Time Series Lab 4: Global Temp & Air Passengers

Billy Dang

2024-11-25

PSTAT 174/274 Fall 2024 – Lab Assignment 4

Part 1: Modeling Global Temperature

We are working with the gtemp_ocean data from the astsa package. The data is a measure of global mean ocean temperature deviations from 1850-2023. Our aim is to fit an appropriate ARIMA(p, d, q) model and to evaluate our model's performance using residual diagnostics before constructing a 20 year ahead forecast.

1. Using the following code to load the data and produce a time series plot. Is the data stationary? Comment on any possible linear or seasonal trends and how best to remove them

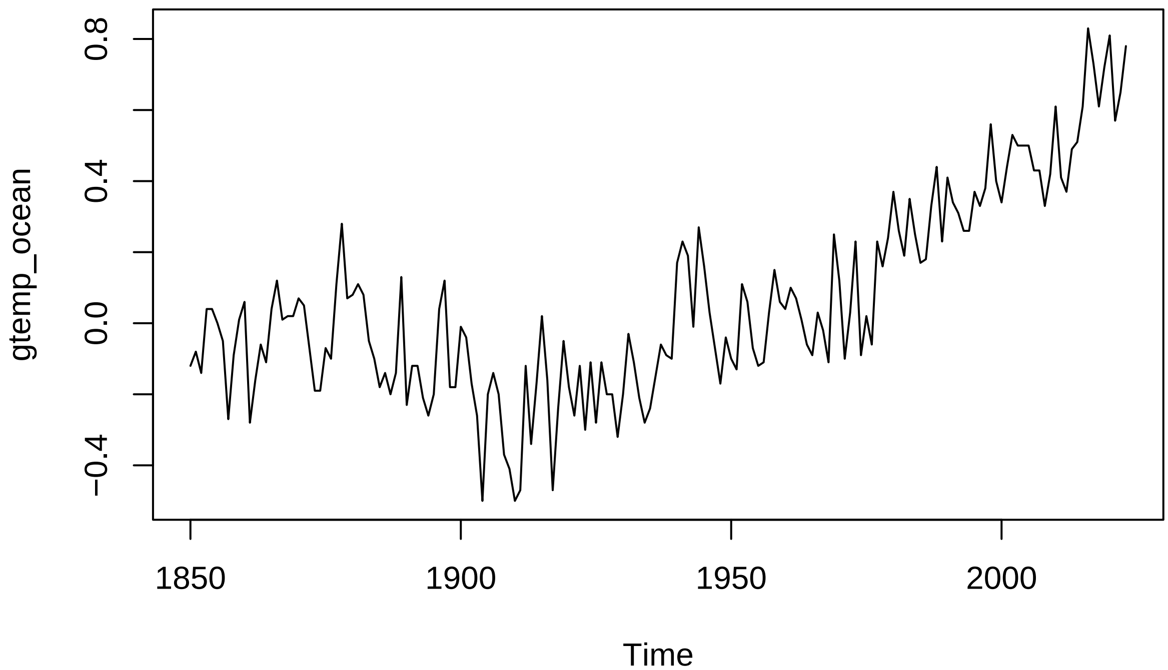
```
library(astsa)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:astsa':
##
##     gas
```

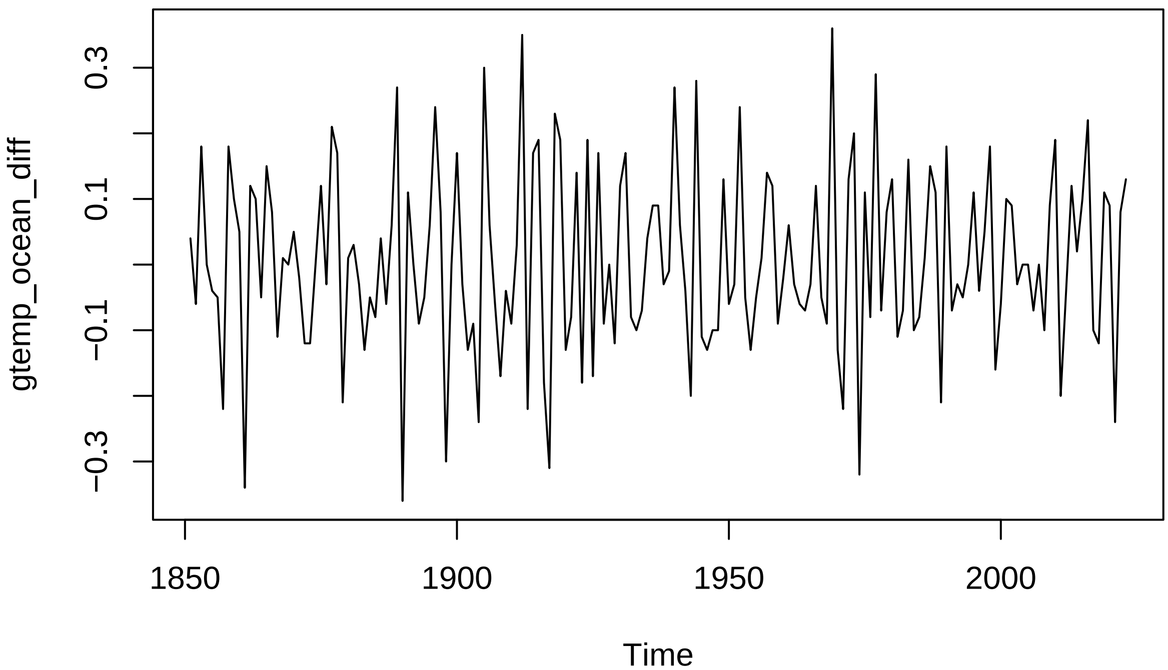
```
plot.ts(gtemp_ocean)
```



After plotting a time series graph of gtemp_ocean, we see an upward trend as time increases, suggesting that the data is in fact not stationary. We can difference the data in order to make the data stationary.

2. Difference the data lag 1 and produce a new time series plot. Does the data now appear stationary?

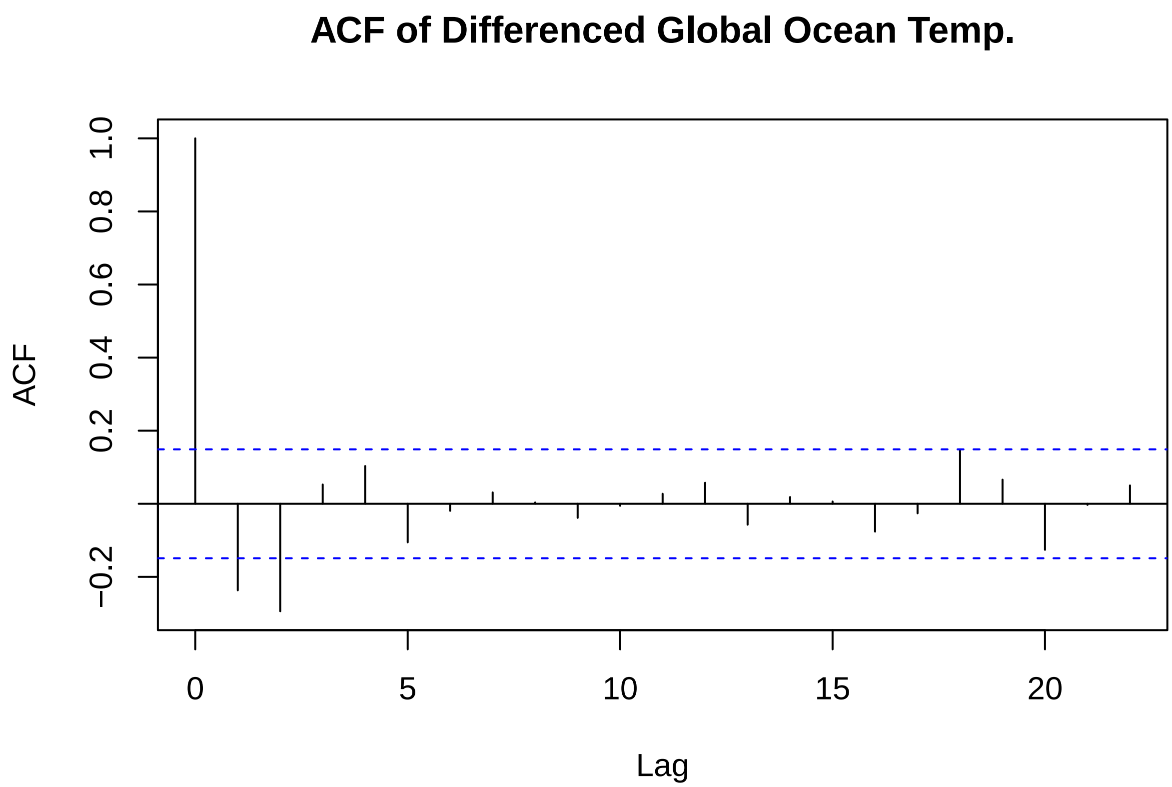
```
gtemp_ocean_diff <- diff(gtemp_ocean)

plot.ts(gtemp_ocean_diff)
```



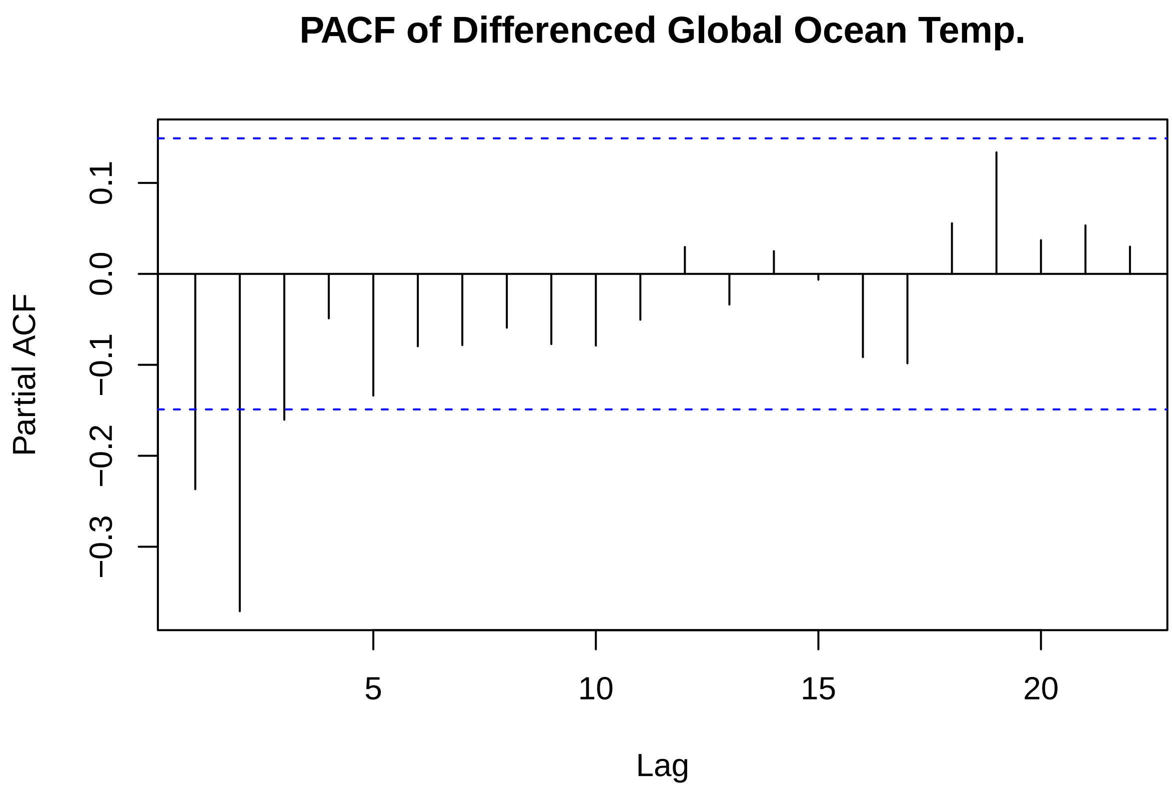
After using diff() to difference the data, it appears stationary as there is no clear trend, a lack of obvious seasonality, and seemingly constant mean and variance.

3. Produce both an ACF and a PACF of the differenced data and comment on your observations. What potential ARIMA(p, d, q) models do they suggest? (Hint: Recall that the d parameter just indicates how many times we needed to differencing to obtain a stationary model.)

```
acf(gtemp_ocean_diff, main = "ACF of Differenced Global Ocean Temp.")
```




```r
pacf(gtemp_ocean_diff,main = "PACF of Differenced Global Ocean Temp.")
```

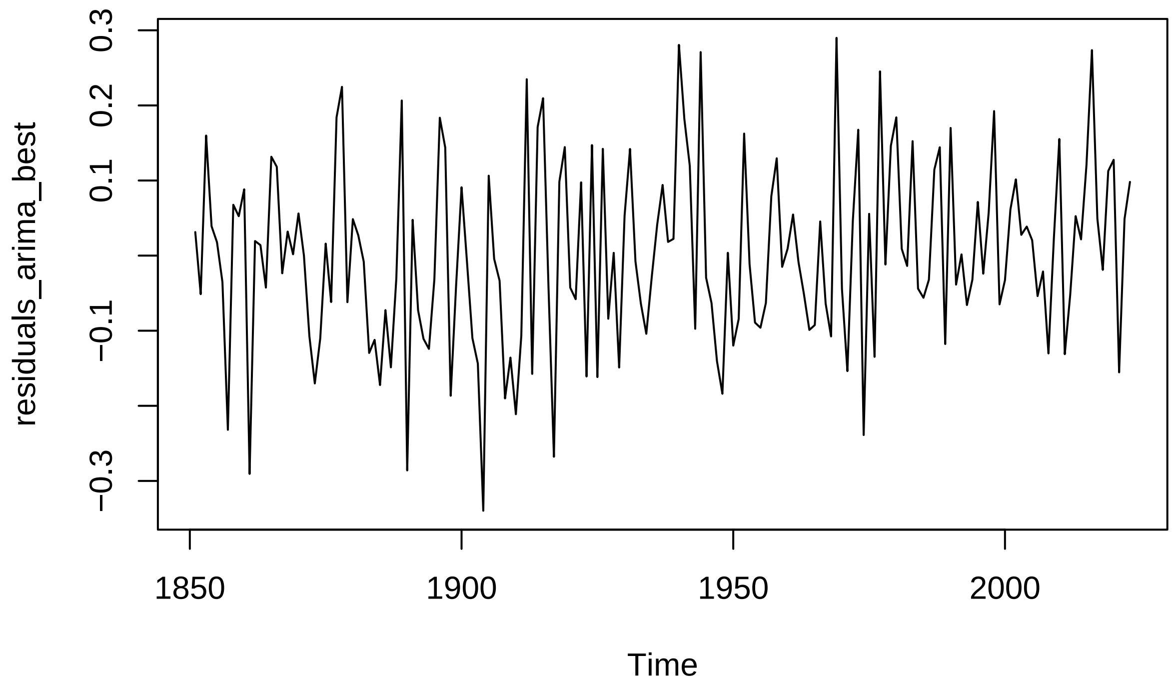


Given the significant lags of 1 and 2 within our ACF plot, and significant lags of 1, 2, and 3 in our PACF plot, the data suggests the implementation of an MA(2) model (as PACF is exponentially decaying to 0 and there are 2 sig. spikes in ACF). We'll plot this model below:

```
#AR(0) & MA(3)
arima_best <- auto.arima(gtemp_ocean_diff)
arima_best
```

```
## Series: gtemp_ocean_diff
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##           ma1      ma2    mean
##       -0.4256  -0.3093  0.0045
## s.e.   0.0728   0.0715  0.0025
##
## sigma^2 = 0.01516:  log likelihood = 118.03
## AIC=-228.06   AICc=-227.82   BIC=-215.45
```

```
residuals_arima_best <- residuals(arima_best)
plot.ts(residuals_arima_best)
```




```r
#testing that there is no autocorrelation remaining in residuals after fitting model
Box.test(residuals(arima_best),type = "Ljung-Box")
```

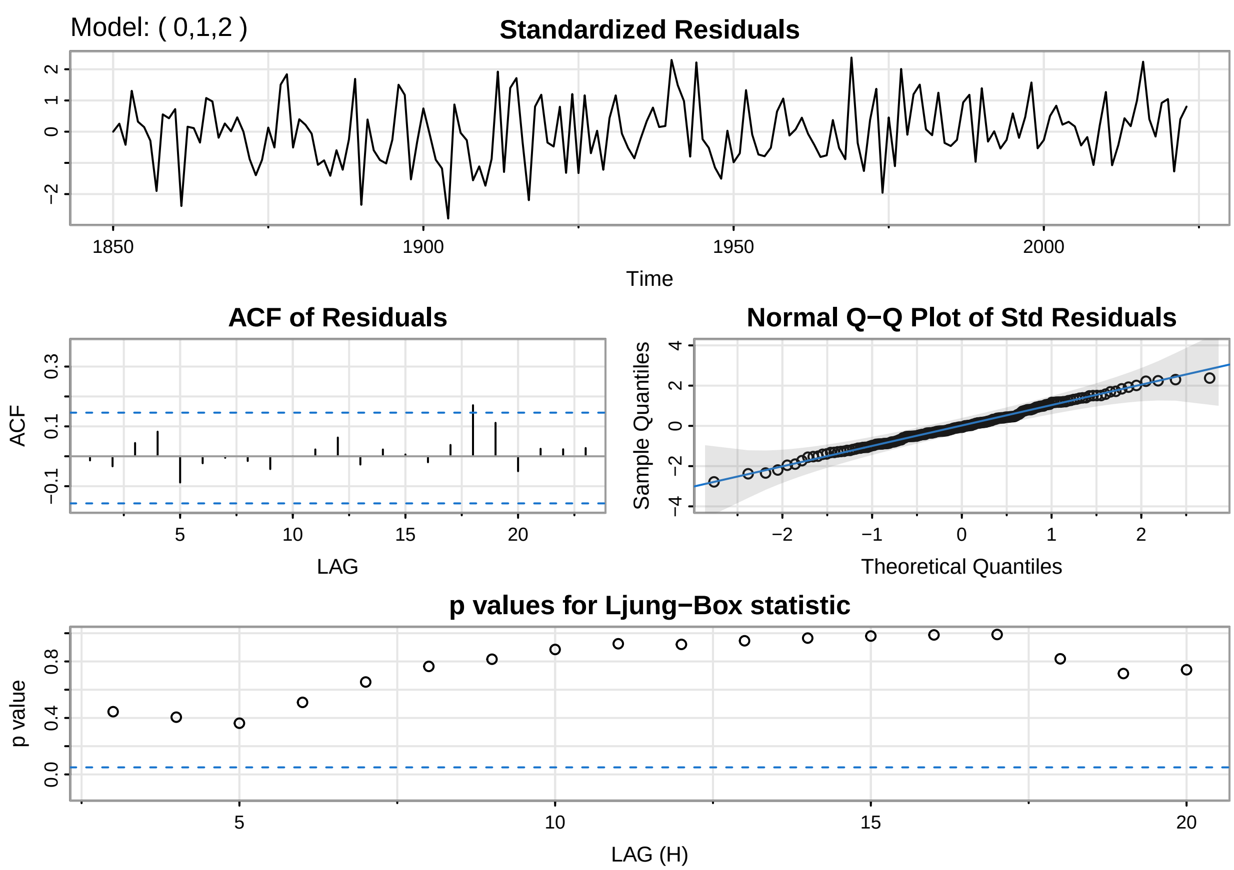
```
##
##  Box-Ljung test
##
## data:  residuals(arima_best)
## X-squared = 0.032911, df = 1, p-value = 0.856
```

4. This time use the sarima() function from the astsa package to fit your selected ARIMA model.

```r
sarima_model <- sarima(gtemp_ocean, p = 0, d = 1, q = 2)
```

```
## initial  value -1.970301
## iter   2 value -2.078881
## iter   3 value -2.097771
## iter   4 value -2.100283
## iter   5 value -2.101226
## iter   6 value -2.102299
## iter   7 value -2.102967
## iter   8 value -2.102974
## iter   9 value -2.102974
## iter  10 value -2.102974
## iter  11 value -2.102974
```

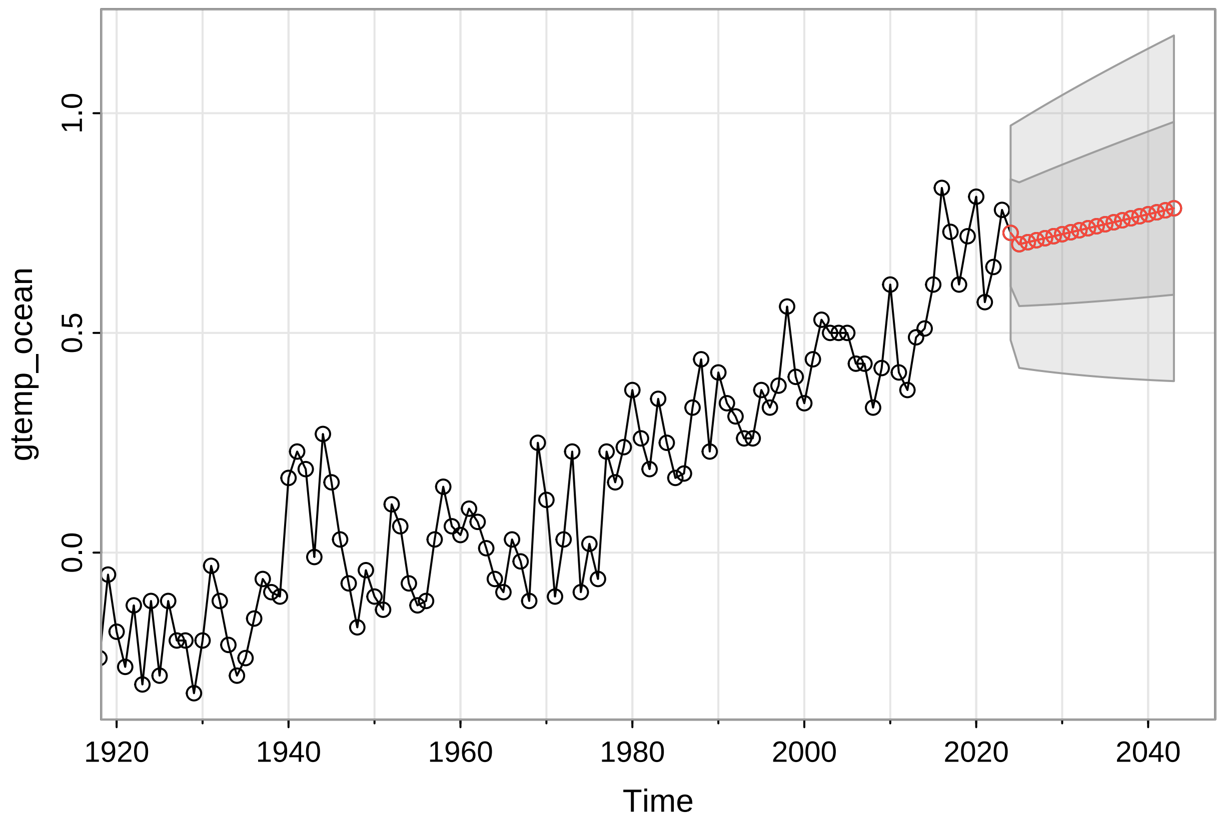
```
## iter  11 value -2.102974
## iter  11 value -2.102974
## final  value -2.102974
## converged
## initial  value -2.101175
## iter   2 value -2.101193
## iter   3 value -2.101198
## iter   4 value -2.101198
## iter   4 value -2.101198
## iter   4 value -2.101198
## final  value -2.101198
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##          Estimate     SE t.value p.value
## ma1       -0.4256 0.0728 -5.8472  0.0000
## ma2       -0.3093 0.0715 -4.3233  0.0000
## constant   0.0045 0.0025  1.8006  0.0735
##
## sigma^2 estimated as 0.01490112 on 170 degrees of freedom
##
## AIC = -1.318277  AICc = -1.317456  BIC = -1.245368
##
```



5. Using our model, produce a 20 year ahead forecast for global temperature using sarima.for() function

from the astsa package. Comment on your forecast.

```r
#producing a 20-year forecast
sarima.for(gtemp_ocean, n.ahead = 20, p = 0, d = 1, q = 2)
```



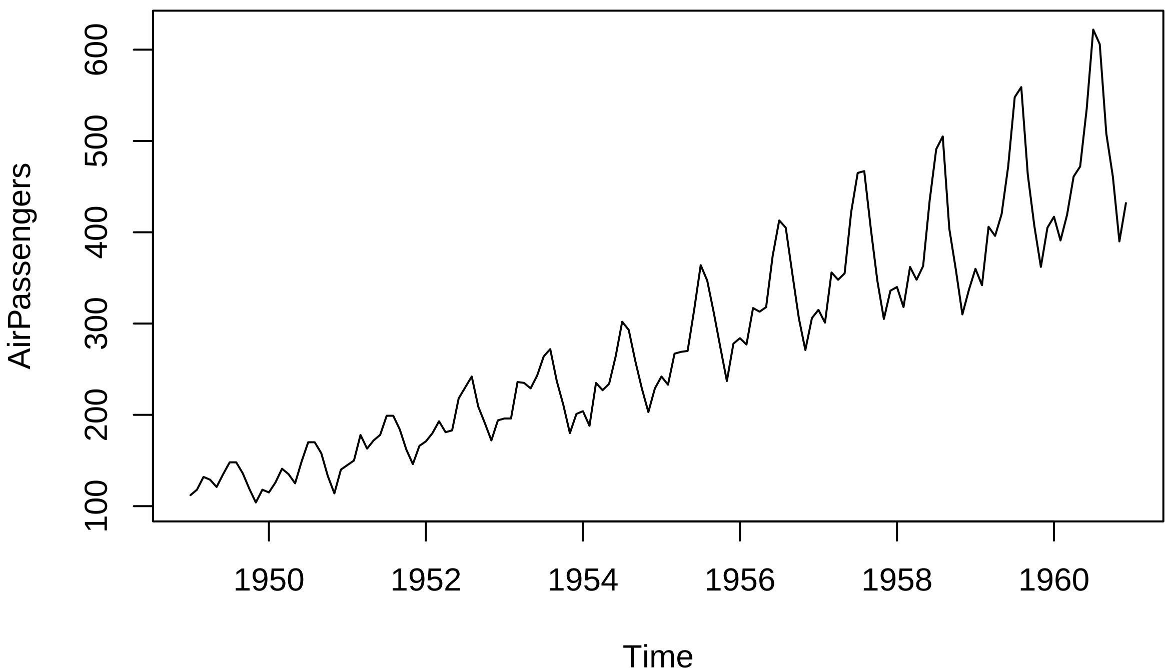

```
## $pred
## Time Series:
## Start = 2024
## End = 2043
## Frequency = 1
##  [1] 0.7276907 0.7018755 0.7064156 0.7109557 0.7154958 0.7200359 0.7245760
##  [8] 0.7291161 0.7336562 0.7381963 0.7427364 0.7472765 0.7518166 0.7563567
## [15] 0.7608968 0.7654369 0.7699770 0.7745171 0.7790572 0.7835973
##
## $se
## Time Series:
## Start = 2024
## End = 2043
## Frequency = 1
##  [1] 0.1220702 0.1407774 0.1444503 0.1480320 0.1515291 0.1549473 0.1582917
##  [8] 0.1615669 0.1647770 0.1679257 0.1710165 0.1740524 0.1770363 0.1799706
## [15] 0.1828579 0.1857004 0.1884999 0.1912585 0.1939779 0.1966596
```

The forecase predicts a rather linear upward trend in gtemp_ocean. The model is rather linear due to the MA(2) component that captures rahter short-term dependencies, smoothing out fluctuations in the forecast.

Part 2: Modeling Airline Passengers For this part we'll analyze the AirPassengers dataset from the astsa package. The dataset contains monly totals of international airline passengers from 1949 to 1960 Our aim is to determine an appropriate model from the SARIMA family.

1. Begin by producing a time series plot of the data using the plot.ts() function. Note your observations about any trends, seasonality and stationarity (is the variance constant?).

```
plot.ts(AirPassengers)
```



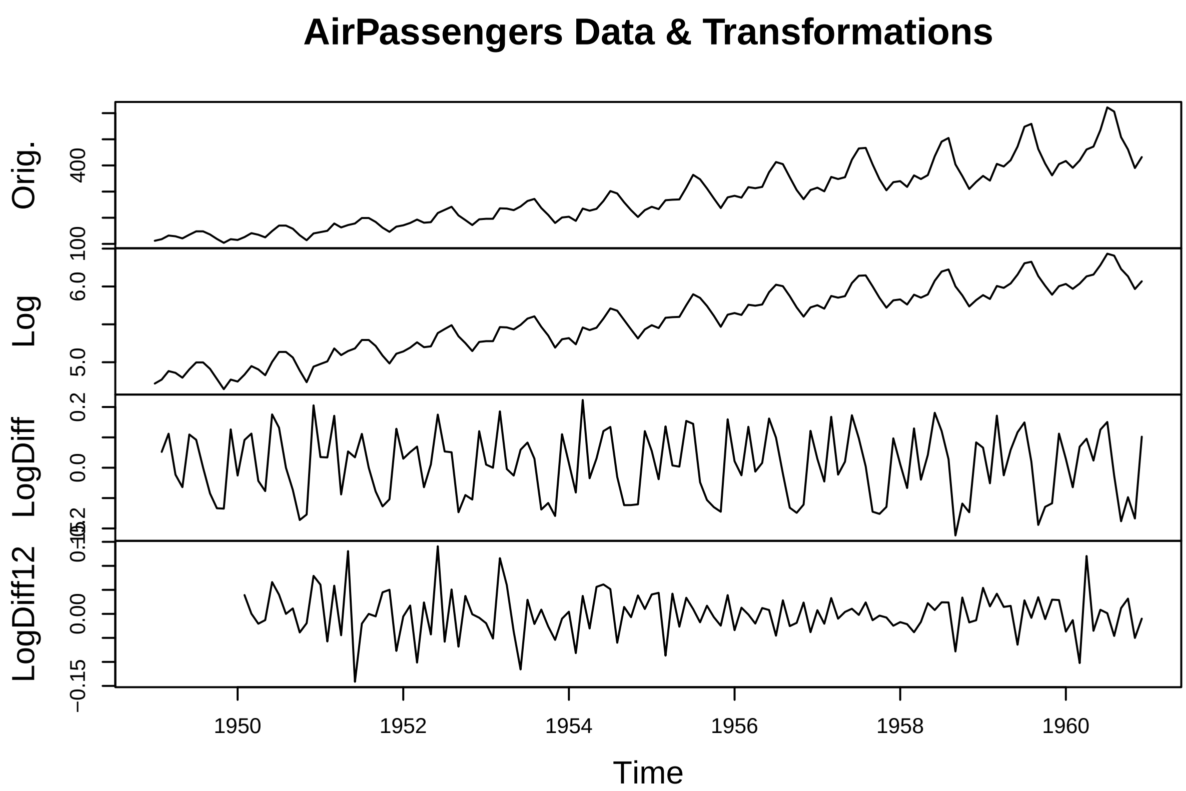
After plotting the time series data, we see that the data is not stationary as there is evidence of seasonality (consistent swings within the data), an upward trend, as well as inconsistent variance as the size of those swings seem to be growing over time.

2. We investigate possible transformations to obtain a stationary time series on which we can consider potential SARIMA models. Use the following codes to compute 3 transformed time series:

(1) log.data taking the natural log of the data;
(2) dlog.data taking the difference (lag 1) of the log data; and
(3) ddlog.data taking the difference (lag 12) of the differenced log data. Create a matrix of all four series called plot.data using the cbind() function and use plot.ts() to produce our combined plot

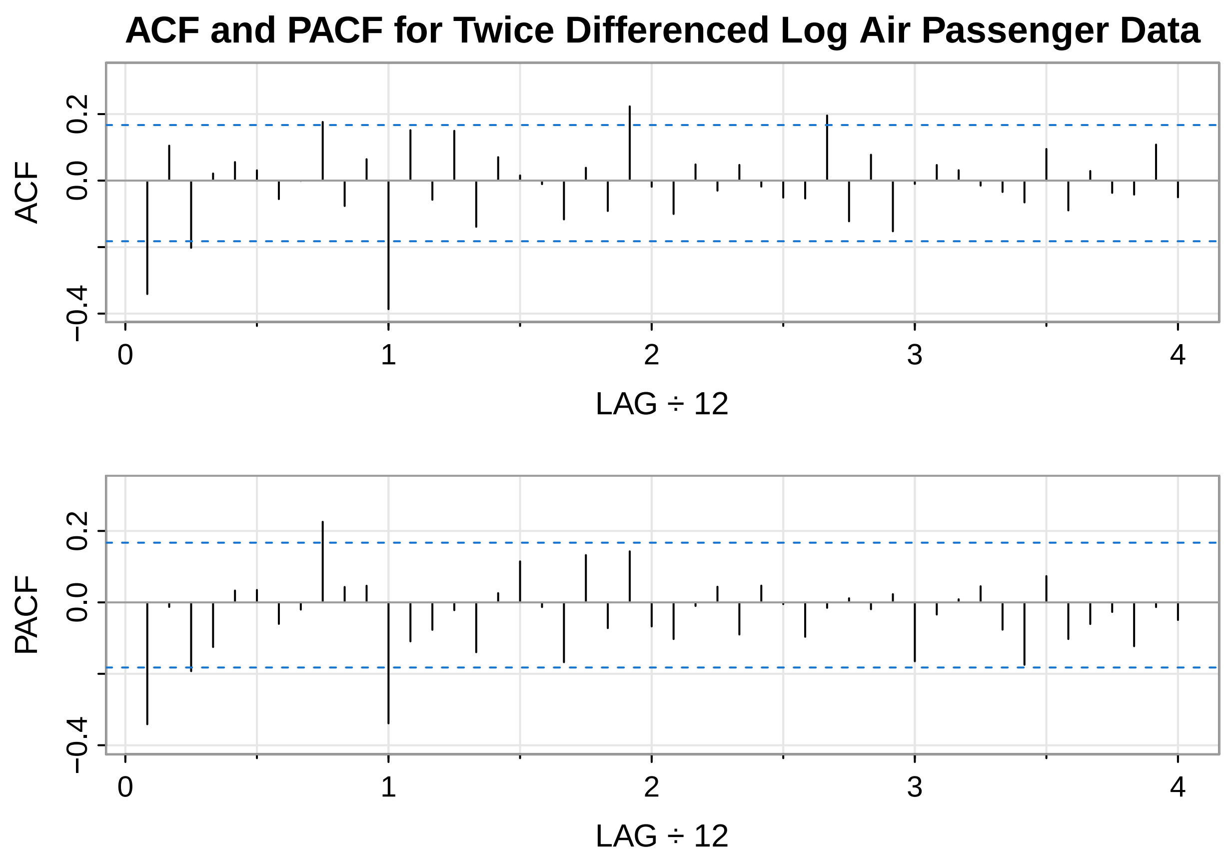
```
data = AirPassengers
log_data = log(data)
dlog_data = diff(log_data, lag = 1)
ddlog_data = diff(dlog_data, lag = 12)
```

```
plot_data = cbind(Orig. = data,
                  Log = log_data,
                  LogDiff = dlog_data,
                  LogDiff12 = ddlog_data)
plot.ts(plot_data, main = "AirPassengers Data & Transformations",
        col = 1:4,
        ylab = "Value",
        xlab = "Time")
```



3. We continue our data exploration by producing both an ACF and PACF plot for our twice differenced log data using the acf2() function. Note your observations about the significant spikes in autocorrelation and partial autocorrelation.

```
acf2(ddlog_data, main = "ACF and PACF for Twice Differenced Log Air Passenger Data")
```




```
##       [,1]  [,2]  [,3]  [,4] [,5] [,6]  [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  -0.34  0.11 -0.20  0.02 0.06 0.03 -0.06  0.00 0.18 -0.08  0.06 -0.39  0.15
## PACF -0.34 -0.01 -0.19 -0.13 0.03 0.03 -0.06 -0.02 0.23  0.04  0.05 -0.34 -0.11
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.06  0.15 -0.14  0.07  0.02 -0.01 -0.12  0.04 -0.09  0.22 -0.02  -0.1
## PACF -0.08 -0.02 -0.14  0.03  0.11 -0.01 -0.17  0.13 -0.07  0.14 -0.07  -0.1
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.05 -0.03  0.05 -0.02 -0.05 -0.05  0.20 -0.12  0.08 -0.15 -0.01  0.05
## PACF -0.01  0.04 -0.09  0.05  0.00 -0.10 -0.02  0.01 -0.02  0.02 -0.16 -0.03
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.03 -0.02 -0.03 -0.07  0.10 -0.09  0.03 -0.04 -0.04  0.11 -0.05
## PACF  0.01  0.05 -0.08 -0.17  0.07 -0.10 -0.06 -0.03 -0.12 -0.01 -0.05
```

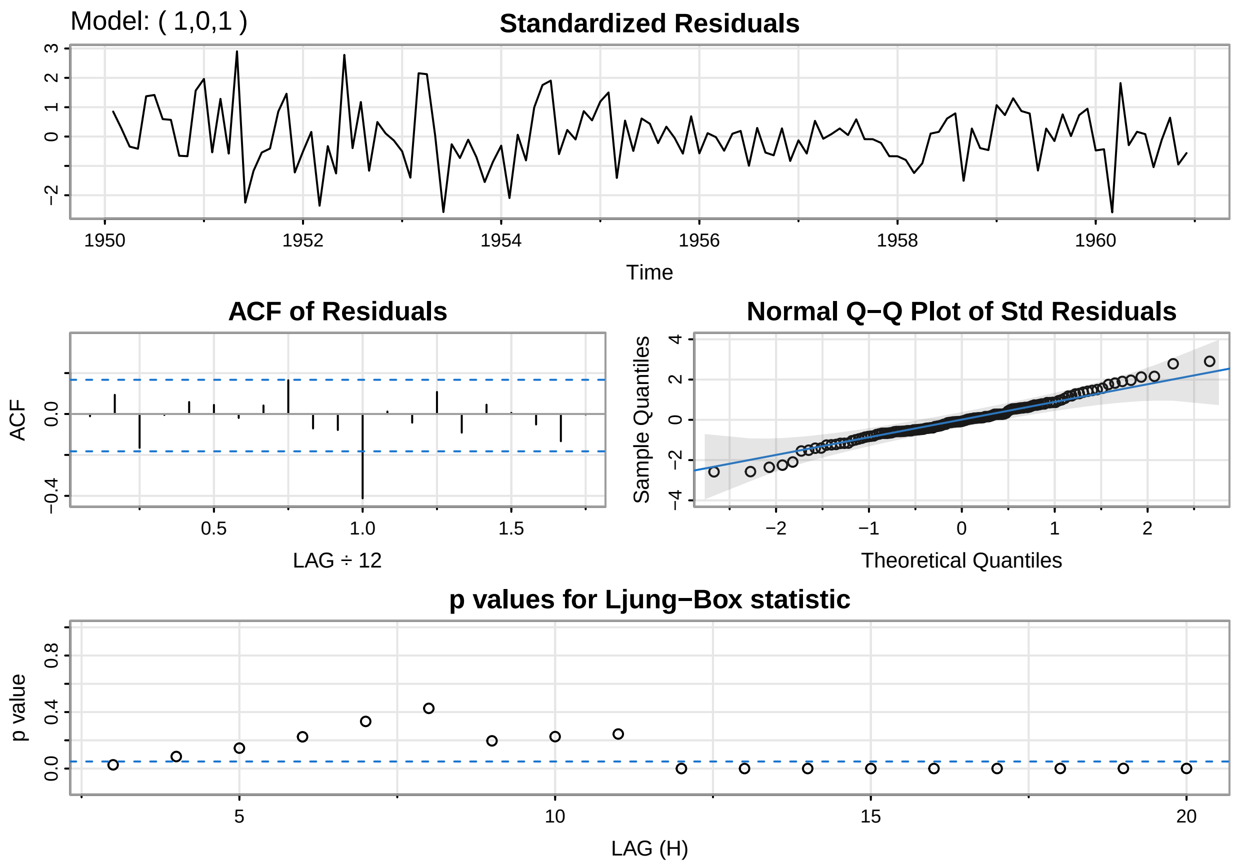
After plotting the ACF and PACF of our twice differenced log data, we see multiple significant spikes at lag 1, lag 12, lag 23, and lag 32. As for the PACF, we see 4 significant spikes as well, this time at lag 1, lag 3, lag 9, and lag 12.

4. Let us first consider a non-seasonal ARMA(1,1) model which we fit using the sarima() function (this time fitting the model to the twice differenced log data)

```
sarima(ddlog_data, p = 1, d = 0, q = 1)
```

```
## initial  value -3.085190
## iter   2 value -3.104607
```

```
## iter   3 value -3.148928
## iter   4 value -3.149123
## iter   5 value -3.149204
## iter   6 value -3.149520
## iter   7 value -3.150137
## iter   8 value -3.151126
## iter   9 value -3.152073
## iter  10 value -3.152362
## iter  11 value -3.152388
## iter  12 value -3.152422
## iter  13 value -3.152531
## iter  14 value -3.152575
## iter  15 value -3.152593
## iter  16 value -3.152594
## iter  17 value -3.152594
## iter  17 value -3.152594
## iter  17 value -3.152594
## final  value -3.152594
## converged
## initial  value -3.152691
## iter   2 value -3.152737
## iter   3 value -3.152754
## iter   4 value -3.152757
## iter   5 value -3.152763
## iter   6 value -3.152778
## iter   7 value -3.152778
## iter   7 value -3.152778
## iter   7 value -3.152778
## final  value -3.152778
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ar1     0.1448 0.2454  0.5900  0.5562
## ma1    -0.5190 0.2179 -2.3822  0.0187
## xmean   0.0003 0.0021  0.1243  0.9013
##
## sigma^2 estimated as 0.001823639 on 128 degrees of freedom
##
## AIC = -3.406611  AICc = -3.405168  BIC = -3.318818
##
```

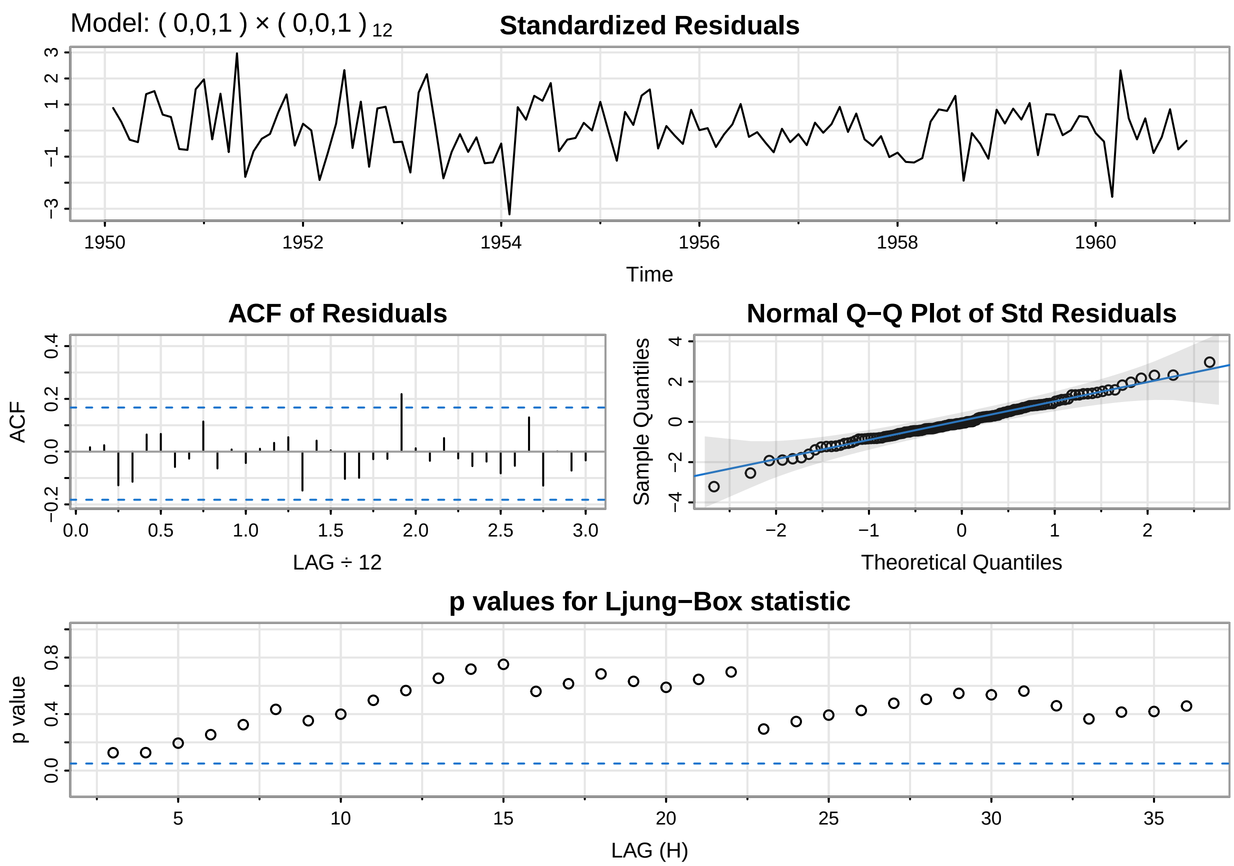


5. Use your ACF and PACF plot above to suggest possible parameters for the parameters SARIMA(p, d, q)(P, D, Q)s. Consider two potential models and fit each in turn using the sarima() function.

```
#model 1
sarima_model1 <- sarima(ddlog_data, p=0, d=0, q=1, P=0, D=0, Q=1, S=12)
```

```
## initial  value -3.086249
## iter   2 value -3.267992
## iter   3 value -3.280018
## iter   4 value -3.284794
## iter   5 value -3.289109
## iter   6 value -3.289719
## iter   7 value -3.289790
## iter   8 value -3.289791
## iter   8 value -3.289791
## final  value -3.289791
## converged
## initial  value -3.286102
## iter   2 value -3.286479
## iter   3 value -3.286948
## iter   4 value -3.286955
## iter   5 value -3.286957
## iter   5 value -3.286957
## iter   5 value -3.286957
## final  value -3.286957
## converged
## <><><><><><><><><><><><><><>
```

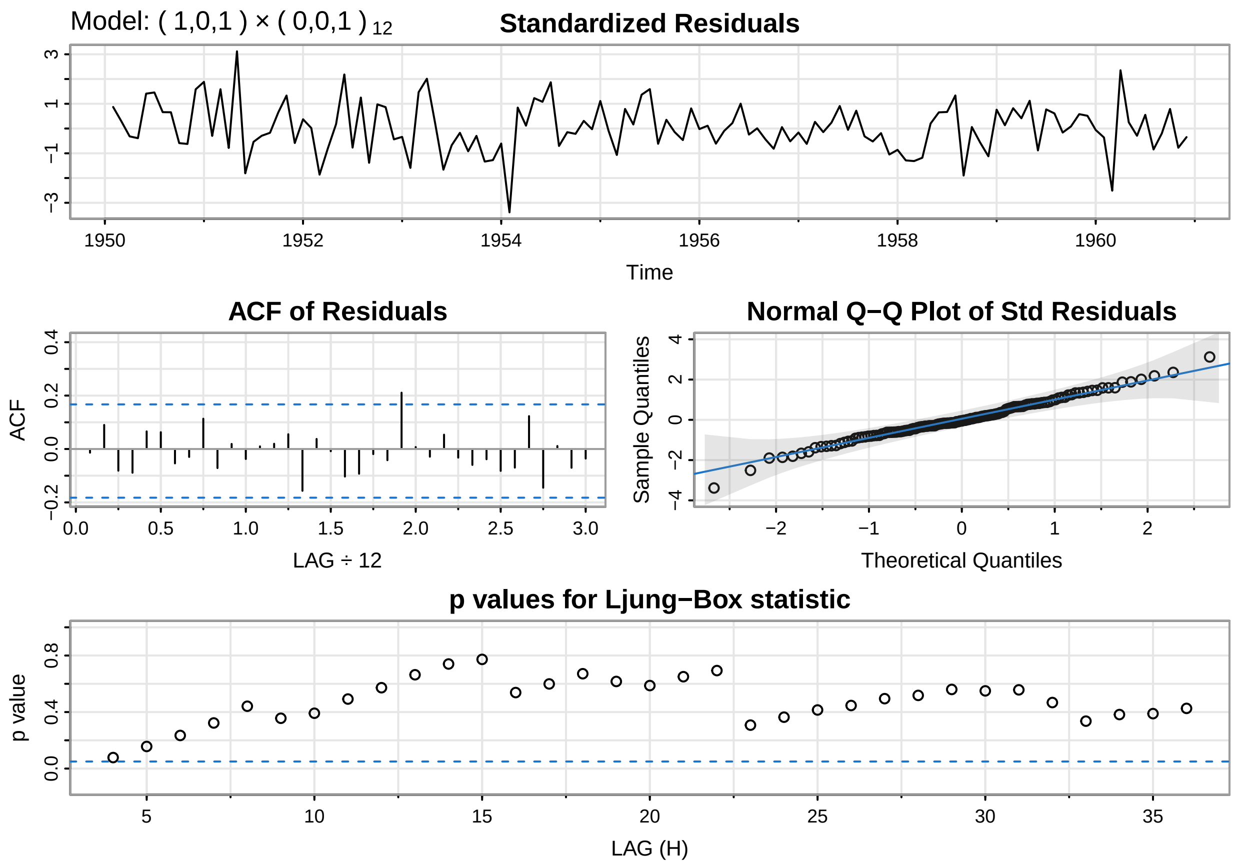
```
##
## Coefficients:
##       Estimate     SE t.value p.value
## ma1    -0.4021 0.0897 -4.4824  0.0000
## sma1   -0.5577 0.0732 -7.6221  0.0000
## xmean  -0.0002 0.0010 -0.1662  0.8683
##
## sigma^2 estimated as 0.001347656 on 128 degrees of freedom
##
## AIC = -3.674967  AICc = -3.673525  BIC = -3.587175
##
```




```r
#model 2
sarima_model2 <- sarima(ddlog_data, p=1, d=0, q=1, P=0, D=0, Q=1, S=12)
```

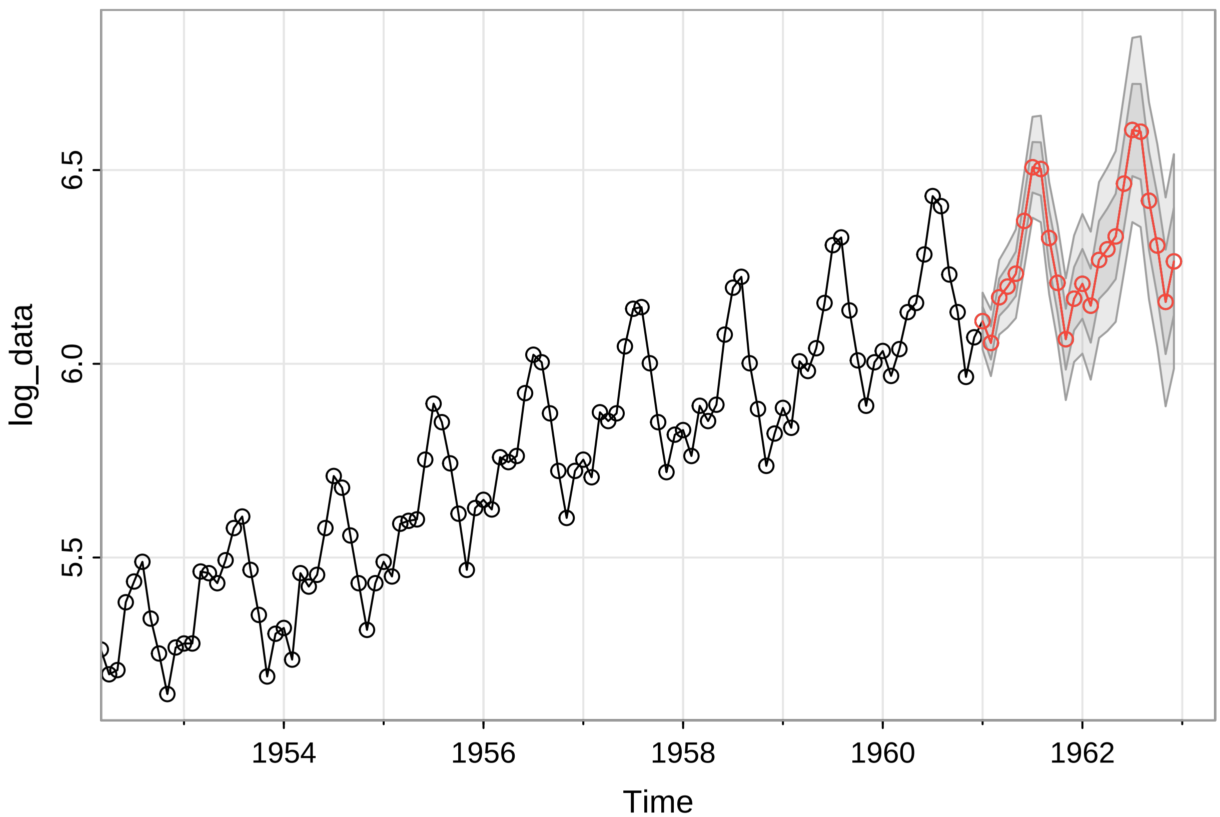
```
## initial  value -3.085190
## iter   2 value -3.225406
## iter   3 value -3.276528
## iter   4 value -3.279837
## iter   5 value -3.281599
## iter   6 value -3.282264
## iter   7 value -3.282693
## iter   8 value -3.283643
## iter   9 value -3.285380
## iter  10 value -3.287348
```

```
## iter  11 value -3.288377
## iter  12 value -3.288617
## iter  13 value -3.288703
## iter  14 value -3.288795
## iter  15 value -3.288984
## iter  16 value -3.289119
## iter  17 value -3.289152
## iter  18 value -3.289152
## iter  18 value -3.289152
## iter  18 value -3.289152
## final  value -3.289152
## converged
## initial  value -3.288422
## iter   2 value -3.288480
## iter   3 value -3.288560
## iter   4 value -3.288640
## iter   5 value -3.288773
## iter   6 value -3.288886
## iter   7 value -3.288919
## iter   8 value -3.288922
## iter   8 value -3.288922
## iter   8 value -3.288922
## final  value -3.288922
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ar1     0.1997 0.2488  0.8028  0.4236
## ma1    -0.5820 0.2141 -2.7185  0.0075
## sma1   -0.5657 0.0750 -7.5404  0.0000
## xmean  -0.0002 0.0008 -0.2066  0.8367
##
## sigma^2 estimated as 0.00134043 on 127 degrees of freedom
##
## AIC = -3.663631  AICc = -3.661208  BIC = -3.553891
##
```



6. Produce a 24 month ahead forecast for our data using the sarima.for() function. Does our forecast appear reasonable?

```
sarima.for(log_data, n.ahead=24, p=0, d=1, q=1, P=0, D=1, Q=1, S=12)
```




```
## $pred
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1961 6.110186 6.053775 6.171715 6.199300 6.232556 6.368779 6.507294 6.502906
## 1962 6.206435 6.150025 6.267964 6.295550 6.328805 6.465028 6.603543 6.599156
##           Sep      Oct      Nov      Dec
## 1961 6.324698 6.209008 6.063487 6.168025
## 1962 6.420947 6.305257 6.159737 6.264274
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 1961 0.03671562 0.04278291 0.04809072 0.05286830 0.05724856 0.06131670
## 1962 0.09008475 0.09549708 0.10061869 0.10549195 0.11014981 0.11461854
##             Jul        Aug        Sep        Oct        Nov        Dec
## 1961 0.06513124 0.06873441 0.07215787 0.07542612 0.07855851 0.08157070
## 1962 0.11891946 0.12307018 0.12708540 0.13097758 0.13475740 0.13843405
```

Our sarima model (using model 1) seems rather reasonable as it has been able to capture the upward trend and seasonality that is evident with past months/years of the time series.