# Initial Project Report

Emily Linebarger

5/11/2022

# Initial Project Report

# Emily Linebarger

## Introduction

For my time series analysis, I propose to investigate how the U.S.'s international investment position changed during the COVID-19 pandemic. The dataset I chose from the federal reserve should show whether the US took on more international debt during the pandemic, or if assets the US held grew in value relative to other countries' holdings. My background is in economics and international development, so I am always interested in how world events change capital flows.

My datasets include data on international investments from the US Bureau of Economic Analysis, US GDP, an aggregate of all US debt, and international airline travel. I hope that the first three datasets give a good picture of how capital flows from the US changed over the pandemic, as well as some supporting financial information that might hint at drivers. I hope the fourth dataset on international airline travel can serve as a proxy for how international commerce shut down during the first part of 2020.

## Exploratory Analysis

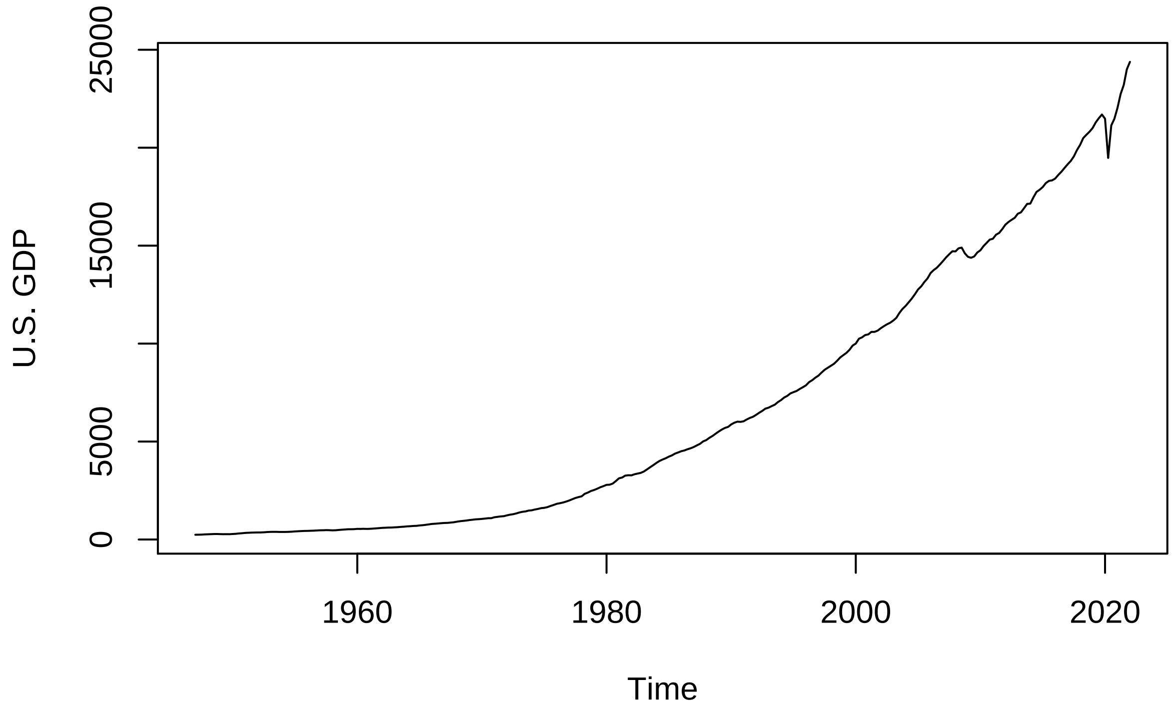
### U.S. GDP

** Data Cleaning **

```
# This is the raw dataset downloaded from https://fred.stlouisfed.org/series/GDP on 5/11/2022.
dt <- fread("C:/Users/eklin/Downloads/GDP.csv")
```

There are 0 NA observations in the series. ** Plotting and Analysis **

```
gdp <- ts(dt$GDP, start = c(1947, 1), frequency = 4)
plot(gdp, ylab = "U.S. GDP")
```




This data is certainly not stationary! It's constantly increasing. However, there doesn't seem to be any strange values other than drops around 2008 and 2020, which were notable recessions.

** Evaluate stationarity with a hypothesis test **

```r
# Null hypothesis: data is stationary
# Alternative hypothesis: data is non-stationary
kpss.test(gdp)
```

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

```
##
##  KPSS Test for Level Stationarity
##
## data:  gdp
## KPSS Level = 4.656, Truncation lag parameter = 5, p-value = 0.01
```

```r
# p-value is 0.01, so we reject our null hypothesis. Data is non-stationary.
gdp_diff <- diff(gdp, lag = 1)
kpss.test(gdp_diff)
```

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

```
##
##  KPSS Test for Level Stationarity
##
```

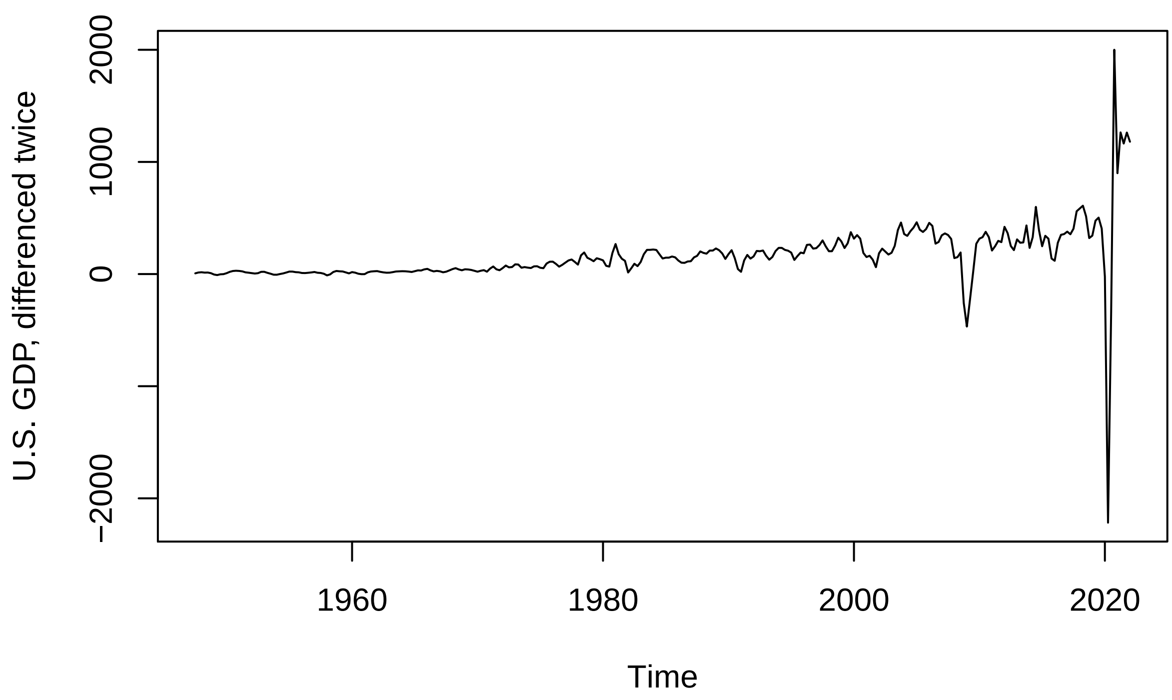
```
## data:  gdp_diff
## KPSS Level = 2.57, Truncation lag parameter = 5, p-value = 0.01
```

```
# After one lag we still don't observe stationarity. Try 2 lags.
gdp_diff <- diff(gdp, lag = 2)
kpss.test(gdp_diff)
```

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

```
##
##  KPSS Test for Level Stationarity
##
## data:  gdp_diff
## KPSS Level = 2.7413, Truncation lag parameter = 5, p-value = 0.01
```

```
plot(gdp_diff, ylab = "U.S. GDP, differenced twice")
```




This is fascinating. Even after differencing the series twice we still fail the stationarity test. It seems to be caused by the major dip that happened in 2020. I might have to truncate the series to use it for modeling, but then this would go against the very goal of my project; to see how all of these time series changed around the COVID-19 pandemic.

Let me try to take a log of this time series before differencing it, and see if that helps with stationarity.

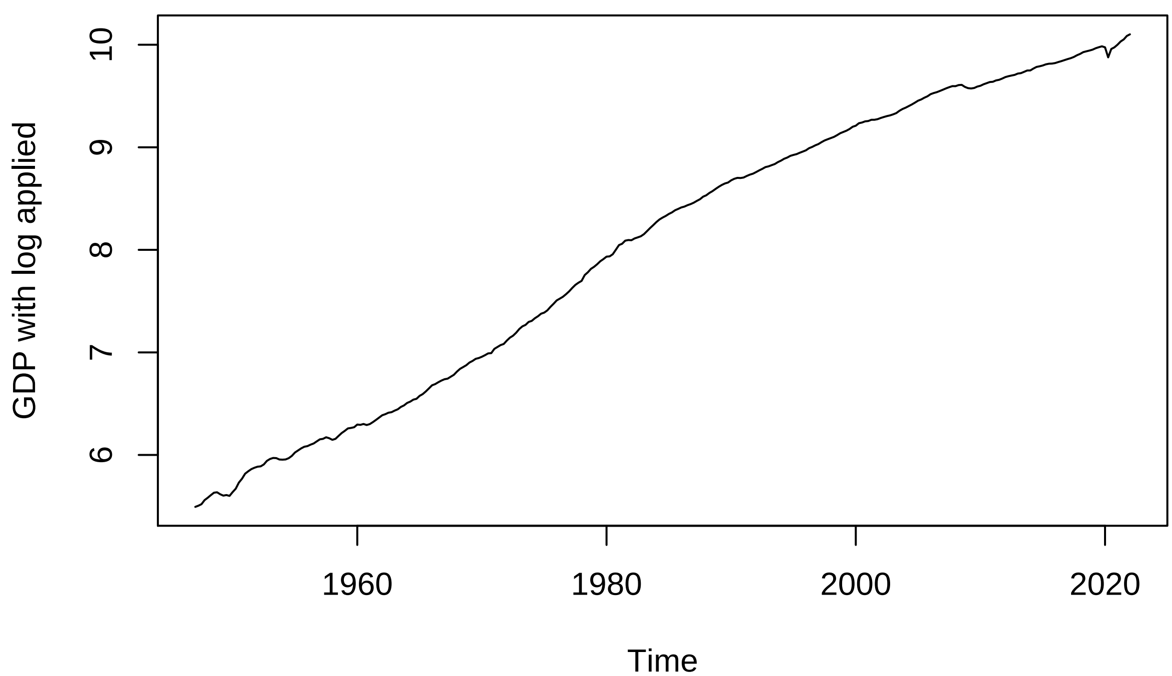
```r
gdp_log = log(gdp)
kpss.test(gdp_log)
```

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

##
##  KPSS Test for Level Stationarity
##
## data:  gdp_log
## KPSS Level = 5.1122, Truncation lag parameter = 5, p-value = 0.01
```

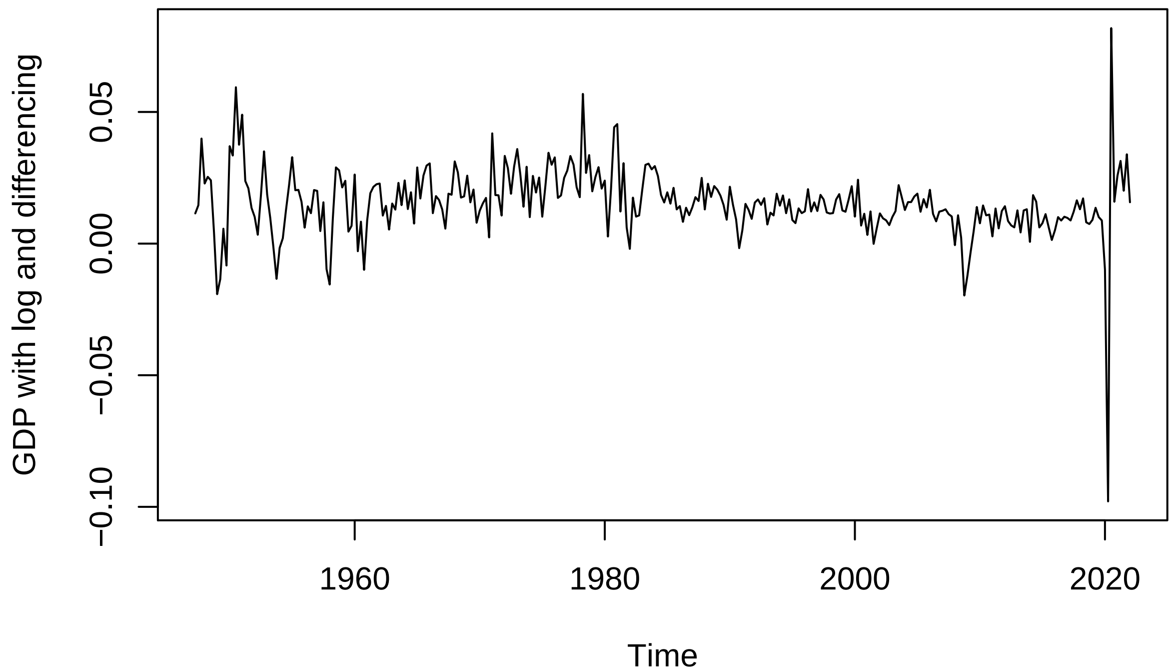
```r
# We reject the null hypothesis, so data is non-stationary without differencing.
gdp_log_diff = diff(gdp_log, lag = 1)
kpss.test(gdp_log_diff)
```

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

##
##  KPSS Test for Level Stationarity
##
## data:  gdp_log_diff
## KPSS Level = 0.96564, Truncation lag parameter = 5, p-value = 0.01
```

```r
# 0.06294 < 0.05, so we fail to reject the null hypothesis.
# This differenced, logged series is stationary!
plot(gdp_log, ylab = "GDP with log applied")
```

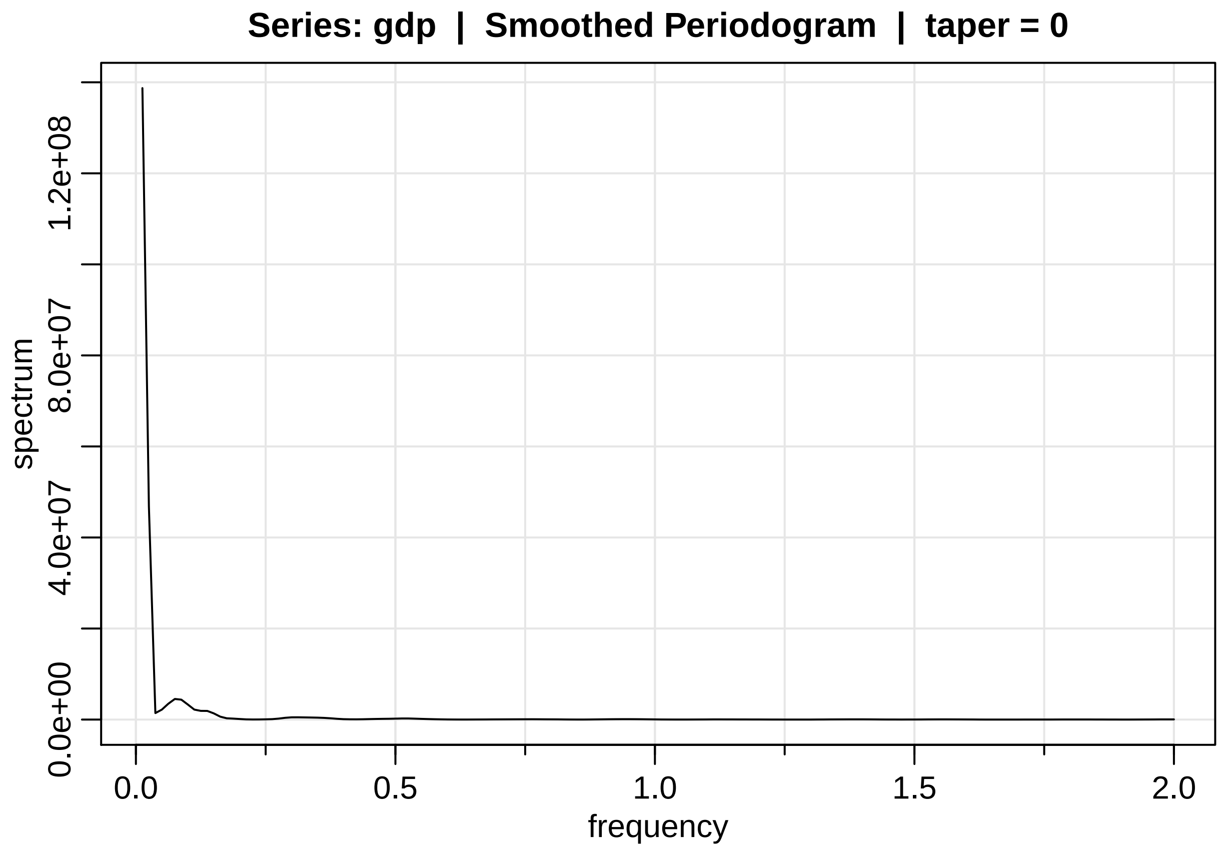


```r
plot(gdp_log_diff, ylab = "GDP with log and differencing")
```




** Evaluate Seasonality **

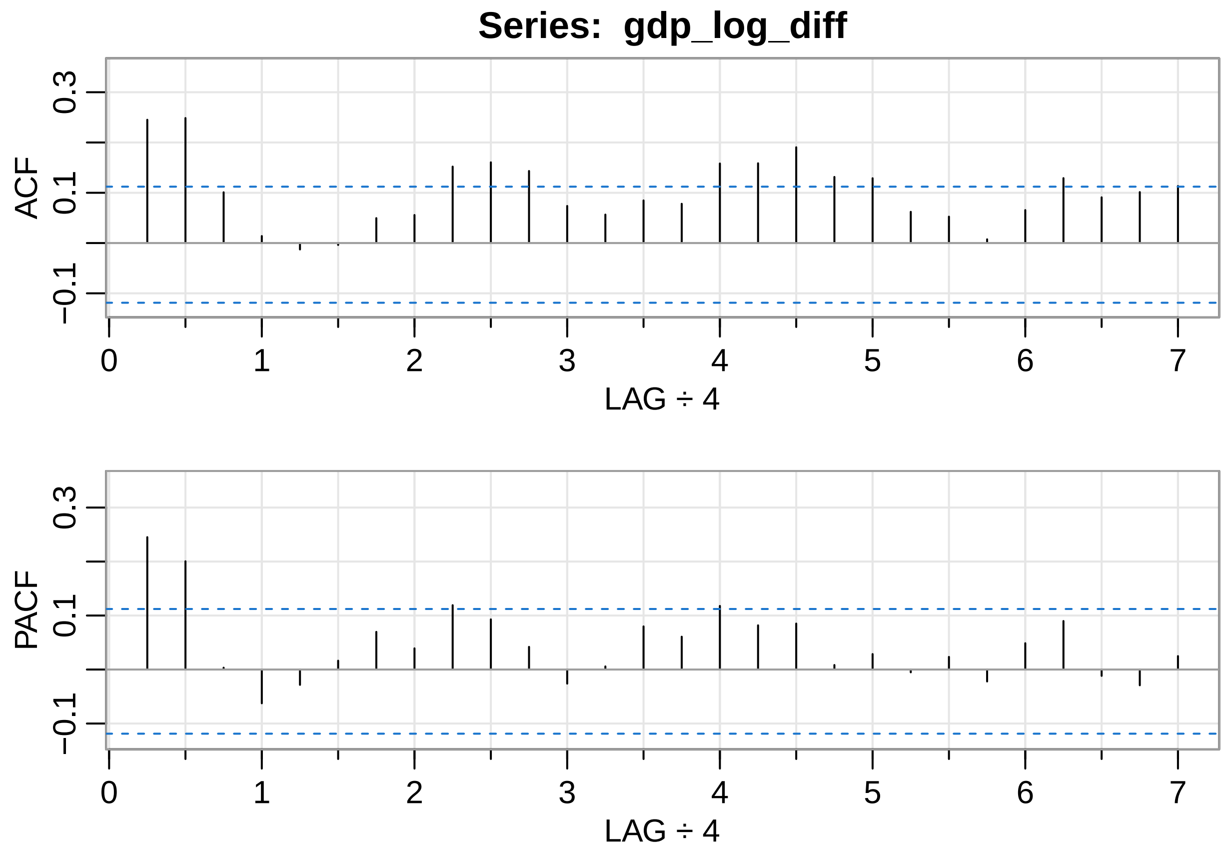
```r
gdp_log_spec = mvspec(gdp, spans = 2, detrend = TRUE)
```




Although it's already pretty clear from the plot of the data, there is no evidence of seasonality in this series. It's clear that the detrend argument is not completely working because there is a huge spike near the left axis. But other than that there is no evidence of seasonal variation.

** ACF/PACF **

```
acf2(gdp_log_diff)
```




```
##      [,1] [,2] [,3]  [,4]  [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.25 0.25  0.1  0.01 -0.01 0.00 0.05 0.06 0.15  0.16  0.14  0.07  0.06
## PACF 0.25 0.20  0.0 -0.06 -0.03 0.02 0.07 0.04 0.12  0.09  0.04 -0.03  0.01
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.08  0.08  0.16  0.16  0.19  0.13  0.13  0.06  0.05  0.01  0.07  0.13
## PACF  0.08  0.06  0.12  0.08  0.09  0.01  0.03 -0.01  0.02 -0.02  0.05  0.09
##      [,26] [,27] [,28]
## ACF   0.09  0.10  0.11
## PACF -0.01 -0.03  0.02
```

There are many significant spikes in the ACF! I might have to experiment with adding MA terms to make sure I've captured all of the variance here. I think to start I'll run two models. Model 1 will be a MA-2, AR-2 model. There are two clear, significant lags on the ACF and PACF which makes me think this could be a good fit. Then, as a comparison, I'll run a MA-5, AR-2 model, to see if adding additional MA terms captures some of the variation in the right tail of the ACF. For both of these models, I'll run them on the logged-GDP data and include one difference term.

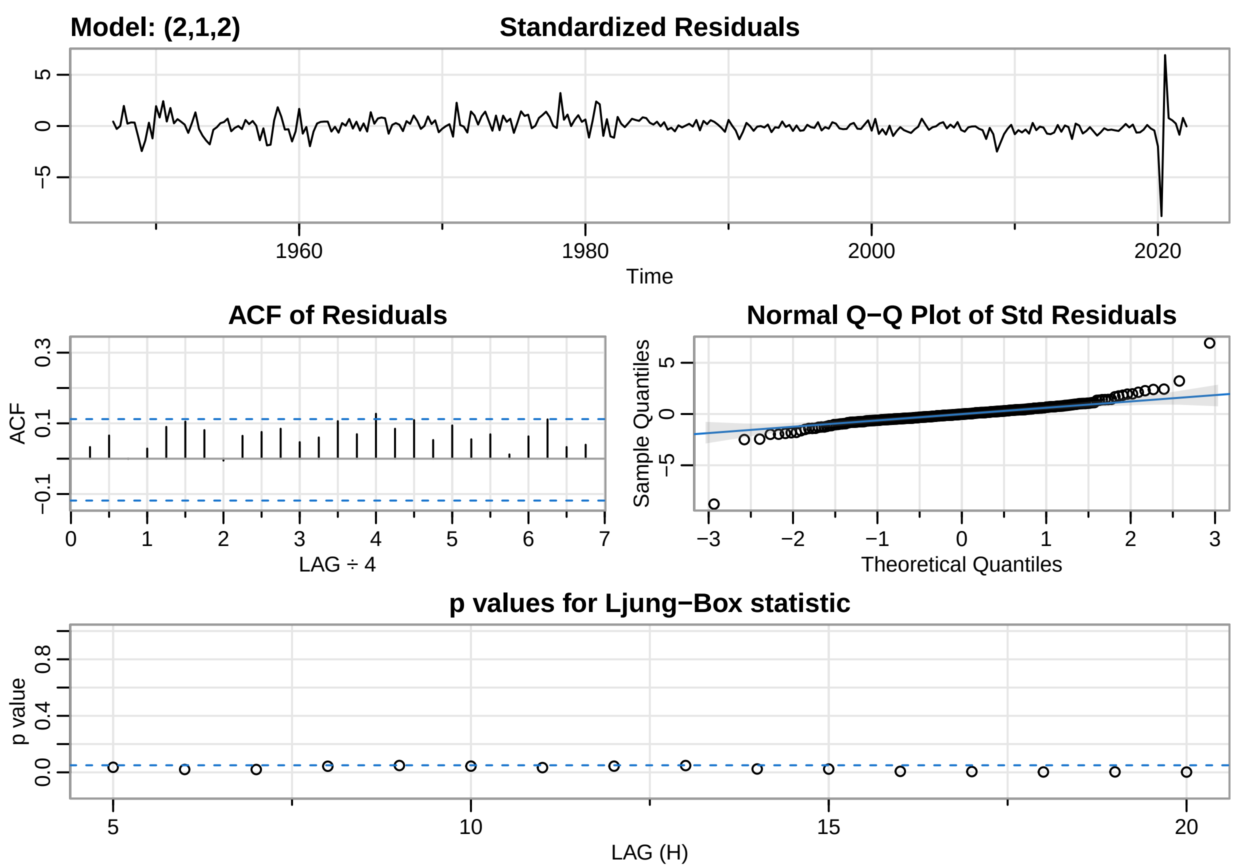
** ARIMA Modeling **

```
ar2_ma2 = sarima(gdp_log, p = 2, d = 1, q = 2)
```

```
## initial  value -4.318719
## iter   2 value -4.355755
## iter   3 value -4.369772
## iter   4 value -4.370312
```

```
## iter   5 value -4.370586
## iter   6 value -4.370692
## iter   7 value -4.371164
## iter   8 value -4.371354
## iter   9 value -4.371522
## iter  10 value -4.371652
## iter  11 value -4.371935
## iter  12 value -4.372610
## iter  13 value -4.372881
## iter  14 value -4.373026
## iter  15 value -4.373120
## iter  16 value -4.373221
## iter  17 value -4.373366
## iter  18 value -4.373380
## iter  19 value -4.373401
## iter  20 value -4.373436
## iter  21 value -4.373516
## iter  22 value -4.373645
## iter  23 value -4.373835
## iter  24 value -4.373853
## iter  25 value -4.373861
## iter  26 value -4.373864
## iter  27 value -4.373869
## iter  28 value -4.373872
## iter  29 value -4.373873
## iter  30 value -4.373874
## iter  31 value -4.373879
## iter  32 value -4.373890
## iter  33 value -4.373922
## iter  34 value -4.373987
## iter  35 value -4.374029
## iter  36 value -4.374049
## iter  37 value -4.374072
## iter  38 value -4.374080
## iter  39 value -4.374090
## iter  40 value -4.374094
## iter  41 value -4.374104
## iter  42 value -4.374129
## iter  43 value -4.374258
## iter  44 value -4.374341
## iter  45 value -4.374363
## iter  46 value -4.374429
## iter  47 value -4.374461
## iter  48 value -4.374499
## iter  49 value -4.374521
## iter  50 value -4.374602
## iter  51 value -4.374752
## iter  52 value -4.375139
## iter  53 value -4.375656
## iter  54 value -4.376342
## iter  55 value -4.376682
## iter  56 value -4.377464
## iter  57 value -4.378545
## iter  58 value -4.379274
```

```
## iter  59 value -4.379520
## iter  60 value -4.379610
## iter  61 value -4.379672
## iter  62 value -4.379777
## iter  63 value -4.379830
## iter  64 value -4.379852
## iter  65 value -4.379852
## iter  65 value -4.379852
## final  value -4.379852
## converged
## initial  value -4.380222
## iter   2 value -4.380230
## iter   3 value -4.380246
## iter   4 value -4.380262
## iter   5 value -4.380269
## iter   6 value -4.380272
## iter   7 value -4.380272
## iter   8 value -4.380272
## iter   9 value -4.380273
## iter  10 value -4.380274
## iter  11 value -4.380275
## iter  12 value -4.380275
## iter  12 value -4.380275
## final  value -4.380275
## converged
```

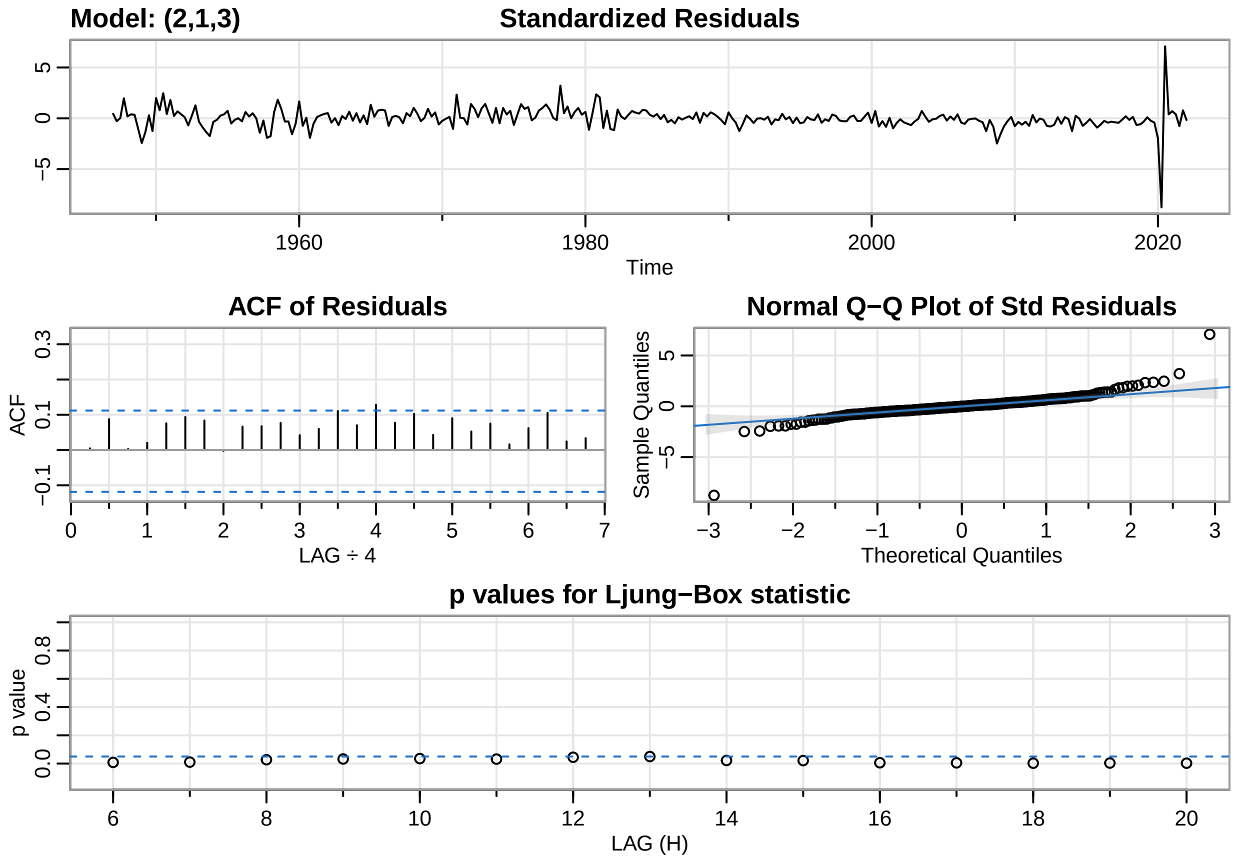


The plot of the standardized residuals looks good. There is not a clear trend here, even though there is a spike near 2020. THe ACF of residuals also looks good, where most of the points are inside the confidence interval. The Ljung-Box plot does not look very good though - many of the points are below the confidence interval.

```
ar2_ma5 = sarima(gdp_log, p = 2, d = 1, q = 3)
```

```
## initial  value -4.318719
## iter   2 value -4.357079
## iter   3 value -4.371727
## iter   4 value -4.372425
## iter   5 value -4.372785
## iter   6 value -4.372839
## iter   7 value -4.373275
## iter   8 value -4.373300
## iter   9 value -4.373304
## iter  10 value -4.373306
## iter  11 value -4.373310
## iter  12 value -4.373312
## iter  13 value -4.373313
## iter  14 value -4.373315
## iter  15 value -4.373316
## iter  16 value -4.373318
## iter  17 value -4.373323
## iter  18 value -4.373324
## iter  19 value -4.373326
## iter  20 value -4.373332
## iter  21 value -4.373346
## iter  22 value -4.373379
## iter  23 value -4.373464
## iter  24 value -4.373885
## iter  25 value -4.374111
## iter  26 value -4.374252
## iter  27 value -4.374277
## iter  28 value -4.374405
## iter  29 value -4.374419
## iter  30 value -4.374539
## iter  31 value -4.374585
## iter  32 value -4.374649
## iter  33 value -4.374688
## iter  34 value -4.374714
## iter  35 value -4.374735
## iter  36 value -4.374781
## iter  37 value -4.374943
## iter  38 value -4.376077
## iter  39 value -4.376564
## iter  40 value -4.376963
## iter  41 value -4.377867
## iter  42 value -4.378269
## iter  43 value -4.378739
## iter  44 value -4.379454
## iter  45 value -4.380230
## iter  46 value -4.380542
## iter  47 value -4.380699
```

```
## iter  48 value -4.380765
## iter  49 value -4.380788
## iter  50 value -4.380796
## iter  51 value -4.380799
## iter  52 value -4.380799
## iter  53 value -4.380799
## iter  54 value -4.380800
## iter  55 value -4.380800
## iter  56 value -4.380800
## iter  57 value -4.380800
## iter  58 value -4.380800
## iter  58 value -4.380800
## iter  58 value -4.380800
## final  value -4.380800
## converged
## initial  value -4.381148
## iter   2 value -4.381161
## iter   3 value -4.381176
## iter   4 value -4.381182
## iter   5 value -4.381222
## iter   6 value -4.381228
## iter   7 value -4.381231
## iter   8 value -4.381231
## iter   9 value -4.381232
## iter  10 value -4.381232
## iter  11 value -4.381233
## iter  12 value -4.381233
## iter  13 value -4.381234
## iter  14 value -4.381235
## iter  14 value -4.381235
## final  value -4.381235
## converged
```




The residuals look very similar between this model and the other. The Ljung-Box looks better though, especially at the beginning of the time series. The key question with this model is, are all five of the MA terms significant?

```
ar2_ma5$fit
```

```
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control = list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3  constant
##       1.3972  -0.8648  -1.2225  0.7828  0.0453    0.0153
## s.e.  0.1287   0.1228   0.1440  0.0956  0.0611    0.0009
##
## sigma^2 estimated as 0.0001562:  log likelihood = 888.69,  aic = -1763.38
```

These NAs in the coefficients suggest that the model is overfit. So this is not a good choice for this data.

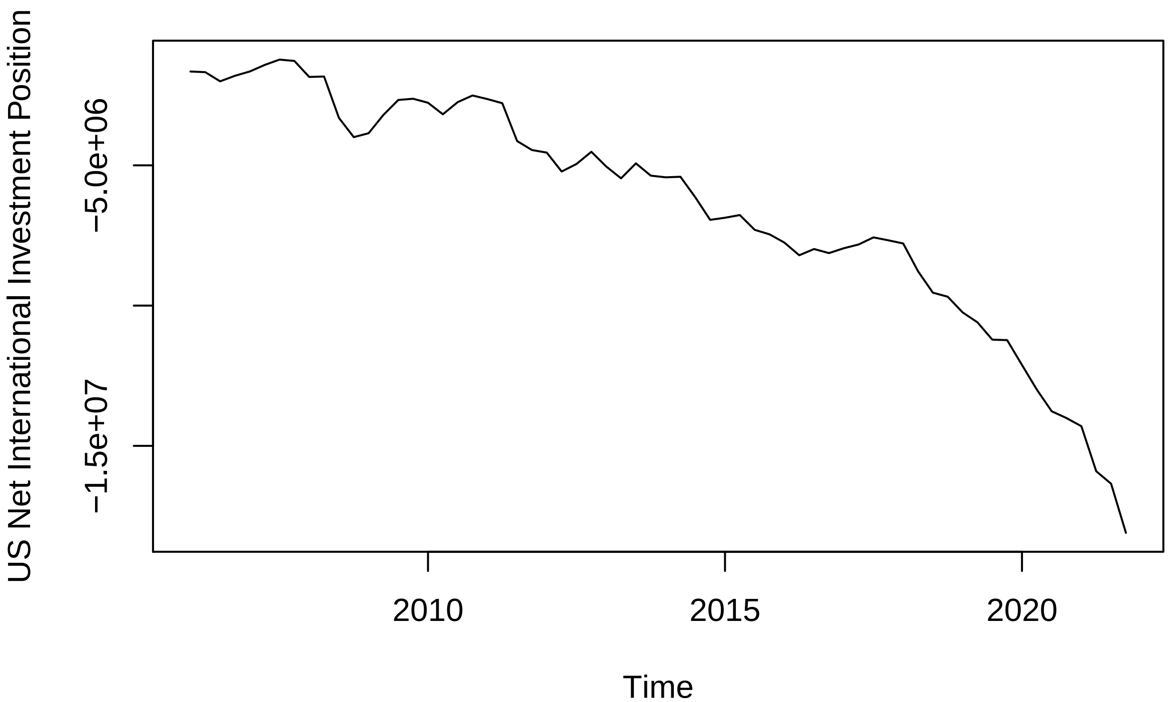
### U.S. Net International Investment Position

** Data Cleaning **

```r
# Data downloaded from this site on 4/24/22:
# https://fred.stlouisfed.org/series/IIPUSNETIQ
invest = fread("C:/Users/eklin/Downloads/IIPUSNETIQ.csv")
```

** Plotting and Analysis **

```r
# This is a quarterly series starting in Q1 2006.
invest = ts(invest$IIPUSNETIQ, frequency = 4, start = c(2006, 1))
plot(invest, ylab = "US Net International Investment Position")
```




The dataset has a strong downwards trend, so it's non-stationary. However, there doesn't appear to be any seasonality from a visual inspection.

** Evaluate stationarity with a hypothesis test **

```r
# Check the stationarity of the series with a KPSS test.
kpss.test(invest)
```

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

```
##
##  KPSS Test for Level Stationarity
##
## data:  invest
## KPSS Level = 1.5676, Truncation lag parameter = 3, p-value = 0.01
```

```r
# We have a p-value of 0.01, so we fail to reject the null hypothesis.
# This dataset is non-stationary, so we difference it.
invest_diff <- diff(invest, differences = 1)

# Evaluate KPSS test again
kpss.test(invest_diff)
```

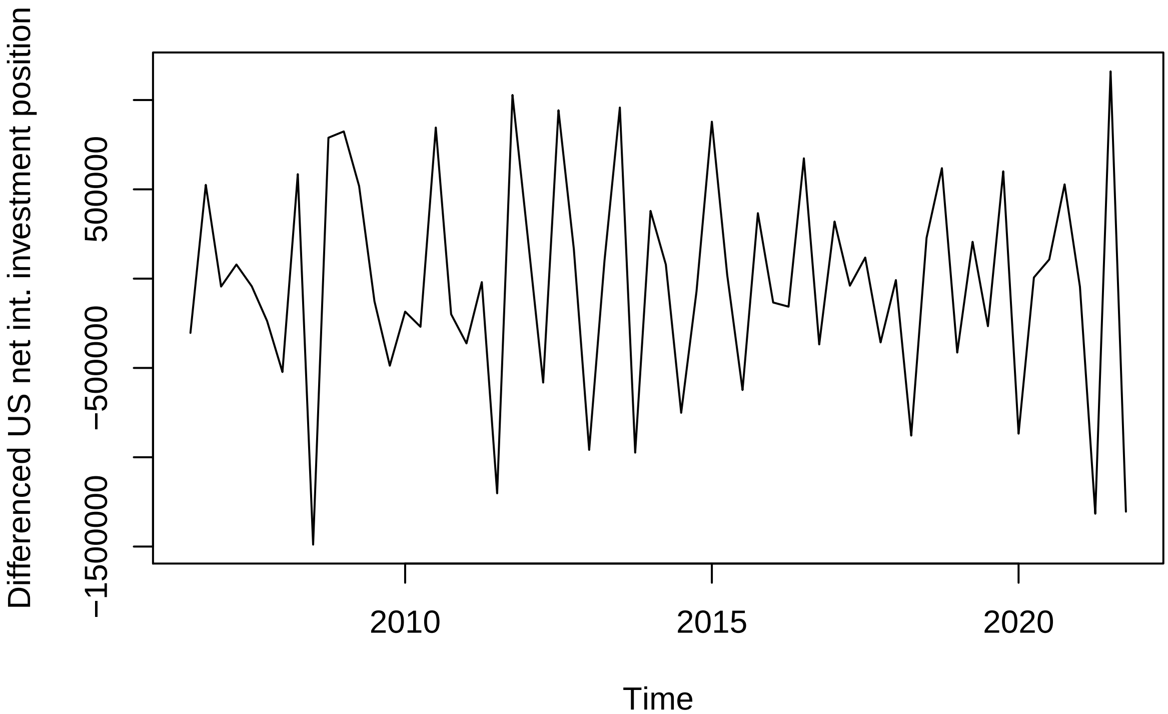
```
##
##  KPSS Test for Level Stationarity
##
## data:  invest_diff
## KPSS Level = 0.60201, Truncation lag parameter = 3, p-value = 0.02245
```

```r
# With a p-value of 0.022, this data is still non-stationary.

# Try a difference of 2.
invest_diff <- diff(invest, differences = 2)
kpss.test(invest_diff)
```

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

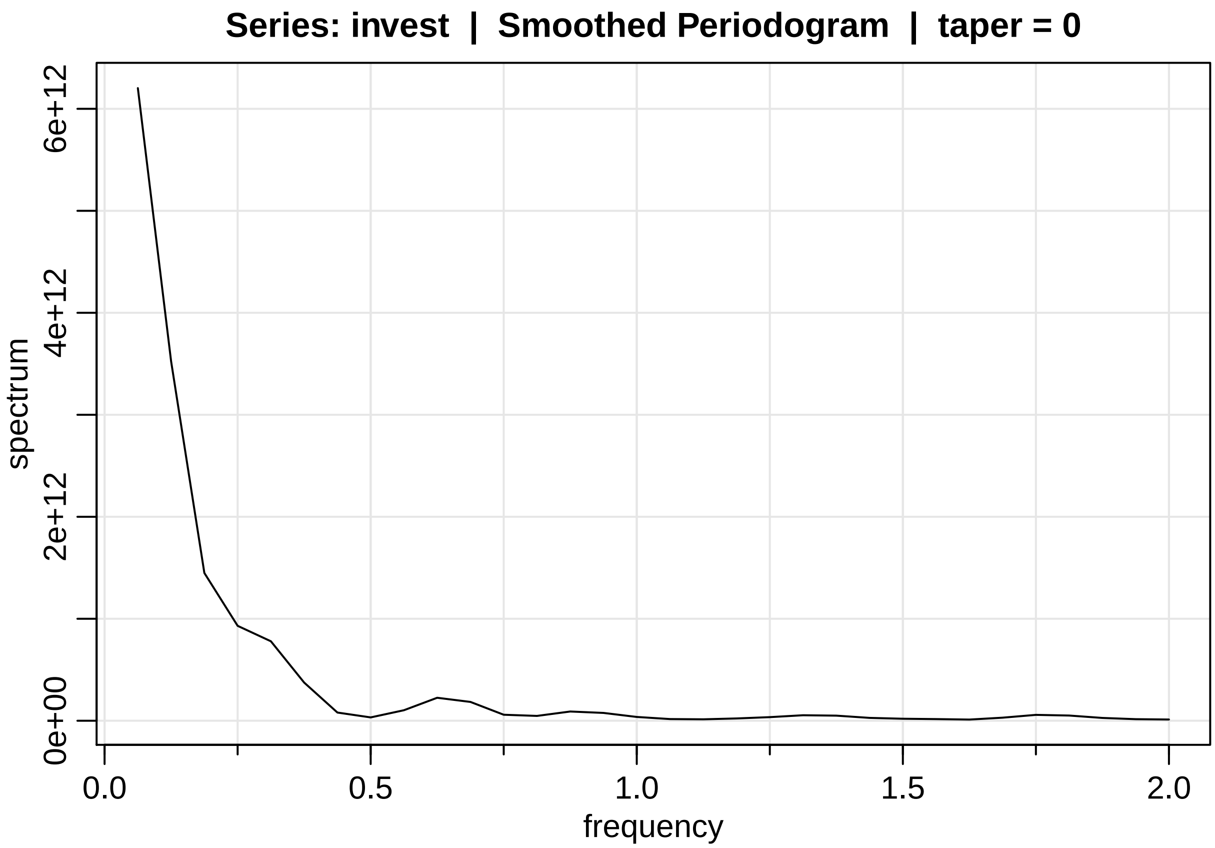
```
##
##  KPSS Test for Level Stationarity
##
## data:  invest_diff
## KPSS Level = 0.086449, Truncation lag parameter = 3, p-value = 0.1
```

```r
# Finally, with a p-value of 0.1, we reject the null hypothesis.
# This differenced dataset is stationary.
plot(invest_diff, ylab = "Differenced US net int. investment position")
```




** Evaluate Seasonality **

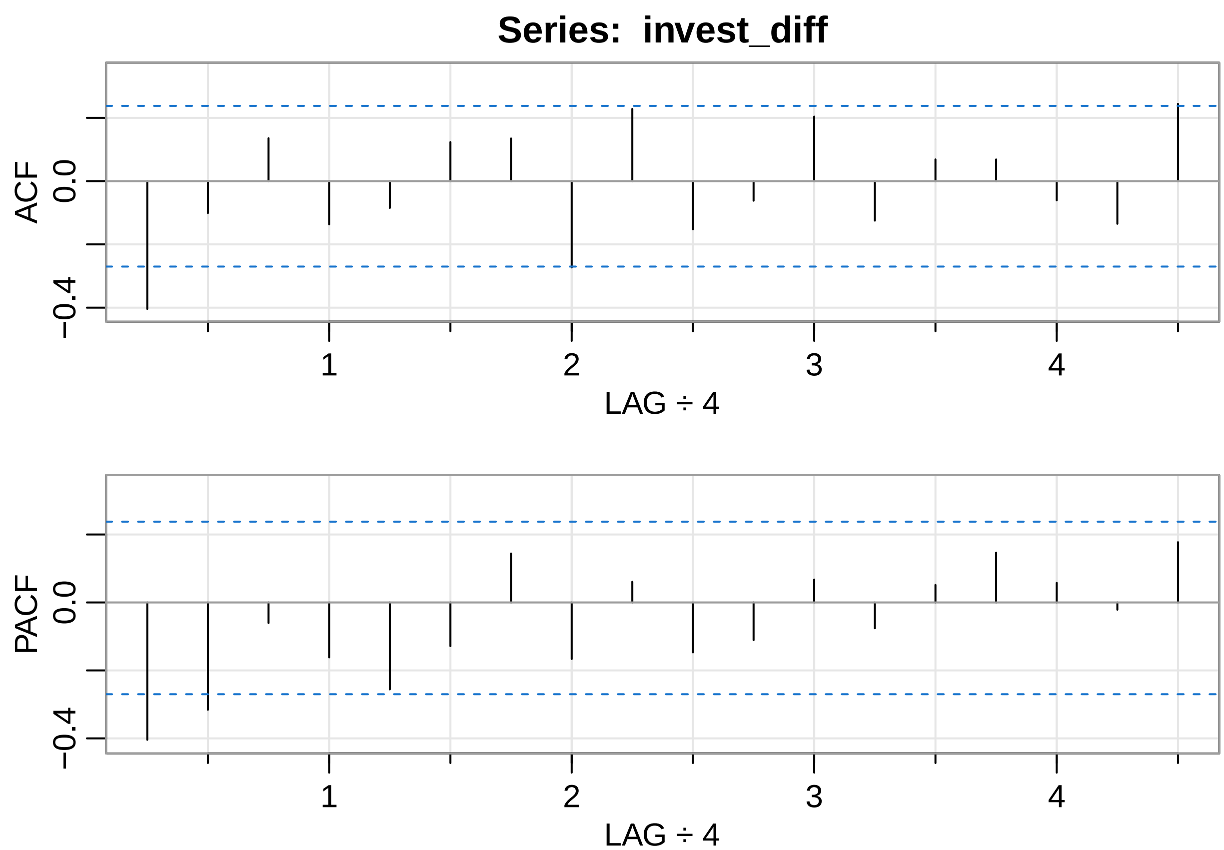
```
invest_spec = mvspec(invest, detrend = TRUE, spans = 3)
```




Again, the function is having some trouble detrending this curve, and there is no obvious seasonality.

** ACF/PACF **

```
# Run ACF/PACF on differenced series.
astsa::acf2(invest_diff)
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6] [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  -0.4 -0.10  0.14 -0.14 -0.08  0.12 0.13 -0.27 0.23 -0.15 -0.06  0.20 -0.12
## PACF -0.4 -0.32 -0.06 -0.16 -0.26 -0.13 0.14 -0.17 0.06 -0.15 -0.11  0.07 -0.08
##       [,14] [,15] [,16] [,17] [,18]
## ACF   0.07  0.07 -0.06 -0.13  0.24
## PACF  0.05  0.15  0.06 -0.02  0.18
```

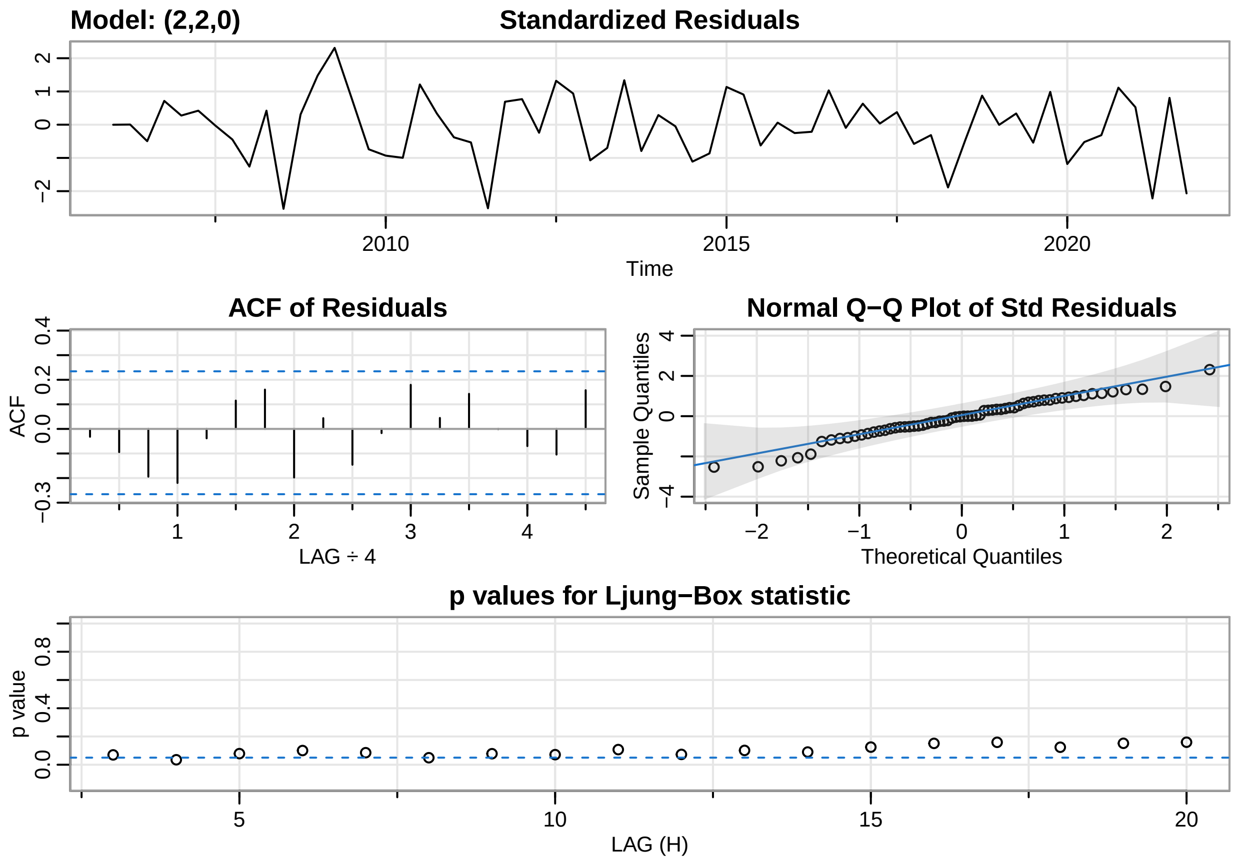
I see two significant lags in the PACF, which tells me that an autoregressive model might be better. I'll model an autoregressive model with 2 lags. I'll also model a MA-1 model for comparison, because there was one significant lag in the ACF.

** ARIMA Modeling **

```
# Model the original series, and include the difference term in the model
ar2 <- astsa::sarima(invest, p = 2, d = 2, q = 0)
```

```
## initial  value 13.333746
## iter   2 value 13.215197
## iter   3 value 13.192191
## iter   4 value 13.187362
## iter   5 value 13.187341
## iter   6 value 13.187340
## iter   6 value 13.187340
## final  value 13.187340
## converged
```

```
## initial  value 13.180413
## iter   2 value 13.180387
## iter   3 value 13.180387
## iter   3 value 13.180387
## iter   3 value 13.180387
## final  value 13.180387
## converged
```

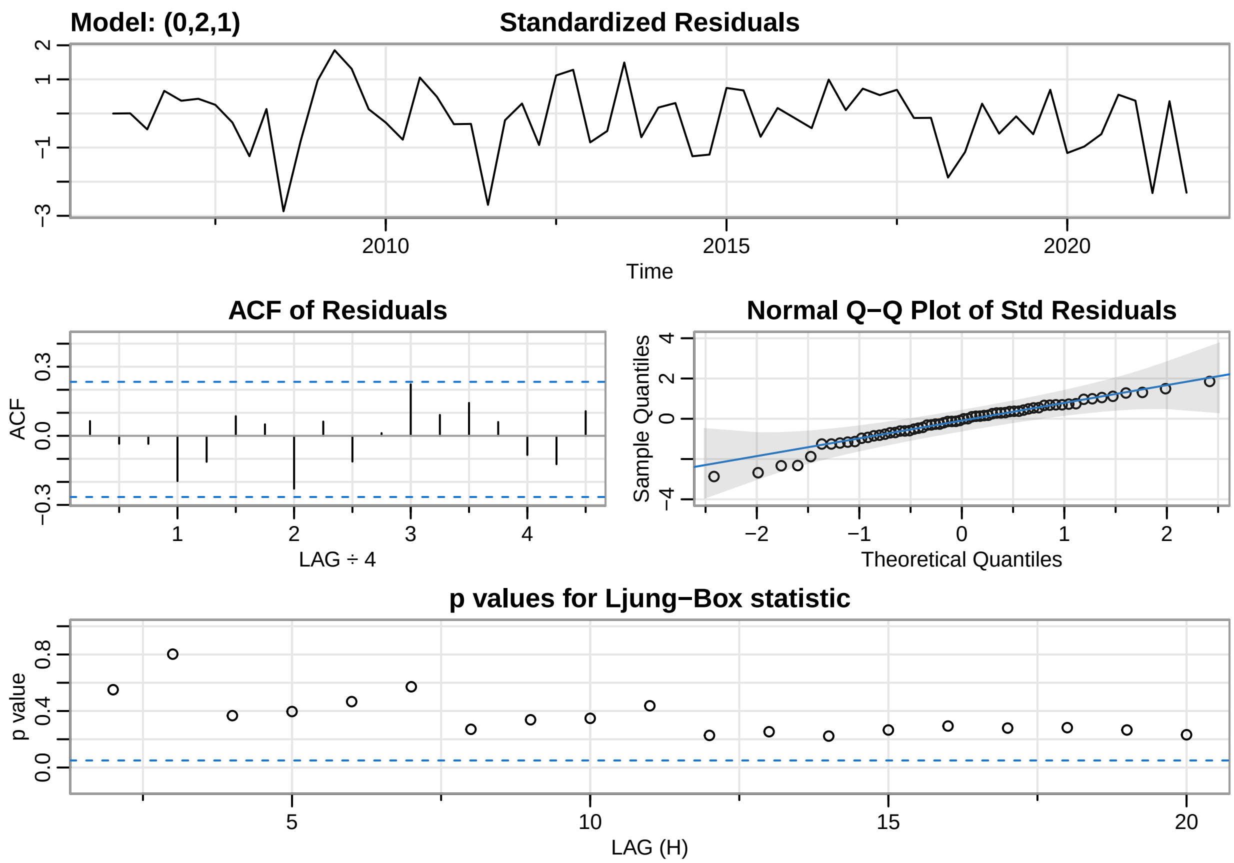



I see mostly white noise in the residuals, based on the residual plot and the residual ACF. The QQ plot indicates normality as well, because most of the plots are on the line. But for the Ljung-Box plot, many of the points are on or within the confidence interval.

```r
# Model the original series, and include the difference term in the model
ma1 <- astsa::sarima(invest, p = 0, d = 2, q = 1)
```

```
## initial  value 13.325322
## iter   2 value 13.195188
## iter   3 value 13.165830
## iter   4 value 13.138301
## iter   5 value 13.115358
## iter   6 value 13.115092
## iter   7 value 13.115049
## iter   8 value 13.115048
## iter   8 value 13.115048
## iter   8 value 13.115048
## final  value 13.115048
```

```
## converged
## initial  value 13.125890
## iter   2 value 13.125555
## iter   3 value 13.125470
## iter   4 value 13.125468
## iter   4 value 13.125468
## final  value 13.125468
## converged
```




For the MA-1 model, the residuals plots and QQ plot look very similar, but the Ljung-Box plot is markedly improved. Many of the points are now above the confidence interval. For this reason alone, I would probably choose the MA-1 model for this data.

One last thing I wanted to consider was the AICc. The AICc for this model is 29.2987048, and the AICc for the MA-1 is 29.1544046. These are practically identical, so they're not a good criterion for choosing a model here.

### U.S. Debt

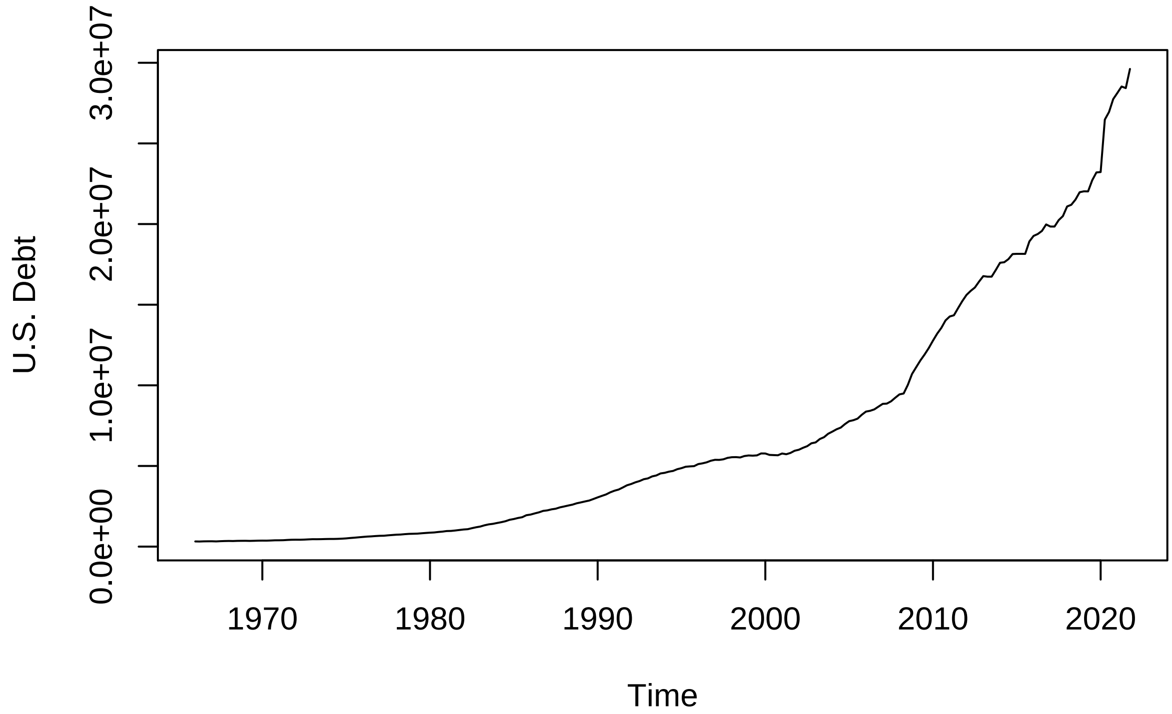
** Data Cleaning **

```r
# Data was downloaded from https://fred.stlouisfed.org/series/GFDEBTN on 5/11/2022
dt <- fread("C:/Users/eklin/Downloads/GFDEBTN.csv")
```

There are 0 NA observations in the data.

** Plotting and Analysis **

```r
debt <- ts(dt$GFDEBTN, start = c(1966, 1), frequency = 4)
plot(debt, ylab = "U.S. Debt")
```




This is another highly non-stationary series. There is also a notable spike in debt during the COVID-19 pandemic, right around the start of 2020.

** Evaluate stationarity with a hypothesis test**

```r
# Null hypothesis: data is stationary
# Alternative hypothesis: data is non-stationary
kpss.test(debt)
```

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

```
##
##  KPSS Test for Level Stationarity
##
## data:  debt
## KPSS Level = 3.82, Truncation lag parameter = 4, p-value = 0.01
```

```r
# p-value is 0.01, so we fail to reject the null hypothesis. Data is non-stationary.
debt_diff <- diff(debt, lag = 1)
kpss.test(debt_diff)
```

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

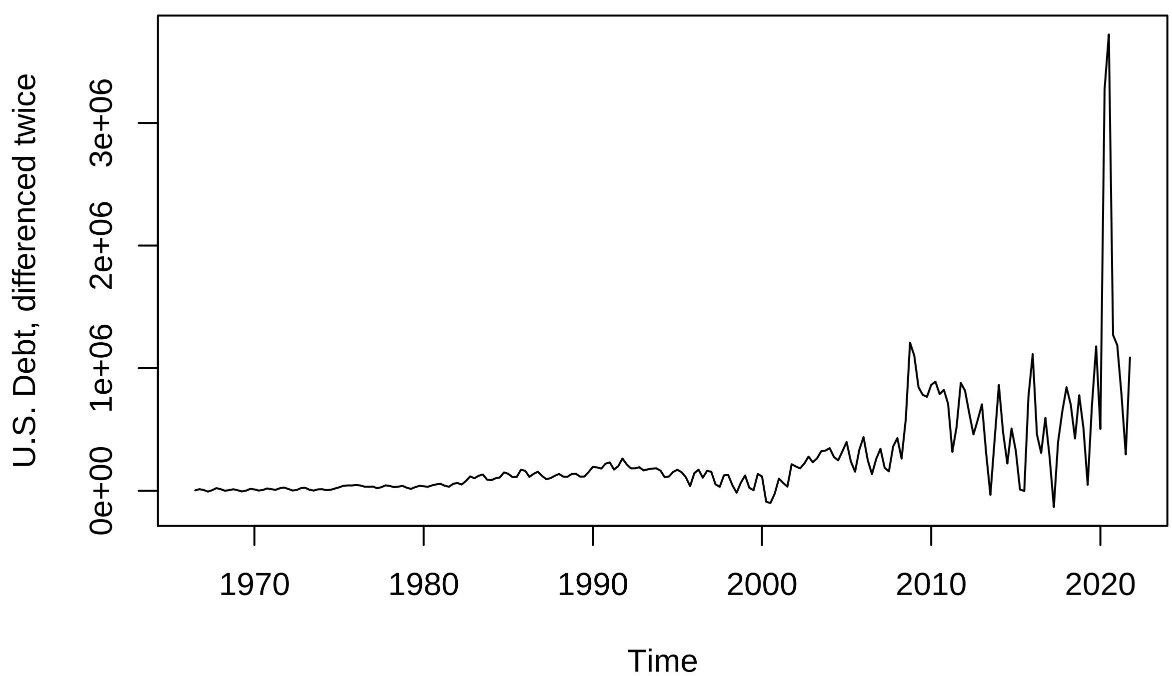
```
##
##  KPSS Test for Level Stationarity
##
## data:  debt_diff
## KPSS Level = 2.2872, Truncation lag parameter = 4, p-value = 0.01
```

```r
# After one lag, p-value is still 0.01. Data is still non-stationary, so try 2 lags.
debt_diff <- diff(debt, lag = 2)
kpss.test(debt_diff)
```

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

```
##
##  KPSS Test for Level Stationarity
##
## data:  debt_diff
## KPSS Level = 2.3024, Truncation lag parameter = 4, p-value = 0.01
```

```r
plot(debt_diff, ylab = "U.S. Debt, differenced twice")
```




This has a very similar problem to the US GDP series. There is very abnormal behavior around the COVID-19 pandemic. I'll try a similar approach of taking a log before differencing.

```r
debt_log = log(debt)
kpss.test(debt_log)
```

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

##
##  KPSS Test for Level Stationarity
##
## data:  debt_log
## KPSS Level = 4.5209, Truncation lag parameter = 4, p-value = 0.01
```

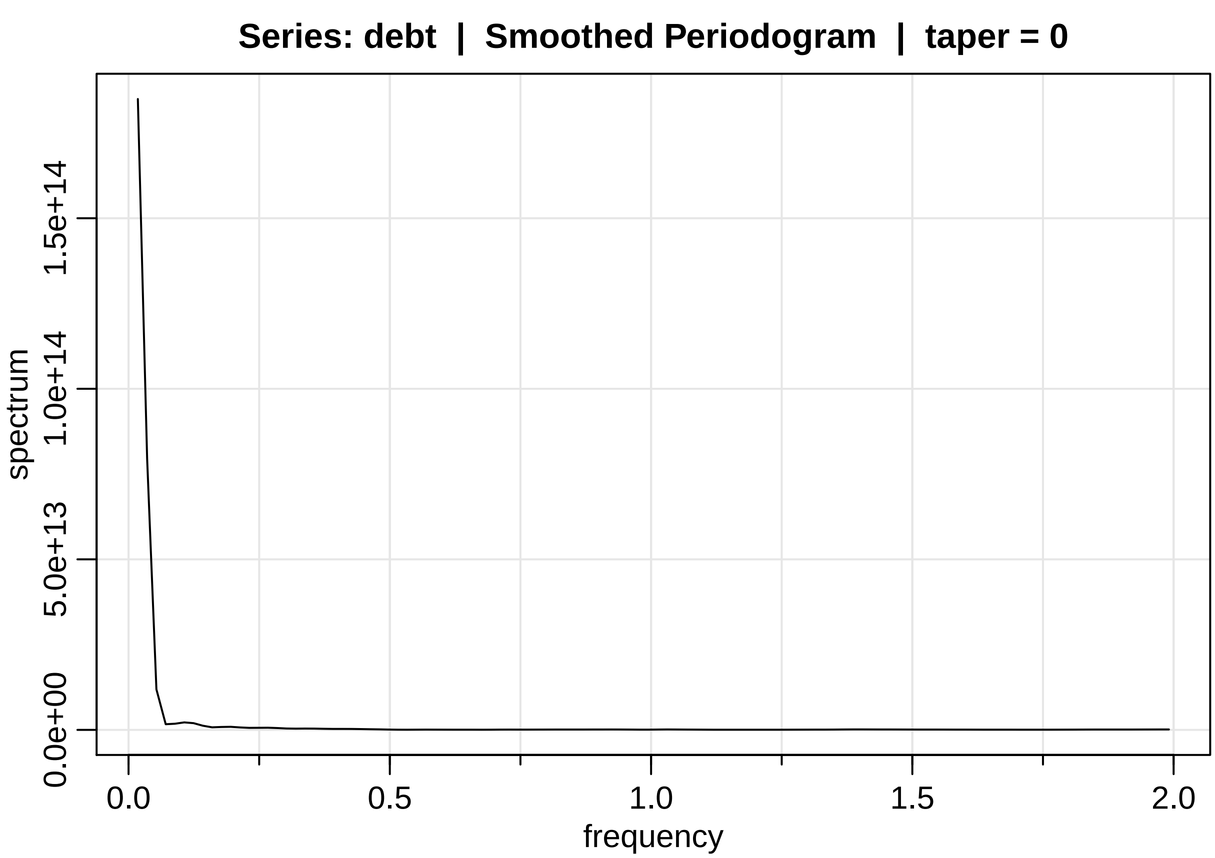
```
# p-value is 0.01 < 0.05, so we reject the null hypothesis. The data is non-stationary.

debt_log_diff <- diff(debt_log, lag = 1)
kpss.test(debt_log_diff)
```

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

##
##  KPSS Test for Level Stationarity
##
## data:  debt_log_diff
## KPSS Level = 0.28154, Truncation lag parameter = 4, p-value = 0.1
```

```
# p-value is 0.1, so we fail to reject the null. The data is stationary after being logged and differen
```

** Evaluate Seasonality **

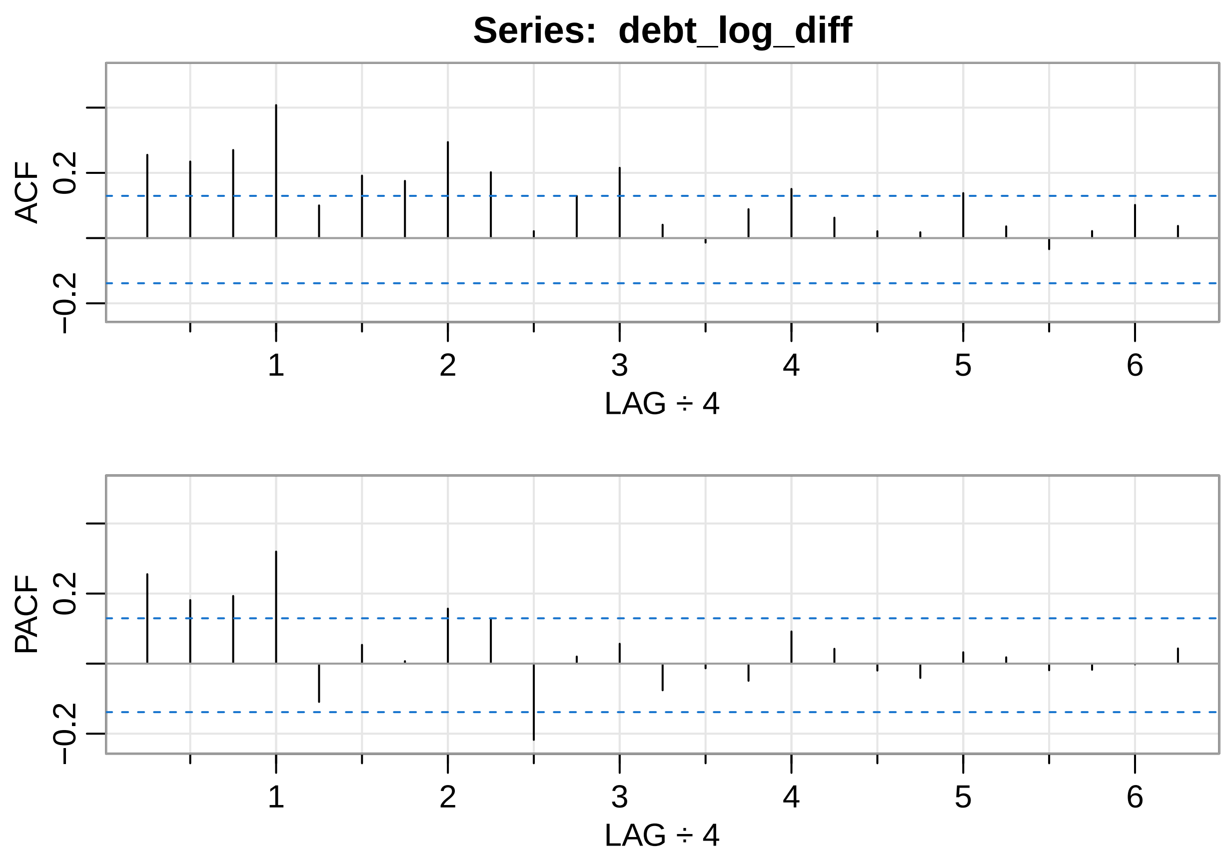
```
debt_spec = mvspec(debt, spans = 2, detrend = TRUE)
```



Again, there is no evidence of seasonality in this series, just a strong trend.

** ACF/PACF **

```r
acf2(debt_log_diff)
```




```
##      [,1] [,2] [,3] [,4]  [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.26 0.23 0.27 0.41  0.10 0.19 0.18 0.29 0.20  0.02  0.13  0.22  0.04
## PACF 0.26 0.18 0.19 0.32 -0.11 0.05 0.01 0.16 0.13 -0.22  0.02  0.06 -0.08
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.01  0.09  0.15  0.06  0.02  0.02  0.14  0.04 -0.03  0.02   0.1  0.04
## PACF -0.01 -0.05  0.09  0.04 -0.02 -0.04  0.03  0.02 -0.02 -0.02   0.0  0.04
```

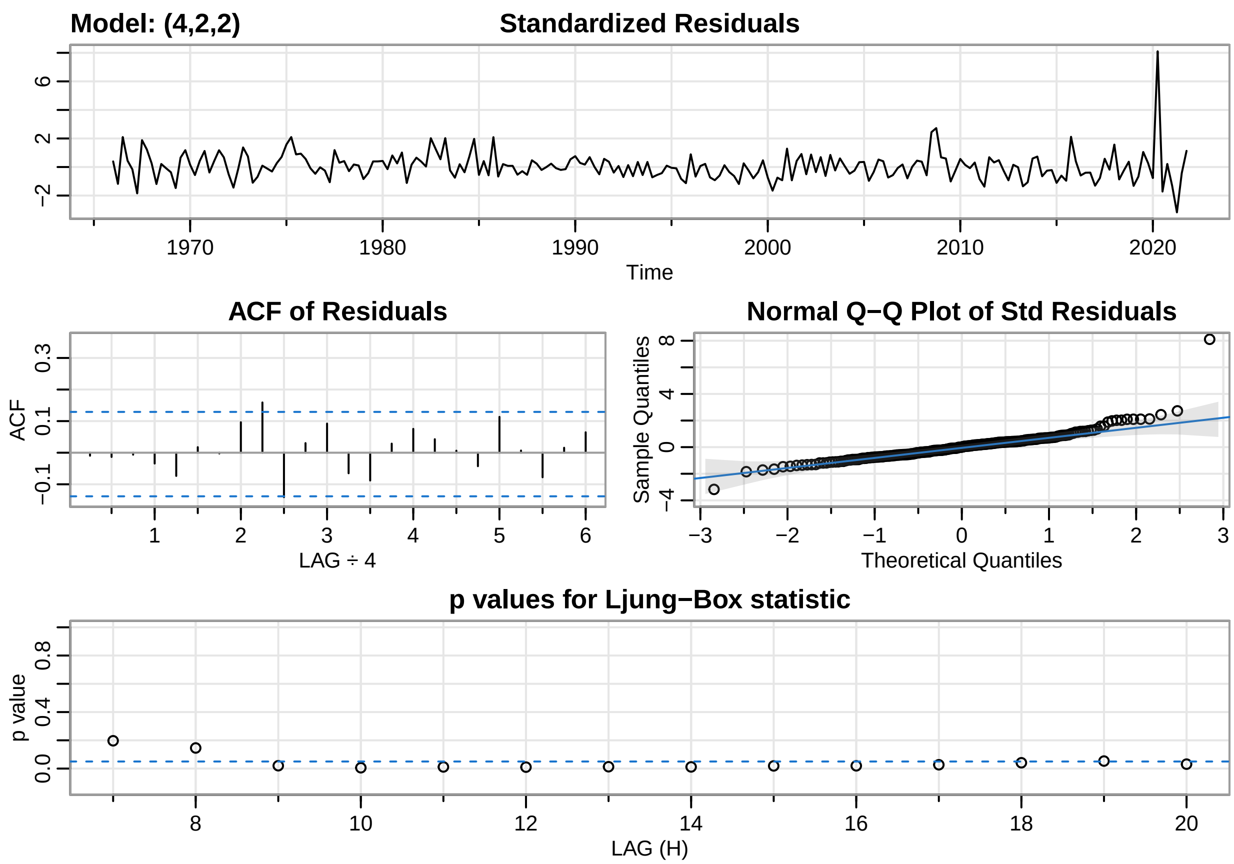
This is very interesting - there are four pronounced lags for both the ACF and PACF. So I'll try a AR-4, MA-4 model with 1 difference term.

** ARIMA Modeling **

```r
ar4_ma4 = sarima(debt_log, p = 4, d = 2, q = 2)
```

```
## initial  value -3.921821
## iter   2 value -4.163928
## iter   3 value -4.198609
## iter   4 value -4.205193
## iter   5 value -4.213691
```

```
## iter   6 value -4.216905
## iter   7 value -4.218569
## iter   8 value -4.221616
## iter   9 value -4.222327
## iter  10 value -4.222601
## iter  11 value -4.222791
## iter  12 value -4.223067
## iter  13 value -4.223331
## iter  14 value -4.223486
## iter  15 value -4.223494
## iter  16 value -4.223498
## iter  17 value -4.223502
## iter  18 value -4.223503
## iter  19 value -4.223509
## iter  20 value -4.223513
## iter  21 value -4.223515
## iter  22 value -4.223515
## iter  23 value -4.223516
## iter  24 value -4.223516
## iter  24 value -4.223516
## iter  24 value -4.223516
## final  value -4.223516
## converged
## initial  value -4.219330
## iter   2 value -4.220286
## iter   3 value -4.221341
## iter   4 value -4.223521
## iter   5 value -4.224604
## iter   6 value -4.225196
## iter   7 value -4.225928
## iter   8 value -4.226094
## iter   9 value -4.226374
## iter  10 value -4.226555
## iter  11 value -4.227022
## iter  12 value -4.227577
## iter  13 value -4.228018
## iter  14 value -4.229114
## iter  15 value -4.229861
## iter  16 value -4.229967
## iter  17 value -4.230046
## iter  18 value -4.230077
## iter  19 value -4.230084
## iter  20 value -4.230087
## iter  21 value -4.230090
## iter  22 value -4.230110
## iter  23 value -4.230112
## iter  24 value -4.230112
## iter  24 value -4.230112
## iter  24 value -4.230112
## final  value -4.230112
## converged
```




This looks very similar to the model performance for US GDP. There is no evidence of a trend in the residuals plots, although there is a spike in 2020. This spike also appears in the far right of the normal-QQ plot. The Ljung-Box plot, though, shows several points within the confidence interval, which is not a great indicator for this model.

#### International airline freight to the United States

This data represents all nonstop commercial airline freight traffic traveling to the United States. It is maintained by the Department of Transportation, and has monthly data from January 1990 - September 2021.

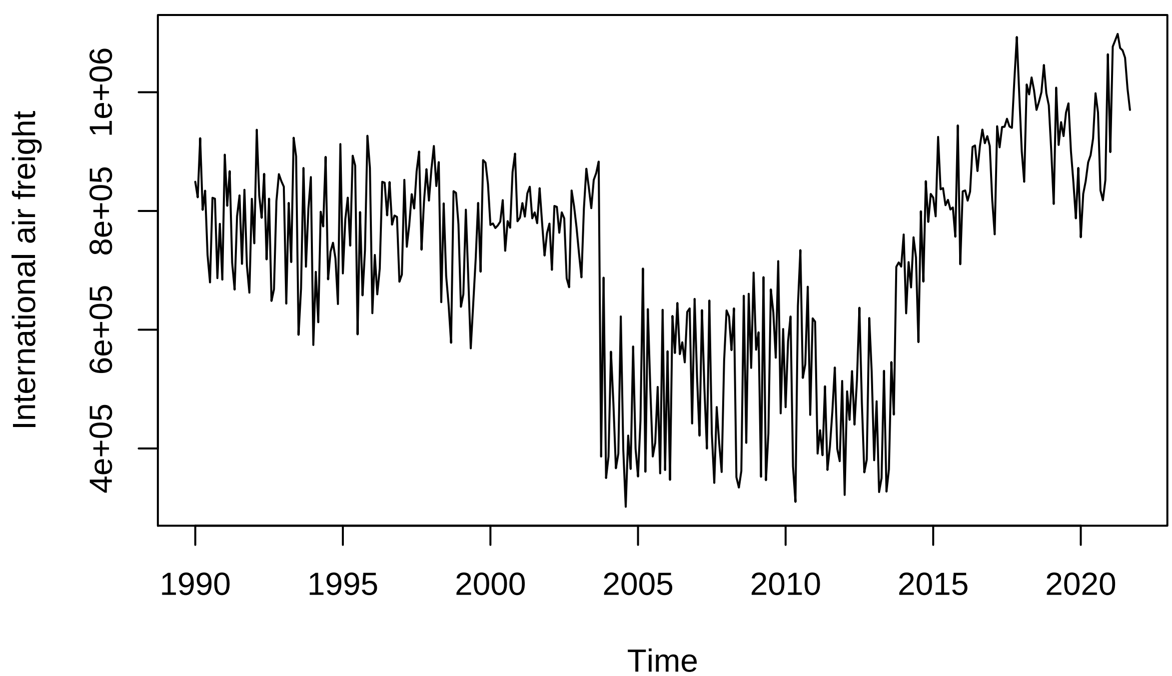
** Data Cleaning **

```
# This is the raw dataset downloaded from https://data.transportation.gov/Aviation/International_Report_
data <- fread("C:/Users/eklin/Downloads/International_Report_Freight.csv")

# I only care about the total number of flights flown for a given year and month.
# So I want to collapse out the airline and type columns.
data <- data[, .(flights = sum(Total, na.rm = F)), by = 'data_dte']
setnames(data, 'data_dte', 'date')
data[, date:=as.Date(data$date, format = "%m/%d/%Y")]

# Here's what the collapsed data looks like.
kable(head(data))
```

| date | flights |
|---|---|
| 2008-05-01 | 849231 |
| 2005-06-01 | 823032 |
| 2006-09-01 | 922600 |
| 2004-08-01 | 801918 |
| 2004-03-01 | 834182 |
| 2002-03-01 | 725762 |

** Plotting and analysis ** There are 0 NA observations in the time series. Next I want to turn the data into a time series and do some exploratory data analysis.

```
flights <- ts(data$flights, start = c(1990, 1), frequency = 12)
plot(flights, ylab = 'International air freight')
```




This is an interesting series - it's definitely non-stationary, but not in a constant way. There is a period of time where international freight traffic really fell from 2002 through 2014.

** Evaluate stationarity with a hypothesis test **

```
# Null hypothesis: data is stationary
# Alternative hypothesis: data is non-stationary
kpss.test(flights)
```

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

```
##
##  KPSS Test for Level Stationarity
##
## data:  flights
## KPSS Level = 1.0539, Truncation lag parameter = 5, p-value = 0.01
```

```
# p-value is 0.01, so we reject the null hypothesis. Data is non-stationary, so we need to difference t
flights_diff = diff(flights, lag = 1)
kpss.test(flights_diff)
```

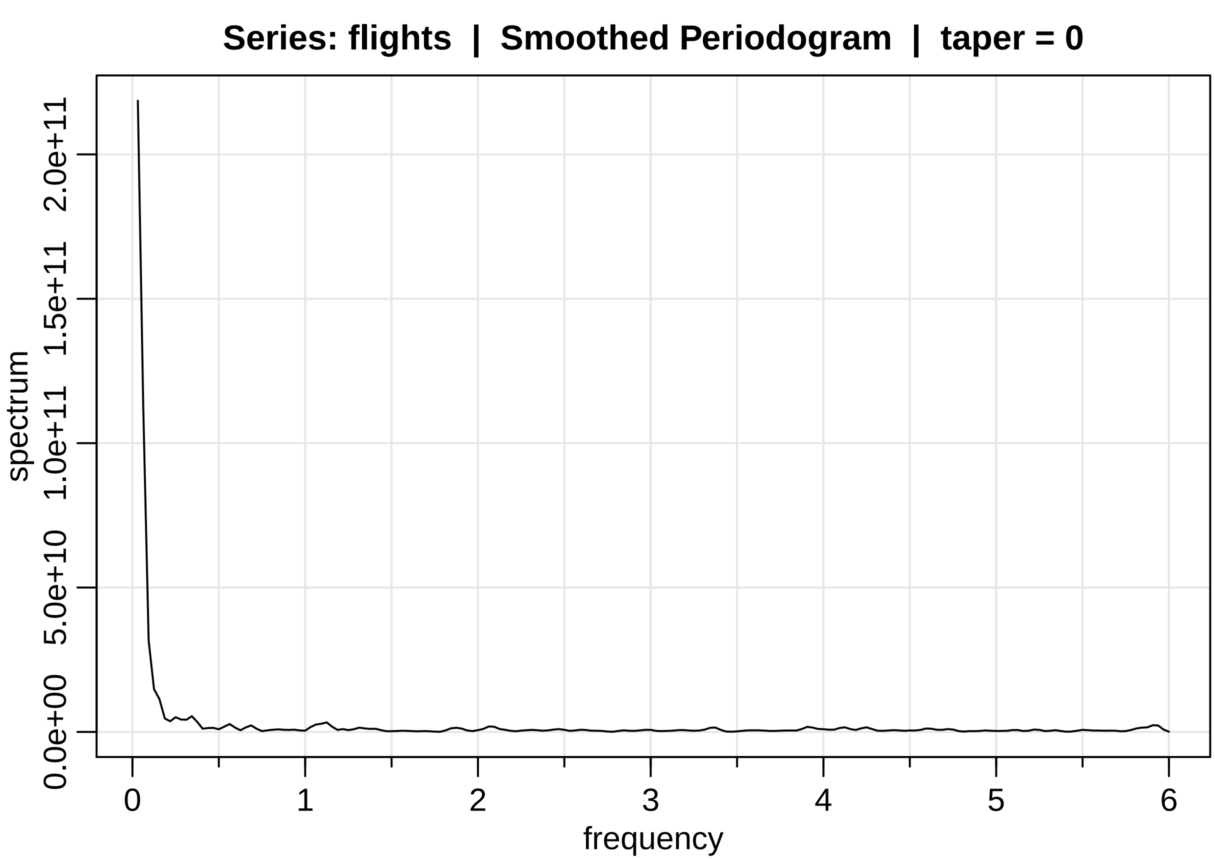
```
## Warning in kpss.test(flights_diff): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  flights_diff
## KPSS Level = 0.062445, Truncation lag parameter = 5, p-value = 0.1
```

```
# After 1 lag we get a p-value of 0.1, so we fail to reject the null hypothesis.
# Data is stationary.
```

** Investigate seasonality **

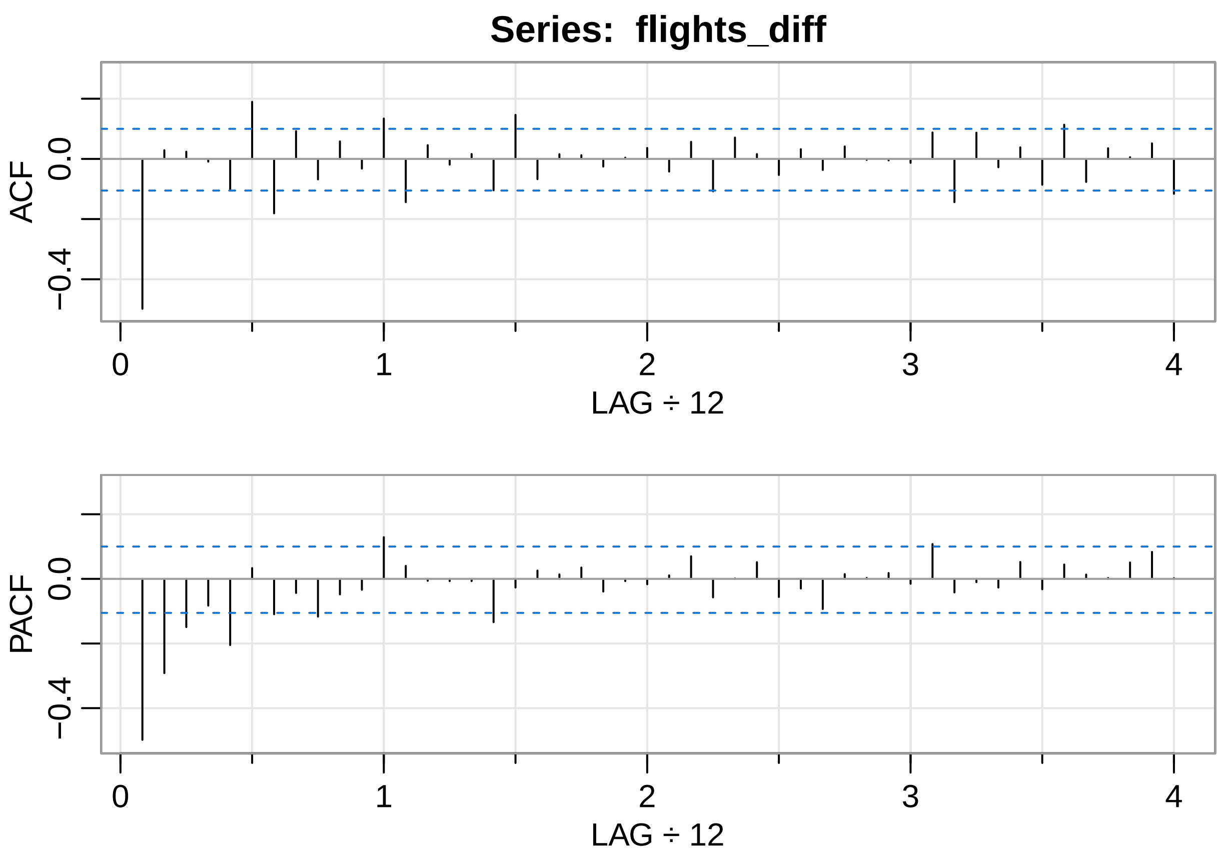
```
flights_spec = mvspec(flights, spans = 2) # No need for detrend because there’s not a strong trend
```




Again, there is not strong evidence of seasonality in this time series.

** ACF/PACF **

```
astsa::acf2(flights_diff)
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5] [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.5  0.03  0.02 -0.01 -0.11 0.19 -0.18  0.09 -0.07  0.06 -0.03  0.13
## PACF -0.5 -0.29 -0.15 -0.08 -0.20 0.03 -0.11 -0.04 -0.12 -0.05 -0.03  0.13
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF  -0.14  0.05 -0.02  0.02 -0.10  0.15 -0.07  0.02  0.01 -0.03  0.00  0.04
## PACF  0.04 -0.01 -0.01 -0.01 -0.13 -0.03  0.03  0.01  0.04 -0.04 -0.01 -0.02
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF  -0.04  0.06 -0.11  0.07  0.02 -0.05  0.03 -0.04  0.04     0  0.00 -0.01
## PACF  0.01  0.07 -0.06  0.00  0.05 -0.06 -0.03 -0.09  0.01     0  0.02 -0.02
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.09 -0.14  0.09 -0.03  0.04 -0.09  0.11 -0.08  0.04  0.01  0.05 -0.12
## PACF  0.11 -0.04 -0.01 -0.03  0.05 -0.03  0.04  0.01  0.00  0.05  0.08  0.00
```

In the ACF, the first lag is significant, as well as some lags around the six month mark and 1 year. In the PACF, the first three lags are significant, as well as the lag at 1 year. The models I might try would be:

Model 1: Modeled seasonally using sarima, with the following terms: * MA-1 * SMA-1 * AR-3 * SAR-1

Model 2: Fourier seasonality with K = 4, AR = 2, MA = 1

Model 3: Fourier seasonality with K = 6, AR = 2, MA = 1

** ARIMA Modeling **

```
model1 = sarima(flights, S=12,
                p = 3, d = 1, q = 1,
                P = 1, D = 0, Q = 1)
```

```
flights_fourier4 = arima(flights,
                     order = c(2, 1, 1), # p, d, q
                     xreg = fourier(flights, K = 4))
flights_fourier4
```

```
##
## Call:
## arima(x = flights, order = c(2, 1, 1), xreg = fourier(flights, K = 4))
##
## Coefficients:
##          ar1     ar2      ma1      S1-12     C1-12       S2-12     C2-12
##       0.0214  0.0797  -0.7576  -377.3723  1376.122  -11626.834  5840.368
## s.e.  0.0871  0.0733   0.0678  7605.0708  7613.849    6360.902  6354.311
##           S3-12     C3-12      S4-12     C4-12
##       -12388.83  2781.208  -6889.850  10352.54
## s.e.    5961.14  5976.374   6087.954   6089.01
##
## sigma^2 estimated as 1.004e+10:  log likelihood = -4915.26,  aic = 9854.52
```

```
flights_fourier6 = arima(flights,
                     order = c(2, 1, 1), # p, d, q
                     xreg = fourier(flights, K = 6))
flights_fourier6
```

```
##
## Call:
## arima(x = flights, order = c(2, 1, 1), xreg = fourier(flights, K = 6))
##
## Coefficients:
##          ar1     ar2      ma1      S1-12     C1-12      S2-12     C2-12
##       0.0238  0.0769  -0.7569  -389.3581  1401.192  -11636.25  5827.604
## s.e.  0.0871  0.0732   0.0679  7599.2297  7608.285    6363.79  6357.417
##            S3-12     C3-12      S4-12      C4-12     S5-12      C5-12     C6-12
##       -12360.456  2749.598  -6826.574  10348.566  6901.087   101.0015  2362.574
## s.e.    5963.709  5978.902   6077.463   6078.245  6468.574  6485.9972  4753.585
##
## sigma^2 estimated as 1.001e+10:  log likelihood = -4914.57,  aic = 9859.15
```

## Additional Analysis

For additional analyses, I was hoping to do three things. 1. Non-ARIMA modeling for the flights dataset 2. Dynamic regression with the four datasets

I have been able to include #1 in the flights section above, and plan to include #2 in my final report.

## Summary and Implications

What this project has shown me is that additional transformations may be needed to work with highly non-stationary series. I've also found that auto.arima() is a really useful tool for checking for "blind spots" in your model, like looking for seasonality where you weren't anticipating it. I don't have too many findings about the time series themselves without doing the dynamic regression, but I'm looking forward to running this analysis.