HW1: Forecasting: Principles and Practices

Group 4 - Subhalaxmi Rout, Kenan Sooklall, Devin Teran, Christian Thieme, Leo Yi

2021-06-18

**Good Job. See Comments and Notes in-line in the document.**

**Three files, kind of a pain - I requested one, please do that next time. Common theme, whether it be your boss, your professor or your customer – you want to make it easy to do business with you. I could tell quality differences across the three files – I have commented in each file. I recommend you getting a copy of group 2 or maybe Group 1 and looking for ideas to improve.**

**I am grading leniently this time, setting expectations - Report Grade is 95.**

# Homework 1: Time Series and Decomposition

## Forecasting: Principles and Practice

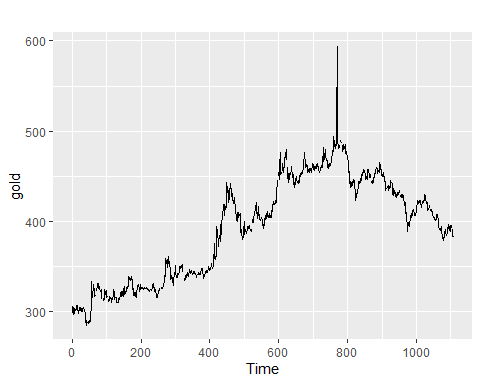
#### 2.1

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

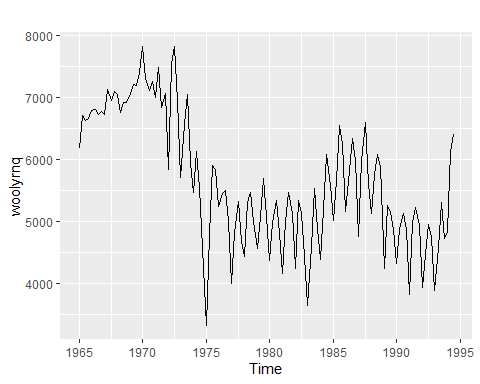
#?gold  
#?woolyrnq  
#?gas

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

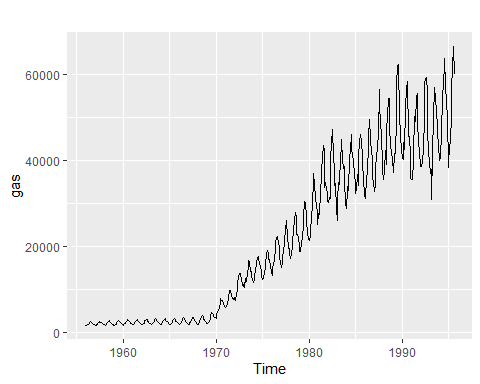
autoplot(gold)



autoplot(woolyrnq)



autoplot(gas)



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

Gold:

frequency(gold)

## [1] 1

* Gold frequency is yearly

Woolyrnq:

frequency(woolyrnq)

## [1] 4

* Woolyrng frequency is quarterly

Gas:

frequency(gas)

## [1] 12

* Gas frequency is monthly

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

which.max(gold)

## [1] 770

Observation 770 is the outlier. It’s value is:

gold[which.max(gold)]

## [1] 593.7

#### 2.3

**Download some monthly Australian retail data from** [**the book website**](https://otexts.com/fpp2/extrafiles/retail.xlsx)**. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file.**

**You can read the data into R with the following script:**

retaildata <- readxl::read\_excel("retail.xlsx", skip=1)

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

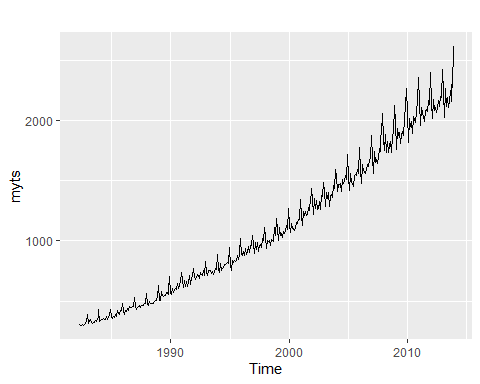
myts <- ts(retaildata[,"A3349335T"],  
 frequency=12, start=c(1982,4))

**Explore your chosen retail time series using the following functions:**

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

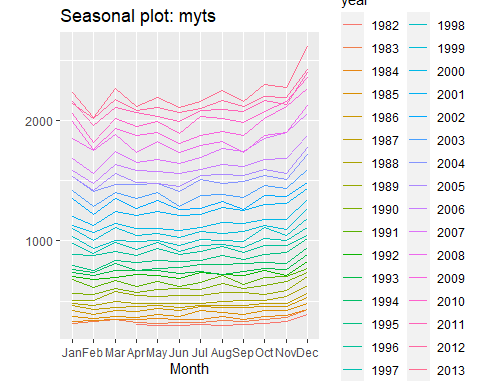
**Can you spot any seasonality, cyclicity and trend? What do you learn about the series?**

autoplot(myts)



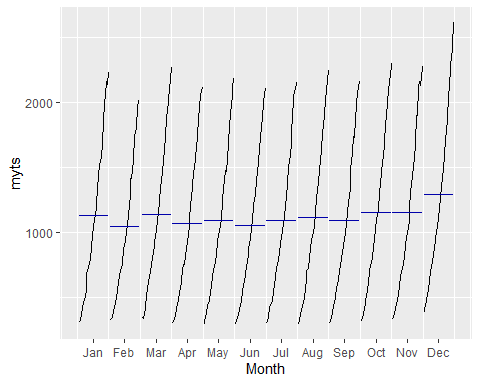
It appears that the data trends upward and that there is a seasonal component to the data as well. There does not appear to be cyclicity.

ggseasonplot(myts)



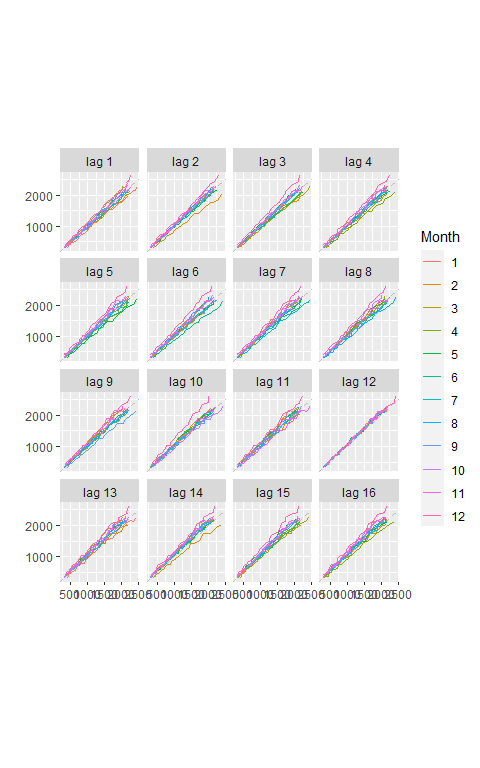
There is a seasonal nature to this data. We can see that sales tend to increase substantially in December and January and then decrease again in February. It also appears that generally, sales increases every other month and then decreases slightly the following month.

ggsubseriesplot(myts)



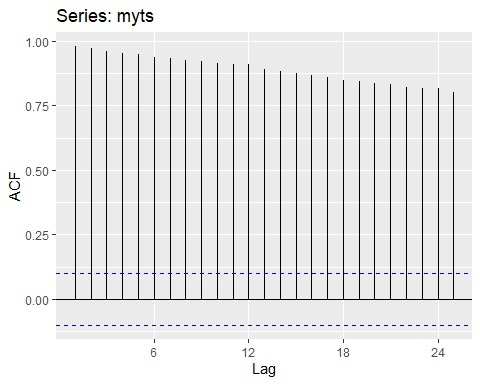
We can see above in the subseries plot the same pattern we mentioned before. It appears that December and January sales tend to be higher and then it appears that there is a general pattern where every other month is higher (except for Dec and Jan).

gglagplot(myts)



Looking at the lag plot we can see there is clear autocorrelation here. All of the lagged values appear to have a strong relationship, however, we see that a lag of 12 has the highest correlation.

ggAcf(myts)



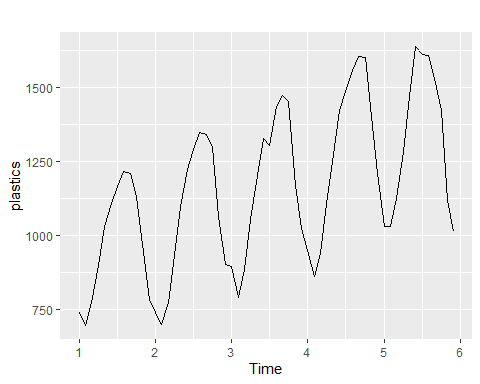
We can see above that all of the correlations are significantly different from zero as indicated by the blue lines. It appears that the strongest relationship occurs at the first lag, and slowly decreases over the next 25 lags.

#### 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?**

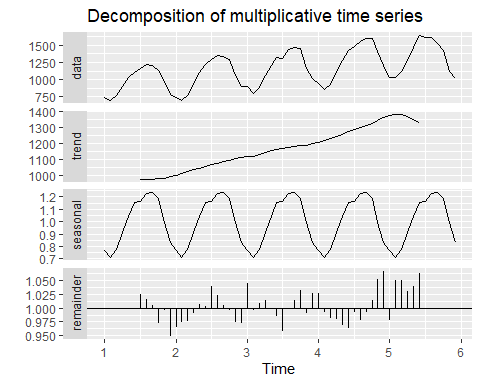
autoplot(plastics)



There appears to be an upward trend over time. In addition we note mid-year seasonality. There is no cyclicity.

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

plastics %>%  
 decompose(type = 'multiplicative') %>%  
autoplot()



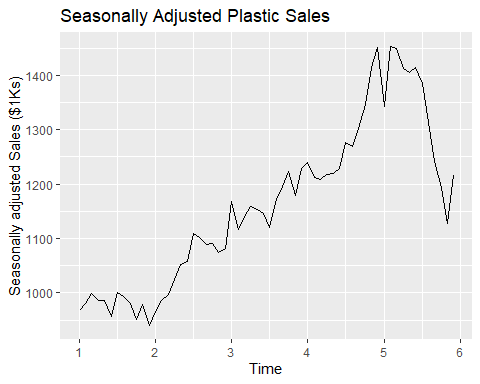
We can see that the plastics dataset has been decomposed into trend, seasonal, and error portions.

**Do the results support the graphical interpretation from part a?**

Yes, the results match our previous assertion. The only difference is we can see that the trend starts to decrease at the end of our dataset which we didn’t mention in part a.

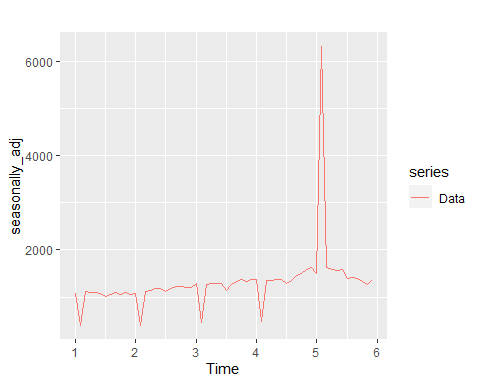
**Compute and plot the seasonally adjusted data.**

dcomp <- plastics %>%  
 decompose(type = 'multiplicative') %>%  
 seasonal()  
  
autoplot(plastics / dcomp) +   
 labs(title = "Seasonally Adjusted Plastic Sales", y = "Seasonally adjusted Sales ($1Ks)")



**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?**

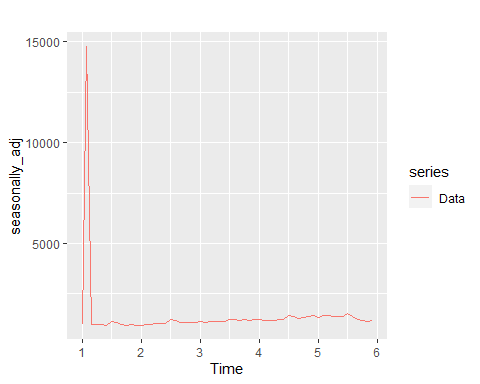
tmp <- plastics[50]  
plastics[50] <- plastics[50] + 9999  
seasonally\_adj <- plastics / (plastics %>% decompose(type='multiplicative'))$seasonal   
autoplot(seasonally\_adj, series="Data")



The outlier has had a tremendous effect. We can see that it has completely changed the plot and there are periods of deep declines that weren’t in the initial plot.

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

plastics[50] <- tmp  
plastics[2] <- plastics[2] + 9999  
seasonally\_adj <- plastics / (plastics %>% decompose(type='multiplicative'))$seasonal   
autoplot(seasonally\_adj, series="Data")



The outlier location makes a difference on where the spike is located. An outlier at the beginning almost completely destroys the seasonal fluctuations.