Employment Analysis

Leala Darby

03/11/2020

First load all required packages:

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

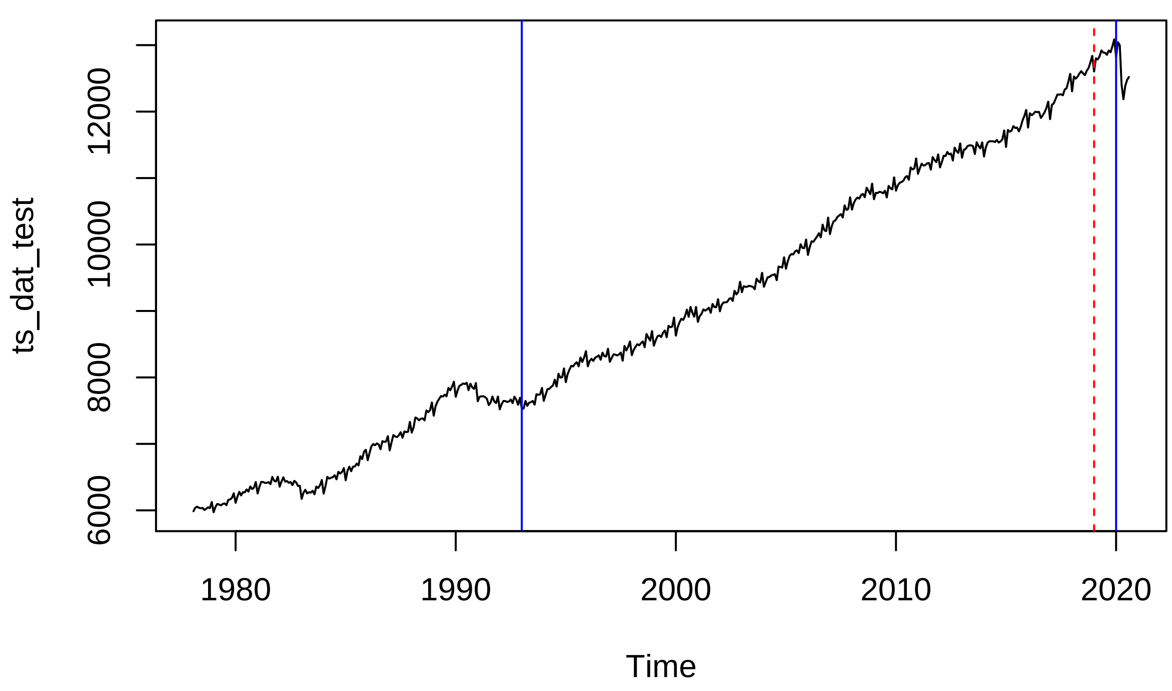
Load in the data:

```r
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. The blue lines are visually detected structural breakpoints - contextual reasoning is PC surge in the 90s and COVID-19. The red line indicates the training/test split.

```r
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")
abline(v = 2019, col = "red", lty = 2)
```



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

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

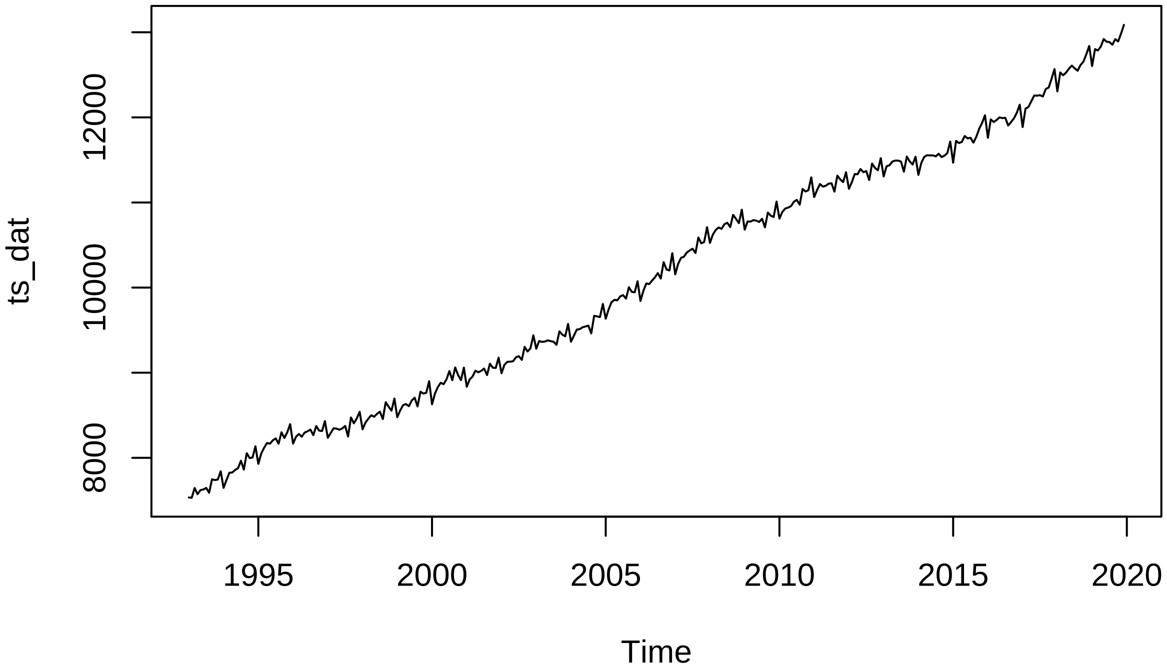
```
##     Observation.times Time.series.values
## 180            Jan-93             7533.7
```

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

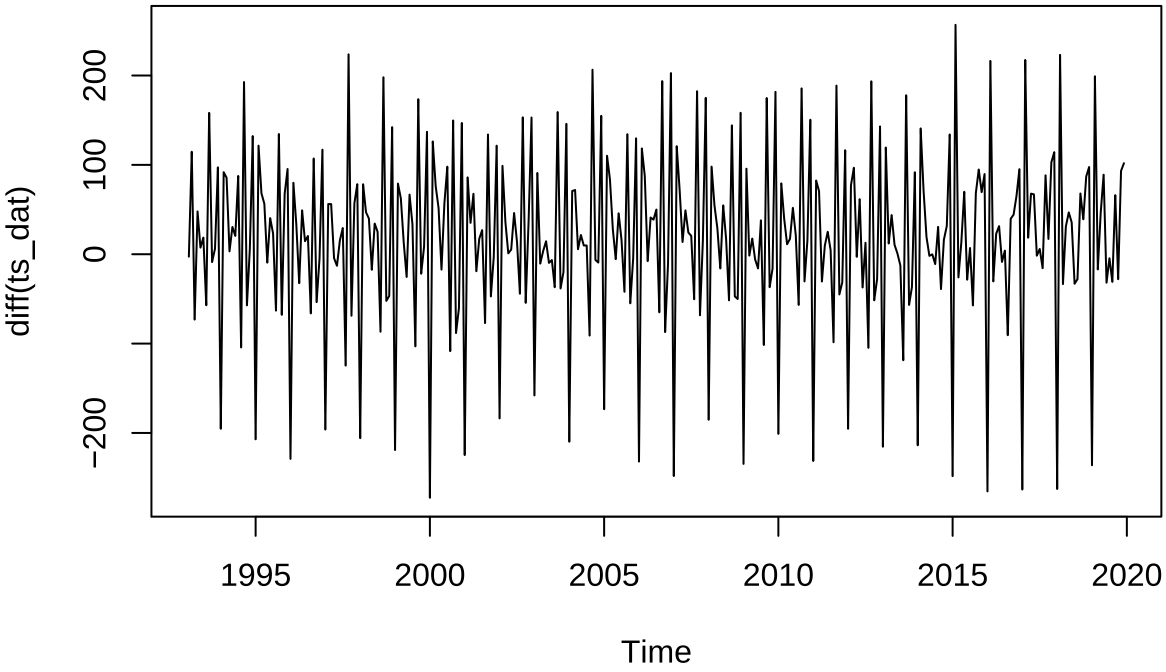
```
##     Observation.times Time.series.values
## 503            Dec-19            13087.1
```

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)) # We are not actually taking the difference yet!
```



```
# This 2nd plot is just to help look for trends in variance.
```

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
```

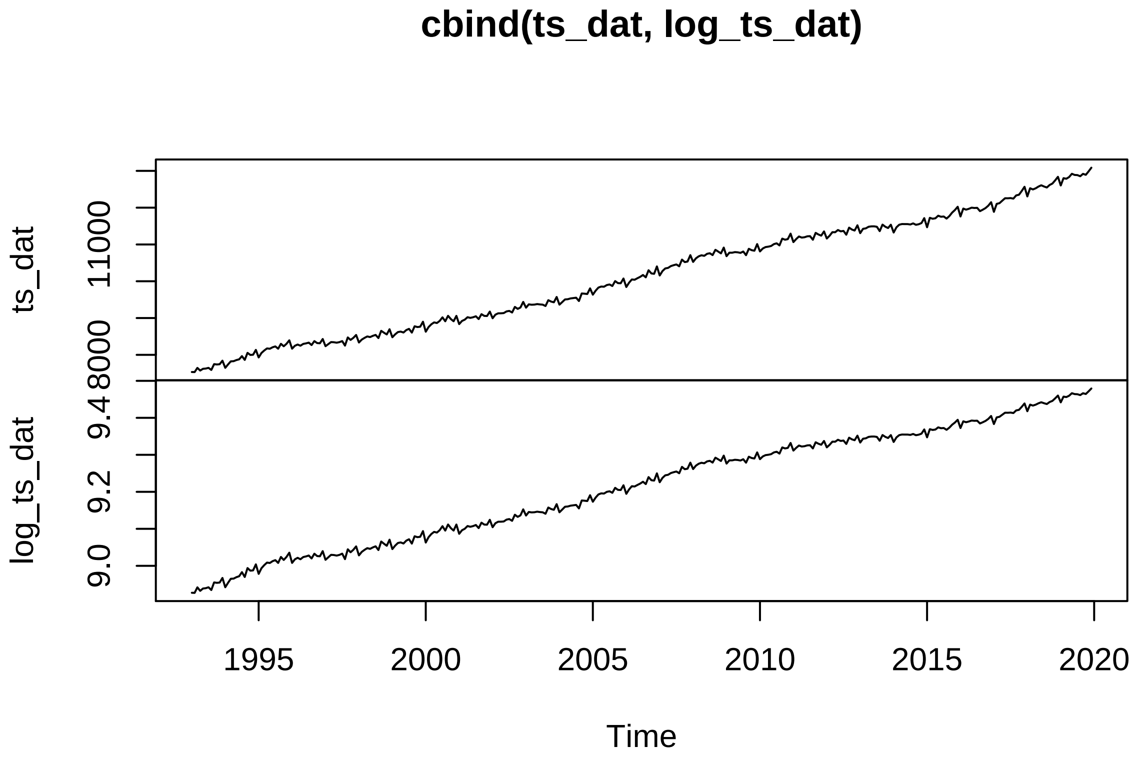
```
Group <- c(rep(1,81), rep(2, 81), rep(3, 81), rep(4, 81))
leveneTest(ts_dat, Group)
```

```
## 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:

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



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

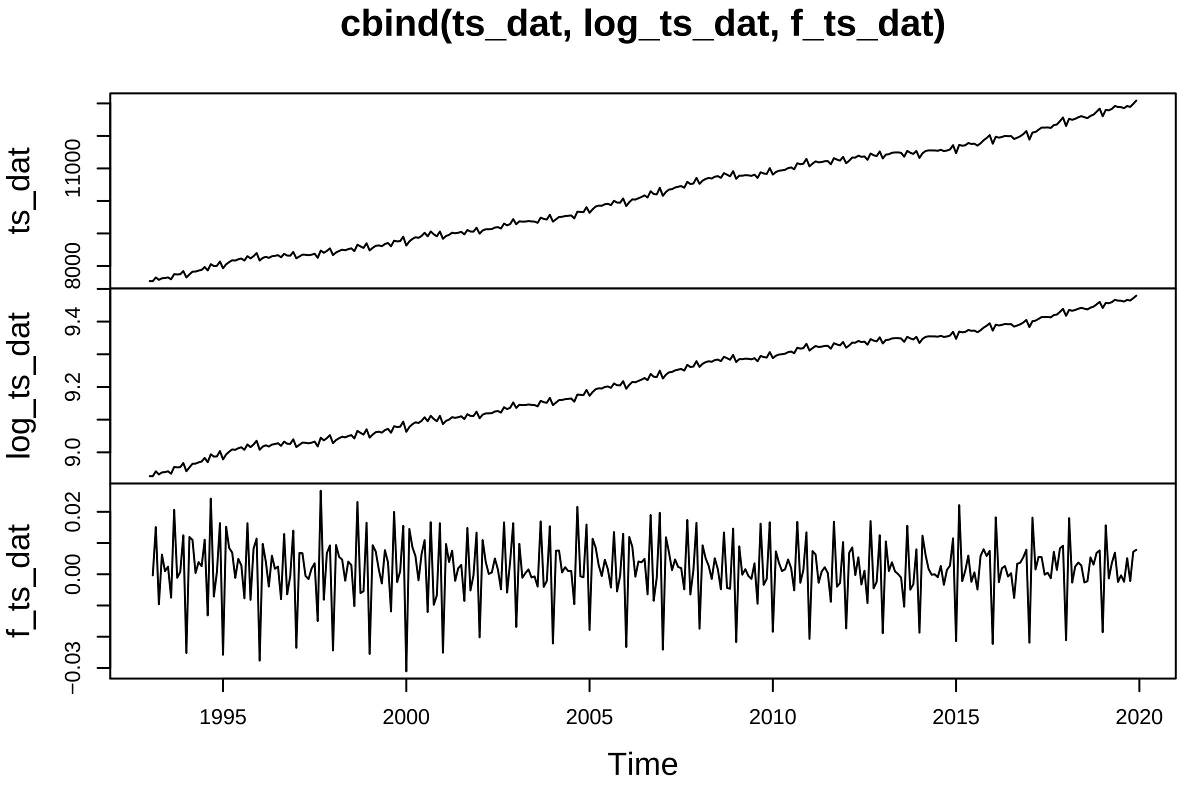
```
## Warning in leveneTest.default(log_ts_dat, Group): Group coerced to factor.
```

```
## 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.

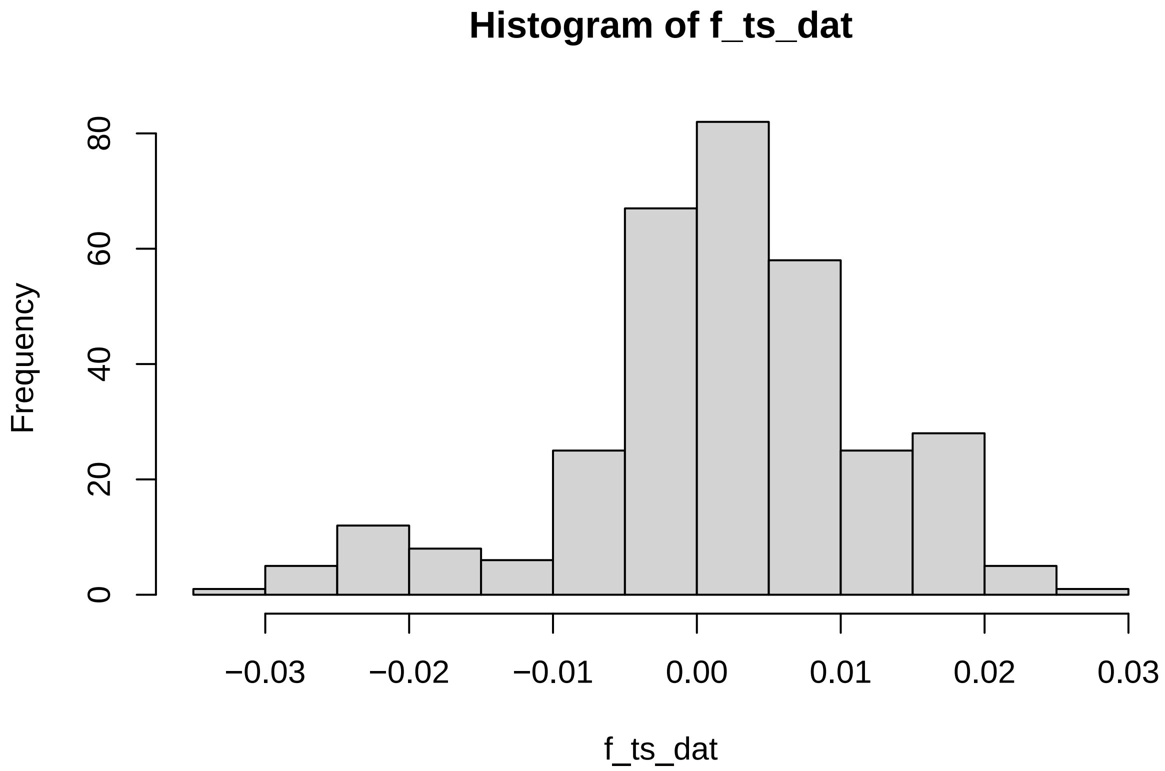
Next, to reduce the trend in mean, apply differencing of 1 lag to our TS with stabilised variance:

```
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)
```



```
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:

```
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
```

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

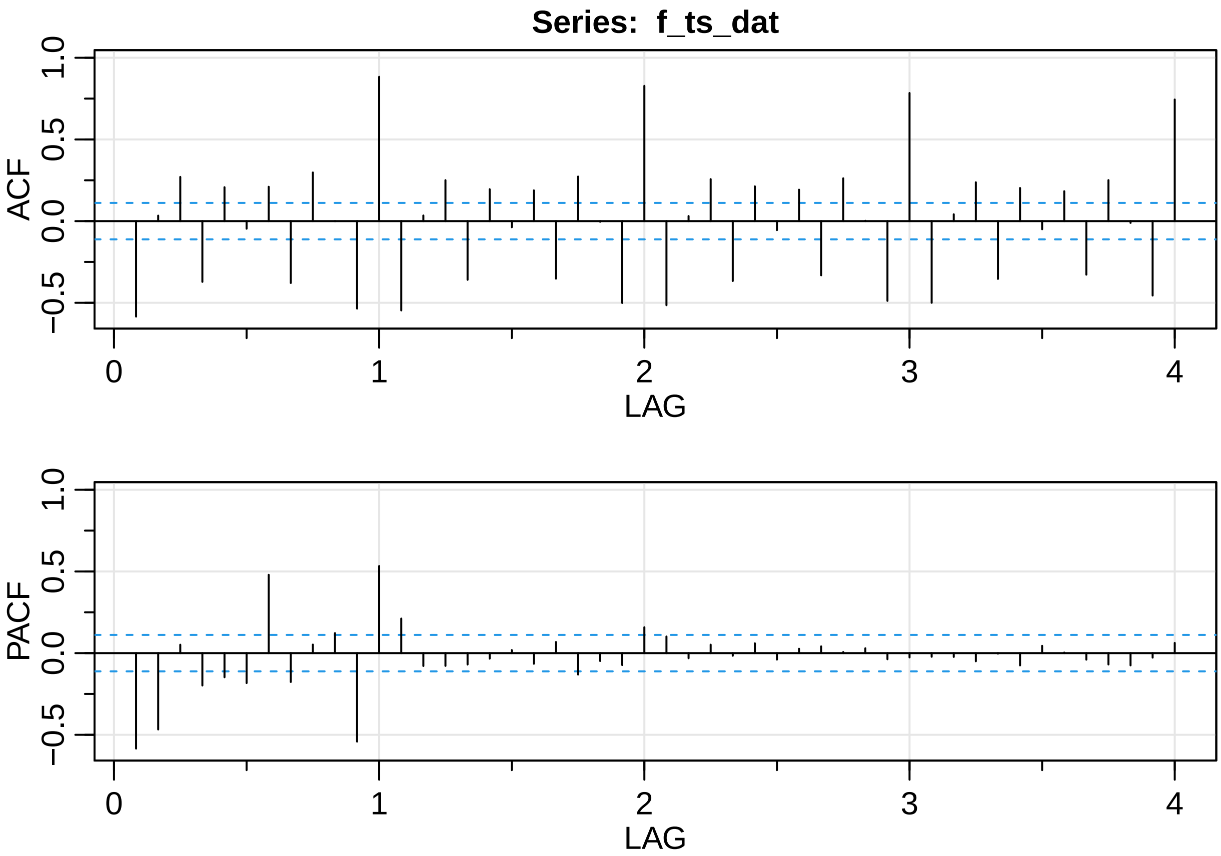
```
## Warning in kpss.test(f_ts_dat): p-value greater than printed p-value
```

```
##
##  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.

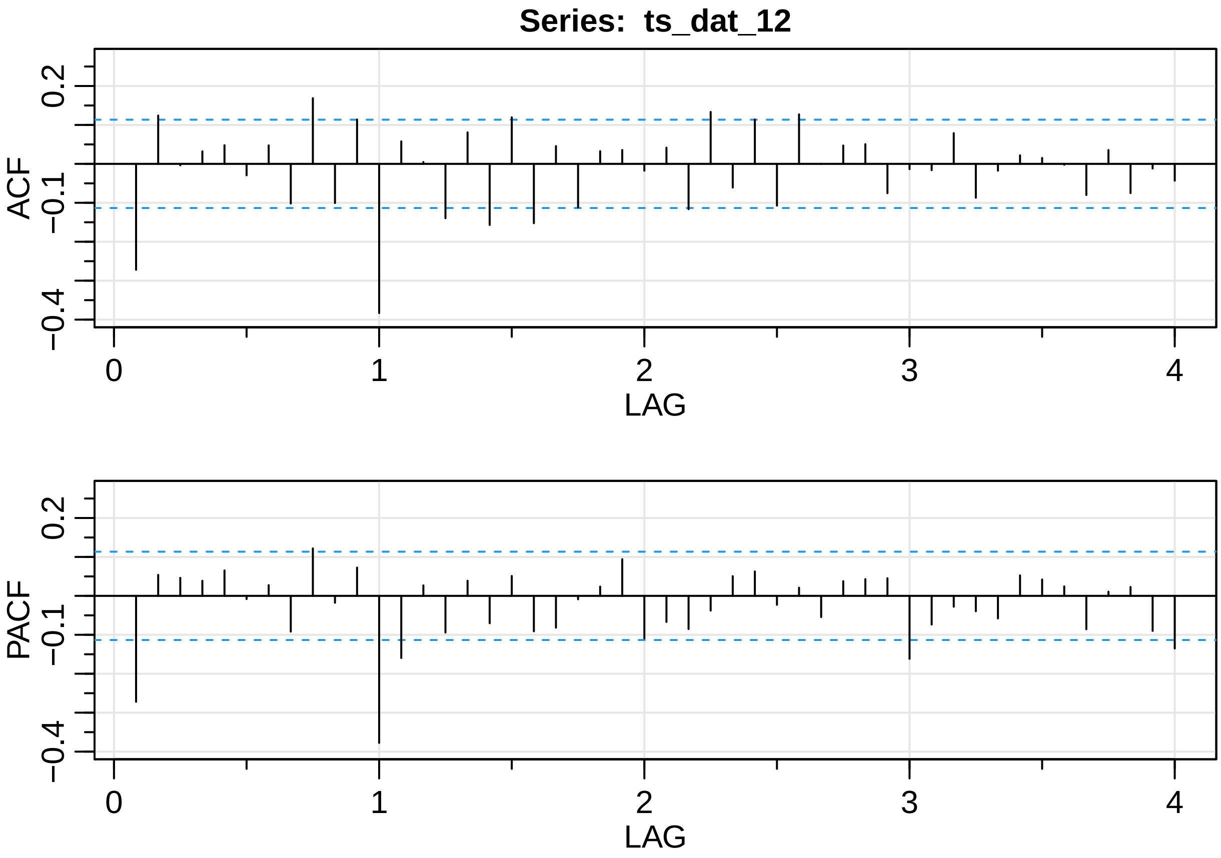
The data being monthly and the ACF plot having its highest peaks at lags $h = 12, 24, 36, 48$ implies a seasonal trend of 12 would be a good choice. Slow decay over these four peaks suggests there is a difference between seasons. To remove this trend, difference the ts on the seasonal lag:

```
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)
```



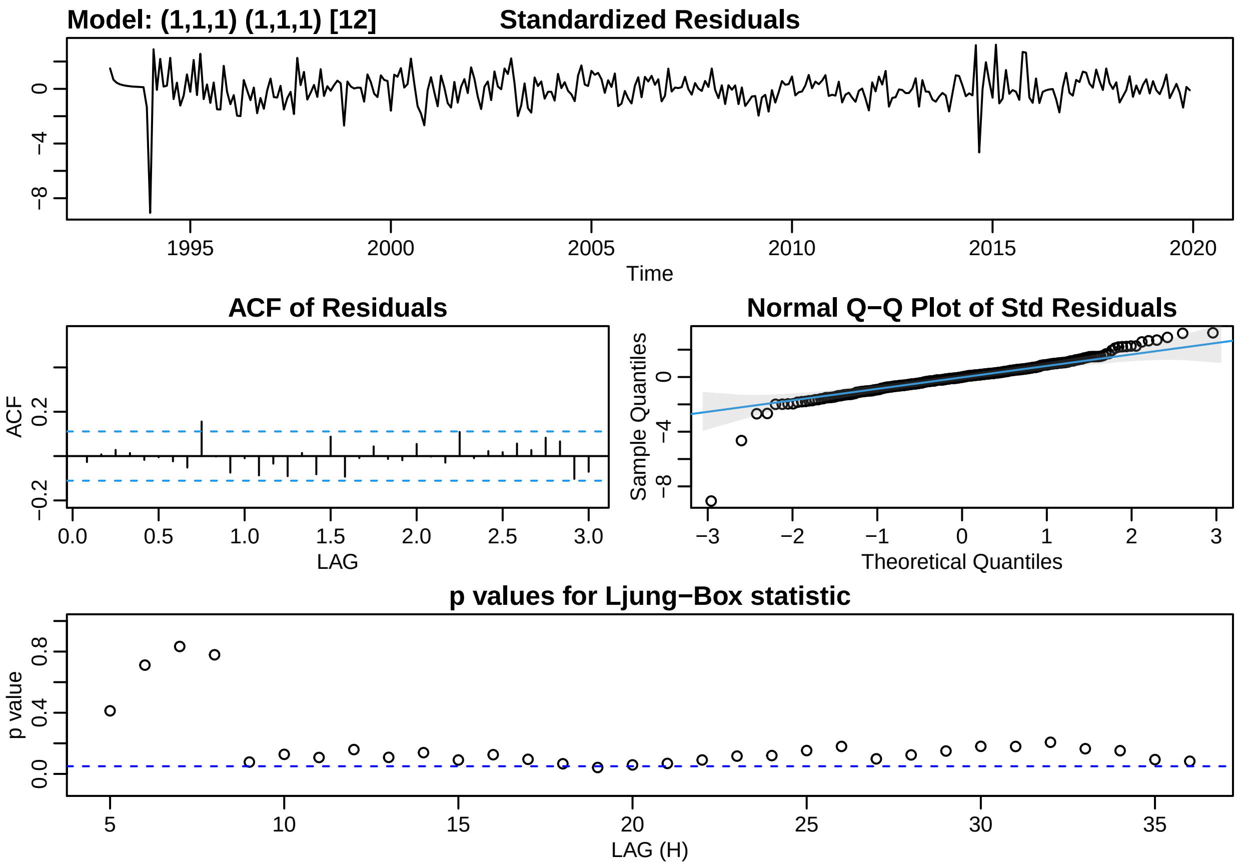
```
##       [,1] [,2] [,3] [,4] [,5]  [,6] [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  -0.27 0.12 0.00 0.03 0.05 -0.03 0.05 -0.10 0.17 -0.10  0.11 -0.38  0.06
## PACF -0.27 0.05 0.05 0.04 0.07 -0.01 0.03 -0.09 0.12 -0.02  0.07 -0.38 -0.16
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,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 $q = 0$ Again $0 \leq p \leq 1$ and $0 \leq q \leq 1$

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

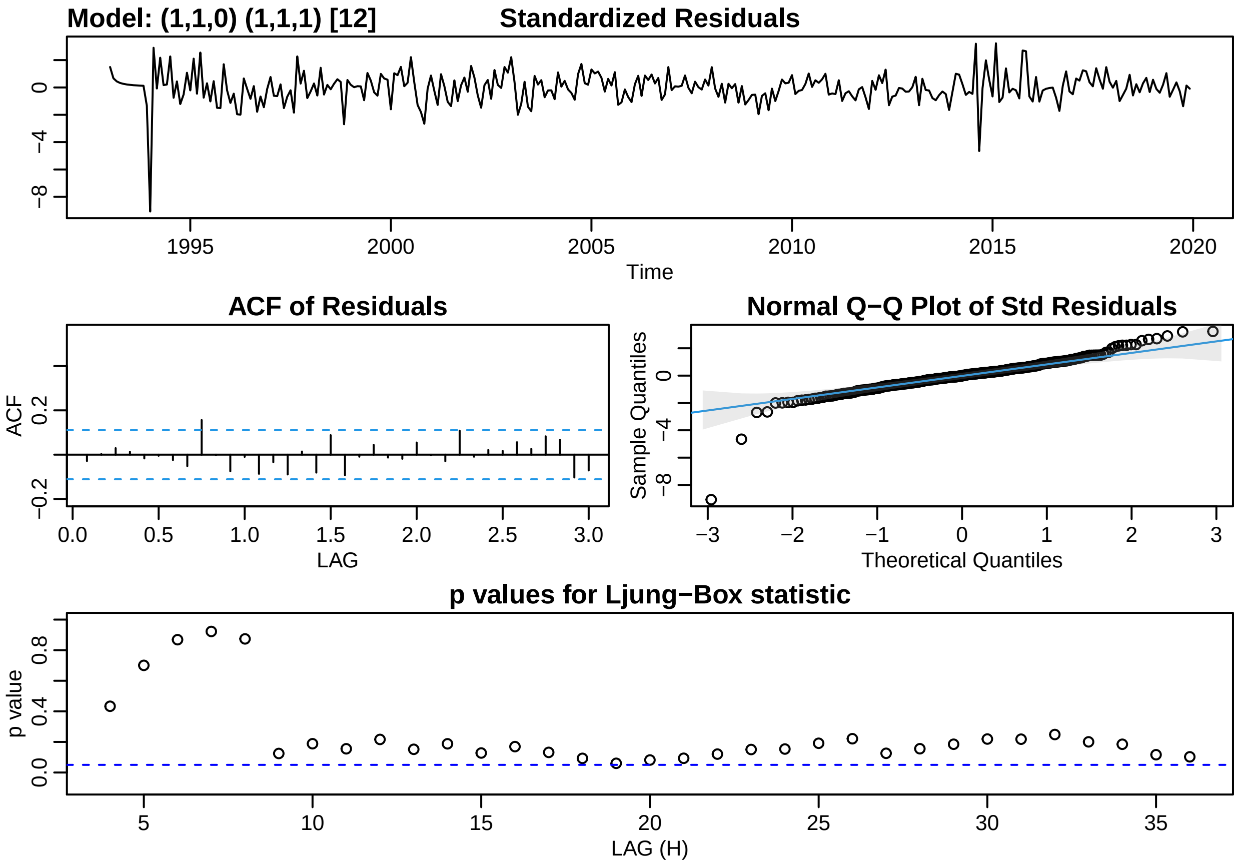
```
## initial  value -5.504137
## iter   2 value -5.599083
## iter   3 value -5.658866
## iter   4 value -5.662290
## iter   5 value -5.666833
## iter   6 value -5.671616
## iter   7 value -5.673051
## iter   8 value -5.673605
## iter   9 value -5.673704
## iter  10 value -5.673742
## iter  11 value -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
```

```
# ttable says ma1 coeff has highest p-value. removing this (model trimming):
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
```

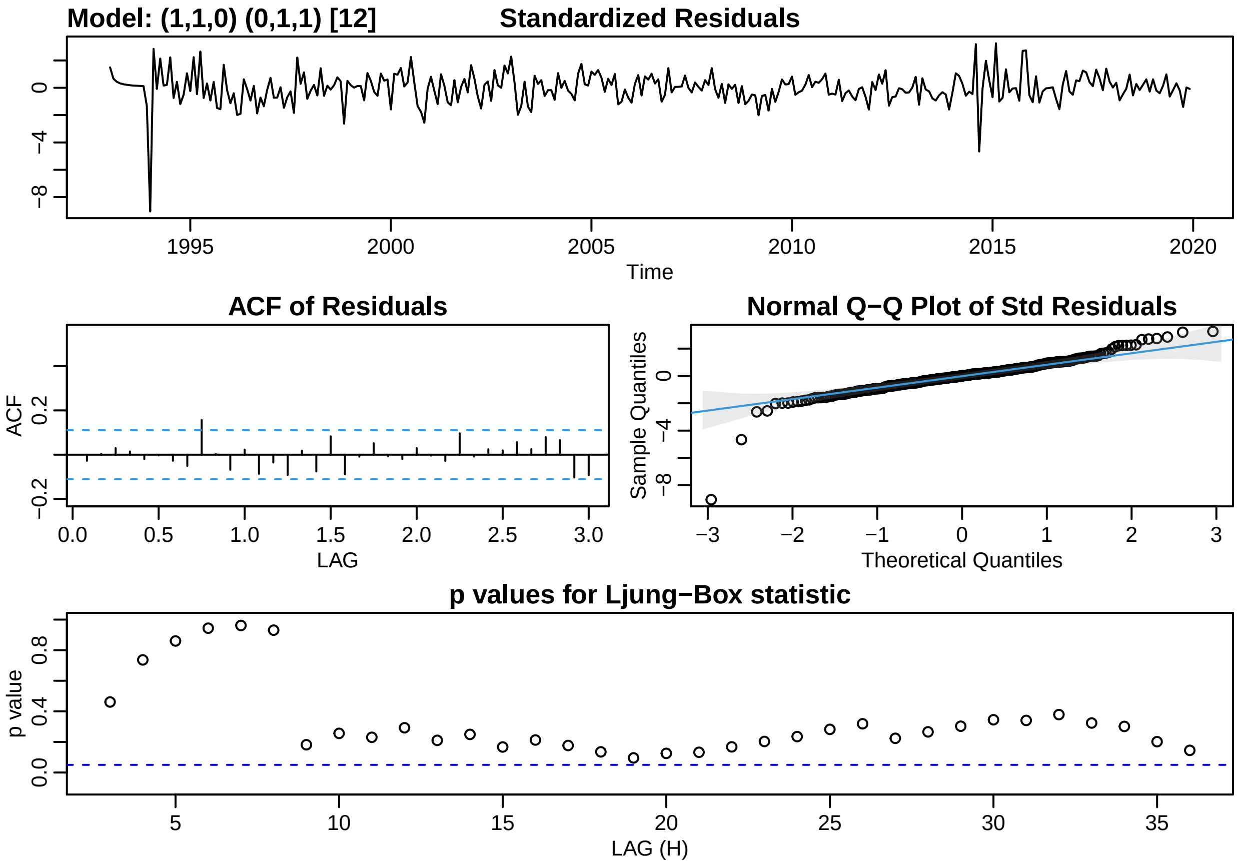


```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1    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
```

We see a couple of outliers - pinpoint what these points are. The ljung-Box statistic is passable at lag 20 or 30.

Now fit the model, make predictions, assess their accuracy and report the final model.

Future work.