W271 Lab 3

Tiffany Jaya, Joanna Huang, Robert Deng, Shan He

```
# add packages
library(forecast)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2018e.
## 1.0/zoneinfo/America/Los_Angeles'
```

```
library(knitr)
library(stats)
library(tseries)
library(xts)
# prevent source code from running off the page
opts_chunk$set(tidy.opts=list(width.cutoff=70), tidy=TRUE)
# remove all objects from current workspace
rm(list = ls())
# set seed number to reproduce results
set.seed(1)
# load data
raw.sales <- read.csv('./data/ECOMPCTNSA.csv', header=TRUE, sep=',')
```

Question 1: Forecasting using a SARIMA model

Since the emergence of the internet, more and more people are shopping at online retailers than brick-and-mortar stores. E-commerce is on the rise, and we would like to see what percentage of total retail sales e-commerce is accounted for in the fourth quarter of 2017. With data from the US Census Bureau ranging from 1999 to 2016, we were able to estimate it to be 10.20% using the seasonal autoregresive integrated moving average model (or SARIMA for short). While the number does not seem substantial compare to the perceived value of e-commerce, we have to remember that retail sales include motor vehicles, gas stations, and grocery stores where e-commerce has yet to play a major role in the field.

The SARIMA model that we use for the projected forecast is ARIMA$(0,1,0)(0,1,1)_4$.

Exploration Data Analysis

The first step once we obtained the dataset was to examine its structure.

```
# convert raw data into a time-series object
sales <- ts(raw.sales$ECOMPCTNSA, start = c(1999, 4), frequency = 4)
# hold out test data for verification in forecast
sales.train <- ts(sales[time(sales) < 2015], start = c(1999, 4), frequency = 4)  # 1999-2014
sales.test <- ts(sales[time(sales) >= 2015], start = c(2015, 1), frequency = 4)  # 2015-2016
# examine the structure
kable(summary(raw.sales))
```

DATE	ECOMPCTNSA
1999-10-01: 1	Min. :0.700
2000-01-01: 1	1st Qu.:2.000
2000-04-01: 1	Median :3.600
2000-07-01: 1	Mean :3.835
2000-10-01: 1	3rd Qu.:5.300
2001-01-01: 1	Max. :9.500
(Other) :63	NA

```
head(sales)
```

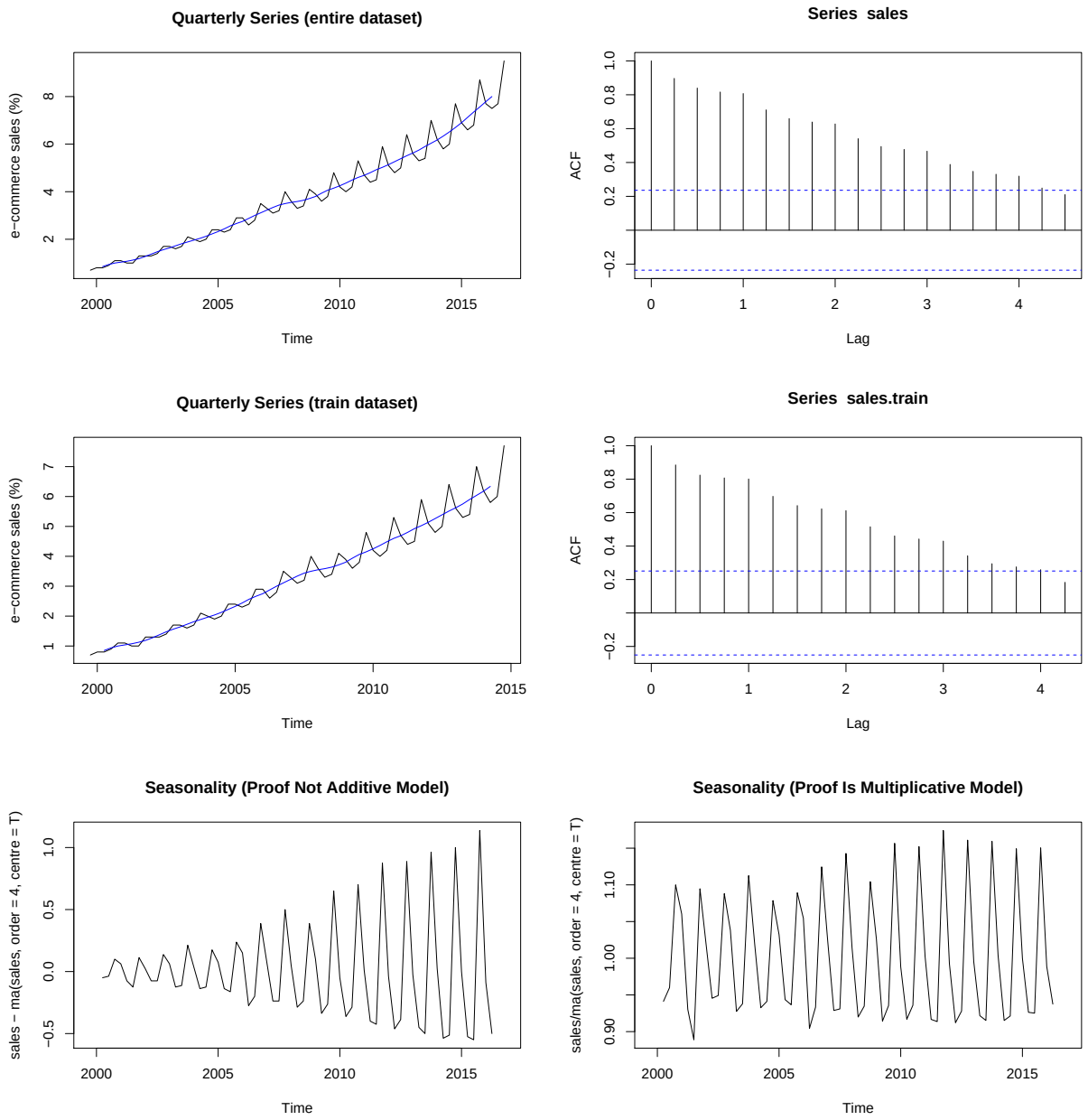
```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 1999                 0.7
## 2000  0.8  0.8  0.9  1.1
## 2001  1.1
```

```
tail(sales)
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2015            6.8  8.7
## 2016  7.7  7.5  7.7  9.5
```

We were able to determine that there was no missing value among the 69 observations with sales appearing to increase overtime from 0.7% in the 4th quarter of 1999 to 9.5% in the 4th quarter of 2016. To confirm, we plot the time series as well as its associating MA(4) model. If the data expressed seasonality every quarter, the MA(4) model smooths out the variances and acts as an annual trend with the seasonal effects within each quarter removed.

```
# using the entire dataset
plot(sales, ylab = "e-commerce sales (%)", main = "Quarterly Series (entire dataset)")
lines(ma(sales, order = 4, centre = T), col = "blue")
acf(sales)
# using the train dataset
plot(sales.train, ylab = "e-commerce sales (%)", main = "Quarterly Series (train dataset)")
lines(ma(sales.train, order = 4, centre = T), col = "blue")
acf(sales.train)
# remove trend to see seasonality
plot(sales - ma(sales, order = 4, centre = T), main = "Seasonality (Proof Not Additive Model)")
plot(sales/ma(sales, order = 4, centre = T), main = "Seasonality (Proof Is Multiplicative Model
```



Given the upward trend and increasing variance, the series is a multiplicative model that is non-stationary with quarterly seasonality. The autocorrelation function further substantiates the series's non-stationary because of its slow decay and r_1s that are large and positive (r_1 indicates how successive values of y relate to each other). For this reason, we will perform two operations. One, we will difference the series to stabilize the mean. And two, we will apply a Box-Cox transformation (logarithm and power transformation) to stabilize the variance. We verify if differencing was necessary by running the unit root test.

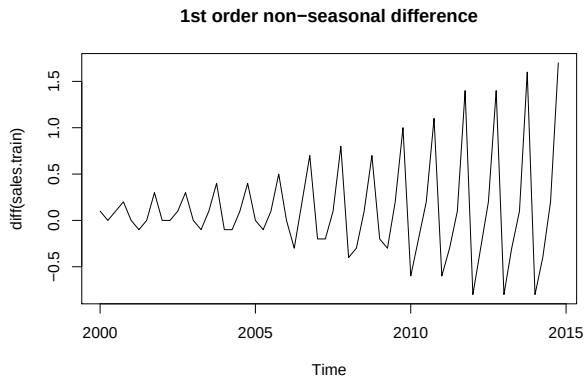
```
# unit root test
cbind(adf.test(sales.train, alternative = "stationary")$p.value, kpss.test(sales.train)$p.value
    ndiffs(sales.train), nsdiffs(sales.train))
```

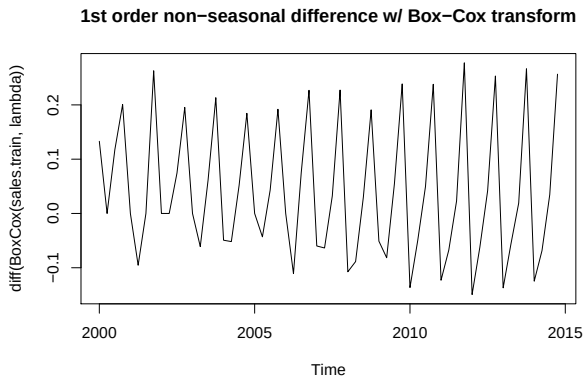
```
##      [,1] [,2] [,3] [,4]
## [1,] 0.99 0.01    1    1
```

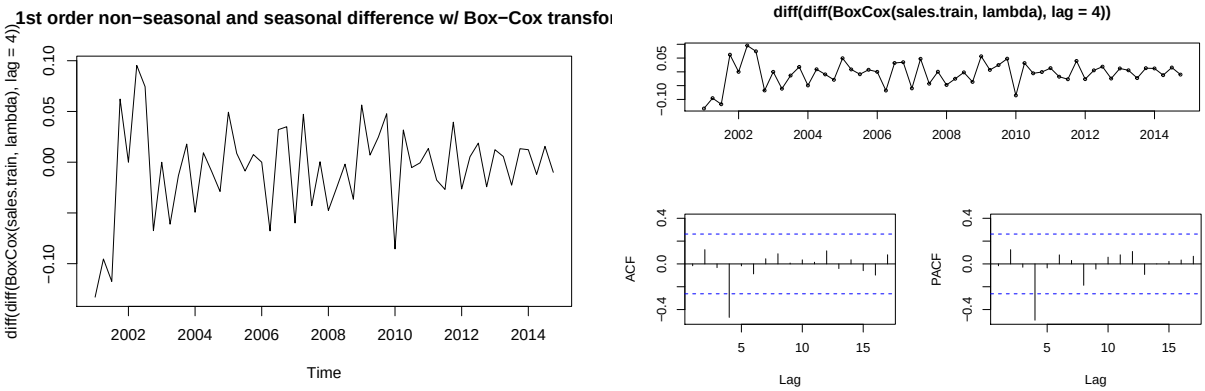
Large p-value in the Augmented Dickey-Fuller (ADF) test and small p-value in the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test confirm our intuition to difference the time series. As suggested by the non-seasonal (ndiffs) and seasonal (nsdiff) unit test, we will perform a non-seasonal difference to the data once and a seasonal difference once in order to make the series stationary in mean. The ADF and KPSS tests we run afterwards validate our differencing decision. We apply log and power transformation to the difference using the Box-Cox transformation by first finding the best lambda value that will give the optimal uniformity in the seasonal variation before administering the said transformation. With a lambda value of 0.01467236, the transformation is similar to a log transformation.

```
# find the best lambda for Box-Cox transformation lambda = 0.01467236,
# similar to a log-transformation
lambda <- BoxCox.lambda(sales.train)
# first-order non-seasonal differenced
plot(diff(sales.train), main = "1st order non-seasonal difference")
# ^ with Box-Cox transformed
plot(diff(BoxCox(sales.train, lambda)), main = "1st order non-seasonal difference w/ Box-Cox tr
# ^ with first-order seasonal differenced
plot(diff(diff(BoxCox(sales.train, lambda), lag = 4)), main = "1st order non-seasonal and seaso
# ^ and ACF and PACF
tsdisplay(diff(diff(BoxCox(sales.train, lambda), lag = 4)))
# unit root test on first-order differenced log-transformed series
cbind(adf.test(diff(diff(BoxCox(sales.train, lambda), lag = 4)), alternative = "stationary")$p
    kpss.test(diff(diff(BoxCox(sales.train, lambda), lag = 4)))$p.value)
```

```
##      [,1] [,2]
## [1,] 0.01  0.1
```







Modeling

Looking at the autocorrelation function (ACF) and partial autocorrelation function (PACF), we estimate the best-fitting model to be ARIMA$(0,1,0)(0,1,1)_4$. Our reasoning is as follows:

- Since we perform first order non-seasonal and seasonal difference, the non-seasonal difference d and seasonal difference D are equal to 1.
- With no significant spike in the non-seasonal lags of ACF and PACF plots, it suggests a possible non-seasonal AR(0) and MA(0) term.
- Since the seasonal correlograms in the ACF plot do not tail off to zero but those in the PACF plot does after lag 4, it signifies a potential moving average model. The only significant spike in ACF is at lag 4; all other autocorrelations are not significant. For this reason, it suggests a potential seasonal MA(1) term.

```
(base.m <- Arima(BoxCox(sales.train, lambda), order = c(0, 1, 0), seasonal = list(order = c(0,
    1, 1), 4)))
```

```
## Series: BoxCox(sales.train, lambda)
## ARIMA(0,1,0)(0,1,1)[4]
##
## Coefficients:
##          sma1
##       -0.5118
## s.e.   0.0973
##
## sigma^2 estimated as 0.00151:  log likelihood=102.31
## AIC=-200.62   AICc=-200.39   BIC=-196.57
```

By iterating through multiple parameters, we can confirm whether or not this model is the best-fitting model under the AICc criterion. We chose AICc instead of AIC since the number of observations, 69, is small and AICc can address AIC's potential problem of overfitting for small sample sizes.

```
best.manual.m <- base.m
for (p in 0:2) for (q in 0:2) for (d in 1:2) for (P in 0:2) for (Q in 0:2) for (D in 1:2) {
    m <- Arima(BoxCox(sales.train, lambda), order = c(p, d, q), seasonal = list(order = c(P,
        D, Q)))
```

```
    if (m$aicc < best.manual.m$aicc)
        best.manual.m <- m
}
best.manual.m
```

```
## Series: BoxCox(sales.train, lambda)
## ARIMA(0,1,0)(0,1,2)[4]
##
## Coefficients:
##          sma1    sma2
##       -0.7670  0.4052
## s.e.   0.1545  0.1342
##
## sigma^2 estimated as 0.001301:  log likelihood=106.17
## AIC=-206.35   AICc=-205.88   BIC=-200.27
```

What we have found is that ARIMA$(0,1,0)(0,1,2)_4$ has a lower AICc score than our estimated model we derived earlier ARIMA$(0,1,0)(0,1,1)_4$. In other words, ARIMA$(0,1,0)(0,1,2)_4$ may better explain the data but that fit might not be worth it at the cost of a loss in parsimony since we have to impose an additional seasonal MA lag into our estimated model. Similarly, ARIMA$(0,1,0)(0,1,1)_4$ may be more parsimonious, but it might not explain the data as well as ARIMA$(0,1,0)(0,1,2)_4$.

We compare our generated best-fitting model ARIMA$(0,1,0)(0,1,2)_4$ to the one generated by the auto.arima function and found it to be the same.

```
(best.auto.m <- auto.arima(BoxCox(sales.train, lambda), ic = "aicc", stepwise = FALSE,
    approximation = FALSE))
```

```
## Series: BoxCox(sales.train, lambda)
## ARIMA(0,1,0)(0,1,2)[4]
##
## Coefficients:
##          sma1    sma2
##       -0.7670  0.4052
## s.e.   0.1545  0.1342
##
## sigma^2 estimated as 0.001301:  log likelihood=106.17
## AIC=-206.35   AICc=-205.88   BIC=-200.27
```

For this reason, the two models we will compare moving forward are ARIMA$(0,1,0)(0,1,1)_4$, which is more parsimonous, and ARIMA$(0,1,0)(0,1,2)_4$, which has a better fit.

```
m1 <- base.m
m2 <- best.auto.m
```

Validating the models

Before we can forecast what percentage of total retail sales e-commerce sales will be in the future, we first need to validate that the residuals from the two models result in the following properties:

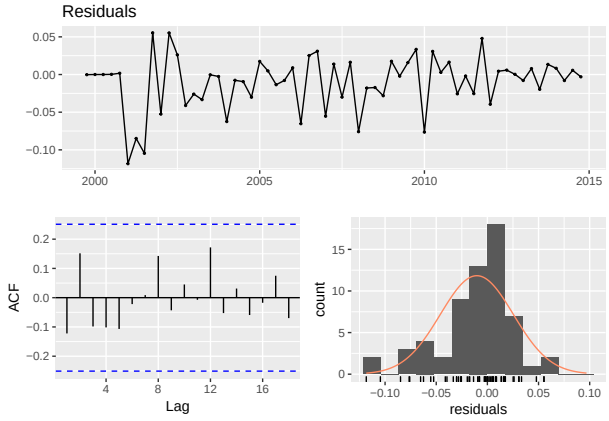
- uncorrelated, meaning there is no information left in the residuals that can be used in computing the forecast
- zero mean
- constant variance
- normally distributed

```
checkresiduals(m1$residuals)
h <- min(2 * 4, nrow(sales.train)/5)  # min(2m, T/5)
Box.test(m1$residuals, type = "Ljung-Box", lag = h)
```

```
##
##  Box-Ljung test
##
## data:  m1$residuals
## X-squared = 6.0888, df = 8, p-value = 0.6373
```

```
shapiro.test(m1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m1$residuals
## W = 0.93758, p-value = 0.003854
```



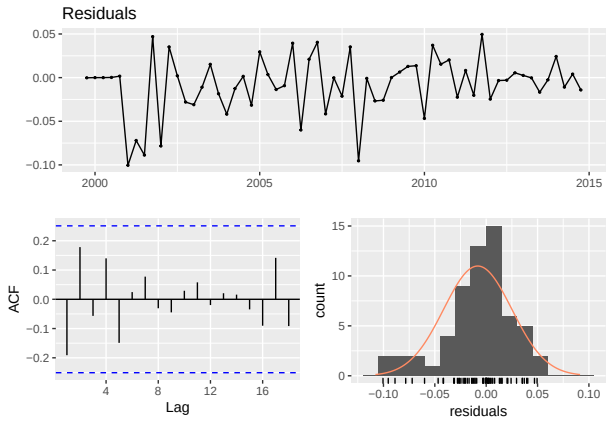

```
checkresiduals(m2$residuals)
h <- min(2 * 4, nrow(sales.train)/5)  # min(2m, T/5)
Box.test(m2$residuals, type = "Ljung-Box", lag = h)
```

```
##
##  Box-Ljung test
##
## data:  m2$residuals
## X-squared = 7.9999, df = 8, p-value = 0.4335
```

```
shapiro.test(m2$residuals)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  m2$residuals
## W = 0.93482, p-value = 0.002911
```



Looking at the ACF plots, all spikes of the two models are within the significant limits, meaning that the residuals are uncorrelated to one another. We then perform a test on a group of autocorrelations with the Box-Ljung test. With large p-values, the Box-Ljung test validates our assumption that the residuals are uncorrelated. The time plot of the residuals shows that the variation of the residuals stays approximately the same for both models, so we can treat the residual variance as constant. However, even with a mean close to zero, the histogram of all models suggests that it follows more of a negative skewed distribution than normal. The Shapiro-Wilk test confirms the non-normality of the distribution for the two models. What this signifies is that when we perform a prediction in the following section, its forecast will generally be quite good but prediction intervals computed assuming a normal distribution may be inaccurate.

Forecasting

Now that we have validated our models, it is time to extrapolate what percentage of e-sales commerce constitutes the total retail sales. First, we compare the forecasts of all the models to the hold out test data from 2015 till 2016 to see if their predictions are comparable to the actual. Then we plot out the forecasts. The blue represents the forecast and the orange represents the test data.

```
sales.test
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2015  6.9  6.6  6.8  8.7
## 2016  7.7  7.5  7.7  9.5
```

```
InvBoxCox(predict(m1, 3 * 4)$pred, lambda)
```

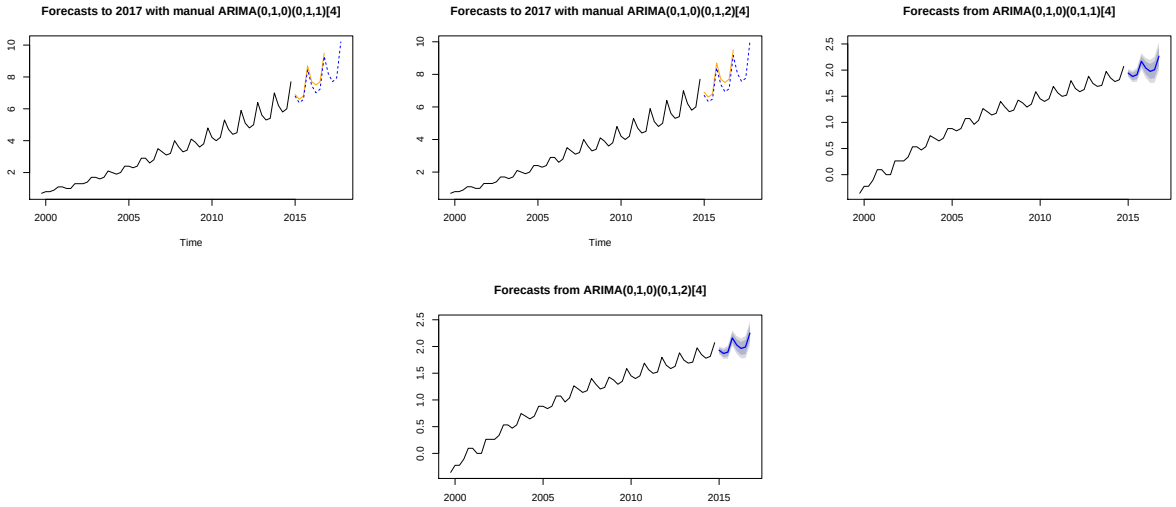
```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2015  6.793138  6.380801  6.582199  8.457164
## 2016  7.462415  7.010058  7.231008  9.287585
## 2017  8.196570  7.700373  7.942741 10.198236
```

```
InvBoxCox(predict(m2, 3 * 4)$pred, lambda)
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
```

```
## 2015 6.705666 6.323977 6.479338 8.392189
## 2016 7.379740 6.930514 7.111839 9.157311
## 2017 8.053882 7.564228 7.761877 9.991076
```

```
ts.plot(cbind(sales.train, sales.test, InvBoxCox(predict(m1, 4 * 3)$pred,
    lambda)), col = c("black", "orange", "blue"), lty = c(1, 1, 2), main = "Forecasts to 2017 w
ts.plot(cbind(sales.train, sales.test, InvBoxCox(predict(m2, 4 * 3)$pred,
    lambda)), col = c("black", "orange", "blue"), lty = c(1, 1, 2), main = "Forecasts to 2017 w
plot(forecast(m1))
plot(forecast(m2))
```



Both the forecasts as well as the graphs tell us that our parsimonous model ARIMA$(0,2,1)(0,2,1)_4$ predict much more closely to the test data than the autogenerated one ARIMA$(0,2,1)(0,2,2)_4$. We use this model to determine that e-commerce makes up approximately 10.20% of all retail sales by the fourth quarter of 2017.

Question 2: Learning how to use the xts library

If we could select one company to represent the e-commerce trend, Amazon is likely to be the first company that comes to mind.

1. Read AMAZ.csv and UMCSENT.csv into R as R DataFrames.

```
raw.amaz <- read.csv("./data/AMAZ.csv", header = TRUE, sep = ",")
raw.sent <- read.csv("./data/UMCSENT.csv", header = TRUE, sep = ",")
```

```
raw.sent
```

```
##          Index UMCSENT
## 1   1978-01-01    83.7
## 2   1978-02-01    84.3
## 3   1978-03-01    78.8
## 4   1978-04-01    81.6
## 5   1978-05-01    82.9
```

```
## 6   1978-06-01    80.0
## 7   1978-07-01    82.4
## 8   1978-08-01    78.4
## 9   1978-09-01    80.4
## 10  1978-10-01    79.3
## 11  1978-11-01    75.0
## 12  1978-12-01    66.1
## 13  1979-01-01    72.1
## 14  1979-02-01    73.9
## 15  1979-03-01    68.4
## 16  1979-04-01    66.0
## 17  1979-05-01    68.1
## 18  1979-06-01    65.8
## 19  1979-07-01    60.4
## 20  1979-08-01    64.5
## 21  1979-09-01    66.7
## 22  1979-10-01    62.1
## 23  1979-11-01    63.3
## 24  1979-12-01    61.0
## 25  1980-01-01    67.0
## 26  1980-02-01    66.9
## 27  1980-03-01    56.5
## 28  1980-04-01    52.7
## 29  1980-05-01    51.7
## 30  1980-06-01    58.7
## 31  1980-07-01    62.3
## 32  1980-08-01    67.3
## 33  1980-09-01    73.7
## 34  1980-10-01    75.0
## 35  1980-11-01    76.7
## 36  1980-12-01    64.5
## 37  1981-01-01    71.4
## 38  1981-02-01    66.9
## 39  1981-03-01    66.5
## 40  1981-04-01    72.4
## 41  1981-05-01    76.3
## 42  1981-06-01    73.1
## 43  1981-07-01    74.1
## 44  1981-08-01    77.2
## 45  1981-09-01    73.1
## 46  1981-10-01    70.3
## 47  1981-11-01    62.5
## 48  1981-12-01    64.3
## 49  1982-01-01    71.0
## 50  1982-02-01    66.5
## 51  1982-03-01    62.0
## 52  1982-04-01    65.5
## 53  1982-05-01    67.5
```

```
## 54  1982-06-01    65.7
## 55  1982-07-01    65.4
## 56  1982-08-01    65.4
## 57  1982-09-01    69.3
## 58  1982-10-01    73.4
## 59  1982-11-01    72.1
## 60  1982-12-01    71.9
## 61  1983-01-01    70.4
## 62  1983-02-01    74.6
## 63  1983-03-01    80.8
## 64  1983-04-01    89.1
## 65  1983-05-01    93.3
## 66  1983-06-01    92.2
## 67  1983-07-01    92.8
## 68  1983-08-01    90.9
## 69  1983-09-01    89.9
## 70  1983-10-01    89.3
## 71  1983-11-01    91.1
## 72  1983-12-01    94.2
## 73  1984-01-01   100.1
## 74  1984-02-01    97.4
## 75  1984-03-01   101.0
## 76  1984-04-01    96.1
## 77  1984-05-01    98.1
## 78  1984-06-01    95.5
## 79  1984-07-01    96.6
## 80  1984-08-01    99.1
## 81  1984-09-01   100.9
## 82  1984-10-01    96.3
## 83  1984-11-01    95.7
## 84  1984-12-01    92.9
## 85  1985-01-01    96.0
## 86  1985-02-01    93.7
## 87  1985-03-01    93.7
## 88  1985-04-01    94.6
## 89  1985-05-01    91.8
## 90  1985-06-01    96.5
## 91  1985-07-01    94.0
## 92  1985-08-01    92.4
## 93  1985-09-01    92.1
## 94  1985-10-01    88.4
## 95  1985-11-01    90.9
## 96  1985-12-01    93.9
## 97  1986-01-01    95.6
## 98  1986-02-01    95.9
## 99  1986-03-01    95.1
## 100 1986-04-01    96.2
## 101 1986-05-01    94.8
```

```
## 102 1986-06-01    99.3
## 103 1986-07-01    97.7
## 104 1986-08-01    94.9
## 105 1986-09-01    91.9
## 106 1986-10-01    95.6
## 107 1986-11-01    91.4
## 108 1986-12-01    89.1
## 109 1987-01-01    90.4
## 110 1987-02-01    90.2
## 111 1987-03-01    90.8
## 112 1987-04-01    92.8
## 113 1987-05-01    91.1
## 114 1987-06-01    91.5
## 115 1987-07-01    93.7
## 116 1987-08-01    94.4
## 117 1987-09-01    93.6
## 118 1987-10-01    89.3
## 119 1987-11-01    83.1
## 120 1987-12-01    86.8
## 121 1988-01-01    90.8
## 122 1988-02-01    91.6
## 123 1988-03-01    94.6
## 124 1988-04-01    91.2
## 125 1988-05-01    94.8
## 126 1988-06-01    94.7
## 127 1988-07-01    93.4
## 128 1988-08-01    97.4
## 129 1988-09-01    97.3
## 130 1988-10-01    94.1
## 131 1988-11-01    93.0
## 132 1988-12-01    91.9
## 133 1989-01-01    97.9
## 134 1989-02-01    95.4
## 135 1989-03-01    94.3
## 136 1989-04-01    91.5
## 137 1989-05-01    90.7
## 138 1989-06-01    90.6
## 139 1989-07-01    92.0
## 140 1989-08-01    89.6
## 141 1989-09-01    95.8
## 142 1989-10-01    93.9
## 143 1989-11-01    90.9
## 144 1989-12-01    90.5
## 145 1990-01-01    93.0
## 146 1990-02-01    89.5
## 147 1990-03-01    91.3
## 148 1990-04-01    93.9
## 149 1990-05-01    90.6
```

```
## 150 1990-06-01    88.3
## 151 1990-07-01    88.2
## 152 1990-08-01    76.4
## 153 1990-09-01    72.8
## 154 1990-10-01    63.9
## 155 1990-11-01    66.0
## 156 1990-12-01    65.5
## 157 1991-01-01    66.8
## 158 1991-02-01    70.4
## 159 1991-03-01    87.7
## 160 1991-04-01    81.8
## 161 1991-05-01    78.3
## 162 1991-06-01    82.1
## 163 1991-07-01    82.9
## 164 1991-08-01    82.0
## 165 1991-09-01    83.0
## 166 1991-10-01    78.3
## 167 1991-11-01    69.1
## 168 1991-12-01    68.2
## 169 1992-01-01    67.5
## 170 1992-02-01    68.8
## 171 1992-03-01    76.0
## 172 1992-04-01    77.2
## 173 1992-05-01    79.2
## 174 1992-06-01    80.4
## 175 1992-07-01    76.6
## 176 1992-08-01    76.1
## 177 1992-09-01    75.6
## 178 1992-10-01    73.3
## 179 1992-11-01    85.3
## 180 1992-12-01    91.0
## 181 1993-01-01    89.3
## 182 1993-02-01    86.6
## 183 1993-03-01    85.9
## 184 1993-04-01    85.6
## 185 1993-05-01    80.3
## 186 1993-06-01    81.5
## 187 1993-07-01    77.0
## 188 1993-08-01    77.3
## 189 1993-09-01    77.9
## 190 1993-10-01    82.7
## 191 1993-11-01    81.2
## 192 1993-12-01    88.2
## 193 1994-01-01    94.3
## 194 1994-02-01    93.2
## 195 1994-03-01    91.5
## 196 1994-04-01    92.6
## 197 1994-05-01    92.8
```

```
## 198 1994-06-01    91.2
## 199 1994-07-01    89.0
## 200 1994-08-01    91.7
## 201 1994-09-01    91.5
## 202 1994-10-01    92.7
## 203 1994-11-01    91.6
## 204 1994-12-01    95.1
## 205 1995-01-01    97.6
## 206 1995-02-01    95.1
## 207 1995-03-01    90.3
## 208 1995-04-01    92.5
## 209 1995-05-01    89.8
## 210 1995-06-01    92.7
## 211 1995-07-01    94.4
## 212 1995-08-01    96.2
## 213 1995-09-01    88.9
## 214 1995-10-01    90.2
## 215 1995-11-01    88.2
## 216 1995-12-01    91.0
## 217 1996-01-01    89.3
## 218 1996-02-01    88.5
## 219 1996-03-01    93.7
## 220 1996-04-01    92.7
## 221 1996-05-01    89.4
## 222 1996-06-01    92.4
## 223 1996-07-01    94.7
## 224 1996-08-01    95.3
## 225 1996-09-01    94.7
## 226 1996-10-01    96.5
## 227 1996-11-01    99.2
## 228 1996-12-01    96.9
## 229 1997-01-01    97.4
## 230 1997-02-01    99.7
## 231 1997-03-01   100.0
## 232 1997-04-01   101.4
## 233 1997-05-01   103.2
## 234 1997-06-01   104.5
## 235 1997-07-01   107.1
## 236 1997-08-01   104.4
## 237 1997-09-01   106.0
## 238 1997-10-01   105.6
## 239 1997-11-01   107.2
## 240 1997-12-01   102.1
## 241 1998-01-01   106.6
## 242 1998-02-01   110.4
## 243 1998-03-01   106.5
## 244 1998-04-01   108.7
## 245 1998-05-01   106.5
```

```
## 246 1998-06-01   105.6
## 247 1998-07-01   105.2
## 248 1998-08-01   104.4
## 249 1998-09-01   100.9
## 250 1998-10-01    97.4
## 251 1998-11-01   102.7
## 252 1998-12-01   100.5
## 253 1999-01-01   103.9
## 254 1999-02-01   108.1
## 255 1999-03-01   105.7
## 256 1999-04-01   104.6
## 257 1999-05-01   106.8
## 258 1999-06-01   107.3
## 259 1999-07-01   106.0
## 260 1999-08-01   104.5
## 261 1999-09-01   107.2
## 262 1999-10-01   103.2
## 263 1999-11-01   107.2
## 264 1999-12-01   105.4
## 265 2000-01-01   112.0
## 266 2000-02-01   111.3
## 267 2000-03-01   107.1
## 268 2000-04-01   109.2
## 269 2000-05-01   110.7
## 270 2000-06-01   106.4
## 271 2000-07-01   108.3
## 272 2000-08-01   107.3
## 273 2000-09-01   106.8
## 274 2000-10-01   105.8
## 275 2000-11-01   107.6
## 276 2000-12-01    98.4
## 277 2001-01-01    94.7
## 278 2001-02-01    90.6
## 279 2001-03-01    91.5
## 280 2001-04-01    88.4
## 281 2001-05-01    92.0
## 282 2001-06-01    92.6
## 283 2001-07-01    92.4
## 284 2001-08-01    91.5
## 285 2001-09-01    81.8
## 286 2001-10-01    82.7
## 287 2001-11-01    83.9
## 288 2001-12-01    88.8
## 289 2002-01-01    93.0
## 290 2002-02-01    90.7
## 291 2002-03-01    95.7
## 292 2002-04-01    93.0
## 293 2002-05-01    96.9
```

```
## 294 2002-06-01    92.4
## 295 2002-07-01    88.1
## 296 2002-08-01    87.6
## 297 2002-09-01    86.1
## 298 2002-10-01    80.6
## 299 2002-11-01    84.2
## 300 2002-12-01    86.7
## 301 2003-01-01    82.4
## 302 2003-02-01    79.9
## 303 2003-03-01    77.6
## 304 2003-04-01    86.0
## 305 2003-05-01    92.1
## 306 2003-06-01    89.7
## 307 2003-07-01    90.9
## 308 2003-08-01    89.3
## 309 2003-09-01    87.7
## 310 2003-10-01    89.6
## 311 2003-11-01    93.7
## 312 2003-12-01    92.6
## 313 2004-01-01   103.8
## 314 2004-02-01    94.4
## 315 2004-03-01    95.8
## 316 2004-04-01    94.2
## 317 2004-05-01    90.2
## 318 2004-06-01    95.6
## 319 2004-07-01    96.7
## 320 2004-08-01    95.9
## 321 2004-09-01    94.2
## 322 2004-10-01    91.7
## 323 2004-11-01    92.8
## 324 2004-12-01    97.1
## 325 2005-01-01    95.5
## 326 2005-02-01    94.1
## 327 2005-03-01    92.6
## 328 2005-04-01    87.7
## 329 2005-05-01    86.9
## 330 2005-06-01    96.0
## 331 2005-07-01    96.5
## 332 2005-08-01    89.1
## 333 2005-09-01    76.9
## 334 2005-10-01    74.2
## 335 2005-11-01    81.6
## 336 2005-12-01    91.5
## 337 2006-01-01    91.2
## 338 2006-02-01    86.7
## 339 2006-03-01    88.9
## 340 2006-04-01    87.4
## 341 2006-05-01    79.1
```

```
## 342 2006-06-01    84.9
## 343 2006-07-01    84.7
## 344 2006-08-01    82.0
## 345 2006-09-01    85.4
## 346 2006-10-01    93.6
## 347 2006-11-01    92.1
## 348 2006-12-01    91.7
## 349 2007-01-01    96.9
## 350 2007-02-01    91.3
## 351 2007-03-01    88.4
## 352 2007-04-01    87.1
## 353 2007-05-01    88.3
## 354 2007-06-01    85.3
## 355 2007-07-01    90.4
## 356 2007-08-01    83.4
## 357 2007-09-01    83.4
## 358 2007-10-01    80.9
## 359 2007-11-01    76.1
## 360 2007-12-01    75.5
## 361 2008-01-01    78.4
## 362 2008-02-01    70.8
## 363 2008-03-01    69.5
## 364 2008-04-01    62.6
## 365 2008-05-01    59.8
## 366 2008-06-01    56.4
## 367 2008-07-01    61.2
## 368 2008-08-01    63.0
## 369 2008-09-01    70.3
## 370 2008-10-01    57.6
## 371 2008-11-01    55.3
## 372 2008-12-01    60.1
## 373 2009-01-01    61.2
## 374 2009-02-01    56.3
## 375 2009-03-01    57.3
## 376 2009-04-01    65.1
## 377 2009-05-01    68.7
## 378 2009-06-01    70.8
## 379 2009-07-01    66.0
## 380 2009-08-01    65.7
## 381 2009-09-01    73.5
## 382 2009-10-01    70.6
## 383 2009-11-01    67.4
## 384 2009-12-01    72.5
## 385 2010-01-01    74.4
## 386 2010-02-01    73.6
## 387 2010-03-01    73.6
## 388 2010-04-01    72.2
## 389 2010-05-01    73.6
```

```
## 390 2010-06-01    76.0
## 391 2010-07-01    67.8
## 392 2010-08-01    68.9
## 393 2010-09-01    68.2
## 394 2010-10-01    67.7
## 395 2010-11-01    71.6
## 396 2010-12-01    74.5
## 397 2011-01-01    74.2
## 398 2011-02-01    77.5
## 399 2011-03-01    67.5
## 400 2011-04-01    69.8
## 401 2011-05-01    74.3
## 402 2011-06-01    71.5
## 403 2011-07-01    63.7
## 404 2011-08-01    55.8
## 405 2011-09-01    59.5
## 406 2011-10-01    60.8
## 407 2011-11-01    63.7
## 408 2011-12-01    69.9
## 409 2012-01-01    75.0
## 410 2012-02-01    75.3
## 411 2012-03-01    76.2
## 412 2012-04-01    76.4
## 413 2012-05-01    79.3
## 414 2012-06-01    73.2
## 415 2012-07-01    72.3
## 416 2012-08-01    74.3
## 417 2012-09-01    78.3
## 418 2012-10-01    82.6
## 419 2012-11-01    82.7
## 420 2012-12-01    72.9
## 421 2013-01-01    73.8
## 422 2013-02-01    77.6
## 423 2013-03-01    78.6
## 424 2013-04-01    76.4
## 425 2013-05-01    84.5
## 426 2013-06-01    84.1
## 427 2013-07-01    85.1
## 428 2013-08-01    82.1
## 429 2013-09-01    77.5
## 430 2013-10-01    73.2
## 431 2013-11-01    75.1
## 432 2013-12-01    82.5
## 433 2014-01-01    81.2
## 434 2014-02-01    81.6
## 435 2014-03-01    80.0
## 436 2014-04-01    84.1
## 437 2014-05-01    81.9
```

```
## 438 2014-06-01    82.5
## 439 2014-07-01    81.8
## 440 2014-08-01    82.5
## 441 2014-09-01    84.6
## 442 2014-10-01    86.9
## 443 2014-11-01    88.8
## 444 2014-12-01    93.6
## 445 2015-01-01    98.1
## 446 2015-02-01    95.4
## 447 2015-03-01    93.0
## 448 2015-04-01    95.9
## 449 2015-05-01    90.7
## 450 2015-06-01    96.1
## 451 2015-07-01    93.1
## 452 2015-08-01    91.9
## 453 2015-09-01    87.2
## 454 2015-10-01    90.0
## 455 2015-11-01    91.3
## 456 2015-12-01    92.6
## 457 2016-01-01    92.0
## 458 2016-02-01    91.7
## 459 2016-03-01    91.0
## 460 2016-04-01    89.0
## 461 2016-05-01    94.7
## 462 2016-06-01    93.5
## 463 2016-07-01    90.0
## 464 2016-08-01    89.8
## 465 2016-09-01    91.2
## 466 2016-10-01    87.2
## 467 2016-11-01    93.8
## 468 2016-12-01    98.2
## 469 2017-01-01    98.5
## 470 2017-02-01    96.3
## 471 2017-03-01    96.9
## 472 2017-04-01    97.0
## 473 2017-05-01    97.1
## 474 2017-06-01    95.1
## 475 2017-07-01    93.4
## 476 2017-08-01    96.8
## 477 2017-09-01    95.1
```

2. Convert them to xts objects.

```
# set local timezone
Sys.setenv(TZ = "America/Los_Angeles")
# assume stock data is collected in EST
amaz <- xts(raw.amaz[, -1], order.by = as.POSIXct(raw.amaz[, 1], tz = "EST"))
# assume sentiment data is collected in EST
sent <- xts(raw.sent[, -1], order.by = as.POSIXct(raw.sent[, 1], tz = "EST"))
```

```
ats <- amaz
uts <- sent
```

3. Merge the two set of series together, preserving all of the observations in both set of series.
    a. Fill all of the missing values of the UMCSENT series with -9999.

```
UMCSENT <- merge(ats, uts, join = "outer", fill = -9999)
# head(UMCSENT) tail(UMCSENT) describe(UMCSENT)
```

b. Then create a new series, named UMCSENT02, from the original UMCSENT series and replace all

```
UMCSENT02 <- UMCSENT
UMCSENT02[UMCSENT02 == -9999] <- NA
head(UMCSENT02)
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            AMAZ.Open AMAZ.High AMAZ.Low AMAZ.Close AMAZ.Volume  uts
## 1978-01-01        NA        NA       NA         NA          NA 83.7
## 1978-02-01        NA        NA       NA         NA          NA 84.3
## 1978-03-01        NA        NA       NA         NA          NA 78.8
## 1978-04-01        NA        NA       NA         NA          NA 81.6
## 1978-05-01        NA        NA       NA         NA          NA 82.9
## 1978-06-01        NA        NA       NA         NA          NA 80.0
```

c. Then create a new series, named UMCSENT03, and replace the NAs with the last observation.

```
UMCSENT03 <- UMCSENT02
UMCSENT03 <- na.locf(UMCSENT03, na.rm = TRUE, fromLast = TRUE)
# head(UMCSENT03) describe(UMCSENT03)
UMCSENT03["2007-01-03"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            AMAZ.Open AMAZ.High AMAZ.Low AMAZ.Close AMAZ.Volume  uts
## 2007-01-03        20        20       16         16         650 91.3
```

```
UMCSENT02["2007-01-03"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            AMAZ.Open AMAZ.High AMAZ.Low AMAZ.Close AMAZ.Volume uts
## 2007-01-03        20        20       16         16         650  NA
```

d. Then create a new series, named UMCSENT04, and replace the NAs using linear interpolation.

```
UMCSENT04 <- UMCSENT02
UMCSENT04 <- na.approx(UMCSENT04, maxgap = 31)
head(UMCSENT04)
```

```
## Warning: timezone of object (EST) is different than current timezone
```

```
## (America/Los_Angeles).
```

```
##            AMAZ.Open AMAZ.High AMAZ.Low AMAZ.Close AMAZ.Volume  uts
## 1978-01-01        NA        NA       NA          NA          NA 83.7
## 1978-02-01        NA        NA       NA          NA          NA 84.3
## 1978-03-01        NA        NA       NA          NA          NA 78.8
## 1978-04-01        NA        NA       NA          NA          NA 81.6
## 1978-05-01        NA        NA       NA          NA          NA 82.9
## 1978-06-01        NA        NA       NA          NA          NA 80.0
```

```
# Check if values for uts were replaced
UMCSENT04["2007-01-03"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            AMAZ.Open AMAZ.High AMAZ.Low AMAZ.Close AMAZ.Volume      uts
## 2007-01-03        20        20       16          16         650 96.53871
```

```
UMCSENT02["2007-01-03"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            AMAZ.Open AMAZ.High AMAZ.Low AMAZ.Close AMAZ.Volume uts
## 2007-01-03        20        20       16          16         650  NA
```

e. Print out some observations to ensure that your merge as well as the missing value imputatio

```
# TODO!!!!!!!  check Jan 3 for both raw datasets
ats["2007-01-03"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            AMAZ.Open AMAZ.High AMAZ.Low AMAZ.Close AMAZ.Volume
## 2007-01-03        20        20       16          16         650
```

```
uts["2007-01-03"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##      [,1]
```

```
# not in uts, check two closesest uts values
uts["2007-01-01"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            [,1]
## 2007-01-01 96.9
```

```
uts["2007-02-01"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            [,1]
## 2007-02-01 91.3
```

```
# check merge
UMCSENT04["2007-01-03"]
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            AMAZ.Open AMAZ.High AMAZ.Low AMAZ.Close AMAZ.Volume      uts
## 2007-01-03        20        20       16          16         650 96.53871
```

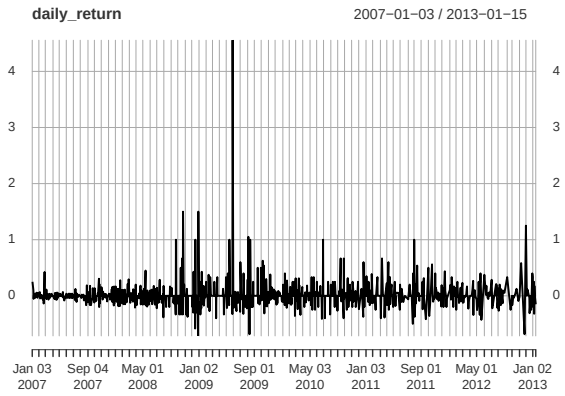
```
# check interpolation
coredata(UMCSENT04["2007-01-03", 6]) == coredata(uts["2007-01-01"]) - (coredata(uts["2007-01-0
    coredata(uts["2007-02-01"])) * 2/31
```

```
##       uts
## [1,] TRUE
```

4. Calculate the daily return of the Amazon closing price (AMAZ.close), where daily return is defined as $(x(t) - x(t-1))/x(t-1)$. Plot the daily return series.

```
daily_return <- (ats[, 4] - lag(ats[, 4], k = 1))/(lag(ats[, 4], k = 1))

plot(daily_return)
```



Looking at Amazon's closing price between January 2017 to January 2013, we see much volatility. A few trends that may be gauged is that January to July generally sees lower daily returns and peaks of 1+ returns happens post-July (which may be related to product launch timelines). These trends hold with the exception of January-July 2009 which had unprecedented returns that may be due to the acquisition of Zappos during that time.

5. Create a 20-day and a 50-day rolling mean series from the AMAZ.close series.

```
head(cbind(ats[, 4], rollapply(ats[, 4], 20, FUN = mean, na.rm = TRUE)),
    30)
```

```
## Warning: timezone of object (EST) is different than current timezone
```

```
## (America/Los_Angeles).
```

```
##            AMAZ.Close AMAZ.Close.1
## 2007-01-03       16.0           NA
## 2007-01-04       20.0           NA
## 2007-01-08       22.0           NA
## 2007-01-09       20.8           NA
## 2007-01-10       20.8           NA
## 2007-01-11       21.6           NA
## 2007-01-12       22.0           NA
## 2007-01-16       21.2           NA
## 2007-01-17       21.6           NA
## 2007-01-22       22.8           NA
## 2007-01-23       22.8           NA
## 2007-01-26       22.0           NA
## 2007-01-29       23.2           NA
## 2007-01-31       24.0           NA
## 2007-02-01       24.0           NA
## 2007-02-02       24.0           NA
## 2007-02-05       25.6           NA
## 2007-02-06       24.4           NA
## 2007-02-09       23.6           NA
## 2007-02-12       23.2        22.28
## 2007-02-13       23.6        22.66
## 2007-02-14       23.6        22.84
## 2007-02-15       23.6        22.92
## 2007-02-16       22.4        23.00
## 2007-02-20       20.8        23.00
## 2007-02-21       20.4        22.94
## 2007-02-22       17.6        22.72
## 2007-02-23       16.0        22.46
## 2007-02-26       22.8        22.52
## 2007-02-27       22.0        22.48
```

```
head(cbind(ats[, 4], rollapply(ats[, 4], 50, FUN = mean, na.rm = TRUE)),
    60)
```

```
## Warning: timezone of object (EST) is different than current timezone
## (America/Los_Angeles).
```

```
##            AMAZ.Close AMAZ.Close.1
## 2007-01-03      16.00           NA
## 2007-01-04      20.00           NA
## 2007-01-08      22.00           NA
## 2007-01-09      20.80           NA
## 2007-01-10      20.80           NA
## 2007-01-11      21.60           NA
## 2007-01-12      22.00           NA
## 2007-01-16      21.20           NA
```

```
## 2007-01-17      21.60          NA
## 2007-01-22      22.80          NA
## 2007-01-23      22.80          NA
## 2007-01-26      22.00          NA
## 2007-01-29      23.20          NA
## 2007-01-31      24.00          NA
## 2007-02-01      24.00          NA
## 2007-02-02      24.00          NA
## 2007-02-05      25.60          NA
## 2007-02-06      24.40          NA
## 2007-02-09      23.60          NA
## 2007-02-12      23.20          NA
## 2007-02-13      23.60          NA
## 2007-02-14      23.60          NA
## 2007-02-15      23.60          NA
## 2007-02-16      22.40          NA
## 2007-02-20      20.80          NA
## 2007-02-21      20.40          NA
## 2007-02-22      17.60          NA
## 2007-02-23      16.00          NA
## 2007-02-26      22.80          NA
## 2007-02-27      22.00          NA
## 2007-02-28      22.00          NA
## 2007-03-01      22.00          NA
## 2007-03-02      22.80          NA
## 2007-03-05      21.60          NA
## 2007-03-06      24.00          NA
## 2007-03-07      22.80          NA
## 2007-03-09      22.80          NA
## 2007-03-12      23.60          NA
## 2007-03-15      22.00          NA
## 2007-03-19      22.00          NA
## 2007-03-20      22.80          NA
## 2007-03-22      22.00          NA
## 2007-03-29      22.80          NA
## 2007-03-30      22.80          NA
## 2007-04-04      22.80          NA
## 2007-04-05      22.40          NA
## 2007-04-09      21.60          NA
## 2007-04-10      20.40          NA
## 2007-04-11      20.00          NA
## 2007-04-12      21.00     22.0520
## 2007-04-13      21.60     22.1640
## 2007-04-16      20.20     22.1680
## 2007-04-17      21.04     22.1488
## 2007-04-18      21.60     22.1648
## 2007-04-19      22.80     22.2048
## 2007-04-20      21.20     22.1968
```

```
## 2007-04-24      22.40      22.2048
## 2007-04-26      21.60      22.2128
## 2007-04-27      21.60      22.2128
## 2007-04-30      21.60      22.1888
```

1. ECOMPCTNSA: E-commerce retail sales as a percent of total sales

https://fred.stlouisfed.org/series/ECOMPCTNSA