DATA624-HW3-Decomposition

FPP-Hyndman exercises 6.2, 6.3

Michael Y.

2/23/2020

Contents

```r
library(fpp2)
```

```
## Loading required package: ggplot2

## Loading required package: forecast

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff

## Loading required package: fma

## Loading required package: expsmooth
```

Homework 3 - Decomposition

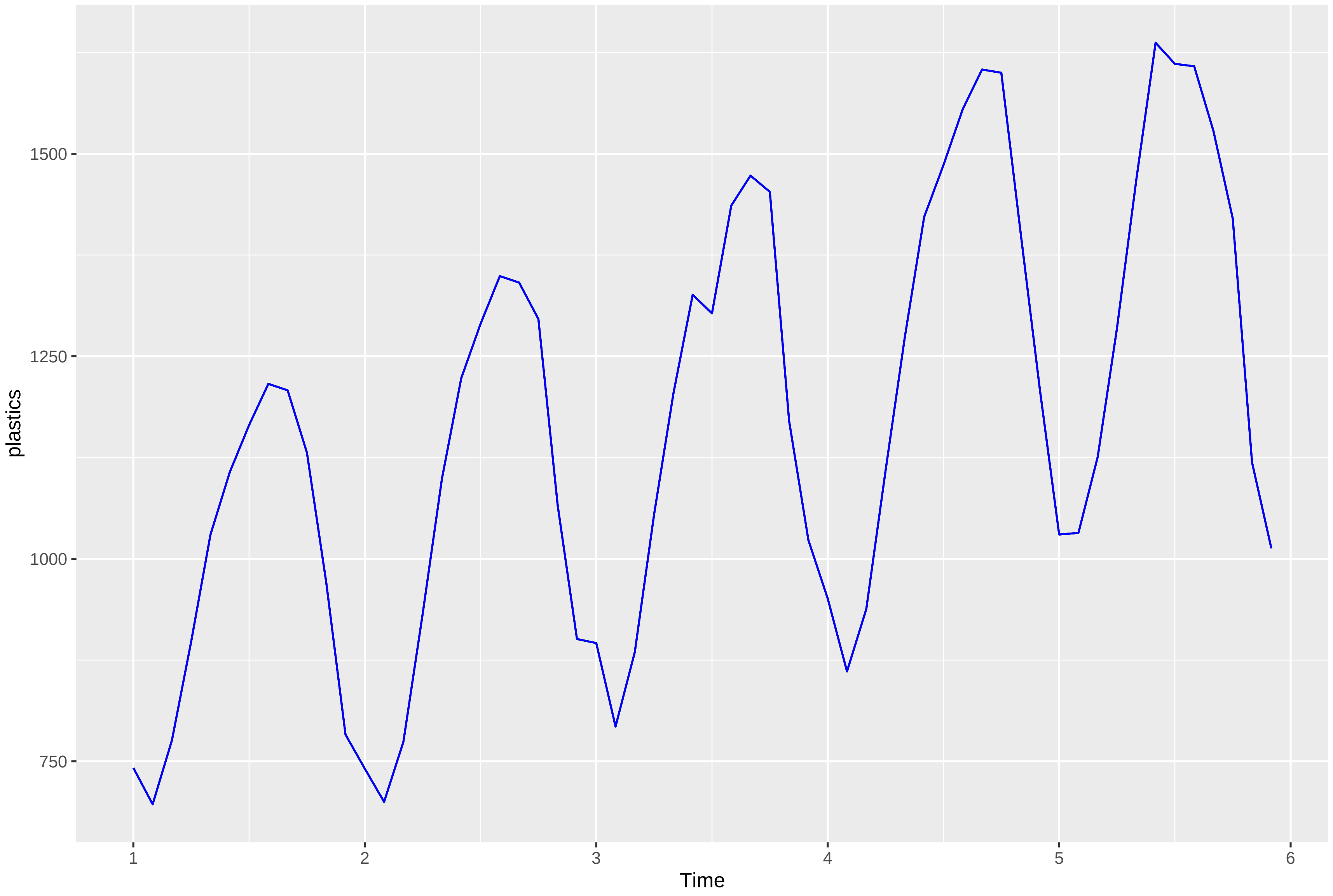
Do exercises 6.2, 6.3 from the online Hyndman book. Please submit both your Rpubs link as well as attach the .rmd file with your code.

6.2 The `plastics` data set consists of the monthly sales (in thousands) of product A for a plastics manufacturer for five years.

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

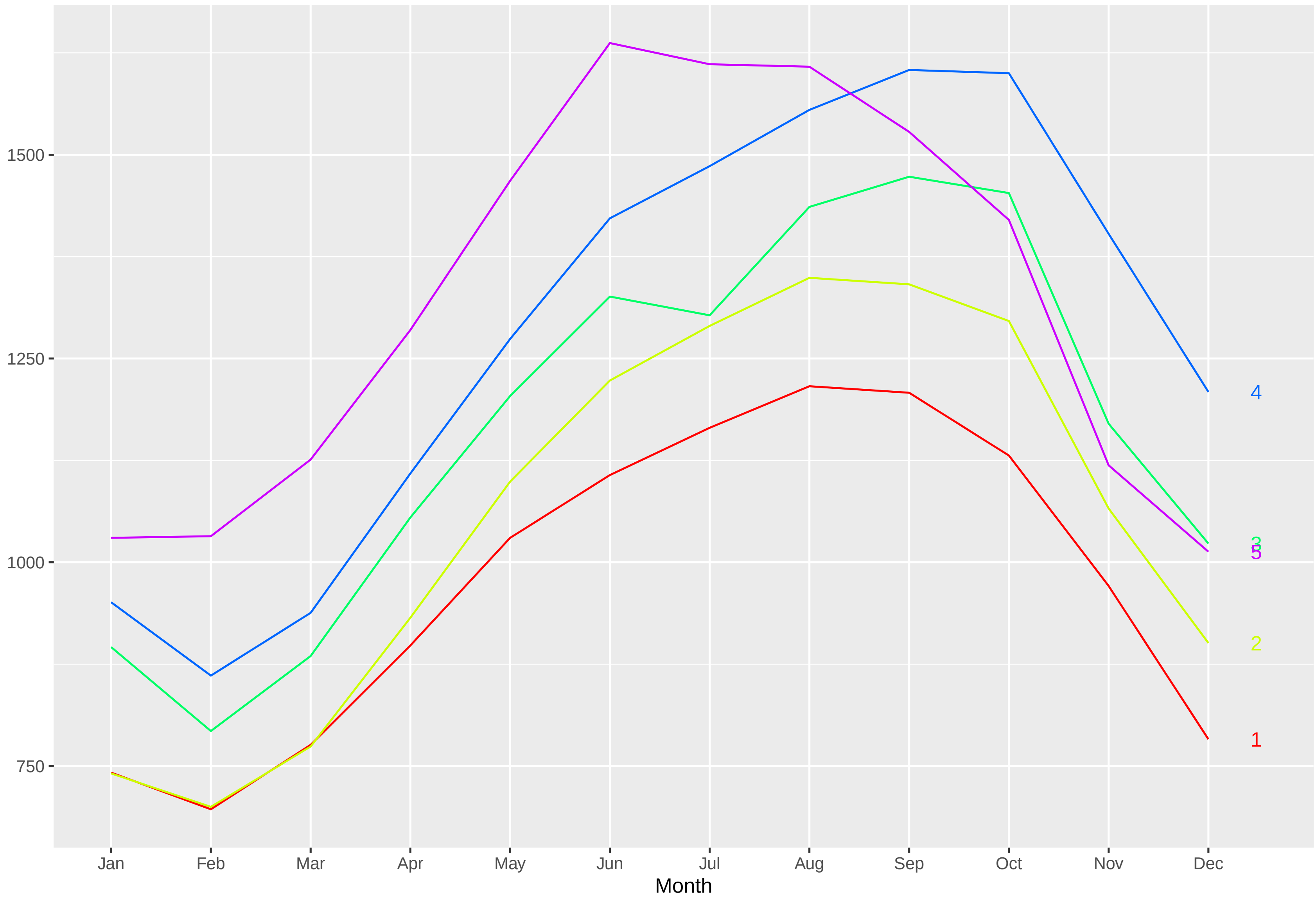
```r
# autoplot
autoplot(plastics)+geom_line(color="blue")+ggtitle("Plastics: Monthly sales of 'product A' for a plastics manufacturer.")
```

Plastics: Monthly sales of 'product A' for a plastics manufacturer.

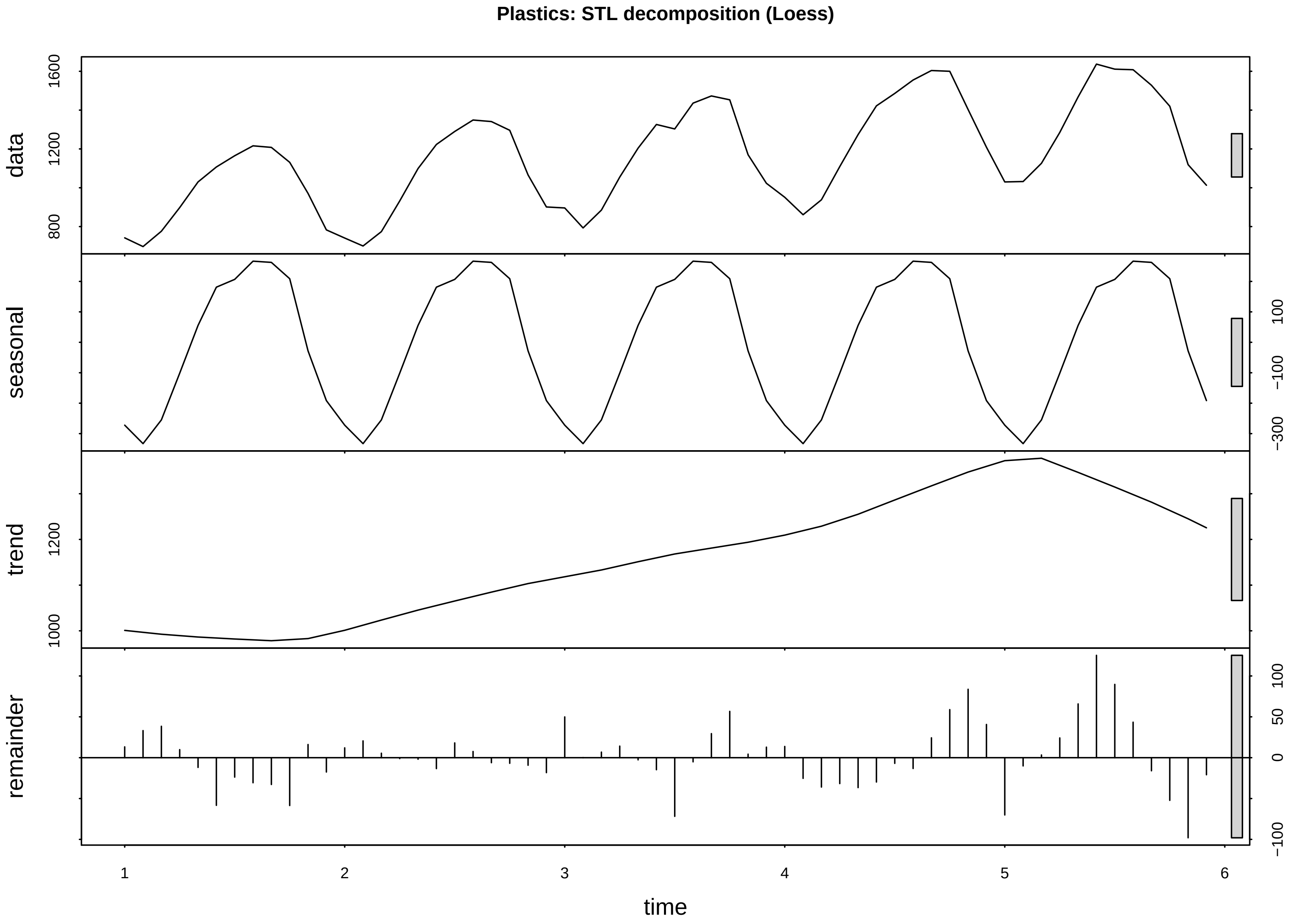


```r
# seasonplot
ggseasonplot(plastics, col=rainbow(5), year.labels=TRUE)
```

Seasonal plot: plastics



```r
# plot(stl)
plot(stl(plastics,"periodic"), main="Plastics: STL decomposition (Loess)")
```

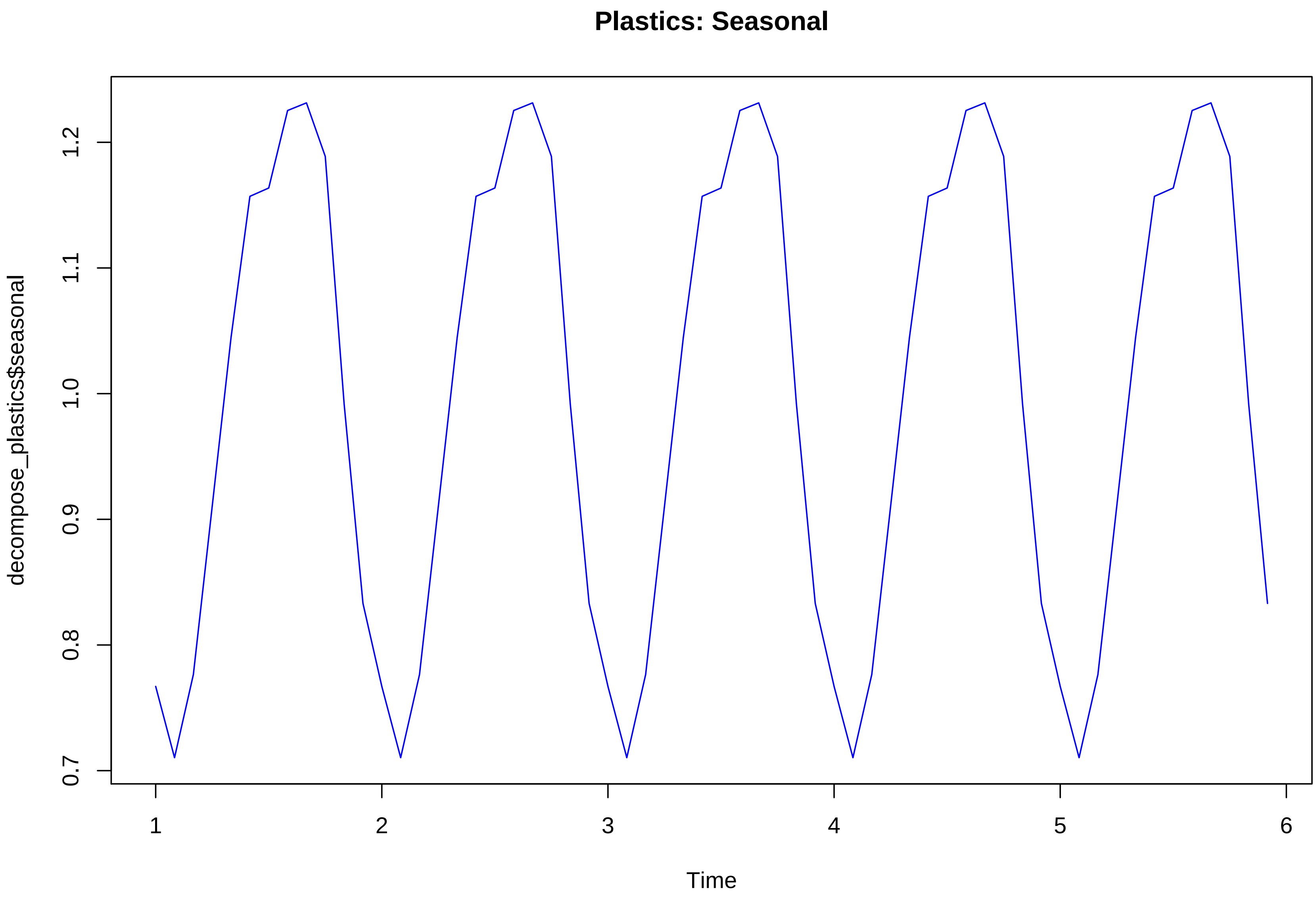

Plastics: STL decomposition (Loess)
data
800
1200
1600
seasonal
100
−100
−300
trend
1000
1200
remainder
100
50
0
−100
1
2
3
4
5
6
time

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

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

# Seasonal
decompose_plastics$seasonal
```

```
##         Jan       Feb       Mar       Apr       May       Jun       Jul
## 1 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317
## 2 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317
## 3 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317
## 4 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317
## 5 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317
##         Aug       Sep       Oct       Nov       Dec
## 1 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834
## 2 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834
## 3 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834
## 4 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834
## 5 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834
```

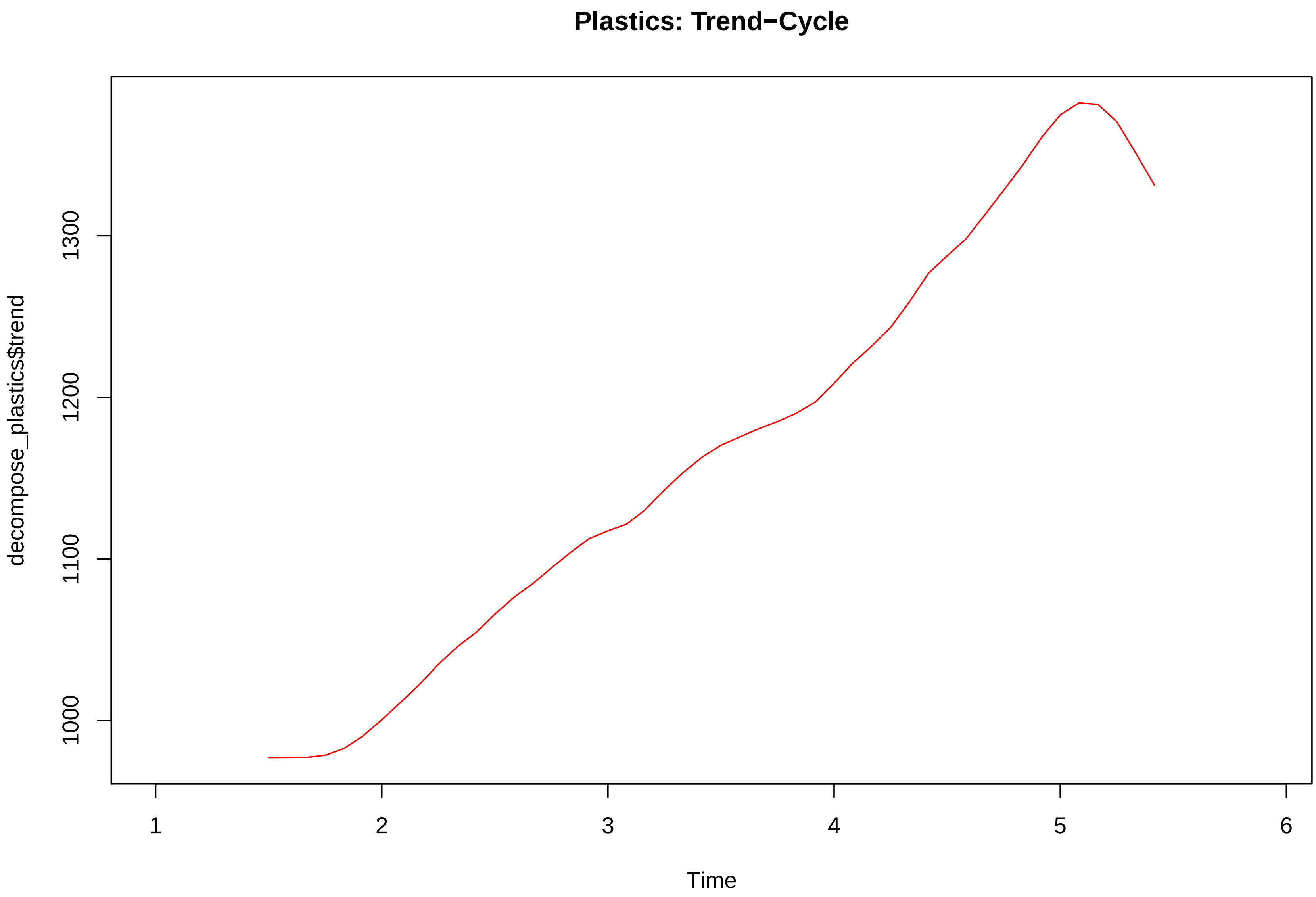
```
plot(decompose_plastics$seasonal, main="Plastics: Seasonal", col="blue")
```


Plastics: Seasonal
decompose_plastics$seasonal
0.7
0.8
0.9
1.0
1.1
1.2
1
2
3
4
5
6
Time

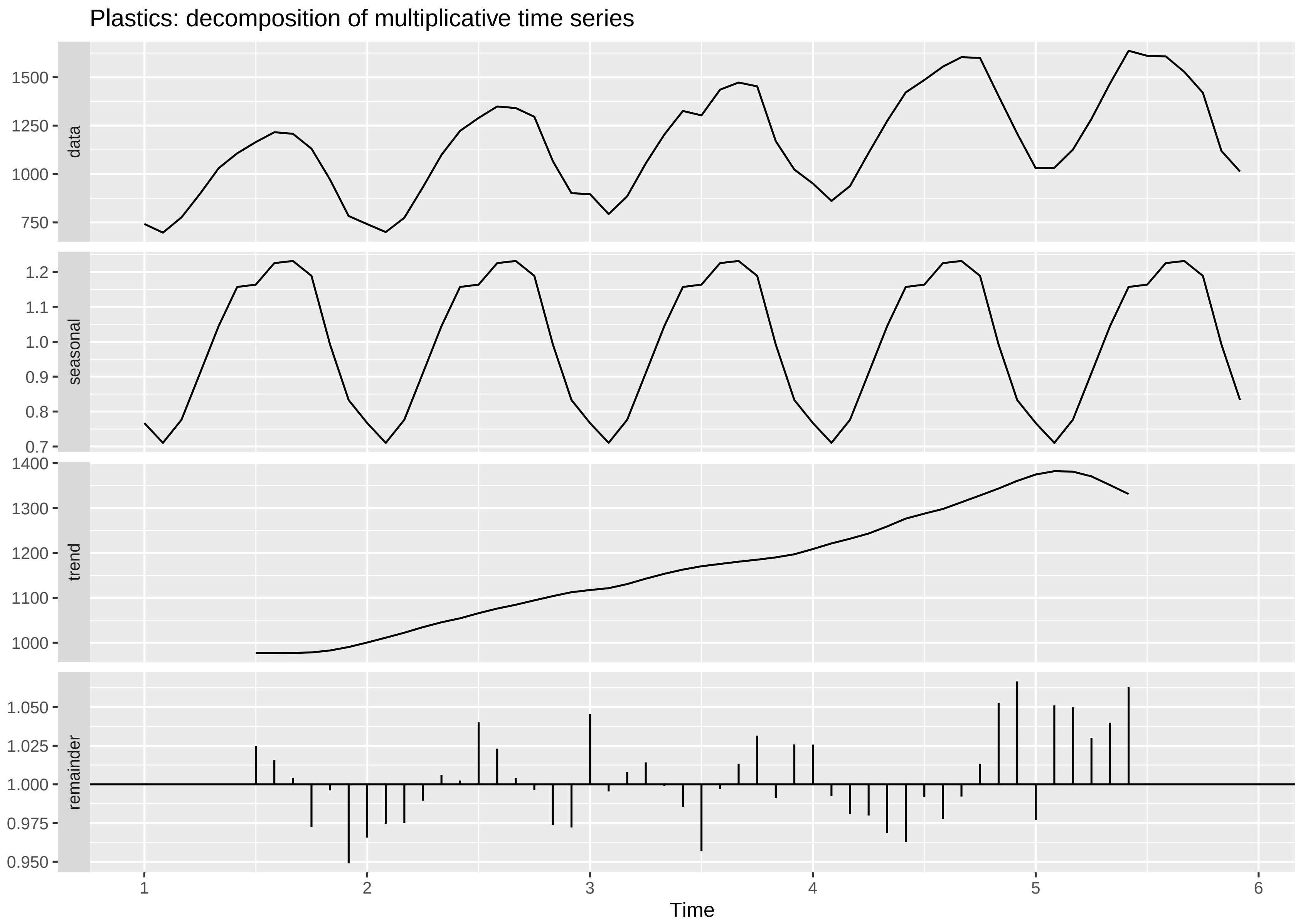
```r
# Trend
decompose_plastics$trend
```

```
##          Jan       Feb       Mar       Apr       May       Jun       Jul
## 1        NA        NA        NA        NA        NA        NA  976.9583
## 2 1000.4583 1011.2083 1022.2917 1034.7083 1045.5417 1054.4167 1065.7917
## 3 1117.3750 1121.5417 1130.6667 1142.7083 1153.5833 1163.0000 1170.3750
## 4 1208.7083 1221.2917 1231.7083 1243.2917 1259.1250 1276.5833 1287.6250
## 5 1374.7917 1382.2083 1381.2500 1370.5833 1351.2500 1331.2500        NA
##          Aug       Sep       Oct       Nov       Dec
## 1  977.0417  977.0833  978.4167  982.7083  990.4167
## 2 1076.1250 1084.6250 1094.3750 1103.8750 1112.5417
## 3 1175.5000 1180.5417 1185.0000 1190.1667 1197.0833
## 4 1298.0417 1313.0000 1328.1667 1343.5833 1360.6250
## 5        NA        NA        NA        NA        NA
```

```r
plot(decompose_plastics$trend, main="Plastics: Trend-Cycle",col="red")
```


Plastics: Trend−Cycle
decompose_plastics$trend
1000
1100
1200
1300
1
2
3
4
5
6
Time

```
# autoplot
autoplot(decompose_plastics) + ggtitle("Plastics: decomposition of multiplicative time series")
```


Plastics: decomposition of multiplicative time series
data
1500
1250
1000
750
seasonal
1.2
1.1
1.0
0.9
0.8
0.7
trend
1400
1300
1200
1100
1000
remainder
1.050
1.025
1.000
0.975
0.950
1
2
3
4
5
6
Time

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

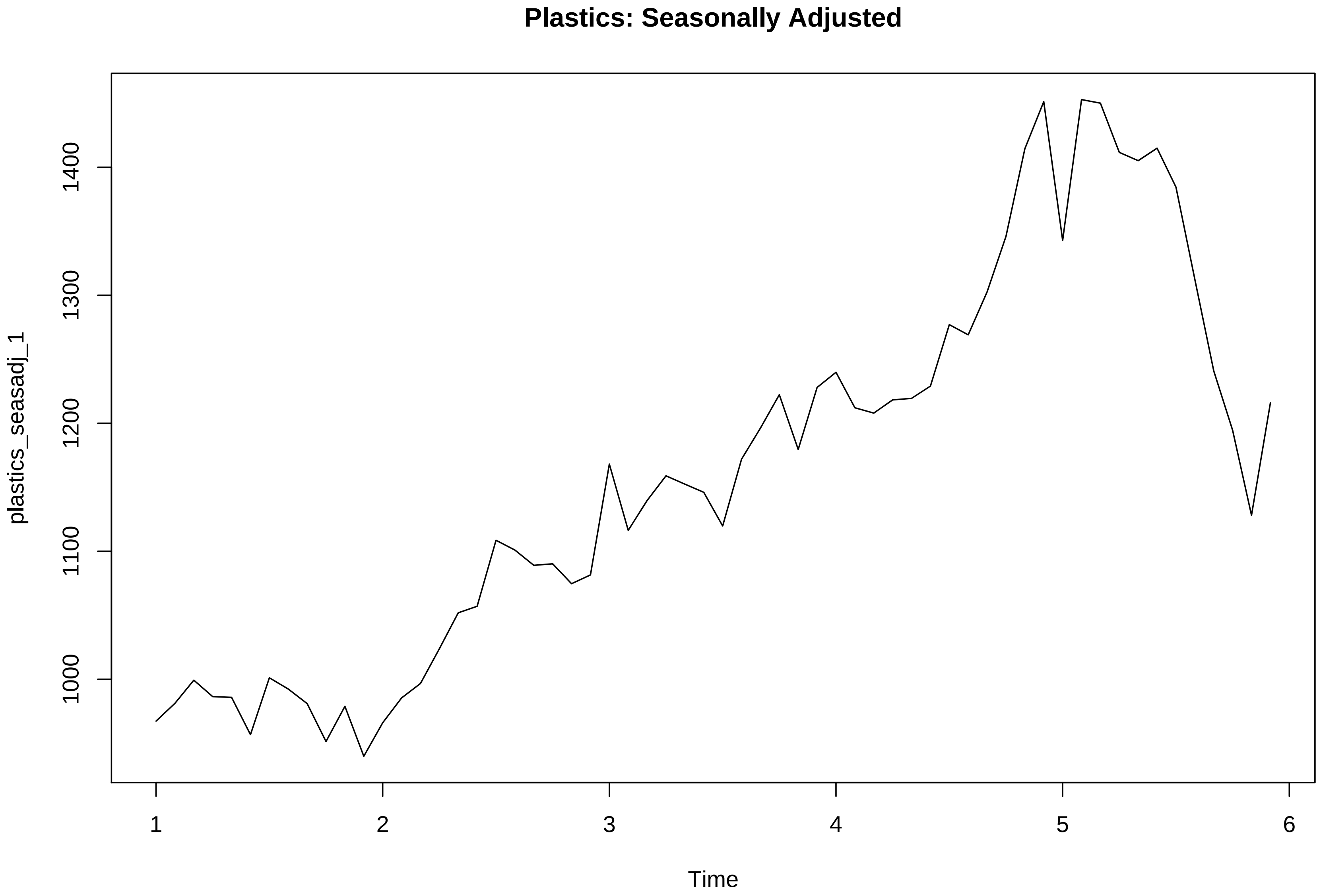
Yes, the results do support the graphical interpretation.

d) Compute and plot the seasonally adjusted data.

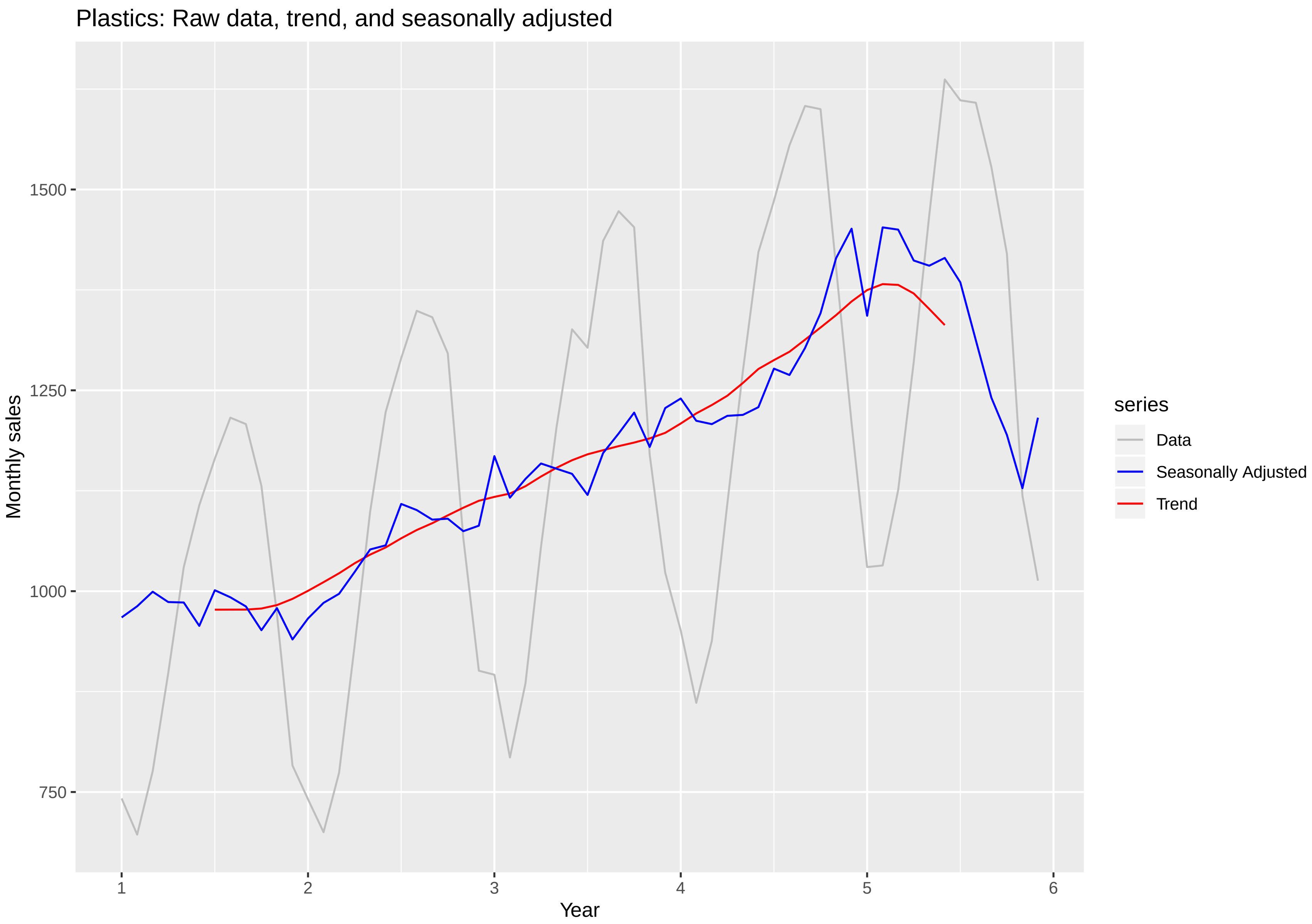
```
plastics_seasadj_1 <- seasadj(decompose_plastics)
plastics_seasadj_1
```

```
##         Jan       Feb       Mar       Apr       May       Jun       Jul
## 1  967.3468  981.2262  999.3182  986.4758  985.8925  956.7826 1001.1759
## 2  966.0431  985.4495  996.7427 1023.8257 1051.9377 1057.0417 1108.5982
## 3 1168.1168 1116.3736 1139.6864 1158.9443 1152.4413 1146.0648 1119.7701
## 4 1239.8204 1212.1029 1207.9388 1218.2647 1219.4437 1229.0378 1277.0364
## 5 1342.8129 1452.8342 1450.0416 1411.6051 1405.1361 1414.8628 1384.4587
##         Aug       Sep       Oct       Nov       Dec
## 1  992.4139  981.0263  951.4241  978.9119  939.8819
## 2 1100.9592 1089.0366 1090.2260 1074.6860 1081.5244
## 3 1171.9625 1196.2349 1222.2981 1179.5334 1227.9683
## 4 1269.0820 1302.6210 1345.9580 1414.4319 1451.2352
## 5 1312.3369 1240.9008 1194.5377 1128.1178 1215.9647
```

```
plot(plastics_seasadj_1, main="Plastics: Seasonally Adjusted")
```


Plastics: Seasonally Adjusted
plastics_seasadj_1
1000
1100
1200
1300
1400
1
2
3
4
5
6
Time

```
autoplot(plastics, series="Data") +
  autolayer(trendcycle(decompose_plastics), series="Trend") +
  autolayer(seasadj(decompose_plastics), series="Seasonally Adjusted") +
  xlab("Year") + ylab("Monthly sales") +
  ggtitle("Plastics: Raw data, trend, and seasonally adjusted") +
  scale_colour_manual(values=c("gray","blue","red"),
             breaks=c("Data","Seasonally Adjusted","Trend"))
```

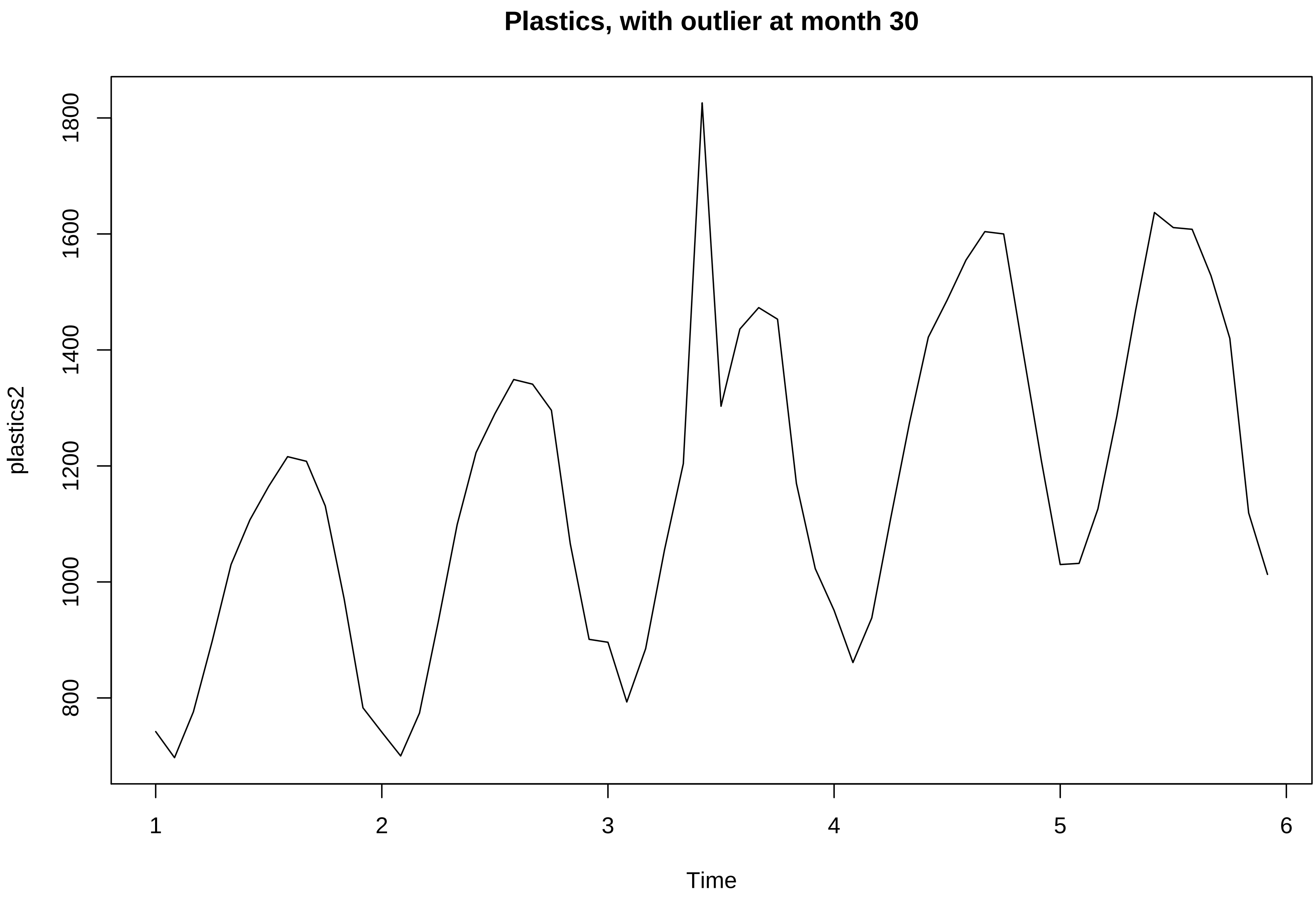

Plastics: Raw data, trend, and seasonally adjusted
Monthly sales
1500
1250
1000
750
1
2
3
4
5
6
Year
series
Data
Seasonally Adjusted
Trend

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

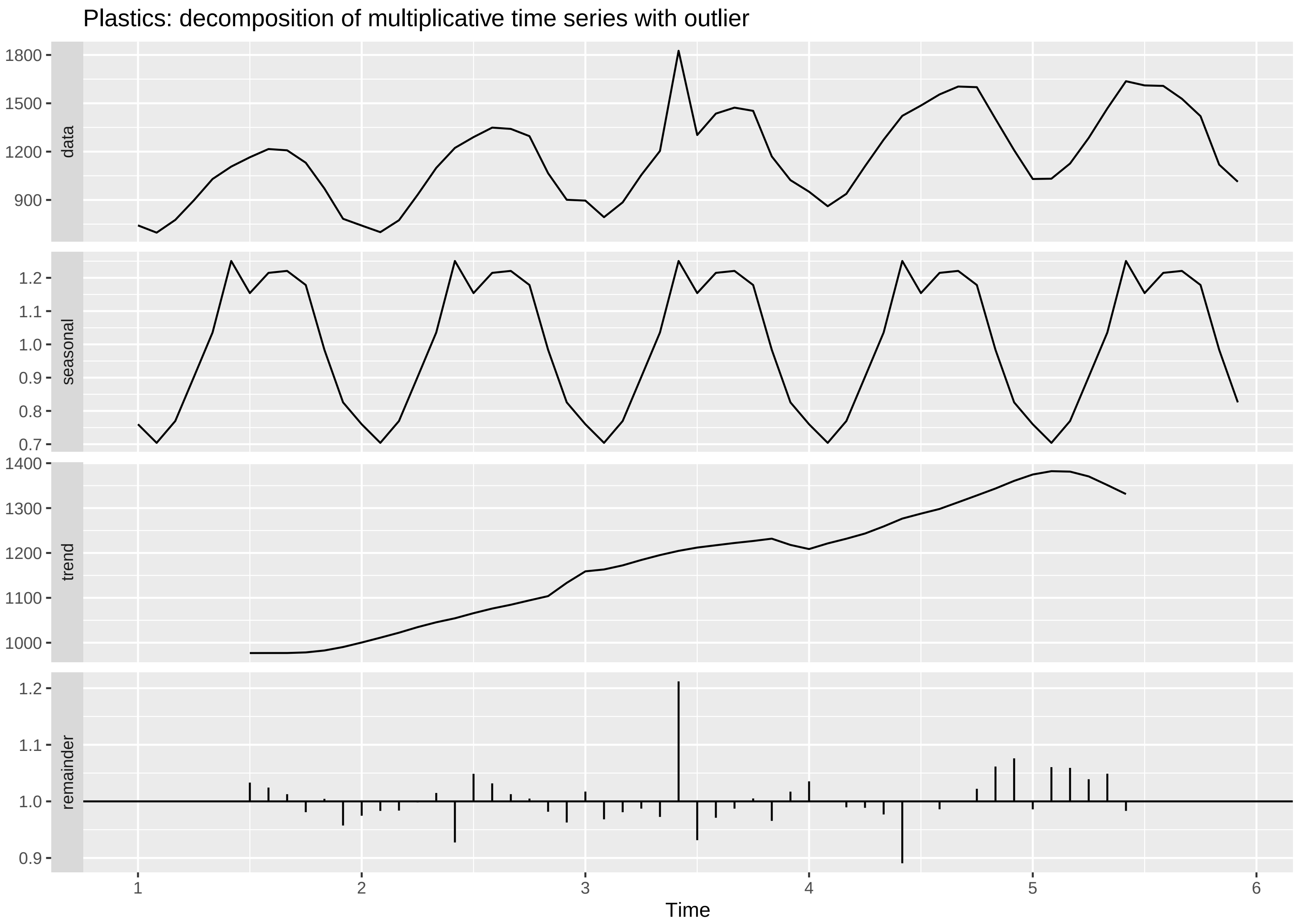
```
plastics2 <- plastics
## changing June of year 3
plastics2[30] <- plastics2[30]+500
plastics2
```

```
##    Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1  742  697  776  898 1030 1107 1165 1216 1208 1131  971  783
## 2  741  700  774  932 1099 1223 1290 1349 1341 1296 1066  901
## 3  896  793  885 1055 1204 1826 1303 1436 1473 1453 1170 1023
## 4  951  861  938 1109 1274 1422 1486 1555 1604 1600 1403 1209
## 5 1030 1032 1126 1285 1468 1637 1611 1608 1528 1420 1119 1013
```

```
plot(plastics2, main="Plastics, with outlier at month 30")
```


Plastics, with outlier at month 30
plastics2
800
1000
1200
1400
1600
1800
1
2
3
4
5
6
Time

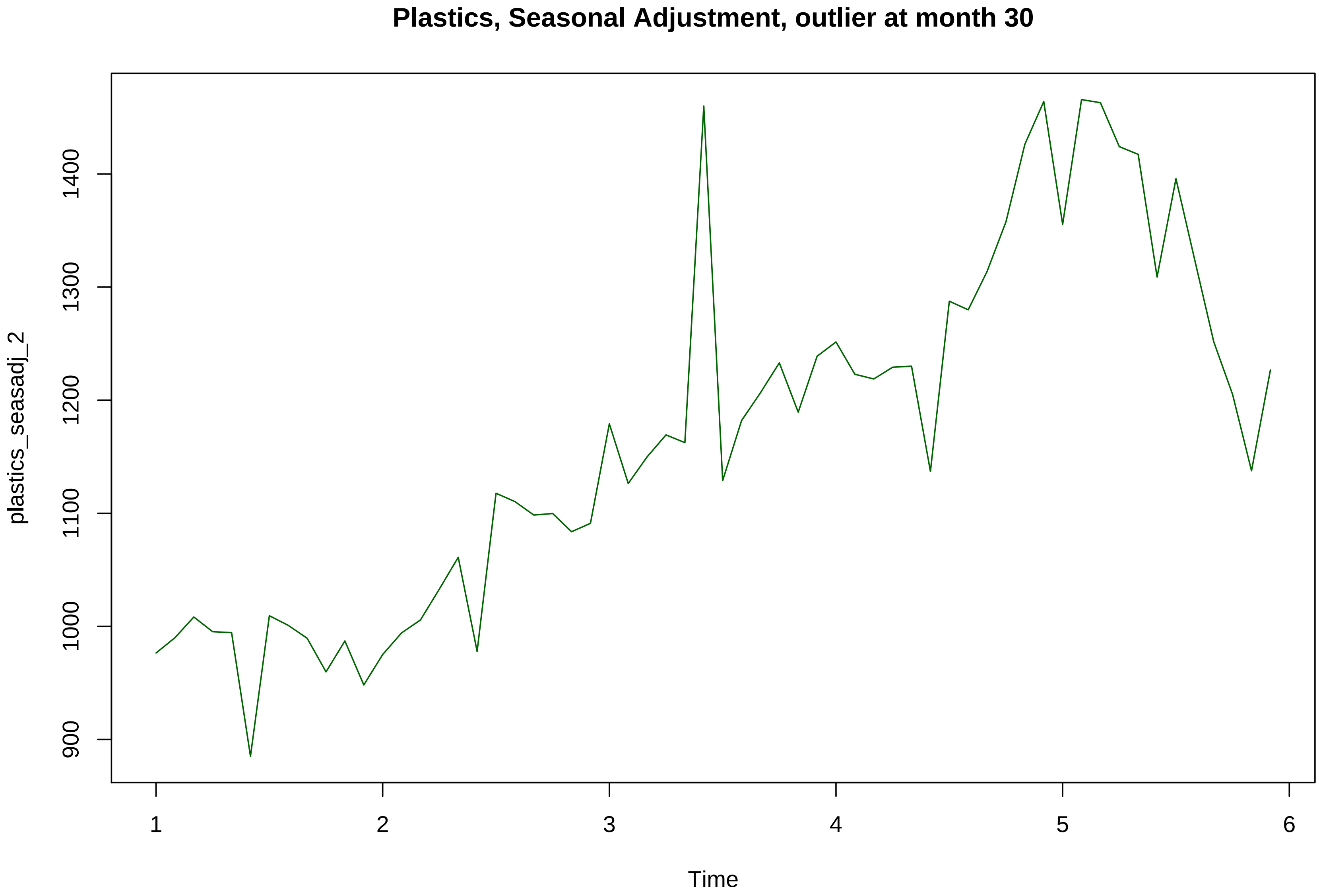
```
## decompose the perturbed series plastics2
decompose_plastics_2 <- decompose(plastics2, type = "multiplicative")
# autoplot
autoplot(decompose_plastics_2) +
  ggtitle("Plastics: decomposition of multiplicative time series with outlier")
```


Plastics: decomposition of multiplicative time series with outlier
data
1800
1500
1200
900
seasonal
1.2
1.1
1.0
0.9
0.8
0.7
trend
1400
1300
1200
1100
1000
remainder
1.2
1.1
1.0
0.9
1
2
3
4
5
6
Time

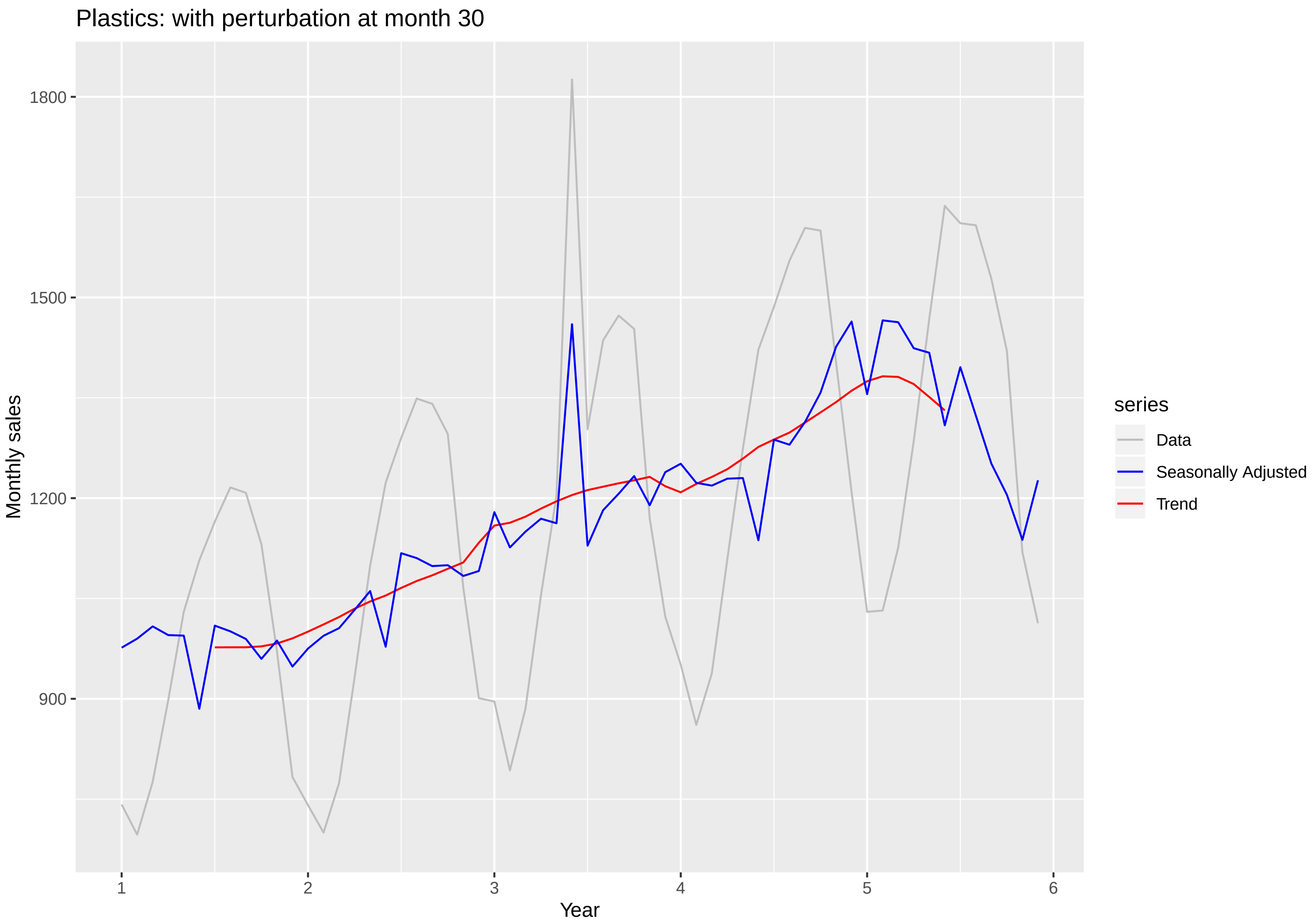
```
plastics_seasadj_2 <- seasadj(decompose_plastics_2)
plastics_seasadj_2
```

```
##          Jan       Feb       Mar       Apr       May       Jun       Jul
## 1  976.4448  989.9751 1008.2715  995.2777  994.4756  885.1399 1009.4012
## 2  975.1288  994.2361 1005.6728 1032.9608 1061.0958  977.8916 1117.7060
## 3 1179.1031 1126.3275 1149.8972 1169.2851 1162.4744 1460.0410 1128.9697
## 4 1251.4811 1222.9104 1218.7611 1229.1347 1230.0601 1137.0089 1287.5280
## 5 1355.4422 1465.7881 1463.0331 1424.2003 1417.3691 1308.9195 1395.8328
##          Aug       Sep       Oct       Nov       Dec
## 1 1000.8786  989.4971  959.7579  987.1109  948.2051
## 2 1110.3498 1098.4400 1099.7756 1083.6872 1091.1019
## 3 1181.9587 1206.5639 1233.0046 1189.4128 1238.8427
## 4 1279.9065 1313.8686 1357.7477 1426.2787 1464.0868
## 5 1323.5303 1251.6155 1205.0011 1137.5666 1226.7328
```

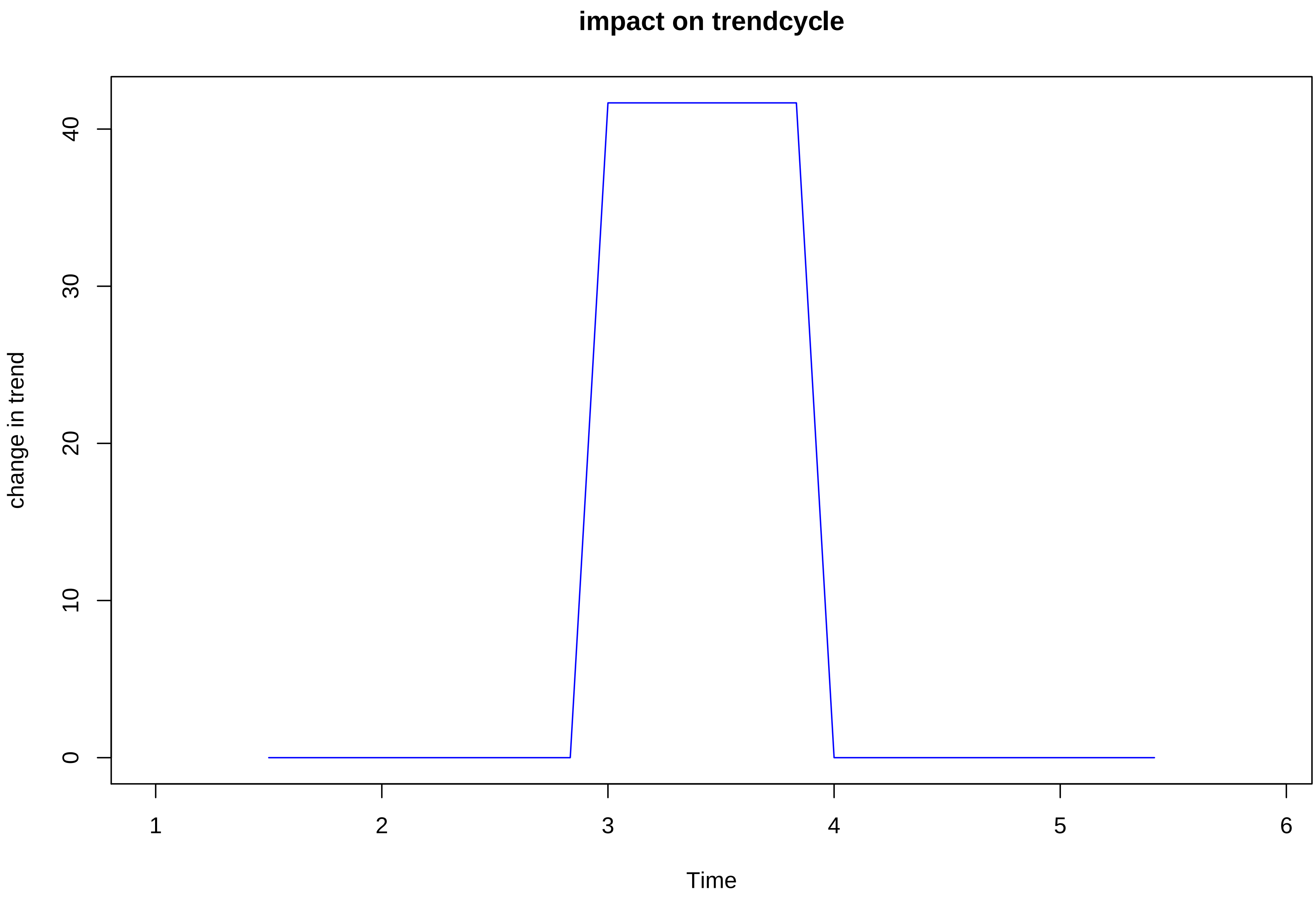
```
plot(plastics_seasadj_2,
     main="Plastics, Seasonal Adjustment, outlier at month 30",
     col="darkgreen")
```


Plastics, Seasonal Adjustment, outlier at month 30
plastics_seasadj_2
900
1000
1100
1200
1300
1400
1
2
3
4
5
6
Time

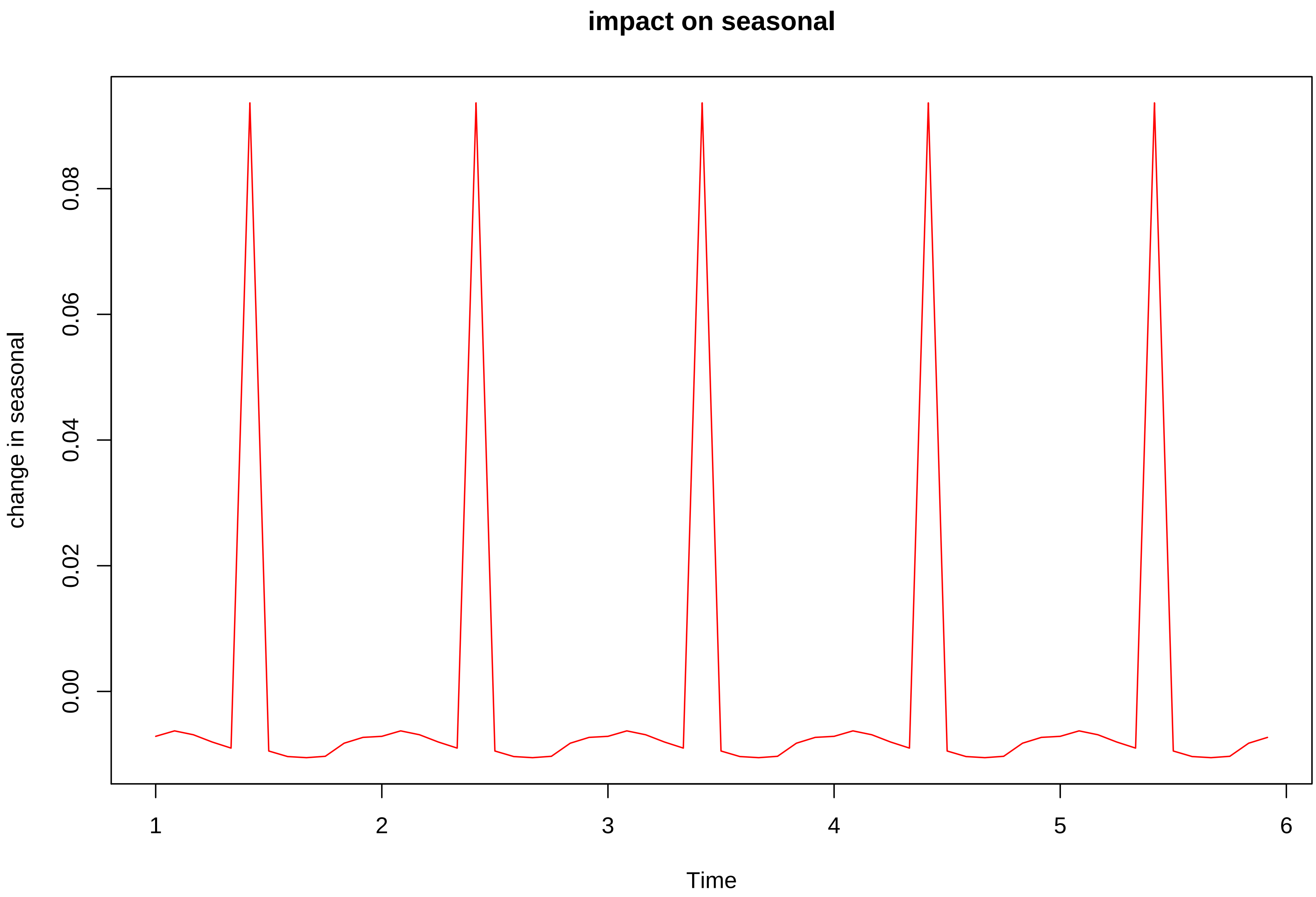
```r
autoplot(plastics2, series="Data") +
  autolayer(trendcycle(decompose_plastics_2), series="Trend") +
  autolayer(seasadj(decompose_plastics_2), series="Seasonally Adjusted") +
  xlab("Year") + ylab("Monthly sales") +
  ggtitle("Plastics: with perturbation at month 30") +
  scale_colour_manual(values=c("gray","blue","red"),
             breaks=c("Data","Seasonally Adjusted","Trend"))
```


Plastics: with perturbation at month 30
Monthly sales
1800
1500
1200
900
Year
1
2
3
4
5
6
series
Data
Seasonally Adjusted
Trend

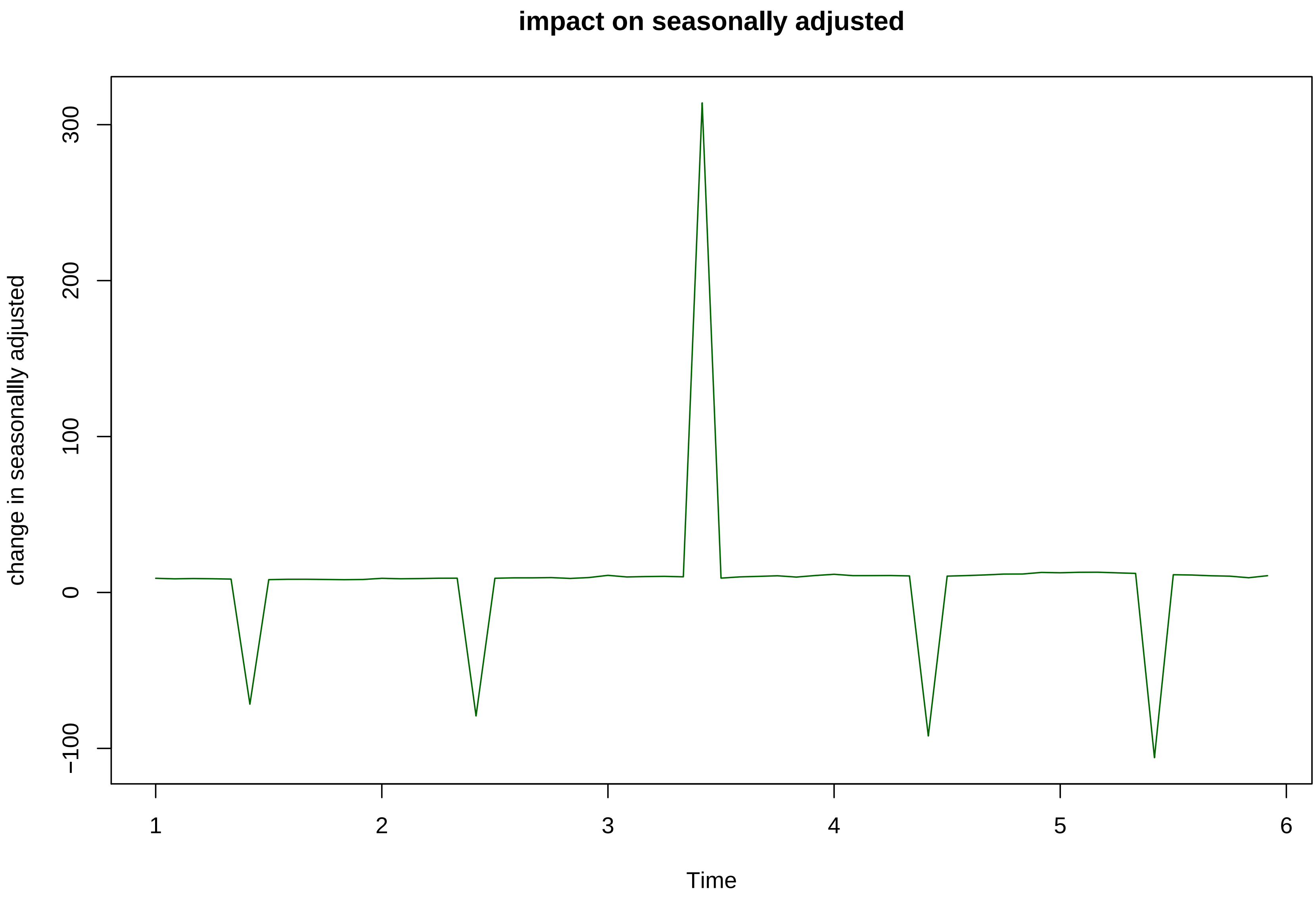
```r
## impact on trendcycle
plot(trendcycle(decompose_plastics_2)-trendcycle(decompose_plastics),
     ylab="change in trend", main="impact on trendcycle", col="blue")
```


impact on trendcycle
change in trend
0
10
20
30
40
1
2
3
4
5
6
Time

```
## impact on seasonal
plot(seasonal(decompose_plastics_2)-seasonal(decompose_plastics),
     ylab="change in seasonal", main="impact on seasonal", col="red")
```


impact on seasonal
change in seasonal
0.00
0.02
0.04
0.06
0.08
1
2
3
4
5
6
Time

```r
## impact on seasonally adjusted
plot(seasadj(decompose_plastics_2)-seasadj(decompose_plastics),
     ylab="change in seasonallly adjusted",
     main="impact on seasonally adjusted", col="darkgreen")
```


impact on seasonally adjusted
change in seasonallly adjusted
300
200
100
0
−100
1
2
3
4
5
6
Time

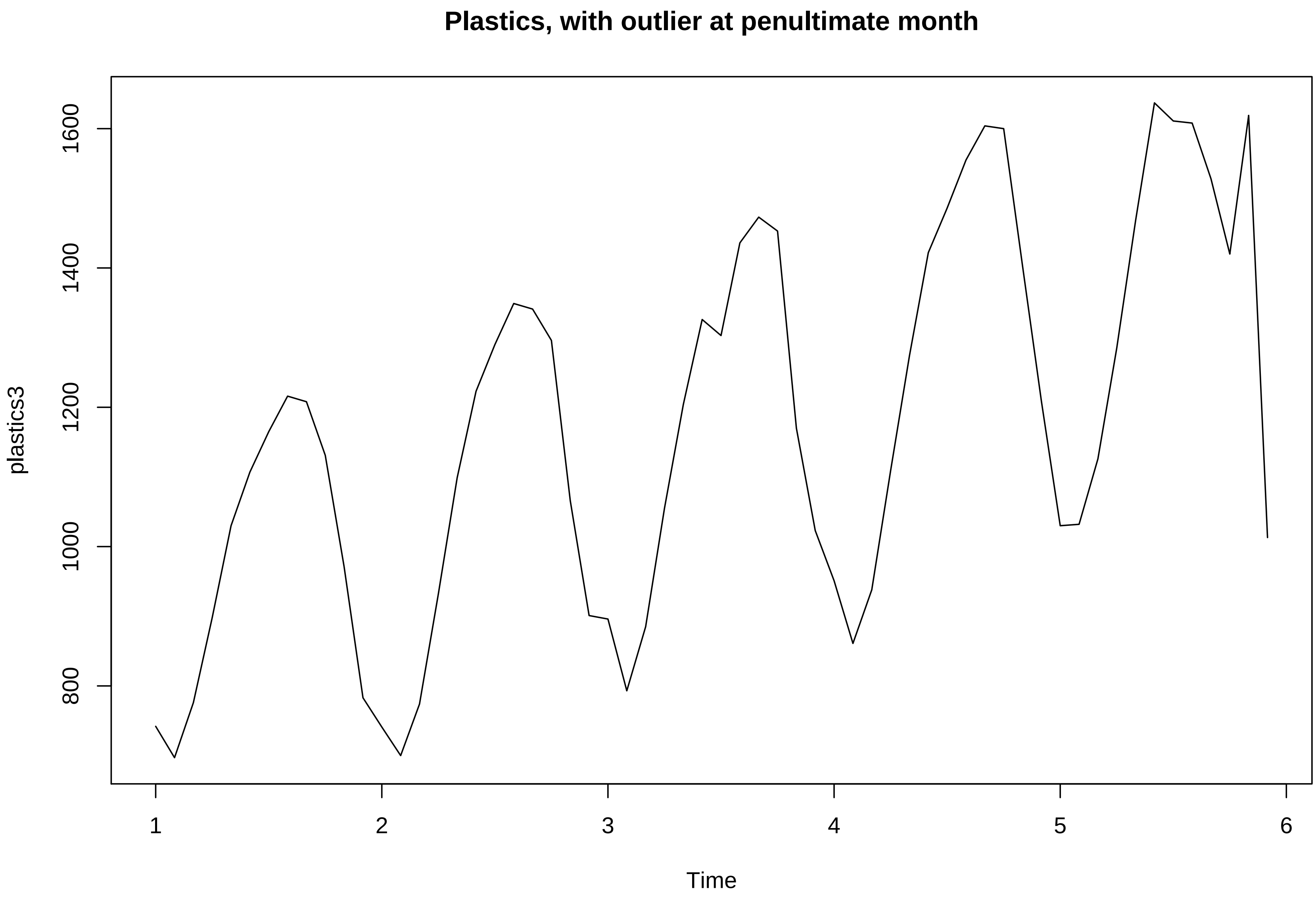
What is the effect of the outlier? The seasonally adjusted series shows a large spike in the corresponding month.

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

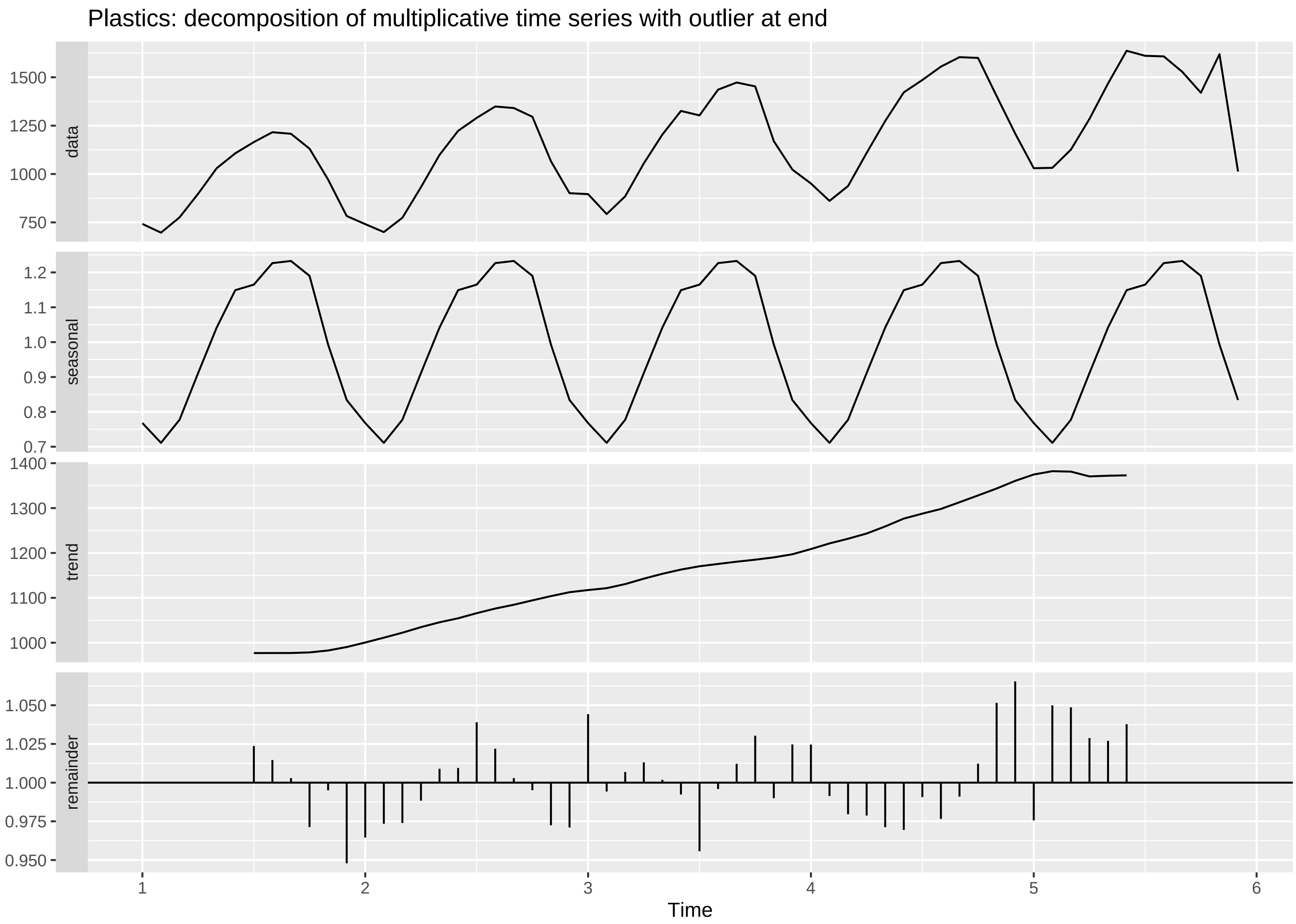
```
plastics3 <- plastics
## changing November of year 5
plastics3[59] <- plastics3[59]+500
plastics3
```

```
##    Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1  742  697  776  898 1030 1107 1165 1216 1208 1131  971  783
## 2  741  700  774  932 1099 1223 1290 1349 1341 1296 1066  901
## 3  896  793  885 1055 1204 1326 1303 1436 1473 1453 1170 1023
## 4  951  861  938 1109 1274 1422 1486 1555 1604 1600 1403 1209
## 5 1030 1032 1126 1285 1468 1637 1611 1608 1528 1420 1619 1013
```

```
plot(plastics3, main="Plastics, with outlier at penultimate month")
```


Plastics, with outlier at penultimate month
plastics3
800
1000
1200
1400
1600
1
2
3
4
5
6
Time

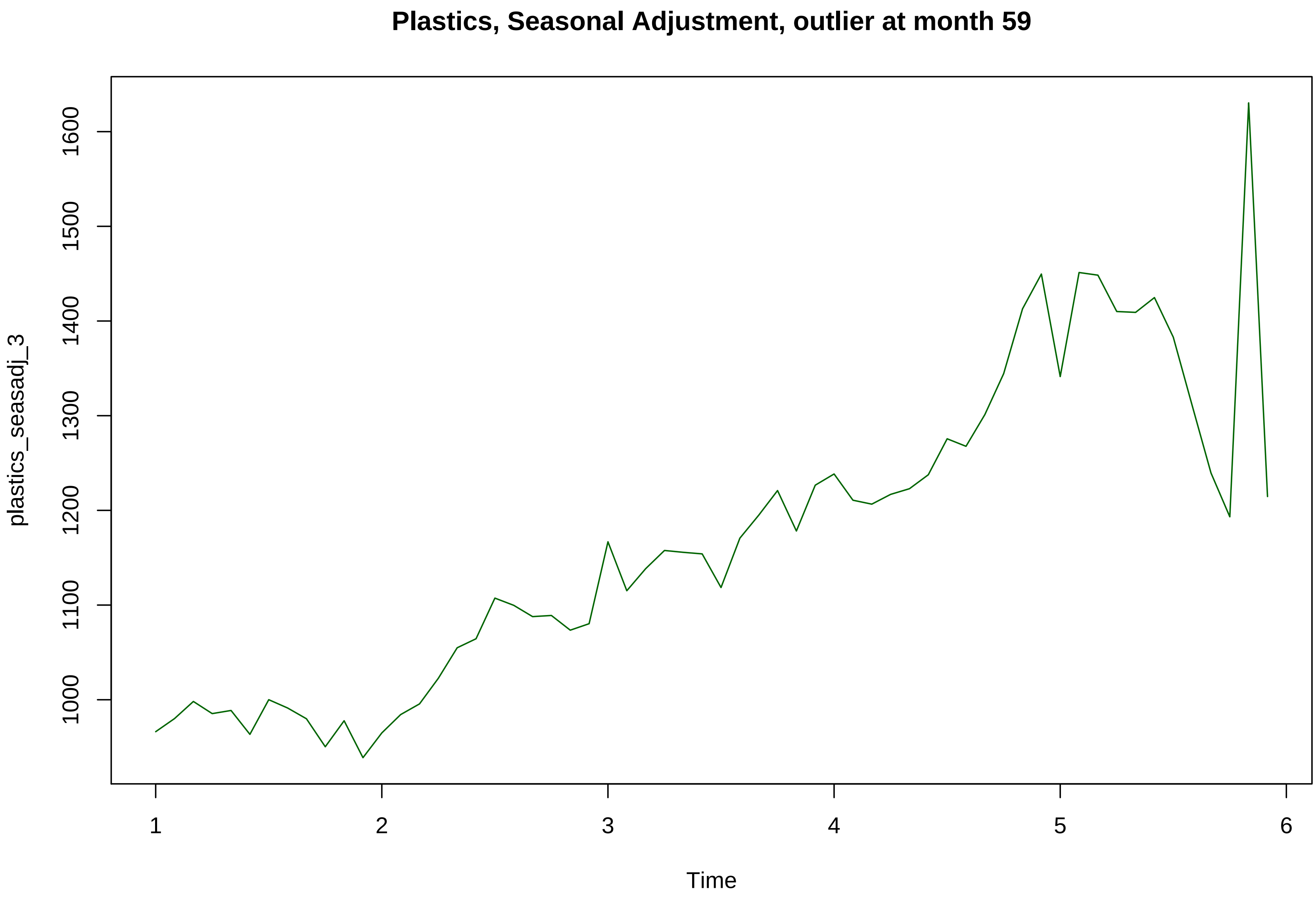
```r
## decompose the perturbed series plastics3
decompose_plastics_3 <- decompose(plastics3, type = "multiplicative")
# autoplot
autoplot(decompose_plastics_3) +
  ggtitle("Plastics: decomposition of multiplicative time series with outlier at end")
```


Plastics: decomposition of multiplicative time series with outlier at end
data
1500
1250
1000
750
seasonal
1.2
1.1
1.0
0.9
0.8
0.7
trend
1400
1300
1200
1100
1000
remainder
1.050
1.025
1.000
0.975
0.950
1
2
3
4
5
6
Time

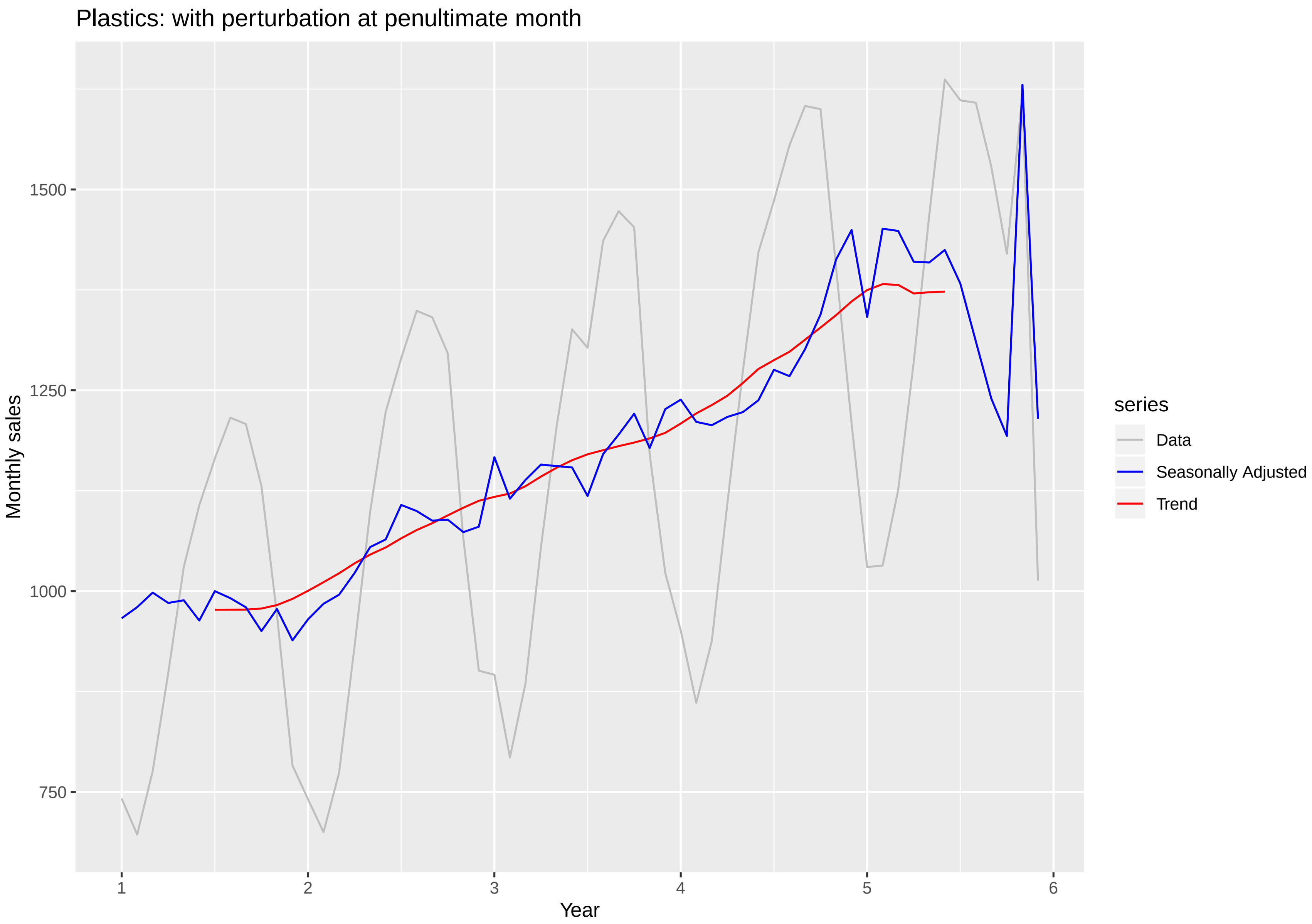
```r
plastics_seasadj_3 <- seasadj(decompose_plastics_3)
plastics_seasadj_3
```

```
##         Jan       Feb       Mar       Apr       May       Jun       Jul
## 1  966.2659  980.1298  998.2016  985.3736  988.6804  963.4565 1000.0572
## 2  964.9636  984.3484  995.6289 1022.6817 1054.9124 1064.4149 1107.3595
## 3 1166.8116 1115.1261 1138.4129 1157.6493 1155.7002 1154.0590 1118.5189
## 4 1238.4351 1210.7486 1206.5891 1216.9034 1222.8921 1237.6108 1275.6095
## 5 1341.3125 1451.2108 1448.4214 1410.0279 1409.1095 1424.7320 1382.9117
##         Aug       Sep       Oct       Nov       Dec
## 1  991.3050  979.9301  950.3610  977.8181  938.8317
## 2 1099.7290 1087.8198 1089.0078 1073.4852 1080.3159
## 3 1170.6530 1194.8982 1220.9323 1178.2154 1226.5962
## 4 1267.6639 1301.1655 1344.4540 1412.8515 1449.6137
## 5 1310.8705 1239.5143 1193.2030 1630.3682 1214.6060
```

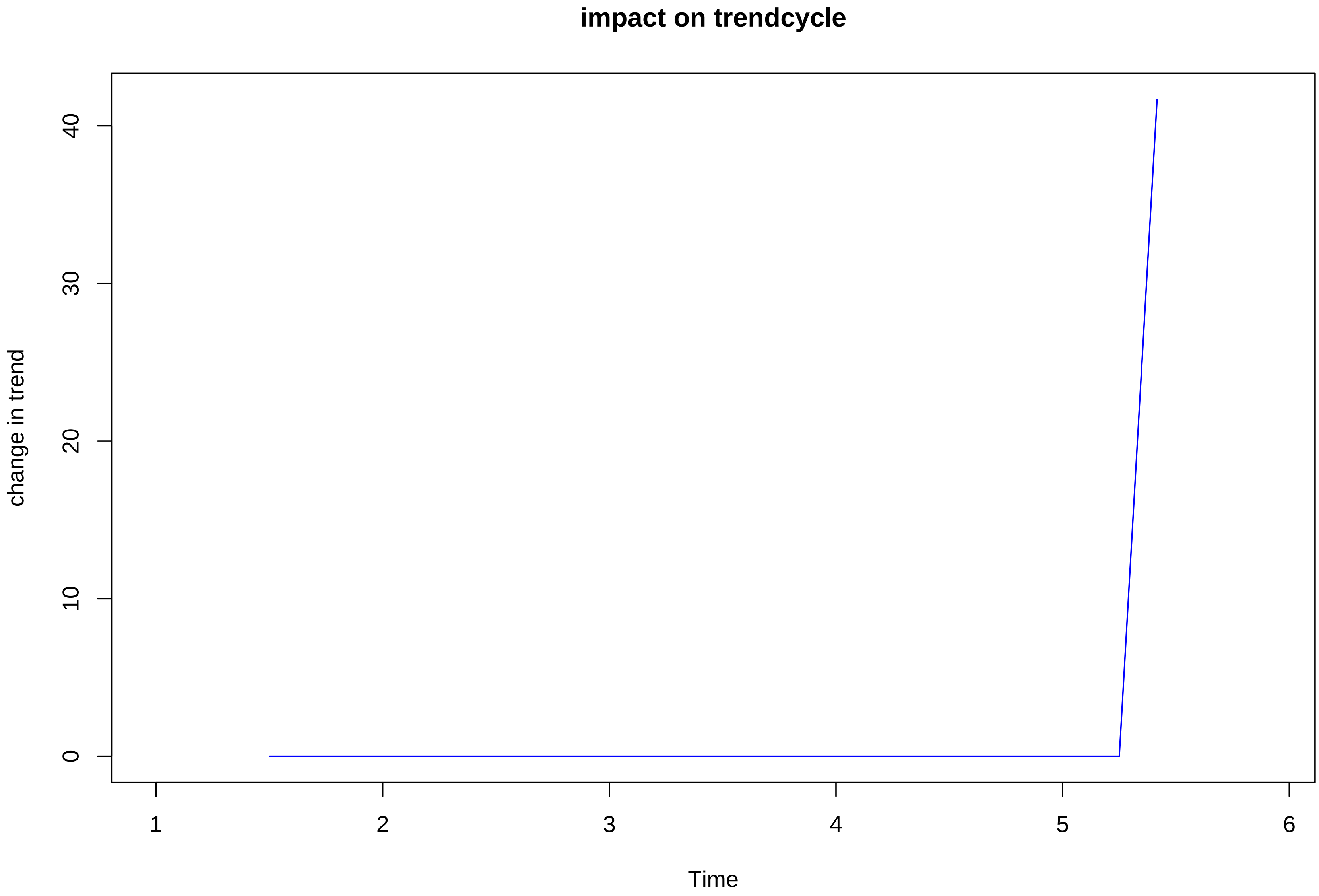
```r
plot(plastics_seasadj_3,
     main="Plastics, Seasonal Adjustment, outlier at month 59",
     col="darkgreen")
```


Plastics, Seasonal Adjustment, outlier at month 59
plastics_seasadj_3
1000
1100
1200
1300
1400
1500
1600
1
2
3
4
5
6
Time

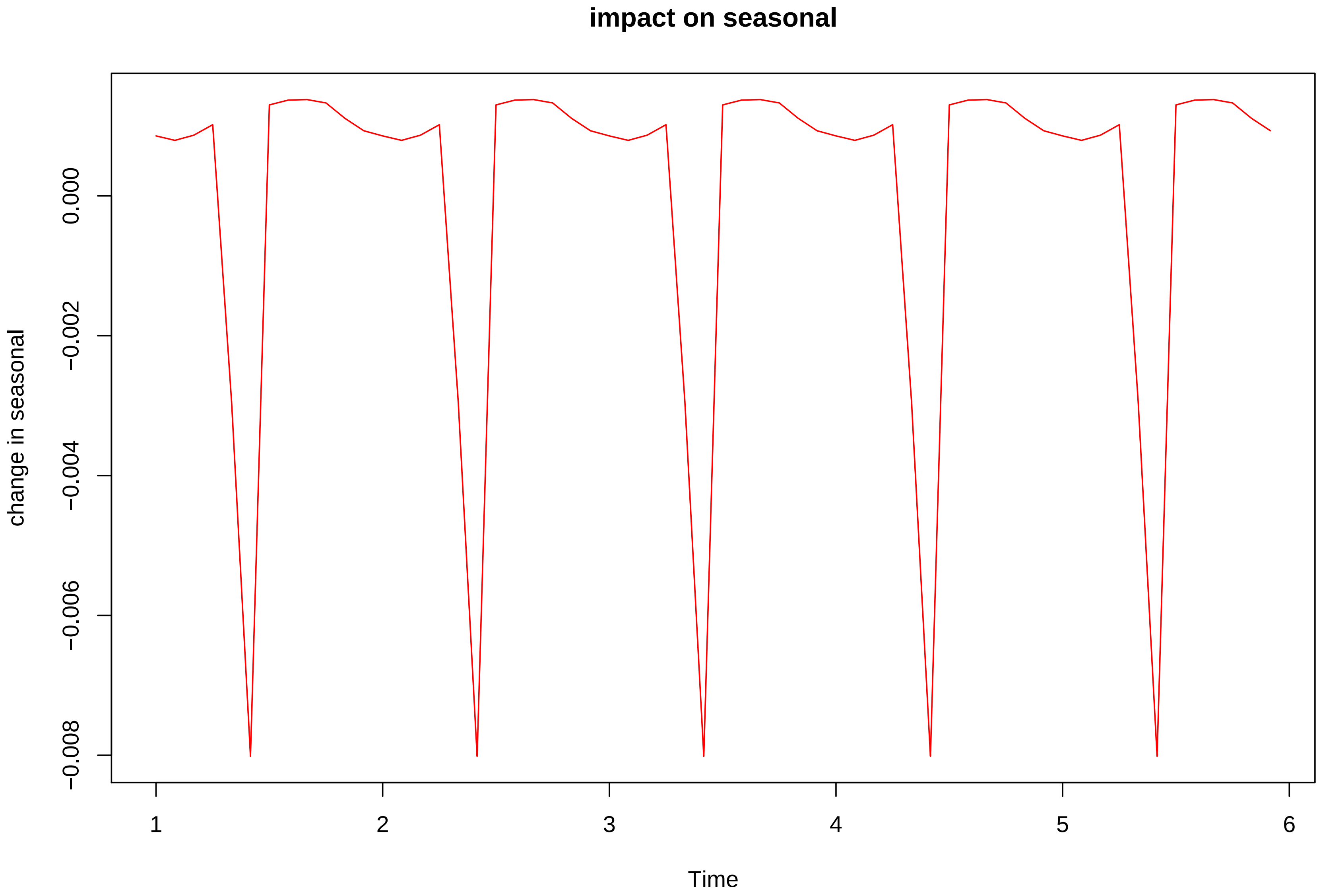
```
autoplot(plastics3, series="Data") +
  autolayer(trendcycle(decompose_plastics_3), series="Trend") +
  autolayer(seasadj(decompose_plastics_3), series="Seasonally Adjusted") +
  xlab("Year") + ylab("Monthly sales") +
  ggtitle("Plastics: with perturbation at penultimate month") +
  scale_colour_manual(values=c("gray","blue","red"),
             breaks=c("Data","Seasonally Adjusted","Trend"))
```


Plastics: with perturbation at penultimate month
Monthly sales
1500
1250
1000
750
1
2
3
4
5
6
Year
series
Data
Seasonally Adjusted
Trend

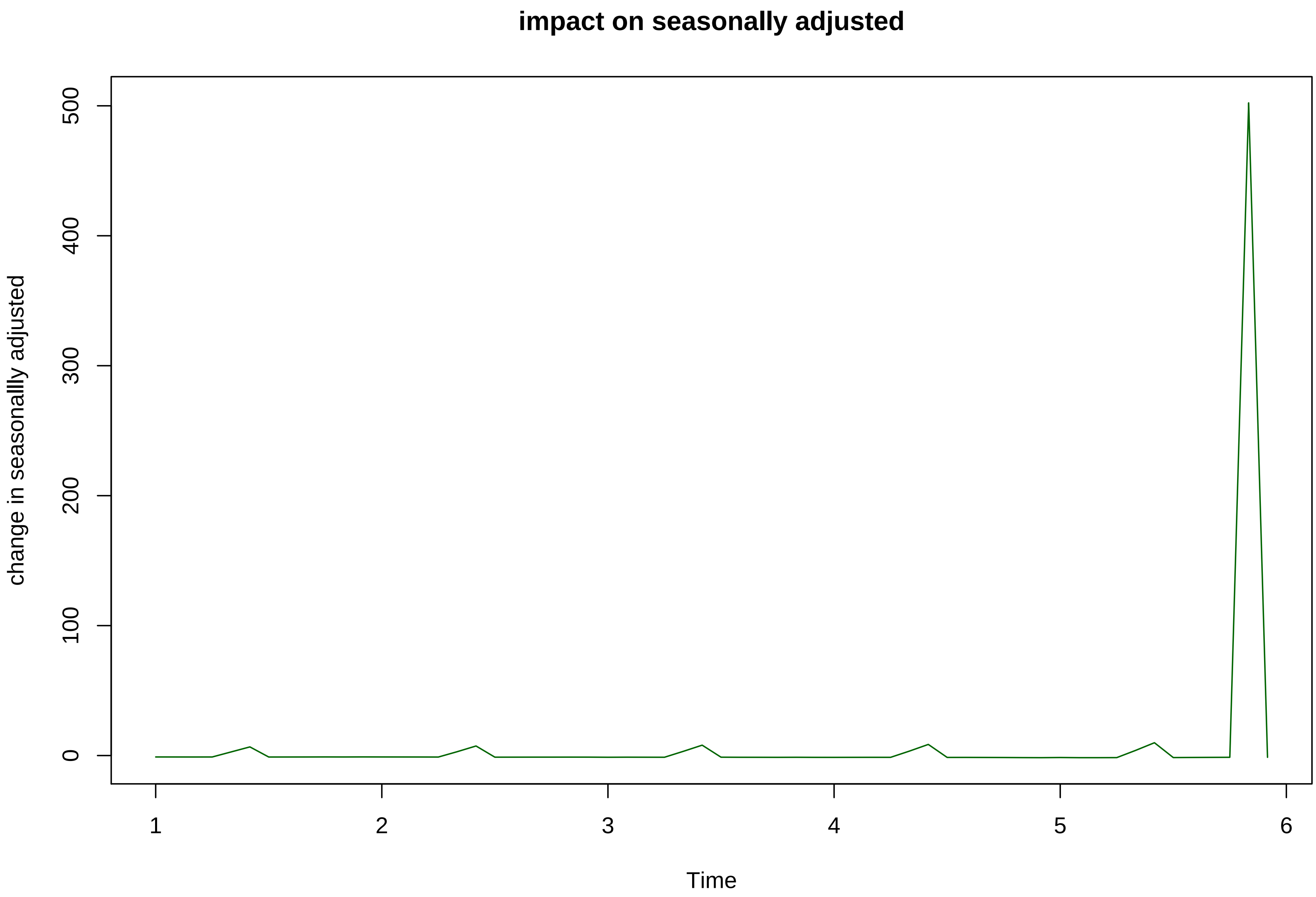
```r
## impact on trendcycle
plot(trendcycle(decompose_plastics_3)-trendcycle(decompose_plastics),
     ylab="change in trend", main="impact on trendcycle", col="blue")
```


impact on trendcycle
change in trend
0
10
20
30
40
1
2
3
4
5
6
Time

```r
## impact on seasonal
plot(seasonal(decompose_plastics_3)-seasonal(decompose_plastics),
     ylab="change in seasonal", main="impact on seasonal", col="red")
```


impact on seasonal
change in seasonal
0.000
−0.002
−0.004
−0.006
−0.008
1
2
3
4
5
6
Time

```r
## impact on seasonally adjusted
plot(seasadj(decompose_plastics_3)-seasadj(decompose_plastics),
     ylab="change in seasonallly adjusted",
     main="impact on seasonally adjusted", col="darkgreen")
```

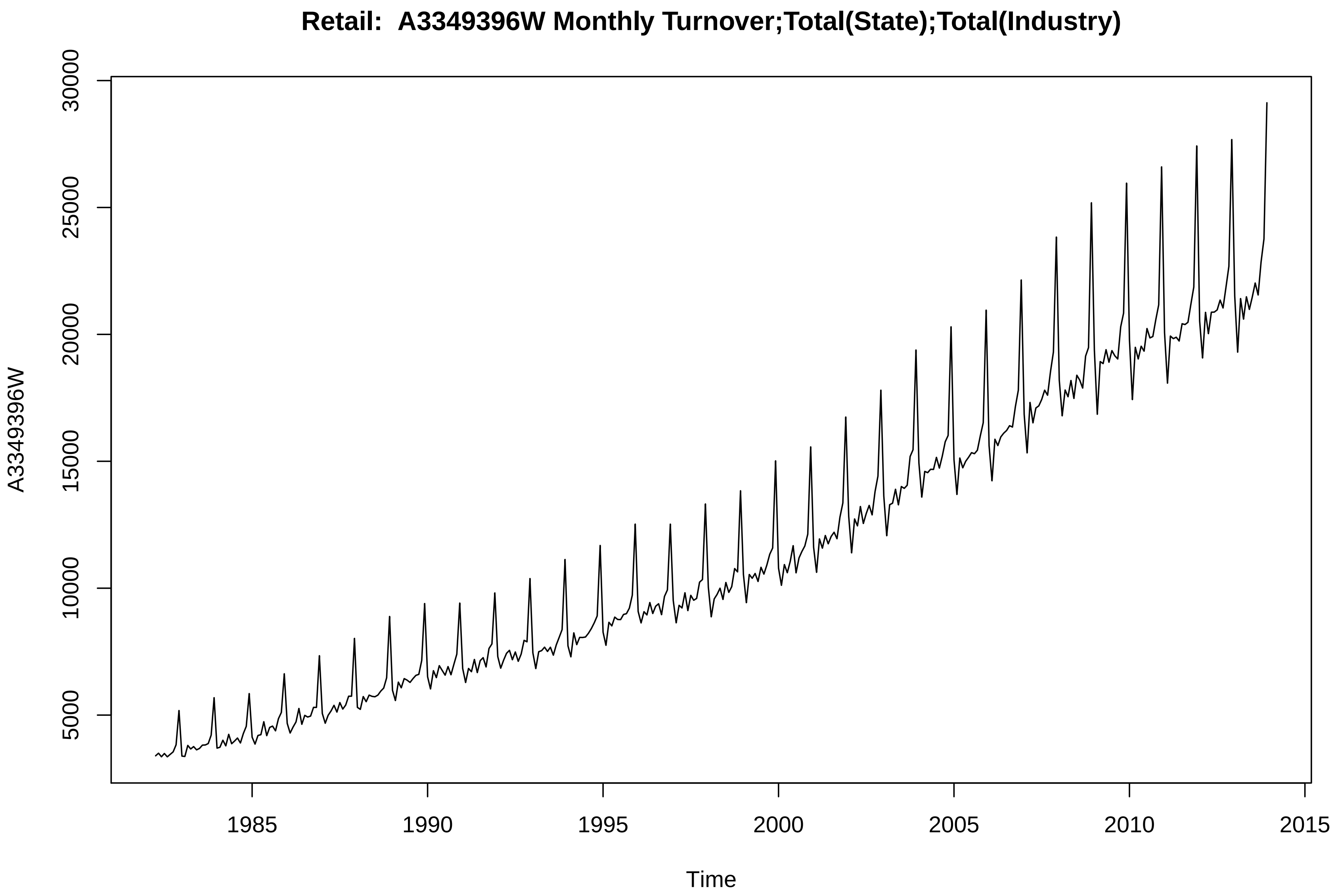

impact on seasonally adjusted
change in seasonallly adjusted
0
100
200
300
400
500
1
2
3
4
5
6
Time

When the perturbation occurs in the middle of the timeseries, the trendcycle is elevated for a year, and then falls back down. The seasonal adjustment at the month of perturbation (June of 3rd year) only reflects about 63 percent of the shock (314 points out of 500), while the seasonal adjustment for the corresponding month in theother years drops by 14 to 21 percent of the shock.

When the perturbation occurs near the end of the timeseries, the trendcycle is only adjusted starting from 6 months before the shock; because this method does not compute trendcycle during the first or last 6 months of the series, only 2 months are affected (i.e., months 53 and 54 for a shock in month 59). The seasonal adjustment in month 59 corresponds exceeds the amount of the shock (502 points adjustment vs. a shock of 500) while the other months receive a negligible adjustment.

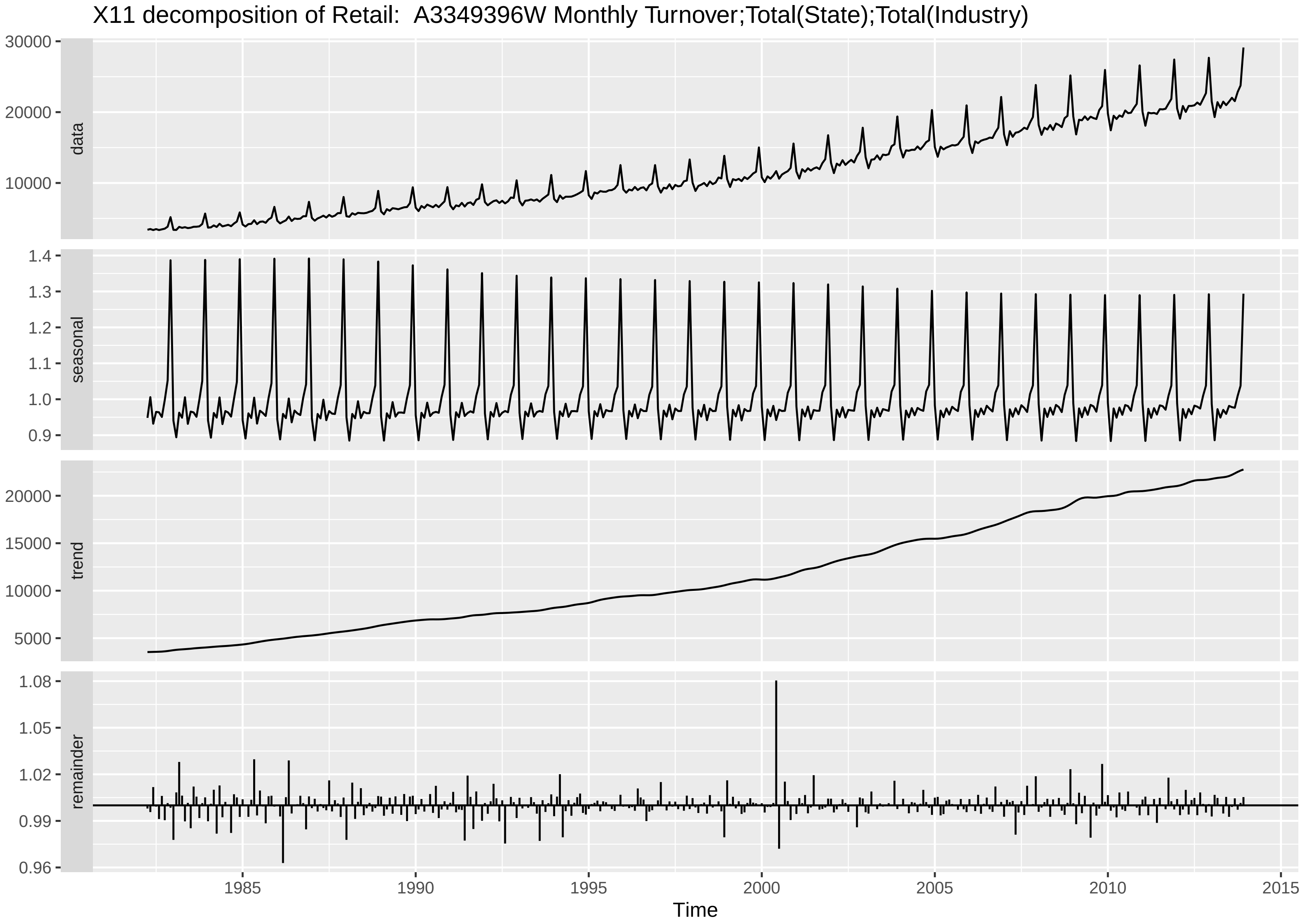
6.3 Recall your `retail` time series data (from Exercise 3 in Section 2.10).

```r
retaildata <- readxl::read_excel("retail.xlsx", skip=1)
mycode <- "A3349396W"
mytitle <-  "Monthly Turnover;Total(State);Total(Industry)"
mymain <- paste("Retail: ", mycode,mytitle)
retail_ts <- ts(retaildata[,mycode],frequency=12, start=c(1982,4))
# plot the data to see if there is a trend or seasonality.

plot(retail_ts, main=mymain)
```

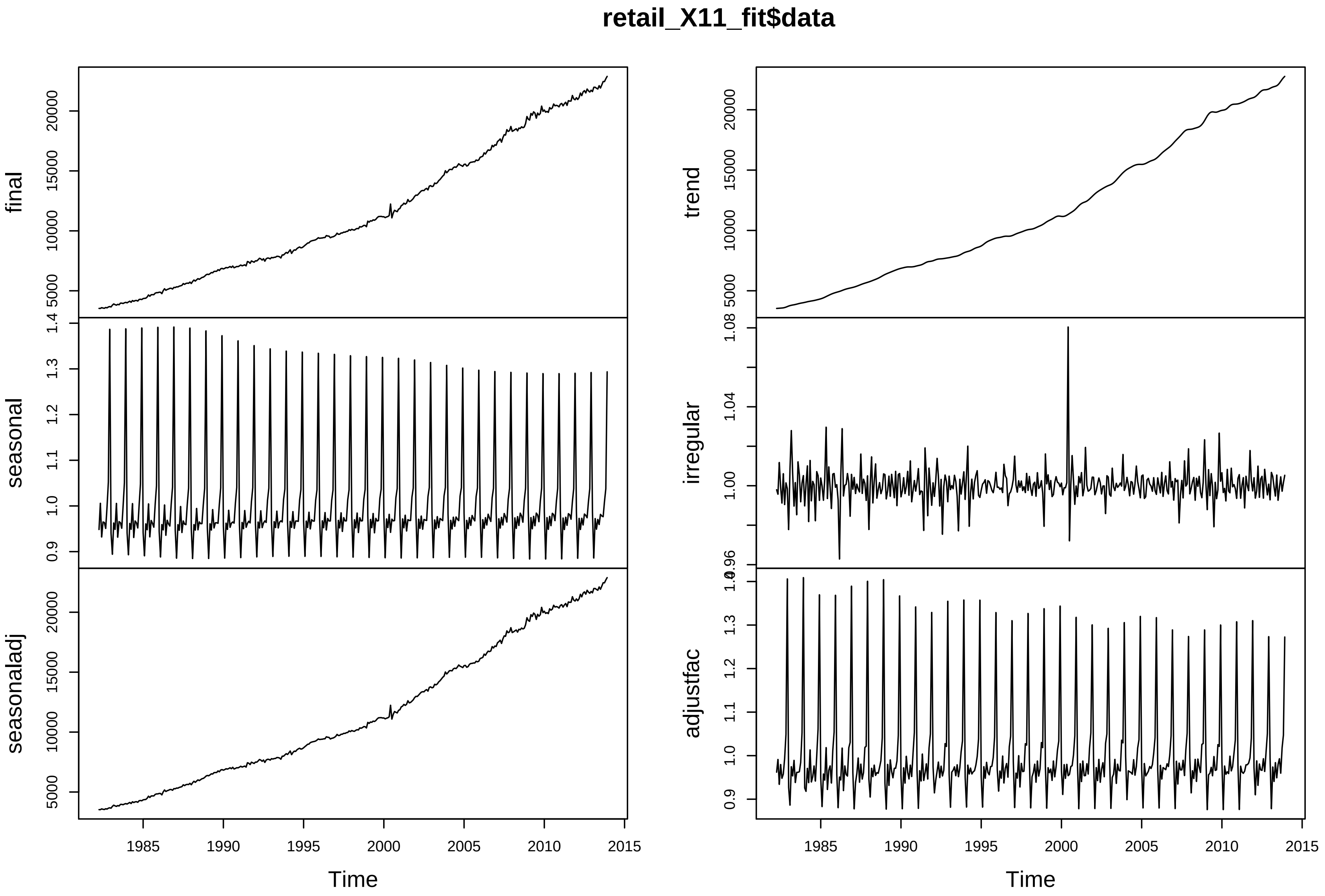

Retail: A3349396W Monthly Turnover;Total(State);Total(Industry)
A3349396W
5000
10000
15000
20000
25000
30000
1985
1990
1995
2000
2005
2010
2015
Time

Decompose the series using X11.

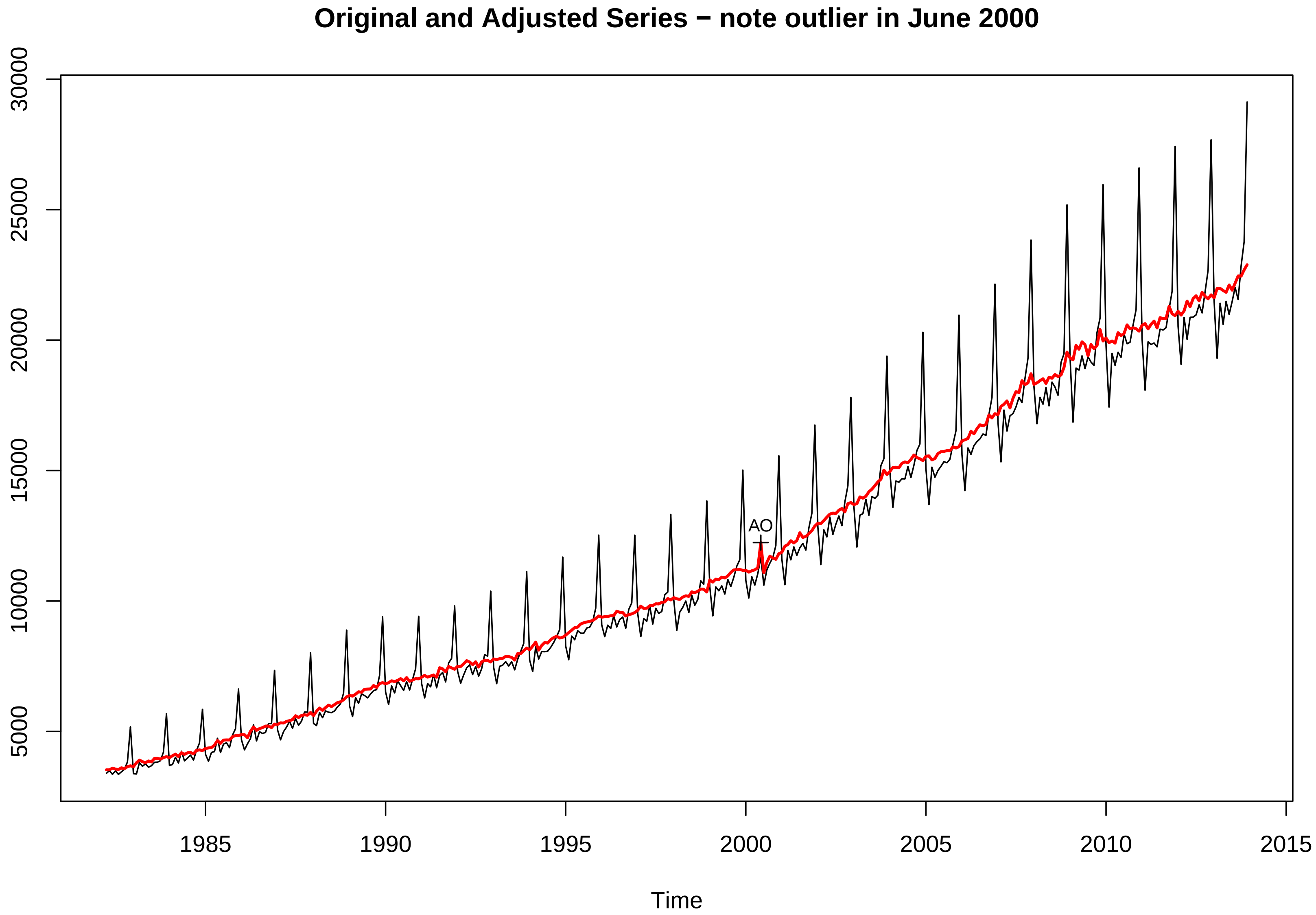
```r
library(seasonal)
retail_ts %>% seas(x11="") -> retail_X11_fit

## Autoplot of fit
autoplot(retail_X11_fit) +
  ggtitle(paste("X11 decomposition of",mymain))
```


X11 decomposition of Retail: A3349396W Monthly Turnover;Total(State);Total(Industry)
data
30000
20000
10000
seasonal
1.4
1.3
1.2
1.1
1.0
0.9
trend
20000
15000
10000
5000
remainder
1.08
1.05
1.02
0.99
0.96
1985
1990
1995
2000
2005
2010
2015
Time

```
## Plot of 6 fit data series
plot(retail_X11_fit$data)
```

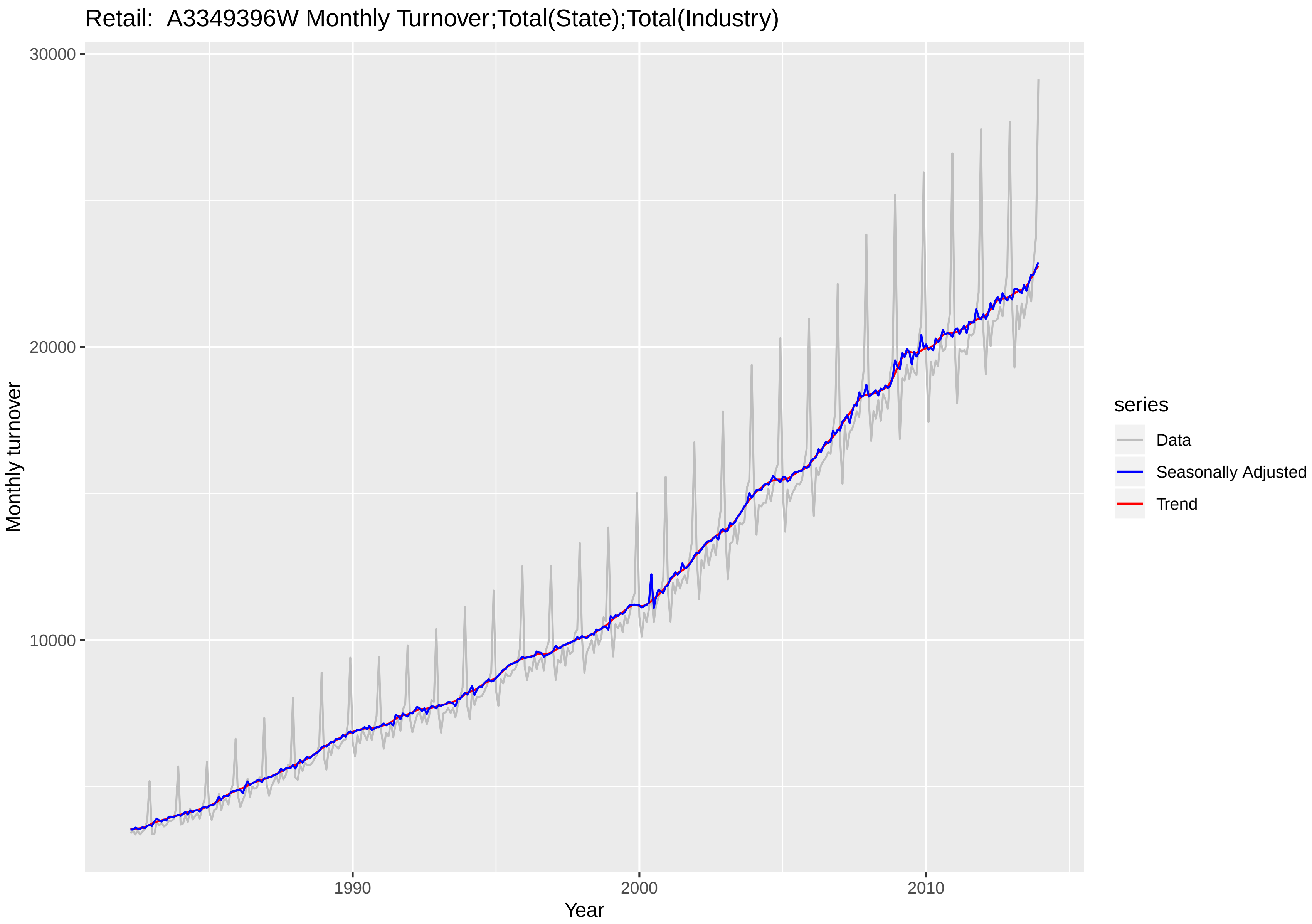

retail_X11_fit$data
final
5000
10000
15000
20000
seasonal
0.9
1.0
1.1
1.2
1.3
1.4
seasonaladj
5000
10000
15000
20000
1985
1990
1995
2000
2005
2010
2015
Time
trend
5000
10000
15000
20000
irregular
0.96
1.00
1.04
1.08
adjustfac
0.9
1.0
1.1
1.2
1.3
1.4
1985
1990
1995
2000
2005
2010
2015
Time

```r
## Plot of the dataframe
plot(retail_X11_fit, main="Original and Adjusted Series - note outlier in June 2000")
```



Original and Adjusted Series − note outlier in June 2000

```r
##Autoplot of data, trend, and seasonally adjusted data

autoplot(retail_ts, series="Data") +
  autolayer(trendcycle(retail_X11_fit), series="Trend") +
  autolayer(seasadj(retail_X11_fit), series="Seasonally Adjusted") +
  xlab("Year") + ylab("Monthly turnover") +
  ggtitle(mymain) +
  scale_colour_manual(values=c("gray","blue","red"),
             breaks=c("Data","Seasonally Adjusted","Trend"))
```


Retail: A3349396W Monthly Turnover;Total(State);Total(Industry)
30000
20000
10000
1990
2000
2010
Year
Monthly turnover
series
Data
Seasonally Adjusted
Trend

Does it reveal any outliers, or unusual features that you had not noticed previously?

There is an outlier in June of 2000.

In most years, the sales drop off from May to June, and then increase in July. (This happens to be the Australian winter.) However, in 2000, the raw data series shows a sharp increase in June 2000, which is not present in other years.