Assignment 7: Time Series Analysis

Joshua Frear

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
getwd()
```

```
## [1] "D:/Data_872/Environmental_Data_Analytics_2021"
```

```
library(tidyverse)
library(lubridate)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.0.4
```

```
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.0.4
```

```
#2 Import datasets

#To import all csv files in a folder quickly, we can use a for loop.
#The first variable includes the path, for the read.csv function later,
#while the second omits it to use as the name for the new data frame
o3_file_list <- list.files(path = "./Data/Raw/Ozone_TimeSeries/", full.names = TRUE)
```

```
o3_file_names <- list.files(path = "./Data/Raw/Ozone_TimeSeries/", full.names = FALSE)

for (i in 1:length(o3_file_list)) {
  file_name <- o3_file_names[i]
  file_df <- read.csv(o3_file_list[i])
  assign(x = file_name, value = file_df)
}

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)

#Combine the 10 datasets into one dataframe. This code chunk is fairly long and
#repetitive but it has the advantage of being clear.
library(dplyr)
GaringerOzone <- full_join(EPAair_O3_GaringerNC2010_raw.csv, EPAair_O3_GaringerNC2011_raw.csv) %>%
  full_join(EPAair_O3_GaringerNC2012_raw.csv) %>%
  full_join(EPAair_O3_GaringerNC2013_raw.csv )%>%
  full_join(EPAair_O3_GaringerNC2014_raw.csv) %>%
  full_join(EPAair_O3_GaringerNC2015_raw.csv) %>%
  full_join(EPAair_O3_GaringerNC2016_raw.csv) %>%
  full_join(EPAair_O3_GaringerNC2017_raw.csv) %>%
  full_join(EPAair_O3_GaringerNC2018_raw.csv) %>%
  full_join(EPAair_O3_GaringerNC2019_raw.csv)
```

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 Format date column as a date for time series
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4 Remove all columns except date, Ozone Conc (ppm), and Daily AQI
GaringerOzone <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
#By setting the seq() as a vector first, we avoid a bad column name in Days dataframe.
Date <- seq(from = GaringerOzone$Date[1], to = GaringerOzone$Date[3589], by = 1)
Days <- as.data.frame(Date)

# 6
```

```
GaringerOzone <- left_join(Days, GaringerOzone)
```

```
## Joining, by = "Date"
```

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?

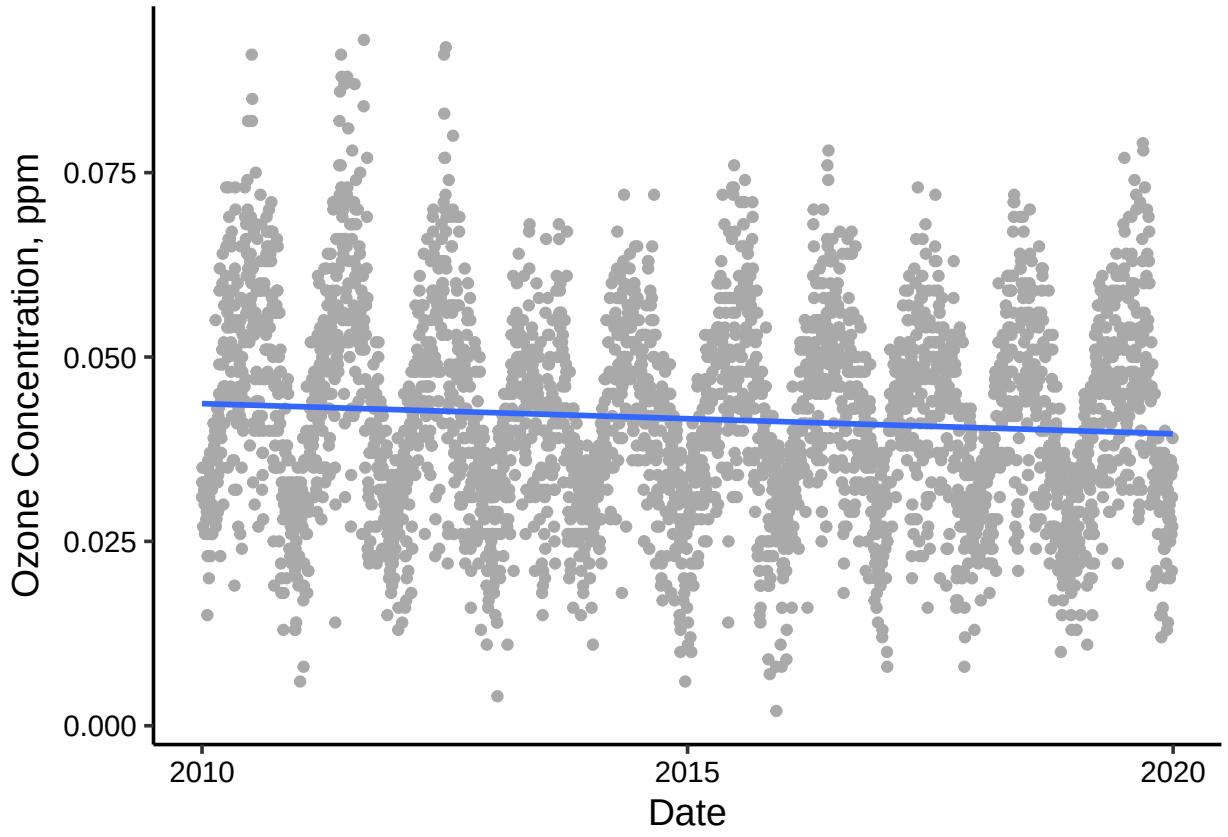
```
#7
library(ggplot2)

plot1 <- ggplot(data = GaringerOzone,
                aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_point(color = "darkgray") + geom_smooth(method = "lm", se = FALSE) +
  ylab(expression("Ozone Concentration, ppm")) + lims()
print(plot1)
```

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

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

```
## Warning: Removed 63 rows containing missing values (geom_point).
```



Answer: It appears that ozone concentration is slightly decreasing over time.

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
#summary(GaringerOzone) reveals 63 NA values for both measurements
GaringerOzone <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %
  mutate(DAILY_AQI_VALUE = zoo::na.approx(DAILY_AQI_VALUE))
```

   Answer: I don't have a reason to believe that a cubic interpolation would be needed in a case like this. In general, use simple linear interpolations unless they prove insufficient.

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)

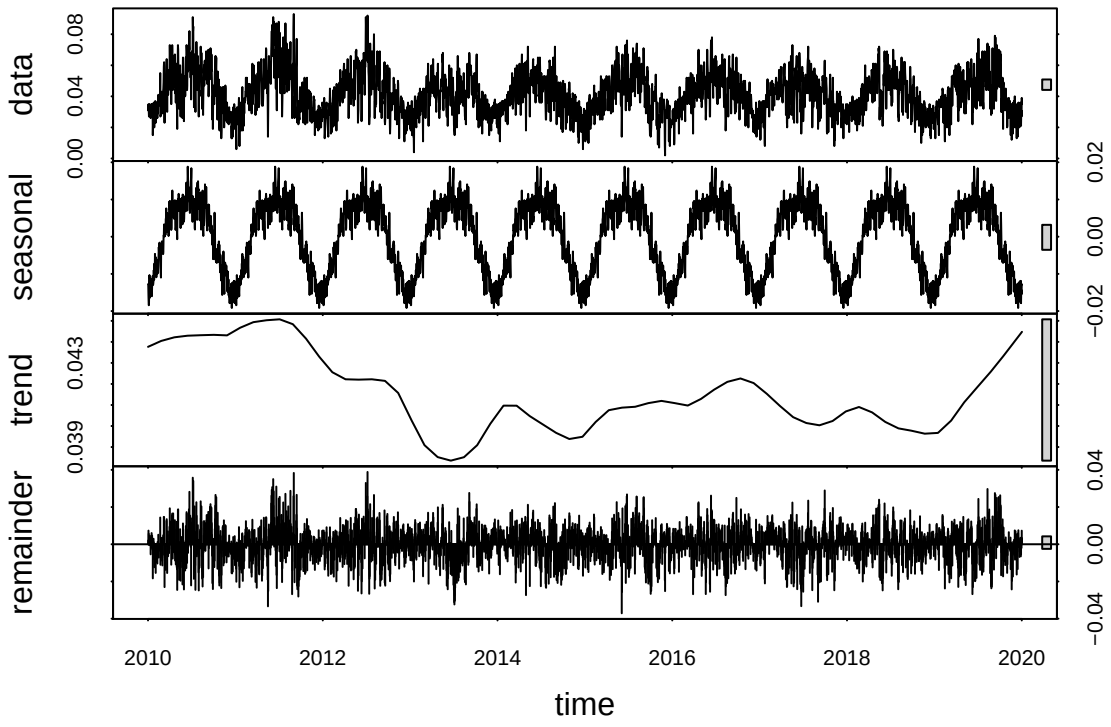
```
#9
#this method uses floor_date to set every date to the first of the month.
#The mean for each date then becomes the mean for a particular month.
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(floordate = floor_date(Date, unit = "month")) %>%
  group_by(floordate) %>%
  summarize(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration), mean_AQI = mean(DAILY_AQI_VALUE))
```

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.

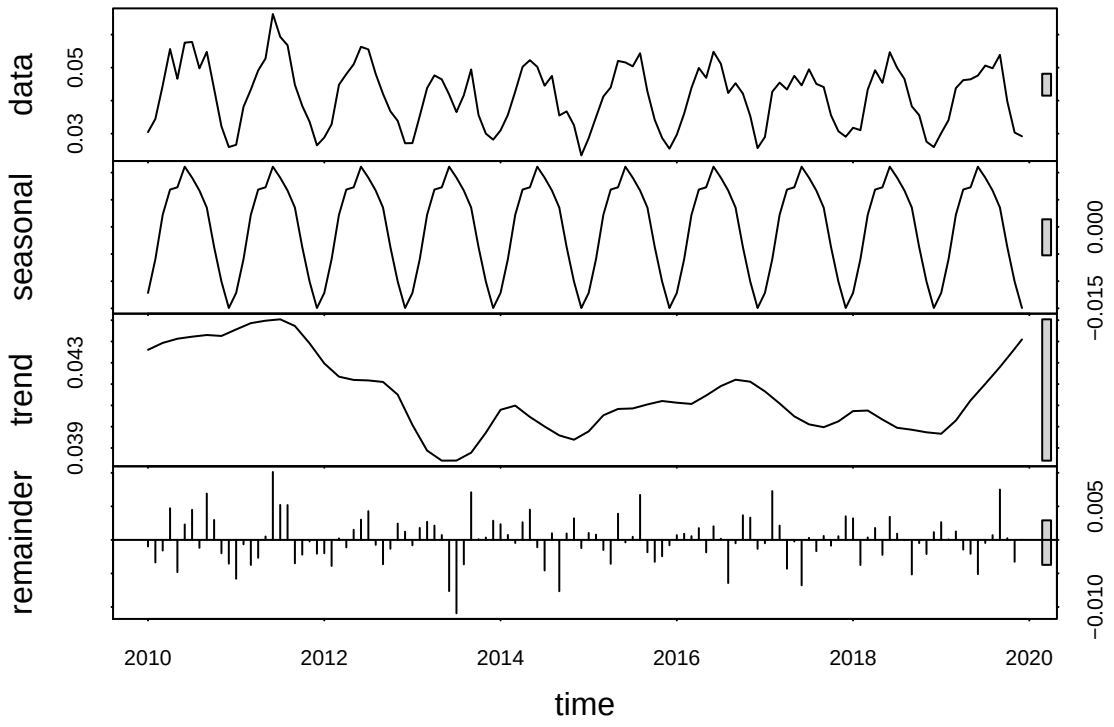
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,1), frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                               start = c(2010,1), frequency = 12)
```

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

```
#11
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
# Visualize the decomposed series.
plot(GaringerOzone.daily.decomposed)
```




```
plot(GaringerOzone.monthly.decomposed)
```



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

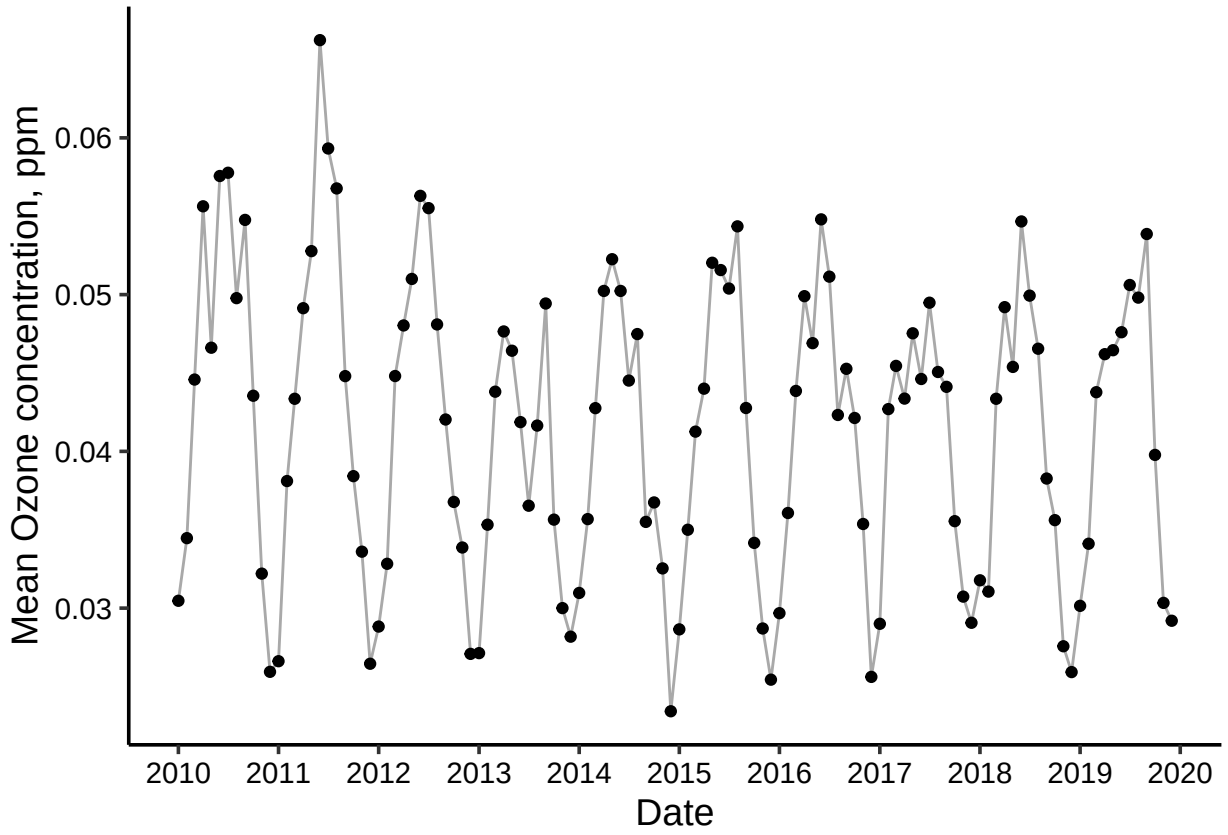
```
#12
monthly_ozone_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(monthly_ozone_trend)
```

```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: There is a clear seasonal trend in the decomposition of the time series earlier. Mann-Kendall is fine because we don't have to assume that the data is parametric.

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
ggplot(data = GaringerOzone.monthly,
       aes(x = floordate, y = mean_ozone)) + geom_line(color = "darkgray") +
  geom_point() + xlab("Date") + ylab("Mean Ozone concentration, ppm") +
  scale_x_date(date_breaks = "1 year",
               date_labels = c("2020", "2010", "2011", "2012", "2013", "2014", "2015",
                               "2016", "2017", "2018", "2019"))
```




```
#manually entered date_labels in c() out of order to have them line up on graph
```

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: Based on the seasonal Mann-Kendall test, we can confirm a decreasing monotonic trend of mean Ozone concentrations over time (tau < 0, p < 0.05).

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
# We can extract the components and turn them into data frames
GaringerOzone.monthly.Components <- as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

GaringerOzone.monthly.Components <- mutate(GaringerOzone.monthly.Components,
        Ozone = GaringerOzone.monthly$mean_ozone,
        Date = GaringerOzone.monthly$floordate)

GaringerOzone.monthly.Components$Ozone_noseason <-
  GaringerOzone.monthly.Components$Ozone - GaringerOzone.monthly.Components$seasonal

#16
GaringerOzone.ts2 <-
```

```
  ts(GaringerOzone.monthly.Components$Ozone_noseason, start = c(2010,1), frequency = 12)

Ozone_trend_noseason <- Kendall::MannKendall(GaringerOzone.ts2)
summary(Ozone_trend_noseason)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The 2-sided pvalue is smaller here, indicating a very high likelihood of a non-seasonal trend.