MUNI ECON

Applied Financial Econometrics

Class 3: Unit root and autoregressive models

Lecturer: Axel A. Araneda, Ph.D.

Reviewing the previous class

A time-series is strictly stationary is the probabilistic behavior of each set of values $\{x_{t1}, x_{t2},\ldots, x_{tk}\}$ is identical to same set displaced on time; i.e., $\{x_{t1+h}, x_{t2+h},\ldots, x_{tk+h}\}$

$$\Rightarrow P(x_t \leq c) = P(x_2 \leq c)$$

A weakly stationary time-series x_t is a finite variance process such that:

- $\mathbb{E}(x_t) = u_t = u$
- $\mathbb{E}(x_t^2) < \infty$
- $\mathbb{E}(x_{t_1}, x_{t_2}) = \mathbb{E}(x_{t_1 - h}, x_{t_2 - h})$

A stochastic process is a martingale if:

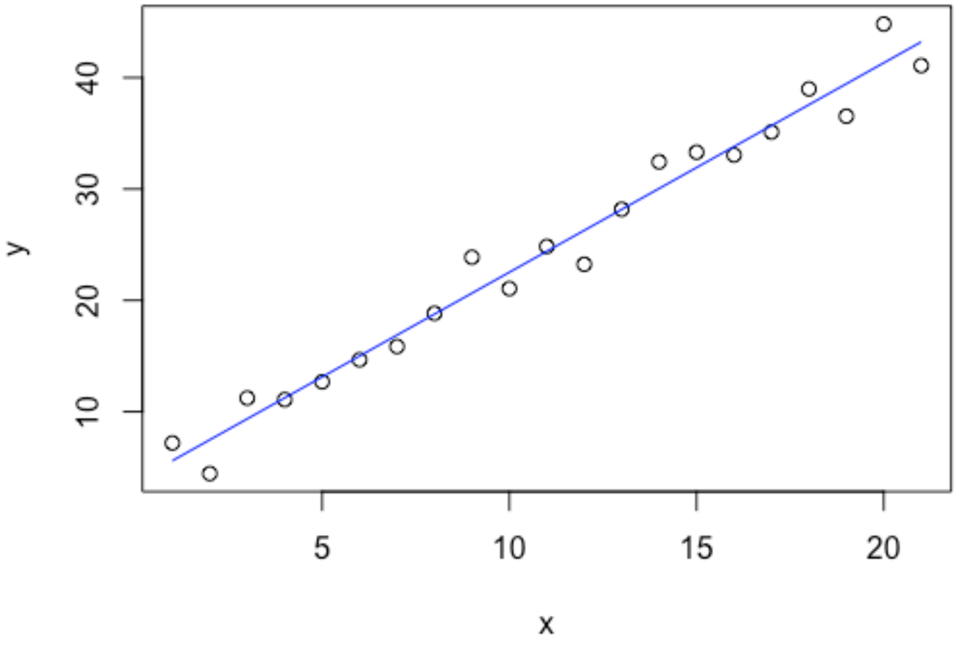
- $\mathbb{E}(x_{t+1} | x_1, x_2, \ldots, x_t) = x_t$

MUNI ECON

Reviewing the previous class

– Simple linear regression:

$$y_t = \alpha + \beta x_t + \epsilon_t$$




```
reg1 <- lm( y ~  x)
summary(reg1)
fitvalues<-fitted(reg1)
plot(y)
lines(fitvalues)
```

```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0462 -1.0274 -0.1558  1.4172  3.4861 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.60072    0.83319   6.722 2.01e-06 ***
x            1.88060    0.07127  26.386  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.978 on 19 degrees of freedom
Multiple R-squared:  0.9734,	Adjusted R-squared:  0.972 
F-statistic: 696.2 on 1 and 19 DF,  p-value: < 2.2e-16
```


MUNI ECON

Reviewing the previous class

- Gauss-Markov Theorem:
  1. There is a linear relationship between X and Y
  2. No multicollinearity (X is linearly independent)
  3. $E(\boldsymbol{\varepsilon}|\boldsymbol{X}) = 0$. Equivalently $E(\boldsymbol{Y}) = \boldsymbol{X}\beta$
  4. $E(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'|\boldsymbol{X}) = \sigma^2\boldsymbol{I}$. Errors are homoscedastic and there is no autocorrelation.
  5. $\boldsymbol{X}$ and $\boldsymbol{\varepsilon}$ are unrelated. $Cov(\boldsymbol{X}\boldsymbol{\varepsilon}) = 0$

- Evaluation of regressions:
  1. F-test.
  2. T-test.
  3. Coefficient of determination R^2.
  4. Adjusted-R^2 (R^2_{adj}).

MUNI ECON

Lecture 2
Unit Root and Autoregressive models

- Stationarity, Unit root, Box-Jenkins methodology, ARIMA.

MUNI ECON

The impacts of shocks

- Let's consider the following AR process: $y_t = \phi y_{t-1} + \varepsilon_t$

- Iteratively: $y_t = \phi^t y + \varepsilon_t + \phi \varepsilon_{t-1} + \phi^2 \varepsilon_{t-2} + \phi^3 \varepsilon_{t-3} + \ldots + \phi^t \varepsilon_0$

1. If $\phi < 1 \Rightarrow \phi^t \to 0$ (as $t \to \infty$) shocks gradually die away
2. If $\phi = 1$ shocks persist
3. If $\phi > 1$ shocks becomes more influential ($\phi < \phi^2 < \phi^3$ ...)

Assesing stationarity: unit root

- Mathematically, the stationarity arises when Module of the roots from the characteristic polynomial should be greater than 1.

- If the process has unit root (non-stationary) we just need to differentiate the process once.

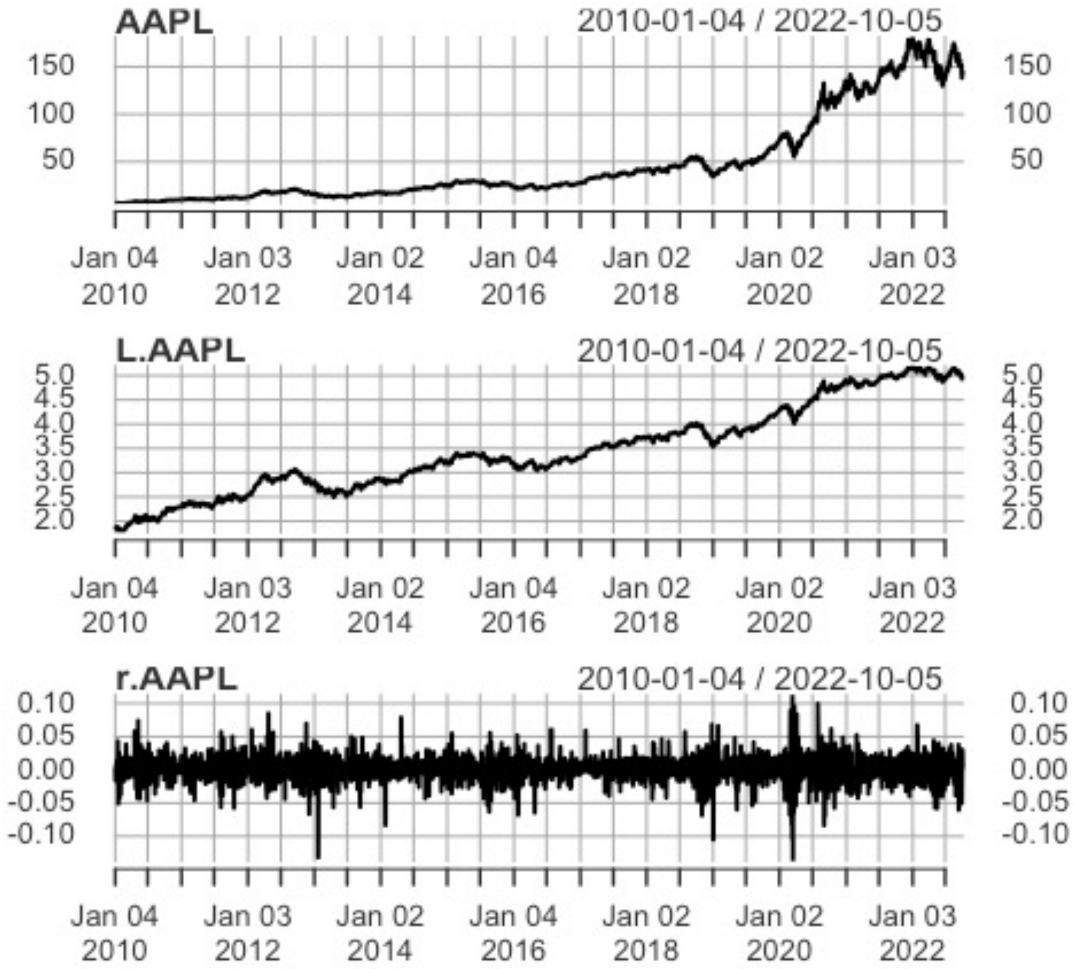
- In general, if the process has unit root with multiplicity d, we just need to differentiate the process d times.

Integrated process

- An Integrated process is a non-stationary process, which can be transformed to a stationary process by differentiating.
- The sequence x_t is integrated of order d, $I(d)$, if it requires to be differentiated d times to become stationary.
- All Integrated Processes are non-stationary, but not all non-stationary processes are integrated
- So if $y_t \sim I(d)$ then $\Delta^d y_t \sim I(0)$.

Unit root test: the Dickey-Fuller test

```
library(quantmod)
getSymbols('AAPL',src='yahoo', from="2010-01-01",
    periodicity = 'daily') # Apple since 2010
AAPL<-AAPL$AAPL.Adjusted # ajd-closing prices
L.AAPL<-log(AAPL) # Log prices
r.AAPL<-diff(L.AAPL) # Log-returns
par(mfrow=c(3,1))
plot(AAPL)
plot(L.AAPL)
plot(r.AAPL)
```



MUNI ECON

Unit root test: the Dickey-Fuller test

```
library(tseries)
adf.test(AAPL)
adf.test(AAPL,k=1) # k: number of lags
adf.test(L.AAPL,k=1)
adf.test(r)
head(r)
adf.test(r[2:length(r)])
adf.test(r[2:length(r)],k=1)
```

```
        Augmented Dickey-Fuller Test

data:  AAPL
Dickey-Fuller = -1.4038, Lag order = 1, p-value = 0.8307
alternative hypothesis: stationary
```

```
        Augmented Dickey-Fuller Test

data:  L.AAPL
Dickey-Fuller = -2.3349, Lag order = 1, p-value = 0.4365
alternative hypothesis: stationary
```

```
        Augmented Dickey-Fuller Test

data:  r.AAPL[2:length(r.AAPL)]
Dickey-Fuller = -40.737, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary
```

AR(p) models

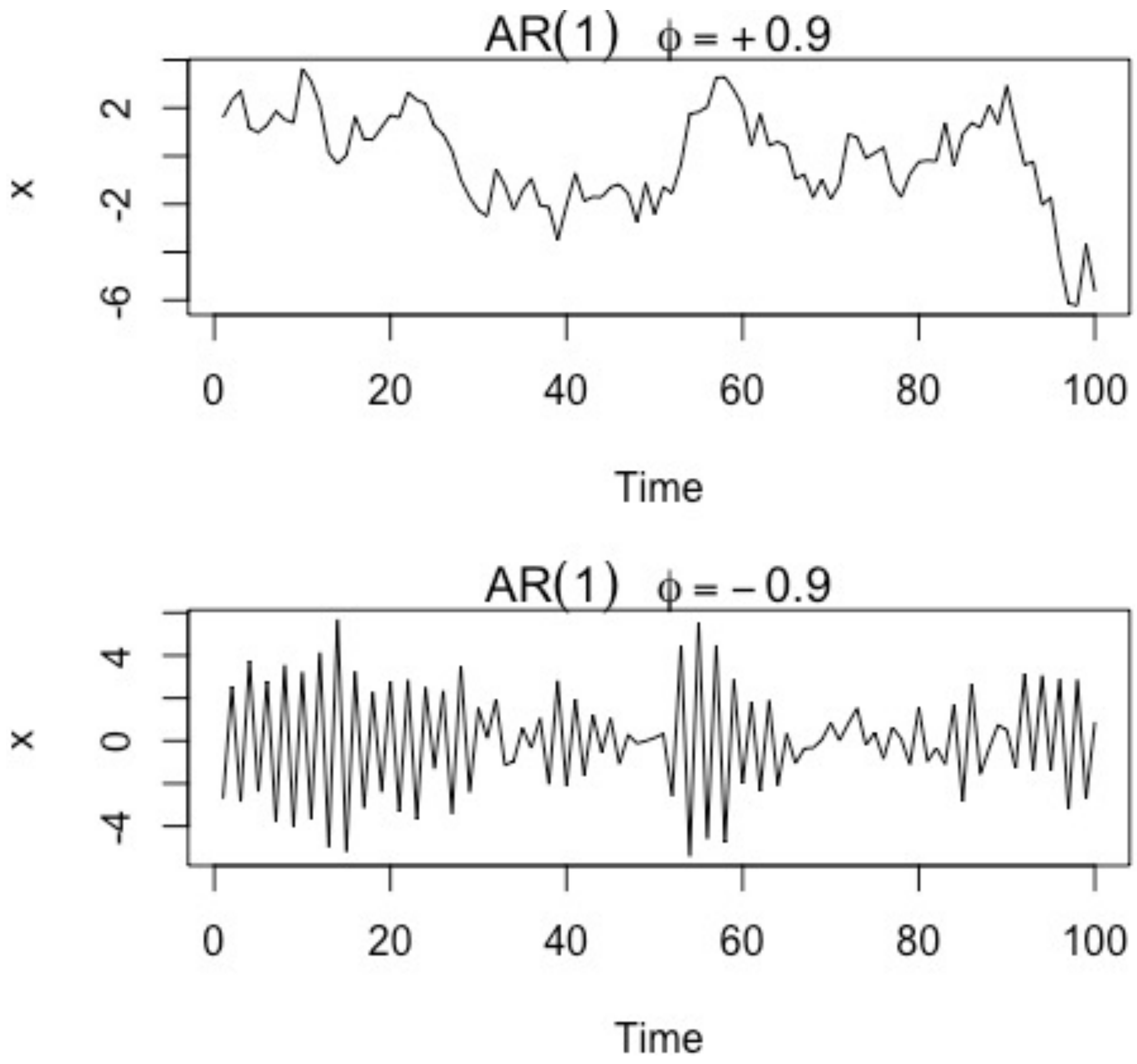
$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + \epsilon_t$$

- Where x_t is an stationary time-series and each $\emptyset_t$ constant.
- AR(1): $$x_t = \phi x_{t-1} + \epsilon_t$$
- Stationarity of AR(1): $\emptyset<1$

MUNI
ECON

AR(1) simulation

```
par(mar=c(5,4,1,1))
par(mfrow=c(2,1))
plot(arima.sim(list(order=c(1,0,0), ar=.9), n=100),
     ylab="x",main=(expression(AR(1)~~~phi==+.9)))
plot(arima.sim(list(order=c(1,0,0), ar=-.9), n=100),
     ylab="x",main=(expression(AR(1)~~~phi==-.9)))
```



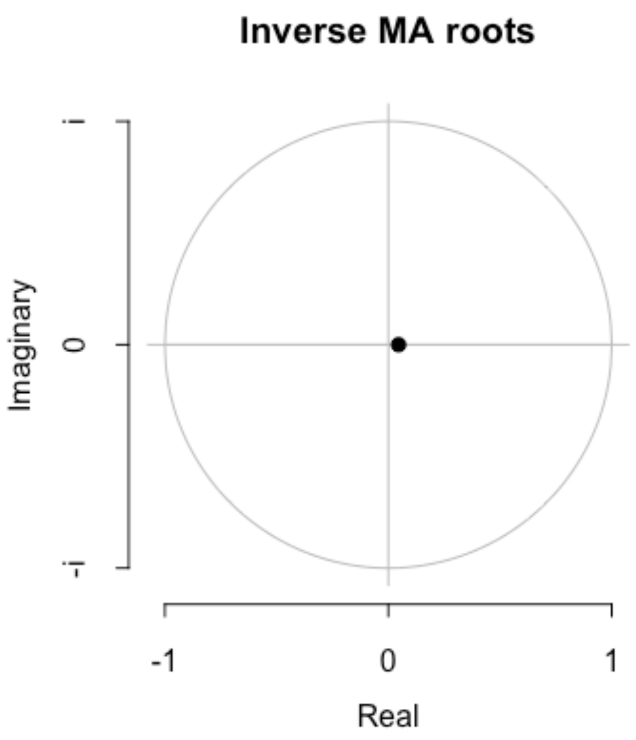
MUNI ECON

ARMA(p,q) models

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}$$

```
library(forecast)
fit <- auto.arima(L.AAPL)
summary(fit)
plot(fit)
```

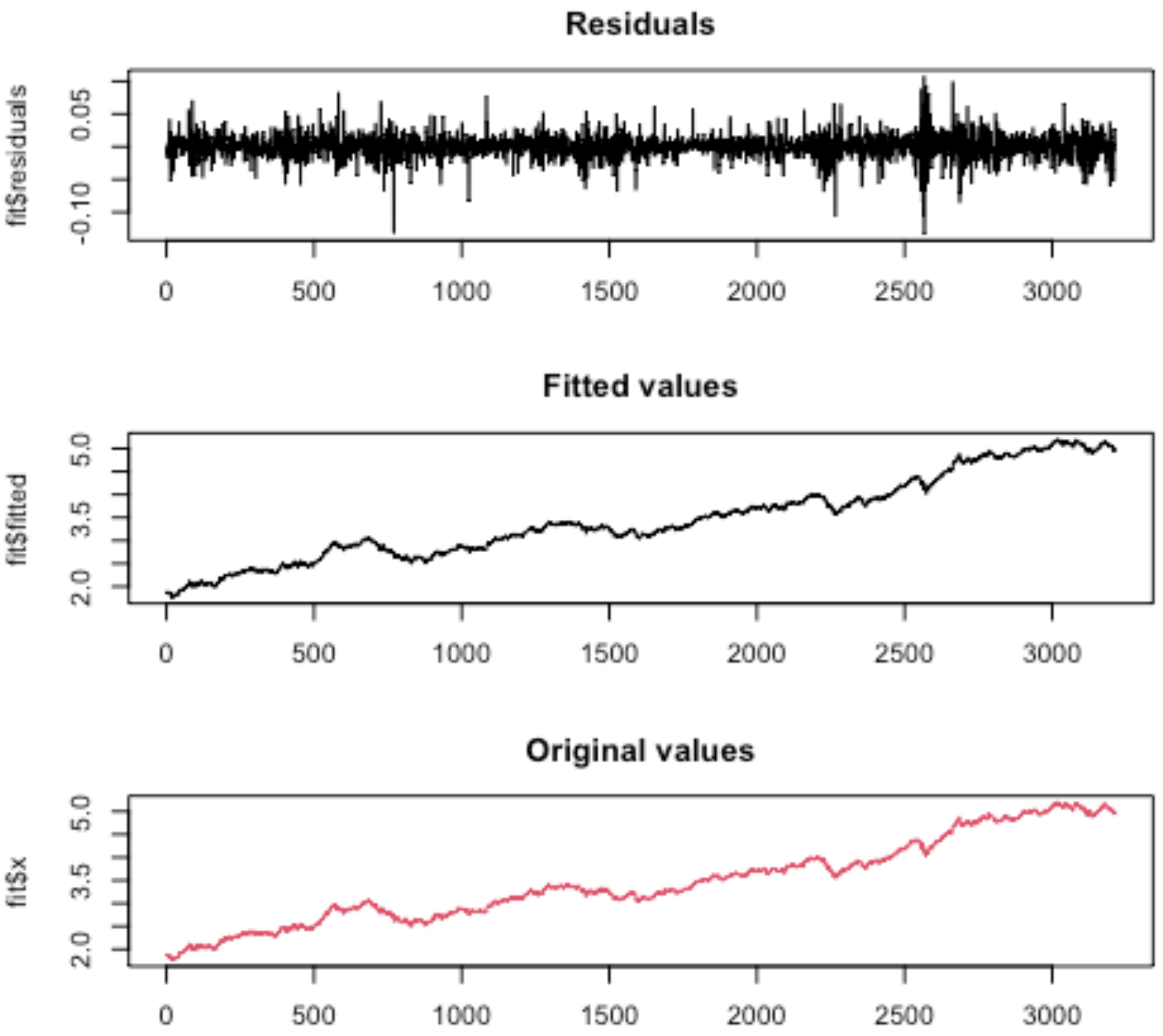
```
Series: L.AAPL
ARIMA(0,1,1) with drift

Coefficients:
          ma1   drift
      -0.0460   1e-03
s.e.   0.0176   3e-04

sigma^2 = 0.0003213:  log likelihood = 8358.08
AIC=-16710.16   AICc=-16710.15   BIC=-16691.93

Training set error measures:
                       ME       RMSE       MAE          MPE      MAPE      MASE          ACF1
Training set 5.539763e-07 0.0179163 0.012619 -0.001111661 0.3853841 0.9967757 -0.0002264444
```



MUNI ECON

ARIMA modelling

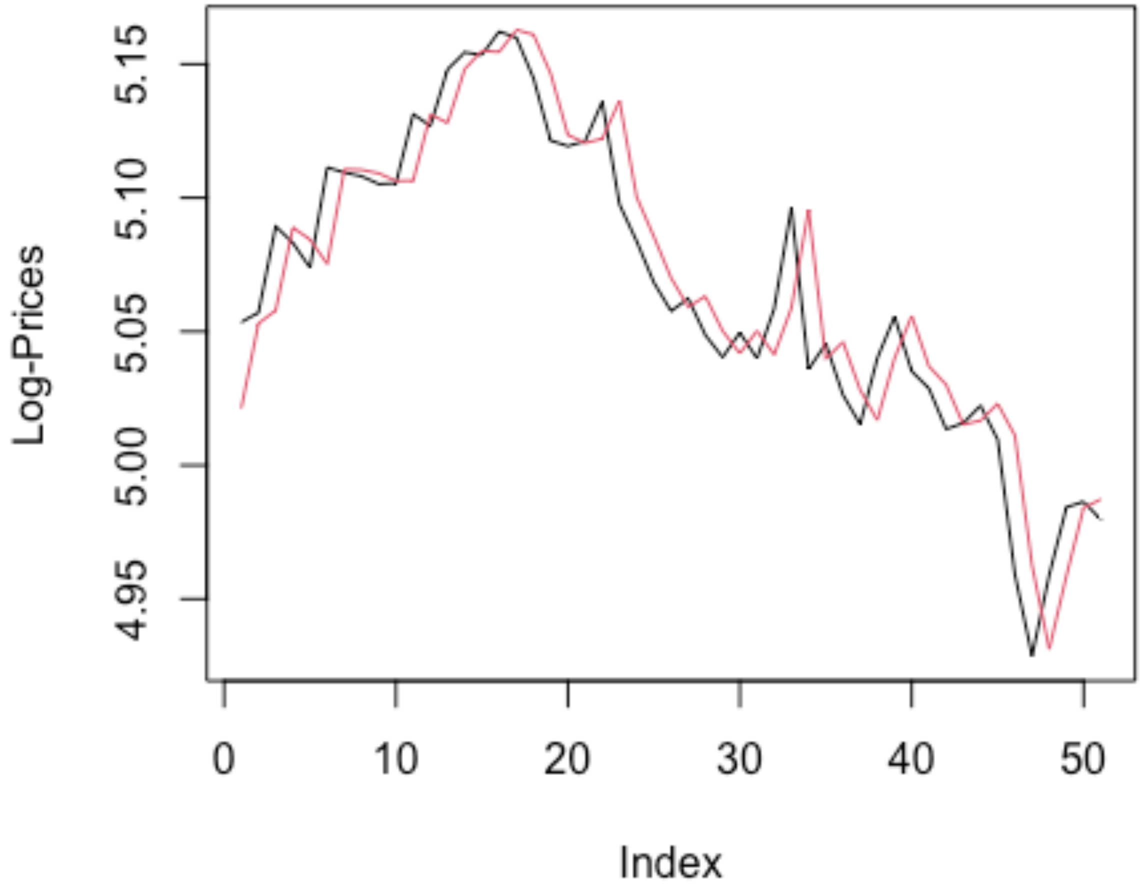
```
fit <- auto.arima(L.AAPL)
par(mfrow=c(3,1),mar=c(3,4,3,1))
plot(fit$residuals,main='Residuals')
plot(fit$fitted,main='Fitted values')
plot(fit$x,col=2,main='Original values')
```



MUNI ECON

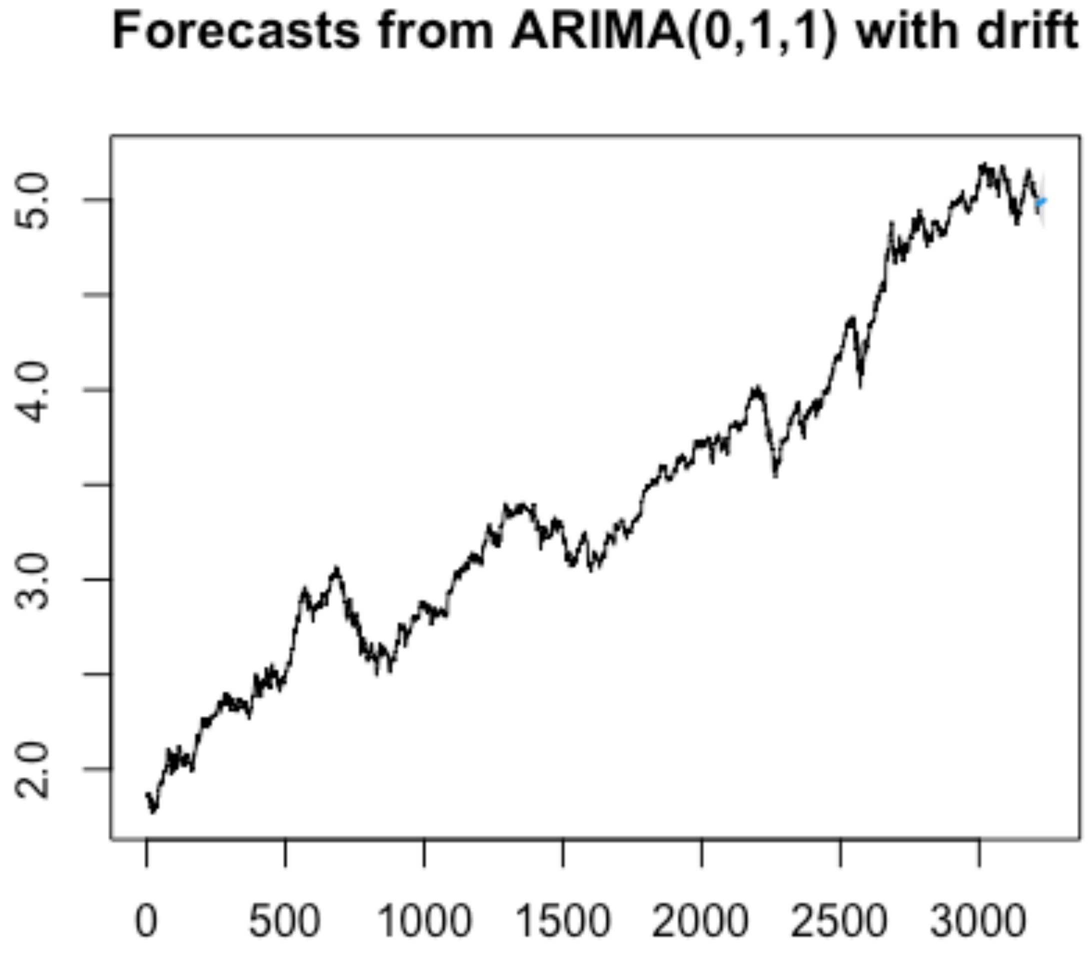
ARIMA modelling

```
##### Just to plot the last 50 values
fit_values<-fit$fitted
orig<-as.numeric(L.AAPL)
dev.off() # reset graphic settings
plot(orig[(length(orig)-50):length(orig)],
     type='l',ylab='Log-Prices')
lines(fit_values[(length(orig)-50):length(orig)],
     col=2)
###########
```



ARIMA modelling (forecasting)

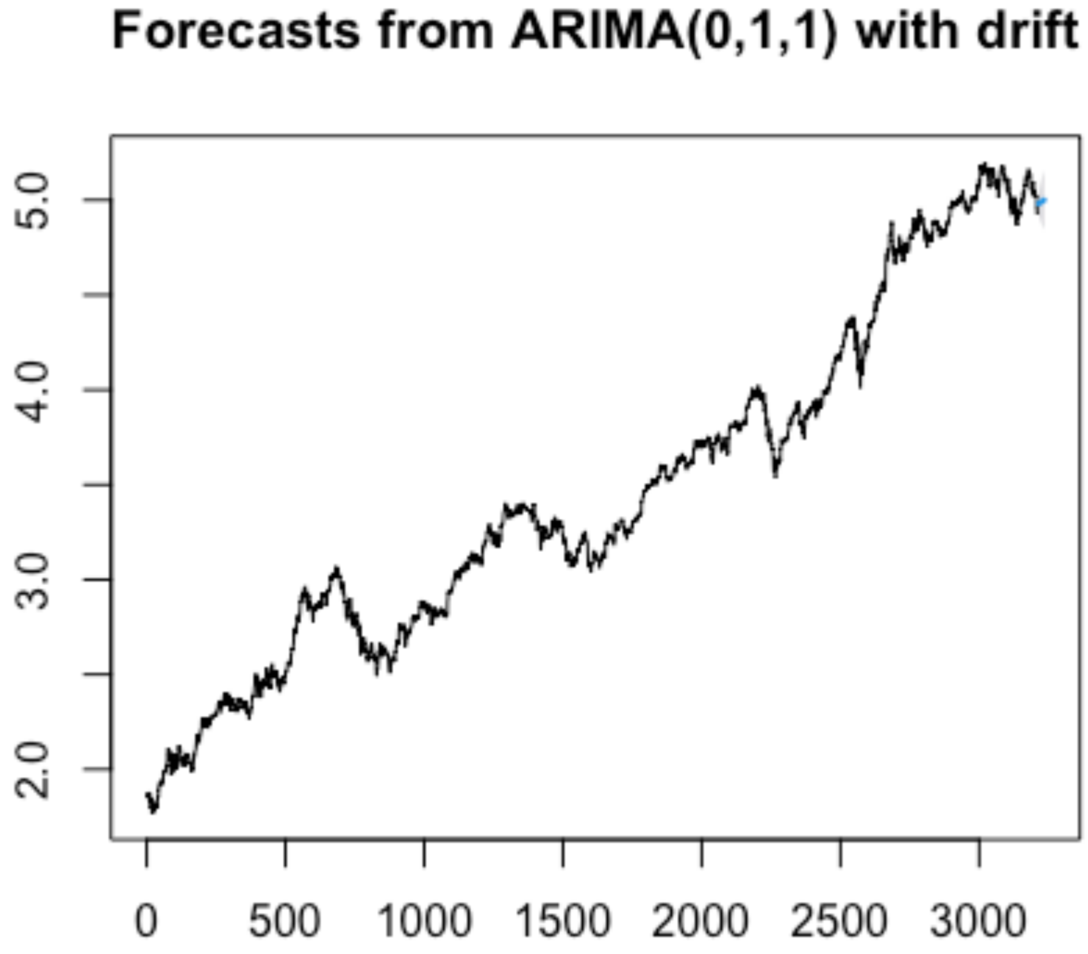
```
###########Forecasting ########################
autoarima_forecasting<-forecast(fit,21,level=95)
# Forecast future 21 values and 95% CI
plot(autoarima_forecasting)
###############################################
```



MUNI ECON

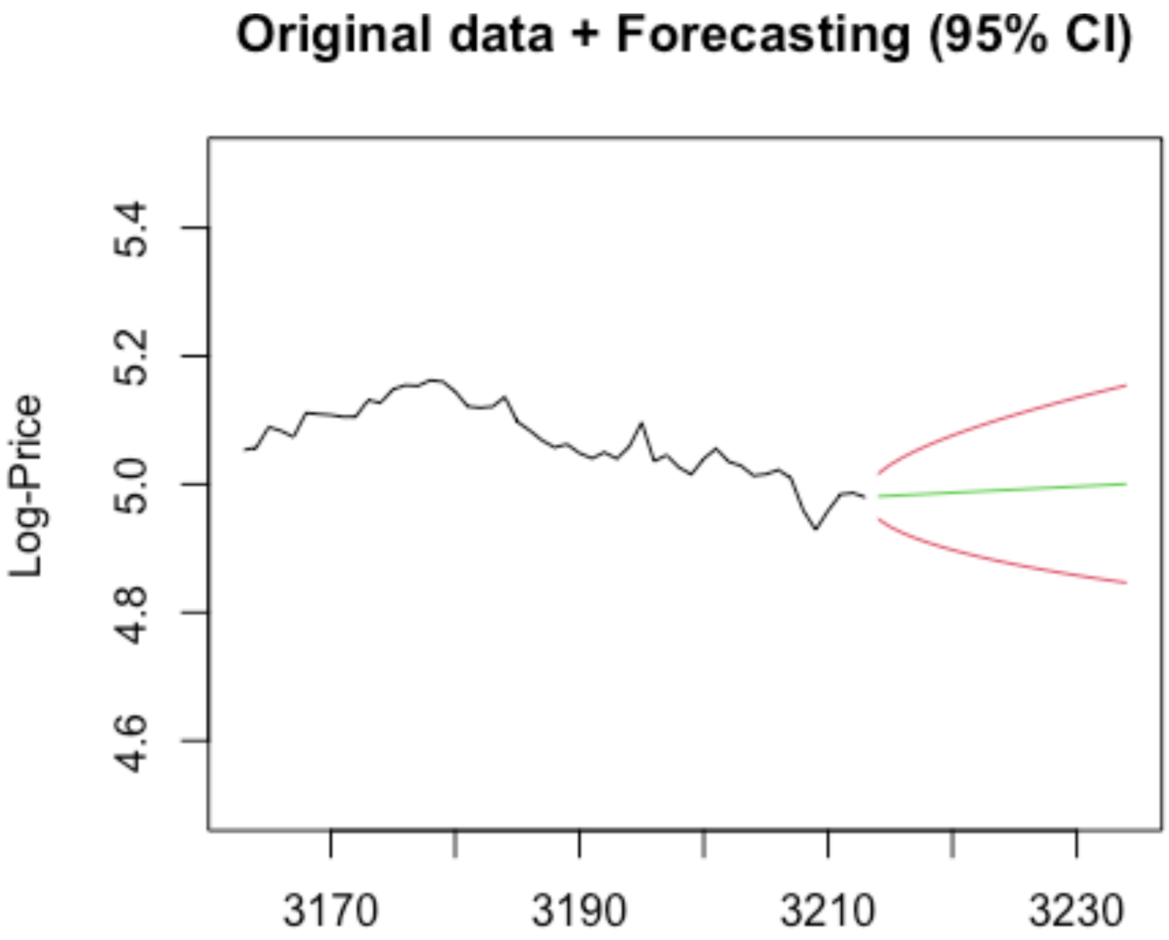
ARIMA modelling (forecasting)

```
###########Forecasting ########################
autoarima_forecasting<-forecast(fit,21,level=95)
# Forecast future 21 values and 95% CI
plot(autoarima_forecasting)
###############################################
```



ARIMA modelling (forecasting)

```
##### Just to plot the last 50 values plus forecasting
mean_forecast<-autoarima_forecasting$mean
lower_95<-autoarima_forecasting$lower
upper_95<-autoarima_forecasting$upper
plot((length(orig)-50):(length(orig)),
     orig[(length(orig)-50):length(orig)],
     type='l',xlim=c((length(orig)-50),
    length(orig)+21),ylim=c(4.5,5.5),
     main='Original data + Forecasting (95% CI)',
    ylab=('Log-Price'),xlab='index')
lines(mean_forecast,ylim=c(4.5,5.5),col=3)
lines(upper_95,,col=2,)
lines(lower_95,col=2)
```



MUNI
ECON

ARIMA modelling (comparison)

```
#ARIMA(1,1,1)
arima_1_1_1=arima(L.AAPL, order=c(1,1,1))
print(arima_1_1_1)
#p-values (t-test) for each coefficient
library(lmtest)
coeftest(arima_1_1_1)
#####
#ARIMA(1,1,0)
arima_1_1_0=arima(L.AAPL, order=c(1,1,0))
print(arima_1_1_0)
coeftest(arima_1_1_0)
```

```
#comparing AIC (lower wins)
arima_1_1_1$aic
fit$aic £
arima_1_1_0$aic
#comparing BIC (lower wins)
BIC(arima_1_1_0)
BIC(arima_1_1_1)
BIC(fit)
```


MUNI ECON