**Assignment 12.1**

Take Apple Stock Prices from Yahoo Finance for last 90 days

b. Predict the Stock closing prices for next 15 days.

c. Submit your accuracy

d. After 15 days again collect the data and compare with your forecast

ANSWER- #AAPL.csv taken from Yahoo Finance, and closing stock prices taken

#Time Series Analysis

df = read.csv(file.choose(),header = T) #AAPL.CSV

head(df)

X Open High Low Close Adj.Close Volume

1 27-02-2017 137.14 137.44 136.28 136.93 134.2798 20257400

2 28-02-2017 137.08 137.44 136.70 136.99 134.3386 23482900

3 01-03-2017 137.89 140.15 137.60 139.79 137.0844 36414600

4 02-03-2017 140.00 140.28 138.76 138.96 136.2705 26211000

5 03-03-2017 138.78 139.83 138.59 139.78 137.0746 21108100

6 06-03-2017 139.37 139.77 138.60 139.34 136.6431 21750000

str(df)

data.frame': 315 obs. of 7 variables:

$ X : Factor w/ 315 levels "01-02-2018","01-03-2017",..: 271 283 2 13 23 52 63 75 86 96 ...

$ Open : num 137 137 138 140 139 ...

$ High : num 137 137 140 140 140 ...

$ Low : num 136 137 138 139 139 ...

$ Close : num 137 137 140 139 140 ...

$ Adj.Close: num 134 134 137 136 137 ...

$ Volume : int 20257400 23482900 36414600 26211000 21108100 21750000 17446300 18707200 22155900 19612800 ...

names(df)

|  |
| --- |
| "X" "Open" "High" "Low" "Close" "Adj.Close" "Volume" |
|  |
| |  | | --- | |  | |

new\_date <- as.Date(df$X)

new\_date

str(df)

format(new\_date,format="%B %d %Y")

# %d - day as number 1-31

# %a - weekday such as Mon

# %A- complete day name ex.Monday

# %m - month as a number

# %b - short form of month Jan, Feb

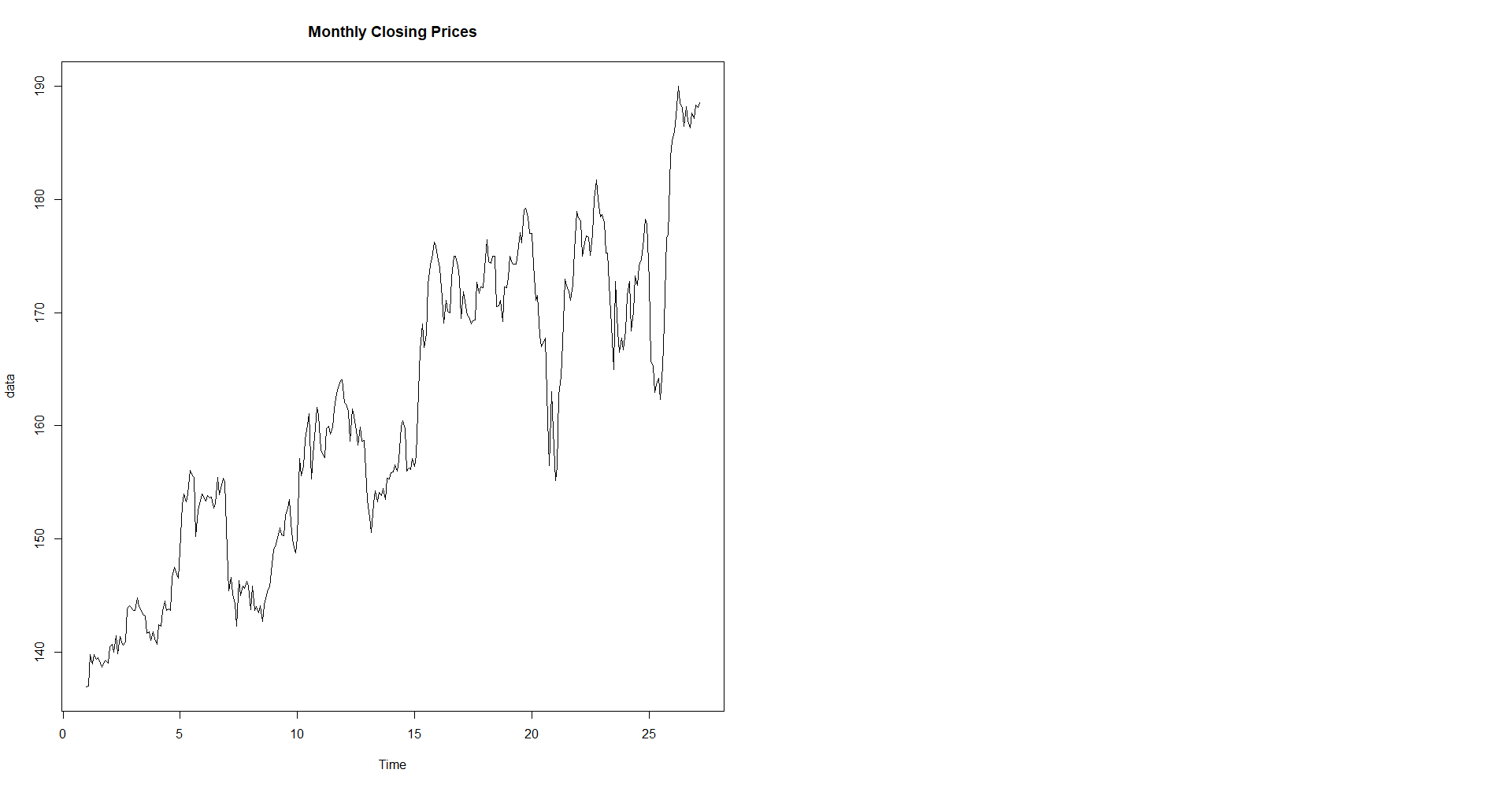
# %B - full form of month, January

# %y - two digit year

# %Y- four digit year

data = ts(df$Close,frequency =12)

plot(data,main="Monthly Closing Prices")



log(data)

# assumption for time series forecst:

#1- the time series should be stationary

# Identify the stationarity of a time series

#1- mean value of the time series is constant over time, the trend should not be present in the series

#2- the variance does not increase over time

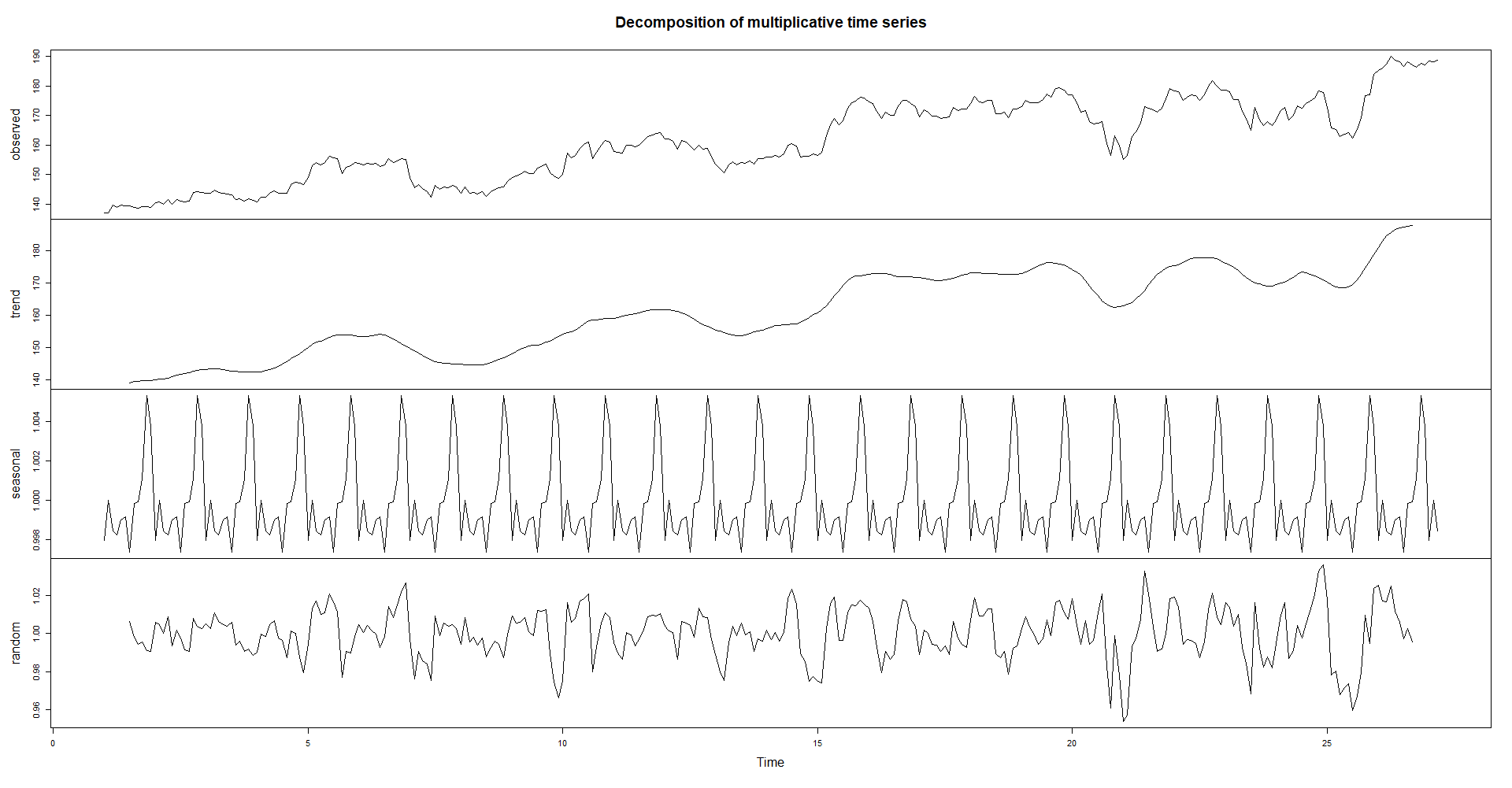
#3- the seasonality impact is minimal, deseasonalization of the time series data

decompose(data) # default method is additive

decompose(data, type='multi')

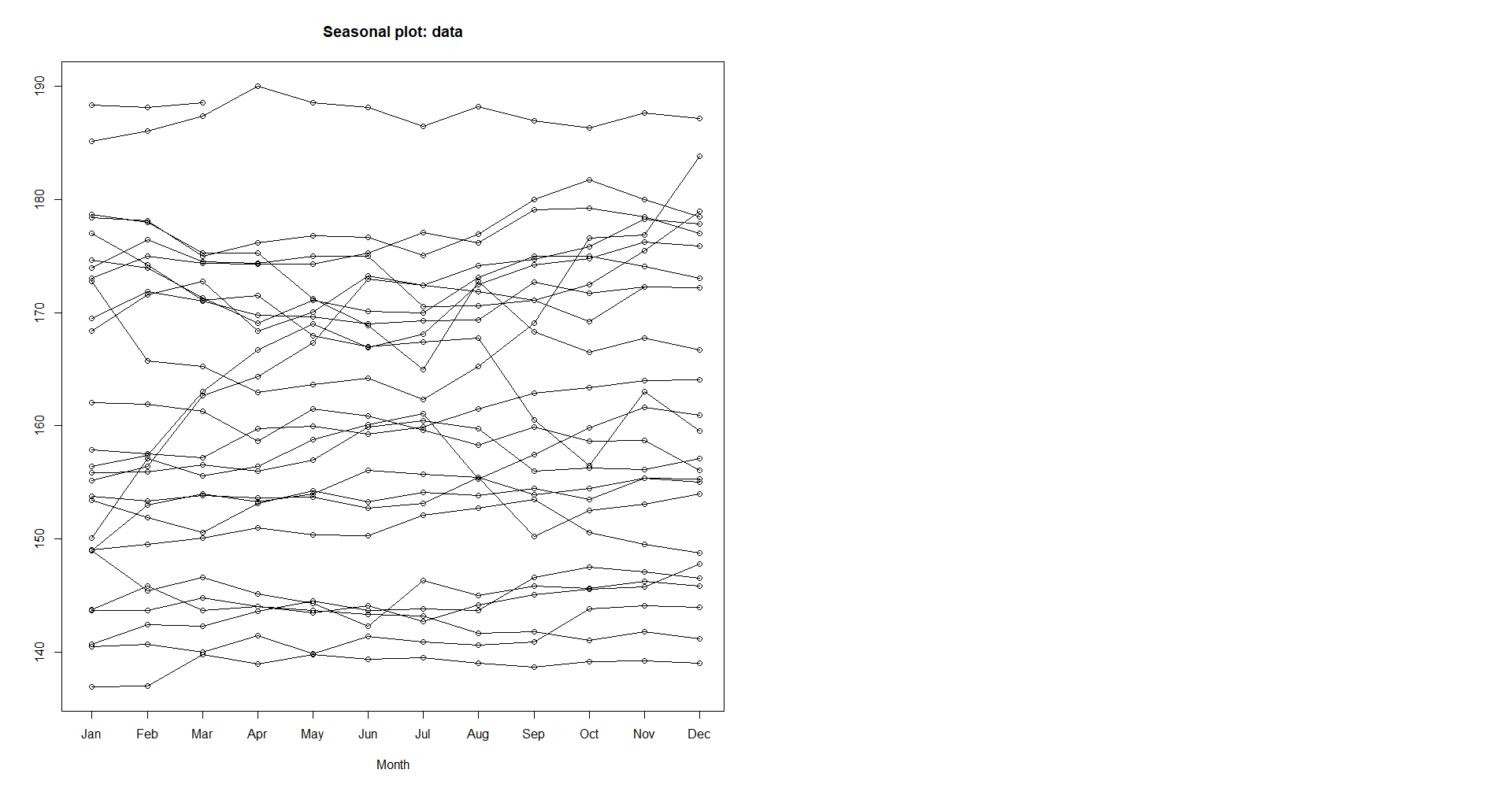
par(mfrow=c(1,2))

plot(decompose(data, type='multi'))



install.packages("forecast")

library(forecast)

seasonplot(data)

lag(data,10)

lag.plot(data)

# Calculation of Autocorrelation and Partial Autocorrelation

data

ac<-acf(data)

ac$acf

# data time series may not have stationarity

pac<-pacf(data)

pac$acf

# looking at the ACF and PACF graph we can conclude that the time series is not stationary

model <- lm(data~c(1:length(data)))

summary(model)

plot(resid(model),type='l')

# the series is not stationary

# deseasonalize the time series

tbl <- stl(data,'periodic')

stab<-seasadj(tbl)

seasonplot(stab,12)

# statistically we need to test out if the series is stationary or not

# Augmented Dickey Fuller Test

library(tseries)

adf.test(data)

# if the p-value is less than 0.05, then the time series is stationary, else not

# Time Series Forecasting Models

# Simple Exponential Smoothing

# Double Expo. Smoothing

# Tripple Expo. Smoothing

# AR-I-MA model

#PACF- p

#diff - d

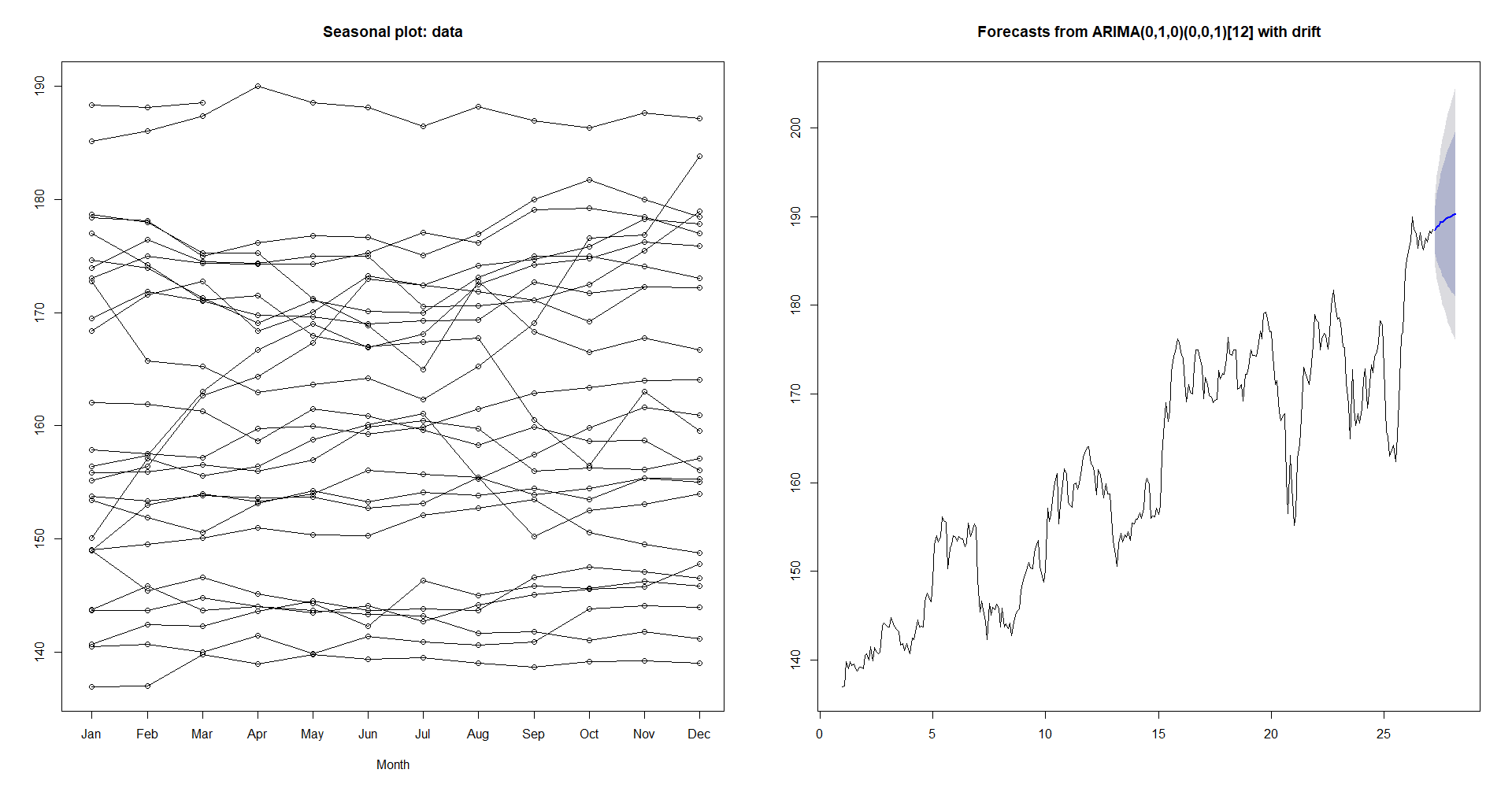
#ACF- q

model2<-auto.arima(data)

accuracy(model2)

|  |
| --- |
| ME RMSE MAE MPE MAPE MASE ACF1  Training set 0.001069574 2.110248 1.494022 -0.009293366 0.9211956 0.2496958 0.06278627 |
|  |
| |  | | --- | |  | |

plot(forecast(model2,h=12))



adf.test(diff(data))

plot(diff(data))

diff(data,differences = 3)

#running a model on diff data

model3<-auto.arima(diff(data))

accuracy(model3)

acf(diff(data))

pacf(diff(data))

#taking random order

model4 <- Arima(diff(data),order=c(4,0,5))

model4

accuracy(model4)

model5 <- Arima(diff(data),order=c(4,0,4))

model5

accuracy(model5)

Series: diff(data)

ARIMA(4,0,4) with non-zero mean

Coefficients:

ar1 ar2 ar3 ar4 ma1 ma2 ma3 ma4 mean

0.0457 -0.8416 0.0176 -0.8231 0.0304 0.8305 0.0458 0.9854 0.1628

s.e. 0.0411 0.0438 0.0437 0.0379 0.0235 0.0224 0.0230 0.0255 0.1253

sigma^2 estimated as 4.108: log likelihood=-666.18

AIC=1352.35 AICc=1353.08 BIC=1389.85

> accuracy(model5)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.0003290108 1.997672 1.481664 -Inf Inf 0.6270232 0.02443933

# MAPE = mean absolute percentage error (should be < 10%) for a good model

par(mfrow=c(1,2))

plot(forecast(model5,h=12))

plot(log(data))

