Time Series Analysis on London Mortality

Egan McClave, Aijin Wang

April 29, 2019

Contents

List of Figures

List of Tables

Abstract

The purpose of this report is to examine and understand the relationship between the number of deaths and environmental variables such as particulate matter and some weather variables. We analyzed 4 time series with 1826 observations per each series. We fit several models (Time Series Regression, Vector autoregression, Neural Network rutoregression) in an attempt to estimate the health risks associated with the given environmental variables. Based on these different models, Temperature appears to have a very influential relationship on understanding the number of deaths.

1 Introduction

Understanding mortality rates is an essential part of environmental epidemiology. Individually, ambient temperature/humidity and air pollution have been important determinants of mortality. It is of interest of us to investigate the associations between exposures such as air pollution, weather variables and human health. In this paper, we attempt to estimate the health risks associated with exposure to particulate matter (PM) and weather variables. Some advanced statistical models are necessary to study the possibly non-linear relationship among these variables of interest.

The data was originally introduced in the paper *Time Series Regression Studies in Environmental Epidemiology* published in International Journal of Epidemiology. The paper can be found here. The aim of the paper was to explore the basic modeling techniques that were appropriate for this problem. The data can be accessed from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3780998/. We will conduct our own analysis for this dataset and compare said results to the existing ones from the academic paper.

2 Methods

2.1 Data Description

The dataset contains daily observations of Ozone, O_3 ($\mu g/m^3$), Temperature (°C), Relative Humidity (%) and number of deaths from January 2002 to December 2006. A brief quantitative summary of the data is described in Table 1.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Ozone	1.18	21.09	34.92	34.77	46.73	119.25
Temperature	-1.40	7.51	11.47	11.72	16.20	28.17
Relative Humidity	31.23	58.69	69.61	69.10	80.36	98.86
Num Deaths	99.00	135.00	148.00	149.51	162.00	280.00

Table 1: Summary Statistics for Individual Time Series

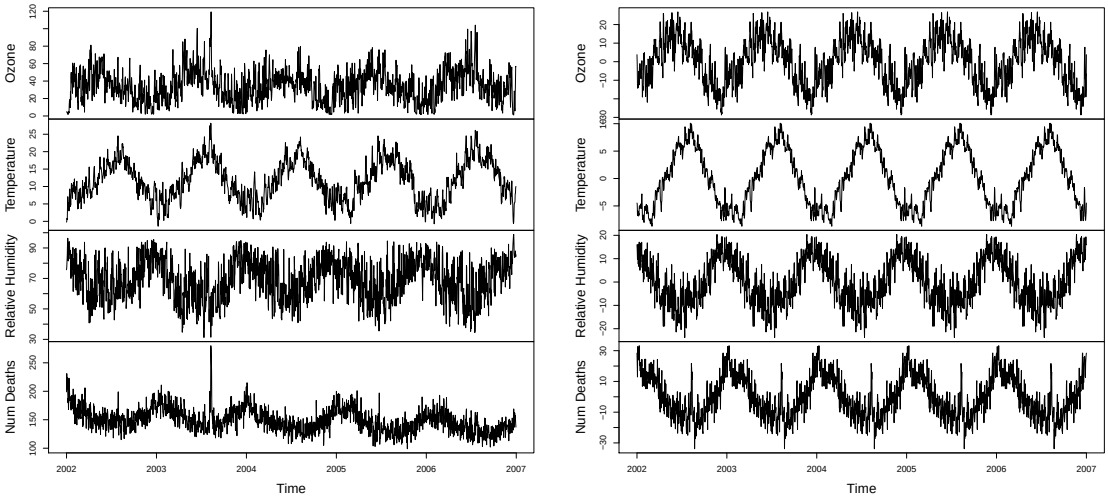
2.2 Exploratory Data Analysis

2.2.1 Univariate EDA

Before fitting time series models, we first want to understand the possible relationships among variables to get a better understanding about the structure of the data.

Figure 1 shows the individual time series plots for the dataset. From Figure 1 (a), we see that all three independent variables and the response variables have constant mean and variance. For number of deaths, there is an observation in 2003 that is extremely higher than the rest of the data. News reported that in August 3, 2003, Britain has recorded the highest temperature in 130 years, and the unusual weather might have led to the increasing number of deaths. Furthermore, the plots also suggests some seasonality effect in the data, since the plots show periodic patterns. Therefore, we explore the patterns by looking into the decomposed time series plots.

Figure 1 (b) displays the decomposed seasonality components for each series. As suggested in the overall series plot, we see an approximate yearly seasonality effect for all the variables. Specifically, `Ozone` and `Temperature` move in phase with one another as do `Num Deaths` and `Relative Humidity` with each other. However, these two pairings are out of phase with the other pair (they all have the same frequency ≈365 but change over time differently). We will consider seasonality in the model fitting.



(a) Overall Series

(b) Decomposed Seasonality Component

Figure 1: Visualizing Individual Time Series for London (2002 - 2007)

2.2.2 Multivariate EDA

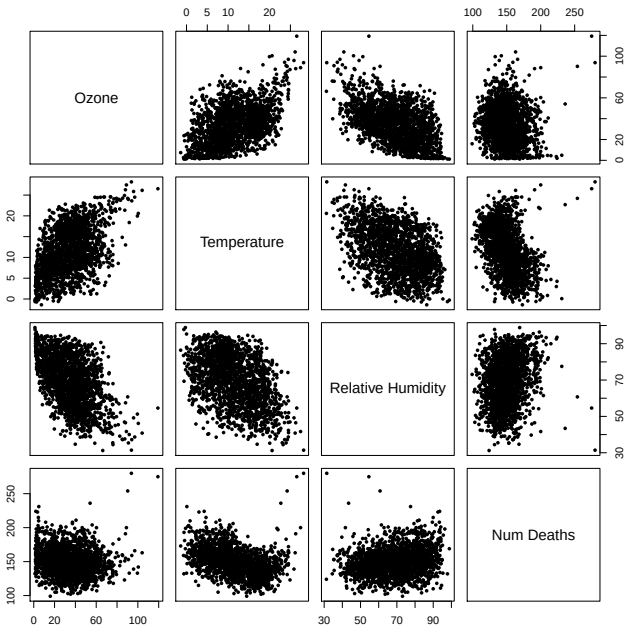
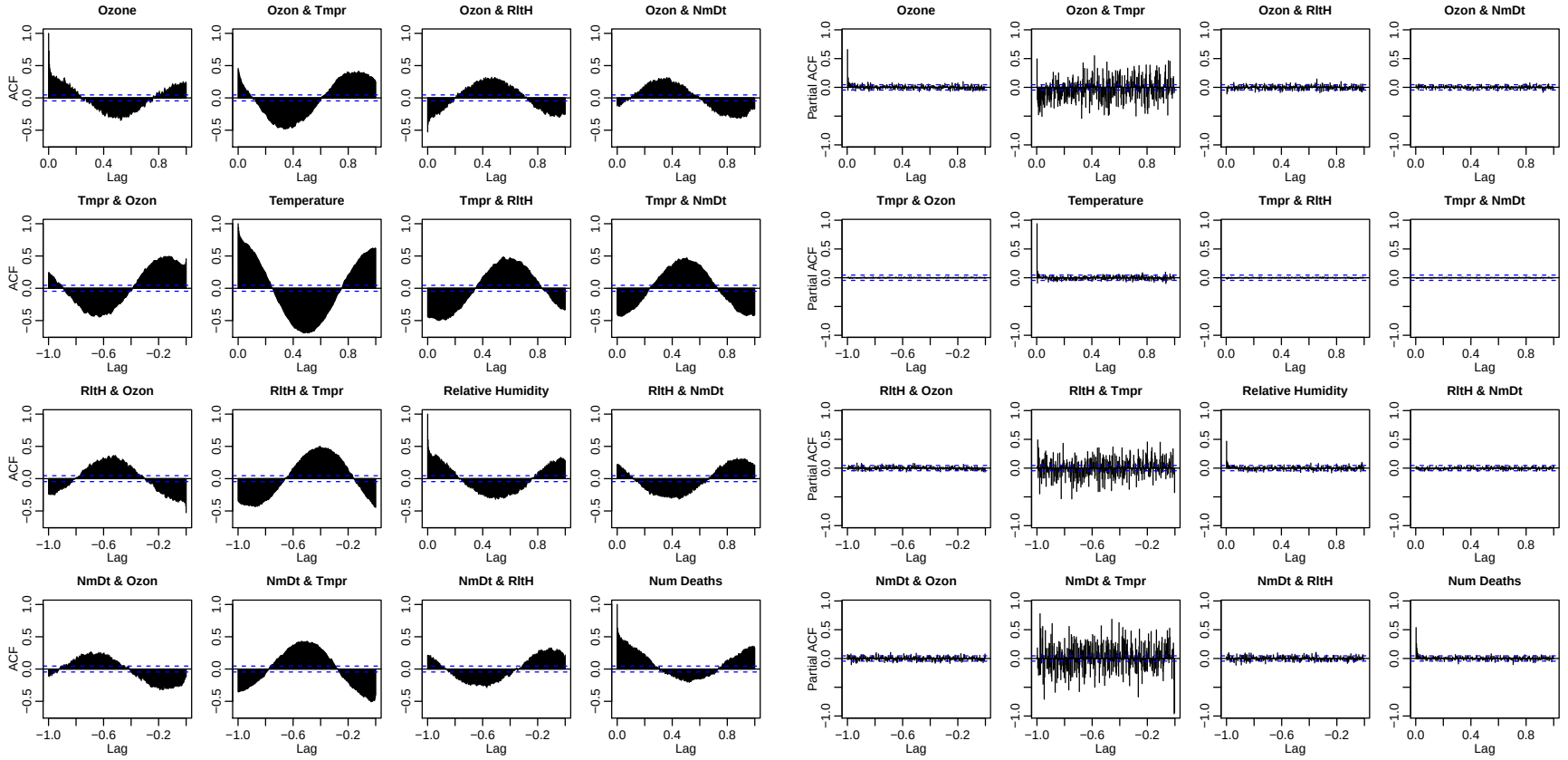


Figure 2: Pairs Plot of all Variables

Figure 2 shows the pairwise scatterplots in the dataset. Both `Ozone` and `Relative Humidity` don't show strong correlation with `Num Deaths`, but there is a weak quadratic relationship between `Temperature` and `Num Deaths`. This suggests that some variable transformations may be useful in order to account for the relationship.



(a) ACF/CCF Plots of Individual Series

(b) PACF/PCCF Plots of Individual Series

Figure 3: Auto-Correlation Plots of Individual Series

Figure 3 illustrate the ACF/CCF and PACF/PCCF of the individual series up to a full 365 days of lag. The ACF/CCF plots in Figure 3 (a) that ACF values depend on the lag and have periodic patterns. Large number of the values are also outside the confidence level. This further proves that the original series is not stationary and have seasonality effects.

There are four PACF plots on the diagonal as shown in Figure 3(b). For all four variables, the plots have tails off, meaning that some AR models are appropriate for the dataset. Out of all the PCCF plots, only the one between number of deaths and temperature has high PCCF values for various lags. This is evidence to prove that there might be some relationship between the two variables.

2.3 Variable Transformation

Based on the exploratory data analysis in subsection 2.2, we discovered that there is potentially a quadratic relationship between `Temperature` and `Num Deaths`. Therefore, we will include a quadratic transformation of `Temperature` in the modeling. Furthermore, we use the mean adjusted version of `Temperature` instead of the original version to ensure the calculations are more stable. In addition, we also engineer the variables `Day of Week` and `Day of Month` from the dates provided.

In summary, below are the variables that we use for the model building after transformation:

- `Time`
- `Ozone Levels`
- `Relative Humidity`
- `Adjusted Temperature`
- `Adjusted Temperature`^{2}
- `Num Deaths`
- `Day of Week`
- `Day of Month`

2.4 Model Identification

Based on the above analysis and the research goal of the project, we will be fitting three types of model, which are:

- Time Series Regression
- Vector Autoregression
- Neural Network Autoregression

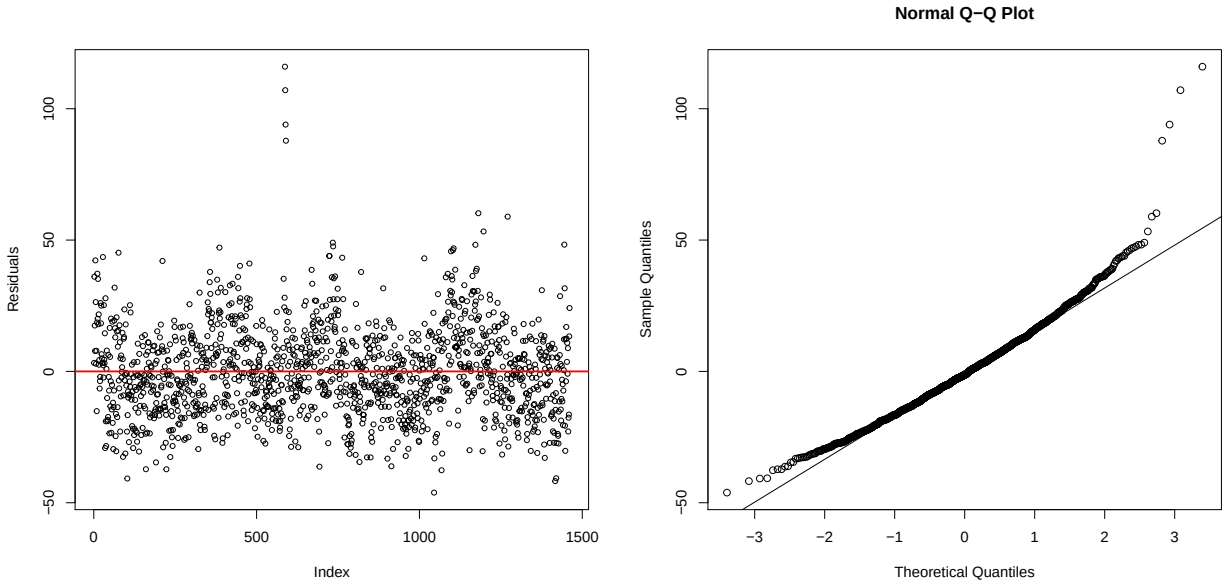
In this section, we will discuss the steps that we take to fit each model.

2.4.1 Time Series Regression

Manual Identification

We first fit a simple linear regression model using OLS with all the variables mentioned above as exogenous variables. The residual plot, Q-Q plot, ACF and PACF of the residuals are displayed in Figure 4 and Figure 5. The residuals in Figure 4(a) are not randomly distributed, and show some periodic patterns. Though the plot has mean centered around zero, it also shows heteroskedasticity. There are also four observations that have relatively high values. The Q-Q plot shows a similar result. The points show a bell curve, with the end of the plot being heavily tailed. These are evidence implying that the simple OLS model may not be a good fit for the dataset.

The ACF and PACF plots prove that the residuals are not white noise, but instead have some AR and MA behavior. Upon further examination of the plots, we identify an `ARMA(2,8)` model for the residuals, and refit time series regression with all the exogenous variables assuming that the residuals follow ARMA with the orders (2,8) found in the previous step.



(a) Residuals over Time

(b) Residual QQ Plot

Figure 4: Visualizing Residuals

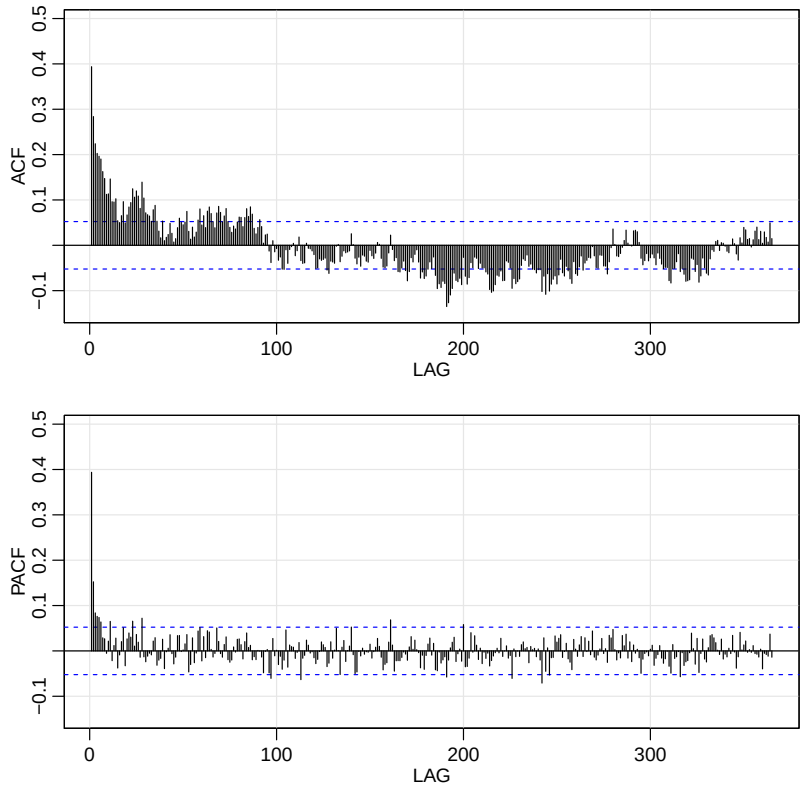


Figure 5: ACF/PACF Plot of Residuals

auto.arima

Another approach that can help identify the order of residuals is the function `auto.arima` in R. Since we already know that the residuals are not white noise, we will use `auto.arima`, which returns the best ARIMA model based on information criteria(AIC or BIC value) to get the order for the residuals. We will also be accounting for the effects of exogenous variables, same as the previous section.

```
model <- auto.arima('Num Deaths', seasonal=T, xreg=..., stepwise=F, approx=F, type='none')
```

After running `auto.arima`, we identidy an `ARMA(1,1)` model for the residuals. Same as above, we then refit time series regression all the exogenous variables, but assuming that the residuals follow ARMA with the orders (1,1).

The specific model fitting, validation and inference will be expanded upon in detail in subsection 3.1. We will compare the results between ARMA(1,1) and ARMA(2,8) and identify a better model.

2.4.2 Vector Autoregression

The parameter selection for VAR models is straightforward and primarily requires only the lag order p for the model. Based on the EDA in subsection 2.2 we know there is a seasonality component to the data at $\approx$365 days. We will also be providing several exogenous variables as mentioned above. For this reason we will be using the `VARselect()` function (from the `vars` package) as shown below to determine the lag order for two different VAR models (`season=NULL` and `season=365`). The lag order parameters for the different criteria are displayed in table Table 2.

```
selection <- VARselect('Num Deaths', season=..., exogen=..., type='none')
```

	AIC(n)	HQ(n)	SC(n)	FPE(n)
season=NULL	6	3	3	6
season=365	8	3	2	6

Table 2: `VARselect()` Order Selection for Different Models

Table 2 contains the information criteria and final prediction error for the two different models. Based on the results of this table, the two models will be a VAR($p = 42$, season = `NULL`) and VAR($p = 42$, season = 365). The specific model fitting, validation and inference will be expanded upon in detail in subsection 3.2.

2.4.3 Neural Network Autoregression

Lastly, our third model is a Neural Network Autoregression (NNAR) utilizing the `nnetar()` function from the `forecast` package as described below.

```
nn_fit <- nnetar('Num Deaths', xreg=...)
```

This is relatively straightforward and requires little to no overhead with implementation other than specifying the exogenous variables with the `xreg` argument. The specific model fitting, validation and inference will be expanded upon in detail in subsection 3.3.

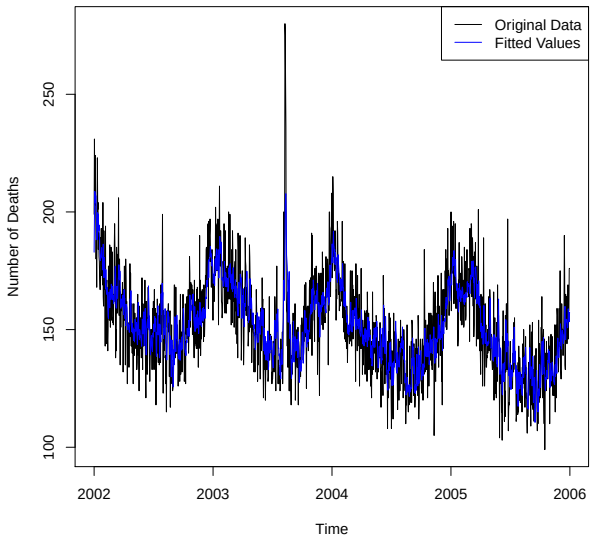
3 Results

We split the data into training and testing sets, and use model trained on the training set to predict number of deaths with all the exogenous variables on test set. The training set contains data from year 2002 to 2006, and the test set contains data from 2006 to 2007. All the models below are only fitted using the training set.

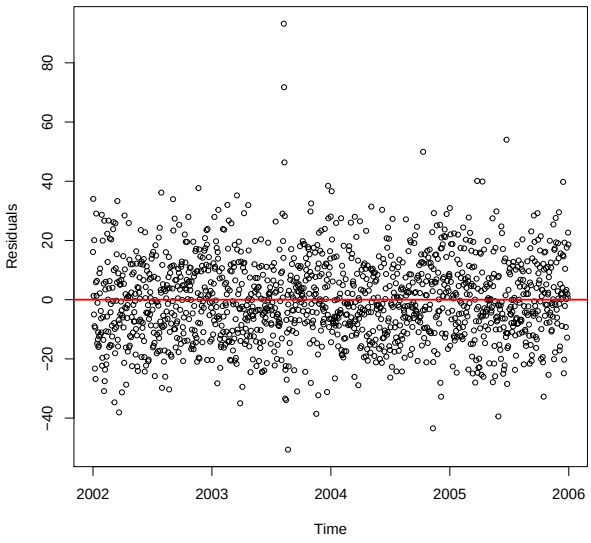
3.1 Time Series Regression

Model Fitting

After the initial model identification, we narrow down to two candidates for the residual models: ARMA(1,1) (from manual identification) and ARMA(2,8) (from `auto.arima`). Assuming that the residuals for OLS follow ARMA model with order (p,q), Figure 6 and Figure 7 show the fitted values and the new residual plots for the refitted time series regression models, corresponding to ARMA(1,1) and ARMA(2,8) respectively.

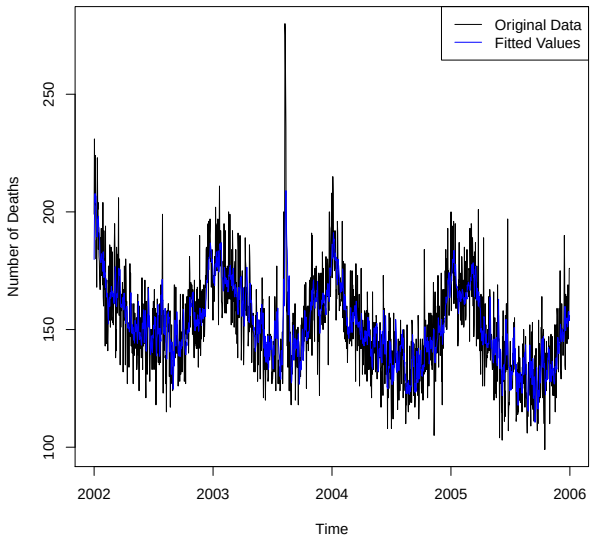


(a) Fitted Values vs Original Series

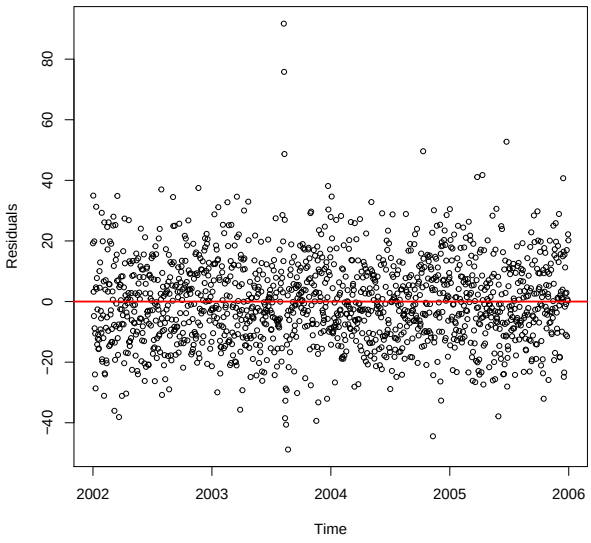


(b) Residuals over Time

Figure 6: Visualizing ARIMA Fit



(a) Fitted Values vs Original Series



(b) Residuals over Time

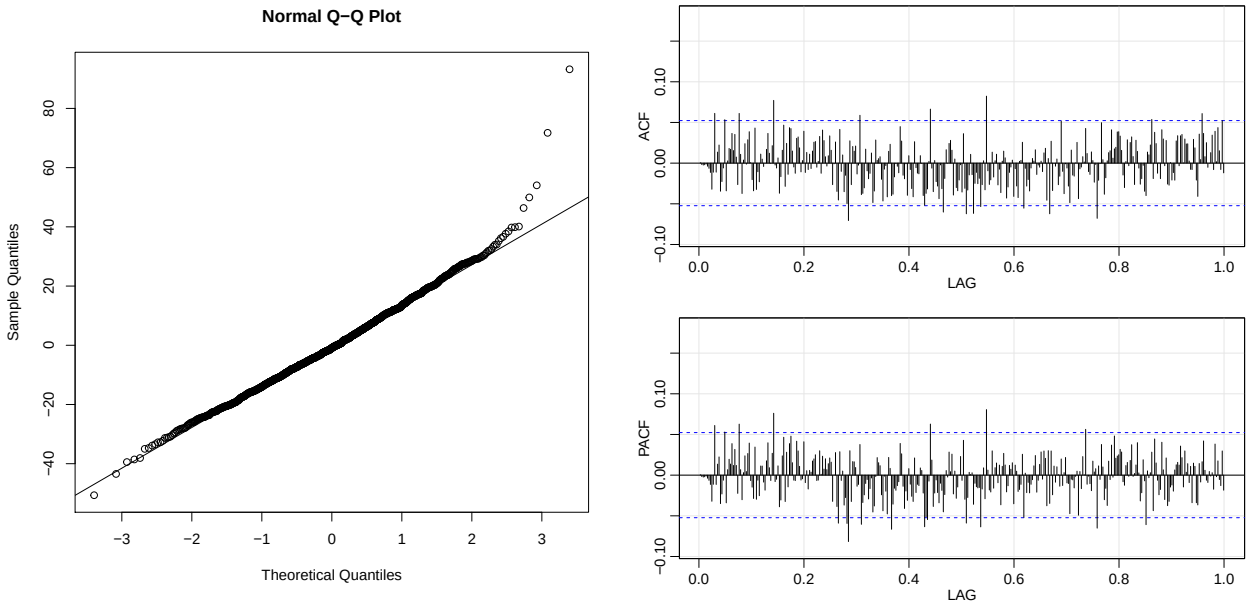
Figure 7: Visualizing `auto.arima` Fit

The fitted values for both models fit the original data pretty well and they both capture the trend and seasonality of the data. Similarly, the residual plots for both models look good. The residual values are

randomly distributed and centered around zero. Both plots show constant variance and have no apparent patterns.

Model Validation

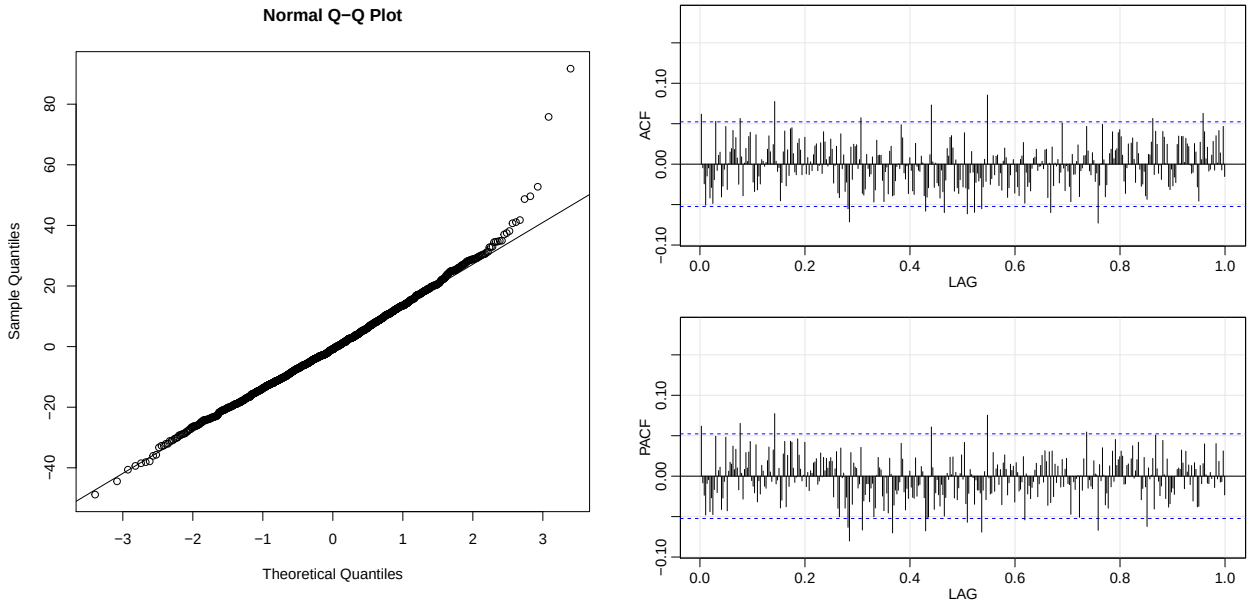
Figure 8 and Figure 9 show the QQ plots and the ACF/PACF plots for the refitted time series regression models, corresponding to ARMA(1,1) and ARMA(2,8) respetively.



(a) QQ Plot of Residuals

(b) ACF/PACF of Residuals

Figure 8: Visualizing ARIMA Residuals



(a) QQ Plot of Residuals

(b) ACF/PACF of Residuals

Figure 9: Visualizing `auto.arima` Residuals

For both QQ plots, the points follow the normal QQ line, with the upper side having a slightly heavy tail. Unlike OLS, majority of the ACF and PACF values are within the confidence bands, meaning these values do not depend on lag. From all plots above, we can conclude that the new residuals for both ARMA(2,8) and ARMA(1,1) model are white noise.

With all the evidence, we can say that both models do a equally good job in model fitting. However, here we will pick the results produced by `auto.arima` (residuals follow ARMA(1,1)) to proceed due to the simplicity of the model.

Model Forecasting

Figure 10 shows the forecast results for `auto.arima`. From the plot, we see that the point estimate do well only in the beginning of the prediction period. The model does capture the downward trend of the series, the daily volatility, and predicts some level of seasonality, but the predicted cycle is not in sync with the original series. The 95% confidence interval are fairly wide, and share the same patterns as the point estimate.

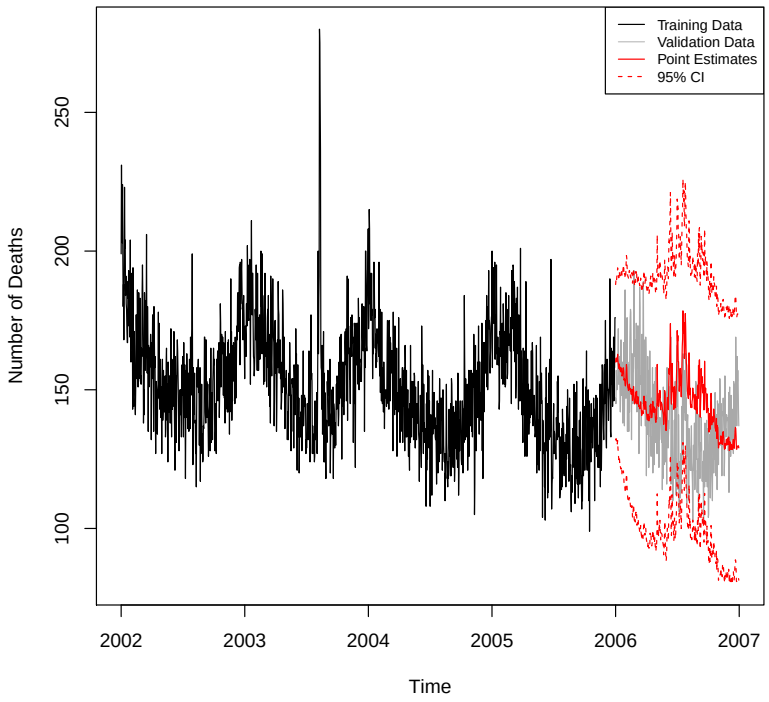


Figure 10: Visualizing `auto.arima` Forecasts

Model Inference

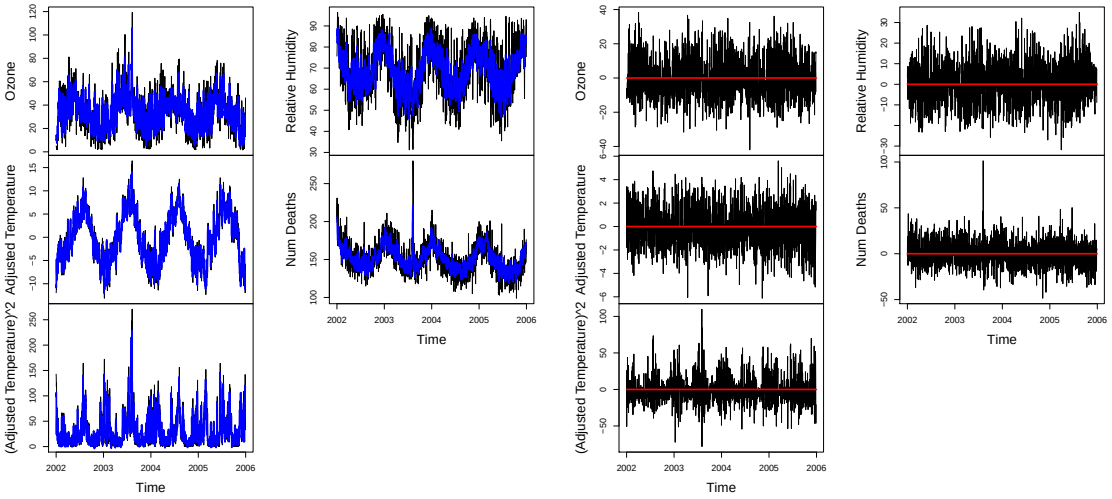
Table 3 shows the coefficient and standard error for each variable in the time series regression. From the table, we see that `Adjusted Temperature` has high positive coefficient. This means that there is a positive correlation between temperature and number of deaths: holding all the other variables fixed, when the temperature gets higher, number of deaths also increases. Ozone levels have a negative correlation with number of deaths and relative humidity has a positive correlation, though the results for both variables are insignificant within 95% confidence interval (because the interval includes zero). Other variables that have significant coefficients are `ar1`, `ma1` and `(Adjusted Temperature)`^{2}.

3.2 Vector Autoregression

Model Fitting

	Estimate	Std. Error
ar 1	0.99	0.00
ma 1	-0.78	0.02
intercept	13289.39	10715.63
Trend	-6.56	5.35
Adjusted Temperature	1.41	0.17
(Adjusted Temperature)^2	0.12	0.02
Ozone	-0.04	0.03
Relative Humidity	0.06	0.04
Day of Week	0.18	0.17
Day of Month	-0.01	0.05

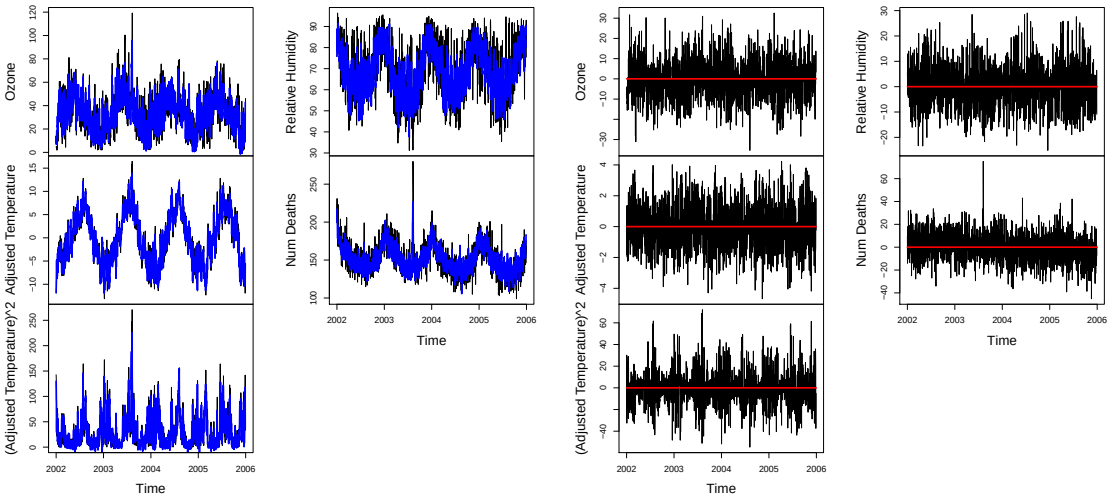
Table 3: Time Series Estimated Coefficients



(a) Fitted Values vs Original Series

(b) Residuals for Individual Series

Figure 11: Visualizing VAR(p=3, season=`NULL`)



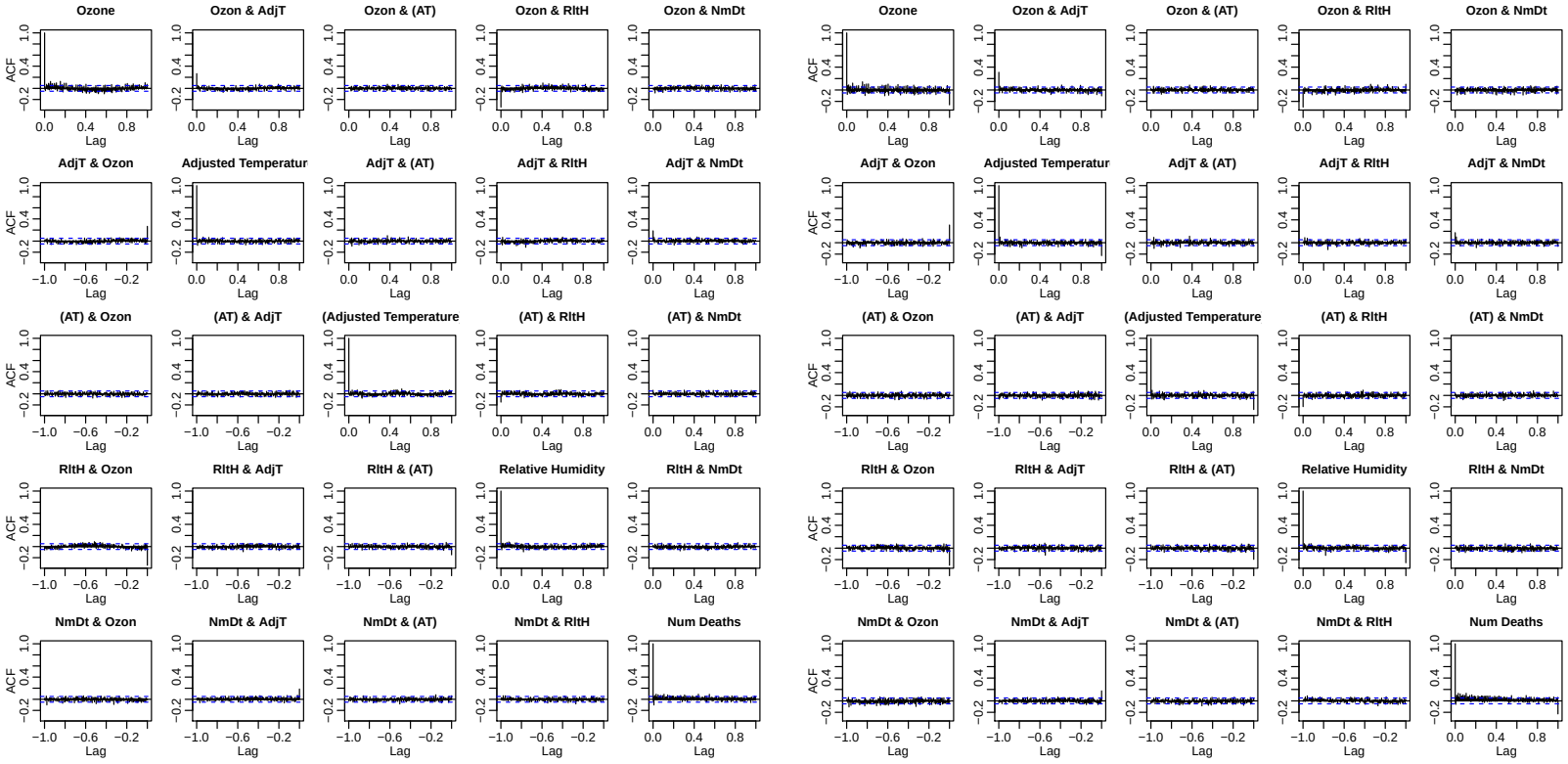
(a) Fitted Values vs Original Series

(b) Residuals for Individual Series

Figure 12: Visualizing VAR(p=2, season=365)

Figure 11 and Figure 12 visualize the fitted values and the residuals for the VAR(p=3, season=`NULL`) and VAR(p=2, season=365) models. Between the two sets of figures the plots appear nearly identical with the second model slightly capturing the variability of the individual series more than the first model. This can be identified in the left hand plots where the fitted values in blue more closely follow the original data specifically in the `Ozone`, `Relative Humidity` and `Num Deaths` series for the second model. On the right hand side, the residuals are mostly centered around the line $y = 0$ and maintain a constant variance over time except for the notable spike in 2003 and for the squared adjusted temperature series as a whole.

Model Validation



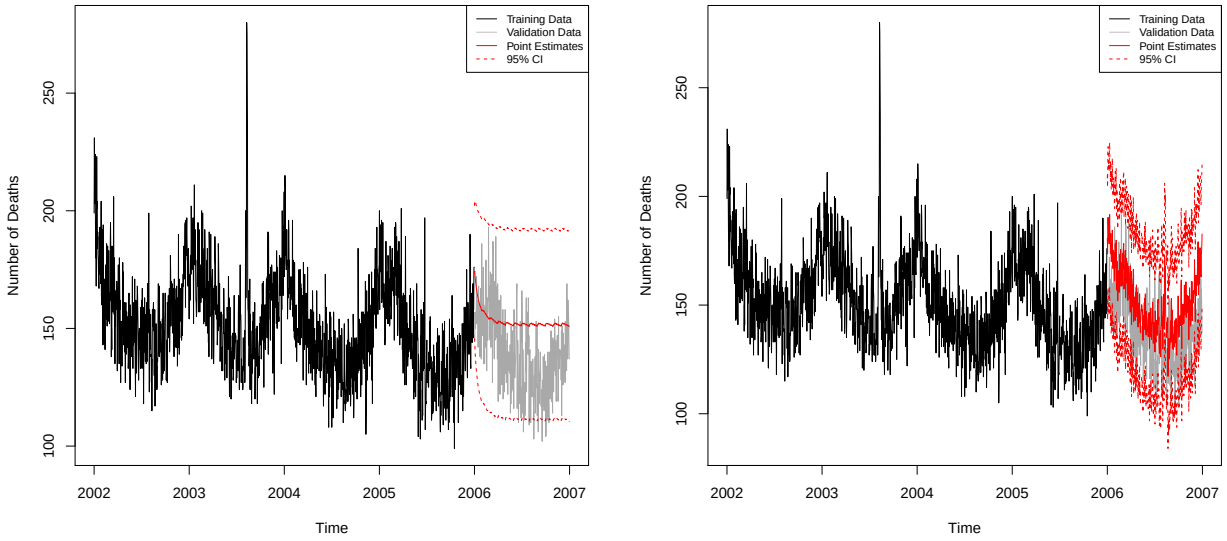
(a) VAR(p=3, season=`NULL`)

(b) VAR(p=2, season=365)

Figure 13: ACF/CCF Plots

Figure 13 features the ACF/CCF plots for the VAR(p=3, season=`NULL`) and VAR(p=2, season=365) models. From this extensive grid of plots we can see that the residuals from the two models appear to come from a white noise process as indicated by the lack of statistically signficant non-zero auto/cross-correlations. It is also important to note that some of the ACF plots for the VAR(p=2, season=365) model have a slight statistically significant value at lag 365 which suggests the data still needs some further manipulation.

Model Forecasting



(a) VAR(p=3, season=`NULL`)

(b) VAR(p=2, season=365)

Figure 14: Visualizing Forecasting

Figure 14 depicts the predictions for the testing data with the VAR(p=3, season=`NULL`) and VAR(p=2, season=365). Figure 14 (a) has an overall poor fit with the point estimates but the 95% confidence interval mostly contains the value from the testing data. These predictions also capture the decreasing trend that the original series displays. Figure 14 (b) has a much better fit and the point estimates follow the observed testing data much very well as well as the 95% confidence intervals capturing the test data within its bounds. Additionally, the point estimates and confidence intervals exhibit the same seasonal and daily volatility that the original data exhibits. This difference in quality of the fits can likely be attributed to the `seasonal` parameter. The inclusion of this parameter provides the second model with an additional 365 estimated coefficients. For this reason we will be examining the more parsimonious model, VAR(p=3, season=`NULL`) since we are more interested in the inference capabilities of our model. It is worth noting how excellent the fit for the second model is and it should be kept in mind for further analysis.

Model Inference

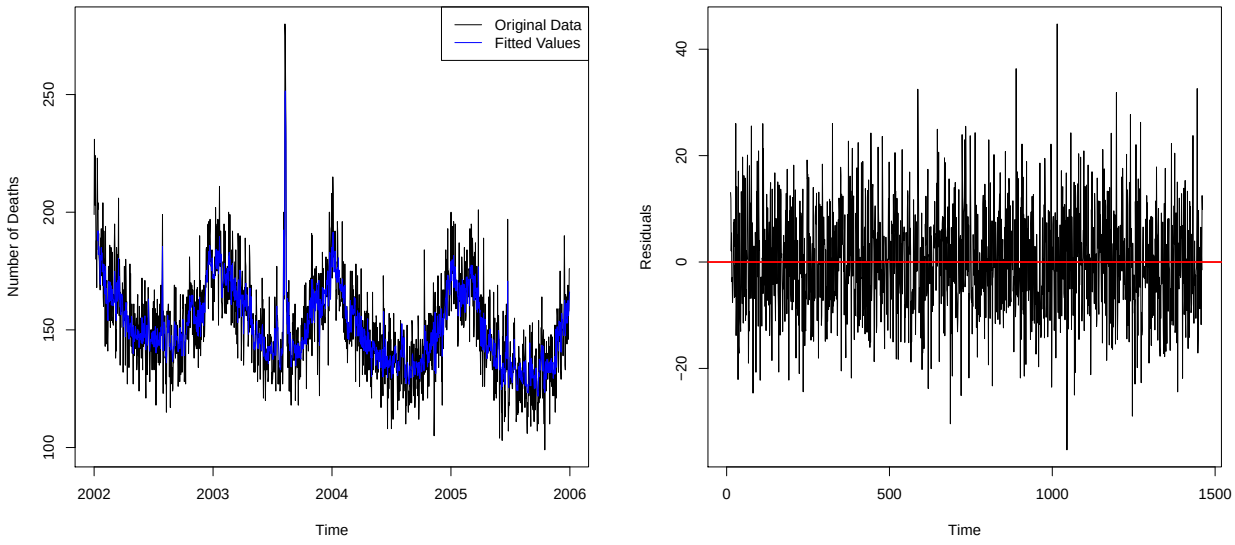
Table 4 displays the summary of the estimated coefficients of the VAR(p=3, season=`NULL`) model when regressing onto `Num Deaths` at time t. Few of the estimated coefficients are statistically significant (using the unadjusted p-values) and even fewer have estimated coefficients that are not near 0. The most important coefficients to note here (given the current variables in the model) are the lagged coefficients for `Adjusted Temperature` and the lagged coefficients for `Num Deaths`. These coefficient estimates are fairly large which suggest that they have some relationship with `Num Deaths` in the current time period. Based on the signage of these coefficients an increase in `Adjusted Temperature` at $(t-1)$ has an increased number of deaths at time t and an increase in `Adjusted Temperature` at $(t-2)$ and $(t-3)$ have a decrease number of deaths at time t. Similarily, an increase in `Num Deaths` at all lags $1, 2, 3$ leads to an increase in the number of deaths at time t.

3.3 Neural Network Autoregression

Model Fitting

	Estimate	Std. Error	t value	Pr(> |t|)
Ozone lag 1	0.043	0.035	1.229	2.19E-01
Adjusted Temperature lag 1	0.466	0.222	2.102	3.57E-02
(Adjusted Temperature)^2 lag 1	0.102	0.020	5.028	5.58E-07
Relative Humidity lag 1	0.016	0.039	0.402	6.88E-01
Num Deaths lag 1	0.280	0.026	10.600	2.48E-25
Ozone lag 2	0.074	0.043	1.726	8.46E-02
Adjusted Temperature lag 2	-0.737	0.306	-2.408	1.62E-02
(Adjusted Temperature)^2 lag 2	-0.009	0.027	-0.343	7.32E-01
Relative Humidity lag 2	0.035	0.042	0.829	4.07E-01
Num Deaths lag 2	0.185	0.027	6.858	1.03E-11
Ozone lag 3	-0.055	0.035	-1.559	1.19E-01
Adjusted Temperature lag 3	-0.636	0.225	-2.823	4.82E-03
(Adjusted Temperature)^2 lag 3	-0.023	0.020	-1.148	2.51E-01
Relative Humidity lag 3	-0.053	0.040	-1.332	1.83E-01
Num Deaths lag 3	0.169	0.026	6.601	5.74E-11
Trend	0.026	0.003	10.256	7.19E-24
Day of Week	0.028	0.192	0.144	8.86E-01
Day of Month	-0.023	0.043	-0.537	5.92E-01

Table 4: `Num Deaths` Coefficients from VAR(3)



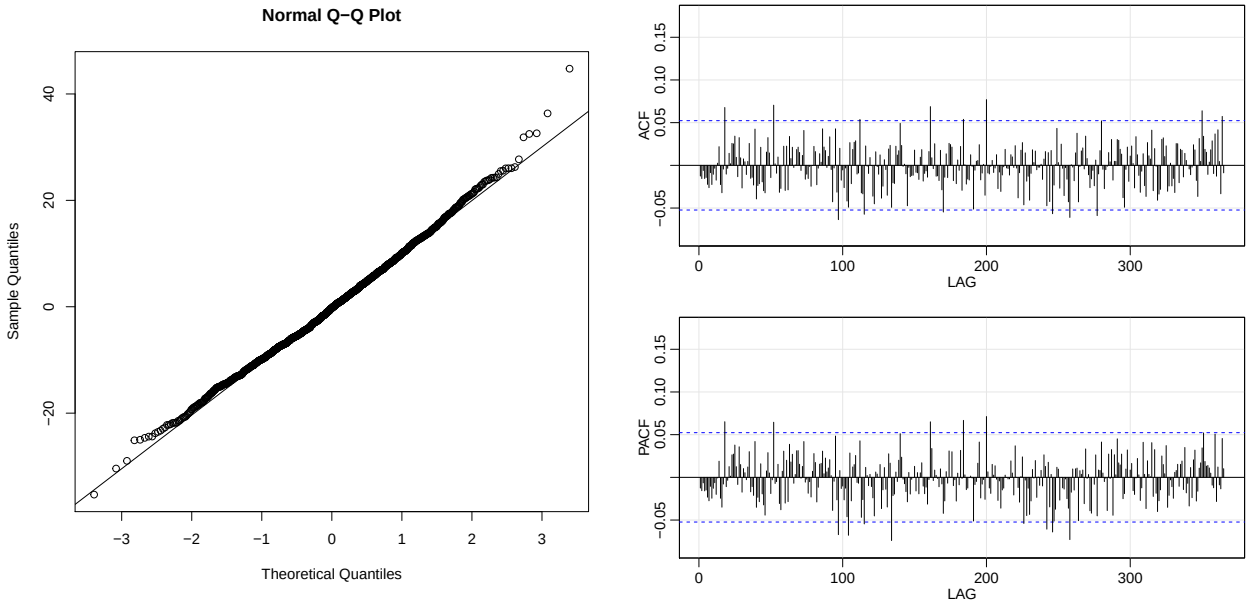
(a) Fitted Values vs Original Series

(b) Residuals over Time

Figure 15: Visualizing NNAR Fit

Figure 15 illustrates the fitted values and residuals for the NNAR model described earlier. The model resulting from `nnetar()` is NNAR(10, 11). Figure 15 (a) shows that the fitted values nearly follow the original training data quite well. Figure 15 (b) shows that the residuals are almost evenly centered around $y = 0$ and maintain a constant variance throughout time. Thus, this model is likely a good fit to the training dataset.

Model Validation



(a) QQ Plot of Residuals

(b) ACF/PACF of Residuals

Figure 16: Visualizing NNAR Residuals

Figure 16 displays the NNAR model residuals in more detail. Figure 16 (a) shows that the residuals approximately follow a normal distribution based on the QQ plot with only a slight deviation in the left tail. Figure 16 (b) shows the ACF/PACF plot which illustrate that the remaining residuals approximately follow a white noise distribution. Thus, supporting that this model does well in the training aspect.

Model Prediction

Forecasting Ahead:

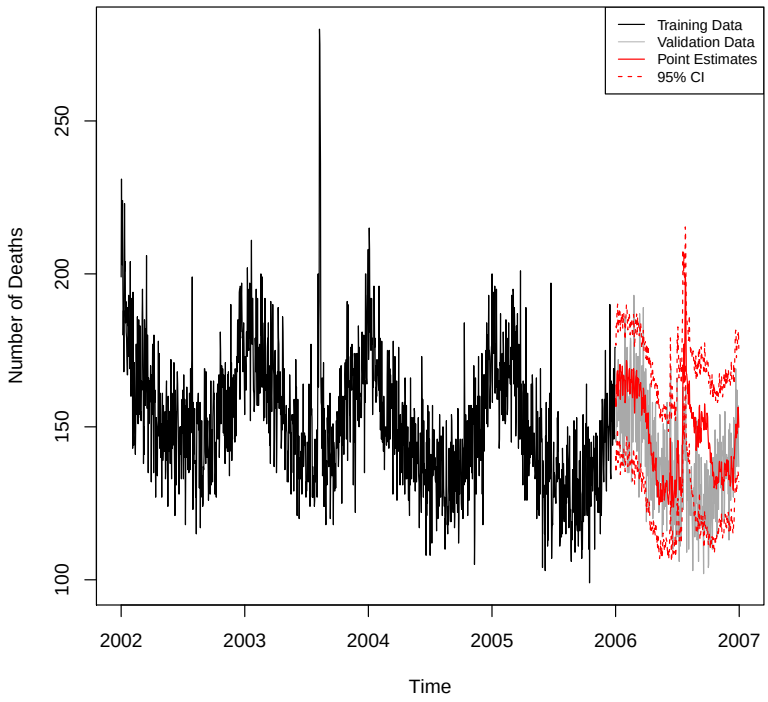
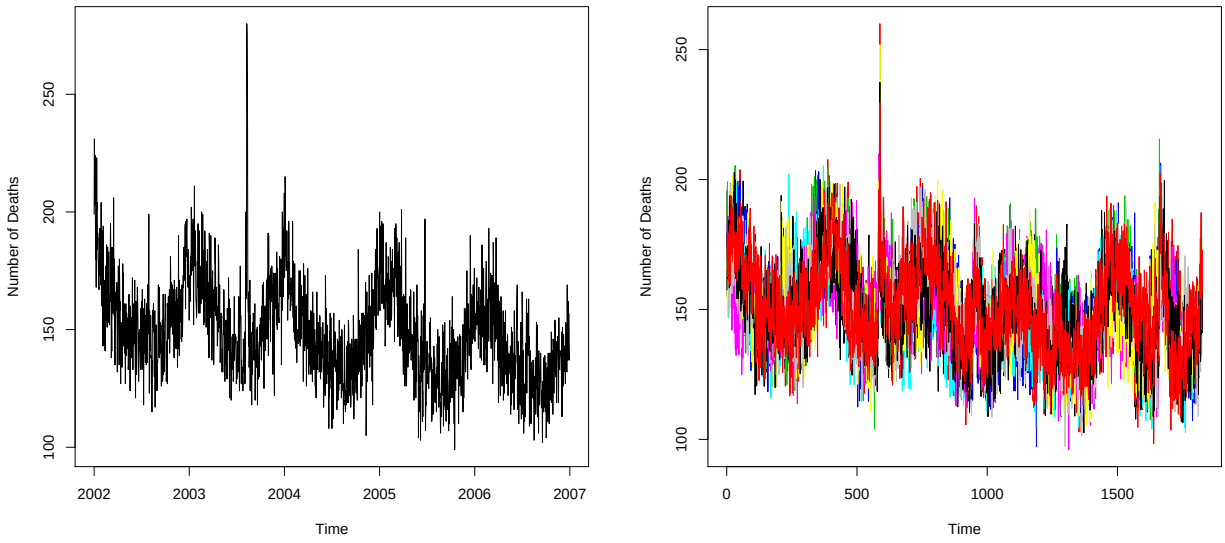


Figure 17: Visualizing `NNAR` Forecasts

Figure 17 displays the predictions for the testing data using the NNAR model. The point estimates do fairly well with prediction except for a period of time roughly half way through the testing year. The 95% confidence interval also manages to capture most of the data within its bounds, with the exception of the same period of time as the point estimations. It is also important to note that the model manages to capture the overall seasonal trend and daily volatility that the original data displays. Thus, the generalizability of the model appear adequate but not great.

Simulated Data:



(a) Original Series

(b) Simulated Data

Figure 18: Comparing Original Series to Simulated Data

Unlike the other models, the NNAR model has the capability to easily simulate data. Figure 18 displays both just a few simulations and the original observed series in full. Based on these two plots, it sound to say that the observed data might come from whatever distribution the Neural Network black box is generating. This is more evidence that the generalizability of the model is quite good.

4 Discussion

In this analysis, we fit three model for the time series data to predict the number of deaths, using time, ozone levels, relative humidity, temperature and temperature^2. The three models that we fit are time series regression, Vector autoregression and Neural Network autoregression.

For time series regression, we first fit a linear regression using OLS, and fit two ARIMA models for the residuals, using ACF/PACF plots and auto.arima respectively. We then refit the regression model, assuming that the residuals follow the ARIMA models, and examine the fit of the new model. When comparing the results from the two approaches, we see that they do an equally good job in model fitting, residual plots, Q-Q plots, and ACF/PACF plots. Therefore, we choose the model with lower order of MA and AR. The final ARIMA model shows a moderately good forecasting behaviour, capturing the overall trend, daily volatility, but it fails to predict the correct seasonality. This model also showed that there is some directly proportional relationship between temperature and number of deaths and an inversely proportional relationship between time and number of deaths.

For Vector autoregression, we examined two models based on the seasonality components from the EDA. They both had relatively the same fit behaviour but drastically different forecasting/prediction behaviour. Thus, we choose the less complicated model in favor of explanation. The simplier model illustrated that there is some lagged effect with temperature onto the number of deaths at time t.

For Neural Network autoregression, we left the fitting to the black box and we were provided a model whose predicted behaviour was mostly good and captured the overall series. However, due to the nature of this statistical learning technique there is little that can learned about the relationship between number of

deaths and other series.

Table 5 displays the AIC and MSE values for the main models of interest for this analysis. Because of how Neural Networks work it does not make much sense to calculate the AIC value so this was omitted.

	ARIMA	VAR$_1$	VAR$_2$	NNAR
AIC	11928.01	52497.02	53802.00	NA
MSE	372.30	430.24	411.99	339.42

Table 5: Evaluation Results Across all Models

In the future, there are several things that might be of interest for further analysis. For example, utilizing `auto.arima()` to explore other parameters for the `nnetar`, gathering other influential series that might have a relation with our response variable, such as precipitation, or number of weather related incidents, ... and re-examining our work with decomposed components of these series.

5 Appendix

Preliminary Setup

```
# Load packages
pkgs <- c('xtable', 'astsa', 'vars', 'MTS', 'foreign', 'forecast', 'knitr', 'lubridate', 'dplyr', 'magr
lapply(pkgs, library, character.only=T)

# Load data
### Custom color palette
cols <- c('#000000', '#999999', '#E69F00', '#56B4E9', '#009E73',
          '#FF0000', '#F0E442', '#0072B2', '#D55E00', '#CC79A7')

### Dataset
df <- read.dta('./data/ije-2012-10-0989-File003.dta')
colnames(df) <- c('Date', 'Ozone', 'Temperature', 'Relative Humidity', 'Num Deaths')

# Combine the series into a ts object with appropriate time series labeling
ts_vars <- ts(df[,2:5], start=c(2002, 1), frequency=365.25)

# Cache chunk options
opts_chunk$set(cache=T, autodep=T, cache.comments=F)

source('./fxns.R')
```

Introduction

```
# Create a sequence of numbers
tab <- t(apply(ts_vars, 2, summary))
xt <- xtable(tab, label='tab:data_summary',
    caption='Summary Statistics for Individual Time Series')
print(xt)
```

Method

```
# Plotting overall time series
plot.ts(ts_vars, main = "")

# Determining seasonality component
seasonal = lapply(df[,2:5], function(x, vars) {
  x <- ts(x, start=c(2002, 1), frequency=365)
  temp <- decompose(x)
  return(temp$seasonal)
})

# Plotting seasonality componen
plot.ts(do.call(cbind, seasonal), main = '')
```

```
# pairs plot of all series
pairs(ts_vars, cex=0.75, pch=16)
```

```
# Variable transformation indicated by EDA
df %<>%
  mutate(`Adjusted Temperature` = Temperature - mean(Temperature),
         `(Adjusted Temperature)^2` = `Adjusted Temperature`^2,
         `Day of Week` = lubridate::wday(Date),
         `Day of Month` = lubridate::mday(Date))

ts_vars <- ts(df[,c(2, 6:7, 4:5, 8:9)], start=c(2002, 1), frequency=365.25)

# Formulate training/testing split where testing is 1 year
train <- window(ts_vars, start=c(2002, 1), end=c(2005, 365.25))
test <- window(ts_vars, start=c(2006, 1))
train_trend = time(train); test_trend = time(test)

results <- matrix(0, nrow=2, ncol=4,
  dimnames=list(c('AIC', 'MSE'),
    c('ARIMA', 'VAR$_1$', 'VAR$_2$', 'NNAR')))
```

```
# Fit `lm()` to exogenous variables to determine ARIMA fit
# Fit `lm()` to exogenous variables to determine ARIMA fit
ts_reg_1 = lm(train[,'Num Deaths'] ~ train_trend + train[,'Ozone'] +
  train[,'Adjusted Temperature'] + train[,'(Adjusted Temperature)^2'] +
  train[,'Relative Humidity'] + train[,'Day of Week'] + train[,'Day of Month'])

# Looking at residuals
plot(resid(ts_reg_1), ylab='Residuals', cex=0.75)
abline(h=0, col='red', lwd=2)
qqnorm(resid(ts_reg_1)); qqline(resid(ts_reg_1))
```

```
# Looking at ACF/PACF for model identification
invisible(astsa::acf2(resid(ts_reg_1), 365.25, main=''))
```

```
###### Parameter selection
train_var <- train[,c('Ozone', 'Adjusted Temperature', '(Adjusted Temperature)^2',
  'Relative Humidity', 'Num Deaths')]
bind <- cbind(trend=train_trend, wday=train[,'Day of Week'], mday=train[,'Day of Month'])

# Calculate Information Criteria for different VAR(p) models with season=NULL
VARselect_res_1 <- VARselect(train_var, season=NULL, exogen=bind, type='none')

# Calculate Information Criteria for different VAR(p) models with season=365
VARselect_res_2 <- VARselect(train_var, season=365, exogen=bind, type='none')

selection <- rbind(VARselect_res_1$selection, VARselect_res_2$selection)
rownames(selection) <- paste0('season=', c('NULL', '365'))
xt <- xtable(selection, label='tab:var_order',
  caption='\\texttt{VARselect()} Order Selection for Different Models')
print(xt, table.placement='H')
```

Results

Time Series Regression

```
# Fit `Arima()` to residuals of exogenous variables
bind <- cbind(trend=train_trend, temp=train[,'Adjusted Temperature'],
  temp2=train[,'(Adjusted Temperature)^2'], Ozone=train[,'Ozone'],
  `Relative Humidity`=train[,'Relative Humidity'], wday=train[,'Day of Week'],
  mday=train[,'Day of Month'])
ts_reg_2 = Arima(train[,'Num Deaths'], order=c(2, 0, 8), xreg=bind, optim.control=list(maxit=1000))

# Looking at fitted values
plot(train[,'Num Deaths'], ylab='Number of Deaths')
lines(fitted(ts_reg_2), col='blue')
legend('topright', c('Original Data', 'Fitted Values'), lty=c(1,1), col=c(1,'blue'))

# Looking at residuals over time
plot(resid(ts_reg_2), ylab='Residuals', type='p', cex=0.75)
abline(h=0, col='red', lwd=2)
```

```
# Fit `Arima()` to residuals of exogenous variables
ts_reg_3 = auto.arima(train[, 'Num Deaths'], xreg=bind, seasonal=T,
  stepwise=F, approximation=F, optim.control=list(maxit=1000))

# Looking at fitted values
plot(train[, 'Num Deaths'], ylab='Number of Deaths')
lines(fitted(ts_reg_3), col='blue')
legend('topright', c('Original Data', 'Fitted Values'), lty=c(1,1), col=c(1,'blue'))
```

```r
# Looking at residuals over time
plot(resid(ts_reg_3), type='p', cex=0.75, ylab = 'Residuals')
abline(h=0, col='red', lwd=2)
```

```r
# Looking at normality of residuals
qqnorm(resid(ts_reg_2)); qqline(resid(ts_reg_2))

# Looking at ACF/PACF for white noise
invisible(astsa::acf2(resid(ts_reg_2), 365.25, main=''))
```

```r
# Looking at normality of residuals
qqnorm(resid(ts_reg_3)); qqline(resid(ts_reg_3))

# Looking at ACF/PACF for white noise
invisible(acf2(resid(ts_reg_3), 365.25, main=''))
```

```r
###### Forecasting test data
bind <- cbind(trend=test_trend, temp=test[, 'Adjusted Temperature'],
  temp2=test[, '(Adjusted Temperature)^2'], Ozone=test[, 'Ozone'],
  `Relative Humidity`=test[,'Relative Humidity'], wday=test[,'Day of Week'],
  mday=test[,'Day of Month'])
fcast <- forecast(ts_reg_3, level=95, xreg=bind)

plot(train[, 'Num Deaths'], ylab='Number of Deaths', xlim=c(2002, 2007),
  ylim=range(fcast$lower, fcast$upper, ts_vars[, 'Num Deaths']))
lines(test[, 'Num Deaths'], col = 'darkgrey')
lines(fcast$upper, col = 'red', lty = 2)
lines(fcast$mean, col = 'red')
lines(fcast$lower, col = 'red', lty = 2)
legend('topright', c('Training Data', 'Validation Data', 'Point Estimates', '95% CI'),
  col=c(1, 'grey', 'red', 'red'), lty=c(1,1,1,2), cex=0.75)
```

```r
# Display coefficients and standard errors
output <- capture.output(ts_reg_3)
output <- strsplit(output, '( *: )|[ ]{2,}', perl=TRUE)
output <- output[5:10]
output <- cbind(do.call(rbind, output[1:3]), do.call(rbind, output[4:6]))
tab <- apply(output[-1,-c(1,9)], 1, as.numeric)
rownames(tab) <- c('ar 1', 'ma 1', 'intercept', 'Trend', colnames(ts_vars)[c(2:3, 1, 4, 6:7)])
rownames(tab)[6] <- '(Adjusted Temperature)$^2$'
colnames(tab) <- c('Estimate', 'Std. Error')
xt <- xtable(tab, caption='Time Series Estimated Coefficients', label='tab:ts_reg', digits=2)
print(xt, scalebox=1, sanitize.rownames.function = function(x) {x})

# Calculate final results for model
results[,'ARIMA'] <- c(AIC(ts_reg_3), mean((test[,'Num Deaths'] - fcast$mean)^2))
```

Vector Autoregression

```
# Determining VAR model based on VARselect_res_1
bind <- cbind(trend=train_trend, wday=train[,'Day of Week'], mday=train[,'Day of Month'])
var_fit_1 <- vars::VAR(train_var, p=min(selection[1,]),
  season=NULL, exogen=bind, type='none')

# Display fitted values and original series
ts_fitted <- ts(fitted(var_fit_1), start=c(2002, 1), frequency=365.25)
temp_plot(train_var, other=ts_fitted, other_col='blue', main='')

# Display residuals over time
ts_resid_1 <- ts(resid(var_fit_1), start=c(2002, 1), frequency=365.25)
colnames(ts_resid_1) <- colnames(train_var)
ts_line <- ts(matrix(0, ncol=ncol(train_var), nrow=nrow(train)), start=c(2002,1), frequency=365.25)
temp_plot(ts_resid_1, other=ts_line, other_col='red', main='')
```

```
# Determining VAR model based on VARselect_res_2
var_fit_2 <- vars::VAR(train_var, p=min(selection[2,]),
  season=365, exogen=bind, type='none')

# Display fitted values and original series
ts_fitted <- ts(fitted(var_fit_2), start=c(2002, 1), frequency=365.25)
temp_plot(train_var, other=ts_fitted, other_col='blue', main='')

# Display residuals over time
ts_resid_2 <- ts(resid(var_fit_2), start=c(2002, 1), frequency=365.25)
colnames(ts_resid_2) <- colnames(train_var)
ts_line <- ts(matrix(0, ncol=ncol(train_var), nrow=nrow(train)), start=c(2002,1), frequency=365.25)
temp_plot(ts_resid_2, other=ts_line, other_col='red', main='')
```

```
# Display ACF/CCF plots
acf(ts_resid_1, lag.max=365.25, mar=c(2.85, 2.5, 2, 0.25))
acf(ts_resid_2, lag.max=365.25, mar=c(2.85, 2.5, 2, 0.25))
```

```
# Forecasting VAR(season=NULL)
bind <- cbind(trend=test_trend, wday=test[,'Day of Week'], mday=test[,'Day of Month'])
fcast <- predict(var_fit_1, n.ahead=nrow(test), dumvar=bind)
names(fcast$fcst) <- colnames(train_var)
colnames(fcast$endog) <- colnames(train_var)

plot(train[, 'Num Deaths'], ylab='Number of Deaths', xlim = c(2002, 2007),
  ylim=range(fcast$fcst$`Num Deaths`[, 1:3], ts_vars[, 'Num Deaths']))
lines(test[, 'Num Deaths'], col='darkgrey')

fcast$fcst$`Num Deaths` <- ts(fcast$fcst$`Num Deaths`, start=c(2006, 1), frequency=365.25)
lines(fcast$fcst$`Num Deaths`[,2], col='red', lty=2)
lines(fcast$fcst$`Num Deaths`[,1], col='red')
```

```
lines(fcast$fcst$`Num Deaths`[,3], col='red', lty=2)
legend('topright', c('Training Data', 'Validation Data', 'Point Estimates', '95% CI'),
  col=c(1, 'grey', 'red', 'red'), lty=c(1,1,1,2), cex=0.75)

# Calculate final results for var_fit_1
results[, 2] <- c(AIC(var_fit_1), mean((test[,'Num Deaths'] - fcast$fcst$`Num Deaths`[,'fcst'])^2))

# Forecasting VAR(season=365)
fcast <- predict(var_fit_2, n.ahead=nrow(test), dumvar=bind)
names(fcast$fcst) <- colnames(train_var)
colnames(fcast$endog) <- colnames(train_var)

plot(train[, 'Num Deaths'], ylab='Number of Deaths', xlim = c(2002, 2007),
  ylim=range(fcast$fcst$`Num Deaths`[, 1:3], ts_vars[, 'Num Deaths']))
lines(test[, 'Num Deaths'], col='darkgrey')

fcast$fcst$`Num Deaths` <- ts(fcast$fcst$`Num Deaths`, start=c(2006, 1), frequency=365.25)
lines(fcast$fcst$`Num Deaths`[,2], col='red', lty=2)
lines(fcast$fcst$`Num Deaths`[,1], col='red')
lines(fcast$fcst$`Num Deaths`[,3], col='red', lty=2)
legend('topright', c('Training Data', 'Validation Data', 'Point Estimates', '95% CI'),
  col=c(1, 'grey', 'red', 'red'), lty=c(1,1,1,2), cex=0.75)

# Calculate final results for var_fit_2
results[, 3] <- c(AIC(var_fit_2), mean((test[,'Num Deaths'] - fcast$fcst$`Num Deaths`[,'fcst'])^2))
```

```
# Display coefficients for VAR_1
tab <- coef(var_fit_1)$`Num.Deaths`
temp_names <- colnames(train_var)
temp_names[3] <- "(Adjusted Temperature)$^2$"
rownames(tab) <- c(paste0(rep(temp_names, 3), ' lag ',
  rep(1:3, each=ncol(train_var))), 'Trend', 'Day of Week', 'Day of Month')
colnames(tab)[4] <- 'Pr($>|t|$)'
xt <- xtable(tab, digits=c(0, 3, 3, 3, -2), label='tab:var_coef',
  caption='\\texttt{Num Deaths} Coefficients from VAR(3)')
print(xt, scalebox=0.75, sanitize.text.function=function(x) {x})
```

Neural Network Autoregression

```
# Fit a NNAR model
bind <- cbind(trend=train_trend, temp=train[, 'Adjusted Temperature'],
  temp2=train[, '(Adjusted Temperature)^2'], `Ozone`=train[, 'Ozone'],
  `Relative Humidity`=train[, 'Relative Humidity'],
  wday=train[,'Day of Week'], mday=train[,'Day of Month'])
attach(as.data.frame(train))
fit <- nnetar(`Num Deaths`, xreg=bind)
detach(as.data.frame(train))

# Plot fitted values vs training data
```

```
plot(train[,'Num Deaths'], ylab='Number of Deaths')
lines(ts(fitted(fit), start=c(2002, 1), frequency=365.25), col='blue')
legend('topright', c('Original Data', 'Fitted Values'), lty=c(1,1), col=c(1,'blue'))

# Plot residuals over time
plot(resid(fit), ylab='Residuals')
abline(h=0, col='red', lwd=2)
```

```
# Plot QQ plot
qqnorm(resid(fit)); qqline(resid(fit))

# Plot ACF/PACF
invisible(acf2(resid(fit), 365.25, main=''))
```

```
bind <- cbind(trend=test_trend, temp=test[, 'Adjusted Temperature'],
  temp2=test[, '(Adjusted Temperature)^2'], `Ozone`=test[, 'Ozone'],
  `Relative Humidity`=test[, 'Relative Humidity'],
  wday=test[,'Day of Week'], mday=test[,'Day of Month'])

fcast <- forecast(fit, level=95, xreg=bind, PI=T)

fcast$x <- ts(fcast$x, start=c(2002, 1), frequency=365.25)
fcast$mean = ts(fcast$mean, start=c(2006, 1), frequency=365.25)
fcast$upper = ts(fcast$upper, start=c(2006, 1), frequency=365.25)
fcast$lower = ts(fcast$lower, start=c(2006, 1), frequency=365.25)

plot(train[,'Num Deaths'], xlim = c(2002, 2007), ylab='Number of Deaths')
lines(test[,'Num Deaths'], col = 'darkgrey')
lines(fcast$upper, col = 'red', lty = 2)
lines(fcast$mean, col = 'red')
lines(fcast$lower, col = 'red', lty = 2)
legend('topright', c('Training Data', 'Validation Data', 'Point Estimates', '95% CI'),
  col=c(1, 'grey', 'red', 'red'), lty=c(1,1,1,2), cex=0.75)

results[, 'NNAR'] <- c(NA, mean((fcast$mean - test[,'Num Deaths'])^2))
```

```
# Calculate simulations
trend <- time(ts_vars)
bind <- cbind(trend, temp=ts_vars[, 'Adjusted Temperature'],
  temp2=ts_vars[, '(Adjusted Temperature)^2'], `Ozone`=ts_vars[, 'Ozone'],
  `Relative Humidity`=ts_vars[, 'Relative Humidity'],
  wday=ts_vars[,'Day of Week'], mday=ts_vars[,'Day of Month'])

sims <- replicate(10, simulate(fit, nsim=nrow(ts_vars), xreg=bind))
plot(ts_vars[,'Num Deaths'], ylab='Number of Deaths')
ts.plot(sims, col=1:10, ylab='Number of Deaths')
```

Discussion

```
xt <- xtable(results, label='tab:results',
  caption='Evaluation Results Across all Models')
print(xt, NA.string='NA', sanitize.rownames.function=function(x) {x})
```