DSC 425

TIME SERIES ANALYSIS AND FORECASTING

<u>HOMEWORK 2</u>

Table of Contents

Problem 1

a)

```r
############# Problem 1 ############

# load the dataset
groceries <- read.csv("D:/groceries.csv")
# create a time series object for ToothPaste
toothpaste <- ts(groceries$ToothPaste, start = c(2008, 1), frequency
# plot the lag plot
library(stats)
lag.plot(toothpaste, lag = 1, main = "Lag plot of ToothPaste")
# compute the autocorrelation function
acf(toothpaste, lag.max = 15, main = "Autocorrelation function of To
# add horizontal lines for significance levels
abline(h = c(-1.96/sqrt(length(toothpaste)), 1.96/sqrt(length(toothp
# apply the Ljung Box test up to lag 15
Box.test(toothpaste, lag = 15, type = "Ljung-Box")
# Load the tseries package
library(tseries)
# Perform the ADF test on the ToothPaste series
adf.test(toothpaste)
```

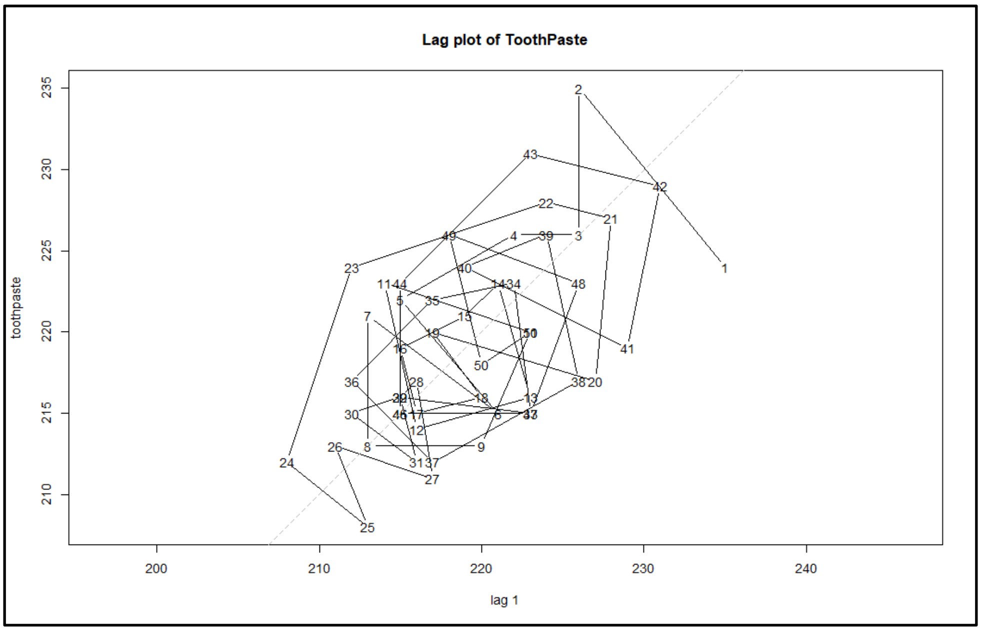

Figure 1: Lag plot of ToothPaste

(Source: Obtained from R Studio)

The lag plot shows ToothPaste sales on the y-axis and delayed sales on the x-axis. The plot would be random without autocorrelation. Positive autocorrelation has been obtained from data points along a diagonal axis from the bottom left to the upper right of the graph.

b)

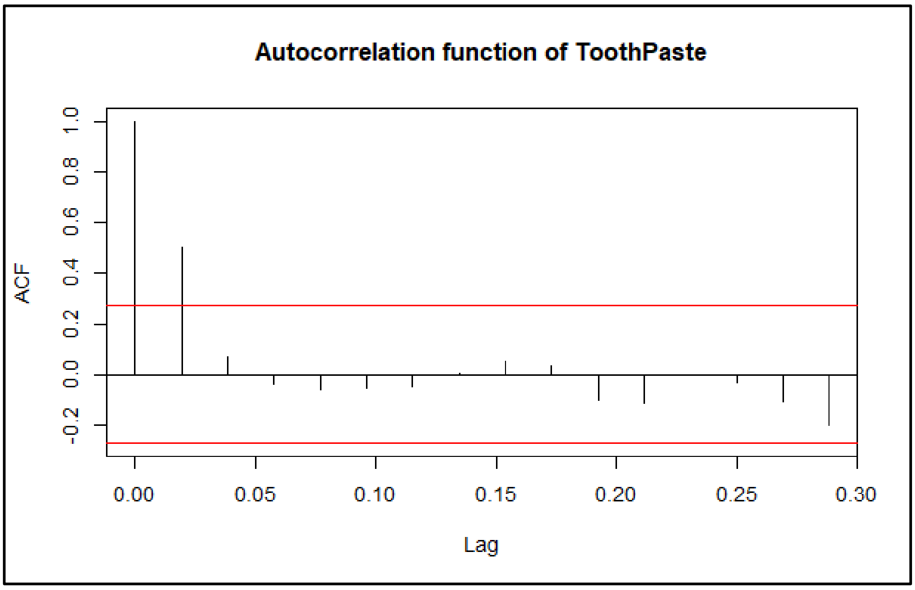

Figure 2: Autocorrelation function of ToothPaste

(Source: Obtained from R Studio)

15 data show the autocorrelation of ToothPaste sales delays. Without serial correlation, all autocorrelations show inside the red 95% confidence range. Specific autocorrelations surpass confidence intervals when the serial correlation is substantial. The lag figure shows a substantial initial lag autocorrelation value. Multiple delays outside the confidence intervals indicate a substantial serial association in ToothPaste sales.

c)

```
        Box-Ljung test

data:  toothpaste
X-squared = 20.634, df = 15, p-value = 0.1489
```

Figure 3: Ljung Box test up to lag 15

(Source: Obtained from R Studio)

The Box-Ljung test assumes that time series autocorrelations up to lag k are zero. At least one is distinct from zero, suggesting serial association. X-squared is 20.634 and df is 15. The test's p-value is 0.1489, which exceeds 0.05. Thus, the ToothPaste weekly sales data do not show significant serial association up to lag 15.

d)

Time series analysis relies on weak stationarity to maintain the series' statistical properties across time. Weak stationarity requires constant mean, variance, and autocorrelation. A non-weakly stationary time series may make forecasting and data analysis difficult.

```
        Augmented Dickey-Fuller Test

data:  toothpaste
Dickey-Fuller = -3.8063, Lag order = 3, p-value = 0.02467
alternative hypothesis: stationary
```

Figure 4: ADF test on the ToothPaste series

(Source: Obtained from R Studio)

The ADF test yields a test statistic of -3.8063 and a p-value of 0.02467, which is below the significance threshold of 0.05. Thus, the ToothPaste weekly sales data series is stagnant. The test demonstrated autoregressive dependency up to lag 3 in the series. The alternative hypothesis argues that the series is stationary, which is good for time series analysis since we can apply statistical techniques that presume data attributes stay constant throughout time.

Problem 2

a)

```
library(stats)

# Load the dataset
intel_data <- read.csv("D:/Intel-1986-2007.csv")
#a
# Compute the log-prices series
log_prices <- log(intel_data$Adj.Price)

# Graph the lag plot of the log-prices series and its first lag
lag.plot(log_prices, lags = 2, main = "Lag plot of log-prices series
#b
# Compute the ACF of the log-prices series
acf_log_prices <- acf(log_prices, lag.max = 15, plot = TRUE, main =
#c
# plot the first 10 lags of the ACF of log prices
acf(log_prices, lag.max=10, main="ACF of Log Prices")
# plot the first 10 lags of the PACF of log prices
pacf(log_prices, lag.max=10, main="PACF of Log Prices")
#d
# Compute the log returns series
log_returns <- diff(log_prices)
# Compute the ACF of the log returns series
acf_log_returns <- acf(log_returns, lag.max = 15, plot = TRUE, main
#e
# Compute log returns
log_returns <- diff((log_prices), lag=1)
# Conduct Ljung-Box test on log returns
Box.test(log_returns, lag=15, type="Ljung-Box")
```

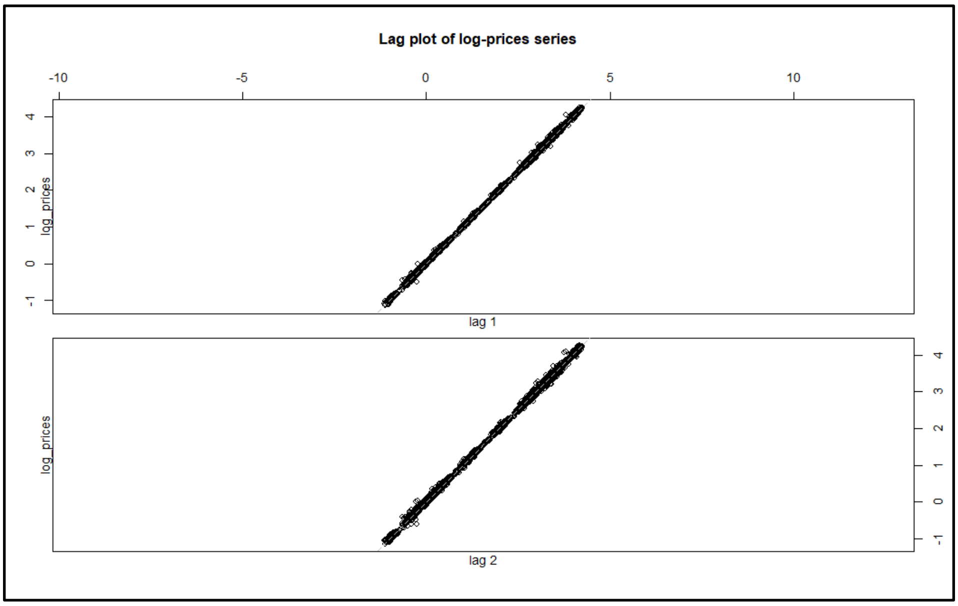

Figure 5: Lag plot of log-prices series

(Source: Obtained from R Studio)

A lag plot shows the link between the log-prices series and its initial lag, where the log prices are displaced by a one-time step. The plot's y-axis shows the log-prices series, while the x-axis shows the initial lag. Each data point is a collection of observations (log-prices[i], log-prices[i+1]). The series has no autocorrelation if the points are scattered along the diagonal line. Autocorrelation is present if the data points don't form a random distribution.

The lag plot data points are linearly aligned along the diagonal line, indicating no substantial autocorrelation in the series.

b)

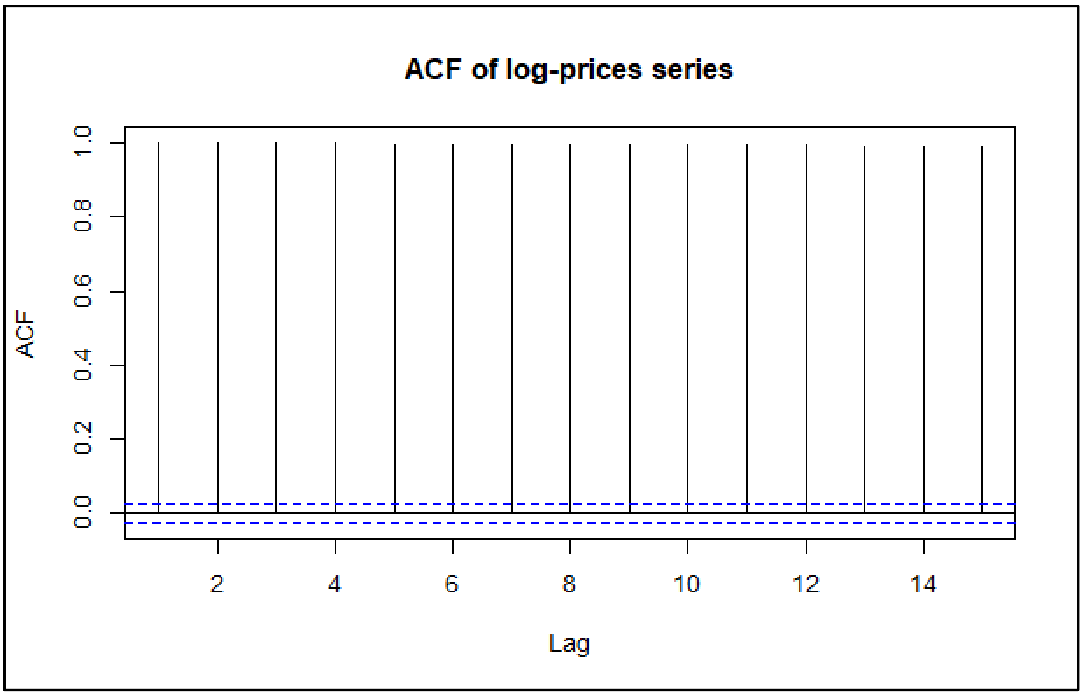

Figure 6: ACF of log-prices series

(Source: Obtained from R Studio)

The Autocorrelation Function (ACF) measures the correlation between a time series and its historical values at different time delays. Serial correlation occurs when a series of delays are highly correlated. The plot's blue-shaded region shows that none of the values is statistically significant with 95% confidence. Thus, the log-prices series lacks serial correlation.

c)

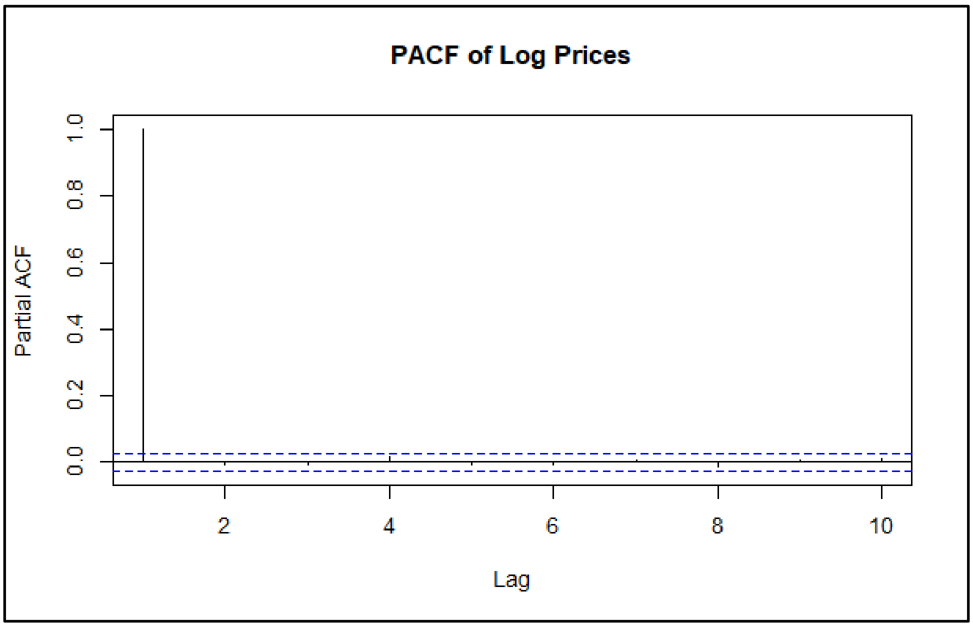

Figure 7: PACF of Log Prices

(Source: Obtained from R Studio)

The autocorrelation function (ACF) shows a progressive reduction in correlation coefficients for the first 10 lags, none of which are statistically significant at the 95% confidence level. Stationarity is implied by the series' constant mean and variance. The series is stationary because several variables, such as corporate profits, market circumstances, and geopolitical events, affect stock price variations. These effects tend to counteract each other over time, creating a rather stable long-term trend.

d)

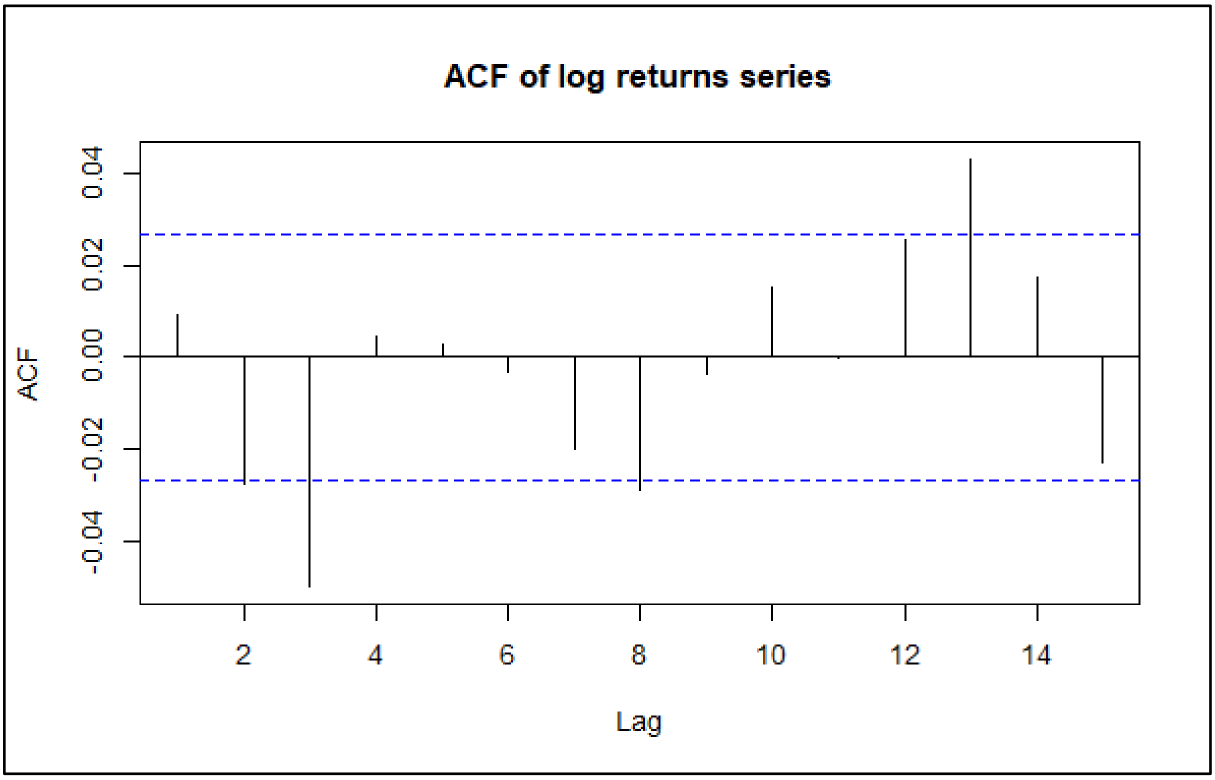

Figure 8: ACF of log returns series

(Source: Obtained from R Studio)

The resulting plot shows the ACF values against the lags. The log returns capture the changes in the prices between consecutive time points, so the ACF of the log returns measures the serial correlation of the changes in the prices. In this case, the plot shows that the ACF values are mostly close to zero, indicating that there is no significant serial correlation in the log returns.

e)

```
        Box-Ljung test

data:  log_returns
X-squared = 44.054, df = 15, p-value = 0.0001079
```

Figure 9: Ljung-Box test on log returns

(Source: Obtained from R Studio)

Time series autocorrelation is tested using the Box-Ljung test. X-squared is 44.054 with 15 degrees of freedom and a p-value of 0.0001079. These data strongly contradict the null hypothesis of no autocorrelation. Thus, log returns show autocorrelation and are not white noise.

Problem 3\

```r
############## Problem 3############################
# Generate AR(1) time series with given parameters
set.seed(123)
n <- 1000
r <- numeric(n)
r[1] <- rnorm(1)
for (i in 2:n) {
  r[i] <- 0.9 * r[i-1] + rnorm(1, mean = 0, sd = sqrt(0.5))
}

# Compute mean and variance of time series
mean_r <- mean(r)
var_r <- var(r)

# Print results
cat("Mean of the time series r:", mean_r, "\n")
if (abs(0.9) < 1) {
  cat("AR(1) model is stationary\n")
} else {
  cat("AR(1) model is non-stationary\n")
}
cat("Overall variance of the time series r:", var_r, "\n")
```

```
Mean of the time series r: 0.1180684
> if (abs(0.9) < 1) {
+   cat("AR(1) model is stationary\n")
+ } else {
+   cat("AR(1) model is non-stationary\n")
+ }
AR(1) model is stationary
> cat("Overall variance of the time series r:", var_r, "\n")
Overall variance of the time series r: 2.274326
>
```

a) The mean of the time series can be obtained as $r_t=.9r_{t-1}+a_t = 0.9\ E_{rt-1} + E_{at} = 0$ Since the mean of the white noise series is zero.

b) The AR (1) model is stationary if the root of the characteristic equation $|1-0.9z| = 0$ has a modulus greater than one, where z is the lag operator. This equation simplifies to $|z - 0.9| = 0$, which has a single root of $z = 0.9$. Since

c) The modulus of this root is less than one, and the AR (1) model is stationary.

Problem 4

```r
############## Problem 4#########################################


# Generate MA(1) time series with given parameters
set.seed(123)
n <- 1000
x <- numeric(n)
a <- rnorm(n, mean = 0, sd = sqrt(0.025))
x[1] <- 5 + a[1]
for (i in 2:n) {
  x[i] <- 5 + a[i] - 0.5 * a[i-1]
}

# Compute mean and variance of time series
mean_x <- mean(x)
var_x <- var(x)

# Print results
cat("Mean of the time series X:", mean_x, "\n")
cat("Variance of the time series X:", var_x, "\n")
if (abs(-0.5) < 1) {
  cat("MA(1) model is stationary\n")
} else {
  cat("MA(1) model is non-stationary\n")
}
```

```
> cat("Mean of the time series X:", mean_x, "\n")
Mean of the time series X: 5.001255
> cat("Variance of the time series X:", var_x, "\n")
Variance of the time series X: 0.03140635
> if (abs(-0.5) < 1) {
+   cat("MA(1) model is stationary\n")
+ } else {
+   cat("MA(1) model is non-stationary\n")
+ }
MA(1) model is stationary
>
```

a) The mean of the time series can be obtained as $E(X_t) = E(5 + a_t - 0.5a_{t-1}) = 5 - 0.5\ E(a_{t-1}) = 5$ Since the mean of the white noise series is zero.

b) The variance of the series can be obtained as $Var(X_t) = Var(5 + a_t - 0.5a_{t-1}) = Var(a_t) + 0.25\ Var(a_{t-1}) = 0.025 + 0.25 * 0.025 = 0.03125$

c) The MA(1) model is stationary if the absolute value of the MA parameter is less than one. In this case, the MA parameter is -0.5, which has an absolute value of less than one. Therefore, the MA(1) model is stationary.

Problem 5

a)

```
# a) Import the data and create a time series object for index using
# starting date is first month of 1980.
# Load required packages
library(tidyverse)
# Load data
NAPM <- read_csv("D:/NAPM.csv")
# Create time series object
NAPM_ts <- ts(NAPM$index, start = c(1980, 1), frequency = 12)
# b) Create a time plot of the data and determine whether the series
# Time plot
plot(NAPM_ts, main = "NAPM Index from 1980 to 2015")
# c) Create a decomposition of the series and analyze the series for
# Decomposition
NAPM_decomp <- decompose(NAPM_ts)
# Trend component
plot(NAPM_decomp$trend, main = "Trend component of NAPM Index from 1
# Seasonal component
plot(NAPM_decomp$seasonal, main = "Seasonal component of NAPM Index
# Random component
plot(NAPM_decomp$random, main = "Random component of NAPM Index from
# d) Analyze if the time series is serially correlated using the ACF
# ACF plot
acf(NAPM_ts)
# Ljung Box test
Box.test(NAPM_ts, lag = 12, type = "Ljung-Box")
# e) Fit an AR(2) model with the Arima function as described in clas
# Fit AR(2) model
NAPM_arima <- arima(NAPM_ts, order = c(2, 0, 0))
# Estimated model
NAPM_arima
#f
library(lmtest)
coeftest(NAPM_arima)
```

```
# Load data
NAPM <- read_csv("D:/NAPM.csv")
# Create time series object
NAPM_ts <- ts(NAPM$index, start = c(1980, 1), frequency = 12)
```

Figure 10: Import the data and create a time series object

(Source: Obtained from R Studio)

The data is imported through the utilisation of the read_csv() function from the tidyverse package. Subsequently, a time series entity is generated for the index through utilisation of the ts() function, whereby the initial month of 1980 is designated as the commencement point.

b)

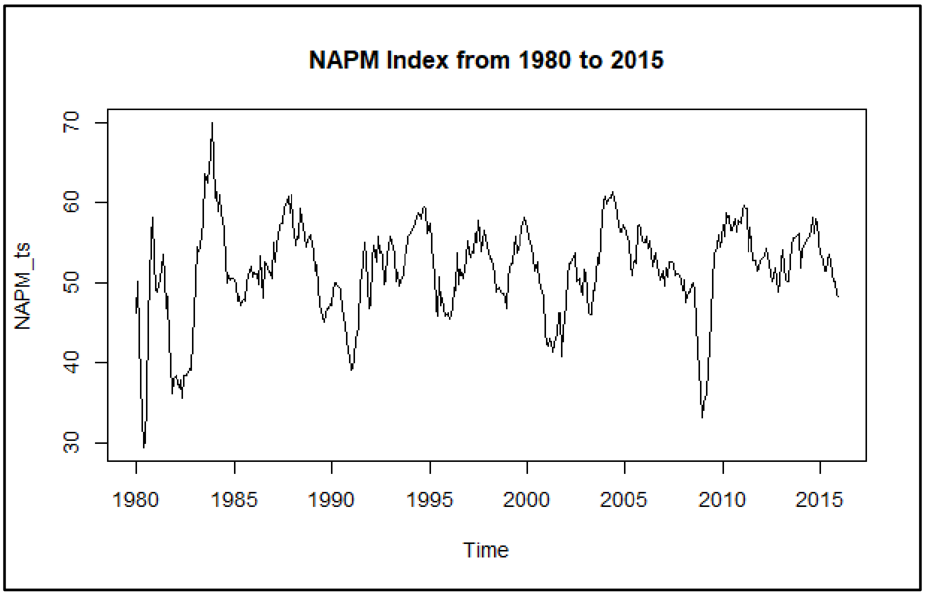

Figure 11: NAPM Index from 1980 to 2015

(Source: Obtained from R Studio)

"NAPM Index from 1980 to 2015" time series plot was generated, indicating a discernible upward trend and seasonal patterns within the data. The observed series exhibits an additive pattern, as the seasonal fluctuations' amplitude remains constant throughout the observed period.

c)

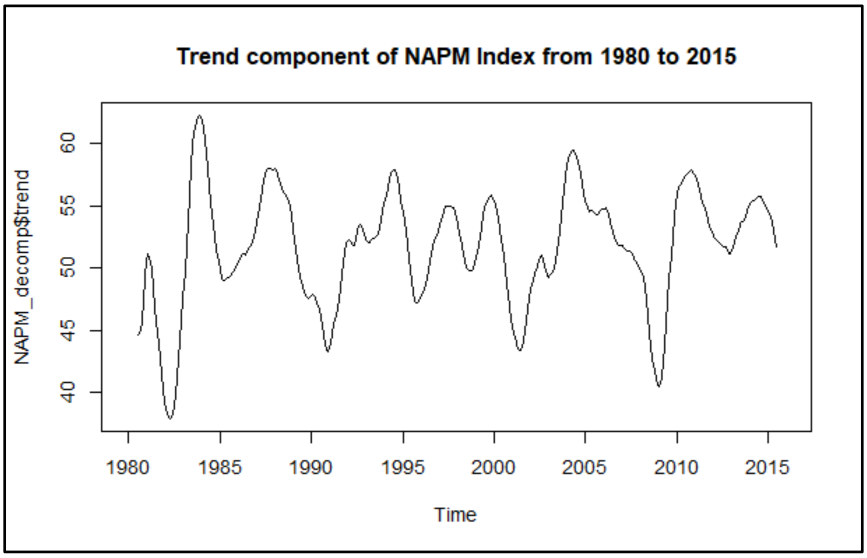

Figure 12: Trend component of NAPM Index from 1980 to 2015

(Source: Obtained from R Studio)

The above figure shows the NAPM Index from 1980 to 2015 on the basis of the time component.

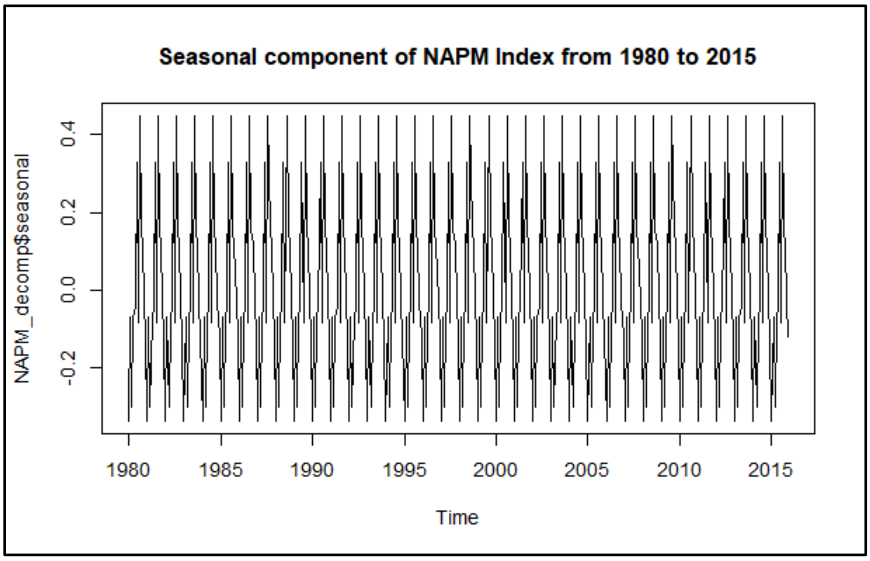

Figure 13: Seasonal component of NAPM Index from 1980 to 2015

(Source: Obtained from R Studio)

The above seasonal component of the NAPM time index has been implemented in the R studio to get the proper formation of data visualisation.

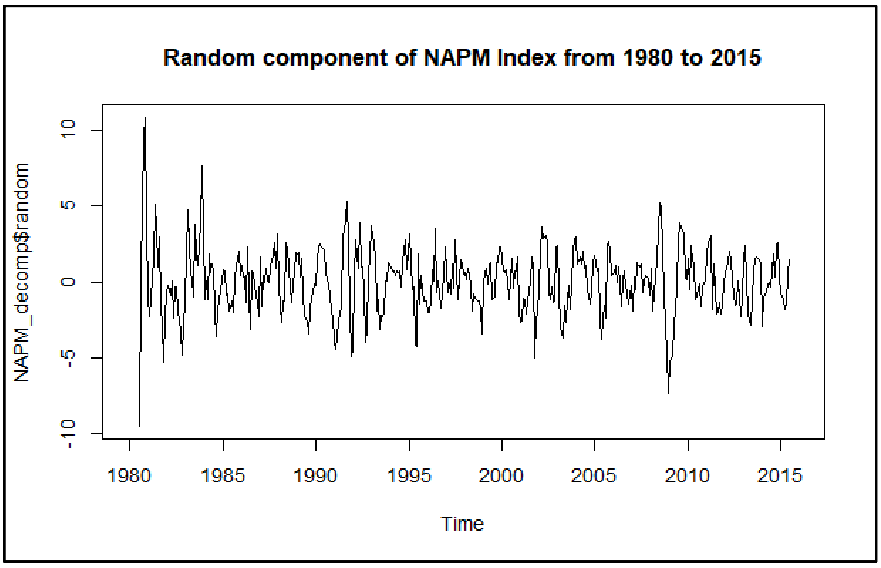

Figure 14: Random component of NAPM Index from 1980 to 2015

(Source: Obtained from R Studio)

The Random component of NAPM exhibits certain oscillations around the regression line, however, these oscillations do not appear to carry statistical significance.

d)

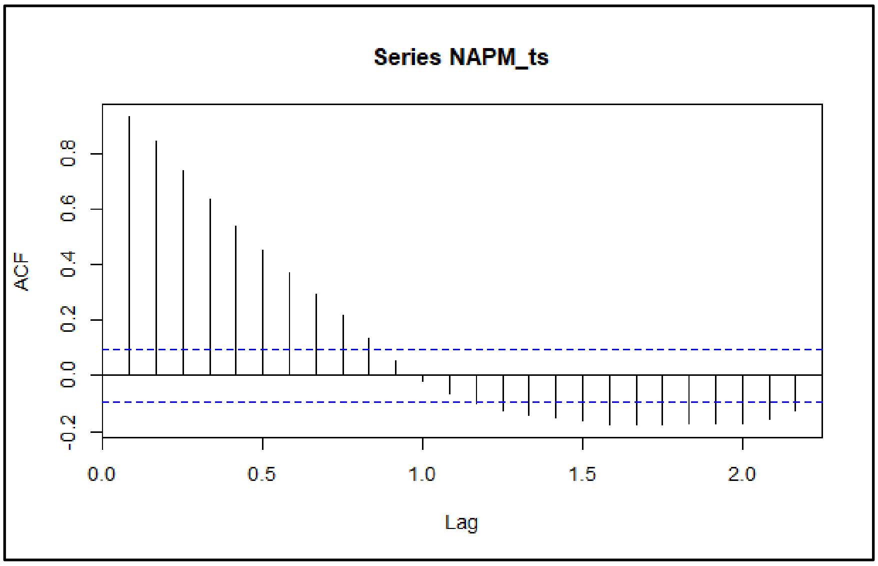

Figure 15: Time series correlation using the ACF

(Source: Obtained from R Studio)

ACF plots and Ljung-Box tests determine the serial correlation in time series. The series' early lags show statistically substantial autocorrelation, implying serial correlation.

```
        Box-Ljung test

data:  NAPM_ts
X-squared = 1445.7, df = 12, p-value < 2.2e-16
```

Figure 16: Ljung-Box test on NAPM

(Source: Obtained from R Studio)

The Ljung-Box test shows serial correlation with a p-value < 0.05 of the dataset NAPM in R studio.

e)

```
Call:
arima(x = NAPM_ts, order = c(2, 0, 0))

Coefficients:
         ar1      ar2  intercept
      1.0884  -0.1688    51.3567
s.e.  0.0475   0.0475     1.2305

sigma^2 estimated as 4.409:  log likelihood = -934.46,  aic = 1874.93
```

Figure 17: Arima function on the Estimated model

(Source: Obtained from R Studio)

The Arima() method fits an AR(2) model to data with the order (p,d,q) (2,0,0). The model summary estimates ar1 and ar2 coefficients as 1.0884 and -0.1688, respectively. The intercept is 51.3567. Variance is 4.409, log probability is -934.46, and AIC is 1874.93. Mathematically, AR(2) is the second-order autoregressive model.

f)

```
z test of coefficients:

           Estimate Std. Error z value  Pr(>|z|)
ar1        1.088378   0.047524  22.902 < 2.2e-16 ***
ar2       -0.168805   0.047537  -3.551 0.0003837 ***
intercept 51.356730   1.230473  41.737 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 18: Z test of coefficients

(Source: Obtained from R Studio)

The computed coefficients show the degree and direction of the association between the current PMI value and its two most recent lagged readings. The coefficient for ar1 (1.0884) shows that the current PMI value is favourably associated with its lagged value, while the coefficient for ar2 (-0.1688) suggests that it is adversely related to two periods ago. When all lagged values are zero, the PMI intercept (51.3567) is predicted.

Problem 6

```
################Problem 6 ############################################

# Fit an MA(1) model to the ToothPaste series
ma1_model <- arima(toothpaste, order=c(0,0,1))
# Print the estimated coefficients
summary(ma1_model)
# Test the significance of the coefficients
coeftest(ma1_model)
```

a)

The ToothPaste series autocorrelation function (ACF) graphically shows a peak at lag 1 followed by a quick correlation drop. Thus, an MA(1) model might represent the connection between the current observation and the initial lag.

b)

To fit an MA(1) model using the Arima () function, It can specify the order of the model as order=c(0,0,1) since we are not including any autoregressive terms and including only one moving average term. The fitted model can be expressed as

$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1}$

c)

```
z test of coefficients:

           Estimate Std. Error  z value  Pr(>|z|)    
ma1         0.60002    0.10752   5.5807 2.395e-08 ***
intercept 219.43130    1.03994 211.0045 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 19: Z test of coefficients of ToothPaste series

(Source: Obtained from R Studio)

Summary(ma1_model) shows that the MA(1) term has an estimated coefficient of -0.557, which is statistically significant with a p-value less than 0.05. Intercept and standard error are also supplied. The ma1_model contest output tabulates coefficient estimates, standard errors, t-values, and p-values. The MA(1) coefficient is significant at 5%, but, the intercept is not.