QBUS6840 Predictive Analytics (2018S1)

Group Project (Assignment 2)

Due date: Monday 28 May 2018

Group Number: ______________69_____________________________

Group Members:

_______460487999__________

_______470200568__________

_______480303419__________

_______460071662__________

_______Member Student ID__________

Note: We only ask you to use your SID and your group number. Please insert your GROUP NUMBER in the header area from the next page.

Table of Contents

Your Group Report from this page.

1. Introduction

Australia's tourism has been one of the pillar industries since 1990s, which plays a significant role in Australian economy. The profit from the tourism once accounted for the 3.2% of Australia's GDP in 2017 and one-third of it was contributed by international tourism (Tourism Research Australia, 2017). However, since the competition in global tourism industry has become increasingly severe, the number of Australia's international visitors may have to be faced with the trend of declination. In this case, this report will establish a series of models to forecast the number of Australian international visitors for the next year, providing insights for government and generating corresponding suggestions to adopt to the dynamic tourism market.
The basic steps of our forecast analysis are as follows (Hyndman and Athanasopoulos, 2014):
Step 1: Problem definition
Step 2: Gathering information
Step 3: Exploratory data analysis
Step 4: Choose and test models
Step 5: Using and evaluating a forecasting model

2. Problem definition

The forecast of the volume of international tourists from April 2018 to March 2019 is critical to the Australian economy and the government decision-making process ahead of the real demand occurred.

The first concern is if the current upward trend would extend into future. Over the past 20 years, the number of international visitor to Australia has been increasing by the twice the amount from 2 and a half million in 1990s to 6.7 million in 2014 (Australian Government, 2015). Although there seems to be a continuously increasing number arriving visitor, under the influence of emerging global market, Australia has lost 0.1% international visitors share from year 2000 to year 2013, which may bring a potential loss of visitors in a certain time periods of the future (Australian Government, 2015). This unsteadily increase of international arrivals may lead to a series of problems caused by the unreasonably seasonal allocation of government resources and poorly constructed infrastructures, putting a potential threat to the growth of Australia's GDP. Furthermore, Tourism Forecasts (Tourism Research Australia, 2017) shows that tourism spending will continue to grow at a rate of 6.3% per year reaching at $224.8 billion by 2026–27, and driven by strong forecasted demand from international visitors, future investment for the industry are also positive.

The second concern is how many the upcoming tourist's number will be in the next 12 month if a certain rising tendency does exit in demand. This is key to the government's decision in investing in relating infrastructure to guarantee that supply can keep pace with growth in demand, and to cater to the increased expectations of tourists (The Productivity Commission, 2015). There are some concerns of the inadequacy of tourism-related infrastructure, both the quantity and quality of which is poor in many regions of Australia. For example, many short-term accommodation like hotels is aged, infrastructure in national parks like boardwalks and signage is neglected and in disrepair, cruise ship infrastructure in Sydney and Brisbane (Australia's major destinations for cruise ships) is overloaded, which may fail to meet the expected growth in the tourism (The Productivity Commission, 2015). Further, the government's policy could achieve the flexibility in terms of resources allocations such as destination marketing according to the forecasted seasonality.

In conclusion, the prediction of the number of Australian international tourist number from April 2018 to March 2019 could provide the Australian government with an intuitive understanding of the Australian international tourism industry with respect to the whole economy and possibly with a direction on how to implement policies to move ahead of the demand.

3. Gathering information

The data of the Australian short-term international arrivals (less than one year) is obtained from Australian Bureau of Statistics (Stat.data.abs.gov.au,2018), which guarantees the reliability of the data source. The selected time-series is the monthly data for the last 20 years from January 1998 to March 2018, consisting of sufficient data to generate an objective outcome for the analysis.

4. Exploratory Data Analysis

The multiplicative decomposition method is applied here to perform exploratory data analysis on the original data set. The detailed procedure is explained in the report 5.53 with the in-sample data set to training the multiplicative decomposition method for model-building purpose.

The original Australian international arrivals from Jan 1998 to Mar 2018 is plotted in Figure 4.1. There is an obvious up-trend way, while the cycle component is not evident. Besides, the plot shows that the data is annually seasonal-presented, which means the number of seasonal period is 12. Further, the magnitude of change increases over time, which indicates it is appropriate to use the multiplicative method for the seasonal component.

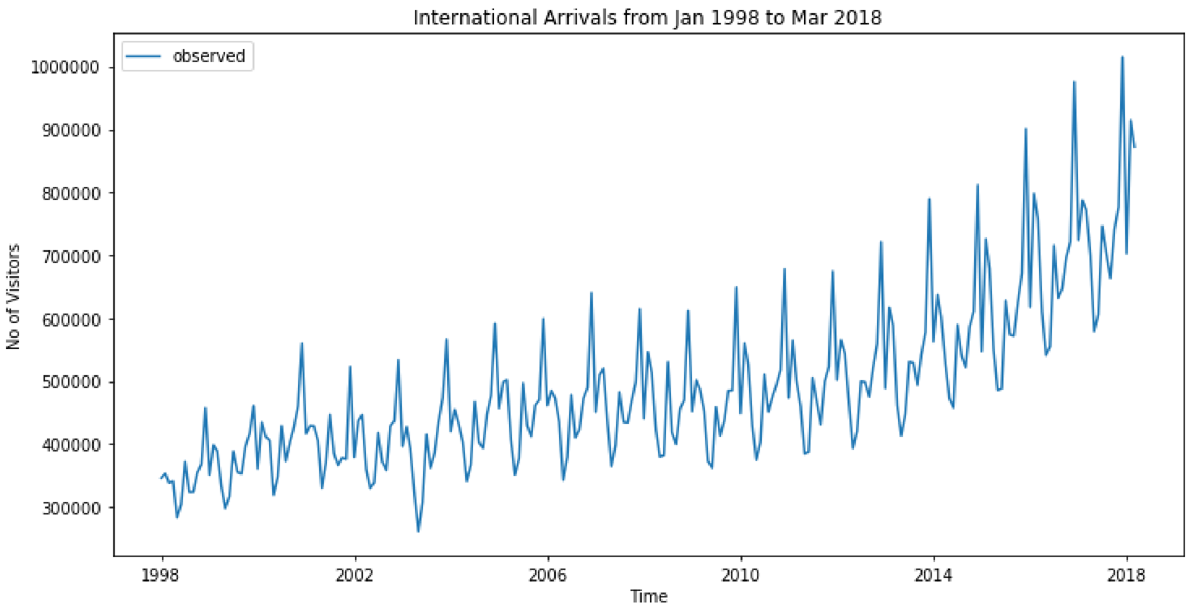

Figure 4.1

Based on the Figure 4.1, there was a sharp decrease of international arrivals since around March of 2003. According to the research of Garnaut and Robotham (2003), in November 2002, the severe acute respiratory syndrome (SARS) epidemic broke out. In February 2003, SARS started to dramatically extend from Hong Kong to all around the world, causing the worldwide death rate to 5.9%. They also suggest that this outbreak put Australia's international tourism industry into an unprecedented challenge, losing over 30% overseas visitors in merely March. This makes sense in the sudden drop of international arrivals to Australia.

The seasonal component is shown in the Figure 4.2. The seasonal component is repeated itself every year over the past 20 years.

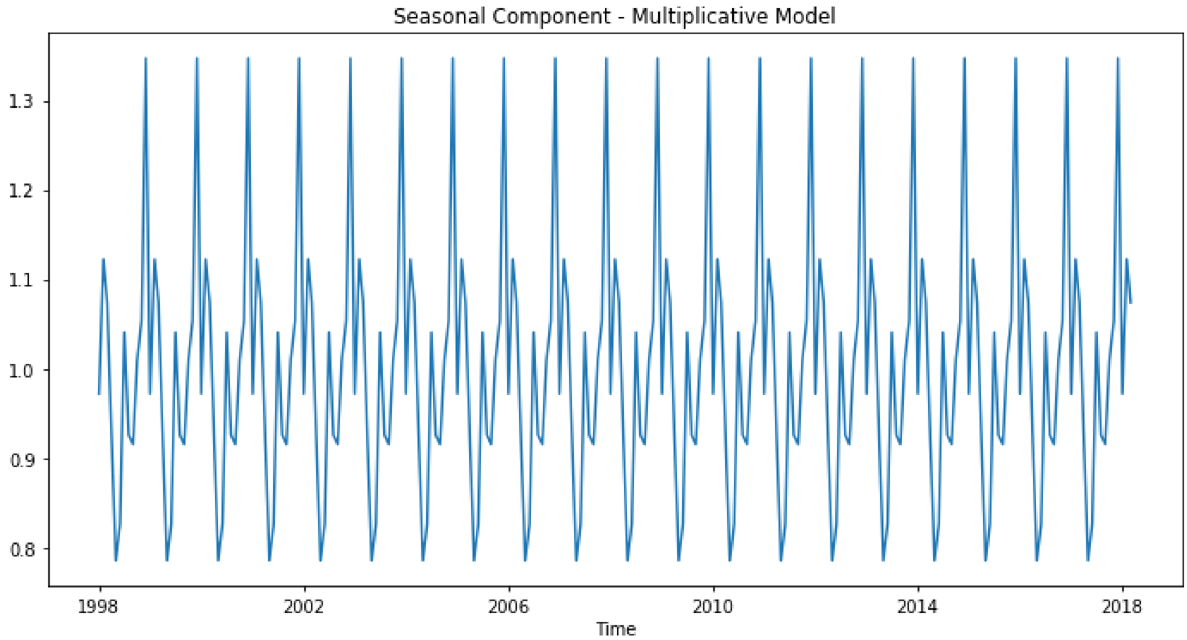

Figure 4.2

The seasonally adjusted series (in orange) and the trend component (in green) is plotted in Figure 4.3. Both lines indicate the same upward trend over the past 20 year. Precisely, the trend follows a polynomial function with degree of 3 and the slop of the line is increasing sharply roughly starting from 2010.

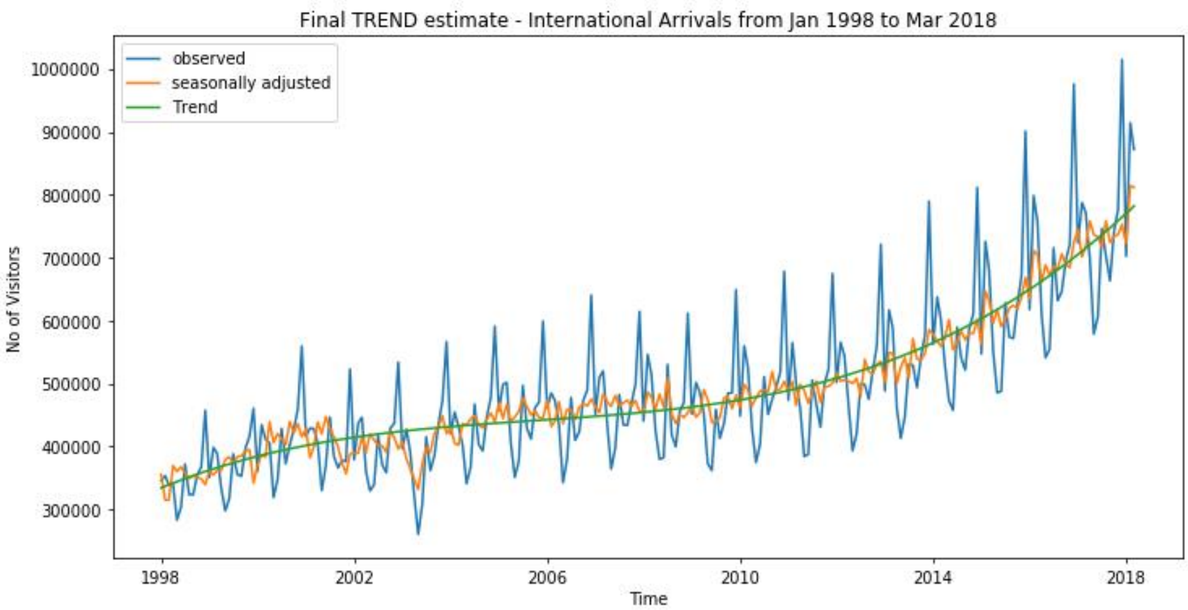

Figure 4.3.

The residual plot after applying the multiplicative decomposition method on the original data is shown in Figure 4.4. The residual is fluctuated around 1, which means the observed seasonal and trend components fit well for the data and little pattern is left in the residuals.

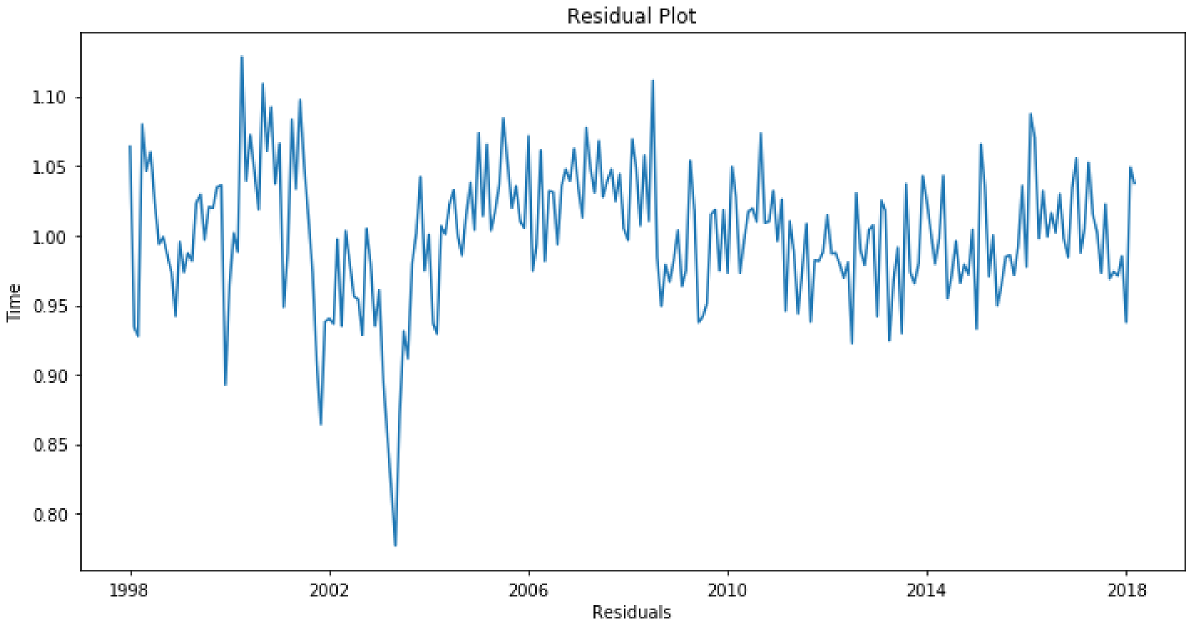

Figure 4.4

5. Motivations of Model Selection

From the plot of the original data above (Figure), there is an obvious seasonal period $m = 12$ for monthly data. The magnitudes of seasonal variations are increasing overtime so that the multiplicative method is selected for certain models. As the nature of the time series contains trending and seasonal components, Holt-Winters seasonal multiplicative method, the multiplicative decomposition method and Seasonal Autoregressive Integrated Moving average (SARIMA) method, are chosen to be appropriate to model this type of data.
To obtain the further insights from different forecasting models applied in this report, the drift method, one of the simplest models, is chosen as the benchmark model. The intuitive results could be observed by comparing the benchmark model with other three methods.

Set up in-sample data and out-of-sample data

The in-sample data (training set) is 90% of the original data set from January 1998 to Dec 2015 and the out-of-sample data (test set) is the rest 10% from January 2016 to March 2018.

5.1. Drift method

Drift method is a simple and effective method to conduct the changing forecasts over time. Based on the historical data, the average change over time is named the drift. This is equivalent to a line connected by the first and last observation, and forecast into times beyond T by extending the line (Hyndman and Athanasopoulos, 2014).

Equation

The forecast for the time point T+h is given by

$\widehat{Y}_{T+h} = Y_T + \frac{h}{T-1}\sum_{t=2}^{T}(Y_t - Y_{t-1})$.

It can be approved that

$$\widehat{Y}_{T+h} = Y_T + h(\frac{Y_T - Y_1}{T-1})$$

$Y_T + \frac{h}{T-1}\sum_{t=2}^{T}(Y_t - Y_{t-1}) = Y_T + h(\frac{Y_T-Y_1}{T-1})$

Estimate the model

Based on the in-sample data from Jan 1998 to Dec 2015, the first observation at Jan 1998 is 246200, the last observation at Dec 2015 is 901300 and the total training sample size is 216.

Thus, the drift is measured accordingly as 2581.8605 = (901300 – 246200) /2016 people per year.

The drift model forecasts a gradual increase line in the future from Jan 2016 to Mar 2018 in Figure5.1.1.

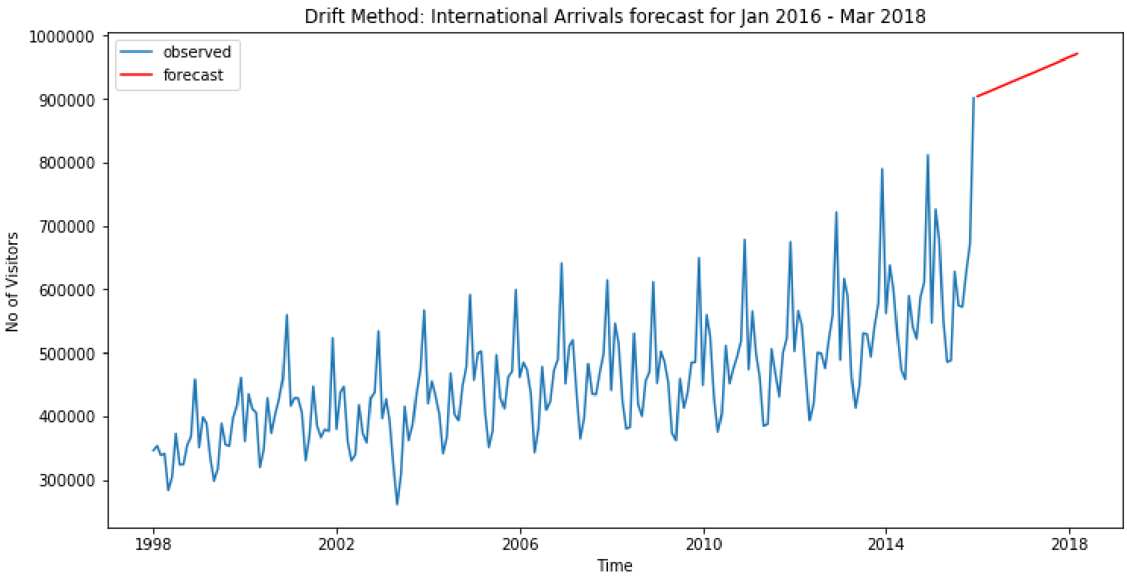

Figure5.1.1

Advantages and disadvantages of the model

Drift method is easy to compute with access to all the historical data. However, in terms of forecasting details, it needs to be more sophisticated by using other models for the further research.

5.2. Holt-Winters Seasonal Method

The Holt-Winters seasonal method is to allow forecast of data with a trend and seasonal component. The forecast equation is the product of three smoothing component equations: the level component ℓ_t, the trend component b_t, and the seasonal component S_t, and smoothing parameters are α, $\beta^*$and γ respectively for each component equation. The period of the seasonality is denoted by m within a year. As those smoothing parameters are being updated with all the latest information up to time t before the forecast for the next period, this model is called local forecasting method. Besides, α, β^* and γ are multipliers, which means the component and forecast equations are not flat but changing exponentially over the time series.

When the magnitudes of seasonal changes are changing proportionally through the series, the multiplicative method is preferred to capture variations. The multiplicative seasonality is expressed in relative terms (percentages) and the series is seasonally adjusted by dividing through by the seasonal component (Hyndman and Athanasopoulos, 2014).

Equation

The component form for the multiplicative method is:

$$\hat{y}_{t+h|t} = (l_t + hb_t)s_{t-m+h_m^+}$$

$$l_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$$

$$s_t = s_t + \gamma \frac{y_t}{(l_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}$$

And the error correction formulation is:

$$l_t = l_{t-1} + b_{t-1} + \alpha\beta^* \frac{e_t}{s_{t-m}}$$

$$b_t = b_{t-1} + \alpha\beta^* \frac{e_t}{s_{t-m}}$$

$$s_t = s_t + \gamma \frac{e_t}{(l_{t-1} + b_{t-1})}$$

where $e_t = y_t - (l_{t-1} + b_{t-1})s_{t-m}$

Forecasting formula is:

$$\hat{y}_{t+h} = (l_t + hb_t)s_{t-M+(h \, mod \, M)}$$

Advantages of the model

Holt-winters multiplicative method is most suitable for the trending data with seasonal patterns varying proportionally to the level of series and it could properly filter the effects of random fluctuations (Hyndman and Athanasopoulos, 2014)
Besides, simplicity and known accuracy are Holt-winters multiplicative method's outstanding features (Chatfield & Yar, 1988). This forecast model produces reliable predictions quickly and thus it has great economic advantages and significance for industrial applications, particularly for for sales or demand time-series data (Chatfield & Yar, 1988).

Disadvantages of the model

The Holt's method can merely accommodate one seasonality into the model and can't appeal to a time series containing more than one seasonality (Taylor, 2003). This indicates it might not be suitable for the long-term forecast as the seasonality needs to be revised when the economic environment differs or the magnitudes of seasonal component change.
In addition, this model cannot take one-off extreme event into account, which might somehow mislead the forecast result for certain time periods.

Estimate the model

Based on the in-sample data, the Holt-winters multiplicative model generates the forecast for the next 27 months from Jan 2016 to Mar 2018 in Figure5.2.1.

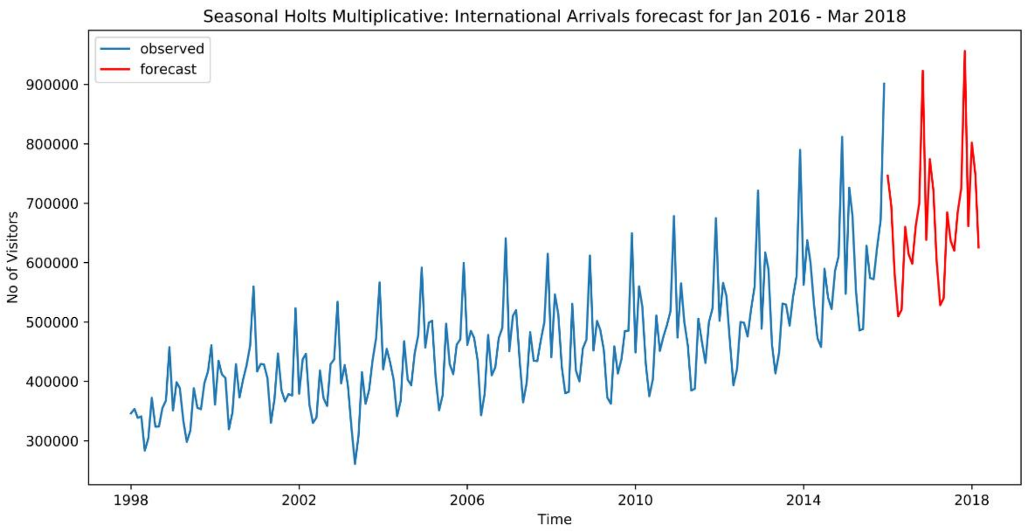

Figure5.2.1

5.3. Multiplicative Decomposition Models

The decomposition method is a comparatively simple strategy to decompose the trend component$\widehat{T}_t$, the seasonal component$\widehat{S}_t$, the possible cycle component $\widehat{C}_t$ and the irregular fluctuation component $\widehat{E}_t$from the original time series. The seasonal component is repeated itself for a time series with seasonal period m. Since the magnitudes of variations of the data changes over time, the multiplicative method is chosen. The forecast function theoretically is the product of each component (Hyndman and Athanasopoulos, 2014).

Equation

In multiplicative decomposition, the forecast function is denoted by

$$y_t = \hat{S}_t \widehat{A}_t \text{ , where } \widehat{A}_t = \widehat{T}_t \widehat{E}_t.$$

Assumptions

1. The common practice is to assume trend and cycle component as one trend-cycle component as$\widehat{T}_t$, or assume that there is no cycle component as cycle is hard to evaluate.

2. To forecast the decomposed time series, the seasonal component $\hat{S}_t$ and the seasonally adjusted component $\widehat{A}_t$ need to be separately forecasted. The assumption is that the seasonal component stays the same or changes slightly, hence, using the estimated component in the last year can forecast the seasonal component$\widehat{S}_t$. Using non-seasonal forecasting methods can forecast the seasonally adjusted component $\widehat{A}_t$.

Estimate the Model

1. As the original data has seasonal period m = 12 for monthly data, the initial trend-cycle estimate $\widehat{T}_t$ is calculated by using a 2×12-MA to obtain in Figure5.3.1.

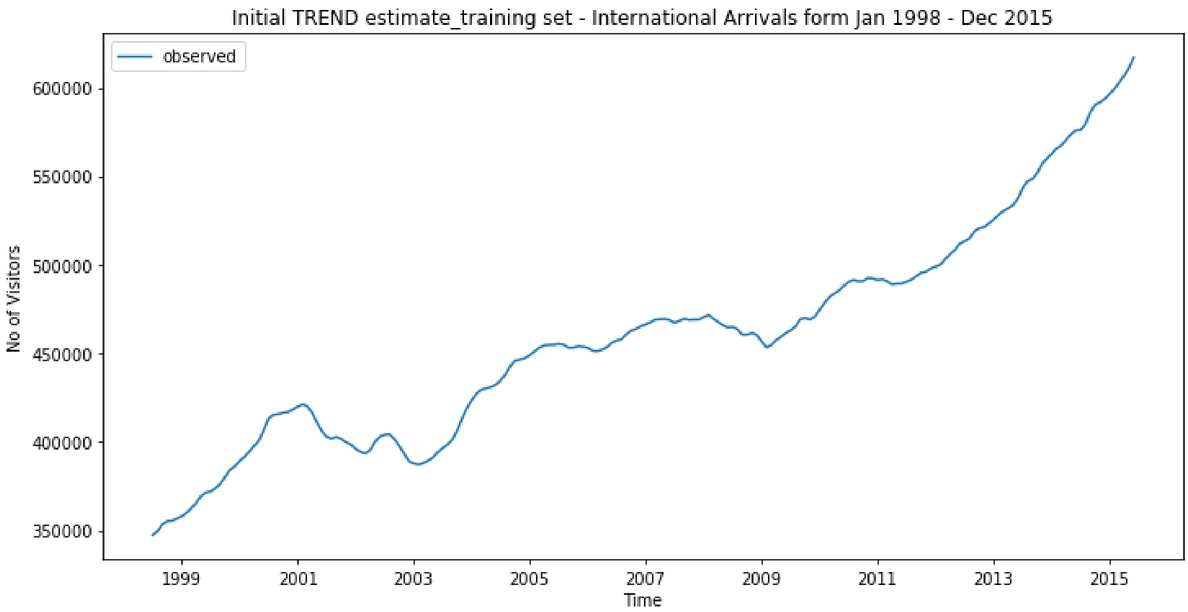

Figure5.3.1

2. The de-trended series:$y_t/\widehat{\mathrm{T}}_t$ is computed afterwards.
3. To estimate the seasonal component $\hat{\mathrm{S}}_t$for each month
12 monthly average values are computed simply by averaging the de-trended values for each month from Jan 1998 to Dec 2015.
The common multiplier, C is 12 over sum of 12 monthly average.
The 12 monthly indices are the product of C and 12 monthly average values.
Thus, the seasonal component $\hat{\mathrm{S}}_t$is the aggregated set of 12 monthly indices.
The below plot (Figure5.3.2) shows the seasonal component repeated itself over the 18 years.

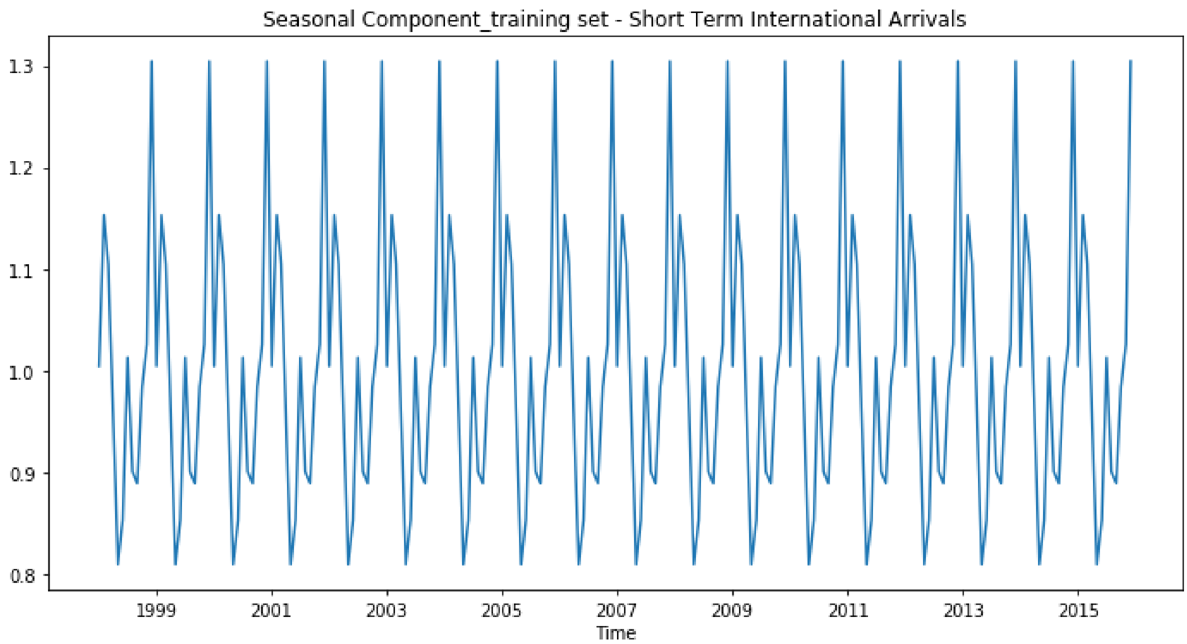

Figure5.3.2

4. Calculate seasonally adjusted series by $\frac{Y_t}{S_t}$ (Figure5.3.3) and denoted by$\widehat{T_t \times \epsilon_t}$.

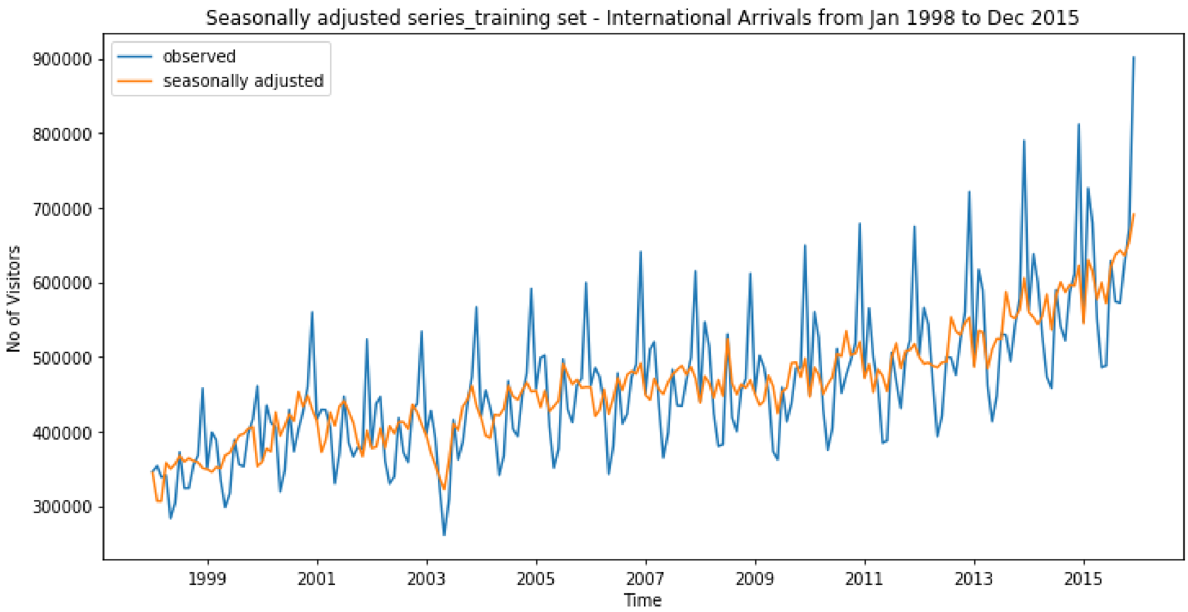

Figure5.3.3

5. Regress the seasonally adjusted series with time (Figure5.3.4) and the trend component might be polynomial with degree = 3.

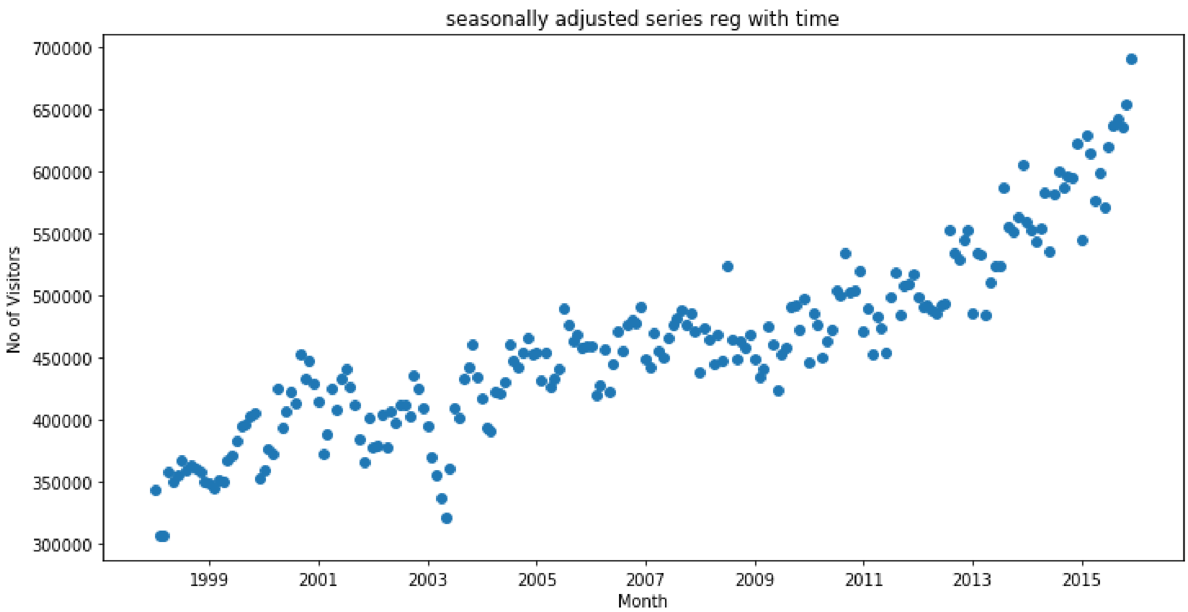

Figure5.3.4

5. To estimate the final trend component T_t by fitting the polynomial function with degree = 3 (Figure5.3.5). Tt = $0.075\ t^3-21.3t^2+2498t+3.328*10^{-5}$

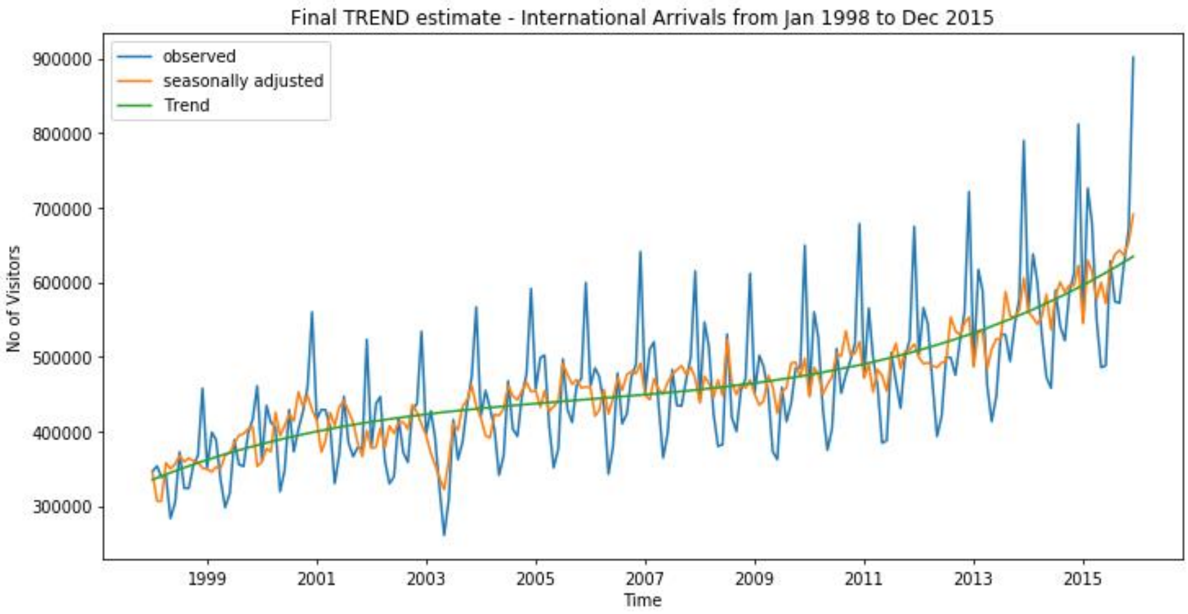

Figure5.3.5

6. Check the residual plot (Figure 5.3.6). The residuals fluctuated with 1, which means little pattern is left in the residuals and the model fits well.

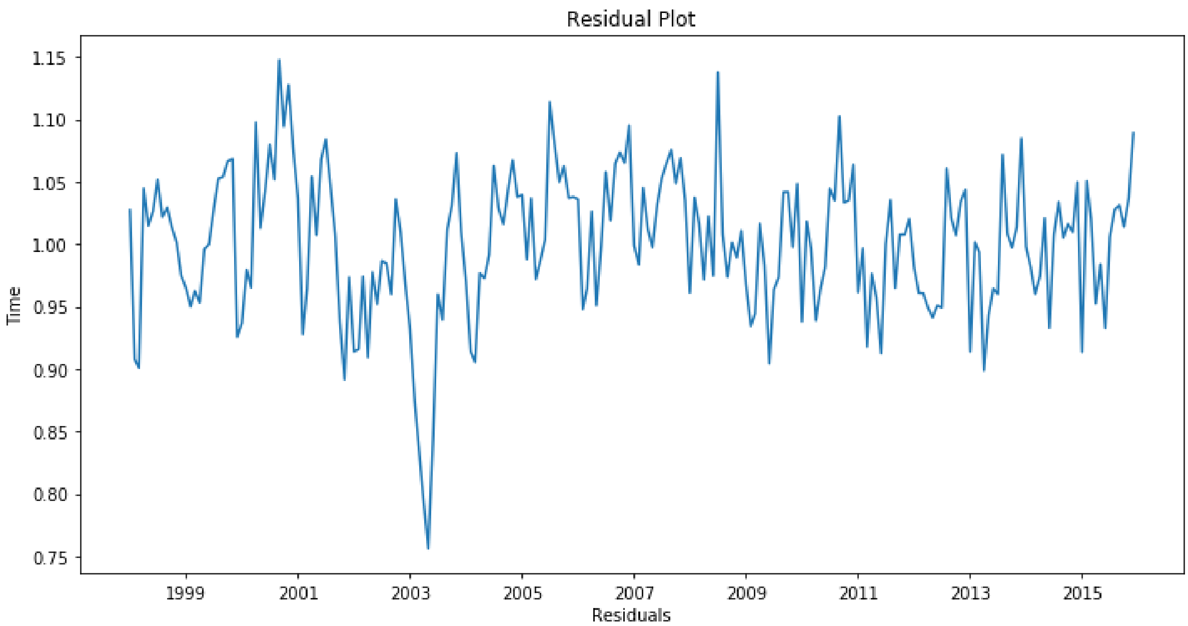

Figure 5.3.6

7. The decomposition forecast is $\widehat{T_t \times S_t}$ (Figure 5.3.7)

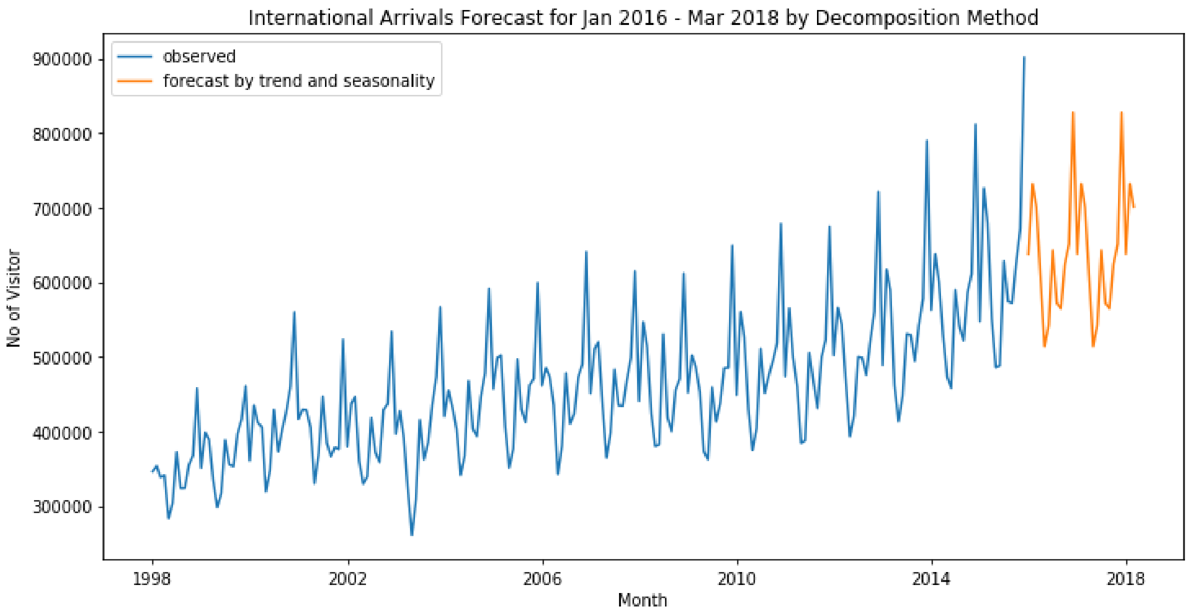

Figure 5.3.7

Advantages and disadvantages of the model

The model is simple to compute and easily to be interpreted. However, the estimate of the trend misses the first and last 6 points (if m =12) due to the nature of moving average method. As a result, no estimate could be compute for the beginning and ending periods.
Besides, the assumption 2 is not valid for some longer series, which means the decomposition method cannot capture the dynamic seasonality. Furthermore, the model cannot distinguish the unusual observations from one-off events in some short periods.

5.4. Seasonal Autoregressive Integrated Moving Average (SARIMA)

Since the Australian international tourist arrivals time series has an upward trend with constant seasonality, Seasonal Autoregressive Integrated Moving Average (SARIMA) is chosen over Autoregressive Integrated Moving Average (ARIMA) to predict forecast. SARIMA model is an extension to ARIMA model; it is a forecasting method for time series with seasonal patterns. SARIMA combines the main features of ARIMA with the additional parameters capture the seasonallity of the model. Including the number of seasonal periods m, there are 6 parameters in SARIMA model: autoregressive model(p/P), integration or number differencing applied(d/D), and moving average model(q/Q) after adding the seasonal sector.
To fit the model, the time series has to be at least weak stationary, which means the data has finite mean, variance and corvariance. This property could be detected by the autocorrelation fucntion plot(ACF) and the partial autocorrelation fucntion plot(PACF). Therefore the variable of interest could perform the autogresisive (AR) model: it could regress with against its past values. Similarly, the moving average (MA) model takes past forecast errors into account (Hyndman and Athanasopoulos, 2014).

Equation

The SARIMA model is defined as below:

$$\mathrm{ARIMA}(\mathrm{p,d,q})\ (P,D,Q)_m$$

$$\left(1-\Phi_1\mathrm{B}-\Phi_2\mathrm{B}^2-\cdots-\Phi_\mathrm{p}\mathrm{B}^\mathrm{p}\right)\left(1-\phi_1\mathrm{B}^\mathrm{m}-\phi_2\mathrm{B}^{2\mathrm{m}}-\cdots-\phi_\mathrm{p}\mathrm{B}^{\mathrm{pm}}\right)(1-B)^d(1-\mathrm{B}^\mathrm{m})^dY_t=c+(1+\theta_1B+\theta_2B^2+\cdots+\theta_qB^q)(1+\mathit{\Theta}_1\mathrm{B}^\mathrm{m}+\mathit{\Theta}_1\mathrm{B}^{2\mathrm{m}}+\cdots+\mathit{\Theta}_Q\mathrm{B}^{\mathrm{Qm}})\varepsilon_t,$$

where m is a number of periods, e.g. If the time series is monthly, m = 12. The P, D, Q represents the seasonal part of the model, and p, d, q represents the non-seasonal part of the model.

Advantages and disadvantages of the model

ARIMA model is a proper way to perform short-term time series forecasting (Box, 1970). As the model only requires the prior data of a time series to generalize the forecast, the model forecast accuracy could be improved with the minimum number of parameters (Jarrett, 1991).

However, Thomas (1983) suggests the underlying theoretical model and structural relationships are not distinct as some simple forecast methods. Moreover, the ARIMA models is "backward looking" and long-term forecast would become a straight line. Also, the order selection is a manual process in this report, and therefore it hardly to try all the order combinations and pick up the best one.

Estimate the model

1.Check the stationarity of the origigal data
To apply this model, the time series must be at least weakly stationary.
The original time series indicates it is trending with seasonality in Figure.
The ACF and PACF of original data (Figure4.1) also shows the non-stationarity as they both die down slow with repeated patterns.

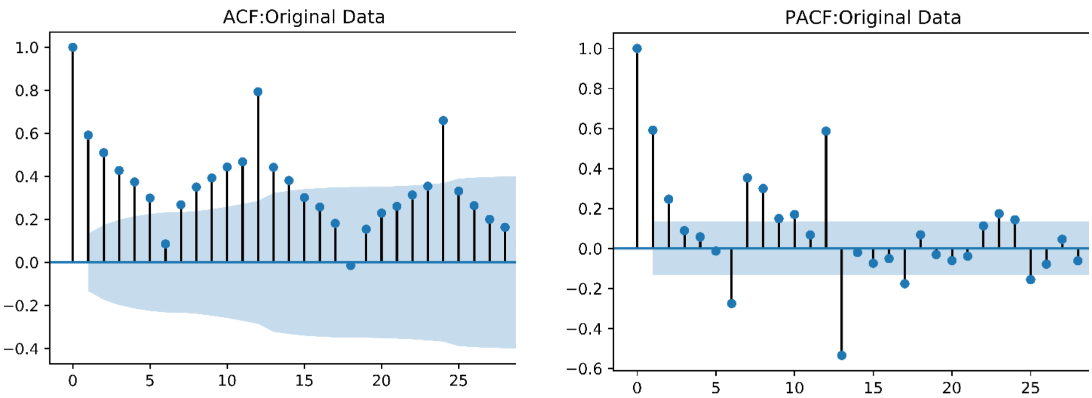

Figure 5.4.1

2. Data transformation
To stabilize the variable, take the log transformation of the original data. The Figure 5.4.2 shows the magnitude of change is relatively improved to be fixed after transformation but it is still trending.

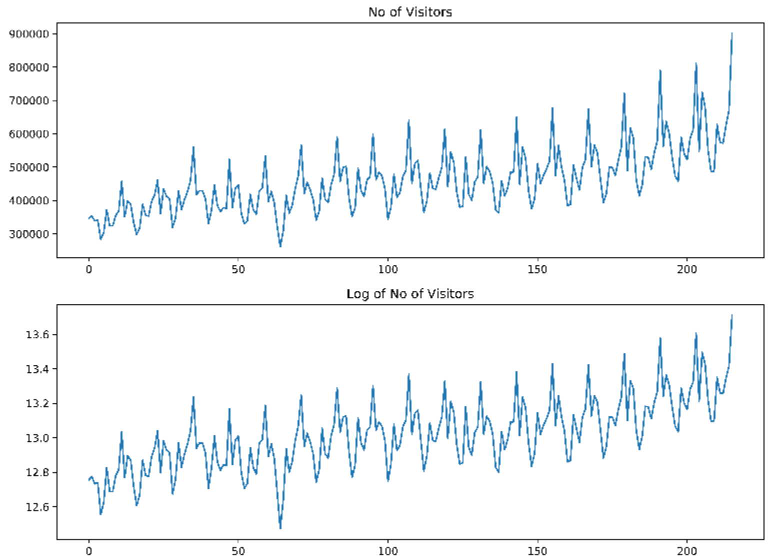

Figure 5.4.2

3. To stabilize the mean, take the first the first order difference to the log data. The Figure 5.4.3 shows the magnitude of change is stable and the transformed data is moves around 0.

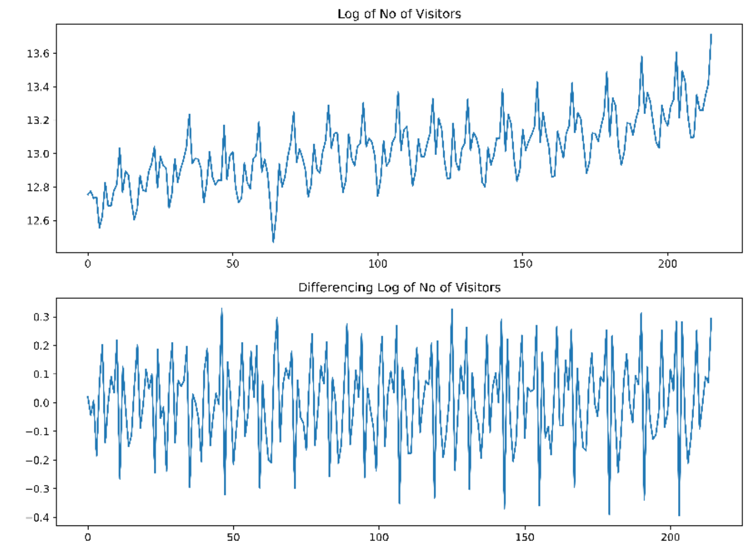

Figure 5.4.3

The ACF and PACF of differenced original data (Figure5.4.4) shows they both die down quickly after lag one or lag two. But significant seasonality lag spikes at lag 12, lag 24, and lag 36 in both plots indicates the seasonality.

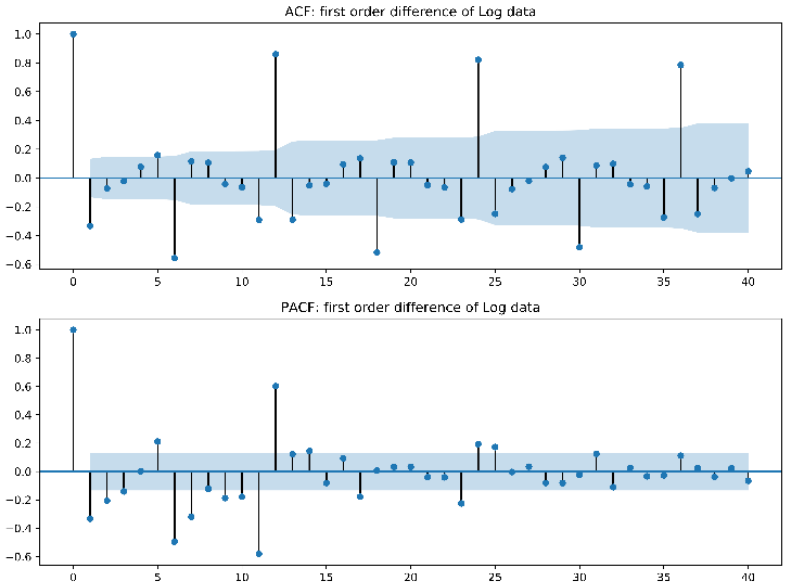

Figure 5.4.4

4. Transform the data on the seasonal level
Conduct the seasonal differencing,$Y_t - Y_{t-12}$ on lag data to remove the seasonality. At non-seasonal level, the ACF dies down fairly slow up to lag 8 and PACF has high spikes at lag 21, 23 and 25 (Figure 5.4.5).

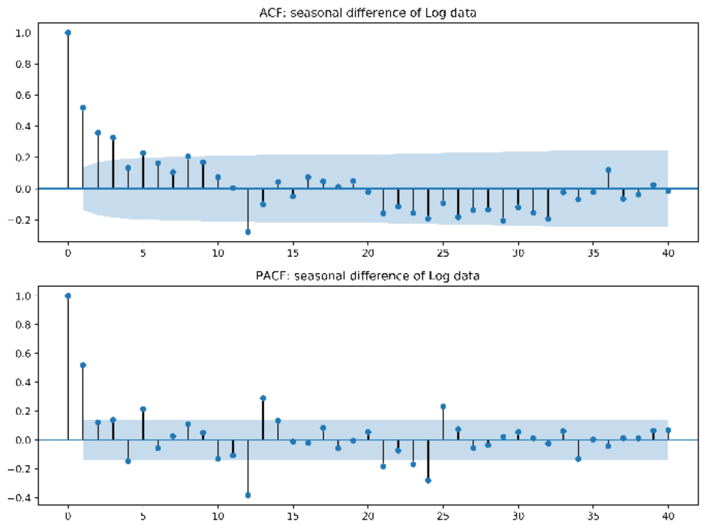

Figure 5.4.5

5. Further take the first order difference on the seasonally differenced log data. Figure 5.4.6 shows that the final transformed data has relative stable variance and zero mean. Figure 5.4.7 shows the ACF dies down quickly after lag 4 at the non-seasonal level and at the seasonal level it has spike at lag 1 and lag3. The PACF dies down quickly after lag 4 at non-seasonal level and at seasonal level it cuts off at lag 2 and no more spikes afterwards.

Therefore, the log data can considered as stationary after another difference from seasonally differenced.

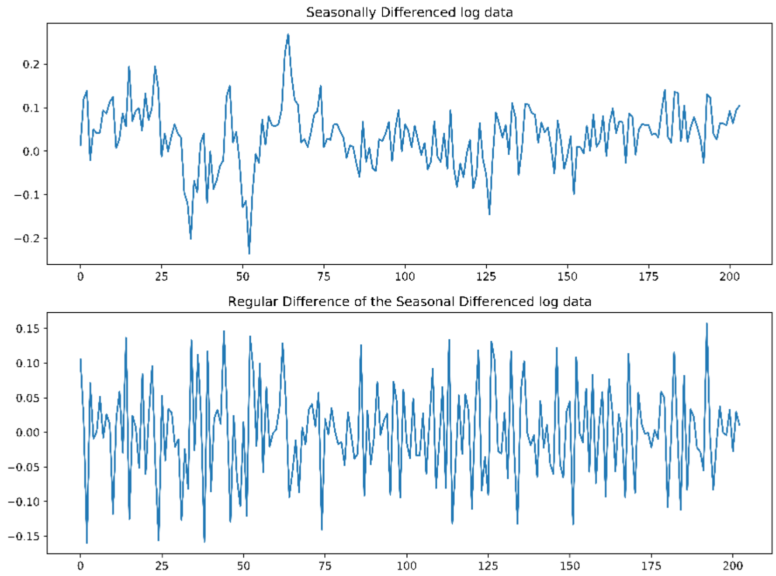

Figure 5.4.6

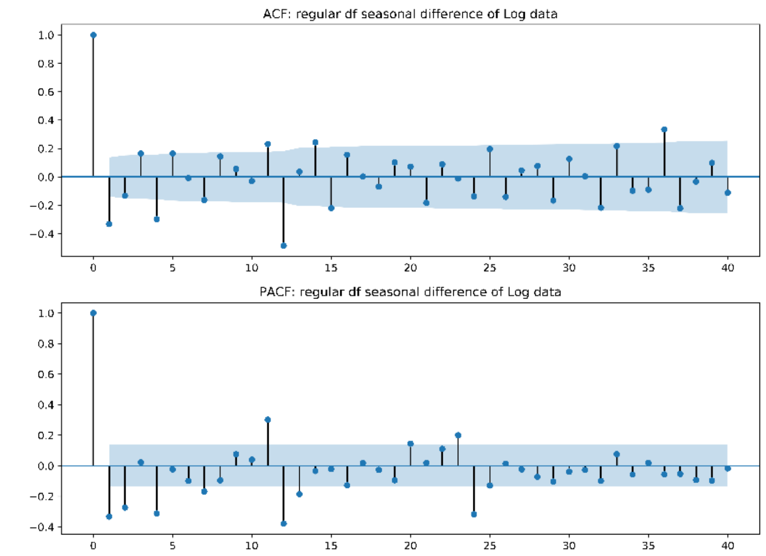

Figure 5.4.7

6. SARIMA Order Selection
From ACF in Firgure 4.7, at non-seasomal level, lag spikes at lag 1 or at lag 4, which indicates MA(1) or MA(4) (p = 1 or 4). At the non-seasnol level, it spikes at lag 2 and lag 3, which suggests MA (1) or MA (3) (P =1 or 3).
From PACF in Figure 4.7, the non-seasonal lag spikes cuts off after lag 4 but with spike at lag 1 and 2. This suggests AR(1), AR(2) or AR(4) (p = 1,2or4). At seasonal level, it cuts off at lag 2 indicating AR (2) (P=2).
From the data transformation, D (1) and d (1) can be applied to SARIMA model and the seasonal period m =12.

The best order combination is selected by Akaike's Information Criterion (AIC) by find the one minimize AIC score. The (AIC) is defined as$AIC = -2\log(L) + 2(p + q + k + 1)$, where the L is the likelihood of data and k = 1 if the model has a constant (Gao, 2018). The Table5.1 shows the 7 outcomes of different order combination with Seasonal ARIMA process.

Model	AIC
ARIMA$(1,1,1)(2,1,0)_{12}$	-661.425
ARIMA$(1,1,4)(2,1,0)_{12}$	-667.954
ARIMA$(1,1,5)(2,1,0)_{12}$	-665.984
ARIMA$(2,1,4)(0,1,1)_{12}$	-661.474
ARIMA$(2,1,5)(0,1,1)_{12}$	-658.398
ARIMA$(4,1,5)(0,1,1)_{12}$	-647.727
ARIMA$(1,1,4)(0,1,3)_{12}$	-663.971

Table 5.1

Since SARIMA $(1,1,4)(2,1,0)_{12}$ model has the smallest value of AIC (-667.954), it is selected to forecast the data.
The model of Seasonal ARIMA $(1,1,4)(2,1,0)_{12}$ with backshift operator (B) can be defined as:

$$\begin{aligned}(1-\phi_1 B)(1-\Phi_1 B^{12}-\Phi_2 B^{24})(1-B)(1-B^{12})y_t \\ = c + (1+\theta_1 B+\theta_2 B^2+\theta_3 B^3+\theta_4 B^4)\varepsilon_t\end{aligned}$$

7. Seasonal ARIMA $(1,1,4)(2,1,0)_{12}$ forecasts the number of international tourist from Jan 2016 to Mar 2018 in Figure 5.4.8.

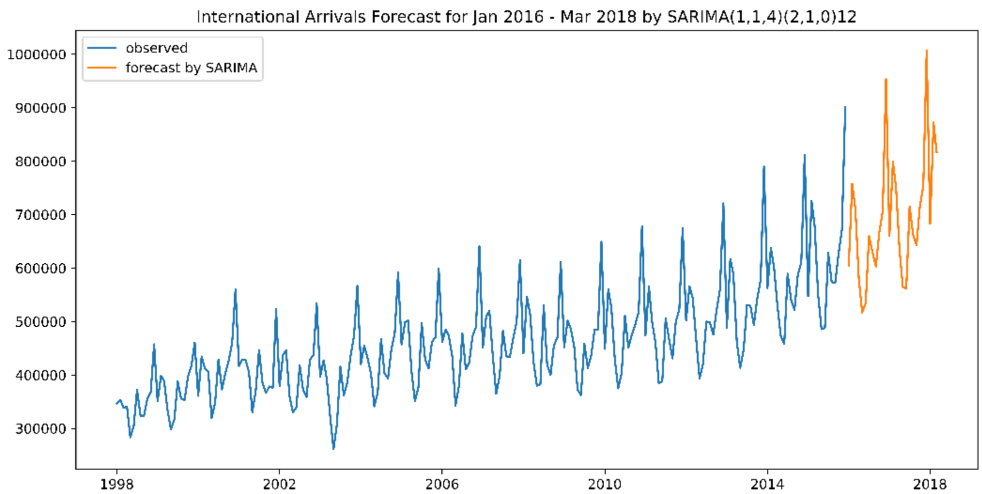

Figure 5.4.8

5.5 Model Combination

The aim for the model combination is to yield lower the error than independent forecasts so that it could result in a better forecast accuracy. The error generated in the test set will be used in determining the weight of two or more unbiased individual forecasts in forming the combined forecast, and 4 different combination methods will be used to combine all the unbiased and independent forecasts. To be noticed, the benchmark model, drift method, is not included in the combination process. The final combination forecast results examine whether it is necessary to do model combination after all. The RMSEs are listed in table 6.1 and 6.2 in Model Accuracy and Selection.

5.5.1 The equal-weighted combination

The first combination is an equally weighted combination. There are three forecasting models which are Holt's winter seasonal method (HW), multiplicative decomposition model and SARIMA. Therefore, the weight will be 1/3 for each forecast so that the first combination can be defined as:

$$\hat{y}^{C1}_{T+1|T} = \frac{1}{3}\hat{y}^{Holt's\ Winter}_{T+1|T} + \frac{1}{3}\hat{y}^{Decomposition}_{T+1|T} + \frac{1}{3}\hat{y}^{SARIMA}_{T+1|T}$$

5.5.2 The optimal weights combination

Two unbiased forecasts model are selected and calculated their forecast variances and correlations, and the optimal weights could be obtained by the formula $w = \frac{\sigma_2^2 - \rho\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2-2\rho\sigma_1\sigma_2}$.

The two unbiased forecasts that we selected are SARIMA and decomposition model, the reason we selected these two forecasts is that they have better accuracy than others. The weight for SARIMA is 0.9391, rounded in 4 decimal points. The weight for Decomposition model is 1 - 0.9391 that is 0.0609, so the second combination can be defined as:

$$\hat{y}^{C2}_{T+1|T} = 0.9391\hat{y}^{SARIMA}_{T+1|T} + 0.0609\hat{y}^{Decomposition}_{T+1|T}$$

5.5.3 The weights proportionally inverse to RMSE

This combination sums up the inverse RMSE from three unbiased forecasts, and the size of the weight will be the unbiased forecast inverse RMSE over the sum of inverse RMSE from the three unbiased forecasts (Timmermann, 2013). As result, the weight for SARIMA is0.6276, 0.2243 for Decomposition and 0.1481 for Holt's winter. The third combination can be defined as:

$$\hat{y}^{C3}_{T+1|T} = 0.6276\hat{y}^{SARIMA}_{T+1|T} + 0.2243\hat{y}^{Decomposition}_{T+1|T} + 0.1481\hat{y}^{Holt's\ Winter}_{T+1|T}$$

5.5.4 The weights inverse to rankings

First, unbiased forecasts are ranked based on rank (Timmermann, 2013), so SARIMA will be number 1, and followed by decomposition model, and Holt's winter. Then sum up the rank inverse: $1^{-1} + 2^{-1} + 3^{-1}$ and calculate the weight by using the inverse ranking number over the sum of rank inverse to get the weights for each forecast. As a result, the weight for SARIMA is 0.5455, 0.2727 for Decomposition and 0.1818 for Holt's winter and the last combination can be defined. as:

$$\hat{y}^{C4}_{T+1|T} = 0.5455\hat{y}^{SARIMA}_{T+1|T} + 0.2727\hat{y}^{Decomposition}_{T+1|T} + 0.1818\hat{y}^{Holt's\ Winter}_{T+1|T}$$

6. Model Accuracy and Selection

The mean absolute distance (MAD), root mean square error (RMSE) and mean absolute percentage error (MAPE), all three can be used to assess the model accuracy in the test set. The best performed model is selected when it gives the smallest forecast error. And this model will be used to forecast future patterns.

MAD measures the absolute average distance between actual values and predicted values.

$$\mathrm{MAD} = \frac{1}{\mathrm{h}}\sum\nolimits_{\mathrm{t=T+1}}^{\mathrm{T+h}} |\mathrm{Y_t} - \widehat{\mathrm{Y}}_\mathrm{t}|$$

RMSE measures the average root of the squared distance between actual values and forecast values, the greater the distance, the larger the RMSE.

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{h}}\sum\nolimits_{\mathrm{t=T+1}}^{\mathrm{T+h}} (\mathrm{Y_t} - \widehat{\mathrm{Y}}_\mathrm{t})^2}$$

MAPE measures the absolute average percentage of the distance between predicted values and actual values. The requirement is $\mathrm{Y_t}$ does not equal to zero.

$$\mathrm{MAPE} = \frac{1}{\mathrm{h}}\sum\nolimits_{\mathrm{t=T+1}}^{\mathrm{T+h}} \frac{|Y_t - \widehat{Y}_t|}{\mathrm{Y_t}} \times 100$$

If many large forecast errors need to be guarded against, RMSE is a better estimate in this case. Same situation can apply to the forecast models in this report since the errors are typically large. It will be easier to differentiate biased or unbiased forecast models. Therefore, RMSE is selected for model accuracy estimates in this report as it gives a punishment on a longer distance between original values and predicted values.
The following Table6.1 shows the forecast models accuracy, and Table 6.2 shows the model combination accuracy.

	Drift method	Holt's winter method	Model Decomposition	Seasonal ARIMA
RMSE	**238637.1866**	**148296.5701**	**97896.109**	**34995.79906**
MAD	219561.0853	117694.9229	85693.46257	30694.59434
MAPE	32.986%	15.296%	11.304%	4.295%

Table6.1

	Combination 1	Combination 2	Combination 3	Combination 4
RMSE	**80560.33419**	**37738.27739**	**55782.72679**	**61669.14334**
MAD	64526.04448	33702.18057	46836.69158	51348.2922
MAPE	8.3534%	4.6754%	6.1697%	6.7297%

Table 6.2

From Table 6.1, the Drift method has the highest error in all three model accuracy measures, while Seasonal ARIMA has the lowest error in all. Therefore, the ranking in terms of model accuracy is SARIMA > Decomposition > Holt's winter >Drift.

From Table 6.2, the Model Combination 2 stands out with lowest forecast error, and Model Combination 1 has the highest forecasts error. Therefore, the ranking in terms of combination accuracy is Combination 2>Combination 4>Combination 3>Combination 1.

However, in overall accuracy estimate, the Model Combination did not help to improve the accuracy, since Seasonal ARIMA has better accuracy in all. This might be the combined models did not contain independent advantage or features. According to Bates, J. M., and C. W. J. Granger (1969), to improve the model accuracy with Combination, the selected forecasts will need to have independent "information" to yield improvement on model accuracy. For example, in the optimal weight combination, the selection is to combine SARIMA and Decomposition model to get an improvement in forecast accuracy. However, these two forecasts do not have independent features but shares the same features, such as they both have trend and seasonality. The same result is expected to combine the Holt's winter method. There is one situation that the combination method might help improve the accuracy where the combined forecast models that have different features. For example, one has the trend and the other one has the seasonality. Combining them with different weight settings might yield the improvement in forecast accuracy. Consequently, all other combinations cannot improve the model accuracy compare to single forecasts SARIMA.

In conclusion, it is unexpected that the Model Combination cannot improve the model accuracy and the forecast model SARIMA has better accuracy than all Combinations. So SARIMA$(1,1,4)(2,1,0)_{12}$ model is selected for the final forecast.

7. Limitations of the forecast models

1. There are always incidents and irresistible factors, which cannot be predicted through data analysis and theoretical models. Therefore, our prediction results cannot cover all the possibilities, and there may be large differences between reality and forecast at some points of time.

2. The seasonal adjustments used in this article are all based on theoretical methods, which are comparatively simple, patterned and direct. And the adjustments lost touch with real life to some extent without considering the calendar-related events, like Chinese New Year, Ramada. Therefore, a correction method which removes some significant calendar-related effects should be implemented in the seasonal adjustment to improve the forecasting.

8. Conclusion

The SARIMA $(1,1,4)(2,1,0)_{12}$ model is selected to conduct the realizable forecast for the next year from April 2018 to March 2019. The overall number of the Australian international tourist is forecasted to increase over the next 12 month. The slack season will be May and June 2018, followed by the first small peak in July (811841). The number will drop to 739544 in August and gradually increase to 1135215 in December 2018, which will be the new historically high volume. The tourists in January 2019 will decrease largely to 795468, after which the number will go back to 986151 in February and 944312 in March. The specific forecast volume is shown in the below Table8.1:

	Apr 18	**May 18**	**Jun 18**	**Jul 18**
No of tourists	718578	629141	635287	811841
	Aug 18	**Sep 18**	**Oct 18**	**Nov 18**
No of tourists	739544	733439	802470	849435
	Dec 18	**Jan 19**	**Feb 19**	**Mar 19**
No of tourists	1135215	795468	986151	944312

Table8.1

This forecast sill follows the same seasonal pattern as before from January 1998 to March 2018 but with larger magnitude which is shown in the Figure 8.1 and Figure 8.2.

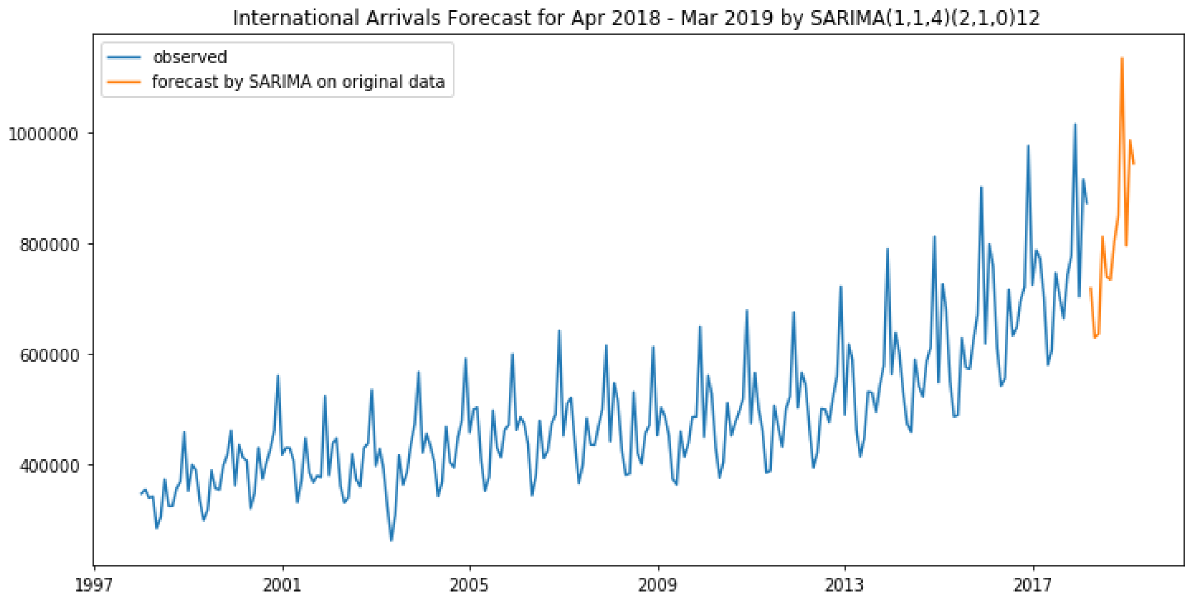

Figure 8.1

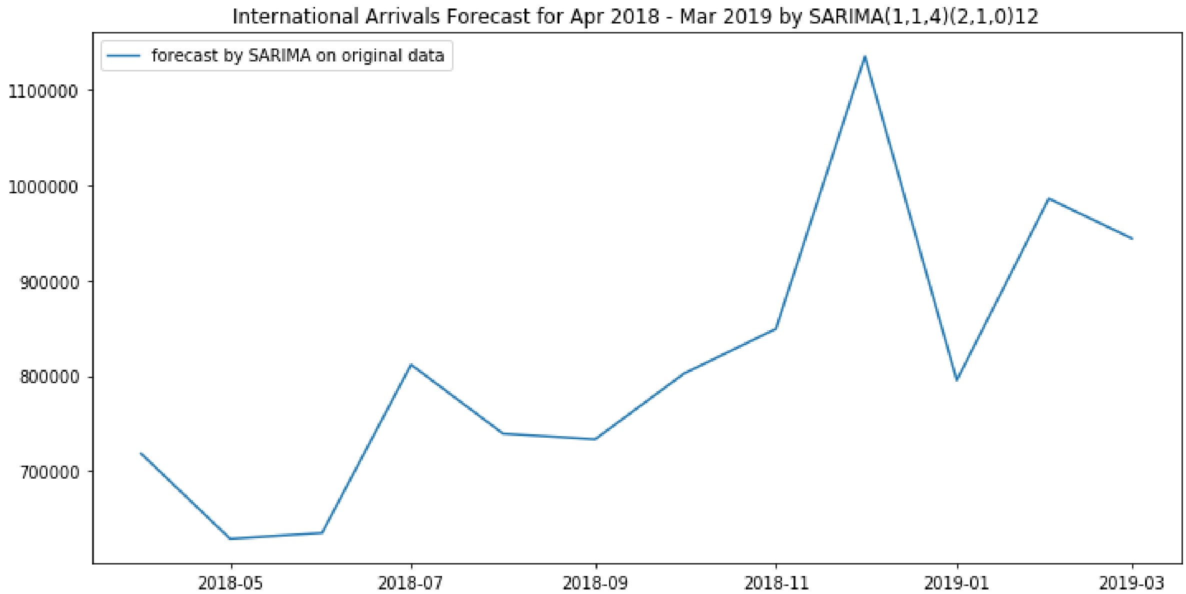

Figure 8.2

And the result indicates that there is necessity for the government to implement corresponding policies to ensure that tourism transport capacity and infrastructure will allow increasing numbers of international visitors arriving in Australia, and flexible adjustment in allocation of resource in tourism-relevant industries such as hospitality is also needed.

9. Recommendations

9.1 In short term (seasonality)

1. *Put efforts on carrying out destination marketing to attract tourists*

For the off-season where tourists reduce, the government should increase its efforts in marketing, publicity, and other aspects to create new selling points to attract tourists from all over the world. For example, set up some innovative and fantastic festivals, or celebrations with Australian cultural characteristics, and supplement with a series of interesting activities like 'Light Show', so that visitors can have a high degree of participation. And through various media such as networks, social media and advertising in target countries, vigorously promote these festival activities.

2. *Introduce short-term preferential policies to attract tourists*
For the off-season, the government can use many short-term limited preferential policies as incentives, such as reducing travel visa fees, launching special-priced airline tickets with airlines, and even adjusting the high exchange rate to attract tourists.

3. *Make flexible control and arrangement of resources*
For the international tourism industry, seasonality is an inherent dynamic attribute that is inevitable. In addition to actively responding to seasonal declines by attracting tourists, the government must also timely regulate and reconfigure relevant resources according to the predicted changes. For example, the government can provide short-term jobs for the unemployed in the tourism industry to cope with the demand of peak period.

9.2 In long term (trend)

1. Increase government investment in tourism-related infrastructure
The government should increase investment in tourism-related infrastructure, from increasing supply, renovating to repairing, especially in short-term accommodation, transport and visitor attractions, and it can estimate the approximate investment value based on the result of forecasting.

2. *Remove unnecessary barriers to private investment in tourism-related infrastructure*
In some cases, the government plays the role of regulating the privately provided infrastructure, and it should make sure not to pose unnecessary barriers in regulatory frameworks at both the state and national level. It can streamline environmental assessment processes, remove duplication of regulation, and simplify planning systems (The Productivity Commission, 2015).

Reference list

Allan, Timmermann. (2013). Forecast Combinations Oxford, July-August 2013

Bates, J. M., and C. W. J. Granger (1969). The combination of forecasts, Operational Research Quarterly, 20, 451–468. Retrieved from ftp://rammftp.cira.colostate.edu/Knaff/Annular_pres/3008764.pdf

Box, G.E.P. and G.M. Jenkins (1970) Time series analysis: Forecasting and control, San Francisco: Holden-Day.

Chatfield, C., & Yar, M. (1988). Holt-Winters Forecasting: Some Practical Issues. Journal of the Royal Statistical Society. Series D (The Statistician), 37(2), 129-140. doi:10.2307/2348687

Garnaut, J. & Robotham, J. (2003, April 25). SARS spells disaster for inbound tourism.

Retrieved from https://www.smh.com.au/articles/2003/04/24/1050777360453.html

Hyndman, R. and Athanasopoulos, G. (2014). Forecasting: Principals & Practices. 1st ed. University of Western Australia.

Jeffrey, Jarrett. Business Forecasting Methods, Second Edition, Basil Blackwell (1990).

Thomas M. O"Donovan. Short Term Forecasting, An introduction to the Box-Jenkins Approach, John Wiley & Sons (1983).

Stat.data.abs.gov.au. (2018). ABS Statistics. [online] Available at: http://stat.data.abs.gov.au/ [Accessed 28 May 2018]

Taylor, J. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. Journal Of The Operational Research Society, 54(8), 799-805. doi: 10.1057/palgrave.jors.2601589

Timmermann, A. (2013). Forecast Combinations.

Tourism Australia. (2011). Tourism 2020 (pp. 1-5).

The Productivity Commission. (2015). Australia's International Tourism Industry (pp. 1-148). Canberra.

Tourism Research Australia. (2017). Tourism Forecasts (pp. 1-7).

Tourism Research Australia. (2017). Tourism Satellite Account (pp. 1-8).

Appendix

1. Python Code

```
# PART A - EDA on original data

"""

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import holtwinters as ht
import datetime as dt

# EDA on original data
# Load the dataset;
visitors = pd.read_csv('InternationalArrivals.csv')

# Seperate the time and variable y
numVisitors = visitors[visitors.columns[1]]
months = visitors[visitors.columns[0]]

# Product time for plotting purpose
# %y stands for years 00-99; %b stands for months
x = np.array([dt.datetime.strptime(d, '%b-%y') for d in
months])

# Plot the original data
plt.figure(figsize=(12,6))
plt.plot(x,numVisitors,label = 'observed')
plt.title('International Arrivals from Jan 1998 to Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')
plt.legend(loc=2)

# Highlight the seasonal component peaks
plt.figure(figsize=(12,6))
plt.plot(x,np.power(numVisitors, 2) )
plt.title('Power 2 - International Arrivals from Jan 1998 to
Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')

# To calculate the initial trend-cycle estimate by moving
average smoothing
# m=12, 2x12 MA by chaining a 2-MA and a 12-MA as follows:
T = numVisitors.rolling(2, center =
True).mean().rolling(12,center = True).mean()
T = T.shift(-1)

# Plot the initial trend estimate
plt.figure(figsize=(12,6))
```

```
plt.plot(x,T)
plt.title('Initial TREND estimate - International Arrivals
from Jan 1998 to Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')

# Calculate the detrend series
S_multiplicative = numVisitors / T

# Plot the seasonal components to verify
plt.figure(figsize=(12,6))
plt.plot(x, S_multiplicative)
plt.title('the detrend series - International Arrivals from
Jan 1998 to Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')

# Calculate 12 monthly average data over the time
safe_S = np.nan_to_num(S_multiplicative)
monthly_S =
np.reshape(np.concatenate( (safe_S,[0,0,0,0,0,0,0,0,0]), axis
= 0), (21,12))
monthly_avg = np.mean(monthly_S[1:19,:], axis=0)

# Constant multiplier C, M =12
c = 12 / np.sum(monthly_avg)

# Montnly indicies, which is the seasonal component St
S_bar = c * monthly_avg

# Repeat the average over 21 years
tiled_avg = np.tile(S_bar, 21)

# Plot seasonal component
plt.figure(figsize=(12,6))
plt.plot(x,tiled_avg[:243])
plt.title('Seasonal Component - Multiplicative Model')
plt.xlabel('Time')

# Seasonally adjusted series
seasonally_adjusted = numVisitors / tiled_avg [:243]

# Plot seasonally adjusted series
plt.figure(figsize=(12,6))
plt.plot(x,numVisitors, label = "observed")
plt.plot(x,seasonally_adjusted,label = 'seasonally adjusted')
plt.title('Seasonally adjusted series - International Arrivals
from Jan 1998 to Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')
plt.legend()

# Seasonally adjusted series regress with time
plt.figure(figsize=(12,6))
plt.title("seasonally adjusted series regress with time")
plt.scatter(x,seasonally_adjusted)
```

```
plt.xlabel("Month")
plt.ylabel("No of Visitors")
plt.show(block=False)

# Use polynomail with degree = 3 to fit the trend
import numpy as np

X_poly = np.arange(1, len(seasonally_adjusted)+1, 1)
Y_poly = seasonally_adjusted

polymodel_fit = np.polyfit(X_poly,Y_poly,3)
polymodel = np.poly1d(polymodel_fit)

# Trend formula coefficient
print(polymodel_fit )

# Trend formula
print(polymodel )

x_polypred = np.arange(1, len(seasonally_adjusted)+1, 1)

poly_pred= np.polyval(polymodel_fit,x_polypred)

# Trend prediction
trend_poly = poly_pred

# Plot the final trend by polynomail model (power = 3)
plt.figure(figsize=(12,6))
plt.plot(x,numVisitors, label = "observed")
plt.plot(x,seasonally_adjusted,label = 'seasonally adjusted')
plt.plot(x,trend_poly, label="Trend")
plt.legend()
plt.title("Final TREND estimate - International Arrivals from
Jan 1998 to Mar 2018")
plt.xlabel("Time")
plt.ylabel("No of Visitors")
plt.show(block=False)

# Predicted by trend * seasonality
prediction_ploy_seasonal = poly_pred * tiled_avg [:243]

# Plot decomposition method (trend and seasonality)
plt.figure(figsize=(12,6))
plt.plot(x,numVisitors,label='observed')
plt.plot(x,prediction_ploy_seasonal, label='multiplicative
decomposition method')
plt.legend(loc=2)
plt.title("International Arrivals from Jan 1998 - Mar 2018 by
Decomposition Method")
plt.xlabel("Month")
plt.ylabel("No of Visitor")

# Residual scatter plot
# The residuals are fluctuated around 1
# The multiplicative model fits well
plt.figure(figsize=(12,6))
```

```
plt.title("Residual Plot")
plt.plot(x, seasonally_adjusted / trend_poly)
plt.xlabel("Residuals")
plt.ylabel('Time')
plt.show(block=False)
```

 # PART B Modelling based on training set

```
"""

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import holtwinters as ht
import datetime as dt


#########
# Modelling
# Load the dataset; Data from Jan 1998 to Mar 2018
visitors = pd.read_csv('InternationalArrivals.csv')

# Seperate the time and variable y
numVisitors = visitors[visitors.columns[1]]
months = visitors[visitors.columns[0]]

# Training set Jan 1998 - Dec 2015 & test set Jan 2016 - Mar
2018
training = numVisitors[:216]
test = numVisitors[216:]

months_tr = months[:216]
months_test = months[216:]

# Product time for plotting purpose
x_tr = np.array([dt.datetime.strptime(d, '%b-%y') for d in
months_tr])
x_test = np.array([dt.datetime.strptime(d, '%b-%y') for d in
months_test])

# Plot the original test set
plt.figure(figsize=(12,6))
plt.plot(x_tr,training, label= 'observed')
plt.title('International International Arrivals from Jan 1998
to Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')
plt.legend(loc=2)


#1 Drift Method (Base Model)
first_observation = training[0]
last_observation = training[215]
T = len(training)

# Prediction
predicted_1 = []
for i in range (0,27):
    h = i + 1
    predicted_1.append(last_observation + h *
(last_observation - first_observation)/(T-1))
```

```
# Plot prediction
fig1 = plt.figure(figsize=(12,6))
plt.plot(x_tr,training,label='observed')
plt.plot(x_test,predicted_1,'-r', label='forecast')
plt.title('Drift Method: International Arrivals forecast for
Jan 2016 - Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')
plt.legend(loc=2)


#2 Seasonal holtwinter multiplicative
from holtwinters import multiplicative
# variable, we need convert a Series to a python list
ts = training.tolist()

# Now we define 99 predictions we wish to predict
fc = 27

m = 12
x_smoothed, m, Y, alpha, beta, gamma, rmse = ht.
multiplicative(ts,m,fc)

hh =x_smoothed[:-1]

# Prediction
predicted_2 = hh[216:]

# Plot prediction
fig1 = plt.figure(figsize=(12,6))
plt.plot(x_tr,training,label='observed')
plt.plot(x_test,predicted_2,'-r', label='forecast')
plt.title('Seasonal Holts Multiplicative: International
Arrivals forecast for Jan 2016 - Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')
plt.legend(loc=2)


# 3 Decomposition Method - Multiplicative Method
# To calculate the initial trend-cycle estimate we need to do
moving average smoothing
# m=12, 2x12 MA by chaining a 2-MA and a 12-MA as follows:
T_tr = training.rolling(2, center =
True).mean().rolling(12,center = True).mean()
T_tr = T_tr.shift(-1)

# Plot the initial trend estimate
plt.figure(figsize=(12,6))
plt.plot(x_tr,T_tr, label = 'observed')
plt.title('Initial TREND estimate_training set - International
Arrivals form Jan 1998 - Dec 2015')
plt.xlabel('Time')
plt.ylabel('No of Visitors')
plt.legend(loc=2)
```

```
# Calculate seasonal components
S_multiplicative_tr = training / T_tr

#Then plot the detrend series
plt.figure(figsize=(12,6))
plt.plot(x_tr, S_multiplicative_tr)
plt.title('detrend series_training set - International
Arrivals form Jan 1998 - Dec 2015')
plt.xlabel('Time')

#Calculate seasonally adjusted data
safe_S_tr= np.nan_to_num(S_multiplicative_tr)
monthly_S_tr= np.reshape(safe_S_tr,(18,12))
monthly_avg_tr= np.mean(monthly_S_tr[1:18,:], axis=0)

# constant C, M =12
c_tr= 12 / np.sum(monthly_avg_tr)

# montnly indicies, which is the seasonal component St
S_bar_tr = c_tr* monthly_avg_tr

# Repeat the average over 18 years
tiled_avg_tr= np.tile(S_bar_tr, 18)

# plot seasonal component
plt.figure(figsize=(12,6))
plt.plot(x_tr,tiled_avg_tr[:216])
plt.title('Seasonal Component_training set - Short Term
International Arrivals')
plt.xlabel('Time')

# seasonally adjusted series
seasonally_adjusted_tr = training / tiled_avg_tr

# plot seasonally adjusted series
plt.figure(figsize=(12,6))
plt.plot(x_tr,training, label = "observed")
plt.plot(x_tr,seasonally_adjusted_tr,label = 'seasonally
adjusted')
plt.title('Seasonally adjusted series_training set -
International Arrivals from Jan 1998 to Dec 2015')
plt.xlabel('Time')
plt.ylabel('No of Visitors')
plt.legend(loc=2)

# seasonally adjusted series regress with time
plt.figure(figsize=(12,6))
plt.title("seasonally adjusted series regress with time")
plt.scatter(x_tr,seasonally_adjusted_tr)
plt.xlabel("Month")
plt.ylabel("No of Visitors")
plt.show(block=False)

# use polynomail with degree = 3 to fit the trend
import numpy as np
```

```
X_poly_tr= np.arange(1, len(seasonally_adjusted_tr)+1, 1)
Y_poly_tr= seasonally_adjusted_tr


polymodel_fit_tr = np.polyfit(X_poly_tr,Y_poly_tr,3)
polymodel_tr = np.poly1d(polymodel_fit_tr)

# trend formula coefficient matrix
print(polymodel_fit_tr )

# trend formula
print(polymodel_tr )

x_polypred_tr = np.arange(1, len(seasonally_adjusted_tr)+1, 1)

poly_pred_tr= np.polyval(polymodel_fit_tr,x_polypred_tr)

# prediction made by trend on training set
trend_poly_tr = poly_pred_tr

# Plot the final trend by polynomail model (power = 3)
plt.figure(figsize=(12,6))
plt.plot(x_tr,training, label = "observed")
plt.plot(x_tr,seasonally_adjusted_tr,label = 'seasonally
adjusted')
plt.plot(x_tr,trend_poly_tr, label="Trend")
plt.legend()
plt.title("Final TREND estimate - International Arrivals from
Jan 1998 to Dec 2015")
plt.xlabel("Time")
plt.ylabel("No of Visitors")
plt.show(block=False)

# Predicted by trend * seasonality
x_polypred_test = np.arange(len(seasonally_adjusted_tr),
len(training)+1, 1)
poly_pred_test = np.polyval(polymodel_fit_tr,x_polypred_test)

prediction_ploy_seasonal_tr = poly_pred_test * tiled_avg_tr
[:27]

predicted_3 = prediction_ploy_seasonal_tr.tolist()

# Plot the prediction by decomposition method
plt.figure(figsize=(12,6))
plt.plot(x_tr,training,label='observed')
plt.plot(x_test,predicted_3, label='forecast by trend and
seasonality')
plt.legend(loc=2)
plt.title("International Arrivals Forecast for Jan 2016 - Mar
2018 by Decomposition Method")
plt.xlabel("Month")
plt.ylabel("No of Visitor")

# The residuals are fluctuated around 1
# The multiplicative model fits well
```

```
plt.figure(figsize=(12,6))
plt.title("Residual Plot")
plt.plot(x_tr, seasonally_adjusted_tr / trend_poly_tr)
plt.xlabel("Residuals")
plt.ylabel('Time')
plt.show(block=False)


# 4 sarima
import statsmodels as sm
import statsmodels.api as smt
from pandas.tools.plotting import autocorrelation_plot
from datetime import datetime

#check stationary for the original data
#original data
smt.graphics.tsa.plot_acf(training.values,lags=30,alpha = 0.05)
plt.title('ACF:Original Data')
smt.graphics.tsa.plot_pacf(training.values,lags=30,alpha =
0.05)
plt.title('PACF:Original Data')
#ACF dies down slowly, non-stationary

#take log (as magnitude is gradually increasing, like
multiplicative method)
training_log = np.log(training.values)

#get variance stablized
fig = plt.figure(figsize=(12,9))
ax1 = fig.add_subplot(211)
ax1.plot(training.values)
plt.title('No of Visitors')
ax2 = fig.add_subplot(212)
ax2.plot(training_log)
plt.title('Log of No of Visitors')

# Draw ACF and PACF
# For the log data
fig = plt.figure(figsize=(12,9))
ax1 = fig.add_subplot(211)
fig = smt.graphics.tsa.plot_acf(training_log, lags=40, ax=ax1)
ax1.set_title("ACF: Log Data")
ax2 = fig.add_subplot(212)
fig = smt.graphics.tsa.plot_pacf(training_log, lags=40, ax=ax2)
ax2.set_title("PACF: Log Data")

# get mean stablized
training_log_d = np.diff(training_log)

# plot the transformed data
fig = plt.figure(figsize=(12,9))
ax1 = fig.add_subplot(211)
ax1.plot(training_log)
plt.title('Log of No of Visitors')
ax2 = fig.add_subplot(212)
ax2.plot(training_log_d)
```

```
plt.title('Differencing Log of No of Visitors')

# Check stationality again
# Plot ACF AND PACF for the ordinary difference of log data
# clear seasonality at 12 24 36 spike in ACF, NOT Stationary
fig = plt.figure(figsize=(12,9))
ax1 = fig.add_subplot(211)
fig = smt.graphics.tsa.plot_acf(training_log_d, lags=40,
ax=ax1)
ax1.set_title("ACF: first order difference of Log data")
ax2 = fig.add_subplot(212)
fig = smt.graphics.tsa.plot_pacf(training_log_d, lags=40,
ax=ax2)
ax2.set_title("PACF: first order difference of Log data")

# data contains seasonal patterns, nonstationary
# Do seasonally differencing of log data
training_ds_log = training_log[12:] - training_log[:-12]
# Then first order difference
training_dsd_log = np.diff(training_ds_log);

# Plot seasonally differencing of log data
# Plot the first order difference on seasonally differenced
log data
fig = plt.figure(figsize=(12,9))
ax1 = fig.add_subplot(211)
ax1.plot(training_ds_log)
plt.title('Seasonally Differenced log data')
ax2 = fig.add_subplot(212)
ax2.plot(training_dsd_log)
plt.title('Regular Difference of the Seasonal Differenced log
data')

# Check stationality again
# Plot ACF and PACF for the seasonally differenced log data
fig = plt.figure(figsize=(12,9))
ax1 = fig.add_subplot(211)
fig = smt.graphics.tsa.plot_acf(training_ds_log, lags=40,
ax=ax1)
ax1.set_title("ACF: seasonal difference of log data")
ax2 = fig.add_subplot(212)
fig = smt.graphics.tsa.plot_pacf(training_ds_log, lags=40,
ax=ax2)
ax2.set_title("PACF: seasonal difference of log data")

# Check stationality again
# Plot ACF and PACF# For first order difference of seasonally
differenced log data
# Stationary series
fig = plt.figure(figsize=(12,9))
ax1 = fig.add_subplot(211)
fig = smt.graphics.tsa.plot_acf(training_dsd_log, lags=40,
ax=ax1)
ax1.set_title("ACF: differencing the seasonally differenced
Log data")
ax2 = fig.add_subplot(212)
```

```
fig = smt.graphics.tsa.plot_pacf(training_dsd_log, lags=40,
ax=ax2)
ax2.set_title("PACF: differencing the seasonally differenced
Log data")


#at seasonal level
#ACF cutoff at lag 1 or 3(MA(Q=1 or 3)
#PACF cutoff at lag 2 (AR(P=2)
#D =1
#only choose one, either Seasonal MA or Seasonal AR

#at normal level for log data
#ACF dies down after lag 1 or 4 or 5 (MA(q=1 or 4 or 5))
#PACF dies down after lag 1Ô°å2Ô°å4 (AR(p=1 or2 or4))
#d=1

# Define the model according to the identified pattern
# 1 Sarima(1,1,1)(2,1,0)12
sarima_model_1= smt.tsa.statespace.SARIMAX(training_log,
order=(1,1,1),seasonal_order=(2,1,0,12))

# Estimating the model
result_1= sarima_model_1.fit(disp=False)
print(result_1.summary())

# 2 Sarima(1,1,4)(2,1,0)12
sarima_model_2= smt.tsa.statespace.SARIMAX(training_log,
order=(1,1,4),seasonal_order=(2,1,0,12))
# Estimating the model
result_2= sarima_model_2.fit(disp=False)
print(result_2.summary())

# 3 Sarima(1,1,5)(2,1,0)12
sarima_model_3= smt.tsa.statespace.SARIMAX(training_log,
order=(1,1,5),seasonal_order=(2,1,0,12))
# Estimating the model
result_3= sarima_model_3.fit(disp=False)
print(result_3.summary())

# 4 Sarima(2,1,4)(0,1,1)12
sarima_model_4= smt.tsa.statespace.SARIMAX(training_log,
order=(2,1,4),seasonal_order=(0,1,1,12))
# Estimating the model
result_4= sarima_model_4.fit(disp=False)
print(result_4.summary())

# 5 Sarima(2,1,5)(0,1,1)12
sarima_model_5= smt.tsa.statespace.SARIMAX(training_log,
order=(2,1,5),seasonal_order=(0,1,1,12))
# Estimating the model
result_5= sarima_model_5.fit(disp=False)
print(result_5.summary())

# 6 Sarima(4,1,5)(0,1,1)12
```

```
sarima_model_6= smt.tsa.statespace.SARIMAX(training_log,
order=(4,1,5),seasonal_order=(0,1,1,12))
# Estimating the model
result_6= sarima_model_6.fit(disp=False)
print(result_6.summary())

# 7 Sarima(1,1,4)(0,1,3)12
sarima_model_7= smt.tsa.statespace.SARIMAX(training_log,
order=(1,1,4),seasonal_order=(0,1,3,12))
# Estimating the model
result_7= sarima_model_7.fit(disp=False)
print(result_7.summary())

# based on AIC, model #2 Sarima(1,1,4)(2,1,0)12 is the best
(AIC MIN)
# Forecasting
forecasts_sarima = result_2.forecast(27)
predicted_4 = np.exp(forecasts_sarima).tolist()

# Display forecasting on log data
fig = plt.figure(figsize=(12,6))
plt.plot(x_tr,training_log,label='log transformed data')
plt.plot(x_test,forecasts_sarima,label='forecast by SARIMA on
log data')
plt.title('International Arrivals Forecast for Jan 2016 - Mar
2018 by SARIMA(1,1,4)(2,1,0)12' )
plt.legend(loc=2)

# Display forecasting on original data
fig = plt.figure(figsize=(12,6))
plt.plot(x_tr,training,label='observed')
plt.plot(x_test,predicted_4,label='forecast by SARIMA on
original data')
plt.title('International Arrivals Forecast for Jan 2016 - Mar
2018 by SARIMA(1,1,4)(2,1,0)12' )
plt.legend(loc=2)

# Model accuracy
# Evaluate the model performance
def rmse (x,y):
    return np.sqrt(np.average(np.power(x-y,2)))

def mad (x,y):
    return np.average(np.abs(x-y))

def mape (x,y):
    return np.average(((np.abs(x-y))/x)*100)
#1
rmse_drift = rmse (test, predicted_1)
print ("the RMSE of Drift Method is {0}".format(rmse_drift))
mad_drift = mad (test, predicted_1)
print ("the MAD of Drift Method is {0}".format(mad_drift))
mape_drift = mape (test, predicted_1)
print ("the MAPE of Drift Method is {0}".format(mape_drift))
var_DriftMethod = np.var(test-predicted_1)
```

```
#2
rmse_SeasonalHoltsMultiplicative = rmse (test, predicted_2)
print ("the RMSE of Seasonal Holts Multiplicative is
{0}".format(rmse_SeasonalHoltsMultiplicative))
mad_SeasonalHoltsMultiplicative = mad (test, predicted_2)
print ("the MAD of Seasonal Holts Multiplicative is
{0}".format(mad_SeasonalHoltsMultiplicative))
mape_SeasonalHoltsMultiplicative= mape (test, predicted_2)
print ("the MAPE of Seasonal Holts Multiplicative is
{0}".format(mape_SeasonalHoltsMultiplicative))
var_SeasonalHoltsMultiplicative = np.var(test- predicted_2)

#3 Decomposition Method
rmse_DecompositionMultiplicative = rmse (test, predicted_3)
print ("the RMSE of Decomposition Method is
{0}".format(rmse_DecompositionMultiplicative))
mad_DecompositionMultiplicative = mad (test, predicted_3)
print ("the MAD of Decomposition Method is
{0}".format(mad_DecompositionMultiplicative))
mape_DecompositionMultiplicative = mape (test, predicted_3)
print ("the MAPE of Decomposition Method is
{0}".format(mape_DecompositionMultiplicative))
var_DecompositionMethod = np.var(test-predicted_3)

#4
rmse_SARIMA = rmse (test, predicted_4)
print ("the RMSE of SARIMA is {0}".format(rmse_SARIMA))
mad_SARIMA = mad (test, predicted_4)
print ("the MAD of SARIMA is {0}".format(mad_SARIMA))
mape_SARIMA = mape (test, predicted_4)
print ("the MAPE of SARIMA is {0}".format(mape_SARIMA))
var_SARIMA = np.var(test- predicted_4)

# Based on aboved rmse,ranking is as below:
# #1 Sarima #2 Decomposition Method #3Seasonal Holts
Multiplicative #4 Drift


# 5 combination of models
# combination will be based on only 3 models (benchmark, drift
method will not be included)
# 5.1 equally weighted
weight_equal = 1/3

forecast_combination_1 = weight_equal*
(pd.Series(predicted_2)+pd.Series(predicted_3)+pd.Series(predi
cted_4))

predicted_5_1 = forecast_combination_1.tolist()

# 5.1 equally-weighted combination
rmse_forecast_combination_1 = rmse (test, predicted_5_1)
print ("the RMSE of equally-weighted combination is
{0}".format(rmse_forecast_combination_1))
mad_forecast_combination_1 = mad (test, predicted_5_1)
```

```
print ("the MAD of equally-weighted combination is
{0}".format(mad_forecast_combination_1))
mape_forecast_combination_1 = mape (test, predicted_5_1)
print ("the MAPE of equally-weighted combination is
{0}".format(mape_forecast_combination_1))


# 5.2 optimal weights
#residuals for selected 2 models (rmse min:sarima and
decomposition)
residual_sarima = test - predicted_4
residual_decomposition = test - predicted_3

#covariance (will be a matrix)
covariance = np.cov(residual_sarima,residual_decomposition)

var_sarima = covariance[0,0]
var_decomposition = covariance[1,1]

r =  covariance [0,1] / (np.sqrt(var_sarima*var_decomposition))

#variance optimization weights
w_sarima = (var_decomposition -
r*np.sqrt(var_sarima*var_decomposition)) / (var_sarima
+var_decomposition-2*r*np.sqrt(var_sarima*var_decomposition))
w_decomposition = 1 - w_sarima

forecast_combination_2 = w_sarima * pd.Series(predicted_4)  +
w_decomposition * pd.Series(predicted_3)
predicted_5_2 = forecast_combination_2.tolist()

# 5.2 optimal weight combination
rmse_forecast_combination_2 = rmse (test, predicted_5_2)
print ("the RMSE of optimal weight combination is
{0}".format(rmse_forecast_combination_2))
mad_forecast_combination_2 = mad (test, predicted_5_2)
print ("the MAD of optimal weight combination is
{0}".format(mad_forecast_combination_2))
mape_forecast_combination_2 = mape (test, predicted_5_2)
print ("the MAPE of optimal weight combination is
{0}".format(mape_forecast_combination_2))

from statistics import variance
var_forecast_combination_2 = variance (test - predicted_5_2)

# 5.3 weights proportionally inverse to rmse
sum_inversermse = 1/rmse_SeasonalHoltsMultiplicative +
1/rmse_DecompositionMultiplicative + 1/rmse_SARIMA
weight_SARIMA = (1/rmse_SARIMA)/sum_inversermse
weight_DecompositionMultiplicative =
(1/rmse_DecompositionMultiplicative)/sum_inversermse
weight_SeasonalHoltsMultiplicative =
(1/rmse_SeasonalHoltsMultiplicative )/sum_inversermse
```

```
forecast_combination_3 = weight_SARIMA * pd.Series(predicted_4)
+ weight_DecompositionMultiplicative * pd.Series(predicted_3)+
weight_SeasonalHoltsMultiplicative*pd.Series(predicted_2)
predicted_5_3 = forecast_combination_3.tolist()

# 5.3 weights proportionally inverse to rmse
rmse_forecast_combination_3 = rmse (test, predicted_5_3)
print ("the RMSE of weights proportionally inverse to rmse
combination is {0}".format(rmse_forecast_combination_3))
mad_forecast_combination_3 = mad (test, predicted_5_3)
print ("the MAD of weights proportionally inverse to rmse
combination is {0}".format(mad_forecast_combination_3))
mape_forecast_combination_3 = mape (test, predicted_5_3)
print ("the MAPE of weights proportionally inverse to rmse
combination is {0}".format(mape_forecast_combination_3))

# 5.4 weights inverse to their ranks
# rank1 sarima rank2 decomposition # rank3 holts
sum_rankinverse = 1 + 1/2 + 1/3
w_SARIMA = 1 / sum_rankinverse
w_DECOMPOSITION = (1/2) / sum_rankinverse
w_HOLTS = (1/3)/sum_rankinverse

forecast_combination_4 = w_SARIMA * pd.Series(predicted_4)  +
w_DECOMPOSITION  * pd.Series(predicted_3)+
w_HOLTS*pd.Series(predicted_2)
predicted_5_4 = forecast_combination_4.tolist()

# 5.4 weights inverse to their ranks
rmse_forecast_combination_4 = rmse (test, predicted_5_4)
print ("the RMSE of weights inverse to their ranks is
{0}".format(rmse_forecast_combination_4))
mad_forecast_combination_4 = mad (test, predicted_5_4)
print ("the MAD of weights inverse to their ranksis
{0}".format(mad_forecast_combination_4))
mape_forecast_combination_4 = mape (test, predicted_5_4)
print ("the MAPE of weights inverse to their ranks is
{0}".format(mape_forecast_combination_4))


# Based on RMSE, combination_2 is the best, followed by
combination 3 combination 4 and combination 1.
# combination 2, it's actually the formula to optimize the
portfolio variance, not rmse
# combination 2 has higher rmse than sarima and higher
variance than sarima and decomposition

# overall
# sarima is the best
# the forecast for the next year will be based on
Sarima(1,1,4)(2,1,0)12
```

```
# PART C - SARIMA forecast from Apr 18 to Mar 19

"""

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import holtwinters as ht
import datetime as dt

#########
# Forecast for the time from Apr 2018 to Mar 2019 based on all
sample data from Jan 1998 to Mar 2018
# Load the dataset; Data from Jan 1998 to Mar 2018
visitors = pd.read_csv('InternationalArrivals.csv')

# Seperate the time and variable y
Y_a=numVisitors = visitors[visitors.columns[1]]
X_a= months = visitors[visitors.columns[0]]

# Training set Jan 1998 - Dec 2015 & test set Jan 2016 - Mar
2018
training = numVisitors[:216]
test = numVisitors[216:]

months_tr = months[:216]
months_test = months[216:]

# Product time for plotting purpose
x_a = np.array([dt.datetime.strptime(d, '%b-%y') for d in
months])
x_tr = np.array([dt.datetime.strptime(d, '%b-%y') for d in
months_tr])
x_test = np.array([dt.datetime.strptime(d, '%b-%y') for d in
months_test])

# Prediction Year
xp = np.array([dt.datetime.strptime(d, '%b-%y') for d in
('Apr-18','May-18','Jun-18','Jul-18','Aug-18','Sep-18','Oct-
18','Nov-18','Dec-18','Jan-19','Feb-19','Mar-19')])
xp_a = np.hstack((x_a,xp))

# Plot the original test set
plt.figure(figsize=(12,6))
plt.plot(x_a, numVisitors, label= 'observed')
plt.title('International International Arrivals from Jan 1998
to Mar 2018')
plt.xlabel('Time')
plt.ylabel('No of Visitors')
plt.legend(loc=2)


# 4 sarima
import statsmodels as sm
import statsmodels.api as smt
```

```
from pandas.tools.plotting import autocorrelation_plot
from datetime import datetime

#check stationary for the original data
#original data
smt.graphics.tsa.plot_acf(Y_a.values,lags=30,alpha = 0.05)
plt.title('ACF:Original Data')
smt.graphics.tsa.plot_pacf(Y_a.values,lags=30,alpha = 0.05)
plt.title('PACF:Original Data')
# non-stationary

#take log (as magnitude is gradually increasing, like
multiplicative method)
Y_a_log = np.log(Y_a.values)

# best Sarima(1,1,4)(2,1,0)12, order from training test
sarima_model_a= smt.tsa.statespace.SARIMAX(Y_a_log,
order=(1,1,4),seasonal_order=(2,1,0,12))
# Estimating the model
result_a= sarima_model_a.fit(disp=False)
print(result_a.summary())

forecasts_sarima_a = result_a.forecast(12)
predicted_4_a = np.exp(forecasts_sarima_a)

# Display forecasting on log data
fig = plt.figure(figsize=(12,6))
plt.plot(x_a,Y_a_log,label='log transformed data')
plt.plot(xp,forecasts_sarima_a,label='forecast by SARIMA on
log data')
plt.title('International Arrivals Forecast from Apr 2018 - Mar
2019 by SARIMA(1,1,4)(2,1,0)12' )
plt.legend(loc=2)

# Display forecasting on original data
fig = plt.figure(figsize=(12,6))
plt.plot(x_a,Y_a,label='observed')
plt.plot(xp,predicted_4_a,label='forecast by SARIMA on
original data')
plt.title('International Arrivals Forecast from Apr 2018 - Mar
2019 by SARIMA(1,1,4)(2,1,0)12' )
plt.legend(loc=2)

fig = plt.figure(figsize=(12,6))
plt.plot(xp,predicted_4_a,label='forecast by SARIMA on
original data')
plt.title('International Arrivals Forecast from Apr 2018 - Mar
2019 by SARIMA(1,1,4)(2,1,0)12' )
plt.legend(loc=2)
```

2. Meeting Agenda and Minutes

2.1 Meeting Agenda

TEAM TASK MEETING AGENDA

TEAM MEETING AGENDA

________________QBUS 6840 Group 69____**(Company Name)**

Meeting to be held __________Fisher Library_______________(Where)

_________________________14/May/2018_________ (Date)

________2:00pm – 3:22pm________(Time)

Chairperson: ________480303419______________________

Minute-Taker: 460071662

1. Apologies: NONE
2. Confirmation of agenda (x minutes)(Chair)
3. Confirmation of minutes of __13/May/2018_______________ (Date) (x minutes)(Chair)
4. Business arising from minutes of ____13/May/2018_______ (Date) (x minutes)(Chair)
5. Items

   (1) Go through the assignment

   (2) Narrow down the alternatives of topics

   (3)

   (4)

   (5)

   (6)
6. Any other business (x minutes)(Chair)
7. Forward agenda items (x minutes)(Chair)
8. Next meeting : 18/MAY/2018

(Chair)

TEAM MEETING AGENDA

_________________QBUS 6840 Group 69____**(Company Name)**

Meeting to be held ___________Fisher Library________________(Where)

__________________________18/May/2018__________ (Date)

_______________1:00pm-3:22pm_____(Time)

Chairperson: _____460487999______________

Minute-Taker: 470200568

1. Apologies: NONE
2. Confirmation of agenda (x minutes)(Chair)
3. Confirmation of minutes of __14/May/2018_________________ (Date) (x minutes)(Chair)
4. Business arising from minutes of _____14/May/2018________ (Date) (x minutes)(Chair)
5. Items

   (1) Go through the assignment

   (2) Narrow down the alternatives of topics

   (3)

   (4)

   (5)

   (6)
6. Any other business (x minutes)(Chair)
7. Forward agenda items (x minutes)(Chair)
8. Next meeting : 20/MAY/2018

(Chair)

TEAM MEETING AGENDA

________________QBUS 6840 Group 69____**(Company Name)**

Meeting to be held ___________Fisher Library________________(Where)

__________________________20/May/2018__________ (Date)

__________9:00am -1:27pm__________(Time)

Chairperson: ___470200568________________

Minute-Taker: 460487999

1. Apologies: NONE
2. Confirmation of agenda (x minutes)(Chair)
3. Confirmation of minutes of __18/May/2018________________ (Date) (x minutes)(Chair)
4. Business arising from minutes of _____18/May/2018_______ (Date) (x minutes)(Chair)
5. Items

   (1) Find the reference backing up the characteristics of plots

   (2) Determine the forecasting models

   (3)

   (4)

   (5)

   (6)
6. Any other business (x minutes)(Chair)
7. Forward agenda items (x minutes)(Chair)
8. Next meeting : 24/MAY/2018 (Chair)

TEAM TASK MEETING AGENDA

TEAM MEETING AGENDA

QBUS 6840 Group 69 **(Company Name)**

Meeting to be held Fisher Library (Where)

24/MAY/2018 (Date)

3:00pm-5:08pm (Time)

Chairperson: 460071662

Minute-Taker: 480303419

1. Apologies: None
2. Confirmation of agenda (x minutes)(Chair)
3. Confirmation of minutes of 22/MAY/2018 (Date) (x minutes)(Chair)
4. Business arising from minutes of 23/MAY/2018 (Date) (x minutes)(Chair)
5. Items

   (1) Further study different models (definition & formulas)

   (2) Analyze and compare the advantages and disadvantages of the models

   (3)

   (4)

   (5)

   (6)
6. Any other business (x minutes)(Chair)
7. Forward agenda items (x minutes)(Chair)
8. Next meeting 27/MAY/2018 (Chair)

THE UNIVERSITY OF SYDNEY

TEAM TASK MEETING AGENDA

TEAM MEETING AGENDA

QBUS 6840 Group 69 **(Company Name)**

Meeting to be held ABS (Where)

27/MAY/2018 (Date)

2:30pm-5:18pm (Time)

Chairperson: 460071662

Minute-Taker: 480303419

1. Apologies: None
2. Confirmation of agenda (x minutes)(Chair)
3. Confirmation of minutes of 26/MAY/2018 (Date) (x minutes)(Chair)
4. Business arising from minutes of 27/MAY/2018 (Date) (x minutes)(Chair)
5. Items
   (1) Reflect the limitations of analysis.
   (2) Adjust the structure of the report for accuracy and consistency.
   (3)
   (4)
   (5)
   (6)
6. Any other business (x minutes)(Chair)
7. Forward agenda items (x minutes) (Chair)
8. Next meeting this is the last meeting (Chair)

2.2 Meeting Minutes

MINUTES TEMPLATE

Minutes of meeting for QBUS 6840 Group 69

Date: 14/MAY/2018 Time: 2:00pm – 3:22pm Location: Fisher Library

Chairperson: 480303419

Minute-Taker: 460071662

Document tabled: Group Meeting 1

Present: 480303419, 460071662, 460487999, 470200568

Apologies: None

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
1. Go through the assignment	* Team building	Self-introduction	480303419	2:00pm – 2:08pm	Be nice and a good listener.
	* Read through and understand the requirements	Read through the criteria of group assignment	460071662	2:10pm – 2:38pm	
2. Narrow down the alternatives of topics	* Briefly outline the forecasting steps	Research about Australia's tourism industry	460487999	2:42pm – 3:00pm	Present ideas actively and confidently.
	* Summarize the forecasting models	Brainstorming the problem definition	470200568	3:08pm – 3:22pm	

Souce: TAFE Access Division "Communication for Business", 2000

MINUTES TEMPLATE

Minutes of meeting for QBUS 6840 Group 69

Date: 18/MAY/2018 Time: 1:00pm-3:22pm Location: Fisher Library

Chairperson: 460487999

Minute-Taker: 470200568

Document tabled: Group Meeting 2

Present: 480303419, 460071662, 460487999, 470200568

Apologies: None

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
1. Obtain information	* Data source	Download the international arrival from ABS website from Jan 1998 to Mar 2018	460487999	1:00pm – 1:32pm	State the opinions efficiently and effectively.
	* Determine the time series		470200568	1:46pm – 2:23pm	
2. Discuss the structure of the report	* problem definition	Search Australia's tourism industry year books	480303419	2:42pm – 3:06pm	Create a friendly atmosphere for discussion.
	* Exploratory Data Analysis	Describe the plot of original data	460071662	3:10pm – 3:22pm	

Souce: TAFE Access Division "Communication for Business", 2000

MINUTES TEMPLATE

Minutes of meeting for QBUS 6840 Group 69

Date: 20/MAY/2018 Time: 9:00am -1:27pm Location: ABS

Chairperson: 470200568

Minute-Taker: 460487999

Document tabled: Group Meeting 3

Present: 480303419, 460071662, 460487999, 470200568

Apologies: None

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
1. Find the reference backing up the characteristics of plots	* Find out the declining reasons in plot	Research the Australia's tourism in 2003	460071662	9:00am – 9:45pm	Clarify a point of view while drawing a logical flowchart.
	* Looking for methods to process the time-series data	Apply decomposition methods to run	470200568	10:11am – 11:03am	
2. Determine the forecasting models	* Highlight the forecasting models that we learned from the lectures	Go through the tutorial materials	480303419	11:13pm – 11:50pm	
		Start with using simple models	460487999	12:03pm – 1:27pm	

Souce: TAFE Access Division "Communication for Business", 2000

MINUTES TEMPLATE

Minutes of meeting for QBUS 6840 Group 69

Date: 24/MAY/2018 Time: 3:00pm-5:08pm Location: Fisher Library

Chairperson: 460071662

Minute-Taker: 480303419

Document tabled: Group Meeting 4

Present: 480303419, 460071662, 460487999, 470200568

Apologies: None

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
1. Further study different models	* The definition of models	Go through the online textbook	480303419	3:00pm – 3:42pm	Summarize ideas and briefly communicate with key points.
	* The assumptions and formulas	Explain the definition in our words	460071662	3:45pm – 4:13pm	
2. Analyze and compare the advantages and disadvantages of the models	* Advantages and disadvantages	Go through the lecture slides and summarize advantages and disadvantages	460487999 470200568	4:23pm – 5:08pm	

Souce: TAFE Access Division "Communication for Business", 2000

MINUTES TEMPLATE

Minutes of meeting for QBUS 6840 Group 69

Date: 27/MAY/2018 Time: 2:30pm-5:18pm Location: FISHER LIBRARY

Chairperson: 460071662

Minute-Taker: 480303419

Document tabled: Group Meeting 5

Present: 480303419, 460071662, 460487999, 470200568

Apologies: None

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
1. Reflect the limitations of analysis	* Accuracy of the outcome * Objectiveness of the report	Critical thinking in the context of real life experience, researching about the suggestions	460487999 470200568 480303419	2:30pm – 3:43pm	Be proactive and engage as more perspectives as possible.
2. Adjust the structure of the report	* Consistency	Add transitional paragraphs with logic	460071662	4:04pm – 5:18pm	Streamline the whole process and be open-minded.

Souce: TAFE Access Division "Communication for Business", 2000