Regression, Time Series and Spurious Relationships

Yang(Vivian) Tang

Advisors: Dr.Maureen Kennedy, Professor Rita Than

July 13, 2020

Abstract

In this paper, we introduce ordinary least-square regression analysis as the first approach to analyzing a real-world time series data set. We will then demonstrate that the use of regression analysis on autocorrelated time series could be problematic because the independent and identically distributed assumption of residuals is violated in our real world data set. Then we introduce the technique of prewhitening and apply it to a simulated data set. With the use of the prewhitening technique on the simulated data set, we show that we can correct the rate of detecting spurious relationships from around 40 percent to the correct significant level (0.05) at the sacrifice of statistical power.

Contents

1 Introduction

In today's data-rich society, data has become the competitive edge that can mean the difference between success and failure. How do we transform data into meaningful information? It would be beneficial if we could determine the correlation between variables and extract useful statistics to understand how the world runs. When deciding whether there is a correlation between two variables, the ordinary regression model is the first instinct for many people. However, how do we know if the relationship we found is true or spurious? A spurious relationship is a significant correlation in which two or more variables are statistically related, when in fact, more nuanced tests show they are not[Bur97].

In real life, an immense amount of data is time series data. One definition of time series is a sequence of observations taken sequentially in time[Box15].In the field of economics, we witness daily closing stock prices, monthly price indices, and yearly sales records. In nature, we observe monthly accumulated precipitation, average daily temperatures, and hourly wind speeds[CC11]. It is a common tendency for many people to do regression analysis on time series data, with or without acknowledging that many time series are highly autocorrelated. For example, the temperature of today is strongly correlated to the temperature of yesterday and the temperature of tomorrow will be strongly correlated to the temperature of today. In this paper, we will demonstrate the consequence of ignoring autocorrelation in time series data sets. We will first explain the concepts and terms in regression and time series. Then we will use real-world data, the Beijing PM2.5 data set, to show that using regression analysis on an autocorrelated time series can be problematic. Then, we will demonstrate the advantage of using time series analysis and the technique of prewhitening on a simulated data set. The results of the data set show us the severe consequence of modeling two autocorrelated time series. The results also show that we can avoid the server consequence by using the prewhitening technique with a long time series data.

2 Methodology

To understand the logic and to prepare to understand Sections 3 and 4 of this paper, it is essential to review the core concepts in regression analysis and time series analysis. This crucial information will provide a foundation for sections 3 and 4. In Section 3, we will use real-world data to demonstrate that the relationship found by regression modeling on highly autocorrelated data could result in a spurious correlation even if it is statistically significant. In section 4, we will use R to simulate autocorrelated data and test the rate of Type I errors and Type II errors before and after the method of prewhitening, which will be explained in section 2.2.8. In the following sections, therefore, we will first go over the core material of Ordinary Least-Square Regression then explain the essential information for time series analysis and modeling.

2.1 Ordinary Least-Square Regression

2.1.1 Ordinary Least-Square Regression Definition

In the book *Regression Analysis by Example*, the authors defined ordinary Least-Square as a data set consists of n observations on a dependent or response variable Y and p predictors or explanatory variables, $X_1, X_2, \ldots, X_p$. The meaning of the notation of Y and X_1 is shown below, where $\{1, 2, \ldots n\}$ are the observation numbers.

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} X_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \end{bmatrix}$$

The relationship between Y and $X_1, X_2, \ldots, X_p$ is formulated as a linear model shown in equation 1, where $\beta_0, \beta_1, \beta_2, ..., \beta_p$ are the regression parameters and ε is assumed to be a random error.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon \tag{1}$$

This model assumes that, with the set of fixed values of $X_1, X_2, , X_p$ that fall within the range of the data, if the standard regression assumptions are satisfied, the equation above provides an approximation of the true relationship between Y and Xs[CH13].

If we only estimate the relationship between one response variable and one predictor variable, we will use simple linear regression, which is one type of ordinary least-square regression. The following equation illustrates the model.

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon \tag{2}$$

The best fitted $y_i = \beta_0 + \beta_1 x_{1i} + \varepsilon_i$ will not be the same as the observation values. In order to estimate the "accurateness" and "goodness" of the fit, we often need to calculate the residuals. Residual, denoted as ε_i, is the i^{th} observation value minus the i^{th} fitted value. In the form of mathematical symbol, $\varepsilon_i = y_i - \beta_0 + \beta_{1i} x_i$[CH13].

2.1.2 The Standard Regression Assumption

The properties of least squares estimators and the statistical analysis of multiple linear regression based on four assumptions.

1. The first assumption is the linearity assumption which holds that the relationship between response variables and predictor variables is assumed to be linear with the regression parameter $\beta_0, \beta_1, \beta_2 ... \beta_p$. When the linearity assumption is violated, the transformation of the data can sometimes lead to linearity.

2. The second assumption is that the errors are independently and identically distributed(iid) normal variables each with mean zero and a constant variance. In other words, the error terms of the model must contain no trend for determining Y that is not already captured by the Xs.

3. The third assumption is about predictors: the predictor variable must be nonrandom, the values must be measured without error, and predictor variables must be independent of each other..

4. The fourth assumption is that all observations should serve an equally important role and be equally reliable when determining the results and conclusions[CH13].

For time series data, the second assumption that errors must be iid is often violated because the variables are correlated with themselves over time. We will then discuss the time series analysis. Then we use time series analysis as a tool to disentangle the autocorrelated properties with real-world data and to investigate the degree of severity of autocorrelated data with simulated data.

2.2 Time Series Analysis

2.2.1 The Sample Auto-correlation Function(ACF) and The Sample Partial Autocorrelation Function(PACF)

The sample autocorrelation function is an essential diagnostic tool for examining dependence in data. The sample autocorrelation function is defined as, r_k, at lag k, as shown in equation 3. The Sample Autocorrelation Function asses how the observations in a time series are related to each other, and is calculated by a simple correlation between the current observation Y_t and the observation k periods away, Y_k. The following equation illustrates the function.

$$r_k = Corr(Y_t, Y_{t-k}) = \frac{\sum_{t=k+1}^{n} (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^{n} (Y_t - \bar{Y})^2} \tag{3}$$

In general, we consider any $\mid r_k \mid \geq \frac{2}{\sqrt{n}}$ to be statistically significant[CC11].

The partial autocorrelation function is the autocorrelation between Y_t and Y_{t-k} given the effects of the variables in between t and $t-k$. The following equation illustrates the function[CC11].

$$\phi_{kk} = Corr(Y_t, Y_{t-k} \mid Y_{t-1}, Y_{t-2}, \ldots, Y_{t-k-1}) \tag{4}$$

We try to predict monthly pollution(PM2.5) in Beijing using the monthly temperature. The first approach is using a simple linear regression model to forecast. The details of the model are in Section 3. Figure 1 gives ACF for monthly averages of hourly temperature. Our null hypothesis is that the temperature is not autocorrelated. Because the black lines are much over the blue dash line at the majority of legs, we can reject our null hypothesis at those legs. The graph concludes that this temperature time series is significantly autocorrelated. In fact, we can see a strong seasonality trend. Figure 2 gives the PACF of the residuals of the regression model. The existence of significant autocorrelation patterns at lag 6 and 10 indicates that the residuals are serially dependent. The dependence of the errors is a violation of the second assumption of the Ordinary Least-Square Regression Model that the error terms must be iid. To predict the level of pollution with appropriate methods, we will introduce time series and time series modeling.

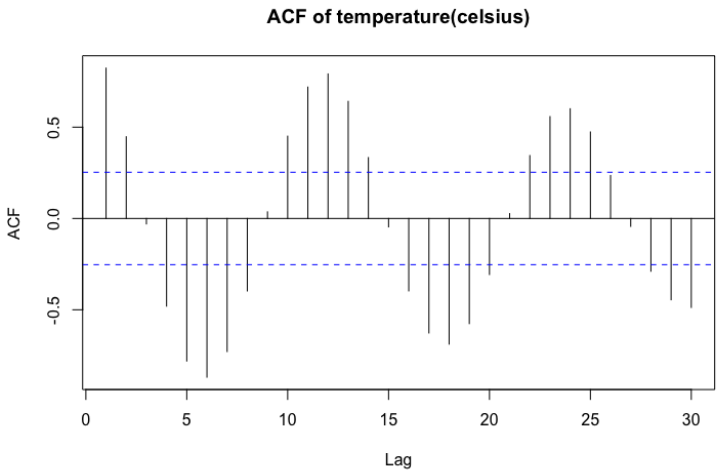

Figure 1: ACF of Temperature Time Series

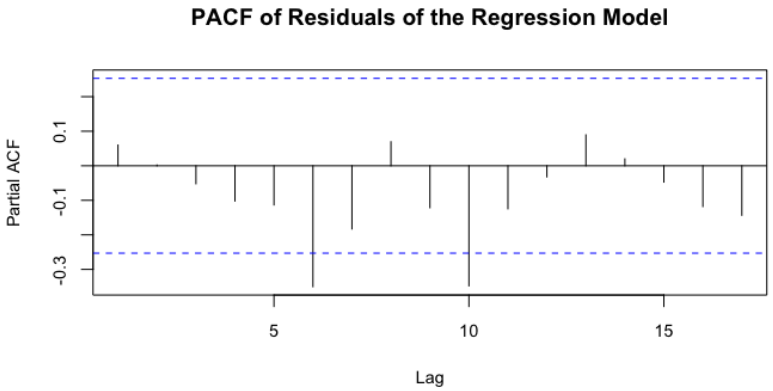

Figure 2: PACF of the Residuals of a Regression Model

2.2.2 Time Series Definition and the Stationary Assumption

Time series is a type of data obtained from observations that are collected chronologically over time. A stochastic process is a random sequence of variables $Y_t : t = 0, \pm 1, \pm 2, \pm 3$ changing with time. To make statistical extrapolations about the structure of a stochastic process on the foundation of an observed record of that process, we must make simplifying assumptions, including the most important stationary assumption.The definition of strictly stationary and weakly stationary is defined as

> "A process $\{Y_t\}$ is said to be strictly stationary if the joint distribution of $Y_{t_1}, Y_{t_2}, \ldots, Y_{t_n}$ is the same as the joint distribution of $Y_{t_1 k}, Y_{t_2 k}, \ldots, Y_{t_n k}$ for all choices of time points t_1, t_2, t_n and all choices of time lag k"[CC11].

A stochastic process $\{Y_t\}$ is weakly (or second-order) stationary if

$$\begin{cases} \text{the mean is constant over time} \\ \\ r_{t,t-k} = r_{0,k} \text{ , for all time t and lag } k \end{cases}$$

That is to say that the mean shall be constant regardless of time and co-variance must only depend on the distance between the two observations[CC11]. For the rest of this paper, the term stationary will always be referring to weakly stationary.

Stationary requires constant mean over time. When a time series contains a trend or trends, it is not stationary. Trends can be linear, quadratic, cyclic or seasonal. A seasonal time series is a time series that displays a very periodic pattern. In particular, seasonality for monthly value is the observed data regularly alter in the twelve months or other time intervals[CC11]. When a time series has a trend, it is not stationary and violates our assumption for time series analysis. We will talk about how to resolve this problem in Section 2.2.4. Figure 1 showed an example of a non-stationary seasonal time series.

2.2.3 The Cross-correlation Function(CCF)

The sample cross-correlation function is a useful tool for determining the degree of significance of cross-correlation.Let $Y = \{Y_t\}$ be time series of the response variable and $X = \{X_t\}$ be a covariate time series variable. For jointly stationary processes, the theoretical cross-correlation function(CCF) between X and Y at lag K is defined as

$$\rho_k(X, Y) = Corr(X_t, Y_{t-k}) = Corr(X_{t+k}, Y_t) \tag{5}$$

When Y = X, the cross-correlation is the same the the autocorrelation of Y at lag k. In practice, the sample cross-correlation function can be asses using the function[CC11].

$$r_k(X, Y) = \frac{\sum (X_t - \bar{X})(Y_{t-k} - \bar{Y})}{\sqrt{\sum (X_t - \bar{X})^2} \sqrt{\sum (Y_t - \bar{Y})^2}} \tag{6}$$

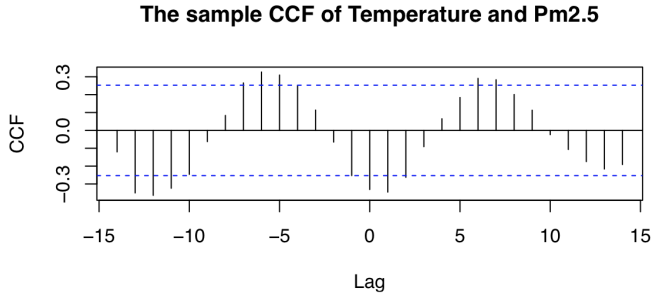

Figure 3: CCF of Temperature and PM2.5

Let us look at the example of CCF in Figure 3. Figure 3 displays the strong cross-correlation between temperature and PM2.5 at many lags. This nonstationarity in the two time series might cause non-independent errors and possibly lead to a spurious relationship.

2.2.4 Moving Average Processes, Autoregressive Processes and Differencing

A moving average model is a model such that Y_t depends only on the random error terms[CC11]. $1, -\theta_1, -\theta_2, ..., -\theta_q$ are the coefficients for the variable and the error terms. $e_t, e_{t-1}, e_{t-2}e_{t-q}$, are the variable. In other words, Y_t is a linear function of the p most recent error terms.The error term e_t are assumed to be white noise processes with zero mean and constant variance.

$$Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} \tag{7}$$

Autoregressive processes are regressions depends only on its past value $Y_{t-1}, Y_{t-2}, Y_{t-3}Y_{t-p}$. The book defines a p^{th}-order autoregressive process Y_t as:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ + \phi_p Y_{t-p} + e_t \tag{8}$$

Many real-life time series cannot be practically modeled by stationary processes such as the AR and MA because they have trends and are altering over time. We can eradicate or reduce the trend and seasonality by differencing. In this way, we can sometimes transform non-stationary time series into stationary time series. B is denoted for the backshift operator and was commonly used to express and manipulate ARIMA models. Especially, $BY_t = Y_{t-1}$. The first differencing can be defined in terms of B as

$$\Delta Y_t = Y_t - Y_{t-1} = Y_t B Y_t = (1 - B)Y_t \tag{9}$$

2.2.5 ARIMA(p,d,q) Model

The ARIMA model is a combination of Autoregressive (AR) Model in order p, differencing with order d and Moving Average(MA) Model with order q. The technique to determine the best ARIMA model firstly estimates the coefficient for p,d,q and check its adequacy. We can choose our model type approximately by the general behavior of the ACF and PACF. The sample partial autocorrelation function is used to asses whether an AR(p) model is correct. Cryer and Chan state in their book that "Quenoulle has shown, under the hypothesis that an AR(p) model is correct, the sample partial autocorrelation at lags greater than p are approximately normally distributed with zero means and variances 1/n. Thus, for $k > p$, $\pm 2/\sqrt{n}$ can be used as critical limits to test the null hypothesis that an AR(p) model is correct[CC11].

Table 1, extracted on page 116 in *Time series analysis with applications in R*, also shows the criteria to select proper ARIMA model using ACF and PACF. In model selection, we want to choose the model with the minimum AIC to avoid overfitting[CC11].

Table 1: The criteria to select the model using ACF and PACF

	AR(p)	MA(q)	ARMA(p,q)(p>0,q>0)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

2.2.6 Seasonal ARIMA Model

A seasonal AR(P)model of order P with seasonal period s is given as following.

$$Y_t = e_t + \Phi e_{t-s} + \Phi e_{t-2s} + \ldots + \Phi e_{t-Qs} \tag{10}$$

A seasonal MA(Q) model of order Q with seasonal period s is given as follows.

$$Y_t = e_t - \Theta_1 Y_{t-s} - \Theta_2 Y_{t-2s} - \ldots - \Theta_Q Y_{t-Qs} \tag{11}$$

The use of AR and MA explicitly assumes weak stationary. When a time series exhibits a seasonal trend, it is not stationary. However, we can resolve that by an essential and common tool, the seasonal difference. s denotes as a seasonal period. For example, s = 12 for monthly series. The seasonal difference of period s for the series Y_t is denoted $\nabla_s Y_t$ and is defined as following[CC11].

$$\nabla_s Y_t = Y_t - Y_{t-s} \tag{12}$$

2.2.7 Akaikes Information Criterion (AIC)

AIC stands for Akaikes Information Criterion and acts as a safeguard against overfitting. This criterion says to select the best model by minimizing AIC, where k = p + q + 1 if the model contains an intercept or constant term and k = p + q else. p is the p^{th} order of Autoregressive model and q is the q^{th} order of the Moving Average model. The equation of AIC is shown in the following equation. The likelihood function is the joint probability density of obtaining the data observed[CC11].

$$AIC = -2log(\text{Maximum Likelihood}) + 2k \tag{13}$$

The addition of the k serves as a punishment to help select the simplest models and avoid choosing a model with too many parameters. Maximum likelihood is a measure of the goodness of model fit. The higher value of maximum likelihood indicates a better model fit. The lower the AIC, the better the fit.

2.2.8 Prewhitening and Spurious Relationship

Spurious relationship happened when two variables exhibit a significant correlation without an underlying connection. For time series objects, there is a common tendency to be autocorrelated since what happened yesterday usually influences what will happen today. It is difficult to assess dependence between two strongly autocorrelated data sets because the degrees of freedom are related to the number of independent observations. Higher autocorrelation means a lower effective sample size. Also, when the independent and identically distributed assumption of residuals is violated, error rates are different from the specified significance level. In section 4, we will use simulated autocorrelated data to show the severe result of ignoring autocorrelation in time series objects with regression modeling.

A way to solve this dilemma is to disentangle the trend association between X and Y, from their autocorrelation. A useful strategy for disentangling is prewhitening the data with a suitable ARIMA model. In other words, we transformed the data to approximately random

process by replacing the data by the residuals from a fitted ARIMA model.$\widetilde{X}$ is defined as a white noise process which is sequentially uncorrelated, has zero mean and constant variance over time. Prewhitening is the process of transforming the X's to $\widetilde{X}$ via filer $\pi(B)$, and the same filter also applied to Y[CC11].

3 Example 1: Real World Data: Climate Change And Air Pollution

Spurious relationships readily occur in statistics. However, they are not often detected by society. Booming urbanization and industrialization among the world might plausibly raise many problems of air pollution, putting humans at risk of new health problems and encouraging people to pay attention to air pollution. Air pollution results from the combination of excessive emissions and hostile weather, just as air quality strongly relies on weather and therefore might be sensitive to climate change[JW09]. Pm2.5, known as fine particulate matter, is defined as particles or droplets in the air that have a diameter of 2.5 microns or less. The particulate matter is the major contributing factor to poor air quality and has been connected many times to increases in mortality [DPX$^+$93].

To asses the correlation between two time series objects, in this section, we focus on assessing the relationship between the concentration of particulate matter 2.5(PM2.5) and temperature. We will first use the common technique of ordinary least-square regression to model the level of particulate matter 2.5 and temperature. Then, we use time series modeling and prewhitening to diagnose the true relationship between them further.

The hourly data set, which includes the period between Jan 1st, 2010 to Dec 31st, 2014, contains the concentration of PM2.5 data and the temperature of Beijing. This data set downloaded from the UCI Machine Learning Repository. The website acknowledged Song Xi Chen as the source of the Beijing PM2.5 data set. Out of the 43824 hourly data of concentration of PM2.5, we have 2067 missing data, possibly due to a power outage, detection machine maintenance or unknown factors. To minimize the impact of missing data, we process the data from hourly data to monthly averages of hourly data before fitting any model.

Figure 4 shows the two time series plot the monthly average of hourly particular matter 2.5 concentration(ug/m^3) and the monthly average of hourly temperature in Celsius.

Table 2 shows the coefficients of the ordinary least-square regression modeling with response variable the monthly average of hourly PM2.5 concentration (ug/m^3) and explanatory variables the monthly average of hourly of temperature (Celsius). The intercept coefficients in this model predict the monthly average of hourly PM2.5 concentration (ug/m^3) when the monthly average of hourly temperature (Celsius) is equal to 0. The slope value represents the expected change in per ug/m^3 with per (Celsius) unit increase in the monthly average of hourly of temperature. The null hypothesis for this model is $H_o \neq 0$, which means that there is no significant relationship between PM2.5 and temperature. In this case, the very small p-value for both intercept and temperature indicates robust evidence again the null hypothesis, so we reject the null hypothesis. In other words, we have strong evidence that per unit increase in temperature decrease the level of pollution in term of the concentration or particular matter 2.5.

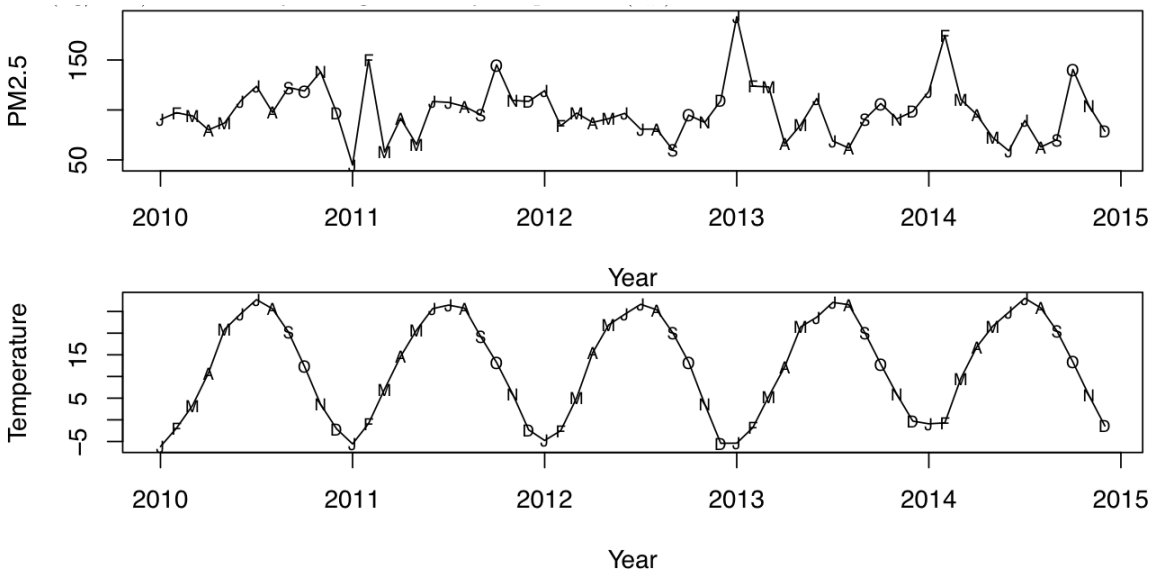

Figure 4: Time series plots of PM2.5 and Temperature

Table 2: Coefficients of regression summary

	Estimate	StandardError	Pvalue	Lower95	Upper95
Intercept	108.86	5.06	0.00	98.73	118.99
Temperature	-0.81	0.30	0.01	-1.42	-0.20

In Figure 5, the sample autocorrelation function for the standardized residuals to assess the possible dependence in the data set. All values are within the horizontal dashed lines except there seem to be a significant value at lag 6, which placed at zero plus and minus two approximate standard errors of the sample autocorrelation, namely $\pm 2/\sqrt{n}$. The significant value at lag 6 indicates a violation of the second assumption of regression modeling. The residual plot also shows that the error is almost randomly distributed with zero 0 but seems to show some seasonality, which might be a violation of our regression model. The histogram and qq-plot show that the residuals mostly follow a normal distribution with possibly a minor left-skew. Since the ACF and the residuals plot show possible violations of the iid assumption of residuals. Let us take a step further to look at the partial autocorrelation function of the residuals. The PACF of residuals of our regression model, shown in Figure 1, indicates the existence of significant autocorrelation patterns at lag 6 and 10, which shows the residuals are serially dependent. The dependency of error terms violates the iid assumption of residuals.

In summary, the correlation of temperature and PM2.5 is valid and significant when we are looking at the coefficients of regression summary and skim through the residuals plot, histogram, and qq-plot. However, the more nuanced analysis on the residuals shows possible violations to our regression assumption. When we look at PACF and ACF of the residuals,

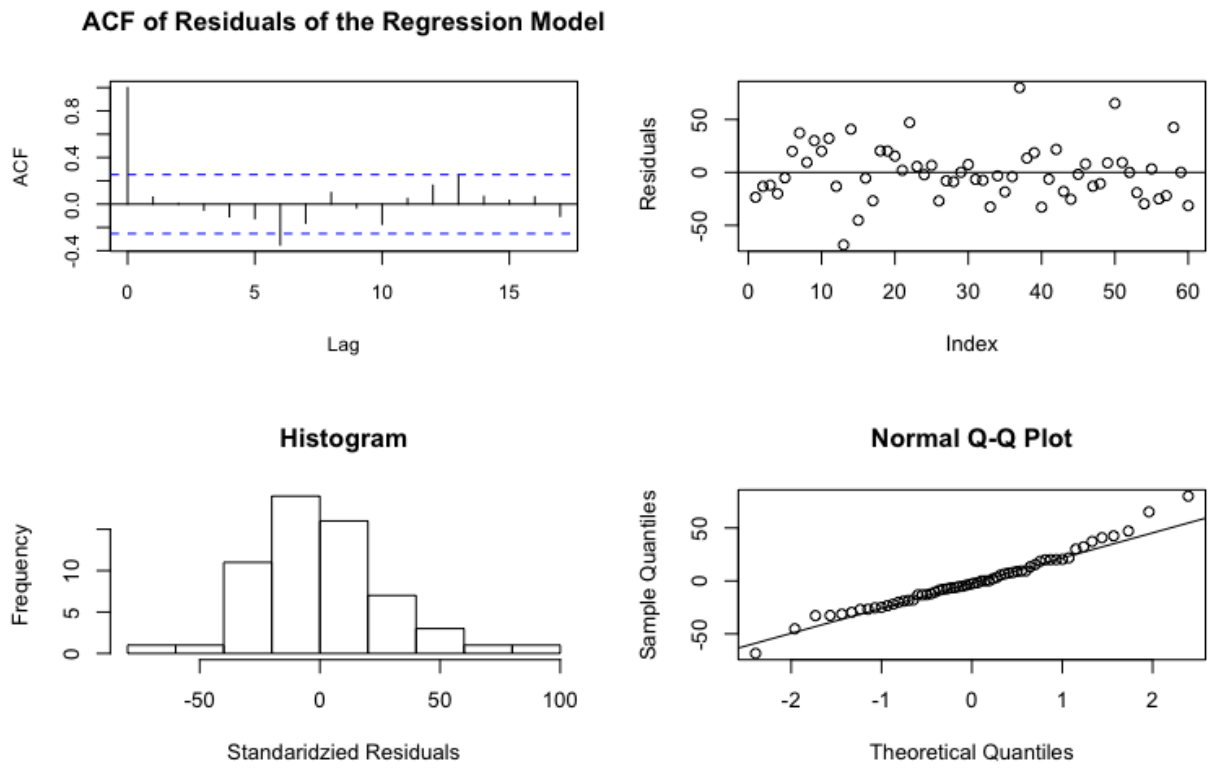

Figure 5: Residual Analysis

the regression assumption of residuals is violated. The residual plot also shows possible seasonality. We should not limit ourselves to coefficients of regression, residuals plot, qq-plot, and histogram when our data are time series. It is essential to look at ACF and PACF of residuals when doing residuals analysis for regression modeling. To find the actual relationship between these two variables, we will fit a seasonal ARIMA model to prewhite our time series data temperature and PM2.5.

In Figure 3, the sample CCF of Temperature and PM2.5, calculation shows that these series have a significant cross-correlation coefficient at lag 0,1,6 and 7 that is statistically significantly different from zero. However, it is possible that the seasonal trends that found in both temperature and PM2.5 time series cause spurious correlations. Since it is difficult to validate dependence between two strongly autocorrelated, we will prewhitening our data set by prewhitening our data set with the best fit seasonal ARIMA model of temperature by looking at the ACF and PACF and minimizing the AIC.

Figure 1 shows the sample autocorrelation of the monthly averages of hourly temperature. We see strong seasonality and strong autocorrelation in it. To meet the assumption of stationary and do ARIMA modeling, we will use differencing to remove the trends.

Figure 6 shows the sample autocorrelations of the monthly averages of hourly temperature after three different differencing. The first graph shows the ACF for temperature after the 1^{st}order of differencing. Almost all lags are outside of the significant dash line which still violating the assumption of independence. For the ACF for temperature after the 1^{st} order

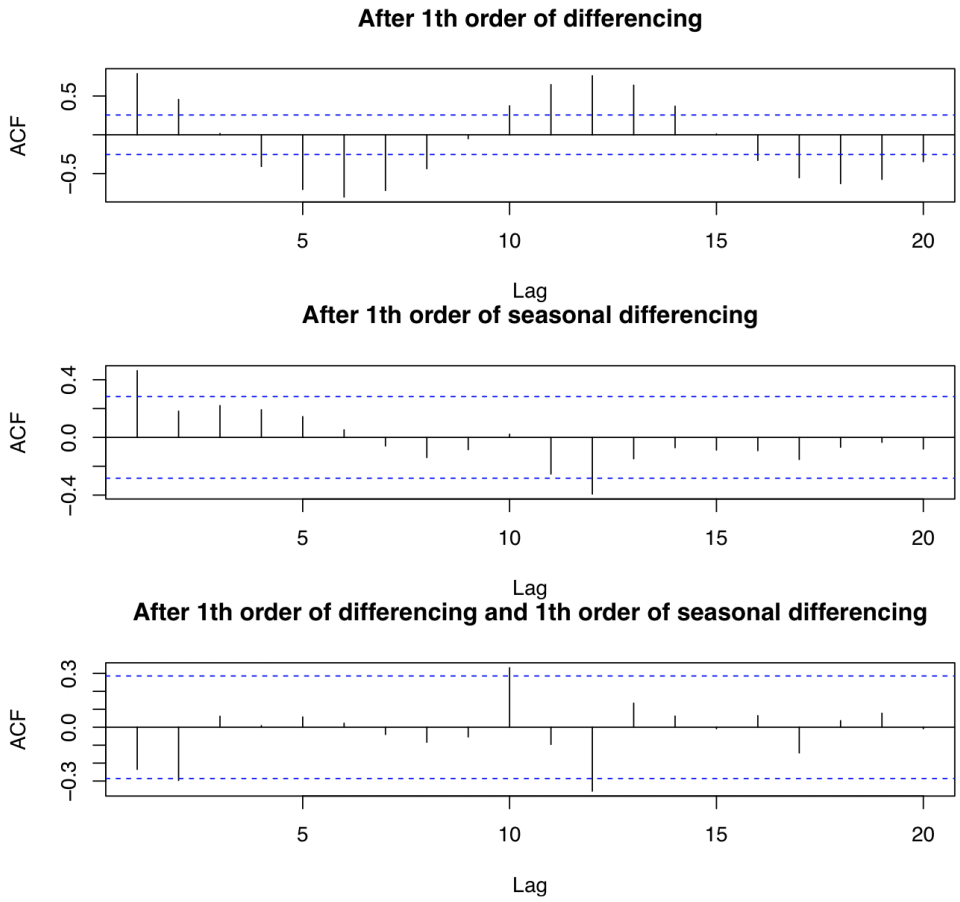

Figure 6: The the sample autocorrelation of temperature after there different differencing

of seasonal differencing, we can see that lag 1 and lag 12 are significant. Lag 1 to lag 12 still shows some autocorrelation. Since almost all value of lag 2 to lag 11 are within the dashed line of significant, the 1^{st} order of seasonal differencing is acceptable. For the ACF after the 1^{st} order of differencing and the 1^{st} order of seasonal differencing, lag 2, 10 and 12 are significant. We can barely recognize any autocorrelation. This type of differencing is also satisfactory. There is not much of improving by adding a simple differencing to a seasonal differencing. To avoid over differencing, it is appropriate and reasonable to choose the 1^{st} order of seasonal differencing.

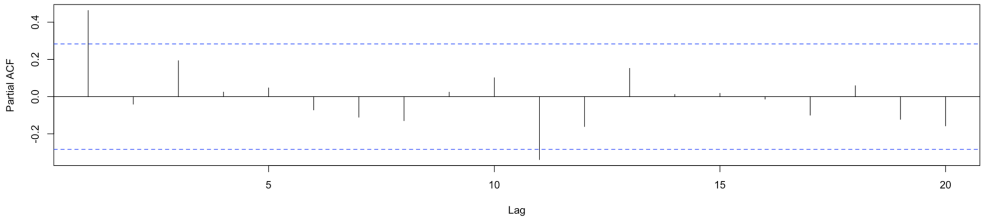

Figure 7: PACF for temperature after 1th order of seasonal differencing

Figure 7 shows PACF for temperature after 1^{st} order of seasonal differencing. ACF tails

off gradually to 0 while PACF cuts off. According to Figure 8, an AR(p) model should be select.

Table 3 shows the AIC for 24 different ARIMA models. AIC is defined in section 2.2.5 and is stand for Akailkes Infromation Criterion to estimate the model fit. The lower the AIC, the better the fit.

Table 3: The AIC for different ARIMA models

	ARIMA(0,0,1)	ARIMA(1,0,0)	ARIMA(1,0,1)	ARIMA(0,1,1)	ARIMA(1,1,0)	ARIMA(1,1,1)
Seasonal(0,1,0)	188.00	187.10	189.05	189.05	194.45	187.72
Seasonal(0,1,1)	180.34	177.29	178.29	176.19	182.01	176.86
Seasonal(1,1,0)	181.78	179.21	180.69	179.66	185.10	179.82
Seasonal(1,1,1)	181.50	178.42	179.26	177.20	183.06	178.00

The first order of seasonal differencing is good enough in this case. By selecting the lowest AIC with first order of seasonal differencing, we get $ARIMA(1,0,0) \times Seasonal(0,1,1)$. Figure 13 and Figure 14 show the ACF and PACF for temperature after the first order of seasonal differencing. ACF tails off gradually to 0 while PACF cuts off. According to figure 8, an AR(p) model should be selected. So, this model selection makes sense, and we will use this model to prewhiten our data.

Figure 8 shows the CCF for temperature and concentration of PM2.5 after prewhitening. The calculation indicates that these series do not correlate with lag 0. When calculating the CCF, we are also conducting many tests simultaneously, so it is reasonable to ignore small correlations. Thus, it seems that the monthly average of hourly temperature and the monthly averages of the concentration of hourly particular matter 2.5 are mostly uncorrelated, and the strong cross-correlation pattern found between the raw data series might be spurious.

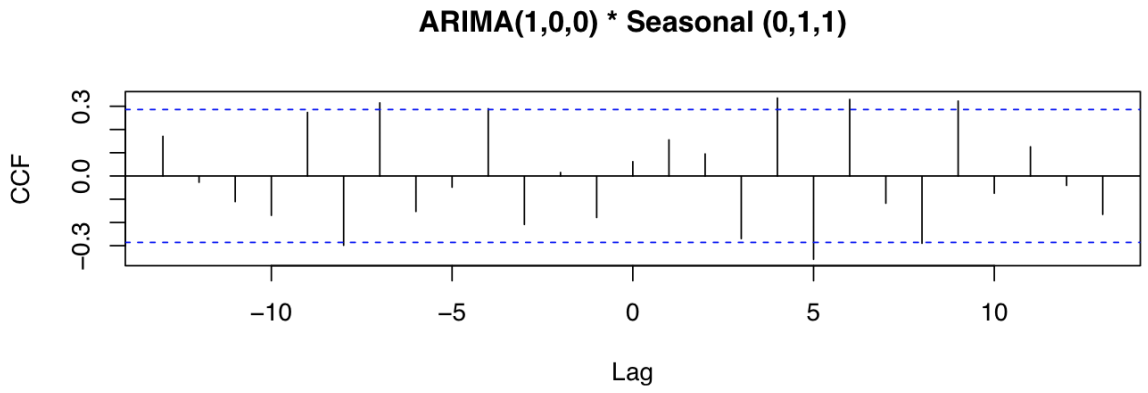

Figure 8: The sample CCF of PM2.5 and temperature

4 Example 2: Simulated Data

For statistical hypothesis testing, it is always possible to make systematic mistakes. While there is no way to eliminate receiving errors, we can always do what we can to control the rate of errors at the expected level. In general, there are two types of errors, TYPE I error and TYPE II error. TYPE I error is rejecting a true null hypothesis while the TYPE II error is failing to reject a false hypothesis. The goal of this section is to control the rate of Type I error. TYPE I specified by the significance level (alpha), and if all assumptions are met then the type I error rate should be equal to alpha.

Table 4: TYPE I and TYPE II error in statistic

	When Ho is actually True	When Ho is actually Flase: Siginificant Relationship
Accepted Ho	Correct Decision	Type 2 Error: Accepted a False Hypothesis
Rejected Ho	Type 1 Error: Reject True Hypothesis	Correct Decision

In section 2.2.8, we mentioned that prewhitening is a powerful tool to detect a potential spurious relationship. In this example, we will use the ARIMA model simulation in R to generate correlated and uncorrelated time series. Then, we test the rate of TYPE I error and TYPE II error before prewhitening and after prewhitening with different sample sizes and sample standard deviations. The codes are obtained from Dr. Maureen Kennedy (mkenn@uw.edu). We will demonstrate that the rate of TYPE I error can be reduced immensely by the technique of prewhitening and be corrected to the significant level. The codes for the process is in Appendix B. The simulation process is explained in next paragraph and also in Figure 9.

For each simulation, we use the R function arima.sim to simulate two time series with size n and standard deviation d, namely X and Y. X and Y should be two uncorrelated time series since they generate randomly by the arima.sim function. Afterward, we perform hypothesis testing between X and Y with a 0.05 significance level. Since X and Y are uncorrelated, we should see a type 1 error at a rate of 0.05. If H_0 is rejected, we save 1 in an array, save 0 otherwise. Second, we create Y_2 and let $Y_2 = 0.5X + 10 + \varepsilon$, where ε is independent and identically distributed error with zero mean and standard deviation d. X and Y_2 should be closed correlated. the same time series. Afterward, we perform hypothesis testing between X and Y_2 with 0.05 significant level. In this case, rejecting null hypothesis is a correct decision, and the error rate of interest is the type 2 error. The rate at which the correct decision is made is the statistical power of the test. If H_0 is rejected, we save 1 in an array; we save 0 otherwise. Thirdly, we prewhiten the data and repeat hypothesis testing, X and Y and X and Y_2, saving the results in arrays. Finally, we repeat the simulation 500 times. Afterward, we summed each array and divided them by 500. In this way, we get the rate of TYPE I error and power before and after prewhitening. We can easily calculate the rate of TYPE II error by using 1-Power. We repeat this process over and over again at increasing sample size (n) and increasing the standard deviation(sd). The results are displayed in Table 5 and Table 6.

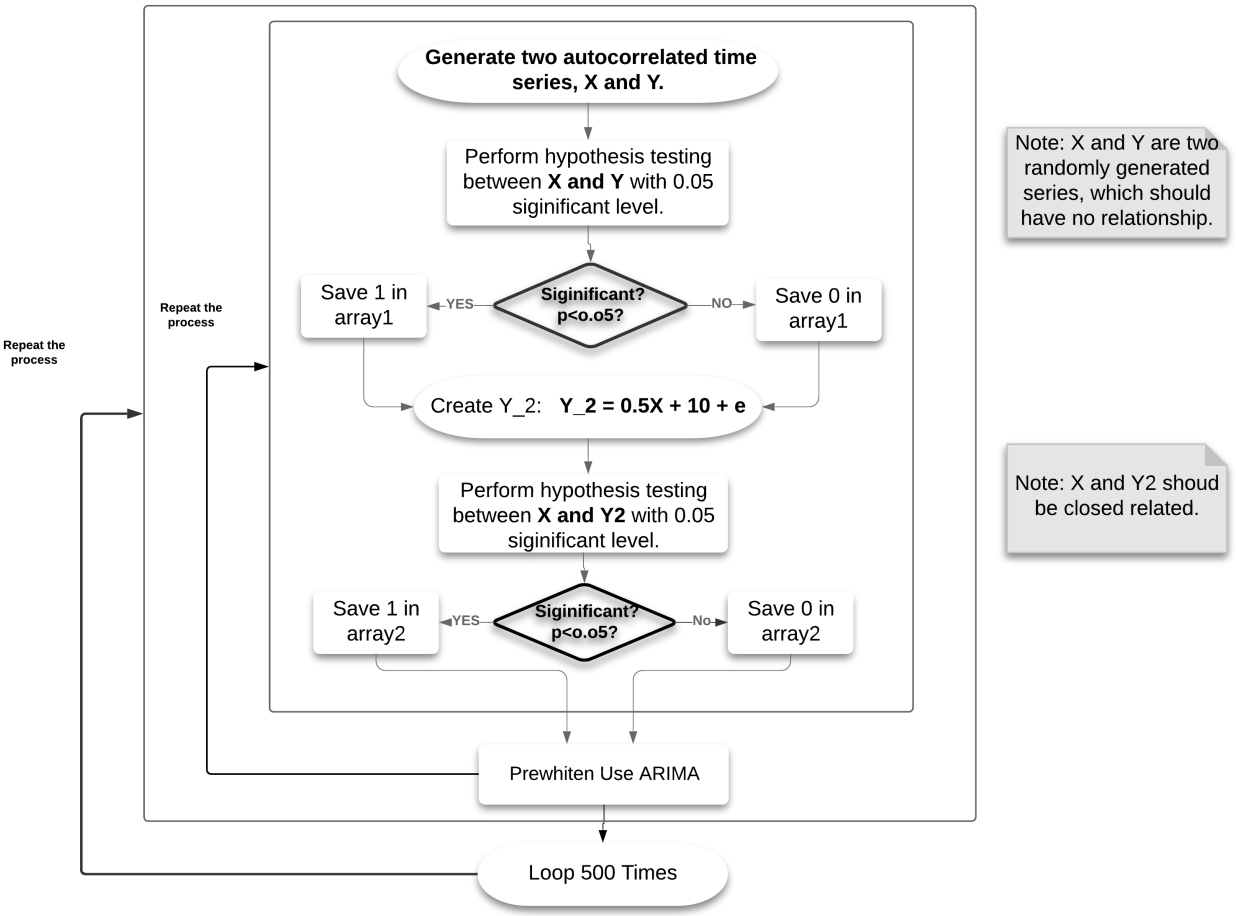

Figure 9: Simulation Process

Table 5: The rate of TYPE I and TYPE II error before and after prewhitening with different sample size

	n=30	n=60	n=300	n=500	n=1000
TYPE I Error Rate	0.336	0.406	0.466	0.478	0.468
TYPE II Error Rate	0.108	0.002	0.000	0.000	0.000
TYPE I Error Rate after Prewhitening	0.044	0.050	0.048	0.044	0.054
TYPE II Error Rate after prewhitening	0.670	0.378	0.000	0.000	0.000

In the result of Table, our rate of type 1 error is much higher than the specified significance level Before pre-whitening. The simulated data demonstrate that all the rate of Type I error became close to our specify significance level, 0.05, after prewhitening. However, the rate of Type II error increases after prewhitening. As the sample size increases, the rate of TYPE II error decreases. This example shows that by using the prewhitening technique, we can effectively reduce the chance of making TYPE I error and make a more reliable decision base on the existed data set at the sacrifice of statistical power. By prewhitening, we detect spurious relationships at the correct rate(TYPE I error), but we are then more likely to miss any truly significant relationships.

Table 6: The rate of TYPE I and TYPE II error before and after prewhitening with different standard deviations

	sd = 0.0001	sd=0.001	sd=0.01	sd=0.1	sd=0.2	sd=1
TYPE I Error Rate	0.39	0.46	0.41	0.37	0.46	0.41
TYPE II Error Rate	0.00	0.00	0.00	0.01	0.19	0.94
TYPE I Error Rate after Prewhitening	0.04	0.06	0.08	0.06	0.07	0.05
TYPE II Error Rate after prewhitening	0.50	0.51	0.53	0.59	0.67	0.92

In Table 6, simulated data demonstrate that all the rate of TYPE I error reduces dramatically after prewhitening. However, the rate of Type II error increases after prewhitening. As the standard deviations rising, the rate of TYPE II error is increasing. This example shows that by using the prewhitening technique, we can effectively correct the chance of making TYPE I error to our significant level, 0.05, and make a more reliable decision base on the existed data set.

5 Discussion

In conclusion, it is complicated to determine the true relationship of two time series that are themselves autocorrelated because the degrees of freedom is related to the number of independent observations, especially with a small sample size. When all assumptions of regression are satisfied, the probability of making the wrong decision when the null hypothesis is correct should be at most 5 percent, our assumed significant level. In particular, from Table 5 and Table 6 in Simulated Data, we can see the TYPE I error rates for simulated data are between 33 percent and 48 percent, which is way beyond our expected 5 percent significance level. Any prediction with this level of TYPE I error rates is not useful for future forecasting. One way to solve this dilemma is to use the prewhitening technique in time series analysis.

The method of prewhitening can effectively correct the rate of TYPE I errors to a specified significant level rate at the expense of increasing the TYPE II error rate. Type I errors happen when we determine a correlation is significant when in reality it is not. Type II errors occur when we fail to identify a significance relationship when it actually exists. In other words, we correct the rate of making wrong predictions with an increasing rate of ignoring true

relationships. One of the ultimate goals for statistics is to find the correlation between two variables. The trade-off that we made in prewhitening with TYPE I error to TYPE II error is valuable because it is better to miss a significant relationship than found a relationship with a high probability of false positive. By doing so, we gain more control of error rate in terms of predicting and forecasting.

However, the application of prewhitening also has many limitations. One primary limitation is the striking increased in TYPE II errors. TYPE II errors depend mainly on sample size and population variance. In the case of a larger sample size, this limitation will be automatically avoided. In the case of a smaller sample size, we will have a much higher chance of committing TYPE II errors. In other words, it becomes very likely for us to ignore the actual relationship between the time series data. In Table 5, we say the rate of TYPE II error after prewhitening is 0.67 for sample size 30 and 0.378 for sample size 60. Any conclusion with this level of TYPE II error is not convincing. We are back in a quandary again.

One way to solve the plight is to choose a larger sample size and correctly use the technique of prewhitening. Correlation does not mean causation. To further diagnose time series and obtain significant information from them, we need to learn more modeling techniques, such as financial time series and spectral analysis. However, from this paper, we learn that the use of Ordinary-Least Regression for autocorrelated time series objects is often doubtful due to the fact that the second assumption of regression often violated. The reason is the degrees of freedom is related to the number of independent observations. We can effectively correct our likelihood of having a spurious relationship to the anticipated significant level for some time series data using Time Series Modeling and Prewhitening with large sample sizes.

6 Appendix A: R codes for Section 3

```
################
# Analysis for the relationship between Temperature
# using regression and time series analysis.
# Author: Yang Tang(tangy32@uw.edu)
################

# clear encironment
rm(list=ls())
#install TSA
library(TSA)
# Import the data
raw_data <- read.csv("PRSA_data_2010.1.1-2014.12.31.csv")
# Declare a new variable every time we do this loop
monthly_data <- c()
## Using for loop to calculate the monthly mean for
# pm2.5 and temperaturefor each month.
#For each year in 2010 to 2014
for (i in 2010:2014){
  # For each month
  for(j in 1:12){
    monthly_data$pm2.5 <- append(monthly_data$pm,mean(raw_data$pm2.5
      [which(raw_data$month == j & raw_data$year == i)],
      na.rm = TRUE))
    monthly_data$temperature <- append(monthly_data$temperature,mean(
      raw_data$TEMP[which(raw_data$month == j & raw_data$year == i)],
      na.rm = TRUE))
    monthly_data$year <- append(monthly_data$year, i)
    monthly_data$month <- append(monthly_data$month, j)
  }
}

# Put the data into a data frame
monthly_data.df<-data.frame(pm2.5=monthly_data$pm,
                            temperature=monthly_data$temperature,
                            year=monthly_data$year,
                            month=monthly_data$month)

# Declear them as Time-Series Variables.
# It's the exactly same data with the dataframe. For different purpose in terms of graphing.
```

```
monthly_ts <- ts(monthly_data.df,start=c(2010,1), end=c(2014,12),frequency=12)
pm_ts <- ts(monthly_data$pm, start=c(2010,1), end=c(2014,12),frequency=12)
temperature_ts <- ts(monthly_data$temperature, start=c(2010,1), end=c(2014,12),frequency=12)

# Plot monthly temperature and PM2.5 as time series.
par(mfrow=c(1,1))
plot(pm_ts,ylab ="PM2.5",xlab = "Year",type = "l")
points(pm_ts, x = time(pm_ts), pch = as.vector(season(pm_ts)),cex = 0.7)
plot(temperature_ts,ylab ="Temperature",xlab = "Year",type = "l")
points(temperature_ts, x = time(temperature_ts), pch = as.vector(season(temperature_ts)),cex = 0.7)

# Use lm function in R to fit a simple linear regression model
# pm2.5 is the response varible, temperature is the predicator variable.
pm.lm <- lm(pm2.5~temperature,data = monthly_data)

# Create a table to display the result for the previous linear model.
pm.ci  = confint(pm.lm)
summary.tab = data.frame(Estimate = round(pm.lm$coefficients,2),
                         StandardError = round(summary(pm.lm)$coefficients[,2],2),
                         Pvalue = round(summary(pm.lm)$coefficients[,4],2),
                         Lower95 = round(pm.ci[,1],2),
                         Upper95 = round(pm.ci[,2],2))
row.names(summary.tab) = c("Intercept","Temperature")
col.names = c("Estimate","Standard Error","P-value",
              "Lower 95% Confidence Bound",
              "Upper 95% ConfidenceBound")
knitr::kable(summary.tab)

# Perform residual analysis
par(mfrow=c(2,2))
acf(pm.lm$residuals,main = "ACF of Residuals of the Regression Model",cex.lab=0.9, cex.axis=0.9, cex.main=0.9, cex.sub=0.9)
plot(pm.lm$residuals, ylab = " Residuals")
abline(h=0)
hist(pm.lm$residuals,
     xlab = "Standaridzied Residuals", main = "Histogram")
qqnorm(pm.lm$residuals)
qqline(pm.lm$residuals)

# Plot the PACF graph
par(mfrow=c(1,1))
pacf(pm.lm$residuals,main = "PACF of Residuals of the Regression Model",
     cex.lab=0.9, cex.axis=0.9, cex.main=0.9, cex.sub=0.9)

# Plot the CCF for temperature and PM2.5
tp=ts.intersect(temperature_ts,pm_ts)
ccf(as.numeric(tp[,1]),as.numeric(tp[,2]),
    main='The sample CCF of Temperature and Pm2.5',ylab='CCF')

# Sample ACF and PACF of temperature.
par(mfrow=c(1,1))
acf(as.vector(temperature_ts),lag.max = 30, main = "ACF of temperature(celsius)")

# Generate ACF graph after different type of differencing
tp.dif1=ts.intersect(diff(temperature_ts),diff(pm_ts))
tp.dif2=ts.intersect(diff(temperature_ts,12),diff(pm_ts,12))
tp.dif3=ts.intersect(diff(diff(temperature_ts,12)),diff(diff(pm_ts,12)))
acf(as.vector(tp.dif1[,1]),lag.max = 20, main = "After 1th order of differencing")
acf(as.vector(tp.dif2[,1]),lag.max = 20, main = "After 1th order of seasonal differencing")
acf(as.vector(tp.dif3[,1]),lag.max = 20, main = "After 1th order of differencing and 1th order of seasonal differencing")

# PACF after 1th order of seasonal differencing
par(mfrow=c(1,2))
pacf(as.vector(tp.dif2[,1]),lag.max = 20, main = "PACF: After 1th order of seasonal differencing")


############Create the table of AIC for different sesonal ARIMA model ##################
# The lower the AIC, the better.
# ARIMA (0,0,1)
mo.temp_001_010 = arima(monthly_ts[,2], order = c(0,0,1),
                        seasonal = list(order=c(0,1,0),period=12))
mo.temp_001_011 = arima(monthly_ts[,2], order = c(0,0,1),
                        seasonal = list(order=c(0,1,1),period=12))
mo.temp_001_110 = arima(monthly_ts[,2], order = c(0,0,1),
                        seasonal = list(order=c(1,1,0),period=12))
mo.temp_001_111 = arima(monthly_ts[,2], order = c(0,0,1),
                        seasonal = list(order=c(1,1,1),period=12))
aic001 = c(mo.temp_001_010$aic,mo.temp_001_011$aic, mo.temp_001_110$aic,mo.temp_001_111$aic)


# ARIMA (1,0,0)
mo.temp_100_010 = arima(monthly_ts[,2], order = c(1,0,0),
                        seasonal = list(order=c(0,1,0),period=12))
mo.temp_100_011 = arima(monthly_ts[,2], order = c(1,0,0),
                        seasonal = list(order=c(0,1,1),period=12))
mo.temp_100_110 = arima(monthly_ts[,2], order = c(1,0,0),
                        seasonal = list(order=c(1,1,0),period=12))
mo.temp_100_111 = arima(monthly_ts[,2], order = c(1,0,0),
                        seasonal = list(order=c(1,1,1),period=12))
aic100 = c(mo.temp_100_010$aic, mo.temp_100_011$aic, mo.temp_100_110$aic,mo.temp_100_111$aic)

# ARIMA (1,0,1)
mo.temp_101_010 = arima(monthly_ts[,2], order = c(1,0,1),
```

```
                          seasonal = list(order=c(0,1,0),period=12))
mo.temp_101_011 = arima(monthly_ts[,2], order = c(1,0,1),
                          seasonal = list(order=c(0,1,1),period=12))
mo.temp_101_110 = arima(monthly_ts[,2], order = c(1,0,1),
                          seasonal = list(order=c(1,1,0),period=12))
mo.temp_101_111 = arima(monthly_ts[,2], order = c(1,0,1),
                          seasonal = list(order=c(1,1,1),period=12))
aic101 = c(mo.temp_101_010$aic,mo.temp_101_011$aic, mo.temp_101_110$aic,mo.temp_101_111$aic)

# ARIMA (0,1,1)
mo.temp_011_010 = arima(monthly_ts[,2], order = c(0,1,1),
                          seasonal = list(order=c(0,1,0),period=12))
mo.temp_011_011 = arima(monthly_ts[,2], order = c(0,1,1),
                          seasonal = list(order=c(0,1,1),period=12))
mo.temp_011_110 = arima(monthly_ts[,2], order = c(0,1,1),
                          seasonal = list(order=c(1,1,0),period=12))
mo.temp_011_111 = arima(monthly_ts[,2], order = c(0,1,1),
                          seasonal = list(order=c(1,1,1),period=12))
aic011 = c(mo.temp_011_010$aic, mo.temp_011_011$aic, mo.temp_011_110$aic,mo.temp_011_111$aic)

# ARIMA (1,1,0)
mo.temp_110_010 = arima(monthly_ts[,2], order = c(1,1,0),
                          seasonal = list(order=c(0,1,0),period=12))
mo.temp_110_011 = arima(monthly_ts[,2], order = c(1,1,0),
                          seasonal = list(order=c(0,1,1),period=12))
mo.temp_110_110 = arima(monthly_ts[,2], order = c(1,1,0),
                          seasonal = list(order=c(1,1,0),period=12))
mo.temp_110_111 = arima(monthly_ts[,2], order = c(1,1,0),
                          seasonal = list(order=c(1,1,1),period=12))
aic110 = c(mo.temp_110_010$aic, mo.temp_110_011$aic, mo.temp_110_110$aic,mo.temp_110_111$aic)

# ARIMA (1,1,1)
mo.temp_111_010 = arima(monthly_ts[,2], order = c(1,1,1),
                          seasonal = list(order=c(0,1,0),period=12))
mo.temp_111_011 = arima(monthly_ts[,2], order = c(1,1,1),
                          seasonal = list(order=c(0,1,1),period=12))
mo.temp_111_110 = arima(monthly_ts[,2], order = c(1,1,1),
                          seasonal = list(order=c(1,1,0),period=12))
mo.temp_111_111 = arima(monthly_ts[,2], order = c(1,1,1),
                          seasonal = list(order=c(1,1,1),period=12))
aic111 = c(mo.temp_111_010$aic, mo.temp_111_011$aic, mo.temp_111_110$aic,mo.temp_111_111$aic)


mo.df<-data.frame(aic001,aic100,aic101,aic011,aic110,aic111)
row.names(mo.df) = c("Seasonal(0,1,0)", "Seasonal(0,1,1)","Seasonal(1,1,0)","Seasonal(1,1,1)")
colnames(mo.df) = c("ARIMA(0,0,1)","ARIMA(1,0,0)","ARIMA(1,0,1)",
                    "ARIMA(0,1,1)","ARIMA(1,1,0)","ARIMA(1,1,1)")

knitr::kable(round(mo.df,digits=2))

#################################################################


tp.dif=ts.intersect(diff(diff(temperature_ts,12)),diff(diff(pm_ts,12)))
tp.dif2=ts.intersect(diff(temperature_ts,12),diff(pm_ts,12))

# Use the build in prewhiten function in R with the best model we select.
# We perform the prewhitening with seasonal ARIMA model (1,0,0) * (0,1,1).
prewhiten(as.numeric(tp.dif2[,1]),as.numeric(tp.dif2[,2]),
          x.model = arima(monthly_ts[,2], order = c(1,0,0),seasonal =
          list(order=c(0,0,1),period=12)),main = "ARIMA(1,0,0) * Seasonal (0,1,1)")
```

7 Appendix B: R codes for Section 4

```
################
# simulate ARIMA and test ccf
# Author: Dr. Maureen Kennedy (mkenn@uw.edu )
# and Yang Tang(tangy32@uw.edu)
################

rm(list=ls()) # clear environment
library(TSA) # import the TSA library

nsim=100 # repeat nsim times

#Sample Size
sd = 0.1
n_sequence <- c(30,60,300,500,1000)
sim_data_n <- c()
sim_data_n$type1.error <- rep(0,5)
sim_data_n$type2.error <- rep(0,5)
sim_data_n$type1.error.after <- rep(0,5)
sim_data_n$type2.error.after <- rep(0,5)

for (k in  1:5){
  # each of this is a flag that takes
  # a value 1 if the null is rejected, 0 otherwise
   reject.h0<-rep(0,nsim) # sample
   reject2.h0<-rep(0,nsim) # true relationship
```

```
    reject3.h0<-rep(0,nsim)
    reject4.h0<-rep(0,nsim)

    #This set the standard deviation
    ts_size = n_sequence[k]

    for(i in 1:nsim) {

      tmp.simx<-arima.sim(n=ts_size,list(ar=0.89,ma=-0.25),sd=0.1)
      # one arima(1,1,0) process, of length 60
      tmp.simy<-arima.sim(n=ts_size,list(ar=0.89,ma=-0.25),sd=0.1)
      # a second, independent arima(1,1,0) process, of length 50
      tmp.cor<-cor.test(tmp.simx,tmp.simy)
      # test a simple correlation between these two

      # tmp.simx and temp.simy are two different time series,
      # which should have no relationship.
      # We should accpted the NULL Hypothesis.
      # When reject the True NULL hypothesis, we make TYPE 1 error.
      if(tmp.cor$p.value<=0.05) # is it significant?
        reject.h0[i]<-1

      # tmp.2simy is set to be the same time series with by
      # times 0.5 and adding some random errors.
      # We should reject the NULL hypothesis.
      # When reject the NULL hypothesis, we made good decision.
      # When fail to reject the NULL pothesis,
      # we made TYPE 2 error (equal to 1-POWER).
      tmp2.simy<-tmp.simx*0.5+10+rnorm(ts_size,0,0.1)
      tmp2.cor<-cor.test(tmp.simx,tmp2.simy)
      if(tmp2.cor$p.value<=0.05)
        reject2.h0[i]<-1

      # We prewhiten our data to find out what is the impact of
      # prewhitening in terms of TYPE 1 and TYPE2 error

      # reject3.h0 is the rate of TYPE 1 error after prewhitening
      tmp.pre1<-prewhiten(tmp.simx,tmp.simy,x.model =
                            arima(tmp.simx, order = c(1,0,1)),plot=FALSE)
      if(abs(tmp.pre1$ccf[0]$acf)>1.96/sqrt(ts_size))
        reject3.h0[i]<-1

      # reject4.h0 is the power after prewhitening
      # 1-POWER is the rate of TYPE 2 error
      tmp.pre2<-prewhiten(tmp.simx,tmp2.simy,x.model =
                            arima(tmp.simx, order = c(1,0,1)),plot=FALSE)
      if(abs(tmp.pre2$ccf[0]$acf)>1.96/sqrt(ts_size))
        reject4.h0[i]<-1
    }
    sim_data_n$type1.error[k] <- sum(reject.h0)/nsim
    sim_data_n$type2.error[k] <- (1-sum(reject2.h0)/nsim)
    sim_data_n$type1.error.after[k] <- sum(reject3.h0)/nsim
    sim_data_n$type2.error.after[k] <- (1-sum(reject4.h0)/nsim)

  }

  ############################################################
  # Standard deviation
  ts_size = 60
  sd_sequence <- c(0.0001, 0.001, 0.01, 0.1, 0.2, 1 )
  sim_data_sd <- c()
  sim_data_sd$type1.error <- rep(0,6)
  sim_data_sd$type2.error <- rep(0,6)
  sim_data_sd$type1.error.after <- rep(0,6)
  sim_data_sd$type2.error.after <- rep(0,6)


  for (k in  1:6){
    # each of this is a flag that takes
    # a value 1 if the null is rejected, 0 otherwise
    reject.h0<-rep(0,nsim) # sample
    reject2.h0<-rep(0,nsim) # true relationship
    reject3.h0<-rep(0,nsim)
    reject4.h0<-rep(0,nsim)

    # This sets the standard deviation.
    sd_ = sd_sequence[k]

    for(i in 1:nsim) {

      tmp.simx<-arima.sim(n=ts_size,list(ar=0.89,ma=-0.25),sd = 0.1)
      # one arima(1,1,0) process, of length 60
      tmp.simy<-arima.sim(n=ts_size,list(ar=0.89,ma=-0.25),sd = 0.1)
      # a second, independent arima(1,1,0) process, of length 50
      tmp.cor<-cor.test(tmp.simx,tmp.simy)
      # test a simple correlation between these two

      # tmp.simx and temp.simy are two different time series,
      # which should have no relationship.
      # We should accpted the NULL Hypothesis.
      # When reject the True NULL hypothesis, we make TYPE 1 error.
      if(tmp.cor$p.value<=0.05) # is it significant?
```

```
118         reject.h0[i]<-1
119
120       # tmp.2simy is set to be the same time series with by
121       # times 0.5 and adding some random errors.
122       # We should reject the NULL hypothesis.
123       # When reject the NULL hypothesis, we made good decision.
124       # When fail to reject the NULL pothesis,
125       # we made TYPE 2 error (equal to 1-POWER).
126       tmp2.simy<-tmp.simx*0.5+10+rnorm(ts_size,0,sd_)
127       tmp2.cor<-cor.test(tmp.simx,tmp2.simy)
128       if(tmp2.cor$p.value<=0.05)
129         reject2.h0[i]<-1
130
131       # We prewhiten our data to find out what is the impact of
132       # prewhitening in terms of TYPE 1 and TYPE2 error.
133
134      # reject3.h0 is the rate of TYPE 1 error after prewhitening
135       tmp.pre1<-prewhiten(tmp.simx,tmp.simy,
136                           x.model = arima(tmp.simx, order = c(1,0,1)),
137                           method="ML",plot=FALSE)
138       if(abs(tmp.pre1$ccf[0]$acf)>1.96/sqrt(ts_size))
139         reject3.h0[i]<-1
140
141       # reject4.h0 is the power after prewhitening
142       # 1-POWER is the rate of TYPE 2 error
143       tmp.pre2<-prewhiten(tmp.simx,tmp2.simy,
144                           x.model = arima(tmp.simx, order = c(1,0,1)),
145                           method="ML",plot=FALSE)
146       if(abs(tmp.pre2$ccf[0]$acf)>1.96/sqrt(ts_size))
147         reject4.h0[i]<-1
148     }
149     sim_data_sd$type1.error[k] <- sum(reject.h0)/nsim
150     sim_data_sd$type2.error[k] <- (1-sum(reject2.h0)/nsim)
151     sim_data_sd$type1.error.after[k] <- sum(reject3.h0)/nsim
152     sim_data_sd$type2.error.after[k] <- (1-sum(reject4.h0)/nsim)
153
154   }
```

References

[Box15] George E. P. Box. *Time series analysis: forecasting and control*. John Wiley Sons, 2015.

[Bur97] William C. Burns, 1997.

[CC11] Jonathan D. Cryer and Kung-sik Chan. *Time series analysis: with applications in R*. Springer, 2011.

[CH13] Samprit Chatterjee and Ali S. Hadi. *Regression Analysis by Example*. Wiley, 2013.

[DPX+93] Douglas W. Dockery, C. Arden Pope, Xiping Xu, John D. Spengler, James H. Ware, Martha E. Fay, Benjamin G. Ferris, and Frank E. Speizer. An association between air pollution and mortality in six u.s. cities. *New England Journal of Medicine*, 329(24):17531759, Sep 1993.

[JW09] Daniel J. Jacob and Darrell A. Winner. Effect of climate change on air quality. *Atmospheric Environment*, 43(1):5163, Jan 2009.