

# Time Series Analysis - STAT 478 - HW #6

Alex Towell (atowell@siue.edu)

## Contents

<b>Problem 1</b>	<b>1</b>
Preliminary analysis . . . . .	1
Part (a) . . . . .	2
Part (b) . . . . .	3
Part (c) . . . . .	4
Part (d) . . . . .	5
Part (e) . . . . .	6
<b>Problem 2</b>	<b>6</b>
Preliminary steps . . . . .	6
Part (a) . . . . .	6
Part (b) . . . . .	9
Part (c) . . . . .	12
Additional thoughts: evaluating model performance via data splitting . . . . .	13
<b>Problem 3</b>	<b>15</b>
Part (a) . . . . .	16
Part (b) . . . . .	17
Part (c) . . . . .	18
Part (d) . . . . .	19

## Problem 1

A data set of 324 measurements of an industrial robot's positions are in the robot object in the TSA package.

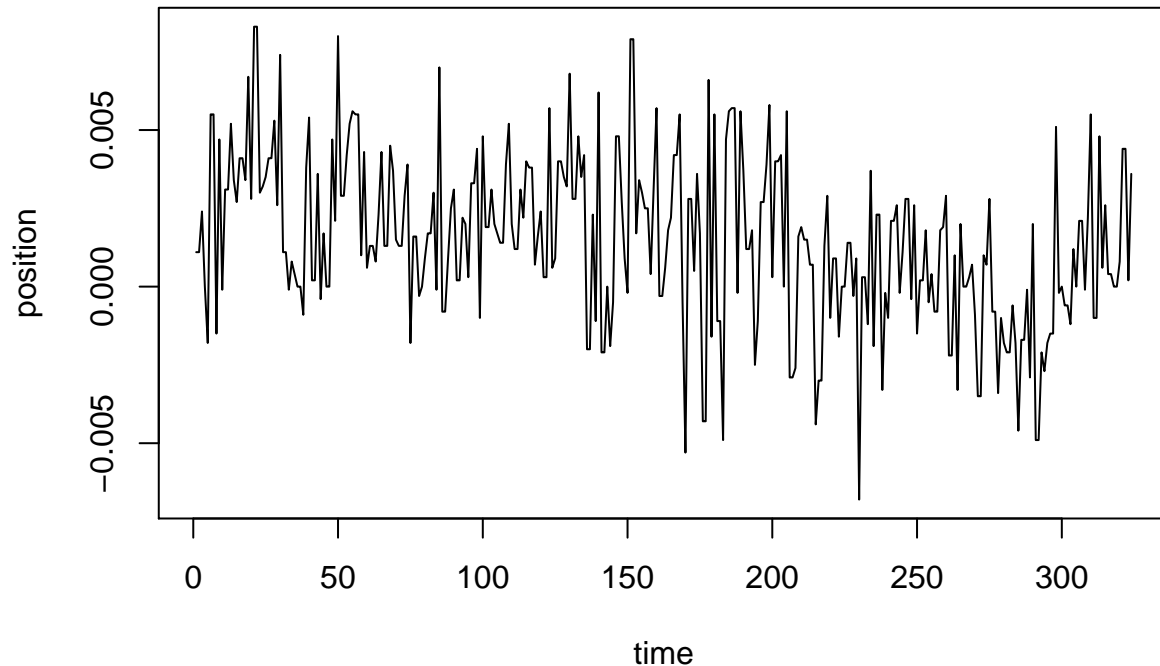
```
library(TSA)
data(robot)
head(robot)
```

```
## [1] 0.0011 0.0011 0.0024 0.0000 -0.0018 0.0055
```

### Preliminary analysis

We denote the industrial robot's position time series as  $\{P_t\}$ . A quick plot:

```
plot(robot,xlab="time",ylab="position")
```



### Part (a)

Fit an AR(1) model for these data. Give the equation of the estimated model.

We suppose that  $\{P_t\} \sim \text{AR}(1)$ , defined as

$$P_t = c + \phi P_{t-1} + e_t$$

where  $e_t$  are the residuals in the model. If the model is a good fit,  $e_t$  approximates  $\text{WN}(0, \sigma_e^2)$ , zero-mean white noise.

We fit the data to the AR(1) model with:

```
robot.ar1 <- arima(robot, order=c(1,0,0))
print(robot.ar1)
```

```
##
## Call:
## arima(x = robot, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##          0.3074    0.0015
## s.e.    0.0528    0.0002
##
## sigma^2 estimated as 6.482e-06:  log likelihood = 1475.54,  aic = -2947.08
robot.ar1.phi <- coef(robot.ar1)[1]
robot.ar1.c <- coef(robot.ar1)[2]
robot.ar1.mu <- robot.ar1.c / (1-robot.ar1.phi^2)

# robot.ar1.resid <- robot.ar1 - robot
# plot(robot.ar1.resid)
# robot.ar1.resid.mu <- mean(robot.ar1.resid)
# print(robot.ar1.resid.mu)
```

```
# robot.ar1.resid.var <- var(robot.ar1.resid)
# print(robot.ar1.resid.var)
```

The estimate of the AR(1) model is given by

$$\hat{P}_t = 0.3073973\hat{P}_{t-1} + 0.0014551 + \hat{e}_t,$$

where ideally  $\hat{e}_t \sim \text{WN}(0, \sigma_{\hat{e}}^2 = 6.4822192 \times 10^{-6})$ . which has an estimated mean

$$E(\hat{P}_t) = \frac{\hat{c}}{1 - \hat{\phi}} = 0.001607.$$

and an estimated variance

$$\text{Var}(\hat{P}) = \frac{\sigma_{\hat{e}}^2}{1 - \hat{\phi}^2} = 7.1586634 \times 10^{-6}.$$

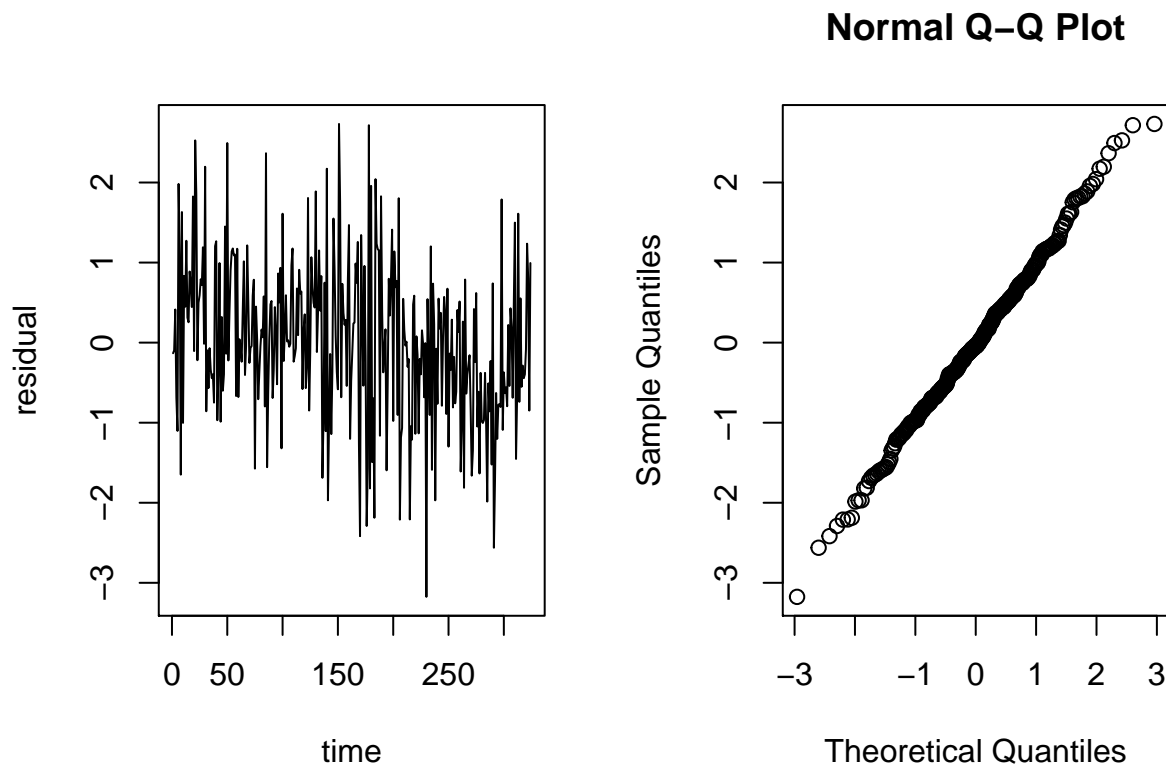
Note that the sample mean and sample variance of  $\{P_t\}$  is given respectively by 0.0014515 and  $7.1844868 \times 10^{-6}$ , which is reasonably close to the AR(1) model estimates. However, we need the residuals of this model to model white noise for this to be a good model of  $\{P_t\}$ , which we check for next.

## Part (b)

Give a basic plot of the standardized residuals over time and a Q-Q plot of the residuals. Comment on what these tell you about the adequacy of the model.

We do time series and Q-Q plots of the standardized residuals with:

```
par(mfrow=c(1,2))
plot(rstandard(robot.ar1), xlab="time", ylab="residual")
qqnorm(rstandard(robot.ar1))
```



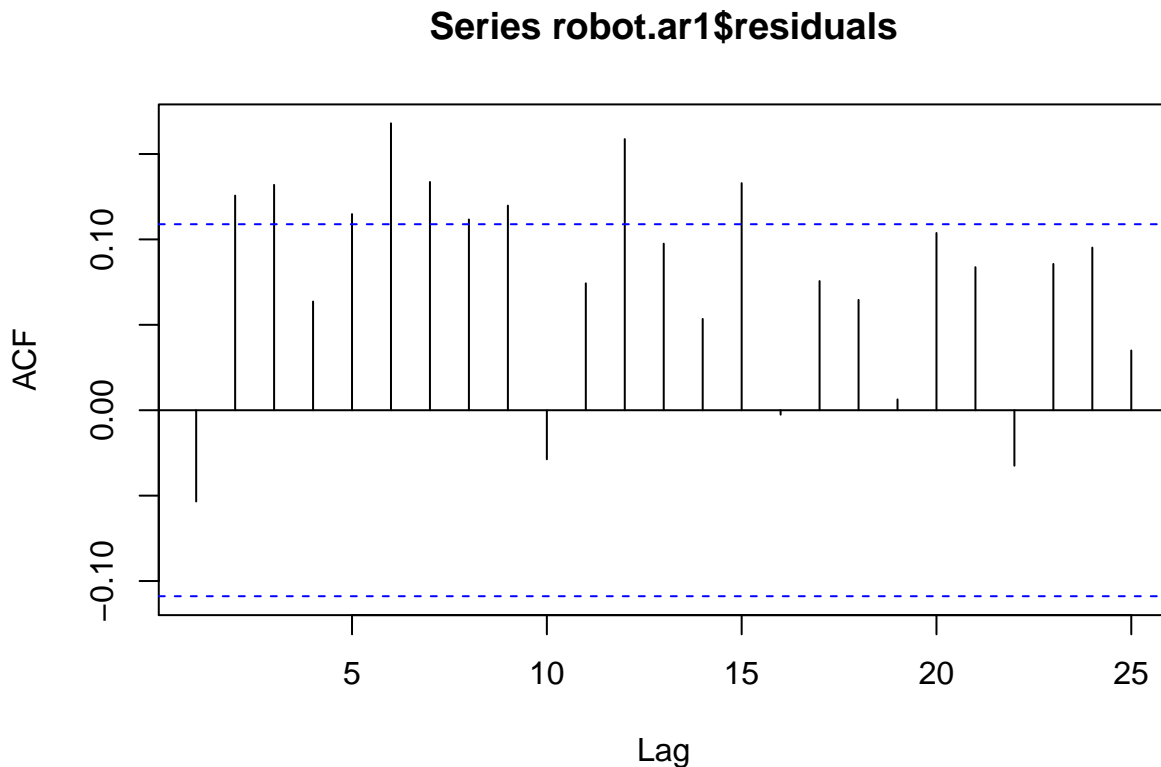
The Q-Q plot is consistent with normality, but there may be an issue with the time series plot of the residuals, which does not appear to model zero mean white noise. There appears to be some regularity in the residuals which should be captured by our model.

### Part (c)

Give a plot of the sample autocorrelation function of the residuals. Also perform a Ljung-Box test (with  $K = 30$ ). Comment on what these tell you about whether the errors are independent in this model.

First, we plot the sample ACF:

```
acf(robot.ar1$residuals)
```



The sample ACF does not indicate zero mean white noise since we see autocorrelations that frequently fall outside of the confidence intervals for zero mean white noise.

However, it may be difficult to determine from these plots whether the ACF is compatible with white noise when we look at the lags separately. We perform the Ljung-Box to perform the hypothesis test

$$H_0 : r_1 = r_2 = \dots = r_{30} = 0$$

whose test statistic is  $\chi^2$  distributed with  $df = K - p = 29$  degrees of freedom under the null hypothesis of zero mean white noise.

```
K <- 30
p <- 1
q <- 0
Box.test(robot.ar1$residuals,
         lag=K,
         type="Ljung-Box",
         fitdf=p+q)
```

```
##
## Box-Ljung test
##
## data:  robot.ar1$residuals
## X-squared = 83.243, df = 29, p-value = 3.845e-07
```

We see that the probability that a white noise process generates the observed test statistic  $X^2 = 83.243$  occurs with probability less than 0.001, which we consider to be very strong evidence against the hypothesis that the residuals are a zero mean white noise process.

We conclude that the model fails to capture some of the regularity in the time series data. We would prefer a model that essentially models all the regularity, leaving only uncorrelated white noise in the residuals.

## Part (d)

Fit an IMA(1,1) model for these data. Give the equation of the estimated model.

The IMA(1,1) model is equivalent to the AMIMA(0,1,1) model,

$$\nabla\{Y_t\} \sim \text{MA}(1).$$

```
robot.ima11 <- arima(robot,order=c(0,1,1))
robot.diff <- diff(robot,1)
robot.diff.ma1 <- arima(robot.diff,order=c(0,0,1))

print(robot.ima11)
```

```
##
## Call:
## arima(x = robot, order = c(0, 1, 1))
##
## Coefficients:
##          ma1
##       -0.8713
## s.e.    0.0389
##
## sigma^2 estimated as 6.069e-06:  log likelihood = 1480.95,  aic = -2959.9

print(robot.diff.ma1)
```

```
##
## Call:
## arima(x = robot.diff, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       -0.8716         0e+00
## s.e.    0.0397         1e-04
##
## sigma^2 estimated as 6.069e-06:  log likelihood = 1480.95,  aic = -2957.91
```

The estimate of the IMA(1,1) model is given by

$$\hat{P}_t =$$

where  $\hat{e}_t \sim \text{WN}(0, \sigma_e^2 = 6.4822192 \times 10^{-6})$ . which has an estimated mean

$$E(\hat{P}_t) = \frac{\hat{c}}{1 - \hat{\phi}} = 0.001607.$$

and an estimated variance

$$\text{Var}(\hat{P}) = \frac{\sigma_{\hat{e}}^2}{1 - \hat{\phi}^2} = 7.1586634 \times 10^{-6}.$$

We can undo the transformations to model the original time series,

$$\hat{P}_t =$$

## Part (e)

Compare the results from parts (a) and (d) using AIC.

```
print(robot.ar1$aic)
```

```
## [1] -2947.078
```

```
print(robot.ima11$aic)
```

```
## [1] -2959.901
```

The AIC is smaller (better) for the IMA(1,1) model, thus if we use the AIC measure to perform the model selection between these two candidate models, the IMA(1,1) should be selected.

## Problem 2

I have put the following dataset on blackboard: Gasprices: average price (US dollars per gallon) for regular gasoline in the United States; there are  $n = 145$  weekly observations collected from 1/5/2009 to 10/10/2011 (Source: Rajon Coles, Fall 2011). Using the methods from Chapter 5, identify a small set of candidate ARIMA( $p, d, q$ ) models for the dataset. You may need to transform the data before considering differencing. There may be a single model that emerges as a “clear favorite” or there may not. Write up detailed notes that describe how you decided on the model(s) you did. Your summary should convince me that your model(s) is (are) worthy of further consideration.

### Preliminary steps

We load the gas price data into a data frame named *gasprices*:

```
library(dplyr)
```

```
Yt <- ts(read.table(file="./gasprices.txt"))
```

The head of the data frame is:

```
head(Yt)
```

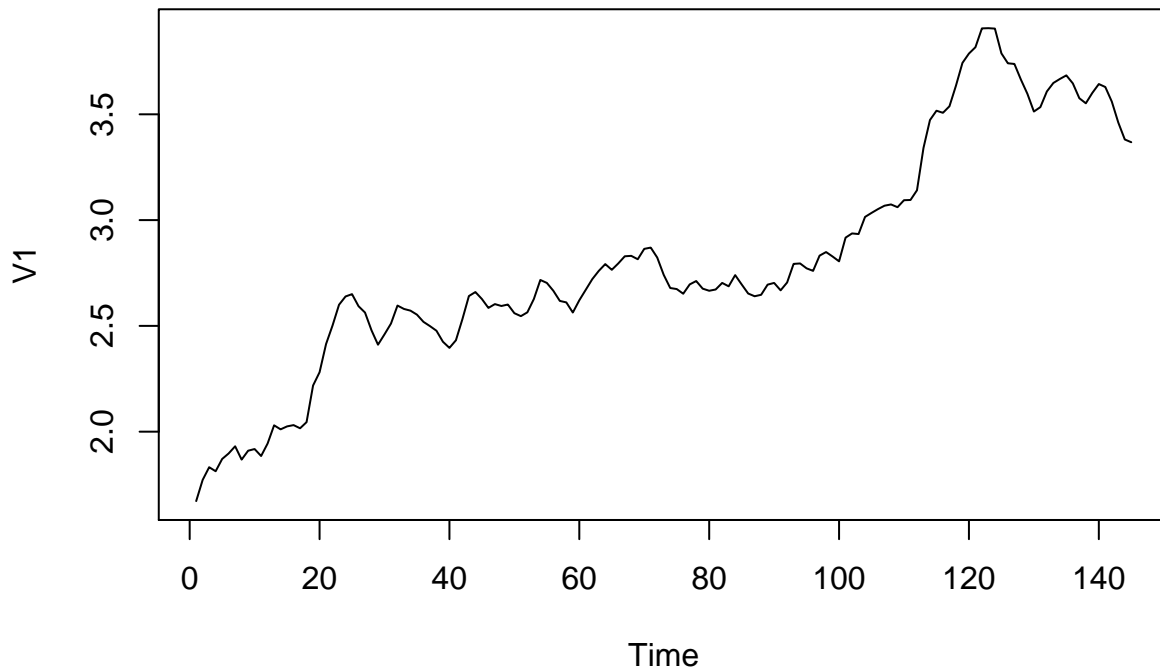
```
##          V1
## [1,] 1.672
## [2,] 1.772
## [3,] 1.832
## [4,] 1.813
## [5,] 1.871
## [6,] 1.897
```

### Part (a)

After a potential transformation, fit a set of candidate models (3 or fewer models) to the dataset. Discuss your findings.

We load the gas prices into a time series object and then plot it:

```
plot(Yt)
```



From the plot, the gas price is non-stationary. A few key observations:

1. The variance appears constant, so something like a log-transformation to transform a time series with non-constant variance to a time series with constant variance seems unnecessary.
2. The mean appears non-constant. In particular, the gas prices in the data seem to be trending upwards, but this is not necessarily the case, e.g., it may be a random walk process. I think we can summarize the plot with the idea that there is no natural mean.
3. There are no obvious cycles or seasonality in the data.

We model gas prices as a time series  $\{Y_t\}$ . The 1st order difference process,  $\{\nabla Y_t\}$ , is defined as

$$\nabla Y_t = Y_t - Y_{t-1}$$

and  $d$ -th order difference process  $\{\nabla^d Y_t\}$  is defined as

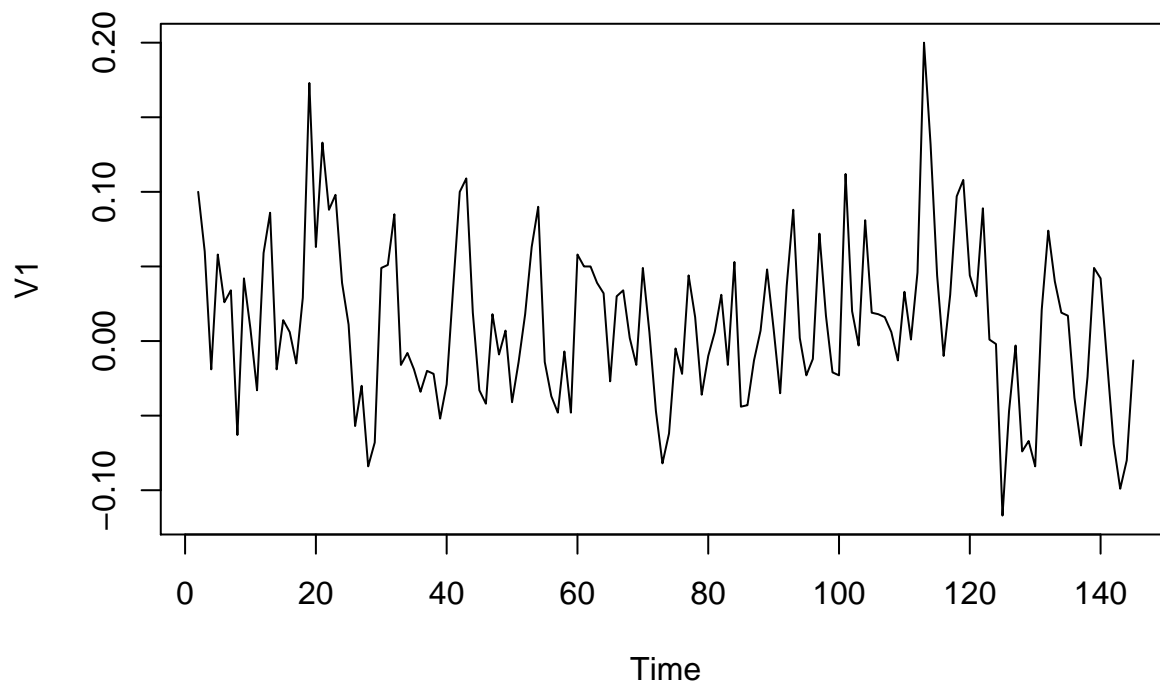
$$\begin{aligned}\nabla^0 Y_t &= Y_t \\ \nabla^d Y_t &= \nabla(\nabla^{d-1} Y_t).\end{aligned}$$

where  $\nabla^0 Y_t$  is the base case in this recursive definition.

If  $\{Y_t\}$  is non-stationary, it generally has a 1-st or 2-nd-order difference stationary process. (Higher-order difference processes are not commonly necessary.)

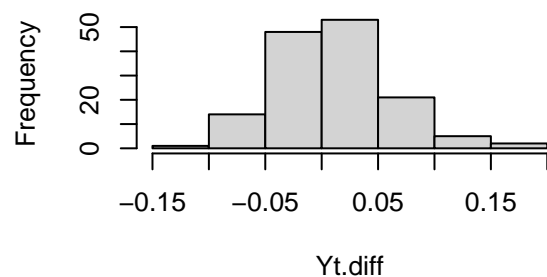
The first-order difference process of gas prices,  $\{\nabla Y_t\}$ , has the following plots:

```
Yt.diff <- diff(Yt,1)
plot(Yt.diff)
```

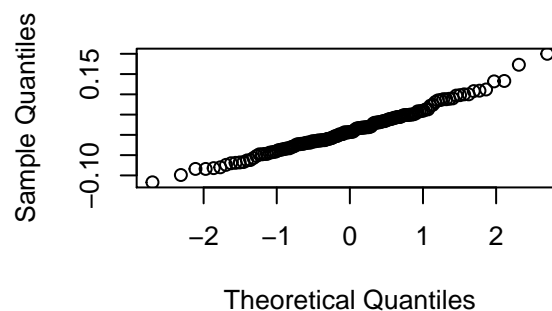


```
par(mfrow=c(2,2))
hist(Yt.diff)
qqnorm(Yt.diff)
acf(Yt.diff)
pacf(Yt.diff)
```

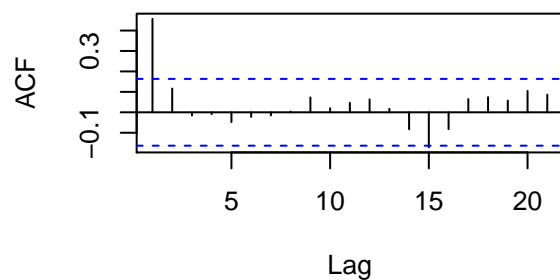
**Histogram of Yt.diff**



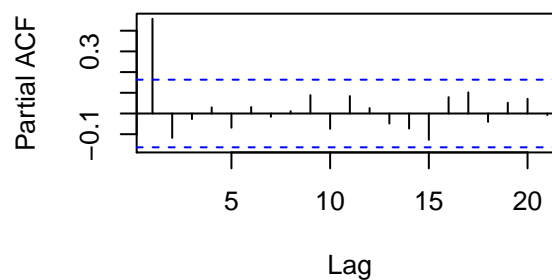
**Normal Q-Q Plot**



**Series Yt.diff**



**Series Yt.diff**



These plots of  $\{\nabla Y_t\}$  seem to sufficiently approximate white noise. In subsequent analysis, we will use this



differenced time series.

Looking at the ACF,  $\{\nabla Y_t\}$  seems consistent with MA, but the PACF is not what we expect. Likewise, looking at the PACF,  $\{\nabla Y_t\}$  is consistent with AR, but the ACF is not what we expect. We have somewhat of a mixture of both.

We plot the EACF to help decide on plausible ARMA models.

```
eacf(Yt.diff,ar.max=3,ma.max=3)
```

```
## AR/MA
##    0 1 2 3
## 0 x o o o
## 1 x o o o
## 2 x x o o
## 3 x o o o
```

We see that ARIMA(0, 1, 1), ARIMA(0, 1, 2), and ARIMA(1, 1, 1) seem like reasonable candidate models for  $\{Y_t\}$ .

This does not seem too suprising, given the ACF and PACF plots.

## Part (b)

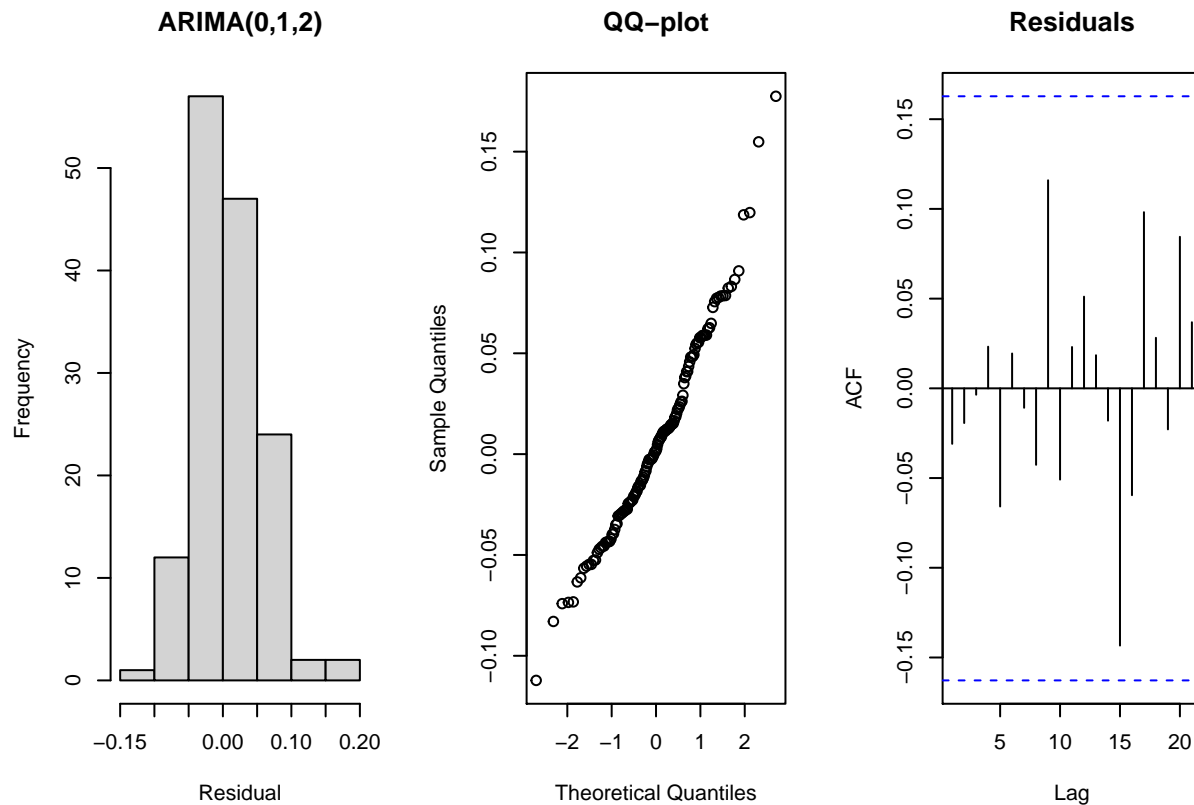
Perform diagnostic check for each model: check residuals for normality (histogram, qq-plot), the ACF, and the Ljung-Box test.

We evaluate each of the candidate models in the following subsections.

### Model 1: ARMA(0, 1, 2)

The histogram, QQ-plot, and ACF of the residuals for ARMA(0, 1, 1) are given by the following plots.

```
library(forecast)
modell1 <- Arima(Yt,order=c(0,1,2))
par(mfrow=c(1,3))
hist(modell1$residual,xlab="Residual",main="ARIMA(0,1,2)")
qqnorm(modell1$residual,main="QQ-plot")
acf(modell1$residual,main="Residuals")
```



The residuals seem like a reasonable approximation of white noise. Next, we do a Box-Ljung test:

```
K <- 30
p <- 0
q <- 2
Box.test(model1$residuals,
         lag=K,
         type="Ljung-Box",
         fitdf=p+q)

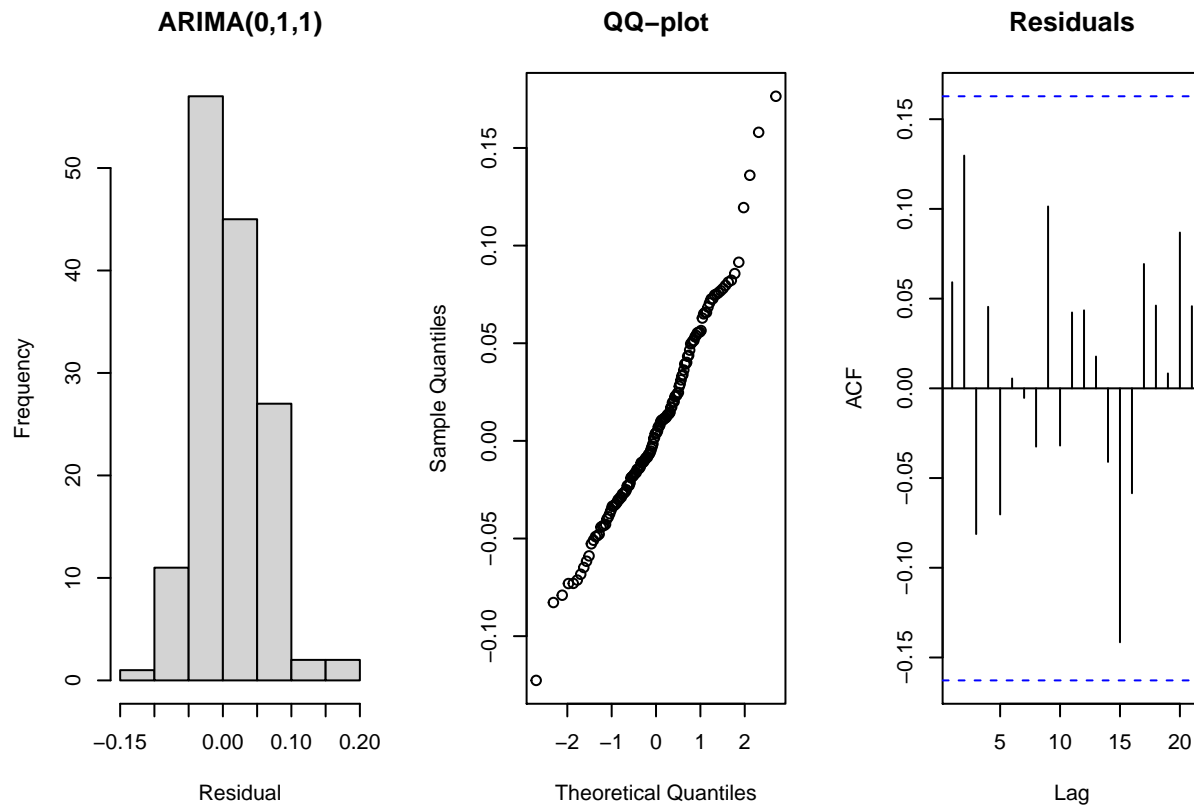
##
## Box-Ljung test
##
## data: model1$residuals
## X-squared = 17.798, df = 28, p-value = 0.9311
```

The observed test statistic is compatible with the hypothesis that the residuals are white noise.

## Model 2: ARMA(0,1,1)

The histogram, QQ-plot, and ACF of the residuals for ARMA(0,1,1) are given by the following plots.

```
library(forecast)
model2 <- Arima(Yt,order=c(0,1,1))
par(mfrow=c(1,3))
hist(model2$residual, xlab="Residual", main="ARIMA(0,1,1)")
qqnorm(model2$residual, main="QQ-plot")
acf(model2$residual, main="Residuals")
```



The residuals seem like a reasonable approximation of white noise. Next, we do a Box-Ljung test:

```
p <- 0
q <- 1
Box.test(model2$residuals,
         lag=K,
         type="Ljung-Box",
         fitdf=p+q)

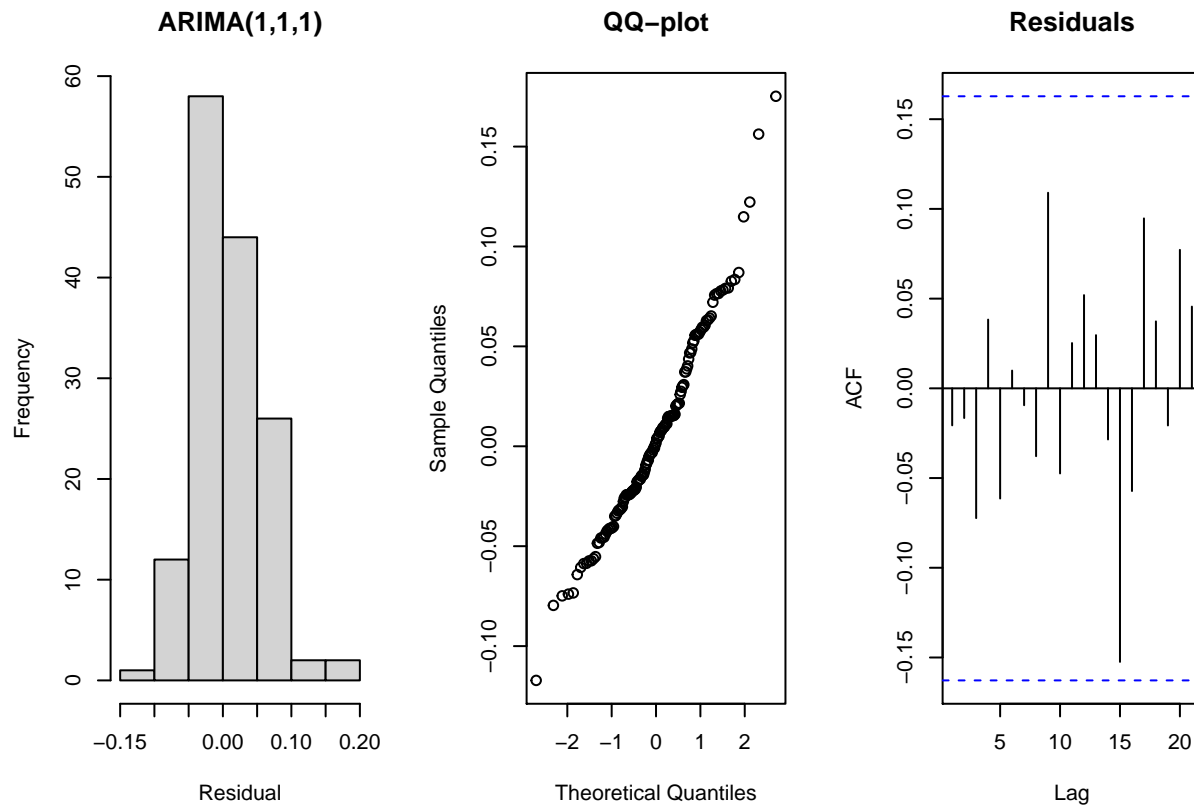
##
## Box-Ljung test
##
## data: model2$residuals
## X-squared = 22.266, df = 29, p-value = 0.809
```

The observed test statistic is compatible with the hypothesis that the residuals are white noise.

### Model 3: ARMA(1,1,1)

The histogram, Q-Q-plot, and ACF of the residuals for ARMA(1,1,1) are given by the following plots.

```
library(forecast)
model3 <- Arima(Yt,order=c(1,1,1))
par(mfrow=c(1,3))
hist(model3$residual, xlab="Residual", main="ARIMA(1,1,1)")
qqnorm(model3$residual, main="QQ-plot")
acf(model3$residual, main="Residuals")
```



The residuals seem like a reasonable approximation of white noise. Next, we do a Box-Ljung test:

```
p <- 1
q <- 1
Box.test(model3$residuals,
         lag=K,
         type="Ljung-Box",
         fitdf=p+q)
```

```
##
## Box-Ljung test
##
## data: model3$residuals
## X-squared = 18.654, df = 28, p-value = 0.9085
```

The observed test statistic is compatible with the hypothesis that the residuals are white noise.

## Part (c)

Choose a final model (if all fit fine, choose the one with the smallest AIC). Report your final model and calculate forecasts and prediction intervals for 5 future values. Display the forecasts and prediction bands visually like.

Both models fit fine, so we will use the model with the minimum AIC.

```
round(c(model1$aic,model2$aic,model3$aic),3)
```

```
## [1] -459.062 -455.297 -458.351
```

We see that the ARIMA(0,1,2) model for  $\{Y_t\}$  has the minimum AIC of these models with an AIC of -459.062, so we choose this model.

The model summary is given by:

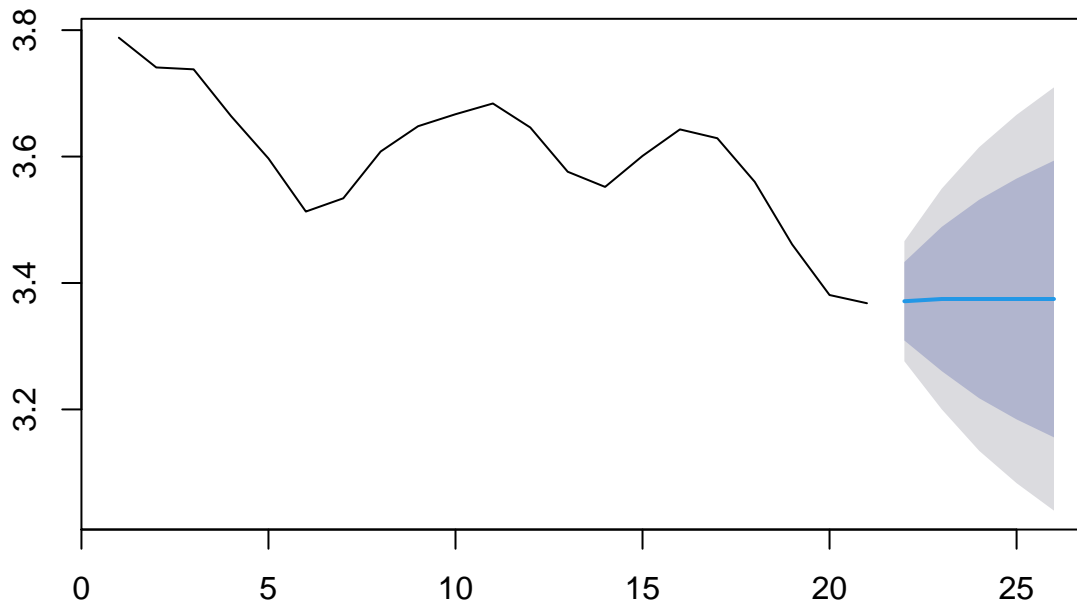
```
summary(model1)
```

```
## Series: Yt
## ARIMA(0,1,2)
##
## Coefficients:
##          ma1      ma2
##       0.5433  0.1946
## s.e.  0.0843  0.0779
##
## sigma^2 estimated as 0.002345:  log likelihood=232.53
## AIC=-459.06   AICc=-458.89   BIC=-450.15
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.006717107 0.04791789 0.03709482 0.2712316 1.381354 0.8635071
##              ACF1
## Training set -0.03092411
```

The forecast and prediction intervals for the ARMA(1,1,1) model are given by:

```
library(forecast)
model1.fc <- forecast(Yt[125:145],model=model1,h=5)
plot(model1.fc)
```

### Forecasts from ARIMA(0,1,2)



### Additional thoughts: evaluating model performance via data splitting

Out of curiosity, we evaluate the performance of the selected model, the fitted ARIMIA(1,1,1) model, via data splitting, i.e., learning on a training set and comparing the forecast with the test set.

```
library(forecast)
```

```

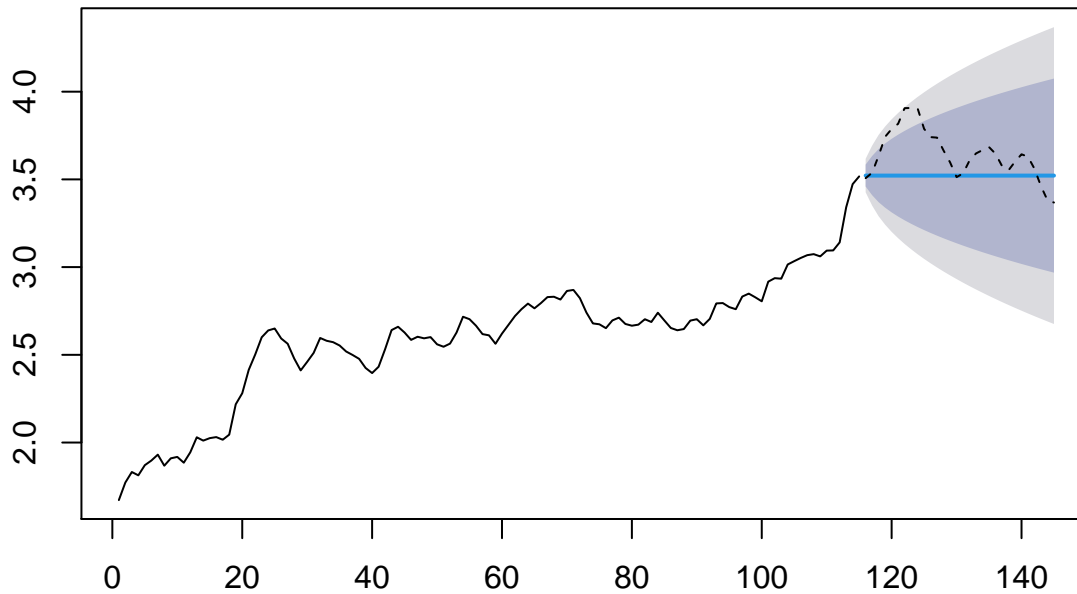
Yt.train <- ts(Yt[1:115],start=1)
Yt.test  <- ts(Yt[116:145],start=116)

model1.train <- Arima(Yt.train,order=c(0,1,2))
model1.train.fc <- forecast(Yt.train,model=model1.train,h=30)

plot(model1.train.fc)
lines(Yt.test, lty=2)

```

## Forecasts from ARIMA(0,1,2)



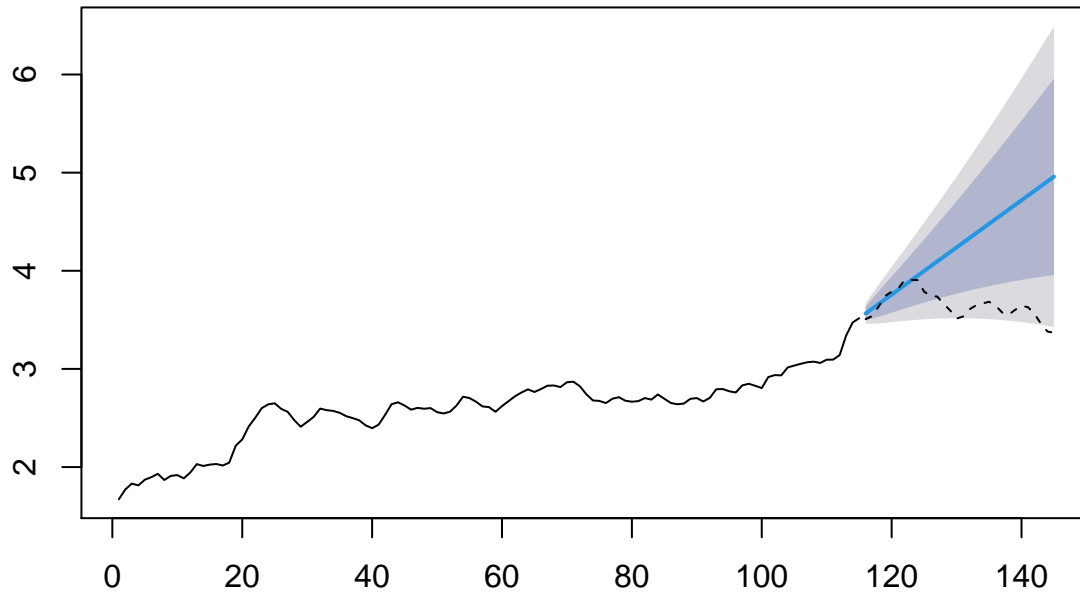
Over the long run, the forecast is not bad, but it is not very sensitive to recent history (the positive trend near the end of the training time series). Out of curiosity, we fit a double exponential smoothing model to the training set and then plot its forecast and prediction intervals:

```

dema.train <- holt(Yt.train,h=30,initial="optimal")
dema.train.fc <- forecast(Yt.train,model=dema.train,h=30)
plot(dema.train.fc)
lines(Yt.test, lty=2)

```

## Forecasts from ETS(A,A,N)



This produces a better short-term forecast that follows the recent trend found at the end of the training time series. As the forecast extends further into the future, it diverges significantly from the test data. (Forecasts will often perform better if any trends are dampened over time, asymptotically converging to a 0 slope. Such a dampening effect would improve the forecast for this model.)

As a famous quote puts, “It is difficult to make predictions, especially about the future.” Due to the often extreme uncertainty about the future, we tend not to put much stock in long term forecasting (unless the data is highly regular, e.g., we assume the Sun will continue to rise in the morning long into the future). Given this uncertainty and the *present bias* (we tend to discount far-off rewards), we usually seek a model that we expect to perform well on near-term forecasts. On this metric, we are inclined to prefer the double exponential smoothing model presented above.

## Problem 3

A data set of public transportation boardings in Denver from August 2000 through December 2005 are in the `boardings` object in the `TSA` package. These data are already logged.

```
library(TSA)
data(boardings)
log.boardings = boardings[,1]
print(log.boardings)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2000                                12.49114
## 2001 12.45759 12.51886 12.49804 12.52874 12.52780 12.52162 12.48012 12.52364
## 2002 12.50106 12.52229 12.49063 12.55226 12.54457 12.51336 12.46190 12.51414
## 2003 12.49483 12.53297 12.44026 12.50992 12.50498 12.44860 12.44798 12.48586
## 2004 12.47789 12.54076 12.54592 12.56986 12.57247 12.49189 12.47962 12.51481
## 2005 12.53937 12.58296 12.57383 12.59280 12.58038 12.52468 12.51634 12.57870
## 2006 12.60070 12.62888 12.61389
##           Sep      Oct      Nov      Dec
## 2000 12.58577 12.53457 12.52406 12.39809
## 2001 12.60110 12.58120 12.54482 12.46953
```

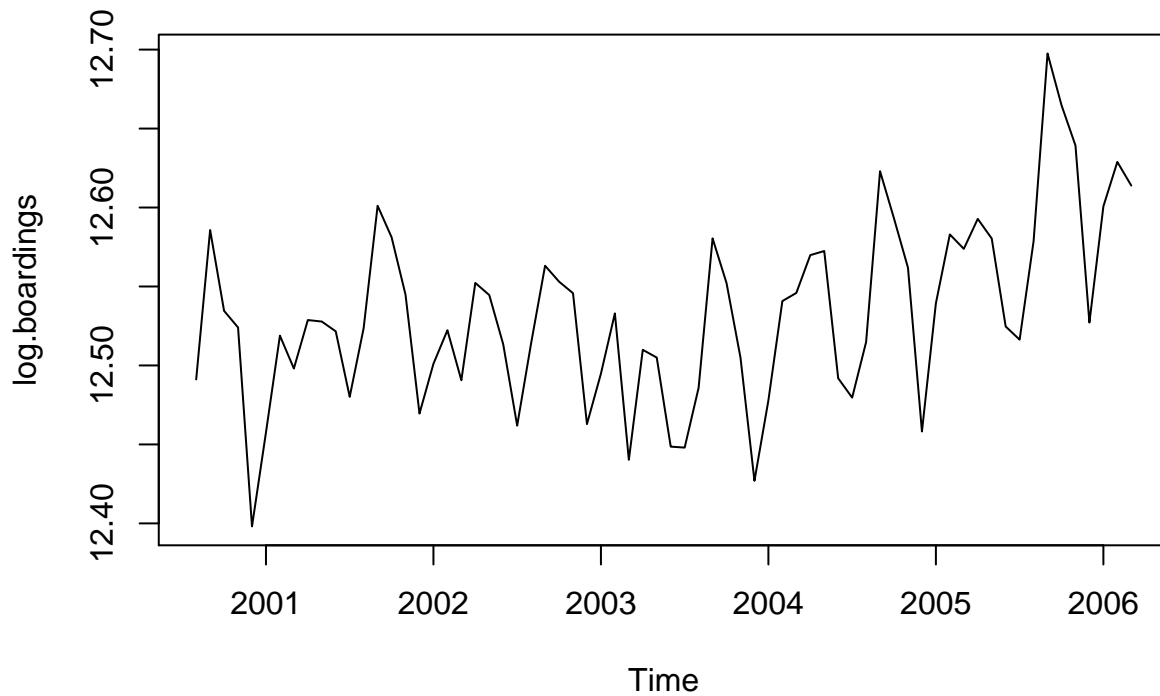
```
## 2002 12.56313 12.55305 12.54575 12.46286
## 2003 12.58046 12.55216 12.50488 12.42697
## 2004 12.62301 12.59301 12.56187 12.45818
## 2005 12.69752 12.66482 12.63930 12.52714
## 2006
```

## Part (a)

Give a time series plot of these data. Comment on the plot and any seasonality. Is it reasonable to use a stationary model for this time series?

The time series is plotted with:

```
plot(log.boardings)
```



First, there does appear to be a slight seasonable component, although admittedly it is not especially pronounced.

It does not appear to be stationary. We perform the Dickey-Fuller test to further convince ourselves:

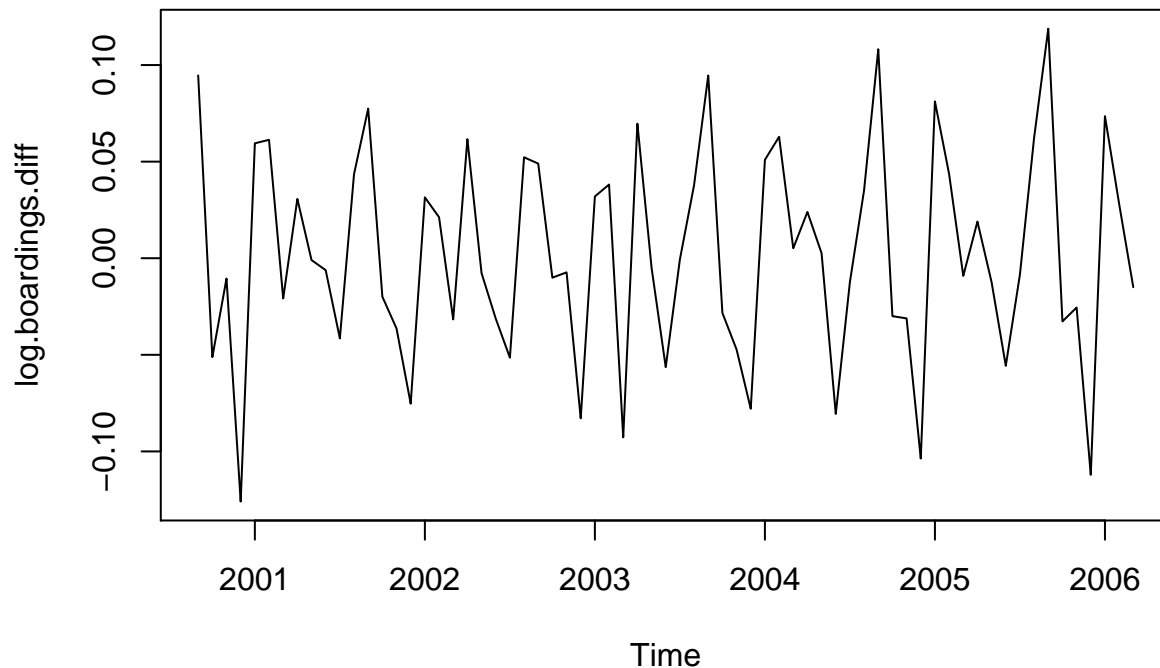
```
tseries::adf.test(log.boardings)
```

```
##
## Augmented Dickey-Fuller Test
##
## data: log.boardings
## Dickey-Fuller = -2.163, Lag order = 4, p-value = 0.5089
## alternative hypothesis: stationary
```

At this  $p$ -value, we do not reject the null hypothesis of non-stationary data. We take the difference and perform the Dickey-Fuller test on that transformation:

```
log.boardings.diff <- diff(log.boardings,1)
plot(log.boardings.diff)
```





```
tseries::adf.test(log.boardings.diff)
```

```
##
## Augmented Dickey-Fuller Test
##
## data: log.boardings.diff
## Dickey-Fuller = -6.939, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

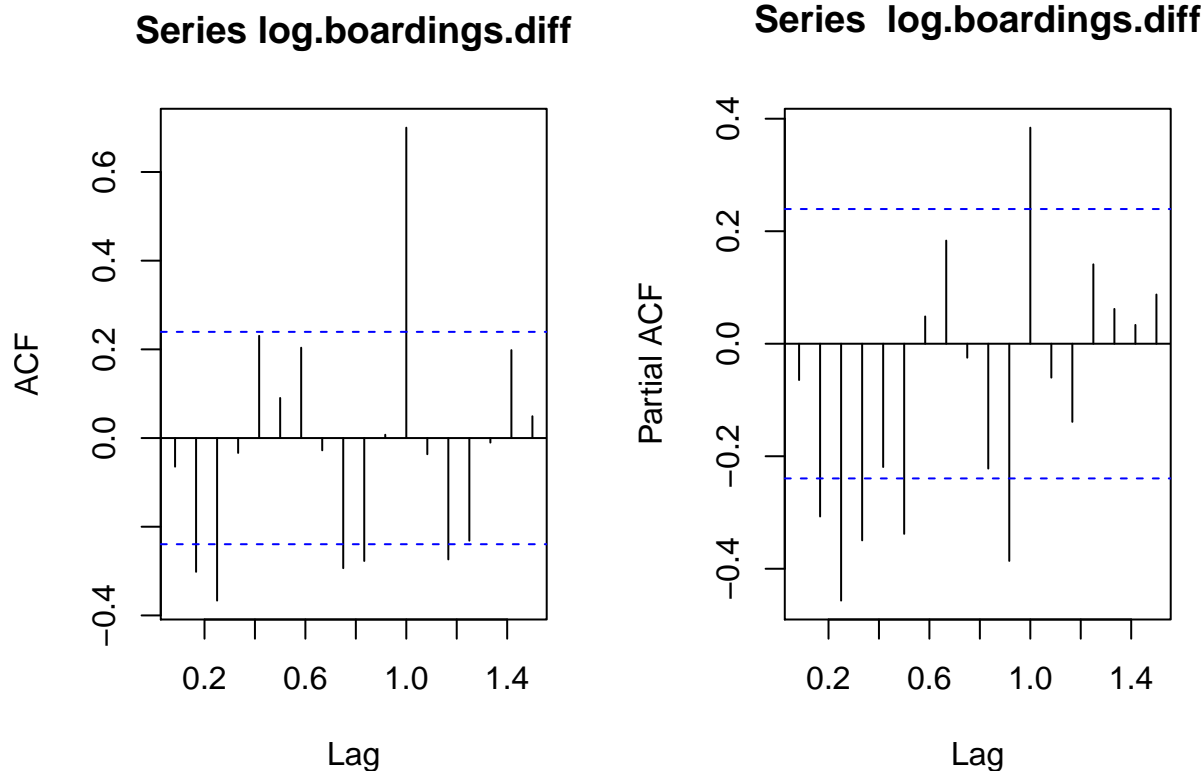
The plot seems to resemble a stationary process. Moreover, the  $p$ -value of the Dickey-Fuller hypothesis test is around 0.01, which we consider to be very strong evidence against the null hypothesis of non-stationary data.

## Part (b)

Plot the sample ACF of the series. Interpret what the plot tells you.

We plot the ACF with:

```
par(mfrow=c(1,2))
acf(log.boardings.diff)
pacf(log.boardings.diff)
```



This ACF diverges from the theoretical ACF of stationary data, but a 2-nd order difference does not improve the situation. This does not seem too bad.

### Part (c)

Choose a seasonal ARIMA model that fits the data. Write down your model and the corresponding AIC.

We look at the EACF with:

```
library(TSA)
eacf(log.boardings.diff, ar.max=4, ma.max=4)
```

```
## AR/MA
##   0 1 2 3 4
## 0 o x x o o
## 1 o o x o o
## 2 x x o o o
## 3 x o o o o
## 4 x x o o o
```

This EACF indicates  $\text{ARIMA}(p, d = 1, q)$  with  $(p, q) \in \{(0, 0), (1, 0), (1, 1)\}$ .

Given the seasonality of the data, we elect to fit a seasonal ARIMA model with parameters  $(p, d, q) \times (P, D, Q)$ , found by using the *auto-arima* function and the data indicates a periodicity of  $s = 12$  ( $S_t = S_{t+s}$ ).

```
fit <- auto.arima(log.boardings)
summary(fit)
```

```
## Series: log.boardings
## ARIMA(1,1,0)(0,1,1)[12]
##
```

```
## Coefficients:
##          ar1      sma1
##       -0.2919 -0.6089
## s.e.    0.1281   0.2280
##
## sigma^2 estimated as 0.0005886:  log likelihood=126.56
## AIC=-247.12   AICc=-246.65   BIC=-241.1
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.0002045332 0.02141851 0.0154136 0.001459963 0.1230202 0.4291575
##              ACF1
## Training set 0.04400124
print(fit$aic)

## [1] -247.1219
```

We see that the AIC is  $-247.1219$  and the best fit model is of type

$$\text{Seasonal-ARIMA}(p = 1, d = 1, q = 0) \times (P = 0, D = 1, Q = 1).$$

with coefficients given by

```
coef(fit)

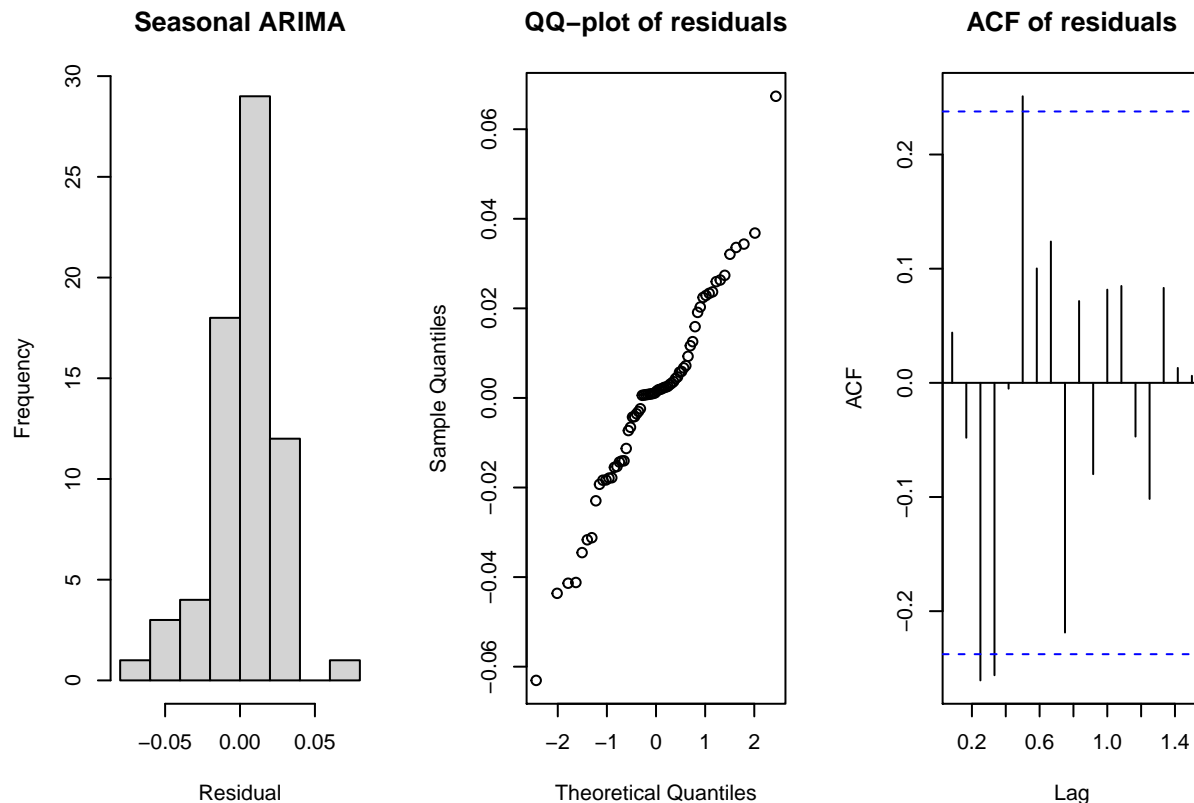
##          ar1      sma1
## -0.2918525 -0.6088785
```

## Part (d)

Check residuals for normality (histogram, qq-plot), for independence (ACF and the Ljung-Box test). Comment on your findings.

We plot a histogram, Q-Q plot, and ACF of the residuals with:

```
par(mfrow=c(1,3))
hist(fit$residual, xlab="Residual", main="Seasonal ARIMA")
qqnorm(fit$residual, main="QQ-plot of residuals")
acf(fit$residual, main="ACF of residuals")
```



We are not entirely comfortable with these plots representing idealized white noise, but real data sets are not typically going to have a data generating process that is truly distributed according to the ARIMA models, so some discrepancies are expected.

As the saying goes, all models are wrong, but hopefully some are useful. These plots satisfy us that we have sufficiently modeled the significant patterns in the data.

However, we conduct one last hypothesis test on the residuals with:

```
Box.test(fit$residuals,
        lag=K,
        type="Ljung-Box",
        fitdf=3)
```

```
##
## Box-Ljung test
##
## data: fit$residuals
## X-squared = 38.809, df = 27, p-value = 0.06594
```

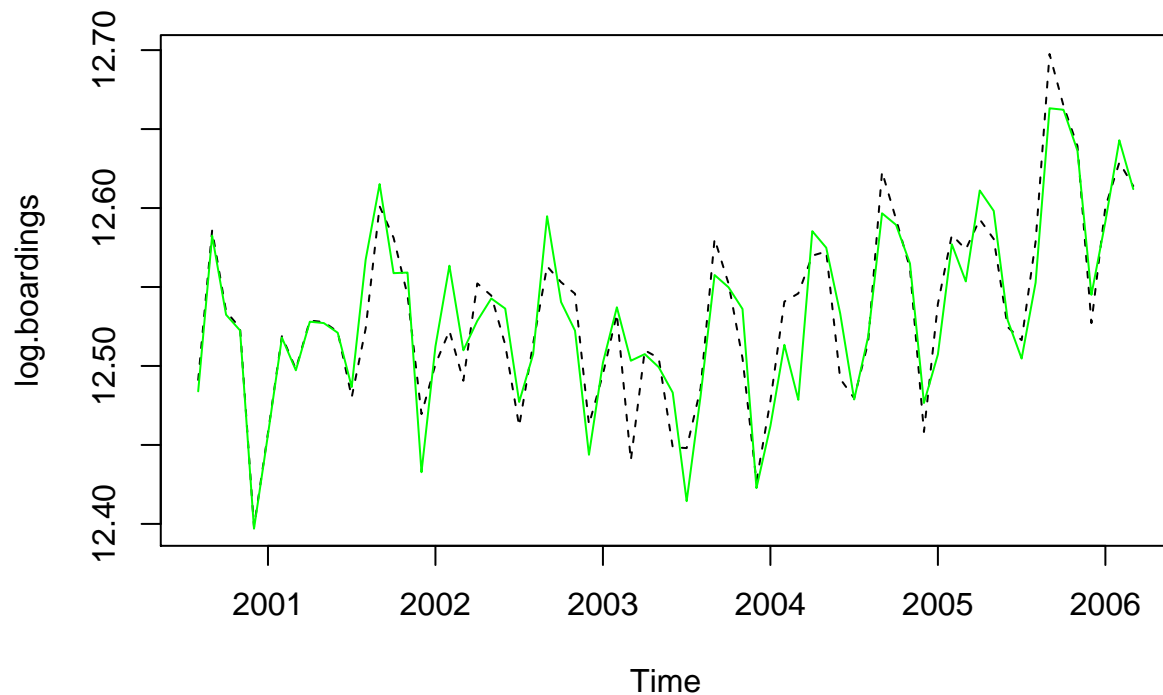
According to the Box-Ljung test, at a typical significance level  $\alpha = 0.05$ , the residuals are compatible with a white noise process.

### Additional plots

For fun, we do some forecasting with the model:

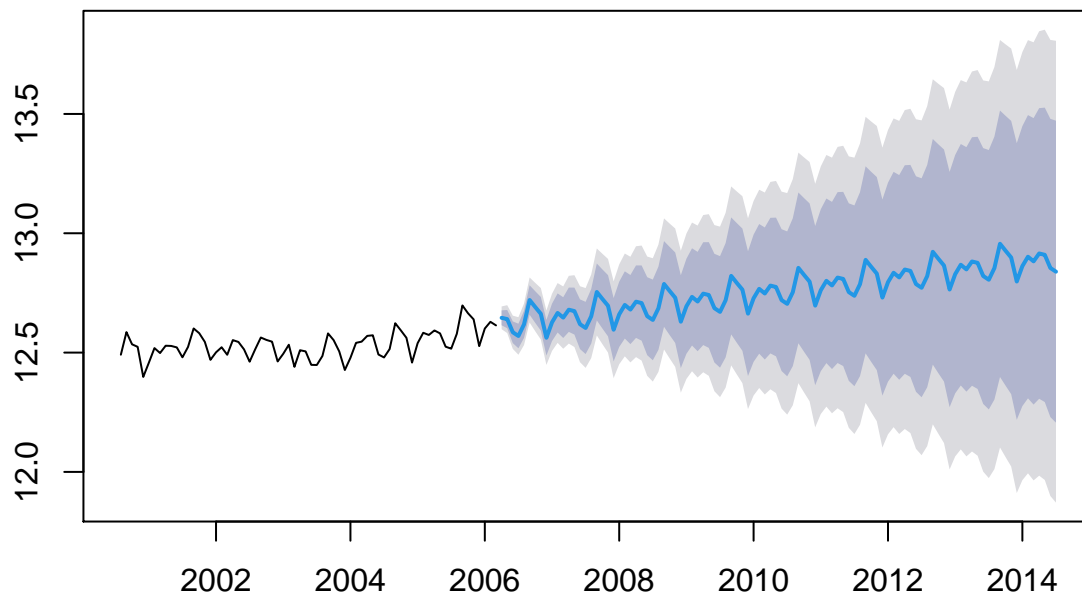
```
library(forecast)

plot(log.boardings,lty=2)
lines(fit$fitted,col="green")
```



```
fit.fc <- forecast(log.boardings,model=fit,h=100)
plot(fit.fc)
```

### Forecasts from ARIMA(1,1,0)(0,1,1)[12]



Note that this is with the log-transformed time series, so the forecast of the original time series is given by taking the exponential value of these forecasts.