2017

Forecasting US Tourism

BUSINESS FORECASTING REPORT

ASHISH UPPIN

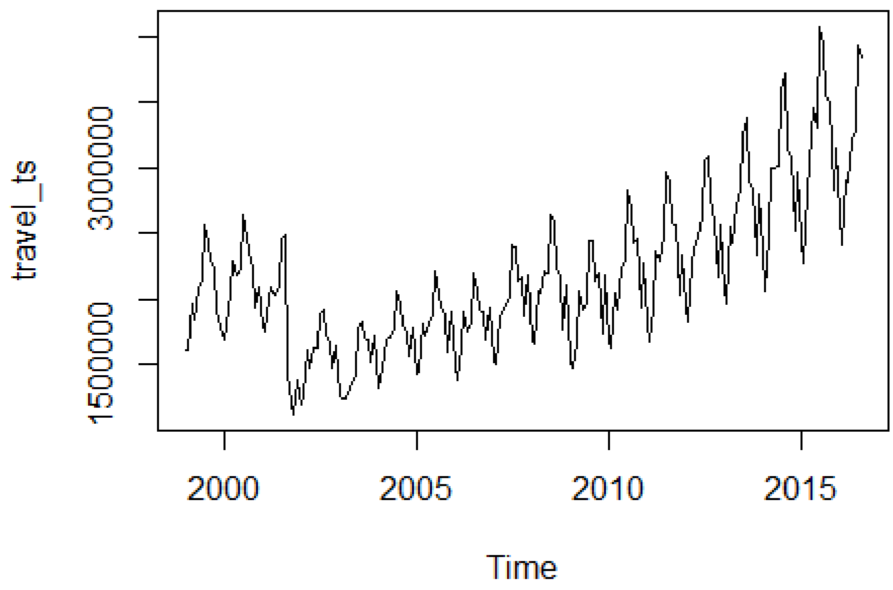
Contents

Forecasting US Tourism

Introduction

The given data is from US Department of Tourism which maintains statistics on visitors to America. We are going to focus on just a general statistic that keeps track of monthly visitors.   See details at http://travel.trade.gov/research/monthly/arrivals/index.asp

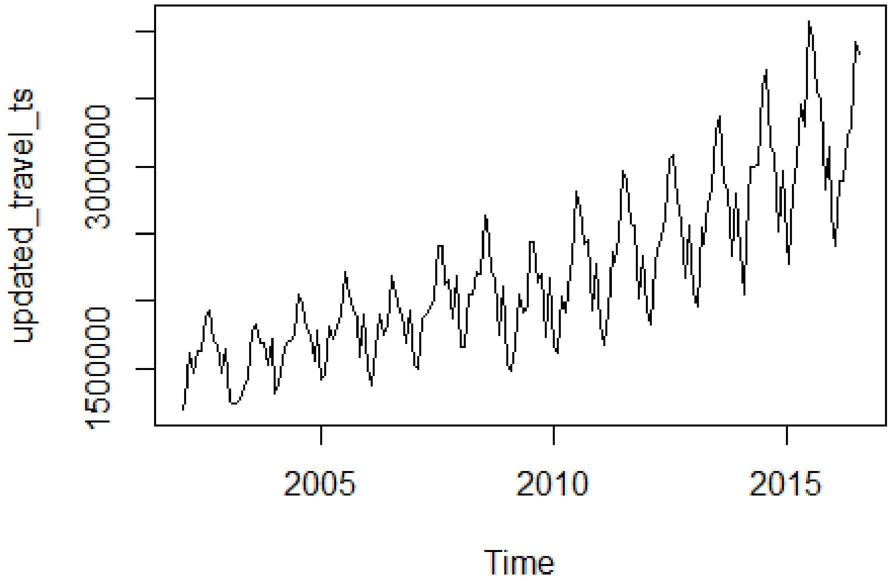
```
library(readr)
library(forecast)

Final_Travel <- read_csv("E:/Masters/Sem2/Business Forecasting/EndTerm/Fin
al_Travel.csv")

travel <- Final_Travel$Value
travel_ts <- ts(travel,start=c(1999,1),frequency = 12)
plot(travel_ts)
```



Through the visualization of the dataset in form of a timeseries plot, we can see a big drop in the number of tourists in the year 2001. This is because of an event which occurred in US, The World Trade center terrorist attack. This being a disaster, tourist tend to avoid visiting that location, hence a significant drop in tourism for the USA. But after 2002, it seems to gradually pick up the numbers.

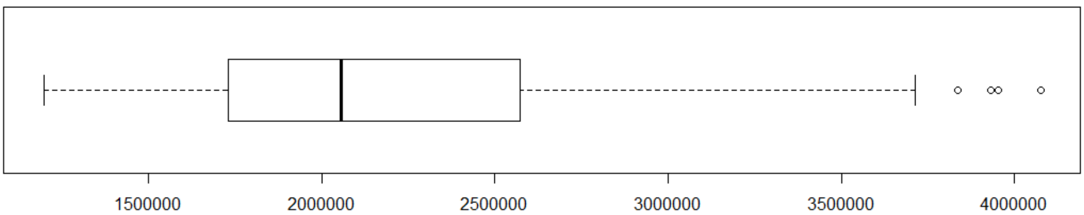
Since the drop was due to an external factor of security, I will be removing the data prior to 2002, and will use the data from 2002 onwards for my forecast analysis.

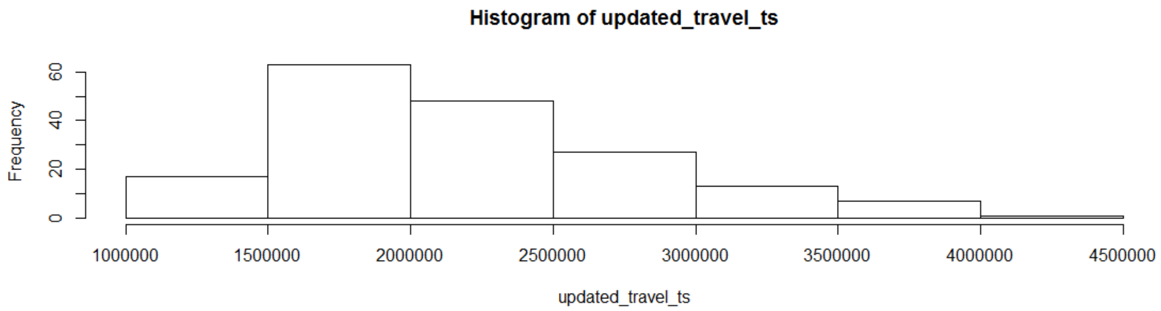
```
updated_travel_ts= window(travel_ts, start=2002)
plot(updated_travel_ts)
```



In the plot shown above, we can also observe a drop in the tourism in the year 2009. This was due to the economic crisis in the world, hence people sending less income to travel. But I will not be removing this from the forecast as it's not that significant of a drop.

```
summary(updated_travel_ts)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 1197000 1735000 2056000 2206000 2569000 4075000

par(mfrow=c(2,1))
boxplot(updated_travel_ts,horizontal=TRUE)
hist(updated_travel_ts)
```



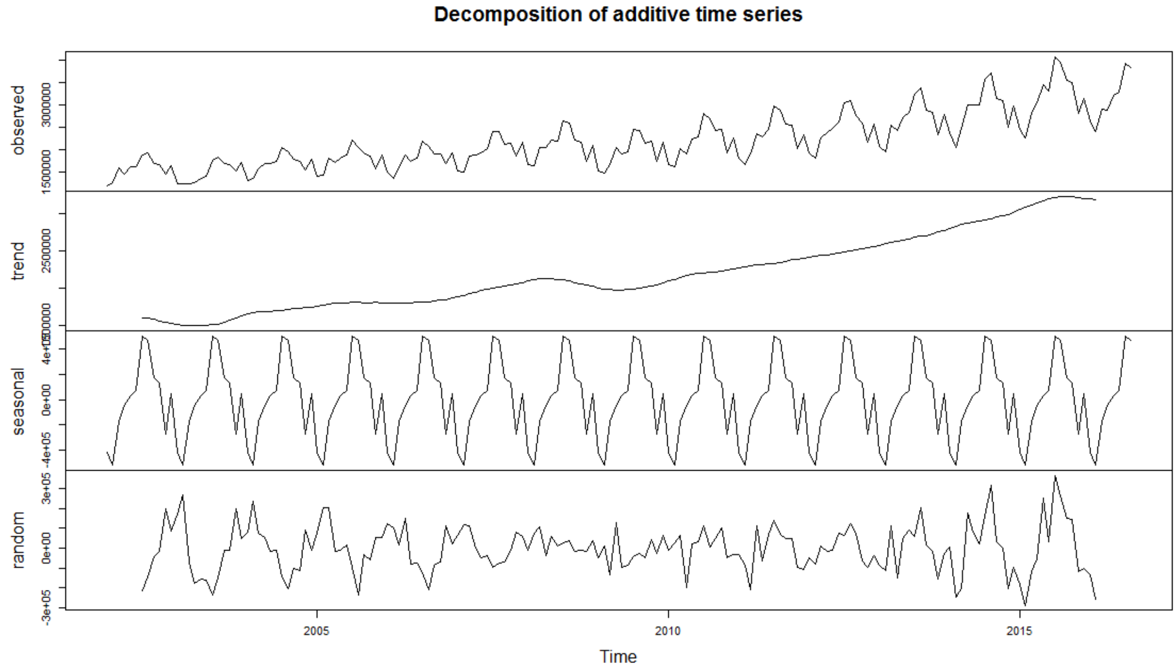


From the boxplot and the histogram, it can be observed that the normal distribution is leaning towards the right hand side. In the box plot, we can see outliers towards the maximum of the dataset. We can analyze these outlier numbers as we would like to achieve these numbers, as this means more tourism business for the country.

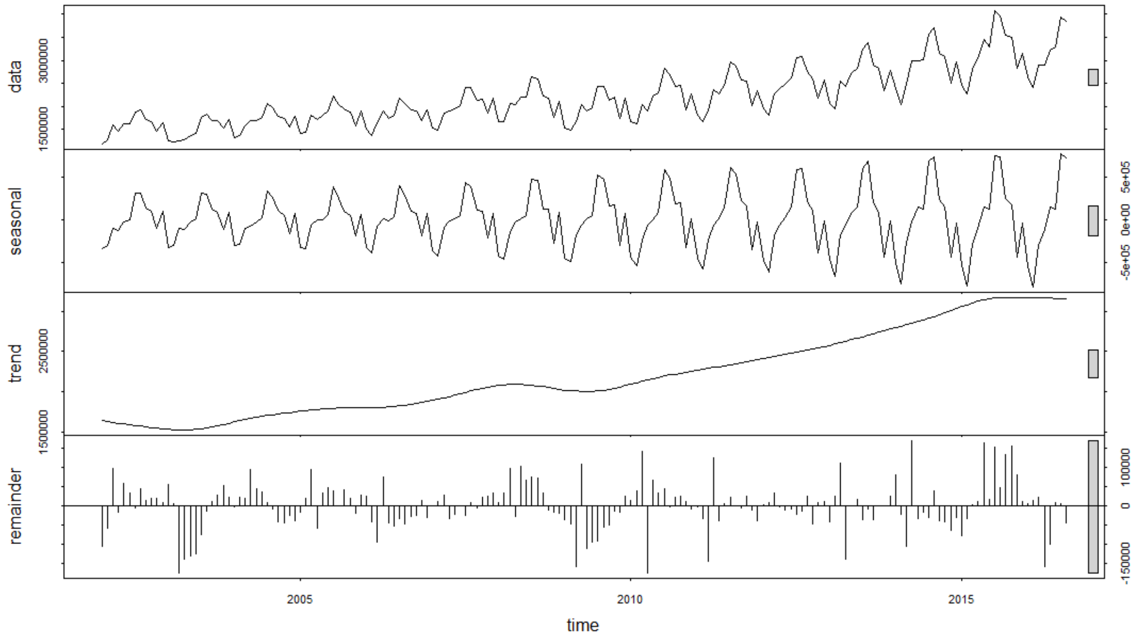
Decomposition

Looking at the tie series plot, we can observe that it's an additive series. But with help of decompose function, we can confirm that it's an additive series.

```
decompose_travel <- decompose(updated_travel_ts)
print(decompose_travel$type)

"additive"

plot(decompose_travel)
```




```
stl_decomp <- stl(updated_travel_ts,s.window=5)
plot(stl_decomp)
```



With the stl function plot graph and the decompose plot graph, we can notice that the trend component is the majority of the data. But we do see seasonality component present. The seasonal plot from stl function shows a multiplicative increase.

Looking at the trend and seasonality components I think Naïve model should not be chosen for forecasting method. I think Holt-Winter or ARIMA models will work best for this dataset. The best option to use dimensionless error measure, thus I choose MAPE for comparison towards the end of the report.

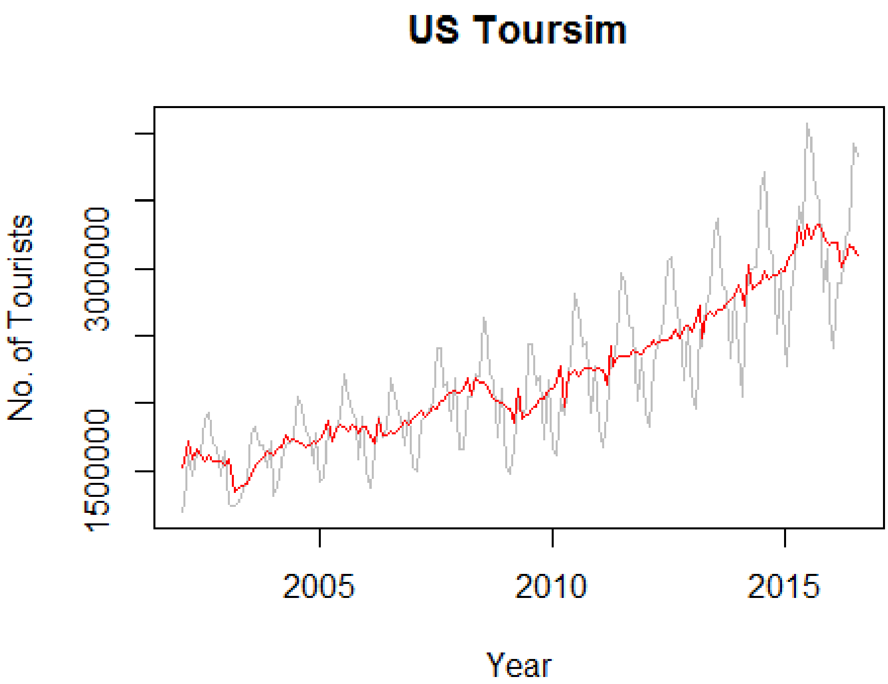
The values below show the seasonal adjustment monthly indices.

```
seaAdjusted <-seasadj(stl_decomp)
print(seaAdjusted)
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug
## 2002 1529638 1567995 1713904 1589847 1655788 1622363 1573844 1615986
## 2003 1585959 1532958 1345597 1380746 1392158 1403417 1458516 1534426
## 2004 1618482 1658569 1673682 1765030 1724490 1729452 1713752 1705959
## 2005 1738321 1780929 1864258 1716838 1814268 1833520 1830469 1795975
## 2006 1826571 1761778 1711254 1881155 1767806 1763826 1788961 1782109
## 2007 1887799 1917062 1948366 1905204 1933884 1973990 1967473 2017131
## 2008 2086026 2115418 2184828 2064435 2192381 2152517 2158107 2143668
## 2009 1982532 1964799 1847984 2108262 1891364 1909996 1916196 1963938
## 2010 2106923 2148708 2270940 1972101 2230004 2212341 2238509 2205071
## 2011 2264672 2244143 2145509 2424469 2271737 2327898 2357550 2347418
## 2012 2414083 2430625 2468042 2444723 2451577 2467083 2467869 2488944
## 2013 2536070 2617688 2721490 2487808 2648028 2683332 2648728 2695134
## 2014 2878936 2795381 2731964 3025382 2842198 2879008 2885315 2975308
## 2015 2981522 3046208 3105697 3137217 3306207 3174441 3324167 3220423
## 2016 3178977 3186219 3191436 3010006 3064742 3170170 3163369 3106814
##          Sep     Oct     Nov     Dec
## 2002 1577500 1574879 1565887 1546506
## 2003 1570098 1599939 1640022 1627348
## 2004 1680479 1687686 1713930 1708820
## 2005 1839260 1820795 1781280 1829968
## 2006 1810711 1823217 1876375 1844846
## 2007 2018727 2063687 2079868 2098542
```

```
## 2008 2095896 2042191 2024592 2011443
## 2009 1979759 2028762 2040996 2099981
## 2010 2242006 2256926 2254786 2248452
## 2011 2355114 2396170 2373903 2358488
## 2012 2542764 2483619 2554568 2572480
## 2013 2687767 2742323 2762503 2804462
## 2014 2922058 2942861 2944224 3001890
## 2015 3306856 3330332 3253506 3183978
## 2016
```

```
plot(updated_travel_ts, col="grey",
     main="US Toursim", xlab="Year", ylab="No. of Tourists")
lines(seaAdjusted,col="red",ylab="Seasonally adjusted")
```

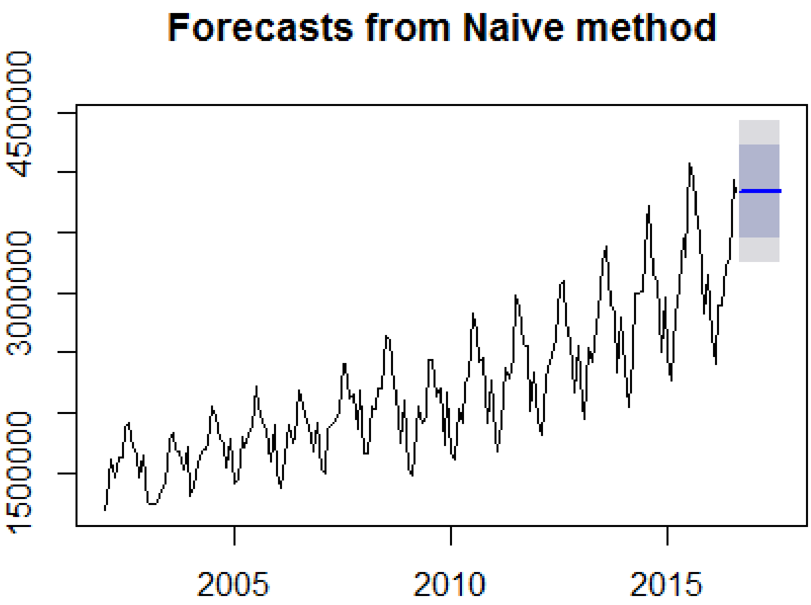


The values from the seasonal adjusted time series for being high is in the month of October 2015 and lowest in the month of March 2003. But with the time series we can see a decreased toursm rate during the winter (Oct-Jan) of each year, and the highest during the months (April-July) as it's the summer and a pleasant weather to tour the country.

Naïve Method

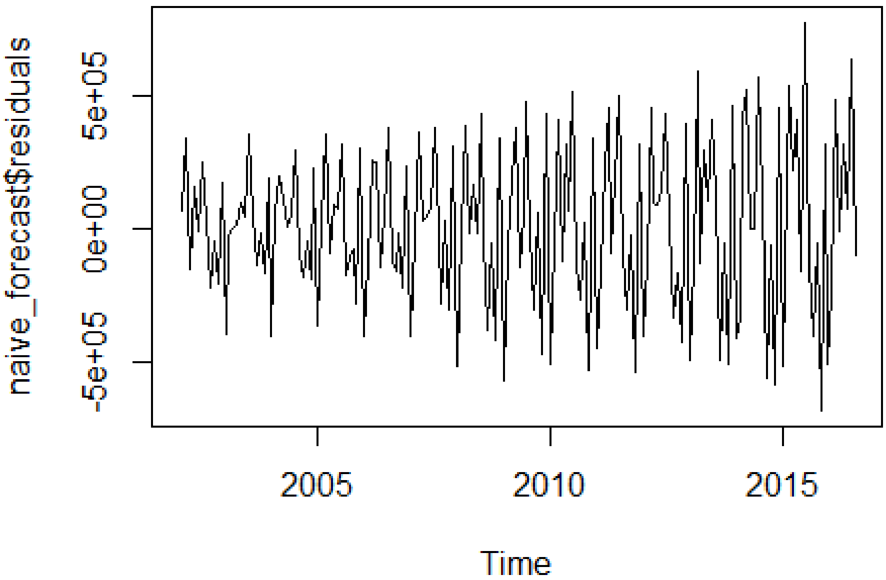
Though this is considered as a benchmark for most of the forecast, I believe this model will perform poorly, as the dataset has trend and seasonality.

```
naive_forecast <- naive(updated_travel_ts,12)
plot(naive_forecast)
```



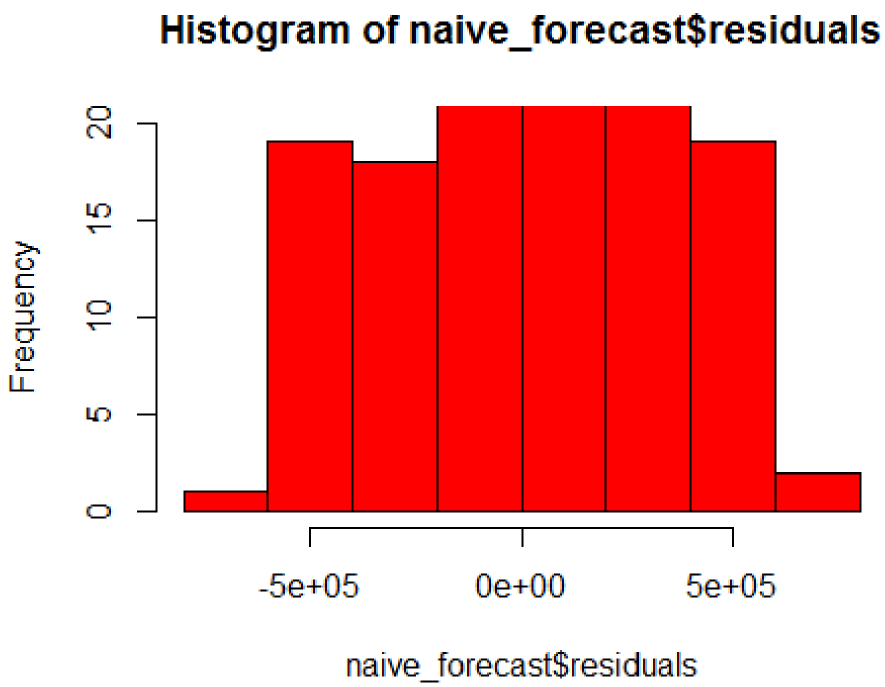
To check whether the forecast errors have constant variance we plot a residual graph.

```
plot(naive_forecast$residuals)
```



The plot shows that the forecast errors seem to have roughly constant variance over time, although the size of the fluctuations in the start of the time series (2002-2005) may be slightly less than that at later dates (eg. 2010-2016).

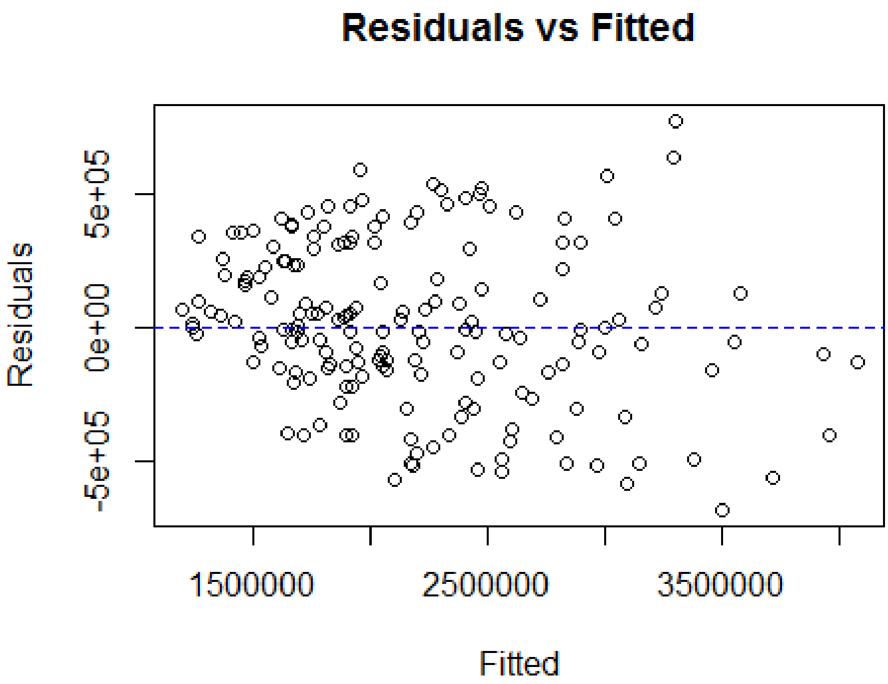
```
hist(naive_forecast$residuals, breaks=10, col="red" ,ylim=c(0,20))
```



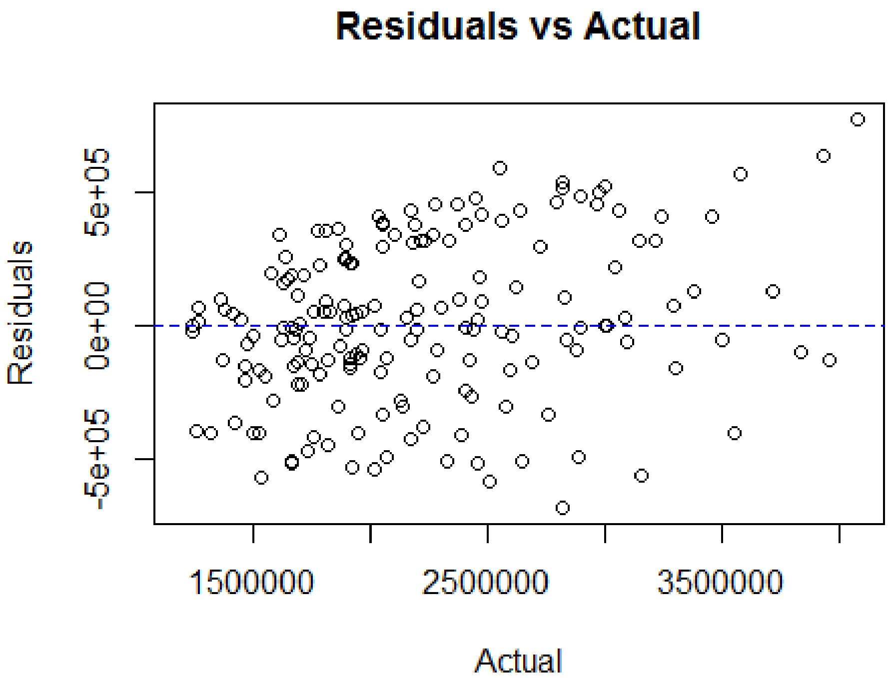

```
shapiro.test(naive_forecast$residuals)
##  Shapiro-Wilk normality test
## data:  naive_forecast$residuals
## W = 0.98491, p-value = 0.05595
```

The plot shows that the distribution of forecast errors is roughly centred on zero, and is more or less normally distributed. This can be confirmed with the Shapiro test where the p-value is greater than the significance level (0.05), thus the null-hypothesis that it is normally distributed cannot be rejected.

```
plot(as.matrix(naive_forecast$fitted), as.matrix(naive_forecast$residuals)
,main="Residuals vs Fitted", xlab = "Fitted", ylab = "Residuals") # plot o
f fitted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```

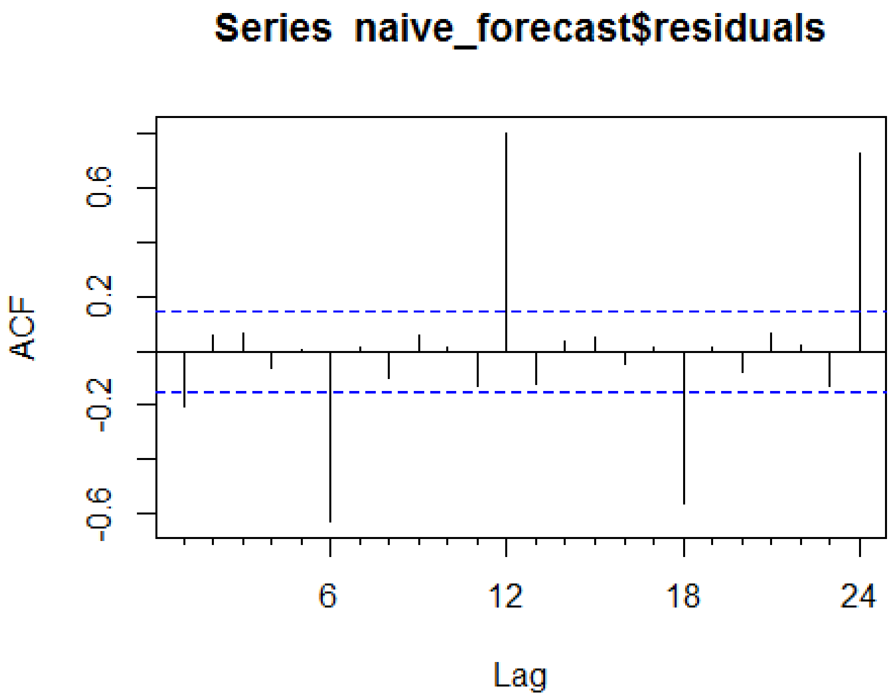



```
plot(as.matrix(updated_travel_ts), as.matrix(naive_forecast$residuals),main="Residuals vs Actual", xlab = "Actual", ylab = "Residuals") # plot of fitted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```



The residual plots they're pretty symmetrically distributed, tending to cluster towards the middle of the plot. Though we can observe one or two outliers. They're clustered around 0. In general there aren't clear patterns. But this model needs improvement as we see many outliers in the plots.

```
Acf(naive_forecast$residuals)
```



With the Acf plot, we see if the forecast residuals show non-zero autocorrelations. With the plot, at the lags 6,12,18,24 it exceeds the significance bounds. We can verify this with the he Ljung-Box test.

```
Box.test(naive_forecast$residuals, lag=10, type="Ljung-Box")
##  Box-Ljung test
## data:  naive_forecast$residuals
## X-squared = 84.001, df = 10, p-value = 8.216e-14
```

In the test it showed that there is evidence of non-zero autocorrelations in the forecast residual. This suggests that this model is not a good model for prediction.

```
accuracy(naive_forecast)

##                    ME   RMSE      MAE        MPE     MAPE     MASE
## Training set 15081.85 301462 240545.6 -0.2643395 11.06543 1.397038
##                    ACF1
## Training set -0.2024705

print(naive_forecast)

##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Sep 2016        3836721 3450382 4223060 3245866 4427576
## Oct 2016        3836721 3450382 4223060 3245866 4427576
## Nov 2016        3836721 3450382 4223060 3245866 4427576
## Dec 2016        3836721 3450382 4223060 3245866 4427576
## Jan 2017        3836721 3450382 4223060 3245866 4427576
## Feb 2017        3836721 3450382 4223060 3245866 4427576
## Mar 2017        3836721 3450382 4223060 3245866 4427576
## Apr 2017        3836721 3450382 4223060 3245866 4427576
## May 2017        3836721 3450382 4223060 3245866 4427576
```

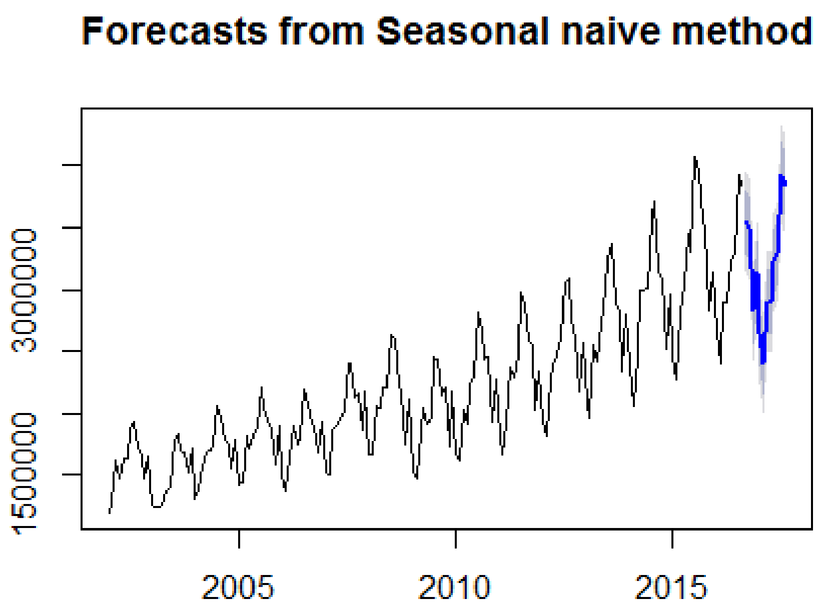
```
## Jun 2017         3836721 3450382 4223060 3245866 4427576
## Jul 2017         3836721 3450382 4223060 3245866 4427576
## Aug 2017         3836721 3450382 4223060 3245866 4427576
```

It predicts that the value after a year will be Aug 2017: 3836721.

S-Naïve Method

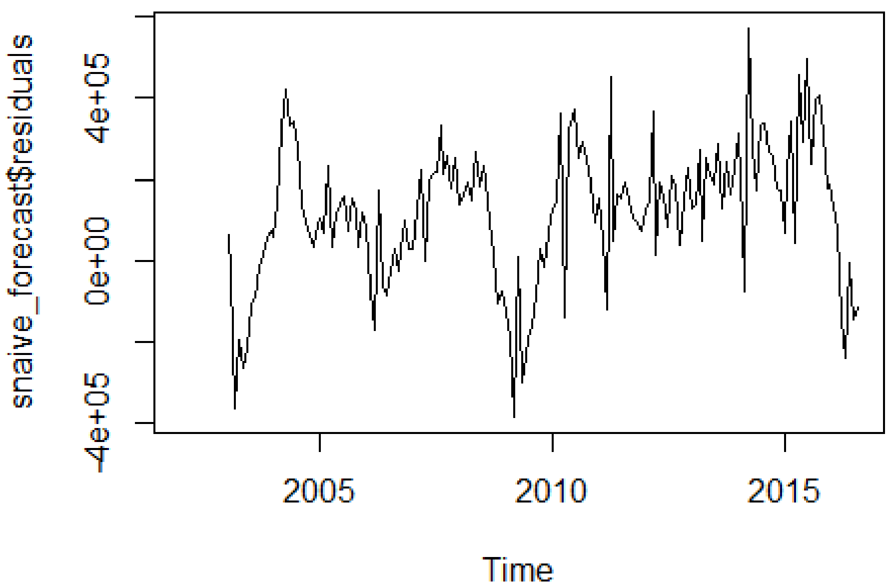
As this method considers the seasonality factor, I am hoping that this would be a better model than the normal Naïve model.

```
snaive_forecast <- snaive(updated_travel_ts,12)
plot(snaive_forecast)
```



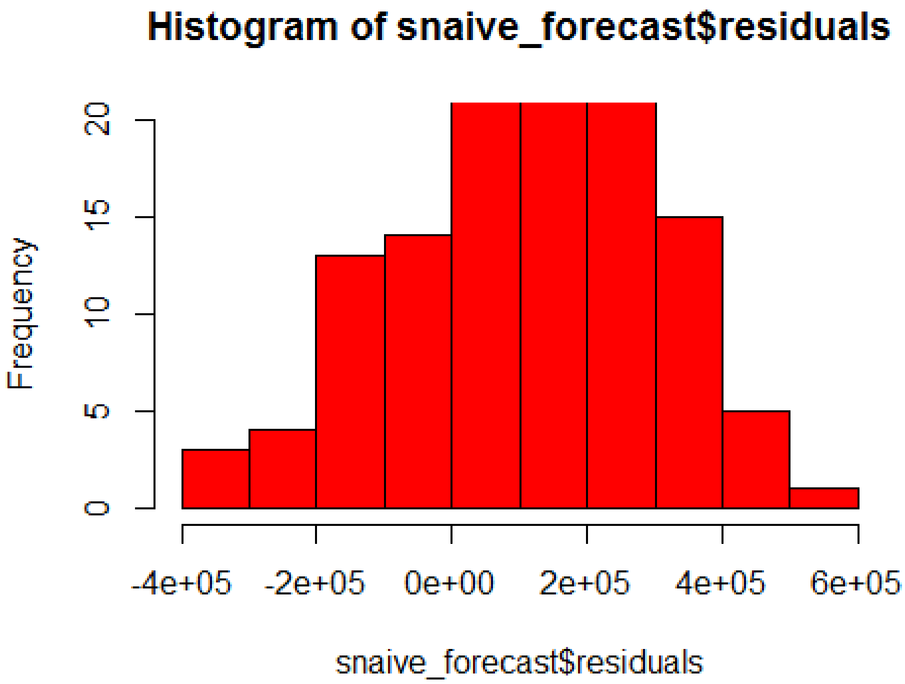
To check whether the forecast errors have constant variance we plot a residual graph.

```
plot(snaive_forecast$residuals)
```



The plot shows that the forecast errors seem to have roughly constant variance over time, although the size of the fluctuations from (2002-2004) and (2008-2010) is more than the other dates (2005-2008) and (2010-2015).

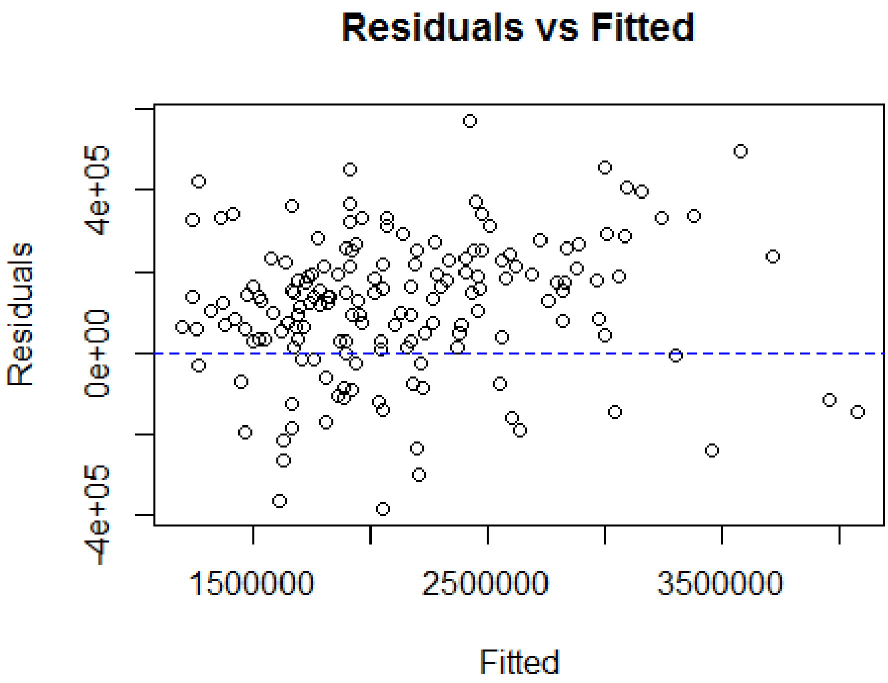
```
hist(snaive_forecast$residuals, breaks=10, col="red" ,ylim=c(0,20))
```



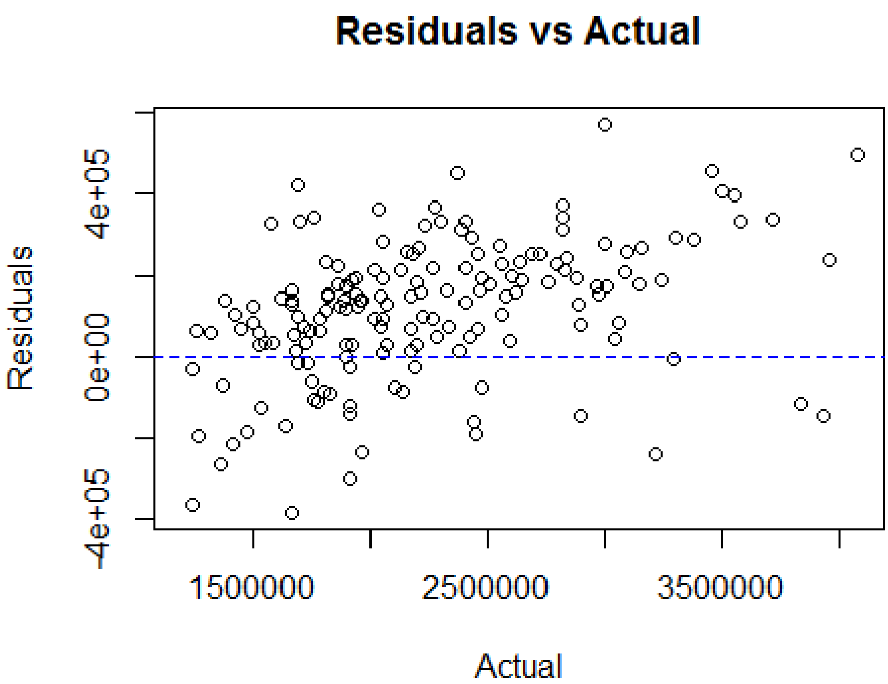

```
shapiro.test(snaive_forecast$residuals)

##  Shapiro-Wilk normality test
## data:  snaive_forecast$residuals
## W = 0.98735, p-value = 0.1455
```

The plot shows that the distribution of forecast errors is skewed towards the right hand side of 0, and shows a right skew distribution. This shows that it has many positive residual terms, and say that it is not normally distributed. We can confirm that it’s not normally distributed with the Shapiro test, as the p-value is greater than (0.05).

```
plot(as.matrix(snaive_forecast$fitted), as.matrix(snaive_forecast$residual
s),main="Residuals vs Fitted", xlab = "Fitted", ylab = "Residuals") # plot
of fitted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```

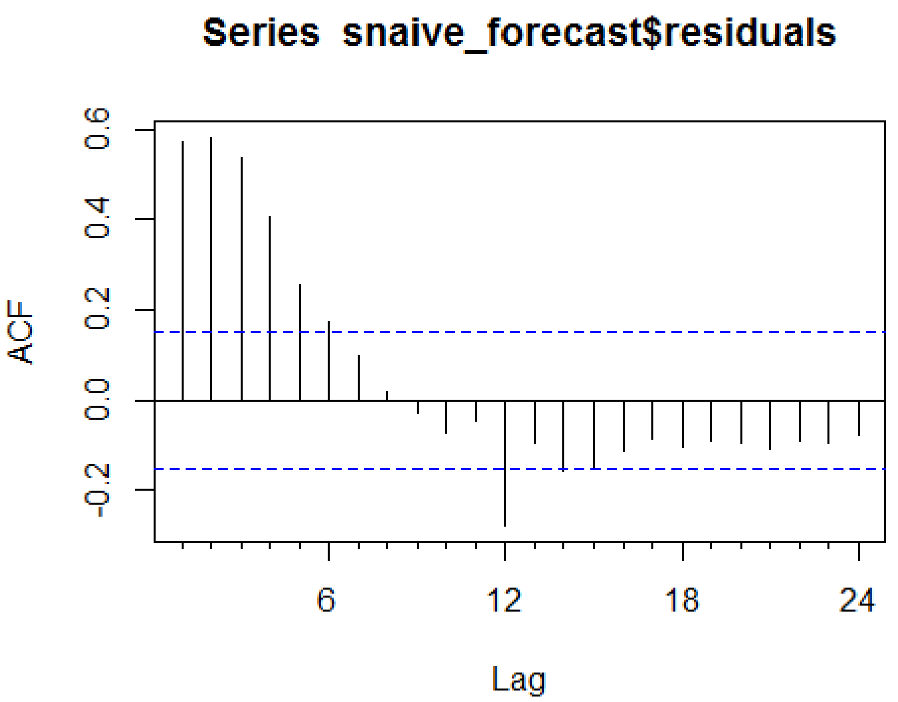



```
plot(as.matrix(updated_travel_ts), as.matrix(snaive_forecast$residuals),main="Residuals vs Actual", xlab = "Actual", ylab = "Residuals") # plot of fitted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```



The residual plots show that many points are above 0 and not symmetrically distributed. We can see a cluster form which shows some patterns. This shows it’s not a good model to form a prediction.

```r
Acf(snaive_forecast$residuals)
```



With the Acf plot, we see if the forecast residuals shows non-zero autocorrelations. With the plot, at the lags 0-6,12 it exceeds the significance bounds. We can verify this with the he Ljung-Box test.

```r
Box.test(snaive_forecast$residuals, lag=10, type="Ljung-Box")
##  Box-Ljung test
## data:  snaive_forecast$residuals
## X-squared = 207.21, df = 10, p-value < 2.2e-16
```

In the test it showed that there is evidence of non-zero autocorrelations in the forecast residual, as the p-value is less than the significant level (0.05). This suggests that this model is not a good model for prediction.

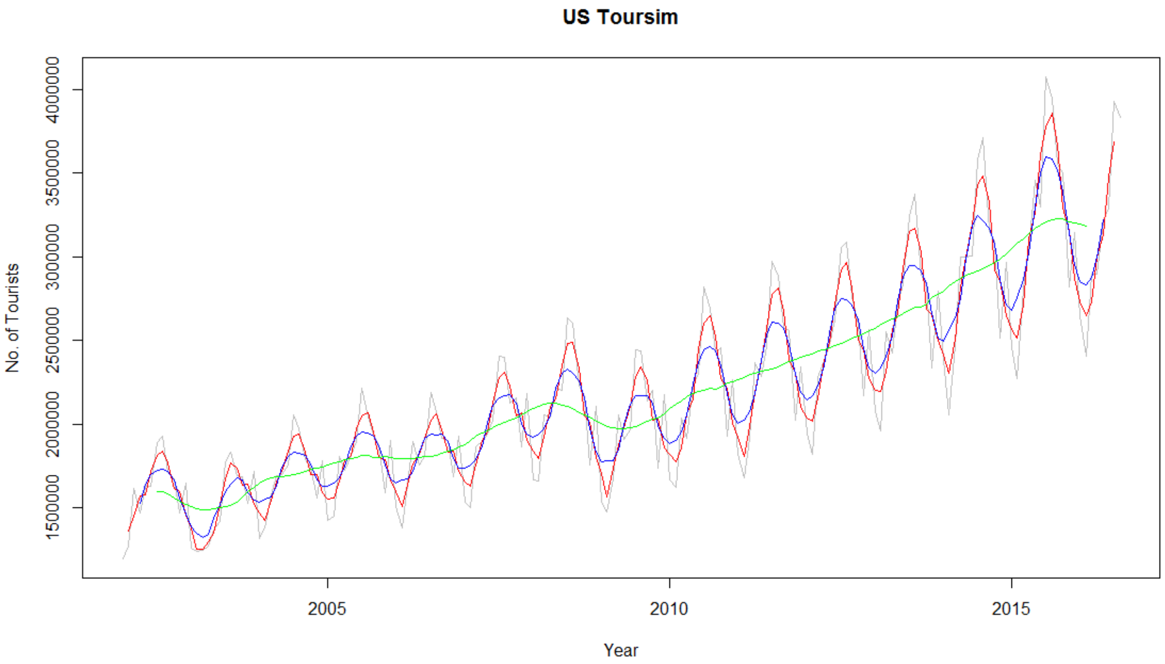
```r
accuracy(snaive_forecast)

##                    ME     RMSE      MAE      MPE     MAPE MASE    ACF1
## Training set 115998.2 205853.8 172182.5 4.595344 7.741379    1 0.5732204

print(snaive_forecast)

##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Sep 2016        3551833 3288021 3815645 3148367 3955299
## Oct 2016        3500510 3236698 3764322 3097044 3903976
## Nov 2016        2822990 2559178 3086802 2419524 3226456
## Dec 2016        3145903 2882091 3409715 2742437 3549369
## Jan 2017        2645349 2381537 2909161 2241883 3048815
## Feb 2017        2405849 2142037 2669661 2002383 2809315
## Mar 2017        2897042 2633230 3160854 2493576 3300508
## Apr 2017        2895487 2631675 3159299 2492021 3298953
## May 2017        3213869 2950057 3477681 2810403 3617335
## Jun 2017        3293795 3029983 3557607 2890329 3697261
## Jul 2017        3931057 3667245 4194869 3527591 4334523
## Aug 2017        3836721 3572909 4100533 3433255 4240187
```

It predicts that the value after a year will be Aug 2017: 3836721.

Both the Naïve and S-Naïve gave the same value of 3836721 for August 2017.

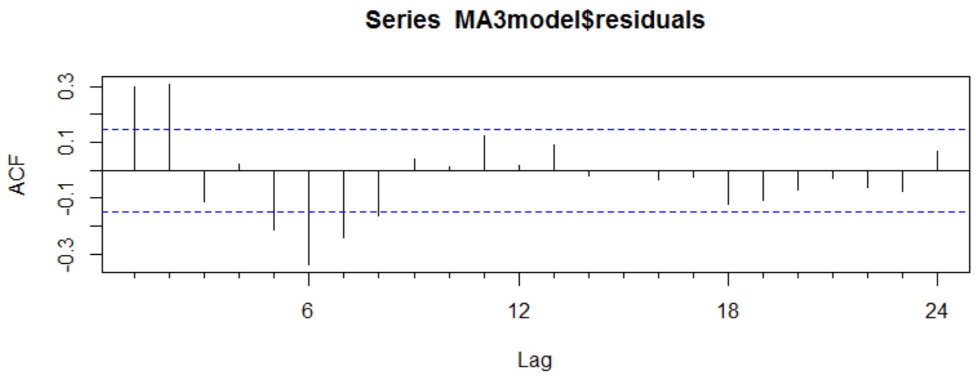
Simple Moving Averages

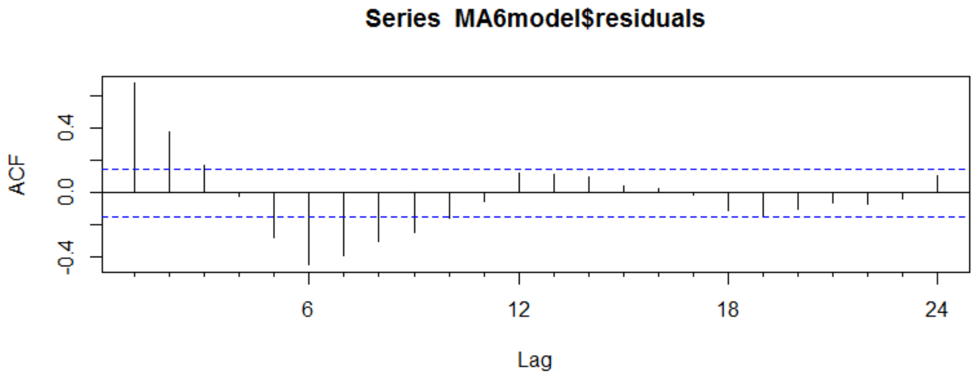
```
plot(updated_travel_ts, col="grey",
     main="US Toursim", xlab="Year", ylab="No. of Tourists")
lines(ma(updated_travel_ts,3),col="red")
lines(ma(updated_travel_ts,6),col="blue")
lines(ma(updated_travel_ts,12),col="green")
```



We can decide the best Moving Average order by plotting the Acf of the forecasted residual values

```
MA3model = forecast(ma(updated_travel_ts,3),h=12)
Acf(MA3model$residuals)
MA6model = forecast(ma(updated_travel_ts,6),h=12)
Acf(MA6model$residuals)
```

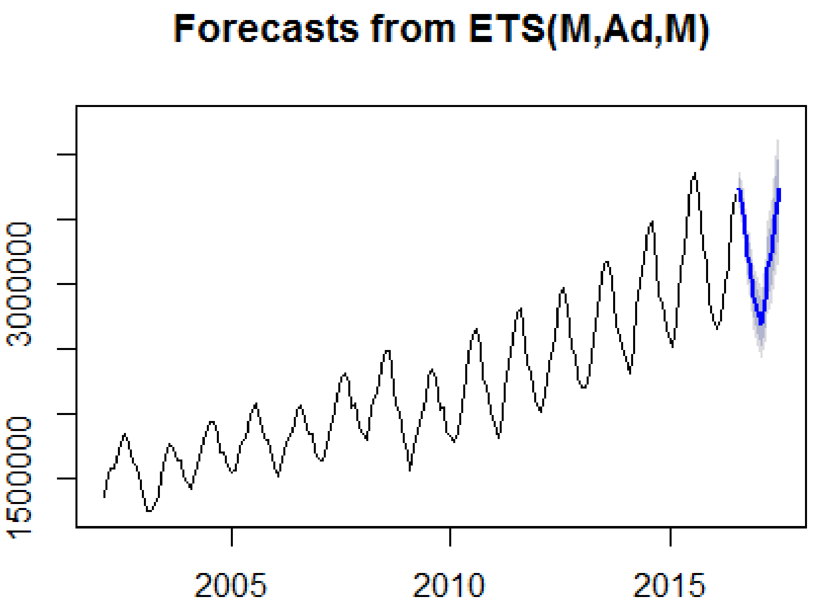




The Acf plot shows that MA3 Model has fewer lags(1,2,5,6) out of 24 lags outside of the confidence bounds, which indicates it shows no non-zero autocorrelations, and will be a better mode for this dataset than MA6 Model.

Plotting the forecast for MA 3 Model:

```
plot(forecast(ma(updated_travel_ts,3),h=12))
```




```
accuracy(forecast(ma(updated_travel_ts,3),h=12))

##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 6181.567 39827.99 30795.72 0.2420915 1.466381 0.1882029
```

```
##                      ACF1
## Training set 0.2192467
```

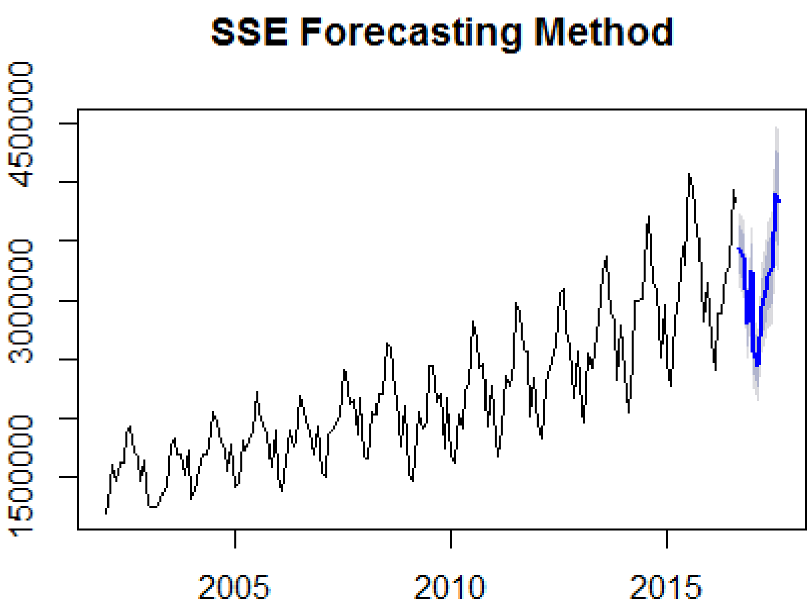
As the order goes up, the fitted values have more smoothing factor. In this dataset, it makes the model less accurate with the forecasted residual losing the normal distribution shape and having more non-zero autocorrelation.

```
print(MA3model)

##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Aug 2016        3727870 3635409 3820332 3586463 3869278
## Sep 2016        3547538 3422580 3672495 3356432 3738644
## Oct 2016        3208127 3069151 3347104 2995581 3420673
## Nov 2016        3165970 3006950 3324991 2922770 3409171
## Dec 2016        2906392 2742516 3070268 2655765 3157019
## Jan 2017        2792232 2619076 2965389 2527412 3057052
## Feb 2017        2689605 2508739 2870472 2412994 2966217
## Mar 2017        2865313 2658526 3072099 2549060 3181566
## Apr 2017        3120941 2881128 3360754 2754179 3487703
## May 2017        3238094 2974831 3501357 2835468 3640720
## Jun 2017        3522509 3221028 3823990 3061433 3983585
## Jul 2017        3734491 3399432 4069550 3222062 4246920
```

The MA3 Model predicted the value to be Jul 2017: 3734491

Simple Smoothing

```
sse_forecast <- forecast(updated_travel_ts,h=12)
plot(sse_forecast,main="SSE Forecasting Method")
```

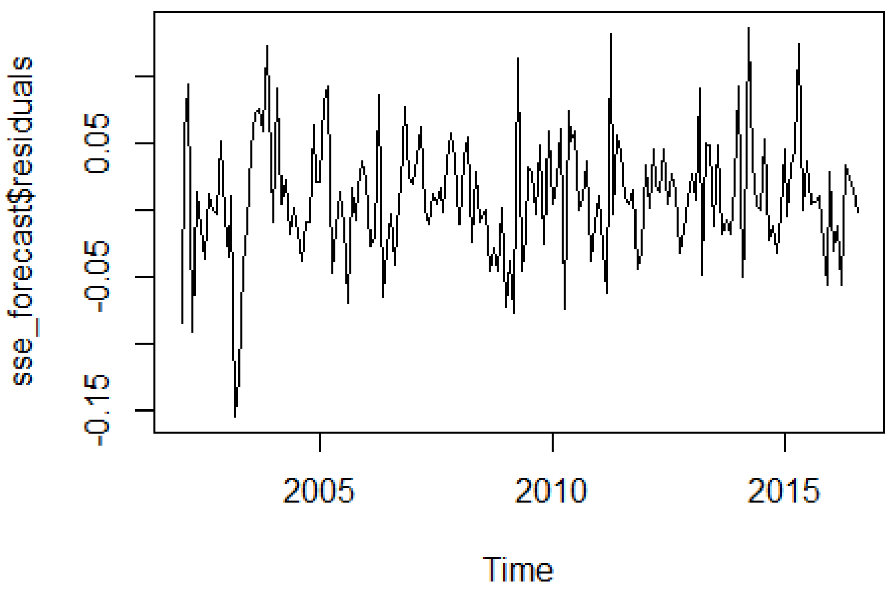


```r
summary(sse_forecast)
```

```
##   Smoothing parameters:
##     alpha = 0.4147
##     gamma = 1e-04
##   Initial states:
##     l = 1615326.4368
##     s=1.0256 0.8825 1.0596 1.0823 1.2087 1.2262
##            1.0286 1.0079 0.972 0.9245 0.7721 0.81
##   sigma:  0.0469
```

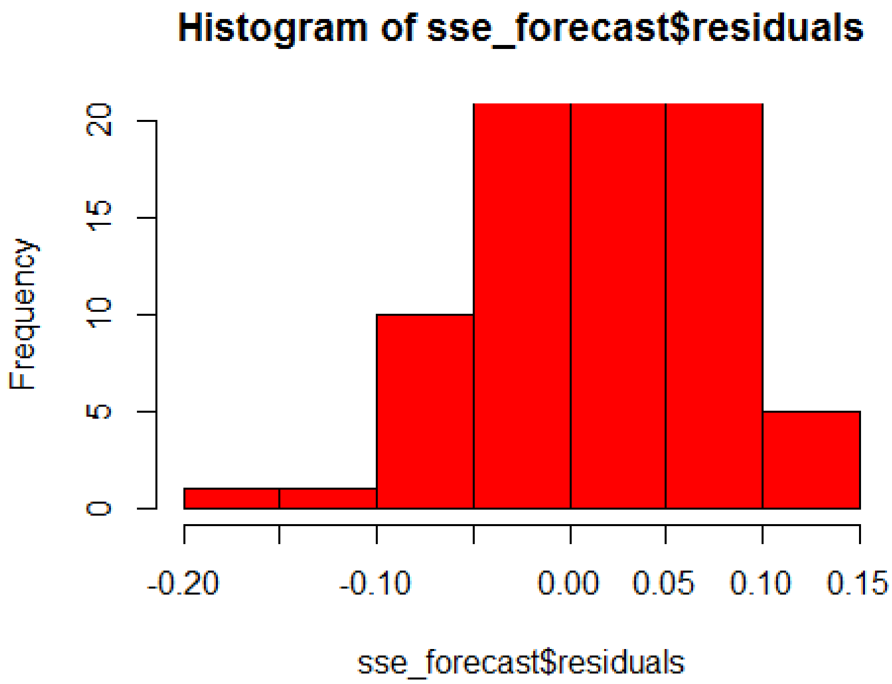
Alpha is the smoothing parameter. If alpha is small (i.e., close to 0), more weight is given to observations from the more distant past, if alpha is large (i.e., close to 1), more weight is given to the more recent observations. In this case its 0.4147 which is closer to 0 than to 1. Hence it shows more weightage towards distant past. Sigma is the Standard deviation of residuals. The value of sigma is 0.0469.

```r
plot(sse_forecast$residuals)
```



The plot shows that the forecast errors seem to have roughly constant variance over time, with fluctuations from (2004-2016) close to 0.

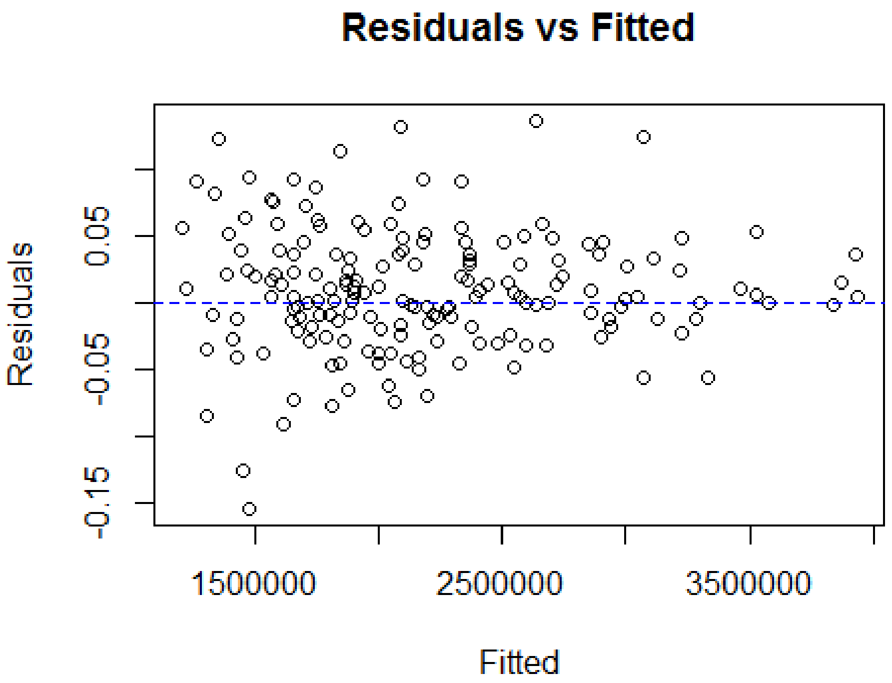
```r
hist(sse_forecast$residuals, breaks=10, col="red" ,ylim=c(0,20))
```



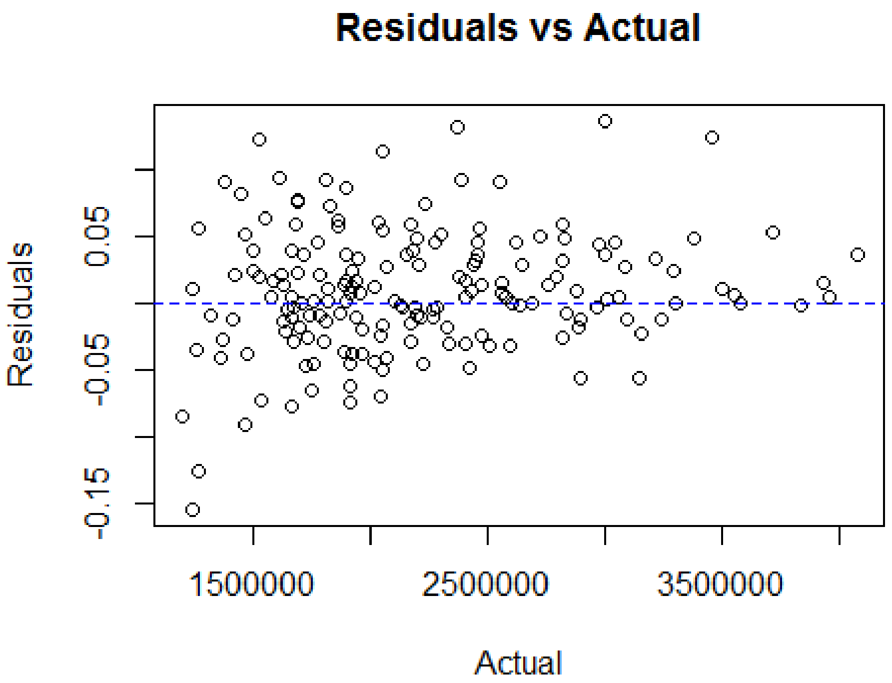

```
shapiro.test(sse_forecast$residuals)
##  Shapiro-Wilk normality test
## data:  sse_forecast$residuals
## W = 0.98587, p-value = 0.07341
```

The plot shows that the distribution of forecast errors is skewed slightly towards the left hand side of 0, and shows a left skew distribution. Looking at the Shapiro test, the p-value seems to be close to 0.05, but still isn't below it. Thus it's not normally distributed, but very close to normal distribution with left skew.

```
plot(as.matrix(sse_forecast$fitted), as.matrix(sse_forecast$residuals),mai
n="Residuals vs Fitted", xlab = "Fitted", ylab = "Residuals") # plot of fi
tted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```

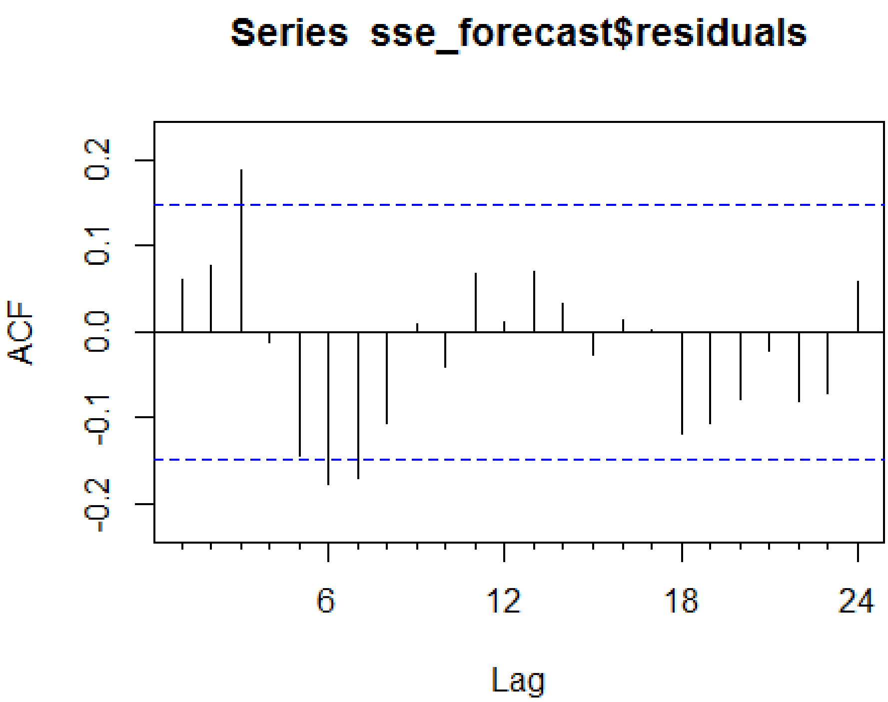



```r
plot(as.matrix(updated_travel_ts), as.matrix(sse_forecast$residuals),main=
"Residuals vs Actual", xlab = "Actual", ylab = "Residuals") # plot of fitt
ed values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```



The residual plots are pretty symmetrically distributed, tending to cluster between the actual values of (1500k-2500k). The plot shows several outliers which go upto -0.15 near 1500k. In general there aren't clear patterns. This model can be a good fit for forecasting if these outliers are removed.

```
Acf(sse_forecast$residuals)
```



Here the Acf plot shows that the autocorrelation for forecast residuals at lag 3,6,7 exceeds the significance bounds. However, we would expect one or two in 20 of the autocorrelations for the lags to exceed the 95% significance bounds by chance alone. We can verify this by carrying out the Ljung-Box test. The p-value is 0.02, which is less than 0.05. Thus there is evidence of non-zero autocorrelations in forecast residuals. Though it is closer to 0.05, so far this has proved to be our better models.

```
Box.test(sse_forecast$residuals, lag=20, type="Ljung-Box")
##  Box-Ljung test
## data:  sse_forecast$residuals
## X-squared = 34.025, df = 20, p-value = 0.02596

accuracy(sse_forecast)

##                    ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 21543.29 96382.89 72295.69 0.756152 3.456318 0.4198782
##                    ACF1
## Training set 0.02090026

print(sse_forecast)
```

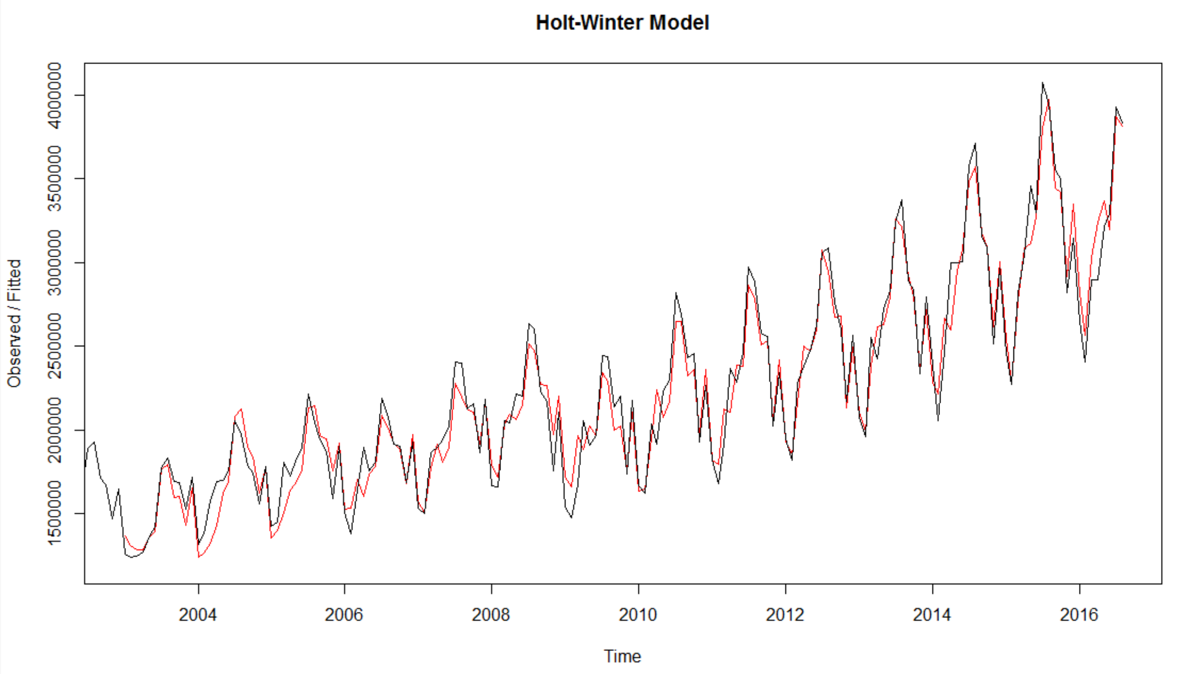
```
##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Sep 2016        3437539 3231023 3644055 3121700 3753378
## Oct 2016        3365511 3146587 3584435 3030695 3700327
## Nov 2016        2803153 2607859 2998448 2504476 3101831
## Dec 2016        3257644 3016566 3498721 2888948 3626340
## Jan 2017        2572956 2372011 2773901 2265637 2880275
## Feb 2017        2452432 2251354 2653511 2144909 2759955
## Mar 2017        2936402 2684727 3188077 2551498 3321306
## Apr 2017        3087290 2811679 3362901 2665780 3508800
## May 2017        3201511 2904728 3498294 2747620 3655401
## Jun 2017        3267131 2953460 3580802 2787413 3746850
```

```
## Jul 2017        3894858 3508463 4281252 3303918 4485797
## Aug 2017        3839291 3446509 4232072 3238584 4439998
```

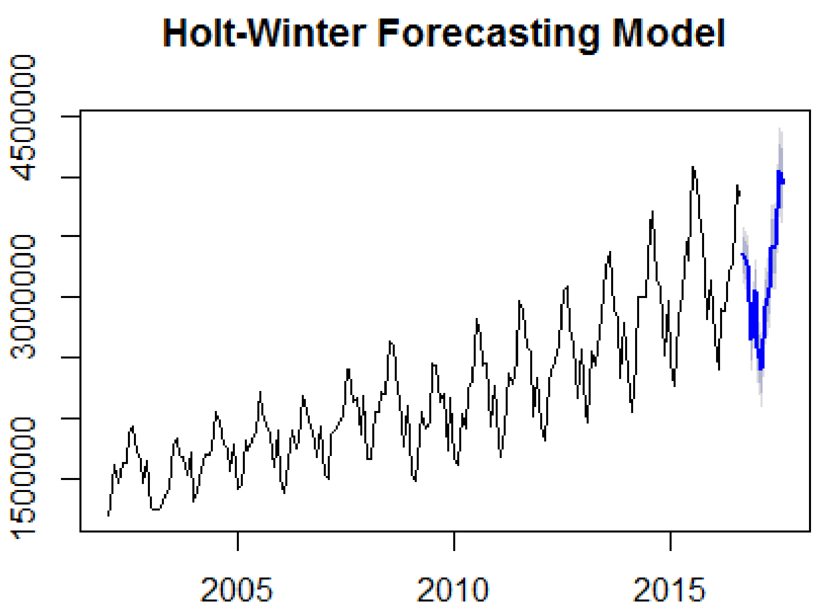
The SSE Model predicted the value to be Aug 2017: 3839291

Holt-Winters

```
holtwinter_travel <- HoltWinters(updated_travel_ts)
plot(holtwinter_travel, main="Holt-Winter Model")
```




```
holtwinter_forecast <- forecast(holtwinter_travel,h=12)
plot(holtwinter_forecast, main="Holt-Winter Forecasting Model")
```




```
print(holtwinter_travel)

## Holt-Winters exponential smoothing with trend and additive seasonal com
ponent.
## Smoothing parameters:
##  alpha: 0.3272698
```

```
##   beta : 0.02480278
##   gamma: 0.7684698
##
## Coefficients:
##              [,1]
## a    3009108.503
## b       9420.709
## s1    339813.279
## s2    249777.147
## s3   -387797.749
## s4      1381.222
## s5   -465808.327
## s6   -668332.884
## s7   -149157.471
## s8    -15436.641
## s9    313995.641
## s10   314927.905
## s11   929179.097
## s12   824051.837
```

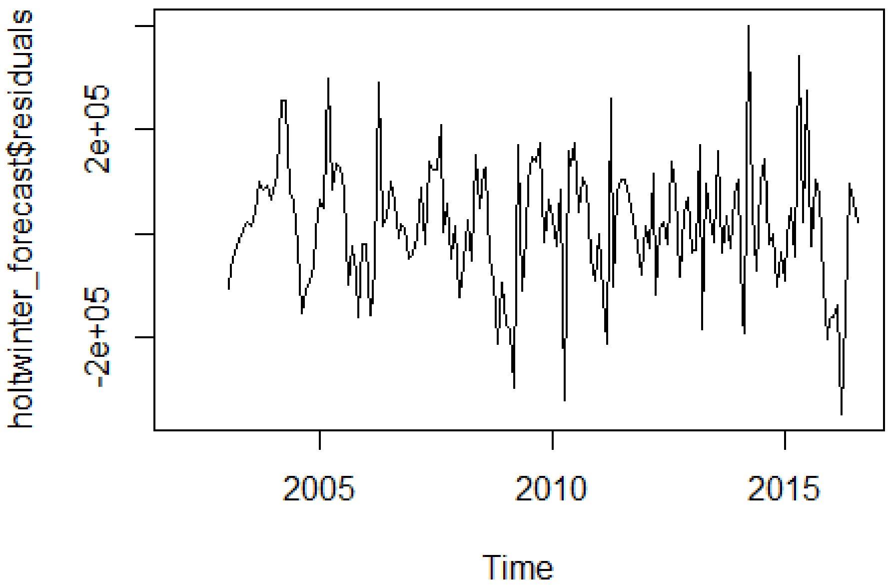
Alpha parameter specifies how smooth the level component is. If alpha is small (i.e., close to 0), more weight is given to observations from the more distant past, if alpha is large (i.e., close to 1), more weight is given to the more recent observations. In this case its 0.327 which is closer to 0 than to 1. Hence it shows more weightage towards distant past.

Beta parameter specifies how smooth the trend component is. In this case its 0.0248, which indicates older values in dataset are weighted more heavily.

Gamma parameter specifies how smooth the seasonal component is. In this case its 0.7684. This indicates that the latest value has more weight.

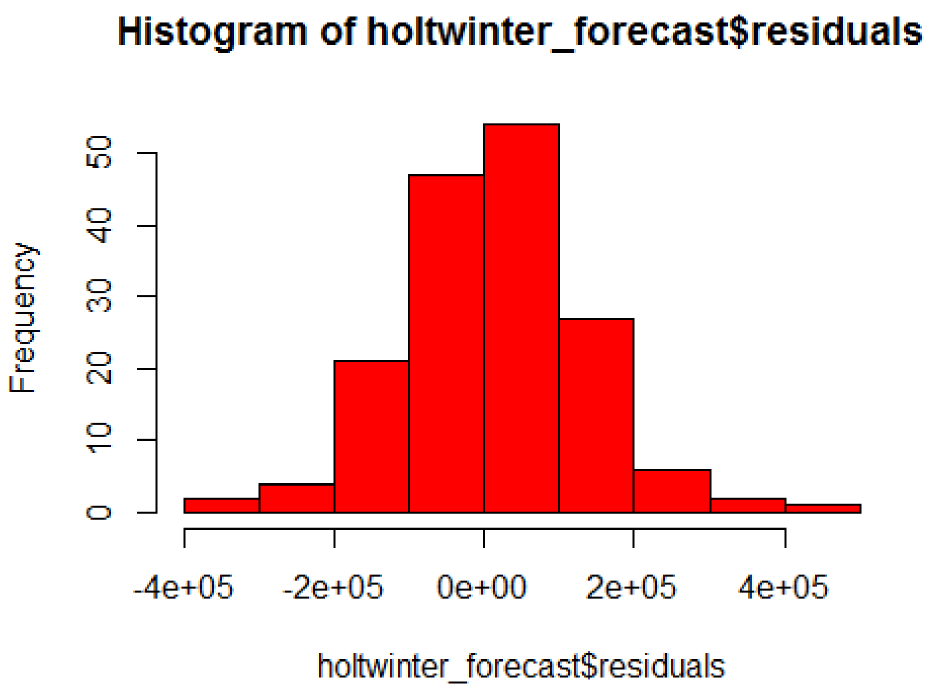
Sigma is the Standard deviation of residuals. The value of sigma is 0.0469.

```
plot(holtwinter_forecast$residuals)
```



The plot shows that the forecast errors seem to have roughly constant variance over time, with fluctuations from (2002-2016).

```
hist(holtwinter_forecast$residuals, breaks=10, col="red")
```

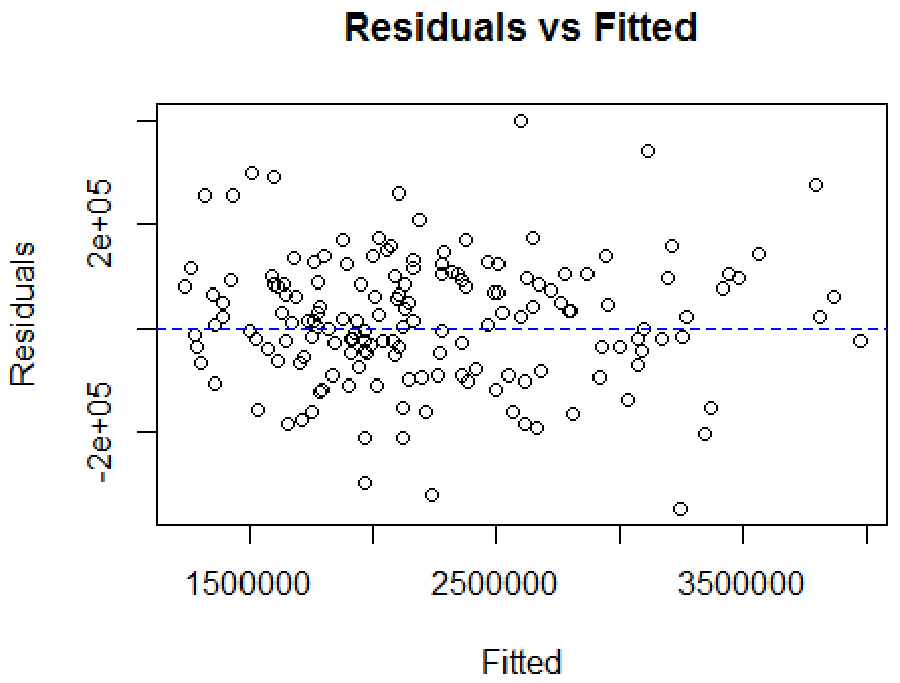



```
shapiro.test(holtwinter_forecast$residuals)

##  Shapiro-Wilk normality test
## data:  holtwinter_forecast$residuals
## W = 0.98921, p-value = 0.2442
```

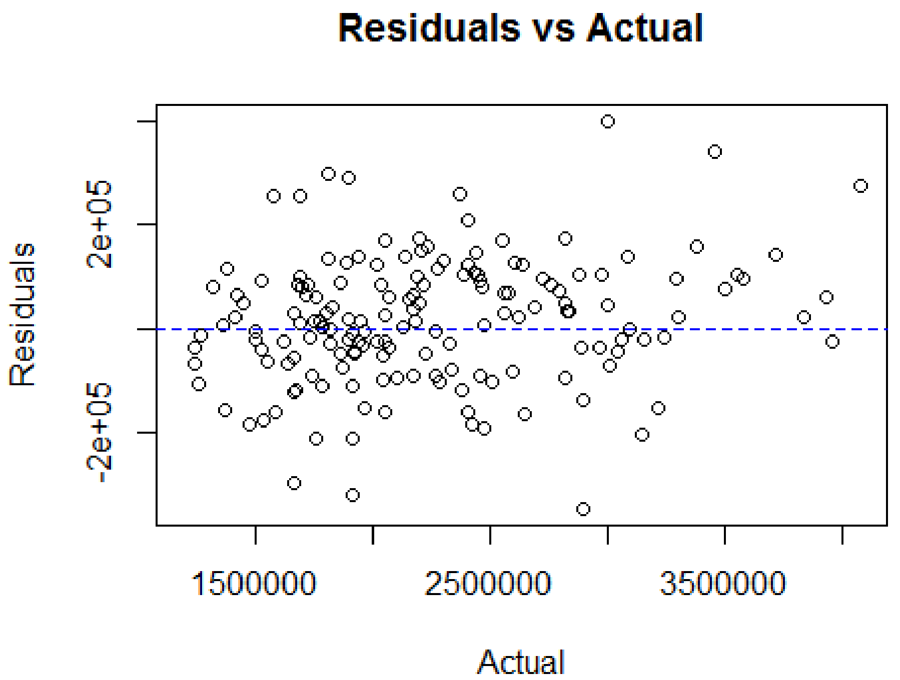
The plot shows that the distribution of forecast errors is roughly centered on zero, and is more or less normally distributed. However, the right skew is relatively small, and so it is plausible that the forecast errors are normally distributed with mean zero. This can be seen with the Shapiro test as well, were the p-value is greater than 0.05.

Residual Plots

```
plot(as.matrix(holtwinter_forecast$fitted), as.matrix(holtwinter_forecast$
residuals),main="Residuals vs Fitted", xlab = "Fitted", ylab = "Residuals"
) # plot of fitted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```

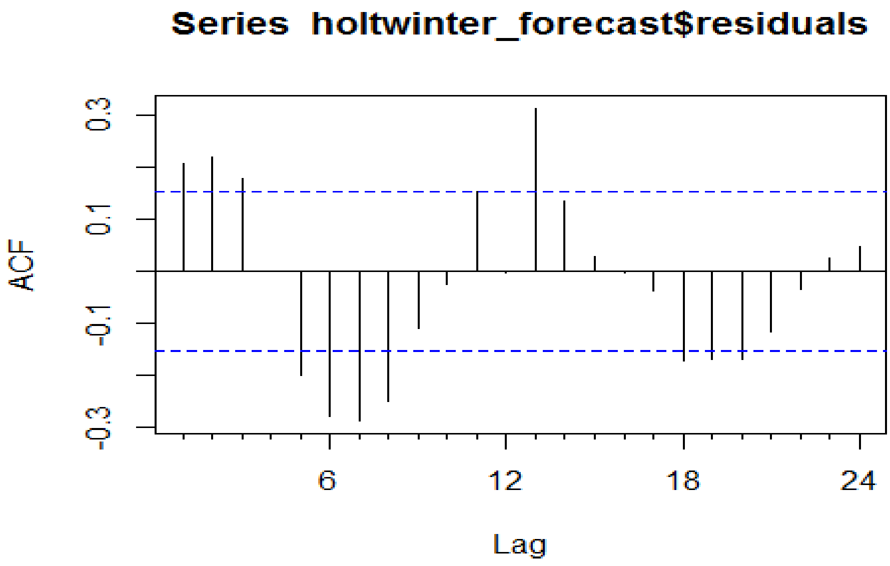



```
plot(as.matrix(updated_travel_ts), as.matrix(holtwinter_forecast$residuals
),main="Residuals vs Actual", xlab = "Actual", ylab = "Residuals") # plot
of fitted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```



The residual plots are pretty symmetrically distributed. The plot shows several outliers. In general there aren't clear patterns. This model can be a good fit for forecasting if these outliers are removed.

```r
Acf(holtwinter_forecast$residuals)
```




```r
Box.test(holtwinter_forecast$residuals, lag=20, type="Ljung-Box")

##  Box-Ljung test
## data:  holtwinter_forecast$residuals
## X-squared = 108.75, df = 20, p-value = 3.308e-14
```

With the Acf plot, we see if the forecast residuals shows non-zero autocorrelations. With lags at 0-8,13 it exceeds the significance bounds. We can verify this with the he Ljung-Box test. With p-value way less than 0.05 it proves that it shows non-zero autocorrelations.

```r
accuracy(holtwinter_forecast)

##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 15175.03 123247.5 95446.26 0.4461066 4.438431 0.5543318
##                 ACF1
## Training set 0.20383

print(holtwinter_forecast)

##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Sep 2016        3358342 3201116 3515569 3117886 3598799
## Oct 2016        3277727 3111894 3443560 3024107 3531347
## Nov 2016        2649573 2475167 2823979 2382841 2916304
## Dec 2016        3048173 2865214 3231131 2768361 3327984
## Jan 2017        2590404 2398902 2781906 2297527 2883281
## Feb 2017        2397300 2197255 2597345 2091357 2703242
## Mar 2017        2925896 2717300 3134492 2606876 3244916
## Apr 2017        3069038 2851876 3286199 2736917 3401158
## May 2017        3407891 3182143 3633638 3062640 3753141
## Jun 2017        3418243 3183886 3652601 3059825 3776662
## Jul 2017        4041915 3798919 4284912 3670284 4413547
## Aug 2017        3946209 3694541 4197877 3561316 4331102
```

The Holt-Winter Model predicted the value to be Aug 2017: 3946209

ARIMA or Box-Jenkins

Looking at the time series plot of the dataset, we understand that it is not stationery. This can be confirmed with the KPSS Test

```
kpss.test(updated_travel_ts)

##
##  KPSS Test for Level Stationarity
## data:  updated_travel_ts
## KPSS Level = 3.5156, Truncation lag parameter = 3, p-value = 0.01
```

KPSS Test says differences is required if p-value is < 0.05. The P-value we got was 0.01 <0.05.

The NSDIFFS function tells us how many differences we need for the dataset.

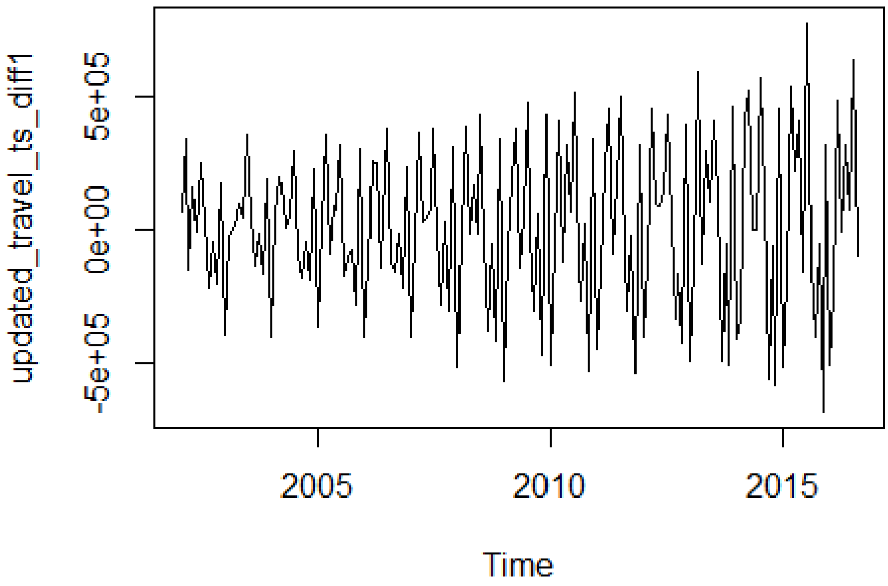
```
nsdiffs(updated_travel_ts)

## [1] 1
```

Hence performing the first difference, and executing the nsdiffs function on the differenced dataset to check if we need more differences. Seasonality component is required, as the dataset shows little seasonality. If that component is added to the ARIMA model, it will perform better.

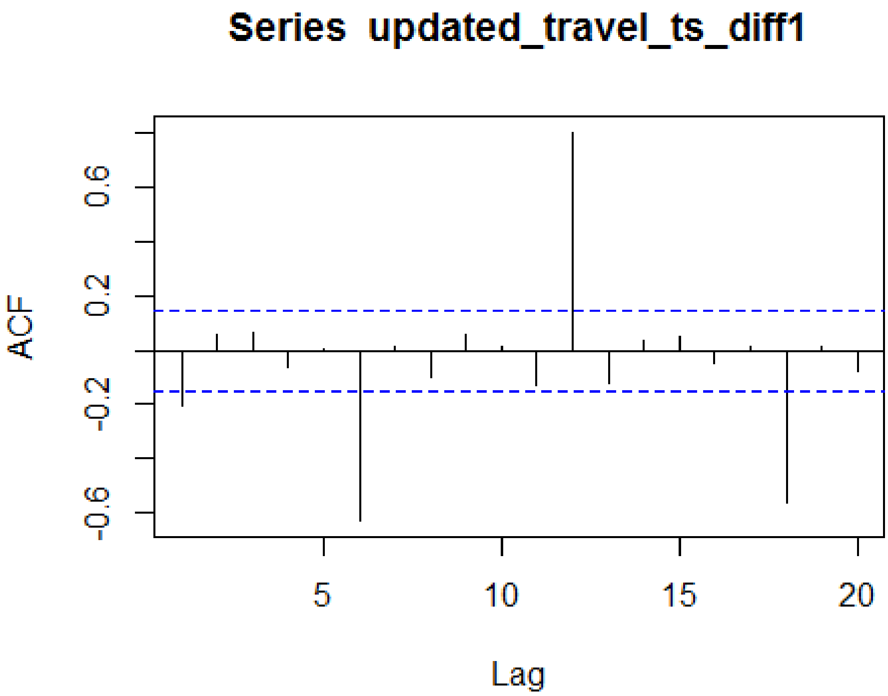
```
updated_travel_ts_diff1 <- diff(updated_travel_ts, differences=1)
ndiffs(updated_travel_ts_diff1)

## [1] 0

kpss.test(updated_travel_ts_diff1)

## Warning in kpss.test(updated_travel_ts_diff1): p-value greater than pri
nted
## p-value

##  KPSS Test for Level Stationarity
## data:  updated_travel_ts_diff1
## KPSS Level = 0.023059, Truncation lag parameter = 3, p-value = 0.1
```

After running the KPSS test on the first difference data, we see that the p-value 0.1 > 0.05, hence the dataset is now stationary.

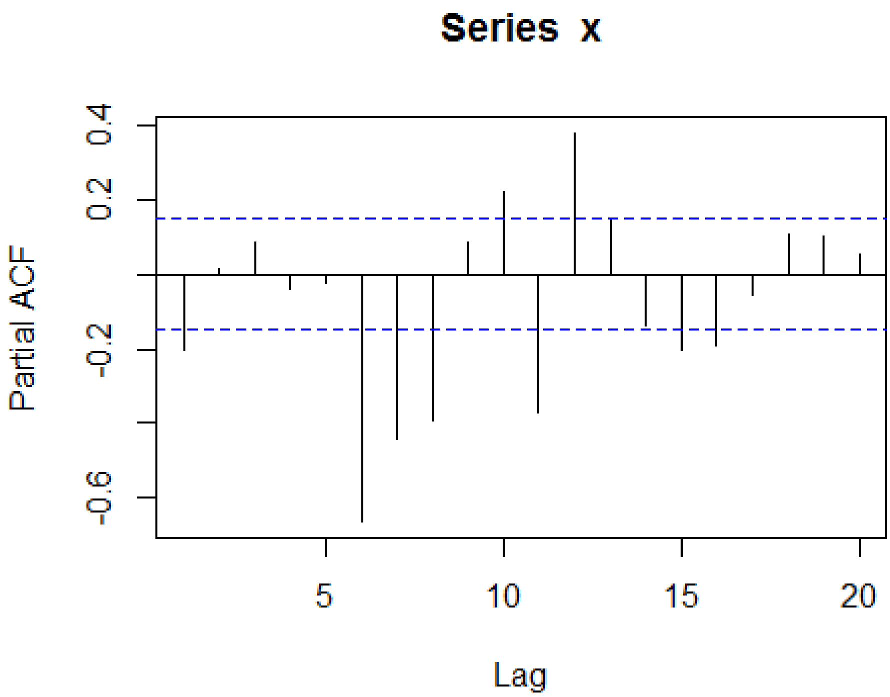
```
plot(updated_travel_ts_diff1)
```




```
Acf(updated_travel_ts_diff1, lag.max=20)
```



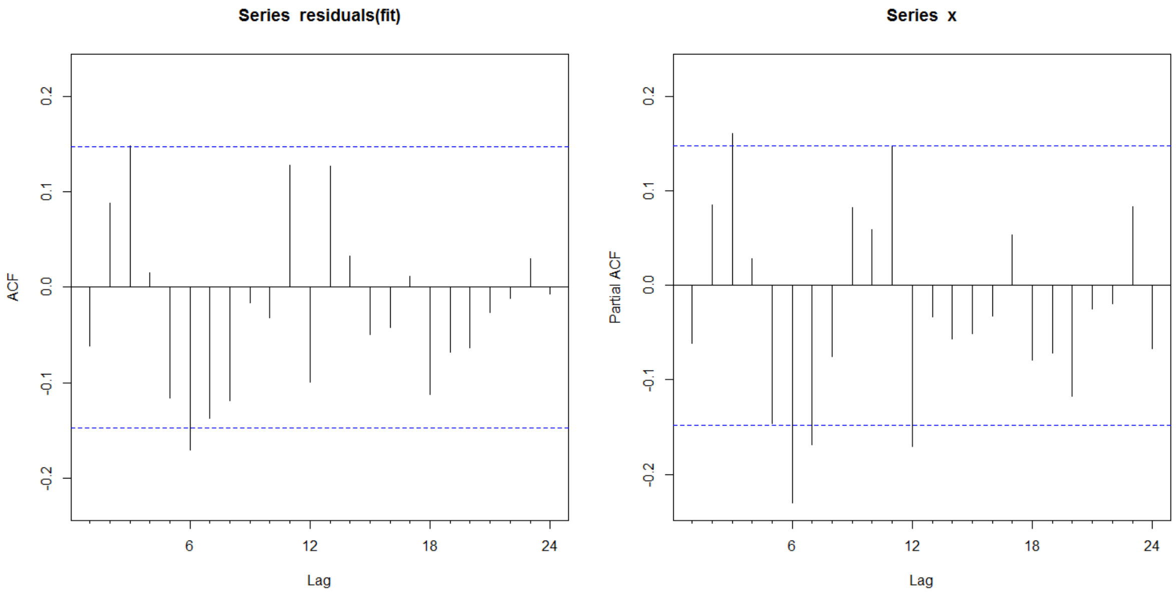
```
Pacf(updated_travel_ts_diff1, lag.max=20)
```



Based on the ACF and PACF plots, we can derive with an ARIMA model manually.

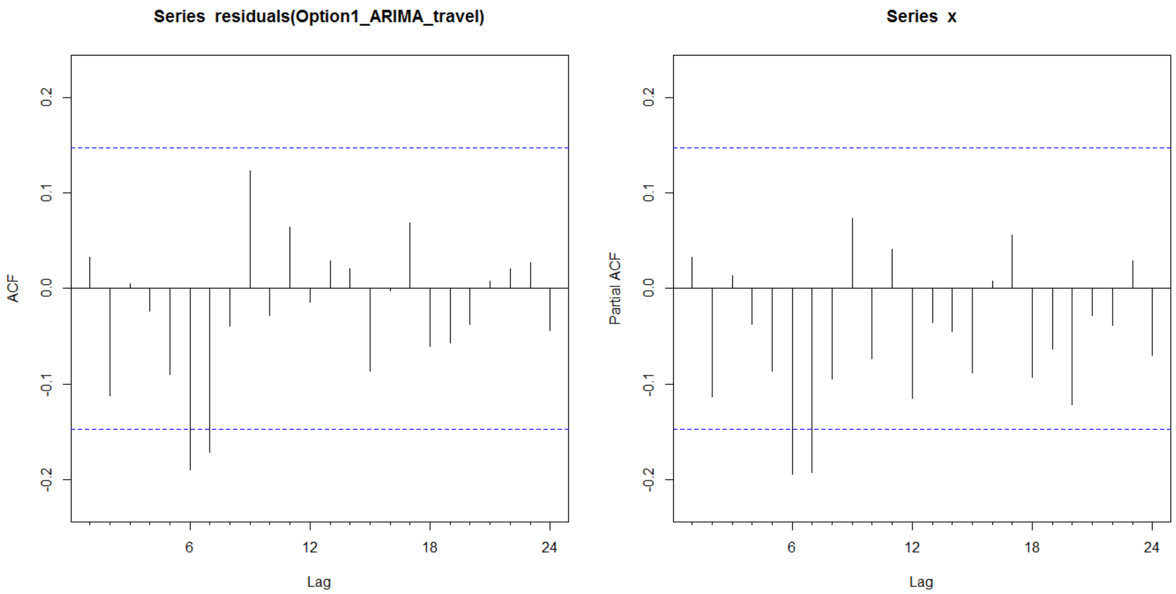
The spike at lag 1 above the significance level in the ACF suggests a non-seasonal MA(1) component, and the significant spike at lag 6 and lag 12 in the ACF suggests a seasonal MA(1) component. Consequently, we begin with an ARIMA(0,1,1)(0,1,1)$_{12}$ model, indicating a first and seasonal difference, and non-seasonal and seasonal MA(1) components. (By analogous logic, we could also have started with an ARIMA(1,1,0)(1,1,0)$_{12}$ model, it shows AR(1) for both non-seasonal and seasonal component). After running an ARIMA function, we need to check the Acf and Pacf of the residuals to make sure, the analysis is complete.

```
fit <- Arima(updated_travel_ts, order=c(0,1,1), seasonal=c(0,1,1))
par(mfrow=c(1,2))
Acf(residuals(fit))
Pacf(residuals(fit))
```



In the Acf plot, we see an almost significance spike at lag 3 and in Pacf an significant lag 3, indicating some additional non-seasonal terms need to be included in the model. ARIMA(1,1,2)(2,1,1)$_{12}$ is a good model if we compare AICc(4237.74), but looking at the Acf and Pacf plots we can still observe non-zero autocorrelations. The Box-Ljung Test also confirms that.

```
Option1_ARIMA_travel <- Arima(updated_travel_ts, order=c(1,1,2), seasonal=
c(2,1,1))
Box.test(residuals(Option1_ARIMA_travel), lag=16, fitdf=4, type="Ljung")

##  Box-Ljung test
## data:  residuals(Option1_ARIMA_travel)
## X-squared = 21.984, df = 12, p-value = 0.0377

par(mfrow=c(1,2))
Acf(residuals(Option1_ARIMA_travel))
Pacf(residuals(Option1_ARIMA_travel))
```

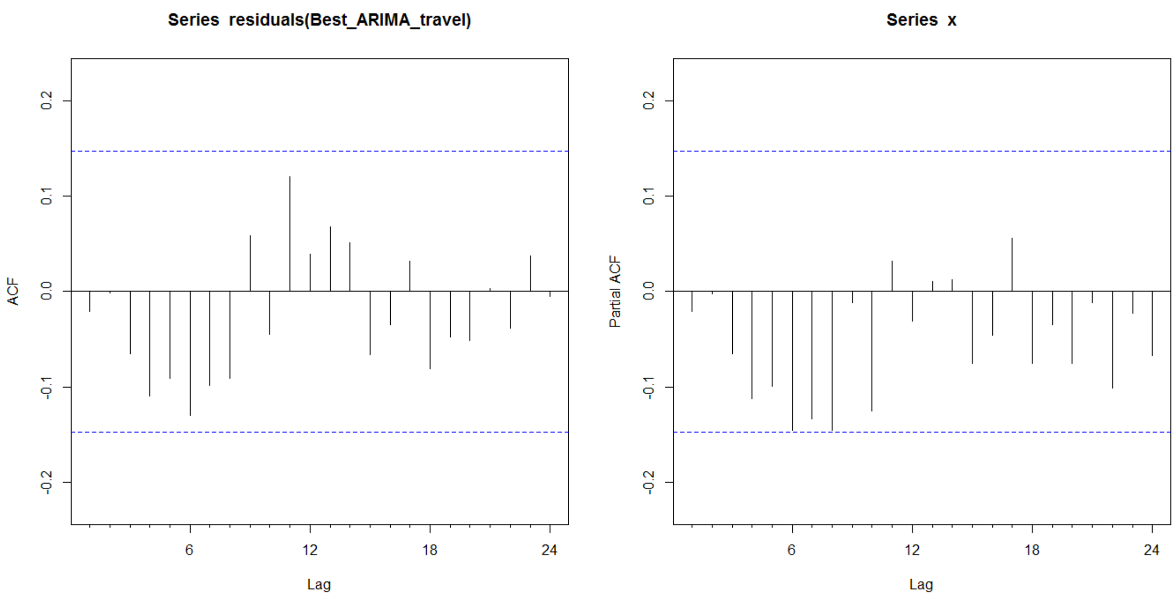


After several iterations I arrived at the best **ARIMA(2,1,2)(2,1,0)$_{12}$** with no autocorrelations and an AICc close to the option model given above.

```
Best_ARIMA_travel <- Arima(updated_travel_ts, order=c(2,1,2),
seasonal=c(2,1,0))
print(Best_ARIMA_travel)
```

```
## Series: updated_travel_ts
## ARIMA(2,1,2)(2,1,0)[12]
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sar1     sar2
##       0.4141  -0.3244  -0.9809  0.7657  -0.6614  -0.3741
## s.e.  0.1664   0.1043   0.1352  0.0787   0.0790   0.0801
##
## sigma^2 estimated as 1.041e+10:  log likelihood=-2112.19
## AIC=4238.39   AICc=4239.11   BIC=4260.05
```

```
par(mfrow=c(1,2))
Acf(residuals(Best_ARIMA_travel))
Pacf(residuals(Best_ARIMA_travel))
```




```
Box.test(residuals(Best_ARIMA_travel), lag=16, fitdf=4, type="Ljung")
```

```
##  Box-Ljung test
## data:  residuals(Best_ARIMA_travel)
## X-squared = 17.617, df = 12, p-value = 0.1278
```

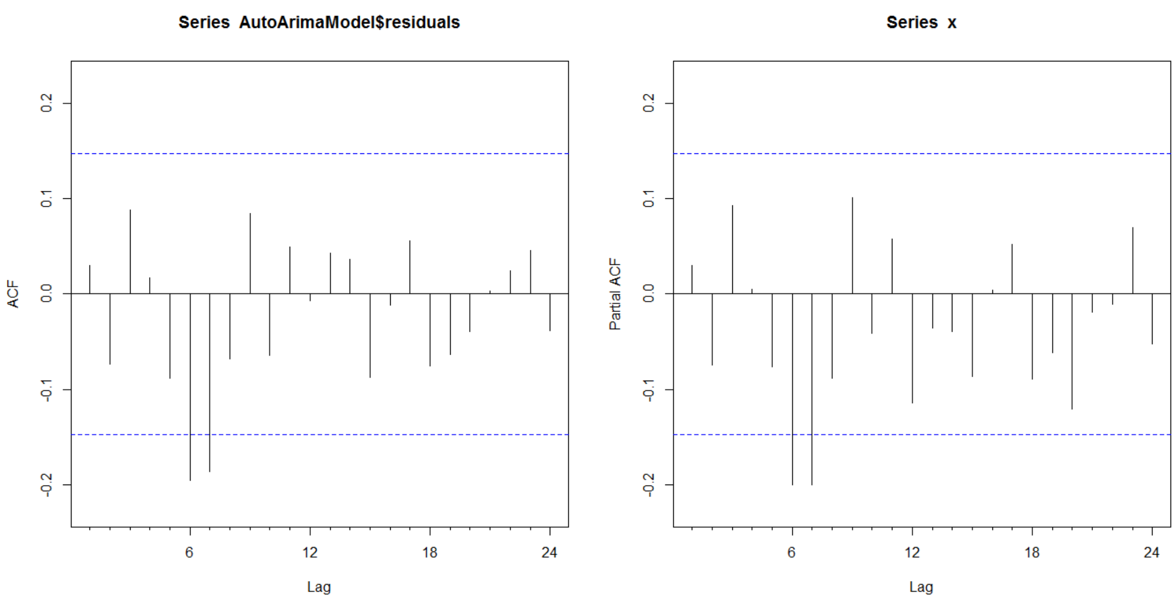
All the spikes are now within the significance limits. A Ljung-Box test also shows that the residuals have no remaining autocorrelations.
We can use the auto.Arima function in R to get the ARIMA Model

```
AutoArimaModel <- auto.arima(updated_travel_ts, stepwise=FALSE, approximat
ion=FALSE)
```

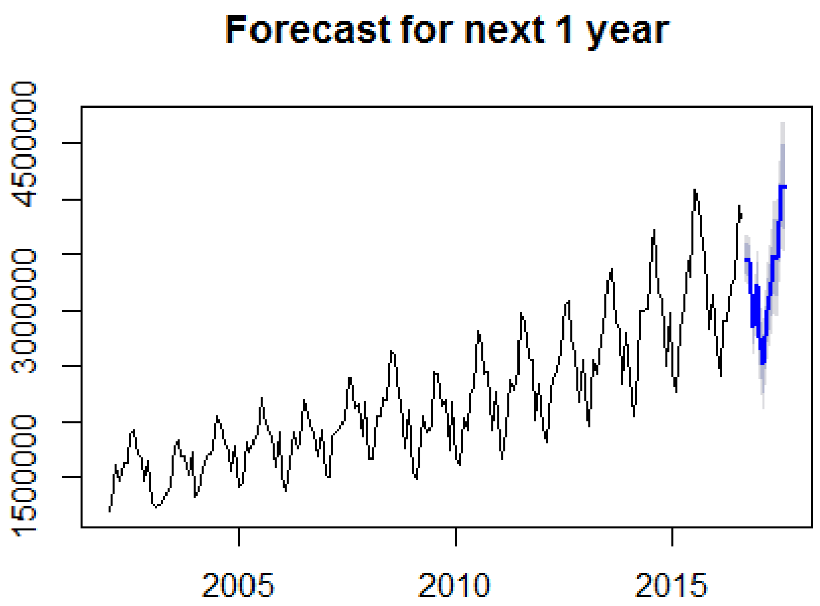
```
print(AutoArimaModel)
```

```
## Series: updated_travel_ts
## ARIMA(0,1,2)(2,1,1)[12]
##
## Coefficients:
##           ma1     ma2     sar1     sar2    sma1
##       -0.5586  0.2425  -1.1258  -0.6276  0.5921
## s.e.   0.0908  0.1114   0.0910   0.0624  0.1163
##
## sigma^2 estimated as 1.022e+10:  log likelihood=-2112.69
## AIC=4237.39   AICc=4237.93   BIC=4255.95
```

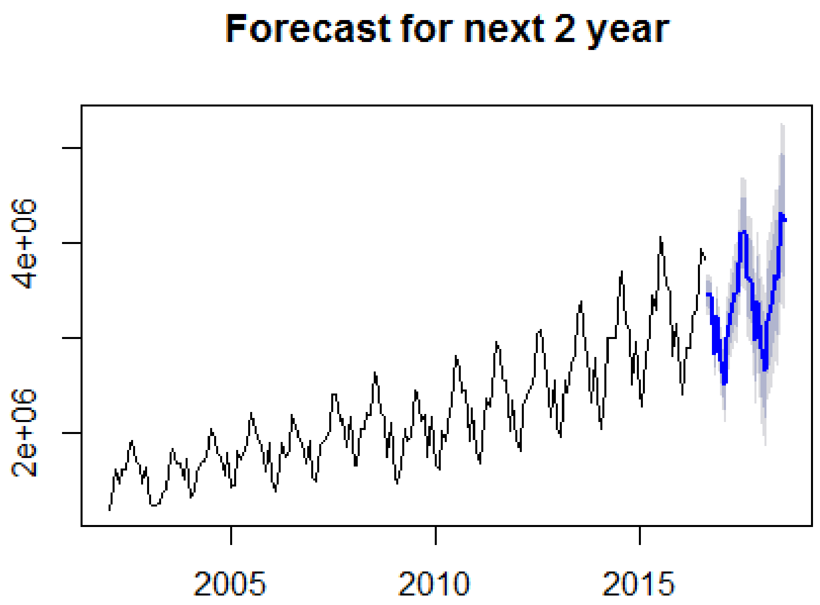
The Auto.Arima Function gives u ARIMA(0,1,2)(2,1,1)$_{12}$ which is different from the best model I derived to ARIMA(2,1,2)(2,1,0)$_{12}$. The AICc is better in the auto.arima model. But looking at the Acf and Pacf plots, we see lags outside of the confidence line thus showing non-zero autocorrelations. This can be confirmed with the Box-Ljung test.



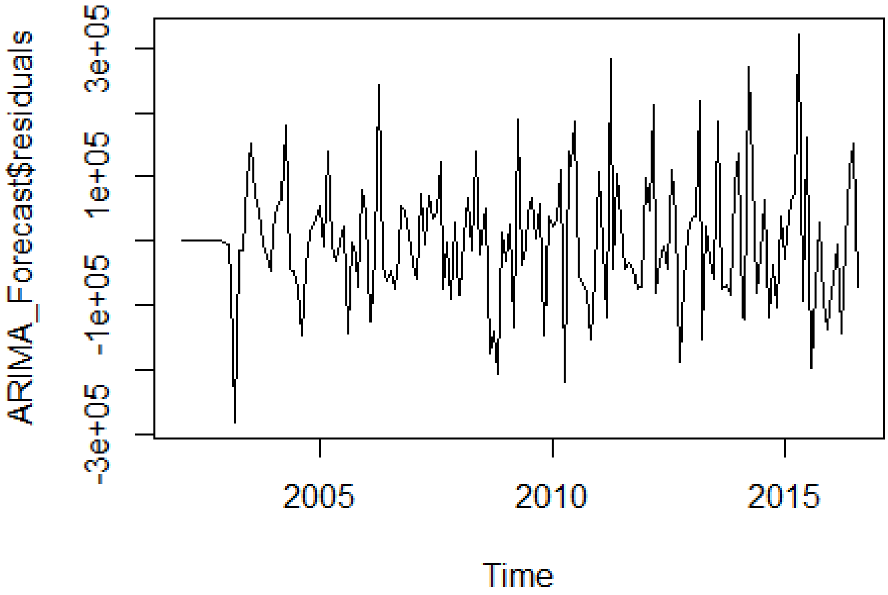

```
Box.test(residuals(AutoArimaModel), lag=16, fitdf=4, type="Ljung")
##  Box-Ljung test
## data:  residuals(AutoArimaModel)
## X-squared = 23.12, df = 12, p-value = 0.02672

ARIMA_Forecast <- forecast(Best_ARIMA_travel, h=12)
plot(ARIMA_Forecast, main="Forecast for next 1 year")
```




```
plot(forecast(Best_ARIMA_travel, h=24), main="Forecast for next 2 year")
```

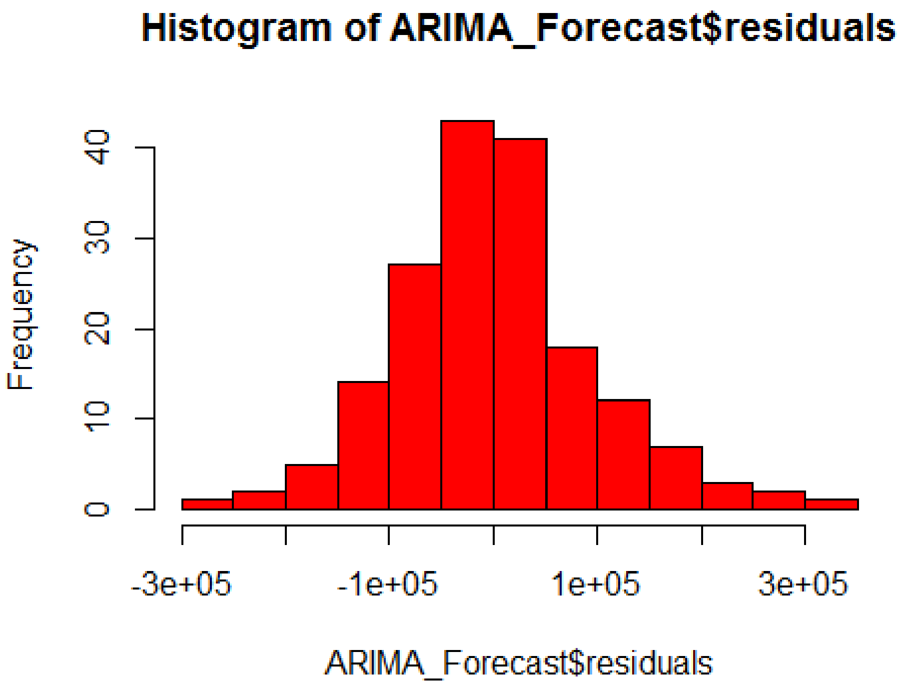


```r
plot(ARIMA_Forecast$residuals)
```



The plot shows that the forecast errors seem to have roughly constant variance over time, with fluctuations from (2002-2016).

```r
hist(ARIMA_Forecast$residuals, breaks=10, col="red")
```

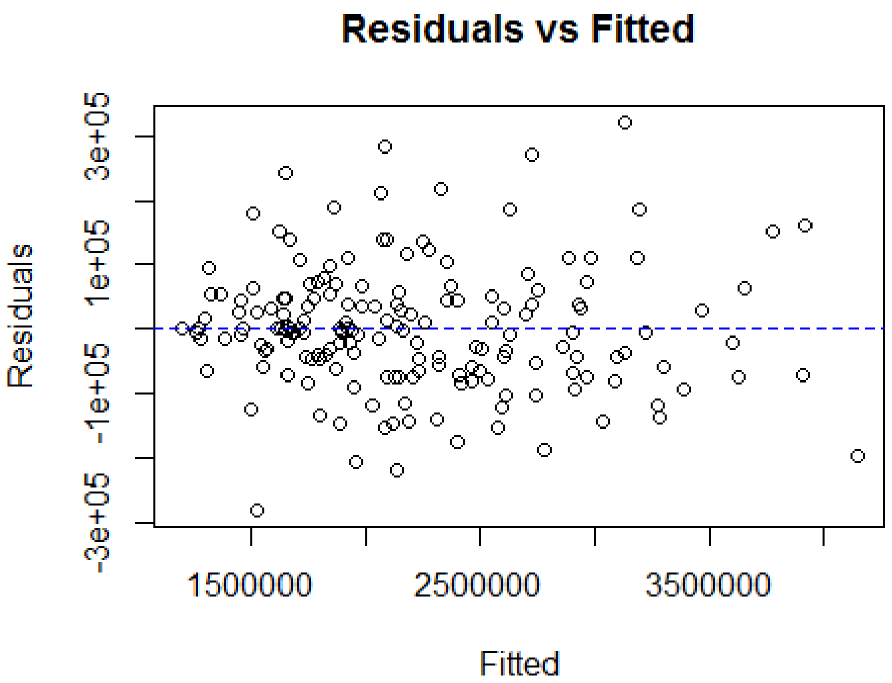



```r
shapiro.test(ARIMA_Forecast$residuals)

##
##  Shapiro-Wilk normality test
##
```

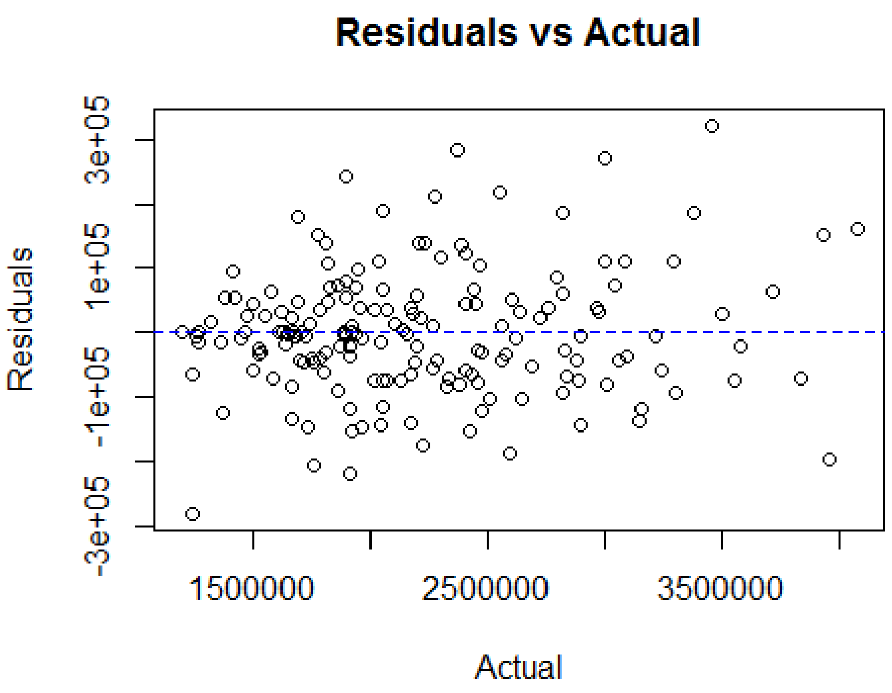
```
## data:  ARIMA_Forecast$residuals
## W = 0.98099, p-value = 0.01662
```

The plot shows that the distribution of forecast errors is roughly centered on zero, and is more or less normally distributed. However, the right skew is relatively small, and so it is plausible that the forecast errors are normally distributed with mean zero. However the Shapiro test says otherwise. With the p-value less than 0.05. We can say that it is not normally distributed.

```
plot(as.matrix(ARIMA_Forecast$fitted), as.matrix(ARIMA_Forecast$residuals)
,main="Residuals vs Fitted", xlab = "Fitted", ylab = "Residuals") # plot o
f fitted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```

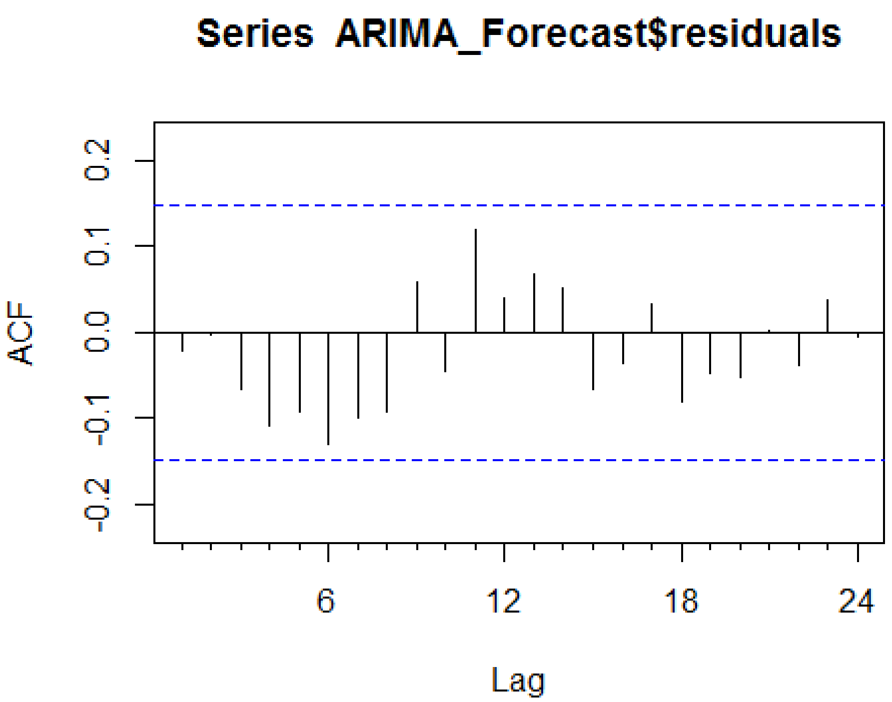



```
plot(as.matrix(updated_travel_ts), as.matrix(ARIMA_Forecast$residuals),mai
n="Residuals vs Actual", xlab = "Actual", ylab = "Residuals") # plot of fi
tted values vs residuals
abline(h=0,lty=2,col ="blue") # plotting a horizontal line at 0
```



The residual plots are almost symmetrically distributed. Though we can observe few outliers. They're clustered around 0. In general there aren't clear patterns. But this model needs improvement as we see many outliers in the plots.

```
Acf(ARIMA_Forecast$residuals)
```




```
Box.test(ARIMA_Forecast$residuals, lag=20, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ARIMA_Forecast$residuals
## X-squared = 20.094, df = 20, p-value = 0.452
```

The Acf shows that the autocorrelations for the forecast residual do not exceed the significance bounds for lags 1-24. Furthermore, the p-value for Ljung-Box test is 0.452, indicating that there is little evidence of non-zero autocorrelations at lags 1-24. With this we can suggest that this is a good model to use for this dataset for forecasting.

```
accuracy(ARIMA_Forecast)

##                    ME     RMSE      MAE         MPE     MAPE     MASE
## Training set 792.0475 96352.65 71413.39 -0.09110435 3.262138 0.414754
##                     ACF1
## Training set -0.02112524

print(ARIMA_Forecast)

##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Sep 2016        3467157 3336418 3597896 3267208 3667105
## Oct 2016        3449247 3306767 3591727 3231343 3667151
## Nov 2016        2843185 2677961 3008408 2590497 3095872
## Dec 2016        3236264 3032717 3439810 2924966 3547561
## Jan 2017        2765069 2526360 3003778 2399996 3130142
## Feb 2017        2512747 2247482 2778011 2107060 2918434
## Mar 2017        2995051 2707859 3282244 2555828 3434275
## Apr 2017        3255633 2947857 3563408 2784931 3726335
## May 2017        3480889 3153117 3808661 2979605 3982173
## Jun 2017        3464808 3117975 3811641 2934373 3995244
## Jul 2017        4118044 3753267 4482821 3560166 4675922
## Aug 2017        4101734 3719968 4483500 3517873 4685595
```

The ARIMA Model predicted the value to be Aug 2017: 4101734

Accuracy Summary

	ME	RMSE	MAE	MAPE	MASE
Naïve	15081.85	301462	240545.6	11.06543	1.397038
S-Naïve	115998.2	205853.8	172182.5	7.741379	1
Simple Moving Averages(3)	6181.567	39827.99	30795.72	1.466381	0.188203
Simple Smoothing	21543.29	96382.89	72295.69	3.456318	0.419878
Holt-Winter	15175.03	123247.5	95446.26	4.438431	0.554332
ARIMA Model	189.002	95022.39	68934.82	3.123434	0.400359

Looking at the MAPE error measure. Simple Moving Average(3) shows best accuracy measure with ARIMA next and then simple smoothing after that.

Seasonal naïve forecast:
In this case, we set each forecast to be equal to the last observed value from the same season of the year. This forecast could be useful for highly seasonal data.
Simple Moving Average:

Forecasting US Tourism

The simple moving average technique helps to get an overall idea of the trends in a data set; it is an average of the moving average order that we choose. The moving average is extremely useful for forecasting long-term trends. Moving averages "smooth out" data fluctuations or smoothens the noise present in the data. In this technique the past observations are weighted equally.
Simple Smoothing:
This method is suitable for forecasting data with no trend or seasonal pattern. In this technique, new observations are given relatively more weight in the average calculation than older observations. The forecast is a constant value that is the smoothed value of the last observation. This forecast is efficient when the data set contains no seasonality.
Holt-Winters :
Holt Winters Smoothing introduces a third parameter (gamma) to account for seasonality (or periodicity) in a data set. The Holt-Winters method can be used on data sets involving trend and seasonality (alpha, beta, and gamma). Values for all three parameters can range between 0 and 1.
ARIMA:
An autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series. ARIMA models are applied in some cases where data show evidence of non-stationarity

Conclusion

The dataset shows us the US tourism over a period of 1999 – 2016, but we cannot use the entire data for our forecast, as there was an external event occurred which was the reason in the drop of the numbers. After the trimming, we noticed that the data had trend and seasonal components. Naïve forecast would have been a bad choice for the forecasting model, and by running the analysis, it proved so. Though S-Naïve could have been used, there are other models which suit the dataset better, and will make a better model. After applying the various forecasting models to dataset, I can conclude that the ARIMA model is best model to choose for the forecast. The model can be justified with the Acf plots, the Box-Ljung tests and shows that it uses most of the data for analysis and thus presents the forecast.
Over the next 1 year the values there will be a slight increase in the numbers. Same goes for the forecast for the next 2 years as well.

Final Question

I would give an A grade to myself. The reasons being, in the beginning of the class, I wasn't proficient in R nor in the forecasting techniques or analysis. With this course I am able to understand which accuracy measure to choose for a dataset. How does the forecasting help in business and what analysis we can bring out of data. I understand now that, we would need more information about the dataset, to understand and forecast it better than compared to the simple application of forecasting models. I understand which forecasting model will work best for a dataset. With this, I will save time on eliminating models and concentrate on models which will work best. My mid-semester and final report show my technical skills and my analytical skills. Though the final report will have a better analysis than then mid-sem report, it shows how my analysis got better.