Assignment 5: Nonstationary Univariate ARMA Models

Andrew G. Dunn

January 25, 2016

Andrew G. Dunn, Northwestern University Predictive Analytics Program

Prepared for PREDICT-413: Time Series Analysis and Forecasting.

Formatted using the LaTeX, via pandoc and R Markdown. References managed using pandoc-citeproc.

Setup

```
require(fBasics)    # for calculations
require(fpp)        # for data
require(knitr)      # for table output
require(ggplot2)    # for graphing
require(ggfortify)  # for graphing time series
require(ggthemes)   # for graphing beautifully
require(gridExtra)  # for laying out graphs
```

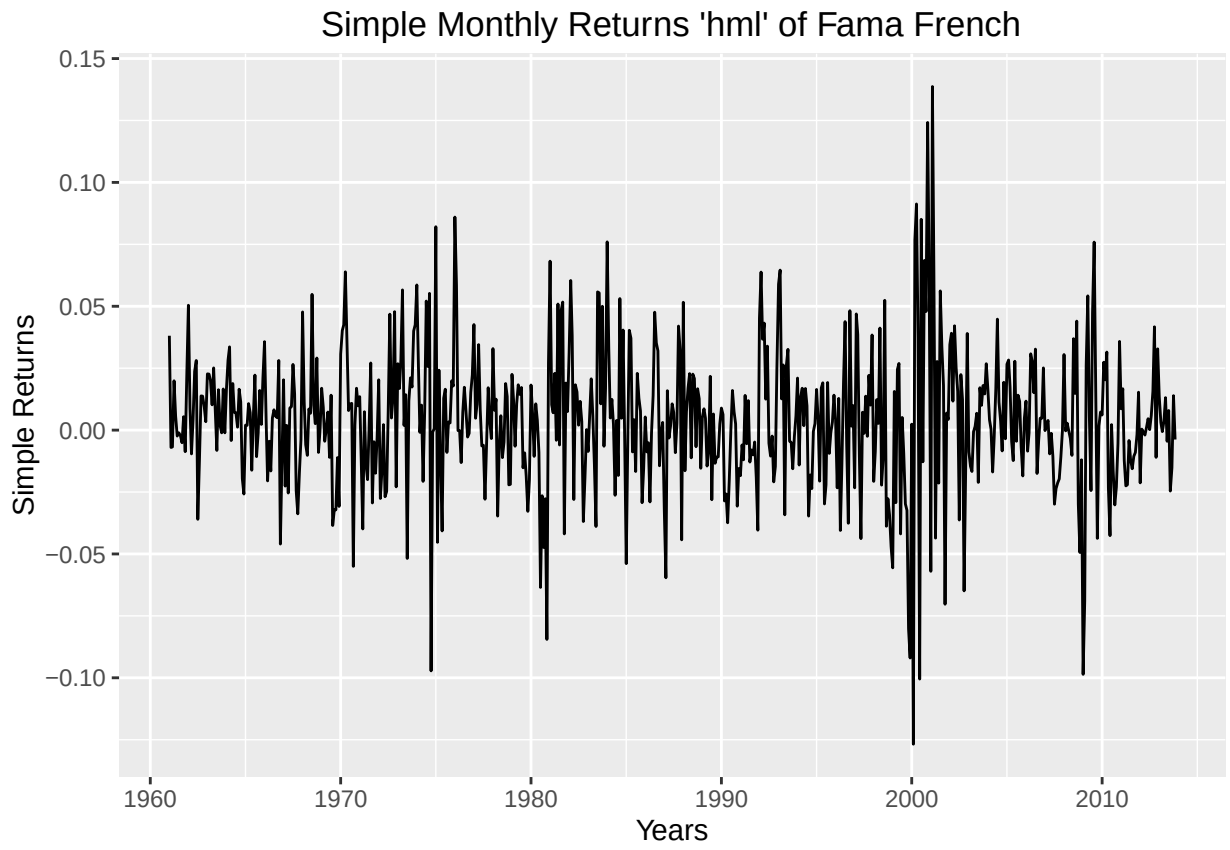
Part 1

Consider the monthly returns of Fama-French factors from January 1961 to November 2013. The data are in the file `m-FamaFrench.txt`. Focus on the simple returns of the factor hml.

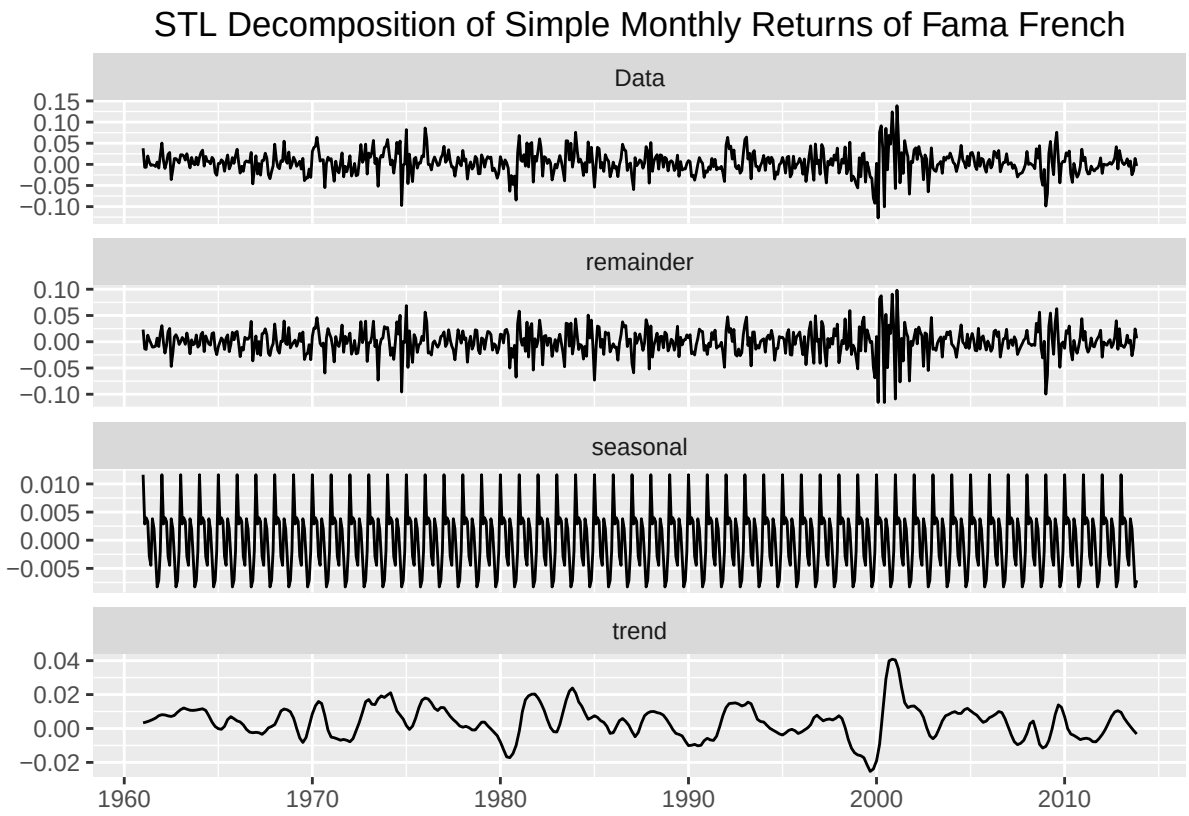
```
d1 = read.table("data/m-FamaFrench.txt", header=T)
head(d1)
```

```
##     dateff     smb     hml   mktrf     rf     umd
## 1 19610131  0.0079  0.0381  0.0620 0.0019 -0.0402
## 2 19610228  0.0399 -0.0070  0.0356 0.0014  0.0102
## 3 19610330  0.0323 -0.0068  0.0288 0.0020  0.0407
## 4 19610428  0.0009  0.0199  0.0029 0.0017  0.0347
## 5 19610531  0.0196  0.0056  0.0240 0.0018 -0.0156
## 6 19610630 -0.0248 -0.0024 -0.0308 0.0020  0.0026
```

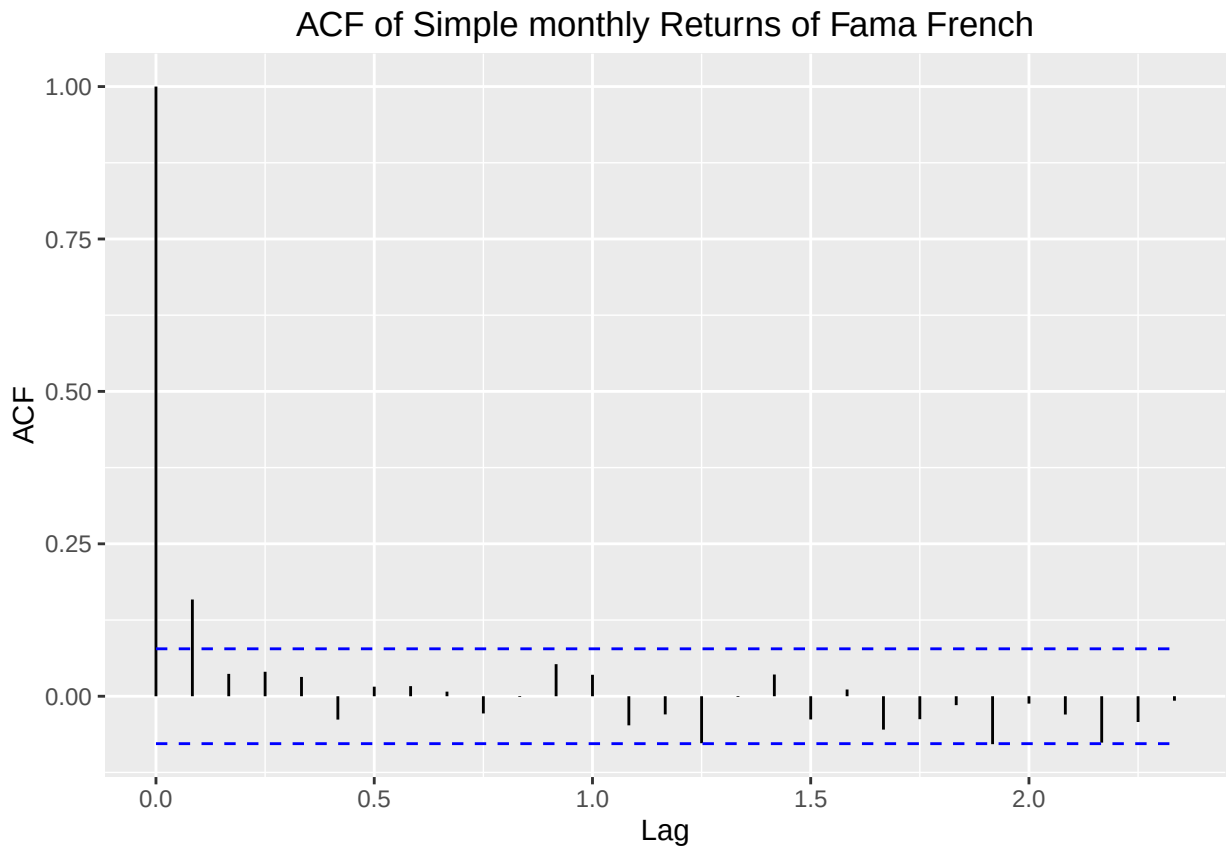
```r
t1 = ts(d1$hml, start = 1961, frequency = 12)
autoplot(t1, main = "Simple Monthly Returns 'hml' of Fama French", ylab = "Simple Returns", xlab = "Yea
```



```
t1_stl = stl(t1, s.window="periodic")
autoplot(t1_stl, main = "STL Decomposition of Simple Monthly Returns of Fama French")
```



```r
t1_acf = acf(t1, plot = FALSE)
autoplot(t1_acf, main = "ACF of Simple monthly Returns of Fama French")
```



Part A

Build a time series model for the mean equation of the hml factor. Write down the fitted model.

```r
m1 = arima(t1, order = c(0, 0, 1))
print(m1)
```

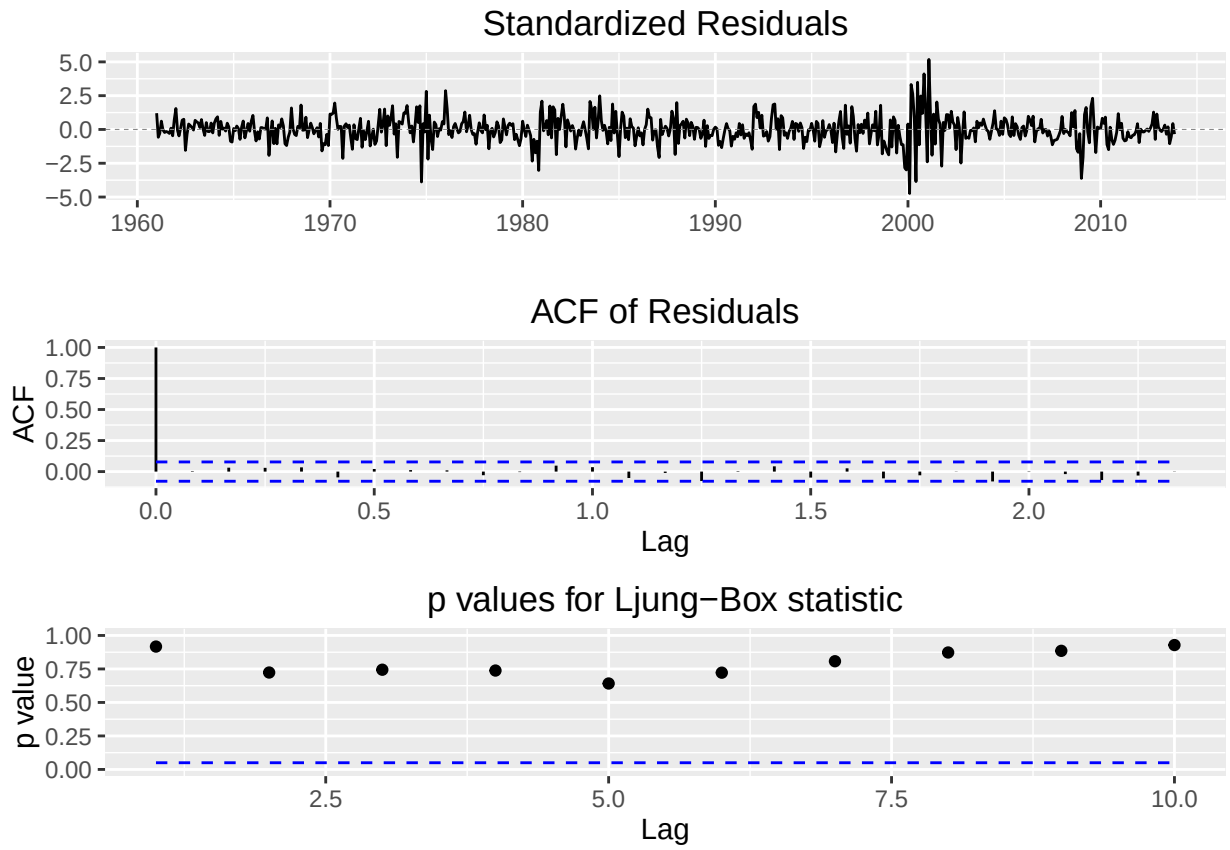
```
##
## Call:
## arima(x = t1, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.1536     0.0040
## s.e.  0.0383     0.0013
##
## sigma^2 estimated as 0.0007836:  log likelihood = 1369.59,  aic = -2733.19
```

$$y_t = 0.004 + 0.1536 a_{t-1}$$

Part B

Is the model adequate? Why?

```
ggtsdiag(m1)
```



The model appears adequate since all Ljung-Box p-values are > 0.05

Part C

Obtain 1-step and 2-step ahead point and 95% intervalforecasts for the hml factor at the forecast origin Novemer 2013 (Last data point)

```
pm1 = predict(m1, 2)
print(pm1)
```

```
## $pred
##              Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov         Dec
## 2013                                                    0.002503115
## 2014 0.003985785
##
## $se
##             Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov        Dec
## 2013                                                   0.02799306
## 2014 0.02832126
```

```
pm1_lcl = pm1$pred - 1.96 * pm1$se
pm1_ucl = pm1$pred + 1.96 * pm1$se
print(pm1_lcl)
```

```
##              Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov         Dec
## 2013                                                   -0.05236329
## 2014 -0.05152387
```

```
print(pm1_ucl)
```

```
##             Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov        Dec
## 2013                                                   0.05736952
## 2014 0.05949545
```

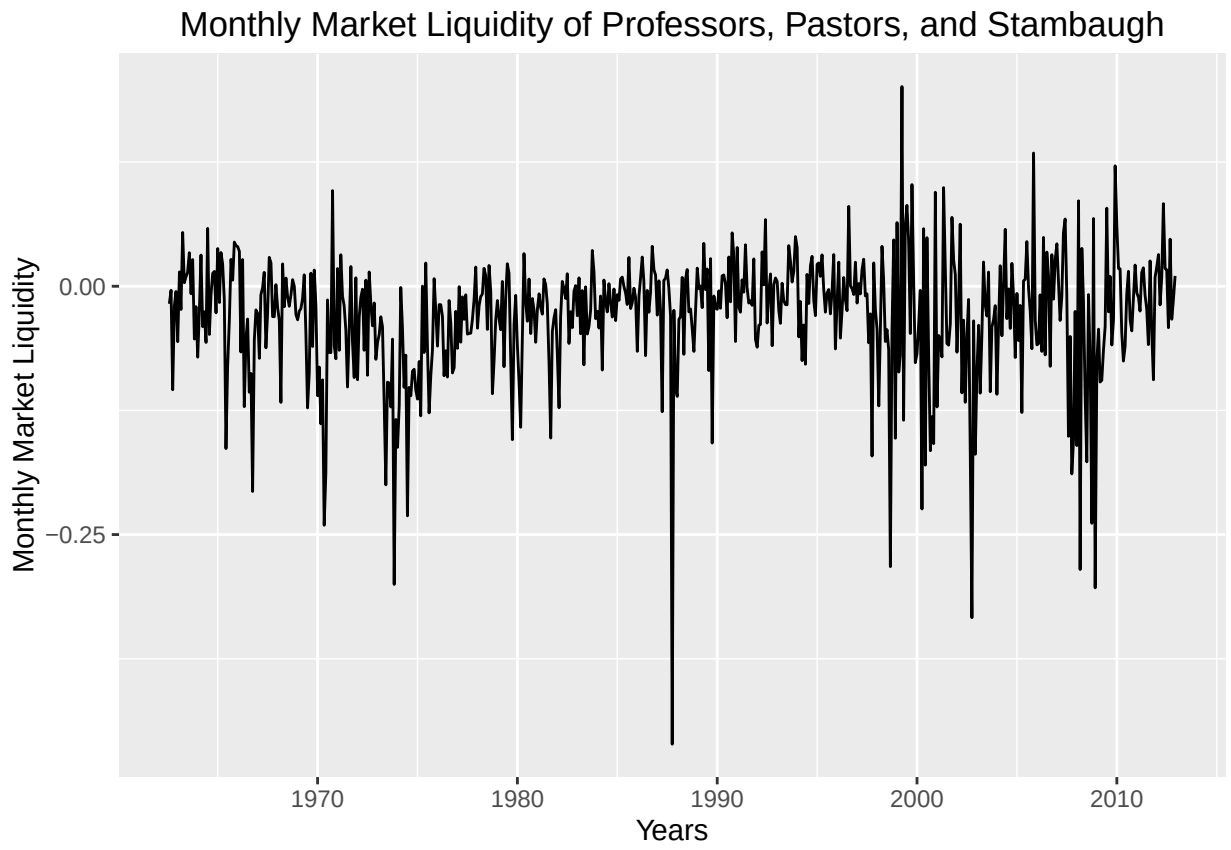
Part 2

Condiser the monthly market liquidity measure of Professors Pastors and Stambaugh. The data are available from Wharton WRDS and are in the file `m-PastorStambaugh.txt`. Focus on the variable PS level and denote the series by x_t.

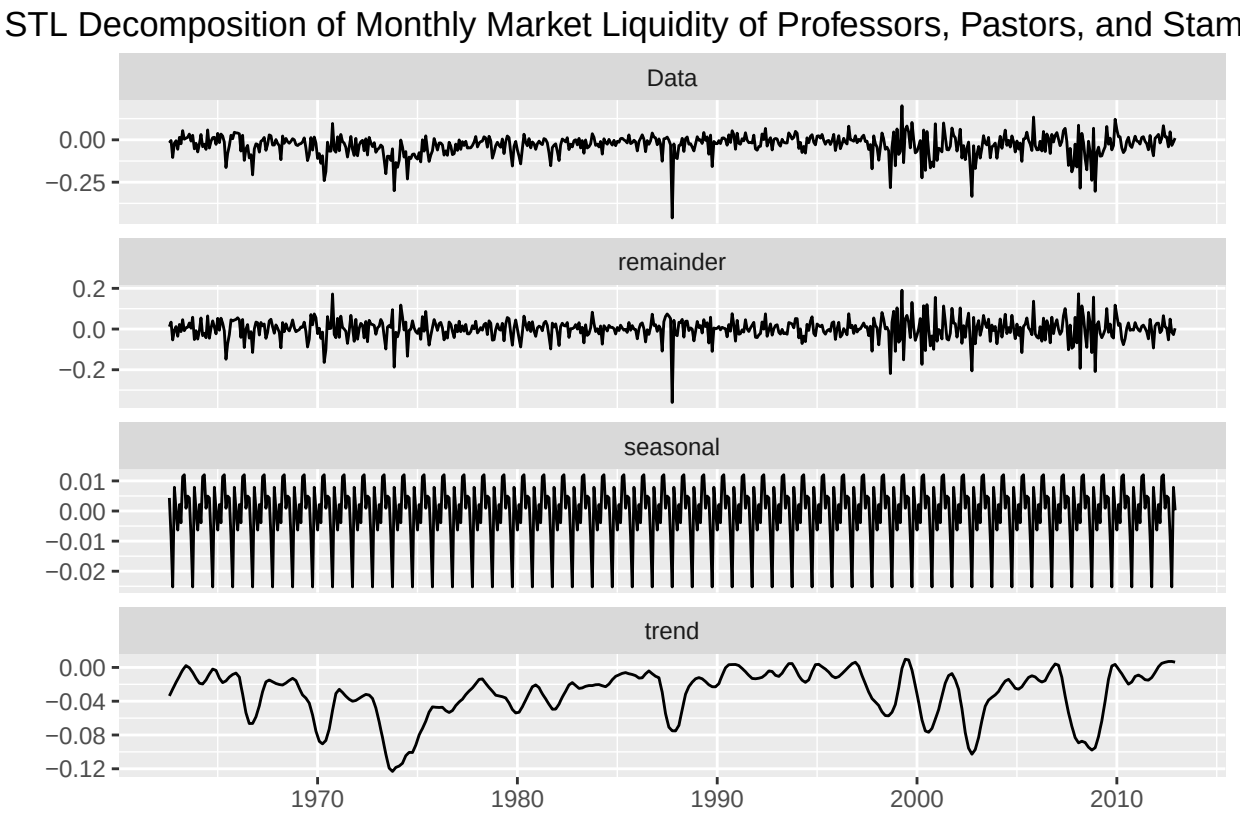
```
d2 = read.table("data/m-PastorStambaugh.txt", header=T)
head(d2)
```

```
##       DATE     PS_LEVEL     PS_INNOV PS_VWF
## 1 19620831 -0.017627253  0.004329499    -99
## 2 19620928 -0.004085483  0.014020110    -99
## 3 19621031 -0.104229891 -0.072041842    -99
## 4 19621130 -0.019620252  0.031439181    -99
## 5 19621231 -0.005288012  0.015044966    -99
## 6 19630131 -0.055790672  0.012292481    -99
```

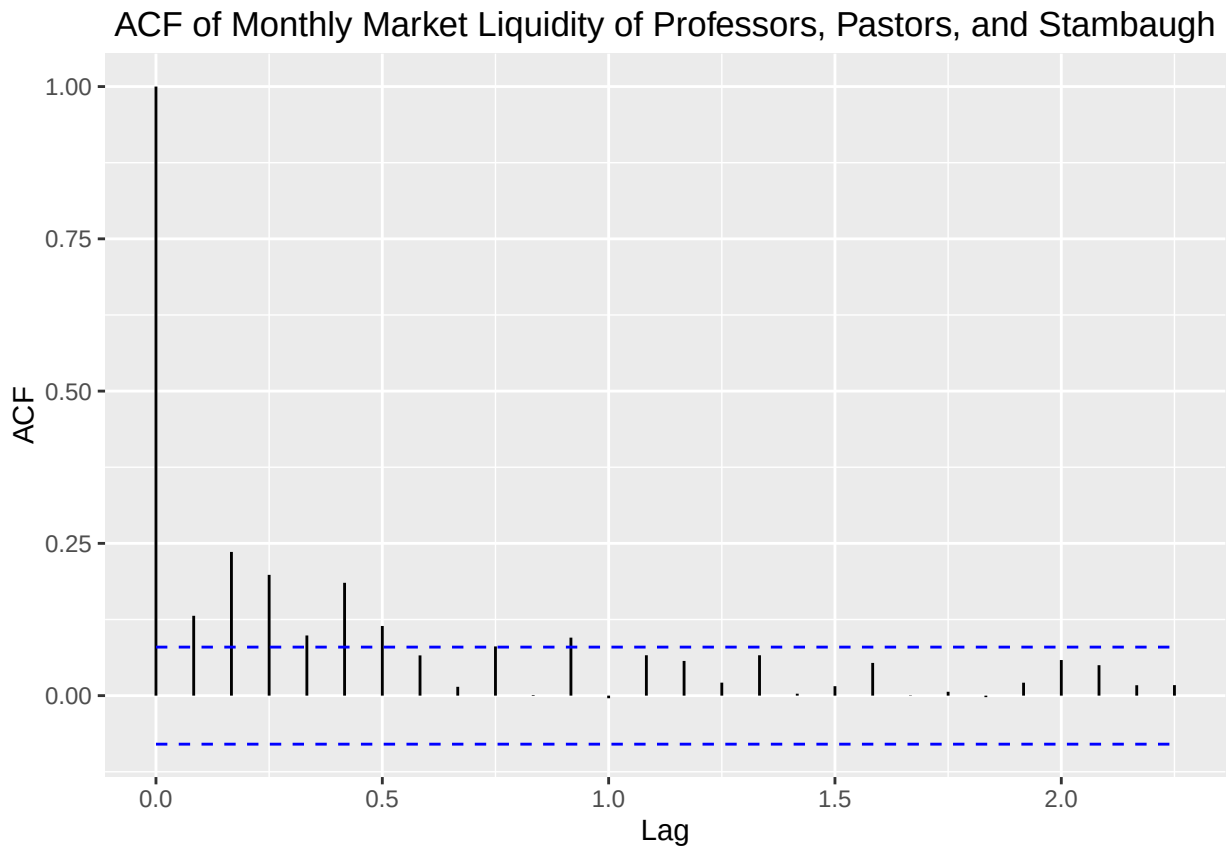
```
t2 = ts(d2$PS_LEVEL, start = c(1962, 8), frequency = 12)
autoplot(t2, main = "Monthly Market Liquidity of Professors, Pastors, and Stambaugh", ylab = "Monthly Ma
```



```
t2_stl = stl(t2, s.window="periodic")
autoplot(t2_stl, main = "STL Decomposition of Monthly Market Liquidity of Professors, Pastors, and Stam
```



```r
t2_acf = acf(t2, plot = FALSE)
autoplot(t2_acf, main = "ACF of Monthly Market Liquidity of Professors, Pastors, and Stambaugh")
```



Part A

Build a time series model for the mean equation. Write down the fitted model.

```
m2 = arima(t2, order = c(5,0,0))
print(m2)
```

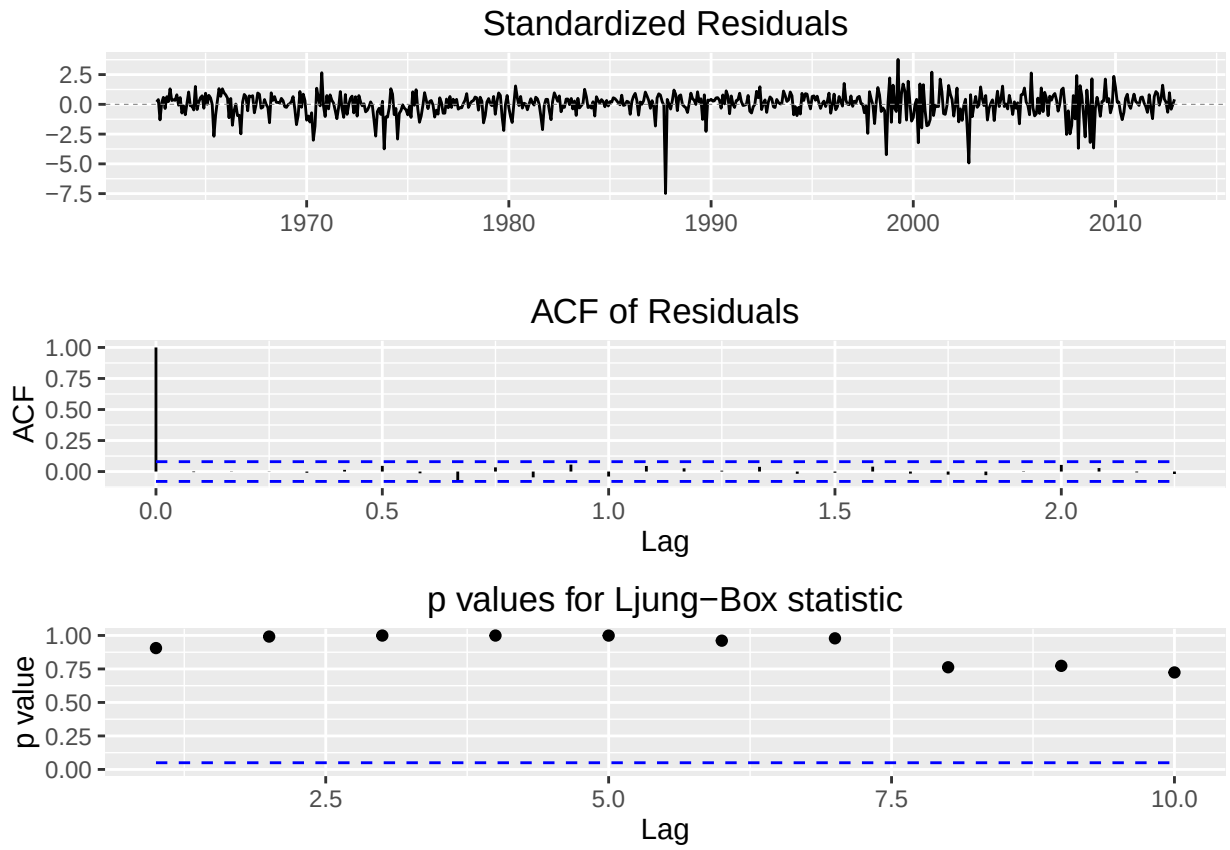
```
##
## Call:
## arima(x = t2, order = c(5, 0, 0))
##
## Coefficients:
##          ar1     ar2     ar3     ar4     ar5  intercept
##       0.0624  0.1858  0.1313  0.0105  0.1096    -0.0306
## s.e.  0.0404  0.0404  0.0408  0.0405  0.0404     0.0048
##
## sigma^2 estimated as 0.003555:  log likelihood = 847.3,  aic = -1680.6
```

$$y_t = -0.0306 + 0.0624y_{t-1} + 0.1858y_{t-2} + 0.1313y_{t-3} + 0.0105y_{t-4} + 0.1096y_{t-5} + e_t$$

Part B

Is the model adequate? Why?

```
ggtsdiag(m2)
```



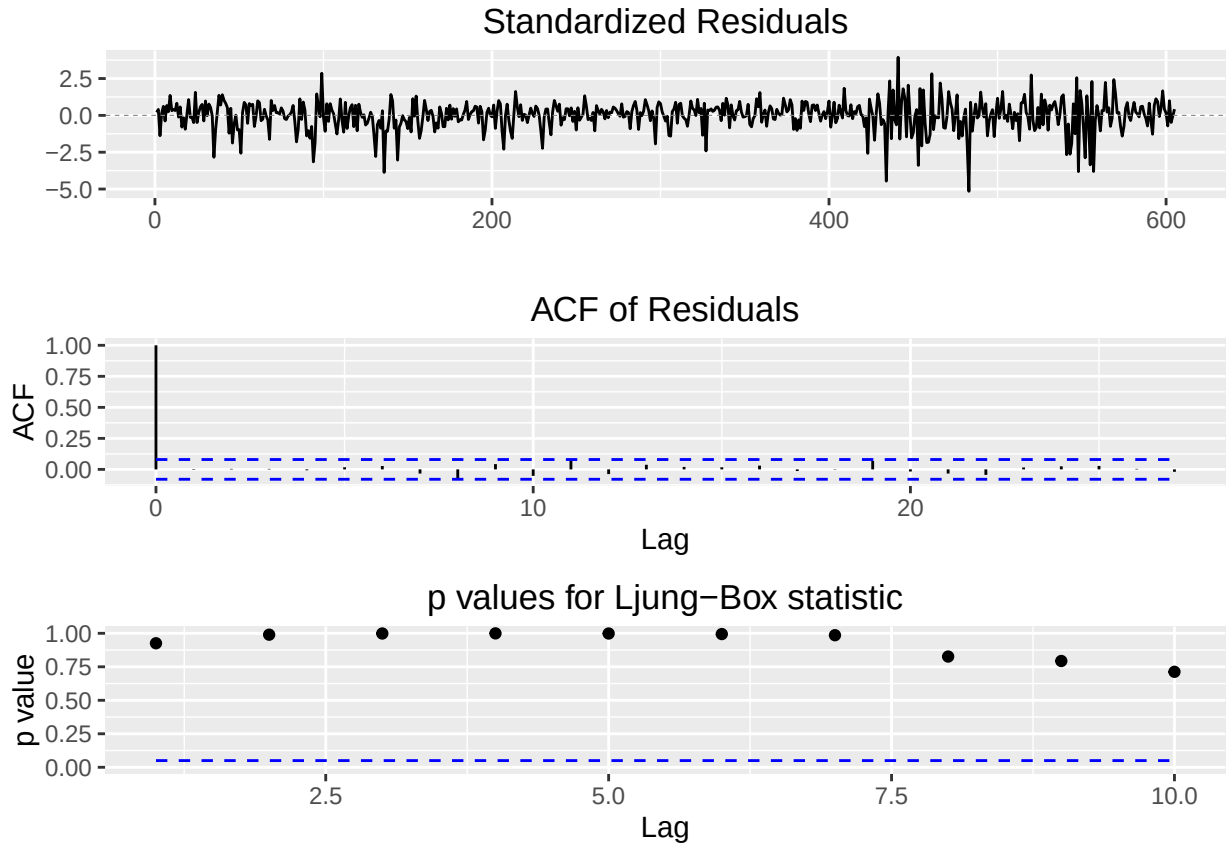
Part C

Identify the largest outlier in the series. Refine the fitted model by using an indicator for the outlier. Write down the refined model.

```
m2oi = which.min(m2$residuals)
d2$outlier = 0
d2$outlier[m2oi] = 1
m3 = arima(d2$PS_LEVEL, order = c(5,0,0), xreg = d2$outlier)
```

$$y_t = -0.0299 + 0.0657y_{t-1} + 0.1884y_{t-2} + 0.1309y_{t-3} + 0.0162y_{t-4} + 0.1261y_{t-5} - 0.4255_{\text{outlier}} + e_t$$

```
ggtsdiag(m3)
```



Part D

Further refine the model by fixing the least significant parameter to zero. Write down the revised model.

```
m3.se = sqrt(diag(vcov(m3)))
m3.tratio = abs(m3$coef/m3.se)
print(m3.tratio)
```

```
##        ar1        ar2        ar3        ar4        ar5  intercept
##   1.631196   4.665825   3.209140   0.401377   3.126644   6.134486
## d2$outlier
##   7.733949
```

From the above, the AR parameter is not significant (t $<$ 1) and the smallest of all the parameters. The outlier parameter is the most important.

From this we see we need to create a mask of `0, NA, NA, 0, NA`.

```
m4_mask = c(NA, NA, NA, 0, NA, NA, NA)
```

```
m4 = arima(d2$PS_LEVEL, order = c(5,0,0), xreg = d2$outlier, fixed = m4_mask)
```

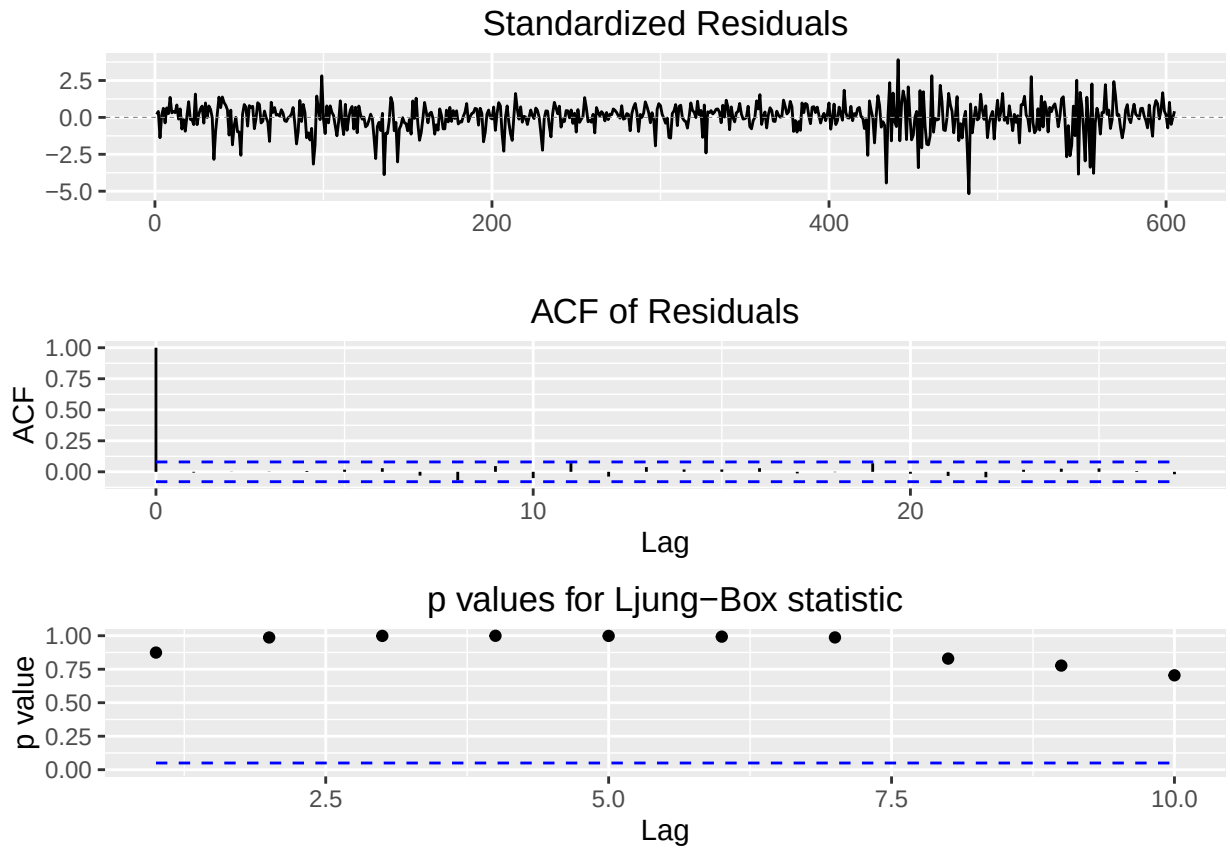
```
## Warning in arima(d2$PS_LEVEL, order = c(5, 0, 0), xreg = d2$outlier, fixed
## = m4_mask): some AR parameters were fixed: setting transform.pars = FALSE
```

```
print(m4)
```

```
##
## Call:
## arima(x = d2$PS_LEVEL, order = c(5, 0, 0), xreg = d2$outlier, fixed = m4_mask)
##
## Coefficients:
##          ar1     ar2     ar3  ar4     ar5  intercept  d2$outlier
##       0.0683  0.1917  0.1318    0  0.1272    -0.0299     -0.4248
## s.e.  0.0398  0.0395  0.0407    0  0.0402     0.0048      0.0549
##
## sigma^2 estimated as 0.003236:  log likelihood = 875.71,  aic = -1737.43
```

$$y_t = -0.0299 + 0.0683y_{t-1} + 0.1917y_{t-2} + 0.1318y_{t-3} + 0.1272y_{t-5} - 0.4248_{\text{outlier}} + e_t$$

```
ggtsdiag(m4)
```


Standardized Residuals
2.5
0.0
−2.5
−5.0
0
200
400
600
ACF of Residuals
ACF
1.00
0.75
0.50
0.25
0.00
0
10
20
Lag
p values for Ljung−Box statistic
p value
1.00
0.75
0.50
0.25
0.00
2.5
5.0
7.5
10.0
Lag

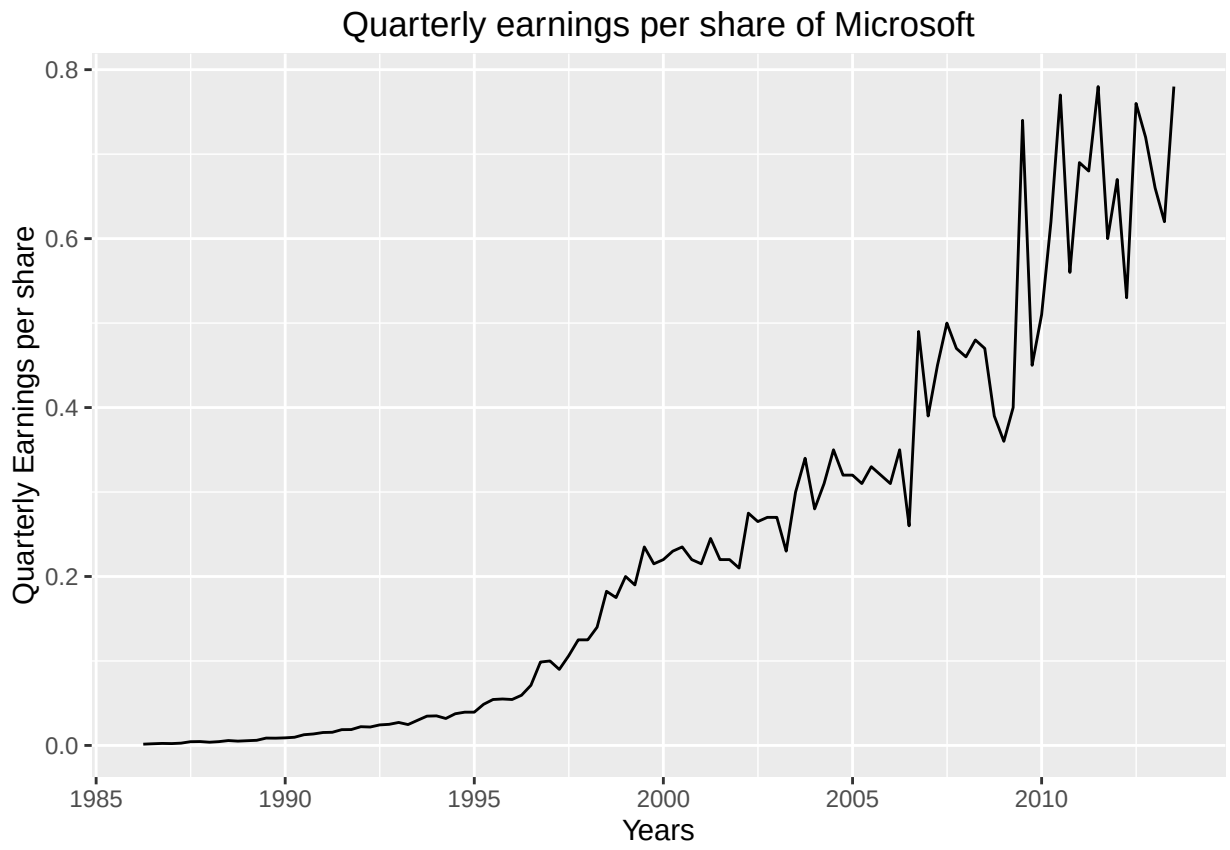
Part 3

Consider the quarterly earnings per share of Microsoft from the second quarter of 1986 to the third quarter of 2013. The original data were from IBES, but contain four missing values in 2002 and 2003. The data are in the file `q-earn-msft.txt`. Other sources are used to fill in the missing values. Focus on the log earnings per share.

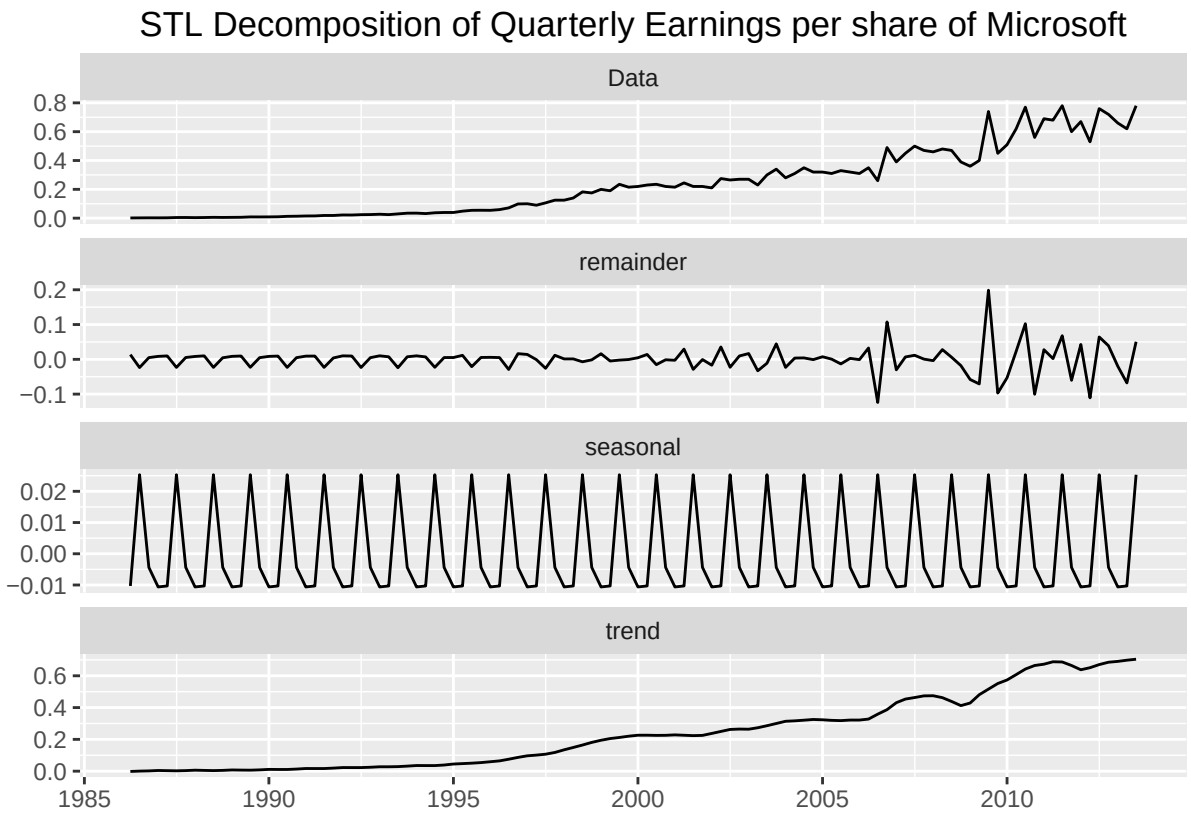
```
d3 = read.table("data/q-earn-msft.txt", header=T)
head(d3)
```

```
##     yr qr  value
## 1 1986  2 0.0015
## 2 1986  3 0.0020
## 3 1986  4 0.0024
## 4 1987  1 0.0022
## 5 1987  2 0.0027
## 6 1987  3 0.0044
```

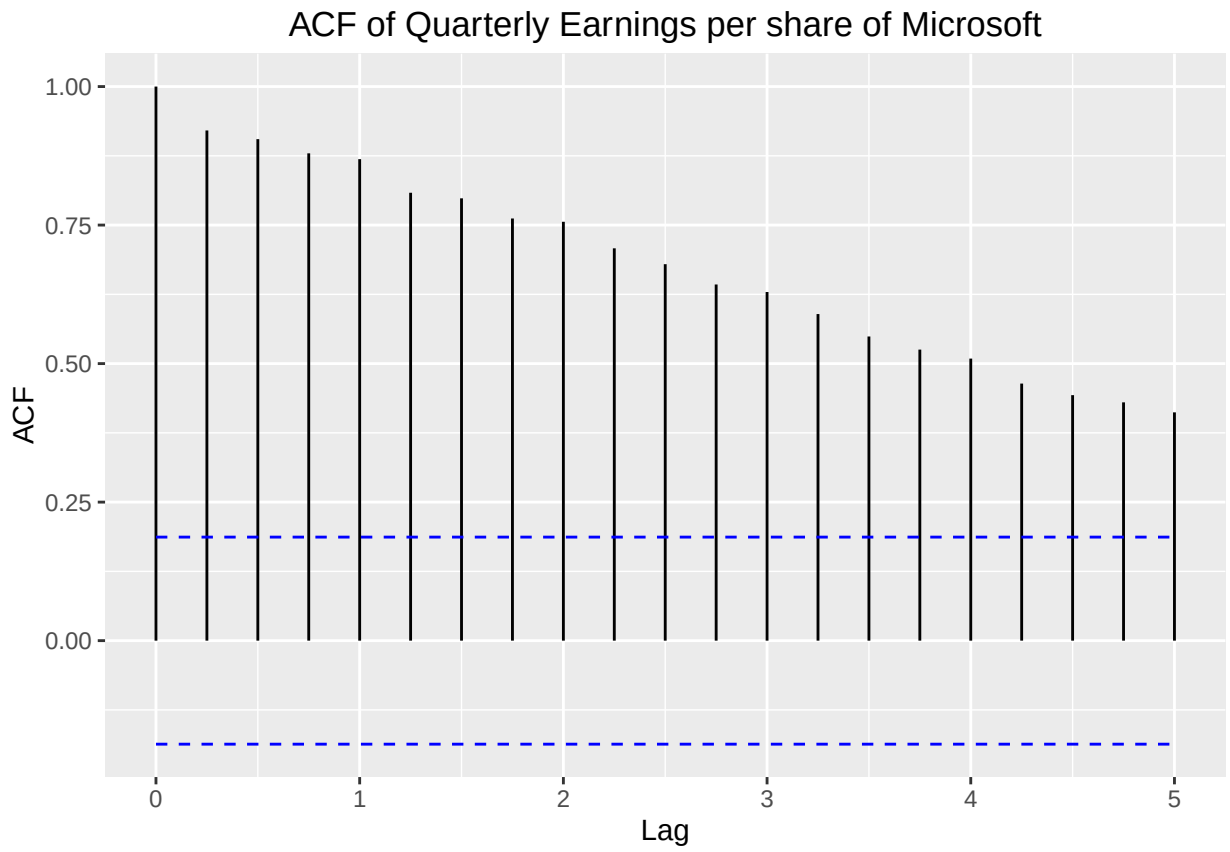
```
t3 = ts(d3$value, start = c(1986, 2), frequency = 4)
autoplot(t3, main = "Quarterly earnings per share of Microsoft", ylab = "Quarterly Earnings per share",
```



```
t3_stl = stl(t3, s.window="periodic")
autoplot(t3_stl, main = "STL Decomposition of Quarterly Earnings per share of Microsoft")
```

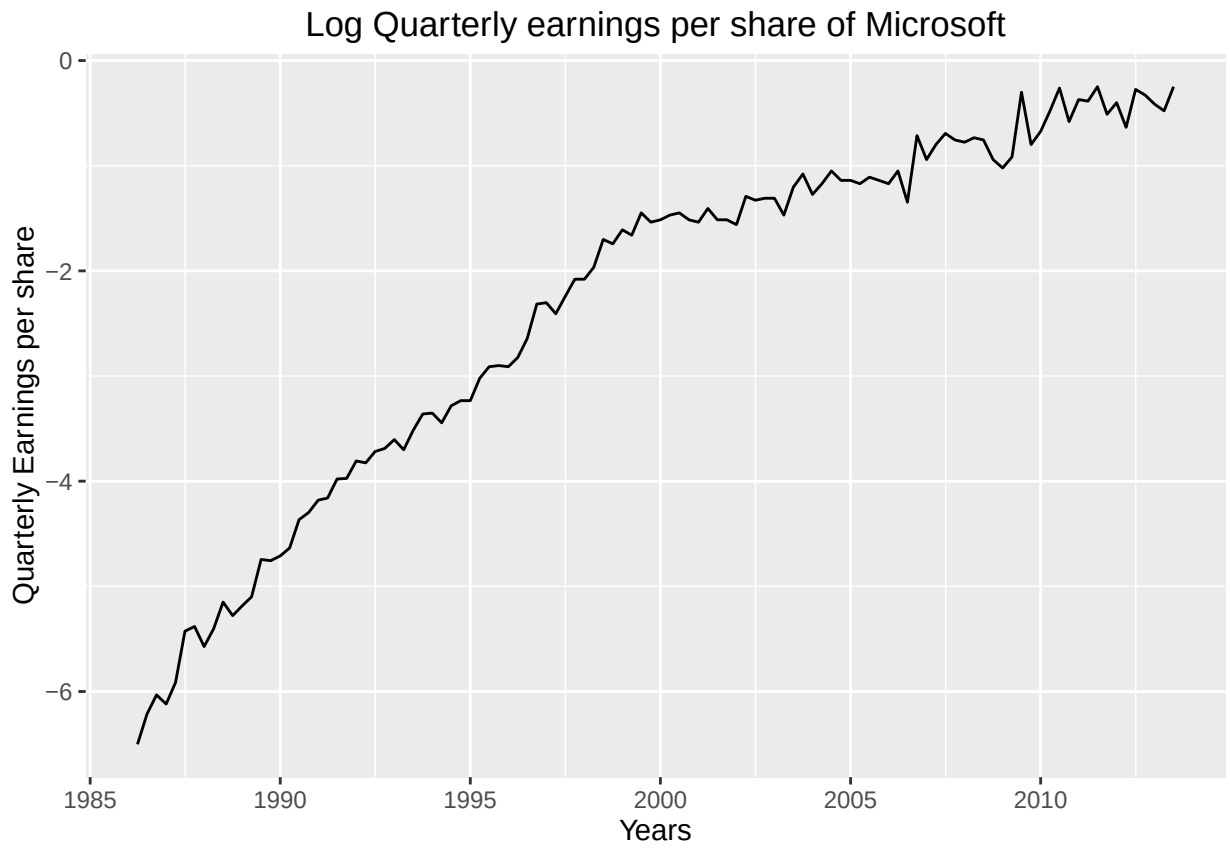


```r
t3_acf = acf(t3, plot = FALSE)
autoplot(t3_acf, main = "ACF of Quarterly Earnings per share of Microsoft")
```

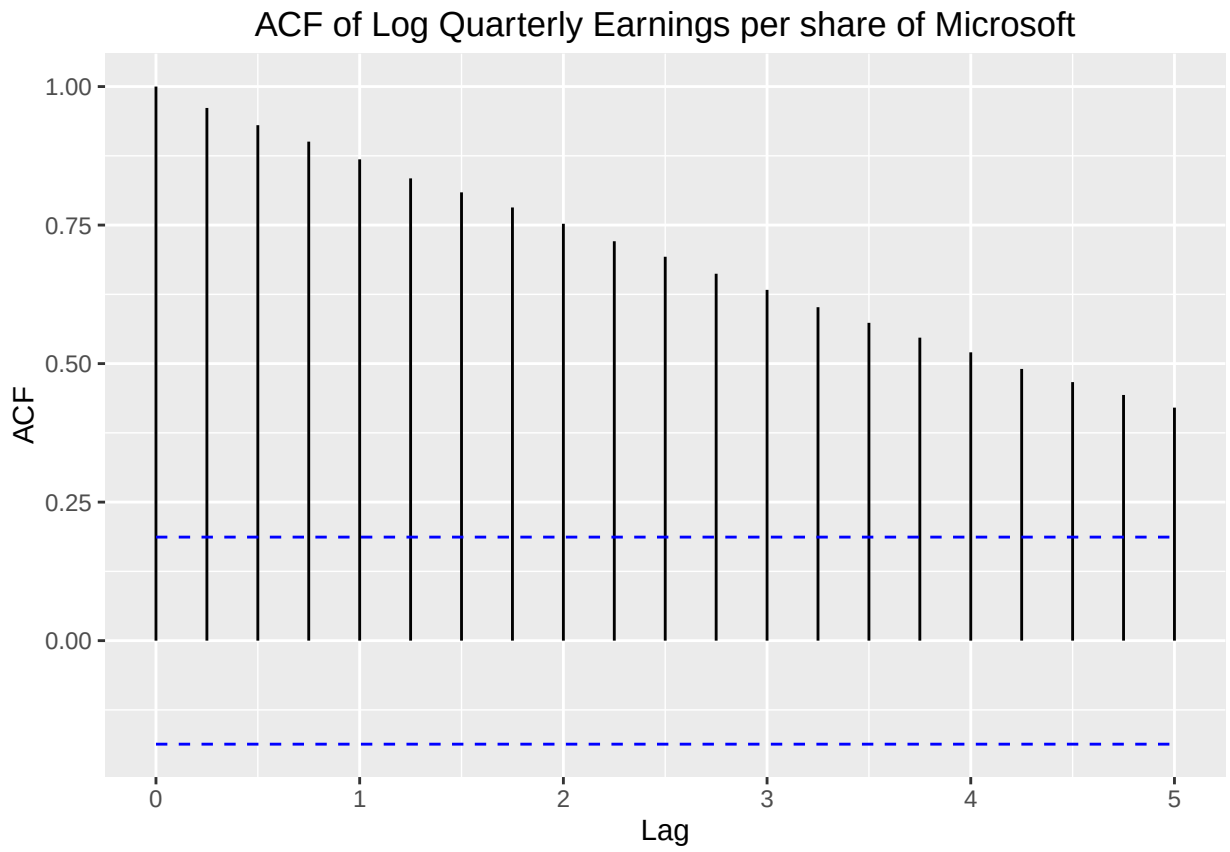


We'll create another time series of the log transform:

```
t4 = ts(log(d3$value), start = c(1986, 2), frequency = 4)
autoplot(t4, main = "Log Quarterly earnings per share of Microsoft", ylab = "Quarterly Earnings per sha
```



```r
t4_acf = acf(t4, plot = FALSE)
autoplot(t4_acf, main = "ACF of Log Quarterly Earnings per share of Microsoft")
```



Part A

Build a time series model for the log earnings series. Perform model checking and write down the fitted model.

```
m5 = arima(t4, order = c(0,1,1), seasonal = list(order=c(0,1,1), period=4))
print(m5)
```

```
##
## Call:
## arima(x = t4, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 4))
##
## Coefficients:
##           ma1     sma1
##       -0.4826  -0.7149
## s.e.   0.0809   0.0909
##
## sigma^2 estimated as 0.02122:  log likelihood = 51.68,  aic = -97.37
```

$$(1-B)(1-B^4)y_t = (1-0.4826)(1-0.7149^4)a_t$$

Part B

Fit the following model to the log earnings series: `arima(xt, order = c(0,1,1), seasonal = list(order = c(0,0,1), period = 4))`, Where xt denotes the log earnings series. Write down the fitted model.

```
m6 = arima(t4, order = c(0,1,1), seasonal = list(order=c(0,0,1), period=4))
print(m6)
```

```
##
## Call:
## arima(x = t4, order = c(0, 1, 1), seasonal = list(order = c(0, 0, 1), period = 4))
##
## Coefficients:
##           ma1    sma1
##       -0.2723  0.3894
## s.e.   0.0803  0.0762
##
## sigma^2 estimated as 0.02565:  log likelihood = 44.62,  aic = -83.24
```

$$(1-B)(1-B^4)y_t = (1-0.2723)(1-0.3894^4)a_t$$

Part C

Compare the two time series models. Which model is preferred in terms of fitting? Why?

Model m5 has a lower AIC.

Part D

Use the backtest procedure to compare the two models via 1-step ahead forecasts. You may use $t = 81$ as the starting forecast origin. Which model is preferred? Why?

```
source('backtest.R')
backtest(m5, t4, 81, h=1, inc.mean=F)
```

```
## [1] "RMSE of out-of-sample forecasts"
## [1] 0.2131425
## [1] "Mean absolute error of out-of-sample forecasts"
## [1] 0.1553725
```

```
backtest(m6, t4, 81, h=1, inc.mean=F)
```

```
## [1] "RMSE of out-of-sample forecasts"
## [1] 0.22943
## [1] "Mean absolute error of out-of-sample forecasts"
## [1] 0.1640435
```

Model m5 is preferred because it has a lower RMSE. This model doesn't have any seasonal differencing.

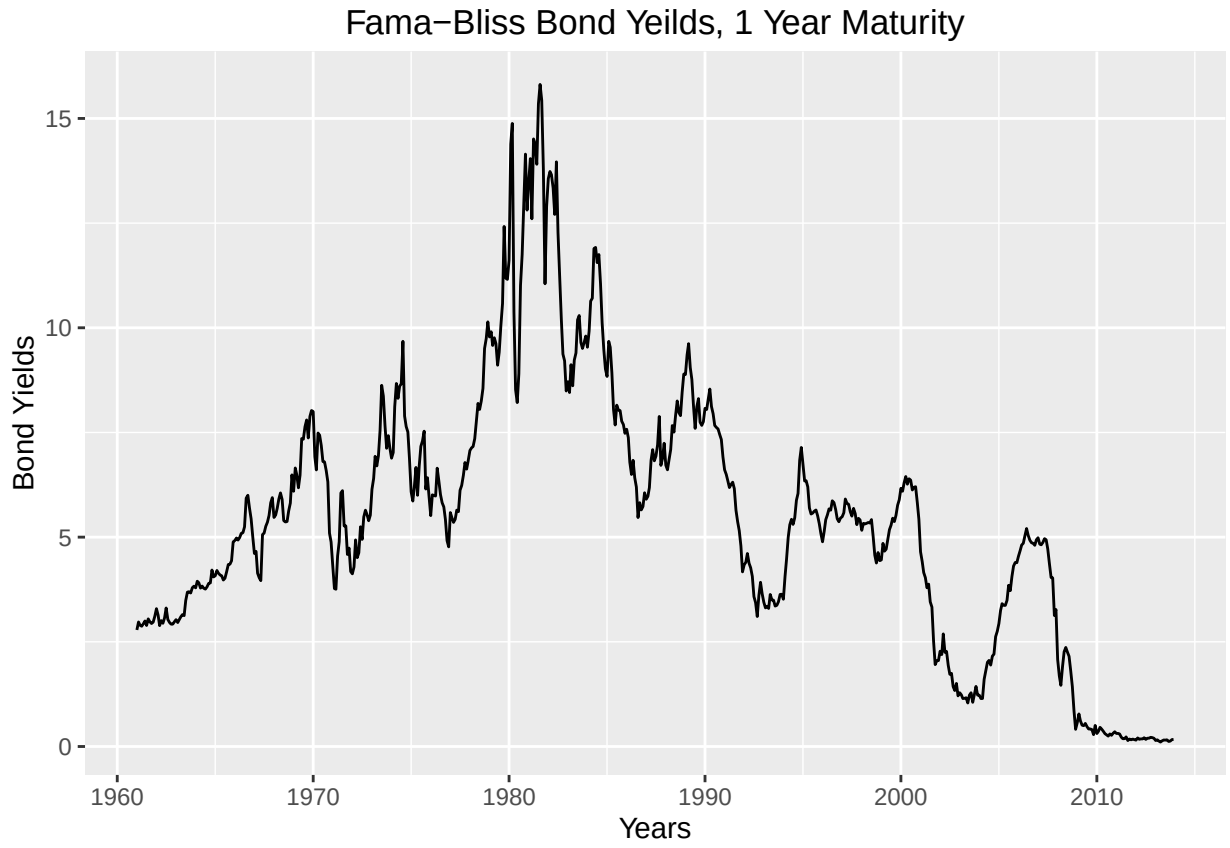
Part 4

Consider the monthly Fama-Bliss bond yields with maturities 1 and 3 years. The data are available from CRSP and in the file `m-FamaBlissdbndyields.txt`. Denote the yields by y_{1t} and y_{3t}, respectively. The goal here is to explore the dependence of the 3-year yield on the 1-year yield.

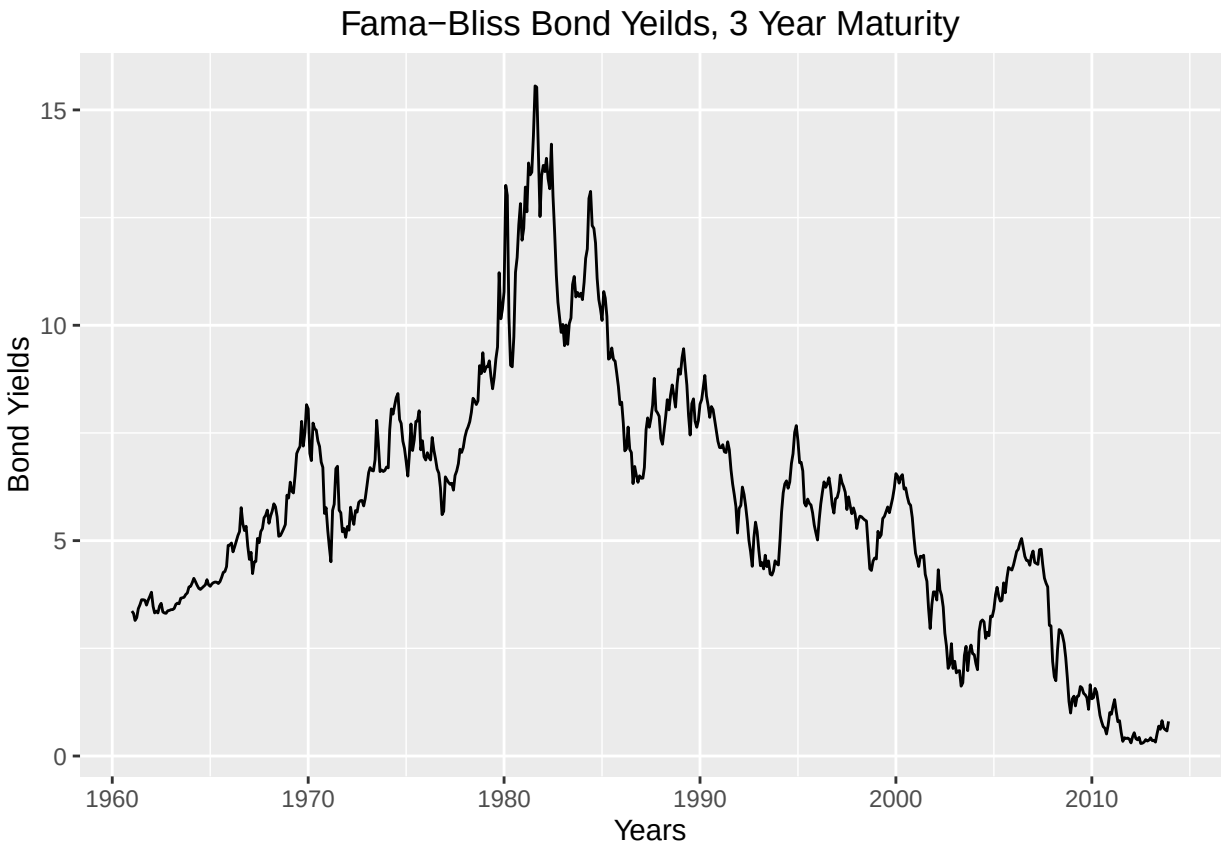
```
d4 = read.table("data/m-FamaBlissdbndyields.txt", header=T)
head(d4)
```

```
##      qdate yield1 yield3
## 1 19610131  2.783  3.365
## 2 19610228  2.974  3.311
## 3 19610330  2.896  3.147
## 4 19610428  2.872  3.211
## 5 19610531  2.929  3.420
## 6 19610630  2.998  3.513
```

```
t5 = ts(d4$yield1, start = c(1961), frequency = 12)
t6 = ts(d4$yield3, start = c(1961), frequency = 12)
autoplot(t5, main = "Fama-Bliss Bond Yeilds, 1 Year Maturity", ylab = "Bond Yields", xlab = "Years")
```




```
autoplot(t6, main = "Fama-Bliss Bond Yeilds, 3 Year Maturity", ylab = "Bond Yields", xlab = "Years")
```


Fama−Bliss Bond Yeilds, 3 Year Maturity
Bond Yields
15
10
5
0
1960
1970
1980
1990
2000
2010
Years

Part A

Fit the linear regression model $y_{3t} = \alpha + \beta y_{1t} + e_t$. Write down the fitted model. What is the R^2? Is the model adequate? Why?

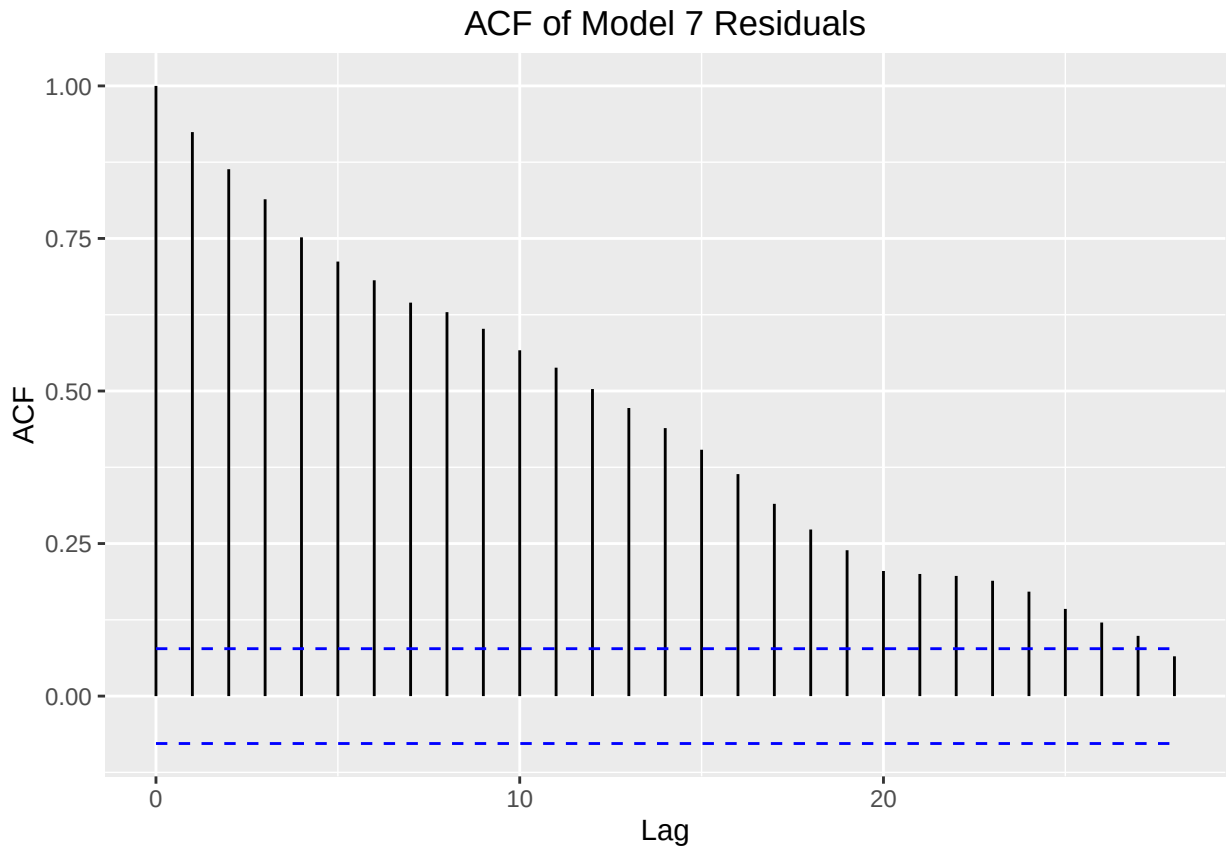
```
m7 = lm(t6~t5)
summary(m7)
```

```
##
## Call:
## lm(formula = t6 ~ t5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.69599 -0.42038 -0.03045  0.37993  1.41445
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.712645   0.041909    17.0   <2e-16 ***
## t5          0.940816   0.006676   140.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.529 on 634 degrees of freedom
## Multiple R-squared:  0.9691, Adjusted R-squared:  0.969
## F-statistic: 1.986e+04 on 1 and 634 DF,  p-value: < 2.2e-16
```

$$t_{3t} = 0.7126 + 0.9408 y_{1t} + e_t$$

The R^2 of the model is 0.9691.

```
m7_acf = acf(m7$residuals, plot = FALSE)
autoplot(m7_acf, main = "ACF of Model 7 Residuals")
```



There appears to be a number of significant lags, the model does not appear to be adequate (exhibits serial autocorrelation)

Part B

Let $d_{1t} = (1 - B)y_{1t}$ and $d_{2t} = (1 - B)y_{3t}$, where B is the back-shift operator. Here it denotes the change in monthly bond yields. Consider the linear regression $d_{3t} = \beta d_{1t} + e_t$. Write down the fitted model. What is the R^2? Justify that it is appropriate to taking the first difference of the bond yields.

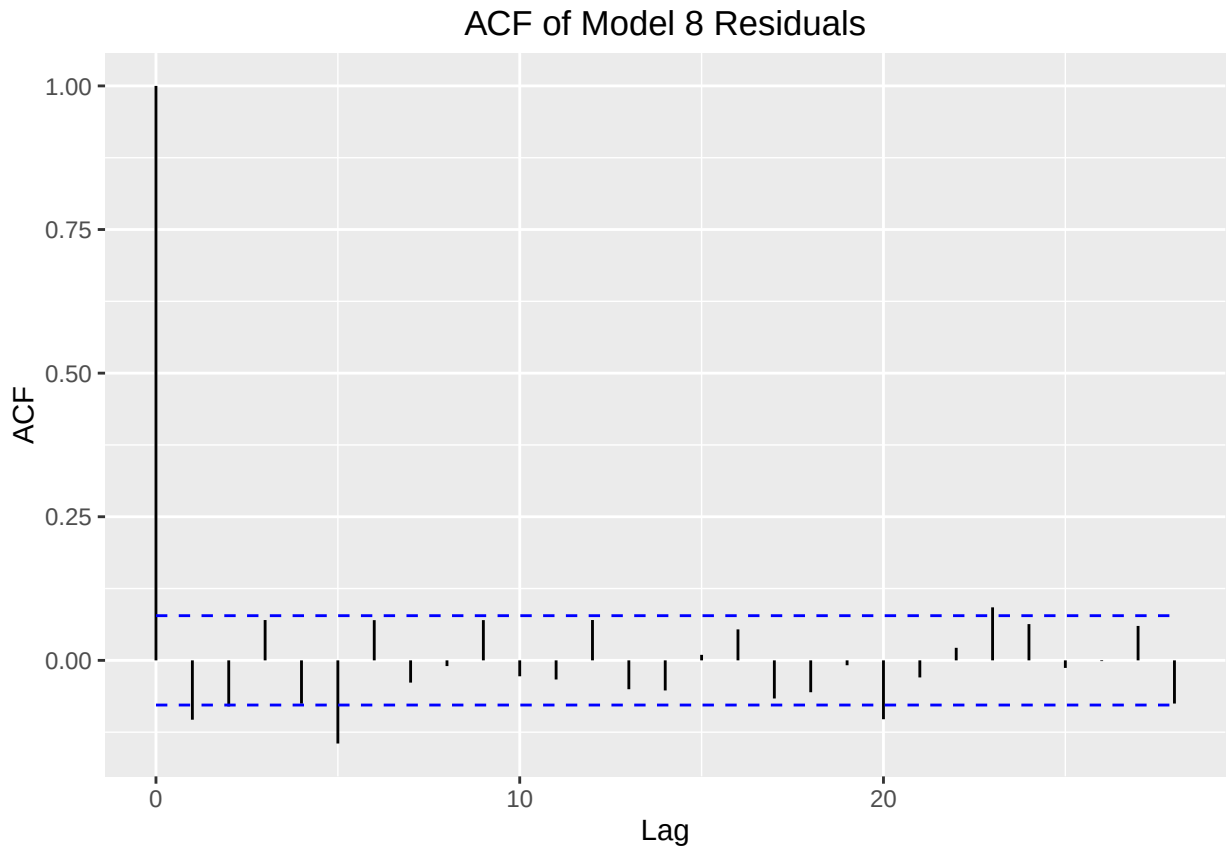
```
d1t = diff(t5)
d3t = diff(t6)

m8 = lm(d3t~1 + d1t)
summary(m8)
```

```
##
## Call:
## lm(formula = d3t ~ 1 + d1t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65567 -0.11032 -0.00954  0.09722  0.81726
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.001003   0.007160   -0.14    0.889
## d1t          0.735957   0.014796   49.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1804 on 633 degrees of freedom
## Multiple R-squared:  0.7963, Adjusted R-squared:  0.796
## F-statistic:  2474 on 1 and 633 DF,  p-value: < 2.2e-16
```

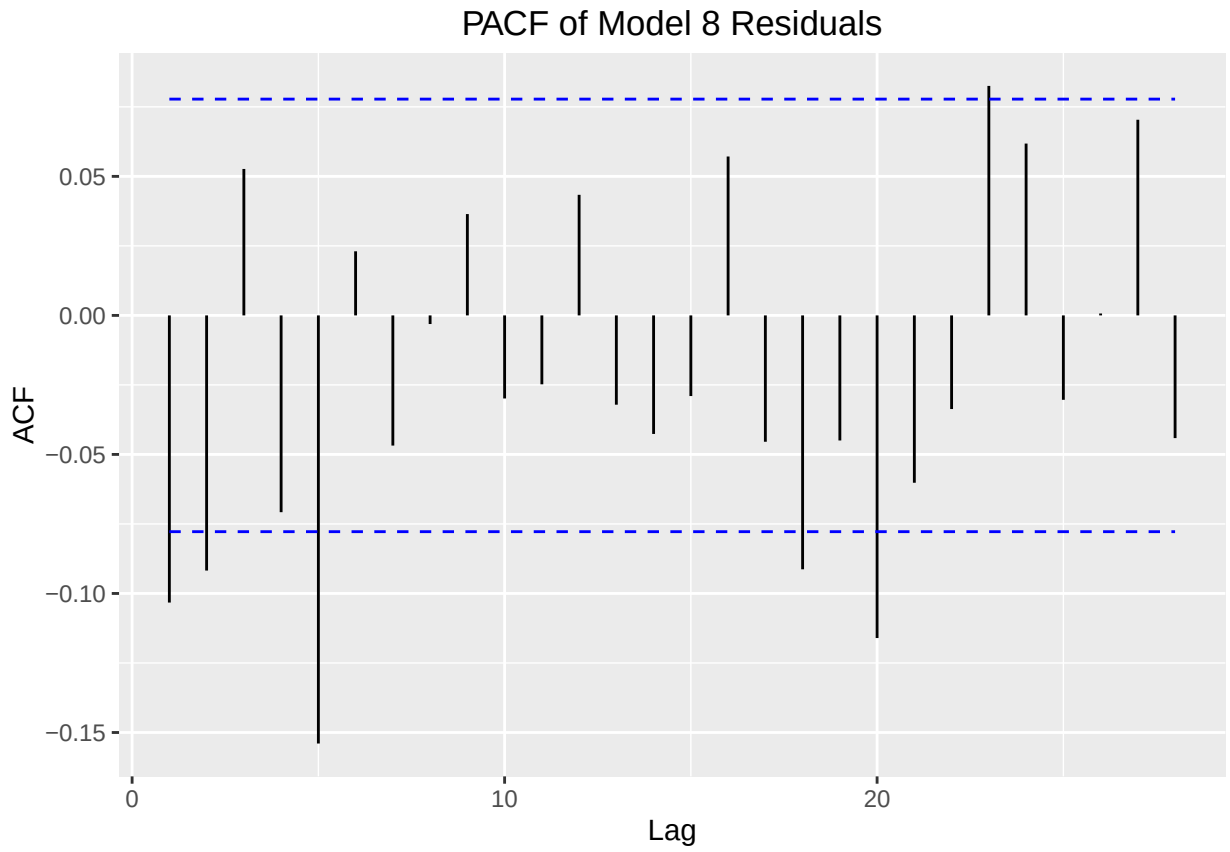
$$t_{d3t} = -0.001 + 0.7359 y_{d1t} + e_t$$

The R^2 of the model is 0.7963.

```
m8_acf = acf(m8$residuals, plot = FALSE)
autoplot(m8_acf, main = "ACF of Model 8 Residuals")
```




```
m8_pacf = pacf(m8$residuals, plot = FALSE)
autoplot(m8_pacf, main = "PACF of Model 8 Residuals")
```



Although the R^2 value is less in model 8, taking the difference of the time series is justified as it reduces the serial autocorrelation and produces a more accurate model.

Part C

Is the model accurate? If not, refine the model and write down the refined model.

The model is not accurate due to significant lags at 1 and 5 (seen in the PACF).

Part D

Based on the refined model, describe the linear dependence between the bond yields.

The 3-year bond yield at time t can be found by taking previous period 3-year bond yield and adding 73.6% of the change in the 1-year bond yield.

Part 5

Consider again the bond yields of Problem 4. Suppose that one is concerned with taking the first difference. To mitigate the concern, one can perform the analyses below;

Part A

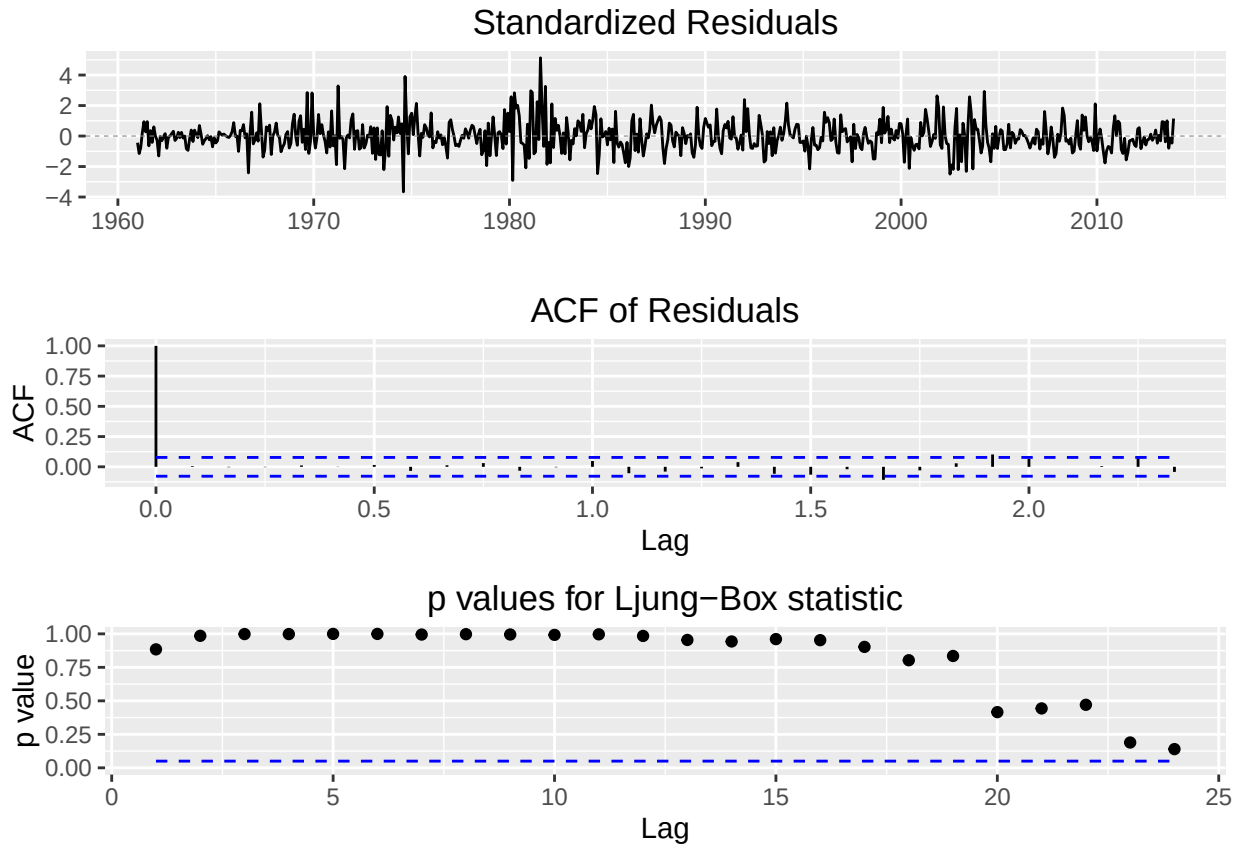
Fit an AR(6) model to y_{3t} using y_{1t} as an explanatory variable. Write down the fitted model. You should include the intercept term in the model as the original data are used.

```
m9 = arima(t6, order=c(6,0,0), xreg = t5)
print(m9)
```

```
##
## Call:
## arima(x = t6, order = c(6, 0, 0), xreg = t5)
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ar5     ar6  intercept      t5
##       0.8744  0.0329  0.1175  -0.1243  -0.0635  0.1457     1.6424  0.7469
## s.e.  0.0395  0.0525  0.0522   0.0527   0.0529  0.0394     0.3652  0.0144
##
## sigma^2 estimated as 0.03063:  log likelihood = 204.46,  aic = -390.92
```

$$(1 - 0.8744B - 0.0329B^2 - 0.1175B^3 + 0.1243B^4 + 0.0635B^5 - 0.1457B^6)(y_{3t} - 1.6424 - 0.7469y_{1t}) = a_t$$

```
ggtsdiag(m9, gof.lag = 24)
```



Part B

Refine the model by letting the insignificant coefficients of lags 2 and 5 to zero. Write down the fitted model.

```
mask2 = c(NA,0,NA,NA,0,NA,NA,NA)
m10 = arima(t6, order = c(6,0,0), xreg = t5, fixed = mask2)
```

```
## Warning in arima(t6, order = c(6, 0, 0), xreg = t5, fixed = mask2): some AR
## parameters were fixed: setting transform.pars = FALSE
```

```
print(m10)
```

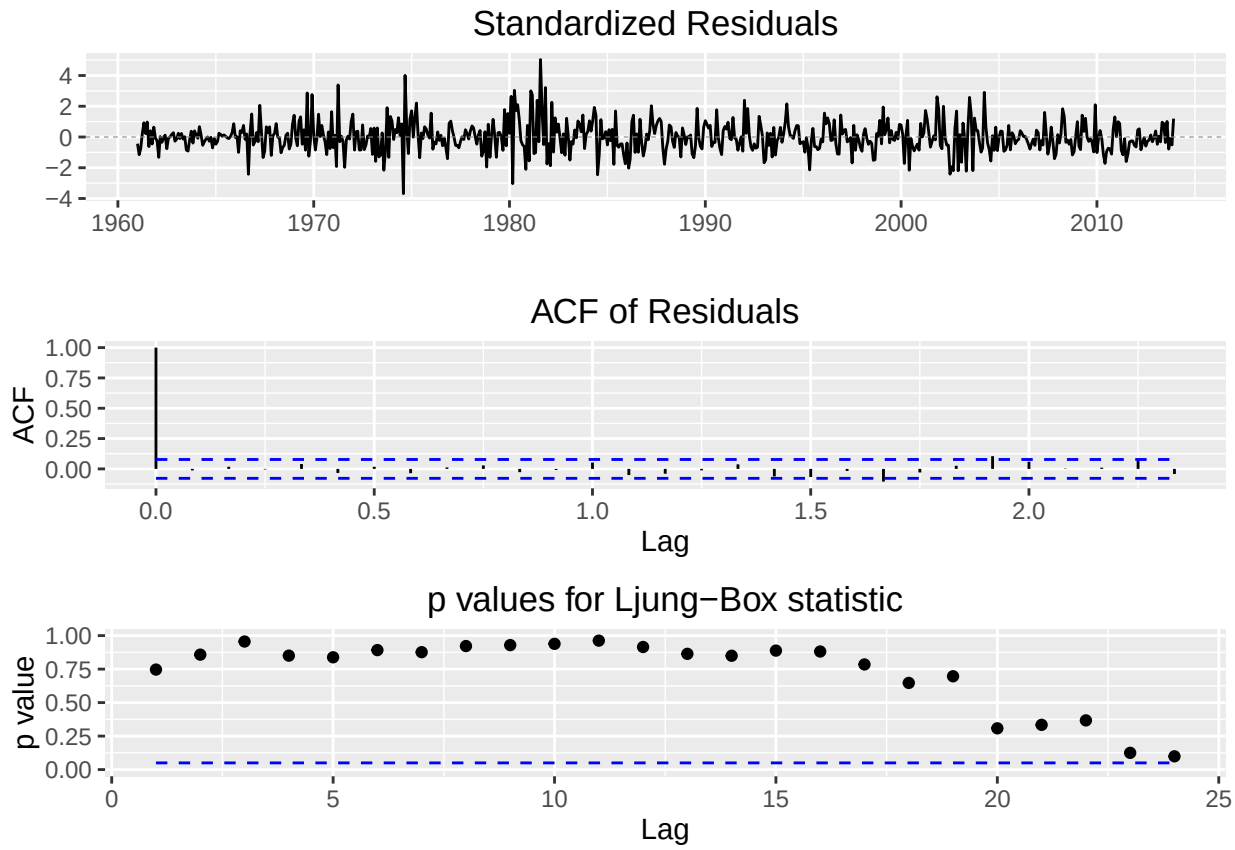
```
##
## Call:
## arima(x = t6, order = c(6, 0, 0), xreg = t5, fixed = mask2)
##
## Coefficients:
##          ar1  ar2     ar3      ar4  ar5     ar6  intercept      t5
##       0.8904    0  0.1332  -0.1553    0  0.1142     1.6336  0.7486
## s.e.  0.0295    0  0.0458   0.0458    0  0.0295     0.3622  0.0143
##
## sigma^2 estimated as 0.03071:  log likelihood = 203.63,  aic = -393.25
```

$$(1 - 0.8904B - 0.1332B^3 + 0.1553B^4 - 0.1142B^6)(y_{3t} - 1.6336 - 0.7486y_{1t}) = a_t$$

Part C

Is the refined model adequate? Why?

```
ggtsdiag(m10, gof.lag = 24)
```



The model appears adequate since all Ljung-Box p-values are > 0.05

Part D

Use the command polyroot in R to find the solutions of the characteristic equation of the refined AR(6) model. How many real solutions are there?

```
p1 = c(1, -m10$coef[1:6])
s1 = polyroot(p1)
print(s1)
```

```
## [1]  1.012815-0.000000i -0.642515+1.303265i -0.642515-1.303265i
## [4]  1.097951-0.960919i  1.097951+0.960919i -1.923688-0.000000i
```

There are 2 real solutions (1.012815 and -1.923688)

Part E

Compute the inverse of the absolute values of the solutions of the characteristic equation. Write down the maximum value of the inverses. The maximum should be close to 1, implying that the AR(6) model likely contains a unit root.

```
Mod(s1)
```

```
## [1] 1.012815 1.453040 1.453040 1.459062 1.459062 1.923688
```

```
1 / Mod(s1)
```

```
## [1] 0.9873474 0.6882125 0.6882125 0.6853719 0.6853719 0.5198348
```

This verified the maximum (0.9873) is close to 1 so the model likely contains a unit root.