Упражнения 6.9

library(fpp2)

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

## ── Attaching packages ───────────────────────────────────────── fpp2 2.4 ──

## ✓ ggplot2 3.3.3 ✓ fma 2.4   
## ✓ forecast 8.13 ✓ expsmooth 2.3

##

library(seasonal)

**1. Show that a 3×5 MA is equivalent to a 7-term weighted moving average with weights of 0.067, 0.133, 0.200, 0.200, 0.200, 0.133, and 0.067.**

In case of a 3x5 moving average, this signifies a 3 moving average of a 5 moving average. Weights = c(0.067, 0.133, 0.200, 0.200, 0.200, 0.133, 0.067)

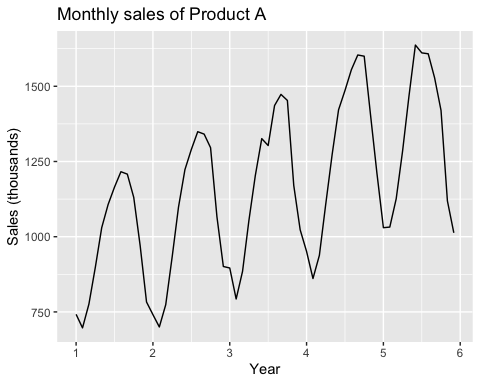
3x5 MA = [((Y1 + Y2 + Y3 + Y4 + Y5)/5) + ((Y2 + Y3 + Y4 + Y5 + Y6)/5) + ((Y3 + Y4 + Y5 + Y6 + Y7)/5)] / 3 = = 1/15 (Y1 + Y2 + Y3 + Y4 + Y5) + 1/15 (Y2 + Y3 + Y4 + Y5 + Y6) + 1/15 (Y3 + Y4 + Y5 + Y6 + Y7) = = Y1/15 + 2/15 Y2 + Y3/5 + Y4/5 + Y5/5 + 2/15 Y6 + Y7/15 = 0.067 Y1 + 0.133 Y2 + 0.200 Y3 + 0.200 Y4 + 0.200 Y5 + 0.133 Y6 + 0.067 Y7

Done.

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

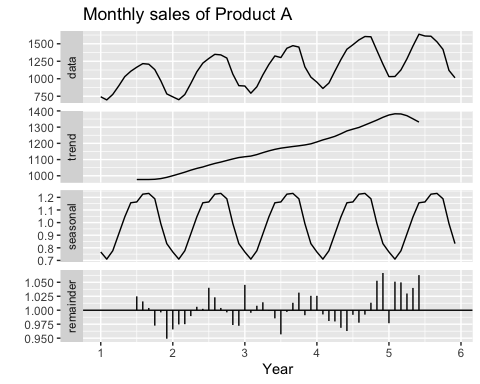
*a. Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle?*

autoplot(plastics, main="Monthly sales of Product A", xlab="Year", ylab="Sales (thousands)")

 The plot shows seasonal fluctuations of sales at the beginning and ending months of each year. Sales are at their lowest at the beginning of each year and the highest past the middle of each year. There is also a positive trend.

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

plastics %>% decompose(type="multiplicative") %>%  
 autoplot() + xlab("Year") +  
 ggtitle("Monthly sales of Product A")

 *c. Do the results support the graphical interpretation from part a?* These results do support the above statements regarding seasonal trends that exist in the data. It also shows there is an overall positive trend in sales.

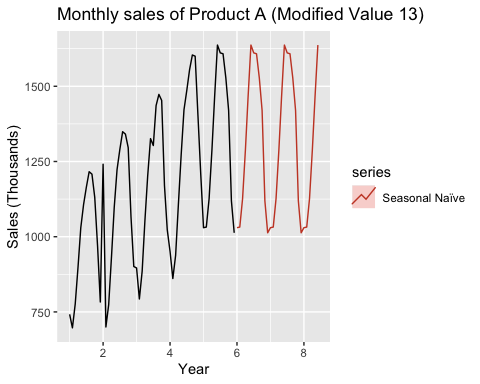
*d. Compute and plot the seasonally adjusted data.*

autoplot(plastics, main="Monthly sales of Product A", ylab="Sales (Thousands)", xlab="Year") + autolayer(snaive(plastics, h=30), series="Seasonal Naïve", PI=FALSE)



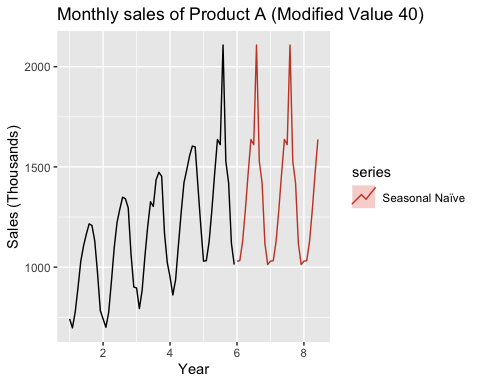
*e. Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?*

plasticsAdd <- plastics  
plasticsAdd[13] <- plasticsAdd[13] + 500  
  
autoplot(plasticsAdd, main="Monthly sales of Product A (Modified Value 13)", ylab="Sales (Thousands)", xlab="Year") + autolayer(snaive(plasticsAdd, h=30), series="Seasonal Naïve", PI=FALSE)

 The forecasted values stay relatively stable with the addition of 500 to value 13. This shows the strength of the seasonally adjusted data prediction.

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

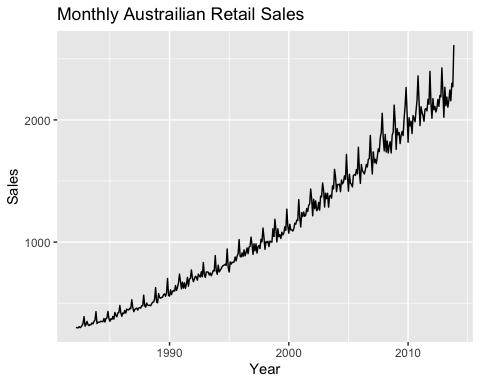
# Add 500 to end Value  
plasticsAddEnd <- plastics  
plasticsAddEnd[56] <- plasticsAddEnd[56] + 500  
  
autoplot(plasticsAddEnd, main="Monthly sales of Product A (Modified Value 40)", ylab="Sales (Thousands)", xlab="Year") + autolayer(snaive(plasticsAddEnd, h=30), series="Seasonal Naïve", PI=FALSE)



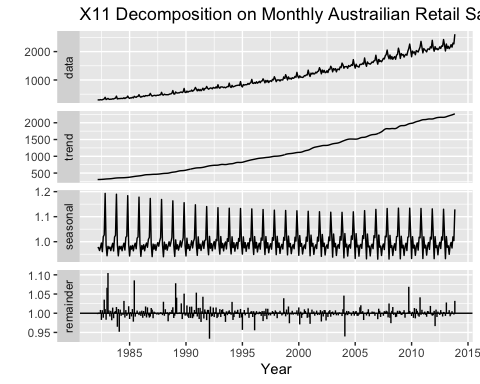
Adding the outlier to an end value does shift the forecast up and likely created larger values than intended. For this reason, outliers should be evaluated when using Multiplicative decomposition.

**3. Recall your retail time series data (from Exercise 3 in Section 2.10). Decompose the series using X11. Does it reveal any outliers, or unusual features that you had not noticed previously?**

library(readxl)  
retail <- read\_excel("retail.xlsx", skip=1)  
  
retailTS <- ts(retail[,2], frequency=12, start=c(1982, 3))  
  
autoplot(retailTS, main="Monthly Austrailian Retail Sales", ylab="Sales", xlab="Year")



retailx11 <- seas(retailTS, x11="")  
  
autoplot(retailx11, main="X11 Decomposition on Monthly Austrailian Retail Sales", xlab="Year")

 Outliers are revealed in the remainder plot. We can see some major outliers around 1982, 1986, 1989, 1993, 2004, and 2009.

**4. Figures 6.16 and 6.17 show the result of decomposing the number of persons in the civilian labour force in Australia each month from February 1978 to August 1995.**

*a. Write about 3–5 sentences describing the results of the decomposition. Pay particular attention to the scales of the graphs in making your interpretation.*

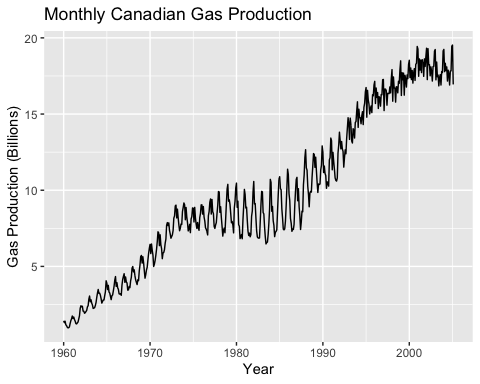
Decomposition shows the results from the number of persons in the civilian labour force in Austrialia each month from February 1978 to August 1995. There is a strong positive trend in the number of workers during the time period. Seasonality chart shows that the growth is cyclical and has strong seasonal trends. There are some major outliers that exist in the data around 1991-1992 and 1994. The second chart provides insight into seasonal components of each month. A large increase in July and a large decrease in March and August can be observed.

*b. Is the recession of 1991/1992 visible in the estimated components?* The major outliers in the data observed in the previous question is bound to be explained by the recession of 1991/1992, provided by the remainder plot of the decomposition.

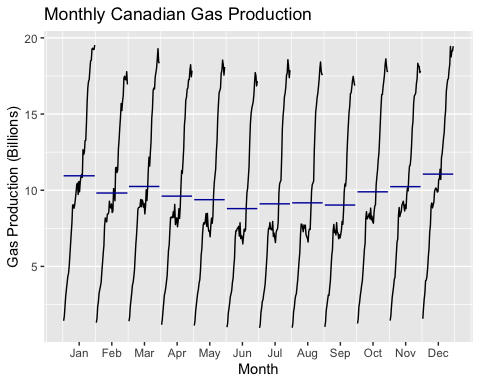
**5. This exercise uses the cangas data (monthly Canadian gas production in billions of cubic metres, January 1960 – February 2005).**

*a. Plot the data using autoplot(), ggsubseriesplot() and ggseasonplot() to look at the effect of the changing seasonality over time. What do you think is causing it to change so much?*

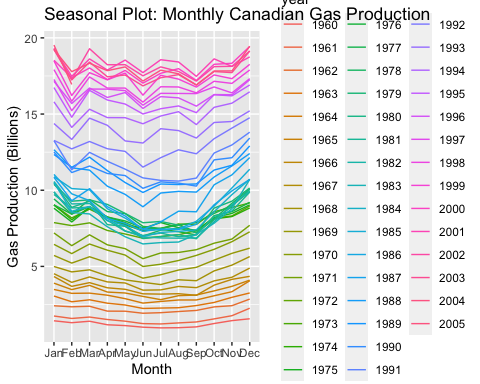
autoplot(cangas, main="Monthly Canadian Gas Production", ylab="Gas Production (Billions)", xlab="Year")



ggsubseriesplot(cangas, main="Monthly Canadian Gas Production", ylab="Gas Production (Billions)")

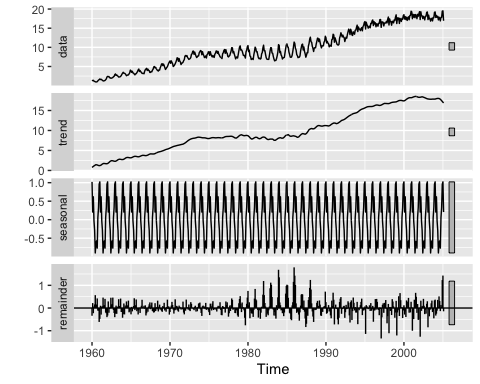


ggseasonplot(cangas, main="Seasonal Plot: Monthly Canadian Gas Production", ylab="Gas Production (Billions)")

 The seasonaility may be due to fuel prices and demand based on season. It could be cheaper to produce gas in winter, or the demand for the gas could be higher in winter.

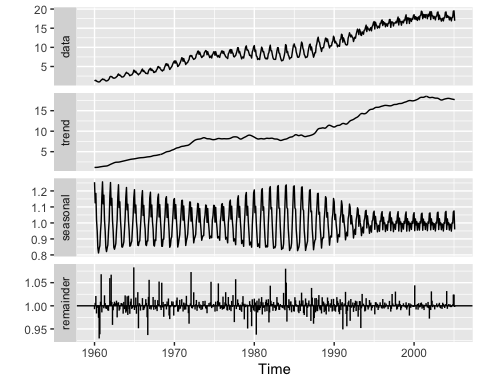
*b. Do an STL decomposition of the data. You will need to choose s.window to allow for the changing shape of the seasonal component.*

cangas %>%  
stl(t.window=13, s.window="periodic", robust=TRUE) %>%  
autoplot()

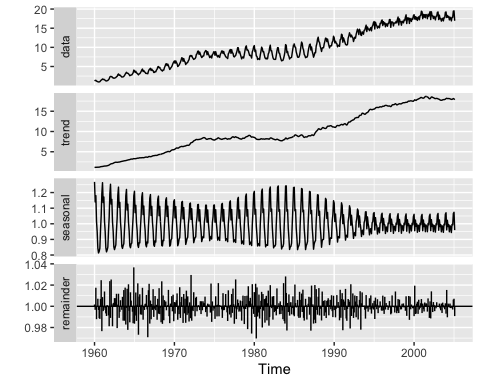


*c. Compare the results with those obtained using SEATS and X11. How are they different?*

cangas %>% seas(x11="") %>%  
autoplot()



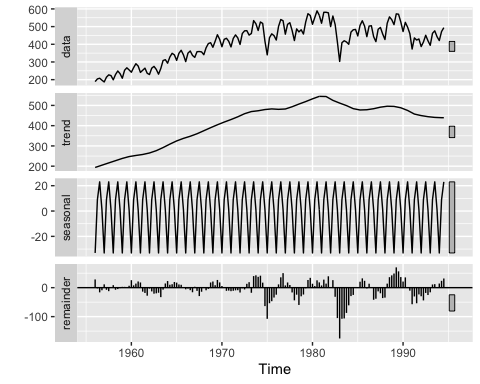
cangas %>% seas() %>%  
autoplot()

 The results appear fairly similar but some differences can be seen. The seasonality graph shows a consistent seasonality throughout the time series. The remainder plot provides some insight into the outliers, however, X11 gives less remainders meaning using a better seasoning and trending.

**6. We will use the bricksq data (Australian quarterly clay brick production. 1956–1994) for this exercise.**

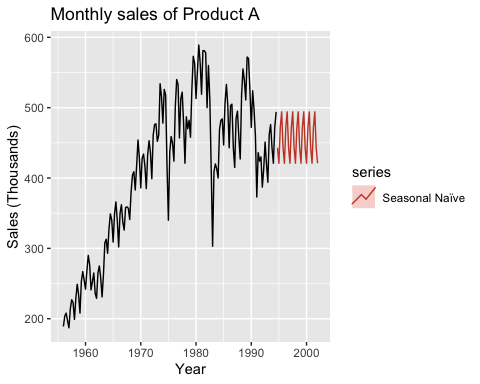
*a. Use an STL decomposition to calculate the trend-cycle and seasonal indices. (Experiment with having fixed or changing seasonality.)*

bricksq %>%  
stl(t.window=26, s.window="periodic", robust=TRUE) %>%  
autoplot()



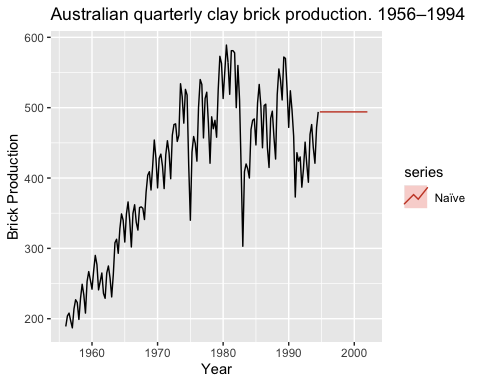
*b. Compute and plot the seasonally adjusted data.*

autoplot(bricksq, main="Monthly sales of Product A", ylab="Sales (Thousands)", xlab="Year") + autolayer(snaive(bricksq, h=30), series="Seasonal Naïve", PI=FALSE)



*c. Use a naïve method to produce forecasts of the seasonally adjusted data.*

autoplot(bricksq, main="Australian quarterly clay brick production. 1956–1994", ylab="Brick Production", xlab="Year")+ autolayer(naive(bricksq, h=30), series="Naïve", PI=FALSE)

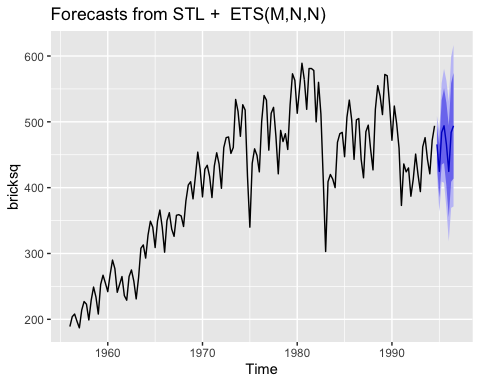


*d. Use stlf() to reseasonalise the results, giving forecasts for the original data.*

brickF1 <- stlf(bricksq)  
brickF1

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1994 Q4 465.7326 437.4903 493.9748 422.5398 508.9254  
## 1995 Q1 424.6727 384.7119 464.6335 363.5580 485.7874  
## 1995 Q2 483.9914 435.0232 532.9595 409.1010 558.8817  
## 1995 Q3 493.9988 437.4242 550.5734 407.4754 580.5222  
## 1995 Q4 465.7326 402.4454 529.0198 368.9431 562.5220  
## 1996 Q1 424.6727 355.3067 494.0387 318.5865 530.7589  
## 1996 Q2 483.9914 409.0259 558.9568 369.3416 598.6411  
## 1996 Q3 493.9988 413.8128 574.1848 371.3650 616.6326

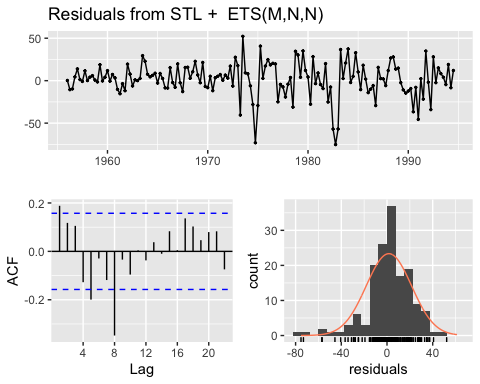
autoplot(brickF1)



*e. Do the residuals look uncorrelated?*

checkresiduals((brickF1))

## Warning in checkresiduals((brickF1)): The fitted degrees of freedom is  
## based on the model used for the seasonally adjusted data.

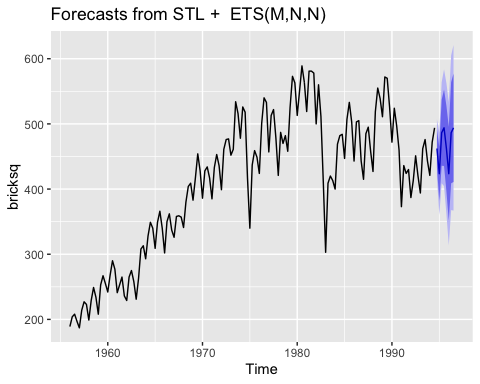


##   
## Ljung-Box test  
##   
## data: Residuals from STL + ETS(M,N,N)  
## Q\* = 41.128, df = 6, p-value = 2.733e-07  
##   
## Model df: 2. Total lags used: 8

The residuals appear to be approximately normal and show correlation.

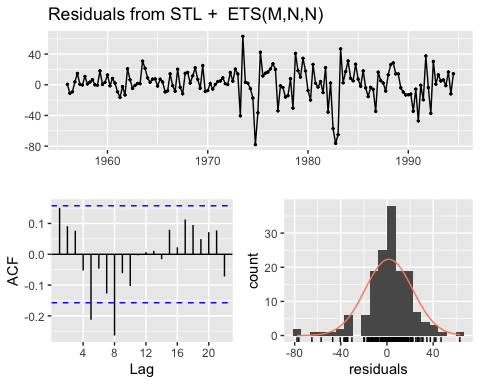
*f. Repeat with a robust STL decomposition. Does it make much difference?*

brickF1R <- stlf(bricksq, robust=TRUE)  
autoplot(brickF1R)



checkresiduals(brickF1R)

## Warning in checkresiduals(brickF1R): The fitted degrees of freedom is based  
## on the model used for the seasonally adjusted data.



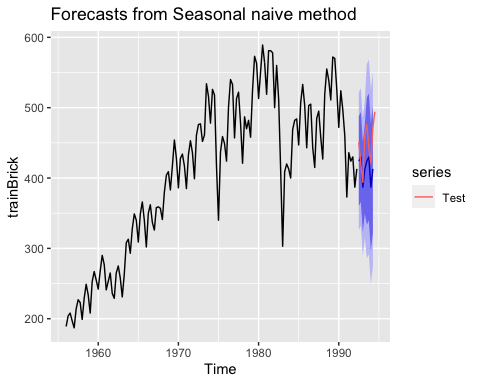
##   
## Ljung-Box test  
##   
## data: Residuals from STL + ETS(M,N,N)  
## Q\* = 28.163, df = 6, p-value = 8.755e-05  
##   
## Model df: 2. Total lags used: 8

It appears to be have reduced the normality of the residuals some but still fairly normal and has autocorrelation issues.

*g. Compare forecasts from stlf() with those from snaive(), using a test set comprising the last 2 years of data. Which is better?*

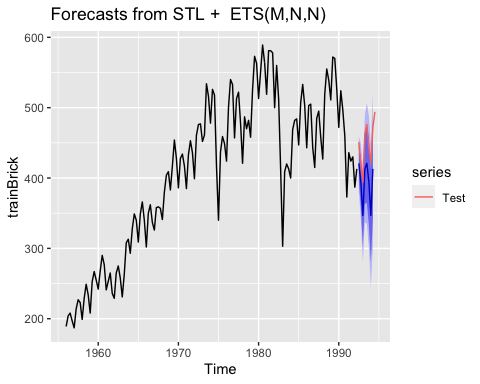
trainBrick <- subset(bricksq, end=length(bricksq) - 9)  
testBrick <- subset(bricksq, start=length(bricksq) - 8)  
  
sBrick <- snaive(trainBrick)  
stlfBrick <- stlf(trainBrick, robust=TRUE)  
  
autoplot(sBrick) + autolayer(testBrick, series="Test", PI=FALSE)

## Warning: Ignoring unknown parameters: PI



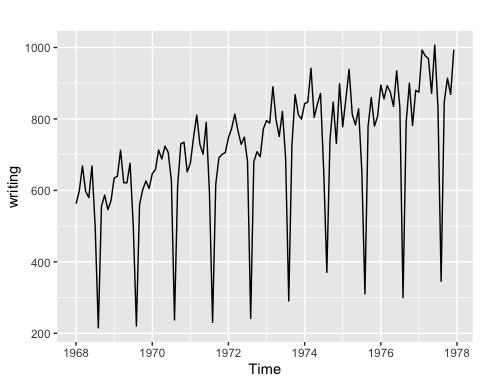
autoplot(stlfBrick)+ autolayer(testBrick, series="Test", PI=FALSE)

## Warning: Ignoring unknown parameters: PI

 The stlf() function appears to create less variable forecast and serves as a better predictor of brick production.

**7. Use stlf() to produce forecasts of the writing series with either method=“naive” or method=“rwdrift”, whichever is most appropriate. Use the lambda argument if you think a Box-Cox transformation is required.**

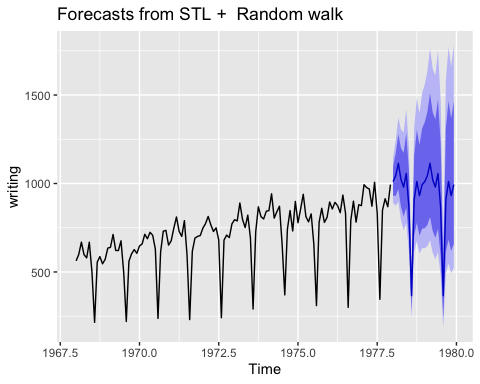
autoplot(writing)



stlf(writing, method='naive')

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 1978 990.5776 930.7390 1050.4162 899.06240 1082.0929  
## Feb 1978 1021.2781 936.6535 1105.9026 891.85600 1150.7002  
## Mar 1978 1075.1129 971.4694 1178.7564 916.60387 1233.6219  
## Apr 1978 1002.2253 882.5481 1121.9025 819.19485 1185.2558  
## May 1978 967.7282 833.9251 1101.5314 763.09393 1172.3625  
## Jun 1978 1032.4996 885.9256 1179.0736 808.33399 1256.6652  
## Jul 1978 881.2474 722.9294 1039.5655 639.12088 1123.3740  
## Aug 1978 469.7581 300.5090 639.0072 210.91393 728.6023  
## Sep 1978 909.9409 730.4251 1089.4567 635.39518 1184.4866  
## Oct 1978 995.4665 806.2402 1184.6927 706.06988 1284.8630  
## Nov 1978 929.5874 731.1252 1128.0495 626.06566 1233.1090  
## Dec 1978 993.7330 786.4460 1201.0200 676.71493 1310.7511  
## Jan 1979 990.5776 774.8265 1206.3288 660.61476 1320.5405  
## Feb 1979 1021.2781 797.3826 1245.1736 678.85943 1363.6967  
## Mar 1979 1075.1129 843.3590 1306.8668 720.67593 1429.5499  
## Apr 1979 1002.2253 762.8709 1241.5797 636.16438 1368.2863  
## May 1979 967.7282 721.0074 1214.4491 590.40124 1345.0552  
## Jun 1979 1032.4996 778.6260 1286.3733 644.23336 1420.7659  
## Jul 1979 881.2474 620.4170 1142.0778 482.34177 1280.1531  
## Aug 1979 469.7581 202.1518 737.3644 60.48953 879.0267  
## Sep 1979 909.9409 635.7260 1184.1558 490.56540 1329.3164  
## Oct 1979 995.4665 714.7986 1276.1344 566.22196 1424.7110  
## Nov 1979 929.5874 642.6115 1216.5632 490.69570 1368.4790  
## Dec 1979 993.7330 700.5849 1286.8811 545.40174 1442.0643

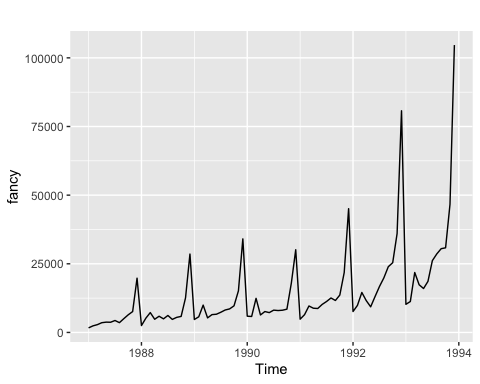
writingBC <- stlf(writing, method='naive', robust=TRUE, lambda = BoxCox.lambda(writing))  
autoplot(writingBC)



**8. Use stlf() to produce forecasts of the fancy series with either method=“naive” or method=“rwdrift”, whichever is most appropriate. Use the lambda argument if you think a Box-Cox transformation is required.**

As there were no signifficant difference for 7 and 8 tasks in using naive or rwdrift, they were chosen arbitrary.

autoplot(fancy)



stlf(fancy, method='rwdrift')

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 1994 62016.88 54142.09 69891.67 49973.43 74060.33  
## Feb 1994 63653.72 52450.20 74857.24 46519.41 80788.02  
## Mar 1994 69171.00 55368.11 82973.89 48061.31 90280.70  
## Apr 1994 66186.51 50154.83 82218.19 41668.17 90704.85  
## May 1994 66161.22 48133.34 84189.09 38589.96 93732.47  
## Jun 1994 67578.60 47716.88 87440.32 37202.72 97954.48  
## Jul 1994 70352.01 48777.36 91926.67 37356.42 103347.60  
## Aug 1994 71702.28 48508.79 94895.77 36230.89 107173.66  
## Sep 1994 73083.34 48346.63 97820.04 35251.81 110914.86  
## Oct 1994 74310.72 48093.07 100528.37 34214.28 114407.16  
## Nov 1994 83745.93 56099.59 111392.27 41464.50 126027.36  
## Dec 1994 113495.30 84464.81 142525.80 69096.99 157893.61  
## Jan 1995 70851.51 40475.31 101227.70 24395.13 117307.89  
## Feb 1995 72488.35 40800.01 104176.69 24025.21 120951.49  
## Mar 1995 78005.63 45034.69 110976.58 27580.93 128430.34  
## Apr 1995 75021.14 40793.82 109248.46 22674.97 127367.31  
## May 1995 74995.85 39535.58 110456.11 20764.06 129227.64  
## Jun 1995 76413.23 39741.10 113085.36 20328.05 132498.42  
## Jul 1995 79186.64 41321.70 117051.59 21277.20 137096.09  
## Aug 1995 80536.91 41496.45 119577.37 20829.67 140244.15  
## Sep 1995 81917.97 41717.77 122118.16 20437.08 143398.86  
## Oct 1995 83145.35 41799.89 124490.82 19912.92 146377.79  
## Nov 1995 92580.56 50103.11 135058.01 27616.91 157544.22  
## Dec 1995 122329.93 78732.75 165927.12 55653.79 189006.07

fancyBC <- stlf(fancy, method='rwdrift', robust=TRUE, lambda = BoxCox.lambda(fancy))  
autoplot(fancyBC)

