Assignment 7: Time Series Analysis

Emma Brentjens

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
##checking working directory
getwd()
```

```
## [1] "/home/guest/R/EDA-Fall2022"
```

```
##loading packages
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
library(ggplot2)
library(Kendall)

##creating and setting theme
Emma_theme <- theme_linedraw() +
  theme(axis.text = element_text(color = "black", size = 10), legend.position = "right")

theme_set(Emma_theme)
```

```r
#2
##importing datasets
O3_2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                    stringsAsFactors = T)

O3_2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                    stringsAsFactors = T)

O3_2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                    stringsAsFactors = T)

O3_2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                    stringsAsFactors = T)

O3_2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                    stringsAsFactors = T)

O3_2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                    stringsAsFactors = T)

O3_2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                    stringsAsFactors = T)

O3_2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                    stringsAsFactors = T)

O3_2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                    stringsAsFactors = T)

O3_2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                    stringsAsFactors = T)

##combining datasets
GaringerOzone <- rbind(O3_2010, O3_2011, O3_2012, O3_2013, O3_2014, O3_2015, O3_2016,
                       O3_2017, O3_2018, O3_2019)

#View(GaringerOzone)
```

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
##changing class of date column
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format="%m/%d/%Y")
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

```
#4
##selecting columns
GaringerOzone_2 <- select(GaringerOzone, c("Date", "Daily.Max.8.hour.Ozone.Concentration",
                                           "DAILY_AQI_VALUE"))
#View(GaringerOzone_2)


#5
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), "days"))
#View(Days)

colnames(Days)[1] <- "Date"


#6
##combining data frames
GaringerOzone <- left_join(Days, GaringerOzone_2, by = "Date")
#View(GaringerOzone)
dim(GaringerOzone)
```

```
## [1] 3652    3
```

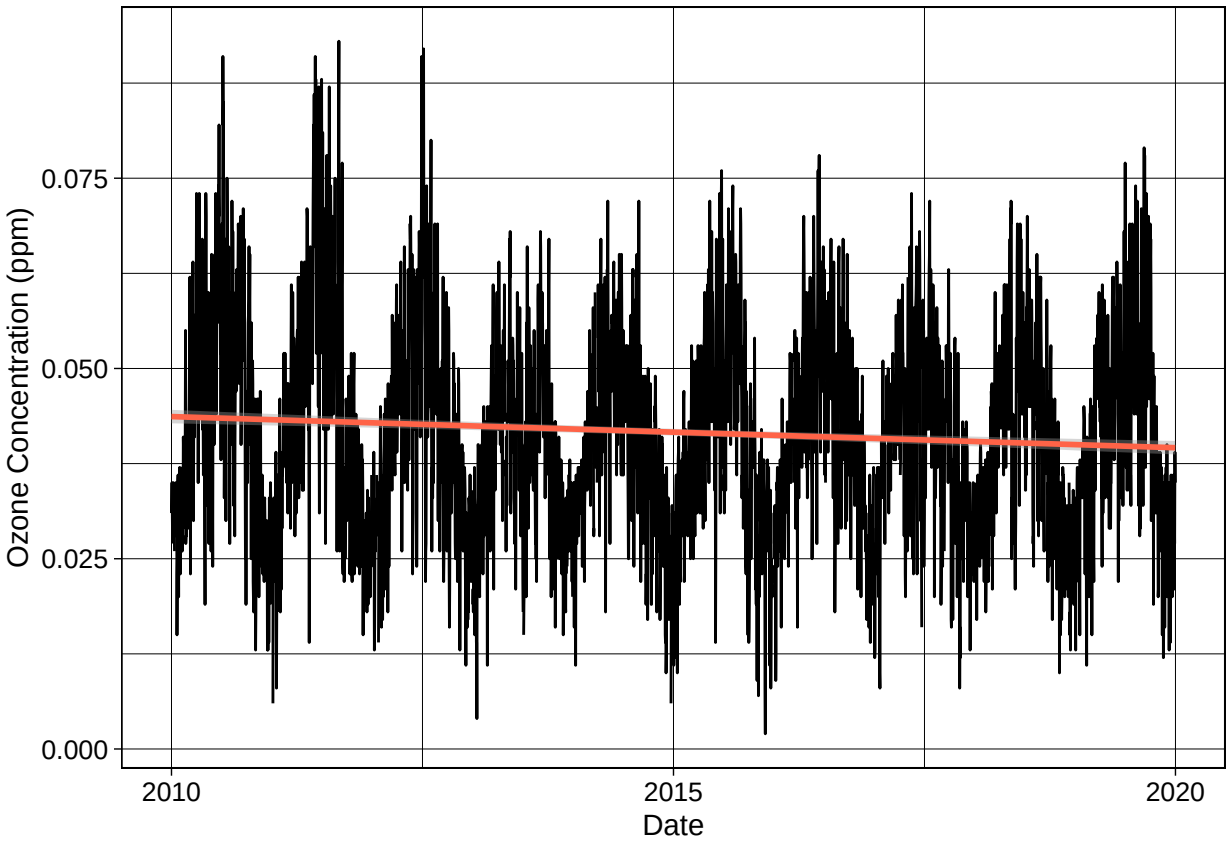
Visualize

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
##creating line plot
GaringerOzone_lineplot <- ggplot(GaringerOzone, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method=lm, color="tomato") +
  ylab("Ozone Concentration (ppm)") +
  Emma_theme

GaringerOzone_lineplot
```

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

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



Answer: There appears to be a slight negative trend in ozone over time, but this relationship does not seem to be very strong.

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
##filling in NAs
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <-
  na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)

summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

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

Answer: We don't use the spline interpolation becuase the data do not fit a quadratic distribution. The piecewise constant interpolation does not approximate the value, but sets NAs equal to the nearest value.

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
##creating new dataframe
GaringerOzone.monthly <-
  GaringerOzone %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  group_by(Year, Month) %>%
  summarize(mean_monthly_O3 = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  mutate(month_year = my(paste(Month, "-", Year)))
```

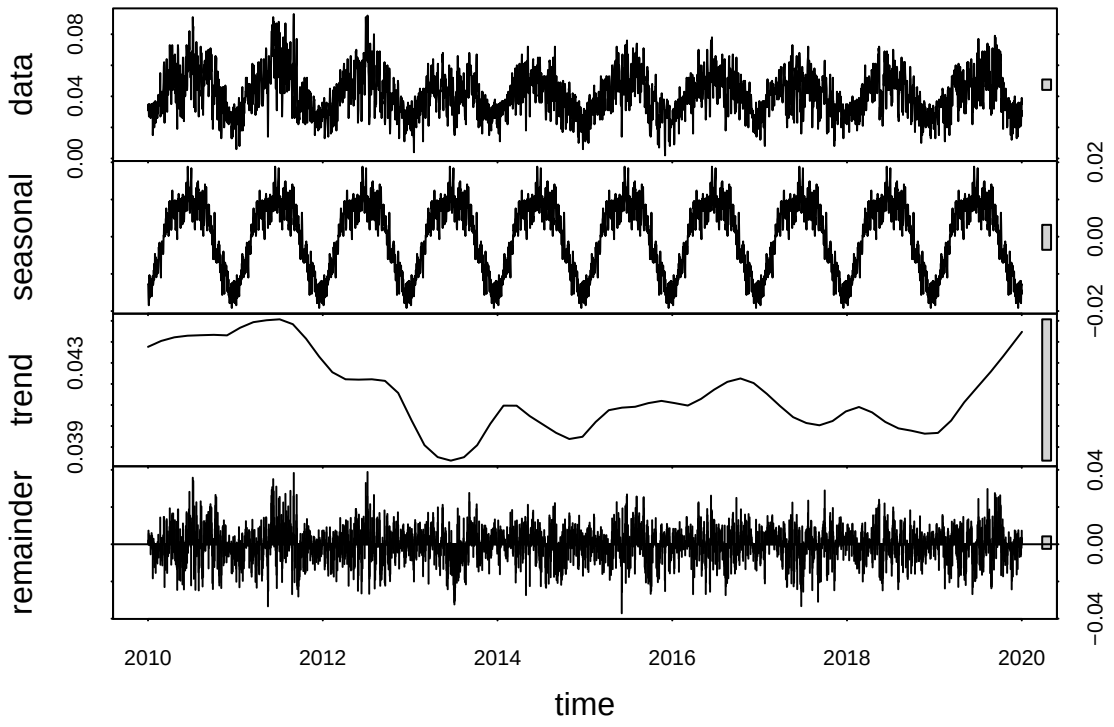
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

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.

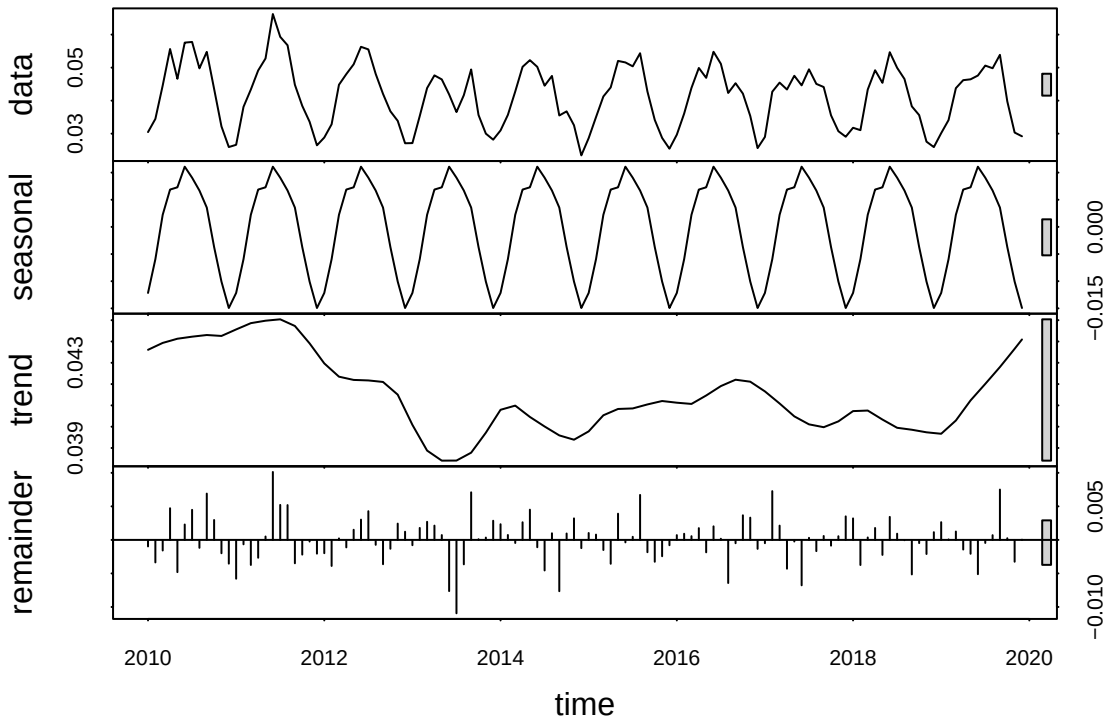
```
#10
##creating time series objects
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010, 01), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_monthly_O3,
                               start = c(2010, 01), frequency = 12)
```

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

```
#11
##Decomposing time series
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.decomp)
```




```
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?

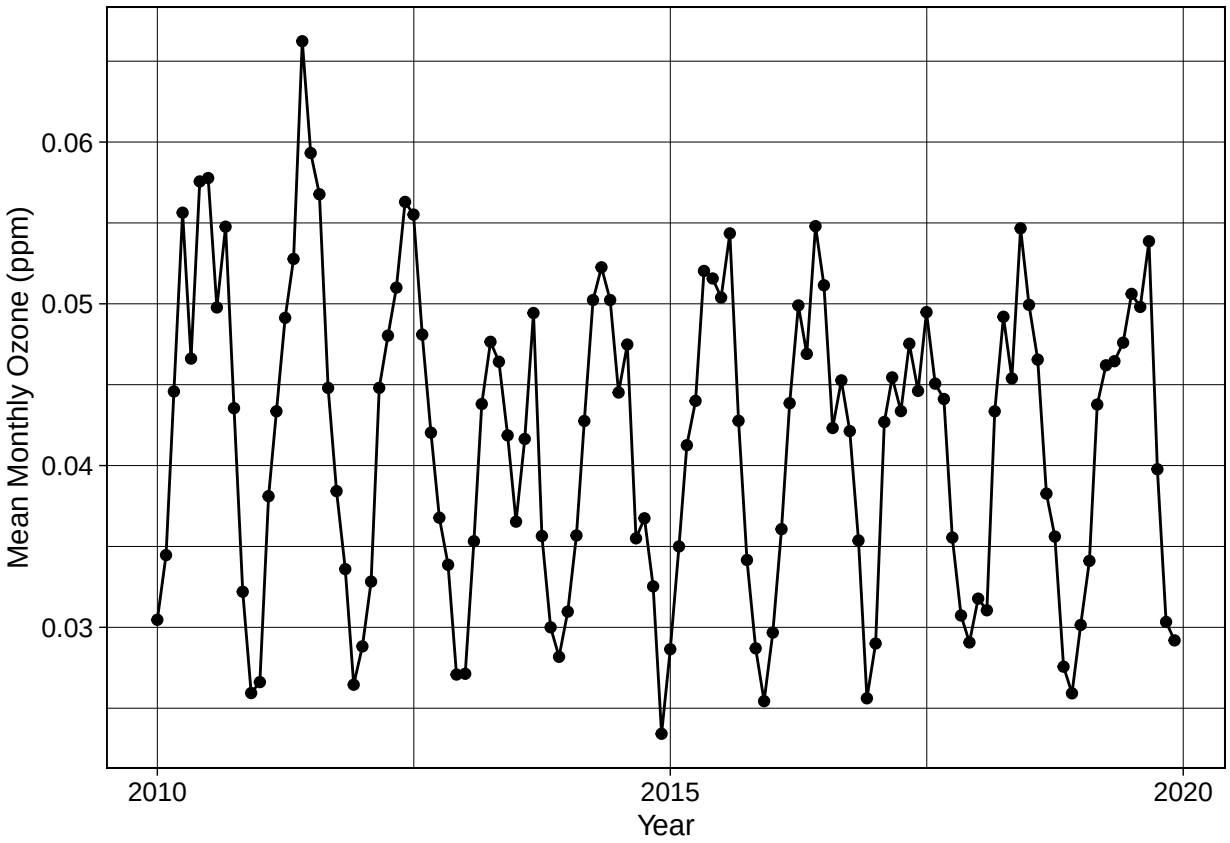
```
#12
##Mann-Kendall test
SeasonalMannKendall(GaringerOzone.monthly.ts)
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Based on the decomposition plot, it seems the data are mainly driven by seasonality, so it makes sense to account for seasonality in our analysis.

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
##creating monthly ozone plot
mean_monthly_O3_plot <- ggplot(GaringerOzone.monthly, aes(x=month_year, y=mean_monthly_O3)) +
  geom_point() +
  geom_line() +
  ylab("Mean Monthly Ozone (ppm)") +
  xlab("Year")

mean_monthly_O3_plot
```



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 null hypothesis for the seasonal Mann-Kendall test is that there is no trend over time. Therefore, our analysis shows that there is a seasonal positive or negative trend in ozone concentrations over time ($p = 0.047$).

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
##Subtracting seasonal component
monthly_O3_no_season <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,2:3])

monthly_O3_no_season <- mutate(monthly_O3_no_season,
        Observed = GaringerOzone.monthly$mean_monthly_O3,
        Date = GaringerOzone.monthly$month_year)

#View(monthly_O3_no_season)


#16
##make new dataframe a time series
monthly_O3_no_season.ts <- ts(monthly_O3_no_season$Observed, start = c(2010, 01), frequency = 12)
```

```
MannKendall(monthly_O3_no_season.ts)
```

```
## tau = -0.0594, 2-sided pvalue =0.33732
```

Answer: The Mann-Kendall test on the non-seasonal ozone time series shows no trend in ozone values over time (p = 0.337), while the Seasonal Mann-Kendall test showed a significant seasonal trend (p = 0.047). The results of these tests further demonstrate the influence of seasonality on the data.