Homework 5

Joshua Ingram

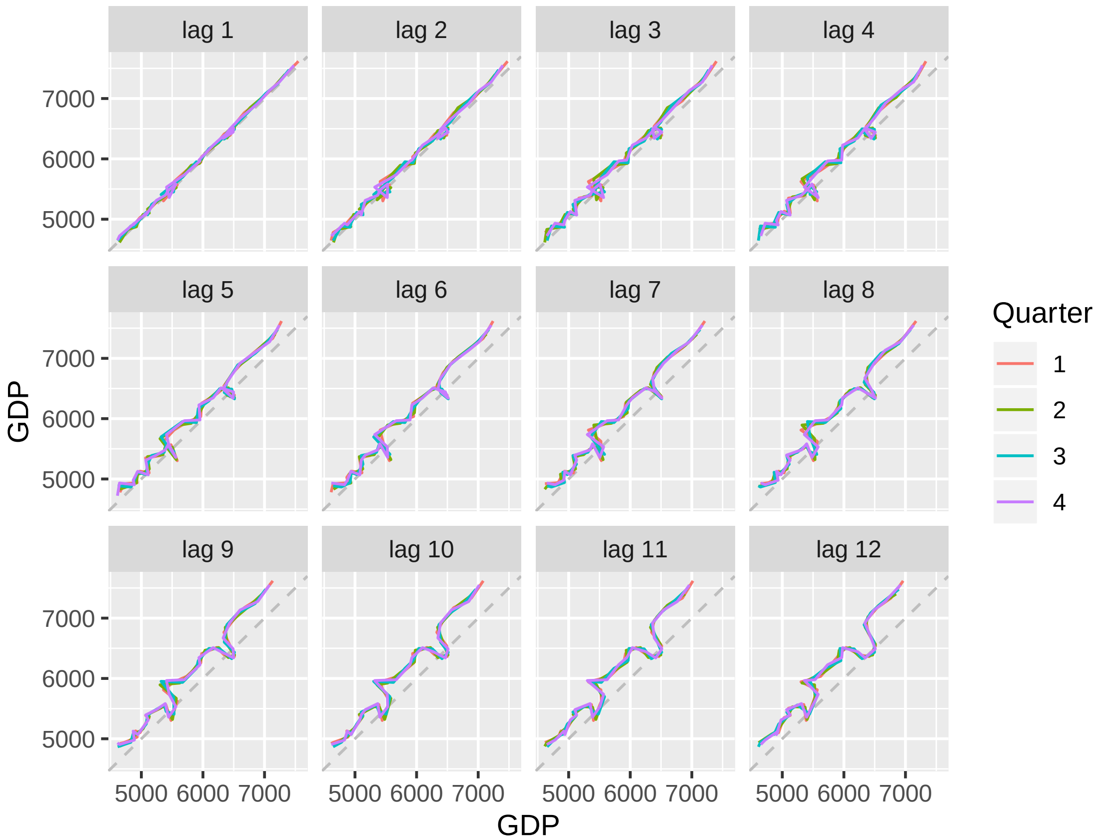
10/1/2020

Problem 1

1.

Australian GDP

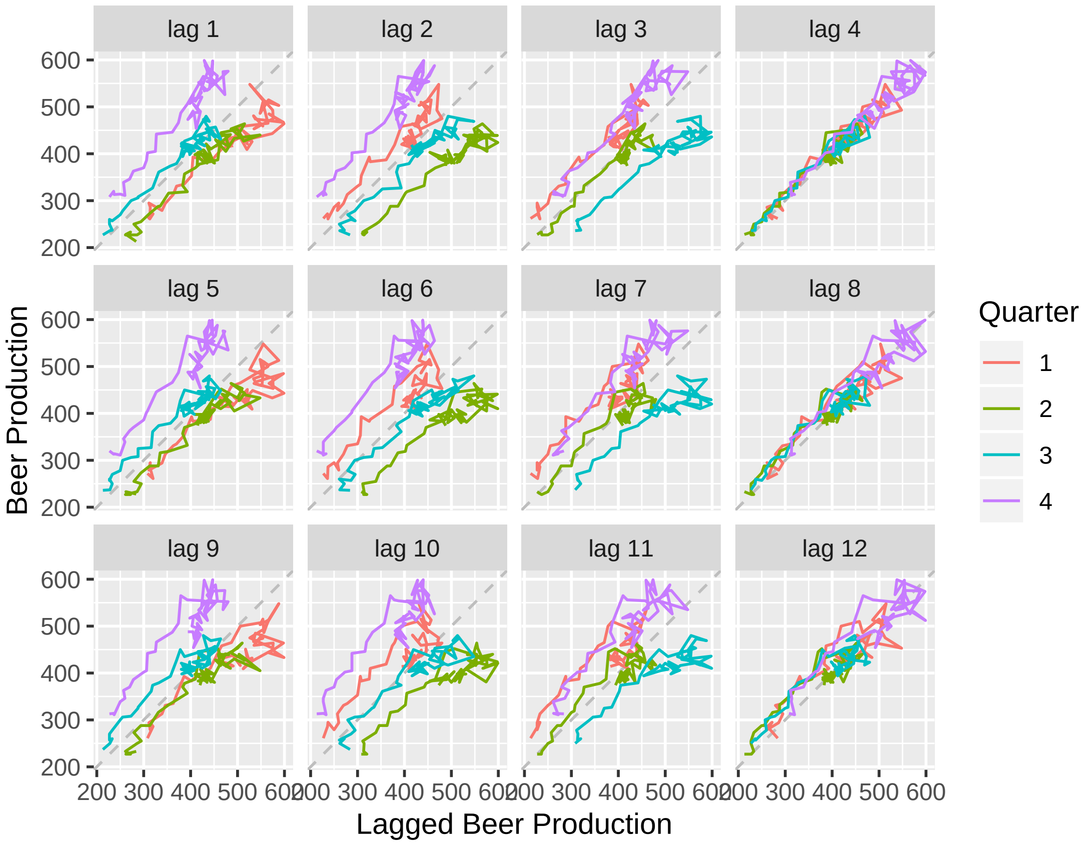
```r
gglagplot(gdp, lags=12) +
  ylab("GDP") + xlab("GDP")
```



There does not appear to be any seasonality based on the lag-plots. Each quarter-line is basically the same as the others. As we increase the number of lags included, there is more of a pattern in the plots (e.g. more dips, different rates of increases depending on time, etc), but no evidence of seasonality.

Australian Beer

```
gglagplot(beer, lags=12) +
  ylab("Beer Production") + xlab("Lagged Beer Production")
```



For lag 4, 8, and 12, there are similar trends in the quarters, with the plots being clustered closely together. The trends differ noticeably for the other lag plots. There does appear to be seasonality because of the clustering seen for lag 4, 8, and 12, meaning that there is dependence of Q4 on the last Q4, whereas the plots of lag 1-3, 5-7, and 9-11 show that there is not much dependence of say Q4 on Q3.

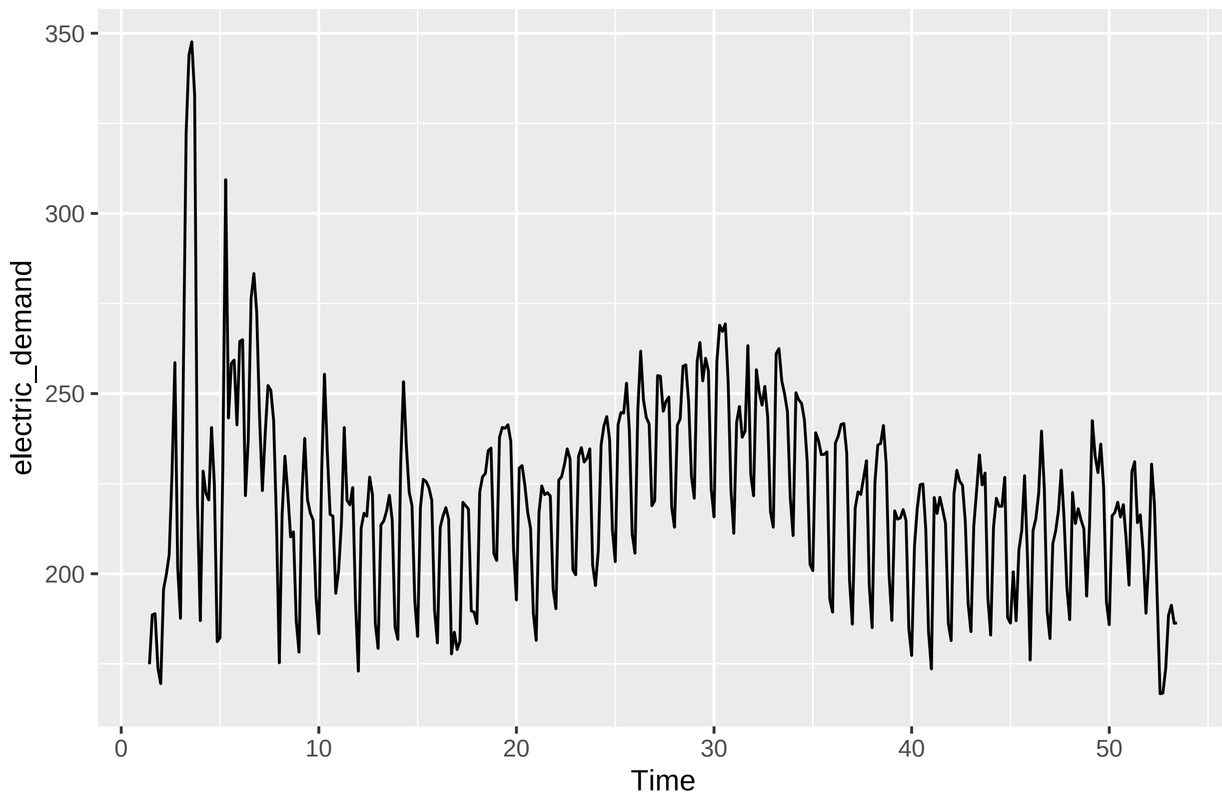
2.

Even if our data has underlying "seasons", such as quarterly data, it does not mean that we observe seasonality/seasonal patterns. This can be shown in the Australian GDP data, where there is an uptrend in GDP but there is not seasonal trend, even though we have quarterly data.

Problem 2

1.

```r
electric_demand <- electric[,"Demand"]
autoplot(electric_demand)
```

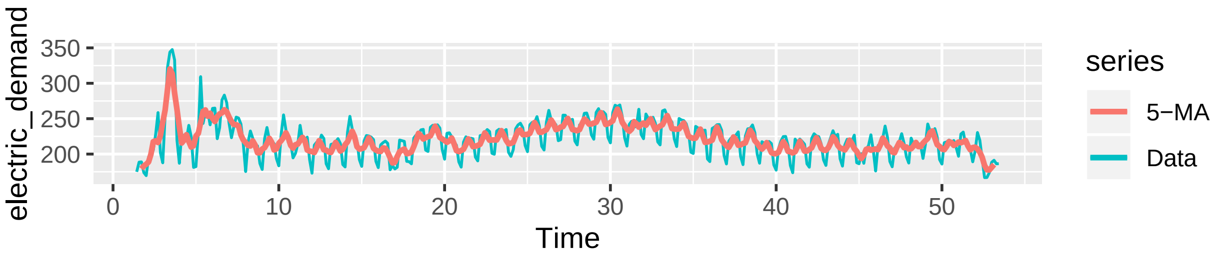


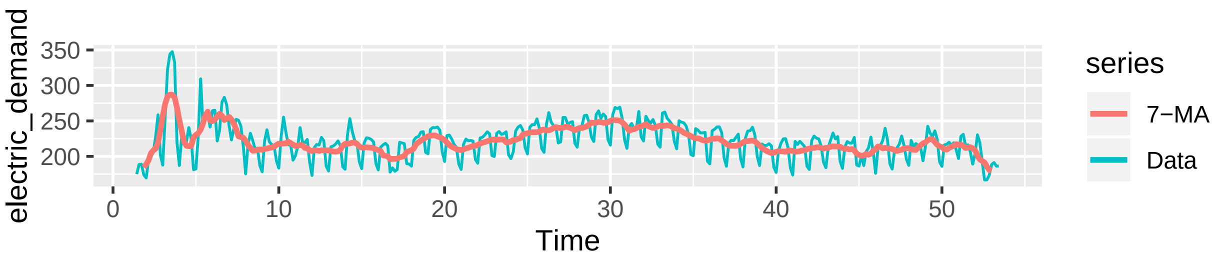
There are 365 observations, with the frequency of our demand data being daily. It seems that there could be some weekly seasonality to the data, as there is a fairily consistent pattern that appears to occur every 7 days. There is also a slight increase in demand around week 30, which would be around June. This is (after a quick Google search) winter time in Australia.

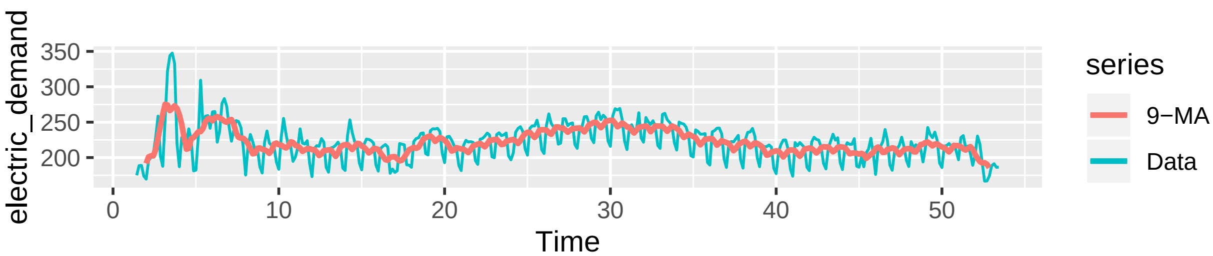
2.

```r
ma_5 <- autoplot(electric_demand, series = "Data") +
  autolayer(ma(electric_demand, 5), series = "5-MA", lwd = 1)
ma_7 <- autoplot(electric_demand, series = "Data") +
  autolayer(ma(electric_demand, 7), series = "7-MA", lwd = 1)
ma_9 <- autoplot(electric_demand, series = "Data") +
  autolayer(ma(electric_demand, 9), series = "9-MA", lwd = 1)

grid.arrange(ma_5, ma_7, ma_9)
```







The 5 day moving average is the least smooth, followed by the 9 day. The 7 day moving average is the smoothest estimate of the trend-cycle component.

3.

This decrease in smoothness, even with an increase in m, is caused by a "bias" towards other days of the week. See the examples for m=5, 7, and 9 below for an explanation at the end.

m = 5 on a Wednsday

$$\hat{T}_{Wed} = \frac{1}{5}\sum_{j=-2}^{2} y_{wed+j},\ \text{j} = \text{number of days away from Wednesday}$$

m = 7 on a Wednsday

$$\hat{T}_{Wed} = \frac{1}{7}\sum_{j=-3}^{3} y_{wed+j},\ \text{j} = \text{number of days away from Wednesday}$$

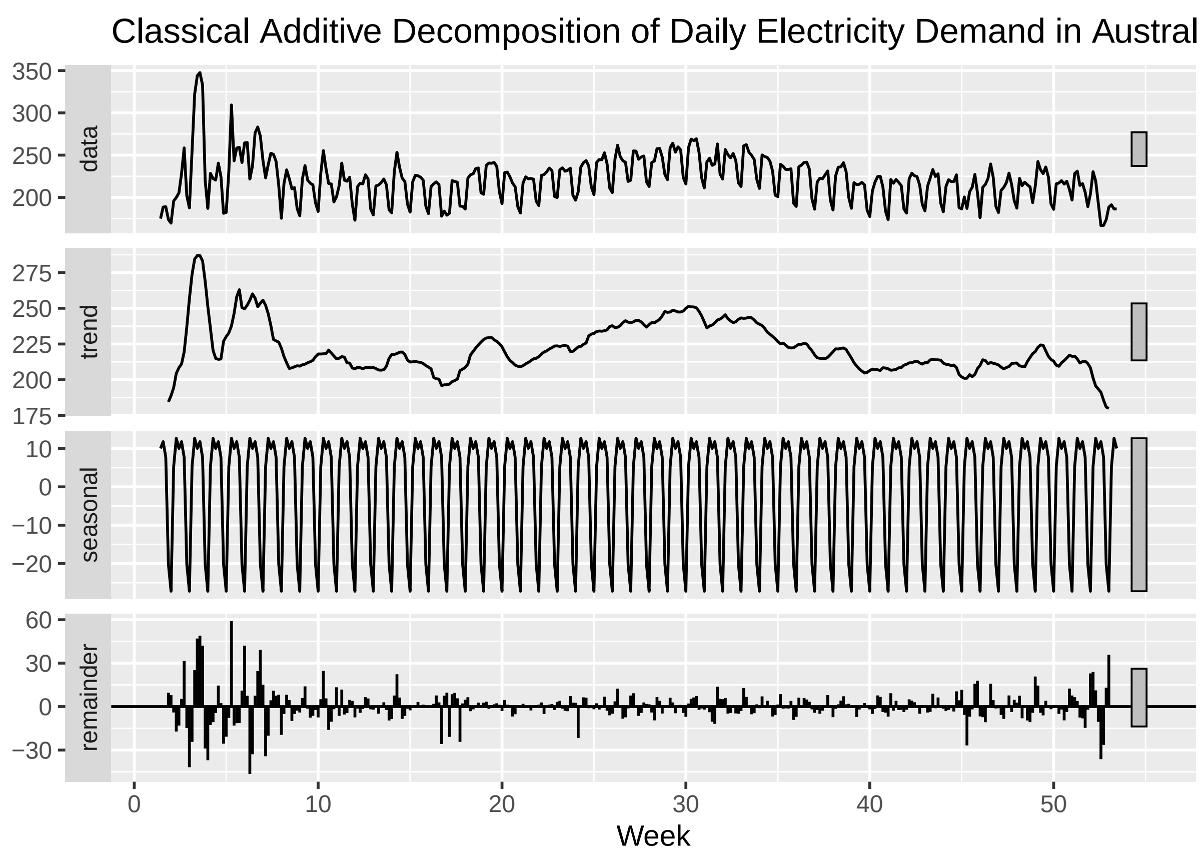
m = 9 on a Wednsday

$$\hat{T}_{Wed} = \frac{1}{9}\sum_{j=-4}^{4} y_{wed+j},\ \text{j} = \text{number of days away from Wednesday}$$

Each y_i is weighted by $\frac{1}{m}$, but when m differs from the season of the data (in this case weeks, so 7), the moving average will become more "biased" twoards specific days. In the case with m=5, Monday through Friday have a weight of $\frac{1}{5}$, but Saturday and Sunday are not included. For m = 7, each day has an equal weight of $\frac{1}{7}$. When m = 9, Saturday and Sunday are given weights of $\frac{2}{9}$ each while Monday-Friday are given weights of $\frac{1}{9}$, thus the moving average is more weighted towards weekend days, causing there to be less "smoothness" with an increase in m.

4.

```r
electric_demand %>% decompose(type="additive") %>% autoplot() +
  labs(title = "Classical Additive Decomposition of Daily Electricity Demand in Australia",
       x = "Week")
```



There is an obvious seasonal component pattern to the demand for electricity, as seen in the "seasonal" graph. The data starts on a Wednesday, so we can see that as we reach Saturday and Sunday the demand drops significantly. On Monday, the demand spikes again. This seasonal pattern is the same throughout the year as seen in the decomposition, though the "classical" approach doesn't allow for changess in the seasonal component.

The trend-cycle component reveals some increase during the middle of the year, which is Australia's winter time. This is likely when people start breaking out the heaters, so the demand for electricity would increase. The data the returns to a mostly "steady" level of demand for the rest of the year. In the beginning of the year, there is also a spike. This could be due to significantly hot months, so people may be using their A/Cs more. I would imagine if we had data for more than one year, we would see the "cyclical" trend of increased demand in winter and hot summer months, then a return to a mostly steady state of demand.

5.

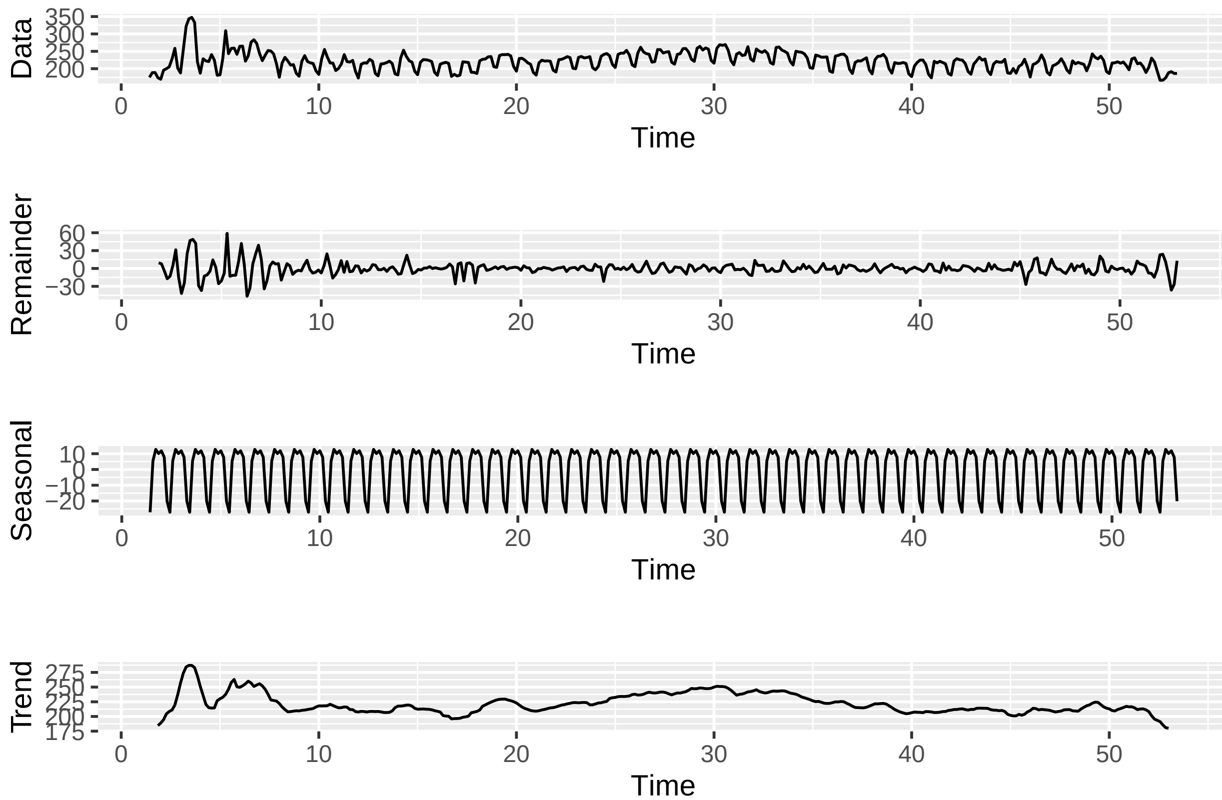
```r
classic_decom <- function(data, freq = 7, len = 52, start = 4){
  # Original data
  Data <- data
  # trend from moving average
  Trend <- ma(Data, freq)
  # detrended data
  detrend <- Data - Trend
  # seasonal component from averaging each season
  seasonal <- c()
  for (i in seq(1,freq,1)){
    subs <- subset(detrend, cycle(detrend) == i)
    subs <- subs[!is.na(subs)]
    mean_day <- mean(subs)
    seasonal <- c(seasonal, mean_day)
  }
  Seasonal <- ts(rep(seasonal, len), start = c(1), frequency = freq)

  # remainder component
  Remainder <- detrend - Seasonal

  Seasonal <- ts(rep(seasonal, len), start = c(1,start), frequency = freq)

  grid.arrange(autoplot(Data),
               autoplot(Remainder),
               autoplot(Seasonal),
               autoplot(Trend),
               ncol = 1)
}

classic_decom(electric_demand)
```

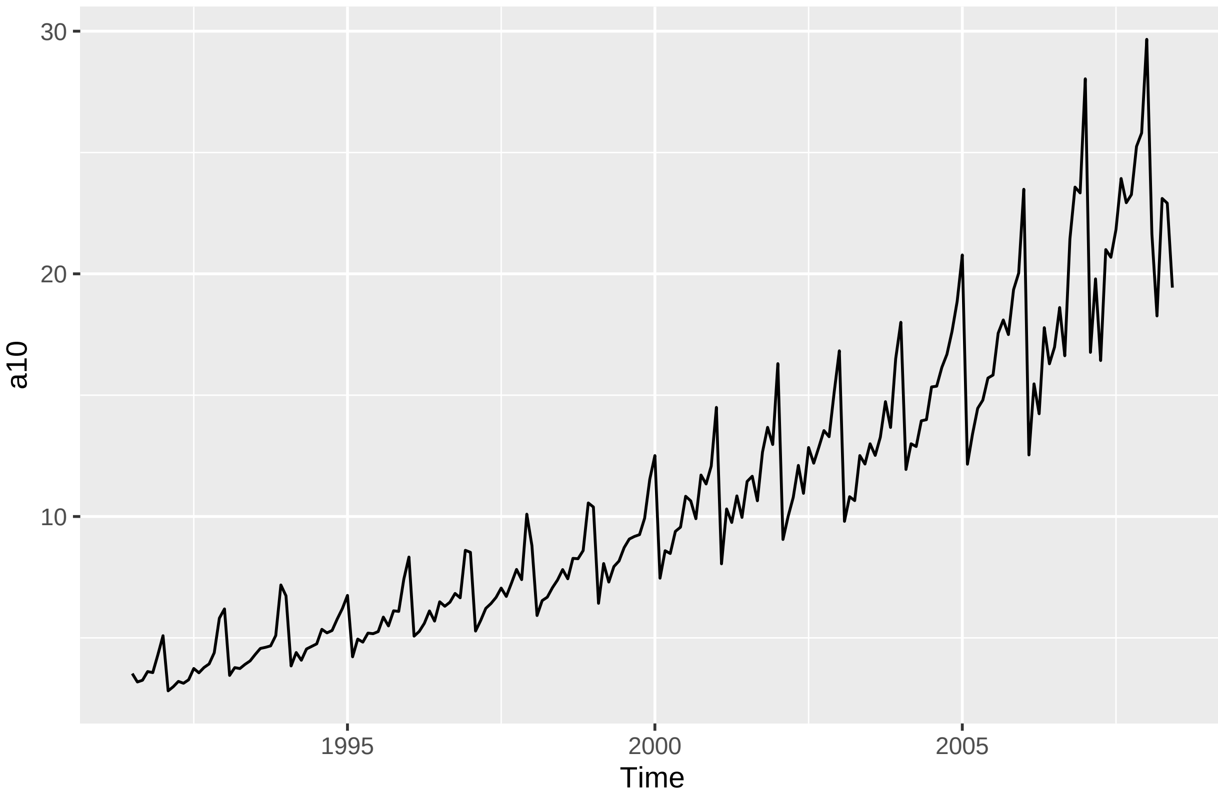


Notice: The plot given in the homework assignment seems to be off in the values for remainder and seasonal plots. My graph is pretty much identical to the output from decompose function from part 4 above, which should be the proper classical decomposition method.

Problem 3

1.

```
autoplot(a10)
```

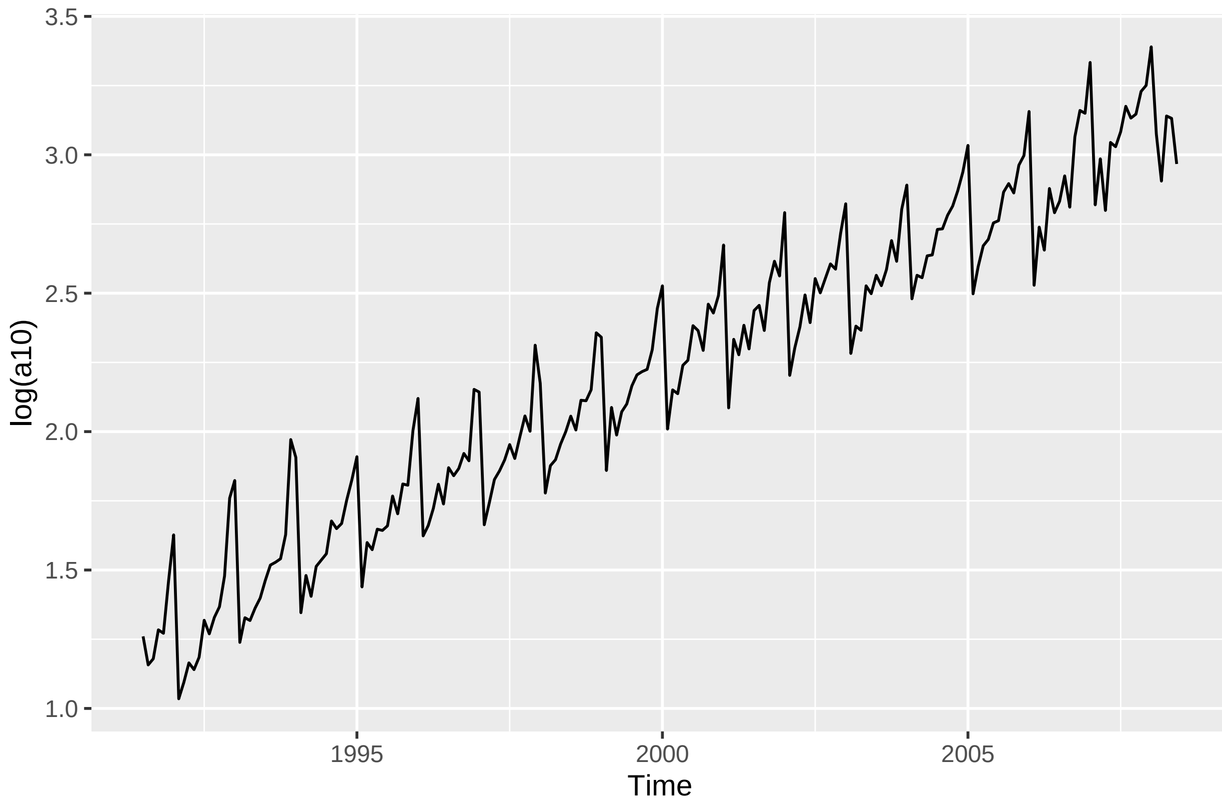


As time goes by and sales increase, the variance increases. We should not apply additive decomposition, but should apply multiplicative decomposition.

2.

We can log the data to stabalize the variance, allowing for us to apply additive decomposition.

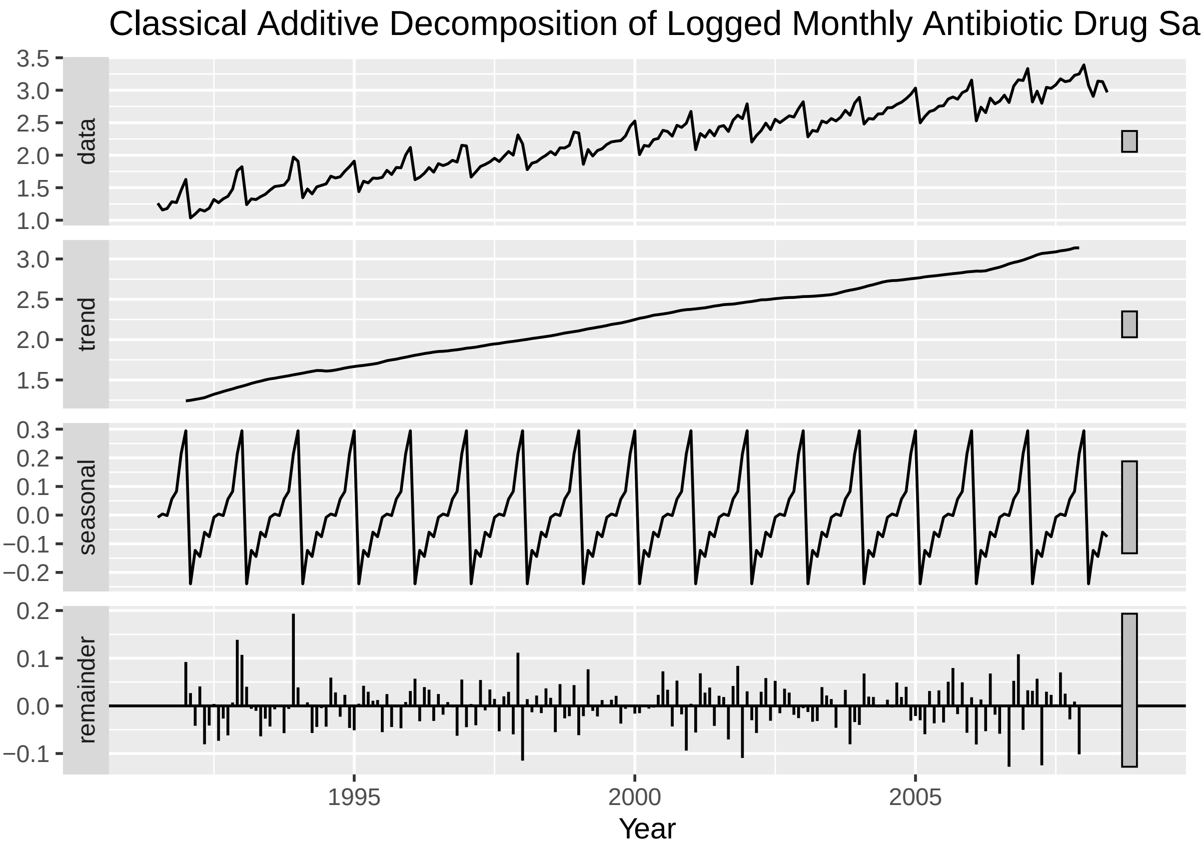
```r
autoplot(log(a10))
```



After logging the data, our variance is now stabalized. However, the units are now in log scale.

3.

```
log(a10) %>% decompose(type="additive") %>% autoplot() +
  labs(title = "Classical Additive Decomposition of Logged Monthly Antibiotic Drug Sales",
       x = "Year")
```



(the following comments are in log scale of sales) The sales of antibiotic drugs increase near the end of the year, followed by a drop sudden drop. This seasonal component continues throughout all the years included in our data. There is an overall global uptrend in sales throughout the years, steadily increasing at a mostly constant rate. The cycles (in log scale) are constant throughout all years. For the m-MA smoothing, m = 2 * 12 was used.