Time Series Forecast for Ozone Level: Applications of Exponential Smoothing & ARIMA models

Anh Duc Le

Cork, Republic of Ireland

May 2020

Table of Contents

1. Introduction

Time series forecast is one of the most common tasks for many industries and sectors. There are countless attempts of decoding time series to get better understandings as well as to provide reliable forecasts. Particularly, exponential smoothing and ARIMA models are the two most widely used approaches for time series forecasting.

This report will employ those two approaches for making predictions of ground ozone levels. The ozone dataset, which is obtained from the UCI Machine Learning Repository via this link (Dua & Graff, 2019), features 73 variables documenting daily information from 1st January 1998 to 31st December 2004. The report focuses on making predictions for a subsection of 11 variables including WSR_PK, T_PK, T_AV, T85, RH85, HT85, T70, KI, TT, SLP and SLP_. The details of those 11 variables are documented in Appendix 1. Accordingly, the report will be constructed as follows: A.2. Theoretical Background, A.3. Methodology, A.4. Results & Evaluations, and A.5. Conclusion.

2. Theoretical Background

2.1. ARIMA

Autoregressive Integrated Moving Average (ARIMA) model is a generalisation of an Autoregressive Moving Average (ARMA) model being fitted to data to capture and predict stationary time series. ARIMA, on another hand, is used for non-stationary time series. ARIMA can be broken down into:

- "Integrated" I: this is the initial differencing step applied one or more times to eliminate the non-stationarity.
- "Autoregressive" AR: it is the forecast of the variable of interest by using a linear combination of past values of the variable. In another word, the variable of interest is regressed on its own lagged/prior values. AR at order p or AR(p) can be formularised as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

Where y_t is the variable of interest at time t; $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$ are the variable's prior values; ; $\phi_{t-1}, \phi_{t-2}, \ldots, \phi_{t-p}$ are the coefficients; c is the constant & ε_t is the white noise at time t.

- "Moving Average" MA: it is the forecast of the variable of interest by using a linear combination of past forecast white noise errors of the variable. MA at order q or MA(q) can be formularised as:

$$y_t = c \ + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

Where y_t is the variable of interest at time t; $\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$ are the variable's white noises; ; $\theta_{t-1}, \theta_{t-2}, \ldots, \theta_{t-p}$ are the coefficients & c is the constant.

Taken together, non-seasonal ARIMA(p, d, q) model can be constructed as:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Where y'_t is the differenced series (it may have been differenced more than one), the "predictors" on the right-hand side include both lagged values of y_t and lagged errors. It is the ARIMA(p, d, q) where:

- $p =$ order of the autoregressive part
- $d =$ degree of first differencing involved
- $q =$ order of the moving average part.

Due to the complexity of the lags as well as the strong presence of noise in the daily time series being investigated, seasonal ARIMA is not discussed nor applied in this study. It is consolidated by Hyndman and Athanasopoulos (2018) about the complexity of long daily time series that dynamic harmonic regression or TBATS dealing with multiple seasonalities are more suitable approaches. Therefore, the study is limited to applying non-seasonal ARIMA together with non-seasonal exponential smoothing methods which are discussed in the following section.

2.2. Exponential Smoothing

Exponential smoothing is the univariate time series forecast methods which use weighted averages of past observations with the weights decaying exponentially as the observations getting older. In another word, the more recent the observation the higher the associated weight. Table 1 displays common exponential smoothing methods. Due the complexity of the seasonality in the time series being investigated, exponential smoothing methods associated with seasonal components are not used in this study. Rather, non-seasonal exponential smoothing methods, i.e. Simple Exponential Smoothing, Holt's Linear Method & Additive Damped Trend Method, are applied and, thus, discussed as follows.

Trend Component	**Seasonal Component**		
	None	*Additive*	*Multiplicative*
None	Simple Exponential Smoothing		
Additive	Holt's Linear Method	Additive Holt-Winter's Seasonal Method	Multiplicative Holt-Winter's Seasonal Method
Additive Damped	Additive Damped Trend Method		Damped Multiplicative Holt-Winter's Method

Table 1. Common Exponential Smoothing Methods

2.2.1. Simple Exponential Smoothing

Simple Exponential Smoothing, which is the simplest form of exponential smoothing, is suitable for forecasting time series without clear trend or seasonality. Simple Exponential Smoothing is formularised as:

Forecast equation $\hat{y}_{t+h|h} = \ell_t$

Level equation $\ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1}$

Where y_t is the current observation at time t; $\hat{y}_{t+h|h}$ is the forecast h steps ahead given data up to time t; ℓ_t is the level (or the smoothed value) of the time series at time t;

$0 \leq \alpha \leq 1$ is how much weight is placed on the most recent observation or how quickly the weight decays away.

Large α means more weight is placed on the most recent observation and the weights decay away very quickly; and vice versa. The optimal weight α and the initial level component ℓ_0 are chosen by minimising the sum of squared errors $SSE = \sum_{t=1}^{T}\left(y_t - \hat{y}_{t+h|h}\right)^2 = \sum_{t=1}^{T} e_t^2$.

2.2.2. Holt's Linear Method

Holt's linear method is similar to simple exponential smoothing but with trend component. This method is formularised as:

$$\begin{aligned} \text{Forecast equation} \quad & \hat{y}_{t+h|h} = \ell_t + hb_t \\ \text{Level equation} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation} \quad & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \end{aligned}$$

The Forecast equation is now the linear function of the forecast horizon giving the trended forecasts with slope b_t. The Level equation is similar to simple exponential smoothing which, however, is adjusted to allow time series being trended. The Trend equation describes how the slope changes over time. This phenomenon of changing slope is called "Local linear trend". Value $0 \leq \beta \leq 1$ controls how quickly the slope can change. Large β means the slope changes rapidly, allowing for highly nonlinear trend; and vice versa. Choosing $\alpha, \beta, \ell_0, b_0$ to minimise SSE.

2.2.3. Additive Damped Trend Method

Unlike the Holt's Linear Method allowing the trend to continue indefinitely into the future, the Additive Damped Trend Method, which is the modified Holt's Linear Method, allows trend to dampen over time so that trend will level off to a constant value. The modification is the introduction of parameter ϕ controlling the damping. It turns out the short-run forecasts are trended, and long-run forecasts are constant.

$$\begin{aligned} \text{Forecast equation} \quad & \hat{y}_{t+h|h} = \ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t \\ \text{Level equation} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1}) \end{aligned}$$

Trend equation $b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)\phi b_{t-1}$

Where $0 \le \phi \le 1$. The larger the value of ϕ, the less damping there is. With $\phi = 1$, the method is equivalent to Holt's Linear Method. Choosing $\alpha, \beta, \phi, \ell_0, b_0$ to minimise SSE

3. Methodology

This section includes data gathering (A.3.1), data pre-processing (A.3.2) & the suggested protocol of forecasting the time series (A.3.3).

3.1. Data gathering

With the given files "Assignment Data.csv" & "Assignment Data Details.txt", Figure 1 shows how the dataframe is loaded. There are 73 variables with 2534 observations. The variables/columns, which however are not in the right formats, are pre-processed in the following step[1].

	Date	WSR0	WSR1	WSR2	WSR3	WSR4	WSR5	WSR6	WSR7	WSR8	WSR9	WSR10
1	1/1/1998	0.8	1.8	2.4	2.1	2	2.1	1.5	1.7	1.9	2.3	3.7
2	1/2/1998	2.8	3.2	3.3	2.7	3.3	3.2	2.9	2.8	3.1	3.4	4.2
3	1/3/1998	2.9	2.8	2.6	2.1	2.2	2.5	2.5	2.7	2.2	2.5	3.1
4	1/4/1998	4.7	3.8	3.7	3.8	2.9	3.1	2.8	2.5	2.4	3.1	3.3
5	1/5/1998	2.6	2.1	1.6	1.4	0.9	1.5	1.2	1.4	1.3	1.4	2.2
6	1/6/1998	3.1	3.5	3.3	2.5	1.6	1.7	1.6	1.6	2.3	1.8	2.5
7	1/7/1998	3.7	3.2	3.8	5.1	6	7	6.3	6.4	6.3	5.4	6.1
8	1/8/1998	2.2	2.9	3.4	4.2	4.7	4.7	5.3	4.9	5.2	6	5.9
9	1/9/1998	1	1.5	1.2	1.2	0.7	0.5	1.2	1.4	1.5	2.1	2.6
10	1/10/1998	0.9	0.6	0.5	0.5	0.6	0.4	0.4	0.6	1.3	1.5	2.8
11	1/11/1998	1.1	1.7	1.4	1.5	0.9	1.5	1.4	1.6	1.9	1.9	2.7

Figure 1. Loaded Dataframe (more columns not being shown)

3.2. Data Pre-processing

All of "?" are replaced by missing values NA. All of columns except for Date are converted into numeric columns[2].

Then, the correlation relationships between variables is examined. Figure 2, which shows the linear correlation relationships between 72 numeric variables, indicate most of

1 R Code: A.3.1. Data Gathering
2 R Code: A.3.2.1

the positively correlated variables in a systematic manner, i.e. two big blue squares/clusters of variables. The highly correlated relationships suggest dealing with missing values should be done by wholly considering all of the variables rather than treating missing values for each of the variables individually[3].

All of the 72 numeric variables have missing values. The detailed number of missing values for each of the variables is reported in the Appendix 2. The variable WSR0 has the most missing values with 299 missing observations. The variable Precp has the least missing values with 2 missing observations. Figure 3 shows the number of missing values for each of the variables and the combinations of missing values for each of the observations having missing values[4].

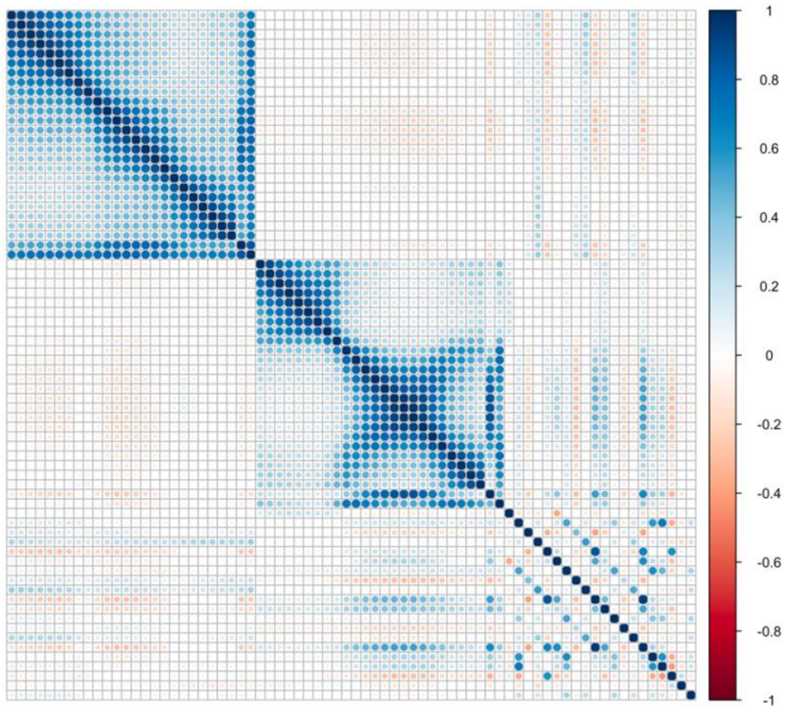

Figure 2. Correlation Matrix between 72 numeric variables

[3] R Code: A.3.2.2
[4] R Code: A.3.2.3

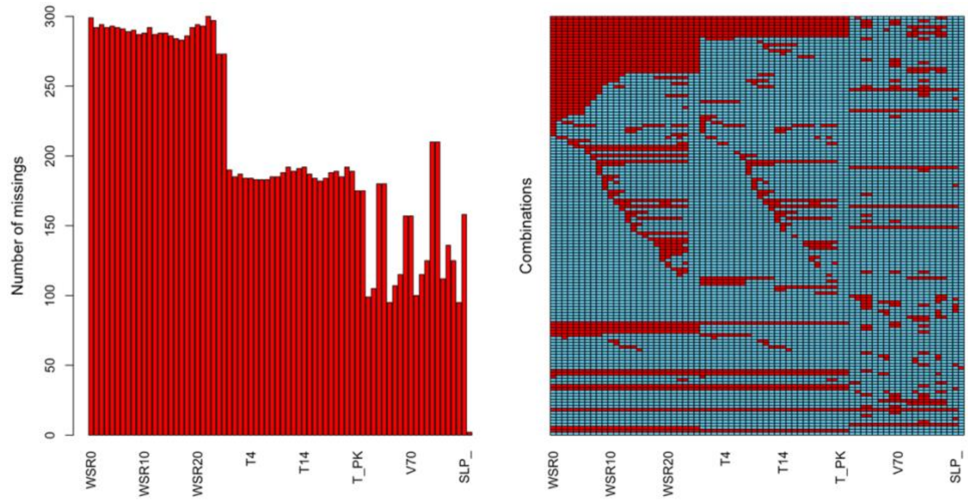

Figure 3. Missing values

The missing values are then imputed by applying Predictive Mean Matching (PMM) is a semi-parametric imputation approach supported by the mice package in R. PMM is similar to the regression method except that for each missing value, it fills in a value randomly from among the observed donor values from an observation whose regression-predicted values are closest to the regression-predicted value for the missing value from the simulated regression models[5]. Thus, the missing values are estimated from other variables' values. Keeping other variables than the chosen ones is needed to comprehensively deal with the missing values.

Examining the Date column, the starting date is 1/1/1998, and the ending date is 31/12/2004. Using the column Date as the index, the dataframe is turned into "xts" object[6] which will be used for cross validations instead of the time series objects. Although results of cross validations using "xts" or time series objects are the same, the former promotes much faster results than the latter.

Eleven variables of interest are WSR_PK, T_PK, T_AV, T85, RH85, T70, KI, TT, SLP & SLP_. Those variables are now converted into time series by using function **ts(variable_name, start = c(1998,1,1), frequency = 365.25)**. To simplify the names, the corresponding time series for those 11 variables above are named as var1, var2…, var11 respectively[7].

5 R Code: A.3.2.4
6 R Code: A.3.2.5
7 R Code: A.3.2.6

3.3. Time Series Forecast Protocol

The protocol of analysing and predicting the time series **var** having its name as **var_name** is as follows[8]:

1. Plotting & summarising descriptive statistics of the time series.

```r
# summary
descr(var)
# histogram
hist(var,
     probability = T,
     main = paste('Histogram of ', var_name),
     xlim = c(min(var) - sd(var), max(var) + sd(var)),
     col = 'grey93')
lines(density(var), col='red')
# time series plot
plot(var, main = paste('Time Series ', var_name), ylab = "variable's unit")
abline(tslm(var~time(var)), col='red')
legend('topleft',
       legend=c(var_name, "Fitted Regression Line"),
       lty=1,
       col = c('black', 'red'),
       cex=0.8)
```

2. Firstly, checking whether the time series is additive model or multiplicative one by examining its autocorrelation of the random elements for both models. The sum of squared coefficients is used to choose the correct model which has the smaller sum of squared coefficients. This is because the correct model should be able to identify the trend and seasonality well so the remaining random is supposed not be correlated. Secondly, after that, the time series is decomposed with the right model so that trend, seasonality or even cycle could be detected.

```r
# Checking additive or Multiplicative
sum_squared_corr_add <- sum((acf(decompose(var, type = 'additive')$random, na.action = na.omit)$acf)^2)
sum_squared_corr_add
sum_squared_corr_multi <- sum((acf(decompose(var, type = 'multiplicative')$random, na.action = na.omit)$acf)^2)
sum_squared_corr_multi
if(sum_squared_corr_add < sum_squared_corr_multi){
  type <- 'additive'
}else{
  type <- 'multiplicative'
}
type
# Decomposing time series
plot(decompose(var, type))
```

3. Examining autocorrelation of the time series by plotting auto-correlation function (ACF) plot.

```r
# Autocorrelation
ggAcf(var, lag = length(var)) + theme_classic() + ggtitle(paste(var_name,'Autocorrelation'))
```

[8] R Code: The whole section A.4.2. "Gathering results"

4. Fitting simple exponential smoothing:
    a. Automatically getting optimal parameters including smoothing α and initial state ℓ_0 by using the function **ses( )**.
    b. Plotting one-year prediction and fitted values
    c. Checking whether residuals of prediction are white noise – Ljung-Box test.
    d. Calculating cross-validated mean squared error

```r
# fitting the simple exponential smoothing model with 1-year prediction
var_ses <- ses(var, h=365)
# getting the model's parameters
var_ses$model
# plotting the model's prediction and its fitted values
autoplot(var_ses) +
  autolayer(fitted(var_ses)) +
  theme_classic()
# checking model's residuals - Ljung-Box test
checkresiduals(var_ses)
# 1-step cross validation
var_ses_error <- tsCV(df[,colnames(df) == var_name], ses, h =1)
# cross-validated mean squared error
mean(var_ses_error^2, na.rm = TRUE)
```

5. Fitting Holt's Linear Method:
    a. Automatically getting optimal parameters including smoothing α, β and initial states ℓ_0, b_0 by using the function **holt( )**.
    b. Plotting one-year prediction and fitted values
    c. Checking whether residuals of prediction are white noise – Ljung-Box test.
    d. Calculating cross-validated mean squared error

```r
# fitting the holt's linear method with 1-year prediction
var_holt <- holt(var, h=365)
# getting the model's parameters
var_holt$model
# plotting the model's prediction and its fitted values
autoplot(var_holt) + autolayer(fitted(var_holt)) + theme_classic()
# checking model's residuals - Ljung-Box test
checkresiduals(var_holt)
# 1-step cross validation
var_holt_error <- tsCV(df[,colnames(df) == var_name], holt, h =1)
# cross-validated mean squared error
mean(var_holt_error^2, na.rm = TRUE)
```

6. Fitting Additive Damped Trend Method:
    a. Automatically getting optimal parameters including smoothing α, β, ϕ and initial states ℓ_0, b_0 by using the function **holt(, damped = T)**.
    b. Plotting one-year prediction and fitted values.
    c. Checking whether residuals of prediction are white noise – Ljung-Box test.
    d. Calculating cross-validated mean squared error.

```
# fitting damped trend method with 1-year prediction
var_dam <- holt(var, h=365, damped = T)
# getting the model's parameters
var_dam$model
# plotting the model's prediction and its fitted values
autoplot(var_dam) + autolayer(fitted(var_dam)) + theme_classic()
# checking model's residuals - Ljung-Box test
checkresiduals(var_dam)
# 1-step cross validation
var_dam_error <- tsCV(df[,colnames(df) == var_name], holt, damped=T, h =1)
# cross-validated mean squared error
mean(var_dam_error^2, na.rm = TRUE)
```

7. Fitting ARIMA model:
    a. Automatically finding constant, AR factors, MA factors & I, i.e. differencing, factors by using the function **auto.arima( )**.
    b. Plotting one-year prediction and fitted values
    c. Checking whether residuals of prediction are white noise – Ljung-Box test.
    d. Calculating cross-validated mean squared error

```
# fitting arima model
var_arima <- auto.arima(var)
# getting the model's parameters
var_arima
# plotting the model's 1-year prediction and its fitted values
autoplot(forecast(var_arima, h=365)) + autolayer(fitted(var_arima)) + theme_classic()
# checking model's residuals - Ljung-Box test
checkresiduals(forecast(var_arima, h=365))
# 1-step cross validation
var_arima_error <- tsCV(df[,colnames(df) == var_name], farima, h = 1)
# cross-validated mean squared error
mean(var_arima_error^2, na.rm = TRUE)
```

8. Choosing the best model having the smallest cross-validated mean squared errors.

This process is repeated for the 11 chosen variables. The results are documented in the following section.

4. Results & Evaluations

Based on the protocol in section A.3.3, this section includes a demonstration of analysing and predicting the time series WSR_PK (var1) (A.4.1) and the summary of the analysis and forecast for all of the 11 investigated time series (A.4.2). This would reduce the repetitions of individually presenting results of the 11 time series objects, but rather synthesising the key information in the summary section.

4.1. WSR_PK (var1)

4.1.1. Descriptive Statistics Summary

Figure 4 and Table 2 provides a summary of descriptive statistics for WSR_PK. The variable has a unimodal distribution which is positively skewed. The time series plot indicates a large volatility together with a large amount of noise presenting in the time series. There is a faint annual seasonality. No trend can be explicitly recognised although the fitted regression line might indicate a general downward trend. No clear cycle could be detected.

4.1.2. Decomposing Time Series

After examining the autocorrelation of the random, the time series WSR_PK is the additive model. Additively decomposing the time series, Figure 5 shows that the overall trend is decreasing, the seasonality is annual, the variance of the random is however heteroscedastic. In fact, decomposing could not comprehensively explain the time series without compromising the random term. Decomposing might be suitable for briefly recognising overall trend or seasonality only.

4.1.3. Autocorrelation

Figure 6 indicates a damped sine wave having 1-year frequency for the autocorrelation. It might suggest the time series might repeat annually as its seasonality pattern. Furthermore, there might be a "long-lasting effect" presenting in the time series since the pattern continues to repeat in several years after the first observation being recorded.

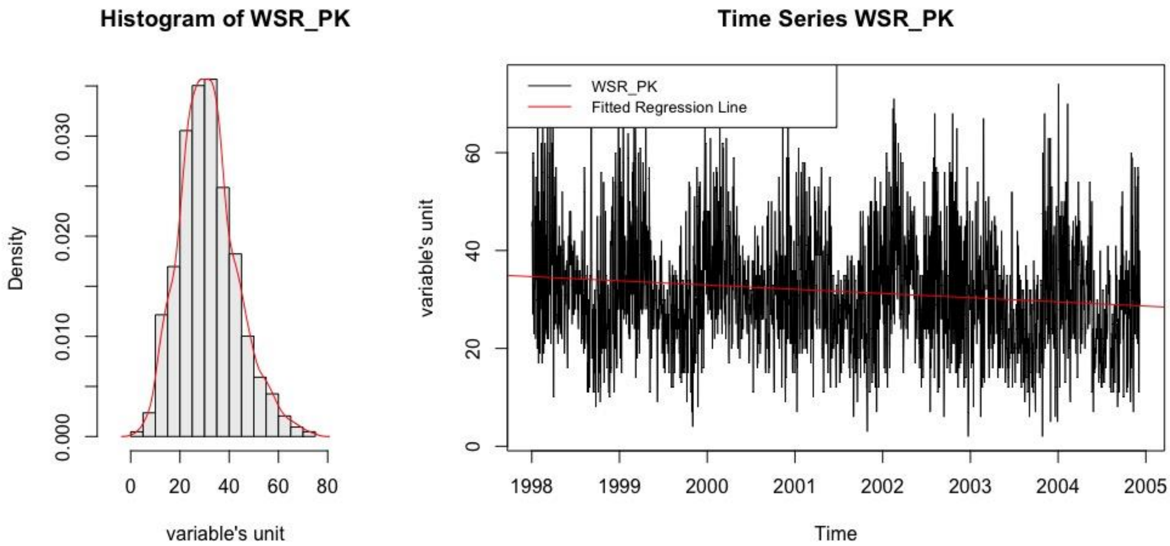

Figure 4. Histogram and Time series plot – WSR_PK

Period	**No. observations / Missing** (counts)	**Skewness** (no unit)	**Amount of Variability** (no unit)
1/1/98 – 31/12/04	2534 / 0	0.52	0.37
Min (variable's unit)	**Mean** (variable's unit)	**Max** (variable's unit)	**Standard Deviation** (variable's unit)
2.00	31.67	75.00	11.74
First Quartile Q1 (variable's unit)	**Median Q2** (variable's unit)	**Third Quartile Q3** (variable's unit)	**Inter-Quartile Range** (variable's unit)
24.00	31.00	38.00	14.00

Table 2. Descriptive Statistics Summary – WSR_PK

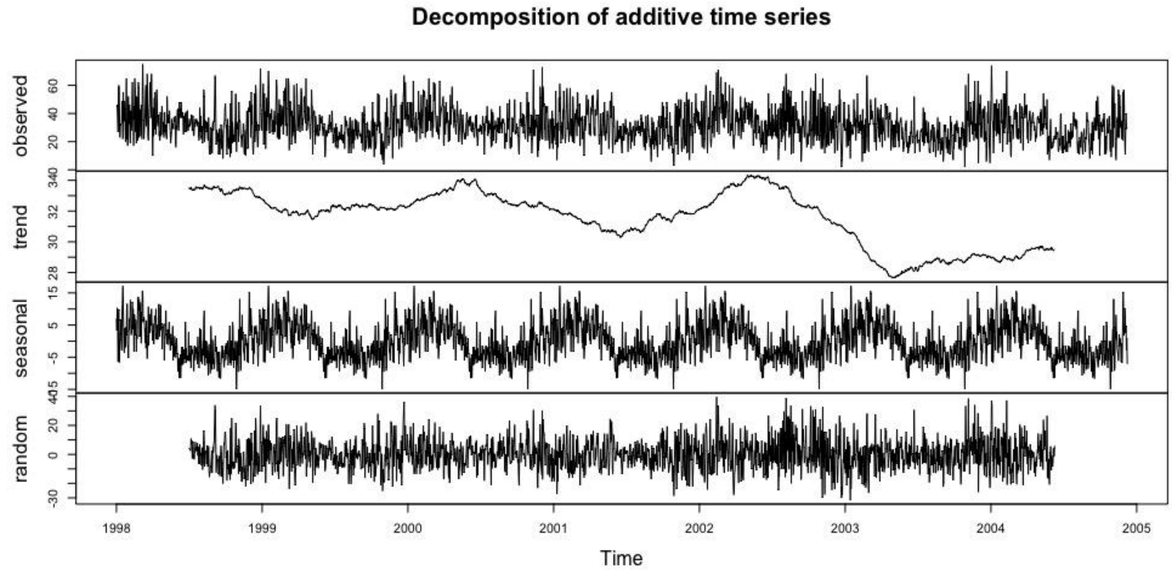

Figure 5. Time series decomposition - WSR_PK

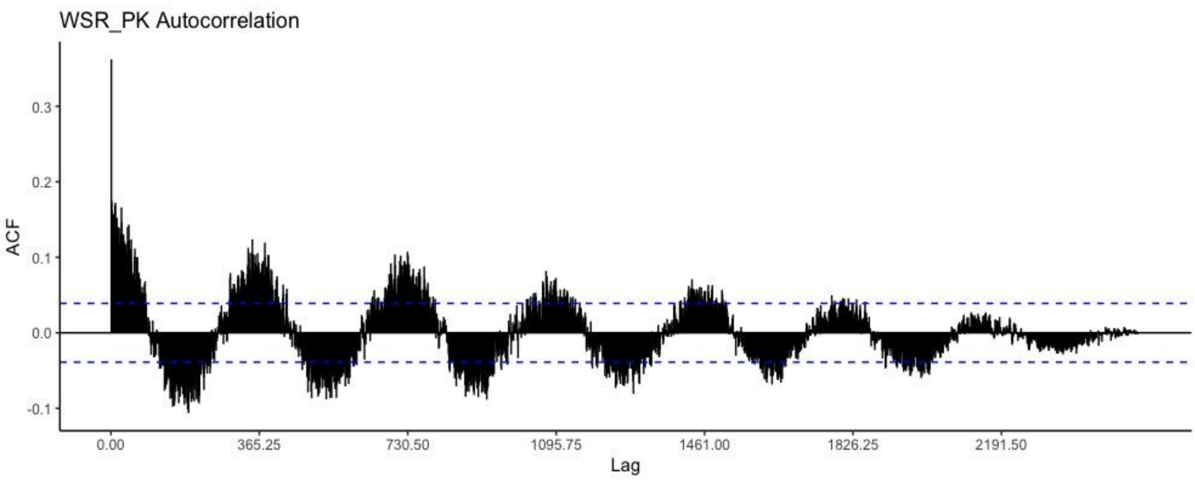

Figure 6. Autocorrelation - WSR_PK

4.1.4. Simple Exponential Smoothing

Fitting the simple exponential smoothing to the time series, the smoothing α is 0.0757 and the initial state ℓ_0 is 40.3507. Performing Ljung-Box test for the prediction's residuals, the p-value of 2.564e-07 suggests that the residuals are not the white noise. In another word, the model could not fully capture the dynamics of the time series. Besides, implementing 1-step cross validation, the cross-validated mean squared error is 121.0784. This value will be compared with other models' cross-validated mean squared errors so that the best model having the smallest cross-validated mean squared error could be chosen.

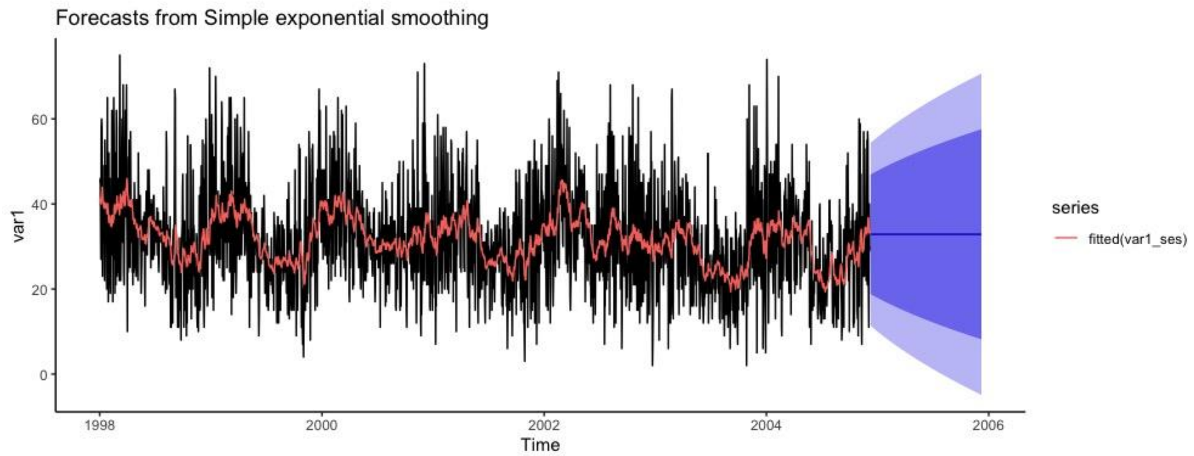

Figure 7. Simple Exponential Smoothing - WSR_PK

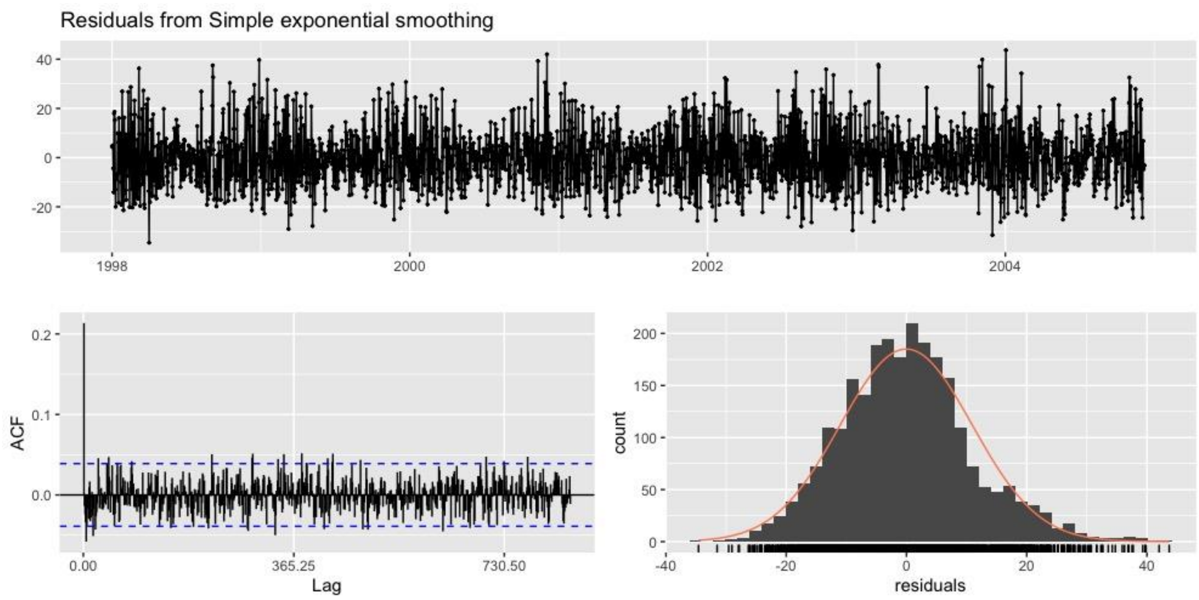

Figure 8. Prediction's residuals analysis

4.1.5. Holt's Linear Trend

Fitting the Holt's Linear Trend to the time series, the smoothing α & β are 0.0876 and 0.0001 respectively; the initial state ℓ_0 & b_0 are 47.4961 and -0.0146 respectively. Performing Ljung-Box test for the prediction's residuals, the p-value of 2.933e-07 suggests that the residuals are not the white noise. In another word, the model could not fully capture the dynamics of the time series. Besides, implementing 1-step cross validation, the cross-validated mean squared error is 122.5736. This value will be compared with other models' cross-validated mean squared errors so that the best model having the smallest cross-validated mean squared error could be chosen.

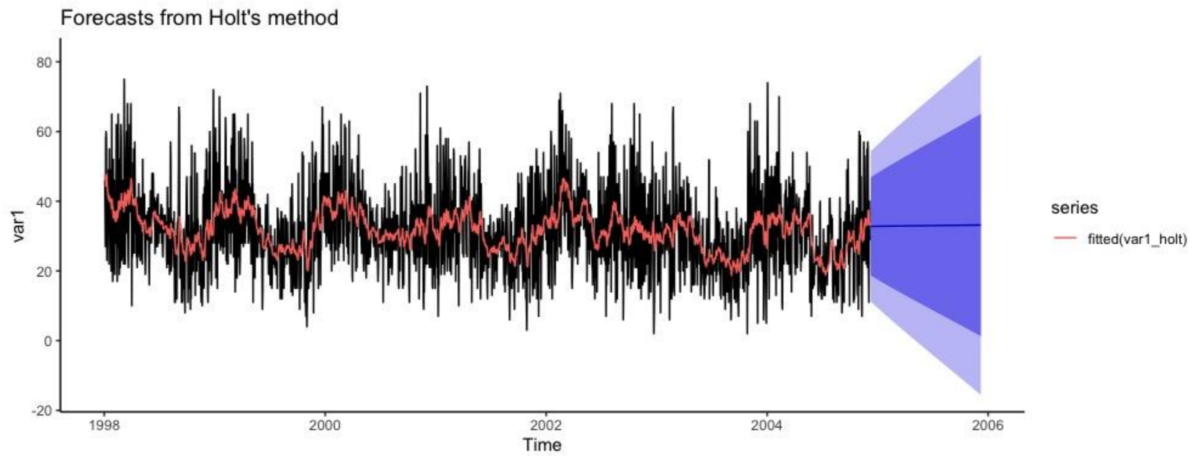

Figure 9. Holt's Linear Method - WSR_PK

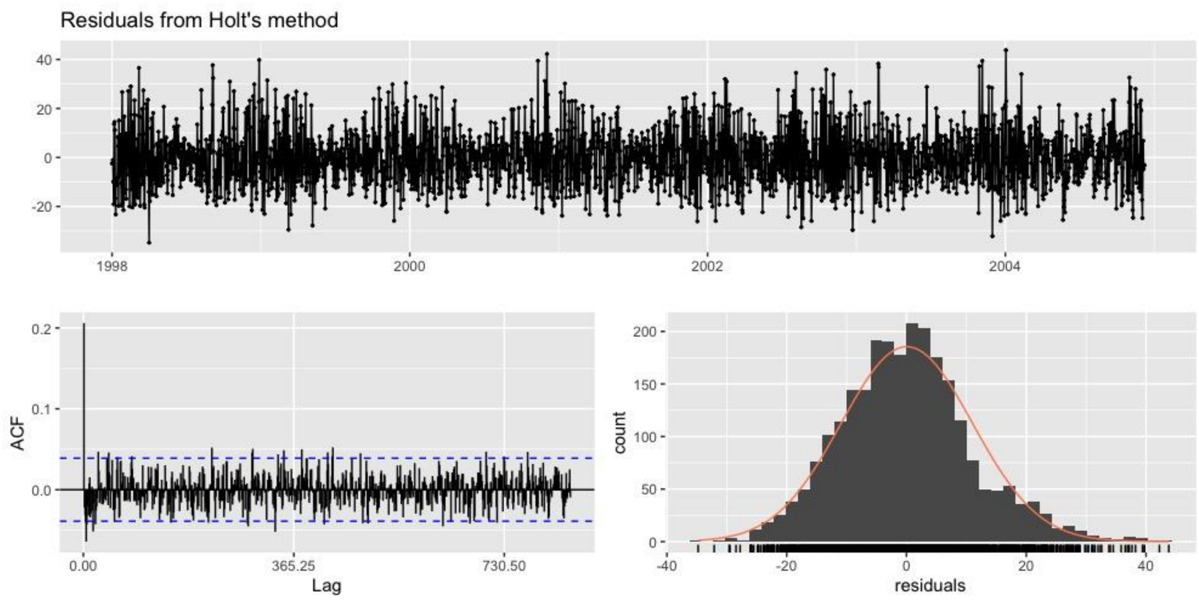

Figure 10. Prediction's residuals analysis

4.1.6. Additive Damped Method

Fitting the Holt's Linear Trend to the time series, the smoothing α, β & ϕ are 0.0769, 0.0001 and 0.8 respectively; the initial state ℓ_0 & b_0 are 47.319 and -0.6592 respectively. Performing Ljung-Box test for the prediction's residuals, the p-value of 1.796e-07 suggests that the residuals are not the white noise. In another word, the model could not fully capture the dynamics of the time series. Besides, implementing 1-step cross validation, the cross-validated mean squared error is 122.0876. This value will be compared with other models' cross-validated mean squared errors so that the best model having the smallest cross-validated mean squared error could be chosen.

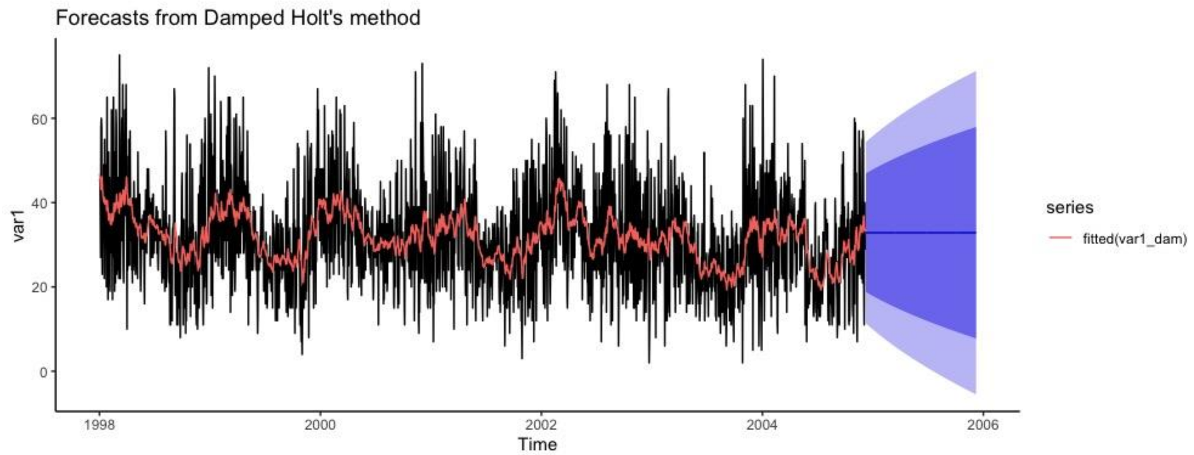

Figure 11. Additive Damped Method - WSR_PK

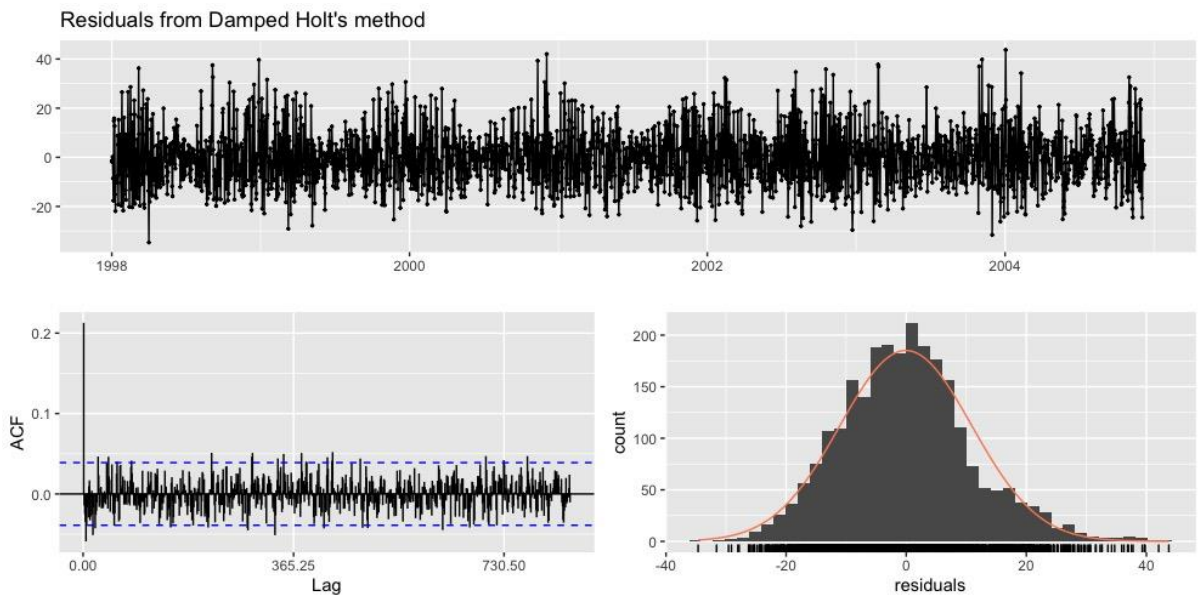

Figure 12. Prediction's residuals analysis

4.1.7. ARIMA model

Fitting the ARIMA to the time series, the model is ARIMA(0,1,2) including one differencing, MA1 of -0.7069 and MA2 of -0.2345. Performing Ljung-Box test for the prediction's residuals, the p-value of 0.1304 suggests that the residuals are the white noise. In another word, the model could somehow fully capture the dynamics of the time series. Besides, implementing 1-step cross validation, the cross-validated mean squared error is 115.4132. This value will be compared with other models' cross-validated mean squared errors so that the best model having the smallest cross-validated mean squared error could be chosen.

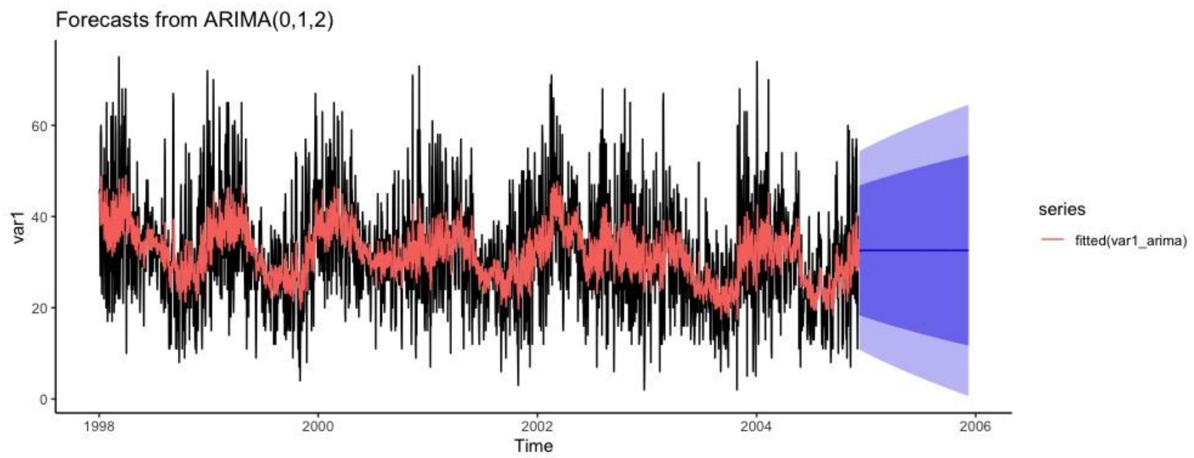

Figure 13. ARIMA model - WSR_PK

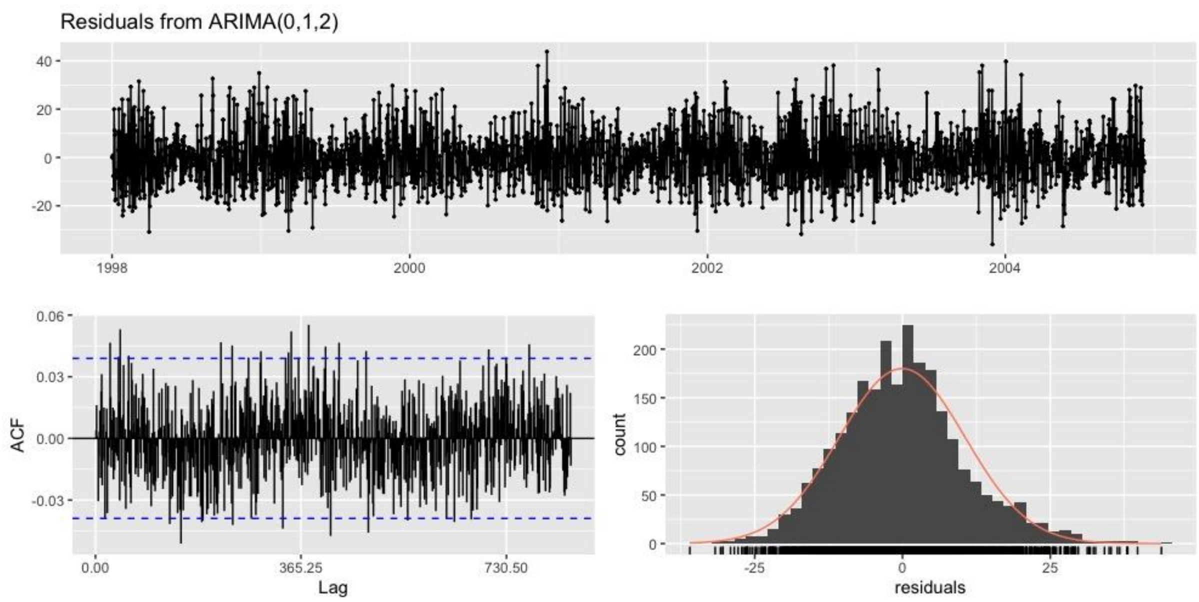

Figure 14. Prediction's residuals analysis

4.1.8. The best model

Comparing the cross-validated mean squared errors, the best model is ARIMA featuring white noise residuals derived from the model's prediction (Table 3).

	Simple Exponential Smoothing	**Holt's Linear Trend**	**Additive Damped Method**	**ARIMA model**
Are prediction's residuals white noise?	No	No	No	Yes
Cross-validated Mean Squared Error	121.0784	122.5736	122.0876	115.4132

Table 3. Models' performance comparison – WSR_PK

4.2. Gathering results

4.2.1. Descriptive Statistics Summary

Table 4 summarises descriptive statistics for each of the time series. All of the time series have moderately large coefficients of variability indicating the extent of variability of the time series in relation to their means. In another word, the time series have much of the volatility. The skewness values vary from the time series to another. Modalities of the time series' distributions are also different.

	Mean*	SD*	Median*	IQR*	Coeff. of Varia.	Skewn.	Skewn. of Dist.	Mod. of Dist.
WSR_PK (Var1)	31.67	11.74	31.00	14.00	0.37	0.52	Positive	Unimodal
T_PK (Var2)	165.61	68.30	169.00	96.00	0.41	-0.23	Negative	Unimodal
T_AV (Var3)	146.86	68.90	154.00	101.00	0.47	-0.04	Negative	Bimodal
T85 (Var4)	115.22	61.24	103.00	62.00	0.53	0.82	Positive	Bimodal
RH85 (Var5)	58.55	25.70	64.00	41.00	0.44	-0.49	Negative	Unimodal
HT85 (Var6)	211.67	69.66	218.00	91.75	0.33	-0.52	Negative	Unimodal
T70 (Var7)	181.06	59.86	201.00	87.75	0.33	-1.22	Negative	Unimodal
KI (Var8)	577.39	293.05	624.00	469.50	0.51	-0.34	Negative	Bimodal
TT (Var9)	370.17	156.32	409.00	222.75	0.42	-0.53	Negative	Unimodal
SLP (Var10)	31.95	10.57	31.00	13.00	0.33	0.54	Positive	Unimodal
SLP_ (Var11)	26.63	15.56	29.00	29.00	0.58	0.07	Positive	Multimodal

Table 4. Descriptive Statistics Summary | * indicates the time series' unit

4.2.2. Decomposing Time Series

Table 5 indicates the different elements of decomposition. There is a mix of additive and multiplicative models. Majority of the time series does not have a clear trend. Meanwhile, most of the time series have annual seasonality. All of the time series do not exhibit any clear cycle.

	Additive/ Multiplicative	**Trend**	**Season**	**Cycle**
WSR_PK (Var1)	Additive	Decreasing	Annual	Unclear
T_PK (Var2)	Multiplicative	Decreasing	Annual	Unclear
T_AV (Var3)	Multiplicative	Unclear	Annual	Unclear
T85 (Var4)	Multiplicative	Unclear	Annual	Unclear
RH85 (Var5)	Multiplicative	Unclear	Annual	Unclear
HT85 (Var6)	Multiplicative	Unclear	Annual	Unclear
T70 (Var7)	Additive	Unclear	Annual	Unclear
KI (Var8)	Multiplicative	Increasing	Annual	Unclear
TT (Var9)	Additive	Unclear	Annual	Unclear
SLP (Var10)	Additive	Unclear	Annual	Unclear
SLP_ (Var11)	Multiplicative	Periodic	Unclear	Unclear

Table 5. Decomposing Time Series

4.2.3. Autocorrelation

Table 6 shows autocorrelation for each of the time series. Apart from SLP_ (var11), all of the time series have strong magnitudes of autocorrelations whose patters are the damped sine waves of 1-year frequency.

	Magnitude	**Pattern**
WSR_PK (Var1)	Strong	Damped sine wave with 1-year frequency
T_PK (Var2)	Strong	Damped sine wave with 1-year frequency
T_AV (Var3)	Strong	Damped sine wave with 1-year frequency
T85 (Var4)	Strong	Damped sine wave with 1-year frequency
RH85 (Var5)	Strong	Damped sine wave with 1-year frequency
HT85 (Var6)	Strong	Damped sine wave with 1-year frequency
T70 (Var7)	Strong	Damped sine wave with 1-year frequency
KI (Var8)	Strong	Damped sine wave with 1-year frequency
TT (Var9)	Strong	Damped sine wave with 1-year frequency
SLP (Var10)	Strong	Damped sine wave with 1-year frequency
SLP_ (Var11)	Moderate	Tailing off

Table 6. Autocorrelation

4.2.4. Simple Exponential Smoothing

Table 7 shows the results of fitting simple exponential smoothing models to the time series. All of the models cannot capture the dynamics of the time series as the predictions' residuals are still autocorrelated as suggested by the Ljung-Box tests.

	Model's Parameters		Ljung-Box test for residuals		Cross validated Mean Squared Error
	Smoothing α	Initial state ℓ_0	p-value	Are residuals white noise?	
WSR_PK (Var1)	0.0757	40.3507	2.564e-07	No	121.0784
T_PK (Var2)	0.1755	103.1731	6.278e-11	No	2368.2130
T_AV (Var3)	0.1208	106.6004	1.066e-14	No	3685.27
T85 (Var4)	0.2204	190.535	5.331e-11	No	3364.612
RH85 (Var5)	0.0795	57.1864	2.2e-16	No	613.0315
HT85 (Var6)	0.9999	192.0312	2.2e-16	No	2578.639
T70 (Var7)	0.2989	90.4082	2.485e-09	No	2442.25
KI (Var8)	0.0889	486.604	4.328e-05	No	71843.45
TT (Var9)	0.0829	380.2271	1.308e-09	No	21439.89
SLP (Var10)	0.8946	63.6928	2.2e-16	No	68.76217
SLP_ (Var11)	0.0001	26.6648	2.2e-16	No	242.2227

Table 7. Simple Exponential Smoothing

4.2.5. Holt's Linear Trend

Table 8 shows the results of fitting Holt's Linear Trends to the time series. All of the models cannot capture the dynamics of the time series as the predictions' residuals are still autocorrelated as suggested by the Ljung-Box tests.

	Model's Parameters		**Ljung-Box test for residuals**		**Cross validated Mean Squared Error**
	Smoothing α, β	Initial states ℓ_0, b_0	p-value	Are residuals white noise?	
WSR_PK (Var1)	0.0876, 0.0001	47.4961, -0.0146	2.933e-07	No	122.5736
T_PK (Var2)	0.1778, 0.0001	119.3219, 0.0235	6.362e-11	No	2385.072
T_AV (Var3)	0.1236, 0.0001	82.0125, 0.0259	9.659e-15	No	3732.688
T85 (Var4)	0.2233, 0.0001	207.9781, -0.0297	8.292e-11	No	3400.211
RH85 (Var5)	0.0859, 0.0001	50.9524, 0.0134	2.2e-16	No	619.0419
HT85 (Var6)	0.9999, 0.0001	369.8738, -0.0397	2.2e-16	No	2618.763
T70 (Var7)	0.2989, 0.0001	117.3141, 0.0413	1.782e-09	No	2450.848
KI (Var8)	0.0912, 0.0001	499.7085, 0.0566	4.168e-05	No	72347.57
TT (Var9)	0.8932, 0.0001	57.9788, -0.009	2.2e-16	No	21572.73
SLP (Var10)	0.8932, 0.0001	57.9788, -0.009	2.2e-16	No	69.35677
SLP_ (Var11)	0.0267, 0.0001	4.9604, -0.0038	2.2e-16	No	249.9622

Table 8. Holt's Linear Trends

4.2.6. Additive Damped Method

Table 9 shows the results of fitting additive damped trend methods to the time series. All of the models cannot capture the dynamics of the time series as the predictions' residuals are still autocorrelated as suggested by the Ljung-Box tests.

	Model's Parameters		**Ljung-Box test for residuals**		**Cross validated Mean Squared Error**
	Smoothing α, β, ϕ	Initial states ℓ_0, b_0	p-value	Are residuals white noise?	
WSR_PK (Var1)	0.0769, 0.0001, 0.8	47.3190, -0.6592	1.796e-07	No	122.0876
T_PK (Var2)	0.1757, 0.0001, 0.8	118.6252, -1.6888	3.946e-11	No	2374.522
T_AV (Var3)	0.1207, 0.0001, 0.9081	83.5149, 2.2625	5.44e-15	No	3705.013
T85 (Var4)	0.2196, 0.0001, 0.8755	213.9951, -7.182	3.79e-11	No	3389.686
RH85 (Var5)	0.0795, 0.0001, 0.8	49.1586, 1.9756	2.2e-16	No	614.5073
HT85 (Var6)	0.9999, 0.0001, 0.9475	372.3005, -19.2421	2.2e-16	No	2592.272
T70 (Var7)	0.2982, 0.0001, 0.8747	117.3337, 1.1486	1.334e-09	No	2456.004
KI (Var8)	0.0886, 0.0001, 0.8166	505.7174, 8.3762	2.601e-05	No	72127.3
TT (Var9)	0.0827, 0.0001, 0.8	350.5833, 7.2635	7.298e-10	No	21503.57
SLP (Var10)	0.8929, 0.0001, 0.9413	58.8839, -2.4697	2.2e-16	No	69.39961
SLP_ (Var11)	0.0001, 0.001, 0.863	3.0619, 3.7472	2.2e-16	No	242.8118

Table 9. Additive Damped Models

4.2.7. ARIMA models

Table 10 shows the results of fitting ARIMA models to the time series. 8 out of 11 models sufficiently capture the dynamics of the time series as the predictions' residuals behave like the white noise as suggested by the Ljung-Box tests.

	Mean + ARIMA(p, d, q)				**Ljung-Box test for residuals**		**Cross validated Mean Squared Error**
	Mean	AR $p_1, p_2, \ldots$	Differencing $d = 1,2, \ldots$	MA $q_1, q_2, \ldots$	p-value	White noise	
WSR_PK (Var1)			1	-0.7069, -0.2345	0.1304	Yes	115.4132
T_PK (Var2)	165.6092	0.4337, 0.1322, 0.0874, 0.0324, 0.1469			2.2e-16	No	2367.296
T_AV (Var3)	147.1542	1.2662, -0.2874, -0.0272, 0.0381		-0.9189	0.03801	No	3449.353
T85 (Var4)	115.6966	0.3581, 0.8589, -0.2362, -0.0731, 0.0307		0.0047, -0.8063	0.1751	Yes	3119.425
RH85 (Var5)		0.3818, -0.1188, 0.0291, -0.0191	1	-0.9614	0.8218	Yes	547.5903
HT85 (Var6)	211.6423	1.0371, 0.2317, -0.4755, 0.2099, -0.0258		-0.1528, -0.6808	0.2597	Yes	2113.373
T70 (Var7)	181.0605	1.3342, -0.3494, 0.0015		-0.8961	0.09803	Yes	2288.481
KI (Var8)	576.9603	1.1782, -0.2173, 0.0561, -0.0323, 0.0048		-0.9303	0.0500	Yes	69823.45
TT (Var9)	368.5817	1.7177, -0.9024, 0.1895, -0.0053, -0.0054		-1.3896, 0.4237	0.1846	Yes	20541.74
SLP (Var10)	32.1242	1.6850, -0.9193, 0.2599, -0.0361		-0.9209	0.2358	Yes	56.56671
SLP_ (Var11)	26.6704	0.8109, -0.3063, 0.0567		-0.6866	3.683e-05	No	226.594

Table 10. ARIMA models

4.2.8. The best models

Table 11 shows the cross-validated mean squared errors (MSE) for different forecast models applied for the time series. The best models are ones having the smallest MSEs. It is clear that ARIMA models outperform all of the exponential smoothing methods. However, there is not a huge difference between Simple Exponential Smoothing's MSE and ARIMA model's MSE for the time series T_PK (var2). To a certain extent, the simple exponential smoothing, which is simpler than the ARIMA, could be chosen for the time series T_PK (var2). Besides, considering the white noise, not all of the ARIMA models could capture sufficiently the dynamics of the time series.

	Simple Exponential Smoothing MSE	**Holt's Linear Trend MSE**	**Additive Damped Trend MSE**	**ARIMA MSE**
WSR_PK (Var1)	121.0784	122.5736	122.0876	115.4132*
T_PK (Var2)	2368.2130	2385.072	2374.522	2367.296
T_AV (Var3)	3685.27	3732.688	3705.013	3449.353
T85 (Var4)	3364.612	3400.211	3389.686	3119.425*
RH85 (Var5)	613.0315	619.0419	614.5073	547.5903*
HT85 (Var6)	2578.639	2618.763	2592.272	2113.373*
T70 (Var7)	2442.25	2450.848	2456.004	2288.481*
KI (Var8)	71843.45	72347.57	72127.3	69823.45*
TT (Var9)	21439.89	21572.73	21503.57	20541.74*
SLP (Var10)	68.76217	69.35677	69.39961	56.56671*
SLP_ (Var11)	242.2227	249.9622	242.8118	226.594

Table 11. Cross-validated MSEs | * indicates white noise residuals

5. Conclusion

The report sums up how to implement time series forecasts by employing exponential smoothing methods and ARIMA models. Due to the nature of the time series, only non-seasonal exponential smoothing methods and non-seasonal ARIMA are used. The results suggest that ARIMA models outperform exponential smoothing methods. However, not all ARIMA models can efficiently capture the dynamics of the time series that the residuals generated by the predictions of the models are still autocorrelated. There might be two possible scenarios associated with the problem. First, the models could be wrong. Second, the models could be somewhat right; but failing to pick up the time series' signals that some adjustments, therefore, are needed for the models' parameters.

To improve forecast results, i.e. lowering cross-validated MSEs and achieving white noise residuals, different time series forecast models should be experimented and employed. Firstly, dynamic regression, which makes predictions for a particular time series by using external information from other time series, can be taken into consideration. This possibility is reasonable since there are many strongly correlated time-series presenting in the dataset (Figure 2). Secondly, another good forecast model is the dynamic harmonic regression using Fourier term to handle complex/long seasonal periods which are clearly observed in the studied time series. To be more precise, the dynamic harmonic regression employs a series of sine and cosine terms with the right frequencies to approximate periodic functions correctly. Lastly, an automated framework, TBATS, can be effectively applied for the studied time series. It is the combination of Trigonometric terms for seasonality (T), Box-Cox transformations for heterogeneity (B), ARMA errors for short-term dynamics (A), Trend or Damped Trend (T), and Seasonality (S). TBATS seems to be a comprehensive method for the time series forecast. However, carefully-chosen models' parameters are needed to avoid the overfitting problem.

Appendix

Appendix 1. 11 Variables of Interest

1) WSR_PK: continuous. peek wind speed -- resultant (meaning average of wind vector)
2) T_PK: continuous. Peak T
3) T_AV: continuous. Average T
4) T85: continuous. T at 850 hpa level (or about 1500 m height)
5) RH85: continuous. Relative Humidity at 850 hpa
6) HT85: continuous. Geopotential height at 850 hpa, it is about the same as height at low altitude
7) T70: continuous. T at 700 hpa level (roughly 3100 m height)
8) KI: continuous. K-Index
9) TT: continuous. T-Totals
10) SLP: continuous. Sea level pressure
11) SLP_: continuous. SLP change from previous day

Appendix 2. Missing values

Variable_names no_missing_values

WSR0 299, WSR1 292, WSR2 294, WSR3 292, WSR4 293,

WSR5 292, WSR6 291, WSR7 289, WSR8 290, WSR9 287,

WSR10 288, WSR11 292, WSR12 287, WSR13 288, WSR14 288

WSR15 286, WSR16 284, WSR17 283, WSR18 286, WSR19 292

WSR20 294, WSR21 293, WSR22 300, WSR23 297, WSR_PK 273,

WSR_AV 273, T0 190, T1 185, T2 187, T3 184, T4 184

T5 183, T6 183, T7 183, T8 185, T9 185, T10 188, T11 192, T12 189, T13 191, T14 192, T15 187, T16 184

T17 182, T18 184, T19 188, T20 189, T21 185, T22 192

T23 189, T_PK 175, T_AV 175, T85 99, RH85 105, U85 180

V85 180, HT85 95, T70 107, RH70 115, U70 157, V70 157

HT70 100, T50 115, RH50 125, U50 210, V50 210, HT50 112

KI 136, TT 125, SLP 95, SLP_ 158, Precp 2

References

- Dua, D. & Graff, C. (2019). *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml/index.php]. Irvine, CA: University of California, School of Information and Computer Science [Online]. Avaliable from: http://archive.ics.uci.edu/ml/datasets/Ozone+Level+Detection [Accessed 28th May 2020].
- Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice* (2nd edition). OTexts: Melbourne, Australia.