

Final Project

Joan Zhang

Sunday, November 15, 2015

Objective : make a prediction for the direction of DJIA

ets() function Automatically chooses a model by default using the AIC, AICc or BIC. Can handle any combination of trend, seasonality and damping

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 3.2.5
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.2.5
```

```
## Loading required package: zoo
```

```
##  
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':  
##  
##   as.Date, as.Date.numeric
```

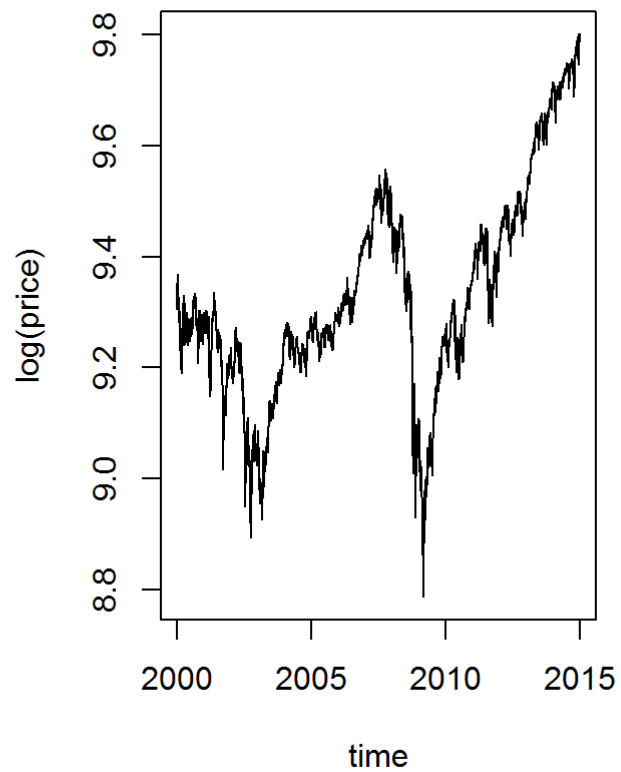
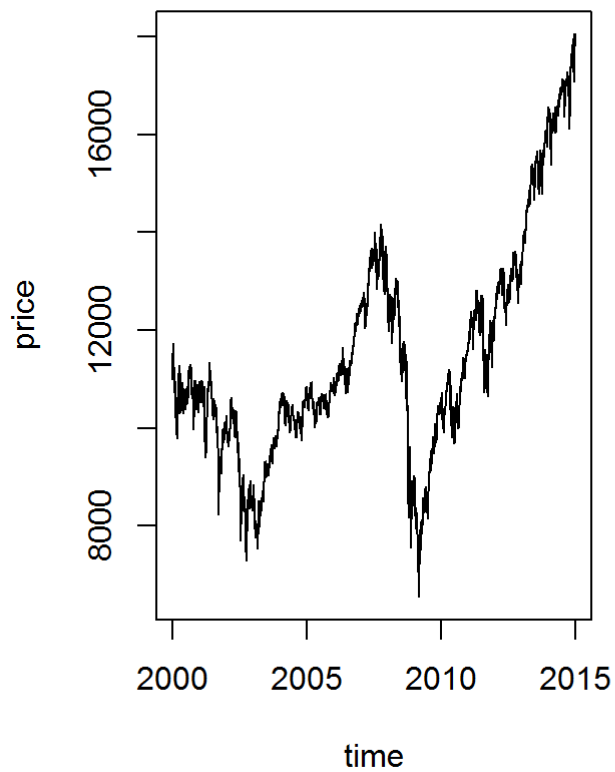
```
## Loading required package: timeDate
```

```
## This is forecast 7.1
```

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 3.2.5
```

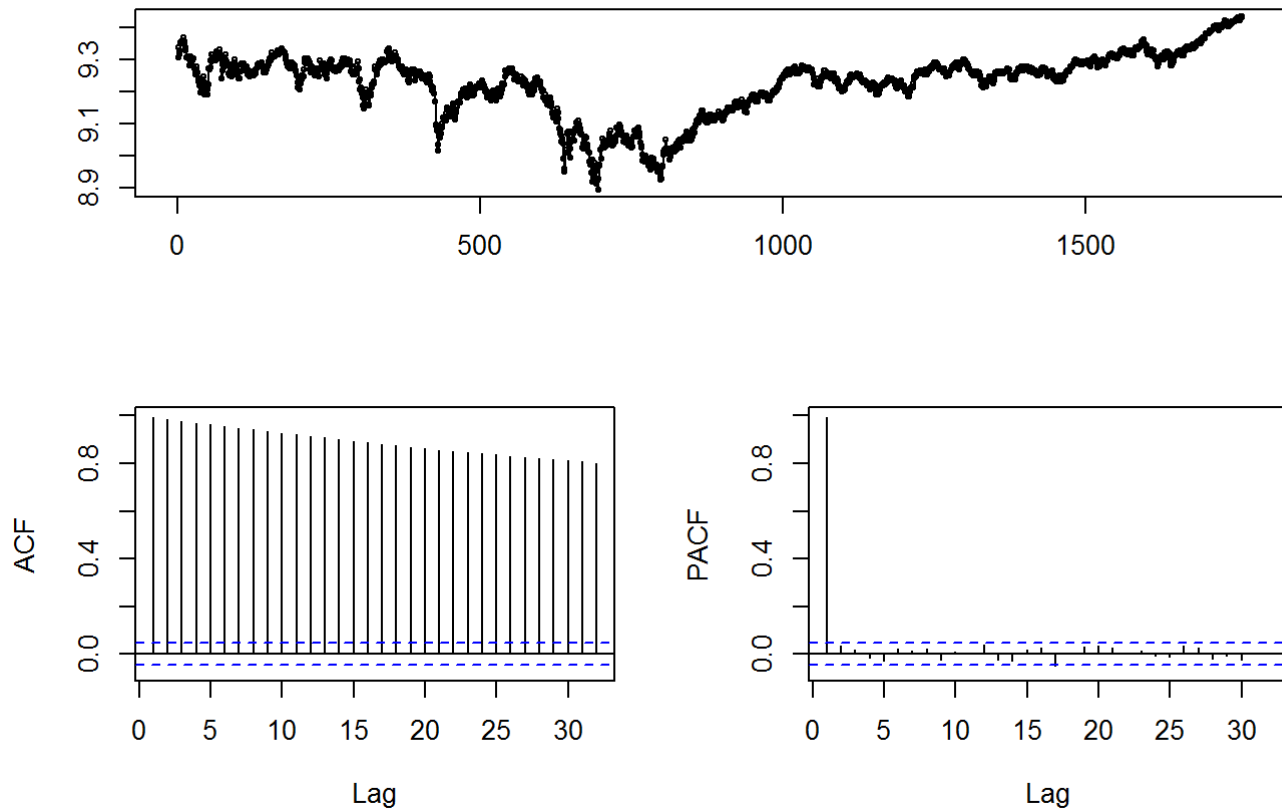
```
DJIA <- read.csv(file="C:/Users/jzhangn/Documents/DJA.csv", header=TRUE, sep=",")  
time <- as.Date(DJIA$Date, "%m/%d/%Y")  
price <- DJIA$DJIA  
par(mfrow=c(1,2))  
plot(time, price, type='l'); plot(time, log(price), type='l')
```



```
DJIA <- as.ts(DJIA)
```

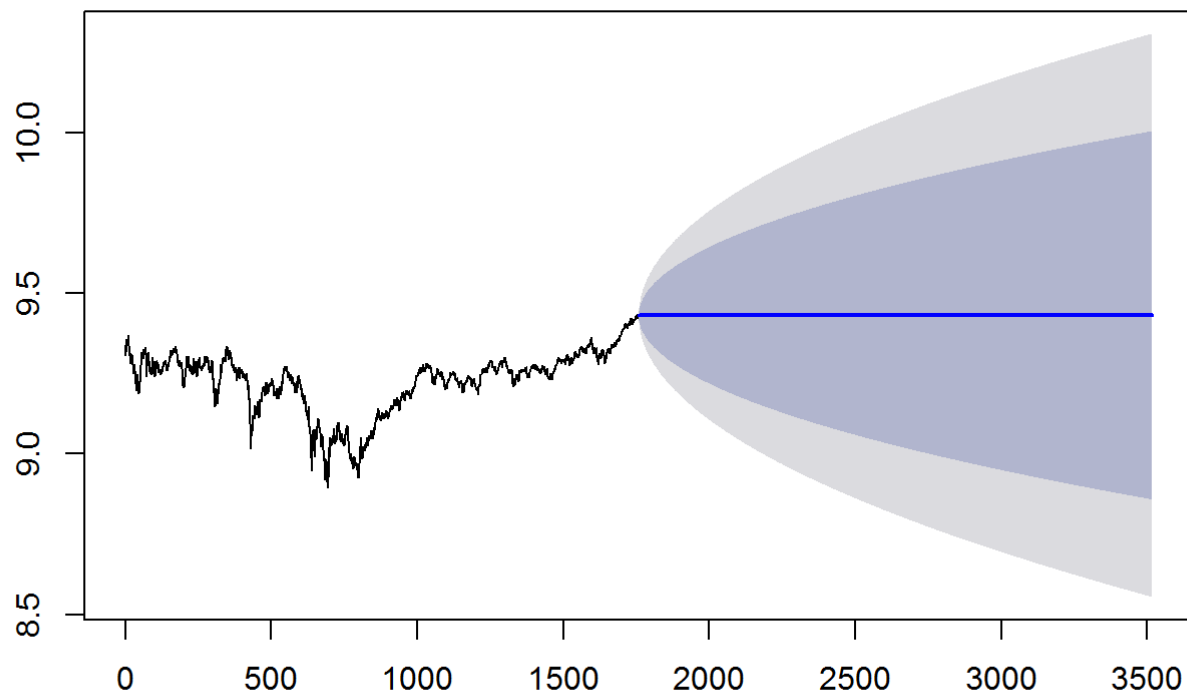
1. Exponential moving average

```
#data 2000-2006  
DJIA <- as.ts(DJIA)  
djia.data.train1 <- window(DJIA,start=1,end=1759)  
djia.data.train1 <- log(djia.data.train1[,2])  
#djia.data.train1 <- diff(djia.data.train1)  
#adf.test(djia.data.train1)  
tsdisplay(djia.data.train1)
```

djia.data.train1

```
fit1 <- ets(djia.data.train1, model="ZZZ")  
fcast1 <- forecast(fit1, h=251*7) # forecast 251*7. Last 251 is year 2013,  
plot(fcast1)
```

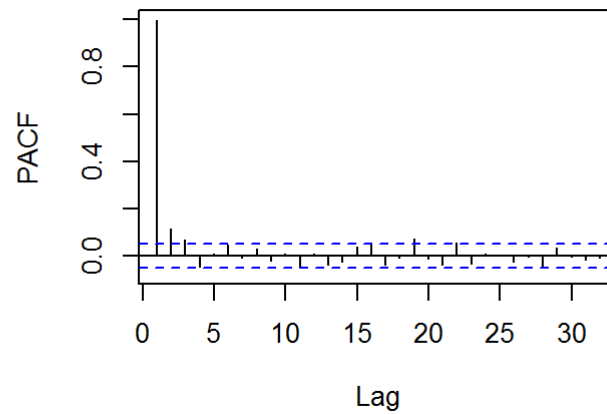
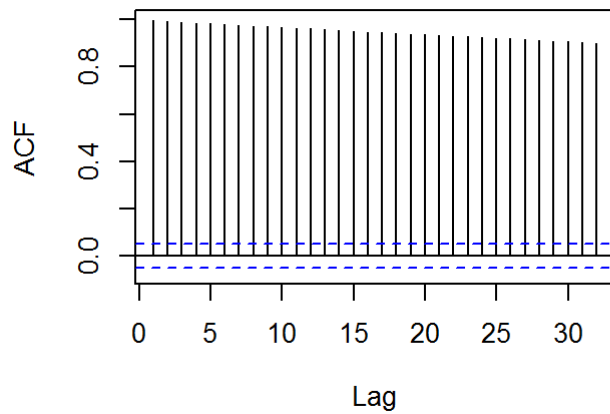
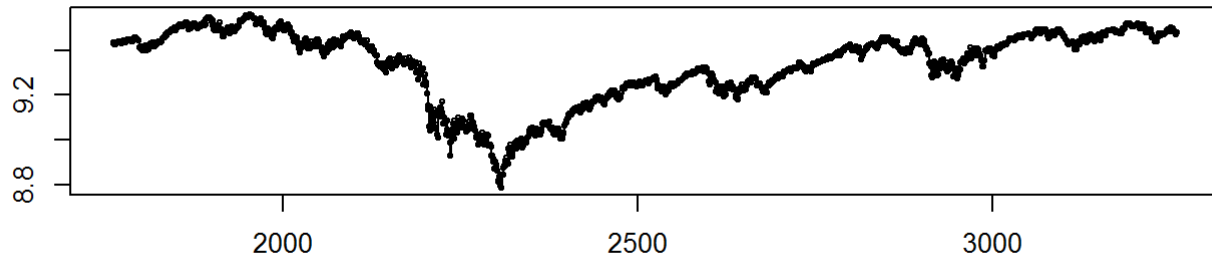
Forecasts from ETS(A,N,N)



```
fcast1$mean.2013 <- fcast1$mean[1507:1757]
# date from 2007- 2012
djia.data.train2 <- window(DJIA, start=1760, end=3261)
djia.data.train2 <- log(djia.data.train2[,2])
#djia.data.train2 <- diff(djia.data.train2)
adf.test(djia.data.train2)
```

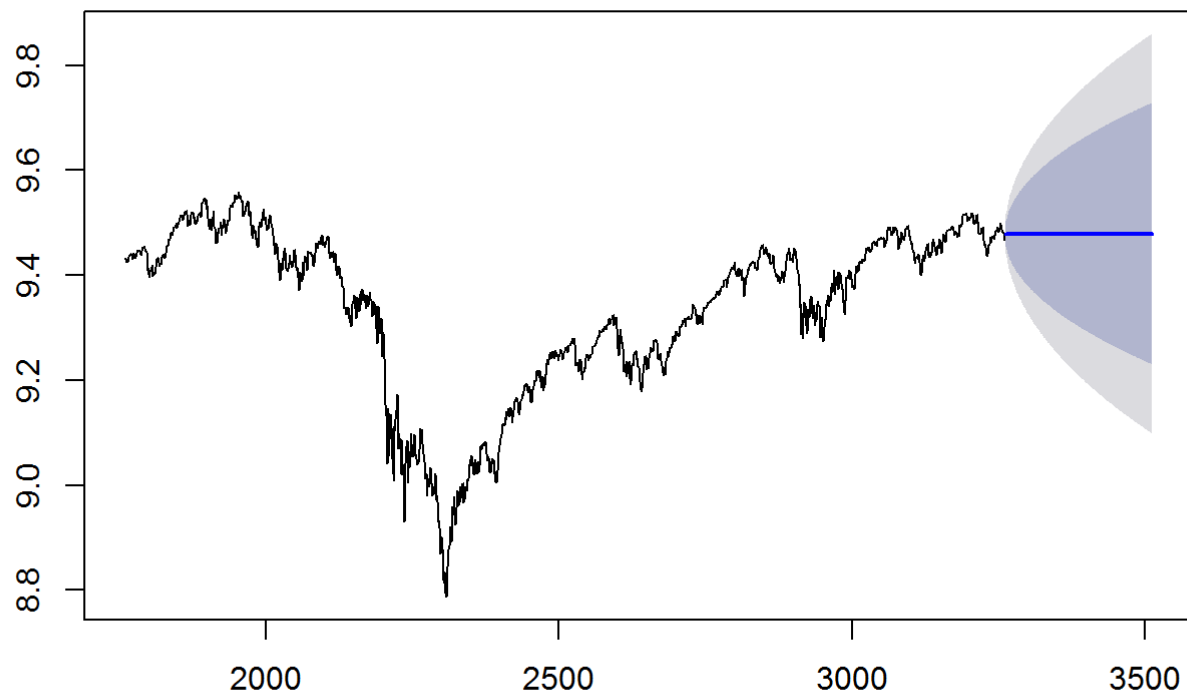
```
##
## Augmented Dickey-Fuller Test
##
## data: djia.data.train2
## Dickey-Fuller = -1.4574, Lag order = 11, p-value = 0.808
## alternative hypothesis: stationary
```

```
tsdisplay(djia.data.train2)
```

djia.data.train2

```
fit2 <- ets(djia.data.train2, model = "ZZZ")  
fcast2 <- forecast(fit2, h=251)  
plot(fcast2)
```

Forecasts from ETS(A,N,N)



```
fcast2$mean
```

```
## Time Series:
## Start = 3262
## End = 3512
## Frequency = 1
## [1] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [9] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [17] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [25] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [33] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [41] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [49] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [57] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [65] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [73] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [81] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [89] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [97] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [105] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [113] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [121] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [129] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [137] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [145] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [153] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [161] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [169] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [177] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [185] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [193] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [201] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [209] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [217] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [225] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [233] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [241] 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917 9.47917
## [249] 9.47917 9.47917 9.47917
```

```
# take mean of two forecast means to get mean
f <- function (x, y) {
  df <- data.frame(x, y)
  ave <- rowMeans(df)
  ave
}
ave.mean <- f(fcast1$mean.2013, fcast2$mean)

# data 2013
djia.data.valid <- window(DJIA, start=3262, end=3512)
djia.data.valid <- log(djia.data.valid[,2])
exp.s.mv.mse <- mse(djia.data.valid, ave.mean)
exp.s.mv.mse
```

```
## [1] 0.02801211
```

2. HoltWinters Method

```
# fit data 2000-2006
fit1 <- HoltWinters(djia.data.train1, beta=TRUE, gamma=FALSE, l.start=9.337634)
fcast1 <- forecast.HoltWinters(fit1, h=251*7)
fcast1$mean.2013 <- fcast1$mean[1507:1757]

#fit data 2007-2012
fit2 <- HoltWinters(djia.data.train2, beta=FALSE, gamma=FALSE, l.start=9.431443)
fcast2 <- forecast.HoltWinters(fit2, h=251)
ave.mean <- f(fcast1$mean.2013, fcast2$mean)
holtwinters.mse <- mse(djia.data.valid, ave.mean)
holtwinters.mse
```

```
## [1] 0.7114913
```

3. ARIMA

```
fit1 <- auto.arima(djia.data.train1, seasonal=FALSE, max.order=10, stepwise=FALSE,
approximation=FALSE) # auto.arima suggest ARIMA(2,0,2)
fcast1 <- forecast(fit1, h=251*7)
fcast1$mean.2013 <- fcast1$mean[1507:1757]

fit2 <- auto.arima(djia.data.train2, seasonal=FALSE, max.order=10, stepwise=FALSE,
approximation=FALSE) #auto.arima suggest ARIMA(1,0,5)
fcast2 <- forecast(fit2, h=251)
ave.mean <- f(fcast1$mean.2013, fcast2$mean)
arima.mse <- mse(djia.data.valid, ave.mean)
arima.mse
```

```
## [1] 0.02796217
```

Compare three models

```
cbind(exp.s.mv.mse, holtwinters.mse, arima.mse)
```

```
##      exp.s.mv.mse holtwinters.mse arima.mse
## [1,] 0.02801211      0.7114913 0.02796217
```

looks like ARIMA fits better with slightly lower MSE

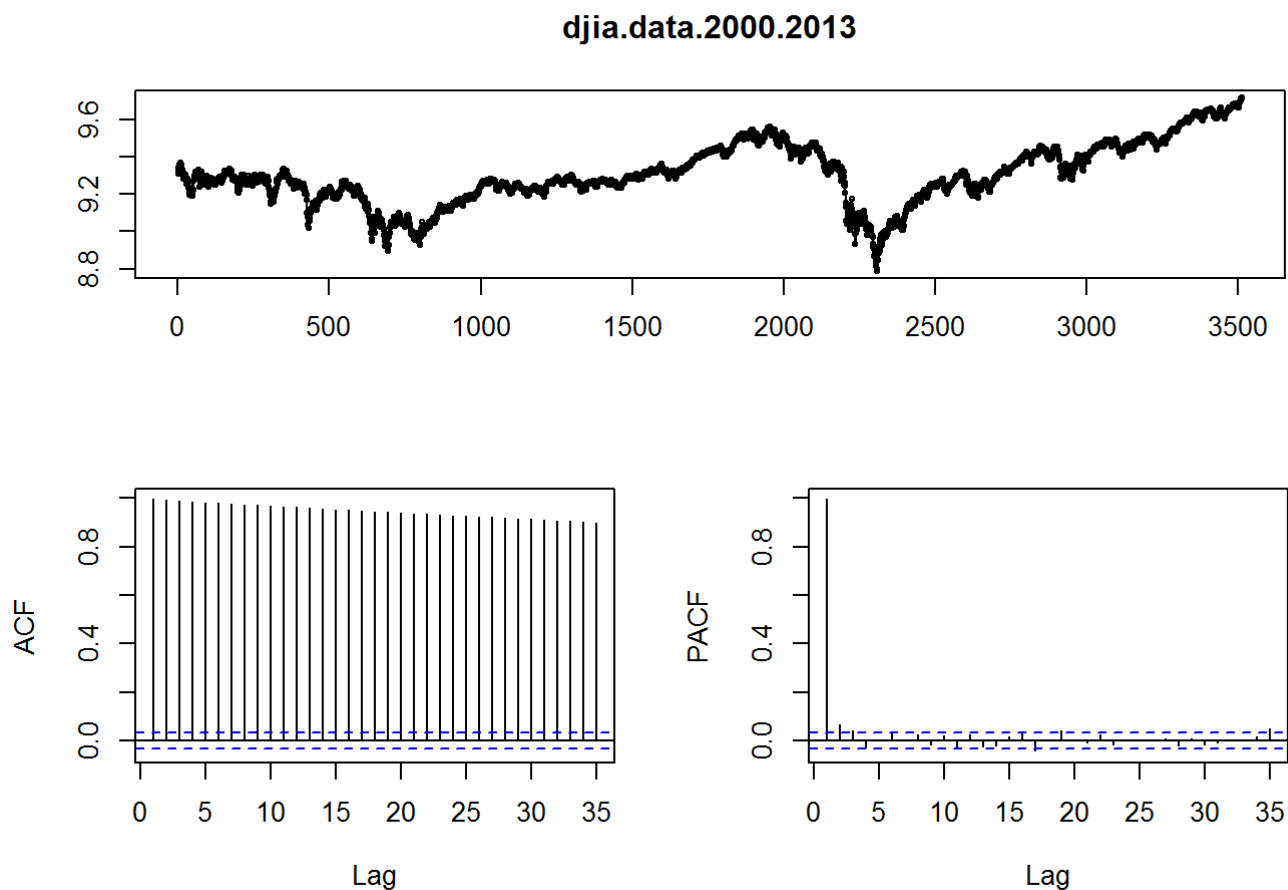
Use ARIMA method to predicate 2014

```
djia.data.2000.2013 <- window(DJIA, start =1, end = 3512) # data 2000-2013
djia.data.2000.2013 <- log(djia.data.2000.2013[,2])
#djia.data.2000.2013 <- diff(djia.data.2000.2013)
adf.test(djia.data.2000.2013)
```



```
##
## Augmented Dickey-Fuller Test
##
## data: djia.data.2000.2013
## Dickey-Fuller = -2.0307, Lag order = 15, p-value = 0.5653
## alternative hypothesis: stationary
```

```
tsdisplay(djia.data.2000.2013)
```



```
fit.arima <- auto.arima(djia.data.2000.2013, seasonal=FALSE, max.order=10, stepwise=FALSE, approximation=FALSE)
fit.arima
```

```
## Series: djia.data.2000.2013
## ARIMA(4,1,3)
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##          ar1          ar2          ar3          ar4          ma1          ma2          ma3
##      0.0243 -0.3778  0.7713  0.0424 -0.1079  0.3537 -0.7935
## s.e.      NaN          NaN          NaN          NaN          NaN          NaN          NaN
##
## sigma^2 estimated as 0.00015: log likelihood=10479.26
## AIC=-20942.51   AICc=-20942.47   BIC=-20893.2
```

```
fcast.arima <- forecast(fit.arima, h=251)
djia.data.test <- window(DJIA, start=3513, end=3762) # year 2014 data
djia.data.test.2014 <- log(djia.data.test[,2])
djia.data.test.2014.mse <- mse(djia.data.test.2014, fcast.arima$mean)
djia.data.test.2014.mse
```

```
## [1] 0.00127234
```

```
djia.data.test.2014.mae <- mae(djia.data.test.2014, fcast.arima$mean)
djia.data.test.2014.mae
```

```
## [1] 0.02743191
```

The MSE for 2014 is: 0.00127234

4. Fit intervention model Oct-06-2008, DJA drops from 10325.38 to 9955.5 pre-intervention series (2000-Jan-1 to 008-Oct-06)

```
library(tseries)
library(forecast)
library(TSA)
```

```
## Warning: package 'TSA' was built under R version 3.2.5
```

```
## Loading required package: leaps
```

```
## Loading required package: locfit
```

```
## Warning: package 'locfit' was built under R version 3.2.5
```

```
## locfit 1.5-9.1    2013-03-22
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
##  
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:forecast':  
##  
##     getResponse
```

```
## This is mgcv 1.8-9. For overview type 'help("mgcv-package")'.
```

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

```
## The following objects are masked from 'package:forecast':  
##  
##     fitted.Arima, plot.Arima
```

```
## The following objects are masked from 'package:timeDate':  
##  
##     kurtosis, skewness
```

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

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

```
djia.data.preinterv <- window(DJIA, start=1,end= 2202) # before 6, Oct 2008  
auto.arima(log(djia.data.preinterv[,2]), seasonal=FALSE, trace=TRUE, stepwise=FALSE, approximat  
ion=FALSE) # return (2,1,2)
```

```
##
## ARIMA(0,1,0) : -13449.84
## ARIMA(0,1,0) with drift : -13447.87
## ARIMA(0,1,1) : -13460.11
## ARIMA(0,1,1) with drift : -13458.14
## ARIMA(0,1,2) : -13461.13
## ARIMA(0,1,2) with drift : -13459.16
## ARIMA(0,1,3) : -13460.06
## ARIMA(0,1,3) with drift : -13458.08
## ARIMA(0,1,4) : -13458.51
## ARIMA(0,1,4) with drift : -13456.53
## ARIMA(0,1,5) : -13459.43
## ARIMA(0,1,5) with drift : -13457.45
## ARIMA(1,1,0) : -13459.08
## ARIMA(1,1,0) with drift : -13457.11
## ARIMA(1,1,1) : -13459.96
## ARIMA(1,1,1) with drift : -13457.98
## ARIMA(1,1,2) : -13459.52
## ARIMA(1,1,2) with drift : -13457.54
## ARIMA(1,1,3) : -13458.13
## ARIMA(1,1,3) with drift : -13456.17
## ARIMA(1,1,4) : -13457.12
## ARIMA(1,1,4) with drift : -13460.43
## ARIMA(2,1,0) : -13461.72
## ARIMA(2,1,0) with drift : -13459.75
## ARIMA(2,1,1) : -13459.76
## ARIMA(2,1,1) with drift : -13457.93
## ARIMA(2,1,2) : -13468.37
## ARIMA(2,1,2) with drift : -13466.38
## ARIMA(2,1,3) : -13468.04
## ARIMA(2,1,3) with drift : -13466.06
## ARIMA(3,1,0) : -13460.12
## ARIMA(3,1,0) with drift : -13458.15
## ARIMA(3,1,1) : -13457.73
## ARIMA(3,1,1) with drift : -13456.32
## ARIMA(3,1,2) : -13467.95
## ARIMA(3,1,2) with drift : -13465.98
## ARIMA(4,1,0) : -13459.24
## ARIMA(4,1,0) with drift : -13457.26
## ARIMA(4,1,1) : -13457.6
## ARIMA(4,1,1) with drift : -13455.67
## ARIMA(5,1,0) : -13458.53
## ARIMA(5,1,0) with drift : -13456.55
```

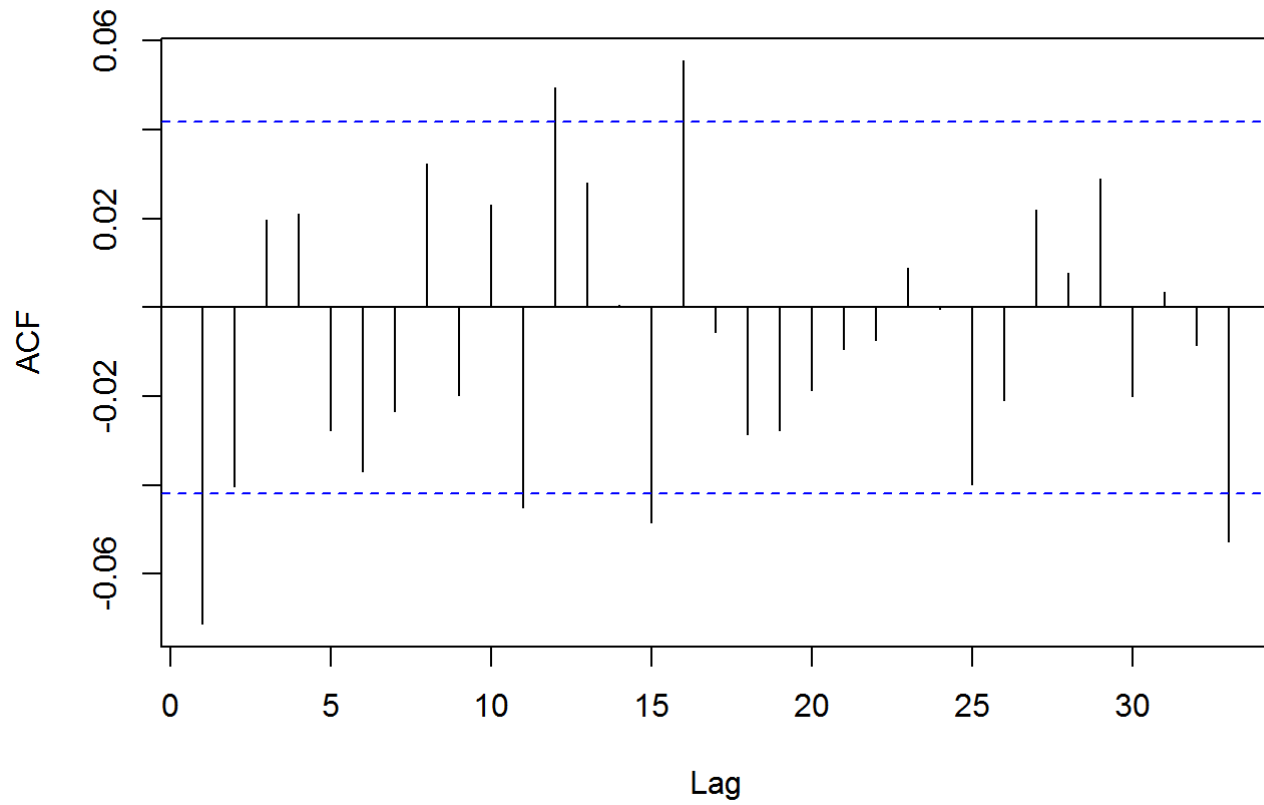
```
## Series: log(djia.data.preinterv[, 2])
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1      ma2
##      0.3387 -0.8615 -0.3942  0.8583
## s.e.  0.0590  0.0612  0.0570  0.0642
##
## sigma^2 estimated as 0.0001285:  log likelihood=6739.2
## AIC=-13468.39  AICc=-13468.37  BIC=-13439.91
```

```
djia.data.preinterv <- diff(log(djia.data.preinterv[,2]))
adf.test(djia.data.preinterv) # p-value = 0.01 reject H0, stationary
```

```
## Warning in adf.test(djia.data.preinterv): p-value smaller than printed p-
## value
```

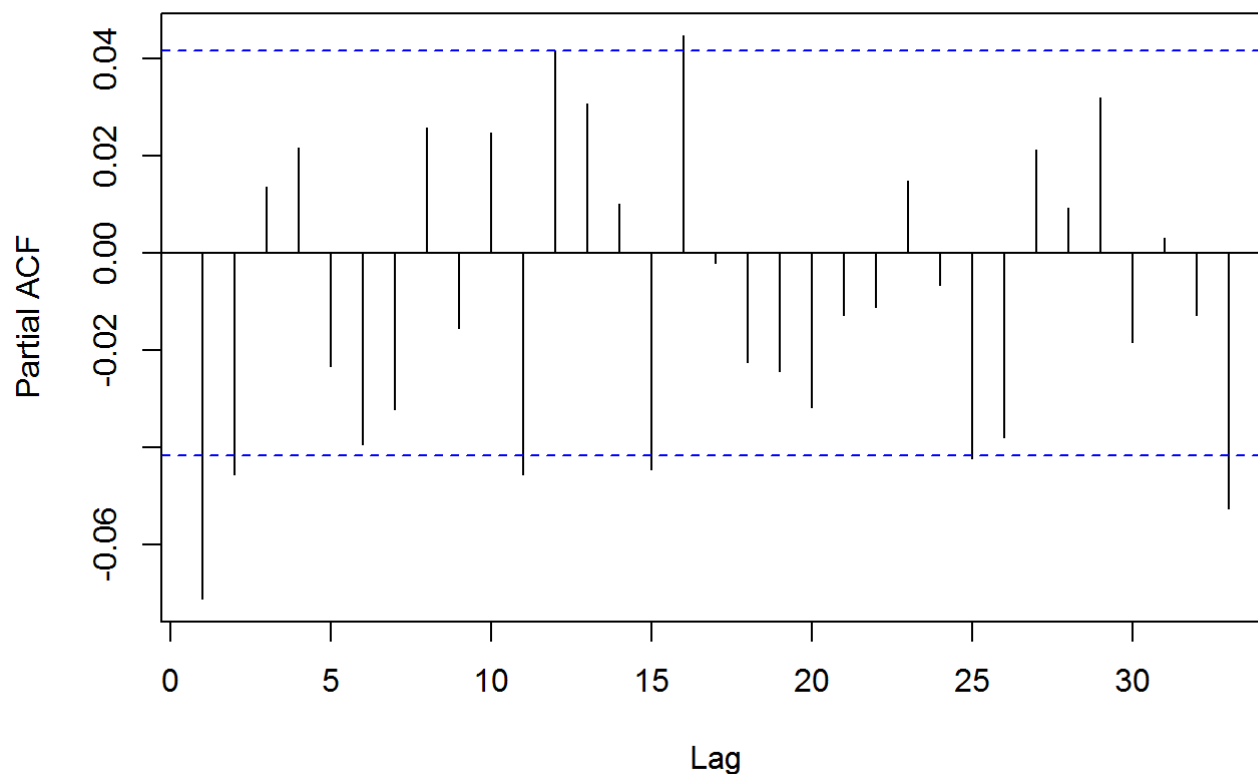
```
##
## Augmented Dickey-Fuller Test
##
## data:  djia.data.preinterv
## Dickey-Fuller = -12.149, Lag order = 13, p-value = 0.01
## alternative hypothesis: stationary
```

```
acf(djia.data.preinterv) # shows lag1 spike MA(1)
```

Series djia.data.preinterv

```
pacf(djia.data.preinterv)
```

Series djia.data.preinterv



```
price.2000.2003 <- window(DJIA, start = 1, end = 3512)
price.2000.2003 <- log(price.2000.2003[,2])

fit.interv <- arimax(price.2000.2003, order=c(2,1,2), xtransf=data.frame(I.1=1*(seq(price.2000.2
003)==2203), I.0=1*(seq(price.2000.2003)==2203)), transfer=list(c(0,0),c(1,0)), method='ML')
fit.interv
```

```
##
## Call:
## arimax(x = price.2000.2003, order = c(2, 1, 2), method = "ML", xtransf = data.frame(I.1 = 1 *
## (seq(price.2000.2003) == 2203), I.0 = 1 * (seq(price.2000.2003) == 2203)),
## transfer = list(c(0, 0), c(1, 0)))
##
## Coefficients:
##          ar1          ar2          ma1          ma2  I.1-MA0  I.0-AR1  I.0-MA0
##        -0.3126  -0.3772   0.2272   0.3045  -0.0252  -0.3999   0.0294
## s.e.    0.2211   0.1233   0.2257   0.1309   0.0384   0.2728   0.0340
##
## sigma^2 estimated as 0.0001503:  log likelihood = 10471.16,  aic = -20928.32
```

```
summary(fit.interv) #loglikelihood = 10465.63,  aic = -20921.25, MAE=0.008305585
```

```
##
## Call:
## arimax(x = price.2000.2003, order = c(2, 1, 2), method = "ML", xtransf = data.frame(I.1 = 1 *
##      (seq(price.2000.2003) == 2203), I.0 = 1 * (seq(price.2000.2003) == 2203)),
##      transfer = list(c(0, 0), c(1, 0)))
##
## Coefficients:
##          ar1          ar2          ma1          ma2  I.1-MA0  I.0-AR1  I.0-MA0
##      -0.3126  -0.3772   0.2272   0.3045  -0.0252  -0.3999   0.0294
## s.e.   0.2211   0.1233   0.2257   0.1309   0.0384   0.2728   0.0340
##
## sigma^2 estimated as 0.0001503:  log likelihood = 10471.16,  aic = -20928.32
##
## Training set error measures:
##              ME          RMSE          MAE          MPE          MAPE
## Training set 0.0001213312 0.01226032 0.008295931 0.001176996 0.08970643
##              MASE          ACF1
## Training set 0.9969304 -0.0008066632
```

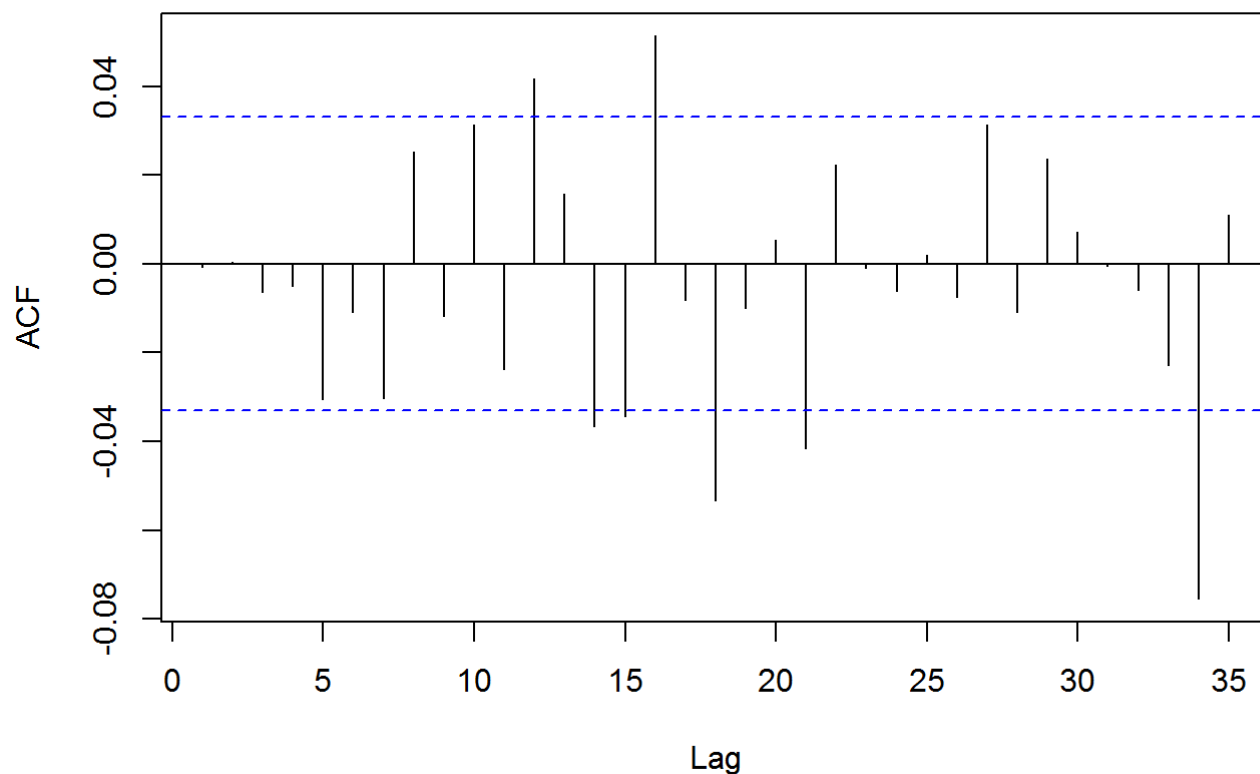
```
#Test residual -portmanteau' tests
```

```
Box.test(residuals(fit.interv), lag=48, type="Ljung") # failed to reject H0-0 autocorrelation
```

```
##
## Box-Ljung test
##
## data:  residuals(fit.interv)
## X-squared = 114.17, df = 48, p-value = 2.581e-07
```

```
acf(fit.interv$residuals)
```


Series fit.interv\$residuals



The Ljung box test failed to reject H_0 of independency, but we don't see significant spikes on lag1. Although there are few spikes exceed bounds. that might be due to chance

Compare with Holtwinter model using SSE

```
cbind(fit.arma$loglik, fit.interv$loglik)
```

```
##           [,1]      [,2]
## [1,] 10479.26 10471.16
```

```
cbind(fit.arma$aic, fit.interv$aic)
```

```
##           [,1]      [,2]
## [1,] -20942.51 -20928.32
```

```
fit.arma.mae = 0.0082958
fit.interv.mae = 0.008305585
```

looks like that arima fits better even with intervention added in

Future Work

```
detectA0(fit.interv, robust=F); detectI0(fit.interv, robust=F)
```

```
##          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## ind      73.00000 427.00000 638.00000 641.00000 686.00000 2198.00000
## lambda2 -4.70692 -5.980335 -4.360481 4.908215 4.744066 -5.434128
##          [,7]      [,8]      [,9]      [,10]     [,11]
## ind      2206.00000 2208.00000 2210.00000 2215.00000 2219.00000
## lambda2 -5.642978 7.955783 -6.194112 -5.082575 8.161684
##          [,12]     [,13]     [,14]     [,15]     [,16]
## ind      2225.00000 2235.00000 2237.00000 2238.00000 2242.00000
## lambda2 -4.353159 -4.455457 4.67717 4.47162 -5.917519
##          [,17]     [,18]     [,19]
## ind      2309.00000 2318.00000 2914.00000
## lambda2 4.821204 5.042813 -4.835103
```

```
##          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## ind      73.00000 427.00000 641.00000 686.00000 2198.00000 2206.00000
## lambda1 -4.965143 -6.07189 4.985771 5.160104 -5.778937 -6.008619
##          [,7]      [,8]      [,9]      [,10]     [,11]
## ind      2208.00000 2210.00000 2215.00000 2219.00000 2231.00000
## lambda1 8.341065 -6.205401 -4.966892 8.070278 4.791764
##          [,12]     [,13]     [,14]     [,15]     [,16]
## ind      2236.00000 2237.00000 2242.00000 2309.00000 2318.00000
## lambda1 -4.761923 4.469385 -6.284452 4.64674 5.164824
##          [,17]
## ind      2914.00000
## lambda1 -4.802836
```

There are significant outliers out there, so we could add these outliers to improve model for a better fit