Module 2

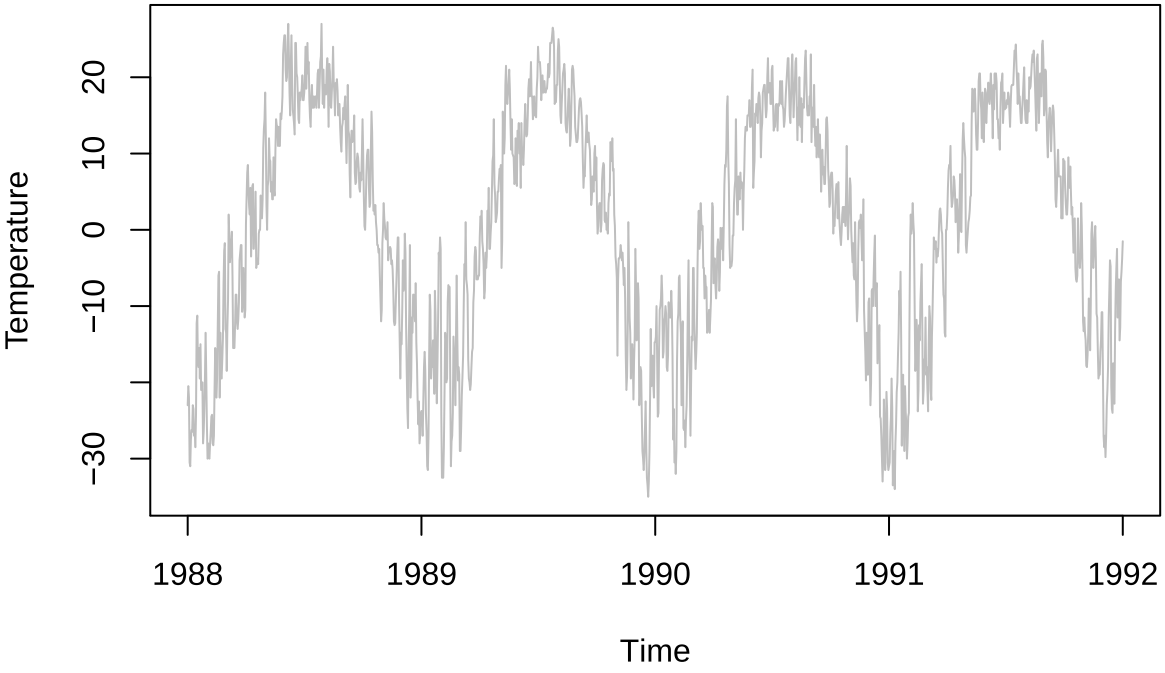
Sri Seshadri

4/11/2018

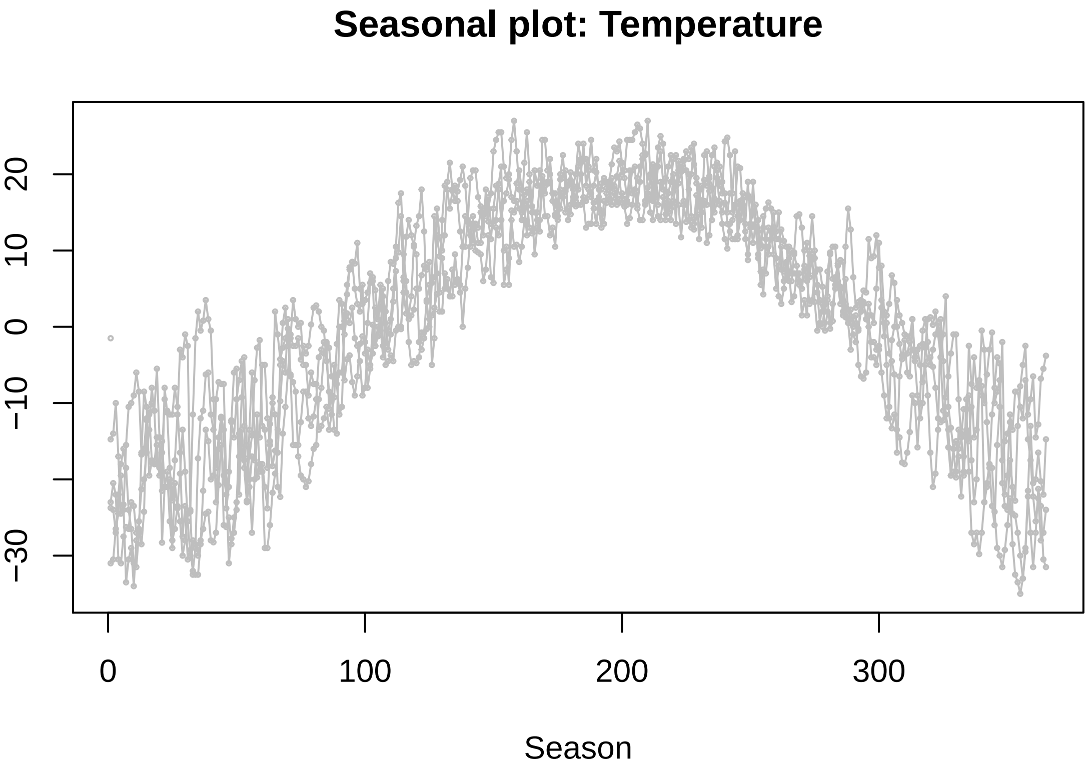
I chose to look at the mean daily temperature of Fisher river at Dallas TX (https://datamarket.com/data/set/235d/mean-daily-temperature-fisher-river-near-dallas-jan-01-1988-to-dec-31-1991#!ds=235d&display=line)

As you can see the data is seasonal (yearly). I set the frequency to be at 365 days for the time series. The seasonal plot gives it away too.

```
library(lubridate)
MeanTemp <- readxl::read_xlsx("mean-daily-temperature-fisher-ri-2.xlsx")
Temperature <- ts(MeanTemp$Temp,start = decimal_date(MeanTemp$Time[1]), frequency = 365)
plot(Temperature,col = "grey")
```



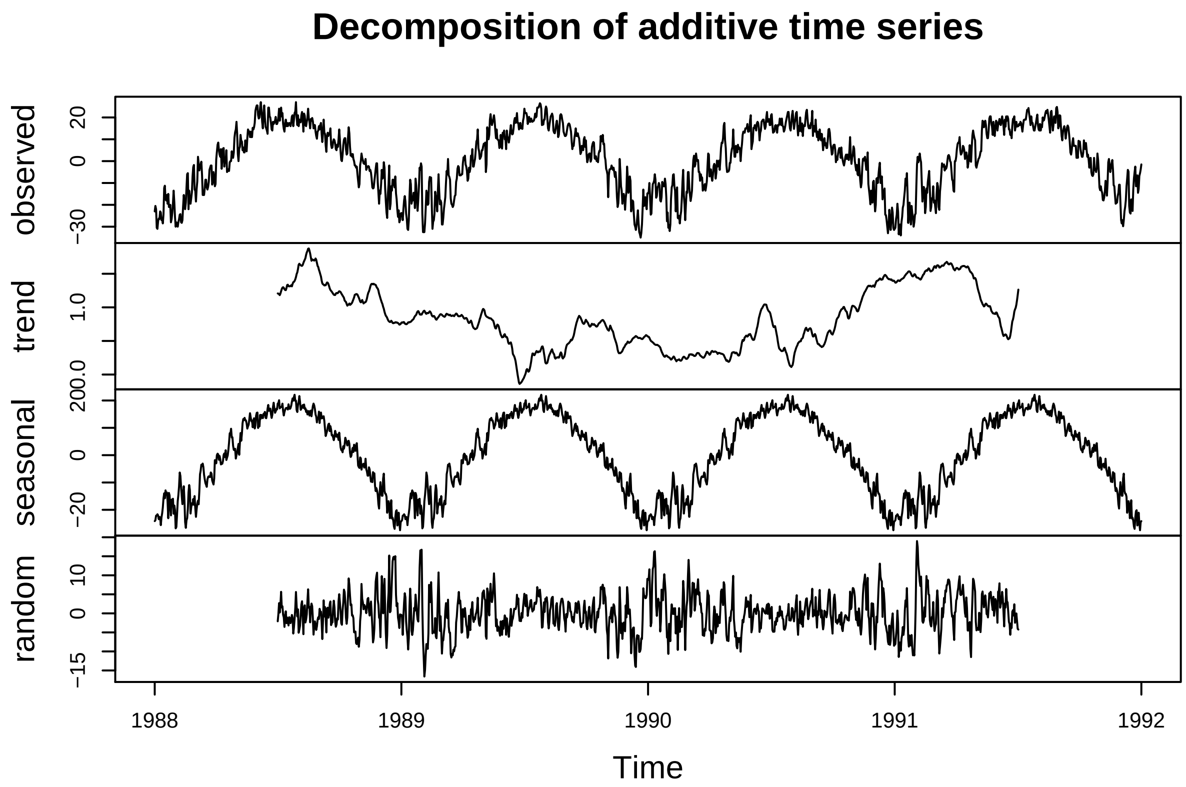
```
seasonplot(Temperature,cex = 0.3,col = "grey")
```



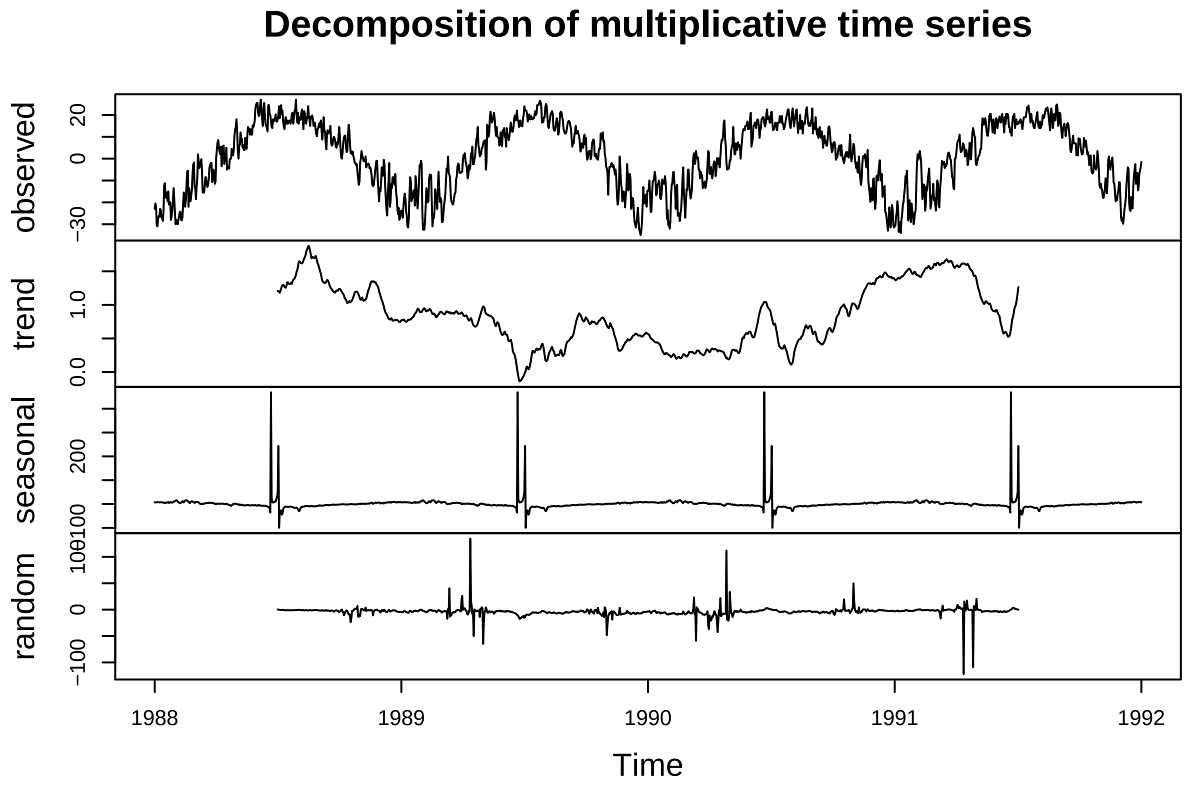
Decomposition

The magnitute of the seasonal variations is low and the there is not much of a trend. (Please pay attention to the scale of the plots below). The additive model would be appropriate here. Also one can see the random error spikes in the multiplicative decomposition to reach as high as 100, whereas the random error in the additive model is within +/- 15.

```
plot(decompose(Temperature,type = "additive"))
```



```r
plot(decompose(Temperature,type = "multiplicative"))
```



Forecasting

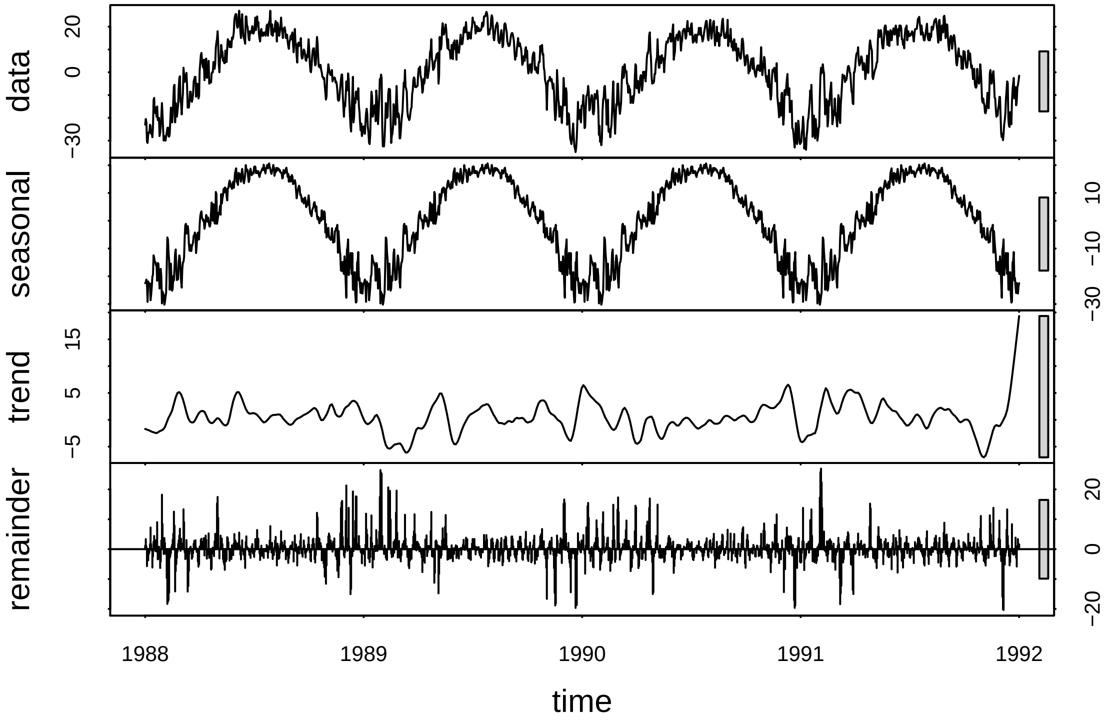
Classical decomposition's drawback

While the classical decomposition is good, the moving averages employed in the decomposition makes us lose the forecasts for the inital half a year and the final half a year. Which is a big drawback. Hence cannot be used for forecasting. The classical decomposition is more for studying the data more so than prediction.

STL - Seasonal trend decomposition using loess

I tried STL with a window of 10, 15 and 30. I am showing 30 here. May be I should split the data into training and test and do some kind of cross validation to choose the right t.window parameter. I am not sure, thoughts??

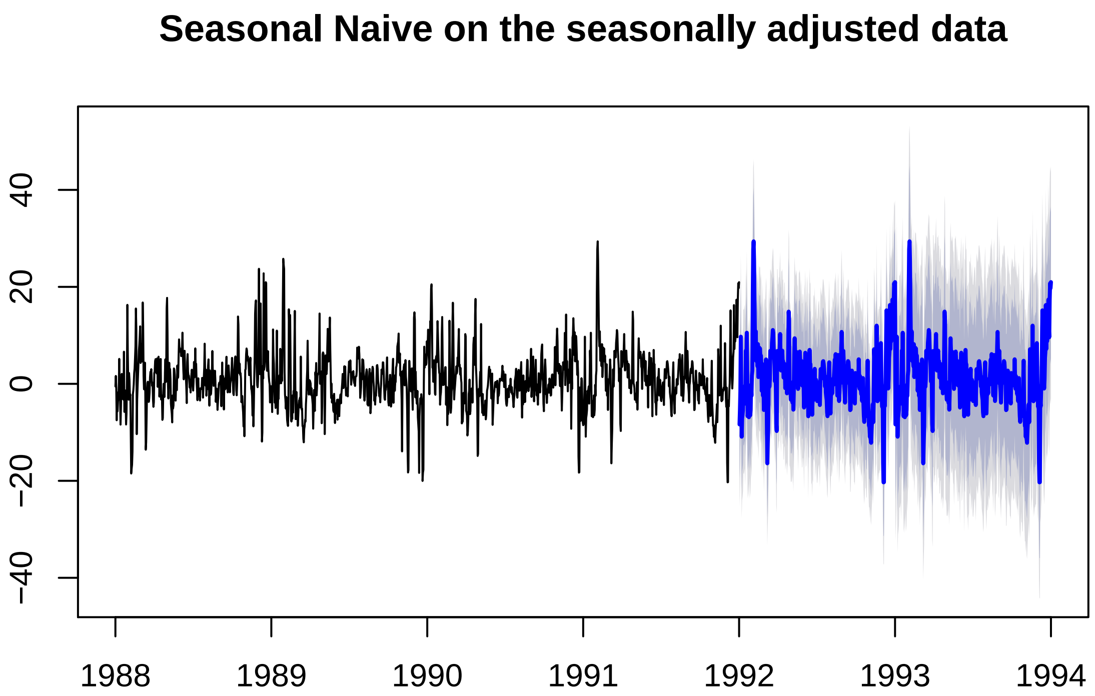
```
stl.fit <- stl(x = Temperature,s.window = "periodic",t.window = 30,robust = T)
plot(stl.fit)
```



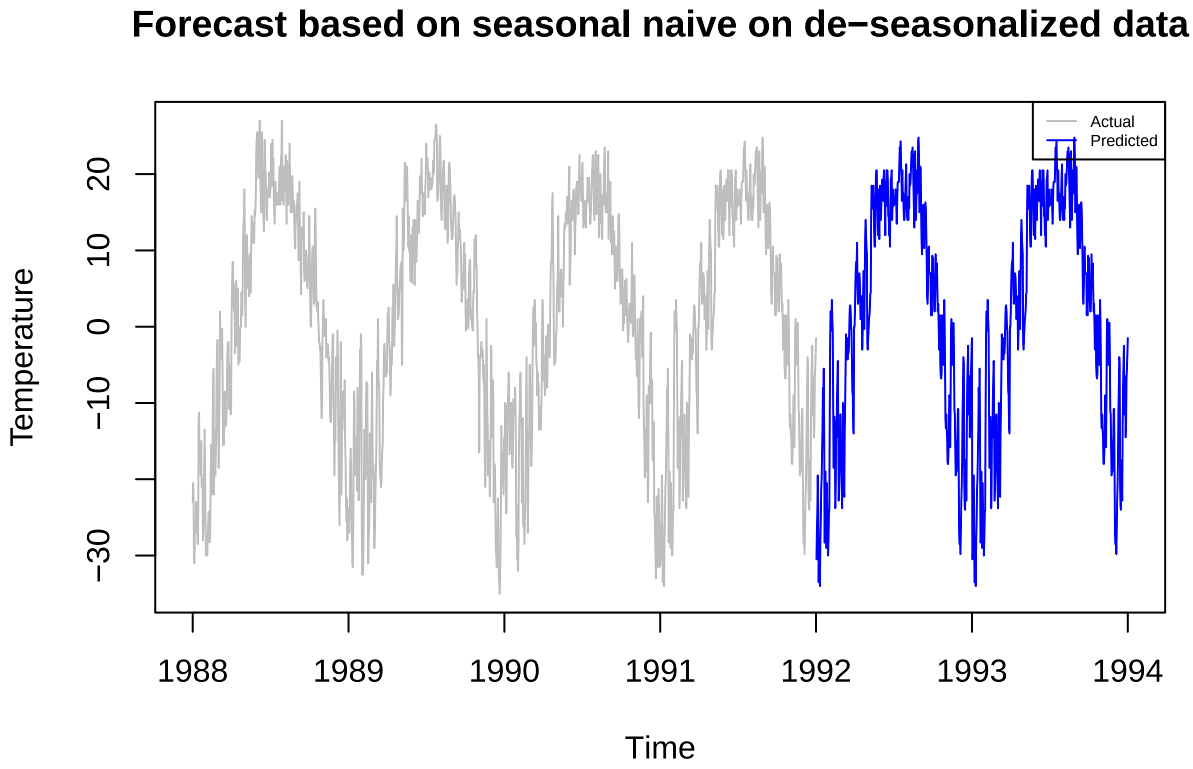
After seasonally adjusting the data, I tried to fit a seasonal naive model on the seasonally adjusted data. You can see there are some fluctuations at the begining of each year. This seasonal naive came out to be a better model.

As you can see from the plots below, the seasonal naive predictions on the seasonally adjusted data seems better than the out of the boc STL.

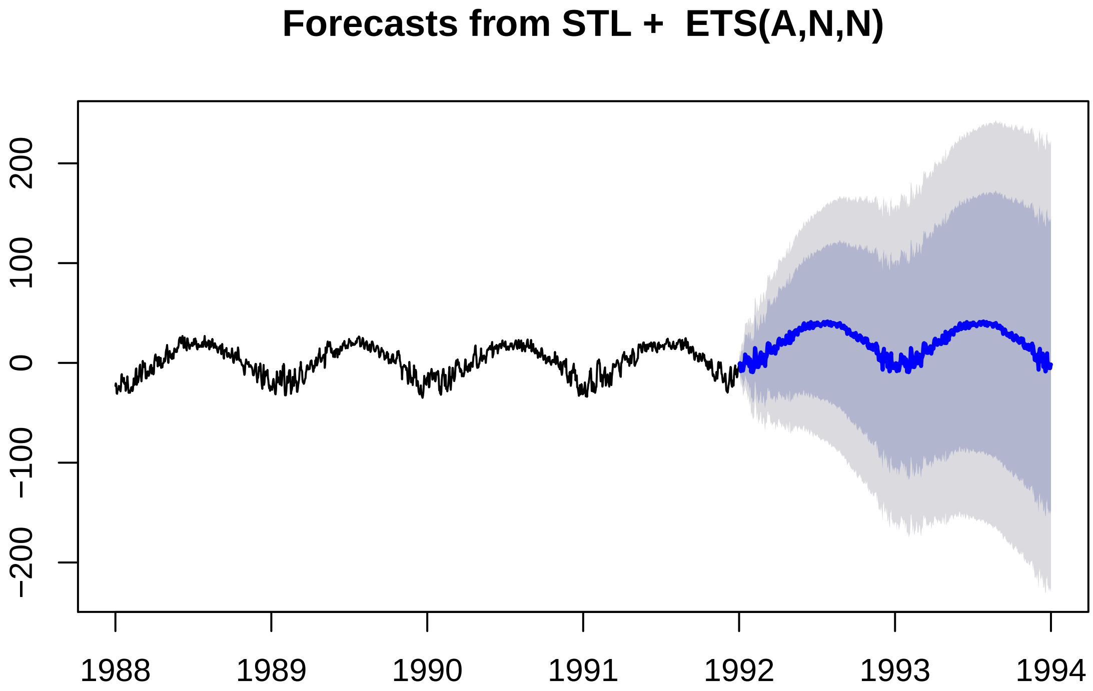
```
seasonallyAdj <- forecast::seasadj(stl.fit)
plot(snaive(seasonallyAdj), main = "Seasonal Naive on the seasonally adjusted data")
```



```
seasonalDataFit <- snaive(stl.fit$time.series[,1],h = 365*2)
seasonallyAdjFit <- snaive(seasonallyAdj, h= 365*2)
fcts <- seasonallyAdjFit$mean + seasonalDataFit$mean
plot(Temperature,col = "grey",xlim = c(1988,1994), main = "Forecast based on seasonal naive on de-seaso
lines(fcts, col = "blue")
legend("topright", legend = c("Actual", "Predicted"), col = c("grey", "blue"),cex = 0.5, lty = 1)
```



```
forecasts <- forecast::forecast(stl.fit)
plot(forecasts)
```



Your thoughts??

Sri