

Contents

S1 Historical context of trend detection techniques	2
S2 Recommendations for violations of fundamental assumptions related to the ordinary least squares	3
S3 Incorporation of autocorrelation	5
S4 Illustration on the OMI/MLS tropospheric column ozone: Influence of autocorrelation on short-term trend uncertainty	7
S5 README: a short introduction of R code	12
S6 R code for Section 5	14
S7 R code for Section 6	19

List of Figures

S-1 Maps of OLS and GLS-AR1 trend estimates [DU/decade] from the OMI/MLS satellite instruments.	8
S-2 Maps of OLS and GLS-AR1 associated standard errors [DU/decade] corresponding to Figure S1.	9
S-3 AR(1) coefficients corresponding to the OMI/MLS observations in Figures S1 and S2.	10

S1 Historical context of trend detection techniques

We provide some historical context that describes the gradual development of trend detection techniques over many decades and the role of increased computing power that made more advanced techniques feasible.

Ordinary least squares (OLS) has a long history dating back to the early 19th century ([Legendre \(1805\)](#), [Gauss \(1809\)](#), see the historical review by [Stigler \(1981\)](#)). Independent of the regression models that depend on it, OLS itself is simply an algorithm designed for finding a linear combination of multiple dependent variables that has the minimal mean squared error (MSE) against one independent variable. Since OLS has an explicit and elegant solution, with limited data points (typically less than a few dozen points), this estimation can be made without any computational resources such as calculators or computers, which did not exist in the 19th and early 20th centuries. The exact assumptions to reach the full potential of OLS were completed by the Gauss-Markov theorem and an amendment by Alexander Aitken (i.e., an adjustment for correlated errors, [Aitken, 1936](#)) in the early 20th century. After that, several attempts have been made to provide an improved estimator, or at least an alternative when the OLS assumptions are violated, i.e. the errors in the linear model have a mean zero and constant variances, and are uncorrelated, according to the Gauss-Markov theorem.

Additional estimators were proposed around the mid-20 century, including the classical approaches such as Sen-Theil and Siegel's estimators. These methods were based on median values and did not rely on the normal distribution and they are known to be robust against outliers. Like OLS, these methods were relatively simple and could be solved without calculators or computers.

Meanwhile, the mainstream discipline of robust regression development focused on avoidance of the worst case scenario, i.e., certain contaminated data points can completely ruin the OLS estimator, and therefore several estimators were proposed that could tolerate aberrant outliers, such as least trimmed squares, least median of squares, or ridge regression. These methods are made by either a variant of OLS or by requiring much heavier computational resources than OLS; in general these estimators cannot be implemented without a computer.

As increased computing power became available in the late 20th century, many complex and powerful methods were developed accordingly. For example, 1) the rather limited classical methods could be replaced by the more generalizable quantile regression; 2) the worst case scenario of aberrant outliers could be ruled out by database quality control methods; and 3) traditional multiple linear regression required pre-specification of the functional form of the trends to be detected (either linear or nonlinear), but with modern computational power, the novel algorithm can automatically identify any form of nonlinearity.

While the earlier, less-computationally intensive methods are still valid if the basic assumptions of the data distributions are met, and while we describe these methods in the main text, this review focuses on the advantages of the more complex modern methods, made possible by today's advanced computing resources.

S2 Recommendations for violations of fundamental assumptions related to the ordinary least squares

Here we recommend alternative approaches that might provide a better solution or are less influenced by a violation of OLS assumptions, while clearly describing the assumptions associated with these methods.

- *Normality*: The error term from a fitted model is assumed to be normally distributed (instead of the time series itself), which implies that unexplained variation is an unstructured zero-centered random noise. Violation of the normality alone does not adversely affect the trend estimation (minimization of OLS criterion does not involve any normal assumption), rather it might represent that the proposed regression model is not able to capture entirely the systematic pattern of the data, or some important components (but not necessarily directly related to the trend) are missing in the model. A normality is assumed in order to find the confidence interval for the regression coefficients (e.g. slope). When normality is inappropriate (e.g., outliers would produce deviations from normality), median based approaches (including Sen-Theil estimator, Siegel's repeated medians, quantile regression and least median of squares) or LTS can offer a robust estimator.

In addition, for a trend study of human exposure to air pollution, rather than investigating the mean or median change, it is generally interesting to see the exposure at different quantiles (e.g., [Porter et al. \(2015\)](#); [Sousa et al. \(2009\)](#)). Extreme weather conditions, such as precipitation or temperature, are also of interest in many atmospheric studies (e.g., [Friederichs and Hense \(2007\)](#); [Haugen et al. \(2019\)](#)). In those cases the classical nonparametric trend methods are no longer appropriate, because they focus on a single quantile (median); alternatively the quantile regression can be used to study multiple quantiles at the same time ([Koenker and Hallock, 2001](#)). An autocorrelation structure can also be incorporated into the quantile regression ([Koenker and Xiao, 2006](#)).

- *The errors are uncorrelated*: See next section for more details.
- *Homoscedasticity or constant error variance*: The error term from a fitted model is assumed to have an homogeneous variance (i.e., a finite constant). With a focus placed on atmospheric chemistry, this assumption implies that the variance (e.g. the uncertainty due to aggregation of sparsely sampled observations) is about the same across all seasons and years. Similar to the autocorrelation, the heteroscedasticity does not adversely affect the estimation of the trend value, but it can produce an inappropriate inference about its uncertainty (hence also the result of hypothesis testing and p-value), since all the sub-scale variabilities were inappropriately represented by a universal constant variance. The conventional recommendation to reduce heteroscedasticity is to apply the weighted least squares. The idea is that, at each temporal point, the data with larger variability receives less weight in the estimation. Ideally, we like to specify the weight as being proportional to the inverse of the variance of the data. However, the exact variability at each point is typically unknown and largely dependent on the sampling frequency. Common approaches are 1) to fit a homoscedastic model and then use the residuals to approximate the associated variances as weights in a refit of the model; and 2) to use the uncertainty of the seasonal cycle to account for the seasonal heteroskedasticity ([Zuur et al., 2009](#)).

Analogy to the autoregressive process, ARCH (autoregressive conditional heteroscedasticity, [Engle, 1982](#)) or its extension the generalized ARCH model ([Bollerslev, 1986](#)), is also a promising approach to account for both autocorrelation and heteroscedasticity. The ARCH model is well known for its ability in modeling time-varying volatility and volatility clustering (i.e., large variabilities are likely to be followed by large variabilities) in financial time series ([Brooks, 2019](#)). Thus, a better characterization of heterogeneous variability in atmospheric sciences is expected to be achievable with such an approach.

- *Linearity*: Here we focus on the non-linearity of the trend, rather than the nonlinear regression model, and assume that a nonlinear trend can be approximated by a linear combination of piecewise linear, orthogonal polynomial or a valid set of basis functions. The remedies for nonlinearity can be found in Section 3 of the main text.
- *No or weak multicollinearity*: Multicollinearity indicates the dependence between covariates, and is a numerical problem, not directly associated with statistical concepts. In the case when perfect multicollinearity is found in the multivariate regression model (one covariate can be fully represented by other covariates), the unique solution of OLS does not exist. For example, meteorological variables are critical and need to be adjusted in the multiple regression model for studying the short term surface ozone trend, but these variables might be closely related (i.e. highly correlated) with each other, and should be selected carefully to mitigate the multicollinearity ([Paschalidou et al., 2009](#); [Otero et al., 2016](#)). A less severe multicollinearity could make the estimation of the trend and its uncertainty inaccurate. A statistical regularization, e.g., ridge regression or Lasso, can be used to alleviate mild or moderate multicollinearity, since it introduces additional information to constrain the regression coefficients within a reasonable range and mitigate the ill posed numerical problems ([Tikhonov et al., 2013](#)).

S3 Incorporation of autocorrelation

Autocorrelation is an inevitable factor that needs to be addressed in the trend analysis of atmospheric time series (Box et al., 2015; Tiao et al., 1990; Weatherhead et al., 1998). If autocorrelation is ignored by the trend estimation of an autocorrelated time series, the resulting trend uncertainty is expected to be systematically underestimated.

Conventionally the method that takes autocorrelation into account is called generalized least squares (GLS), and is based on the OLS. The statistical assumption for the error term is that N_t are expected to have a mean zero and be uncorrelated, i.e. $\text{Cov}(N_t, N_{t'}) = 0$, if $t \neq t'$, however, if the latter equality is not held, then it indicates that autocorrelation is present. If the functional form of the autocorrelation can be approximated by an autoregressive process of order 1 (known as AR(1)), then according to the definition, i.e., $N_t = \phi N_{t-1} + \varepsilon_t$, where ε_t is assumed to be white noise. A full scale description of the variance-covariance matrix of the error structure is as follows:

$$\begin{aligned} \Omega &= \sigma_N^2 \begin{pmatrix} 1 & \text{Corr}(N_1, N_2) & \text{Corr}(N_1, N_3) & \cdots & \text{Corr}(N_1, N_T) \\ \text{Corr}(N_1, N_2) & 1 & \text{Corr}(N_2, N_3) & \cdots & \text{Corr}(N_2, N_T) \\ \text{Corr}(N_1, N_3) & \text{Corr}(N_2, N_3) & 1 & \cdots & \text{Corr}(N_3, N_T) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \text{Corr}(N_1, N_T) & \text{Corr}(N_2, N_T) & \text{Corr}(N_3, N_T) & \cdots & 1 \end{pmatrix} \\ &= \frac{\sigma_\varepsilon^2}{1 - \phi^2} \begin{pmatrix} 1 & \phi & \phi^2 & \cdots & \phi^{T-1} \\ \phi & 1 & \phi & \cdots & \phi^{T-2} \\ \phi^2 & \phi & 1 & \cdots & \phi^{T-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi^{T-1} & \phi^{T-2} & \phi^{T-3} & \cdots & 1 \end{pmatrix}, \end{aligned} \quad (1)$$

where σ_N^2 and σ_ε^2 are constant variances for N_t and ε_t , respectively, and $\{\text{Corr}(N_t, N_{t'}), t \neq t'\}$ are the autocorrelations. So the AR(1) structure for a correlation matrix implies that the correlation decays with the increase of temporal lag, i.e., $\text{Corr}(N_t, N_{t+h}) = \phi^{|h|}$, and the correlation structure is controlled by a single parameter ϕ . It should be noted that this formulation is called a stationary time series, because the correlations in the residuals are only dependent on the length of the temporal lag (so the correlations from various autoregressive-moving-average processes are only determined by the time lag).

To incorporate the variance-covariance matrix into the optimization algorithm, we can modify the loss function for the OLS as:

$$\sum_{t=1}^T (L^{-\top} y_t - L^{-\top} \mathbf{X}_t \beta)^2, \quad \text{where } L^\top L = \Omega. \quad (2)$$

To solve this problem, we start with an initial estimate for the autocorrelation coefficient ϕ , achieved through the following iterative procedure: the initial condition is set to $\phi = 0$ (i.e., equivalent to OLS), then from the residuals of the fitted model the AR(1) coefficient is calculated by

$$\phi = \frac{\sum_{t=2}^T (N_t - \bar{N})(N_{t-1} - \bar{N})}{\sum_{t=1}^T (N_t - \bar{N})^2}. \quad (3)$$

The next iteration takes the previous coefficient into Ω , and updates the estimation in Equations (2) and (3). This loop proceeds until the coefficient has converged within a given tolerance level. Even though the above demonstration is only made for an AR(1) process, the same iteration process can be applied to a more complex variance-covariance structure, as long as the matrix Ω is well defined. Nevertheless, in most cases the

autocorrelation of atmospheric composition time series can be sufficiently well described by an AR(1) process ([Weatherhead et al., 1998](#)).

Theoretically, the autocorrelation structure can be incorporated by the same iterative method into the estimators with similar loss functions, such as LTS, LMS, Lasso, ridge regression and QR, as long as the loss function $f(y_t, \mathbf{X}_t\boldsymbol{\beta})$ can be replaced by $f(L^{-\top}y_t, L^{-\top}\mathbf{X}_t\boldsymbol{\beta})$, where $L^\top L = \boldsymbol{\Omega}$, and this application is beyond the conventional scope of GLS. Therefore, we do not claim any particular algorithm to be the best trend technique, because it involves different considerations to tackle the numerical issues, but the appropriateness of a trend model should be evaluated through the model specification.

S4 Illustration on the OMI/MLS tropospheric column ozone: Influence of autocorrelation on short-term trend uncertainty

Here we demonstrate the impact of autocorrelation on short-term trend detection. Rather than limiting this analysis to just one or a handful of time series, which may result in an incomplete or biased view of the impact of autocorrelation, our demonstration uses 1,728 time series based on monthly tropospheric column ozone values detected by the OMI/MLS satellite instruments (Ziemke et al., 2006, 2011, 2019). The OMI/MLS record spans October 2004 through December 2019 (183 monthly values) and is available globally at $5^\circ \times 5^\circ$ resolution from 60°S to 60°N . Figure S-1 shows the $5^\circ \times 5^\circ$ cell-wise ozone trends estimated by the OLS and GLS-AR1 methods. The OLS method does not account for autocorrelation, while the GLS-AR1 method does account for autocorrelation. Gray crosses indicate that the 95% confidence interval for the trend value at that grid cell excludes zero. These two methods produce a similar pattern of trends, in terms of magnitude and distribution, with the greatest discrepancies found above the tropical Pacific Ocean. From a statistical point of view, the OLS estimate of the trend value remains unbiased (i.e. the estimate is not systematically different from the truth) in the presence of autocorrelation, but the autocorrelation does affect the trend estimate in the GLS method through the covariance structure.

The corresponding standard errors of the fitted trends from OLS and GLS-AR1 are shown respectively in Figures S-2(a) and (b). The uncertainty pattern from the OLS method has a lack of features compared to the GLS-AR1 method. These figures clearly highlight the fact that standard errors associated with the trend estimates tend to be underestimated (and the test statistics overestimated) when the positive autocorrelation of the residuals is ignored (Hurlbert, 1984; Bence, 1995). For this particular problem the rejection rate of the hypothesis test is 86.7% for OLS and 78.4% for GLS-AR1 (i.e. 1499 v 1354 out of 1728 cells, with an overall 8.3% difference); the false rejections are primarily found above northern Australia and the tropical Pacific Ocean. All the discrepancies present a rejection of the null hypothesis at the 95% confidence level by the OLS, but this evidence is not supported by the GLS-AR1. These regional scale differences are sufficiently critical to demonstrate why it is inappropriate to ignore autocorrelation.

The distribution of the fitted AR(1) coefficients is shown across all grid cells in Figure S-3. This example clearly demonstrates that the autocorrelation is not just a random behavior; it does reveal certain spatial patterns. Rejection of the null hypothesis in the Durbin-Watson test at the 95% confidence level (i.e., an indication that the AR(1) coefficient is different from zero) occurred in 78% of grid cells, a very strong indication that ignoring autocorrelation results in systematic bias in the uncertainty estimates. Overall, the influence of autocorrelation on the magnitude of the trend estimate is generally weak, but it does have an impact on the uncertainty associated with the trend. In this case study, we found a false rejection rate of 8.3% due to the non-independence of time series data.

References

- Aitken, A. C. (1936). On least squares and linear combination of observations. *P. Roy. Soc. Edinb.*, 55:42–48.
- Bence, J. R. (1995). Analysis of short time series: correcting for autocorrelation. *Ecology*, 76(2):628–639.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *J. Econ.*, 31(3):307–327.

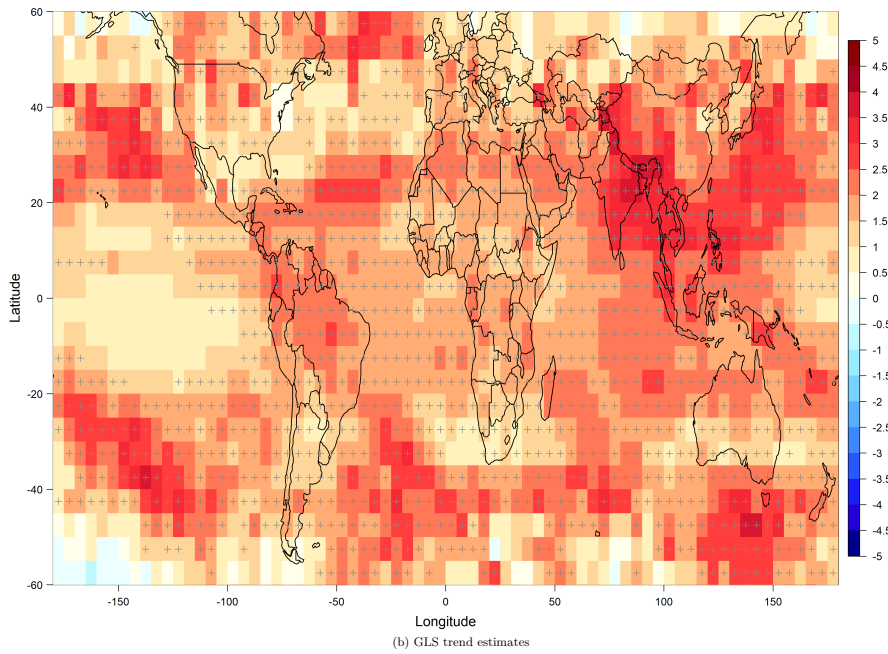
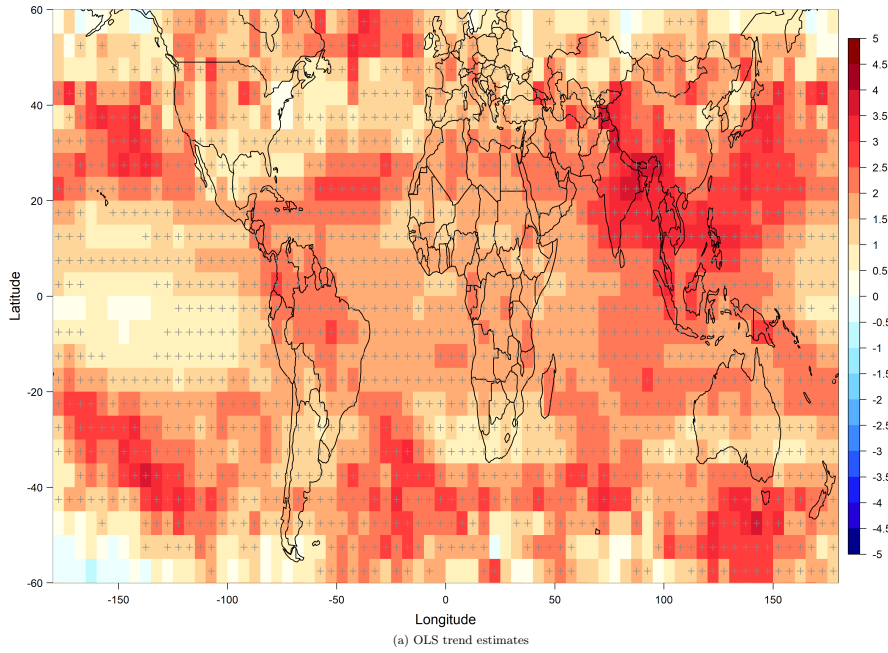


Figure S-1: Maps of OLS and GLS-AR1 trend estimates [DU/decade] from the OMI/MLS satellite instruments.

Estimates are based on the period 2004-2019, and 5×5 degree resolution. Gray crosses indicate grid cells with trends distinguished from zero at the 95% confidence level.

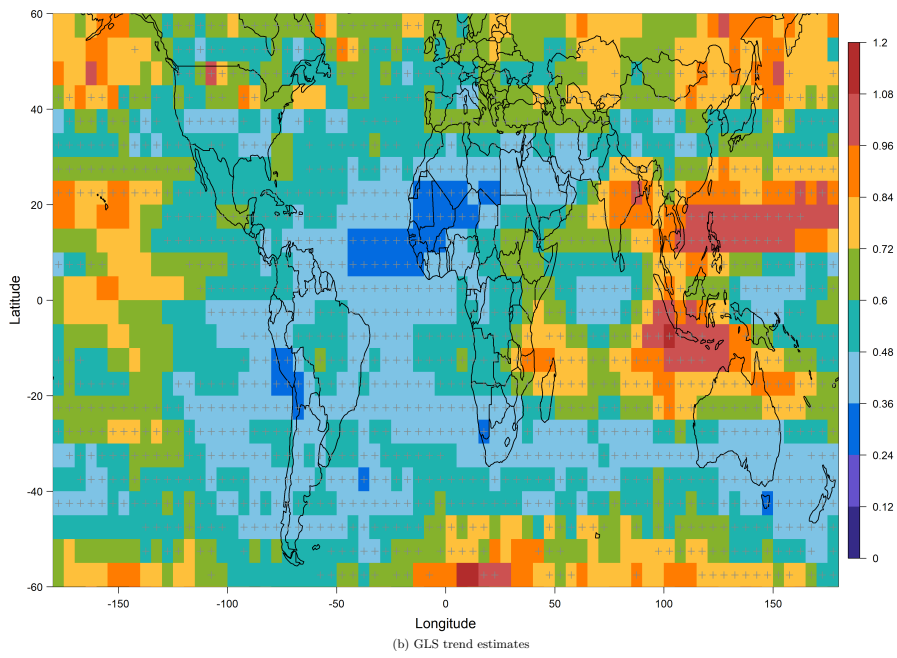
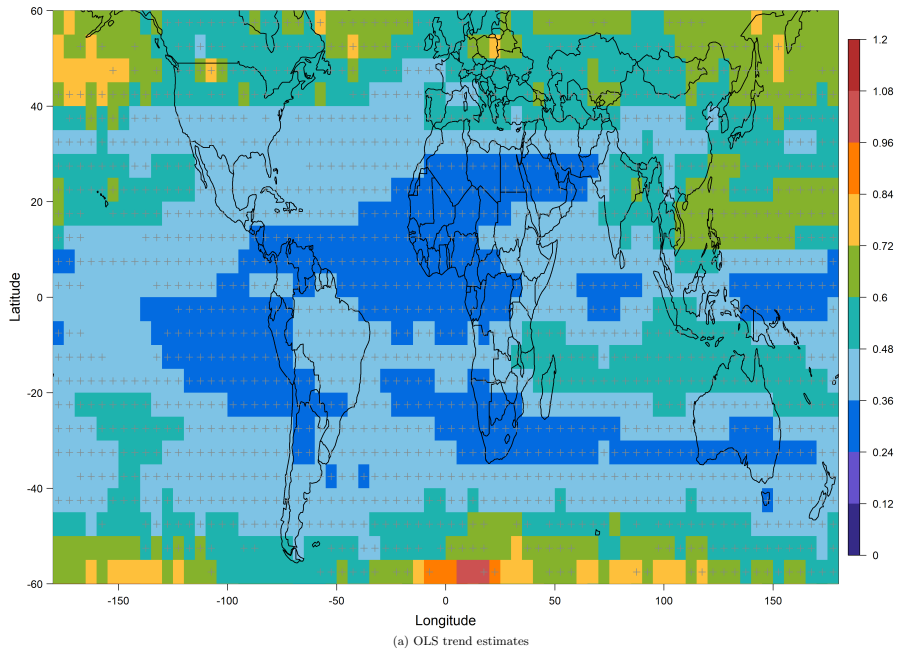


Figure S-2: Maps of OLS and GLS-AR1 associated standard errors [DU/decade] corresponding to Figure S1.

Gray crosses indicate grid cells with trends distinguished from zero at the 95% confidence level.

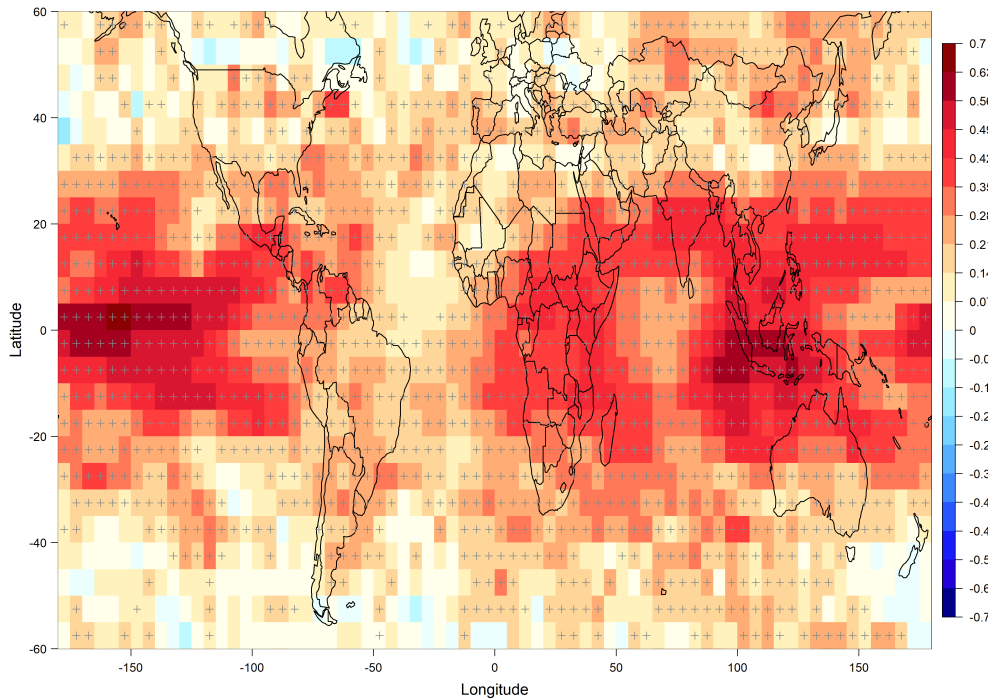


Figure S-3: AR(1) coefficients corresponding to the OMI/MLS observations in Figures S1 and S2.

Gray crosses indicate grid cells with rejections of the Durbin-Watson test at the 95% confidence level.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control (5th ed.)*. John Wiley & Sons.

Brooks, C. (2019). *STATA Guide for Introductory Econometrics for Finance*. Cambridge University Press.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, pages 987–1007.

Friederichs, P. and Hense, A. (2007). Statistical downscaling of extreme precipitation events using censored quantile regression. *Mon. Weather Rev.*, 135(6):2365–2378.

Gauss, K. F. (1809). *Theoria Motus Corporum Coelestium [Theory of the motion of the heavenly bodies moving about the sun in conic section]*. Perthes, Hamburg.

Haugen, M. A., Stein, M. L., Sriver, R. L., and Moyer, E. J. (2019). Future climate emulations using quantile regressions on large ensembles. *Adv. Stat. Clim. Meteorol. Oceanogr.*, 5(1):37–55.

Hurlbert, S. H. (1984). Pseudoreplication and the design of ecological field experiments. *Ecol. Monogr.*, 54(2):187–211.

Koenker, R. and Hallock, K. F. (2001). Quantile regression. *J. Econ. Perspect.*, 15(4):143–156.

Koenker, R. and Xiao, Z. (2006). Quantile autoregression. *J. Am. Stat. Assoc.*, 101(475):980–990.

Legendre, A. M. (1805). *Nouvelles méthodes pour la détermination des orbites des comètes [New methods for determining the orbits of comets]*. F. Didot.

Otero, N., Sillmann, J., Schnell, J. L., Rust, H. W., and Butler, T. (2016). Synoptic and meteorological drivers of extreme ozone concentrations over europe. *Environ. Res. Lett.*, 11(2):024005.

Paschalidou, A. K., Kassomenos, P. A., and Bartzokas, A. (2009). A comparative study on various statistical techniques predicting ozone concentrations: implications to environmental management. *Environ. Monit. Assess.*, 148(1):277–289.

- Porter, W. C., Heald, C. L., Cooley, D., and Russell, B. (2015). Investigating the observed sensitivities of air-quality extremes to meteorological drivers via quantile regression. *Atmos. Chem. Phys.*, 15(18):10349–10366.
- Sousa, S., Pires, J., Martins, F., Pereira, M., and Alvim-Ferraz, M. (2009). Potentialities of quantile regression to predict ozone concentrations. *Environmetrics*, 20(2):147–158.
- Stigler, S. M. (1981). Gauss and the invention of least squares. *Ann. Stat.*, pages 465–474.
- Tiao, G. C., Reinsel, G. C., Xu, D., Pedrick, J. H., Zhu, X., Miller, A. J., DeLuisi, J. J., Mateer, C. L., and Wuebbles, D. J. (1990). Effects of autocorrelation and temporal sampling schemes on estimates of trend and spatial correlation. *J. Geophys. Res. Atmos.*, 95(D12):20507–20517.
- Tikhonov, A. N., Goncharsky, A. V., Stepanov, V. V., and Yagola, A. G. (2013). *Numerical methods for the solution of ill-posed problems*, volume 328. Springer.
- Weatherhead, E. C., Reinsel, G. C., Tiao, G. C., Meng, X.-L., Choi, D., Cheang, W.-K., Keller, T., DeLuisi, J., Wuebbles, D. J., Kerr, J. B., Miller, A. J., Oltmans, S. J., and Frederick, J. E. (1998). Factors affecting the detection of trends: Statistical considerations and applications to environmental data. *J. Geophys. Res. Atmos.*, 103(D14):17149–17161.
- Ziemke, J. R., Chandra, S., Duncan, B. N., Froidevaux, L., Bhartia, P. K., Levelt, P. F., and Waters, J. W. (2006). Tropospheric ozone determined from aura omi and mls: Evaluation of measurements and comparison with the global modeling initiative’s chemical transport model. *J. Geophys. Res. Atmos.*, 111(D19).
- Ziemke, J. R., Chandra, S., Labow, G. J., Bhartia, P. K., Froidevaux, L., and Witte, J. C. (2011). A global climatology of tropospheric and stratospheric ozone derived from aura omi and mls measurements. *Atmos. Chem. Phys.*, 11(17):9237–9251.
- Ziemke, J. R., Oman, L. D., Strode, S. A., Douglass, A. R., Olsen, M. A., McPeters, R. D., Bhartia, P. K., Froidevaux, L., Labow, G. J., Witte, J. C., Thompson, A. M., Haffner, D. P., Kramarova, N. A., Frith, S. M., Huang, L.-K., Jaross, G. R., Seftor, C. J., Deland, M. T., and Taylor, S. L. (2019). Trends in global tropospheric ozone inferred from a composite record of toms/omi/mls/omps satellite measurements and the merra-2 gmi simulation. *Atmos. Chem. Phys.*, 19(5):3257–3269.
- Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., Smith, G. M., et al. (2009). *Mixed effects models and extensions in ecology with R*, volume 574. Springer.

S5 README: a short introduction of R code

A very short demonstration for using each trend detection technique is provided as follows. Given y is a vector of deseasonalized anomaly series and x is corresponding temporal index, then each of the technique can be specified by:

- *Ordinary least squares (OLS):*

```
mod.lm=lm(y~x)
```

- *Generalized least squares (GLS):*

```
library(nlme)
mod.ar=glS(y~x, correlation=corAR1(form=~x)) #AR1 model
mod.ar=glS(y~x, correlation=corARMA(form=~x, p=2)) #AR2 model
```

- *Quantile regression:*

```
library(quantreg)
mod.rq=rq(y~x, tau=0.5) #median regression or least absolute deviations (LAD)
mod.rq=rq(y~x, tau=seq(0.05, 0.95, by=0.05)) #multiple quantiles can be specified
```

- *Lasso and ridge regression:*

```
library(glmnet)
mod.l1=cv.glmnet(cbind(1,x),y,alpha=1) #Lasso
mod.l1=glmnet(cbind(1,x), y, alpha = 1, lambda = mod.l1$lambda.min)
mod.l2=cv.glmnet(cbind(1,x),y,alpha=0) #ridge regression
mod.l2=glmnet(cbind(1,x), y, alpha = 0, lambda = mod.l2$lambda.min)
```

- *Least trimmed squares:*

```
library(robustbase)
mod.lts=ltsReg(y~x, alpha=0.9)
```

- *Least median of squares:*

```
library(MASS)
mod.lms=lmsreg(y~x)
```

- *Sen-Theil and Siegel's estimators:*

```
library(mblm)
mod.m1=mblm(y~x, repeated=F) #Sen-Theil estimator
mod.m2=mblm(y~x, repeated=T) #Siegel's estimator
```

If data is not deseasonalized (i.e., with a corresponding monthly index m) and a change point is expected at a point tc , then the expression can be rewritten as (Equation (3) of the main text):

```
xc=pmax(x-tc, 0)
mod.ar=glS(y~x+xc+
  cos(m*pi/12)+sin(m*pi/12)+cos(m*pi/6)+sin(m*pi/6),
  correlation=corAR1(form=~x))
```

For the specification of GAMs, a nonlinear trend and fixed seasonal cycle can be expressed as:

```
library(mgcv)
library(qgam)
mod.gamm=gamm(y~ s(x, bs="cr", k=15) + s(m, bs="cc", k=6),
  correlation=corAR1(form=~x))
```

Given a data frame dt with multiple columns (e.g., y =data, x =temporal index, lon=longitude, lat=latitude, id=unique number for each station), then the model for multi-site data can be specified through:

```

#mean estimator
mod.gam=gam(y~ s(x, bs="cr", k=15) +
  s(lon, lat, bs="gp", k=60) +
  ti(x,id, bs=c("cr","fs"),d=c(1,1), k=c(15,10)), data=dt)

#median estimator
mod.qgam=qgam(y~ s(x, bs="cr", k=15) +
  s(lon, lat, bs="gp", k=60) +
  ti(x,id, bs=c("cr","fs"),d=c(1,1), k=c(15,10)), data=dt, qu=0.5)

#with AR1 included
mod.gamm=gamm(y~ s(x, bs="cr", k=15) +
  s(lon, lat, bs="gp", k=60) +
  ti(x,id, bs=c("cr","fs"),d=c(1,1), k=c(15,10)),
  correlation=corAR1(form=~x), data=dt)

```

This is only a few of many possible approaches, and the arguments (e.g. basis type bs and dimension k) are only specified for an illustrative purpose and should be tailored to specific applications. Details should be referred to the manual of R package `mgcv` or Simon N. Wood's book "Generalized Additive Models: an introduction with R".

S6 R code for Section 5

```
#####  
#This code produces Figures 7 and 8 in the paper, using various trend detection  
#techniques for the analysis of a single time series.  
#####  
#setwd("C:/Users/Usur/Dropbox/2.CurrentProjects/TrendDetection/remotedO3")  
  
#####  
#load packages  
#####  
  
library(mblm)  
library(quantreg)  
library(nlme)  
library(trend)  
library(lmtest)  
library(robustbase)  
library(tseries)  
library(glmnet)  
library(mgcv)  
library(MASS)  
col=c(rgb(0.2081, 0.1663, 0.5292),  
rgb(0.3961, 0.3176, 0.8000),  
rgb(0.0123, 0.4213, 0.8802),  
rgb(0.4941, 0.7647, 0.8980),  
rgb(0.1157, 0.7022, 0.6843),  
rgb(0.5216, 0.6980, 0.1725),  
rgb(0.9968, 0.7513, 0.2325),  
rgb(1.0000, 0.4863, 0.0000),  
rgb(0.8000, 0.3176, 0.3176),  
rgb(0.6980, 0.1725, 0.1725))  
  
#####  
#read data  
#Remote surface ozone data sets can be downloaded  
#at doi:10.34730/e792cad833174ebcafd9f052711e5660.  
#Reference: Cooper et al. "Multi-decadal surface ozone trends at globally  
#distributed remote locations." Elementa: Science of the Anthropocene 8 (2020).  
#####  
  
mm=read.table("remote_sites_monthly_means.txt",header=F, sep=" ",skip=40)  
mm[mm>=998 & mm<=1000]=NA  
colnames(mm)=c("ind","year","month","S1","S2","S3","S4","S5","S6",  
"S7","S8","S9","S10",  
"S11","S12","S13","S14","S15","S16","S17","S18","S19","S20",  
"S21","S22","S23","S24","S25","S26","S27","S28","S29","S30",  
"S31","S32","S33")  
  
suo3=read.table("remote_sites_monthly_anomalies.txt",header=F, sep=" ",skip=40)  
suo3[suo3>=998 & suo3<=1000]=NA  
dt=suo3  
colnames(dt)=c("ind","year","month","S1","S2","S3","S4","S5","S6",  
"S7","S8","S9","S10",  
"S11","S12","S13","S14","S15","S16","S17","S18","S19","S20",  
"S21","S22","S23","S24","S25","S26","S27","S28","S29","S30",  
"S31","S32","S33")  
  
#####  
#Figure 7  
#####  
i=8  
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]  
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]  
yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]  
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month))[,3])  
vt=as.data.frame(cbind(x,y,mo))  
v2=aggregate(y~mo,vt,FUN=function(x) {sd(x,na.rm=T)})  
vt=merge(vt,v2,by="mo",all=T)  
vt=vt[order(vt$x),]  
v2=vt[,4]  
mod.lm=lm(y~x)  
mod.ar=gls(y~x, correlation=corAR1(form=~x))  
mod.war=gls(y~x, correlation=corAR1(form=~x), weights=varIdent(form=~1|mo), glsControl(maxIter=90))  
mod.war=lm(y~x, weights=1/v2^2)  
mod.lts=ltsReg(y~x, alpha=0.9)  
mod.rq=rq(y~x, tau=c(0.05, 0.5, 0.95))  
mod.m1=mblm(y~x, repeated=F)  
mod.m2=mblm(y~x, repeated=T)  
mod.l1=cv.glmnet(cbind(1,x),y,alpha=1)  
mod.l1=glmnet(cbind(1,x), y, alpha = 1, lambda = mod.l1$lambda.min)  
mod.l2=cv.glmnet(cbind(1,x),y,alpha=0)  
mod.l2=glmnet(cbind(1,x), y, alpha = 0, lambda = mod.l2$lambda.min)  
mod.lms=lmsreg(y~x)  
mod.loess33=loess(y~x, span=10/(max(yr)-min(yr)))  
pd=predict(mod.loess33, se=T)  
pd1=pd$fit - qt(0.975,pd$df)*pd$se  
pd2=pd$fit + qt(0.975,pd$df)*pd$se  
  
png("trend_fit_remote.png", height=10, width=10, units="in", res=500)  
par(mar=c(0, 4, 0, 1), mgp=c(2.4, 0.8, 0), mfrow=c(3,1), oma=c(3,2,0,0), las=1)  
plot(dt$ind,dt[,i+3], xlab="", ylab="Ozone_[ppb]", cex.lab=2, type="b", xaxt="n", yaxt="n", lwd=1.5, xlim=c(379,732))  
axis(2,lwd.ticks=2, cex.axis=1.5)  
axis(1,at=setdiff(seq(min(x),max(x), by=1),x),labels=FALSE,col=2, lwd.ticks=2)
```

```

#axis(1,at=seq(37,732, by=60),labels=seq(1960,2018,by=5), lwd.ticks=2, cex.axis=1.5)
polygon(c(x,rev(x)),c(pd2,rev(pd1)),col=rgb(0.,0.,0.5),lty=2)
lines(x, predict(mod.loess33), lwd=2.5, col='gray')
xp=seq(min(x),max(x),by=0.1)
lines(xp, predict(mod.lm, newdata=data.frame(x=xp)), lwd=4.4, col=col[1])
lines(xp, predict(mod.ar, newdata=data.frame(x=xp)), lwd=4.3, col=col[2])
lines(xp, predict(mod.war, newdata=data.frame(x=xp)), lwd=4.2, col=col[3])
segments(x0=min(x), y0=summary(mod.lts)$coefficients[1,1] + summary(mod.lts)$coefficients[2,1]*min(x), x1=max(x),
y1=summary(mod.lts)$coefficients[1,1] + summary(mod.lts)$coefficients[2,1]*max(x), lwd=3.1, col=col[4])
lines(xp, predict(mod.l1, newx=cbind(1,x=xp), type="response"), lwd=3.0, col=col[5])
lines(xp, predict(mod.l2, newx=cbind(1,x=xp), type="response"), lwd=2.9, col=col[6])
lines(xp, predict(mod.m1, newdata=data.frame(x=xp)), lwd=2.8, col=col[7])
lines(xp, predict(mod.m2, newdata=data.frame(x=xp)), lwd=2.7, col=col[8])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[2], lwd=2.6, col=col[9])
lines(xp, predict(mod.lms, newdata=data.frame(x=xp)), lwd=2.5, col=col[10])
#legend(330,max(y)+1, c("OLS","GLS-AR1", "WLS", "LTS", "Lasso"), lwd=2.,
# text.col=col[c(1,2,3,4,5)], col=col[c(1,2,3,4,5)], lty=0, cex=2., bty='n')
#legend(640,-5, c("Ridge", "Sen-Theil", "Siegel", "QR-50th", "LMS"), lwd=2.,
#text.col=col[c(6,7,8,9,10)], col=col[c(6,7,8,9,10)], lty=0, cex=2., bty='n')
legend(600,-7, "Mace_Head", lwd=2., lty=0, cex=2.5, bty='n')

i=16
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]
yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month)))[,3])
vt=as.data.frame(cbind(x,y,mo))
v2=aggregate(y~mo,vt,FUN=function(x) {sd(x,na.rm=T)})
vt=merge(vt,v2,by="mo",all=T)
vt=vt[order(vt$x),]
v2=v2[,4]
mod.lm=lm(y~x)
mod.ar=gls(y~x, correlation=corAR1(form=-x))
mod.war=gls(y~x, correlation=corAR1(form=-x), weights=varIdent(form=-1|mo), glsControl(maxIter=90))
mod.war=lm(y~x, weights=1/v2^2)
mod.lts=ltsReg(y~x, alpha=0.9)
mod.rq=rq(y~x, tau=c(0.05, 0.5, 0.95))
mod.m1=mblm(y~x, repeated=F)
mod.m2=mblm(y~x, repeated=T)
mod.l1=cv.glmnet(cbind(1,x),y,alpha=1)
mod.l1=glmnet(cbind(1,x), y, alpha = 1, lambda = mod.l1$lambda.min)
mod.l2=cv.glmnet(cbind(1,x),y,alpha=0)
mod.l2=glmnet(cbind(1,x), y, alpha = 0, lambda = mod.l2$lambda.min)
mod.lms=lmsreg(y~x)
mod.loess33=loess(y~x, span=10/(max(yr)-min(yr)))
pd=predict(mod.loess33, se=T)
pd1=pd$fit - qt(0.975,pd$df)*pd$sse
pd2=pd$fit + qt(0.975,pd$df)*pd$sse

plot(dt$ind,dt[,i+3], xlab="", ylab="Ozone_[ppb]", cex.lab=2, type='b', xaxt="n", yaxt="n", lwd=1.5, xlim=c(379,732))
axis(2,lwd.ticks=2, cex.axis=1.5)
axis(1,at=setdiff(seq(min(x),max(x), by=1),x),labels=FALSE,col=2, lwd.ticks=2)
#axis(1,at=seq(37,732, by=60),labels=seq(1960,2018,by=5), lwd.ticks=2, cex.axis=1.5)
polygon(c(x,rev(x)),c(pd2,rev(pd1)),col=rgb(0.,0.,0.5),lty=2)
lines(x, predict(mod.loess33), lwd=2.5, col='gray')
xp=seq(min(x),max(x),by=0.1)
lines(xp, predict(mod.lm, newdata=data.frame(x=xp)), lwd=4.4, col=col[1])
lines(xp, predict(mod.ar, newdata=data.frame(x=xp)), lwd=4.3, col=col[2])
lines(xp, predict(mod.war, newdata=data.frame(x=xp)), lwd=4.2, col=col[3])
segments(x0=min(x), y0=summary(mod.lts)$coefficients[1,1] + summary(mod.lts)$coefficients[2,1]*min(x), x1=max(x),
y1=summary(mod.lts)$coefficients[1,1] + summary(mod.lts)$coefficients[2,1]*max(x), lwd=3.1, col=col[4])
lines(xp, predict(mod.l1, newx=cbind(1,x=xp), type="response"), lwd=3.0, col=col[5])
lines(xp, predict(mod.l2, newx=cbind(1,x=xp), type="response"), lwd=2.9, col=col[6])
lines(xp, predict(mod.m1, newdata=data.frame(x=xp)), lwd=2.8, col=col[7])
lines(xp, predict(mod.m2, newdata=data.frame(x=xp)), lwd=2.7, col=col[8])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[2], lwd=2.6, col=col[9])
lines(xp, predict(mod.lms, newdata=data.frame(x=xp)), lwd=2.5, col=col[10])
legend(355,max(y)+1, c("OLS","GLS-AR1", "WLS", "LTS", "Lasso","Ridge", "Sen-Theil", "Siegel", "QR-50th", "LMS"), lwd
=2.,
text.col=col[1:10], col=col[1:10], lty=0, cex=2.3, bty='n')
legend(600,-3, "Mt._Waliguan", lwd=2., lty=0, cex=2.5, bty='n')

i=30
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]
yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month)))[,3])
vt=as.data.frame(cbind(x,y,mo))
v2=aggregate(y~mo,vt,FUN=function(x) {sd(x,na.rm=T)})
vt=merge(vt,v2,by="mo",all=T)
vt=vt[order(vt$x),]
v2=v2[,4]
mod.lm=lm(y~x)
mod.ar=gls(y~x, correlation=corAR1(form=-x))
mod.war=gls(y~x, correlation=corAR1(form=-x), weights=varIdent(form=-1|mo), glsControl(maxIter=90))
mod.war=lm(y~x, weights=1/v2^2)
mod.lts=ltsReg(y~x, alpha=0.9)
mod.rq=rq(y~x, tau=c(0.05, 0.5, 0.95))
mod.m1=mblm(y~x, repeated=F)
mod.m2=mblm(y~x, repeated=T)
mod.l1=cv.glmnet(cbind(1,x),y,alpha=1)
mod.l1=glmnet(cbind(1,x), y, alpha = 1, lambda = mod.l1$lambda.min)
mod.l2=cv.glmnet(cbind(1,x),y,alpha=0)
mod.l2=glmnet(cbind(1,x), y, alpha = 0, lambda = mod.l2$lambda.min)
mod.lms=lmsreg(y~x)

```

```

mod.loess33=loess(y~x, span=10/(max(yr)-min(yr)))
pd=predict(mod.loess33, se=T)
pd1=pd$fit - qt(0.975,pd$df)*pd$se
pd2=pd$fit + qt(0.975,pd$df)*pd$se

plot(dt$ind,dt[,i+3], xlab="Year", ylab="Ozone_[ppb]", cex.lab=2, type='b', xaxt="n", yaxt="n", lwd=1.5, xlim=c(379,732))
axis(2,lwd.ticks=2, cex.axis=1.5)
axis(1,at=setdiff(seq(min(x),max(x), by=1),x),labels=FALSE,col=2, lwd.ticks=2)
axis(1,at=seq(37,732, by=60),labels=seq(1960,2018,by=5), lwd.ticks=2, cex.axis=1.5)
polygon(c(x,rev(x)),c(pd2,rev(pd1)),col=rgb(0.,0.,0.5),lty=2)
lines(x, predict(mod.loess33), lwd=2.5, col='gray')
xp=seq(min(x),max(x),by=0.1)
lines(xp, predict(mod.lm, newdata=data.frame(x=xp)), lwd=4.4, col=col[1])
lines(xp, predict(mod.ar, newdata=data.frame(x=xp)), lwd=4.3, col=col[2])
lines(xp, predict(mod.war, newdata=data.frame(x=xp)), lwd=4.2, col=col[3])
segments(x0=min(x), y0=summary(mod.lts)$coefficients[1,1] + summary(mod.lts)$coefficients[2,1]*min(x), x1=max(x),
y1=summary(mod.lts)$coefficients[1,1] + summary(mod.lts)$coefficients[2,1]*max(x), lwd=3.1, col=col[4])
lines(xp, predict(mod.ll, newx=cbind(1,x=xp), type="response"), lwd=3.0, col=col[5])
lines(xp, predict(mod.l2, newx=cbind(1,x=xp), type="response"), lwd=2.9, col=col[6])
lines(xp, predict(mod.m1, newdata=data.frame(x=xp)), lwd=2.8, col=col[7])
lines(xp, predict(mod.m2, newdata=data.frame(x=xp)), lwd=2.7, col=col[8])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp)),[2], lwd=2.6, col=col[9])
lines(xp, predict(mod.lms, newdata=data.frame(x=xp)), lwd=2.5, col=col[10])
#legend(550,max(y)+1, c("OLS","GLS-ARI", "WLS", "LTS", "Lasso"), lwd=2.,
# text.col=col[c(1,2,3,4,5)], col=col[c(1,2,3,4,5)], lty=0, cex=2., bty='n')
#legend(640,max(y)+1, c("Ridge","Sen-Theil", "Siegel", "QR-50th", "LMS"), lwd=2.,
#text.col=col[c(6,7,8,9,10)], col=col[c(6,7,8,9,10)], lty=0, cex=2., bty='n')
legend(600,20, "Schwarzwald-Sued", lwd=2., lty=0, cex=2.5, bty='n')

mtext(text="Year",side=1,line=2,outer=TRUE,cex=1.5)
dev.off()

#####
#Figure 8
#####

i=8
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]
yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month))[,3])
mod=gls(y~x, correlation=corAR1(form=-x))
ar=(exp(mod$modelStruct$corStruct[1])-1)/(exp(mod$modelStruct$corStruct[1])+1)
Cov=matrix(NA, ncol=length(y), nrow=length(y))
for (k in 1:length(y)){
  for (j in 1:length(y)){
    Cov[k,j]=ar^abs(k-j)
  }
}
L=solve(t(chol(Cov)))
mod.rq=summary(rq(y~x, tau=seq(0.05,0.95,by=0.05)), se="boot")
mod.rq.ar=summary(rq(L%*%y~-1+L%*%cbind(1,x), tau=seq(0.05,0.95,by=0.05)), se="boot")
lq=length(seq(0.05,0.95,by=0.05))
sqr=rep(0,lq)
lqr=rep(0,lq)
uqr=rep(0,lq)
for (j in 1:lq){
  sqr[j]=mod.rq[[j]]$coefficients[2,1]*120
  lqr[j]=(mod.rq[[j]]$coefficients[2,1]-2*mod.rq[[j]]$coefficients[2,2])*120
  uqr[j]=(mod.rq[[j]]$coefficients[2,1]+2*mod.rq[[j]]$coefficients[2,2])*120}
sqr.ar=rep(0,lq)
lqr.ar=rep(0,lq)
uqr.ar=rep(0,lq)
for (j in 1:lq){
  sqr.ar[j]=mod.rq.ar[[j]]$coefficients[2,1]*120
  lqr.ar[j]=(mod.rq.ar[[j]]$coefficients[2,1]-2*mod.rq.ar[[j]]$coefficients[2,2])*120
  uqr.ar[j]=(mod.rq.ar[[j]]$coefficients[2,1]+2*mod.rq.ar[[j]]$coefficients[2,2])*120}

png("trend_qrfit_remote.png", height=10, width=10, units="in", res=500)
par(mar=c(0, 4, 0, 1), mgp=c(2.4, 0.8, 0), mfrow=c(3,1), oma=c(4,2,0,0), las=1)
plot(seq(0.05,0.95,by=0.05), sqr.ar, xlab="", ylab="Slope_[ppb/decade]", ylim=c(-1, 3.5), type='b', cex.lab=2, xaxt="n", yaxt="n", lwd=3, col=1)
axis(2,cex.axis=1.3)
axis(1,at=seq(0.05,0.95,by=0.05),labels=seq(5,95,by=5), cex.axis=1.3)
polygon(c(seq(0.05,0.95,by=0.05),rev(seq(0.05,0.95,by=0.05))), c(uqr.ar,rev(lqr.ar)),col=rgb(0.87,0.87,0.87,0.5),lty=2)
abline(h=0, col=1, lwd=2, lty=3)

fit2 = gls(y~x, correlation=corAR1(form=-x))
abline(h=summary(fit2)$tTable[2,1]*120, col=2, lwd=3, lty=1)
abline(h=120*(summary(fit2)$tTable[2,1] + qt(0.975,fit2$dims$N-2)*summary(fit2)$tTable[2,2]), col=2, lwd=3, lty=2)
abline(h=120*(summary(fit2)$tTable[2,1] + qt(0.025,fit2$dims$N-2)*summary(fit2)$tTable[2,2]), col=2, lwd=3, lty=2)

lines(seq(0.05,0.95,by=0.05), sqr.ar, lwd=3, col=1, type="b")
legend(0.3, 3.6, c("GLS", "QR"), lwd=2.5, text.col=c(2,1), col=c(2,1), lty=c(0,0), cex=3., bty='n')
legend(0.6, 3.6, "Mace_Head", lwd=2., lty=0, cex=2.5, bty='n')

i=16
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]

```



```

yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month))[,3])
mod=gls(y~x, correlation=corAR1(form=-x))
ar=(exp(mod$modelStruct$corStruct[1])-1)/(exp(mod$modelStruct$corStruct[1])+1)
Cov=matrix(NA, ncol=length(y), nrow=length(y))
for (k in 1:length(y)){
  for (j in 1:length(y)){
    Cov[k,j]=ar^abs(k-j)
  }
}
L=solve(t(chol(Cov)))
mod.rq=summary(rq(y~x, tau=seq(0.05,0.95,by=0.05)), se="boot")
mod.rq.ar=summary(rq(L%*%y~-1+L%*%cbind(1,x), tau=seq(0.05,0.95,by=0.05)), se="boot")
lq=length(seq(0.05,0.95,by=0.05))
sqr=rep(0,lq)
lqr=rep(0,lq)
uqr=rep(0,lq)
for (j in 1:lq){
  sqr[j]=mod.rq[[j]]$coefficients[2,1]*120
  lqr[j]=(mod.rq[[j]]$coefficients[2,1]-2*mod.rq[[j]]$coefficients[2,2])*120
  uqr[j]=(mod.rq[[j]]$coefficients[2,1]+2*mod.rq[[j]]$coefficients[2,2])*120}
sqr.ar=rep(0,lq)
lqr.ar=rep(0,lq)
uqr.ar=rep(0,lq)
for (j in 1:lq){
  sqr.ar[j]=mod.rq.ar[[j]]$coefficients[2,1]*120
  lqr.ar[j]=(mod.rq.ar[[j]]$coefficients[2,1]-2*mod.rq.ar[[j]]$coefficients[2,2])*120
  uqr.ar[j]=(mod.rq.ar[[j]]$coefficients[2,1]+2*mod.rq.ar[[j]]$coefficients[2,2])*120}

plot(seq(0.05,0.95,by=0.05), sqr.ar, xlab="", ylab="Slope_[ppb/decade]", ylim=c(0, 5), type='b', cex.lab=2, xaxt="n", yaxt="n",
      lwd=3, col=1)
axis(2,cex.axis=1.3)
axis(1,at=seq(0.05,0.95,by=0.05),labels=seq(5,95,by=5), cex.axis=1.3)
polygon(c(seq(0.05,0.95,by=0.05),rev(seq(0.05,0.95,by=0.05))), c(uqr.ar,rev(lqr.ar)),col=rgb(0.87,0.87,0.87,0.5),lty=2)
abline(h=0, col=1, lwd=2, lty=3)

fit2 = gls(y~x, correlation=corAR1(form=-x))
abline(h=summary(fit2)$tTable[2,1]*120, col=2, lwd=3, lty=1)
abline(h=120*(summary(fit2)$tTable[2,1] + qt(0.975,fit2$dims$N-2)*summary(fit2)$tTable[2,2]), col=2, lwd=3, lty=2)
abline(h=120*(summary(fit2)$tTable[2,1] + qt(0.025,fit2$dims$N-2)*summary(fit2)$tTable[2,2]), col=2, lwd=3, lty=2)

lines(seq(0.05,0.95,by=0.05), sqr.ar, lwd=3, col=1, type="b")
legend(0., 5., "Mt. Waliguan", lwd=2., lty=0, cex=2.5, bty="n")

i=30
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]
yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month))[,3])
mod=gls(y~x, correlation=corAR1(form=-x))
ar=(exp(mod$modelStruct$corStruct[1])-1)/(exp(mod$modelStruct$corStruct[1])+1)
Cov=matrix(NA, ncol=length(y), nrow=length(y))
for (k in 1:length(y)){
  for (j in 1:length(y)){
    Cov[k,j]=ar^abs(k-j)
  }
}
L=solve(t(chol(Cov)))
mod.rq=summary(rq(y~x, tau=seq(0.05,0.95,by=0.05)), se="boot")
mod.rq.ar=summary(rq(L%*%y~-1+L%*%cbind(1,x), tau=seq(0.05,0.95,by=0.05)), se="boot")
lq=length(seq(0.05,0.95,by=0.05))
sqr=rep(0,lq)
lqr=rep(0,lq)
uqr=rep(0,lq)
for (j in 1:lq){
  sqr[j]=mod.rq[[j]]$coefficients[2,1]*120
  lqr[j]=(mod.rq[[j]]$coefficients[2,1]-2*mod.rq[[j]]$coefficients[2,2])*120
  uqr[j]=(mod.rq[[j]]$coefficients[2,1]+2*mod.rq[[j]]$coefficients[2,2])*120}
sqr.ar=rep(0,lq)
lqr.ar=rep(0,lq)
uqr.ar=rep(0,lq)
for (j in 1:lq){
  sqr.ar[j]=mod.rq.ar[[j]]$coefficients[2,1]*120
  lqr.ar[j]=(mod.rq.ar[[j]]$coefficients[2,1]-2*mod.rq.ar[[j]]$coefficients[2,2])*120
  uqr.ar[j]=(mod.rq.ar[[j]]$coefficients[2,1]+2*mod.rq.ar[[j]]$coefficients[2,2])*120}

plot(seq(0.05,0.95,by=0.05), sqr.ar, xlab="", ylab="Slope_[ppb/decade]", ylim=c(-7, 2.5), type='b', cex.lab=2, xaxt="n", yaxt="n",
      lwd=3, col=1)
axis(2,cex.axis=1.3)
axis(1,at=seq(0.05,0.95,by=0.05),labels=seq(5,95,by=5), cex.axis=1.3)
polygon(c(seq(0.05,0.95,by=0.05),rev(seq(0.05,0.95,by=0.05))), c(uqr.ar,rev(lqr.ar)),col=rgb(0.87,0.87,0.87,0.5),lty=2)
abline(h=0, col=1, lwd=2, lty=3)

fit2 = gls(y~x, correlation=corAR1(form=-x))
abline(h=summary(fit2)$tTable[2,1]*120, col=2, lwd=3, lty=1)
abline(h=120*(summary(fit2)$tTable[2,1] + qt(0.975,fit2$dims$N-2)*summary(fit2)$tTable[2,2]), col=2, lwd=3, lty=2)
abline(h=120*(summary(fit2)$tTable[2,1] + qt(0.025,fit2$dims$N-2)*summary(fit2)$tTable[2,2]), col=2, lwd=3, lty=2)

lines(seq(0.05,0.95,by=0.05), sqr.ar, lwd=3, col=1, type="b")
legend(0.6,2.6, "Schwarzwald-Sued", lwd=2., lty=0, cex=2.5, bty="n")

mtext(text="Percentile",side=1,line=2,outer=TRUE,cex=1.5)
dev.off()

```

```

#####
#Same as Figure 7, but only quantile trends are shown
#####

png("trend_fit_qrts_remote.png", height=10, width=10, units="in", res=500)
par(mar=c(0, 4, 0, 1), mgp=c(2.4, 0.8, 0), mfrow=c(3,1), oma=c(3,2,0,0), las=1)
i=8
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]
yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month)))[,3]
mod.rq=rq(y~x, tau=seq(0.05, 0.95, by=0.05))

plot(dt$ind,dt[,i+3], xlab="", ylab="Ozone_[ppb]", cex.lab=2, type='b', xaxt="n", yaxt="n", lwd=1.5, xlim=c(379,732))
axis(2,lwd.ticks=2, cex.axis=1.5)
axis(1,at=setdiff(seq(min(x),max(x), by=1),x),labels=FALSE,col=2, lwd.ticks=2)
#axis(1,at=seq(37,732, by=60),labels=seq(1960,2018,by=5), lwd.ticks=2, cex.axis=1.5)
xp=seq(min(x),max(x),by=0.1)
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,2], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,4], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,6], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,8], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,10], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,12], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,14], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,16], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,18], lwd=2.6, col=col[9])
legend(600,-7, "MaceHead", lwd=2., lty=0, cex=2.5, bty='n')

i=16
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]
yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month)))[,3]
mod.rq=rq(y~x, tau=seq(0.05, 0.95, by=0.05))

plot(dt$ind,dt[,i+3], xlab="", ylab="Ozone_[ppb]", cex.lab=2, type='b', xaxt="n", yaxt="n", lwd=1.5, xlim=c(379,732))
axis(2,lwd.ticks=2, cex.axis=1.5)
axis(1,at=setdiff(seq(min(x),max(x), by=1),x),labels=FALSE,col=2, lwd.ticks=2)
#axis(1,at=seq(37,732, by=60),labels=seq(1960,2018,by=5), lwd.ticks=2, cex.axis=1.5)
xp=seq(min(x),max(x),by=0.1)
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,2], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,4], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,6], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,8], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,10], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,12], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,14], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,16], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,18], lwd=2.6, col=col[9])
legend(345,max(y)+1, c("From_90th_percentile_(top_line)", "to_10th_percentile_(bottom_line)", "by_every_10th_percentile"),
lwd=2.,
text.col=col[9], col=col[9], lty=0, cex=2.3, bty='n')
legend(600,-3, "Mt. Waliguan", lwd=2., lty=0, cex=2.5, bty='n')

i=30
y=na.omit(cbind(dt[,i+3], dt$ind))[,1]
x=na.omit(cbind(dt[,i+3], dt$ind))[,2]
yr=na.omit(cbind(dt[,i+3], dt$ind, dt$year))[,3]
mo=as.factor(na.omit(cbind(dt[,i+3], dt$ind, dt$month)))[,3]
mod.rq=rq(y~x, tau=seq(0.05, 0.95, by=0.05))

plot(dt$ind,dt[,i+3], xlab="Year", ylab="Ozone_[ppb]", cex.lab=2, type='b', xaxt="n", yaxt="n", lwd=1.5, xlim=c(379,732))
axis(2,lwd.ticks=2, cex.axis=1.5)
axis(1,at=setdiff(seq(min(x),max(x), by=1),x),labels=FALSE,col=2, lwd.ticks=2)
axis(1,at=seq(37,732, by=60),labels=seq(1960,2018,by=5), lwd.ticks=2, cex.axis=1.5)
xp=seq(min(x),max(x),by=0.1)
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,2], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,4], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,6], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,8], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,10], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,12], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,14], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,16], lwd=2.6, col=col[9])
lines(xp, predict(mod.rq, newdata=data.frame(x=xp))[,18], lwd=2.6, col=col[9])
legend(600,20, "Schwarzwald-Sued", lwd=2., lty=0, cex=2.5, bty='n')

mtext(text="Year",side=1,line=2,outer=TRUE,cex=1.5)
dev.off()

```

S7 R code for Section 6

```
#####  
#This code produces Figures 11–12 in the paper, using the class of (quantile)  
#generalized additive models for the analysis of a single time series.  
#####  
  
#####  
#load packages  
#####  
  
setwd("C:/Users/User/Dropbox/2.CurrentProjects/TrendDetection/toar")  
col=c(rgb(0.6627, 0.6627, 0.6627),  
       rgb(0.2081, 0.1663, 0.5292),  
       rgb(0.3961, 0.3176, 0.8000),  
       rgb(0.0123, 0.4213, 0.8802),  
       rgb(0.4941, 0.7647, 0.8980),  
       rgb(0.1157, 0.7022, 0.6843),  
       rgb(0.5216, 0.6980, 0.1725),  
       rgb(0.9968, 0.7513, 0.2325),  
       rgb(1.0000, 0.4863, 0.0000),  
       rgb(0.8000, 0.3176, 0.3176),  
       rgb(0.6980, 0.1725, 0.1725))  
  
library(fields)  
library(mgcv)  
library(qgam)  
library(mblm)  
library(cartography)  
library(dichromat)  
dcol=colourschemes$DarkRedtoBlue.18  
dcol=c('blue_4',dcol,'red_4')  
dcol2=c(rev(carto.pal(pal1="purple.pal",n1=10)), carto.pal(pal1="orange.pal",n1=10))  
  
#####  
#read data and define the region (Southeast US)  
#Reference: Schultz, Martin G., et al. "Tropospheric Ozone Assessment Report: Database  
#and metrics data of global surface ozone observations." Elementa: Science of the  
#Anthropocene 5 (2017).  
#####  
  
annmean=read.csv("toar-dma8.csv")  
annmean=annmean[annmean$year<=2014,]  
ny=15  
date=cbind(year=rep(seq(2000,2014), each=6), month=rep(seq(4,9),ny), time=seq(1,6*ny))  
annmean=merge(date, annmean, by=c("year","month"), all=T)  
annmean$id= as.numeric(factor(annmean$station_id))  
  
annmean=annmean[annmean$station_id!=12227,]  
annmean=annmean[annmean$station_id!=17714,]  
annmean=annmean[annmean$station_id!=13550,]  
  
strs=readLines("TOAR_sfc_ozone_summer_global_2000-2014.csv")  
dataset=read.csv(text=strs, header=T, sep=";", skip=79, quote = "", fill=T)  
ann=dataset  
ann=ann[ann$station_htap_region==3,]  
ann=ann[ann$station_lon<=-109 & ann$station_lon>=-125,]  
ann=ann[ann$station_lat<=43 & ann$station_lat>=30,]  
ann=ann[ann$numid!=12227,]  
  
length(unique(annmean$station_id))  
annmean=annmean[annmean$station_id %in% ann$numid,]  
length(unique(annmean$station_id))  
  
#####  
#the upper panel of Figure 11  
#####  
  
annmeanx=annmean[annmean$year<=2002,]  
mod1=gam(dma8~s(station_lon, station_lat, bs="gp", k=60), data=annmeanx)  
aa=plot(mod1, select=2,scheme=2,n2=60,too.far=0.05)  
surf1=matrix(aa[[1]]$fit,60) + unlist(summary(mod1)$p.coeff[1])  
LL1=aa[[1]]$x  
ll1=aa[[1]]$y  
  
mod2=qgam(dma8~s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.5)  
aa=plot(mod2, select=2,scheme=2,n2=60,too.far=0.05)  
surf2=matrix(aa[[1]]$fit,60) + unlist(summary(mod2)$p.coeff[1])  
LL2=aa[[1]]$x  
ll2=aa[[1]]$y  
  
mod3=qgam(dma8~s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.05)  
aa=plot(mod3, select=2,scheme=2,n2=60,too.far=0.05)  
surf3=matrix(aa[[1]]$fit,60) + unlist(summary(mod3)$p.coeff[1])  
LL3=aa[[1]]$x  
ll3=aa[[1]]$y  
  
mod4=qgam(dma8~s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)  
aa=plot(mod4, select=2,scheme=2,n2=60,too.far=0.05)  
surf4=matrix(aa[[1]]$fit,60) + unlist(summary(mod4)$p.coeff[1])  
LL4=aa[[1]]$x  
ll4=aa[[1]]$y
```

```

annmeanx=annmean[annmean$year>=2012,]
mod5=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
aa=plot(mod5, select=2,scheme=2,n2=60,too.far=0.05)
surf5=matrix(aa[[1]]$fit,60) + unlist(summary(mod5)$p.coeff[1])
LL5=aa[[1]]$x
ll5=aa[[1]]$y

mod6=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
aa=plot(mod6, select=2,scheme=2,n2=60,too.far=0.05)
surf6=matrix(aa[[1]]$fit,60) + unlist(summary(mod6)$p.coeff[1])
LL6=aa[[1]]$x
ll6=aa[[1]]$y

mod7=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
aa=plot(mod7, select=2,scheme=2,n2=60,too.far=0.05)
surf7=matrix(aa[[1]]$fit,60) + unlist(summary(mod7)$p.coeff[1])
LL7=aa[[1]]$x
ll7=aa[[1]]$y

mod8=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.95)
aa=plot(mod8, select=2,scheme=2,n2=60,too.far=0.05)
surf8=matrix(aa[[1]]$fit,60) + unlist(summary(mod8)$p.coeff[1])
LL8=aa[[1]]$x
ll8=aa[[1]]$y

png("CA_summer_dma8_daily_kriging.png", height=10, width=15, units="in", res=500)
par(mar=c(3.5, 3.5, 2, 1), mgp=c(2.4, 0.8, 0), mfrow=c(2,3), las=1)
image.plot(LL3,ll3,surf3, xlab="",ylab="",horizontal=TRUE, main="5th_(2000-2002)", breaks=seq(10, 120, length.out=12),
  col=col, cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5))
points(cbind(annmeanx$station__lon, annmeanx$station__lat), lwd=1, col='white', pch=3)
US(add=TRUE, lwd=1.5)
image.plot(LL2,ll2,surf2, xlab="",ylab="",horizontal=TRUE, main="50th_(2000-2002)", breaks=seq(10, 120, length.out=12),
  col=col, cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5))
US(add=TRUE, lwd=1.5)
image.plot(LL4,ll4,surf4, xlab="",ylab="",horizontal=TRUE, main="95th_(2000-2002)", breaks=seq(10, 120, length.out=12),
  col=col, cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5))
US(add=TRUE, lwd=1.5)
image.plot(LL7,ll7,surf7, xlab="",ylab="",horizontal=TRUE, main="5th_(2012-2014)", breaks=seq(10, 120, length.out=12),
  col=col, cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5))
points(cbind(annmeanx$station__lon, annmeanx$station__lat), lwd=1, col='white', pch=3)
US(add=TRUE, lwd=1.5)
image.plot(LL6,ll6,surf6, xlab="",ylab="",horizontal=TRUE, main="50th_(2012-2014)", breaks=seq(10, 120, length.out=12),
  col=col, cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5))
US(add=TRUE, lwd=1.5)
image.plot(LL8,ll8,surf8, xlab="",ylab="",horizontal=TRUE, main="95th_(2012-2014)", breaks=seq(10, 120, length.out=12),
  col=col, cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5))
US(add=TRUE, lwd=1.5)
dev.off()

#####
#the lower panel of Figure 11
#####

nx=length(seq(-123,-109,by=0.1))
ny=length(seq(32,42, by=0.1))
ndt=expand.grid(station__lon=seq(-123,-109,by=0.1), station__lat=seq(32,42, by=0.1))
annmeanx=annmean[annmean$year==2014,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf14=predict(mod, newdata=ndt)

aa=plot(mod, select=2,scheme=2,n2=100,too.far=0.05)
obj=list(x=aa[[1]]$x, y=aa[[1]]$y, z=matrix(aa[[1]]$fit, nrow=100))
loc=make.surface.grid(list(seq(-123,-109,by=0.1), seq(32,42, by=0.1)))
intep=interp.surface(obj, loc)
image.plot(as.surface(loc, intep))
intep=intep*0
intep=matrix(intep, nrow=nx)

annmeanx=annmean[annmean$year==2013,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf13=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2012,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf12=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2011,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf11=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2010,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf10=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2009,]

```

```

mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf09=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2008,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf08=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2007,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf07=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2006,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf06=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2005,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf05=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2004,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf04=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2003,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf03=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2002,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf02=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2001,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf01=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2000,]
mod=gam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx)
surf00=predict(mod, newdata=ndt)

surf00=cbind(ndt, po3=surf00, yr=0)
surf01=cbind(ndt, po3=surf01, yr=1)
surf02=cbind(ndt, po3=surf02, yr=2)
surf03=cbind(ndt, po3=surf03, yr=3)
surf04=cbind(ndt, po3=surf04, yr=4)
surf05=cbind(ndt, po3=surf05, yr=5)
surf06=cbind(ndt, po3=surf06, yr=6)
surf07=cbind(ndt, po3=surf07, yr=7)
surf08=cbind(ndt, po3=surf08, yr=8)
surf09=cbind(ndt, po3=surf09, yr=9)
surf10=cbind(ndt, po3=surf10, yr=10)
surf11=cbind(ndt, po3=surf11, yr=11)
surf12=cbind(ndt, po3=surf12, yr=12)
surf13=cbind(ndt, po3=surf13, yr=13)
surf14=cbind(ndt, po3=surf14, yr=14)

kdt=rbind(surf00,surf01,surf02,surf03,surf04,surf05,surf06,surf07,surf08,surf09,surf10,surf11,surf12,surf13,surf14)

lt=matrix(0,nrow=nx,ncol=ny)
ls=matrix(0,nrow=nx,ncol=ny)
pv=matrix(0,nrow=nx,ncol=ny)
ar=matrix(0,nrow=nx,ncol=ny)
for (i in 1:nx){
  for (j in 1:ny){
    x=-123+0.1*(i-1)
    y=32+0.1*(j-1)
    cdt=kdt[kdt$station__lon==x & kdt$station__lat==y,]
    trend=gls(po3~yr, data=cdt, correlation=corAR1(form=-yr), na.action=na.omit)
    lt[i,j]=summary(trend,se="boot")$tTable[2,1]
    ls[i,j]=summary(trend,se="boot")$tTable[2,2]
    pv[i,j]=summary(trend,se="boot")$tTable[2,4]
    ar[i,j]=(exp(trend$modelStruct$corStruct[1])-1)/(exp(trend$modelStruct$corStruct[1])+1)
  }
}
ls=ls+intep
lt=lt+intep
lt[lt>=2]=1.99
lt[lt<=-2]=-1.99
ln=lt/ls
ln[ln>=5]=4.99
ln[ln<=-5]=-4.99

nx=length(seq(-123,-109,by=0.1))
ny=length(seq(32,42, by=0.1))
ndt=expand.grid(station__lon=seq(-123,-109,by=0.1), station__lat=seq(32,42, by=0.1))
annmeanx=annmean[annmean$year==2014,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf14=predict(mod, newdata=ndt)

aa=plot(mod, select=2,scheme=2,n2=100,too.far=0.05)
obj=list(x=aa[[1]]$x, y=aa[[1]]$y, z=matrix(aa[[1]]$fit, nrow=100))
loc=make.surface.grid(list(seq(-123,-109,by=0.1), seq(32,42, by=0.1)))
intep=interp.surface(obj, loc)
image.plot(as.surface(loc, intep))

```

```

intep=intep*0
intep=matrix(intep, nrow=nx)

annmeanx=annmean[annmean$Year==2013,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf13=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2012,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf12=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2011,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf11=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2010,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf10=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2009,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf09=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2008,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf08=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2007,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf07=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2006,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf06=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2005,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf05=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2004,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf04=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2003,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf03=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2002,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf02=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2001,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf01=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$Year==2000,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf00=predict(mod, newdata=ndt)

surf00=cbind(ndt, po3=surf00, yr=0)
surf01=cbind(ndt, po3=surf01, yr=1)
surf02=cbind(ndt, po3=surf02, yr=2)
surf03=cbind(ndt, po3=surf03, yr=3)
surf04=cbind(ndt, po3=surf04, yr=4)
surf05=cbind(ndt, po3=surf05, yr=5)
surf06=cbind(ndt, po3=surf06, yr=6)
surf07=cbind(ndt, po3=surf07, yr=7)
surf08=cbind(ndt, po3=surf08, yr=8)
surf09=cbind(ndt, po3=surf09, yr=9)
surf10=cbind(ndt, po3=surf10, yr=10)
surf11=cbind(ndt, po3=surf11, yr=11)
surf12=cbind(ndt, po3=surf12, yr=12)
surf13=cbind(ndt, po3=surf13, yr=13)
surf14=cbind(ndt, po3=surf14, yr=14)

kdt=rbind(surf00,surf01,surf02,surf03,surf04,surf05,surf06,surf07,surf08,surf09,surf10,surf11,surf12,surf13,surf14)

lt=matrix(0,nrow=nx,ncol=ny)
ls=matrix(0,nrow=nx,ncol=ny)
pv=matrix(0,nrow=nx,ncol=ny)
ar=matrix(0,nrow=nx,ncol=ny)
for (i in 1:nx){
  for (j in 1:ny){
    x=-123+0.1*(i-1)
    y=32+0.1*(j-1)
    cdt=kdt[kdt$station__lon==x & kdt$station__lat==y,]
    trend=glm(po3-yr, data=cdt, correlation=corAR1(form=~yr), na.action=na.omit)
    lt[i,j]=summary(trend,se="boot")$tTable[2,1]
    ls[i,j]=summary(trend,se="boot")$tTable[2,2]
    pv[i,j]=summary(trend,se="boot")$tTable[2,4]
    ar[i,j]=(exp(trend$modelStruct$corStruct[1])-1)/(exp(trend$modelStruct$corStruct[1])+1)
  }
}
ls=ls+intep
lt=lt+intep

```

```

lt[lt>=2]=1.99
lt[lt<=-2]=-1.99
ln=lt/ls
ln[ln>=5]=4.99
ln[ln<=-5]=-4.99
lt2=lt
ln2=ln

nx=length(seq(-123,-109,by=0.1))
ny=length(seq(32,42, by=0.1))
ndt=expand.grid(station__lon=seq(-123,-109,by=0.1), station__lat=seq(32,42, by=0.1))
annmeanx=annmean[annmean$year==2014,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf14=predict(mod, newdata=ndt)

aa=plot(mod, select=2,scheme=2,n2=100,too.far=0.05)
obj=list(x=aa[[1]]$x, y=aa[[1]]$y, z=matrix(aa[[1]]$fit, nrow=100))
loc=make.surface.grid(list(seq(-123,-109,by=0.1), seq(32,42, by=0.1)))
intep=interp.surface(obj, loc)
image.plot(as.surface(loc, intep))
intep=intep*0
intep=matrix(intep, nrow=nx)

annmeanx=annmean[annmean$year==2013,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf13=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2012,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf12=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2011,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf11=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2010,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf10=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2009,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf09=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2008,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf08=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2007,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf07=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2006,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf06=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2005,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf05=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2004,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf04=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2003,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf03=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2002,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf02=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2001,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf01=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2000,]
mod=qgam(dma8-s(station__lon, station__lat, bs="gp", k=60), data=annmeanx, qu=0.05)
surf00=predict(mod, newdata=ndt)

surf00=cbind(ndt, po3=surf00, yr=0)
surf01=cbind(ndt, po3=surf01, yr=1)
surf02=cbind(ndt, po3=surf02, yr=2)
surf03=cbind(ndt, po3=surf03, yr=3)
surf04=cbind(ndt, po3=surf04, yr=4)
surf05=cbind(ndt, po3=surf05, yr=5)
surf06=cbind(ndt, po3=surf06, yr=6)
surf07=cbind(ndt, po3=surf07, yr=7)
surf08=cbind(ndt, po3=surf08, yr=8)
surf09=cbind(ndt, po3=surf09, yr=9)
surf10=cbind(ndt, po3=surf10, yr=10)
surf11=cbind(ndt, po3=surf11, yr=11)
surf12=cbind(ndt, po3=surf12, yr=12)
surf13=cbind(ndt, po3=surf13, yr=13)
surf14=cbind(ndt, po3=surf14, yr=14)

```

```

kdt=rbind(surf00,surf01,surf02,surf03,surf04,surf05,surf06,surf07,surf08,surf09,surf10,surf11,surf12,surf13,surf14)

lt=matrix(0,nrow=nx,ncol=ny)
ls=matrix(0,nrow=nx,ncol=ny)
pv=matrix(0,nrow=nx,ncol=ny)
ar=matrix(0,nrow=nx,ncol=ny)
for (i in 1:nx){
  for (j in 1:ny){
    x=-123+0.1*(i-1)
    y=32+0.1*(j-1)
    cdt=kdt[kdt$station_lon==x & kdt$station_lat==y,]
    trend=glsp(po3-yr, data=cdt, correlation=corAR1(form=-yr), na.action=na.omit)
    lt[i,j]=summary(trend,se="boot")$tTable[2,1]
    ls[i,j]=summary(trend,se="boot")$tTable[2,2]
    pv[i,j]=summary(trend,se="boot")$tTable[2,4]
    ar[i,j]=(exp(trend$modelStruct$corStruct[1])-1)/(exp(trend$modelStruct$corStruct[1])+1)
  }
}
ls=ls+intep
lt=lt+intep
lt[lt>=2]=1.99
lt[lt<=-2]=-1.99
ln=lt/ls
ln[ln>=5]=4.99
ln[ln<=-5]=-4.99
lt3=lt
ln3=ln

nx=length(seq(-123,-109,by=0.1))
ny=length(seq(32,42, by=0.1))
ndt=expand.grid(station_lon=seq(-123,-109,by=0.1), station_lat=seq(32,42, by=0.1))
annmeanx=annmean[annmean$year==2014,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf14=predict(mod, newdata=ndt)

aa=plot(mod, select=2,scheme=2,n2=100,too.far=0.05)
obj=list(x=aa[[1]]$x, y=aa[[1]]$y, z=matrix(aa[[1]]$fit, nrow=100))
loc=make.surface.grid(list(seq(-123,-109,by=0.1), seq(32,42, by=0.1)))
intep=interp.surface(obj, loc)
image.plot(as.surface(loc, intep))
intep=intep*0
intep=matrix(intep, nrow=nx)

annmeanx=annmean[annmean$year==2013,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf13=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2012,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf12=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2011,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf11=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2010,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf10=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2009,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf09=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2008,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf08=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2007,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.5)
surf07=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2006,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf06=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2005,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf05=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2004,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf04=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2003,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf03=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2002,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf02=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2001,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)

```



```

surf01=predict(mod, newdata=ndt)

annmeanx=annmean[annmean$year==2000,]
mod=qgam(dma8-s(station_lon, station_lat, bs="gp", k=60), data=annmeanx, qu=0.95)
surf00=predict(mod, newdata=ndt)

surf00=cbind(ndt, po3=surf00, yr=0)
surf01=cbind(ndt, po3=surf01, yr=1)
surf02=cbind(ndt, po3=surf02, yr=2)
surf03=cbind(ndt, po3=surf03, yr=3)
surf04=cbind(ndt, po3=surf04, yr=4)
surf05=cbind(ndt, po3=surf05, yr=5)
surf06=cbind(ndt, po3=surf06, yr=6)
surf07=cbind(ndt, po3=surf07, yr=7)
surf08=cbind(ndt, po3=surf08, yr=8)
surf09=cbind(ndt, po3=surf09, yr=9)
surf10=cbind(ndt, po3=surf10, yr=10)
surf11=cbind(ndt, po3=surf11, yr=11)
surf12=cbind(ndt, po3=surf12, yr=12)
surf13=cbind(ndt, po3=surf13, yr=13)
surf14=cbind(ndt, po3=surf14, yr=14)

kdt=rbind(surf00,surf01,surf02,surf03,surf04,surf05,surf06,surf07,surf08,surf09,surf10,surf11,surf12,surf13,surf14)

lt=matrix(0,nrow=nx,ncol=ny)
ls=matrix(0,nrow=nx,ncol=ny)
pv=matrix(0,nrow=nx,ncol=ny)
ar=matrix(0,nrow=nx,ncol=ny)
for (i in 1:nx){
  for (j in 1:ny){
    x=-123+0.1*(i-1)
    y=32+0.1*(j-1)
    cdt=kdt[kdt$station_lon==x & kdt$station_lat==y,]
    trend=glm(po3-yr, data=cdt, correlation=corAR1(form=-yr), na.action=na.omit)
    lt[i,j]=summary(trend,se="boot")$tTable[2,1]
    ls[i,j]=summary(trend,se="boot")$tTable[2,2]
    pv[i,j]=summary(trend,se="boot")$tTable[2,4]
    ar[i,j]=(exp(trend$modelStruct$corStruct[1])-1)/(exp(trend$modelStruct$corStruct[1])+1)
  }
}
ls=ls+intep
lt=lt+intep
lt[lt>=2]=1.99
lt[lt<=-2]=-1.99
ln=lt/ls
ln[ln>=5]=4.99
ln[ln<=-5]=-4.99
lt4=lt
ln4=ln

cx=seq(-123,-109,by=0.1)
cy=seq(32,42, by=0.1)
lab.breaks=c('<-2', rep(NA,4), -1, rep(NA,4), 0, rep(NA,4), 1, rep(NA,4), '>2')
png("CA_summer_dma8_daily_kriging_trends.png", height=10, width=15, units="in", res=500)
par(mar=c(3.5, 3.5, 2, 1), mgp=c(2.4, 0.8, 0), mfrow=c(2,3), las=1)
image.plot(cx,cy,lt3, xlab="",ylab="",horizontal=TRUE, main="5th_trends", breaks=seq(-2, 2, length.out=21), col=dcol2,
  cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5), lab.breaks=lab.breaks)
US(add=TRUE, lwd=1.5)
points(cbind(annmean$station_lon, annmean$station_lat), lwd=1, col='white', pch=3)
image.plot(cx,cy,lt2, xlab="",ylab="",horizontal=TRUE, main="50th_trends", breaks=seq(-2, 2, length.out=21), col=dcol2,
  cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5), lab.breaks=lab.breaks)
US(add=TRUE, lwd=1.5)
image.plot(cx,cy,lt4, xlab="",ylab="",horizontal=TRUE, main="95th_trends", breaks=seq(-2, 2, length.out=21), col=dcol2,
  cex.main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5), lab.breaks=lab.breaks)
US(add=TRUE, lwd=1.5)
image.plot(cx,cy,ln3, xlab="",ylab="",horizontal=TRUE, main="5th_SNR", breaks=seq(-5, 5, length.out=21), col=dcol, cex.
  main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5), lab.breaks=c('<-5',round(seq
    (-4.5,4.5,by=0.5),1), '>5'))
US(add=TRUE, lwd=1.5)
image.plot(cx,cy,ln2, xlab="",ylab="",horizontal=TRUE, main="50th_SNR", breaks=seq(-5, 5, length.out=21), col=dcol, cex
  .main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5), lab.breaks=c('<-5',round(seq
    (-4.5,4.5,by=0.5),1), '>5'))
US(add=TRUE, lwd=1.5)
image.plot(cx,cy,ln4, xlab="",ylab="",horizontal=TRUE, main="95th_SNR", breaks=seq(-5, 5, length.out=21), col=dcol, cex
  .main=2,
  ylim=c(32,41.9), xlim=c(-123,-109.5), cex.axis=1.4, axis.args=list(cex.axis=1.5), lab.breaks=c('<-5',round(seq
    (-4.5,4.5,by=0.5),1), '>5'))
US(add=TRUE, lwd=1.5)
dev.off()

#additional figure shows the scatter of OLS trend value vs p-value
annmean$dma8.pv=NA
annmean$dma8.tr=NA
annmean$dma8.2s=NA
for (i in unique(annmean$station_id)){
  it=annmean[annmean$station_id==i,]
  mod.ar=lm(dma8-time, it, na.action=na.pass)

```

```

annmean[annmean$station_id==i,$dma8.pv=summary(mod.ar)$coefficients[2,4]
annmean[annmean$station_id==i,$dma8.tr=summary(mod.ar)$coefficients[2,1]
annmean[annmean$station_id==i,$dma8.2s=summary(mod.ar)$coefficients[2,2]*2
}
ar1.dt=aggregate(cbind(dma8.pv,dma8.tr,dma8.2s)-station_id, data=annmean, FUN=function(x) {mean(x,na.rm=T)})

library(ggplot2)
library(ggExtra)
library(gridExtra)
df = data.frame(p.value=ar1.dt$dma8.pv, trend=ar1.dt$dma8.tr)

png("CA_summer_2000-2014_ddma8x_hist.png", height=5, width=5, units="in", res=500)
p = ggplot(df, aes(p.value, trend)) + geom_point(pch='+', size=5) + theme_classic() + labs(x="P-value",y="Trend[ppb/
year]") +
geom_hline(yintercept=0, linetype="dotted", color="red", size=1)
ggExtra::ggMarginal(p, type = "histogram", bins=20)
dev.off()

#####
#the upper panel of Figure 12
#####

annmean1=aggregate(.-year+id, data=annmean, FUN=function(x) {mean(x,na.rm=T)})
annmean1$time=annmean1$year-1999
mod1=gam(dma8~ s(time, bs="cr", k=15) + s(station_lon, station_lat, bs="gp", k=60)+ti(time,id, bs=c("cr","fs"),d=c(1,1), k
=c(15,10)), data=annmean1)
summary(mod1)
aa=plot(mod1, select=2,scheme=2,n2=60,too.far=0.05)
surf1=matrix(aa[[2]]$fit,60) + unlist(summary(mod1)$p.coeff[1])
LL=aa[[2]]$x
ll=aa[[2]]$y

annmean2=aggregate(.-year+id, data=annmean, FUN=function(x) {quantile(x,probs=0.5, na.rm=T)})
annmean2$time=annmean2$year-1999
mod2=gam(dma8~ s(time, bs="cr", k=15) + s(station_lon, station_lat, bs="gp", k=60)+ti(time,id, bs=c("cr","fs"),d=c(1,1), k
=c(15,10)), data=annmean2)
summary(mod2)
aa=plot(mod2, select=2,scheme=2,n2=60,too.far=0.05)
surf2=matrix(aa[[2]]$fit,60) + unlist(summary(mod2)$p.coeff[1])

annmean3=aggregate(.-year+id, data=annmean, FUN=function(x) {quantile(x,probs=0.05, na.rm=T)})
annmean3$time=annmean3$year-1999
mod3=gam(dma8~ s(time, bs="cr", k=15) + s(station_lon, station_lat, bs="gp", k=60)+ti(time,id, bs=c("cr","fs"),d=c(1,1), k
=c(15,10)), data=annmean3)
summary(mod3)
aa=plot(mod3, select=2,scheme=2,n2=60,too.far=0.05)
surf3=matrix(aa[[2]]$fit,60) + unlist(summary(mod3)$p.coeff[1])

annmean4=aggregate(.-year+id, data=annmean, FUN=function(x) {quantile(x,probs=0.95, na.rm=T)})
annmean4$time=annmean4$year-1999
mod4=gam(dma8~ s(time, bs="cr", k=15) + s(station_lon, station_lat, bs="gp", k=60)+ti(time,id, bs=c("cr","fs"),d=c(1,1), k
=c(15,10)), data=annmean4)
summary(mod4)
aa=plot(mod4, select=2,scheme=2,n2=60,too.far=0.05)
surf4=matrix(aa[[2]]$fit,60) + unlist(summary(mod4)$p.coeff[1])

#not in the paper
png("CA_summer_2000-2014_dma8_daily.png", height=10, width=10, units="in", res=500)
par(mar=c(3.5, 3.5, 2, 1), mgp=c(2.4, 0.8, 0), mfrow=c(2,2), las=1)
image.plot(LL,ll,surf1, xlab="",ylab="",horizontal=TRUE, main="Mean", breaks=seq(10, 120, length.out=12), col=col, cex.
main=2)
points(cbind(annmean$station_lon, annmean$station_lat), lwd=1, col='white', pch=3)
US(add=TRUE, lwd=1.5)
image.plot(LL,ll,surf2, xlab="",ylab="",horizontal=TRUE, main="Median", breaks=seq(10, 120, length.out=12), col=col, cex.
main=2)
US(add=TRUE, lwd=1.5)
image.plot(LL,ll,surf3, xlab="",ylab="",horizontal=TRUE, main="5th_Percentile", breaks=seq(10, 120, length.out=12), col=
col, cex.main=2)
US(add=TRUE, lwd=1.5)
image.plot(LL,ll,surf4, xlab="",ylab="",horizontal=TRUE, main="95th_Percentile", breaks=seq(10, 120, length.out=12), col=
col, cex.main=2)
US(add=TRUE, lwd=1.5)
mtext('DU]', side=3, line=0, at=-100, cex=1.5)
dev.off()

time=seq(1,6*ny)
aa=plot(mod1, select=1, n=90)
ts1=aa[[1]]$fit + unlist(summary(mod1)$p.coeff[1])
aa=plot(mod2, select=1, n=90)
ts2=aa[[1]]$fit + unlist(summary(mod2)$p.coeff[1])
aa=plot(mod3, select=1, n=90)
ts3=aa[[1]]$fit + unlist(summary(mod3)$p.coeff[1])
aa=plot(mod4, select=1, n=90)
ts4=aa[[1]]$fit + unlist(summary(mod4)$p.coeff[1])
x=aa[[1]]$x

pd1=predict(mod1, type="terms", se.fit=T)
pd1=cbind(annmean1, rm=pd1$fit[,1] + unlist(summary(mod1)$p.coeff[1]), rs=pd1$se.fit[,1])
pd1=aggregate(cbind(rm,rs)-time, data=pd1, FUN=function(x) {mean(x,na.rm=T)})
rv=pd1$rs^2/mean(pd1$rs^2)
mod.mean=glm(rm~time, pd1, correlation=corAR1(form=~time))

```

```

mod.mean=gls(rm~time, pd1, correlation=corAR1(form=~time), weights=varIdent(~rv))

pd2=predict(mod2, type="terms", se.fit=T)
pd2=cbind(annmean2, rm=pd2$fit[,1]+ unlist(summary(mod2)$p.coeff[1]), rs=pd2$se.fit[,1])
pd2=aggregate(cbind(rm,rs)~time, data=pd2, FUN=function(x) {mean(x,na.rm=T)})
rv=pd2$rs^2/mean(pd2$rs^2)
mod.med=gls(rm~time, pd2, correlation=corAR1(form=~time), weights=varIdent(~rv))

pd3=predict(mod3, type="terms", se.fit=T)
pd3=cbind(annmean3, rm=pd3$fit[,1]+ unlist(summary(mod3)$p.coeff[1]), rs=pd3$se.fit[,1])
pd3=aggregate(cbind(rm,rs)~time, data=pd3, FUN=function(x) {mean(x,na.rm=T)})
rv=pd3$rs^2/mean(pd3$rs^2)
mod.5th=gls(rm~time, pd3, correlation=corAR1(form=~time), weights=varIdent(~rv))

pd4=predict(mod4, type="terms", se.fit=T)
pd4=cbind(annmean4, rm=pd4$fit[,1]+ unlist(summary(mod4)$p.coeff[1]), rs=pd4$se.fit[,1])
pd4=aggregate(cbind(rm,rs)~time, data=pd4, FUN=function(x) {mean(x,na.rm=T)})
rv=pd4$rs^2/mean(pd4$rs^2)
mod.95th=gls(rm~time, pd4, correlation=corAR1(form=~time), weights=varIdent(~rv))

png("CA_summer_2000-2014_dma8_ts_daily.png", height=10, width=20, units="in", res=500)
par(mar=c(3.5, 4, 2, 1), mgp=c(2.4, 0.8, 0), las=1)
plot(x, ts1, type="l", lwd=3, ylab="Ozone[ppb]", xlab="Year", cex.lab=2, xaxt="n", col=1, ylim=c(33,81), cex.axis=1.5)
abline(h=35, col='gray')
abline(h=40, col='gray')
abline(h=45, col='gray')
abline(h=50, col='gray')
abline(h=55, col='gray')
abline(h=60, col='gray')
abline(h=65, col='gray')
abline(h=70, col='gray')
abline(h=75, col='gray')
lines(x, ts4, col=1, lwd=3, lty=2)
lines(x, ts3, col=1, lwd=3, lty=2)
lines(x, ts2, col=1, lwd=3, lty=2)
lines(x, ts1, col=1, lwd=3, lty=1)

for (i in 1:90){
arrows(i, pd4[i,2], i, pd4[i,2]-2*pd4[i,3], length=0.05, code=2, col=1, lwd=2.5, angle=90, lty=2)
arrows(i, pd4[i,2], i, pd4[i,2]+2*pd4[i,3], length=0.05, code=2, col=1, lwd=2.5, angle=90, lty=2)
arrows(i, pd3[i,2], i, pd3[i,2]-2*pd3[i,3], length=0.05, code=2, col=1, lwd=2.5, angle=90, lty=2)
arrows(i, pd3[i,2], i, pd3[i,2]+2*pd3[i,3], length=0.05, code=2, col=1, lwd=2.5, angle=90, lty=2)
arrows(i, pd2[i,2], i, pd2[i,2]-2*pd2[i,3], length=0.05, code=2, col=1, lwd=2.5, angle=90, lty=2)
arrows(i, pd2[i,2], i, pd2[i,2]+2*pd2[i,3], length=0.05, code=2, col=1, lwd=2.5, angle=90, lty=2)
arrows(i, pd1[i,2], i, pd1[i,2]-2*pd1[i,3], length=0.05, code=2, col=1, lwd=2.5, angle=90, lty=1)
arrows(i, pd1[i,2], i, pd1[i,2]+2*pd1[i,3], length=0.05, code=2, col=1, lwd=2.5, angle=90, lty=1)
}

abline(a=summary(mod.5th)$tTable[1,1], b=summary(mod.5th)$tTable[2,1], lwd=3, lty=2, col=2)
abline(a=summary(mod.95th)$tTable[1,1], b=summary(mod.95th)$tTable[2,1], lwd=3, lty=2, col=2)
abline(a=summary(mod.med)$tTable[1,1], b=summary(mod.med)$tTable[2,1], lwd=3, lty=2, col=2)
abline(a=summary(mod.mean)$tTable[1,1], b=summary(mod.mean)$tTable[2,1], lwd=3, col=2)

axis(1,at=seq(1,15,by=2),labels=seq(2000,2014,by=2), cex.axis=1.5)
legend(7.85, c("Mean", "95th_(upper)", "50th_(middle)", "5th_(lower)"), lwd=2.5, text.col=1, lty=c(1,2,2,2),cex=3.0,bty='n',
ncol=2)
dev.off()

#####
#the lower panel Figure 11
#####

qu=c(0.01, seq(0.05,0.95,by=0.05), 0.99)
output=matrix(NA, ncol=2, nrow=length(qu))
for (q in 1:length(qu)){
annmeanx=aggregate(~year+id, data=annmean, FUN=function(x) {quantile(x,probs=qu[q], na.rm=T)})
annmeanx$time=annmeanx$year-1999
mod = gam(dma8~ s(time, bs="cr", k=15) + s(station_lon, station_lat, bs="gp", k=60)+ti(time,id, bs=c("cr","fs"),d=c(1,1),
k=c(15,10)), data=annmeanx)
pd=predict(mod, type="terms", se.fit=T)
pd=cbind(annmeanx, rm=pd$fit[,1]+ unlist(summary(mod)$p.coeff[1]), rs=pd$se.fit[,1])
pd=aggregate(cbind(rm,rs)~time, data=pd, FUN=function(x) {mean(x,na.rm=T)})
rv=pd$rs^2/mean(pd$rs^2)
mod=gls(rm~time, pd, correlation=corAR1(form=~time), weights=varIdent(~rv))
output[q,1]=summary(mod)$tTable[2,1]
output[q,2]=summary(mod)$tTable[2,2]
}

png("CA_summer_2000-2014_dma8_qr_daily.png", height=10, width=20, units="in", res=500)
par(mar=c(3.5, 4, 2, 1), mgp=c(2.4, 0.8, 0), las=1)
plot(qu, output[,1], xlab="Percentile", ylab="Slope_[ppb/year]", ylim=c(-1.0, 0.1), type='b', cex.lab=2, xaxt="n", yaxt="n",
lwd=3, col=1)
axis(2,cex.axis=1.5)
axis(1,at=qu, labels=qu*100, cex.axis=1.5)
polygon(c(qu,rev(qu)), c(output[,1]+2*output[,2],rev(output[,1]-2*output[,2])),col=rgb(0.87,0.87,0.87,0.5),lty=2)
abline(h=0, col=1, lwd=2, lty=3)

abline(h=summary(mod.mean)$tTable[2,1], col=2, lwd=4, lty=1)
abline(h=summary(mod.mean)$tTable[2,1] + 2*summary(mod.mean)$tTable[2,2], col=2, lwd=3, lty=2)
abline(h=summary(mod.mean)$tTable[2,1] - 2*summary(mod.mean)$tTable[2,2], col=2, lwd=3, lty=2)
lines(qu, output[,1],type='b', lwd=3, col=1)

```

```
legend(0.6, 0.07, c("GLS", "QR"), lwd=2.5, text.col=c(2,1), col=c(2,1), lty=c(0,0), cex=3.5, bty='n')  
dev.off()
```