Lab 2

PSTAT W 174/274

Plotting time series

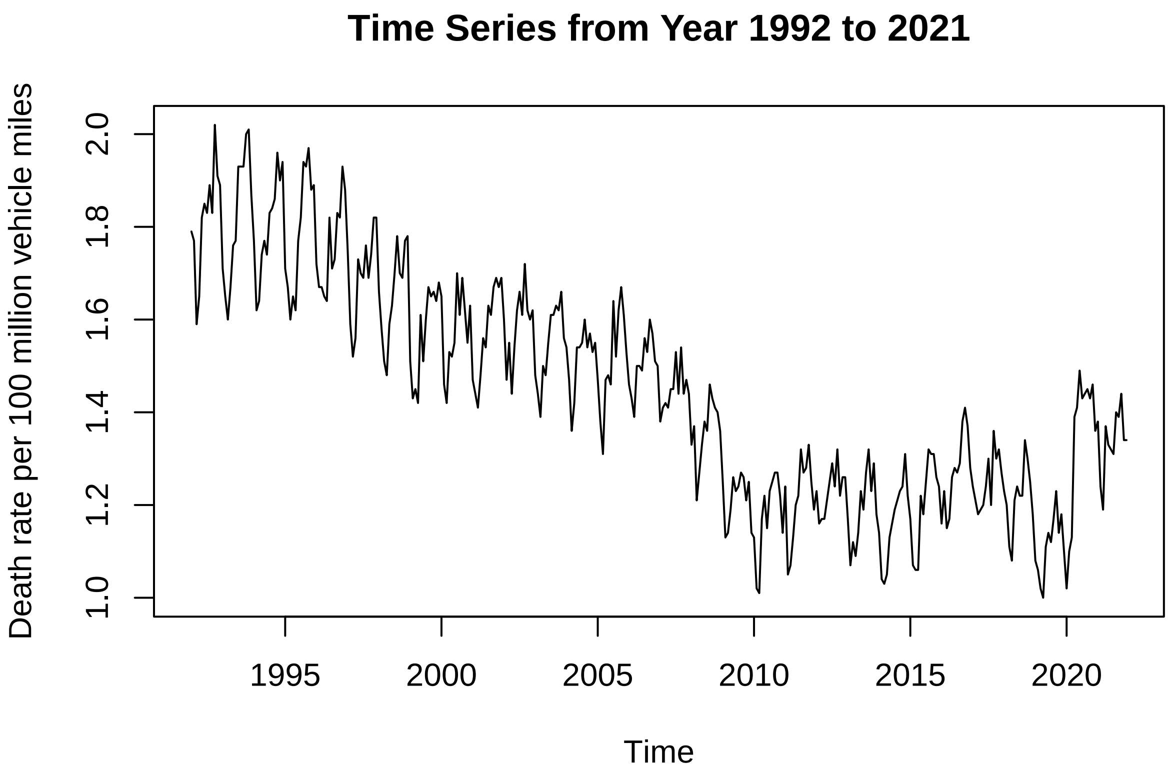
1. Travel patterns largely returned to normal in 2021 compared to the highly unusual patterns experienced during the peak of the pandemic in 2020. The file 'Deaths by Month Table.xlsx' contains information about vehicle-related deaths from January 1992 to December 2021. We can import the data into R by typing:

```r
library("readxl")
motor_death_month = read_excel("Deaths by Month Table.xlsx"
                               ,skip=1, n_max=391)[-c(1,2),]
#delete summery information
delete = c(1:30)*13
motor_death_month = motor_death_month[-delete,]
#reorder the data.
groups = split(motor_death_month, rep(1:30,each=12))
motor_death_month = do.call(rbind, rev(groups))
```

When the data is presented with multiple columns, such as in this dataset where we have columns for "Death," "Death rate per 100 million vehicle miles," and "Vehicle miles in billions," we need to select one column as our time series data. In this case, we will use the "Death rate per 100 million vehicle miles" column for the following demonstration.

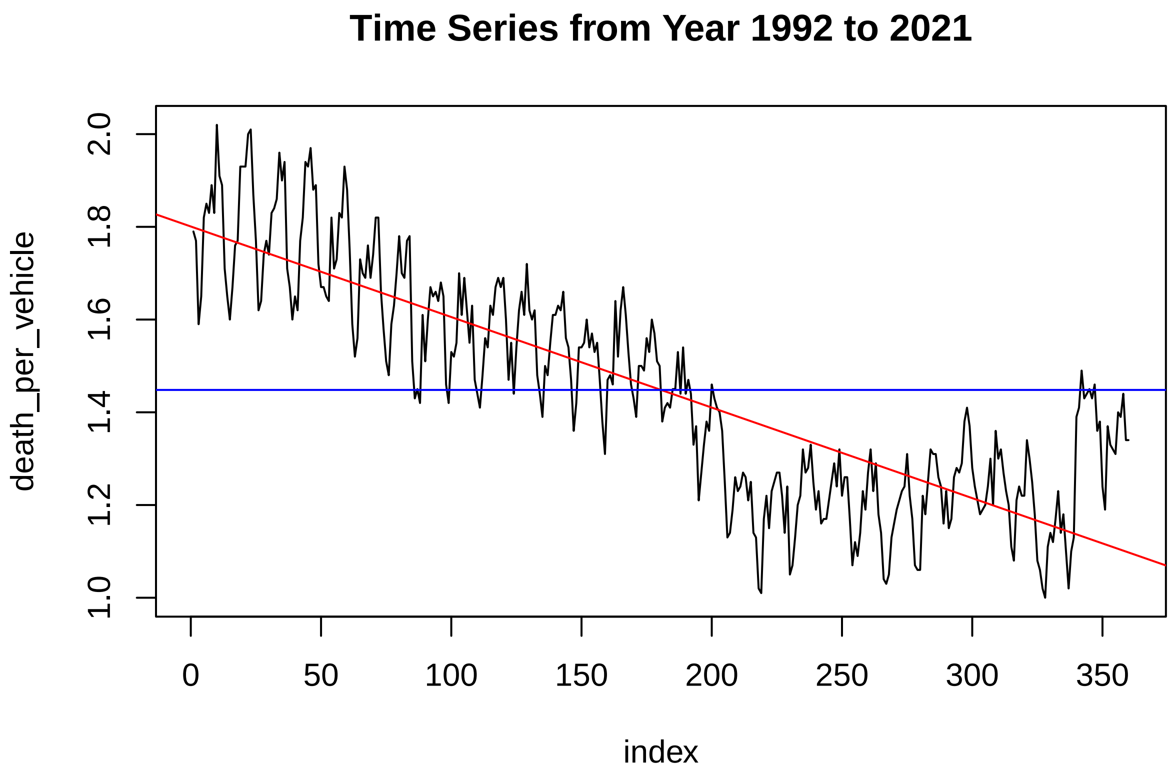
(a). We can plot the data versus each month using the following command:

```r
death_per_vehicle = ts(motor_death_month[,4], start = c(1992,1), frequency = 12)
plot(death_per_vehicle, main = "Time Series from Year 1992 to 2021")
```



(b). In order to see the trend and mean lines on the same plot, we can plot the data versus index number from 1, 2, . . . n using the following command:

```
plot(1:length(death_per_vehicle),death_per_vehicle, main =
   "Time Series from Year 1992 to 2021", type = 'l',xlab='index')
index = 1: length(death_per_vehicle)
trend <- lm(death_per_vehicle ~ index)
abline(trend, col="red")
abline(h=mean(death_per_vehicle) , col='blue')
```

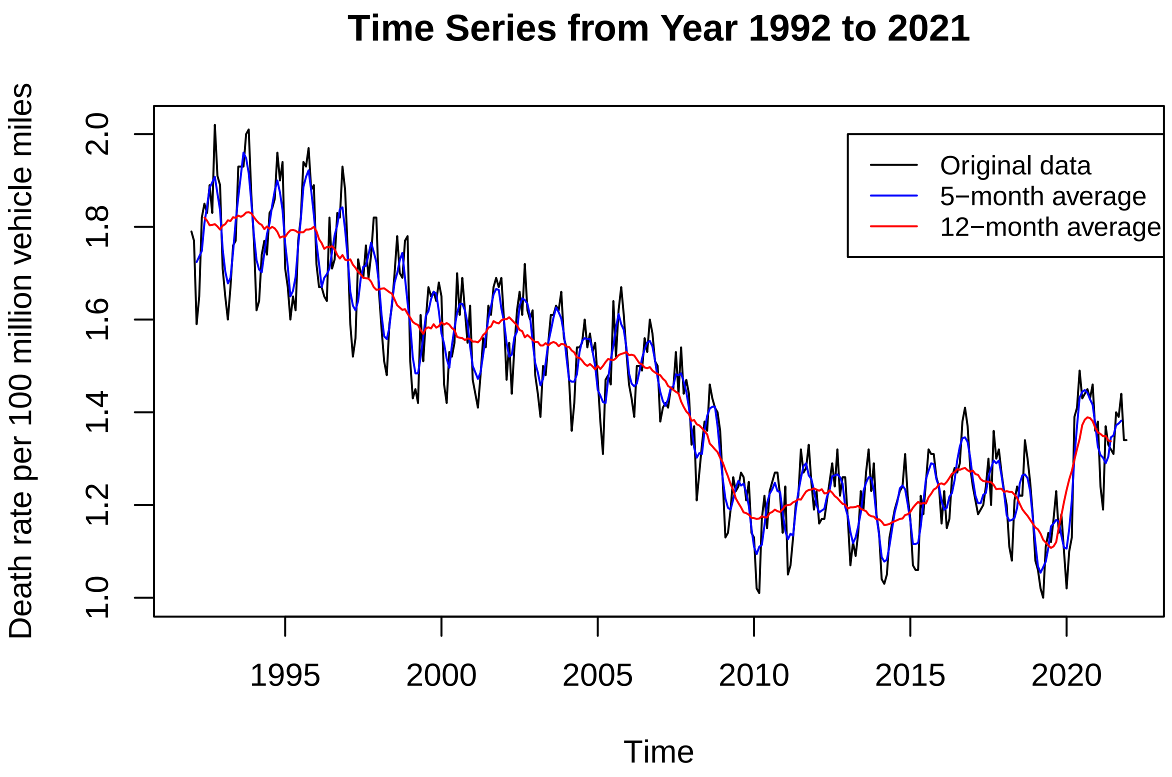


(c). We will cut the data starting at Jan 2009 and end at Dec 2018 and use the last three year(COVID-19 period) as testing data.

```
select <- c(1:(17*12))
data_split <- death_per_vehicle[-c(select)]
training = data_split[1:(10*12)]
testing = data_split[((10*12)+1):(13*12)]
```

(d). Compared to the aggregated method introduced in Lab 1, here we will create averages of different time intervals to reduce the roughness of the data.

```
library(zoo)
smooth_data_5m = zoo::rollmean(death_per_vehicle, k = 5, fill = NA)
smooth_data_12m = zoo::rollmean(death_per_vehicle, k = 12, fill = NA)
plot(death_per_vehicle, main = "Time Series from Year 1992 to 2021")
lines(smooth_data_5m, col='blue' )
lines(smooth_data_12m, col='red')
legend(2013, 2, legend=c("Original data","5-month average", "12-month average"),
   col=c("black" ,"blue", "red"), lty=1, cex=0.8)
```

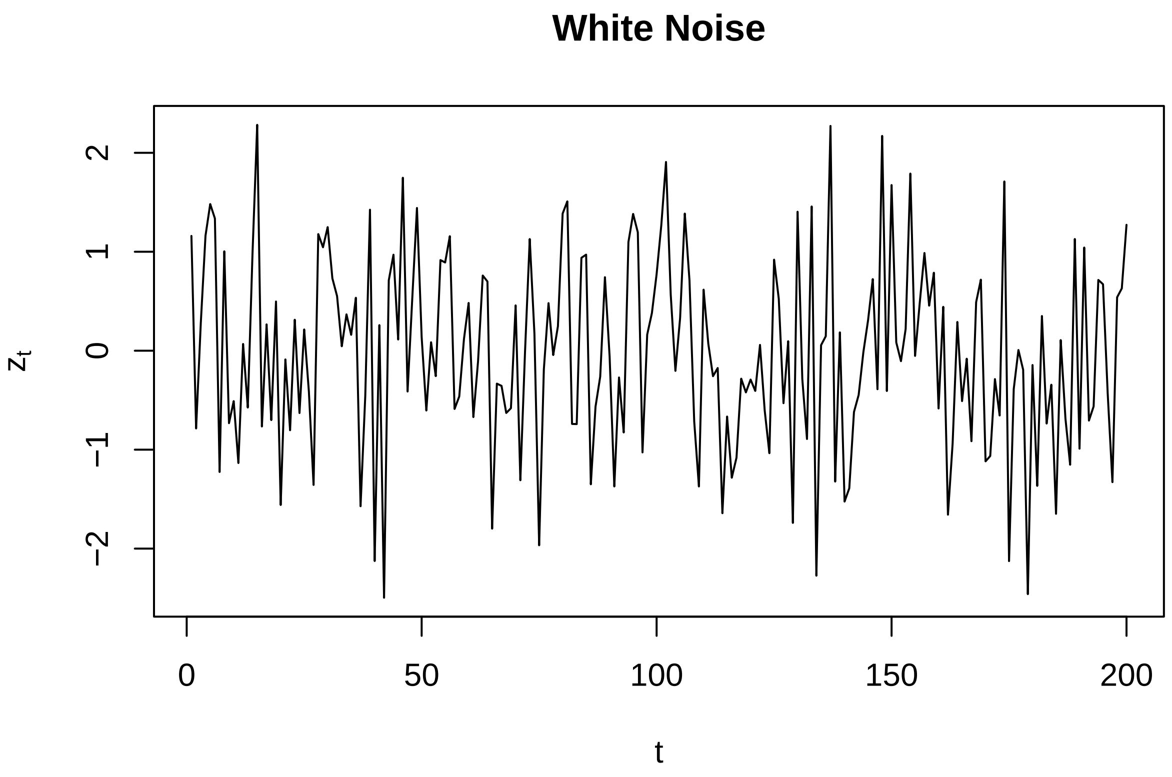


From the plot above, we can assess the roughness or smoothness of the data. By using longer time intervals, such as months, to create the average time series, we can achieve smoother data with reduced variance. For instance, the 12-month average data tends to be smoother compared to the 5-month average.

Characteristics of Time Series

2. **White noise.** Simulate and plot n = 500 values of a Gaussian white-noise process with variance ${\sigma_Z}^2 = 1$. i.e., $Z_t \overset{\text{iid}}{\sim} N(0, \sigma_Z^2)$.

```
z_t <- rnorm(200,0,1)
plot(z_t,xlab = "t",ylab = expression(z[t]),type = "l",main = "White Noise")
```

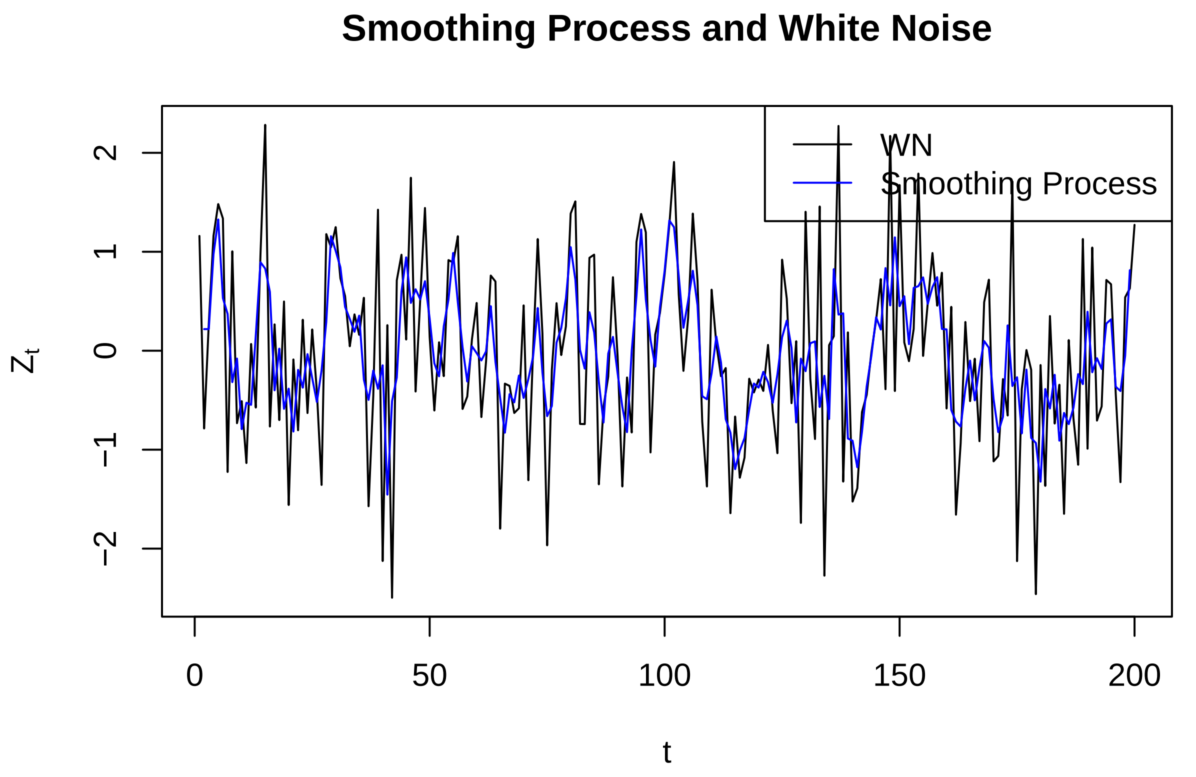


3. **Apply smoothing operation.** Using the above Gaussian process Z_t, use the filter command to construct a smoothing process of the form:

   $X_t = (Z_{t-1} + Z_t + Z_{t+1})/3$

   (notice that for this example the current value of X_t is the average of the previous, current and next values of Z_t). Then, plot X_t and Z_t together with different colors. Do you observe any difference?

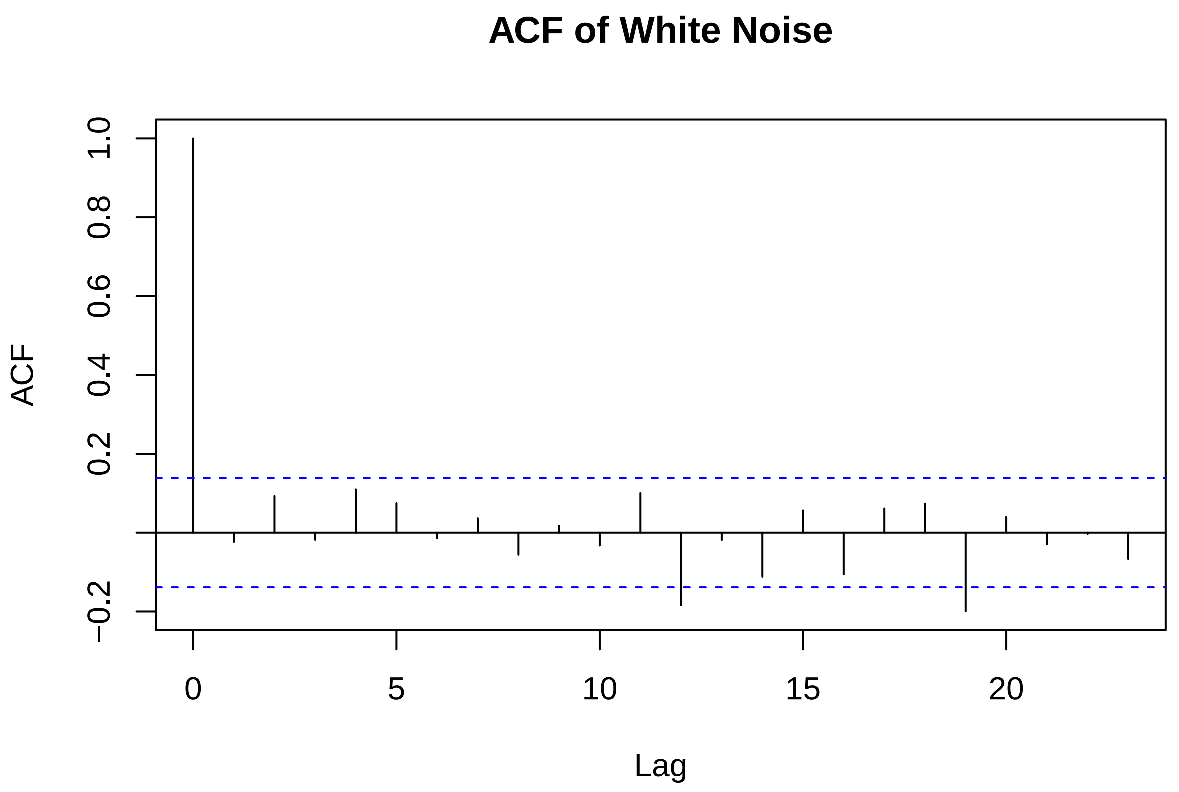
```r
# See help for filter
?filter
x_t = filter(z_t, filter = rep(1/3,3), sides = 2, method = "convolution")
# Plot of white-noise
plot(z_t,xlab = "t",ylab = expression(Z[t]),type = "l",
     main = "Smoothing Process and White Noise")
# Plot of Smoothing Process
lines(x_t,col = "blue")
# Add legend
legend("topright",c("WN", "Smoothing Process"),col = c("black","blue"),
       lty = 1)
```



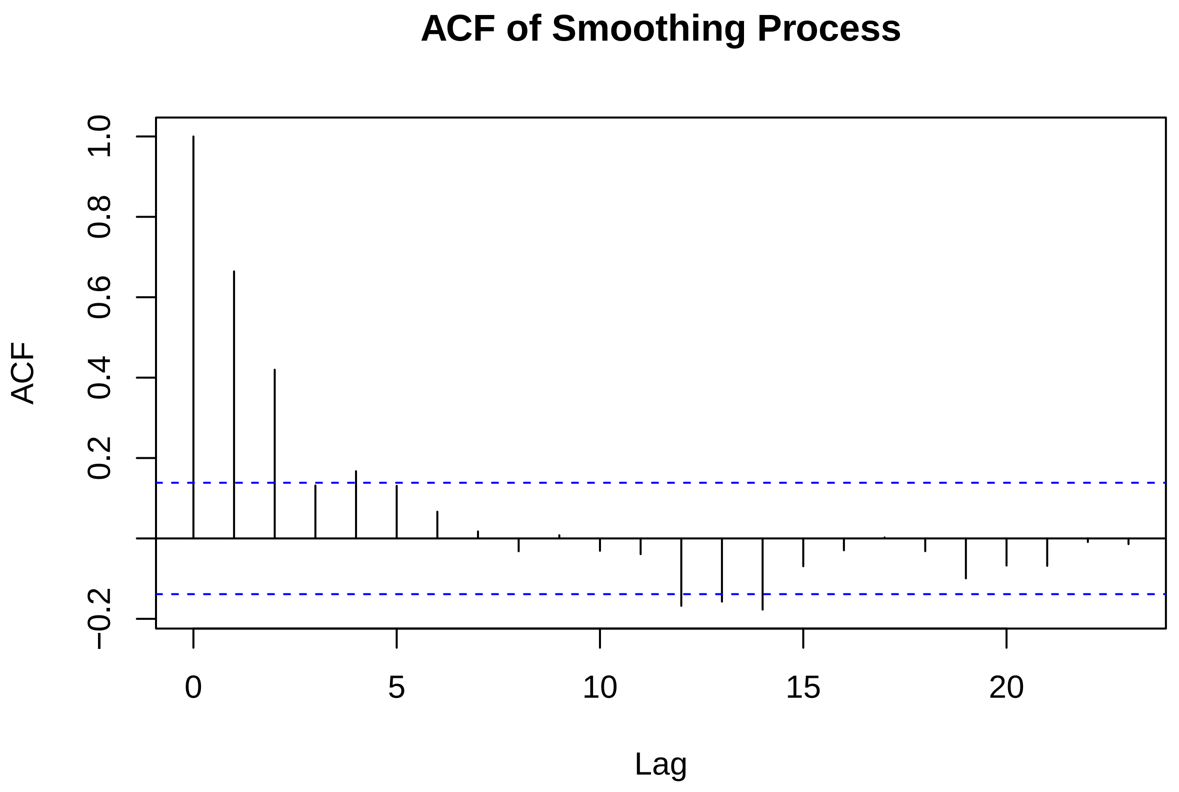
Compare smoothness of two paths: The graph of X is much smoother than that of Z because we take average of the previous, current and next values of Z_t

4. **Sample ACF.** Plot sample acfs of Z_t and X_t and compare the two plots.

```
acf(z_t, main="ACF of White Noise")
```



```r
acf(x_t,na.action = na.pass, main="ACF of Smoothing Process")
```



From the sample acf of Z_t. It represents White Noise. In comparison X_t, which indicates a MA(2) model.

5. **Simulation MA and AR models.**

   arima.sim in stats library is used to simulate observations from an ARIMA model.

   (a). Simulate MA model: arima.sim(n , model = list(ma=c(theta_1,theta_2)))

   (b). Simulate AR model: arima.sim(n , model = list(ar=c(theta_1,theta_2)))

   Here, n is the length of output series and model is a list with component ar and/or ma giving the AR and MA coefficients respectively.

   Example:

   (a). Simulate 1000 observations from MA(2): $X_t = Z_t + 0.3Z_{t-1} + 0.5Z_{t-2}$, where $Z_t \overset{\text{iid}}{\sim} N(0,1)$

```
ma_simulation = arima.sim(n=1000, model = list(ma=c(0.3,0.5)))
```

   (b). Simulate 1000 observations from MA(2): $X_t = 0.5X_{t-1} + Z_t$, where $Z_t \overset{\text{iid}}{\sim} N(0,1)$

```
ar_simulation = arima.sim(n=1000, model=list(ar=.5))
```

6. **Moving Average.**

   Consider an MA(2) process, given by

   $X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2}$, where $Z_t \overset{\text{iid}}{\sim} N(0,1)$

   Using R, simulate a time series of length 100 from an MA(2) process with the following values of the coefficients:

   a. $\theta_1 = 0.45$, $\theta_2 = 0.55$

   b. $\theta_1 = -0.45$, $\theta_2 = 0.55$

   For each simulated time series, plot the time series, sample ACF using acf and the theoretical ACF. What do you notice?

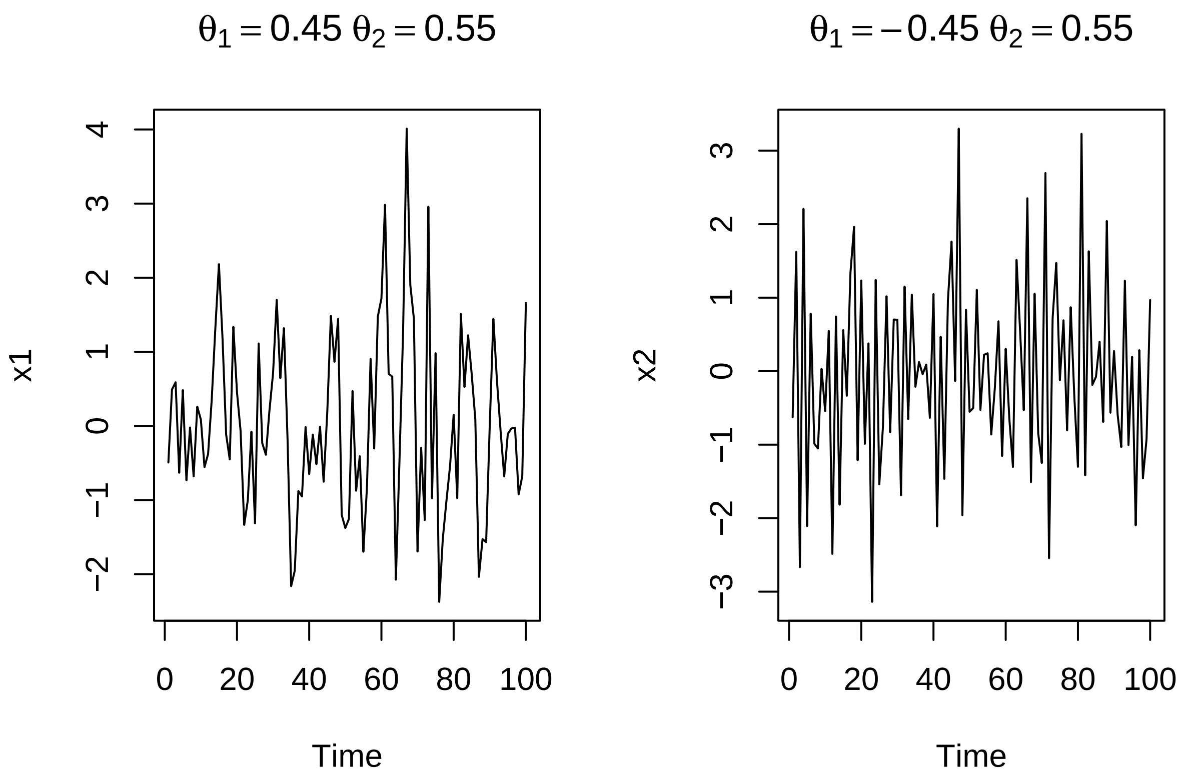
   Simulate the data and calculate theoretical ACFs.

```
#a
theta_1 <- 0.45
theta_2 <- 0.55
var_ma <- 1+theta_1^2+theta_2^2
# Simulate MA
x1 <- arima.sim(n = 100,model = list(ma=c(theta_1,theta_2)))
# Theoretical ACF
theo_acf1 <- c(var_ma,(theta_1 + theta_1*theta_2),theta_2,rep(0,18))/var_ma

#b
theta_1 <- -0.45
theta_2 <- 0.55
var_ma <- 1+theta_1^2+theta_2^2
x2 <- arima.sim(n = 100,model = list(ma=c(theta_1,theta_2)))
theo_acf2 <- c(var_ma,(theta_1 + theta_1*theta_2),theta_2,rep(0,18))/var_ma
```

   Plot the two time series.

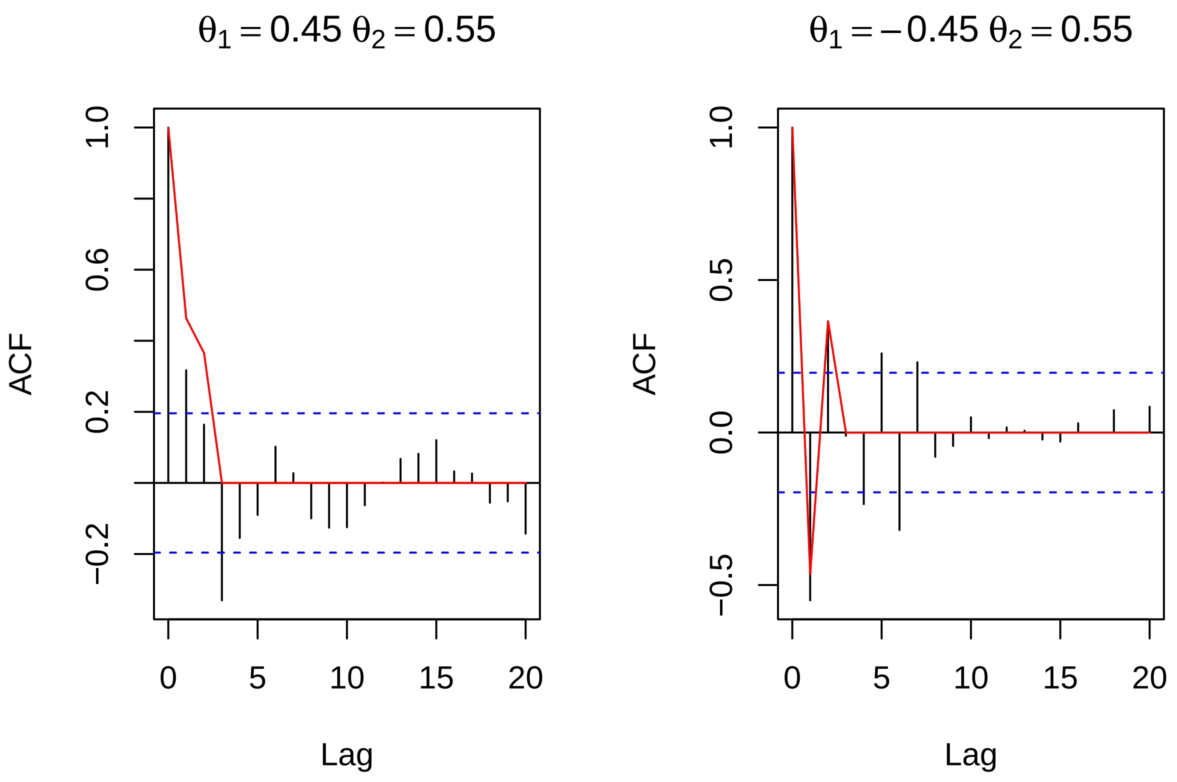
```
op <- par(mfrow = c(1,2))
#plot the time series x1 and x2
plot(x1, main = expression(theta[1] == 0.45~theta[2] == 0.55))
plot(x2, main = expression(theta[1] == -0.45~theta[2] == 0.55))
```



```
par(op)
```

Plot sample and theoretical ACFs.

```
# Plot both ACFs
op <- par(mfrow = c(1,2))
acf(x1,main = expression(theta[1] == 0.45~theta[2] == 0.55))
lines(x = 0:20,y = theo_acf1,col = "red")
# Sample auto-correlation
# Add theoretical ACF
acf(x2,main = expression(theta[1] == -0.45~theta[2] == 0.55))
lines(x = 0:20,y = theo_acf2,col = "red")
```



```r
par(op)
```

The sample ACFs are close to the theoretical ACFs. After Lag 2, theoretical ACFs become 0 and sample ACFs are mainly within 95 percent confidence interval.

7. **Random Walk Process.**

   a. Show that the random walk

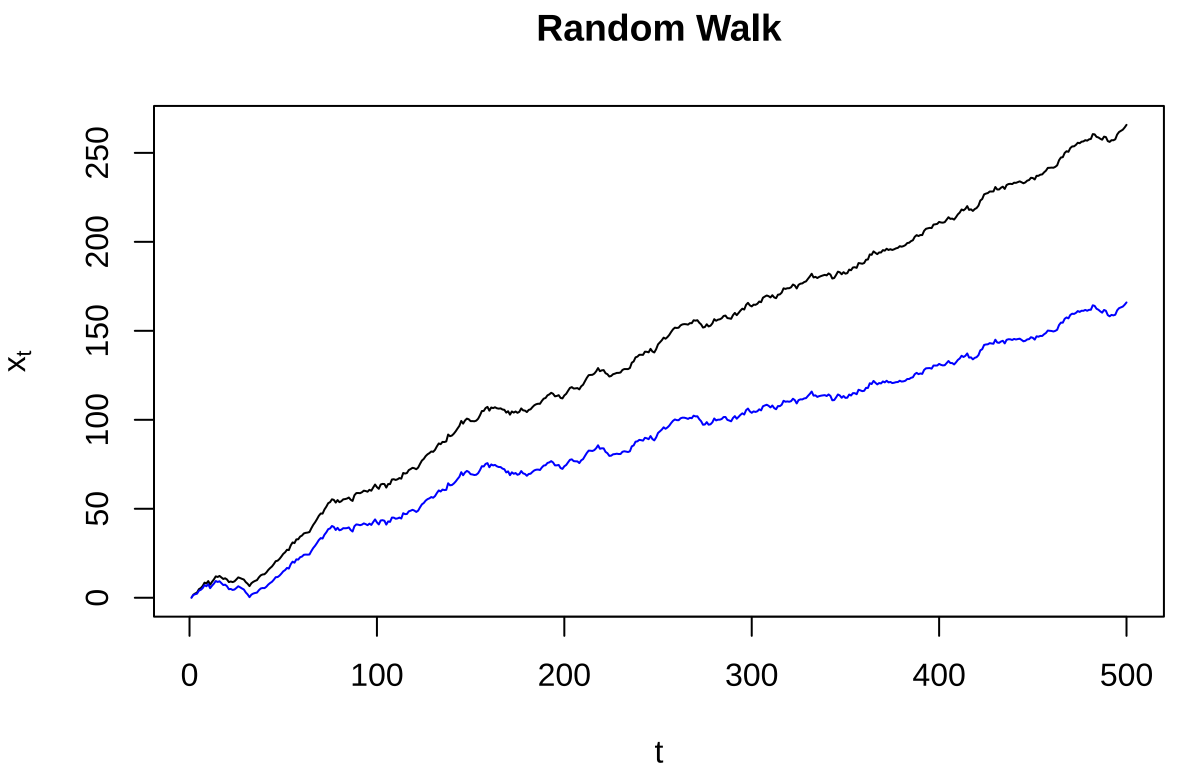
      $X_t = \delta + X_{t-1} + Z_t$, where $Z_t \overset{\text{iid}}{\sim} WN(0, {\sigma_Z}^2)$

      can be re-written as the cumulative sum of white noise variates: $X_t = \delta t + X_0 + \sum_{j=1}^{t} Z_j$

   b. Simulate n = 200 observations of a random-walk with drift $\delta = 0.6, 0.4$; initial condition $X_0 = 0$ and ${\sigma_Z}^2 = 1$. Then, plot both realizations using different colors.

   c. Is the random-walk with drift δ a (weakly) stationary process?

```r
z_t <- rnorm(499,0,1)
x_t <- c(0,cumsum(z_t))
rw1 <- 0:499*0.6 + x_t
rw2 <- 0:499*0.4 + x_t
plot(rw1,type = "l",xlab = "t",ylab = expression(x[t]),
main = "Random Walk")
lines(rw2,col = "blue",type = "l")
```



From the plot, both time series processes are non-stationarity becuase the mean increases over time.