Week 4 - AYU - Pod

Contents

1. Time Series in R

There are multiple R objects to store a time series. In this practice, we will use the `ts` object (come with bases R) and the `xts` object to work with time series. Install the `xts` package using

```
install.packages('xts')
```

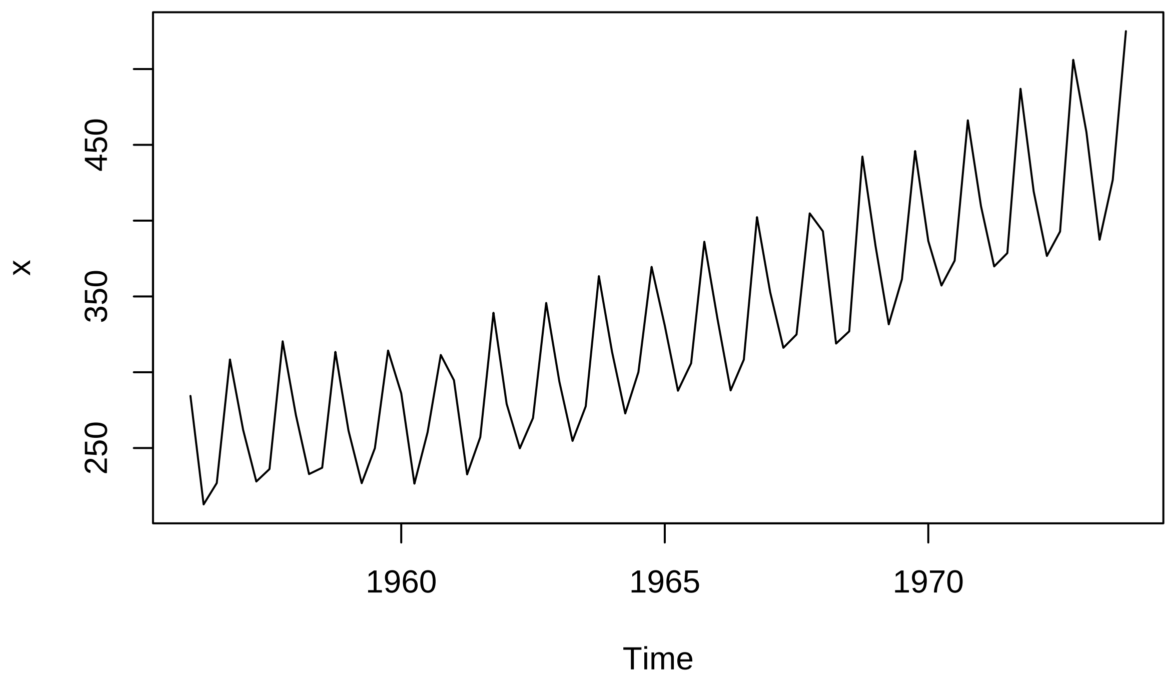
```
library(tidyverse)
library(xts)
library(fpp2)
library(TTR)
```

Consider the series of quarterly beer production in Australia stored in the dataset beer.csv. Use the following codes to read the dataset and create a time series object `ts`.

```r
t1 = read_csv("beer.csv")
t1 = ts(t1, start = c(1956, 1), freq = 4)
```

In the `ts` function, we use `freq = 4` for quarterly data, `freq = 12` for monthly data. The argument `start = c(1956, 1)` means the series start with the first quarter in 1956. We then can plot the series using the plot function.

```r
plot(t1)
```



2. Moving Average Smoothing

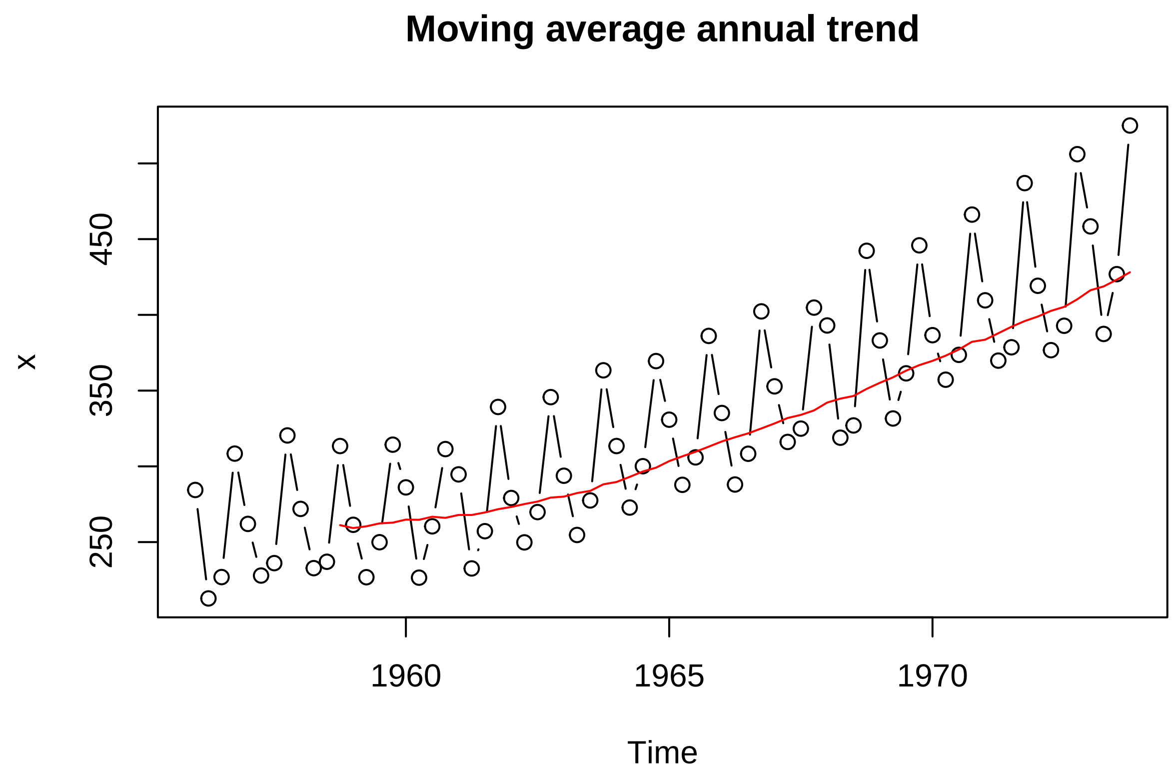
The plot shows that there is a seasonal component in the series. To see the trend of the series better we can use smoothing techniques. We will consider two techniques: moving average and exponential. With moving average, a new series is created by average the most recent observations of the original series. For example, moving average using the last two observations using the below formula.

$$s_{n+1} = \frac{y_n + y_{n-1} + y_{n-2}}{3}$$

Using this code to perform moving average smoothing and plot the smoothed series.

```r
## Data 1
t1_sma = SMA(t1, n = 12)
```

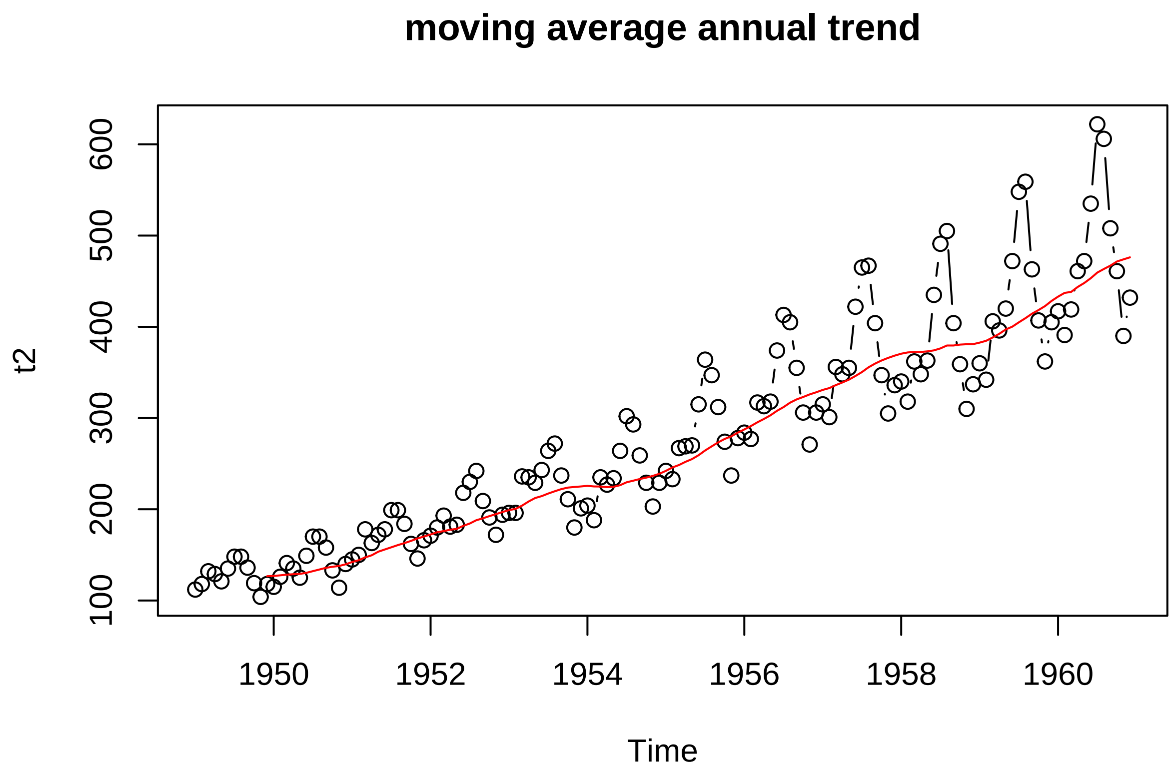
```r
plot(t1, type= "b", main = "Moving average annual trend")
lines(t1_sma, col = "red")
```



Question

- Import an airline passenger series data in AirPassengers.csv and plot the smoothed moving average series and the original series.

```r
## Data 2
d2 <-  read_csv('AirPassengers.csv')
t2 = ts(d2$Number_Passengers, start = c(1949, 1), frequency = 12)

t2_sma = SMA(t2, n =12)
plot(t2, type= "b", main = "moving average annual trend")
lines(t2_sma, col = "red")
```



3. Exponentioal Smoothing

While simple moving average distribute the weight equally to a few most recent observations, exponential smoothing distribute the weight unequally to ALL observations. The weights distribution is controlled by parameter α. Large α (close to 1) will give higher weights to the most recent observations.

$$s_{n+1} = \alpha y_n + \alpha(1-\alpha)^1 y_{n-1} + \alpha(1-\alpha)^2 y_{n-2} + ... + \alpha(1-\alpha)^{n-1} y_1$$

Question

- Import an airline passenger series data in AirPassengers.csv and plot the smoothed moving average series and the original series.

4. Auroregression first order/AR(1) Model

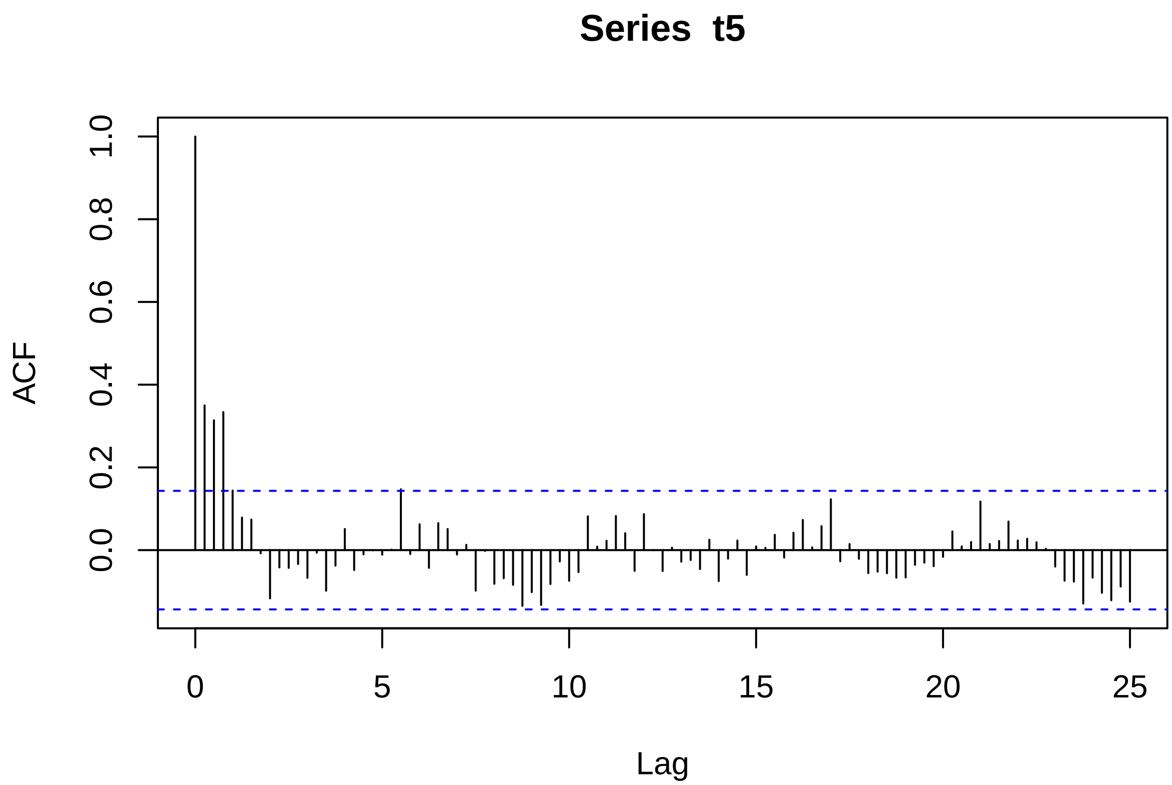
Autoregression model of the first order is defined as

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t$$

where $|\beta_1| < 1$, $\epsilon \sim (0, \sigma^2)$.

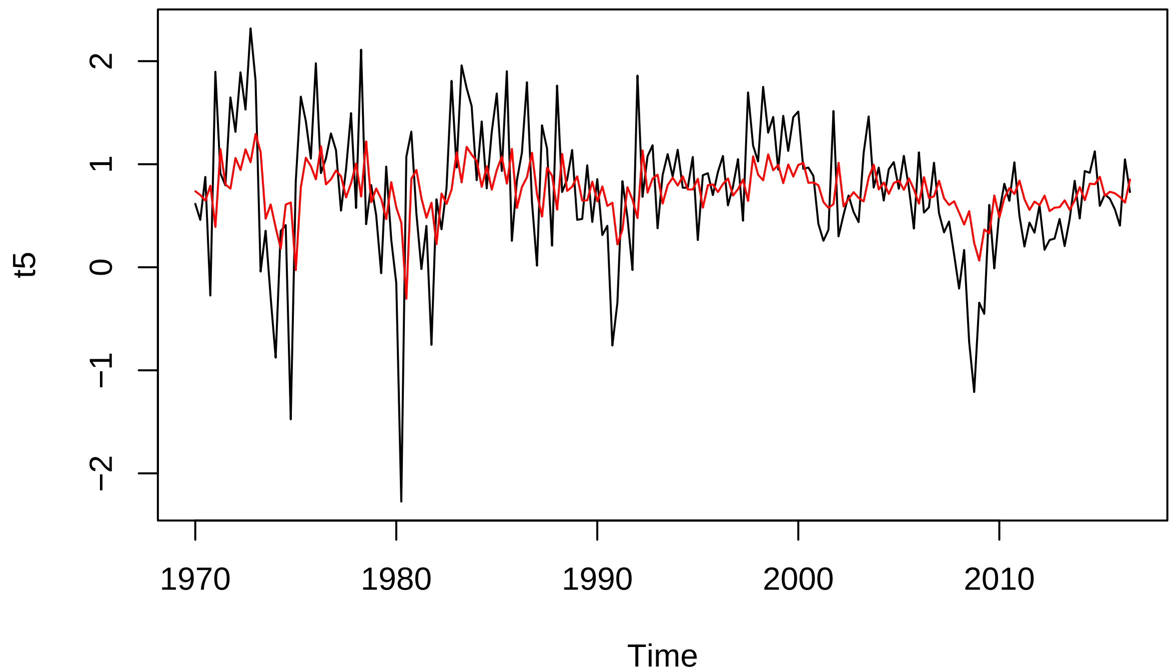
We can use AR(1) for forecasting time series. The series should be tested for stationary before modeled by an autoregression model. In this example, we will use the uschange data in the `fpp2` package to demonstrate the application of the AR(1) model.

```r
t5 = uschange[,"Consumption"]
acf(t5, lag.max = 100)
```



We can see the the the autocorrelations when the lag increases are within the blue line and die out to zero. This indicates that the series is stationary and we can use AR(1) model on it.

```r
a5 = arima(t5, order = c(1,0,0))
plot(t5)
lines(t5-a5$residuals, col = "red")
```



Forecast the next

```
forecast(a5, h = 1)
```

```
##         Point Forecast       Lo 80    Hi 80      Lo 95    Hi 95
## 2016 Q4      0.7403201 -0.04432921 1.524969 -0.4596975 1.940338
```

Question:

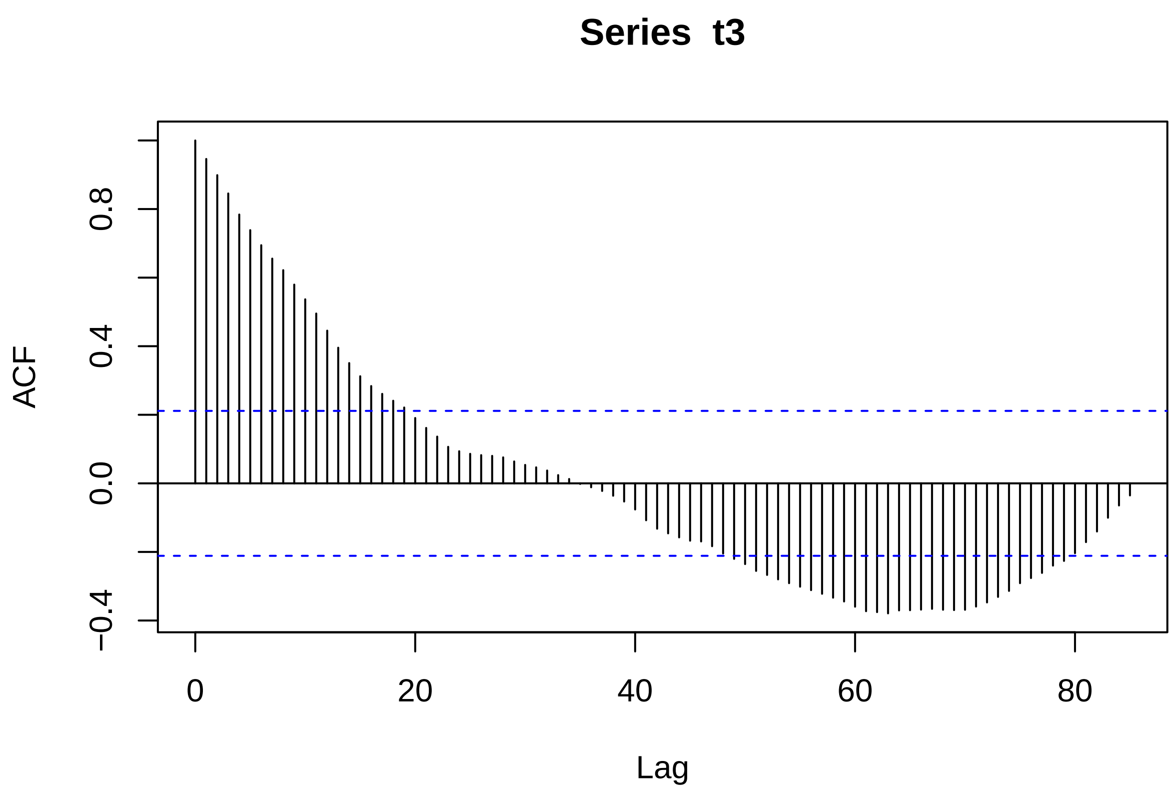
- Plot the ACF of the US Income Series. Verify that the series is stationary.
- Use the AR(1) model to predict the income of the next quarter in the series.

5. AR(1) model on Stock time series

```
# Check stationary of the stock

# Using ACF: for stattionary: ACF drop to 0 quickly while ACF of non-stattionary drop to 0 slowly.
# r1 of non-stationary is also large and positive.

library(xts)
library(quantmod)
library(forecast)
getSymbols("AAPL")
```

```
## [1] "AAPL"
```

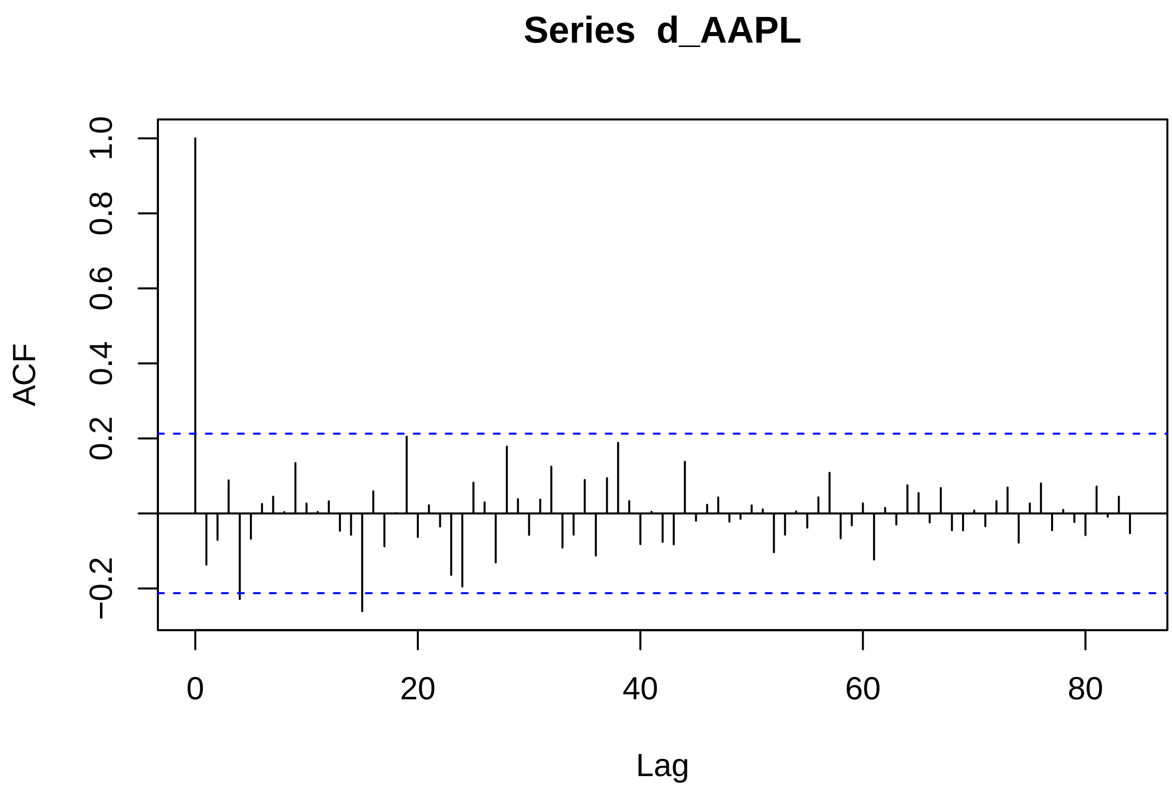
```
t3 = AAPL$AAPL.Open
t3 <- t3[index(t3) > "2023-01-01"]

AAPL = ts(as.numeric(t3))

acf(t3, lag.max = 100)
```



Looking at the ACF, we observe that the correlations does not die out to zero but later go out of the blue lines. Thus this series is non-stationary and it is not reasonable to model the stock using the AR model. We will use a differencing techniques to transform the stock to a stationary series. Consider the difference stock

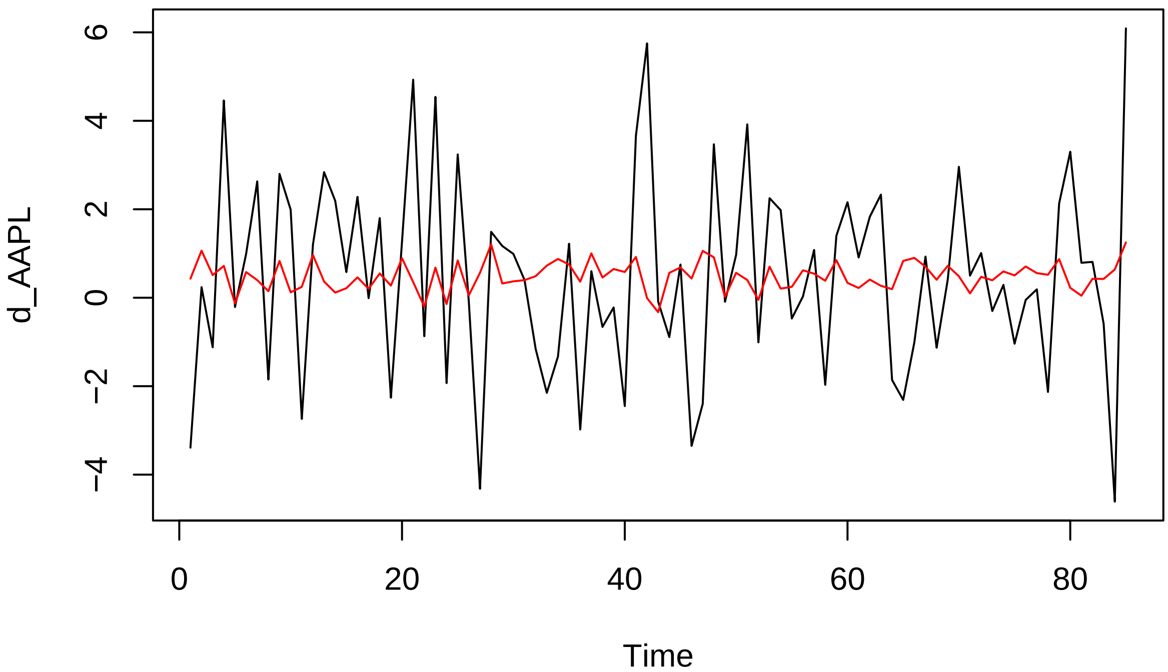
$$d_n = y_{n+1} - y_n$$

```
d_AAPL = ts(as.numeric(diff(t3))[-1])
acf(d_AAPL, lag.max = 100)
```



The ACF plot shows that the difference series is stationary and can be model by an AR model.

```
ar_AAPL = arima(d_AAPL, order = c(1,0,0))
plot(d_AAPL)
lines(d_AAPL-ar_AAPL$residuals, col = "red")
```



Forecast the next observation of d_n. Notice that $y_{n+1} = y_n + d_n$, we can forecast y_{n+1} using y_n and d_n

```
d_n = forecast(ar_AAPL, h = 1)

y_next = d_n$mean + t3[length(t3)]
y_next = as.numeric(y_next)
y_next
```

```
## [1] 170.6011
```

Question

- Follow the above example to predict the next opening value of the Microsoft Stock.