

Time_Series_Analysis

Dharmi Malde

2023-09-19

Getting Data

```
data = read.csv(file.choose("Time_Series_Data"))
ts_data = ts( data$Mean,start=2019,frequency = 365)
```

Installing imp library

```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union
```

```
library(fpp3)
```

```
## -- Attaching packages ----- fpp3 0.5 --
```

```
## v tibble      3.2.1      v tsibbledata 0.4.1
## v dplyr       1.1.3      v feasts      0.3.1
## v tidyr       1.3.0      v fable       0.3.3
## v tsibble     1.1.3      v fabletools  0.3.3
```

```
## -- Conflicts ----- fpp3_conflicts --
```

```
## x lubridate::date()      masks base::date()
## x dplyr::filter()       masks stats::filter()
## x tsibble::intersect()  masks base::intersect()
## x tsibble::interval()   masks lubridate::interval()
## x dplyr::lag()          masks stats::lag()
## x tsibble::setdiff()    masks base::setdiff()
## x tsibble::union()      masks base::union()
```

```
library(tsibble)
library(stats)
```

Converting to time variable

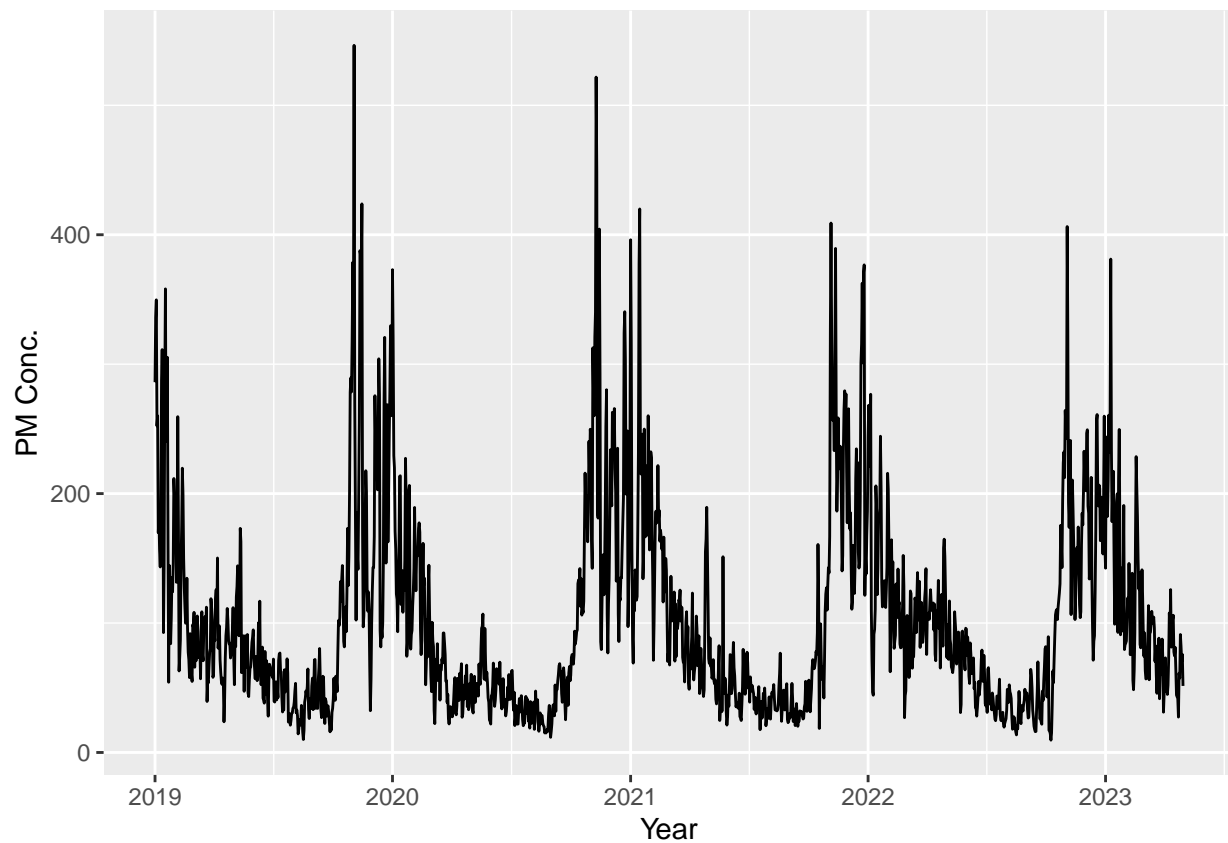
```
data$Dates <- as.Date(data$Dates)
```

Plotting time series data

```
ggplot(data)+
  geom_line(aes(x=data$Dates,y=data$Mean))+
  xlab("Year") +
  ylab("PM Conc.")
```

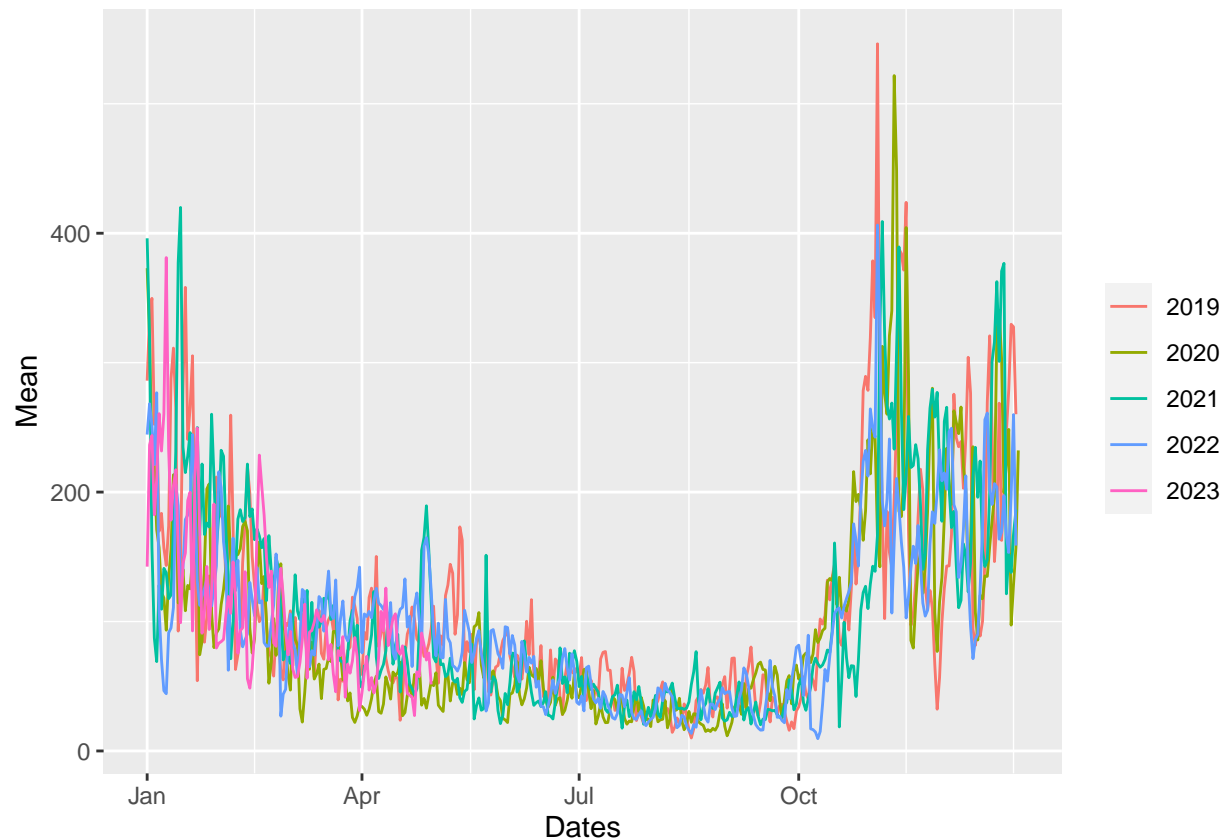
```
## Warning: Use of 'data$Dates' is discouraged.
## i Use 'Dates' instead.
```

```
## Warning: Use of 'data$Mean' is discouraged.
## i Use 'Mean' instead.
```



```
gg_season(as_tsibble(data))
```

```
## Using 'Dates' as index variable.  
## Plot variable not specified, automatically selected 'y = Mean'
```

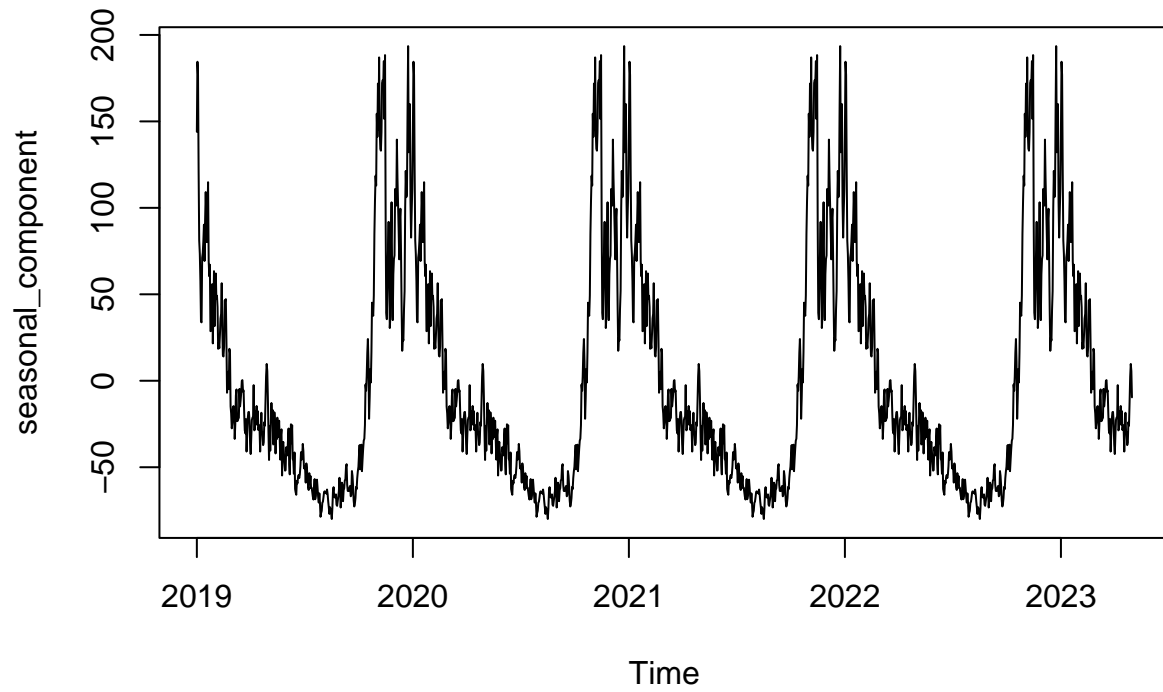


```
# Decomposition of Time Series
```

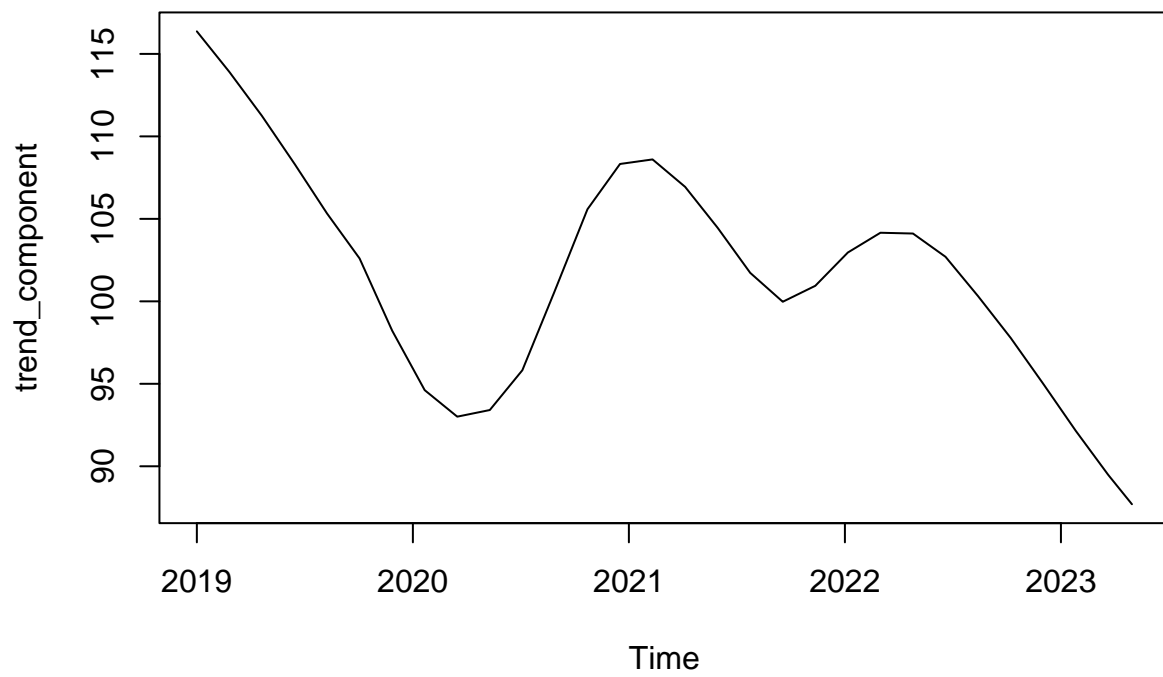
```
stl_result = stl(ts_data, s.window = "periodic")
```

```
seasonal_component = stl_result$time.series[, "seasonal"]  
trend_component = stl_result$time.series[, "trend"]  
remainder_component = stl_result$time.series[, "remainder"]
```

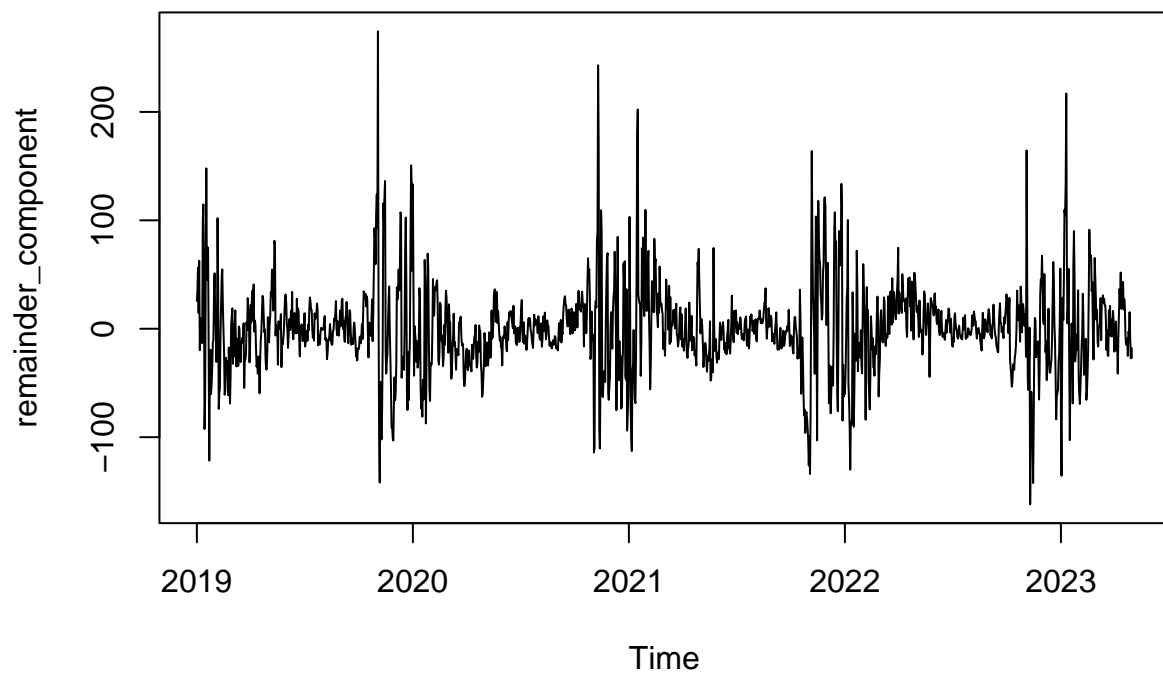
```
plot(seasonal_component)
```



```
plot(trend_component)
```



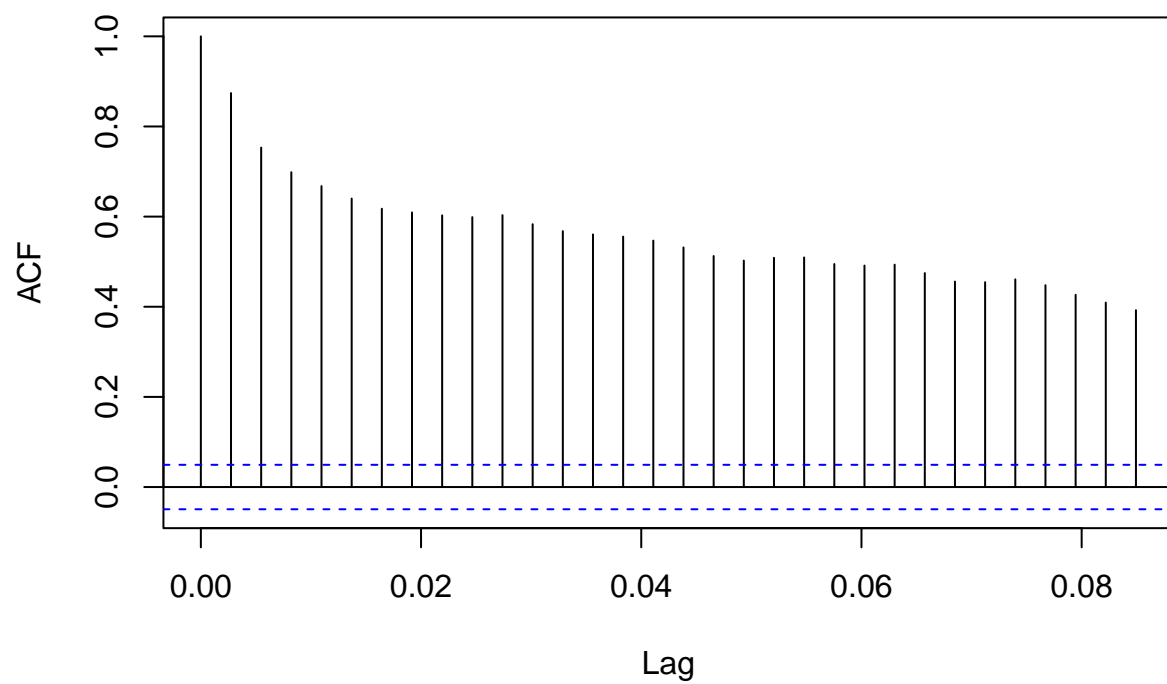
```
plot(remainder_component)
```



ACF and PACF graphs

```
acf(ts_data)
```

Series ts_data



```
pacf(ts_data)
```

Series ts_data

