

# Assignment 3: ARIMA Models – Fundamental Concepts

*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 L<sup>A</sup>T<sub>E</sub>X, 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(ggthemes)    # for graphing beautifully
require(gridExtra)   # for laying out graphs
```

## Part 1

Consider the monthly series of Consumer Sentiment of the University of Michigan. This survey series is widely used to indicate the consumer confidence about the U.S. economy. The data are available from FRED of the Federal Reserve Bank of St. Louis and also in the file `m-umcsent.txt`. The sample period is from January 1978 to August 2013.

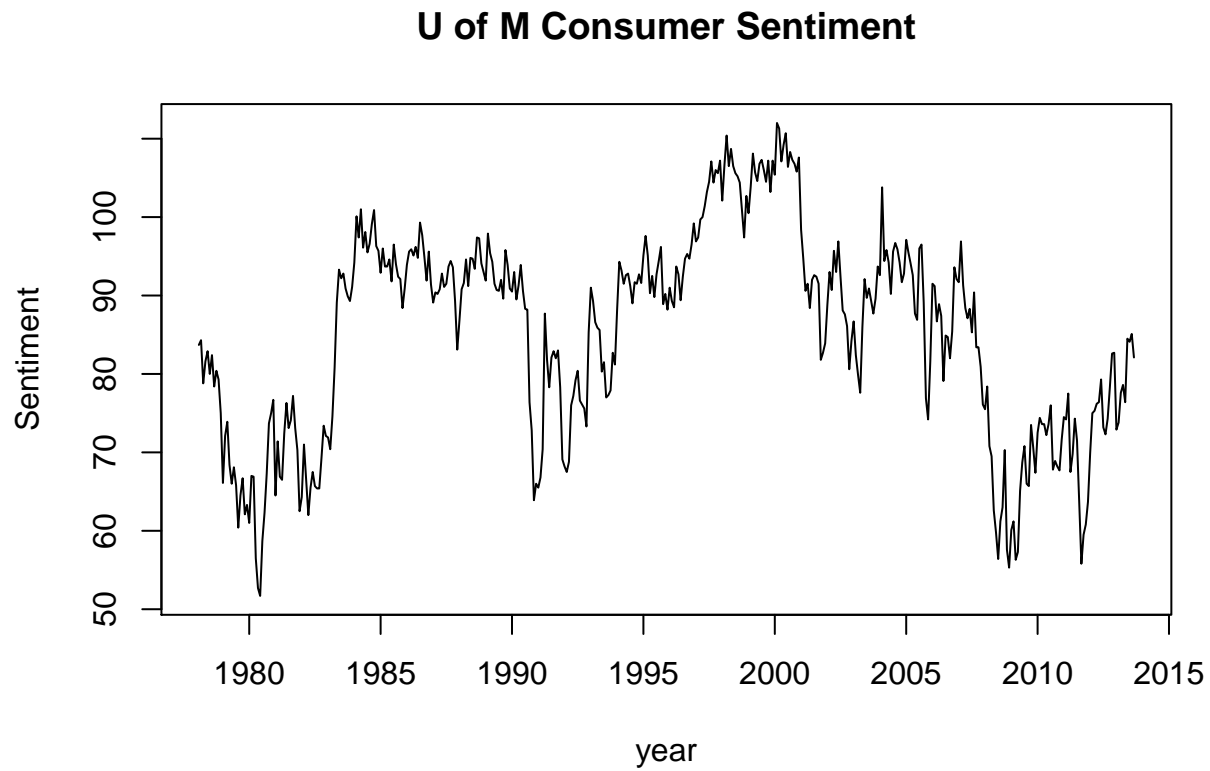
```
d1 = read.table("data/m-umcsent.txt", header=T)
head(d1)
```

```
##   year mon day VALUE
## 1 1978   1   1  83.7
## 2 1978   2   1  84.3
## 3 1978   3   1  78.8
## 4 1978   4   1  81.6
## 5 1978   5   1  82.9
## 6 1978   6   1  80.0
```

## Part A

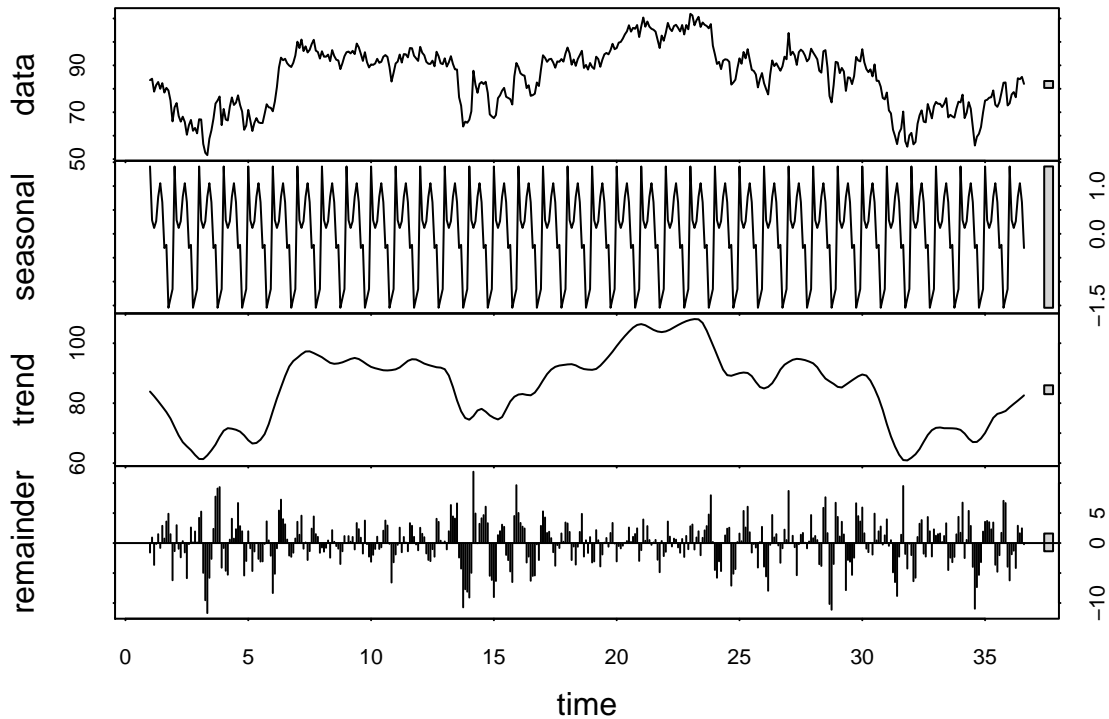
Plot the consumer monthly sentiment series:

```
consent = d1$VALUE
tdx = d1[,1] + d1[,2] / 12 # assemble the time axis
plot(tdx, consent, xlab = "year", ylab = "Sentiment", type = "l", main = "U of M Consumer Sentiment")
```



We can see a meandering trend in sentiment. We know that the datas are recoreded at monthly intervals so we will perform an STL decomposition to investigate our suspicion that this data has a seasonal component.

```
a <- consent
a.ts <- ts(a, frequency = 12)
consent_stl = stl(a.ts, s.window="periodic")
plot(consent_stl)
```



We see a seasonal component, however with this decomposition we are able to notice more distinctly the meandering of the trend.

## Part B

Is there a unit root in the monthly sentiment series? Why?:

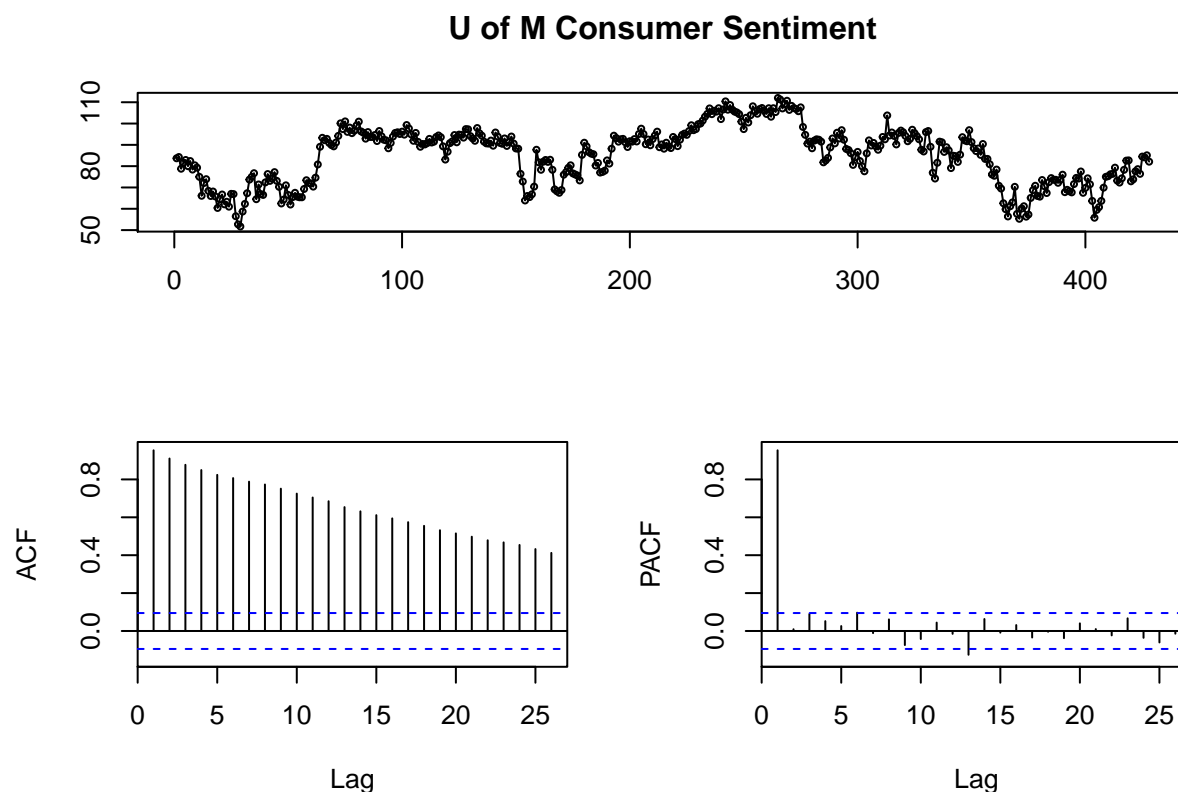
We will use the Augmented Dickey-Fuller Test, which computes the Augmented Dickey-Fuller test for the null that the time series has a unit root:

```
adf.test(consent)

##
## Augmented Dickey-Fuller Test
##
## data: consent
## Dickey-Fuller = -2.1108, Lag order = 7, p-value = 0.5307
## alternative hypothesis: stationary
```

Which, due to the p-value of 0.5306 meaning we fail to reject the null hypothesis. This provides us with strong indication that the timeseries is non-stationary.

```
tsdisplay(consent, main = "U of M Consumer Sentiment")
```



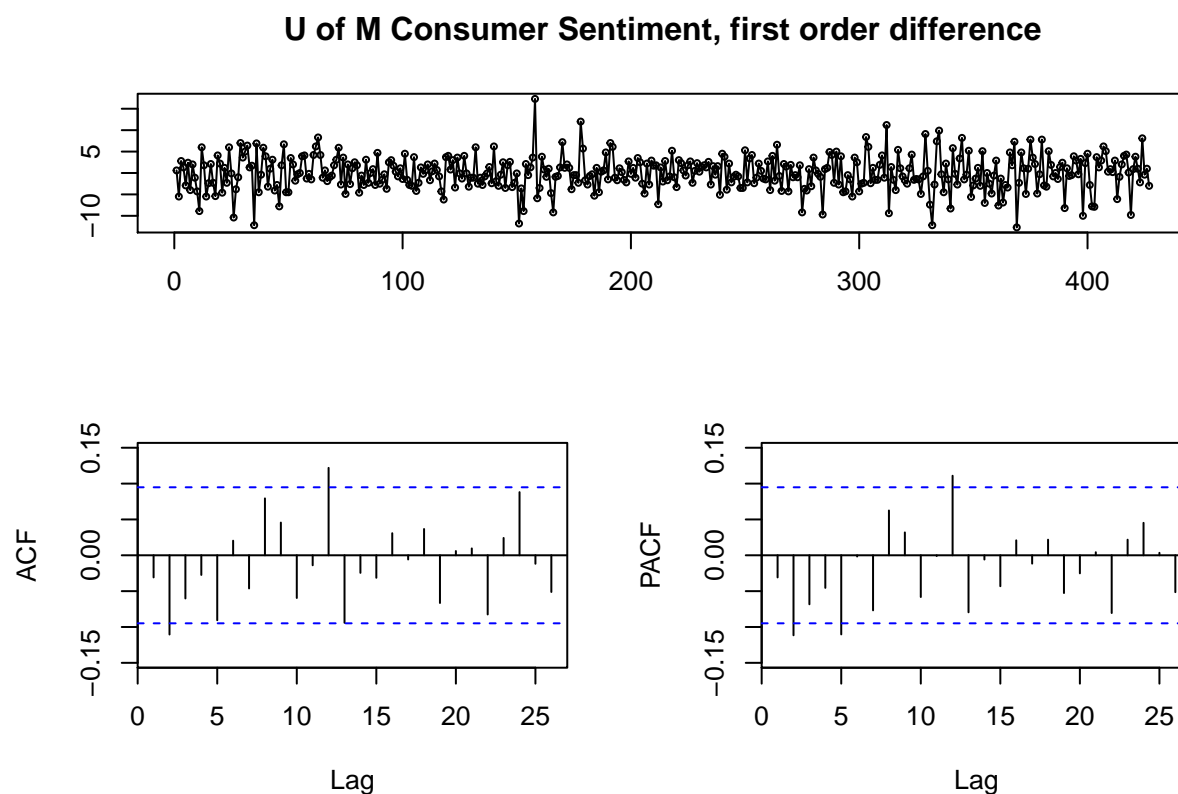
We're seeing what we think is an AR(1) process, because we see a large spike at the first lag, but we don't see any more significant spikes beyond the first lag. Where as, if we saw a second large spike at the second lag it would be an indicator of an AR(2) process.

This also means we're not seeing a series that is stationary, as we'd expect to see the PACF graph not have any particular spikes. We should consider performing a first order differencing of the series.

## Part C

Consider the change series of the sentiment, (i.e. the first differenced data). Test the hypothesis that the expected change of the sentiment is zero versus the alternative that the expected change is non-zero.

```
change = diff(sentiment)
tsdisplay(change, main = "U of M Consumer Sentiment, first order difference")
```



We see spikes above the interval at 2, 5, and 12. So we'll choose a lag order of 12 to calculate the test statistic.

We'll further investigate by fitting an autoregressive time series model to the data to see what the automatic order selection turns up:

```
ar(change, method = "mle")

##
## Call:
## ar(x = change, method = "mle")
##
## Coefficients:
##      1      2      3      4      5
## -0.0499 -0.1266 -0.0830 -0.0498 -0.1107
##
## Order selected 5  sigma^2 estimated as 15.46
```

We observe that the order selected is 5.

```
adf.test(change, k = 5)
```

```
## Warning in adf.test(change, k = 5): p-value smaller than printed p-value
```

```
##
```

```
## Augmented Dickey-Fuller Test
```

```
##
```

```
## data: change
```

```
## Dickey-Fuller = -9.9655, Lag order = 5, p-value = 0.01
```

```
## alternative hypothesis: stationary
```

From this we see reject the null hypothesis and posit that the first order difference of consent is a stationary series.

## Part D

Focus on the change series. Test the null hypothesis  $H_0 : \rho_1 = \rho_2 = \dots \rho_{12} = 0$  versus the alternative  $H_a : \rho_i \neq 0$  for some  $i \in [1, 12]$ . Draw your conclusion.

We first will examine whether the mean of the first order difference series is equal to zero:

```
t.test(change)
```

```
##  
## One Sample t-test  
##  
## data: change  
## t = -0.019349, df = 426, p-value = 0.9846  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.3843808 0.3768867  
## sample estimates:  
## mean of x  
## -0.003747073
```

from the p-value of 0.9846 we are not able to reject the null hypothesis, indicating it is highly likely that the true mean is equal to zero.

We will perform a Box-Pierce and Ljung-Box Test to compute a Ljung test statistic for examining the null hypothesis of independence given a time series. This is also known as a **portmanteau** test.

```
Box.test(change, lag = 12, type = "Ljung")
```

```
##  
## Box-Ljung test  
##  
## data: change  
## X-squared = 24.111, df = 12, p-value = 0.01965
```

From the p-value of 0.01965 we must reject  $H_0$ . This is an indicator that there are some significant serial correlations at the 5% level for the first order difference series. We should have expected as much from consideration of the ACF plot above, when we observed exceedences in 2, 5, and 12.



## Part 2

Again, Consider the change series of consumer sentiment within Problem 1.

### Part A

Use the command `ar` with method `mle` to find the order:

We did this in Part 1, C and found the order to be 5.

### Part B

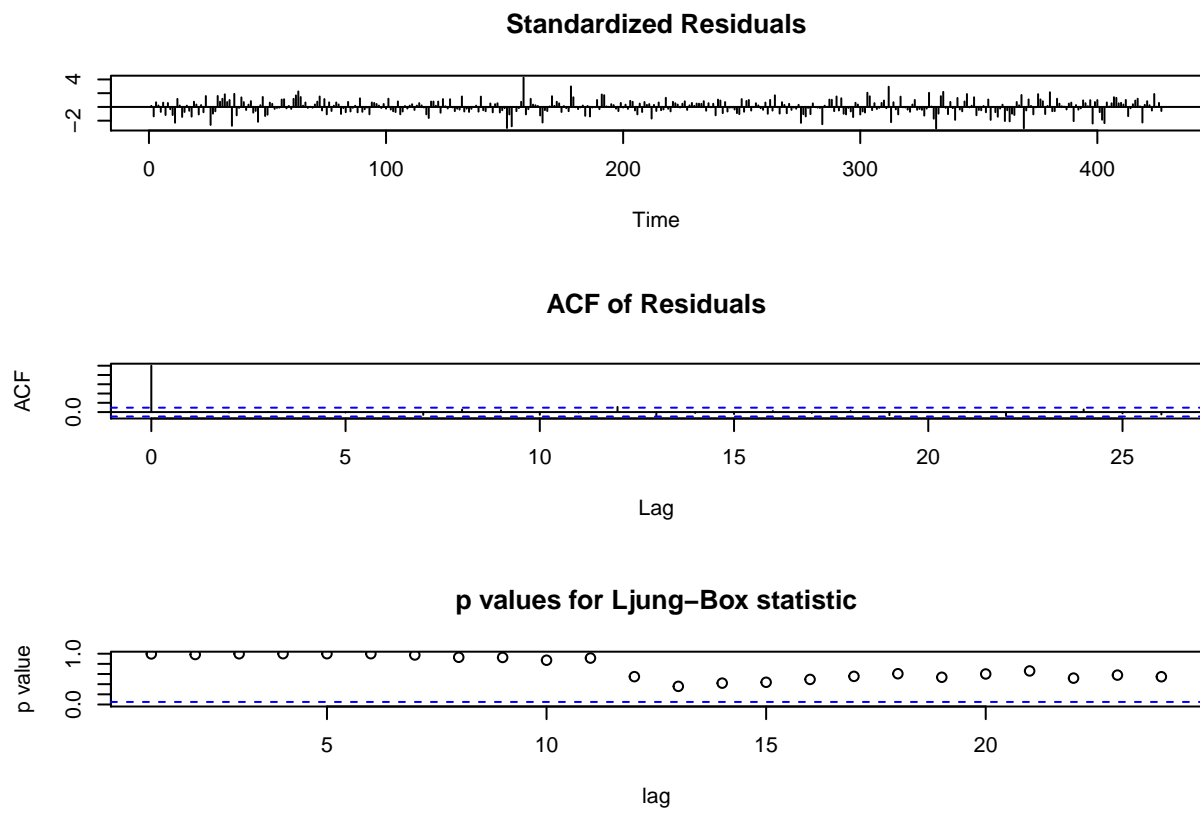
Build an Auto Regressive model using `ar` based on the selected order for the change series. Perform model checking to validate the fitted model. Write down the model form.

```
model1 = arima(change, order= c(5,0,0))
print(model1)
```

```
##
## Call:
## arima(x = change, order = c(5, 0, 0))
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5  intercept
##      -0.0499  -0.1266  -0.0830  -0.0498  -0.1107   -0.0019
## s.e.    0.0481   0.0481   0.0483   0.0483   0.0483    0.1343
##
## sigma^2 estimated as 15.46:  log likelihood = -1190.59,  aic = 2395.17
```

$$y_t = -0.0019 - 0.0499y_{t-1} - 0.1266y_{t-2} - 0.083y_{t-3} - 0.0498y_{t-4} - 0.1107y_{t-5} + e_t$$

```
tsdiag(model1, gof.lag = 24)
```



## Part C

Does the model imply the existence of business cycles in consumer sentiment? Why?

We will use our model coefficients, along with the expression of the characteristic equation to compute the average length of business cycles:

```
poly1 = c(1, -model1$coef) # characteristic equation
roots1 = polyroot(poly1)   # find the zeros of the complex polynomial
print(roots1)
```

```
## [1] 1.073056+1.039425i -0.477160+1.506788i -0.477160-1.506788i
## [4] 1.073056-1.039425i -1.632829+0.000000i -58.820675+0.000000i
```

```
print(Mod(roots1))
```

```
## [1] 1.493939 1.580535 1.580535 1.493939 1.632829 58.820675
```

```
print(2 * pi / acos(1.073056 / 1.493939))
```

```
## [1] 8.165501
```

This indicates implication of a two year business cycle in consumer sentiment.

Calculations were drawn from Tsay (2005) page 45.

## Part D

Obtain 1-step, to 4-step ahead point and 95% interval forecasts for the change series of consumer sentiment at the forecast origin of August 1, 2013.

```
model1_pred = predict(model1, 4)
print(model1_pred$pred)
```

```
## Time Series:
## Start = 428
## End = 431
## Frequency = 1
## [1] -0.1065526 -0.5769096 0.2829894 0.1039243
```

```
print(model1_pred$se)
```

```
## Time Series:
## Start = 428
## End = 431
## Frequency = 1
## [1] 3.932174 3.937061 3.967210 3.976876
```

```
model1_pred_lcl = model1_pred$pred - 1.96 * model1_pred$se
model1_pred_ucl = model1_pred$pred + 1.96 * model1_pred$se
print(model1_pred_lcl)
```

```
## Time Series:
## Start = 428
## End = 431
## Frequency = 1
## [1] -7.813613 -8.293549 -7.492742 -7.690752
```

```
print(model1_pred_ucl)
```

```
## Time Series:
## Start = 428
## End = 431
## Frequency = 1
## [1] 7.600508 7.139730 8.058721 7.898601
```

## Part 3

Consider, Again, the monthly change series of consumer sentiment withing Problem 1.

### Part A

Simplify the fitted Auto Regressive model of Problem 2 by removing parameter estimates with a t-ratio less than 1.2 in absolute (use the fixed subcommand).

In looking at , we create a mask to remove the paramater estimates with a t-ratio less than 1.2:

```
model1.se = sqrt(diag(vcov(model1)))
model1.tratio = abs(model1$coef/model1.se)
print(model1.tratio)
```

```
##          ar1          ar2          ar3          ar4          ar5 intercept
## 1.0373573 2.6310629 1.7174694 1.0323956 2.2933072 0.0139051
```

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

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

We then use the fixed argument in arima to construct a second model with this mask:

```
model2 = arima(change, order=c(5,0,0), include.mean = FALSE, transform.pars = FALSE, fixed=mask)
print(model1)
```

```
##
## Call:
## arima(x = change, order = c(5, 0, 0))
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5 intercept
##      -0.0499  -0.1266  -0.0830  -0.0498  -0.1107   -0.0019
## s.e.    0.0481   0.0481   0.0483   0.0483   0.0483    0.1343
##
## sigma^2 estimated as 15.46:  log likelihood = -1190.59,  aic = 2395.17
```

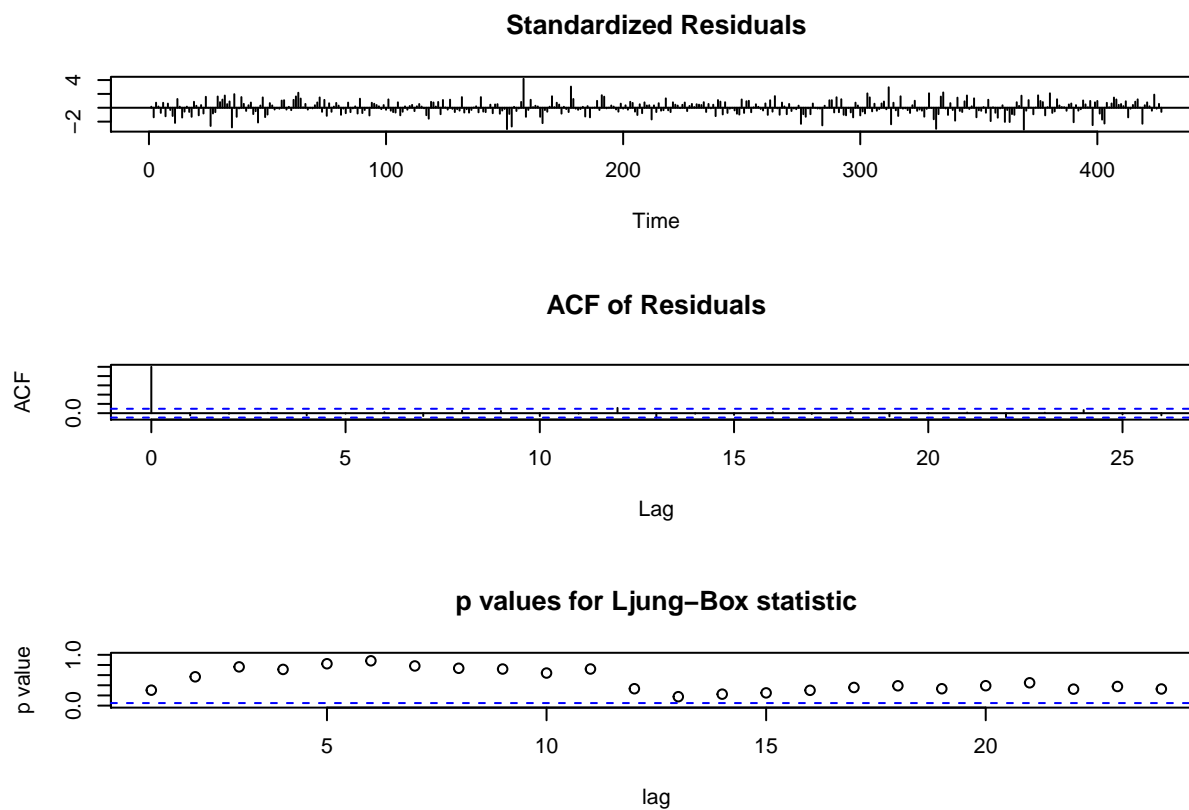
```
print(model2)
```

```
##
## Call:
## arima(x = change, order = c(5, 0, 0), include.mean = FALSE, transform.pars = FALSE,
##      fixed = mask)
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5
##           0  -0.1191  -0.0754           0  -0.1067
## s.e.       0   0.0479   0.0480           0   0.0483
##
## sigma^2 estimated as 15.54:  log likelihood = -1191.59,  aic = 2391.17
```

## Part B

Is the model adequate? Why?

```
tsdiag(model2, gof.lag = 24)
```



```
Box.test(model1$residuals, lag = 24, type = "Ljung")
```

```
##  
## Box-Ljung test  
##  
## data: model1$residuals  
## X-squared = 22.553, df = 24, p-value = 0.5463
```

The Ljung-Box test of the model residuals reveals a p-value that is not significant, we surmise that the model is adequate.

## Part C

Compare the simplified model with the Auto Regressive model built in Problem 2. In terms of in-sample fitting, which model is preferred? Why?

```
accuracy(model1)
```

```
##               ME      RMSE      MAE MPE MAPE      MASE      ACF1
## Training set -0.00235076 3.932174 3.00338 NaN  Inf 0.6689532 -0.000263695
```

```
accuracy(model2)
```

```
##               ME      RMSE      MAE MPE MAPE      MASE      ACF1
## Training set -0.004376082 3.941464 3.008938 NaN  Inf 0.6701912 -0.04960276
```

In terms of in sample fitting the RMSE of the first model is lower.

```
Box.test(model1$residuals, lag = 5, type = "Ljung")
```

```
##
## Box-Ljung test
##
## data:  model1$residuals
## X-squared = 0.075281, df = 5, p-value = 0.9999
```

```
Box.test(model2$residuals, lag = 3, type = "Ljung")
```

```
##
## Box-Ljung test
##
## data:  model2$residuals
## X-squared = 1.1605, df = 3, p-value = 0.7625
```

If both models are adequate, we will then have to choose based on AIC or BIC criteria.

```
model1$aic
```

```
## [1] 2395.173
```

```
model2$aic
```

```
## [1] 2391.174
```

The second model has slightly lower AIC.

## Part D

Does the simplified model imply the existence of business cycles? Why?

We will use our model coefficients, along with the expression of the characteristic equation to compute the average length of business cycles:

```
poly2 = c(1, -model2$coef) # characteristic equation
roots2 = polyroot(poly2)    # find the zeros of the complex polynomial
print(roots2)
```

```
## [1] 1.157751+1.052643i -1.565269-0.000000i -0.375116-1.518011i
## [4] 1.157751-1.052643i -0.375116+1.518011i
```

```
print(Mod(roots2))
```

```
## [1] 1.564751 1.565269 1.563672 1.564751 1.563672
```

```
print(2 * pi / acos(1.157751 / 1.594751))
```

```
## [1] 8.285405
```

This indicates implication of a two year business cycle in consumer sentiment. It shows a 0.12 longer business cycle than the first model.

Calculations were drawn from Tsay (2005) page 45.

## Part E

Use the backtest to compare the two Auto Regressive models. You may start the forecast origin at  $t = 380$ . Which model is preferred? Why?



## References

Tsay, Ruey S. 2005. *Analysis of Financial Time Series*. Vol. 543. John Wiley & Sons.