Assignment 7: Time Series Analysis

Ying Liu

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

2. Import the ten datasets from the Ozone_TimeSeries folder in the Raw data folder. These contain ozone concentrations at Garinger High School in North Carolina from 2010-2019 (the EPA air database only allows downloads for one year at a time). Import these either individually or in bulk and then combine them into a single dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
#1
library(tidyverse)
library(dplyr)
library(readxl)
library(lubridate)
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.2.2
```

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.2.2
```

```
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.2.2
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.2
```

```
library(tidyverse)

#import datasets
GaringerOzone <- list.files(path = "C:/Users/Alina/Desktop/DUKE_22FALL/872/EDA-Fall2022/Data/Raw/Ozone_T
pattern = "*.csv", full.names = TRUE) %>%
 lapply(read_csv) %>%
 bind_rows
ncol(GaringerOzone)
```

```
## [1] 20
```

```
nrow(GaringerOzone)
```

```
## [1] 3589
```

```
# Set theme
mytheme <- theme_classic(base_size = 10) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

Wrangle

3. Set your date column as a date class.

4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.

5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need to fill in any missing days with NA. Create a new data frame that contains a sequence of dates from 2010-01-01 to 2019-12-31 (hint: `as.data.frame(seq())`). Call this new data frame Days. Rename the column name in Days to "Date".

6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined data frame GaringerOzone.

```
# 3
GaringerOzone$Date<-as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
# 4
GaringerOzoneProcessed<-
  GaringerOzone%>%
```

```
  select(1,5,7)%>%
  rename_at( 2, ~"Ozone_Concentration" )
# 5
Days<- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"),by = "days"))
Days<-
  Days%>%
  rename_at( 1, ~"Date")
# 6
GaringerOzoneProcessed <- left_join(Days,GaringerOzoneProcessed, by = c("Date"))

ncol(GaringerOzoneProcessed)
```

```
## [1] 3
```

```
nrow(GaringerOzoneProcessed)
```

```
## [1] 3652
```

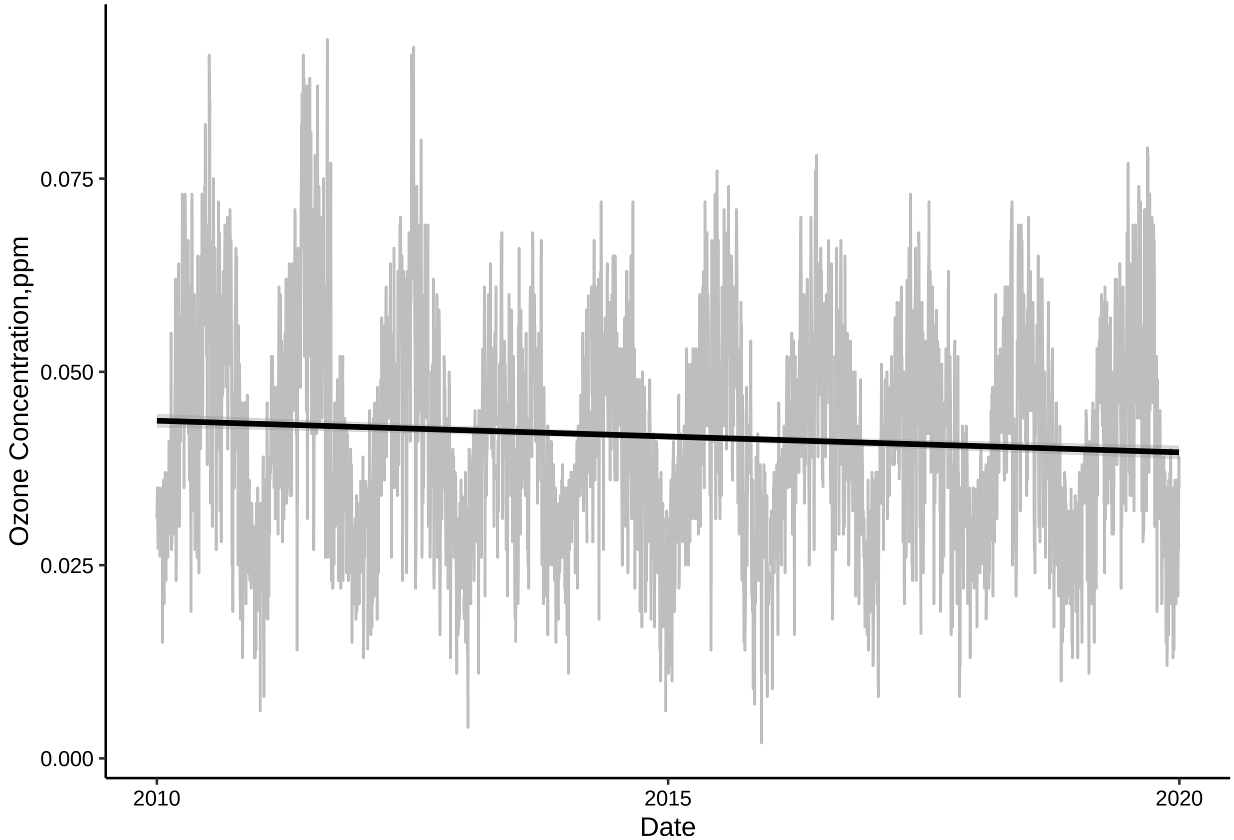
Visualize

7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
ggplot(GaringerOzoneProcessed,aes(x=Date,y=Ozone_Concentration))+
  geom_line(color="gray")+
  geom_smooth(method="lm",color="black")+
  labs(y="Ozone Concentration,ppm")+
  mytheme
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 63 rows containing non-finite values (stat_smooth).
```



Answer:slowly decreasing

Time Series Analysis

Study question: Have ozone concentrations changed over the 2010s at this station?

8. Use a linear interpolation to fill in missing daily data for ozone concentration. Why didn't we use a piecewise constant or spline interpolation?

```
#8
#see how many NAs
summary(GaringerOzoneProcessed$Ozone_Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
GaringerOzoneProcessed$Ozone_Concentration<- na.approx(GaringerOzoneProcessed$Ozone_Concentration)

summary(GaringerOzoneProcessed$Ozone_Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

```
#note that all NAs are gone
```

Answer: because ozone concerntration has a linear trend of decreasing, so using the nearest neighbor number as piecewise constant or use a quadratic as the spline can't reflect the trend.

9. Create a new data frame called `GaringerOzone.monthly` that contains aggregated data: mean ozone concentrations for each month. In your pipe, you will need to first add columns for year and month to form the groupings. In a separate line of code, create a new Date column with each month-year combination being set as the first day of the month (this is for graphing purposes only)

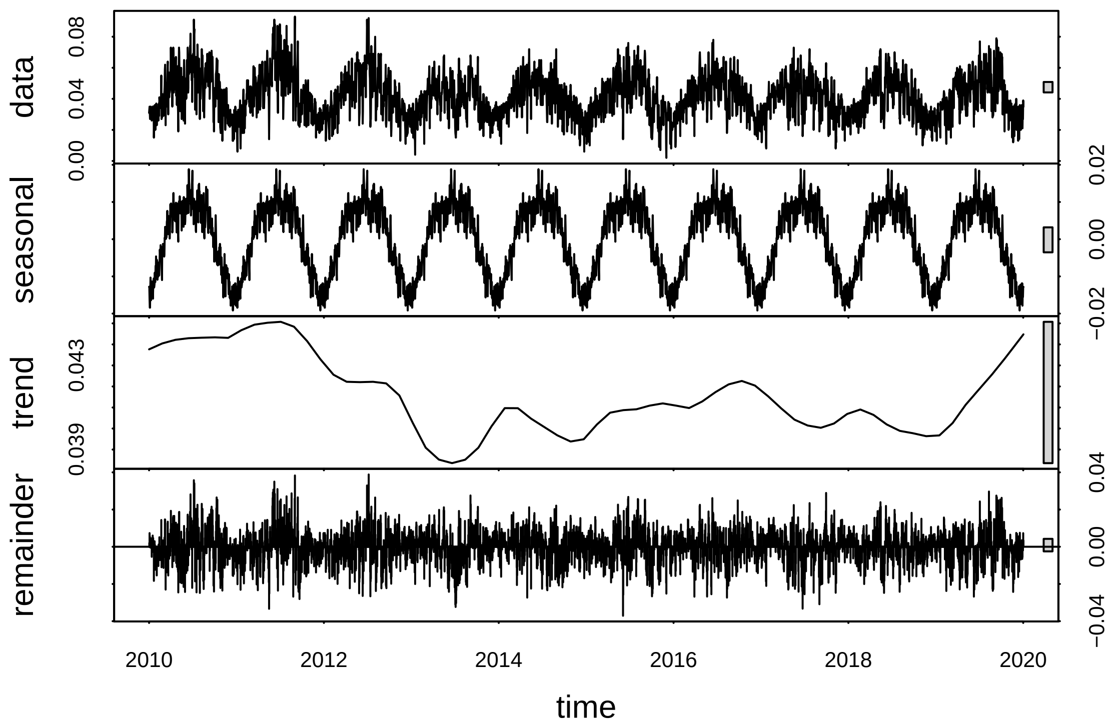
```
#9
GaringerOzone.monthly<-
  GaringerOzoneProcessed%>%
  separate(Date,into = c('year', 'month', 'day'))%>%
  group_by(month)%>%
  mutate(month = my(paste0(month,"-",year)))%>%
  summarise( meanOzone = mean(Ozone_Concentration))
```

10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

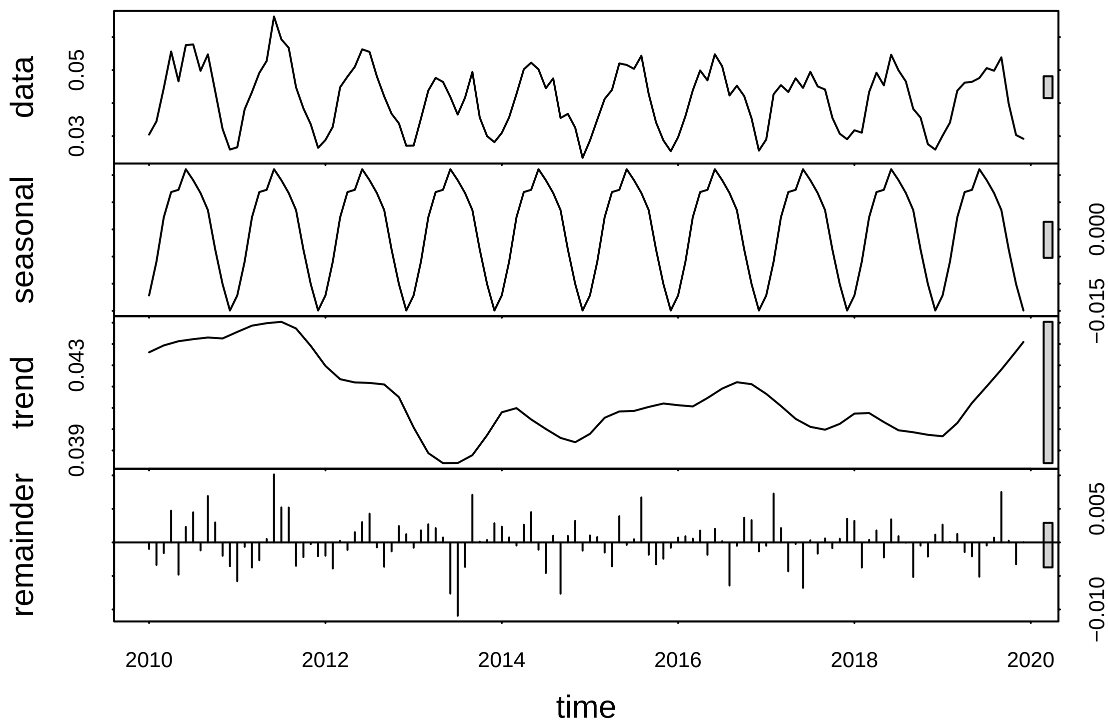
```
#10

GaringerOzone.daily.ts <- ts(GaringerOzoneProcessed$Ozone_Concentration, start = c(2010,1,1), frequency

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanOzone, start = c(2010,1,1),frequency = 12)
```

11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11
#decompose
GaringerOzone.daily_decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(GaringerOzone.daily_decomp)
```




```r
GaringerOzone.monthly_decomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
plot(GaringerOzone.monthly_decomp)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

```
#12

# run SMK test
GaringerOzone.monthly_trend1 <- trend::smk.test(GaringerOzone.monthly.ts)
# Inspect results
GaringerOzone.monthly_trend1
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

```
summary(GaringerOzone.monthly_trend1)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
```

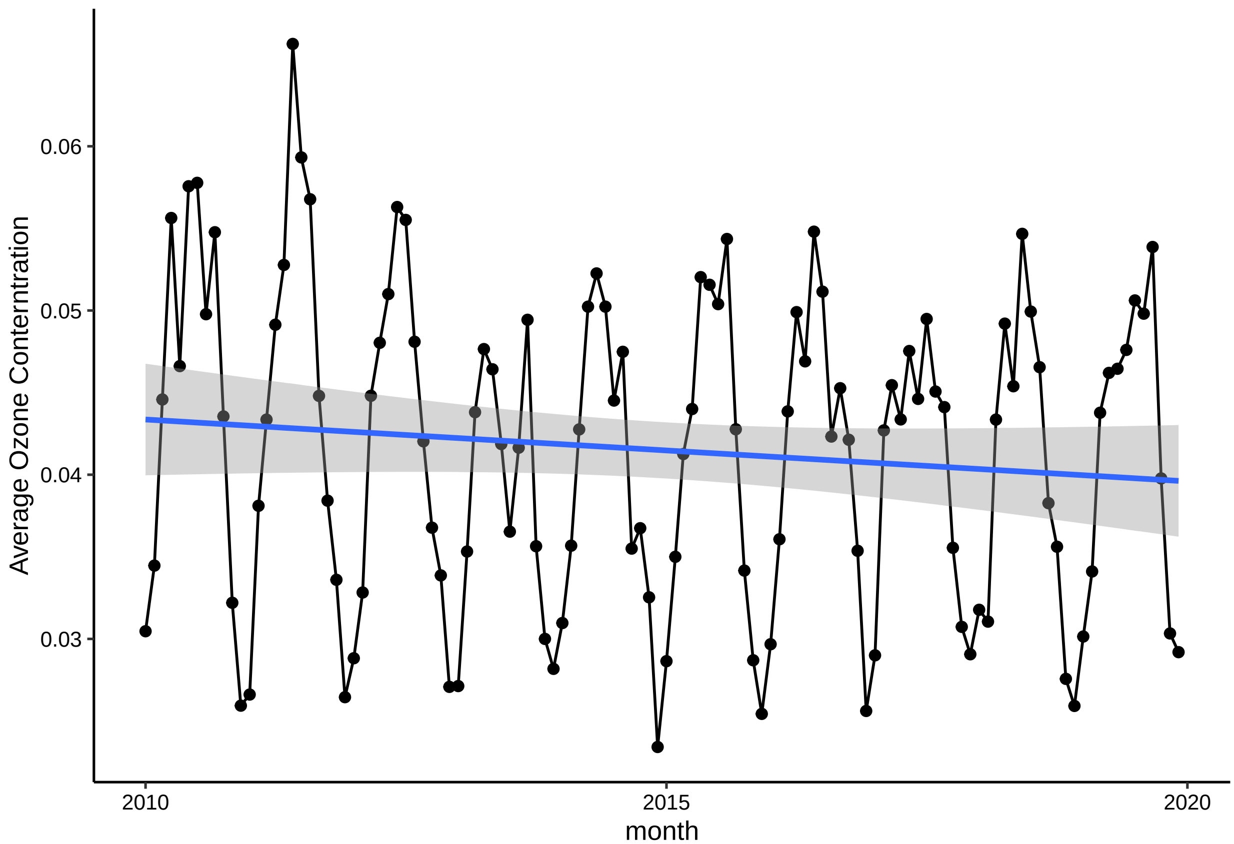
```
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                         S  varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 12:   S = 0  11  125  0.244  0.894  0.37109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The decomposed plot show clear periodical trend though, much more obvious than the daily fluctuation.

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13
#Visualization
GaringerOzone.monthly_plot <-
ggplot(GaringerOzone.monthly, aes(x = month, y = meanOzone)) +
  geom_point() +
  geom_line() +
  ylab("Average Ozone Conterntration") +
  geom_smooth( method = lm )+
  mytheme
print(GaringerOzone.monthly_plot)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: The mean ozone concerntration showed clear seasonal fluctuation, featuring more small changes in summer and stable low in winter. Overall, the mean ozone concerntration is slowly decreaing in the last ten years.

15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
#15
#extract the components of each series and turn them into data frames.

GaringerOzone.monthly_Components <- as.data.frame(GaringerOzone.monthly_decomp$time.series[,1:3])
GaringerOzone.unseasonal<- mutate(GaringerOzone.monthly_Components,
        Observed = GaringerOzone.monthly$meanOzone,
        Date = GaringerOzone.monthly$month)
#16

GaringerOzone.unseasonal.ts <- ts(GaringerOzone.unseasonal$Observed, start = c(2010-01-01), frequency =
```

```r
#run MK test
GaringerOzone.unseasonal_trend2 <- trend::mk.test(GaringerOzone.unseasonal.ts)
# Inspect results
GaringerOzone.unseasonal_trend2
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.unseasonal.ts
## z = -0.95947, n = 120, p-value = 0.3373
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -4.240000e+02  1.943647e+05 -5.939207e-02
```

```r
summary(GaringerOzone.unseasonal_trend2)
```

```
##             Length Class  Mode
## data.name   1      -none- character
## p.value     1      -none- numeric
## statistic   1      -none- numeric
## null.value  1      -none- numeric
## parameter   1      -none- numeric
## estimates   3      -none- numeric
## alternative 1      -none- character
## method      1      -none- character
## pvalg       1      -none- numeric
```

Answer: Seasonal Mann-Kendall trend test (Hirsch-Slack test) z = -1.963, p-value = 0.04965 alternative hypothesis: true S is not equal to 0 sample estimates: S varS -77 1499 The s in MSK is significantly bigger than SK, whcih means there are much more observation to arrive at more stastically significant results. p-value in SMK is less than0.05, which means we need to reject non hypothesis;the p-value in MK is more than 0.05, which means accept non hypothesis. Both have the same meaning that there are seasonal trend with the monthly mean ozone data.