

week1

Chunjie Nan

6/5/2021

2.1

Use the help function to explore what the series gold, woolyrnq and gas represent.

```
library(forecast)
```

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

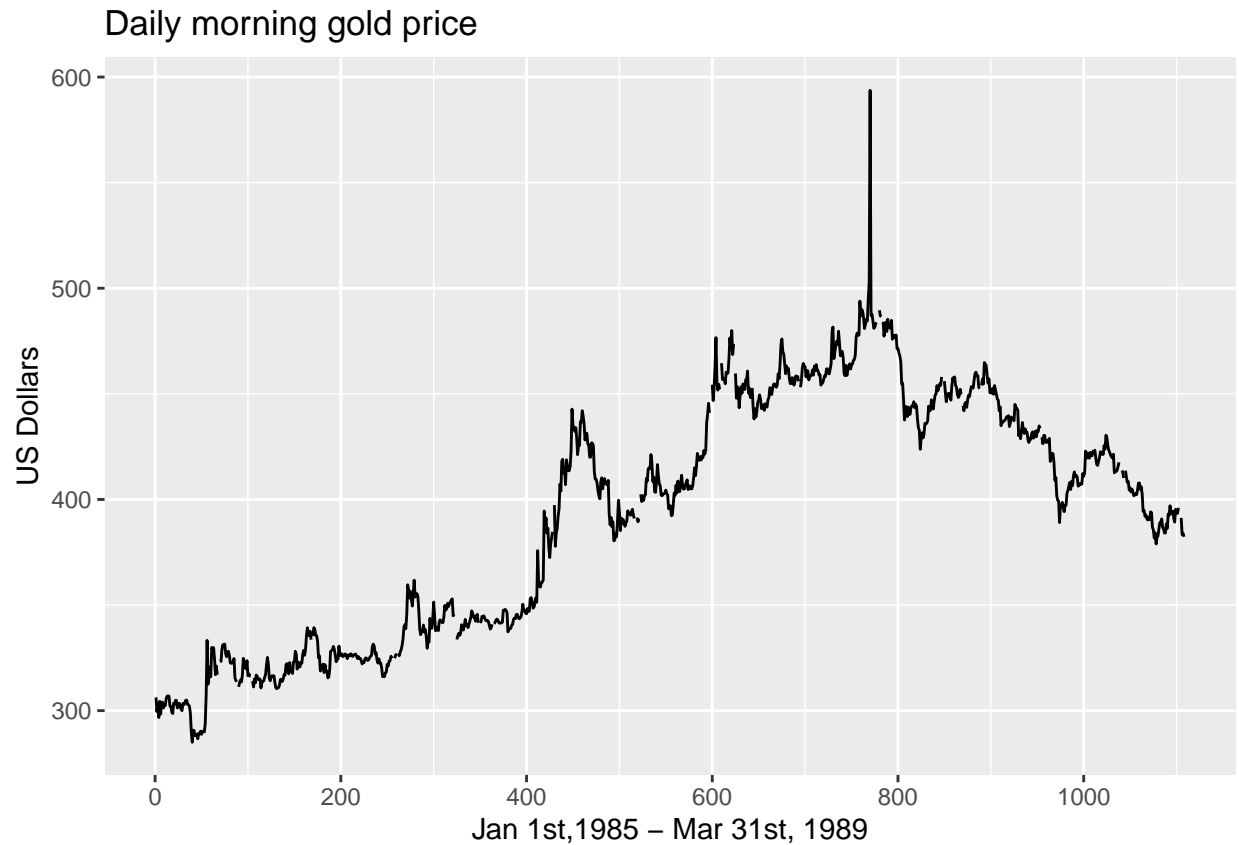
```
library(ggplot2)  
help(gold)      # Daily morning gold prices in US dollars from Jan 1st 1985 to Mar 31st 1989.  
help(woolyrnq)  # Quarterly production of woollen yarn in Australia in tonnes. Mar 1965 to Sep 1994.  
help(gas)       # Australian monthly gas production from 1956 to 1995.
```

a. Use autoplot() to plot each of these in separate plots.

```
# gold  
head(gold)
```

```
## Time Series:  
## Start = 1  
## End = 6  
## Frequency = 1  
## [1] 306.25 299.50 303.45 296.75 304.40 298.35
```

```
autoplot(gold)+  
  ggtitle("Daily morning gold price")+  
  xlab("Jan 1st,1985 - Mar 31st, 1989")+  
  ylab("US Dollars")
```

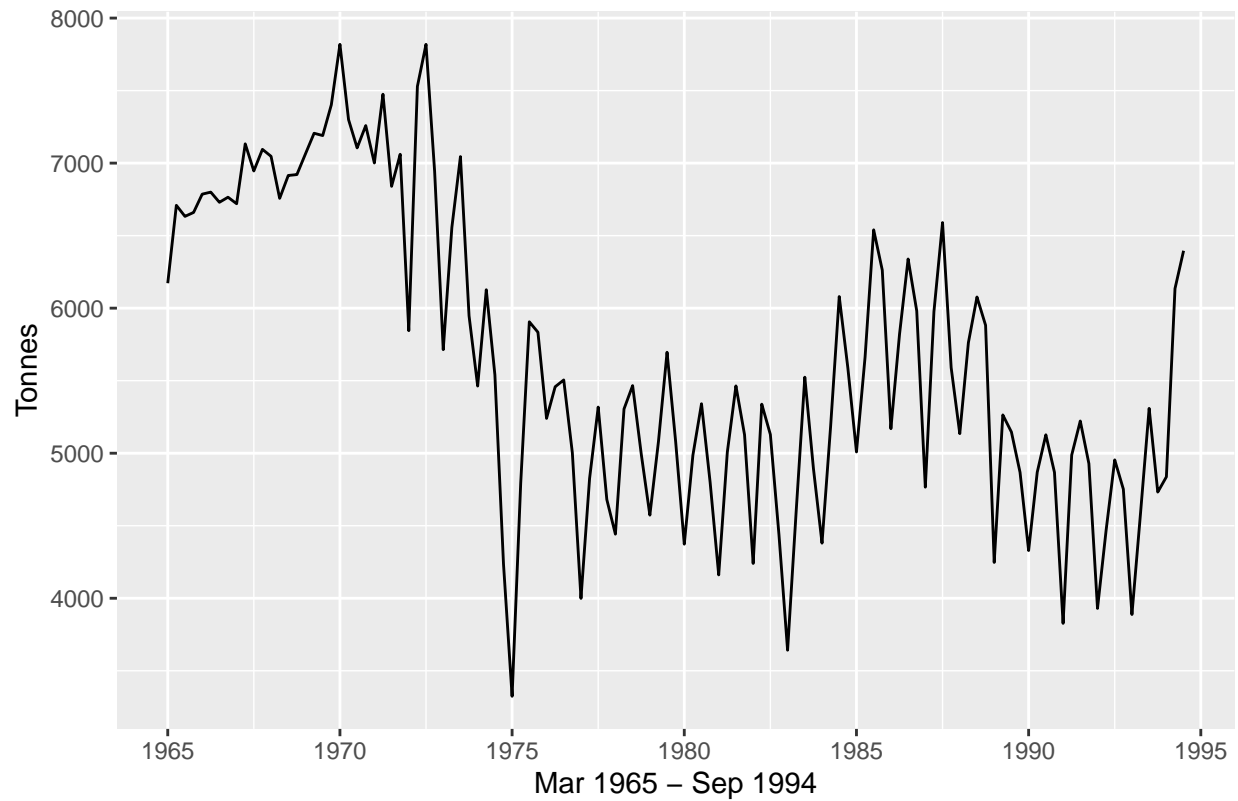


```
# woolen yarn
head(woolyrnq)
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 1965 6172 6709 6633 6660
## 1966 6786 6800
```

```
autoplot(woolyrnq)+
  ggtitle("Quarterly production of woollen yarn in Australia")+
  xlab("Mar 1965 - Sep 1994")+
  ylab("Tonnes")
```

Quarterly production of woollen yarn in Australia

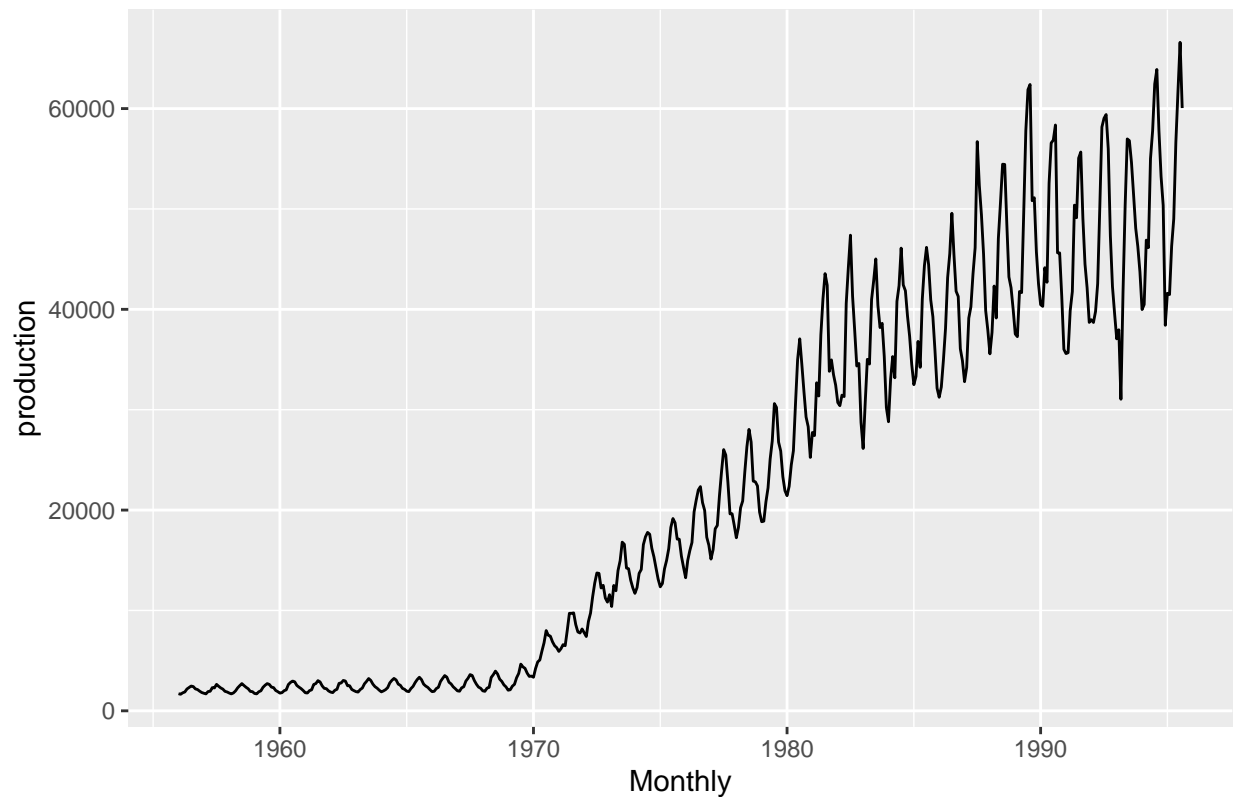


```
# gas
head(gas)
```

```
##      Jan  Feb  Mar  Apr  May  Jun
## 1956 1709 1646 1794 1878 2173 2321
```

```
autoplot(gas)+
  ggtitle("Australian monthly gas production, 1956 to 1995")+
  xlab("Monthly")+
  ylab("production") # No specific unit
```

Australian monthly gas production, 1956 to 1995



b. What is the frequency of each series? Hint: apply the `frequency()` function.

```
frequency(gold)
```

```
## [1] 1
```

```
frequency(woolyrnq)
```

```
## [1] 4
```

```
frequency(gas)
```

```
## [1] 12
```

With `frequency()`, gold has frequency 1, matches the description of daily morning price of gold price. woollen yard has frequency 4, matches the description of quarterly production of woollen yarn. gas has frequency 12, matches the description of monthly gas production.

c. Use `which.max()` to spot the outlier in the gold series. Which observation was it?

```
summary(gold)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's  
##    285.0   337.7   403.2   392.5   443.7   593.7      34
```

```
max(gold, na.rm = T)
```

```
## [1] 593.7
```

```
which.max(gold)
```

```
## [1] 770
```

The Maximum gold price in this series is \$593.7, and it is on the 770th row.

2.3

Download some monthly Australian retail data from the book website. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file.

- a. You can read the data into R with the following script:

```
retaildata <- readxl::read_excel("retail.xlsx", skip=1)
```

The second argument (skip=1) is required because the Excel sheet has two header rows.

- b. Select one of the time series as follows (but replace the column name with your own chosen column):

```
myts <- ts(retaildata[, "A3349335T"],  
           frequency=12, start=c(1982, 4))  
head(myts)
```

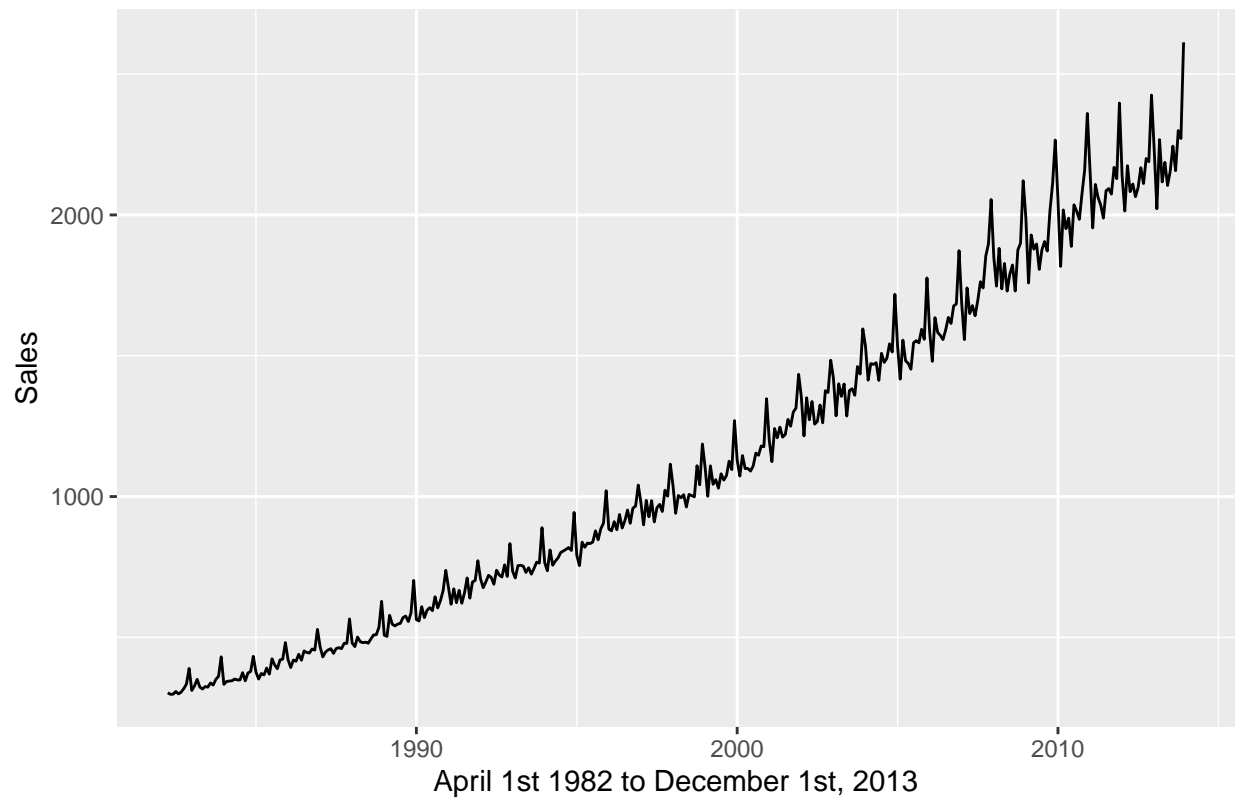
```
##           Apr   May   Jun   Jul   Aug   Sep  
## 1982 303.1 297.8 298.0 307.9 299.2 305.4
```

- c. Explore your chosen retail time series using the following functions:

```
autoplot(), ggseasonplot(), ggsubseriesplot(), gglagplot(), ggAcf()
```

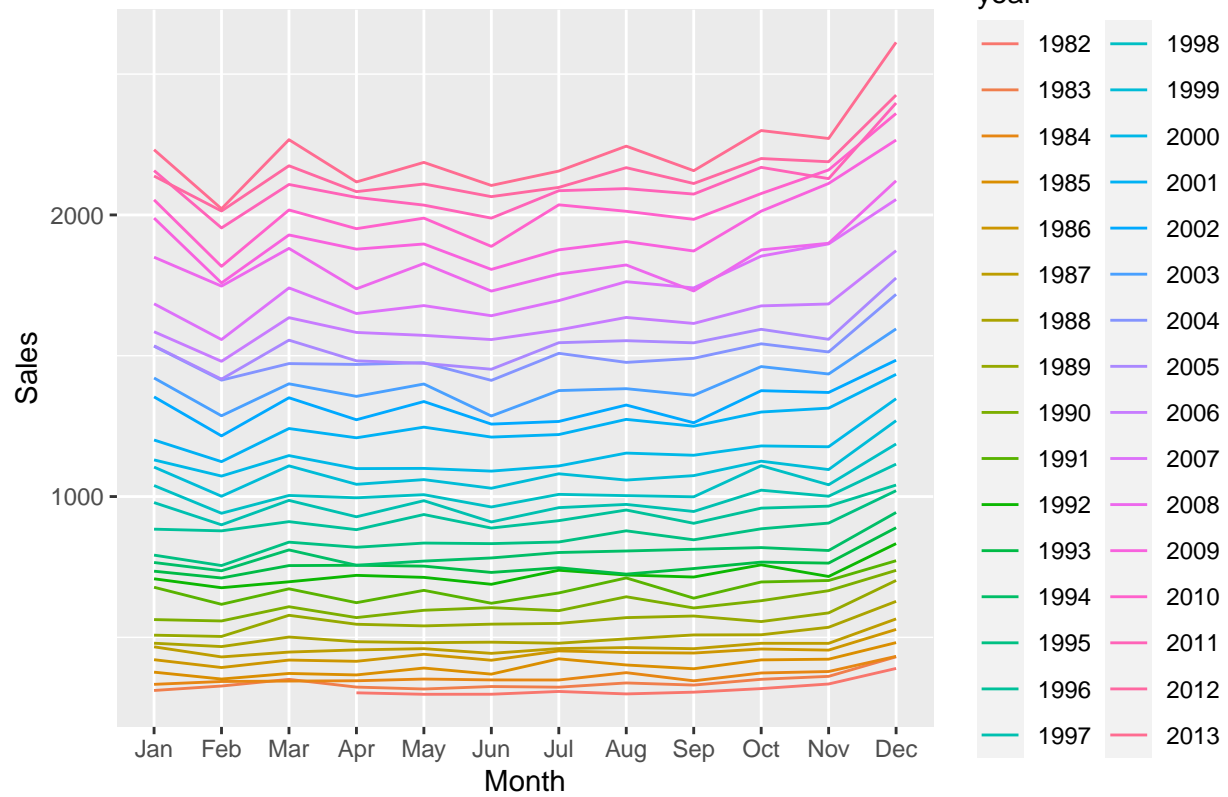
```
autoplot(myts)+  
  ggtitle("Monthly Australian retail Sales")+  
  xlab("April 1st 1982 to December 1st, 2013")+  
  ylab("Sales")
```

Monthly Australian retail Sales

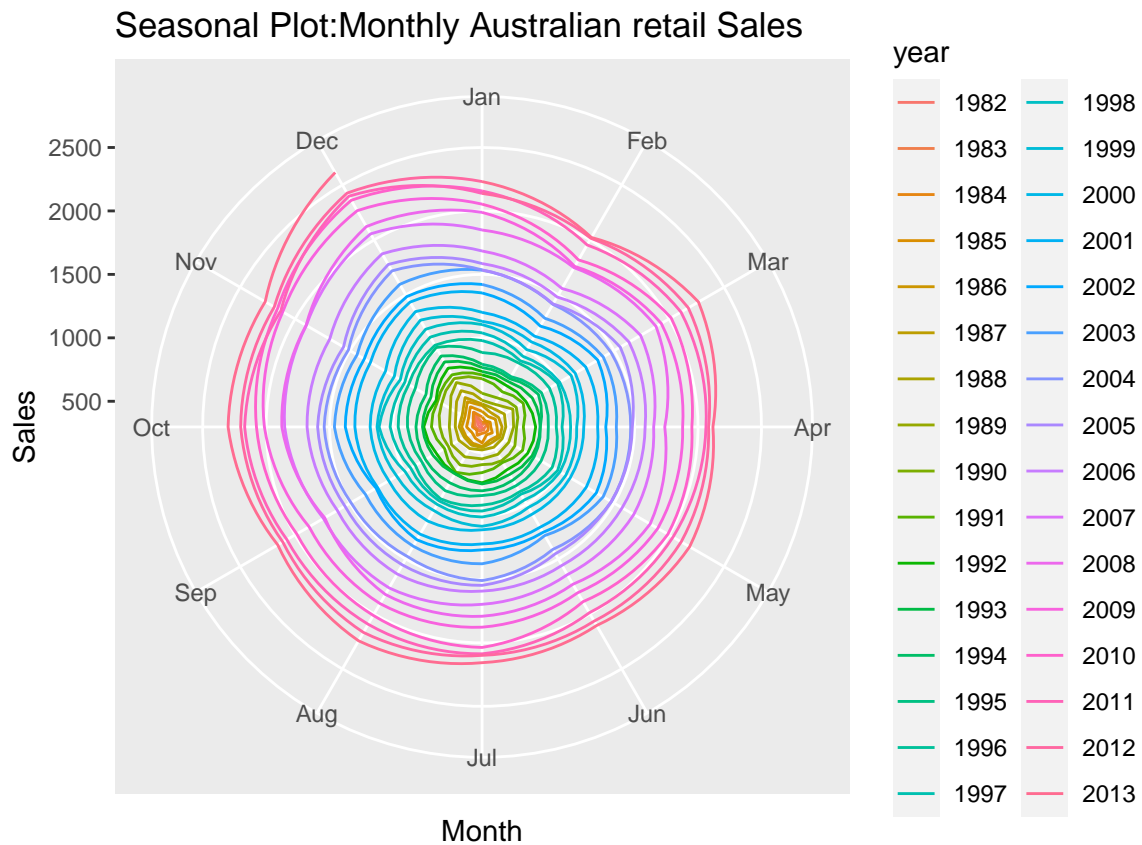


```
ggseasonplot(myts, year.labels = F)+  
  ggtitle("Seasonal Plot:Monthly Australian retail Sales")+  
  ylab("Sales")
```

Seasonal Plot:Monthly Australian retail Sales

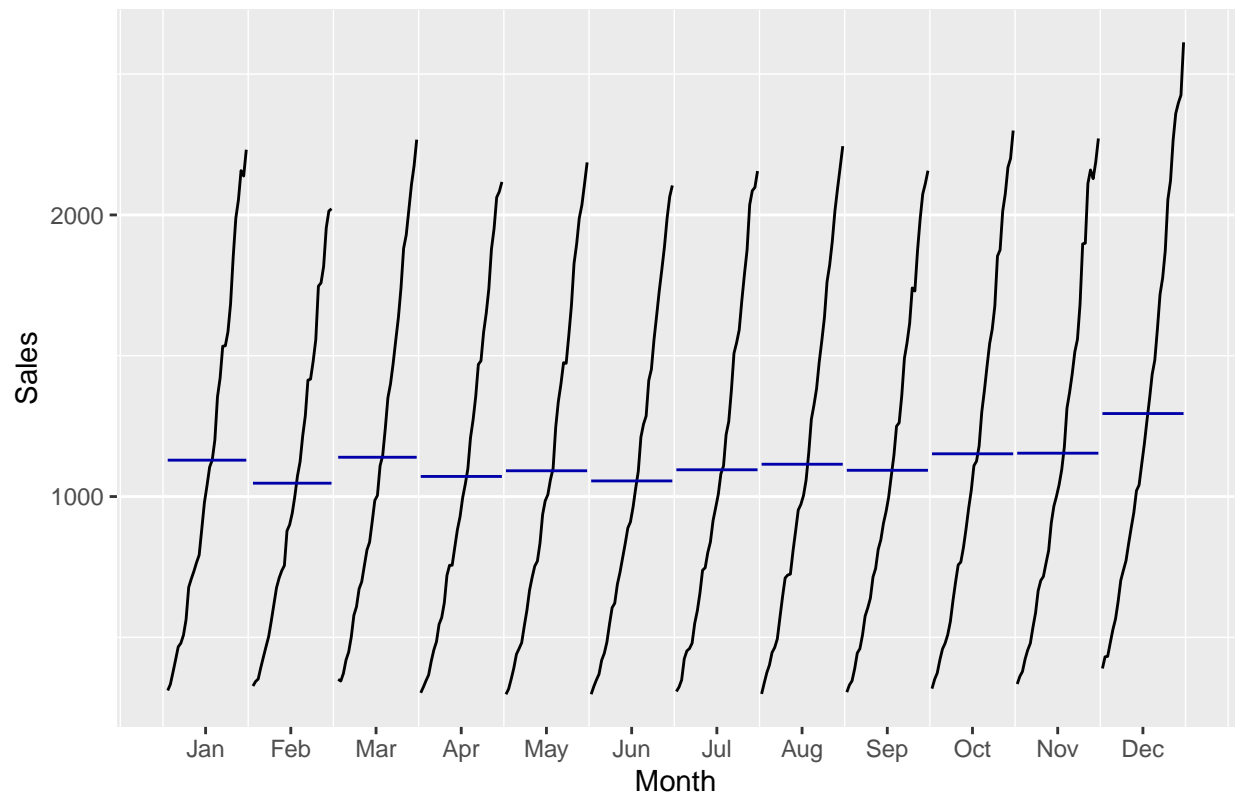


```
ggseasonplot(myts, polar = T)+
  ggtitle("Seasonal Plot:Monthly Australian retail Sales")+
  ylab("Sales")
```

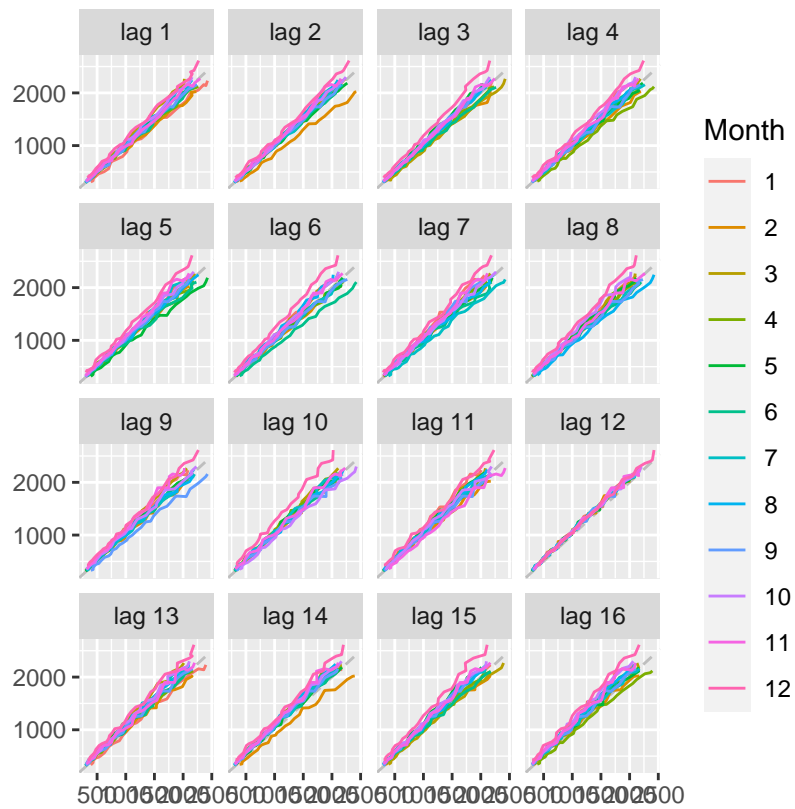


```
ggsubseriesplot(myts)+
  ggtitle("Seasonal Plot:Monthly Australian retail Sales")+
  ylab("Sales")
```

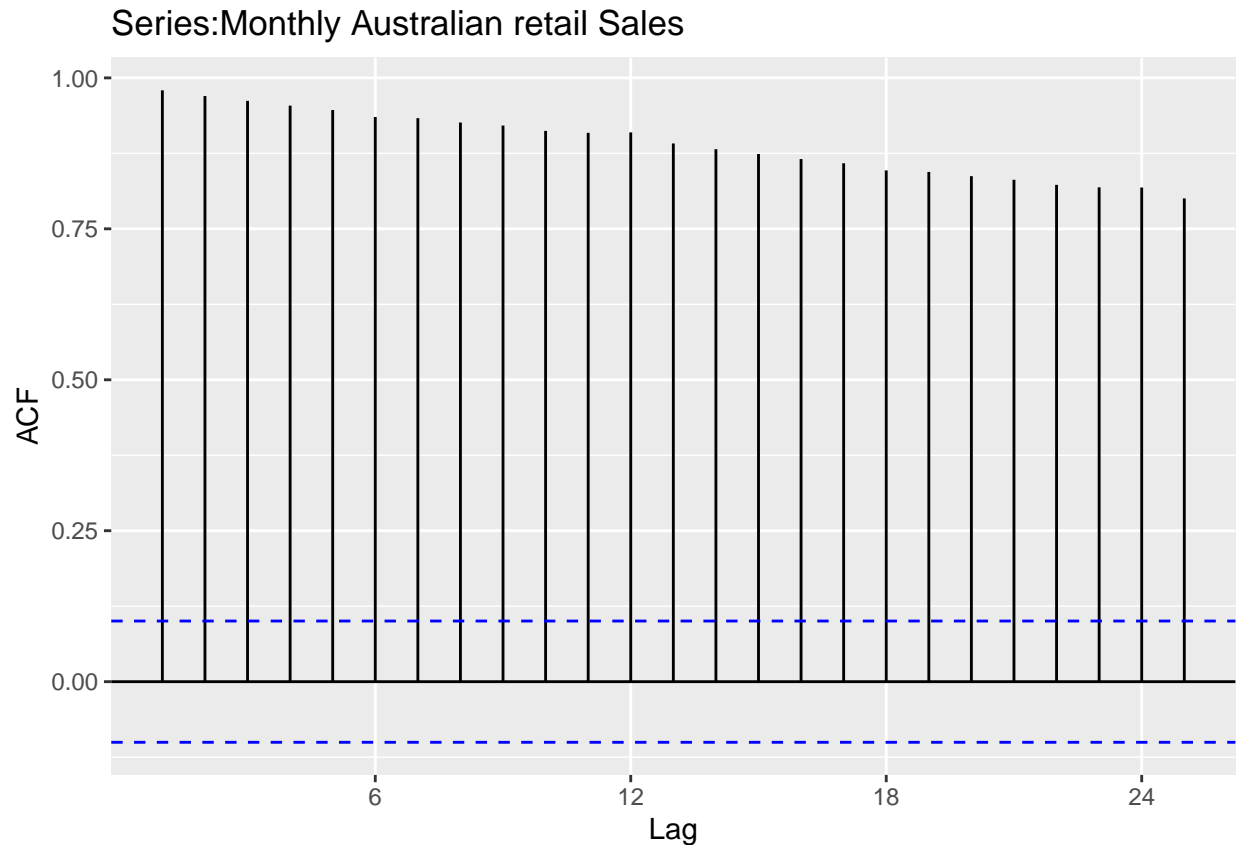

Seasonal Plot:Monthly Australian retail Sales



```
gglagplot(myts)
```



```
ggAcf(myts)+
  ggtitle("Series:Monthly Australian retail Sales")
```



Can you spot any seasonality, cyclicity and Trend? What do you learn about the series?

Seasonality: The seasonal plot tells us that the series has strong seasonality. Dramatic increase of retail selling starts from November and it reaches peaks in December. It is because during the holiday season, includes Thanksgiving, Christmas and New Year, We'd like to buy gifts for families and friends to celebrate holidays. Also, the huge discount and events stimulates the sales of retail stores.

Cyclicity and Trend: With `autoplot()` function, I can assume there is no cyclicity, but have strong trend in this series. For more detail, the `ggAcf()` tells my assumption is correct. If there is cycle exists, it is supposed to show negative autocorrelations among the lags. However, all the lags show as positive, and there is no negative autocorrelation in ACF plot. In ACF plot, the closest lag have the highest positive value and it decreases gradually, it supports this series has strong trend. At lag 7, lag 12, and lag 19 the decrease is getting flat due to the seasonality of good sales.

6.2

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

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

```
library(fpp2)

## -- Attaching packages ----- fpp2 2.4 --

## v fma      2.4      v expsmooth 2.3
```

```
##
```

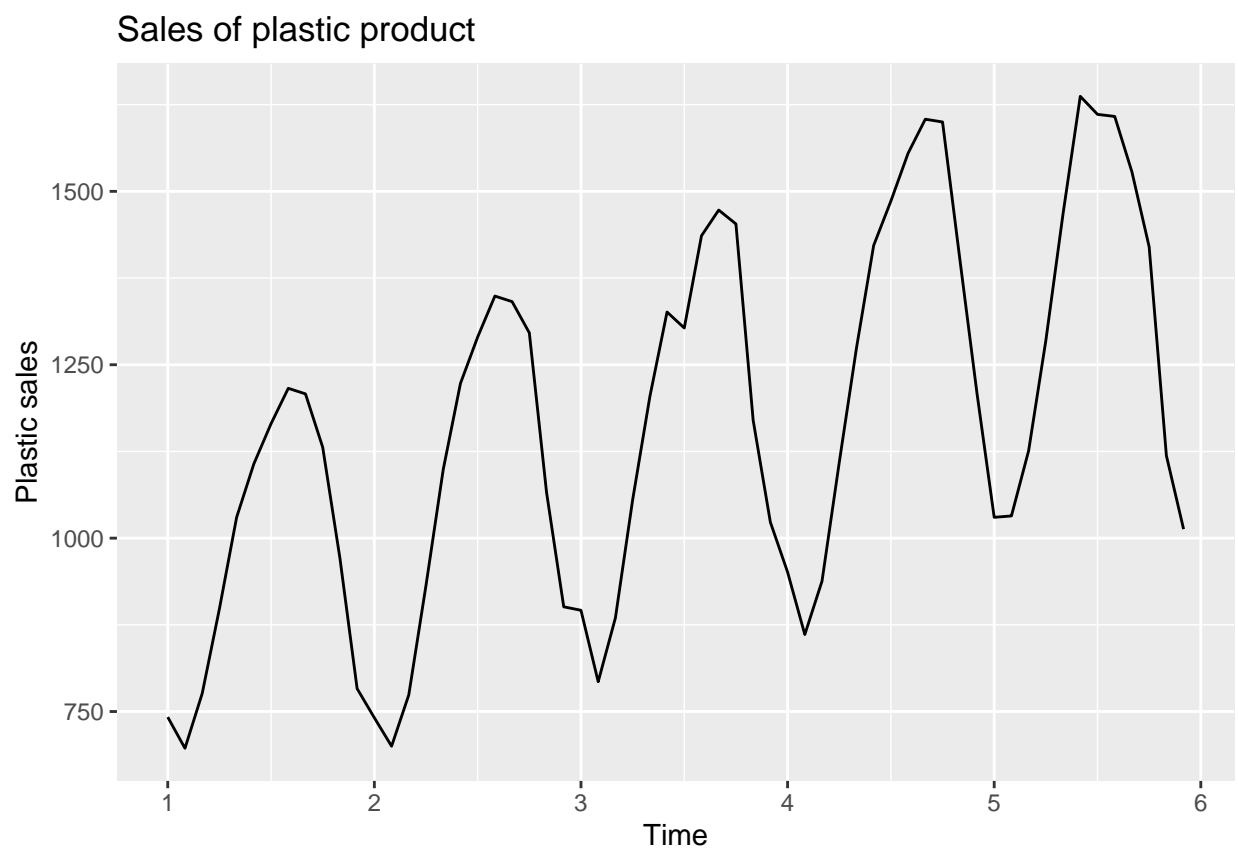
```
plastics
```

```
##      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 1119 1013
```

```
frequency(plastics)
```

```
## [1] 12
```

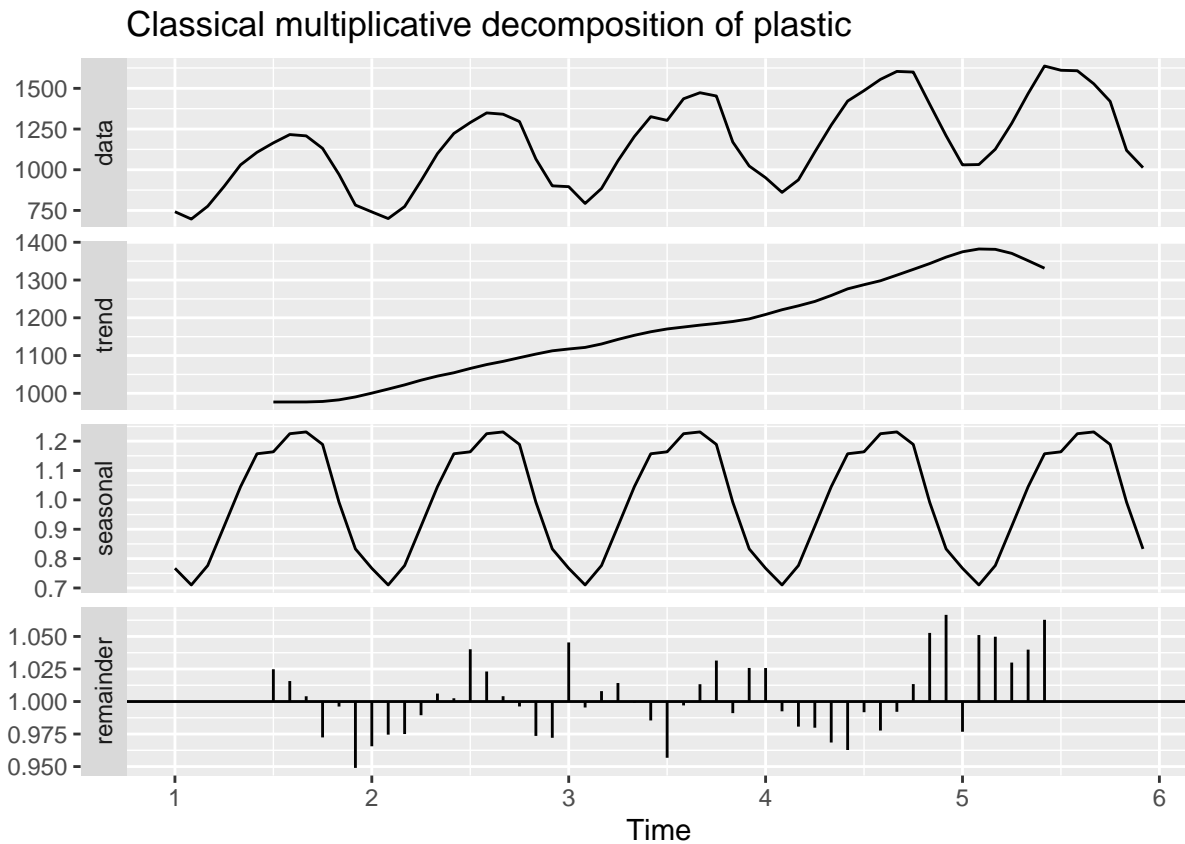
```
autoplot(plastics)+
  ggtitle("Sales of plastic product")+
  ylab("Plastic sales")
```



The plot shows it has strong seasonality with reaching the peak in late summer and early fall. it also has up-trend, the plastic sales increase year by year and refresh the records every year.

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

```
plastics %>% decompose(type = "multiplicative") %>%
  autoplot() + ggtitle("Classical multiplicative decomposition of plastic")
```



The seasonal component has frequency 1 for each year, and the trend tells the increase starts from 1.8 year and turns to decrease at 5.2 years.

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

yes, as the trend is upward overall, even though the trend head seems flat and tail falls, and the seasonality exists every year exactly as we observed in part a.

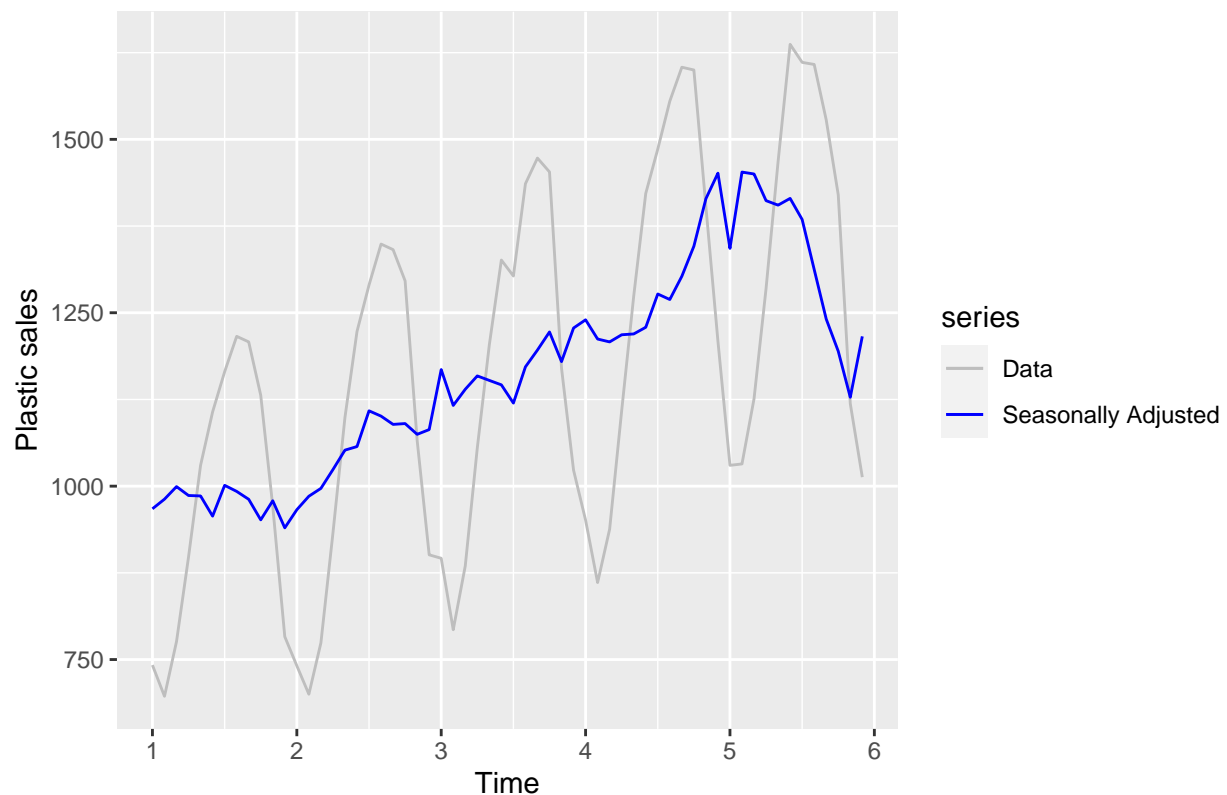
d. Compute and plot the seasonally adjusted data.

use `seasadj()` to find out seasonally adjusted data.

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

autoplot(plastics, series="Data")+
  autolayer(seasadj(mul_decom), series = "Seasonally Adjusted")+
  ggtitle("Seasonally adjusted sales of plastic")+
  ylab("Plastic sales")+
  scale_color_manual(values = c("gray","Blue"), breaks=c("Data", "Seasonally Adjusted"))
```

Seasonally adjusted sales of plastic



The blue line is seasonally adjusted plot, and without the seasonality, the up and down from the original data became a relatively constant uptrend. Mainly, it because the seasonally adjusted is structured with trend-cycle and remainder.

- f. 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?

We are adding 600 on the 45th observation. see how this change affects the seasonally adjusted data.

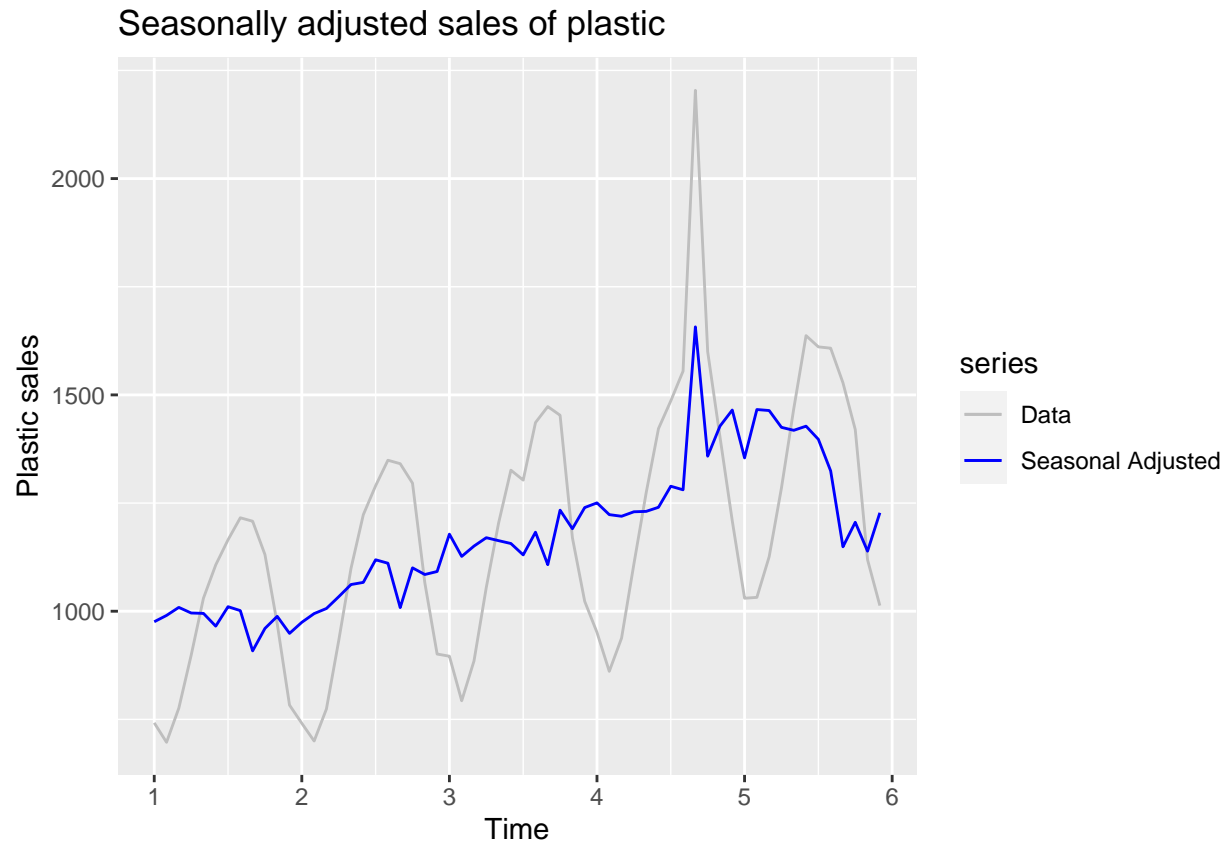
```
plastics[45]
```

```
## [1] 1604
```

```
plastics_600<-plastics
plastics_600[45]<-plastics[45]+600
plastics_600[45]
```

```
## [1] 2204
```

```
mul_decom2<-decompose(plastics_600, type = "multiplicative")
autoplot(plastics_600, series = "Data")+
  autolayer(seasadj(mul_decom2), series = "Seasonal Adjusted")+
  ylab("Plastic sales")+
  ggtitle("Seasonally adjusted sales of plastic")+
  scale_color_manual(values = c("gray","blue"), breaks = c("Data", "Seasonal Adjusted"))
```



The outlier from adding 600 on the 45th point caused a spike on both original and seasonal adjusted plots. The seasonal adjusted follows the outliers trend due to the spike is not from the seasonal reason.

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

There are 60 observations in the Plastics data. I choose the 59th observation that near the end, and add 700 on top of it.

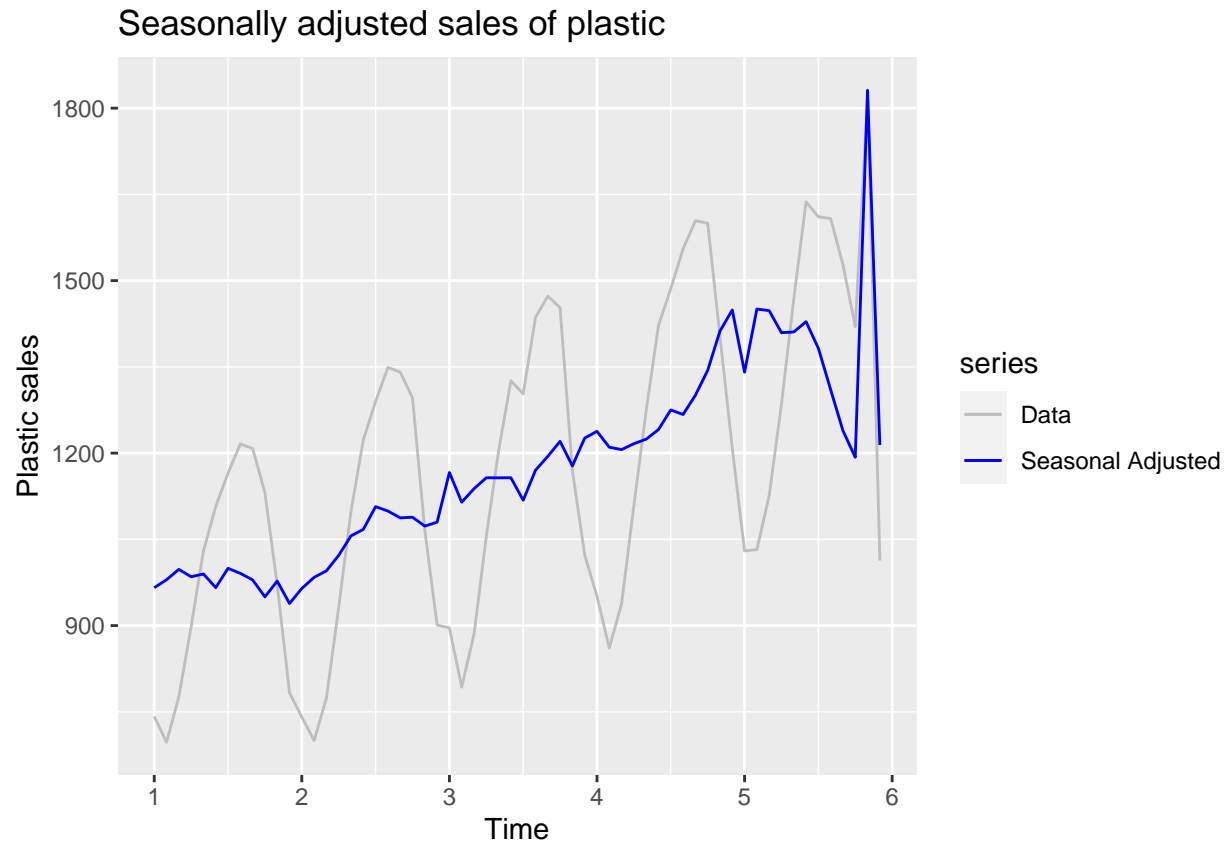
```
plastics[59]
```

```
## [1] 1119
```

```
plastics_700<-plastics
plastics_700[59]<-plastics[59]+700
plastics_700[59]
```

```
## [1] 1819
```

```
mul_decom3<-decompose(plastics_700, type = "multiplicative")
autoplot(plastics_700, series = "Data")+
  autolayer(seasadj(mul_decom3), series = "Seasonal Adjusted")+
  ylab("Plastic sales")+
  ggtitle("Seasonally adjusted sales of plastic")+
  scale_color_manual(values = c("gray","blue"), breaks = c("Data", "Seasonal Adjusted"))
```



Compare to adding large numbers on the one of the middle points, adding on the end leading the seasonal adjusted plot higher spike, very closed to the Data. The trend-cycle and remainder constructed the seasonal adjusted plot, and since the trend and remainder is generated from moving average, the end of trend and remainder doesn't have any information. Therefore, the seasonal adjusted plot follows the Data plot patterns for the end of part.