Employment Analysis

Leala Darby

03/11/2020

First load all required packages:

```
library(car)
library(tseries)
library(astsa)
```

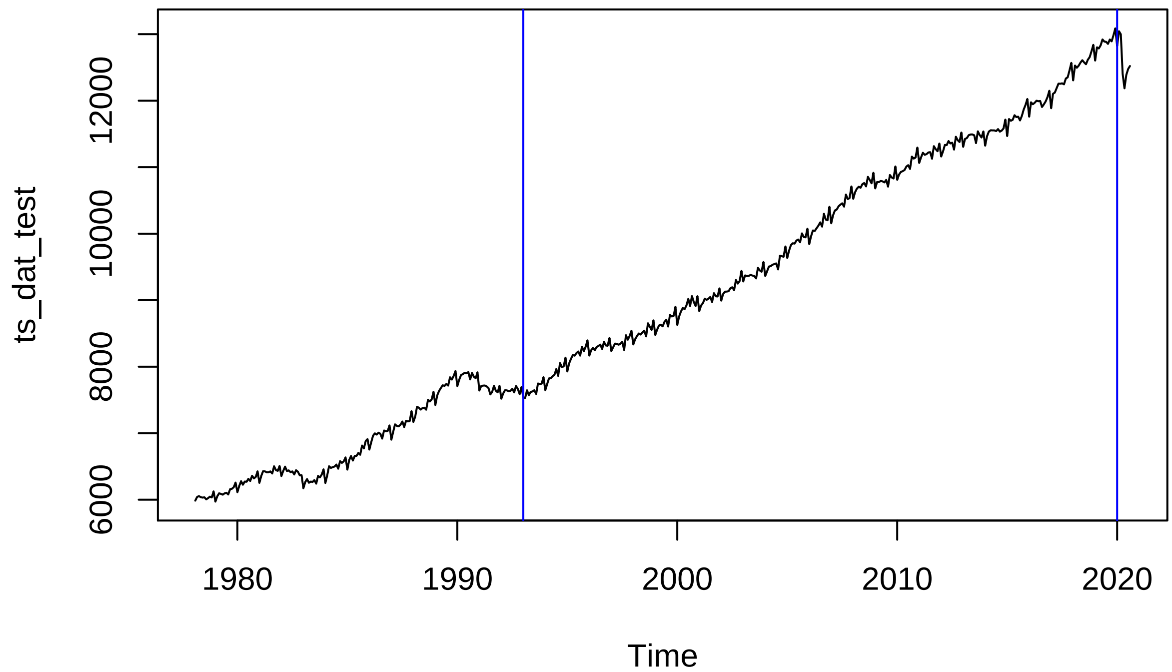
Load in the data:

```
dat <- read.csv("employment_data.csv", fileEncoding = 'UTF-8-BOM')
head(dat)
```

```
##   Observation.times Time.series.values
## 1            Feb-78             5985.7
## 2            Mar-78             6040.6
## 3            Apr-78             6054.2
## 4            May-78             6038.3
## 5            Jun-78             6031.3
## 6            Jul-78             6036.1
```

Create a time series object from the data and plot

```
ts_dat_test <- ts(dat[, 2], start = c(1978, 2), end = c(2020, 8), frequency = 12)
plot.ts(ts_dat_test)
abline(v = 1993, col = "blue")
abline(v = 2020, col = "blue")
```



Instructed to truncate data from January 1993 to December 2019 (inclusive)

```
dat[dat$Observation.times == "Jan-93",]
```

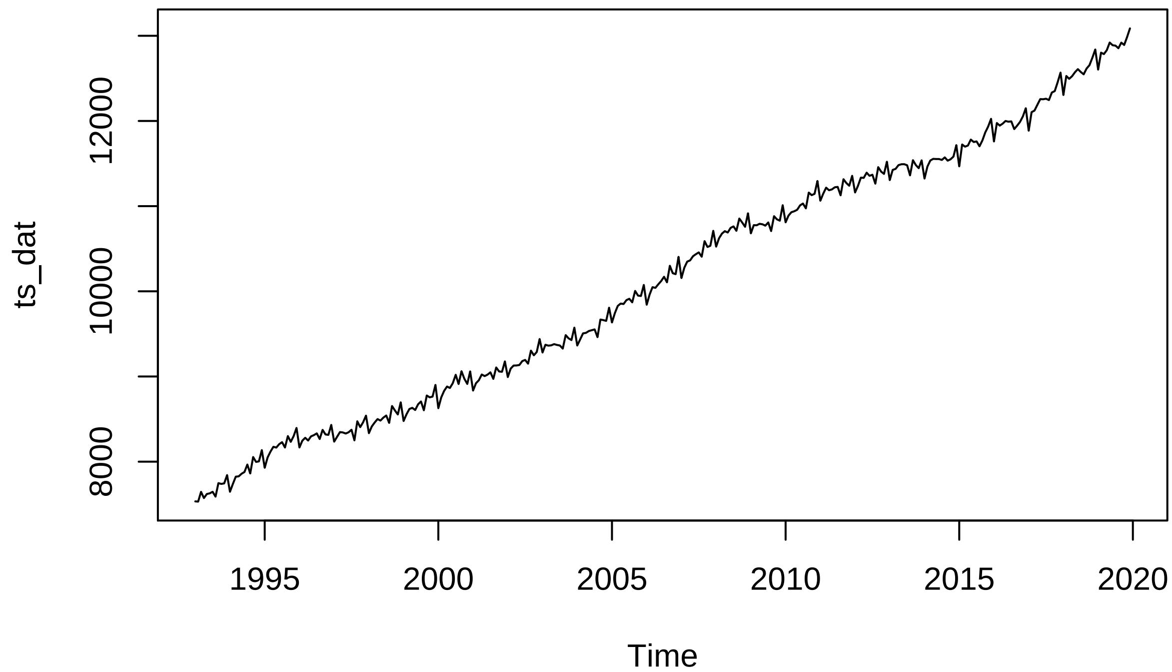
```
##     Observation.times Time.series.values
## 180            Jan-93             7533.7
```

```
dat[dat$Observation.times == "Dec-19",]
```

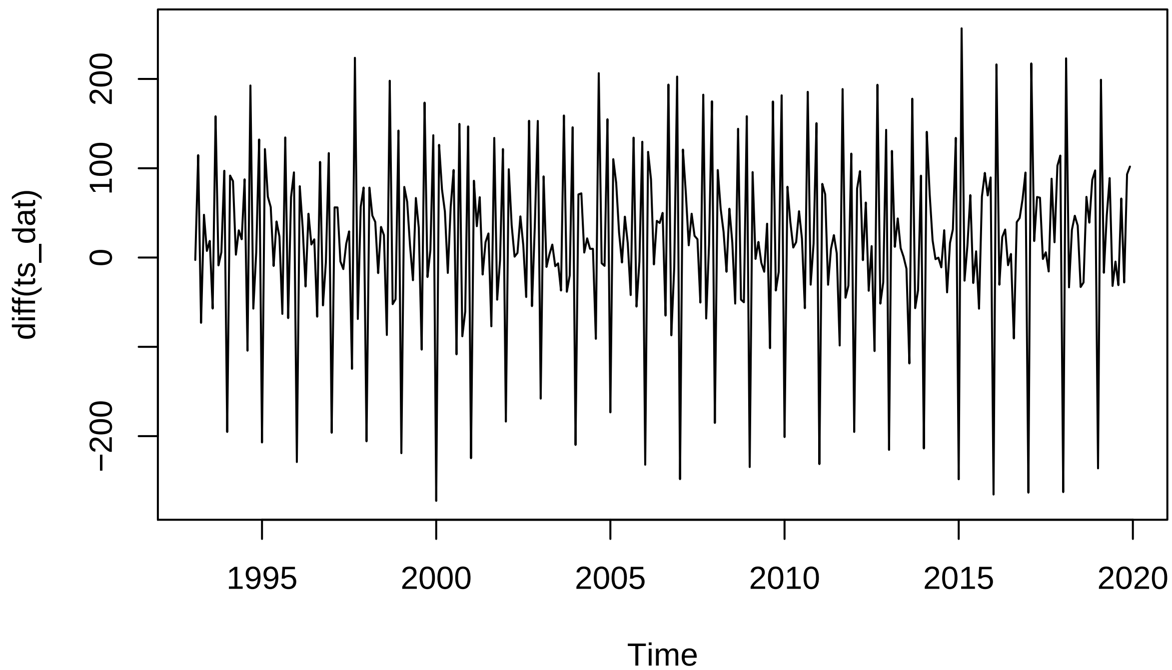
```
##     Observation.times Time.series.values
## 503            Dec-19            13087.1
```

So we only need rows 180-503.

```
trunc_dat <- dat[180:503,]
ts_dat <- ts(trunc_dat[, 2], start = c(1993, 1), end = c(2019, 12), frequency = 12)
plot.ts(ts_dat)
```




```r
plot.ts(diff(ts_dat))
```



The trend in mean is readily observable. Difficult to determine a trend in variance - there appears to be frequent changes, which are easier to see after incorporating lags of 1. Check statistically for stationarity using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, which has the following hypotheses [. . .]:

```
kpss.test(ts_dat)
```

```
## Warning in kpss.test(ts_dat): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts_dat
## KPSS Level = 5.4937, Truncation lag parameter = 5, p-value = 0.01
```

The small p-value indicates that we should reject the null and conclude that the ts is not stationary.

As a rough test of constant variance (Levene's isn't really valid because time series data isn't independent)

```
length(ts_dat)
```

```
## [1] 324
```

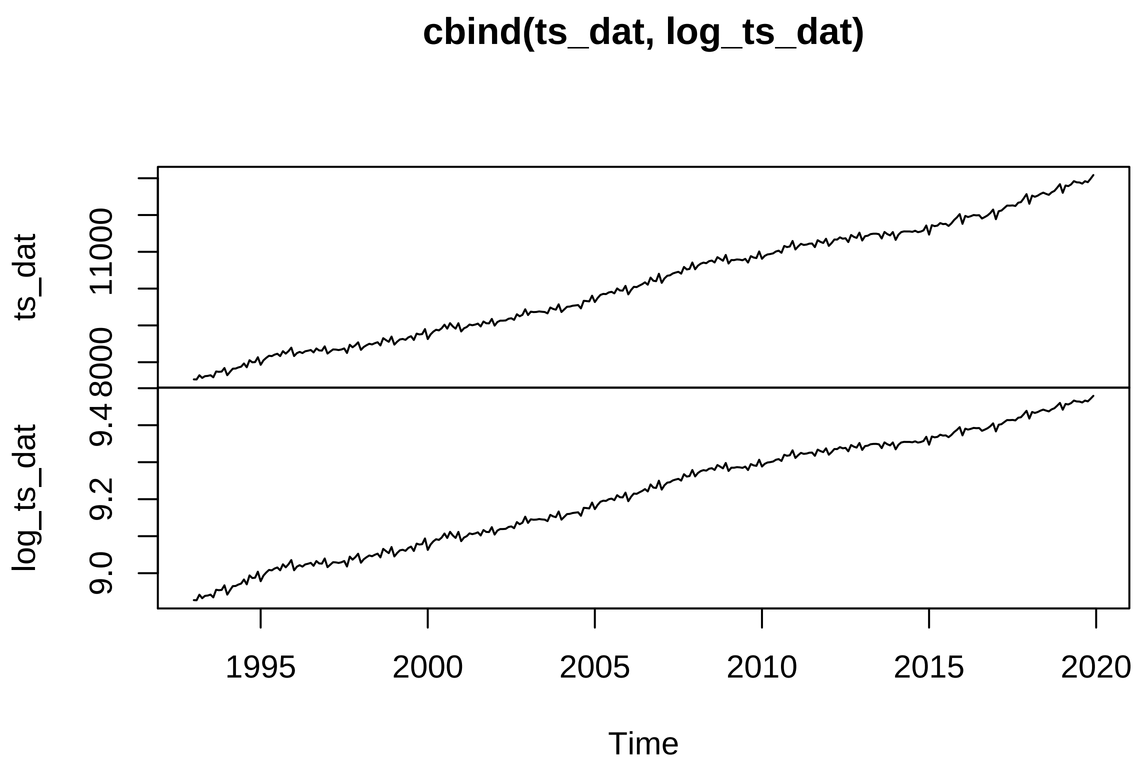
```
Group <- c(rep(1,81), rep(2, 81), rep(3, 81), rep(4, 81))
leveneTest(ts_dat, Group)
```

```
## Warning in leveneTest.default(ts_dat, Group): Group coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  7.2516 0.0001013 ***
##       320
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The small p-value of 0.0001013 confirms that the data exhibits heteroscedasticity. Therefore we will perform a log transformation to attempt to reduce this:

```r
log_ts_dat <- log(ts_dat)
plot.ts(cbind(ts_dat, log_ts_dat))
```




```r
leveneTest(log_ts_dat, Group)
```

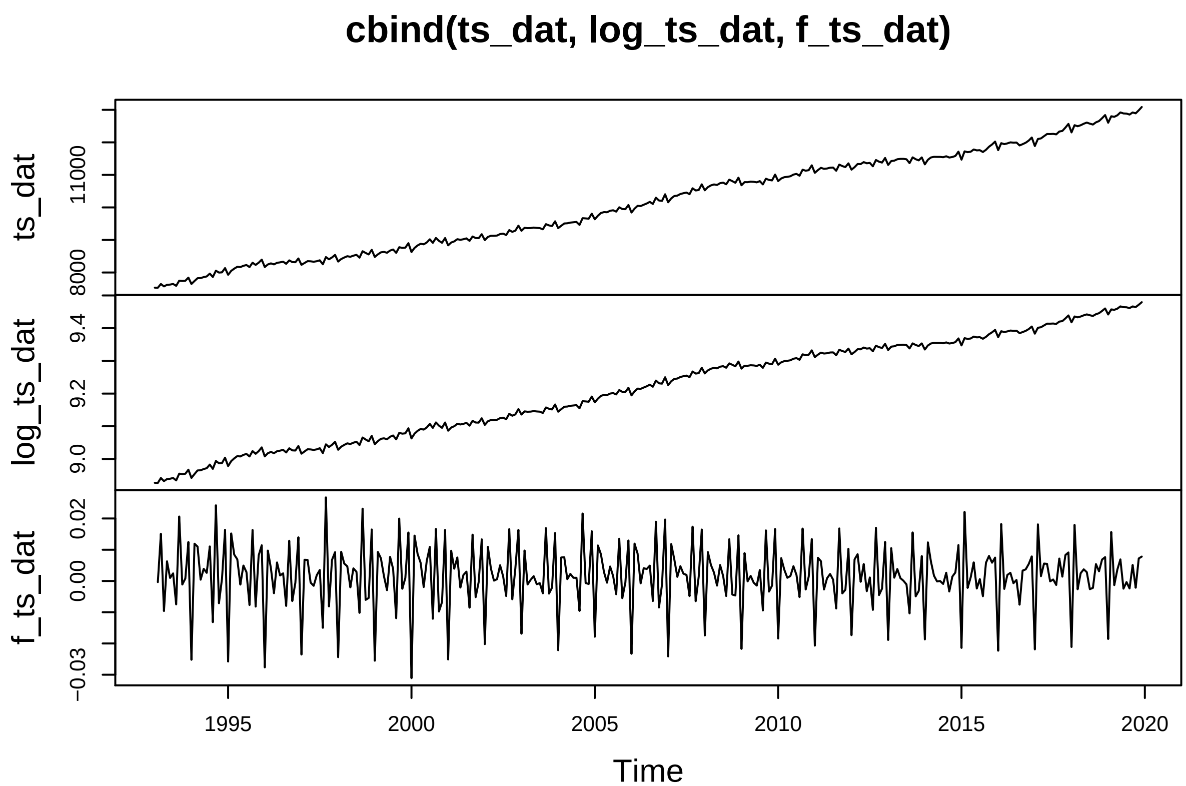
```
## Warning in leveneTest.default(log_ts_dat, Group): Group coerced to factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   3  1.4631 0.2245
##       320
```

At a significance level of 5%, the p-value above of 0.2245 provides very weak evidence and we fail to reject the null hypothesis of equal variance among groups. Thus the heteroscedasticity has been reduced.

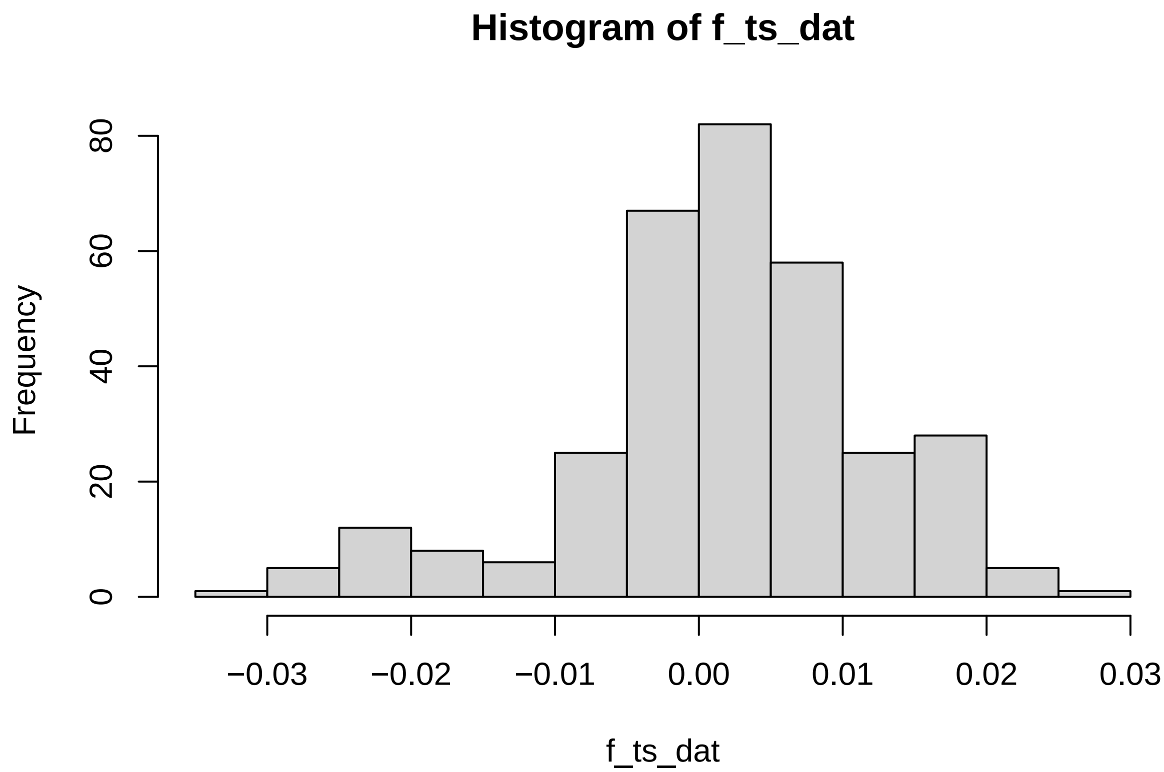
Next, to reduce the trend in mean, apply differencing of 1 lag to our TS with stabilised variance:

```
f_ts_dat <- diff(log_ts_dat, 1)
plot.ts(cbind(ts_dat, log_ts_dat, f_ts_dat))
```



To confirm constant mean and variance and a Gaussian distribution for the time series, a Shapiro-Wilk normality test is performed:

```
hist(f_ts_dat)
```




```r
shapiro.test(f_ts_dat)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  f_ts_dat
## W = 0.96138, p-value = 1.534e-07
```

The small p-value indicates likely non-normality, but this test isn't really valid for TS. Instead, check statistically for stationarity using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```r
kpss.test(log_ts_dat)
```

```
## Warning in kpss.test(log_ts_dat): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  log_ts_dat
## KPSS Level = 5.4933, Truncation lag parameter = 5, p-value = 0.01
```

```r
kpss.test(f_ts_dat)
```

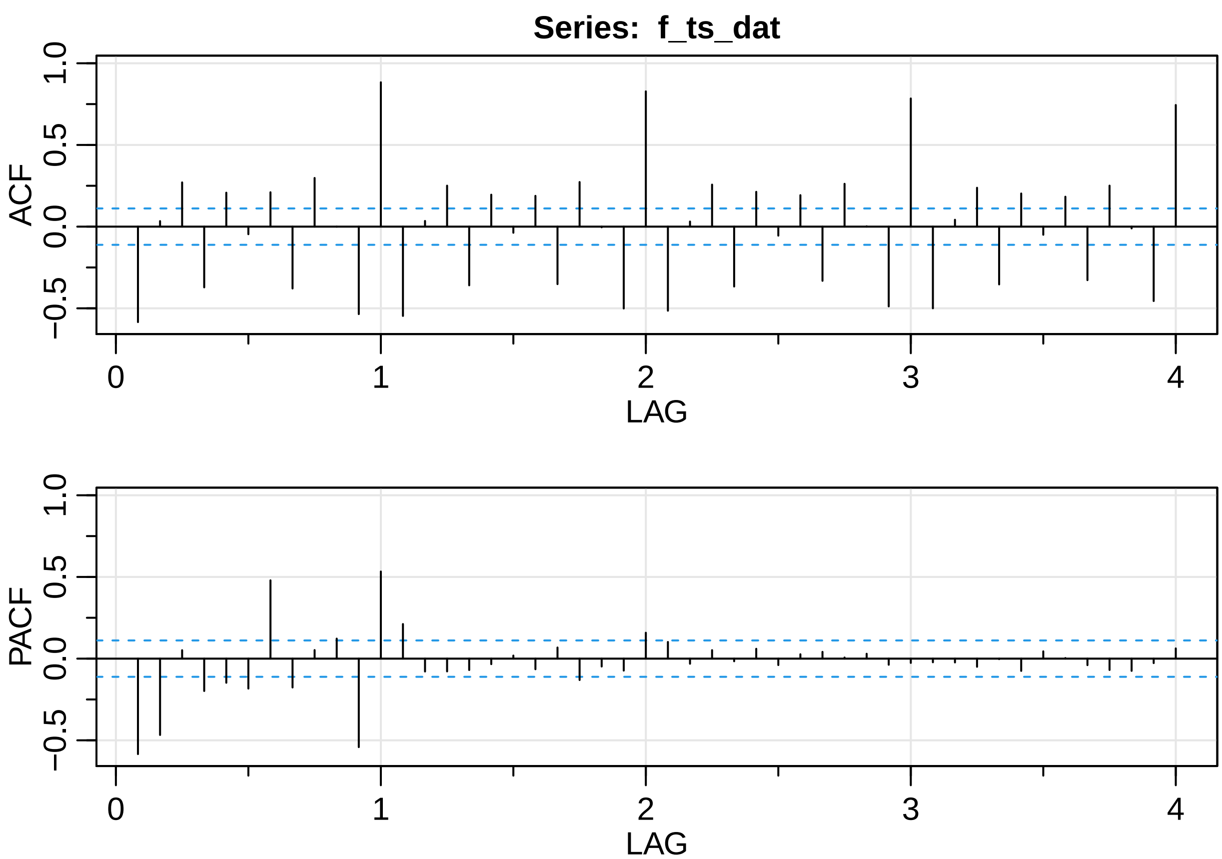
```
## Warning in kpss.test(f_ts_dat): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  f_ts_dat
## KPSS Level = 0.064047, Truncation lag parameter = 5, p-value = 0.1
```

The final ts has a high p-value of 0.1, which is statistically significant at a significance level of 5%. Therefore we fail to reject the null hypothesis, and have reasonable evidence that the final ts is stationary.

Next, the ACF and PACF of the differenced ts are plotted in order to estimate p and q.

```
acf2(f_ts_dat)
```




```
##       [,1]  [,2] [,3]  [,4]  [,5]  [,6] [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  -0.58  0.03 0.27 -0.37  0.21 -0.05 0.21 -0.38 0.30  0.00 -0.54  0.88 -0.55
## PACF -0.58 -0.47 0.05 -0.20 -0.15 -0.18 0.48 -0.18 0.05  0.12 -0.54  0.53  0.21
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.03  0.25 -0.36  0.20 -0.04  0.19 -0.35  0.27  0.00 -0.50  0.83 -0.51
## PACF -0.08 -0.08 -0.07 -0.03  0.02 -0.06  0.07 -0.13 -0.05 -0.07  0.16  0.10
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.03  0.26 -0.37  0.21 -0.06  0.19 -0.33  0.26  0.00 -0.49  0.78 -0.50
## PACF -0.03  0.05 -0.02  0.06 -0.04  0.03  0.04  0.01  0.03 -0.04 -0.03 -0.02
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.04  0.24 -0.35  0.20 -0.05  0.18 -0.33  0.25 -0.01 -0.46  0.74
## PACF -0.02 -0.05  0.00 -0.08  0.04  0.00 -0.04 -0.07 -0.07 -0.03  0.06
```

Seasonal patterns are clear, more strongly in the ACF plot.
Will fit a SARIMA(p,d,q)(P,D,Q)_s model.

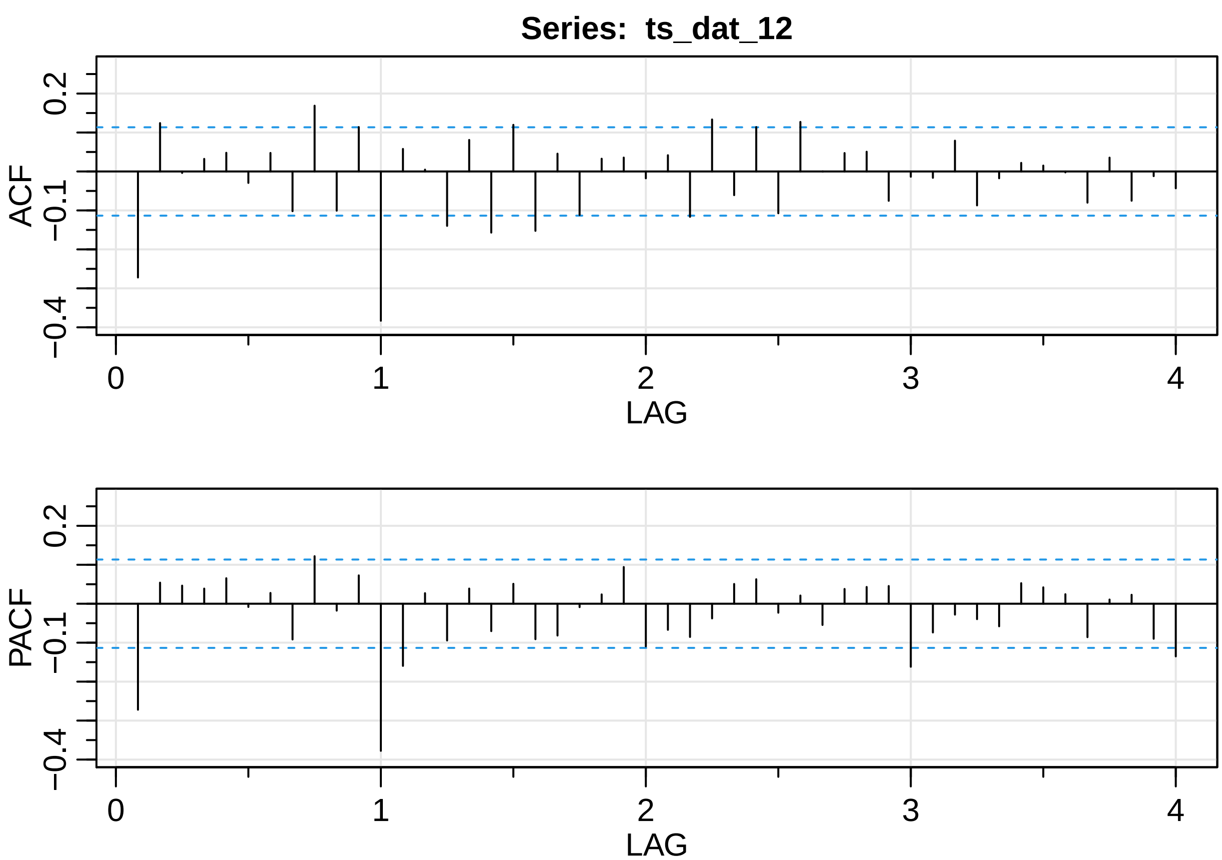
The data being monthly and the ACF plot having its highest peaks at lags $h = 12, 24, 36, 48$ implies a seasonal trend of 12 would be a good choice. Slow decay over these four peaks suggests there is a difference between seasons. To remove this trend, difference the ts on the seasonal lag:

```
ts_dat_12 <- diff(f_ts_dat, 12)
kpss.test(ts_dat_12) #Big enough to call stationary
```

```
## Warning in kpss.test(ts_dat_12): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts_dat_12
## KPSS Level = 0.025427, Truncation lag parameter = 5, p-value = 0.1
```

```
acf2(ts_dat_12)
```




```
##        [,1] [,2] [,3] [,4] [,5]  [,6] [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  -0.27 0.12 0.00 0.03 0.05 -0.03 0.05 -0.10 0.17 -0.10  0.11 -0.38  0.06
## PACF -0.27 0.05 0.05 0.04 0.07 -0.01 0.03 -0.09 0.12 -0.02  0.07 -0.38 -0.16
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.00 -0.14  0.08 -0.16  0.12 -0.15  0.05 -0.11  0.03  0.04 -0.02  0.04
```

```
## PACF  0.03 -0.09  0.04 -0.07  0.05 -0.09 -0.08 -0.01  0.02  0.09 -0.11 -0.07
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF  -0.12  0.13 -0.06  0.11 -0.11  0.13  0.00  0.05  0.05 -0.08 -0.01 -0.02
## PACF -0.09 -0.04  0.05  0.06 -0.02  0.02 -0.05  0.04  0.04  0.05 -0.16 -0.07
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.08 -0.09 -0.02  0.02  0.02  0.00 -0.08  0.04 -0.08 -0.01 -0.04
## PACF -0.03 -0.04 -0.06  0.05  0.04  0.02 -0.09  0.01  0.02 -0.09 -0.14
```

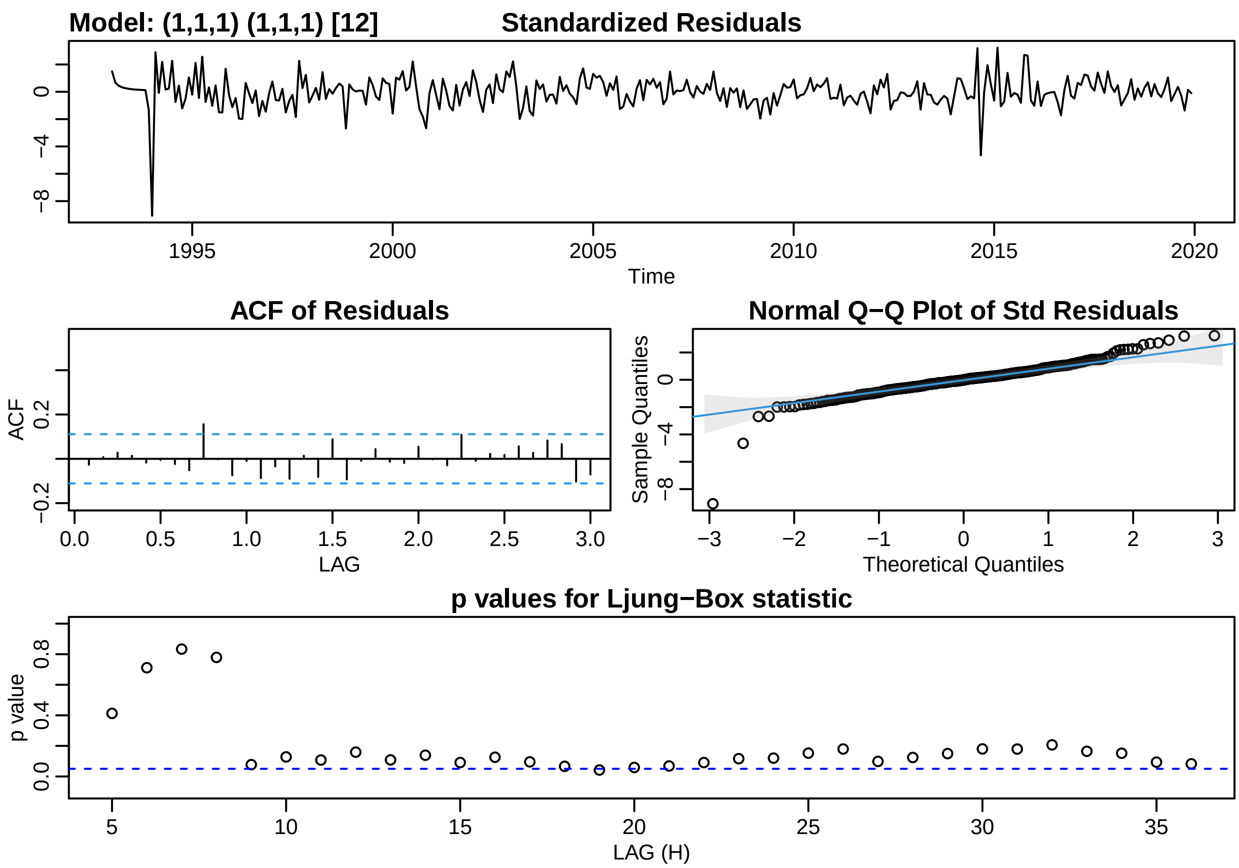
First examine these plots at seasonal lags h = 1S(=12), 2S,... Strong peak at 1S in both the ACF and PACF. Might indicate: 1) ACF and PACF both tail off at seasonal lags after spikes at 1S in both, suggesting $P = 1$ and $Q = 1$ 2) ACF cuts off after lag 1S and PACF tails off at seasonal lags, suggesting $P = 0$ and $Q = 1$ 3) ACF tails off at seasonal lags and PACF cuts off after lag 1s, suggesting $P = 1$ and $Q = 0$ So $0 \leq P \leq 1$ and $0 \leq Q \leq 1$.

Now examine at $h = 1, 2, ..., 11$ to estimate p and q. This is kind of hard? They don't really seem to tail/cut off in either plot. Try: 1) ACF and PACF both tail off, suggesting $p = q = 1$ 2) ACF cuts off and PACF tails off: $p = 0$ and $q = 1$ 3) ACF tails off and PACF cuts off: $p = 1$ and $q = 0$

```
sarima(log_ts_dat, p = 1, d = 1, q = 1, P = 1, D = 1, Q = 1, S = 12) #AICc -8.161924
```

```
## initial  value -5.504137
## iter   2 value -5.599083
## iter   3 value -5.658866
## iter   4 value -5.662290
## iter   5 value -5.666833
## iter   6 value -5.671616
## iter   7 value -5.673051
## iter   8 value -5.673605
## iter   9 value -5.673704
## iter  10 value -5.673742
## iter  11 value -5.673767
## iter  12 value -5.673873
## iter  13 value -5.673902
## iter  14 value -5.673916
## iter  15 value -5.673927
## iter  16 value -5.673931
## iter  16 value -5.673931
## iter  16 value -5.673931
## final  value -5.673931
## converged
## initial  value -5.656482
## iter   2 value -5.658138
## iter   3 value -5.659588
## iter   4 value -5.660390
## iter   5 value -5.660497
## iter   6 value -5.660506
## iter   7 value -5.660510
## iter   8 value -5.660514
## iter   9 value -5.660520
## iter  10 value -5.660523
## iter  11 value -5.660523
## iter  12 value -5.660523
## iter  13 value -5.660523
## iter  13 value -5.660523
```

```
## iter  13 value -5.660523
## final  value -5.660523
## converged
```

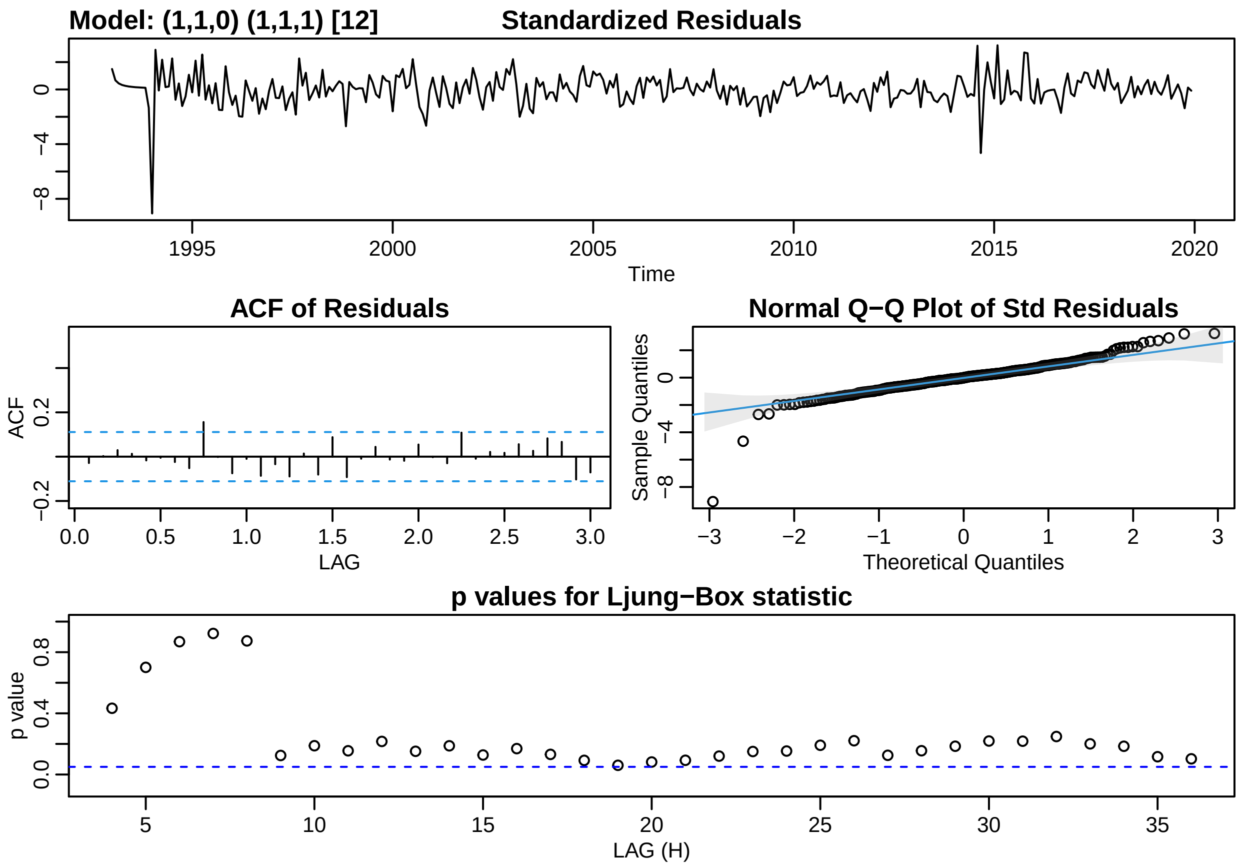



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     ma1    sar1     sma1
##       -0.2909  -0.022  0.0975  -0.6923
## s.e.   0.1496   0.153  0.0904   0.0682
##
## sigma^2 estimated as 1.187e-05:  log likelihood = 1319.13,  aic = -2628.27
##
## $degrees_of_freedom
## [1] 307
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1   -0.2909 0.1496  -1.9448  0.0527
## ma1   -0.0220 0.1530  -0.1436  0.8859
```

```
## sar1   0.0975 0.0904   1.0787  0.2816
## sma1  -0.6923 0.0682 -10.1501  0.0000
##
## $AIC
## [1] -8.162316
##
## $AICc
## [1] -8.161924
##
## $BIC
## [1] -8.104245
```

```r
# ttable says ma1 coeff has highest p-value. removing this:
sarima(log_ts_dat, p = 1, d = 1, q = 0, P = 1, D = 1, Q = 1, S = 12) #AICc -8.168226
```

```
## initial  value -5.504137
## iter   2 value -5.629051
## iter   3 value -5.662992
## iter   4 value -5.665957
## iter   5 value -5.673409
## iter   6 value -5.673859
## iter   7 value -5.673905
## iter   8 value -5.673908
## iter   9 value -5.673909
## iter  10 value -5.673910
## iter  10 value -5.673910
## iter  10 value -5.673910
## final  value -5.673910
## converged
## initial  value -5.656550
## iter   2 value -5.658467
## iter   3 value -5.660091
## iter   4 value -5.660386
## iter   5 value -5.660475
## iter   6 value -5.660489
## iter   6 value -5.660489
## iter   6 value -5.660489
## final  value -5.660489
## converged
```

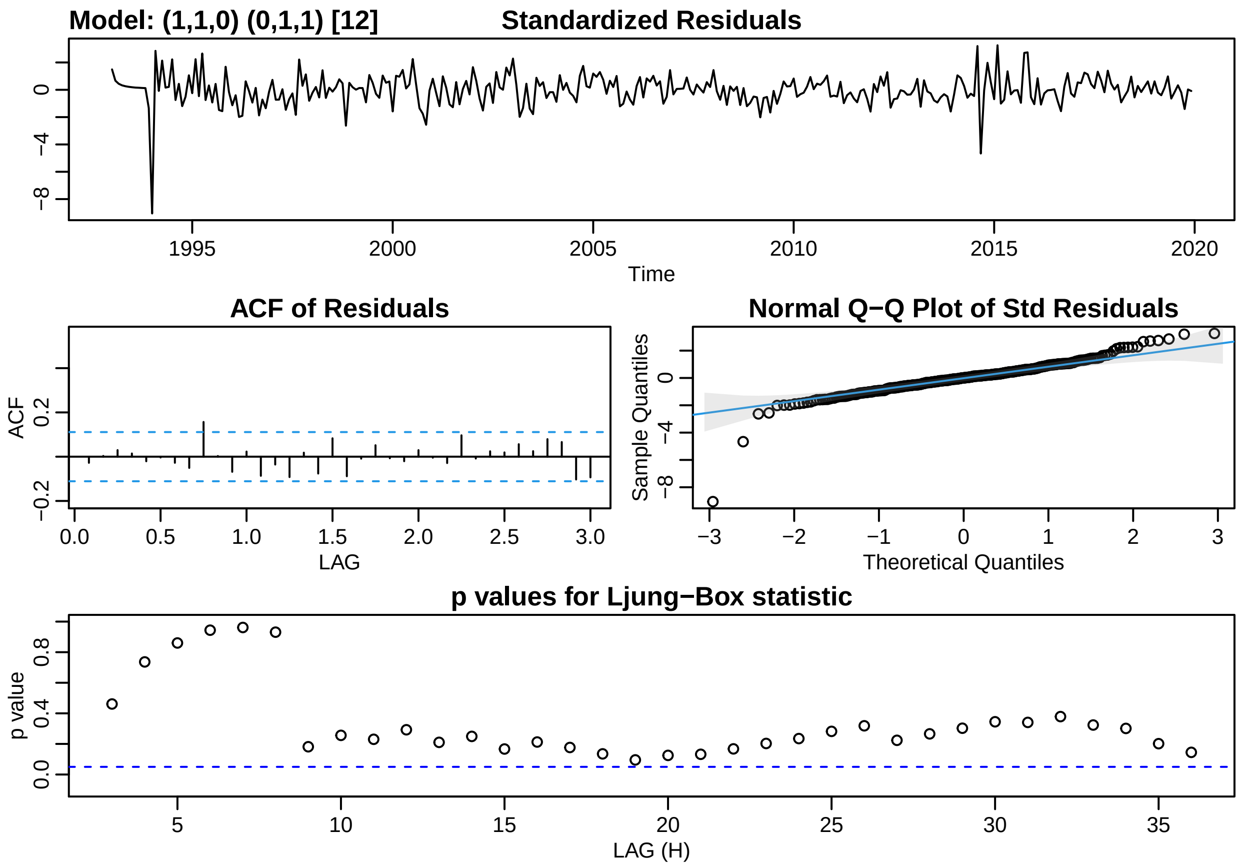



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1    sar1     sma1
##       -0.3106  0.0975  -0.6910
## s.e.   0.0551  0.0905   0.0678
##
## sigma^2 estimated as 1.187e-05:  log likelihood = 1319.12,  aic = -2630.24
##
## $degrees_of_freedom
## [1] 308
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1   -0.3106 0.0551  -5.6372  0.0000
## sar1   0.0975 0.0905   1.0765  0.2825
## sma1  -0.6910 0.0678 -10.1910  0.0000
##
## $AIC
## [1] -8.16846
##
```

```
## $AICc
## [1] -8.168226
##
## $BIC
## [1] -8.122003
```

```
# ttable says sar1 coeff has highest p-value. removing this:
sarima(log_ts_dat, p = 1, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12) #AICc -8.170977
```

```
## initial  value -5.493660
## iter   2 value -5.650636
## iter   3 value -5.665285
## iter   4 value -5.669238
## iter   5 value -5.670283
## iter   6 value -5.670338
## iter   7 value -5.670339
## iter   8 value -5.670339
## iter   8 value -5.670339
## iter   8 value -5.670339
## final  value -5.670339
## converged
## initial  value -5.658068
## iter   2 value -5.658608
## iter   3 value -5.658636
## iter   4 value -5.658637
## iter   4 value -5.658637
## iter   4 value -5.658637
## final  value -5.658637
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     sma1
##       -0.3084  -0.6318
## s.e.   0.0551   0.0499
##
## sigma^2 estimated as 1.192e-05:  log likelihood = 1318.55,  aic = -2631.09
##
## $degrees_of_freedom
## [1] 309
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1   -0.3084 0.0551  -5.5947       0
## sma1  -0.6318 0.0499 -12.6597       0
##
## $AIC
## [1] -8.171094
##
## $AICc
```

```
## [1] -8.170977
##
## $BIC
## [1] -8.136251
```

Is the standardised residuals plot problematic? The normal Q-Q plot has 2 outliers. The ljung-Box statistic is passable at lag 20.

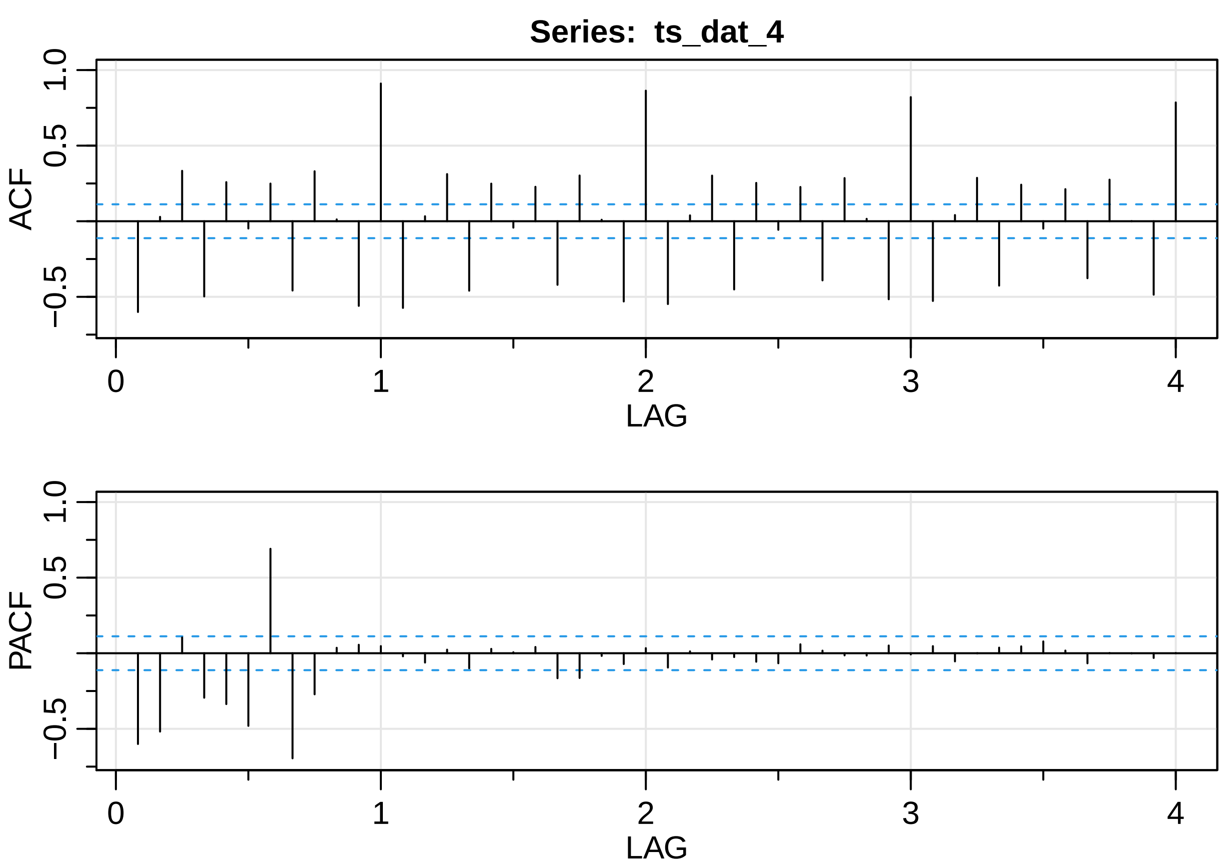
What if the patterns are quarterly, not yearly?

```
ts_dat_4 <- diff(f_ts_dat, 4)
kpss.test(ts_dat_4) # Again, big enough to call stationary
```

```
## Warning in kpss.test(ts_dat_4): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts_dat_4
## KPSS Level = 0.016543, Truncation lag parameter = 5, p-value = 0.1
```

```
acf2(ts_dat_4)
```




```
##      [,1]  [,2] [,3]  [,4]  [,5]  [,6] [,7]  [,8]  [,9] [,10] [,11] [,12] [,13]
```

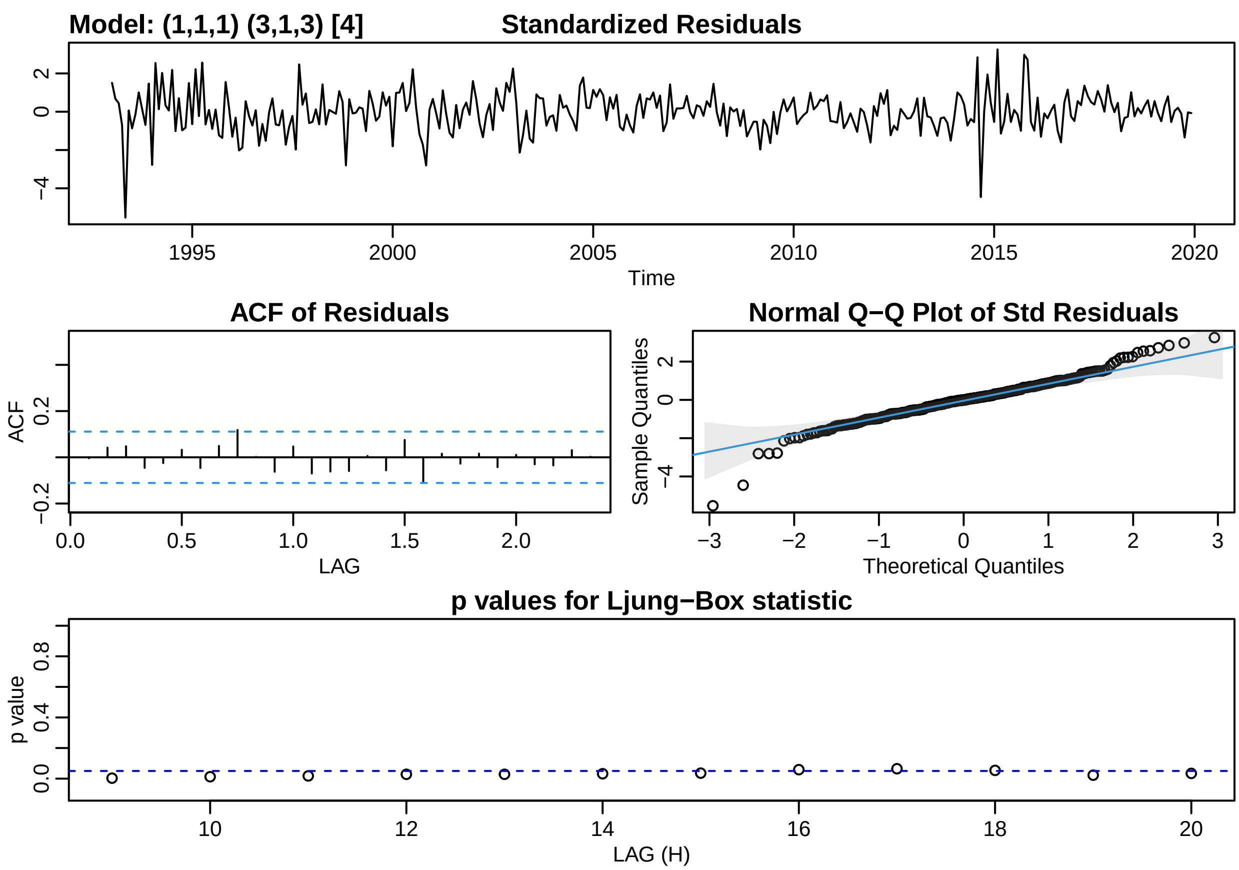
```
## ACF  -0.6  0.03 0.33 -0.50  0.26 -0.05 0.25 -0.46  0.33  0.01 -0.56  0.91 -0.57
## PACF -0.6 -0.52 0.10 -0.29 -0.34 -0.48 0.69 -0.70 -0.27  0.04  0.06  0.05 -0.02
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.03  0.31 -0.46  0.25 -0.04  0.23 -0.42  0.30  0.01 -0.53  0.86 -0.55
## PACF -0.06  0.02 -0.10  0.03  0.01  0.04 -0.17 -0.16 -0.02 -0.07  0.03 -0.09
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.04  0.30 -0.45  0.25 -0.06  0.23 -0.39  0.29  0.02 -0.52  0.82 -0.53
## PACF  0.01 -0.04 -0.03 -0.06 -0.07  0.06  0.02 -0.01 -0.02  0.05 -0.01  0.05
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.04  0.29 -0.43  0.24 -0.05  0.21 -0.38  0.28     0 -0.49  0.79
## PACF -0.05  0.00  0.04  0.05  0.08  0.02 -0.07  0.00     0 -0.03  0.00
```

Struggling with the logic here, but the models are marginally better according to AICc

```
sarima(log_ts_dat, p = 1, d = 1, q = 1, P = 3, D = 1, Q = 3, S = 4) #AICc: -8.299593
```

```
## initial  value -4.081505
## iter   2 value -4.727549
## iter   3 value -5.271372
## iter   4 value -5.302587
## iter   5 value -5.440285
## iter   6 value -5.496142
## iter   7 value -5.534411
## iter   8 value -5.544453
## iter   9 value -5.549042
## iter  10 value -5.560586
## iter  11 value -5.584249
## iter  12 value -5.597707
## iter  13 value -5.626723
## iter  14 value -5.639339
## iter  15 value -5.652357
## iter  16 value -5.665787
## iter  17 value -5.674638
## iter  18 value -5.685247
## iter  19 value -5.686633
## iter  20 value -5.687953
## iter  21 value -5.688028
## iter  22 value -5.688609
## iter  23 value -5.694614
## iter  24 value -5.696183
## iter  25 value -5.698291
## iter  26 value -5.699727
## iter  27 value -5.700334
## iter  28 value -5.700680
## iter  29 value -5.700839
## iter  30 value -5.700915
## iter  31 value -5.701019
## iter  32 value -5.701090
## iter  33 value -5.701108
## iter  34 value -5.701109
## iter  34 value -5.701109
## iter  34 value -5.701109
## final  value -5.701109
```

```
## converged
## initial  value -5.617842
## iter   2 value -5.625830
## iter   3 value -5.626010
## iter   4 value -5.628487
## iter   5 value -5.630207
## iter   6 value -5.631186
## iter   7 value -5.631803
## iter   8 value -5.632193
## iter   9 value -5.632516
## iter  10 value -5.632932
## iter  11 value -5.633241
## iter  12 value -5.633335
## iter  13 value -5.633357
## iter  14 value -5.633371
## iter  15 value -5.633393
## iter  16 value -5.633415
## iter  17 value -5.633425
## iter  18 value -5.633426
## iter  18 value -5.633426
## final  value -5.633426
## converged
```



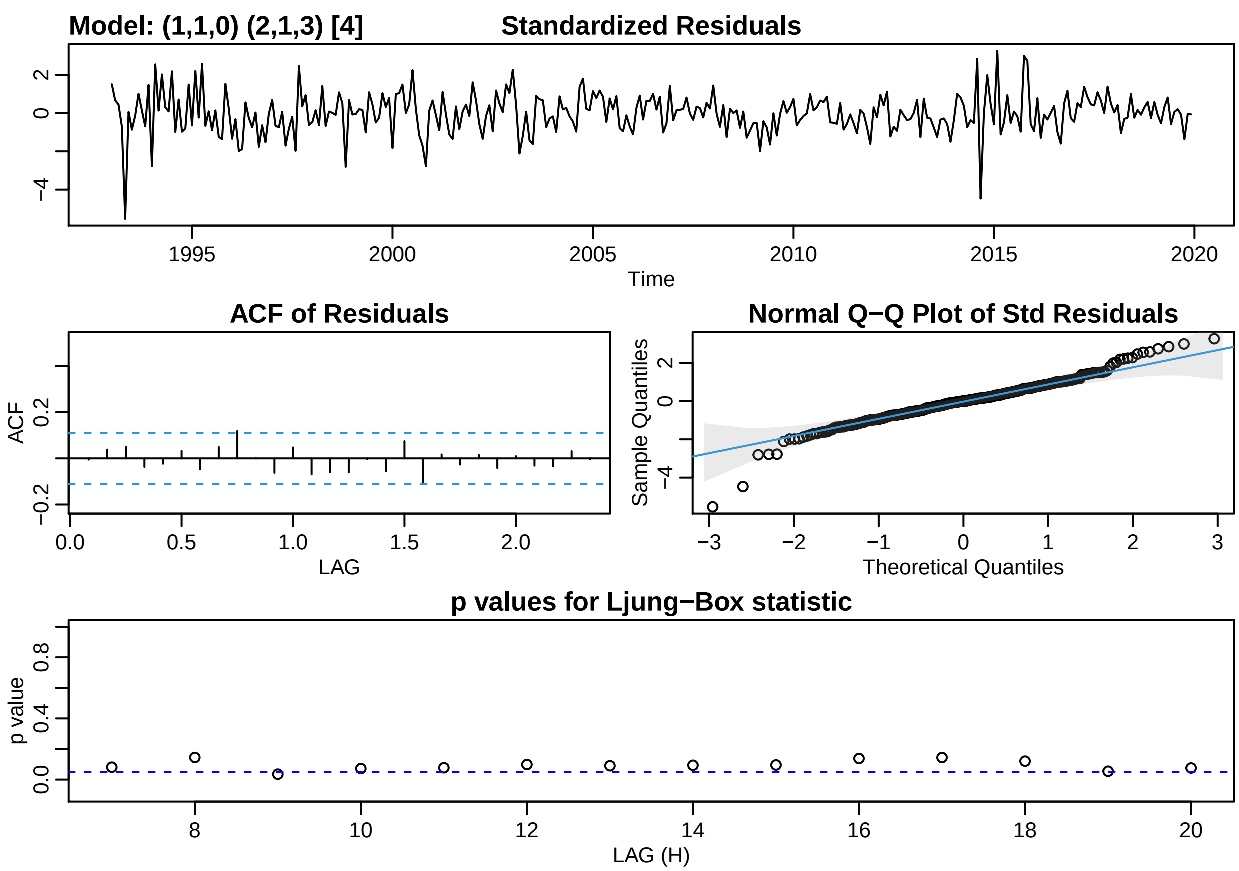

```
## $fit
##
```

```
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ma1     sar1     sar2    sar3    sma1     sma2     sma3
##       -0.3000  -0.0201  -0.9757  -0.9652  0.0289  0.0285  -0.0727  -0.6568
## s.e.   0.1451   0.1490   0.1010   0.1007  0.0992  0.0845   0.0480   0.0554
##
## sigma^2 estimated as 1.17e-05:  log likelihood = 1344.42,  aic = -2670.84
##
## $degrees_of_freedom
## [1] 311
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1   -0.3000 0.1451  -2.0673  0.0395
## ma1   -0.0201 0.1490  -0.1349  0.8928
## sar1  -0.9757 0.1010  -9.6646  0.0000
## sar2  -0.9652 0.1007  -9.5880  0.0000
## sar3   0.0289 0.0992   0.2915  0.7709
## sma1   0.0285 0.0845   0.3372  0.7362
## sma2  -0.0727 0.0480  -1.5135  0.1312
## sma3  -0.6568 0.0554 -11.8555  0.0000
##
## $AIC
## [1] -8.294544
##
## $AICc
## [1] -8.293115
##
## $BIC
## [1] -8.189305
```

```r
sarima(log_ts_dat, p = 1, d = 1, q = 0, P = 2, D = 1, Q = 3, S = 4) #AICc: -8.305847
```

```
## initial  value -4.081256
## iter   2 value -4.591308
## iter   3 value -4.920286
## iter   4 value -5.126698
## iter   5 value -5.153228
## iter   6 value -5.381005
## iter   7 value -5.431987
## iter   8 value -5.560837
## iter   9 value -5.583815
## iter  10 value -5.645621
## iter  11 value -5.665060
## iter  12 value -5.672307
## iter  13 value -5.677953
## iter  14 value -5.679613
## iter  15 value -5.680558
## iter  16 value -5.680821
## iter  17 value -5.683850
```

```
## iter  18 value -5.684054
## iter  19 value -5.684066
## iter  20 value -5.684067
## iter  20 value -5.684067
## iter  20 value -5.684067
## final  value -5.684067
## converged
## initial  value -5.621352
## iter   2 value -5.621695
## iter   3 value -5.625546
## iter   4 value -5.627344
## iter   5 value -5.628543
## iter   6 value -5.630693
## iter   7 value -5.632693
## iter   8 value -5.633256
## iter   9 value -5.633272
## iter  10 value -5.633279
## iter  10 value -5.633279
## iter  10 value -5.633279
## final  value -5.633279
## converged
```




```
## $fit
##
## Call:
```

```
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     sar1     sar2    sma1     sma2     sma3
##       -0.3164  -1.005  -0.9945  0.0478  -0.0712  -0.6575
## s.e.   0.0549   0.006   0.0032  0.0499   0.0480   0.0545
##
## sigma^2 estimated as 1.171e-05:  log likelihood = 1344.37,  aic = -2674.75
##
## $degrees_of_freedom
## [1] 313
##
## $ttable
##      Estimate     SE   t.value p.value
## ar1   -0.3164 0.0549   -5.7650  0.0000
## sar1  -1.0050 0.0060 -168.0239  0.0000
## sar2  -0.9945 0.0032 -312.9062  0.0000
## sma1   0.0478 0.0499    0.9596  0.3380
## sma2  -0.0712 0.0480   -1.4841  0.1388
## sma3  -0.6575 0.0545  -12.0661  0.0000
##
## $AIC
## [1] -8.306675
##
## $AICc
## [1] -8.305847
##
## $BIC
## [1] -8.224823
```

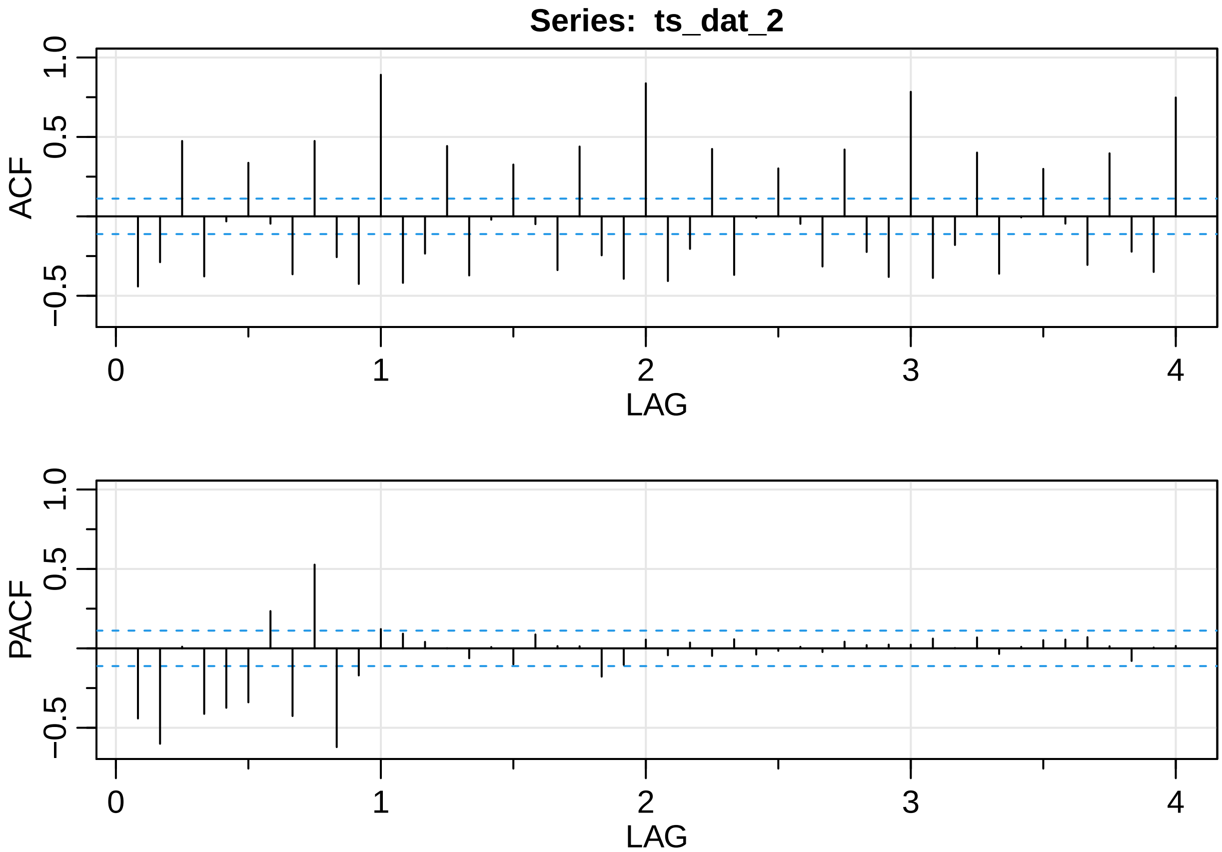
What if the patterns are bi-monthly?

```
ts_dat_2 <- diff(f_ts_dat, 2)
kpss.test(ts_dat_2) # Again, big enough to call stationary
```

```
## Warning in kpss.test(ts_dat_2): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts_dat_2
## KPSS Level = 0.037237, Truncation lag parameter = 5, p-value = 0.1
```

```
acf2(ts_dat_2)
```



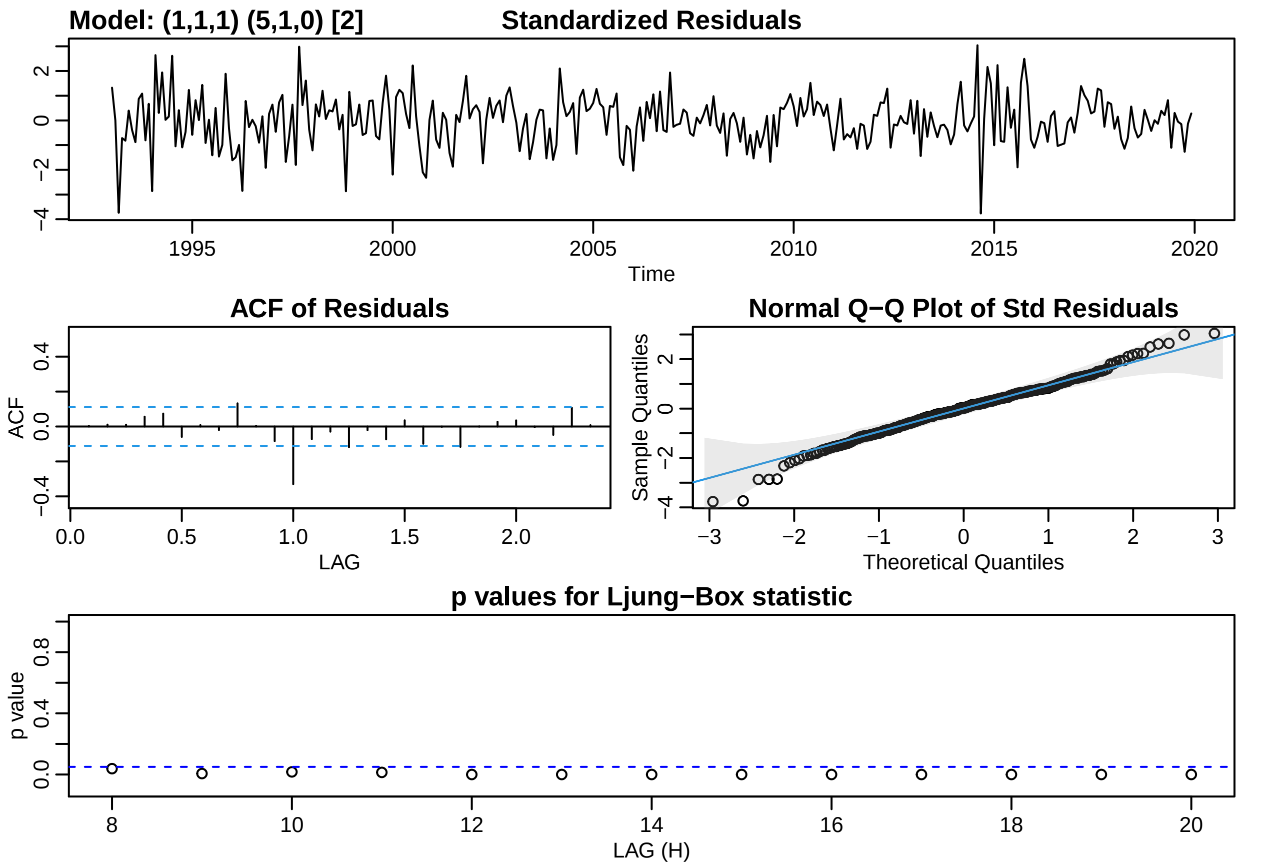

```
##       [,1]  [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8] [,9] [,10] [,11] [,12]
## ACF  -0.44 -0.29 0.47 -0.38 -0.03  0.34 -0.05 -0.36 0.47 -0.26 -0.43  0.89
## PACF -0.44 -0.60 0.01 -0.41 -0.37 -0.34  0.23 -0.43 0.53 -0.62 -0.17  0.12
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF  -0.42 -0.23  0.44 -0.37 -0.02  0.33 -0.05 -0.34  0.44 -0.25 -0.39  0.84
## PACF  0.09  0.04  0.00 -0.06  0.01 -0.10  0.09  0.01  0.01 -0.18 -0.10  0.06
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF  -0.41 -0.20  0.42 -0.37 -0.01  0.30 -0.05 -0.32  0.42 -0.22 -0.38  0.78
## PACF -0.04  0.04 -0.05  0.06 -0.04 -0.02  0.01 -0.02  0.04  0.02  0.02  0.02
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.39 -0.18  0.40 -0.36 -0.01  0.30 -0.05 -0.31  0.40 -0.22 -0.35  0.75
## PACF  0.06  0.00  0.07 -0.04  0.01  0.05  0.06  0.07  0.01 -0.08  0.01  0.02
```

Again, this logic is a bit tricky Seasonal: ACF tails off, PACF cuts off -> P=5 Lags h = 1 One or both tail off, p=q=1?

```
sarima(log_ts_dat, p = 1, d = 1, q = 1, P = 5, D = 1, Q = 0, S = 2) #AICc -8.122763, ttable says to rem
```

```
## initial  value -4.254801
## iter   2 value -4.777723
## iter   3 value -4.998184
## iter   4 value -5.122359
## iter   5 value -5.280841
## iter   6 value -5.353363
## iter   7 value -5.497746
```

```
## iter   8 value -5.510340
## iter   9 value -5.533942
## iter  10 value -5.554532
## iter  11 value -5.561559
## iter  12 value -5.562301
## iter  13 value -5.562840
## iter  14 value -5.563453
## iter  15 value -5.563664
## iter  16 value -5.564776
## iter  17 value -5.564946
## iter  18 value -5.564951
## iter  19 value -5.564994
## iter  20 value -5.565042
## iter  21 value -5.565167
## iter  22 value -5.565374
## iter  23 value -5.565675
## iter  24 value -5.565763
## iter  25 value -5.565872
## iter  26 value -5.565883
## iter  27 value -5.565884
## iter  27 value -5.565884
## iter  27 value -5.565884
## final  value -5.565884
## converged
## initial  value -5.517183
## iter   2 value -5.517320
## iter   3 value -5.517601
## iter   4 value -5.517659
## iter   5 value -5.517966
## iter   6 value -5.518091
## iter   7 value -5.518295
## iter   8 value -5.518338
## iter   9 value -5.518383
## iter  10 value -5.518423
## iter  11 value -5.518431
## iter  12 value -5.518435
## iter  13 value -5.518442
## iter  14 value -5.518446
## iter  15 value -5.518448
## iter  16 value -5.518450
## iter  16 value -5.518450
## final  value -5.518450
## converged
```



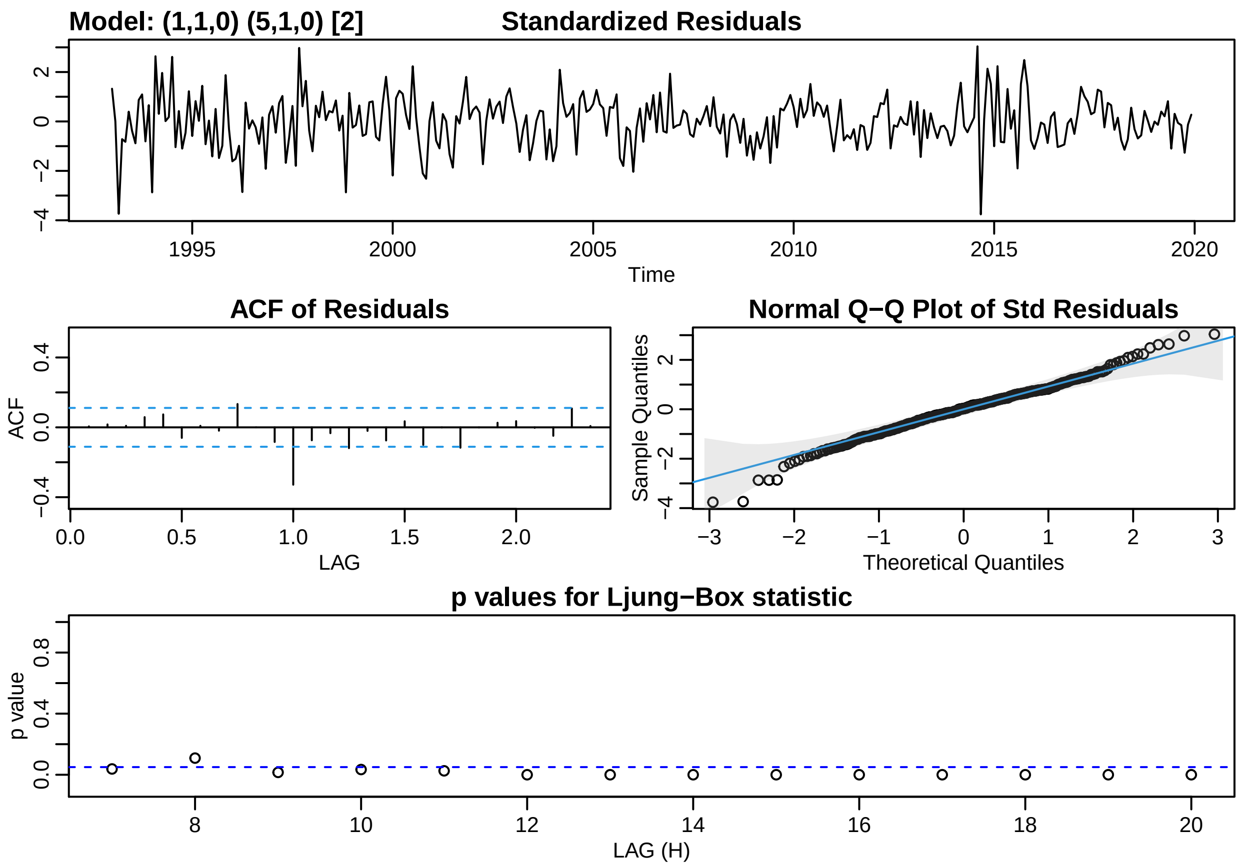

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     ma1     sar1     sar2     sar3     sar4     sar5
##       -0.3247  0.0303  -0.9435  -0.9627  -0.9015  -0.9303  -0.8969
## s.e.   0.1796  0.1846   0.0255   0.0255   0.0313   0.0239   0.0229
##
## sigma^2 estimated as 1.513e-05:  log likelihood = 1315.94,  aic = -2615.89
##
## $degrees_of_freedom
## [1] 314
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1   -0.3247 0.1796  -1.8078  0.0716
## ma1    0.0303 0.1846   0.1644  0.8695
## sar1  -0.9435 0.0255 -36.9329  0.0000
## sar2  -0.9627 0.0255 -37.7129  0.0000
## sar3  -0.9015 0.0313 -28.7829  0.0000
## sar4  -0.9303 0.0239 -38.9392  0.0000
## sar5  -0.8969 0.0229 -39.1799  0.0000
```

```
##
## $AIC
## [1] -8.12387
##
## $AICc
## [1] -8.122763
##
## $BIC
## [1] -8.03017
```

```
sarima(log_ts_dat, p = 1, d = 1, q = 0, P = 5, D = 1, Q = 0, S = 2) #AICc -8.129134
```

```
## initial  value -4.254801
## iter   2 value -4.669800
## iter   3 value -4.994879
## iter   4 value -5.084156
## iter   5 value -5.084377
## iter   6 value -5.182845
## iter   7 value -5.410068
## iter   8 value -5.459143
## iter   9 value -5.521054
## iter  10 value -5.547250
## iter  11 value -5.556718
## iter  12 value -5.559060
## iter  13 value -5.562707
## iter  14 value -5.564062
## iter  15 value -5.564571
## iter  16 value -5.565220
## iter  17 value -5.565574
## iter  18 value -5.565695
## iter  19 value -5.565800
## iter  20 value -5.565812
## iter  21 value -5.565827
## iter  22 value -5.565833
## iter  22 value -5.565833
## iter  22 value -5.565833
## final  value -5.565833
## converged
## initial  value -5.517132
## iter   2 value -5.517239
## iter   3 value -5.517552
## iter   4 value -5.517769
## iter   5 value -5.517954
## iter   6 value -5.518092
## iter   7 value -5.518183
## iter   8 value -5.518274
## iter   9 value -5.518324
## iter  10 value -5.518360
## iter  11 value -5.518376
## iter  12 value -5.518380
## iter  13 value -5.518385
## iter  14 value -5.518390
## iter  14 value -5.518390
## final  value -5.518390
```

```
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     sar1     sar2     sar3     sar4     sar5
##       -0.2964  -0.9410  -0.9616  -0.8993  -0.9299  -0.8951
## s.e.   0.0564   0.0239   0.0247   0.0302   0.0237   0.0223
##
## sigma^2 estimated as 1.514e-05:  log likelihood = 1315.92,  aic = -2617.85
##
## $degrees_of_freedom
## [1] 315
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1   -0.2964 0.0564  -5.2554       0
## sar1  -0.9410 0.0239 -39.4169       0
## sar2  -0.9616 0.0247 -38.8950       0
## sar3  -0.8993 0.0302 -29.7807       0
```

```
## sar4  -0.9299 0.0237 -39.3001       0
## sar5  -0.8951 0.0223 -40.0991       0
##
## $AIC
## [1] -8.129962
##
## $AICc
## [1] -8.129134
##
## $BIC
## [1] -8.047974
```

The residuals look better here, but Ljung-Box is worse