, this means that either presented as Equation (1), Equation (2) or Equation (3), an rm-ANOVA model is only able to accommodate linear patterns in the data. If the data show non-linear behavior, the rm-ANOVA model will approximate this behavior with non-parallel lines.

3.2.2 The Linear Mixed Model Case

A linear mixed model (LMEM) is a class of statistical model that incorporates *fixed effects* to model the relationship between the covariates and the response, and *random effects* to model subject variability that is not the primary focus of the study but that might be important to distinguish [15,40]. A LMEM with interaction between time and treatment for a longitudinal study can be written as:

$$y_{ijt} = \beta_0 + \beta_1 \times time_t + \beta_2 \times treatment_j + \beta_3 \times time_t \times treatment_j + \mu_{ij} + \varepsilon_{ijt} \tag{4}$$

When Equation (1) and Equation (4) are compared, it is easily noticeable that LMEM and rm-ANOVA have the same construction regarding the *fixed effects* of time and treatment, but that the LMEM incorporates an additional source of variation (the term μ_{ij}). This term μ_{ij} is the one that corresponds to the *random effect*, accounting for variability in each subject within each group. The *random* component can also be understood as used to model some "noise" in the response, but that is intended to be analyzed and disentangled from the "global noise" term ε_{ijt} from Equation (1).

For example, if the blood concentration of the drug is measured in certain subjects in the early hours of the morning while other subjects are measured in the afternoon, it is possible that the difference in the collection time introduces some "noise" in the data. As the name suggests, this "random" variability needs to be modeled as a variable rather than as a constant value. The *random effect* μ_{ij} in Equation (4) is assumed to be $\mu_{ij} \sim N(0, \sigma^2_\mu)$. In essence,the *random effect* in a LMEM enables to fit models with different slopes at the subject-level[15]. However,the expected linear relationship of the covariates and the response in Equation (1) and in Equation (4) is essentially the same, representing a major limitation of LMEMs to fit a non-linear response.

3.3 Covariance in rm-ANOVA and LMEMs

In a longitudinal study there is an expected *covariance* between repeated measurements on the same subject, and because repeated measures occur in the subjects within each group, there is a *covariance* between measurements at each time point within each group. The *covariance matrix* (also known as the variance-covariance matrix) is a matrix that captures the variation between and within subjects in a longitudinal study[41] (For an in-depth analysis of the covariance matrix see [40,42]).

In the case of an rm-ANOVA analysis, it is typically assumed that the covariance matrix has a specific construction known as *compound symmetry* (also known as "sphericity" or "circularity"). Under this assumption, the between-subject variance and within-subject correlation are constant across time [26,42,43]. However, it has been shown that this condition is frequently not justified because the correlation between measurements tends to change over time [44]; and it is higher between consecutive measurements [13,25]. Although corrections can be made (such as Huyhn-Feldt or Greenhouse-Geisser)[26,27] the effectiveness of each correction is limited because it depends on the size of the sample,the number of repeated measurements[28], and they are not robust if the group sizes are unbalanced [29]. Because biomedical longitudinal studies are often limited in sample size and can have an imbalanced design, the corrections required to use an rm-ANOVA model may not be able to provide a reasonable adjustment that makes the model valid.

In the case of LMEMs, one key advantage over rm-ANOVA is that they allow different structures for the variance-covariance matrix including exponential, autoregressive of order 1, rational quadratic and others [15]. Nevertheless, the analysis required to determine an appropriate variance-covariance structure for the data can be a challenging process by itself. Overall, the spherical assumption for rm-ANOVA may not capture the natural variations of the correlation in the data, and can bias the inferences from the analysis.

3.4 Missing observations

Missing observations are an issue that arises frequently in longitudinal studies. In biomedical research, this situation can be caused by reasons beyond the control of the investigator [45]. Dropout from patients and attrition or injury in animals are among the reasons for missing observations. Statistically, missing information can be classified as *missing at random* (MAR), *missing completely at random* (MCAR), and *missing not at random* (MNAR) [42]. In a MAR scenario, the pattern of the missing information is related to some variable in the data, but it is not related to the variable of interest [46]. If the data are MCAR, this means that the missingness is completely unrelated to the collected information [47], and in the case of MNAR the missing values are dependent on their value. An rm-ANOVA model assumes complete observations for all subjects, and therefore subjects with one or more missing observations are excluded from the analysis. This is inconvenient because the remaining subjects might not accurately represent the population, and statistical power is affected by this reduction in sample size [48].

In the case of LMEMs, inferences from the model are valid when missing observations in the data exist that are MAR or MCAR [40]. For example, if attrition occurs in all mice that had lower weights at the beginning of a chemotherapy response study, the missing data can be considered MAR because the missigness is unrelated to other variables of interest.

This section has presented the assumptions for analyzing longitudinal data using rm-ANOVA and LMEMs and compared their differences regarding linearity, the covariance matrix and missing data. In particular, LMEMs offer a more robust and flexible approach than rm-ANOVA and if the data follows a linear trend, they provide an excellent choice to derive inferences from a repeated measures study. However, when the data presents high a non-linear behavior, LMEMs and rm-ANOVA fail to capture the trend of the data. To better convey the issues of linearity and correlation in linear models fitted to non-linear data, simulation is used in the next section.

3.5 What does an rm-ANOVA fit looks like? A visual representation using simulated data

To demonstrate the limitations of rm-ANOVA an LMEMs for non-linear longitudinal data, this section presents a simulation experiment of a normally distributed response of two groups of 10 subjects each. An rm-ANOVA model (Equation (1)) is fitted to each group, using R[38] and the package *nlme*[49]. Briefly, two cases for the mean responses for each group are considered: in the first case, the mean response in each group is a linear function with different intercepts and slopes; a negative slope is used for Group 1 and a positive slope is used for Group 2 (Figure 1, A). In the second case, a second-degree polynomial (quadratic) function is used for the mean response per group: the quadratic function is concave down for Group 1 and it is concave up for Group 2 (Figure 1, C). In both the linear and quadratic simulated data, the groups start with the same mean value at the first time point. This is intentional in order to simulate the expected temporal evolution of some physiological quantity.

Specifically, the rationale for the chosen linear and quadratic functions is the likelihood that a measured response in two treatment groups is similar in the initial phase of the study, but as therapy progresses a divergence in the trend of the response indicates a treatment effect. In other words, Group 1 can be thought as a “Control” group and Group 2 as a “Treatment” group. From the mean response per group (linear or quadratic), the variability or “error” of individual responses within each group is simulated using a covariance matrix with compound symmetry (constant variance across time). Thus, the response per subject in both the linear and quadratic simulation corresponds to the mean response per group plus the error (Figure 1 B,D). A more comprehensive exploration of the fit of rm-ANOVA for linear and non-linear longitudinal appears in Figure 5 and Figure 6 in the Appendix, where simulation with compound symmetry and independent errors (errors generated from a normal distribution that are not constant over time) and the plot of simulated errors, and fitted parameters in presented.

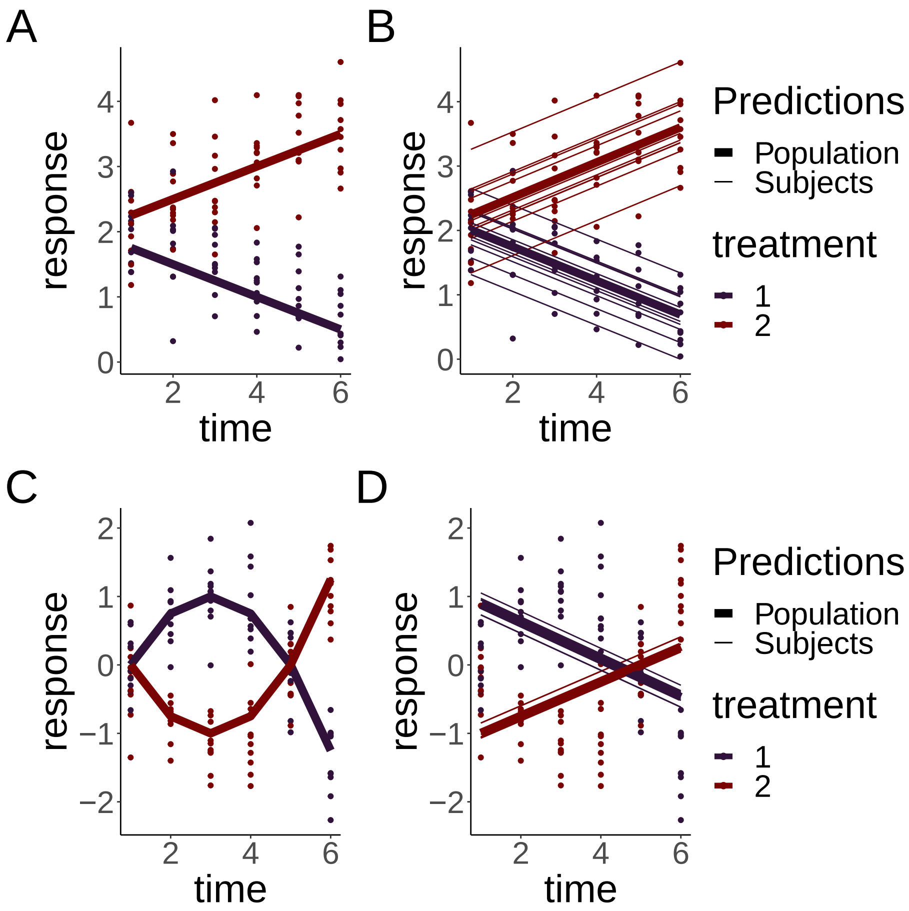


Figure 1: Simulated linear responses from two groups with correlated (top row) or independent (bottom row) errors using a rm-ANOVA model. A, C:Simulated data with known mean response (linear or quadratic, thin lines) and individual responses (points) showing the dispersion of the data. B,D: Estimates from the rm-ANOVA model for the mean group response (linear of quadratic). Thick lines are the predicted mean response per group, thin lines are the random effects for each subject and points represent the original raw data. The rm-ANOVA model not only fails to pick the trend of the quadratic data but it also incorrectly estimates the initial conditions.

The simulation shows that the fit produced by the rm-ANOVA model is good for linear data, as the predictions for the mean response are reasonably close to the “truth” of the simulated data (Figure 1,B). When the linearity and compound symmetry assumptions are met, the model approximates well the individual trends and the mean trends by group.

However, consider the case when the data follows a non-linear trend, such as the simulated data in Figure 1, C. Here, the mean response per group was simulated using a quadratic function but errors, individual responses and the rm-ANOVA model were produced in the same manner as in (Figure 1 A and B) . The mean response in the simulated data with quadratic behavior is changing in each group through the timeline, and the mean value is the same as the initial value by the fifth time point for each group. Fitting an rm-ANOVA model (1) to this data produces the fit that appears in panel D in Figure 1.

A comparison of the fitted mean response of the rm-ANOVA model to the the simulated data in Figure ((1, D) indicates that the model is not capturing the changes within each group in a good way. Specifically, note that the fitted mean response of the rm-ANOVA model (panel D) shows that the change (increase for Treatment 1 or decrease for Treatment 2) in the response through time points 2 and 4 is not being captured by the model. Moreover, the rm-ANOVA model is not being able to capture the fact that the initial values are the same in each group, and instead fits non-parallel lines that have initial values that are markedly different from the “true” initial values in each case (compare panels C and D). If such change has important physiological implications,the rm-ANOVA model omits it from the fitted mean response. Thus, even though the model correctly detects a difference in treatment groups, the exact nature of this difference is not correctly identified, limiting valuable inferences from the data.

This section has used simulation to better convey the limitations of linearity and correlation in the response in non-linear data. Although the model fitted to the simulated data was an rm-ANOVA model, the main issue of an expected linear trend in the response is the same in the case of a LMEM. In the following section, we present generalized additive models (GAMs) as a data-driven alternative method to analyze longitudinal non-linear data.

4 GAMs as a special case of Generalized Linear Models

4.1 GAMs and Basis Functions

Generalized linear models (GLMs) are a family of models that fit a linear response function to data that do not have normally distributed errors[50]. In contrast, GAMs are a family of regression-based methods for estimating smoothly varying trends and are a broader class of models that contain the GLM family as a special case[34,37,51]. A GAM model can be written as:

$$y_{ijt} = \beta_0 + f(x_t \mid \beta_j) + \varepsilon_{ijt} \tag{5}$$

Where y_{ijt} is the response at time t of subject i in group j, β_0 is the expected value at time 0, the change of y_{ijt} over time is represented by the function $f(x_t \mid \beta_j)$ with inputs as the covariates x_t and parameters β_j, and ε_{ijt} represents the residual error.

In contrast to the linear functions used to model the relationship between the covariates and the response in rm-ANOVA or LMEM, GAMs use more flexible *smooth functions*. This approach is advantageous as it does not restrict the model to a linear relationship, although a GAM will estimate a linear relationship if the data is consistent with a linear response. One possible set of functions for $f(x_t \mid \beta_j)$ that allow for non-linear responses are polynomials, but a major limitation is that polynomials create a “global” fit as they assume that the same relationship exists everywhere, which can cause problems with the fit [36]. In particular, polynomial fits are known to show boundary effects because as t goes to $\pm\infty$, $f(x_t \mid \beta_j)$ goes to $\pm\infty$ which is almost always unrealistic, and causes bias at the endpoints of the time period.

The smooth functional relationship between the covariates and the response in GAMs is specified using a semi-parametric relationship that can be fit within the GLM framework, by using *basis functions* expansions of the covariates and by estimating random coefficients for these basis functions. A *basis* is a set of functions that spans the space where the smooths that approximate $f(x_t \mid \beta_j)$ exist [34]. For the linear model in Equation (1), the basis coefficients are β_1, β_2 and β_3 and the basis vectors are $time_t$, $treatment_j$ and $time_t \times treatment_j$. The basis function then, is the combination of basis coefficients and basis vectors that map the possible relationship between the covariates and the response [52], which in the case of Equation (1) is restricted to a linear family of functions. In the case of Equation (5), the basis function is $f(x_t \mid \beta_j)$, which means that the model allows for non-linear relationships among the covariates.

A commonly used *basis function* is the cubic spline, which is a smooth curve constructed from cubic polynomials joined together in a manner that enforces smoothness [34,37]. Cubic splines have a long history in solving semi-parametric statistical problems and are often a default choice to fit GAMs as they are a simple, flexible and powerful option to obtain smoothness [53]. Therefore, this data-driven flexibility in GAMs overcomes the limitation that occurs in LMEMs and rm-ANOVA when the data is non linear.

To further clarify the concept of basis functions and smooth functions, consider the simulated response for Group 1 in Figure (1, C). The simplest GAM model that can be used to estimate such response is that of a single smooth term for the time effect; i.e., a model that fits a smooth to the trend of the group through time.

The timeline can be divided in equally spaced *knots*, each knot being a region where a different basis function will be used. Because there are six timepoints for this group, five knots can be used. The model with five knots to construct the smooth term means that it will have four basis functions (plus one that corresponds to the intercept). The choice of basis functions is already optimized in the package *mgcv* depending on the number of knots. In Panel A of Figure 2, the four basis functions (and the intercept) are shown. Each of the basis functions is composed of six different points (because there are six points on the timeline). To control the "wigliness" of the fit, each of the basis functions of Panel A is penalized by multiplying it by a coefficient according to the penalty matrix of Panel B. The penalty reduces the "wigliness" of the smooth fit to prevent overfitting: A weak penalty estimate will result in wiggly functions whereas a strong penalty estimate provides evidence that a linear response is appropriate.

In other words, the six points of each basis are multiplied by the corresponding coefficient in panel B, thereby increasing or decreasing the original basis functions of Panel A. In Figure 2, Panel C shows the resulting penalized basis functions. Note that the penalization for basis 1 has resulted in a decrease of its overall value (because the coefficient for that basis function is negative and less than 1); on the other hand, basis 3 has roughly doubled its value. Finally, the penalized basis functions are added at each timepoint to produce the smooth term. The resulting smooth term for the effect of *time* is shown in Panel D (orange line) along the simulated values per group, which appear as points.

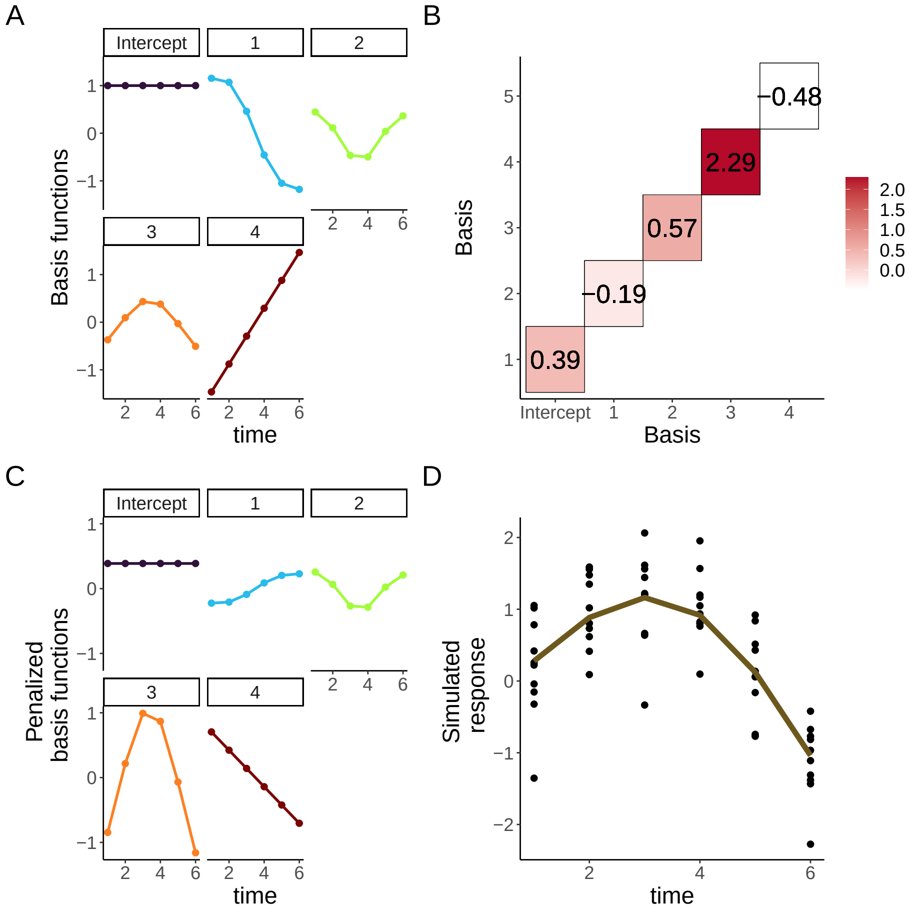


Figure 2: Basis functions for a single smoother for time with five knots. A: Basis functions for a single smoother for time for the simulated data of Group 1 from Figure 2, the intercept basis is not shown. B: Penalty matrix for the basis functions. Each basis function is penalized by a coefficient which can be positive or negative. The coefficient determines the overall effect of each basis in the final smoother. C: Penalized basis functions. Each of the four basis functions of panel A has been penalized by the corresponding coefficient shown in Panel B, note the corresponding increase (or decrease) of each basis. D: Smoother for time and original data points. The smoother (line) is the result of the sum of each penalized basis function at each time point, simulated values for the group appear as points.

5 The analyisis of longitudinal biomedical data using GAMs

The previous sections provided the basic theoretical framework to understand the GAM framework and how these models are more advantageous to analyze non-linear longitudinal data when compared to LMEMs or rm-ANOVA. This section will use simulation to present the practical implementation of GAMs for longitudinal biomedical data using R and the package `mgcv`. The code for the simulated data and figures, and a brief guide for model selection and diagnostics appear in the Appendix.

5.1 Simulated data

The simulated data is based on the reported longitudinal changes in oxygen saturation (StO_2) in subcutaneous tumors that appear in Figure 3(C) in [16]. In the paper, diffuse reflectance spectroscopy was used to quantify StO_2 changes in both groups at the same time points (days 0, 2, 5, 7 and 10). In the "Treatment" group (chemotherapy) an increase in StO_2 is observed through time, while a decrease is seen in the "Control" (saline) group. Following the reported trend, we simulated 10 normally distributed observations at each time point with a standard deviation (SD) of 10% (matching the SD in the original paper). The simulated and real data appear in Figure 3, A and the inlet, respectively.

5.2 An interaction GAM for longitudinal data

An interaction model is typically the main interest in longitudinal biomedical data, as it takes into account the effect of treatment, time, and their combination. In a practical sense, when a GAM is implemented for longitudinal data, a smooth can be added to the model for the *time* effect to account for the repeated measures over time. Although specific methods of how GAMs model correlation structures is a topic beyond the scope of this paper, it suffices to say that GAMs are flexible and can handle correlation structures beyond compound symmetry. A detailed description on basis functions and correlations can be found in [52].

For the data in Figure 3,A it is of interest to know the effect that each treatment has in StO_2. The model then needs to incorporate independent smooths for *Group* and *Day*, respectively. The main thing to consider is that model syntax accounts for the fact that one of the variables is numeric (*Day*) and the other is a factor (*Group*). Because the smooths are centered at 0, the factor variable needs to be specified as a parametric term in order to identify any differences between the groups. Using R and the package `mgcv` the model syntax is:

```
m1<-gam(StO2_sim~Group+s(Day,by=Group,k=5,bs="gp"), method='REML',data=dat_sim
```

This syntax specifies that `m1` will store the model, and that the change in the simulated oxygen saturation (`StO2_sim`) is modeled using independent smooths for *Group* and *Day* (the parenthesis preceded by *s*) using 5 knots. The smooth is constructed using gaussian process smooths, which is indicated by `bs="gp"`. These splines are used to model temporal trends and might be particularly suited for long-term studies where the correlation between measurements changes as a function of the time intervals [34]. The parametric term *Group* is added to quantify differences in the effect of treatment between groups, and the `method` chosen to select the smoothing parameters is the restricted maximum likelihood (REML) [37]. When the smooths are plotted over the raw data, it is clear that the model has been able to capture the trend of the change of StO_2 for each group across time (Figure 3,B). Model diagnostics can be obtained using the `gam.check` function, and the function `appraise` from the package *gratia*[54]. A guide for model selection and diagnostics is in the Appendix, and an in-depth analysis can be found in [37] and [55].

One question that might arise at this point is "what is the fit that an rm-ANOVA model produces for the simulated data?" The rm-ANOVA model, which corresponds to Equation (1) is presented in Figure 3,C. This is a typical case of model misspecification: The slopes of each group are different, which would lead to a *p-value* indicating significance for the treatment and time effects, but the model is not capturing the changes that occur at days 2 and between days 5 and 7, whereas the GAM model is able to do so (Figure 3,B) .

Because GAMs do not require equally-spaced or complete observations for all subjects, they are advantageous to analyze longitudinal data where missingness exists. The rationale behind this is that GAMs are able to pick the trend in the data even when some observations are missing. However, this usually causes the resulting smooths to have wider confidence intervals and less ability to pick certain trends. Consider the simulated StO_2 values from Figure (3, B). If 40% of the total observations are randomly deleted and the same interaction GAM fitted for the complete dataset is used, the resulting smooths are still able to show a different trend for each group, but the "Treatment" smooth takes an overall more linear profile (3, D). Although the confidence intervals have increased for both smooths, the model still shows different trends with as little as 4 observations per group at certain time points.

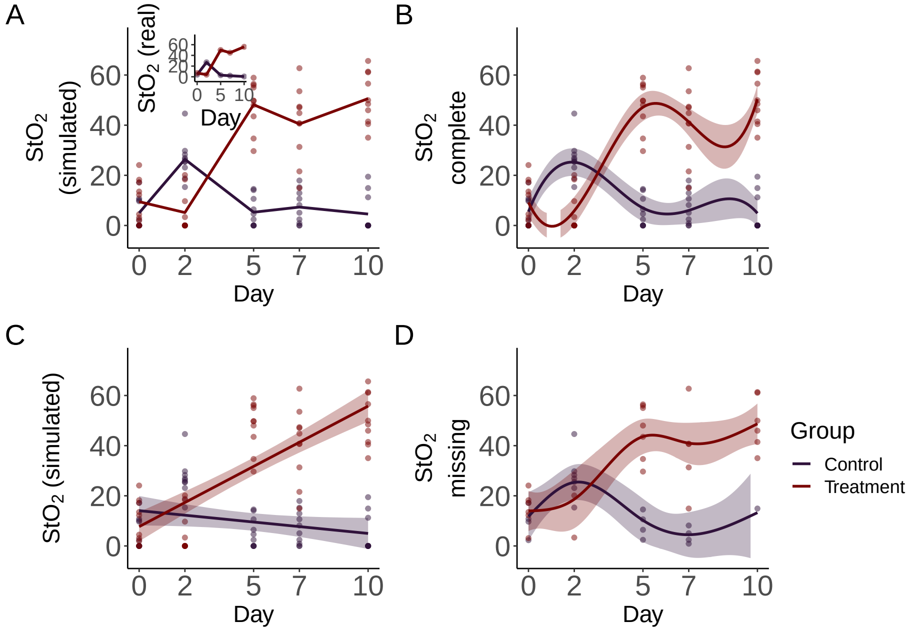


Figure 3: Simulated data and smooths for oxygen saturation in tumors. A: Simulated data that follows previously reported trends (inset) in tumors under chemotherapy (Treatment) or saline (Control) treatment. Simulated data is from a normal distribution with standard deviation of 10% with 10 observations per time point. Lines indicate mean oxygen saturation B: Smooths from the GAM model for the full simulated data with interaction of Group and Treatment. Lines represent trends for each group, shaded regions are 95% confidence intervals. C: rm-ANOVA model for the simulated data, the model does not capture the changes in each group over time. D: Smooths for the GAM model for the simulated data with 40% of its observations missing. Lines represent trends for each group, shaded regions are 95% confidence intervals.

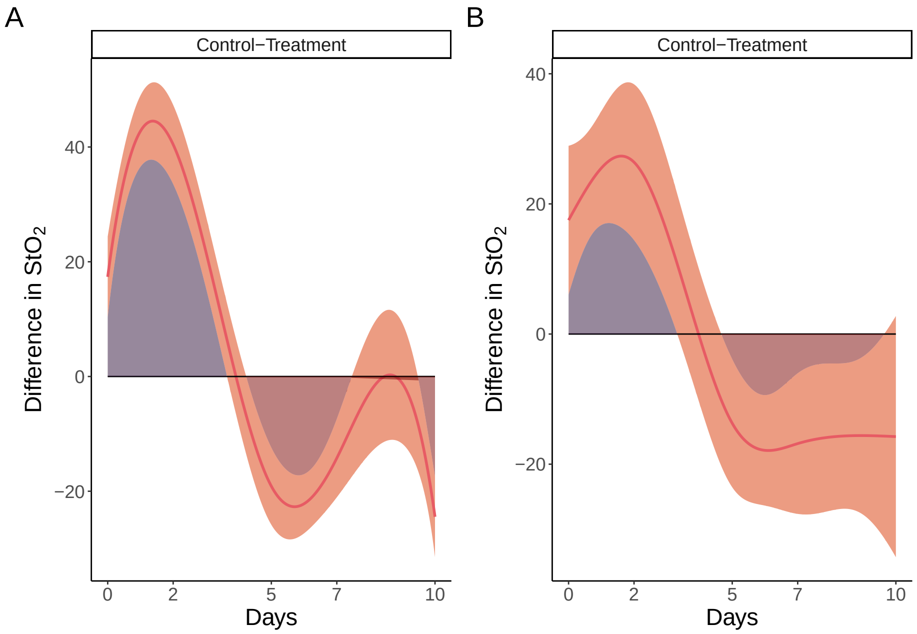


Figure 4: Pairwise comparisons for smooth terms. A: Pairwise comparisons for the full dataset. B: Pairwise comparisons for the dataset with missing observations. Significant differences exist where the interval does not cover 0. In both cases the effect of treatment is significant after day 5.

5.3 Determination of significance in GAMs for longitudinal data

At the core of a biomedical longitudinal study lies the question of a significant difference between the effect of two or more treatments in different groups. Whereas in rm-ANOVA a *post-hoc* analysis is required to answer such question by calculating some *p-values* after multiple comparisons, GAMs can use a different approach to estimate significance. In essence, the idea behind the estimation of significance in GAMs across different treatment groups is that if the *difference* between the confidence intervals of the fitted smooths for such groups is non-zero, then a significant difference exists at that time point(s). The absence of a *p-value* in this case might seem odd, but the confidence interval comparison can be conceptualized in the following manner: Different trends in each group are an indication of an effect by the treatment. This is what happens for the simulated data in Figure 3, A where the chemotherapy causes StO_2 to increase over time.

With this expectation of different trends in each group, computing the difference between the trends will identify if the observed change is significant. The difference between groups with similar trends is likely to yield zero, which would indicate that the treatment is not causing a change in the response in one of the groups (assuming the other group is a Control or Reference group).

Consider the calculation of pairwise differences for the smooths in 3 B and D. Figure 4, shows the comparison between each treatment group.Here, the effect of the "Control" group is compared to that of the "Treatment" group. The shaded region under the curve highlights the interval where each group has a higher effect than the other. Notice that the shaded region between days 0 and 4 indicates that through that time, the "Control" group has higher StO_2, but as therapy progresses the effect is reversed and by day 5 it is the "Treatment" group the one that has greater contribution. This would suggest that the effect of chemotherapy is significant after day 5 for the model used; which is noticeable in the data with missing observations albeit the missing data model has a wider confidence interval and is not able to pick the change between days 7 and 10 that the full dataset model is able to detect.

6 Conclusion

The traditional methods to analyze biomedical longitudinal data (rm-ANOVA, LMEMs) are restricted to cases when the data follows a linear pattern. When the trend of the data is non-linear (as it is frequently the case in biomedical research), the use of rm-ANOVA and LMEMs leads to unreliable inference and biased estimates. GAMs present a suitable alternative to analyze non-linear longitudinal biomedical data, as they overcome the limitations of linearity, compound symmetry, and missing observations imposed by traditional methods. By presenting in a clear and visual manner the limitations of rm-ANOVA and LMEMs, and the implementation of GAMs using simulated data that follows previously reported trends in the literature, we aim at encouraging biomedical researchers to consider different statistical tools to analyze longitudinal data. Finally, by providing the data and code used in this paper we hope to address the need of creating and sharing reproducible work in biomedical research.

7 References

[1] D. Roblyer, S. Ueda, A. Cerussi, W. Tanamai, A. Durkin, R. Mehta, D. Hsiang, J.A. Butler, C. McLaren, W.P. Chen, B. Tromberg, Optical imaging of breast cancer oxyhemoglobin flare correlates with neoadjuvant chemotherapy response one day after starting treatment, Proceedings of the National Academy of Sciences of the United States of America. 108 (2011) 14626–14631. https://doi.org/10.1073/pnas.1013103108.

[2] A. Tank, H.M. Peterson, V. Pera, S. Tabassum, A. Leproux, T. O'Sullivan, E. Jones, H. Cabral, N. Ko, R.S. Mehta, B.J. Tromberg, D. Roblyer, Diffuse optical spectroscopic imaging reveals distinct early breast tumor hemodynamic responses to metronomic and maximum tolerated dose regimens, Breast Cancer Research. 22 (2020) 1–10. https://doi.org/doi:10.1186/s13058-020-01262-1.

[3] M.V. Pavlov, T.I. Kalganova, Y.S. Lyubimtseva, V.I. Plekhanov, G.Y. Golubyatnikov, O.Y. Ilyinskaya, A.G. Orlova, P.V. Subochev, D.V. Safonov, N.M. Shakhova, A.V. Maslennikova, Multimodal approach in assessment of the response of breast cancer to neoadjuvant chemotherapy, Journal of Biomedical Optics. 23 (2018). https://doi.org/%7B10.1117/1.JBO.23.9.091410%7D.

[4] V. Demidov, A. Maeda, M. Sugita, V. Madge, S. Sadanand, C. Flueraru, I.A. Vitkin, Preclinical longitudinal imaging of tumor microvascular radiobiological response with functional optical coherence tomography, Scientific Reports. 8 (2018). https://doi.org/%7B10.1038/s41598-017-18635-w%7D.

[5] G. Ritter, L. Cohen, C. Williams, E. Richards, L. Old, S. Welt, Serological analysis of human anti-human antibody responses in colon cancer patients treated with repeated doses of humanized monoclonal antibody A33, Cancer Research. 61 (2001) 6851–6859.

[6] E.M. Roth, A.C. Goldberg, A.L. Catapano, A. Torri, G.D. Yancopoulos, N. Stahl, A. Brunet, G. Lecorps, H.M. Colhoun, Antidrug Antibodies in Patients Treated with Alirocumab, New England Journal of Medicine. 376 (2017) 1589–1590. https://doi.org/%7B10.1056/NEJMc1616623%7D.

[7] J.D. Jones, H.E. Ramser, A.E. Woessner, K.P. Quinn, In vivo multiphoton microscopy detects longitudinal metabolic changes associated with delayed skin wound healing, Communications Biology. 1 (2018). https://doi.org/%7B10.1038/s42003-018-0206-4%7D.

[8] M.C. Skala, A. Fontanella, L. Lan, J.A. Izatt, M.W. Dewhirst, Longitudinal optical imaging of tumor metabolism and hemodynamics, Journal of Biomedical Optics. 15 (2010). https://doi.org/10.1117/1.3285584.

[9] G.J. Greening, K.P. Miller, C.R. Spainhour, M.D. Cato, T.J. Muldoon, Effects of isoflurane anesthesia on physiological parameters in murine subcutaneous tumor allografts measured via diffuse reflectance spectroscopy, Biomedical Optics Express. 9 (2018) 2871–2886. https://doi.org/%7B10.1364/BOE.9.002871%7D.

[10] T.T. Sio, P.J. Atherton, B.J. Birckhead, D.J. Schwartz, J.A. Sloan, D.K. Seisler, J.A. Martenson, C.L. Loprinzi, P.C. Griffin, R.F. Morton, J.C. Anders, T.J. Stoffel, R.E. Haselow, R.B. Mowat, M.A.N. Wittich, J.D. Bearden III, R.C. Miller, Repeated measures analyses of dermatitis symptom evolution in breast cancer patients receiving radiotherapy in a phase 3 randomized trial of mometasone furoate vs placebo (N06C4 [alliance]), Supportive Care in Cancer. 24 (2016) 3847–3855. https://doi.org/%7B10.1007/s00520-016-3213-3%7D.

[11] J.I. Kamstra, P.U. Dijkstra, M. van Leeuwen, J.L.N. Roodenburg, J.A. Langendijk, Mouth opening in patients irradiated for head and neck cancer: A prospective repeated measures study, Oral Oncology. 51 (2015) 548–555. https://doi.org/%7B10.1016/j.oraloncology.2015.01.016%7D.

[12] E.-J. Wagenmakers, M. Lee, T. Lodewyckx, G.J. Iverson, Bayesian versus frequentist inference, in: H. Hoijtink, I. Klugkist, P.A. Boelen (Eds.), Bayesian Evaluation of Informative Hypotheses, Springer New York, New York, NY, 2008: pp. 181–207. https://doi.org/10.1007/978-0-387-09612-4_9.

[13] R. Gueorguieva, J.H. Krystal, Move over ANOVA - Progress in analyzing repeated-measures data and its reflection in papers published in the archives of general psychiatry, Archives of General Psychiatry. 61 (2004) 310–317. https://doi.org/10.1001/archpsyc.61.3.310.

[14] P. Schober, T.R. Vetter, Repeated measures designs and analysis of longitudinal data: If at first you do not succeed-try, try again, Anesthesia and Analgesia. 127 (2018) 569–575. https://doi.org/10.1213/ane.0000000000003511.

[15] J. Pinheiro, D. Bates, Mixed-effects models in S and S-PLUS, Springer Science & Business Media, 2006. https://doi.org/https://doi.org/10.1007/b98882.

[16] K. Vishwanath, H. Yuan, W.T. Barry, M.W. Dewhirst, N. Ramanujam, Using optical spectroscopy to longitudinally monitor physiological changes within solid tumors, Neoplasia. 11 (2009) 889–900. https://doi.org/10.1593/neo.09580.

[17] B. Dennis, J.M. Ponciano, M.L. Taper, S.R. Lele, Errors in Statistical Inference Under Model Misspecification: Evidence, Hypothesis Testing, and AIC, Frontiers in Ecology and Evolution. 7 (2019). https://doi.org/%7B10.3389/fevo.2019.00372%7D.

[18] B. Wang, Z. Zhou, H. Wang, X.M. Tu, C. Feng, The p-value and model specification in statistics, General Psychiatry. 32 (2019). https://doi.org/%7B10.1136/gpsych-2019-100081%7D.

[19] C. Liu, T.P. Cripe, M.-O. Kim, Statistical issues in longitudinal data analysis for treatment efficacy studies in the biomedical sciences, Molecular Therapy. 18 (2010) 1724–1730. https://doi.org/10.1038/mt.2010.127.

[20] L.G. Halsey, D. Curran-Everett, S.L. Vowler, G.B. Drummond, The fickle P value generates irreproducible results, Nature Methods. 12 (2015) 179–185. https://doi.org/%7B10.1038/nmeth.3288%7D.

[21] H. Abdi, Holm's Sequential Bonferroni Procedure, Encyclopedia of Research Design. 1 (2010) 1–8. https://doi.org/10.4135/9781412961288.n178.

[22] S. Nakagawa, A farewell to Bonferroni: the problems of low statistical power and publication bias, Behavioral Ecology. 15 (2004) 1044–1045. https://doi.org/%7B10.1093/beheco/arh107%7D.

[23] A. Gelman, J. Hill, M. Yajima, Why We (Usually) Don't Have to Worry About Multiple Comparisons, Journal of Research on Educational Effectiveness. 5 (2012) 189–211. https://doi.org/%7B10.1080/19345747.2011.618213%7D.

[24] C. Albers, The problem with unadjusted multiple and sequential statistical testing, Nature Communications. 10 (2019). https://doi.org/%7B10.1038/s41467-019-09941-0%7D.

[25] C. Ugrinowitsch, G.W. Fellingham, M.D. Ricard, Limitations of ordinary least squares models in analyzing repeated measures data, Medicine and Science in Sports and Exercise. 36 (2004) 2144–2148. https://doi.org/10.1249/01.mss.0000147580.40591.75.

[26] H. Huynh, L.S. Feldt, Estimation of the box correction for degrees of freedom from sample data in randomized block and split-plot designs, Journal of Educational Statistics. 1 (1976) 69–82. https://doi.org/10.3102/10769986001001069.

[27] S.W. Greenhouse, S. Geisser, On methods in the analysis of profile data, Psychometrika. 24 (1959) 95–112. https://doi.org/10.1007/bf02289823.

[28] N. Haverkamp, A. Beauducel, Violation of the Sphericity Assumption and Its Effect on Type-I Error Rates in Repeated Measures ANOVA and Multi-Level Linear Models (MLM), Frontiers in Psychology. 8 (2017). https://doi.org/%7B10.3389/fpsyg.2017.01841%7D.

[29] H. Keselman, J. Algina, R. Kowalchuk, The analysis of repeated measures designs: A review, British Journal of Mathematica & Statistical Psychology. 54 (2001) 1–20. https://doi.org/%7B10.1348/000711001159357%7D.

[30] Jaykaran. Charan, N. Kantharia, How to calculate sample size in animal studies?, Journal of Pharmacology and Pharmacotherapeutics. 4 (2013) 303–306. https://doi.org/10.4103/0976-500X.119726.

[31] D.J. Barr, R. Levy, C. Scheepers, H.J. Tily, Random effects structure for confirmatory hypothesis testing: Keep it maximal, JournalL of Memory and Language. 68 (2013) 255–278. https://doi.org/%7B10.1016/j.jml.2012.11.001%7D.

[32] N.L. Rose, H.D. Yang, S.D. Turner, G.L. Simpson, An assessment of the mechanisms for the transfer of lead and mercury from atmospherically contaminated organic soils to lake sediments with particular reference to scotland, UK, Geochimica Et Cosmochimica Acta. 82 (2012) 113–135. https://doi.org/10.1016/j.gca.2010.12.026.

[33] E.J. Pedersen, D.L. Miller, G.L. Simpson, N. Ross, Hierarchical generalized additive models in ecology: An introduction with mgcv, Peerj. 7 (2019). https://doi.org/10.7717/peerj.6876.

[34] G.L. Simpson, Modelling palaeoecological time series using generalised additive models, Frontiers in Ecology and Evolution. 6 (2018). https://doi.org/10.3389/fevo.2018.00149.

[35] L. Yang, G. Qin, N. Zhao, C. Wang, G. Song, Using a generalized additive model with autoregressive terms to study the effects of daily temperature on mortality, BMC Medical Research Methodology. 12 (2012). https://doi.org/%7B10.1186/1471-2288-12-165%7D.

[36] N. Beck, S. Jackman, Beyond linearity by default: Generalized additive models, American Journal of Political Science. (1998) 596–627.

[37] S.N. Wood, Generalized additive models: An introduction with , second edition, CRC Press LLC, Philadelphia, PA, 2017.

[38] R Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 2020. https://www.R-project.org/.

[39] S.N. Wood, N. Pya, B. Saefken, Smoothing Parameter and Model Selection for General Smooth Models, Journal of the American Statistical Association. 111 (2016) 1548–1563. https://doi.org/%7B10.1080/01621459.2016.1180986%7D.

[40] B.T. West, K.B. Welch, A.T. Galecki, Linear mixed models: A practical guide using statistical software, second edition, Taylor & Francis, 2014. https://books.google.com/books?id=hjT6AwAAQBAJ.

[41] R.D. Wolfinger, Heterogeneous variance: Covariance structures for repeated measures, Journal of Agricultural, Biological, and Environmental Statistics. 1 (1996) 205–230. http://www.jstor.org/stable/1400366.

[42] R.E. Weiss, Modeling longitudinal data, Springer New York, 2005. https://books.google.com/books?id=MQ/_bvWDPsEAC.

[43] S. Geisser, S.W. Greenhouse, An Extension of Box's Results on the Use of the F Distribution in Multivariate Analysis, The Annals of Mathematical Statistics. 29 (1958) 885–891. https://doi.org/10.1214/aoms/1177706545.

[44] S.E. Maxwell, H.D. Delaney, K. Kelley, Designing experiments and analyzing data: A model comparison perspective, third edition, Taylor & Francis, 2017. https://books.google.com/books?id=NmFQDwAAQBAJ.

[45] G. Molenberghs, H. Thijs, I. Jansen, C. Beunckens, M. Kenward, C. Mallinckrodt, R. Carroll, Analyzing incomplete longitudinal clinical trial data, Biostatistics. 5 (2004) 445–464. https://doi.org/%7B10.1093/biostatistics/kxh001%7D.

[46] J. Scheffer, Dealing with missing data, Research Letters in the Information and Mathematical Sciences. 3 (2002) 153–160.

[47] R.F. Potthoff, G.E. Tudor, K.S. Pieper, V. Hasselblad, Can one assess whether missing data are missing at random in medical studies?, STATISTICAL METHODS IN MEDICAL RESEARCH. 15 (2006) 213–234. https://doi.org/%7B10.1191/0962280206sm448oa%7D.

[48] Y. Ma, M. Mazumdar, S.G. Memtsoudis, Beyond Repeated-Measures Analysis of Variance Advanced Statistical Methods for the Analysis of Longitudinal Data in Anesthesia Research, Regional Anesthesia and Pain Medicine. 37 (2012) 99–105. https://doi.org/%7B10.1097/AAP.0b013e31823ebc74%7D.

[49] J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, R Core Team, nlme: Linear and nonlinear mixed effects models, 2020. https://CRAN.R-project.org/package=nlme.

[50] J.A. Nelder, R.W.M. Wedderburn, Generalized linear models, Journal of the Royal Statistical Society. Series A (General). 135 (1972) 370–384. http://www.jstor.org/stable/2344614.

[51] T. Hastie, R. Tibshirani, Generalized additive models: Some applications, Journal of the American Statistical Association. 82 (1987) 371–386. https://doi.org/10.1080/01621459.1987.10478440.

[52] T.J. Hefley, K.M. Broms, B.M. Brost, F.E. Buderman, S.L. Kay, H.R. Scharf, J.R. Tipton, P.J. Williams, M.B. Hooten, The basis function approach for modeling autocorrelation in ecological data, ECOLOGY. 98 (2017) 632–646. https://doi.org/%7B10.1002/ecy.1674%7D.

[53] E.J. Wegman, I.W. Wright, Splines in statistics, Journal of the American Statistical Association. 78 (1983) 351–365. https://doi.org/10.1080/01621459.1983.10477977.

[54] G.L. Simpson, Gratia: Graceful 'ggplot'-based graphics and other functions for GAMs fitted using 'mgcv', 2020. https://CRAN.R-project.org/package=gratia.

[55] J. Harezlak, D. Ruppert, M.P. Wand, Semiparametric regression with r, Springer New York, 2018. https://doi.org/10.1007/978-1-4939-8853-2.

A Code for Manuscript data

This section presents the code used to generate figures, models and simulated data from Sections 3 and 4 from the main manuscript.

A.1 Compound symmetry and independent errors in linear and quadratic responses

This section simulated linear and quadratic data in the same manner as in Section 3.5. The linear simulations using Figure 5 show in panels A and D the simulated mean responses and individual data points. Panels C and G show a visual interpretation of "correlation" in the responses: In panel C, subjects that have a value of the random error ε either above or below the mean group response are more likely to have other observations that follow the same trajectory, thereby demonstrating correlation in the response. In panel G,because the errors are independent, there is no expectation that responses are likely to follow a similar pattern. Panels D and H show the predictions from the rm-ANOVA model.

The following code produces a more comprehensive exploration of Figure 1 in the main manuscript.

```
###################Section for calculations
   ######################################
#
   ################################################################################


## Example with linear response

#This function simulates data using a linear or quadratic mean response
   and each with correlated
#or uncorrelated errors. Each group has a different slope/concavity.
example <- function(n_time = 6, #number of time points
                    fun_type = "linear", #type of response
                    error_type = "correlated") {

  if (!(fun_type %in% c("linear", "quadratic")))
    stop('fun_type must be either "linear", or "quadratic"')
  if (!(error_type %in% c("correlated", "independent")))
    stop('fun_type must be either "correlated", or "independent"')


  x <- seq(1,6, length.out = n_time)

  #Create mean response matrix: linear or quadratic
  mu <- matrix(0, length(x), 2)
  # linear response
  if (fun_type == "linear") {
    mu[, 1] <- - (0.25*x)+2
    mu[, 2] <- 0.25*x+2
  } else {
    # quadratic response (non-linear)

    mu[, 1] <-  -(0.25 * x^2) +1.5*x-1.25
    mu[, 2] <- (0.25 * x^2) -1.5*x+1.25
  }

  #create an array where individual observations per each time point for
     each group are to be stored. Currently using 10 observations per
     timepoint
  y <- array(0, dim = c(length(x), 2, 10))

  #Create array to store the "errors" for each group at each timepoint.
     The "errors" are the
  #between-group variability in the response.
  errors <- array(0, dim = c(length(x), 2, 10))
  #create an array where 10 observations per each time point for each
     group are to be stored

  #The following cycles create independent or correlated responses. To
     each value of mu (mean response per group) a randomly generated error
      (correlated or uncorrelated) is added and thus the individual
     response is created.
  if (error_type == "independent") {
    ## independent errors
    for (i in 1:2) {
      for (j in 1:10) {
        errors[, i, j] <- rnorm(6, 0, 0.25)
        y[, i, j] <- mu[, i] + errors[, i, j]
      }
    }
  } else {
    for (i in 1:2) {     # number of treatments
      for (j in 1:10) {  # number of subjects
        # compound symmetry errors: variance covariance matrix
        errors[, i, j] <- rmvn(1, rep(0, length(x)), 0.1 * diag(6) + 0.25
           * matrix(1, 6, 6))
        y[, i, j] <- mu[, i] + errors[, i, j]
      }
    }
  }


  ## subject random effects

  ## visualizing the difference between independent errors and compound
     symmetry
  ## why do we need to account for this -- overly confident inference

#labelling y and errors
  dimnames(y) <- list(time = x,
                      treatment = 1:2,
                      subject = 1:10)

  dimnames(errors) <- list(time = x,
                           treatment = 1:2,
                           subject = 1:10)

  #labeling the mean response
  dimnames(mu) <- list(time = x,
                       treatment = 1:2)

  #convert y, mu and errors to  dataframes with time, treatment and
     subject columns
  dat <- as.data.frame.table(y,
                             responseName = "y")
  dat_errors <- as.data.frame.table(errors,
                                    responseName = "errors")
  dat_mu <- as.data.frame.table(mu,
                                responseName = "mu")

  #join the dataframes to show mean response and errors per subject
  dat <- left_join(dat, dat_errors,
                   by = c("time", "treatment", "subject"))
  dat <- left_join(dat, dat_mu,
                   by = c("time", "treatment"))
  #add time
  dat$time <- as.numeric(as.character(dat$time))
```

```r
  #label subjects per group
  dat <- dat %>%
    mutate(subject = factor(paste(subject,
                                  treatment,
                                  sep = "-")))


  ## repeated measures ANOVA in R
#time and treatment interaction model, compound symmetry required by the
   model
  fit_lme <- lme(y ~ treatment + time + treatment:time,
                 data = dat,
                 random = ~ 1 | subject,
                 correlation = corCompSymm(form = ~ 1 | subject)
  )

  #create a prediction frame where the model can be used for plotting
     purposes
  pred_dat <- expand.grid(
    treatment = factor(1:2),
    time = unique(dat$time)
  )

  #add model predictions to the dataframe that has the simulated data
  dat$y_pred <- predict(fit_lme)

  #return everything in a list
  return(list(
    dat = dat,
    pred_dat = pred_dat,
    fit_lme = fit_lme

  ))
}
##################Section for plotting#################################
#######################################################################
#This function will create the plots for either a "linear" or "quadratic"
   response

plot_example <- function(sim_dat) {
  ## Plot the simulated data (scatterplot)
  p1 <- sim_dat$dat %>%
    ggplot(aes(x = time,
               y = y,
               group = treatment,
               color = treatment)
           ) +
    geom_point(show.legend=FALSE) +
    labs(y='response')+
    geom_line(aes(x = time,
                  y = mu,
                  color = treatment),
              show.legend=FALSE) +
    theme_classic() +
    theme(plot.title = element_text(size = 30,
                                  face = "bold"),
        text=element_text(size=30))+
    thm

  #plot the simulated data with trajectories per each subject
  p2 <- sim_dat$dat %>%
    ggplot(aes(x = time,
               y = y,
               group = subject,
               color = treatment)
           ) +
    geom_line(aes(size = "Subjects"),
              show.legend = FALSE) +
    # facet_wrap(~ treatment) +
    geom_line(aes(x = time,
                  y = mu,
                  color = treatment,
                  size = "Simulated Truth"),
              lty = 1,show.legend = FALSE) +
    labs(y=’response’)+
    scale_size_manual(name = "Type", values=c("Subjects" = 0.5, "Simulated
        Truth" = 3)) +
    theme_classic()+
     theme(plot.title = element_text(size = 30,
                                face = "bold"),
     text=element_text(size=30))+
    thm

  #plot the errors
   p3 <- sim_dat$dat %>%
    ggplot(aes(x = time,
               y = errors,
               group = subject,
               color = treatment)) +
    geom_line(show.legend=FALSE) +
     labs(y=’errors’)+
     theme_classic()+
     theme(plot.title = element_text(size = 30,
                                  face = "bold"),
        text=element_text(size=30))+
    thm

   #plot the model predictions
  p4 <- ggplot(sim_dat$dat,
               aes(x = time,
                   y = y,
                   color = treatment)) +
    geom_point()+
    labs(y=’response’)+
    geom_line(aes(y = predict(sim_dat$fit_lme),
                  group = subject, size = "Subjects")) +
    geom_line(data = sim_dat$pred_dat,
              aes(y = predict(sim_dat$fit_lme,
                              level = 0,
                              newdata = sim_dat$pred_dat),
                  size = "Population")) +
    guides(color = guide_legend(override.aes = list(size = 2)))+
    scale_size_manual(name = "Predictions",
                      values=c("Subjects" = 0.5, "Population" = 3)) +
    theme_classic() +
    theme(plot.title = element_text(size = 30,
                                  face = "bold"),
        text=element_text(size=30))+
    thm

  return((p1+p3+p2+p4)+plot_layout(nrow=1)+plot_annotation(tag_levels = 'A'))


}

txt<-18

#Store each plot in a separate object
A1<-plot_example(example(fun_type = "linear", error_type = "correlated"))

B1<-plot_example(example(fun_type = "linear", error_type = "independent"))

C1<-plot_example(example(fun_type = "quadratic", error_type = "correlated"))

D1<-plot_example(example(fun_type = "quadratic", error_type = "independent"))
```

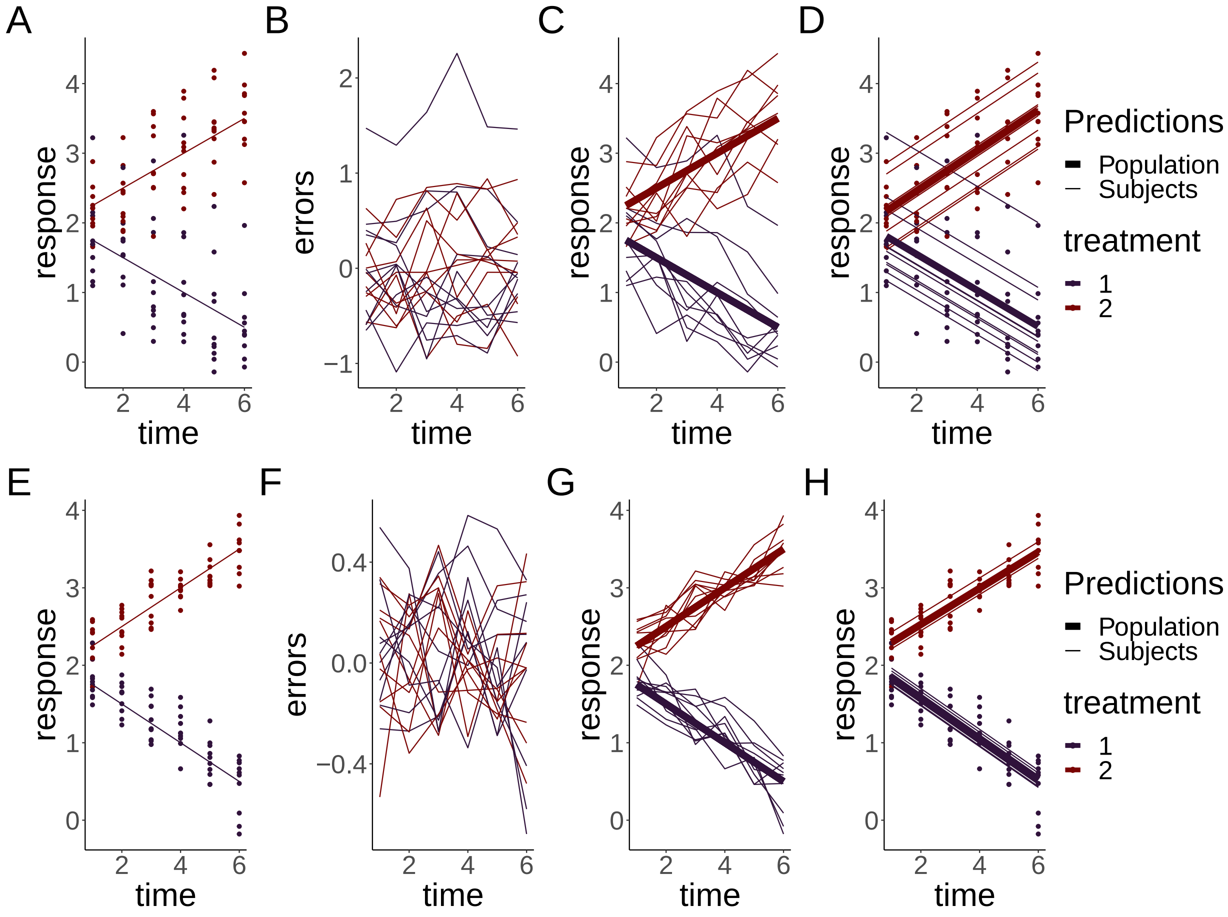


Figure 5: Simulated linear responses from two groups with correlated (top row) or independent (bottom row) errors using a rm-ANOVA model. A, C:Simulated data with known mean response (linear or quadratic, thin lines) and individual responses (points) showing the dispersion of the data. B,D: Estimations from the rm-ANOVA model for the mean group response (linear of quadratic). Thick lines are the predicted mean response per group, thin lines are the random effects for each subject and points represent the original raw data. The rm-ANOVA model does not pick the trend of the quadratic data.

For the quadratic response case, Figure 6 shows the simulated responses using compound symmetry and independent errors.

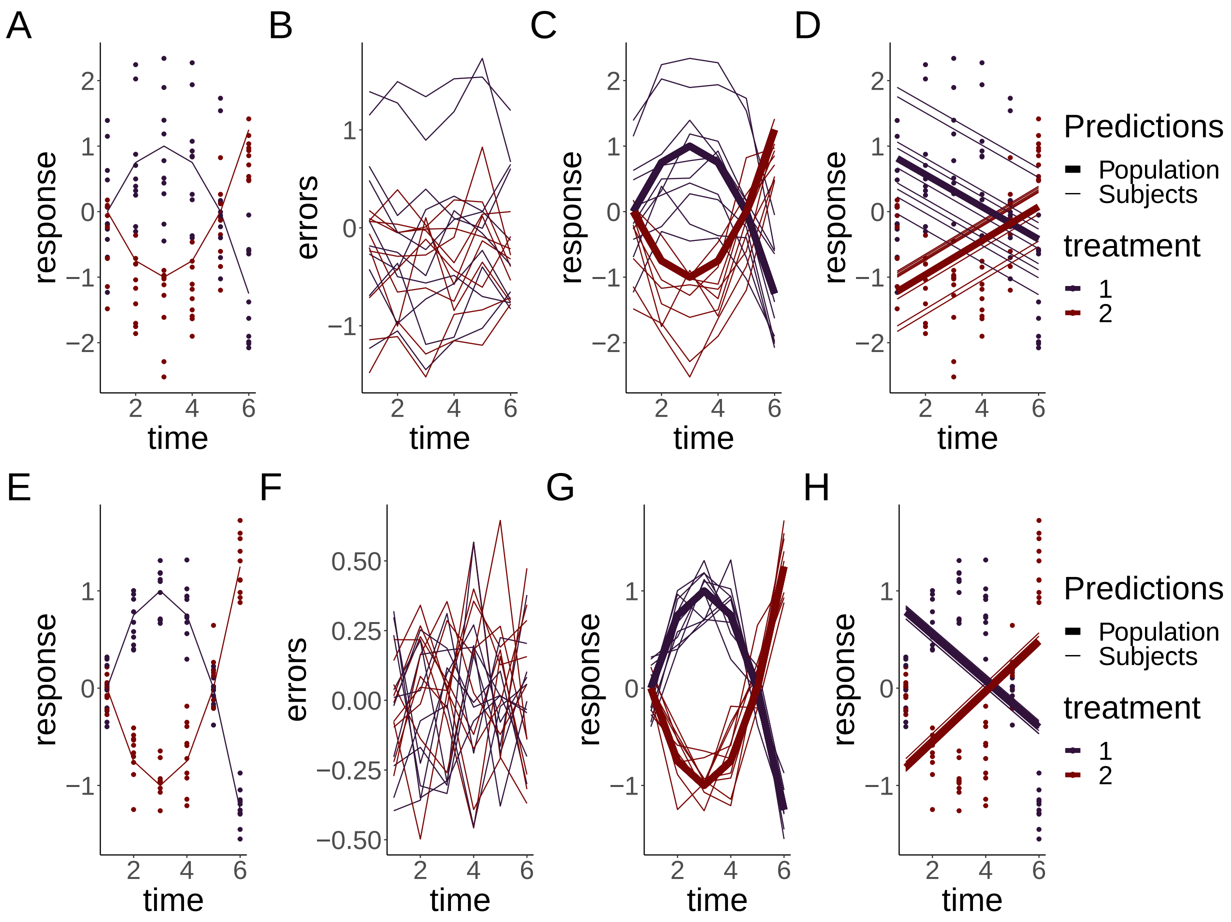


Figure 6: Simulated quadratic responses from two groups with a rm-ANOVA model fitted. A,E:Simulated data with known mean response (lines) and individual responses (points) showing the dispersion of the data. B,F: Generated errors showing the difference in the behavior of correlated and independent errors. C,G: Simulated known response per group (thick lines) with individual trajectories (thin lines), note that subjects with observations in the area above the mean response tend to stay in that region through the timeline. D,H: Estimations from the rm-ANOVA model for the mean group response. Thick lines are the predicted mean response per group, thin lines are the random effects for each subject and points represent the original raw data.

A.2 Basis functions and GAMs

This code produces Figure 2 from the main manuscript. Briefly, a non-linear (quadratic) response is simulated a gam model is fitted and the basis are extracted in order to explain how the smooth is constructed. The code for data simulation is used again here for the sake of keeping the same structure, although the data can be simulated in a more simple fashion.

```
#generate the response: the same initial procedure from the previous
   section to simulate
#the response
n_time = 6
 x <- seq(1,6, length.out = n_time)
 mu <- matrix(0, length(x), 2)
 mu[, 1] <-  -(0.25 * x^2) +1.5*x-1.25 #mean response
 mu[, 2] <- (0.25 * x^2) -1.5*x+1.25 #mean response
 y <- array(0, dim = c(length(x), 2, 10))
 errors <- array(0, dim = c(length(x), 2, 10))
 for (i in 1:2) {     # number of treatments
     for (j in 1:10) {  # number of subjects
         # compound symmetry errors
         errors[, i, j] <- rmvn(1, rep(0, length(x)), 0.1 * diag(6) + 0.25
             * matrix(1, 6, 6))
         y[, i, j] <- mu[, i] + errors[, i, j]
     }
 }

 #label each table
  dimnames(y) <- list(time = x, treatment = 1:2, subject = 1:10)
 dimnames(errors) <- list(time = x, treatment = 1:2, subject = 1:10)
 dimnames(mu) <- list(time = x, treatment = 1:2)

 #Convert to dataframes with subject, time and group columns
 dat <- as.data.frame.table(y, responseName = "y")
 dat_errors <- as.data.frame.table(errors, responseName = "errors")
 dat_mu <- as.data.frame.table(mu, responseName = "mu")
 dat <- left_join(dat, dat_errors, by = c("time", "treatment", "subject"))
 dat <- left_join(dat, dat_mu, by = c("time", "treatment"))
 dat$time <- as.numeric(as.character(dat$time))

 #label subject per group
 dat <- dat %>%
     mutate(subject = factor(paste(subject, treatment, sep = "-")))

 #extract  "Group 1" to fit the GAM
  dat<-subset(dat,treatment==1)
 #keep just the response and timepoint columns
   dat<-dat[,c(’y’,’time’)]

   #GAM model of time, 5 knots
gm<-gam(y~s(time,k=5),data=dat)

#model_matrix (also known as) ’design matrix’
#will contain the smooths used to create  model ’gm’
model_matrix<-as.data.frame(predict(gm,type=’lpmatrix’))


time<-c(1:6)

basis<-model_matrix[1:6,] #extracting basis (because the values are
   repeated after every 6 rows)
#basis<-model_matrix[1:6,-1] #extracting basis
colnames(basis)[colnames(basis)=="(Intercept)"]<-"s(time).0"
basis<-basis %>% #pivoting to long format
  pivot_longer(
    cols=starts_with("s")
  )%>%
  arrange(name) #ordering

#length of dataframe to be created: number of knots by number of
   timepoints (minus 1 for the intercept that we won’t plot)
```

```
ln<-6*(length(coef(gm)))

basis_plot<-data.frame(Basis=integer(ln),
                       value_orig=double(ln),
                       time=integer(ln),
                       cof=double(ln)
)

basis_plot$time<-rep(time) #pasting timepoints
basis_plot$Basis<-factor(rep(c(1:5),each=6)) #pasting basis number values
basis_plot$value_orig<-basis$value #pasting basis values
basis_plot$cof<-rep(coef(gm)[1:5],each=6) #pasting coefficients
basis_plot<-basis_plot%>%
  mutate(mod_val=value_orig*cof) #the create the predicted values the
     bases need to be
#multiplied by the coefficients

#creating labeller to change the labels in the basis plots

basis_names<-c(
  ‘1‘="Intercept",
  ‘2‘="1",
  ‘3‘="2",
  ‘4‘="3",
  ‘5‘="4"
)

#calculating the final smooth by aggregating the basis functions

smooth<-basis_plot%>%
  group_by(time)%>%
  summarize(smooth=sum(mod_val))


#original basis
sz<-1
p11<-ggplot(basis_plot,
            aes(x=time,
                y=value_orig,
                colour=as.factor(Basis)
                )
            )+
  geom_line(size=sz,
            show.legend=FALSE)+
  geom_point(size=sz+1,
             show.legend = FALSE)+
  labs(y=’Basis functions’)+
  facet_wrap(~Basis,
             labeller = as_labeller(basis_names)
             )+
  theme_classic()+
  thm


```

```r
#penalized basis
p12<-ggplot(basis_plot,
            aes(x=time,
                y=mod_val,
                colour=as.factor(Basis)
                )
            )+
  geom_line(show.legend = FALSE,
            size=sz)+
  geom_point(show.legend = FALSE,
             size=sz+1)+
  labs(y='Penalized \n basis functions')+
  scale_y_continuous(breaks=seq(-1,1,1))+
  facet_wrap(~Basis,
             labeller=as_labeller(basis_names)
             )+
  theme_classic()+
  thm

#heatmap of the penalization coefficient
x_labels<-c("Intercept","1","2","3","4")
p13<-ggplot(basis_plot,
            aes(x=Basis,
                y=Basis))+
  geom_tile(aes(fill = cof),
            colour = "black") +
    scale_fill_gradient(low = "white",
                        high = "#B50A2AFF")+ #color picked from KikiMedium
  labs(x='Basis',
       y='Basis')+
  scale_x_discrete(labels=x_labels)+
  geom_text(aes(label=round(cof,2)),
            size=7,
            show.legend = FALSE)+
  theme_classic()+
  theme(legend.title = element_blank())

#plotting simulated datapoints and smooth term
p14<-ggplot(data=dat,
            aes(x=time,y=y))+
  geom_point(size=sz+1)+
  scale_color_aaas()+
  labs(y='Simulated \n response')+
  geom_line(data=smooth,
            aes(x=time,
                y=smooth),
            color="#6C581DFF",
            size=sz+1)+
  theme_classic()


#Combining all
b_plot<-p11+p13+p12+p14+plot_annotation(tag_levels='A')&
  theme(
     text=element_text(size=18)
     )
```

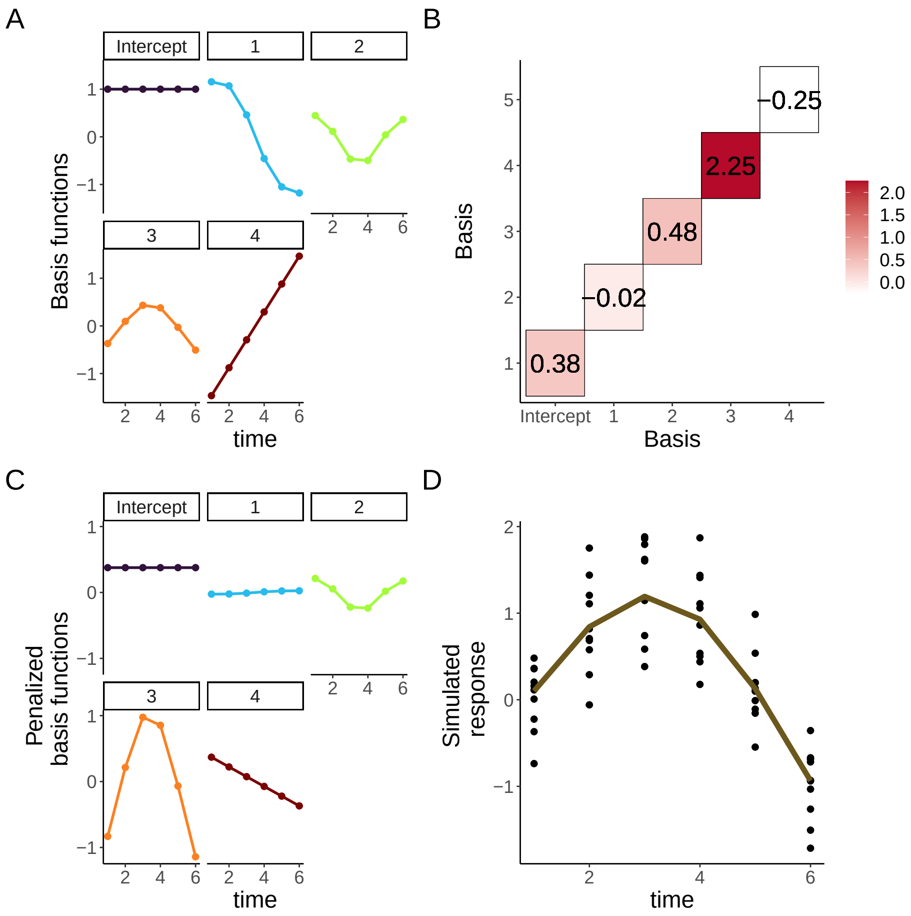


Figure 7: Basis functions for a single smoother for time with five knots. A: Basis functions for a single smoother for time for the simulated data of Group 1 from Figure 2, the intercept basis is not shown. B: Penalty matrix for the basis functions. Each basis function is penalized by a coefficient which can be positive or negative. The coefficient determines the overall effect of each basis in the final smoother. C: Penalized basis functions. Each of the four basis functions of panel A has been penalized by the corresponding coefficient shown in Panel B, note the corresponding increase (or decrease) of each basis. D: Smoother for time and original data points. The smoother (line) is the result of the sum of each penalized basis function at each time point, simulated values for the group appear as points.

B Longitudinal biomedical data simulation and GAMs

This section describes how to fit GAMs to longitudinal data using simulated data. First, data is simulated according to Section 5, where reported data of oxygen saturation (StO_2) in tumors under either chemotherapy or saline control is used as a starting point to generate individual responses in each group.

```
#Dataframe that contains the original reported trends
dat<-tibble(StO2=c(4,27,3,2,0.5,7,4,50,45,56),
            Day=rep(c(0,2,5,7,10),times=2),
            Group=as.factor(rep(c("Control","Treatment"),each=5))
            )
```

```r


## plot the mean response
f1<-ggplot(dat,
           aes(x = Day,
               y = StO2,
               color = Group)) +
    geom_line(size=1,
              show.legend = FALSE)+
    geom_point(show.legend = FALSE,
               size=1.5,
               alpha=0.5)+
  labs(y=expression(paste(StO[2],
                          ’ (real)’)))+
  theme_classic()+
  thm+
    scale_x_continuous(breaks=c(0,5,10))+
    scale_y_continuous(breaks=c(0,40))+
  plot_layout(tag_level = ’new’)+
  theme(
    plot.background = element_rect(fill = "transparent",
                                   color = NA),
    axis.text=element_text(size=14)
  )


#This function simulates data for the tumor data using default parameters of 10 observations per time point,and Standard deviation (sd) of 5%.
#Because physiologically StO2 cannot go below 0%, data is  generated with a cutoff value of 0.0001 (the "StO2_sim")

simulate_data <- function(dat, n = 10, sd = 5) {
    dat_sim <- dat %>%
        slice(rep(1:n(), each = n)) %>%
        group_by(Group, Day) %>%
        mutate(
               StO2_sim = pmax(rnorm(n, StO2, sd), 0.0001),
               subject=rep(1:10),
               subject=factor(paste(subject, Group, sep = "-"))
               ) %>%
        ungroup()

    return(dat_sim)
}


#subject = factor(paste(subject, treatment, sep = "-")))

n <- 10 #number of observations
sd <- 10 #approximate sd from paper
df <- 6
dat_sim <- simulate_data(dat, n, sd)

#plotting simulated data
```

```
f2<-ggplot(dat_sim,
           aes(x = Day,
               y = StO2_sim,
               color = Group)) +
    geom_point(show.legend=FALSE,
               size=1.5,
               alpha=0.5)+
    stat_summary(aes(y = StO2_sim,
                     group=Group),
                 fun=mean, geom="line",
                 size=1,
                 show.legend = FALSE)+
  labs(y=expression(atop(StO[2],
                         ’(simulated)’)))+
  theme_classic()+
  theme(
    axis.text=element_text(size=22)
  )+
  thm+
    scale_x_continuous(breaks=c(0,2,5,7,10))
```

B.1 A basic Workflow for GAMs

This section shows a basic workflow to fit a series of increasingly complex GAMs to the simulated data from the previous section. Graphical and parameter diagnostics for goodness of fit are discussed, as well as model comparison via AIC (Aikake Information Criterion).

B.1.1 First model

The first model fitted to the data is one that only accounts for different smooths by day. The model syntax specifies that `gam_00` is the object that will contain all the model information, and that the model attempts to explain changes in `StO2_sim` (simulated StO_2) using a smooth per `Day`. The model will use 5 knots (`k=5`) for the smooth. And that the smooth is constructed using gaussian process basis (`bs="gp"`). The smoothing parameter estimation method used is the restricted maximum likelihood (`REML`).

```
gam_00<-gam(StO2_sim ~ s(Day, k = 5,bs="gp"),
            method=’REML’,
            data   = dat_sim)
```

To obtain model diagnostics, two methodologies are to be used: 1) graphical diagnostics, and 2) a model check. In the first case, the functions `appraise` and `draw` from the package *gratia* can be used to obtain a single output with all the graphical diagnostics. For model check, the functions `gam.check` and `summary` from *mgcv* provide detailed information about the model fit and its parameters.

```
layout1 <- c(
  area(1, 1),
  area( 1, 2),
  area(2, 1),
  area(2, 2),
  area(1, 3, 2)
)

```

```
layout2 <- c(
  area(1, 1),
  area( 1, 2),
  area(2, 1),
  area(2, 2),
  area(1, 3, 2),
  area(1,4,2),
  area(1,5,1)
)
```

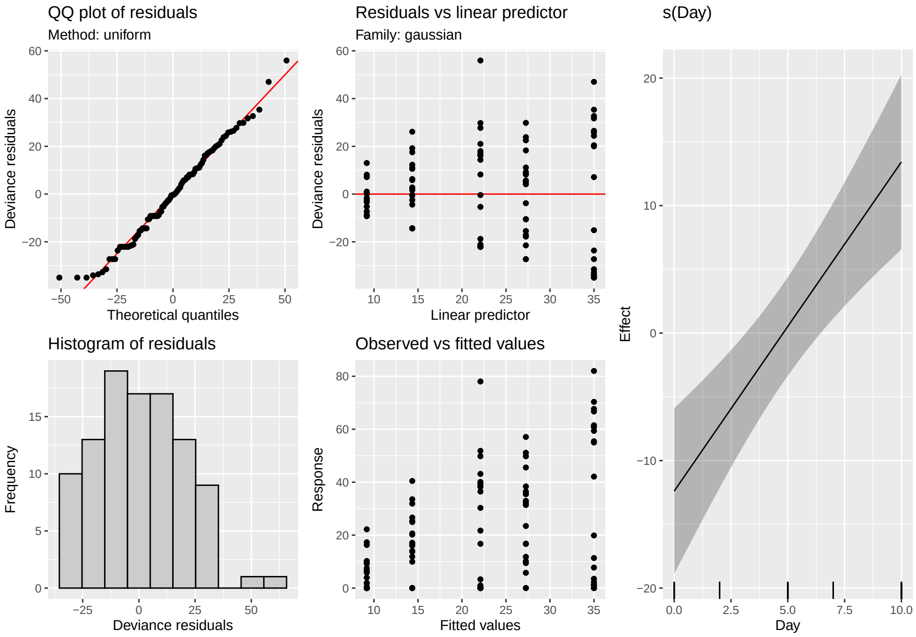


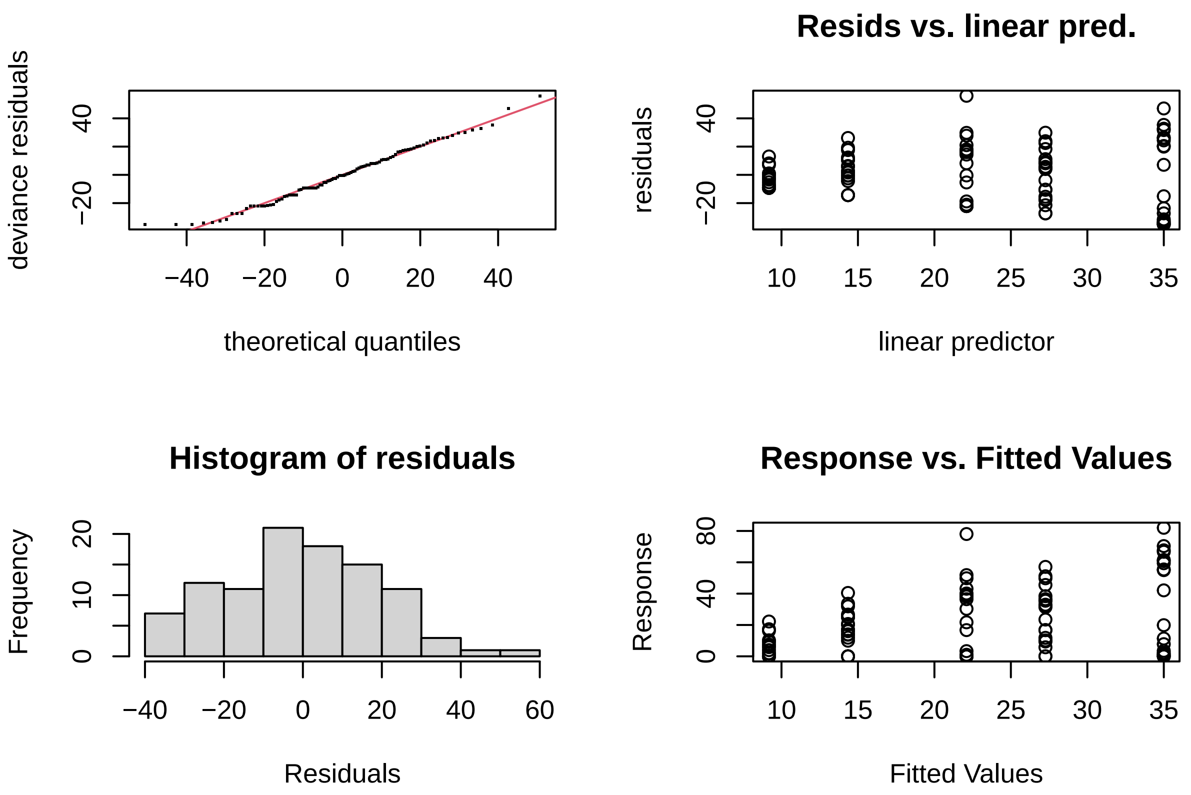
Figure 8: Graphical diagnostics for the first GAM model. Left: Graphical diagnostics provided by the function `appraise` from the package *gratia*. Right: Fitted smooth for the model, provided by the function `draw`.

B.1.1.1 Graphical diagnostics From the output of the function `appraise` in Figure 8, the major indicators of concern about the model are the QQ plot of residuals and the histogram of residuals. The QQ plot shows that the errors are not reasonably located along the 45° line (which indicates normality), as there are multiple points that deviate from the trend, specially in the tails. The histogram also shows that the variation (residuals) is not following the assumption of a normal distribution.

The `draw` function permits to plot the smooths as `ggplot2` objects, which eases subsequent manipulation, if desired. Because model `gam_00` specifies only one smooth for the time covariate (Day), the plot only contains only one smooth. Note that the smooth shows an almost linear profile.

B.1.1.2 Model check

```
#need to add figure number and caption
gam.check(gam_00)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-0.0001585015,0.0008415702]
## (score 436.939 & scale 387.386).
## Hessian positive definite, eigenvalue range [0.0001600441,48.99916].
## Model rank =  5 / 5
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##         k' edf k-index p-value
## s(Day)  4   1    0.37  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(gam_00)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## StO2_sim ~ s(Day, k = 5, bs = "gp")
##
## Parametric coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.578      1.968   10.96   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F  p-value
## s(Day) 1.002  1.004 21.54 1.09e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.172   Deviance explained = 18.1%
## -REML = 436.94  Scale est. = 387.39    n = 100
```

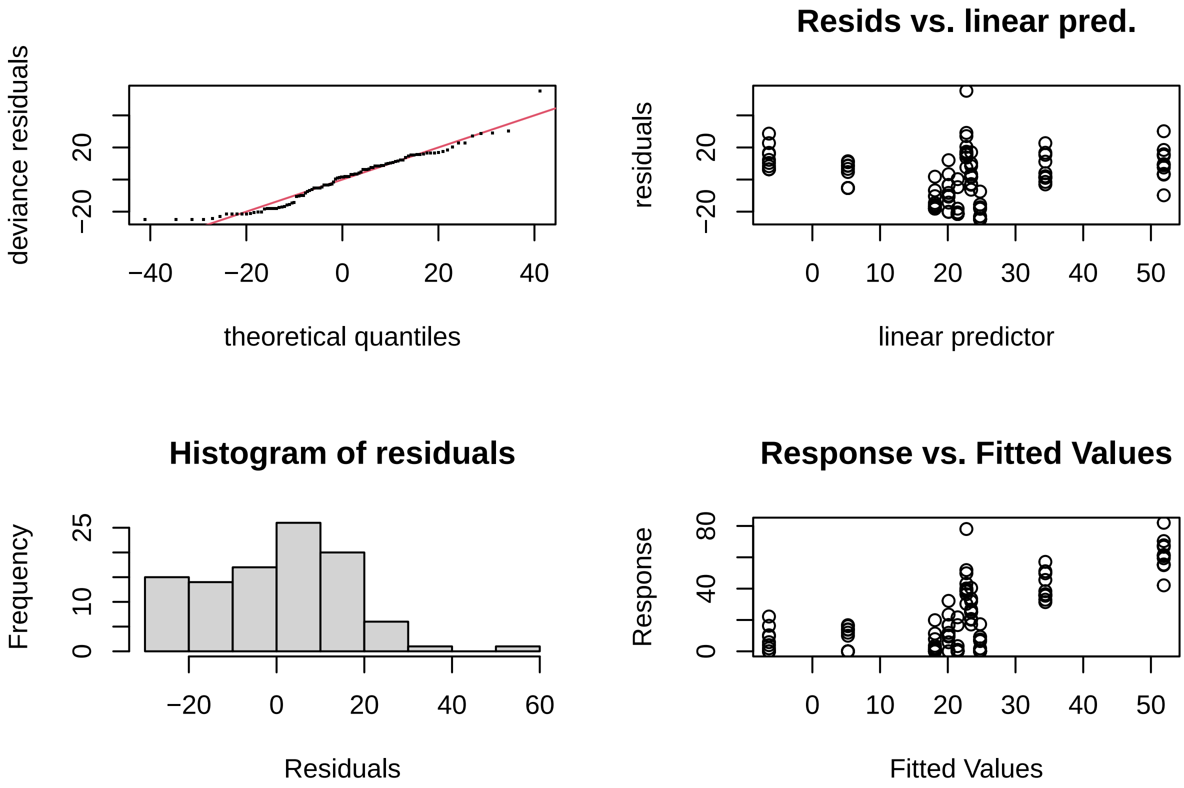
Special attention must be paid to the ‘k-index’ from gam.check. This parameter indicates if the basis dimension of the smooth is adequate, i.e., it checks that the basis used to create the smooth are adequate to capture the trends in the data. If the model is not adequately capturing the trens in the data, this is indicated by a low k-index value (<1). From the output, it can be seen that the k-index is effectively <0.3, which indicates that the model is not capturing the variability in the data. The ‘edf’ (effective degrees of freedom) is an indicator of the complexity of the smooth. Here the complexity of the smooth is comparable to that of a 4th degree polynomial.

From the summary function, information about the assumed distribution of the errors (Gaussian in this case) and the link function can be obtained. The link function is ‘identity’ as the model does not make any transformation on the predictors. The ‘significance of smooth terms’ *p-value* indicates if each smooth is adding significance to the model. Here, the *p-value* is low but we have seen that there are issues with the model from the previous outputs. Finally, the ‘deviance explained’ indicates how much of the data the model is able to capture, which in this case corresponds to ~12%.

B.1.2 Second model

The major flaw of gam_00 is that this model is not taking into account the fact that the data is nested in groups. The next iteration is a model where a different smooth of time (Day) is assigned for each group using by=Group in the model syntax.

```
gam_01<-gam(StO2_sim ~ s(Day, by=Group,k = 5,bs="gp"),
            method=’REML’,
            data  = dat_sim)

gam.check(gam_01)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 8 iterations.
## Gradient range [-0.0001335263,0.001505324]
## (score 415.0769 & scale 254.693).
## Hessian positive definite, eigenvalue range [9.414522e-05,48.49849].
## Model rank =  9 / 9
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k’.
##
##                         k’  edf k-index p-value
## s(Day):GroupControl     4   1    0.46  <2e-16 ***
## s(Day):GroupTreatment   4   1    0.46  <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
summary(gam_01)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## StO2_sim ~ s(Day, by = Group, k = 5, bs = "gp")
##
```

```
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.578      1.596   13.52   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df      F p-value
## s(Day):GroupControl   1.004  1.008  1.083   0.299
## s(Day):GroupTreatment 1.000  1.001 83.768  <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.456   Deviance explained = 46.7%
## -REML = 415.08  Scale est. = 254.69    n = 100
```

Diagnostics for this model indicate that the k-index is still below 1 (0.32 from gam.check), and that the residuals are still not following a normal distribution (Figure 9). Moreover, the smooths (plotted via the draw() function) appear with a fairly linear profile, which indicates they are still not capturing the trends observed in the data. From summary(), the deviance explained by the model is ~43%.

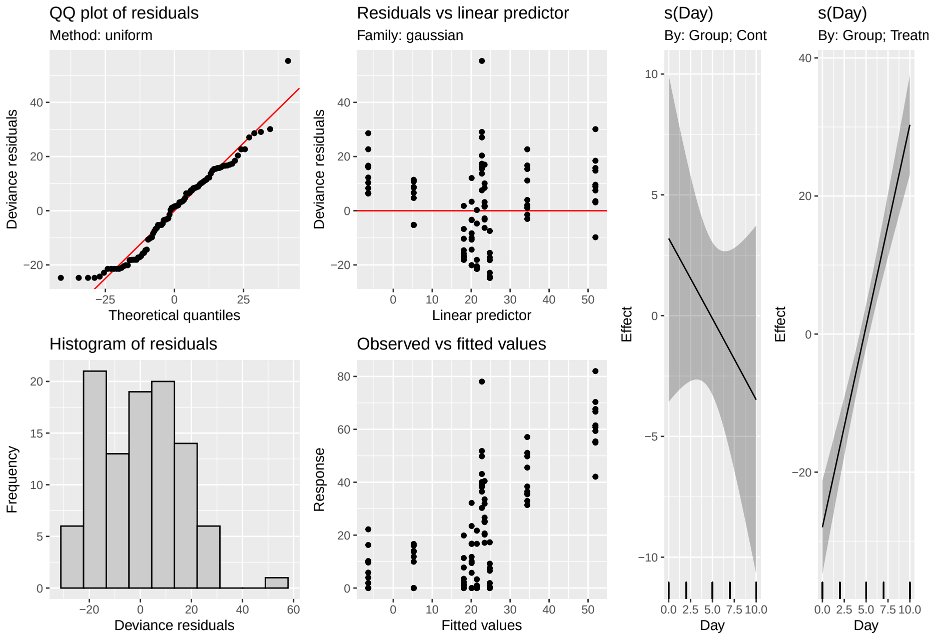
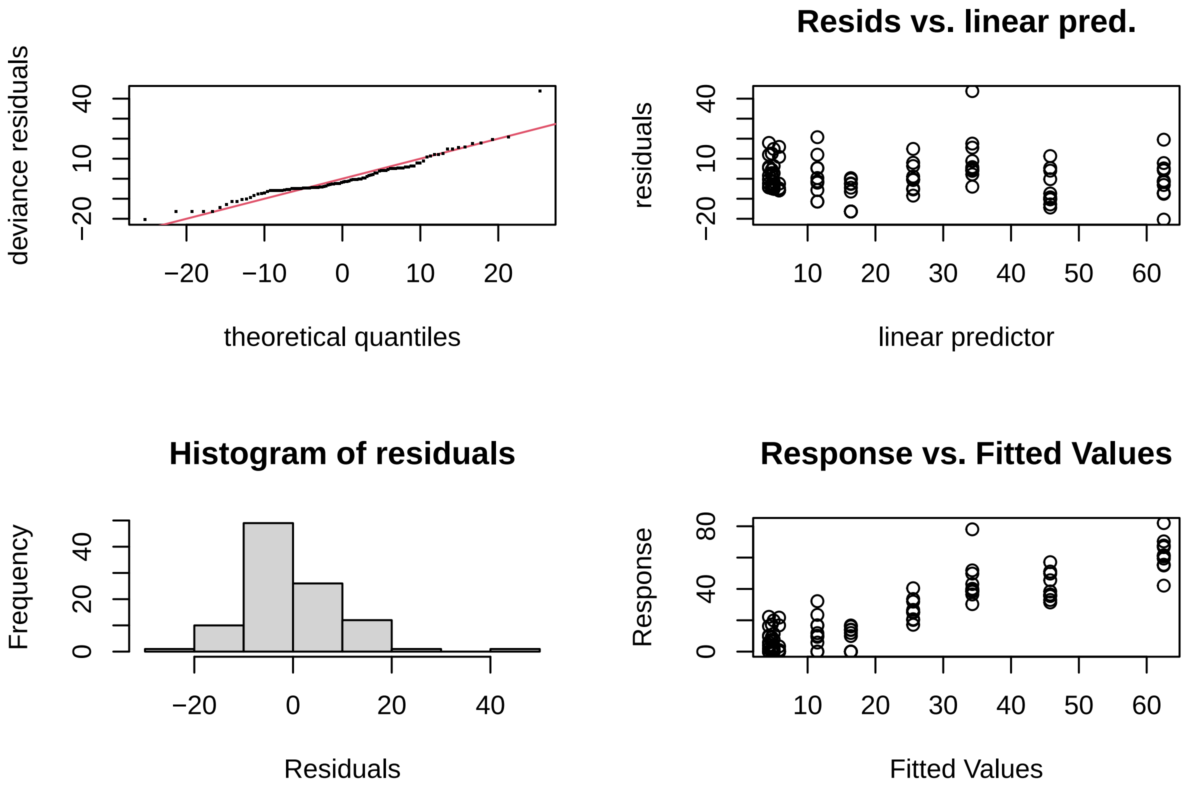


Figure 9: Graphical diagnostics for the second GAM model. Left: Graphical diagnostics provided by the function appraise from the package *gratia*. Right: Fitted smooth for the model, provided by the function draw.

B.1.3 Third model

Model gam_00 was built for didactic purposes to cover the simplest case, but it does not account for the nesting of the data by Group, which is apparent from the type of smooth fitted, the model diagnostics, and, the low variance explained by the model. On the other hand, gam_01 takes into account the nesting within each group and provides better variance explanation, but as indicated in Section 5, in order to differentiate between each group a parametric term needs to be added to the model for the interaction of *Day* and *Group*.

```
#GAM for StO2

gam1 <- gam(StO2_sim ~ Group+s(Day, by = Group, k = 5,bs="gp"),
            method=’REML’,
            data   = dat_sim)

gam.check(gam1)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 9 iterations.
## Gradient range [-1.291119e-06,1.637752e-06]
## (score 374.509 & scale 96.64781).
## Hessian positive definite, eigenvalue range [0.02491842,48.04328].
## Model rank =  10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k’.
##
##                          k’  edf k-index p-value
## s(Day):GroupControl   4.00 3.84    0.85   0.075 .
## s(Day):GroupTreatment 4.00 1.24    0.85   0.085 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
summary(gam1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## StO2_sim ~ Group + s(Day, by = Group, k = 5, bs = "gp")
##
## Parametric coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10.503      1.390   7.554 2.87e-11 ***
## GroupTreatment   22.150      1.966  11.266  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df       F  p-value
## s(Day):GroupControl   3.843  3.976   7.975 8.99e-06 ***
## s(Day):GroupTreatment 1.236  1.419 159.701  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.794   Deviance explained = 80.6%
## -REML = 374.51  Scale est. = 96.648    n = 100
```

The resulting model is model gam1, which is the model fitted in the main manuscript. By using appraise() and draw on this model (Figure 10) we see that the trend on the QQ plot has improved, the histogram of the residuals appears to be reasonably distributed, and the smooths are capturing the trend of the data within each group . From gam.check, the k-index is now at an acceptable value (~1.02), and summary now indicates that the model is able to capture 87% of the variance data.

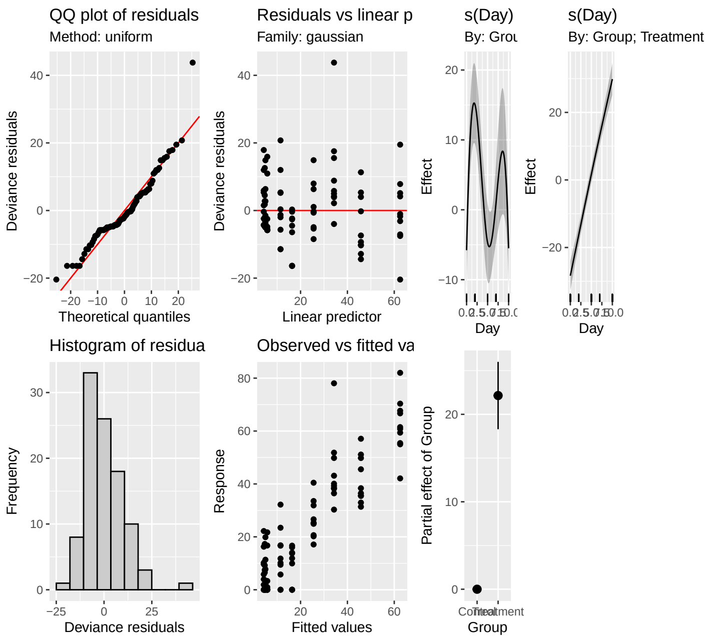


Figure 10: Graphical diagnostics for the final GAM model. Left: Graphical diagnostics provided by the function `appraise` from the package *gratia*. Right: Fitted smooths for the model, provided by the function `draw`.

B.1.4 Comparing models via AIC

One final comparison that can be made for model selection involves the use of the Aikake Information Criterion (AIC). This metric is used to estimate information loss, which we want to minimize with an appropriate model. Therefore, when 2 or more models are compared, the model with lower AIC is preferred. In R, the comparison is done using the `AIC` function.

```
AIC(gam_00,gam_01,gam1)
```

```
##              df      AIC
## gam_00 3.004277 883.7160
## gam_01 4.008467 842.7602
## gam1   8.395441 750.3446
```

The output in this case is expected: model `gam1` has a lower AIC (711.46) whereas the initial two models have higher AICs (891 and 850). The AIC should not be considered as the only estimator of model goodness, instead to be used as complimentary information to the graphical diagnostics and model checks described above.

B.1.4.1 Pairwise comparisons of smooth confidence intervals

The estimation of significant differences between each treatment group can be achieved via pairwise comparisons of the smooth confidence intervals as described in section 5.3. In this case, the “design matrix” is used to estimate the pairwise comparisons (see main manuscript for details and associated references). Briefly, the “design matrix” (also known as the “Xp matrix”) from the selected model (`gam1`) is used to calculate a 95% confidence interval of the difference between the smooth terms for each group. This approach allows to estimate the time intervals where a significant difference exists between the groups (confidence interval above or below 0).

```
##Pairwise comparisons
```

```

##matrix that contains the basis functions evaluated at the points in pdat
    xp <- predict(gam1, newdata = pdat, type = ’lpmatrix’)


#Find columns in xp where the name contains "Control"
    c1 <- grepl(’Control’, colnames(xp))

#Find columns in xp where the name contains ’Treatment’
    c2 <- grepl(’Treatment’, colnames(xp))

#Find rows in pdat that correspond to either ’Control’ or ’Treatment’
    r1 <- with(pdat, Group == ’Control’)
    r2 <- with(pdat, Group == ’Treatment’)

# In xp: find the rows that correspond to Control or Treatment, those that
    do not match will be
    #set to zero. Then, substract the values from the rows corresponding
       to ’Control’ from those that correspond
    #to ’Treatment’
    X <- xp[r1, ] - xp[r2, ]

    ## remove columns that do not contain name ’Control’ or ’Treatment’
    X[, ! (c1 | c2)] <- 0
    ## zero out the parametric cols, those that do not contain in the
       characters ’s(’
    X[, !grepl(’^s\\(’, colnames(xp))] <- 0

    #Multiply matrix by model coefficients. X has (p,n) (rows, columns)
       and the coefficient matrix has
    #dimensions (n,1). The resulting matrix has dimensions (p,1)
    dif <- X %*% coef(gam1)

    #comp<-test %*% coef(gam1)[3:10]

#Calculate standard error for the computed differences using the variance-
   covariance matrix
    #of the model
    se <- sqrt(rowSums((X %*% vcov(gam1, unconditional = FALSE)) * X))
    crit <- qt(0.05/2, df.residual(gam1), lower.tail = FALSE)
    #upper  limits
    upr <- dif + (crit * se)
    #lower limits
    lwr <- dif - (crit * se)
    #put all components in a dataframe for plotting
    comp1<-data.frame(pair = paste(’Control’, ’Treatment’, sep = ’-’),
               diff = dif,
               se = se,
               upper = upr,
               lower = lwr)



#add time point sequence
```

```r
comp_StO2 <- cbind(Day = seq(0, 10, length = 400),
                   rbind(comp1))

#plot the difference
c1<-ggplot(comp_StO2, aes(x = Day, y = diff, group = pair)) +
  #ribbon for difference confidence interval
  geom_ribbon(aes(ymin = lower, ymax = upper),
                alpha = 0.5,
                fill='#DB3A07FF') +
    geom_line(color='black',size=1) +
    geom_line(data=comp_StO2,aes(y=0),size=0.5)+
  #highlight area under the curve where "Control" is higher
  geom_ribbon(data=comp_StO2%>%
                    filter(lower>0),
                aes(ymin =0, ymax =lower),
                alpha = 0.5,
                fill='#30123BFF') +
  #highlight area under the curve where "Treatment" is higher
  geom_ribbon(data=comp_StO2 %>%
                    filter(upper<0),
                    aes(ymin =0, ymax =upper),
                alpha = 0.5,
                fill='#7A0403FF') +
    facet_wrap(~ pair) +
    theme_classic()+
    labs(x = 'Days', y = expression(paste('Difference in StO'[2] )))+
    scale_x_continuous(breaks=c(0,2,5,7,10))+
    theme(
        text=element_text(size=18),
        legend.title=element_blank()
    )
```

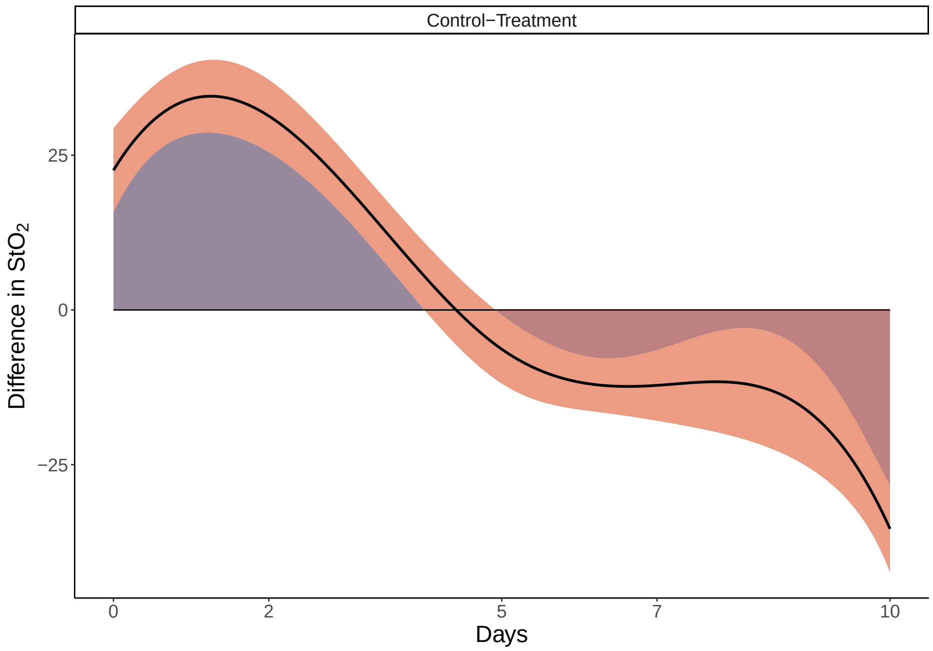


Figure 11: Smooth pairwise comparisons for model gam1 using a 95% confidence interval for the difference between smooths.

Of notice, a convenient wrapper for the function described above exists in the package gratia. In this package, difference_smooths is a function that makes the comparisons and produces Figure 11 when is used on a fitted model. The function syntax and an example can be found at:

https://cran.r-project.org/web/packages/gratia/gratia.pdf

C GAM and Linear model plots and Missing data

This section covers the code used to generate Figure 3, where the simulated data, fit of the "final" GAM (gam1), linear model and GAM on data with missing observations are presented. Note that panel A in Figure 3 and the inlet are generated in the code chunk where the data is simulated in Section B, and are called later to build the figure.

C.1 GAM and Linear model plots

This code chunk creates panels B and D in Figure 3. Note that this code uses the final GAM from the previous section (gam1), so the simulated data and the model should be generated before running this section.

```
#linear model
lm1<-lm(StO2_sim ~ Day + Group + Day * Group, data = dat_sim)


#creates a dataframe using the length of the covariates for the GAM
gam_predict <- expand_grid(Group = factor(c("Control", "Treatment")),
                         Day = seq(0, 10, by = 0.1),
                         subject=factor(rep(1:10)))

#creates a dataframe using the length of the covariates for rm-ANOVA
```

```
lm_predict<-expand_grid(Group = factor(c("Control", "Treatment")),
                         Day = c(0:10),
                        subject=factor(rep(1:10)),
                          )
lm_predict$subject<-factor(paste(lm_predict$subject, lm_predict$Group, sep
    = "-"))

#adds the predictions to the grid and creates a confidence interval for
   GAM
gam_predict<-gam_predict%>%
    mutate(fit = predict(gam1,gam_predict,se.fit = TRUE,type='response')$
       fit,
           se.fit = predict(gam1, gam_predict,se.fit = TRUE,type='response
              ')$se.fit)

#using lm
lm_predict<-lm_predict%>%
    mutate(fit = predict(lm1,lm_predict,se.fit = TRUE,type='response')$fit
       ,
           se.fit = predict(lm1, lm_predict,se.fit = TRUE,type='response')
              $se.fit)

#plot smooths and confidence interval for GAM
f3<-ggplot(data=dat_sim, aes(x=Day, y=StO2_sim, group=Group)) +
    geom_point(aes(color=Group),size=1.5,alpha=0.5,show.legend = FALSE)+
  geom_ribbon(aes( x=Day,ymin=(fit - 2*se.fit),
                   ymax=(fit + 2*se.fit),
                   fill=Group
                   ),
              alpha=0.3,
              data=gam_predict,
            show.legend=FALSE,
                inherit.aes=FALSE) +
  geom_line(aes(y=fit,
                color=Group),
              size=1,data=gam_predict,
              show.legend = FALSE)+
  #facet_wrap(~Group)+
  labs(y=expression(atop(StO[2],'complete')))+
    scale_x_continuous(breaks=c(0,2,5,7,10))+
      theme_classic()+
  theme(
    axis.text=element_text(size=22)
  )+
      thm+
  thm1

#plot linear fit for rm-ANOVA
f4<-ggplot(data=dat_sim, aes(x=Day, y=StO2_sim, group=Group)) +
    geom_point(aes(color=Group),size=1.5,alpha=0.5,show.legend = FALSE)+
  geom_ribbon(aes( x=Day,ymin=(fit - 2*se.fit),
                   ymax=(fit + 2*se.fit),fill=Group),
              alpha=0.3,
              data=lm_predict,
              show.legend = FALSE,
                inherit.aes=FALSE) +
  geom_line(aes(y=fit,
                color=Group),
              size=1,data=lm_predict,
              show.legend = FALSE)+
  #facet_wrap(~Group)+
  labs(y=expression(paste(’StO’[2],’ (simulated)’)))+
    scale_x_continuous(breaks=c(0,2,5,7,10))+
      theme_classic()+
  theme(
    axis.text=element_text(size=22)
  )+
      thm+
  thm1



#posthoc comparisons for the linear model
#library(multcomp)


#summary(glht(lm1, linfct = mcp(Group = ’Tukey’)))
#summary(glht(lm1, linfct=mcp(Group="Tukey", interaction_average=TRUE)))
```

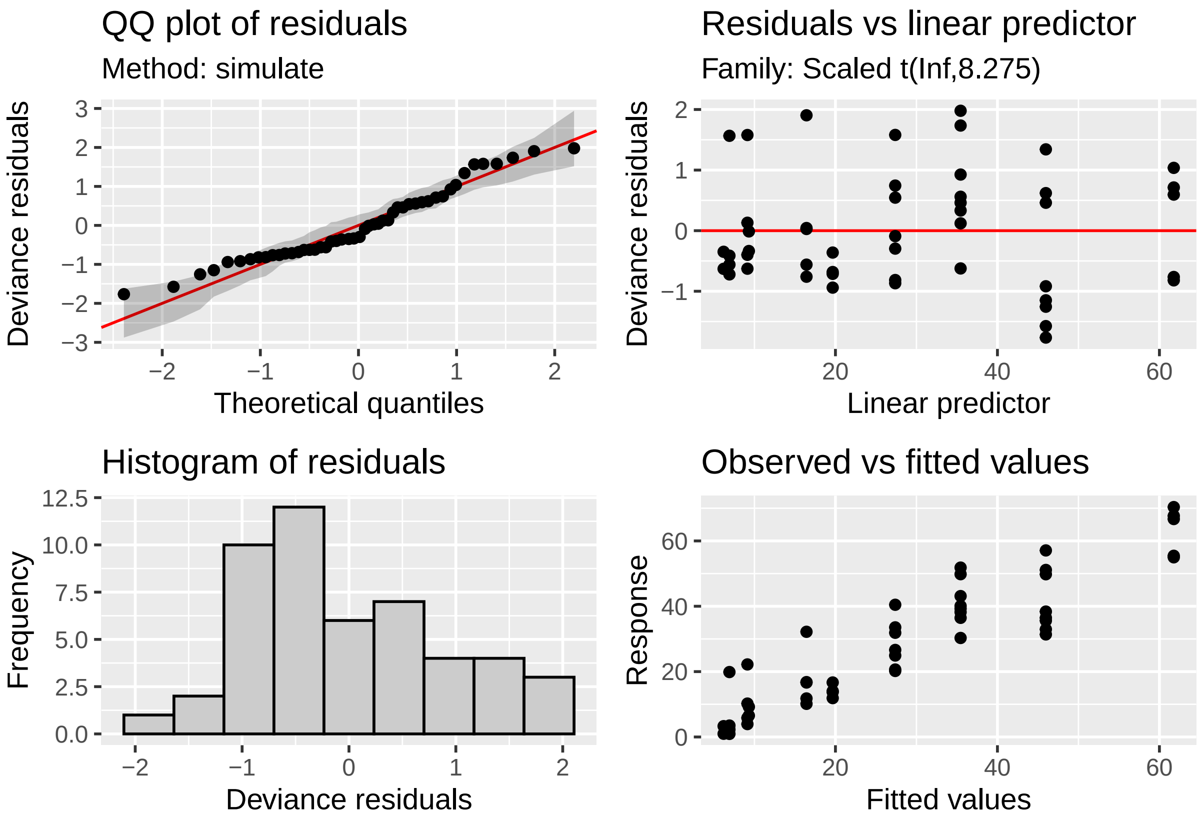
C.2 Working with Missing data in GAMs

This code chunk first randomly deletes 40% of the total observations in the original simulated data, and then an interaction GAM is fitted to the remaining data. Model diagnostics are presented, and an object that stores the fitted smooths is saved to be called in the final code chunk to build the figure.

```
#missing data
#create a sequence of 40 random numbers between 1 and 100, these numbers
   will
#correspond to the row numbers to be randomly erased from the original
   dataset
missing <- sample(1:100, 40)

#create a new dataframe from the simulated data with 40 rows randomly
   removed, keep the missing values as NA

ind <- which(dat_sim$StO2_sim %in% sample(dat_sim$StO2_sim, 40))

#create a new dataframe, remove the StO2 column
dat_missing <- dat_sim[,-1]

#add NAs at the ind positions
dat_missing$StO2_sim[ind]<-NA

#Count the number of remaining observations per day (original dataset had
   10 per group per day)
dat_missing %>%
    group_by(Day,Group) %>%
    filter(!is.na(StO2_sim))%>%
  count(Day)
```

```
## # A tibble: 10 x 3
## # Groups:   Day, Group [10]
##       Day Group         n
##     <dbl> <fct>     <int>
##   1     0 Control       2
##   2     0 Treatment     4
##   3     2 Control       7
##   4     2 Treatment     4
##   5     5 Control       2
##   6     5 Treatment     8
##   7     7 Control       5
##   8     7 Treatment     8
##   9    10 Control       4
##  10    10 Treatment     5
```

```
#the same model used for the full dataset
mod_m1 <- gam(StO2_sim ~ Group+s(Day,by=Group,k=5), data  = dat_missing,
      family=scat)
#appraise the model
appraise(mod_m1)
```



```
m_predict <- expand_grid(Group = factor(c("Control", "Treatment")),
                         Day = seq(0, 10, by = 0.1))

#adds the predictions to the grid and creates a confidence interval
m_predict<-m_predict%>%
    mutate(fit = predict(mod_m1,m_predict,se.fit = TRUE,type='response')$
       fit,
           se.fit = predict(mod_m1, m_predict,se.fit = TRUE,type='response
              ')$se.fit)


f6<-ggplot(data=dat_missing, aes(x=Day, y=StO2_sim, group=Group)) +
    geom_point(aes(color=Group),size=1.5,alpha=0.5,show.legend = FALSE)+
  geom_ribbon(aes( x=Day,ymin=(fit - 2*se.fit),
                   ymax=(fit + 2*se.fit),
                   fill=Group
                   ),
              alpha=0.3,
              data=m_predict,
            show.legend=FALSE,
                inherit.aes=FALSE) +
  geom_line(aes(y=fit,
                color=Group),
              size=1,data=m_predict,
              show.legend = TRUE)+
  #facet_wrap(~Group)+
  labs(y=expression(atop(StO[2],'missing')))+
    scale_x_continuous(breaks=c(0,2,5,7,10))+
      theme_classic()+
  theme(
    axis.text=element_text(size=22)
  )+
      thm+
  thm1
```

```
mult_plot<-f2+inset_element(
  f1, left = 0.01,
  bottom = 0.5,
  right = 0.5,
  top = 1.0)+
  f3+f4+f6+
   plot_annotation(tag_levels='A')&
   ylim(c(-5,75)) &
  theme(
     text=element_text(size=18)
     )&
  thm

mult_plot
```

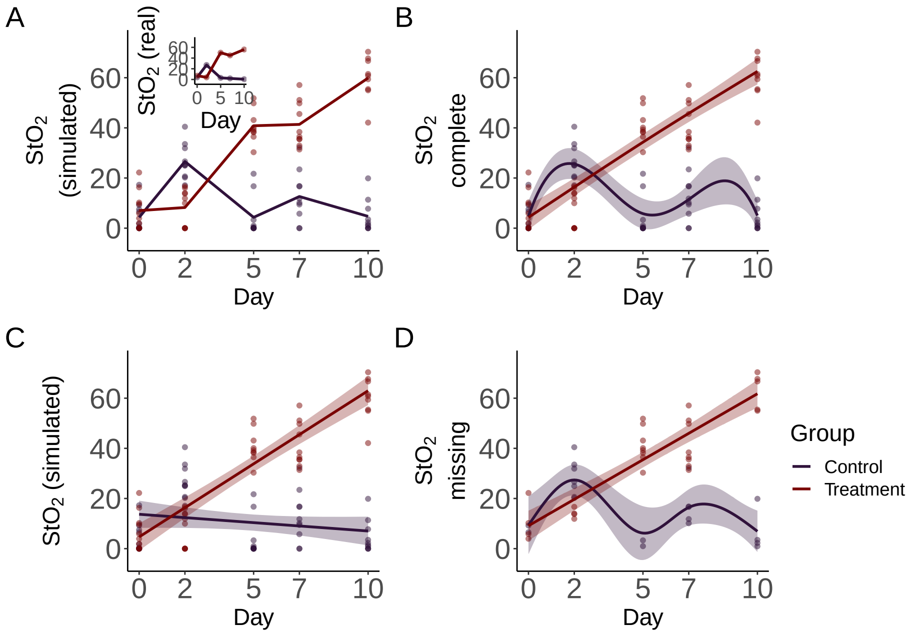


Figure 12: Simulated data and smooths for oxygen saturation in tumors. A: Simulated data that follows previously reported trends (inset) in tumors under chemotherapy (Treatment) or saline (Control) treatment. Simulated data is from a normal distribution with standard deviation of 10% with 10 observations per time point. Lines indicate mean oxygen saturation B: Smooths from the GAM model for the full simulated data with interaction of Group and Treatment. Lines represent trends for each group, shaded regions are 95% confidence intervals. C: rm-ANOVA model for the simulated data, the model does not capture the changes in each group over time. D: Smooths for the GAM model for the simulated data with 40% of its observations missing. Lines represent trends for each group, shaded regions are 95% confidence intervals.

C.3 Pairwise comparisons in GAMs: full and missing data cases

The next code chunk reproduces Figure 4. Here pairwise comparisons are made for the full and missing datasets.

```
##Pairwise comparisons

pdat <- expand.grid(Day = seq(0, 10, length = 400),
                    Group = c(’Control’, ’Treatment’))

#this function takes the model, grid and groups to be compared using the
   lpmatrix

smooth_diff <- function(model, newdata, g1, g2, alpha = 0.05,
                        unconditional = FALSE) {
    xp <- predict(model, newdata = newdata, type = ’lpmatrix’)
    #Find columns in xp where the name contains "Control"
    col1 <- grepl(g1, colnames(xp))
    #Find columns in xp where the name contains ’Treatment’
    col2 <- grepl(g2, colnames(xp))
    #r1 <- newdata[[var]] == f1
    #r2 <- newdata[[var]] == f2
    row1 <- with(newdata, Group == g1)
    row2 <- with(newdata, Group == g2)
    ## difference rows of xp for data from comparison
    X <- xp[row1, ] - xp[row2, ]
    ## zero out cols of X related to splines for other lochs
    X[, ! (col1 | col2)] <- 0
    ## zero out the parametric cols
    X[, !grepl(’ˆs\\(’, colnames(xp))] <- 0
    dif <- X %*% coef(model)
    se <- sqrt(rowSums((X %*% vcov(model, unconditional = unconditional))
       * X))
    crit <- qt(alpha/2, df.residual(model), lower.tail = FALSE)
    upr <- dif + (crit * se)
    lwr <- dif - (crit * se)
    data.frame(pair = paste(g1, g2, sep = ’-’),
               diff = dif,
               se = se,
               upper = upr,
               lower = lwr)
}

comp1<-smooth_diff(gam1,pdat,’Control’,’Treatment’)

comp_StO2_full <- cbind(Day = seq(0, 10, length = 400),
                   rbind(comp1)) %>%
  mutate(interval=case_when(
    upper>0 & lower<0~"no-diff",
    upper<0~"less",
    lower>0~"greater"
  ))

c1<-ggplot(comp_StO2_full, aes(x = Day, y = diff, group = pair)) +
    geom_ribbon(aes(ymin = lower, ymax = upper),
                alpha = 0.5,
                fill=’#DB3A07FF’) +
    geom_line(color=’#E75B64FF’,size=1) +
    geom_line(data=comp_StO2_full,aes(y=0),size=0.5)+
    geom_ribbon(data=comp_StO2_full%>%
                    filter(lower>0),
                aes(ymin =0, ymax =lower),
                alpha = 0.5,
                fill=’#30123BFF’) +
    geom_ribbon(data=comp_StO2_full %>%
                    filter(upper<0),
                    aes(ymin =0, ymax =upper),
                alpha = 0.5,
                fill=’#7A0403FF’) +
    facet_wrap(~ pair) +
    theme_classic()+
    labs(x = ’Days’, y = expression(paste(’Difference in StO’[2] )))+
    scale_x_continuous(breaks=c(0,2,5,7,10))+
    theme(
        text=element_text(size=18),
        legend.title=element_blank()
    )
```

```
###for missing data
comp2<-smooth_diff(mod_m1,pdat,'Control','Treatment')
comp_StO2_missing <- cbind(Day = seq(0, 10, length = 400),
                   rbind(comp2))

missing_plot<-ggplot(comp_StO2_missing, aes(x = Day, y = diff, group =
   pair)) +
    geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +
    geom_line(color='black',size=1) +
    facet_wrap(~ pair) +
    labs(x = 'Days',
         y = expression(paste('Difference in StO'[2],'\n (missing data)'
                              )))+
  scale_x_continuous(breaks=c(0,2,5,7,10))+
  theme_classic()+
  theme(
     text=element_text(size=18),
     legend.title=element_blank()
     )

c2<-ggplot(comp_StO2_missing, aes(x = Day, y = diff, group = pair)) +
    geom_ribbon(aes(ymin = lower, ymax = upper),
                alpha = 0.5,
                fill='#DB3A07FF') +
    geom_line(color='#E75B64FF',size=1) +
    geom_line(data=comp_StO2_missing,aes(y=0),size=0.5)+
    geom_ribbon(data=comp_StO2_missing%>%
                    filter(lower>0),
                aes(ymin =0, ymax =lower),
                alpha = 0.5,
                fill='#30123BFF') +
    geom_ribbon(data=comp_StO2_missing %>%
                    filter(upper<0),
                    aes(ymin =0, ymax =upper),
                alpha = 0.5,
                fill='#7A0403FF') +
    facet_wrap(~ pair) +
    theme_classic()+
    labs(x = 'Days', y = expression(paste('Difference in StO'[2] )))+
    scale_x_continuous(breaks=c(0,2,5,7,10))+
    theme(
        text=element_text(size=18),
        legend.title=element_blank()
    )

pair_comp<-c1+c2
```

Smooth pairwise comparisons for model gam1 using a 95% confidence interval for the difference between smooths. Pairwise comparisons for smooth terms. A: Pairwise comparisons for the full dataset. B: Pairwise comparisons for the dataset with missing observations. Significant differences exist where the interval does not cover 0. In both cases the effect of treatment is significant after day 5.

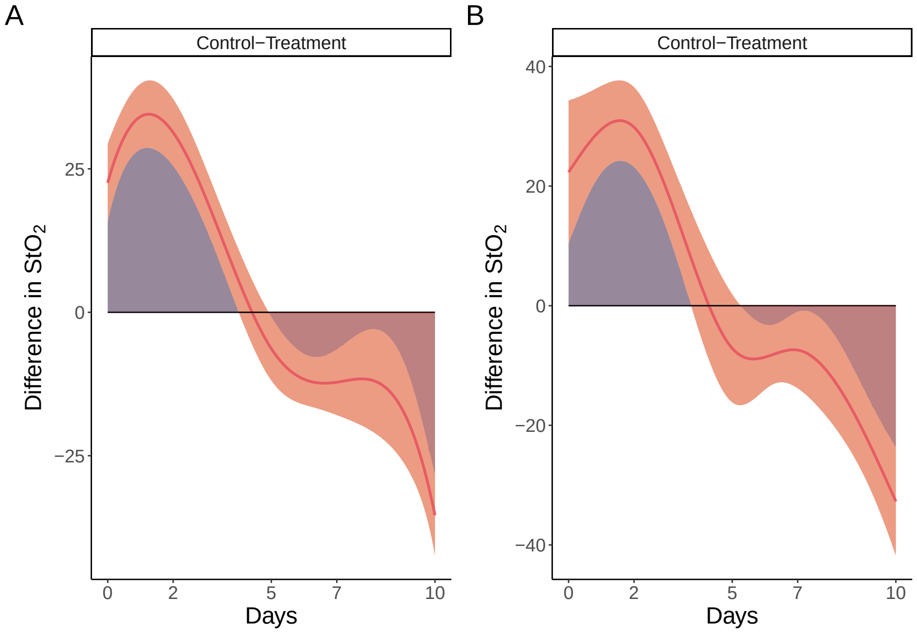


Figure 13: Smooth pairwise comparisons for model gam1 using a 95% confidence interval for the difference between smooths.