ECON 144 Project #3

Deniz Gorgulu and Yüksel Polat Akbıyık

3/6/2022

Contents

Introduction

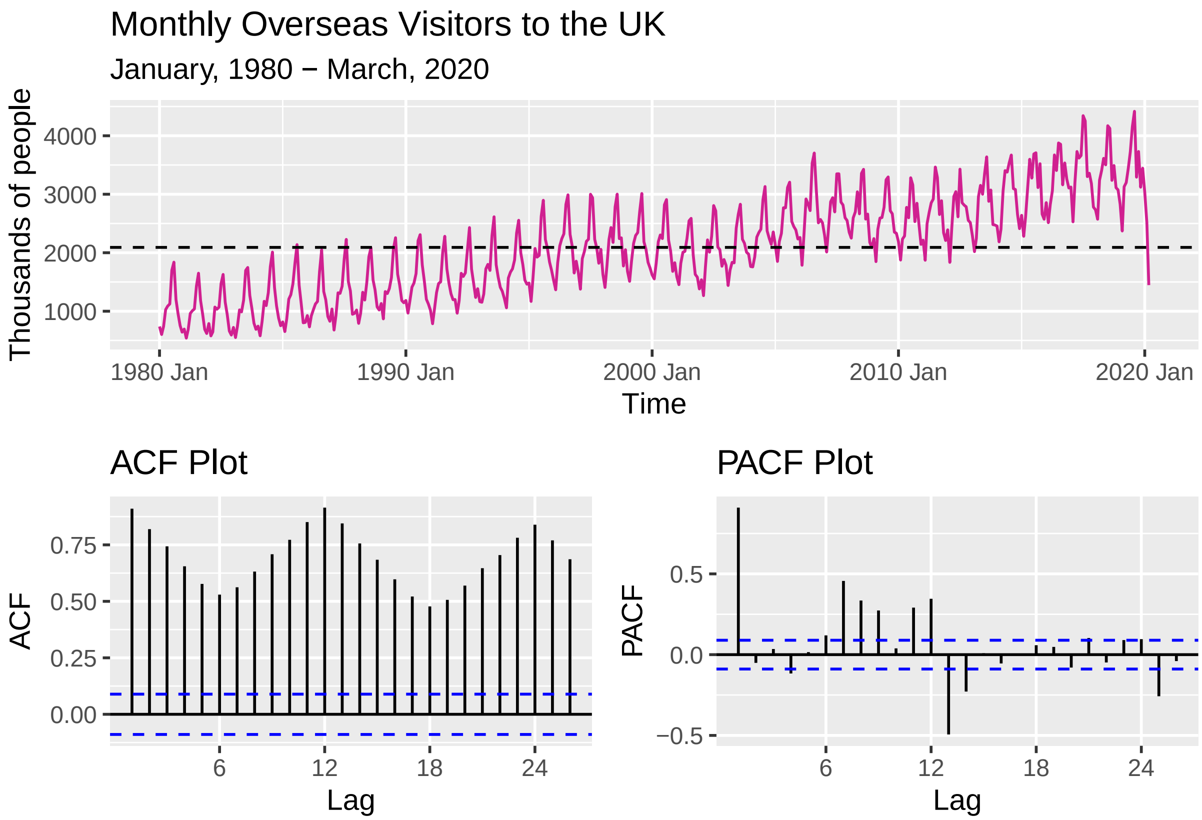
In this project we will be looking at Monthly Overseas Visitors to the UK data from the United Kingdom Office for National Statistics. The data shows the monthly non-seasonally adjusted estimates of completed international visits to the UK from January, 1980 to March, 2020, based on the International Passenger Survey data. The series consists of two variables: *Date* and *Visitors*. The *Visitors* variable represents the estimated number of people visiting the UK as overseas residents by thousands of people.

We chose to analyze this data because the tourism industry in the UK employs %5 of the workforce directly and around %8 of the work force indirectly (visitbritain.org). Therefore, it is an important sector for the second biggest economy in Europe. We expect to see that the number of overseas visitors to the UK to be increasing over the years due to the significant growth in the tourism industry during that time span and increasing travel accessibility. We also expect to see a strong seasonal component in the data. This is because there tends to be more touristic travel during holiday seasons, when it is more likely that households will have joint time off. The summer season is another period where we would expect more visitors, since the weather tends to be warmer and schools are out for break.

Results

Time series plot of the data with the respective ACF and PACF

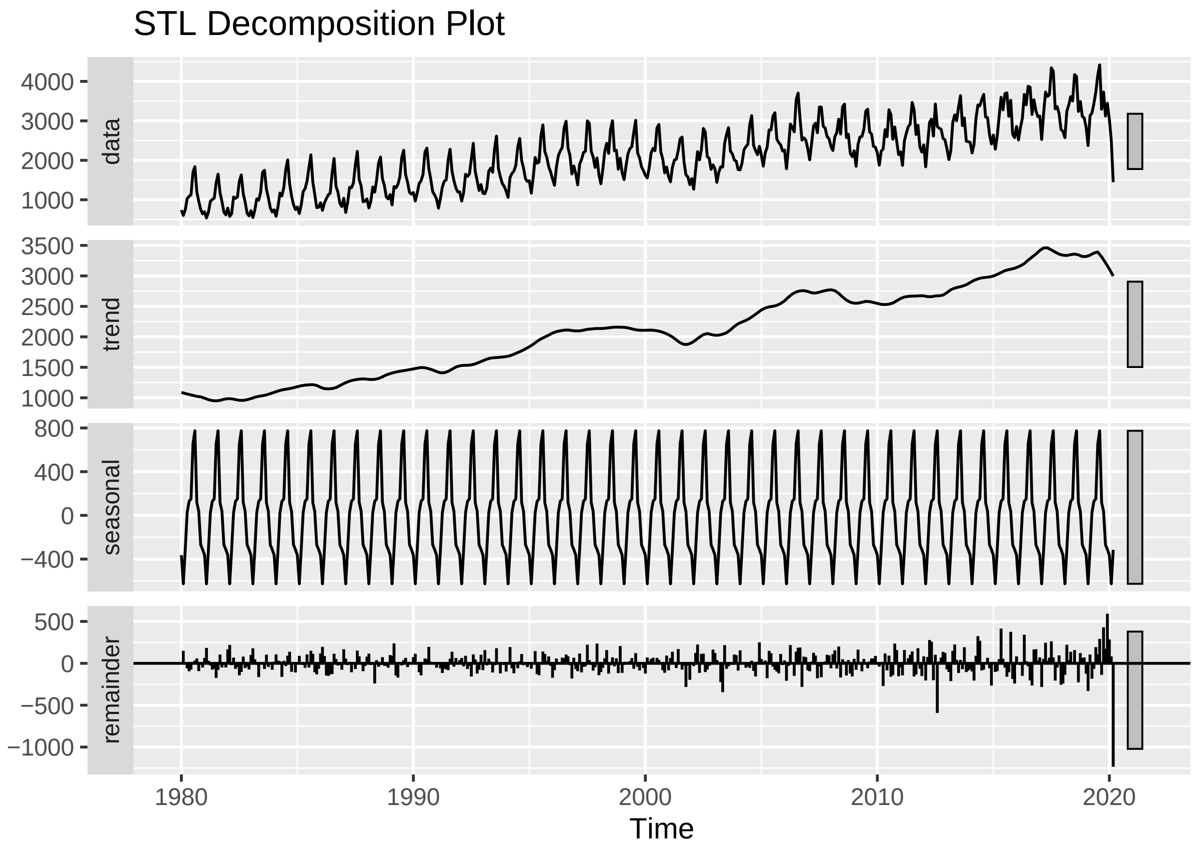
```r
ukvis = read.csv("UKVIS.csv", col.names = c("Date", "Visitors"))
ukvis$Date = yearmonth(ukvis$Date)
ukvis$Visitors = as.numeric(ukvis$Visitors)
vis = as_tsibble(ukvis, index = Date)
vis_ts = ts(ukvis[,2], start = c(1980,1), end = c(2020,3), frequency = 12)
ts_p = ggplot(vis, aes(x=Date, y=Visitors)) + geom_line(col = "violetred") +
  labs(x = "Time", y = "Thousands of people",
       title = "Monthly Overseas Visitors to the UK",
       subtitle = "January, 1980 - March, 2020")+
  geom_hline(yintercept = mean(vis$Visitors), lty=2)
acf_p = ggAcf(vis_ts)+ ggtitle("ACF Plot")
pacf_p = ggAcf(vis_ts, type = "partial")+ ggtitle("PACF Plot")
ts_p/(acf_p+pacf_p)
```



The plot of the series suggests that the number of Monthly Overseas Visitors to the UK has been steadily growing over the period of 40 years, as we had already expected. We also see some strong seasonality, which is not surprising. The data seems to be nonstationary, since it does not cross the mean that much, except for during seasonal fluctuations. The ACF plot on the lower left hand side shows some strong serial correlation. This is expected, however, as we have already determined that the series is nonstationary and shows seasonal variations.

STL decomposition of the data

```
dcmp = stl(vis_ts, s.window = "periodic")
autoplot(dcmp)+ggtitle("STL Decomposition Plot")
```

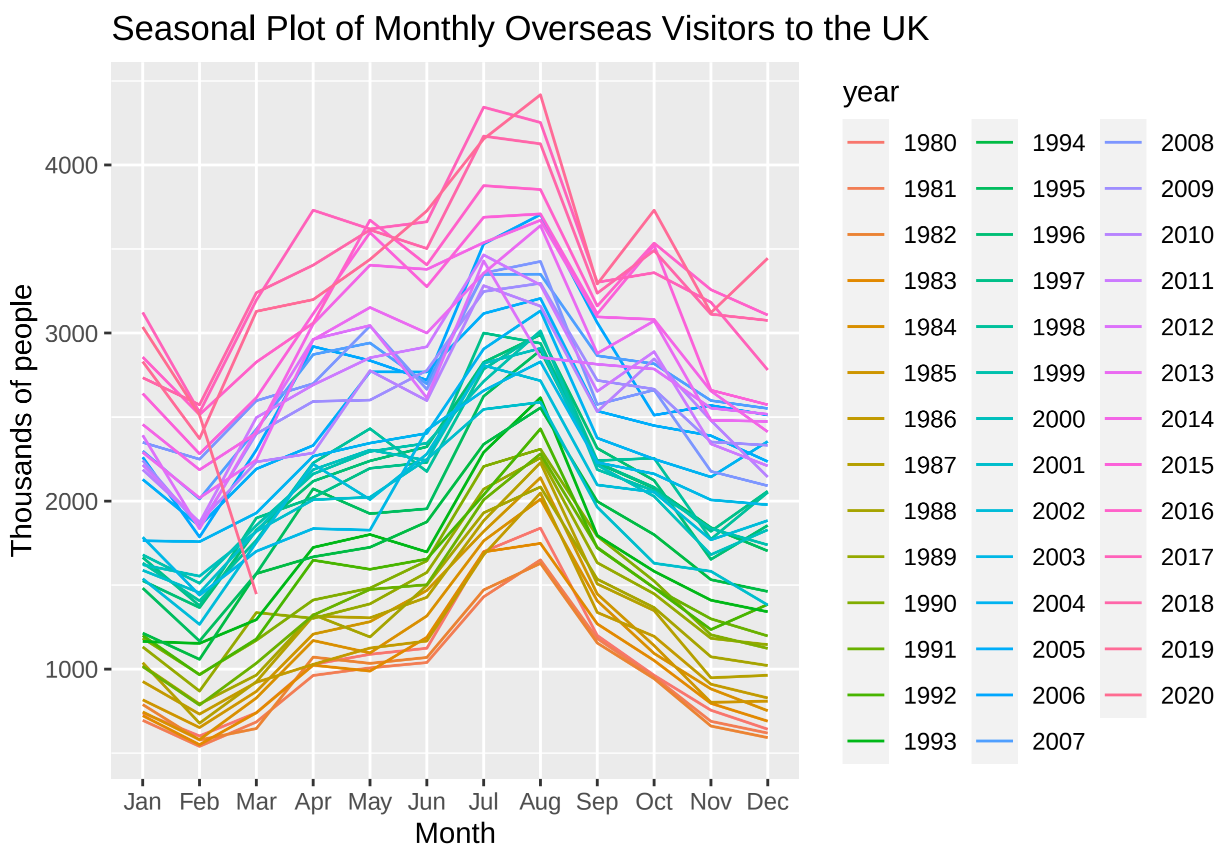


The STL decomposition plot allows us to look at trend and seasonality components more closely. There seems to be an increasing damped trend in the data along with the seasonal fluctuations. Although there seems to be a downturn in the trend near the end of the series, this is due to the sudden drop in the number of overseas visitors at the start of the COVID-19 pandemic. So, instead of being the start of a downward turning trend, this may just be the overwhelming effect of the outlier in March, 2020.

The remainder component is mostly random with some amount of constant variance up until the past few years. This suggests that most of the variation in our data has been due to the trend and seasonal components. There does seem to be some out of the ordinary increase in the variation of the series recently, implying that there is something else in the data other than the trend and seasonality that is influencing the observed type of behavior.

Seasonal analysis of the data

```
ggseasonplot(vis_ts) +
  labs(x = "Month", y = "Thousands of people",
       title = "Seasonal Plot of Monthly Overseas Visitors to the UK")
```



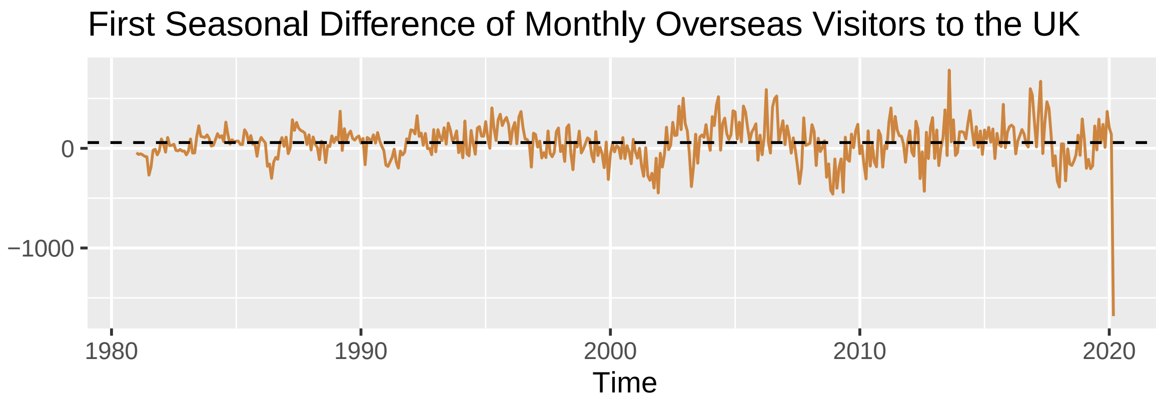
The seasonal plot above, shows a detailed breakdown of the seasonal variation in the series. We can see that the UK gets the most overseas visitors during the summer season, specifically in July and August. This may be because in most EU countries, like France and Germany, factories are likely to shut down around July and August. In addition to this, schools are also on break during the summer, which creates an opportunity for families and households to travel together. The UK seems to get the least number of overnight visitors in February, the coldest month in the year for the region. The seasonal plot has also captured the meteoric fall in the number of visitors in March, 2020.

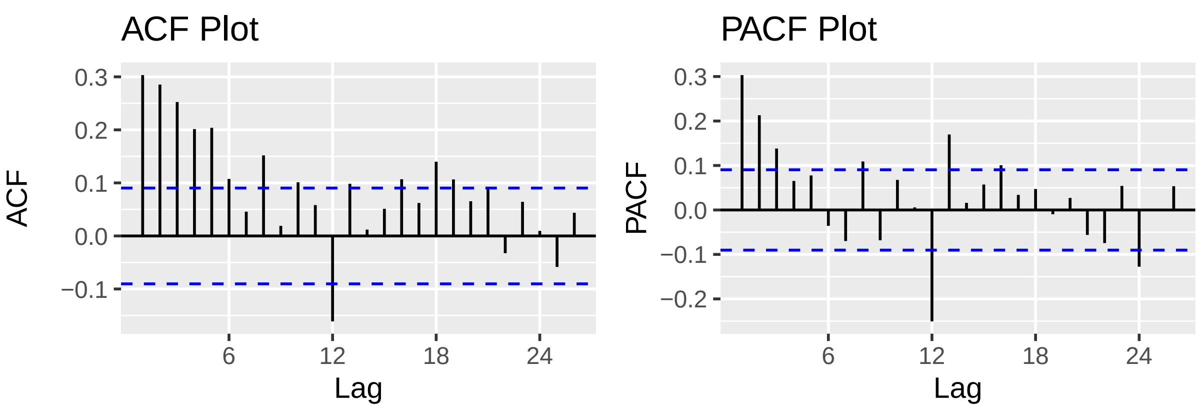
Fitting Models

Manual Model Fit

Since the data is nonstationary and shows some strong seasonal variation, we will look at the ACF and PACF of the first seasonal difference of the series to see what kind of seasonal ARIMA model to fit to the data.

```
s_diff = diff(vis_ts,12)
sd_p = autoplot(s_diff, col = "peru") +
  labs(x="Time", y="",
       title = "First Seasonal Difference of Monthly Overseas Visitors to the UK")+
  geom_hline(yintercept = mean(s_diff), lty=2)
acf_sd_p = ggAcf(s_diff)+ggtitle("ACF Plot")
pacf_sd_p = ggAcf(s_diff, type = "partial")+ggtitle("PACF Plot")
sd_p/(acf_sd_p+pacf_sd_p)
```





Taking the first seasonal difference of the series has made it stationary, so we will not additionally take the first difference. We will be looking at the ACF and PACF of the first seasonal difference to determine the order of the ARIMA model for our cycles and the order of the S-ARIMA model for our seasonality component. The ACF shows a steady decay to zero, which shows that we can fit an AR(p) process for the cycles. We also see a strong spike at lag 12 in the ACF, suggesting a S-MA(1) process for the seasonality. The PACF plot shows spikes in the first three lags and lags 12 and 13. While the first three spikes might suggest an AR(3) model for the cycles, the spikes in lags 12 and 13 may indicate that we also need a S-AR(2) term for the seasonality. The cycles could also be a combination ARMA(p,q) process, like ARMA(1,1) or ARMA(2,2).

Since it is hard to determine which seasonal ARIMA model would be best for our data from the plots alone, we will try to fit three different ARIMA models for the cycles:

- ARIMA(3,0,0)
- ARIMA(1,0,1)
- ARIMA(2,0,2)

```
mod1 = Arima(vis_ts, order = c(3,0,0), seasonal = list(order = c(2,1,1)))
mod2 = Arima(vis_ts, order = c(1,0,1), seasonal = list(order = c(2,1,1)))
mod3 = Arima(vis_ts, order = c(2,0,2), seasonal = list(order = c(2,1,1)))

summary(mod1)
```

```
## Series: vis_ts
## ARIMA(3,0,0)(2,1,1)[12]
```

```
##
## Coefficients:
##          ar1     ar2     ar3    sar1    sar2     sma1
##       0.4152  0.2875  0.2165  0.1525  0.0144  -0.8076
## s.e.  0.0526  0.0532  0.0513  0.0727  0.0646   0.0518
##
## sigma^2 estimated as 26409:  log likelihood=-3068.11
## AIC=6150.23   AICc=6150.47   BIC=6179.31
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE    MAPE      MASE        ACF1
## Training set 15.62724 159.4507 106.8746 0.3716434 5.21567 0.7127896 -0.04596464
```

```
summary(mod2)
```

```
## Series: vis_ts
## ARIMA(1,0,1)(2,1,1)[12]
##
## Coefficients:
##          ar1      ma1    sar1    sar2     sma1
##       0.9885  -0.7123  0.1333  0.0043  -0.8253
## s.e.  0.0079   0.0481  0.0708  0.0642   0.0469
##
## sigma^2 estimated as 25022:  log likelihood=-3056.26
## AIC=6124.51   AICc=6124.69   BIC=6149.44
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 9.203544 155.3732 104.5718 0.04962852 5.178288 0.6974317
##                    ACF1
## Training set 0.02925511
```

```
summary(mod3)
```

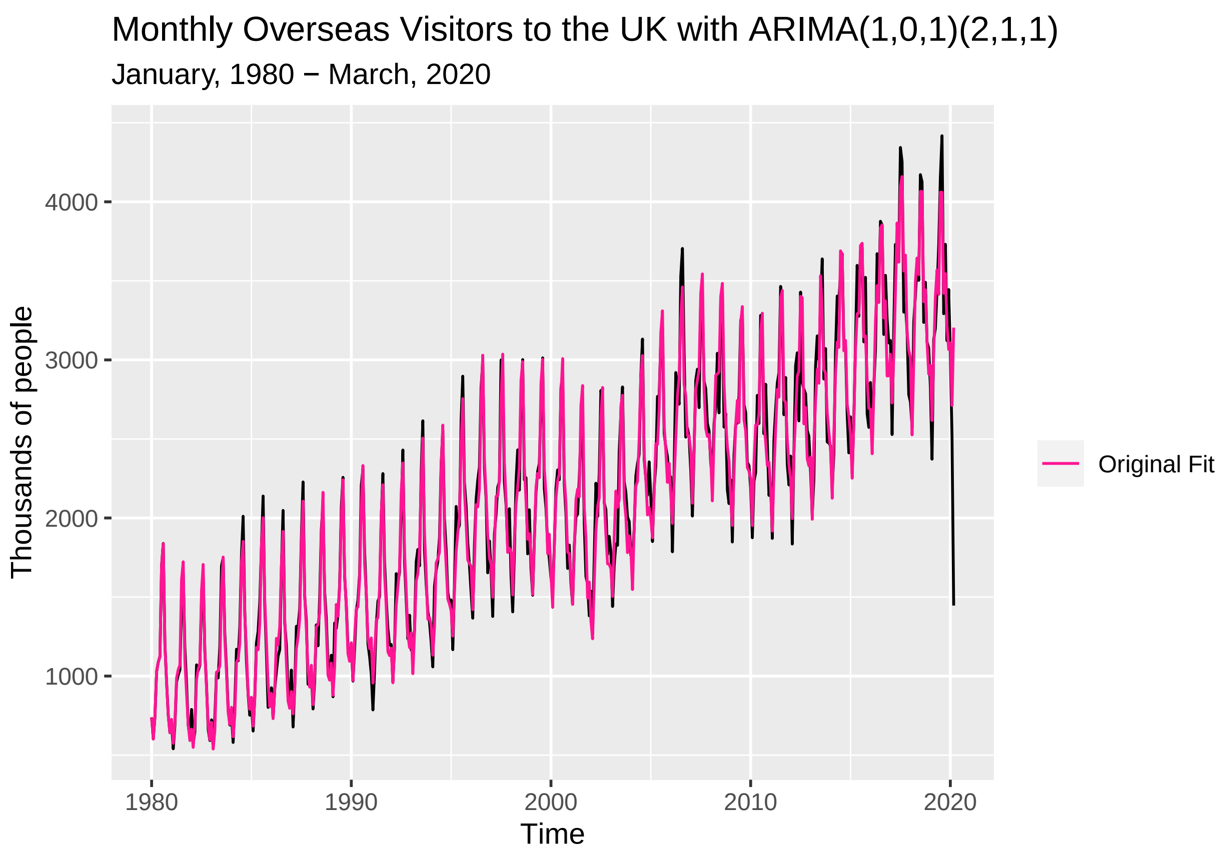
```
## Series: vis_ts
## ARIMA(2,0,2)(2,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1    sar2     sma1
##       1.5999  -0.6028  -1.2815  0.3722  0.1545  0.0196  -0.8406
## s.e.  0.2865   0.2842   0.3029  0.2346  0.0719  0.0640   0.0450
##
## sigma^2 estimated as 24987:  log likelihood=-3055
## AIC=6125.99   AICc=6126.3   BIC=6159.23
##
## Training set error measures:
##                    ME    RMSE      MAE        MPE     MAPE      MASE
## Training set 8.278959 154.934 104.5769 0.02368912 5.157184 0.6974653
##                      ACF1
## Training set -0.002481671
```

From the summaries above we can see that the ARIMA(1,0,1) has the lowest AIC value out of the three models. Therefore, our final model for the Monthly Overseas Visitors to the UK will be an ARIMA(1,0,1)(2,1,1) model.

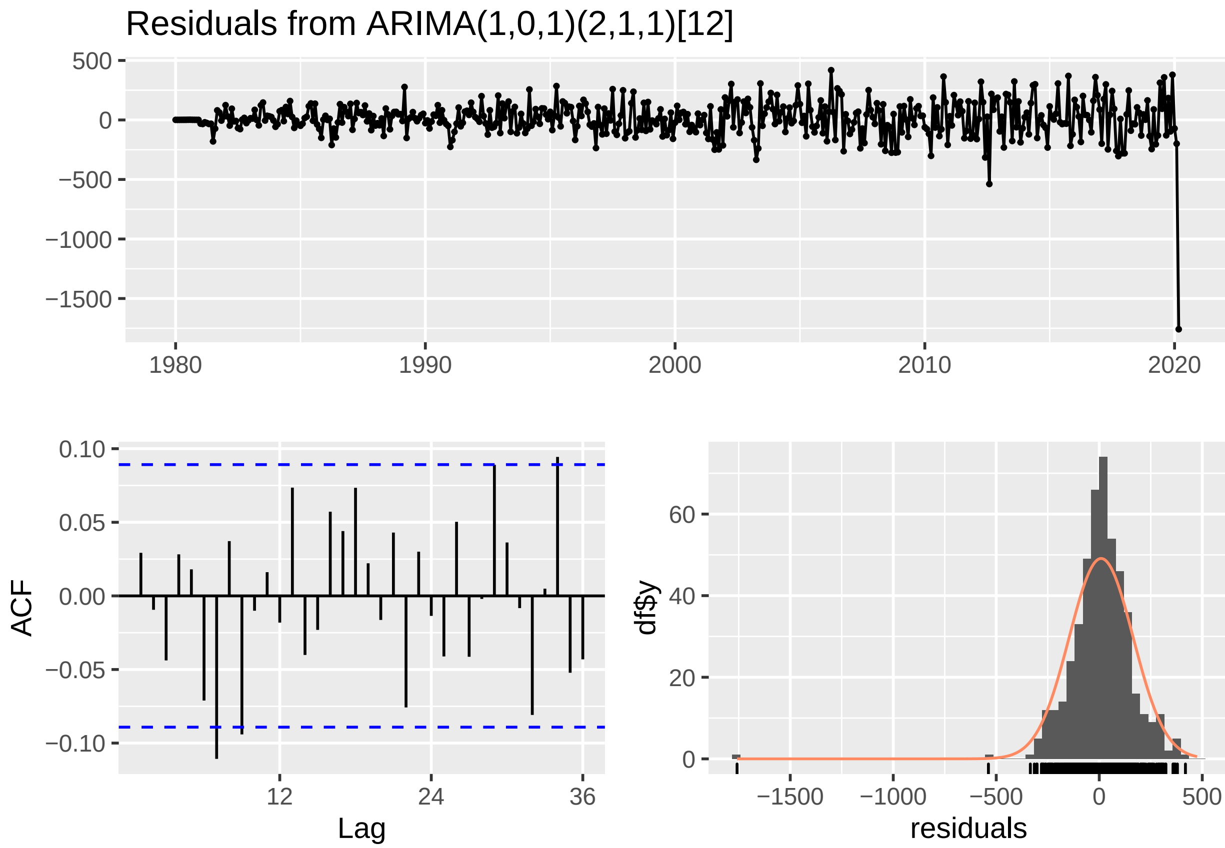
The model summary of the ARIMA(1,0,1)(2,1,1) model shows that the standard errors of the estimated coefficients are small, suggesting that the estimates are accurate. Furthermore, the estimated coefficient for the S-AR(2) is really small and close to zero, which may be indicating that we do not need that term.

We will be looking at how well the model is fitting the data through plotting the original data along with the fitted values and looking at whether the residuals are white noise or not.

```
m_og = Arima(vis_ts, order = c(1,0,1), seasonal = list(order = c(2,1,1)))
autoplot(vis_ts)+autolayer(fitted(m_og), series = "Original Fit")+
  labs(x="Time", y="Thousands of people",
       title = "Monthly Overseas Visitors to the UK with ARIMA(1,0,1)(2,1,1)",
       subtitle = "January, 1980 - March, 2020")+
  scale_color_manual("", values = "deeppink")
```




```
checkresiduals(m_og)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(2,1,1)[12]
## Q* = 29.752, df = 19, p-value = 0.05504
##
## Model df: 5.   Total lags used: 24
```

The model fit vs. original data plot above shows that the model seems to be doing a really good job at fitting the data. We will have to look at the residuals plots, however, to come to a more definitive conclusion. The ACF plot indicates that the residuals are white noise, with almost no significant spikes in the lags, except for one. Due to the small magnitude of this spike, we will treat this as negligible. The distribution plot shows that the residuals are close to being normally distributed. However, we also see that the outlier is causing the left hand tail to be heavier, distorting the shape of the distribution. We look at the results of the Ljung-Box test to officially determine whether the residuals are normally distributed or not. The p-value of the test indicates that we fail to reject to reject the null hypothesis and thus conclude that the residuals are normally distributed.

We will be comparing our model of choice with best fit models from different methods.

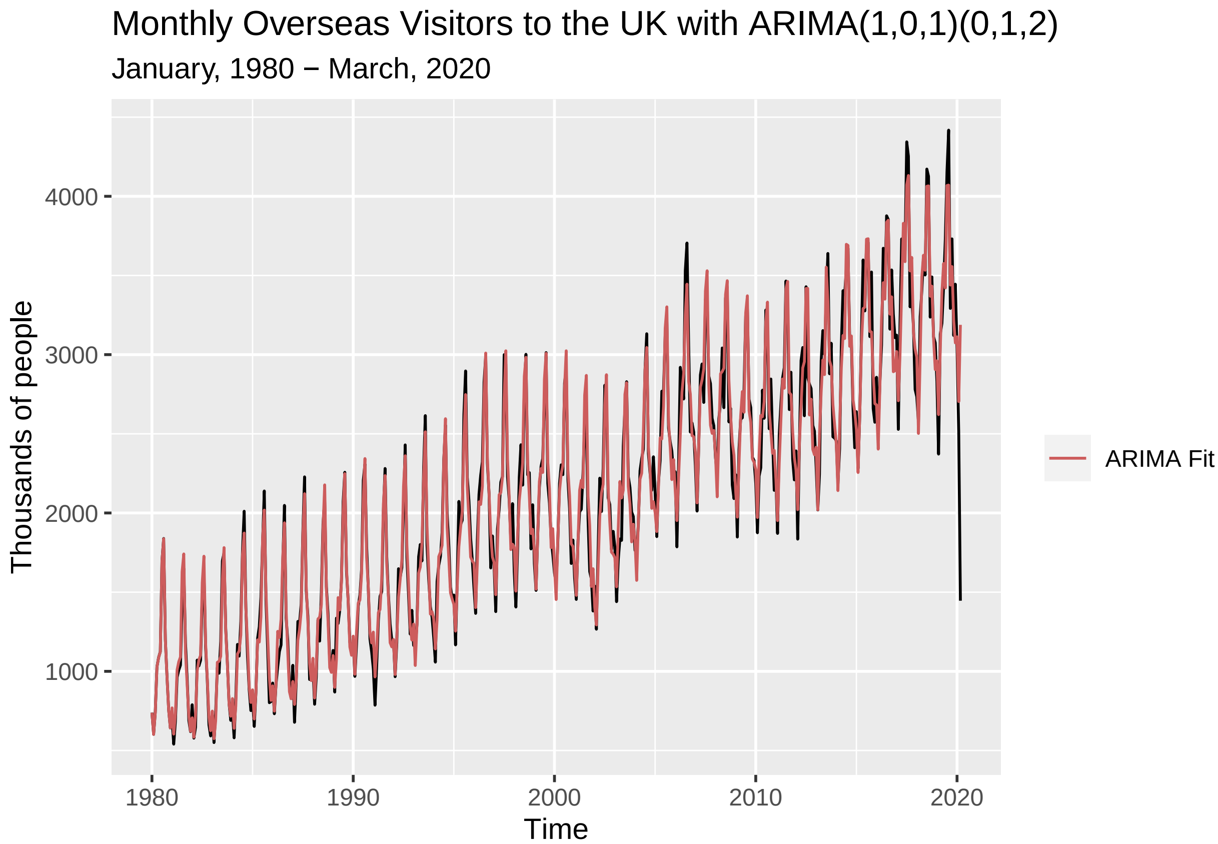
ARIMA Model Fit

```
m_arima = auto.arima(vis_ts)
summary(m_arima)
```

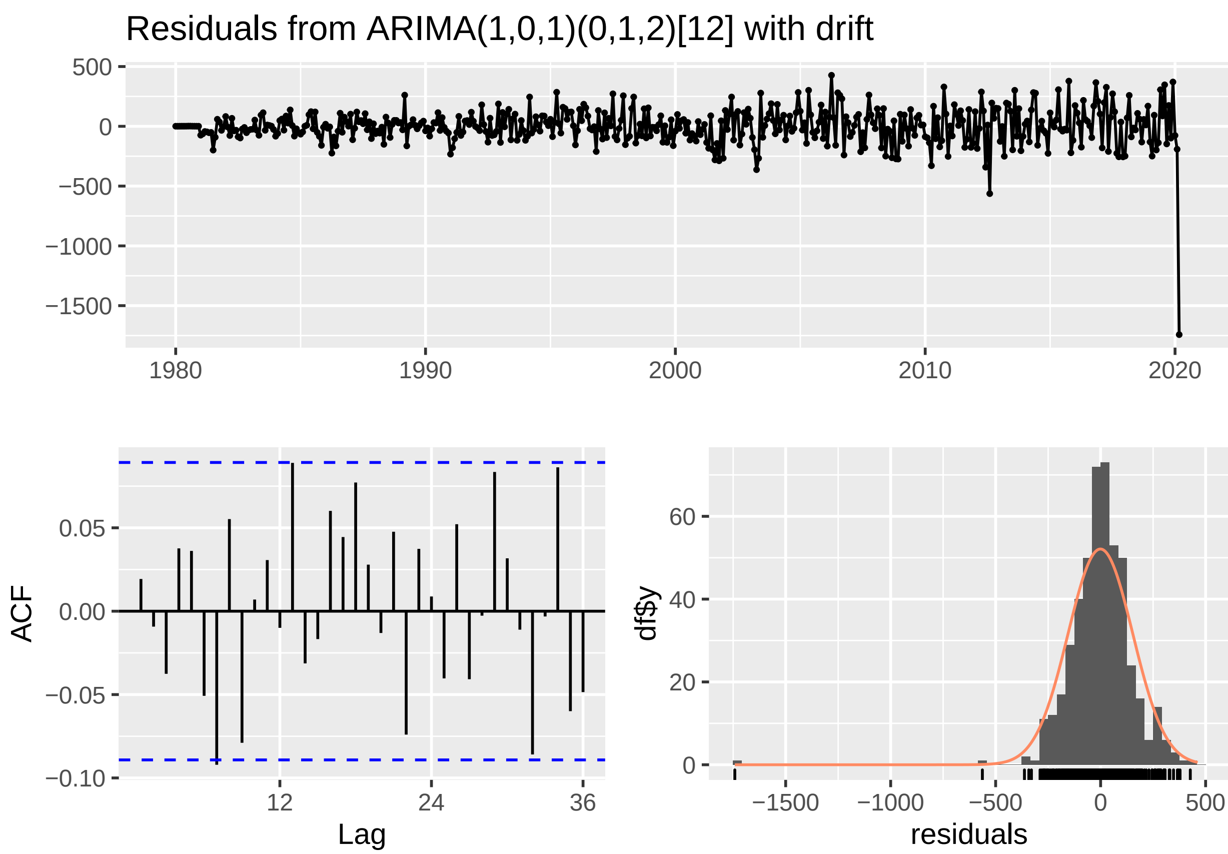
```
## Series: vis_ts
## ARIMA(1,0,1)(0,1,2)[12] with drift
##
## Coefficients:
##          ar1      ma1     sma1     sma2   drift
##       0.9418  -0.6643  -0.6882  -0.1188  4.7830
## s.e.  0.0260   0.0617   0.0540   0.0524  0.7554
##
## sigma^2 estimated as 24623:  log likelihood=-3052.83
## AIC=6117.65   AICc=6117.83   BIC=6142.58
##
## Training set error measures:
##                      ME     RMSE     MAE        MPE     MAPE      MASE
## Training set -0.9131417 154.1322 103.772 -0.7322099 5.195594 0.6920977
##                     ACF1
## Training set 0.01931226
```

The estimated best fit ARIMA model for the data is an ARIMA(1,0,1)(0,1,2) model with drift. Although the model for the cycles matches our originally estimated model, the `auto.arima` generated a S-ARIMA(0,1,2) model instead of a S-ARIMA(2,1,1) model. The standard errors of the coefficients are small except for that of the drift coefficient, which is significantly larger than the rest. Furthermore, the AIC of the model generated by `auto.arima` is smaller than the AIC of our original model, indicating that the former is a better model than the latter.

```
autoplot(vis_ts)+autolayer(fitted.values(m_arima), series = "ARIMA Fit")+
  labs(x="Time", y="Thousands of people",
       title = "Monthly Overseas Visitors to the UK with ARIMA(1,0,1)(0,1,2)",
       subtitle = "January, 1980 - March, 2020")+
  scale_color_manual("", values = "indianred")
```




```
checkresiduals(m_arima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(0,1,2)[12] with drift
## Q* = 28.347, df = 19, p-value = 0.07698
##
## Model df: 5.   Total lags used: 24
```

The model generated by `auto.arima` seems to be fitting the data as well as our original model, if not better. We would have to do a more formal test to determine which is actually better though. The residuals seem to be white noise, with no lags crossing the significance bands. Furthermore, the residuals seem to be mostly normally distributed, since we fail to reject the null hypothesis in the Ljung-Box test.

ETS Model Fit

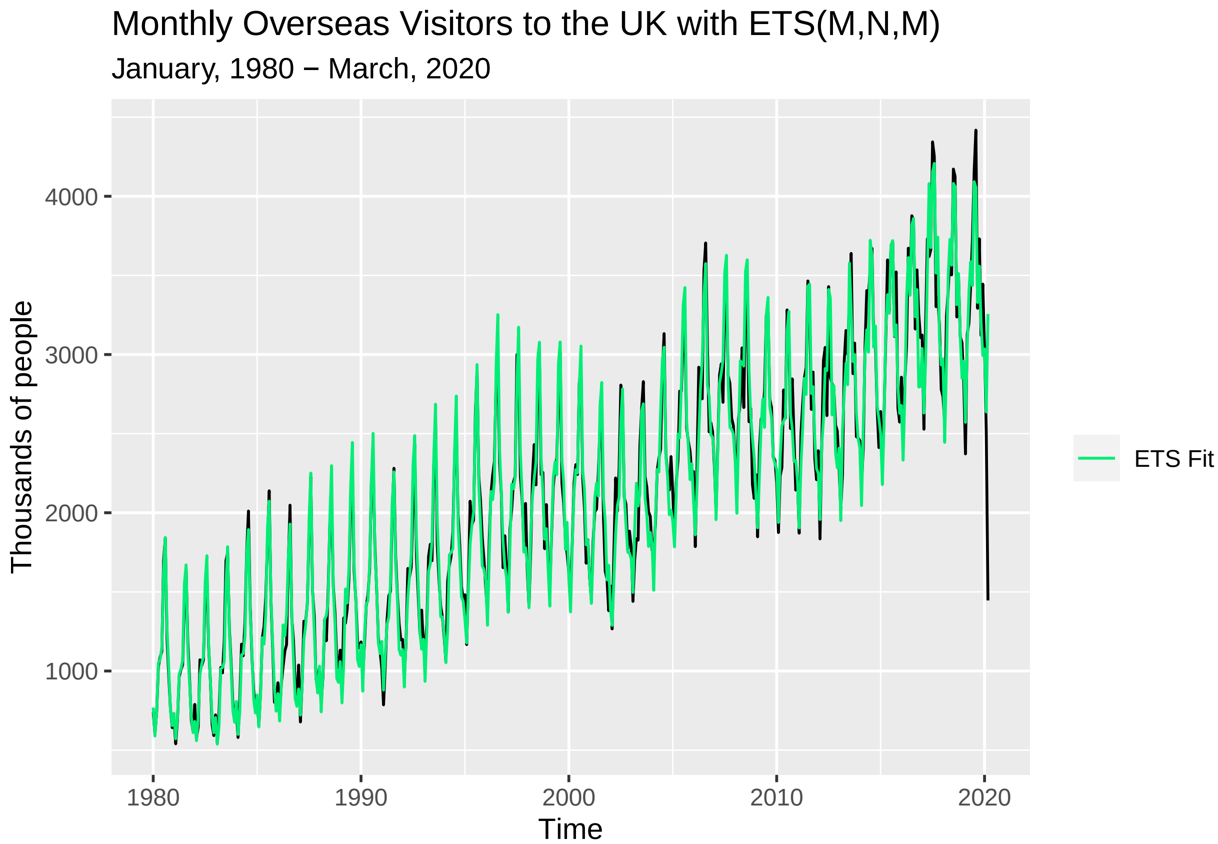
```
m_ets = ets(vis_ts)
summary(m_ets)
```

```
## ETS(M,N,M)
##
## Call:
##  ets(y = vis_ts)
##
```

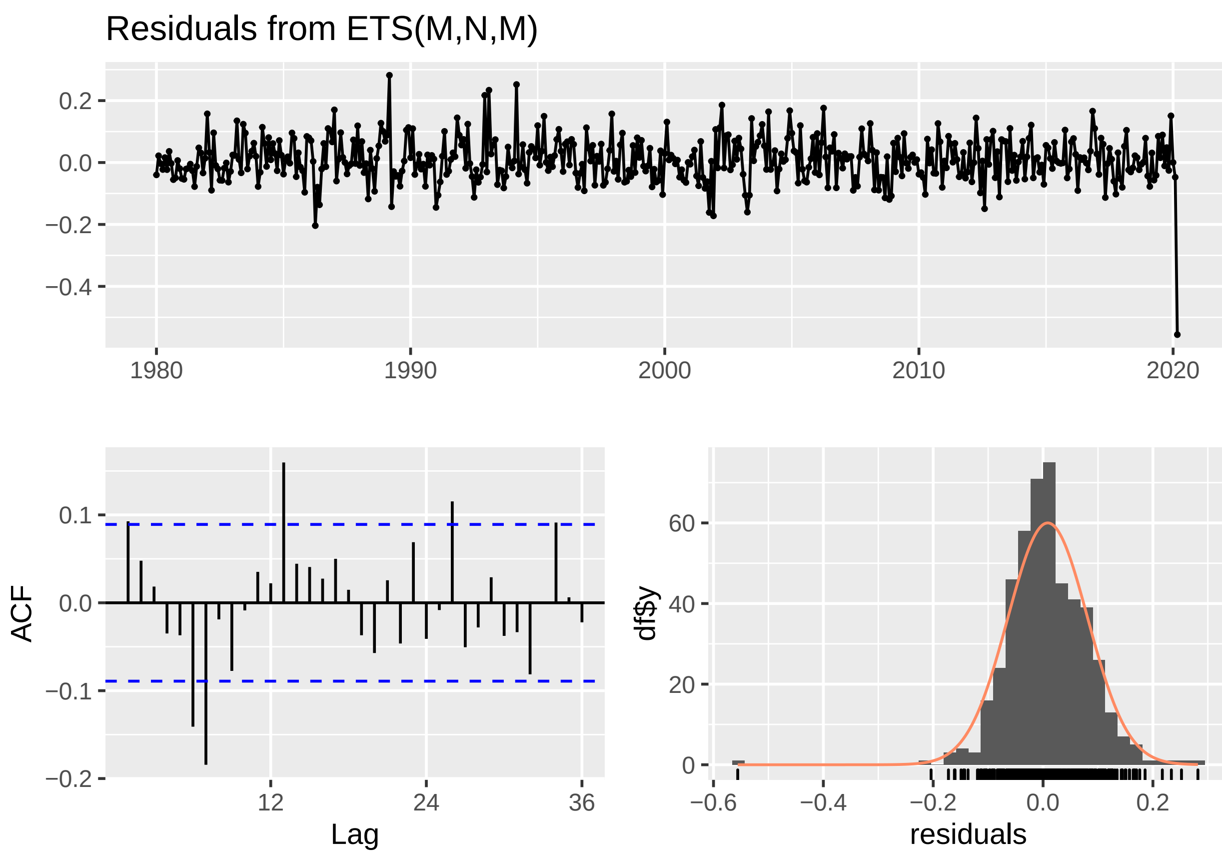
```
##   Smoothing parameters:
##     alpha = 0.2586
##     gamma = 0.34
##
##   Initial states:
##     l = 1056.9641
##     s = 0.6387 0.7327 0.9771 1.2076 1.752 1.5723
##            1.0965 1.0249 1.0007 0.706 0.5631 0.7283
##
##   sigma:  0.0742
##
##      AIC     AICc      BIC
## 7777.468 7778.495 7840.168
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE    MAPE      MASE       ACF1
## Training set 6.996711 163.5554 109.5634 0.2520558 5.43295 0.7307225 0.06226344
```

From the summary above, we can see that the automatic ETS model chosen to fit the data is an ETS model with multiplicative errors and multiplicative seasonality, without the trend component. The AIC value of the estimated model is higher than both our original model and the model generated by `auto.arima`. Therefore, we conclude that the ETS model is not doing a better job than the ARIMA models in fitting the data.

```r
autoplot(vis_ts)+autolayer(fitted.values(m_ets), series = "ETS Fit")+
  labs(x="Time", y="Thousands of people",
       title = "Monthly Overseas Visitors to the UK with ETS(M,N,M)",
       subtitle = "January, 1980 - March, 2020")+
  scale_color_manual("", values = "springgreen2")
```




```
checkresiduals(m_ets)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,N,M)
## Q* = 60.551, df = 10, p-value = 2.851e-09
##
## Model df: 14.   Total lags used: 24
```

Although we have concluded that the ETS model is not fitting the data better than the ARIMA models, it is still doing a really good job at matching the actual series. The ACF plot suggests that there may be some autocorrelation left in the residuals and the histogram shows that the residuals may be slightly positively skewed compared to a normal distribution. Since we reject the null hypothesis that the residuals are normally distributed in the Ljung-Box test, we conclude that the residuals are not white noise for the ETS model.

Holt-Winters Model Fit

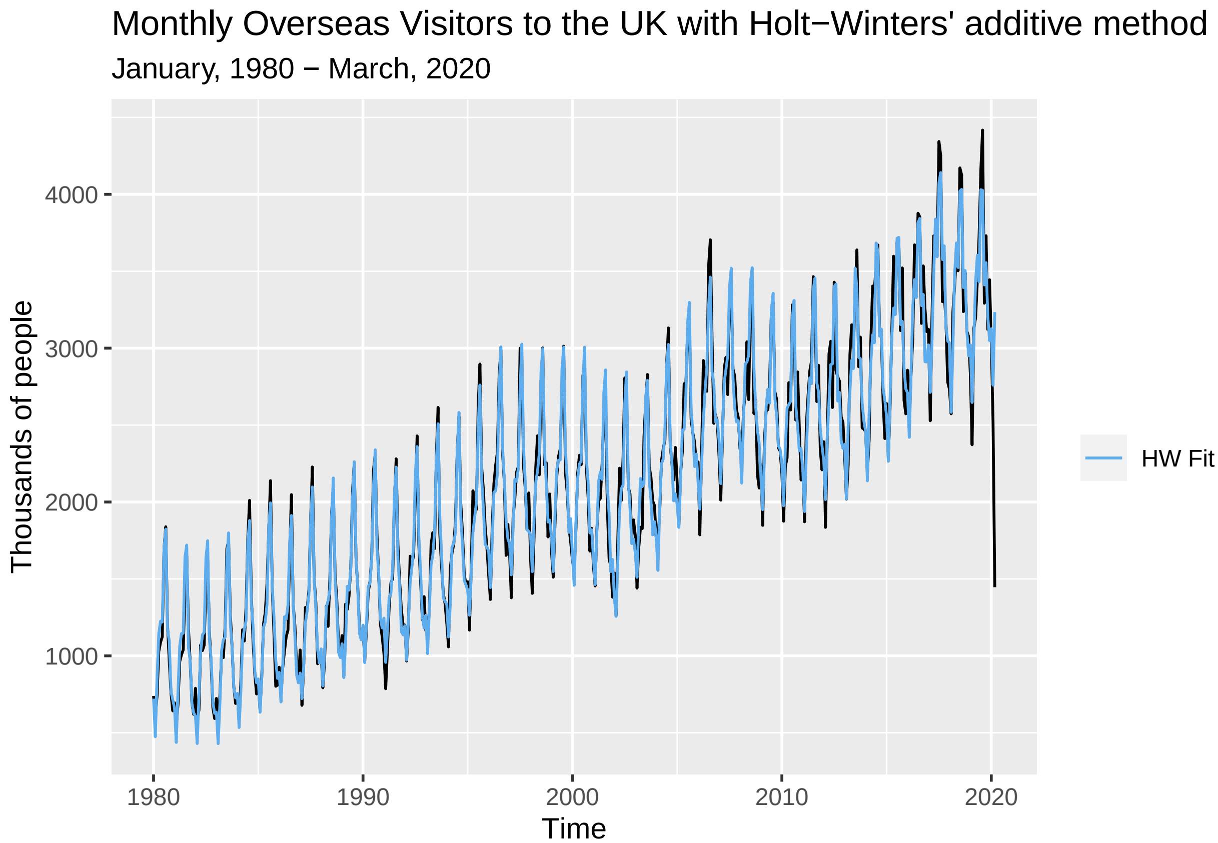
```
m_hw = hw(vis_ts)
summary(m_hw$model)
```

```
## Holt-Winters' additive method
##
## Call:
```

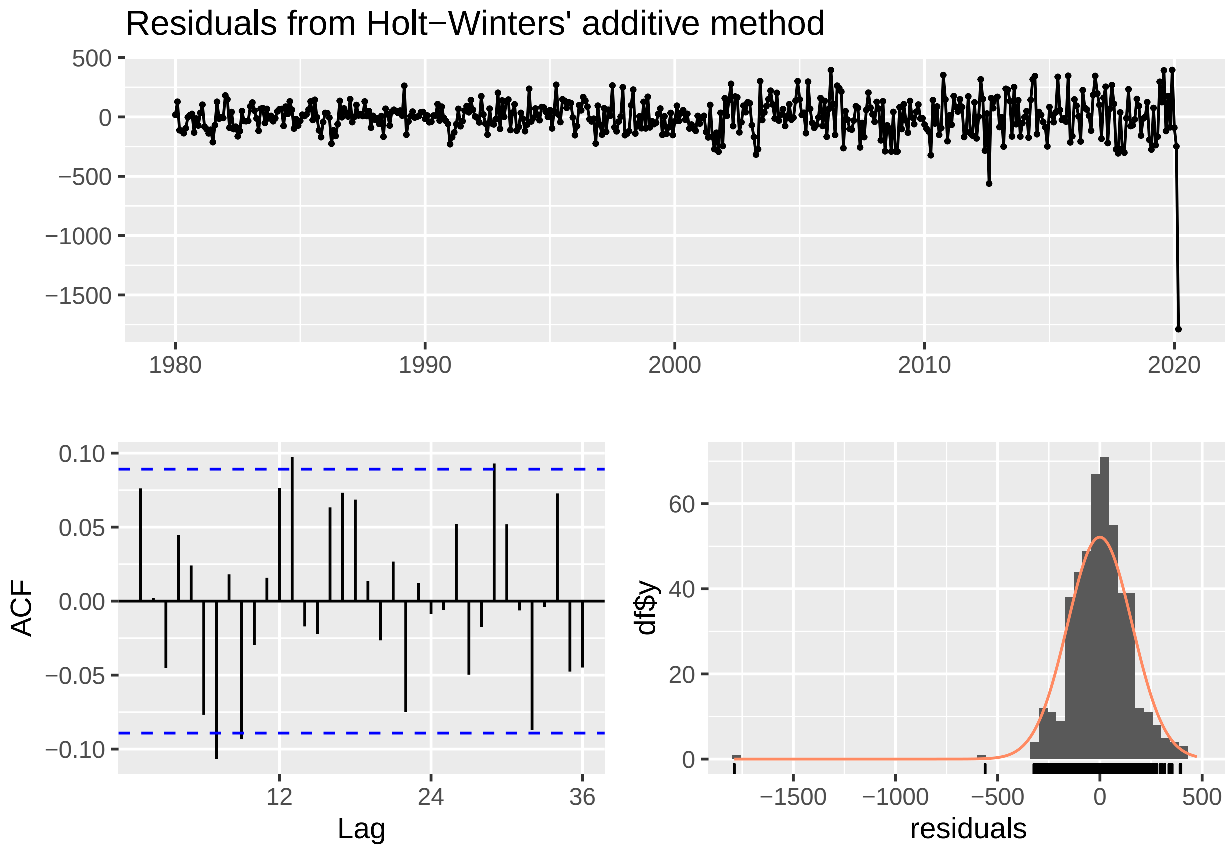
```
##  hw(y = vis_ts)
##
##   Smoothing parameters:
##     alpha = 0.2279
##     beta  = 1e-04
##     gamma = 0.1773
##
##   Initial states:
##     l = 1090.1289
##     b = 4.1509
##     s = -329.1413 -273.6925 26.3737 116.2643 773.2684 654.5027
##            154.5215 133.2232 29.7769 -283.5354 -628.5599 -373.0015
##
##   sigma:  160.7454
##
##      AIC     AICc      BIC
## 7909.785 7911.101 7980.845
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.3974943 158.0605 107.7899 -0.4974107 5.621484 0.7188944
##                    ACF1
## Training set 0.07614338
```

The best fit Holt-Winters model, according to the summary above, is a Holt-Winters' additive method. This model also does not seem to be an any better fit to the data than the ARIMA models, as the AIC value indicates that it does not perform better than the ETS model.

```
autoplot(vis_ts)+autolayer(fitted.values(m_hw), series = "HW Fit")+
  labs(x="Time", y="Thousands of people",
      title = "Monthly Overseas Visitors to the UK with Holt-Winters' additive method",
      subtitle = "January, 1980 - March, 2020")+
  scale_color_manual("", values = "steelblue2")
```




```
checkresiduals(m_hw)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' additive method
## Q* = 37.481, df = 8, p-value = 9.385e-06
##
## Model df: 16.   Total lags used: 24
```

The fit vs. original data plot shows that the Holt-Winters model is still generally performing well in terms of matching the actual data. The residuals plots suggest that the model has mostly captured the dynamics present in the data. Although there are a few small spikes in a couple lags, their magnitudes are small so the residuals are mostly white noise. They are not normally distributed, however, as the Ljung-Box test rejects the null hypothesis at the 1% level.

STL+ETS Model Fit

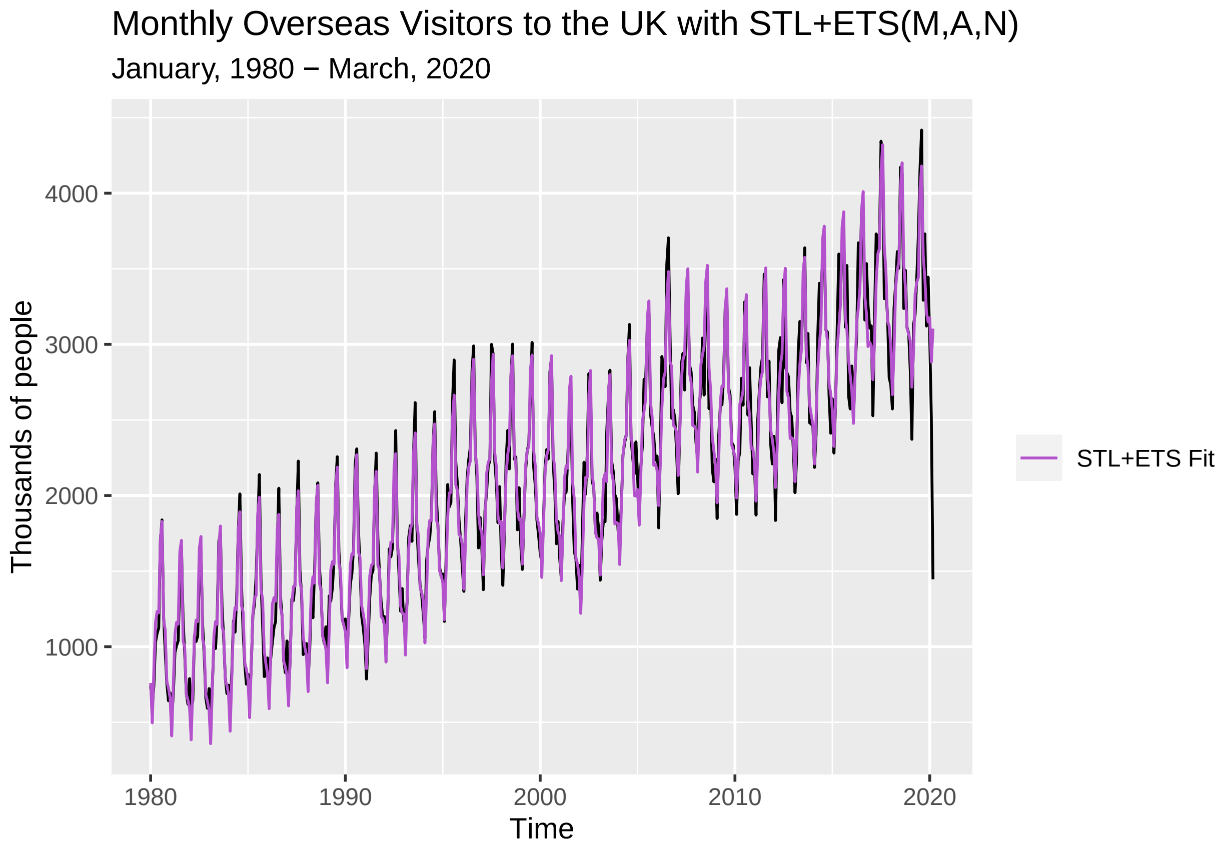
```
m_stl = stlf(vis_ts, s.window = "periodic")
summary(m_stl$model)
```

```
## ETS(M,A,N)
##
## Call:
##  ets(y = na.interp(x), model = etsmodel, allow.multiplicative.trend = allow.multiplicative.trend)
```

```
##
##   Smoothing parameters:
##     alpha = 0.2514
##     beta  = 1e-04
##
##   Initial states:
##     l = 1118.9408
##     b = 4.3981
##
##   sigma:  0.0755
##
##      AIC     AICc      BIC
## 7817.518 7817.644 7838.418
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE     MASE
## Training set -1.857126 161.6889 114.2462 -0.6850784 5.867722 0.761954
##                    ACF1
## Training set 0.09064399
```

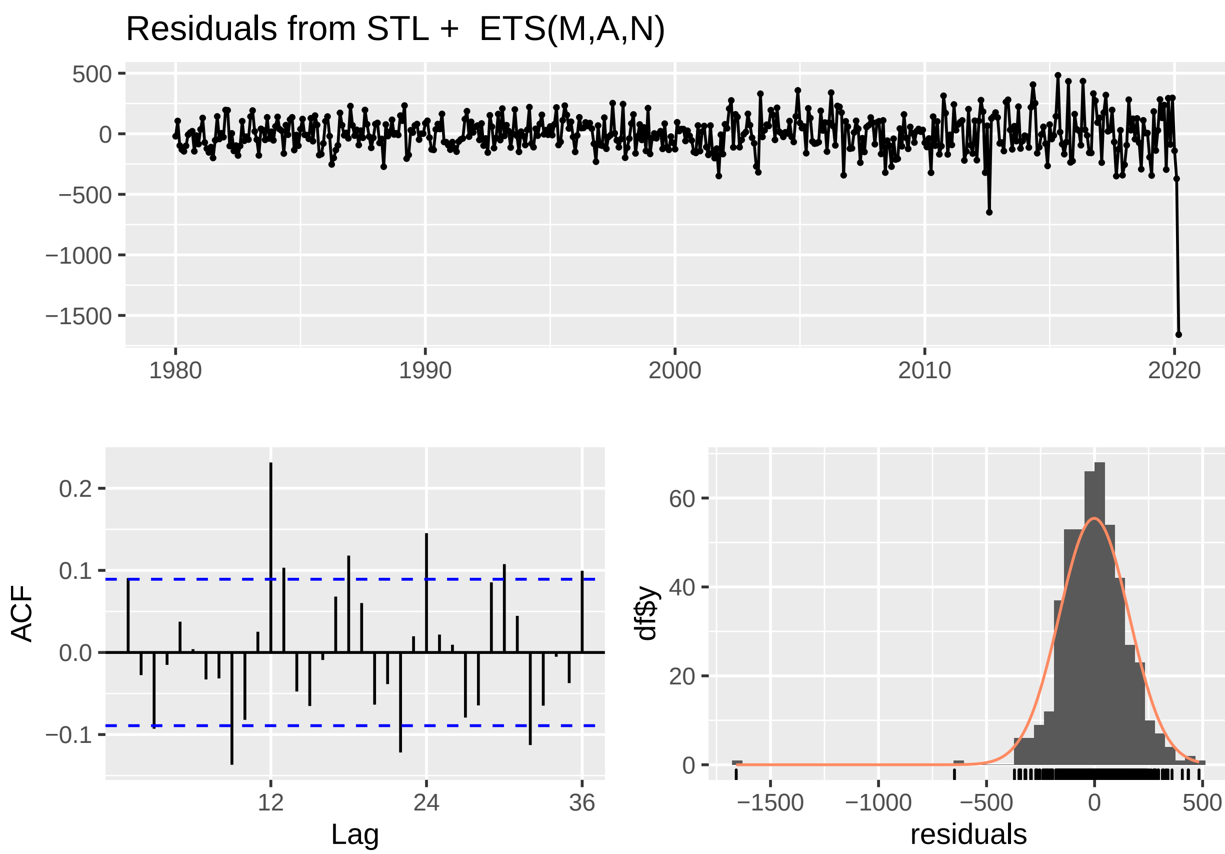
The summary of the STL+ETS model shows that an ETS is being used to model the series, while the seasonal variation is modeled by the STL decomposition. Since we have allowed the seasonal variation to be modeled by STL, the best fit ETS model accompanying is now estimated to be an ETS model with multiplicative errors and additive trend. The estimated coefficient for the smoothing parameter, `beta`, seems to be really close to zero, indicating that there isn't much trend in the data after accounting for seasonal variations. The AIC value of the STL+ETS model seems to be lower than that of the Holt-Winters model. However, it still cannot beat ETS, and therefore, the ARIMA models in fitting the data.

```r
autoplot(vis_ts)+autolayer(fitted.values(m_stl), series = "STL+ETS Fit")+
  labs(x="Time", y="Thousands of people",
       title = "Monthly Overseas Visitors to the UK with STL+ETS(M,A,N)",
       subtitle = "January, 1980 - March, 2020")+
  scale_color_manual("", values = "mediumorchid3")
```




```
checkresiduals(m_stl)
```

```
## Warning in checkresiduals(m_stl): The fitted degrees of freedom is based on the
## model used for the seasonally adjusted data.
```



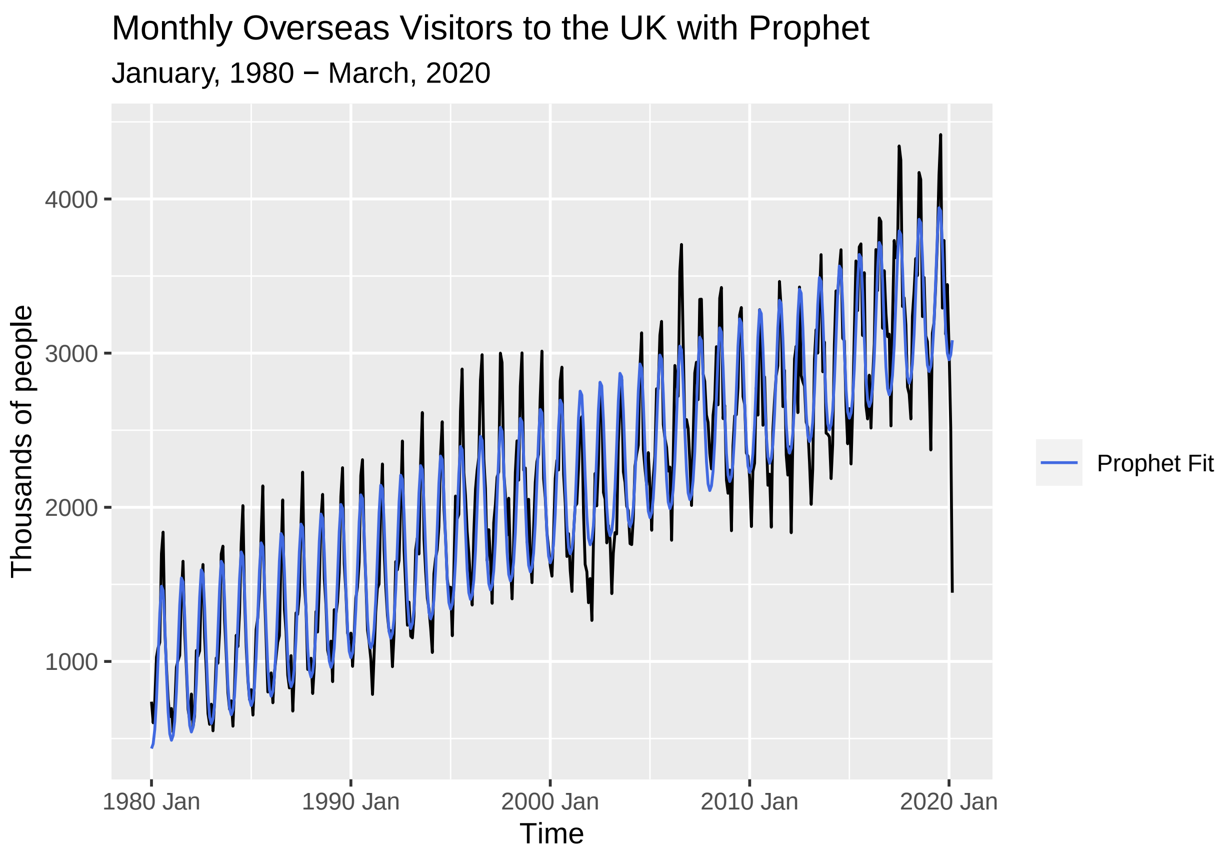

```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ETS(M,A,N)
## Q* = 91.056, df = 20, p-value = 4.842e-11
##
## Model df: 4.   Total lags used: 24
```

The STL+ETS model is also performing well in terms of matching the actual values. The residual plots, however, indicate that there are some strong dynamics left unaccounted for by the model. First of all, the time plot of the residuals shows that the variation in the residuals increases over time, implying heteroskedasticity. Furthermore, the ACF of the residuals shows some strong autocorrelation, indicating that they are not white noise. Finally, we reject the null hypothesis that the residuals are normally distributed in the Ljung-Box test.

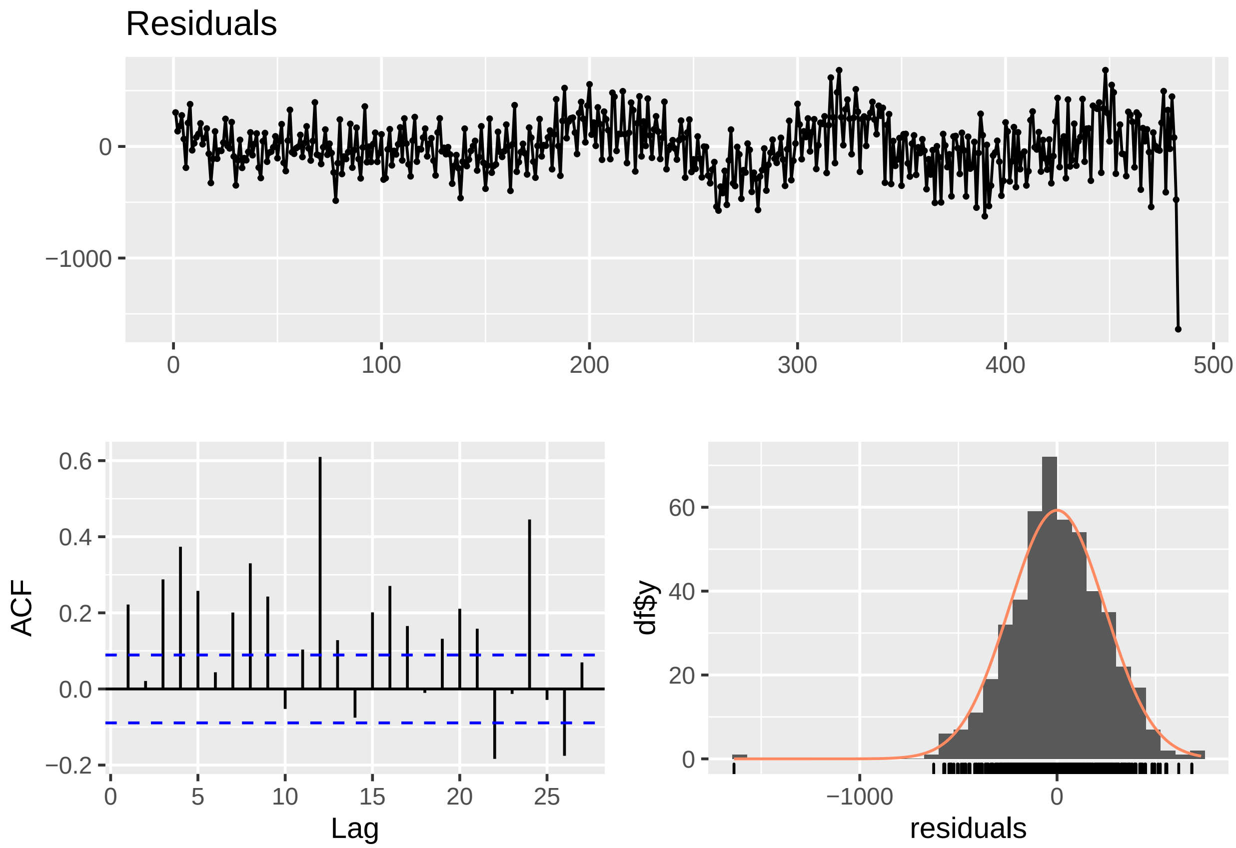
Prophet Model Fit

```
m_p = model(vis,
            "Prophet" = prophet(Visitors~season(period = 12, order=2,
                                                 type = "additive")))
fitv = fitted.values(m_p)
ggplot(vis, aes(x=Date, y=Visitors)) + geom_line()+
  geom_line(aes(x=Date, y=fitv$.fitted, col = "Prophet Fit"))+
```

```
  scale_color_manual("", labels = "Prophet Fit", values = "royalblue")+
  labs(x="Time", y="Thousands of people",
       title = "Monthly Overseas Visitors to the UK with Prophet",
       subtitle = "January, 1980 - March, 2020")
```



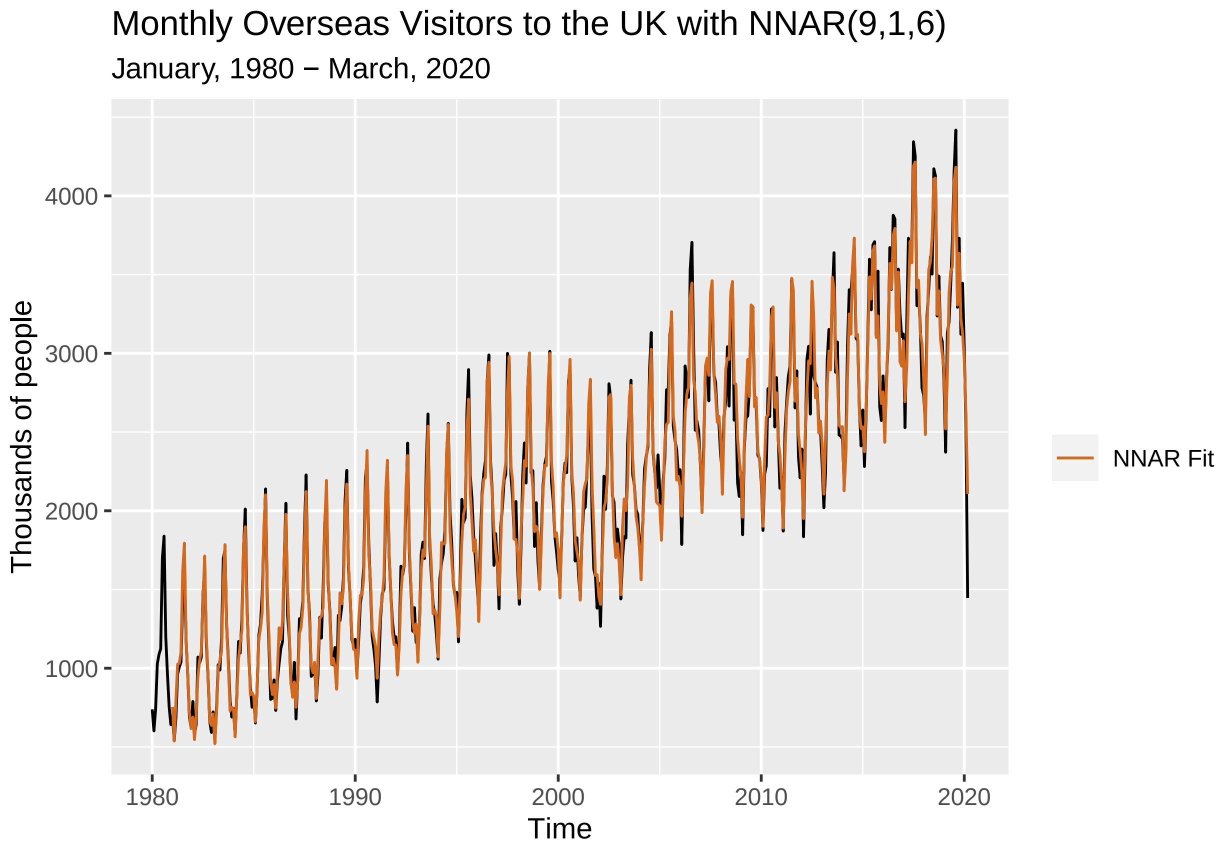

```
res_p = resid(m_p)
checkresiduals(res_p$.resid)
```



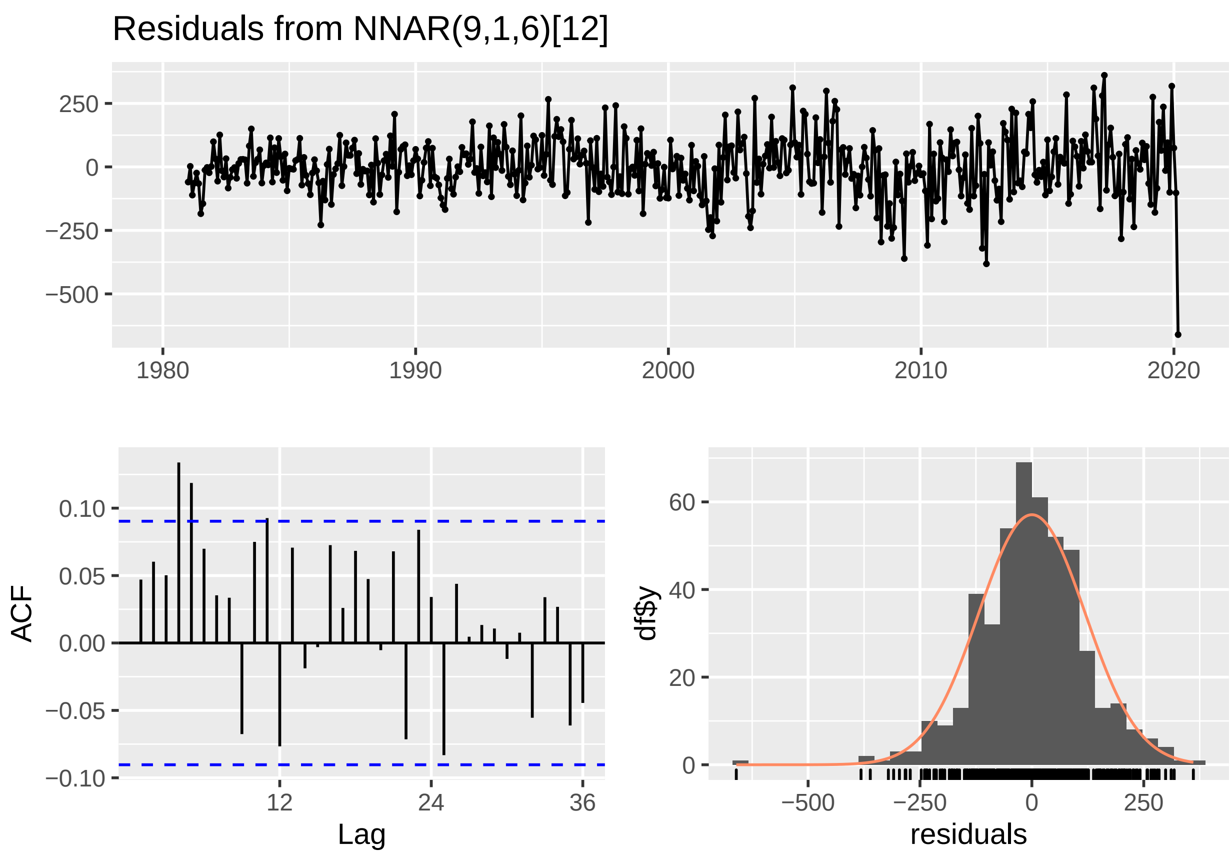
The Prophet model seems to be doing the worse out of all of the models in fitting the data. The seasonal variation estimated by the model seems to be constant and cannot account for the changes in amplitude over time. The time plot of the residuals clearly shows some variation with time, suggesting that our model failed to account for all of the dynamics in the data. The ACF of the residuals displays strong autocorrelation, confirming that they are not white noise.

NNAR Model Fit

```
m_nnetar = nnetar(vis_ts)
autoplot(vis_ts)+autolayer(fitted.values(m_nnetar), series = "NNAR Fit")+
  labs(x="Time", y="Thousands of people",
       title = "Monthly Overseas Visitors to the UK with NNAR(9,1,6)",
       subtitle = "January, 1980 - March, 2020")+
  scale_color_manual("", values = "chocolate")
```




```
checkresiduals(m_nnetar)
```

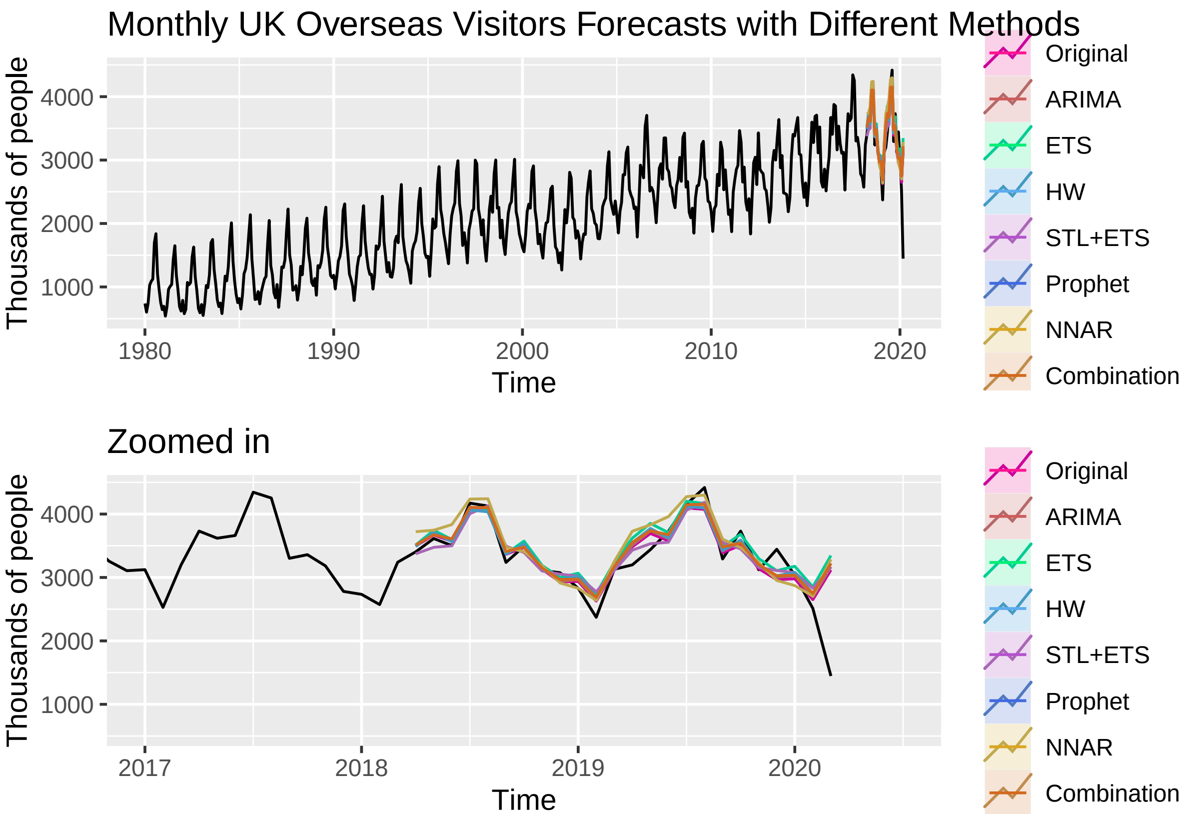


The NNAR model is also doing a good job at fitting the data, and has even managed to capture the sudden downturn in the number of overseas visitors in March, 2020. The time plot of the residuals, however, seems to show some clear dynamics. Although, the ACF plot shows some spikes, the magnitudes may be too small to disregard. We conclude that the residuals are clearly not white noise, as they are heteroskedastic.

Comparing forecasts with a training set that ends in March, 2018

```
train = window(vis_ts, end = c(2018,3))
frcog = forecast::forecast(train, h=24, model = m_og)
frcarima = forecast::forecast(auto.arima(train), h=24)
frcets = forecast::forecast(train, h = 24)
frchw = hw(train, h=24)
frcstl = stlf(train, s.window = "periodic", h=24)
train2 = vis[1:459,]
pfit = model(train2,
            "Prophet" = prophet(Visitors~season(period = 12, order=2,
                                                  type = "additive")))

frcp = forecast(pfit, h=24)
frcnnetar = forecast::forecast(nnetar(train), h=24)
combination = (frcog[["mean"]] + frcarima[["mean"]] + frcets[["mean"]] + frchw[["mean"]] +
                 frcstl[["mean"]]+frcnnetar[["mean"]])/6

colors = c("Original" = "deeppink",
```

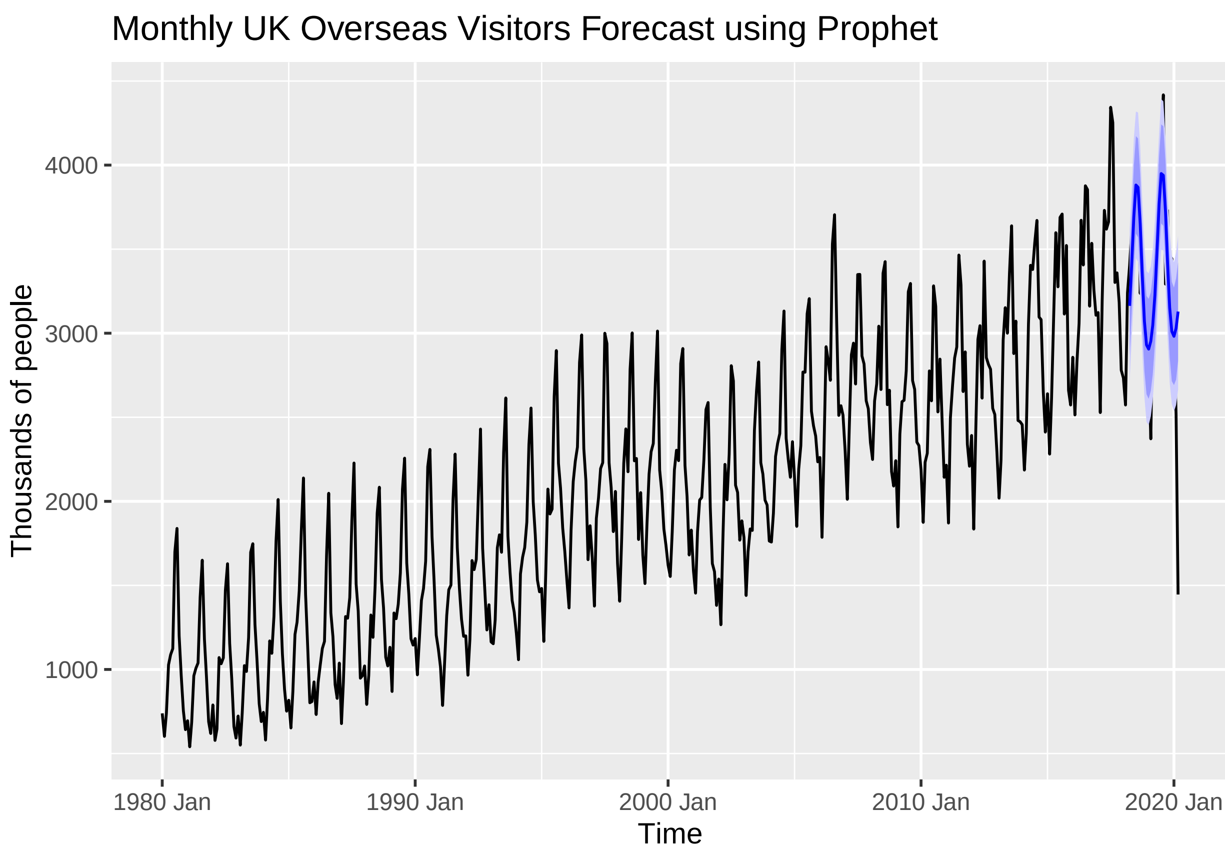
```r
            "ARIMA" = "indianred",
            "ETS" = "springgreen2",
            "HW" = "steelblue2",
            "STL+ETS" = "mediumorchid3",
            "Prophet" = "royalblue",
            "NNAR" = "goldenrod",
            "Combination" = "chocolate")
frc_p1 = autoplot(vis_ts) +
  autolayer(frcog, PI=FALSE, series = "Original")+
  autolayer(frcarima, PI = FALSE, series = "ARIMA") +
  autolayer(frcets, PI = FALSE, series = "ETS")+
  autolayer(frcstl, PI = FALSE, series = "STL+ETS")+
  autolayer(frchw, PI = FALSE, series = "HW")+
  autolayer(frcnnetar, PI = F, series = "NNAR")+
  autolayer(combination, series = "Combination")+
  labs(x = "Time", y = "Thousands of people",
       title = "Monthly UK Overseas Visitors Forecasts with Different Methods",
       color = "")+
  scale_color_manual(values = colors)

frc_p2 = autoplot(vis_ts) +
  autolayer(frcog, PI=FALSE, series = "Original")+
  autolayer(frcarima, PI = FALSE, series = "ARIMA") +
  autolayer(frcets, PI = FALSE, series = "ETS")+
  autolayer(frcstl, PI = FALSE, series = "STL+ETS")+
  autolayer(frchw, PI = FALSE, series = "HW")+
  autolayer(frcnnetar, PI = F, series = "NNAR")+
  autolayer(combination, series = "Combination")+
  labs(x = "Time", y = "Thousands of people", title = "Zoomed in",
       color = "")+
  scale_color_manual(values = colors)+
  coord_cartesian(xlim = c(2017, 2020.5))

frc_p1/frc_p2
```



From the plots above, we can see that all models are performing well in forecasting the data. In fact, the models are so close to each other and the actual data, that it is hard to visually determine which one is performing better than the others. Unsurprisingly, all of the models have failed to capture the significant drop in the number of overseas visitors to the UK during March, 2020.

We will also forecast the Prophet model on a seperate graph below:

```
ggplot(vis, aes(x=Date, y=Visitors)) + geom_line()+
  autolayer(frcp)+
  labs(x="Time", y="Thousands of people",
       title = "Monthly UK Overseas Visitors Forecast using Prophet")
```



The Prophet model does not seem to be performing as good as the rest of the forecasts. Even though the actual values are within the 80% prediction interval, the fitted values poorly match the amplitude of the actual series. Since we had already established that the Prophet model was doing a poor job at fitting the data, these results are not surprising.

To determine which one of the models is actually performing the best in terms of fitting and forecasting the data, we need to look at the forecast accuracy measures of each.

Check the forecast accuracy measures

```
acc = rbind(
  accuracy(frcog, vis_ts)["Test set",1:7],
  accuracy(frcarima, vis_ts)["Test set",1:7],
  accuracy(frcets, vis_ts)["Test set",1:7],
  accuracy(frchw, vis_ts)["Test set",1:7],
  accuracy(frcstl, vis_ts)["Test set",1:7],
  accuracy(frcp, vis)[,3:10],
  accuracy(frcnnetar, vis_ts)["Test set",1:7],
  accuracy(combination, vis_ts)["Test set",1:7])
rownames(acc) = c("Original","ARIMA", "ETS", "HW", "STL+ETS", "Prophet","NNAR",
                  "Combination")
kable(acc, digits = 4, caption = "Forecast Accuracy for Monthly UK Overseas Visitors") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 1: Forecast Accuracy for Monthly UK Overseas Visitors

	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
Original	-66.5218	383.8677	199.2412	-5.1296	8.7238	1.3651	0.0747	-66.5218
ARIMA	-91.7210	392.5076	202.3405	-5.9789	8.9517	1.3863	0.0765	-91.7210
ETS	-170.5956	439.6481	244.3611	-8.7509	10.7063	1.6742	0.1190	-170.5956
HW	-108.6052	407.9990	219.1751	-6.6638	9.5914	1.5016	0.1020	-108.6052
STL+ETS	-81.6465	398.3288	214.8103	-5.8917	9.4366	1.4717	0.1237	-81.6465
Prophet	-47.3277	442.7116	288.6806	-5.0899	11.5422	1.9778	2.3284	0.1001
NNAR	-171.6190	447.9411	278.1377	-8.3321	11.4109	1.9056	0.0849	-171.6190
Combination	-115.1182	404.6980	210.3542	-6.7912	9.3597	0.0868	0.9650	-115.1182

To determine which model actually performs best, we will be looking at the Root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE) of the forecasts. According to the table above the lowest RMSE and MAPE belongs to the forecast using our original model for Monthly Overseas Visitors to the UK, which was the ARIMA(1,0,1)(2,1,1) model. This is surprising because the ARIMA model generated by the `auto.arima` function had a lower AIC score, thus indicating that it is the better model. The AIC values were not that far apart, however, and the AIC score for the `auto.arima` model could be slightly lower because there are fewer parameters being estimated.

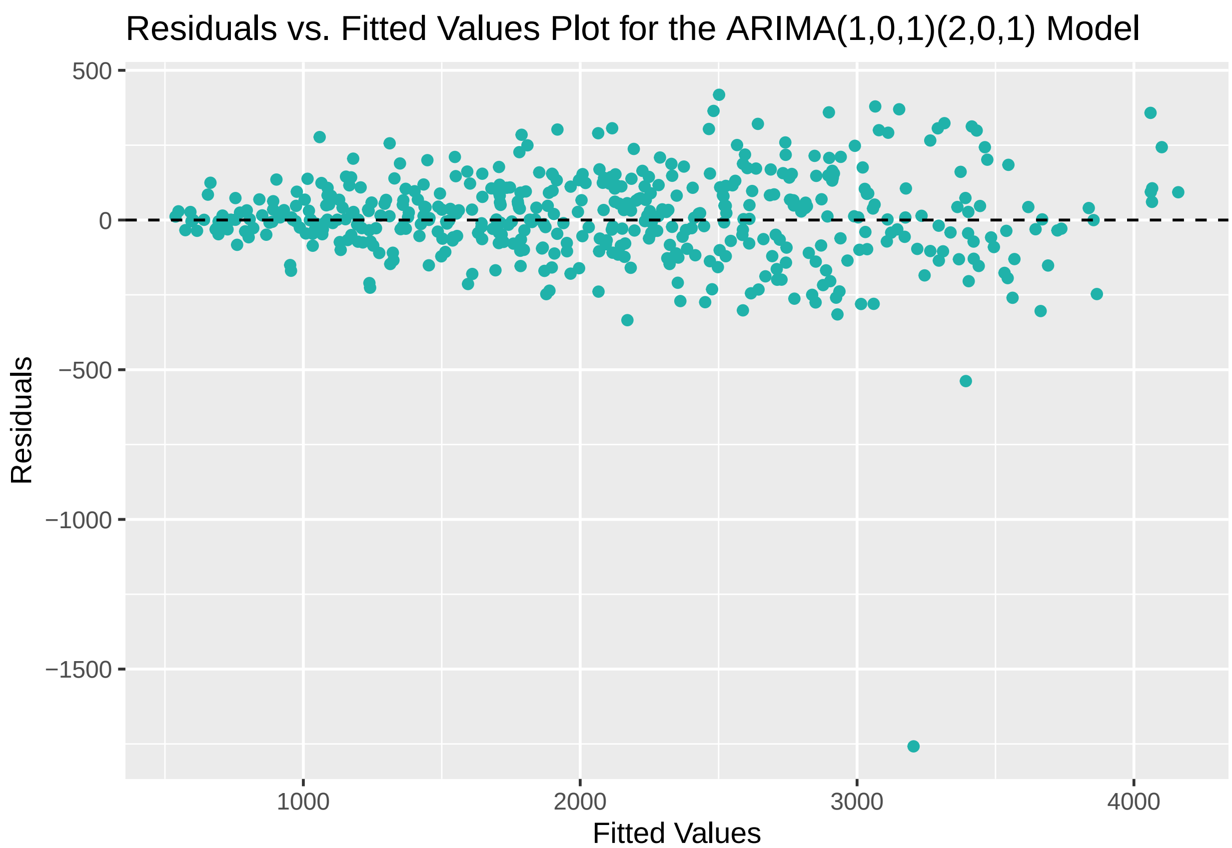
The worst RMSE and MAPE scores belong to the Prophet model, confirming our claims that it was doing a poor job in fitting and forecasting the data. Although the Combination forecast did not ultimately beat our original model, it still has lower RMSE and MAPE scores than 5 of the 8 models estimated.

Since the RMSE and MAPE values determine that the original model is the best forecast model, we will continue with a residual analysis for that model.

Residual analysis of the best fit model

Residuals vs. fitted values plot

```
res = residuals(m_og)
fitv = fitted.values(m_og)
df = data.frame(x = fitv, y = res)
ggplot(df, aes(x = x, y = y)) + geom_point(col = "lightseagreen")+
  geom_hline(yintercept = 0, lty = 2)+
  labs(x = "Fitted Values", y = "Residuals",
       title = "Residuals vs. Fitted Values Plot for the ARIMA(1,0,1)(2,0,1) Model")
```

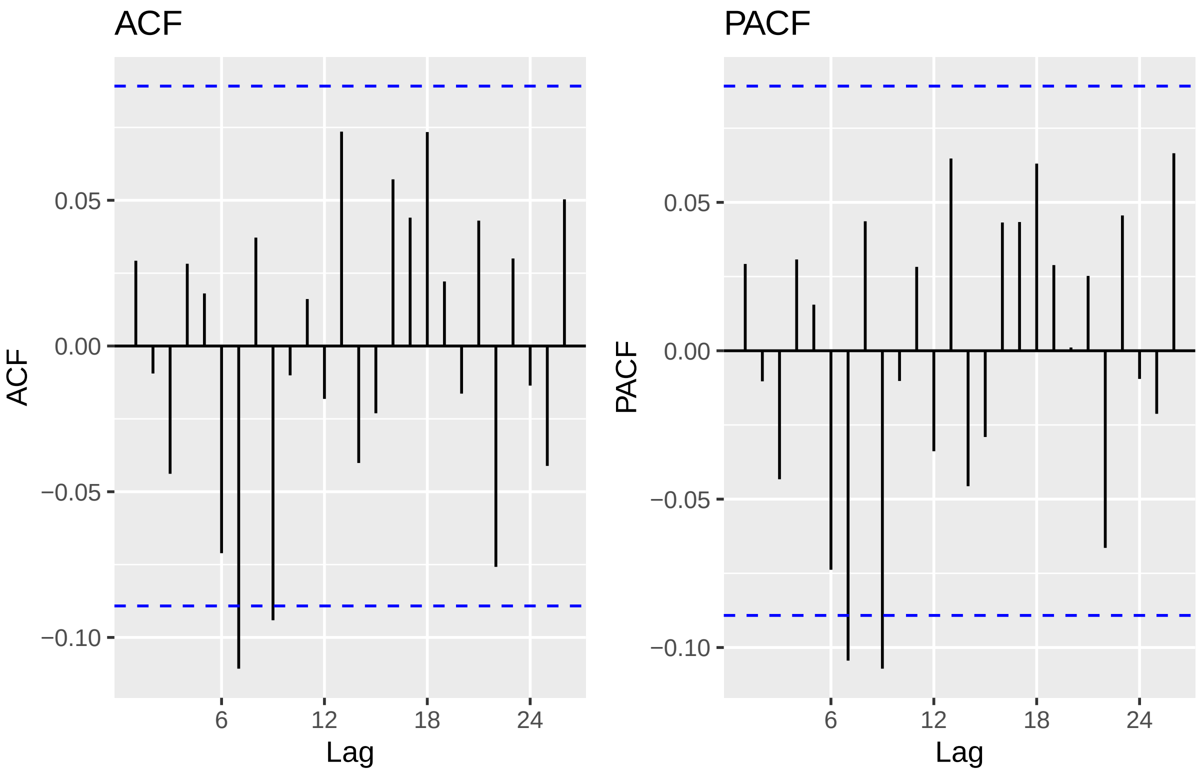


The residuals vs. fitted values plot shows that the residuals are mostly distributed randomly around zero, though there does appear to be some outliers that affect the variance of the residuals. Specifically, an outlier in the residuals around 2012 or 2013 seems to be significantly affecting the residuals. To determine whether the residuals are actually white noise or not, we will be looking at the ACF and PACF plots.

ACF and PACF of residuals

```
acf_r_p = ggAcf(res)+ggtitle("ACF")
pacf_r_p = ggAcf(res, type = "partial")+ggtitle("PACF")
acf_r_p+pacf_r_p+plot_annotation(title = "Residual ACF and PACF Plots")
```

Residual ACF and PACF Plots

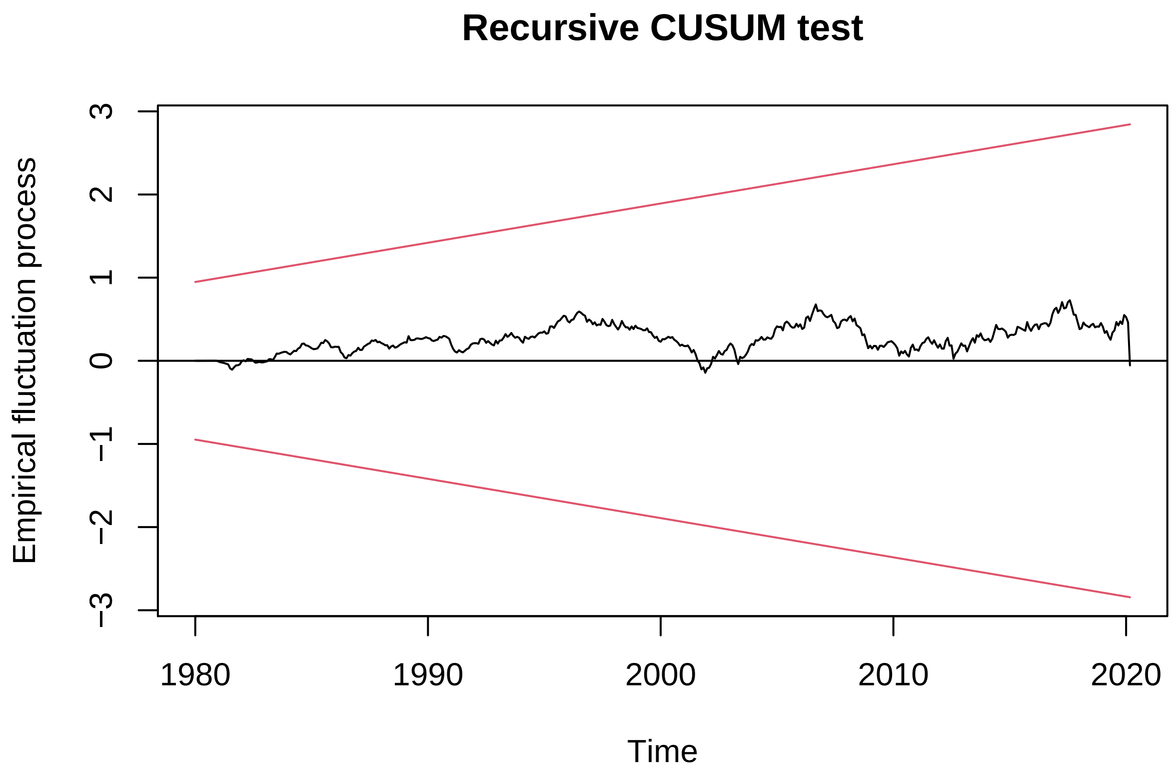


The ACF and PACF plots show that the residuals seem to be close to white noise, with there being almost no significant lags that cross the threshold. Although there seems to be a couple significant lags in both the ACF and the PACF, since the magnitude of these spikes are low, we will treat them as negligible.

After establishing that our model is performing well in fitting and forecasting the data, we need to make sure that the residuals are stable over time. The next step in our analysis will looking at the recursive CUSUM plot for any structural breaks in the data.

CUSUM plot

```
cusum = efp(res ~ 1, type = "Rec-CUSUM")
plot(cusum)
```

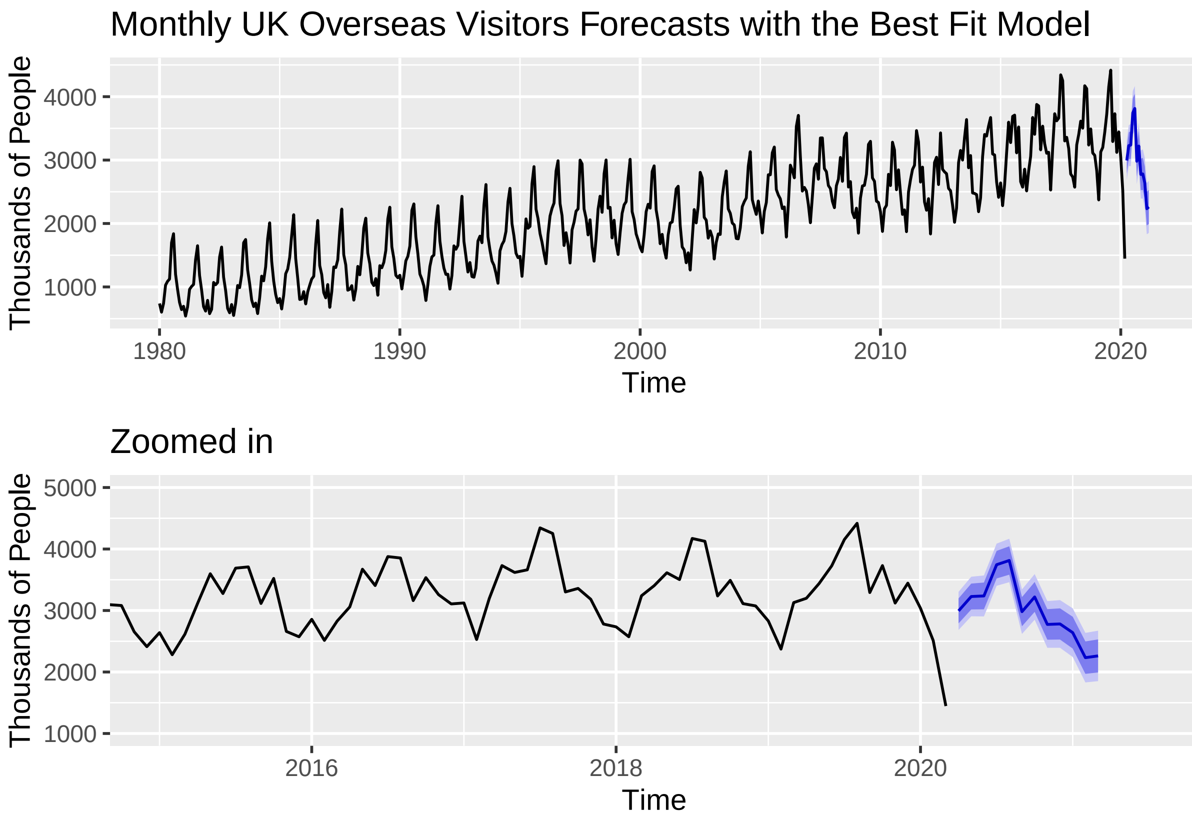


The Recursive CUSUM test plot shows that our residuals are stable over time with no structural breaks in the series, which implies that our model is doing a good job at explaining the dynamics in the series.

Now that we have concluded our analysis of the best fit model residuals, we can go ahead an forecast the 12-steps-ahead number of monthly overseas visitors to the UK.

12-steps-ahead forecasts into the future with the best model

```
f_frc = forecast::forecast(m_og, h=12)
f_frc_p1 = autoplot(f_frc)+
  labs(x="Time", y="Thousands of People",
       title = "Monthly UK Overseas Visitors Forecasts with the Best Fit Model")
f_frc_p2 = autoplot(f_frc)+
  labs(x="Time", y="Thousands of People",
       title = "Zoomed in")+
  coord_cartesian(xlim = c(2015,2021.5), ylim = c(1000,5000))

f_frc_p1/f_frc_p2
```



Our forecast indicates that the number of monthly overseas visitors to the UK will rise back up again after its sudden drop in March, 2020. We know this is not true, however, as lockdowns continued well into 2021 for some countries and most people refrained from travel during the pandemic.

Conclusions and Future Work

Our analysis concludes that the best model to forecast the Monthly Overseas Visitors to the UK between January, 1980 and March,2020 is an ARIMA(1,0,1)(2,1,1) model, with its forecast producing the smallest RMSE and MAPE values. This, however, is not the best model according to the AIC selection criterion, which has designated the ARIMA(1,0,1)(0,1,2) model as the best one. All of the other models we have considered have performed fairly well in fitting and forecasting the data, except for the Prophet model, which has failed to capture most of the dynamics present in the data.

Future work should focus on addressing some of the limitations of this report. Our biggest limitation was that our data only spanned until March, 2020. This was because the International Passenger Survey was suspended during the COVID pandemic. This, of course, also means that any future data will have to have a gap during the pandemic, causing a structural break in the series. The analysis could also be replicated for tourism data from other countries to see whether the conclusions hold for them as well. Other forecast methods for comparison could also be tested.

References

Data retrieved from https://www.ons.gov.uk/peoplepopulationandcommunity/leisureandtourism/timeseries/gmaa/ott

https://www.visitbritain.org/value-tourism-england#:~:text=Taking%20into%20account%20direct%20and,billion%2C%20supporting%201.4%20million%20jobs