Supplement of Geosci. Model Dev., 16, 573–595, 2023
https://doi.org/10.5194/gmd-16-573-2023-supplement

Geoscientific Model Development
Open Access
EGU

Supplement of

The AirGAM 2022r1 air quality trend and prediction model

Sam-Erik Walker et al.

Correspondence to: Sam-Erik Walker (sew@nilu.no)

S1 Introduction

This supplement to our paper (Walker et al., 2023) represents a user's guide to the AirGAM 2022r1 air quality trend and prediction model. The main paper describes the background and motivation for developing this model in more detail. It contains results from an extensive trend study with the model using the European Environment Agency (EEA) air quality data at a large number of European stations from 2005-2019 (Solberg et al., 2021). Parts of these data are also used in this guide to illustrate input data and results from the model.

This user's guide to AirGAM focuses on the more practical aspects of using the model, such as:

- How to download and install the model (Section S7 and Appendix A)
- How example data can be downloaded (Section S7)
- How to define the input data (Section S3)
- How to run the model on Windows or Linux (Section S4)
- How to interpret the result files (Section S5)
- Some example runs (Section S6)

Here we briefly describe each of these parts for a prospective model user.

The AirGAM model is implemented in the statistical language R (R Core Team, 2022) as a single R script which can be downloaded from a Zenodo data repository. This is described in Section S7. This R script can be run on Windows or Linux and most likely also on MacOS, although we haven't tried that yet. Appendix A contains installation details for Windows and Linux. Note that the *original* version of the model used to produce all results in the main paper and the examples in this supplement can be downloaded from (Walker, 2022b), while the *latest* version can be downloaded from (Walker, 2022b).

Example data from the EEA 2005 – 2019 trend study is also downloadable from the Zenodo repository, as described in Section S7. We recommend downloading the smaller data sets (Walker and Solberg, 2022a-b) containing just enough data to get started.

All input data to the model is described in Section S3. The inputs consist of the following three types: (1) Static station data such as name, longitude and latitude, height above sea level, type of station etc.; (2) Dynamic station data such as daily averages of air quality and meteorology over each year; and (3) The options file, which contains a set of control variables such as compound to run for, start and end the year for trend estimation, type of trend to be estimated, etc. Section S3 contains the necessary details for defining these three types of files.

Section S4 describes running the model on Windows or Linux, preferably using the accompanying batch/shell scripts on these two operating systems. It also explains how the model can be run in parallel on either system. This can be very useful when there are many stations to be processed.

Section S5 describes all result files produced by the model, while Sect. S6 contains some example runs.

The following sub-section briefly reviews some of the main features of the AirGAM model, as described more fully in the main paper. It is repeated partly here for the user's convenience.

S1.1 Review of the AirGAM model's main features

AirGAM estimates trends in daily measured pollutant concentrations at one or more monitoring stations over a given period by adjusting for trends and time variations in corresponding meteorological data. It is based on nonlinear regression GAM modelling as given by Eq. (S1) in Sect. S2.1 and has been developed primarily for the compounds NO_2, O_3, PM_{10} and $PM_{2.5}$.

Meteorological data consist of temperature, wind speed and direction, planetary boundary layer height, relative and absolute humidity, cloud cover and precipitation. The exact set of meteorological variables used in the model depends on the compound selected for analysis, as given in Table S1 in Sect. S2.1. In addition to meteorological variables introduced as covariates, i.e., explanatory variables for the concentrations, the model also uses time variables as covariates, such as the day of the week, day of the year (seasonality), and total time (days) over the period; the latter of which is associated with the model's trend term. The trend analysis is performed at each station separately.

The model is implemented using the R language for statistical computing (R Core Team, 2022) and, in particular, the GAM – generalised additive modelling – statistical modelling package `mgcv` (Wood, 2017). The program also uses the air pollution data analysis package `openair` (Carslaw and Ropkins, 2012; Carslaw, 2019) for analysis and plotting purposes and the `sandwich` package (Zeileis, 2004) for some statistical calculations. Using the GAM regression approach, the relationships between concentrations and meteorological and time covariates are represented and estimated as smooth nonlinear functions of the variables. Thus, the trend term is defined and estimated as a smooth nonlinear function of time (days) over the period selected for analysis.

Once fitted to training data, the model may be used as a prediction tool capable of predicting air pollutant concentrations for new sets of meteorological and time data which are not in the training set – e.g. for cross-validation or forecasting purposes. The model's predictive capability is evaluated with associated plots using several deterministic and probabilistic model evaluation metrics. A leave-1-year-out cross-validation procedure is incorporated in AirGAM and is usually performed automatically as part of the model run.

The model has been mainly developed for trend studies based on the air quality (AQ) measurement data hosted by the EEA, including the AirBase data (before 2013) and the Air Quality e-Reporting (AQER) data from 2013 and onwards. The EEA data provide daily or hourly surface concentrations at individual monitoring stations. For the input meteorological data, we usually extract time series from the gridded meteorological re-analysis data (ERA5) provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) for each monitoring station (Hersbach et al., 2018; Hersbach et al., 2020). Users may apply similar data sets or replace them with their own air quality and meteorology data.

Figure S1 shows a schematic of the data flow of AirGAM.

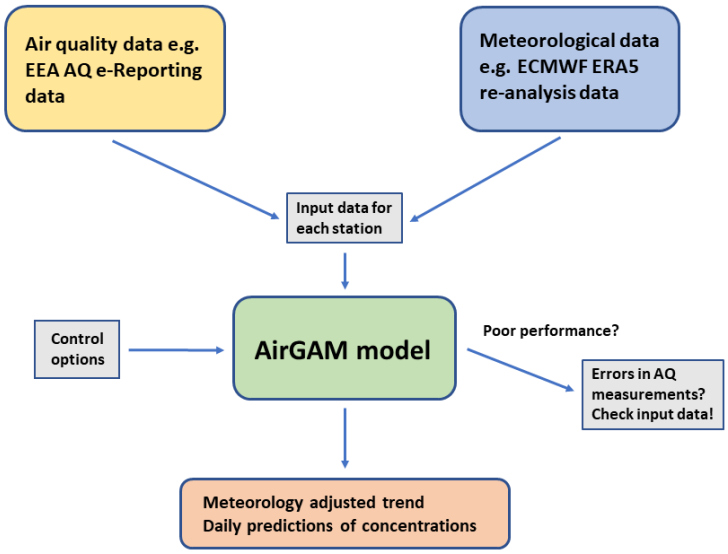

Figure S1. AirGAM data flow scheme.

In addition to concentrations and meteorology, the program reads several control options for the model run. Another feature of AirGAM is that it may sometimes check for errors in the air quality data. We have often found the poor performance of the model, e.g. low correlations between observed and model-predicted concentrations from cross-validation, associated with dubious measurement data.

The model estimates trends over a user-defined period, from a minimum of two years and upwards. For each year, the user may select the whole year; or a sub-part of the year, e.g. only winter months (say October-March), summer months (say April-September), or any user-defined interval of months for the trend analysis. Usually, only a single set of smooth relations between

the concentrations and the covariates is estimated from the data in the model. However, it is possible to operate with different groups of estimated smooth relations for different parts of the year (or sub-year) if needed, e.g. one set for the winter, say October-March, and another for the summer, say April-September. This latter capability of the model is typically necessary for modelling O_3 and $PM_{2.5}$ using data for the whole year since the relationships usually are different in the wintertime than in the summer.

S1.2 Outline of this user's guide

Section S2 briefly reviews the statistical GAM methodology implemented in the AirGAM model and details its numerical implementation. Sections S3-S5 describe the input data to the model, how to run it on Windows and Linux, and the result files. A few run examples are provided in Sect. S6, while Sect. S7 contains information about code and data availability. Appendix A describes downloading and installing the model for Windows and Linux. Appendix B has a list of the model's warning and error messages. Appendix C describes how wind direction and relative humidity are obtained from the ECMWF ERA5 data used in the EEA 2005 – 2019 study.

S2 Model formulation and implementation

Sub-section S2.1 briefly reviews the AirGAM model formulation more fully described in the main paper. It is repeated partly here for the user's convenience.

S2.1 AirGAM model formulation

In statistics, a GAM model (Hastie and Tibshirani, 1990; Wood, 2017) is a nonlinear regression model linking expected values μ_i of a given response variable Y_i to one or more explanatory variables x_{ij} through the following relations:

$$g(\mu_i) = \beta_0 + \sum_{j=1}^{p} \beta_j(x_{ij}); \quad \mu_i = E(Y_i), \tag{1}$$

where β_0 is a constant (the intercept), and where $\beta_j(\cdot)$, for $j = 1,..., p$, represents smooth functions of the covariates x_{ij}, with p the number of covariates. In our implementation of this model for air quality analysis, the response variable Y_i in Eq. (S1) represents a daily average (NO_2 or PM) or maximum 8-hour running mean (O_3) concentration at day number i at a given site, while x_{ij} represent the values of the explanatory variables, for $j = 1,..., p$, at the same location and day. These consist of various meteorological variables such as temperature, wind, etc., and time variables such as the day of the week, day of the year, etc. The meteorological covariates depend on the air pollutant being modelled, as shown in Table S1.

In Eq. (S1), $g(\cdot)$ is a function (the link function) that links the statistically expected value of the response variable Y_i, i.e., μ_i, to the covariates x_{ij}. Also, Y_i is assumed to have a definite probability distribution, the response distribution, with mean μ_i and variance V_i. Further, in Eq. (S1), each β_j is a smooth function of x_{ij}, and not simply a constant to be multiplied with x_{ij} as in multiple or generalised linear regression models.

The current AirGAM model has been developed for NO_2, O_3, PM_{10} and $PM_{2.5}$. This has resulted in a set of meteorological and time covariates found to model and predict concentrations of these compounds well, as shown in Table S1.

For NO_2, PM_{10} and $PM_{2.5}$, we apply a log link $g(\mu) = \log \mu$ and gamma distributions as response distributions. This is because these compounds generally have a somewhat more extensive range of concentration variations than O_3, with the variance of Y_i, i.e., V_i, typically proportional to μ_i^2. For such variables, it is usual practice in GAM modelling to select a logarithmic link function and a distribution potentially skewed to the right, such as a gamma, as a response distribution for Y_i (Wood, 2017). This was also applied in the previous trend studies (Solberg et al., 2018a; 2018b; 2019).

For O_3, we apply an identity link $g(\mu) = \mu$ and normal distribution as a response distribution. This choice is because O_3 has a relatively small range of concentration variations where the variance of Y_i, V_i, does not change much with the mean μ_i. Thus the response distribution is well represented with a symmetric distribution such as a normal.

The input variables have been selected by combining a priori knowledge of the main physicochemical processes and experience during the model development. Extensive research in previous work with the model (Solberg et al., 2018a; 2018b; 2019) resulted in meteorological and time variables being used, as presented in Table S1. Absolute humidity is introduced as a variable for O_3 since the gas-phase reaction $O'D + H_2O \rightarrow 2OH$ is the main production path for OH in the atmosphere and since OH, in turn, is decisive for the O_3 formation. For PM and NO_2, we used relative humidity to reflect the importance of wet deposition and cloudiness. Appendix C describes how relative humidity (and wind direction) are obtained from the ECMWF ERA5 data.

In the model, the trend term is represented as a smooth function of time ($\mathbf{x}_{10} = t$) rather than as a straight line. The main reason for this choice is for the model to be better prepared for trend studies over extended periods. In such cases, it is less relevant to represent the trend over the entire period as a straight line.

Table S1. List of meteorological and time variables used in the AirGAM model (Eq. (S1)) for various compounds. The short names refer to those used in the text and graphics files in Sect. S5.

	Meteorological variable	**Short name**	**Unit**	**Used by compound**
$\mathbf{x}_1$	Daily mean temperature at 2 m Daily temperature at 2 m at 18 UT	`temp` `temp`	°C °C	All except O_3 O_3
$\mathbf{x}_2$	Daily mean wind speed at 10 m	`ws`	m s^{-1}	All
$\mathbf{x}_3$	Daily mean wind direction at 10 m	`wd`	°	All
$\mathbf{x}_4$	Daily mean planetary boundary layer height	`pblh`	m	All
$\mathbf{x}_5$	Daily mean relative humidity Daily absolute humidity at 18 UT	`rh` `h2o`	% g kg^{-1}	All except O_3 O_3
$\mathbf{x}_6$	Medium-height cloud cover	`mcc`	%	All
$\mathbf{x}_7$	Daily total precipitation	`prec`	mm day^{-1}	PM_{10} and $PM_{2.5}$
$\mathbf{x}_8$	Weekday number	`dayofweek`	day	All
$\mathbf{x}_9$	Day number in the year or sub-part of the year	`dayofyear`	day	All
$\mathbf{x}_{10}$	Continuous-time in fraction of years (0.0 at the start of the period). This is the trend term.	`years`	year	All

Since meteorological variables are included in this GAM model to explain the expected (μ_i) and observed (Y_i) concentrations of air pollutants at each time point t_i, the estimated trend $\beta_{10}(t)$ in Eq. (S1) will represent a so-called meteorology-adjusted trend, i.e., a trend discounting for the effects of trends or time variations in these meteorological variables over the period selected for the analysis. This represents the main output from AirGAM.

In addition, AirGAM may also estimate so-called unadjusted trends. These are trends produced by the same GAM regression model set-up as above, but only including the time covariates $\mathbf{x}_8$-$\mathbf{x}_{10}$, i.e., removing all the meteorological covariates $\mathbf{x}_1$-$\mathbf{x}_7$. Both trends can be produced individually and output from the same run, making it possible to compare them.

S2.2 Numerical implementation

The GAM model (Eq. (S1)) is fitted using the R package `mgcv` (Wood, 2017). The model fitting is done independently for each station for the selected period of trend estimation. All data are used first to estimate each station's meteorology-adjusted trend. Then, cross-validation is performed by leaving out one year of data, training the GAM model on the remaining data, and predicting the left-out year's concentrations. This is repeated each year in the trend estimation period or a specific period selected for cross-validation. In the following sections, we describe the details of the implementation.

S2.2.1 Solution methods

The GAM model fitting is primarily done by applying the function `bam` (short for big additive models) in the `mgcv` library. If fitting using `bam` fails for some reason, the `gam` routine in `mgcv` is used instead. These routines are very similar, and both fit GAM models to data, but `bam` is generally much faster. It uses less memory than `gam`, which is essential for many stations with long data periods. The `bam` routine is, therefore, always tried first. It is run with the numerical solution method fREML (`method=fREML`), short for fast restricted maximum likelihood. This is the default solution method in `bam`. We use the default setting `discrete=FALSE` for all compounds in this routine, i.e., we apply no discretisation of the covariates. We do not use parallel processing in the call to the `bam` routine, only applying the default setting `cluster=NULL` and `nthreads=1` in this routine. Thus, each `bam` call only occupies the workload of a single CPU core. This makes it easier to utilise a multi-core computer when running several R sessions concurrently, as described in Sects. S4.1.1 and S4.2.2. For `gam`, we apply the REML (restricted maximum likelihood) solution method (`method=REML`). The GAM modelling community now favours these numerical solution methods rather than the older GCV (generalised cross-validation) approach to model fitting GAMs, mainly due to improved numerical stability of estimating the penalty parameters.

The `bam` and `gam` routines in `mgcv` give robust and fast solutions to the regression equations – and consistent estimates of the smooth nonlinear relations between the concentrations and meteorology and time covariates. However, these routines do not consider autocorrelations in the model residuals, leading to somewhat underestimated confidence regions for the smooth nonlinear functions, including the nonlinear trend term. To amend this, AirGAM contains an option (Sect. S3.3.14) to include an autoregressive time series model of order 1 (AR(1)) for the model residuals to handle the autocorrelations. The model is then solved using the `gamm` routine in the `mgcv` package. However, this routine is much slower and slightly less robust than the `bam`/`gam` routines. Therefore, when running for many stations, we prefer to ignore the autocorrelation, at least initially, and apply the `bam`/`gam` routines instead. Additional runs with the `gamm` routine may be performed for stations to obtain a more proper confidence region for the trend, e.g. to check whether it differs significantly from a zero function. Daily autocorrelations are expected to have a relatively small impact on long-term trends.

S2.2.2 Automatic model selection

The `bam` and `gam` routines generally estimate the covariates' smooth functions by penalising variations in these functions. However, they can only penalise a covariate function to become a straight line, not zero. Thus, the standard penalisation cannot delete unnecessary covariates from the model. However, using `select=TRUE` in the call to these routines, a further penalty to straight lines in the GAM is introduced. This may lead to straight lines becoming zero functions, deleting covariates from the GAM. The advantage is that more parsimonious models with no unnecessary covariates can be found. Thus, it represents a form of automatic model selection as part of the solution method. It has been found (Marra and Wood, 2011) that this approach, in most cases, leads to better model selections than more traditional regression approaches, typically based on adding

or removing individual covariates in a step-wise fashion. Thus, we apply `select=TRUE` in the calls to both `bam` and `gam`. Such automatic model selection is also of great value in our system since we usually have many stations and periods to model. Thus, applying the more traditional model selection approaches of step-wise adding and removing covariates would be cumbersome even if this were done automatically.

S2.2.3 Number of basis functions

Important user input in GAM modelling defines each smooth covariate's number of basis functions. These numbers represent the maximum model complexity allowed in the model for each smooth covariate. The precise value of each number is not vital. Still, they should be large enough to accommodate the GAM to correctly identify the smooth relationship between the response and each covariate. A setting somewhat higher than needed is usually not a problem since the penalisation in the GAM will usually take care of that and appropriately reduce the degree of variability (wiggliness) of the function curve. It will, however, increase the computing time. For the meteorological covariates and the time covariate `dayofyear`, we have found it sufficient to operate with ten basis functions (`k`=10) which is the default setting for smooth covariates in the calls to the `bam` and `gam` routines. For the `dayofweek` variable, seven basis functions are used, which is the maximum since this variable can take only seven discrete values (1, 2, …, 7) corresponding to Monday-Sunday. However, our system still handles this variable as a continuous time of the week variable.

Defining the number of basis functions `k_years` for the trend variable `years` is more delicate. Ideally, it should be defined high enough so that the `gam.check` routine in `mgcv` returns with the empirical number of degrees of freedom for this term somewhat lower, say 0.5-1 lower, than the theoretical number of degrees of freedom `k_years` minus 1. This means trying several values and choosing, say, the smallest, that fulfil the above criterion. The large number of stations typically used in our studies makes searching for the best value in each case somewhat intractable. Instead, we have chosen to introduce a simple empirical rule for this, introducing a basis function every three years, which seems to work well for our studies since it captures the main features of and more long-term variations in the trend quite well in most cases. Thus, our current formula used in AirGAM as a *default* for the number of basis functions for the trend term is

$$k_{\text{years}} = \max\left(2, \left\lceil n_{\text{years}}/3 \right\rceil\right), \tag{2}$$

where n_{years} is the total number of years for the trend analysis, and $\lceil\cdot\rceil$ means rounding to the nearest integer. The maximum operator in Eq. (S2) ensures that there are always at least two basis functions in the trend term, irrespective of years. The formula is only used if the user defines `k_years` as missing, i.e., the R-value NA, upon input. It is conservative, typically leading to fewer basis functions than the ideal number. If the user is interested in more details and short-term variations in the trend, `k_years` can be set to a higher value upon input as described in Sect. S3.3.11.

S2.2.4 Choice of basis function

As recommended by Wood (2017), we have speeded up the AirGAM model by applying cubic regression splines (`bs=cr`) as basis functions rather than the default thin-plate splines (`bs=tp`) in both `bam` and `gam`. This is done for all covariates except for the wind direction, which contains circular data, where cyclic cubic regression splines (`bs=cc`) are used as basis functions. This ensures equal covariate function values for angles close to 0 and 360°. Note that we do not consider the `dayofweek` and `dayofyear` covariates to be cyclic since the concentration levels on Sunday and Monday and 31 December and 1 January may differ significantly.

S2.2.5 Standard deviation of regression trend coefficients

The `vcovHAC` routine in the R package `sandwich` (Zeileis, 2004) is used to calculate the standard deviation of the linear regression trend coefficient `beta.linreg` as output to the `<station>_gam.coef_<ya>-<yb>.csv` files as described in Sect. S5.1.5. It is also used to calculate the corresponding p-value `p.linreg` in this file.

There are three reasons for using these for the simple linear regression model to obtain the `beta_linreg` coefficient. First, since a linear regression model is not an exact model, in this case, we prefer to use the sandwich construction $J^{-1}KJ^{-1}$ for the variance matrix for the linear regression coefficients rather than the simpler but more inaccurate J^{-1}. Secondly, in this case, the regression uses time series data (concentrations) that are auto-correlated. This is considered by the AC part of the `vcovHAC` routine. Thirdly, the `vcovHAC` routine also considers heteroscedasticity in the time series, i.e., variances vary with the level of the series (the concentrations). The benefit is that we obtain a better estimate of the variance of the estimated `beta_linreg` parameter and, thus, a better estimate of the p-value `p.linreg`.

S2.3 Smooth function and trend uncertainty

Smooth covariate functions are estimated by the AirGAM model using data for the whole period of trend estimation. In particular, there is such a smooth curve representing the estimated trend. These smooth curves have an estimated uncertainty representing a 95 % confidence region for each curve. These regions are depicted as the grey-shaded areas around each of the curves described in Sects. S5.1.1-S5.1.3. Note that these 95 % regions do not necessarily correspond to 95 % confidence intervals pointwise, i.e., for each point of the curve or covariate value, but rather as an average across the curve, over all covariate values (Nychka, 1988; Marra and Wood, 2012).

S2.4 Model prediction uncertainty

The AirGAM model performs predictions of daily concentrations at stations based on the actual meteorological and time covariates for each day for each left-out year in the cross-validation calculations. These predictions come with associated

255 statistical uncertainty, which is essential to estimate correctly and communicate to the user. Considering this uncertainty when comparing the model predictions with observations is crucial to accurately interpret how well the model performs in predicting concentrations not used as part of the training. Our implementation estimates a 95 % probability prediction interval (credibility interval) associated with each point prediction. The prediction intervals are displayed as grey-shaded areas around the prediction curves in the time series plots of observed and model-calculated values described in Sect. S5.1.6.

260

The 95 % prediction intervals are defined as intervals of 95 % probability for the unconditional (compound) distribution of predicted concentrations. These cannot be expressed analytically, so a Monte Carlo approach is needed. Two different procedures are used: One for O_3, where the conditional response distribution is normal, and another for NO_2 and PM, where it is a gamma. We will now briefly describe these.

265

For O_3, the procedure is as follows. On day number i, N samples of expected values $\hat{\mu}_{ij}$, $j = 1,...,N$, are drawn from a normal distribution with mean $\hat{\mu}_i$ and standard deviation $\hat{\sigma}_i$, where these last values are obtained in the same way as in the previous procedure. Next, the scale (s) parameter of the normal conditional response distribution given the expected value $\hat{\mu}_{ij}$ is defined as $\hat{s} = \hat{s}_i$, with $\hat{s}_i$ the estimated scale or dispersion parameter obtained as the square root of the `sig2` output

270 value from the `bam` and `gam` routines (Wood, 2017). Then, N samples of predicted concentrations $\hat{y}_{ij}$ are obtained by a random draw from each of the N normal distributions, i.e., $\hat{y}_{ij} \sim N\left(\hat{\mu}_{ij}, \hat{s}_i\right)$, $j = 1,...,N$, representing samples from the unconditional (compound) response distribution. Finally, a 95 % prediction interval is again obtained for day number i as the interval between these predicted concentrations between the 0.025 and 0.975 sample quantiles.

275 For NO_2 and PM, the procedure is as follows. At day number i, N samples of log-expected values $\log \hat{\mu}_{ij}$, $j = 1,...,N$, are drawn from a normal distribution with mean $\hat{\mu}_i$ and standard deviation $\hat{\sigma}_i$. These last values correspond to the estimated expected value and standard error of the linear predictor (Eq. (S1)) at day number i. These values are obtained from the `fit` and `se.fit` output values from the `predict.gam` routine in `mgcv`. Next, the shape (a) and scale (s) parameters of the gamma conditional response distribution given the expected value $\hat{\mu}_{ij}$ is defined as $\hat{a} = \hat{\phi}^{-1}$ and $\hat{s} = \hat{s}_{ij} = \hat{\mu}_{ij} / \hat{a}$, with $\hat{\phi}$ the

280 estimated scale or dispersion parameter obtained as the `sig2` output value from the `bam` and `gam` routines (Wood, 2017). Then, N samples of predicted concentrations $\hat{y}_{ij}$ are obtained by a random draw from each of the N gamma distributions, i.e., $\hat{y}_{ij} \sim \text{Gamma}\left(\hat{a}, \hat{s}_{ij}\right)$, $j = 1,...,N$, representing samples from the unconditional (compound) response distribution. Finally,

a 95 % prediction interval is obtained for the day number i as the interval between the 0.025 and 0.975 sample quantiles of these predicted concentrations.

After testing with several values of N, 100 samples were found to give satisfactory results in defining the 95 % prediction intervals for all compounds, with a good trade-off between the accuracy of the final intervals and computational efforts. This has thus been implemented in the model.

S3 Description of input to the model

Input data to the AirGAM model consists of control variables defined in two files: (1) The main run script file; and (2) The model options file. Both of these files are placed in the `airgam` main directory. Further input to the model is read from files in the directory `<inp_dir>`, which is the value of the input directory variable `inp_dir` as defined in the model options file. The name of the options file is defined in the main run script but is usually called `airgam_options.txt`. An example of this file is given in the model's software distribution.

We also provide examples of main run script files for Windows and Linux in the software distribution. For Windows, the script file is called `airgam_run.bat` and is a Windows batch file. It is called `airgam_run.sh`, a Bash shell script file for Linux. It is also possible to run the model on a Linux cluster under a Slurm (Simple Linux Utility for Resource Management) job scheduler and workload manager (https://slurm.schedmd.com). For this, we provide a template Slurm batch file `airgam_run.sl`. The AirGAM model options file (`airgam_options.txt`) is the same for Windows and Linux.

Sects. S3.1 and S3.2 below describe Windows and Linux run script files, respectively. A description of the options file is given in Sect. S3.3. Section S3.4 describes the input files as read from the `<inp_dir>` directory, containing station descriptions and data.

S3.1 The main run script file for Windows

The Windows batch file `airgam_run.bat` contains control variables needed to run the AirGAM model for Windows. These control variables are given in the form of a sequence of statements on the form `set <variable>=<value>`, where `<variable>` is the variable to be set and `<value>` its value. This script is included in the model's software distribution with an initial setting of the variables. The variables to be set are described in the following subsections.

S3.1.1 The R script executable path

The first variable you need to set in this file is the `R_script` variable, which contains the path to the R script executable. The initial setting of this variable in the software distribution as of this writing is `"C:\Program Files\R\R-4.1.2\bin\i386\Rscript.exe"`. You need to set this variable to point to where the R script executable is installed on your system. If the R script executable is already in your path when you log in to Windows, you may set `R_script="Rscript.exe"`.

S3.1.2 Model version

The following variable in the file is `version`, which describes the AirGAM model version to run as given in the model R script name. As of this writing, this variable is set to `2022r1`, and the model R script name is `airgam_2022r1.R`. The variable `model` is automatically assigned to this latter value in the script.

S3.1.3 The work directory

Next, you set the working directory, i.e., the main directory under which all the input and output (results) directories and files will be stored. The variable is `wrkdir`, and its initial value is `"C:\Users\xxx\Documents\My documents\airgam"`. You need to change this to whatever is appropriate for you.

S3.1.4 The model options file

The following variable is `optfn`, the file name for the model options. Its initial value is `"airgam_options.txt"`.

S3.1.5 Number of blocks for parallel processing

The last variable in the batch file is `nb` which is short for the number of blocks of stations to split the calculations over when performing parallel processing with the model R script. This variable's initial value is 1, meaning we want to run one instance of the program handling all stations. Suppose you run the model on a computer with multiple CPUs. In that case, you may wish to perform parallel processing utilising several CPU cores to perform the GAM model calculations faster. For example if your computer has a CPU with four cores, you may wish to set `nb`=4. Four copies of the model R script will then be set up to run in parallel, handling roughly a quarter of the stations each. Hyper-threading on Windows with Intel processors can also effectively run two separate model R scripts on each core. Thus, you may wish to set `nb` between 5 and 8 to use this on a computer with four CPU cores. Section S4.1.1 contains a more detailed description of parallel processing on Windows.

S3.2 The main run script file for Linux

The Bash shell script file `airgam_run.sh` contains all control variables needed to run the AirGAM model for Linux. The variables are defined in the shell script as a sequence of statements of the form `<variable>=<value>`, where `<variable>` is the variable to be set and `<value>` its value. This script is included in the model's software distribution with an initial setting of the variables. The variables that can be set are the same as those in the Windows batch script file, and we thus refer the reader to Sect. S3.1 for a description of these.

S3.2.1 Linux cluster

The AirGAM model can also be run parallel on a Linux cluster with multiple nodes and CPUs. The file `airgam_run.sl` is a script similar to the `airgam_run.sh` file but contains Slurm job scheduling and workload managing directives to utilise parallelisation on such a system. See Sect. S4.2.1 for a description of this file and how to perform parallel processing with it.

S3.3 The model options file

The model options file, usually called `airgam_options.txt`, contains the major control variables needed to run the AirGAM model. As stated above, this file is common to Windows and Linux. The file includes statements of the form `<variable>=<value>`, where `<variable>` is the variable to be set and `<value>` is its value. This file is included in the model's software distribution with an initial setting of the variables.

Note that no single or double quotes should be used around the text strings on the left or right of this file's equal (=) sign. However, you can have as many blank characters as you wish before and after the variable's name and before and after its value. All values are read as text strings and converted to numerical or logical values as necessary by the program. Further, the file may contain any number of blank or empty lines or comment lines, the latter of which must start with the # character. Each `<variable>=<value>` line may also include a comment at the end after a # character. The sequence of variables does not matter; you can freely permute this as you wish. The variables to be set are described in the following subsections. If a variable is commented out or removed from the options file, it will be given a default value as described in each subsection below.

S3.3.1 Input and output directories

The first two variables you need to set in this file are the program's directories for input and output (results). The defaults are `inp_dir=airgam_input` and `out_dir=airgam_results`. The model's input and output directories can be defined using full paths or relative to your defined work directory, as described in Sect. S3.1.3. The program will create the output directory if it does not already exist.

S3.3.2 The compound to run for and its unit

The following two variables are `comp` and `unit`, respectively, the compound the model will run for and its unit, i.e., the unit used for the concentrations. The default values are `no2` and `ugm-3`, respectively. For `comp`, you may use the values `no2`, `o3`, `pm10` and `pm2.5`. For `unit`, there are no specific legal values. It is only used as a text string in output plots to indicate the unit; thus, any value is permitted. However, it is interpreted and formatted by the routines in the openair package for some plots. Therefore, you may wish to stick to the conventions used by this package. For example, use `ug/m3` or `ugm-3` to indicate concentrations in µg m^{-3}. Other possible strings are `ppm`, `umol/mol`, `ppb`, `nmol/mol`, etc.

S3.3.3 The start and end year of the trend calculations

Then you need to define the start and end year of the trend calculations. This is done using the variables `year_a` and `year_b`, respectively. For example, setting `year_a=2005` and `year_b=2019` defines the trend calculation period as 2005-2019. There are no defaults for these variables.

S3.3.4 The start and end year of the cross-validations

Next, you need to define the start and end year of the cross-validation part of the calculations. This is done using the variables `year_c` and `year_d`. You may set these to the same values as `year_a` and `year_b`, respectively, default, or opt for a shorter cross-validation period. E.g., setting `year_c=2017` and `year_d=2018` means you will only perform cross-validation for the shorter period of 2017-2018. If `year_c=year_d`, the cross-validation will be performed for a single year. If `year_c > year_d`, the cross-validation part of the calculations will be skipped entirely.

S3.3.5 Sub-part of the year

Next, you need to set the sub-part of the year you will use for the trend calculations. This is done using the `subyear` variable as follows `subyear=mma-mmb`, where `mma` and `mmb` are three-letter abbreviations for the start- and end-month of the sub-part of the year. Valid values for `mma` and `mmb` are: `jan`, `feb`, `mar`, `apr`, `may`, `jun`, `jul`, `aug`, `sep`, `oct`, `nov` and `dec`. E.g., setting `subyear=nov-feb` means running the model only for November-February each year. You may also use the values `winter`, `summer`, and `year`, which are short for `oct-mar`, `apr-sep` and `jan-dec`, respectively. The default is `jan-dec`. Usually, you will want to run the model for whole years, i.e., using `subyear=year` or `jan-dec`.

S3.3.6 Seasonal conditioning

The control variable `use_season_cond` is a 0/1 logical variable of whether or not you want to use the season indicator strings as optionally given in the `season` column in the station data files. If `use_season_cond=0`, the default, then the season indicator strings are not used. This means that all dates with data throughout each year or sub-part belong to a single

"season", which indicates to the model that we only need to estimate a single set of smooth functions based on all the data. If `use_season_cond=1` but the `season` column is not present in the station data files; or exists but only contains a single type of value, e.g. `all` (the value in itself does not matter), a single set of smooth functions will again be estimated based on all data. However, suppose `use_season_cond=1` and the station data files contain season columns with at least two different values, e.g. `winter` and `summer`. In that case, conditional seasonal modelling will be turned on, and the model will estimate a set of smooth functions based on the data belonging to each unique value of the season string. For example, in the above case, one set of smooth functions will be estimated for the winter period, e.g. from October-March, based on data given in the station data files with season=winter; and one for the summer period, e.g. from April-September, based on the data with `season=summer`. The user can choose the number of unique string values and their actual values. In AirGAM, the season string variable will be converted to a factor variable in R and used in a so-called "by"-construct for the smooth functions in the call to the GAM routines. However, this "by"-construct is only used for the meteorological variables and not for the time variables such as `dayofweek`, `dayofyear` and the trend. Thus, a single smooth function will always be output for the time variables. If you need to estimate different trends for each season, you will need to use the `subyear` control variable described in Sect. S3.3.5 and run AirGAM separately for each sub-part of the year thus defined.

Such individual modelling and estimation of the smooth functions of the meteorological variables depending on the season are often essential for specific compounds. The relationship between the concentration level and the meteorological variables will often differ for seasons, e.g. winter or summertime. This is true, in particular for O_3 and $PM_{2.5}$. Note that it is also possible to use a 4-season type of modelling with AirGAM by defining the season variable to have four different values, e.g. `djf`, `mam`, `jja` and `son`, corresponding to, e.g. December-February, March-May, June-August and September-November. Likewise, it is possible to separate on an even finer basis, e.g. monthly, if need be. However, the run time will quickly increase with such finer partitioning. In practice, we have found that separating winter and summer is often sufficient, at least for the compounds mentioned above.

S3.3.7 Filename with static station data

Next comes the file's name with static station data, the variable being `statfn`. The default is `stations.csv`. It must be either a comma-separated value (CSV) file with file extension `.csv`; or a text file with one or more blank characters separating the data, in which case the file extension must be `.txt`, e.g. `stations.txt`. The file lists all stations to be used in the calculations, and one such file needs to exist for each year with data. A description of these files' content and placement under the work directory is given in Sect. S3.4.

S3.3.8 Data coverage percentages

Then comes two percentages for data coverage: The variables are `perc1` and `perc2`, respectively. The variable `perc1` describes the percentage coverage needed for the data in each year or sub-part of the year to use this year in the trend calculations. E.g., setting `perc1=75` means that at least 75 % of the data needs to exist (not be missing) in any given year for that year to be included in the trend calculations. The variable `perc2` describes the percentage coverage of years fulfilling the previous criterion of non-missing data in a specific year or sub-part to perform a trend calculation for a given station. E.g., `perc2=100` means that 100 % of the years need to fulfil the data coverage criterion for individual years (controlled by `perc1`) to perform the trend calculation. The default is 75 for both variables. Note that it is only possible to give these percentages as integers.

S3.3.9 Meteorology-adjusted and unadjusted trend modelling

The following variables indicate whether you want to perform a meteorology-adjusted trend modelling, unadjusted trend modelling, or both. The variables to be set are `incl_metadj` and `incl_unadj`, respectively. You can specify either of these to 1 if you want to include the corresponding type of trend. If both are set to 1, both kinds of trends will be estimated and output. If only one is set to 1, the indicated type will be output. If both are set to 0, no modelling will be performed, and neither of the trends will be produced, but the program will run through, reading all station data. Thus, you may wish to use this as an initial quick test of your data setup. The default for these variables is 1.

If you select to model only unadjusted trends, no meteorological data are needed in the station data files, only observed concentrations. This makes it possible to quickly run the model at stations with only air quality observations and no meteorology.

S3.3.10 Trend type

The following variable is the `trend_type`. It defines the kind of trend to use in the model. This variable can be set to nonlinear, linear, or zero values. The default is `nonlinear`, which means the trend is modelled as a nonlinear smooth function. In this case, a cubic regression spline (`bs="cr"`) is used for the trend term with penalty parameters determined automatically by the GAM solution routines, i.e., `bam`, `gam` and `gamm`, using `method="REML"` in the calls to these routines. You may use `linear` if you want to model the trend as a straight line and `zero` if you run the model without a trend. When choosing `linear`, a thin plate regression spline (`bs="tp"`) is used instead for the trend term with both penalty parameters set to a high value; currently, 10^2 is used. This leads to a straight line for the trend (approximately). When choosing `zero`, a shrinkable version of the same thin plate regression spline (`bs="ts"`) is used for the trend term with its single penalty parameter set to a very high value; currently, 10^5 is used to obtain a zero trend (again approximately). These settings are used in the calls to the GAM solution routines `bam` and `gam` in `mgcv`.

S3.3.11 Number of basis functions for the trend term

The following variable is `k_years` which is the number of basis functions defined for the trend term, i.e., for the time variable `years` of the model. The default setting of this variable when the `trend_type` is `nonlinear` is `NA`, i.e., missing value, which means that the value will be calculated based on the number of years for the trend estimation using Eq. (S2). This results in two basis functions (i.e., a straight line) for up to 8 years, where the number of basis functions switches to three. For 10, 15, 20, 25 and 30 years of trend estimation, it corresponds to 3, 5, 7, 8, and 10 basis functions, respectively. Introducing a basis function for the trend term every three years is often appropriate if the focus is to investigate the trend's main features and long-term variations. If the user is interested in more details and short-term variations, it should be set to a higher value. If `trend_type` is `linear` or `zero`, only two basis functions are used for the trend term irrespective of the `k_years` value set.

S3.3.12 Calling bam

Next is the variable `incl_bam`. This is set to 1 as default, meaning that a call to the `bam` routine in `mgcv` is always tried first. If the call to `bam` fails, the `gam` routine in `mgcv` will be called. The `bam` routine is usually much faster than `gam`. However, if you want to bypass `bam` and only call `gam`, you may set `incl_bam=0`.

S3.3.13 Automatic model selection

The following variable is `incl_select`. This is set to 1 as default which means that automatic model selection will be turned on in the calls to the `bam` and `gam` routines in `mgcv` via the `select=TRUE` setting in the calls to these routines. Usually, you will want to use this as part of the GAM modelling. However, if you want to exclude such automatic model selection, you may set `incl_select=0`, which sets `select=FALSE` in the calls to `bam` and `gam`.

S3.3.14 Include a time series AR(1) model for the residuals

Next comes the variable `incl_ar1`. This is set to 0 as default. If `incl_ar1=1`, an AR(1) model, i.e., a time series autoregressive model with a single 1-day time lag, will be used for the residuals. In this case, the `gamm` routine in `mgcv` will be used instead of `bam` and `gam`.

S3.3.15 Robust predictions

Next is a variable `rob_pred`, which can turn on two robust predictions from the fitted GAM model. This variable's default value is `limcov`. In this case, covariate values outside the interval of values encountered in the training data will be set to the nearest covariate value before being used in a prediction. In this way, we ensure that only covariate values within the training data boundaries will be used for prediction. A second possibility is `rob_pred=outmiss`. In this case, if a covariate value

is outside the interval of values encountered during training, an additional analysis is performed to check whether the corresponding predicted concentration is a potential outlier compared with the concentration values of the training data as judged by a generalised box plot method (Bruffaerts et al., 2014). If so, the prediction will be set to a missing value NA. If `rob_pred=none`, the robust prediction will not be performed.

S3.3.16 GAM seed

The variable `gam_seed` defines the seed value used in AirGAM before calling the routines `gam.check` and `k.check` from `mgcv`. It is also used before producing the 100 random samples from the unconditional (compound) response distribution when creating a 95% prediction interval for each day in the leave-1-year-out cross-validation part. This ensures exact reproducibility regarding output from the program. You can set this value to any positive whole number; the default is 1234.

S3.3.17 Legend position on plots

The variable `leg_pos` can define the legends' vertical position in the program's time series output plots. For example, you may use `top` or `bottom` to place the legends at the top or bottom of the plots. Note, however, that the legends will always be placed on the right of each plot. You may also use `leg_pos=` (an empty string) to put it in the right middle position. The default value of this variable is `top`.

S3.3.18 Autocorrelation results

If `incl_acf=1,` an analysis of the autocorrelation of the residuals is performed. This analysis checks to see to what degree the residuals are dependent or not. Ideally, in a fitted GAM model, the residuals should be independent, i.e., all autocorrelation values should be zero or close to zero. The default value of this variable is 0.

S3.3.19 Concurvity analysis

If `incl_ccuv=1,` a concurvity analysis will be performed. This type of analysis checks to what degree the covariates are independent. Thus, concurvity is to GAM modelling as multicollinearity is multiple linear regression. However, concurvity also considers to what degree the covariates are nonlinearly independent. The default value of this variable is 0.

S3.3.20 Conditional quantile plots

If `incl_cond_quant=1,` a conditional quantile plot of observations versus GAM predicted values will be produced. The routine `conditionalQuantile` in the `openair` package in R produces this plot. The default value of this variable is 0.

S3.3.21 Taylor diagram plots

If `incl_taylor=1`, a Taylor diagram plot of observations versus GAM predicted values will be produced. The routine `TaylorDiagram` in `openair` produces this plot. The default value of this variable is 0.

S3.3.22 Probabilistic evaluation results

This is controlled by the last five control variables in the model options file: `incl_pit_hist`, `incl_pit_ecdf`, `incl_marg_ecdf`, `incl_sharp` and `incl_crps`. By setting these variables to 1, one obtains PIT (Probability Integral Transform) histograms, PIT empirical CDFs (Cumulative Distribution Functions), marginal CDFs, sharpness diagrams, and CRPS (Continuous Ranked Probability Score) plots and results, respectively, based on the observed and GAM model predicted values for each year of the cross-validation period. The default values of these variables are 0.

S3.4 The input data directory and station data files

When the model is started from the run script, it reads its input from the data directory `<inp_dir>`, where `<inp_dir>` is the input directory given in the options file. This input directory can either be provided with a full path or relative to the working directory of the run script.

The model output result files will be written to the directory `<out_dir>`, where `<out_dir>` is the output (results) directory as given in the options file. This directory can also be provided either using a full path or relative to the working directory of the run script. The result files in this directory are described in Sect. S5.

The input data directory must be organised with one or more sub-directories `<ccc>/<yyyy>` where `<ccc>` denotes the compound, e.g. `<ccc>=no2` and `<yyyy>` denotes the year with data, e.g. `<yyyy>=2005`. The `<ccc>` string must be the same as the compound string `comp` as given in the options file, e.g. `no2`, `o3`, `pm10` or `pm2.5`. There must be one such sub-directory `<ccc>/<yyyy>` for each year `<yyyy>` in the period defined for the trend calculation in the options file (`year_a-year_b`).

In each sub-directory `<ccc>/<yyyy>`, there needs to be a single file with a list of all stations active for that year. This file's name is defined in the options file by the variable `statfn`, e.g. by default `statfn=stations.csv`. Each such station file is a text file with one header line with field names and one or more subsequent lines with the following station data:

- Station EoI code (`name`), e.g. `EE0018Ah`[1]
- Station longitude (`lon`) in degrees (°)
- Station latitude (`lat`) in degrees (°)

[1] Note that we added the letter ‘h’ or ‘d’ to the EoI code to distinguish between hourly and daily based data

- Station height above mean sea level (`z`) in m
- Station type (`type`), e.g. traffic, industrial or background
- Station area (`area`), e.g. urban, suburban or rural
- Country where the station is located (`country`)

The file must be either a comma-separated value (CSV) file with file extension `.csv`; or a text file where one or more blank characters separate the data, in which case the file extension must be `.txt`, e.g. `stations.txt`. In either case, the header field names must be exactly as given in the parenthesis above, without double quotes around the terms. The model actively uses `name`, `type`, `area` and `country` values for file naming and plotting purposes. However, the other data may be used to create sub-sets of stations for specific purposes, e.g. if one only wishes to run for stations between certain latitudes or longitudes, below certain mean sea levels, or stations of a particular type, in certain types of areas or situated in a given country, etc. However, this must be done manually by the user. There are currently no filters built into the program to select sub-sets of stations automatically. Usually, stations are pre-screened for altitude, and only stations below a certain height above sea level are used in the AirGAM model, e.g. only stations below 1000 m. This is based on the view that the model and the meteorological data are less appropriate for mountain stations.

We have used the EEA's EoI codes for naming the stations. However, any station code or name could be used as long as the names in the station list file agree with the names of the individual data files (see below). Note that the EoI codes were the central entity in AirBase until 2012, while the station local-id was introduced in AQER. To link the time series across the AirBase/AQER databases, we used the Sampling Point Identifier (provided in both databases), a unique code referring to the combination of pollutant and monitoring stations. We added the letter ‘h’ or ‘d’ to the EoI code to distinguish hourly-based data from daily ones when both types of measurements of the same compounds have been carried out at a station.

When the model starts, it reads the station list files for each year and builds up a global list of stations internally. In this build-up, the program tolerates missing years. A station listed for a given year is added to the global list if it is not on the global list already. It is also checked if there is insufficient data coverage based on the number of years remaining until the last year compared with the `perc2` data coverage percentage defined in the options file. If so, the station is excluded. Otherwise, the station is accepted and added to the global list. When the global list is finally built, the model will consider each station in this list, one at a time. Whether or not trend calculations and cross-validation analysis will be performed for a station will depend on the actual station data read and whether or not these meet the data coverage criteria defined by the `perc1` and `perc2` coverage percentages described in Sect. S3.3.8.

When the model performs calculations for a given station, it reads the station data. For each year `<yyyy>`, the station data are read from a separate file in the `<ccc>/<yyyy>` sub-directory. The name is `<station>_<ccc>_<yyyy>.csv` when

it is a comma-separated value (CSV) file. It is also possible to use blank-separated values files, wherein the file names are the same but with the extension `.txt` instead. The program automatically detects which file name type is present in each sub-directory. In either case, `<station>` must be the station EoI code name string, e.g. `EE0018Ah`, and `<ccc>` and `<yyyy>` must be the compound name and year, respectively.

Each station data file is a text file with one header line with field names and one or more subsequent lines with daily station data of air quality, meteorology and optionally a season indicator string. Each line of data in this file consists of the following values:

- A date string (`date`) on the form `yyyy-mm-dd` (year-month-day)
- Observed concentration (`<ccc>`) of the given compound in µg m^{-3}
- Air temperature (`temp`) in °C
- Wind speed (`ws`) in ms^{-1}
- Wind direction (`wd`) in degrees (0-360 °)
- Planetary boundary layer height (`pblh`) in m
- Relative humidity (`rh`) in % (for all compounds other than O_3)
- Absolute humidity (`h2o`) in g kg^{-1} dry air (for O_3)
- Medium height cloud cover (`mcc`) in %
- Precipitation (`prec`) in mm day^{-1} (only for PM_{10} and $PM_{2.5}$)
- Optional season indicator string

Note again that the header field names in these files must be as given in the parenthesis above. However, upper case letters in these names are allowed but converted to lower case internally in AirGAM. Here the header `<ccc>` means to use, e.g. the header-name `no2` for the observed concentrations if we run for NO_2. Again, there should be no double quotes in the header names. Missing data are denoted in these files with the two-letter value NA, standard for R missing data. The program tolerates missing data values in the station data files in the sense that the model uses only whole rows with non-missing data. The program also accepts full missing years with data, i.e., the station data file need not exist for all years of the trend calculation.

Note again that if you opt for running only an unadjusted trend model with AirGAM (see Sect. S3.3.9), no meteorological data are needed in the station data file, only dates and air quality observations, and optionally, the season indicator strings.

Whether or not a trend calculation will be performed for a given station depends on this station's available data and the coverage percentages as read from the options file. At an absolute minimum, the program needs at least two years with data to run, i.e., at least a period with data in two different years. For example, if the period 2005-2019 is chosen and a 75 % coverage of years is specified in the options file, one needs to have at least twelve years with data for a given compound available for a station to perform trend calculation.

S4 Running the model

S4.1 Running the batch script for Windows

The simplest way to run the AirGAM model on Windows is to double-click on the batch script file `airgam_run.bat`, placed in the `airgam` directory. This will run the model based on the control variables set in this batch file and the model options file, usually named `airgam_options.txt`. These are described in Sects. S3.1 and S3.3, respectively.

Alternatively, the user may start a command prompt window in Windows and navigate to the same `airgam` directory. The model may then be started by issuing the command `airgam_run.bat` in the command prompt window. However, note that for this to work, the %%i construction in the batch file must be replaced by %i. However, this second approach's advantage is that the command prompt window will not disappear in the case of errors, and it is possible to see any eventual error messages from the batch run. After a model run, the user should consult the model's log file (see below).

In either case, the model reads its input data from the `<inp_dir>` directory and writes its output in text and plot files to the `<out_dir>` directory. The content of these directories is described in Sects. S3 and S5, respectively. Status messages and any warning or error messages are written to the program log-file `AirGAM_log.txt` in the sub-directory `main` of the `<out_dir>` directory. After a model run, this file should be inspected to check for status and any warnings or errors. A description of this file is given in Sect. S5.1.12. The warning and error messages are described in more detail in Appendix B.

S4.1.1 Parallel processing

If you have a large number of stations, you may wish to split the number of stations into `nb` > 1 blocks of stations and run each block in parallel utilising multiple CPUs or CPU cores concurrently. You must set the control variable `nb` in the batch script file `airgam_run.bat` to your desired number of blocks. For example, if you run on a Windows computer with four CPU cores, you may wish to set `nb`=4, or if you want to utilise hyper-threading running up to two processes per core, you may set `nb` to some number between 5 and 8, e.g. `nb`=7, so that you half a core available to other tasks.

When you start the batch script file `airgam_run.bat`, `nb` copies of the AirGAM R script will be started, each in a separate run window by the last command in this batch file. Each copy of the R script will process its block of stations indicated by the variable `ib` in the call to the R script. This variable ranges from 1 to `nb`. Each R script copy will create the same global list of stations but only process the part indicated by the block number `ib`. For example, if an R script copy receives the argument variable `ib=3,` only stations in the third block of the global list will be processed by this R script. This way, `nb` copies of the R script will be run parallel on a Windows computer handling separate blocks of stations.

There will be no conflicts writing to the result files since station names are used as unique identifiers in these files. The only exception is the result files containing all station results, i.e., the `AirGAM_*.csv` files and the `AirGAM_log.txt` file. To avoid conflicts when writing to these files, the results will be written to files with names `AirGAM<ib>_*.csv` and `AirGAM<ib>_log.txt`, thus a unique set of files for each block `ib` of stations when performing parallel processing. After all model runs are finished, all the separate `AirGAM`-files must be concatenated to one common set of `AirGAM_*.csv` and `AirGAM_log.txt` files. This can be accommodated by using the script `airgam_cat.bat` in the `airgam` directory.

S4.2 Running the shell script for Linux

Running the model on Linux is similar to running it on Windows (see Sect. S4.1). The most straightforward way is to use the Linux Bash script file `airgam_run.sh` in the `airgam` directory. This will start and run the AirGAM model based on the variables defined in this file. These are described in Sect. S3.2.

As for Windows, the model reads input data from the directory `<inp_dir>` and writes output to the directory `<out_dir>`. The input and output (results) files are described in Sects. S3 and S5, respectively. Similarly, status and eventual warning or error messages are written to the program log-file `AirGAM_log.txt` in the sub-directory `main` of the `<out_dir>` directory. This file should be inspected to check the status and any warnings or errors from the model's run. The description of this file is given in Sect. S5.1.12. The warning and error codes and messages are described in more detail in Appendix B.

S4.2.1 Linux cluster

The model can also be run on a Linux cluster with multiple nodes and CPUs per node. Such a cluster usually employs a system to submit jobs through a queue system and run parallel programs. Slurm is a very common job scheduler and workload manager for Linux clusters (https://slurm.schedmd.com). The file `airgam_run.sl` in the `airgam` directory provides a Slurm batch script file template for running the model on a Linux cluster using Slurm. The file contains `#SBATCH` Slurm directives for starting and running several parallel model instances. As for Windows (see Sect. S4.1.1), this is done by splitting the number of stations into `nb` > 1 blocks of stations to be run in parallel.

After you have decided on the number of station blocks `nb` you wish to run in parallel, you need to edit two lines of the `airgam_run.sl` file. First, you need to edit the line `nb=<value>` to insert the total number of station blocks, e.g. `nb=20` if you want to use 20 blocks of stations. Next, you need to edit the line `#SBATCH –array=<ab>-<bb>` to edit the start and end indices of the blocks you wish to run in parallel. For example, setting `#SBATCH –array=1-20` will run for station blocks 1 to 20 in parallel using 20 CPUs. Finally, you submit the job simply by issuing the command `sbatch airgam_run.sl`.

S5 Description of result files

The AirGAM model produces several graphics and text files for each station. These result files are placed in the two sub-directories `main` and `eval` of the output directory `<out_dir>/<ccc>_<ya>_<yb>_<ma>-<mb>`. Here `<out_dir>` is the output directory as set in the model options file `airgam_options.txt`, `<ccc>` the compound used (`no2`, `o3`, `pm10`, `pm2.5`), `<ya>` and `<yb>` the start and end year respectively of the period selected for the trend estimation, i.e., `year_a` and `year_b` as defined in the options file, and `<ma>` and `<mb>` a three-letter abbreviation of the start and end month (`jan`, `feb`, …, `dec`), of the sub-part of the year used for the calculations, as defined by the variable `subyear` in the options file.

In the following subsections, each result file is described in more detail. We separate the main results, deterministic model evaluation results, and probabilistic model evaluation results. These three types of output are described in Sects. S5.1-S5.3 below. All main result files are described in Sect. S5.1 and these are written to the sub-directory `main` of the output directory. Deterministic and probabilistic evaluation files are described in Sects. S5.2 and S5.3 and are written to the `eval` sub-directory.

Overall, three types of files are being produced by the model:

- Plot files using the format PNG (Portable Networks Graphics) (`.png`)
- Text files of comma-separated (`.csv`) or blank-separated (`.txt`) data with one header line with field names
- Text files with results in a more free-format style (`.txt`)

In describing the result files below, `<station>` will denote the station name acronym, and `<ya>` and `<yb>` the start and end year, respectively, for the period selected for the trend estimation. Further, `<yy>` will denote a specific year in the period chosen for cross-validation, with the start and end year of cross-validation `<yc>` and `<yd>`, respectively. The cross-validation period is always the same or shorter than the period selected for the trend calculation. All plots are high quality with a resolution of 300 dpi (dots per inch), a height of 2000 pixels, and a width of either 2000 or 4000 pixels, depending on the plot type.

Some results are being produced separately for the meteorology-adjusted and unadjusted GAM models. These files will include the `<adj>` specifier in the file name, with `<adj>=metadj` for the meteorology-adjusted model and `<adj>=unadj` for the unadjusted model.

There is also a set of files containing specific results for all stations. They have file names on the form `AirGAM_*.csv` and `AirGAM_*.txt`, where the asterisk is replaced by an indication of the type of results. When performing parallel processing with the model, these files will be named `AirGAM<ib>_*.csv` and `AirGAM<ib>_*.txt` instead, where `<ib>` is the index of the block of stations to run for. These indices range from 1 to `nb`, where `nb` is the number of station blocks. After the parallel runs are finished, the user can use the script `run_cat.bat` on Windows or `run_cat.sh` on Linux, which both

reside in the airgam main directory, to concatenate these into a set of common `AirGAM_*.csv/txt` files. After this concatenation operation, the user may wish to delete the individual `AirGAM<ib>_*.csv/txt` files.

As an example of the output in this section, we use the station EE0018Ah, a background station in an urban part of Tallin, Estonia. It represents the median station regarding cross-validation correlation results for NO_2, which means that half of all stations had a poorer correlation than this and half had a better one for NO_2. Thus, the station should represent results at individual stations for NO_2.

S5.1 Main results

Below we describe the most central result files from a run with the AirGAM model.

S5.1.1 The estimated trend curve

The file name is `<station>_gam.trend_<adj>_<ya>_<yb>.png`. An example of this type of plot is shown in Fig. S2.

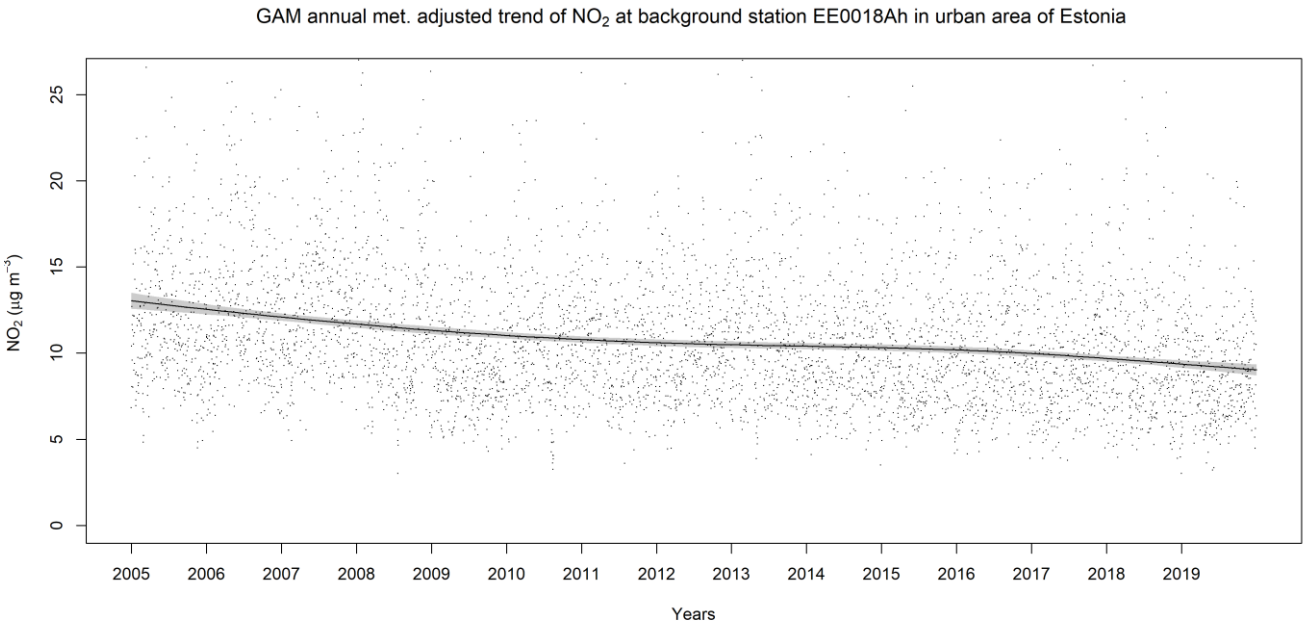

Figure S2. The meteorology-adjusted trend curve for NO_2 at station EE0018Ah for 2005-2019 (whole years). The units are year (x-axis) and µg m^{-3} (y-axis).

This plot shows the smooth response function corresponding to the estimated meteorology-adjusted trend for NO_2 at station EE0018Ah ((`<station>=EE0018Ah`) over the period 2005-2019 (`<ya>-<yb>`). The dots in the plot are the partial residuals from fitting the current GAM model, i.e., residuals that would have been obtained if dropping this specific term from

the model while leaving all other estimates unchanged. The grey shaded area represents a 95 % confidence region for the smooth trend curve. Note that this 95 % region does not necessarily correspond to 95 % confidence intervals pointwise, i.e., for each point on the curve or covariate value, but rather as an average across the curve, over all covariate values (Nychka, 1988; Marra and Wood, 2012). The confidence region and intervals are calculated by the `plot.gam` function in the `mgcv` package. In the call to this routine, we set the parameter `seWithMean=TRUE`; thus, the confidence region also includes the uncertainty about the overall mean. Work by Marra and Wood (2012) suggests that this setting results in improved coverage performance.

As shown in Fig. S2, the trend for NO_2 at station EE0018Ah decreases from 2005 to around 2011, where it becomes relatively flat up to 2015 before falling again slightly towards 2020. The trend declined from about 13 µg m^{-3} in 2005 to approximately 10 µg m^{-3} at the end of 2019, thus decreasing to about 3 µg m^{-3} over 15 years.

S5.1.2 The estimated trend data values

The file name is `<station>_gam.trend_<adj>_<ya>_<yb>.csv`. This is a comma-separated (CSV) text file containing the data used to produce the plot in the previous section. Each row of this file contains a time indicator (year value) (`years`), followed by the trend curve value of the smooth response function for the trend (`trend`) and the lower and upper 95 % confidence region levels (`trend.025`, `trend.975`). The file always contains 100 values of the trend curve.

S5.1.3 Smooth response functions plots

The file name is `<station>_gam.smooth_<adj>_<ya>_<yb>.png`. An example of this type of plot is shown in Fig. S3.

This panel of plots shows the smooth response function for each covariate of the meteorology-adjusted model for NO_2 at station EE0018Ah (`<station>=EE0018Ah`) based on the years 2005-2019 (`<ya>-<yb>`). Each response function describes an estimated smooth relationship (smooth curve) between the log of the concentrations and the corresponding covariate values from the GAM model regression.

Again, the dots in each plot are the partial residuals from fitting the current GAM model, i.e., residuals that would have been obtained if dropping the specific term from the model while leaving all other estimates unchanged. The grey-shaded areas represent 95 % confidence regions for each smooth curve. Again, these 95 % regions do not necessarily correspond to 95 % confidence intervals pointwise, i.e., for each point on the curve or covariate value, but rather as an average across the curve, over all covariate values (Nychka, 1988; Marra and Wood, 2012). For these plots, we again set the parameter `seWithMean=TRUE` in the call to the routine `plot.gam` in `mgcv`. Hence, the intervals also include uncertainty about the

overall mean, improving coverage performance. The last plot in the panel shows the smooth trend curve as described in Sect. S5.1.1.

As shown in Fig. S3, the concentrations of NO_2 at station EE0018Ah decrease with temperature (top left plot) up to about 0 °C and then increase with the temperature above this. For wind speed (top centre), the concentrations continuously decrease with wind speed which is natural. The concentrations vary quite a bit with wind direction (top right), with the lowest concentrations for wind directions from around 90° and the highest from around 250-300°. Concentrations decrease with planetary boundary layer height (middle left), which is also natural and reduces but only slightly with relative humidity (middle centre). Medium cloud cover does not influence the concentration levels much (middle right). Concentrations are also relatively flat during weekdays (bottom left) except for a slight increase on Fridays but are lower during the weekend. The day of the year seems to influence concentrations in a sinusoidal pattern (bottom centre), with the lowest concentrations during summertime and the highest during wintertime. The trend curve plot (bottom right) is the same as in Sect. S5.1.1 and is commented upon there. The estimated relations between the concentrations of NO_2 and the meteorological and time covariates are typical for most AirBase/AQER stations in Europe during 2005-2019. There are different relations for the other compounds, although several similar patterns, e.g., wind speed, planetary boundary layer height and the day of the week.

If seasonal conditioning is used (`use_season_cond=1`), smooth functions will be estimated for each season and output to separate files. The file names will be `<station>_gam.smooth_<season>_<adj>_<ya>_<yb>.png`, with `<season>` the season string. These season strings are taken from the station data files. There will be one such file for each unique value of the season string with plots of the smooth response functions for the indicated season.

S5.1.4 Smooth response functions values

The file name is `<station>_gam.smooth_<adj>_<ya>_<yb>.csv`. This is a comma-separated (CSV) text file containing the data used to produce the plots in the previous section. Each row of this file contains a row index (`i=1,2,…`), followed by the x- and y-coordinates of the smooth response functions for each covariate (`<cov>.x <cov>.y`) where `<cov>` is the name of the covariate. The files always contain 100 pairs of x- and y-coordinates for each smooth function. Again, if seasonal conditioning is used (`use_season_cond=1`), there will be one such text file for each season. The file names will be `<station>_gam.smooth_<season>_<adj>_<ya>_<yb>.csv`, with `<season>` the season strings.

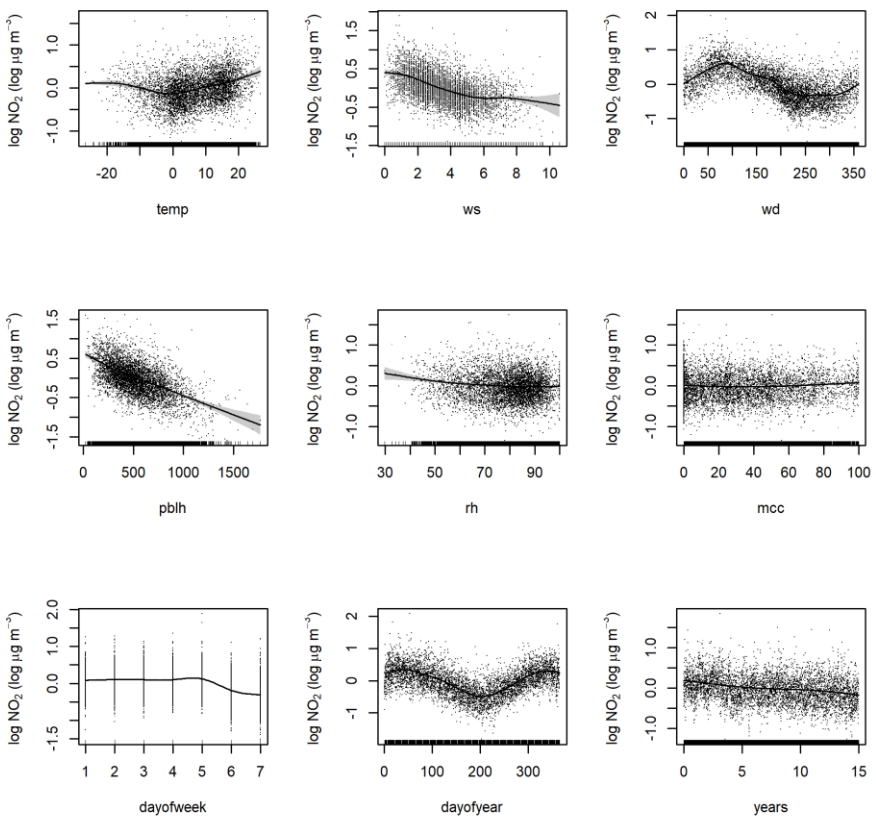

Figure S3. Smooth response functions for each covariate for NO_2 at station EE0018Ah for 2005-2019. The covariates (x-axes) units are described in Table S1. The unit on the y-axis is log µg m^{-3}.

S5.1.5 Regression coefficients and p-values

The file name is `AirGAM_gam.coef_<ya>_yb>.csv`. This is a CSV file containing the smooth response functions beta-coefficients and p-values plus some other results related to the fitted model based on all years used for the trend estimation (`<ya>-<yb>`). Note that this file is only being produced from the meteorology-adjusted model. The file is common to all stations, with a header line and one row of results per station.

Each row contains the station name acronym (`name`), beta coefficients for each covariate (`beta.<cov>`), corresponding p values (`p.<cov>`), GAM regression R^2 value (`r.sq`), deviance explained (`dev.expl`), Akaike information criterion (`aic`), a linear regression trend slope coefficient (`beta.linreg`), and its p-value (`p.linreg`). The `beta.<cov>` coefficients are calculated for each covariate based on the smooth response function's slope between the 0.25 and 0.75 quantiles of the corresponding set of covariate values. The `p.<cov>` values are associated with a null hypothesis of an exactly zero response

function for the corresponding covariate. They can be used to reject this null hypothesis in the same way as in linear regression. In addition to the GAM model, a simple linear regression model is run based on concentration (O_3) or log of concentration (NO_2 and PM) as the response variable using only the time variable `years` as a covariate. The `beta.linreg` coefficient with its p-value `p.linreg` corresponds to the slope coefficient from this linear regression. For NO_2 and PM, the slope is transformed back to the original scale.

For NO_2 at station EE0018Ah for 2005-2019, all p-values are close to zero ($< 3 \cdot 10^{-5}$), while R^2 and the deviance explained are 0.69 and 0.77, respectively. Finally, the linear regression slope is -0.21 µg m^{-3} per year with a p-value of $3.4 \cdot 10^{-7}$.

S5.1.6 Plots of observations and model predictions from cross-validation

The file name is `<station>_gam.pred_<yy>_<yy>.png`. An example of this type of plot is shown in Fig. S4.

The plot shows observed (blue curve) and model-predicted (red curve) concentrations of NO_2 at station EE0018Ah (`<station>=EE0018Ah`) for 2019 (`<yy>=2019`). The model predictions are based on training the meteorology-adjusted GAM model on all years for the trend estimation (2005-2019) except for the plotted year (2019). There is one such file being produced for each year `<yy>` of the leave-1-year-out cross-validation period (`<yc>-<yd>`). Here the start and end years for the cross-validation `<yc>` and `<yd>` can be different from `<ya>` and `<yb>`, corresponding to a possible sub-period of the whole period defined for trend estimation. In this way, we show how well the model can predict concentrations left out from the training of the GAM model for each year of the cross-validation period.

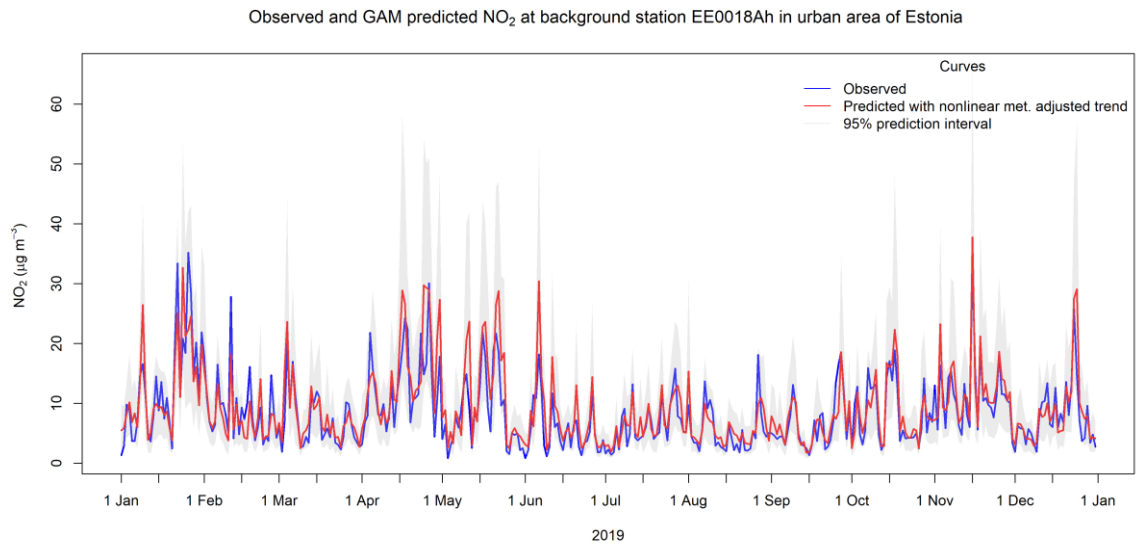

Figure S4. Observed (blue curve) and model-predicted (red curve) concentrations of NO_2 at station EE0018Ah for 2019.

As shown in Fig. S4, there is quite a good correspondence (root mean squared error (RMSE) = 3.5 μg m^{-3}, R^2 = 0.72) between observed and predicted values for NO_2 at station EE0018Ah for the year 2019.

S5.1.7 Values of observations and model predictions from cross-validation

The name of the file is `<station>_gam.pred_<yy>_<yy>.csv`. This CSV file contains the data used to produce the file in the previous section. Each row of the file includes the date (`yyyy-mm-dd`), followed by the observed and model-predicted concentrations for the station `<station>` for the left-out year `<yy>`.

S5.1.8 Model linear predictor values

The name of the file is `<station>_gam.linpred_<ya>_<yb>.csv`. This is a CSV file containing observed and model linear predicted daily concentrations for the indicated station (`<station>`) using predictions from a fit to the data for all years or sub-parts of years used for the meteorology-adjusted trend estimation (`<ya>-<yb>`). Each row of the file contains the date (`yyyy-mm-dd`) and the following data: observed concentration (`linpred.obs`), predicted concentration (`linpred.pre`), the constant or intercept term (`beta0`), followed by the contribution to the predicted concentration from each smooth covariate response function for the covariate values for the current date (`term.<cov>`), where `<cov>` ranges over the set of covariate names. The sum of the covariates' contributions plus the constant term equals the predicted concentration value. It is important to note that the observations and predictions in this file are the concentrations on the scale of the GAM linear predictor. This means that the concentrations are on the original scale for O_3 (μg m^{-3}) and the logarithmic scale for NO_2, $PM_{10,}$ and $PM_{2.5}$ (log μg m^{-3}).

S5.1.9 Plots of (sub-) annual and monthly averages and medians of observations and model predictions

The file names are `<station>_gam.aave_<ya>_<yb>.png` and `<station>_gam.mave_<ya>_<yb>.png`, for (sub) annual and monthly averages, with examples of plots shown in Fig. S5 and S6, respectively. For medians, the string `ave` in the filenames is replaced by `med`.

The plots show observed (blue curve) and predicted (red curve) annual and monthly average concentrations of NO_2 at station EE0018Ah (`<station>`) for 2005-2019 (`<ya>-<yb>`). The orange and dark green curves, respectively, show the unadjusted and meteorology-adjusted trends. In these plots, we use annual and monthly averages of the predictions from the cross-validation for all years used for the trend estimation (`<ya>-<yb>`). Thus, the model predictions will always be from the meteorology-adjusted model.

As shown by Figs. S5-S6, there is a good correspondence between the averaged observations and predictions for NO_2 at station EE0018Ah over this period.

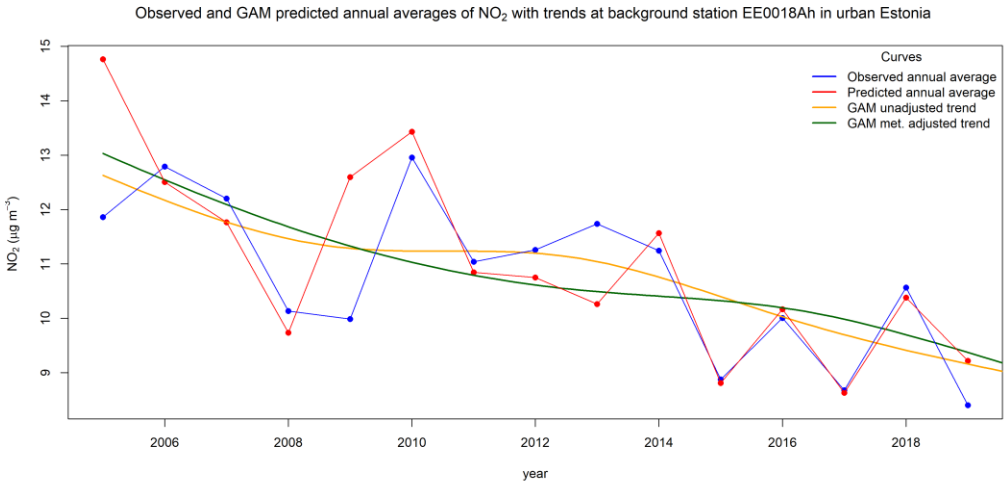

Figure S5. Observed (blue curve) and predicted (red curve) annual average concentrations of NO_2 at station EE0018Ah for 2005-2019. The orange and dark green curves, respectively, show the unadjusted and meteorology-adjusted trends.

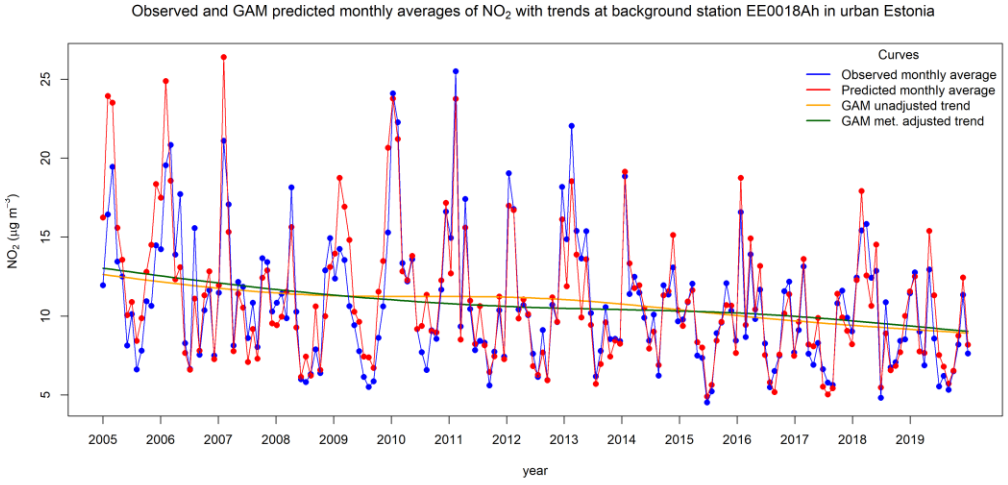

Figure S6. Observed (blue curve) and predicted (red curve) monthly average concentrations of NO_2 at station EE0018Ah for 2005-2019. Again, the orange and dark green curves show the unadjusted and meteorology-adjusted trends, respectively.

Examples of plots of annual and monthly medians of observed and predicted concentrations are shown in Figs. S7-S8. For the median plots, the trend curves are not plotted.

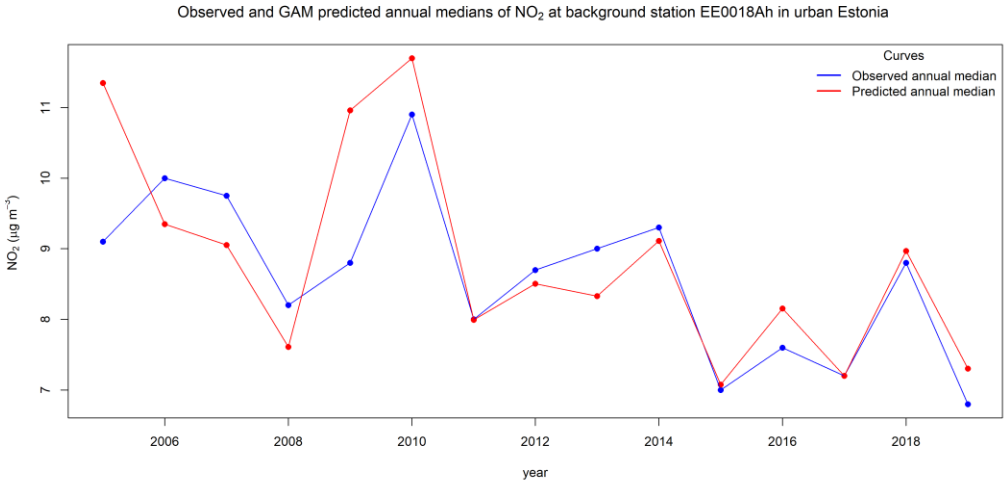

Figure S7. Observed (blue curve) and predicted (red curve) annual median concentrations of NO_2 at station EE0018Ah for 2005-2019.

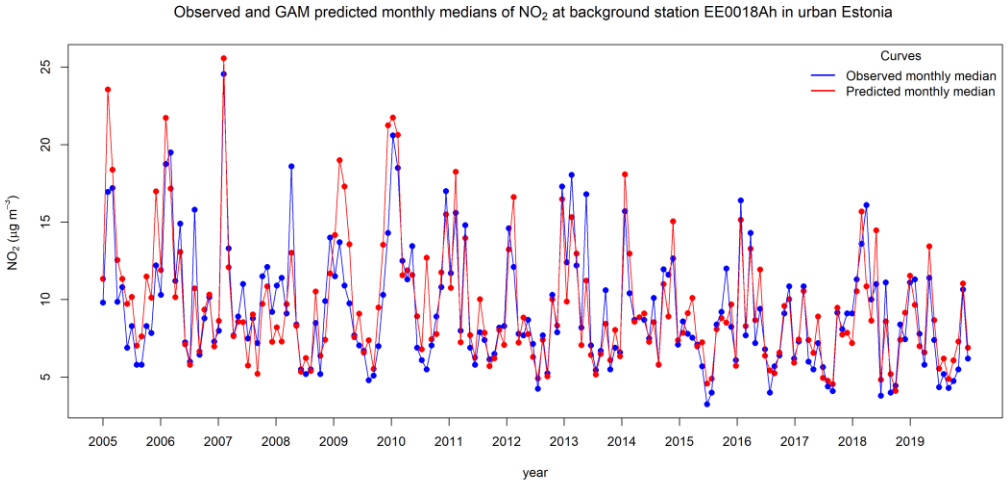

Figure S8. Observed (blue curve) and predicted (red curve) monthly median concentrations of NO_2 at station EE0018Ah for 2005-2019.

Again, there is overall a good correspondence between median values of observed and predicted concentrations of NO_2 at station EE0018Ah over this period.

S5.1.10 Values of (sub-) annual and monthly averages and medians of observations and model predictions

The observed and predicted (sub) annual and monthly averages with trend curve values are also written in text files. The file names are `<station>_gam.aave_<ya>_<yb>.csv` and `<station>_gam.mave_<ya>_<yb>.csv`, respectively. Each row of the (sub) annual averages file contains the year (`year`), the observed (`obs.aave`), and predicted (`pre.aave`) (sub) annual averages for each year, followed by the trend curve values (`trend.<adj>`), with the year ranging from `<ya>`

to <yb>. Likewise, each row of the monthly averages file contains the year (year) and month (month), the observed (obs.mave) and predicted (pre.mave) monthly averages for each year, followed by the trend curve values (trend.<adj>), again with the year ranging from <ya> to <yb>. Missing values (NA) are inserted in months outside the defined sub-year period (<ma> – < mb>) or outside the period for cross-validation (<yc> – <yd>). Similarly, in text files, observed and predicted (sub) annual and monthly medians are written. In this case, the string ave in the filenames and the headings in the files are replaced by med. No trend values are written in this case.

S5.1.11 Processed stations

The file name is AirGAM_stations.csv. This CSV file contains a list of all stations processed by the AirGAM model when run. Each row of the file includes the station name acronym (name), longitude (lon), latitude (lat), height above sea level (z in m), station type (type), and station area characteristics (area). Here type is a text string describing the station type (background or traffic). The area is a text string describing the station's surrounding area (rural, suburban or urban). Only stations the model actively processes are listed in the file. It will thus contain a subset of the stations in the input stations file described in Sect. S3.4.

S5.1.12 Program log-file

The file name is AirGAM_log.txt. This is a text file containing statuses and eventual warnings and errors produced by the AirGAM model when run. Status messages include model version, time and date of the model run, details of the run environment such as OS version, machine and user information, R and R packages versions, the working directory, top input/output directories, most of the options used, and major milestones reached during execution. Lines with warning/error messages contain the warning/error code, the station name acronym, the current date (year, month, day) of the data processed, and some explanatory text. A list of the various types of warnings and errors issued by the model, with each description, is given in Appendix B.

S5.2 Deterministic model evaluation

Below we describe the result files from the deterministic model evaluation part of the AirGAM model.

S5.2.1 Model summary

The file name is <station>_gam.summary_<adj>_<ya>_<yb>.txt. This is a text file containing the results of running the summary.gam function in the mgcv package (Wood, 2017) in connection with a GAM model run for the whole period for the trend analysis (<ya>-<yb>). This file contains first the name of the response distribution (normal or gamma), the type of link function used (identity or log), the formula used in the call to the GAM model solver (bam or gam), and the results for the intercept (estimate, standard error, t-value, and significance probability). Then for each smooth covariate in the

GAM model, the empirical degrees of freedom (edf), the reference degrees of freedom (ref.df), the F-value (F), and the p-value (p) are given, together with the corresponding significance codes. Finally, the file contains the adjusted R^2 value, the percentage of deviance explained (dev.expl), the penalised likelihood final objective function value (fREML), the scale estimate (scale), the number of data values (n), and the residual degrees of freedom (res.df).

All covariates for NO2 at station EE0018Ah for 2005-2019 are highly significant, with p-values very close to zero. The R^2 value is 0.693, which means that around 69 % of the variation in the concentrations can be explained by the covariates, which is quite good. Also, the empirical degrees of freedom value edf for each covariate is well below the corresponding reference degrees of freedom value ref.df, except perhaps for the covariate dayofweek, but this represents only a minor issue here. However, if edf should become close to ref.df for the trend term, one should consider increasing the number of basis functions for the trend term, especially if one wants to capture more of the variation in the trend. This can be done through the control variable k_years.

S5.2.2 Model check plots

The file name is <station>_gam.check_<adj>_<ya>_<yb>.png. An example of this type of plot is shown in Fig. S9.

This panel of plots shows various evaluation plots for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2005-2019 (<ya>-<yb>) produced by the gam.check routine in the mgcv package. The upper left plot shows the model residual quantiles against theoretical quantiles based on a normal distribution assumption for the residuals. The black data points corresponding to the individual residual values should follow the straight red line for a good model fit. The upper right plot shows model residuals against the model linear predictor. Ideally, the individual data points (circles) should have the same distribution along the y-axis for all x-axis values. The lower left plot shows a frequency histogram of the model residuals. Ideally, the histogram should be symmetric and normal in shape. And finally, the lower right plot shows the response, i.e., the observed concentrations, against the model-fitted values. Ideally, the data points (circles) should be as close as possible to a 1:1 reference line through the origin.

As shown in Fig. S9, we see that for NO_2 at station EE0018Ah for 2005-2019, the model residual quantiles (upper left plot) follow the theoretical quantiles of a normal distribution quite well except for the upper tail part, where there is a certain deviation. The other plots in this panel show excellent results, with the ideal type of plots in all cases.

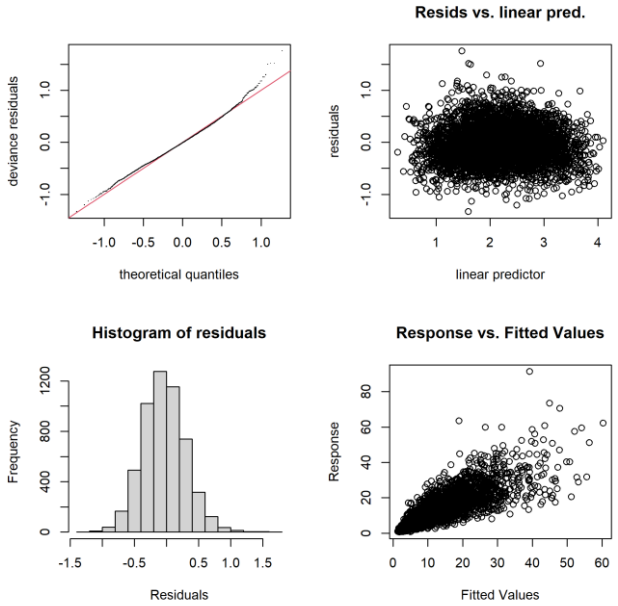

Figure S9. Model check plots for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2005-2019.

S5.2.3 Model check table

The file name is `<station>_gam.check_<adj>_<ya>_<yb>.txt`. This text file contains additional data output from the `gam.check` routine in `mgcv`. This file includes details from the convergence of the numerical solution method used for the GAM model. The most important output, however, is a table which, for each smooth covariate function, shows the number of basis functions minus 1 (`k'`), the empirical number of degrees of freedom (`edf`), the k-index value (`k-index`), and the associated p-value (`p-value`). The user should check this table for any low `p-value` (< 0.05) with a `k-index` < 1 to ensure that the `edf` value is not too close to the `k'` value.

For NO_2 at station EE0018Ah for 2005-2019, the model converged in 14 iterations with an objective function gradient close to zero and a positive definite Hessian matrix. The basis dimension checking results are all ok for the covariates with high p-values, except for the trend term, where the p-value is very small ($< 2 \cdot 10^{-16}$). Note that the p-values are not associated with the significance of covariates here and should be high for all variables. However, for the trend term, `edf`=3.38 and well below `k'`=4, and thus, the number of basis functions is sufficiently large also for this variable.

S5.2.4 Model evaluation

The file name is `AirGAM_gam.eval_<yc>_<yd>.csv`. This is a CSV file containing the results of evaluating the model predictions against observations from the cross-validation period (`<yc>-<yd>`) using the routine `modStats` from the

openair package in R. This is also a file common to all stations with a header line and one row of results per station. Each row contains the station name acronym (name), the number of cases (days) used for the model evaluation (n), and then the following model evaluation statistics: fraction of predictions within a factor of 2 of observations (fac2), mean bias (mb), mean gross error (mge), normalised mean bias (nmb), normalised mean gross error (nmge), root mean squared error (rmse), Neyman-Pearson correlation coefficient (r), coefficient of efficiency (coe), and index of agreement (ioa). The manual pages for the modStats routine in openair contain a detailed description of what these parameters represent and how they are calculated.

For NO_2 at station EE0018Ah for 2005-2019, the evaluation results are based on 5398 cases (days). In around 94 % of these days, the predicted concentrations are within a factor of 2 of the observations (fac2=0.944). Further, the mean bias (mb) is only around 0.25 µg m^{-3} with a normalised mean bias (nmb) of only 0.023, which is quite good. The root mean squared error (rmse) is also relatively low, with a value of 4.8 µg m^{-3}. Also, the correlation coefficient (r) and index of agreement (ioa) are pretty good, with values of 0.82 and 0.75, respectively. Finally, the coefficient of efficiency (coe) also shows a decent value of around 0.5 for this compound and station.

S5.2.5 Concurvity analysis

The file name is AirGAM_gam.ccuv_<adj>_<ya>_<yb>.csv. This is a CSV file containing so-called concurvity values for each smooth covariate in the AirGAM model based on all years used for the trend estimation (<ya>-<yb>). This is a common file for all stations with a header line and one row of results per station. Concurvity is to GAM modelling as collinearity is to multiple linear regression; it describes the degree to which covariates can be viewed as independent of each other. More specifically, for GAM models, the concurvity value for a given smooth covariate indicates to what degree this covariate is superfluous and could be replaced by a linear or nonlinear combination of the remaining smooth covariates in the model. It is thus important to check for this as part of the modelling. Concurvity values are calculated using the concurvity routine in the mgcv package and range from 0 (best value) to 1 (worst value). Each row of the result file contains the station name acronym (name), type of concurvity value (type), followed by a concurvity value for each smooth covariate (ccuv.beta<i>) for <i>=1,2,… .

A concurvity value of type worst from the concurvity routine below 0.8 (approximately) is often taken to indicate that the corresponding smooth covariate is probably not severely dependent on the other smooth covariates (Ross, 2022). A higher value is more troublesome and suggests that it might be redundant and replaced by a linear or nonlinear combination of the other smooth covariates. In this case, the covariate response function will be challenging to estimate appropriately due to identifiability problems. However, this is a relatively pessimistic measure of concurvity according to the help pages for the concurvity routine in mgcv. There the estimate type of concurvity is presented as somewhat better balanced than the

other two, i.e., `worst` and `observed`, since "It does not suffer from the pessimism or potential for over-optimism of the previous two measures", even though, as also stated, that it is "less easy to understand". Thus, due to this better balance, we apply this measure of concurvity in AirGAM rather than the overly pessimistic one. However, we reduce the limit to 0.4 to indicate potential problems with identifiability. For values above this, a warning is issued to the log file. If seasonal conditioning is used (`use_season_cond=1`), separate concurvity values will be output to this file for each season. This will be indicated in the header line.

For NO_2 at station EE0018Ah for 2005-2019, the concurvity values for the various smooth covariates are all small (below 0.4), which is good and indicates that they are all reasonably independent.

S5.2.6 Autocorrelation and partial autocorrelation function plots

The file name is `<station>_gam.acf_<adj>_<ya>_<yb>.png`. An example of this type of plot is shown in Fig. S10.

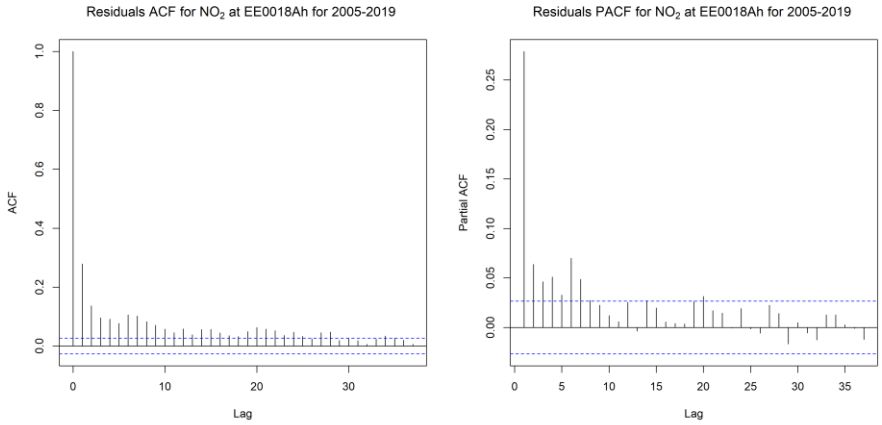

Figure S10. A plot of the autocorrelation function (left) and partial autocorrelation function (right) for the meteorology-adjusted model error residuals for NO_2 at station EE0018Ah for 2005-2019.

This figure shows plots of the autocorrelation function (left) and the partial autocorrelation function (right) for the meteorology-adjusted model error residuals for NO_2 at station EE0018Ah for 2005-2019. Ideally, the GAM model residuals should be independent random variables; thus, the autocorrelation function values should be close to zero for all positive time lags. The same applies to the partial autocorrelation function values; they should also be close to zero for all positive time lags. The level below which the autocorrelation values are non-significant (close to zero) is indicated by the horisontal dashed line(s).

As shown in Fig. S10, autocorrelation values for this compound and station are significantly positive from time lag one onwards, decaying slowly with the time lag. For the partial autocorrelation, the lag-1 value is the most significant, with a value of around 0.25, while the other values are much smaller (although a few significantly different from zero).

990 Running the `gamm` routine, in this case, using the option `incl_ar1=1`, handles autocorrelations by including an AR(1) model for the residuals. This results in (nearly) non-significant correlations at all time lags, as shown in the plots to the right in Fig. S11.

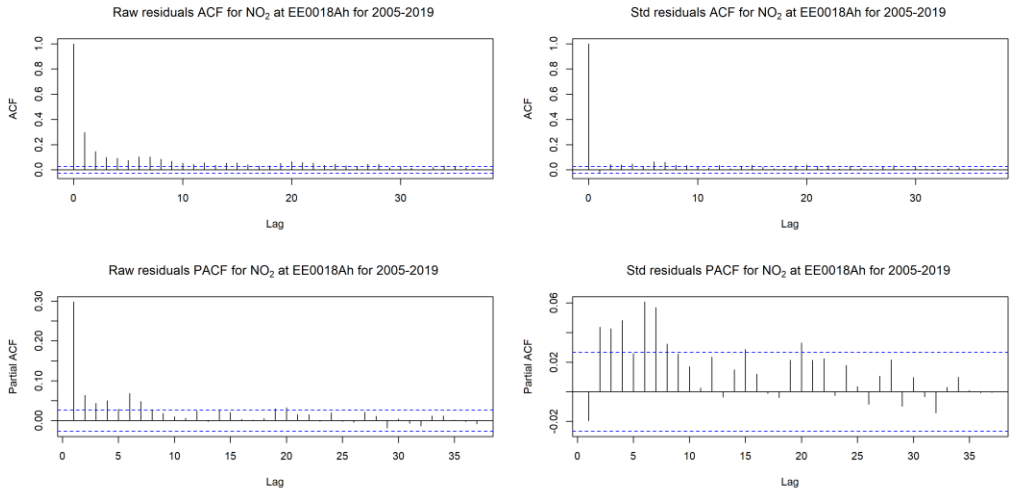

Figure S11. The left plots are as in Fig. S10, while the right plots show the effect of including an AR(1) model for the residuals.

995 **S5.2.7 Autocorrelation and partial autocorrelation function values**

The file name is `AirGAM_gam.acf_<adj>_<ya>_yb>.csv`. This is a CSV file containing the autocorrelation and partial autocorrelation values for the residuals for the first 10 lags (days) based on the fitted model for all years used for the trend estimation (`<ya>-<yb>`). It is a file common to all stations with a header line and one row of results per station. Each row of this file contains the station name acronym (`name`), lag-1 to lag-10 autocorrelation values (`acf.1`,…, `acf.10`), and lag-1 to
1000 lag-10 partial autocorrelation values (`pacf.1`,…, `pacf.10`). Ideally, all these values should be zero or close to zero, corresponding to independent or nearly independent model error residuals.

S5.2.8 A conditional quantile plot

The file name is `<station>_gam.cond_quant_<yc>_<yd>.png`. An example of this type of plot is shown in Fig. S12.

1005

This is a so-called conditional quantile plot for the model here shown for NO_2 at station EE0018Ah for the cross-validation years 2005-2019 (`<yc>-<yd>`). It is produced by the `conditional.Quantile` routine in `openair`. The plot shows the meteorology-adjusted model prediction quantiles against the observed concentration quantiles. The median of the model quantiles is shown as the dark red curve, while 25/75 and 10/90 percentiles are shown as the light yellow and orange-brown

shaded regions. Ideally, the dark red curve should perfectly follow the straight light blue line. The background shows histograms of the model predictions in dark grey and histograms of the observations in light blue. Ideally, these two histograms should be identical.

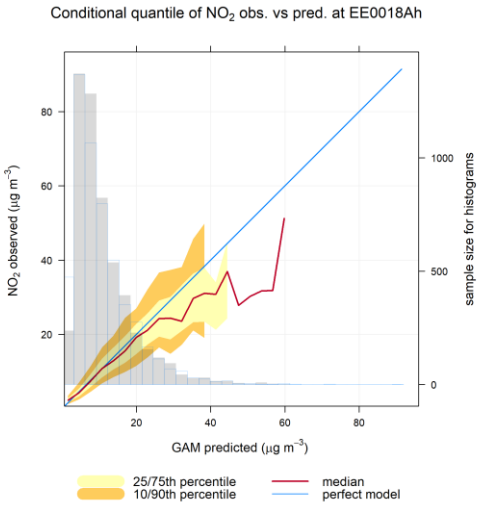

Figure S12. Conditional quantile plot for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2005-2019.

As shown in Fig. S12, the median of model predicted quantiles follows the observed ones almost perfectly up to around 25 µg m^{-3} before the two start to deviate. But the 25/75 percentile light-yellow region of the model predicted quantiles still contains the observed concentration quantiles (straight light blue line) for all values up to around 40 µg m^{-3}, which is good. For the higher concentrations, the quantiles deviate more. We also note that the two histograms are similar, which is good.

S5.2.9 Taylor diagram plot

The file name is `<station>_gam.taylor_<yc>_<yd>.png`. An example of this type of plot is shown in Fig. S13.

This is a so-called Taylor diagram plot for the model here shown for NO_2 at station EE0018Ah for the cross-validation years 2005-2019 (`<yc>-<yd>`). The TaylorDiagram routine produces it in `openair`.

As shown in Fig. S13, the model point (red dot) is not very far from the ideal observed point (purple dot). More specifically, the model point is in the sector between the correlation levels of 0.8 and 0.9 and is quite close to the dashed black curve (circle) emanating from the observed point. Further, the dashed orange-brown lines indicate an RMSE value between 4 and 6 µg m^{-3}.

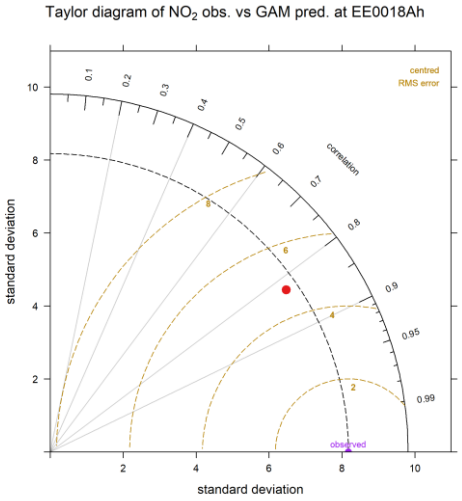

Figure S13. Taylor diagram plot for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2005-2019.

S5.3 Probabilistic model evaluation

Since model predictions from AirGAM are point predictions and come with an associated probability distribution for each predicted concentration value, it is also possible to evaluate the model against observations using probabilistic tools and concepts. Gneiting et al. (2007) and Wilks (2019, Ch. 9) describe this way of assessing a prediction model thoroughly. Below we describe the result files from AirGAM for this model evaluation. Note that these results are only produced from the meteorology-adjusted model predictions in AirGAM.

S5.3.1 PIT histogram

The file name is `<station>_gam.pit_hist_<yy>_<yy>.png`. An example of this type of plot is shown in Fig. S14.

This is a so-called PIT (Probability Integral Transform) histogram plot for the model shown here for NO_2 at station EE0018Ah for 2019 (`<yy>=2019`). The plot shows a histogram of the observed concentrations compared with the model probabilistic predictions converted into corresponding probability values between 0 and 1.

The PIT value at day t is simply the value of the GAM model predictive distribution (CDF) F_t for this day, at the observed concentration the same day y_t, i.e., $\mathrm{PIT}_t = F_t(y_t)$. We can plot a histogram of these values by collecting PIT values over a certain period, e.g. a year. The PIT histogram can be viewed as a continuous limit of the rank histogram, where the latter is based on a finite set of samples from the predictive distribution (Gneiting et al., 2007; Wilks, 2019, Ch. 9).

In the figure, the predictions are based on training the model on all years for the trend estimation (2005-2019) except for the plotted year (2019).

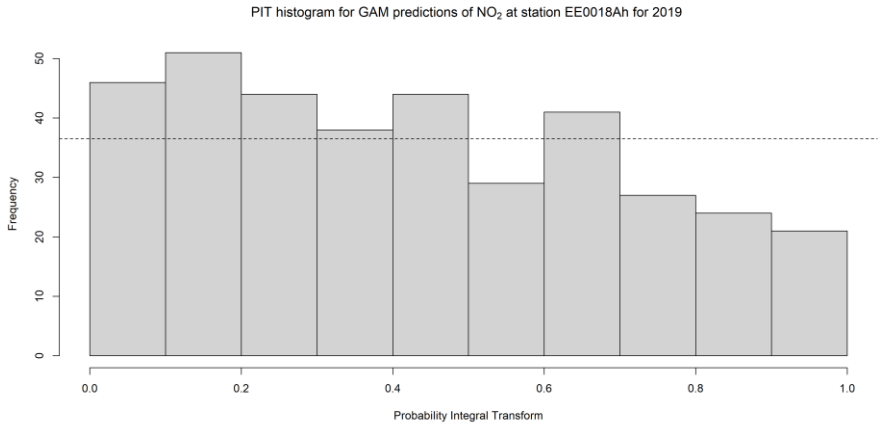

Figure S14. PIT histogram plot for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2019.

Since the predictive distributions cannot be represented analytically (see Sect. S2.4), the calculations of the PIT values in AirGAM are generally based on taking a sufficiently large number of samples from the unconditional predictive distribution for each day (currently 100) and calculating an empirical cumulative distribution probability value using the corresponding observed value. Ideally, the PIT histogram should be uniform if the model's predictions are properly probabilistically calibrated relative to the actual observations. If the model predictions are too low, the histogram will be biased (skewed) to the right; if they are too high, it will be biased (skewed) to the left. Also, if the predictions are too narrow (too low prediction uncertainty), the histogram will be U-shaped, while it will have an inverse U-shape if the predictions are too broad (too high prediction uncertainty).

As shown in Fig. S14, the histogram is somewhat biased and skewed to the left, i.e., lower PIT values than high. This means that the model predictions for the station EE00a8Ah for 2019 seem too high compared with the observations.

The plot always shows PIT values on the x-axis and frequency on the y-axis, and the horisontal dashed line corresponds to a uniform histogram. According to Gneiting et al. (2007), 10-20 bins used to define a PIT histogram seem sufficient for most purposes. We apply 10 bins in our implementation generally.

S5.3.2 Empirical CDF of PIT values

The file name is `<station>_gam.pit_ecdf_<yy>_<yy>.png`.. An example of this type of plot is shown in Fig. S15.

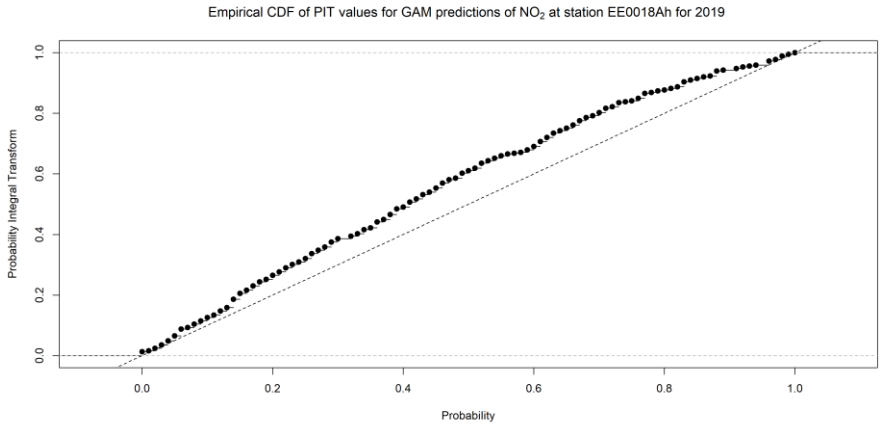

Figure S15. The plot of the empirical cumulative distribution function of the PIT values for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2019.

This is a plot of the empirical cumulative distribution function (CDF) of the PIT values for the model here shown for NO_2 at station EE0018Ah for 2019 (`<yy>=2019`). The plot always shows theoretical cumulative probability values on the x-axis and PIT cumulative probability values on the y-axis. Ideally, the empirical CDF of the PIT values should stay close to the ideal 1:1 dashed reference line if the predictions from the model are correctly probabilistically calibrated relative to the actual observations (Gneiting et al., 2007). If the model predictions are too low, the CDF values will tend to lie below the 1:1 line, while if predictions are too high, the CDF values will tend to lie above the 1:1 line.

From Fig. S15, the CDF values all lie above the 1:1 line. This indicates that the model predictions are too high compared to the observations for this station and year.

S5.3.3 Marginal empirical CDFs of observations and predictions

The file name is `<station>_gam.marg_ecdf_<yy>_<yy>.png`.. An example of this type of plot is shown in Fig. S16.

This is a plot of the marginal empirical CDFs of observed (blue curve) and predicted (red curve) values for the model here shown for NO_2 at station EE0018Ah for 2019 (`<yy>=2019`). Ideally, the two marginal empirical CDFs should stay close together if the model's predictions are properly marginally calibrated relative to the actual observations (Gneiting et al., 2007). The plot always shows concentration values on the x-axis and marginal CDF probability values on the y-axis.

From Fig. S16, we can see that the marginal CDF probabilities for the observations are generally higher than the marginal CDF probabilities for the predictions for all concentration levels except for concentrations above around 30 µg $m^{-3,}$ where the curves are pretty close. This again shows that the model predictions are too high compared with the observations, also marginally, for this station and year, except for the highest concentrations.

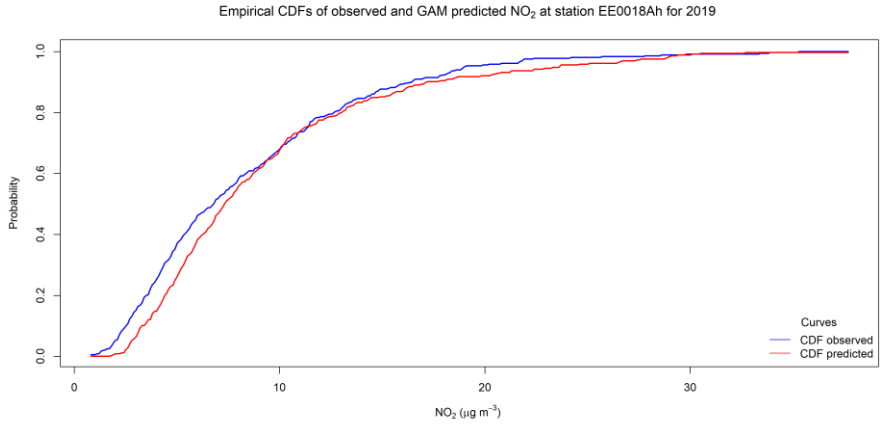

Figure S16. The plot of the marginal empirical CDFs of observed and predicted values for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2019.

S5.3.4 Sharpness diagram

The file name is `<station>_gam.sharp_<yy>_<yy>.png`. An example of this type of plot is shown in Fig. S17.

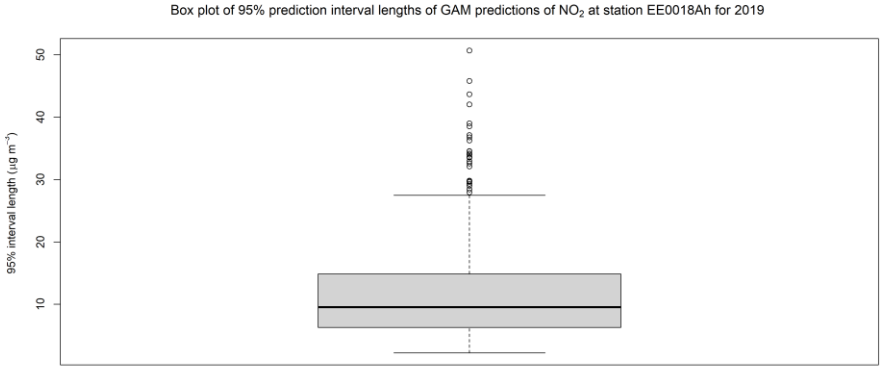

Figure S17. A sharpness diagram box plot of 95 % uncertainty intervals of predictions for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2019.

This is a so-called sharpness diagram in the form of a box plot of 95 % uncertainty interval lengths of predictions from the model here shown for NO_2 at station EE0018Ah for 2019 (`<yy>=2019`). Ideally, such box plots should be relatively tight if the model produces sharp predictions, i.e., predictions with low uncertainties (Gneiting et al., 2007). The plot always shows 95% uncertainty interval lengths on the y-axis.

From Fig. S17, we can see that the model predictions have 95 % uncertainty intervals of length around 10 µg m^{-3} on average, with 50 % of the interval lengths between 8-13 µg m^{-3}. Only occasionally are the interval lengths above 28 µg m^{-3}. Thus the model predictions are reasonably sharp overall for this station and year.

S5.3.5 CRPS scatter plots

The file name is `<station>_gam.crps_<yy>_<yy>.png`. An example of this type of plot is shown in Fig. S18.

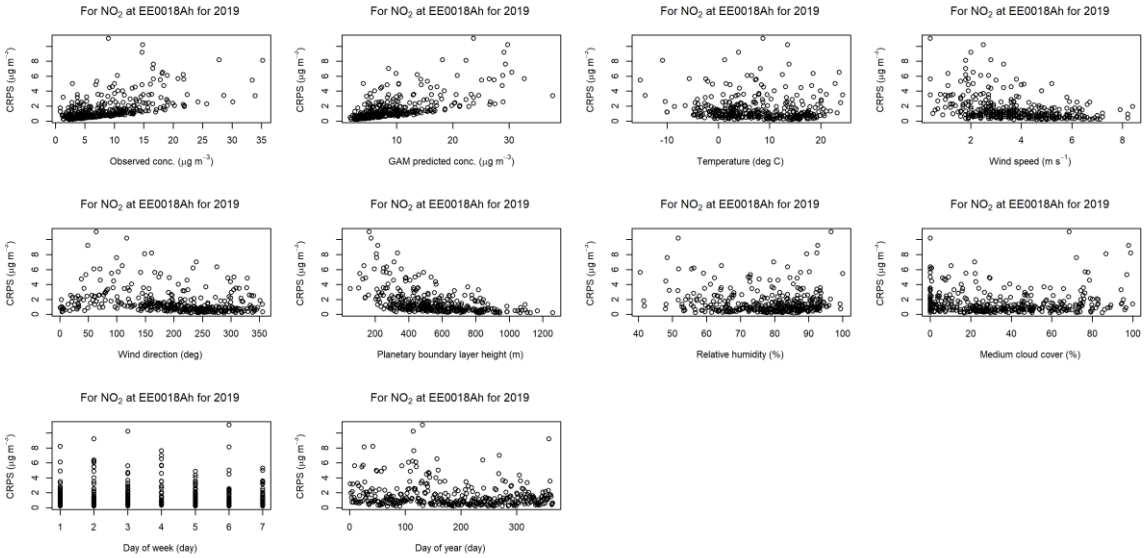

Figure S18. Scatter plots of daily CRPS values against observations, model predictions and covariates for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2019 (`<yy>=2019`).

The figure shows scatter plots of daily CRPS (Continuous Ranked Probability Score) values against daily values of observations, model predictions and covariates for the model for NO_2 at station EE0018Ah for 2019 (`<yy>`= 2019).

The Continuous Ranked Probability Score (CRPS) (Gneiting and Raftery, 2007; Wilks, 2019, Ch. 9) is a numerical measure of a model's predictive performance considering both calibration and sharpness. It is calculated from the model's predictive distribution and daily observed values.

The CRPS at day t is defined as follows from the GAM model's predictive distribution (CDF) F_t and the observed daily mean concentration y_t on the same day:

$$\mathrm{CRPS}_t = \mathrm{CRPS}(F_t, y_t) = \int_{y=0}^{y=\infty} \left\{ F_t(y) - 1(y \geq y_t) \right\}^2 dy \,,$$

where $1(y \geq y_t)$ denotes the indicator function, i.e.,1 if $y \geq y_t$ and 0 otherwise.

For CRPS, smaller values are better, the optimal value being zero, which corresponds to a predictive distribution placed precisely at the observation value with no spread. Smaller values of CRPS correspond to predictive distributions being close to the observations, with a small spread. In comparison, larger values indicate the opposite, either through a significant bias between the prediction and the observation (poor calibration) or the predictive distribution has a large spread around the observation (poor sharpness). The CRPS always has the same unit as the concentrations, i.e., µg m^{-3}.

As shown in Fig. S18, the daily CRPS values generally increase with increasing observations and model predictions (first two scatter plots in the top row). We also see that the CRPS values are pretty even with temperature (following plot in the top row). Further, CRPS is highest for the lower wind speeds with wind directions from the east (50°-100°). This is related to situations with relatively low planetary boundary layer heights (below around 300 m). There is no clear pattern for relative humidity and medium cloud cover, but the highest CRPS values seem to occur during wintertime and spring of 2019. Thus, during these conditions, the model has more difficulties accurately predicting observed concentrations of NO_2 at this station.

S5.3.6 CRPS box plots

The file name is `<station>_gam.crps_<yc>_<yd>.png`. An example of this type of plot is shown in Fig. S19.

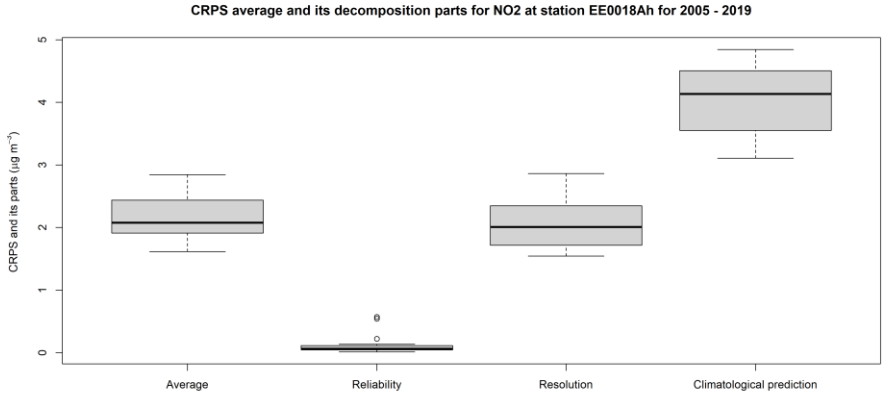

Figure S19. Box plots of CRPS averages and their reliability, resolution and climatological prediction uncertainty parts for the meteorology-adjusted model for NO_2 at station EE0018Ah for 2005-2019.

The figure shows box plots of CRPS annual averages with their reliability, resolution and climatological prediction uncertainty parts for the model predictions of NO_2 at station EE0018Ah for the cross-validation years 2005-2019 (`<yc>-<yd>`).

According to Hersbach (2000), an average CRPS over a given period (e.g. a year) can be partitioned into a reliability part, a resolution part, and a climatological prediction uncertainty part as follows:

$$\mathrm{CRPS}_{\mathrm{aver}} = \mathrm{Reli} - \mathrm{Reso} + \mathrm{CRPS}_{\mathrm{clim}} .$$

Here the reliability part is closely connected to the probabilistic calibration condition, i.e., the uniformity of rank or PIT-histograms. In contrast, the resolution and climatological prediction uncertainty are related to the sharpness of the predictive distributions (average spread or width). The climatological prediction uncertainty part is the value of $\mathrm{CRPS}_{\mathrm{aver}}$ if we only use the overall observed climatology based on the observations as the predictive distribution for each time instance, e.g. in our case, day. In this case, there will be zero reliability and resolution. The reliability part is a nonnegative quantity, with $\mathrm{Reli} = 0$ only for a perfectly reliable system, i.e., a system that is probabilistically calibrated with a uniform rank or PIT-histogram, which will be the case for predictions based on the above-observed climatology. However, such a predictive system will have zero resolution, i.e. $\mathrm{Reso} = 0$, i.e., no sharpness, since all predictions will be based on the same (average) climatology.

We may, however, achieve lower values of $\mathrm{CRPS}_{\mathrm{aver}}$ for predictive systems with $\mathrm{Reli} - \mathrm{Reso} < 0$. The optimal case will be obtained if we use perfect deterministic point predictions. In this case, the system will still be perfectly reliable, i.e., $\mathrm{Reli} = 0$, corresponding to a uniform rank or PIT-histogram. In contrast to the climatological system, it will have an optimal positive resolution (sharpness) in the sense that $\mathrm{Reso} = \mathrm{CRPS}_{\mathrm{clim}}$, with a resulting value of $\mathrm{CRPS}_{\mathrm{aver}} = 0$. Generally, we will obtain values of reliability and resolution between the above two extremes, i.e., $-\mathrm{CRPS}_{\mathrm{clim}} \le \mathrm{Reli} - \mathrm{Reso} \le 0$, and thus $0 \le \mathrm{CRPS}_{\mathrm{aver}} \le \mathrm{CRPS}_{\mathrm{clim}}$. An excellent predictive system is hence characterised as one having a small (positive) value of reliability, and a high (positive) value of resolution, resulting in a small (positive) value of $\mathrm{CRPS}_{\mathrm{aver}}$.

As shown in Fig. S19, the model predictions are highly reliable for most years, with reliability values close to zero except for a few cases. Also, for most years, we have a reasonably high degree of resolution (around 2), reducing the climatological prediction uncertainty from about 4 to about 2 for the CRPS average for the predictive model at this station for the whole period 2005-2019. Note that all data in the box plots have the same unit as for concentration, i.e., µg m^{-3}.

S5.3.7 CRPS box plots data

The file name is `<station>_gam.crps_<yc>_<yd>.csv`. The data from each box plot in Sect. S5.3.6 is written in this file. Each row of the file contains the year, followed by the CRPS average, reliability, resolution and climatological prediction uncertainty parts of the CRPS average for that year. There is one line of data for each year in the cross-validation period `<yc>`–`<yd>`, and all numbers have the same unit as for concentration, i.e., µg m^{-3}.

S5.3.8 CRPS box plots median values

The file name is `AirGAM_gam.crps_<yc>_<yd>.csv`. The median values from each box plot in Sect. S5.3.6 are written in this file. Each row of the file contains the station name, followed by the median values of the CRPS averages and the reliability, resolution, and climatological prediction uncertainty parts of the CRPS averages over the years for the cross-validation `<yc>-<yd>`. Again, the numbers' units are the same as for concentration, i.e., µg m^{-3}.

For NO_2 at station EE0018Ah and cross-validation years 2005-2019, we obtain the values of 2.08, 0.06, 2.01 and 4.12 as median values of CRPS average, reliability, resolution, and climatological prediction uncertainty, respectively. Thus, at the median, the model predictions are highly reliable (reliability value close to zero), with a relatively high degree of resolution (2.01), reducing the climatological prediction uncertainty from 4.12 to 2.08 for the predictive model at this station for the whole period 2005-2019.

S6 Some run examples

In this section, we provide some examples of AirGAM model runs. The data used in these run examples can be downloaded from the Zenodo data repository for the model (Walker and Solberg, 2022a-b). In particular, we focus on the stations in the EEA 2005-2019 trend study with median performance regarding cross-validation correlation results for each compound (NO_2, O_3, PM_{10} and $PM_{2.5}$) which means that half of all stations had a poorer correlation than this and half had a better for the respective compound. Thus, the stations and results should be representative for each compound, neither too good nor too bad.

S6.1 NO_2

In this case, the median station is the Estonian station EE0018Ah, as used to illustrate the result files in Sect. S5. Input data and all result files for this station can be found in the Zenodo data repository (Walker and Solberg, 2022a), which we recommend downloading.

After unzipping the files, the input data can be found in the `airgam_2022r1_input_EE0018Ah` directory, under the `no2` directory and each year directory `2005-2019`. The file `stations.txt` contains the static data for this station, the same file repeated for each year, while the files `EE0018Ah_no2_<year>.txt` includes the measurement data of NO_2 and meteorological data individually for each year `<year>`. These files can easily be viewed in any text editor.

The options file is the file `airgam_options.txt`. This is a text file where all the options, i.e., variables used to control the run, are defined. In this file, the compound is set by the statement `comp=no2`. The period used for trend estimation is defined by `yyyy_a=2005` and `yyyy_b=2019`. Cross-validation for the whole period is chosen by `yyyy_c=2005` and

yyyy_d=2019. The entire year is modelled by setting subyear=jan-dec. The other control variables can be altered as described in Sect. S3.3, but this is usually not necessary.

The scripts used to run the model on Windows and Linux are the files airgam_run.bat and airgam_run.sh, respectively. The user needs to edit the two variables R_script and wrkdir in these files to reflect the user's environment before running the scripts. The scripts are run as described in Sects. S4.1 and S4.2, respectively.

The result files from the run are described in Sect. S5 and will thus not be repeated here. All the files can be found in the airgam_2022r1_results_EE0018Ah/no2_2005_2019_jan-dec directory under the main and eval sub-directories for main results and evaluation results, respectively. The AirGAM_log.txt file in the main directory contains a complete run log.

S6.2 O_3

In this case, the median station is the station CH0017Ah. This is a background station in Basel, Switzerland. The input data for this station is in Walker and Solberg (2022b), while all result files can be found in Walker and Solberg (2022a).

After unzipping the files, the input data can be found under the airgam_2022r1_input_all/o3 directory, while the results are in the airgam_2022r1_results_median/o3_2005_2019_apr-sep directory. The file stations.txt contains the static data for this station, the same file repeated for each year, while the files CH0017Ah_o3_<year>.txt includes the measurements of O_3 and meteorology individually for each year <year>.

For this run, the control variables comp and subyear in the options.txt file were changed to o3 and apr-sep, respectively. The latter option states that this compound's trend estimation will be only for the summer period. The rest of the options were unchanged. Figure S20 shows the combined results from the trend estimation and cross-validation.

The file name of this result is VH0017Ah_gam.aave_2005_2019.png. The plot shows observed (blue curve) and predicted (red curve) annual average concentrations of O_3 at station CH0017Ah for 2005-2019. The orange and dark green curves, respectively, show the unadjusted and meteorology-adjusted trends. In these plots, we use annual averages of the predictions from the cross-validation for all years used for the trend estimation. The year 2014 is missing from the cross-validation due to observations not fulfilling the data coverage criteria for this year (at least 75% coverage of daily measurements each year).

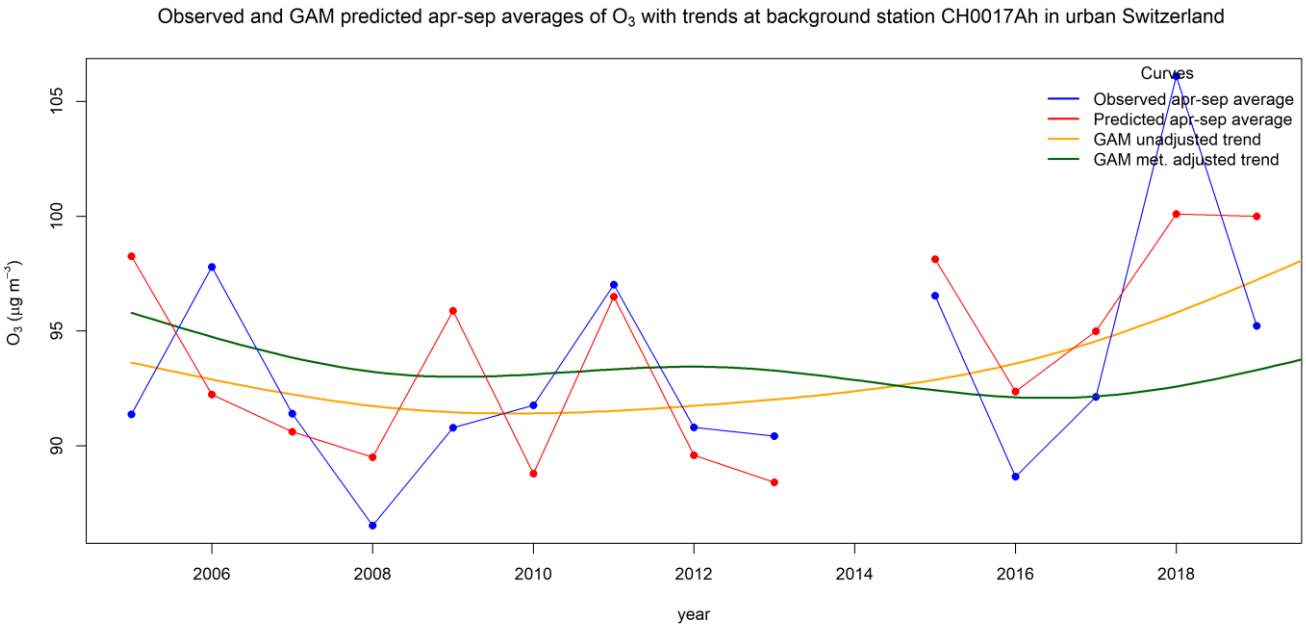

Figure S20. Observed (blue curve) and predicted (red curve) annual average concentrations of O_3 at station CH0017Ah for 2005-2019. The orange and dark green curves, respectively, show the unadjusted and meteorology-adjusted trends.

As shown in Fig. S20, the meteorology-adjusted trend (green curve) is relatively flat over the period. Overall, there is a good correspondence between the averaged observations and predictions for O_3 at this station which increases the trust in the trend results.

S6.3 PM_{10}

Here, the median station is the station DEBW029h. This is a background station in Baden-Württemberg, Germany. The input data for this station is in Walker and Solberg (2022b), while all result files can, as before, be found in Walker and Solberg (2022a).

After unzipping the files, the input data can be found under the `airgam_2022r1_input_all/pm10` directory, while the results are in the `airgam_2022r1_results_median/pm10_2005_2019_jan-dec` directory. The file `stations.txt` contains the static data for this station, the same file repeated for each year, while the files `DEBW029h_pm10_<year>.txt` contains the measurements of PM_{10} and meteorology individually for each year `<year>`.

For this run, the control variables `comp` and `subyear` were `pm10` and `jan-dec`, respectively, while the rest of the options were as before. Figure S21 again shows the combined results from the trend estimation and cross-validation.

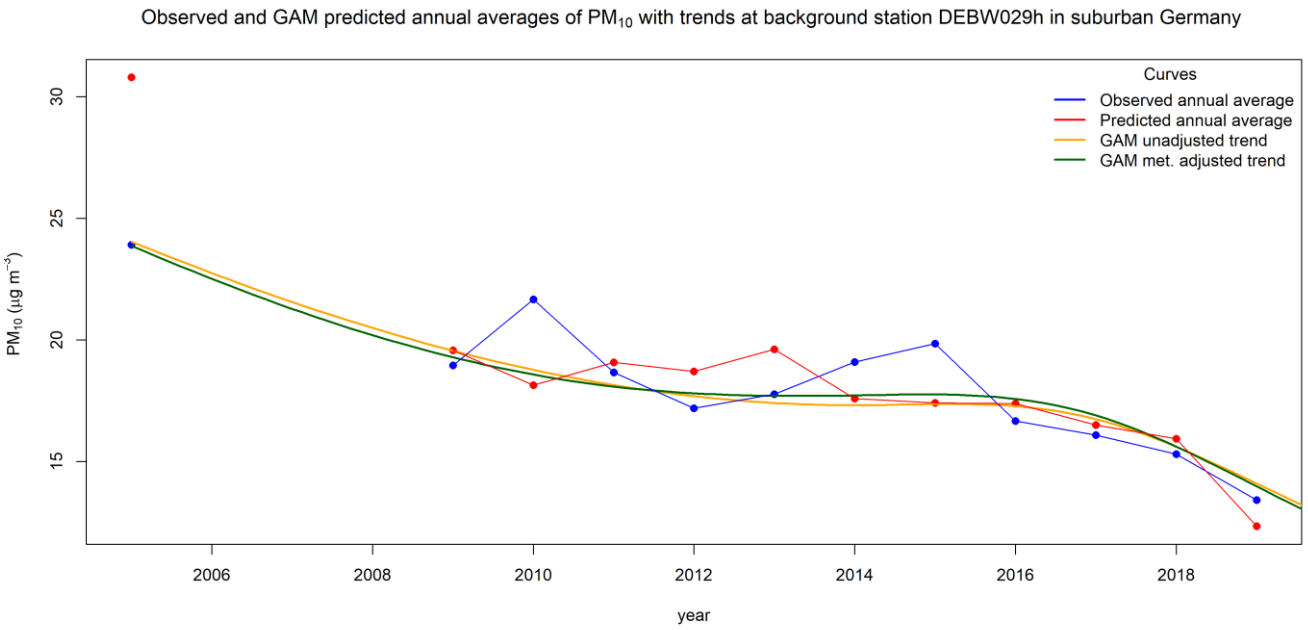

Figure S21. Observed (blue curve) and predicted (red curve) annual average concentrations of PM_{10} at station DEBW029h for 2005-2019. The orange and dark green curves, respectively, show the unadjusted and meteorology-adjusted trends.

The file name of this result is `DEBW029h_gam.aave_2005_2019.png`. The plot shows observed (blue curve) and predicted (red curve) annual average concentrations of PM_{10} at station DEBW029h for 2005-2019. The orange and dark green curves show, respectively, the unadjusted and meteorology-adjusted trends. In these plots, we use annual averages of the predictions from the cross-validation for all years used for the trend estimation. Here the years 2006-2008 are missing from the cross-validation due to not fulfilling the data coverage criteria (at least 75% coverage of daily measurements each year).

As shown in Fig. S21, the trend curves are very similar and generally fall over the period, with a flatter part from 2012-2016. There is overall a reasonably good correspondence between the averaged observations and predictions for PM_{10} at this station which again increases the trust in the trend results.

S6.4 $PM_{2.5}$

The median station for this compound is station DENW024d. This is a background station in Essen-Vogelheim, Germany. The input data for this station is in Walker and Solberg (2022b), while all result files can be found in Walker and Solberg (2022a).

After unzipping the files, the input data can be found under the `airgam_2022r1_input_all/pm2.5` directory, while the results are in the `airgam_2022r1_results_median/pm2.5_2005_2019_jan-dec` directory. The file `stations.txt` contains the static data for this station, the same file repeated for each year, while the files

DENW024d_pm2.5_<year>.txt contains the measurements of $PM_{2.5}$ and meteorology individually for each year <year>.

Figure S22 shows the combined results from the trend estimation and cross-validation. For this run, the control variables comp and subyear were pm10 and jan-dec, respectively. The rest of the options were as before.

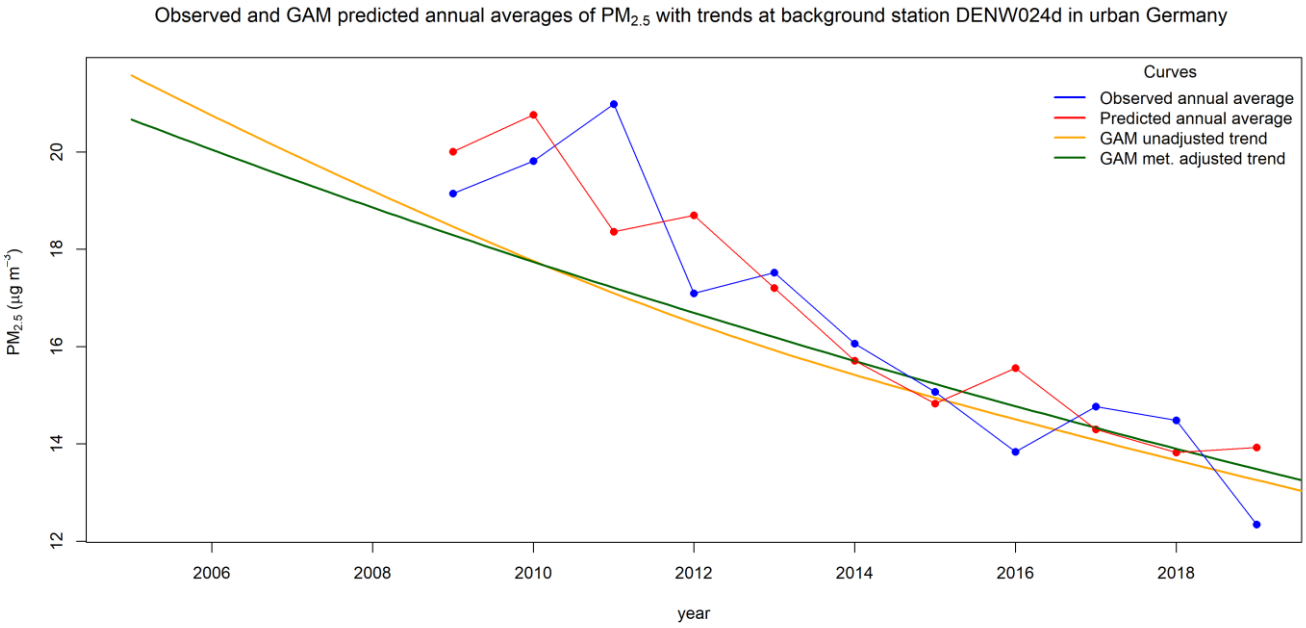

Figure S22. Observed (blue curve) and predicted (red curve) annual average concentrations of $PM_{2.5}$ at station DENW024d for 2005-2019. The orange and dark green curves, respectively, show the unadjusted and meteorology-adjusted trends.

The file name of this result is DENW024d_gam.aave_2005_2019.png. The plot shows observed (blue curve) and predicted (red curve) annual average concentrations of $PM_{2.5}$ at station DENW024d for 2005-2019. The years 2005-2008 are missing from the cross-validation due to not fulfilling the data coverage criteria (at least 75% coverage of daily measurements each year). The orange and dark green curves, respectively, show the unadjusted and meteorology-adjusted trends. In these plots, we use annual averages of the predictions from the cross-validation for all years used for the trend estimation.

As shown in Fig. S22, the trend curves are again very similar and generally fall over the period. Overall, there is a reasonably good correspondence between the averaged observations and predictions for $PM_{2.5}$ at this station which again increases the trust in the trend results.

S7 Code and data availability

The current version of the AirGAM model is available on the Zenodo repository (Walker, 2022a) under the GPL-2 licence. The exact version of the model (2022r1) used to produce the results used in this paper is archived on Zenodo (Walker, 2022b), as are input data and scripts to run the model and produce the plots for the results presented in this paper (Walker and Solberg, 2022a-b). The results for all individual stations and compounds can also be found on Zenodo (Walker and Solberg, 2022c-f).

Appendix A: Installing the AirGAM model

Here we describe installing the AirGAM model for Windows (Sects. A.1-A.3) and Linux (Sects. A.4-A.6).

A.1 System requirements for Windows

R (R Core Team, 2022) and AirGAM can be used on various versions of Windows. We recommend using later versions of Windows, preferably a 64-bit version and a computer with at least 16 GB of RAM. There is no specific requirement regarding disk space except that it should be sufficient to store R and its packages and the data files for AirGAM. The latter depends on the number of stations and years defined for the trend calculation, cross-validation, and the selected amount of output. The disk space used in our present study for European air quality stations from 2005-2019 was about 3.2 GB.

A.2 R and R packages for Windows

In the present study, we used the R 4.1.2 version. In addition to R, the AirGAM model relies upon the following R packages:

- `mgcv`
- `openair`
- `sandwich`

We recommend always using the latest version of R and installing the latest version of these packages.

A.3 Installing AirGAM for Windows

The latest version of the model can be downloaded from Zenodo (Walker, 2022a). The exact version used to produce the results in this paper (AirGAM 2022r1) can be downloaded from the same site (Walker, 2022b). The model is installed simply by copying the AirGAM R script to the same directory as the run scripts. The latter can be downloaded from Walker and Solberg (2022a) with input NO_2 data for station EE0018Ah and results for this and median stations for the other compounds.

A.4 System requirements for Linux

R and AirGAM can also be used on various versions of Linux. We have good experience running it on Ubuntu and Red Hat (CentOS). Again, the computer should have at least 16 GB of RAM, and the amount of disk space should be sufficient to store R and its packages and all data files for the model; see Sect A.1.

A.5 R packages for Linux

The AirGAM model relies upon the same R packages for Linux as for Windows; see Sect. A.2.

A.6 Installing AirGAM for Linux

Installing the AirGAM model on Linux is similar to installing it on Windows; see Sect. A.3. The same model R script, options file and input data files are used in both systems. However, the script files used to run the model differ as described in Sect. S3.

Appendix B: List of warning and error codes and messages

B.1 Warning codes and messages

The following is a list of possible warning codes from the program with a short explanation. Together with the station name acronym, these codes, the data's current date (year, month, day), and some explanatory text are written to the program log file `AirGAM_log.txt`.

Warning #1a: Insufficient data coverage for years.
This warning is issued if there are insufficient data for a station relative to the data coverage criterion `perc2` for years. It is given early as part of building the global list of stations. The station will not be added to the global list and will not be processed.

Warning #1b: Insufficient data coverage for years.
This warning is also issued if there are insufficient data for a station relative to the data coverage criterion `perc2` for years. Still, it will be given only after reading all data for the station and considering the `perc1` data coverage criterion for each year. The station will not be processed.

Warning #1c: Negative concentration detected and replaced by the value 0.1.
This warning is issued if the station data contain zero or negative concentrations for the compounds NO_2, PM_{10} and $PM_{2.5}$, which are log-transformed by the model. Such concentrations are replaced by the value 0.1.

Warning #2a: Covariate not significant.
This warning is issued from the trend estimation if a covariate gets a p-value higher than 0.05. The smooth function associated with the covariate is considered not significantly different from a flat zero function at the 5 % level.

Warning #2b: Number of basis functions perhaps too low.

This warning is issued from the trend estimation based on the check routine `k.check` in `mgcv`. Suppose, for a given covariate, the p-value from the `k.check` output table is smaller than 0.05, and the corresponding k-index is smaller than 1. In that case, a warning will be issued if the value of `k' – edf` from the table is smaller than 0.5, where `k'` is the theoretical and `edf` the empirical number of degrees of freedom for this covariate, respectively. This indicates that the number of basis functions for the covariate might be too low. Here `k'=k – 1`, where `k` is the number of basis functions defined for the covariate on input. Also, note that the output from the routines `k.check` and `gam.check` (producing the output to the *gam.check* files) are usually different since the two routines use different seed values.

Warning #2c: Covariate might be dependent.
This warning is issued from the trend estimation based on concurvity values of the `estimate` type from the `concurvity` routine in `mgcv`. It is triggered if a concurvity value of a given covariate is higher than 0.4. This is then taken to indicate that the covariate might depend on one or more of the other covariates, either linearly or nonlinearly.

Warning #3a: A covariate value used in prediction is outside the interval of values from the training.
This warning is issued from the cross-validation part. It is only triggered when the control variable is `rob_pred=limcov` or `rob_pred=outmiss`. In the former case, the covariate value is adjusted to the training set's nearest value before predicting. In the second case, the covariate value is not altered, but warning #3b is also issued with potential action, as described below. See Sect. S3.3.15 for more description of the `rob_pred` variable.

Warning #3b: This warning is issued from the cross-validation part. A covariate value used in prediction is outside the interval of covariate values from the training. In addition, the predicted concentration value is outside the whisker fences of a generalised box plot of concentration values from the training data. In this case, the predicted concentration is considered a potential outlier. The covariate value is not adjusted, but the predicted concentration is set to the missing value (NA).

B.2 Error codes and messages

The following is a list of possible error codes from the program with a short explanation. Together with the station name acronym, these codes, the data's current date (year, month, day), and some explanatory text are written to the program log file `AirGAM_log.txt`.

Error #1a: Error when reading the station list file for a given year.
Something went wrong when reading the station list file for a given year. The user should inspect this file to find the reason for the error.

Error #1b: A necessary column is not in the station list file for a given year.

The program checks if all required columns with names `name`, `lon`, `lat`, `z`, `type`, `area`, and `country` are in the station data frame as read from the station list file. If this is not the case, errors are issued stating which columns are missing.

Error #1c: Error when reading the station data file for a given year.

Something went wrong when reading the station data file for a given year. The user should inspect this file to find the reason for the error.

Error #1d: A necessary date column, concentration column, or meteorological covariate column is not in the station data file.

The program checks to see if all required columns with data are present in the station data file. If this is not the case, errors are issued stating which columns are missing.

Error #1e: No data were found for a given station.

No data were found when reading the station data for a given station. The user should inspect the data directory to find the reason.

Error #2a: Error when trying to run the `bam` routine in the `mgcv` R package.

Something went wrong when calling the `bam` routine for solving the GAM model equations as part of the trend calculations. The user should inspect the station data for the whole period of the trend estimation to find the reason for the error. The same error code may also be issued when running the `gamm` routine in the trend calculations (`incl_ar1=1`). The text message makes it clear which routine is involved.

Error #2b: Error when trying to run the `gam` routine in the `mgcv` R package.

Something went wrong when calling the `gam` routine for solving the GAM model equations as part of the trend calculations. The user should inspect the station data for the whole period of the trend estimation to find the reason for the error.

Error #2c: Error when trying to run the `bam` routine in the `mgcv` R package during cross-validation.

As part of the cross-validation, something went wrong when calling the bam routine for solving the GAM model equations. The user should inspect the station data for the whole period of the trend estimation minus the current year for cross-validation to find the reason for the error. The same error code may also be issued when running the `gamm` routine as part of the cross-validation calculations (`incl_ar1=1`). The text message makes it clear which routine is involved.

Error #2d: Error when trying to run the `gam` routine in the `mgcv` R package during cross-validation.

As part of the cross-validation, something went wrong when calling the bam routine for solving the GAM model equations. The user should inspect the station data for the whole period of the trend estimation minus the current year for cross-validation to find the reason for the error.

Appendix C: Pre-processing of the ECMWF ERA-5 meteorological data

This appendix briefly describes how some of the meteorological input data to AirGAM used in the EEA 2005-2019 study was pre-processed from the available ECMWF ERA5 data. The wind direction at 10 m is not provided in the ECMWF data, but the horisontal wind components (u and v at 10 m) are provided. We used these to compute the direction.

Also, relative humidity is not given in the ECMWF data, but absolute humidity is. We used this with the surface temperature, the surface pressure and the height of the monitoring station to calculate the relative humidity based on formulas given in Vaisala (2013). The details of these calculations are as follows. First, the saturation pressure of water vapour in the air at temperature T is calculated as:

$$P_{ws} = A \cdot 10^{mT/(T+T_n)},$$

where $A = 6.116441$, $m = 7.591386$, and $T_n = 240.7263$. The partial pressure of water vapour in g/kg is calculated as:

$$P_w = h2o \cdot P_s / (h2o + B),$$

where $h2o = q \cdot 1000$ with q the absolute humidity given in the ECMWF data, $B = 621.9907$, and P_s the estimated atmospheric pressure at the station's height H_s, $P_s = 0.01 \cdot (P_{msl} - 1.2 \cdot 9.81 \cdot H_s)$, with P_{msl} the mean sea level pressure. Then the relative humidity in percent is calculated by:

$$RH = 100 \cdot P_w / P_{ws}.$$

According to Vaisala (2013), these formulas may be used for a temperature range of [-20°C, +50°C]. On a few occasions, these calculations lead to RH higher than 100% or lower than 0%. We forced it to be 100% and 0% in these cases.

References

Bruffaerts, C., Verardi, V. and Vermandele, C.: A generalized boxplot for skewed and heavy-tailed distributions, Stat. Probab. Lett., 95, 110-117, https://doi.org/10.1016/j.spl.2014.08.016, 2014.

Carslaw, D. C.: The openair manual - open-source tools for analysing air pollution data, Manual for version 2.6-6, University of York, https://github.com/davidcarslaw/openair (last access: 25 November 2022), 2019.

Carslaw, D. C. and Ropkins K.: openair - an R package for air quality data analysis, Environ. Model. Softw., 27-28, 52–61, https://doi.org/10.1016/j.envsoft.2011.09.008, 2012.

Gneiting, T., Raftery, A. E.: Strictly proper scoring rules, prediction and estimation. J. Am. Stat. Assoc. 122 (477), 359-378, https://doi.org/10.1198/016214506000001437, 2007.

Gneiting, T., Balabdaoui, F., Raftery, A. E.: Probabilistic forecasts, calibration and sharpness. J. R. Stat. Soc. B 69 (2), 243-268, https://doi.org/10.1111/j.1467-9868.2007.00587.x, 2007.

Hastie, T. J. and Tibshirani, R. J.: Generalized Additive Models, CRC Press, Boca Raton, FL, https://doi.org/10.1201/9780203753781, 1990.

Hersbach, H.: Decomposition of the continuous ranked probability score for ensemble prediction systems. Weather and Forecasting 15, 559-570, https://doi.org/10.1175/1520-0434(2000)015%3C0559:DOTCRP%3E2.0.CO;2, 2000.

Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., Thépaut, J.-N.: ERA5 hourly data on single levels from 1979 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS), https://doi.org/10.24381/cds.adbb2d47, 2018.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J.-N.: The ERA5 global reanalysis, 146, Q. J. R. Meteorol., 146 (730), 1999–2049, https://doi.org/10.1002/qj.3803, 2020.

Marra, G. and Wood, S. N.: Practical variable selection for generalized additive models, Comput. Stat. Data Anal., 55 (7), 2372-2387, https://doi.org/10.1016/j.csda.2011.02.004, 2011.

Marra, G. and Wood, S. N.: Coverage Properties of Confidence Intervals for Generalized Additive Model Components, Scan. J. Stat., 39 (1), 53-74, https://doi.org/10.1111/j.1467-9469.2011.00760.x, 2012.

Nychka, D.: Bayesian Confidence Intervals for Smoothing Splines, J. Am. Stat. Assoc., 83 (404), 1134-1143, https://www.tandfonline.com/doi/abs/10.1080/01621459.1988.10478711, 1988.

R Core Team: R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/ (last access: 25 November 2022), 2022.

Ross, N.: GAMs in R. A free interactive course using mgcv, https://noamross.github.io/gams-in-r-course/chapter2 (last access: 25 November 2022), 2022.

Solberg, S., Walker, S.-E., Schneider, P., Guerreiro, C. and Colette, A.: Discounting the effect of meteorology on trends in surface ozone: Development of statistical tools, ETC/ACM Technical paper 15/2017, European Topic Centre on Air Pollution and Climate Change Mitigation,

https://www.eionet.europa.eu/etcs/etc-atni/products/etc-atni-reports/etcacm_tp_2017_15_discount_meteo_on_o3_trends (last access: 25 November 2022), 2018a.

Solberg, S., Walker, S.-E. and Schneider, P.: Trend in measured NO_2 and PM: Discounting the effect of meteorology, ETC/ACM Eionet Report 9/2018, European Topic Centre on Air Pollution and Climate Change Mitigation, https://www.eionet.europa.eu/etcs/etc-atni/products/etc-atni-reports/eionet_rep_etcacm_2018_9_no2_pm_trends (last access: 25 November 2022), 2018b.

Solberg, S., Walker, S.-E., Guerreiro, C. and Colette, A.: Statistical modelling for long-term trends of pollutants - Use of a GAM model for the assessment of measurements of O_3, NO_2 and PM, ETC/ATNI Report 14/2019, European Topic Centre on Air Pollution and Climate Change Mitigation, https://www.eionet.europa.eu/etcs/etc-atni/products/etc-atni-reports/etc-atni-report-14-2019-statistical-modelling-for-long-term-trends-of-pollutants-use-of-a-gam-model-for-the-assessment-of-measurements-of-o3-no2-and-pm-1 (last access: 25 November 2022), 2019.

Solberg, S., Colette, A., Raux, B., Walker, S.-E., Guerreiro, C.: Long-term trends of air pollutants at national level 2005-2019, ETC/ATNI Eionet Report 9/2021, European Topic Centre on Air Pollution and Climate Change Mitigation, https://www.eionet.europa.eu/etcs/etc-atni/products/etc-atni-reports/etc-atni-report-9-2021-long-term-trends-of-air-pollutants-at-national-level-2005-2019 (last access: 25 November 2022), 2021.

Vaisala: Humidity conversion formulas, calculation formulas for humidity, B210973EN-F, Vaisala OY, Helsinki, Finland, https://www.vaisala.com (last access: 25 November 2022), 2013.

Walker, S.-E.: AirGAM 2022r1 model (exact for results), Zenodo [code], https://doi.org/10.5281/zenodo.6334104, 2022a.

Walker, S.-E.: AirGAM 2022r1 model (latest). Zenodo [code], https://doi.org/10.5281/zenodo.6334104, 2022b.

Walker, S.-E., Solberg, S.: AirGAM 2022r1 basic data 2005-2019 and scripts, Zenodo [data set], https://doi.org/10.5281/zenodo.6334131, 2022a.

Walker, S.-E., Solberg, S.: AirGAM 2022r1 input data for all stations 2005-2019, Zenodo [data set], https://doi.org/10.5281/zenodo.6334171, 2022b.

Walker, S.-E., Solberg, S.: AirGAM 2022r1 NO_2 results for all stations 2005-2019, Zenodo [data set], https://doi.org/10.5281/zenodo.6334195, 2022c.

Walker, S.-E., Solberg, S.: AirGAM 2022r1 O_3 results for all stations 2005-2019, Zenodo [data set], https://doi.org/10.5281/zenodo.6334317, 2022d.

Walker, S.-E., Solberg, S.: AirGAM 2022r1 PM_{10} results for all stations 2005-2019, Zenodo [data set], https://doi.org/10.5281/zenodo.6334327, 2022e.

Walker, S.-E., Solberg, S.: AirGAM 2022r1 $PM_{2.5}$ results for all stations 2005-2019, Zenodo [data set], https://doi.org/10.5281/zenodo.6334334, 2022f.

Walker, S.-E., Solberg, S., Schneider, P., and Guerreiro, C.: The AirGAM 2022r1 air quality trend and prediction model, Geosci. Model Dev., 16, 573-595, https://doi.org/10.5194/gmd-16-573-2023, 2023.

Wilks, D. S.: Statistical Methods in the Atmospheric Sciences (4th ed.), Elsevier, Amsterdam,

https://doi.org/10.1016/C2017-0-03921-6, 2019.

Wood, S. N.: Generalized Additive Models. An introduction with R, Chapman and Hall/CRC Press, Boca Raton, Florida, https://doi.org/10.1201/9781315370279, 2017.

Zeileis A.: Econometric computing with HC and HAC covariance matrix estimators, J. of Stat. Software, 11(10), 1–17, https://doi.org/10.18637/jss.v011.i10, 2004.