DATA624 Homework 2

Bin Lin

2018-3-25

The data below represent the monthly sales (in thousands) of product A for a plastics manufacturer for years 1 through 5 (data set plastics).

a. Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend?

There is both seasonal fluactuation and a trend. The highest sale of the year always falls around July or August. Since January the sale will consistently goes up. After it reaches the peak during July or August it will consistently fall. Year over year, we observe rise in sales.

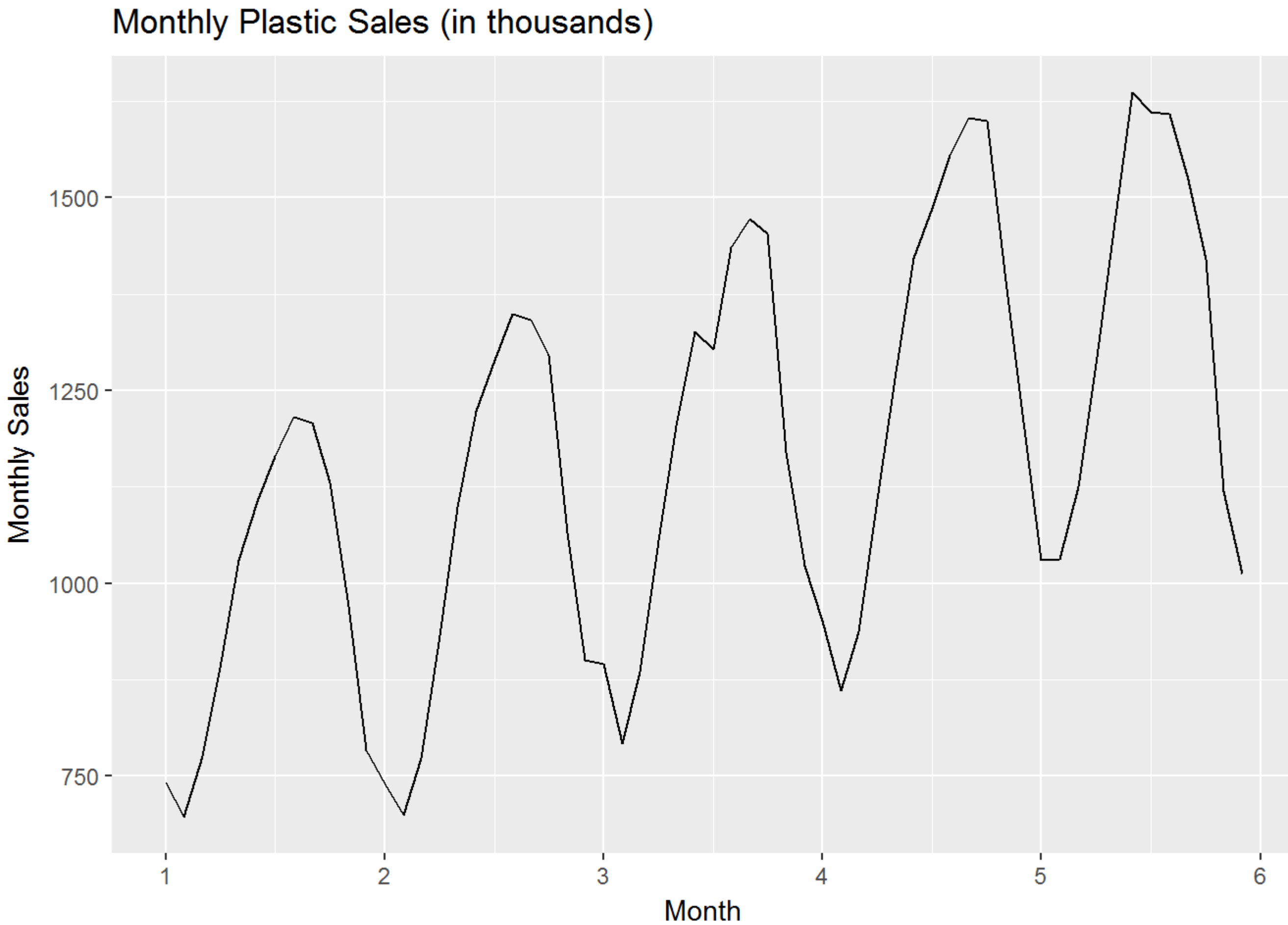
```
#install.packages("fma")
#data(package="fma")

library(fma)
```

```
## Loading required package: forecast
```

```
library(ggplot2)

head(plastics)
```

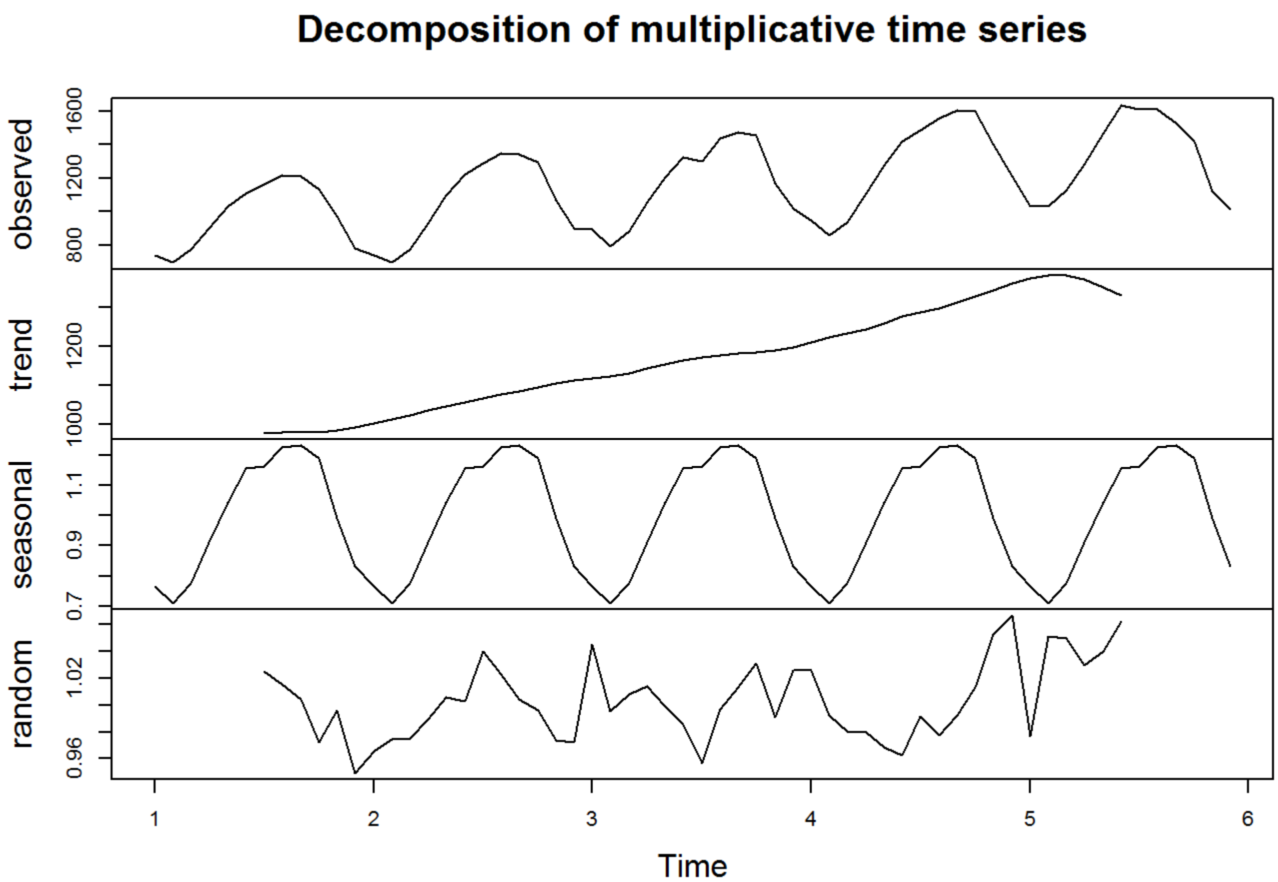
```
##    Jan  Feb  Mar  Apr  May  Jun
## 1  742  697  776  898 1030 1107
```

```
autoplot(plastics, main="Monthly Plastic Sales (in thousands)", ylab="Monthly Sales", xlab = "Mo
nth")
```



b. Use a classical multiplicative decomposition to calculate the trend-cycle and seasonal indices.

```
fit_decom <- decompose(plastics, type="multiplicative")
plot(fit_decom)
```



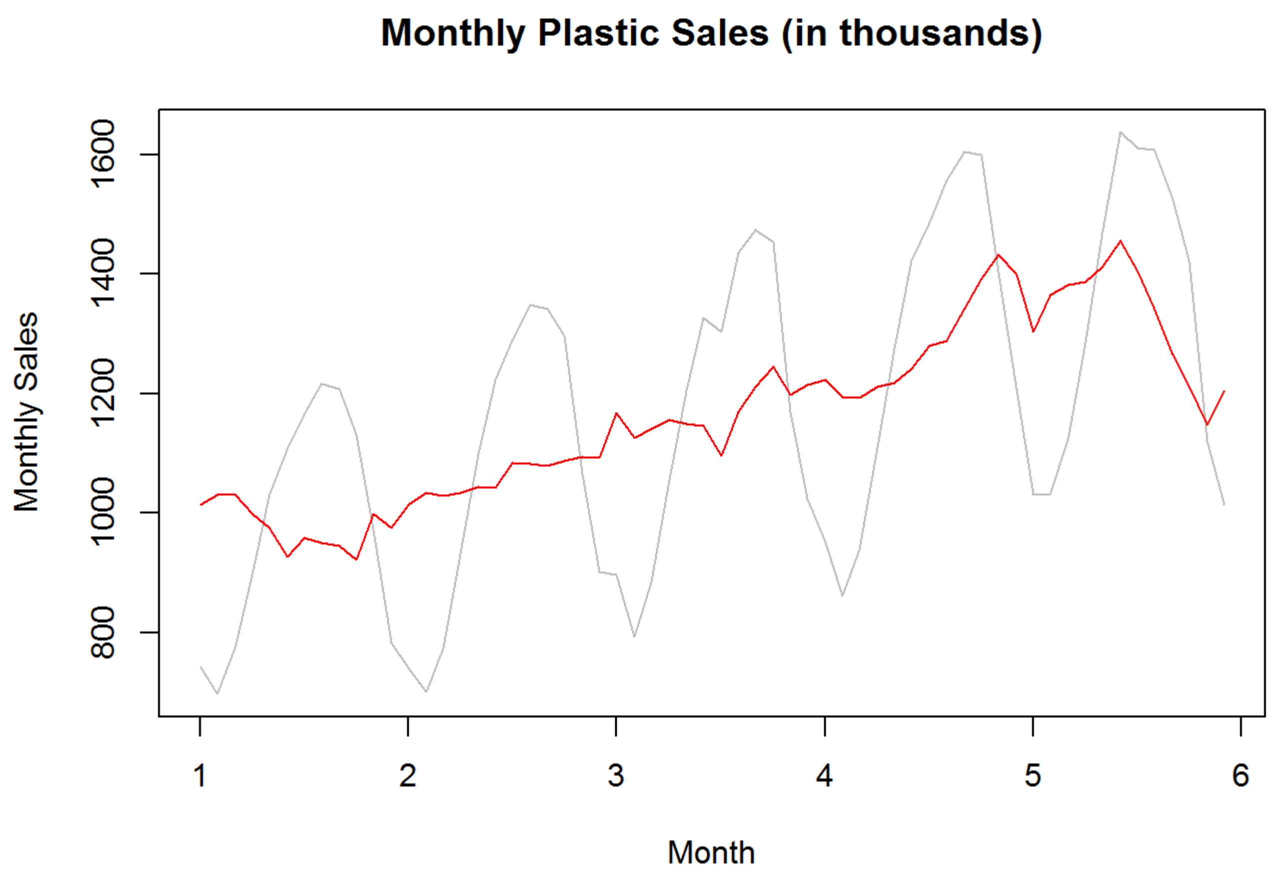
c. Do the results support the graphical interpretation from part (a)?

The trend-cycle and seasonal indices support the graphical interpretation from part (a). There is upward trend for the year over year sales. Furthermore,the season component shows periodic up and down movements on the sales in each individual year.

d. Compute and plot the seasonally adjusted data.

FOr multiplicative data, the seasonally adjusted values are obtained using yt/St. The red line in the following graph shows the seasonally adjusted data. After we removed the seasonal component, the monthly plastic sales actually goes down after year five.

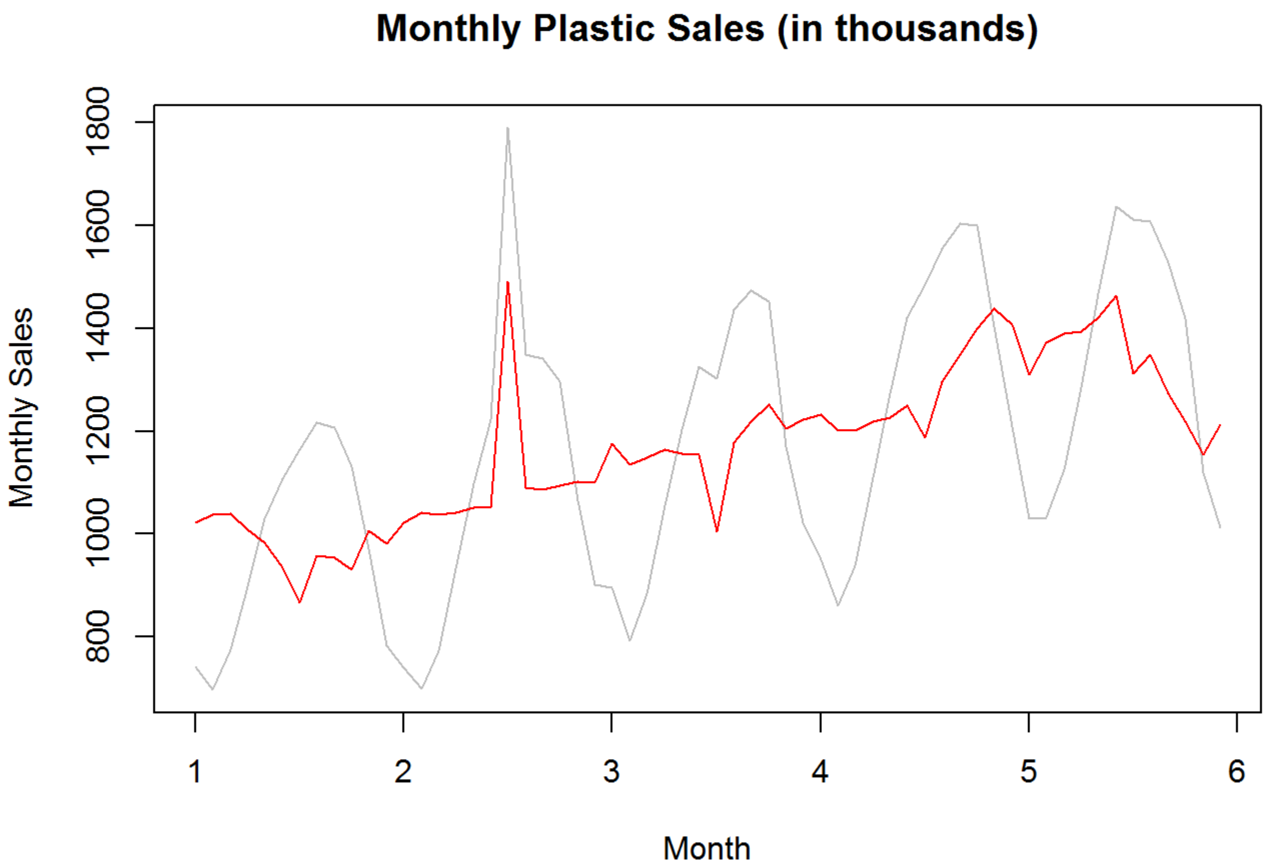
```
fit <- stl(plastics, s.window = "periodic")

plot(plastics, col="grey", main = "Monthly Plastic Sales (in thousands)", xlab="Month", ylab =
"Monthly Sales")
lines(seasadj(fit), col="red", ylab="Seasonally adjusted")
```



e. Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?

The outlier changed the Time Plot of the plastic sales dramatically. In the meantime, it also changed the seasonally adjusted data. The magnitude seems to be smaller than the change seen on the Time Plot graph.

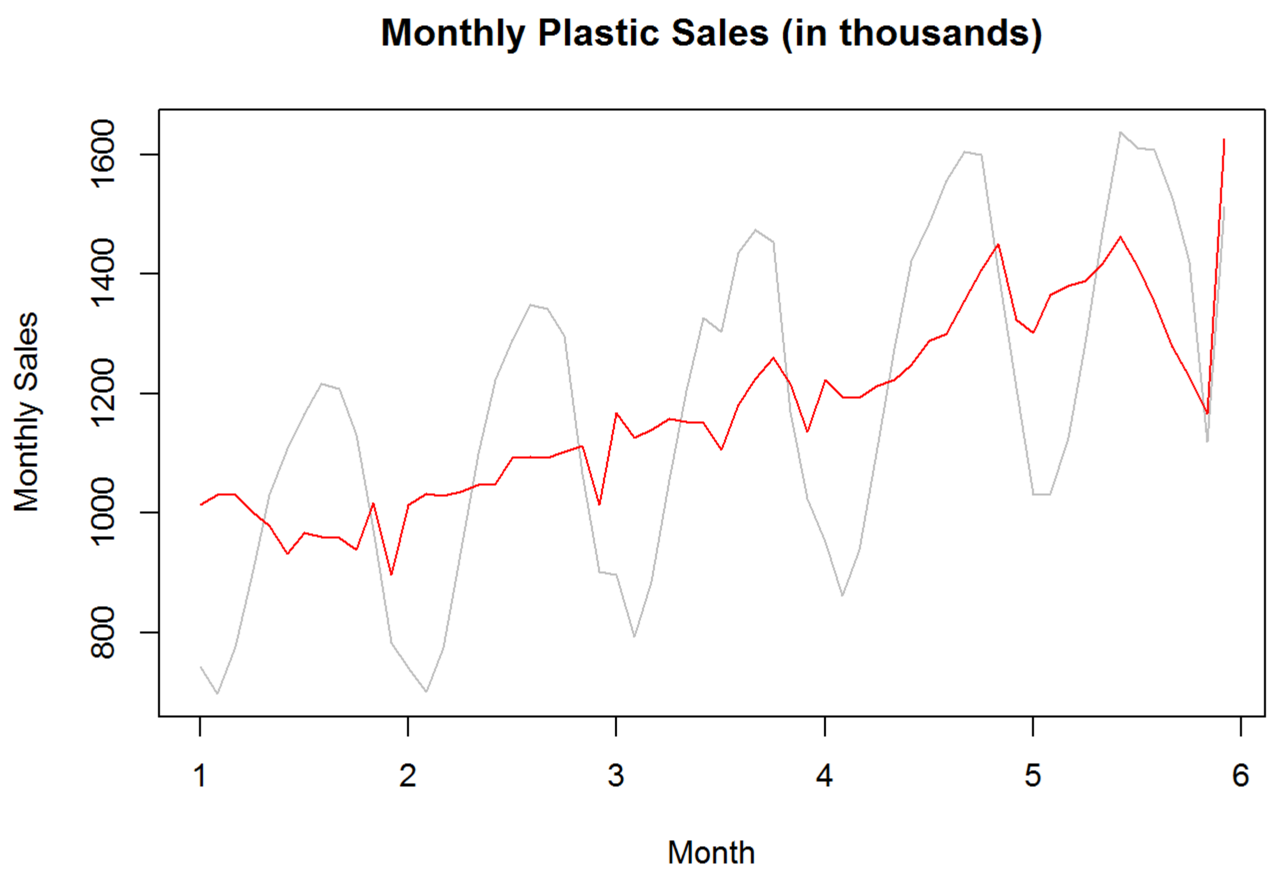
```
set.seed(100)

plastics2 <- plastics

x <- sample(1:length(plastics2), 1)

plastics2[x] <- plastics2[x] + 500
plot(plastics2, col="grey", main = "Monthly Plastic Sales (in thousands)", xlab="Month", ylab =
"Monthly Sales")

fit2 <- stl(plastics2, s.window = "periodic")
lines(seasadj(fit2), col="red", ylab="Seasonally adjusted")
```



f. Does it make any difference if the outlier is near the end rather than in the middle of the time series?

It seems like it does not make any difference if the outlier is near the end or in the middle of the time series. Both the Time Plot and seasonally adjusted data get changed about the same magnitude.

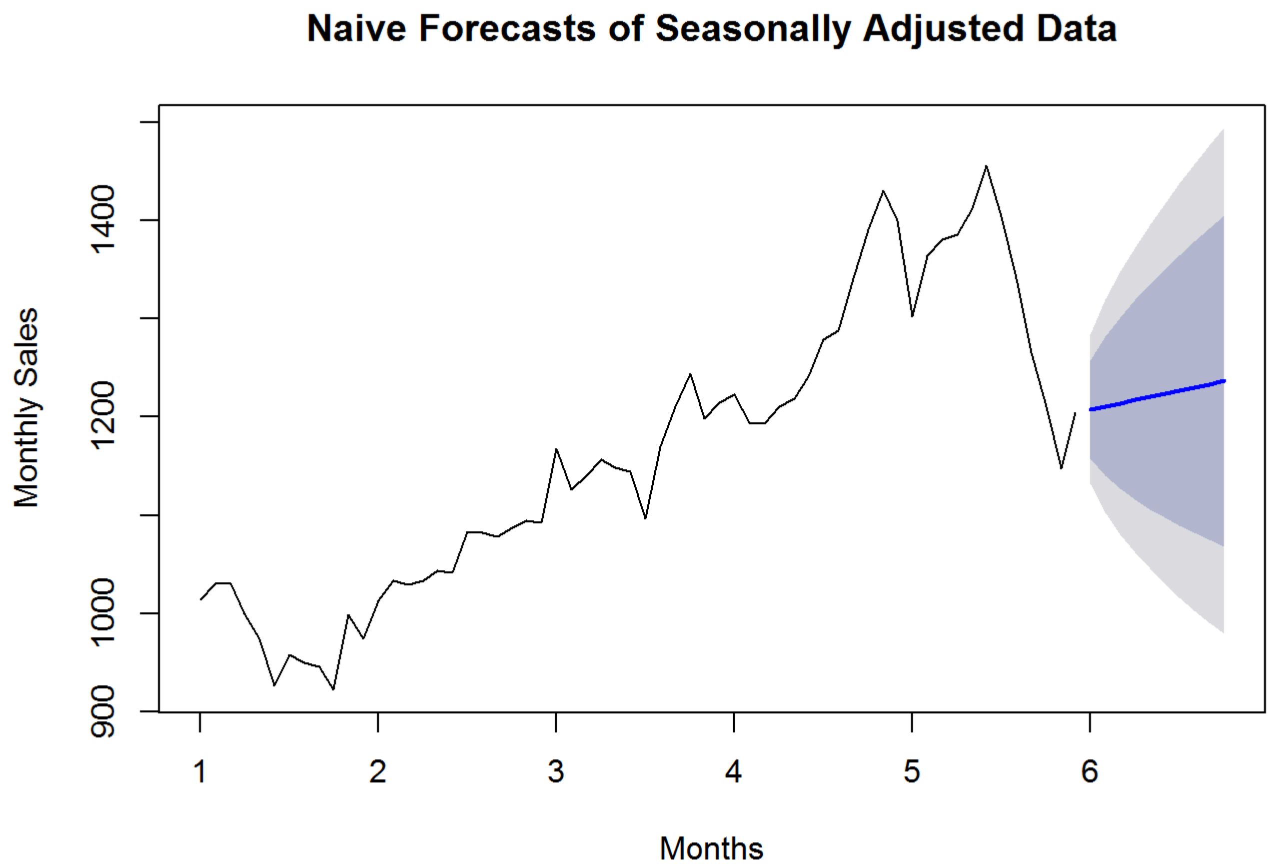
```
plastics3 <- plastics
plastics3[length(plastics3)] <- plastics3[length(plastics3)] + 500
plot(plastics3, col="grey", main = "Monthly Plastic Sales (in thousands)", xlab="Month", ylab =
"Monthly Sales")

fit3 <- stl(plastics3, s.window = "periodic")
lines(seasadj(fit3), col="red", ylab="Seasonally adjusted")
```



g. Use a random walk with drift to produce forecasts of the seasonally adjusted data.

Random walk with drift method is one of the non-seasonal forecasting emthod. It can be used to forecase seasonally adjusted data.

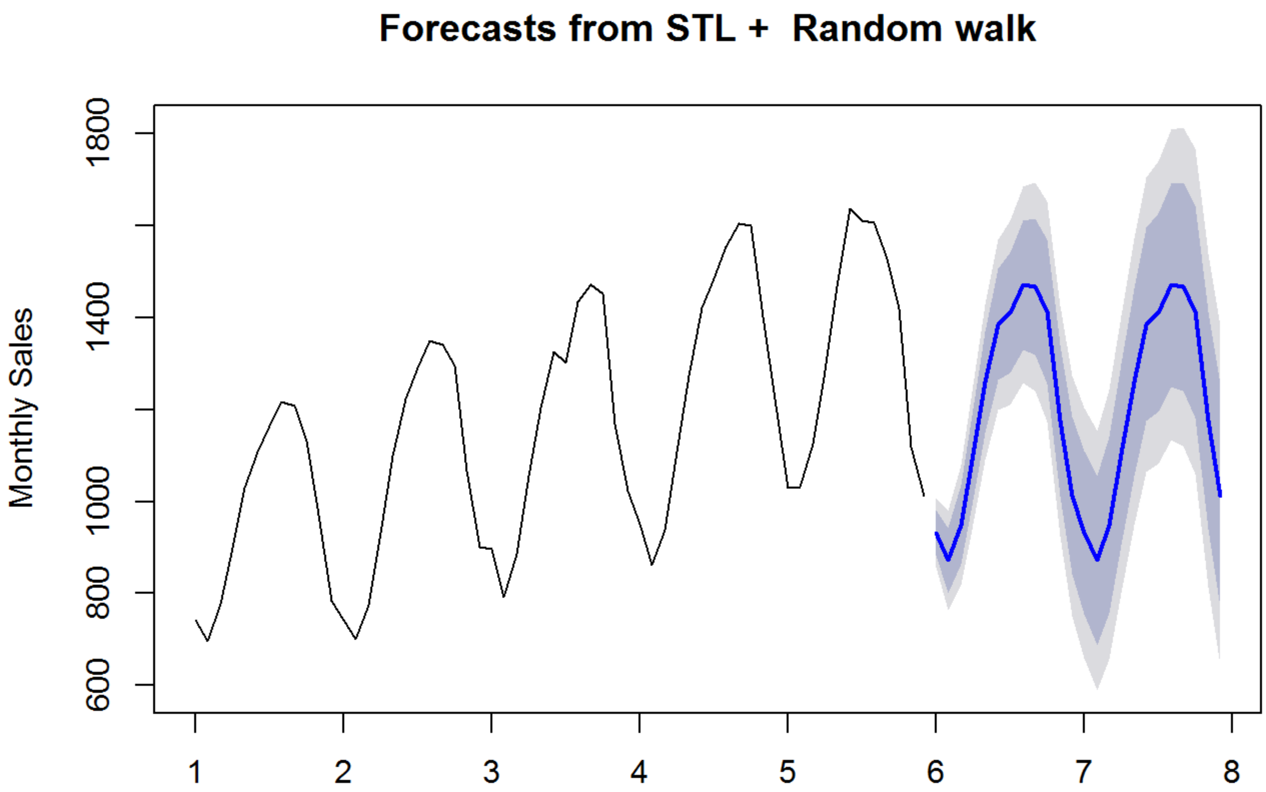
```
plastics_rwf <- rwf(seasadj(fit), drift = TRUE)

plot(plastics_rwf, main = "Naive Forecasts of Seasonally Adjusted Data", xlab="Months", ylab =
"Monthly Sales")
```



h. Reseasonalize the results to give forecasts on the original scale.

Once we add the forecast of the seasonal component to the seasonally adjusted data (the process is called reseasonalized), the resulting forecast will be more accurately capture the actual future values. The method forecast is enable us to do that. The following figure produced the result for the year 7 and 8.

```
fcast <- forecast(fit, method="naive")
plot(fcast, ylab="Monthly Sales")
```


Forecasts from STL + Random walk
Monthly Sales
600
800
1000
1400
1800
1
2
3
4
5
6
7
8