Week 3 Homework

Matthew Dunne

January 25, 2019

Week 3: Homework Assignment

This assignment helps understanding linear models for time series

This assignment is individual

1. Exercise 2 on page 125

Consider the monthly simple returns of CRSP Decile 1, 2, 5, 9, and 10 portfolios based on the market capitalization of NYSE/AMEX/NASDAQ. The data span is from January 1961 to September 2011.

```
datapath<-"C:/Users/mjdun/Desktop/Financial Analytics/Week 3"
data1=read.table(file=paste(datapath,"m-dec125910-6111.txt",sep="/"),header=T)
#load the library you will need
suppressWarnings(library(TSA))
```

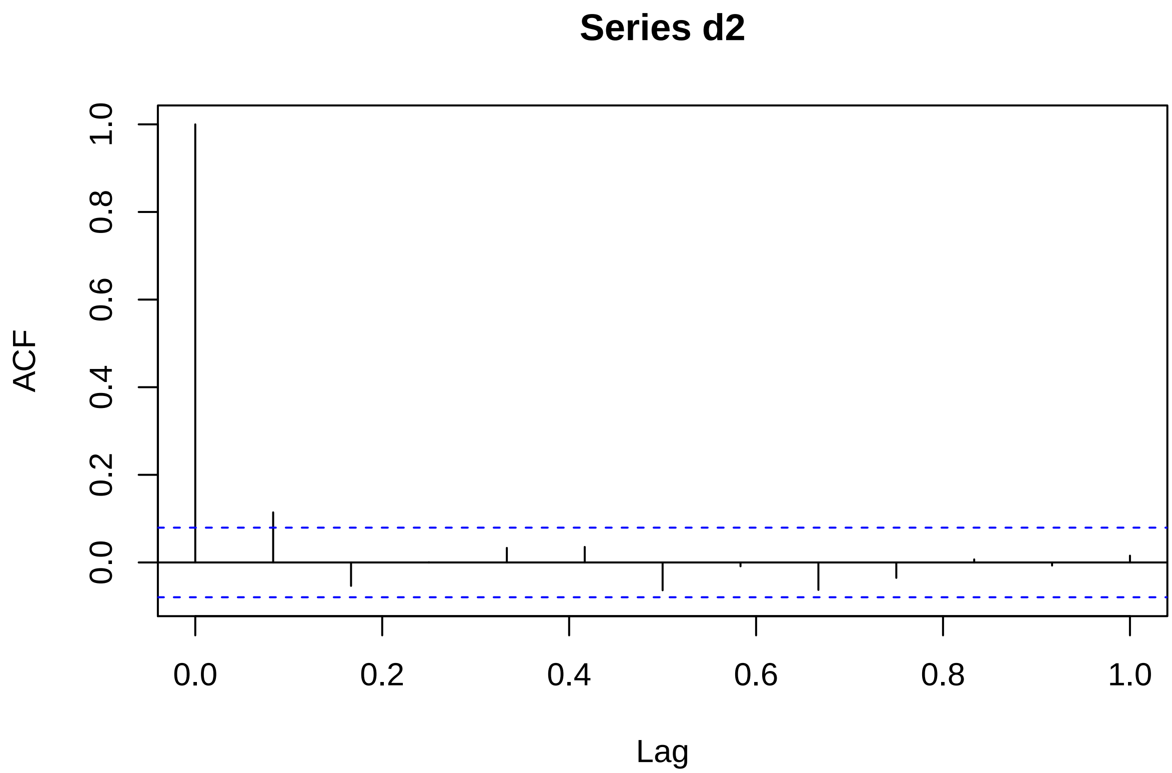
```
##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```
#separate out for Decile 2, 10
d2<-data1$dec2
d10<-data1$dec10
#convert to time series
d2<-ts(d2, frequency = 12, start=c(1961, 1))
d10<-ts(d10, frequency = 12, start=c(1961, 1))
```

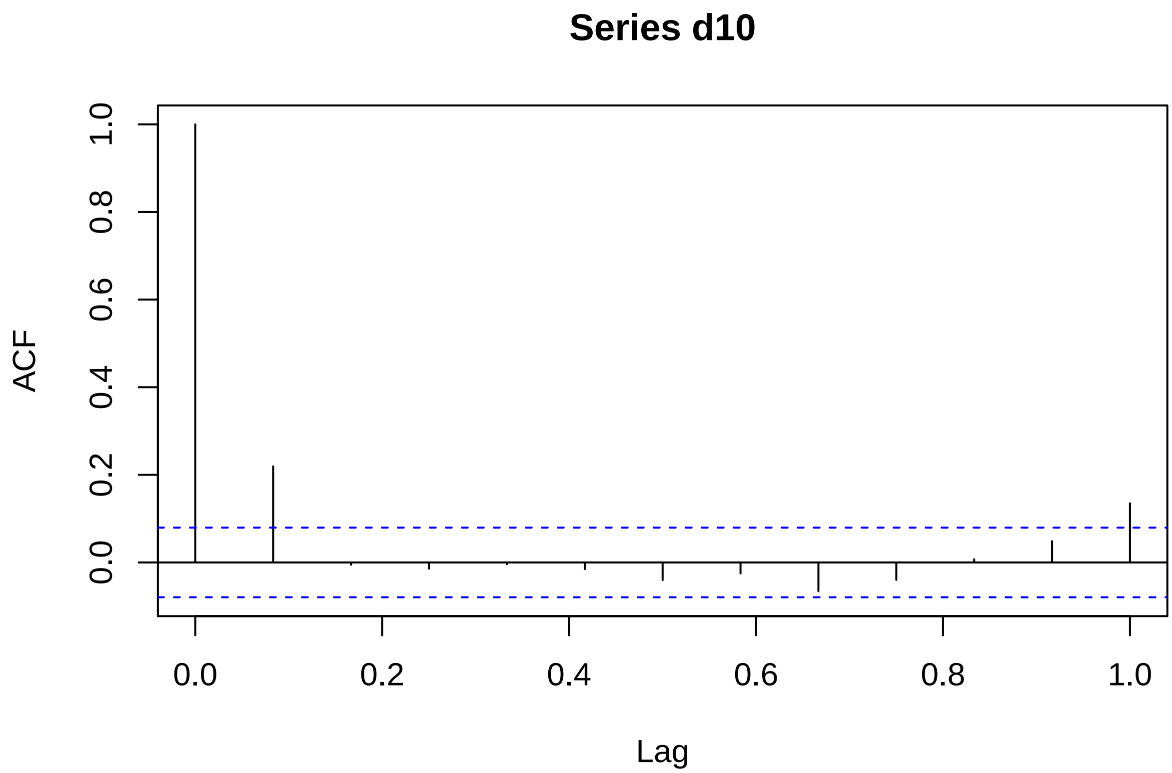
(a) For the return series of Decile 2 and Decile 10, test the null hypothesis that the first 12 lags of autocorrelations are 0 at the 5% level. Draw your conclusion.

Plot the ACF to see what it looks like.

```
f2<-acf(d2, lag = 12, drop.lag.0 = FALSE)
```




```
f10<-acf(d10, lag = 12, drop.lag.0 = FALSE)
```



For both deciles we see a possible autocorrelation at lag=1. For decile 10 we also see a possible autocorrelation at lag=12. The dashed lines represent plus or minus two standard deviations away from zero.

There are two ways to check this.

The first is by getting the tt statistic for these possible lags and comparing them to the relevant critical value (1.959964 at the 5% level). If the tt stat is greater than the critical value it means we are seeing a very unusual event given our assumptions - meaning we would reject the null hypothesis that the autocorrelation for that lag=0.

```
#the tt stat for lag=1 in the Decile 2, Decile 10 data
d2tt<-f2$acf[2]*sqrt(length(d2))
d10tt<-f10$acf[2]*sqrt(length(d10))
cbind(d2tt, d10tt)
```

```
##          d2tt    d10tt
## [1,] 2.822691 5.412805
```

Both are clearly above the critical value. So, we would **reject the null hypothesis that the first 12 lags of autocorrelations are 0 at the 5% level.**

The other way to check whether all autocorrelation lags are 0 is to do the Box-Ljung test.

```
Box.test(d2, lag=1, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  d2
## X-squared = 8.0069, df = 1, p-value = 0.00466
```

```r
Box.test(d10, lag=1, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  d10
## X-squared = 29.443, df = 1, p-value = 5.759e-08
```

We see for both of these deciles at lag=1 the p-value is well below 5%, confirming that we would reject the null hypothesis that the autocorrelation at all lags is 0.

(b) Build an ARMA model for the return series of Decile 2. Perform model checking and write down the fitted model.

To get a sense of possible models, use the eacf function.

```r
first_eacf<-eacf(d2, 12, 12)
```

```
## AR/MA
##    0 1 2 3 4 5 6 7 8 9 10 11 12
## 0  x o o o o o o o o o o  o  o
## 1  x o o o o o o o o o o  o  o
## 2  x x o o o o o o o o o  o  o
## 3  x o x o o o o o o o o  o  o
## 4  x o x o o o o o o o o  o  o
## 5  x x x x x o o o o o o  o  o
## 6  x x o x o x o o o o o  o  o
## 7  x x o x x x x o o o o  o  o
## 8  x x x x x o o x o o o  o  o
## 9  x o o o x o x x x o o  o  o
## 10 x o o o x x o o x x o  o  o
## 11 x x o o x x o o x o o  o  o
## 12 x o o x x x o o o o o  x  o
```

This would suggest a model of MA(1) or ARMA(1,1). But it might help to get a sense of which EACF values are barely above the threshold (i.e. borderline cases marked “x” that could be marked “o”).

```r
#find the threshold
Compare.with<-2/sqrt(length(d2))
Compare.with
```

```
## [1] 0.08104409
```

```r
print(abs(first_eacf$eacf)-Compare.with, digits = 2)
```

```
##        [,1]    [,2]   [,3]   [,4]   [,5]    [,6]   [,7]     [,8]   [,9]
##  [1,] 0.033 -0.0275 -0.081 -0.048 -0.046 -0.0175 -0.072 -0.01831 -0.046
##  [2,] 0.354 -0.0277 -0.080 -0.064 -0.060 -0.0102 -0.070 -0.03340 -0.038
##  [3,] 0.130  0.1130 -0.020 -0.068 -0.064 -0.0302 -0.075 -0.01818 -0.035
##  [4,] 0.334 -0.0165  0.077 -0.057 -0.076 -0.0258 -0.072  0.00032 -0.034
##  [5,] 0.419 -0.0062  0.188 -0.027 -0.069 -0.0124 -0.073 -0.03958 -0.044
##  [6,] 0.278  0.2865  0.220  0.084  0.187 -0.0069 -0.062 -0.07960 -0.059
##  [7,] 0.077  0.3909 -0.047  0.303 -0.013  0.1337 -0.052 -0.07153 -0.078
##  [8,] 0.063  0.3782 -0.037  0.302  0.069  0.0292  0.063 -0.05634 -0.075
##  [9,] 0.176  0.0792  0.050  0.143  0.172 -0.0555 -0.042  0.03818 -0.026
## [10,] 0.240 -0.0080 -0.036 -0.074  0.190 -0.0619  0.019  0.07584  0.062
## [11,] 0.413 -0.0287 -0.052 -0.079  0.134  0.1361 -0.081 -0.07150  0.115
## [12,] 0.259  0.0890 -0.075 -0.062  0.172  0.1128 -0.029 -0.07223  0.096
```

```
## [13,] 0.268 -0.0633 -0.011  0.026  0.153  0.1315 -0.069 -0.03579 -0.033
##        [,10]  [,11]  [,12]  [,13]
##  [1,] -0.074 -0.074 -0.066 -0.039
##  [2,] -0.070 -0.078 -0.075 -0.051
##  [3,] -0.071 -0.078 -0.076 -0.064
##  [4,] -0.072 -0.072 -0.076 -0.059
##  [5,] -0.072 -0.075 -0.058 -0.056
##  [6,] -0.046 -0.063 -0.080 -0.080
##  [7,] -0.038 -0.079 -0.078 -0.074
##  [8,] -0.040 -0.051 -0.074 -0.068
##  [9,] -0.039 -0.065 -0.069 -0.065
## [10,] -0.046 -0.045 -0.063 -0.070
## [11,]  0.011 -0.060 -0.066 -0.081
## [12,] -0.003 -0.057 -0.057 -0.067
## [13,] -0.042 -0.057  0.100 -0.066
```

For those borderline cases we should see a very small positive number, meaning the value was just above the threshold. We see no such values in the vicinity of our previously identified models [MA(1), ARMA(1,1)].

So compare those models.

```r
#MA(1)
arima(d2, order = c(0,0,1))
```

```
##
## Call:
## arima(x = d2, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.1307     0.0093
## s.e.  0.0425     0.0022
##
## sigma^2 estimated as 0.002223:  log likelihood = 996.04,  aic = -1988.08
```

```r
#ARMA(1,1)
arima(d2, order = c(1,0,1))
```

```
##
## Call:
## arima(x = d2, order = c(1, 0, 1))
##
## Coefficients:
##           ar1     ma1  intercept
##       -0.4039  0.5265     0.0093
## s.e.   0.3885  0.3639     0.0021
##
## sigma^2 estimated as 0.002217:  log likelihood = 996.84,  aic = -1987.68
```

I would choose the MA(1) model. Looking at the Standard Error for the AR(1) coefficient in the ARMA model, it seems that the coefficient is not significant. Also the AIC of the ARMA model is slightly higher.

The fitted model would be:

$$X_t = \mu + \epsilon_t + \theta\epsilon_{t-1}$$

, or

$$0.0093 + \epsilon_t + 0.1307\epsilon_{t-1}$$

(c) Use the fitted ARMA model to produce 1- to 12-step ahead forecasts of the series and the associated standard errors of forecasts.

```
#fit the model to all but the last 12 data points for Decile 2
modelfit<-arima(d2[1:597], order = c(0,0,1))
#check that the coefficients don't change that much
modelfit
```

```
##
## Call:
## arima(x = d2[1:597], order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.1230     0.0095
## s.e.  0.0434     0.0022
##
## sigma^2 estimated as 0.002237:  log likelihood = 974.46,  aic = -1944.91
```

Our coefficients are very close to what we had when we fit the model on all the data. Make the prediction 12 steps out.

```
pred<-predict(modelfit, 12)
cbind(Actual=d2[598:609], Predicted=pred$pred, Residuals=(d2[598:609]-pred$pred), S.E.=pred$se)
```

```
## Time Series:
## Start = 598
## End = 609
## Frequency = 1
##        Actual   Predicted    Residuals       S.E.
## 598  0.036242 0.021960199  0.014281801 0.04730190
## 599  0.012035 0.009461795  0.002573205 0.04765859
## 600  0.065768 0.009461795  0.056306205 0.04765859
## 601  0.018406 0.009461795  0.008944205 0.04765859
## 602  0.038320 0.009461795  0.028858205 0.04765859
## 603  0.005935 0.009461795 -0.003526795 0.04765859
## 604  0.031454 0.009461795  0.021992205 0.04765859
## 605 -0.001103 0.009461795 -0.010564795 0.04765859
## 606 -0.014384 0.009461795 -0.023845795 0.04765859
## 607 -0.036875 0.009461795 -0.046336795 0.04765859
## 608 -0.059275 0.009461795 -0.068736795 0.04765859
## 609 -0.080109 0.009461795 -0.089570795 0.04765859
```

We see that the forecast converges to the unconditional mean when our time horizon exceeds the q in our MA(q) model.

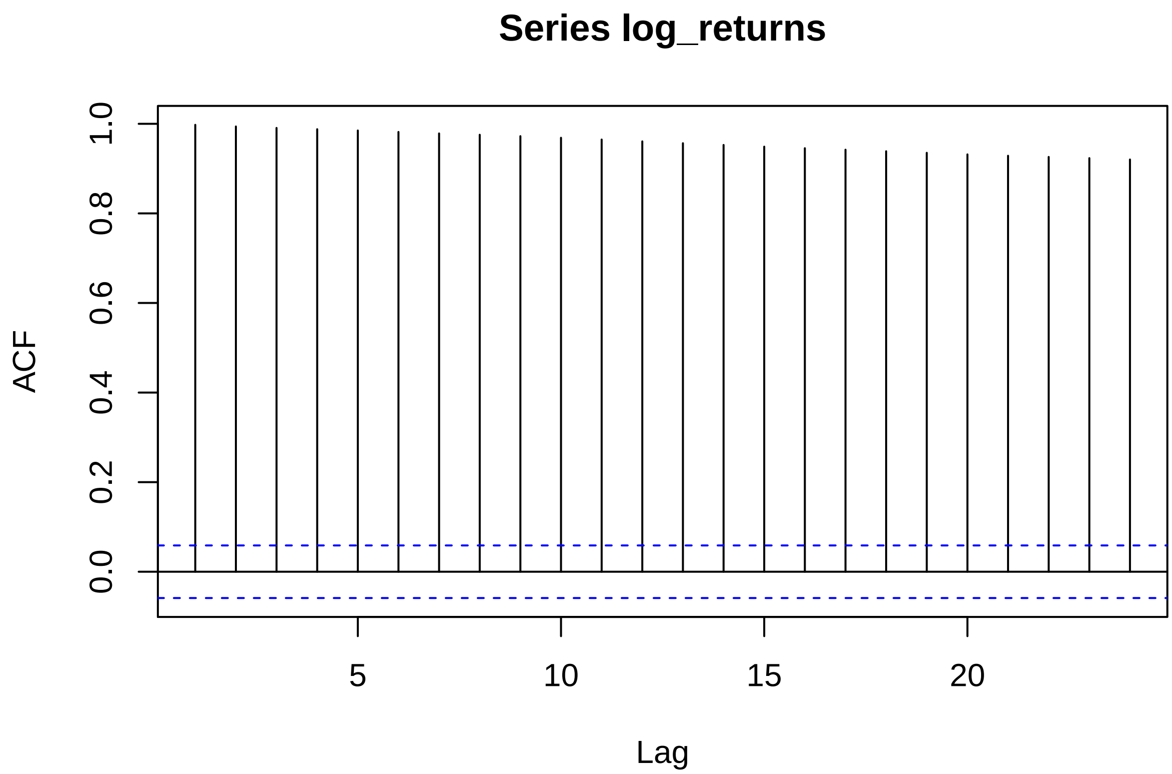
2. Exercise 4 on page 126

Consider the monthly yields of Moody's Aaa & Baa seasoned bonds from January 1919 to November, 2011. The data are obtained from FRED of Federal Reserve Bank of St. Louis. Consider the log series of monthly Aaa bond yields. Build a time series model for the series, including model checking.

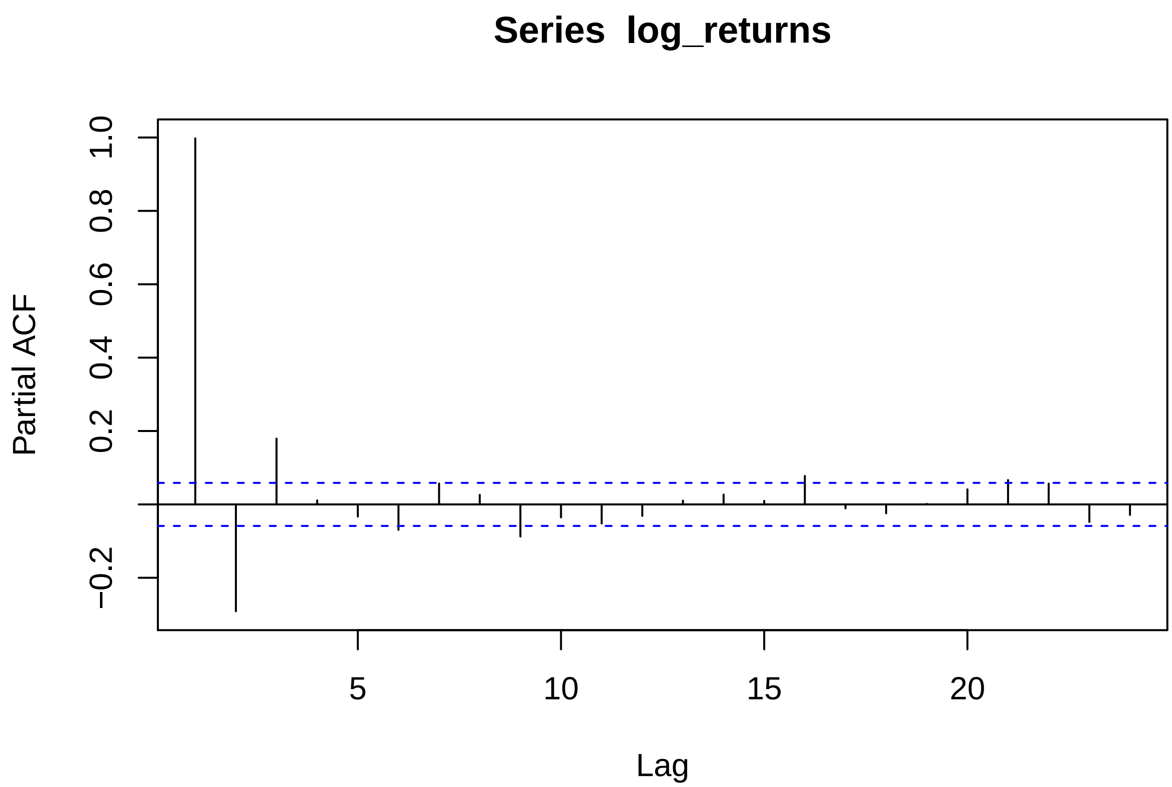
```
data2<-read.table(file=paste(datapath,"m-aaa-1911.txt",sep="/"),header=T)
#convert the relevant column to decimals
data2<-data2$yield/100
log_returns<-log(data2+1)
```

Let us look at the ACF and PACF plots. Use a maximum lag of 24 as this is monthly data.

```
acf(log_returns, lag=24)
```




```
pacf(log_returns, lag=24)
```



Given the gradual decay in the ACF and the fact that the PACF appears to be 0 for lags > 3 (with some possible exceptions at lags = 6, 9, 16) this appears to be at least an AR(3) model and possibly AR(6) or AR(9).

The gradual decay in the ACF also suggests this is not a pure MA model, and so we shall not consider any. We will however, consider several ARMA models.

First the AR(9) model:

```
#look at an unrestricted model with AR up to 9
ar_model<-arima(log_returns, order=c(9, 0, 0))
ar_model
```

```
##
## Call:
## arima(x = log_returns, order = c(9, 0, 0))
##
## Coefficients:
##          ar1      ar2     ar3     ar4     ar5      ar6     ar7     ar8
##       1.4110  -0.6240  0.2222  0.0077  0.0639  -0.1252  0.0025  0.1248
## s.e.  0.0298   0.0516  0.0549  0.0552  0.0553   0.0553  0.0550  0.0518
##           ar9  intercept
##       -0.0857     0.0568
## s.e.   0.0299     0.0132
##
## sigma^2 estimated as 2.167e-06:  log likelihood = 5685.86,  aic = -11351.73
```

Many of these coefficients of these lags are not significant given the standard errors. Restrict them.

Restricted AR(9) model:

```r
ar_restricted<-arima(log_returns, order=c(9, 0, 0), fixed=c(NA, NA, NA, 0, 0, NA, 0, 0, NA, NA))
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : some AR parameters were fixed: setting transform.pars = FALSE
```

```r
ar_restricted
```

```
##
## Call:
## arima(x = log_returns, order = c(9, 0, 0), fixed = c(NA, NA, NA, 0, 0, NA, 0,
##     0, NA, NA))
##
## Coefficients:
##          ar1      ar2     ar3  ar4  ar5      ar6  ar7  ar8     ar9
##       1.4140  -0.6453  0.2668    0    0  -0.0382    0    0  0.0000
## s.e.  0.0292   0.0479  0.0336    0    0   0.0193    0    0  0.0136
##       intercept
##          0.0558
## s.e.     0.0132
##
## sigma^2 estimated as 2.194e-06:  log likelihood = 5679.1,  aic = -11346.2
```

We see that the model does not improve in terms of AIC (from -11351.73 to -11346.2), and that some of the remaining coefficients, those for lags 6 and 9 no longer appear significant. So let us compare with a simple AR(3) model

```r
ar_three<-arima(log_returns, order=c(3, 0, 0))
ar_three
```

```
##
## Call:
## arima(x = log_returns, order = c(3, 0, 0))
##
## Coefficients:
##          ar1      ar2     ar3  intercept
##       1.4063  -0.6289  0.2200     0.0568
## s.e.  0.0292   0.0477  0.0292     0.0143
##
## sigma^2 estimated as 2.21e-06:  log likelihood = 5675.16,  aic = -11342.32
```

All coefficients are significant and the AIC is only slightly larger.

Let us check if the residuals are white noise under each model.

```r
Box.test(ar_model$residuals, lag = 12, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  ar_model$residuals
## X-squared = 7.8622, df = 12, p-value = 0.7958
```

```r
Box.test(ar_restricted$residuals, lag = 12, type = "Ljung")
```

```
##
##  Box-Ljung test
##
```

```
## data:  ar_restricted$residuals
## X-squared = 21.922, df = 12, p-value = 0.0384
```

```
Box.test(ar_three$residuals, lag = 12, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  ar_three$residuals
## X-squared = 29.056, df = 12, p-value = 0.003866
```

Then we adjust the degrees of freedom.

```
#df=12-9=3 because you had 9 coefficients
ar_model_pval<-1-pchisq(7.8622, 3)
#df=12-5=7 because you had 5 coefficients
ar_restricted_pval<-1-pchisq(21.922, 7)
#df=12-3=9 because you had 3 coefficients
ar_three_pval<-1-pchisq(29.056, 9)
cbind(ar_model_pval, ar_restricted_pval, ar_three_pval)
```

```
##      ar_model_pval ar_restricted_pval ar_three_pval
## [1,]    0.04894714        0.002620267  0.0006340959
```

After adjusting we see that for all models we would reject the Null Hypothesis (at the 5% level) of the residuals being white noise. In essence there is some signal we are not picking up with just our AR models. The only one that is borderline is the one that makes little intuitive sense: AR(9).

Let us look at some ARIMA models instead.

```
log_returns_eacf<-eacf(log_returns, 12, 12)
```

```
## AR/MA
##    0 1 2 3 4 5 6 7 8 9 10 11 12
## 0  x x x x x x x x x x x  x  x
## 1  x x x o x o o o x x o  o  o
## 2  x x x o x o o o o o o  o  o
## 3  x x x o o o o x o o x  o  o
## 4  x x x o o o o o x o o  o  o
## 5  x x x x o o o o x o o  o  o
## 6  x x x x o o o o x o o  o  o
## 7  x x x o x o o o o o o  o  o
## 8  o x x o x o o o o o o  o  o
## 9  x x o x x o o o o o o  o  o
## 10 x x o x o x o o x o o  o  x
## 11 x o x x o x o x o o o  o  x
## 12 x x x x x x o o o x o  x  o
```

This suggests something like an ARMA(1,3) or ARMA(2,3) model. Let us also see if there are any borderline cases.

```
Compare.with<-2/sqrt(length(log_returns))
print(abs(log_returns_eacf$eacf)-Compare.with, digits = 2)
```

```
##         [,1]    [,2]    [,3]   [,4]    [,5]    [,6]    [,7]    [,8]
##  [1,] 0.9378  0.9342  0.9309  0.928  0.9252  0.9219  0.9187  0.9157
##  [2,] 0.2764  0.0230  0.0237 -0.015  0.0472 -0.0580 -0.0034 -0.0122
##  [3,] 0.4148  0.2558  0.0185 -0.012  0.0467 -0.0557 -0.0343 -0.0214
##  [4,] 0.0720  0.1709  0.1450 -0.054 -0.0076 -0.0203 -0.0389  0.0130
```

```
##  [5,]  0.3363  0.0008  0.1485 -0.043 -0.0284 -0.0597 -0.0440 -0.0036
##  [6,]  0.4411  0.0942  0.2289  0.159 -0.0209 -0.0570 -0.0512 -0.0226
##  [7,]  0.4404  0.3006  0.1595  0.153 -0.0012 -0.0168 -0.0502 -0.0282
##  [8,]  0.0029  0.2515  0.2050 -0.046  0.0967 -0.0095 -0.0076 -0.0458
##  [9,] -0.0151  0.3874  0.2142 -0.035  0.0757 -0.0529 -0.0338 -0.0575
## [10,]  0.3256  0.1767 -0.0048  0.194  0.0719 -0.0443 -0.0256 -0.0588
## [11,]  0.4355  0.0494 -0.0338  0.358 -0.0380  0.0141 -0.0297 -0.0267
## [12,]  0.4393 -0.0035  0.1184  0.351 -0.0347  0.0402 -0.0321  0.0039
## [13,]  0.4398  0.4256  0.3990  0.267  0.2085  0.0878 -0.0240 -0.0210
##           [,9]  [,10]    [,11]  [,12]   [,13]
##  [1,]  0.91251  0.909  0.90500  0.901  0.8968
##  [2,]  0.04620  0.029 -0.01017 -0.034 -0.0128
##  [3,] -0.00071 -0.030 -0.00089 -0.023 -0.0260
##  [4,] -0.02549 -0.042  0.00368 -0.036 -0.0282
##  [5,]  0.00584 -0.053 -0.00889 -0.049 -0.0288
##  [6,]  0.02002 -0.032 -0.05656 -0.044 -0.0534
##  [7,]  0.03684 -0.019 -0.04744 -0.043 -0.0545
##  [8,] -0.05540 -0.018 -0.05175 -0.022 -0.0519
##  [9,] -0.05742 -0.014 -0.05302 -0.032 -0.0274
## [10,] -0.04168 -0.027 -0.04073 -0.042 -0.0424
## [11,]  0.00742 -0.029 -0.05785 -0.036  0.0035
## [12,] -0.04979 -0.027 -0.02331 -0.022  0.0007
## [13,] -0.05134  0.055 -0.03303  0.037 -0.0236
```

We are looking for positive numbers that are very small. The only one we see corresponds to an ARMA (4,1) model which makes no intuitive sense.

Check our possible ARMA models.

```
arima(log_returns, order = c(1,0,3))
```

```
##
## Call:
## arima(x = log_returns, order = c(1, 0, 3))
##
## Coefficients:
##          ar1     ma1      ma2      ma3  intercept
##       0.9969  0.4114  -0.0477  -0.0715     0.0459
## s.e.  0.0023  0.0299   0.0327   0.0274     0.0179
##
## sigma^2 estimated as 2.215e-06:  log likelihood = 5673.86,  aic = -11337.72
```

```
arima(log_returns, order = c(2,0,3))
```

```
##
## Call:
## arima(x = log_returns, order = c(2, 0, 3))
##
## Coefficients:
##          ar1      ar2     ma1      ma2      ma3  intercept
##       1.1153  -0.1183  0.2949  -0.0948  -0.0709     0.0555
## s.e.  0.2122   0.2114  0.2103   0.0895   0.0281     0.0138
##
## sigma^2 estimated as 2.214e-06:  log likelihood = 5674.11,  aic = -11336.22
```

Both models look like they have coefficients that are not significant. Try getting rid of them for the ARMA (1,3) model.

```r
arma_model<-arima(log_returns, order = c(1,0,3), fixed = c(NA, NA, 0, NA, NA))
arma_model
```

```
##
## Call:
## arima(x = log_returns, order = c(1, 0, 3), fixed = c(NA, NA, 0, NA, NA))
##
## Coefficients:
##          ar1     ma1  ma2      ma3  intercept
##       0.9962  0.4295    0  -0.0587     0.0568
## s.e.  0.0022  0.0283    0   0.0259     0.0136
##
## sigma^2 estimated as 2.219e-06:  log likelihood = 5672.9,  aic = -11337.8
```

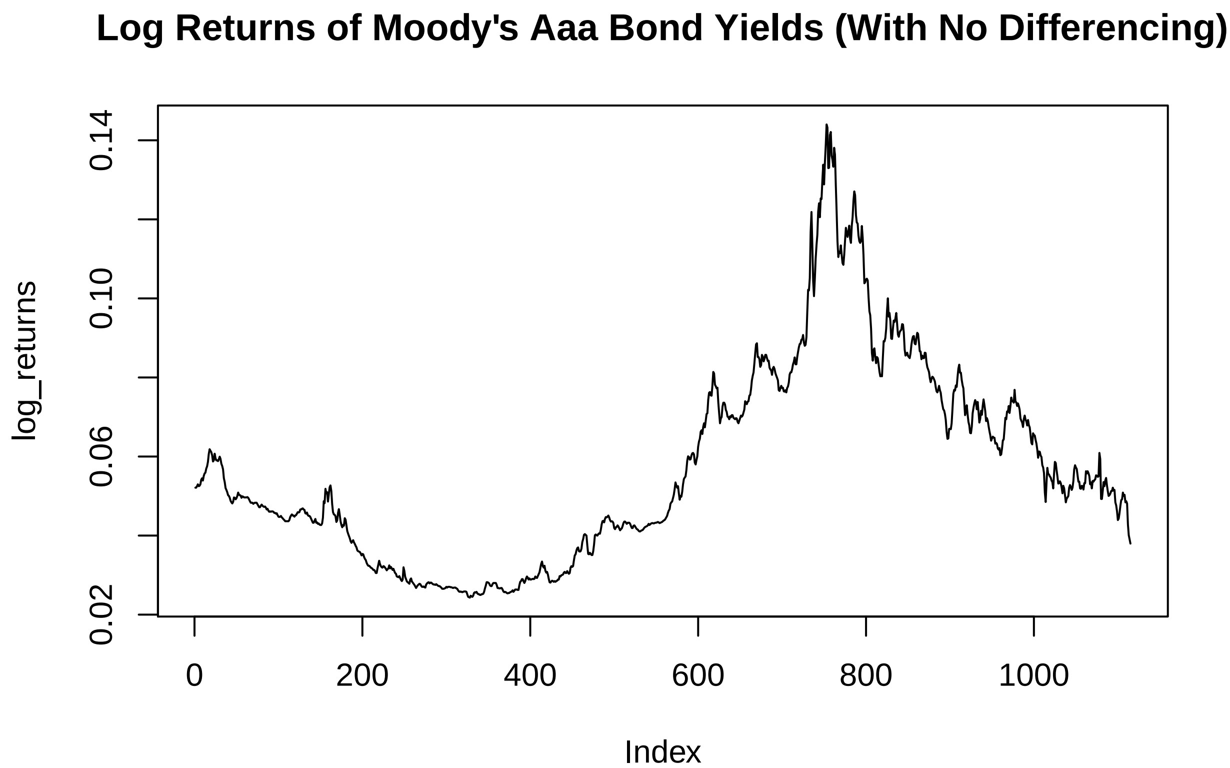
There is an ever so slight improvement in the AIC. Unfortunately the residuals are not white noise.

```r
Box.test(arma_model$residuals, lag = 12, type = "Ljung")
```

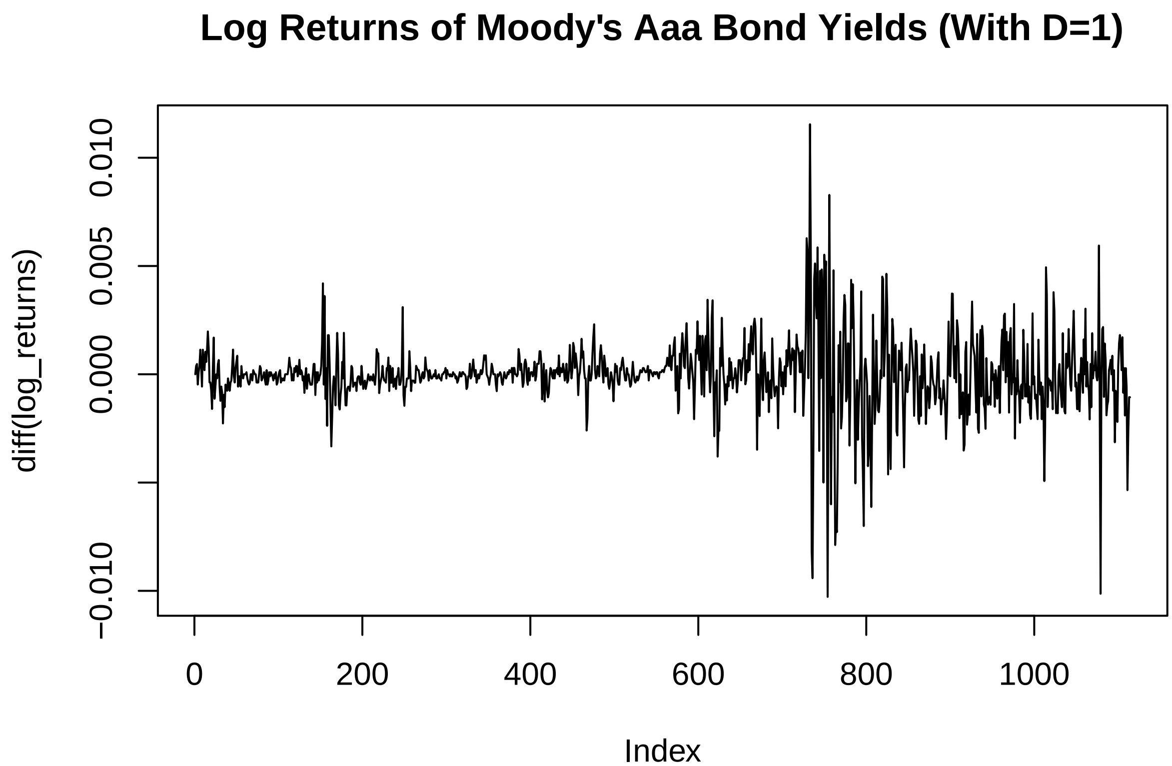
```
##
##  Box-Ljung test
##
## data:  arma_model$residuals
## X-squared = 34.489, df = 12, p-value = 0.0005646
```

Still, I would choose the ARMA(1,3) Restricted Model. None of the pure AR models are satisfactory or intuitive, and its performance is not substantially worse. This may be due to the nature of the data. It does not appear stationary, even when we take a difference = 1.

```r
plot(log_returns, type = "l", main = "Log Returns of Moody's Aaa Bond Yields (With No Differencing)")
```




```
plot(diff(log_returns), type="l", main = "Log Returns of Moody's Aaa Bond Yields (With D=1)")
```


Log Returns of Moody's Aaa Bond Yields (With D=1)
diff(log_returns)
0.010
0.005
0.000
−0.010
0
200
400
600
800
1000
Index