DATA624 Homework 1

Esteban Aramayo, Coffy Andrews-Guo, LeTicia Cancel, Joseph Connolly, Ian Costello

5/31/2022

Table of Contents

library(ggfortify)  
library(openxlsx)  
library(fpp2)  
library(fma)  
library(gridExtra)  
library(seasonal)

# Week 1

## HA 2.1

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

#help function for each series  
??gold

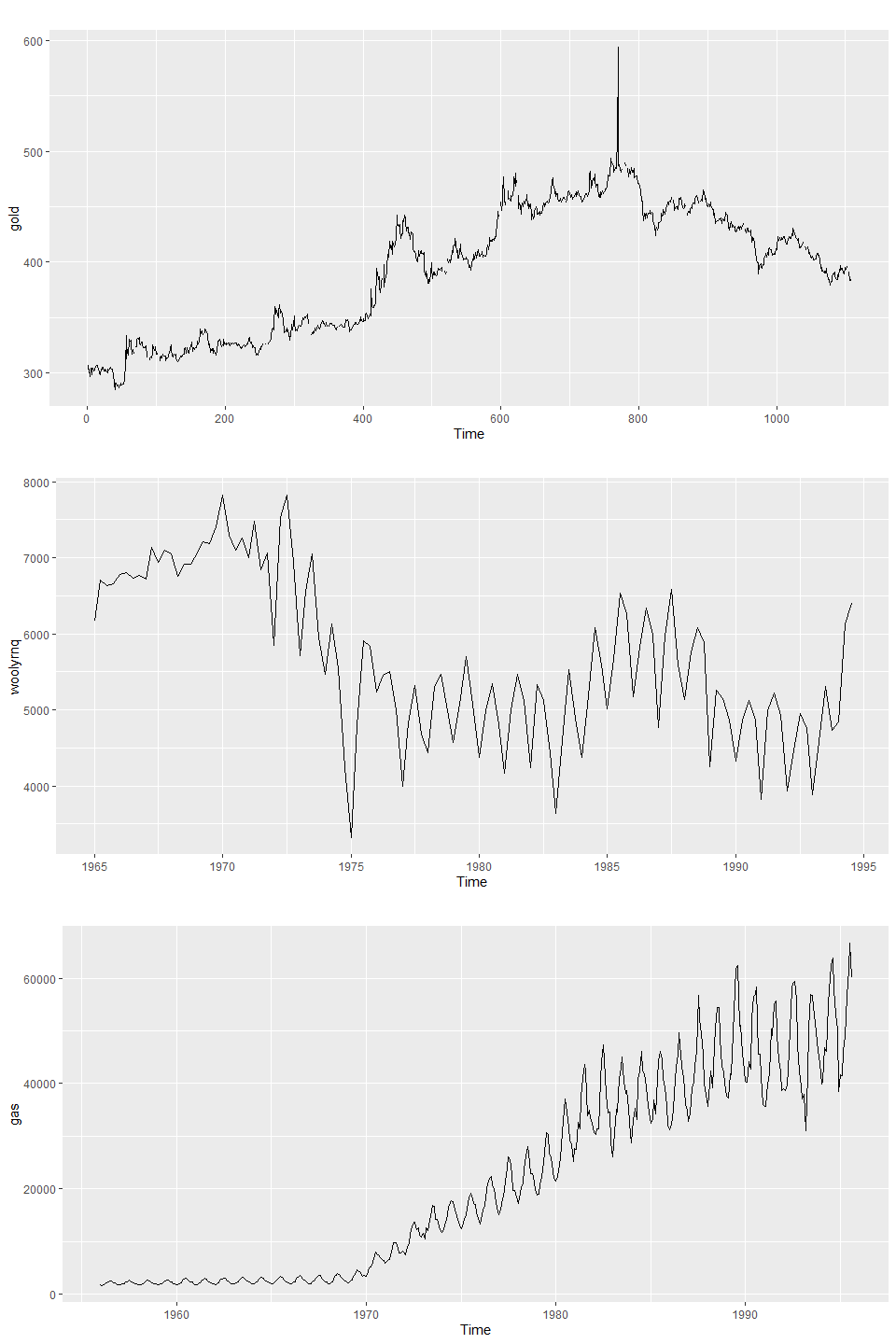
## starting httpd help server ... done

??woolyrnq

??gas

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

#autoplots of each series  
grid.arrange(autoplot(gold),autoplot(woolyrnq),autoplot(gas))



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

* gold has a frequency of 1, meaning it is annual.
* Woolyrng has a frequency of 1, meaning it is quarterly.
* gas has a frequency of 12, meaning it is monthly.

#frequency of each series  
frequency(gold)

## [1] 1

frequency(woolyrnq)

## [1] 4

frequency(gas)

## [1] 12

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

The outlier for the gold series is observation number 770, with the value of 593.7.

which.max(gold)

## [1] 770

gold[which.max(gold)]

## [1] 593.7

## HA 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 <- read.xlsx("https://otexts.com/fpp2/extrafiles/retail.xlsx",startRow = 2)

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

myts <- ts(retaildata[,"A3349873A"],  
 frequency=12, start=c(1982,4))  
myts

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1982 62.4 63.1 59.6 61.9 60.7 61.2 62.1 68.3 104.0  
## 1983 63.9 64.8 70.0 65.3 68.9 65.7 66.9 70.4 71.6 74.9 83.4 122.8  
## 1984 69.0 71.8 74.9 70.2 76.6 68.7 70.1 74.6 70.6 80.5 87.2 121.3  
## 1985 73.3 71.1 75.7 76.0 86.1 75.2 83.4 85.3 81.3 93.9 104.7 143.8  
## 1986 88.5 85.2 86.2 92.4 100.9 90.1 96.1 97.2 96.8 107.7 110.9 161.0  
## 1987 98.1 94.5 97.7 99.3 106.3 98.5 107.1 105.9 108.5 117.1 121.4 170.1  
## 1988 109.0 110.7 115.5 105.7 114.3 107.5 108.8 109.6 118.4 125.5 151.8 232.4  
## 1989 129.4 120.6 133.2 129.3 142.8 127.6 126.0 136.7 144.5 147.8 168.4 242.6  
## 1990 141.2 139.8 152.1 135.8 148.0 135.8 138.7 144.8 139.9 151.6 163.9 215.8  
## 1991 135.1 135.5 142.4 137.3 146.5 137.6 147.0 152.9 157.5 169.3 184.8 250.1  
## 1992 164.4 169.8 171.0 167.5 173.2 150.8 160.9 164.5 173.6 182.7 196.9 255.5  
## 1993 156.1 152.6 162.0 151.5 160.5 144.9 147.0 151.5 161.6 169.4 186.7 270.1  
## 1994 159.6 161.0 171.3 152.6 159.5 157.4 156.9 169.6 186.2 206.3 198.3 269.5  
## 1995 176.6 170.8 179.7 174.9 174.9 169.1 184.9 192.5 201.5 210.5 227.9 316.5  
## 1996 202.2 210.0 204.5 203.3 209.4 194.8 215.7 228.6 226.6 229.8 242.6 336.5  
## 1997 228.4 212.9 222.3 217.2 225.4 217.2 228.2 227.9 234.9 257.6 280.7 390.1  
## 1998 235.6 224.4 219.1 242.2 239.6 230.5 240.5 233.9 242.7 227.3 243.9 337.8  
## 1999 211.2 197.0 194.3 218.5 222.6 195.0 215.2 222.7 232.6 236.7 252.2 364.6  
## 2000 219.2 215.2 221.0 212.6 228.6 239.4 201.0 211.4 241.1 253.9 261.2 362.6  
## 2001 244.9 236.1 249.7 263.4 268.1 248.9 253.3 266.0 262.2 291.6 316.8 445.0  
## 2002 268.6 248.4 272.4 261.5 283.1 254.4 265.3 284.9 291.2 299.7 332.0 454.8  
## 2003 271.8 261.3 266.7 275.8 287.3 277.5 285.4 297.1 314.4 323.0 346.5 456.0  
## 2004 268.5 256.8 270.7 250.9 266.4 255.2 261.0 263.9 276.3 291.2 304.8 427.0  
## 2005 279.4 255.7 268.3 260.6 260.1 254.4 249.9 262.4 269.9 277.8 303.0 417.3  
## 2006 265.8 248.7 273.1 261.0 266.3 260.4 268.3 275.9 278.2 284.1 299.2 429.1  
## 2007 266.0 251.1 269.9 261.7 273.7 254.8 275.2 290.4 306.7 309.8 324.3 472.0  
## 2008 285.9 286.8 275.3 257.2 285.8 259.7 261.2 273.4 275.2 300.5 323.5 457.3  
## 2009 290.8 285.2 300.6 294.4 304.9 292.5 305.3 289.1 296.2 298.6 321.0 408.9  
## 2010 266.2 240.0 267.5 260.7 272.8 260.5 268.5 277.0 278.7 279.0 319.3 400.2  
## 2011 296.2 302.5 310.8 274.8 267.0 263.8 294.6 317.8 320.4 308.6 427.5 463.9  
## 2012 288.6 287.1 315.6 291.2 309.3 330.0 327.0 331.1 344.6 366.0 534.2 535.4  
## 2013 364.5 360.1 400.3 379.4 395.1 373.6 400.1 384.1 388.4 418.2 577.9 564.3

myts2 <- ts(retaildata[,"A3349791W"],  
 frequency=12, start=c(1982,4))  
myts2

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1982 22.2 23.1 22.8 23.2 21.4 21.8 21.0 23.5 31.4  
## 1983 20.7 22.1 24.9 24.5 25.4 24.6 23.9 25.0 24.5 24.6 27.8 36.8  
## 1984 21.4 22.8 24.5 23.1 27.1 24.3 25.6 25.5 23.3 26.1 27.8 35.4  
## 1985 23.4 21.8 24.5 25.3 29.7 26.2 30.4 29.0 27.7 30.7 32.8 41.3  
## 1986 28.4 26.3 27.1 29.8 34.4 29.9 35.2 33.6 33.5 34.0 34.4 46.4  
## 1987 30.8 28.3 30.8 32.2 35.5 34.8 35.7 34.4 34.6 35.9 37.0 51.3  
## 1988 32.9 34.3 37.0 34.8 38.6 37.0 38.1 37.3 39.3 39.5 46.7 68.1  
## 1989 37.8 34.3 41.4 38.3 44.8 44.1 42.2 46.2 48.7 49.6 54.4 75.3  
## 1990 47.3 46.0 49.6 45.6 49.9 45.2 46.2 47.4 46.2 51.0 54.2 72.5  
## 1991 45.4 43.5 46.7 46.4 48.3 43.2 47.7 49.3 52.1 55.8 59.5 82.4  
## 1992 53.1 53.5 52.9 52.4 52.8 48.1 51.1 49.8 55.2 57.9 61.4 80.3  
## 1993 48.4 45.1 48.9 45.7 48.1 46.2 49.5 50.3 54.2 58.9 63.9 95.4  
## 1994 52.7 51.4 54.1 48.1 50.9 52.5 55.8 57.7 51.9 52.9 59.4 94.4  
## 1995 49.2 43.0 48.1 48.2 51.2 50.0 54.9 53.3 54.9 55.3 67.5 105.9  
## 1996 53.6 54.7 56.6 53.9 57.4 60.3 64.5 62.2 61.4 59.2 65.5 104.9  
## 1997 53.8 51.5 53.1 50.8 55.2 60.4 63.5 57.5 60.1 59.3 64.8 102.0  
## 1998 58.9 49.5 53.3 56.0 52.4 54.2 60.6 54.7 56.3 61.7 67.2 107.4  
## 1999 60.4 49.4 55.6 56.2 53.2 56.8 59.6 58.9 64.4 60.4 66.9 102.4  
## 2000 63.1 56.2 60.6 59.4 62.0 69.2 57.3 61.8 68.1 64.2 71.1 110.4  
## 2001 69.7 61.6 67.2 63.7 59.7 60.4 64.2 61.8 62.9 60.8 72.8 110.5  
## 2002 64.3 54.2 62.2 60.2 64.4 63.4 68.6 71.1 67.2 71.4 85.3 126.2  
## 2003 71.1 59.2 62.8 63.3 64.6 69.4 73.1 71.4 72.1 79.0 88.7 130.8  
## 2004 78.0 70.1 76.3 67.0 67.6 72.5 78.6 72.3 73.8 79.1 83.9 125.1  
## 2005 77.7 64.4 71.0 70.7 70.0 74.8 73.4 72.3 73.9 73.7 80.4 126.4  
## 2006 79.3 62.8 71.3 72.8 72.8 74.8 72.5 73.9 76.4 87.6 94.4 152.3  
## 2007 91.9 79.9 91.3 80.6 82.3 88.5 87.7 91.6 95.3 102.1 110.1 177.7  
## 2008 104.0 87.9 93.3 92.3 91.6 89.1 83.4 85.1 84.3 99.6 108.0 186.4  
## 2009 101.7 83.2 95.5 101.5 99.3 92.7 99.1 95.0 100.8 106.8 111.1 201.0  
## 2010 97.7 74.6 88.2 88.7 83.5 83.4 90.8 85.1 91.4 104.4 112.9 200.6  
## 2011 107.2 78.2 87.8 97.6 83.0 87.9 96.9 92.0 105.8 107.4 131.3 209.3  
## 2012 108.4 89.3 92.5 85.8 83.8 80.2 81.3 87.2 100.9 108.1 123.1 182.2  
## 2013 116.0 90.6 96.3 112.9 100.8 93.1 101.9 109.1 107.6 122.7 154.8 237.1

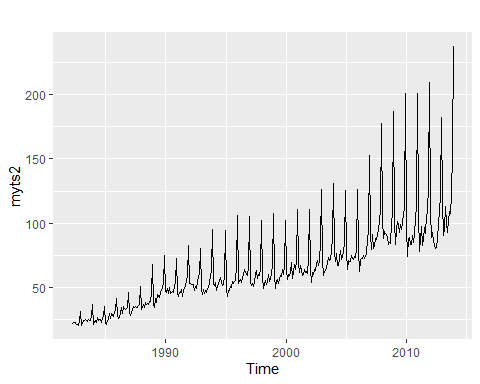
#### Explore your chosen retail time series using the following functions:

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

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

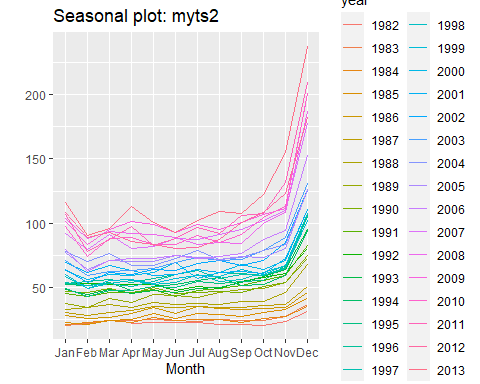
The first plot using autoplot() shows some seasonality for each year telling us that the price spikes at different points. There is also an upward trend showing that the price of gold continues to increase overtime. The length between the min and max for each of the seasonal spike grows longer each year telling us that the price difference between the minimum and maximum price spreads out more over time.

#time series qutoplot  
autoplot(myts2)



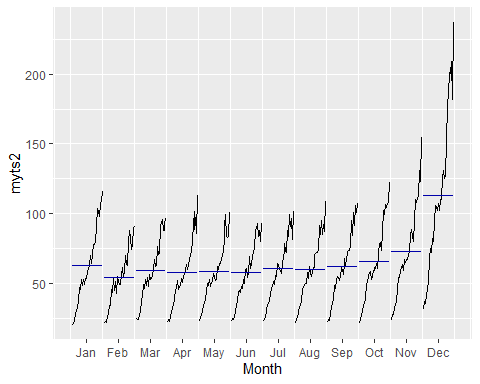
The seasonal plot below using ggseasonlot() gives us more clarity on when the seasonality takes place. Each year has a spike in December with the max prices increasing each year. We can take a guess as to why December is the month that spikes since this is when holiday shopping takes place.

#time series seasonal plot  
ggseasonplot(myts2)



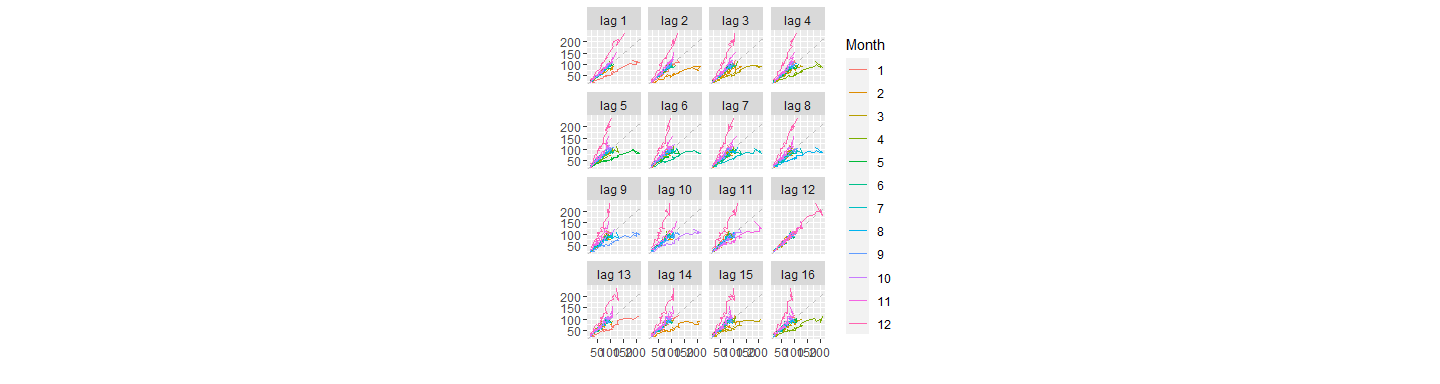
The subseries plot confirms the seasonality shown in the previous plots with December being the seasonal month. This plot creates a better visualization for a couple of details better than the previous plots. The difference between the minimum and maximum over the years is clearer. The minimum has a minor increase in december but the maximum has a dramatic increase when compared to the other months. The mean also has a big increase in December but then levels out during the spring and summer months.

#time series sub series plot  
ggsubseriesplot(myts2)



The lag plot shows strong positive correlation for all months in lag 12.

#time series lag plot  
gglagplot(myts2)



The ACF plot supports the lag plot by showing a high autocorrelation coefficient for lag 12. There are also no negative correlations which was not as clear in the subseries plot. They are also all significantly different from zero.

#time series ACF  
ggAcf(myts2)

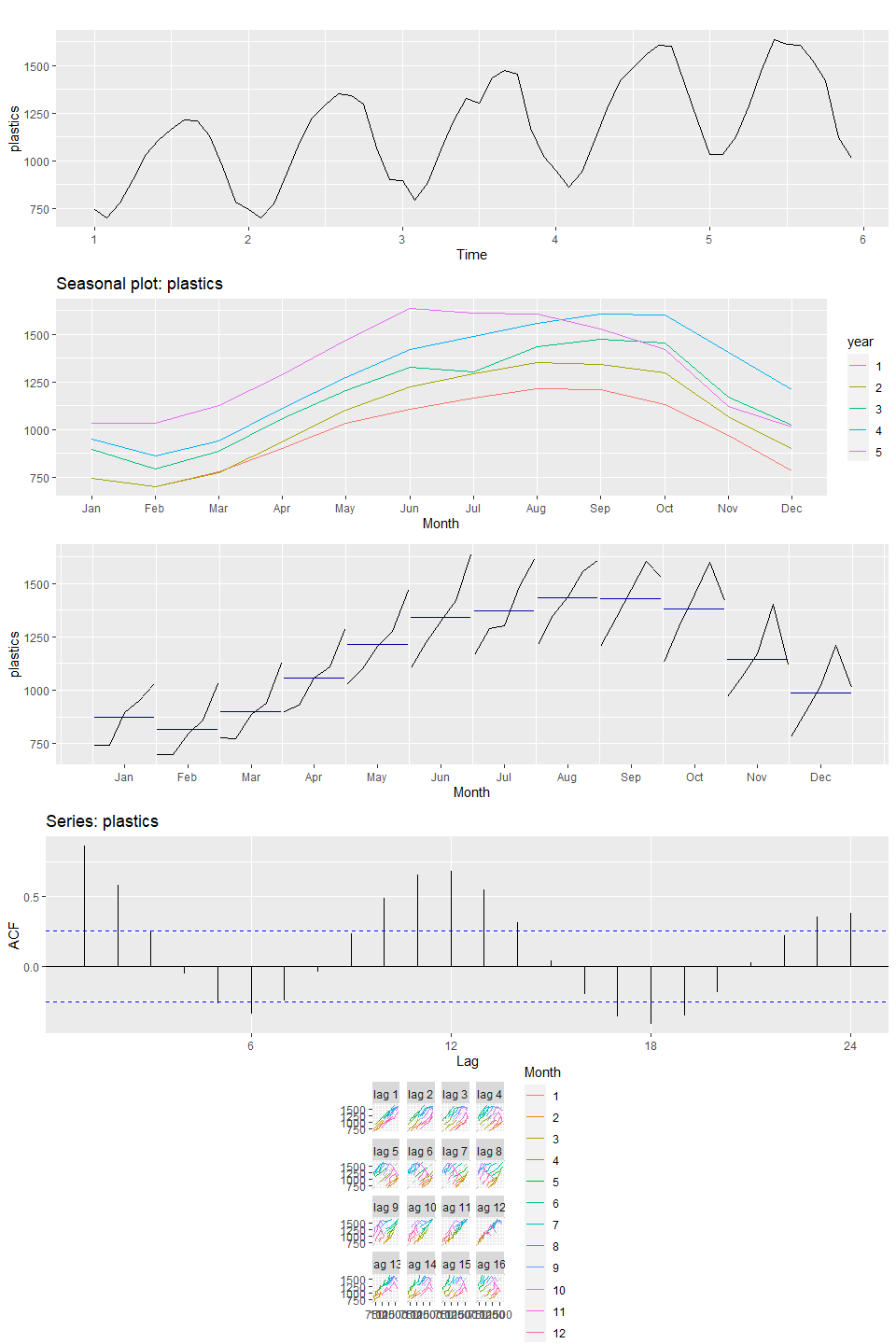


## HA 6.2

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

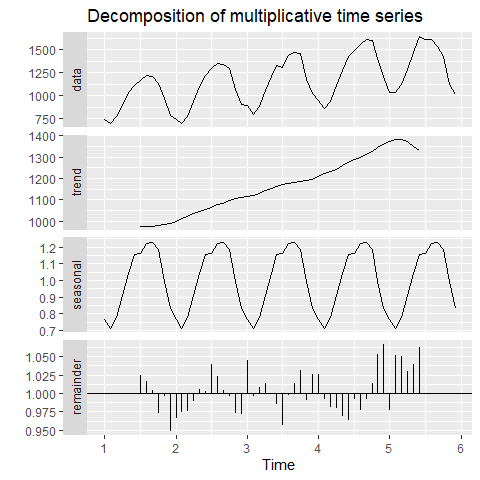
### a. Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle?

#autoplot(), ggseasonplot(), ggsubseriesplot(), gglagplot(), ggAcf()  
p1 <- autoplot(plastics)  
p2 <- ggseasonplot(plastics)  
p3 <- ggsubseriesplot(plastics)  
p4 <- gglagplot(plastics)  
p5 <- ggAcf(plastics)  
grid.arrange(p1,p2,p3,p5,p4, ncol=1)



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

#multiplicative decomposition  
decomp\_plastics <- plastics %>%  
 decompose(type = "multiplicative")  
  
decomp\_plastics %>% autoplot()



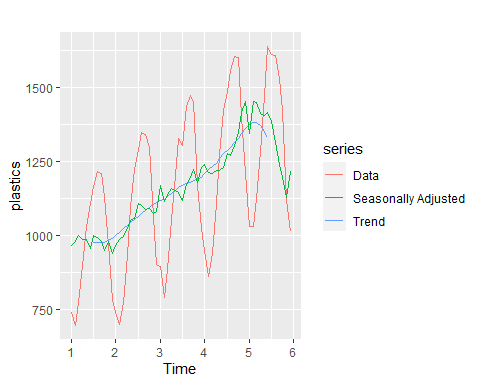
### c. Do the results support the graphical interpretation from part a?

The “Data” and “Seasonal” sections of the Decomposition shows similar results to the autoplot() chart. There is a steady seasonal trend that has a similar duration for each seasonal cycle. The “Trend” section of the Decomposition also supports the visuals from part A and shows a steady upward trend from years 1-5.

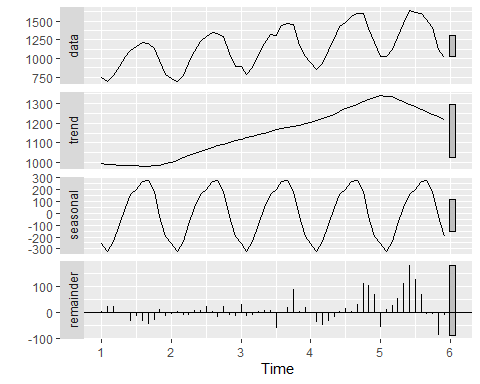
### d. Compute and plot the seasonally adjusted data.

autoplot(plastics, series = "Data") +   
 autolayer(trendcycle(decomp\_plastics), series = "Trend") +  
 autolayer(seasadj(decomp\_plastics), series = "Seasonally Adjusted")

## Warning: Removed 12 row(s) containing missing values (geom\_path).



#fit <- seas(x = plastics, x11="")  
  
plastics %>%  
 stl(s.window = "periodic", robust = TRUE) %>%  
 autoplot()

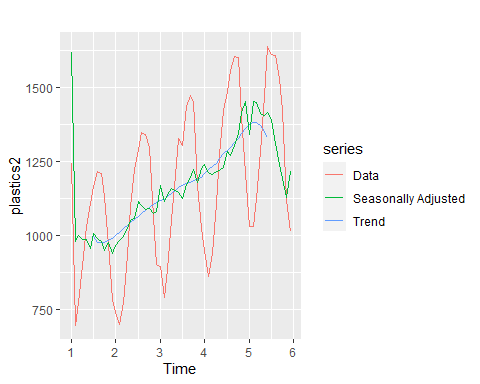


### e. Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?

The outlier was applied to the first observation in the plastics data so the chart changed only in year 1. The outlier being in the beginning of the data had no impact on the rest of the plot.

#creating a copy of plastics and adding 500 to one observation  
plastics2 <- plastics  
plastics2[1] <- plastics2[1] + 500  
  
decomp\_plastics2 <- plastics2 %>%  
 decompose(type = "multiplicative")  
  
autoplot(plastics2, series = "Data") +   
 autolayer(trendcycle(decomp\_plastics2), series = "Trend") +  
 autolayer(seasadj(decomp\_plastics2), series = "Seasonally Adjusted")

## Warning: Removed 12 row(s) containing missing values (geom\_path).

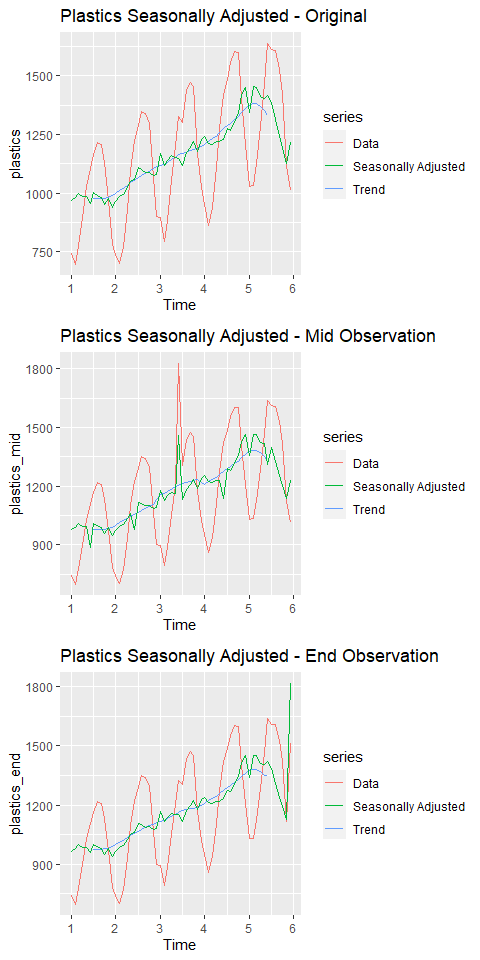


### f. Does it make any difference if the outlier is near the end rather than in the middle of the time series?

The middle outlier has a greater impact on the Seasonally adjusted line than the end outlier when compared to the original plot. When the last observation is the outlier the Data and Seasonally Adjusted lines trend upward and all other areas on the chart are not impacted. When a middle observation is the outlier we see a different seasonally adusted line when compared to the original chart. The downward and upward spikes around the outlier are sharper on the middle outlier chart while the original chart has subtle changes with a fairly even upward trend.

plastics\_mid <- plastics  
plastics\_end <- plastics  
plastics\_mid[30] <- plastics\_mid[30] + 500  
plastics\_end[60] <- plastics\_end[60] + 500  
  
decomp\_mid <- plastics\_mid %>%  
 decompose(type = "multiplicative")  
  
p <- autoplot(plastics, series = "Data") +   
 autolayer(trendcycle(decomp\_plastics), series = "Trend") +  
 autolayer(seasadj(decomp\_plastics), series = "Seasonally Adjusted") +  
 ggtitle("Plastics Seasonally Adjusted - Original")  
  
p1 <- autoplot(plastics\_mid, series = "Data") +   
 autolayer(trendcycle(decomp\_mid), series = "Trend") +  
 autolayer(seasadj(decomp\_mid), series = "Seasonally Adjusted") +  
 ggtitle("Plastics Seasonally Adjusted - Mid Observation")  
  
decomp\_end <- plastics\_end %>%  
 decompose(type = "multiplicative")  
  
p2 <- autoplot(plastics\_end, series = "Data") +   
 autolayer(trendcycle(decomp\_end), series = "Trend") +  
 autolayer(seasadj(decomp\_end), series = "Seasonally Adjusted") +  
 ggtitle("Plastics Seasonally Adjusted - End Observation")  
  
grid.arrange(p, p1,p2)

## Warning: Removed 12 row(s) containing missing values (geom\_path).  
## Removed 12 row(s) containing missing values (geom\_path).  
## Removed 12 row(s) containing missing values (geom\_path).



# Week 2

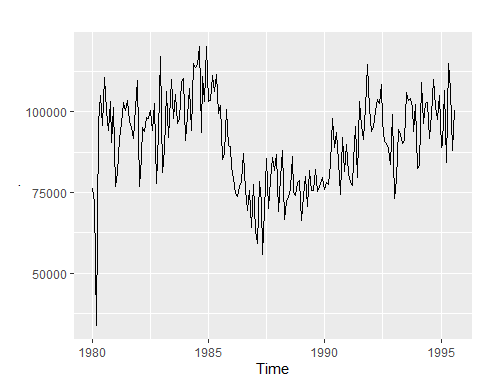
## KJ 3.1

## KJ 635

## HA 7.1

### Consider the pigs series — the number of pigs slaughtered in Victoria each month.

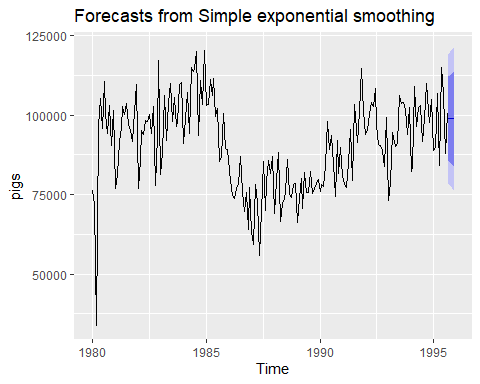
help(pigs)  
pigs %>% autoplot()



### a. Use the ses() function in R to find the optimal values of and and generate forecasts for the next four months.

Using the summary() function and .

fc<- ses(pigs, h=4)  
  
fc %>%  
 autoplot()



summary(fc)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = pigs, h = 4)   
##   
## Smoothing parameters:  
## alpha = 0.2971   
##   
## Initial states:  
## l = 77260.0561   
##   
## sigma: 10308.58  
##   
## AIC AICc BIC   
## 4462.955 4463.086 4472.665   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 385.8721 10253.6 7961.383 -0.922652 9.274016 0.7966249 0.01282239  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 1995 98816.41 85605.43 112027.4 78611.97 119020.8  
## Oct 1995 98816.41 85034.52 112598.3 77738.83 119894.0  
## Nov 1995 98816.41 84486.34 113146.5 76900.46 120732.4  
## Dec 1995 98816.41 83958.37 113674.4 76092.99 121539.8

#### b. Compute a 95% prediction interval for the first forecast using where is the standard deviation of the residuals. Compare your interval with the interval produced by R.

From chapter 3.5. Prediction intervals are calculated as . The multiplier for 95% interval is 1.96 and the residuals are also equal to the RMSE. Using the SES model formula from part a, the $ values are stored in a vector as are the residuals. Then, the values are subbed in for the formula.

The computed intervals vs. the predicted intervals are quite close. The September 1995 low interval at 95% is 78611.97, compared to the computed 78679.97 only a difference of 68. The high values for the predicted interval at 95% is 119020.80, compared to 118952.84 again differing by 68.

y\_hat <- c(1.96, -1.96)  
s <- sd(residuals(fc))  
  
ses(pigs, h=4)$mean[1]+(y\_hat\*s)

## [1] 118952.84 78679.97

## HA 7.2

## HA 7.3