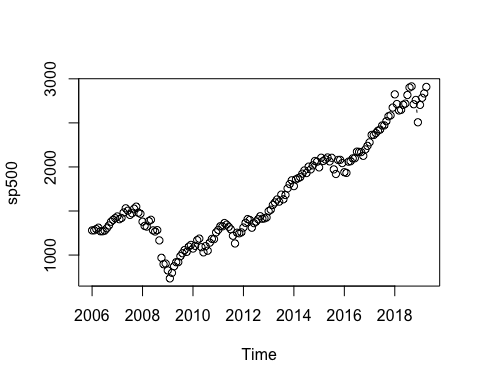
HW 4 Kailu Wang

stock <- read.csv("/Users/Sebastian/Desktop/data analysis/SP500.csv")  
sp500 <- ts(stock$Adj.Close, freq = 12, start = c(2006,1))  
head(sp500)

## [1] 1280.08 1280.66 1294.87 1310.61 1270.09 1270.20

plot(sp500, type = 'b')

 This is strong seasons in positive trend and is heteroscedastic.

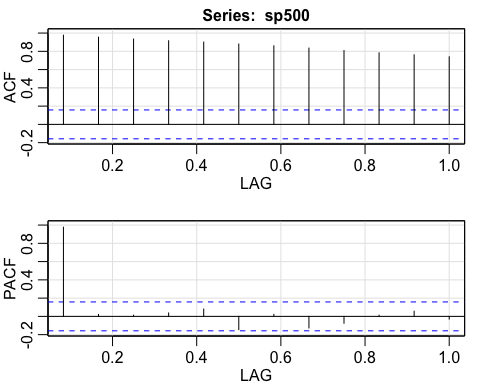
#install.packages("astsa")  
#install.packages("forecast")  
library(astsa)  
library(forecast)

## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2018i.  
## 1.0/zoneinfo/America/New\_York'

##   
## Attaching package: 'forecast'

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

acf2(sp500, max.lag = 12)



## ACF PACF  
## [1,] 0.98 0.98  
## [2,] 0.95 0.02  
## [3,] 0.93 0.01  
## [4,] 0.92 0.04  
## [5,] 0.90 0.08  
## [6,] 0.88 -0.15  
## [7,] 0.86 0.02  
## [8,] 0.84 -0.13  
## [9,] 0.81 -0.07  
## [10,] 0.78 0.01  
## [11,] 0.76 0.06  
## [12,] 0.74 -0.03

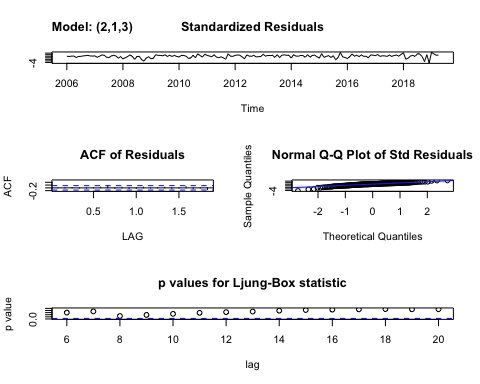
By observing these two graphs, we can see that ACF is high auto-correlation and PACF is low auto-correlation. This is Arima (0,0,0) with constant, and it should perform difference. Also, it has 1 significant lag.

auto.arima(sp500)

## Series: sp500   
## ARIMA(0,1,0) with drift   
##   
## Coefficients:  
## drift  
## 10.2383  
## s.e. 5.0631  
##   
## sigma^2 estimated as 4102: log likelihood=-886.48  
## AIC=1776.97 AICc=1777.04 BIC=1783.1

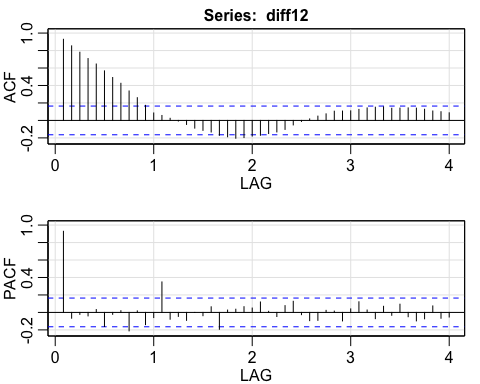
sarima(sp500, 2,1,3)

## initial value 4.162672   
## iter 2 value 4.162269  
## iter 3 value 4.162262  
## iter 4 value 4.162249  
## iter 5 value 4.162249  
## iter 6 value 4.162216  
## iter 7 value 4.162135  
## iter 8 value 4.161881  
## iter 9 value 4.161828  
## iter 10 value 4.161427  
## iter 11 value 4.161417  
## iter 12 value 4.161386  
## iter 13 value 4.161318  
## iter 14 value 4.161082  
## iter 15 value 4.160712  
## iter 16 value 4.160388  
## iter 17 value 4.160205  
## iter 18 value 4.160128  
## iter 19 value 4.160024  
## iter 20 value 4.158479  
## iter 21 value 4.158379  
## iter 22 value 4.156992  
## iter 23 value 4.156461  
## iter 24 value 4.155913  
## iter 25 value 4.155764  
## iter 26 value 4.155348  
## iter 27 value 4.152792  
## iter 28 value 4.152069  
## iter 29 value 4.151582  
## iter 30 value 4.150904  
## iter 31 value 4.150900  
## iter 32 value 4.149378  
## iter 33 value 4.148161  
## iter 34 value 4.146283  
## iter 35 value 4.144614  
## iter 36 value 4.142031  
## iter 37 value 4.141247  
## iter 38 value 4.140283  
## iter 39 value 4.140051  
## iter 40 value 4.138449  
## iter 41 value 4.137712  
## iter 42 value 4.137388  
## iter 43 value 4.136609  
## iter 44 value 4.134560  
## iter 45 value 4.133854  
## iter 46 value 4.132668  
## iter 47 value 4.129378  
## iter 48 value 4.128633  
## iter 49 value 4.127435  
## iter 50 value 4.127252  
## iter 51 value 4.126609  
## iter 52 value 4.125581  
## iter 53 value 4.125028  
## iter 54 value 4.124128  
## iter 55 value 4.122543  
## iter 56 value 4.122516  
## iter 57 value 4.120507  
## iter 58 value 4.120494  
## iter 59 value 4.120185  
## iter 60 value 4.120169  
## iter 61 value 4.119969  
## iter 62 value 4.119959  
## iter 63 value 4.119243  
## iter 64 value 4.119200  
## iter 64 value 4.119200  
## iter 65 value 4.116672  
## iter 66 value 4.116668  
## iter 67 value 4.116531  
## iter 68 value 4.115513  
## iter 68 value 4.115513  
## iter 69 value 4.114776  
## iter 70 value 4.114763  
## iter 71 value 4.114763  
## iter 71 value 4.114763  
## iter 72 value 4.114506  
## iter 73 value 4.114504  
## iter 74 value 4.114495  
## iter 74 value 4.114495  
## iter 75 value 4.114387  
## iter 76 value 4.114359  
## iter 77 value 4.114357  
## iter 77 value 4.114357  
## iter 78 value 4.114248  
## iter 79 value 4.114247  
## iter 80 value 4.114198  
## iter 80 value 4.114198  
## iter 81 value 4.114094  
## iter 82 value 4.114093  
## iter 83 value 4.114035  
## iter 83 value 4.114035  
## iter 84 value 4.113955  
## iter 85 value 4.113955  
## iter 86 value 4.113883  
## iter 87 value 4.113830  
## iter 87 value 4.113830  
## iter 88 value 4.113698  
## iter 89 value 4.113696  
## iter 90 value 4.113647  
## iter 91 value 4.113644  
## iter 92 value 4.113562  
## iter 93 value 4.113559  
## iter 94 value 4.113502  
## iter 95 value 4.113292  
## iter 95 value 4.113292  
## iter 96 value 4.113143  
## iter 97 value 4.113140  
## iter 98 value 4.113078  
## iter 98 value 4.113078  
## iter 99 value 4.112930  
## iter 100 value 4.112915  
## final value 4.112915   
## stopped after 100 iterations  
## initial value 4.156427   
## iter 2 value 4.156052  
## iter 3 value 4.156046  
## iter 4 value 4.156034  
## iter 5 value 4.156033  
## iter 6 value 4.156023  
## iter 7 value 4.156008  
## iter 8 value 4.155979  
## iter 9 value 4.155918  
## iter 10 value 4.155860  
## iter 11 value 4.155817  
## iter 12 value 4.155814  
## iter 13 value 4.155809  
## iter 14 value 4.155794  
## iter 15 value 4.155740  
## iter 16 value 4.155647  
## iter 17 value 4.155430  
## iter 18 value 4.154916  
## iter 19 value 4.154746  
## iter 20 value 4.154474  
## iter 21 value 4.154429  
## iter 22 value 4.154372  
## iter 23 value 4.154075  
## iter 24 value 4.153219  
## iter 25 value 4.152842  
## iter 26 value 4.152522  
## iter 27 value 4.152048  
## iter 28 value 4.151146  
## iter 29 value 4.150978  
## iter 30 value 4.150833  
## iter 31 value 4.150623  
## iter 32 value 4.150576  
## iter 33 value 4.150503  
## iter 34 value 4.150111  
## iter 35 value 4.149642  
## iter 36 value 4.149108  
## iter 37 value 4.148623  
## iter 38 value 4.148276  
## iter 39 value 4.147858  
## iter 40 value 4.147334  
## iter 41 value 4.146635  
## iter 42 value 4.144534  
## iter 43 value 4.143711  
## iter 44 value 4.142606  
## iter 45 value 4.142487  
## iter 46 value 4.142381  
## iter 47 value 4.142056  
## iter 48 value 4.141875  
## iter 49 value 4.141851  
## iter 50 value 4.141821  
## iter 51 value 4.141514  
## iter 52 value 4.140778  
## iter 53 value 4.140500  
## iter 54 value 4.140242  
## iter 55 value 4.139461  
## iter 56 value 4.139202  
## iter 57 value 4.139170  
## iter 58 value 4.139169  
## iter 59 value 4.139169  
## iter 60 value 4.139168  
## iter 61 value 4.139166  
## iter 62 value 4.139165  
## iter 63 value 4.139162  
## iter 64 value 4.139161  
## iter 65 value 4.139161  
## iter 66 value 4.139161  
## iter 66 value 4.139161  
## iter 66 value 4.139161  
## final value 4.139161   
## converged



## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = constant, optim.control = list(trace = trc, REPORT = 1,   
## reltol = tol))  
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 ma3 constant  
## -0.2898 -0.9447 0.2983 0.9789 -0.0591 10.0926  
## s.e. 0.0378 0.0320 0.0856 0.0442 0.0836 4.8878  
##   
## sigma^2 estimated as 3852: log likelihood = -883.74, aic = 1781.48  
##   
## $degrees\_of\_freedom  
## [1] 153  
##   
## $ttable  
## Estimate SE t.value p.value  
## ar1 -0.2898 0.0378 -7.6634 0.0000  
## ar2 -0.9447 0.0320 -29.4835 0.0000  
## ma1 0.2983 0.0856 3.4841 0.0006  
## ma2 0.9789 0.0442 22.1604 0.0000  
## ma3 -0.0591 0.0836 -0.7066 0.4809  
## constant 10.0926 4.8878 2.0649 0.0406  
##   
## $AIC  
## [1] 9.331276  
##   
## $AICc  
## [1] 9.348382  
##   
## $BIC  
## [1] 8.446595

diff12 <- diff(sp500, 12)  
acf2(diff12)

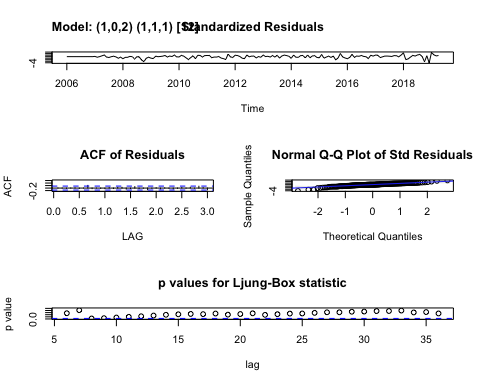


## ACF PACF  
## [1,] 0.93 0.93  
## [2,] 0.85 -0.07  
## [3,] 0.78 -0.02  
## [4,] 0.71 -0.04  
## [5,] 0.65 0.03  
## [6,] 0.57 -0.16  
## [7,] 0.49 -0.02  
## [8,] 0.43 0.02  
## [9,] 0.34 -0.21  
## [10,] 0.26 0.02  
## [11,] 0.17 -0.14  
## [12,] 0.09 -0.06  
## [13,] 0.06 0.35  
## [14,] 0.02 -0.08  
## [15,] -0.01 -0.05  
## [16,] -0.05 -0.09  
## [17,] -0.09 0.00  
## [18,] -0.11 -0.04  
## [19,] -0.13 0.06  
## [20,] -0.17 -0.19  
## [21,] -0.19 0.03  
## [22,] -0.20 0.04  
## [23,] -0.20 0.07  
## [24,] -0.18 0.05  
## [25,] -0.17 0.12  
## [26,] -0.15 0.01  
## [27,] -0.13 -0.05  
## [28,] -0.10 0.08  
## [29,] -0.05 0.13  
## [30,] -0.01 -0.03  
## [31,] 0.02 -0.09  
## [32,] 0.05 -0.09  
## [33,] 0.08 0.02  
## [34,] 0.11 0.01  
## [35,] 0.11 -0.10  
## [36,] 0.11 0.04  
## [37,] 0.13 0.12  
## [38,] 0.14 0.03  
## [39,] 0.15 -0.07  
## [40,] 0.16 0.07  
## [41,] 0.14 -0.03  
## [42,] 0.14 0.09  
## [43,] 0.14 -0.05  
## [44,] 0.14 -0.10  
## [45,] 0.13 -0.07  
## [46,] 0.11 0.07  
## [47,] 0.10 -0.07  
## [48,] 0.09 -0.05

For ACF, it is still highly auto-correlated, For PACF, it is low auto-correlation

mod1 <- sarima(sp500, 1,0,2,1,1,1,12)

## initial value 5.484889   
## iter 2 value 5.039941  
## iter 3 value 4.967923  
## iter 4 value 4.544991  
## iter 5 value 4.386038  
## iter 6 value 4.353940  
## iter 7 value 4.320190  
## iter 8 value 4.307682  
## iter 9 value 4.301571  
## iter 10 value 4.294652  
## iter 11 value 4.292263  
## iter 12 value 4.290152  
## iter 13 value 4.289396  
## iter 14 value 4.287608  
## iter 15 value 4.287452  
## iter 16 value 4.286706  
## iter 17 value 4.286402  
## iter 18 value 4.286290  
## iter 19 value 4.286107  
## iter 20 value 4.285865  
## iter 21 value 4.285686  
## iter 22 value 4.285640  
## iter 23 value 4.285638  
## iter 23 value 4.285638  
## iter 23 value 4.285638  
## final value 4.285638   
## converged  
## initial value 4.298736   
## iter 2 value 4.290898  
## iter 3 value 4.287904  
## iter 4 value 4.282537  
## iter 5 value 4.277575  
## iter 6 value 4.276453  
## iter 7 value 4.275542  
## iter 8 value 4.274729  
## iter 9 value 4.274277  
## iter 10 value 4.274159  
## iter 11 value 4.274135  
## iter 12 value 4.274065  
## iter 13 value 4.274043  
## iter 14 value 4.274033  
## iter 15 value 4.274032  
## iter 16 value 4.274031  
## iter 17 value 4.274031  
## iter 18 value 4.274030  
## iter 19 value 4.274030  
## iter 20 value 4.274029  
## iter 21 value 4.274028  
## iter 22 value 4.274027  
## iter 23 value 4.274027  
## iter 24 value 4.274026  
## iter 24 value 4.274026  
## iter 24 value 4.274026  
## final value 4.274026   
## converged



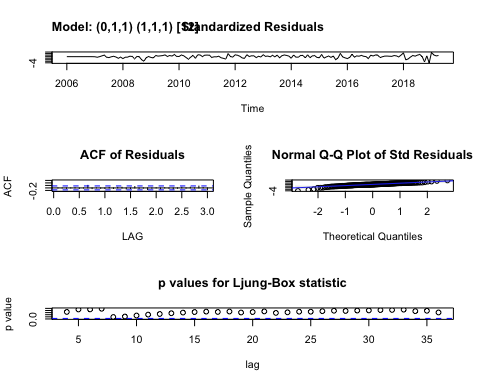
mod1

## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), xreg = constant, optim.control = list(trace = trc, REPORT = 1,   
## reltol = tol))  
##   
## Coefficients:  
## ar1 ma1 ma2 sar1 sma1 constant  
## 0.9856 0.0036 0.0036 0.0655 -0.9999 10.1436  
## s.e. 0.0220 0.0839 0.0840 0.1014 0.1837 3.4400  
##   
## sigma^2 estimated as 4206: log likelihood = -842.56, aic = 1699.12  
##   
## $degrees\_of\_freedom  
## [1] 142  
##   
## $ttable  
## Estimate SE t.value p.value  
## ar1 0.9856 0.0220 44.7653 0.0000  
## ma1 0.0036 0.0839 0.0432 0.9656  
## ma2 0.0036 0.0840 0.0430 0.9658  
## sar1 0.0655 0.1014 0.6465 0.5190  
## sma1 -0.9999 0.1837 -5.4416 0.0000  
## constant 10.1436 3.4400 2.9488 0.0037  
##   
## $AIC  
## [1] 9.419184  
##   
## $AICc  
## [1] 9.436289  
##   
## $BIC  
## [1] 8.534503

The coefficient explains the change of independent variable influcne the change of depedent variable. The plot is stationary.

mod2 <- sarima(sp500, 0,1,1,1,1,1,12)

## initial value 4.489714   
## iter 2 value 4.325868  
## iter 3 value 4.317266  
## iter 4 value 4.303021  
## iter 5 value 4.302261  
## iter 6 value 4.301216  
## iter 7 value 4.301116  
## iter 8 value 4.301114  
## iter 8 value 4.301114  
## final value 4.301114   
## converged  
## initial value 4.287563   
## iter 2 value 4.283955  
## iter 3 value 4.279471  
## iter 4 value 4.279235  
## iter 5 value 4.279195  
## iter 6 value 4.279194  
## iter 6 value 4.279194  
## iter 6 value 4.279194  
## final value 4.279194   
## converged

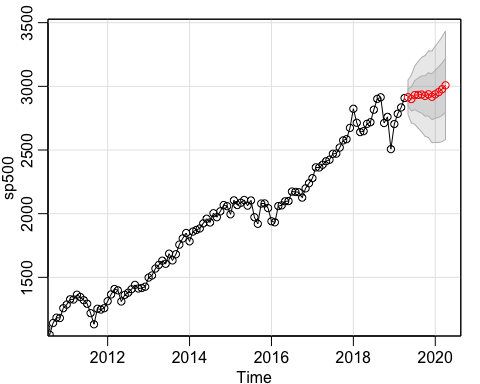


mod2

## $fit  
##   
## Call:  
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,   
## Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,   
## REPORT = 1, reltol = tol))  
##   
## Coefficients:  
## ma1 sar1 sma1  
## 0.0059 0.0374 -0.8869  
## s.e. 0.0823 0.1220 0.1544  
##   
## sigma^2 estimated as 4634: log likelihood = -837.63, aic = 1683.25  
##   
## $degrees\_of\_freedom  
## [1] 144  
##   
## $ttable  
## Estimate SE t.value p.value  
## ma1 0.0059 0.0823 0.0715 0.9431  
## sar1 0.0374 0.1220 0.3067 0.7595  
## sma1 -0.8869 0.1544 -5.7439 0.0000  
##   
## $AIC  
## [1] 9.478636  
##   
## $AICc  
## [1] 9.492749  
##   
## $BIC  
## [1] 8.536296

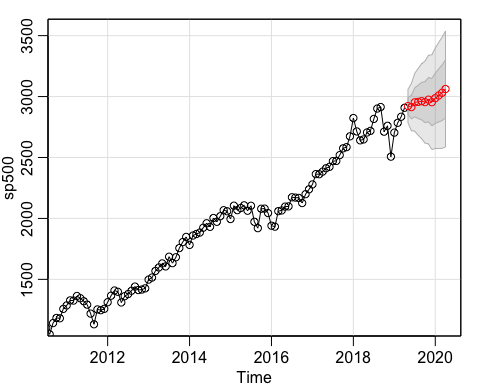
The coeffients are similar and it’s also stationary

sarima.for(sp500, n.ahead = 12, 1,0,2,1,1,1,12 )



## $pred  
## Jan Feb Mar Apr May Jun Jul  
## 2019 2914.706 2899.917 2931.430  
## 2020 2937.858 2954.663 2977.272 3008.302   
## Aug Sep Oct Nov Dec  
## 2019 2932.679 2935.855 2924.526 2937.505 2917.529  
## 2020   
##   
## $se  
## Jan Feb Mar Apr May Jun Jul  
## 2019 67.18317 94.37517 114.80980  
## 2020 189.33144 197.96895 205.96690 213.40361   
## Aug Sep Oct Nov Dec  
## 2019 131.56083 145.91450 158.53394 169.81579 180.02085  
## 2020

sarima.for(sp500, n.ahead = 12, 0,1,1,1,1,1,12 )



## $pred  
## Jan Feb Mar Apr May Jun Jul  
## 2019 2921.996 2913.632 2953.036  
## 2020 2986.521 3008.785 3031.026 3062.231   
## Aug Sep Oct Nov Dec  
## 2019 2956.426 2962.221 2954.277 2974.539 2953.685  
## 2020   
##   
## $se  
## Jan Feb Mar Apr May Jun Jul  
## 2019 68.39996 97.01720 118.93746  
## 2020 206.27362 217.42260 228.02712 238.15992   
## Aug Sep Oct Nov Dec  
## 2019 137.40415 153.66744 168.36705 181.88251 194.46086  
## 2020

These two are predict values 12 months in the future