Ly,Cuong-Lab-3

Cuong Ly

2023-02-08

Introduction

Load the Climate data `climateDC.csv` that you used for the Quiz.

Downloaded the dataset using NOAA.

https://www.ncdc.noaa.gov/cdo-web/search

Here, I'm pre-processing the data.

```
data<-read.csv("climateDC.csv",header =TRUE)
head(data)
```

```
##       STATION                          NAME LATITUDE LONGITUDE ELEVATION   DATE
## 1 USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2 1/1/73
## 2 USC00182325  DALECARLIA RESERVOIR, MD US 38.93850 -77.11340      45.7 1/1/73
## 3 USC00189035  U S SOLDIERS HOME DC, MD US 38.93333 -77.01667      70.1 1/1/73
## 4 USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2 1/2/73
## 5 USC00182325  DALECARLIA RESERVOIR, MD US 38.93850 -77.11340      45.7 1/2/73
## 6 USC00189035  U S SOLDIERS HOME DC, MD US 38.93333 -77.01667      70.1 1/2/73
##   PRCP SNOW SNWD TMAX TMIN TOBS
## 1 0.00    0    0   64   46   46
## 2 0.01    0    0   63   51   53
## 3 0.00    0    0   65   53   55
## 4 0.00    0    0   63   38   43
## 5 0.00    0    0   54   32   47
## 6 0.00    0    0   55   37   40
```

```
#########################
### Check for missing values ########

# is.na(data)

#Extract rows with missing values
df_na_rows <- data[which(rowSums(is.na(data)) > 0),]

#Extract columns with missing values
df_na_cols <- data[, which(colSums(is.na(data)) > 0)]

library(imputeTS)
imputed_time_series <- na_ma(data, k = 4, weighting = "exponential")
```

```
df<-data.frame(imputed_time_series)
str(df)
```

```
## 'data.frame':    46893 obs. of  12 variables:
##  $ STATION  : chr  "USC00186350" "USC00182325" "USC00189035" "USC00186350" ...
##  $ NAME     : chr  "NATIONAL ARBORETUM DC, MD US" "DALECARLIA RESERVOIR, MD US" "U S SOLDIERS HOME DC
##  $ LATITUDE : num  38.9 38.9 38.9 38.9 38.9 ...
##  $ LONGITUDE: num  -77 -77.1 -77 -77 -77.1 ...
##  $ ELEVATION: num  15.2 45.7 70.1 15.2 45.7 70.1 15.2 45.7 70.1 15.2 ...
##  $ DATE     : chr  "1/1/73" "1/1/73" "1/1/73" "1/2/73" ...
##  $ PRCP     : num  0 0.01 0 0 0 0 0 0 0 0.52 ...
##  $ SNOW     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SNWD     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ TMAX     : num  64 63 65 63 54 55 45 48 44 51 ...
##  $ TMIN     : num  46 51 53 38 32 37 24 25 29 32 ...
##  $ TOBS     : num  46 53 55 43 47 40 37 38 37 50 ...
```

```
df$DATE<-as.Date(df$DATE,format = "%m/%d/%y")

library(lubridate)
df$year <- year(df$DATE)
head(df)
```

```
##       STATION                          NAME LATITUDE LONGITUDE ELEVATION
## 1 USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2
## 2 USC00182325  DALECARLIA RESERVOIR, MD US 38.93850 -77.11340      45.7
## 3 USC00189035  U S SOLDIERS HOME DC, MD US 38.93333 -77.01667      70.1
## 4 USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2
## 5 USC00182325  DALECARLIA RESERVOIR, MD US 38.93850 -77.11340      45.7
## 6 USC00189035  U S SOLDIERS HOME DC, MD US 38.93333 -77.01667      70.1
##         DATE PRCP SNOW SNWD TMAX TMIN TOBS year
## 1 1973-01-01 0.00    0    0   64   46   46 1973
## 2 1973-01-01 0.01    0    0   63   51   53 1973
## 3 1973-01-01 0.00    0    0   65   53   55 1973
## 4 1973-01-02 0.00    0    0   63   38   43 1973
## 5 1973-01-02 0.00    0    0   54   32   47 1973
## 6 1973-01-02 0.00    0    0   55   37   40 1973
```

```
######## getting data for only one station
table(df$NAME)
```

```
##
##  DALECARLIA RESERVOIR, MD US NATIONAL ARBORETUM DC, MD US
##                        17919                        17592
##       THE WHITE HOUSE, DC US  U S SOLDIERS HOME DC, MD US
##                          972                         1826
##     WASHINGTON 1.2 SE, DC US     WASHINGTON 1.9 NW, DC US
##                          145                          262
##    WASHINGTON 1.9 WNW, DC US    WASHINGTON 2.0 ESE, DC US
##                          654                          392
##    WASHINGTON 2.0 SSW, DC US     WASHINGTON 2.6 NE, DC US
##                         1540                          776
```

```
##    WASHINGTON 3.0 ENE, DC US    WASHINGTON 3.1 NNE, DC US
##                          122                          655
##    WASHINGTON 3.7 NNW, DC US    WASHINGTON 4.0 WNW, DC US
##                          112                          128
##      WASHINGTON 5.0 N, DC US    WASHINGTON 5.0 WNW, DC US
##                          284                          116
##     WASHINGTON 5.1 NW, DC US
##                         3398
```

```
newdf<-df[df$NAME=="NATIONAL ARBORETUM DC, MD US",]
head(newdf)
```

```
##        STATION                          NAME LATITUDE LONGITUDE ELEVATION
## 1  USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2
## 4  USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2
## 7  USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2
## 10 USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2
## 13 USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2
## 16 USC00186350 NATIONAL ARBORETUM DC, MD US 38.91329 -76.97009      15.2
##          DATE PRCP SNOW SNWD TMAX TMIN TOBS year
## 1  1973-01-01 0.00    0    0   64   46   46 1973
## 4  1973-01-02 0.00    0    0   63   38   43 1973
## 7  1973-01-03 0.00    0    0   45   24   37 1973
## 10 1973-01-04 0.52    0    0   51   32   50 1973
## 13 1973-01-05 0.00    0    0   51   38   44 1973
## 16 1973-01-06 0.00    0    0   45   29   30 1973
```

```
############ for a larger window
#library(dplyr)
start_date <- as.Date("2012-01-01")
data_subset <- filter(newdf, DATE >= start_date)

##### Just looking at some Variables ############
fig <- plot_ly(data_subset, x = ~DATE, y = ~TOBS,name = 'Temperature', type = 'scatter', mode = 'lines'
fig <- fig %>% add_trace(y = ~SNOW, size = ~SNOW,name = 'Snow Fall', mode = 'markers')
fig <- fig %>% add_trace(y = ~SNWD, size = ~PRCP,name = 'Percipitation', mode = 'markers')

fig <- fig %>% layout(title = 'Weather Data in Washington DC',
                      xaxis = list(showgrid = FALSE),
                      yaxis = list(showgrid = FALSE))

#fig
```

Here I'm creating time series object for the original data for temperature

`temp` is the original data for temperature.Whereas `temp_month` is data that I aggregated over month so I get monthly temperature.

```
temp<-ts(newdf$TOBS,star=decimal_date(as.Date("1973-01-01",format = "%Y-%m-%d")),frequency = 365.25)


### can get monthly data
# Get mean value for each month
```

```
mean_data <- newdf %>%
  mutate(month = month(DATE), year = year(DATE)) %>%
  group_by(year, month) %>%
  summarize(mean_value = mean(TOBS))

temp_month<-ts(mean_data$mean_value,star=decimal_date(as.Date("1973-01-01",format = "%Y-%m-%d")),freque
```

Here I'm creating time series object for a smaller window of the data for temperature

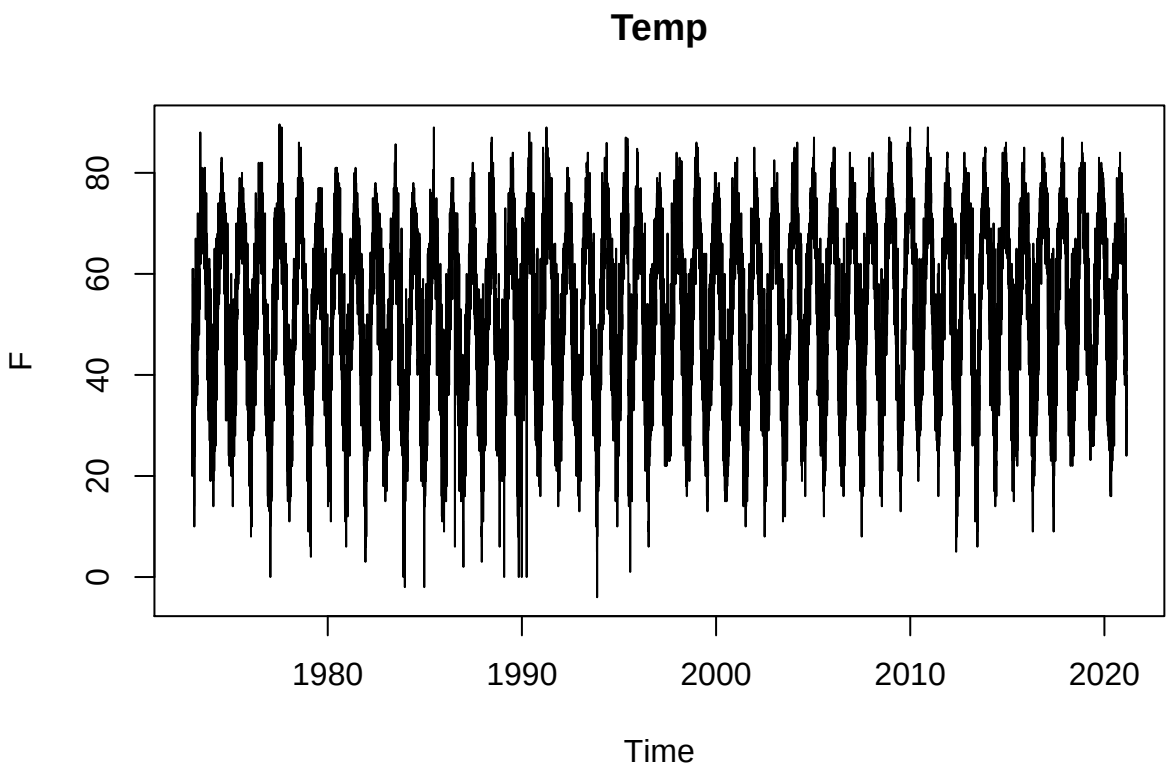
`tempL` is the for temperature for this data in a smaller window. Similarly, `temp_monthL` is data that I aggregated over month so I get monthly temperature.

```
########### for a smaller window ###################

tempL<-ts(data_subset$TOBS,star=decimal_date(as.Date("2012-01-01",format = "%Y-%m-%d")),frequency = 365

### can get monthly data
# Get mean value for each month
mean_data <- data_subset  %>%
  mutate(month = month(DATE), year = year(DATE)) %>%
  group_by(year, month) %>%
  summarize(mean_value = mean(TOBS))


temp_monthL<<-ts(mean_data$mean_value,star=decimal_date(as.Date("2012-01-01",format = "%Y-%m-%d")),freq
```

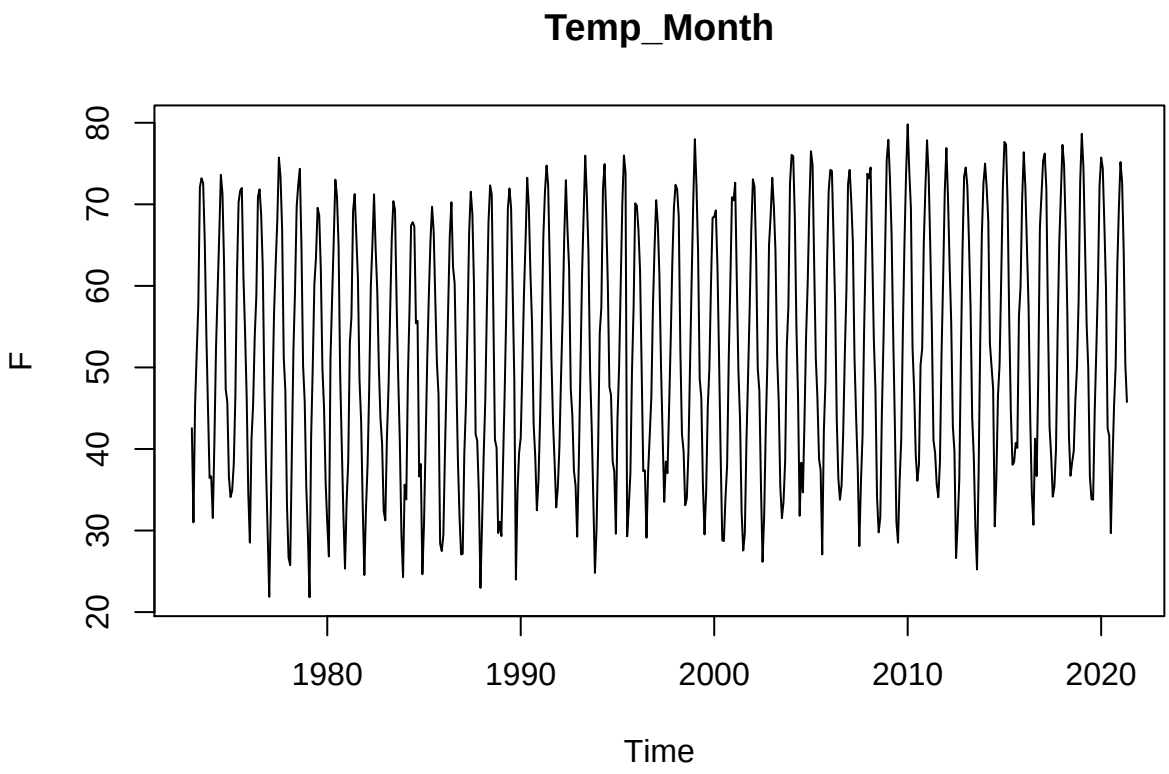
Problem 1

a. Provide a data visualizations for `temp` and `temp_month` and `tempL` and `temp_monthL`. Compare and discuss.

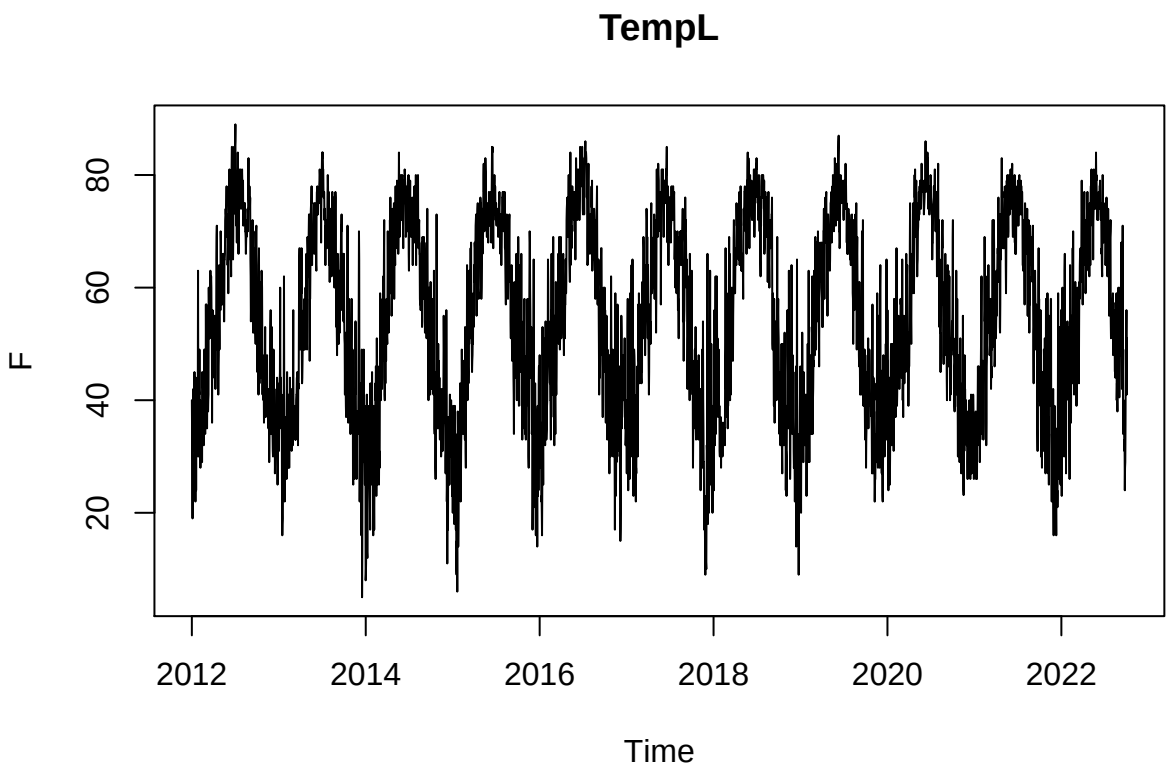
```
plot(temp,main = 'Temp',ylab='F')
```



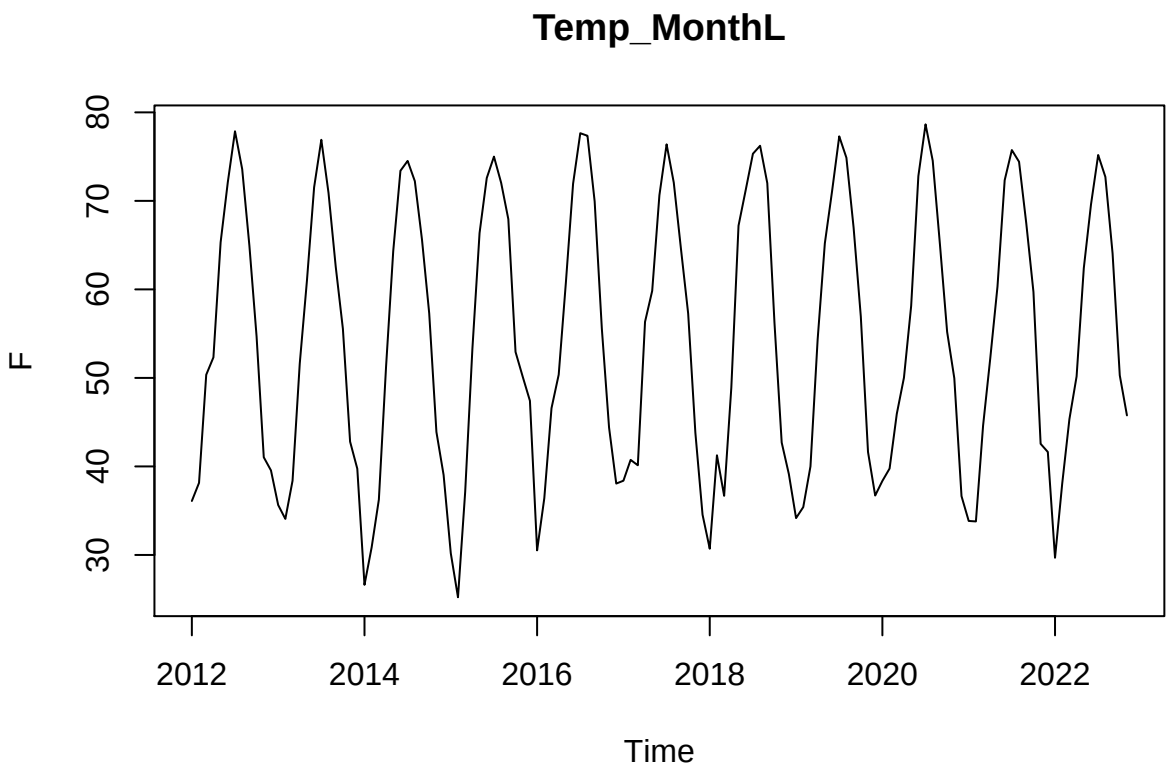

```
plot(temp_month,main = 'Temp_Month',ylab='F')
```




```
plot(tempL,main = 'TempL',ylab='F')
```



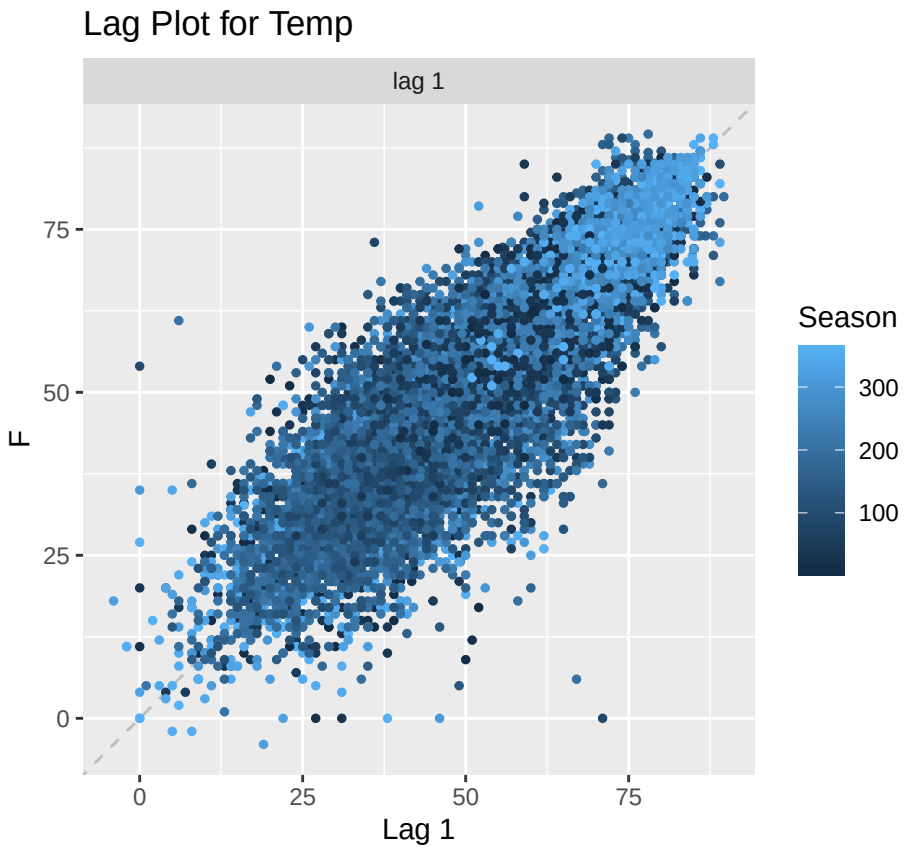

```
plot(temp_monthL,main = 'Temp_MonthL',ylab='F')
```



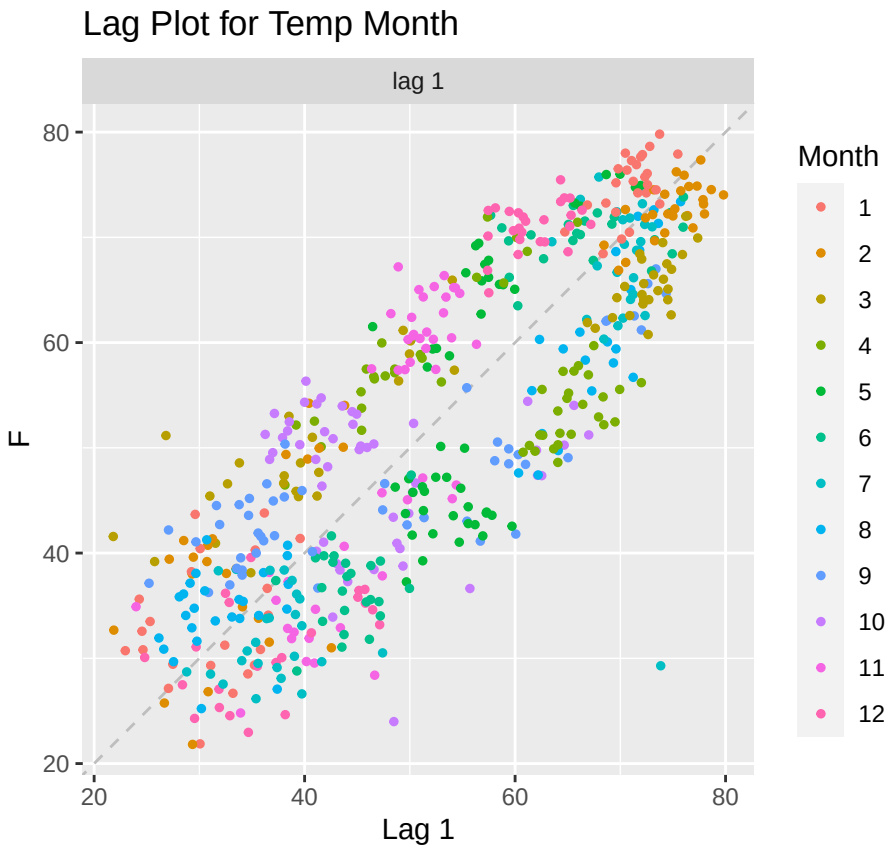
The temperature from 1973 to 2022 ranging from -3 to 85. There's seasonality in all the data, although we can see it better in tempL and temp_monthL, since they have a smaller window. In the temp chart, we can say that in the recent years, the temperature is less extreme than in the past. The range in recent years is smaller than the 80's or 90's. The graph of temp_month and temp_monthL are smoother than their counterparts, as expected, since they are computed as the moving average. The data also confirms that the temperature is high during summer and low in the winter months.

b. Obtain Lag plots for `temp` and `temp_month` and `tempL` and `temp_monthL`. Compare and discuss.

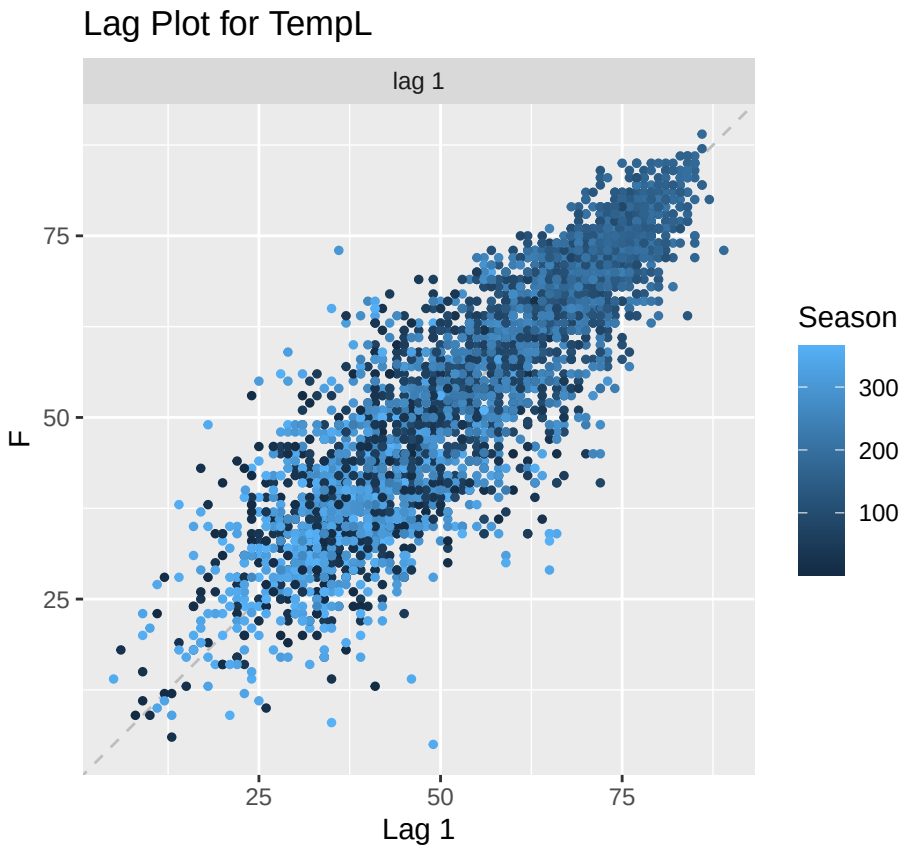
```
gglagplot(temp, do.lines=FALSE, lags=1)+xlab("Lag 1")+ylab("F")+ggtitle("Lag Plot for Temp")
```



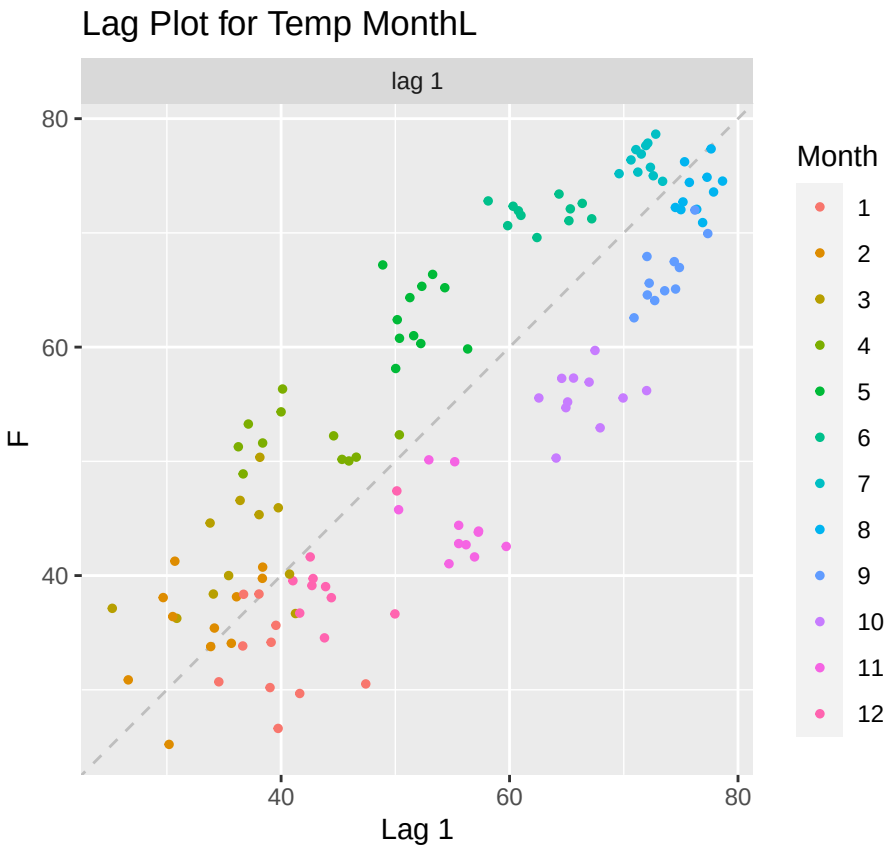

```
gglagplot(temp_month, do.lines=FALSE, lags=1)+xlab("Lag 1")+ylab("F")+ggtitle("Lag Plot for Temp Month")
```




```
gglagplot(tempL, do.lines=FALSE, lags=1)+xlab("Lag 1")+ylab("F")+ggtitle("Lag Plot for TempL")
```



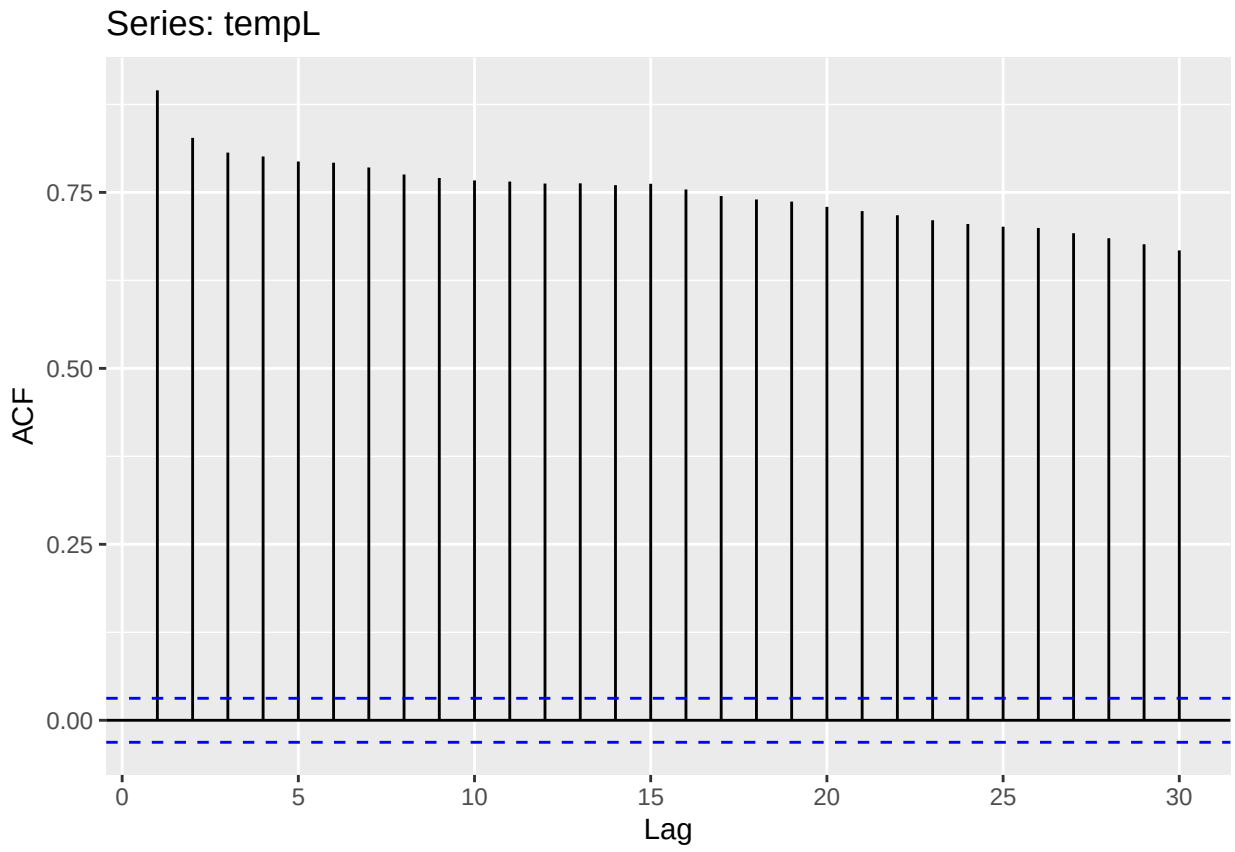

```
gglagplot(temp_monthL, do.lines=FALSE, lags=1)+xlab("Lag 1")+ylab("F")+ggtitle("Lag Plot for Temp Monthl
```



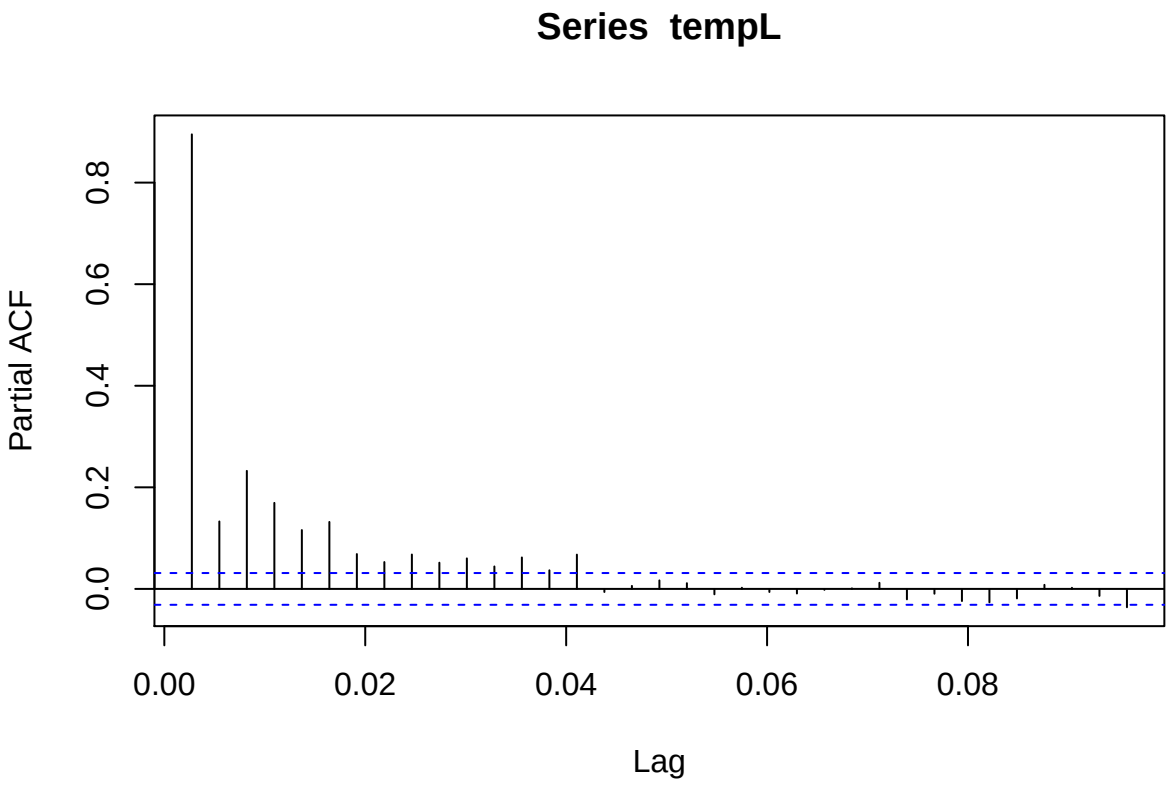
The data mostly concentrated along the diagonal line. The linear shape lag plot suggests there's autocorrelation and we can use the autoregressive model to predict and forecast data. However, the points do not perfectly line up on the diagonal, there are outliers, suggesting that there might be some random component to the data.

c. Check the ACF and PACF plots for `tempL` and `temp_monthL`. Discuss about stationary of the series, correlation that you can observe between lag variables..etc

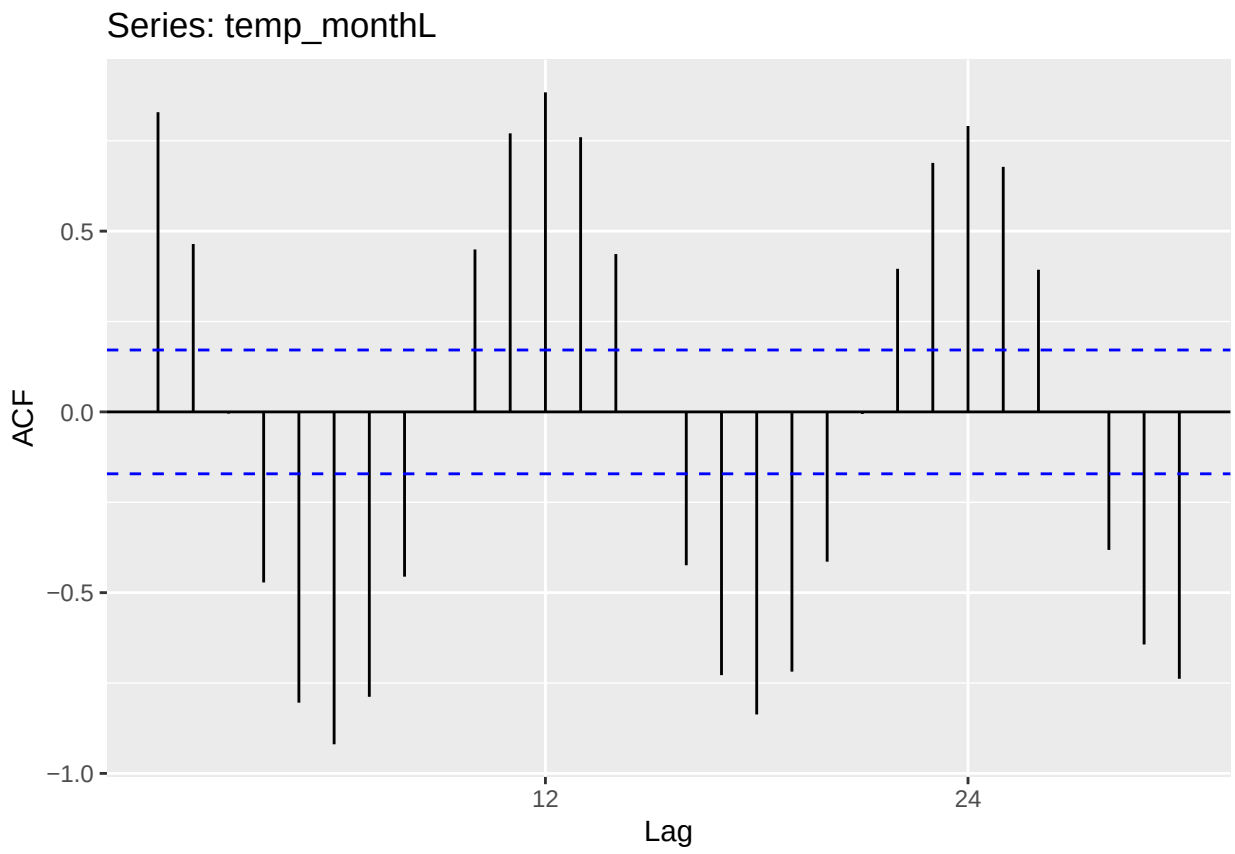
```
ggAcf(tempL, 30)
```



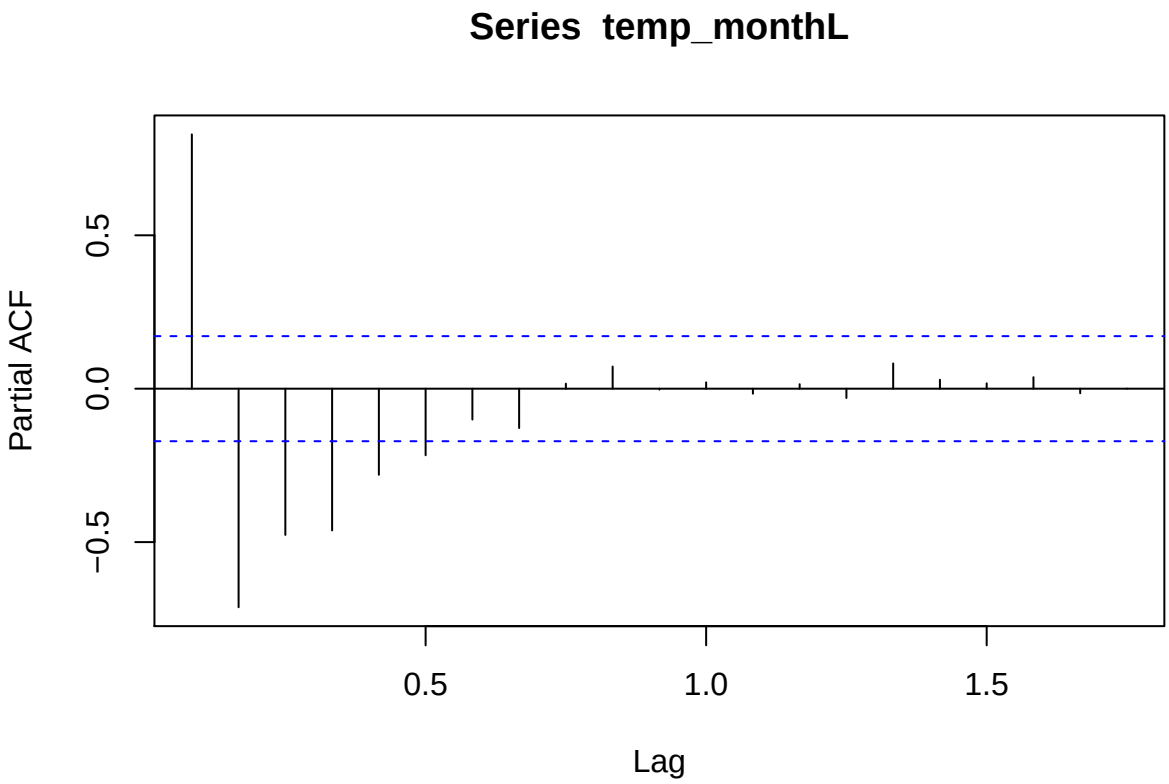

```
pacf(tempL)
```




```
# The tempL is stationary

ggAcf(temp_monthL, 30)
```




```
pacf(temp_monthL)
```



From the ACF of tempL, the correlation decreases slowly, which indicate that this is an AR model. The series is not stationary since the correlation decreases slowly. From the PACF of tempL, strong correlation from lag 1 to lag 12, with the strongest correlation at lag 2 for tempL. For temp_monthL, there's definitely stationary displayed. The data is stationary since the ACF drop to 0 relatively quickly.

d. Compare the results you obtained from part b) with part c) (do this only for the `tempL` and `temp_monthL`) The result in part c corroborated the findings from part b

e. Use an ADF test to check for the Stationarity of the series of `tempL` and `temp_monthL`. Does this result comply with what you have found in part c) from ACF plots? Discuss.

```
tseries::adf.test(tempL)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  tempL
## Dickey-Fuller = -3.6502, Lag order = 15, p-value = 0.0278
## alternative hypothesis: stationary
```

```
tseries::adf.test(temp_monthL)
```

```
## Warning in tseries::adf.test(temp_monthL): p-value smaller than printed p-value
```

```
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  temp_monthL
## Dickey-Fuller = -11.294, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

The p-value for both data is smaller than 0.05, which we reject the null hypothesis. The data is not stationary, confirming previous findings.

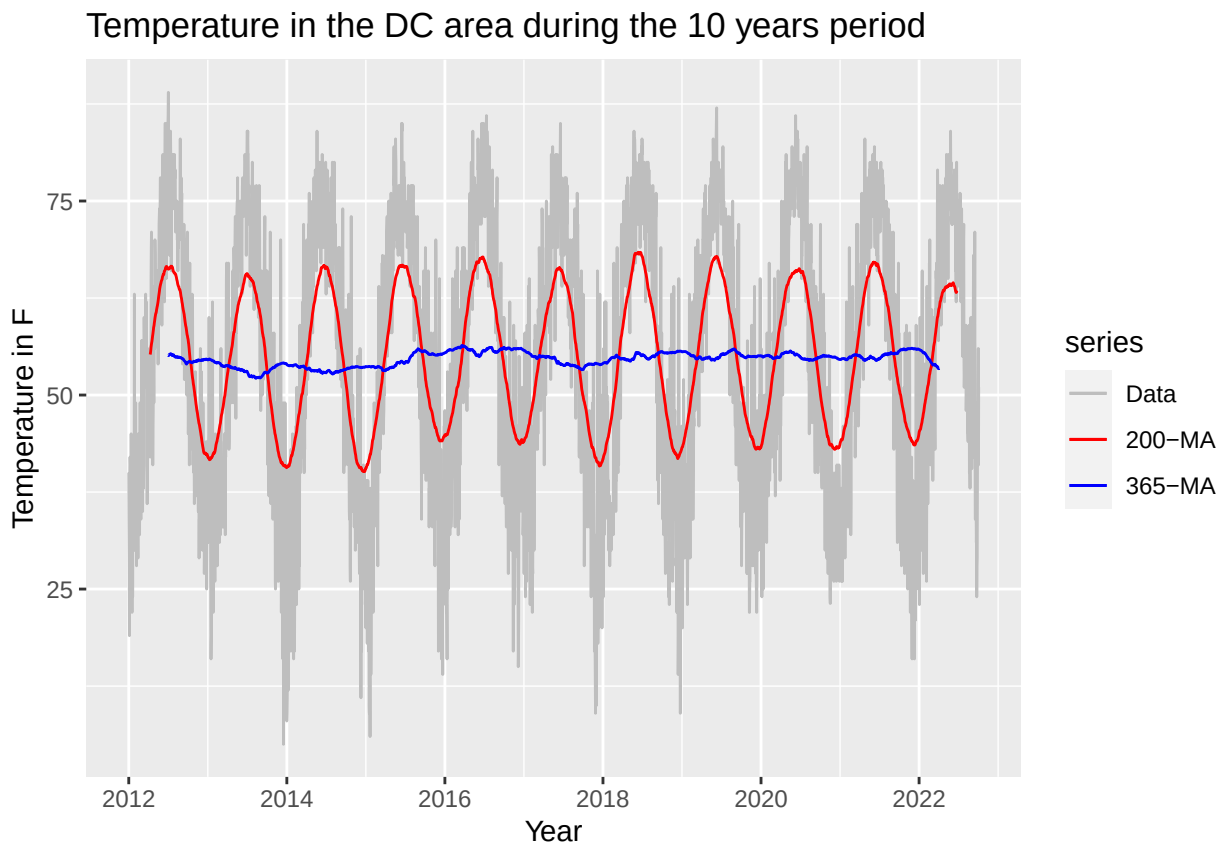
Problem 2:

a. use a SMA Smoothing to discuss about the underline patterns of `tempL` and `temp_monthL`. Use appropriate `order m` for `m-MA` smoothing for the 2 series. Discuss about the underline patterns.

```
autoplot(tempL, series="Data") +
  autolayer(ma(tempL, order=200, centre=FALSE), series="200-MA") +
  autolayer(ma(tempL, order=365, centre=FALSE), series="365-MA") +
  xlab("Year") + ylab("Temperature in F") +
  ggtitle("Temperature in the DC area during the 10 years period") +
  scale_colour_manual(values=c("Data"="grey","200-MA"="red","365-MA"="blue"),
                      breaks=c("Data","200-MA","365-MA"))
```

```
## Warning: Removed 199 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 364 row(s) containing missing values (geom_path).
```

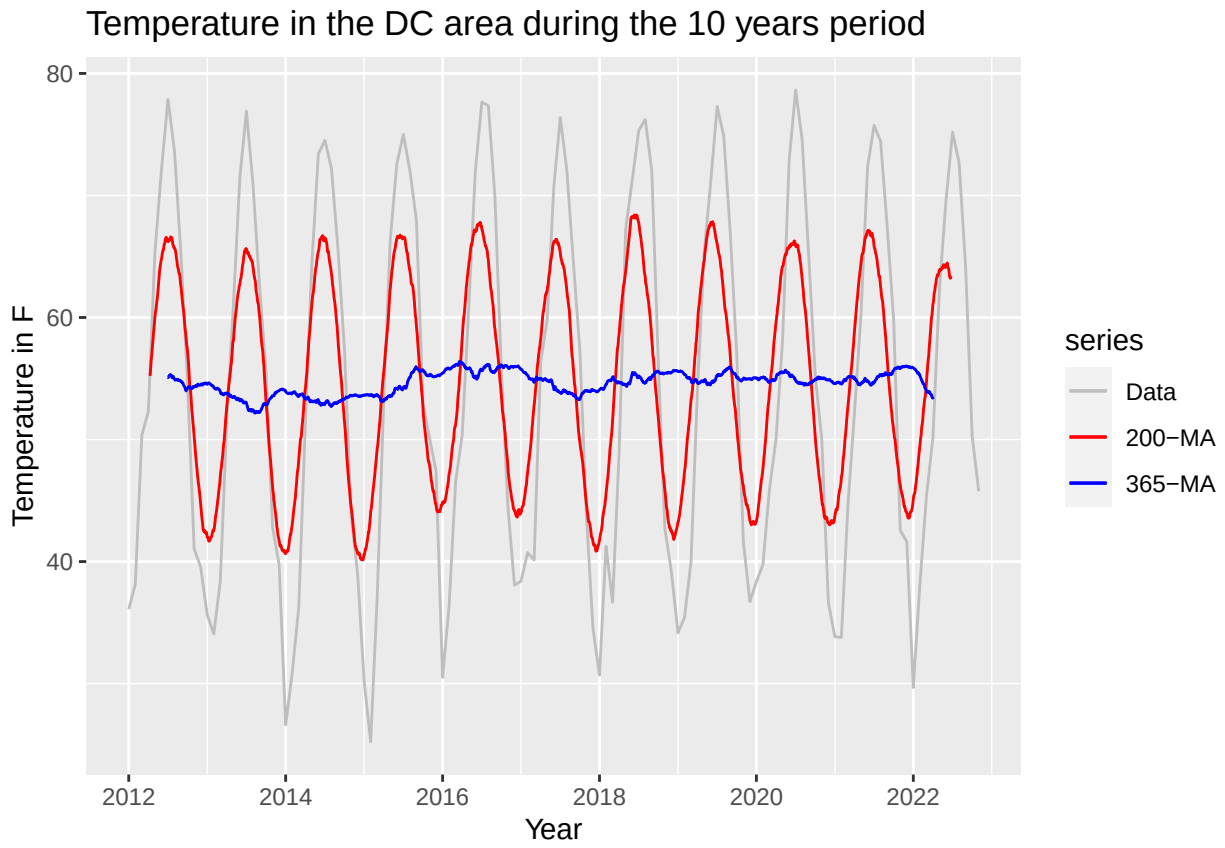


I chose 200 because 200 is a good indicator to detect the long term trend. Similarly, 365-MA since I want to see how the temperature changes yearly. From the graph, the temperature stays constanly at 55 from 2012 to 2022. Judging from the graph and the two MA, we can say that the data is stationary but there's seasonality.

```
autoplot(temp_monthL, series="Data") +
  autolayer(ma(tempL, order=200, centre=FALSE), series="200-MA") +
  autolayer(ma(tempL, order=365, centre=FALSE), series="365-MA") +
  xlab("Year") + ylab("Temperature in F") +
  ggtitle("Temperature in the DC area during the 10 years period") +
  scale_colour_manual(values=c("Data"="grey","200-MA"="red","365-MA"="blue"),
                      breaks=c("Data","200-MA","365-MA"))
```

```
## Warning: Removed 199 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 364 row(s) containing missing values (geom_path).
```



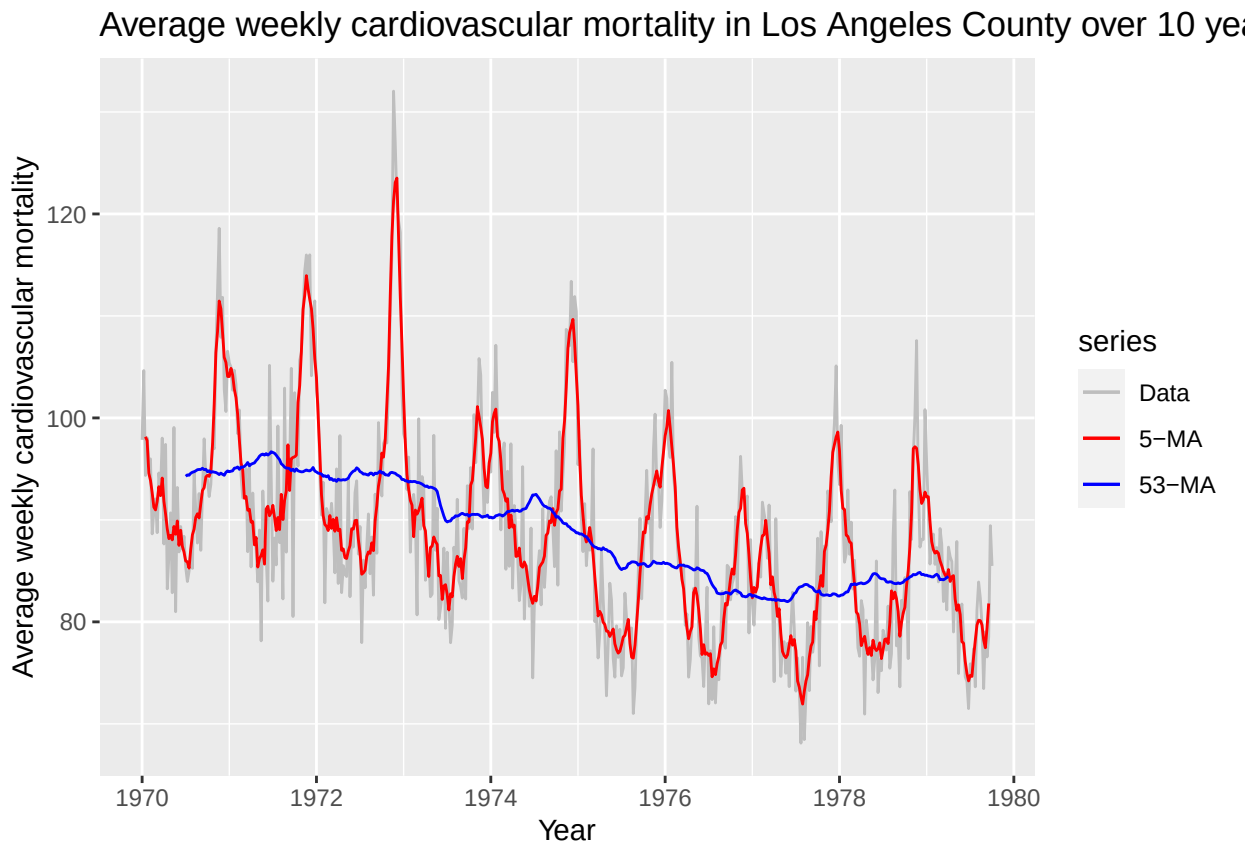
The data is stationary but there's seasonality.

b. For `"cmort"` data from `{astsa}` package, produce a five-point moving average that helps bring out the seasonal component and a 53-point moving average that helps bring out the (negative) trend in cardiovascular mortality. Plot both lines in the same graph and comment/discuss.

```
autoplot(cmort, series="Data") +
  autolayer(ma(cmort, order=5, centre=FALSE), series="5-MA") +
  autolayer(ma(cmort, order=53, centre=FALSE), series="53-MA") +
  xlab("Year") + ylab("Average weekly cardiovascular mortality") +
  ggtitle("Average weekly cardiovascular mortality in Los Angeles County over 10 years") +
  scale_colour_manual(values=c("Data"="grey","5-MA"="red","53-MA"="blue"),
                      breaks=c("Data","5-MA","53-MA"))
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 52 row(s) containing missing values (geom_path).
```



The larger moving average 53-MA (in blue) is smoother than the original data and captures the main movement of the time series without all of the minor fluctuations. The 5-MA is also smoother than the original data but not as smooth as the 53-MA. A larger order of moving average makes a smoother curve. We can see over time, after taking out the seasonality component, the average weekly cardiovascular mortality decreases. It is good and understandable. The cardiovascular mortality is the number of people that die from cardiovascular disease. As medical and technology advance, treatments and doctor's skills also get better. Thus, less people die from disease.