

# Navigating Spatio-Temporal Data with GAMs

James Bailie

September 30, 2021

In ordinary least squares (OLS) regression, we assume the mean  $\mu_i$  of the response  $y_i$  is linearly related to the covariates:  $y_i = \mathbf{x}_i\boldsymbol{\beta}$ . In generalised linear models (GLMs), we allow  $\mu_i$  to depend on the linear predictor  $\eta_i = \mathbf{x}_i\boldsymbol{\beta}$  through a known transformation  $g$ . Unfortunately, we often need further flexibility. For example, the effects of time-of-year frequently exhibit non-linear behaviour with seasonal patterns. Spatial data can also present problems: the relationship between latitude, longitude and, for example, annual rainfall tends to be complex. In these settings, the effects of  $\mathbf{x}_i$  on  $g(\mu_i)$  are non-linear and hence cannot be modelled by GLMs.

Spatio-temporal data abounds in the real world; as just a few examples, consider infectious disease mapping, stock exchange forecasting, modelling neural activity patterns in brain imaging data, and assessing ecological biodiversity. If systematic spatio-temporal variation is not captured by the model, then this variation will be present in the errors  $\boldsymbol{\epsilon}$ . As such,  $\epsilon_1, \dots, \epsilon_n$  will not be independent and any model inference will be invalid. One approach to solving this problem includes quadratic, cubic or higher-order terms in the design matrix  $\mathbf{X}$ . However, this approach is ad-hoc and can easily result in overfitting. It is also

difficult to handle non-linear interactions between variables (such as longitude and latitude in spatial data).

Another approach is to expand the model class to include non-linear functions. Here we replace the GLM model specification  $g(\mu_i) = \sum_{j=1}^p x_{ij}\beta_j$  with

$$g(\mu_i) = \sum_{j=1}^p f_j(x_{ij}), \quad (1)$$

where  $f_j$  are smoothing functions (e.g. piecewise polynomials). While we no longer require linearity, the covariate effects are still additive; hence equation (1) is called a generalised additive model (GAM) [Hastie and Tibshirani, 1986, 1987].

As with GLMs, GAMs assume that the responses  $y_i$  follow a known exponential family, or that the variance function  $V(\cdot)$  is known so that a quasi-likelihood can be determined. GAMs also assume that  $\mu_i$  relates to  $\eta_i$  through a known link  $g$ . The key difference is that  $\eta_i$  can include a large class of smooth – not just linear – functions  $f_j$  of the covariates  $\mathbf{x}_i$ .

Maximum quasi-likelihood estimation is used to fit the model. Similar to regularisation techniques (e.g. lasso), deviance is minimised subject to constraining a smoothness penalty (more on this in a later section).

While GAMs provided much needed flexibility as compared to GLMs, they are not without their disadvantages. GLMs are considerably more interpretable, since effects are linearly related to  $g(\mu_i)$ . It is also harder to do inference using GAMs.

This report aims to be a short primer on GAMs, with a specific focus on modelling spatio-temporal data. I will start by developing some of the theory behind GAMs; while the second half of the report will be application focused.

## Splines

Before we can fit GAMs, we need to understand what the smoothing functions  $f_j$  are. In this section, I will introduce one class of smoothing functions, the natural cubic splines. As we shall see later, there are other classes of smoothers.

In OLS, we are accustomed to using polynomials to approximate non-linear relationships. This approach is justified by Taylor’s theorem: locally, smooth functions are well-approximated by polynomials. However, we know from experience that polynomials can interpolate data yet oscillate widely – and unrealistically – in between data points. Figure 1 gives one example of this behaviour.

This motivates the definition of natural cubic splines: given a set of points  $\{x_k : k = 1, \dots, m\}$  with  $x_k < x_{k+1}$ , a cubic spline is a twice continuously differentiable function consisting of cubic polynomials on  $[x_k, x_{k+1}]$ . Further, a cubic spline  $f$  is natural if  $f''(x_1) = f''(x_m) = 0$ ; this constraint discourages extreme oscillation at either end of the data, prohibiting the large swings seen in Figure 1. The points  $\{x_k\}$  where two polynomials meet are called knots. The analogy here is that the knots ‘tie’ the piecewise polynomials together, since the polynomials are forced to agree at the knots.

Natural cubic splines are the smoothest interpolators\*, in the sense that they minimise the  $L^2$  norm of second derivatives, amongst absolutely continuous differentiable interpolators [Green and Silverman, 1994; Schoenberg, 1964]. (Intuitively, the naturality condition  $f''(x_1) = f''(x_m) = 0$  enforces smoothness at the end points, while the cubic condition enforces smoothness between the knots – in the sense that  $f'''(x)$  is constant between the

---

\*A function  $f$  interpolates a set of points  $\{(x_i, y_i) : i = 1, \dots, m\}$  if  $f(x_i) = y_i$  for all  $i = 1, \dots, m$ .

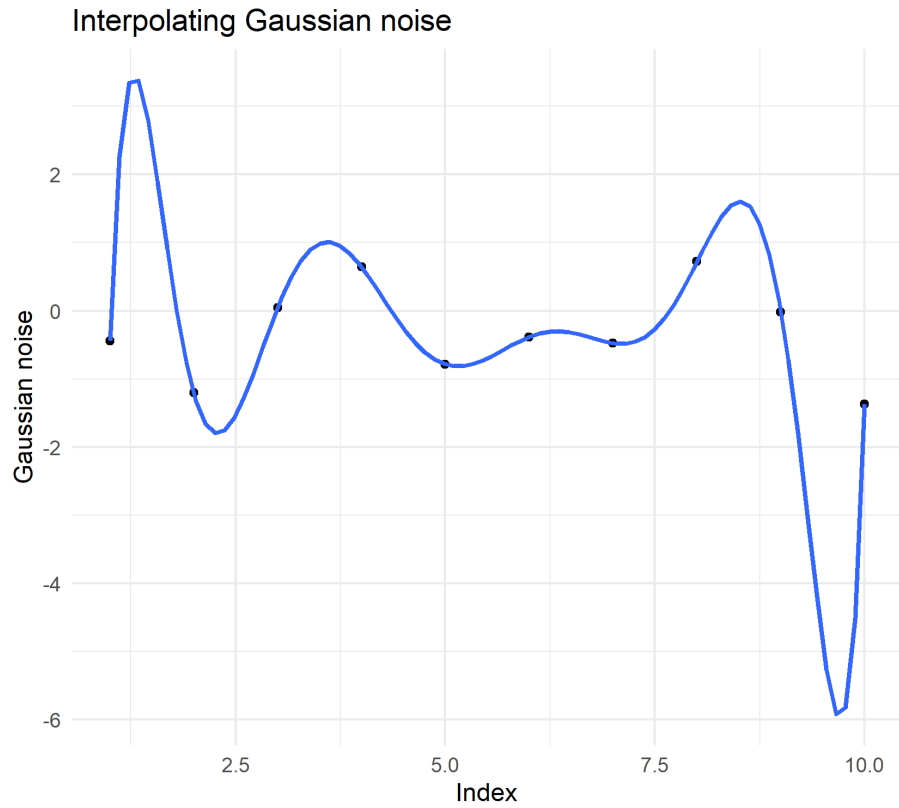


Figure 1: Standard Normal random noise (black dots,  $n = 10$ ) with a polynomial (degree 9) interpolator (blue line). The polynomial interpolator is an unrealistic model for the data, since it is prone to extreme behaviour in between data points.

knots.) This makes natural cubic splines a clear choice for our smoothing functions  $f_j$ .

However, how can we model splines based on noisy data?

## Modelling splines and fitting GAMs – basis functions and penalised regression

The polynomials of order  $d$  form a  $d$ -dimensional vector space, with basis  $\{1, x, x^2, \dots, x^d\}$ .

Similarly, the natural cubic splines on the knots  $\{x_k\}_{k=1}^m$  form a  $m$ -dimensional vector space, with basis functions  $b_1, \dots, b_m$ . (Due to space reasons, we will omit the complicated definition of  $b_k$ , see [Wood, 2017, section 5.3.1] for details.) This means we can write any natural cubic spline as  $f(x) = \sum_{k=1}^m b_k(x)\beta_k$ .

We have now effectively turned a non-parametric model consisting of unknown functions  $f_j$  into a parametric model with unknown constants  $\beta_k$ . Moreover, we can write our smoothness penalty  $\int_{x_1}^{x_m} f''(x)^2 dx$  as a quadratic form  $\boldsymbol{\beta}^\top \mathbf{S} \boldsymbol{\beta}$  (see Lancaster and Šalkauskas [1986] for a proof), where  $\mathbf{S}$  is a fixed and known “wiggleness penalty coefficient” matrix.

Now we are set up to describe the model fitting process as penalised regression. For now, suppose there is a single covariate  $x_i$  with smoother  $f(x) = \sum_{k=1}^m b_k(x)\beta_k$ . Defining  $x_{ik} = b_k(x_i)$  gives a GLM:

$$g(\mu_i) = \mathbf{x}_i \boldsymbol{\beta},$$

which is estimated by maximising the penalised quasi-likelihood:

$$l_p(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \lambda \boldsymbol{\beta}^\top \mathbf{S} \boldsymbol{\beta}.$$

The degree of wiggleness  $\boldsymbol{\beta}^\top \mathbf{S} \boldsymbol{\beta}$  penalises overfitting in the model and the smoothing parameter  $\lambda$  controls the tradeoff between model goodness-of-fit and model smoothness<sup>†</sup>.

---

<sup>†</sup>An astute reader may worry that the questions of the number of knots  $m$  and their values  $x_1, \dots, x_m$  has

When there is more than one covariate, write  $f_j = \sum_{k=1}^m b_{kj}(x)\beta_{kj}$ ; define  $\mathbf{x}_i$  as the (row) vector of the  $b_{kj}(x_i)$ 's and let  $\boldsymbol{\beta}$  be the (column) vector of the  $\beta_{kj}$ 's. As before, we obtain a GLM  $g(\mu_i) = \mathbf{x}_i\boldsymbol{\beta}$ , yet now our smoothness penalty is the sum  $\sum_j \lambda_j \boldsymbol{\beta}^\top \mathbf{S}_j \boldsymbol{\beta}$  of each  $f_j$ 's wiggleness. We would also often like to include linear terms (such as an intercept) in our model matrix  $\mathbf{X}$ . Model identifiability issues (always) arise here due to over-specification; this can be resolved by adding constraints  $\sum_i f_j(x_{ij}) = 0$  and reparametrising; see [Wood, 2017, section 5.4.1] for details.

As in regularisation, the penalty parameters  $\lambda_j$  are usually unknown and must be estimated, using cross validation for example. In practice, more advanced techniques – generalised cross validation (GCV) or restricted maximum likelihood (REML) – are used. Again see [Wood, 2017, section 6.2] for details.

## Warming Up: Historical Australian Annual Temperature

The historical Australian annual mean temperature [Australian Bureau of Meteorology, 2020] is plotted in Figure 2. The trend is roughly flat until 1960, at which point the annual temperature steadily increases. While a quadratic or cubic OLS model may fit the data, a GAM will prove to be a more useful approach. Using the `mgcv` package in `R`, we fit a GAM with a natural cubic spline and knots spread evenly across the range of covariate values. The fitted model is shown in Figure 3. Model diagnostic plots are given in Figure 4; these can not be addressed. How are these parameters chosen and do they affect the model analysis? Fortunately, as long as  $m$  is sufficiently large, the effects are minimal [Clark, 2019]. Knots are typically spaced evenly across the range – or the quantiles – of the data. Other classes of smoothers which we will see later do not use knots and so avoid this issue entirely.

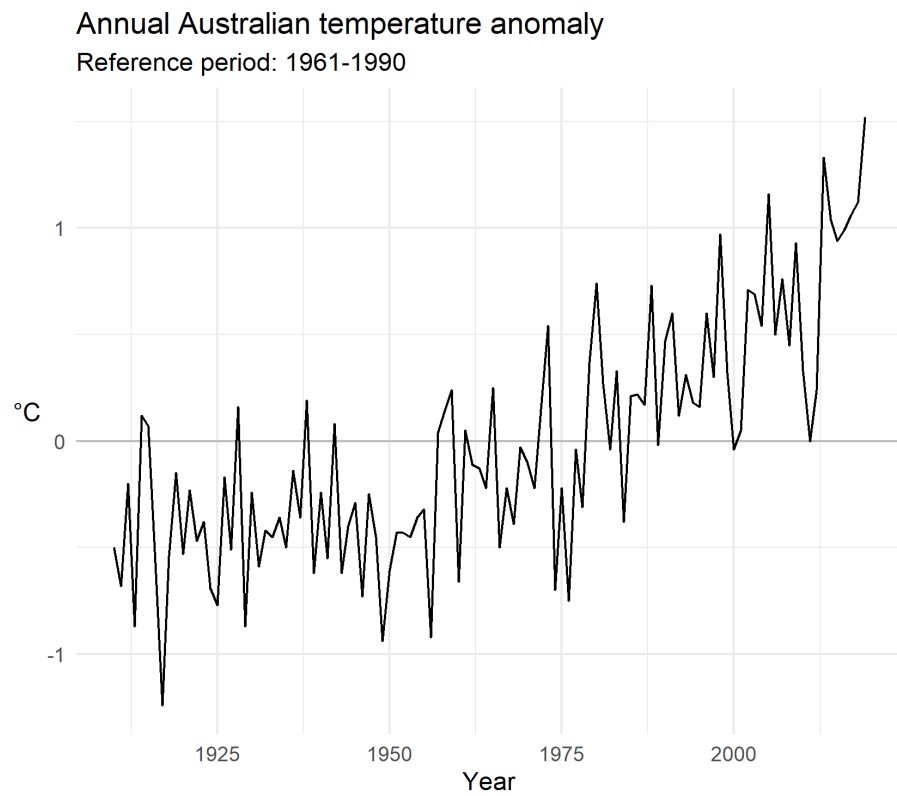


Figure 2: The annual Australian temperature anomaly. (An anomaly is the difference between the temperature at a given point in time and the average temperature across a reference period, in this case 1961-1990.)

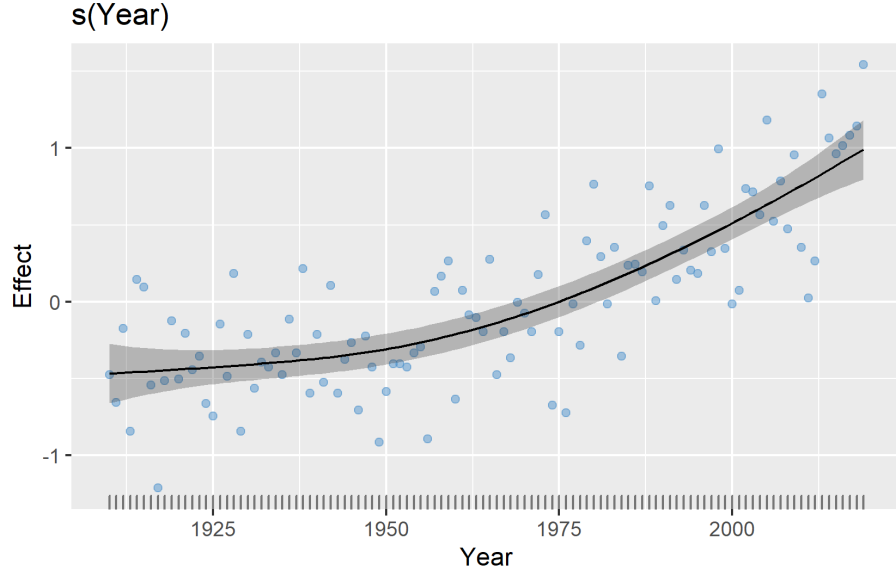


Figure 3: The cubic spline GAM fitted to the Australian temperature anomaly data. (The  $y$ -axis is the temperature anomaly in  $^{\circ}\text{C}$  and the  $x$ -axis is the year.) The black line is the fitted values  $\mu$ ; the grey area is a 95% confidence interval for  $\mu$ ; and the blue dots are the observed responses  $y$ .

be interpreted in a similar way to the OLS diagnostic plots and suggest a good model fit. Since we are modelling a time series, it is also important to check for residual autocorrelation (Figure 5).

Let's compare this to a OLS approach. Here linear model residuals exhibit quadratic variation (Figure 6) but the quadratic model  $\text{Temperature} \sim \text{Year} + \text{Year}^2$  appears to fit well (Figure 7). Figure 8 shows that the OLS model fit is almost identical to the GAM! So have we gained anything by using a GAM? Here are two reasons why a GAM might be better:

1. The GAM and OLS model begin to differ at either ends of the time series. If the data extended back further in time with the same flat trend as seen in the first 50 years, a



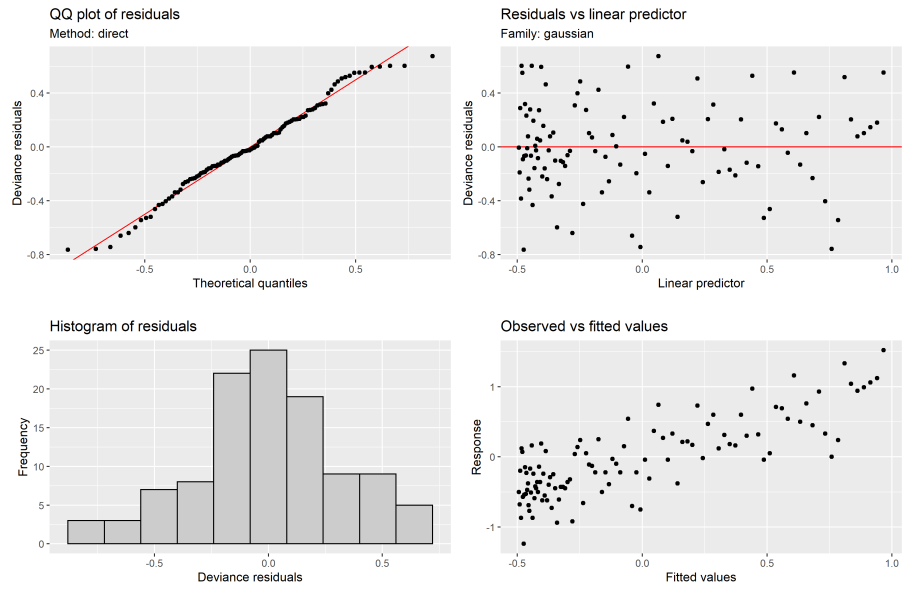


Figure 4: Diagnostic plots for the GAM fitted to the Australian temperature anomaly data.

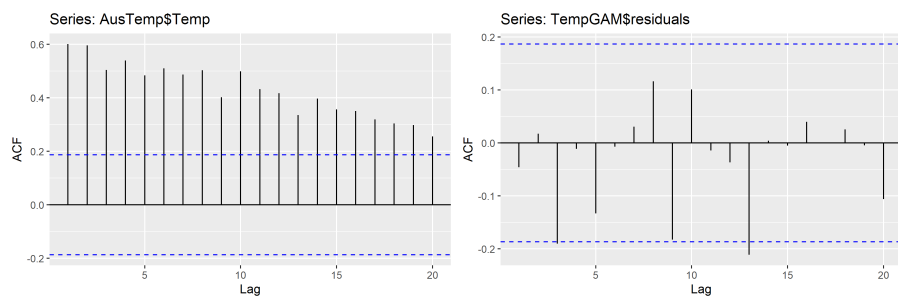


Figure 5: ACF plots for the Australian temperature anomaly time series (left) and the GAM residuals (right).

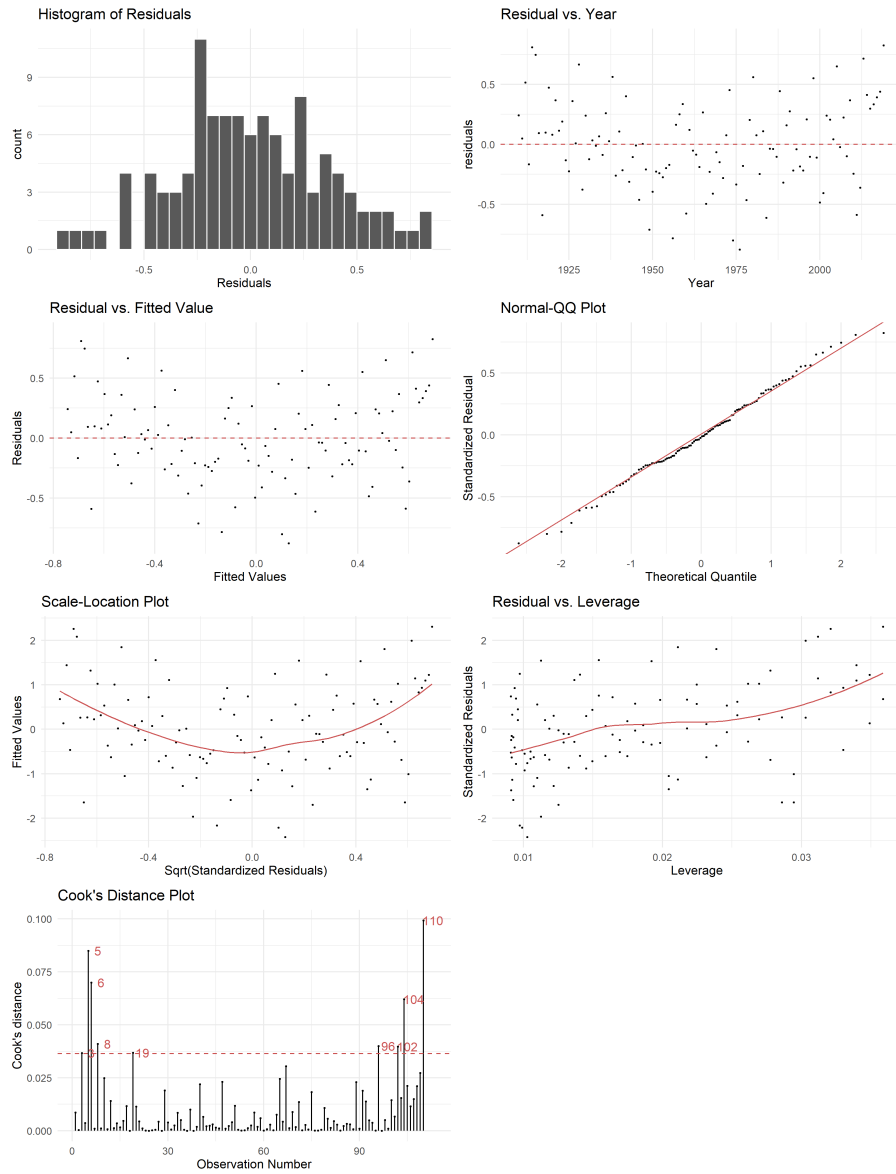


Figure 6: Diagnostic plots for an OLS model  $\text{Temperature} \sim \text{Year}$ .

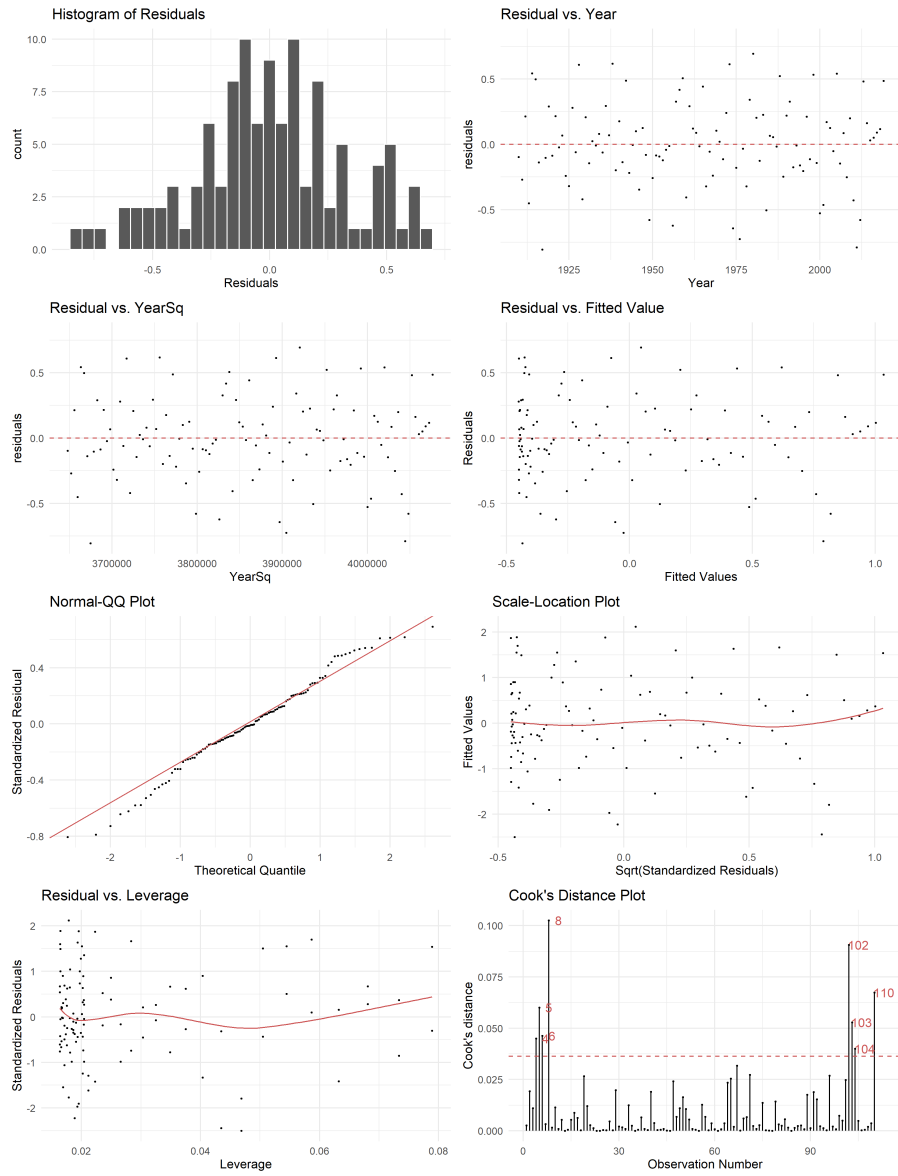


Figure 7: Diagnostic plots for an OLS model  $\text{Temperature} \sim \text{Year} + \text{Year}^2$ .

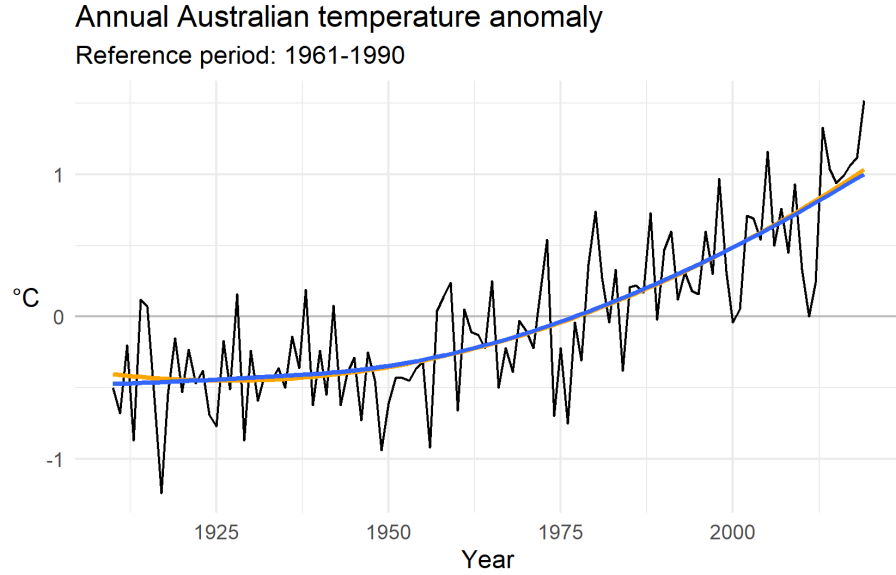


Figure 8: The Australian temperature anomaly data with the GAM fitted values (in blue) and OLS fitted values (in orange).

quadratic model would no longer provide a good fit.

2. We are interested in understanding if there has been significant warming (or cooling) within a given time period. That is, we are interested in whether the derivative of the fitted model equation is significantly different from zero. In the (quadratic) OLS model, changes in the first derivative are linear across time and so will likely not reflect the posited underlying non-linear pattern (flat then increasing). On the other hand, a GAM is piecewise polynomial, so first derivatives can change non-linearly. Thus, GAMs are a more appropriate tool here. Figure 9 plots the first derivative of the GAM fitted trend, showing that there has been a sustained, significant increase in Australia-wide temperatures since the 1940s.

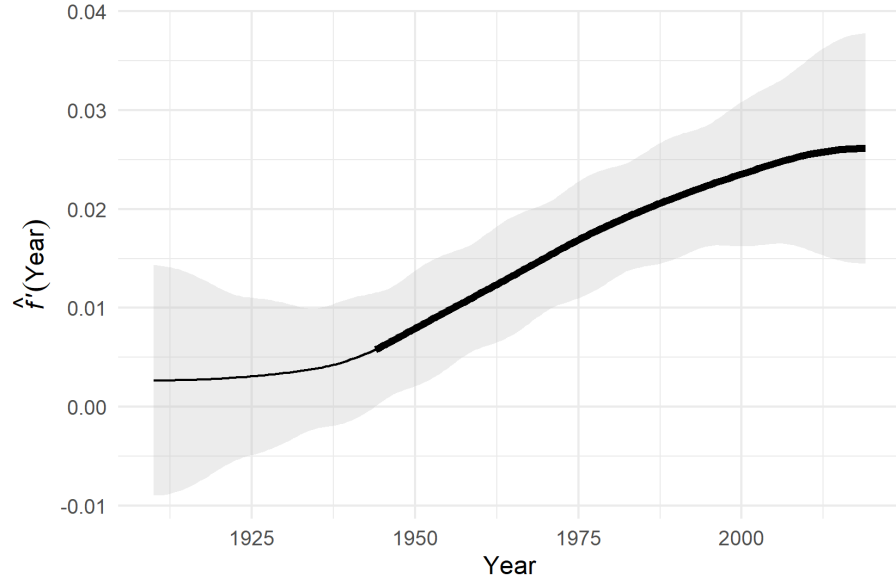


Figure 9: The first derivative of the GAM fitted trend. The grey area is a 95% simultaneous confidence interval and the thicker line denotes a significantly non-zero derivative.

## Modelling bird migrations through time and space

Let's move on to a spatial example, taken from Pedersen et al. [2019]. In this simulated data, the response  $y_i$  are the counts of 100 tagged individuals of six bird species at different latitudes and times during their annual migration. See Figure 10 for a plot of the observed locations of (species A) birds across time.

Two points are apparent from this plot:

1. There are interaction effects between latitude and time. So we need a smoother which can take multiple covariates as arguments.
2. There are a number of symmetries in the observed data. Firstly, our data is cyclic: we would expect similar counts at week 1 as at week 52. Secondly, we would want the fitted values  $\hat{\mu}_i$  to remain constant under any rotation or reflection of the covariates.

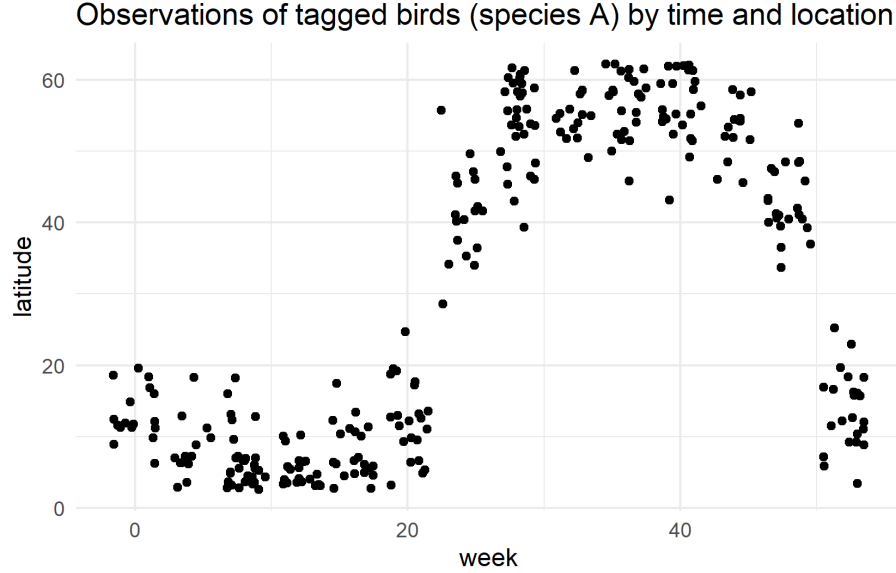


Figure 10

To make meaningful and realistic inference, our model should satisfy (at least some of) these symmetries. This calls for new smoothers!

For point 1., thin plate regression splines (TPRS) are smoothers which allow for multiple covariates. As with natural cubic splines, TPRS use polynomials as basis functions and the smoothness penalty is typically proportional to the squared second derivatives:  $\int \int \left( \frac{\partial^2 f}{\partial x_1^2} \right)^2 + \left( \frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 + \left( \frac{\partial^2 f}{\partial x_2^2} \right)^2 dx_1 dx_2$  (when there are two covariates). Moreover, there are in some sense optimal smoothers [Wood, 2017, section 5.5.1].

For point 2., smoothers  $f$  which agree on boundaries –  $f(x_1) = f(x_m)$  – are called cyclic. We can modify the basis functions of the natural cubic splines to be cyclic (see [Wood, 2017, section 5.3.1] for details). This produces the class of (unsurprisingly named) cyclic cubic splines. We will use cyclic cubic splines to model the effect of time-of-year on the bird counts.

Smoothers which are invariant to rotation or reflection are called isotropic. TPRS are

isotropic and they also have the nice property of doing away with the problem of knot locations (since TPRS uses polynomials, not piecewise polynomials). As such, we will use TPRS to model the effect of latitude on bird counts. While TPRS allow for multiple covariates, the covariates are assumed to have the same degree of smoothness. (More exactly, each covariate included in the TPRS is assumed to have the same marginal smoothness penalty.) This assumption does not hold if the covariates are measured on different scales, as it would equate for example a one degree change in latitude with a one unit increase in time. As such, TPRS is not appropriate for modelling the interaction of latitude and time.

Instead, tensor product smooths are built up from univariate smoothers, so that they allow different marginal degrees of smoothness: Given two<sup>‡</sup> univariate smooths  $f_1(x) = \sum_{k=1}^K a_k(x)\alpha_k$  and  $f_2(z) = \sum_{l=1}^L b_l(z)\beta_l$ , the tensor product smoother is given by

$$f(x, z) = \sum_{k=1}^K \sum_{l=1}^L \gamma_{kl} a_k(x) b_l(z).$$

To recap, our model will use a cyclic cubic spline for marginal time-of-year effects, a TPRS for marginal latitude effects, and a tensor product smoother for joint effects<sup>§</sup>. We will assume that the counts  $y_i$  are Poisson distributed and use the canonical link function  $g(\mu) = \log \mu$ .

The fitted model is presented as a density plot in Figure 11. The plot gives a clear picture that the tagged migratory bird species spend winter in the tropics before migrating north over summer and then returning down towards the equator before the end of the year.

---

<sup>‡</sup>For illustration, we will describe the process for two smoothers, but this can easily be generalised to an arbitrary number of smoothers. Moreover, the component smoothers need not actually be univariate, but may have vector arguments.

<sup>§</sup>This model is largely taken from Pedersen et al. [2019].

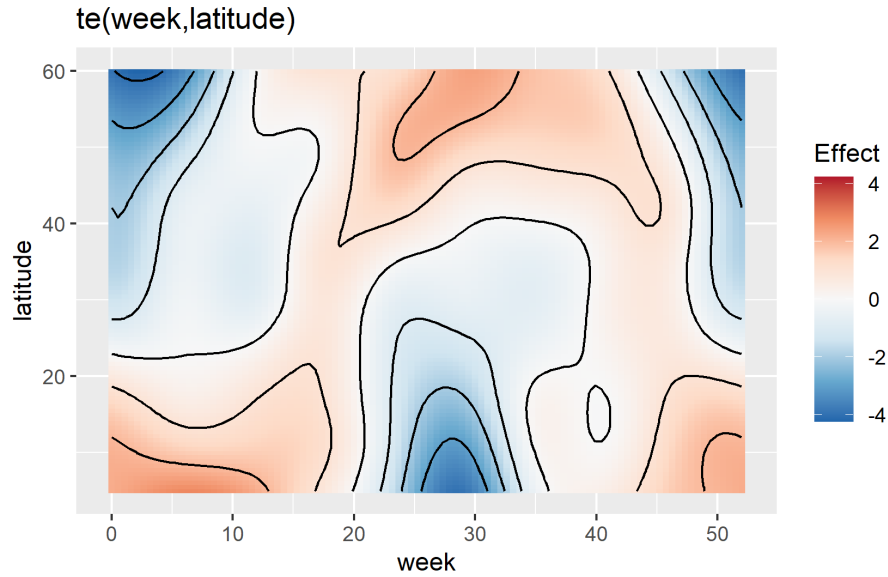


Figure 11: The GAM model fit for the bird migration data, using cyclic cubic splines for time-of-year marginal effects, TPRS for latitude marginal effects and tensor product smooths for joint effects. Red regions indicate latitudes and time-of-year where birds are more frequently observed. The use of cyclic splines is evident by the symmetry of fitted values at the start and end of the year.



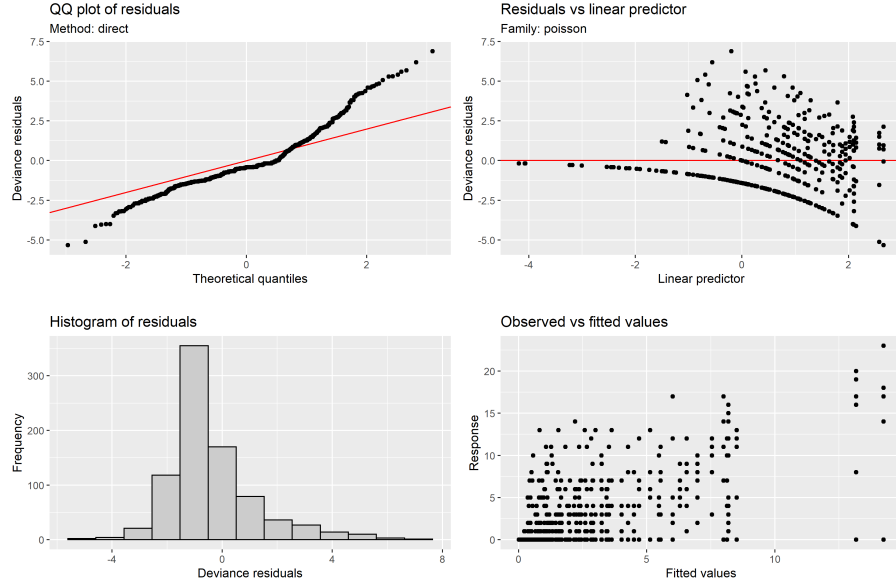


Figure 12: Diagnostic plots for the GAM in Figure 11.

Model diagnostics are given in Figure 12. Unfortunately, the residual plot indicates heteroskedasticity and the QQ plot suggests some model misspecification. This is possibly caused by omitting the species indicator for the model, since the six different species have systematically different migratory patterns. A possible solution using mixed models is briefly sketch in the following section.

## Further Explorations

It should be obvious that much has been omitted from this short primer on GAMs for spatio-temporal data. I have avoided large portions of the theory behind GAMs, especially relating to model testing, inference, effective degrees of freedom and GAMs close relationship to mixed models [Pedersen et al., 2019]. There are also many other smoothers beyond natural cubic splines and TPRS, which can prove useful in spatial modelling, particularly soap film smoothing for modelling boundary effects. Generalised additive mixed models (GAMMs) –

which add random effect terms to GAMs – are also extremely useful for partial pooling of spatially-correlated data. (For example, we could add species as a random effect in the bird migration data; thereby pooling data across species while still accounting for within-species variation, to get a better model fit for the effects of time and location.) All of this – along with further spatio-temporal modelling examples (sections 7.4-7.6) – can be found in Wood [2017].

Beyond GAMs, there are many other methods that address spatio-temporal modelling – e.g. kriging, hierarchical models and state-space models (and more specifically dynamic spatio-temporal models) [Cressie and Wikle, 2011; Wikle et al., 2019] – yet these are mostly Bayesian in flavour. I focused on GAMs for no good reason other than they are not Bayesian and they are closely related to the Stat244 material. A proper treatment of geospatial modelling should describe other methods, which allow for more complex dependence structures as well as explicit modelling of clustering. Nevertheless, we have seen that due to their flexibility GAMs can do a good job with some types of spatio-temporal data.

## References

- Australian Bureau of Meterology. Annual mean temperature anomaly - Australia (1910-2019), 2020. <http://www.bom.gov.au/climate/change/#tabs=Tracker&tracker=time series> [Accessed: 07/12/2020].
- M. Clark. *Generalized Additive Models*. Unpublished, 2019. <https://m-clark.github.io/generalized-additive-models/> [Accessed: 30/11/2020].

- N. A. C. Cressie and C. K. Wikle. *Statistics for Spatio-Temporal Data*. Wiley Series in Probability and Statistics. Wiley, Hoboken, N.J, 2011. ISBN 978-0-471-69274-4.
- P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Number 58 in Monographs on Statistics and Applied Probability. Chapman & Hall, London ; New York, 1st ed edition, 1994. ISBN 978-0-412-30040-0.
- T. Hastie and R. Tibshirani. Generalized Additive Models. *Statistical Science*, 1(3):297–310, Aug. 1986. ISSN 0883-4237, 2168-8745. doi: 10.1214/ss/1177013604.
- T. Hastie and R. Tibshirani. Generalized Additive Models: Some Applications. *Journal of the American Statistical Association*, 82(398):371–386, June 1987. ISSN 0162-1459. doi: 10.1080/01621459.1987.10478440.
- P. Lancaster and K. Šalkauskas. *Curve and Surface Fitting: An Introduction*. Academic Press, London; Orlando, 1986. ISBN 978-0-12-436060-0 978-0-12-436061-7.
- Y. Y. Lee and S. Ventura. *lindia: Automated Linear Regression Diagnostic*, 2017. URL <https://CRAN.R-project.org/package=lindia>. R package version 0.9.
- E. J. Pedersen, D. L. Miller, G. L. Simpson, and N. Ross. Hierarchical generalized additive models in ecology: An introduction with mgcv. *PeerJ*, 7:e6876, May 2019. ISSN 2167-8359. doi: 10.7717/peerj.6876.
- I. J. Schoenberg. Spline functions and the problem of graduation. *Proceedings of the National Academy of Sciences of the United States of America*, 52(4):947–950, Oct. 1964. ISSN 0027-8424.

- G. L. Simpson. *gratia: Graceful 'ggplot'-Based Graphics and Other Functions for GAMs Fitted Using 'mgcv'*, 2020. URL <https://gavinsimpson.github.io/gratia>. R package version 0.4.2.
- H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4 (43):1686, 2019. doi: 10.21105/joss.01686.
- C. K. Wikle, A. Zammit-Mangion, and N. A. C. Cressie. *Spatio-Temporal Statistics with R*. Chapman & Hall/CRC the R Series. CRC Press, Taylor and Francis Group, Boca Raton, 2019. ISBN 978-1-351-76972-3 978-0-429-64978-3.
- S. N. Wood. *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press/Taylor & Francis Group, Boca Raton, second edition, 2017. ISBN 978-1-4987-2833-1.

## Code

The code to reproduce all my analyses can be found on my Github page. I used the following packages in R:

1. `tidyverse` [Wickham et al., 2019]: for data wrangling;
2. `mgcv` [Wood, 2017]: for fitting GAMs;

3. `gratia` [Simpson, 2020]: for easy `ggplot2` GAM visualisations.
4. `lindia` [Lee and Ventura, 2017]: for easy `ggplot2` linear model diagnostic plots.

### R code for Figure 1

```
library("tidyverse")

n <- 10

set.seed(1000)

data <- tibble(y=rnorm(n),x=1:n)

f <- y ~ poly(x,n-1,raw=TRUE)

interpolate <- lm(f, data)

ggplot(data, aes(x,y)) +

geom_point() +

geom_smooth(method="lm", formula = f, se = FALSE) +

labs(y="Gaussian_noise", title="Interpolating_Gaussian_noise",

x = "Index") +

theme_minimal()

ggsave("Interpolating.png")
```

### R code for the Australian temperature anomaly data

```
library("tidyverse")
```

```

library("mgcv")

library("gratia")

library("lindia")

library("forecast")


AusTemp <- read_delim("http://www.bom.gov.au/web01/ncc/www/cli_chg/
  ↳ timeseries/tmean/0112/aus/latest.txt",
  delim="␣",
  col_names = c("Year", "Temp"),
  col_types = "cd",
  trim_ws = TRUE) %>%

mutate(Year = substr(Year, 0, 4)) %>%

mutate(Year = as.numeric(Year)) %>%

filter(Year < 2020) #for backwards compatibility, remove years
  ↳ after 2019


ggplot(AusTemp, aes(Year,Temp)) +

geom_hline(yintercept = 0, colour="grey") +

geom_line() +

labs(y="C", title="Annual␣Australian␣temperature␣anomaly",
  subtitle = "Reference␣period:␣1961-1990") +

```

```

theme_minimal() +

theme(axis.title.y = element_text(angle = 0, vjust=0.5))

ggsave("TempLine.png")


TempGAM <- gam(Temp ~ s(Year, bs="cr"), data=AusTemp)

summary(TempGAM)

draw(TempGAM, residuals = TRUE)

ggsave("TempGAMFit.png")

g <- appraise(TempGAM)

ggsave("TempGAMDiag.png", g, width = 5.59*2, height = 3.64*2)


plot_grid(ggAcf(AusTemp$Temp), ggAcf(TempGAM$residuals))

ggsave("TempGAMACF.png", width = 5.59*2, height = 3.64)


AusTemp$YearSq <- AusTemp$Year**2

TempLM <- lm(Temp ~ Year, data=AusTemp)

g <- gg_diagnose(TempLM, theme = theme_minimal())

ggsave("TempLMDiag.png", g, width = 5.59*2, height = 3.64*4)


TempLM2 <- lm(Temp ~ Year + YearSq, data=AusTemp)

g <- gg_diagnose(TempLM2, theme = theme_minimal())

```

```

ggsave("TempLM2Diag.png", g, width = 5.59*2, height = 3.64*4)

ggplot(AusTemp, aes(Year,Temp)) +
  geom_hline(yintercept = 0, colour="grey") +
  geom_line() +
  labs(y="C", title="Annual_Australian_temperature_anomaly",
  subtitle = "Reference_period:_1961-1990") +
  theme_minimal() +
  theme(axis.title.y = element_text(angle = 0, vjust=0.5)) +
  stat_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE,
  ↪ colour="orange") +
  stat_smooth(method = "gam", formula = y ~ s(x, bs = "cr"), se =
  ↪ FALSE)

ggsave("TempGAMvsLM.png")

deriv <- derivatives(TempGAM)

deriv$sigIncreasing <- (deriv$lower > 0)

deriv$sigIncreasing <- ifelse(deriv$sigIncreasing, deriv$derivative,
  ↪ NA)

deriv$sigDecreasing <- (deriv$upper < 0)

deriv$sigDecreasing <- ifelse(deriv$sigDecreasing, deriv$derivative,
  ↪ NA)

```



```

deriv$Year <- deriv$data

ggplot(deriv, aes(x = Year, y = derivative)) +

geom_ribbon(aes(ymax = upper, ymin = lower), alpha = 0.3, fill = "
  ↪ grey") +

geom_line() +

geom_line(aes(y = sigIncreasing), size = 1.5) +

#geom_line(aes(y = sigDecreasing), size = 1.5) +

ylab(expression(italic(hat(f) * "'") * (Year))) +

xlab("Year") +

theme_minimal()

ggsave("TempGAMDeriv.png")

```

## R code for the bird migration data

```

library("tidyverse")

library("mgcv")

library("gratia")

df <- bird_move %>% filter(species == "sp1")

df <- df[rep(row.names(df), df$count), 1:2]

ggplot(df, aes(x=week, y=latitude)) +

geom_jitter() +

```

```

theme_minimal() +

labs(title="Observations of tagged birds (species A) by time and
      ↪ location")

ggsave("species1.png")


birdGAM <- gam(count ~ te(week, latitude, bs=c("cc", "tp"), k=c(10,
      ↪ 10))),

data=bird_move, method="REML", family="poisson")

draw(birdGAM)

ggsave("birdGAM.png")


g <- appraise(birdGAM)

ggsave("birdGAMdiag.png", g, width = 5.59*2, height = 3.64*2)

```