STA9701 Time Series Analysis on

TOURIST ARRIVALS IN NEPAL

Annapurna Mountain Range[1]

Anustha Shrestha
Baruch College

November 14, 2019

[1] Neupane, Avishek (October 2017)

Introduction

Nepal is a small landlocked country situated between India and China. Although the country is primarily an agricultural economy, one of the promising sectors and a major source of income and foreign exchange is tourism. The country's rich geography—ranging from the highest mountains in the world to the low-lying subtropical forests that is home to endangered Bengal tigers and one-horned rhinoceros—the cultural heritage, and religious sites, attract tourists every year.

In 2018, annual tourist arrivals in Nepal reached a million mark for the first time with 1.17 million tourists who visited the country during that year[2]. In order to celebrate and promote tourism in the country, Nepal has designated, the year 2020 as Visit Nepal Year. Understanding the pattern of tourist flow into the country and forecasting for tourist arrivals can help local businesses prepare and provide exceptional service to the visitors. Especially, for Visit Nepal Year, preparation is absolutely essential as tourists will have high expectations of good hospitality and excellent service.

Therefore, in this project, I will use monthly data of tourist arrivals in Nepal, starting from January 1992 to December 2018 and try to forecast monthly tourist arrivals for the years 2019 and 2020. Tourism in Nepal is seasonal; although tourists visit the country all year round, the popular seasons are when it is neither too hot, cold or rainy. The window of visit is especially narrow for tourists who come for activities like mountaineering, trekking and rafting. The popular seasons are during the Spring between March through June and in the Fall, between September through November. However, for this project, I plan to prioritize simplicity of the model and use simple ARMA or ARIMA models that captures the essence of the time series, without using multiplicative seasonal model or seasonal differencing. Eventually, in future projects, I will try to fit seasonal models and compare it against the simple ARIMA model that I come up with in this project.

Data Source

Nepal's tourism dataset has been retrieved from Nepal Tourism Statistics 2018 published by the Government of Nepal Ministry of Culture, Tourism and Civil Aviation[3]. The dataset contains monthly tourist arrivals in Nepal starting from January 1992 to December 2018. Altogether there are 324 observations over the years.

Summary

The objective of the project is to use monthly tourist arrivals from January 1992 to December 2018 to fit a simple ARIMA model i.e. AR, MA, ARMA or ARIMA model without seasonal differencing or seasonal multiplicative ARIMA models. The fitted model will be used to forecast for 24 periods, i.e. for the years 2019 and 2020. When forecasting, the project aims to select simplest models that captures the characteristics of the data well.

In order to make the time series stationary with constant mean and stable variance, log transformation and differencing (d=1) was applied. The resulting value can simply be interpreted as monthly growth rate of tourist arrivals. After analyzing the ACF and PACF structures of the log and differenced data, a few selections of p and q order were made to test a few ARIMA models. Out of the candidates, the best model selected was ARIMA (12, 1, 1). Since not all the coefficients were significant, a restricted ARIMA (12, 1, 1) was used to perform the final forecasting.

Although the final forecast seems to capture the peak and falls of the time series well, this model is simplistic, and may not capture the seasonality that is characteristic of a tourism-related data. Testing the predictions against the real 9-month data for 2019 shows that the model is performing better than when we forecast using a basic white noise model, and is doing a good job capturing the pattern that the real time series follows; however, the predictions are more optimistic i.e. the predicted tourist arrivals are slightly higher than the real tourist arrivals. Future developments of this project may include incorporating seasonal differencing and identifying seasonal models. Furthermore, the data also include some outliers; future developments might also include accounting for such one-time occurrences.

[2] Government of Nepal Ministry of Culture, Tourism and Civil Aviation. (2018). Nepal Tourism Statistics. Retrieved from: http://tourism.gov.np/statistic
[3] Government of Nepal Ministry of Culture, Tourism and Civil Aviation. (2018). Nepal Tourism Statistics. Retrieved from: http://tourism.gov.np/statistic

Exploratory Data Analysis

Data Visualization

We first plot to see the time series data, which helps us identify trends, patterns and any issues with the dataset. Figure1 shows the tourist arrivals in Nepal starting from January 1992 to December 2018. From the plot, it can be noted that the tourist arrivals in the country has increased over the period. The increasing trend indicates non-constant mean over time; hence, the data is most likely not stationary. Furthermore, it can also be seen that the variance also is not stable over time implying that the time series data needs to be transformed. The data also has potential outliers – in 2015, we can notice a big dip in the number of tourists, which is because of the earthquakes that year. Although it doesn't look very clear at first, tourist arrivals slowed down between 2001 to 2008 due to the Maoist insurgency and a civil unrest in the country making it unsafe for travel.

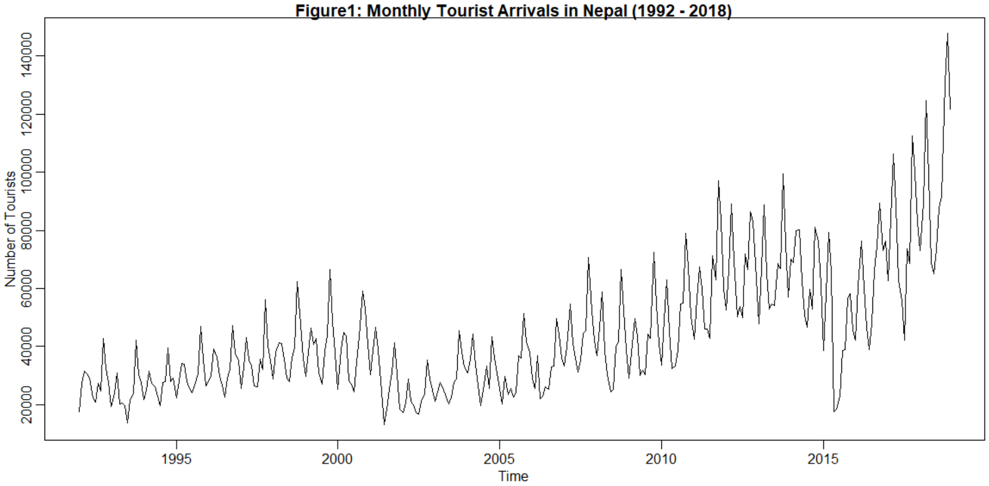


Figure1: Monthly Tourist Arrivals in Nepal (1992 - 2018)

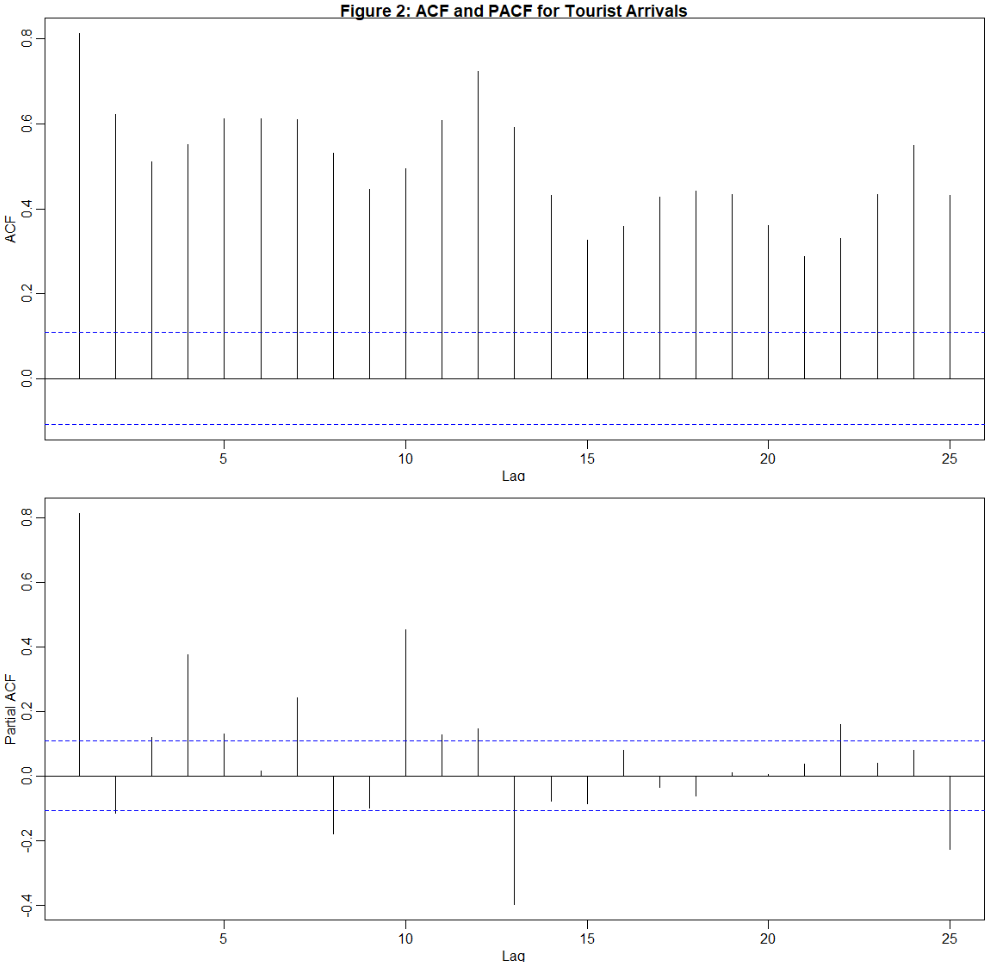


Figure 2: ACF and PACF for Tourist Arrivals

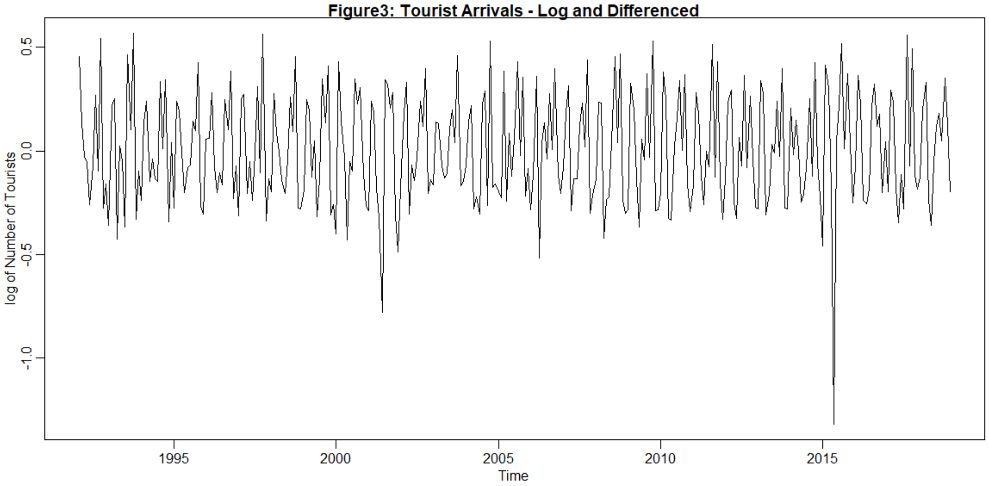
As noted earlier, the slowly tailing off ACF in Figure2 is indicative of a nonstationary behavior. Furthermore, the rise and fall in the ACF, which resembles a cyclical pattern suggests seasonality. It is not surprising that tourism follow a seasonal pattern in a country like Nepal, where large number of tourist activities is based on seasons—the peak season for mountaineering is around Spring, which brings a considerable number of tourists around that time. Nepal also has a good weather around Fall, which is another popular season for tourists. Winters and Summers are usually slow periods in terms of tourist flows.

Data Transformation and Differencing

Since the plots shows a non-stationary behavior and the variance does not look stable, Box-Cox Lambda function in *R* was used to examine if any transformation is needed. The Box-Cox Lambda function returned λ = 0.0353, which suggests log transformation. In order to remove the trend, differencing was applied to the data and the resulting plot is seen in Figure3. Log and differencing the data also has an added advantage of easy interpretation. The resulting value is just the percentage change in number of tourist arrivals every month.

$$\nabla \log(x_t) = (1-B)\log(x_t) = \log(x_t) - \log(x_{t-1})$$

Figure3: Tourist Arrivals - Log and Differenced



It is important to note that as suggested earlier, the outliers seem more prominent in this plot. However, I decided to keep these potential outliers in the current project because natural disasters like earthquakes are common in Nepal. Nepal is in an earthquake prone zone so there are always risks of earthquakes that might impact tourism at all time. Therefore, I wanted to review a model that included these types of volatility.

Nevertheless, it is important to check for stationarity. Dickey-fuller test (k=0, p-value = 0.01) and augmented Dickey-Fuller test (k=6, p-value = 0.01) both are statistically significant showing that log and differenced time series is stationary.

```
Augmented Dickey-Fuller Test

data:  dlogtourist
Dickey-Fuller = -18.226, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary

Augmented Dickey-Fuller Test

data:  dlogtourist
Dickey-Fuller = -8.6361, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

Model Selection

Analyzing ACF and PACF of transformed data

Figure4 shows the ACF and PACF of the Log and Differenced Tourist Arrivals. ACF is gradually tailing off with the values significant at 6th and 12th lags only. This is indicative of seasonality. The PACF seems to be cut off after 12th lag (Note that the PACF is also significant at 24th lag, but not as significant as some of the PACF before 12th lag).

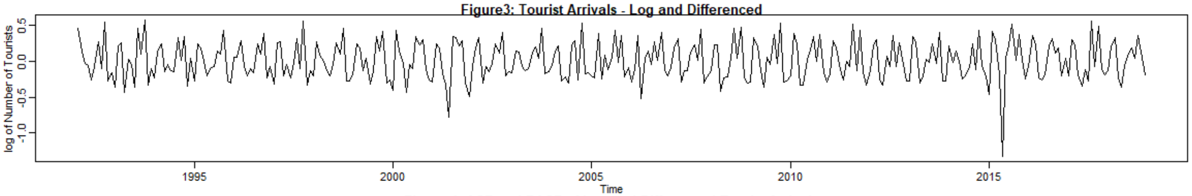


Figure3: Tourist Arrivals - Log and Differenced

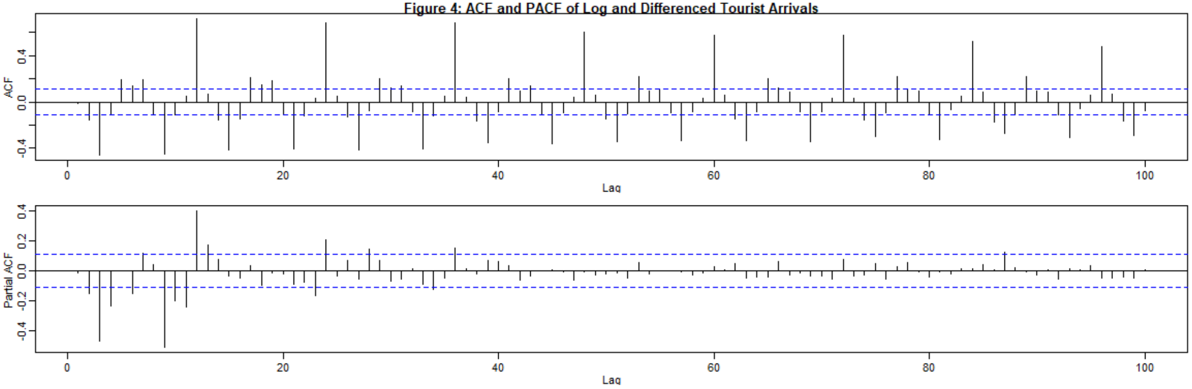


Figure 4: ACF and PACF of Log and Differenced Tourist Arrivals

Since both the ACF and PACF is gradually decreasing, ARIMA models are going to be helpful. Since PACF seems to be cut off at lag 12, order p=12 needs to be explored. Similarly, since ACF values at every 6th lag is also significant, we can also try some models with p=6. Furthermore, simplicity is another aspect that needs to be kept into mind and I wanted to try some simple ARIMA models with the orders p and q less than 12. In order to select simple models, Extended Autocorrelation Function (eacf) was used in *R* to identify some simple ARIMA models. The candidate models picked are listed below along with the respective AIC and BIC:

Table1: Model Selection

Candidate Models: AIC and BIC		
Model	**AIC**	**BIC**
ARIMA (1, 1, 1)	10.08048	25.19109
ARIMA (2, 1, 1)	-13.92555	4.962708
ARIMA (2, 1, 0)	59.95337	75.06398
ARIMA (1, 1, 0)	65.9178	77.25075
ARIMA (12, 1, 0)	-235.1842	-182.2971
ARIMA (12, 1, 1)	-249.7707	-193.1059
ARIMA (12, 1, 2)	-250.5256	-190.0831
ARIMA (6, 1, 0)	-42.78451	-12.56329
ARIMA (6, 1, 2)	-47.40584	-9.629319

Based on the AIC and BIC values for each candidate model provided on Table1, the final three candidates selected were ARIMA (12, 1, 1), ARIMA (2, 1, 1) and ARIMA (1, 1, 1).

- ARIMA (12, 1, 1) has the best BIC; the AIC is not very different than that of ARIMA (12, 1, 2), which has the best AIC.
- Out of the simpler models i.e. the models with p and q that are either 0, 1, or 2, ARIMA (2, 1, 1) is the best model
- ARIMA (1, 1, 1) was selected as the simplest model for comparison

Model Diagnostics

The diagnostic plots for all the three models are provided in Figure5, Figure6 and Figure7. The residuals from the diagnostic plots are mostly spread around 0 for all three models, except for the outliers we had noted earlier. The Q-Q plot checks for the normality of the residuals. The Q-Q plot looks decent, with a few outliers.

From the diagnostic plots for the simple models, ARIMA (1, 1, 1) and ARIMA (2, 1, 1) (shown in Figure5 and Figure6), we can see that the ACF of the residuals are significant at multiple lags, which shows that the residuals are highly correlated. Similarly, the p-value for Ljung-Box statistic are also close to 0, which shows that the correlation is significant. In fact, most of the simple models fails when it comes to testing the ACF of the residuals.

The ACF of residuals for ARIMA (12, 1, 1) is much better; still the ACF at lag 12 is significant. The p-values of Ljung-Box statistic are still problematic for this model as well.

Figure5: Diagnostics - ARIMA (1, 1, 1)

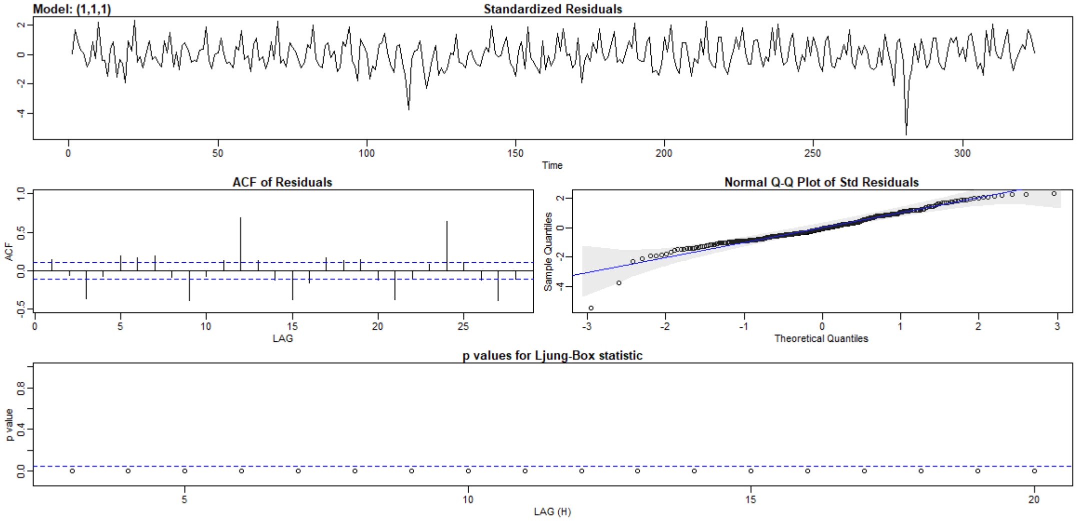


Figure6: Diagnostics - ARIMA (2, 1, 1)

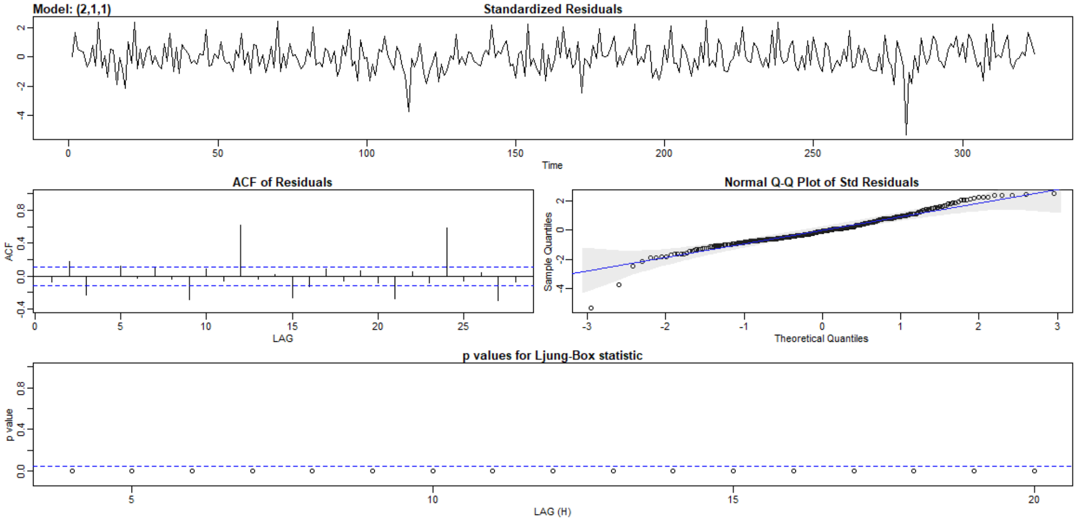
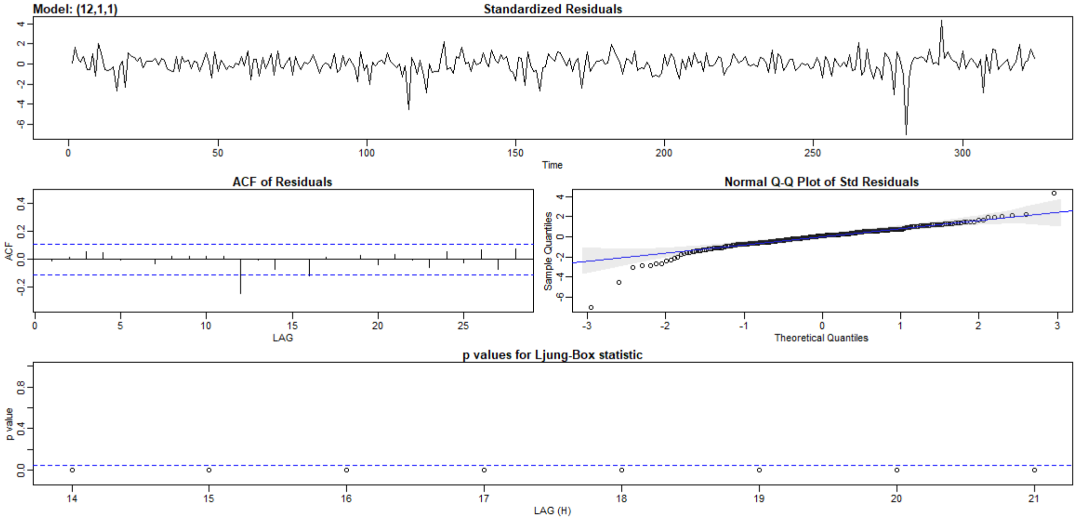


Figure7: Diagnostics - ARIMA (12, 1, 1)



Final Model Selection and Forecasting

From the diagnostics, the best model is ARIMA (12, 1, 1). Therefore, future 24 observation were predicted using ARIMA (12, 1, 1) model. Figure8 shows the forecast with confidence intervals for 24 periods (2019 and 2020). This model seems to capture the variability at each lag better than the forecasts in Figure9, which was built using ARIMA(1, 1, 1) and ARIMA(2, 1, 1).

Figure8: Forecasting for 24 periods- ARIMA (12, 1, 1)

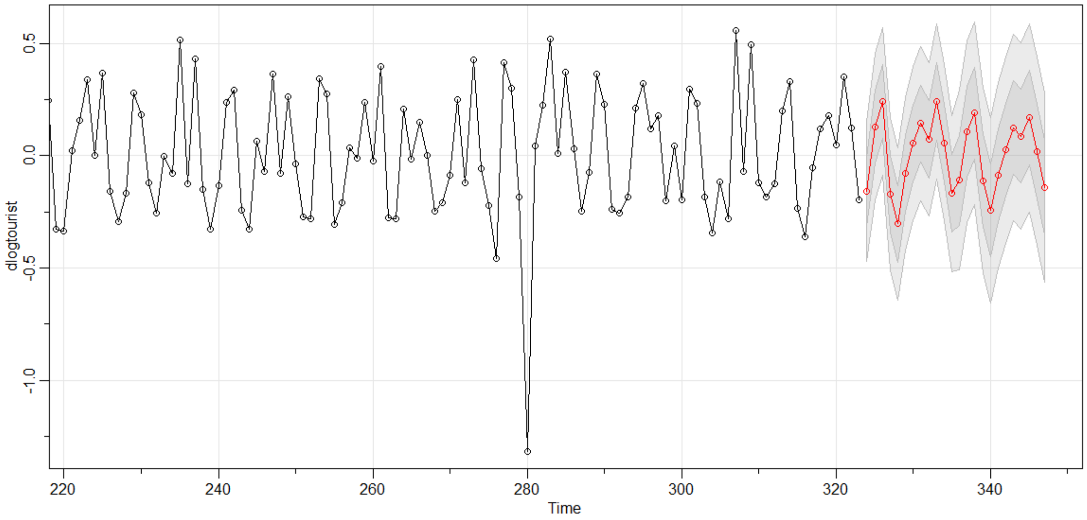
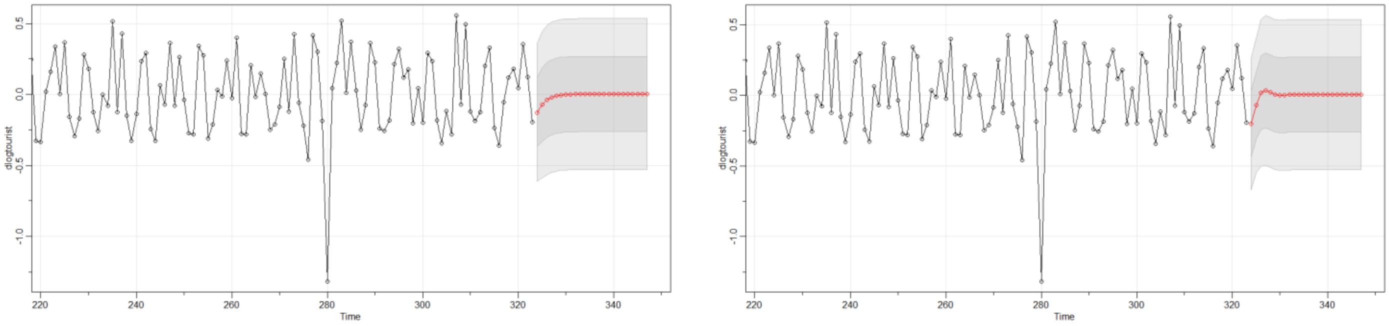


Figure9: Forecasting for 24 periods using Simple models

ARIMA (1, 1, 1) **ARIMA (2, 1, 1)**



The resulting ARIMA (12, 1, 1) model estimates shows that some of the coefficients were not statistically significant, therefore, a Restricted ARIMA model was used. The final model is given below along with the coefficient estimates:

Restricted ARIMA (12, 1, 1):

$$\boldsymbol{\phi(B)(1-B)(X_t) = \ \theta(B)w_t}$$

where,

$\boldsymbol{\phi(B) = 1 - \ \phi_3 B^3 - \ \phi_6 B^6 - \phi_9 B^9 - \phi_{12} B^{12}}$

$\boldsymbol{\theta(B) = 1 + \ \theta_1 B}$

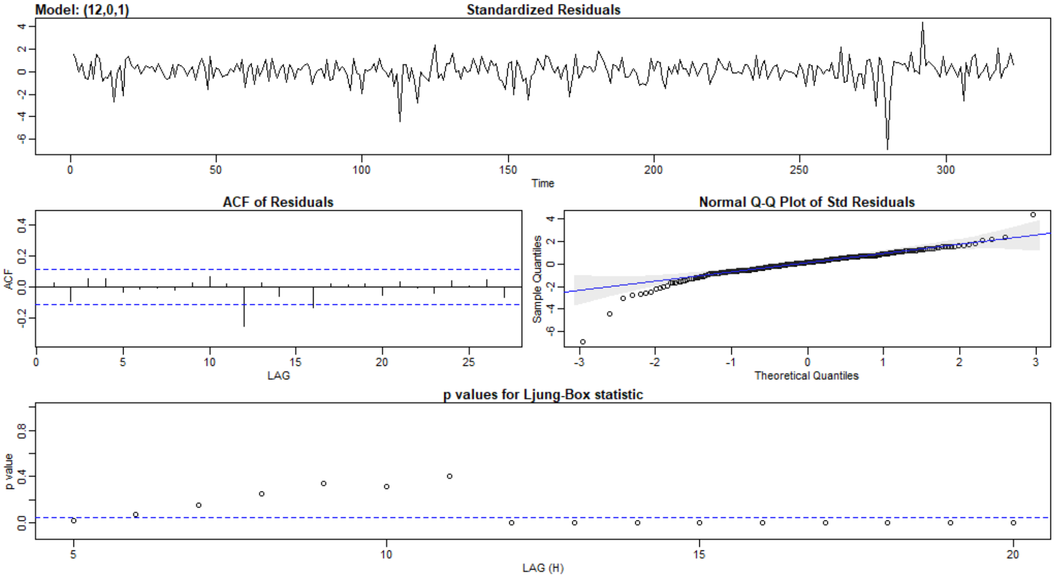
$Note$: X_t is the log of tourist arrivals

Table2: Coefficient Estimates

Coefficients	Estimates	SE	t-value	p-value
$\boldsymbol{\phi_3}$	-0.2906	0.0478	-6.0826	0.00E+00
$\boldsymbol{\phi_6}$	-0.1921	0.0473	-4.0605	1.00E-04
$\boldsymbol{\phi_9}$	-0.2763	0.0479	-5.7704	0.00E+00
$\boldsymbol{\phi_{12}}$	0.5253	0.0477	11.0042	0.00E+00
$\boldsymbol{\theta_1}$	-0.3884	0.0532	-7.305	0.00E+00

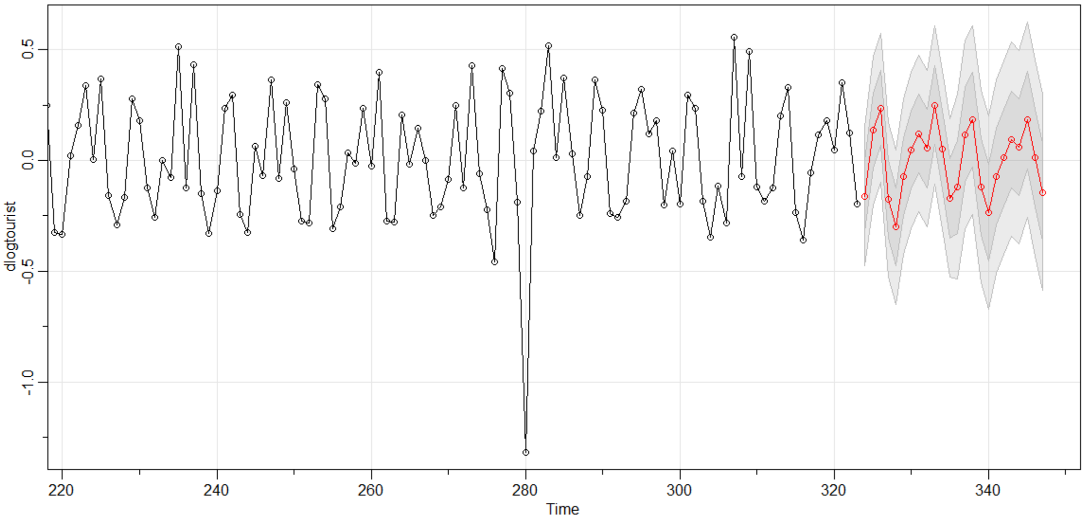
The diagnostic plots for the Restricted ARIMA (12, 1, 1) model are given in Figure10. The ACF at the 12th lag is still significant. Although the p-values of the Ljung-Box statistic appears better than the other models, there are some lags that are still correlated, which calls for a better model.

Figure10: Diagnostics - ARIMA (12, 1, 1)



The final forecast of the Restricted ARIMA (12, 1, 1) is given in Figure11. Please note that the forecast in the figure and the amount provided below are log differenced tourist arrivals, i.e. percentage change in monthly tourist arrivals.

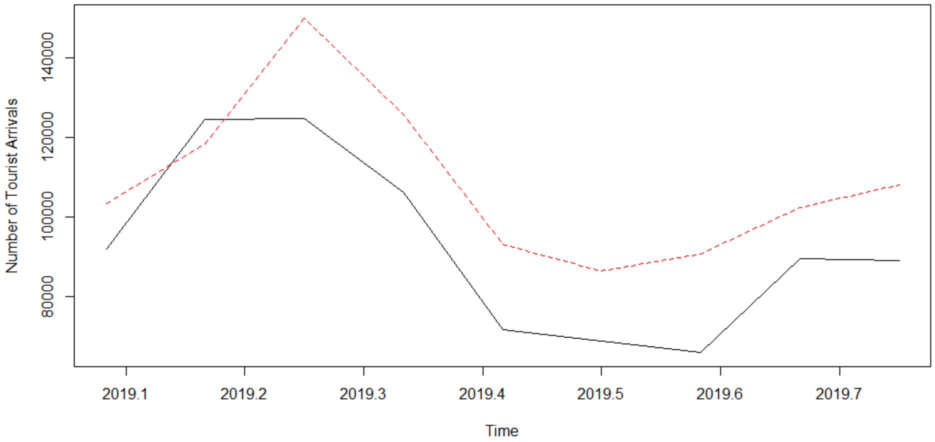
Figure11: Forecasting for 24 periods- Restricted ARIMA (12, 1, 1)



```
[1] -0.16257435  0.13500377  0.23703481 -0.17601514 -0.30152019 -0.07301594
 [7]  0.04660231  0.12192646  0.05420930  0.25051845  0.04980484 -0.16983086
[13] -0.11851536  0.11632563  0.18361935 -0.11901371 -0.23543586 -0.07406509
[19]  0.01261534  0.09635510  0.06164708  0.18352985  0.01124788 -0.14362492
```

Finally, the forecast for 9 periods (January 2019 to September 2019) was compared against the actual tourist arrival data that is available[4]. The forecasted data (in red dotted) against the real 2019 data (black line) is shown in Figure12. The prediction seems a little optimistic, but we notice that it is generally following the pattern of the real data. The SSE for Restricted ARIMA (12,1,1) model is 0.3584, which is slightly better compared to SSE 0.4030 of the ARIMA (12,1,1) model. Both the models perform better against a white noise model that has an SSE of 1.4807. It can be assured that the model presented in the report is performing better than just modelling white noise. However, it should be noted that only 9 new data points were used for testing model performance. Increasing the number of test data would allow for a better comparison.

Figure12: Forecast against Real Tourist Arrivals for First 9 months of 2019



[4] CEIC, (2019). Retrieved from: https://www.ceicdata.com/en/indicator/nepal/visitor-arrivals

Conclusion

Although the forecast looks a bit optimistic, the Restricted ARIMA (12,1,1) model seems to be capturing the peaks and the falls of the original time series well. This is a decent model that could be generated without using seasonal differencing. As mentioned earlier, this model does not incorporate seasonality, nor does it adjust for the outliers present in the dataset. Future considerations include incorporating seasonal differencing or using seasonal multiplicative model to predict for future periods. The model performance will be compared with the performance of the simple ARIMA model presented in this report. Furthermore, considerations will also include identifying and adjusting for outliers.

Appendix

```
library(forecast)
library(TSA)
library(tseries)
library(astsa)
setwd("C:/Users/Anustha Shrestha/Documents/RSTA9701")

#Reading the data
tourism = read.csv("NepalTourism.csv")
touristArrivals = tourism[,3] #timeseries data
tourist = ts(touristArrivals, frequency = 12, start = c(1992,1)) #just for plotting


#plotting the time series data
plot (tourist, ylab = "Number of Tourists", main= "Figure1: Monthly Tourist Arrivals in Nepal (1992 - 2018)")
acf(touristArrivals, main = "Figure 2: ACF and PACF for Tourist Arrivals")
pacf(touristArrivals)


#Box-Cox
BoxCox.lambda(touristArrivals)


#0.035 which is close to 0, so we can use log transformation
# log of tourist arrivals and sample acf and pacf of differenced series
logtourist = log(touristArrivals)
dlogtourist <- diff(logtourist, 1)
par(mfrow=c(1,1))
tsdlogtourist<-ts(dlogtourist, frequency = 12, start=c(1992,2))
par(mfrow=c(3,1))
plot.ts(tsdlogtourist, ylab = "log of Number of Tourists", main="Figure3: Tourist Arrivals - Log and Differenced")
acf(dlogtourist,100, main="Figure 4: ACF and PACF of Log and Differenced Tourist Arrivals")
pacf(dlogtourist,100, main="PACF")


#df tests / unit root tests
adf.test(dlogtourist,k=0) #df test
adf.test(dlogtourist) #adf test
#p-value for both test is 0.01 so we can say that the process is stationary


#EACF
eacf(dlogtourist)


#Fitting models
arima111<-arima(dlogtourist, order=c(1,0,1)) #arima(1,1,1)
arima211<-arima(dlogtourist, order=c(2,0,1)) #arima(2,1,1)
arima210<-arima(dlogtourist, order=c(2,0,0)) #arima(2,1,0)
arima110<-arima(dlogtourist, order=c(1,0,0)) #arima(1,1,0)
arima1210<-arima(dlogtourist, order=c(12,0,0)) #arima(12,1,0)
```

```
arima1211<-arima(dlogtourist, order=c(12,0,1)) #arima(12,1,1)
arima1212<-arima(dlogtourist, order=c(12,0,2)) #arima(12,1,2)
arima610<-arima(dlogtourist, order=c(6,0,0)) #arima(6,1,0)
arima612<-arima(dlogtourist, order=c(6,0,2)) #arima(6,1,2)

#Model Selection
AIC(arima111); BIC(arima111);
AIC(arima211); BIC(arima211);
AIC(arima210); BIC(arima210);
AIC(arima110); BIC(arima110);
AIC(arima1210); BIC(arima1210);
AIC(arima1211); BIC(arima1211);
AIC(arima1212); BIC(arima1212);
AIC(arima610); BIC(arima610);
AIC(arima612); BIC(arima612);


#Model diagnostics
sarima(logtourist, 1,1,1)
sarima(logtourist, 2,1,1)
sarima(logtourist,2,1,0)
sarima(logtourist,1,1,0)
sarima(logtourist,12,1,0)
sarima(logtourist,12,1,1)
sarima(logtourist,12,1,2)
sarima(logtourist,6,1,0)
sarima(logtourist,6,1,2)


#Forecasting
logtourist = ts(logtourist, frequency =1)


#arima (1,1,1)
sarima(logtourist, 1,1,1)
sarima.for(logtourist, 24,1,1,1, newxreg = (nobs+1): (nobs+24))
sarima.for(dlogtourist, 24,1,0,1, newxreg = (nobs+1): (nobs+24))


#arima (2,1,1)
sarima(logtourist, 2,1,1)
sarima.for(logtourist, 24,2,1,1, newxreg = (nobs+1): (nobs+24))
sarima.for(dlogtourist, 24,2,0,1, newxreg = (nobs+1): (nobs+24))


#arima (12,1,1)
sarima(logtourist, 12,1,1)
sarima.for(logtourist, 24,12,1,1, newxreg = (nobs+1): (nobs+24))
sarima.for(dlogtourist, 24,12,0,1, newxreg = (nobs+1): (nobs+24))
```

```
#Restricted Arima(12,1,1)
nobs = length(logtourist)
RArima1211<- arima(logtourist,c(12,1,1),fixed=c(0,0,NA,0,0,NA,0,0,NA,0,0, NA, NA,0), xreg=1:nobs)
sarima(logtourist,12,1,1,fixed=c(0,0,NA,0,0,NA,0,0,NA,0,0, NA, NA, 0))
sarima.for(dlogtourist, 24,12,0,1, fixed=c(0,0,NA,0,0,NA,0,0,NA,0,0, NA, NA, 0), newxreg = (nobs+1): (nobs+24))

#Model comparison using 9 period prediction against real data
#model for white noise
whitenoise <-arima(logtourist, order=c(0,1,0), xreg=1:nobs)

#actual data
true <- c(91793, 124421, 124697, 106136, 71640, 68782, 65749, 89382, 89078)
logtrue = log(true)

#forecasting for 9 periods
testforeR1211 = predict(RArima1211, 9, newxreg = (nobs+1): (nobs +9))
testfore1211 = predict (arima1211, 9, newxreg = (nobs+1): (nobs+9))
testforeWN = predict (whitenoise, 9, newxreg = (nobs+1): (nobs+9))

#SSE for 3 models
RArimaSSE = sum((testforeR1211$pred - logtrue)^2)
ArimaSSE = sum((testfore1211$pred - logtrue)^2)
WNSSE = sum((testforeWN$pred - logtrue)^2)
RArimaSSE
ArimaSSE
WNSSE

#Plotting out the predicted vs. 9 period
truets <- ts(true, start = c(2019,2), frequency=12)
Rarima1211P <-exp(testforeR1211$pred)
ts.plot(cbind(truets, ts(Rarima1211P, start=c(2019,2), frequency=12)), gpars = list(lty =c(1:3),col=1:2, ylab="Number of Tourist Arrivals", main="Forecast against Real Tourist Arrivals for First 9 months of 2019"))
```